3L-3

計算誤差があっても破綻しない線分交点列挙アルゴリズム

金沢裕治　　杉原厚吉
東京大学工学部

1．はじめに

平面上の線分の交点の列挙は、CAD、地理情報処理など、広い範囲で応用される重要な問題である。

平面上にn本の線分が与えられたとき、それらの間の交点をすべて求めるには、素朴な算法で、$O(n^2)$の時間が必要である。Bentlay と Ottmann は、kを交点の数として、$O((n+k)\log n)$の時間で求められるアルゴリズムを示した［1］。しかし、このアルゴリズムがすべての交点を求めることが保証されているのは、計算誤差が存在しない場合のみであり、計算誤差が存在すると、交点を見落としたり、計算中に矛盾が生じて破綻する可能性がある。

本論文では、$O((n+k)\log n)$という時間を変えることなく、計算中にデータの矛盾を生じないことを保証するアルゴリズムの構成を試るとともに、見落とす可能性のある交点の性質を考察する。

2．線分の交点列挙アルゴリズム

Bentley と Ottmannによって示されたアルゴリズムでは、x軸に垂直な直線（これを sweep line と呼ぶ）を$x=-\infty$から$x=+\infty$まで移動させる。sweep line がある位置にあるとき、sweep line と交点を持つ線分を、その交点のy座標によって順序づける。sweep line が$x=-\infty$から$x=+\infty$まで移動するとき、交点を持つ線分対は、少なくとも交点の周辺で sweep line 上での順序が隣接する。従って、sweep line 上での順序がどこかで隣接する線分の間でのみ、交差判定をすればよく、それによって不必要な交差判定を省略できる。

このアルゴリズムは次の通りである。ただし、アルゴリズムのQとRは、Q：x座標によって順序づけられた点の集合、R：sweep line との交点のy座標により順序づけられた線分の集合である。

Ⅰ：　Q←すべての線分の端点のx座標
　　R←ϕ

Ⅱ：　Qが空でない限り、Qの中のx座標最小の点pに対し、次の操作を行う。

Ⅱa：pが線分sの左側の端点のとき：
　sをRに加える。
　R上で、sの、上、下で隣にある線分とsの間の交点の有無をそれぞれ調べ、交差していれば交点をQに加える。

Ⅱb：pが線分sの右側の端点のとき：
　R上でsの上下の線分の間の交差を、まだ調べていなければ調べ、新しく交点が見つかればそれをQに加える。
　sをRから除く。

Ⅱc：pが線分sとtの交点のとき：
　R上でsが上、tが下にあるものとする。
　sとtの、R上での順序を交換する。
　tと、R上でtのすぐ上にある線分の交点をまだ調べていなければ調べ、交点があればQに加える。
　sについても同様にする。
　sとtの交点を出力する。

3．計算誤差の影響

2．で述べたアルコリズムは、計算誤差があると様々な不都合を生じる。不都合な場合の例には、たとえば、次のようなものがある。

a）図1で、bをRに加えるとき、計算誤差のために、aよりも下に入ってしまったとする。aとbは交点を持っていないので、bは常にaより下にあると判断され、aより上にあるcと、bの交点が見落とされる。

b）計算誤差のため、交点のx座標が実際よりも小さくなり、sweep line がすでに通過した領域に交点がいれられてしまうと、線分の順序を交換する機会が失われてしまう。

c）図2において、線分a，b，cは、ごく近い点で交差しているものとする。このとき、b，cの交点がa，bの交点よりも右に挿入されてしまうと、sweep line がa，bの交点に到達したとき、bとcはまだ交換されていないため、aとbの間にcが残っている。ここで単純にa，bの位置を交換すると、a，b間だけでなく、aとc、bとcも交換することになり、順序関係が破壊されてしまう。

4．計算誤差対策

3．で述べた3つのケースのうち、

a）は、現在の sweep line 上での上下関係と、線分の右端での上下関係が逆になっていれば、交点があるとみなす、

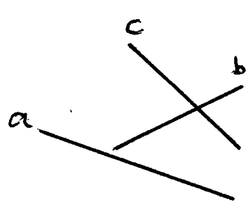

図1

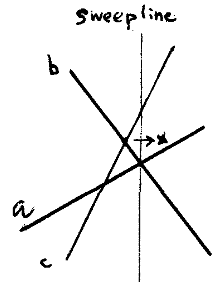

図2

Numerically Stable Algorithm for Reporting Geometric Intersections
Kanazawa Yuji, Sugihara Kokichi
University of Tokyo

