Fusion of 3D Shapes

*Philippe Decaudin**
Andre Gagalowicz

INRIA Rocquencourt, SYNTIM
BP 105, 78153 Le Chesnay Cedex, FRANCE

* http://www.antisphere.com

Abstract

Many modeling techniques for geometric objects have been developed; they allow the user to create, deform, animate objects and produce complex ones from simple primitives. This paper presents a new technique which composes two 2D or 3D interpenetrating shapes by creating another one including them and having a smooth aspect. By moving into each other, composed objects fuse smoothly and naturally. To determine the fusion, the technique is based upon a constraint : the resulting shape has the same volume (or area in 2D) as the sum of the components. This technique is available for shapes having a polygonal (in 2D) or a polyhedral (in 3D) description. Inside a modeler, it can be utilized to design modeling and animation tools. We present three of them: a creation tool, an animated deformation tool and a morphing tool which is an adaptation of our technique to this problem. One of the advantages of these tools is that the user may easily guess what will be the result.

1 Introduction

An important task for modeling and animating techniques is to produce complex objects and animate them. Many tools have been already developed to create such objects.

Some of them use the composition of simple primitives to obtain more complex shapes. Surfaces of smooth shapes can be generated by patches like B-spline ([3],[1]). They can be approximated by a vertex/edge/facet network which allows to mix them with surfaces obtained by other methods. By moving control points of the patches, surfaces can be animated.

Implicit surfaces are an other way to create and animate smooth shapes. They can simulate smooth fusions between objects ([15], [14], [13]).

A different manner of constructing a complex shape is to deform a preexisting one with a given deformation tool (Free-Form Deformation [11], [4]). Then, the tool can be animated (Animated Free-Form Deformation [8]).

Interesting animated shapes may also be obtained through the metamorphosis from one shape to another one ([10],[7]) or by using composition laws between two shapes. As an example, in [9], the Minkovski sum between two object shapes is computed.

We present here a new technique to compose two interpenetrating shapes. It creates another shape corresponding to a type of fusion of these shapes where the final shape includes the initial ones and the volume of the final shape is the sum of the volumes of the initial ones (see figure 1). The volume preservation is not a main feature, but we uses this constraint to define the fusion method (see section 3). Our method allows the composition of shapes having sharp edges because it has been designed to deal with polyhedral shape models (this property differs from implicit modeling). The result have also a polyhedral description on which all tools available for polyhedral objects can be applied. An adaptive scheme has been developed to control the quality of the smoothness of the composed shapes.

Case of star-shaped[1] objects will be first presented. Star-shaped objects and convex objects (which are included in the star-shaped object class) can be described by a polar function. We will use this description to present the composition technique. However our technique does not only work on star-shaped objects; we present an extension of the algorithm which allows to compose a vast class of objects with a star-shaped object. We have developed tools based upon our composition technique : they allow the user to create complex shapes and to apply animated deformation on shapes.

2 Composition rules

Let A and B be two two-dimensional or two three-dimensional objects represented by a model of their shapes. In 2D, we consider polygonal models, and in 3D, polyhedral ones (corresponding to a vertex/edge/facet network). Our aim is to create a third shape C also represented by a polygonal or polyhedral model so that :

- C includes A and B, i.e. every point in A or B is in C,
- the area (or volume) of C is equal to the area (or volume) of A plus the area (or volume) of B,
- The shape of C is as smooth as possible (i.e. if the surfaces of A and B are continuously differentiable, the surface of C is continuously differentiable).

Figure 1 shows the result of the composition of two discs. The next section describes the algorithm used.

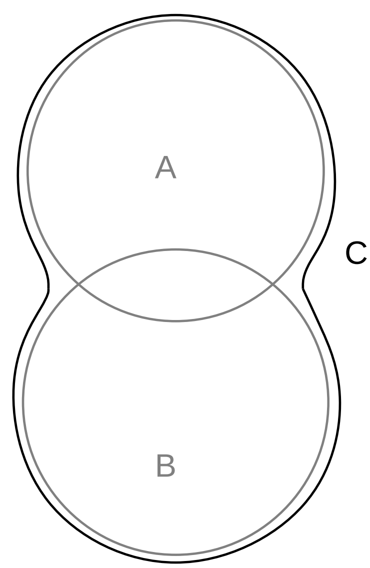

Figure 1: Composition of two discs

[1]an object A is star-shaped with respect to a point H if, for any point M in A, the segment [HM] is in A

3 Fusion of two shapes

3.1 Composing star-shaped objects

In this section, A and B are two objects star-shaped with respect to a single point H (then H belongs to the intersection between A and B), this point is fixed[2].

