NEW IMPLICIT PRIMITIVES
USED IN RECONCUSTRUCTION BY SKELETONS

Dominique Rigaudière, Gilles Gesquière, Dominique Faudot

Equipe Synthèse d'Images – LE2I
Université de Bourgogne - Dijon
Dominique.Rigaudiere@u-bourgogne.fr, Gilles.Gesquiere@u-bourgogne.fr,
Dominique.Faudot@u-bourgogne.fr
http://deboor.u-bourgogne.fr

ABSTRACT

Implicit surfaces defined by density functions allow an easy modelling for different objects with a property of soft blending. So they can be used in reconstruction of objects which are defined by their skeletons. The discrete skeleton of an object can be break down in several primitives. In this article we present new implicit primitives which allow us to build a corresponding element for each skeleton's type.

Keywords : Implicit surface, anisotropic blob, skeleton, reconstruction

1. INTRODUCTION

The context of this study is the reconstruction of three-dimensional objects sampled by a set of points. The reconstruction uses two tools:

- the skeleton step that gives us a discrete skeleton of the object from the set of points
- implicit surfaces defined by density functions (that we call blobs) which allow the rebuilding of the shape of the object

Then the second step of the reconstruction is the dressing of the skeleton using blobs. This step is the purpose of this article.

Several methods using both skeleton and implicit surfaces have already been suggested. These methods use a finished set of isotropic blobs [Murak91] involving bulges on the object's surface. Bulges can be avoided using for example convolutions between blob and skeleton [Bloom91]. But convolution has an expensive cost of computation.

The object's skeleton, which is extracted using Voronoï and Delaunay's diagrams [Attal95] [Brand91] for instance, can be break down in three primitives: point, segment and plan. We have chosen to define new anisotropic implicit objects for these basis elements.

In a first time we will present a survey of implicit surfaces defined by density functions, the blending functions used then three-dimensional objects' skeletons. Next we will introduce new implicit primitives based on these elements. We will conclude this article with a possible extension for semi-continuous skeleton.

2. BASIC KNOWLEDGE

2.1. Isotropic implicit surfaces

To define an isotropic implicit surface we give to each point P in space a value computed with the density function f. The implicit surface will be defined by the set of points which density is equal to a threshold s [Blinn82].

$$Surface = \{P(x, y, z) / f(P) = s\} \qquad \text{(Eq. 1)}$$

An implicit object is defined by:

- its centre (point's co-ordinates in space).
- a density function f that gives the object's contribution at each point of space. It consist in estimate the distance from the considerate point to

the object's centre and then computing the function for this distance.

- its threshold s.

We find several density functions in literature [Blinn82] [Gascu93] [Wyvil86]. We have chosen to use the Murakami's one [Nishi94] (see eq. 2) because of its computing simplicity and its finished support:

$$f(r) = \left(1 - \left(\frac{r}{R}\right)^2\right)^2 \qquad \text{(Eq. 2)}$$

The parameter r is the distance from the studied point to the centre of the blob. R is the influence area of the blob. That is the distance from which the blob's influence is equal to zero. Isotropic implicit surface alone looks like a sphere (see fig. 1).

Fig. 1 : Blob defined by Murakami's function (rendered using the ray-tracer zeno [Hart93] [Hart96])

When several objects are in influence they can blend together to form a single object: whatever point in space the density at this point is equal to the sum for each object of their contributions (see eq. 3). We can also use negative contributions to dig another blobs.

$$Surface = \left\{ P / \sum_i f_i(P) = s \right\} \qquad \text{(Eq. 3)}$$

Thus, when for any point between two objects the sum of the influences is higher than the threshold s, the implicit objects seem to be melted (see fig. 2).

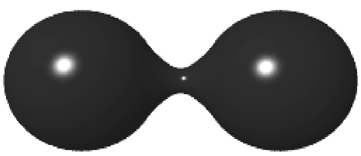

Fig. 2 : Beginning of the blending of two blobs (Murakami's function)

However the blending of blobs is usually soft, it is difficult to construct some basis shapes like cylinders (see fig. 3) or plane surfaces using a finite set of blobs.

