Feature-based shape transformation for polyhedral objects

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

Abstract

A new technique is presented for computing transformations between polyhedral objects. Here, the correspondence and the interpolation problems are considered jointly. The approach gives the animator high level control of the shape transformation by providing natural specification and interaction.

Keywords: Computer Animation, Computer-Aided Geometric Design, Interpolation, Deformation, Shape transformation.

1 Introduction

We are interested in simulating deformations of geometric shapes. The way the deformation is specified is important: it determines both the flexibility and the usability of the deformation tool. There exists many approaches to deform a 3D shape from an object using a modelization of a time-dependent deformation process (cf. [Bar84, TPBF87, PEF+90, CJ91, BD93]).
Our goal is to provide a general and intuitive tool that computes a shape transformation mainly specified by two shapes: the initial and the final shapes.
Such a tool is useful in both animation and design.

• In animation, it is well adapted to traditional techniques such as keyframing: the objects are given at each keyframe 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" ([BN92, Hug92]) and is often used in movies.

• In design, it is used to create new shapes by combining two objects ([CP89, KR91]).

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 [CP89, BN92, SG92, SGWM93]).
The 3-D methods can be classified into two families according to the type of information used to compute the intermediate shapes.

The volumic approaches [KR91, Hug92].
As the volumic information is used to compute the shape transformation, there is no restriction for the topological correspondence between the initial and the final shapes: for example, a torus can be transformed into a sphere.

- Kaul and Rossignac [KR91] define the internal points of intermediate polyhedral objects using linear interpolation based on Minkowski sums of the internal points 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.
- Hughes [Hug92] 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 during the transformation.

These two approaches are robust but there is no control over the intermediate shapes: only one shape transformation is computed from two original solids.

The boundary approaches [BU89, KPC91, KPC92, Par92].
In all the cases, the original solids are polyhedral objects and they have equivalent surface manifold topologies. For these methods, the shape transformation process is decomposed in two parts:

1. a correspondence process, where the correspondence between the two surfaces is established. As the surfaces are polyhedral, the process is reduced by computing the correspondence between the topological structures of the two objects. In fact, all the approaches build a vertex/edge/face network containing the topological structures of the original objects.

2. an interpolation process: once the correspondence is established, the in-between polyhedra are computed using the corresponding vertices. This is achieved 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 [KPC92]).

All the approaches consider the two processes separately and propose solutions to the correspondence problem.

- First of all, Bethel and Uselton [BU89] tackle the correspondence process mainly as a graph correspondence problem. As the geometry of the two polyhedra is not taken into account, the visual effects produced during the shape transformation are often not intuitive in the general case and are difficult to control. For instance, as the metric relationships between points are ignored, two closed points of the first object may be associated with two distant points of the second object. If this occurs randomly, it will result in a discontinuous transformation.

- Kent, Parent and Carlson [KPC91, KPC92] use the geometric information of the initial and the final shapes to build an adjacency graph containing their adjacency graphs. 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 in two steps:

  ⋆ a projection of each adjacency graph is done onto the surface of the unit sphere,

  ⋆ the common adjacency graph is computed by merging these projected graphs.

  They generalize their approach in [KPC92] mainly by extending the projection method to other classes of polyhedral models. In most of these cases the relationship between the geometry of the object and the mapping method seems less natural, since the correspondence process uses a "physical" approach which is abandoned during the interpolation process.

- Parent [Par92] uses a recursive subdivision process to build the common adjacency graph. In his method, the user can indicate 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.

Our opinion is that dissociating the correspondence from the interpolation process leads to a difficult control of the deformation of the shape during the interpolation. In a sense, the shape transformation problem has to be taken as a whole, even if two problems have to be solved.

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

- First, we solve jointly the correspondence and the interpolation processes.

- We do not built a common adjacency graph from the initial and the final adjacency graphs. A common parametrized polyhedral mesh is associated to the initial and final objects during a sampling process.
- We propose to the user the control of the "global" evolution of the deformation. This control is done 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, such that each point of the object can be attached to the axis without ambiguity. These objects can naturally be split into 3 sheets:

- two hemispherical parts, corresponding to the points that are attached to one extremity of the axis.
- one cylindrical part, corresponding to the points that are attached orthogonally to the axis, i.e. that belong to one normal cross-section of the axis.

