3D Metamorphosis: a Survey

Francis Lazarus[†] and Anne Verroust[‡]

† I.R.C.O.M.-S.I.C. SP2MI, Boulevard 3, Teleport 2

B.P. 179, 86960 Futuroscope Cedex, FRANCE

‡ INRIA Rocquencourt, Domaine de Voluceau

B.P. 105, 78153 Le Chesnay Cedex, FRANCE

Abstract

A metamorphosis or a (3D) morphing is the process of continuously transforming one object into another. 2D and 3D morphing are popular in computer animation, industrial design or growth simulation. Since there is no intrinsic solution to the morphing problem, user interaction can be a key component of a morphing software. Many morphing techniques have been proposed in the recent years for 2D or 3D objects. We present a survey of the different approaches in 3D giving a special attention to the user interface. We show how the approaches are intimately related to the object representations. We conclude by sketching some morphing strategies for the future.

Key Words: metamorphosis, shape transformation, interpolation, computer animation, geometric modeling.

1 Introduction

Shape interpolation is the process of transforming one shape into another. A metamorphosis or a (3D) morphing of 3D graphical objects [GCDV96] includes the interpolation of their shapes as well as an interpolation of their attributes. Objects attributes include color, textures or normal fields. Note that an object with attributes can be considered as a geometric object in a higher dimensional space. For instance a grey level image can be considered as a terrain surface if the color is interpreted as a height field. In that way shape interpolation can be regarded as essential to the morphing problem.

In recent years many algorithms were proposed to compute a transformation between two shape models. Among other applications morphing is a popular technique used in computer animation systems and industrial design. It gives the animator the ability to "fill" an animation between key-framed objects by in-betweening. It allows the designer to blend existing shapes in order to create new shapes.

Given two shapes there are an infinity of transformations that takes the first shape into the other. Although it seems impossible to define an intrinsic morphing sequence between any two shapes there clearly exist intuitive solutions. By just looking at a particular sequence we can tell whether it is pleasant or not. The morph should be smooth and it should keep as much as possible of the two shapes during the transformation. Of course these are purely subjective aesthetic criteria and they depend on the context in which the transformation is performed. For this reason the ability of the user to control the metamorphosis is an important feature in a morphing tool. This control should be intuitive, not too heavy and adapted to the user's knowledge. Providing such a control is a non trivial task and some approaches were proposed that do not take this aspect into account. On the contrary, their authors may insist on automating the morph generation. For specific classes of objects or applications this can also be an effective approach.

Because the problem is difficult and because the quality of a morph is subjective there exist many different approaches dealing with different types of objects and based on different techniques.

To our knowledge no survey paper was published on 3D metamorphosis. The scope of this paper is to discuss, classify and compare the different approaches. Our classification helps in understanding the efficiency and the limits of the different approaches.

This paper is structured as follows:

Section 2 presents a classification of the morphing techniques based on the object models they apply to.

Section 3 surveys the existing techniques according to this classification.

In section 4 we attempt to analyze what could be improved in the existing approaches on the user point of view.

Then, in section 5, we give a brief discussion on the notion of shape followed by some new ideas for morphing strategies.

2 Influence of the object representation

There are several ways we can represent an object in a computer. The object representations and their corresponding data structures usually have a strong impact on the type (and difficulty) of algorithms involved in order to transform an object into another.

At a very coarse level one can consider three kinds of shape representation:

1. Objects can be described as level sets of functions defined on the whole 3D space. Implicit surfaces [BBB+97] and voxelized objects fall into this category. The voxelized objects may indeed be considered as a level set of its characteristic function with value one at the object voxels and zero elsewhere. One can also interpret a voxelized object as the zero level set of the discrete distance to the object voxels as in [COLS98]. If an object is defined by the set of points p such that $f(p) = c$ for some function f and level c, it is easy to define the interior of this object by considering all the points p such that $f(p) \leq c$. For this reason we will qualify morphing techniques based on this category of objects as *volume based*.

2. Objects, such as terrains, can be represented as an elevation map over a planar domain.

3. Objects can be represented by their boundary as a 2 dimensional surface such as a polyhedral surface or a spline surface.

2.1 Volume based approaches

The first category of objects is often preferred for morphing applications [LGL95, Hug92, HWK94, COLS98]. There are (apparently) little restrictions on the form of the functions defining the level sets. It follows that any kind of continuous interpolation between the functions that define the source object and the target object will at least produce some "smooth" transformation. If the source object is expressed by the set of points p such that $f_0(p) = c_0$ and the target object is given by $f_1(p) = c_1$, we can define an interpolated object for each $t \in [0,1]$ by the following equation:

$$(1-t)f_0(p) + tf_1(p) \quad = \quad (1-t)c_0 + tc_1.$$

Pasko and Savchenko [PS95] precisely use this formulation for the interpolation. Of course, as recalled by Ranjan and Fournier [RF96], one expects some nice conservative properties during the morph sequence. These properties include no unnecessary distortions or change in topology such as the creation of many connected components. This is why it is often necessary to combine such straightforward interpolation with more complex continuous deformations (warp) of the whole space as in [COLS98]. These warps aim at deforming the source space such that the warped source object matches the target object. Generally it is difficult to obtain a perfect match (after all the warp would in itself solve the morphing problem !). If W_t is the warp, with W_0 equals to identity and W_1 approximately maps the source object to the target object, then we can define an interpolated object for each $t \in [0,1]$ by the following equation:

$$(1-t)f_0(W_t^{-1}(p)) + tf_1(W_1 \circ W_t^{-1}(p)) =$$
$$(1-t)c_0 + tc_1.$$

It is not always easy to compute at the same time W_t and its inverse. Cohen-Or et al. [COLS98] use an approximation B_t^0 of the backward mapping W_t^{-1} that goes from the deformed space $W_t(\mathbb{R}^3)$ back to the source space and an approximation B_t^1 of the backward mapping $W_1 \circ W_t^{-1}$ that goes from the target space back to the deformed space $W_t(\mathbb{R}^3)$. The interpolated object is then defined by:

$$(1-t)f_0(B_t^0(p)) + tf_1(B_t^1(p)) = (1-t)c_0 + tc_1.$$

Other techniques are based on a decomposition of the object functions over some functional space basis. Hughes considers a Fourier decomposition [Hug92] and He et al. decompose the functions with a wavelet transform [HWK94]. The coefficients of the decomposition are interpolated in order to define the coefficients of the interpolated object function. The interpolated object function is further recomposed using these coefficients.

Kaul and Rossignac [KR91, RK94] do not explicitly refer to the function that defines the objects but still use a weighted Minkowski sum with time changing coefficients to compute the metamorphosis.

More specific algorithms are used when dealing with implicit objects such as soft objects [WBB$^+$90, BBB$^+$97]. The functions defining the implicit objects are themselves defined by some parameters. The parameters are composed of points or more complex skeletons and of a couple scalar values that define the potential functions around the skeleton. In order to interpolate such functions it is more relevant to directly interpolate their parameters. Galin and Akkouche [GA96b] have used Minkowski sums in order to interpolate the skeletons. Note, however, that the interpolation of complex skeletons can lead to another difficult (skeleton) morphing problem.