An intuitive way to describe the fusion technique is to consider that A and B have homogeneous material density. When superposing A and B, the material density of $A \cap B$ is double; then, we have an excess of material in the region corresponding to the area (or volume) of $A \cap B$. The idea is to transport this excess to the boundary of the objects using H as the center of a radial projection. The result C, the composition of A and B, is an object also star-shaped with respect to the point H. The following subsections give further details about this method.

3.1.1 2D objects

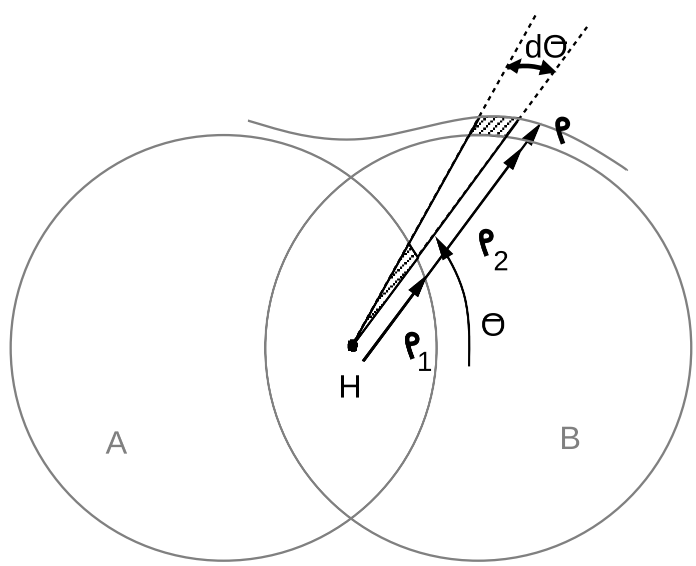

Figure 2: Notations (in 2D-space)

The shapes of objects A and B can be described by two polar functions $\rho_1(\theta)$ and $\rho_2(\theta)$ where H is the coordinates center (see figure 2). Let $\Delta v_1(\theta)$ and $\Delta v_2(\theta)$ be the areas of angular sectors defined by $(\rho_1(\theta), d\theta)$ and $(\rho_2(\theta), d\theta)$ respectively.

$$\Delta v_1(\theta) = \frac{1}{2}\rho_1(\theta)^2 d\theta$$

$$\Delta v_1(\theta) = \frac{1}{2}\rho_1(\theta)^2 d\theta$$

We want to find a polar function $\rho(\theta)$ associated with object C so that $\Delta v(\theta)$, the area of the angular sector defined by $(\rho(\theta), d\theta)$, satisfies:

$$\Delta v(\theta) = \Delta v_1(\theta) + \Delta v_2(\theta)$$

This formula can be interpreted as the preservation of the total quantity of material inside an angular sector.

$$\Delta v(\theta) = \frac{1}{2}\rho(\theta)^2 d\theta$$

$$\frac{1}{2}\rho(\theta)^2 d\theta = \frac{1}{2}\rho_1(\theta)^2 d\theta + \frac{1}{2}\rho_2(\theta)^2 d\theta$$

[2]The problem of the choice of H will be discussed later.

By solving this equation, we determine that

$$\rho(\theta) = \sqrt{\rho_1(\theta)^2 + \rho_2(\theta)^2} \quad .$$

Using this polar function, the shape of C can easily be computed and drawn.

The object C satisfies properties defined in section 2:

- C includes A and B (for each θ, $\rho(\theta) \geq \rho_1(\theta)$ and $\rho(\theta) \geq \rho_2(\theta)$),
- C is built so that its area is equal to the area of A plus the area of B,
- we can prove that if the shapes of A and B are smooth, the shape of C is also smooth (if ρ_1 and ρ_2 are $\mathcal{C}^n$ then ρ is $\mathcal{C}^n$)[3] .

3.1.2 3D objects

The extension to 3D objects is immediate: A and B are described by two spherical functions $\rho_1(\theta, \varphi)$ and $\rho_2(\theta, \varphi)$. By solving the conservation of material equation along a 3D angular sector, C can be described by the spherical function

$$\rho(\theta, \varphi) = \sqrt[3]{\rho_1(\theta, \varphi)^3 + \rho_2(\theta, \varphi)^3} \qquad (a).$$

The next step is to design an efficient algorithm able to create a polyhedral shape model of the object C. We first suppose that A and B have no sharp edges and in the next section, we extend this algorithm to objects containing sharp edges.

