Multidimensional free-form deformation tools

Dominique Bechmann

LSIIT ULP-CNRS UPRES-A 7005, Department of Computer Science, Louis Pasteur University, Strasbourg, France

Abstract

A survey of free-form deformation tools where the deformation is controlled by manipulating a 0-D to 3-D tool. Characteristics of a model that includes all these and generalises the concept will be presented.

1. Introduction

In this state of the art, we propose a survey of free-form deformation techniques where the deformation is controlled by manipulating a user-defined deformation tool. Recent researchs have provided deformation tools of any topological dimensions : 0-D (points), 1-D (lines), 2-D (surfaces) or 3-D (volumes).

The most famous of these models (see section 2) is certainely the one using 3-D parallelepipedical lattices [26] and all its extensions [10, 16, 12, 18, 19, 5, 22]. The latest invented [15] is using 2-D rectangular surfaces as deformation tool (see section 3). Axial deformation [20, 14, 23] are specifying the deformation through the modification of a 1-D line (see section 3). Finally, with models such as [7, 17, 8, 21, 1, 25] the deformation is simply defined by several user-defined point displacements i.e. 0-D constraints (see section 4).

These deformation techniques are all independent of the underlying object representation. It can be proved [4] that the underlying mathematical formalism for several of these models is just the same. The fact that they fall into a unique mathematical formalism involving a mapping establishes the links between these models. Thus, we will study the differences and the links from one formulation to the other.

However for the interactive modeling, the important questions are more : what kind of deformation can be easily achieved with these techniques ? Are they highly interactive and intuitive ? Thus, to answer these questions, we will explain, for each class of models, how to control the resulting deformation by manipulating the deformation tool. In particular, we will precise the position, the size and the boundary of the deformed area as well as the shape of the deformation. So a formal comparison as well as a practical one will be done (see section 5).

Finally, we will introduce the caracteristics of a free-form deformation model that would generalize and include all these deformation models using a multidimensional deformation tool (see section 6).

2. 3-D deformation tools

Deformation models requiring a 3-dimensional deformation tool such as a parallelepipedical volume called lattice are presented in this section. Using these techniques the deformation of an object is computed from the deformation applied to the 3-dimensional deformation tool.

2.1. FFD

The deformation tool used for the free-form deformation technique called FFD [26] is a trivariate volume defined by an array of $(p_1+1)(p_2+1)(p_3+1)$ control points $P_{i,j,k}$.

To deform an object the user deforms the lattice by moving its control points. Any point lying inside the lattice is deformed accordingly to the lattice deformation. In particular, the deformation of an object inside the lattice follows the displacement of the lattice control points.

Before deforming the lattice, each point of the object should be associated to the lattice. Let $U = (u_1, u_2, u_3)$ be the cartesian co-ordinates of a point in the global co-ordinate system (O, X, Y, Z), and $U = (u\prime_1, u\prime_2, u\prime_3)$ be its co-ordinates in the lattice co-ordinate system (U_0, R, S, T). The transformation of U from its global co-ordinates to its lattice co-ordinates is obtained from :

$$u\prime_1 = \frac{S \wedge T(U-U_0)}{S \wedge T.R} \; u\prime_2 = \frac{S \wedge T(U-U_0)}{S \wedge T.R} \; u\prime_3 = \frac{S \wedge T(U-U_0)}{S \wedge T.R}$$

where $\wedge$ represents the vectorial product and . the scalar product. The lattice co-ordinates of a point lying inside the lattice are such that : $0 \leq u\prime_1 \leq 1, 0 \leq u\prime_2 \leq 1, 0 \leq u\prime_3 \leq 1$. The cartesian co-ordinates of any point lying inside the lattice can be expressed as :

$$U = \sum_{i=0}^{p_1} \sum_{j=0}^{p_2} \sum_{k=0}^{p_3} P_{i,j,k} B_i^{p_1}(u\prime_1) B_j^{p_2}(u\prime_2) B_k^{p_3}(u\prime_3)$$

where the $P_{i,j,k}$ are the $(p_1+1)(p_2+1)(p_3+1)$ control points with $0 \leq i \leq p_1, 0 \leq j \leq p_2, 0 \leq k \leq p_3$ and $B_i^{p_1}$, $B_j^{p_2}$, $B_k^{p_3}$ are polynomials of degree p_1, p_2, p_3 respectively. Sederberg and Parry use the Bernstein polynomials to define the functions $B_i^{p_1}$, $B_j^{p_2}$, $B_k^{p_3}$.

