Hierarchical Implicit Surface Refinement

Xikun Liang Brian Wyvill

Dept. of Computing Science
University of Calgary, Calgary, Alberta, Canada
xkliang@cpsc.ucalgary.ca blob@cpsc.ucalgary.ca

Abstract

A novel method of hierarchical implicit modeling is presented, in which an implicit object is modeled by using a hierarchy of implicit surfaces. The hierarchy provides both layered local refinement and global deformation. Local refinement allows the introduction of higher-level detailed surfaces. Global deformation changes the overall shape of the surface while maintaining the integrity of surface details. The model is gradually refined by introducing appropriate new primitives in the specified surface areas. Refinement constraints, such as local surface area and level of the surface (within the hierarchy) are designed to be applied to the implicit object so as to achieve finer control over the local surface. The method provides a dynamic representation of implicit surfaces and can be used not only in modeling complex implicit objects, but also in animating the surfaces and simulating the deformations of various objects.

keywords: implicit surfaces, implicit modeling, surface refinement, global deformation, animation.

1 Introduction

Implicit surfaces have received increased attention in recent years [3]. They are a natural method for representing solid objects particularly when blending between primitives is required. Skeletal implicit surfaces have the advantages that the skeleton can articulated and very realistic models can be built and animated. Models of realistic natural objects such as sea shells ([9]) require a procedural definition or a developmental processes, and replication with alterations, all of these features can be supported by skeletal implicit modeling.

Both point primitives and non-point primitives such as line, polyline, curve, polygon, solid, etc., have been used in various implicit modeling applications. Complex models can be built with CSG boolean operations and field warping functions [3, 26], providing implicit surface the capability of building engineering objects. In addition, skeletal-based implicit modeling [2] and various well-defined field functions [1, 3, 19, 29] provide control for implicit surfaces. However, with the current implicit modeling techniques, once the model is defined, it is very difficult to do further local refinement and global deformation without having to re-design the whole model. There is no successful approach to arbitrarily adding details to subsections of the surface and changing the overall shape of the surface while maintaining the complete surface integrity. In particular, refining the implicit surface directly instead of working on the polygon mesh remains a challenge.

We present a new, intuitive method for implicit modeling based on implicit primitives. Instead of modeling implicit objects with shape transformations [12, 15], refinement of the polygon mesh [10, 14, 21, 24] or implicit particles [22, 25], we directly manipulate the implicit primitives. This new method provides a way to build a model in a hierarchy of implicit surfaces with both local refinement and global deformation. Since the accuracy of the surface interaction is critical in the case of refinement, we obtain the precise interaction by ray-tracing the surface at the interaction area. Constrained by the local surface area and level of the surface, the model is progressively refined by introducing primitives locally, while maintaining the global properties of the other parts of the surface. With a global deformation at a lower level of the surface hierarchy, the overall shape of the surface can be changed while maintaining the integrity of the higher levels surface (Figure 8 and 9).

1.1 Previous Work

Hierarchical surface refinement [7, 8] lends itself to better shape control in hierarchical B-splines than in traditional B-spline surfaces. The main advantage is to allow the addition of more patches to a particular region rather than across the entire surface. Moreover, it provides local refinement and the ability to change the overall shape of the surface without having to re-edit (or re-animate in the case of animated surfaces) the details on that surface. Motivated by this concept, we propose a new method for implicit surface refinement.

Most techniques for implicit modeling are based on skeletal primitives such as point, line, circle, polygon, etc., these primitives are blended together to obtain a smooth surface. To represent solid object, CSG operations [26, 3] are introduced into implicit modeling which makes implicit surfaces even more attractive.

Skeleton based methods, [4] provide the main stream of implicit modeling. It associates each skeletal element with a locally defined implicit function. Individual functions are blended using a polynomial weighting function that can be controlled by the user. This has been widely accepted as a basic implicit surface modeling approach. Physically-based particle systems [25] uses constraints to sample and control implicit surfaces. A simple constraint locks a set of particles onto a surface while the particles and surface move. For implicit local refinement and global shape control, several approaches have been proposed. Galin [10] updates the surface meshing incrementally based on a recursive decomposition of space that focus on regions where changes in the potential field f occurred. Van Overveld and Wyvill [24] use shrinkwrap to refine the polygonal mesh for manifold objects. Guo [12] proposed a method to control the shapes of implicit patches through manipulating their control points using Bernstein-Bezier representation of polynomials. The shape of the model is controlled by maintaining convexity of the patch. For implicit surface reconstruction, Muraki [17] introduced a "Blobby" model for automatically generating a shape description from range data. The model expresses a 3D surface as an isosurface of a scalar field which is produced by a number of field generating primitives or skeletons. The method is computational intensive due to the optimization process. Using generalized metaballs as deformation constraints, Jin [15] developed an approach to control the shape of the surface. All these methods cannot directly control the local detail of the implicit surfaces and maintain the integrity of local details during broad-scale changes. In particular, although based on skeletons, most methods don't directly manipulate primitives, which makes modeling and animating complex implicit models much more difficult when building primitives that need to be changed afterward.

