Shape Modelling with Skeleton based Implicit Primitives

Dominique Rigaudière, Gilles Gesquière, Dominique Faudot

Equipe Synthèse d'Images – Le2i

Université de Bourgogne – Dijon - France

Abstract

Implicit surfaces defined by density functions are often chosen to model different kind of objects with property of soft blending. In spite of their qualities, the use of such primitives is not really intuitive. That is why we present in this paper a new description of implicit primitives to skin a set of skeletons made of different kind of basic elements. The user defines the set of skeletons and the way he want to skin them, so the implicit objects are created automatically.

Keywords: *Skeleton, Implicit Primitives, Modelling.*

1. INTRODUCTION

Needs of computers graphics have involved the use of different descriptions of objects. An interesting model is based on implicit surfaces which bring solutions to the drawbacks given by the use of parametric or polyhedral surface representation: in fact for these two representations fusion or blending between objects are not easy to define [1]. So implicit surfaces seems to be a good model to represent different kind of objects.

The choice of skinned skeletons with implicit surfaces is driving by the need of soft blending between several objects. However implicit surfaces are a well studied model, researches are currently very active [2]. Methods allow creating complex shapes without skeletons [3], [4]. In our case we have built our model upon the implicit surfaces based on density functions and skeleton.

An implicit surface is defined by the set of points M of space which density F_i is equal to a threshold T :

$$Surface = \left\{ M \in \Re^3 / F_i(M) = T \right\}$$

Since this first inverse exponential density function [5] others have been studied [6], [7], [8]. An interesting property of this model is to allow a simple formulation of blending between several implicit objects. The resulting equipotential function F is defined by the sum of the density functions of the n blobs :

$$F(M) = \sum_i F_i(M)$$

We use here the sum of each contribution in space. This can be generalised easily with other kind of operations, which can be found in the constructive solid geometric models [9] [10].

In our model the density function F_i is defined as the composition of two functions: a potential function $f_i : \Re^+ \to \Re$ and a distance function $d : \Re^3 \to \Re^+$, normalised by the radius of influence R_i :

$$F_i(M) = f_i\left(\frac{Min(d(M, Sk_i))}{R_i} \right)$$

The distance function d creates the shape around the skeleton Sk_i of the object while the function f_i manages the stiffness of the fusion between an object and its neighbours.

A skeleton Sk_i may be seen has the set of the centres of all maximal spheres included in an object [11]. For instance Blinn [5] has defined *blob* as an implicit object around a point skeleton. Therefore it is not easy to model complex shapes with several primitives without making bulges on the shape of the object (see Figure 1). So we are using different kinds of skeleton in $3D$: point, segment, curve, plane surface, curved surface or any combination of these basic objects. The skinning of such skeletons depends on the distance computation of every point to the skeleton. Distance computation to a complex skeleton is not as easy as to point skeletons. Moreover distance function may be anisotropic to increase the set of objects to build.

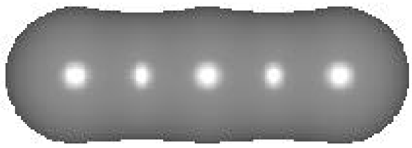

Figure 1 : Bulges on the shape after fusion of 3 blobs.

In this paper we discuss the problems set by the use of implicit surfaces based on skeletons. We will present different kinds of solutions like implicit generalised cylinders, extrusion or the use of complex skeletons. Then we suggest solutions for the use of parametric curves and surfaces as skeletons for implicit surfaces. Moreover our formulation will allow us to create anisotropic implicit surfaces.

2. PREVIOUS WORK

Several works have been done to use implicit surfaces in a modelisation process. Methods using convolution of an implicit function on a kernel (the object's skeleton) [12] are not affected by the bulges' problem but are not really intuitive to use and are expensive in time computing. Grimm [13] uses the notion of profile curve but the expression of blending in her model is not quite simple.

Another way is to modify the distance function to obtain more complex objects [8]. The density depends on the localisation around the skeleton. Only star shaped objects may be produced. This method has been generalised in [14]. Profile curves are defined to translate, rotate and sweep the implicit basic shapes. But it is very difficult to imagine object produced using a given profile. Moreover implicit shapes defined in [14] are not general enough. For instance we do not find plane skeletons.

In [15] the authors chose to use the notion of complex skeletons. They define an anisotropic function, which allows skinning all kinds of skeletons. Nevertheless the density function is still complex and the generalisation to parametric surfaces was not realised. We would like to perform this generalisation. That is why we will present in the following a study of skeletons with their skinning with anisotropic implicit surfaces.

