Computer Aided Geometric Design 28 (2011) 349–356

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Embedding a triangular graph within a given boundary ☆

Yin Xu [a], Renjie Chen [b], Craig Gotsman [b,*], Ligang Liu [a]

[a] Dept. of Mathematics, Zhejiang University, Hangzhou 310027, China
[b] Computer Science Faculty, Technion – Israel Institute of Technology, Haifa 32000, Israel

ARTICLE INFO

Article history:
Received 23 March 2010
Accepted 4 July 2011
Available online 23 July 2011

Keywords:
Triangulation
Embedding
Planar graph

ABSTRACT

Given a 3-vertex-connected triangular planar graph and an embedding of its boundary vertices, can the interior vertices be embedded to form a valid triangulation? We describe an algorithm which decides this problem and produces such an embedding if it exists.

1. Introduction

Given a 3-vertex-connected triangular planar graph and an embedding of its boundary vertices in the plane, can the interior vertices be embedded to form a valid triangulation? If the boundary vertices are in convex position, Tutte's celebrated spring-embedding theorem (Tutte, 1963) states that such an embedding is always possible, and may be constructed by solving a linear system of equations for the coordinates of the interior vertices. This linear system is a discrete version of the classical Laplace equation expressing the fact that each interior vertex is positioned at the centroid of its immediate neighbors. Floater (1997) later generalized Tutte's theorem to arbitrary convex combinations, namely that the weights used to form the centroid may be arbitrary non-negative numbers summing to unity. This "convex combinations" method can generate all possible triangulations with the given convex boundary.

Hong and Nagamochi (2006) showed that an embedding is still always possible if the boundary embedding forms a star-shaped polygon. Their proof is constructive, but results in a rather complicated algorithm. For a general-shaped boundary, an embedding may not always exist, as demonstrated by the simple input shown in Fig. 1: the graph has a single interior vertex connected to all boundary vertices. Obviously, if the boundary is not embedded as a star-shaped polygon, this boundary polygon will have no kernel, so the set of valid positions for the interior vertex will be empty.

In this paper we treat the general case. We provide a simple algorithm which will either compute a valid embedding if it exists, or reject the input in case no valid embedding exists.

2. Preliminaries

Let $T = (G, B, X)$ be a 2D triangulation with n vertices. $G = \langle V, E\rangle$ is the planar *graph*, $B \subseteq V$ is the set of *boundary* vertices of G, and $X = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ is the *geometry* of the graph, namely, the coordinates of the vertices in V.

Since G is planar, there is a consistent orientation of all its triangles. Based on this orientation, given geometry X of the vertices, a normal vector may be computed for each triangle.

☆ This paper has been recommended for acceptance by H.-P. Seidel.

* Corresponding author. Tel.: +972 4 8294336; fax: +972 4 8295538.
E-mail addresses: yinxu.zju@gmail.com (Y. Xu), renjie.c@gmail.com (R. Chen), gotsman@cs.technion.ac.il (C. Gotsman), ligangliu@zju.edu.cn (L. Liu).

0167-8396/$ – see front matter
doi:10.1016/j.cagd.2011.07.001

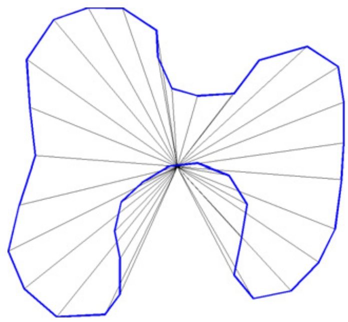

Fig. 1. An input for which no valid embedding exists: a non-star-shaped boundary with a single interior vertex.

Definition. T is *valid* if all the normals of the triangles of T point in the same direction along the z axis, otherwise T is *invalid*.

Definition. The *signed area* of a triangle is defined to be positive if the triangle normal points in the positive z direction, otherwise it is negative. The *signed area* of T, denoted $A(T)$, is the sum of the signed areas of the triangles of T. The *unsigned* area of T, denoted $U(T)$, is the sum of the areas of the triangles of T, ignoring sign.