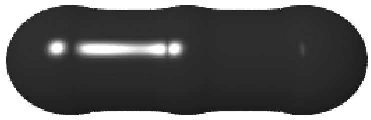

Fig. 3 : Bumps of the object's surface construct using blobs

In this paragraph we have seen the definition of implicit surfaces. Let us see now how to blend them.

2.2. Implicit surfaces and R-functions

The "traditional" fusion created by addition of the implicit primitives is not always appropriated to have an optimal reconstruction of the object. It can be interesting to obtain blending between various implicit objects without creating hollows or bumps on the surface of the object. Pasko [Pasko94] proposes to redefine the operators in order to make possible the blend of objects by introducing the concept of R-functions. An operator F is applied to two geometrical objects *f1* and *f2*. As we have seen previously in section 1, the operator F was the addition. We use two other operators: union (noted |) and intersection (noted &).

Union between two objects is defined as follow:

$$f_1 | f_2 = \frac{1}{1+\alpha}\left(f_1 + f_2 + \sqrt{f_1^2 + f_2^2 - 2\alpha f_1 f_2}\right) \qquad \text{(Eq. 4)}$$

We can see differences between union and a simple addition on fig. 4.

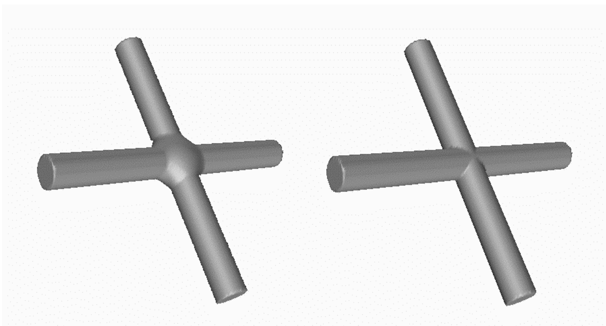

Fig. 4 : Fusion of "cylindrical blobs". On the left image the operator used is the addition, on the

right one union is used (picture computed with an implicit surface polygonalizer [Bloom94])

Intersection between two primitives is:

$$f_1 \,\&\, f_2 = \frac{1}{1+\alpha}\left(f_1 + f_2 - \sqrt{f_1^2 + f_2^2 - 2\alpha f_1 f_2}\right) \qquad \text{(Eq. 5)}$$

Fig. 5 shows an intersection between to cylindrical primitives (described in section 3.1.).

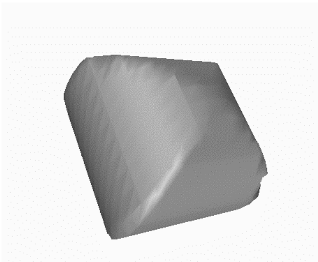

Fig. 5 : Intersection between two “cylindrical blobs”

The use of these functions of blending extends the set of realisable objects created using implicit surfaces. But in our case we want to apply these objects on skeleton. In the paragraph below we give a brief overview on this topic.

2.3. Three-dimensional objects’ skeletons

Although implicit surfaces present many interests for the modelling of shapes, it is easier to set their position if we have a description of the object with skeletons. The skeleton of an object can be defined as the locus of a point that are at equal distance from at least two points of the shape [Atta95] with the respective distance from the skeleton to the object. These two data make the skeleton of an object.

Maximal spheres [ORour79] also allow us to define a skeleton (a sphere included in an object is maximal if there is no other sphere included in the object including the first sphere). So the skeleton is the set of all maximal spheres’ centres.

For example (in the plane) the skeleton of a circle is defined by its centre. The skeleton of a rectangle is an ‘extending cross’ (see fig. 6). See fig. 7 for a three-dimensional example: the skeleton of a parallelepiped.

