

Deformations Expressed as Displacement Maps

An Easy Way to Combine Deformations

Hubert Peyré and Dominique Bechmann

Université Louis Pasteur — LSIIT
Pôle API — Boulevard Sébastien Brant
67400 Strasbourg — France
peyre, bechmann@dpt-info.u-strasbg.fr

Abstract. Up to now, assuming a geometric object as being a set of points, a deformation was defined as a mathematical function which mapped the original object set of points into a new one. In short, a new point was linked with each object point in the zone of influence of the deformation. With such a statement, it was not possible to apply several deformations simultaneously as soon as their zones of influence overlapped each other. A new formal definition of a deformation is brought out in this paper. Instead of a new point, this definition links a displacement with each object point in the zone of influence of the deformation. Thus, applying several deformations simultaneously is made feasible by blending, for each point of the original object, the displacement due to each deformation.

1 Introduction

In geometric modeling, deformations are both used to refine or to change the shape of a geometric object. Deformations have been introduced a long time ago into the field of geometric modeling. They have quickly and widely been accepted as a powerful technique and many deformation models have been worked out. Formally, in an Euclidean n -dimensional space, if Ω is a geometric object set of points and if Γ is the subset of points of Ω with regard to a given deformation, then this deformation is represented as the mathematical function denoted f , which maps each point from Γ into a new point in the Euclidean n -dimensional space.

$$\begin{aligned} f : \Gamma \subseteq \Omega &\rightarrow \mathbb{R}^n \\ P &\mapsto P' = f(P) \end{aligned} \quad (1)$$

In the remainder of this paper, for convenience reasons, whereas any point P in the domain Γ of the deformation f will be labeled *point to be deformed*, any point P' in the range $f(\Gamma)$ of this deformation will be denoted *deformed point*. If all the points of the geometric object Ω are in the domain Γ of the deformation, the deformation is said global (i.e. the whole geometric object is modified), else the deformation is said local (i.e. only a part of the geometric object is modified).

Afterwards, deformations have been divided into two classes depending on whether they resorted to a deformation tool or not. Barr's deformations, put

forward in [1], belong to the deformations class which require no deformation tool since they are matrix functions of the points location. As for the deformation models which have recourse to a deformation tool, most of them are laid-out in [2].

In its most basic form, a couple of geometric entities are enough to define a deformation tool : one will be labeled the *initial geometric entity* (IGE), the other being denoted the *final* or *deformed geometric entity* (DGE). The geometric entities may be of various types. For instance, the initial geometric entity of Sederberg and Parry's free-form deformations [8] is a parametric volume, the control points of which belong to a parallelepipedical lattice.

The process to perform a deformation which requires a deformation tool always follows the same broad outlines.

- First and foremost, the object is embedded into the initial geometric entity of the deformation tool. In other words, the location of the points to be deformed, so far expressed in the Euclidean coordinate system, is now expressed in a suitable coordinate system for the initial geometric entity of the deformation tool. This way, although the location of the points to be deformed remains unchanged, these points are possessed of new coordinates with regard to a new space coordinate system. These new coordinates are labeled *local coordinates with regard to the deformation tool*.
- Next, the local coordinates with regard to the deformation tool of each object point to be deformed are used in an appropriate coordinate system for the final geometric entity of the deformation tool in order to determine its new location in space. Therefore, to make a deformation intuitive, the coordinate system linked to the initial geometric entity of the deformation tool which characterizes this deformation must be consistent with the one associated to the final geometric entity.

From the user's point of view, the only tasks to carry out are first to specify the initial geometric entity and next to modify it into the final one. Both of these tasks may be achieved in an efficient interactive manner.

It has previously been pointed out that many deformation tools have been worked out and that the scope of their potential is indisputably wide. Nonetheless, combining several *auxiliary deformation models*, each based on an *auxiliary deformation tool*, in order to build up a new deformation model based on what might be called a *deformation multi-tool*, would be susceptible to grow up the possibilities of the existing deformation models. This paper aims to describe a deformation model based on such a deformation multi-tool.

The main problem is to specify the way the auxiliary deformation models have to be combined together. In fact, this problem is indistinguishable of the one which consists in defining the expected result of the achievement of a deformation, resorting to a deformation multi-tool, on a geometric object. Applying the auxiliary deformations successively would be an easy answer. This answer is not satisfactory at all however. Effectively, the fact that a deformation multi-tool combines several auxiliary deformation tools involves both their initial and

final geometric entities have to be defined. Assuming that the deformation multi-tool combines two auxiliary deformation tools and that one object point to be deformed is concerned with both the two auxiliary deformation tools, nothing guarantees that, once one of the two auxiliary deformations have been applied, the corresponding deformed point is still concerned with the other auxiliary deformation. On the other hand, even though the deformed point is still concerned with the other auxiliary deformation, whatever the auxiliary deformation applied first, the result of the successive auxiliary deformations depends on the order in which these auxiliary deformations are applied. Stating deformations as in (1), this remark can be proved easily since functional composition is not commutative with most of the existing deformation functions.

