EUROGRAPHICS '99 / P. Brunet and R. Scopigno
(Guest Editors)

Volume 18 (1999), Number 3

A Shrink Wrapping Approach to Remeshing Polygonal Surfaces

Leif P. Kobbelt   Jens Vorsatz   Ulf Labsik   Hans-Peter Seidel

Computer Graphics Group
Max-Planck-Institut für Informatik
Im Stadtwald, 66123 Saarbrücken, Germany

Abstract

Due to their simplicity and flexibility, polygonal meshes are about to become the standard representation for surface geometry in computer graphics applications. Some algorithms in the context of multiresolution representation and modeling can be performed much more efficiently and robustly if the underlying surface tesselations have the special subdivision connectivity. In this paper, we propose a new algorithm for converting a given unstructured triangle mesh into one having subdivision connectivity. The basic idea is to simulate the shrink wrapping process by adapting the deformable surface technique known from image processing. The resulting algorithm generates subdivision connectivity meshes whose base meshes only have a very small number of triangles. The iterative optimization process that distributes the mesh vertices over the given surface geometry guarantees low local distortion of the triangular faces. We show several examples and applications including the progressive transmission of subdivision surfaces.

1. Introduction

Unstructured triangle meshes are the most versatile surface representation in computer graphics since they enable the description of surfaces with arbitrary shape and topology without having to observe complicated compatibility conditions known from surface patching based on splines [7, 12]. The genuinely non-smooth characteristics of piecewise planar surfaces can be reduced by refining the tesselation until a prescribed angle tolerance between adjacent triangles is reached. The prohibitive complexity of highly detailed triangle meshes can be controlled by introducing a hierarchical level-of-detail decomposition based on the intermediate results of a mesh decimation algorithm [2, 26, 24, 11, 8, 17].

However, especially in the context of multiresolution representation and modeling, many algorithms require a specific structure of the meshes, namely *subdivision connectivity*. This special type of mesh connectivity is generated by iteratively applying a uniform subdivision operator to a coarse base mesh $\mathcal{S}_0$. The uniformity of the refinement operator implies that the resulting mesh is piecewise regular, i.e., each submesh which topologically emerges from one single triangle of the base mesh has the connectivity of a regular grid (cf. Fig. 1).

In this paper, we present a new method for converting a given mesh with arbitrary connectivity into one with subdivision connectivity. The benefits from this conversion are manifold. The primary advantage of subdivision connectivity meshes is the direct availability of multiresolution semantics since the different refinement levels provide the dyadic levels of detail which enable the generalization of wavelet techniques to parameteric surfaces [21, 25, 27, 35]. Moreover, the remeshing procedure is equivalent to constructing a global parameterization of the given surface over a polyhedral parameter domain. Hence, the result provides the basis for many computer graphics and engineering algorithms like texturing and milling path generation. In the result section we will present another interesting application in the context of progressive transmission of surface geometry.

The paper is organized as follows. After properly defining the remeshing problem in Section 2 and discussing some previous work in Section 3, we explain the basic principle of the algorithm in Section 4. We then identify the central problems with the straight forward approach in Section 5 and pro-

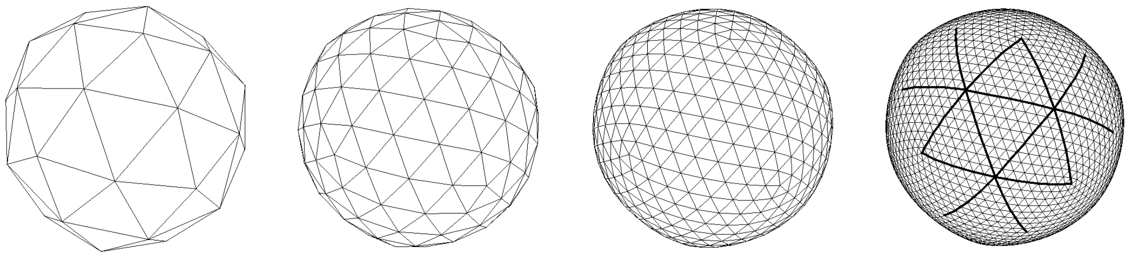

Figure 1: *Meshes with subdivision connectivity are generated by uniformly subdividing a coarse base mesh $\mathcal{S}_0$. On the refined meshes $\mathcal{S}_m$ we can easily identify regular submeshes (*subdivision patches*) which topologically correspond to one single triangle of the base mesh $\mathcal{S}_0$ (right).*

pose solutions which extend the basic scheme in two ways: In Section 6 a technique for finding an appropriate base mesh with a rather small number of triangles is described and in Section 7 we explain how to make the basic algorithm numerically more stable. Finally in Section 8 we illustrate the effectiveness of our algorithm by several example meshes.

2. Remeshing requirements

We start by fixing the notation. Let an arbitrary (manifold) triangle mesh $\mathcal{M}$ be given which represents the genus-zero surface of a geometric model. The task of remeshing the input data $\mathcal{M}$ means to find a sequence of meshes $\mathcal{S}_0, \ldots, \mathcal{S}_m$ such that each $\mathcal{S}_{i+1}$ emerges from $\mathcal{S}_i$ by the application of a uniform subdivision operator which performs a 1-to-4 split on every triangular face of $\mathcal{S}_i$ (cf. Fig. 1). Since the $\mathcal{S}_i$ should be differently detailed approximations of $\mathcal{M}$, the vertices $\mathbf{p} \in \mathcal{S}_i$ have to lie on the *continuous* geometry of $\mathcal{M}$ but they not necessarily have to coincide with $\mathcal{M}$'s vertices. We allow but do not require that the vertices of $\mathcal{S}_i$ are a subset of $\mathcal{S}_{i+1}$'s vertices (*interpolatory / non-interpolatory subdivision*).

In general, it would be enough to generate the mesh $\mathcal{S}_m$ since the coarser levels of detail $\mathcal{S}_i$ can be extracted by subsampling. Nevertheless, building the whole sequence $\mathcal{S}_0, \ldots, \mathcal{S}_m$ from coarse to fine often leads to more efficient multi-level algorithms.

The special connectivity of $\mathcal{S}_m$ gives rise to a partitioning into regular submeshes (*subdivision patches*) consisting of 4^m triangles. Each subdivision patch $\mathcal{T}_j \in \mathcal{S}_m$ corresponds to one triangle T_j of the base mesh $\mathcal{S}_0$ and can be parameterized naturally by assigning dyadic barycentric coordinates to the vertices of $\mathcal{T}_j$. Combining the local parameterizations of the subdivision patches yields a global parameterization for $\mathcal{S}_m$ over the polyhedral parameter domain $\mathcal{S}_0$.

