A UNIFORM TRIANGLE MESH GENERATION OF CURVED SURFACES

Naoki KATOH[1*]

Makoto OHSAKI[2]

Yinfeng XU[3]

[1]Department of Architecture and Architectural Engineering, Kyoto University, Japan
[2]Department of Precision Engineering, Kyoto University, Japan
[3]School of Management, Xi'an Jiaotong University, P.R. China

Abstract:

The purpose of this study is to maximize the number of edges with almost equal lengths for a triangular mesh of a curved surface, so that the distortion of the triangulation used for finite element model or architectural latticed roofs is to be minimized. An algorithm is first presented for triangulation of a convex polygon whose edge lengths are almost equal. A domain on a parametric surface is mapped to a plane to obtain a polygon. The proposed algorithm is applied to the polygon to find a triangulation, which is remapped to the surface. The performance of the proposed method is demonstrated by using a tensor product Bézier surface.

Keywords: Triangulation, Parametric Surface

1. introduction

It is well known that the performance of Finite Element Method (FEM) strongly depends on the quality of the meshes. Therefore, numerous number of researches have been carried out for minimizing distortion of the meshes.

In the field of architectural design, triangular meshes are often used for latticed domes covering large space. For such structures, the lengths of edges (members) and the angles between the adjacent edges connected to a node are critical properties that determine the structural performances. There have been many studies on triangulation of convex polygons. However, few methods can be found for curved surfaces. In the field, from several viewpoints, it is desired to produce a triangulation whose edge lengths are uniform. In particular, it is important to reduce the different number of edge lengths. From this standpoint, the paper is concerned with producing triangulations with almost equal lengths. In this study, in particular, we shall first give two algorithms for triangulating a convex polygon in the plane which produce edges whose lengths are between $\underline{l}$ and $2\underline{l}$, where $\underline{l}$ is a predetermined value. The first algorithm guarantees that the number of nodes used in the triangulation obtained by the proposed algorithm is at most $16/3$ times the one which is minimum possible among all triangulations satisfying that edge lengths are between $\underline{l}$ and $2\underline{l}$. The second one poduces a triangulation in which most of edge lengths is $\underline{l}$ while the other edge lengths are at most $2\underline{l}$. Based on the second algorithm, an algorithm is presented for triangulating a surface that maximizes the number of edges with almost equal lengths. This is done as follows: (1) map the domain of a surface to a plane, (2) find triangulation by using the proposed algorithm, (3) remap the polygon to the surface.

2. Problem Formulation

[1]Kyoto University, Kyoto University Katsura, Nishikyo, Kyoto 615-8510, Japan, Fax: +81-75-383-2939, naoki@archi.kyoto-u.ac.jp

[2]Kyoto University, Kyoto University Katsura, Nishikyo, Kyoto 615-8510, Japan, Fax: +81-75-383-2901, ohsaki@archi.kyoto-u.ac.jp

[3]School of Management, Xi'an Jiaotong University Xi'an, 710049 P.R.China, yfxu@mail.xjtu.edu.cn

Suppose we are given a convex polygon $\boldsymbol{P}$ and vertex set V of $\boldsymbol{P}$. We assume that every inner angle of $\boldsymbol{P}$ is at least $90°$, and that the length of every edge of $\boldsymbol{P}$ is between $\underline{l}$ and $2\underline{l}$. For points u, v, let $d(u, v)$ denote the Euclidean distance between u and v. For a point set X, Y, let $d(X, Y) = \min\{d(x, y) \mid x \in X, y \in Y, x \neq y\}$. S denotes the point set which we are going to place inside $\boldsymbol{P}$. From the above assumption, $d(V, V) \geq \underline{l}$ holds.

$$\begin{array}{rl} \text{Problem 1:} \quad \min_{S \subset \boldsymbol{P}} & |S| \\ \text{subject to} & \exists \text{ triangulation } T \in \mathcal{T} \\ & \underline{l} \leq l(e) \leq 2\underline{l} \text{ for } e \in T \end{array}$$

Here, $\mathcal{T}$ denotes the set of all triangulation for $S \cup V$. Incremental Voronoi insertion algorithm places a point one by one whose distance is farthest from the current point set.

