Disney Effects Using Implicit Surfaces

Agata Opalach
Steve Maddock
Department of Computer Science, The University of Sheffield,
Regent Court, 211 Portobello Street, Sheffield, S1 4DP, UK
Tel: +44 742 825579 Fax: +44 742 780972
E-mail: a.opalach@dcs.shef.ac.uk

1. Introduction

In their wonderful book “Disney Animation: the Illusion of Life” [18], Frank Thomas and Ollie Johnston tell a story about traditional animation and how, within a short time, it developed into a new, successful art form. The animators at the Walt Disney Studios succeeded in creating characters that “came to life” on cels. This success was made possible through endless refinement of drawings until a desired emotion was apparent for the audience. As a result of these experiments a list of fundamental principles of traditional animation was compiled, which verbalised the experience gained by the animators at the Walt Disney Studios.

These principles have been successfully applied to computer animation. Lasseter [9] describes the creation of “Luxo Jr.”, an animation sequence involving an anglepoise lamp that appeals to the audience in the same way as Disney animation. In this work, traditional animation principles were applied manually. The fact that an anglepoise lamp was chosen as the character helped the squash and stretch effects to be achieved more easily. It would be difficult to generalise this method to be used with arbitrary objects. Also, a lot of artistic skill is required which makes it very hard for an ordinary user to reproduce these effects in a computer animation system. In our system we are attempting to incorporate some of the traditional animation principles as the default behaviour of implicitly defined objects.

Implicit surfaces as a modelling technique were initially described by Blinn (blobby molecules) [2], Nishimura et al. (metaballs) [13] and Wyvill et al. (soft objects) [20, 21]. All three approaches used scalar fields around 3D points to build their models. The technique was later generalised for arbitrary skeletal elements [3, 4, 22] and has found practical applications in a wide variety of areas. For instance, Tatsumi et al present the use of metaballs in the process of modelling a cod's liver [16], Fujita et al. represent splashing water using metaballs [6], Muraki uses the blobby model for volumetric shape description (including human face modelling) [12], Max and Wyvill render corals as soft objects [10] and Payne and Toga model a rat's brain using distance fields [15].

Implicit surfaces have also been appreciated in the field of computer animation. Beier from Pacific Data Images commented on the usefulness of blobby primitives for character animation [1] and, in Japan, metaballs have been widely used for modelling and animating shapes as complex as human anatomy parts [8]. Another use for the technique is in visualising deformable material modelled as particles [11, 19] or mass points [17]. The motion of individual particles is modelled by physically based simulation using the Newton equations of motion and by modelling the interaction between the particles (e.g. their reaction to heat). The particles are then treated as skeletal elements (points) and rendered as an implicit surface.

Gascuel has developed a method for modelling precise contact between colliding implicit solids [7]. Her method has been used to model a highly deformable material [5], for which the skeletal elements are animated in a generalised physically based particle system. The isosurface around skeletons is generated and collisions between objects are processed to calculate the resulting deformations during the collisions. This method models inelastic behaviour of material with fewer particles (skeletal elements) than previous approaches, e.g [11, 19].

There has been some work in applying the principles of traditional animation to implicit surfaces. This was done by Wyvill who has developed a technique for modelling squash and stretch effects[23]. In this work, the scalar field around skeletal elements is deformed by warping different regions of space so that the implicit surfaces squashes on contact with the ground and stretches during motion. Effects such as slug-like motion or waves can be achieved by choosing a suitable warping function.

In our approach we use implicit surfaces for character animation and attempt to incorporate the following traditional animation principles into the system: Squash and Stretch, Anticipation, Follow Through and Exaggeration. The work is based on our earlier work on the ABC (appearance, blending and coherence) of implicit surfaces [14], which addresses the problems of (i) creating characters which are recognisable during an animation sequence (preserve their appearance), (ii) unwanted blending (e.g. between an arm and the body) and (iii) preserving the coherence of a character during motion (i.e. to prevent the "scrambling" effect when an object breaks into pieces). Our models are built as articulated figures clothed with implicit flesh. Animation is controlled at a higher level by moving the articulated figure. There is no direct motion control of the skeletal elements which form the implicitly defined flesh. They follow the motion of the articulated figure. Therefore the flesh always "sticks" to the links and joints of the articulated figure.