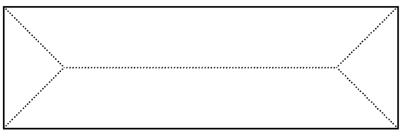

Fig. 6 : A rectangle and its skeleton (in dashed line)

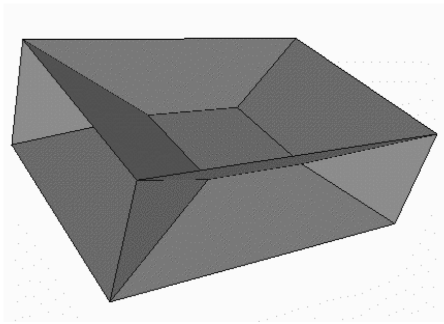

Fig. 7 : The skeleton of a box

In 3D the skeleton of a continuous object can be a point, a segment, a plane surface, a curve, a curved surface or any combination of these basic objects. We always compute the skeleton of objects sampled by a finite set of points, so it restricts the possibilities of skeletons at the points, segments and plane surfaces. We thus propose to define new implicit objects allowing reconstructing these elements simply.

3. ANISOTROPIC IMPLICIT SURFACES

We have seen before that skeletons can be represented with three different elements. In the case of point skeleton we can use isotropic implicit surfaces [Bitta95]. We now see how to define new objects for segments and plane surfaces.

3.1. Implicit surfaces on segment

We have seen that it is hard to construct cylinders using a finite set of blobs without having bumps on the object’s surface. We have to define a new object. The Murakami’s function $f(x,y,z)$ is used. In this one we set x free. So we obtain the new equation:

$$f(0, y, z) = \left(1 - \frac{y^2 + z^2}{R^2}\right)^2 = s \qquad \text{(Eq. 6)}$$

This is the equation of an infinite implicit cylinder whose skeleton is the x-axis. Density at a point of space depends on the distance from the x-axis to this point. Such function can define an implicit cylinder as we wish it to reconstruct our segment skeleton. We choose to consider that the skeleton is on the x-axis between -1 and 1 (see fig. 8).

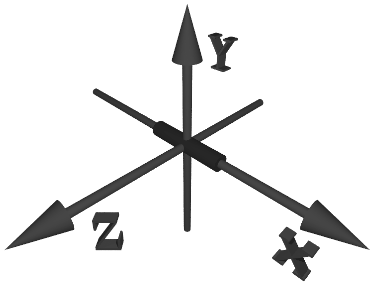

Fig. 8 : Representation of the skeleton (centred on the axis)

To obtain the wished shape, the idea is to split the density function in two sub-functions f_{radial} and f_{length}. The first one defines the radial influence. The second one defines the longitudinal influence. With f_{radial} we obtain a cylindrical shape for the object (see eq. 7 and representation fig. 9). To do this we use the Murakami's function with only y and z components:

$$f_{radial}(y,z) = \left(1 - \left(\frac{\sqrt{y^2 + z^2}}{R}\right)^2\right)^2 \qquad \text{(Eq. 7)}$$

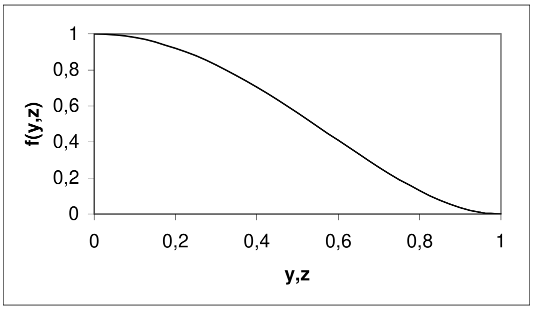

Fig. 9 : Representation of f_{radial}

The longitudinal function allows us to define the shape of the object on the third axis. This function is given by the second data of the skeleton: the distance from the skeleton to the object along the segment. For a point P on the skeleton for which the distance is maximal $f_{length}(P_x)=1$. The longitudinal function is decreasing with C^2 continuity to zero for a point of the skeleton for which the distance is zero.

