研 究 速 報

特 集 5

UDC 681.3：165：621.397.3

近接図形検索の高速化

A Fast Algorithm for Detection of Neighboring Graphic Objects

林　英明*・相良　毅*・大沢　裕*・坂内正夫*

Ying Ming LIN, Takeshi SAGARA, Yutaka OHSAWA and Masao SAKAUCHI

1. は じ め に

地理情報処理等のグラフィックス，機能図形情報の応用分野では，しばしば「勢力域」を求めることが必要となる．地図上に点群が与えられたとき，地図平面をその中のおのおのの点にとってそれが最も近い点になる領域に分割する問題である．たとえば，小学校の学区を決定する場合に，もし距離が最も近い小学校に通学させようという基準で学区分割をする場合に，このような勢力図を求める必要がある．この分割はボロノイ分割と呼ばれている．古くは，このボロノイ分割の研究は，結晶学の分野で，結晶の形状を求めるために行われた．近年では，例にあげた地理情報処理での応用も含め，さまざまな幾何的な問題の解決に応用されており，計算幾何学の中心的な課題の一つとなっている[1]．一方，ボロノイ分割の逆グラフはドロネー網と呼ばれる．これは，ボロノイ分割されたおのおのの領域について，辺を共有する点（母点）同士を結ぶグラフである．

従来，ボロノイ分割のアルゴリズムの研究は，点群が与えられた時，その点群全体のボロノイ分割をいかに高速に，かつ少ないメモリー量で行うかに興味視点がおかれていた．一方，地理処報システム等では演算の１つとしてボロノイ分割も必要となるが，これを用いない地理的演算も多い．したがって，ある特定の処理（この場合ボロノイ分割を必要とする処理）のために，ボロノイ分割を行った膨大な量のデータを常に持つことは得策ではない．必要が生じた時，必要な部分に対してのみボロノイ分割を行えば十分である．ドロネー網についても同様であり，隣接する点を求める必要が生じる都度計算することを時間効率よく行えば実用上十分である．

汎用多次元データ構造で管理されているデータを対象として，必要が生じるつどドロネー網（またはボロノイ領域）を計算することは，動的性能の向上という利点もある．従来開発されている方法では，母点に追加や削除などの動きがあった場合，再構成が必要となる．追加の場合の再構成は比較的容易であるが，削除の場合には追加ほど容易ではなくなる．さらに，母点すべてのボロノイ図を求める際には，母点の数が多い場合，計算誤差の蓄積[4]も深刻な問題になる．一方，必要に応じてそのつど作成する方法では，データの動きに無関係になり，計算誤差の影響も少ない．

本稿では，このような立場で，汎用多次元データ構造で管理されているデータを対象にして，必要になる母点に対してのみドロネー網を効率よく求めるアルゴリズムを提案する．

2. ボロノイ図とドロネー網

与えられた母点群 P_1，P_2，...，P_n に対して，母点 P_i のボロノイ領域（Voronoi region）$Vor(P_i)$ とは「他の母点より P_i に近い点のなす集合」．すなわち，$Vor(P_i) = \{p \mid d(p, P_i) < d(p, P_k), \forall k \neq i\}$ である．ただし $d(p, q)$ はユークリッドの距離である．これらの領域は，各母点の優勢域（domain）を定義しており，全体でボロノイ図（Voronoi diagram）を構成する．

ドロネー網（Delaunay net）はボロノイ図の双対グラフ（dual graphs）である．ボロノイ図から，境界を共有するボロノイ領域の母点同士を線分で結ぶとドロネー網になる．これは，与えられた母点群のなす凸包（convex hull）を三角形に分割するため，「ドロネー三角形分割」（Delaunay triangulation）とも言う．この分割は次の「空円性質」（empty circle property）をもっている[2]．

定理 1 三つの母点 P，Q，R を頂点とする三角形がドロネー三角形（*Delaunay triangle*）である $\longleftrightarrow$ P，Q，R を通る円の内部に母点が存在しない．

母点 P_i のドロネーネーバー（Delaunay neighbors）と

*東京大学生産技術研究所　第3部

は，ドロネー網で，P_iにつながっている母点の集合．

ボロノイ図の構成については，種々の算法が提案されているが，その主なものは「分割統治法」(divide and conquer) と「逐次添加法」(incremental method) の二種類に大別できる．前者は，母点群の二等分割を繰り返して扱いやすい数の点に分割してから，それぞれのボロノイ図を求め，次にそれらを合併する処理により全体のボロノイ図を求める方法である．後者は，母点を一つずつ添加しながら，ボロノイ図を調整し成長させる方法である．理論的な計算量は，最悪の場合に「逐次添加法」の $O(N^2)$ より「分割統治法」の $O(NlogN)$ が優れるが，一般の場合には「逐次添加法」が効率的で (平均 $O(N)$)，実用的であるとされている[3]．

