

異なる辺長種類数制約下における一様 三角形メッシュ生成アルゴリズム

谷川眞一 加藤直樹
京都大学大学院 工学研究科 建築学専攻

概要

平面閉領域内部における Steiner 点を用いた三角形分割問題は、計算幾何学における主要な問題のうちの 1 つである。本論では特に単純多角形内部を異なる辺長種類数を最小化するような三角形分割に着目する。異なる辺長種類数をできるだけ少なく抑えるとともに、辺の下限値を与えることで辺長がほぼ一様となるような三角形分割を出力するアルゴリズムを提案し、アルゴリズムの理論的精度を示すとともに、コンピュータ上に実装して実験および性能評価を行う。

Finding a triangular mesh with a constant number of different edge lengths

Shin-ichi Tanigawa Naoki Katoh
Department of Architecture and Engineering, Kyoto University

Abstract

Generating triangular meshes using some Steiner points is one of the fundamental problems in computational geometry. We are interested in the problem of triangulating a simple polygon with a small number of different edge lengths. An algorithm is presented for finding almost uniform triangular mesh with a constant number of different edge lengths. We give a theoretical guarantee and experimental evaluation of this algorithm.

1 はじめに

三角形分割は、コンピュータグラフィックスや有限要素法におけるメッシュ生成など様々な応用があり、その目的に応じてアルゴリズムの研究がなされてきた [2, 3, 4, 7, 5, 8]。建築の分野に関して言えば、三角形メッシュは大空間を覆うドームなどの構造トラスなどにしばしば用いられている。そのような大規模構造物においては、十分な外力に耐えうるために棒部材の長さの比や接合部における部材間の角度が重要であるとともに、実現に至るためのコストも十分に評価されなければならない。

文献 [3] においては、球面や曲面トラス材への応用を目指し、最大辺長/最小辺長の最小化、辺長分散最小化を目的関数として球面上へ適用可能なアルゴリズムが提案されている。しかし建築への現実的応用を考えた際、上述したように建築コストもまた重要な評価基準であり、標準規格からの部材長さのずれは生産、再利用の観点からも大きな問題となる。そのため異なる

辺長種類数最小化 (又は、同じ長さを持つ辺の種類数最大化) に着目した三角形メッシュ生成法が必要とされる。この問題は計算幾何学の分野においてもまだほとんど取り扱われていない問題であり、これまでの研究としては [4] のみである。

2 アルゴリズムの概要

頂点集合 V_p からなる多角形 P と辺の下限値 l が与えられているものとする。この時、多角形 P 内部を異なる辺長種類数ができるだけ少なくなるような三角形分割を考えるのだが、そのために幅 αl で与えられた正方形グリッド G を導入する。幅 αl で与えられた正方形グリッドの格子点の集合を V_G とした時、まず初めに与えられた多角形 P を、頂点がすべて V_G の要素であるような多角形 Q へと近似する。次に Steiner 点をこの正方形グリッドの格子点上に逐次的に配置していき、最終的に Delaunay 三角形分割を出力する。出力さ

れる三角形分割の頂点は境界上の点も含めすべて正方形グリッド上に存在するため、すべての辺長はグリッド幅 αl の整数倍の組み合わせとなる。

本論で提案する三角形メッシュ生成のアルゴリズムは [2, 3, 4] で用いられている逐次ボロノイ分割 (canonical Voronoi insertion) を拡張したものである。そこで [2, 3, 4] による幾つかの定義を用いてアルゴリズムの精度評価を行う。

主な結果として、このアルゴリズムによって出力される三角形分割はすべての辺長が l 以上 $(2 + \sqrt{2}\alpha)l$ 以下であり、すべての角度が $\arcsin\left(\frac{1}{2 + \sqrt{2}\alpha}\right)$ 以上、また異なる辺長種類数は $O(\frac{1}{\alpha})$ となる。そのため [4] におけるアルゴリズムでは、異なる辺長の種類数は $O(|V_G|)$ であり、境界の頂点数に大きく依存するのに対し、本論のアルゴリズムでは正方形グリッドの幅のみに依存し境界の頂点数や形状に依存しない。

3 境界多角形近似

上述したように本論文で紹介するアルゴリズムは、多角形で与えられた境界を正方形グリッド上の頂点からなる多角形に近似することで実現する。そのため多角形近似をおこなう際に伴う誤差を最小化することが重要となる。

多角形近似は、コンピュータービジョンやパターン認識、アニメーションなど様々な分野への応用が可能なテーマとして盛んに研究がなされており、その目的に応じて近似精度を計る評価尺度がいくつか提案されている。本論では計算幾何学の分野などで最も一般的に用いられている、Hausdorff Distance および 2 つの物体の間に生じる面積の誤差を評価尺度として用いる。

