数理解析研究所講究録
第471巻 1982年 10-32

10

計算幾何学について

東大　工学部　浅野　孝夫
ASANO Takao

1. まえがき

計算幾何学（computational geometry）は，幾何学(geometry)に計算量（computational complexity）の概念を導入したものであり，幾何学的な問題を計算機で効率的に処理することを目的としている。幾何学的な問題に対して，問題の規模が小さければ，人間は目の働きで直観的に判断できすぐに解を求めることが可能であるが，計算機はこの種の能力が乏しかった。しかし，近年の目覚しい計算機科学の発展により，計算幾何学の研究も進み，多くの幾何学上の問題が効率的に処理されるようになってきた。

本文では，計算幾何学において最も重要な４つの問題：(i) 重なり問題；(ii) 凸包問題；(iii) 点位置決定問題；(iv) Voronoi線図；に焦点を当てて，最近の研究とその応用面について紹介する。

2. 重なり問題

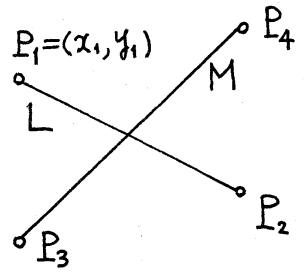
重なり問題とは，図形が与えられたとき，互いに重なる（交わる）図形が存在するかどうかを判定したし，列挙したり，共通部分を求めたりする問題である。

2.1 2個の線分の交差判定

平面上に与えられた2個の線分 $L=(P_1, P_2)$, $M=(P_3, P_4)$ （$P_i=(x_i, y_i)$）が交差するための必要十分条件は

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \cdot \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_4 & y_4 & 1 \end{vmatrix} < 0$$

$$\begin{vmatrix} x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \\ x_1 & y_1 & 1 \end{vmatrix} \cdot \begin{vmatrix} x_3 & y_3 & 1 \\ x_4 & y_4 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} < 0$$



であることが容易に示される。従って，8回の乗算で2つの線分の交差を判定できる。更に，少し工夫すれば6回の乗算で可能である。吉田[1982]は，実は5回の乗算が必要でかつ十分であることを示した。以下，5回の乗算で十分であることを説明する。

P_1 と P_2 を通る直線を L'，P_3 と P_4 を通る直線を M' とし，L' と M' の交点を $Q=(x, y)$ とする。直線 L', M' の方程式は

$$L' = tP_2 + (1-t)P_1 \quad , \quad M' = sP_4 + (1-s)P_3$$

と表せる。Qを表わす，(t,s)を求めるには

$$tP_2+(1-t)P_1=sP_4+(1-s)P_3$$

を解けばよい。従って，

$$t=\Delta_t/\Delta\ ,\quad s=\Delta_s/\Delta$$

である。ここで，

$$\Delta=\begin{vmatrix} x_2-x_1 & y_2-y_1 \\ x_3-x_4 & y_3-y_4 \end{vmatrix}\quad \Delta_t=\begin{vmatrix} x_3-x_1 & y_3-y_1 \\ x_3-x_4 & y_3-y_4 \end{vmatrix}\quad \Delta_s=\begin{vmatrix} x_2-x_1 & y_2-y_1 \\ x_3-x_1 & y_3-y_1 \end{vmatrix}$$

である。Qが線分L，M上にあるための必要十分条件は，

$$0\le t\le 1\ ,\quad 0\le s\le 1$$

であるので，Δ,Δ_t,Δ_sが求まっていれば，加減算と符号の判定だけで，線分L，Mが交差するかどうかを判定できる。

Δ,Δ_t,Δ_sが5回の乗算で求まることを示すには

$$\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}\begin{pmatrix} -b_3 \\ a_3 \end{pmatrix}\ ,\quad a_1b_2-a_2b_1$$

が5回の乗算で計算できることを示せばよい。実際，

$$a_1=x_2-x_1\ ,\quad a_2=x_3-x_4\ ,\quad a_3=x_3-x_1$$

$$b_1=y_2-y_1\ ,\quad b_2=y_3-y_4\ ,\quad b_3=y_3-y_1$$

とおけば，Δ,Δ_t,Δ_sの計算になっている。

$$A_1=(a_1+a_3)(b_2-b_3)\ ,\quad A_2=(a_2-a_3)(b_1+b_3)$$

$$A_3=a_3(b_1-(b_2-b_3))\ ,\quad A_4=a_1b_2\ ,\quad A_5=a_2b_1$$

を用いれば，

$a_1b_2 - a_2b_1 = A_4 - A_5$, $-a_1b_3 + a_3b_1 = A_1 + A_3 - A_4$

$-a_2b_3 + a_3b_2 = -A_2 - A_3 + A_5$

である［吉田(1982)］。

2.2 n個の線分の重なり判定・列挙

平面上に，与えられたn個の線分に対して，互いに交差する線分対が存在するどうかは，素朴な算法で$O(n^2)$の手間で判定できる。ShamosとHoeyは算法を工夫して$O(n\log n)$の手間で判定できることを示した［Shamos & Hoey(1975)］。更に，重なり判定問題の下限が$\Omega(n\log n)$であることを示し，彼らの算法が最適であることを示した。また互いに交差する線分対の個数がkならば，$O(n\log n + k)$の列挙算法があるのではないかと提案した。BentleyとOttmann［1979］はShamos-Hoeyの算法を少し改良して，$O(n\log n + k\log n)$の列挙算法を示した。以下，Shamos-Hoeyの算法を紹介する。