Algorithm INCREMENT
Step 1 Let $S := \emptyset$.
Step 2 Place points V' on the boundary of $\mathbf{P}$ so that $\underline{l} \leq d(V \cup V', V \cup V') \leq 2\underline{l}$ holds.
Step 3 Find a point $p^* = x$ such that $d(x, V \cup S)$ is largest inside $\boldsymbol{P}$. For this, we use Voronoi diagram $\mathrm{Vor}(V \cup S)$.
Step 4 If $d(p^*, S \cup V') \geq \underline{l}$, let $S := S \cup \{p^*\}$ and return to Step 3. Else return to Step 5.
Step 5 Output Delaunay triangulation for $S \cup V' \cup V$.

Regarding Step 1, from $d(V, V) \geq 2\underline{l}$, and since every inner angle of $\boldsymbol{P}$ is at least $90°$, we can place the points on the boundary of $\boldsymbol{P}$ so that the distance between consecutive points on B is between $\underline{l}$ and $2\underline{l}$.

Theorem 1 *Let T' denote the triangulation obtained by the algorithm. Then, the length of every edge is between $\underline{l}$ and $2\underline{l}$.*

proof. Let S' denote the point set which is a union of points placed in Steps 2 and 4. Let us consider the delaunay triangulation $\mathrm{DT}(S' \cup V)$. Let us classify a triangle Δ of $\mathrm{DT}(S' \cup V)$ as either *critical* or *non-critical* depending on whether the Voronoi vertex dual to Δ (which is the circumcenter of Δ) lies outside of the polygon $\mathbf{P}$ or not. Whereas edges of critical triangles can be arbitrarily long, edge lengths are bounded in non-critical triangles.

Lemma 1 *No edge e of a non-critical triangle Δ of $DT(S' \cup V)$ is longer than $2\underline{l}$.*

proof. Let $e = (p, q)$ and denote with x the Voronoi vertex dual to Δ. Recall that we halted the algorithm because there is no point y whose distance to the existing point set is at least $\underline{l}$. Thus, $d(x, S' \cup V) < \underline{l}$ holds. As x lies inside of $\mathbf{P}$, we get $l(x, p) = l(x, q)$ and $l(x, p) < \underline{l}$. The triangle inequality now implies $l(p, q) \leq 2\underline{l}$. ◻

Our next observation is on critical triangles. Consider some edge e of $\mathrm{DT}(S' \cup V)$ on the boundary of $\mathbf{P}$. Edge e cuts off some part of the Voronoi diagram $\mathrm{Vor}(S' \cup V)$ that lies outside of $\mathbf{P}$. If that part contains Voronoi vertices then we define the *critical region*, $R(e)$, for e as the union of all the (critical) triangles that are dual to these vertices. It is not hard to see that each critical triangle of $\mathrm{DT}(S' \cup V)$ belongs to a unique critical region.

Lemma 2 *No edge f of a critical triangle in $R(e)$ is longer than e.*

proof. Let a and a' be the end points of e, and let p and p' be the end points of f. Then consider the point x (resp. x') which is at the intersection of e and the line segment connecting the circumcenter and p (resp. p'). $d(x, p) < d(x, a)$ holds from the definition of the Voronoi diagram. Similarly, $d(x', p') < d(x', a')$ holds. Thus, $d(x, p) + d(x', p') < d(x, a) + d(x', a')$ follows. Thus,

$$d(x, p) + d(x, x') + d(x', p') < d(x, a) + d(x, x') + d(x', a') = d(a, a')$$

holds. Since $d(p, p') < d(x, p) + d(x, x') + d(x', p')$ holds from the triangle inequality, $d(p, p') < d(a, a')$ follows. ◻

□

Let T^* denote an optimal solution of Problem 1. Let $S^* \cup V$ and E^* denote the set of vertices and the set of edges in T^*, and $S' \cup V$ and E' denote the set of vertices and the set of edges in T'.

Let $\#n$ and $\#n'$ denote the number of vertices in T^* and T', respectively.

Theorem 2 $$\#n' \leq \frac{16}{3}\#n.$$

proof. Suppose the theorem is not true. Let $r = \frac{2}{\sqrt{3}}\underline{l}$. For each point $p \in S^* \cup V$ draw a circle with radius r around p. Let $\mathcal{C}^*$ denote the resulting set of circles. For each triangle Δ of T^* its area is entirely covered by the circles of $\mathcal{C}^*$ centered at its three endpoints. So $\mathcal{C}^*$ entirely covers the area of $\mathbf{P}$.

