

3次元ボロノイ図構成のための数値的に安定な逐次添加法

稲垣 宏[†] 杉原厚吉^{††} 杉江 昇^{†††}

幾何図形を扱う効率のよいアルゴリズムが多く提案されているが、それらは数値誤差が発生しないという仮定のもとでのみ正しさが証明されたものであるため、計算機プログラムに翻訳すると計算誤差のために正しく動作しないことがある。本論文では3次元ボロノイ図を対象とし、これを構成するアルゴリズムの数値的安定化を試みた。従来のアルゴリズムが破綻するのは、誤差を含んだ数値計算結果を信用し過ぎるためであることを反省し、ここでは、計算誤差を含む数値判定より幾何図形の持つべき位相構造の保持を優先させた。本方針に基づいたアルゴリズムを設計し、それを実際に計算機へインプリメントして計算機実験によりその振舞いを観察した。その結果、従来の算法ではほとんどの場合に処理が破綻をきたすような悪条件の入力に対しても、位相的に矛盾のない結果が出力されることが確かめられ、その数値的安定性が実証された。

Robust Algorithm for Incremental Construction of Three-Dimensional Voronoi Diagrams

HIROSHI INAGAKI,[†] KOKICHI SUGIHARA^{††} and NOBORU SUGIE^{†††}

Many efficient geometrical algorithms have been proposed. However, these algorithms were designed on the assumption that numerical errors do not take place, and hence computer programs based on these algorithms often fail because of computational errors. Here, a numerically stable algorithm for constructing three-dimensional Voronoi diagram is proposed. In this algorithm higher priority is placed on the topological structure than on numerical values, so that, no matter how large numerical errors may take place, the algorithm will never come across topological inconsistency and thus can always complete its task. Furthermore, the output of the algorithm "converges" to the true Voronoi diagram as the precision in computation becomes higher. The proposed algorithm was implemented into a computer program, and numerical robustness of the algorithm was proved by computational experiments.

1. はじめに

幾何図形を計算機で扱うにあたって、効率のよいアルゴリズムを見つけようとする努力が、計算幾何学と呼ばれる分野において活発になされ、多くの成果を上げている。しかし、そのようなアルゴリズムをそのまま計算機にインプリメントすると、計算誤差が原因となり処理が破綻してしまうことがある。計算機は有限精度の値を扱うものであるために、理論的には正しいアルゴリズムであっても、計算機プログラムに翻訳し

た際には必ずしも正しく動作するとは限らない。最近になって、このような理論と現実のギャップを埋めようとする試みが、平面ボロノイ図構成算法、ソリッドモデリングなどに対してなされている^{1)~5)}。

本稿では、文献 1), 2), 5) で2次元ボロノイ図構成問題に対して提案された考え方を3次元へ拡張し、3次元ボロノイ図を構成するための計算誤差に強いアルゴリズムを提案する。3次元ボロノイ図は、空間内の補間、点パターンの解析、3次元有限要素法のためのメッシュ生成、3次元図形の薄面化、および骨格線抽出、ロボットの衝突回避作業計画など、多くの応用をもっており^{6)~13)}、本稿はそのための数値的に安定な道具を提供するものである。

そもそも幾何図形処理が破綻するのは、計算誤差のために幾何図形の持つ位相構造が破壊されるからであり、その位相構造を保持するような処理だけを進めていく限りにおいては、処理は破綻しない。そこで本算法では、数値計算結果よりも図形の持つ位相的性質を

[†] 豊田工業高等専門学校情報工学科
Department of Information and Computer Engineering, Toyota College of Technology

^{††} 東京大学工学部計数工学科
Department of Mathematical Engineering and Information Physics, Faculty of Engineering, University of Tokyo

^{†††} 名古屋大学工学部情報工学科
Department of Information Engineering, Faculty of Engineering, Nagoya University

優先させ、位相構造を破壊するような数値計算結果は採用しないようにすることにより、数値的安定化を実現させている。2次元ポロノイ領域(多角形)の境界と異なり、3次元ポロノイ領域(多面体)の境界には自然な順序がつかないなどの差があるため、ここで必要な位相的性質の保持は、2次元の場合^{1),2),5)}の単純な延長ではなく、本質的に新しい工夫を含むものである。また、本算法を実際に計算機にインプリメントし、計算機実験によりその数値的安定性を確認したので、それも併せて報告する。

2. ポロノイ図とその構成算法

2.1 ポロノイ図

d 次元空間の有限点集合 $S = \{p_1, p_2, \dots, p_n\}$ に対して、 $V(p_i) = \{x | x \in R^d, d(x, p_i) < d(x, p_j), j = 1, 2, \dots, n, j \neq i\}$ (ただし、 $d(x, p_i)$ は、点 x と点 p_i のユークリッド距離を表す) を、点 p_i のポロノイ領域といい、ポロノイ領域 $V(p_1), V(p_2), \dots, V(p_n)$ が定める d 次元空間の分割を、 S に対するポロノイ図という。このとき、 S の要素を母点という。すなわち、ポロノイ領域 $V(p_i)$ は、他のどの点よりも点 p_i に近い点全体のなす領域であり、点 p_i の勢力圏とよばれることもある^{6),7)}。3次元の場合には、 $V(p_i)$ は多面体となり、ポロノイ図は3次元空間の多面体分割を与える。この多面体の頂点をポロノイ点、辺をポロノイ辺、面をポロノイ面とよぶ。

