I N F O R M A T I K

Convex Boundary Angle Based Flattening

Rhaleb Zayer Christian Rössl
Hans-Peter Seidel

MPI–I–2003–04-003 April 2003

FORSCHUNGSBERICHT RESEARCH REPORT

M A X - P L A N C K - I N S T I T U T
F Ü R
I N F O R M A T I K

Stuhlsatzenhausweg 85 66123 Saarbrücken Germany

Authors' Addresses

Rhaleb Zayer, Christian Rössl, Hans-Peter Seidel
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
{zayer,roessl,hpseidel}@mpi-sb.mpg.de

Abstract

Angle Based Flattening is a robust parameterization method that finds a quasi-conformal mapping by solving a non-linear optimization problem. We take advantage of a characterization of convex planar drawings of triconnected graphs to introduce new boundary constraints. This prevents boundary intersections and avoids post-processing of the parameterized mesh. We present a simple transformation to effectively relax the constrained minimization problem, which improves the convergence of the optimization method. As a natural extension, we discuss the construction of Delaunay flat meshes. This may further enhance the quality of the resulting parameterization.

Keywords

triangular meshes, parameterization, angle based flattening, constrained optimization

Convex Boundary Angle Based Flattening

Rhaleb Zayer, Christian Rössl and Hans-Peter Seidel

Max-Planck-Institut für Informatik, Saarbrücken, Germany

Abstract

Angle Based Flattening is a robust parameterization method that finds a quasi-conformal mapping by solving a non-linear optimization problem. We take advantage of a characterization of convex planar drawings of triconnected graphs to introduce new boundary constraints. This prevents boundary intersections and avoids post-processing of the parameterized mesh. We present a simple transformation to effectively relax the constrained minimization problem, which improves the convergence of the optimization method. As a natural extension, we discuss the construction of Delaunay flat meshes. This may further enhance the quality of the resulting parameterization.

1. Introduction

Surface parameterization is a fundamental problem in computer graphics. Consider a topologically disk-like surface patch. Then the goal is to find a bijective mapping from a parameter domain to the surface, that fulfills certain quality constraints. For a triangulated surface this is a piecewise linear mapping between the original and an isomorphic planar mesh. So intuitively, we can think of the parameterization as *flattening* the original surface to a valid planar configuration, i.e. one without foldovers or self-intersections. In addition to the validity, quality constraints that minimize the distortion induced by flattening, should be taken into consideration as well.

The importance of the problem makes surface parameterization a very active field of research. Numerous approaches have been proposed so far, inspired by results from different areas of research. Tutte[21] starts from graph theory and uses barycentric maps for embedding a planar graph. Floater's shape-preserving weights[7] improve the conformality of the mapping, while Eck et al.[6] use discrete harmonic maps to minimize angular distortion. All the above methods require a predefined convex boundary in the parameter domain. Hormann and Greiner construct a most-isometric parameterization[12] by minimizing a non-linear deformation functional without need to fix the boundary.Desbrun et al [4] and Levy et al[13] achieve quasi-conformal mappings with an evolving boundary by solving linear systems. Other recent approaches apply multi-dimensional scaling[22] or an iterative algorithm that locally flattens the triangulation until a prescribed distortion bound is reached[20].

Regarding quasi-conformal parameterizations[4, 6, 11, 13, 16], i.e. angular distortion is to be minimized, it seems natural to formulate the problem in terms of interior angles of the mesh. This leads to *Angle Based Flattening*[18]. A characterization of drawings of planar graphs in terms of angles provides conditions for the validity of the flat mesh, in spirit similar to Tutte's barycentric embedding[21]. Minimizing a functional that punishes angular distortion under these constraints will yield a valid parameterization. This algorithm is robust and usually converges after only a few iterations as the original angles provide a good initial guess to the final solution. Although this method strictly falls into the class of non-linear algorithms like e.g. [12, 17], it seems to be a good compromise between linear and non-linear optimization as most conditions are in fact only linear.

2. Contribution

In this paper we focus on *angle based flattening* (ABF) and propose modifications on the original ABF parameterization method by Sheffer and de Sturler[18] that eliminate the need for post-processing, relax the optimization problem, and construct Delaunay flat meshes.

The ABF algorithm constructs a parameterization by minimizing the angular distortion of a planar mesh w.r.t the angles of the original mesh in 3-space. A set of linear and non-linear equality constraints on the planar angles guarantees the validity of the parameterization. These constraints however, do not prevent the boundary from self-intersection. Hence a post-processing of the flat mesh is needed to handle edge crossings at the boundary. Each post-processing step first identifies the nodes causing intersections in the flat mesh. And in order to avoid intersections, it then adds constraints on the local configurations and recomputes the flat mesh as a solution of the updated system. This post-processing algorithm is iterated until no more intersections are found.