3. CURVE SKELETONS

As seen previously the different kinds of shapes of skeletons are points, curves, and surfaces. With curves some problems arise [15] as distance computation, which is not as simple as distance between two points. Before studying curve skeletons we will introduce the problem with segment skeletons.

3.1 Segment skeletons

Distance from a point $P(x,y,z)$ to a parametric segment skeleton is computed using its orthogonal projection Q on the segment. Then the norm of the vector $\overrightarrow{PQ}$ gives the minimal distance from the point to the skeleton. So we obtain an implicit tube without bulges on its shape but whose extremities are half spheres (see Figure 2).

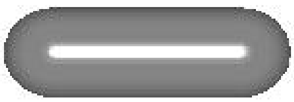

Figure 2 : Implicit surface on segment skeleton.

Therefore if the artist using this tool wants to represent an implicit cone or for instance the tail of a mouse this model is not compliant enough. So to enlarge the set of shapes which can be represented with implicit surfaces we have to extend our model to parametric curves in a way to use them as skeletons of anisotropic implicit surfaces.

3.2 Creation of anisotropic implicit surfaces on segment skeletons

The goal is to manage the density in space to obtain different kind of shapes. In fact the designer would like to represent an implicit cone or a snake for instance. For this objects the distance from the skeleton to the boundary of the surface is not constant. Current models [14], [8] could be used for this work. But these models are not based on complex skeletons. They use anisotropic distances from a point skeleton and create anisotropic shapes depending on angles around the point. In our model we have chosen to modify the distance function. So we use the Euclidean distance no longer but an anisotropic one. The resulting density function becomes :

$$F_i(M)=f_i\left(\frac{Min(d(M,Sk_i))def(M,Sk_i)}{R_i}\right)$$

The deformation function $def(M,Sk_i)$ depends on the parameter t of the orthogonal projection on the segment. As we can see on Figure 3.b the shape of the object is modified from its original shape (Figure 2) by using a normalised squared cosine function on the parameter of the segment (see Figure 3.a).

Another kind of cylinder can be obtained using this method such cylindrical implicit shape with flat extremities (see Figure 4). We can also notice that the new definition of these objects still allows blending between those (Figure 5).

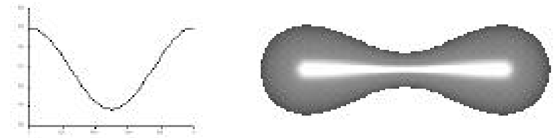

Figure 3.a and 3.b : The deformation function and the anisotropic implicit surface on a segment skeleton.

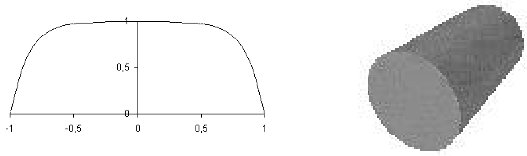

Figure 4 : Anisotropic implicit cylinder and its deformation function.

3.3 Parametric curves as skeletons

To build a more complex shape like a curve for instance the first idea is to use more than one skinned segment skeleton. But the modelling of a curve whose skeleton is composed of segments will give place to an object with sudden changes of continuity in the shape. This problem is solved by the skinning of curve skeletons.

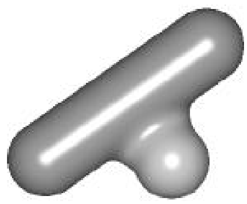

Figure 5 : Fusion between a skinned segment skeleton and a point skeleton.

The main problem to be solved is to compute the distance from the skeleton to the point of space where we want to evaluate the density. In our model we have used Bezier curves [16] as skeletons but the computation of the distance can be extended to another parametric curves.

To compute the distance from a point $P(x,y,z)$ to the curve, we have to search for the orthogonal projection $Q(t)$ of P on the curve. This orthogonal projection satisfies the equation :

$$\frac{\overrightarrow{dQ(t)}}{dt}\overrightarrow{(P-Q(t))}=0$$

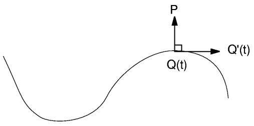

Figure 6 : Orthogonal projection on parametric curve.

The distance used to evaluate the density function is the norm of the vector $\overrightarrow{PQ}$. The orthogonal projection gives us the *t* parameter too. If a point has more than one orthogonal projection on the curve (as illustrated in Figure 7) we obtain a set of values for the parameter *t*. We choose *t* corresponding to the minimal distance. A problem remains if many distances are equal for several values of the parameter *t*. We cannot automatically choose the correct value of *t*. The user has to specify what he wants himself.

