

Metamorphosis of Non-homeomorphic Objects

Mehdi Elkouhen and Dominique Bechmann

Université Louis Pasteur – LSI IT, Pôle API –
Boulevard Sébastien Brant, 67400 Strasbourg France
elkouhen, bechmann@dpi-info.u-strasbg.fr

Abstract. In this article, we present our axial four dimensional deformation tool. This tool is defined in the context of geometrical modeling of animations, where animations are represented by four dimension polyhedrons, and it permits to control the shape/topology of animations.

We illustrate this tool by deforming animations representing a motionless object into the merging of two similar objects. And we show how controlling the shape of the tool enables to control the path the objects follow during the merging but also the smoothness of the merging.

1 Introduction

A metamorphosis is the process of transforming continuously a source object into a target object, while keeping the main features of both objects. The solutions are generally classified in volume-based and boundary-representation based approaches [17]. In a simplified way the volume-based approach of objects metamorphosis can be stated as a "blend" of objects. In the boundary-based approach, the process is harder: it requires to construct a supermesh that collapses on the source and target object. So in this approach, the metamorphosis is the continuous deformation of the supermesh from the first collapsed shape to the second.

The algorithms of construction of object metamorphosis are various. They are guided by the user who can set correspondence points [18], skeletons [28] or features [9]. The algorithms take in account the geometry of the objects: they deal differently for star-shaped objects [13, 14], cylinder like objects [16] or objects homeomorphic to a disk [12]. The shape of the inbetween objects can be controlled by influence shapes [19, 21]. Except in [7, 23], the boundary-representation based techniques deal only with homeomorphic objects or at least homeomorphic to a sphere while the volume-based approach techniques deal with arbitrary objects. But none of these techniques permit to control precisely the topology of the inbetween objects. For example, how can a designer make new features (e.g. holes) appear only on the inbetween objects? In techniques that use influence shapes, one can use a toroidal influence shape but this strongly modifies the shape of the inbetween objects by giving them the shape of a torus. Geometrical modeling of animation (GMA) permits to overcome this. The main

reason is that the animations are modeled as four dimensional polyhedrons and many operations of control of shape/topology can be adapted to the fourth dimension.

In GMA, animations are visualized as polyhedrons [11]. The movies are extracted by displaying cuts of the object by a set of planes. Different algorithms of construction of animations have been studied. In [4], Brandel constructs the animations by extruding surfaces following an arbitrary polyline. When the extrusion is linear and the set of cut planes are chosen normal to the line, the animations represent the initial surface translating along the extrusion line. In [5], the animations are constructed by thickening 3D graphs so that the animations represent a sequence of merging and scission of spheres. In [20], Skapin discusses of the creation of animations by computing the Cartesian product of two objects. This technique involves that users must have a strong intuition on the way to choose the objects. As the topology of a cut depends not only on the topology of the 4D polyhedron but also on its shape, deformation tools have been defined. Aubert and Brandel [1, 4] defined animations with topological changes using Dogme [3] the N-dimensional deformation tool . To increase the control on the shape of the animations, we defined a set of tools [2] exhibiting different kinds of control.

The main goal of our work is the creation of metamorphosis of non homeomorphic objects; we deal with this problem by deforming 4D polyhedrons. In this article, we explain how to deform an animation defined by a motionless object in the merging of two similar surfaces. For this, we use our axial deformation tool, and we show how a simple control on the shape of the tool permits to control the path of the merging and its smoothness. We illustrate this technique by explaining the different steps of construction of the merging of two tori. In section 2, we describe our axial deformation tool. In section 3, we give the different steps of construction of the animation. In section 4, we explain how to create sharp mergings.

2 Deformation Tool

This tool is an adaptation, of Chang and Rockwoods [6] axial deformation tool, to a four dimensional tool. It is defined by a control polygon P_i and a set of handles. The polygon defines a Bézier curve, which is the axis of deformation. The handles are introduced for a four dimensional control of the deformation. As shown in Fig. 1(a), there are three handles S_i , T_i and L_i (represented by yellow, red and blue vectors) associated to each control point P_i except the last one.

In section 2.1, we describe the techniques to create and manipulate such a tool and in Sect. 2.2 we give the algorithm of deformation.

2.1 Creation and Manipulation

A designer can create a tool in two ways. The first one consists in specifying a linear control polygon and three directions A, B and C : the tool created is linear (Fig. 1(a)) and all the handles S (respectively T and L) share the same direction A (resp. B and C). The user can also create a tool by simply inputting a Bézier curve (Fig. 1(b)); the handles are computed automatically by the algorithm described in Sect. 3.4.

