Implicit Extrusion Fields :
General Concepts and Some Simple Applications

L. Barthe♦ V. Gaildrat♦ R. Caubet♦
Department of Computer Science, Toulouse University, France

Abstract

This paper presents a new interpretation of the binary blending operator of implicit modeling. Instead of considering the operator as a composition of potential functions, we propose to consider it as an implicit curve extruded in an implicit extrusion field. An implicit extrusion field is a 2D space for which each coordinate is a potential field.
The study of general concepts around implicit extrusion fields allows us to introduce the theoretical notion of free-form blending controlled point-by-point by the user. Through the use of functional interpolation functions, we propose modeling tools to create, sculpt or combine implicit primitives by extrusion of a profile in an implicit extrusion field.

1. Introduction

One of the main constraints of interactive modeling is the control of a wide variety of shapes which must be easy, precise and intuitive. Parametric representation respects this constraint and allows the creation of free-form shapes. Parametric patches are controlled by attraction points and the shape is the result of their assemblage. On the other hand, implicit modeling involves volumetric object representation, defined by a single equation, and allows the introduction of automatic continuous blending between combined implicit primitives. Though many easily controlled primitives have been proposed, their blending suffers from a lack of precision.
Implicit interactive modeling is based on the combination of various implicit primitives with operators integrating the blend or not [1,2]. The blending notion is usually seen and computed as a smooth and regular curved or inflated transition. After blobs [3], soft objects [4] and metaballs [5], where spheres are blended by the sum of their potential fields, many models have been proposed to define new implicit primitives. Skeletons are the extension of spheres to a wider family. A skeleton is a simple geometric object [6,7,8] (like a point, line segment, free-form curve or polygon) and the shape is a set of points located at a fixed distance from the skeleton. Distance can be Euclidean or anisotropic [9,10,11,12]. Other primitive families have been proposed. Superquadrics [13,14] are degree-two algebraic functions controlled by parameters merged in their equation. Directly adapted from parametric sweep objects, implicit sweep primitives [12,15] are controlled by geometric parameters like trajectory and key profiles (interpolated along or around the trajectory). They greatly extend the panel of shapes produced. Improvement of intuitive shape control through the development of new primitives has been an important area of investigation but only few blending models exist and transition is approximately controlled by parameters that are merged in the surface equation. Our goal is to increase shape control precision at blend level. The solution proposed is based on a different interpretation of fundamental blending theory [16,17].
After a short overview of different blending models, a new interpretation of binary combination operators allows us to introduce implicit extrusion fields. An implicit extrusion field can be seen as a 2D implicit space where the value of each coordinate is an iso-potential surface in a potential field. Curves defined in an implicit extrusion field are represented by a surface in 3D user space. We indicate how surfaces can be precisely controlled by acting on curve properties and we deduce how free-form implicit curves defined point-by-point can be theoretically extruded in those fields to precisely combine, sculpt or model implicit primitives. We then present an implementation of implicit extrusion fields using curves defined by functions of $R \to R$ and a 2D elementary interface. Those functions do not exactly generate free-form curves but they are well known and they allow us to easily validate our theory. 3D visualization with an octree [18] is used to validate the resulting object shape if necessary.

♦ lbarthe@irit.fr, gaildrat@irit.fr, caubet@irit.fr
http://lbarthe.free.fr
Tel: 00 33 5 61 55 83 29 / Fax: 00 33 5 61 55 62 58
IRIT / University Paul Sabatier – 118 route de Narbonne
31062 Toulouse Cedex 4 – France

2. Implicit surface

Function f is of $R^3 \rightarrow R$. Function f associates a potential value C_p (of R) at each point p (of R^3) of the 3D user space. Function f defines a potential field. The set of points p of R^3, for which f(p) associates the same potential $C_p=C_0$, defines an iso-surface in the potential field. This iso-surface, called the C_0 iso-surface, is an implicit surface S and function f is called potential function.

$$\begin{cases} f : R^3 \rightarrow R \\ p(x,y,z) \rightarrow f(p) = C_p \end{cases}$$

$$S = \{p \in R^3 \,/\, f(p) = C_0\} \text{ where } C_0 \in R.$$

The potential function f splits space into two half spaces. One where $f(p)>C_0$ and one where $f(p)<C_0$. If f defines a closed object, the convention of inside/outside can be chosen as follows:

- If $f(x,y,z) > C_0$, the point p(x,y,z) is outside the volume defined by the surface.
- If $f(x,y,z) < C_0$, the point p(x,y,z) is inside the volume defined by the surface.

The inverse convention (where point p is inside if $f(p)>C_0$) can be chosen as well. Depending on the implicit model used, one or other of the conventions is applied.

3. Blending implicit models

- In its elementary form, blending is performed as follows:

The potential function f_i defining the primitive is first composed with a blending function g_i. The resulting function $g_i(f_i)$ is a decreasing positive function with $g_i(f_i) \rightarrow 0$ when $f_i \rightarrow +\infty$ (Figure 1).

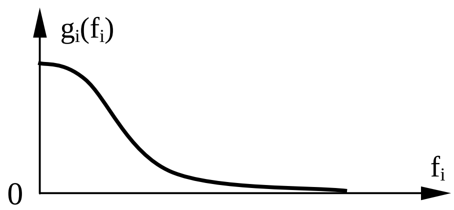


Figure 1: blending function g_i representation.

The blend is computed by summing the functions $g_i(f_i)$:

$$F = \sum_i g_i(f_i)$$

The blending function g was originally defined with an exponential function [3]. To increase computation speed and to localize the influence of the primitive, polynomial functions including an influence radius R have been proposed [4,5,11,18,19,20] (see [11,21] for an overview). The use of a blending function g as a first step and a sum as a second step generates a double abstraction level to control the transition precisely. For this reason we did not pursue our research in this direction beyond the results presented in [22].

- Another approach consists in generalizing the blend as an operator on R-functions[1]. An n-ary operator Ψ can be written as the composition of each combined primitive f_i [23]. One expression for the classical union binary operator R is then:

$$R(f_1, f_2) = f_1 + f_2 + \sqrt{\left(f_1^2 + f_2^2\right)}.$$

Equation 1.

