Reverse Loop Subdivision

Faramarz Samavati Hamid-Reza Pakdel Colin Smith Przemyslaw Prusinkiewicz
University of Calgary

Abstract

Reverse subdivision aims at constructing a coarse representation of an object given by a fine polygon mesh. In this paper, we derive a mask for reverse Loop subdivision. In contrast to the mask proposed before by Samavati and Bartels, the new mask applies to both regular and extraordinary vertices. The mask is parameterized, and thus can also be used to reverse variants of Loop subdivision, such as those proposed by Warren and Litke.

We apply this mask not only to mesh geometry, but also to texture coordinates. This reverses the texture-mapping process described by DeRose, Kass and Truong, in which a texture originally defined for a coarse mesh was carried to the finer meshes obtained by subdivision. By combining forward and reverse subdivision, the proposed technique yields a multiresolution representation of textured subdivision surfaces. We illustrate its use with a set of examples.

Keywords: subdivision, polygon mesh, texture mapping, multiresolution

1. Introduction

Each application of a subdivision rule applied to a surface generates a refined surface of increased detail. A sequence of subdivisions then produces a hierarchy of surfaces with increasing resolution. Such a hierarchy allows the selection of a representation of a surface at a particular level of detail, which is useful in many applications in both *modeling* and *rendering*.

While subdivision rules were initially conceived for refinements in geometric space, they are also applicable to refinements in texture space. This allows texture coordinates of a coarse mesh to be refined in the same process as the geometry [4, 21, 16] (Figure 1). Consequently, we only need a texture map for the given coarse mesh. The texture coordinates of the subsequent finer meshes are carried from the coarse mesh to the refined ones during subdivision.

Here we consider the inverse problem: *"How can texture coordinates be carried from a fine mesh over to a coarse one?"* We describe the possibility of recovering coarse meshes, with their texture coordinates, from a fine mesh using reverse subdivision. This new technique together with the technique of DeRose, Kass and Truong [4] can generate a hierarchy of meshes and texture coordinates at different levels of detail, so that all meshes and their texture coordinates can be produced from a specific mesh, either coarse or fine, using forward or reverse subdivision (Figure 9).

A hierarchy of levels of detail is useful in various applications, such as view dependent rendering and progressive transmission. For example, a fine mesh can be used when looking at an object from a close distance and a coarse mesh when looking from afar. In other words, with the multiresolution representation, only one approximation mesh of a surface together with its texture coordinates is needed, and meshes at other resolutions, both coarser and finer, can be generated.

Our framework for a multiresolution hierarchy (MR) is an extension of subdivision and is consistent with the hierarchies presented in [3, 13, 17, 22]. The progressive mesh (PM) [8, 20, 14] is another framework for constructing a level-of-detail hierarchy of an object, using vertex-split and collapse operations. PM can be applied to a mesh with an arbitrary topology, while MR can be applied directly only to those meshes that satisfy connectivity conditions required by the subdivision rule being used. In spite of this restriction, subdivision methods are widely used in various graphics applications. In addition, several techniques have been proposed for *remeshing* [7, 9, 10], which replaces meshes unsuitable for a given subdivision with suitable approximations, broadening the applicability of MR.

A mask for decreasing the resolution of a mesh is an important element of a multiresolution surface implementation. It reverses the process of refining, or increasing, the resolution of a mesh. While subdivision masks are usually given in local terms, the existence of local formulas for the corresponding reverse subdivision is less evident. For example, Lounsbery et al. [13] describe a technique for reverse subdivision using a global process.

