High Level Control of Implicit Surfaces for Character Animation

Agata Opalach
Steve Maddock

Department of Computer Science, The University of Sheffield,
Regent Court, 211 Portobello Street, Sheffield, S1 4DP, UK
Tel: +44 114 282 5577 Fax: +44 114 278 0972
E-mail: a.opalach@dcs.shef.ac.uk

Abstract

Implicit surfaces, also known as blobby molecules [2], metaballs [14] or soft objects [22], are particularly suitable for modelling smoothly blended, "plasticine" objects. One of their most attractive features is their deformability. The shape of an object can be altered by simply moving the skeletal elements. A closed, smoothly blended surface is always guaranteed although the topology of an object may change. In this paper we will look closer at the use of implicit surfaces for character animation. We use a layered model to represent characters. A graph defines an articulated figure. The limbs which cover the articulated figure are designed by a user and form the body of a character. The body is covered by flesh which can be automatically generated by the system (skin) or partially designed by a user (strands of flesh: hair, a tail etc.). We use a mixture of manual and automatic motion control. Keyframes for the articulated figure are specified using dance notation and the motion of flesh is automatically generated by the system. A user can control the motion of flesh at a high level using a set of adjustable parameters (gravity, flesh firmness and exaggeration). We illustrate our approach using a dancer performing to the given choreography.

Keywords: Implicit Surfaces, Character Animation, Dance Notation, Automatic Motion Generation

1. Introduction

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 around them. An implicit surface can be defined as $\{P \in \Re^3 \mid F(P) - Iso = 0\}$ 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. We will refer to skeletal elements as *primitives*.

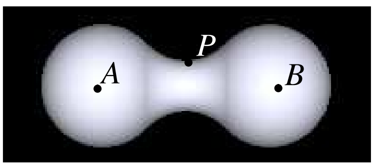

Figure 1. *An implicit surface around two skeletal elements, or primitives, A and B*

Wyvill *et al.* [23] and Beier [1] comment on the usefulness of implicit surfaces for character animation. They stress the ability of implicit surfaces to conform to the environment, to metamorphose and to change topology - effects often required in character animation and more difficult to achieve using polygonal or parametric surface models. However, implicit surfaces introduce a few problems during animation: an object can melt into an unrecognisable cluster of primitives, unwanted blending can occur between its parts or it can break into pieces. To overcome these problems, we identified the ABC of implicit surfaces for character animation: appearance, blending and coherence [15]. A character has to have *appearance* which we define as a recognisable look, that is clearly maintained throughout the animation. *Blending* is an attractive property of implicit surfaces but it is sometimes undesirable, e.g. blending between the limbs and the body. Avoiding unwanted blending is the second important aspect of character animation with implicit surfaces. Finally, maintaining *coherence* (constant topology) prevents an object from breaking into pieces during the animation. However, topology changes may be required for performing other animation effects, e.g. metamorphosis. Unless such a change in topology is requested, a character should stay coherent.

Another advantage of using implicit surfaces for character animation is the implicit skin defined by the isosurface. Implicit surfaces impose a natural layered structure on an object: the inner skeleton (the primitives) and the outer skin (the isosurface). If we define an articulated figure by specifying a hierarchy among primitives, the implicit surface can represent the skin which covers the character. Such skin is always C^1 continuous regardless of the movement of the primitives. During animation however, deformation of the skin is difficult to control. Unwanted blending may occur if two adjacent limbs are allowed to blend. A solution to this problem [15, 6, 10] requires a blending graph to be defined for the articulated structure which determines which parts of a character can blend with each other and which deform during contact. Creases and bulges of the skin can then be modelled but the blending graph has to be chosen with care to avoid surface discontinuities.

Alternative methods for modelling skin surface of an articulated figure use polygons [12, 4] or parametric surfaces [8]. In [12] a method called *joint-dependent local deformation (JLD)* is developed to modify polygons around a joint. [4] uses *free-form deformation (FFD)* [17] and modifies a deformation lattice to alter the surface polygons. [8] describes *hierarchical B-spline surface* which can generate folds and creases around joints. Although these methods give more control over the way the skin deforms, they require polygon or parametric patch fitting to preserve surface continuity, which is automatic in the case of implicit surfaces.

