Using Most Isometric Parametrizations for Remeshing Polygonal Surfaces

U. Labsik K. Hormann G. Greiner

Computer Graphics Group, University of Erlangen-Nuremberg,
Am Weichselgarten 9, 91058 Tennenlohe, Germany
Tel: +49-9131-8529919, Fax: +49-9131-8529931
Email: labsik@informatik.uni-erlangen.de

Abstract

The importance of triangle meshes with a special kind of connectivity, the so-called subdivision connectivity *is still growing. Therefore it is important to develop efficient algorithms for converting a given mesh with arbitrary connectivity into one with subdivision connectivity. In this paper we have focused on 2-manifold triangle meshes with boundary and no holes. We discuss the importance of a parametrization with minimal distortion for the process of remeshing. Based on the concept of most isometric parametrizations we have developed a remeshing algorithm for the given class of triangle meshes. A series of examples shows the advantages of our approach.*

1. Introduction

In computer graphics, triangle meshes are a standard representation for surfaces. The use of 3D acquisition techniques like laser range scanning results in dense meshes with a large number of triangles. Because of their complexity it is necessary to use efficient algorithms for storing, transmitting and editing those meshes. Many of the algorithms in the context of multiresolution modeling need a special structure of the mesh, the so called *subdivision connectivity*. This special kind of connectivity is generated by iteratively subdividing a coarse base mesh $\mathcal{S}^0$ with a uniform refinement operator.

With most meshes not having this special kind of connectivity it is necessary to have algorithms that enable the transformation of a mesh with arbitrary connectivity into a mesh with subdivision connectivity. This process is called *remeshing*. Several approaches exist to solve this problem, e. g. [3, 12, 10]. There are many advantages resulting from this conversion. The different refinement levels automatically provide levels of detail which can be used by multiresolution algorithms like level-of-detail rendering [2], wavelets [14, 18], progressive transmission [11] or multiresolution editing [21].

In this paper we present an algorithm for remeshing triangulated, topologically disk-like surfaces. Such a triangle mesh can be parametrized over a planar domain. We demonstrate the influence of the parametrization method for the quality of the remesh. For our algorithm we use most isometric parametrizations (MIPS) [7] providing minimal distortion of the triangles in the mesh. This parametrization has the main advantage that the boundary is allowed to develop naturally and is not fixed in advance.

The paper is organized as follows. In Section 2 a short overview of related work in the field of remeshing polygonal surfaces is given. The general concept of our algorithm is explained in Section 3. In Section 4 we show how to compute a most isometric parametrization of a given triangle mesh and in Section 5 the remeshing algorithm is explained in detail. In Section 6 we demonstrate the efficiency of our algorithm by some examples. We close with a summary in Section 7.

2. Previous Work

During the last years a lot of effort has been spent on the task of remeshing polygonal surfaces. In this section we will give an overview of the most important work.

In [3] Eck et al. have presented a remeshing algorithm consisting of three steps. First, a partitioning of the given mesh $\mathcal{M}$ into a number of triangular regions has to be found. This is done by distributing vertices over $\mathcal{M}$ and growing Voronoi tiles around each vertex. With the help of the constructed Voronoi diagram a triangulation of the new vertices can be found. In the next step a parametrization of the given mesh within each triangle of the base mesh is computed. This is done with harmonic maps trying to minimize the local distortion. The remesh is achieved by uniformly subdividing each base triangle and mapping the vertices into 3-space using the parametrization.

A different approach is proposed by Lee et al. in [12]. Here the mesh $\mathcal{M}$ is coarsened by vertex removal leading to a base mesh for the remeshing. A parametrization is obtained by projecting the removed vertex onto the remaining mesh. This parametrization is also only locally smooth. To achieve global smoothness of the remesh the sampling of the parametrization is not done at the dyadic points. A smoothed version of the dyadic points is computed instead by applying a variant of Loop's subdivision scheme [13] and mapping these vertices into 3-space.

The basic idea of the shrink wrapping approach [10] by Kobbelt et al. is to place a triangle mesh with subdivision connectivity around an arbitrary triangle mesh and shrink it onto the surface of that mesh. In this process two forces have to be simulated, an attracting force, moving the vertices of the subdivision in the direction of the original surface and a relaxing force, distributing the vertices over the surface. With this approach it is possible to remesh topologically sphere-like (genus-0) meshes.

3. General concept

The task of remeshing can be understood as the problem of finding an approximation operator $\mathfrak{A} : \mathbb{M} \to \mathbb{S}$ that maps from a given set $\mathbb{M}$ of meshes to the set $\mathbb{S}$ of all meshes with subdivision connectivity such that the resulting mesh is a good approximation of the original mesh, i. e., the distance of the two meshes is below a user-specified threshold. The resulting subdivision connectivity mesh is called a *remesh* of the original mesh.

In this paper we concentrate on the set of 2-manifold triangle meshes that have a boundary and no holes. In such a *spatial* mesh the intersection of two triangles may be either an edge, a vertex or empty. Additionally one edge of the mesh belongs to either two or only one triangle. In this case it is a boundary edge. The mesh $\mathcal{M}_\Psi = (P, T)$ is represented by a set of *vertices* $P = \{P_i\}$ in the *space* $\Psi \subset \mathbb{R}^3$, representing the geometry, and a set $T = \{T_j\}$ of triangles $T_j = \Delta(P_{j_0}, P_{j_1}, P_{j_2})$, defining the connectivity of the mesh.

