Interactive shape design using metaballs and splines

Jianhua Shen and Daniel Thalmann

Computer Graphics Lab.
Swiss Federal Institute of Technology
CH 1015 Lausanne
{shen,thalmann}@di.epfl.ch

Abstract

Implicit surfaces have received increased attention in the design and animation of 3D objects, since they posses some useful advantages over the traditional parametric surfaces. However, interactive design of such surfaces generally suffers from the relatively slow display speed and lack of efficient control facilities. This paper experiments the combination of the implicit surface technique and spline technique, in order to enhance the modeling capability of the implicit surface in practice, particularly in the design and animation of articulated characters.

Keywords: Implicit surface, Modeling, Design, Interaction techniques, Metaball,Soft object, B-splines, Deformation, Animation.

1. Introduction

An implicit surface is a surface consisting of those points p that satisfy the arbitrary implicit function f, f(p) = 0. Because implicitly defined surfaces possess some unique and useful attributes for modeling, such as blending and constraint properties, they have received increased attention in the design of 3D objects[BLOO93]. A particular subset of implicit surfaces, called soft objects [WYVI86] (or metaballs[NISH85], blobs[BLIN82]), is becoming a hot topic in computer graphics. Since metaballs join smoothly and gradually, they give shape to realistic, organic-looking creations, suitable for modeling human bodies, animals and other organic figures, which are very hard to model using traditional geometric methods[GRAV93].

The metaball technique is inherent to interactive design. One can begin with a rough shape consisting just a few metaballs, then add details by simply editing metaballs: add, delete, transform, adjust the parameters of metaballs. However, the metaball technique suffers from two serious drawbacks: first, it requires considerable skill to construct complex objects by properly generating numerous overlapping functions. Second, interactive shape design demands quick feedback when the designer is editing blobs. Unfortunately rendering an implicit surface is often a difficult. task Ray tracing is computational expensive. Polygonalization is either algorithmically complex, or difficult to achieve interactive rates.

In order to enhance the modeling capability of the metaball technique, we address two inter-dependent issues here: to devise some ways to reduce the number of metaballs needed for a desired detail of the shape, and to allow the designer to manipulate blobby surface at interactive speed. Bloomenthal [BLOO90] has discussed ways for interactive display of implicit surfaces. In his approach, fast display of implicit surface, to some degree, sacrifies the level of surface detail. We think shape accuracy is often required in interactive shape design. Beier [BEIE93] discusses the use of a small number of blob primitives for design. Our cross-sectional based isosurface sampling method, combined with the B-spline blending, experiments new ways to achieve above two goals.

1. An interactive metaball editor

We start the shape design by first creating a articulated skeleton for the organic body to be modeled. Metaballs are used to approximate the shape of internal structures which have observable effect on the surface form. Each metaball is attached to its proximal joint, defined in the joint local coordinate system of the underlying skeleton which offers a convenient reference frame for position and editing metaball primitives, because the relative proportion, orientation, and size of different body parts are already well-defined.

We have written an interactive metaball editor for shape designers on SGI Indigo2. The workstation can display shaded or wireframe color metaballs and high resolution skin surface in near real time. Metaballs are displayed as ellipsoids either with effective radius or threshold radius. The "threshold" mode shows the visible size of metaballs, while the "effective" mode shows how far the metaball will influence. Each metaball is treated separately and does not blend with each other, because we can interactively display the B-spline surface which is constructed from blended metaballs(refer section 3). Some widget panels are used to interactively adjust the size, weight, position, orientation of metaballs. Spaceball or trackball enables the user to rotate models around in space for different viewing. By turning on/off various display entities of different layers, the designer can selectively check skeleton, metaballs, cross-section contours, and skin envelope simultaneously. The designer can interactively create, delete, pick, joint attach/detach metaballs. A file format is designed which can store both joint hierarchy and metaball information. Models can be saved into a file and loaded in later for successive sculpting. The designer can get quick feedback of the resulting skin form.