Once the lattice is associated to the object, the user moves its control points. Finally, the cartesian co-ordinates of a point U after the deformation are computed from the new control points $P\prime_{i,j,k}$:

$$D(U) = \sum_{i=0}^{p_1} \sum_{j=0}^{p_2} \sum_{k=0}^{p_3} P\prime_{i,j,k} B_i^{p_1}(u\prime_1) B_j^{p_2}(u\prime_2) B_k^{p_3}(u\prime_3)$$

Due to the Bernstein polynomials properties, the displacement of one control point influences the whole space inside the lattice. Thus, the lattice exactly corresponds to the deformed area. However, the deformation is applied only to the points of the object lying inside the lattice.

Global deformations are obtained when the whole object lies inside the lattice since the displacement of a unique control point modifies the whole object. To localize the deformation, the technique consists in shrinking the size of the lattice. Local deformations are obtained when only a limited part of the object lies inside the lattice. In that case, the lattice intersects with the object. To preserve the continuity of the object's surface, the control points lying on the face of the initial lattice intersecting the object should not move. To preserve the continuity of the tangent or the curvature, stronger constraints are imposed on the lattice control points.

The deformation of the object (or its part included inside the lattice) "follows" the displacement of the control points of the lattice. Thus, it is not easy to get a precise displacement of a given point of the object. The user proceeds using a trial and error process to obtain the desired deformation. The shape of the polynomials defines the shape of the deformation inside the lattice.

To summarize, the deformed area is the volume embedded in the interior of the lattice. To modify its size or its position the user modifies the lattice. The boundary of the deformed area is a parallelepiped since the initial lattice is limited to a parallelepipedical volume. This technique is particularly well suited to global deformation. On the contrary, a given displacement of a point of an object is hard to obtain.

2.2. Extended FFD

Extended free-form deformation or shorter EFFD [12], extends the FFD technique to allow non-parallelepipedical lattices. In particular, elementary or composite prismatic lattices are defined. Elementary prismatic lattices are obtained by moving or merging control points of a parallelepipedical lattice. For example, the cylindrical lattice is obtained by welding two opposite faces of a parallelepipedical lattice and by merging all control points of the cylinder axis. Composite prismatic lattices are defined as several elementary lattices welded together. The welding operation is realised by merging the control points of each lattice. Some continuity problems may occur specially when several control points are merged together. In [12] one can see on the figure 10b page 196 that the continuity of the surface in the center of the lattice can not be checked. However in this example the fold effect was expected, and gives a visually satisfaying result. As continuity constraints would be penalizing for this technique, the authors insure lattice continuity only for the simplest cases. Non prismatic lattices can also be used. However the use of complex lattices can lead to unpredictable results.

In addition, elementary lattices are composed by several "chunks" where a chunk is a trivariate volume represented by a tensor product of Bernstein polynomials of degree three. Such a chunk is defined by $4*4*4$ control points.

To compute the deformation due to an elementary lattice, the co-ordinates of the object points in the lattice parameter space are computed. First, the chunk where the point is supposed to lie is determined. Then, the co-ordinates inside the chunk are computed using Newton approximation. Finally, the deformation of a point inside a given chunk is defined by

$$D(U) = \sum_{i=0}^{3} \sum_{j=0}^{3} \sum_{k=0}^{3} P\prime_{i,j,k} B_i^3(u\prime_1) B_j^3(u\prime_2) B_k^3(u\prime_3)$$

where the B^3, B_j^3, B_k^3 are Bernstein polynomials of degree three and the $P\prime_{i,j,k}$ are the $4*4*4$ control points with $0 \leq i \leq 3$, $0 \leq j \leq 3$, $0 \leq k \leq 3$. When a composite lattice is used, the deformations due to each elementary lattice are applied successively.

The definition of elementary lattices in chunks provides a local control of the deformation. Indeed, the displacement of a given control point modifies only the points lying inside the corresponding chunk. It is possible to localize a deformation by modifying the number of subdivisions of the lattice. However, the subdivision process is not easy to understand for a user not familiar with the Bernstein polynomials. Unfortunately, no intuitive tools are provided to control the deformed area : its size, its position and its boundary.

The deformed area corresponds exactly to the volume of the lattice only when every chunk is modified. Then, the user controls the deformed area by editing the position, the size and the boundary of the lattice. As with parallelepipedical lattices, the deformation could also be local when only a limited part of the object lies inside the lattice.