A subdivision connectivity mesh (SCM) $\mathcal{S}_\Psi$ of level m has the further property of being a refinement of a coarse base mesh $\mathcal{S}^0$, i. e., it is the last element of a sequence of meshes $\mathcal{S}^0, \ldots, \mathcal{S}^m = \mathcal{S}_\Psi$ where each $\mathcal{S}^{l+1}$ emerges from $\mathcal{S}^l$ by uniformly subdividing each triangle of $\mathcal{S}^l$ into 4 subtriangles (cf. Fig. 1). If the vertices P^l of $\mathcal{S}^l$ are a subset of the vertices P^{l+1}, this operation is called *interpolatory subdivision*, and *non-interpolatory subdivision* otherwise. Note that all vertices of $\mathcal{S}_\Psi$ have valence 6, except for the boundary vertices and those who correspond to the vertices P^0 of the base mesh.

Two different aspects relate to the *quality* of a SCM. First, the number of triangles of the base mesh $\mathcal{S}^0$ shall be as small as possible, which results in a large number of hierarchy levels m of the remesh $\mathcal{S}_\Psi$. This assures a maximal utilization of the multiresolution techniques that can be applied to $\mathcal{S}_\Psi$. Another criterion for the quality of a SCM is the visual appearance, i. e., all triangles T_j of $\mathcal{S}_\Psi$ should have uniform size and aspect ratio.

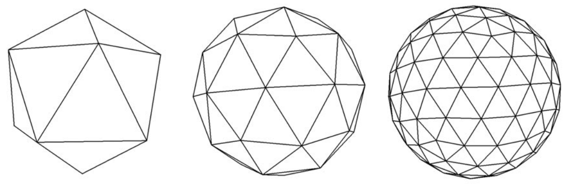


Figure 1. Starting with a coarse base mesh $\mathcal{S}_0$, a sequence of SCMs can be generated by iteratively applying the uniform subdivision operator that performs a 1-to-4 split on every triangle.

Since we consider only meshes with boundary and no holes, these meshes are topologically disk-like and can therefore be parametrized over a simply connected *planar domain* $\Omega \subset \mathbb{R}^2$ by assigning each vertex P_i a *parameter value* $p_i \in \Omega$. By triangulating these parameter values in the same way as the original data points, i. e., by defining a set of triangles $t = \{t_j\}$, $t_j = \Delta(p_{j_0}, p_{j_1}, p_{j_2})$ we obtain a *planar* mesh $\mathcal{M}_\Omega = (p, t)$ that corresponds to $\mathcal{M}_\Psi$ in a natural way. We call the function $f : \Psi \to \Omega$ that linearly maps each spatial triangle to the corresponding planar triangle (i. e., $f(T_j) = t_j$, $f(P_i) = p_i$ and $f(\mathcal{M}_\Psi) = \mathcal{M}_\Omega$) the *projection* of $\mathcal{M}_\Psi$, whereas the inverse function $F = f^{-1}$ is called the *parametrization*.

The general concept of our remeshing algorithm is to use the projection and the parametrization in order to shift the approximation problem from the space Ψ to the planar domain Ω and consider the operator $\mathfrak{A}_\Omega$ instead of $\mathfrak{A}_\Psi$ (cf. Fig. 2). After determining a projection f with minimal distortion we approximate the planar mesh $\mathcal{M}_\Omega$ with a high quality SCM $\mathcal{S}_\Omega$ that is mapped back into space by the parametrization F to yield a remesh $\mathcal{S}_\Psi = F(\mathcal{S}_\Omega)$ of the original mesh $\mathcal{M}_\Psi$. Note that this method guarantees that the vertices of the remesh lie on the surface of the given mesh.

4. Parametrization

The parametrization problem is vital for many applications in computer graphics such as surface fitting, texture mapping, and remeshing. The most important type of this problem is the parametrization of 3D data points which has been addressed in many papers before [1, 3, 4, 5, 15, 16, 17].

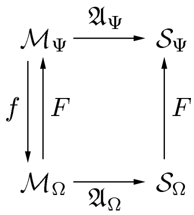


Figure 2. The general concept of our method: $\mathfrak{A}_\Psi \approx F \circ \mathfrak{A}_\Omega \circ f$.

In general, a set of points $P_i \in \mathbb{R}^3$, a certain kind of neighborhood information that defines the topology of the point set, e. g. a triangulation, and a domain Ω that is of the same topological type are given. The task is now to assign every vertex P_i a parameter value $p_i \in \Omega$ such that the topology is preserved. Typical domains are the disk $D_2 = \{x \in \mathbb{R}^2 : \|x\| \leq 1\}$ and the sphere $S_2 = \{x \in \mathbb{R}^3 : \|x\| = 1\}$, but other objects like the torus are possible as well. In this section we will concentrate on the special case of parametrizing triangulated point sets that are topologically disk-like and a simply connected planar domain Ω, but the general idea of the presented parametrization technique is also applicable for other domain types.