In this paper we describe a layered model for character animation using implicit surfaces. Three layers are constructed for each character. A graph of links and joints specifies an articulated body. Each link in the graph represents a limb which is designed by a user. All limbs blended together constitute the body of a character. The final layer, flesh, covers the body. The system offers two kinds of flesh: skin (automatically generated) and strands of flesh (partly specified by a user). We try to balance between fully automatic motion generation and total manual control over the movement. Keyframes of an articulated figure which defines a character are given by a user in a script file using Eshkol and Wachmann's dance notation [7]. The motion of flesh is calculated by the system from the motion of the articulated figure. An animator can control the flesh behaviour by changing a set of parameters, which can be adjusted by turning each feature up or down. Gravity (none to high), flesh firmness (wooden to very loose) and exaggeration (none to high) are the available parameters.

2. Animation Control of Implicit Surfaces

Since a model built using implicit surfaces consists of multiple primitives, controlling their animation is difficult. We do not want to move a character as a rigid object by applying geometric transformations to all primitives (unless we model a smoothly blended wooden sculpture). We want the primitives to change their spatial relationship slightly to create more natural motion, with deformation. On the other hand we do not want to specify a separate path for each primitive in a character.

Keyframe animation is often used for motion control of implicit surfaces [23, 1]. A set of keyframe positions for all primitives is given and the inbetweens are automatically generated by interpolating between the corresponding primitives in each keyframe. The same approach can be used for metamorphosis of two implicit models [24]. In this case, a matching between the primitives in two keyframes has to be found first. A C^1 continuous surface is always created but the way the isosurface deforms during animation is difficult to predict or control.

Another approach is to apply deformation to the shape of the scalar field around primitives to achieve realistic and controlled results. For instance, Gascuel [9] uses dynamic simulation to animate implicitly defined solids. She detects collisions between them and calculates the precise contact surface between them and the local deformation during collisions. Wyvill [25] uses space warping to create squash and stretch on a bouncing ball. The ball squashes when it enters the region of space closer to the ground and stretches when it bounces back up.

Physical simulation in a particle system can also be used to animate implicit surfaces. It gives particularly good results for modelling inanimate materials, e.g. mud, clay or dough. Deformable material modelled in a particle system can be visualised using implicit surfaces [13, 18, 20]. The motion of individual particles is modelled by physically based simulation using the Newton equations of motion and by considering the interaction forces between the particles. The particles are then treated as skeletal elements (points) and rendered as an implicit surface. A generalised physically based particle system for skeletal elements has been used to model a highly deformable material [5]. Such material can undergo fractures and its chunks can join back together. Collision detection and response between unblendable materials can also be modelled.

In our system we use a mixture of keyframing and automatically generated motion to animate implicit surfaces. We control the way a character moves by specifying keyframes for the articulated figure which defines it and let the system create the motion of the outer layer (skin, flesh, plasticine coating, hair, tail etc.). We used this approach for generating traditional animation effects using implicit surfaces [16]. The flesh deforms during motion according to both physical laws (e.g. gravity) and traditional animation principles [19]. In [16], we implemented automatic generation of *squash and stretch*, *follow through* and *exaggeration*, a few of the rules of traditional animation. As stressed by Lasseter [11], these effects are important in computer generated character animation.

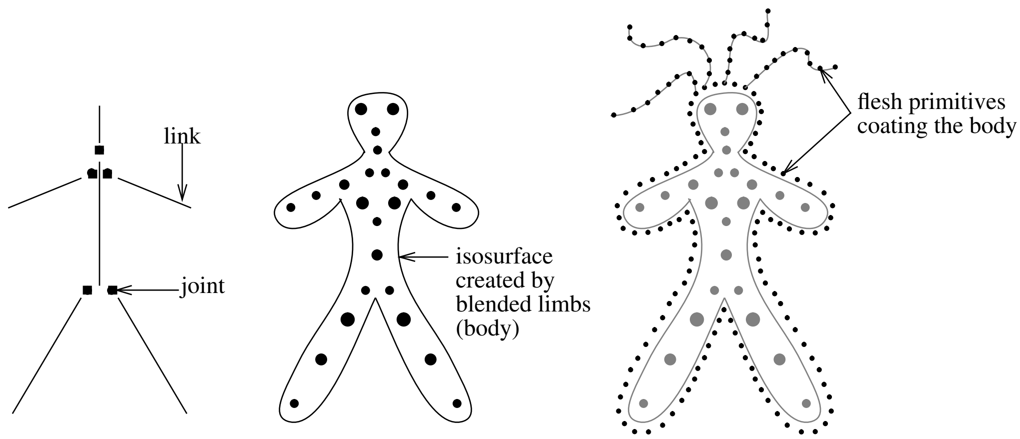

Figure 2. The layers in the model: a graph specifying the articulated figure, a body created from blended limbs designed by a user and flesh covering the body