To conclude, the deformed area corresponds to the volume embedding the interior of the modified chunks but no intuitive tools are provided to control it. So, this technique is recommended for global deformation obtained when every chunk is modified through the deformation. Only in that case, the user controls the deformed area by manipulating the lattice. Due to that, for the user, local deformation obtained by moving a few control points seems hard to predict.

2.3. Rational FFD

Rational free-form deformation [18] is an other extension of FFD. It allows incorporation of weights defined at each control point of the parallelepipedical lattice. However, when the weights at each control point are unity, the deformations are equivalent to the FFD. To control the deformation, the user either moves the lattice control points or modifies their associated weights. The co-ordinates of a point are computed in the lattice parameter space before editing the lattice of control points. Then, the deformed point is computed from:

$$D(U) = \frac{\sum_{i=0}^{p_1}\sum_{j=0}^{p_2}\sum_{k=0}^{p_3} P'_{i,j,k} W_{i,j,k} B_i^{p_1}(u'_1) B_j^{p_2}(u'_2) B_k^{p_3}(u'_3)}{\sum_{i=0}^{p_1}\sum_{j=0}^{p_2}\sum_{k=0}^{p_3} W_{i,j,k} B_i^{p_1}(u'_1) B_j^{p_2}(u'_2) B_k^{p_3}(u'_3)}$$

where $W_{i,j,k}$ represents the weight associated to the control point $P'_{i,j,k}$ with $0 \leq i \leq p_1, 0 \leq j \leq p_2, 0 \leq k \leq p_3$ and where the $B_i^{p_1}$, $B_j^{p_2}$, $B_k^{p_3}$ are the polynomials of degree p_1, p_2, p_3.

The deformed area and the shape of the deformation both depend on the polynomial basis. The implementation of Rational FFD is presented with Bernstein polynomials of degree p_1, p_2, p_3 respectively. Thus, the deformed area corresponds to the lattice. The difference with FFD lies in the fact that a weight associated at each control point provides one more degree of freedom to define the deformation. However the unpredictability of the deformation obtained by changing the weight at a control point could be a limitation of this technique for the uninitiated user.

2.4. Others FFD techniques

Actually others extensions of the classical FFD model called NURBS-based FFD [16, 19] adopted a trivariate B-Spline volume as deformation tool. The principle is the same but, due to the B-Spline properties, local deformations can be easily defined. Indeed, by moving one control point of the lattice, only a limited area of space defined by some chunks of the lattice is deformed.

In order to obtain a deformation tool which can describe any 3D space subdivision, and to permit an easy control of the continuity of an object embedded in these lattices, deformation models [5, 22] based on the same principles as the ones of FFD are defined.

With the first one called Continuous FFD [5] by the authors, the deformation tool is a set of tetrahedral Bézier volumes. Initial lattices are made only of regular Bézier tetrahedrons, i.e. tetrahedrons which control points are regularly distributed with respect to the four vertices. This permits to directly compute the local coordinates of the object points in the lattices, without using a Newton algorithm. This model which can describe any subdivision of the tridimensional space also permits to control the C^1 continuity of the deformation of an object embedded in these lattices.

The other one [22] uses a lattice of artitrary topology that is subdivided with the Catmull-Clark subdivison method. After the first subdivision all cells of the resulting lattice have an hexahedral structure. After a few subdivisions, all cells are arranged in a regular pattern except at a finite number of points. Finally the initial lattice of arbitrary topology is refined n times until the largest cell of the lattice has a volume less than a specified size. The local co-ordinates of each point of the object in the subdivided lattice is computed. To deform the object the control points of the initial lattice are moved. Then the deformed lattice is once again subdivided n times. The new position of each point is easily obtained since its local co-ordinates according to a cell are invariant through the subdivision process.

Since the user only manipulates the initial lattice of arbitrary topology and not the refined one, the link between the displacement of control points and the deformation of the object might be quite hard to predict. Some problems also come from the subdivision process itself : very small cells together with very large ones can be obtained in the resulting lattice. The resulting deformation may suffer from such configuration. In addition, the deformed area is arbitrarily constrained by the lattice topology.

3. 2-D deformation tools

In this section, the object is controlled through the manipulation of a two-dimensional deformation tool. A model [15] where a so-called shape surface is used to control the deformation of an object is introduced.