ポロノイ図は距離の概念に基づいた母点の勢力圏分布を表しているので、ポロノイ領域が境界を共有する2個の母点は“隣り合っている”と見なすことができる。この性質を利用して、ポロノイ図は、補間・点パターンの解析・領域分割などの広い応用分野を持っている^{6),7)}。

2.2 ポロノイ図構成算法

これまで提案されているポロノイ図構成算法のうちで実用的な意味で最も効率がよいとされているのは母点逐次添加法である。これは、数個の母点に対するポロノイ図から出発して、母点の一つずつ添加しながらポロノイ図を逐次更新していく方法である。この方法にデータ構造上の工夫を加えれば、ランダムに配置された n 個の母点に対しては、平均的に $O(n)$ の時間でポロノイ図を構成できる⁸⁾。

ここで、母点逐次添加法を効率よく実行するためには、母点添加時において添加済みの母点が常に空間内に一様に分布している必要がある。本稿では、これを

実現するために、バケットと八分木のデータ構造を用いる。これは、文献8)で提案された方法の3次元への拡張である。まず、母点が分布する空間を母点の数に比例したバケットに分割し、それらのバケットを葉とする八分木を用意する。この八分木の各ノードに対して、それを根とする部分木の葉全体に対する領域をそのノードの支配領域とよぶ。各ノードおよびバケットにはその支配領域内の母点を格納しておく。そして、この八分木をもとに次の手順で母点の添加を行う。まず、深さ1の8個のノードの支配領域よりそれぞれ1個ずつ母点を添加する。次に深さ2の64個のノードの支配領域よりそれぞれ1個ずつ母点を添加し、同様のことを葉の深さまで繰り返す。もしも該当する支配領域に母点がなければそこは飛ばしていく。このように母点添加を行えば、添加済みの母点が常に一様に分布しているような状態を保つことができる。

3. 3次元ポロノイ図構成算法の数値的安定化

母点逐次添加法をはじめとする従来のポロノイ図構成算法は、母点同士が特殊な位置関係(退化とよばれる)にあるとき、またはその状態に近いとき、計算誤差が原因となって処理が破綻する危険性がある。そのために、そのまま計算機にインプリメントしても数値的に不安定であり、実用的なシステムを構築できない。以下で、母点逐次添加型の3次元ポロノイ図構成算法に対し、その数値的安定化を試みる。

3.1 3次元ポロノイ図の位相的性質

まず、3次元ポロノイ図の位相構造を見てみよう。図1に示すように、1個のポロノイ点に対し、通常は、4本のポロノイ辺、6個のポロノイ面、4個のポロノイ領域が隣接している。また、1本のポロノイ辺には、3個のポロノイ面、3個のポロノイ領域が隣接している。

退化した場合(5個以上の母点が同一の球面上にあり、その球の内部には他の母点がない場合)には、ポ

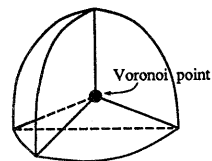


図1 ポロノイ点のまわりの位相構造
Fig. 1 Topological structure around a Voronoi point.

ロノイ点のまわりのボロノイ領域が5個以上になるなど、上の通常の場合からはずれず。しかし、数値誤差のある世界では、そもそも退化しているか否かを厳密に判定することはできないから、退化のない場合だけを扱えば十分である。

ボロノイ図は次のような位相的性質を持つ。

- P1** それぞれの母点は空でないボロノイ領域を持つ。
P2 ボロノイ領域は単連結である。
P3 二つのボロノイ領域の境界は、高々1個のボロノイ面を共有する。

P1は定義から明らかであり、P2, P3はボロノイ領域が凸であることから導かれる。

3.2 数値的に安定化のための基本方針

3次元ボロノイ図の構成過程において処理が破綻するのは、計算誤差のために判定を誤り、3次元ボロノイ図が有すべき位相構造が破壊される場合である。幾何図形処理においては、常にその図形の持つ位相的性質に矛盾が生じないように処理を進めなければならない。そこで本算法では、3次元ボロノイ図構成手続きの主要部分を位相構造の更新と捉え、その更新が矛盾なく行われることを最優先する。もし、ボロノイ図が満たすべき位相的性質に反するような数値判定がなされた場合には、それは採用しない。そして、位相的に矛盾の起きない範囲でなお残る任意性を解消するためにのみ、数値計算結果を用いる。

4. データ構造とアルゴリズム

前節で述べた基本方針を実現するための、データ構造とアルゴリズムについて説明する。

4.1 データ構造

位相的性質を確認しながら処理を進めることができ、母点添加時の更新作業も効率良く行えるように、3次元ボロノイ図の各図形要素に持たせるデータを次のように決めた。データ項目の右肩に*がついているものは、3次元ボロノイ図の計量的性質に関するデータであり、その他のものはすべて位相構造に関するデータである。

ボロノイ点

- そのボロノイ点の x, y, z 座標値*
- そのボロノイ点に隣接する4個のボロノイ辺

ボロノイ辺

- そのボロノイ辺の端点となる2個のボロノイ点
- そのボロノイ辺に隣接する3個のボロノイ面

ボロノイ面

- そのボロノイ面に属するボロノイ辺のうち1個
 - そのボロノイ面の両側に隣接するボロノイ領域
- #### ボロノイ領域
- そのボロノイ領域を勢力圏とする母点の x, y, z 座標値*
 - そのボロノイ領域に属するボロノイ面のうちの1個