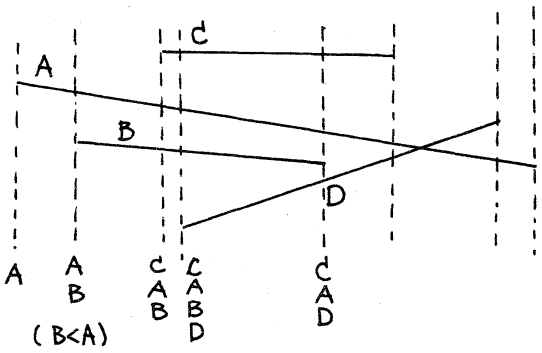
基本的なアイデアは次の3点である。

(1) n個の線分の端点を通りy軸に平行な直線で平面を$2n+1$個の領域，スラブ，に分割する。

(2) y軸に平行な直線$x=u$と交わる線分に対して，その交点の位置の上下関係で順序〝$\leq_u$〟を導入する。

(3) 2つの線分が交差するならば，ある線分の端点を通るy軸に平行な直線における順序でその2つの線分は連続して

いる。



Shamos-Hoeyの重なり算法

1° n個の線分の端点を (x, y) に関する辞書式順序で小さい順にとれるようにpriority queue Qにいれる。線分の上下関係を表現するTを空にする。

2° Qが空になるまで3°－6°を繰り返す。

3° Qから先頭の端点uを取り，uを端点とする線分をSとおく。

4° uが線分Sの左端点ならば5°を実行し，右端点ならば6°を実行する。

5° TにSを挿入する。TでSに連続する2つの線分A,Bに対して，AとSあるいはBとSが交差すれば，"yes"とし，Qを空にする。3°に戻る。

6° TでSに連続する2つの線分A，Bに対して，AとBが交差すれば，"yes"と出しQを空にする。TからSを削除して3°に戻る。

上図にこの算法を適用すれば線分Bの右端点でAとDが交差することを発見する。Tはバランスのとれた木構造のデータ構造を用いる。3°－6°は1回当たり $O(\log n)$ で可能である

ので，全体の手間は $O(n \log n)$ である。

Bentley-Ottmann の算法は交差する線分対が見つかった時点で算法が終了するのではなく，検出された交点もQに挿入して進行する。y軸に平行な直線 $x=u$ の走査で u が線分の端点ならば Shamos-Hoey と同様であるが，u がある線分対A，Bの交点に対応するときには，$x=u$ の走査の前後でA，Bの順序を交換する。このようにして，交差する線分対をすべて列挙する。なお，$k=\Omega(n^2)$ のときには，Bentley-Ottmann の算法は $O(n^2 \log n)$ となり，素朴な算法 $O(n^2)$ より効率的でない。

2.3 応用例

(1) 与えられた n 角形 $P=a_0 a_1 \cdots a_{n-1} a_0$ が単純であるかどうか，すなわち，交差する2辺 $a_i a_{i+1}$，$a_j a_{j+1}$ $(j \neq i, i-1, i+1)$ が存在するかどうかを $O(n \log n)$ で判定できる［Shamos(1978)］。

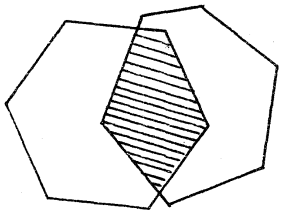
(2) 単純な2つの n 角形P，Qが重なるかどうかを $O(n \log n)$ で判定できる［Shamos(1978)］。

(3) 平面上で描れた直線グラフが端点以外で交差するかどうかを $O(n \log n)$ で判定できる［Shamos(1978)］（n は枝数）。

(4) n 個の単純な k 角形に対して，重なる k 角形の対が存在するかどうか $O(nk \log nk)$ で判定できる（凸角形ならば $O(nk + n \log n \log k)$）［Shamos(1978)］。

(5) n 個の長方形の重なり判定は $O(n\log n)$ で，重なり列挙は $O(n\log n+k\log n)$ で求まる［Bentley & Ottmann (1979)］.