Fact 1. *$A(T)$ is equal to the area of the boundary polygon of T.*

Fact 2. *For any triangulation T, $U(T) \geqslant A(T)$. T is valid iff $U(T) = A(T)$.*

Denote by L_0 the $n \times n$ matrix (combinatorial) graph Laplacian matrix of G. Tutte's algorithm for embedding the interior vertices subject to a convex boundary solves the linear system $L_0^* = b$ for the geometry $X = (x, y)$. The matrix L_0^* comes from replacing the rows of L_0 related to boundary vertices by the identity matrix. The vector b is derived from the boundary embedding $X(B)$ (X and b are $n \times 2$ matrices).

Definition. A *geometric Laplacian* $L(X)$ of G is a matrix, having the same structure as L_0, which takes into account the geometry of the triangulation X:

$$L_{ij}(X) = \begin{cases} -w_{ij} & j \in VN(i) \\ \sum_{k \in VN(i)} & j = i \\ 0 & \text{othervise} \end{cases}$$

where $VN(i)$ is the set of vertices neighboring to vertex i, and w_{ij} are dependent on X (e.g. the Laplacian matrices defined in Hormann and Floater, 2006, Pinkall and Polthier, 1993). In particular, $L(X)$ is called the *cotangent Laplacian* if w_{ij} are the so-called *cotangent* weights (Pinkall and Polthier, 1993), involving the two angles opposite the edge (i, j):

$$w_{ij} = \frac{1}{2}(\cot\alpha_{ij} + \cot\alpha_{ji})$$

For a boundary edge, only one cotangent is involved in the definition of w_{ij}.

From now on $L(X)$ will denote only the cotangent Laplacian. $L(X)$ is known to be symmetric and positive semi-definite (SPSD) (even though the w_{ij} may have mixed signs). This Laplacian is particularly appealing since the product $L(X)X$ vanishes at the interior vertices, namely, these vertices satisfy the *Laplace equation* subject to so-called *Dirichlet boundary conditions*. This is sometimes called the *reproduction* property of the cotangent Laplacian, namely, the geometry of the interior vertices can be recovered from the geometry of the boundary by solving the Laplace equation.

Since the unsigned area U of any triangle with edge lengths a, b, c and corresponding angles α, β, γ, may be expressed as:

$$U = \frac{1}{4}\left(\alpha^2 \cot\alpha + b^2 \cot\beta + c^2 \cot y\right)$$

It is straightforward to see that the unsigned area of a triangulation may be expressed in terms of $L(X)$, where $X = (x, y)$ is the geometry of T:

$$U(T) = \frac{1}{2}\operatorname{tr}\left(X^t L(X) X\right) = \frac{1}{2}\left(x^t L(X) x + y^t L(X) y\right) \tag{1}$$

implying, by Fact 2:

Algorithm **Embed:**

1. $L_0 \leftarrow$ combinatorial Laplacian of G. $U_0 \leftarrow 0$. $n \leftarrow 0$.
2. Solve the Tutte system: $L_n(I,I)X_{n+1}(I) + L_n(I,B)X_{n+1}(B) = 0$ for X_{n+1}, where I is the set of interior vertices, and B the set of boundary vertices.
3. Update the cotangent Laplacian: $L_{n+1} \leftarrow L(X_{n+1})$.
4. $U_{n+1} \leftarrow \frac{1}{2}\,\mathrm{tr}(X^t_{n+1} L_{n+1} X_{n+1})$.
5. If $U_{n+1} = U_n$ (up to a tolerance), output X_{n+1} and stop.
6. Else $n \leftarrow n+1$, Goto Step 2.

Fact 3. *Let $T = (G, B, X)$ be a triangulation for which a valid embedding exists subject to the given boundary embedding $X(B)$. T is valid iff its geometry X minimizes* $\mathrm{tr}(X^t L(X)X)$.

Note that $U(T)$ is not a simple quadratic form in X since the matrix involved depends on X.

3. The embedding algorithm

In this section we describe our iterative embedding algorithm. We will prove its correctness by showing that the iteration reduces $U(T) = \frac{1}{2}\,\mathrm{tr}(X^t L(X)X)$ until it reaches $A(T)$, at which point the triangulation is valid, if such an embedding exists.