In the remainder of this paper we briefly summarise the ABC of implicit surfaces, present and justify the model used in our system and describe how Disney effects are automatically created. Each effect will be discussed separately pointing out the advantages and drawbacks of our techniques.

2. The ABC of Implicit Surfaces [14]

First let us briefly introduce the terminology of implicit surfaces. Consider a set of skeletal elements (points, lines, polygons) with a scalar field function $f_i: \Re^3 \rightarrow \Re$ defined for them. An implicit surface can be defined as $\{P \in \Re^3 | \; F(P) - Iso = 0\}$ where Iso is the isovalue at which an isosurface is extracted and $F(P) = \sum_{i=1}^{n} f_i(P)$. Figure 1 shows an implicit surface created around two points.

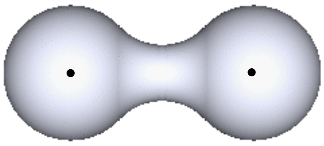


Figure 1. An implicit surface around two points

One of the most attractive features of implicit surfaces is their deformability. The shape of an object can be changed by simply moving the skeletal elements. A closed, smoothly blended surface is always guaranteed although the topology of the object may change. This makes the technique very attractive for modelling deformable objects which may undergo topology changes.

2.1 Appearance

Appearance is essential for character animation. A character should remain recognisable throughout an animation sequence regardless of the deformations it undergoes. For instance, a human arm should not become a "blob" when it is moving from one position to another. Its parts (shoulder, upper arm, lower arm, elbow, wrist and hand) should remain distinct at any given time. One solution to maintain such a general structure within a model is to impose it on the skeletal primitives. A graph structure of links and joints with primitives attached to them can be used.

2.2 Blending

The unwanted blending problem occurs when primitives blend when we do not want them to. Therefore, some parts of an object should be blendable with each other and other parts unblendable with each other. Such blending properties should be specified for each object. Assuming the above graph structure of links and joints, the blending properties can be automatically derived. The default blending properties assume that each link only blends with its joints and nothing else. The blending properties are considered in the rendering process in order to only blend blendable primitives.

2.3 Coherence

We define coherence as the property of an object which causes it to stay in one piece without breaking apart during an animation. Coherence can be easily lost when one primitive is moved too far apart from the others. Hence, constraints on the motion of primitives are required to prevent this. When some primitives are moved, the others should follow this motion until an object stretches to its extremes.

3. The Model

There are three components in our model (Figure 2). Firstly, a graph structure of *graphlinks* and *graphjoints* is specified for an object. This is an abstract specification of a model which helps to maintain its appearance. Secondly, the abstract graphlinks are clothed with physical *flexilinks*. These consist of skeletal primitives placed on each graphlink. This is the physical skeleton of the object which is analogous to the skeleton of a human. Finally, the flexilinks are clothed with flesh, i.e. a number of layers of skeletal primitives attached to the flexilink primitives.

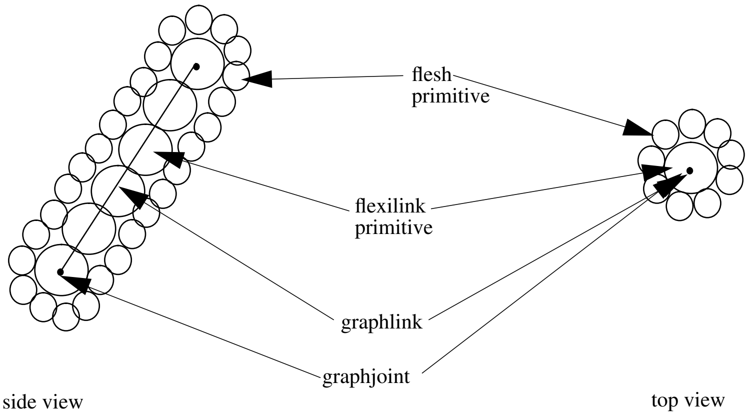


Figure 2. The four components in the model (graphlink, graphjoint, flexilink and flesh)