Blend operator G can be obtained by adding matter at the transition [24]. The matter adding operator d is summed with the R operator to give a blend operator G:

$$G(f_1, f_2) = R(f_1, f_2) + d(f_1, f_2) \quad \text{with}$$

$$d(f_1, f_2) = \frac{a_0}{1 + \left(\frac{f_1}{a_1}\right)^2 + \left(\frac{f_2}{a_2}\right)^2}.$$

Transition is controlled by acting on the parameters of matter adding operator d parameters. These parameters (a_0, a_1, a_2) are merged in an equation and are not directly liked to geometric parameters (like control points, etc).

- Whatever the blending method used, transition creation is then an iterative succession of 'adapting' the value of parameters and visualization.
- As specified by C. Hoffmann and J. Hopcroft in a first time [16] and later by A.P. Rockwood [17], the binary blending operator G is created in two steps. Operator G, called the blending function, is first defined as a function of $R^2 \rightarrow R$ that creates a smooth transition between the two axes. The blend is then, in the second step, extended to primitives by composing G with f_1: $R^3 \rightarrow R$ and f_2: $R^3 \rightarrow R$. The resulting surface is the 0 iso-surface of the potential field defined by G: $R^3 \rightarrow R$. Many blending functions are proposed, especially functions allowing the control of the starting point of the blend on each blended surface and functions extending the blend to n primitives. If the iso-value is not 0, explanations and solutions are given to adapt the blending functions.

The starting point of our approach is very close to these works. The main difference is to consider operator G as a 2D potential field defined in space where each coordinate (X and Y) is a 3D potential field (f_1 and f_2 respectively) instead of considering G as a composition of functions f_1 and f_2. This special space is called the implicit extrusion field. This difference seems to be insignificant but we will see that it allows us to extend the blend to a theoretically free-form blend and, at the same time, to propose tools for implicit modeling.

[1] Implicit surfaces are defined by 0 iso-surface and the convention of inside/outside is : if f(p)>0, p is inside the volume, if f(p)<0, p is outside the volume.

4. Implicit extrusion fields: introduction and concepts

4.1. Presentation, nomenclature and conventions

- Implicit surfaces are 0 iso-surfaces and volumes are defined by $f(p)\leq 0$.
- An implicit extrusion field is a 2D implicit space called I^2.
- Geometric entities defined in implicit extrusion fields are noted in capitals and entities defined in 3D Euclidean space are noted in small letters.
- I^2 is a space where each coordinate is a potential field. To define an implicit extrusion field, each coordinate is instantiated with a selected potential function of $R^3 \rightarrow R$: $X\equiv f_1$ and $Y\equiv f_2$.
- POINT $P(X_P,Y_P)$ of I^2 is defined by its two coordinates $f_1=X_P$ and $f_2=Y_P$. The abscissa is a set of points $p(x_p,y_p,z_p)$ of R^3 for which $f_1(p)=X_P$. It is the X_P iso-surface S_1 of the potential field defined by f_1 and the ordinate is a set of points $p(x_p,y_p,z_p)$ of R^3 for which $f_2(p)=Y_P$. The ordinate is the Y_P iso-surface S_2 of the potential field defined by f_2. POINT P is represented by the intersection between its two coordinates, which means that its representation is the intersection between the two surfaces S_1 and S_2. This intersection is a curve V (Figure 2) if it is not empty (or reduced to a single point).

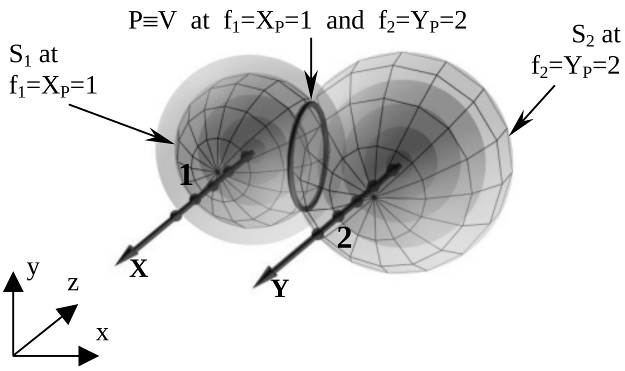


Figure 2: POINT P(1,2) defined in I^2 and represented in R^3 (I^2 is instantiated with two spherical fields).

$$V = S_1 \cap S_2 \text{ where}$$
$$S_1 = \{p(x,y,z) \in R^3 \,/\, f_1(p) = X_P\} \text{ and}$$
$$S_2 = \{p(x,y,z) \in R^3 \,/\, f_2(p) = Y_P\}.$$

Curve V is said to be the extrusion result of POINT P in the implicit extrusion field.

- Function G(X,Y) of $I^2 \rightarrow R$ defines a 2D potential field. The set of POINTS $P(X_P,Y_P)$ where G(P)=0 defines the 0 ISO-CURVE. This CURVE is called PROFILE and continuous PROFILE can be seen as a succession of juxtaposed POINTS P. POINT P is represented by a curve in R^3, PROFILE G is represented by a succession of curves juxtaposed in R^3, which means that PROFILE G is represented by surface S in R^3 (Figure 3). This implicit surface S is given by the set of points $p(x,y,z) \in R^3$ where $G(f_1(p),f_2(p)) = 0$.

$$S = \{p(x,y,z) \in R^3 \,/\, G(f_1(p), f_2(p)) = 0\}.$$

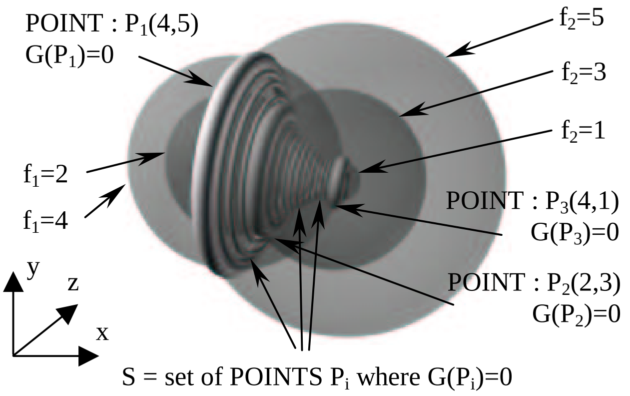


Figure 3: PROFILE G defined in I^2 and represented in R^3 (I^2 is instantiated with two spherical fields).

Surface S is said to be the extrusion result of PROFILE G in the implicit extrusion field.