2. Cross-sectional sampling

An important issue in shape design applications is to reduce the complexity of the 3D interaction. The complexity of free-form surface construction and manipulation can often be reduced with the method of cross-sectional design. The principle of cross-sectional design is to reduce the surface construction problems to the level of curve analysis. We can use this approach to construct human, animals, and other organic shapes, because they have in common a skeleton framework upon which muscle, fat and flesh tissue are supported, all clothed in a skin. Their limbs exhibit a cylindrical topology and the underlying skeleton provides natural centric axis upon which a number of cross-sections can be easily defined.

Each limb link is associated with a number of cross-section contours. We can automatically extract cross-sectional skin boundaries using the ray casting method, as illustrated in figure 1.

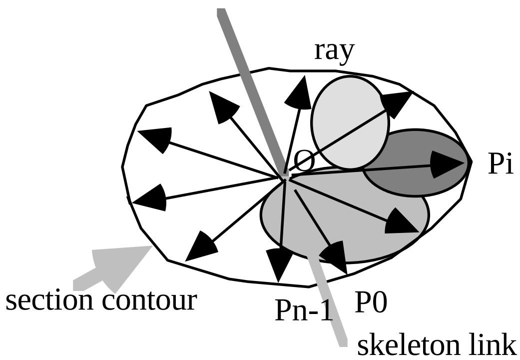

Figure 1. Ray-casting sampling of cross-sectional contours.

We cast rays in a star-shaped manner in a contour, with ray origins sitting on the skeleton link. For each ray, we compute the outmost intersection point with the blobby isosurface surrounding the link. The intersection is a sample point on the cross-section contour. By repeating this intersection procedure through each ray and contour, we get a series of discretized contours of the body.

We use the angle between two links to drive the configuration of the cross-section planes In figure 2a, L1 is the direction of upper link, L2 the lower. Nu and Nl are precomputed section normals at link ends. We set the orientation of the cross-section plane at the joint position as the bisection plane of the two links. Let this bisection plane normal be N0. Suppose Oi, Ni be the center and normal respectively of i-th cross-section plane along the upper link. Oi is evenly distributed along upper link by linear distance interpolation of upper link length, Ni by direction interpolation of end normals N0,Nu. For each point on i-th contour to be sampled, we cast a ray with origin Oi. The direction Dj is calculated by circle rotate along Ni axis as shown in figure2b.

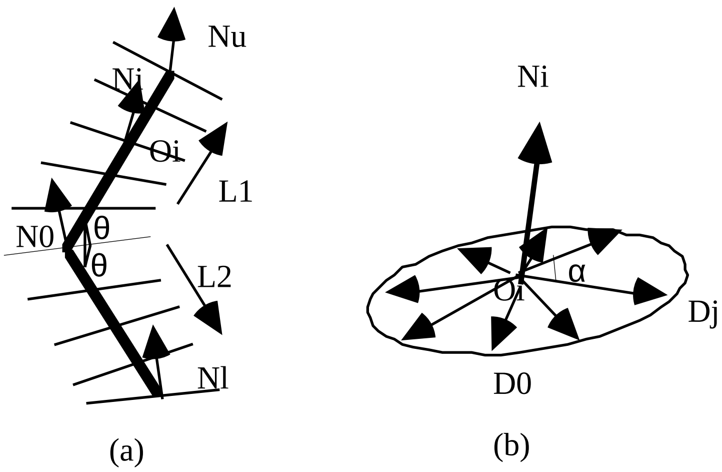

Figure 2. cross-section orientation and ray distribution

3. B-spline smooth blending

We associate m_k contours, and each contour contains n_k points for a body part. Let us denote P_{ij} is the i-th sample point on j-th contour, where i=0,..., n_k -1, j=0,..., m_k-1. We simply construct a bicubic B-spline patch with control points corresponding exactly to the sample points associated with skin contours:

$$S_k(u,v) = \sum_{i}^{m_k-1} \sum_{j}^{n_k-1} B_{i,4}(u) B_{j,4}(v) P_{ij}$$

As twice smooth blending(potential filed blending and B-spline blending) has been applied in our method, our experiments indicate that an equivalent detailed model build with our technique uses considerable fewer primitives than that required by simply potential filed blending of metaball technique.

