This paper has been published in : *Proc. Implicit Surfaces'95, (Grenoble, France), p21-32, April 1995*

and complementary variables t^2 and s : the first one characterizes the distance between the point and the source, the second one characterizes the position of the point according to the primary direction.

Obviously, a distance function $d_i(s,t)$ is symmetric around the primary direction V_i. This may appear as a serious limitation of the variety of shapes that can be produced. But non-symmetric functions can be created easily by defining a secondary vector $V_i'(a_i', b_i', c_i')$ and eventually a secondary radius r_i' which provides another distance function $d_i'(s', t')$. The global distance function $d_i(x,y,z)$ associated to P_i can then be defined by $d_i(x,y,z) = d_i(s,t) + d_i'(s',t')$ or better $d_i^2(x,y,z) = d_i^2(s,t) + {d_i'}^2(s',t')$ and this process can be generalized to any number of vectors.

4.4 Axial Distance Functions

A first example using the model defined above is what we call the *axial distance* :

$$d_i^2(s,t) = s^2 > m^2 \ ? \ \frac{t^2 + m^2 - 2sm}{(1-m)^2} \ : \ \frac{t^2+s^2}{(1-m)^2} \quad \text{where} \quad m \in [0,1] \tag{19}$$

The function depends on a *linearity factor* $m \in [0,1]$ that expresses the width/length ratio (see Figure 10). Note that the resulting object has rounded boundaries. This comes from the underlying Euclidian distance used in Equation 19. By replacing this distance by a D^n distance, squared or pinched boundaries (or any shape inbetween) can be obtained.

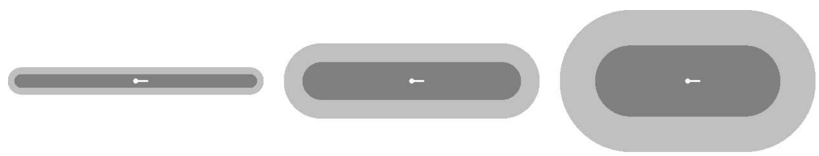

Figure 10: *Soft object with axial distance* **(a)** $m = 0.8$ **(b)** $m = 0.5$ **(c)** $m = 0.2$
(The small vector near the source indicates the primary direction)

4.5 Radial Distance Functions

If we call θ, the angle between P_iP and V_i, we can easily compute $\cos\theta = s/t$ and then define

$$d^2(s,t) = \frac{t^2}{r^2(\cos\theta)} = \frac{t^2}{r^2(s^2/t^2)} \tag{20}$$

where $r(\cos\theta)$ is a radius function expressed in polar coordinates. Using well-chosen functions for r provides a large variety of shapes for the resulting objects. For instance, Figure 11 shows three examples obtained with what we call the *clover distance* :

$$d^2(s,t) = \frac{t^2}{m + (1-m)(2\cos^2\theta - 1)^2} \quad \text{where} \quad m \in [0,1] \tag{21}$$

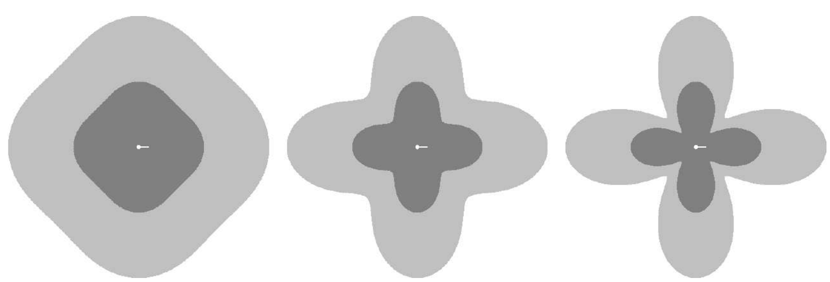

Figure 11: *Soft object with clover distance* **(a)** $m = 0.5$ **(b)** $m = 0.3$ **(c)** $m = 0.2$

5 Interactive Design

In the previous section, we have proposed a basic set of ready-to-use potential and distance functions that should be easy to implement in any existing software dealing with soft objects. In this section, we propose some ideas to extend this basic set in order to allow the user to design his own field functions. More precisely, we are going to describe the driving ideas of GOUTTE (a software environment for designing and rendering soft objects) that is currently under implementation at the Université de Bordeaux.

