Quadrilateral Remeshing

Kai Hormann[*] Günther Greiner

Computer Graphics Group, University of Erlangen-Nürnberg
Am Weichselgarten 9, 91058 Erlangen, Germany
Email: {hormann, greiner}@informatik.uni-erlangen.de

Abstract

The use of polygonal meshes, especially triangle meshes, is manifold but a lot of algorithms require the mesh to be structured in a certain way and cannot be applied to an arbitrarily shaped mesh. The process of replacing an arbitrary mesh by a structured one is called *remeshing*. Triangle meshes with *subdivision connectivity* are an important class of structured meshes and have been studied thoroughly in the past. In this paper we concentrate on another class of regular meshes instead and present a method for replacing an arbitrary triangle mesh by a *regular quadrilateral* mesh. These kind of meshes can later be used for a very simple and efficient surface reconstruction method with tensor product B-spline surfaces.

1 Introduction

Polygonal meshes are used in many fields of computer science for various purposes, e. g. modeling, visualization, and simulation. Especially in computer graphics, triangle meshes are of great importance as a standard surface representation due to the simplicity and flexibility of this piecewise linear description and the fact that they are widely supported by the graphics hardware. But since meshes with millions of triangles are not uncommon these days, even this simple surface description can still be quite awkward to handle because of the sheer size of data that has to be processed.

If the triangle meshes are structured, however, efficient multiresolution algorithms like level-of-detail rendering [2], progressive transmission [17], wavelet decomposition [19, 21], or multiresolution editing [22] can be applied. The special structure required by these algorithms is the *subdivision connectivity*, which is generated by iteratively refining a base mesh and is furthermore characterized by the property of almost all vertices having valence six (cf. Fig. 1a). The process of transforming a mesh with arbitrary connectivity into one with subdivision connectivity is called *remeshing*, and several approaches exist to solve this problem [4, 15, 16, 18].

In this paper we have focused on remeshing triangle meshes with another type of structured meshes and present an algorithm for converting a given mesh into a *regular quadrilateral* one, i. e. a mesh with quadrilateral facets and all vertices having valence four except for the boundary vertices (cf. Fig. 1b).

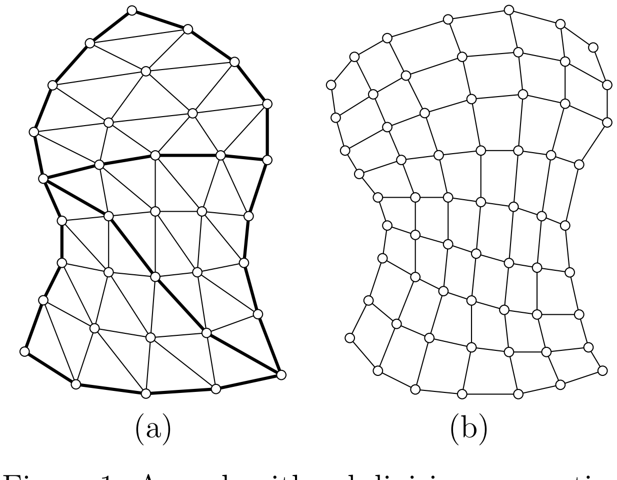
(a) (b)

Figure 1: A mesh with subdivision connectivity (a) and a regular quadrilateral mesh (b).

[*] The author has been supported by the SFB 603.

It follows easily from *Euler's Characteristic* that the only closed objects that can be modeled by a non-degenerated regular quadrilateral mesh are the torus and Klein's bottle, so we regard only meshes with boundary. But since any closed object can be partitioned into a collection of such bordered patches, this is no serious restriction.

The main idea of our algorithm is to circumvent the three-dimensional remeshing problem $\mathfrak{R}_3$ by flattening the initial mesh and solving the two-dimensional problem $\mathfrak{R}_2$ instead (cf. Fig. 2). Since we consider only meshes with boundary, it is always possible to flatten them, i. e. to find a corresponding two-dimensional mesh $\mathcal{T}_2$, which is called the *parametrization*. The *deflation* function $f : \mathbb{R}^3 \to \mathbb{R}^2$ is then defined by linearly mapping each triangle of $\mathcal{T}_3$ to the corresponding triangle in $\mathcal{T}_2$ while the inverse *inflation* function $F = f^{-1}$ enables us to get back from $\mathbb{R}^2$ to $\mathbb{R}^3$. Section 2 shows different methods of constructing parametrizations. Section 3 discusses how to remesh the projected mesh $\mathcal{T}_2$ with a regular quadrilateral mesh $\mathcal{Q}_2$ and how this mesh is lifted back into $\mathbb{R}^3$ to obtain a remesh $\mathcal{Q}_3$ of the initial mesh. The remesh $\mathcal{Q}_3$ can be either used as the base mesh of a quadrilateral subdivision scheme [13] or as the input data of a simple and efficient reconstruction method that approximates the given data by a tensor product B-spline surface as explained in Section 4. In Section 5 we illustrate the results of the algorithm by showing some examples and close with a summary in Section 6.

2 Parametrizations

The problem of parametrizing triangle meshes is fundamental for many applications in computer graphics and computer aided geometric design (CAGD) and has been addressed by many authors before. In this section we give a brief overview of the different methods and discuss which of them is applicable for the remeshing algorithm.