When manipulating an axial deformation tool, one manipulates three kinds of parameters : Position of the control points, Length of the handles, Orientation of the handles. To show the link between the modification of each of these parameters and the shape of the deformations, three examples are presented. In order to stay as clear as possible, the objects considered are cubes (embedded in the xyz space) and only the handles S and T are displayed (as the time component of the vertices of the object are null, the handles L are not taken into account by the deformation algorithm 2.2).

The Position of the control points : In Figure 2(d), the designer moved the control points defining the initial linear axis in order to define a bent axis : the cube is thus warped along the bent axis (Fig. 2(a)).

The Length of the handles : In Figure 2(e), the designer created a linear tool and then pulled the handles : the cube is tapered (Fig. 2(b)).

The Orientation of the handles : In Figure 2(f), the designer created a linear tool and then turned the handles around the axis of deformation : the cube is twisted (Fig. 2(c)).

2.2 Algorithm

The algorithm is a simple extension of the one given by Chang and Rockwood[6], obtained by taking into account the new handle L and the time component of the input vertices. We have modified the original algorithm in order to make the axis of deformation represent the evolution of time instead of representing the x axis and this by interpolating in the time direction instead of interpolating in the x direction.

The algorithm takes in input a point M , of coordinates (x, y, z, t) , and interpolates the result of affine transformations applied to the point; the result $P_0^n(M)$ is the deformed point.

$$\begin{cases} P_i^0(M) = P_i & 0 \leq i \leq n \\ P_i^1(M) = \Theta[P_{i-1}^0(M), P_i^0(M), S_i, T_i, L_i](M) & 0 \leq i \leq n-1 \\ P_i^j(M) = \Theta[P_{i-1}^{j-1}(M), P_i^{j-1}(M), \mathbf{0}, \mathbf{0}, \mathbf{0}](M) & \begin{cases} 1 \leq j \leq n \\ 0 \leq i \leq n-j \end{cases} \end{cases}$$

With Θ defined as following :

$$\Theta[O, T, \mathbf{i}, \mathbf{j}, \mathbf{k}] \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix} = O + tOT + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

In this section we adapted Chang and Rockwoods mathematical formalism. We will now show how to instantiate this tool and to define merging of polyhedrons in a smooth way.

3 Construction of the Smooth Merging

Intuitively, the steps to construct the merging of two similar shapes consists in extruding an initial shape, bending the resulting object and then extracting the animation. The bending step enables "symmetric" sections of the initial object to be defined in the same plane.

For example, if we wish to merge two circles, we extrude a circle (the result is a cylinder); and then we bend the cylinder 3(a). Figures 3(b)–3(d), represents a set of cuts of the cylinder.

In the following, we show how to merge two tori. The reason we define the merging of two similar objects comes from the way we construct our four-dimensional polyhedrons by extruding 3D shapes : the extrusion operator naturally associates to each vertex of one object a vertex of the other object. The techniques we present in next section are not dependant on the way we construct our objects, so they could be applied on more general objects [5, 20, 21].

3.1 Construction of the Four Dimensional Polyhedron

We construct our four dimensional polyhedron by extruding a torus, embedded in the xyz space, following the time direction. The points having their time component null are called "bottom" points, and symmetrically the points having their time component equal to 1 are called "top" points.

3.2 Sketching the Curve

The designer sketches a Bézier curve, which represents the path the objects follow during the merging. For the current animation, the designer sketched a curve having the shape of an "S". This curve (Fig. 4(a)) defines the axis of our deformation tool. In the two next subsections, we will explain how to compute the final parameters of the tool.

3.3 Send the Inner Control Points to the Future

The two endpoints of the tool represent the positions of the two objects at the beginning of the animation (time=0). The inner part of the curve represents the motion path of the two objects, so we have to modify the position of the control points by sending them to the future.

The strategy we followed in this example, for a curve defined by $n+1$ control points P_i , consists in sending the points of index $i < n/2$ to the time i and sending the points of index $i > n/2$ to the time $n-i$.

The strategy can be chosen freely; each strategy generates a different kind of merging. It is by choosing another strategy that we show, in Sect. 4, how to control the smoothness of the merging.

3.4 Compute the Handles

The most important stage in the definition of the animation is to compute the handles correctly. Otherwise the 4D polyhedron self-intersects or is flattened; in these cases the animations are not appealing. The solution consists in computing the handles using the rotation minimizing frame [15]. This algorithm ensures the change of direction between two successive handles of same kind is minimal. In this way, we eliminate sudden changes of direction of handles and so we avoid self intersecting deformations. In this subsection, we present a four dimensional version of the way we compute our handles.

The first control point and the last one represent the beginning of the animation (time=0). In order to make the "top" and "bottom" points of the constructed object share the same time, the time component of the handles, associated to the point P_0 and P_{n-1} , have to be null. This is a consequence of the fact that the result of the deformation of a point of coordinates $(x, y, z, 0)$ is $P_0 + xS_0 + yT_0 + zT_0$ and the result for a point of coordinates $(x, y, z, 1)$ is $P_n + xS_{n-1} + yT_{n-1} + zT_{n-1}$.

