

Shape-driven deformations of functionally defined heterogeneous volumetric objects

Schmitt B.*
LaBRI,
Bordeaux I University
Bordeaux, France

Pasko A.†
Department of Digital Media,
Hosei University,
Tokyo, Japan

Schlick C.‡
LaBRI,
Bordeaux I University
Bordeaux, France

Abstract

In this paper, we propose a framework for deforming heterogeneous volumetric objects defined as point sets with attributes. In contrast to homogeneous volumes with uniform distribution of material and other properties, a heterogeneous volumetric object has a number of attributes assigned at each point. An attribute is a mathematical model of an object property of arbitrary nature (material, photometric, physical, statistical etc.). It is not necessary for an attribute to be described by a continuous function.

In our approach, the function representation (FRep) is used as the basic model for both object geometry and attributes represented independently using real-valued scalar functions of point coordinates. While FRep directly defines object geometry, for an attribute it specifies a space partition used to define the attribute function.

Deformation of an existing object is usually considered as the final step of the modelling process. In this paper, we propose to define a new node in the FRep tree based on shape-driven deformations. These deformations can be controlled by additional shapes (points, curves, surfaces, or solids) and can be applied to object geometry and attributes at any modelling step.

CR Categories: I.3.5 [Computational Geometry and Object Modeling]: Object representation, Hierarchy and geometric transformations —;

Keywords: Deformation, Implicit Surface, Function Representation, Heterogeneous Objects, Constructive Hypervolumes

1 Introduction

Volume modelling has become an important research topic. In contrast to homogeneous volumes with uniform distribution of properties, a heterogeneous volumetric object has a number of attributes assigned at each point and varying in 3D space. It is not necessary for an attribute to be continuous. Heterogeneous volumetric objects can be modelled as 3D point sets with non-uniform distribution of attributes of an arbitrary nature (photometric, material, physical,

statistical, etc.). Heterogeneous objects are considered in such different areas as CAD/CAM and rapid prototyping of objects with multiple materials and varying material distribution, representing results of physical simulations, geological and medical modelling, volume modelling and rendering.

Different models based on the boundary representation [Kumar and Dutta 1997; Kumar et al. 1999], volumetric [Chen and Tucker 2000; Park et al. 2001; Martin and Cohen 2001], and function-based constructive techniques [Pasko et al. 2001] have been proposed for heterogeneous objects. As it was shown in the mentioned publications, real-valued functions serve well for modelling both geometry and attributes. We follow in this paper a generalised constructive hypervolume model [Pasko et al. 2001; Pasko et al. 2002], which supports uniform constructive modelling of point set geometry and attributes using real-valued functions of several variables.

We propose in this paper to define deformations for both geometry and attributes of heterogeneous objects, based on the constructive hypervolume model. We present in the following subsections existing deformation techniques, the constructive hypervolume model and then the motivation of our work.

1.1 Previous works

One of the first deformation scheme, called *warping*, was proposed in [Parent 1988]. Given a polygonal surface, a vertex is selected, and moved towards the outside of the object. Neighbour vertices of the mesh are then displaced according to a distribution function depending on their distance to the displaced vertex. One drawback of this method is that it can not be used to define global deformations.

Another technique, called FFD [Sederberg and Parry 1986; Coquillart 1988] which stands for *Free-Form Deformation*, embeds the object to be deformed into a rectangular volume defined using a control point lattice. Then, while moving the control points, the embedding volume and the embedded object are deformed. Several extensions have followed, namely EFFD for *Extended Free-Form Deformation* [Coquillart 1990], where the embedding volume is replaced by some more complex one, or the RFFD (*Rational Free-Form Deformation*), where another degree of freedom is provided while adding a weight factor to the control points of the lattice. One of the last extensions of the FFD model is presented in [MacCracken and Joy 1996], where a subdivision volume is used to embed the object.

Several other deformation techniques exist. For instance, instead of embedding the object into a volume, an axis can be defined in the object. Then, as one deforms the axis, the object deformation follows. The axis can be a broken line [Parent 1988; Lazarus et al. 1994], a Bézier curve [Chang and Rockwood 1994] or even a Bézier surface [Mikita 1996]. Other approaches to deformation include the "simple constrained deformation" using ellipsoids *Scodef* [Borrel and Bechmann 1991; Borrel and Rappoport 1994; Bechmann 1994], and its extension with generalised meatballs [Jin et al. 1998], the "implicit free-form deformation" *IFFD* [Crespin 1998], and the "Wires" model [Singh and Fiume 1998].

*e-mail: schmitt@labri.fr

†e-mail:pasko@k.hosei.ac.jp

‡e-mail:schlick@labri.fr

Control of 2D image deformation using feature shapes (points, segments) was described in [Beier and Neely 1992]. Similar approaches were proposed independently for controlling 3D deformations [Savchenko et al. 1995; Ruprecht and Mueller 1995] and 3D metamorphosis of homogeneous volumes [Lerios et al. 1995].

1.2 Constructive hypervolume model

In this paper, we choose to use the constructive hypervolume model for representing heterogeneous objects. A point set with attributes can be considered as a hypervolume object [Pasko et al. 2001; Pasko et al. 2002], and expressed as the tuple:

$$o = (G, A_1, \dots, A_k) : (G, S_1(X), \dots, S_k(X)) \quad (1)$$

where :

- $X = (X_1, \dots, X_n)$ is a point of the Euclidean space E^n
- G is a point set.
- S_i is a real-valued scalar function representing an attribute A_i

In 3D, the point set G can be defined using any existing representational schemes for solids: BRep, CSG, spatial partitioning, generative models, ray implementation, and others.