1.2 Overview

This paper presents a new method for implicit surface refinement and global deformation. The main goal is to create implicit objects by modifying the surface directly and dynamically based on primitives. Each model is associated with a hierarchy of implicit surfaces. The advantage of using this hierarchy is that both hierarchical local refinement and global surface deformation can be performed dynamically at any level of the surface. Constrained by the local area and level of the surfaces, the method leads to an easy control over the implicit surface and provides an intuitive way of representing deformed object. In addition, the precise interaction with the surface is achieved by ray-tracing the implicit surface at the surface interaction area.

The remaining sections are organized as follows: Section 2 describes implicit modeling, the general idea for implicit surface refinement, and the *BlobTree*. The method of hierarchical implicit surface refinement is detailed in Section 3, including the hierarchical representation of implicit surfaces, the implicit surface local refinement and global deformation. Section 5 discusses applications of this method in modeling and animation. Section 6 presents our conclusion and future works.

2 Implicit Modeling

In this section, we present the background of implicit modeling. First we give a definition of implicit surface. Then the implicit surface refinement methods are reviewed. Finally we introduce the *BlobTree* structure which provides an implicit modeling system through a hierarchy of blends, CSG and warping applied to primitives.

2.1 Implicit Surfaces

An implicit surface S is characterized as the points of space whose potential $f(x, y, z)$ equals a threshold value

denoted by T.

$$S = \{M(x, y, z) \in \mathcal{R}^3, f(x, y, z) = T\}$$

The primitives are generally points, lines, polygons etc. These techniques were introduced by Blinn [1], and subsequently developed by Wyvill et al. [29] and Nishimura et al. [19]. In this paper, we address implicit surfaces built from skeletons or primitives, also known as *soft objects* [29]. A soft object is described by its scalar field $f(x, y, z)$, that is generated by summing the influences of scalar field elements $f_i(x, y, z)$ associated to their skeletons S_i.

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

In general, the field contributions f_i are decreasing functions of the distance to a primitive such as in [29].

2.2 Shape Control of Implicit Surfaces

Shape control in implicit modeling has proven difficult because small changes to implicit primitives result in global changes to the implicit surface. Efforts have been made to address this problem. One such method is the skeletal method [4], which associates each skeletal element with a locally defined implicit function. Individual functions are blended using a polynomial weighting function that can be controlled by the user. A procedural implicit function permits a greater degree of localized control as compared to a simple blend of implicit primitive in which each primitive has a global affect on the surface. However, interaction with complex procedures is somewhat awkward. Skeleton-based implicit modeling is effective and widely used. Therefore, our method is based on skeletons. The basic idea is to add new primitives into the model where the local surface is being refined.

In order to guarantee the topology of the implicit surface, [14] developed a polygonization technique which can directly and accurately manipulate polygonized implicit surfaces by using techniques from catastrophe theory and Morse theory. Another method uses the Bernstein-Bezier representation to control the shapes of implicit patches [12, 20]. In particular, Guo [12] proposed to control the implicit shape by the convexity of quadric patches or cubic patches.

An incremental technique [10] describes an incremental polygonization approach for implicit surfaces built from skeleton elements. Based on a recursive decomposition of space that focus on regions where changes in the potential field f occurred, the method can interactively update the surface meshing where needed. An alternative method, shrinkwrap [24] creates an initial coarse polygonal mesh and then adaptively refines the mesh to obtain a better approximation of the object.

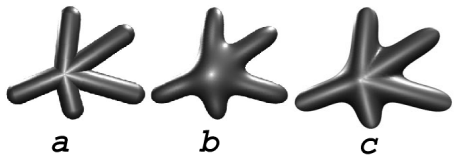

Figure 1: Union (a), convolution (b), and combined (c) surfaces

To control the blending bulge, Bloomenthal[2] combines union surface Figure 1a together with convolution surface 1b as shown in Figure 1c. Other shape control methods for implicit surfaces include controlled blending [6, 13] and precise contact modeling [11]. Since blending a very small primitive with a large one fails with the normal blending method, Wyvill [28] proposed restricted blending which modifies the blending distance of one primitive (A) to match that of the other (B). The field function of A is scaled, then translated, resulting in the field function shown in Figure 2. The field function is finally clipped so that $f(r) = 1$ if $r < (R_0 - R_1)/2$ and $f(r) = 0$ if $r > (R_0 + R1)/2$. This leads to a better blending between two primitives. Dealing with more primitives is still an open problem.

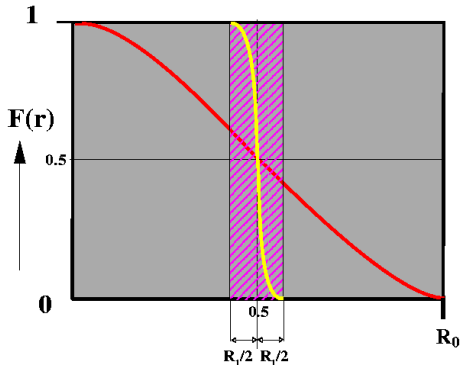

Figure 2: The modified field function used for larger sphere for the restricted blend.

All the methods mentioned for building implicit

surfaces are basically dealing with a static model. The refinement is based on the polygonization mesh, either controlling the mesh points or updating the mesh as needed. Intuitively, we wish to build a dynamic model which directly controls the implicit objects and refines the surface with skeletons or primitives as the modeling process continues. In addition, the method should be able to undergo global deformation without having to re-edit all the detailed surfaces.

