

Free-Boundary Linear Parameterization of 3D Meshes in the Presence of Constraints

Zachi Karni

Max-Planck-Institut für Informatik
karni@mpi-sb.mpg.de

Craig Gotsman

Technion
gotsman@cs.technion.ac.il

Steven J. Gortler

Harvard University
sjg@eecs.harvard.edu

Abstract

Linear parameterization of 3D meshes with disk topology is usually performed using the method of barycentric coordinates pioneered by Tutte and Floater. This imposes a convex boundary on the parameterization which can significantly distort the result. Recently, several methods showed how to relax the convex boundary requirement while still using the barycentric coordinates formulation. However, this relaxation can result in other artifacts in the parameterization. In this paper we explore these methods and give a general recipe for “natural” boundary conditions for the family of so-called “three point” barycentric coordinates. We discuss the shortcomings of these methods and show how they may be rectified using an iterative scheme or a carefully crafted “virtual boundary”. Finally, we show how these methods adapt easily to solve the problem of constrained parameterization.

1. Introduction

One of the most widespread methods for parameterization of 3D manifold mesh data with the topology of a disk is based on Floater's generalization [3] of the powerful “spring-embedding” theorem of Tutte [15]. This theorem postulates that if the boundary of the mesh is embedded as a convex shape in the plane, the position of each interior vertex is a convex combination of those of its neighbors, and edges are drawn as straight lines – the result will be a *planar* embedding with convex faces. A planar embedding means that the edges of the embedded mesh will not intersect. The planarity of the embedding is crucial for many applications using parameterization, including texture mapping, remeshing and morphing, as it provides a bijective mapping between the mesh and a region of the plane (the parameter domain). Embeddings

which are not planar may contain “flipped” orientation faces, or regions “covered” by more than one face. The convex combinations are also sometimes called barycentric coordinates.

Tutte's theorem provides a very general and efficient method for parameterizing a disk-like mesh which essentially boils down to solving two sparse linear systems: one for the x coordinates of the embedded mesh vertices, and one for the y coordinates. The geometric properties of the embedding may be tuned to reflect the geometry of the original mesh by using different recipes for the barycentric coordinates. The most popular are the so-called shape-preserving weights [3], conformal (also known as the discrete harmonic) weights [14] and mean-value weights [4]. Conformal weights are derived as a discretization of the continuous conformal mapping, hence tend to minimize angular distortion. See the survey by Floater and Hormann for more details [5]. Recently, Floater et al. [6] have showed that many of these weights are actually members of a larger parametric family which we will elaborate on later.

1.1. The convex boundary

The main drawback of the basic method of Tutte is the convex boundary requirement. Many inputs have an inherently non-convex shape, and forcing the boundary to be convex introduces a glaring distortion into the result. This is especially disturbing when the input mesh is flat with a very non-convex boundary. Although there exists a planar embedding of the input which has no distortion whatsoever (the input itself!), the method will generate something quite different. This means that the method does not have the desirable property of *reproducing* 2D inputs. This property is sometimes called *linear precision*. See Figure 1a-b.

Realizing that for many applications, less distortion in the parameterization is more important than efficient computation of it, a few methods have been devised to

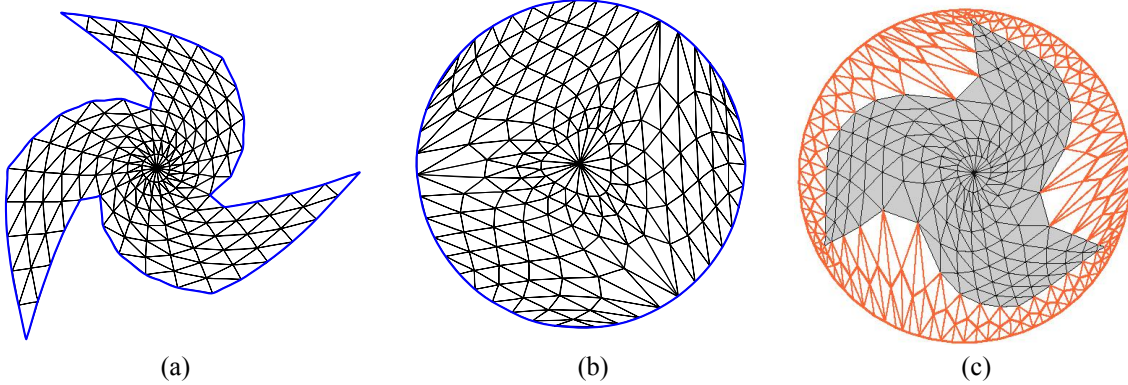


Figure 1: Embedding of a flat mesh. (a) The input mesh. (b) Planar embedding using Tutte's method: convex boundary and uniform weights. (c) Planar embedding of Lee et al [12] using virtual boundary (red triangles) and uniform weights.

accommodate non-convex boundaries. The most prominent is the Angle-Based-Flattening method [16] for triangle meshes, which operates in angle space. This is a highly non-linear method, and sophisticated numerical techniques must be employed in order to solve it efficiently [17]. Two other methods are related to Tutte's linear method: The first is the so-called "virtual boundary" of Lee et al. [12], also used in the polygon morphing methods of Gotsman and Surazhsky [8]. Here the mesh is "padded" with a small number of layers of triangles. The padded mesh is then embedded as usual with a convex boundary, which is then "peeled" off to leave just the original connectivity, which will hopefully have had sufficient freedom to relax to a less distorted non-convex shape. Lee proposes a number of heuristics that maximize the freedom of the boundary, but there seems to be a fair amount of arbitrariness in the method, which undoubtedly introduces bias in the result in unpredictable ways. While the result is always a planar embedding, it still does not reproduce 2D inputs, even those with convex boundaries. See Figure 1c. The second, more direct linear method, is that of Desbrun et al. [1] that works only for triangle meshes. In this method, explicit linear conditions are formulated for the boundary vertices. These boundary conditions are "natural" extensions of the conformal weight recipe. The resulting embedding is guaranteed to reproduce 2D inputs, and usually yields very nice results for 3D inputs. However, there is no guarantee that the resulting embedding will be planar, even if the conformal weights are positive. An interesting difference between this method and the regular Tutte method is that it requires the solution of one $2n \times 2n$ linear system coupling the x and y coordinates (instead of two $n \times n$ decoupled linear systems).

