Field Functions for Implicit Surfaces

Article *in* The Visual Computer · January 1989

DOI: 10.1007/BF01901483 · Source: DBLP

2 authors:

Brian Wyvill
University of Victoria

Geoff Wyvill
University of Otago

Abstract

The use of 3D computer generated models is a rapidly growing part of the animation industry. But the established modelling techniques, using polygons or parametric patches, are not the best to define characters which can change their shape as they move. A newer method, using iso-surfaces in a scalar field, enables us to create models that can make the dynamic shape changes seen in hand animation. We call such models *Soft Objects*

From the user's point of view, a soft object is built from primitive key objects that blend to form a compound shape. In this paper, we examine some of the problems of choosing suitable keys and introduce some new field functions that increase the range of shapes available as keys.

Key words: *soft* objects, geometric modelling, computer animation.

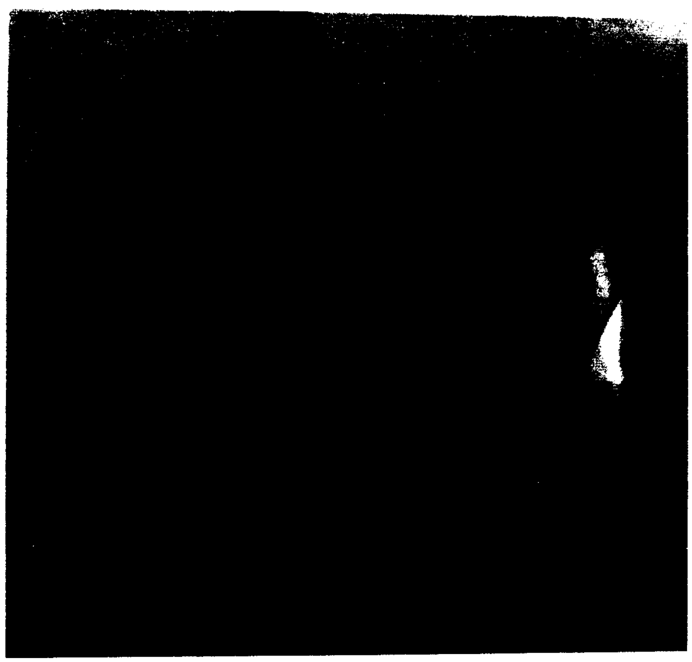

Figure 1: An iron age bloomery furnace (Camaroon). The model is constructed from five positive keys (n = 4) and four negative keys.

Introduction

The use of 3D computer generated models in the animation industry has increased dramatically over the last ten years. The animator has the problem of designing and animating 3D characters and backgrounds. The most popular primitives for building such models in commercial systems are spline surfaces (e.g. B spline patches (Alias Research) and polygon meshes (Vertigo Imagery). Primitive operations such as *extrusion* and *surfaces of revolution* aid the building of such meshes. However these techniques do not lend themselves to the creation of flexible objects such as 3D cartoon characters or the surface of a liquid. Even if polygons or patches are used as part of the rendering process, they are not usually convenient as a tool for describing 3D shapes.

A technique more suited to the representation of flexible surfaces was developed by Blinn to model constant energy surfaces in molecules. (see [Blinn, 82]). A somewhat different approach to producing blended surfaces is provided by volume modelling systems. A good technique for providing blending in such systems is given in [Middleditch, 85].

In this paper the technique of building 3D models from an iso-surface in a scalar field is referred to as the *Soft Object* or implicit surface method.

About the same time new algorithms for finding these implicit surfaces, along with field functions based on a cubic, were developed in Japan [Nishimura, 85] and in Canada [Wyvill, 86]. In the latter work the surface is sampled by uniformly subdividing space into cubic regions. Each cube can then be replaced by polygons which form an approximation to the surface. This process is referred to as *polygonisation* [Bloomenthal, 87]. Blinn rendered the surface directly from the function. Using the polygonisation technique enables surfaces to be prototyped. This is particularly important in an interactive environment where the surface is required in real time, see [Jevans 88]. By altering the size of the cubic cells the effective sampling rate is altered. Surface accuracy can thus be traded for speed for fast prototyping. Another advantage for polygonising the surface is that many manufacturers (e.g. Silicon Graphics) produce special purpose hardware with display lists, geometry pipelines and fast rendering oriented towards polygon primitives. *Soft Object* modelling can be easily integrated into these systems at the polygon level.