3. Model

Figure 2 shows the three layers in our model. The graph structure (Figure 2a) specifies an articulated figure defining a character. Each link in the graph represents a limb, which is designed by a user. The designed limbs blended together form the body of the character (Figure 2b). The final layer, flesh, covers the body (Figure 2c).

3.1 Body

Each limb in the body is designed by a user to represent the bone-muscle structure. The shape of a limb remains unchanged during the animation. In the future however, a more complex behaviour might be modelled. Each limb could be subdivided into separate layers of bone, muscle and fat. Then, the use of implicit surfaces eases the simulation of the appropriate deformation. For instance, a flexed muscle could be produced by increasing the radius of one of the primitives in a limb. Similarly, a stretched tendon could be modelled by increasing the distance between the primitives which represent it.

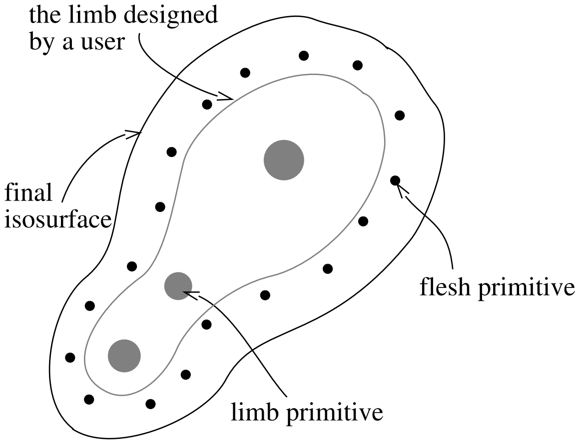

Figure 3. Two kinds of flesh attached to a limb: skin (left) and a strand of flesh (right)

3.2 Flesh

There are two kinds of flesh in the system (Figure 3): skin, which covers a limb and strands of flesh attached to a limb, e.g. hair or a tail.

Skin: Skin around a limb is automatically generated by our system. The algorithm is adapted from the scattering method described in [3], in which particles travel to the isosurface. Initially, for each primitive in a limb (a *parent* primitive), flesh primitives are evenly distributed around it at its radius of influence. Then, they migrate towards the parent until they reach the isosurface. The primitives which reach the parent without intersecting the isosurface are ignored. The rest constitutes the skin. The distribution of flesh primitives is not even. Some recent work [21, 6] proposes adaptive sampling of implicit surfaces which produces a more uniform distribution of sample points. Figure 4 shows the skin generation process for a simple limb.