2.2 Approaches based upon objects defined as elevation maps

The second category is more restrictive on the set of representable objects but has important application in terrain modeling and image morphing. A 2D image can be viewed as one (or three when using RGB coding) color map taken as an elevation map over the pixel grid. Again, a first solution to the interpolation problem is obtained by linearly interpolating the source and target elevation maps. In the case of image morphing more effective techniques were developed by Beier and Nelly [BN92] or Lee et al. [LCHS94, LCSW95]. These techniques include the use of a warp between the two map domains in order to match features such as ridges or peaks. One can note the resemblance between such techniques and some volume based techniques described above. Lerios et al. [LGL95] have developed a 3D morphing techniques which is in essence strictly equivalent to the image morphing proposed by Beier and Nelly [BN92]. This similarity is due to the fact that an object expressed as an elevation map $z = f(x, y)$ can be considered as the 0-level set of the function $F(x, y, z) = z - f(x, y)$. Conversely one can express a level-set object $f(x, y, z) = c$ as an elevation map $t = f(x, y, z)$ in 4D and “cut” this 4D object by the threshold $t = c$ before projecting back the result in 3D. With some minor modifications morphing methods that apply to the first category of objects also applies to the second category and vice-versa. In the following we will merge the two first categories of representation into a single one.

2.3 Approaches based on boundary representations

Boundary representations are very popular for representing 3D objects and 3D virtual worlds. A large number of models and data structures have been proposed to represent objects by their boundaries. The polygonal surfaces and the parameterized surfaces - such as spline surfaces - are the two main models used in the graphics community. The corresponding data structures usually contains topology and geometry informations. The topology tells the adjacency relationship between polygonal faces or parameterized patches while the geometry describes the precise coordinates of the vertices or control points that defines faces or patches. Note that the term topology may in general refer to two different concepts. It may refer to the specific subdivision (i.e. the vertex/edge/face network) of an object given by its polyhedral or patch model. It may also refer to the global topological space underlying the whole set of points of the object surface. (Remark that the notion of topological space does not depend on any particular subdivision while a subdivision, or mesh, entirely determines the characteristic of its underlying topological space.) Except where a confusion is possible we will use the same word for both meanings.

The use of boundary representations has several advantages. The corresponding data structure are relatively compact compare to the storage of voxelized objects. Many practical and efficient algorithms are available to visualize objects represented by their boundaries. It is also very easy to attach to boundary representations properties such as color, normal or texture. As a counterpart boundary representations are quite constrained and rigid. This mean that changing at random even a couple values in the data structure - one could modify the number of faces or the coordinates of a control points - may result in an invalid object: the modified data structure will not correspond to any representable object. As a consequence morphing becomes a more complex challenge. The presence of topology and geometry in a boundary representation generally leads to split the morphing problem into two steps: establishing a correspondence between the source and target object and interpolating the positions - or geometry - of the corresponding features. The correspondence step aims at constructing a single mesh with two geometric instantiations: one for each source and target object. This single mesh can be obtained by merging the two object meshes as in [BU89, KPC91, KCP92] or by creating a new common mesh as in [LV97]. The correspondence problem remains a difficult step and some authors [HMTT88, BU89, CP89, Par92, DG96, KHSK98, ST98] focus on the correspondence problem independently of the interpolation. According to the above remark the existence of a common mesh for the source and target object implies that they have the same (global) topology. This forbid the transformation of a sphere into a torus. To get round this constraint DeCarlo and Gallier [DG96] propose to use degenerate geometric instantiations of the common mesh where an edge or a face can be embedded onto a single point or edge. Once the correspondence is established, most of the papers use a linear or an Hermite spline interpolation between the corresponding vertices to compute the in-between shapes. A specific model is introduced in some papers: Delingette et al. [DWS93] use a physically-based algorithm to interpolate intrinsic geometric parameters, Sun et al. [SWC97] also interpolate intrinsic geometric parameters using a propagation paradigm, for restricted classes of shapes the interpolation problem is solved jointly with the correspondence problem in [KPC91, KCP92, DG94, LV97], and in [GSL+98] trajectories are interactively defined for a subset of vertices of the transforming mesh and further propagated to the entire mesh.

In the following section we review the existing work.

3 Description of the main approaches

In the following we review separately the volume based approaches and the boundary based approaches. We simply report the works in each of these two categories according to the chronology of their publication.

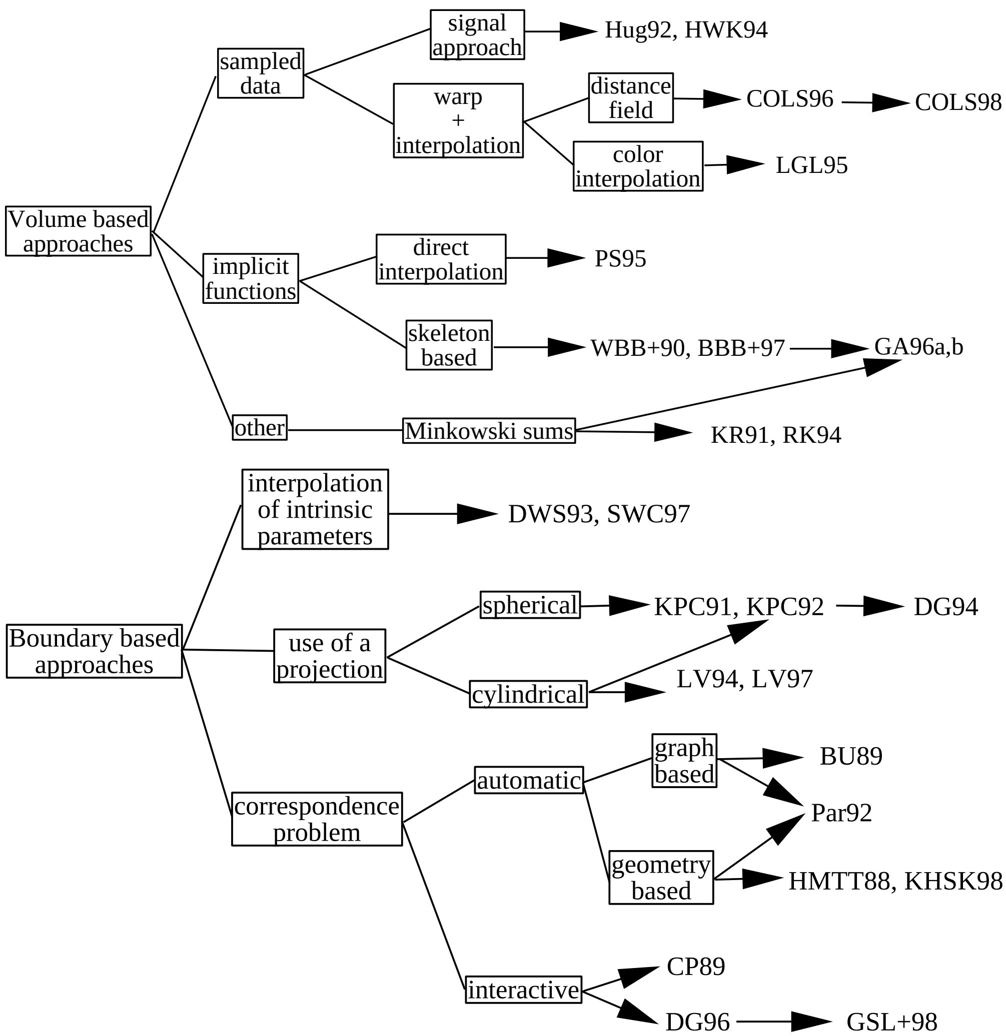

Figure 1: The boxes contain the main characteristics of the approaches. Arrows express affiliations to a set of characteristics or to previous approaches.