In this paper, the point set G and the attribute functions S_i are defined by real-valued functions using FRep [Pasko et al. 1995]. The function F corresponding to an FRep object is at least C^0 continuous and defined in the Euclidean space E^n . F is called a *defining function*. The object is then defined by the inequality $F(X) \geq 0$, since this inequality defines three closed sub-sets in E^n , a point set where F is positive, another where F is equal to zero, and a last one where F is negative. In the 3D case, the equality $F = 0$ is called in the literature *implicit surface*. The function F is evaluated by a procedure traversing a tree structure, where primitives are placed at the leaves, and operations at the nodes. Because the function F is built using a constructive method, the hypervolume object defined in eq. 1 can be called *constructive hypervolume*. For a similar reason, to the F and S_i functions correspond trees, which can be called respectively *constructive geometric tree* and *constructive attribute trees*.

An attribute can be defined as a mathematical model of an object property of arbitrary nature defined at any point of the point set. For example, to model a mechanical part with varying internal material distribution one can introduce a three-dimensional solid as a point set and a real-valued scalar function to represent material density as an attribute. In this paper, we consider attributes as the photometric properties of an object. It includes colour, ambient, diffuse, specular and reflectance property. In this case, a special technique for texturing objects has been proposed [Pasko et al. 2002], called *constructive solid texturing*. This method consists in defining a space partition using constructive trees, and in each subset, different attributes are defined, i.e., solid textures [Peachey 1985; Perlin 1985].

Figure 1 is given as an illustration. The constructive geometric tree is composed of two twisted blocks, blended with a cylinder (Fig. 1a). The space partition is shown in Fig. 1b. It consists of three subsets. The first one is defined as a union of four swept spirals; the second of a union of three tori; and the last one is the remaining space. Then, for each given point, a tree traversing procedure is applied to the geometric tree to determine whether or not the point belongs to the object. In the affirmative case, another tree traversing procedures are applied to determine which subset of the space partition the point belongs to. Depending on the result, different colours (i.e., the attributes in this example) are then calculated. If the point belongs to the swept spirals, the result is a constant blue colour, or if it belongs to the tori, a constant grey colour is returned. If none of those cases happen, then a colour based on the Perlin noise function is generated, as well as the opacity attribute. The result is shown in Fig. 1c.

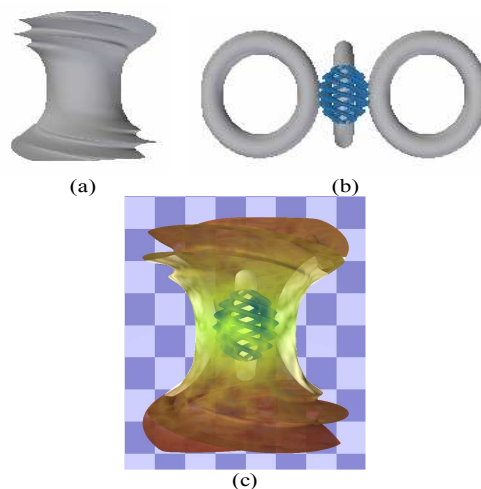


Figure 1: Example of a constructive hypervolume. (a) The geometrical tree composed of two twisted blocks blended to a cylinder. (b) The space partition defined as three tori and four swept spirals. (c) The resulting textured object.

1.3 Contribution

Most of the techniques for deformations presented in the previous works subsection can not be directly used in the constructive approach. Indeed, those techniques are applicable only to a polygonal surface, as they are defined using a forward mapping. Even if some deformation tools deform the whole space using a function from $\mathcal{R}^3 \rightarrow \mathcal{R}^3$, such as in [Crespin 1998; Bechmann 1994], they are applied directly to the vertices of a polygonal surface.

In [Schmitt et al. 1999], the constructive approach and the volume sculpting approach were combined using the unifying FRep model. In this paper, we propose to define a new node for the FRep tree for deformations. One important goal is to provide the possibility for the user to model an object without the traditional separation between the constructive approach, the sculpting process and the deformation steps. Usually, the modelling scheme is as follows: first one models an object in some way, and then uses deformation tools to obtain the desired shape. As most of the existing tools for deformations are using forward mapping, once the object is deformed, one can hardly return to the modelling step, and combine the existing object with some other.

In order to be able to switch from the modelling step to the deformation step whenever it is needed, one solution is to define the deformation using inverse mapping, and thus to provide a new node for the FRep tree. To calculate the inverse mapping from the forward mapping of the previous work is a very difficult operation, and is even sometimes impossible. A preferable solution is to define a new node from the scratch. Of course, several similarities can be found with the previous works, and we used general ideas for defining this new node. But one has to keep in mind that the definition we propose is based on inverse mapping, and thus, even if the visual result of the deformation is close to existing deformations, the underlying idea is different.

Deformation nodes already exist in the FRep tree. One can easily twist, taper or stretch an object along an axis. These deformations are based on the work presented in [Barr 1984], but the set of available deformations is quite limited. A general framework for deformations in the FRep model has been proposed in [Savchenko and Pasko 1998], where the so-called *extended space mapping* is defined. In [Savchenko et al. 1995], deformations were defined using point-controlled space mapping, but it had a too global character due to the interpolation with radial-basis functions.

We propose in the following to define new deformation tools to deform an object, either locally or globally. Let us first consider a simple translation, as in [Savchenko et al. 1995; Savchenko and Pasko 1998]. Let f be a defining function of some geometric object, and T a translation vector defined as (dx, dy) . Then, given a defining function f of a 2D object, the inverse mapping for this transformation is defined as:

$$T : f(x, y) \rightarrow f(x - dx, y - dy) \quad (2)$$

This operation is globally applied to the entire object, as dx, dy are constants. Now, let us define the deformation by non-linear space mapping. Consider the same 2D object and the displacement of a point A towards a point A' . To define a local deformation centred on the point A , one has to respect the two following requirements:

- (dx, dy) have maximum values at A' .
- (dx, dy) drop uniformly to zero when (x, y) is far from A' .