For example a function found in a purely practical way to correspond to the skeleton of a cylinder can be equation 8 (see fig. 10 for its representation).

$$f_{length}(x) = -\tfrac{1}{2}x^8 - \tfrac{1}{2}x^6 + \tfrac{1}{10}x^4 - \tfrac{1}{10}x^2 + 1 \qquad \text{(Eq. 8)}$$

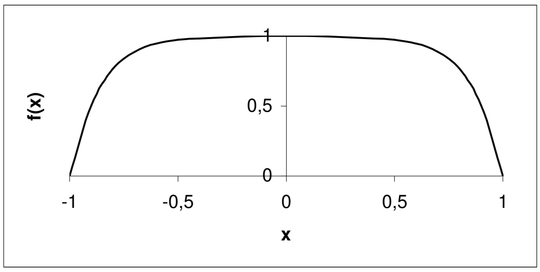

Fig. 10 : Representation of f_{length}

To obtain the resulting blob we just have to multiply the radial component with the longitudinal one (see eq. 9). The resulting object is (for the function f_{length} given in the example) a cylinder (see fig. 11) centred on the origin along the x-axis. Its radius depends on R and s.

$$f(x,y,z) = f_{radial}(y,z) \times f_{length}(x) \qquad \text{(Eq. 9)}$$

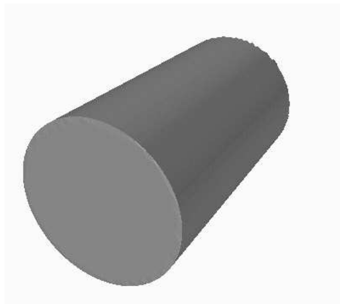

Fig. 11 : Example of "cylindrical blob"

This method, breaking down the implicit function in two sub-functions, allows us, using geometrical transformations (translations, rotations and scales), to build an object from its skeleton at whatever co-ordinates in the space, and for any scale. An example of fusion, which uses such primitives, is given in fig. 12.

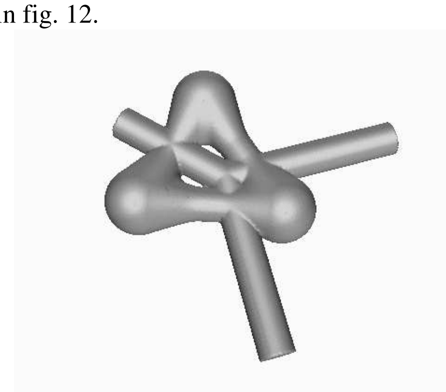

Fig. 12 : Example of fusion between blobs and "cylindrical blobs". Operations used are union and addition.

3.2. Implicit surfaces for plane surfaces

The second kind of skeleton is plane surface. As for cylinders we can not reconstruct such surface using a finite set of isotropic blobs (or "cylindrical blobs"). We extend the primitive proposed previously to define a model for plane implicit surfaces. We split the function into three sub-functions f_{normal}, f_{length} and f_{width}. The first one allows to define the influence on y component (see eq. 10).

$$f(0,y,0) = \left(1 - \frac{y^2}{R^2}\right)^2 = f_{normal}(y) \qquad \text{(Eq. 10)}$$

This function defines a implicit plane whose skeleton is the plane which equation is $y=0$. To restrict this plane we use two another attenuation functions f_{width} and f_{lenght}. Four coplanar points that will be the extremities of the implicit primitive (see fig. 13) define the skeleton. So we can reconstruct any kind of plane skeleton.

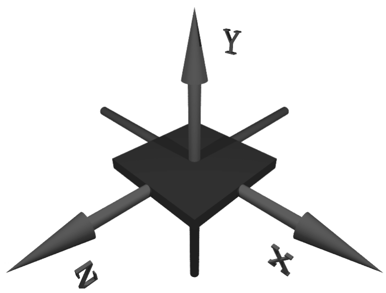

Fig. 13 : Position of the quadrilateral skeleton in space

For instance we can use this method to reconstruct a parallelepiped. Then the attenuation functions f_{length} and f_{width} will be the same as for the cylinder (see fig. 14).

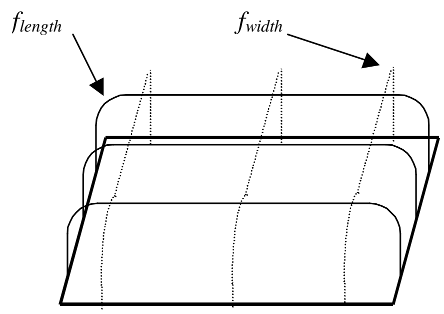

Fig. 14 : Computation of points on the implicit quadrilateral primitive using f_{length} and f_{width}

The product of the three functions gives the global density function. An example of "plane blob" is proposed in fig. 15, and an example of object reconstructed using "plane blob" is shown in fig. 16.