The methods in [4, 7, 9, 20] are generalizations of the well-known parametrization methods for curves and have the following strategy in common: First, they define a parametrization of the *boundary vertices*, then the *interior vertices* of the parametrization are found by solving a sparse linear system. All these approaches aim at minimizing the distortion that inevitably occurs when a complex triangle mesh is flattened and, except for the spring-energy method of Greiner and Hormann in [9], have a reproduction property, i. e. whenever the initial mesh $\mathcal{T}_3$ is planar, the resulting parametrization $\mathcal{T}_2$ is an affine mapping of $\mathcal{T}_3$. Since the discrete harmonic map that is used by Pinkall and Polthier in [20] and Eck et al. in [4] is not guaranteed to avoid foldovers, we propose to use the shape preserving parametrization presented by Floater in [7]. This method is a generalization of the other approaches and generates parametrizations without foldovers that are very similar to those obtained by the harmonic map.

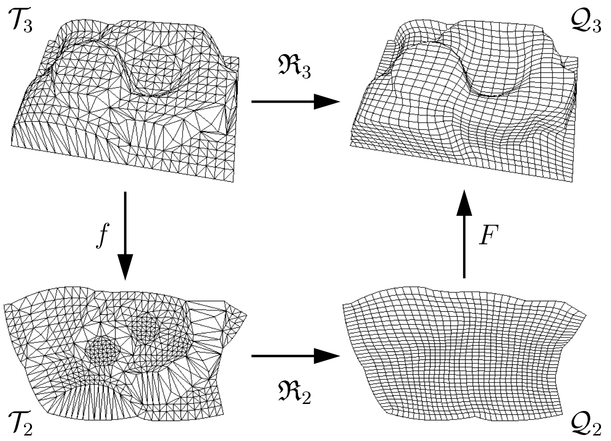

Figure 2: The main idea: $\mathfrak{R}_3 \approx F \circ \mathfrak{R}_2 \circ f$.

A drawback of the previously mentioned methods is that the criterion used for minimizing the distortion can only by applied to the interior vertices while the parametrization of the boundary vertices is chosen heuristically. An approach to overcome this restriction that also guarantees to avoid foldovers and has the reproduction property are the most isometric parametrizations presented by Hormann and Greiner in [11]. They determine the parametrization by minimizing a deformation functional that measures the distortion of the deflation function f that linearly maps

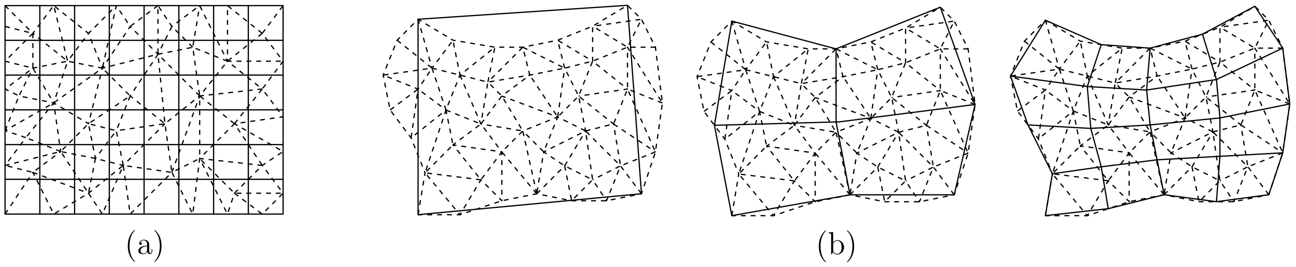

Figure 3: Two-dimensional remeshing of a triangle mesh with a regularly spaced rectangular grid (a) and a regular quadrilateral mesh (b).

each triangle of $\mathcal{T}_3$ to the corresponding triangle in $\mathcal{T}_2$. The main advantage of this method is that the boundary vertices can be included in the optimization process, thereby avoiding the distortions that tend to occur near the boundary of the parametrizations obtained by the other techniques. On the other hand, this approach requires the minimization of a rational quadratic function and is therefore quite slow. But the computational costs can be reduced dramatically by using the hierarchical parametrization technique presented in [12], which can also be applied to the other methods. Naturally, this approach also allows to fix the boundary vertices at previously chosen positions as needed for one of the remeshing techniques illustrated in the next section. In this case the results are very similar to the shape preserving and the discrete harmonic parametrization.

3 Remeshing

After having flattened down the initial triangle mesh to a planar domain, the remeshing procedure reduces to the two-dimensional problem $\mathfrak{R}_2$ that is much easier to solve than the original problem $\mathfrak{R}_3$. All we have to do is to create a planar regular quadrilateral mesh $\mathcal{Q}_2$ such that its boundary $\partial\mathcal{Q}_2$ coincides with $\partial\mathcal{T}_2$. Once this planar remesh is found, we can use the inflation function F to lift it back in $\mathbb{R}^3$ in order to get the remesh $\mathcal{Q}_3$ of $\mathcal{T}_3$. The vertices P_i of $\mathcal{Q}_3$ are determined by detecting for each vertex p_i of $\mathcal{Q}_2$ the surrounding triangle of $\mathcal{T}_2$, computing the barycentric coordinates with respect to this triangle and linearly interpolating the corresponding triangle of $\mathcal{T}_3$ in $\mathbb{R}^3$ with these barycentric coordinates.

The easiest way of finding such a $\mathcal{Q}_2$ is enabled by forcing the boundary vertices of the parametrization $\mathcal{T}_2$ to form a rectangle. Then we can simply take a regularly spaced rectangular grid as $\mathcal{Q}_2$ (cf. Fig. 3a). This approach requires the specification of four corner vertices in the given mesh $\mathcal{T}_3$, which correspond to the corners of the rectangular parameter domain. This can be done by considering the boundary polygon $\partial\mathcal{T}_3$ of the mesh and taking those four vertices with the smallest interior angle. If the shape of the boundary polygon fails to show four outstanding vertices, one can also choose these corner vertices to be uniformly distributed on $\partial\mathcal{T}_3$. The latter method was used for the head data set in Fig. 12, since the boundary of that mesh more resembles a circle than a quadrilateral. The remaining boundary vertices will then be distributed on the sides of the rectangle by one of the standard parametrization method for curves. In our examples the *chord length* parametrization yielded the best results.