4. Discussion

4.1 Deformation artifact

It is a common knowledge that a smooth, natural-looking B-spline patch requires a regular, evenly arrangement of its control points in 3D space. Abnormal arrangement of control net will usually lead to anomalies in its surface shape. When the joint angle between two neighboring links is not very small, our method guarantees the evenly distribution of cross-section orientations, and the sampling points inside each contour, so the B-spline patch obtained has a nice and natural look. Because the sampling is undertaken dynamically when the underlying skeleton is in motion, the shape explicitly constructed has already accommodated movements and deformations.

However, when the joint angle becomes smaller, the implicit surface inflates due to the blending process. This artifact can be reduced by grouping metaballs. Beier[BEIE93] have discussed this technique to avoid unwanted blending. We take the deformation of our human leg model as an example to limit the artifact in practice. See figure 3. Metaballs of the leg are assigned in two groups: the *upper leg group* and *lower leg group*., classified by their attachments. Several small metaballs, located near the joint, belong to the two group simultaneously, as they have contributions to both upper and lower legs. To compute the upper leg cross-sections, we calculate the ray-blobby isosurface intersections by only considering metaballs in the *upper leg group* . The lower leg cross-sections are treated in the same way.

Another artifact associated with small joint angle: the neighboring cross sections near the articulation intersect, producing overlapping in the B-spline control mesh. However, this does not cause serious problems in practice. First, due to the structure of anatomy, the joint angle has a limitation for human and animal limbs. Thus it can not be too small. Second, even if the neighboring cross sections near the articulation intersect, unless the overlapping region in the B-spline control mesh is too big, the visual appearance of the skin surface is generally acceptable, as long as we just care of the appearance of the outer surface shape during animation, and do not mind if the surface intersects itself internally. As the neighboring cross sections move closer to one end, the B-spline surface constructed will produce wrinkles around the joint. Third, we can limit the overlapping artifact by using *deformable metaballs*. They are the metaballs whose axis lengths and origins change according to current joint angle value. By adjusting the metaball parameters in the limit case to reduce overlapping, we use linear interpolation of deformable metaball parameters between the limit case and the relaxed case during animation. However, the complete elimination of the artifact in the limit case appears to be very difficult. See left of figure 3.

4.2 Texture mapping

In our method, the skin surface is regenerated for each frame in animation. As the skeleton moves, the attached metaballs transform and deform accordingly, hence the blobby surface continuously changes. We reconstruct the skin surface in each frame to reflect those changes. Although the skin surface is changing all the time, the mapping of (u,v) parameter to the skin surface point is relatively fixed in the relationship with skeleton, because the center of cross-section is fixed along the skeleton link, and the number of control points is fixed. Texture mapping of B-spline surface is straightforward, since a parametric patch, by definition, already possesses u-v values over its surface.

4.3 Comparison with Generalized cylinders

Our method of cross-sectional construction of B-spline surface from implicit defined blobby surface bears a conceptual resemblance with *Generalized cylinders* (GCs) [BINF71, SHAN84]. Both representations have a main axis, upon which cross-sectional curves are defined. Our approach has much fewer constraints and is simple to implement. In GCs, the main axis is an open 3D curve, and it should pass through the centroids, perpendicular to cross-sections. Our main axis is a straight line (a link of the underlying skeleton), which needs not pass through centroids or perpendicular to cross-sections. In GCs, a cross-sectional curve is defined on the cross-sectional plane relative to the plane-local coordinate systems. The *orientation problem* (converting points in the 2-D definition of a cross-sectional curve to their appropriate locations on the GCs surface) is mathematically complex and may introduce ambiguities and singularities. In our method, cross-section curves are arbitrarily oriented B-spline curves, built from discrete sample points which are computed directly in the world coordinate system, thus eliminating the *orientation problem*. Finally, in our method, there is no sweeping rule or interpolation scheme for cross-sections, discrete sample points on a cross section are the coefficients of the basis functions of the B-spline surface for approximating the shape.