(6) 2つの凸 m 角形と凸 n 角形の重なりは $O(m+n)$ で求まる［Shamos (1978)］。



(7) n 個の半平面の共通部分は $O(n\log n)$ で求まる［Shamos (1978)］。

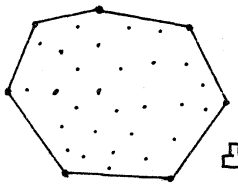
(8) n 個の正方形の交グラフ（intersection graph）の連結成分は $O(n\log n)$ で求まる［Imai (1981)］。

(9) n 個の長方形の交グラフの連結成分と最大クリークは $O(n\log n)$ で求まる［今井 & 浅野 (1981), Imai & Asano (1981)］。

(10) n 個の円の交グラフの連結成分は $O(n\log n)$ で求まる［今井，伊理 & 室田 (1981)］。

3. 凸包問題

k 次元の n 個の点に対して，それらの点すべてを内部に含む最小の凸多面体を凸包（convex hull）という。Graham (1972) は $k=2$ の場合 $O(n\log n)$ で凸包を求められることを示した。以来，凸包問題は活発に研究され，$k=2$ に対しては，



凸包

$O(mn)$（m は凸包の頂点数）の Jarvis の算法［1973］，最悪では $O(n\log n)$ であるが，平均的には $O(n)$ の Shamos［1978］

の算法が得られた。Graham の算法は凸包の内部に1点Oを固定し，Oを原点とするn個の点の偏角を求め，偏角の小さい順に $z_1, z_2, \cdots, z_n$ と並べ，$\angle z_i z_{i+1} z_{i+2}$ が180°未満かどうかを調べながら凸包を構成する。Shamosの算法は，n個の点を勝手に（randomに）2つに分け（大きさは $\lceil n/2 \rceil$ と $\lfloor n/2 \rfloor$），それらについての凸包を用いて全体の凸包を構成するもので分割統治法（divide and conquer）を用いている。Jarvis の算法は凸包の左側外部に1点Oを選び，Oを原点として偏角θの最も小さい点を z_0 とする（$-90° < \theta < 90°$）。以下，z_i を原点として偏角θ（$0° < \theta < 180°$）の最小の点を z_{i+1} と定め，z_0 になるまで繰り返すものである。

2次元のn個の点に対する凸包を（辺まで）求める問題の下限は $\Omega(n \log n)$ である［Shamos (1978)］。これは，凸包問題が解ければ直ちにソーティングの問題が解けるからである。実際，n個の正数 $x_1, x_2, \cdots, x_n$ に対して，$z_i = (x_i, x_i^2)$ $(i = 1, \cdots, n)$ を対応させて，$z_1', z_2', \cdots, z_n', z_1'$ という形で凸包が求まれば，$x_1' \leq x_2' \leq \cdots \leq x_n'$ が得られる（$x_1' \leq x_i : i = 2, \cdots, n$）。これに対して，凸包の頂点集合を求める問題を凸包問題と解釈する立場もある。Yao［1981］はこの凸包問題に対して，$z_i = (x_i, y_i)$ $(1 \leq i \leq n)$ ならば，$x_1, \cdots, x_n, y_1, \cdots, y_n$ の2次以下の多項式 $f(x_1, \cdots, x_n, y_1, \cdots, y_n)$ の計算とその値が正，負また

は0であるかにより分岐することのみを許した"quadratic decision tree model"のもとでは,下限が$\Omega(n\log n)$であることを示した。一方, van Emde Boas は線形関数$f(x_1,\cdots,x_n,y_1,\cdots,y_n)$の計算とその値の正, 負, 0による分岐の"linear decision tree model"でも凸包問題の下限が$\Omega(n\log n)$であると報告した。しかし, 後に, "linear decision tree model"では凸包問題を解くことはできないことが示された[Avis (1980), Jaromczyk (1981)]。x_i, y_iがすべて整数ならば, "linear decision tree model"のもとでも凸包問題が解けて, 下限は$\Omega(n\log n)$である[Avis (1982)]が, "quadratic decision tree model"に対してはまだ結果が得られていない。

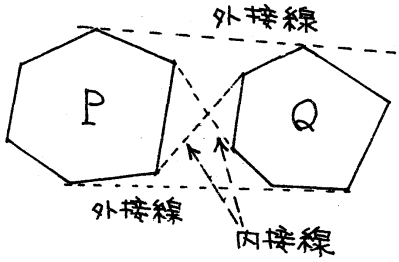
単純なn角形$z_1z_2\cdots z_nz_1$に対する凸包問題は初め$O(n)$で解けると報告された[Sklansky (1972)]が, Bykat [1978]は誤りのあることを指摘した。現在は$O(n)$の算法が与えられている[MaCallum & Avis (1979)]。

3次元以上の空間における凸包問題に対しては, Appel & Will (1976), Preparata & Hong (1977), Toussaint, Akl & Devroye (1978), Chand & Kapur (1970) を参照されたい。4次元の場合, n個の点で定まる凸包は$\Omega(n^2)$の頂点と辺をもつので凸包問題の下限は$\Omega(n^2)$である[Grünbaum (1967), Brown (1978)].