1.2. Accommodating constraints

For many computer graphics applications, in particular those involving texture mapping and remeshing, it is desirable that the parameterization satisfy hard constraints on the embedded positions of some of the mesh vertices. This makes the parameterization problem much more difficult. It renders the linear systems used in the method of barycentric coordinates over-determined, no matter whether the boundary is fixed or free. Hence, satisfying the constraints usually comes at the expense of the planarity of the embedding.

More sophisticated parameterization methods have been devised to solve the problem of constrained planar parameterization [2][11], but these are quite complex and difficult to implement. Furthermore, they introduce so-called Steiner (extra) vertices into the embedding in order that it both satisfy the constraints and be planar.

2. Our Contribution

This paper introduces two methods to generate planar free-boundary embeddings of manifold 3D triangle meshes with disk topology. Both methods are 2D reproducing. One is an iterative method capable also of satisfying a set of given constraints, if this may be done without introducing Steiner vertices.

Both our methods rely on the free-boundary linear systems introduced by Desbrun et al. [1], which, as mentioned above, is not guaranteed to generate an planar embedding by itself. For the iterative method, we also rely on recent results by Gortler et al. [7] on barycentric embeddings with non-convex boundaries. These provide precise conditions on the boundary vertices of an embedding generated using barycentric coordinates, in order that

it be planar. We show how to iterate the linear barycentric systems in order to satisfy the conditions stipulated by Gortler et al., incorporating also constraints. Thus our embeddings are guaranteed to be planar.

3. The Free-Boundary Equations

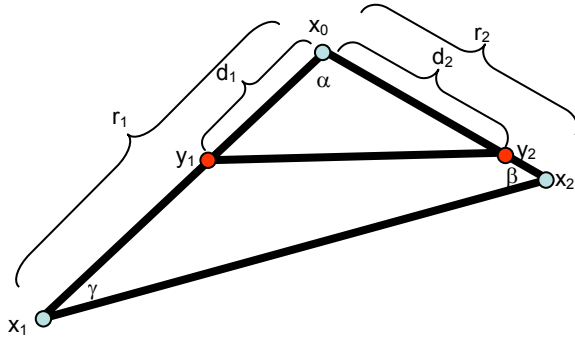
Desbrun et al. [1] derived natural boundary conditions for a triangle mesh in the special case of the conformal barycentric weights. We now show how to derive analogous boundary conditions for the larger family of barycentric coordinates containing these.

3.1. The general case

Floater et al. [6] showed that the following construction leads to the family of so-called “three point” barycentric coordinates of a point x_0 relative to its neighbors x_1, \dots, x_k in the plane.

Denote by r_i the lengths of the edges $\{x_0, x_i\}$ and choose points y_i along the edges, at distances d_i from x_0 . Now observe just the oriented triangle $\{x_0, x_1, x_2\}$ in Figure 2a. In this triangle we start from the trivial identity: $R^{90}(y_2 - y_1) = R^{90}(y_2 - x_0) - R^{90}(y_1 - x_0)$ where R^{90} is the 90° rotation operator transforming $(u, v) \rightarrow (-v, u)$. We may express the two vectors on the right-hand side of this identity as linear combinations of unit vectors aligned with the two triangle edges $\{x_0, x_1\}$ and $\{x_0, x_2\}$:

$$\begin{aligned} R^{90}(y_1 - x_0) &= \frac{d_1}{\sin(\alpha)} \cdot \frac{x_2 - x_0}{r_2} \\ &\quad - d_1 \cot(\alpha) \cdot \frac{x_1 - x_0}{r_1} \quad \text{and} \\ R^{90}(y_2 - x_0) &= d_2 \cot(\alpha) \cdot \frac{x_2 - x_0}{r_2} \\ &\quad - \frac{d_2}{\sin(\alpha)} \cdot \frac{x_1 - x_0}{r_1} \quad \text{so} \end{aligned}$$



(a)

$$\begin{aligned} R^{90}(y_2 - y_1) &= \left(d_2 \cot(\alpha) - \frac{d_1}{\sin(\alpha)} \right) \cdot \frac{x_2 - x_0}{r_2} \\ &\quad + \left(d_1 \cot(\alpha) - \frac{d_2}{\sin(\alpha)} \right) \cdot \frac{x_1 - x_0}{r_1} \end{aligned}$$

However, by definition:

$$R^{90}(y_2 - y_1) = R^{90} \left(d_2 \frac{x_2 - x_0}{r_2} - d_1 \frac{x_1 - x_0}{r_1} \right)$$

So, for this triangle we finally get:

$$\begin{aligned} &\left(d_2 \cot(\alpha) - \frac{d_1}{\sin(\alpha)} \right) \cdot \frac{x_2 - x_0}{r_2} + \\ &\left(d_1 \cot(\alpha) - \frac{d_2}{\sin(\alpha)} \right) \cdot \frac{x_1 - x_0}{r_1} = \quad (1) \\ &R^{90} \left(d_2 \frac{x_2 - x_0}{r_2} - d_1 \frac{x_1 - x_0}{r_1} \right) \end{aligned}$$

Let us examine this identity for some special cases:

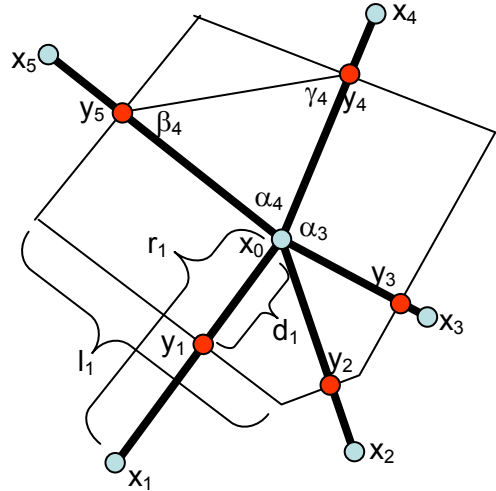
- Conformal ($d_i = r_i$):