4.2. Links between implicit extrusion field I^2 and 3D modeling space R^3

We have seen that PROFILE, defined in an implicit extrusion field, is represented by an implicit surface in user modeling space R^3. But different instantiations of implicit extrusion fields give different shapes for the same PROFILE. This is why the form and position of the generated surface are difficult to predict by the user. Our goal is to propose a precise modeling tool, so an intuitive link must be established between spaces R^3 and I^2.

The user can easily select a point $p(x_p,y_p,z_p)$ of R^3 in the modeling space (using a suitable modeling interface). Potential function f_1 associates the potential value X_P at this point: $f_1(p)=X_P$ and potential function f_2 associates the potential value Y_P at the same point: $f_2(p)=Y_P$. POINT $P(X_P,Y_P)$ selected from point $p(x_p,y_p,z_p)$ has the following coordinates: X_P iso-surface of potential field f_1 as abscissa and Y_P iso-surface of potential field f_2 as ordinate (Figure 4).

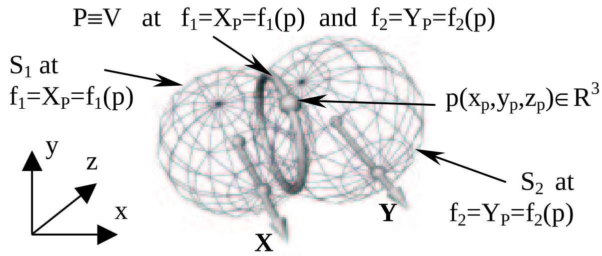


Figure 4: POINT P of I^2 selected from point p of R^3.

It is important to note that point p belongs to the curve representing POINT P in R^3. So, by selecting points in the modeling space, the user precisely and simply selects POINTS of I^2.

By selecting two points, $p_1(x_{p1},y_{p1},z_{p1})$ and $p_2(x_{p2},y_{p2},z_{p2})$, the user can choose vector $u(x_u,y_u,z_u)$ of R^3. The initial point p_1 defines POINT $P_1(X_{P1},Y_{P1})$ of I^2. From this POINT P_1 and vector u, differential geometry equations allow the computation of VECTOR $U(X_U,Y_U)$ coordinates as follows:

If $A=(f_1,f_2)$ is an application from R^3 to I^2, and ∇A is Jacobean matrix of A,

$$U = \nabla A(p_1).u \, .$$

$$U \begin{pmatrix} X_U = \frac{\partial f_1(x_{p1},y_{p1},z_{p1})}{\partial x}.x_u + \frac{\partial f_1(x_{p1},y_{p1},z_{p1})}{\partial y}.y_u + \frac{\partial f_1(x_{p1},y_{p1},z_{p1})}{\partial z}.z_u \\ Y_U = \frac{\partial f_2(x_{p1},y_{p1},z_{p1})}{\partial x}.x_u + \frac{\partial f_2(x_{p1},y_{p1},z_{p1})}{\partial y}.y_u + \frac{\partial f_2(x_{p1},y_{p1},z_{p1})}{\partial z}.z_u \end{pmatrix}$$

VECTOR U is represented by a family of directions in R^3 (Figure 5).

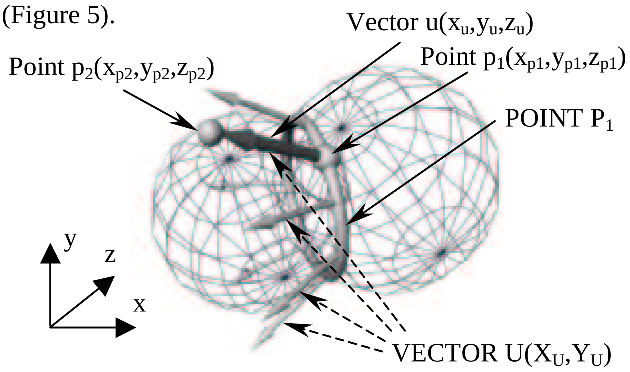


Figure 5: VECTOR U of I^2 selected from vector u of R^3.

Like points, vector u belongs to the family of directions representing U. This property allows the user to precisely select VECTORS of I^2 from vectors of the modeling space.

4.3. Correspondence between function G represented in I^2 and the same function G in R^3

We recall that the normal N(P) at a POINT P to PROFILE G is given by the gradient vector $\nabla G(P)$:

$$N(P) = \nabla G(P) = \begin{pmatrix} \frac{\partial G(X_P,Y_P)}{\partial X} \\ \frac{\partial G(X_P,Y_P)}{\partial Y} \end{pmatrix}.$$ We can deduce the tangent VECTOR:

$$T(P) = \begin{pmatrix} \frac{\partial G(X_P,Y_P)}{\partial Y} \\ \frac{-\partial G(X_P,Y_P)}{\partial X} \end{pmatrix}.$$

The user can control the resulting surface by controlling the POINTS and VECTORS defining the PROFILE from the points and vectors selected in the modeling space. So, POINTS and VECTORS must be control parameters of G PROFILE. Figure 6 reports an example of G PROFILE represented in I^2 and figure 7 shows the same G PROFILE represented in a 2D section of R^3. POINTS P_i (i=1..3) and TANGENTS T_i (i=1,3) are control parameters of PROFILE G.

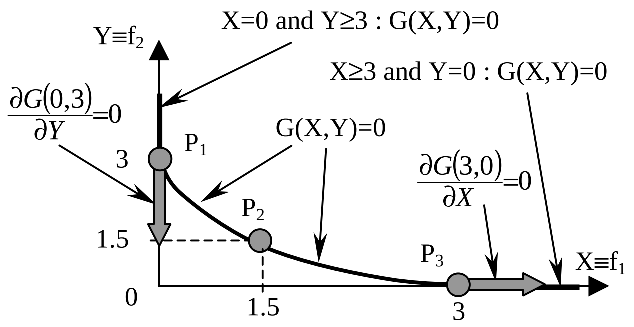


Figure 6: Function G represented in I^2.

