

平面上における三角形分割を用いた点位置決定問題 とその地図への応用

1Q-4

工藤 靖之*

今井 桂子†

1 はじめに

平面グラフにおいて質問点が与えられているとき、質問点がどの領域にあるのかを決定する問題である点位置決定問題について考える。現在、多くの点位置決定問題の手法が提案されている。中でも最も実用的な手法としてバケット法がある[1]。バケット法が前処理を必要とするのに対し最近 Delaunay 三角形分割を用い前処理を必要としない手法として Mücke らは Jump and march を提案した[2]。

本稿では、Mücke らの手法がどれだけ有効かを検証するために 2 次元の Delaunay 三角形分割を用いて計算機実験を行った。同じデータを用いてバケット法の実験も行った。又、実際上の有用性を調べるのに東京都文京区の地図データに応用した結果について報告する。

2 Jump and march 法

Jump and march 法は前処理を必要としない手法で、そのための記憶容量も必要としない。探索時間は $O(n^{\frac{1}{2}})$ である。

入力： n 点からなる三角形分割と 1 点の質問点 q

出力： q を含む三角形

与えられた点集合、辺集合により Delaunay 三角形分割を生成し、以下の手順により q を含む三角形を探索する。

1. Delaunay 三角形分割上の辺から $n^{\frac{1}{2}}$ 本の辺をランダムにサンプリングする
2. サンプリングした辺の中で辺の中点と q との距離が最短となる辺を選びこの辺を e とする
3. 選択された辺の中点と q を結びこの線分を L とする

4. e を含む三角形の中で L と交差する辺を選びこれを新しい e とする

手順 3 と 4 を交差する辺が無くなるまで繰り返す。交差する辺が無くなったときは、 q を含む三角形が見付けられたことになり、三角形を出力し終了する。新しく選んだ e が凸包上の辺ならば、 q が Delaunay 三角形分割内部に存在しないため三角形は出力されずに終了する。

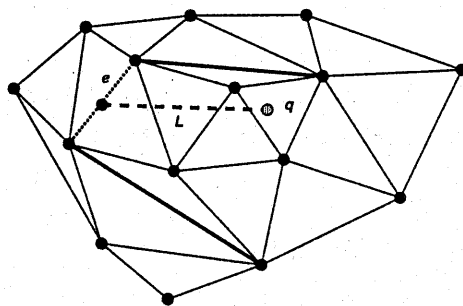


図 1. Jump and march 法

3 バケット法

バケット法は点位置決定問題の代表的な手法であり、地図データでの実用性も確認されている。この手法は通常 $O(n)$ の前処理時間と記憶容量を必要とし、探索時間は $O(1)$ である。最悪の場合は前処理時間と記憶容量は $O(n\sqrt{n})$ 、探索時間は $O(n)$ となる。

まず前処理では三角形分割全体を覆うような長方形を与える。これを $O(n)$ 個の等面積の四角形(バケット)に分割する。さらに次のデータ構造を作る。

- 各バケット内の三角形分割の点を y 座標に関してソートしたリスト
- バケットの垂直辺と交わる三角形分割の辺を交点の y 座標に関してソートしたリスト

*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

- バケットの水平辺と交わる三角形分割の辺のリスト
- バケットの各頂点を含む面

探索では、まず質問点を含むバケットを見付ける(図2)。

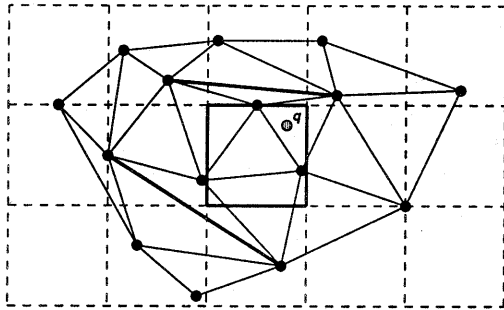


図2. バケット法

選ばれたバケット内をスラブ分割し、質問点を含むスラブを二分探索により求める。さらにスラブ内の領域を二分探索することにより質問点を含む領域、すなわち質問点を含む三角形を見付けることができる。

4 計算機実験

二つの手法を Delaunay 三角形分割を用いて計算機実験を行った。プログラム言語は C 言語、機械は pentium III 500 MHz, メモリ 128 MB のものを用いた。図3は 10^6 回計算した平均をとり、さらに異なるデータにより 10 回これを繰り返した結果である。

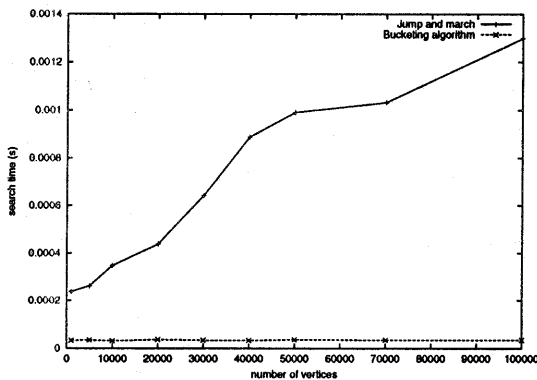


図3. 探索時間の比較

5 地図への応用

Jump and march 法を各領域が多角形であるような地図データに応用するときは領域を予め三角形分割しておく必要がある。Bucket 法は多角形領域に対して前処理を行う。[3] を用いて東京都文京区の地図データとして、区と街区の境界線を辺とする constrained Delaunay 三角形分割を作成した。入力点数は 208 点、辺数は 259 であった。探索時間は Jump and march が 0.000212 秒、バケット法が 0.000032 秒であった。図4は文京区の境界線のみを取り出したものである。

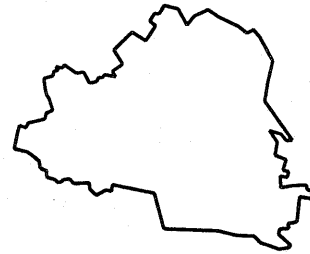


図4. 文京区の境界線

6 まとめ

探索時間では前処理の無いバケット法が速いが、質問回数が少なく、データが膨大で前処理で作るデータ構造が構築できないような場合には、Jump and march 法は前処理にかかるメモリが無いため、メモリを必要とするバケット法よりも有効であろう。今後の課題として大規模な地図データの実験が挙げられる。

参考文献

- [1] Masato Edahiro, Iwao Kokubo, Takao Asano: "A New Point-Location Algorithm and Its Practical Efficiency - Comparison with Existing Algorithms", ACM Transactions on Graphics, Vol.3, No.2, April 1984, pp.86-109.
- [2] Ernst P. Mücke, Isaac Saias, Binhai Zhu: "Fast randomized point location without preprocessing in two- and three-dimensional Delaunay triangulations", Computational Geometry 12 (1999), pp.63-83.
- [3] 数値地図 2500, 国土地理院.