The *quality* of a subdivision connectivity mesh is measured in two different aspects. First, we want the above parameterization which maps points from the base mesh $\mathcal{S}_0$ to the corresponding points on $\mathcal{S}_m$ to be close to isometric, i.e., the *local distortion* of the triangles should be small and evenly distributed over each patch. To achieve this, it is necessary to adapt the shape of the triangles in the base mesh $\mathcal{S}_0$ carefully to the shape of the corresponding surface patches in the given mesh $\mathcal{M}$.

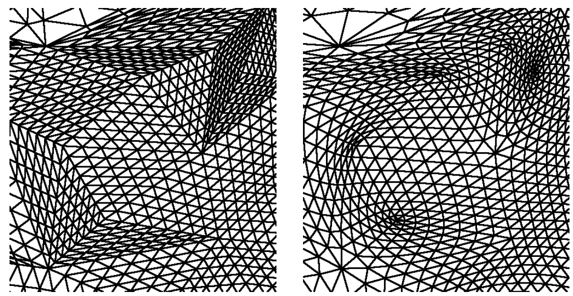

Figure 2: *Minimizing the distortion of the map from $\mathcal{S}_0$ to $\mathcal{S}_m$. On the left, the distortion is minimized within each base triangle while on the right the distortion control is extended across the base edges.*

As mentioned in [19], distortion control can be extended *across* the edges of the original base mesh (cf. Fig. 2). While this leads to an improved visual appearance of the mesh structure, it is shown in [16] that the mathematical quality in terms of parametric distortion is not necessarily improved.

The second quality requirement rates the base mesh $\mathcal{S}_0$ itself according to the usual quality criteria for triangle meshes, i.e., uniform size and aspect ratio of the base triangles.

While general purpose mesh generation and mesh decimation algorithms typically adapt the size of the triangular faces to the local curvature of the underlying surface, one has to follow a different concept in the context of multiresolution representations and modeling. In order to have a clear relation between the *size* of a geometric feature and the respective subdivision level $\mathcal{S}_i$ on which it is introduced during refinement (*space-frequency localization*), it is more reasonable (to try) to make all triangles from the same refinement level to have the same size. Adaption to the local curvature can then be achieved by selectively refining the mesh with a red-green triangulation strategy [31, 32], i.e. we locally switch to higher refinement levels instead of adapting the size of the triangles on the same level (cf. Fig. 3).

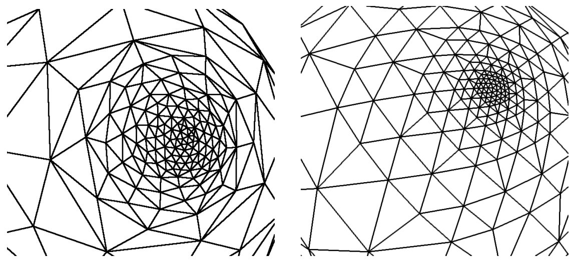

Figure 3: *Two different strategies for adapting the size of the triangles to some local refinement criterion. Left: the triangle size is continuously adapted without classifying the vertices according to their respective level of detail. Right: Adaption is achieved by locally switching the (discrete) refinement level. The surface gaps resulting from T-vertices at edges where triangles from different refinement levels meet, are fixed by red-green triangulation.*

Since the shape of the triangles in the base mesh determines the global parameterization with respect to which the multiresolution decomposition is defined, the notion of uniformity (inherent to all subdivision based multiresolution representations) is best emulated by equalizing the sizes of the base triangles. This is quite different from the general setting of the surface tesselation problem where the distribution of the vertices is usually driven by the local approximation error in order to minimize the total number of triangles while meeting a prescribed tolerance.

In order to satisfy both quality criteria (low local distortion and uniform triangle size), a remeshing algorithm has to start by finding a fairly regular base mesh $\mathcal{S}_0$ such that each subdivision patch $\mathcal{T}_j \in \mathcal{S}_m$ has approximately the same surface area. This guarantees equal vertex density on the refined mesh and hence a uniform sampling density of the original geometry. In terms of surface parameterization it means that the local stretching and contracting is bounded since the parameter triangles have similar areas as well. When uniformly subdividing the base mesh $\mathcal{S}_0$, the local distortion can be further minimized by applying a relaxation operator which tends to locally equalize the length of the edges and the angles between them (e.g. Laplace smoothing [28], Umbrella algorithm [18]).

3. Previous work

In [6] a remeshing algorithm is proposed which starts by placing the vertices of the base mesh $\mathcal{S}_0$ randomly on the input mesh $\mathcal{M}$. Their local density and the connectivity of the base mesh $\mathcal{S}_0$ is controlled by growing geodesic Voronoi-tiles around each vertex. This guarentees that the base mesh is optimal with respect to the global distribution since all triangles of the base mesh cover approximately the same area element of the given surface. The parameterization within each triangle of the base mesh is found by solving a discrete optimization problem in order to minimize the local distortion. The resulting parameterization is optimal for each base triangle but not smooth across the boundary between surface patches which belong to adjacent base triangles. The actual remeshing is done by uniformly sampling the obtained parameterization at the dyadic barycentric coordinates on each base triangle.

In [19] the base mesh is found by applying a mesh decimation algorithm to the original mesh. This provides more control on the generation of the base mesh since feature lines can be taken into consideration and local curvature estimates can be used to determine regions where more or less vertices are removed. An initial parameterization is constructed by incrementally projecting the removed vertices onto the remaining coarse mesh. This parameterization is, again, not globally smooth but only locally within each patch corresponding to one base triangle. The actual remeshing is not done by directly sampling the initial parameterization at the dyadic barycentric parameter values but an additional smoothing step based on a variant of Loop's subdivision scheme is used to shift the sampling sites within the parameter domain.

In both algorithms the surface sampling is done by solving the point location problem for the barycentric coordinates in the base triangles and then finding the corresponding point on the original mesh. In some sense both algorithms are more general than the shrink wrapping approach since they can process meshes with non-zero genus.