Hausdorff Distance は点や線、多角形などの集合において集合間の距離を測る尺度として用いられる。 a_i, b_i を点や線分などの要素とし、集合 $A = \{a_1, a_2, \dots, a_m\}$ と $B = \{b_1, b_2, \dots, b_n\}$ について、 $A = \cup_{i=1}^m a_i, B = \cup_{i=1}^n b_i$ とした時、集合 A, B における Hausdorff Distance は次のように定義される。

$$H(A, B) = \max(h(A, B), h(B, A))$$

ここで

$$h(A, B) = \sup_{\alpha \in A} \inf_{\beta \in B} d(\alpha, \beta)$$

文献 [1] によると頂点数 m, n の 2 つの多角形 A, B において A, B 間の Hausdorff Distance は $O((m+n) \log(m+n))$ で求めることができる。

ここでまず多角形近似における問題の定式化をしておきたい。入力として与えられた単純な多角形を P 、出力の近似多角形を Q とし、それぞれの各頂点を枝

の一定順序に $\{a_1, a_2, \dots, a_m\}, \{b_1, b_2, \dots, b_n\}$ とおき、 P, Q 間の誤差を $W(Q)$ とする。また幅 αl の正方形グリッドの格子点の集合を V_G とする。このとき問題は以下のように定式化される。

$$\begin{aligned} \min & \quad W(Q) \\ \text{subject to} & \quad b_i \in V_G \\ & \quad l \leq d(b_i, b_{i-1}) \leq \beta l \end{aligned}$$

境界近似アルゴリズムの流れとしては、まず解となりうる候補点を列挙、次にそれらの候補点を頂点とするネットワークの形成を行い、誤差が最小となる最適パスを求めることで新たな境界を決定する。

候補となるグリッド頂点の列挙 制約条件 $b_i \in V_G$ のために解となりうる頂点は離散的に存在するが、目的は誤差最小化であるため解となりうる候補点がある程度限定することができ、またこの事は計算時間の効率化のためにも必要なことである。

解となりうる候補点の集合を C とした時、 C を次のように定める。

$$C = \{p \mid d(p, P) \leq \epsilon, p \in V_G\}$$

これは P の各辺を幅 ϵ だけ持たした領域内に含まれるグリッド点の集合である。

ϵ が大きすぎると計算時間が大きくなるが、小さすぎると正確な解が得られない可能性がある。 ϵ の値に関しては、計算時間の算定の際後述する。

今 i をある整数とした時、直線 $l: x = i\alpha l$ 上の候補点を列挙することを考える。ここで多角形 P 上のすべての点 (線分) を母点とするボロノイ図を $Vor(P)$ と記す。直線 l 上にある点が y 座標値をパラメーターとして移動する際この点と多角形 P との最短距離は、 $Vor(P)$ と直線 l の交点においてのみ極値をとる。つまり同じボロノイ領域内部においてでは直線 l 上の点と P との距離は単調に変化する。よってこのような交点から y 座標値が大きい順または小さい順に l 上のグリッド点を探索し、 P までの距離が ϵ を上回るかまたはボロノイ領域の外部に出るまで調べることで、 C の点を列挙することができる。

幾何情報をもつネットワークの生成 次に先ほど列挙された候補点をノードとするネットワークの生成を行う。生成されるネットワークの頂点の集合を V 、枝の集合を E とする。ネットワークを構成するための基本操作は次のようになる。まず候補点集合 C の各点とネットワークのノードを 1 対 1 に対応させる。次にある候補点 p のデータを持つノード n_p と候補点 q のデータを持つノード n_q に関して、 $l \leq d(p, q) \leq \beta l$ ならば n_p と n_q の間に枝を結ぶ。また枝の重みは点 p と

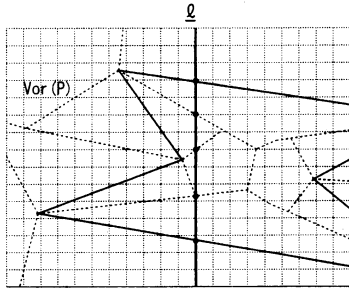


図 1: 直線 l 上の候補点列挙



図 2: 辺 pq と多角形 P の面積差

q を線分で結んだ際に多角形 P との間に生じる誤差となる。面積差の和を最小化する時、線分 pq と多角形 P の誤差は図 2 の色塗り部分とする。ここで図 2 中の $ref(p)$ は以下のように定める。