So a deformation model based on a deformation multi-tool have to meet both the two following properties.

- Any point to be deformed, concerned with several auxiliary deformations, must really undergo the effect of each auxiliary deformation.
- The result of a deformation based on a deformation multi-tool must not depend on the order in which the auxiliary deformations are achieved.

Stating a deformation as in (1) does not allow the first property to be met. Thus, another definition of a deformation is proposed in section 2. This definition describes a deformation through the expedient of the point displacement it generates. This way, when an object point to be deformed is concerned with several auxiliary deformations, the corresponding displacements can be blended so that the result of the deformation does not depend on the order in which the auxiliary deformations are applied. Section 3 depicts a deformation model based on a deformation multi-tool expanding the displacements blend concept. Implementation results of the deformation model based on a deformation multi-tool are finally set out in section 4.

2 Another Approach to the Definition of a Deformation

According to its own definition, a deformation maps each object point with regard to the deformation into a new point of the Euclidean n -dimensional space. In other words, any object point to be deformed undergoes a displacement from its initial location within the original object, to its final location within the deformed object. Consequently, any deformation can be defined through the expedient of the displacements it generates within the set of points of the geometric object undergoing this deformation. Formally stated, in an Euclidean n -dimensional space, if Ω is the set of points of a geometric object and if Γ is the subset of points of Ω with regard to a given deformation f , then the displacement of the points of Ω , generated by f , can be defined with the following mathematical function denoted g .

$$g : \Omega \rightarrow \mathbb{R}^n$$

$$P \mapsto g(P) = \begin{cases} \overrightarrow{PP'} = f(P) - P & \forall P \in \Gamma \\ \vec{0} = P - P & \forall P \in \Omega \setminus \Gamma \end{cases} \quad (2)$$

where f is the function stated in (1). From now on g will be called the *displacement function linked with the deformation f* . The fact that f is representative of a single deformation (i.e. of a single deformation tool) involves that g is also representative of a single deformation (i.e. of a single deformation tool).

Several works deal with displacement mapping [3, 6, 7]. Lewis [6] is the only one to use this concept to define a deformation model however.

It is significant to notice the displacement function puts on an intermediary act in the deformation definition. As soon as a displacement function is linked with a given deformation, the mathematical function which really describes this deformation is:

$$\begin{aligned} f : \Omega &\rightarrow \mathbb{R}^n \\ P &\mapsto P' = f(P) = P + g(P) \quad . \end{aligned} \quad (3)$$

Thus, the location of a deformed point is the vectorial sum between the location of the corresponding point to be deformed and the displacement generated by the deformation for this point to be deformed.

As a matter of fact, any deformation can be formulated through the expedient of a displacement function. Effectively, the traditional expression (1) of the definition of a deformation is inferred straight from the new one (3), substituting $g(P)$ for the expression (2) of its definition.

3 Definition of a Multi-Tool Deformation Model

This section aims to describe a multi-tool deformation model, that is a deformation model likely to combine several auxiliary deformation models, each based on an auxiliary deformation tool. As mentioned in the introduction, the accepted issue to reach that goal consists in blending, for each object point, its displacements resulting from each auxiliary deformation. The operator used to blend the displacements will be called *blend operator* henceforth and will be denoted \oplus . This way, assuming the multi-tool combines m auxiliary deformation tools, each described with its representative displacement function g_i , ($i \in \{1, \dots, m\}$), the deformation based on this multi-tool and applied to a geometric object Ω is defined with the following mathematical function denoted \tilde{f} .

$$\begin{aligned} \tilde{f} : \Omega &\rightarrow \mathbb{R}^n \\ P &\mapsto P' = \tilde{f}(P) = P + \bigoplus_{i=1}^m g_i(P) \quad . \end{aligned} \quad (4)$$

The blend operator needs necessarily to be commutative so that the result of the deformation does not depend on the order in which the auxiliary deformations are applied. This property is the only one this operator really needs to meet. Formally stated, assuming that Ω have to be modified by m auxiliary deformations, each resorting to an auxiliary deformation tool described by a displacement function g_i ($i \in \{1, \dots, m\}$):

$$\forall k, l \in \{1, \dots, m\}, \forall P \in \Omega : g_k(P) \oplus g_l(P) = g_l(P) \oplus g_k(P) \quad . \quad (5)$$