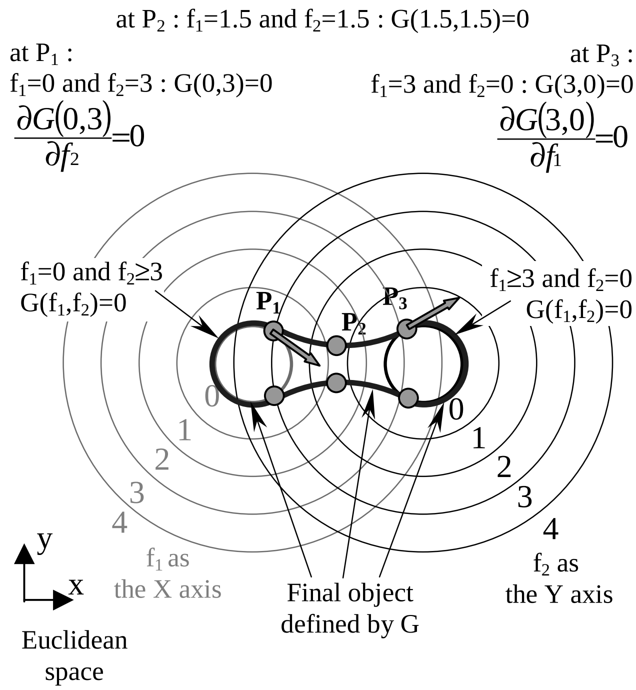


Figure 7: Function G, defined in figure 6, represented in a 2D section of modeling space R^3 (I^2 is instantiated with two spherical fields).

In figures 6 and 7, specific properties are revealed for implicit extrusion fields:

1. For the regions of the PROFILE where POINTS P have a fixed abscissa ($X=X_P$) as the ordinate Y varies: the associated points p of R^3 are situated on the X_P iso-surface of the potential field defined by f_1. If in addition $X_P=0$, these points of R^3 are on the surface defined by f_1.

2. For the regions of PROFILE where POINTS P have a fixed ordinate ($Y=Y_P$) as the abscissa X varies: the associated points p of R^3 are situated on the Y_P iso-surface of the potential field defined by f_2. If in addition $Y_P=0$, these points of R^3 are on the surface defined by f_2.

3. If $\partial G(X_P, Y_P)/\partial X = 0$: a null value of the differential in X at a POINT $P(X_P,Y_P)$ of I^2 leads to the surface representing PROFILE G being tangential at P to the Y_P iso-surface defined by f_2. If in addition $Y_P=0$, this surface is tangential to the implicit surface defined by f_2.

4. If $\partial G(X_P, Y_P)/\partial Y = 0$: a null value of the differential in Y at a POINT $P(X_P,Y_P)$ of I^2 leads to the surface representing PROFILE G being tangential at P to the X_P iso-surface defined by f_1. If in addition $X_P=0$, this surface is tangential to the implicit surface defined by f_1.

We obtain a model which allows implicit surfaces defining implicit extrusion field coordinates to be partially or totally conserved in the final object (if desired). Continuity C^0 or C^1 at the junction between the surface representing PROFILE G and the one defined by one or other of the coordinates can be controlled by the value of partial differentials. In general, C^1 continuity depends on functions G, f_1 and f_2 continuity (the final function is given by the composition of G: $I^2 \rightarrow I$ with functions f_1: $R^3 \rightarrow R$ and f_2: $R^3 \rightarrow R$).
In this example, the final object is the result of the blend operator applied on two spheres. The transition is smooth, continuous and controlled point-by-point.

4.4. Extrusion models

Depending on the inclusion of implicit surfaces, defined by coordinates of the implicit extrusion field, in the final object, different tools can be theoretically created.

4.4.1. Extrusion objects If 0 iso-surfaces defined by coordinates of the implicit extrusion field are not conserved in the final object, this object is directly the result of the extrusion of the PROFILE in the implicit extrusion field (Figure 8).

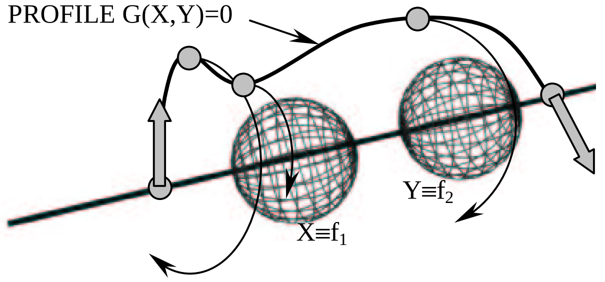


Figure 8: Extrusion of PROFILE G in an implicit extrusion field instantiated with two spherical fields.

Extrusion trajectories are given by the intersection between iso-surfaces of each coordinate of the implicit extrusion field. This is an abstraction level which makes our extrusion model less general and more complicated to use than models of translational or rotational extrusion [12,15]. But if free-form PROFILES are defined (which is not exactly done in this paper), they would be extruded with our approach, and this should greatly extend the variety of shapes produced.

4.4.2. Sculpture If one of the 0 iso-surfaces defined by the coordinates of the implicit extrusion field is conserved in the final object, PROFILE extrusion directly sculpts the conserved surface (Figure 9). Particular attention must be paid to the complexity of the sculpted surface. If its potential field is too irregular, the shape produced from the sculpting will be uncontrollable.

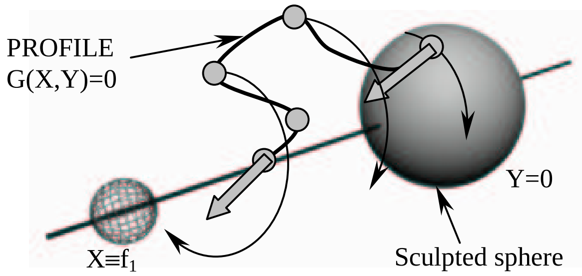


Figure 9: Spherical ordinate 0 iso-surface sculpted by PROFILE extrusion.

4.4.3. Binary blending operator If the two 0 iso-surfaces defined by the coordinates of the implicit extrusion field are conserved, PROFILE extrusion performs the blending (as seen in figure 7). If free-form PROFILES are proposed, the classical notion of a smooth and regular curved transition will be extended to free-form blending. If in addition these PROFILES are defined point-by-point, the transition will be created simply and precisely.

5. Applications using PROFILES defined by functions of R→R

To validate the theory presented, we propose to use PROFILES defined by functions of R→R. Indeed, these functions are well known whereas defining implicit curves represented by the G(X,Y)=0 equation and controlled point-by-point is a research topic on its own.