First we show that Algorithm **Embed** always reduces $U(T)$.

Theorem 1. *For all $n \geqslant 0$, $U(T_{n+1}) \leqslant U(T_n)$.*

Proof. First, consider the piecewise linear mapping $f : R^2 \to R^2$ defined on $T_n = (G, R, X_n)$ by

$$f(X_n) = X_{n+1}$$

The *Dirichlet energy* of f can be written as follows,

$$E_D(f) = \frac{1}{2}\,\mathrm{tr}\big(f^t L_n f\big) = \frac{1}{2}\,\mathrm{tr}\big(X^t_{n+1} L_n X_{n+1}\big)$$

With the following ordering of X_{n+1} and L_n:

$$X_{n+1} = \begin{pmatrix} X_{n+1}(I) \\ X_{n+1}(B) \end{pmatrix}, \qquad L_n = \begin{pmatrix} L_n(I,I) & L_n(I,B) \\ L_n(I,B) & L_n(B,B) \end{pmatrix}$$

we can rewrite $E_D(f)$ as

$$E_D(f) = \frac{1}{2}\,\mathrm{tr}\big(X^t_{n+1}(I)L_n(I,I)X_{n+1}(I) + 2X^t_{n+1}(I)L_n(I,B)X_{n+1}(B) + X^t_{n+1}(B)L_n(B,B)X_{n+1}(B)\big)$$

implying the following gradient for interior vertices:

$$\frac{\partial E_D(f)}{\partial X_i} = \big(L_n(I,I)X_{n+1}(I) + L_n(I,B)X_{n+1}(B)\big)_i \quad \forall i \in I$$

Critical points of $E_D(f)$ are obtained when all such gradients vanish. Since L_n is SPSD, $L_n(I,I)$ is also SPSD, implying that all critical points are minima, X_{n+1} is the solution to the following (convex) quadratic program:

$$\min_Y E_D(Y) \quad \text{s.t.} \quad Y(B) = X_n(B)$$

Obviously, by (1):

$$U(T_n) = E_D(id)$$

where *id* is the identity map on T_n. Thus

$$U(T_n) = E_D(id) \geqslant E_D(f) \tag{2}$$

Second, following the definition of the Dirichlet energy, with f_t denoting the restriction of f to triangle t:

$$E_D(f) = \frac{1}{2}\int_T \big\|\nabla f\big\|^2 = \frac{1}{2}\sum_{t\in T_n} U(t)\left[\left(\frac{\partial f_t}{\partial x}\right)^2 + \left(\frac{\partial f_t}{\partial y}\right)^2\right] \geqslant \sum_{t\in T_n} U(t)\frac{\partial f_t}{\partial x}\frac{\partial f_t}{\partial y} = \sum_{t\in T_{n+1}} U(t) = U(T_{n+1}) \tag{3}$$

Finally, combining (2) and (3) yields:

$$U(T_n) \geqslant E_D(f) \geqslant U(T_{n+1}) \qquad \square$$

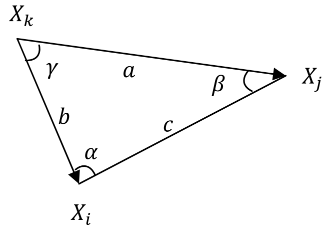

Fig. 2. Notation for a single triangle in a triangulation.

Next we show that Algorithm **Embed** always converges to valid embeddings, if they do exist for the given connectivity and boundary conditions.

Theorem 2. *$U(T)$ has a unique minimal value (as a function of X), obtained at all valid embeddings.*

Note that this is quite a strong statement. $U(T)$ has no local minima, rather multiple global minima.