Another remeshing method has to be applied if we do not fix the boundary of $\mathcal{T}_2$ and rather let it develop naturally by using the most isometric parametrization scheme. Again we have to identify four vertices to form the corners of the remesh, but this time we can detect them directly on the two-dimensional boundary of $\mathcal{T}_2$ rather than analyzing the three-dimensional $\partial\mathcal{T}_3$. The quadrilateral formed by these vertices already is a coarse remesh $\mathcal{Q}_2$ of the parametrization $\mathcal{T}_2$ which is now refined to obtain more detailed versions of $\mathcal{Q}_2$ (cf. Fig. 3b). In order to achieve this, we iteratively split all faces of $\mathcal{Q}_2$ into four quadrilaterals and determine the positions of the new vertices as follows. The new boundary vertices are located on the boundary of the parametrization $\mathcal{T}_2$ such that the distance along $\partial\mathcal{T}_2$ to the neighboring boundary vertices is the same. The new interior vertices are first set at the center of the edge or face they arise from and then optimized by applying discrete Laplace smoothing, which is also called the *umbrella* operator [14]. This operator successively moves each interior vertex p to the barycenter of its four neighboring vertices (cf. Fig. 4a)

$$p = \frac{p_{\text{N}} + p_{\text{E}} + p_{\text{S}} + p_{\text{W}}}{4}$$

and needs only a few iterations to converge to a stable configuration.

By applying this smoothing operator we aim at two of the three criterions that characterize a good remesh. On the one hand the shape of $\mathcal{Q}_2$'s faces become as rectangular as possible, on the other hand they all have approximately the same size. Note that the simple method explained before perfectly meets both conditions. But since we finally want to attain a good remesh $\mathcal{Q}_3$ of the initial mesh $\mathcal{T}_3$, we should think about the effect of the inflation function on the appearance of the quadrilateral faces. As the parametrizations we use for flattening $\mathcal{T}_3$ minimize the distortion of the deflation function f, the same property holds for the inverse inflation function F, and thus the shape deformation of the faces is kept small. I. e., a rectangle in $\mathcal{Q}_2$ will also be similar to a rectangle in $\mathcal{Q}_3$, but the size of the faces may be modified by the inflation mapping, as we can see in Fig. 11. So instead of using the umbrella operator for smoothing the planar remesh, we apply a weighted version of this operator (cf. Fig. 4b)

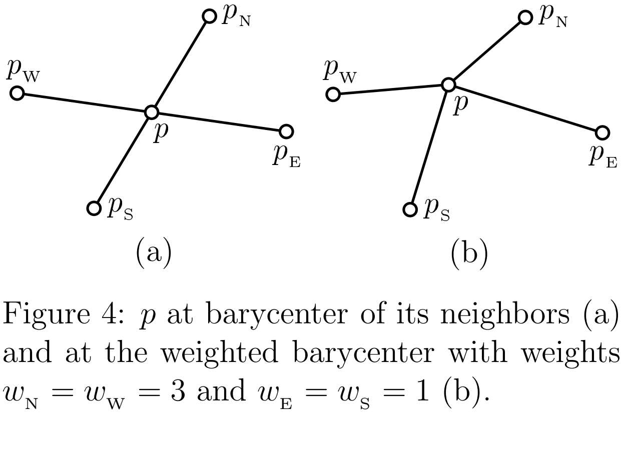

Figure 4: p at barycenter of its neighbors (a) and at the weighted barycenter with weights $w_{\text{N}} = w_{\text{W}} = 3$ and $w_{\text{E}} = w_{\text{S}} = 1$ (b).

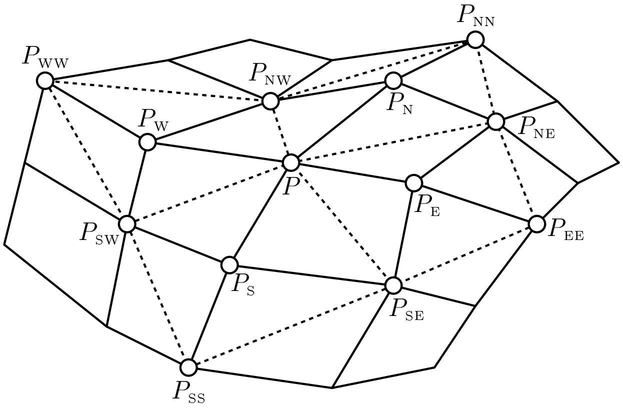

Figure 5: Calculating the weights for the modified umbrella operator. E. g., w_{S} is the lateral area of the pyramid with top point P_{S} and lower base $\square(P, P_{\text{SE}}, P_{\text{SS}}, P_{\text{SW}})$, marked by dotted lines.

$$p = \frac{w_{\text{N}}p_{\text{N}} + w_{\text{E}}p_{\text{E}} + w_{\text{S}}p_{\text{S}} + w_{\text{W}}p_{\text{W}}}{w_{\text{N}} + w_{\text{E}} + w_{\text{S}} + w_{\text{W}}},$$

choosing the weights such that the smoothing in $\mathbb{R}^2$ imitates a smoothing of the final remesh $\mathcal{Q}_3$ in $\mathbb{R}^3$. This effect is achieved by transforming p and all its surrounding vertices to $\mathbb{R}^3$ (e. g. $P = F(p)$) and assigning the area of the pyramid formed by P_i, $i \in \{\text{N}, \text{E}, \text{S}, \text{W}\}$ and the four neighboring vertices to the corresponding weight w_i (cf. Fig. 5). This results in the vertex p being pushed towards the neighbor with the largest weight and surrounding area resp. and thus in more uniformly sized quadrilaterals of the three-dimensional remesh $\mathcal{Q}_3$.

