

組合せ構造を優先した多角形 Voronoi 図の構成法

今井敏行

東京大学 工学部

点, 線分, 折れ線, 多角形を生成元とする Voronoi 図を構成する算法を示す. 位相優先法に基づく線分 Voronoi 図の構成算法を拡張することによって実現した. 図形のもつ組合せ構造を優先的に扱っている. 一般の算法で正常な出力が得られる程度の計算精度の下では, この算法でも正常な出力を同等の計算量と記憶量で得ることができる. また, どのような計算精度の下でも正常終了し, 出力が本来もつ組合せ構造のいくつかを保証する. この意味で本算法は計算誤差に対して強い. また誤差を前提にする算法であるため, 入力が退化している状況でも算法は正常に働き, 退化のための例外処理が不要である.

A Combinatorial-Structure Oriented Algorithm
for Voronoi Diagrams of Polygons

Toshiyuki Imai

Faculty of Engineering, University of Tokyo

An algorithm to construct Voronoi diagrams whose generators are points, line segments, chains and polygons is shown. This algorithm is adapted from a combinatorial-structure oriented algorithm for Voronoi diagrams of line segments. In this algorithm, Combinatorial structure of the diagrams is considered as prior information to numerical values. If the precision in computation is high enough for conventional algorithms to work well, this algorithm also constructs the correct Voronoi diagrams. Moreover, even when precision in computations is not enough, this algorithm ends up with a diagram as an output with several combinatorial properties which Voronoi diagrams must have. In this sense, this algorithm is robust against numerical error. This algorithm is designed in the assumption that numerical error exists; hence it has not any exceptional rules for degeneracy but still works for degenerate cases.

1 はじめに

平面上に与えられた有限個の点に対し、それらの点の勢力圏による平面の分割を Voronoi 図という。Voronoi 図は計算幾何学の中心的な研究対象であるとともに、画像処理、数値解析など多くの分野に実用されている [5, 6]。Voronoi 図の自然な拡張として、初めに与える有限個の点を他の図形へ一般化した Voronoi 図を考え、その構成法を確立することは理論と実用の両面から求められている。特に実用面からは、単に計算量のオーダーが小さいだけでなく、実際に速く、また、数値的に安定に動作する算法が要求されている。そのような要求に応えるものとして、位相優先法という設計手法を基礎に、点に対する Voronoi 図 [4, 8] や線分に対する Voronoi 図 [3] の構成算法が提案され計算機上に実装されている。本稿では、位相優先法を基礎にした、点、線分、折れ線、多角形に対する Voronoi 図の構成算法を示す。この算法は実数計算に誤差があってもデータ破壊や無限ループや異常終了を生じないという特徴を持つ。

2 Voronoi 図と諸定義

まず、本稿で扱う各種の Voronoi 図の定義と記号の導入を行なっておく。本稿では多角形は辺集合として考える。

g_i ($i = 1, \dots, N$) を平面 \mathbf{R}^2 上の点、線分、折れ線または多角形で、次の生成元条件をみたすものとする：

生成元条件: 各 g_i は単純で、任意の g_i, g_j ($i \neq j$) は共有点を持たない。

ここで、図形が単純であるとは、図形の辺が自己交差をもたないことである。グラフ理論の言葉を使って、生成元条件は、生成元が線分を枝とする平面グラフで、各頂点に接続する枝が 2 個以下である、ともいえる。集合 $G = \{g_1, \dots, g_N\}$ を生成元集合、各 g_i を生成元とよぶ。

平面 \mathbf{R}^2 上の点 v と生成元 g_i の距離 d を、 $d(v, g_i) = \inf\{\|u - v\| \mid u \in g_i\}$ で定める。ただし $\|\bullet\|$ は Euclid ノルムを表す。 $i \neq j$ のとき、領域 $R(g_i, g_j)$ を次のように定める：

$$R(g_i, g_j) = \{v \in \mathbf{R}^2 \mid d(v, g_i) < d(v, g_j)\}. \quad (1)$$

$R(g_i) = \bigcap_{j \neq i} R(g_i, g_j)$ とおく。 $R(g_1), \dots, R(g_N)$ が定める平面の分割を $V(G)$ で表し、 G に対する (一般化) Voronoi 図という。 $R(g_i)$ を生成元 g_i の Voronoi 領域という。2 個の Voronoi 領域の共通境界である曲線分を Voronoi 辺とよぶ。また、3 個以上の Voronoi 領域に囲まれた境界上の点を Voronoi 点とよぶ。これらの Voronoi 点、Voronoi 辺はグラフ的な構造をもつ。以下では、記述を簡潔にするため、グラフの用語を用いて、Voronoi 領域、Voronoi 辺、Voronoi 点を、単に、領域、辺、節点とよぶ。また本稿で扱う問題を次のように設定する。

