

Hierarchical Parametrization of Triangulated Surfaces

Kai Hormann* Günther Greiner Swen Campagna

University of Erlangen, Computer Graphics Group
Am Weichselgarten 9, D-91058 Erlangen-Tennenlohe, Germany
Email: hormann@immd9.informatik.uni-erlangen.de

Abstract

The problem of creating global parametrizations of triangulated surfaces is fundamental for many applications in computer-aided geometric design (CAGD). Although several solutions to this problem exist, most of them become considerably slow as the number of triangles increases, because large, ill-conditioned problems have to be solved. We show how the hierarchical representation of a triangulated surface can be exploited to establish a strategy that is similar to multigrid methods and guarantees fast convergence rates.

1 Introduction

The creation of global parametrizations of triangulated surfaces is required by many applications in CAGD, e. g.:

- *surface fitting*, where a discrete set of measured or computed 3D data points is approximated by a smooth parametric surface [6, 7, 8, 11, 13],
- *texture mapping*, where color information is defined in the parameter domain and mapped onto a 3D object to make it look more realistic [1, 4, 14, 15, 21] and
- *remeshing*, where the triangulated surface is converted into a mesh with subdivision connectivity, so that multiresolution analysis and subdivision techniques can be applied [5, 12].

The general problem can be stated as follows: Given a set of data points $P_i \in \mathbb{R}^3$, triangles $T_j = \Delta(P_{j_0}, P_{j_1}, P_{j_2})$ and a parameter domain $\Omega \subset \mathbb{R}^2$, find parameter values $p_i \in \Omega$, one for each data point P_i , such that the topology is preserved, i. e. the triangles in the parameter domain $t_j = \Delta(p_{j_0}, p_{j_1}, p_{j_2})$ must not overlap.

The *local* problem, i. e. parametrizing a set of points W_0, \dots, W_n surrounding a reference point V , can easily be solved by an exponential mapping:

$$\|w_i - v\| = \|W_i - V\|, \quad \gamma_i = \frac{\Gamma_i}{\sum_{j=0}^n \Gamma_j},$$

for $i = 0, \dots, n$, with Γ_i denoting the angle between the edges $\overline{W_i V}$ and $\overline{W_{i+1} V}$ and γ_i defined resp. (cf. Fig. 1). But the *global* problem is more complicated and has been dealt with in several papers before.

Bennis et al. [1] propose a method that is based on differential geometry: isoparametric curves of the surface are mapped onto curves in the parameter domain such that the geodesic curvature at each point is preserved. The parametrization is then extended to both sides of that initial curve until some distortion threshold is reached. But this method, as well as the one presented in [14] by Maillot et al., require the surface to be split into several independent regions and can therefore not be considered as solutions to the *global* problem.

Ma and Kruth [13] project the data points P_i onto a base surface $S : \Omega \rightarrow \mathbb{R}^3$ and the parameter values of the projected points are taken as p_i . But since this method may produce folds in the mapping when the shape of

*The author has been supported by the SFB 603.

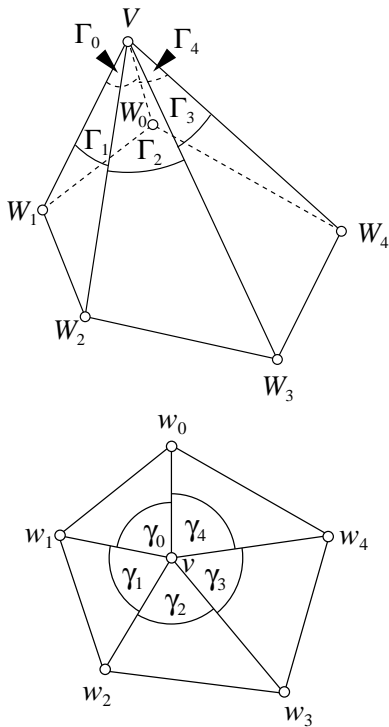


Figure 1: Local exponential mapping.

the base surface is not close enough to that of the triangulated data, it is not suitable for arbitrary data sets.