The vectorial sum is an attractive blend operator for displacements since, besides the fact it is a commutative operator, each point of the original object is moved to the accurate location expected by the user.

$$\forall P \in \Omega, \bigoplus_{i=1}^m g_i(P) = \sum_{i=1}^m g_i(P) . \quad (6)$$

Another example of blend operator for displacements might be the weighted average of the displacements.

$$\forall P \in \Omega, \bigoplus_{i=1}^m g_i(P) = \frac{\sum_{i=1}^m \alpha_i g_i(P)}{\sum_{i=1}^m \alpha_i} . \quad (7)$$

where α_i is a weight assigned to the auxiliary deformation tool the displacement function of which is g_i .

In fact, any blend operator can be used as long as it meets the commutative property (5).

As regards the computation process of a deformation based on a deformation multi-tool, its pseudocode is the one of Fig. 1. This pseudocode shows that achieving a deformation based on a deformation multi-tool is hardly more time-consuming than if all the auxiliary deformations was achieved sequentially. Effectively, in addition to the computation of each auxiliary deformation, the computation process only requires, for each point of the object:

- to compute, for each auxiliary deformation, the displacement (which is, actually, only a vectorial subtraction),
- to blend in the computed displacements,
- to translate the original point along the computed blend displacement (which is, actually, only a vectorial sum).

As regards blending in the computed displacements, it might seem absolutely necessary to compute, in a first stage and for each point of the object, the displacement generated by each auxiliary deformation before to blend them. This would be particularly inefficient. In fact, if the blend operator is of the type of those defined above (Equations (6) and (7)), the various sums can be updated each time a new displacement is computed, adding themselves in specific variables initially set to zero. This way, once all the sums are processed it only remains, if the case arises, to compute the quotients.

4 Results

The deformations based on deformation multi-tools utmost interest is to broaden the deformation scope. The modeling of a teapot, proposed in Fig. 2 and reached by deforming a sphere, a cylinder and a tube, shows the power of that new kind of deformations. The multi-tool used to arrive at this teapot combines nine deformation tools: three 1D-deformation tools for the lid, another 1D-deformation tool

```

For each object point Pi{
  For each auxiliary deformation{
    - Compute the deformation mapping from Pi
    - Compute the displacement of Pi
  }
  - Blend the computed displacements
  - Translate Pi along the blend displacement
}

```

Fig. 1. Pseudocode of the computation process of a deformation based on a deformation multi-tool.

for the base, a 1D-deformation tool, a 2D-deformation tool and a 3D-deformation tool for the spout, and both a 1D-deformation tool and a 3D-deformation tool for the handle. Besides, the vectorial sum is used as the blend operator.

Resorting to a deformation multi-tool certainly forces another way of handling deformations on the user. Most of the existing deformation tools provide the user with a strong intuition about the result they produce. As soon as several deformation tools are combined into a deformation multi-tool, the user must take into account the result locally produced by each auxiliary deformation in order to have an intuition of the one given by the multi-tool based deformation. Fig. 3, which displays the modeling of the teapot spout by deforming a tube, illustrates this remark. On the one hand, the tube is bent in a complex way with an axial deformation such as the one put forward in [5] and on the other hand it is tapered along two perpendicular directions with a 2D-deformation such as the one proposed in [4]; both of the two deformation tools being combined into one deformation multi-tool. This new “philosophy” may seem constraining for the user. Experimentation has proved the careful thought required from the user is rather slight however and that it quickly turns into a habit, nay a reflex, as soon as the user knows the behavior of the auxiliary deformations and of the blend operator.

Despite this fact, resorting to a deformation multi-tool affords a decisive advantage. Racked shapes which could only be reached handling a very complex deformation tool, can easily be produced splitting up this complex deformation tool into several auxiliary ones (easily handled) and combining them into one deformation multi-tool. Fig. 5 shows that there are three different ways for modeling the racked shaped object of Fig. 5(g), 5(h) and 5(i). The first way resorts to a rather complex 3D-deformation tool which is much too hard to define. The second and the third ways both resort to two easier to build 3D-deformation tools. Whereas the second way forces the user to apply the deformations in a given order, the third way combines both the deformation tools in one deformation multi-tool. The obvious advantage of the third way, compared with the

second, is that the user need not to bother to determine the order in which the deformations have to be applied.