The shape surface is defined by a B-Spline tensor product surface $S(u,v)$, initially forming a rectangular planar grid on the $X0Y$-plane. To deform an object, the user deforms the associated shape surface by moving its control points $P_{i,j}$.

First, for each point $U = (u_1, u_2, u_3)$ of the object, its projection U_P on the shape surface and along its normal is computed. Let (u'_1, u'_2) be the parametric co-ordinate of U_P such that $U_P = S(u'_1, u'_2)$ and u'_3 be the distance between U and U_P. So we have :

$$U = S(u'_1, u'_2) + u'_3 N(u'_1, u'_2)$$

where $N(u,v)$ is the vector normal to the shape surface.

The deformation model is defined such that the co-ordinate (u'_1, u'_2, u'_3) of any point U relatively to the shape surface are invariant. This property allows to compute the new position $D(U)$ of U :

$$D(U) = S'(u'_1, u'_2) + u'_3 N'(u'_1, u'_2)$$

where $S'(u,v)$ is the modified shape surface obtained after moving its control points and $N'(u,v)$ is the vector normal of the modified shape surface.

Through the control of a shape surface, it is quite easy to bent or to twist any object. However to obtain, for exemple, a tapered object, the user should be able to modify through the deformation, the distance $u\prime_3$ between a point and its projection on the shape surface.

The authors [15] propose to use the z component of a so-called height surface $H(u,v)$ as an additional parameter such that :

$$D(U) = S\prime(u\prime_1, u\prime_2) + H_z(u\prime_1, u\prime_2)u\prime_3 N\prime(u\prime_1, u\prime_2)$$

Initially, the height surface is a B-Spline tensor product surface forming the same rectangular planar grid than the shape surface, but on the plane $z = 1$.

The user controls the deformation by moving the control points of the shape surface and of the height surface. The shape of the deformation is strongly linked with the modified shape surface. Precise displacements of the object are hardly obtained.

The deformed area is not explicitly defined. Indeed, the deformation is global if the projection of the object lies totally on the shape surface. Then, global deformations [3] such as bending, tapering and twisting can be accomplished.

Otherwise, the deformation of the object is local and, to insure the continuity of the surface of the deformed object, the user should not move some control points of the shape surface.

4. 1-D deformation tools

Deformation models where the object deformation is controlled by a one-dimensional deformation tool i.e. an axis are presented in this section. The object deformation is linked to the axis deformation.

4.1. AxDf

Axial deformation called AxDf [20] is explained here. The first step is to attach the axis to the object. To each point $U = (u_1, u_2, u_3)$ of the object a point A_U on the axis is associated. A local co-ordinate system is defined around each point A_U of the axis. Then, the co-ordinates $U = (u\prime_1, u\prime_2, u\prime_3)$ of the object point in the local co-ordinate system associated to A_U are calculated using the matrix R_1. The axis is deformed and to a point A_U on the initial axis corresponds a point $A\prime_U$ on the deformed axis. The co-ordinates $D(U)$ of the deformed point in the local co-ordinate system associated to $A\prime_U$ are equal to the co-ordinate of $U = (u\prime_1, u\prime_2, u\prime_3)$ in the local co-ordinate system associated to A_U. The co-ordinate of $D(U)$ in the global co-ordinate system are calculated using matrix R_2 as follows :

$$D(U) = T_2 R_2 (T_1 R_1)^{-1} U$$

where T_1 is the translation matrix of vector OA_U and T_2 is the translation matrix of vector $OA\prime_U$.

The shape of the deformation is linked to the axis deformation. The only deformations that can be obtained are specified by bending or stretching the axis. In addition, scaling and twisting operations are possible by associating to each point of the axis a scale and a twist factors before and after the deformation. Finally, the axial deformation of a point is defined by :

$$D(U) = T_2 S_2 W_2 R_2 (T_1 S_1 W_1 R_1)^{-1} U$$

where the matrix W_1 of twisting angle q_1 and the matrix S_1 of scaling factor θ_1 are specified at each point A_U of the axis, and the matrix W_2 of angle q_2 and the matrix S_2 of factor θ_2 are applied to the deformed object points associated to $A\prime_U$.

Axial deformation influences the whole three-dimensional space although it is applied only to the point representing an object. In order to localize the deformed area, a zone of influence is introduced to defined the portion of the three-dimensional space to be deformed. A general cylinder around the axis is proposed to define the deformed area. It defines the size, the position and the boundary of the deformed area.