The third characterization of a good remesh is the distance to the original mesh, which can be measured by the *Hausdorff distance.* This is obviously fulfilled for the planar remesh because the Hausdorff distance of $\mathcal{T}_2$ and $\mathcal{Q}_2$ is zero, but it becomes more important if we look at $\mathcal{T}_3$ and $\mathcal{Q}_3$ in $\mathbb{R}^3$. Since the inflation function F guarantees the vertices of $\mathcal{Q}_3$ to lie on the surface of $\mathcal{T}_3$, the distance between the two meshes is the maximum distance between

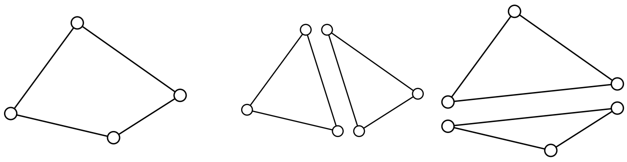

Figure 6: The convex hull of a quadrilateral (left) is formed by four triangles (right).

the vertices of $\mathcal{T}_3$ and the facets of $\mathcal{Q}_3$. This distance can be approximated by determining the surrounding quadrilateral in $\mathbb{R}^2$ and measuring the distance to the convex hull of the corresponding facet in $\mathbb{R}^3$, which is the minimum distance to the four triangles that can be constructed with the corners of that quadrilateral (cf. Fig. 6).

It is easy to show that any user specified error tolerance for the distance between $\mathcal{T}_3$ and $\mathcal{Q}_3$ can be kept by iteratively refining the remesh up to a certain level, but the number of vertices will grow exponentially this way. It is therefore more adequate to refine $\mathcal{Q}_3$ adaptively, i. e. splitting only those facets where the distance to one of the vertices of $\mathcal{T}_3$ exceeds the tolerance. Unfortunately, it is impossible to retain the regularity of the quadrilateral mesh this way, so we must either accept T-vertices or resolve them by adding Y-elements (see [13]) to obtain a non-regular but quadrilateral remesh (cf. Fig. 7).

4 Surface Reconstruction

Despite the practical use of triangle meshes there exist a lot of applications that require surface descriptions with a higher order of continuity, which is essential for representing surfaces with continuous normal fields or continuous curvatures. Common CAGD software as well as modern modeling and animation tools typically use tensor product B-spline or NURBS surfaces for these reasons. The representation of real world objects in such systems necessitates to scan them and to reconstruct surfaces from the measured data, a process known as *reverse engineering.* Since the scanned data is usually unstructured and contains measurement errors, *Scattered Data Approximation* is a typical surface reconstruction method: First, a two-dimensional parameter value p_i is assigned to each data point P_i (e. g. by one of the parametrization methods discussed in Sec. 2), then the control points of the approximating surface $S : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ are found in an optimization process that minimizes the distance $\|P_i - S(p_i)\|$ between the data points and the corresponding points on the surface. Unfortunately, the resulting surfaces tend to oscillate heavily and fairness functionals have to be added if smooth surfaces are desired [3, 8, 9, 10].

Since this problem does not occur as badly for other types of surfaces, e. g. triangular Bézier surfaces, it seems as if the rigid tensor product structure does not get along with the scattered data points very well, although this phenomenon has not been fully understood yet. A solution to this problem is provided by the remeshing algorithm explained above, which replaces the scattered data points by a set of structured vertices and enables a very simple and efficient surface reconstruction method: tensor product B-spline interpolation. I. e., we interpolate the vertices of the quadrilateral remesh $\mathcal{Q}_3$, thereby approximating the original data values of $\mathcal{T}_3$ *indirectly* rather than approximating them *directly* as in the previously mentioned approach. We will now briefly explain how the interpolating spline surface is computed and refer to [1, 6] for a detailed discussion.

Suppose we have a regular quadrilateral remesh $\mathcal{Q}_3$ with vertices $P_{00}, \ldots, P_{mn}$. First, we choose two knot sequences $\mathbf{u}$ and $\mathbf{v}$ and the cubic B-spline basis functions M_i and N_j defined thereon. Since the knot spacing

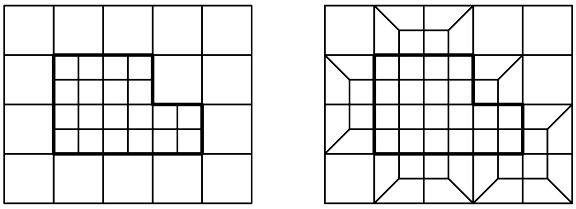

Figure 7: Adaptively refined quadrilateral mesh with T-vertices (left) and Y-elements (right).