We propose a different approach that eliminates boundary intersections in the first place. This is achieved due to the introduction of new inequality constraints that guarantee the validity of the flattened mesh without the cost for iterative post-processing while the imposed inequalities can be handled efficiently.

The constrained optimization problem arising in the ABF algorithm is solved using Lagrange multipliers in combination with the Newton method. In every Newton iteration, a system of equations is solved by using an iterative solver, nameley GMRES or BiCGStab. The convergence of the resulting systems can be improved by the use of sophisticated preconditioning as proposed by Liesen et al.[14].

We show how similar results can be achieved by a simple yet effective transformation of the problem that relaxes the non-linear equality constraints. In fact, The hessian becomes diagonal and the sparsity of the system becomes independant of the valences of the vertices of the input mesh. Since, the system of equations is symmetric we propose to use the more appropriate symmetric numerical solvers instead of the non-symmetric ones mentioned above.

The characterization of the ABF problem leads to a natural extension that guarantees the flattened meshes to be a Delaunay triangulation. We make use of a new condition to improve the quality of the flattened meshes by enforcing the Delaunay property without change of connectivity.

3. Conventions

Throughout the paper, we try to restrict ourselves to the essential amount of formalism only, where the following notations are used:

- N is the total number of interior mesh angles.
- α_i^* $(i = 1,\dots,N)$ denote the angles of the *original* mesh,
- α_i are the corresponding angles of the *flat* mesh. As these are the variables of the optimization problem, then in this context, the more usual notation x_i is used as an alternative.
- v denotes the central vertex in a centered drawing of *wheel*, i.e. of its 1-neighborhood. d is the number of direct neighbors of v or its *valence*. α_j $(j = 1,\dots,d)$ refer to the angles at v, while β_j and γ_j denote the opposite left and right angles of a face with central angle a_j, respectively. All faces are oriented counter-clockwise.
- Variables and functions without subscripts may refer to multivariate vectors as explained by the context.

4. Characterization of drawings of planar graphs

The existence of a planar straight line drawing of angular graphs is a well-studied problem in graph theory [5, 9]. To our knowledge, Di Battista and Vismara were the first to provide a characterization of the convex planar straight line drawing of a tri-connected graph for a given set of positive angles[5]. Their minimal constraints for the planarity of the graph require that the following conditions hold:

- *Vertex consistency*
  For each internal vertex v, with central angles $\alpha_1,\dots,\alpha_d$:
$$\sum_{i=1}^{d} \alpha_i - 2\pi = 0 \qquad (1)$$
- *Triangle consistency*
  For each triangular face with angles α, β, γ the face consistency:
$$\alpha + \beta + \gamma - \pi = 0 \qquad (2)$$
- *Wheel consistency*
  For each internal vertex v with left angles β_1,..,β_d and right angles $\gamma_1,\dots,\gamma_d$:
$$\prod_{i=1}^{d} \frac{\sin(\beta_i)}{\sin(\gamma_i)} = 1 \qquad (3)$$
- *Convex external face condition*
  For each external vertex v, with internal angles $\alpha_1,\dots,\alpha_d$:
$$\sum_{i=1}^{d} \alpha_i \le \pi \qquad (4)$$

Sheffer and de Sturler[18] found similar constraints on internal vertices independently. These conditions (1),(2),(3) guarantee the centered embedding of internal vertices of wheels without overlapping of interior edges.

Condition (4) guarantees the *convexity of the boundary* and hence prevents boundary overlapping. Note that the inequality (4) prevents local and global self-intersection simultaneously. So it not only prevents adjacent boundary edges from overlapping, but it also guarantees that the boundary loop as a whole does not cross itself. For the local configuration it would in fact be sufficient to require the following weakened condition to hold

Adjacent boundary consistency
For each external vertex v, with internal angles $\alpha_1,\dots,\alpha_d$:
$$\sum_{i=1}^{d} \alpha_i \le 2\pi. \qquad (5)$$

This prevents adjacent boundary triangles from crossing each other. Fig. 1 illustrates the situation. However, condition (5) is not strong enough to globally enforce a valid mesh with no boundary intersections as shown in Fig. 2 (center).

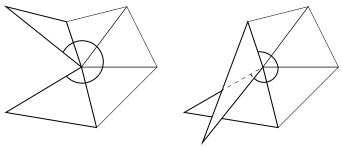


Figure 1: *Valid (left) and invalid local wheel configuration. The adjacent boundary edges intersect as condition (5) and hence (4) are violated (right).*

Intuitively, we can think of condition (4) as applying a force on the boundary edges to straighten them and to stretch the parts of the boundary that cause fold overs in the flat mesh. Fig. 2 (right) illustrates this effect.

We take advantage of this fact in order to avoid an iterative post-processing and thus have better control over the convergence of the constrained optimization problem. In the next sections we show how this problem with the additional inequalities included can be solved efficiently.

