

三次元 Delaunay 三角形分割におけるランダム化点位置決定アルゴリズム*

小林 陽介[†] 今井 桂子[‡]

中央大学大学院理工学研究科情報工学専攻[†]
中央大学理工学部情報工学科[‡]

概要

点位置決定問題は計算幾何学の中でも基本的な問題の一つで、様々な手法が提案されている。1996年には Mücke らにより、二次元および三次元の Delaunay 三角形分割内における前処理無し的手法が提案された。本研究では、この手法に前処理を施し、問題を局所的に解くことで効率化を図る手法を提案する。さらに計算機実験による評価も報告する。

1 はじめに

点位置決定問題とは、幾何的な要素の集合に分割された空間の中に質問点を与え、その質問点を含む幾何的要素を答える問題である。この問題は計算幾何学の中でも最も基本的な問題の一つであり、地理情報システムやコンピュータグラフィックス、CAD/CAM などの分野で応用されている。

点位置決定問題は過去に多くの研究がなされており、理論的に最適となるアルゴリズムが数多く提案されている。しかし、これらのアルゴリズムは複雑な前処理や莫大な記憶容量を必要とする。応用分野ではサブルーチンとして使用されることが多いため、実用には不向きであるとされている。そこで 1996 年に Mücke らによりランダム化点位置決定手法が提案された [1]。この手法は前処理や付加的な記憶容量を必要とせず、実装も比較的容易であるという性質を持っているため、実用的といえる。しかし、質問が繰り返し多数回行なわれる場合にはあまり効率的ではない。そこで本稿では、この手法に三角形分割された領域をボックスに分割するという前処理を施し、問題を局所的に解くことで一回当たりの探索時間を速くする手法を提案する。

2 Delaunay 三角形分割

Delaunay 三角形分割とは、3 次元では各四面体の外接球の内部に他の四面体の頂点を含まない四面体分割のことである。メッシュ生成や有限要素法などの応用分野では、Delaunay 三角形分割における点位置決定問題が行なわれることが多い。また近年では、平面だけでなく、空間内に対象物がある場合も考えなければならない状況も起こっており、高次元における研究が進んでいる。そのため本研究では 3 次元 Delaunay 三角形分割における点位置決定問題に焦点を当てている。

*Randomized Point Location in Three-Dimensional Delaunay Triangulation.

[†]Yosuke KOBAYASHI, Information and System Engineering Course, Graduate School of Science and Engineering, Chuo University.

[‡]Keiko IMAI, Department of Information and System Engineering, Faculty of Science and Engineering, Chuo University.

3 ランダム化点位置決定手法

本章では Mücke らが [1] で提案した Jump and march 法を紹介する。この手法は、ランダムサンプリングにより良い探索開始点を決定し、それと質問点を結んだ線分に沿って隣接関係を利用して三角形分割の中を横切り、質問点を含む領域を見付ける。

3.1 一般次元におけるアルゴリズム

d 次元におけるアルゴリズムは次のような流れで行なわれる。 $d-1$ 単体とは 2 次元では三角形、3 次元では四面体のことである。

入力： n 点の点集合 $\{X_1, X_2, \dots, X_n\}$ からなる Delaunay 三角形分割 D と質問点 q

出力：(存在するならば) q を含む D の $d-1$ 単体

- (ランダムサンプリング) 点集合 X_1, X_2, \dots, X_n を置き換えることなくランダムに m 点選択し、それを Y_1, Y_2, \dots, Y_m とする。
- (探索開始点決定) 距離 $d(Y_j, q)$ が最短となるような $j \in \{1, \dots, m\}$ を計算し、探索開始点 $Y = Y_j$ とする。
- (探索) 線分 (Y, q) と交差するすべての $d-1$ 単体を横切っていくことによって q を含む $d-1$ 単体を見付ける。

D 内のある $d-1$ 単体から隣接する $d-1$ 単体へ移動する操作は $O(1)$ で行なわれる。この手法の期待実行時間は、 d 次元上でランダムに配置された n 点の点集合の Delaunay 三角形分割において $O(n^{1/(d+1)})$ である [2]。2 次元でのアルゴリズムは図 1 のようになる。

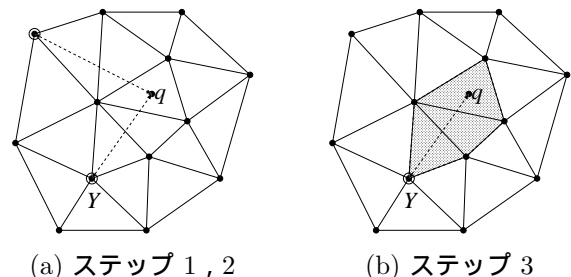


図 1. 2 次元における Jump and march 法

3.2 3次元におけるアルゴリズムの変更点

計算機実験では3次元のときのみ実装している。実装に際して、幾つかの変更を行なった。まず、頂点をサンプリングする代わりに四面体の面である三角形をサンプリングしている。質問点までの距離の計算では三角形の各点からの距離の中で最小のものをを用いている。ここではサンプル数は $m = 2n^{1/4}$ としている。