The behaviour of the deformation depends on the calculation of (dx, dy) . In the remaining of this paper, we present different approaches that respect those requirements. The first set of deformations, proposed in the next section (2), contains point to point based deformations. Traditional field functions, i.e., a combination of potential and distance functions, are used to define new deformation tools. The section 3 proposes another set of deformations, and in some senses, extends the point to point based deformations. This set offers the possibility to define an area of influence for the deformation, as well as a target area. Several examples are given to illustrate our approach, and in the last section (4), complex examples are given.

2 Deformations using field functions

2.1 Potential functions

We propose in this section to define a set of deformation tools based on field functions. The analogy of the above requirements with blobby objects, soft objects, metaballs and others [Blinn 1982; Blanc and Schlick 1995] is straightforward. Given a skeleton point, an object is defined for every given points in the space as a composition of a distance function γ with a potential function f . The resulting function, $f \circ \gamma$ is a real-valued function in $[0, 1]$. When the given point X is located on the skeleton point, the resulting real value is equal to one, and it drops to zero as X is far from this skeleton point. On the base of these properties, we can formulate an inverse mapping for a deformation as follows:

$$\begin{cases} X' = X - \Delta(X) \\ \Delta(X) = f(\gamma(X))(A' - A) \end{cases} \quad (3)$$

where X is an input point of the Euclidean space, and X' its image after applying the inverse mapping, A and A' are the source and the target points of the deformation. By comparison, the point A is the skeleton point, and as one can see, the deformation is maximal when the given point X is located on A , and equal to $A - A$.

The choice of the potential function and of the distance function is important. Several authors studied the definition and the behaviour of potential functions. A good survey can be found in [Blanc and Schlick 1995]; and a potential function is also defined based on piecewise rational polynomial. This function has the benefits to have a low computational cost and to include a hardness factor. This function is defined as follows:

$$f(\gamma) = \begin{cases} 1 - \frac{(3\gamma^2)^2}{p + (4.5 - 4p)\gamma^2} & \text{if } 0.25 \leq \gamma^2 < 1 \\ \frac{(1 - \gamma^2)^2}{0.75 - p + (1.5 + 4p)\gamma^2} & \text{if } 0 \leq \gamma^2 < 0.25 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

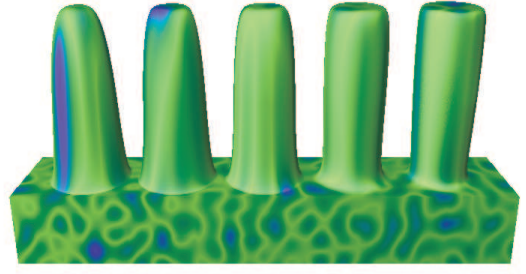


Figure 2: Deformation of a block by space mapping using the potential function 4. Five different deformations using the same displacement along the vertical axis with five different hardness factors.

where γ is a distance function. Another valuable property of this potential function is it is based on a squared distance function, and thus avoids the use of a square root. The use of the hardness factor is intuitive, and the application of this function as a deformation tool is straightforward, as Fig. 2 shows, where a block is deformed using the above potential function. For a similar displacement along the vertical axis, different hardness factors have been used. The distance function used for this example is similar to the one defined in eq. 6. More details on the distance functions can be found in the next section.

2.2 B-spline based deformations

The set of deformations generated by the potential function proposed in [Blanc and Schlick 1995] and other may not offer a sufficient variety. Thus, we propose here to use another distance function based on a parametric curve [Nishita and Nakamae 1994; Kasic-Alesic and Wyvill 1991; Blanc and Schlick 1995]. We choose the cubic B-spline functions as illustrations. Any other interpolation function can also be used. As it was previously mentioned, the main requirement to a potential function is to drop to zero given a certain radius of influence.

Let us consider a B-spline function of one variable $f(t)$, defined by a set of n control points $\{P_i\}$. The parameter t belongs to the parametric space $[0, 1]$. Control points are defined in a 2D space. While the first coordinate is used to locate the point in the space, i.e., regularly placed along the x -axis to insure that $f^x(t) = t$, the second coordinate is a scalar coefficient. We also assume that outside this domain, the B-spline function remains equal to zero (see the book [Farin 1990] for more details). In the following, the B-spline function will be used as a potential function, and the distance function will define the parameter t , i.e., $t = \gamma^2(X)$. Then, by changing the scalar values of the control points, different shapes, and thus, different deformations, can be obtained using the B-spline function interpolation.

Let us consider two points A and A' , as respectively the source and the target point, and a distance function γ . Then, for every given point X , one can evaluate the distance between X and A' , $\gamma(X)$. When X is placed at A' , $\gamma(x)$ is equal to zero, the B-spline function is equal to $f(0)$, i.e., equal to the scalar value of the control point P_0 . When X is far from A and the distance $\gamma^2(X)$ increases, as it was stated in the definition of the B-spline function, this function will drop to zero when the distance function is greater than 1.