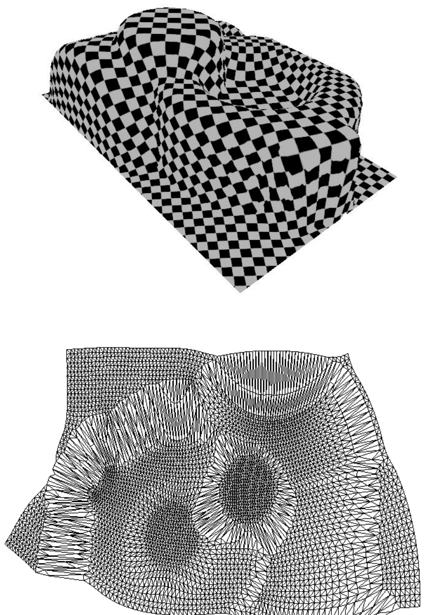


Figure 3: *Angle based flattening ($t = 2$) of a technical data set consisting of a regular triangulation of 4100 vertices. The model (top) is shown with the parameterization (bottom) applied as texture map. The runtime of the algorithm was 25 minutes (1.8 GHz Xeon).*

5. Constrained optimization problem

The flattening procedure that is applied for parameterization minimizes the objective function

$$f(x) = \sum_{i=1}^{N} w_i (x_i - a_i)^2$$

with the weights $w_i = \frac{1}{a_i^2}$. The variables a_i represent the *optimal angle* of the flat mesh, which is according to[18]

$$a_i = \begin{cases} \alpha_i^* \frac{2\pi}{\sum_{i=1}^{d} \alpha_i^*} & \text{around an interior vertex} \\ \alpha_i^* & \text{around a boundary vertex} \end{cases}$$

This function measures the distortion of the flat mesh w.r.t the initial mesh. Our goal is to minimize this function in order to produce a quasi-conformal parameterization. The validity of the flat mesh is guaranteed by respecting the constraints from the previous section.

At this point we apply a simple transformation on part of the problem. Since the angles are strictly positive we can safely rewrite condition (3) as

$$\sum_{i=1}^{d} \log(\sin\beta_i) - \log(\sin\gamma_i) = 0. \qquad (6)$$

The virtue of this modification will be made clear in the following section. We can now formulate the optimization problem as

$$\begin{array}{lrcl} \text{minimize} & f(x) & & \\ \text{subject to} & h(x) & = & 0 \\ & g(x) & \leq & 0, \end{array} \qquad (7)$$

where g and h are multivariate functions of the equality (1),(2),(6) and the inequality constraints (4) or (5), respectively.

6. Solving the optimization problem

Large constrained optimization systems of the form (7) are still open problems in the field of non-linear optimization[3]. The adequacy of a method for their solution depends very much on the properties of the objective function as well as on the constraints.

We apply the so called active set approach, a variant of Newton-like methods, that proceeds by replacing inequalities by equalities which are easier to handle.

The active set is defined as the set of indices for which the inequality constraint (4) is active. Formally

$$A(x,\mu) = \{i | g_i \geq -\frac{\mu_i}{c}, i = 1, ..., r\}$$

where μ_i is the Lagrange multiplier associated with g_i, and c a fixed positive scalar.

The active set approach converts inequality constraints to

(a) (b) (c)

Figure 2: *Flattening an α-shaped mode: (a) Original mesh. (b) The ABF flattened mesh self-intersects without post-processing ($t \geq 2$). (c) Result from convex boundary ABF ($t = 1$). (The views are scaled differently.)*

equality constraints by altering the Lagrange multipliers associated with them. If a constraint does not figure in the active set, its associated multipliers are set to zero. Otherwise it is treated as an equality constraint. The numerical advantage of this method is that as the iterates get closer to the solution, the active set becomes more and more stable. A detailed description of the active set method can be found in [1].

In every Newton iteration the following system is solved

$$\begin{bmatrix} \nabla^2_{xx}L & J_h^T & J_g^T \\ J_h & 0 & 0 \\ J_g & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\mu_h \\ \Delta\mu_g \end{bmatrix} = - \begin{bmatrix} \nabla_x L \\ h \\ g \end{bmatrix} \qquad (8)$$

where the Lagrangian L is given by

$$L = f(x) + \mu_h^T h(x) + \mu_g^T g(x).$$

Now we can point out the advantage of using the *modified wheel condition* (6) instead of the original term (3). In the classic ABF algorithm, the computation of the Hessian matrix $\nabla^2_{xx}L$ involves finding the second derivatives of the products involved in condition (3). In fact, the resulting matrix is sparse, but it still contains a considerable number of non-zero elements (cf. [14]). In contrast, the modified wheel condition results in the diagonal Hessian matrix.

$$\nabla^2_{xx}L = \text{diag}(2w_i^2 + m_i \frac{-1}{\sin^2(x_i)})$$

where m_i is the linear combination of the Lagrange multipliers involved with x_i in condition (6).