Using the notation of the previous section, we are looking for a projection $f : \Psi \rightarrow \Omega$. Since the given mesh $\mathcal{M}_\Psi$ is normally geometrically complex, this function will inevitably cause some deformation to the shape of the triangles T_j. A projection without distortion is called *isometric* and can only be found for developable surfaces, e. g. planes, cylinders and conical surfaces. Such a projection would be optimal for our purposes, because the parametrization $F = f^{-1}$, that is used to create the final remesh $\mathcal{S}_\Psi$ from the planar remesh $\mathcal{S}_\Omega$, would be isometric, too. Hence, $\mathcal{S}_\Psi$ would inherit the quality from $\mathcal{S}_\Omega$, since the shape of the triangles would not change. Therefore, we need to find a projection f that is "as isometric as possible".

The concept of MIPS [7] can be used to find such projections. This approach decomposes the piecewise linear projection f into *atomic linear maps* f_j that map the spatial triangle T_j to the corresponding planar triangle t_j. By introducing a local coordinate system at T_j with the third axis chosen to be perpendicular to T_j, these linear maps can be written as $f_j : \mathbb{R}^2 \rightarrow \mathbb{R}^2, x \mapsto A_j x + b_j$ and the deformation of the triangle T_j can be measured by the *2-norm condition* of the matrix A_j,

$$\kappa_2(A_j) = \|A_j\|_2 \, \|A_j^{-1}\|_2 = \frac{\sigma_1}{\sigma_2},$$

where $\sigma_1 \geq \sigma_2 \geq 0$ are the *singular values* of A_j. However, for practical reasons it is easier to use the *Frobenius norm condition* κ_F, which is closely related to κ_2 and can be computed by the following formula (see [17] and [7] for details)

$$\begin{aligned} \kappa_F(A_j) &= \kappa_2(A_j) + \frac{1}{\kappa_2(A_j)} \\ &= \frac{\cot\alpha_{j_0}\|\ell_{j_0}\|^2 + \cot\alpha_{j_1}\|\ell_{j_1}\|^2 + \cot\alpha_{j_2}\|\ell_{j_2}\|^2}{2\,\mathrm{area}(t_j)} \end{aligned} \quad (1)$$

with the notations of Fig. 3, i. e.,

$$\ell_{j_0} = p_{j_2} - p_{j_1}, \quad \ell_{j_1} = p_{j_2} - p_{j_0}, \quad \ell_{j_2} = p_{j_1} - p_{j_0},$$

and

$$2\,\mathrm{area}(t_j) = \|\ell_{j_1} \times \ell_{j_2}\| = \ell^x_{j_1}\ell^y_{j_2} - \ell^y_{j_1}\ell^x_{j_2}.$$

Minimization of $\kappa = \sum_j \kappa_F(A_j)$ will then lead to an optimal projection f in the sense of minimal deformation.

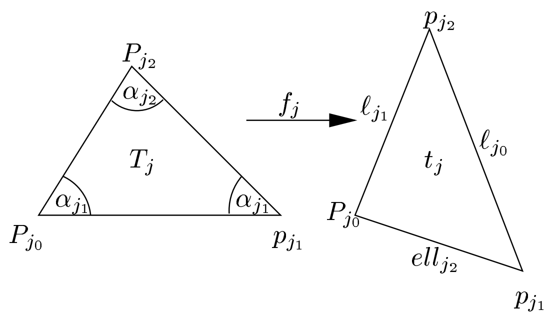


Figure 3. The atomic linear map f_j maps the spatial triangle T_j to the corresponding planar triangle t_j.

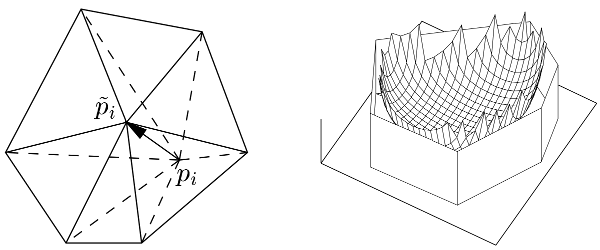
This is an optimization problem in the unknown parameter values p_i and according to Eq. (1) highly nonlinear. Instead of solving this problem globally, we use a Gauss-Seidel approach to minimize κ with a series of local optimization steps. Starting with an initial set of parameter values that can be obtained by one of the methods described in [3, 4, 5], we perform the following algorithm, illustrated in Fig. 4(a).

> repeat
>
> choose a vertex p_i by random
>
> let $t_{i_1}, \ldots, t_{i_n}$ be the triangles that surround p_i
>
> fix the positions of all parameter values except for p_i
>
> minimize the local deformation energy $\kappa_i = \sum_{k=1}^{n} \kappa_F(A_{i_k})$ in order to get the optimal position $\tilde{p}_i$ of p_i
>
> until numerical convergence

As we can see in Fig. 4(b), the local functional κ_i is convex and grows to infinity along the edges that surround the parameter value p_i. Usually the optimal $\tilde{p}_i$ can be found with two of three Newton steps. It can be proven that these local optimization steps converge to the global optimum and are stable in the sense of validity: if the initial configuration is valid (i. e., neighboring triangles do not overlap) this property is preserved by each optimization step.

A major advantage of MIPS is that the optimization step can also be performed on the parameter values at the boundary of the triangulation and does not need the boundary to be fixed in advance. Of course this may lead to overlaps of different boundary regions (cf. Fig. 5) although we never experienced it in our examples. However, if you think of the different overlapping parts being assigned to different depth layers, it is still possible to use this configuration in order to “lift” the planar SCM mesh $\mathcal{S}_\Omega$ up to the space Ψ as long as you add some depth layer information to the geometric information of the vertices p_i of $\mathcal{S}_\Omega$, e. g. by storing the number of the triangle of $\mathcal{M}_\Omega$ in which each p_i is located.