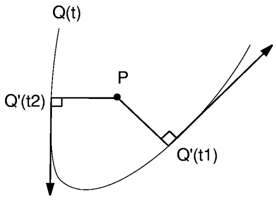

Figure 7 : The non-unicity problem for orthogonal projection.

Several numerical methods have been studied to find the solutions of the distance computation problem [17] but convergence is not always guaranteed. Furthermore these methods cannot find all the roots of the problem and are generally expensive. So we have chosen to use an analytical solution halfway between those.

3.4 The computation of the distance

Let *C(t)* a parametric curve :

$$C(t)=\begin{pmatrix} x \\ y \\ z \end{pmatrix}=\begin{pmatrix} a_0 & a_1 & a_2 & \ldots & a_m \\ b_0 & b_1 & b_2 & \ldots & b_m \\ c_0 & c_1 & c_2 & \ldots & c_m \end{pmatrix}\begin{pmatrix} 1 \\ t \\ t^2 \\ \ldots \\ t^m \end{pmatrix}$$

As seen previously the orthogonal projections of a point *P(x,y,z)* on a curve *C(t)* are given by solving the following equation :

$$\overrightarrow{C'(t)}\overrightarrow{(C(t)-P)}=0$$

Using the previous notations solutions are given by solving :

$$\sum_{i=1}^{m}(ia_i t^{i-1})(\sum_{i=0}^{m}(a_i t^i)-x_P)+$$
$$\sum_{i=1}^{m}(ib_i t^{i-1})(\sum_{i=0}^{m}(b_i t^i)-y_P)+$$
$$\sum_{i=1}^{m}(ic_i t^{i-1})(\sum_{i=0}^{m}(c_i t^i)-z_P)=0$$

t is found computing the roots of this equation. To extract roots of a polynomial expression of degree unspecified its interval of values is split recursively into sub-intervals until the function is monotonous. Then we can find the intervals of the roots of the first expression using the latest roots backward. If the interval is bounded we find the root using a dichotomy method else we find a bounded interval in which the root will be and use the dichotomy. We can see a Bezier skeleton and its skin as an implicit surface in Figure 8.a and 8.b.

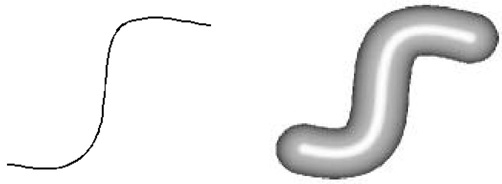

Figure 8.a and 8.b : Skinning of a curve skeleton.

3.5 Curve skeletons for anisotropic distance function

As for segment skeletons we can use anisotropic implicit surfaces to skin curve skeletons. We apply a deformation function, which depends on the parameter *t* of the orthogonal projection of the point *P* on the curve. This function takes its values on the definition set of *t*, often [0,1]. As we can see on Figures 9 the distance function may be as different as wanted to extend the number of feasible shapes. Furthermore the use of anisotropic distances does not alter the fusion between different objects as seen in Figure 10.

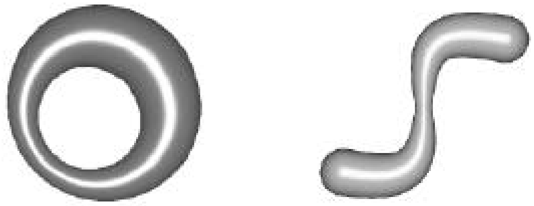

Figure 9 : Anisotropic implicit surfaces on Bezier curves.

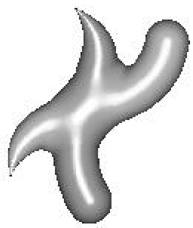

Figure 10 : Fusion between two anisotropic implicit surfaces around Bezier curves.

4. EXTENSION TO PARAMETRIC SURFACES

Skinning parametric surfaces follows the same method than parametric curves. The main difficulty is to evaluate the distance from the surface to a point of space. In fact the distance from a point *P* to a parametric surface *S(u,v)* is computed solving:

$$(S-P)\times\left(\frac{\partial S}{\partial u}\times\frac{\partial S}{\partial v}\right)=\vec{0}$$

However this equation is quite simple to write, its solution needs to solve a high degree equation with two unknown factors that increase the computation cost. Another way is solving the system of equations :

$$\begin{cases} (S-P)\dfrac{\partial S}{\partial u}=0 \\ (S-P)\dfrac{\partial S}{\partial v}=0 \end{cases}$$

This system means that scalar product of derivatives in u and v and the vector from the considerate point in space to its orthogonal projection on the surface are equal to zero.

