4F–06

制約付き Delaunay 三角形分割の実装と評価

佐藤 弘隆　　久保田 光一
中央大学大学院理工学研究科*

1 はじめに

2 次元以上の領域分割は, 多くの工学的な応用において基本的な処理である. 2 次元の領域分割を行う場合には, 三角形分割や四角形分割が使われる. 形状が複雑な領域を分割する場合には, 領域の境界を多角形で近似し, 領域の内部に適当に複数の点を取り, 多角形の頂点と内部の点を頂点とする三角形分割が適している. よく知られている三角形分割には Delaunay 三角形分割があり, データの補間や有限要素法のためのメッシュ生成などに使用されることが多く, その用途は広い.

しかし Delaunay 三角形分割は, 領域が凸でない場合には不適切な三角形分割を与えることがある. 領域の境界の多角形の辺が必ずしも三角形分割の際の三角形の辺として採用されないことがあるからである. そこで, 点集合だけでなく, 辺集合も与えて定義される制約付き Delaunay 三角形分割が提案された [2, 3, 4]. それは, 点集合を三角形の頂点とし, 辺集合を三角形分割の辺として使わなければならないというものである.

制約付き三角形分割には constrained Delaunay 三角形分割と conforming Delaunay 三角形分割の 2 種が提案されている [2, 3, 4]. Constrained Delaunay 三角形分割は, 与えられた辺集合を三角形分割の辺として使う分割である. Conforming Delaunay 三角形分割は, 頂点を追加することによって, 追加された頂点を含む Delaunay 三角形分割の辺が, 与えられた辺集合と重なるようにする三角形分割である. Constrained Delaunay 三角形分割と似ているが, 新しく点を追加するため, より細かい三角形に分割することになる.

本稿では, それらを生成するアルゴリズムを実装し, 性能を評価する.

2 Constrained Delaunay 三角形分割

平面グラフ $G = (V, E)$ が与えられたと仮定する. V は頂点集合であり $|V| = n$, E は辺集合であり $|E| = m$ とする.

定義

G の constrained Delaunay 三角形分割 $T = (V, F)$ とは, $E \subseteq F$ でかつ, $^{\forall} t \in F - E$ について, 次の性質を持つ円 c が存在するものである [2].

1. 辺 t の両端点は, 円 c の境界上にある.
2. V の点 v が, 辺 t の両端点を境界上に持つ円 c の内部にあるとき, 辺 t の少なくとも 1 つの端点から点 v を見ることができない. (すなわち, v から t の端点に向かって線分を書いたとき, 少なくとも 1 本は E の辺と交差する.)

$E = \emptyset$ のときは, constrained Delaunay 三角形分割は Delaunay 三角形分割と同じである.

図 1 では, 左側に点集合と辺集合を持った平面グラフを示し, 右側にその平面グラフの点集合のみを使った一般的な Delaunay 三角形分割を示す. 図 2 は, 図 1 の平面グラフの点集合と辺集合を使った constrained Delaunay 三角形分割を示している.

アルゴリズム [2]

Constrained Delaunay 三角形分割を構成するために分割統治法を使う. 平面グラフ G が与えられたとし, 各頂点の x 座標は互いに異なると仮定する. 最初に x 座標によって G の頂点をソートする. G 全体を内部に含む矩形を考え, これを垂直に切って n 個の短冊に分ける. 各短冊は頂点を 1 つだけ含むようにする. この短冊をストリップという. 分割統治法に従い, 2 つの隣接したストリップを繋合わせる. この際, 2 つのストリップに含まれる点に関して, 2 つのストリップにまたがる constrained Delaunay 三角形を作る. 最終的にストリップが 1 つになるまでこの作業を続ける. 分割統治法を用いたことによって, constrained Delaunay 三角形分割を $\mathrm{O}(n \log n)$ で作ることができる.