問題: 単位正方形 $[0, 1]^2$ 内に与えられた生成元に対し、多角形 Voronoi 図をこの正方形内に描け。

生成元に適当な縮小と平行移動を施すことを考えると、このように生成元の範囲を設定しても一般性を失わないことがわかる。

ここで、算法の説明のため分割線分 Voronoi 図とよぶもう一つの平面分割を導入する。生成元集合 G の各生成元 g_i から、

1. g_i が点ならば自分自身、
2. g_i が線分ならば両端の 2 点、
3. g_i が折れ線ならば両端の 2 点と角の頂点全部、
4. g_i が多角形ならば角の頂点全部

を集めてできる集合(点だけからなる)を $G_0 = \{p_1, \dots, p_m\}$ とする. 各 g_i から G_0 の元になっている点を取り除くと, g_i が点の場合は空となり, それ以外の場合は線分から両端の点を除いたものがいくつか得られる. この, 両端点を除いた線分を開線分とよぶ. 各 g_i から G_0 の元を取り除いて得られる開線分の全体を $\{e_1, \dots, e_n\}$ とする. また, 各 k ($0 < k \leq n$) に対して $G_k = G_0 \cup \{e_1, \dots, e_k\}$ とかくことにする. $p, q \in G_n$ に対して, 多角形 Voronoi 図と同様に, $R(p, q)$ を以下で定める:

p, q が接続する点と開線分するとき ($\{p, q\} = \{p_i, e_k\}$ とし, e_k の両端点を p_i, p_j とする),

$$R(p_i, e_k) = \{x \in \mathbf{R}^2 \mid (x - p_i) \cdot (p_j - p_i) < 0\}, \quad (2)$$

$$R(e_k, p_i) = \{x \in \mathbf{R}^2 \mid (x - p_i) \cdot (p_j - p_i) > 0\}. \quad (3)$$

それ以外のときは, 多角形 Voronoi 図での定義式 (1) で定める. $R(q) = \bigcap_{p \neq q} R(q, p)$ とおく. $R(q)$ ($q \in G_k$) が定める平面の分割を G_k に対する分割線分 Voronoi 図とよび, G_k に対する分割線分 Voronoi 図も $V(G_k)$ とかく. この記法は多角形 Voronoi 図と同一であるが, 各種の Voronoi 図は生成元集合で区別することにし記号を共用にする. また生成元, 生成元集合, 領域, 辺, 節点などの用語も共用する.

多角形 Voronoi 図 $V(G)$ を分割線分 Voronoi 図 $V(G_n)$ から得るには, 辺のうち, 両側の生成元の端点や開線分が同一の多角形に属するものを消去するだけですむ (図 1). そこで以後では分割線分 Voronoi 図の構成を扱う. また, 特に分割線分 Voronoi 図 $V(G_n)$ において, 開線分とその端点の領域の間の辺を仮想辺とよぶことにする.

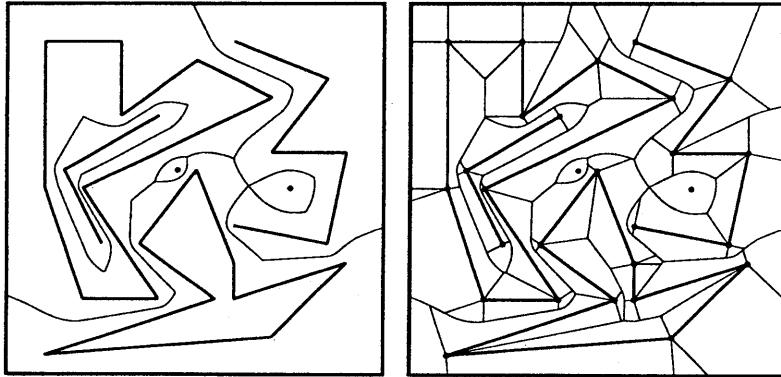


図 1: 多角形 Voronoi 図とその分割線分 Voronoi 図.