The system matrix is symmetric although not necessarily definite, with the addiational advantage of having a diagonal Hessian. Fig. 4 illustrates the structure of a typical system matrix. We can exploit this structure by using adequate solvers like the iterative solvers MINRES or SYMMLQ both developed by Paige and Saunders[15] instead of the non-symmetric GMRES and BiCGStab that are applied in[14] with sophisticated preconditioning. We chose MINRES in our experiments because of its efficient error minimization

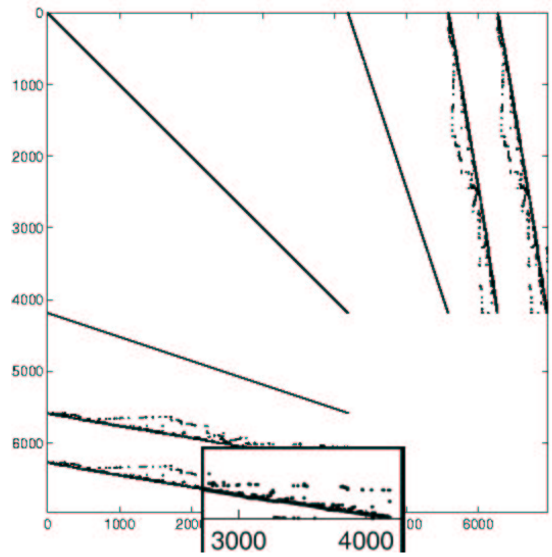


Figure 4: *System matrix (cf. equation (8)) generated from the* body *model with our* modified wheel condition *(6). Notice the diagonal Hessian and the overall symmetry. The contribution of the active set can be seen at the bottom line of the zoomed region and symmetrically at the middle-right bounds of the matrix. The number of non-zero elements of the sparse matrix depends largely on the average valence or the regularity of the mesh.*

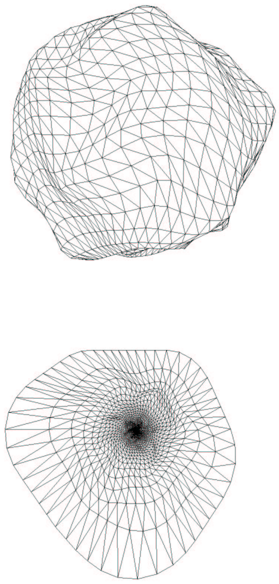


Figure 5: *Flattening of the* clumpy *model. Top: original mesh. Bottom: flattened mesh. Notice the convexity of the boundary of the flattened mesh.*

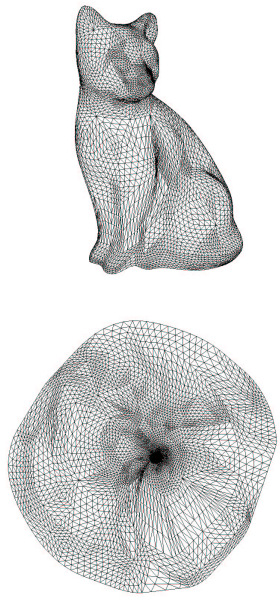


Figure 7: *Flattening of the* cat *model. Top: Original mesh, the cat's bottom is open. Bottom: Flattened mesh.*

properties[10]. Since the MINRES algorithm does not suffer from break downs, our method cannot stagnate, and it converges to the solution of the parameterization problem if it exists.

Note that the initial guess for the x is the vector of the optimal angles a. In consequence, at every Newton iteration the solution stays within the positive domain. However, in order to guarantee that our algorithm does not step into negative domain, we have to apply a similar technique as in [18] that rejects too small iterates and appends increased weights to the corresponding angles. However, our experiments with different meshes show that we hardly ever run into this situation and have to update degeneracies.

7. Generating a Delaunay flat mesh

Many applications benefit from nicely shaped or "round" triangles in contrast to long, thin or almost degenerated triangles. For this reason, a Delaunay triangulation may be preferred as the result of flattening. This means that no vertex falls within the circumcircle of any triangle of the triangulation (cf. Fig. 6). It turns out that the construction of a Delaunay flat mesh can easily be incorporated into our algorithm merely by adding another condition.