4. Shrink wrapping

The physical model behind our algorithm for remeshing arbitrary triangle meshes is the process of *shrink wrapping* where a plastic membrane is wrapped around an object and shrunk either by heating the material or by evacuating air from the space inbetween the membrane and the object's surface. At the end of this process, the plastic skin provides an exact imprint of the given geometry.

To simluate the shrink wrapping process, we approximate the plastic membrane by a triangle mesh $\mathcal{S}_m$ with subdivision connectivity. During the process each vertex is moved according to a force applied to it. The force is a combination of two components. One component is pulling each vertex in the direction of the given surface (*attracting force*) and the other component is pushing the vertices in order to minimize the local distortion energy within the membrane (*relaxing force*). The relaxing force supports the distribution of the vertices over the surface. Without this internal force local clustering of vertices and self-intersections of the mesh (*folding*) could not be prevented.

The proposed approach is very similar to the concept of *snakes* or *deformable surfaces* known in image processing and computer vision [3, 14, 23, 29, 30, 22]. These techniques are used to extract contours from two- and three-dimensional image data by optimizing the shape of a polygon or a triangle mesh. The optimization is controlled by an internal stabilization force (bending energy minimization) and an external force pulling the vertices towards the contour. The external force is usually derived from the gradient information in the underlying two- or three-dimensional scalar field.

We could easily adapt the deformable surface approach to our specific setting by voxelizing the given object [13, 33], i.e. by sampling its characteristic function on a three dimensional cartesian grid. However, this method is not appropriate here due to the insufficient spatial resolution. As the quality requirements for the remeshing procedure are rather high in terms of geometric tolerance, we cannot approximate the given geometry by an iso-surface contour of a piecewise trilinear scalar field.

We therefore suggest to replace the attracting force by a projection operator **P** which maps each vertex of the shrinking mesh $\mathcal{S}_m$ onto the closest point of the given geometry $\mathcal{M}$. This operator has a similar effect on the vertex positions like imposing an attracting force but the vertices are placed exactly on the target surface and not only nearby.

The projection is computed by shooting rays in normal direction from the mesh $\mathcal{S}_m$ and computing the intersections with $\mathcal{M}$. We can use a space partition technique known from ray-tracing [9] to considerably accelerate this step.

For the internal relaxing force we use the density weighted umbrella operator **U** [18] which iteratively minimizes the surface area of the mesh (membrane energy) by updating each vertex **p** according to

$$\mathbf{p} \quad \leftarrow \quad (1-\alpha)\,\mathbf{p} + \frac{\alpha}{\sum_j d_j} \sum_{j=0}^{n-1} d_j\,\mathbf{q}_j \qquad (1)$$

with n being the valence of the vertex **p** and $\mathbf{q}_0, \ldots, \mathbf{q}_{n-1}$ denoting its adjacent neighbors in $\mathcal{S}_m$. The density coefficients d_j are set to the average length of the edges emanating from $\mathbf{q}_j$ respectively. The effect of the density weights is that clustering is prevented more effectively and the vertex distribution is performed more aggressively. The damping factor is initially set to $\alpha = \frac{1}{2}$.

The algorithm that simulates the shrink wrapping is an iterative process where we alternate the **P** operator and the **U** operator, i.e., we alternate projection onto the target surface $\mathcal{M}$ and relaxing the mesh. By slowly fading out the relaxing force ($\alpha \to 0$) from one iteration to the next, we guarantee effective vertex distribution during the first steps and eventual convergence of the vertices to locations on $\mathcal{M}$ afterwards.

The iterative shrinking algorithm can be greatly accelerated by using a multi-level approach [10, 18] for solving the underlying optimization problem: Instead of immediately projecting and relaxing the mesh $\mathcal{S}_m$, we first apply the algorithm to an intermediate mesh $\mathcal{S}_i$ with lower resolution. Once the scheme converges (on the ith refinement level), we subdivide the mesh $\mathcal{S}_i \to \mathcal{S}_{i+1}$ and continue the $\mathbf{P}/\mathbf{U}$ iteration on $\mathcal{S}_{i+1}$. The advantage of the multi-level strategy is that we can find coarse approximations with meshes of moderate complexity (where the computational costs of **P** and **U** are low). When we eventually switch to higher refinement levels, the current starting mesh is already close to the final solution and only few additional iteration steps are necessary.

5. Problems

The simple mesh fitting scheme explained in the last section works very effectively on simple surfaces. However, it does not generate reasonable results on more complicated objects. One reason for the occurring mesh artifacts can be found by observing the behavior of real shrink wrapping material. In fact, most packages made by this technique suffer from severe wrinkles. Mathematically this effect can be explained by the fact that the object's surface and the plastic membrane have rather different surface metrics [4], i.e., their first fundamental forms differ significantly. Hence, when fitting the membrane, there are regions where the plastic has to be stretched extremely and regions where there is too much material and hence creases are generated from superfluous material.

On the shrunk subdivision connectivity mesh $\mathcal{S}_m$ we have the same difficulties with the different surface metrics. The strong stretching in some regions results in significantly larger triangles and in regions with relatively high vertex density, we can observe creases and ripples (cf. Fig. 4). As explained in Sect. 2 we have to adapt the base mesh $\mathcal{S}_0$ to the shape of $\mathcal{M}$ in order to avoid these artifacts since extreme stretching and folding (contraction) can only be avoided if all surface patches $\mathcal{T}_j$ corresponding to the triangles T_j of the base mesh $\mathcal{S}_0$ have approximately the same area.

Another severe problem with the projection/smoothing approach is the phenomenon that projection algorithms can fail or at least produce counter-intuitive results if the shape of the starting mesh and the shape of the target mesh are too different from each other. This is caused by the fact that in this case the normal vectors of the starting mesh might not intersect with the target mesh. In order to derive a robust algorithm for the remeshing, we have to construct good starting configurations which are sufficiently close to the original geometry $\mathcal{M}$ such that the projection operator becomes well-defined and stable.