The local reverse subdivision masks for Butterfly and Loop subdivision applicable to regular vertices, of valence six, were determined by Samavati and Bartels in [19], and for Doo-Sabin subdivision in [18]. In this paper, we present

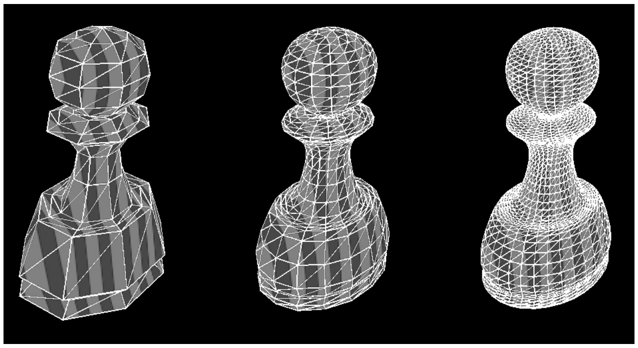

Figure 1. *Carrying texture from the coarse mesh to fine mesh through Loop subdivision scheme.*

a reverse mask for Loop subdivision that works for extraordinary vertices as well. We focus on Loop subdivision because it has several desirable properties. The limit surface is smooth, there is a convenient local formula for extraordinary vertices, and the subdivision works on triangular meshes. Loop subdivision has been widely used in computer graphics [24]; therefore, a reverse mask of Loop subdivision is of practical use. There are variations of Loop subdivision such as those proposed by Warren [23] and Litke [11], which we consider as well. Thus, we construct a parametric reverse mask that works for all Loop-style subdivisions.

Most vertices found in triangular polygon meshes used in practice are regular vertices. However, there is usually a small number of extraordinary vertices present as well. Therefore, determining a local formula for a reverse process that considers both regular and irregular valences is important. The main contributions of this work are the construction of such a mask and the concept of carrying texture coordinates from a fine mesh to a coarse mesh.

2. Background

Subdivision is a repetitive refinement process that gradually converts a given coarse mesh to smoother meshes. An arbitrary mesh M can be denoted by the pair (F, V), where F is the set of faces, and V the set of its vertices. Each element $\nu \in V$ has spatial coordinates, (x, y, z), and each element $f \in F$ is a list vertices, $\nu_i \in V$, that form the faces of the mesh.

The input for a particular subdivision method, of which Catmull-Clark[2], Doo-Sabin[5], Butterfly[6] and Loop[12] schemes are some important cases, is a control mesh, $M^0 = (F^0, V^0)$. In each step of subdivision, the mesh $M^k = (F^k, V^k)$ is used to generate a refined mesh $M^{k+1} = (F^{k+1}, V^{k+1})$. This conversion is using affine operations on V^k, combined with a mapping from the faces of F^k to those of F^{k+1}. The affine operations are smoothing filters, usually described by masks or matrices. Consequently, by successive application of a subdivision method, a sequence $M^0, M^1, M^2, \ldots$ of meshes is obtained that usually converges to a smooth surface.

2.1. Loop Subdivision

Loop subdivision is an extension of triangular B-spline subdivision to arbitrarily triangulated surfaces. In each step of this subdivision, each face in F^k is replaced by four new triangles that become the faces of F^{k+1} (Figure 2).

The set of new vertices V^{k+1} includes two types of vertices. Some vertices in V^{k+1} correspond to vertices in V^k, and are called vertex-vertices or even vertices (e.g. vertex ν^{k+1} in Figure 2). The other vertices in V^{k+1} correspond to edges in V^k, and are called edge-vertices or odd vertices (e.g. vertex ν_1^{k+1} in Figure 2).

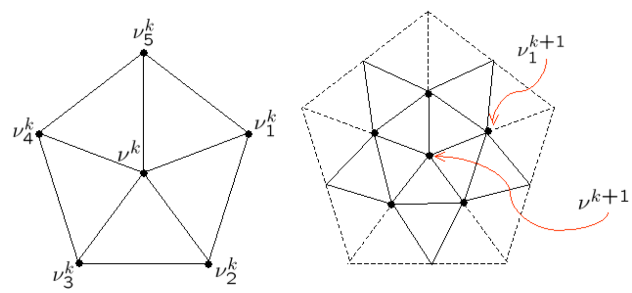

Figure 2. *A segment of a polygon mesh around some vertex ν^k before and after subdivision*