Using the sine rule $r_2/r_3 = \sin(\beta)/\sin(\gamma)$ and the identity $\sin(\gamma) = \sin(\alpha + \beta)$, we reduce to the identity stated by Desbrun et al. [1]:

$$\begin{aligned} \cot(\gamma) \cdot (x_2 - x_0) + \cot(\beta) \cdot (x_1 - x_0) &= \\ R^{90}(x_2 - x_1) \end{aligned}$$

- Mean-Value ($d_i = 1$):

Using the identity $-\cot(\alpha) + 1/\sin(\alpha) = \tan(\alpha/2)$, we reduce to the following identity:



(b)

Figure 2: Part of a mesh, showing: (a) one triangle face, (b) a vertex x_0 and its neighbors.

$$\frac{\tan(\alpha/2)}{r_2}(x_2 - x_0) + \frac{\tan(\alpha/2)}{r_1}(x_1 - x_0) = R^{90} \left(\frac{x_2 - x_0}{r_2} - \frac{x_1 - x_0}{r_1} \right)$$

Now, in order to obtain the barycentric coordinates of x_0 relative to x_1, \dots, x_k , consider the closed ring of triangles around x_0 , as in Figure 2b. The sum of equation (1) over all triangles in the ring vanishes, and this leads to the following equation:

$$\sum_{i=1}^k w_i(x_i - x_0) = 0$$

$$w_i = \lambda_i / \sum_{i=1}^k \lambda_i \quad (2)$$

$$\lambda_i = \frac{1}{r_i} \left(\frac{d_{i-1}}{\sin(\alpha_{i-1})} - d_i \cot(\alpha_i) \right)$$

where α_i is the angle between the i -th and $(i+1)$ -th edges. Note that the indices are cyclic, meaning that $i_{k+1} = i_1$. It is straightforward to show that three known recipes for barycentric coordinates are special cases within this family:

- Conformal: $d_i = r_i$, $w_i = \cot(\beta_i) + \cot(\gamma_{i-1})$
- Mean-Value: $d_i = 1$, $w_i = (\tan(\alpha_{i-1}/2) + \tan(\alpha_i/2))/r_i$
- Wachspress: $d_i = 1/r_i$, $w_i = (\cot(\beta_{i-1}) + \cot(\gamma_i))/r_i$

where β_i and γ_i are the angles in the i -th triangle opposite y_i and y_{i+1} respectively. The weight w_i will be positive only if $\beta_{i-1} + \gamma_i < \pi$. This is always guaranteed for the mean-value coordinates because then the triangles $\{y_{i-1}, x_0, y_i\}$ are all isosceles, hence their base angles are all acute.

There is a simple geometric interpretation of these barycentric coordinates, observed by Kós [10]: Draw perpendiculars to the edges through the y_i until they meet each other, forming a polygon whose edges have length l_i . See Figure 2b. Then:

$$\lambda_i = \frac{l_i}{r_i}$$

Given an entire mesh of triangles, equation (2) will hold for every interior vertex, and asserts that the vertex has harmonic behavior. The more interesting observation is what equation (1) implies for a boundary vertex, which is not surrounded by a closed ring of triangles. If one triangle is “missing” from the ring, the sum of equation (1) over those triangles does not vanish, but is the negative of the value associated with that missing triangle. This yields the “natural” boundary condition for a boundary vertex v_0 with neighboring vertices v_1 and v_2 along the boundary:

$$\sum_{i=1}^k w_i(x_i - x_0) = R^{90} \left(d_2 \frac{x_2 - x_0}{r_2} - d_1 \frac{x_1 - x_0}{r_1} \right) \quad (3)$$

Geometrically, this means that the boundary vertex is not positioned at the barycenter of its neighbors, as are the interior vertices. Rather it is located at an offset from this barycenter, given by either side of equation (1). Note that this statement is valid also for a convex boundary vertex. In both cases, reflex and convex, the offset vector points “outwards”. Since these equations mix the x and y coordinates, we obtain a $2n \times 2n$ system of linear equations in $2n$ unknowns.

3.2. Lévy’s equations

Lévy et al. [13] proposed a different linear system for solving for a free boundary. This is based on the following simple trigonometric identity for a triangle: Two edges of a triangle will coincide if you rotate one around their common vertex by the angle between them, and then rescale its length by the ratio between their lengths:

$$\frac{r_1}{r_2}(x_2 - x_0) = R^\alpha(x_1 - x_0)$$

Since there are many more equations than unknowns

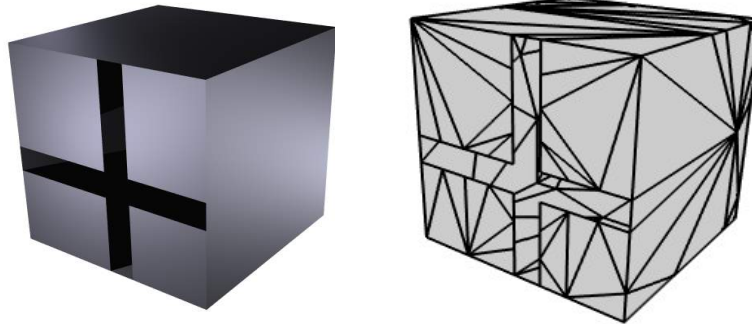


Figure 3: A model of a cube with one boundary. This model was used to test various embedding methods.