最後に２つの凸包の共通接線を求める算法を紹介する。

２つの凸包の外接線を求める算法［Shamos (1978)］

1° 凸m角形Pの内部に
１点z_0を固定する。

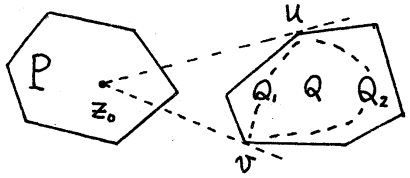


2° z_0が凸n角形Qの内部にある
がどうかを判定する。

3° z_0がQの内部にあれば4°を実行し，外部にあるときには
5°を実行する。

4° PとQのそれぞれの頂点はz_0の回りで偏角順に並んでいる。PとQの全体の頂点をz_0の回りで偏角順になるように併合してリストRをつくる。6°へ行く。

5° z_0からQに接線を引く。
接点をu，vとする。Qを
uとvを結ぶ２つの頂点列Q_1, Q_2
とみなす。Q_1を時計回り，Q_2を反時計回りの順に考えれば，Q_1とQ_2の頂点列はそれぞれz_0の回りで偏角順に並んでいる。３つのz_0の回りで偏角順に並んだ頂点列P, Q_1, Q_2を併合して，z_0の回りで偏角順に並んだリストRをつくる。6°へ行く。



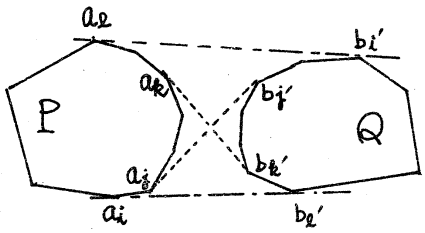
6° Rに関して，Graham［1972］の凸包算法を適用して凸包を求める。新しい２つの辺が外接線である。

この算法の手間は $O(m+n)$ である。Graham の算法のソーティングの部分は必要としないからである。

これに対して，2つの凸多角形の内接線を求める問題は本質的には外接線を求める問題に等価である。

2つの凸包の内接線を求める算法

1° 上記の外接線を求める算法を用いて，凸 m 角形 P と凸 n 角形 Q の共通外接線を求める。それらを，$a_i b_{\ell'}$, $a_\ell b_{i'}$ とおく。新しい凸包の内部に含まれる P の部分を $a_i a_{i+1} \cdots a_\ell$ とし，Q の部分を $b_{i'} b_{i'+1} \cdots b_{\ell'}$ とおく。いずれも反時計回りの順である。



直線 $a_j b_{j'}$, $a_k b_{k'}$ が内接線

2° a_j を求める。求め方は，直線 $a_i a_{i+1}$ と交差する辺を辺 $b_{\ell'} b_{\ell'-1}$ から，辺 $b_{\ell'-1} b_{\ell'-2}$, $b_{\ell'-2} b_{\ell'-3}$ … と進んで見つける。初めて交差する辺を $b_{\ell_1} b_{\ell_1+1}$ とおく。辺 $b_{i'-1} b_{i'}$ まで調べても交差する辺が見つからなかった場合は $a_j = a_i$ とおく。次に直線 $a_{i+1} a_{i+2}$ と交差する辺を辺 $b_{\ell_1} b_{\ell_1+1}$ から出発して見つける。以下これを繰り返して，直線 $a_{i+p} a_{i+p+1}$ が Q の $b_{i'} b_{i'+1} \cdots b_{\ell'}$ の部分と交わらなくなるまで実行する。このとき $j = i+p$ である。

3° 同様にして，a_k, $b_{j'}$, $b_{k'}$ を求める。

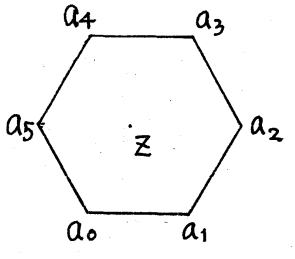
この算法の手間も $O(m+n)$ である。一般に2つの凸多角形の内接線が存在するならば，外接線も容易に求められる。

4. 点位置決定問題

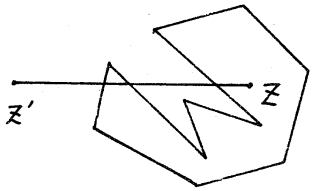
日本地図が与えられているとき，○○市は何県に属するかというような質問に答えるのを点位置決定(point location)という。この種の質問(query)は地理的情報データベース等で頻繁に起こる。質問が同一の地図に対して何度も起こるので，後から起こる質問に備えて地図表現のデータ構造などに適当に前処理を施こす対策が取られている。この時には，前処理に要する計算時間，記憶領域と質問1回当たりの探索時間の3つ手間を総合的に評価する必要がある。この節では，多角形と平面描画グラフの点位置決定算法を紹介する。