The shapes of A and B are described by vertex/edge/facet networks. The aim is to create a vertex/edge/facet network corresponding to the shape of C. The algorithm we propose creates a simple object like a tetrahedron centred on H, and "projects" its shape on the surface of the object C so that it approximates the shape of C as closely as possible, by refining the vertex/edge/facet network. Details of the iterative algorithm are:

- 1. Create a tetrahedron centred on H (4 vertex, 6 edges, 4 facets).
- 2. Project each vertex M ($\neq$ H) on C using the following projection formula:

  $$M' = H + \left(\sqrt[3]{\left\|\vec{HI}\right\|^3 + \left\|\vec{HJ}\right\|^3}\right)\frac{\vec{HM}}{\left\|\vec{HM}\right\|}$$

  where I (resp. J) is the intersection of the half straight line [HM) with the object A (resp. B). This formula is directly implied by the formula (a).
- 3. For each facet, check if the facet is lying on the surface of C. Here is a method to perform this test (which is not the only possible solution).
  - 3.1. For each vertex, compute a normal vector by interpolating the normals of the facets connected to the vertex. Objects A and B are assumed smooth (no sharp edges), therefore C is also smooth[4]. "Smoothness" means the continuity of normal vectors at surface points. This smoothness criterion is used as the basis of the following test:
  - 3.2. For each edge, let $\vec{n_1}$ and $\vec{n_2}$ be the two normal vectors of vertices of the edge as defined above. Let ε represent the divergence between $\vec{n_1}$ and $\vec{n_2}$ (if $\vec{n_1}$ and $\vec{n_2}$ are normalized, $\varepsilon = \|\vec{n_1} - \vec{n_2}\|$). If $\varepsilon > \varepsilon_{max}$ then the edge does not lie on C, and it has to be subdivided (ε_{max} is a constant defined by the user: the lower ε_{max}, the better the approximation of C, but the greater the number of facets necessary for the approximation) .
  - 3.3. For each facet, if an edge of the facet has to be subdivided, divide the facet accordingly (figure 3 presents a way to divide triangular facets). The division of some facets may create new vertices.

[3]The proof is obvious and is not presented here.

[4]The proof is the same as in the 2D case and is not presented here

- 4. If new vertices have been created, go to step 2, otherwise a "good" approximation of the shape of B has been built.

The result is a polyhedral description of the shape C, adjusted to be a good approximation (an ε_{max} normal continuity). This means that if the shape C is very smooth, few facets will be created, but if the shape C has a high curvature, many facets will be created (Figure 4 illustrates this point).

This algorithm is only available for smooth star-shaped (with respect to the center of composition H) objects, but it is the basis of the general technique. We will now extend this algorithm to the case of star-shaped objects containing sharp edges.

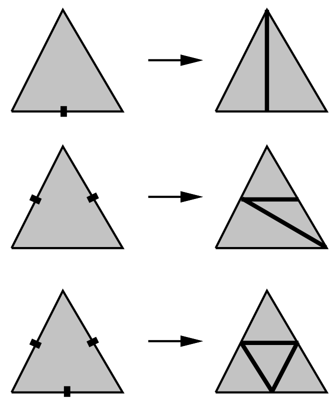

Figure 3: Division of a triangular facet

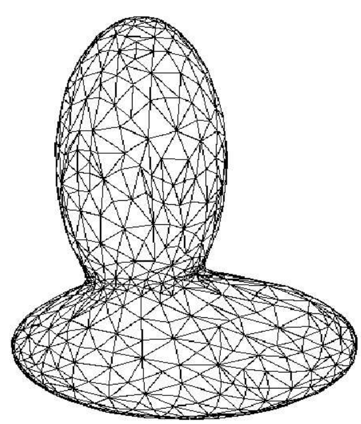

Figure 4: Composition of two ellipsoids

3.1.3 Dealing with sharp edges

Let A and B be two star-shaped 3D objects (with respect to one point H) which have to be composed, but one or both contain some sharp edges. The previous algorithm cannot be applied to such objects because it supposes that the shape of the composed object C is smooth. Now, however C contains sharp edges corresponding to the projections (defined in previous section) of sharp edges of objects A and B.