But this method does not simplify the computation of the distance since we have to solve a non-linear system of two equations. Johnson and Cohen [18] have shown that these analytical methods for surfaces do not always converge and are very expensive. There is a lot of other methods to compute this distance for instance Newton's method [17]. But these methods are specific to a particular kind of surface and do not find all the orthogonal projections on the surface. So as we can have more than one orthogonal projection on the surface (Figure 11) we have to find another computation of the distance.

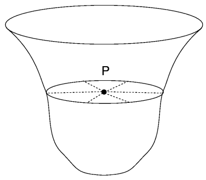

Figure 11 : More than one projection for the point P.

So we have chosen to use an approximation of the parametric surface by a succession of patches. Thus to compute the distance from a point to the surface we have just to evaluate the distances to the set of patches and to keep the minimal.

4.1 Solution for the distance computation

As say previously we subdivide the surface into a large number of patches, which approximate the surface as good as possible. The first idea is to regularly divide the surface into a determinate number of patches without taking its curvature into account. Unfortunately this regular subdivision may divide some plane parts of the surface. Then we subdivide the surface according to its local curvature that allows a better representation of the surface with a lower number of patches. That is illustrated in Figure 12 and 13.

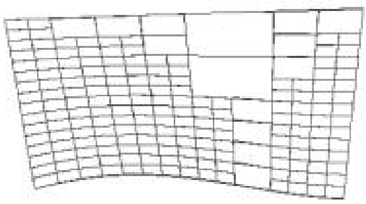

Figure 12 : Surface subdivision at depth 4.

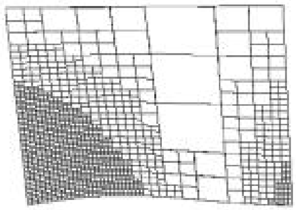

Figure 13 : Surface subdivision at depth 6.

A good way to estimate the curvature of a parametric patch is to evaluate the linearity on its sides. So we check the co-linearity of the derivatives in u and v on the sides of the patch (see Figure 14).

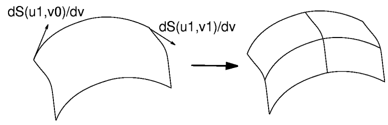

Figure 14 : Subdivision of the surface.

We compute those on the four sides of the patch. If the angle between the derivatives on the same parameter (u or v) computed on the two vertices of the same edge of the patch is greater than an epsilon value then we subdivide the patch in four sub-patches and iterate the process on the four. The process stops if each patch is as plane as needed or if the maximal depth is reached. More the depth is high, better the approximation of the surface is. This model allows us to have the best representation of the surface with the cheaper computing cost. Moreover in a modelisation way the designer can deliberately decrease the maximal depth to have a quick preview of his realisation. Finally he will increase the depth to obtain the final result.

When the surface is subdivided, we can easily compute the distance from a point to each patch and then keep the minimal to compute the density function of the implicit surface. See Figures 15 and 16 for example of implicit surfaces skinning parametric surfaces.

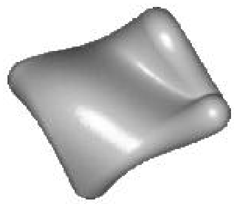

Figure 15 : Skinning a parametric surface using Murakami's function [6].

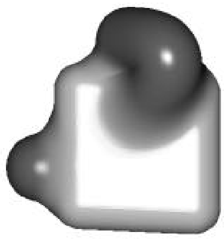

Figure 16 : Fusion of a blob, a skinned curve and a skinned surface.

4.2 Extension to any parametric surfaces : the superquadric example

Every kind of parametric surface can be used as skeleton for implicit objects. For instance, we can cover superquadric surfaces. These kinds of surfaces have been described in Barr [19].

The subdivision method described above can be used. In Figure 17 we can see two steps of the fusion of a blob and a skinned superquadric skeleton.

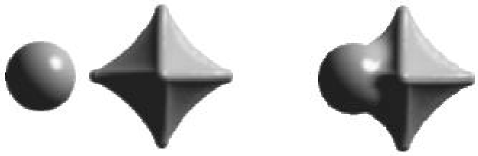

Figure 17 : Fusion between a skinned superquadric surface skeleton and a blob.