Fig. 1 is an attempt to organize the different morphing techniques according to their very basic characteristics. At the end of this section, we also summarize their features in table 1 and 2.

3.1 Volume based approaches

The approaches presented in this section are all based on a volumetric representation of the objects. Note that we do not address the image morphing problem (see [BN92, LCHS94, LCSW95, RF96, GS98, LWS98, TF98] on that subject and a survey of G. Wolberg [Wol98]).

The work of Wyvill [WBB$^+$90, BBB$^+$97]

Description: Wyvill presents a metamorphosis technique for soft objects whose skeletal elements are composed of points, lines, circles and polygons. Skeletons elements are first put into correspondence. The transformation is decomposed into a transformation between the corresponding skeletons and an interpolation of the associated field functions. Wyvill proposes heuristics to automate the correspondence between skeletons. He also defines simple interpolation rules for interpolating skeletal elements of different type.

Object representation: The method applies to soft objects having skeletal elements composed of points, lines, circles and polygons.

Interaction: The user has the ability to choose an automatic matching or to select interactively pairs of corresponding skeletons.

Comments: This method is very simple and intuitive as far as the skeletal element are not too numerous. See also below the related work of Galin and Akkouche [GA96b, GA96a].

The work of Kaul and Rossignac [KR91, RK94]

Description: Kaul and Rossignac provide an interpolation algorithm based on Minkowski sums (see Fig. 2.). If A and B are two sets of points, taken as vector endpoints, their Minkowski sum is the set of points of the form $a+b$ with a in A and b in B. The method is essentially volumetric and no correspondence is required between points of the source and target objects. However, it happens that the boundary of the Minkowski sum is included in the Minkowski sum of the boundaries. The proposed algorithm is thus based on a boundary representation. Kaul and Rossignac extend their method to a metamorphosis between a set of polyhedra using Bézier formulation and Minkowski sums.

Object representation: The technique applies to polyhedra represented by their adjacency graph.

Interaction: The user may only modify the relative orientation of the inputs polyhedra.

Comments: The method is efficient and give good results for convex polyhedra. For non convex shapes the transformation is much less intuitive and the computation may be inefficient. The user has no real control over the transformation. It takes time $\mathcal{O}(n^k)$ for computing a transformation between k polyhedra of size n with the Bézier formulation. Note that the complexity of the algorithm was improved by [GA96a].

The work of Hughes and He et al. [Hug92, HWK94]

Description: Hughes and He et al. use a signal approach. Hughes considers a Fourier transform of the functions defining the objects. He et al. use a wavelet transform. In both

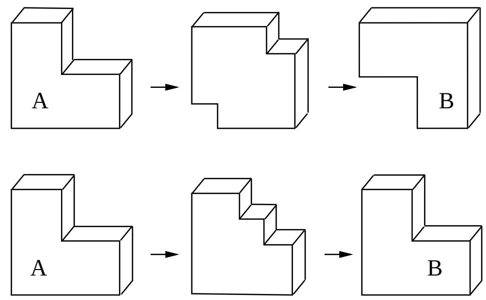

Figure 2: in-between shapes $(A \oplus B)/2$ using the Minkowski sum (Kaul and Rossignac 1991).

cases a discrete version of the transform is applied on voxelized data. The interpolation is performed in the transform domain. This allows to blend the different frequency (or space/frequency for wavelets) components of the signal with different schedules. The wavelet approach of He and al. establishes a correspondence at a low level resolution. This is not the case in [Hug92].

Object representation: The two approaches are devoted to sampled volumetric data.

Interaction: In both case the process is entirely automatic. according to the authors some interaction could be introduced in [HWK94] for the correspondence process.

Comments: Both methods are time consuming. For a volume of size n^3 the computation of every in-between image take at least $\mathcal{O}(n^3 \log n)$ in [Hug92] and $\mathcal{O}(n^3)$ in [HWK94]. The method based on the Fourier transform suffers from aliasing problems. The He and al. approach seems more promising but the user interface should be developed.

The work of Lerios et al. [LGL95]

Description: Lerios et al. propose an extension of Beier and Neely's 2D approach [BN92] for volume-based representations of objects. As in 2D image-morphing techniques, a warping of the two volumes enclosing the objects is first computed. The intermediate objects are further obtained by blending the warped volumes.

Object representation: The method applies to sampled volumetric data.

Interaction: The user specifies corresponding features (either points, segments, rectangles or boxes) that should be transformed one into another during the morph process.

Comments: The method seems to be space and time consuming. Lerios et al. provides an example of a metamorphosis between a human's skull and an orangutan's skull. The correspondence, which is specified by 26 elements pairs, was established in 8 hours by a novice and 3 hours by an expert. Then 8 hours were necessary to obtain on a SGI indigo 2 a sequence of 50 in-between shapes, considering that the volumes are represented using a 300^3 voxels grid. The interface looks very heavy. It should also be noted that the intermediate shapes are not defined with precision and that a fuzzy zone appears around them.

The work of Pasko and Savchenko [PS95]

Description: Pasko and Savchenko define a metamorphosis between two general implicit surfaces by linearly interpolating the corresponding functions.

Object representation: The method is devoted to implicit surfaces defined by any type of real functions. A rasterization may be required to display the interpolated surface.

Interaction: The method is entirely automatic.

Comments: The definition of a metamorphosis is straightforward. There is no control over the transformation and the in-between shapes may be expensive to compute.

The work of Galin and Akkouche [GA96b, GA96a]

Description: Galin and Akkouche address the metamorphosis of implicit soft objects built from skeletons composed of convex shapes (points, line segments, polygons or polyhedra). As in [WBB+90, BBB+97], a correspondence between skeletal elements is first established. Interpolated shapes are expressed as soft objects whose skeletal elements are interpolated using Minkowski sums. As in [RK94] the use of Minkowski sums between two polyhedral skeletons is extended to Bézier transformation for a set of skeletons and the metamorphosis is extended to a blending between a set of soft objects.

Object representation: The method applies to soft objects built from convex skeletons.

Interaction: The user can establish a partial or a full correspondence between skeletal elements. The correspondence may also be automated using heuristics as in [WBB+90, BBB+97].

Comments: The user interaction may be crucial for the quality of the transformation. Once the correspondence graph is created, the complexity of the method is mainly the complexity of computing the Minkowski sums of all the corresponding skeletal components. The fact that the components are convex reduces the cost of the computation. Moreover, computing the Minkowski sums in the local coordinate systems of the component improve the control of the metamorphosis.

The work of Cohen Or et al. [COLS96, COLS98]

Description: Cohen-Or et al. propose a 3D metamorphosis technique based on a distance field interpolation (see Fig. 3). The objects are expressed as level-sets of distance functions composed with some space deformations. The technique involves two steps: a warp step , and an interpolation step. The warp is used to deform the 3D space in order to make the two objects (to be morphed) coincide as much as possible. The warp process is derived from the matching of two sets of feature points. The warp is decomposed into a rigid transformation followed by a small perturbation expressed in terms of radial functions. The interpolation reduces to a linear interpolation of distances fields deformed by the warp. The distance fields are computed from the voxelized objects using a 3D distance transform.

