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

Extended Field Functions for Soft Objects

Carole Blanc Christophe Schlick

LaBRI [1]*, 351 cours de la libération, 33405 Talence (FRANCE)*
[blanc|schlick]@labri.u-bordeaux.fr

Abstract : *In the field of geometric design, the generic term "soft object" embeds several implicit models (blobs, metaballs, distance surfaces, convolution surfaces) proposed over the years for modelling and animating free-form 3D objects. All these models share the property that curved surfaces are defined by computing isosurfaces of a set of potential fields. The topic of this paper is to presents some innovative ways for defining these potential fields. First, it proposes a set of ready-to-use functions and second describes an environment which allows the user to design interactively his own functions.*
Keywords : Implicit Surfaces, Soft Objects, Blobs, Metaballs, Convolution Surfaces, Field Functions.

1 Introduction

Mathematical representations that are used in the field of geometric design can be divided in two main families : parametric surfaces and implicit surfaces. The most famous members of the first family are *spline surfaces* which include two desirable features. First, they enable the modelling of free-form objects, and second, they offer an intuitive relationship between the parameters that exist in their mathematical formulation (control points, knots, weights) and the shape of the resulting surfaces. The family of implicit surfaces also contains a member which includes these two desirable features. In this model, a 3D object is created by computing the equipotential surface of a set of potential fields defined by the user. Several terms have been proposed in the literature for such a model ; perhaps the most generic one is "soft objects" [11] which can be used to qualify various models (blobs [2], metaballs [8, 10], distance surfaces [3], convolution surfaces [4]).

This paper focuses on the mathematical expression of the potential fields used in the definition of soft objects. Previous work conducted by Wyvill [11, 12, 7] has shown that, for a given set of potential sources, a great variety of shapes can be obtained by using well-chosen field functions. The work presented here follows the same direction. Its purpose is first, to propose some generic or specific field functions that can be directly implemented in any existing implicit modelling/rendering software, and second, to describe some techniques which allow to design and combine new field functions during an interactive process.

2 Soft Objects

2.1 Definition

In the basic formulation proposed by Blinn [2], soft objects are defined by a set of points $P_i(x_i, y_i, z_i)$ where each point is the source of a potential field. Each source is defined by a *field function* $F_i(x, y, z)$ that maps $\mathbb{R}^3$ to $\mathbb{R}$ (or a subset of $\mathbb{R}$). At a given point $P(x, y, z)$ of the Euclidian space, the fields of all the sources are computed and added together, leading to a global field function $F(x, y, z)$:

$$F(x,y,z) = \sum_{i=1}^{n} F_i(x,y,z) \quad (1)$$

A curved surface can then be defined from this global field function $F(x, y, z)$ by giving a threshold value T and rendering the equipotential surface S (Equation 2) for this threshold. Several solutions have been proposed to visualize such a surface on a raster device. They can be mainly divided into tessellation techniques [11, 5] and ray-tracing techniques [2, 10, 13, 9].

$$S = \{(x,y,z) \in \mathbb{R}^3 \ / \ F(x,y,z) = T\} \quad (2)$$

[1] Laboratoire Bordelais de Recherche en Informatique (*Université Bordeaux I* and *Centre National de la Recherche Scientifique*). The present work is also granted by the *Conseil Régional d'Aquitaine*.

Instead of manipulating the field functions $F_i(x,y,z)$ as a whole, it is usually easier to consider F_i as the composition of two functions, d_i (let call it the *distance function*) which maps $\mathbb{R}^3$ to $\mathbb{R}^+$ and f_i (let call it the *potential function*) which maps $\mathbb{R}^+$ to $\mathbb{R}$:

$$F_i(x,y,z) = f_i \circ d_i(x,y,z) \tag{3}$$

The function $d_i(x,y,z)$ characterizes the distance between a given point $P(x,y,z)$ and the source point $P_i(x_i,y_i,z_i)$. A usual convention [8, 11, 10] is to define d_i as a function of a user-provided parameter $r_i \in \mathbb{R}^+$ (called *radius*) which expresses the growing speed of the distance function. The most obvious solution for $d_i(x,y,z)$ is the Euclidian distance (see Equation 4) but several other functions have been proposed in the literature, especially when the potential source is not reduced to a single point (this will be discussed in Section 4).