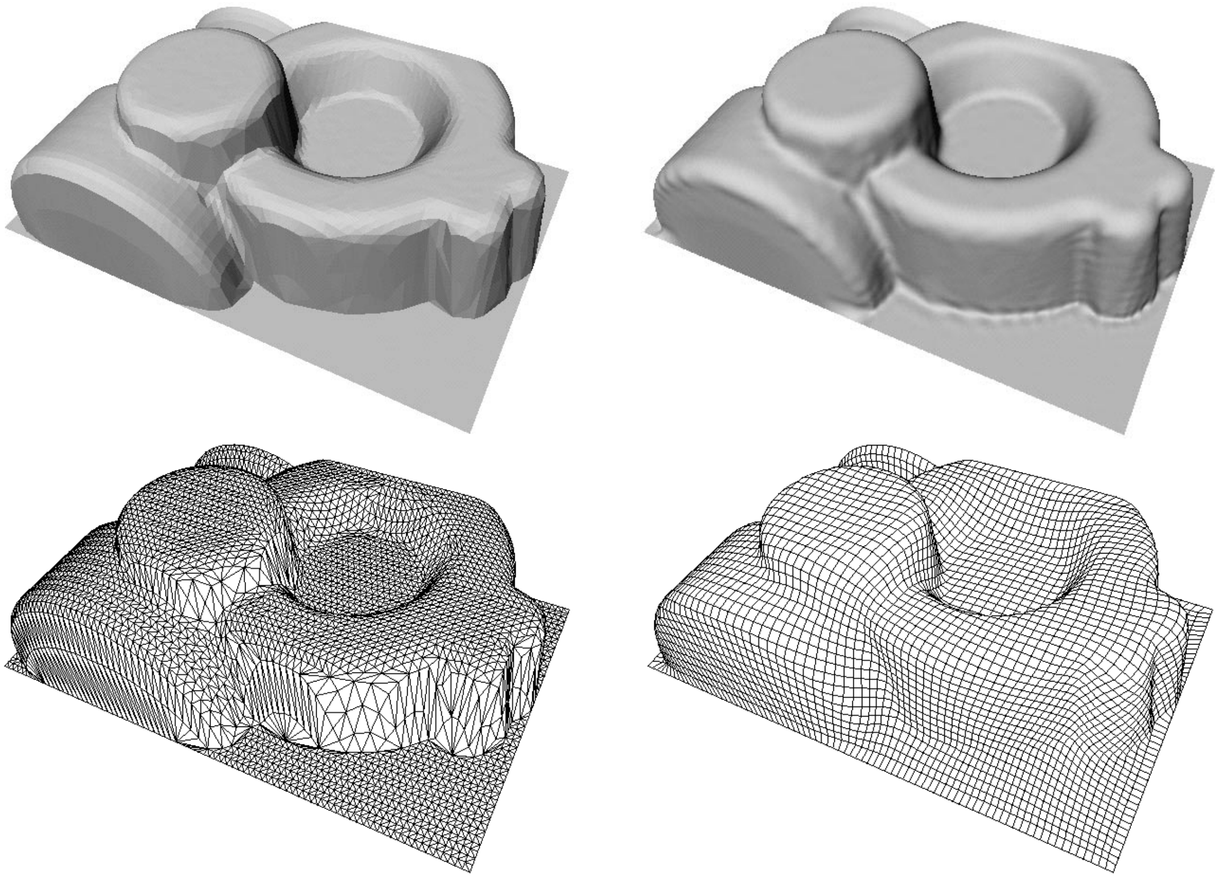

Figure 8: Surface reconstruction of a technical data set: original mesh $\mathcal{T}_3$ shaded (top left) and as wireframe (bottom left), regular quadrilateral remesh $\mathcal{Q}_3$ (bottom right) and reconstructed tensor product B-spline surface (top right).

should relate to the distance of the vertices that are to be interpolated and because of the second remesh criterion aiming at evenly sized facets we can use uniform knot sequences with quadruple end points for simplicity, i. e. $\mathbf{u} = \{0, 0, 0, 0, 1, 2, \ldots, m-1, m, m, m, m\}$ and $\mathbf{v} = \{0, 0, 0, 0, 1, 2, \ldots, n-1, n, n, n, n\}$. The interpolation problem for the bicubic B-spline surface $S(u, v) = \sum_{i,j} M_i(u) N_j(v) d_{ij}$ leads to the $(m + 1)(n + 1)$ *interpolation conditions* $S(i, j) = P_{ij}$. In order to obtain a uniquely solvable problem we have to add further conditions. We have chosen the *natural end conditions*, i. e. setting all second derivatives along the border and the fourth cross derivatives at the corners to zero, but other choices are conceivable. The conditions now add up to $(m+3)(n+3)$, matching exactly the number of unknown control points d_{ij}, which can be determined by solving $m+3$ linear systems with the same $(n+3)\times(n+3)$ matrix and $n+3$ systems of order $(m + 3) \times (m + 3)$. Due to the *local support* property of the B-spline basis functions the two matrices are tridiagonal so that the linear problems can each be solved in linear time. Assuming $m = n$, this approach amounts to $\mathcal{O}(m^2)$ computations in contrast to the $\mathcal{O}(m^6)$ operations needed for solving the scattered data approximation problem.

Another advantage of this approximation method is that it does not require any fairing of the resulting surface. We have observed in our examples that whenever the remesh is of a nice shape the interpolating surface somehow inherits this smoothness. This is probably due to the energy minimizing properties of interpolating cubic B-splines. So instead of applying rather complicated fairing functionals to the approximating surface we can consider the discrete and therefore much simpler problem of smoothing the quadrilateral remesh $\mathcal{Q}_3$.

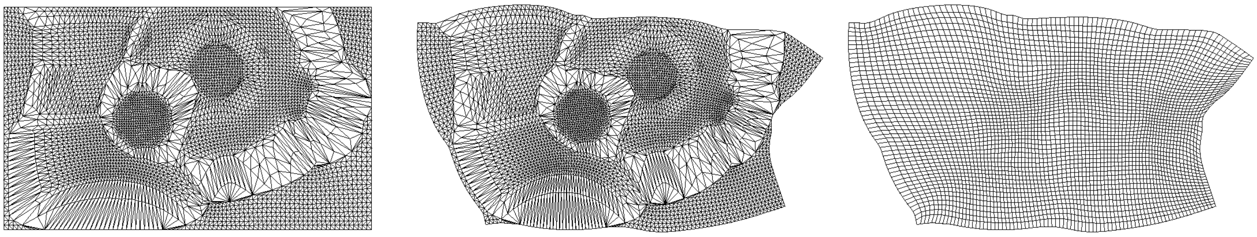

Figure 9: Shape preserving parametrization (left), most isometric parametrization (center) and two-dimensional remesh (right) of the data set in Fig. 8.