More recently interest has been shown by several researchers in polygonising an implicit surface. An adaptive subdivision algorithm is presented in [Von Herzen, 87]: Where the surface is changing rapidly the cubic cells are subdivided until certain criteria are satisfied. Von Herzen indentifies one major problem in his algorithm, when polygons replace adjacent cells of different size then "Cracks" between the polygons can be left in certain cases. An elegant solution to this problem is presented in [Bloomenthal 87]. After finding the cubic grid, by whatever method, each cube has to be replaced by a set of polygons. An algorithm for doing this was described in [Wyvill, 86] and a slightly more efficient algorithm given in [Lorensen 87]. A third, yet more efficient algorithm is presented in [Wyvill, 88a]. Work on implicit surfaces has led to more efficient implementation than in Blinn's original paper, although in most subsequent work a single key is used to generate a spherical surface.

In this paper we introduce a new family of field functions that lead to implicit surfaces of various shapes which have so far proved to be more useful for modelling than previous field functions. We also show how a single model can be built using more than one field function, giving rise to some models that would be extremely difficult to build by other techniques. We show that the new field functions are useful for modelling and we examine some of the outstanding problems in using this technique. Figure 1 shows an example of a model built from several different field functions. It represents an iron age bloomery furnace which was reconstructed in cooperation with archaeologist Nick David at the University of Calgary. The real structure is built from clay and the computer model from *Soft Objects* provides a simple and appropriate model.

Figure 2

Problems with Soft Object Techniques for Modelling

Over the last three years we have gained some experience with using the *soft object* for modelling. The major problems are as follows:

a) Primitive Shape: A modeller's view of the Soft Object technique is that each key is a building block of some primitive shape, such as a sphere. Primitives can be made to blend by placing keys close together. A spherical primitive has severe limitations when flat surfaces are desired. A large number of primitives are required to produce a given effect. Figure 2 shows the letters SFOT made from primitive spheres. The letter S has insufficient primitives (22 spheres) to describe the constantly changing curve and the underlying spheres may be observed as undesirable bulges. The T (22 spheres) requires fewer sphere primitives to manufacture the cylindrical vertical section so that it appears to be smooth. Many objects are best built from a variety of primitives rather than relying on one shape. A technique for doing this is described in this paper.

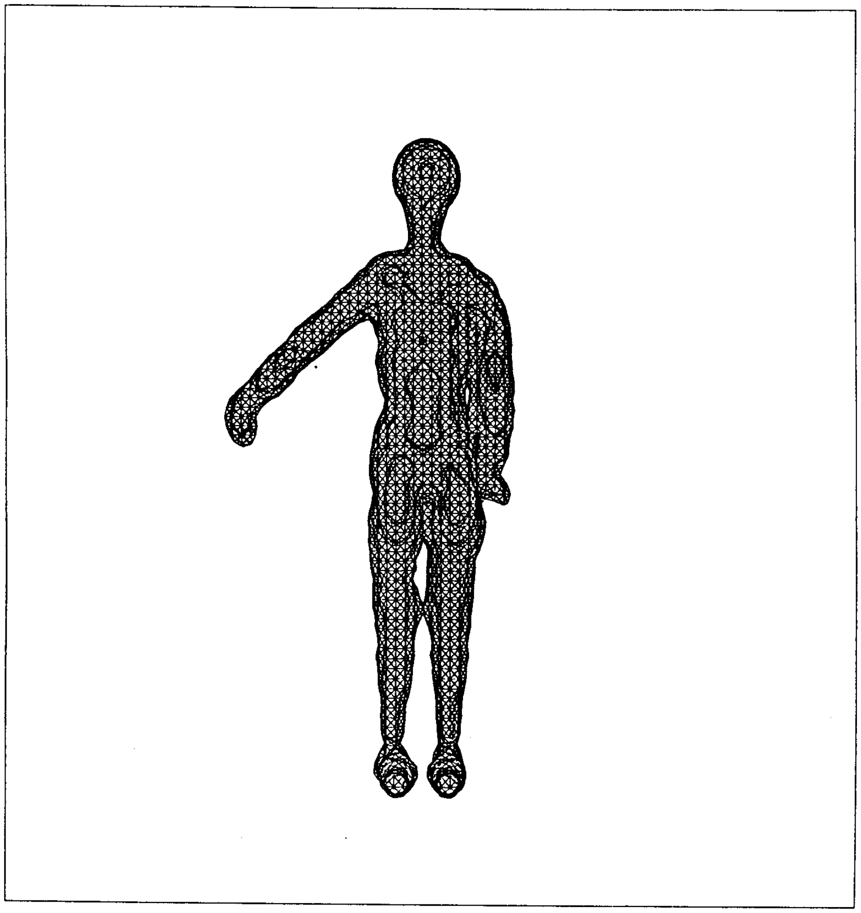

Figure 3

b) Undesirable Blending: A common problem with *Soft Objects* is that it is often required to blend certain groups of objects but not others. Consider the crude man in Figure 3. When the arms bend they should not blend with the body although the upper arm should blend with the shoulder, as shown by the right arm of the figure. The left arm of the crude man blends both with the shoulder and unfortunately with the hips. The knees also blend undesirably. A technique for solving a sub-set of such blending problems is presented in this paper.

c) Under Sampling Effects: As a *Soft Object* model is moved through space in an animation sequence it is also moving through the 3D cubic grid from which the surface is sampled. If the grid is too large the surface will appear to ripple as it makes moves the grid. This is a form of aliasing characteristic of undersampling the surface.