$$d_i(x,y,z) = \frac{1}{r_i}\sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2} \tag{4}$$

The function $f_i(d)$ characterizes the potential of the source point $P_i(x_i,y_i,z_i)$ according to the distance. A nice feature [2, 7, 6] is to introduce a user-provided parameter $p_i \in \mathbb{R}$ (called *hardness* or *stiffness*) which expresses the amount of blending[2] between the individual soft objects. Many formulation for the potential function $f_i(d)$ have been proposed in the literature (this will be discussed in Section 3). Some of these functions are infinite ($\forall d \in \mathbb{R}^+, f_i(d) > 0$) and other are finite ($\forall d \geq 1, f_i(d) = 0$). Wyvill [11, 12, 7] proposed to use only "normalized" functions (for which, $f_i(1/2) = 1/2$) combined with a threshold value $T = 1/2$. This restriction enables to get predictible results because the extend of the object around the source is always $r_i/2$.

Figure 1 shows a soft object defined by four point sources and illustrates the role of the *radius* and *hardness* factors by providing different combinations of these parameters.

Figure 1: (a) *Initial configuration* **(b)** *The radius of the source at the bottom has been increased* **(c)** *The hardness of the two sources at the top has been increased*

2.2 Soft Objects in Flatland

A nice characteristic of soft objects is that their blending properties are preserved when they are restricted to a 2D space. The main advantage of these "soft objects in flatland" is that there exists a very inexpensive visualization algorithm which simply loops over the image, computes the field function for each pixel and displays a black or white information according to the threshold value. Rather than a pure boolean information, we propose the following four-state display process :

```
clear (white);
foreach pixel (x, y) {
  t = 0;
  foreach source i {
    d = d_i(x, y);
    if (d >= 1) next-source; /* only for finite potential functions */
    if (d < ε) next-pixel;   /* close to the source = white */
    t += f_i(d);
  }
```

[2] In fact, the hardness p_i is usually the slope of the potential function f_i in the neighbourhood of the threshold value.

```
    if (t > 1/2) pixel (darkgray);      /* soft object = darkgray */
    elseif (t > 0) pixel (lightgray); /* blending zone = lightgray */
}
```

For instance, Figure 2 shows the flatland picture corresponding to Figure 1. By its simplicity and its speed (an implementation including bounding boxes and other optimization tricks provides real-time rates for moderately complex objects) such a flatland tool represents a framework of choice for many theoretical and experimental studies on soft objects. In the remainder of this paper, every model of field function will be illustrated by such greyscale thresholded pictures.

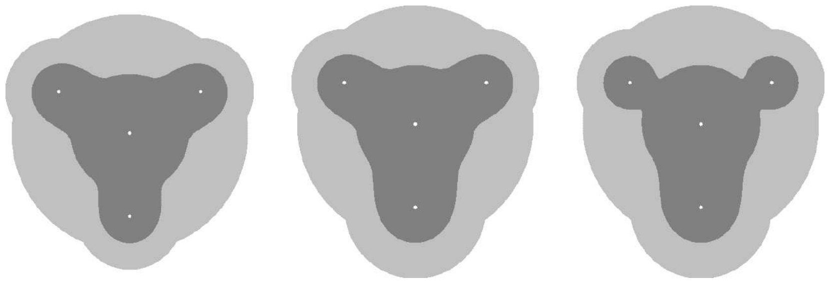

Figure 2: *Flatland soft object corresponding to Figure 1*

3 Potential Functions

In this section, we assume that a given distance function $d_i(x, y, z)$ has been chosen (Equation 4, for instance) and focus on the possible formulation of the potential function $f_i(d)$. After having recalled the expressions proposed by several authors, we propose some original low-cost alternatives. Note that every existing formulation will be rewritten in Wyvill's normalized form ($f(1/2) = 1/2$) which allows to compare the functions in a unified framework.