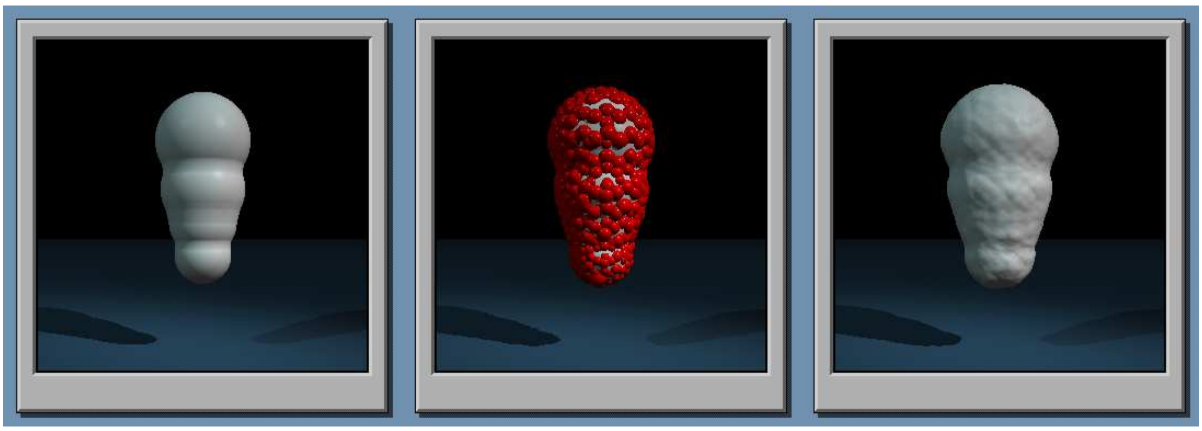

Figure 4. The skin generation process

Strands of flesh: These are modelled as chains of primitives attached to a limb. A user has to specify the parent primitive in the limb, the attachment point and the length of the strand. The system will then automatically create the strand. This kind of flesh is useful for modelling "boneless" parts of a character, e.g. hair or a soft tail. Figure 5 shows the process of creating a strand of flesh.

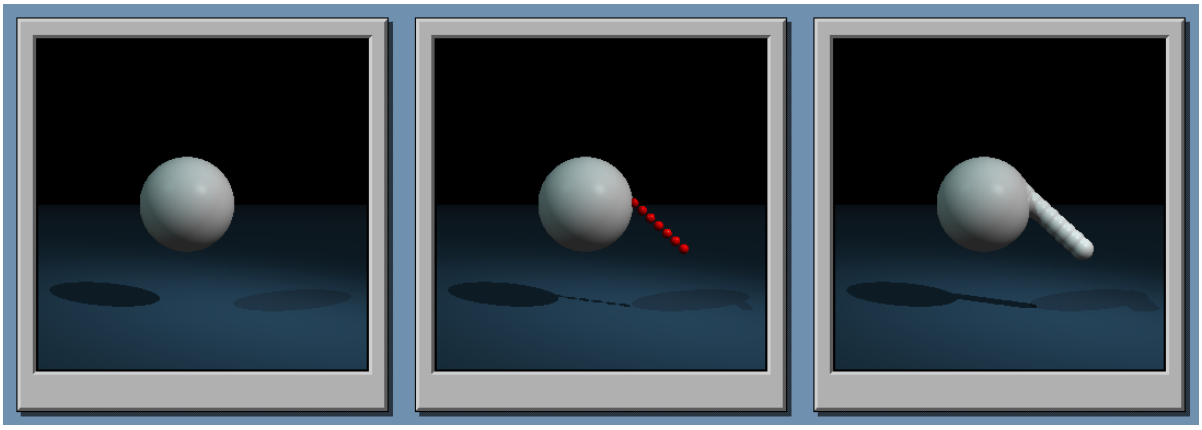

Figure 5. The strand generation process

4. Animation Control

In order to specify an animation sequence we first define keyframes for an articulated figure which defines a character in a script file. The specification language consists of commands describing the movement in Eshkol and Wachmann's dance notation [7]. Our system animates the limbs according to the given choreography. During the animation, the primitives in a limb stay in a constant position in relation to each other. The deformation is achieved through the motion of the flesh. The flesh always follows the limb it belongs to and its motion is automatically calculated by the system. The flesh moves as a result of the body movement, which creates the traditional animation effect of the flesh being dragged behind (Figure 11). Another interesting effect is created when the flesh drops down to the floor and squashes in a big bulge to be stretched later when a character jumps up (Figure 12). A set of parameters (gravity, flesh firmness and exaggeration factor) is provided to give an animator control over the way the flesh motion is generated.

Figure 6. A model of a dancer

4.1 Dance Notation

In their movement notation [7], Eshkol and Wachmann define a model of limbs and joints to represent a human body. They then define movement for each limb using three types of transformations: rotatory movements, plane movements and curved surface movements. Each keyframe position can be described as a sequence of these three types of movement starting from a zero position in which the amounts of rotation for all limbs are zero (Figure 6). This makes Eshkol and Wachmann's notation suitable for immediate computer implementation, since it directly maps into storing local transformation matrices for limbs.

Rotatory movements: These are performed around the longitudinal axis of a limb. For instance, Figure 7 shows a rotatory movement of a head. The parameters which need to be specified are the amount of rotation and its direction (clockwise and anticlockwise). Note that limbs have different constraints for the amount of possible rotation.

Figure 7. Rotatory movement of a head

Plane movements: A limb moves in a plane movement when it describes a plane in space with its longitudinal axis. Figure 8 shows a plane movement of an arm. The amount of rotation, the plane of motion and the direction have to be specified.