Voronoi 図には無限にのびる辺が存在する. これがグラフと Voronoi 図で構造の異なる点である. 本稿の算法では, 正方形 $[-1, 2]^2$ を内部に含む三角形の 3 頂点を生成元集合に加えることにする. このような 3 点によって, 単位正方形 $[0, 1]^2$ の内部で Voronoi 図に影響を与えることなしに, 無限にのびる辺を 3 個に限定できる.

3 位相優先法と構成算法

本稿で実現した算法は位相優先法に基づく. この位相優先法とは, 幾何的算法を浮動小数計算を利用してデータ構造を加工していく課程とみなし, データが本来持つはずの組合せ的な構造の確保を一義的に扱う幾何的算法の設計指針である.

従来の幾何的算法の設計法では, 実数の計算であっても誤差なく実行できることを前提としている. そのため, 算法の内部ではデータ構造は, 矛盾なく加工されていくはず, として扱われてい

る。このデータ構造の扱いが算法の実装時に問題となる。すなわち、誤差に起因して、矛盾する判定、無限ループや異常終了、データ破壊を引き起こす。位相優先法はデータの組合せ的な構造を確保することによって、この問題を回避する。

本稿で示す多角形 Voronoi 図の構成算法は、線分 Voronoi 図の構成算法 [3] を拡張したものである。以下、拡張した部分を中心にその算法と確保した組合せ構造について説明する。

計算誤差を念頭におくので、算法を実装する計算機として、組合せ計算 (整数計算) が正確にでき、実数計算に誤差はあるが同一の計算には同一の結果を返す計算機を仮定する。また入力生成元集合はこの計算機で交差しないと判定され、すべての開線分はこの計算機で 0 でない長さが計算されるとする。これらの仮定は計算機の現実に近い。

本稿の多角形 Voronoi 図の構成算法のもとになった線分 Voronoi 図の算法の概略は次のようである。開線分を生成元集合に 1 個ずつ加えていく逐次添加型の算法である。

算法 1 (線分 Voronoi 図の構成)

1.1 点の Voronoi 図 $V(G_0)$ を作る。

1.2 $k = 1, \dots, n$ の順に 1.2.1 を行なう。

1.2.1 開線分を e_k を添加し $V(G_{k-1})$ を $V(G_k)$ に描き換える。

これで最終的に分割線分 Voronoi 図 $V(G_n)$ を得るというものである。前節で述べたように、この $V(G_n)$ から所要の Voronoi 図は不要な辺を消去するだけで求められる。算法 1 中で 1.1 の点の Voronoi 図 $V(G_0)$ の構成は位相優先法の既存算法 [8] を用いる。したがって、今回拡張した部分は 1.2.1 の開線分を添加し Voronoi 図を描き替えていく部分である。

線分 Voronoi 図の構成算法におけるこの部分の骨格を以下で説明していく。開線分 e_k を添加するとする。あらかじめ各開線分に任意に向きを付け、その始点、終点の端点が求められるデータ構造を採用しておく。始点側端点を前 endpoint、終点側端点を後 endpoint とよぶことにする。開線分 e_k の前後の端点を p_i, p_j とする。このときに Voronoi 図の変化する部分は次のような組合せ的な構造をもつ (図 2):

- 新しく生じる辺や節点は消去される部分を囲む閉路である、
- その消去される部分は木構造をもつ、
- その木は両端点の領域をつなぐただ一つの道をもつ、
- その道は開線分 e_k の左右にある生成元の領域の間を通る。
- 仮想辺はどの辺も少なくとも一部分は残る。

本稿の算法ではこれらの構造を逆順に利用する。線分 Voronoi 図の構成算法では、開線分 e_k を添加し Voronoi 図を描き替える部分は次のとおりである:

算法 2 (開線分 e_k の添加)