Next, consider the solution T' obtained by the above algorithm. Again, around each point in $S' \cup V$ draw a circle with radius $r' = \underline{l}/2$. Let $\mathcal{C}'$ be the resulting set of circles. Circles in $\mathcal{C}'$ do not overlap each other since $d(S' \cup V, S' \cup V) \geq \underline{l}$. So $\mathcal{C}'$ does not entirely cover the area of $\mathbf{P}$.

We now analyze what happens outside of $\mathbf{P}$. Let Q be the convex hull of $\mathcal{C}^*$. Consider an arbitrary edge e of $\mathbf{P}$, and let $R(e)$ denote the rectangle spanned by e and the boundary edge of Q parallel to e. Since $\mathbf{P}$ is convex, the rectangles that arise in this way from all the edges of $\mathbf{P}$ are mutually disjoint. Let $\mathcal{R}$ deonte the set of such rectangles. For an original vertex $v \in V$, the area of the circle in $\mathcal{C}^*$ whose center is v is clearly larger than that of the circle in $\mathcal{C}'$ whose center is v because of $r > r'$. Thus, the area covered by circles in $\mathcal{C}^*$ which is outside $\mathbf{P} \cup \mathcal{R}$ is larger than that in $\mathcal{C}'$ which is outside $\mathbf{P} \cup \mathcal{R}$. Let R^* denote the area of $R(e)$ covered by circles of $\mathcal{C}^*$, and let R' denote the area of $R(e)$ covered by circles of $\mathcal{C}'$. It can be proven in the same manner as in [1] that $R' < R^*$ holds. Then $\mathcal{C}^*$ covers less area outside of $\mathbf{P}$ than $\mathcal{C}^*$ does. In conjunction with the observations above, concerning the interior of $\mathbf{P}$, the *total* area covered by $\mathcal{C}'$ is now less than the total area covered by $\mathcal{C}^*$. But this is a contradiction because $n' > \frac{16}{3}n$ implies that the area covered by $\mathcal{C}'$ is larger than that by $\mathcal{C}^*$. □

3. Second Problem

Given the same input as in Problem 1, we consider the problem of finding a triangulation T such that the number of different edge lengths used in T is minimized under the constraint that $\underline{l} \leq d(e) \leq 2\underline{l}$ for any edge $e \in T$. We assume that for any boundary edge e on $\boldsymbol{P}$,

$$\underline{l} \leq d(e) \leq \alpha \underline{l}$$

holds, where $\alpha = 2$. Notice that depending on the shape of $\boldsymbol{P}$, the number of edges required to satisfy the constraint becomes $O(|V|^2)$. We shall give an algorithm which produces a triangulation T such that the number of different edge lengths is $O(|V|)$. Let $V(\mathbf{P})$ and $E(\mathbf{P})$ denote the set of vertices and edges of $\mathbf{P}$, respectively. For the easeness of exposition, we assume $\underline{l} = 1$ hereafter.

Let us overlay a triangular grid $\mathcal{G}$ with unit edge length on $\boldsymbol{P}$. Let V_G and E_G be the set of edges of $\mathcal{G}$ which are entirely contained in $\boldsymbol{P}$. Let H denote the plane graph composed of V_G and E_G.

Fact 1 Since $\boldsymbol{P}$ is convex, the plane graph H is connected.

Let P' be the set of vertices v of H such that $d(v, V(\mathbf{P})) < 1$. Let H' be the subgraph of H obtained by removing vertices of P' together with incident edges.

The plane graph H' may not be connected. Let the boundary vertices and edges of H' be denoted by V'_B and E'_B, respectively.

Now consider the graph $G = (V(\mathbf{P}) \cup V'_B, E(\mathbf{P}) \cup E'_B)$.

Lemma 3 *(1) The length between any pair of vertices of G is at least 1.*

(2) For any point p of an edge of G, the vertex nearest to p is one of end vertices of G.

Now we consider the problem of triangulating the region which is inside of $\mathbf{P}$ and outside of H' such that every edge length is between 1 and 2. Let $\mathcal{R}$ denote this region. For this, we first apply the Voronoi insertion algorithm with the minimum edge length equal to 1.