5.1. How to define a PROFILE G(X,Y) with a function H of R→R

A function H of R→R is defined by the following expression: Y=H(X). This expression can be written as:

Y-H(X)=0. So, we can directly deduce a possible definition of PROFILE G:

$$G = Y - H(X).$$

The use of a functional definition generates a limitation in the form of the curves generated. Indeed curves defined by functions must be monotonous in the abscissa direction (at a fixed $X=X_0$, at most one Y value must exist such that $Y=H(X_0)$). This implies a direct limitation: The 0 iso-surface of the abscissa field cannot be included in the final object (Figure 10).

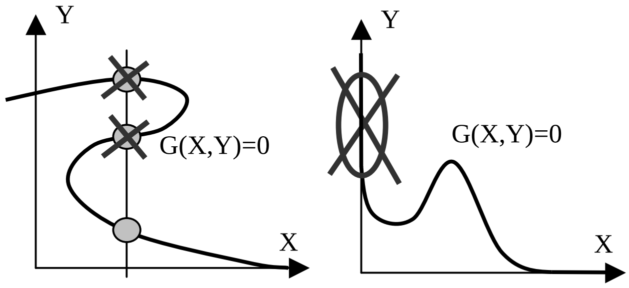


Figure 10: Form restriction of function curves.

PROFILE parameters are POINTS and VECTORS. This is why we propose to use interpolation functions. We have chosen 1D cubic polynomial splines [25] for their good smoothness and oscillation properties.

5.2. 2D elementary interface of validation

The 2D space visualized (Figure 11) is a plane section of the 3D working space. It is important to choose a plane which intersects the potential fields correctly. The plane is set where the outline of the final shape is to be controlled. A poor choice of the plane will considerably decrease the intuitive link between the 2D outline and the 3D shape. This condition obliges the user to have a working knowledge of potential functions and of the fields generated by implicit primitives.
To allow the user to respect the function properties, the X axis must be visualized. The f_1 potential field is visualized as a background picture using gray graduations (black when $f_1=0$ and white when $|f_1|$ is max). To complete the field reference, outlines of f_1 and f_2 0 iso-surfaces are visualized. The user can act on the shape outline by moving, adding, or removing control points or by acting on tangency at the first or the last point of the profile. Points and tangents are interactively selected in the interface with the mouse.

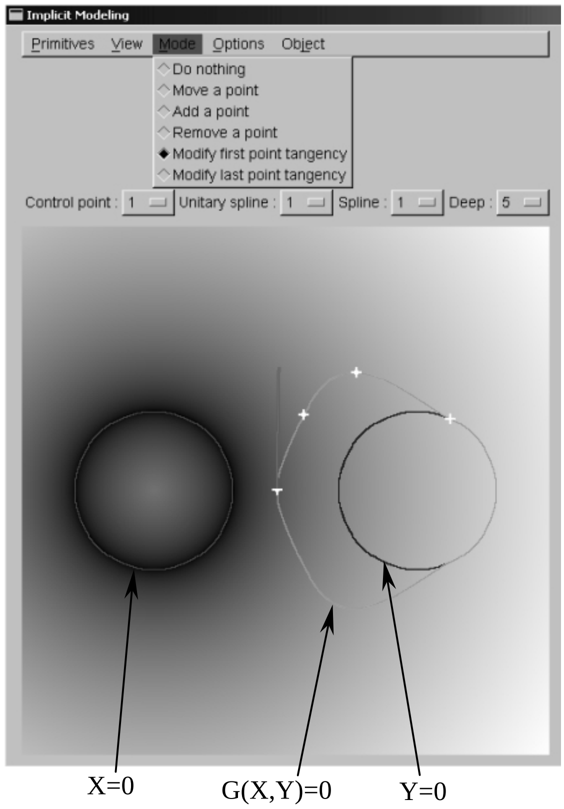
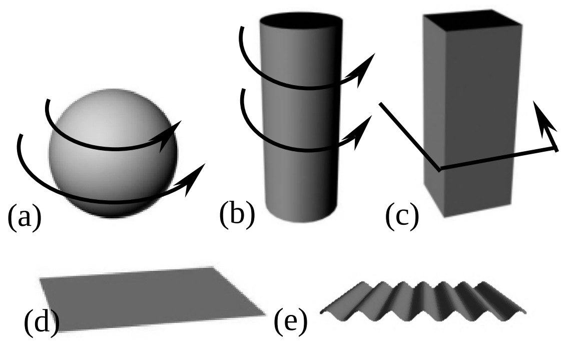


Figure 11: Our interface of validation.

5.3. Control of the extrusion trajectories

To create an object, extruded PROFILE (section 5.1.) and extrusion trajectories have to be defined. Trajectories are defined by intersections between the iso-surfaces of the coordinates of the implicit extrusion field (section 4.1.). To control trajectories, we propose to use one of the potential fields to define the extrusion support. PROFILE is extruded 'around' the iso-surfaces of this potential field. The other potential field is used to define the direction of extrusion. PROFILE is extruded around the support and along the direction of extrusion (Figure 12).



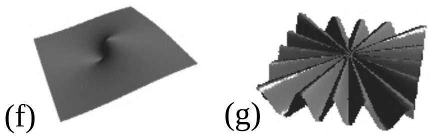


Figure 12: Examples of support (a,b,c) and direction (d,e,f,g) of extrusion.

For example, directions can be simply defined by functions like:

$$l : \Re \rightarrow \Re \qquad f : \Re^3 \rightarrow \Re$$
$$x \rightarrow y = l(x) \qquad (x, y, z) \rightarrow y - l(x)\cdot$$

Some functions used to create our directions are:

$$f(x, y, z) = y - a.\cos(b.x)$$

or $\alpha = \arccos\left(\frac{x}{\sqrt{x^2 - z^2}}\right)$, $f(x, y, z) = y - a.\cos(b.\alpha)$,

and surfaces shown in Figure 12 (e,f,g) are given by the following equations:

e: $y - \cos(x) = 0$, **f:** $y - \cos(\alpha) = 0$, **g:** $y - \cos(8.\alpha) = 0$.

5.4. Modeling tools

As seen in section 5.1., due to the use of functional PROFILES, implicit surfaces defined in the abscissa of the implicit extrusion field cannot be included in the final object. Depending on the use of the abscissa or the ordinate as support and direction of extrusion, different modeling tools are obtained. Once the tool has been selected and the extrusion fields instantiated, the PROFILE can be created using a suitable interface (see section 5.2. for an elementary validation interface).