To define deformations using the B-spline potential function is simple. An additional step is needed as one needs to model the shape of the B-spline function, i.e., to change the scalar coefficient of the control points. In the following, this shape will be called *profile*. Once this profile is defined, the application of the space mapping is straightforward. Previously, we formulated the requirements that first the displacement has to be maximum for the target point, and drops uniformly to zero as the given point is far from the

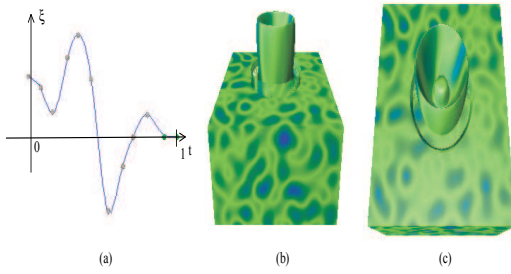


Figure 3: Block deformed using a B-spline based potential function. (a) Profile of the B-spline function. (b,c) Different views of the deformed object.

target point. This statement is still exact, but may be too restrictive in the case of the deformation based on B-spline function. Consider for instance Fig. 3a, where a profile has been designed. The scalar value of the control point P_0 is not the greatest one, and even negative values have been assigned to some other control points. As one can see in Figs. 3b and c, where different views of the resulting object is shown, the deformation follows the profile. So the requirements of a deformation are first the deformation has to be centred along the displacement vector, and the displacement should reach zero at some given distance with no restriction of the monotone decreasing.

2.3 Distance functions

Once the potential function is chosen, (eq. 4 is a good candidate), one can use different distance functions. Indeed, as the potential function is a function of one variable, it depends only on the distance value, and even if different values for the hardness factor produce different deformations, the set of available deformations is quite limited. The simplest definition for the distance function is:

$$\gamma^2 = \frac{(x - x_{A'})^2 + (y - y_{A'})^2 + (z - z_{A'})^2}{r^2} \quad (5)$$

where A' is the target point, and r is the radius of influence. This distance function is the traditional Euclidean distance function, also known as the *spherical distance*. Figure 4 shows in greyscale the area of influence around the target point. Black colour corresponds to $\gamma = 0$, and white colour to $\gamma \geq 1$. Due to the definition of the distance, it is clear that the deformation is symmetric along each axis. Furthermore, the displacement of the source and the target point is limited, depending on the radius of influence r . In Fig. 4a, the distance between the source and the target point is lower than r ; the deformed part remains connected to the object. When the point is moved too far from the source, a disconnected part appears. This case is illustrated in Fig. 4b.

A simple solution to insure that no disconnected part will appear is to normalize the definition of γ depending on each axis, and thus change its definition from a spherical distance function to an *ellipsoidal distance* function:

$$\gamma^2 = \frac{1}{(r_x)^2} \frac{(x - x_{A'})^2}{1 + (x_{A'} - x_A)^2} + \frac{1}{(r_y)^2} \frac{(y - y_{A'})^2}{1 + (y_{A'} - y_A)^2} + \frac{1}{(r_z)^2} \frac{(z - z_{A'})^2}{1 + (z_{A'} - z_A)^2} \quad (6)$$

where r_x, r_y and r_z are three parameters which define the area of influence along each axis, and can be called radii of influence. Figure 4c shows the result while applying this new definition of gamma. The same displacement of the target point has been used for both Figs. 4b and c, without changing the radii of influence.

One natural solution extension is to use the generalisation of the Euclidean distance function, known as the D^n distance. This extension offers only one additional degree of freedom. Furthermore,

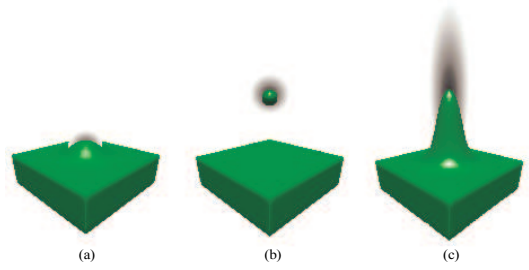


Figure 4: Distance function. A point is moved towards the outside of the object, along an axis. The grey area shows the influence around the target point. Black colour corresponds to distances equal to 0, and white colour to distances greater or equal to 1. (a) Spherical distance. The displacement is small; the displaced material is still linked to the object. (b) Spherical distance. The target point is moved too far from the object. (c) Ellipsoidal distance. Area of influence now depends on the distance of the given point to the source point and to the target point.

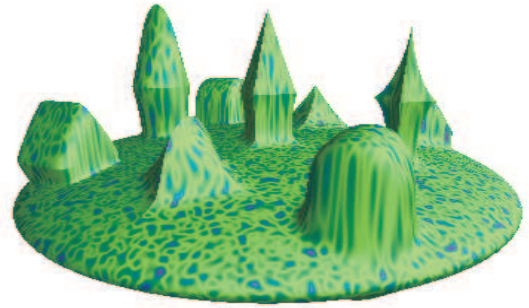


Figure 5: Ellipsoid deformed using the superellipsoidal distance function. Nine target points have been used.

if one uses the potential function defined in eq. 4, the behaviour of the deformation depending on either the hardness factor or the power n of the distance function is redundant. A more intriguing solution is to use the *superquadric distance* functions. The direct application is shown in Fig. 5, where several superellipsoids are used to deform an ellipsoid. Notice the texture pattern follows the deformation, as the deformations also occur in the attribute trees. Extension to quadrics and other functions can easily be deduced.

3 Shape driven deformation

In the previous section, we defined an inverse mapping for point to point based deformations, where a source point was moved towards a target point. This case is the simplest, and we propose to use it as a framework for the following extensions. First, we will define a general deformation technique, and consider particular applications of this general framework.