This is due to the fact, that for a flat mesh the Delaunay property can be expressed as a condition on angles: Given two triangles sharing a common edge with opposite angles α_1 and α_2, the Delaunay criterion can be expressed as

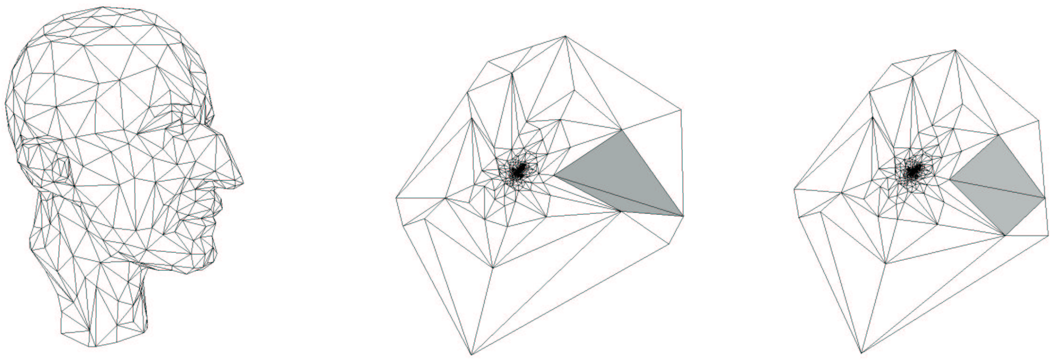
$$\alpha_1 + \alpha_2 \leq \pi. \tag{9}$$

This is a result from computational geometry[2, 8]. Di Battista et al.[5] propose to use this criterion in addition to (1)-(4) in order to generate a Delaunay triangulation. Clearly, the Delaunay property does not come for free. The angular distortion will suffer, and the new set of additional inequality constraints increases the number of variables in the optimization problem by the number of internal edges.

The existence of a unique Delaunay triangulation for a given set of of points is a classic result from algorithmic geometry[2]. In such case, the Delaunay triangulation is constructed by applying edge-flips or constructing the appropriate connectivity. However, in our case we are constrained by the given connectivity and by the geometry of the input mesh. Hence the existence of a Delaunay flat triangulation is not guaranteed. Still, it may be worthwhile to add the constraints (9) after an initial solution has been computed. In general, we observed better results in terms of convergence with this sort of Delaunay post-processing compared to including the Delaunay conditions from the beginning. Fig. 6 shows an example of a Delaunay flat mesh.

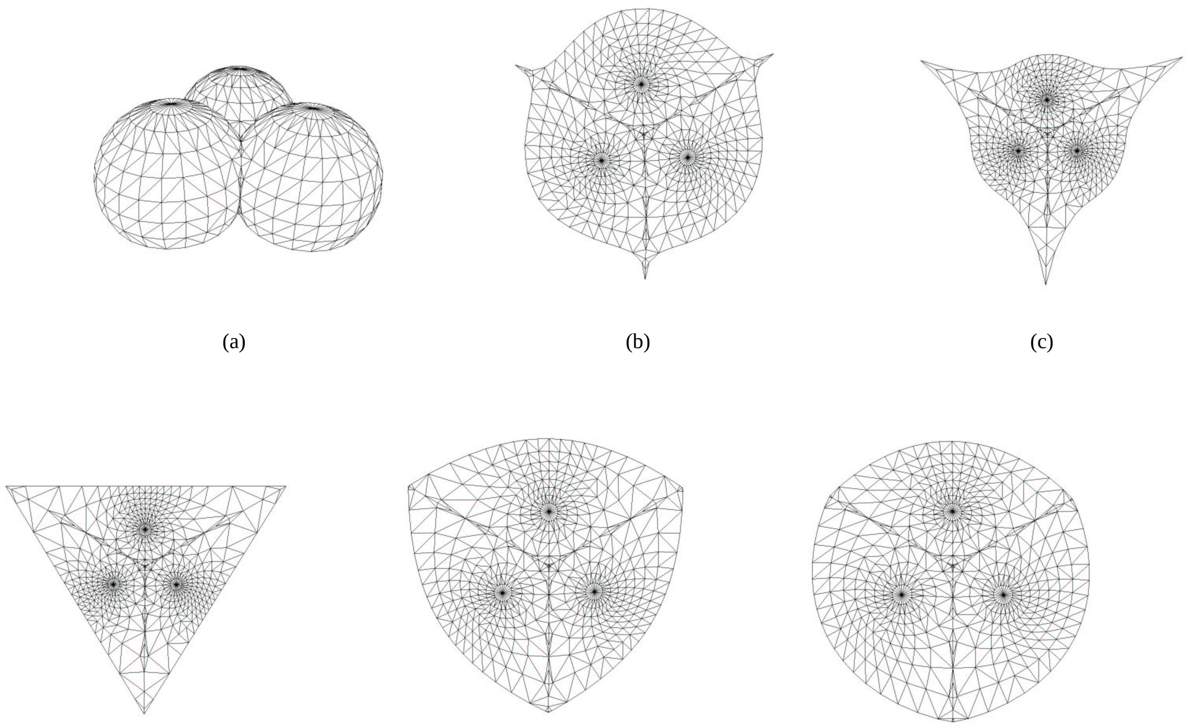
8. Results and Discussion

We applied our convex boundary ABF algorithm to a set of different triangle meshes. Table 1 summarizes the numeri-



(a) (b) (c)

Figure 6: *Delaunay parameterized mesh. (a) Max-Planck model. (b) Parameterization with convex boundary, the highlighted configuration is one example for disregarding the local Delaunay property. (c) Delaunay parameterization.*



(a) (b) (c)

(d) (e) (f)

Figure 8: *Effect of the boundary control coefficient t on the* 3-balls *model. (a) Original mesh. (b) Flat mesh for $t \geq 2$ *ABF). (c) $t = 1.05$. (d) $t = 1$ (convex boundary ABF). (e) $t = .98$. (f) $t = 0.968$*

cal results of our method, all timings were measured on a 1.8 GHz Intel Xeon CPU. The parameterization time depends on the number of triangles and on the geometry as well as on the connectivity of the input mesh. For consistency with the original ABF we use the same metrics as in [19] to measure angular and length distortion.