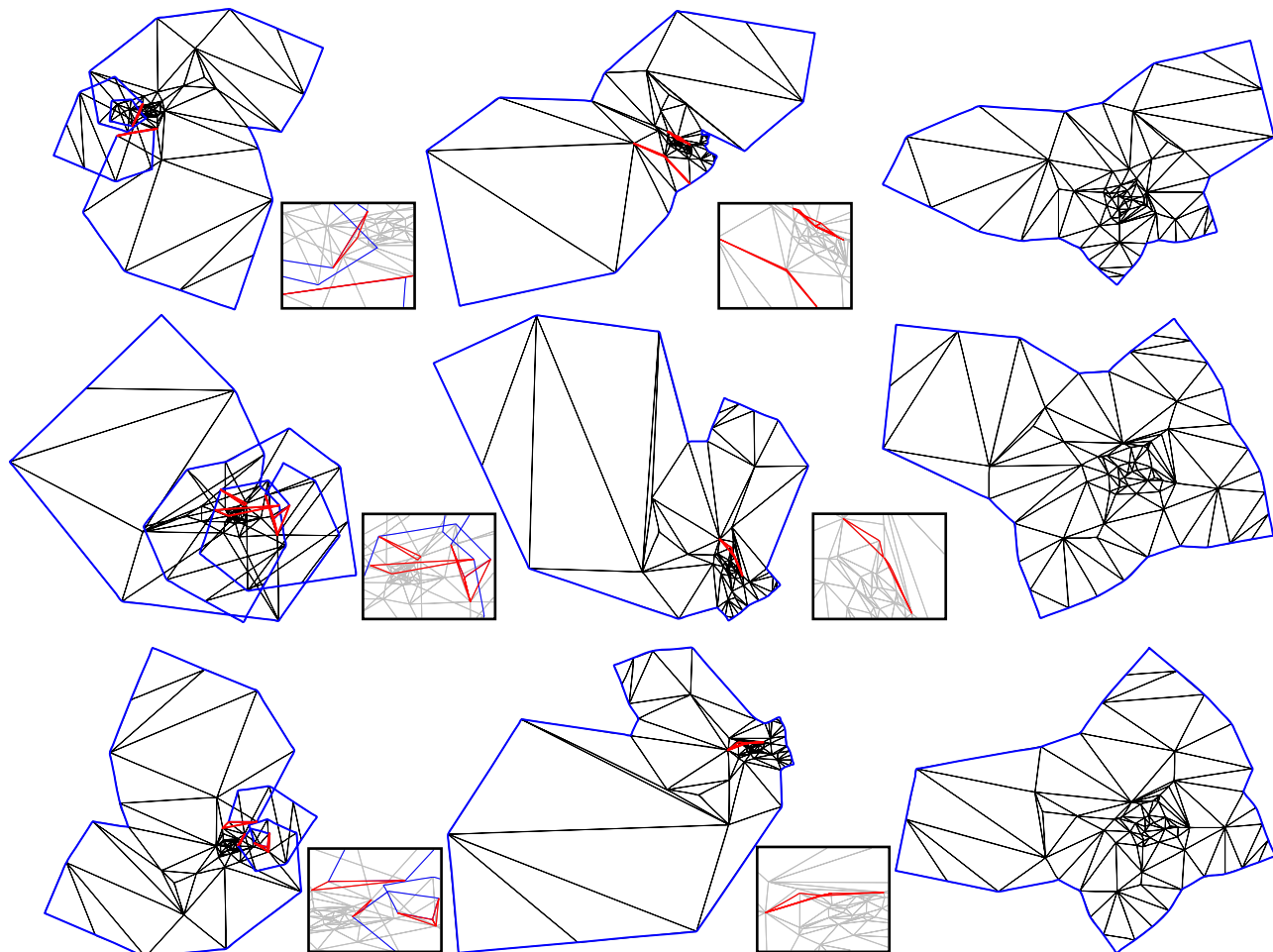


Figure 4: Free-boundary embeddings of the cube model using: (left) mean-value weights. (center) Lévy's weights (right) conformal weights. Red triangles have “flipped” orientation, hence the embedding is not planar. Small images zoom into the non-planar area. Each row corresponds to fixing a different pair of vertices

(twice the number of triangles vs. twice the number of vertices), Lévy proposed to solve this system in the least-squares sense. However, it is also possible to do with these triangle equations what we did with the family of barycentric coordinates discussed in the previous section: sum over the triangles around each vertex. This will then result in a $2n \times 2n$ linear system for the x and y coordinates of the mesh vertices.

3.3. Solving the free boundary equations

If the 3D mesh is developable, meaning that it can be embedded in the plane without any distortion, the rank of the $2n \times 2n$ matrix derived from the barycentric coordinate equations (2) and (3) (when the weights are computed from the 3D geometry) is $2n-4$. The co-rank of 4 is due to the equations being invariant to the four degrees of freedom of a similarity transform: two translation parameters, one rotation and one scale parameter. So in order to pro-

duce a concrete solution, these degrees of freedom must be fixed. An easy way to do this is fix two of the vertices in the plane (two coordinates each) and solve for the others. The fact that the embedding is 2D reproducible means that any developable triangle mesh, including one which is already a planar embedding, satisfies the barycentric equations derived from its geometry.

When the mesh is not developable, the rank of the matrix is only $2n-2$. This is because there is no solution apart from the trivial solution where all the triangles are degenerate, meaning that all vertices are positioned at one point, which is invariant to translation only. To obtain a meaningful solution, some of the constraints must be relaxed. This is usually done by fixing two vertices and eliminating the boundary conditions at those vertices.

We would like the solution to the free boundary equations to yield a planar embedding. While for many practical inputs the solution is indeed such, this is not guaranteed. In fact, it is easy to construct simple 3D triangle

meshes whose conformal and mean-value free boundary embeddings are not planar. We distinguish between *local planarity* and *global planarity*. Local planarity is violated when triangles cover some of the neighborhood around a vertex more than once, either by changing orientation (flipping), or cycling around the vertex more than once (double covering). Global planarity is violated if the boundary of the embedding is not simple, or has turning number other than 2π . Figure 4 shows some embeddings produced by the conformal, mean-value and Lévy free-boundary equations when parameterizing the same 3D input shown in Figure 3. The different embeddings were generated by fixing different pairs of vertices. It may be seen that the conformal embeddings are relatively well-behaved, and not too sensitive to the two points fixed. The mean-value embeddings seem to be quite sensitive.

4. The Iterative Method

Since none of the free boundary methods can guarantee a planar embedding, a post-processing step is necessary to "fix" any non-planarities that might be present in the results. The previous section showed that planar embeddings satisfy their mean-value equations (this is the 2D reproduction property). Theorem 2, proven in the Appendix, shows that *only* planar embeddings, when considered as 3D meshes, satisfy their mean-value equations. This fact implies that a possible way to achieve a planar embedding is by iterating the mean-value weighted embedding process (The proof of Theorem 1 in the Appendix shows that the mean-value weights alone are suitable). We can do this by applying the free-boundary method again on the output of the previous stage, modifying the barycentric coordinates used. Thus a planar input will not be changed, but a non-planar input will (and, hopefully, become "closer" to planar). This iterative process is not guaranteed to converge, but Theorem 2 guarantees that *if* the process *does* converge, the limit embedding will *always* be planar.