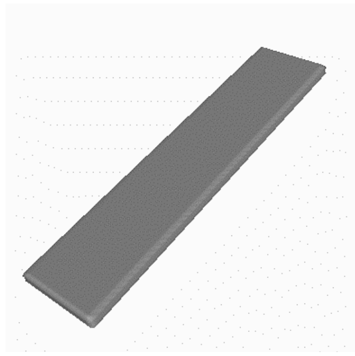

Fig. 15 : Example of "plane blob"

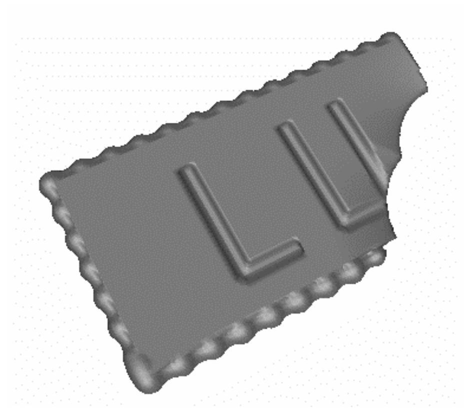

Fig. 16 : A biscuit reconstructed using a plane blob, five cylindrical blobs (negative), and several isotropic blobs

3.3. Extension to cylinder whose skeleton is not a segment

Implicit cylinders allow us to build segment with blobs. Although the blending between implicit cylinders is soft, the modelling of a curve whose skeleton is composed of segments will give place to an object with sudden changes of direction. This problem is solved by the utilisation of another implicit primitive based on the Bezier's curves [Bezie74]. For instance a cubic Bezier's curve $Q(t)$ is a polynomial curve defined by four control points P_i. It can be written as:

$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4 \qquad \text{(Eq. 12)}$$

If the skeleton is defined by such curve, we must then define an implicit function to reconstruct this skeleton. To obtain the desired shape we compute the product of a distance function (from the skeleton to the considered point) and of a function of attenuation (the same way as for the "cylindrical blobs"). The chosen distance is the smallest measurement of the segments that is joining the studied point and the points of the curve whose tangent is orthogonal with the segment.

The resulting function at a point *P* is:

$$f(P) = f_{normal}(d(P,Q(t)) \times f_{length}(d'(P,Q(t)) \qquad \text{(Eq. 13)}$$

d(P,Q(t)) is the distance from *P* to the curve *Q(t)*. *d'(P,Q(t))* give us the parameter *t* of the Bezier's curve for this given distance. With this parameter we can compute the attenuation function.

Implicit surfaces on Bezier's curves offer soft shapes and bring us an easy definition of curve by given four control points. See fig. 17 for an example of "Bezier's blob" using the same attenuation function as the "cylindrical blob" (eq. 8). The fig. 18 proposes the reconstruction of a mouse using a "Bezier's blob".

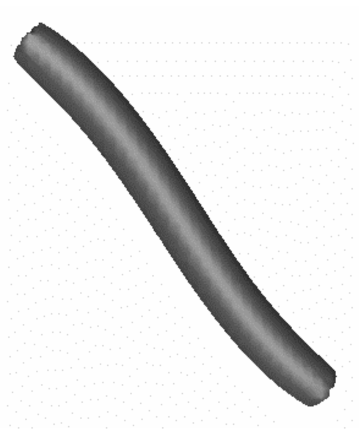

Fig. 17 : Example of "Bezier's blob"

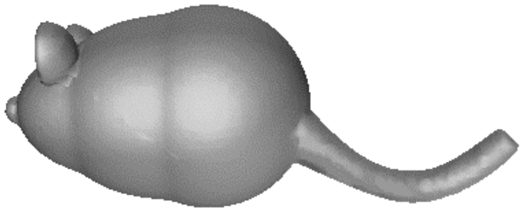

Fig. 18 : Example of reconstruction using a "Bezier's blob" (the tail of the mouse)