Each of these parts admits a natural parametrization, respectively spherical and cylindrical. We shall refer, in this paper, to a more general cylindrical coordinate system, where the z-axis is replaced by a 3D curve (the axis of the object).

2 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 point of our method: using the fact that a three dimensional object is not defined by a particular polygonal mesh but by any mesh that captures its global geometry and its peculiar features, we choose to associate to the object the mesh corresponding to the parametrization that will be used both during the correspondence and the interpolation processes.

To do so, 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 to each discretization point. As in [LCJ93], we have chosen a rotation minimizing orthogonal frame along each 3D curve [Bis75, Klo86]. Each axis is used to define an appropriate parametrization of the associated object, such as described earlier. The shape transformation process will then consist in the interpolation of such parametrization. We propose here an algorithm decomposed in two main steps. Each of these steps will be explained in the following sections.

- Using the axes, the sampling of the objects are computed using the *natural* parametrizations. This sampling process must be followed by an adjustment of the sampling vertices/edges/faces to the *singular features* of each object. For polyhedral objects these features are sharp edges or conic points. This adjustment is of prime importance for a good approximation of the shapes.

  This process will be detailed in the next section.
- Then we interpolate simultaneously the parameter of each corresponding parametrization (i.e. a 3D frame for a spherical parametrization and a whole 3D curve for a cylindrical parametrization) and the local coordinates of each sampled objects with respect to these parametrizations.

  This will be described in section 4.

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.

3 Sampling and adjustment

3.1 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 following two different ways:

- they radiate from one of the two axis extremities, in the half-space determined by the normal cross-section of the axis at this extremity. They intersect the object at two hemispherical sheets. Given one extremity, each half-line is defined by two angular parameters, as shown in Figure 1.
- they belong to one normal cross-section of the axis. Here, the collection of intersection points compose 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 may vary in a rectangular domain. The sampling is thus obtained by mapping square lattices in the parameter spaces onto the object. By connecting the sampling points according to the square grid associated to the square lattice, 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). Let us point out that for the hemisphere, one edge of the associated square domain $[0, 2\pi(\times[0, \frac{\pi}{2}]$ is mapped on a single point.

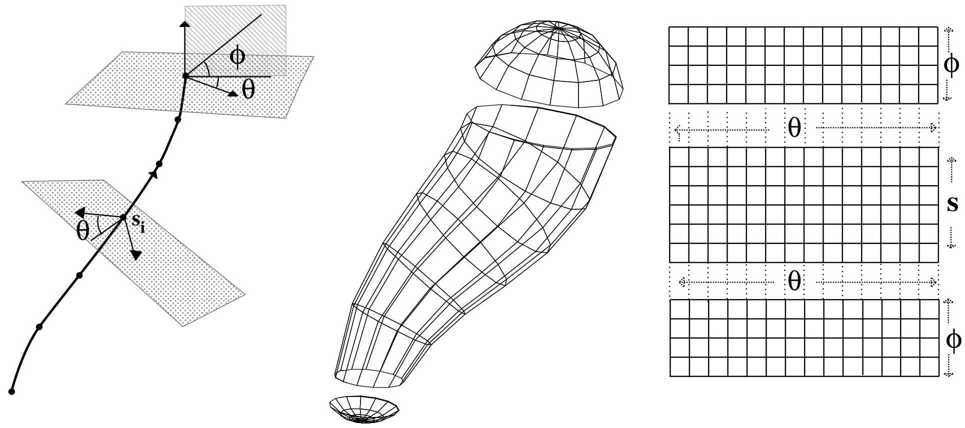

Figure 1: The three sheets

The sampling of an object is computed as follows:

- each vertex of the original object is examined and localized w.r.t. the parametrization induced by the axis. As in [LCJ93] this is done by finding the closest point on the axis.
- for each face $\mathcal{F}$ of the object, we compute the intersections with the sampling half-lines and $\mathcal{F}$. More precisely,
  - given the parameters of the vertices of $\mathcal{F}$, the face $\mathcal{F}$ is cut in three parts w.r.t. the three sheets (one or two may be empty),
  - inside each sheet, $\mathcal{F}$ is projected in the parameter space, using the parameters of its vertices previously evaluated. More precisely, the sampling parameters appearing in the projected region are located and the corresponding sampling half-lines are intersected with $\mathcal{F}$. This is done by a sweep, the length of the sweep being determined by the the extremal parameters of the vertices of $\mathcal{F}$:

    ⋆ for the cylindrical part, we cut $\mathcal{F}$ with every intermediate cross section and find the sampling parameters for each intersecting segment (see Figure 2).