2.3 The *BlobTree*

The *BlobTree* [26] provides a hierarchical data structure for the definition of complex models built from implicit surfaces, CSG Boolean operations and field warping functions. The model is defined as the hierarchical composition of multiple objects. 2D textures has also been applied to the *BlobTree* as described in [23, 30]. Moreover, it can be extended to include some other implicit attributes as well. Figure 3 shows an example built from *BlobTree* with 2D texture mapped on the *BlobTree* surface.

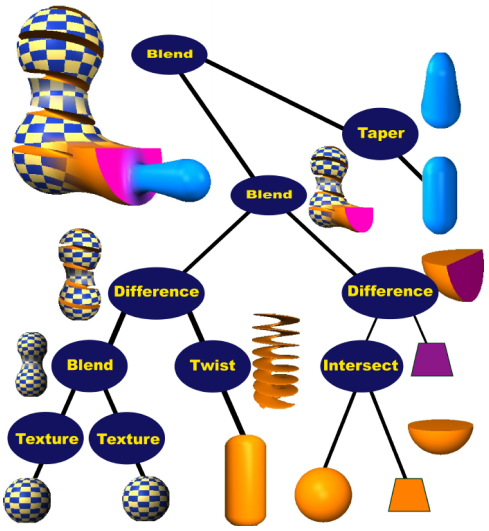

Figure 3: Example of a model built up from a *BlobTree*

The *BlobTree* widely opens the application areas of implicit surfaces. It can be used to build complicated soft and engineering objects. However, since the *BlobTree* is basically a static model, once defined, it is difficult to change. To further extend the use of *BlobTree* , we wish to dynamically introduce implicit primitives and change the overall shape of the surface as required. We'll present a method which enables modifications to the tree at the specified local surface area.

3 Hierarchical Implicit Surface Refinement

Our method of implicit surface refinement works directly on the surface and primitives. we build a hierarchy of implicit surfaces representing the object. Each surface at a hierarchy node represents both a local surface relative to its parent surface and a global surface relative to all its sub-surfaces. In this section, we first describe the hierarchical representation of implicit surfaces. Then we introduce ray-tracing based direct surface interaction, generation of surface details, local refinement, and global deformation.

3.1 Hierarchical Representation of Implicit Surfaces

For a given implicit object, we create a hierarchy of implicit surfaces representing the object. In the hierarchy, each surface is a node in the tree, either an internal node at the coarser level or a leaf node at a finer level. Each hierarchy contains a local surface and a set of sub-hierarchies. The surface can be any implicit surface which is composed of a set of primitives: $P^i = \{p_k | p_k \in level\ i\}$. It is either a leaf surface relative to the higher level hierarchy or an internal surface relative to the surfaces of its sub-hierarchies. Each sub-hierarchy represents a local surface detail which is created in its own local coordinate system. Surfaces related to the same surface being refined are at the same level of hierarchy. With this hierarchical representation of implicit surfaces, implicit surface represented at level i of the hierarchy consists of the local surface S^i_{global} and all the surfaces of its sub-hierarchies $\sum S^i_{sub-surface}$:

$$S^i_{surface} = S^i_{global} \mathcal{OP}_{refinement} \sum S^i_{sub-surface}.$$

where $\mathcal{OP}_{refinement}$ represents the operation of the hierarchy. Figure 4 shows the hierarchical structure of implicit surfaces.

According to the refinement operations, four kinds of hierarchies are designed:

- Blending
- Controlled Blending

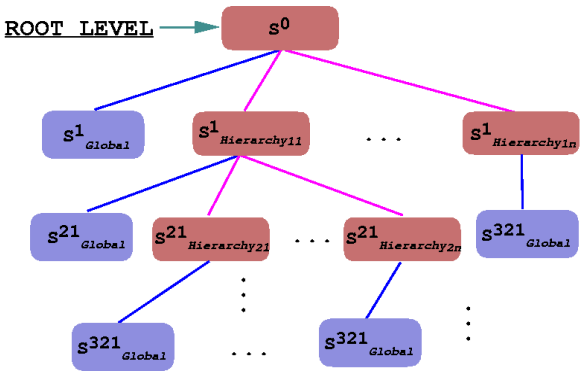

Figure 4: Hierarchy of the implicit surface

- Precise Contact Modeling
- CSG

They represent four different operations to combine the global surface with the details. Blending uses super-ellipsoid blending method described in section 2.1. Controlled blending deals with the blending properties of the surfaces. Precise contact modeling combines the local surface and the surface details using the method proposed in [11]. CSG operations perform intersection, difference, or union on the local surface and the refined surface.

1. **Blending Hierarchy** contains a surface and a number of sub-hierarchies. It represents an implicit surface which blends its local surface and all the surface details of its sub-hierarchies using the implicit model defined in formula 1. The surface of blending hierarchy can be described as :

$$S^i_{surface} = S^i_{leaf} + \sum S^i_{sub-surface}.$$

2. **Controlled Blending Hierarchy** consists of a surface and a number of sub-surfaces which are blended with their parent surface, but not with each other. We have chosen to use the controlled blending technique from Guy [13] because of its simplicity. Other method such as Desbrun [6] may be applied as well. The surface represented by this hierarchy is:

$$S^i_{surface} = S^i_{leaf} + \sum ct(S^i_{sub-surface}).$$

where $ct(S^i_{sub-surface})$ represents the controlled field value from the sub-hierarchies.