ｂ）は、求められた交点のｘ座標が、 sweep line のｘ座標よりも小さければ、 sweep line の直後に挿入する、ことにより矛盾の発生を阻止できる。

残ったｃ）のケースであるが、 sweep line が線分ｓとｔの交点に到達したとき、ｓ，ｔの間に他の線分が残っているときは、

①ｕとｓまたはｔの交点が、実はｓ，ｔの交点の左にある、

②ｕの右の端点が、ｓ，ｔの交点の左にある、

③実は、ｓ，ｔは交差していないのだが、計算誤差のため、交点が求まってしまっている（図３）、

の３つの場合が考えられる。原因を推定し、ｕをｓ，ｔの間から追い出さないと、ｓとｔを交換することはできない。

このケースに対処するため、２．のアルゴリズムのⅡｃを次のように置き換える。

```
Ⅱｃ’：ｐが線分ｓ，ｔの交点のとき：
    exchange(s, t)
```

ただし、exchange は次のように定義される手続きである。

```
exchange(s, t) （以下ｓがｔより上にあるとする）
  while Ｒ上でｓ，ｔの間に他の線分が存在している
    ｕ←ｓのすぐ下の線分
    ｕとｔの交差をまだ調べていなければ調べる。
    if  ｕとｔ、ｕとｓの交点が存在しないか、あ
        るいは交点が sweep line よりも左にある
      if  ｕの右端点がｓ，ｔの右端点より左に
          ある（②のケース）
        ｕの右端点に到達したときの処理を行う。
      else   （③のケース）
        ｕとｓが交点をもっていると見なして交換する
    else  （ｕとｔ，ｓの交点で、 sweep line より右
          にあるものが存在している。①のケース）
      ｔとｓのうち、ｕと交点をもち、その交点が最も
      左にあるものをｗとする。
      exchange(u, w)
  end of while
  ｓとｔのＲ上の順序を交換し、ｔとｔの上にある線分
  の交差をまだ調べていなければ調べ、交点が見つかっ
  たらＱに入れる。ｓについても同様。
  ｓとｔの交点を出力する。
```

これによって交点が見つかった線分対は必ず１回順序が交換され、交点がみつからなかった線分対は順序が交換されないことが保証される。また、ここで挙げた操作はすべてＯ(1)でできる操作か、あるいはいずれ行わなければならない操作を先にしただけなので、アルゴリズム全体の手間は変わらない。

５．見つからない交点の性質

４．で述べた対策を講じることによって、データに矛盾が生じないことは保証されるが、交点が見逃される可能性も残っている。

例えば図４で、線分ａ，ｂの交点がｃの右端点より右に求まったとすると、 sweep line がａ，ｂの交点に到達してａとｂが交換されるときにはｃはなくなっているので、ａ，ｃの交点は見つからない。このように、交点のｘ座標の誤差範囲内に端点があるときに、交点が見落とされる可能性がある。

計算誤差によって、交差しているかしていないかが誤るような線分対が存在しているときは、事情はもっと複雑になり、解析は困難になる。

例えば、図５で線分s_iとs_{i+1}は、計算誤差のために交差していないと判断されるとする。このとき、隣接する線分の間では、すべて交差なしと判断されるので、１つも交点が見つからないことになる。

６．おわりに

線分の交点列挙のための sweep line 法を、計算量を劣化させないで、数値誤差が生じても破綻することのないアルゴリズムに改良した。改良に際しては交点をなるべく見落とさないように配慮したが、まだ完全ではない。検出すべき交点を必ず検出できるアルゴリズムへの改良が次の課題である。

有益な御意見をいただいた東京大学伊理正夫教授、大阪電気通信大学浅野哲夫教授に感謝の意を表する。本研究は、文部省科学研究費補助金（課題番号01550279）の援助を受けている。

参考文献

［１］J.L.Bentley and T.A.Ottmann：Algorithms for Reporting and Counting Geometric Intersections, IEEE Transactions on Conputers, vol.C-28(1979),pp. 643-647.

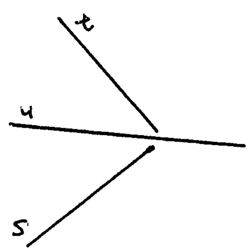

図３

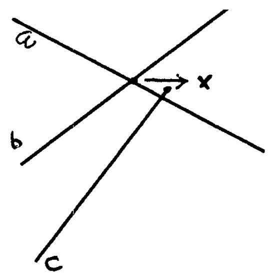

図４

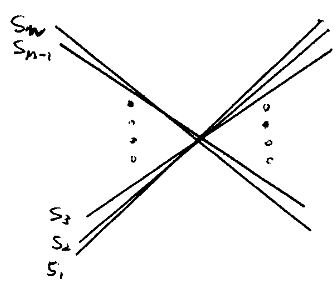

図５