$ref(p)$: 多角形 P の辺上の点の内、点 p から最も近い点

これは先に述べた候補点を列挙する際に同時に求めておくことができる。よって $ref(p)$ および $ref(q)$ 、線分 pq と多角形 P の交点を用いて図 2 の面積は容易に求めることができる。

Hausdorff Distance 最小化の場合の誤差は図 3 のようになる。



図 3: 線分 pq の誤差 (Hausdorff Distance)

最短パスを見つける際、始点と終点が必要であり、また解は幾何学的に閉じていなくてはならない。それを解決するために解となる多角形の辺の長さは l 以上 βl 以下でなければならないことを利用する。 P の適当な頂点を中心として半径 $3l$ の内部に少なくとも 1 点は最適解の頂点が存在するので、この円の内部に含まれる候補点の集合を始点集合 V_{start} とする。また終点は $n_{start} \in V_{start}$ のノードを複製することで、 n_{start} からこれを複製した点までの最短パスを求めればよい。

出力を凸多角形に制限した場合 入力が凸多角形であった場合、出力も凸多角形となることが望ましい場合が考えられる。そこでここでは出力が凸多角形に限定されるようなネットワークの生成について考える。

多角形の凸条件は、2 頂点では判別できないためネットワークに改良が必要となる。前述した方法では候補点とネットワークのノードを対応させていたが、ここではノード集合 N を導入し、候補点 p とノード集合 N_p を対応させる。ある候補点 p および q 間の距離が l 以上 βl 以下ならば、対応するノード集合 N_p および N_q に新たなノード n_{pq} および n_{qp} を追加しそれらを繋ぐ。そして最後にノード集合内部の各ノードをそれと接続するノードとの間で凸条件を満たすならば重み 0 の枝で結ぶ。

Step1 始点と、ダミーの終点の集合を定める。

Step2 すべての候補点 $p \in C$ に対して、 p のデータを持つノード集合 N_p を用意する。

Step3 すべての候補点 $p \in C$ に対して、 $l \leq d(p, q) \leq \beta l$ となる候補点 q をみつけ、このとき新たなノード n_{pq}, n_{qp} をそれぞれ N_p, N_q に加え n_{pq} と n_{qp} を枝で結ぶ。

Step4 すべてのノード集合 N_p に対して、この集合のある要素 n_{pq} と n_{pr} がそれぞれ接続しているもう一方の頂点 q, r と頂点 p の三点が q, p, r の順に結ばれた際、凸条件を満たすならば、 n_{pq} と n_{pr} の間に重さ 0 の枝を結ぶ。

この流れを図 4-7 に示す

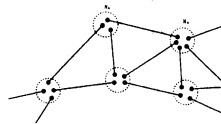


図 4: Step3 後

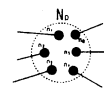


図 5: Step3 後のノード集合

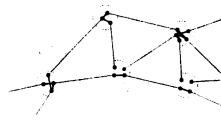


図 6: Step4 後

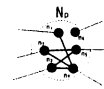


図 7: Step4 後のノード集合

境界近似の計算時間 計算時間の算出のために以下の補題を用いる。

補題 1 初期多角形を P 、Hausdorff Distace 最小化アルゴリズムによって得られる近似多角形を Q とした時、 $H(P, Q) \leq \sqrt{2}l$ である。

証明 幅 αl の正方形グリッドの格子点 V_G の要素を用いて、幅 l の正方形グリッドをつくることができる。このような正方形グリッドのうちの一つを G' 、格子点を $V_{G'}$ とする。 G' の正方形の内いくつかのものは P の辺によって切り取られる。このような正方形の集合を G'' とおくと、任意の $x \in V_{G''}$ について $h(x, P) \leq \sqrt{2}l$ である。また G'' は連結であるので、 $V_{G''}$ の要素を用いて必ず一辺が l の多角形をつくることができる。よって $\epsilon \geq \sqrt{2}l$ の時、 $V_{G''} \subset C$ であるので $H(P, Q^*) \leq \sqrt{2}l$ となる多角形 Q^* が必ず存在する。

□

今初期境界 P についてその頂点を V_p について、 $l \leq d(V_p, V_p) \leq \beta l$ であるものとする。(あらたな頂点を境界上に追加することによって、この条件を実現することは容易である。) また P の境界の長さを $Peri(P)$ とおく。