    ⋆ for the hemispherical parts, the process is similar but the intermediate cross sections are the planes determined by the angular parameters θ (cf. Figure 1).

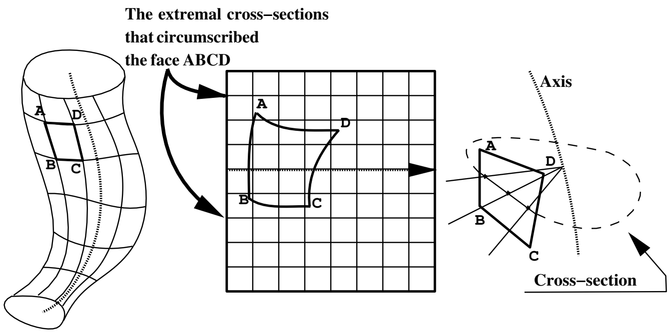

Figure 2: Image of a face in the parameter space

3.2 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. One can notice that the salient features (i.e. sharp edges and conic points, for a polyhedron) are of prime importance for the appearance of an object, since they mark (with the contour lines) the discontinuities of the light reflectance function.

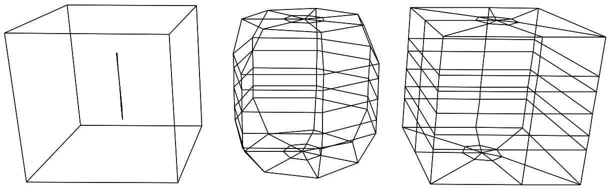

Figure 3: The sampling of a cube (center) and the same sampling after adjustment (right).

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

In order to minimize the topological change of the sampling mesh previously calculated, we choose to adjust the sampling mesh by locally deforming this mesh, moving its vertices and adding some diagonal edges to the rectangular faces. Thus we proceed in the following two steps:

- we find the sharp edges and conic points of the original object,
- we deform locally the sampling mesh: for each of these entities, we locate the vertices to be moved, compute their displacement and, if necessary, insert diagonal edges in the sampling mesh.

3.2.1 Sharp edges

A simple angle criterion between the adjacent faces to an edge allows us to select accurately the sharp edges. A similar criterion can be applied to conic points. An interactive module is also proposed to the user in order to give him greater freedom of choice. We obtain from the above two lists of salient features: a list of sharp edges and a list of conic points.

3.2.2 Deforming the sampling mesh

Deforming the sampling mesh consists of fitting the vertices and the edges of the sampling mesh to the sharp edges and conic points and adding diagonal edges in the sampling mesh in such a way that, at the end:

- each sharp edge can be decomposed in a set of disjoint edges of the sampling mesh,
- the conic points belong to the vertices of the sampling mesh.

The displacements of the vertices of the sampling mesh are local and bounded w.r.t. the parametrization. More precisely, a covering of the parameter space in "deformation zones" associated to the vertices of the sampling mesh is done. These zones are rectangular neighborhoods constructed from a parallel mesh passing by medium values (in Figure 4, the deformation zones are the rectangles filled in grey surrounding the selected vertices).
This parallel mesh is used to localize the vertices to be moved and adjust them on the sharp edges and the conic points as follows:

- for each sharp edge $\mathcal{E}$, a sweep is done (similar to the sweep previously used to sample a face):

  $\mathcal{E}$ is cut with every medium intermediate cross section and the parameters of the intersection points are used to determine the vertices to be moved. One can notice here that the number of selected vertices is not fixed: in Figure 4, two vertices have been selected on section D (between intersections of $\mathcal{E}$ with sections C' and D') whereas only one has been selected on section B (between intersections of $\mathcal{E}$ with sections A' and B').