To solve this problem, the surface of C will be considered as a collection of smooth surfaces delimited by its sharp edges, and the previous algorithm will be applied to these smooth surfaces. The sharp edges of C

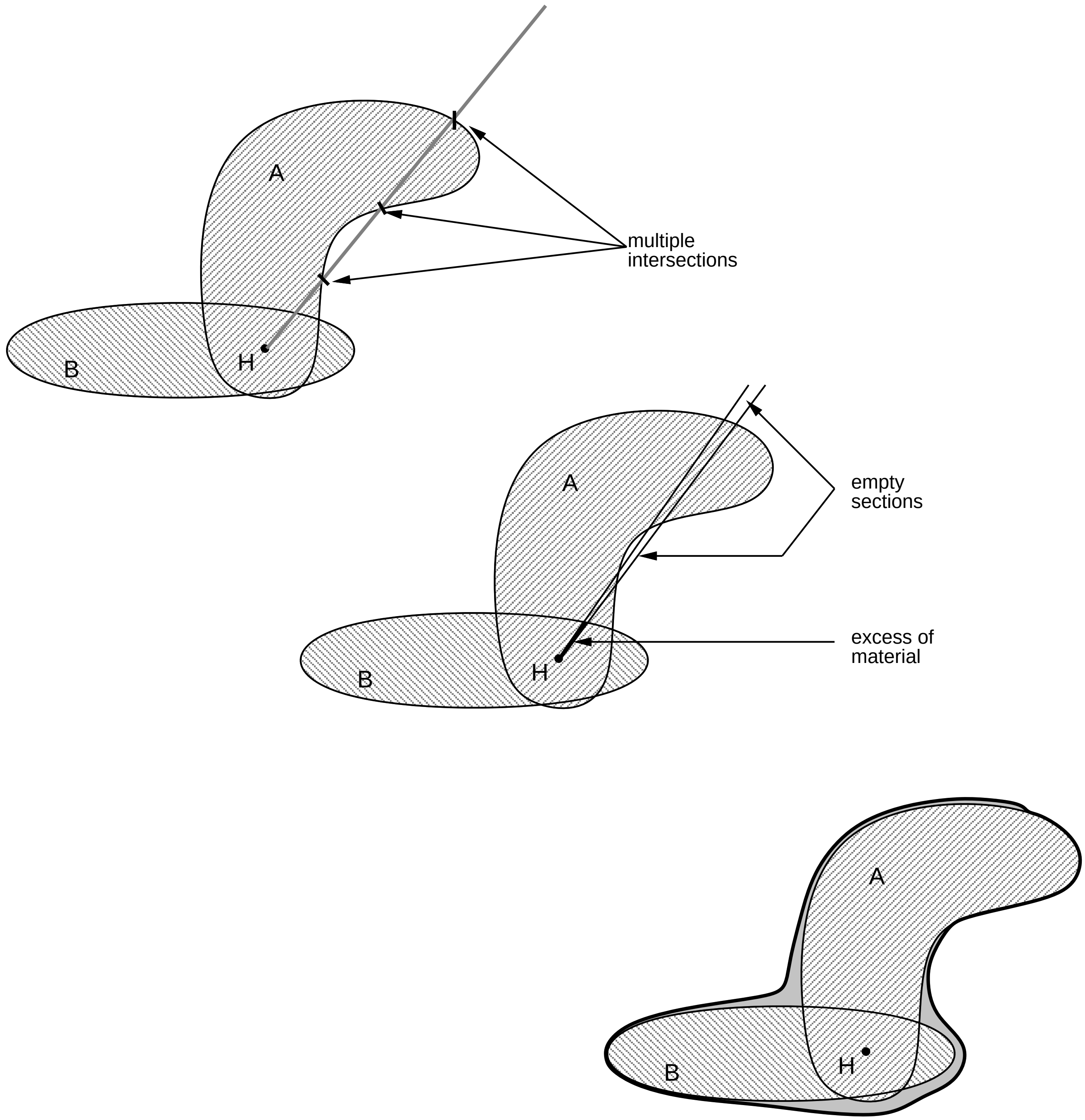

Figure 7: Composing any object with a star-shaped object

model of the shape of the composed object C. The solution proposed here preserves the global volume only if $A \cap B$ has got only one connected component.

As above, H is a fixed point in $A \cap B$. As A is not necessarily convex or star-shaped with respect to H, $A \cap B$ could be a non connex region. Thus the region considered as "the excess of material" corresponds to the connected component of $A \cap B$ including the point H. Now, let us consider an angular sector starting from H. As B is star-shaped, this sector intersects the boundary of B only once; but it may intersect the boundary of A several times (see figure 7). In fact, the angular sector is composed of sections included in $A \cap B$, A, B, or $\overline{A \cap B}$ ("empty sections"). The solution we propose is to move material included in $A \cap B$, into the first empty section. If this section is not large enough to enclose all this material, the remainder is moved to the next empty section, and so on.