2.1 前 endpoint 領域 $R(p_i)$ の周上で道の出発点を決める。

2.2 探索点を出発点におく。

2.3 探索点が後 endpoint 領域 $R(p_j)$ に達するまで 2.3.1 を行ない、道を発見する。

2.3.1 探索点に接続する未探索辺のうち開線分 e_k に関して左右の生成元の領域の間の辺を探索し現探索点でない側の節点をあらためて探索点にする。

2.4 道上の節点から消去される辺を探索していき、消去される木構造を決定する。

2.5 消去される木構造を囲むように新しい節点や辺を付加し木構造を消去して、添加開線分 e_k の領域を決定する。

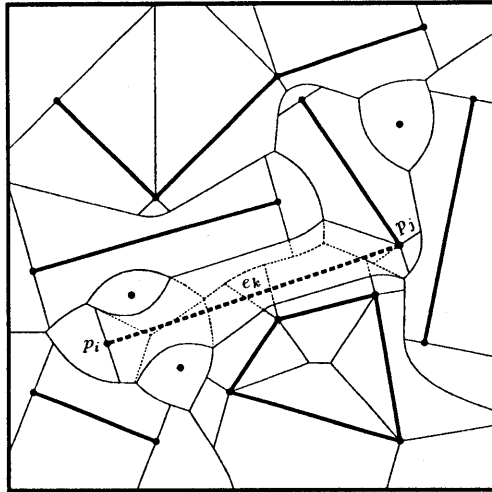


図 2: 開線分の添加.

ただし、2.3 の道を見つけていく課程で領域 $R(p_j)$ に道が達する前に仮想辺や無限にのびる辺を選んだり、閉路を作ったり、出発点の領域 $R(p_i)$ に戻ってきたときには、左右の判定のもっとも怪しそうな所から探索をやり直す。また、道を見つけた後、2.4 で木構造を決定するために探索している辺も、仮想辺や無限にのびる辺に達したり、閉路を作りそうになったときには強制的にその辺の途中で新しく節点が生じるものとし、消去される部分が木構造をもつことを確保する。このようにすることで次のような組合せ的な性質が保証された分割線分 Voronoi 図をどのような計算誤差の下でも得ることができる:

- すべての点、開線分が単連結な領域をもつ、
- 開線分とその端点の領域は必ず仮想辺で隣接する。

この線分 Voronoi 図の算法を多角形 Voronoi 図の算法に拡張する。まず、開線分 e_k の添加前に、前 endpoint p_i で e_k に隣接する開線分 e_l が添加されるとする。このとき、 e_k が添加されるとき、算法 2 中の 2.1 で道の出発点を決めるときには前 endpoint 領域 $R(p_i)$ の周上では仮想辺に接続する節点のうち片方は絶対に出発点にならない(図 3)。終点領域 $R(p_j)$ でも同様に、 e_k の添加前に p_j に接続する他の開線分が添加されるならば、絶対に出発点にならない節点があることがいえる。多角形 Voronoi 図の構成算法ではこれらの節点を道の探索で採用しないことにした。もう一つ問題となるのは、線分 Voronoi 図とは違い多角形 Voronoi 図の構成では、計算誤差によって仮想辺を経由しないと、算法 2 中の 2.3 で道の探索が行き詰まる場合があることである(図 4)。これは、開線分添加の過程で 1 個の多角形のすべての辺が添加され、他の未添加の開線分の端点がこの多角形の内部と外部に分離されてしまうからである。このため、入力の開線分に前処理をして、多角形を閉じる辺を添加順序で最後になるよう並べかえるようにした。このようにして、どのような計算誤差でも正常終了する多角形 Voronoi 図の構成算法を得ることができる。

位相優先法は算法を意識的に数値計算部分と組合せ構造構築の部分にわけるので、絶対に正常終了し、組合せ的な構造を保証する算法が得たのちに、数値計算部分に計算上の工夫を加えることができる。線分 Voronoi 図における数値計算上の工夫は文献 [3] にあるので本稿では、線分 Voronoi 図から多角形 Voronoi 図へと構成算法を拡張したときに変更を加えた数値計算部分について述べる。

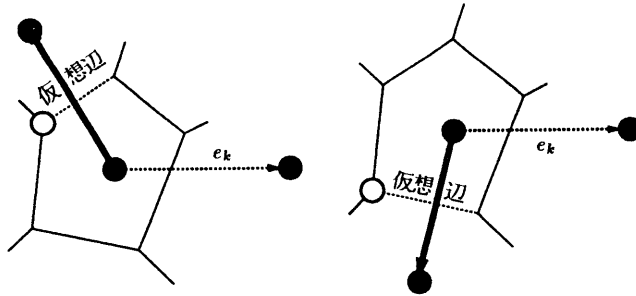


図 3: 出発点にならない節点 (白丸).

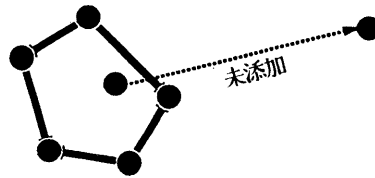


図 4: 多角形で分離される端点.