4.2. Deformation using De Casteljau algorithm

An other deformation model [14] using 1-dimensional deformation tool is introduced here. The 1-dimensional deformation tool is a Bézier curve.

A Bézier curve is defined by a set of control points forming the control polygon. The De Casteljau algorithm transforms a segment of the polygon of control into a segment of the curve by an iterative process.

Initially, the object to deform is mapped by an affine transformation on each segment of the control polygon. Then by applying the De Casteljau algorithm, the resulting object takes the shape of the Bézier curve.

Only global deformation such as streching, bending, twisting, tapering can be obtained with this model. To obtain local deformation the authors suggest to use De Boor algorithm [23] instead of De Casteljau.

5. 0-D deformation tools

For the models presented in this section the deformation tool is reduced to points such that the deformation of the object follows the displacement of these points.

5.1. DOGME

The deformation model called DOGME [7, 8] is summarized in this section. The deformation of an object is defined by the displacement of points called constraint points. In particular, a constraint point can be an object point. Then, it is trivial to achieve exact placement of object points. The model can satisfy as many constraints as the user enters unless two opposite constraints are applied to a unique point in space. In that case, the system computes the best approximation to the solution.

The deformation is expressed as the composition of an extrusion function f and a matrix M computed so as to achieve the constraints :

$$D(U) = U + \underbrace{M}_{(n*m)} \underbrace{f(U)}_{(m*1)}$$

The resolution method is based on pseudo-inverses. The dimensions of the matrix M and of the function f are written underneath the system.

The shape of the deformation around each constraint point depends on the so-called extrusion function f.

Local deformation around the constraint point is obtained using B-Splines polynomials for example. Due to the local support of the B-Splines polynomials only a limited area of space is deformed. Furthermore, the shape of the extrusion function is imprinted on the deformed area. Using B-Splines of degree greater or equal to two, the shape of the resulting deformation is very intuitive and smooth. Arbitrary shaped bumps can easily be designed using this technique.

A bounding box centred at each constraint point visualizes the extent of the deformation. The user interactively manipulates it to localize the deformation or to modify its shape. The size of the bounding box is linked to the support of the extrusion function. The boundary of the bounding box depends on the formulation of the extrusion function. In the initial model [7] the bounding box around each constraint point is a parallelepiped. A simplified version proposes a spherical deformed area [8]. An extension of the model [6] allows to define a different polyhedral bounding box around each constraint point.

On the contrary, the whole space can be influenced by a constraint using power basis polynomials. In that case, global deformation is obtained. The user could either choose the function f among a predefined set of functions or interactively build the appropriate one.

Cylindrical deformation are also defined [2] by using cylindrical co-ordinates instead of cartesian coordinates. In order to preserve the volume [1] of the object through the deformation, additional constraints also can be added.

5.2. Direct FFD and Multiple levels FFD

As the techniques reviewed in section 2, direct free-form deformation (Direct FFD) [17] also consists in embedding the object that has to be deformed inside a trivariate lattice defined by an array of control points. The object deformation follows the lattice deformation but the displacements of the lattice control points are computed from actions such that : "move this point of the object to there". Points displacements called constraints in the previous deformation model correspond to a 0-dimensional deformation tool.

Using Direct FFD it is trivial to achieve exact placement of object points. The computation of the deformed lattice satisfying the displacement of object points is transparent to the user. However, the displacement of a given object point drags the surrounding points.

Recall that the lattice is defined by trivariate B-Splines polynomials. Due to their local properties, the displacement of one control point influences a limited space area. The shape of the B-Splines polynomials is imprinted in the area surrounding a given displaced point. The size, the position and the boundary of the deformed area are strongly linked to the distribution of the control points of the lattice. Thus, to control the locality of the deformation, the user has to modify the initial lattice. No tools are offered to the user to control the deformed area. Due to that, part of the advantages of the direct manipulation are lost.

Multiple levels FFD is introduced [21] to solve problems that may occurs when several constraints influences the same control point of the lattice. The lattice is recursively refined by inserting control points. The resulting deformation tool is a multi-level lattice where each lattice is contributing to the displacement of only one constraint point.

5.3. Dirichlet FFD

A very interesting deformation model called Dirichlet free-form deformation [25] is presented in this section. This deformation model uses a 0-dimensional deformation tool since the deformation of the object follows the displacement of a set of points.