5. Example

Figure 4 shows a dolphin modeled by twelve metaballs. Figure 5 shows a man's leg modeled by 15 metaballs. Figure 6 shows a leg composited by twelve metaballs. Some deformation results are also presented. Note: all the metaballs in the pictures(except figure 5b) are not blended, GL rendering refers SGI Graphics Library rendering.

6. Conclusions and Future work

We have discussed the combination of metaball technique and spline technique for interactive shape design and animation. An explicit spline surface construction overcomes two main limitations inherent to metaball techniques: the integration with surface-based environment and the texture mapping, hence greatly enhancing the modeling capability of metaballs.

We have extended our method to human body modeling and deformations[SHEN95]. B-spline surfaces are tessellated into polygon mesh for welding different body parts together. However, B-spline surface is not a natural choice for arbitrary topology. To smoothly join different skin parts together, many boundary constraints and ad hoc tricks have to be applied for branching topology such as human and animals. Future work should explore the possibility of using spline surface over irregular meshes , such as presented by Charles Loop[LOOP94].

References

[BEIE93] Thaddeus Beier, "Practical Uses for Implicit Surfaces in Aninamation", in *Siggraph'93 course notes* 26

[BINF71] Binford, T. in *Visual Perception by Computer, Proc. of the IEEE conference on systems and Control* , Miami, FL, December 1971.

[BLOO90] Bloomenthal, J. and Wyvill B, "Interactive Techniques for Implicit Modeling", *Computer Graphics* 24, 2, 109-116(May 1990).

[BLOO93] Bloomenthal, J. et. al, "Modeling, Visualizing, and Animating Implicit Surfaces", *Siggraph'93 course notes* 26

[BLIN82] Blinn, J., A Generalization of Algebraic Surface Drawing, *ACM Transactions on Graphics*, Vol.1, pp.235-256

[GRAV93] G.L. Graves, "The Magic of Metaballs", *Computer Graphics World*, pp.27-32, May, 1993

[LOOP94] Loop, C. "Smooth Spline Surfaces over Irregular Meshes", Proc. of *SIGGRAPH'94.*

[NISH85] H. Nishimura, M. Hirai, T.Kawai, et al ."Object modeliing by distribution function and a method of image gerneration" *Trans. IECE, J68-D(4),718-725(1985)* Journal of Papers given at the Electronics Communications Conference 1985(In Japanese).

[NISH94] T. Nishita, E. Nakamae. " A method of displaying metaballs by using Bézier clipping", pp.271-280, *Eurographics'94*

[SHAN84] Shani, U. and Ballard, D.H. "Splines as Embeddings for Generalized Cylinders", *Computer Vision, Graphics and Image processing*, 27(1984),129-156.

[SHEN95] J. Shen, R. Boulic, T. Molet, N. mMagnenat-Thalmann, D. Thalmann, "Realistic Human Body Modeling and Deformations using Metaballs and Splines" (submitted paper).

[WYVI86] Wyvill, G, McPheeters, C. and Wyvill, B. Data Structure for soft objects, *The Visual Computer*, Vol.2, No.4.

Figure 3: human leg deformation when the joint angle are small. Upper: GL rendering of polygonized B-spline surface; Lower: metaballs and control net of B-spline patch.

Figure 4. Dolphin model, the tail in (d) rotates from vertical to horizontal

Figure 5: a) metaball model b)ray tracing with Murskami's field function c) B-spline control net d) GL rendering of B-spline surface

Figure 6: Leg deformation examples