3.1 Infinite Potential Functions

The first potential function has been proposed by Blinn [2] and was based on a Gaussian function :

$$f(d) = \frac{1}{2}\ \exp\ (p - 4pd^2) \tag{5}$$

As said earlier, the parameter p controls the slope of the function at the threshold value $d = 1/2$ (see Figure 3a). Notice the use of d^2 in Equation 5 which avoids the expensive calculation of the square root in Equation 4. The main problem with Blinn's function (in its normalized form) is that extremely high values are reached when p increases ; this way lead to numerical instabilities. An alternative formulation based on the arctangent function has been proposed by Kacic-Alesic and Wyvill [7] :

$$f(d) = \frac{1}{2} + \frac{1}{\pi}\arctan\ (p - 2pd) \tag{6}$$

This time the values taken by the function are bounded by $[0, 1]$ (see Figure 3b). As a counterpart, the function involves the computation of a square root and an arctangent which makes it more expensive as the previous one. For that reason, we propose the following function that behaves as Equation 6 (see Figure 3c) but is even less expensive as Equation 5. The function is a piecewise rational quadratic polynomial (for compactness, we use notation from the C programming language) :

$$f(d) = (d^2 < 1/4)\ \ ?\ \ 1 - \frac{1}{2 + p - 4pd^2}\ \ :\ \ \frac{1}{2 - p + 4pd^2} \tag{7}$$

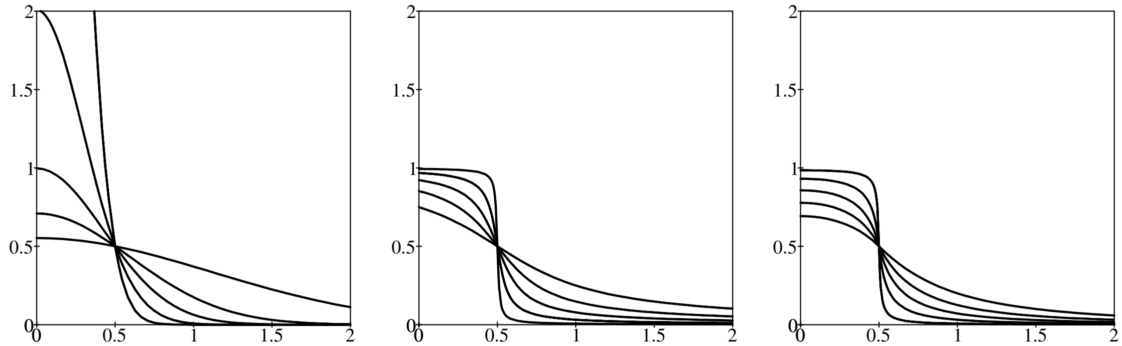

Figure 3: *Infinite potential functions* **(a)** *Blinn* **(b)** *Kacic-Wyvill* **(c)** *New model*

Figure 4 shows a simple object made of five sources, using Equation 7 with three different hardness values.

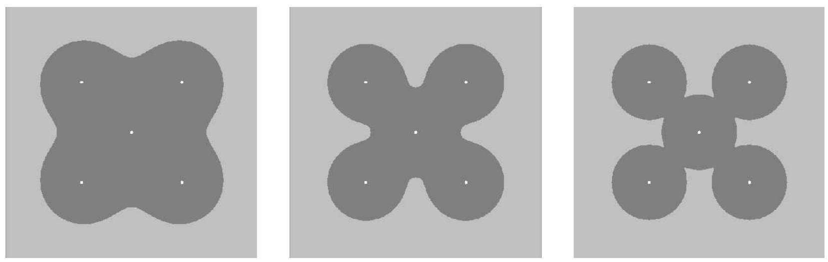

Figure 4: *Soft object with infinite fields* **(a)** *p=1* **(b)** *p=4* **(c)** *p=50*