Let $\nu_1^k, \nu_2^k, \ldots, \nu_n^k$ be the set of neighbors of ν^k in M^k. In addition, let ν^{k+1} be the corresponding vertex-vertex of ν^k, and $\nu_1^{k+1}, \nu_2^{k+1}, \ldots, \nu_n^{k+1}$ be the corresponding edge-vertices. The position of ν^{k+1} is then obtained by the vertex-vertex mask for Loop subdivision [12]:

$$\nu^{k+1} = \beta \nu^k + \alpha \sum_{j=1}^{n} \nu_i^k, \tag{1}$$

where

$$\alpha = \frac{1}{n}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\left(\frac{2\pi}{n}\right)\right)^2\right) \tag{2}$$

and

$$\beta = 1 - n\alpha.$$

The weight α is a function of n, and has been selected such that the limit surface is smooth.

The edge-vertex mask is:

$$\nu_j^{k+1} = \frac{3}{8}\nu^k + \frac{3}{8}\nu_j^k + \frac{1}{8}\nu_{j+1}^k + \frac{1}{8}\nu_{j-1}, \quad (3)$$

$$j = 1, 2, \ldots, n.$$

where operations on indices j are carried out modulo n.

For triangular meshes, a *regular vertex*, or *ordinary* vertex, has the valance of six (i.e., $n = 6$). Otherwise, the vertex is called *extraordinary*. In the regular case, $\alpha = \frac{1}{16}$, and $\beta = \frac{5}{8}$; these are obtainable from the triangular B-spline surface. But masks 2 and 3 are more general and can be applied to any type of vertex, either ordinary and extraordinary. The local nature of formula 2 is one of the advantages of Loop subdivision.

2.2. Boundaries

For vertices on a mesh boundary, we need to use masks that are different from those for interior vertices. It is important that subdivision at any point on the boundary be independent of any point in the interior of the mesh, because this makes it possible to join two surfaces along a boundary curve [24]. Therefore, the masks for cubic B-spline curves are used as the boundary masks for Loop subdivision:

$$\begin{aligned} \nu^{k+1} &= \tfrac{1}{8}\nu_1^k + \tfrac{3}{4}\nu^k + \tfrac{1}{8}\nu_2^k \\ \nu_1^{k+1} &= \tfrac{1}{2}\nu^k + \tfrac{1}{2}\nu_1^k \\ \nu_2^{k+1} &= \tfrac{1}{2}\nu^k + \tfrac{1}{2}\nu_2^k \end{aligned} \quad (4)$$

where ν_1^k and ν_2^k are the direct neighbors of ν^k on the boundary (Figure 3).

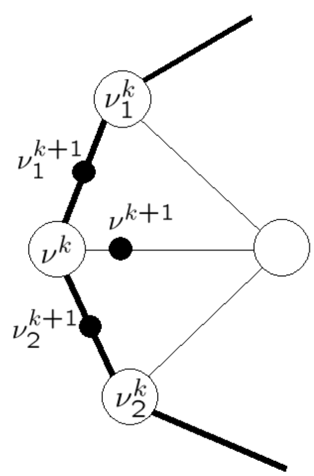

Figure 3. *A boundary situation for Loop subdivision*

2.3. Subdivision Schemes Related to Loop's

Several variants of Loop subdivision exist, such as the triangle averaging scheme of Warren and Weimer [23], and the quasi-interpolation scheme of Litke, Levin and Schröder [11]. The triangle averaging scheme can produce a smooth surface, but not necessarily at extraordinary vertices. The α and β values of this scheme are:

$$\begin{aligned} \alpha &= \tfrac{3}{8n}, \\ \beta &= \tfrac{5}{8}. \end{aligned} \quad (5)$$

In the quasi-interpolation scheme, the mask values are:

$$\begin{aligned} \alpha &= -\tfrac{1}{2n}, \\ \beta &= \tfrac{3}{2}. \end{aligned} \quad (6)$$

Examples of these subdivision schemes are given in Figure 4.