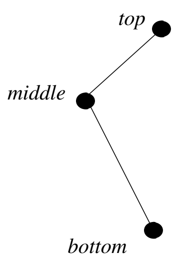
When animating such an object, the flexilinks are moved first (based on the motion of the graphjoints and graphlinks) and the flesh follows the motion. The flexilinks can be rigid or flexible. The flesh is always deformable and fluid like.

The models in our system are input through script files. A script file specifies the graph structure (graphjoints and graphlinks). It also specifies the iso-value to be used for the iso-surface extraction, the radii of primitives to be used for the flexilinks and the flesh and the number of layers in the flesh. For the graphlinks, the flexibility of the corresponding flexilink can be specified, varying from 0 (rigid) to 1 (flexible). Flexilinks and flesh are automatically created by the system. Figure 3 shows a sample model: the script file, the graph structure, the flexilinks and the flesh. The object consists of three joints (*top, middle* and *bottom*) connected by two links.

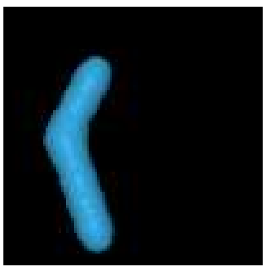
(a)

```
iso .6
  object stickleg
    joint bottom -10 0 0
    joint middle -20 0 40
    joint top -10 0 60
    link bottom middle .2
    link top middle .2
    skeleton 5.
    flesh 2. 7.
end
```

(b)



(c)



(d)

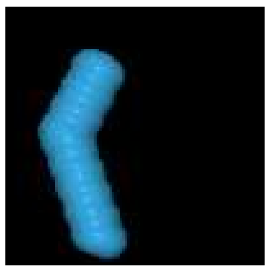


Figure 3. A sample model: (a) script file (b) graph structure (c) flexilinks (d) flesh

4. Motion

In order to specify an animation sequence, we specify the movement of the graphjoints in an animation script file. The motion of a graphlink is calculated based on the motion of its graphjoints. The flexilink primitives move according to the motion of the graphlink they belong to. Finally, the flesh follows the motion of the relevant flexilink. We will now describe each of these movements.

```
frames 2
  object stickleg
  times 0 2
    translate top
       force 5 0 0
       velocity 0 0 0
end
```

Figure 4. A sample animation script file

4.1 Scripting the movement (graphjoints and graphlinks)

An animation script (Figure 4) specifies the forces and velocities applied to graphjoints at different times of an animation sequence. Depending on the flexibility factor specified for graphlinks in the modelling script (Figure 3a), the graph moves as a rigid or a flexible structure. For the rigid motion (flexibility factor zero), the movement of each graphjoint is propagated in the graph structure so that the length of each graphlink remains constant. In this case the distances between flexilink primitives, which are distributed along graphlinks, do not change.

If the flexibility factor of a graphlink is greater that zero, the movement of graphjoints can change the length of this graphlink. This means that the motion of a graphjoint is only propagated to the other graphjoints in the graph structure when the distance between the two graphjoints is less than the minimum or greater than the maximum graphlink length. The minimum and maximum limits depend on the initial length of a graphlink and on the flexibility factor. Figure 4 illustrates the two types of graphjoint motion. The spheres represent flexilink primitives.