Previous Field Functions

In Blinn's system, [Blinn, 82], the principal field function used was an exponential function:

$$D(x, y, z) = exp(-ar^2)$$

where r is the distance from the key point. But he also generalised the exponent for non spherical keys. The function:

$$-ar^2 = (x-xi)^2 + (y-yi)^2 + (z-zi)^2$$

is a special case of a quadric in x, y, z and he showed how this could be replaced by a general quadric to obtain a range of primitive shapes to blend. He also suggested using hyperellipsoids by allowing exponents larger than 2 in the quadric forms.

Blinn's choice of an exponential function was suggested by the electron density fields he was modelling. But it had the disadvantage that, in principle, every key point affected the field at all points in space. We [Wyvill, 86] replaced it by a cubic function of similar shape. We chose the cubic coefficients so that the field and its derivative (with respect to r) dropped to zero at a known distance R. Beyond this 'radius of influence' the field was defined to be zero. This enabled us to combine fields, efficiently and without approximation, using large numbers of key points, because the field at any point depends only on key points in that locality.

Blinn was clearly aware of the limitations of his method. He suggested using alternative decay functions but he did not include any demonstration of their effect. Replacing the r^2 by a general quadric, turns out to be a very clever generalisation. The combined function:

$$exp(Q(x, y, z))$$

where $Q(x, y, z)$ is a general quadric works very well. So does our cubic approximation:

$$C(Q(x, y, z))$$

The coefficients of the cubic, C, can be chosen to approximate the exponential over the relevant range of Q. Problems arise, however, as soon as we depart from the quadric function. This is illustrated in Figure 4. Field functions are shown as a sequence of contour lines. The one on the left is from a simple ellipsoid represented by a quadric. The one on the right is from a hyperellipse. The quadric provides a neat generalisation of the idea of radius, but the super ellipsoid does not. The result is that the function falls off far too quickly. The close contours show this. We need to be able to use arbitrary shapes for our primitives and yet keep the controlled decay of the field with distance. Our method is not as elegant as Blinn's quadrics but it is more general.

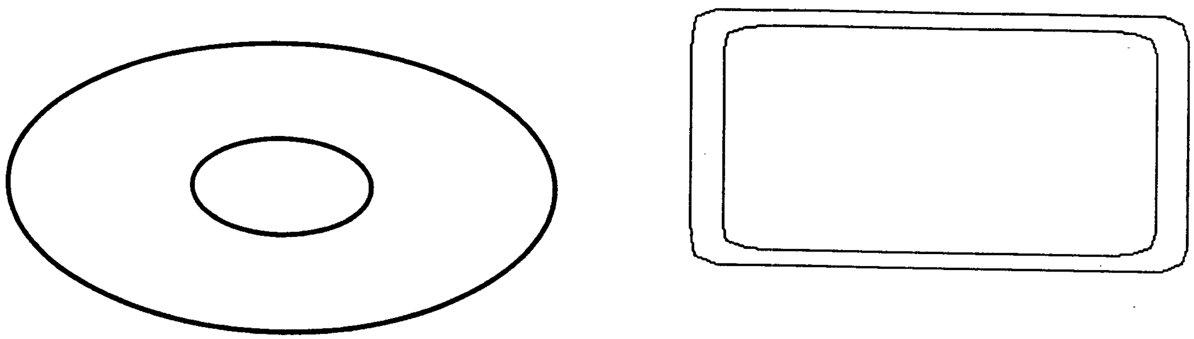

Figure 4: An ellipsoid and a hyperellipsoid are shown in cross section. The outer contour represents the limit of influence of the field function and the inner one represents the surface. The ellipsoid function produces a large region between the contours and these shapes blend well. In the smaller region between the contours of the hyperellipsoid, the field drops rapidly to zero and the shapes blend only locally.

The actual functions are:

$C((0.5 * x)^2 + y^2$ and $C((0.5 * x)^{16} + y^{16})$ where $C(r)$ is our cubic.

The contours show the function for f=0 and f=magic (0.5).

A New Family of Field Functions

In order to find the position of the iso-surface formed by a generalised version of the field function presented in [Wyvill, 86], a few concepts have to be defined:

The *primitive* shape is defined by the *field function.* The field function is calculated from a *key*, given by 3 vectors normal to each other, which intersect at the origin of the key. These can be thought of as the axes of the primitive. The field function defines the *surface* due to the key. At the origin of the key the field has a fixed value, due to that key, known as the "force" [Nishimura, 85]1. In most of our examples, this value is fixed at plus or minus one. The contribution to the field due to the key decreases with distance from the origin, until the radius of influence is reached when the contribution is deemed to be zero.