The first three handles S_0 , T_0 and L_0 are set by the user in order to form an orthonormal set of vectors and that the time component of these vectors is null. In this way, the four vectors P_0P_1 , S_0 , T_0 and L_0 form a orthogonal basis of \mathbb{R}^4 .

The other handles are computed in an iterative way. The handle H_{i+1} is computed by turning the handle H_i . It is the rotation defined in the plane spanned by the two vectors $P_{i-1}P_i$ and P_iP_{i+1} ; the angle of rotation is the angle between the vectors $P_{i-1}P_i$ and P_iP_{i+1} . For a precise presentation of four dimensional rotations, one can refer to Hanson's article on N-dimensional rotations [10].

As we compute iteratively the new set of vectors P_iP_{i+1} , S_i , T_i and L_i by turning the previous set, these sets always form an orthonormal basis of \mathbb{R}^4 .

As the three vectors S_{n-1} , T_{n-1} and L_{n-1} are normal to the time vector $P_{n-1}P_n$, they are space vectors. This is the condition we needed, so the algorithm is valid.

3.5 Apply the Deformation and Extract the Animation

Figures 4(b)–4(e) represent different cuts of this object, by a set of hyperplanes normal to time.

An advantage of controlling the shape of the metamorphosis by a deformation tool, is that we can control the continuity of the mergings with visual parameters. This will be the subject of next section where we will discuss of another strategy for the step 3.3, and of the continuity between axial tools.

4 Controlling the Smoothness of the Merging

The animations extracted from the curves constructed using the strategy described in 3.3, represent two points that merge softly. In order to make the points merge sharply, one can, for example, use another strategy that flattens the curve and extract the animation using a set of planes parallel to the flat area.

For example, we defined a tool defined by two continuously connected Bézier curves. In order to flatten the axis around the connection point, we set the control points, neighboring the connection point, to be on a same line.

We deform our extruded torus, using the two tools defined by the two previous curves. Figures 4(f)–4(i) represent different steps of the sharp merging of the tori.

As the definition of the continuity between two axial 3D De Casteljau Generalized deformation tools was discussed in [6], we only precise, in next paragraph, the way we deal with two continuously linked tools and the computing of the handles over the global curve.

Let the first tool be defined by $n + 1$ control points P_i^0 and the handles S_i^0 , T_i^0 and L_i^0 , and the second tool be defined by $m + 1$ control points P_i^1 and the handles S_i^1 , T_i^1 and L_i^1 . The points P_n^0 and P_0^1 are merged in order to make the three control points P_{n-1}^0 , P_n^0 and P_0^1 collinear. The user specifies the handles for the control point P_0^0 ; the rotation minimizing frame algorithm computes the other handles of the tool. For continuity reasons, the first handles associated to the second tool have to be set to be equal to their "equivalent" handles of index $n - 1$ of the first tool. Finally, the other handles of the tool are computed automatically.

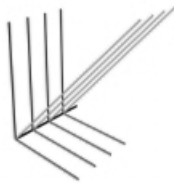
5 Conclusion

In this article, we presented our four dimensional extension of Chang and Rockwood's axial deformation tool and discussed the way it can be used to control sharp/smooth merging of shapes. This work will be continued by a study of the merging of N different objects and the control of the areas of contact during the mergings. The first point should require to create an algorithm, of construction of 4D polyhedrons, that brings together the strength of the Cartesian product [20] and the interpolation of objects as proposed by Turk and O'Brien[21]. The second point should require to use deformation tools of higher topological dimension such as tools defined by surfaces [2] etc.