We close this section by briefly mentioning that the optimization process of MIPS can be significantly accelerated by a multiresolution approach as explained in [8]. The idea



(a) (b)

Figure 4. Each optimization step finds the optimal position $\tilde{p}_i$ of the parameter value p_i (a) by minimizing the convex local functional κ_i (b).

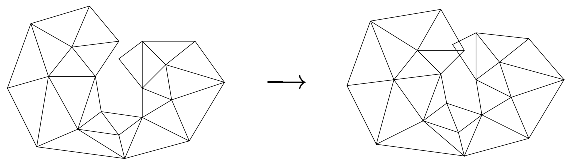


Figure 5. MIPS may generate overlapping parts in the parameter domain.

is to use the concept of Progressive Meshes [6] to build a hierarchy $\mathcal{M}_\Psi = \mathcal{M}^0, \ldots, \mathcal{M}^k$ of meshes and create the optimal projection for the coarsest mesh $\mathcal{M}^k$ first. Then the remaining vertices of $\mathcal{M}_\Psi$ are successively inserted and optimized until the final planar mesh $\mathcal{M}_\Omega$ is generated.

5. Remeshing

As described in Section 3 the task of generating a remesh $\mathcal{S}_\Psi$ with subdivision connectivity of a given mesh $\mathcal{M}_\Psi$ consists of three steps. We have already explained how to compute the parametrization of $\mathcal{M}_\Psi$. The second step is to find an appropriate remesh $\mathcal{S}_\Omega$ for the mesh $\mathcal{M}_\Omega = f(\mathcal{M}_\Psi)$ in the parameter domain Ω.

In order to find the remesh $\mathcal{S}_\Omega$ in the parameter domain we have to solve several problems which are discussed in this section. First, we have to determine an appropriate base mesh $\mathcal{S}^0_\Omega$, then we have to subdivide this base mesh until a prescribed error bound is reached. In this subdivision process we have to reconstruct the boundary polygon of $\mathcal{M}_\Omega$ with the remesh $\mathcal{S}^l_\Omega$ and the triangles $t_j \in \mathcal{S}^l_\Omega$ have to be scaled so that the corresponding triangles $T_j = F(t_j)$ are of equal size.

5.1. Constructing the base mesh

For remeshing meshes with boundary it would be possible to use just a single triangle as a base mesh. But in most cases that would not lead to a satisfying remesh of the given surface because of a high distortion of the triangles. So it is important to find a good starting mesh $\mathcal{S}^0_\Omega$ to avoid this problem. This base mesh should meet all quality criterions mentioned in Section 3, i. e., a few number of triangles being as equilateral as possible. Note that the latter criterion refers to the triangles mapped into 3-space. Furthermore the base mesh should approximate the boundary of $\mathcal{M}_\Omega$.

One important aspect of the remeshing algorithm is the reconstruction of boundary of $\mathcal{M}_\Psi$. So we start the construction of the base mesh by specifying extraordinary vertices on the boundary polygon of the given mesh $\mathcal{M}_\Psi$ which consists of a set of vertices $B = \{B_0, B_1, ..., B_n = B_0\}$. To achieve this we compute the angle $\measuredangle(\overrightarrow{B_iB_{i-1}}, \overrightarrow{B_iB_{i+1}})$

of the boundary polygon in every vertex. If the value exceeds a given threshold α the corresponding vertex $b_i = f(B_i)$ of the mesh $\mathcal{M}_\Omega$ will be a vertex $\tilde{b}_j$ of the base mesh $\mathcal{S}_\Omega^0$.

These vertices $\tilde{b} = \{\tilde{b}_i\}$ may be distributed irregularly over the boundary polygon. To avoid triangles of different size in the base mesh we now specify vertices on the boundary polygon so that the distance between all boundary vertices in the base mesh is approximately equal. The distance of two boundary vertices $\tilde{b}_i$ and $\tilde{b}_j$ of the base mesh $\mathcal{S}_\Omega^0$ is computed as follows. Using the parametrization we obtain $\tilde{B}_i = F(\tilde{b}_i)$ and $\tilde{B}_j = F(\tilde{b}_j)$ on the boundary polygon of $\mathcal{M}_\Psi$. We now compute the distance between these points by following the boundary line from one point to the other (cf. Fig. 6).

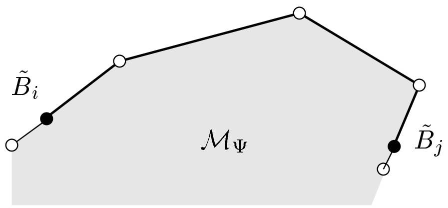


Figure 6. Computing the distance between two boundary points.

The set of vertices we have chosen up to now defines the boundary of the base mesh and has to be triangulated. Notice that the vertices also lie on the boundary polygon of $\mathcal{M}_\Omega$. Triangulating only the boundary vertices would not lead to a regular structure with almost equilateral triangles. Therefore the triangulation algorithm is allowed to insert new interior vertices, called *Steiner points*, to the base mesh. There are two constraints for the position of these interior vertices. The angles of the triangles should be close to $60°$ and the area representing the triangles in the original mesh $\mathcal{M}_\Psi$ should be about the same.