5.1 Design of Potential Functions

A system for interactive design of potential functions, based on Bézier and B-spline curves has been described in [7]. In our opinion such a flexibility is not really needed. Indeed there are so many imperative or desirable constraints on what represents a "good" potential function ($f(0) = 1, f(1/2) = 1, f(1) = 0, f'(d) \leq 0, f'(0) = 0, f'(1/2) = -p, f'(1) = 0$) that there are very few candiates. The function defined by Equation 13 fullfils most of these properties. Nevertheless, the modelling environment should provide a visual interface to select the hardness factor for each source by ploting the function $f(d)$.

5.2 Design of Distance Functions

Our environment is exclusively based on anisotropic point sources rather than skeleton sources. To enable free-form modelling, one has to provide the ability for the user to define the shape of the anisotropy. As said in Section 4.3, the isovalues of the distance function $d_i(s,t)$, and thus the isovalues of the field function $f_i \circ d_i(s,t)$, represent a surface of revolution that has the primary direction as the axis of symmetry. Therefore the shape of the anisotropy is controlled by the shape of the outline curve that is revolved around the axis. We propose an interactive process to define this outline curve which can be applied both to the axial distance and the radial distance model.

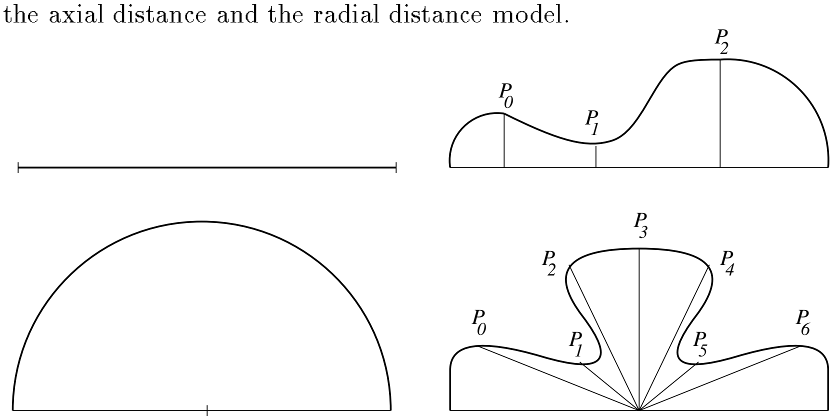

Figure 12: *Interactive design of anisotropic outlines*
Axial model **(a)** *Initial configuration* **(b)** *Final configuration with control points*
Radial model **(c)** *Initial configuration* **(d)** *Final configuration with control points*

Figure 12 shows how this process works. In the axial model, the user designs the function $m(s)$ which expresses the variation of parameter m in Equation 19. Starting from an initial straight line configuration (*i.e.* $\forall s, m(s) = 0$), the user can model a free-form outline curve by defining some points that will be used as the control points of a cardinal spline (see Figure 12, upper row). Note that the extremities of the outline are automatically rounded up, thanks to the axial model (Figure 13 illustrates some examples). In the radial model, the user designs the function $r(\cos\theta) = r(s/t)$ which expresses the variation of parameter r in Equation 21. Starting from an initial half-circle configuration (*i.e.* $\forall s/t, r(s/t) = 1$), the same process is applied to create a free-form outline curve by a cardinal spline (see Figure 12, lower row).