各図形要素に上記のような位相データを持たせることで、3次元ボロノイ図内を自由に探索することが可能となる。例えば、一つのボロノイ領域(多面体)の境界となっているボロノイ面の集合のように、自然な順序づけができない図形要素を検索する場合にも、各図形要素に持たせてある位相データに基づき、次のような操作を行う。

まず、そのボロノイ領域に属する一つのボロノイ面をキューに入れる。次に、キューからボロノイ面を一つ取り出し、そのボロノイ面に隣接する面のうちで、対象としているボロノイ領域の境界となっており、かつ未検索のものをキューに入れる。これをキューが空になるまで繰り返す。

以上の操作により、一つのボロノイ領域(多面体)の境界となっているすべてのボロノイ面が、キューから順次取り出されることになる。このようにすることで、一つのボロノイ面から始めて、そのまわりに徐々に検索範囲を拡大していき、最終的には、多面体全体を包み込むようにして検索が行われる。

4.2 位相構造に着目したボロノイ図更新手続き

母点逐次添加法においては、新しい母点が添加されたときにそれまでにできている図形をどう更新するかが最も重要な作業である。この作業を位相構造の更新として捉えてみる。いま、母点 p_1, p_2, \dots, p_{i-1} に対するボロノイ図 V_{i-1} が得られているとし、新たな母点 p_i の添加によってボロノイ図 V_{i-1} を V_i に更新する手続きを位相構造のみに着目して捉えたのが以下のアルゴリズムAである。その様子を図2に順に示す。

アルゴリズムA[ボロノイ図の位相構造の更新手続き]

1. ボロノイ図 V_{i-1} に属するあるボロノイ点集合 T を選ぶ(図2(a);ただしここでは、図が複雑でわかりにくくなるのを避けるために、 T の要素が1点の場合について示している)。
2. T に属するボロノイ点と、 T に属さないボロノイ点を結ぶ辺上に、新しいボロノイ点を一つずつ作る(図2(b))。

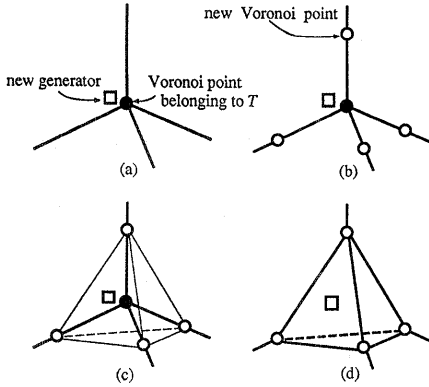


図 2 母点添加時の位相的变化

Fig. 2 Change of the topological structure on addition of a new generator.

3. 2で作られたボロノイ点のうち同じボロノイ領域に属する点を順に結んで、新しいボロノイ面を作る (図 2(c)).
4. T に属するボロノイ点と、それらを結ぶ辺、およびそれらで張られるボロノイ面で作られる V_{i-1} の部分構造を削除する (図 2(d)).

ここで、削除されるボロノイ点集合 T は、次に示す C1~C5 の位相的性質を満たさなければならない。

削除すべきボロノイ点集合 T の位相的性質

- C1 T は空集合でない。
- C2 T に属する点とそれらを結ぶ辺によってできる V_{i-1} の部分グラフは連結している。
- C3 ひとつのボロノイ領域に属するすべてのボロノイ点 p_i が、 T の要素であることはない。
- C4 T の要素のうちボロノイ領域 $V(p_i)$ に属するものの集合を $T(p_i)$ と表すことにすると、任意の $i=1, 2, \dots, l-1$ に対して、 $T(p_i)$ の要素であるボロノイ点とそれらを結ぶ辺によってできる V_{i-1} の部分グラフも連結している。
- C5 T の要素でないボロノイ点のうち、ボロノイ領域 $V(p_i)$ に属するものの集合を $\bar{T}(p_i)$ と表すことにすると、任意の $i=1, 2, \dots, l-1$ に対して、 $\bar{T}(p_i)$ の要素であるボロノイ点とそれらを結ぶ辺によってできる V_{i-1} の部分グラフも連結している。

T がこの五つの位相的性質を満たさないときは、以下に示すような事態が生じ、3次元ボロノイ図の位相的性質と矛盾することになる。

- C1 を満たさない場合は、新しい母点のボロノイ

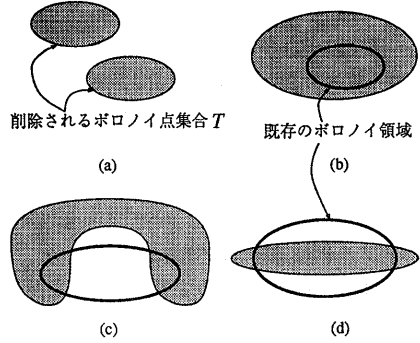


図 3 位相構造の破壊を引き起こすような T の選び方
Fig. 3 Choice of T that causes topological inconsistency.