This algorithm enables us to compute a base mesh $\mathcal{S}_\Omega^0$ which is the starting point of the remeshing process. In practice these base meshes will have only a small number of triangles.

5.2. Border adaption

When parametrizing the mesh $\mathcal{M}_\Psi$ using the MIPS scheme, normally the boundary is allowed to develop naturally. Hence, in general case the parametrization will not be a rectangular area, not even convex. All boundary vertices of $\mathcal{S}_\Omega^l$ also belong to the boundary polygon of $\mathcal{M}_\Omega$.

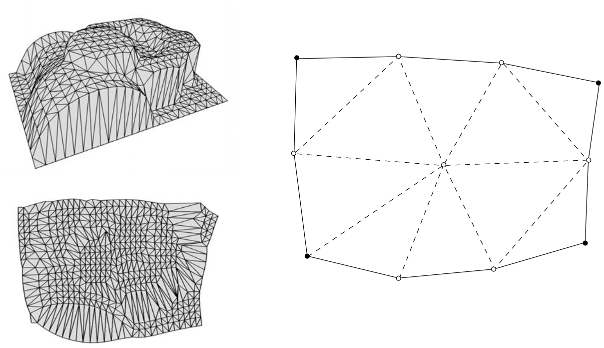


Figure 7. Constructing a base mesh. The filled vertices were found by searching for extraordinary vertices of the original mesh, the outlined boundary vertices were inserted to adapt the distances on the boundary polygon and the interior vertex was inserted to improve the mesh structure.

The subdivision scheme we use just computes the new vertices as the midpoints of the edges on the coarser level $\mathcal{S}_\Omega^l$ and retriangulates the mesh leading to $\mathcal{S}_\Omega^{l+1}$. The additional boundary vertices of $\mathcal{S}_\Omega^{l+1}$ will not be on the boundary of $\mathcal{M}_\Omega$ but somewhere inside or outside the mesh (cf. Fig. 8).

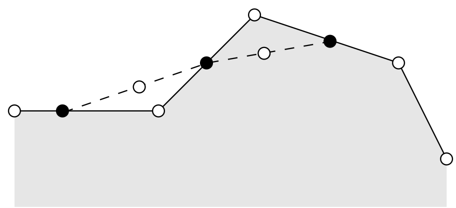


Figure 8. New boundary vertices of the remesh (dashed) have to be moved onto the boundary polygon of the parametrization.

For these vertices a new position on the boundary polygon of the parametrization has to be found. The algorithm we use works as follows. We have to find a position for the vertex $\tilde{q}$, which has been inserted between the boundary vertices $\tilde{b}_i$ and $\tilde{b}_{i+1}$. Exploiting the parametrization we obtain $\tilde{B}_i = F(\tilde{b}_i)$ and $\tilde{B}_{i+1} = F(\tilde{b}_{i+1})$ which are points on the boundary of $\mathcal{M}_\Psi$. We now compute the distance between these points over the boundary line of the mesh as described in the last section and search for the position $\tilde{Q}$ midway between these points. The new position of the vertex $\tilde{q}$ in the parameter domain is then computed by $\tilde{q} = f(\tilde{Q})$. If the distances of the boundary vertices in the mesh $\mathcal{S}_\Omega^l$ are equal then this strategy is optimal and will lead to equidistant boundary vertices in $\mathcal{S}_\Omega^{l+1}$.

5.3. Density weighted smoothing

The deformation functional κ that we use to obtain the parametrization is invariant to scalings (see Eq. (1)). This may lead to scalings of triangles between the meshes $\mathcal{M}_\Psi$ and $\mathcal{M}_\Omega$. Hence, to minimize the distortion and to get approximately the same size for each triangle T_j of $\mathcal{S}_\Psi^m$ it is essential for the remeshing algorithm to use a relaxing operator to optimize the interior vertices of the remesh. For this relaxing we use the umbrella operator $\mathfrak{U}$ [9]. The umbrella operator minimizes the membrane energy of a mesh, i. e., the surface area. The update rule for a vertex $p_i \in S_\Omega^l$ is

$$\mathfrak{U} : p_i \mapsto \tilde{p}_i = (1-\omega)p_i + \frac{\omega}{\sum_j d_{i_j}} \sum_{j=1}^{n} d_{i_j} p_{i_j} \qquad (2)$$

with n being the valence of vertex p_i and $p_{i_1}, ..., p_{i_n}$ its adjacent neighbors in S_Ω^l. With the help of the density coefficients d_i the distribution of the vertices can be controlled.

In our case we want a vertex p to be moved so that neighboring triangles transformed into 3-space have approximately the same size. Therefore we use the sum of the area of the adjacent triangles T_j as the density coefficient d_i for the vertex p_i. We apply the density weighted umbrella only to the interior vertices of the mesh. The boundary vertices are not affected by this relaxing operation.