3.2 Finite Potential Functions

A serious drawback of the previous potential functions is that they have an infinite support, which means that a point situated very far from a source point is still influenced by it. Therefore several authors have proposed finite support formulations which enable a better control of the resulting shapes. In these models, the potential of a source P_i drops to zero when the distance is larger that the radius of influence r_i (in other words, $\forall d \geq 1, f_i(d) = 0$). To avoid discontinuities in the resulting soft object, another important property required by a finite potential function is to have $f_i'(1) = 0$ and eventually $f_i''(1) = 0$.

The idea of finite potential functions was first developped by Nishimura *et al.* [8] who proposed a piecewise quadratic polynomial (as usual, we present here the function in its normalized form) :

$$f(d) = (d < 1) \ ? \ (d < \frac{1}{3}) \ ? \ \frac{4}{3} - 4d^2 \ : \ 2\,(1-d)^2 \ : \ 0 \tag{8}$$

Alternative formulations which do not need the calculation of the square root involved in Equation 4 have been proposed by Wyvill *et al.* [11] :

$$f(d) = (d^2 < 1) \ ? \ 1 - \frac{22}{9}d^2 + \frac{17}{9}d^4 - \frac{4}{9}d^6 \ : \ 0 \tag{9}$$

and by Murakama *et al.* [10] :

$$f(d) = (d^2 < 1) \ ? \ \frac{8}{9}\,(1-d^2)^2 \ : \ 0 \tag{10}$$

The plots of these three functions (see Figure 5a) are relatively close (especially on the right part where the blending actually occurs) and therefore the resulting objects are really similar. Nevertheless, none of the function includes a hardness factor and therefore the user cannot control the shape of the object with the same precision as previously. M-P. Gascuel [6] has proposed a piecewise polynomial which offers such a control of the blending :

$$f(d) = (d < 1) \ ? \ (d < 1/2) \ ? \ \frac{1}{4}\,(2 + p - 2pd) \ : \ (-2 + p + 8d - 2pd)\,(1-d)^2 \ : \ 0 \tag{11}$$

Unfortunately, for large values of the hardness parameter p, the function reaches negative values (see Figure 5b). Other polynomial or piecewise polynomial functions have been proposed (for instance, using functional Bézier or B-splines [7]) but all formulations include the same limitation : there exists a maximal value for the hardness factor. It means that one cannot create hard blending objects like the one presented on Figure 4c. To our knowledge, the only potential function with finite support and unlimited hardness factor has been proposed by Kacic-Alesic and Wyvill [7] and is based on the arctangent function (see Figure 5c) :

$$f(d) = (d < 1) \ ? \ \frac{1}{2} + \frac{1}{2}\frac{\arctan\,(p - 2pd)}{\arctan p} \ : \ 0 \qquad (12)$$

Nevertheless this function is not completely satisfying, first because of its cost (it involves the computation of a square root and an arctangent) but mainly because its derivative at $d = 1$ is not zero. Thus the resulting objects may have unwanted discontinuities, especially for small values of p.