The approaches in [5, 7, 8, 16] choose an initial parametrization of the *boundary points* and minimize an edge-based energy functional

$$E = \frac{1}{2} \sum_{\{i,j\} \in \text{Edges}} c_{ij} \|p_i - p_j\|^2 \quad (1)$$

to determine the parameter values of the *interior points*. The edge coefficients c_{ij} can be chosen in different ways, but in all cases, minimizing (1) is equivalent to solving a non-singular sparse matrix system, that is (apart from Floater's method [7]) even symmetric positive definite.

But unfortunately, the smaller the ratio of the number of *boundary* and *interior points* is, the more ill-conditioned the matrix becomes, i. e. the condition number grows and the matrix is near to singularity. Therefore, direct methods (like LU decomposition or Cholesky factorization) may produce significant round-off errors and iterative methods (like Gauss-Seidel or Conjugate Gradients) may show intolerably slow convergence rates.

Another parametrization method has been introduced by Hormann and Greiner [10]. They determine the parameter values p_i by minimizing a functional that measures the deformations of the atomic linear functions $f_j : T_j \rightarrow t_j$ that map the surface triangles $T_j = \Delta(P_{j_0}, P_{j_1}, P_{j_2})$ to the corresponding parameter triangles $t_j = \Delta(p_{j_0}, p_{j_1}, p_{j_2})$, thus resulting in a parametrization with minimal distortion.

In contrast to the energy functional (1), their deformation functional is no longer quadratic and cannot be minimized by solving a linear system. Instead, they choose to optimize the functional locally for a randomly chosen parameter value and iterate these local optimization steps. This is comparable to the Gauss-Seidel iteration for solving matrix systems and the convergence rate of that method is just as bad for large data sets.

In this paper we present a method that guarantees fast convergence rates for the *global* parametrization techniques explained above. It can be compared to multigrid methods for matrix systems and exploits the hierarchical representation of a triangulated surface. In Section 2 we briefly explain how such hierarchical representations of meshes can be generated. Section 3 discusses the proposed hierarchical parametrization method and examples are given in Section 4.

2 Hierarchical Meshes

In recent years, the graphics hardware of modern computers has been optimized for displaying and rendering the simplest geometric objects: triangles. Therefore, triangle meshes have gained a lot of attention in the context of displaying geometric models. Modern laser range scanners, iso-surface-generators and many other applications generate complex meshes consisting of many triangles, often much more than a million. But even recent graphic workstations are not capable of interactively displaying such extensive data sets. Furthermore, complex triangle meshes are very expensive to transmit and

store. Many algorithms have been proposed for the decimation of triangle meshes. Surveys of the large variety of techniques can be found in [17, 18, 19].

One way to represent a triangle mesh hierarchically is to store the coarsest resolution and add the finer resolutions as detail coefficients with respect to the next coarser resolution. This may be achieved by using the *edge collapse* and *vertex split* operations (cf. Fig. 2) to modify a mesh. Hoppe has coined the term *Progressive Meshes (PM)* for this sort of hierarchical representation [9].

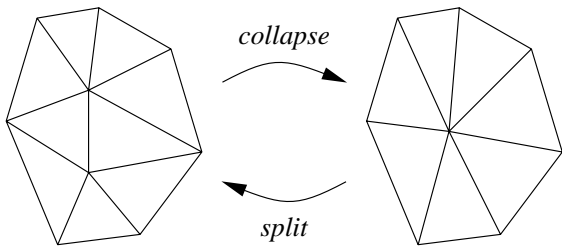


Figure 2: Edge collapse and vertex split.