Proof of Theorem 2. For any triangle $t = (X_i, X_j, X_k)$ as shown in Fig. 2, its unsigned area may be expressed as:

$$A(t) = \frac{1}{2}(X_i - X_k) \times (X_j - X_k)$$

Recall that $X = (x, y)$. Hence the derivative of $A(t)$ is:

$$\frac{\partial A(t)}{\partial x_i} = \frac{1}{2}(1,0) \times (X_j - X_k) = \frac{1}{2}(y_j - y_k)$$
$$\frac{\partial A(t)}{\partial y_i} = \frac{1}{2}(0,1) \times (X_j - X_k) = \frac{1}{2}(x_j - x_k)$$

Combining the two equations yields,

$$\frac{\partial A(t)}{\partial X_i} = \frac{1}{2}R_{\pi/2}(X_k - X_j)$$

where $R_{\pi/2}$ is the operator representing counter-clockwise rotation by $\pi/2$ in the plane. Then for each vertex X_i in T, when $A(t) \neq 0$:

$$\begin{aligned}\frac{\partial U(T)}{\partial X_i} &= \sum_{t\in TN(i)} \frac{\partial U(T)}{\partial X_i} = \sum_{t\in TN(i)} \frac{\partial\, \mathrm{sgn}(A(t))A(t)}{\partial X_i} = \sum_{t\in TN(i)} \mathrm{sgn}\big(A(t)\big)\frac{\partial A(t)}{\partial X_i} \\ &= \sum_{t\in TN(i)} \mathrm{sgn}\big(A(t)\big)\frac{1}{2}R_{\pi/2}(X_{t,k} - X_{t,j}) \\ &= \frac{1}{2}R_{\pi/2}\Bigg[\sum_{t\in TN(i)} \mathrm{sgn}\big(A(t)\big)(X_{t,k} - X_{t,j})\Bigg] \end{aligned} \tag{4}$$

where $TN(i)$ is the set of triangles incident on vertex i, and

$$\mathrm{sgn}\big(A(t)\big) = \begin{cases} 1, & \text{if } t \text{ is valid} \\ -1, & \text{if } t \text{ is invalid} \end{cases}$$

The geometric meaning of the vector $\sum_{t\in TN(i)} \mathrm{sgn}(A(t))(X_{t,k} - X_{t,j})$ in (4) is the weighted sum of the edge vectors around the 1-ring of the vertex x_i. When the triangle t associated with an edge is invalid, the weight associated with that edge, $\mathrm{sgn}(A(t))$, is negative. See Fig. 3.

For an arbitrary *interior* vertex X_i with a valid 1-ring, we obviously have

$$\frac{\partial U}{\partial X_i} = \frac{1}{2}R_{\pi/2}\Bigg[\sum_{t\in TN(i)} (X_{t,k} - X_{t,j})\Bigg] = \frac{1}{2}R_{\pi/2}[0] = 0$$

while for an invalid 1-ring in general position, the signed sum of edge vectors $\sum_{t\in TN(i)} \mathrm{sgn}(A(t))(x_{t,k} - x_{t,j})$ cannot be zero. This implies that only valid embedding have vanishing gradients on all the interior vertices. Thus there are no minima of $U(T)$ except when X is a valid embedding (and for these cases, $U(T) = A(T)$). □

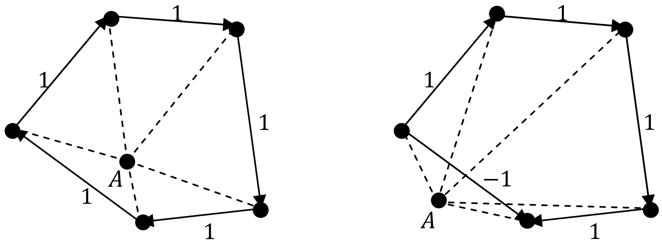

Fig. 3. Weights of edge vectors of a 1-ring in valid and invalid triangulations. Left: valid 1-ring; Right: invalid 1-ring.

Theorems 1 and 2 imply the correctness of Algorithm **Embed**:

Theorem 3. *Algorithm **Embed** will converge to a valid embedding of T, if it exists.*

An immediate question is what will happen if *no* valid embedding exists for the given boundary embedding? We have already proven that Algorithm **Embed** will not converge to any invalid embedding, and because the cotangent Laplacian is not well-defined for a degenerate triangle, it will collapse due to numerical error. This can be detected and failure reported.

4. Relation to standard optimization methods

Since our embedding algorithm minimizes the unsigned area of the triangulation, the informed reader may ask why a standard non-linear optimization procedure is not used, in particular, the Newton method. In this section, we compare our algorithm with the Newton method, which requires knowledge of the Jacobian and Hessian of the cost function. We will show that our algorithm is actually a "generalized" Newton method, which approximates the Hessian of the unsigned area by linearizing its Jacobian.