4.1 多角形に対する点位置決定算法

単純な多角形 $P = a_0 a_1 \cdots a_{n-1} a_0$ (反時計回り)に対して，点 z が与えられたとき，z が P の内部にあるかどうかを判定する問題を考える。P が凸 n 角形ならば，P の各辺 $\overrightarrow{a_i a_{i+1}}$ に対して点 z が左側にあるかどうかを判定し，すべての辺の左側にあるとき z は P の内部にあることがわかる。そうでないときには z は P の外部にある。この手間は $O(n)$ である。



Pが凸多角形でない場合には，Pの左側外部に1点 $z'=(x', y')$ を取る。ただし，$z=(x, y)$ かつ $y'=y$ とする。z, z' を結ぶ線分 $L(z, z')$ とPの各辺が交差するかどうかを判定し，交差する辺の個数が奇数ならば，z はPの内部にあり，偶数ならば外部にある。この算法の手間も $O(n)$ であるが凸多角形の算法の2倍くらいの時間がかかる。



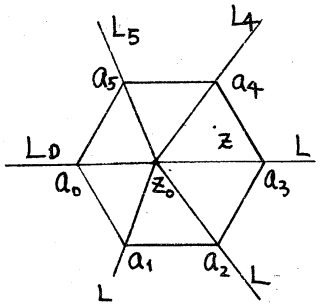
次に前処理を施こす算法について述べる。以下の算法は凸 n 角形に対して，前処理の計算時間が $O(n)$，記憶領域が $O(n)$，1回当たりの探索時間が $O(\log n)$ のものである。

凸多角形の点位置算法 [Shamos & Hoey (1975)]

1° 凸多角形 $P=a_0 \cdots a_{n-1}a_0$ の内部に1点 z_0 を固定する。

2° 点 z_0 と頂点 a_i $(0 \leq i \leq n-1)$ を通る半直線を L_i とする。

3° 半頂角 $\angle a_{i-1}z_0a_i$ が効率的に探索できるように，$\{L_i\}$ を適当なデータ構造で表現する。（以上，前処理）



探索

1° 点 z が与えられたら，$\{L_i\}$ を2分探索し，z が属する半頂角（L_i と L_{i+1} で囲まれた領域）を見つける。

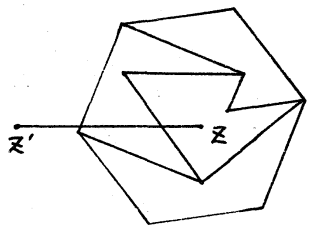
2° 点 z が線分 $\overrightarrow{a_ia_{i+1}}$ の左側にあれば，z はPの内部にあり，

右側にあれば，zはPの外部にある。

この算法は凸でない多角形には適用できない。

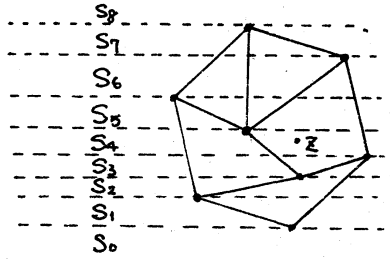
4.2　平面描画グラフに対する点位置決定算法

平面上に直線で描かれた平面グラフ $G=(V,E,F)$ に対して，点zが与えられたとき，zを内部に含む面 $f\in F$ を求める算法を紹介する。素朴な算法は，すべての面 $f\in F$ に対してfが点zを含むかどうかを判定する（一般の多角形に対する点位置決定算法を用いて）。計算の手間は $O(\Sigma_{f\in F}|\Delta f|)=O(2|E|)=O(n)$ である。これに対して，$z=(x,y)$ のとき，Gの外面上に1点 $z'=(x',y')$ を $x'<x$, $y'=y$ となるように選び，線分 $L(z,z')$ と交わる辺をもつ面 $f\in F$ についてのみ，zがfに含まれるかどうかを判定する方法も考えられる。この方法はかなり効率的であると思われる。



$G=(V,E,F)$

次に前処理を施こす算法を述べる。次の表はこれまでに提案された算法の性能評価である。



Shamos & Hoey (1975) の算法は平面グラフGの各点を通りx軸に平行な直線で平面を $n+1$ 個の領域，スラブ（slab）に分割する。各スラブ内にGの点は存在しないので，スラブで切断されるGの辺はそのスラブ内で交差しないので順序を導入で

24

算法	探索時間	記憶領域	前処理時間
Shamos & Hoey (1975)	$O(\log n)$	$O(n^2)$	$O(n^2)$
Lee & Preparata (1977)	$O(\log^2 n)$	$O(n)$	$O(n \log n)$
Lipton & Tarjan (1977)	$O(\log n)$	$O(n)$	$O(n \log n)$
Preparata (1981)	$O(\log n)$	$O(n \log n)$	$O(n \log n)$
Kirkpatrick (1981)	$O(\log n)$	$O(n)$	$O(n \log n)$