As shown in Fig. 2, an *edge collapse* reduces the complexity of a mesh by unifying two neighboring vertices and removing the connecting edge and the adjacent triangles. Thus, by applying a sequence of n successive edge collapses, an initial mesh $\hat{M} = M^n$ can be reduced to a coarse mesh M^0 (see Fig. 4):

$$\hat{M} = M^n \xrightarrow{\text{collapse}} \dots \xrightarrow{\text{collapse}} M^0.$$

On the other hand, the inverse operation of the edge collapse, the *vertex split*, exactly reconstructs the former constellation and an arbitrary triangle mesh \hat{M} can therefore be represented by a simple mesh M^0 and a sequence of n vertex splits:

$$M^0 \xrightarrow{\text{split}} \dots \xrightarrow{\text{split}} M^n = \hat{M}.$$

The order of edge collapse operations determines the quality of the coarse mesh and many possible PM construction algorithms with varying trade-offs of speed, accuracy and quality have been proposed [2, 9, 20].

3 Hierarchical Parametrization

As explained in Sec. 1, the global parametrization methods tend to perform badly for large data sets due to the slow convergence rates of the minimization algorithms. Nevertheless, if the triangulated surface that is to be parametrized consists of only a few triangles or if good estimates of the optimal parameter values are known, they perform quite well.

Therefore, the basic idea of our hierarchical approach is to generate a PM of the given data set $\hat{M} = M^n$ first and start with the parametrization of the coarsest level M^0 of that PM representation. This will involve only a very limited number of triangles and the result will be a good estimation of the optimal parametrization of the next level M^1 which emerges from the coarse mesh M^0 by performing a vertex split operation. This leads to the following algorithm:

1. Build hierarchy levels M^n, \dots, M^0 .
2. For i from 0 to n
 - (a) parametrize M^i ,
 - (b) insert the triangles of $M^{i+1} \setminus M^i$.

We will now explain the single steps in detail:

Building hierarchy levels. The simplest type of an edge collapse operation is the *half edge collapse*, where a vertex V simply collapses with a neighboring vertex W_0 without changing the position of W_0 (cf. Fig. 3). Since accuracy and quality of the coarser mesh levels are not required by our parametrization algorithm, we decided to use this type of mesh simplification operation as it is the fastest.

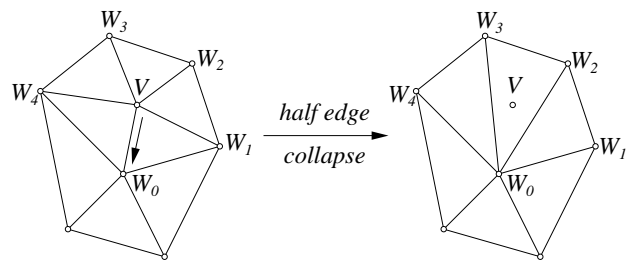


Figure 3: Half edge collapse.