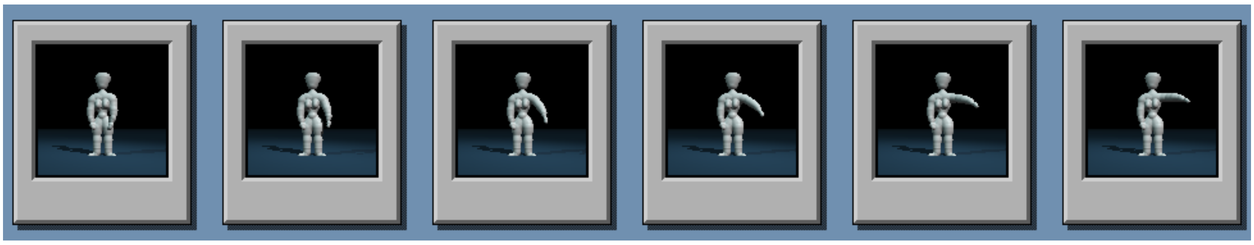

Figure 8. Plane movement of an arm

Curved surface (conical) movements: A limb moves in a curved surface movement when its longitudinal axis describes a curved surface. The simplest example of a curved surface movement is a conical movement, when a limb describes a cone. Figure 9 shows a conical movement of a leg. The amount of rotation, the direction and the cone have to be specified.

Figure 9. Conical movement of a leg

The notation defines the current base of the body, e.g. the feet. The hierarchy in our articulated figure starts from the current base, from where the movement is propagated to the rest of the body. Figure 10 shows a simple dance movement (lean to the side) specified using the dance notation.

Figure 10. A simple dance movement - lean to the side specified using dance notation

4.2 Flesh Motion

The flesh motion is automatically calculated based on the motion of the body. Each flesh primitive has a parent (for skin it is a limb primitive, and for a strand of flesh it is either a limb primitive or another flesh primitive in the strand). The movement of flesh follows the Newtonian laws of motion but it is geometrically restricted to a distance within the range [R_{min}, R_{max}] from its parent (the parent-child connection has properties of a spring). Limb 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 is not within the [R_{min}, R_{max}] range. During its motion, the flesh acquires a velocity from the movement of a limb. When the limb terminates its motion, the flesh continues to move due to this velocity. During this follow through motion, each time a flesh primitive reaches the minimum or maximum distance from its parent (R_{min} or R_{max}), it bounces back with its velocity reversed and dampened.

In addition to the above geometric interaction, the flesh primitives experience attraction/repulsion forces between each other. This introduces irregularities to the appearance of the flesh, which follows the traditional animation rule of avoiding symmetry in order to make a character more interesting. The interaction between two primitives is given by the Lennard-Jones forces implemented after [13]:

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

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.

Figure 11 shows the animation sequence from Figure 10 with the motion of hair (strands of flesh) and skin around one hand calculated by the above method.

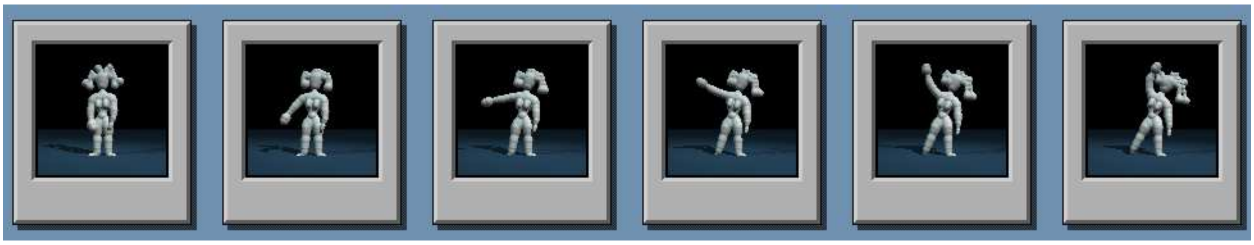

Figure 11. Adding flesh to a simple dance movement (hair and skin on one hand)

4.3 Flesh motion control

The motion of the flesh is controlled by a set of parameters, which can be adjusted by turning each feature up or down. At the moment an animator can alter gravity (none to high), flesh firmness (wooden to very loose) and exaggeration (none to high). Gravity changes the external force applied to flesh primitives during the initial motion calculation. The higher the gravity, the more the flesh is stretched down during its motion. Flesh firmness is translated to [R_{min}, R_{max}] values of the springs which connect the flesh to the body. The looser the flesh, the further it can travel from the body. Exaggeration controls the time the flesh spends in the R_{min} and R_{max} extremes of its motion. Increasing this time results in the flesh being "suspended" before following the motion of the body, hence an exaggerated squash and stretch effect.