3.1 Framework for deformations

Given a source point A and a target point A' , we also consider two special areas corresponding to an area of influence and an area of projection. The area of influence is defined by a real-valued function Z , and takes value in the interval $[0, 1]$. The area of projection is defined explicitly, and can consist in a line segment, a plane or others. Let us note H , the projection of a given point X on this area. Using the same notation as in the previous section, we propose to

define a general deformation as follows:

$$\begin{cases} X' = X - \Delta(X) \\ \Delta(X) = f(\gamma(X))\tilde{Z}(X)(H - A) \end{cases} \quad (7)$$

The use and the influence of each term of this equation are explained in the following sub-sections. The next sub-section (3.2) illustrates the use of different functions for the area of influence \tilde{Z} , and the sub-section 3.3 considers different areas of projection. We suppose that the potential function and the distance function are already chosen (eqs. 4 and 6 are good candidates).

3.2 Area of influence

In this subsection, our interest is turned towards the function \tilde{Z} defining the area of influence. It can consist in any shapes, such as a block, a cone, or any other FRep object. Let Z be the defining function of this FRep object. Z is a real-valued function, at least C^0 continuous, that takes value in \mathfrak{R} . To insure that the resulting real value is in the interval $[0, 1]$, we propose to use the following mapping function:

$$\tilde{Z}(X) = \frac{1}{2} \left(1 + \frac{Z(X)}{\sqrt{(\vartheta + Z^2(X))}} \right) \quad (8)$$

The variable ϑ is an important feature in the behaviour of the deformation. In some sense, it can be compared to the hardness factor of the potential functions (eq. 4), and controls the blend of the deformation with the initial object. Figure 6 shows a deformation of an ellipsoid with different values of ϑ . The Z function is defined as a simple block, where the vertical axis is the displacement vector. As one can see, as ϑ is small, as the resulting deformation is close to the area of influence with less blend, and as it gets bigger, as the deformation is smoother and blends more with the initial object.

In the next example, Fig. 7, the area of influence is also defined as a block, but a tapering operation has been added. From left to right, one can see the initial deformation, the deformation tapered along one axis, and then along both axes.

Figure 8 shows another deformed ellipsoid, but in this case, Z is defined as a torus. Therefore, the deformation that can be achieved looks like a torus. To emphasise the influence of the parameter ϑ , three different deformations are shown, with different values for ϑ . One important feature is that one can change the topology of the initial object using the proposed deformation method. None of the existing method based on forward mapping can handle this problem.

Despite the topology change, the result is also interesting if one considers the attributes of the object, i.e., the texture in this case. In the torus example and for a given ϑ , the result is similar to a blending union of the initial object and a torus. However one can notice that the texture is stretched along the deformation.

Extensions to arbitrary orientations are straightforward. Considering the displacement vector AA' , one can apply a set of two inverse rotations and a translation to obtain the desired orientation and location. Furthermore, as the area of influence is defined as a FRep object, all the set of available operations and primitives can be combined to define it. These possibilities extend considerably the set of possible deformations.

3.3 Area of projection

The area of projection increases considerably the set of available deformations. We consider in the following two areas of projection, i.e., a plane and a line segment. Furthermore, the proposed examples also consider projections of different natures, respectively a perspective and a parallel projection.

Given a source point A and a target point A' , one can define a cone of height AA' , and a plane orthogonal to AA' and containing

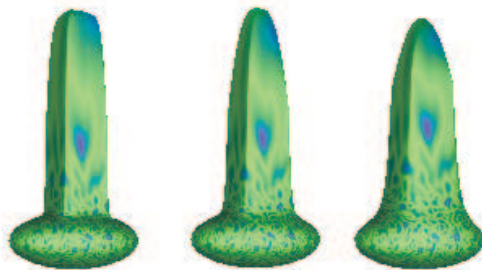


Figure 6: Influence of the ϑ parameter in the \tilde{Z} function. From left to right, $\vartheta < 1$, $\vartheta = 1$ and $\vartheta > 1$.

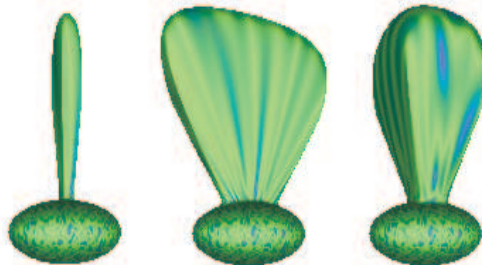


Figure 7: Changing the area of influence. The Z function is defined as a FRep tree, composed of tapering operation and a block primitive. From left to right, the first figure shows the initial deformation, then a tapering operation along one axis, and then along two axes.



Figure 8: Changing the area of influence and ϑ . The Z function is defined as a FRep tree, composed of an intersection of a block and a torus. The parameter varies in a similar way as in 6

A' . The cone is the area of influence of the deformation, and corresponds to the Z function. For each given point X , one can calculate its projection H onto the plane. In this example, we choose a projection similar to a perspective one, where the vanishing point is A , and H is defined as the intersection of the line AX and the plane. The distance function is then calculated depending on H (and not A' as previously). The use of the area of influence takes now its full meaning. Indeed, if Z is not included in the definition of the inverse mapping, an infinite deformation occurs. Figure 9a shows the result of such deformation. The object to be deformed is an ellipsoid. The source point A coincides with its centre, and the displacement of the target point A' is along the vertical axis, z . The area for the projection is defined as the xy plane, translated along the z axis of AA' . The result is an infinite deformation. It naturally comes that to cut the unwanted part of the deformation, one has to define the area of influence Z . Different results can be obtained depending on the choice of Z . Figure 9b illustrates the previous explanations, as we choose a cone with axis AA' for the area of influence, and in Fig. 9c, we choose to replace the cone by a block.