Although the original Voronoi insertion algorithm is developed for a convex polygon, it works also for non-convex region. Actually, in the same manner as in the previous section, at every iteration, the algorithm finds the point inside the region $\mathcal{R}$ which is farthest from the current point set, and adds it to the point set. We repeat this iteration as long as the distance between the new point and the current point set is at least 1. The new feature in this algorithm is in the definition of the distance. Here for points $u, v \in \mathcal{R}$, $\tilde{d}(u, v)$ is defined to be

$$\tilde{d}(u, v) = \begin{cases} d(u, v) & \text{if } u \text{ is visible from } v \\ \infty & \text{otherwise} \end{cases}$$

Once the distance between the new point and the current point set gets smaller than 1, the algorithm outputs the constrained Delaunay triangulation, where the edges of $\mathcal{R}$ are treated as constraints (in other words obstacles). The constrained Delaunay triangulation is defined as follows.

Definition 1 The *constrained Delaunay triangulation* (CDT) contains the edge (a, b) between two input points, if and only if a is visible to b, and some circle through a and b contains no input point c visible to segment ab.

This definition implies that the circumcircle of a triangle abc in the CDT cannot contain an input point other than a, b or c, which is visible from the interior of abc.

Now we shall prove the following theorem which is a counterpart of Theorem 1 for the non-convex region.

Theorem 3 *Let T' denote the triangulation obtained by the algorithm. Then, the length of every edge is between $\underline{l}$ and $2\underline{l}$.*

proof. The proof is done in the same manner as that of Theorem 1.

Let S' denote the point set which is a union of points placed in Steps 2 and 4. Let us consider the constrained delaunay triangulation $\mathrm{DT}(S' \cup V)$. Let us classify a triangle Δ of $\mathrm{DT}(S' \cup V)$ as either *critical* or *non-critical* depending on whether the circumcenter x of Δ is visible from three vertices of Δ. We first claim that all three vertices of Δ are visible from x, or none three vertices are visible from x. Let a, b and c denote the three vertices. Suppose that a vertex a is visible from x while the other vertex b is not. Then, a is not visible from b, or there exists a point in the circumcircle other than a, b and c. In any case, this violates the definition of the constrained Delaunay triangulation. Thus, the claim was proved.

We can show that Lemma 4 holds for a non-cirtical triangle, and that Lemma 5 holds for a cirtical triangle. The proofs are done in essentially the same manner as those given in Section 2.

$\square$

Notice that the number of points inserted by the Voronoi insertion algorithm is $O(|V|)$ because $\mathcal{R}$ is covered by a union of circles with radius $3\underline{l}$ such that centers are vertices of $\mathbf{P}$, and inside a circle with radius $3\underline{l}$ there can exist a constant number of points whose inter-distance is at least $\underline{l}$. Thus, it follows that the number of edges in the triangulation for $\mathcal{R}$ is $O(|V|)$.

4. Triangulation of a parametric surface.

Consider a surface defined by parameters (u, v), where $u \in [0, 1]$, $v \in [0, 1]$. Triangulation of surface is carried out as follows:

1. Define the surface $\mathbf{X}^s(u, v) \in R^3$ using, e.g. tensor product Bé zier surface. Divide the parameter plane (u, v) into a $n_u \times n_v$ grid, and further divide each rectangular unit to two right triangles to obtain an *auxiliary triangulation*. The coordinate of node i in the parametric space is denoted by $\mathbf{u}_i^0 \in R^2$.

2. Let $\mathbf{X}_i^{s0} \in R^3$ denote the coordinate of node i on the surface. The length of edge j is denoted by L_j^{s0}. Map $\mathbf{X}_i^{s0}$ to $\mathbf{X}_i^{p0} \in R^2$ on a plane so that the following energy E of distortion is minimized:

$$E = \frac{1}{2} \sum_{j=1}^{n_e} \left(\frac{L_j^{p0} - L_j^{s0}}{L_j^{s0}} \right) \tag{1}$$

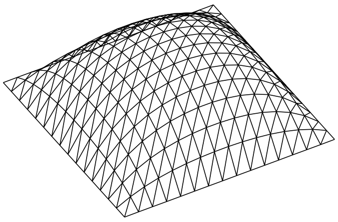

Fig 1. Auxiliary triangulation of the surface.