Each deformation model presents particular interests so that it is generally used to carry out a specific task. Deformation multi-tools concept obviously dissociates itself from that limitation. Cases where the expected shape can not be reached easily resorting to a deformation multi-tool are likely to happen however. Fig. 4 illustrates such a case.

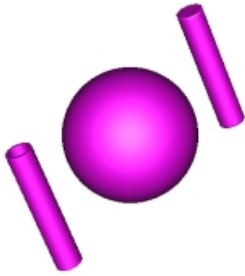
5 Conclusion

This paper aimed at combining several deformation tools, the geometric entities of which might be of various topologies or space dimensions. To reach this target, the developed technique consists, at first, in defining a deformation as a mathematical function which maps each point of the object to modify into a displacement instead of describing it the traditional way, that is as a mathematical function which maps each point of that object into a new one. Thus, while several deformations are combined, blending the displacements generated by these deformations, for each point of the object to modify, is enough.

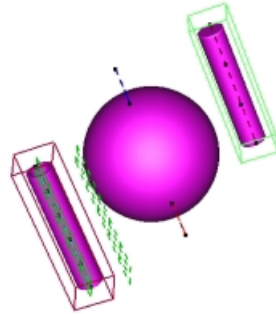
This technique has led to the concept of deformations based on deformation multi-tools. A deformation multi-tool is a deformation tool abstraction susceptible of combining several traditional deformation tools. One of the points of great interest regarding deformation tools lies in the fact they can combine any type of deformation tools. Therefore, deformation multi-tools can be extended endlessly since any deformation tool which would be worked out in the future could be combined with existing deformation tools into a deformation multi-tool.

References

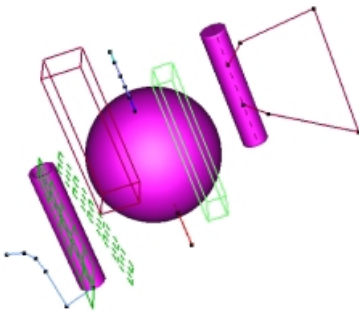
1. Barr, A. H.: Global and Local Deformations of Solid Primitives. *ACM Computer Graphics (SIGGRAPH'84)*. **18** (3) 21–34, 1984.
2. Bechmann D.: Multidimensional Free-Form Deformation Tools. State of the Art Report (EUROGRAPHICS'98), Lisbon (Portugal), 1998.
3. Cook R. L.: Shade Trees. *ACM Computer Graphics (SIGGRAPH'84)*. **18** (3) 223–231, 1984.
4. Feng J., Ma L. and Peng Q.: A new Free-Form Deformation Through the Control of Parametric Surfaces. *Computer and Graphics*, Pergamon Press. **20** (4) 531–539, 1996.
5. Lazarus F., Coquillart S. and Jancène P.: Axial Deformations: An intuitive Technique. *Computer-Aided Design*. **26** (8) 607–613, 1994.
6. Lewis J. P.: Algorithms for Solid Noise Synthesis. *ACM Computer Graphics (SIGGRAPH'89)*. **23** (3) 263–270, 1989.
7. Pedersen H. K.: Displacement Mapping Using Flow Fields. *ACM Computer Graphics Proceedings (SIGGRAPH'94)*, Annual Conference Series. 279–286, 1994.
8. Sederberg T. W. and Parry S. R.: Free-Form Deformation of Solid Geometric Models. *ACM Computer Graphics (SIGGRAPH'86)*. **20** (4) 151–160, 1986.



(a) Solid primitives: a sphere, a cylinder and a tube.



(b) Nine deformation tools IGEs — 6 1D-deformation tools (3 of them are superposed at the top of the sphere), 1 2D-deformation tool and 2 3D-deformation tools — combined into one deformation multi-tool.

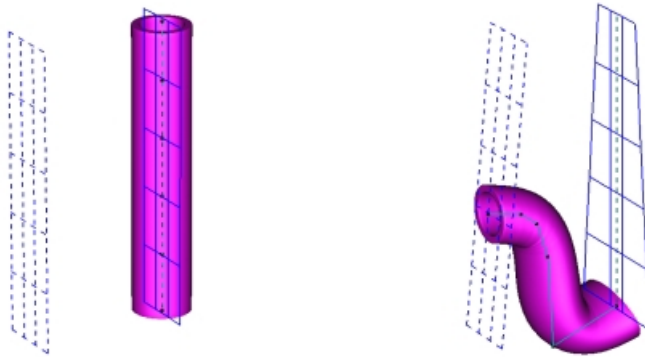


(c) Nine deformation tools DGEs combined into the deformation multi-tool.