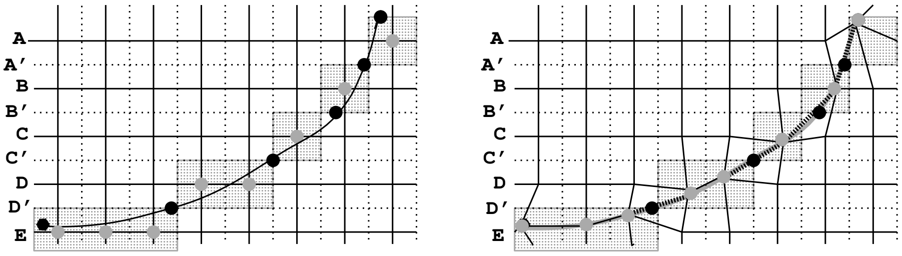

Figure 4: The selected vertices are marked in grey and the intersection points in black. The black curve is the projection of $\mathcal{E}$ in the parameter space. On the left, the vertices have been selected and, on the right, the sampling mesh have been deformed w.r.t. $\mathcal{E}$.

- When the set of vertices to be moved have been selected, each vertex V is displaced on a point P belonging to $\mathcal{E}$ as follows:
  - 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.
  - 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 centered at the extremity (see Figure 5). We take P=Q when the parameters of Q are inside the deformation zone associated to 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 to V.

- The cases of the conic points and of the extremities of the sharp edges are similar: the points are located w.r.t. the mesh of the deformation zones and the vertices containing these points in their deformation zone are moved on them.

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 subpath of the sampling mesh. In this case, a diagonal edge must be added each time there is a "hole" in the sequence of

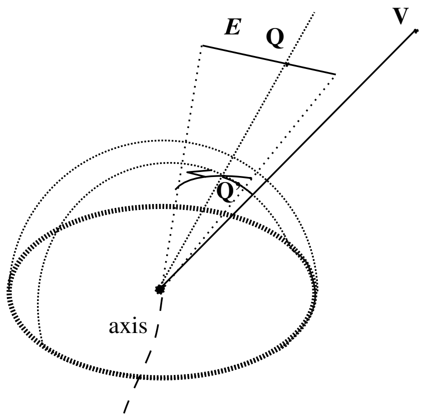

Figure 5: Computation of Q in the hemispherical case.

vertices displaced on $\mathcal{E}$. For example, in Figure 4, 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 has no sense, we move the vertex even though it has been displaced before. Since the position of the conic points are of prime importance, we choose to examine first the sharp edges (w.r.t. their order of appearance in the list of sharp edges) and then the conic points.
The order of the edges inside the list of sharp edges, which determines the relative importance of the salient features, 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.

4 Interpolation

We must ensure, before the interpolation process, that the two sampled meshes of the objects have the same topology. This was true before we adjusted these meshes on the sharp edges. But in the previous section we saw that diagonals could be added to the meshes, thus we have to report in each mesh the same diagonals. This is done by splitting the rectangular faces where diagonal edges have been added in two or four triangular faces:

- if only one diagonal appears in one of the two sampling meshes, then it cuts the rectangular face in two,
- 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.

As we have already stated, the interpolation is performed at two levels: we interpolate simultaneously the coordinates systems, which are defined by the axes, and the local coordinates of each object in these coordinates systems.

4.1 Axes interpolation

We consider the discretized curve 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. Simply it is necessary to find a good interpolation process between the two polygonal lines. Henceforth, the terms axis, curve or polygonal line will designate the same entity.

As noticed in [SGWM93], a linear interpolation of the two polygonal lines seems to be inadequate. For instance, the interpolation of two parallel segments oriented in opposite directions collapses for some interpolation value; this is not acceptable.

Actually we consider the axis as a moving frame, which means that a 3D frame is associated to each point of the axis and we interpolate the whole structure (see Figure 6).