The density weighted umbrella operator is applied iteratively. With the vertices moving on the parameter domain the size of the triangles has to be recomputed after every iteration of the operator $\mathfrak{U}$. The iterations stop when the change of the vertex positions falls below a certain threshold. Then the mesh $\mathcal{S}_\Omega^l$ can be switched to the next finer level $\mathcal{S}_\Omega^{l+1}$ and the process of border adaption and relaxing iterations starts again. When reaching a certain level m the subdivision process is stopped and the mesh is transformed into 3-space by evaluating the parametrization F. Therefore we have to find the triangle $\triangle(A_\Omega, B_\Omega, C_\Omega) \in \mathcal{M}_\Omega$ which contains a vertex $p_i \in \mathcal{S}_\Omega^m$. This vertex can be expressed in barycentric coordinates

$$p_i = \alpha A_\Omega + \beta B_\Omega + \gamma C_\Omega, \quad \alpha + \beta + \gamma = 1.$$

The position of p_i in 3-space is determined by applying the same barycentric combination to the corresponding vertices A_Ψ, B_Ψ and C_Ψ of the original mesh $\mathcal{M}_\Psi$, i. e.,

$$P_i = \alpha A_\Psi + \beta B_\Psi + \gamma C_\Psi.$$

By using the MIPS parametrization which generates minimal distortion it is not necessary to use the smoothing and projecting technique in 3-space as proposed in [10]. So the risk of failing projections is excluded. Examples which are remeshed with our algorithm are shown in Section 6.

5.4. Adaptive remeshing

In order to reach the user specified error threshold for the distance of the two meshes $\mathcal{M}$ and $\mathcal{S}$ in regions of small local features the mesh $\mathcal{S}$ has to be subdivided up to a very fine level. When using uniform subdivision the number of triangles will grow exponentially. As in [12] we use a straight forward adaptive subdivision approach to keep the number of triangles small and avoid the overhead of first subdividing the mesh to a fine level and afterwards decimating triangles by a wavelet threshold like in [3, 2].

Adaptive subdivision means to split only those triangles which do not satisfy some prescribed criterion. Other triangles can remain coarse. There are a few restrictions in this approach. In order to keep the number of special configurations small we will only allow *balanced* meshes, i. e., the refinement level of two neighboring triangles may only differ by one. In order to avoid cracks in the mesh where two triangles from different levels meet, we have to use a special technique, the so-called *red-green triangulation* [19, 20]. A normal 1-to-4 split is called *green split*. To fix cracks in the mesh, triangle bisection is used, which is called *red split*. A red split is only temporary, i. e., if a red split triangle is to be further subdivided in a subsequent refinement step then the red split is undone first and a green split is applied to the original triangle.

As a subdivision criterion we compute the distance function

$$D(t_j) = \max_i \ \text{dist}(F(p_i), F(t_j)) \qquad (3)$$

for every triangle of the actual mesh $\mathcal{S}_\Omega^l$. In this equation p_i are vertices of $\mathcal{M}_\Omega$ which reside in the triangle t_j of $\mathcal{S}_\Omega^l$. The computation of the distance function is performed in 3-space Ψ with the help of the parametrization F. If the maximum distance between such a vertex $F(p_i)$ and the plane defined by the triangle $T_j = F(t_j)$ is greater than a given threshold ε then the triangle will be further subdivided. With this method only a minimal number of triangles has to be refined.

For adaptive remeshing we also have to adapt the density weighted umbrella. The density of a vertex can only be computed on the lowest level of all adjacent triangles. When computing the density weighted umbrella function for a vertex, we have to check the level of the densities of all adjacent vertices and can perform the umbrella on the lowest level of all these densities. Another problem occurs on vertices where a red and a green split meet. For these vertices we can not perform the density weighted umbrella function. So we first apply the smoothing operation to all possible vertices, for the rest we compute the new position as the midpoint of the edge the vertex is dividing.