きる。スラブ間の順序を表現するデータ構造T_Sと各スラブS_iに含まれる辺の順序を表現するデータ構造T_i ($1 \leq i \leq n-1$) を用意する。それらは2分探索が効率的に実行可能なものとする。点zが与えられたとき，探索はまずT_Sを用いてzの属するスラブS_iを求め，次にS_iのどの辺で囲まれる領域にzが属するかをT_iを用いて判定する。探索は$O(\log n)$の手間で可能であるが，記憶領域が最悪の場合$\Omega(n^2)$必要である。スラブの考え方を生かして，算法を改良したものがLee & Preparata (1977)とPreparata (1981)の算法である。Lipton & Tarjan (1977)の算法は平面グラフの分離定理 (Separator Theorem) を利用したものである。一方，Kirkpatrick (1981) の算法は，平面グラフの面による分割を階層的にとらえ，分割数の小さい分割で点zの属する面fを決定し，次に分割数の小さい分割ではfと重なる面についてのみzが内部にあるかどうかを判定するものであ

る。この算法は非常に華麗で，証明法は Chiba, Nishizeki & Saito (1981) の平面グラフの線形時間5色算法の証明法と共通点があり，いずれも問題の規模を $1/c$ (c: 定数) にするのに $O(n)$ でできその解を用いて最初の問題を解くのも $O(n)$ できることを示し，$T(n) \leq T(n/c) + c'n$ ($c>1$, c': 定数) を解いて $T(n) = O(n)$ を示している。

上記の算法で，Lipton & Tarjan と Kirkpatrick の算法はともに最適であるが，Lipton & Tarjan の算法は非常に複雑でありとても実用的でなく，kirkpatrick の算法は明快だが比例係数が大きくて実用性には問題がありそうだ。

5. Voronoi線図

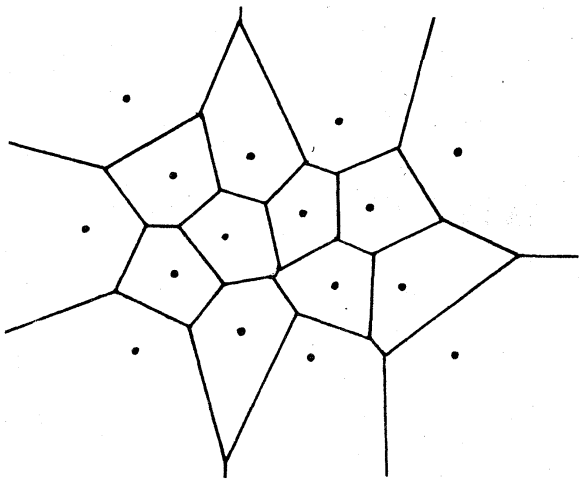
平面上の n 個の点 $z_i = (x_i, y_i)$ ($i = 1, 2, \cdots, n$) に対して，点 z_i からの距離が他のどの z_j からの距離よりも小さいような点の集合を Voronoi 多角形 $V(z_i)$ という。すなわち，

$$V(z_i) = \bigcap_{j=1}^{n} \{z \mid d(z, z_i) \leq d(z, z_j)\}$$

である。

$$d(z, z_i) = \sqrt{(x-x_i)^2 + (y-y_i)^2}$$

のときには，Shamos & Hoey (1975) により $O(n \log n)$ で n 個の点



Voronoi 線図

のVoronoi多角形$\{V(z_i)\}$が求められることが示された。Voronoi多角形で分割された平面図形をVoronoi線図(Voronoi diagram)という。一旦，Voronoi線図が求まれば次の問題が容易に解ける[Shamos & Hoey (1975)]。(5)を除いて手間はすべて$O(n)$。

(1) n個の各点に対して最も近い点全部.

(2) n個の点を結ぶ長さ最小の木.

(3) n個の点の凸包.

(4) n個のどの点も内部に含まない最大の円(円の中心はn個の点の凸包の内部に存在するものとする)。

(5) 新しい点zに最も近い点(Voronoi線図を平面グラフと考えれば，点位置決定問題に一致する)。$O(n\log n)$の前処理を施こせば$O(\log n)$で求まる。