領域が空集合となって、P1 に反する。

- C2 を満たさない場合は、新しい母点の添加によってできるボロノイ領域が2個以上の多面体に分裂し、P2 に反する (図 3(a)).
- C3 を満たさない場合は、既存のボロノイ領域が、新しい母点の添加によって完全に消滅することになり、P1 に反する (図 3(b)).
- C4 を満たさない場合は、既存のボロノイ領域において、新しい母点の添加によって切り取られる部分とその領域内で2カ所以上に分かれていることになり、P3 に反する (図 3(c)).
- C5 を満たさない場合は、既存のボロノイ領域が新しい母点の添加によって2個以上に分断され、P2 に反する (図 3(d)).

一方、位相的性質 C1~C5 を満たす範囲内で T を選び、前述のアルゴリズム A を実行する限りにおいては、P1~P3 が満たされる。したがって、位相構造が破壊されて処理が破綻することはない。いかに低精度の計算を行おうとも P1~P3 を満たすという意味で位相的に矛盾のない結果が出力される。

このような位相的無矛盾性のチェックを加えても、従来の逐次添加型算法と比べて平均的計算量が増すことはないと思される。なぜなら、1個の母点を添加したとき変更されるボロノイ図の部分構造の大きさは、平均的には母点数に依存しない一定規模であると期待できるからである。

4.3 数値判定

性質 C1~C5 を満たすボロノイ点集合 T は多数ある。そのうち、ボロノイ図の更新手続きのために採用すべきものがどれであるかを決定しなければならない。 T に属するべきボロノイ点は、新しい母点のボロ

ノイ領域に含まれるものである。したがって、その判定は（数値誤差がなければ）次の方法で行うことができる。

判定したいボロノイ点に隣接する4個のボロノイ領域の母点を通る球を考え、新しく添加された母点はその球の内部にあれば、そのボロノイ点は削除され、外部にあれば、削除されないと判定することができる。隣接する4個の母点を p_i, p_j, p_k, p_l とし、新しく添加された母点を p とする。右手系の正規直交座標系に関する p の座標を (x, y, z) 、 p_i の座標を (x_i, y_i, z_i) 、などと表すことにする。実数 $H(p_i, p_j, p_k, p_l, p)$ を

$$H(p_i, p_j, p_k, p_l, p) = \begin{pmatrix} 1 & x_i & y_i & z_i & (x_i^2 + y_i^2 + z_i^2)/2 \\ 1 & x_j & y_j & z_j & (x_j^2 + y_j^2 + z_j^2)/2 \\ 1 & x_k & y_k & z_k & (x_k^2 + y_k^2 + z_k^2)/2 \\ 1 & x_l & y_l & z_l & (x_l^2 + y_l^2 + z_l^2)/2 \\ 1 & x & y & z & (x^2 + y^2 + z^2)/2 \end{pmatrix} \cdot \begin{pmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_k & y_k & z_k \\ 1 & x_l & y_l & z_l \end{pmatrix}$$

と定義する。4点 p_i, p_j, p_k, p_l が同一平面上にないとき、 $H(p_i, p_j, p_k, p_l, p) = 0$ を満たす p は4点を通る球面上にあり、

$H < 0$ ならば球の内部、

$H > 0$ ならば球の外部にある、

と判定する。

したがって、“原理的”には H の符号に基づいて、各ボロノイ点が T に属するか否かを判定できる。しかし、実際には数値誤差があるため、 H の符号に頼りきることはできない。そこで、C1~C5 という位相的性質が満たされることを最優先し、それに反しない範囲で一番もってもらいたい T を選ぶための優先度の低い情報として H の符号を利用する、というのが本稿で提案する数値的安定化の骨子である。

p_i, p_j, p_k, p_l のボロノイ領域の境界が共有するボロノイ点を v とし、新しく添加する母点を p とするとき、 $H(p_i, p_j, p_k, p_l, p)$ を、以下では $H(v)$ と略記することにする。

4.4 位相的チェックのアルゴリズム

4.2 節で述べたアルゴリズム A を利用する場合に最も難しい部分が、位相的性質 C1~C5 を満たすような削除すべきボロノイ点集合 T を選び出すところであろう。ここでは、その部分のアルゴリズムについて説明する。

アルゴリズムの記述を簡潔にするために一つの準備をする。1個のボロノイ領域の境界上のボロノイ点とボロノイ辺は、前者を頂点とし後者を枝とするグラフ

G と見なすことができる。このグラフ G の頂点の集合を N とする。 $v_1, v_2 \in N$ 、 $N' \subseteq N - \{v_1, v_2\}$ に対して、グラフ G において v_1 と v_2 をつなぐすべての道が、 N' の要素を少なくとも1個含むとき、 v_1 と v_2 は N' によって隔てられているということにする。

次に述べるアルゴリズムでは、残すことに決めたボロノイ点集合を T_0 、 T に隣接する頂点のうち T にも T_0 にも属さないボロノイ点集合を T_N で表すことにする。 T_N を表現するにはキューを用いる。したがって、 T_N から要素を取り出すときには、最も古いものから順に取り出す。はじめは、 T, T_0, T_N はすべて空集合である。

アルゴリズムB [削除すべきボロノイ点集合 T を選ぶアルゴリズム]