線分 Voronoi 図の構成算法では、探索中の道の先端の節点 v から生成元の点や開線分が添加開線分 e_k に対し左右どちらにあるかを判定し、次に探索する辺を決定している。この左右判定では生成元が開線分 e_l のときには節点 v から e_l に下ろした垂線の足を p とし、生成元が点のときにはその点を p とし、 p が e_k から左右どちらにどのくらい離れているか計算してその値を左右判定の度合いとしている (図 5)。特に前者の場合、すなわち p が垂線の足であるときには多くの計算が必要で、誤差の影響を強く受けると考えられる。そこで e_k と e_l が隣接するときには垂線の足を求めず、この 2 個の開線分の端点を p_i, p_j, p とし、 p の e_k に対する左右判定を行ない、左右判定の度合いとしてはより信頼性が高いと判断し、一定の値を与えることにした。

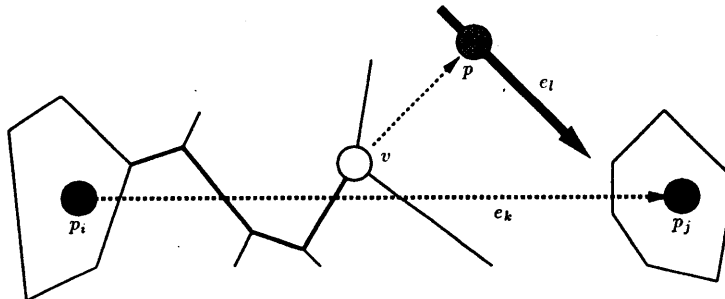


図 5: 生成元の開線分に対する左右判定.

他には、例えば入力では線分 Voronoi 図の場合、初めの 2 点が第 1 線分、次の 2 点が第 2 線分、のように解釈して、頂点座標の列で入力を済ませることができ、多角形 Voronoi 図の場合、入力として点の座標値の他に開線分と端点の接続関係も必要になり、入力はより複雑になる。出力も同様である。そのような算法内部に大きくは影響しない変更点に関しては詳細を省く。

計算誤差の存在を仮定するということは、入力退化を発見できないことを意味する。したがっ

て、本稿の算法では入力退化のための特別な処理を用意していない。しかしながら、本稿の算法は、退化した入力に対しても非退化入力と同一の処理をして正常終了する。

4 計算量と記憶量

多角形 Voronoi 図の構成算法の計算量について考える。多角形 Voronoi 図に対応する分割線分 Voronoi 図では線分 Voronoi 図と同じく、節点、辺の総数は領域の総数の線形オーダーである。線分 Voronoi 図の構成算法にかかる時間は、生成元数を n とすると、点 Voronoi 図の構成に $O(n^2)$ 、すべての開線分の添加に $O(n^3)$ かかり、探索のやり直しが起こらない程度の計算精度があれば開線分の添加も $O(n^2)$ ですむ [3]。記憶量は $O(n)$ である。多角形 Voronoi 図では算法の拡張は計算量と記憶量のオーダーにかかわらない部分に施されている。したがって分割線分 Voronoi 図の生成元の点と開線分の総数を N とすると、算法の計算量は最悪で $O(N^3)$ 、ある程度の計算精度があれば $O(N^2)$ である。また記憶量は $O(N)$ である。計算量は最適の $O(N \log N)$ [1] に及ばないが、従来の逐次添加型の構成算法の計算量と同等である。記憶量の $O(N)$ は最適である。

算法を実装し、実際の計算量を測定する実験を行なった。入力は多角形だけからなるもので、基本的に次の算法で得た。

算法 3(入力多角形集合の構成)

- 3.1 端点座標を $[0, 1]$ 内の一様乱数で発生させ、端点 p_1, \dots, p_n をとる。
- 3.2 初期開線分 e_1, \dots, e_n を、 $e_i = (p_i, p_{i+1})$ ($i = 1, \dots, n-1$); $e_n = (p_n, p_1)$ にとる。
- 3.3 交差開線分対 (e_i, e_j) が見つかるうちは、3.4 を実行する。
 - 3.4 e_i, e_j の後端点を交換しても二角形が生じないとき、
 e_i, e_j の後端点を交換する。
 - そうでないとき、
 e_i の前 endpoint と e_j の後端点を交換する。