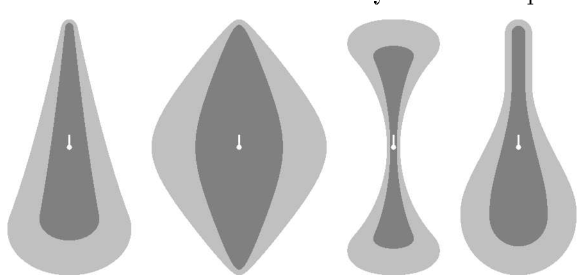

Figure 13: *Examples of user-defined axial distance functions*

5.3 Design of Soft Objects

Once the shape of the individual field functions have been defined, the anisotropic sources have to be combined together. Here again, we are going to use flatland visualization (front view, side view, top view). Instead of showing *projections* of the object, we rather display *slices* for which the cut plane is represented by a line that can be interactively moved by the user (see Figure 14). Displaying a slice implies only to freeze one coordinate and to compute the flatland image of the two other coordinates. Even with very complex objects, this is done within a second on modern workstations. After some training, such a visualization tool provides a very good overview of the shape of the object which means that only a small number of complete 3D rendering step (using polygonalization or ray-tracing) will be needed during the modelling process.

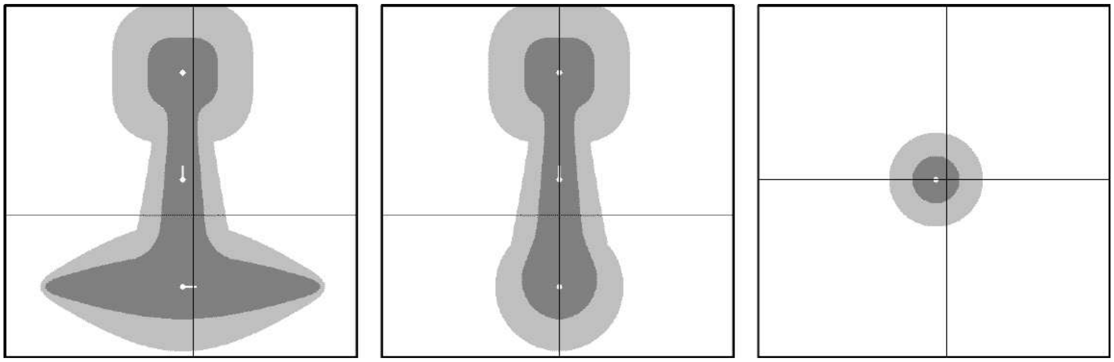

Figure 14: *Flatland visualization of slices* **(a)** *Front view* **(b)** *Side view* **(c)** *Top view (In the environment, the top view is actually located under the front view, as usual)*

To conclude, Figure 15 shows an more complex example representing a clown head defined by 8 anisotropic (5 radial and 3 axial) sources. The computation of the global field function $F(x, y, z)$ of the clown head at a given point $P(x, y, z)$ involves only 87 additions, 43 multiplications and 37 divisions for the distance functions, and 16 additions, 16 multiplications and 8 divisions for the potential functions, and no square root, no exponentation, no trigonometric function. This should totally outperform the computation cost of the skeleton model for an object with a equivalent complexity (we are currently doing exact measurements on that topic).