Another way of thinking about this iteration is as a solution to non-linear mean-value equations using the well-known fixed point iteration method ([9], Chap. 5). More formally, given a geometry x , the mean-value matrix $A(x)$ is derived from that geometry x , and x will satisfy its mean-value equations if $A(x) \cdot x = b$, or $x = A(x)^{-1} \cdot b$ for an appropriate b . The iteration is therefore $x_{k+1} = A(x_k)^{-1} \cdot b$ where $A(x_0)$ is derived from the original 3D mesh geometry as described in the following pseudo-code.

1. Solve the free boundary equations using conformal weights derived from the 3D geometry and fixing two boundary points. Let E be the resulting 2D embedding.
2. If E is planar – stop and output E .
3. Solve the free boundary equations using mean value weights derived from the 2D geometry of E . Let E be the result of the new embedding.
4. Goto 2.

In step 4 the mean value parameterization is applied to an embedding which is possibly non-planar, namely, containing negative angles if orientation is considered. Nonetheless we consider all angles as positive. The Appendix states and proves Theorem 2. Figure 5 shows the results of applying the iterated mean-value process to some of the non-planar embeddings of Figure 4.

5. The Virtual Boundary Method

Another way to generate a planar embedding with a free boundary is using a technique inspired by the virtual boundary technique of Lee [11]. Since now we operate purely in 2D, there is more information to build on. The following technique is proposed:

1. $k \leftarrow 2$;
2. Solve the free boundary equations using weights derived from the 3D geometry and fixing k boundary points to a convex polygon.
3. If the resulting embedding E is planar – stop and output E .
4. If B - the boundary of E - is not simple: $k \leftarrow k+1$, goto 2
5. Construct a virtual boundary B' that is the convex hull of B
6. Triangulate the simple polygon holes between B and B' and add the new faces to E , yielding an extended embedding E'
7. Solve the mean-value equations with boundary B' with weights derived from the 2D geometry of E' .

By Tutte's theorem, the resulting embedding is guaranteed to be planar. Furthermore, because the mean-value equations are 2D reproducing, the geometry of the result will probably not be too different from the non-planar geometry of E , if it was such. Also note that if the input to this procedure is developable, it will be reproduced.

Figure 6 shows the results of applying the virtual boundary process to some of the non-planar embeddings of Figure 4.

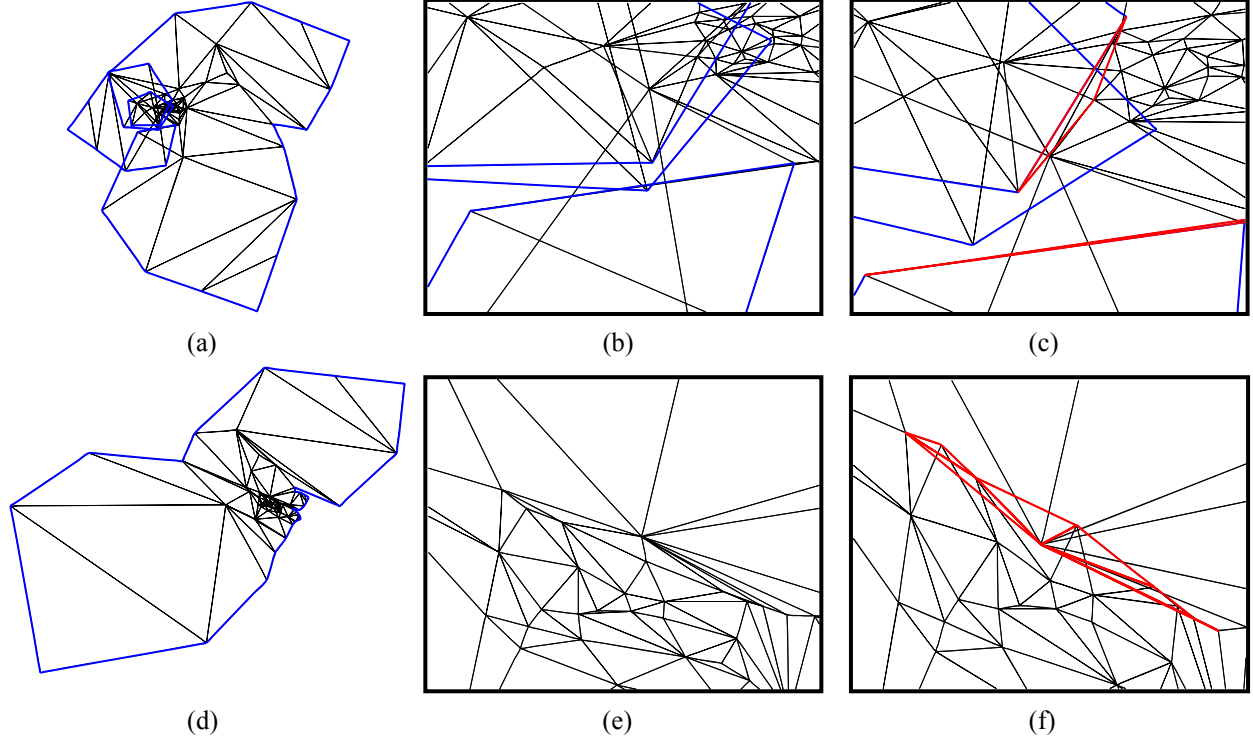


Figure 5: The iterative method: (a), (d) input (taken from Figure 4 (a) and (b)). (b),(e) and (c),(f) zoom into the fixed and non-planar regions.

6. Texture Mapping with Hard Constraints

One possible application that can benefit from the planar embedding techniques presented above is texture mapping with hard constraints. In texture mapping an image is wrapped on a mesh surface in order to give it a realistic look (e.g. a picture of a face mapped to a 3D mesh of a human head). However, in many cases it is important to register the image and the 3D mesh in order to generate a realistic result. For example, in a 3D face mesh, it is important to register the eyes of the mesh and the eyes in the image, and the same for the mouth, nose and any other prominent features. Figure 8 shows some examples of these registrations (also known as correspondences) taken from Kraevoy et al. [11].