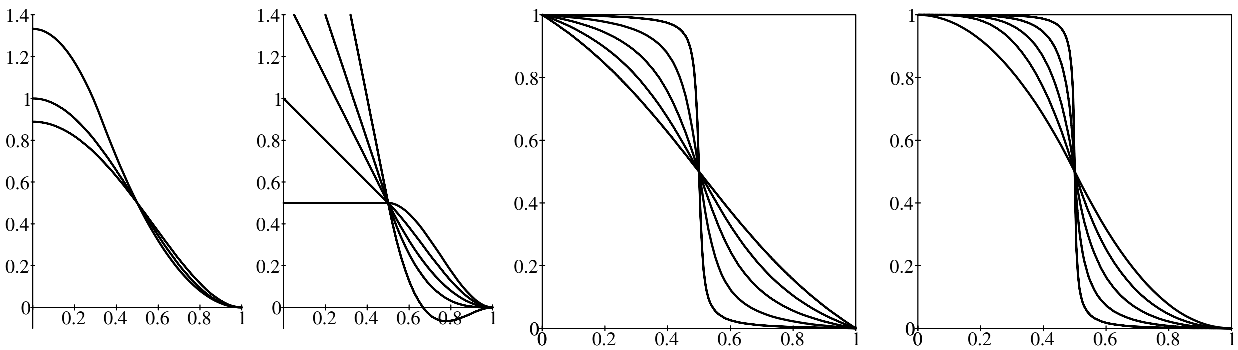

Figure 5: *Finite potential functions*
(a) *Nishimura, Wyvill and Murakami* **(b)** *Gascuel* **(c)** *Kacic and Wyvill* **(d)** *New model*

We propose here a new potential function based on a piecewise rational polynomial which includes all the desirable features that we have exhibit : it has a finite support and a finite image (more precisely, it maps $[0, 1]$ to $[0, 1]$), it is normalized and (almost) symmetric according to $1/2$ (see Figure 5d), it enables an unbounded hardness factor ($p \in \mathbb{R}^+$) and it is really inexpensive (an optimized implementation needs only 1 division, 2 multiplications and 2 additions) :

$$f(d) = (d^2 < 1) \ ? \ (d^2 < \frac{1}{4}) \ ? \ 1 - \frac{(3d^2)^2}{p + (4.5 - 4p)d^2} \ : \ \frac{(1 - d^2)^2}{0.75 - p + (1.5 + 4p)d^2} \ : \ 0 \qquad (13)$$

Figure 6 shows the same object as Figure 4 but using finite potentials provided by Equation 13 instead of infinite ones. Note that the same amount of control of the blending is enabled by the hardness factor.

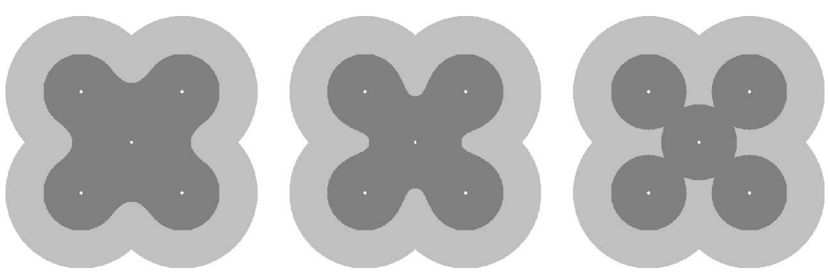

Figure 6: *Soft object with finite fields* **(a)** $p = 0$ **(b)** $p = 1$ **(c)** $p = 50$

4 Distance Functions

In this section, we assume that a given potential function $f_i(d)$ has been chosen (Equation 13, for instance) and focus on the possible formulation of the distance function $d_i(x, y, z)$. As in Section 3, after having recalled the expressions proposed by several authors, we propose some original low-cost alternatives.

4.1 Non-Euclidian Distance Functions

All the examples presented in the previous section have been created by using the Euclidian distance, also known as the *spherical distance*. An unsatisfying consequence of such a distance is a "bubble-shape" appearance of the resulting soft objects. This point has already been noticed by Blinn in his original paper [2]. To weaken this drawback, Blinn proposed to replace the spherical distance either by ellipsoidal or by superellipsoidal [1] ones. The ellipsoidal distance involves only a small overhead (*i.e.* affine transformation) compared to the spherical one, but it does not greatly extend the variety of shapes for the resulting objects. The superellipsoidal distance provides a much greater variety of shapes but unfortunately, as noted by Wyvill [12], multiple superellipsoidal sources do not blend well.