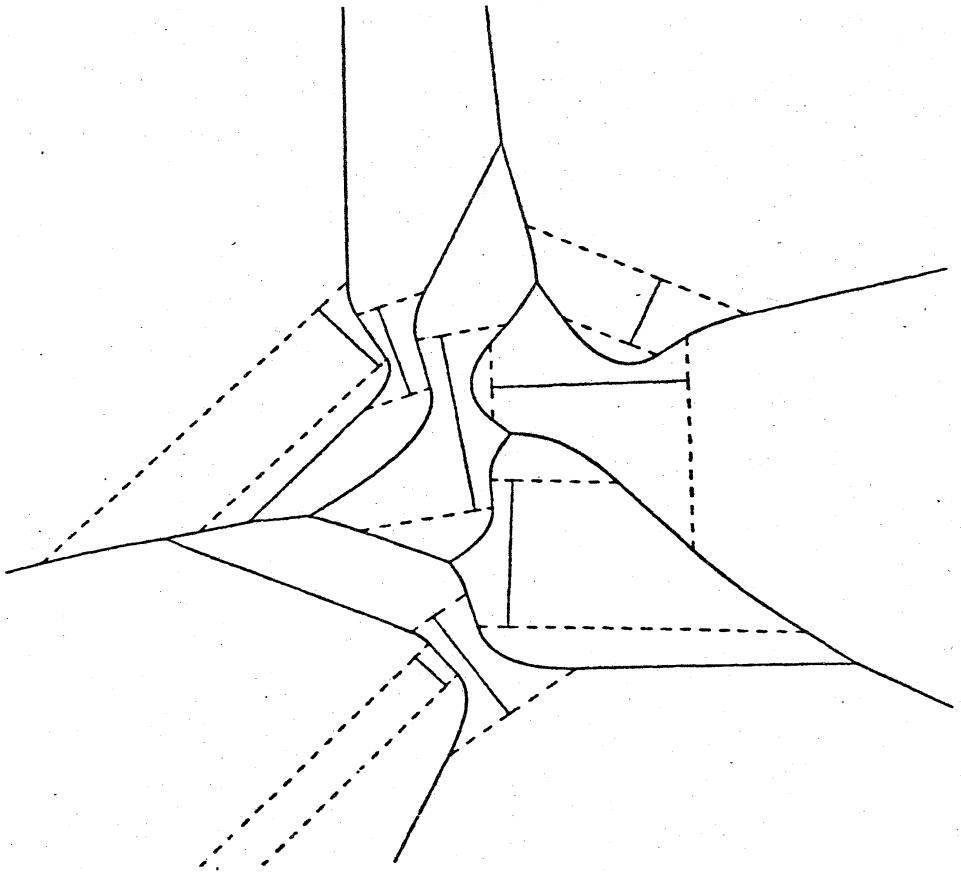
実用的な算法・プログラムも考案されている[Horspool (1979), 大屋 & 室田 (1982)]。特に，大屋，伊理 & 室田(1982)はほぼ平均的に$O(n)$でVoronoi線図が求まることを実証している。

距離関数$d(\cdot,\cdot)$を適当に選ぶことにより，違った形のVoronoi線図が得られる。Lee (1980), Lee & Wong (1981)はL_p-距離のもとでのVoronoi線図を$O(n\log n)$で求めている。今井，伊理 & 室田(1981)は$d(z,z_i)=(x-x_i)^2+(y-y_i)^2-r_i^2$を採用して，Lagerre幾何のVoronoi線図を$O(n\log n)$で求めている。なお$d(\cdot,\cdot)<0$の場合も認めていることに注意されたい。Lagerre幾何のVoronoi

線図はn個の円に関連した問題を取り扱うのに非常に有用である。

3次元以上の空間におけるVoronoi線図も同様に定義でき，効率的な算法も提案されている[Avis & Bhattacharya (1982)]。

一方，線分に関するVoronoi線図も定義され，Lee & Drysdale III (1981)により$O(n\log^2 n)$の算法が提案され，黒田(1982)は実際にプログラミングし，その算法を評価した。下図は黒田による線分のVoronoi線図の出力例である。



線分のVoronoi線図

6. むすび

計算幾何学のパターン認識への応用については Toussaint (1980), (1982) を参照されたい。地理情報処理への応用については，伊理，浅野 & 室田 (1981)，伊理 (1981)，伊理 (1982) を参照されたい。

謝辞　日頃より暖かい御指導を賜わっている東京大学工学部伊理正夫教授に感謝致します。多くの事を教えて頂きました伊理研究室の皆様にも感謝致します。本研究の一部は文部省科学研究費補助金：奨励研究 (A) 57750285 (昭57) の援助のもとで行なわれたものである。

文献

A. Appel and P.M. Will (1976), Determing the three-dimensional convex hull of a polygon, IBM J. Res. Develop., 590-601.

D. Avis (1980), Comments on a lower bound for a convex hull determination, Information Processing Letters, 11, 3, 126.

D. Avis (1982), On the complexity of finding the convex hull of a set of points, Discrete Applied Math., 4, 81-86.

D. Avis and B.K. Bhattacharya (1982), Algorithms for computing d-dimensional Voronoi diagrams and their duals, McGill Univ.

Technical Report, SOCS-82.5.

J.L. Bentley and T.A. Ottmann (1979), Algorithms for reporting and counting geometric intersections, IEEE Trans. Computers, C-28, 9, 643-647.

K.Q. Brown (1978), Voronoi diagrams from convex hulls, Infomation Processing Letters, 9, 5, 223-228.