5 Examples

In this section we show and discuss some results of the methods explained above. Fig. 8 shows a technical data set with 4.100 vertices and 7.938 triangles that has been remeshed with a regular quadrilateral mesh with 80×48 facets and reconstructed with a tensor product B-spline surface with 4.233 control points.

In the first step of the reconstruction process we have computed and compared two different parametrizations of the given triangular mesh: the shape preserving and the most isometric parametrization (see Fig. 9). Both parametrizations lead to very similar results but we preferred the latter one, despite the fact that its computation is more expensive. Since the boundary vertices are not fixed, the most isometric parametrization is able to reduce the distortion of the inflation and the deflation function especially near the boundary, as we can see in Fig. 10, where a part of the two corresponding three-dimensional remeshes $\mathcal{Q}_3$ is shown. The inflation function defined by the shape preserving parametrization tends to deform the square faces of the two-dimensional remesh $\mathcal{Q}_2$ to rhombuses while the most isometric parametrization preserves the quadratic shape almost entirely.

In the second step we have performed different variants of the two-dimensional remeshing algorithm. The simplest remesh of the shape preserving parametrization is a regularly spaced rectangular grid, but Fig. 11 shows that the inflation function can modify the size of the faces and generates quite differently sized quadrilaterals in the final remesh $\mathcal{Q}_3$. This effect can be compensated by applying the modified umbrella operator.

Finally, the two-dimensional remesh of the most isometric parametrization (see Fig. 9) was used for generating the three-dimensional remesh in Fig. 8 whose vertices served as interpolation points for the surface reconstruction algorithm.

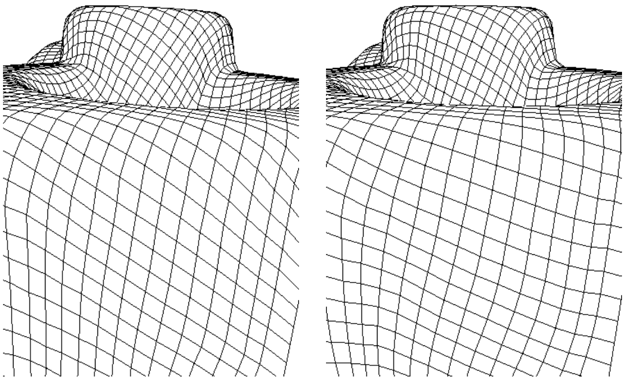

Figure 10: Comparison of the remeshes generated with shape preserving (left) and most isometric parametrization (right).

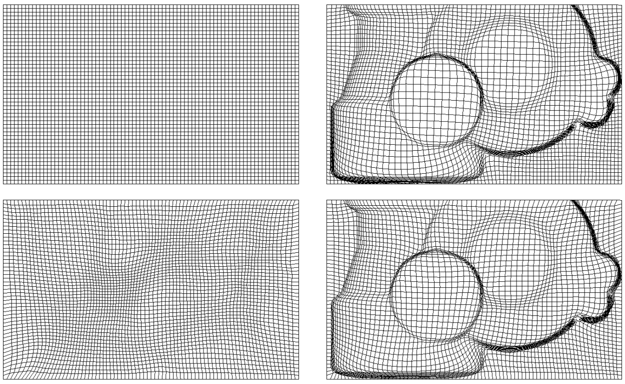

Figure 11: Comparison of the remeshes generated without smoothing (top) and with the modified umbrella operator (bottom).