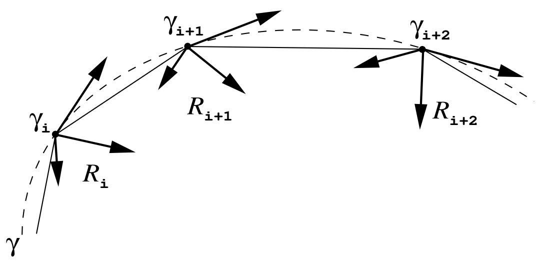

Figure 6: A discretized axis with its frames

A moving frame associated to a curve γ is a couple $(\gamma, \mathcal{R})$, where $\mathcal{R}$ is an orthonormal moving frame (e_1, e_2, e_3) such that $e_1(t)$ has the same direction as the tangent $\gamma'(t)$ at the point $\gamma(t)$. The basic idea is to interpolate the evolution of these frames rather than the frames themselves. Thus we interpolate the relative coordinates and the relative rotation between two consecutive points and frames of the polygonal line. Let's note $(\gamma_1, ..., \gamma_n)$ (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 coordinates of γ_{i+1} in $(\gamma_i, \mathcal{R}_i)$, and r_{i+1} is the rotation between $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$. By interpolating the X_i and the r_i of the two axes, we can construct for any intermediate time t a polygonal line: starting from a point on the curve (for instance an extremity) we recursively compute the next point 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 quaternion interpolation for the rotation r_i (more exactly for the corresponding quaternion q_i). We follow the notations of [Sho87] 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.

It is clear from the description of the whole process that the interpolation is continuous. Let us point out here that continuity does not only mean that a slight change in the polygonal lines would result in a slight change in their interpolation, but also that the process applied to digitized curves, whose steps of discretization go towards zero, would result in continuous interpolating curves.

Moreover, equation (1) shows that X_i has a second and a third components close to zero since it is almost parallel to the tangent curve. One can note from equation (3) that the interpolation is linear at a local level which holds true for the complete length of the curve. For an exact linear interpolation of the length, we could have interpolated separately the length and the orientation of X_i. However we found this of no practical use.

This curve interpolation leads to good visual results as one can see in Figure 11. Moreover the computation is simple and fast: the whole interpolation process is fully interactive on a Silicon Graphics workstation.

4.2 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.

Thus 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 designed directly !

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.

5 A sample of animations

Let us comment our method using a sample of animations.

- Figures 7, 8 and 9 illustrate the role played by the axes to control the shape transformation. The original objects are a cube and a cone. Two axes are defined: a vertical line segment and an oblique curve. These axes are discretized using seven frames and, for each planar section, sixteen angular parameters are used to sample the objects. They are drawned in bold for each sampled model associated to an initial or a final object.

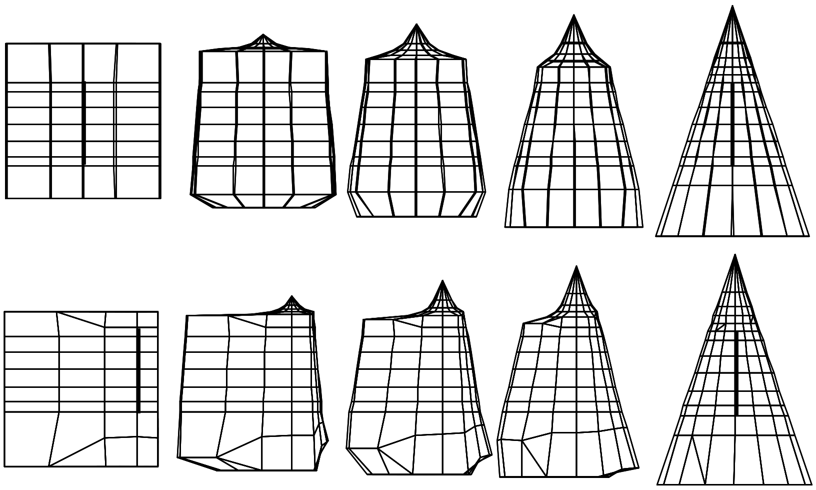

Figure 7: The same axis is used to sample the two original objects.

  In Figure 7, 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 way to control the shape transformation is similar to what Kent, Parent and Carlson [KPC91] showed by moving the center of projection inside one of the star-shaped objects.

  When taking two different axes (here an oblique curve and the vertical line segment used in Figure 7) to sample the initial and the final objects, the axis interpolation notably influences the shape transformation. In Figures 8 and 9, the two shape transformations differ: on the top, the axis rotate in clockwise order from the left to the right whereas on the bottom, the rotation is made in counterclockwise order. These rotations lead to different movements of the salient features and thus to different intermediate shapes, as one can see on Figure 8.

- Three objects have been sampled and three transformations have been constructed in Figure 10. Here, the shape transformations are showned in vertical columns.

  The parametrization associated to the fish is nearly cylindrical: its associated axis begins and ends near the mouth and the tail. Then, when transformed in a cube, the mouth and the tail form the bases, as it is visible on the intermediate shape of the left column.