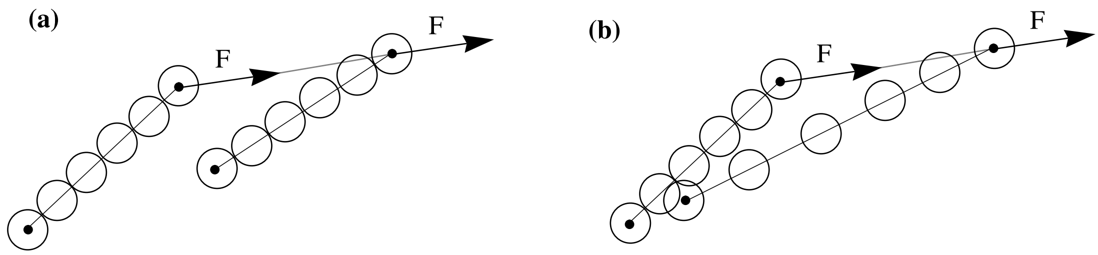


Figure 5. The motion of graphlinks (a) rigid - the length of graphlinks is constant; (b) flexible - the length of graphlinks can be changed within certain limits

4.2 Flexilinks

The flexilink primitives are distributed along graphlinks. Therefore, their motion can easily be calculated knowing the movements of a graphlink and two graphjoints. Figure 6 illustrates the basic principle of this process.

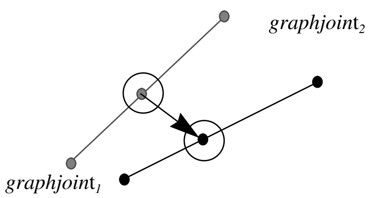


Figure 6. Moving flexilinks - a flexilink primitive moves in relation to the motion of two graphjoints

4.3 Flesh

The flesh is attached to the flexilinks. Each flesh primitive has its parent primitive in a flexilink and its motion is constrained around the parent. A flesh primitive has to be within the minimum and maximum distances from its parent (Figure 7). Fleshlink primitives, when moved, push the flesh primitives in front of them (in the direction of the motion) and pull the flesh primitives behind them. The push/pull action only occurs when the distance between a flesh primitive and its parent in not within the minimum and maximum range. During its motion, the flesh acquires a velocity from the movement of a flexilink. When the flexilink terminates its motion, the flesh continues to move due to this velocity. During this follow through motion, each time a flesh primitive reaches the maximum or minimum distance from its parent, it bounces back with its velocity reversed and dampened. In the plot in Figure 7b the paths of two flesh primitives falling under gravity are shown. Dampening occurs at the maximum allowed distance from the parent. The velocity magnitude (speed) acquired by a flesh primitive from the motion of its parent is given by

$$v = d\frac{\vec{V} \cdot \vec{P}}{\|\vec{P}\|^2} \qquad \text{(EQ 1)}$$

where v - acquired velocity magnitude
V - parent velocity
P - the position vector from the flesh primitive to the parent primitive
d - scaling factor

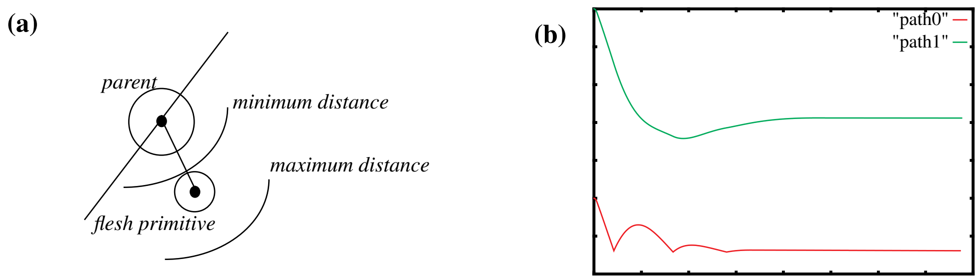


Figure 7. The relation between flesh and flexilinks: (a) a flesh primitive has to be within the maximum and minimum distance from its parent; (b) a plot of the down translation for two flesh primitives falling under gravity

In addition to the above geometric interaction, the flesh primitives experience attraction/repulsion forces between each other. The interaction between two primitives is given by EQ 2 [11].