1. 新しく添加された母点に最も近い母点のボロノイ領域がなす多面体を見つける。
2. その多面体に属する頂点のうち、 H の値が最小となるものを、 T の最初の要素とし、その頂点に隣接するすべての頂点を T_N に入れる。
3. T_N から最も古い要素 v を取り出す。
4. v に関して、以下の5~7のステップがすでに1回以上実行されており、最後に実行されたときの集合 T と T_0 が、現在のものと同じであったなら、 v を T_0 に入れて、8へ行く。
5. $H(v)$ を計算する。
6. $H(v) \geq 0$ の場合：すぐに“残留”と決定して、 v を T_0 に入れ、8へ行く。
7. $H(v) < 0$ の場合：
 - (a) v が載っている4個の多面体に関して、一つでも以下の i) または ii) または iii) の条件に当てはまれば、 v を T_0 に入れる（すなわち、数値判定結果を覆し、“残留”と決定する）。
 - i) v 以外の頂点がすべて T に属す。
 - ii) T に属す頂点と v とが、 T_0 によって隔てられている。
 - iii) v を T に入れると、 T_0 に属す2個の頂点で、 T によって隔てられてしまうものがある。
 - (b) 上記 i), ii), iii) の条件には当てはまらないが、 v が載っている四つの多面体に関して、一つでも以下の iv) または v) の条件に当てはまれば、 v を T_N に戻す（決定を“保留”する）。

- iv) T に属す頂点と v とが, $T_0 \cup T_N$ によって隔てられている。
 v) v を T に入れると, $T_0 \cup T_N$ に属す 2 個の頂点のうち T によって隔てられてしまうものがある。

(c) 上記 i), ii), iii), iv), v) のどれにも当てはまらない場合のみ, v を T に入れ (“削除” と決定する), v に隣接し $T \cup T_0 \cup T_N$ に属さない頂点をすべて T_N に入れる。

8. T_N に属するものがなくなるまで, 3~7 を繰り返す。

このアルゴリズムでは, ステップ 2 で位相的性質 C1 を保証し, それ以降は, T に属す頂点に隣接するものしか見ていないので, 位相的性質 C2 を満たすことは明らかである。また, ステップ 7 の i) で C3 をチェックし, ii), iv) で C4, を, iii), v) で C5 のチェックを行っている。

ここで, iv) および v) においては, 未決定の頂点も併せた位相的チェックを行っており, 後の判定次第では位相的性質を満たさなくなる恐れがある場合には, 決定を保留し, 確定できるところから決めていくことにしている。この処理は, 2次元ボロノイ図の構成アルゴリズムを 3次元へ拡張するにあたり, 新しく必要になった処理であるので, 以下に説明を加える。

2次元ボロノイ図の構成過程においては, 一つのボロノイ領域 (多角形) の境界上のボロノイ点のうちで T に属するものと, それらを結ぶボロノイ辺で作られる部分グラフが, 非連結になるような数値判定がなされたときには (図 4(a)), 即座に数値判定結果を覆す。なぜなら, 2次元の場合には, T に属するボロノイ点とそれらを結ぶボロノイ辺で作られる部分グラフには閉路があつてはならないため^{1),2)}, 図 4(b) のようになることはない。よって, それ以降の判定がどのようになされようとも, そのボロノイ領域上の部分グラフは非連結のままであり (図 4(c)), T が有すべき位相的性質を満たさなくなるからである。

一方, 3次元の場合には, 新しい母点の添加により, 既存のボロノイ面が消滅してしまうことがあるため, T に属するボロノイ点とそれらを結ぶボロノイ辺で作られる部分グラフには, 閉路が含まれていてもよい。このため, 一つのボロノイ領域 (多面体) の境界上のボロノイ点のうちで T に属するものと, それらを結ぶボロノイ辺で作られる部分グラフが, 非連結になるような数値判定がなされたときでも (図 5

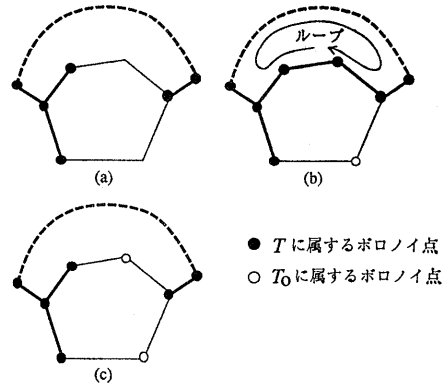


図 4 2次元ボロノイ図の構成過程における T の位相的性質のチェック

Fig. 4 Topological check on T in the case of two-dimensional Voronoi diagram.

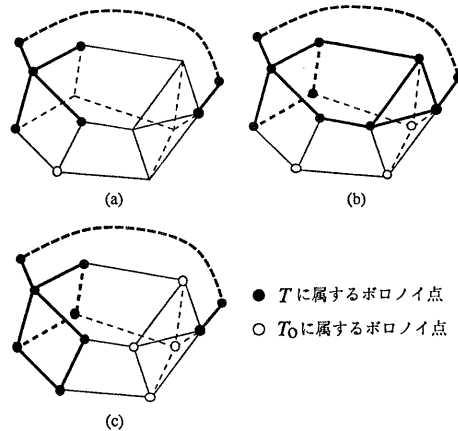


図 5 3次元ボロノイ図の構成過程における T の位相的性質のチェック

Fig. 5 Topological check on T in the case of three-dimensional Voronoi diagram.