まず候補点集合 C について、この要素数 m を調べる。 C の各要素を中心に幅 αl の正方形を配置すると、この正方形の面積の総和は境界 P から ϵ だけ幅をもたした領域の面積とほぼ同程度である。具体的には次のような等式が成立する。

$$m(\alpha l)^2 < 2(\epsilon + \alpha l)(Peri(P) + \epsilon) \quad (1)$$

ここで $Peri(P) \leq 2l|V|$ である。 ϵ は Hausdorff Distance 最小化の際には補題 1 より $\sqrt{2}l$ でよいが、面積差最小化では最適解における Hausdorff Distance の上限は得られていない。 l の定数倍となることが予想されるが今後確かな証明が必要である。(以下では ϵ は l の定数倍で十分なものとして進める。) よって解の候補点数 m は式 1 を変形して、 $O(\frac{|V|}{\alpha^2})$ となる。ネットワーク形成について、半径 l の外部で半径 βl の内部となる領域に含まれる候補点の個数は $O(\frac{1}{\alpha^2})$ である。よって出力に制限がない時、ネットワーク形成に必要な時間は $O(\frac{m}{\alpha})$ 、つまり $O(\frac{|V|}{\alpha^2})$ となる。またネットワークの枝の個数も同様に $O(\frac{|V|}{\alpha^2})$ 。よって 1 つの始点が与えられた際 Dijkstra 法による計算時間は $O(\frac{|V|}{\alpha^2} + \frac{|V|}{\alpha^2} \log \frac{|V|}{\alpha^2})$ となる。始点集合 V_{start} の要素数は $O(\frac{1}{\alpha^2})$ であり、最悪の場合 V_{start} のすべての要素に対して Dijkstra 法を行うことから、全体にかかる計算時間は $O(\frac{|V|}{\alpha^2})(\frac{1}{\alpha^2} + \log \frac{|V|}{\alpha^2})$

出力を凸に制限した場合、ノード集合の個数は m であり、各候補点は半径 l の外部で半径 βl の内部となる領域に含まれる候補点を見つけるごとにノード集合内にノードを追加する。そのため全ノード数は $O(\frac{|V|}{\alpha^2})$ となる。また枝の個数は各ノード集合間を結ぶものが $O(\frac{|V|}{\alpha^2})$ 、ノード集合内のノードを結ぶ枝の個数は各ノード集合ごとに $O(\frac{1}{\alpha^2})$ であることから、ノード集合内を結ぶ枝の総数は $O(\frac{|V|}{\alpha^2})$ 。よって枝の総数も $O(\frac{|V|}{\alpha^2})$ となる。このことから Dijkstra 法一回あたりに必要な

計算時間は、 $O(\frac{|V|}{\alpha^2})(\frac{1}{\alpha^2} + \log \frac{|V|}{\alpha^2})$ である。すべての V_{start} の要素に Dijkstra 法を行うことから、全体の計算時間は $O(\frac{|V|}{\alpha^2})(\frac{1}{\alpha^2} + \log \frac{|V|}{\alpha^2})$

最適なグリッド配置 今まではグリッドを固定し、多角形の最適近似について考えてきた。しかし例えば図 89 に見られるようにグリッドの位置によって得られる最適解が異なってくる。

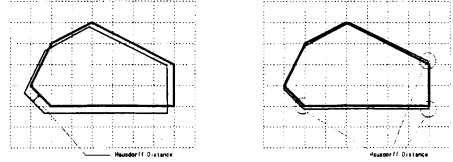


図 8: グリッドの移動前 図 9: グリッドの移動後

そこで多角形の形状が決まっている際のグリッドの移動方法、つまり二つの多角形の誤差最小化最適配置問題が考えられる。

2 つの多角形 P と Q が与えられている時、この問題は次のように定式化される。

$$\min H(P, I(Q))$$

ここで

$$I(\mathbf{x}) = \begin{pmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{pmatrix} \cdot \mathbf{x} + \mathbf{t}$$

ただし $\phi \in [0, 2\pi]$ 、 \mathbf{t} は線形変換。

この問題に関して H.Alt [1] によるアルゴリズムでは回転移動まで認めたとき計算時間は $O((mn)^4(m+n) \log(m+n))$ となる。

P.Chew(文献 [6]) によるアルゴリズムでは計算時間は $O((m+n)^6 \log^2 mn)$ で、H.Alt のアルゴリズムより改善されている。しかしそれでもかなりの計算時間が必要となる。またこれらのアルゴリズムで解を得たとしてもグリッドの移動によって近似多角形の形状の最適性の保証はなくなる。そのため形状の決定とグリッドの配置問題ともに考慮したアルゴリズムの提案は今後の課題である。