4.3 Use of surfaces as skeleton of anisotropic implicit surfaces

In a same way as curve skeletons, we can use the two parameters u and v of the surface to change the density value. In this case the deformation function of the distance takes its values on the set of definition of the parameters u, v of the parametric surface that is to say [0,1]x[0,1]. For instance we can see in Figure 18 a deformation function (cos(u)) applied on a plane.

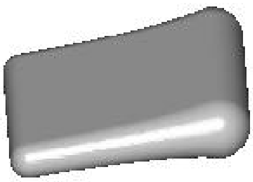

Figure 18 : Anisotropic implicit surface using a cosine deformation function.

5. CONCLUSION AND PERSPECTIVES

In this paper we have presented a method to skin all kinds of skeletons. The shape of the final object is represented using implicit surfaces that allow a soft blending between several primitives of the object. Moreover our model can be used to skin skeletons with anisotropic implicit surfaces. In this case the shape which skinning the skeleton is given by one or two functions (depending on the dimension of the skeleton). We have also presented distance computation methods, which decrease the cost of computing with a better approximation of the shape in the curved regions. Some works remain in the following of this study. For instance we could use another distance computation to generate new kinds of shapes.

6. REFERENCES

[1] Gesquière G., Faudot D., Rigaudière D., "Volume control of equipotential implicit surfaces", *Implicit Surface '99*, 4, pp. 43-49, 1999.

[2] Cani M. P., "Implicit representation in computer animation : a compared study", *Implicit Surface '99*, 4, pp. 1-8, 1999.

[3] Ferley E., "Practical volumetric sculpting", *Implicit Surface '99*, 4, 1999.

[4] Galyean T. A., Hughes J. F., "Sculpting : An interactive volumetric modeling technique", *Computer Graphics (Proceedings of SIGGRAPH '91)*, 25 (4), pp. 267-274, 1991.

[5] Blinn J. F., "A generalization of algebraic surface drawing", *ACM Transactions on Graphics*, 1 (3), pp. 235-256, 1982.

[6] Nishita T., Nakamae E., "A method for displaying metaballs by using bezier clipping", *Eurographics '94*, 13 (3), pp. 271-280, 1994.

[7] Wyvill G., McPheeters C., Wyvill B., "Data structure for soft objects", *The Visual Computer*, 2 (4), pp. 227-234, 1986.

[8] Blanc C., Schlick C., "Extended field functions for soft objects", *Implicit Surface '95*, pp. 21-32, 1995.

[9] Pasko A. A., Savchenko V. V., "Blending operations fo the functionally based constructive geometry", *CSG '94*, pp. 151-161, 1994.

[10] Ricci A., "A constructive geometry for computer graphics", *The Computer Journal*, 16 (2), pp. 157-160, 1973.

[11] O'Rourke J., Badler N., "Decomposition of three dimensional objects into spheres", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, PAMI-1 (3), pp. 295-305, 1979.

[12] Bloomenthal J., Shoemake K., "Convolution surfaces", *Computer Graphics*, 25 (4), pp. 251-256, 1991.

[13] Grimm C. M., "Implicit generalized cylinders using profile curves", *Implicit Surface '99*, 4, pp. 33-41, 1999.

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

[15] Rigaudière D., Gesquière G, Faudot D., "New implicit primitives used in reconstruction by skeletons", *WSCG '99*, 2, pp. 433-439, 1999.

[16] Bézier P., "Mathematical and practical possibilities of unisurf", *Computer Aided Geometric Design*, 1974.

[17] Engeln-Mullges G., Uhlig F., "Nonlinear equations in one variable", *Numerical algorithm in C*, pp. 13-57, 1996.

[18] Johnson D., Cohen E., "A framework for efficient minimum distance computations", *tech. rep., Departement of Computer Science, University of Utah, Salt Lake City*, 1999.

[19] Barr A., "Superquadric and angle-preserving transformations", *IEEE Computer Graphics and Applications*, pp. 11-23, 1981.

About the author

Dominique Rigaudière and Gilles Gesquière are PhD students of the computer graphics team in the *Université de Bourgogne* in France.

Dominique Faudot is professor at the *Université de Bourgogne..*

E-mail: **Dominique.Rigaudiere@u-bourgogne.fr ; Gilles.Gesquiere@u-bourgogne.fr ; Dominique.Faudot@u-bourgogne.fr - http://deboor.u-bourgogne.fr**