Instead of considering a plane as the projection area, one can also consider a line segment. In the example shown in Fig. 10, the initial configuration for the source and the target points is identical

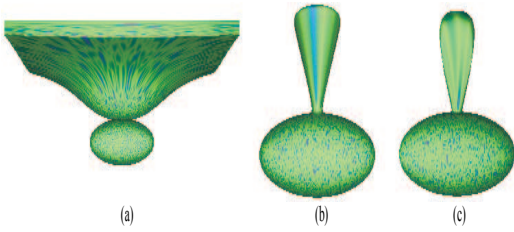


Figure 9: Shape driven deformation of an ellipsoid. Effect of applying the area of influence. The area of projection is a plane. (a) No area of influence is defined, resulting in an infinite deformation. (b,c) The areas of influence are defined respectively using a cone function and a block function.

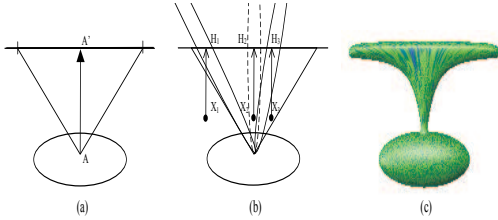


Figure 10: Shape driven deformation of an ellipsoid. Influence of the projection. The area of projection is straight line. (a) Initial configuration. (b) Parallel projection of some points. (c) Resulting deformation.

to the previous example of Fig. 9. The difference comes from the projection area and from the influence area. The projection area is defined as a line segment, and we choose to use a parallel projection to map every point X on it. The area of influence is defined as a convolution triangle. In Fig. 10a, one can see the initial state, where the object to be deformed is an ellipsoid, the source and the target points are positioned, as well as the line for the projection, and the triangle for the area of influence. Figure 10b shows the projection of three different points of the space, X_1 , X_2 and X_3 . The three new ellipsoids in the figure correspond to the distance function (as usual, the internal part of the ellipsoid represents distance to centre below or equal to 1). The point X_1 is mapped onto the projection line, parallel to AA' . Its image is H_1 , but as X_1 is located outside the area of influence, the function $\tilde{Z}(X_1)$ is equal to zero and X_1 will be mapped onto itself. The second point X_2 is mapped onto the line, and its image H_2 corresponds to the centre of the ellipsoidal distance function γ . As $\gamma(X_2)$ is lower than 1, and as X_2 lies inside the area of influence, the point X_2 is mapped to some other location (close to A to be more precise). The third point X_3 emphasises the importance of the choice of the projection. As we mentioned previously, in this example, the projection is a parallel one. The point X_3 lies inside the area of influence. Its image generates an ellipsoidal distance field, but as one can see, the value $\gamma(X_3)$ is greater than one. Thus, the corresponding potential value $f(\gamma(x_3))$ is equal to zero, and the point X_3 will be finally mapped onto itself. Figure 10c shows the resulting deformed object.

4 Examples

The proposed deformation framework can be used in various directions. One interesting method is to use a curve as a path for the deformation. Source points and target points are obtained while sampling regularly the curve. The resulting deformation is expressed in terms of FRep, i.e., successive small deformations correspond to the

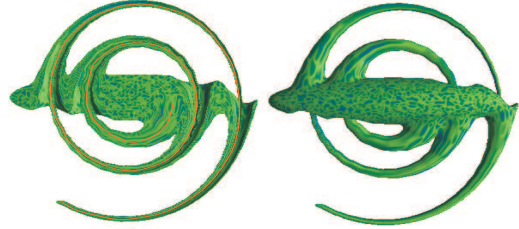


Figure 11: Self intersecting deformations. An ellipsoid is deformed using a spiral curve. Several intersections between the original shape and the deformation occur. Right picture shows the inside of the deformed object.

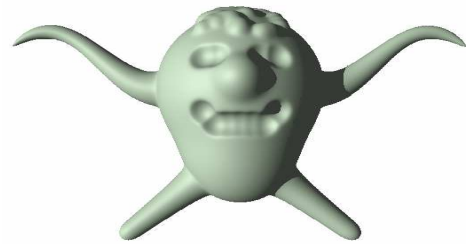


Figure 12: Example of the deformation using space mapping. Eyes, nose, mouth, bumps and bottom horns result from point to point based deformations, and top horns are obtained as sets of deformations along a sampled curve.

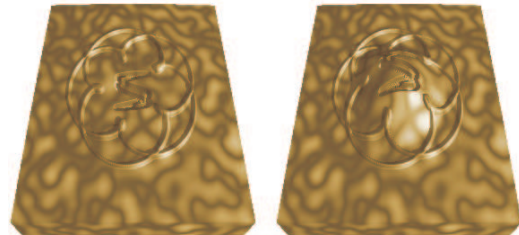


Figure 13: Deformations as carving tool. (Left) A block is carved according to a path defined along a curve. (Right) An additional global deformation is then added, and the carving follows it.

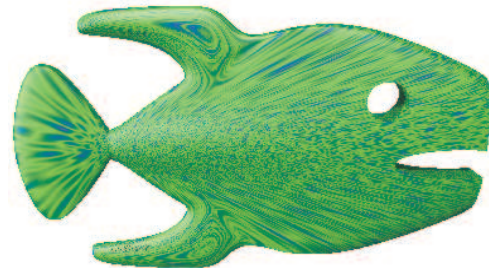


Figure 14: Space mapping as new node in a FRep tree. Several deformations are applied to an ellipsoid and are then combined to a FRep tree (intersection operation with a cylinder and another ellipsoid).

nodes of a FRep tree. One important feature of such tree structure is that the proposed deformation supports self-intersection. Consider for instance an ellipsoid deformed along a spiral, as shown in Fig. 11. The spiral intersects several times the ellipsoid. Figure 11a shows a vertical cut of the ellipsoid, with the spiral shown in red, and Fig. 11b shows the whole object. Here and there of the spiral, holes are created as the neighbouring points are also displaced. Resulting shape is similar to a shell.