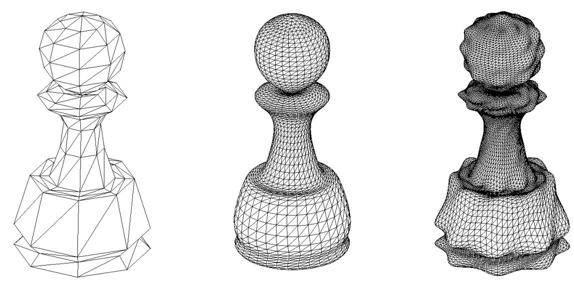

Figure 4. *From left to right: the control mesh and the results of triangle averaging and quasi-interpolation subdivision schemes*

2.4. Texture Mapping

DeRose, Kass and Truong [4] present the concept of carrying texture coordinates from the coarse mesh to the finer ones by using the same subdivision rules as those used for geometry. Lets assume that texture coordinates of the control mesh M^0 are given. Each texture coordinate is a pair, (s_j, t_j), assigned to the vertex $\nu_j \in V$. This pair presents a point in the texture space whose value defines the color of ν_j. The coordinates (x_j, y_j, z_j) of vertex ν_j are extended with coordinates (s_j, t_j) resulting in a point in five-dimensional space. Now, it is sufficient to apply the subdivision mask for elements of $\mathbf{R}^5$.

We extend this concept by adding the possibility of carrying texture coordinates from a fine mesh to a coarse one. It relates to the reversal of subdivision and construction of a multiresolution representation.

2.5. Reverse Subdivision and Multiresolution Surfaces

Subdivision methods produce a hierarchy

$$M^0, M^1, M^2, \ldots, M^k, \ldots$$

where $M^k = (V^k, F^k)$. Each vertex $\nu \in V$ has a coordinate in $\mathbf{R}^5$, (x, y, z, s, t). Suppose M^k is a good approximation of the limit surface in a particular view. Now, if the object position is changed to a new position that is closer to the view point, it will be necessary to construct a finer mesh M^{k+i}, where $i > 0$. This can be achieved by applying i iterations of the subdivision rule to M^k. In other words, knowing M^k is sufficient for calculating for all finer approximations. On the other hand, if the object position is moved to a more distant position, we may need an approximation that is coarser than M^k. We thus face the problem: *"How can coarser meshes $M^{k-\ell}$ be obtained from M^k?"* This question is interesting in two ways. Firstly, for $1 \leq \ell \leq k$, procedural reversal makes it unnecessary to store meshes $M^0, \ldots, M^{k-1}$. Secondly, for $\ell > k$, the reversal may provide us with coarser meshes than M^0, which might be useful if M^0 has a fairly high level of detail to begin with.

The reverse mask together with the subdivision mask make it possible to construct a multiresolution hierarchy of objects that is suitable for applications such as view-dependent rendering, flexible editing and progressive transmission. A subdivision mask increases the resolution of an object, and a reverse mask decreases it. In section 3, we construct the reverse mask for Loop subdivision.

3. Reverse Mask for Loop Subdivision

3.1 Derivation of the Mask for Ordinary and Extraordinary Vertices

For the reverse subdivision process, it is necessary to construct a mask to map V^{k+1} to V^k. Assume a general situation for Loop subdivision around extraordinary vertex, as shown in Figure 5. In this Figure, we know vertices $\nu^{k+1}, \nu_1^{k+1}, \nu_2^{k+1}, \ldots, \nu_n^{k+1}$, and we want to find vertex ν^k using a mask that meets the following conditions:

1. The operation of the new mask must be affine.
2. The weights applied to the neighbors of ν^{k+1} in the mask must be equal to each other, as in the mask in Equation 1.
3. The new mask must be a reverse of the Loop mask. The action of subdivision mask of Equation 1 and 2 on ν^{k+1} and its neighbors must exactly reconstruct ν^k.