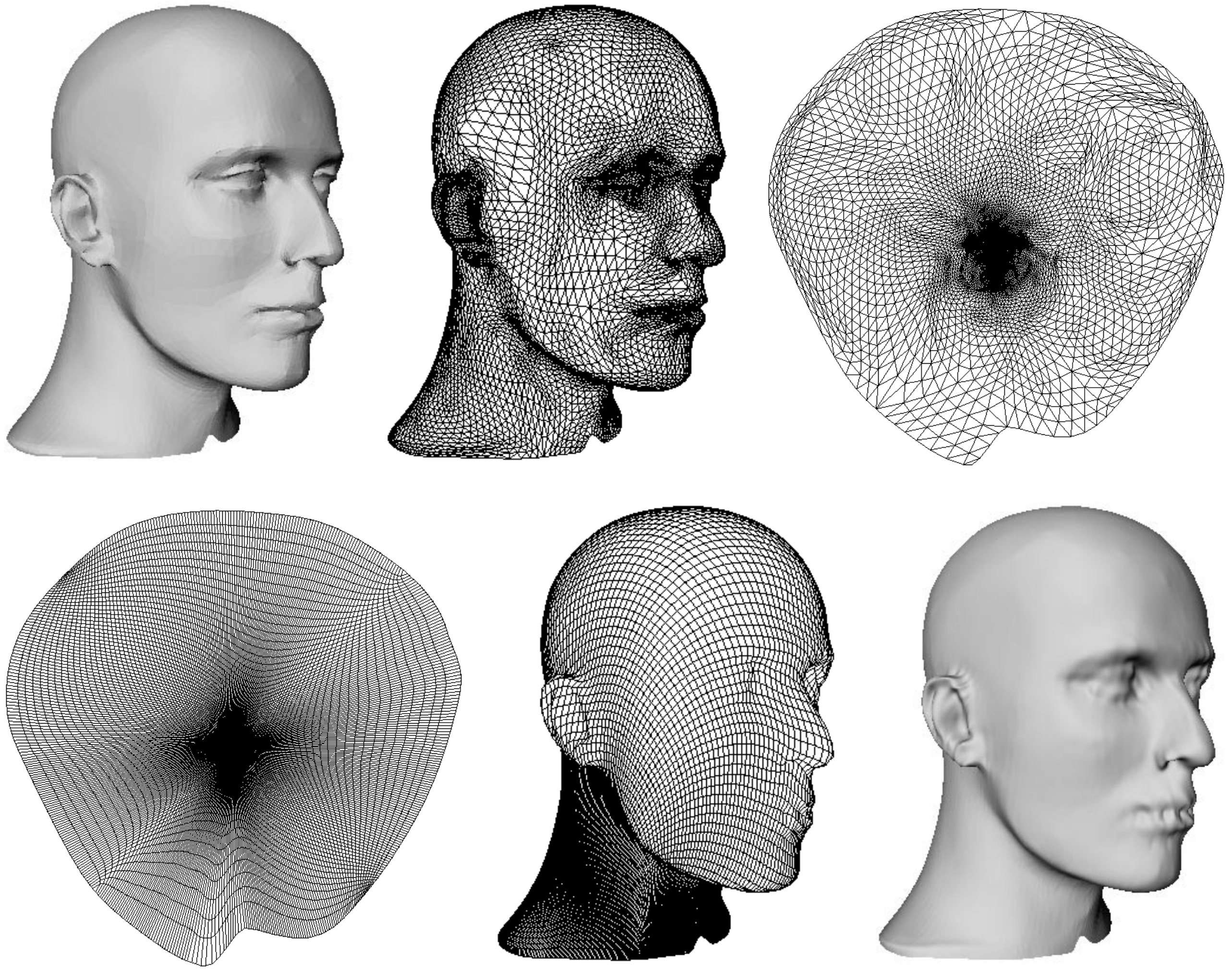

Figure 12: Surface reconstruction of the mannequin head. Top row: original mesh $\mathcal{T}_3$ shaded (left) and as wireframe (center), most isometric parametrization $\mathcal{T}_2$ (right). Bottom row: regular quadrilateral remesh $\mathcal{Q}_2$ (left) and $\mathcal{Q}_3$ (center), reconstructed tensor product B-spline surface (right).

We have also used the most isometric parametrization for remeshing the head data set with 10.883 vertices and 21.680 triangles in Fig. 12 and the face data set with 1.042 vertices and 1.999 triangles in Fig. 13. The remeshes consist of 128×128 and 32×32 facets resp. and the reconstructed surfaces therefore have 17.161 and 1.225 control points resp.

6 Conclusion

We have presented a new method for converting unstructured triangle meshes with boundary into regular quadrilateral meshes. Essential for this remeshing procedure is the construction of a global parametrization of the initial triangle mesh that minimizes geometric distortion. This parametrization enables us to circumvent the three-dimensional remeshing problem and consider a two-dimensional problem instead. Mapping the two-dimensional remesh of the parametrization back into $\mathbb{R}^3$ results in the remesh of the initially given triangle mesh, which can be further used for a simple and efficient surface reconstruction method.

Our future work will aim at improving the surface reconstruction process. Up to now this method does not provide any control of the approximation error, so we will think about an iterative process in which the interpolation