A. Bykat (1978), Convex hull of a finite set of points in two dimensions, Information Processing Letters, 7, 6, 296-298.

D.R. Chand and S.S. Kapur (1970), An algorithm for convex polytopes, J. of ACM, 17, 1, 78-86.

N. Chiba, T. Nishizeki and N. Saito (1981), A linear 5-coloring algorithm of planar graphs, J. of Algorithms, 2, 317-327.

R.L. Graham (1972), An efficient algorithm for determining the convex hull of a finite planar set, Information Processing Letters, 1, 132-133.

B. Grünbaum (1967), Convex Polytopes, Wiley Interscience, New York.

R.N. Horspool (1979), Constructing the Voronoi diagrams in the plane, McGill University Technical Report, SOCS-79.12.

H. Imai (1981), Finding connected components of an intersection graph of squares in the Euclidean plane, Information Processing Letters, to appear.

今井, 浅野 (1981), 長方形の交グラフの連結成分を求める算

法，日本OR学会1981年度秋季研究発表会アブストラクト集，E-1-6，118－119.

H. Imai and T. Asano (1981), Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane, J. of Algorithms, to appear.

今井，伊理，室田（1981），Voronoi Diagramの一般化，日本OR学会1981年度秋季研究発表会アブストラクト集，E-1-5，116-117.

伊理（1981），ネットワークにおける算法とデータ構造，IBM第2回地域計画と地域データベース・シンポジウム.

伊理（1982），地理情報アルゴリズム，日本OR学会報文集.

伊理，浅野，室田（1981），地理情報処理について，日本OR学会1981年度秋季研究発表会アブストラクト集，E-1-4，114-115.

R.A. Jarvis (1973), On the identification of the convex hull of a finite set of points in the plane, Information Processing Letters, 2, 18-21.

J.W. Jaromczyk (1981), Linear decision tree are too weak for convex hull problem, Information Processing Letters, 12, 3, 138－141.

D. Kirkpatrick (1981), Optimal search in the planar subdivisions, Univ. of British Coloumbia, Technical Report, 81-13.

黒田（1982），一般化ボロノイダイアグラムの構成算法の研究，東京大学工学部計数工学科卒業論文.

D.T. Lee (1980), Two-dimensional Voronoi diagrams in the Lp-metric,

J. of ACM, 27, 4, 604-618.

D.T. Lee and R.L. Drysdale, III (1981), Generalization of Voronoi diagrams in the plane, SIAM J. on Computing, 10, 1, 73-87.

D.T. Lee and F.P. Preparata (1977), Location of a point in a planar subdivision and its applications, SIAM J. on Computing, 6, 3, 594-606.

D.T. Lee and C.K. Wong (1981), Voronoi diagrams in L_1-(L_∞-) metrics with 2-dimensional storage applications, SIAM J. on Computing, 9, 1, 200-211.

R.J. Lipton and R.E. Tarjan (1977), Applications of planar separator theorem, Proc. 18th Annual IEEE Sympo. Foundations of Computer Science, 162-170.

D. McCallum and D. Avis (1979), A linear algorithm for finding the convex hull of a simple polygon, Information Processing Letters, 9, 201-206.

大屋, 室田 (1982), Voronoi Diagram を求める算法とデータ構造の比較・検討, 日本OR学会1982年度春季研究発表会アブストラクト集, 2C-3, 185-186.

大屋, 伊理, 室田 (1982), Voronoi Diagramを求める算法の改良, 日本OR学会1982年度秋季研究発表会アブストラクト集, E-8, 152-153.

F.P. Preparata (1981), A new approach to planar point location, SIAM J. on Computing, 10, 3, 473-482.

F.P. Preparata and S.J. Hong (1977), Convex hulls of finite set of

points in two or three dimensions, C. of ACM, 20, 2, 87-93.

M.I. Shamos (1978), Computational Geometry, Yale Univ, Ph.D. Thesis.

M.I. Shamos and D. Hoey (1975), Closest-point problems, Proc. 16th Annual IEEE Sympo. Foundations of Computer Science, 151-162.

J. Sklansky (1972), Measuring concavity on a rectangular mosaic, IEEE Trans. Computers, C-21, 1355-1364.

G.T. Toussaint (1980), Pattern recognition and geometrical complexity, 5th International Conference on Pattern Recognition, 1324-1347.

G.T. Toussaint (1982), Computational geometric problems in pattern recognition, in J. Kittler, K.S. Fu, and L.F. Pau (eds.), Pattern Recognition Theory and Applications, D. Reidel Publishing Company 73-91.

G.T. Toussaint, S.G. Akl and L.P. Pevroye (1978), Efficient convex hull algorithms for points in two and more dimensions, McGill Univ. Technical Report, SOCS-78.5.

A.C. Yao (1981), A lower bound to finding convex hulls, JACM 28, 4, 780-787.

吉田 (1982), 伊理研究室合同輪講資料 (1982-06-22).