By the gradient property proved in Appendix A, the Jacobian J of $U(T)$ satisfies $J = \frac{\partial U}{\partial X} = LX$. Given the Hessian H and the Jacobian J, the standard Newton method solves the linear system $H\delta_X = -J$ to update X with $X + \delta_X$. Recall that the cotangent Laplacian L is actually a function of X. But if we fix L, treating it as a constant and approximate the Hessian linearly by:

$$H = \frac{\partial J}{\partial X} = \frac{\partial (LX)}{\partial X} \approx L$$

then the linear system which arises in the Newton method becomes:

$$H\delta_X = -J \quad \Leftrightarrow \quad L_n(X_{n+1} - X_n) = -L_n X_n \quad \Leftrightarrow \quad L_n X_{n+1} = 0$$

which is exactly step (2) of our iterative Algorithm **Embed**.

Because of the gradient property, we have an accurate expression for the Jacobian of the unsigned area. Thus the classical gradient descent method may also be used to minimize the unsigned area. However, this requires a step size, and estimating a good value for this may be difficult.

5. Experimental results

To demonstrate the effectiveness of the embedding algorithm, we show its output on two triangulations whose boundaries are significantly non-convex, as illustrated in Fig. 4.

Another experiment shows the embeddings generated when the boundary embedding of a valid triangulation is modified, as illustrated in Fig. 5.

5.1. The weighted Laplacian

As evident in Figs. 4 and 5, the resulting valid embeddings may be quite ugly, containing "skinny" triangles. It is possible to alleviate this by using a modified Laplacian, which we call a "weighted Laplacian". The non-zero entries of this Laplacian are defined as:

$$w_{ij} = |A_{ij}| \cot \alpha_{ij} + |A_{ji}| \cot \alpha_{ji}$$

where A_{ij} and A_{ji} are the signed areas of the triangles adjacent to edge (i, j), respectively. Since using the common cotangent Laplacian in the embedding algorithm minimizes the sum of unsigned areas of the triangulation, it is easy to see that the analogous cost function based on the weighted Laplacian minimizes the sum of squares of the areas of the triangles, namely $E = \sum_t A_t^2$. Since $A(T) = \sum_t A_t$ is constant, minimizing E is equivalent to minimizing the variance of the triangle areas. This has the effect of equalizing the areas of the triangles, thus "fattening" them. See some results in Fig. 6.

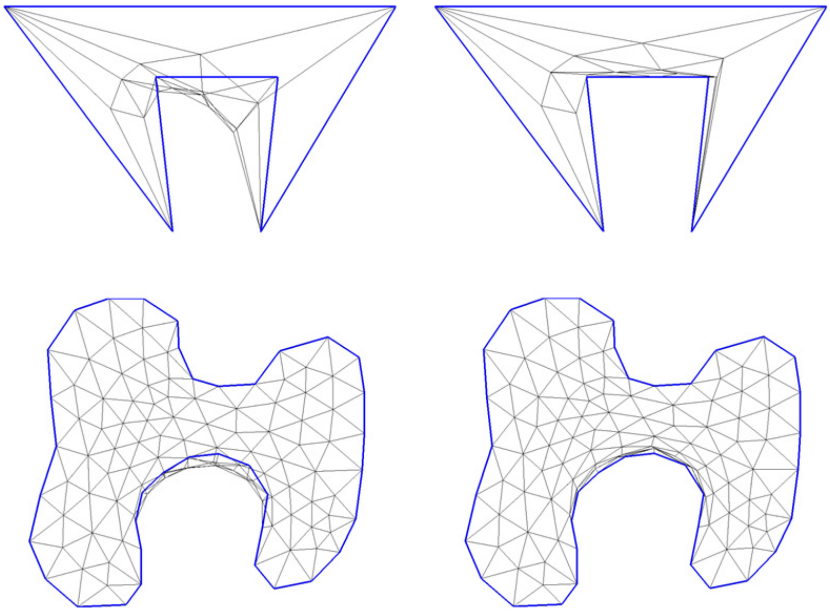

Fig. 4. Embedding with a non-convex boundary. Left: initial invalid embedding (generated using Tutte's method). Right: final valid embedding generated by our embedding algorithm.