In the next two sections, we present solutions for these problems which extend the basic shrink wrapping algorithm. The key idea is to find a parameterization of the given model

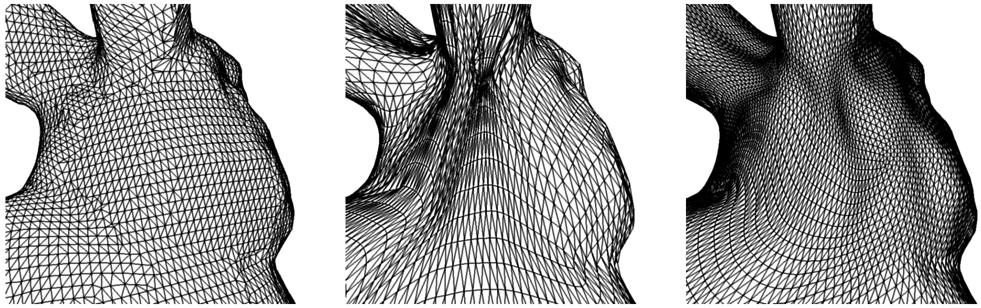

Figure 4: *Behind the ears of the Stanford bunny is a concave region where excess material causes wrinkles in the shrinking membrane. Adapting the base mesh $\mathcal{S}_0$ to the surface metric of the given mesh $\mathcal{M}$ removes this artifact. (From left to right: original mesh, non-adapted membrane, adapted membrane)*

over the unit sphere. Instead of using the original model, the necessary operations for the shrink wrapping can then be performed on the sphere which is easier since the projection operator **P** becomes trivial.

6. Constructing the base mesh

The process of "custom tayloring" the initial mesh $\mathcal{S}_0$ for a given geometry $\mathcal{M}$ requires to find a coarse triangulation which partitions the mesh $\mathcal{M}$ into triangular patches with equal surface area. This guarantees an approximately constant stretching factor for the map $\mathcal{S}_0 \to \mathcal{S}_m$.

While the techniques proposed in [6, 19] operate directly on the given triangle mesh $\mathcal{M}$, we solve this problem by first computing a parameterization for $\mathcal{M}$ over a *bounding sphere* (cf. Fig. 5). This will make it easier to control the number of triangles in the base mesh $\mathcal{S}_0$ which we want to keep as small as possible. A collection of algorithms how to compute a low distortion projection map from an arbitrary genus-zero object to a bounding sphere can be found in [15]. The basic idea of these algorithms is to let the object grow towards its convex hull and then project it directly onto the sphere. Without loss of generality, we can assume the bounding sphere to be the unit sphere.

In our implementation we use a variation of this principle. We first compute a voxelization of the given model. The gradient of the corresponding volumetric distance function is then used to guide the vertices of the given mesh towards a convex configuration. This process can be considered as some kind of "inverse" deformable surface technique.

A parameterization **F** of the mesh $\mathcal{M}$ over a bounding sphere is given by projecting every vertex $\mathbf{p} \in \mathcal{M}$ onto a point $\mathbf{p}'$ on the sphere. Using the connectivity of $\mathcal{M}$, the points $\mathbf{p}'$ define a spherical mesh $\mathcal{M}'$ where the edges are replaced by circular arcs. *Evaluating* the parameterization **F** at some point $\mathbf{q}'$ on the sphere requires to first find that spherical triangle $\triangle'(A',B',C')$ of $\mathcal{M}'$ which contains $\mathbf{q}'$. The projection $\tilde{\mathbf{q}}'$ of $\mathbf{q}'$ onto the corresponding plane triangle $\triangle(A',B',C')$ can be expressed in barycentric coordinates

$$\tilde{\mathbf{q}}' = \alpha A' + \beta B' + \gamma C', \qquad \alpha + \beta + \gamma = 1.$$

Applying the same barycentric combination to the associated vertices A, B, and C of the original mesh $\mathcal{M}$ finally yields the function value $\mathbf{q} = \mathbf{F}(\mathbf{q}') \in \mathcal{M}$.

In order to make the parameterization well-defined, it is necessary to guarantee that the projection operator $\mathcal{M} \to \mathcal{M}'$ does not flip any triangles. Moreover, for numerical robustness of the evaluation routine, the projection method should bound the aspect ratio of the triangles in $\mathcal{M}'$.

Once we have the parameterization over the bounding sphere, we construct the base mesh $\mathcal{S}_0$. Our primary goal is to use a minimum number of triangles. We start with a fairly regular convex polyhedron $\mathcal{C}$ whose vertices lie on the sphere (e.g. an icosahedron). The segmentation of the sphere induced by the corresponding spherical polyhedron $\mathcal{C}'$ partitions the triangles of $\mathcal{M}'$ into disjoint subsets. For each of these subsets, we sum up the surface area of the associated original triangles in $\mathcal{M} = \mathbf{F}(\mathcal{M}')$. By this we assign an estimated surface area to every spherical triangle (to every *cell*) in $\mathcal{C}'$.

We can evaluate the parameterization **F** at the vertices of $\mathcal{C}$ to obtain the base mesh $\mathcal{S}_0 = \mathbf{F}(\mathcal{C})$. The surface areas associated with the cells of $\mathcal{C}$ then provide good estimates for the surface areas of the subdivision patches associated with the corresponding base triangle of $\mathcal{S}_0$.

Since the base triangles of $\mathcal{S}_0$ are large compared to the triangles of $\mathcal{M}$, we can neglect those triangles which are mapped onto one of the edges (great circles) of the spherical polyhedron $\mathcal{C}'$.

We perform a simple optimization procedure which applies relaxation and split operations to $\mathcal{C}$. The goal of the relaxation is to move the vertices in order to equalize the surface areas associated with each cell. The split operation supports this egalization by introducing new vertices in regions with high surface area per cell.

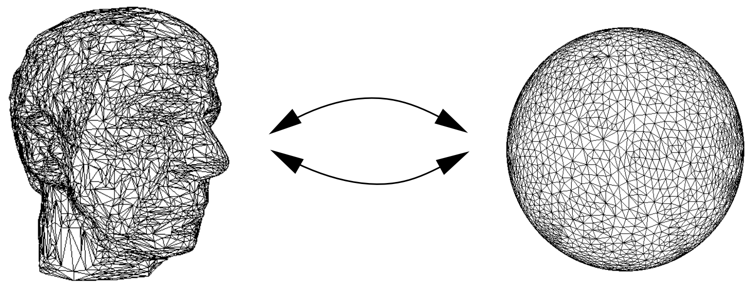

Figure 5: *An arbitrary genus-zero triangle mesh $\mathcal{M}$ can be parameterized over the sphere by projecting its vertices. Since both meshes have the same connectivity it is easy to establish a one-to-one correspondence between the continuous surfaces by barycentric parameterization of the triangles.*

The relaxation operator is a variant of the weighted umbrella operator (1). In this case the density d of a vertex $\mathbf{p}$ in the mesh $\mathcal{C}$ is set to the sum of the surface areas associated with its adjacent cells. Performing several iterations of (1) quickly converges to a stable configuration with all surface areas being approximately equal. The parameter α acts like a damping factor to prevent degenerate configurations. Notice that the vertices of $\mathcal{C}$ have to be projected back to the bounding sphere after every relaxation step and surface areas per cell have to be recomputed.