3．ドロネーネーバーの算出

3.1　ドロネーネーバーの逐次修正

ここでは，ある1つの母点 P のドロネーネーバーを求める方式を検討する．方式の基本は，まず多次元データ構造上で母点 P の近傍の点群を探し，その1点1点に対してドロネーネーバーであるかどうかチェックしながらドロネーネーバーの集合を調整する方法である．$NBRST$ を P のドロネーネーバーのなす集合としておく．P のドロネーネーバーの候補として，近傍の母点 Q をチェックする際に，もし Q がドロネーネーバーであると判明すれば，$NBRST$ の中で，新しいドロネーネーバー Q の影響で，ドロネーネーバーでなくなるメンバが存在する可能性がある．このため，母点 Q を $NBRST$ に挿入するとともに，ドロネーネーバーでなくなる母点を $NBRST$ から削除する必要がある．定理1を用いれば，この処理は単純な円と点の包含判定に帰着させることができる．

■*DELAUNAY_NEIGHBORS*

● 入力：ドロネーネーバー $NBRLST$，候補ネーバー Q

● 出力：調整したドロネーネーバー $NBRLST$

Step. 1　$NBRLST$ で，Q を挟む角 $\angle P_kPP_{k+1}$ を捜す．
/*図1のように，この領域は，円 $D_{P_kPP_{k+1}}$，$D_{P_{k-1}PP_k}$，$D_{P_kPP_{k+1}}$ により，A，B，C，D，E の五つの範囲に分けられる．*/

Step. 2　if ($Q \notin D_{P_kPP_{k+1}}$) then
/*図1のE範囲：reject Q*/
goto stap 9.

Step. 3　if ($Q \notin D_{P_{k-1}PP_k}$) and ($Q \notin D_{P_{k+1}PP_{k+2}}$) then
/*図1のA範囲：accept Q as a new Delaunay neighbor*/

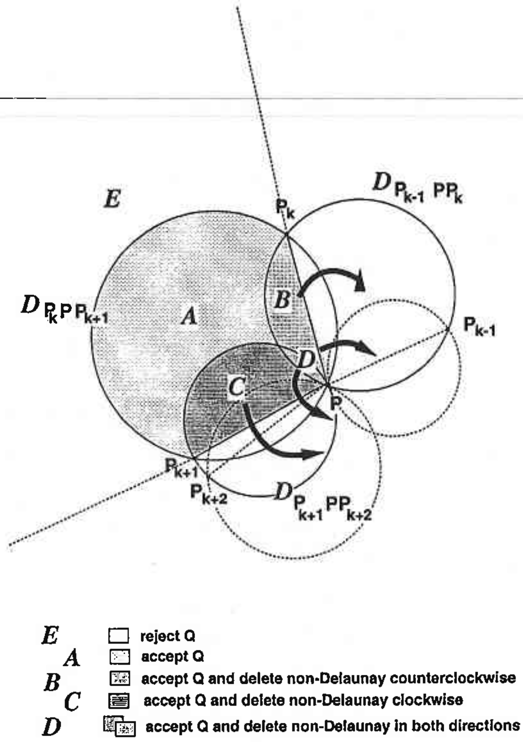

図1　ドロネーネーバーの修正

Step. 3.1　Q を P_k と P_{k+1} の間に挿入する．

Step. 3.2　goto stap 9

Step. 4　$m \leftarrow k+1$.

Step. 5　$n \leftarrow k$.

Step. 6　if ($Q \in D_{P_{k+1}PP_{k+2}}$) then
/*図1のCとD範囲：反時計方向に Q を含む円を検出し削除する*/

Step. 6.1　delete P_m.

Step. 6.2　$m \leftarrow m+1$.

Step. 6.3　if ($Q \in D_{P_mPP_{m+1}}$) then
goto step 6.1.

Step. 7　if ($Q \in D_{P_{k-1}PP_k}$) then
/*図1のBとD範囲：時計方向に Q を含む円を検出し削除する*/

Step. 7.1　delete P_n.

Step. 7.2　$n \leftarrow n-1$.

Step. 7.3　if $(Q \in DP_nPP_{n-1})$ then
goto stap 7.1.

Step. 8　QをP_nとP_mの間に挿入する．
/*accept Q*/

Step. 9　STOP.

3.2 空間データ構造に基づく実現

実際の応用で，母点群は常に空間データ構造に管理されるので，Pのドロネーネーバーの全体がちょうど一つの葉ノードに格納されていることは期待できない．そのため，先に述べた処理（*DELAUNAY_NEIGHBORS*）は，まずPがおかれている葉ノードNに格納される母点群に施され，検出した*NBRLST*のメンバと母点Pを頂点とするドロネー三角形の外接円群を含む最小外接長方形*MBR*も同時に求められる．次に，Pの「近傍」として，*MBR*の内部にある母点群から，Pのドロネーネーバーを捜し続ける．すなわち，葉ノードNでの処理が終わってから，その親ノードに戻り，*MBR*と重なる葉ノードを辿って処理しながら，*MBR*を調整する．ある中間ノードが*MBR*を完全に含む時点で，処理を終了する．