The constrained texture mapping problem can be considered as a planar embedding of the 3D mesh while constraining some of the vertices to lie in a specified position in the texture coordinate plane. However, depending on the given correspondence, it is sometimes impossible to satisfy all of them and still provide a planar embedding. Eckstein et al. [2] and the *Matchmaker* method of Kraevoy et al. [11] added Steiner vertices to the mesh in order to increase the degrees of freedom of the problem and to enable it to satisfy all the constraints. In addition,

Matchmaker requires that the ABF method first be run to generate a simple free boundary. This is a major disadvantage of that method.

Although Steiner vertices can be crucial for the embedding, they sometimes can be avoided. The following method uses one of the free-boundary embedding methods described above to initially embed the mesh into the plane. Although only two vertices must be fixed in order to get an embedding, the entire set of correspondences can be fixed as well, by eliminating their harmonic equations or by using a hard-constrained least-squares solution. Fixing the constrained points means that the embedding satisfies them. If the resulting embedding is planar – the mission is accomplished. However, in most cases the embedding will not be planar and the iterative solution described above can be used to fix it. The drawback is that if the initial embedding is not planar, its boundary must be a simple polygon or at least have turning number of 2π in order to apply the iterative method. By using a least-squares system it is possible to weight the different constraints according to their importance. Figure 9 shows the results of embedding the mesh of Figure 8 with its correspondences. Although the initial embedding with the constraints was not planar, the iterative method managed to correct this within a few iterations.

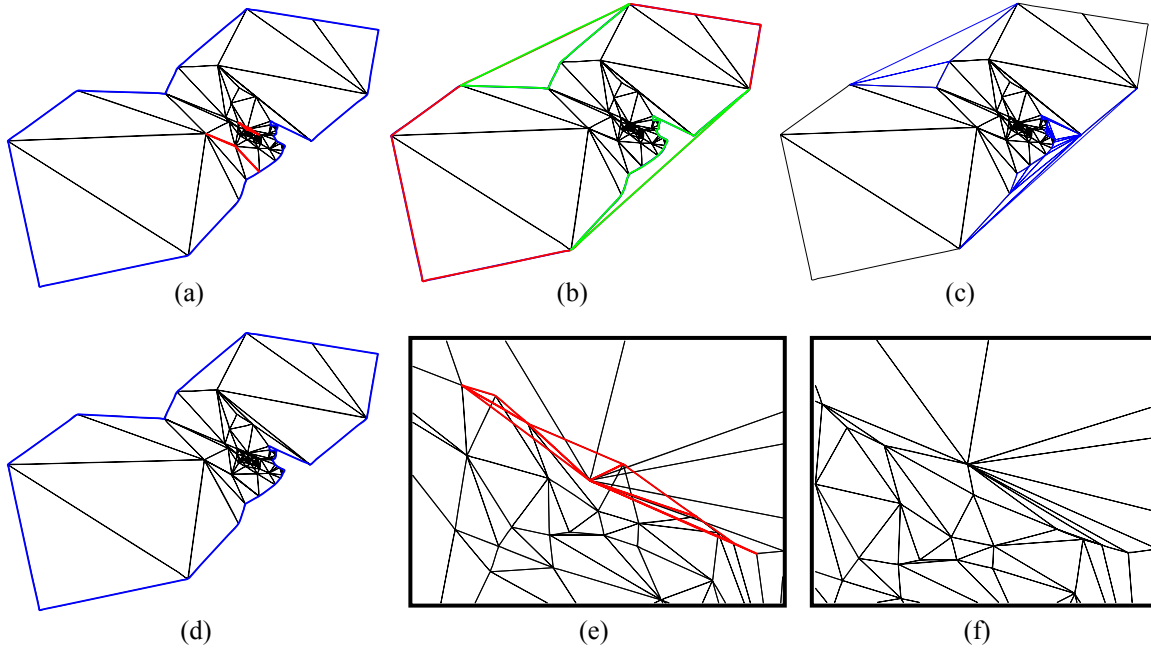


Figure 6: The virtual boundary method: (a) input (taken from Figure 4b), (b) the convex hull (red) and the holes (green), (c) holes triangulation, (d) the result, (e) zoom in on “flipped” faces in the non-planar input, (f) zoom in on same faces in the planar output.

7. Discussion and Conclusion

This paper has shown how to fix some of the shortcomings in the linear methods for free-boundary parameterization based on the method of barycentric coordinates. It has also shown how to incorporate constraints into the parameterization.

Our iterative method bears some resemblance to the method of Belyaev et al. [18] for minimizing parametric stretch distortion, in the sense that they modify barycentric weights between iterations in order to achieve less distortion. Our method, however, concentrates on achieving the planarity effect.

Some questions still remain open. First and foremost, we would like to formulate precise conditions for the convergence of the iterative procedure. This would allow us to determine in advance whether it is possible to apply the method. It is also not clear how to prevent a non-simple boundary from developing during the iteration without significantly constraining it to some specific shape.

8. Acknowledgements

We thank Mathieu Desbrun, Michael Floater, Geza Kós, Vladislav Kraevoy, Bruno Lévy and Alla Sheffer for

helpful discussions during the course of this work. Thanks also to Seungyong Lee and Yunjin Lee for providing outputs from their virtual boundary software.

This work was supported in part by AIM@SHAPE, a Network of Excellence project (506766) within EU's Sixth Framework Program and the Israel Ministry of Science, grant 01-01-01509.