Object representation: The method applies to a discrete 3D space so that the objects should be first voxelized.

Interaction: The user interface allows to select feature - or anchor - points in each voxelized object space and to map the anchor points of the source object to the anchor points of the target object.

Comments: The authors report that it took 40 minutes on a SGI R4400 to create an intermediate 200^3 volume with 20 anchor points. The results are impressive and it is one of the most demonstrative volume approach up to date.

3.2 Boundary based approaches

In the following section we restrict our attention to morphing techniques that apply to objects represented by their boundaries. In this case, the computation of a metamorphosis is decomposed into two parts:

1. a **correspondence process** where each point of the boundary of the source object is mapped to a point of the target object boundary and vice-versa,
2. an **interpolation process** that defines the trajectory between each pair of corre-

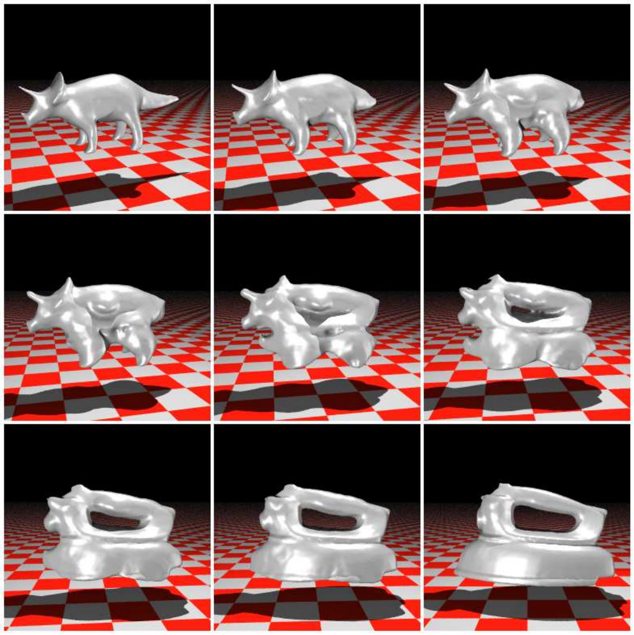

Figure 3: The metamorphosis of triceratops into an iron (Cohen-Or et al. 1998) (with the permission of ACM Transaction On Graphics)

sponding points.

The boundary approach for 3D objects is the natural extension of the 2D approach for contours. Again we concentrate on 3D techniques. The reader is referred to [SG92, SGWM93, ER95, GG95] for the 2D case.

The work of Hong et al. [HMTT88]

Description: Hong et al. propose a correspondence method for objects represented by facets that minimizes the distance of the corresponding facets centroids. The faces of the interpolated objects correspond to the faces of the object that has the greater number of faces.

Object representation: The method applies to objects represented by a list of faces.

No adjacency relationship is required.

Interaction: The method is entirely automatic.

Comments: The method seems to only work when the two shapes are similar. No timing is reported.

The work of Chen and Parent [CP89]

Description: Chen and Parent define a shape averaging based on planar contour interpolations. The shapes are supposed to be sliced into planar contours so that the averaging of two shapes reduces to the averaging of their corresponding contours. Two contours are interpolated using the intermediate cross-sections of a cylindrical volume built from the two contours.

Object representation:The technique applies to 3D objects represented by a set of planar polygons.

Interaction: In order to compute the interpolation between pairs of contours the user may indicate the first pair of corresponding points in each contour pair.

Comments: This is one of the first paper on 3D morphing. It is based on several heuristics for handling non convex polygons or cross-sections composed of more than one loop.

The work of Bethel and Uselton [BU89]

Description: Bethel and Uselton define a morph between two object meshes. They build a super mesh such that by collapsing edges or faces it can be reduced to either of the two input meshes. The super mesh is constructed while traversing simultaneously the dual graphs of the input meshes, adding vertices or faces to ensure the correspondence. The vertices of the super mesh are linearly interpolated between their corresponding location on the two input meshes.

Object representation: The technique applies to oriented polyhedra with equivalent (global) topologies.

Interaction: The user initiate the super mesh construction by specifying two corresponding faces.

Comments: The method is purely topological, it does not takes the geometry into account. As a result it may produce unappealing metamorphosis. The complexity of the correspondence process is proportional to the maximum number of faces of the two polyhedra.

The work of Kent et al. [KPC91, KCP92]

Description: Kent et al. define a morph where each point moves radially on a ray issued from a center point (see Fig. 4.). This is accomplished by projecting and further merging each object mesh onto a sphere.

Object representation: The technique applies to star-shaped polyhedra. It was further extended in [KCP92] to other classes of objects that can be easily projected onto a sphere.

Interaction: The user interface is reduced to the selection of a center point inside each polyhedron and to the selection of the relative orientation of the two objects.

Comments: The validity of the selected center point can be checked in time $\mathcal{O}(n)$ where n is the number of vertices of a polyhedron. The mass center of the kernel of each polyhedron is taken as a default center point and can be computed in time $\mathcal{O}(n \log n)$. The overall algorithm complexity is bounded by the merge step. It takes time $\mathcal{O}(I_{tot} log I_{tot})$ where I_{tot} is the total number of intersections between edges of the projected meshes. The approach is simple and perform very well for star-shaped polyhedra.

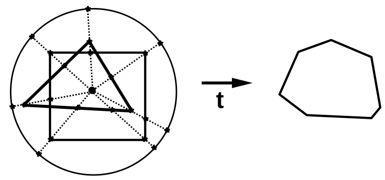

Figure 4: Radial interpolation (in 2D) between a square and a triangle (Kent et al. 1991)

The work of Parent [Par92]

Description: Parent uses a recursive process to build a common mesh subdivision: the user breaks each polyhedron into two sheets, tracing a closed path of edges on the surface of the polyhedron. Each boundary is subdivided into the same number of points in order to establish the vertex correspondence. The sheets are recursively split in parallel for each object. The process stops when each sub-sheet is composed of a single face. The so subdivided polyhedra have the same vertex/edge/face network. The interpolation between corresponding vertices is linear.

Object representation: The method applies to polyhedra homeomorphic to a sphere and can be extended to pairs of homeomorphic polyhedra.

Interaction:The algorithm can work automatically or can be guided by the user. In the later case, the user can control the sheet subdivisions at each step in the recursion.

Comments: This approach can be considered as an intermediate step between the work of [BU89] which is entirely automatic and the work of [DG96] where the user subdivide interactively each polyhedra into corresponding sheets. The process of sheets splitting introduces some coherence between geometry and topology. As this coherence is still not strong enough and as the interpolation is linear the resulting metamorphosis are not very smooth.

The work of Delingette et al. [DWS93]

Delingette et al. use a physical approach to deform a mesh under a set of predefined constraints. The objects are first approximated by simplex meshes (such meshes are dual of triangulations) obtained by deforming a sphere. A set of Eulerian operators is used to modify the mesh connectivity during the transformation. Once the objects are remeshed, their geometry is expressed by intrinsic parameters well adapted to simplex meshes. The metamorphosis between two such meshes is obtained by putting constraints that amounts to the transformation of the intrinsic parameters.

Object representation: Since the object are remeshed in a first place, the technique applies to any boundary representation.

Interaction: The user guides the approximation process by positioning and scaling the initial mesh for each object. He may intervene during the transformation in order to drag point when the animation falls into a local energy minimum.