Figure 12 provides an example where a single sphere is transformed using several different space mappings. Some parts are deformed using single control points, such as the nose, the eyes, the bumps on the top of the head, and the mouth. The straight horns at the bottom of the head are modelled while using two deformations. The horns at the top of the head are modelled by sampling two B-spline curves. As one can see, deformations in arbitrary directions can be obtained, and several deformation nodes can be combined together. Figure 14 provides another example, where an ellipsoid is deformed using four deformations, and is then combined to a cylinder and another ellipsoid using the set-theoretic intersection.

Figure 13 is given to show that the deformation scheme we propose can be also used to carve object, and even if the most of the examples show major deformation of an object, subtle details can be also defined. On the left part, an object has been carved (according to two parametric functions of Lissajous). On the right, an additional deformation has been applied. As one can see, the carved details also follow the deformation.

Figure 15 shows the combination of a deformed object with another primitive. The central object is a sphere deformed by several space mappings. It is also deformed by a twisting operation. As one can see, the shape driven deformations also follow the twist. A torus is added to this object using a union operation. We choose deliberately to apply only one deformation to the torus, identical to one of the deformations applied to the ellipsoid.

The last example is a vase (Fig. 16). To model this object, we combined the constructive approach with the sculpting and deforming steps. First, the body was created using a B-spline object, and then its top was deformed. The next step was to combine it with an ellipsoid. Once both parts were combined, other deformations are achieved along two curves, such as they get close to the body. The final step was to create the handles of the vase. If the deformation step was the last one as it is usually the case, to find the correct location for the handles may be difficult, but as we built the tree node by node, regardless of the nature of the operation, this task was easy.

5 Conclusion

In this paper, we proposed different techniques for deforming constructive hypervolumes on the base of the FRep model. A first set of deformations is point based, as arbitrary points in the space are moved to arbitrary positions. Traditional field functions are then used to define the deformation. The second technique consists in moving arbitrary points too, but an area of influence for the deformation is defined, as well as a target area. A large number of new deformations can be obtained with this method, and it becomes possible to change easily the topology of the initial shape.

These techniques can be easily applied to traditional implicit objects. Nevertheless, the most intriguing results are achieved while using the constructive hypervolume model. Indeed, if the proposed deformations are applied only to the geometry of the object, similar visual results can be obtained with traditional set-theoretic and blending operations. The justification of the proposed work takes its full meaning when both attributes and geometry are considered. If the deformation node is included in the geometric and attribute trees, the deformation of geometry is followed by the corresponding deformation of attributes. The complex examples that have been given show that the texture intuitively follows the deformation of

the objects geometry.

We choose to define the deformations as a new node in the FRep model. This brings the important feature that it becomes possible to combine several modelling approaches to define objects. The first one was the constructive approach, inherent property of the FRep model, the second one was volume sculpting while using a node based on trivariate B-spline and Bézier functions. In this paper, we proposed a different technique of deforming existing object with the use of different control shapes (points, segments, curves, surfaces, and solids) and corresponding field functions.

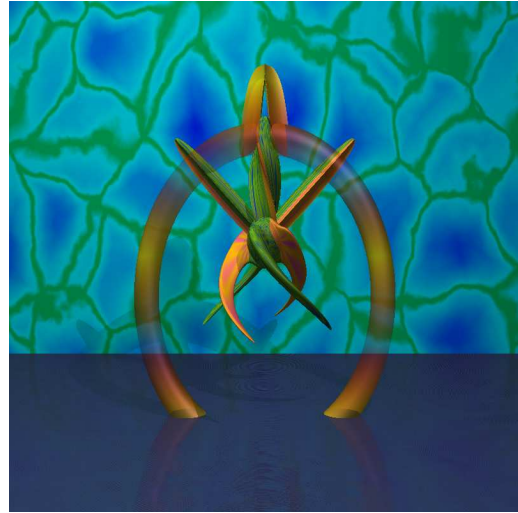


Figure 15: Example of the shape-driven deformation. A sphere is deformed, combined with a torus. Successive deformations are applied at different level of the tree, deforming either the sphere and the torus or only the sphere. A twisting operation is applied to the deformed sphere.



Figure 16: Example of the shape-driven deformation. Constructive modelling and deformation steps were performed in arbitrary order while designing the shape and photometric attributes of this object.