4 正方形グリッドを利用した三角形分割

多角形近似によって得られた新たな境界多角形を Q 、 Q を構成する頂点集合を V_Q 、 Q の内部に追加される点 (Steiner 点) の集合を S とする。またある頂点 p に関して、正方形グリッドのうちで p を含む正方形を構成する 4 つの頂点をそれぞれ p_1, p_2, p_3, p_4 と記す。

(グリッドの線分上に p が存在する時は、その線分に接するとの正方形でも良い。)

アルゴリズム

Step 1 集合 S を \emptyset とする。

Step 2 $V_Q \cup S$ のポロノイ図 $Vor(V_Q \cup S)$ を計算する。

Step 3 $Vor(V_Q \cup S)$ のポロノイ頂点を含みグリッドセルの頂点で、 P の内部にあるものの内、 $d(p_i^*, V_Q \cup S)$ が最大となる頂点 p_i^* をみつける。

Step 4 $d(p_i^*, V_Q \cup S) \geq l$ ならば $S := S \cup \{p_i^*\}$ とし Step 2 へ。

Step 5 $V_Q \cup S$ の Constrained Delaunay Triangulation を出力。

近似精度に入る前に得られた三角形の性質に応じて次のように三角形を分類する。出力によって得られたある三角形 Δ について、 Δ の外心点 x が Δ の3つの頂点から可視の時 non-critical triangle、外心点が Δ のすべての頂点から可視でないとき critical triangle、と定める。ここで注意しなければならないのは、 x から Δ のすべての頂点が可視である場合か、 x から Δ のどの頂点も可視でない場合のみしか起こらないということである。この事は Constrained Delaunay Triangulation の以下の定義による。

定義 辺 ab が Constrained Delaunay Triangulation の辺であるための必要十分条件は点 a が点 b から可視であり、かつ点 a, b を通り線分 ab から可視な点 c を含まないような円が存在することである。

補題 2 このアルゴリズムで得られた三角形分割のうちの non-critical triangle Δ の辺 e の辺長は $(2 + \sqrt{2}\alpha)l$ 以下であり、またこの上限値は最適である。

証明 Δ の各頂点を p, q, r 、外接円の半径を r 、中心を x とする。点 x を含むグリッドセルの4頂点 x_i ($i = 1, 2, 3, 4$) のうち x に最も近い点を x_1 とすると $d(x_1, x) \leq \frac{1}{\sqrt{2}}\alpha l$ が成り立つ。 p, q, r のいずれかの頂点を母点とする $(Vor(V_Q \cup S))$ のポロノイ領域に x_1 が含まれる時、 x_1 を含むポロノイ領域の母点が p であったとすると、アルゴリズムの終了条件より $d(x_1, p) \leq l$ が成り立つ。よって

$$r \leq d(x_1, x) + d(x_1, p) \leq (1 + \frac{\alpha}{\sqrt{2}})l$$

x_1 が p, q, r 以外の頂点 k を母点とするポロノイ領域に存在する時、上と同様に

$$d(x, k) \leq d(x_1, x) + d(x_1, k) \leq (1 + \frac{\alpha}{\sqrt{2}})l$$

が成り立つ。 $r = d(p, x) \leq d(k, x)$ であるので結局 non-critical triangle では以下が成り立つ。

$$r \leq (1 + \frac{\alpha}{\sqrt{2}})l$$

Δ のすべての辺は $2r$ 以下であるから補題は成立する。

$\alpha = \frac{1}{m\sqrt{2}}$ (m :整数) の時、上限値は実現しうる。(図 10)

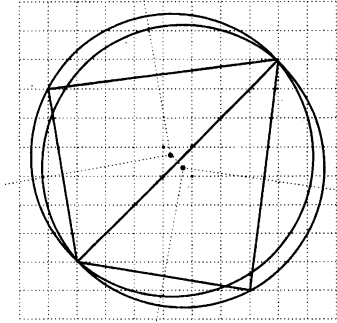


図 10: $\alpha = \frac{1}{3\sqrt{2}}$ の時

次に critical triangle について見ていきたい。 Q の境界上のある辺 e について、 Q の各辺は $Vor(V_Q \cup S)$ の幾つかのポロノイ辺を切り取る。これらの切り取られる辺の端点のうちで Q の外部にある端点(ポロノイ頂点)と双対な三角形をまとめて critical region $R(e)$ と定義する。critical triangle に関しては [4] における補題が同様に成り立つ。

補題 3 $R(e)$ の critical triangle の辺 f は e より長くならない。

証明 e の端点を a, a' 、 f の端点を p, p' とする。点 p と辺 f に双対するポロノイ頂点を結ぶ線分と辺 e の交点 x について、 x は点 p を母点とするポロノイ領域に存在するので、 $d(x, p) < d(x, a)$ が成り立つ。同様に $d(x, p') < d(x, a')$ 。よって