As condition (4) imposes a strong constraint on the boundary, we propose to relax it by multiplying the left hand side by a positive scalar t, formally

$$\sum_{i=1}^{d} \alpha_i \leq t\pi$$

The scalar t can be interpreted as *boundary control coefficient* that steers the convexity of the boundary. It is clear that for $t > 1$ we cannot rule out global boundary intersections and hence require ABF-like post-processing. Still, this is a way to control the probability of self-crossings and to reduce the amount of post-processing. We experimented with different values for t, and summarize the following interpretations:

- $t \geq 2$ results in the classic ABF method. No adjacent edge overlapping or boundary self crossing is taken into consideration.
- $t < 2$ prevents adjacent edges overlapping, but does not necessarily prevent global self-intersections of the boundary loop. We experienced such cases only for "boundary-heavy" (w.r.t. ratio boundary to inner vertices, e.g. Fig. 2) surfaces with non-trivial geometry (e.g. twists).
- $t = 1$ globally prevents the boundary loop from self-intersection for any valid input mesh, note that this suffices to induce a convex boundary .
- $t < 1$ forces the boundary to become even “more convex”.

Fig. 8 and 2 illustrate the behavior of the boundary for different values of t. Table 2 shows numerical results for varying t and typical meshes.

Note that depending on the input mesh, convergence and validity of the flat mesh cannot be achieved for all values of t. Our results suggest, that for each mesh there might be an optimal value for t that produces best results in terms of angular and length distortion. This could be an interesting subject of further investigation.

9. Conclusions and future work

We presented and discussed several extensions to angle based flattening. With additional inequality constraints we can eliminate global and/or local boundary self-intersections. This leads to a nice interpretation of boundary behaviour through the introduction of the new boundary control coefficient. While its use initially targets on the avoidance of an iterative post-processing.We see potential use for optimizing the parameterization with this coefficient variable.

The arising non-linear constrained optimization problem can be solved efficiently. With a simple and intuitive transformation we can take advantage of a symmetric system matrix, enabling the application of robust iterative solvers. We will experiment with more numerical algorithms, e.g. the sequential quadratic method as a variant of the active set method, to further improve the numerical results.

Our algorithm can be easily adapted to generate Delaunay flat meshes. While such a solution does not necessarily exist, we propose to apply this criterion as a single post-processing step as the Delaunay property may improve the quality of the flat mesh for special applications.

We tested our algorithm on a variety of triangle meshes. The examples show that our implementation can handle moderately sized meshes in reasonable time. In direct analogy to angle based flattening, similar constraints on edge lengths would lead to a formulation of a quasi-isometric parameterization method (such as [12]), which we plan to investigate in the future.

References

1. D.P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods.* Athena Scientific, 1996.
2. J-D. Boissonnat and M. Yvinec. *Algorithmic Geometry* Cambridge University Press, 1998
3. R.H. Byrd, J. Nocedal Active Set and Interior Methods for Nonlinear Optimization *Doc. MATH*, Extra Volume ICM III, 1998, pp. 667–676
4. Mathieu Desbrun, Mark Meyer, and Pierre Alliez. Intrinsic Parameterizations of Triangle Meshes. *Proc. Eurographics 2002*, pp. 209-218
5. G. Di Battista and L. Vismara. Angles of Planar Triangular Graphs. *SIAM Journal on Discrete Mathematics*, 9 (3), 1996, pp. 349–359
6. M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution Analysis of Arbitrary Meshes. *Proc. SIGGRAPH 95*, pp. 173–182
7. M.S. Floater. Parametrization and smooth approximation of surface triangulations. *Comp. Aided Geom. Design*, (14), 1997, pp. 231-250
8. P. Fleischmann. Mesh Generation for Technology CAD in Three Dimensions. Ph.D Thesis, Wien, 1999.
9. A. Garg. New results on drawing angle graphs. *Computational Geometry* (9), (1-2), 1998, pp. 43–82
10. A. Greenbaum. *Iterative Methods for Solving Linear Systems* SIAM, Philadelphia, 1997
11. S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE Transactions*

model	# faces	runtime (sec.)	angular distortion	length distortion
Cat head (small)	257	<1	0.030	0.560
3 balls	1032	8	0.115	0.539
Isis Statue	879	37	0.015	0.279
Nefertiti	562	1	0.0025	0.018
Body	1396	16	0.011	1.688
Cat head (large)	3660	100	0.003	0.371
Max Planck	988	12	0.025	4.698
Clumpy	2340	180	0.0063	2.468
Cat	11168	32 min	0.001	0.943
Hand	3000	215	0.029	4.473