3. **Precise Contact Modeling hierarchy** is composed of a surface and a sub-hierarchy which represents the surface detail. Based on the technique of precise contact modeling [11], this hierarchy represents two separate surfaces which are precisely contacted. Similarly we model the contact surface by considering both interpenetration region and propagation region of the contact surfaces. Therefore the surface of a precise contact modeling hierarchy can be described as:

$$S^i_{surface} = f^i_{surface} + g^i_{surface}$$

$$S^i_{sub} = f^i_{sub} + g^i_{sub}$$

where $f^i_{surface}$ and f^i_{sub} are the field functions for the surface $S^i_{surface}$ and S^i_{sub} respectively. $g^i_{surface}$ and g^i_{sub} are the deformation fields for the surface and the sub-surface which are defined as follows:

$$g^i_{surface} = \begin{cases} iso - f^i_{sub}, & interpenetration \\ g^i_{propagation}, & propagation \end{cases}$$

$$g^i_{sub} = \begin{cases} iso - f^i_{surface}, & interpenetration \\ g^i_{propagation}, & propagation. \end{cases}$$

Where *interpenetration* and *propagation* refer to the corresponding region. Details of choosing the propagation function $g^i_{propagation}$ can be referred to Gascuel [5, 11].

4. **CSG hierarchy** deals with refinement related to CSG operations such as intersection, difference, and union. It contains a surface and a sub-hierarchy representing another implicit surface. The surface of this hierarchy is therefore

$$S^i_{surface} = S^i_{leaf} \mathcal{OP}_{CSG} S^i_{sub-surface}.$$

where $\mathcal{OP}_{CSG}$ is intersection, difference, or union.

3.2 Construction of the Hierarchy

We start with an initial implicit surface and create a root hierarchy. A sequence of sub-hierarchies then can be built by recursively refining the selected surfaces. Given a description of the surface detail, a sub-hierarchy S^{i+1}

at level i is defined by applying a splitting scheme which searches the related surfaces according to the refinement constraints (section 3.5) and creates a new surface at level $i+1$, i.e.

$$S^{i+1} = \{p_k | p_k \in split(S^i_{surface})\}.$$

Each hierarchy, except the root hierarchy in the world coordinate system, is constructed at its local coordinate system relative to its parent hierarchy. The local coordinate system is determined by the center of the given surface refinement area and the refinement constraints.

With this hierarchy of implicit surfaces, the final object is obtained by combining all surfaces in the hierarchy according to the different hierarchical properties. A bottom-up recursive surface construction strategy is applied. Lower level surfaces are processed first. Then they are combined with higher level surfaces using blending, controlled blending, precise contact modeling, or CSG operations. Due to the hierarchical representation of the surfaces, local details can be maintained according to the refinement constraints.

The general algorithm for implicit surface refinement is as follows:

- Define constraints if necessary,
- Pick the surface points and define the displacement,
- Search the *hierarchy* to locate the surfaces which have the most influence to the selected points,
- Generate new primitives to approximate the local surface detail,
- Introduce new hierarchy at the given level based on the constraint,
- Repeat previous steps to build the complete model,
- Render the hierarchy to get the implicit surface.

Here, step 2 and 3 are used for interactive modeling only. Constraints can be specified as either local area or surface level. The following sections provide further details.

3.3 Direct Implicit Surface Interaction

Polygon mesh has been extensively used for implicit surface deformation, visualization, animation, modeling etc. The implicit object is manipulated through the control of the meshing points [10, 14, 15, 21]. To finely interact with the local surface, we use ray-tracing, instead of a polygon mesh, to locate the exact position on the surface for picking and selecting operations.

Direct implicit surface interaction includes pick operations and displacement of the surface. To be precise, a ray-tracing technique is used to convert each selected point (x, y) (screen coordinate) to a surface point in the object coordinate system. The surface normal is also obtained and used to control the displacement of the surface. The displacement of the surface point P to a new surface point Q is then used to determine the parameters of the local surface detail, for example, the range of influence and other transformations expected for the surface. If any of the refinement constraints is defined, the range of influence of the new surface will be constrained within the local area or the specified level of surface.

3.4 Generation of Surface Details

Surface details are introduced according to the refinement constraints. In order to add surface details into the hierarchy, we first specify the refinement constraint and then traverse the hierarchy to obtain all surfaces contributing to the refinement (surface) area. Based on the refinement constraints, new surface is then created and added as a sub-hierarchy at the chosen level. If local surface area constraint is defined, the boundary of the area, either surface or volume, determines the range of the influence of the new surface. Similarly, level constraint restrict the refinement on the specified level of the hierarchy.

A simple splitting technique is to introduce a new primitive for each refinement while keeping other primitives within the group unchanged, i.e.

$$split(S^i_{surface}) = p_{new-primitive}$$

The properties of the primitive such as radius of influence, displacement vector, and potential function, are determined based on the constraints. Figure 5 shows a

simple case using this scheme. The refinement introduces a spike based on local area constraints, displacement vector, radius of influence, and some tapering and bending constraints.