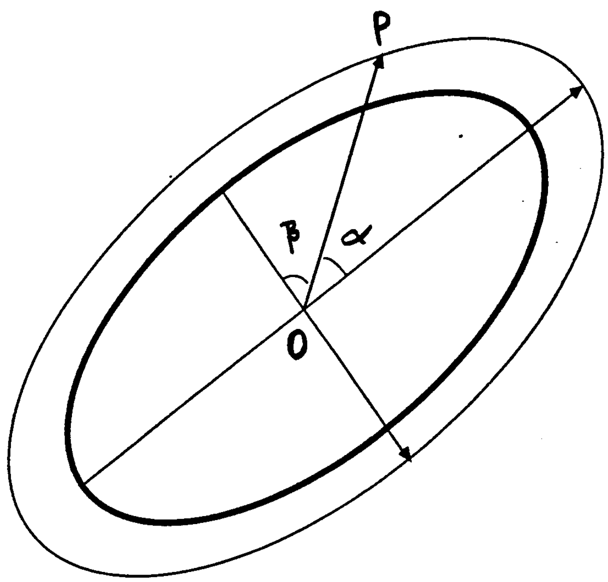

Figure 5: 2D representation of the iso-surface (in bold) due to a single key and the surface at which the force falls to zero (outer light faced ellipse).

Given a point P at distance r from the origin O of the key the contribution of that key is determined as follows:

1. Calculate the distance R where the field value turns to zero along the line OP due to the key by solving the field function.
2. If $r \geq R$ then the contribution is zero.
3. Else the field is found from r and R. (Decay function).

The field function must provide a continuous closed surface. A useful family of such functions called *super ellipsoids* was made popular by the Danish scientist and poet, Piet Hein, see [Gardner, 65]. Other functions that could be used are the family of super quadrics [Barr, 81]. The super ellipsoids are found from the equation of the ellipsoid:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \qquad (1)$$

Where a, b and c are the axes of the ellipsoid. Piet Hein observed that some pleasing shapes can be made by changing the power from 2 to some real power, n. Figure 6 shows a family of these function in $2D$ for various values of n. To make this useful for building soft objects the field function must provide a means of calculating R (see step 1 above).

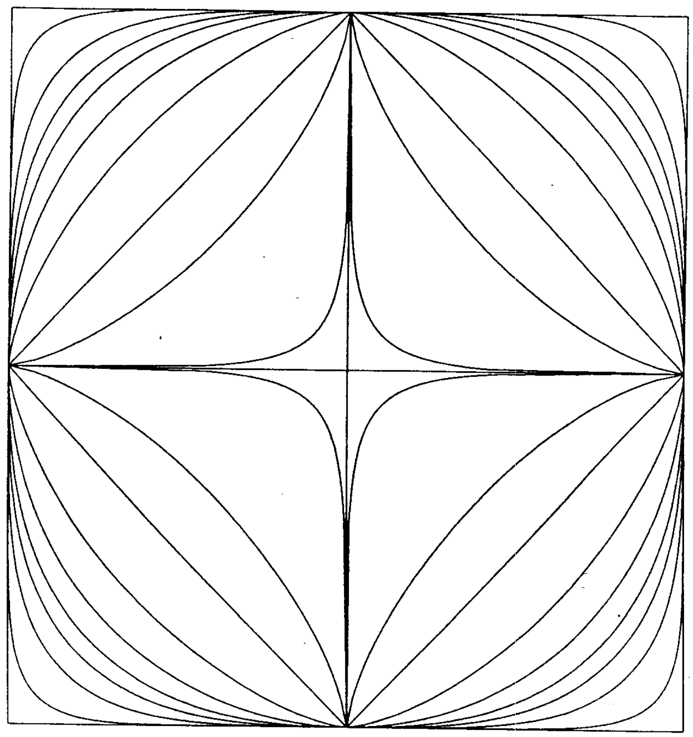

Figure 6: Piet Hein function

$$x^n + y^n = 1$$

From the centre n = 0,0.3, 0.7, 1.0, 1.5, 2, 2.5, 3, 4, 8, ∞ The function was calculated in the positive quadrant then duplicated by symmetry.