To build a model of the shape of the composed object C, we will initialize C as a copy A' of the shape model of A and then deform it to approximate the shape of C. This technique is described as follows:

- Project the vertex/edge/facet network of B on A' and merge the networks. This step is the same as in section 3.1.3.
- Deform the model of A' by projecting *some* of its vertices onto the object C. In fact, vertices of A' are dispatched in three classes. We will describe the three classes and for each one define how vertices are deformed. Let M be a vertex of A'; consider the half straight line [HM); let J be the intersection point between [HM) and the shape of B; let P be the point corresponding to the end of the moved material along an angular sector defined by [HM); let I be, the one of the intersection points between [HM) and the shape of A which is the closest to P and inside the segment [HP].

  M′ will refer to the deformation of point M.

  - If $\left\|\vec{HM}\right\| > \left\|\vec{HP}\right\|$, then M′=M (M corresponds to a point of the shape of A' which does not need to be deformed).
  - If $\left\|\vec{HM}\right\| < \left\|\vec{HI}\right\|$, then M has to be removed from the model of the shape of A' (M corresponds to a point that cannot be projected onto the shape of C). Figure 8 shows a way to remove a vertex from a network having only triangular facets.
  - Else M′=P (M is projected onto the shape of C).
- Refine the resulting network with the same method as in sections 3.1.2 and 3.1.3.