ただし、ここで二角形とは2個の開線分であって、向きを無視して両端点が一致するものをいう。算法 3 中の 3.3 での交差線分対 (e_i, e_j) の探索には2通りの方式をとった。第1の方式は交差開線分対の添字対 (i, j) の辞書式順序で最小のものを探すもので、総当り方式とよぶことにする。第2の方式は文献 [2, 7] による平面走査法によるもので、平面走査方式とよぶことにする。両者の方式で得られた開線分の総数 1024 での例を図 6 に示す。一見して開線分の分布に差があることがわかる。各方式、各線分数ごとに10個の異なる生成元集合での計算時間を測定した。結果は各方式、各線分数とも計算時間のばらつきが小さかったので、計算時間の平均をとり結果を表 1 に示す。両方式で若干の差はあるが、計算量は開線分数の2乗のオーダーよりむしろ線形に近いことがわかる。

表 1: 計算機実験の結果 (単位は秒)。

開線分数	256	512	1024	2048	4096
総当り方式	2.94	5.90	11.87	24.50	51.78
平面走査方式	3.11	6.31	12.73	26.64	57.19

5 まとめ

位相優先法による線分 Voronoi 図の構成算法を拡張して多角形 Voronoi 図構成算法を考案し計算機上に実装した。この算法は、どのような計算精度でも正常終了し、本来の多角形 Voronoi 図

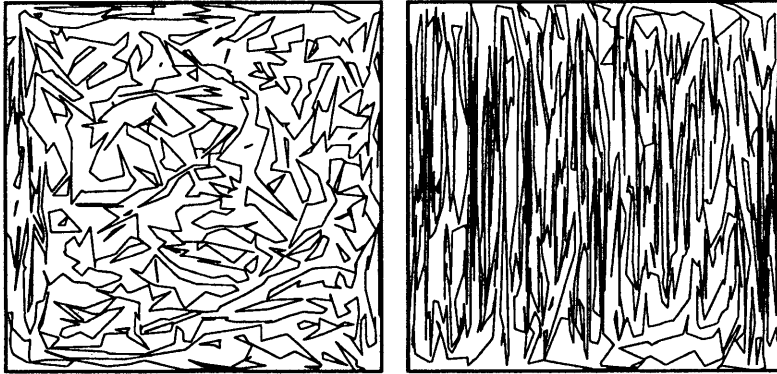


図 6: 総当り方式(左)と平面走査方式(右)(開線分数 1024).

がもつ組合せ構造のいくつかを保証する。計算量は対応する分割線分 Voronoi 図の生成元の総数を N とすれば、従来の逐次添加型の算法が動く程度の計算精度があれば、従来と同等の $O(N^2)$ であり、任意の計算精度でも $O(N^3)$ である。2 種類の方法で生成した多角形入力に対しては、計算機実験の結果、計算量のオーダーは $O(N^2)$ よりかなり小さく、むしろ線形に近い。今後の課題としては、3 個以上の開線分が端点を共有する場合への拡張があげられる。この場合、入力データの組合せ構造の部分だけで矛盾したデータが存在し得るので、入力データが組合せ的に無矛盾であることを算法中で確認するか、前提として要求する必要がある。また、各開線分がどの多角形に属するか指定しなければ決定できないので、その指定のため入出力の再度の拡張が必要であることもわかっている。

参考文献

- [1] Aurenhammer, F.: Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure, *ACM Comput. Surv.*, vol. 23, pp. 345-405(1991).
- [2] Bentley, J. L., and Ottmann, T., Algorithms for Reporting and Counting Geometric Intersections, *IEEE Trans. Comput.*, C-28, pp. 643-647.(1979)
- [3] 今井敏行, 杉原厚吉: 誤差による破綻の心配のない線分 Voronoi 図の構成算法, *情報処理学会論文誌*, vol. 35, No. 10, pp. 1966-1977(1989).
- [4] Inagaki, H., Sugihara, K. and Sugie, N.: Numerically Robust Incremental Algorithm for Constructing three-dimensional Voronoi Diagrams, *Proc. 4th Canadian Conference on Computational Geometry*, pp. 334-339(1992).
- [5] 伊理正夫(監修), 腰塚武志(編集): *計算幾何学と地理情報処理*, 第 2 版, 共立出版 (1993).
- [6] Okabe, A., Boots, B. and Sugihara, K.: *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons(1992).
- [7] Sedgewick, R., *Algorithms*, 2nd ed, Addison-Wesley, (1988).
- [8] Sugihara, K. and Iri, M.: Construction of the Voronoi Diagram for "One Million" Generators in Single-precision Arithmetic. *Proc. IEEE*, vol. 80, pp. 1471-1484(1992).