The following function is formed by replacing x, y, z in (1) with $R\cos(\alpha)$, $R\cos(\beta)$ and $R\cos(\gamma)$ where α, β and γ are the angles made by the axes of the key and the vector OP. (See Figure 5).

$$\frac{R^n\cos^n(\alpha)}{a^n} + \frac{R^n\cos^n(\beta)}{b^n} + \frac{R^n\cos^n(\gamma)}{c^n} = 1 \qquad (2)$$

where n is a real value. This form of the super ellipsoid provides a primitive defined by a key oriented at an arbitrary angle. The axes a, b, c can be used to alter the aspect ratio of the primitive. Since the axes are orthogonal, γ can be found in terms of α and β. Also since OP and the key axes' vectors are known, the cosine can be evaluated with a few multiplications. The value of the field can now be found from the values of r and R. It has been found empirically that substituting r and R into the cubic function given in [Wyvill, 86] provides good results. Reasonable results can also be obtained with fewer floating point calculations using:

$$F = 1 - \frac{r^2}{R^2} \tag{3}$$

The contribution to the field F, of some key can be scaled by the *force* characteristic of that key. The force can be positive or negative providing the user with further control on the shape of the objects being modelled. Figure 7 shows four primitives with values of n at 2 (an ellipsoid) 2.5, 3 and 4.

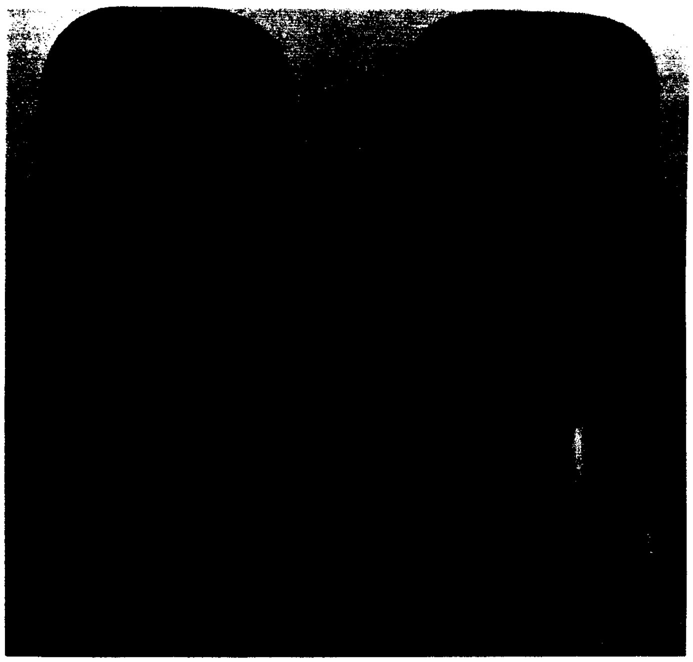

Figure 7: n = 3 (yellow) n = 4 (cyan)
n = 2 (magenta) n = 2.5 (red)

Using Several Field Functions Simultaneously

Each key is identified with a number n, that refers to the nth power in equation (2). When the field at some point in space is evaluated the contribution from each key is calculated using the appropriate value of n and added into the total. Figure 8a shows a key with $n = 4$ merging with a key with $n = 2$. Keys can have a negative contribution to the field. Figure 9 shows the result of combining two primitives. The block is made from a single key with $n = 4$ and the depression is made by adding a second primitive with $n = 2.5$ but with negative force. Blinn showed this effect using a spherical shape function for both positive and negative primitives [Blinn, 82].

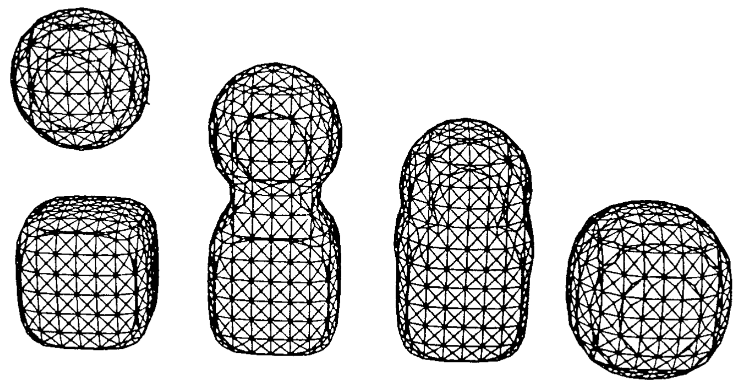

Figure 8a: Key (n=2) merging with key (n=4)

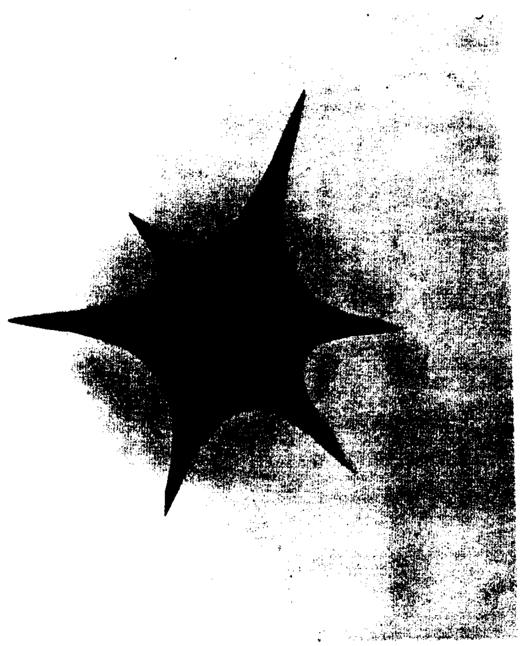

Figure 8b: Key (n = 0.5) Showing the effect of inadequate resolution in the space grid. The six points should have the same form.