Figure 12 presents the effects of altering the above parameters for another short dance movement. A dancer performs a jump in the *first dance position* landing in the *second dance position*. (*Dance positions* describe standard dance poses pictured respectively in the first and the last frame of a dance sequence in Figure 12) The animation sequences show the results of changing the levels of gravity, flesh firmness or exaggeration.

Figure 12. A "jump in first landing in second": the top sequence uses values: gravity =1, firmness =1.5, exaggeration =1; the middle sequence uses values: gravity =2, firmness =1.5, exaggeration =1 and the bottom sequence uses values: gravity =1, firmness =3, exaggeration =3

5. Conclusions

In this work we investigate the use of implicit surfaces for character animation. We focus on animation control and try to balance between manual control (total control left for the animator) and automatic control (the system generates the motion). Our characters are created as articulated figures covered by a body designed by a user as a set of limbs coated with implicit skin and strands of flesh (e.g. hair or a tail). An animator choreographs an animation sequence using dance notation and the system calculates the motion and deformation of the layer of flesh. The model in the examples shown in this paper consists of 64 primitives in the articulated skeleton and 176 primitives in the flesh. Yet only 16 limbs had to be keyframed to produce the results presented in Figures 11 and 12. We are currently working on creating a set of pre-defined standard dance positions which can then be used by an animator to choreograph a dance sequence.

6. 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. Chadwick, J.E., D.R. Haumann and R.E. Parent, "Layered Construction for Deformable Animated Characters", Computer Graphics, Vol. 23, Proc. of SIGGRAPH'89, pp 243-252.
5. Desbrun, M. and M.-P. Gascuel, "Highly Deformable Material for Animation and Collision Processing", Proc. 5th Eurographics Workshop on Animation and Simulation, Oslo, September 94.
6. Desbrun, M. and M.-P. Gascuel, "Adaptive Sampling of Implicit Surfaces for Interactive Modelling and Animation", Proc. 1st International Workshop on Implicit Surfaces'95, Grenoble, April 1995.
7. Eshkol, N. and A. Wachmann, "Movement Notation", Weidenfeld and Nicolson, 1958.
8. Forsey, D.R., "A Surface Model for Skeleton-Based Character Animation", Proc. Eurographics Workshop on Animation and Simulation'91, Vienna, pp. 55-73.
9. 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
10. Guy, A. and B. Wyvill, "Controlled Blending For Implicit Surfaces", Proc. 1st International Workshop on Implicit Surfaces'95, Grenoble, April 1995.
11. Lasseter, J., "Principles of Traditional Animation Applied to 3D Computer Animation", Computer Graphics, Vol. 21, No 4, July 1987, pp 35-44
12. Magnenat-Thalmann, N., R. Laperriere and D.Thalmann, "Joint-Dependent Local Deformations for Hand Animation and Object Grasping", Proc. Graphics Interface'88, pp. 26-33.
13. Miller, G. and Pearce, A., "Globular Dynamics: A Connected Particle System for Animating Viscous Fluids", Vol. 13, No 3, 1989, pp 305-309
14. 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
15. 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.
16. Opalach, A. and Maddock, S., "Disney Effects Using Implicit Surfaces", Proc. 5th Eurographics Workshop on Animation and Simulation (Oslo, 17-18th September, 1994)

17. Sedeberg, T. and S. Parry, “Free Form Deformations of Solid Geometric Models”, Computer Graphics, Vol. 20, Proc. SIGGRAPH’86.

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

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

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

21. Witkin, A. and P. Heckbert, “Using Particles to Sample and Control Implicit Surfaces”, Computer Graphics, Proc. SIGGRAPH’94, pp 269-278.

22. 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

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

24. Wyvill, B., “Metamorphosis of Implicit Surfaces”, Siggraph 90 Course Notes 23, “Modelling and Animating with Implicit Surfaces”

25. Wyvill, B., “Warping Implicit Surfaces”, Proc. Western Computer Graphics Symposium (SKIGRAPH 92), 1992, pp 55-63