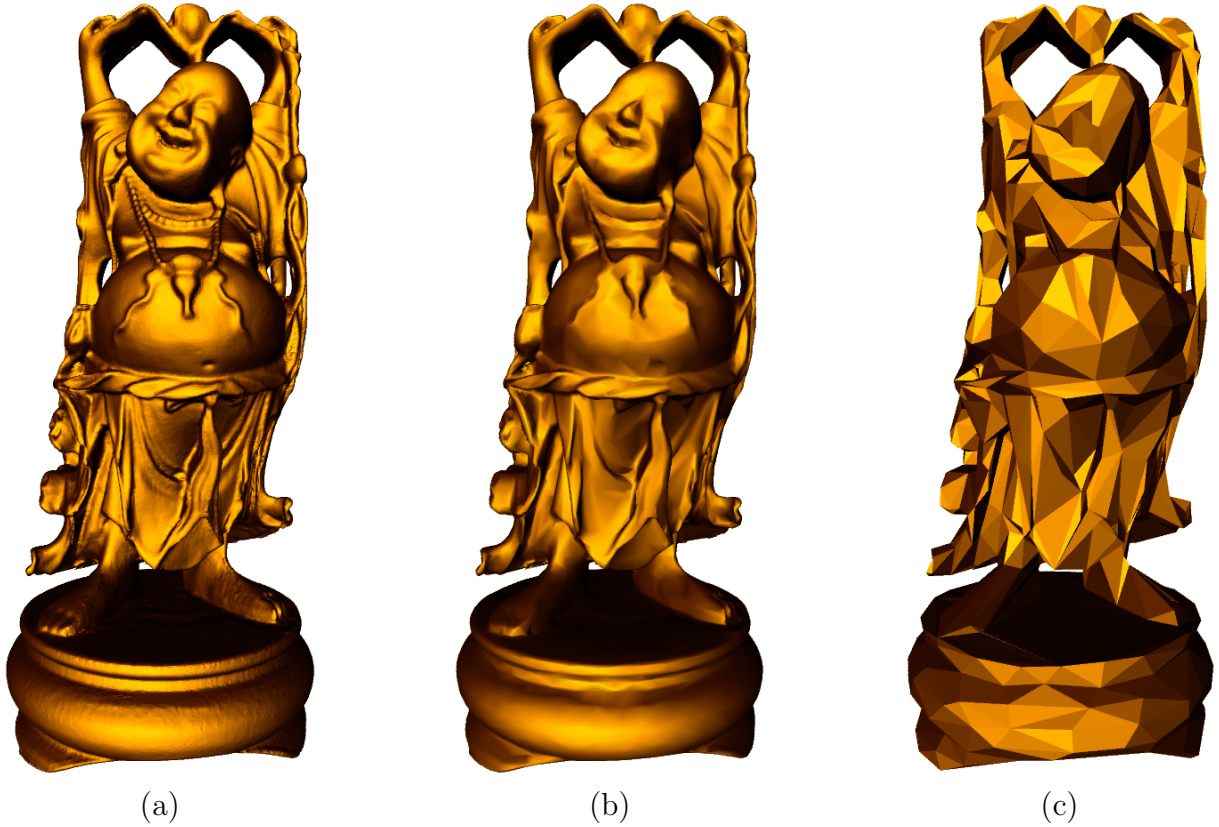


Figure 4: The Stanford buddha. (a) original mesh $\hat{M} = M^{541971}$ with 1,087,716 triangles, (b) simplified mesh M^{13309} with 30,392 triangles and (c) coarse mesh M^0 with 3,774 triangles.

With regard to step 2(b) of our algorithm, we keep track of the barycentric coordinates of the collapsing vertex V with respect to the surrounding triangle as follows: after having applied the exponential mapping to V and the surrounding vertices W_0, \dots, W_n (see Sec. 1), v will lie inside one of the triangles $\Delta_i(w_0, w_i, w_{i+1})$ for some index $1 \leq i \leq n-1$. We store the index i_V of that *parent triangle* and the barycentric coordinates ρ_V, σ_V, τ_V with respect to the vertices w_0, w_{i_V}, w_{i_V+1} , i.e.:

$$v = \rho_V w_0 + \sigma_V w_{i_V} + \tau_V w_{i_V+1},$$

$$\rho_V, \sigma_V, \tau_V \geq 0,$$

$$\rho_V + \sigma_V + \tau_V = 1.$$

This will help us to insert V later in step 2(b).

During our experiments we found out that it is useful to collect several successive edge collapses into m *collapse groups* and consider only those mesh levels \tilde{M}^i that result after having performed these collapse groups,

thus resulting in a coarser mesh hierarchy (see Fig. 5):

$$\hat{M} = \tilde{M}^m \xrightarrow[\text{group}]{\text{collapse}} \dots \xrightarrow[\text{group}]{\text{collapse}} \tilde{M}^0$$

To achieve a uniform elimination of triangles we proceed as follows to get from \tilde{M}^{i+1} to \tilde{M}^i :

- unlock all vertices of \tilde{M}^{i+1}
- repeat
 - choose unlocked vertex V by random
 - collapse V to one of the neighbors W_0
 - lock surrounding vertices W_0, \dots, W_n
- until all remaining vertices are locked

In each of these coarse mesh simplification steps, the number of triangles is reduced by approx. 25%, i.e., $\#\tilde{M}^{i+1}/\#\tilde{M}^i \approx \frac{4}{3}$.