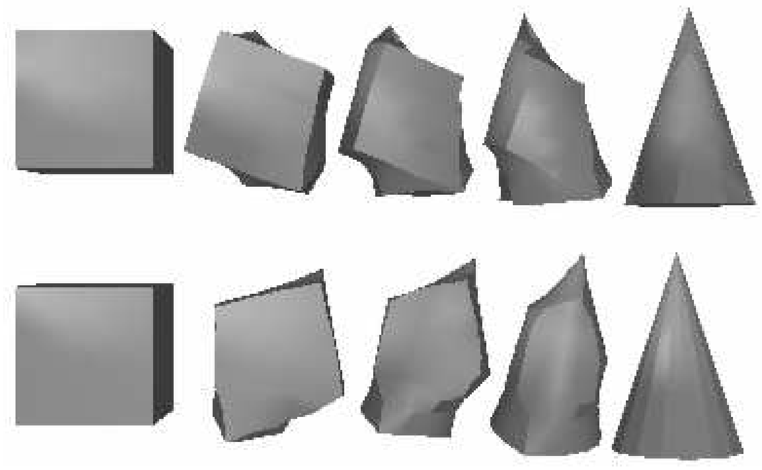

Figure 8: The influence of the axis interpolation in the shape transformations.

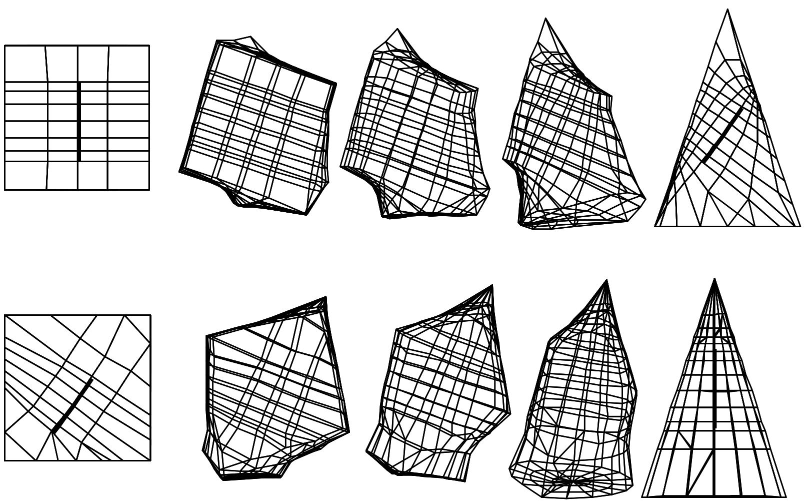

Figure 9: The same shape transformations in wire-frame.