9. References

- [1] M. Desbrun, M. Meyer and P. Alliez, “Intrinsic Parameterization of Surface Meshes”, *Computer Graphics Forum (Proceedings of Eurographics)*, 21(3):209-218, 2002.
- [2] I. Eckstein, V. Surazhsky and C. Gotsman. “Texture Mapping with Hard Constraints.” *Proceedings of Eurographics*, 2001.
- [3] M. S. Floater, “Parameterization of Smooth Approximation of Surface Triangulation”, *Computer Aided Geometric Design*, 14(3):231-250, 1997.
- [4] M. S. Floater, “Mean Value Coordinates”, *Computer Aided Geometric Design*, 20(1):19-27, 2003.
- [5] M. S. Floater and K. Hormann, “Surface Parameterization: a Tutorial and Survey”, *In Advances in Multiresolution for Geometric Modeling*, Springer, 2004.
- [6] M. S. Floater, K. Hormann and G. Kós, “A General Construction of Barycentric Coordinates Over Convex Polygons”, *To appear in Adv. in Comp. Math.*
- [7] S. J. Gortler, C. Gotsman and D. Thurston, “One-Forms on Meshes and Applications to 3D Mesh Parameteriza-

tion”, *To appear in Computer Aided Geometric Design*, 2005.

- [8] C. Gotsman and V. Surazhsky, “Guaranteed Intersection-Free Polygon Morphing”, *Computers and Graphics*, 25:67-75, 2001.
- [9] M. T. Heath, *Scientific Computing. An Introductory Survey*, 2nd Ed McGraw Hill, 2002.
- [10] G. Kós, “Affine and Barycentric Coordinates in 3D Polyhedra”, *Proceedings of Math. Methods for Curves and Surfaces*, Tromso, 2004.
- [11] V. Kraevoy, A. Sheffer and C. Gotsman, “Matchmaker: Constructing Constrained Texture Maps”, *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, 22(3):326-333, 2003.
- [12] Y. Lee, H-S Kim and S. Lee, “Mesh Parameterization with a Virtual Boundary”, *Computers & Graphics*, 26:677-686, 2002.
- [13] B. Lévy, S. Petitjean, N. Ray and J. Maillot, “Least Square Conformal Maps for Automatic Texture Atlas Generation”, *Proceedings of ACM SIGGRAPH*, 362-371, 2002.
- [14] U. Pinkall and K. Polthier, “Computing Discrete Minimal Surfaces and Their Conjugates”. *Experimental Mathematics*, 2(1):15-36, 1993.
- [15] W. T. Tutte, “How to Draw a Graph”, *Proceedings of the London Mathematical Society*, 13(3):743-768, 1963.
- [16] A. Sheffer and E. de Sturler, “Parameterization of Faceted Surfaces for Meshing Using Angle Based Flattening”. *Engineering with Computers*, 17(3):326-337, 2001.
- [17] A. Sheffer, B. Lévy, M. Mogilnitsky and A. Bogomjakov. “ABF++ : Fast and Robust Angle Based Flattening”, *To appear in ACM Transactions on Graphics*, 2005.
- [18] S. Yoshizawa, A. Belyaev and H.-P. Seidel, “A Fast and Simple Stretch-Minimizing Mesh Parameterization”, *Proc. Shape Modeling International*, 200-208, 2004.

Appendix

Lemma 1: Let v_1, \dots, v_n be a set of points in the plane. Assume WLOG that v_1 and v_2 are consecutive on the convex hull $C = \text{CH}(v_1, \dots, v_n)$. Consider the point $u-d$, where u is arbitrary and d is the “offset” vector $d = a(v_1 - u) + b(v_2 - u)$ for some $a, b > 0$ such that the triangles $T(v_1, u, v_2)$ and $T(v_1, u-d, v_2)$ have the same orientation. If $u-d \in C$ then also $u \in C$.

Proof: Assume $u-d \in C$. Since v_1 and v_2 are consecutive on C then:

$$(A1) \quad u-d \in T(v_1, v_2, v_3),$$

where v_3 is some point on C other than the segment between v_1 and v_2 .

Assume that $u \notin T(v_1, v_2, v_3)$. This means u is equal to some affine combination of v_1, v_2, v_3 with unique weights w_1, w_2, w_3 summing to unity, where at least one is negative: $u = w_1 v_1 + w_2 v_2 + w_3 v_3$. w_3 cannot be negative, because then the orientation of $T(v_1, u, v_2)$ would differ from that of

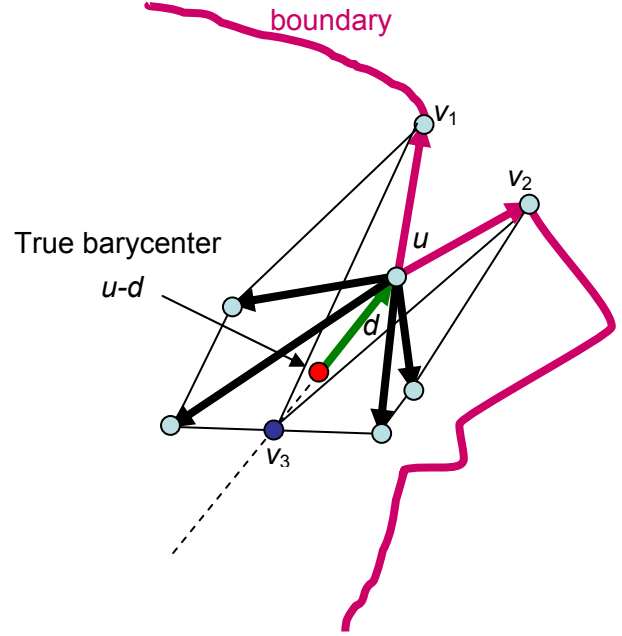


Figure 7: The scenario for a boundary vertex satisfying the mean-value equations: The vertex is offset from the barycenter of its neighbors in a direction which is within the wedge formed by that vertex and its two neighbors.

$T(v_1, u-d, v_2)$. So at least one of w_1, w_2 is negative. Assume WLOG that $w_1 < 0$. Now:

$$u-d = w_1 v_1 + w_2 v_2 + w_3 v_3 - a(v_1 - u) - b(v_2 - u) = (w_1 - a + aw_1 + bw_1)v_1 + (w_2 - b + aw_2 + bw_2)v_2 + w_3(1+a+b)v_3.$$

It is easy to see that the coefficient of v_1 is negative, meaning $u-d \notin T(v_1, v_2, v_3)$, in contradiction to (A1). \square

We are now ready to state our main theorem:

Theorem 1: Let G be a planar 3-connected triangular graph. An embedding of G with boundary having turning number 2π is planar if and only if the embedding satisfies the mean value equations (2) and (3) associated with it.