(d) Result of the deformation, based on the deformation multi-tool, applied onto the solid primitives.

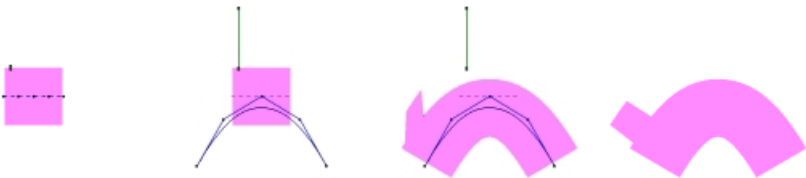
Fig. 2. Modeling of a teapot by deforming three solid primitives (a sphere, a cylinder and a tube) with a deformation multi-tool combining nine deformation tools.



(a) Tube and IGEs of the multi-tool.

(b) DGEs of the multi-tool and result of the deformation based on it.

Fig. 3. Modeling of the spout of the teapot displayed in Fig. 2 by deforming a tube with a deformation multi-tool combining both a 2D-deformation tool and 1D-deformation tool. The user must take into account the result locally produced by each auxiliary deformation tool to have an intuition of the one given by the deformation multi-tool.



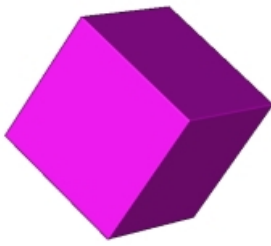
(a) A square plane patch and the IGEs of two axial tools.

(b) The same patch and the corresponding DGEs of the axial tools.

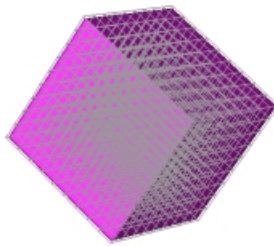
(c) Result of the deformation resorting to the multi-tool combining both the axial tools.

(d) Expected result of the deformation.

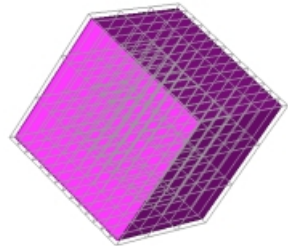
Fig. 4. A typical case where the expected result can not be reached easily resorting to a deformation multi-tool. In this case the expected result is reached more easily applying the “bumping” deformation than the “bending” deformation.



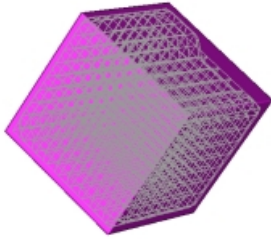
(a) Cubic geometric object to deform into Fig. 5(g), 5(h) and 5(i).



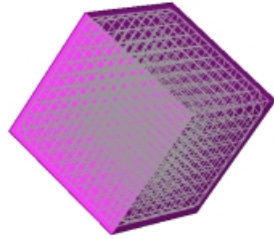
(b) IGE of two tools respectively used to round the edges and to provide the expected shape on its own.



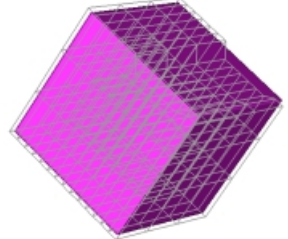
(c) IGE of a 3D-deformation tool used to imprint the recess in the solid primitive.



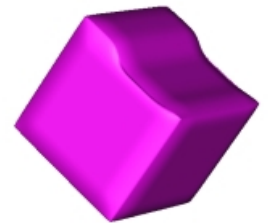
(d) DGE of the tool designed to provide the expected shape on its own.



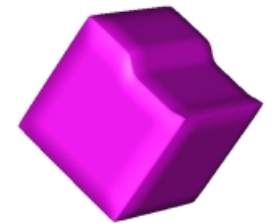
(e) DGE of the tool designed to round the edges.



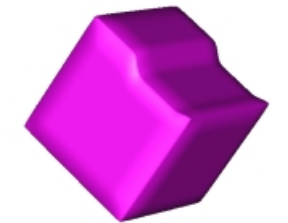
(f) DGE of the tool designed to imprint the recess.



(g) Result of the deformation pointed up by Fig. 5(b) and 5(d).



(h) Result of the successive deformations pointed up by Fig. 5(b) and 5(e) and by 5(c) and 5(f).



(i) Result of the deformation resorting to the multi-tool combining the same tools as in Fig. 5(h).

Fig. 5. Three different ways to reach nearly the same shape.