Figure 5: Surface refinement with a simple splitting scheme

This simple splitting method provides interactivity as well as satisfactory result in most of the cases. But the group of primitives may not be the optimal group that models the local surface. Redundant primitives may exist after a series of refinements. [17, 18] proposed a more complicated method for fitting 3D range data with *Blobbies*. In this case, a better shape description is required to define the refined surface in order to find an optimal group of primitives approximating the surface. This optimization process, however, results in a serious computational overhead. Therefore, we are investigating a better scheme to achieve both interactivity and accuracy.

3.5 Local Surface Refinement

Since implicit surfaces possess blending and constraint properties, they lend themselves to local surface refinement. The range of influence of an implicit primitive is characterized by its parameter R which defines the influence boundary of the primitive.

3.5.1 Local Surface Area Constraints

With the concept of *overlay and offset referencing*, Forsey and Bartels [8] present the local deformation and local surface refinement on a B-Spline surface. The idea is to designate a patch on the surface at any level of refinement and execute a new refinement step to re-represent this patch. Also they refine a surrounding number of patches sufficiently to include the area influenced by any refined control vertex that are to be manipulated. Such a fine control over implicit surface can be achieved by controlled blending [13, 28] and using implicit surface properties such as field function, radius of influence, etc. In particular, by limiting the range of influence of the primitives, the refinement can be restricted within a specified region.

Such a region on the surface defines an area constraint in our local refinement. We select a set of surface points to define a closed surface area. Subsequent refinement will only take place within this local area. The constraint also determines the range of influence of the new surfaces so that the new primitives will only have contributions to the local area. The maximal value of the radius of a new point primitive, for instance, will be the radius of influence for the local area or the minimal distance between the selected surface point to the boundary of the local volume.

3.5.2 Level of the Hierarchy

Since the object is represented in a hierarchy, level information i.e. the level in the hierarchy, can be used for local refinement as well. With the level constraint only the specified level and its related finer level of surfaces will be influenced by the newly introduced details.

3.6 Global Deformation

This section describes the global deformation based on the hierarchy of surfaces. Global deformation here refers to the deformation of any surface contained in an internal node of the hierarchy. Its main purpose is to change the overall shape of the surface at some specified layer while maintaining the consistency and integrity of the local details. Two commonly used types of global deformations are defined, which are the affine transformations and space warping operations. The major issue involved in global deformation is to automatically adjust the sub-surfaces without having to re-edit or re-define the details. We use a simple and common strategy to achieve this. After the deformation of the global surface, we adjust the local coordinate system of the sub-hierarchies to keep the consistency of the surfaces. In the following sections, we assume that:

- $\mathcal{C}$, the original coordinate system,
- $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, the normal vectors of the major axises of $\mathcal{C}$,
- $\mathcal{O}$, the world coordinate of the origin of $\mathcal{C}$
- $\mathcal{C}'$, the deformed coordinate system,

- $\mathcal{X}', \mathcal{Y}', \mathcal{Z}'$, the new normal vectors of the major axises of $\mathcal{C}'$,
- $\mathcal{O}'$, the world coordinate of the origin of $\mathcal{O}'$.

where $\mathcal{X}$ is a normalized tangent vector of the surface at $\mathcal{O}$, $\mathcal{Y}$ is the normal of the surface at $\mathcal{O}$.

3.6.1 Affine Transformations

Affine transformations are the most commonly used global deformations. They change the overall shape of the object through translation, rotation, scaling, or shearing. A 4×4 matrix $\mathcal{T}$ is made to represent an affine transformation or a composition of these transformations. Given a surface in the hierarchy and an expected transformation $\mathcal{T}$, the new coordinate system for any sub-hierarchy of the surface can be obtained as follows:

$$\begin{aligned} \mathcal{O}' &= \mathcal{O} \times \mathcal{T} \\ (\mathcal{X}', \mathcal{Y}', \mathcal{Z}') &= (\mathcal{X}, \mathcal{Y}, \mathcal{Z}) \times \mathcal{T} \end{aligned} \tag{2}$$

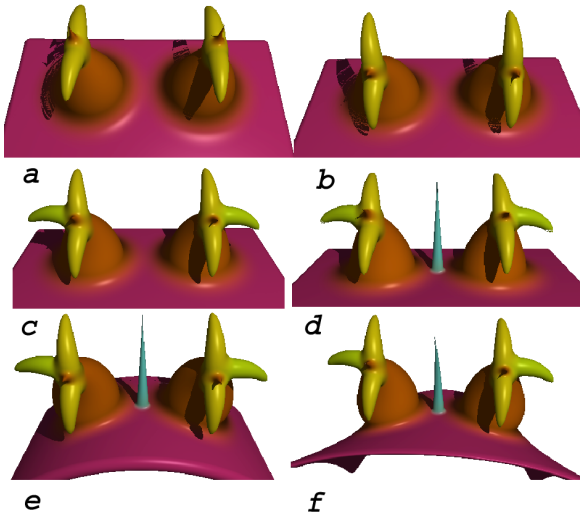

Figure 6: Local refinement and global deformation of the surface

3.6.2 Space Warping

Space warping operations have proved very useful for implicit modeling [26, 27]. After the warping is applied on the surface, the warped layers are then defined in the following coordinate system:

$$\begin{aligned} \mathcal{O}' &= warp(\mathcal{O}) \\ \mathcal{X}' &= \partial f(\mathcal{O}')/\partial x \\ \mathcal{Y}' &= (\partial f/\partial x, \partial f/\partial y, \partial f/\partial z) \\ \mathcal{Z}' &= \mathcal{X}' \times \mathcal{Y}' \end{aligned} \tag{3}$$

where $warp()$ could be any warping function such as bend, taper, twist, or user-defined functions as in [26, 27].