In fact, many metric spaces have been developped over the centuries in the field of affine algebra but, to our knowledge, they have never been used as alternatives to the spherical distance for soft object. Perhaps the most popular non-Euclidian distance is the D^n *distance* (which is a straightforward generalization of the Euclidian distance) :

$$d_i(x, y, z) = \frac{1}{r_i} \left(|x - x_i|^n + |y - y_i|^n + |z - z_i|^n\right)^{1/n} \tag{14}$$

At the limit case ($n \to \infty$), ones obtains the D^∞ *distance* :

$$d_i(x, y, z) = \frac{1}{r_i} \max\left(|x - x_i|, |y - y_i|, |z - z_i|\right) \tag{15}$$

Figure 7 shows the same object as Figure 6b created by using the D^n distance with $n = 1.5, n = 3$ and $n = \infty$ respectively. In fact for isolated sources, the resulting shapes are very similar to the one obtained with a superellipsoidal distance, but this time, multiple sources blend well as illustrated by the figure.

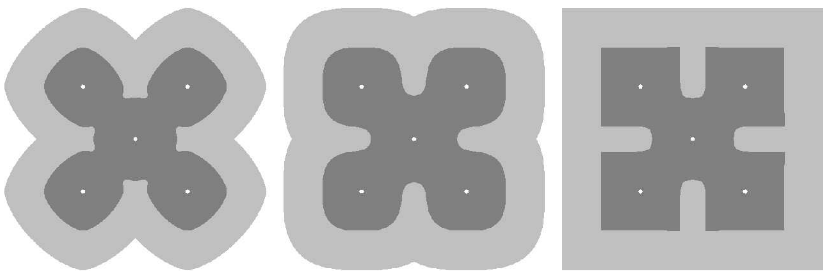

Figure 7: *Soft object with D^n distance* **(a)** $n = 1.5$ **(b)** $n = 3$ **(c)** $n = \infty$

4.2 Skeleton Distance Functions

A major extension of Blinn's basic formulation has been presented by Bloomenthal *et al.* [3, 4]. The idea of this extension is to allow more complex sources than simple points (e.g. lines, curves, polygons, polyhedras). These sources can be then considered as the *skeleton* (let us note it $\mathcal{S}$) of the soft object, and therefore the model provides a very intuitive way to create complex shapes.

A skeleton $\mathcal{S}$ contains in fact an infinite number of point sources P_i. Therefore, at a given point P, the individual field functions F_i cannot be simply added up according to Equation 1 because F would take infinite values. To avoid this problem, a clever scheme has to be used for the computation of the distance function before the application of the potential function. Three different solutions have been proposed for that, in the literature.

In the first one (let us call it the *skeleton distance model*) [3], $d(x, y, z)$ is defined as the minimal distance between $P(x, y, z)$ and the individual points $Q(u, v, w)$ of the whole skeleton :

$$d(x, y, z) = \frac{1}{r} \min_{Q \in \mathcal{S}} \sqrt{(x - u)^2 + (y - v)^2 + (z - w)^2} \tag{16}$$

In the second one (let us call it the *convolution distance model*) [4], $d(x, y, z)$ is defined as the convolution, over the whole skeleton, of all the individual distances[3] :

$$d(x, y, z) = \frac{1}{r} \int_{Q \in \mathcal{S}} \sqrt{(x - u)^2 + (y - v)^2 + (z - w)^2}\, du\, dv\, dw \tag{17}$$

[3] The convolution can also be done on the potentials rather that the distances; similar results are obtained.

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

In the third one (let us call it the *bone distance model*) [3], the skeleton is divided into bones rather than being considered as a whole. For each bone, the minimal distance (and thus the maximal potential) is computed and all these bone potentials are then added up.