(a)), その後の判定結果次第では, 図 5(b) のように, ボロノイ領域上の部分グラフが連結となる場合もあれば, 図 5(c) のように, 非連結のままである場合もある。したがって, その時点で決定してしまうと, あとでその決定を覆さなければならなくなる可能性がある。そこで, 本アルゴリズムにおいては, 決定を一時 “保留” して, 他のボロノイ点の決定を待った後に, 決定することになっている。この工夫により, 位相的性質 C1~C5 を満たす T をいつでも効率よく得ることができる。

なお, このアルゴリズムでは退化に対する例外処理が全く含まれていないことにも注意されたい。すなわ

ち、 $H(v) \geq 0$ の場合 (ステップ6) と、 $H(v) < 0$ の場合 (ステップ7) に分けているだけで、 $H(v) = 0$ の場合を特別扱っていない。したがって、アルゴリズムの構造は従来のもより著しく簡単になっている。

5. 計算機実験

5.1 インプリメント

C言語を用いて計算機へのインプリメントを行った。プログラムの規模は、C言語のソースプログラムで5,000行程度であり、そのうち、入出力処理・八分木構造への母点格納処理などを除いた、母点添加時のポロノイ図更新手続きに関する部分は、2,500行程度である。実験に使用した計算機は、Sun 4/370であり、UNIXオペレーティングシステムのもとXウィンドウ上で稼働する。数値計算は単精度浮動小数点表現で行った。以下に計算例をいくつか示す。

5.2 母点が一樣に分布している場合

3次元空間内の単位立方体領域 $[0, 1) \times [0, 1) \times [0, 1)$ の内部に乱数を用いて一樣に発生させた点を母点として、ポロノイ図を構成した。母点数50の場合の出力結果を図6(a)に示す。中心付近のポロノイ領域の一つを同図(b)に示す。(a)のポロノイ図において、ポロノイ面を共有する母点同士を線分でつないで得られる構造を同図(c)に示す。この構造は、ドロネー図と呼ばれる母点集合の凸包内部の四面体分割を与える。

これらの図は、右眼で左側を、左眼で右側を見たとき、3次元構造が知覚できる(両眼立体視)ような図の対として表示してある(以下の図8、図9も同様の表示方法を採用してある)。

5.3 母点が一樣に分布している場合 (母点数大)

5.2節と同じ条件のもとで、母点数を1,000個まで増やした。母点逐次添加法の処理に要する時間を途中経過も含めて測定した。結果を図7に示す。この図から、母点数 n に対して、 $O(n)$ の処理時間で実現されていることが確認でき、位相的無矛盾性のチェックを入れても、従来の算法に比べ著しく計算効率が悪くならないことがわかる。

また、母点数 n の増加に伴うポロノイ図全体のポロノイ点数・ポロノイ辺数を測定した。測定結果を表1に示す。3次元ポロノイ図は、前述のように退化のない場合には一つのポロノイ点に4本のポロノイ辺が隣接しているという位相構造を有しているので、ポロノイ点数 v とポロノイ辺数 e には次の関係がある。

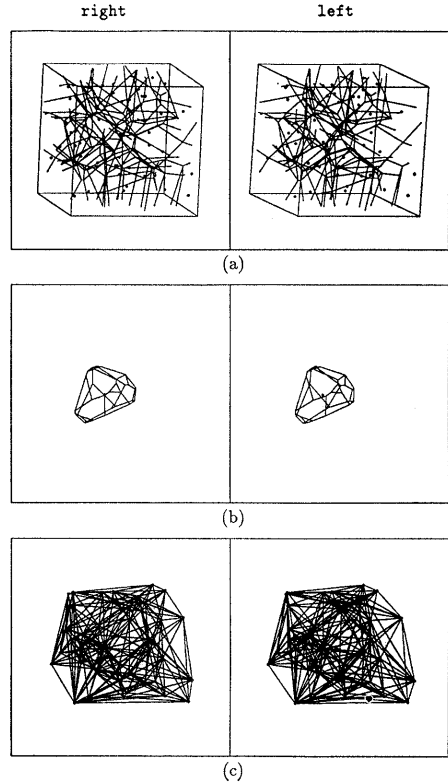


図6 母点が一樣に分布している場合 (母点数50)
(a) ポロノイ図; (b) 中心付近の一つのポロノイ領域; (c) ドロネー図

Fig. 6 Output of the program for fifty generators placed at random in the unit cube.
(a) output; (b) an example of a Voronoi region near the central portion; (c) Delaunay diagram.

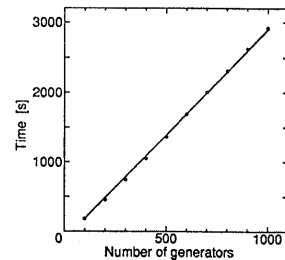


図7 計算効率の測定

Fig. 7 Average time complexity.

$$e = 2v$$

表1を見ると、常に上式を満たしていることがわかる。また、新しく添加した母点のポロノイ領域の平均の面数および1面当たりの角数を求めてみた結果が表2である。これをみると、母点数 n が増えても、一つ

表 1 図形要素数の変化
Table 1 Change of the size of the Voronoi diagram.

母点数	総ポロノイ点数	総ポロノイ辺数
200	1,221	2,442
400	2,520	5,040
600	3,834	7,668
800	5,165	10,330
1,000	6,477	12,954

表 2 ポロノイ領域の変化
Table 2 Change of the average size of a Voronoi region.