Table 1: *Numerical examples of convex boundary flattened meshes ($t = 1$).*

	$t = 2$		$t = 1.05$		$t = 1$		$t = 0.968$	
model	*ang.*	*len.*	*ang.*	*len.*	*ang.*	*len.*	*ang.*	*len.*
3 balls	0.106	0.108	0.108	0.483	0.115	0.539	0.208	0.116
Cat head	0.03	0.560	0.030	0.560	0.030	0.560	0.030	0.560
Body	0.011	1.660	0.011	1.678	0.011	1.688	0.011	1.702
Nefertiti	0.0016	0.013	0.0021	0.017	0.0025	0.018	0.0029	0.019
Hand	0.029	4.455	0.029	4.455	0.029	4.473	0.03	4.46

Table 2: *Comparison of Angular and length distortions for different values of the boundary control coefficient t. (All the runtimes do not differ noticeably from that in table 1 while no self-inrersections occcur, no post-processing is done.)*

on Visualization and Computer Graphics, 6(2), 2000, pp. 181–189

12. K. Hormann and G. Greiner. MIPS: an efficient global parametrization method. *Curve and Surface Design: Saint-Malo 1999*, 2000, pp. 153–162

13. Bruno Levy, Sylvain Petitjean, Nicolas Ray, and Jerome Maillot. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *Proc. SIGGRAPH 2002*, pp. 362–371

14. J. Liesen, E. de Sturler, A. Sheffer, Y. Aydin, and C. Siefert. Preconditioners for Indefinite Linear Systems Arising in Surface Parameterization. *Proceedings of the 10th International Meshing Round Table*, 2001, pp. 71–81

15. C. Paige and M. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal*, 12, 1975, pp. 617–629

16. U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(15), 1993, pp. 15-36

17. P. Sander, J. Snyder, S. Gortler, and H. Hoppe. Texture mapping progressive meshes. *Proc. SIGGRAPH 2001*, pp. 409–416

18. A. Sheffer and E. de Sturler. Parameterization of Faceted Surfaces for Meshing Using Angle Based Flattening. *Engineering with Computers*, 17 (3), 2001, pp. 326–337

19. A. Sheffer, E. de Sturler. Smoothing an Overlay Grid to Minimize Linear Distortion in Texture Mapping. *ACM Transactions on Graphics*, 21 (4), 2002

20. O. Sorkine, D. Cohen-Or, R. Goldenthal, D. Lischinski. Bounded-distortion piecewise mesh parameterization. *Proc. IEEE Visualization 2002*, pp. 355–362

21. W. T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, 13 (3), 1963, pp. 743–768

22. G. Zigelmann, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multi-dimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8(2), 2002

I N F O R M A T I K

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension
MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects
MPI-I-2003-4–003	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-2-002	M. Jaeger	A Representation Theorem and Applications to Measure Selection and Noninformative Priors
MPI-I-2003-2-001	P. Maier	Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete
MPI-I-2003-1-011	P. Krysta, A. Czumaj, B. Voecking	Selfish Traffic Allocation for Server Farms
MPI-I-2003-1-010	H. Tamaki	A linear time heuristic for the branch-decomposition of planar graphs
MPI-I-2003-1-009	B. Csaba	On the Bollobás – Eldridge conjecture for bipartite graphs
MPI-I-2003-1-008	P. Sanders	Soon to be published
MPI-I-2003-1-007	H. Tamaki	Alternating cycles contribution: a strategy of tour-merging for the traveling salesman problem
MPI-I-2003-1-006	H. Tamaki, M. Dietzfelbinger	On the probability of Rendezvous in Graph
MPI-I-2003-1-005	M. Dietzfelbinger, P. Woelfel	Almost Random Graphs with Simple Hash Functions
MPI-I-2003-1-004	E. Althaus, T. Polzin, S.V. Daneshmand	Improving Linear Programming Approaches for the Steiner Tree Problem
MPI-I-2003-1-003	R. Beier, B. Vcking	Random Knapsack in Expected Polynomial Time
MPI-I-2003-1-002	P. Krysta, P. Sanders, B. Vcking	Scheduling and Traffic Allocation for Tasks with Bounded Splittability
MPI-I-2003-1-001	P. Sanders, R. Dementiev	Asynchronous Parallel Disk Sorting
MPI-I-2002-4-002	F. Drago, W. Martens, K. Myszkowski, H. Seidel	Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference
MPI-I-2002-4-001	M. Goesele, J. Kautz, J. Lang, H.P.A. Lensch, H. Seidel	Tutorial Notes ACM SM 02 A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2002-2-008	W. Charatonik, J. Talbot	Atomic Set Constraints with Projection
MPI-I-2002-2-007	W. Charatonik, H. Ganzinger	Symposium on the Effectiveness of Logic in Computer Science in Honour of Moshe Vardi