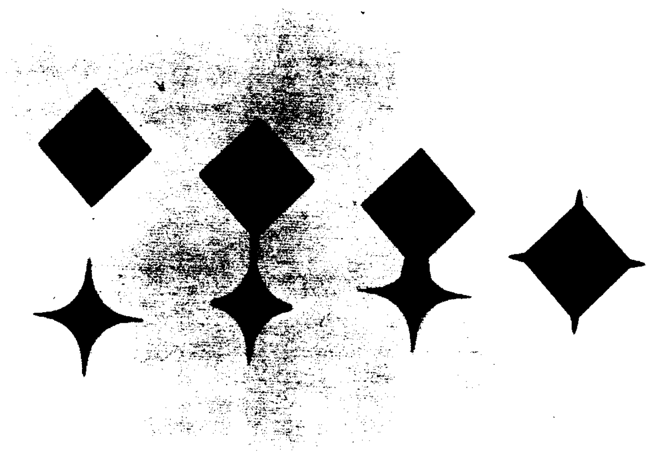

Figure 8c: Key (n = 0.5) merging with key (n = 1)

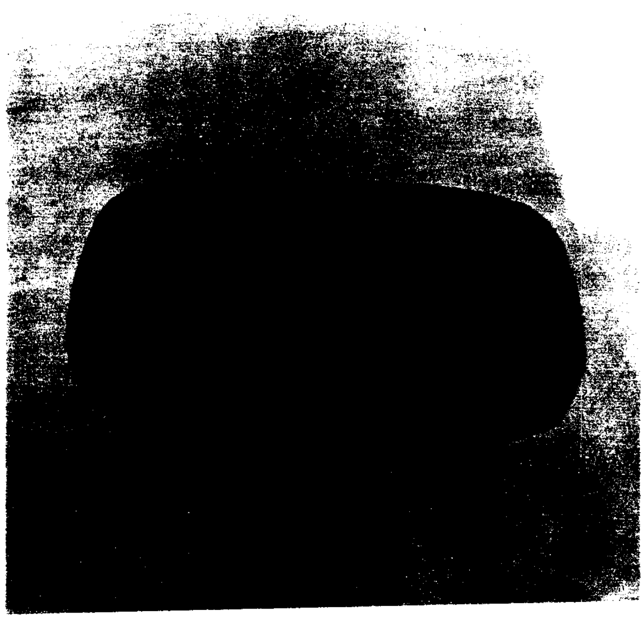

Figure 9: Positive key (n = 4) merging with (-)ve key (n = 2.5)

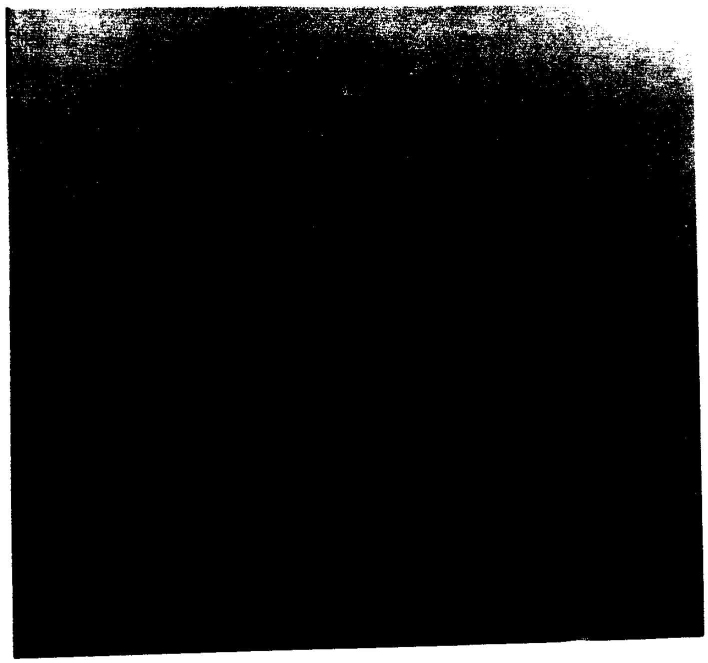

Figure 10: Monday Morning

Figure 10 shows a crude human figure constructed out of keys with $n = 2$ sitting on a bed (a single key with n=4). The human figure is polygonised on a separate pass and therefore does not blend with the bed. (See section on hierachical clothing.) The imprint of the body is made by using the same human figure constructed out of negative keys and allowed to interact with the bed.