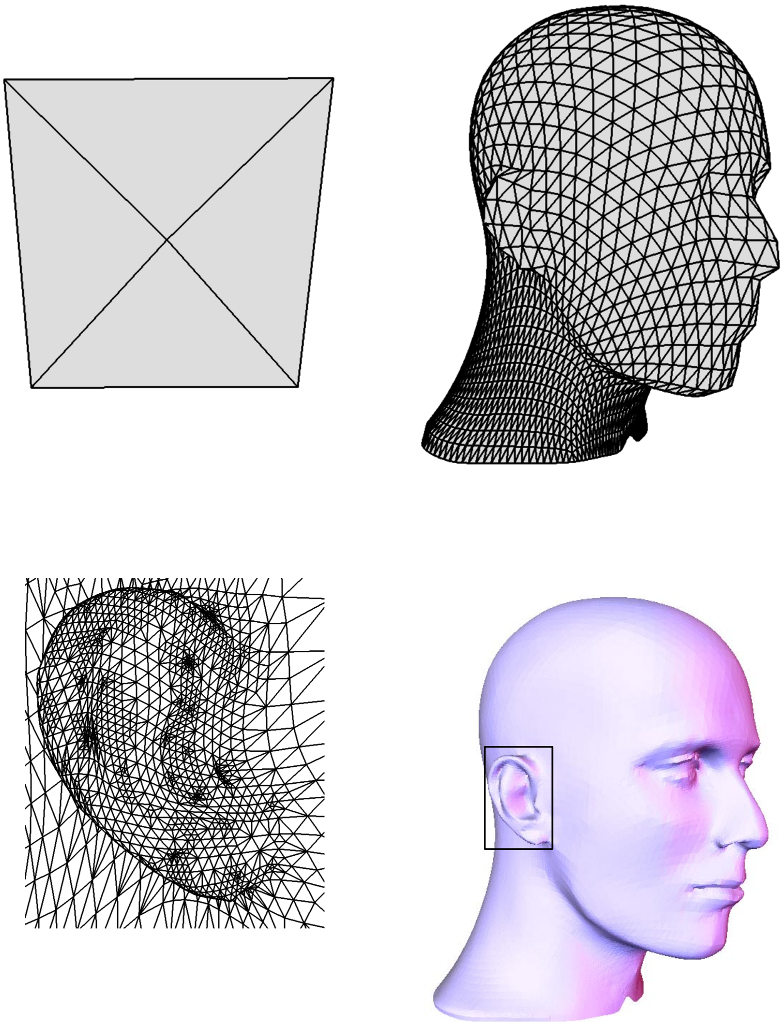


Figure 9. Remeshing the head data set. The base mesh consists of four triangles (top left). The top right image shows the 6th refinement level with uniform subdivision, below the adaptively refined mesh is presented.

6. Examples

In this section we show some of the results we have produced with our remeshing algorithm. We also illustrate the quality of the used parametrization method by showing examples with different parametrizations.

Fig. 9 shows a remesh of a head data set consisting of 21680 triangles. We used a simple mesh of four triangles as a base mesh. By uniform subdivision and smoothing with the density umbrella operator we get triangles of almost equal size in the remesh. But local detail like the ear must be sampled with a higher density. Therefore adaptive subdivision can be used. Using an error bound of $\varepsilon = 0.1\%$ leads to an adaptively refined remesh with 30530 triangles.

In Fig. 10 the remeshing process of a triangulated data set of 7938 triangles is illustrated. The base mesh consists of 10 triangles. The remesh is shown in the 5th refinement level. Using a uniform parametrization for the remeshing process leads to a mesh with highly distorted triangles. The structure of the triangles is much better when the discrete harmonic parametrization is used and can further be improved at the boundary of the mesh by using MIPS.

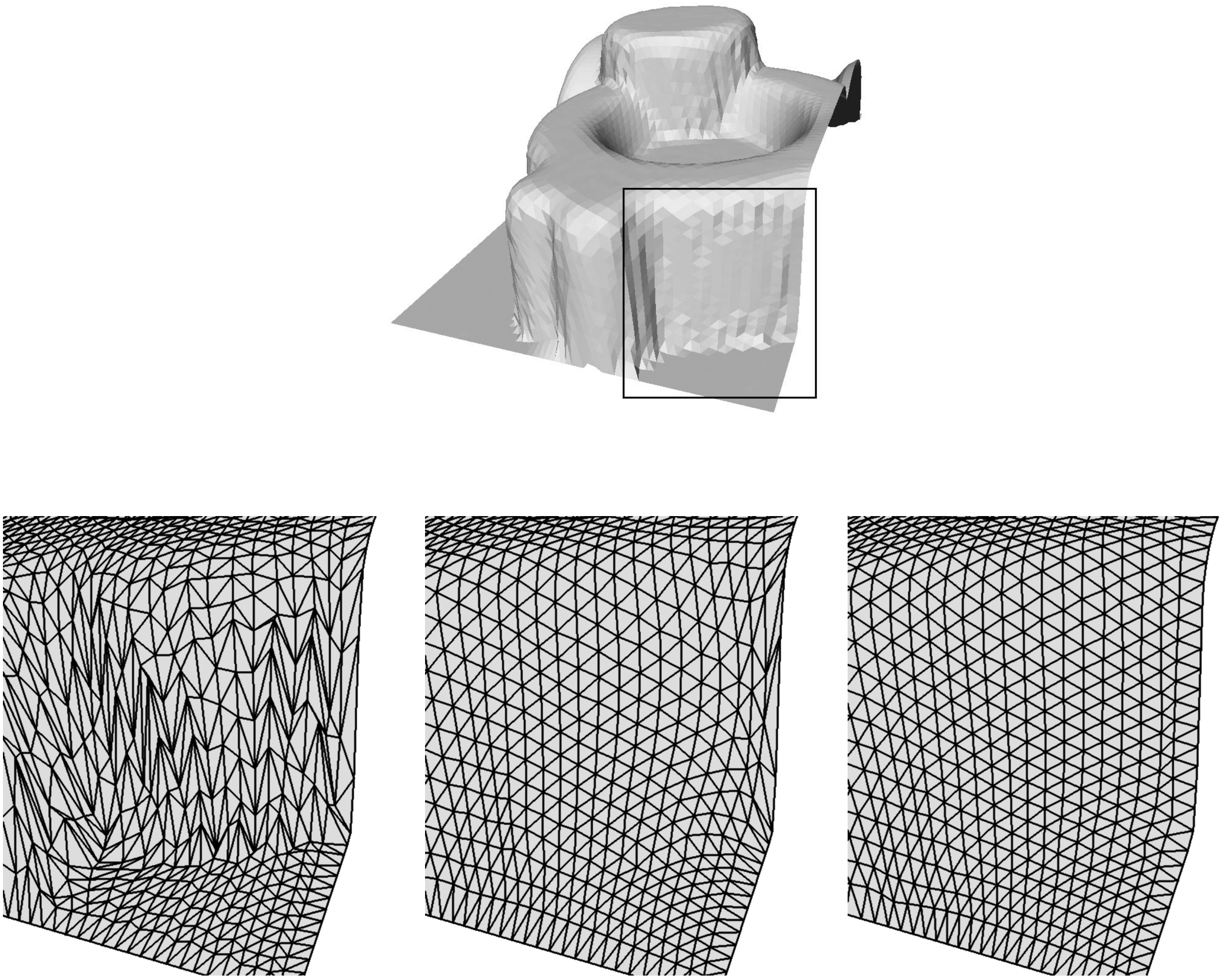


Figure 10. The quality of the remesh with different parametrizations. From left to right: uniform, discrete harmonic and most isometric.