References

- [1] F. Aubert and D. Bechmann. Animation by deformation of space-time objects. *Eurographics'97*, 16(3):57–66, 1997.
- [2] D. Bechmann and M. Elkouhen. Animating with the “multidimensional deformation tool”. *Eurographics Workshop on Computer Animation and Simulation'01*, pages 29–35, 2001.
- [3] P. Borrel and D. Bechmann. Deformation of n -dimensional objects. *International Journal of Computational Geometry & Applications*, 1(4):427–453, 1991.
- [4] S. Brandel, D. Bechmann, and Y. Bertrand. STIGMA: A 4-dimensional modeller for animation. *Eurographics Workshop on Computer animation and simulation 1998*, 1998.
- [5] S. Brandel, D. Bechmann, and Y. Bertrand. Thickening: an operation for animation. *The Journal of Visualization and Computer Animation*, 11(5):261–277, 2000.
- [6] Y. Chang and A. P. Rockwood. A generalized de casteljau approach to 3d free-form deformation. *Proceedings of SIGGRAPH 94*, pages 257–260, 1994.
- [7] D. DeCarlo and J. Gallier. Topological evolution of surfaces. In *Graphics Interface '96*, pages 194–203, 1996.
- [8] E. Galin and S. Akkouche. Blob metamorphosis based on minkowski sums. *Eurographics'96*, 15(3):C143–C154, 1996.
- [9] A. Greogory, A. State, M. Lin, D. Manocha, and M. Livingston. Feature-based surface decomposition for correspondence and morphing between polyhedra. *Proceedings of computer animation'98*, pages 64–71, 1998.
- [10] A J. Hanson. Rotations for n -dimensional graphics. *Graphics Gems V*, pages 55–64, 1995.
- [11] S. R. Hollasch. Four-space visualization of 4d objects. Master's thesis, Arizona state university, 1991.
- [12] T. Kanai, H. Suzuki, and F. Kimura. 3d geometric metamorphosis based on harmonic maps. *Proceedings of Pacific Graphics'97*, pages 97–104, 1997.
- [13] James Kent, Richard Parent, and Wayne E. Carlson. Establishing correspondences by topological merging: A new approach to 3-D shape transformation. *Proceedings of Graphics Interface '91*, pages 271–278, 1991.
- [14] James R. Kent, Wayne E. Carlson, and Richard E. Parent. Shape transformation for polyhedral objects. *Computer Graphics*, 26(2):47–54, 1992.
- [15] F. Klok. Two moving coordinate frames for sweeping along a 3d trajectory. *Computer Aided Geometry Design*, 3:217–229, 1986.

- [16] F. Lazarus and A. Verroust. Metamorphosis of cylinder-like objects. *The Journal of Visualization and Computer Animation*, 8(3):131–146, 1997.
- [17] F. Lazarus and A. Verroust. Three-dimensional metamorphosis: A survey. *The Visual Computer*, 14:373–389, 1998.
- [18] R. E. Parent. Shape transformation by boundary representation interpolation: a recursive approach to establishing face correspondences. *The Journal of Visualization and Computer Animation*, 3(4):219–239, 1992.
- [19] J. Rossignac and A. Kaul. AGRELs and BIPs: Metamorphosis as a bezier curve in the space of polyhedra. *Eurographics'94*, pages C179–C184, 1995.
- [20] X. Skapin and P. Lienhardt. Using cartesian product for animation. *Eurographics Workshop on Computer Animation and Simulation 2000*, pages 187–201, August 2000.
- [21] G. Turk and J. O'Brien. Shape transformation using variational implicit functions. *Proceedings of SIGGRAPH 99*, pages 335–342, 1999.
- [22] B. Wyvill, J. Bloomenthal, T. Beier, J. Blinn, A. Rockwood, and G. Wyvill. Modeling and animating with implicit surfaces. *Siggraph course notes*, 23, 1990.
- [23] M. Zöckler, D. Stalling, and H-C. Hege. Fast and intuitive generation of geometric shape transitions. *The Visual Computer*, 16(5):241–253, 2000.

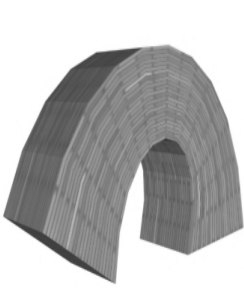


(a) Linear tool

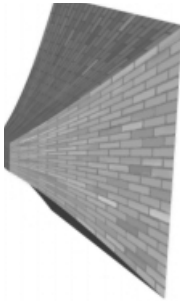


(b) Bent tool

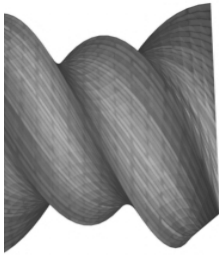
Fig. 1. Axial tools



(a) Bent cube



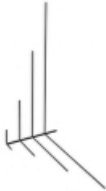
(b) Tapered cube



(c) Twisted cube



(d) Bending tool



(e) Tapering tool



(f) Twisting tool

Fig. 2. 3D examples



(a) Bent cylinder



(b) time=0



(c) time=1

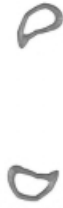


(d) time=2

Fig. 3. Merging of 2 circles



(a) Sketched polyline



(b) soft merging – 2 deformed tori



(c) beginning of the merging



(d) end of the merging



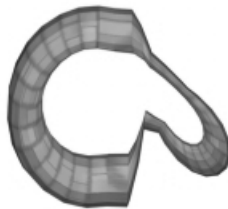
(e) torus warped by an S shaped tool



(f) sharp merging – time=0



(g) time=1



(h) time=2



(i) time=3

Fig. 4. Merging of 2 tori