$$\overrightarrow{F_{ip}} = \begin{cases} s\left(\dfrac{b_1}{D^m} - \dfrac{b_2}{D^n}\right)\dfrac{\vec{P}}{\|\vec{P}\|} & \mathrm{i} \neq p \\ 0 & \mathrm{i} = p \end{cases} \qquad \text{(EQ 2)}$$

where F_{ip}- force between primitive i and p
D - distance between the primitives
P - the position vector from the primitive i to the primitive p
m, n - constants, $m = 2n$
b_1, b_2 - variables, $b_2 = b_1 r_0^{n-m}$
r_0 - the equilibrium distance between the primitives, when there is no interaction between them
s- scaling factor

The equilibrium distance r_0 depends on the initial distance between two primitives. For primitives close to each other, it is set to their initial separation R_{init}. When the initial distance is larger than a maximum allowed distance, the equilibrium distance r_0 is set to the maximum distance. This means that there is very little interaction between primitives which are initially far from each other. Therefore, an object will retain its general shape during the animation rather than being “melted” due to the interaction forces.

4.4 Collisions

The types of collisions which are detected in our system are:

- **flesh-flexilink**: This collision occurs when a flesh primitive moves towards its flexilink. This is handled by monitoring the minimum/maximum distance from its parent.
- **flexilink-flesh**: When flexilink primitives are moved, they push the flesh primitives in front of them. Again this type of collision is detected by minimum/maximum distance control between the flesh and flexilink primitives.
- **flesh-obstacles**: When a flesh primitive collides with a plane (e.g. a wall or a ground plane), the intersection of the primitive with the plane offset from the real obstacle by the radius of a primitive is calculated. The flesh primitive is moved to the intersection point and its velocity is reversed and dampened.
- **flesh-flesh**: For flesh primitives which belong to the same flexilink, the flesh-flesh interaction is calculated as described by equation EQ 2. For flesh primitives from different flexilinks, the interaction depends on the blending properties of the object. For blendable flexilinks, interaction occurs, whereas for unblendable flexilinks interaction is not calculated and the primitives are separated in the rendering process [14].

5. Traditional Animation Effects

We have created the model described in this paper to incorporate the following Disney effects: Squash and Stretch, Anticipation, Follow Through and Exaggeration. We will now discuss how our model facilitates the automatic incorporation of these effects.

5.1 Squash and Stretch

One of the desired features of a character in animation is that it stretches and squashes during its motion. This behaviour makes it more lively and expressive. In our system, character squash and stretch is achieved by moving the flesh primitives. When the only force affecting them is gravity, they stay at the *rest position*. In this position the flexilinks are stationary and the flesh falls down under gravity. The effect, shown in Figure 8, produces a character squashed at the bottom, with all its flesh hanging from its flexilinks.

(a) **(b)** **(c)**

Figure 8. A stickleg "at rest" under gravity: (a) the model (no deformation); (b) the geometric deformation; (c) added attraction/repulsion forces

Figure 8b shows the geometric approach (Section 4.3), where no attraction/repulsion forces are considered. The shape of the object is very regular because all the flesh primitives behave in the same way. Adding attraction/repulsion forces creates the effect in Figure 8c. The scaling factor $s = 0.5$ was used in equation EQ 2. The object is less regular and its flesh "bubbles around" its flexilinks. This effect is perhaps more interesting because all regularities in the drawings are avoided and the character appears to have more "life".

5.2 Follow Through

The Follow Through effect in traditional animation describes the loose parts of the body following the motion of more rigid skeletal parts and flesh continuing its motion after the skeleton has stopped. In our system these effects are automatically generated when the motion of the flesh is calculated. For instance, in the jump pictured in Figure 9, the stickleg jumps into the air and its flesh is dragged behind with a delay. After the flexilinks finish their motion, the flesh is still moving under gravity and acquired velocity until it's movement dampens to zero and it comes back to the rest position. Figure 9a shows the script file used to script the jump. Again, two versions of the animation are shown.

In the first one (Figure 9b), the geometric approach was used which creates a fairly regular effect. When the attraction/repulsion forces are switched on ($s = 0.5$ in EQ 2), the flesh follows in a more fluid manner (Figure 9c).