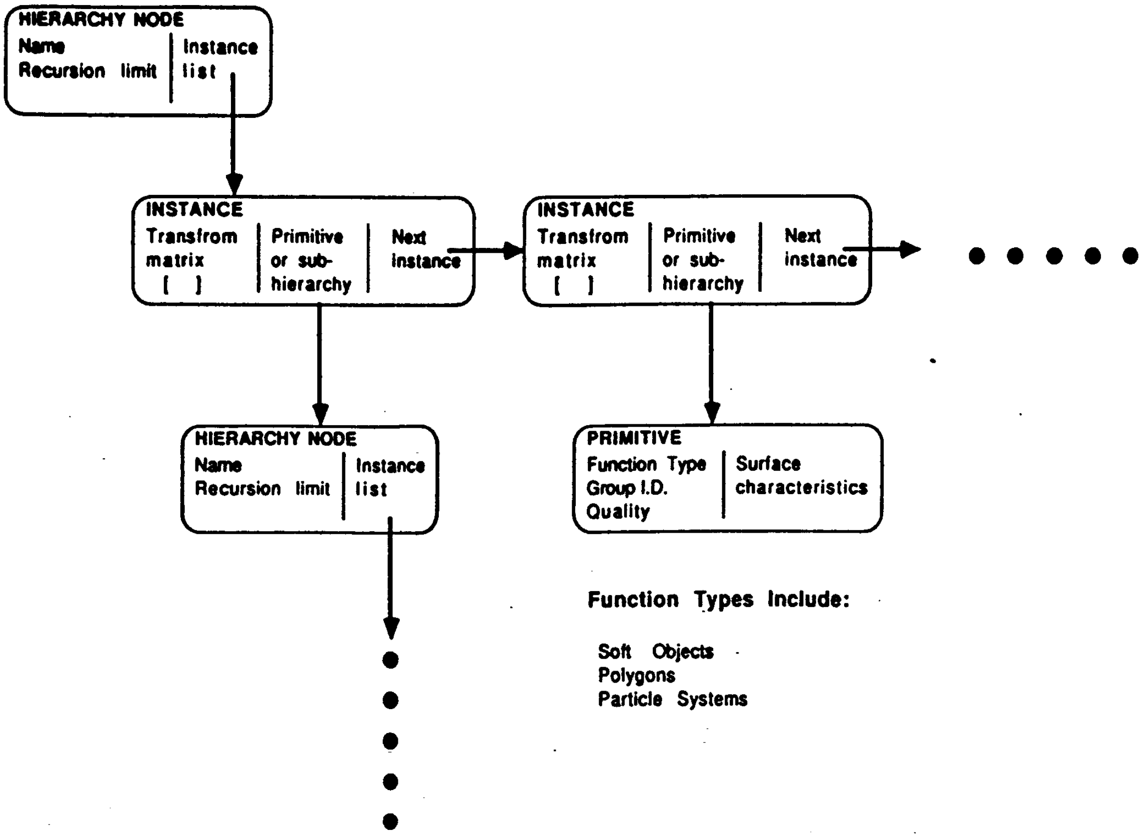

Figure 11: Basic Data Structure of *Graphicsland* Modelling System.

Hierarchical Control

The *Soft Object* system we have built has been integrated into the *Graphicsland* animation system [Wyvill, 86c]. In the system, a model is defined as a list of instances of other models. Recursive references are allowed and controlled using a numerical limit. Primitives in the system can be of a variety of types which may be sub-systems such as a particle system [Reeves, 81] or the *Soft Object* system. Attributes needed to control the way in which the surface is to be produced are inherited through the system. Figure 11 provides an overview of the modelling data structure. Details of the data structure and the traversal algorithm are given in [Wyvill, 86c]. For *Soft Objects* the attributes are:

Function type
Group
Quality
Surface attributes (colour, texture etc.)