Comments: the simplex meshes have faces with average degree 6. This may cause problems to render the mesh since the faces are likely to be non planar. The physical simulation is based on potential minimization and the transformation may stop in local minima. The method look otherwise attractive but few details are provided in the paper.

The work of Lazarus and Verroust [LV94, LV97]

Description: Lazarus and Verroust extend Kent et al.'s method [KPC91] for cylinder-like objects (see Fig. 5). Given a 3D curve inside each object, two cylindrical meshes are built to approximate the two objects, taking into account their salient features. The metamorphosis between the two objects is performed on these cylindrical meshes. It consists in an interpolation of the two 3D curves composed with a radial interpolation of each sampling point of the mesh. In [LV96] the authors propose an algorithm for computing an axis inside a polyhedral shape.

Object representation: The method is devoted to polyhedra that are star-shaped around an axis.

Interaction: The user builds a 3D curve inside each object or select a point on each object to compute automatically a candidate axis. The axes animation can be modified by the user with simple angular parameters. The user can also control the interpolation schedule along the axis.

Comments: This is one of the few morphing techniques with non trivial (linear or spline) interpolation of the vertex positions. The user has a good control over the axis animation which guides the transformation. The cylindrical meshes are computed in a few seconds for objects of 10 thousand faces. Once each object has been remeshed the interpolation is processed at interactive time. Weak point: determining whether an object is star shaped around an axis and finding such an axis can be a non trivial task.

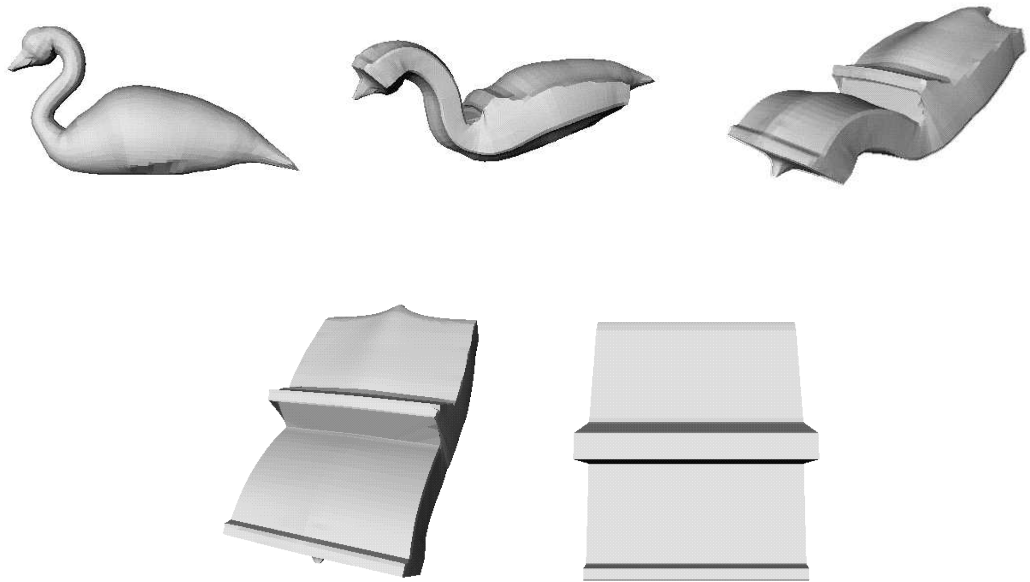

Figure 5: The metamorphosis of a swan into a piano (Lazarus and Verroust)

The work of Decaudin and Gagalowicz [DG94]

Description: Decaudin and Gagalowicz compose two inter-penetrating shapes by creating a new shape including them with a volume equal to the sum of their volume. The metamorphosis between two star shaped polyhedra A and B is obtained by composing the two scaled version of A and B as in the method of Kaul and Rossignac. The volume of the in-between shape varies linearly from the volume of A to the volume of B.

Object representation: The method applies to star-shaped polyhedra.

Interaction: The user specifies a point inside the two objects such that the objects are star-shaped with respect to this point.

Comments: The metamorphosis is directly inspired from the work of Kent et al. [KPC91]. The only change is the interpolation schedule of the points that provides a linear interpolation of the volume. A test is also introduced in order to refine the interpolated mesh in curved regions.

The work of DeCarlo and Gallier [DG96]

Description: DeCarlo and Gallier tackle the problem of morphing two polyhedral objects with different topologies. They use a sparse control mesh on each surface in order to define a mapping between the input objects and they treat the change in topology with degenerate faces.

Object representation: The method applies to general triangulated polyhedral surfaces.

Interaction: The user is asked to define a rough control mesh on both objects and to associate one by one every face of one object to a face of the other object. This implies that the two control meshes have the same number of faces and have almost the same topology except where change of topology should occur. Edges and vertices of the meshes must also be mapped between the two objects. Note that an edge can be mapped to a vertex but that two corresponding edges or vertices must at least belong to a pair of corresponding faces. The user is also asked to create relatively flat faces in order to produce a relatively smooth transformation.

Comments: The user interface could certainly be improved. The authors report that it took one hour and 15 minutes of interaction in order to create a correspondence between two relatively simple shapes.

This paper has the merit to deal with topological changes but the user should clearly get some good notions on topology before trying to use this tool. Also since the interpolation of geometry is quite straightforward it is not clear from the paper that morphing complex shapes will always result in a nice transformation.

The work of Sun et al. [SWC97]

Description: The work presented by Sun et al. addresses the interpolation problem for morphing two polyhedral models. The correspondence is supposed to be known so that the input polyhedra should have isomorphic meshes. As for linear interpolation Sun and al. attempts to define a transformation invariant under rigid displacement. To do so, they interpolate the face normals using a propagation paradigm. These interpolated normals are further used to interpolate the vertex positions using the same propagation paradigm. It should be noted that the normal to the faces of the interpolated polyhedron are actually different from the interpolated normals.

Object representation: The method applies to pairs of polyhedra with isomorphic meshes.

Interaction: In order to initiate the propagation the user selects two adjacent faces and two vertices belonging to the first face on each input polyhedron. The selection can also be done automatically.

Comments: Due to the use of propagations this method is numerically fragile. The resulting interpolation depends on the face and vertex graph traversal used for the propagations. This method is not intrinsic to the object geometry as opposed to what is claimed by the authors. This is however one of the few methods that tackle the interpolation problem.

The work of Kanai et al. [KHSK98]

Description: Kanai et al. present an automatic correspondence method between two polyhedral meshes. Each mesh is first embedded in a planar unit disk using harmonic maps. The correspondence is established by overlaying the two embedded meshes. Since the embedded meshes may have very dense parts, a particular attention is given to the numerical problems. The interpolation is linear.

Object representation: The embedding step applies to polyhedral meshes homeomorphic to a disk. The method is extended to meshes homeomorphic to a sphere by cutting into two sheets.

Interaction: The user can modify the correspondence by rotating the embedded meshes around their center point.

Comments: Thanks to the harmonic mapping the correspondence is quite well related to the geometry of the objects. It remains that the interpolation is linear and there is little control over the transformation. The extension to closed meshes is not clear: the problem of stitching back, in a smooth manner, the two embedded sheets for closed genus-0 meshes is not addressed.

The work of Gregory et al. [GSL+98]

Description: Gregory et al. present an approach similar to the DeCarlo and Gallier's approach. They also make use of a sparse control net on both objects in order to establish a correspondence. They do not deal with object having non equivalent topologies.