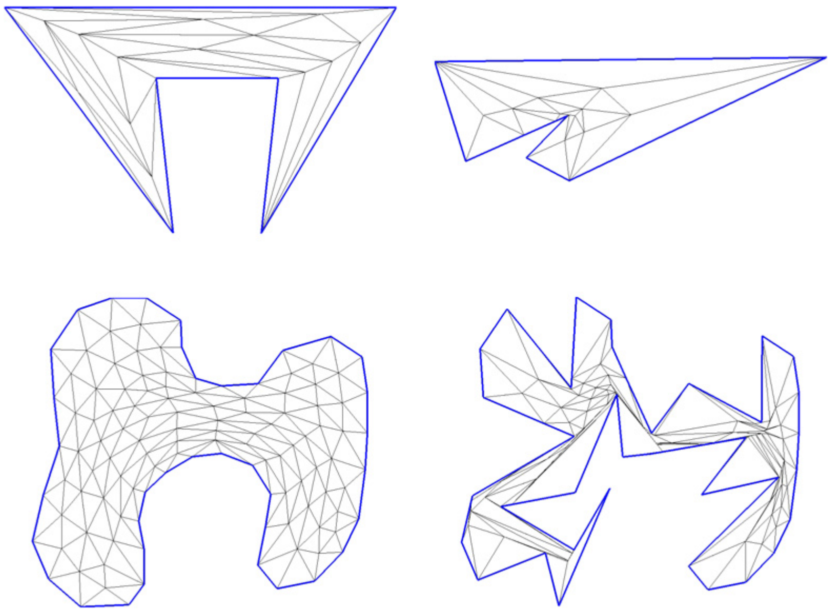

Fig. 5. Modifying the boundary and re-embedding the interior vertices. Left: original triangulation. Right: valid triangulation after changing boundary significantly.

The problem with the weighted Laplacian is that it is not guaranteed to always reduce the original unweighted cost function during the iteration. Thus the triangulation achieving a global minimum might actually be invalid. See Fig. 7 for a simple example. So it is best to use this method as a post process to improve the quality of a valid embedding (stopping before the embedding becomes invalid).

5.2. Numerical issues

The non-zero entries of the cotangent Laplacian may not be well-defined when some of the angles are zero. Since the embedding algorithm may produce some degenerate triangles, especially when no valid embedding exists, this problem may arise during the run. Even if no degenerate triangle occurs, some triangles may become quite skinny, so that the computation of the Laplacian becomes unstable and numerical error can cause the algorithm to diverge.

One way to avoid this numerical issue is to modify the "zero angle" entry of the Laplacian to force it to be well-defined. It turns out that simply setting this offending entry of the Laplacian to zero alleviates the problem. This modification keeps the energy continuous and prevents the algorithm from behaving badly. To measure the degeneracy of a triangle, one could