Parametrization. For each hierarchy level i , the optimal parametrization of \tilde{M}^i is determined by using one of the methods mentioned in Sec. 1. Thereby, the previously determined parametrization of \tilde{M}^{i-1} will serve

as a start value for the optimization process. Fig. 5 shows some hierarchy levels and the corresponding optimal parametrizations for the method of Hormann and Greiner [10].

Triangle Insertion. After having found the parametrization of \tilde{M}^i , the vertices of the next finer hierarchy level \tilde{M}^{i+1} that do not belong to \tilde{M}^i must be inserted to proceed with the parametrization of \tilde{M}^{i+1} . For each vertex V , we take the barycentric coordinates ρ_V, σ_V, τ_V that have been stored in step 1 and take $p = \rho_V p_0 + \sigma_V p_{i_V} + \tau_V p_{i_V+1}$ as the initial parameter value of V , where p_i are the parameter values of the vertices W_i .

Note that our locking method of step 1 guarantees the vertices W_i to belong to the coarser level \tilde{M}^i and therefore, the parameter values p_i are well defined.

The use of the barycentric coordinates has proved to yield very good estimates of the final, optimal parameter values, thus forcing the optimization process in step 2(a) to converge very quickly.

4 Examples

We have tested the hierarchical parametrization method for several data sets and parametrization methods and show some results by means of two examples in this section. In both cases, a triangulated surface has been parametrized using the method of Hormann and Greiner [10].

Fig. 6 shows a face data set with 1042 data points and 1999 triangles which has been decomposed into 10 hierarchy levels. The result of the hierarchical parametrization method after 700,000 local optimization steps (i.e. 70,000 per hierarchy level at an average) can be seen in Fig. 6(c). Using the conventional, non-hierarchical minimization technique, the same number of optimization steps leads to the configuration in Fig. 6(d).

Clearly, the hierarchical solution is much closer to the optimal solution than the conventional one, which is confirmed by the values in Table 1. It took approx. 20 sec. on a 250 MHz R10,000 to determine both the hierar-

parametrization	Fig. 6	Fig. 7
optimal	4009.09	1735.87
hierarchical	4010.98	1736.43
conventional	4034.79	1743.85

Table 1: Distortion energy of the parametrizations in Fig. 6 and Fig. 7

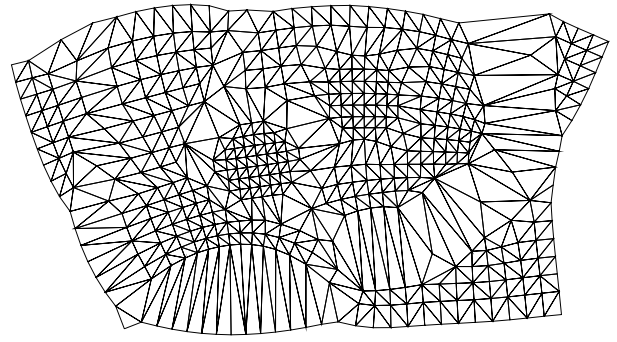
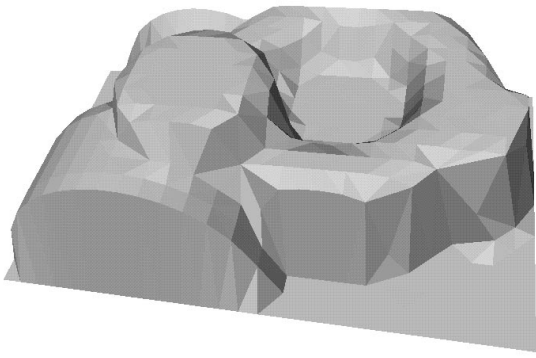
chical and the conventional parametrization. But the conventional method would have to run for many hours to attain a result that were comparable to the result of the hierarchical method, since the convergence rate is intolerably slow.

Another example is given in Fig. 5 and Fig. 7, where a data set with 476 data points and 864 triangles, decomposed into 9 hierarchy levels, has been parametrized. The resulting configurations after 350,000 local optimization steps (approx. 8 sec.) can be seen in Fig. 7 and the values of the energy functional are listed in Table 1.