Condition (2) above yields the diagram of the reverse mask, shown in Figure 6. In this diagram μ is the weight of ν^{k+1} and η is the weight of the each of its neighbors. Note that the weights are equal to each other. We now determine the

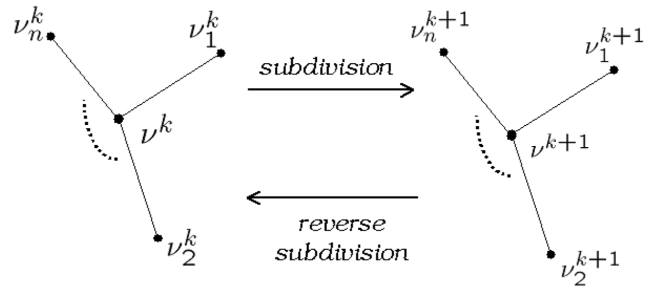

Figure 5. *General situation for an extraordinary vertex*

weights μ and η so that conditions (1) and (3) are also satisfied.

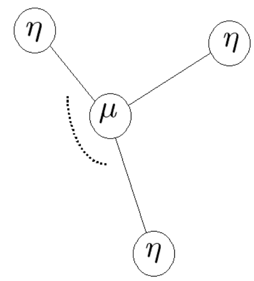

Figure 6. *Reverse mask*

For condition (3) (reversal), the following equation must hold:

$$\mu\,\nu^{k+1} + \eta \sum_{j=1}^{n} \nu_j^{k+1} = \nu^k.$$

From Equation 1, we obtain:

$$\mu(\beta\nu^k) + \mu\left(\alpha \sum_{j=1}^{n} \nu_j^k\right) + \eta\left(\frac{3}{8}n\nu^k\right) + \eta\left(\frac{5}{8}\sum_{j=1}^{n} \nu_j^k\right) = \nu^k,$$

or, equivalently,

$$\left(\mu\beta + \frac{3}{8}n\eta\right)\nu^k + \left(\mu\alpha + \frac{5}{8}\eta\right)\sum_{j=1}^{n} \nu_j^k = \nu^k.$$

This equation must hold true for any vertices ν^k and ν_j^k if

$$\begin{cases} \mu\beta + \frac{3}{8}n\eta = 1 \\ \mu\alpha + \frac{5}{8}\eta = 0 \end{cases}. \qquad (7)$$

In this system, α and β are parameters of the Loop subdivision mask in Equation 2 and thus satisfy the equation:

$$\beta = 1 - n\alpha.$$

We use this equation to eliminate α in system 7. By solving it with respect to μ and η, we obtain:

$$\mu = \frac{5}{8\beta - 3}, \quad (8)$$

$$\eta = \frac{\beta - 1}{n\left(\beta - \frac{3}{8}\right)}. \quad (9)$$

Equation 9 is a parametric formula for the reverse mask and can be applied to both regular and extraordinary vertices. For example, in the case of a regular vertex, with $n = 6$, $\alpha = \frac{1}{16}$ and $\beta = \frac{5}{8}$, Equation 9 yields $\mu = \frac{5}{2}$ and $\eta = -\frac{1}{4}$, as shown in Figure 7. This result matches the **A** mask of width one, reported in [19]. However, unlike that mask, the diagram in Figure 6 together with formula 9 can also be used for extraordinary cases.