Proof: A. (only if) If the embedding is planar, then each triangle in the embedding satisfies the mean-value triangle equality (1) associated with that triangle. Since the associated mean-value equations (2) and (3) are derived by summing the triangle equalities around each vertex, obviously the embedding still satisfies them.

B. (if) If the geometry of the embedding satisfies the associated mean-value equations, this means that each interior vertex is a convex combination of its neighbors. Additionally, each boundary vertex u is either convex or reflex. If u is reflex, then examining the mean-value equation for the “missing” triangle shows that u is offset from the barycenter of its neighbors $u-d$ by a vector d which coincides with the bisector of the exterior angle it forms

with its two neighbors v_1 and v_2 . See Figure 7. Thus the scenario of Lemma 1 holds. Now since $u-d$ is in the convex hull of the neighbors of u , Lemma 1 implies that u is also in this convex hull. Gortler et al. [7] have shown that a straight-line embedding of a 3-connected triangular graph with a boundary having a turning number 2π , such that each interior vertex is in the convex hull of its neighbors and each reflex boundary vertex is also in the convex hull of its neighbors, is planar. As we have just proved that satisfying the mean-value equations guarantees that every reflex vertex of the boundary is contained in the convex hull of its neighbors, the theorem of Gortler

et al is applicable and the embedding will be planar. \square

Theorem 1 allows us to show that the limit of the iterated mean-value process, if it exists, must be a planar embedding.

Theorem 2: If the iterated mean-value process applied to a 3-connected manifold triangle mesh converges to an embedding with boundary having turning number 2π , then this embedding is planar.

Proof: Apply direction B of Theorem 1 to the limit embedding. \square

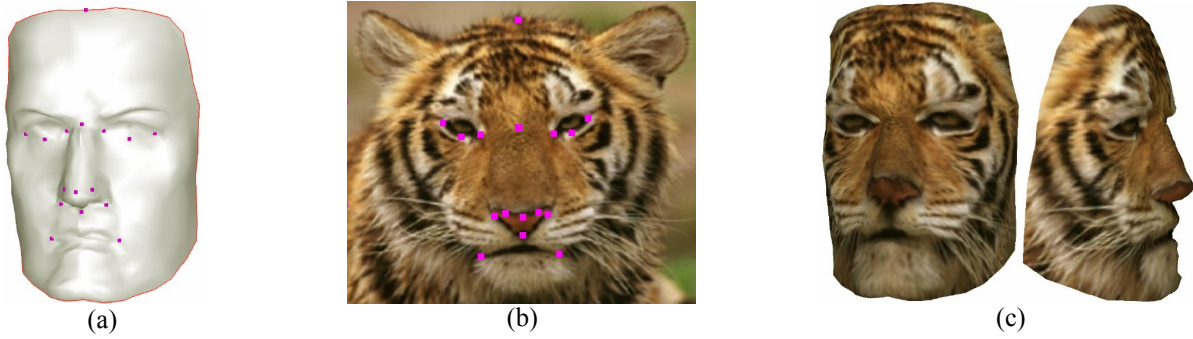


Figure 8: (a) A 3D model of a face together with several correspondence points. (b) A texture image with the same correspondence points. (c) Constrained texture mapping by MatchMaker [11].

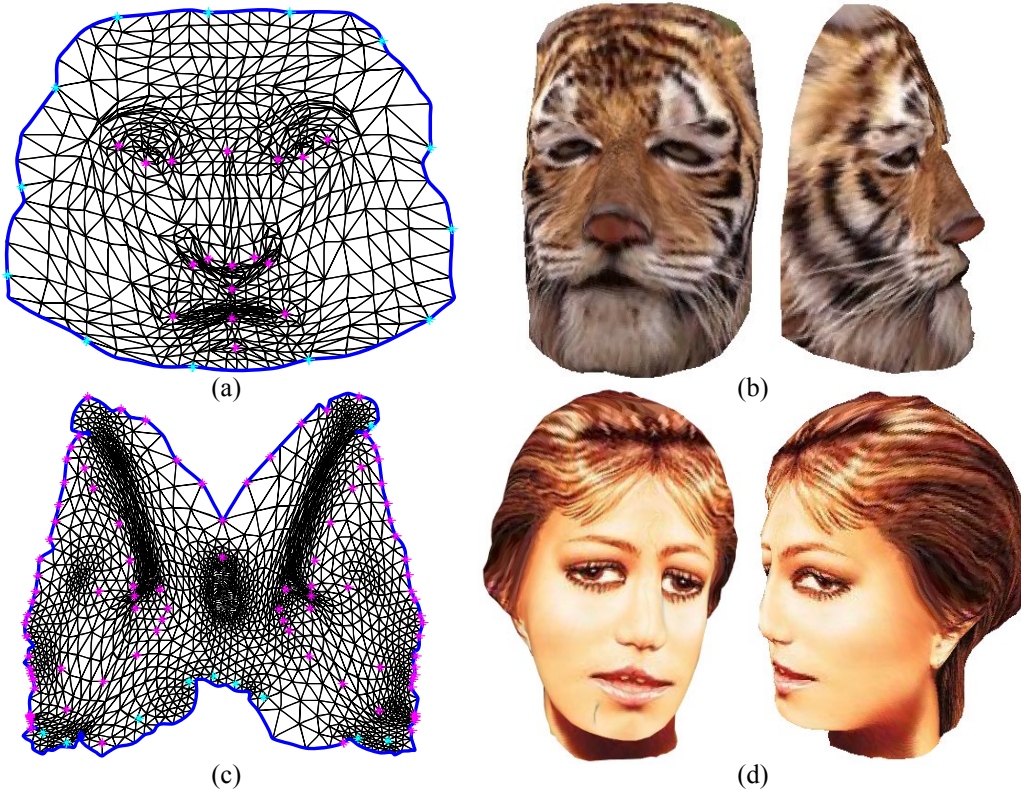


Figure 9: (a,c) Constrained parameterization using free-boundary weighted least-squares. The magenta points were fixed using large weights while the cyan points were fixed using medium weights. Both are planar embeddings (containing no flips). (b,d) The resulting constrained texture mapping.