3.6.3 CSG Operations

One of the important advantages of *BlobTree* is the ability to do CSG operations and local and global space warps, particularly based on Barr deformations. We incorporate CSG operations and warps into our local surface refinement so as to build CSG implicit objects.

Figure 6 shows a sequence of local refinements and global deformations. Given an initial plane, two local horns are first introduced. Then spikes are added as local details for each horn. In Figure 6b, two horns are globally bent forward. Then a spike is added to each horn and a long spike is added on the plane surface (c,d). The plane is then deformed with a global bend along the X axis and Y axis in the subsequent refinement (e,f). Finally, the central spike is deformed locally with an inward bend.

4 Implementation

We have chosen the *BlobTree* to implement our proposed method because it provides hierarchical composition of surfaces, CSG and warping operations. Each surface in the hierarchy is represented by a *BlobTree* . During local refinement and global deformation, the new *BlobTree* is automatically generated for visualization. The four refinement operations are converted to the corresponding *BlobTree* node respectively. For interactive modeling, a polygon mesh is used to show the object and direct implicit surface interaction is applied to obtain the precise interaction point on the surface.

5 Results and Discussion

The method presented has been used in modeling sea anemones in [16] and animating the escape behavior of sea anemone from starfish. It provided flexibility for modeling layered soft objects. In this section, we demonstrate the applications, modeling and animation, of the proposed method. We first describe an interactive building model and then show how to model a complex soft object, sea anemone. The animation of the shape deformation of the sea anemone can be easily achieved using the hierarchical representation of the surfaces. By simply doing local refinement or global deformation, we can simulate the deformation of the object.

5.1 Interactive Dragon Head Modeling

Figure 6 shows an interactive session where a model is under construction using hierarchical refinement. Starting with a simple root layer surface, an implicit plane, a sequence of surface details are gradually added. The method described in section 3.3 is used for surface interaction. First, two symmetrical horns with tapered line are added as the second layer. Then with interactively specified local constraints, two small spikes are introduced to each horn as the third layer 6a. In 6b, we apply a global deformation on two horns, bending both toward the $+Z$ axis. Notice that the two spikes on each of the horn are properly rotated and oriented as well. In 6c and 6d, we add another small spike on each of the horn and a long spike on the root surface. Then the root layer is globally deformed by bending the plane along the X axis 6e and Z axis 6f. As shown in b,e,f, local details are automatically adjusted to maintain the integrity of the whole surface after global deformation. In addition, local refinement allows us to add details at any layer during the modeling process.

5.2 Modeling Sea Anemones

The Sea anemone or the flower animal as they are often called, have a single body cavity that serves as a stomach, intestine and circulatory system. They fasten themselves to something firm with their base and have one body opening (the mouth), through which everything passes in or out. The mouth is surrounded by finger-like tentacles, studded with nematocysts (stinging cells). Nematocysts are active in capturing food and transferring it to the mouth for defense.

There are five main components of the sea anemone. The column which is cylindrical and not divided into regions. The base which is adherent, slightly irregular and much wider than the column is used to attach the anemone to the substrate. The upper disk is circular and transparent with orange patches scattered on it. There is a large central area which is free of tentacles, where there is a slit for the mouth. The surface of the disk is often irregular. The tentacles which surround the mouth may be up to 1.5 cm long, conical, and fully covered when retracted. The number of tentacles are usually 72 and can be 64, although individuals with as many as 86 can be found. They are generally arranged in four or five cycles with those of the inner cycles being slightly longer than those of the outer ones. The six tentacles of the first or inner cycle are usually held pointing inward over the disk, whereas those of the outer two or three cycles bend down over the margin. The color of the tentacles are white or transparent with two orange rings encircling them and a small white spot at the base of each.

Using hierarchical implicit surface refinement, the anemone is modeled in 4 layers. The root hierarchy represents the bottom disk, which is modeled using a torus. The column composes the second hierarchy and the upper disk is at the third layer. All tentacles are modeled at the same layer on top of the upper disk. The first two layers are characterized as a blend hierarchy, the third controlled blending, and all tentacles are created at the fourth layer built as blending hierarchies. The controlled blending third layer is used to avoid unnecessary blending among all tentacles and the body. The anemone mouth is modeled using a CSG hierarchy, which is also a local surface of the upper disk. Figure 7 shows the hierarchy of surfaces used for modeling the anemone.

Figure 8 shows an anemone, the *Stomphia Coccinea*. Two tori are used to model the bottom and upper disks, a cylinder for the column. Tapered and bent cylinders are used for modeling tentacles. The tentacles are initially placed on the upper disk plane and then shifted to the implicit surface of the main body moving from the outer cycle to inner cycles. Various noise functions are applied to control the tentacles shape and orientation. A collision-based model is used to accurately place the tentacles on the implicit surface following the spiral phyllotaxis pattern as mentioned in [16].