5.4.1. Sculpture on a surface The sculpted implicit surface is selected as the ordinate of the implicit extrusion field (to be able to be included in the final object) and as direction of extrusion. It is the extrusion of the PROFILE around the support (selected as the abscissa) which sculpts the surface (Figure 13(a,b,c,d)). The abscissa field represents the sculpture tool. With the same PROFILE, a surface can be sculpted with different supports to generate various extrusion trajectories (Figure 13(e)).

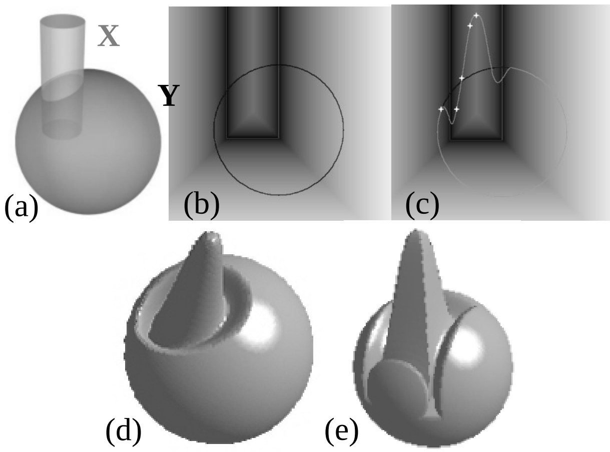


Figure 13: (a) The sphere (ordinate and direction) is sculpted around a cylindrical (abscissa and support) tool, (b) representation in our 2D interface ,(c) a profile is defined by the user, (d) resulting object, (e) the extrusion support is now a parallelepiped.

5.4.2. Sculpture around a surface The sculpted surface is again selected as the ordinate and as the support of extrusion. PROFILE is extruded around the sculpted object and along the direction of extrusion selected as the abscissa (Figure 14). With the same PROFILE, a surface can be sculpted with different direction to generate various extrusion trajectories (Figure 15).

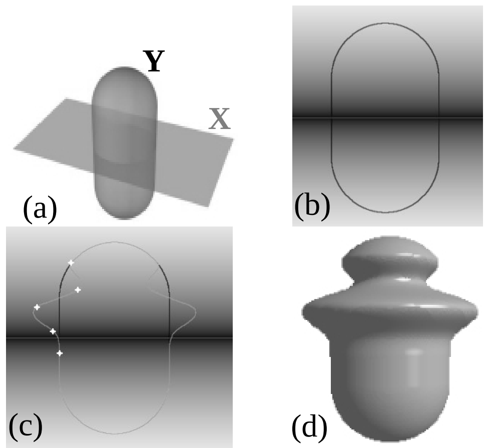


Figure 14: (a) The capsule (ordinate and support) is sculpted along a plane field (abscissa and direction), (b) representation in our 2D interface, (c) a profile is defined by the user, (d) resulting object.

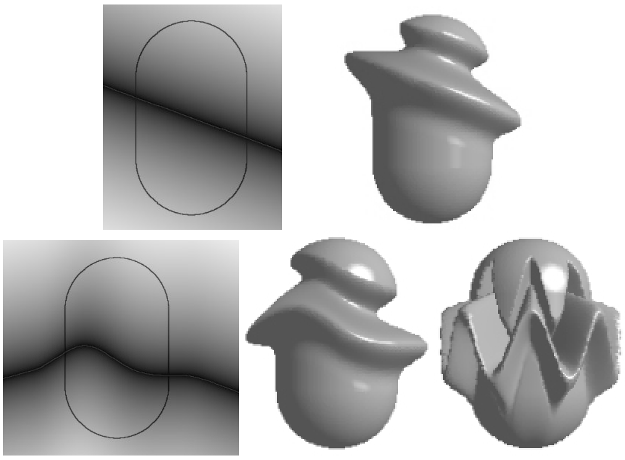


Figure 15: Different forms of direction and resulting objects.

5.4.3. Extrusion objects In this case, trajectories are controlled by acting on the support and the direction. An example of a extrusion object is given in figure 16. In this example, the PROFILE is extruded around cylinders (support) selected as the ordinate. A cylinder is an infinite surface and to be sure that part of the cylinder is not included in the final object, we use a cylindrical field without a 0 iso-surface. The effects generated in changing the direction of extrusion are illustrated in figure 17 and the effects generated in changing the support of extrusion are illustrated in figure 18. Precautions have to be taken when these two parameters are combined to obtain the desired trajectories. Indeed undesired and uncontrollable effects can be generated in the PROFILE extrusion trajectories (Figure 19). A good knowledge and understanding of the creation of extrusion trajectories are necessary to control complex field combinations nicely.

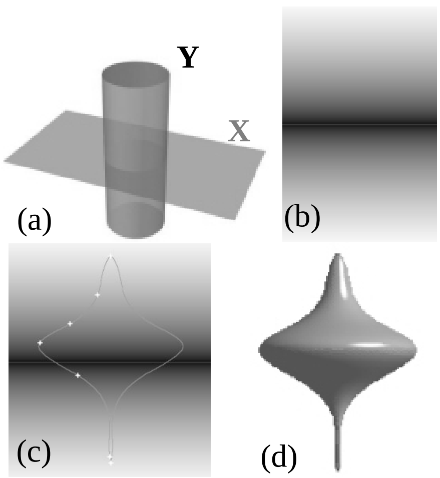


Figure 16: (a) The abscissa is defined by iso-value plane surfaces (direction) and the ordinate by cylindrical field (support), (b) representation in our 2D interface, (c) a profile is defined by the user, (d) resulting object.

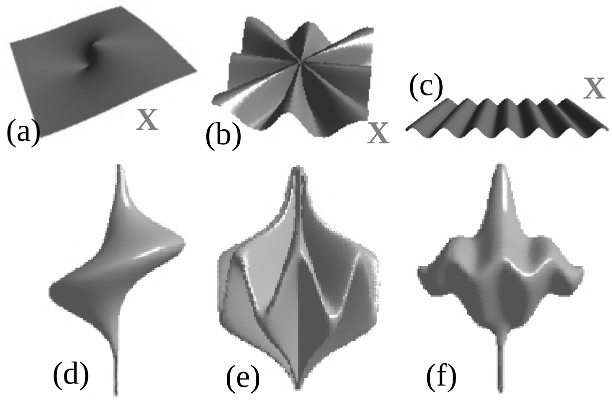


Figure 17: Different forms of direction (a,b,c) and resulting objects (d,e,f).