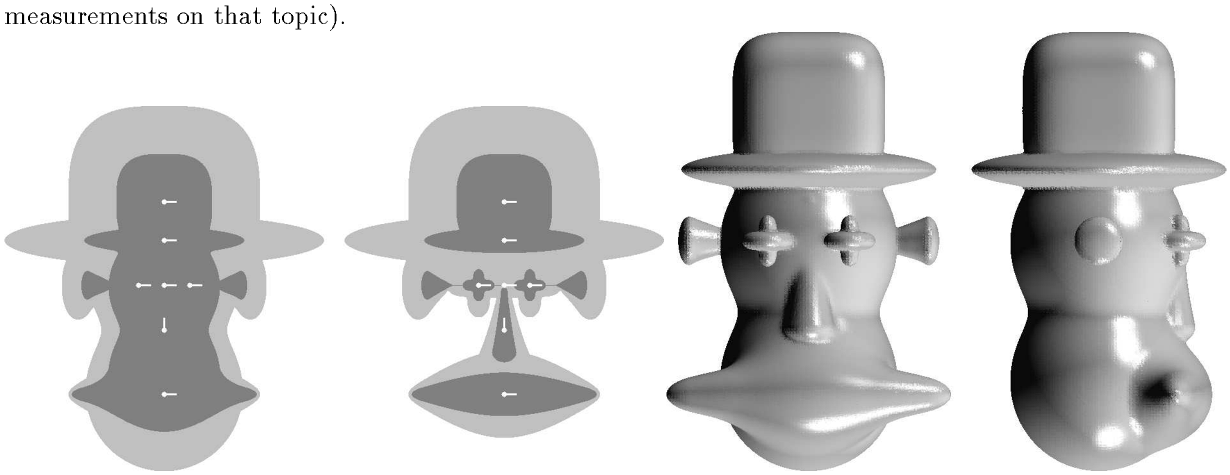

Figure 15: *Clown head defined with 8 anisotropic sources*
(a) *Flatland front view* **(b)** *Same view with one source removed*
(c) *3D front view* **(d)** *3D side view*

6 Conclusion

The topic of this paper was to present some innovative ways to define field functions for soft objects. The ideas presented here are the basic principles of GOUTTE (a software environment dealing with soft objects) that is under implementation at the Université de Bordeaux.

In a first step, we have proposed a basic set of low-cost functions (infinite or finite rational potential function with hardness factor, D^n distance function, axial and radial distance functions) that can be combined to generate a large variety of original field functions.

In a second step, we have shown that the use of anisotropic point sources combined with an interactive process for generating the shapes of the anisotropy may constitute a nice alternative to skeleton sources for designing and animating free-form objects. The main advantage of anisotropic point sources is *speed* both for the previewing (real-time flatland visualization of slices are enabled) and for the final rendering (the computation of the global field functions involves relatively few floating point operations). We are currently making more exhaustive testing to get better quantitative comparisons between the anisotropic and the skeleton approach.

Bibliography

[1] A. Barr, *Superquadrics and Angle-Preserving Transformations*, Computer Graphics & Applications, v1, p11-23, 1981.

[2] J. Blinn, *A Generalization of Algebraic Surface Drawing*, Trans. on Graphics, v1, n3, p235-256, 1982.

[3] J. Bloomenthal, B. Wyvill, *Interactive Techniques for Implicit Modeling*, Computer Graphics, v24, n2, p109-116, 1990.

[4] J. Bloomenthal, K. Shoemake, *Convolution Surfaces*, Computer Graphics, v25, n4, p251-256, 1990.

[5] J. Bloomenthal, *An Implicit Surface Polygonizer*, Graphics Gems, v4, p324-349, 1994.

[6] M-P. Gascuel, *An Implicit Formulation for Precise Contact Modeling between Flexible Solids*, Computer Graphics, v27, n4, p313-320, 1993.

[7] Z. Kacic-Alesic, B. Wyvill, *Controlled Blending of Procedural Implicit Surfaces*, Graphics Interface 91, p236-245, 1991.

[8] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakara, K. Omura, *Object Modelling by Distribution Functions*, Electronics Communications, v68D, n4, p718-725 (in japanese, cited in [9]), 1985.

[9] T. Nishita, E. Nakamae, *A Method for Displaying Metaballs by using Bézier Clipping*, Computer Graphics Forum, v13, n3, p271-280, 1994.

[10] S. Murakami, H. Ichihara, *On a 3D Display Method by Metaball Technique*, Electronics Communications, v70D, n8, p1607-1615 (in japanese, cited in [9]), 1987.

[11] G. Wyvill, C. Pheeters, B. Wyvill, *Data Structures for Soft Objects*, Visual Computer, v2, p227-234, 1986.

[12] B. Wyvill, G. Wyvill, *Field Functions for Implicit Surfaces*, Visual Computer, v5, p75-82, 1989.

[13] G. Wyvill, A. Trotman, *Ray-Tracing Soft Objects*, Proc. of CG International, p439-475, 1990.