MPI-I-2002-1-008	P. Sanders, J.L. Trff	The Factor Algorithm for All-to-all Communication on Clusters of SMP Nodes
MPI-I-2002-1-005	M. Hoefer	Performance of heuristic and approximation algorithms for the uncapacitated facility location problem
MPI-I-2002-1-004	S. Hert, T. Polzin, L. Kettner, G. Schfer	Exp Lab A Tool Set for Computational Experiments
MPI-I-2002-1-003	I. Katriel, P. Sanders, J.L. Trff	A Practical Minimum Scanning Tree Algorithm Using the Cycle Property
MPI-I-2002-1-002	F. Grandoni	Incrementally maintaining the number of l-cliques
MPI-I-2002-1-001	T. Polzin, S. Vahdati	Using (sub)graphs of small width for solving the Steiner problem
MPI-I-2001-4-005	H.P.A. Lensch, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2001-4-004	S.W. Choi, H. Seidel	Linear One-sided Stability of MAT for Weakly Injective Domain
MPI-I-2001-4-003	K. Daubert, W. Heidrich, J. Kautz, J. Dischler, H. Seidel	Efficient Light Transport Using Precomputed Visibility
MPI-I-2001-4-002	H.P.A. Lensch, J. Kautz, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing, Transmission, and Interactive Display of High Quality 3D Models on the Web
MPI-I-2001-4-001	H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, H. Seidel	Image-Based Reconstruction of Spatially Varying Materials
MPI-I-2001-2-006	H. Nivelle, S. Schulz	Proceeding of the Second International Workshop of the Implementation of Logics
MPI-I-2001-2-005	V. Sofronie-Stokkermans	Resolution-based decision procedures for the universal theory of some classes of distributive lattices with operators
MPI-I-2001-2-004	H. de Nivelle	Translation of Resolution Proofs into Higher Order Natural Deduction using Type Theory
MPI-I-2001-2-003	S. Vorobyov	Experiments with Iterative Improvement Algorithms on Completely Unimodel Hypercubes
MPI-I-2001-2-002	P. Maier	A Set-Theoretic Framework for Assume-Guarantee Reasoning
MPI-I-2001-2-001	U. Waldmann	Superposition and Chaining for Totally Ordered Divisible Abelian Groups
MPI-I-2001-1-007	T. Polzin, S. Vahdati	Extending Reduction Techniques for the Steiner Tree Problem: A Combination of Alternative-and Bound-Based Approaches
MPI-I-2001-1-006	T. Polzin, S. Vahdati	Partitioning Techniques for the Steiner Problem
MPI-I-2001-1-005	T. Polzin, S. Vahdati	On Steiner Trees and Minimum Spanning Trees in Hypergraphs
MPI-I-2001-1-004	S. Hert, M. Hoffmann, L. Kettner, S. Pion, M. Seel	An Adaptable and Extensible Geometry Kernel
MPI-I-2001-1-003	M. Seel	Implementation of Planar Nef Polyhedra
MPI-I-2001-1-002	U. Meyer	Directed Single-Source Shortest-Paths in Linear Average-Case Time
MPI-I-2001-1-001	P. Krysta	Approximating Minimum Size 1,2-Connected Networks
MPI-I-2000-4-003	S.W. Choi, H. Seidel	Hyperbolic Hausdorff Distance for Medial Axis Transform
MPI-I-2000-4-002	L.P. Kobbelt, S. Bischoff, K. Khler, R. Schneider, M. Botsch, C. Rssl, J. Vorsatz	Geometric Modeling Based on Polygonal Meshes
MPI-I-2000-4-001	J. Kautz, W. Heidrich, K. Daubert	Bump Map Shadows for OpenGL Rendering
MPI-I-2000-2-001	F. Eisenbrand	Short Vectors of Planar Lattices Via Continued Fractions
MPI-I-2000-1-005	M. Seel, K. Mehlhorn	Infimaximal Frames: A Technique for Making Lines Look Like Segments
MPI-I-2000-1-004	K. Mehlhorn, S. Schirra	Generalized and improved constructive separation bound for real algebraic expressions

MPI-I-2000-1-003	P. Fatourou	Low-Contention Depth-First Scheduling of Parallel Computations with Synchronization Variables
MPI-I-2000-1-002	R. Beier, J. Sibeyn	A Powerful Heuristic for Telephone Gossiping
MPI-I-2000-1-001	E. Althaus, O. Kohlbacher, H. Lenhof, P. Mller	A branch and cut algorithm for the optimal solution of the side-chain placement problem