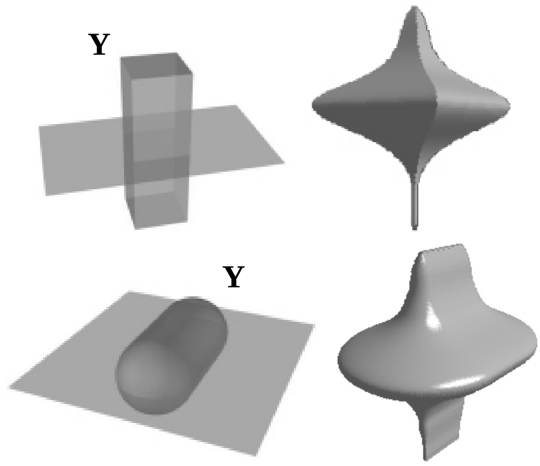


Figure 18: Different forms of support and resulting objects.

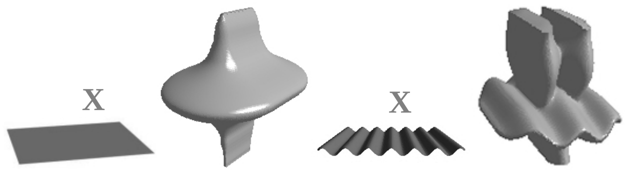


Figure 19: The plane used as the abscissa figure 18 is replaced by a corrugated field. The PROFILE extrusion become uncontrollable.

5.5. Blending operator

The 0 iso-surface of the abscissa field can not be included in the final object. We are not able to create the blend operator as expected (see section 4.4.3.). As suggested by D. Dekkers et al. [26], it is possible to include the two blended 0 iso-surfaces in the ordinate field; they propose the following expression:

$$O_1 \cup O_2 : F = \min(f_1, f_2) - f_b(|f_1 - f_2|),$$

where O_1 and O_2 are the primitive objects respectively defined by potential functions f_1 and f_2. Matter adding function f_b is a function of $R \to R$. This expression can easily be adapted to our approach:

- Function F is our G operator.
- Ordinate Y is defined by $\min(f_1,f_2)$.
- Function f_b is a function of $R \to R$. It can be replaced by our H function.
- Abscissa X is defined by $|f_1-f_2|$.

The min function generates a differential discontinuity in the ordinate field. A solution to control this discontinuity (a G^1 continuity is ensured if necessary) and allow the creation of an operator of "almost" free-form blending controlled point-by-point by the user is presented in [27] (Figure 19(a)). We do not obtain free-form blending because of the limitation of functional profiles used to perform the transition.

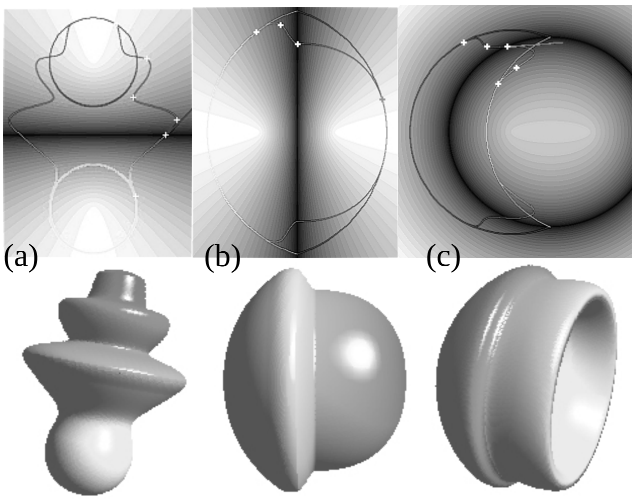


Figure 19: (a) Blending, (b) intersection and (c) difference(with soft transitions) of two spheres.

The blend operator can be seen as the union Boolean operator generated with a soft transition. Intersection and difference operators with point-by-point controlled soft transition are also presented in [27] (Figure 19(b,c)).

Also defined with a matter adding function, operators on R-functions include the two blended 0 iso-surfaces into their expression without generated differential discontinuity [23]. It could be very interesting to instantiate the ordinate field by equation 1 (see section 3), to avoid the discontinuity generated by the min function. Studies have to be done in that way, specially to verify the regular variations of the field produced, and to define the new properties of the function f_b.

6. Visualization

3D shapes and 2D sections are rendered using octrees. The voxelization method is based on interval arithmetic and a point is visualized for each vertex of voxels [18]. The computational time grows widely with the complexity of the potential function. This method is unsuited to interactive modeling but it has the advantage of giving a precise visualization of most of the implicit surfaces. This justifies the visualization method used to validate our model. Specific research will have to be done to propose a probably less precise but faster rendering method. The color variation at the blend level is the sum the colors of the blended objects balanced with their potential values. The weight value is 1 if the potential equals 0 and 0 if the potential is greater than or equal to the potential at the blend intersection with the other object. It is inversely proportional to variation of the potential value along the blend.

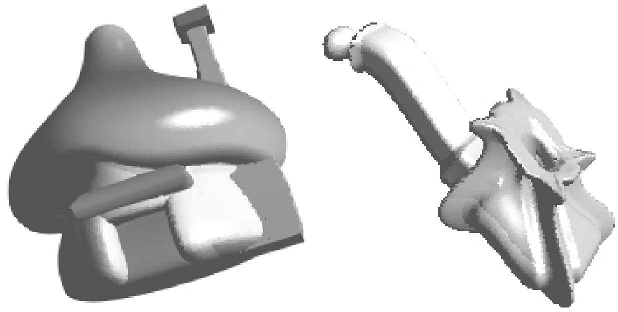


Figure 20: Octrees 256×256×256 of Smurf's house and the Hobbit's pipe visualized with OpenGL.

7. Conclusion

Extruding profiles in implicit extrusion fields allow us to introduce precise control of "almost" free-form blending. Through the instantiation of implicit space coordinates, we propose tools which allow sculpting on or around a simple surface and the creation of original extrusion objects. Nevertheless, the use of potential fields as coordinates of a 2D space represents an abstraction level for the user, and suitable interfaces have to be studied.

