

多角形と図形の交点の高速算出

百地類億

平成 26 年 10 月 20 日

概要

平面上の多角形と様々な図形との交点を高速に求める新たな手法を提案する。今回提案する手法は多角形にたった一つのデータ（辺の最大辺長）を付加するだけで実現できる。多角形と直線との交点を求める際には非常に有力な方法と言える。多角形と多角形の交点を調べる場合には平面走査法が大変有力で、残念ながら平面走査法より提案手法は速い結果は出せなかった。ただ、提案手法は平面操作法よりメモリ消費を削減できる点、また実装が簡潔である点は優れている。

1 始めに

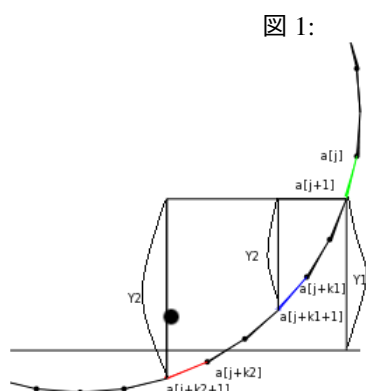
図形と多角形の交点を求める問題はグラフィックス分野では一般的な問題と考えられる。例えば、内点が多角形に内包されているかどうかは内点から半直線をひき多角形と交差する回数で判断する。また、筆者はトーセシステムズで、半導体の割れたワークと切削線の交点を求めるアルゴリズムを開発した。このような問題を解く場合、従来は平面操作法 [1] が用いられていた。今回、多角形の最大辺長を利用して、多角形と図形の交点を求める方法について考察した。また、グラフィックからは離れるが第 5 章では探索する配列の隣り合うキーの値の最大値を利用してデータ探索をするアルゴリズムを考察した。

2 多角形と直線の交点

2.1 アルゴリズムの概略

まず、多角形と直線の交点を求める方法を考察する。これは多角形と多角形の交点を求める際にも基本になる方法である。素朴な方法は全ての多角形の辺と直線とが交わるかと調べる方法である。この方法の計算量は多角形の辺の数を m とすると $O(m)$ で

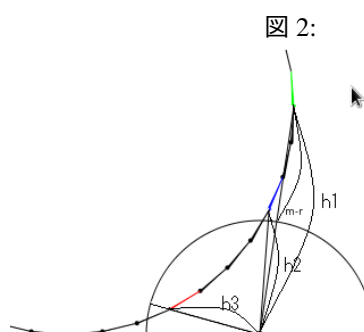
ある。この方法でも今のコンピューターならかなりの速さで計算できるが、多角形の辺の数が多かったり、複数の直線と交点を求めると遅くなってしまふ。素朴な方法を応用して、交点を直線から遠い辺については粗く探し、近い点に関しては密に探す方法は簡単に考えることができるだろう。今回提案する手法では多角形の辺の長さの最大値（これを l とおく）を用いてそのような方法を実現する。すぐに分かることだが、多角形の辺の長さの最大値が既知でない場合辺の長さの最大値を求めるのに $O(m)$ の計算量がかかる。よって、多角形の辺の長さの最大値が分かっている場合かつ、交点を一つの直線と多角形のみから求める場合は提案手法は適さない。さて、多角形の点が配列 a に格納されているとする。すると、多角形の各辺は $(a[j], a[j+1])$ で表される。素朴な方法ではこの j を全て走査して、直線との交点を求めることになる。今回の提案手法では j の値を飛び飛びに増加させる。その増加幅の適切な方法を求めるのが今回の手法の目玉である。飛び飛びに求めるといっても辺が直線を完全に飛びこえてしまつては意味がない。図 1 でいえば、 $(a[j], a[j+1])$ が $(a[j+k1], a[j+k1+1])$ に移動するのは大丈夫だが、 $(a[j+k2], a[j+k2+1])$ に移動するとまずい。適切な移動量 k を求めるのが重要である。ここで、点 $a[j+1]$ と直線との距離を求め、これを $y1$ とする。配列の添字が j から $j+k$ へ移ると


$$y_2 \leq kl \quad (1)$$
$$k = \lfloor y1/l \rfloor \quad (2)$$
$$k = |y1/l| + 1 \quad (3)$$

2.2 アルゴリズムの速さの測定

数を測定した。素朴な方法では時間は 2.27 秒かかりループの回数は 100000000、提案手法では時間は 0.25 秒かかりループの回数は 298842 だった。これを見ると分かるように提案手法はかなりの高速化を図ることができることが分かる。

3 多角形と円の交点

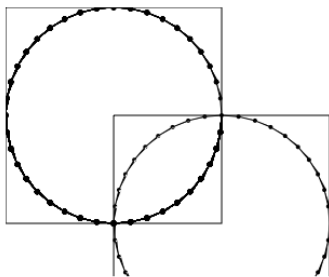

$$\delta h < kl \quad (4)$$
$$k = \lfloor |h - r|/l \rfloor + 1 \quad (5)$$

2

4 多角形と多角形との交点

多角形と多角形との交点を高速算出するアルゴリズムについて述べる。これは直線と多角形との交点を求めるアルゴリズムの応用と言える。片方の多角形の辺を線分とみなし、それにもう片方の多角形に提案手法を適用すればよい。提案手法を適用する多角形は辺の数が多いほうを選ぶのが望ましい。ここ

図 3:



までに述べた方法でも十分に高速化を図ることが可能だが、実際にはさらに、多角形を囲む X 軸と Y 軸に平行な最小の長方形を用いることでさらに高速化を図ることができる。(図 3) 明らかに二つの多角形を囲む長方形同士が重ならなければ二つの多角形は交わらない。これはわずか四つの不等式で判定可能である。さらに、辺の数が少ないほうの多角形の辺と長方形に対して提案手法を適用すれば、さらなる高速化を図れる。提案手法と平面走査法で正一万角形同士の交点を 100 回求めたところ、提案手法で時間は 3.08 秒、ループ回数は 8243054 回、平面走査法で時間は 0.92 秒、ループ回数は 1002106 回だった。残念ながら提案手法では平面走査法を上回ることはできなかった。しかし、提案手法と平面走査法で正一万角形と正千角形とで交点を百回もとめたところ、提案手法で時間は 0.28 秒、ループ回数は 829914 回、平面走査法は 0.58 秒、ループ回数は 408630 回かかり提案手法のほうが高速であった。これは提案手法の計算回数が辺の少ない多角形の辺の数に比例するのに対し、平面操作法は、両方の多角形の辺の数の

合計に計算量が比例するからである。よって、二つの多角形の辺の数が大きく違う時には提案手法が活用できる。

5 提案手法の探索への応用

5.1 アルゴリズム

この章では提案手法を配列のデータ探索に応用する方法について考察する。提案手法を探索に応用するには配列のデータのキーが“整っている”必要がある。ここでは探索する配列はソートされているとし、隣り合うキーの値の差が 1 から 1000 の一様離散分布にしたがうとする。データの個数は百万とする。基本的なアルゴリズムの流れは多角形と直線との交点を求める時に使った流れと同じであるが、一応もう一度考察する。隣り合うキーの値の差の最大値を n とする。調べている配列のキーの値を m 探しているキーを o と求めたいデータを飛び越えないためには移動量 k は

$$k = \lfloor |m - o| / n \rfloor \quad (6)$$

となる。しかし、これだと k が 0 になってしまう可能性があるので実際には

$$k = \max(\lfloor |m - o| / n \rfloor, 1) \quad (7)$$

となる。

5.2 アルゴリズムの評価

線形探索と、2 分探索と、提案手法を配列に一万回適用した場合のループ回数と時間は、線形探索は 767239859 回で、20.61 秒、2 分探索は 205577 回で、1.27 秒、提案手法は 189286 回で 1.27 秒だった。これを見ると提案手法は 2 分探索と同等の性能を示していると言える。これについて考察する。隣り合うデータの差は一様離散分布に従うので、平均は $n/2$ である。よって配列の添字の移動量の平均 k は現在調べている配列の添字を p 目的のデータの配列の添字を q とすると

$$k \approx |p - q| * (n/2)/n = |p - q|/2 \quad (8)$$

よって計算量は配列全体の大きさを r とすると平均計算量は $O(\log_2(r))$ となる。これは 2 分探索と同等の性能である。もっとも配列がソートされているからといって、配列間のキーの値の差が一様離散分布になっているとは限らない。例えば、一つの配列間の値の差が極端に大きいものがあれば計算量は $O(r)$ になってしまう。さらに、提案手法は配列をソートする手間だけでなく隣り合う配列間のキーの最大値を取得する手間もかかる。よって、ソートされているデータに対して、単純に提案手法を適用するメリットはないと考えられる。しかし、提案手法は必ずしも完全にデータがソートされている必要はなく、“整っている”データであれば適用できる点、さらに、データを逆行する必要がない点は優れている。

6 提案手法の長所と短所

6.1 提案手法の長所

原理が単純で簡単、実装も容易

提案手法ではいっさい難しいアルゴリズムを使用していない。線分の距離の公式など初等的な方法のみで構成されている。さらに、素朴な方法を発展させたような形になっているため、既存のシステムに組み込むことも容易である。

メモリ消費量が非常に小さい

提案手法は多角形に対してたったひとつのデータ（辺の最大長）しか付加していない。よってメモリの消費量は他の手法に比べて非常に少なくなっている。

6.2 提案手法の短所

高速化の度合いが多角形の形状に依存している

提案手法は図 4 のように、一辺が極端に長い多角形や、図 5 のように“整っていない”多角形には適用しても高速化が図れない。図 4 のような

図 4:

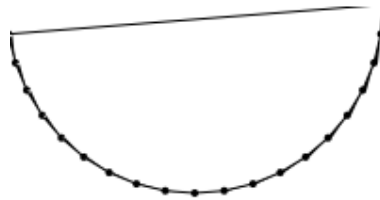


図 5:



多角形にはダミー頂点を追加して問題を回避することが可能であるが、図 5 のような多角形には他の手法を考えるしかない。

7 提案手法の課題

三次元への応用

提案手法を三次元の多面体へ応用することは不可能ではないが二次元の多角形に適用するように単純ではない。多面体の頂点の配列への格納の仕方が、多角形に格納するように一意ではないからである。多面体の頂点をどのように格納すればいいかはさらなる研究が待たれる。

計算量の定量的な評価

今回提案手法で探索アルゴリズムを適用する際

の計算量の平均が $O(\log_2 r)$ になることを示したが、多角形と図形の交点を計算する場合の計算量の定量的な評価は行っていない。こちらもさらなる研究が待たれる。

参考文献

- [1] gopolis から学ぶ計算幾何 第9回
凸多角形の交差計算(前編)
<http://gihyo.jp/dev/serial/01/geometry/0009>
- [2] .セジウィック Algorithms in C 近代科学者 1996