Function type refers to the field function that is to be used to evaluate the contribution due to this soft primitive. Currently we use a float which beomes the power N in the super ellipsoid equation. This could be extended to include an integer selector to choose a class of function to be used, with the float selecting a member of that class. This allows for experimentation with new classes of function by altering the Soft Object sub-system without changing the testbed. When the data structure is traversed, each primitive is passed to its respective sub-sytem. A model in the system can be given a Group attribute. All *Soft* primitives which share the same group are passed to the Soft Object sub-system together. Each new group causes a new version of the sub-system to be executed. Only the primitives within a group will be blended together. An example is shown in Figure 12. The letter T is made from two primitives using the super ellipsoid function with N=4 for the horizontal section and N=2 for the vertical. Figure 12a shows the result of blending the two objects. Figure 12b shows a version where the two parts of the model have been created and given different group attributes and therefore do not blend but intersect when one is placed on top of the other. The *Quality* attribute governs the density of the polygon mesh approximating the surface of a group. The higher the quality the smaller the size of the cubic grid described earlier. Figure 12b also illustrates this feature. The horizontal bar has been given a lower quality attribute (fewer polygons) than the vertical section.

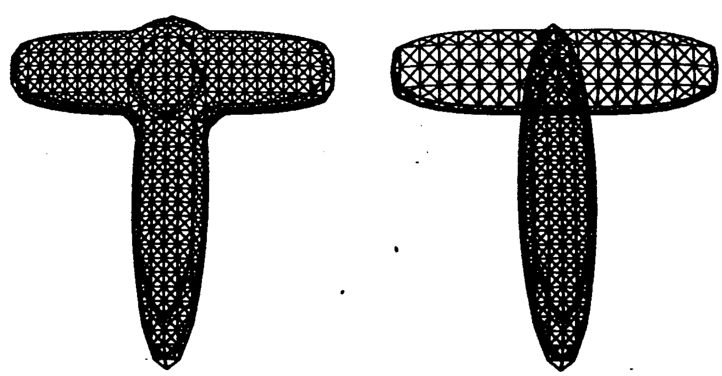

Figure 12: **Example of group attribute. Figure 12a shows blending while 12b does not.**

Further Work

We have found that many of the blending problems can be overcome with a creative use of the hierarchical group structure. However there is still a problem when model A must blend with model B and C, but B must not blend with C. This is exemplified in Figure 3. In such a situation, when calculating the field at a point between a B key and a C key, only the higher of the two contributions should be considered.

Conclusions

We have described further work in our "Soft Object" modelling system. Certain problems with this technique have been identified and some effort has been made towards solving some of these problems. We have developed a new family of field functions and shown how these can be used in a practical way to create models which can be blended together in a hierarchical fashion. During this work it has become apparent that there is great potential here for a really useful approach to modelling for computer graphics.

Acknowledgements

We would like to acknowledge the help of all the students and research staff who have worked on the *Graphicsland* project and soft objects, also the current and former students; Jeff Allan, Angus Davis, Dave Jevans, Trevor Paquette and Craig McPheeters; and the many other graduate and undergraduate students at the University of Calgary who have contributed to the project. This research is partially supported by grants from the Natural Sciences and Engineering Research Council of Canada.

References

1. Barr, A, "Superquadrics and angle preserving transformations," *IEEE Comput. Graphics and Appl.* **1 January** pp. 11-23 (1981).

2. Blinn, J, "A Generalization of Algebraic Surface Drawing," *ACM Transactions on Graphics* **1** p. 235 (1982).

3. Bloomenthal, J, "Boundary Representation of Implicit Surfaces," *Research Report CSL-87-2*, Xerox PARC , (1987).

4. Gardner, M, "Mathematical Games," *Scientific American (July)*, (1965).

5. Jevans, D, Wyvill, B, and Wyvill, G, "Speeding Up 3D Animation For Simulation," *Proc. SCS Conference (inpress)*, (1988).

6. Lorensen, W and Cline, H, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics (Proc. SIGGRAPH 87)* **21**(4) pp. 163-169 ().

7. Middleditch, A and Sears, K, "Blend Surfaces for Set Theoretic Volume Modelling Systems," *Computer Graphics (Proc. SIGGRAPH 85)* **19**(3) pp. 161-170 ().

8. Nishimura, H, Hirai, A, Kawai, T, Kawata, T, Shirakawa, I, and Omura, K, "Object Modeling by Distribution Function and a Method of Image Generation," *Journal of papers given at the Electronics Communication Conference '85 (in Japanese)* **J68-D**(4)(1985).

9. Reeves, William., "Particle systems- A technique for modeling a class of fuzzy objects," *ACM Transactions on Graphics* **2** pp. 91-108 (Apr 1983).

10. Von Herzen, B and Barr, A, "Accurate Triangulations of Deformed, Intersecting Surfaces," *Computer Graphics (Proc. SIGGRAPH 87)* **21**(4) pp. 103-110 ().

11. Wyvill, B, McPheeters, C, and Garbutt, R, "The University of Calgary 3D Computer Animation System," *Journal of the Society of Motion Picture and Television Engineers* **95**(6) pp. 629-636 (1986d).

12. Wyvill, G, Wyvill, B, and McPheeters, C, "Data Structures for Soft Objects," *Visual Computer* **2**(4) pp. 227-234 (1986a).