A set of points called here control points is initially positionned in space. The convex hull of this set of control points is considered. The Delaunay triangulation and the associated Voronoi diagram of this set of control points are computed. For a given point P, its natural neighbors can be defined as the control points with which P would share an edge in the Delaunay triangulation. Then P is linearly defined by its Sibson or natural co-ordinates defined over the set of natural neighbors in terms of areas in 2 dimensions or volumes in 3 dimensions. The region defined by P and its natural neighbors is the region of influence of P. Then, a multivariate Bézier simplex is built over the set of its Sibson neighbors such that the Bernstein polynomials value is tending to 1 when approching P and tending to 0 when approaching the border of the region of influence. Such a surface is called a Dirichlet surface.

Any point U in space so in particular points of an object can be expressed in term of the Dirichlet surface associated to its region of influence :

$$U = \sum_{I=0} P_I B_I^m(U\prime)$$

where $I = (i_1, i_2, \ldots, i_n)$ represents a multi-index, P_I are the control points, $U\prime = (u\prime_1, u\prime_2, \ldots, u\prime_n)$ are the Sibson coordinates of U and B_I^m the n-variate Bernstein polynomials of degree m.

This expression is invariant through the displacement of the constraint points, so the deformation $D(U)$ of a point U is defined as follows :

$$D(U) = \sum_{I=0} P\prime_I B_I^m(U\prime)$$

The author also explains [24] how to integrate various extensions to this deformation model. A weight can be associated to each control points in the same way as in Rational FFD. Exact displacement of an object point is obtained by identifying it with a control point.

Finally, Dirichlet FFD seems very convenient for local deformations. Indeed its application to hand modeling and animation needs a precise control of the deformed area. For each control point the deformed area is the region of influence defined by its natural neighbors. So the deformed area can be of any shape. The shape of the deformation in the deformed area is given by the Bernstein polynomials. Global deformation are also possible by moving control points in a group.

6. Comparison between multidimensional deformation techniques

To compare these deformation models it is important to establish the links between their mathematical formalism. But it is also crucial to compare them in terms of capabilities to deform objects.

All these models involve a mapping represented by the deformation function $D : R^3 \longrightarrow R^3$ (except [7] in R^n) that associates with each point U its new position $D(U)$. In addition, it can be prooved [4] that the mathematical model defining techniques using 3-D deformation tools such as FFD [26] and using 0-D deformation tools such as DOGME [7] is the same. Unfortunately, models using 1-dimensional or 2-dimensional tools are hardly included in the same formalism.

Eventhough, all these techniques involve a mapping represented by the deformation function, the interactive techniques they involve to define the deformation using the deformation tool are quite different.

Table 1 shows for which kind of deformations the different models are appropriate. For example, it is much easier to obtain exact displacement of object points using 0-dimensional deformation tool than by manipulating a 3-dimensional lattice. Deformation techniques using 0-dimensional deformation tool are particularly well suited to imprint arbitrary shaped bumps. Axial deformation is well adapted to bending or stretching. A surfacic deformation tool allows to define global deformations such as bending, tapering and twisting. Deformation with a 3-D lattice seems to be appropriate if it involves a regular set of control points.

The shape of the deformation is the same when using 3-dimensional or 0-dimensional deformation tools since it is simply linked to the polynomials (Bernstein, B-Splines, etc). However, the extrusion function of DOGME that defines the shape of the deformation is not restricted to polynomials. Thus, although these models are meant to provide free-form deformation, physical properties on the deformed object can be obtained with an appropriate choice for the extrusion function.

The extent of the deformation is predictable in all these methods. However, to control the position, the size and the boundary of the deformed area, DOGME provides a more flexible tool than others. For each implementation of the deformation models (first and second columns), Table 2 indicates which entity defines the deformed area (third column) and also indicates if the deformed area is visualized (fourth column).

Indeed, within DOGME only the points lying inside the bounding box move. The bounding box allows to control the extent of the deformation. It corresponds exactly to the deformed area.

Two techniques using a 1-dimensional deformation tool were presented. Deformation using De Casteljau algorithm is applied to the whole object so only global deformations are provided for now. The authors mention that local deformations can be achivied using De Boor's algorithm. AxDf also influences the whole space but the authors explain that it could be limited to a generalized cylinder.

When a shape surface controls the deformation, every point projecting on it is inside the deformed area. Initially the shape surface is planar, forming a regular grid, so the deformed area is simply a parallelepiped infinite in the direction perpendicular to the shape surface.