The three models have their own advantages and drawbacks. Skeleton distances create unwanted discontinuities (see Figure 8a). Convolution distances provide always smooth shapes but, as a counterpart, the skeleton may not stay inside of the object when the threshold is set to high (see Figure 8b) which is not very intuitive. A solution may be to lower dramatically the threshold (see Figure 8c) but this implies to loose all the properties offered by normalized potential functions (see Section 3). Bone distances usually create unwanted bulges in the zone where the bones touch each other (see Figure 9a and its corresponding side view on Figure 9b) but, as shown in [7], this effect can be reduced by increasing the hardness factor (see Figure 9c and Figure 9d).

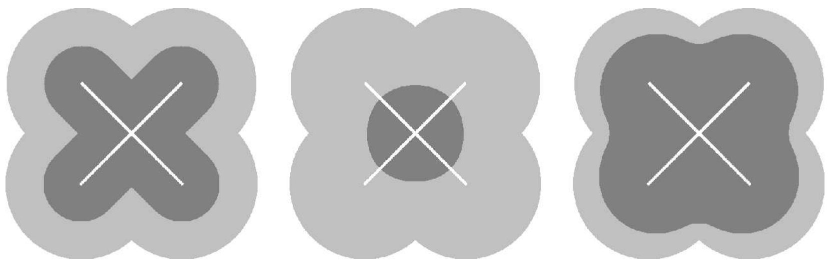

Figure 8: *Soft object with skeletons*
(a) *Skeleton distance* **(b)** *Convolution distance for* $T = 0.5$ **(c)** *Convolution distance for* $T = 0.05$

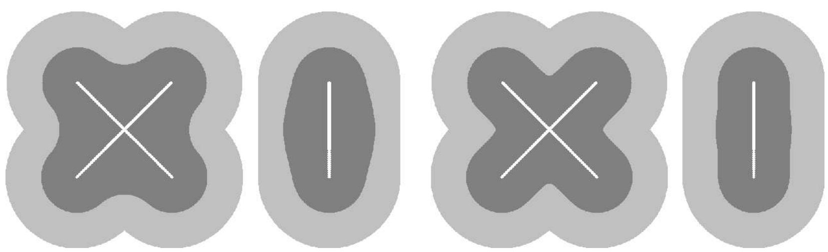

Figure 9: *Soft object with skeletons*
(a) *Bone distance with soft blending* **(b)** *Side view* **(c)** *Bone distance with hard blending* **(d)** *Side view*

But the major drawback of all the skeleton approaches is that the computation is really expensive : except for the case of very simple skeletons (*e.g.* line, circle, rectangle, disc, box) the rendering time for an object with a given number of bones is at least one order of magnitude higher than for an object with the same number of point sources. In the remainder of the paper, our purpose is to show that the use of *anisotropic source points* can be an interesting alternative to skeletons.

4.3 Anisotropic Distance Functions

Replacing the spherical distance by the D^n distance one, as we have done in Section 4.1, creates an anisotropic field when it is considered in the Euclidian space[4]. Nevertheless, one obtains a relatively poor variety of anisotropic because there is only one degree of freedom (parameter n) that can be modified by the user. For that reason, we introduce a new model of point sources which allows the user to control precisely the shape of the resulting soft objects. In this new model, each source is basically defined by 7 parameters : a position $P_i(x_i, y_i, z_i)$, a radius of influence r_i, and a primary direction $V_i(a_i, b_i, c_i)$. At a given point $P(x, y, z)$, these parameters are used to define 5 different variables :

$$u = \frac{x - x_i}{r_i} \qquad v = \frac{y - y_i}{r_i} \qquad w = \frac{z - z_i}{r_i} \qquad t^2 = u^2 + v^2 + w^2 \qquad s = u\ a_i + v\ b_i + w\ c_i \tag{18}$$

When the vector V_i is normalized ($a_i^2 + b_i^2 + c_i^2 = 1$) it is clearly insured that $t^2 \in [0, 1]$ and $s \in [-1, 1]$. All the anisotropic distance functions that we propose hereafter will be defined using this couple of bounded

[4] Of course, it is still isotropic in the metric space associated with the D^n distance.