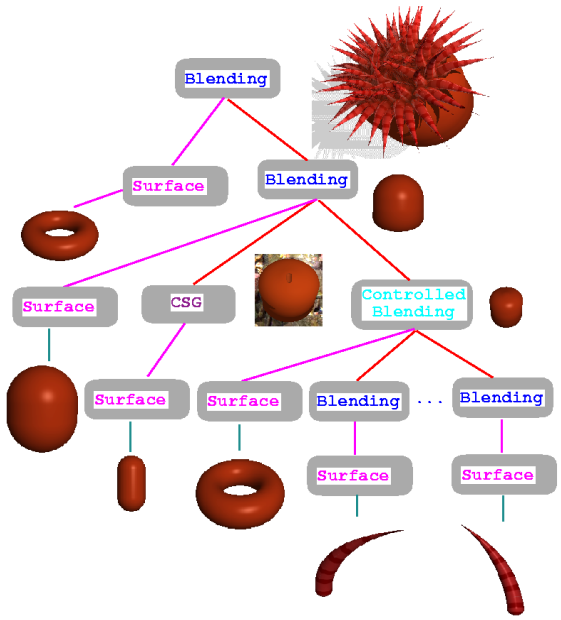

Figure 7: Hierarchy of surfaces modeling sea anemone

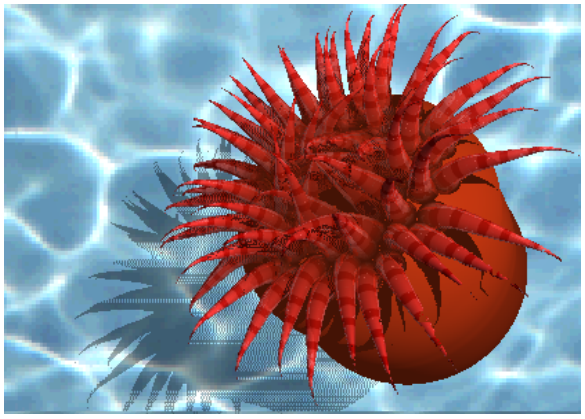

Figure 8: An anemone, *Stomphia Coccinea*, model

5.3 Animating the Sea Anemone with Local Refinement and Global Deformation

The hierarchical representation of implicit surfaces makes animation of the surface much easier. Animation can be created at any level or surface. Local refinement creates a sequence of new objects with different level of details, while global deformation changes the overall object shape at any level. This approach was used to build an animation of the escape response of the sea anemone, *Stomphia Coccinea* from the starfish, *Dermasterias Imbricata*. For instance, by simply applying globally bending and rotating deformations on each layer, we can simulate the

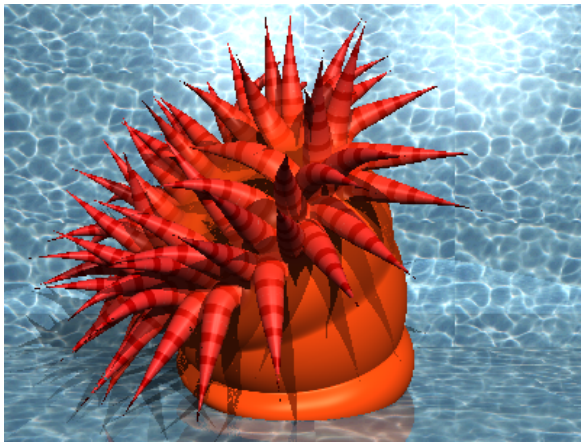

Figure 9: An anemone, *Stomphia Coccinea*, model

movement of the anemone. Figure 9 shows an animation frame with the second layer bent to the left.

6 Conclusion

We have presented our work on hierarchical implicit surface refinement. Using a hierarchical representation of implicit surfaces, implicit object is modeled as a layered surfaces. The hierarchy provides both layered local refinement and global deformation. Local refinement allows the introduction of higher-level detailed surfaces. Global deformation changes the overall shape of the surface while maintaining the integrity of surface details. Due to the innate blending and constraint properties of implicit surfaces, local constraints and hierarchical level of surfaces can be used to finely control the locality of the refinement.

Although the *BlobTree* has been used to represent the surfaces, this method is not limited to the *BlobTree* . It can be applied to other implicit surface representations as well. As mentioned in section 3.4, a more general surface splitting scheme is expected to find an optimal group of primitives approximating local details. To improve the interactivity of the local refinement, we are currently working on a faster implicit surface visualization method. [14], [10], [24] have presented methods for interactive implicit modeling. For local refinement, the incremental techniques [10] seems to be a promising method because we are basically restricting our refinement within local areas. We are also considering the inclusion of the *Blobby* constraints [15] to the hierarchy. With each *Blobby* constraint, we can model the implicit surface with

some magnetic effect. In this case, instead of blending the *Blobby* constraints with other primitives, its field contribution will only be used as a displacement to the surface point of the model. This will provide more flexibility for local refinement and global deformation as shown in [15].

7 Acknowledgements

The authors would like to thank the many students, past and present who have contributed towards the implicit modeling research project at the University of Calgary. We would also like to thank the department of computer science and the MACI project for the use of a large cluster of workstations for high performance computing. This work is partially sponsored by the Natural Sciences and Engineering Research Council.

References

[1] James Blinn. A Generalization of Algebraic Surface Drawing. *ACM Transactions on Graphics*, 1(3):235–256, 1982.

[2] Jules Bloomenthal. *Modelling Natural Forms*. PhD thesis, University of Calgary, 1995.