Object representation: The method applies to a pair of polyhedral objects with equivalent topologies given by their *winged-edge* representation.

Interaction: The user specifies two control nets on both objects. For every arc of the net the user select a pair of points on each surface and a geodesic is computed between each pair to form an arc of the net. As in [DG96] the user must be careful to create equivalent control nets on both objects so that the faces can be put into correspondence while preserving adjacencies. The user is further asked to specify trajectories, using Bézier curves, for each pair of corresponding vertices of the two control nets.

Comments: As in [DG96] the user interaction is extremely heavy and according to the authors it took them 6 hours to create a morph between a human and a triceratops.

4 Analysis

We presented a survey of the main methods that were developed in order to create a metamorphosis between two shapes. We highlighted the relationship between the morphing approaches and the type of object representation chosen for the implementation of the method. The principles lying under each method and the ability of a user to control a morph sequence are strongly dependent on the type of shape description.

Methods based on a volumetric description of the shapes tend to give good results even for complex shapes with different topologies. Fully automatic methods have been proposed [Hug92, HWK94, PS95]. Because there is no intrinsic way for transforming one shape into another this cannot always produce a desirable morph. Though weighted Minkowski sums [KR91, RK94] provide an efficient automatic morphing tool, it only works well, in practice, for the restricted class of convex shapes. Approaches using an underlying skeletal structure as in [WBB+90, BBB+97, GA96b, GA96a] offer a simple and intuitive control to the user. Here also, there are some restrictions on the class of objects that can be morphed together. Other volume based methods [LGL95, COLS98] let the the user put points or feature elements into correspondence in order to control the morph animation. Some of the most impressive and nicest animation where produced with such methods. It remains that the choice of feature elements and their matching can be tricky especially when the source and target shapes are very dissimilar.

The user interface could probably be improved. One could imagine that the objects are made of clay and transform (approximately) the source object into the target object with modeling and sculpting tools such as FFD and EFFD (see the survey of Bechmann on deformation tools ([Bec94]). The main difficulty with this approach is to combine all the successive deformations performed by the user into a single continuous transformation. The risk is indeed to obtain a "piecewise" transformation where the successive steps of the modeling clearly appear in each subsequence of the morph.

The control of the number of connected component during the morph is also a non solved problem. The problem can appear when the source object is very different from the target object (one can think of the transformation of a galleon into a two holed torus). Note also that texture mapping is still an issue when dealing with volume based models.

Apart from the restrictions cited above the volume based methods have proved to be very efficient especially for objects with different topologies. The situation is not so attractive when dealing with objects represented by their boundaries. Again fully automatic methods [HMTT88, BU89, SWC97] generally fail to produce pleasing transformations. While providing little control over the deformation Kent et al. [KPC91, KCP92] obtain nice animations by restricting the type of objects to star shaped polyhedra and some other class of shapes. An intermediate level of interaction is proposed in [LV97]. They apply their method to a larger class of objects (cylinder-like shapes) but this is still a restricted case. They provide, however, good control over the morph sequence through the control of the animation of the shape “axes”. Interactivity reaches its paroxysm in [DG96, GSL$^+$98]. Here the user is asked to construct sparse control meshes on the source and target objects. The construction of the control meshes seems particularly delicate as the source and target control meshes must be topologically equivalent. Also the shapes and sizes of corresponding faces or edges should be relatively similar in order to obtain smooth transformations. DeCarlo and Gallier [DG96] deal with change of topology but this requires even more skills in topology from the user part. Note that these last two approaches mainly tackle the correspondence problem but do not really solve the interpolation issue. In fact, Gregory et al.[GSL$^+$98] recognize themselves that designing trajectories on a sample set of points lying on the surface is not an adequate way to specify the transformation between the two corresponding meshes.

Unlike most of the other methods, three of them [DWS93, LV97, SWC97] address the interpolation problem to produce non trivial vertex position interpolation. In [DWS93, SWC97] the geometry of the objects is converted into a set of intrinsic parameters independent of the coordinate system. Sun et al.[SWC97] attempt to construct objects by interpolating their intrinsic parameters while Delingette et al. [DWS93] simulate a physical animation where the forces are related to a certain distance between the intrinsic parameters of the input objects. The later approach seems more realist as it is not obvious (when possible) to reconstruct a shape from a set of intrinsic parameters.

5 Perspectives

As we already noticed in section 2 interpolation of boundary models are difficult. In this section we try to provide new insights into that particular problem. After a brief discussion on the notion of shape we sketch some possible strategies for further improvement of morphing techniques.

The shape of an object

Boundary representation usually contains both topological and geometrical information. While the topological information has a combinatorial nature the geometry may vary continuously. Moreover these two types of information constrain each other. For instance an object topologically equivalent to a torus cannot be convex (convexity is a geometric characteristic). Actually every closed bounded convex surface is homeomorphic to a sphere. The notion of *shape* of an object - the three dimensional realization of the topological and geometrical information - is thus difficult to apprehend. This is yet a central concept to metamorphosis.

On one hand it is relatively easy to extract the topological characteristics by just looking at an object: one can tell the number of connected components or the number of handles of an object. On the other hand the geometry is more complex to describe. One can describes the geometry in terms of features such as bulges or concavities but this may not be enough to characterize the shape without ambiguities. This description is rather local and the problem is to combine the set of local descriptions into a global form. A cactus may be composed of a tube and 3 bulges but it will be impossible to reconstruct a given cactus if we do not know how the bulges are attached relatively to its tube.

A more elaborate description is obtained by looking at an object from different point of views or by rotating, far away, that object around its center point. Each point of view or rotation gives rise to a silhouette. The set of point of views or the set of rotations can be partitioned into cells according to the similarity of the corresponding silhouettes. This relates to the notion of *aspect graph* [GCS91, PPK92] and gives a complete characterization of the shape. Although attractive because of its combinatorial structure the aspect graph seems far too complex for practical use to morphing purposes. Moreover it is not clear how to reconstruct a 3D shape from its aspect graph.

Another global description can be obtained with a set of intrinsic parameters such as the edge lengths and the dihedral angles between adjacent faces. These parameters encode the mesh geometry independently of any orthonormal coordinate system. Sun et al. [SWC97] and Delingette et al. [DWS93] use such a formulation. The main drawback with this approach is that a set of intrinsic parameters does not necessarily correspond to any embeddable geometry, as opposed to a set of vertex positions. It is therefore difficult to interpolate directly the intrinsic coefficients. As mentioned above Delingette et al. get round this difficulty by forcing the parameters to change through a physical simulation rather than expressing that change explicitly.

Shinagawa et al. [SKK91] propose to encode the shape of an object by means of a graph. This graph called the Reeb graph is constructed from the level sets of a numerical function defined over the surface of the object. Lazarus and Verroust [LV96] use the geodesic distance to a source point as such a numerical function. They obtain a skeleton graph which is a good indicator on the shape of the object. (This skeleton is different from the classical skeleton defined by centers of maximal spheres.) The skeleton graph contains the topological information. The shape of the graph, that is the edge embeddings and the branchings, provide a rough description of the shape of the object. This information is not complete but intuitive and easy to manipulate. The surface of the object is replaced by a one dimensional structure and part of the geometry is "translated" in terms of topology; in the above example of a cactus the three bulges give three branchings on the skeleton while the surface of the cactus is homeomorphic to a sphere.