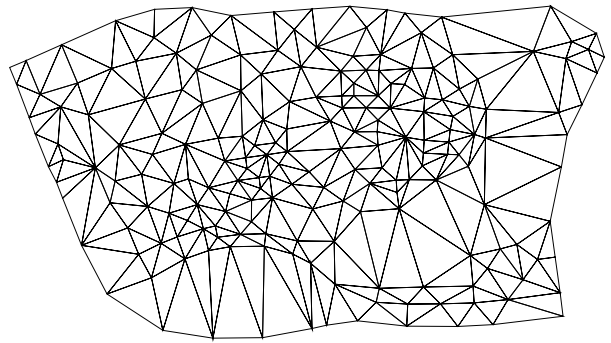
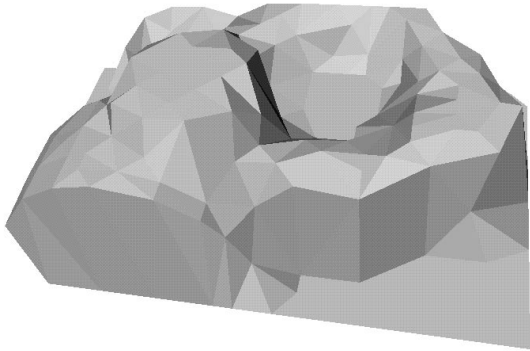
5 Conclusion

In this paper we have shown how the hierarchical representation of a triangulated surface can be used for creating a global parametrization of that surface. This hierarchical approach can be combined with any parametrization technique and guarantees a significant acceleration of the optimization process. The presented idea is especially useful for large data sets and in combination with the non-quadratic parametrization method introduced in [10].

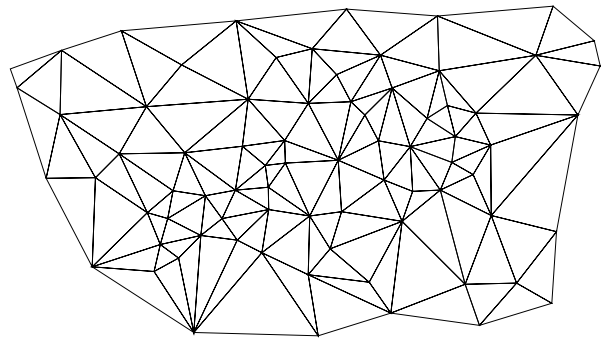
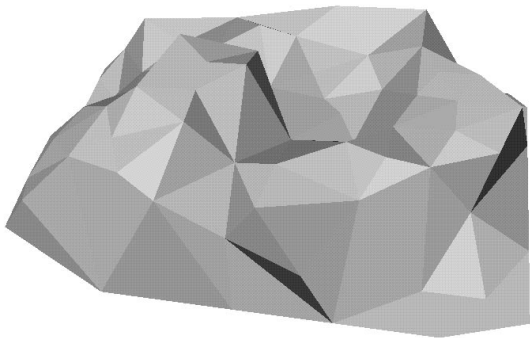
A major insight of this work is not only the efficient solution of the global parametrization problem. Our investigations have shown how powerful hierarchical surface representations like progressive meshes are and how they can be used to accelerate optimization problems dealing with triangulated surfaces. We believe that these hierarchical representations can also be used for efficiently solving e.g. *interpolation* or *approximation problems* and future research will be dedicated to them.