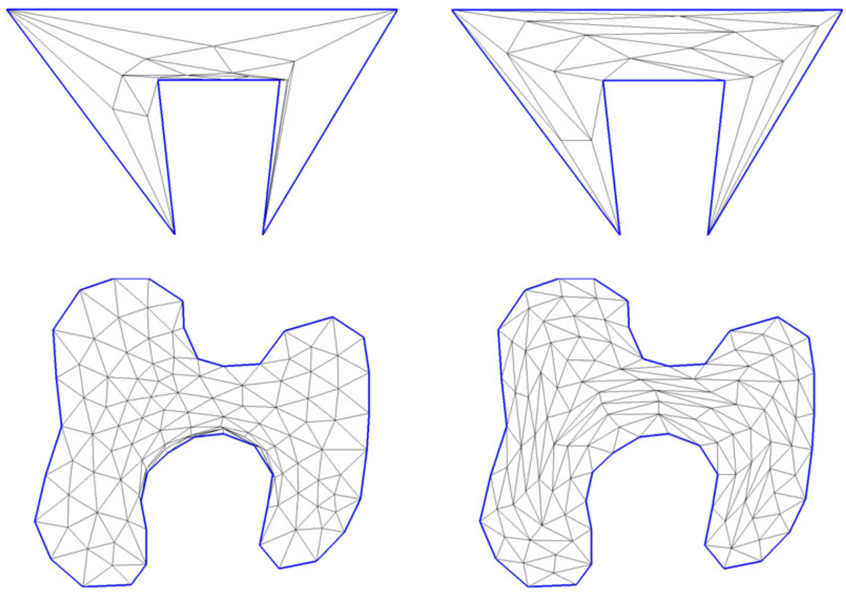

Fig. 6. Embeddings obtained using different Laplacians. Left: cotangent Laplacian. Right: weighted Laplacian. Note how the weighted Laplacian prevents "skinny" triangles.

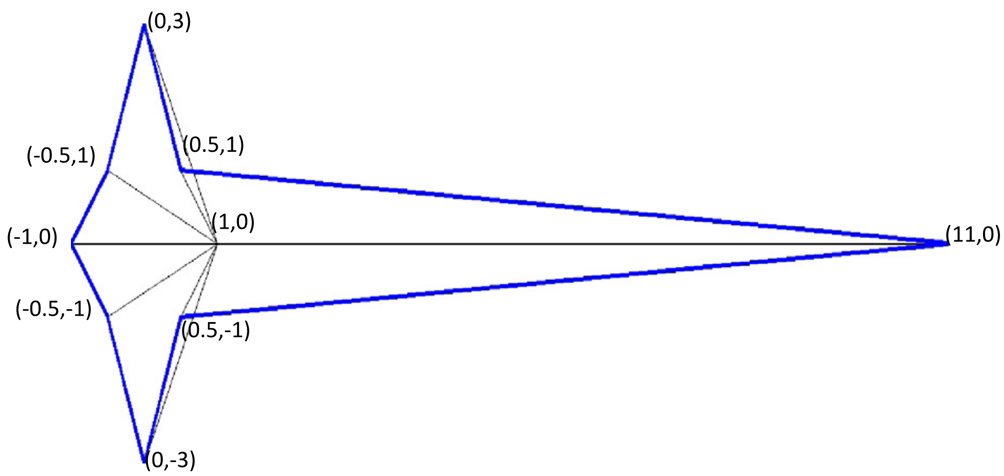

Fig. 7. Minimizing the variance of the areas of a one-ring of triangles may produce an invalid embedding.

compare its three edge lengths. One of them equal or close to the sum of another two would indicate degeneracy. This is a simpler test than examining the angles of the triangle.

5.3. Recognizing no valid embedding

The embedding algorithm can also determine whether a valid embedding exists at all. Since our algorithm can only converge to a valid embedding, if it does not exist, the energy does not always decrease due to previous numerical issues and its value becomes unstable. At this point the iteration may be aborted and the input rejected.

6. Conclusion and future work

This paper presents an algorithm for computing a valid embedding of a triangulated 3-connected planar graph given an embedding of its boundary, if it exists. We provide an improved version of the algorithm (using a "weighted Laplacian") to improve the quality of the resulting triangulation, although it is not always guaranteed to work for all boundary embeddings.

Our embedding algorithm is very similar to that proposed by Pinkall and Polthier (1993) for minimizing the surface area of a three-dimensional triangle mesh. This was recently generalized by dos Santos Crissaff (2009) to a variety of other "geometric" energies, which enable a variety of effects.

An important problem in computational geometry is generating so-called *compatible triangulations* of two planar regions bounded by different polygons, given a correspondence between their boundary vertices (Surazhsky and Gotsman, 2001, 2004). Usually, interior (so-called *Steiner* vertices) must be generated to support the triangulation, and the typical objective is to minimize their number. One way to solve this problem, based on our embedding algorithm, is to randomly generate a triangulation of one polygon with a small number of Steiner vertices, and then try to embed the same triangle graph structure with the second boundary polygon. A probabilistic algorithm could try this until success.