7. Conclusion

In this paper we have presented an efficient algorithm for converting triangle meshes with arbitrary connectivity into meshes with subdivision connectivity. The algorithm works on triangle meshes with boundary and no holes. For this class of meshes we can compute a planar parametrization with minimal distortion. We now compute a remesh for this parametrization which can be mapped into 3-space again to achieve a remesh of the original mesh. By using an iterative relaxation process on every level of the remesh we adapt the remesh to the local scalings of the parametrization.

With this method all triangles of a uniformly refined remesh will have almost equal size. In highly detailed regions it is often necessary to have a higher sampling density of the original mesh. Therefore we use adaptive subdivision to compute remeshes with guaranteed error bounds.

By using the most isometric parametrization on a sphere the results for remeshing polygonal surfaces of genus 0 with the shrink-wrapping approach [10] can be improved, too. There, a parametrization produced with a density weighted umbrella was used which is not able to guarantee minimal distortion of the triangles in the parameter domain.

Future work will aim at remeshing not only disk-like objects without holes. To be able to handle meshes of arbitrary topology an appropriate segmentation of the whole object has to be found. Then the segments can be remeshed separately with the presented algorithm and merged afterwards. Therefore it is important that adjacent segments share the vertices on their common boundary.

References

[1] C. Bennis, J.-M. Vézien, and G. Iglésias. Piecewise surface flattening for non-distorted texture mapping. In *ACM Comp. Graph. (SIGGRAPH '91 Proc.)*, pages 237–246, 1991.

[2] A. Certain, J. Popović, T. DeRose, T. Duchamp, D. Salesin, and W. Stuetzle. Interactive multiresolution surface viewing. In *ACM Comp. Graph. (SIGGRAPH '96 Proc.)*, pages 91–98, 1996.

[3] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *ACM Comp. Graph. (SIGGRAPH '95 Proc.)*, pages 173–182, 1995.

[4] M. S. Floater. Parameterization and smooth approximation of surface triangulations. *CAGD*, 14:231–250, 1997.

[5] G. Greiner and K. Hormann. Interpolating and approximating scattered 3D data with hierarchical tensor product B-splines. In A. Méhauté, C. Rabut, and L. L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, pages 163–172. Vanderbilt University Press, 1997.

[6] H. Hoppe. Progressive meshes. In *ACM Comp. Graph. (SIGGRAPH '96 Proc.)*, pages 99–108, 1996.

[7] K. Hormann and G. Greiner. MIPS: an efficient global parametrization method. In *Saint-Malo Proceedings*. Vanderbilt University Press, 2000. to appear.

[8] K. Hormann, G. Greiner, and S. Campagna. Hierarchical parametrization of triangulated surfaces. In B. Girod, H. Niemann, and H.-P. Seidel, editors, *Vision, Modeling and Visualization '99*, pages 219–226. infix, 1999.

[9] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *ACM Comp. Graph. (SIGGRAPH '98 Proc.)*, pages 105–114, 1998.

[10] L. Kobbelt, J. Vorsatz, U. Labsik, and H.-P. Seidel. A shrink wrapping approach to remeshing polygonal surfaces. In *Comp. Graph. Forum (EUROGRAPHICS '99 Proc.)*, 1999.

[11] U. Labsik, L. Kobbelt, R. Schneider, and H.-P. Seidel. Progressive transmission of subdivision surfaces. 1999. to appear.

[12] A. Lee, W. Sweldens, P. Schröder, L. Coswar, and D. Dobkin. Multirelosution adaptive parametrization of surfaces. In *ACM Comp. Graph. (SIGGRAPH '98 Proc.)*, pages 95–104, 1998.

[13] C. Loop. Smooth sudivsion surfaces based on triangles. Master's thesis, Utah University, 1987.

[14] M. Lounsbery, T. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. Technical report 93-10-05, University of Washington, Department of Computer Science and Engineering, 1993.

[15] W. Ma and J. P. Kruth. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *CAD*, 27:663–675, 1995.

[16] J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In *ACM Comp. Graph. (SIGGRAPH '93 Proc.)*, pages 27–34, 1993.

[17] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Exp. Math.*, 2(1):15–36, 1993.

[18] P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. In *ACM Comp. Graph. (SIGGRAPH '95 Proc.)*, pages 161–172, 1995.

[19] M. Vasilescu and D. Terzopoulus. Adaptive meshes and shells: Irregular triangulation, discontinuities, and hierarchical subdivision. In *Proceedings of Computer Vision and Pattern Recognition conference*, pages 829–832, 1992.

[20] R. Verfürth. *A review of a posteriori error estimation and adaptive mesh refinement techniques*. Wiley-Teubner, 1996.

[21] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In *ACM Comp. Graph. (SIGGRAPH '97 Proc.)*, pages 259–268, 1997.