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

が成り立つ。よって $d(p, p') < d(x, p) + d(x, x') + d(x', p')$ から $d(p, p') < d(a, a')$

□

境界が $l \leq d(V, V) \leq \beta l$ ($\beta \leq 2 + \sqrt{2}\alpha$) の範囲で与えられた時、補題 2, 3 およびアルゴリズムの終了条件から以下の定理が成立する。

定理 1 このアルゴリズムで得られた三角形分割の一辺の長さは、 l 以上 $(2 + \sqrt{2}\alpha)l$ 以下である。

次にこのアルゴリズムで得られる角度に関してみていきたい。文献 [5, 8] では、最小角最大化アルゴリズムが提案されている。[5, 8] における証明を元に、正方形グリッドを考慮することによって本論で提案するアルゴリズムの角度評価を行うことができる。

補題 4 non-critical triangle Δ のすべての角は $\arcsin\left(\frac{1}{2+\sqrt{2}\alpha}\right)$ 以上である。

証明 Δ の外接円の半径を r 、任意の一辺の長さを $d(e)$ 、その辺に対応する角の角度を θ とした時、 $d(e) = 2r \sin \theta$ であるので、 $\theta = \arcsin \frac{d(e)}{2r}$ となる。定理 1 より、 $d(e) \geq l$ 。また補題 1 で述べたように $r \leq (1 + \frac{\alpha}{\sqrt{2}})l$ が成り立つ。

$$\text{よって } \frac{d(e)}{2r} \geq \frac{1}{2(1+\frac{\alpha}{\sqrt{2}})} \text{ より } \theta \geq \arcsin\left(\frac{1}{2+\sqrt{2}\alpha}\right).$$

□

補題 5 critical triangle Δ のすべての角は $\arccos \frac{\beta}{2}$ 以上であり、 $\beta \leq \sqrt{2}$ の時 critical triangle は存在しない。

証明 critical region $R(e)$ に関して、 Δ は $R(e)$ に属する三角形とする。このときまず Δ の頂点で $p \notin V_Q$ である頂点 p について、 $\text{Vor}(S \cup V_Q)$ での p のポロノイ領域は e を分断する。 x をこの領域の内部でかつ、境界の外部に存在する点とすると、 x を中心として p をとり e の端点を含まないような円 C が存在する。 x は境界の外部にあることから e は C を切断する。よって C は e を直径とする円 $C(e)$ に完全に含まれる。(図 4 このことから点 p は $C(e)$ の内部に存在する。またす

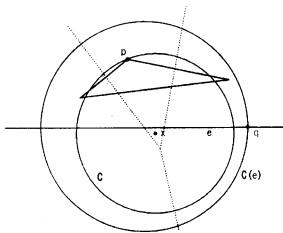


図 11: 頂点 p は $C(e)$ の内部に存在する

べての辺は l 以上であることから、 e の端点を中心に半径 l の円の内部には頂点が存在しない。よって頂点 p は図 12 の影の部分に存在する。

この領域に頂点がただ 1 個存在するとき、最小角となるのは図 12 の太線の三角形のときであるから、このとき最小角は $\arccos \frac{\beta}{2}$ となる。

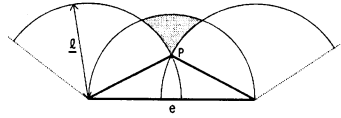


図 12: p は影の部分に存在する

この領域に頂点が 2 個存在するとき、角頂点間の距離は l 以上であることから、頂点が 2 個存在するためには $\beta \geq 2$ でなければならない。 $\beta = 2$ の時、可能な頂点は 3 つあり、これらを用いた三角形での最小角は 30° である。

$\beta \leq \sqrt{2}$ の時、図 12 の影の領域は存在しない。よって critical triangle は存在しない。

□

定理 2 境界が $l \leq d(V, V) \leq \beta l$ ($\beta \leq 2$) の範囲で与えられた時、このアルゴリズムによって得られる三角形の角度は、 $\min\{\arccos \frac{\beta}{2}, \arcsin\left(\frac{1}{2+\sqrt{2}\alpha}\right)\}$ 以上である。