(a)

```
frames 10
  object stickleg
  times 5 10
    translate bottom
      force 5 0 10
      velocity 0 0 0
end
```

(b) **(c)**

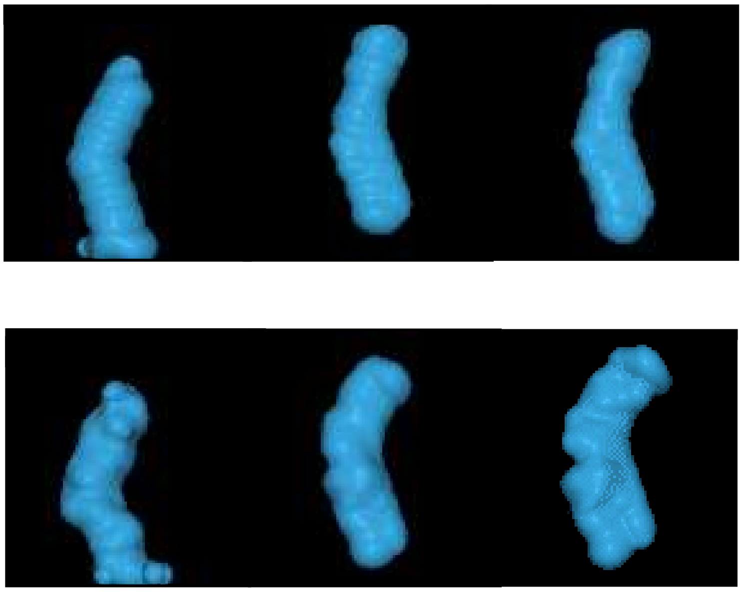


Figure 9. Follow Through: the stickleg jumps into the air from its rest position: (a) the animation script; (b) the geometric approach; (c) with attraction/repulsion forces

5.3 Anticipation

The geometric approach to Anticipation involves performing a small movement in the direction opposite to the movement specified in the animation script. However, the automatic Follow Through of the flesh and attraction/ repulsion forces complicate this. When the anticipation movement is performed, the flesh is dragged behind the flexilinks as for any other movement. We actually need the flesh to be positioned in the direction opposite to the scripted movement.

5.4 Exaggeration

For Exaggeration, the system provides a set of controllable environment parameters (e.g. gravity, minimum and maximum distances for a flesh primitive, an attraction/repulsion factor or a damping factor). By using these parameters, the animator can produce exaggerated effects. For example, in Figure 9 switching attraction/repulsion forces on caused the flesh to appear more fluid. Increasing the scaling factor s in EQ 2 produces extremely fluid flesh. Extreme squashing under gravity can be achieved by increasing gravity and/or increasing the maximum allowed distance from a parent. We are currently experimenting with changing the length of the flexilinks beyond their limits by adding or deleting flesh primitives when required. This will give us extreme stretching effects.

6. Conclusions

Our system provides a new method for scripting animation for models built of implicit skeletal elements. There is no need to directly control the motion of individual primitives since they follow the motion specified for a higher level structure (a graph). The topology of the initial object thus remains the same which is essential for character animation. Another advantage of our system is that some traditional animation principles are implemented as the default behaviour of objects. This provides automatic effects such as Squash and Stretch or Follow Through. More effects, including Anticipation and Exaggeration are currently being investigated.

We are also considering the bending of graphlinks to model characters with flexible spines and metamorphosis between characters by inbetweening their abstract graph structures. Here, the flesh would follow the motion of the parent graph structure and the metamorphosis itself would exhibit Disney effects.

7. References

1. Beier, T. “Practical Uses for Implicit Surfaces in Animation“, Siggraph 90 Course Notes No 23, “Modelling and Animating With Implicit Surfaces”, pp 20.1-20.11

2. Blinn, J. F., "A Generalisation of Algebraic Surface Drawing", ACM Trans. Graphics, Vol. 1, No 3, July 1982, pp 135-256

3. Bloomenthal, J. and Wyvill, B., "Interactive Techniques for Implicit Modelling", Computer Graphics, Vol. 24, No 2, 1990, pp 109-116

4. Bloomenthal, J. and Shoemake, K., "Convolution Surfaces", Computer Graphics, Vol. 25, No 4, July 1991, pp 251-256

5. Desbrun, M. and Gascuel, M-P., ”Highly Deformable Material for Animation and Collision Processing”, Proc. 5th Eurographics Workshop on Animation and Simulation, Oslo, September 94.