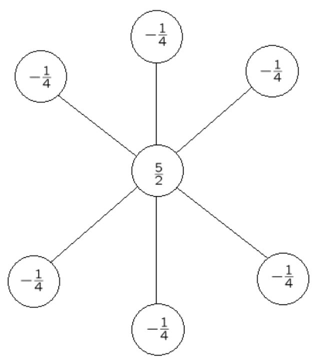

Figure 7. *Reverse mask for a regular vertex*

For example, let us consider a vertex with a valence $n = 3$. From Equation 2, we have:

$$\alpha = \frac{3}{16} \text{ and } \beta = \frac{7}{16}.$$

By substituting this β value into Equation 9, we obtain:

$$\mu = 10 \text{ and } \eta = -3,$$

as illustrated in Figure 8.

Condition (1), the affinity of the reverse mask, is automatically satisfied, because

$$\mu + n\eta = \frac{5}{8\eta - 3} + \frac{\beta - 1}{n(\beta - \frac{3}{8})} = 1.$$

Thus, the sum of the weights of all argument points is equal to one.

3.2. Reverse Masks of Subdivisions Related to Loop's

Due to its parameterized character, Equation 9 can be applied directly to the triangle averaging scheme (5) and the quasi-interpolation scheme (6). For the reverse mask of the triangle averaging scheme, we obtain:

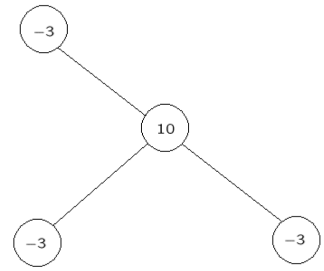

Figure 8. *Reverse mask for an extraordinary vertex with valence $n = 3$*

$$\mu = \frac{5}{2} \text{ and } \eta = -\frac{3}{2n}, \quad (10)$$

and for the reverse mask of the quasi-interpolation scheme, we have:

$$\mu = \frac{5}{9} \text{ and } \eta = \frac{4}{9n}. \quad (11)$$

3.3. Reverse Mask for the Boundary Vertices

We have previously assumed the cubic B-spline mask for the forward subdivision of boundary vertices. Consequently, we now need to find a reverse mask for the cubic B-spline subdivision. Bartels and Samavati [1] present several masks for cubic B-spline subdivision. The simplest one is given by the equation:

$$\nu^k = -\frac{1}{2}\nu_1^{k+1} + 2\nu^{k+1} - \frac{1}{2}\nu_2^{k+1}. \quad (12)$$

We use here the same notation as in section 2.2.

4. Results

Figures 9, 10 and 11 show the results of Loop subdivision of closed textured surfaces, followed by reverse subdivision. Figure 12 illustrates the same operations in the case of an object with boundary. As expected, in all cases the reverse subdivision of a subdivided surface restores the original mesh. Used together, forward and reverse subdivision make it possible to both increase and decrease the level of detail, as needed in a multiresolution object representation (Section 2.5).

Figure 13 presents the results of Loop subdivision followed by a modification of the subdivided surface and two instances of reverse subdivision, obtained with the reverse Loop mask (Equation 9) and reverse quasi-interpolation scheme (Equation 11). Although reverse Loop subdivision exactly restores meshes that have been obtained by forward Loop subdivision, in the case of a modified mesh the quasi-interpolation scheme produces a more plausible result. We attribute this to the fact that all point weights (values μ and