Using FFD techniques although only the portion of space inside the lattice is considered, no mention is made of the deformed area. Actually, when the lattice is defined by a trivariate Bernstein polynomial, its volume defines the deformed area. However with a different polynomial basis such as B-Splines, the size and the boundary of the deformed area when moving one control point depends on various parameters such as the number of control points of the lattice, their distribution through space and the degree of the polynomials. Thus, it is hardly predictable for a user not familiar with the effect of control point displacements. The Direct FFD model relying on FFD techniques has the same characteristics.

The function D defines the deformation of the whole space since it expresses the transformation of any point in R^3. Although the whole space is deformed, the deformation can be applied to a selected set of points only. Consider an object represented by a set of points with topological relations. The deformation modifies the position of the points independently of their topology. This property of indepen-

		Global deformation	Local deformation	Precise displacement
3-D	FFD, EFFD, RFFD	yes	continuity problems	no
3-D	NURBS-based FFD	yes	yes	no
3-D	Continuous FFD	yes	yes	no
3-D	Arbitrary topology FFD	yes	yes	no
2-D	Surface	yes (bend, twist, taper, etc.)	possible	no
1-D	AxDf	yes	possible	no
1-D	Curve	yes (De Casteljau)	yes (De Boor)	no
0-D	DOGME	yes	yes	yes
0-D	Direct FFD	yes	yes	yes
0-D	Dirichlet FFD	yes	yes	yes

Table 1: *Which kind of deformations for which deformation model ?*

		Deformed aera	Visualized ?
3-D	FFD, RFFD	parallelepipedical lattice	yes
3-D	EFFD	some prismatic or any shape chunks of the lattice	no
3-D	NURBS-based FFD	some parallelepipedical volumes	no
3-D	Continuous FFD	some tetrahedral volumes	no
3-D	Arbitrary topology FFD	some hexahedral volumes	no
2-D	Surface	the space projecting on the shape surface	no
1-D	AxDf	the whole space or a generalized cylinder	no
1-D	De Casteljau	the whole object	-
1-D	De Boor	part of the object	-
0-D	DOGME	bounding box (various shape) around constraint point	yes
0-D	Direct FFD	some parallelepipedical volumes	no
0-D	Dirichlet FFD	some voronoi convex cells	no

Table 2: *Possible shapes of the deformed area*

dence from the underlying representation of the object is common to all deformation methods presented here. Due to this independence, these methods can easily be integrated into most existing modelers or animation systems.

7. Discussion

The characteristics of a deformation model that includes all these and generalises the concept are discussed in this section.

First of all, the model should allow a large variety of deformation : global deformation such as bending, tapering, twisting, etc. and local deformation with a precise control of the deformed area and the deformation itself.

In term of interactive tools used to deform objects, it should be of any dimension or even of various dimension. More precisely the deformation tool could be a skeleton of any shape and any dimension.

The ideal model would be a mathematical generalisation of the various existing formulations. In fact it could be a multi-model that would use one existing model or the other depending on the skeleton. A skeleton reduced to a 1-dimensional tool could simply use AxDf for example. While a multi-dimensional skeleton could use a different model for each part.

Finally, such a model would be helpfull for animation since some animation models [13, 6, 2, 9] are already founded on these free-form deformation models.