If the surface area varies drastically among the cells in the initial mesh $\mathcal{C}$, the result of the relaxation can have strongly distorted triangles. To avoid these, we apply a split operation when the variance of the associated areas is too big. The following heuristic leads to satisfactory results: We pick that edge for which the sum of the areas associated with its two adjacent cells is a maximum. Inserting the midpoint of that edge triggers a bisection of the two adjacent cells and hence generates four cells with smaller surface area.

After several steps of this procedure we have a spherical polyhedron $\mathcal{C}'$ on the bounding sphere such that the submeshes of $\mathcal{M}$ which are associated with the faces of the corresponding base mesh $\mathcal{S}_0 = \mathbf{F}(\mathcal{C}')$ according to the parameterization $\mathbf{F} : \mathcal{M}' \to \mathcal{M}$ have approximately the same size. The base mesh $\mathcal{S}_0$ is then used as the initial mesh for the multi-level shrink wrapping, i.e., the alternate density relaxation, projection, and uniform subdivision.

7. Stable projection

The second difficulty that made our primitive shrink wrapping algorithm fail on complicated models is the projection of the vertices of $\mathcal{S}_m$ (or $\mathcal{S}_i$ during the multi-level shrink wrapping) onto $\mathcal{M}$. This projection fails if the shapes of both meshes are too different from each other such that normal rays of $\mathcal{S}_i$ do not intersect with $\mathcal{M}$. To avoid such problems we have to find good starting shapes for the meshes $\mathcal{S}_i$ in order to make the projection save.

We do this by exploiting the sphere parameterization $\mathbf{F} : \mathcal{M}' \to \mathcal{M}$ of the input mesh $\mathcal{M}$ that we already used for the generation of the base mesh $\mathcal{S}_0$. Since the vertices of all intermediate meshes $\mathcal{S}_i$ lie on the surface $\mathcal{M}$, they inherit $\mathcal{M}$'s sphere parameterization in a unique way: For each vertex $\mathbf{p} \in \mathcal{S}_i$ we find the corresponding point $\mathbf{p}'$ on the bounding sphere by first representing $\mathbf{p}$ in barycentric coordinates with respect to a triangle $\triangle(A,B,C)$ of $\mathcal{M}$ and then applying the same barycentric combination to the corresponding triangle $\triangle(A',B',C')$ of $\mathcal{M}'$.

Let $\mathbf{p}$ be a vertex of $\mathcal{S}_i$ and $\mathbf{q}_0, \ldots, \mathbf{q}_{n-1}$ its adjacent neighbors. By sampling the sphere parameterization of $\mathcal{M}$ we obtain the corresponding pre-images $\mathbf{p}'$ and $\mathbf{q}'_0, \ldots, \mathbf{q}'_{n-1}$.

The reason for the instability of the smooth/project iteration is that after applying the umbrella rule (1) the reprojection of the updated vertex $\mathbf{p}$ may fail. We avoid this problem by deriving a modified umbrella rule which is applied to the pre-images instead. The projection of the updated pre-image $\mathbf{p}'$ back onto the bounding sphere is trival and always works correctly. Finally, evaluating the sphere parameterization $\mathbf{F}$ at the resulting location yields the updated position $\mathbf{p}$ which automatically lies on the surface of $\mathcal{M}$. Hence, by doing the relaxation step in the parameter domain $\mathcal{M}'$, we solve the problem of relaxing the mesh $\mathcal{S}_i$ while keeping the vertices on the geometry of $\mathcal{M}$ in a more elegant fashion.

It is obvious that the quality of the relaxation step in the parameter domain strongly depends on the distortion of the map $\mathcal{M}' \to \mathcal{M}$. With the technique proposed in [15] we minimize this distortion but we still have to solve the problem that the metric of the given surface $\mathcal{M}$ and its bounding sphere differ considerably. To cope with this situation, we modify the umbrella rule such that the weight coefficients reflect the surface metric on $\mathcal{M}$ while it is applied to points on $\mathcal{M}'$.

There are two different ways to accomplish this. The first technique has already been used for the generation of the optimized base mesh $\mathcal{S}_0$ in the last section. We use a density weighted umbrella relaxation where the density coefficients for each vertex are found by summing up the associated surface areas for all adjacent triangles.

As the vertices are moving on the bounding sphere, the associated surface areas have to be recomputed after every iteration. In Sect. 6 the area estimation has been simplified by ignoring those triangles of $\mathcal{M}'$ which are intersected by a cell boundary of $\mathbf{F}^{-1}(\mathcal{S}_0)$. This simplification is acceptable as long as the cells are large compared to the size of the individual triangles. However for higher refinement levels, it may occur that some cells of $\mathbf{F}^{-1}(\mathcal{S}_i)$ are empty. If this happens, we switch to the second relaxation technique which is still a variant of the density weighted umbrella but now the density values d are simply obtained by computing the average edge lengths $\|\mathbf{F}(\mathbf{p}') - \mathbf{F}(\mathbf{q}')\|$ on the original surface $\mathcal{M}$ (just like in (1).

With this modified relaxing rule, we can distribute the mesh vertices $\mathbf{p}'$ on the bounding sphere while in fact using the surface metric of the original input mesh $\mathcal{M}$. The overall procedure for computing the mesh $\mathcal{S}_m$ is described in Sect. 4 as a multi-level process where we start with the initial base mesh $\mathcal{S}_0$. We subdivide to obtain $\mathcal{S}_1$ and perform alternate projection and smoothing until we reach a stable configuration. Then we subdivide again to go to the next refinement level. In general, we use the result from level i to find a good starting mesh for level $i+1$. This accelerates the convergence since much fewer iterations are necessary on each level.

As the projection onto the mesh works savely only for higher refinement levels (where original geometry $\mathcal{M}$ and remeshed geometry $\mathcal{S}_i$ are sufficiently close), we use the modified relaxing rule for the lower levels. Hence, we perform the shrink wrapping on the meshes $\mathcal{S}'_0, \ldots, \mathcal{S}'_i$ in the parameter domain $\mathcal{M}'$. Once the approximation is tight enough to make the projection onto $\mathcal{M}$ stable, we switch to $\mathcal{S}_i$ and proceed with $\mathcal{S}_i, \ldots, \mathcal{S}_m$ on $\mathcal{M}$.