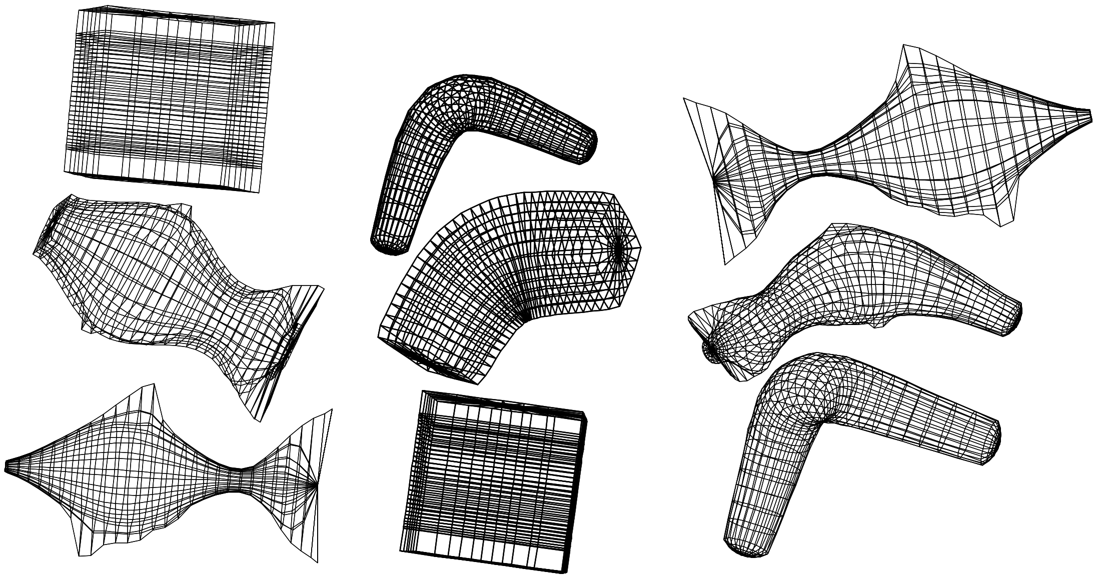

Figure 10: in the horizontal middle row, three intermediate shapes interpolated from the top and bottom original shapes

- 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.
- In Figure 12, the fish has been deformed into a sickle. The axis of the fish is a straight line whereas the axis associated to the sickle is bended 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.

6 Conclusion

We have proposed a method for computing a continuous deformation between two polyhedral shapes. The proposed method achieved:

- A natural way to solve jointly the correspondence and the interpolation problem.
- Interactive specification and control of the shape transformation. The use of axes to specify the transformation is intuitive: it 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 Carslon [KPC91].
- Good visual effects during the transformation, as shown on the examples of the previous section.
- An efficient way to compute transformation between shapes. The use of an intermediate polyhedral mesh reduces the computational time: for each object, it depends on the number of vertices and faces of the object and on the fineness of the discretization. The user may obtain approximate sketches of the deformation process using a rough discretization during the sampling.
  Moreover, our approach can be well adapted to compute transformations between 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.

- A general shape transformation adapted to any objects that are star-shaped around an axis. In particular, our method can be simply extended to implicit surfaces [WBB+90]. 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.

This algorithm has been integrated in ACTION3D, a general interactive modeling system developed jointly by SOGITEC and INRIA. The whole process (sampling and interpolation) is interactive on a Silicon Graphics workstation.

Work in progress include attempts to generalize the method to other families of objects, i.e., to generalize the sampling and parametrization to another kind of skeleton.

References

[Bar84] A. H. Barr. Global and local deformations of solid primitives. *Computer Graphics (SIGGRAPH '84 Proceedings)*, 18:21–30, July 1984.

[BD93] 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.

[Bis75] R.L. Bishop. There is more than a way to frame a curve. *American Mathematical Monthly*, (82):246–251, 1975.

[BN92] T. Beier and S. Neely. Feature-based image metamorphosis. *Computer Graphics (SIGGRAPH'92)*, 26(2):35–42, July 1992.

[BU89] 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*, pages 215–224, Computer Graphics International, 1989. Springer-Verlag.

[CJ91] S. Coquillart and P. Jancène. Animated free-form deformation: An interactive animation technique. *Computer Graphics (SIGGRAPH '91)*, 25:23–26, July 1991.

[CP89] S.E. Chen and R.E. Parent. Shape averaging and its applications to industrial design. *IEEE Computer Graphics and Applications*, 9(11):47–54, January 1989.

[Hug92] J.F. Hughes. Scheduled Fourier volume morphing. *Computer Graphics (SIGGRAPH'92)*, 26(2):43–46, July 1992.

[Klo86] F. Klok. Two moving coordinate frames for sweeping along a 3d trajectory. *Computer Aided Geometric Design*, (3):217–229, 1986.

[KPC91] J.R. Kent, R.E. Parent, and W.E. Carlson. Establishing correspondences by topological merging: a new approach to 3-D shape transformation. In *Graphic Interface'91*, pages 271–278, Calgary, June 1991.

[KPC92] J.R. Kent, R.E. Parent, and W.E. Carlson. Shape transformation for polyhedral objects. *Computer Graphics (SIGGRAPH'92)*, 26(2):47–54, July 1992.

[KR91] A. Kaul and J. Rossignac. Solid-interpolating deformations: construction and animation of PIPS. In *Eurographics'91*, pages 493–505, Vienna, September 1991.

[KTZ92] B.B. Kimia, A. Tannenbaum, and S.W. Zucker. On the evolution of curves via a function of curvature. I. The classical case. *Journal of Mathematical Analysis and Applications*, 163:438–458, 1992.

[LCJ93] F. Lazarus, S. Coquillart, and P. Jancène. Interactive axial deformations. In *Modeling in Computer Graphics*, pages 241–254, Genova, June 1993. Springer-Verlag.