[3] Jules Bloomenthal. *Introduction to Implicit Surfaces*. Morgan Kaufmann, ISBN 1-55860-233-X, 1997. Edited by Jules Bloomenthal With Chandrajit Bajaj, Jim Blinn, Marie-Paule Cani-Gascuel, Alyn Rockwood, Brian Wyvill, and Geoff Wyvill.

[4] Jules Bloomenthal and Brian Wyvill. Interactive Techniques for Implicit Modeling. *Computer Graphics*, 24(2):109–116, 1990.

[5] Marie-Paule Cani-Gascuel. Layered Models with Implicit Surfaces. *Graphics Interface '98*, pages 201–208, June 1998. ISBN 0-9695338-6-1.

[6] M. Desbrun, N. Tsingos, and M.P. Gascuel. Adaptive Sampling of Implicit Surfaces for Interactive Modeling and Animation. *Computer Graphics Forum 96)*, December 1996.

[7] David R. Forsey. *Motion Control and Surface Modeling of Articulated Figures in Computer Animation*. PhD thesis, University of Waterloo, 1990.

[8] David R. Forsey and Richard H.Bartels. Hierarchical B-spline refinement. *Computer Graphics (Proc. SIGGRAPH 88)*, 22(4):205–212, August 1988.

[9] Callum Galbraith, Przemek Prusinkiewicz, and Brian Wyvill. Modeling murex cabritii seashell with a structured implicit surface modeler. *Proc. CG International 2000*, pages 55–64, 2000.

[10] E. Galin and S. Akkouche. Incremental Techniques for Implicit Surface Modeling. In *Proc. Computer Graphics International '98*. IEEE Computer Society, June 1998.

[11] Marie-Paule Gascuel. An Implicit Formulation for Precise Contact Modeling Between Flexible Solids. *Computer Graphics (Proc. SIGGRAPH 93)*, pages 313–320, August 1993.

[12] B. Guo. Shape Control in Implicit Modeling. In *Proceedings of Graphics Interface '91*, pages 230–235, 1991.

[13] Andrew Guy and Brian Wyvil. Controlled blending for implicit surfaces. In *Implicit Surfaces '95*, April 1995.

[14] J.C. Hart, A. Durr, and D. Harsh. Critical Points of Polynomial Metaballs. *ACM Computer Graphics (Proceedings Siggraph 1997)*, 31:279–286, 1997.

[15] X. Jin, Y.F. Li, and Q. Peng. General Constrained Deformations Based on Generalized Metaballs. In *Proceeding of Pacific Graphics '98*, pages 115–124. IEEE Computer Society, October 1998.

[16] Xikun Liang, Mai Ali Nur, and Brian Wyvill. Modeling the Structure of the Sea Anemone, Stomphia Coccinea and the Sea Star, Dermasterias Imbricata Using Implicit Surfaces. Technical report, University of Calgary, Dept. of Computer Science, 2000.

[17] S. Muraki. Volumetric shape description of range data using "blobby model". *Computer Graphics (Proc. SIGGRAPH 91)*, 24(4):227–235, July 1991.

[18] S. Muraki. Volume Data and Wavelet Transformations. *IEEE CG&A*, 13(4):50–60, 1993.

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

[20] T.W. Sederberg. Piecewise Algebraic Surface Patches. *Computer Aided Geometric Design*, 2(1-3):53–60, 1985.

[21] B.T. Stander and J.C. Hart. Gauranteeing the topology of an implicit surface polygonization. *SIGGRAPH'97*, 3:69–76, June 1998.

[22] R. Szeliski and D. Tonnensen. Surface Modeling with Oriented Particle Systems. *Computer Graphics (Proc. SIGGRAPH 92)*, 23(3):185–194, July 1992.

[23] Mark Tigges and Brian Wyvill. Texture Mapping the BlobTree. *Implicit Surfaces '98*, 3:123–130, June 1998.

[24] Kees van Overveld and Brian Wyvill. Potentials, Polygons and Penguins. An efficient adaptive algorithm for triangulating an equi-potential surface . In *Proc. 5th Annual Western Computer Graphics Symposium (SKIGRAPH 93)*, pages 31–62, 1993.

[25] Andrew Witkin and Paul Heckbert. Using particles to sample and control implicit surfaces. *Computer Graphics (Proc. SIGGRAPH 94)*, 28:269–277, July 1994.

[26] Brian Wyvill, Eric Galin, and Andrew Guy. Extending The CSG Tree. Warping, Blending and Boolean Operations in an Implicit Surface Modeling System. *Computer Graphics Forum*, 18(2):149–158, June 1999.

[27] Brian Wyvill and Kees van Overveld. Warping as a modelling tool for csg/implicit models. In *Shape Modelling Conference, University of Aizu, Japan*, pages 205–214. IEEE Society Computer Press ISBN 0-8186-7867-4, March 1997. inivited.

[28] Brian Wyvill and Geoff Wyvill. Better blending of implicit objects at different scales. *ACM Siggraph 2000 presentation*, 2000.

[29] Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Data Structure for Soft Objects. *The Visual Computer*, 2(4):227–234, February 1986.

[30] R. Zonenschein, J. Gomes, L. Velho, and L. H. Figueiredo. Texturing implicit surfaces with particle systems. In *Technical Sketch in Visual Proceedings SIGGRAPH 1997*, page 172, August 1997.