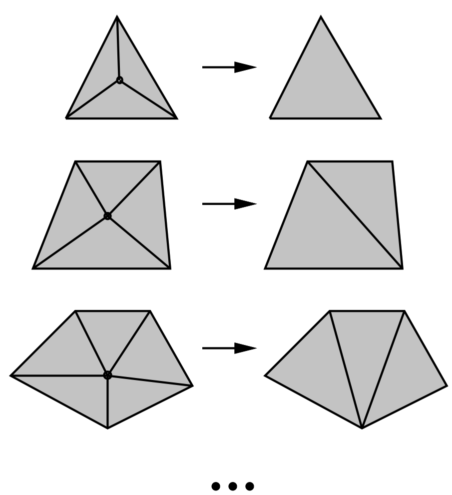

Figure 8: Remove a vertex from a network having only triangular facets

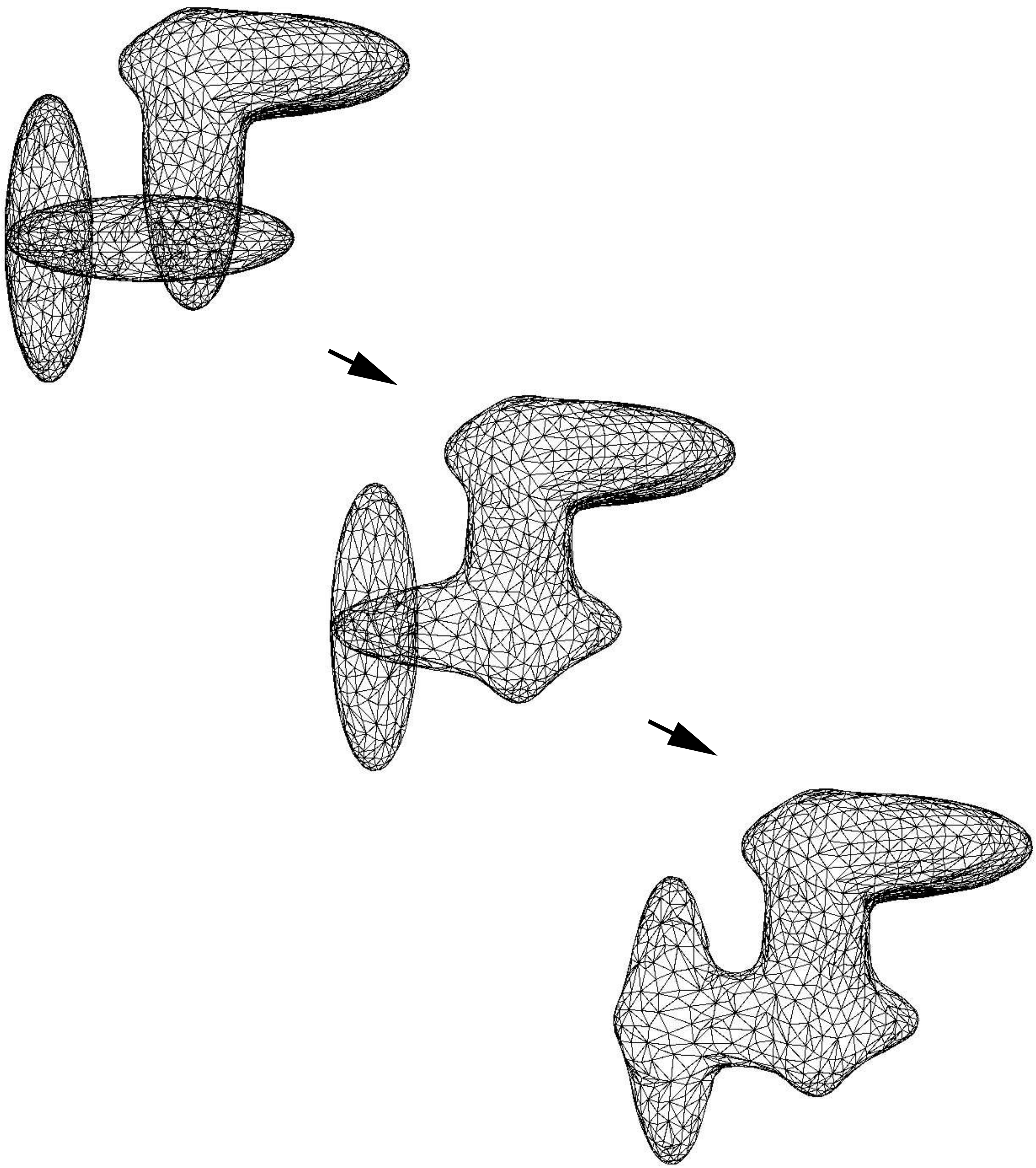

Figure 9: Compositions of ellipsoids

With this algorithm, we are able to compose an object with a star-shaped object with respect to the center of composition. The resulting object, which can be of any shape, can be composed with another convex or star-shaped object. By iterating the process, we are able to build complex objects (see figure 9 to see a complex object built from a few compositions of ellipsoids).

3.3 The center of composition

To compose two objects, a point H is used as the center of composition. We will now see how to choose this point.

This point must verify the following constraint: it is in $A \cap B$ (because it will be used to move the excess of material included in $A \cap B$).

If $A \cap B$ is convex, any point included in $A \cap B$ can be chosen, but the center of mass of $A \cap B$ is the most natural choice, and it can be automatically computed.

If A and B are star-shaped with respect to a single point, this point is a good candidate for the center of composition (then the algorithm that composes star-shaped objects can be directly used).

In other cases, any point in $A \cap B$ can be chosen, but there is no easy way to determine automatically a point; the simplest solution is to let the user choose it.

4 Applications

This section presents different ways to implement the composition technique inside a modeler. Three modeling and animating tools will be described : the first one creates objects, the second one deforms objects and animates deformed objects, finally the third one is a morphing tool which allows a kind of metamorphosis from one object to another.

4.1 A creation tool

This tool is the direct application of the fusion technique. The user selects two objects and a point. The tool verifies that the two objects can be composed with respect to the selected point. It then computes a new vertex/edge/facet network corresponding to the composition of the two objects. The result is a polygonal (in 2D) or polyhedral (in 3D) object on which all tools working on polygonal or polyhedral objects can be applied (especially the fusion tool itself).

4.2 An animated deformation tool

The composition technique can be also utilized as a deformation tool. By simply animating the deformation tool, we produce an animated deformation. The user first selects an object A that he/she wants to deform, and then selects a deformation tool defined by a point H and an object B with a simple shape (an ellipsoid, a cube, ...). Applying the deformation tool to the object A involves composing A with B with respect to the point H and replacing A by the result of the composition.

This deformation tool is intuitive for the user : it adds the material of B to the object A. By changing the position of the point H into $A \cap B$, the user can control the direction along which the material of B will be transported.

Interesting effects can be obtained by deforming an object with an other object which contains sharp edges. For example, figure 5 is seen as a sphere deformed by a tetrahedron.

Through a minimal modification of the composition technique, the user can also subtract the material of B from the object A. Presently the composition technique adds areas or volumes along angular sectors ($\Delta v = \Delta v_1 + \Delta v_2$, see section 3). If we now subtract the volume due to B from the volume due to A ($\Delta v = \Delta v_1 - \Delta v_2$)[5] we obtain a composed object which is similar to A, but from which a volume of material corresponding to the object B has been removed.

An object created or deformed by the composition technique can easily be animated by modifying the objects which compose it. We will illustrate this point with a simple example: simulating an arm (figure 10).

[5] B has to be included in A.