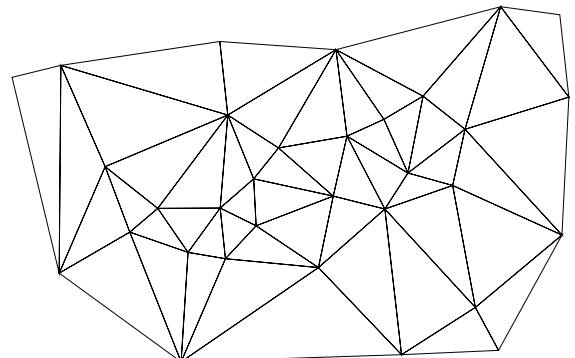
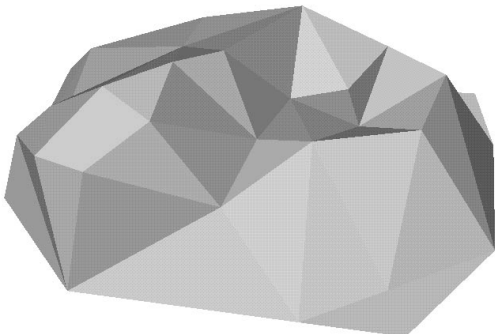
Level $\tilde{M}^9 = \hat{M}$



Level \tilde{M}^6

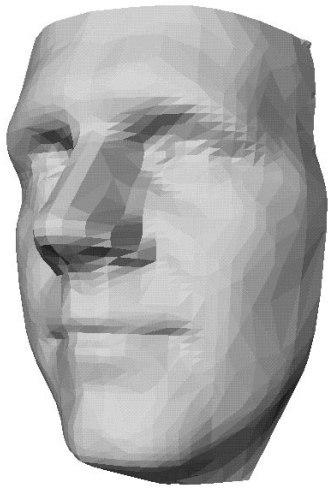


Level \tilde{M}^3

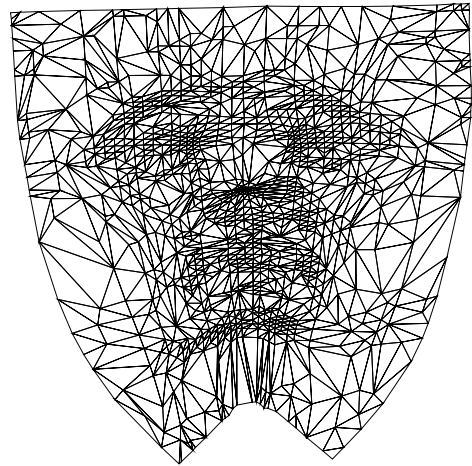


Level \tilde{M}^0

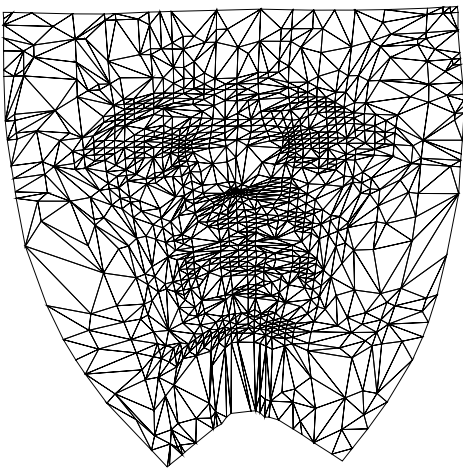
Figure 5: Several coarse hierarchy levels of a triangulated data set (left) and the corresponding optimal parametrizations (right).



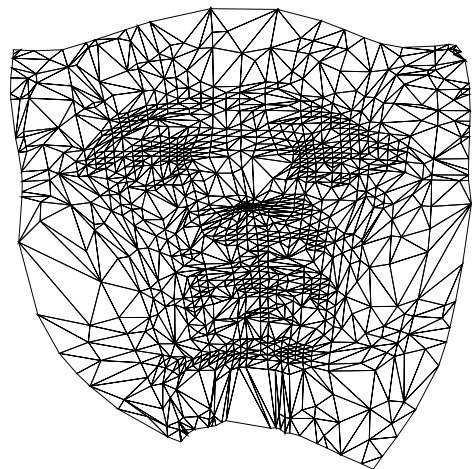
(a)



(b)