母点数	面数/領域	角数/面
200	13.9	5.1
400	14.4	5.2
600	14.7	5.2
800	14.8	5.2
1,000	14.9	5.2

表 3 ポロノイ領域のサイズ
Table 3 Size of Voronoi regions.

面数/領域	該当する領域数
6	1
7	5
8	10
9	26
10	51
11	66
12	104
13	122
14	125
15	113
16	87
17	86
18	66
19	43
20	32
21	27
22	13
23	9
24	5
25	8
26	0
27	1

のポロノイ領域当たりの図形要素数はほぼ一定であることがわかる。

さらに、母点数 1,000 の場合についてすべてのポロノイ領域に対して 1 多面体当たりの面数を調査した結果を表 3 に示す。表 3 のデータから得られる 1 ポロノイ領域当たりの平均面数は 14.8 で、逐次添加によって新しく作られたポロノイ領域の面数の平均値 14.9 (表 2) とよく一致している。

なお、3次元ポロノイ図の面の総数は、最悪の場合には、 $O(n^2)$ 個となる(したがって、1多面体当たり $O(n)$ 個)ことが知られているが、ランダムに分布する母点に対して $n \leq 1000$ の範囲では、そのような傾向は全く見られないことが、この表からわかる。

5.4 母点が同一球面上に分布している場合

著しく退化している状態を作るため、 $[0, 1) \times [0, 1) \times [0, 1)$ の立方体領域に内接する球面上に乱数を用いて一様に発生させた点を母点とした。この場合においても処理が破綻することなく最後まで実行された。従来の数値判定のみに基づいた算法では、このような入力に対して、大抵の場合処理が破綻する。母点数を 100 としたときの出力結果を図 8(a) に示す。中心付

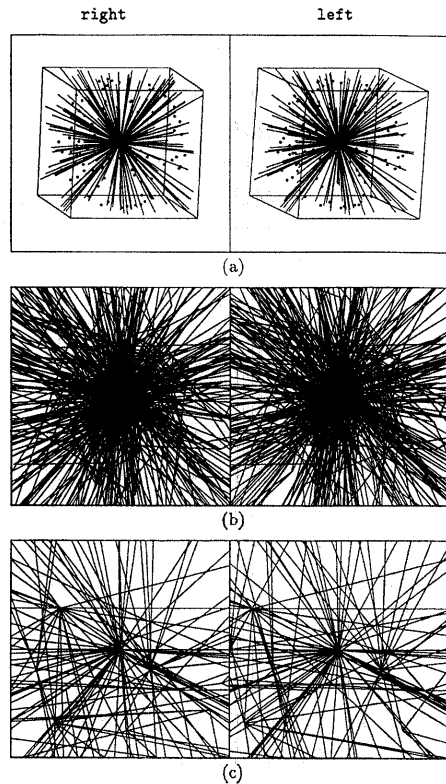


図 8 母点が同一球面上に分布している場合 (母点数 100)

(a) ポロノイ図; (b) 中心付近の拡大図 (1×10^x); (c) 中心付近の拡大図 (3×10^x)

Fig. 8 Output of the program for one hundred generators placed at random on the sphere inscribed in the unit cube.

(a) output; (b) central portion magnified by 10^x ; (c) central portion magnified by 3×10^x .

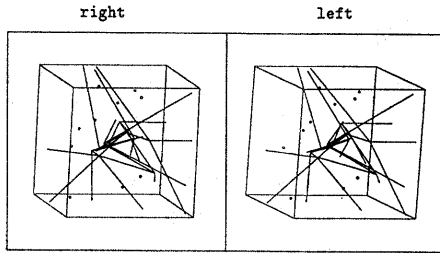


図9 数値判定 H を乱数に置き換えた場合 (母点数 10)
Fig. 9 Output of the program when computed values of H are replaced by random values.

近にボロノイ点が集まり、著しく退化していることがわかる。この中心付近を非常に拡大してみたのが、同図 (b) (c) である。倍率は (b) が 1×10^6 , (c) が 3×10^7 である。これを見ると、中心付近は真のボロノイ図とはかなりかけ離れていることがわかる。しかし、位相的には、 $P_1 \sim P_3$ を満たすという意味で矛盾のないことが理論的に保証されている。この場合のように、間違いのない数値判定をすることがほぼ不可能な状況においても、処理が最後まで実行されることが確認でき、本算法の数値的安定性が実証された。

5.5 数値判定を乱数に置き換えた場合

計算誤差の究極の場合として、プログラムの中で判定のために用いている $H(v)$ の値を乱数に置き換えて実行してみた。この場合においても処理は最後まで実行され、結果を出力した。母点数 10 の場合の出力結果を図 9 に示す。結果はもちろん真のボロノイ図とはかけ離れてはいるが、位相的には矛盾のない処理を行っているので、途中で破綻することはない。

6. おわりに

母点逐次添加法による 3次元ボロノイ図構成算法を改良し、数値的安定化を図った。ここでの基本方針は、母点添加時のボロノイ図更新作業の位相的側面に着目し、数値計算結果より位相構造の無矛盾性をより優先させることによって、計算誤差による処理の破綻を防いだものである。