このアルゴリズムの短所は, 短冊に分けられないほど多量な点を与えられた場合の処理を考えていないことである. そこで, 平面グラフを小さく分け, 別々に三角形分割を作り, 最後に繋ぎ合わせることができれば, 点の数を気にせず constrained Delaunay 三角形分割を作ることができる.

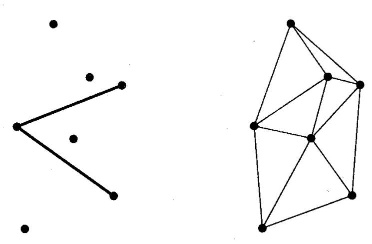

図 1　平面グラフと Delaunay 三角形分割

---

*Implementation and evaluation of constrained / conforming Delaunay triangulations, Hirotaka Satou and Koichi Kubota, Graduate School of Science and Engineering, Chuo University 1–13–27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan.

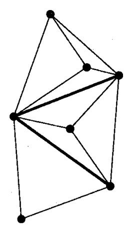

図 2　constrained Delaunay 三角形分割

3 Conforming Delaunay 三角形分割

図 3 では, 図 1 の平面グラフに頂点を追加してできる conforming Delaunay 三角形分割を示す. 追加した頂点は白抜きの四角形で表している.

導入する点の数の上界を $O(m^2n)$ で抑える方法は知られている [3]. 導入する点集合は, blocking phase と propagation phase の 2 つのステップで作成される.

Blocking Phase

Blocking phase の目的は, 頂点を含まず, 互いに交わらない空円を見つけることである. その空円のことを blocking circle と呼ぶ. blocking circle と辺の交点, blocking circle 同士の接点を新しい点として加える.

新しい点を加えることによって, 辺も 2 種類に分類される. 一方が protected edge と呼ばれるもので, blocking circle によって囲まれた辺である. その他の辺は全て unprotected edge と呼ばれる. Protected edge は, 現在の点集合に対し空円を作る. Unprotected edge は, propagation phase によって新たな点を追加され, 分割される.

Propagation Phase

線分 ab に対して, ab を直径とした円を M_{ab} とする. Unprotected edge ab と M_{ab} を考える. c が V に含まれる頂点であるとき, 三角形 abc の辺が protected edge でなく, M_{ab} 内部に c が含まれるなら, c の辺 ab への直行写像である点 c' を新しく追加する.

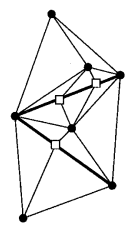

図 3　conforming Delaunay 三角形分割

4 まとめ

この constrained Delaunay 三角形分割のアルゴリズムでは, 万単位の点が与えられた場合, 頂点を 1 つだけ含む短冊に分けることができなくなることが予想されるため, 正常な動作が期待できない. そこで, この問題を解消するために, 記号摂動法 [1] や同座標の複数の点の処理ができる短冊の作成等が考えられる. 現在のアルゴリズムの不安を解消し, 多数の点に対しても正常に三角形分割が作れるアルゴリズムを作成する.

今後, 制約付き Delaunay 三角形分割を地理情報処理で応用できるように, アルゴリズムの改良, 変更, 比較実験 [5] を目標とする.

参考文献

[1] 杉原 厚吉 : FORTRAN 計算幾何プログラミング. 岩波コンピュータサイエンス, 岩波書店, 1998.

[2] L. P. Chew : Constrained Delaunay Triangulations, Proceedings of the third annual symposium on Computational geometry, pp. 215–222.

[3] H. Edelsbrunner and T. S. Tan : An Upper Bound for Conforming Delaunay Triangulations, Discrete & Computational Geometry, 1993, pp. 197–213.

[4] J. R. Shewchuk : Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, Jornal of Algorithms 18(3), 1995, pp. 548–585.

[5] R. J. Renka : TRIPACK: A Constrained Two–Dimensional Delaunay Triangulations Package, ACM Transaction on Mathematical Software, Vol.22, No.1, 1996, pp. 1–8.