[Par92] 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.

[PEF+90] A. Pentland, I. Essa, M. Friedmann, B. Horowitz, and S. Sclaroff. The thingworld modeling system: Virtual sculpting by modal forces. *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, 24(2):143–144, March 1990.

[SG92] T.W. Sederberg and E. Greenwood. A physically based approach to 2-D shape blending. *Computer Graphics (SIGGRAPH'92)*, 26(2):25–34, July 1992.

[SGWM93] 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, August 1993.

[Sho87] K. Shoemake. Quaternion calculus and fast animation, computer animation: 3-D motion specification and control. In *SIGGRAPH 1987 Tutorial*, pages 101–121, 1987.

[TPBF87] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics (SIGGRAPH '87)*, 21:205–214, July 1987.

[WBB+90] B. Wyvill, J. Bloomenthal, T. Beier, J. Blinn, A. Rockwood, and G. Wyvill. Modeling and animating with implicit surfaces. In *Siggraph Course Notes*, volume 23, 1990.

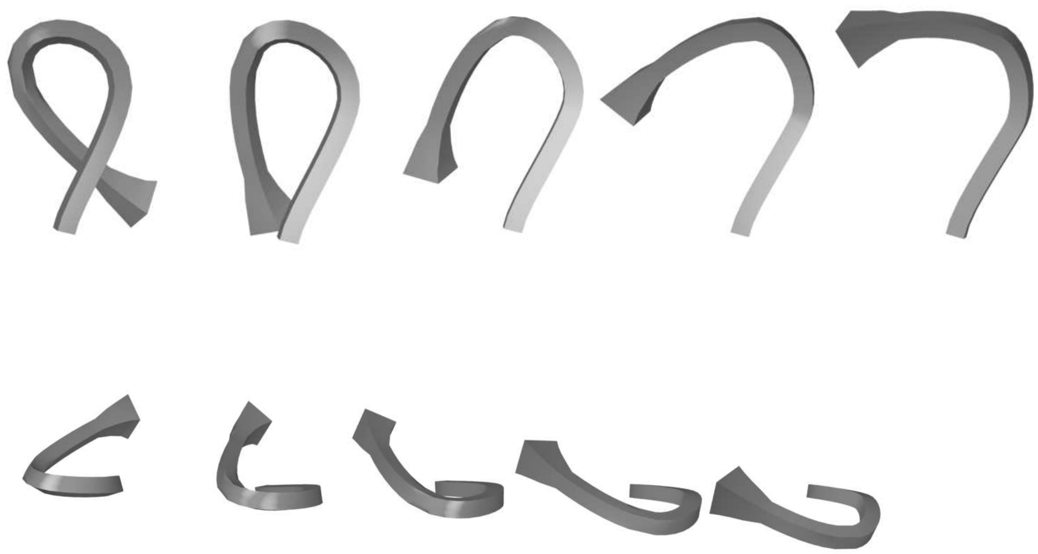

Figure 11: Axis interpolation: front-view (up) and top view (bottom).

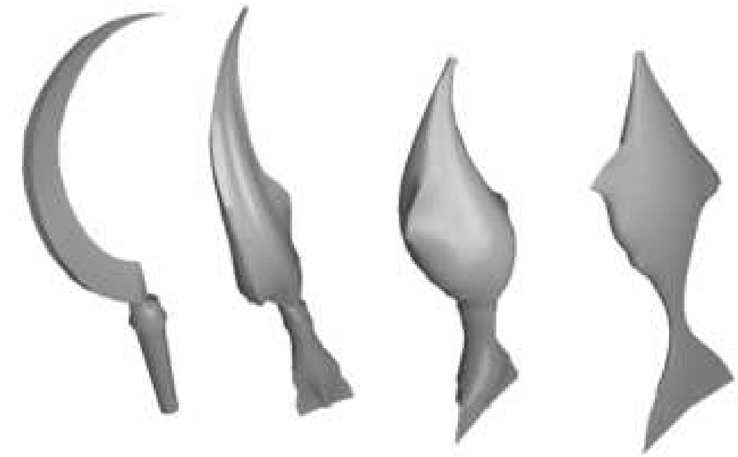

Figure 12: Shape transformation between a sickle and a fish.