三角形分割の計算時間 三角形分割は逐次的に Steiner 点を追加することから、アルゴリズム終了時における Steiner 点の個数の上限が必要となる。いま境界近以後の多角形 Q の内部に存在する任意の点 x について、アルゴリズム終了時の任意の三角形の外接円の半径は $(1 + \frac{\alpha}{\sqrt{2}})l$ 以下であることから半径 $(1 + \frac{\alpha}{\sqrt{2}})l$ の円をすべての $S \cup V_Q$ の点上に置くと、これらは Q を完全に覆い尽くす。Steiner 点の個数を n とすると、このことを利用して以下の上限值を得ることができる。

$$n < \frac{\text{Area}(Q)}{\pi \left(\left(1 + \frac{\alpha}{\sqrt{2}}\right)l \right)^2} - |V_Q|$$

次にアルゴリズムの繰り返し部分にかかる時間であるが、これは文献 [2] における Voronoi insertion アルゴリズムとほぼ同様な議論で求めることができる。結果として三角形分割に必要な時間は $O(n^2 \log n)$ となる。

5 全体の計算時間

すべての操作に必要な計算時間は次のようになる。

出力に制限がない時

$$O\left(n^2 \log n + \left(\frac{|V_P|}{\alpha^4}\right) \left(\frac{1}{\alpha^2} + \log \frac{|V_P|}{\alpha^2}\right)\right)$$

出力が凸の時

$$O\left(n^2 \log n + \left(\frac{|V_P|}{\alpha^6}\right) \left(\frac{1}{\alpha^2} + \log \frac{|V_P|}{\alpha^4}\right)\right)$$

α	辺長種類 数理論値	辺長種類 数	最大辺長 理論値	最大辺長	最小辺長 理論値	最小辺長	角度種類 数	Hausdorff Dist.
0.1	153	84	85.7	79.7	40.0	40.2	432	1.3
0.2	50	29	91.3	79.2	40.0	43.3	181	2.7
0.4	18	11	102.6	86.2	40.0	45.3	50	6.0
0.6	10	6	113.9	86.5	40.0	48.0	12	9.3

表 1: $l = 40$, α による変化

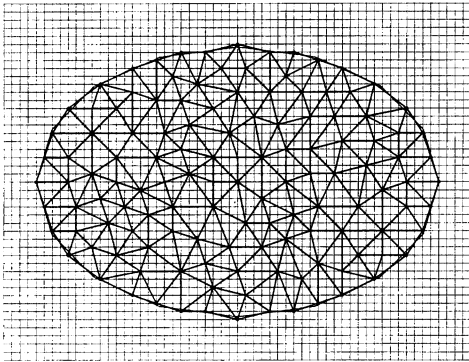


図 13: $\alpha = 0.4, l = 40$

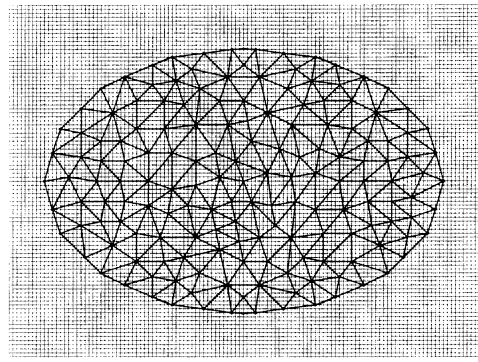


図 14: $\alpha = 0.2, l = 40$

但し

$$n < \frac{\text{Area}(Q)}{\pi \left\{ \left(1 + \frac{\alpha}{\sqrt{2}} \right) l \right\}^2} - |V_Q|$$

三角形分割に必要な時間は実際には $O(n^2 \log n)$ も必要とはならず、 α が小さい時は、ほとんど境界近似によって計算時間は決まる。特に α が小さく凸出力を行うと非常に時間が必要となる。

6 数値実験

ここではこれまで示したアルゴリズムについて計算機実験を行い、理論的精度との比較を行った。

実験 1 ここでは与えられた多角形を l および β 一定のもと、 α を変化させて実験を行った。

初期境界 $x^2 + (y/1.5)^2 = 400^2$, $l = 40.0$, $\beta = 2.0$

この結果を表 1 に示す。辺長の上限值は $(2 + \sqrt{2}\alpha)l$ であるが、実験結果で見られる最大辺長は大幅に理論値を下回っている。そのため辺長種類数も同様に理論値が実験値をかなり上回っている。また図 13 は $\alpha = 0.4$ 、図 14 は $\alpha = 0.2$ での三角形メッシュである。 $\alpha = 0.4$ では多少境界近似に誤差が見られるが、 $\alpha = 0.2$ ではほとんど誤差がない。

実験 2 ここでは実験 1 と同様な初期境界のもと、 $\alpha = 0.4$ で固定し l を変化させた。

表 2 および図 16, 15 がその結果である。図 16 と図 15 では枝の本数は劇的に異なっているが、理論通り l と無関係に辺長種類数はほぼ一定となっている。わずかに l が小さい時、辺長種類数は増加しているがこれは枝の絶対数が大きくなるために理論の上限值に近づいたものと考えられる。