In our experiments, we found the decision when to switch from the relaxing in the parameter domain to the relaxing on the original geometry rather difficult and strongly dependent on the actual model. The two relevant criteria are that we should not switch too early to avoid wrong projection but we also should not switch too late since otherwise the final mesh quality is not optimal or has to be improved by a large number of relaxing steps. This diminishes the performance gain of the multi-level strategy. The reason for this behavior is that the metric in 3-space and the *simulated* metric in the parameter domain are similar but not exactly identical and hence the relaxation on the bounding sphere is a good approximation of the original operator but not optimal.

8. Results

We implemented the remeshing scheme and applied it to several polygonal models. The results are shown in the Figs. 6–8. Notice the extremly small number of triangles in the base mesh. This is achieved by the optimization technique described in Sect. 6.

8.1. Progressive transmission

With the increasing importance of distributed data bases, the need for effective algorithms which enable the transmission of geometric information via low-bandwidth data connections (like the internet) is becoming paramount. For unstructured triangle meshes, the eminent technique of *progressive meshes* [11] has established a de facto standard which provides data reduction potential of several orders of mangitude for the initial shape which can be displayed immediately by the receiver. As more detail information is arriving, the geometric model can be refined until the complete model has been transmitted. A slight disadvantage of this technique is the blocky nature of the coarse mesh models (from an esthetical point of view) and the lacking robustness against the loss of data packages.

In [1, 21, 25], wavelet techniques have been explored for the incremental transmission and load adaptive display of textured geometric models with subdivision connectivity. In these papers, the compression ratios obtained for particular choices of the scaling functions ϕ_i and the wavelet basis $\psi_{i,j}$ are discussed.

The conversion of arbitrary geometric models into meshes with subdivision connectivity enables the effective progressive transmission of surface geometry since a smooth approximation of the original surface can be reconstructed from a rather small amount of data. We implemented a simple method for the progressive transmission of subdivision connectivity meshes based on the Loop subdivision scheme [20] and the Butterfly algorithm [5, 34]. We transmit the base mesh first such that the receiver can reconstruct a smooth model by applying several subdivision steps. As more and more detail information is transmitted, the mesh is refined adaptively. The meshes vertices, for which the detail coefficients have not been received yet, are placed at the position predicted by the subdivision rule.

It turns out that we obtain a rather convincing reconstruction with only a minimal number of detail coefficients included into the model. Fig. 9–11 show several examples. Notice that progressive transmission of subdivision surfaces is robust against the loss and permutations of data packages since the detail coefficients are indexed independent from each other.

9. Conclusion

We presented a very effective approach for converting polygonal surfaces into meshes with subdivision connectivity. Our algorithm is based on a simulation of the shrink wrapping process. By using a low-distortion parameterization of the original model over a bounding sphere, we are able to find very coarse base meshes $\mathcal{S}_0$. The sphere parameterization also enables a stable multi-level relaxation technique which leads to good starting meshes for the actual shrink wrapping algorithm. This is necessary to make the projection step robust.

The algorithm is quite different from previous approaches since we do not construct an explicit parameterization which is sampled at the dyadic barycentric parameter values on each base triangle. Instead we distribute the mesh vertices evenly over the given geometry by an iterative relaxation process.

References

1. CERTAIN, A., POPOVIĆ, J., DEROSE, T., DUCHAMP, T., SALESIN, D., AND STUETZLE, W. Interactive Multiresolution Surface Viewing. In *Computer Graphics (SIGGRAPH 96 Proceedings)* (1996), pp. 91–98.
2. CIAMPALINI, A., CIGNONI, P., MONTANI, C., AND SCOPIGNO, R. Multiresolution Decimation Based on Global Error. *The Visual Computer 13* (1997), 228–246.
3. COHEN, L., AND COHEN, I. Finite element methods for active contour models and ballons for 2-d and 3-d images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1993), pp. 1131–1147.
4. DO CARMO, M. *Differential geometry of curves and surfaces.*
5. DYN, N., LEVIN, D., AND GREGORY, J. A. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics 9*, 2 (April 1990), 160–169.
6. ECK, M., DEROSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERY, M., AND STUETZLE, W. Multiresolution Analysis of Arbitrary Meshes. In *Computer Graphics (SIGGRAPH 95 Proceedings)* (1995), pp. 173–182.
7. FARIN, G. *Curves and Surfaces for CAGD*, 3rd ed. Academic Press, 1993.
8. GARLAND, M., AND HECKBERT, P. S. Surface Simplification Using Quadric Error Metrics. In *Computer Graphics (SIGGRAPH 97 Proceedings)* (1997), pp. 209–218.
9. GOLDSMITH, J., AND SALMON, J. Automatic creation of object hierarchies for ray tracing. *IEEE Computer Graphics and Applications 7*, 5 (May 1987), 14–20.
10. HACKBUSCH, W. *Multi-Grid Methods and Applications.* Springer Verlag, Berlin, 1985.
11. HOPPE, H. Progressive Meshes. In *Computer Graphics (SIGGRAPH 96 Proceedings)* (1996), pp. 99–108.
12. HOSCHEK, J., AND LASSER, D. *Fundamentals of Computer Aided Geometric Design*. AK Peters, 1993.
13. HUANG, J., YAGEL, R., FILIPPOV, V., AND KURZION, Y. An Accurate Method for Voxelizing Polygon Meshes. In *1998 Symposium on Volume Visualization* (1998), ACM, pp. 119–126.
14. KASS, M., A.WITKIN, AND TERZOPOULUS, D. Snakes: Active contour models. *International Journal of Computer Vision* (1988), 321–331.
15. KENT, J. R., CARLSON, W. E., AND PARENT, R. E. Shape transformation for polyhedral objects. In *Computer Graphics (SIGGRAPH '92 Proceedings)* (1992), pp. 47–54.
16. KOBBELT, L. Discrete fairing. In *Proceedings of the Seventh IMA Conference on the Mathematics of Surfaces* (1997), pp. 101–131.
17. KOBBELT, L., CAMPAGNA, S., AND SEIDEL, H.-P. A general framework for mesh decimation. In *Proceedings of the Graphics Interface conference '98* (1998).
18. KOBBELT, L., CAMPAGNA, S., VORSATZ, J., AND SEIDEL, H.-P. Interactive multi-resolution modeling on arbitrary meshes. In *Computer Graphics (SIGGRAPH 98 Proceedings)* (1998), pp. 105 – 114.
19. LEE, A., SWELDENS, W., SCHRÖDER, P., COWSAR, L., AND DOBKIN, D. Multiresolution adaptive parameterization of surfaces. In *Computer Graphics (SIGGRAPH 98 Proceedings)* (1998), pp. 95 – 104.
20. LOOP, C. Smooth subdivision surfaces based on triangles, 1987. Master's thesis, Utah University, USA.
21. LOUNSBERY, M., DEROSE, T., AND WARREN, J. Multiresolution Analysis for Surfaces of Arbitrary Topological Type. *ACM Transactions on Graphics 16*, 1 (January 1997), 34–73.
22. LÜRIG, C., KOBBELT, L., AND ERTL, T. Deformable surfaces for feature based indirect volume rendering. In *Computer Graphics International, IEEE Proceedings* (1998), pp. 752 – 760.
23. NEUENSCHWANDER, W. M. *Elastic Deformable Contour and Surface Models for 2-D and 3-D Image Segmentation.* PhD thesis, Eidgenössische Techn. Hochchule Zürich, 1995.
24. RONFARD, R., AND ROSSIGNAC, J. Full-Range Approximation of Triangulated Polyhedra. In *Computer Graphics Forum, Proceedings of Eurographics '96* (1996), pp. C67–C76.
25. SCHRÖDER, P., AND SWELDENS, W. Spherical wavelets: Efficiently representing functions on the