References

1. F.Aubert, D.Bechmann, "Volume-Preserving Space Deformation", *Computer and Graphics*, **21**(5), Pergamon Press, (September-October 1997).
2. F.Aubert, D.Bechmann, "Animation by Deformation of Space-Time Objects", *EUROGRAPHICS'97* **21**(5), Budapest (Hongrie) (4-8 Septembre 1997).
3. Alan H. Barr, "Global and Local Deformations of Solid Primitives", *SIGGRAPH'84, ACM Computer Graphics* **18**(3), pp. 21–30 (1984).
4. D.Bechmann, "Space Deformation Models Survey", *Computer and Graphics* **18**(4), Pergamon Press, pp. 571–586 (July-August 1994).
5. D.Bechmann, Y.Bertrand, S.Thery, "Continuous Free-Form Deformation", *COMPUGRAPHICS'96 and a special issue of Computer Networks and ISDN Systems* Elsevier Science, Paris (16-18 December 1996).
6. D.Bechmann, N.Dubreuil "Order-controlled Free-Form Animation", *The journal of visualization and computer animation* **6**, pp. 11–32, (1995).
7. P. Borrel and D.Bechmann, "Deformation of N-Dimensional Objects", *Symposium on Solid Modeling Foundations and CAD/CAM Applications* ACM Press, Austin, Texas, (June 1991) Published in the *International Journal of Computational Geometry and Applications* **1**(4), pp. 427–453 (September 1991).
8. P. Borrel and A. Rappoport, "Simple Constrained Deformations for Geometric Modeling and Design", *ACM Transactions on Graphics* **13**(2), pp. 137–155 (April 1994).
9. S.Brandel, D.Bechmann and Y.Bertrand "STIGMA : a 4-dimensional modeller for animation", *9th EUROGRAPHICS Workshop on Animation and Simulation*, Lisbonne (Portugal) (31 August-1 September 1998).
10. John E. Chadwick, David R. Hauman and Richard E. Parent, "Layered Construction for Deformable Animated Characters", *SIGGRAPH'89, ACM Computer Graphics* **23**(3), pp. 243–252 (1989).
11. Yu-Kuang Chang and Alyn P. Rockwood, "A generalized de Casteljau approach to 3D Free-Form Deformation", *SIGGRAPH'94, ACM Computer Graphics* pp. 257–260 (1994).
12. Sabine Coquillart, "Extended Free-Form Deformations : A Sculpturing Tool for 3D Geometric Modeling", *SIGGRAPH'90, ACM Computer Graphics* **24**(4), pp. 187–196 (1990).
13. Sabine Coquillart and Pierre JancŔne, "Animated Free-Form Deformations : An Interactive Animation Technique", *SIGGRAPH'91, ACM Computer Graphics* **25**(4), pp. 23-26 (1991).
14. Yu-Kuang Chang and Alyn P. Rockwood, "A generalized de Casteljau approach to 3D Free-Form Deformation", *SIGGRAPH'94, ACM Computer Graphics* pp. 257–260 (July 1994).
15. Jieqing Feng, Lizhuang Ma and Qunsheng Peng, "A new Free-Form Deformation through the control of parametric surfaces", *Computer and Graphics* Pergamon Press **20**(4), pp. 531-539 (1996).
16. J.Griessmair and W.Purgathofer, "Deformation of Solids with Trivariate B-Splines", *EUROGRAPHICS'89* pp. 137–148 (1989).
17. William M. Hsu, John F. Hughes and Henry Kaufman, "Direct Manipulation on Free-Form Deformation", *SIGGRAPH'92, ACM Computer Graphics* **26**(2), pp. 177–184 (1992).
18. P.Kalra, A. Mangili, N.M. Thalmann and D. Thalmann, "Simulation of Facial Muscle Actions Based on Rational Free-Form Deformation", *EUROGRAPHICS'92, Computer Graphics Forum* **2**(3), pp. C59–C69 (1992).
19. Henry J. Lamousin and Warren N. Waggenspack, "NURBS-Based Free-Form Deformation", *IEEE Computer Graphics and Applications* pp 59–65 (November 1994).
20. Francis Lazarus, Sabine Coquillart and Pierre Jancene, "Axial deformation : an intuitive technique", *Computer-Aided Design* **26**, pp. 607–613 (1994).
21. S-Y. Lee, K-Y. Chwa, S. Y. Shin and G. Wolberg, "Image metamorphosis unsing snakes and free-form deformations", *SIGGRAPH'95, ACM Computer Graphics*, pp. 439–448 (1995).
22. R. MacCracken and K. I. Joy, "Free-form deformations with lattice of arbitrary topology", *SIGGRAPH'96, ACM Computer Graphics*, (1996).
23. M. Mikita, "3-D Free-form deformations : basic and extended algorithms", *12th Spring Conference on Computer Graphics - SCCG96*, pp 183-192, (1996).
24. Laurent Moccozet "Hand modeling and animation for virtual humans', *Ph.D thesis*, MIRA Lab and Geneve University, (1996).
25. Laurent Moccozet, Nadia Magnenat-Thalmann, "Dirichlet Free-Form Deformation and their application to hand simulation", *Computer Animation'97, IEEE Computer Society*, pp. 93-102 (1997).
26. Thomas W. Sederberg and Scott R. Parry, "Free-Form Deformation of Solid Geometric Models", *SIGGRAPH'86, ACM Computer Graphics* **20**(4), pp. 151–160 (1986).