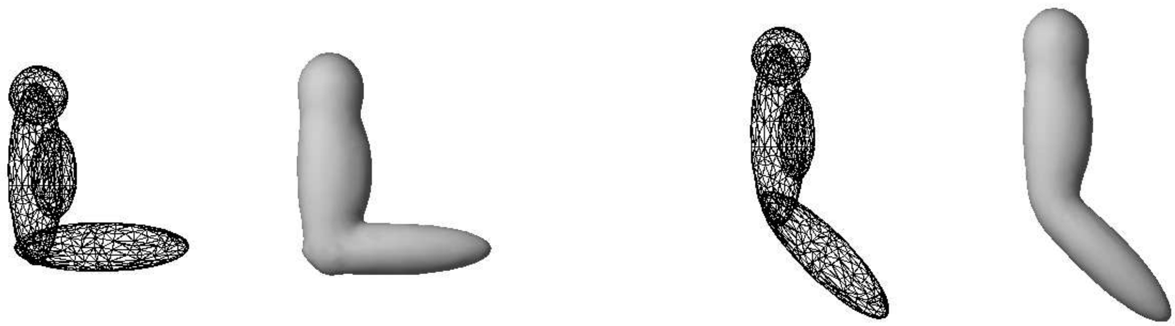

Figure 10: An animated arm

The arm is created by composing some basic objects representing the shoulder, the arm, the biceps and the forearm. The animation is defined by the following motion: the forearm is rotated with respect to a point corresponding to the elbow; the biceps is scaled when the forearm moves. At each step of the animation the composition is computed. This technique is an intuitive way to define an animation of organic deformable structures.

4.3 A morphing tool

In this section, we show how to adapt our technique to produce a morphing tool. Our aim is to produce the metamorphosis from an object A to an object B.

An object $C(t)$ (C depends on a parameter $t \in [0,1]$ which corresponds to the time) has to be created, so that:

- C(0) = A
- C(1) = B
- C(t) is continuous, i.e. the metamorphosis from A to B is continuous.

Methods such as interpolating the two models using the Minkowski sum ([9]) or using a merged topology ([10]) are based upon the same idea as the one we will use here: object A and B are composed at each step of the transformation, but volume of A is decreased from 100% to 0% and the volume of B is increased from 0% to 100%.

If A and B have simple shapes (convex or star-shaped), $C(t)$ can be computed by the following formula:

$$C(t) = comp_H(h_H(\sqrt[3]{1-t}, A), h_H(\sqrt[3]{t}, B)),$$

where H is a fixed point in $A \cap B$; $h_H(\alpha, F)$ represents the similarity of factor α of an object F with respect to point H ; $comp_H(F, G)$ represents the composition of objects F and G with respect to point H.

Let $volume(F)$ represent the volume of an object F. $volume(h_H(\alpha, F)) = \alpha^3 \times volume(F)$, and $volume(comp_H(F,G)) = volume(F) + volume(G)$ because the composition conserves the material. Therefore,

$$volume(C(t)) = (1-t) \times volume(A) + t \times volume(B)$$

This implies the volume of C varies linearly from the volume of A to the volume of B. This is an interesting feature for a morphing tool.

We used the morphing tool to transform a cube into a cross (figure 11). The cross was created by fusing two ellipsoids.