The last two shape descriptions (intrinsic parameters or skeletons) look more appropriate and promising for morphing applications. Another option, using skeletons, would be to "discretize" the set of shapes by storing predefined shapes. These shapes would play the role of *generic* shapes corresponding to various skeletons. For example we can have a generic quadruped, a generic human, a generic car etc. A given shape can then be perceived as a deformed generic shape; once the corresponding closest generic shape is selected it can be deformed into the given shape. The main deformation can be performed using the skeleton as a deformation tool while the details can be obtained with local projections or physical simulations.

Morphing strategies

In the sequel we envision some strategies for the metamorphosis of objects represented by their boundaries. We do not intend to cover all the possibilities but rather to point out some ideas based on the previous analysis and discussion that could be deepened or explored.

It would certainly be useful to provide an interactive deformation tool with the capability of combining a sequence of deformations into a single one. It should also be possible to create a continuous warping between any two deformations. Such a tool could be used to establish a crude transformation between two shapes.

Correspondence is an important stage for the animator. It tells which part of the source object will transform into which part of the target object. The solution proposed in [DG96, GSL+98] seems too constraining for the user. It gives the impression that the user is asked to help the computer rather than the opposite. It should be possible to design a more attractive tool.

The following three steps provides an attempt to design a correspondence tool.

1. In a first step, each input mesh could be projected onto a simple shape (a unit disk, a sphere, ...) using harmonic maps as in [KHSK98] or using other techniques as in ([MYV93, CCO97, ST98]). The type of projection used is not very important. All what is needed is a continuous projection. If the objects are textured we could keep the texture on the projected meshes. These projected meshes should play the role of a planisphere or a globe.

2. The correspondence is really established during a second step. Here, the user should be provided an interactive tool to apply global or local deformations on the projected meshes. Global deformations include rotations of the unit disk or the unit sphere. Local deformations include dragging feature points and can be implemented with the technique described in [FM97]. The aim is to map features from the two input meshes. To help the user in this task we could imagine that picking a point with the mouse on a projected mesh cause the highlight of the corresponding point in the

other projection.

3. The warped projected meshes are merged into a super mesh as in [KPC91] and further projected back on the input meshes.

The first step is probably the most difficult and should be adapted to objects homeomorphic to meshes with non 0 genus. As mentioned in the previous discussion on the shape of an object shape descriptors such as skeletons could help. It still remains that it is not easy to give a precise parameterization of a surface with a skeleton. Branching are especially difficult to deal with. One possible way to cope with this problem is to use a set of generic shapes as mentioned at the end of the previous discussion on shapes. This time we should store a set of generic shapes as well as their projection onto simple *maps* (spherical, toroidal, ...). This would provide a parameterization of a general shapes up to a small deformation that fits a given shape to its associated generic shape.

After the correspondence is established the interpolation could be computed with a physical simulation to interpolate intrinsic geometric parameters as in [DWS93]. There is however a potential issue if intrinsic parameters such as dihedral angles are used in conjunction with a super mesh obtained by overlaying. A number of adjacent faces will be coplanar. These faces correspond to faces of the input meshes subdivided by the overlay. This may introduce discontinuities in the variation of the dihedral angles along the super mesh and consequently the interpolation will not be smooth.

If the correspondence was established with generic shapes one could also imagine to store generic transformations and use them as underlying coarse interpolations. The interpolation should take into account the deformations performed by the user (with the deformation tool mentioned above).

All the previous stage of the morph should certainly take advantage of a multiresolution representation as the one presented in [LDS+98]. This representation is definitely relevant in the correspondence context since it provides simplified meshes together with a mapping (parameterization) between the different level of resolution. Note that the use of simplified meshes may partly solve the problem of coplanar faces mentioned above. The mesh overlay may indeed be performed on simplified meshes and the resulting super mesh may be projected back on the geometry of the *input* meshes. This way, faces are less likely to be coplanar since they do not correspond to the subdivision of the input meshes.

6 Conclusion

Both boundary and volume based approaches can be effective. However, looking at the results it seems that the volume based approach has reached its maturity while the boundary approach could still benefit from further improvements. We outlined here some extensions that could be made in this context, in particular for the correspondence problem. It remains that most efforts have been dedicated to the correspondence problem and we still lack of intuitive solutions to control the shape interpolation. Change of topology during a metamorphosis is another subject that we have not treated. We hope our review will stimulate further interest.

References

[BBB$^+$97] J. Bloomenthal, C. Bajaj, J. Blinn, M-P. Cani-Gascuel, A. Rockwood, B. Wyvill, and G. Wyvill. *An Introduction to Implicit Surfaces*, chapter 8, pages 243–269. Morgan Kaufmann Publishers, 1997.

[Bec94] D. Bechmann. Space deformation models survey. *Computer and Graphics (Special Issue : Modeling, Rendering and Animation Techniques)*, 18(4):571–586, 1994.

[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.

[CCO97] E. Carmel and D. Cohen-Or. Warp-guided object-space morphing. *The Visual Computer*, 13(9-10):465–478, 1997.

[COLS96] D. Cohen-Or, D. Levin, and A. Solomovoci. Contour blending using warp-guided distance field interpolation. In *Proceedings of Visualization'96*, pages 165–172, San Francisco, 1996.

[COLS98] D. Cohen-Or, D. Levin, and A. Solomovoci. Three-dimensional distance field metamorphosis. *ACM Transactions on Graphics*, 17(2), 1998.

[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.

[DG94] P. Decaudin and A. Gagalowicz. Fusion of 3D shapes. In *Fifth Eurographics Workshop on Animation and Simulation*, Oslo, September 1994.

[DG96] D. DeCarlo and J. Gallier. Topological evolution of surfaces. In *Graphics Interface'96*, pages 194–203, 1996.

[DWS93] H. Delingette, Y. Watanabe, and Y. Suenaga. Simplex based animation. In *Computer Animation'93, Models and Techniques in Computer Animation*, Computer Graphics International, 1993.

[ER95] M. Shapira Etzion and A. Rappoport. Shape blending using the star-skeleton representation. *IEEE Computer Graphics and Applications*, 15(2):44–50, March 1995.

[FM97] K. Fujimura and M. Makarov. Homotopic shape deformation. In *International Conference on Shape Modeling*, 1997.

[GA96a] E. Galin and S. Akkouche. Blob metamorphosis based on Minkowski sums. In *Eurographics'96*, pages C–143–C–153, Poitiers, 1996.

[GA96b] E. Galin and S. Akkouche. Shape constrained blob metamorphosis. In *Implicit Surface'96*, pages 9–23, Eindhoven, October 1996.

[GCDV96] J. Gomes, B. Costa, L. Darsa, and L. Velho. Graphical objects. *The Visual Computer*, 12(6):269–282, 1996.

[GCS91] Z. Gigus, J. Canny, and R. Seidel. Efficiently computing and representing aspect graphs of polyhedral objects. *IEEE Transaction on pattern analysis and machine intelligence*, 13(6):5542–551, 1991.

[GG95] E. Goldstein and C. Gotsman. Polygon morphing using a multiresolution representation. In *Graphic Interface'95*, pages 247–254, Québec, Canada, June 1995.

[GS98] P. Gao and T.W. Sederberg. A work minimization approach to image morphing. *The Visual Computer*, 1998. this issue.