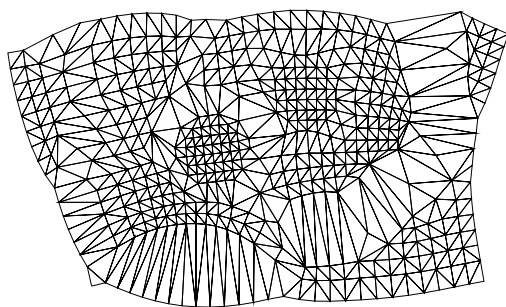


(c)

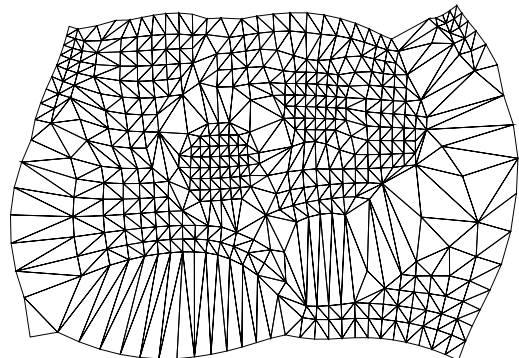


(d)

Figure 6: Face data set (a), optimal parametrization (b) and the result of the hierarchical (c) and the conventional parametrization method (d) after 700,000 local optimization steps.



(a)



(b)

Figure 7: The result of the hierarchical (a) and the conventional parametrization method (b) after 350,000 local optimization steps (data set and optimal parametrization can be seen in the top row of Fig. 5).

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] S. Campagna, L. Kobbelt, and H.-P. Seidel. Efficient decimation of complex triangle meshes. Technical Report 3, Universität Erlangen-Nürnberg, 1998.
- [3] S. Campagna and H.-P. Seidel. Parameterizing meshes with arbitrary topology. In H. Niemann, H.-P. Seidel, and B. Girod, editors, *Image and Multidimensional Digital Signal Processing '98*, pages 287–290. infix, 1998.
- [4] 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.
- [5] 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.
- [6] M. Eck and H. Hoppe. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In *ACM Comp. Graph. (SIGGRAPH '96 Proc.)*, pages 325–334, 1996.
- [7] M. S. Floater. Parameterization and smooth approximation of surface triangulations. *CAGD*, 14:231–250, 1997.
- [8] 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.
- [9] H. Hoppe. Progressive meshes. In *ACM Comp. Graph. (SIGGRAPH '96 Proc.)*, pages 99–108, 1996.
- [10] K. Hormann and G. Greiner. MIPS: an efficient global parametrization method. Technical Report 27, Universität Erlangen-Nürnberg, 1998.
- [11] V. Krishnamurthy and M. Levoy. Automatic reconstruction of B-spline surfaces of arbitrary topological type. In *ACM Comp. Graph. (SIGGRAPH '96 Proc.)*, pages 313–324, 1996.
- [12] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. MAPS: Multiresolution adaptive parameterization of surfaces. In *ACM Comp. Graph. (SIGGRAPH '98 Proc.)*, pages 95–104, 1998.
- [13] 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.
- [14] J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In *ACM Comp. Graph. (SIGGRAPH '93 Proc.)*, pages 27–34, 1993.
- [15] M. Maruya. Generating a texture map from object-surface texture data. In *Comp. Graph. Forum (EUROGRAPHICS '95 Proc.)*, volume 14, pages C397–C405, 1995.
- [16] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.
- [17] E. Puppo and R. Scopigno. Simplification, LOD and multiresolution principles and applications. In *Tutorial Eurographics '97*, 1997.
- [18] J. Rossignac. Simplification and compression of 3D scenes. In *Tutorial Eurographics '97*, 1997.
- [19] W. Schroeder. Polygon reduction techniques. In *IEEE Visualization '95. Advanced Techniques for Scientific Visualization, Section 1.*, 1995.
- [20] W. Schroeder, J. Zarge, and W. Lorensen. Decimation of triangle meshes. In *ACM Comp. Graph. (SIGGRAPH '92 Proc.)*, pages 65–70, 1992.
- [21] M. Soucy, G. Godin, and M. Rioux. A texture-mapping approach for the compression of colored 3D triangulations. *The Visual Computer*, 12:503–514, 1996.