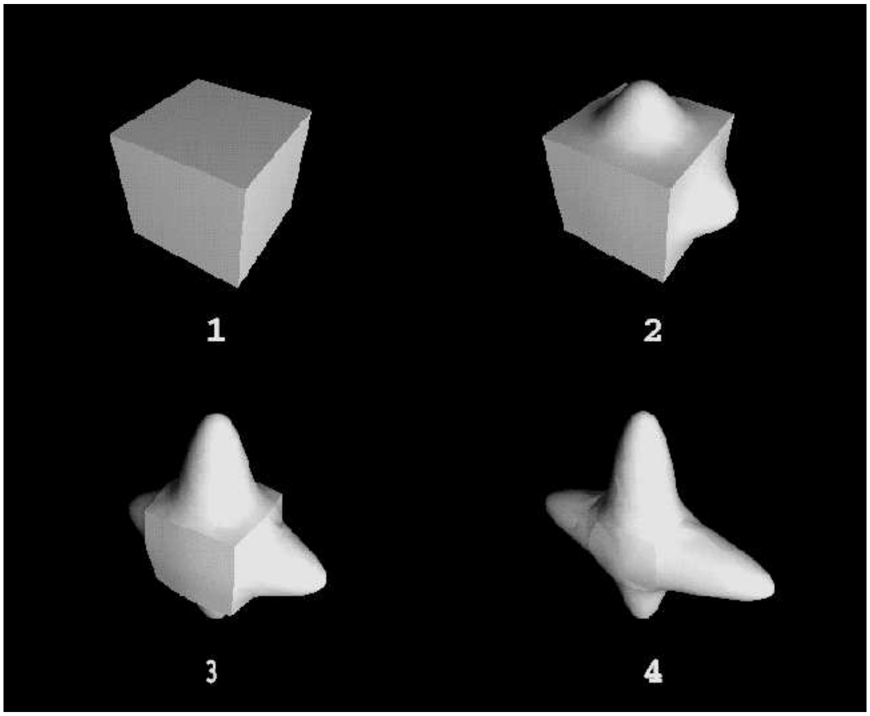

Figure 11: A cube becoming a cross

5 Conclusion

Our fusion technique of two shapes A and B is based upon a material transport. An excess of volume (corresponding to $A \cap B$) is moved using a radial projection. To determine this projection, for each vertex of C (the result of the fusion of A and B), intersections between an half straight line and shapes A and B are evaluated (see 3.1.2). These evaluations need a lot of CPU-time. As an example, the computation of shape of figure 9 takes about 1.5 minute on an Indigo workstation. A fusion cannot be computed in real-time. We are working on overcoming this drawback.

The method allows to compose objects with a star-shaped object and to animate the result. The radial projection is adapted to the star-shaped object. By using new kinds of projections adapted to other kinds of shapes (generalized cylinders, torus, ...), we are studying the composition of general shapes. New effects (deformations, animations) will be obtained.

References

[1] D.R. Forsey, R.H. Bartels. Hierarchical B-Spline Refinement. In *Proc. Computer Graphics, SIGGRAPH'88*, volume 22, pages 205–212. ACM, August 1988.

[2] P. Borrel, D. Bechmann. Deformation of n-Dimensional Objects. In *Symposium on Solid Modeling Foundations and CAD/CAM Applications*, pages 351–369, June 1991.

[3] B.S. Cobb. *Design of Sculptured Surfaces Using the B-Spline Representation.* PhD thesis, University of Utah, June 1984.

[4] S. Coquillart. EFFD: A Sculpturing Tool for 3D Geometric Modeling. In *Proc. Computer Graphics, SIGGRAPH'90*, volume 24, pages 187–196, Dallas, USA, August 1990.

[5] T.W. Sederberg, E. Greenwood. A Physically Based Approach to 2D Shape Blending. In *Proc. Computer Graphics, SIGGRAPH'92*, volume 26, pages 25–34, july 1992.

[6] C. Hoffmann and J. Hopcroft. The potential method for blending surfaces and corners. In G. Farin, editor, *Geometric Modeling: Algorithms and New Trends*, pages 347–365. SIAM, Philadelphia, 1987.

[7] J.F. Hughes. Scheduled Fourier Volume Morphing. In *Proc. Computer Graphics, SIGGRAPH'92*, volume 26, pages 43–46, July 1992.

[8] S. Coquillart, P. Jancène. Animated free-form deformation: An interactive animation technique. In *Proc. Computer Graphics, SIGGRAPH'91*, volume 25, pages 23–26, July 1991.

[9] A. Kaul and J. Rossignac. Solid-interpolating deformations: Construction and animation of PIPs. In *Proc. Eurographics'91*, pages 493–505, September 1991.

[10] J.R. Kent, W.E. Carlson, R.E. Parent. Shape Transformation for Polyhedral Objects. In *Computer Graphics, SIGGRAPH'92*, volume 26, pages 47–54, July 1992.

[11] T.W. Sederberg, S.R. Parry. Free-Form Deformation of Solid Geometric Models. In *Proc. Computer Graphics, SIGGRAPH'86*, volume 20, pages 151–160, August 1986.

[12] R. Szeliski, D. Tonnesen. Surface Modeling with Oriented Particle Systems. In *Proc. Computer Graphics, SIGGRAPH'92*, volume 26, pages 185–194, july 1992.

[13] B. Wyvill, C. McPheeters, G. Wyvill. Animating soft objects. In *The Visual Computer*, volume 2, pages 235–242, 1986.

[14] B. Wyvill, J. Bloomenthal, T. Beier, J. Blinn, A. Rockwood, G. Wyvill. Modeling and Animating with Implicit Surfaces. In *Siggraph Course Notes*, volume 23, 1990.

[15] G. Wyvill, C. McPheeters, B. Wyvill. Data Structure for Soft Objects. In *The Visual Computer*, volume 2, pages 235–242, Springer Verlag 1986.