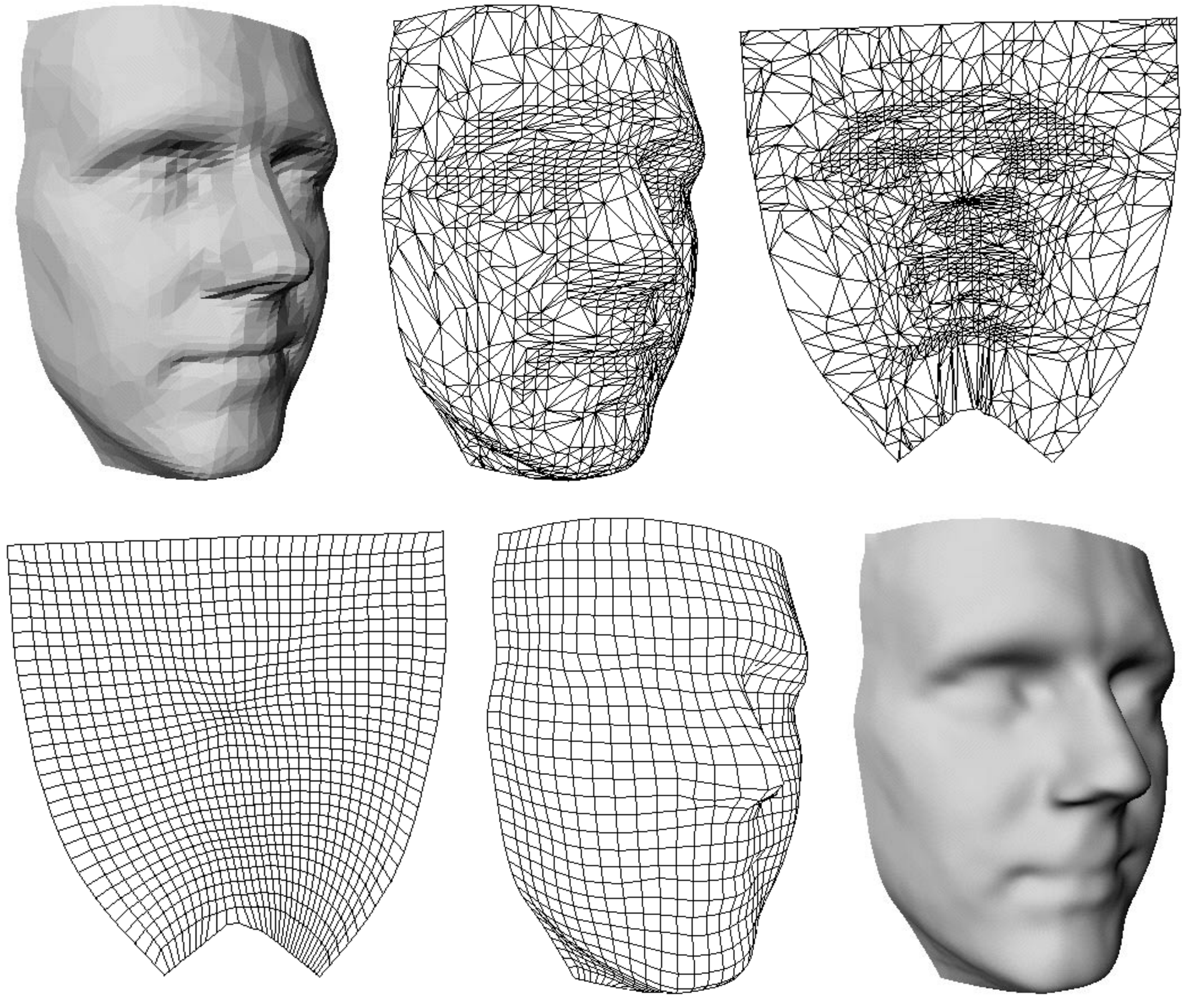

Figure 13: Surface reconstruction of a face data set. Top row: original mesh $\mathcal{T}_3$ shaded (left) and as wireframe (center), most isometric parametrization $\mathcal{T}_2$ (right). Bottom row: regular quadrilateral remesh $\mathcal{Q}_2$ (left) and $\mathcal{Q}_3$ (center), reconstructed tensor product B-spline surface (right).

points are moved in dependency on the approximation error in order to decrease it after solving the interpolation problem again. Another drawback of the current method are the oscillations that occur along crest lines that cross the tensor product structure of the surface (cf. Fig. 8 and Fig. 12). This effect can be avoided by previously detecting such feature lines and aligning the edges of the remesh to them. We will also extend the approach to combine an adaptive remeshing method with hierarchical spline surfaces which will reduce the number of control points of the final surface a lot.

References

[1] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An introduction to splines for use in computer graphics and geometric modeling.* Kaufmann, Los Altos, 1987.

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

[3] U. Dietz, Fair surface reconstruction from point clouds. In *Mathematical Methods for Curves and Surfaces II*, M. Dæhlen, T. Ly-

che, and L. L. Schumaker (eds.), pp. 79–86. Vanderbilt University Press, 1998.

[4] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, Multiresolution analysis of arbitrary meshes. In *ACM Computer Graphics (SIGGRAPH '95 Proceedings)*, pp. 173–182, 1995.

[5] M. Eck, and H. Hoppe, Automatic reconstruction of B-splines surfaces of arbitrary topological type. In *ACM Computer Graphics (SIGGRAPH '96 Proceedings)*, pp. 325–334, 1996.

[6] G. Farin, *Curves and surfaces for computer aided geometric design*. Acadamic Press, Boston, 1993.

[7] M. S. Floater, Parameterization and smooth approximation of surface triangulations. *Computer Aided Geometric Design 14*, pp. 231–250, 1997.

[8] G. Greiner, Variational design and fairing of spline surfaces. In *Computer Graphics Forum (EUROGRAPHICS '94 Proceedings)*, pp. 143–154, 1994.

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

[10] K. Hormann, Fitting free form surfaces. In *Principles of 3D Image Analysis and Synthesis*, B. Girod, G. Greiner, and H. Niemann (eds.), pp. 192–202. Kluwer Academic Publishers, Boston, 2000.

[11] K. Hormann, and G. Greiner, MIPS: an efficient global parametrization method. In *Curve and Surface Design: Saint-Malo 1999*, P.-J. Laurent, P. Sablonnière, and L. L. Schumaker (eds.), pp. 153–162. Vanderbilt University Press, 2000.

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

[13] L. Kobbelt, Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In *Computer Graphics Forum (EUROGRAPHICS '96 Proceedings)*, pp. 409–420, 1996.

[14] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel, Interactive multi-resolution modeling on arbitrary meshes. In *ACM Computer Graphics (SIGGRAPH '98 Proceedings)*, pp. 105–114, 1998.

[15] L. Kobbelt, J. Vorsatz, U. Labsik, and H.-P. Seidel, A shrink wrapping approach to remeshing polygonal surfaces. In *Computer Graphics Forum (EUROGRAPHICS '99 Proceedings)*, pp. 119–130, 1999.

[16] U. Labsik, K. Hormann, and G. Greiner, Using most isometric parametrizations for remeshing polygonal surfaces. In *Geometric Modelling and Processing 2000 Proceedings*, pp. 220–228, 2000.

[17] U. Labsik, L. Kobbelt, R. Schneider, and H.-P. Seidel, Progressive transmission of subdivision surfaces. *Computational Geometry 15*, pp. 25–39, 2000.

[18] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, MAPS: Multiresolution adaptive parameterization of surfaces. In *ACM Computer Graphics (SIGGRAPH '98 Proceedings)*, pp. 95–104, 1998.

[19] M. Lounsbery, T. DeRose, and J. Warren, Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics 16*, pp. 34–73, 1997.

[20] U. Pinkall, and K. Polthier, Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2*, pp. 15–36, 1993.

[21] P. Schröder, and W. Sweldens, Spherical wavelets: Efficiently representing functions on the sphere. In *ACM Computer Graphics (SIGGRAPH '95 Proceedings)*, pp. 161–172, 1995.

[22] D. Zorin, P. Schröder, and W. Sweldens, Interactive multiresolution mesh editing. In *ACM Computer Graphics (SIGGRAPH '97 Proceedings)*, pp. 259–268, 1997.