sphere. In *Computer Graphics (SIGGRAPH 95 Proceedings)* (1995), pp. 161–172.

26. SCHROEDER, W. J., ZARGE, J. A., AND LORENSEN, W. E. Decimation of Triangle Meshes. In *Computer Graphics (SIGGRAPH 92 Proceedings)* (1992), pp. 65–70.

27. STOLLNITZ, E., DEROSE, T., AND SALESIN, D. *Wavelets for Computer Graphics.* Morgan Kaufmann Publishers, 1996.

28. TAUBIN, G. A signal processing approach to fair surface design. In *Computer Graphics (SIGGRAPH 95 Proceedings)* (1995), pp. 351–358.

29. TERZOPOULUS, D. Regularization of inverse visual problems involving discontinuities. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1986).

30. TERZOPOULUS, D., A.WITKIN, AND KASS, M. Symmetrie-seeking models and 3d object construction. *International Journal of Computer Vision* (1987), 211–221.

31. VASILESCU, M., AND TERZOPOULUS, D. Adaptive meshes and shells: Irregular triangulation, discontinuities, and hierarchical subdivision. In *Proceedings of Computer Vision and Pattern Recognition conference* (1992), pp. 829–832.

32. VERFÜRTH, R. *A review of a posteriori error estimation and adaptive mesh refinement techniques.* Wiley-Teubner, 1996.

33. WANG, S., AND KAUFMAN, A. Volume sampled voxelization of geometric primitives. In *Visualization 1993* (1993), IEEE, pp. 78–84.

34. ZORIN, D., SCHRÖDER, P., AND SWELDENS, W. Interpolating subdivision for meshes with arbitrary topology. In *Computer Graphics (SIGGRAPH 96 Proceedings)* (1996), pp. 189–192.

35. ZORIN, D., SCHRÖDER, P., AND SWELDENS, W. Interactive multiresolution mesh editing. In *Computer Graphics (SIGGRAPH 97 Proceedings)* (1997), pp. 259–268.

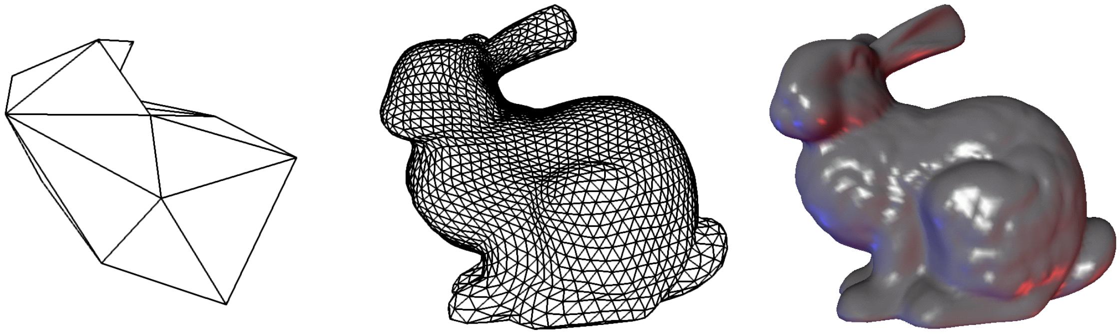

Figure 6: *Remeshing the Stanford bunny. The original data set has 70K triangles. Our remeshing algorithm generates a base mesh with 38 triangles (left). The center and right image shows the 3rd and the 5th refinement level.*

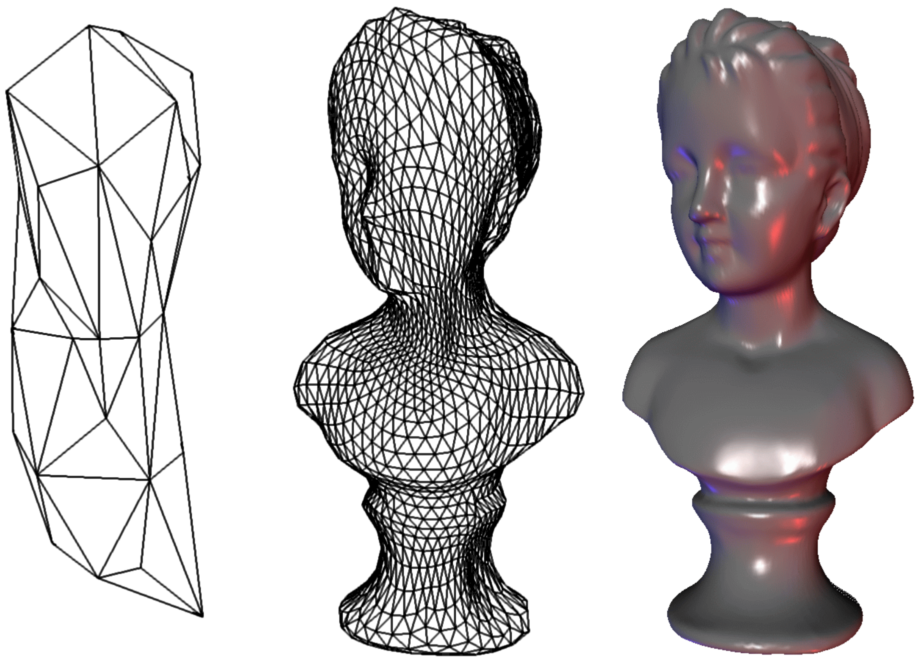

Figure 7: *The original bust model has 61K triangles. The base mesh with 72 triangles is subdivided three times to generate the center mesh and 5 times to generate the right image.*