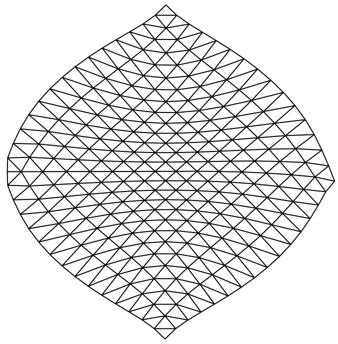

Fig 2. Auxiliary triangulation of the plane.

	no. of members	mean	min.	max.	std. dev.
initial surf.	645	6.8173	5.0000	10.0000	1.6711
plane (G1)	774	5.0000	5.0000	5.0000	0.0000
plane (total)	954	5.3633	4.5893	9.9727	0.9888
final surf. (G1)	774	5.0775	4.2043	6.1972	0.2780
final surf. (total)	954	5.4280	4.2043	10.6405	1.0605
final opt. (G1)	774	4.9597	4.3490	5.3982	0.1915
final opt (total)	954	5.4765	4.3490	13.3777	1.4842

Table 1 Results of triangulation

where L_j^{p0} is the length of edge j on the plane, and n_{e0} is the number of edges.

3. Extract the boundary of the graph defined by $\mathbf{X}_i^{p0}$, and find triangulation using the approach presented in the previous section to obtain the nodal coordinates $\mathbf{X}_i^p \in R^2$.

4. Suppose node i is included in auxiliary triangle k on the plane defined by the nodes q, r, s. $\mathbf{X}_i^p$ can be written by the following interpolation

$$\mathbf{X}_i^p = \mathbf{X}_q^{p0} + \alpha(\mathbf{X}_r^{p0} - \mathbf{X}_q^{p0}) + \beta(\mathbf{X}_s^{p0} - \mathbf{X}_q^{p0}) \tag{2}$$

Then the parameter $\mathbf{u}_i = (u_i, v_i)$ for $\mathbf{X}_i^p$ is approximated by

$$\mathbf{u}_i = \mathbf{u}_q^0 + \alpha(\mathbf{u}_r^0 - \mathbf{u}_q^0) + \beta(\mathbf{u}_s^0 - \mathbf{u}_q^0) \tag{3}$$

The location of node i on the original surface is found from the parameter value as $\mathbf{X}_i = \mathbf{X}^s(u_i, v_i)$

Consider a surface as shown in Fig. 2. divided into 15×15 grid. The mapping to the plane is as shown in Fig. 3.. Note that the boundary of the auxiliary triangulation on the plane is not convex. The algorithm proposed in Section 2 is applied to obtain the equal-length triangulation as shown in Fig. 4., where the number of edges is 954, and the lengths of 774 members are equal to 5. The final triangulation on the original surface is as shown in Fig. 5..

The mean, maximum and minimum length as well as the standard deviation D of the edge lengths are listed in Table 1. Let G1 denote the group of edges that have the same length on the plane. It can be observed from Table 1 that the value of D in G1 can be decreased by using the proposed algorithm.

Finally, the deviation of the edge lengths in G1 from the target value is minimized by considering (u_i, v_i) of inner nodes as variables. The results are as shown in the last row of Table 1. It is seen from this result that the difference between the maximum and minimum edge lengths can be drastically reduced by optimizing the parameter values of the nodes.

References

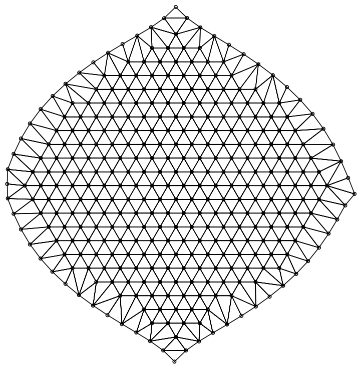

Fig 3. Triangulation on the plane.

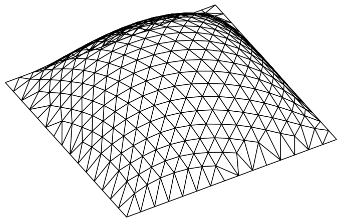

Fig 4. Remapping of triangulation to the surface.

[1] F. Aurenhammer, N. Katoh, H. Kojima, M. Ohsaki and Y-F. Xu, Approximating uniform triangular meshes in polygons, Theoretical Computer Science, Vol. 289, pp. 879-895, 2002.