Another constraint of interactive modeling is fast surface visualization. We have not yet explored this line of investigations. For example, splines create bounded modifications on the primitives. Algorithms computing only these modifications in 2D/3D visualization structures can increase interactivity.

Using free-form curves is an exciting perspective to increase our model efficiency. Implicitization of parametric curves [28,29], projection of a 3D shape onto plane [30] and combination with soft transition of 2D implicit curves [31] are different possibilities to explore.

Further studies remain to be done to find the limits of implicit extrusion fields. If free form PROLILE and interactive visualization are proposed, it will be important to explore different modeling processes and interfaces. Approaches like generative modeling [32] and models such as implicit sweep objects [12,15] will contribute to generate a valid basis for comparison and inspiration.

Acknowledgments

We specially thank Pr. K. Van Overveld and his group for access to their work, Pr M.P. Cani, Pr C. Schlick, Pr L. Amodei and SMI'2001 reviewers for their interest in this work and their excellent remarks and suggestions.

References

[1] B. Wyvill, A. Guy and E. Galin. Extending the CSG tree – warping, blending and Boolean operations in an implicit surface modeling system. Proc. of Implicit Surfaces '98, pp. 128-136, 1998.

[2] A. Sherstyuck. Interactive shape design with convolution surfaces. Shape Modeling International '99, 56-65, 1999.

[3] J.F. Blinn. A generalization of algebraic surface drawing. ACM Transaction on Graphics, **1**(3):235-256, July 1982.

[4] G. Wyvill, C. McPheeters and G. Wyvill. Data structure for soft objects. The visual computer, **2**(4):227-234, February 1986.

[5] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakara and K. Omura. Object modeling by distribution functions. Electronics Communications, **J68-D**(4):718-725, 1985 in Japanese.

[6] J. Bloomenthal and B. Wyvill. Interactive techniques for implicit modeling. Computer Graphics (proc. Of SIGGRAPH '90), **24**(2):109-116, 1990.

[7] B.A. Payne and A.W. Toga. Distance field manipulation of surface models. IEEE Computer Graphics and Applications, **12**(1):65-71, 1992.

[8] B. Crespin, P. Guitton and C. Schlick. Solving the "nearest point on object" problem using recursive subdivision. AFIG '97, pp 193-204, 1997.

[9] B. Wyvill and G. Wyvill. Fields functions for implicit surfaces. The Visual Computer, **5**:227-237, 1986.

[10] R. Parent and K. Singh. Polyhedral shapes as general implicit surface primitives. OSU-CIS Technical Report, OSU-CISRC-5/94-TR24, 1994.

[11] C. Blanc and C. Schlick. Extended field functions for soft objects. Proc. Of Implicit Surfaces '95, pp 21-32, 1995.

[12] B. Crespin, C. Blanc and C. Schlick Implicit sweep objects. Eurographics '96, **15**(3):165-174, 1996

[13] A.H. Barr. Superquadrics and angle-preserving transformations. IEEE Computer Graphics and Applications, **1**(1):11-23, 1981.

[14] C. Blanc and C. Schlick. Ratioquadrics: an alternative model for superquadrics. The Visual Computer, **12**:420-428, 1996.

[15] C. Grimm. Implicit generalized cylinders using profile curves. Proc. of Implicit Surfaces '99, 1999.

[16] C. Hoffmann and J. Hopcroft. Automatic surface generation in computer aided design. The Visual Computer, **1**:92-100, 1985.

[17] A.P. Rockwood. The displacement method for implicit blending surfaces in solid models. ACM Transaction on Graphics, **8**(4):279-297, 1989.

[18] N. Stolte. Espaces discrets de haute résolutions : une nouvelle approche pour la modélisation et la synthèse d'images réalistes. PhD thesis, Paul Sabatier University, Toulouse, France, 1996.

[19] S. Murakami and H. Ichihara. On a 3-D display method by metaball technique. Electronics Communication Conference '87, **J70-D**(8):1607-1615, 1987 in Japanese.

[20] M.P. Gascuel. An implicit formulation for precise contact modeling between flexible solids. Computer Graphics (proc. SIGGRAPH '93), pp 313-320, 1993.

[21] Z. Zacic-Alesic and B. Wyvill. Controlled blending of procedural implicit surfaces. Proc. Graphics Interface 1991, pp 236-245, 1991.

[22] L. Barthe, V. Gaildrat and R. Caubet. Combining implicit surfaces with soft blending in a CSG tree. Proc. of CSG '98, pp 17-31, 1998.

[23] A.A. Pasko and V.V. Savchenko. Blending operations for the functionally based constructive geometry. Proc. of CSG '94, pp 151-161, 1994.

[24] A. Pasko, V. Adzhiev, A. Sourin and V. Savchenko. Function representation in geometric modeling: Concepts, implementation and applications. The Visual Computer, **8**(2), pp 429-446, 1995.

[25] C. de Boor. A practical guide to spline. Applied Mathematical Sciences, **27**:156-162, 1978.

[26] D. Dekkers, K. van Overveld and R. Golsteijn. Combining CSG modeling with soft blending using Lipschitz-based implicit surfaces. Technical Report, Eindhoven University of Technology, Computer Graphics Group, 1997.

[27] L. Barthe, V. Gaildrat and R. Caubet. Implicit extrusion fields. CISST '2000, pp 75-81, 1999.

[28] T.W. Sederberg, D.C. Anderson and R.N. Goldberg. Implicit representation of parametric curves and surfaces. Computer Vision, Graphics, and Image Processing, **28**(1):72-84, 1984.

[29] J. Berchtold and A. Bowyer. Bézier surfaces in set-theoretic geometric modelling. Proc. of CSG '98, 1998.

[30] K.T. Miura, A.A. Pasko and V.V. Savchenko. Parametric patches and volumes in function representation of geometric solids. Proc. of CSG '96, pp 217-231, 1996.

[31] A.A. Pasko, A.V. Savchenko and V.V. Savchenko. Implicit curved polygons. Technical Report 96-1-004, University of Aizu, Japan, 1996.

[32] J.M. Snyder and J.T. Kajiya. Generative modeling: A symbolic System for geometric modeling. Computer Graphics (proc. of SIGGRAPH '92). **26**(2): 369-378, July 1992.