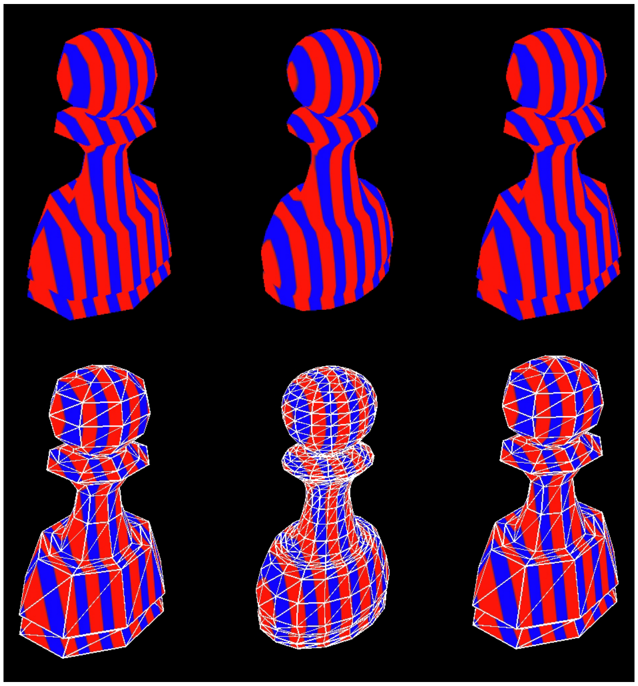

Figure 9. *Top row: textured control polygon, the result of one step of Loop subdivision, and the result of the subsequent reverse Loop subdivision. Bottom row: The same meshes shown with polygon edges. The reverse subdivision of a subdivided surface restores the original mesh.*

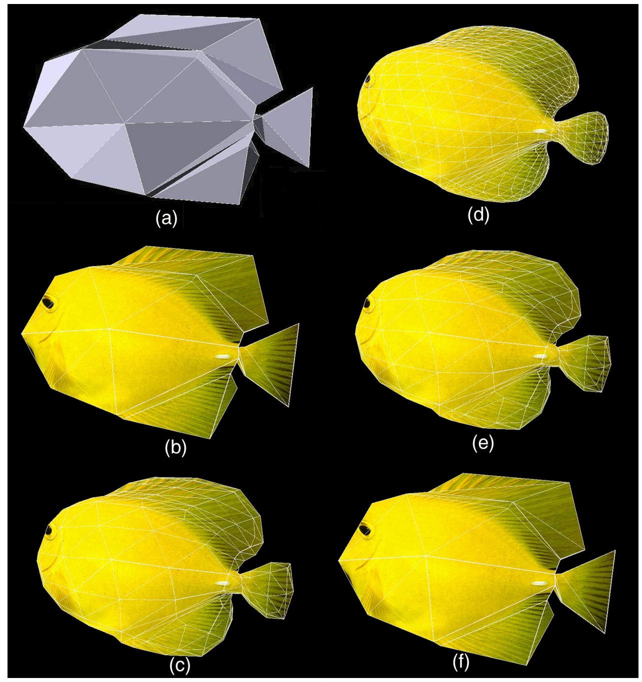

Figure 10. *Subdivision and reverse subdivision of a fish model. (a) The control mesh. (b) The textured control mesh. (c,d) The mesh after one and two steps of Loop subdivision. (e,f) Mesh (d) after one and two steps of reverse Loop subdivision.*

η in Figure 6 are positive in the quasi-interpolation case, which guarantees that the coarse mesh will be contained in the convex hull of the fine mesh.

5. Conclusion

We have constructed a parameterized reverse mask for Loop subdivision. The parameterization makes it possible to reverse related subdivision schemes, such as the triangle-averaging and quasi-interpolation schemes. The reverse subdivision applies not only to the model geometry, but also to its texture coordinates. Taken together, forward and reverse subdivision provide a multiresolution representation of subdivision surfaces, in which the level of detail can be procedurally increases or decreased.

One problem open for further research is the use of reverse subdivision masks with a wider support, not limited to direct neighborhoods of the affected points. Such masks may provide better approximations of objects that have not been obtained by subdivision, or have been modified after subdivision.

Another problem is the extension of the reverse Loop subdivision with detail information. In the current method, subdivision followed be reverse subdivision always preserves the original mesh. The inclusion of detail information would guarantee that reverse subdivision followed by forward subdivision would also restore the mesh.

5.1. Acknowledgments

We would like to thank Richard Bartels, Mario Costa-Sousa, Katayoon Etemad and Masoud Alipoor for insightful discussions. The partial support of this research by grants from the Natural Science and Engineering Research Council of Canada (NSERC) is also gratefully acknowledged.