4. CONCLUSION AND FUTURE WORKS

In this article we define new implicit primitives with the aim of rebuilding three-dimensional objects. These primitives find their place in a general study of the rebuilding from skeletons with implicit surfaces without using geometrical deformations [Wyvil98]. The skeleton is computed using the diagrams of Voronoï and Delaunay of the set of points sampling the object [Ferle96]. It remains however many problems to be solved. The first one is the transition from discrete to semi-continuous skeleton. Indeed in order to increase the smoothing of the rebuilt object it can be interesting to gather basic elements of the skeleton. Thus it would be preferable to group segments of the skeleton together in order to form a Bezier's curve. Its final appearance as 'Bezier's blob" would be smoother than the fusion of "cylindrical blobs". It is however difficult to define precisely the parameters that decide to merge or not. The second problem is the non-desired fusion [Opala93]. In the object rebuilt some parts of the object do not have to blend together (for example the fingers of a hand should not blend). The blending tree in addition to containing the kind of fusion (addition, R-functions) must contain information on non-desired fusion too. Selection criteria on the skeleton are thus necessary to decide of fusion between the various parts of the skeleton. The search for solutions to these two problems falls under a step aiming to use the artificial intelligence as well as the combinative.

REFERENCES

[Atta95] D. Attali. "Squelettes et graphes de Voronoï 2D et 3D". Thesis, TIMC laboratory, IMAG institute, university of Grenoble I, October 1995.

[Bezie74] P. Bézier. "Mathematical and practical possibilities of UNISURF". Barnhill, R.E. and R.F. Riesenfeld, eds., Computer Aided Geometric Design, Academic Press, New York, 1974.

[Bitta95] E. Bittar, N. Tsingos and M.P. Gascuel. "Automatic reconstruction of unstructured 3D data : combining a medial axis and implicit surfaces". Eurographics 95, vol. 14, n° 3.

[Blinn82] J.F. Blinn. "A generalisation of algebraic surface drawing". ACM transactions on graphics, vol. 1, n° 3, pp. 235-256, July 1982.

[Bloom91] J. Bloomenthal and K. Shoemake. "Convolution surfaces". Computer graphics, vol. 25, pp. 251-256, July 1991.

[Bloom94] J. Bloomenthal. "An implicit surface polygonizer". Graphics Gems IV.

[Brand91] J. W. Brandt, V. and R. Algazy. "Continuous skeleton computation by

Voronoï diagram". CVGIP : Image understanding, vol. 55, n° 3, pp. 329-338, 1992.

[Ferle96] E. Ferley, M.P. Gascuel, and D. Attali. "Skeletal reconstruction of branching shapes". Proceedings of international workshop on visual form 1996.

[Gascu93] M.P. Gascuel. "An implicit formulation for precise contact modeling between solids". Computer Graphics Proceedings 1993, pp. 313-320.

[Hart93] J.C. Hart. "Ray tracing implicit surfaces". WSU Technical Report EECS-093-014, 1993.

[Hart96] J.C. Hart. "A geometric method for the antialiased tracing of implicit surfaces". The visual computer, pp. 527-545, December 1996.

[Murak91] S. Muraki. "Volumetric shape description of range data using blobby model". Computer graphics, pp. 227-235, July 1991.

[Nishi94] T. Nishita and E. Nakamae. "A method for displaying metaballs by using Bezier clipping". Eurographics' 94, vol. 13, n° 3, pp. 271-280, 1995.

[Opala93] A. Opalach, Schwerbel and S. Maddock. "Implicit surfaces appearance, blending and consistency". Fourth EG on animation and simulation workshop, pp. 233-245, Barcelone, 1993.

[ORour79] J. O'Rourke and N. Badler. "Decomposition of three-dimensional objects into spheres". IEEE Transactions on pattern analysis and machine intelligence, vol. PAMI-1, n° 3, pp. 295-305, July 1979.

[Pasko94] A. A. Pasko and V. V. Savchenko. "Blending operations for the functionally based constructive geometry". CSG, pp. 151-161, Winchester April 1994.

[Wyvil86] G. Wyvill, C. Mc Pheeters, and B. Wyvill. "Data structure for soft objects". The visual computer, pp. 227-234, 1986.

[Wyvil98] B. Wyvill, A. Guy, and E. Galin. "Extending the CSG tree”. Implicit surface 1998.