4 提案手法

前処理無しの手法は、質問回数が少ないときは有効であるが、質問が繰り返し多数行われる場合にはあまり効率的ではない。そこでこのランダム化手法にも前処理を行なうことにより効率化を図りたい。ただし、ランダム化手法の性質をあまり失わないようにするため、なるべく簡単な前処理でなければならない。

4.1 前処理

前処理として、次の操作を行なう。

1. 三角形分割全体を立方体で覆い、それを等体積の B 個のボックスに分割する (図 2)。
2. 各ボックスに三つの点がすべて含まれる三角形のリストを作成する。

2. のリストは質問点が含まれているボックス内でのみランダムサンプリングするのに使用するものである。複数のボックスにまたがっている三角形は、サンプリングの候補としては適当ではないため外すことにする。これはメモリの節約にもつながる。

4.2 アルゴリズム

前処理を施した後、探索のアルゴリズムは以下のように行なう。

1. 質問点が含まれているボックスを見付ける。
2. 1. で見付けたボックスに含まれる三角形のリストの中でランダムサンプリングを行なう。
3. 距離の計算を行ない、探索開始点 Y を決定する。
4. 線分 (Y, q) と交差するすべての四面体を横切っていくことによって q を含む四面体を見付ける (図 3)。

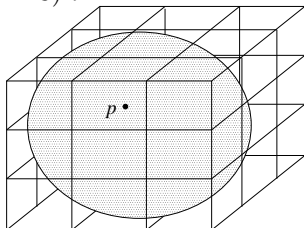


図 2. ボックス分割

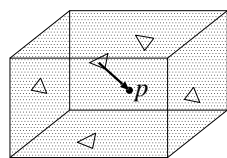


図 3. ボックス内の探索

この手法では、局所的に解くことで問題の規模を縮小することができる。これにより、サンプル数や横切る四面体の数を減らすことができ、計算時間を短縮させることができる。

5 計算機実験

計算機実験では、Jump and march 法と提案手法を実装した。入力には、ランダムに発生させた 1000 点と 5000 点の三次元 Delaunay 三角形分割を用いている。Jump and march 法と提案手法の前処理にかかる時間、一回当たりの平均計算時間、最小計算時間、最大計算時間、横切った四面体の平均個数に関して比較を行なった。提案手法のボックス分割では、三角形分割を覆う立方体を x 軸方向、 y 軸方向、 z 軸方向にそれぞれ二等分にした $B = 8$ の場合と三等分にした $B = 27$ の場合に関して実験を行なった。実験結果は表 1 のようになる。

表 1. 実験結果

	J & M 法	提案手法	
		$B = 8$	$B = 27$
点数	$n = 1000$		
前処理時間 [ms]	-	72.3	120.5
平均時間 [μ s]	632.8	375.6	314.5
最小時間 [μ s]	268.3	275.2	261.5
最大時間 [μ s]	938.2	687.3	412.1
四面体数 [個]	8.4	5.8	4.7
点数	$n = 5000$		
前処理時間 [ms]	-	371.6	619.3
平均時間 [μ s]	921.6	544.3	425.8
最小時間 [μ s]	270.2	284.1	288.4
最大時間 [μ s]	1105.6	952.4	726.9
四面体数 [個]	13.7	9.1	6.6

実験結果より、提案手法では前処理に時間がかかるが、一度前処理をしておけば一回当たりの平均探索時間が速くなることがわかる。そのため、繰り返し多数回質問される場合には効率的である。また、最小探索時間はほとんど変わらないが、最大探索時間を抑えられることも確認できる。

6 おわりに

本研究では、[1] の手法に簡単な前処理を施すことにより計算の効率化を図る手法を提案した。また計算機実験を行なうことにより、実際の動作を確認し、手法の有効性を示すことができた。今後の課題としては、ボックスに分割する際にパラメータを設定することにより、各ボックス内の点数のばらつきをなくすことなどがあげられる。

参考文献

- [1] E. P. Mücke, I. Saias and B. Zhu. "Fast randomized point location without preprocessing in two and three-dimensional Delaunay Triangulation", Computational Geometry: Theory and Applications, special issue for SoCG'96, 12(1/2), pp 63-83, 1999.
- [2] L. Devroye, E. Mücke and B. Zhu, "A note on point location in Delaunay triangulations of random points", Algorithmica 22 (4), 1998.