7 結論

本研究において、平面領域を三角形分割する際、境界をわずかながらずらすことによって部材の種類数を減少させることに成功した。またアルゴリズムの性質をよく理解したうえで、結果に対する理論的精度を示した。

既往研究 [4] では異なる辺の種類数は $O(|V|)$ であった。これに対し本論で提案したアルゴリズムでは、異なる辺の種類数は $O\left(\frac{1}{\alpha^2}\right)$ であるため境界の形状や規模に依存しない。このことは建築への応用を考えた際非常に有益なことである。

l	辺長種類数	最大辺長	最小辺長	角度種類数	最大角	最小角
20	13	46.6	22.6	52	108.4	31.0
40	11	86.2	45.3	50	108.4	31.3
60	11	140.0	67.9	36	108.4	33.7

表 2: l による変化、 $\alpha = 0.4$

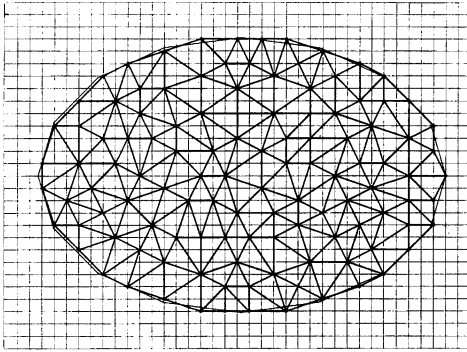


図 15: $\alpha = 0.4, l = 60$

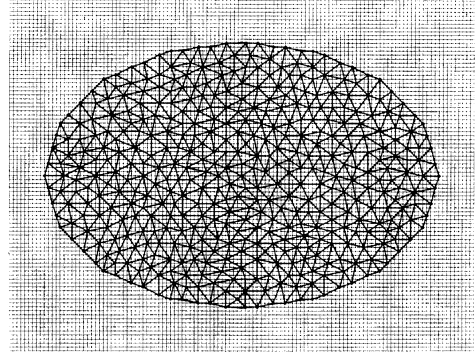


図 16: $\alpha = 0.4, l = 20$

8 今後の課題

境界近似の際に述べたとおり最適なグリッド配置問題が考えられる。グリッドを特徴を用いてさらに高速なアルゴリズムが必要である。またこの研究テーマの最終目標は曲面の三角形分割である。本論のアルゴリズムにおける正方形グリッドを配置する考えをそのまま立方体グリッドに拡張すれば、 $O\left(\frac{1}{\alpha^3}\right)$ 程度の解は得られそうであるが、境界近似は閉曲線だけでなく面全体に及ぶ。そのため平面よりはるかに問題が難しくなる。

参考文献

- [1] Helmut Alt, Bernd Behrends, and Johannes Bloemer. Approximate matching of polygonal shapes. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pp. 186–193, 1991.
- [2] Franz Aurenhammer, Naoki Katoh, Hiromichi Kojima, Makoto Ohsaki, and Yin-Feng Xu. Approximating uniform triangular meshes in polygons. In *Proc. of 6th Annual International Computing and Combinatorics Conference (COCOON2000)*, pp. 22–33. Springer Verlag, 2000.
- [3] Naoki Katoh, Hiromichi Kojima, and Ryo Taniguchi. Approximating uniform triangular meshes on

spheres. In *Proc. of JCDCG 2000*, Lecture Notes in Computer Science (LNCS) Vol. 2098, pp. pp.192–204. Springer-Verlag, 2001.

- [4] Naoki Katoh, Makoto Ohsaki, and Yinfeng Xu. A uniform triangle mesh generation of curved surfaces. In *Proc. of JCDCG 2004*. Springer-Verlag, 2004.
- [5] P.Chew. Guaranteed-quality mesh generation for curved surfaces. In *Proc. of the 10th Ann. ACM Symposium on Computational Geometry*, pp. 274–280, 1993.
- [6] P.Chew, M.T.Goodlich, D.P.Huttenlocher, K.Kedem, J.M.Kleinberg, and D.Kravets. Geometric pattern matching under euclidean motion. In *Comput. Geom.*, pp. 113–124, 1997.
- [7] Jim Ruppert. A delaunay refinement algorithm for quality 2-dimensional mesh generation. *ALGORITHMS: Journal of Algorithms*, Vol. 18, , 1995.
- [8] Jonathan Richard Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *CGTA: Computational Geometry: Theory and Applications*, Vol. 22, , 2002.