6. Fujita, T., Hirota, K. and Murakami, K., "Representation of Splashing Water using Metaball Model", Fujitsu, 1990, Vol. 41, part 2, pp 159-165, in Japanese

7. Gascuel, M.-P., "An Implicit Formulation for Precise Contact Modelling between Flexible Solids", Proceedings of SIGGRAPH 93 (Anaheim, California, August 1-6, 1993). In Computer Graphics Proceedings, Annual Conference Series, 1993, ACM SIGGRAPH, New York, pp 313-320

8. Graves, G., "The Magic of metaballs", Computer Graphics World, May 1993, pp 27-32

9. Lasseter, J., “Principles of Traditional Animation Applied to 3D Computer Animation”, Computer Graphics, Vol. 21, No 4, July 1987, pp 35-44

10. Max, N. L. and Wyvill, B., "Shapes and Textures for Rendering Coral",Scientific Visualisation of Physical Phenomena, N. M. Patrikalakis (Ed.), Springer-Verlag 1991, pp 333-343

11. Miller, G. and Pearce, A., “Globular Dynamics: A Connected Particle System for Animating Viscous Fluids”, Vol. 13, No 3, 1989, pp 305-309

12. Muraki, S., "Volumetric Shape Description of Range Data Using "Blobby Model"", Computer Graphics, Vol. 25, No 4, July 1991, pp 227-235

13. Nishimura, H., Hirai, M., Kawai, T., Kawata, T., Shirakawa, I. and Omura, K., "Object Modelling by Distribution Function and a Method of Image Generation", The Transactions of the Institute of Electronics and Communication Engineers of Japan, 1985, Vol. J68-D, Part 4, pp. 718-725, in Japanese, translated into English by Takao Fujuwara

14. Opalach, A. and Maddock, S., “Implicit Surfaces: Appearance, Blending and Consistency”, Proc. 4th Eurographics Workshop on Animation and Simulation (Barcelona, 4-5th September, 1993), 233-245.

15. Payne, B. A. and Toga, A. W., "Distance Field Manipulation of Surface Models", IEEE Computer Graphics and Applications, January 1992, pp 65-71

16. Tatsumi, H., Takaoki, E., Omura, K. and Fujito, H., "A new method for 3D reconstruction from serial sections by computer graphics using "meta-balls": reconstruction of "Hepatoskeletal System" formed by Ito Cells in the Cod Liver", Computers and Biomedical Research, 1990, Vol. 23, Part 1, pp 37-45

17. Terzopoulos, D., Platt, J. and Fleisher, K. “Heating and melting deformable models (from goop to glop)”, Graphics Interface’89, June 1989, pp 219-226

18. Thomas, F. and O. Johnston, “Disney Animation: the Illusion of Life”, Abbeville Press, 1981

19. Tonnesen, D., “Modelling liquids and solids using thermal particles”, Proceedings of the Graphics Interface 1991, pp. 255-262

20. Wyvill, G. and McPheeters, C. and Wyvill, B., "Data Structure for Soft Objects", The Visual Computer, Vol. 2, No 4, August 1986a, pp 227-234

21. Wyvill, G. and McPheeters, C. and Wyvill, B., "Animating Soft Objects", The Visual Computer, Vol. 2, No 4, August 1986b, pp 235-242

22. Wyvill, B. and Wyvill, G., "Field Functions for Implicit Surfaces", Visual Computer, Vol. 5, 1989, pp 75-82

23. Wyvill, B., “Warping Implicit Surfaces” Siggraph 93 talk, Course No 25, “Modelling, Visualizing and Animating Implicit Surfaces”