退化のない入力データに対して無限の精度で数値計算を行った場合には、本アルゴリズムの位相的チェックは実質的には何の働きもしない (すなわち、数値計算のみに頼っても同じ判定結果が得られる)。したがって、従来のアルゴリズムと同じ結果を出す。この意味で本アルゴリズムの出力は、計算精度を上げると真の解に収束するといえる。ただし、退化した場合には、本アルゴリズムの出力は真の解に長さ 0 のボロノ

イ辺や面積 0 のボロノイ面を適当に追加した構造に収束する。これは、位相構造を優先することによって、退化に対する例外処理が不要になり、退化のない構造のみを扱っているという本アルゴリズムの特徴の結果である。

さらに、本算法を計算機にインプリメントして実験を行った結果、その数値的安定性を確認することができた。よって、実用的なプログラムとして十分耐えることができそうである。

今後は、数値的に安定であるという特長を活かし、より一般的なボロノイ図構成への拡張や、構造解析・コンピュータビジョンなどの分野への応用について考えていきたい。

参考文献

- 1) 伊理, 杉原: 計算誤差を考慮した幾何的アルゴリズム, 情報処理学会アルゴリズム研究会報告, AL-1-1 (1988).
- 2) 杉原: 幾何アルゴリズムの数値的破綻とその対策, 応用数理, Vol. 1, No. 4, pp. 280-299 (1991).
- 3) Milenkovic, V.: Verifiable Implementations of Geometric Algorithms Using Finite Precision Arithmetic, *Artif. Intell.*, Vol. 37, pp. 377-401 (1988).
- 4) Hoffman, C., Hopcroft, J. and Karasick, M.: Towards Implementing Robust Geometric Computations, *Proceedings of the 4th ACM Annual Conference on Computational Geometry*, pp. 106-117 (1988).
- 5) 大石, 杉原: 数値的に安定な分割統治型ボロノイ図構成算法, 情報処理学会論文誌, Vol. 32, No. 6, pp. 709-720 (1991).
- 6) 伊理 (監), 腰塚 (編), ほか: 計算幾何学と地図情報処理, bit 別冊, 共立出版, 東京 (1986).
- 7) Preparata, F. P. and Shamos, M. I.: *Computational Geometry—An Introduction*, Springer-Verlag, New York (1985).
- 8) Ohya, T., Iri, M. and Murota, K.: Improvements of the Incremental Method for the Voronoi Diagram with Computational Comparison of Various Algorithms, *Journal of the Operations Research Society of Japan*, Vol. 27, No. 4, pp. 306-336 (1984).
- 9) Aurenhammer, F.: Voronoi Diagram—A Survey of a Fundamental Geometric Data Structure, *ACM Computing Surveys*, Vol. 23, No. 3, pp. 345-401 (1991).
- 10) Okabe, A., Boots, B. and Sugihara, K.: *Spatial Tessellations—Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, Chichester (1992).

- 11) Dey, T., Bajaj, C. and Sugihara, K.: On Good Triangulations in Three Dimensions, *International Journal of Computational Geometry and Applications*, Vol. 2, No. 1, pp. 75-95 (1992).
- 12) Schumaker, L.: Triangulations in CAGD, *IEEE Computer Graphics and Applications*, Vol. 13, No. 1, pp. 47-52 (1993).
- 13) 名取, 野寺(編): 小特集「グリッド・ジェネレーションとその応用」, 情報処理, Vol. 30, No. 7, pp. 766-795 (1989).

(平成4年6月1日受付)

(平成5年10月14日採録)



稲垣 宏 (正会員)

昭和35年生。昭和58年名古屋大学工学部電気学科卒業。昭和60年同大学大学院工学研究科情報工学専攻修士課程修了。同年トヨタ自動車(株)入社。平成元年豊田工業高等専門学校情報工学科助手。現在に至る。計算幾何学の研究に従事。電子情報通信学会, 人工知能学会各会員。



杉原 厚吉 (正会員)

昭和23年生。昭和46年東京大学工学部計数工学科卒業。昭和48年同大学院工学系研究科計数工学専門課程修士課程修了。電子技術総合研究所, 名古屋大学工学部を経て, 現在東京大学工学部計数工学科教授。工学博士。コンピュータビジョン, 計算幾何学などの研究・教育に従事。著書 *Machine Interpretation of Line Drawings* (MIT Press, 1986), *Spatial Tessellations* (John Wiley, 1992, 共著) など。1980, 1988, 1991年度本学会論文賞受賞。電子情報通信学会, 日本オペレーションズ・リサーチ学会, 日本応用数理学会等の会員。



杉江 昇 (正会員)

昭和32年名古屋大学工学部電気卒業。同年通商産業省電子技術総合研究所入所。昭和37~39年カナダ・マギル大学客員研究員。昭和45年バイオニクス研究室長。昭和53年視覚情報研究室長。昭和54年名古屋大学大学院工学研究科情報工学専攻教授。昭和60年同大学工学部電気工学第二学科教授。平成2年同大学工学部情報工学科教授。現在に至る。バイオニクス, 医用工学, コンピュータビジョン, 自然言語処理などの研究・教育に従事。工学博士。電気学会, 電子情報通信学会, 計測自動制御学会, ロボット学会, エム・イー学会, テレビジョン学会, バイオメカニズム学会, 日本神経回路学会, IEEE 等各会員。