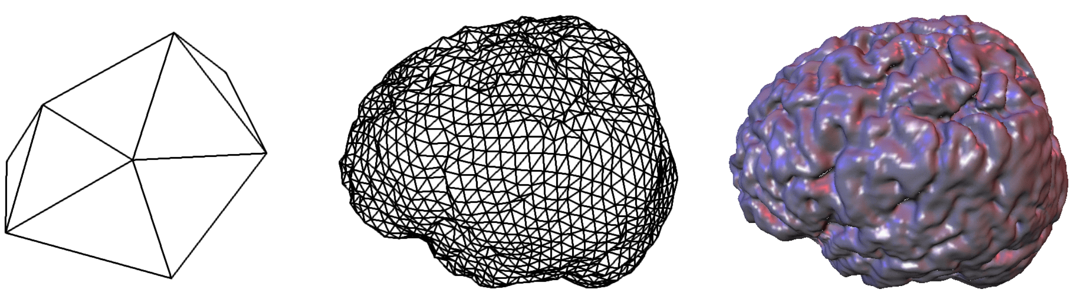

Figure 8: *A brain data set extracted from CT volume data is remeshed. The base mesh has only 20 triangles. In the center, the 4th refinement level is shown and the 6th level on the right side.*

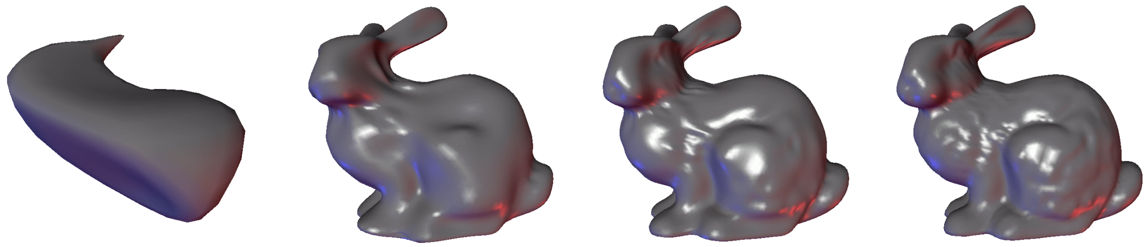

Figure 9: *Progressive transmission of the bunny. The left image shows the Loop subdivision surface generated from the base mesh which is transmitted first. The other models are obtained after including 1%, 3%, and 10% of the detail coefficients.*

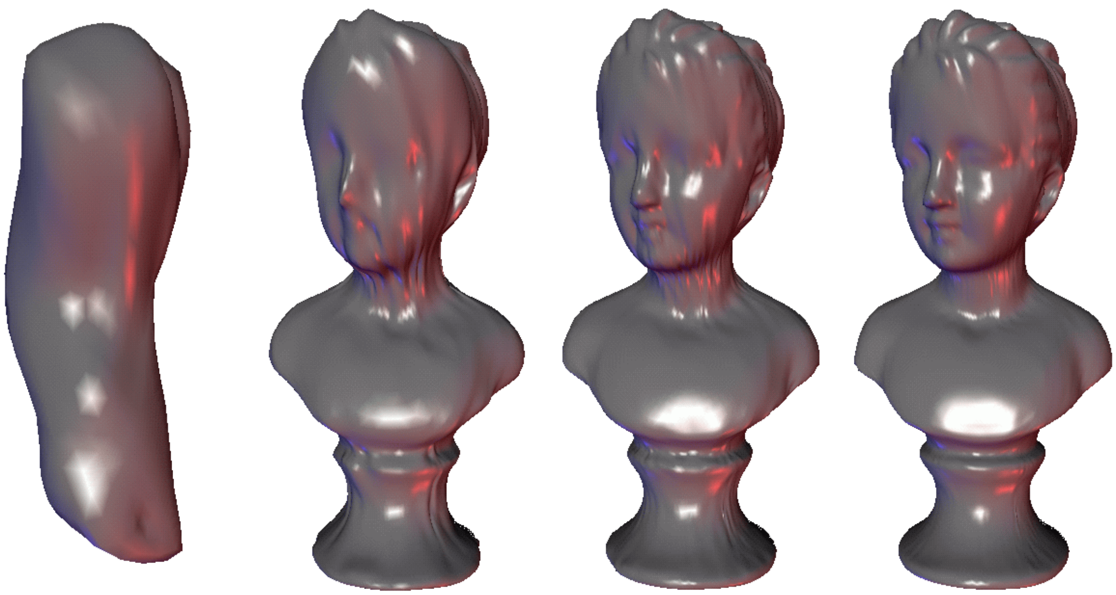

Figure 10: *Progressive Transmission of the bust model. From left to right: 0%, 1%, 3%, and 10% of the detail coefficients.*

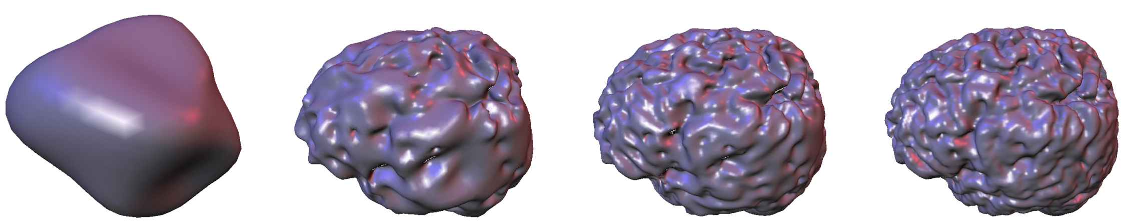

Figure 11: *Progressive Transmission of the brain model. From left to right: 0%, 1%, 3%, and 10% of the detail coefficients.*