References

- BARR, A. 1984. Global and local deformations of solid primitives. *Proceedings of SIGGRAPH '84, Computer Graphics 18*, 3, 21–30.
- BECHMANN, D. 1994. Space deformation models survey. *Computer and Graphics 18*, 4, 571–586.
- BEIER, T., AND NEELY, S. 1992. Feature-based image metamorphosis. *SIGGRAPH'92 Proceedings, ACM Press*, 35–42.
- BLANC, C., AND SCHLICK, C. 1995. Extended field functions for soft objects. *Implicit Surfaces '95, Eurographics/ACM SIGGRAPH Workshop*, 21–32.
- BLINN, J. 1982. A generalization of algebraic surface drawing. *ACM Transactions on Graphics 1*, 3, 235–256.
- BORREL, P., AND BECHMANN, D. 1991. Deformations of n -dimensional objects. *International Journal of Computer Geometry and Applications 1*, 4, 427–453.
- BORREL, P., AND RAPPOPORT, A. 1994. Simple constrained deformations for geometric modeling and interactive design. *ACM Transactions on Graphics 13*, 2, 137–155.
- CHANG, Y., AND ROCKWOOD, A. 1994. A generalised de casteljau approach to 3d free-form deformation. *SIGGRAPH'94 proceedings*, 257–260.
- CHEN, M., AND TUCKER, J. 2000. Constructive volume geometry. *Computer Graphics Forum 19*, 4, 281–293.
- COQUILLART, S. 1988. A sculpting tool for 3d geometric modelling. *Computer Graphics, (SIGGRAPH'88 Proceedings) 24*, 205–212.
- COQUILLART, S. 1990. Extended free-form deformation : A sculpting tool for 3d geometric modelling. *Computer Graphics, (SIGGRAPH'90 Proceedings) 24*, 4, 187–195.
- CRISPIN, B. 1998. Modélisation et déformation de forme libre à base de surfaces splines équipotentielles. *PhD thesis, Bordeaux University I*.
- FARIN, G. 1990. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Second Edition, Academic Press.
- JIN, X., LI, Y., AND PENG, Q. 1998. General constrained deformations based on generalized metaballs. *Proceedings of Pacific Graphics '98*, 115–124.
- KASIC-ALESIC, Z., AND WYVILL, B. 1991. Controlled blending of procedural implicit surfaces. *Graphics Interface'91*, 236–245.
- KUMAR, V., AND DUTTA, D. 1997. An approach to modeling multi-material objects. *Fourth Symposium on Solid Modeling and Applications, ACM SIGGRAPH*, 336–345.
- KUMAR, V., BURNS, D., DUTTA, D., AND HOFFMANN, C. 1999. A framework for object modeling. *Computer-Aided Design 31*, 9, 541–546.
- LAZARUS, F., COQUILLART, S., AND JANCÈNE, P. 1994. Axial deformations: an intuitive deformation technique. *Computer Aided Design 26*, 8, 607–613.
- LERIOS, A., GARFINKLE, C., AND LEVOY, M. 1995. Feature-based volume metamorphosis. *SIGGRAPH'95 Proceedings, Computer Graphics, Annual Conference Series*, 449–456.
- MACCRACKEN, R., AND JOY, K. 1996. Free-form deformation with lattices of arbitrary topology. *SIGGRAPH'96 Proceedings*, 181–188.
- MARTIN, W., AND COHEN, E. 2001. Representation and extraction of volumetric attributes using trivariate splines: a mathematical framework. *Sixth ACM Symposium on Solid Modeling and Applications, D. Anderson, K. Lee (Eds.), ACM Press*, 234–240.
- MIKITA, M. 1996. 3d free-form deformation: Basic and extended algorithms. *12th Spring Conference on Computer Graphics, W. Purgathofer editors, Comenius University, Bratislava*, 183–191.
- NISHITA, T., AND NAKAMAE, E. 1994. A method for displaying metaballs by using bezier clipping. *Computer Graphics Forum 13*, 3, 271–280.
- PARENT, R. 1988. A system for sculpting 3d data. *Computer Graphics 11*, 8, 138–147.
- PARK, S., CRAWFORD, R., AND BEAMAN, J. 2001. Volumetric multi-texturing for functionally gradient material representation. *Sixth ACM Symposium on Solid Modeling and Applications, D. Anderson, K. Lee (Eds.), ACM Press*, 216–224.
- PASKO, A., ADZHIEV, V., SOURIN, A., AND SAVCHENKO, V. 1995. Function representation in geometric modelling: concept, implementation and applications. *The Visual Computer 11*, 8, 429–446.
- PASKO, A., ADZHIEV, V., AND SCHMITT, B. 2001. Constructive hypervolume modelling. *Technical Report TR-NCCA-2001-01, National Centre for Computer Animation, Bournemouth University, UK, ISBN 1-85899-123-4, 34 p. URL: <http://wwwcis.k.hosei.ac.jp/F-rep/BTR001.pdf>*.
- PASKO, A., ADZHIEV, V., SCHMITT, B., AND SCHLICK, C. 2002. Constructive hypervolume modelling. *Graphical Models, Special issue on volume modeling 6*, 2.
- PEACHEY, D. 1985. Solid texturing of complex surfaces. *SIGGRAPH '85, Computer Graphics, USA, ACM Press 19*, 3, 279–286.
- PERLIN, K. 1985. An image synthesizer. *Computer Graphics, ACM Press, USA 19*, 3, 287–296.
- RUPRECHT, D., AND MUELLER, H. 1995. Spatial free form deformation with scattered data ζ interpolation methods. *Computers and Graphics 19*, 1, 63–71.
- SAVCHENKO, V., AND PASKO, A. 1998. Transformation of functionally defined shapes by extended space mapping. *The Visual Computer 14*, 5/6, 257–270.
- SAVCHENKO, V., PASKO, A., KUNII, T., AND SAVCHENKO, A. 1995. Feature based sculpting of functionally defined 3d geometric objects. *Multimedia Modeling. Towards Information Superhighway, T.S.Chua, H.K.Pung and T.L.Kunii (Eds.), World Scientific, Singapore*, 341–348.
- SCHMITT, B., PASKO, A., AND SAVCHENKO, V. 1999. Extended space mapping with bézier patches and volumes. *Implicit Surfaces '99, Eurographics/ACM SIGGRAPH Workshop, J. Hughes and C. Schlick (Eds.) (September)*, 25–31.
- SEDERBERG, T., AND PARRY, S. 1986. Free-form deformations of solid geometric models. *Computer Graphics (SIGGRAPH'86 proceedings) 20*, 4, 151–160.
- SINGH, K., AND FIUME, E. 1998. Wires: A geometric deformation technique. *SIGGRAPH'98*, 405–414.