[GSL+98] A. Gregory, A. State, M. Lin, D. Manocha, and M. Livingston. Feature-based surface decomposition for correspondence and morphing between polyhedra. In *Computer Animation'98*, Philadelphia, June 1998.

[HMTT88] T. M. Hong, N. Magnenat-Thalmann, and D. Thalmann. A general algorithm for 3-D shape interpolation in a facet-based representation. In *Graphic Interface'88*, pages 229–235, 1988.

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

[HWK94] T. He, S. Wang, and A. Kaufman. Wavelet-based volume morphing. In R. Daniel Bergeron and Arie E. Kaufman, editors, *Proceedings of Visualization'94*, pages 85–92. IEEE Computer Society Press, October 1994.

[KCP92] J.R. Kent, W.E. Carlson, and R.E. Parent. Shape transformation for polyhedral objects. *Computer Graphics (SIGGRAPH'92)*, 26(2):47–54, July 1992.

[KHSK98] T. Kanai, H., Suzuki, and F. Kimura. 3d geometric metamorphosis based on harmonic map. *Computer Graphics Forum*, 1998. to appear.

[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.

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

[LCHS94] S.Y. Lee, K.Y. Chwa, J. Hahn, and S.Y. Shin. Image morphing using deformable surfaces. In *Computer Animation'94*, pages 31–39, 1994.

[LCSW95] S.Y. Lee, K.Y. Chwa, S.Y. Shin, and G. Wolberg. Image metamorphosis using snakes and free-form deformations. *Computer Graphics (SIGGRAPH'95)*, 29:439–448, August 1995.

[LDS+98] A. W. F. Lee, D. Dobkin, W. Sweldens, L. Cowsar, and P. Schroeder. Maps: Multiresolution adaptive parameterization of surfaces. In *Computer Graphics (SIGGRAPH '98)*, 1998.

[LGL95] A. Lerios, C.D. Garfinkle, and M. Levoy. Feature-based volume metamorphosis. *Computer Graphics (SIGGRAPH'95)*, 29:449–464, August 1995.

[LV94] F. Lazarus and A. Verroust. Feature-based shape transformation for polyhedral objects. In *Fifth Eurographics Workshop on Animation and Simulation*, Oslo, September 1994.

[LV96] F. Lazarus and A. Verroust. Décomposition cylindrique de polyèdre et courbe squelette. *Revue Internationale de CFAO et d'Infographie*, 11(4), November 1996.

[LV97] F. Lazarus and A. Verroust. Metamorphosis of cylinder-like objects. *International Journal of Visualization and Computer Animation*, 8(3), July 1997.

[LWS98] S. Lee, G. Wolberg, and S. Y. Shin. Polymorph: Morphing among multiple images. *IEEE Computer Graphics and Applications*, 18(1):58–71, 1998.

[MYV93] J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In *Computer Graphics (SIGGRAPH '93)*, volume 27, pages 27–34, August 1993.

[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.

[PPK92] S. Petitjean, J. Ponce, and D. J. Kriegman. Computing exact aspect graphs of curved objects: Algebraic surfaces. *International Journal of Computer Vision*, 9(3):231–255, 1992.

[PS95] A. Pasko and V. Savchenko. Constructing functionally defined surfaces. In *Implicit Surface'95*, pages 97–106, Grenoble, France, April 1995.

[RF96] V. Rajan and A. Fournier. Matching and interpolation of shapes using union of circles. In *Eurographics'96*, pages C–129–C–142, Poitiers, 1996.

[RK94] J. Rossignac and A. Kaul. AGRELS and BIBs: metamorphosis as a Bézier curve in the space of polyhedra. In *Eurographics'94*, pages C–179–C–184, Oslo, September 1994.

[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.

[SKK91] Y. Shinigawa, T.L. Kunii, and Y.L. Kergosien. Surface coding based on Morse theory. *IEEE Computer Graphics and Applications*, 11(5):66–78, 1991.

[ST98] A. Shapiro and A. Tal. Polyhedron realization for shape transformation. *The Visual Computer*, 1998. this issue.

[SWC97] Y. M. Sun, W. Wang, and F. Y. Chin. Interpolating polyhedral models using intrisic shape parameters. *The Journal of Visualization and Computer Animation*, 8:81–96, 1997.

[TF98] R. Tam and A. Fournier. Image interpolation using unions of spheres. *The Visual Computer*, 1998. this issue.

[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.

[Wol98] G. Wolberg. Image morphing: a survey. *The Visual Computer*, 1998. this issue.

Author's biographies

Francis lazarus is an assistant professor of computer science at the University of Poitiers, France. He received his Ph.D. in computer science in 1995 from the University of Paris VII, France. He was a postdoctoral researcher at the IBM T.J. Watson Research Center, Yorktown Heights, NY, from 1996 to 1997. He is a coauthor of the IBM VRML 2.0 binary standard proposal. His research interests include geometric modeling, geometry compression, computer animation and three-dimensional morphing.

Anne Verroust is a senior researcher at INRIA Rocquencourt, France. She obtained her Thèse de 3^e cycle in Computer Science and her Thèse d'Etat in Computer Science from the University of PARIS-SUD, France, in 1983 and 1990. Her research interests include geometric modeling, animation, computer graphics and computer aided design.

approaches	type of data	user's interaction	control
[WBB+90, BBB+97]	soft objects defined by 2D primitives	specifies the correspondence between skeletons or uses heuristics	correspondence & interpolation
[KR91, RK94]	polyhedra	specifies a local reference frame for each object	poor when non convex shapes
[Hug92]	sampled volumetric data	none	none
[HWK94]	sampled volumetric data	none	none
[LGL95]	sampled volumetric data	constructs corresponding feature elements	warping & blending
[PS95]	functionally defined surfaces	none	none
[GA96b, GA96a]	soft objects defined by convex primitives	specifies the correspondence between skeletons	correspondence & interpolation
[COLS96, COLS98]	sampled volumetric data	designates corresponding anchor points	warping & interpolation

Table 1: Classification of the 3D volume-based approaches.

approaches	type of data	user's interaction	problem solved
[HMTT88]	polyhedra	none	correspondence
[CP89]	sliced objects	selects corresponding points on each 2D contour	correspondence
[BU89]	homeomorphic polyhedra	selects a face on each object	correspondence
[KPC91, KCP92]	star-shaped polyhedra	specifies a center point inside each object	correspondence & interpolation
[Par92]	simple polyhedra	constructs a closed path to subdivide the surface of each object	correspondence
[DWS93]	objects represented by their boundary	positions the initial meshes associated to the objects and drags some points during the transformation	correspondence & interpolation
[LV94, LV97]	cylinder-like polyhedra	specifies a 3D curve inside each object (interactively or automatically built)	correspondence & interpolation
[DG94]	star-shaped polyhedra	designates a common center point	correspondence & interpolation
[DG96]	triangulated polyhedra	constructs a rough control mesh and indicates the topological changes	correspondence
[SWC97]	simple polyhedra	selects two adjacent faces on one object (if the correspondence is not established it is done by [BU89])	interpolation
[KHSK98]	simple polyhedra	constructs a closed path to subdivide the surface of each object	correspondence
[GSL+98]	homeomorphic polyhedra	constructs a rough control mesh and gives the trajectories of some vertices	correspondence & interpolation

Table 2: Classification of the 3D boundary-based approaches.