明らかに，全体の計算量は*MBR*に含まれる母点数に依存している．*MBR*は，初期の全空間から，ドロネーネーバー*NBRLST*の修正と共に調整され，収束する．これを速く収束させるため，*quadr*というフラグで，先に辿るべき「方向」を示しておく．母点Pを原点とした四つの象限の中でi-象限でドロネーネーバーが見つかれば，$quadr_i$が0である象限には，ドロネーネーバーはまだ見つかっていないので，象限の全体は*MBR*の一部として収められてしまう．したがって，*MBR*が広くなり，それと重なる余計な葉ノードも増えてしまう．このような象限にある葉ノードを優先に処理すれば，*MBR*の収束が速くなれる．

Stap. 1　初期値の設定

Stap. 1.1　データ空間の四つの頂点をPのドロネーネーバーとして，*NBRLST*に登録する．

Stap. 1.2　ドロネー三角形の外接円の全体を含む最小外接長方形*MBR*を求める．

Stap. 1.3　$quadr_i \leftarrow 0$, $i=1, \ldots, 4$
/*象限flag */

Stap. 2　GBD木でPのある葉ノードNを辿る．

Stap. 3　for（Nに登録された点P_k）do

Stap. 3.1　*DELAUNAY_NEIGHBORS*
(*NBRLST*, P_k)

Stap. 3.2　if（P_K accepted）then
P_kのある象限と対応するフラグ$quadr_i$を1にする．

Stap. 4　*MBR*を調整する．

Stap. 5　$N \leftarrow N$の親ノード

Stap. 6　for（i＝1,2,3,4）do
if（$quadr_i$ ＝＝0）then
象限iでPに最も近い子ノードN_kを選択しドロネーネーバーの探査を続ける．

Stap. 7　for（Nの子ノードN_k）do
if（$N_k \subset MBR \neq \phi$）then
N_kを選びドロネーネーバーの探査を続ける．

Stap. 8　*NBRLST*から，初期値として登録された四つの頂点を削除する．

Stap. 9　STOP.

4．実 験 と 検 討

前章で提案した方式を，多次元データ構造（GBD木）を用いて実現し，処理効率の測定を行った．実験にはSUNワークステーションを用い，時間の計測にはtimesシステム関数を用いた．座標平面上に乱数でN個の点を

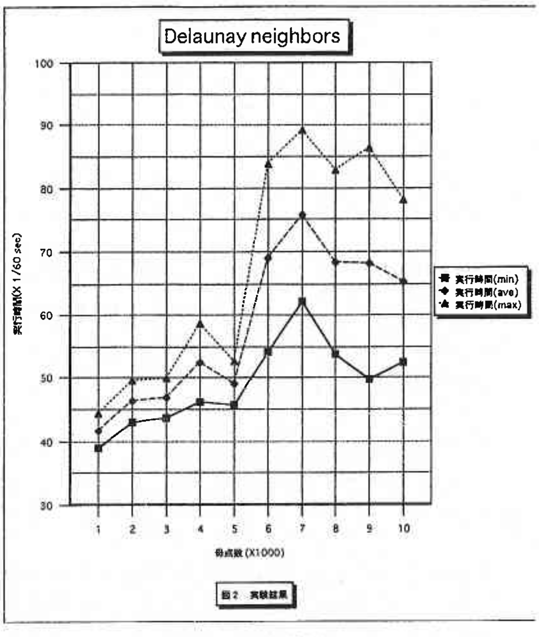

図2　実験結果

発生し，そのうち任意の100個についてドロネーネーバーを算出し，その処理に要した時間を測定した．これを各N（母点数）に対し10回繰り返して平均と分散を調べた．図2により，計算時間は母点の数に対して，不安定な状態になっている．これは，母点の数より母点の分布に敏感であるためと考えられる．

5．おわりに

本稿では，地理処報システムのようにデータの参入や削除が行われる状況で，必要が生じた時に必要となる母点に関するドロネーネーバーを効率よく求める方式を提案した．提案方式では，良好な動的性能を有する他，必要となる小数の母点に対するドロネーネーバーを算出するため，母点の数によらず計算誤差の影響を受けにくいという利点も有している．

(1993年12月10日受理)

参考文献

1) Franz Aurenhammer: Voronoi Diagrams-A Survey of a Fundamental Geometric Data Structure, ACM Computing Surveys, Vol. 23, Nr. 3 (Sep. 1991), 345-403.
2) Leonidas Guibas and Jorge Stolfi: Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams, ACM TOG, Vol. 4, No. 2 (Apr. 1985), 75-123.
3) 杉原厚吉：計算幾何学的手法と画像解析-ボロノイ図の応用を中心として，情報処理，Vol. 30, No. 9 (Sep. 1989), 1067-1075.
4) 大石，杉原：数値的に安定な分割統治型Voronoi図構成算法，情報処理学会誌，Vol. 32, No. 6 (June 1991).