Future work might focus on extensions to three dimensions, given a tetrahedral graph. This is complicated by the fact that even Tutte's theorem does not hold for a convex boundary embedding in the three-dimensional case (Colin de Verdière et al., 2001, Floater and Pham-Trong, 2006). It is also difficult to prove that an embedding algorithm analogous to the one described here (using the analogous geometric Laplacian (Wang et al., 2004)) will always reduce the unsigned volume during iteration.

Acknowledgements

The work of L. Liu was partially supported by the joint grant of the National Natural Science Foundation of China and Microsoft Research Asia (60776799). R. Chen is partially supported by the Ali Kaufmann postdoctoral fellowship at the Technion.

Appendix A

Lemma 1 *(The gradient property). The gradient of $U(T)$ is equal to LX, namely: $\frac{\partial U(T)}{\partial X_i} = (LX)_i$, for any vertex position X_i in T.*

Proof. According to the definition of the cotangent Laplacian,

$$(LX)_i = \frac{1}{2}\sum_{j\in VN(i)} (\cot\alpha_{ij} + \cot\alpha_{ji})(X_i - X_j)$$

$$= \sum_{j\in TN(i)} \frac{1}{4}\left(\frac{(X_i - X_{t,k})\cdot(X_{t,j} - X_{t,k})}{|A(t)|}(X_i - X_{t,j}) + \frac{-(X_{t,j} - X_{t,k})\cdot(X_i - X_{t,j})}{|A(t)|}(X_i - X_{t,k})\right)$$

Using the vector triple product identity:

$$(X\cdot Z)Y - (X\cdot Y)Z = X\times(Y\times Z)$$

we simplify this to:

$$(LX)_i = \sum_{j\in TN(i)} \frac{(X_{t,j} - X_{t,k})\times[(X_i - X_{t,k})\times(X_i - X_{t,j})]}{4|A(t)|}$$

The fact that, for 2D vectors, $X\times(Y\times Z) = -R_{\pi/2}[(Y\times Z)_z X]$ (V_z is the z component of vector V) and the z component of $(X_i - X_{t,k})\times(X_i - X_{t,j})$ is no other than $2A(t)$, further simplifies to:

$$= \sum_{t\in TN(i)} \frac{2A(t)R_{\pi/2}(X_{t,k} - X_{t,j})}{4|A(t)|}$$

$$= \frac{1}{2}\sum_{t\in TN(i)} \operatorname{sgn}\big(A(t)\big)R_{\pi/2}(X_{t,k} - X_{t,j})$$

Combined with (4) in the proof of Theorem 2, one easily obtains $\frac{\partial U(T)}{\partial x_i} = (L\cdot X)_i$. □

References

Colin de Verdière, E., Pocchiola, M., Vegter, G., 2001. Tutte's barycenter method applied to isotopies. Computational Geometry: Theory and Applications 26, 81–97.
dos Santos Crissaff, L., 2009. Geometric energies on triangulated surfaces. Ph.D thesis, PUC Rio de Janeiro.
Floater, M.S., 1997. Parametrization and smooth approximation of surface triangulations. Comp. Aided Geom. Design 14, 231–250.
Floater, M.S., Pham-Trong, V., 2006. Convex combination maps over triangulations, tilings, and tetrahedral meshes. Advances in Computational Mathematics 25 (4), 347–356.
Hong, A.H., Nagamochi, H., 2006. Convex drawings of graphs with non-convex boundary. LNCS 4271, 113–124.
Hormann, K., Floater, M.S., 2006. Mean value coordinates for arbitrary planar polygons. ACM Transactions on Graphics 25, 1424–1441.
Pinkall, U., Polthier, K., 1993. Computing discrete minimal surfaces and their conjugates. Experimental Mathematics 2, 15–36.
Surazhsky, V., Gotsman, C., 2001. Controllable morphing of compatible planar triangulations. ACM Transactions on Graphics 20 (4), 203–231.
Surazhsky, V., Gotsman, C., 2004. High quality compatible triangulations. Engineering with Computers 20 (2), 147–156.
Tutte, W.T., 1963. How to draw a graph. Proc. London Math. Soc. 13, 743–768.
Wang, Y., Gu, X., Yau, S.-T., 2004. Volumetric harmonic map. Communications in Information and Systems 3 (3), 191–202.