

曲線上に配置された母点に対する平均線形時間ボロノイ図構成法

山田和公 杉原厚吉

東京大学大学院工学系研究科計数工学専攻

概要

曲線上に密に配置された母点に対するボロノイ図は、一般図形を生成元とするボロノイ図の近似として用いられる。従来の逐次添加法によってこのような母点配置に対するボロノイ図を構成すると、実際に母点数の線形時間で構成することができない、誤判定を生じやすい、という困難が生じる。本報告では、この困難の原因を取り除くことによって、実際に母点数の線形時間でボロノイ図を構成する逐次添加法、および誤判定をより少なくする数値計算法を提案する。また、計算実験によってこの算法の有効性を確認する。

$O(n)$ Average-Time Construction of the Voronoi Diagram for Generators on Smooth Curves

Kazuhiro Yamada and Kokichi Sugihara

Department of Mathematical Engineering and Information Physics,
Graduate School of Engineering, University of Tokyo

The Voronoi diagram for generators on smooth curves is used for the approximation of the generalized Voronoi diagram for figures. Generators on smooth curves give difficulty in the construction of Voronoi diagram by the incremental method, because both construction time and numerical errors become larger. This paper proposes an $O(n)$ average-time method for constructing the Voronoi diagram for generators densely distributed along curves and a method for computing numerical values more precisely. The performance of the method is also evaluated experimentally.

1 はじめに

曲線上に配置された母点に対するボロノイ図の構成は、一般図形を生成元とするボロノイ図を近似的に求める場合などに必要になる [5].

数値誤差を含む環境で安定して動作するボロノイ図の構成法には逐次添加法 [6], 分割統治法 [2] などがある. これらの算法の最悪計算時間は母点数 n に対して, 前者は $O(n^2)$, 後者は $O(n \log n)$ であるが, パケットを利用することによって, 母点が一様分布に従っている場合には, 両者とも平均 $O(n)$ の時間で構成することができる. この算法は一様分布以外の多くの場合においても実際に $O(n)$ に近い計算時間で実行できることが経験的にわかっている.

このような算法のうち逐次添加法を曲線上に配置された母点に対して適用した場合, $O(n)$ を越える計算時間を要することがある, 計算誤差などによって結果が大きく誤ることがある, などの難点が見つかっている [7]. 本稿では逐次添加法に対する, 数値的により安定にするための数値計算の改良と, 計算時間を短縮するための改良を提案する.

算法を設計するための前提として, 与えられた母点は曲線上でどの点と隣り合っているのか, という情報が分かっており, 母点が隣り合っている順に整列されて与えられていることを仮定する.

2 数値的安定化

逐次添加法では, 数値判定として母点 $p_i = (x_i, y_i)$ の座標を用いて以下の二つの式の正負の判定を行なう.

$$\det H(p_1, p_2, p_3, p_4) = \begin{vmatrix} 1 & x_1 & y_1 & (x_1^2 + y_1^2)/2 \\ 1 & x_2 & y_2 & (x_2^2 + y_2^2)/2 \\ 1 & x_3 & y_3 & (x_3^2 + y_3^2)/2 \\ 1 & x_4 & y_4 & (x_4^2 + y_4^2)/2 \end{vmatrix} \quad (1)$$

$$D(p_1, p_2, p_3) = |p_1 - p_3| - |p_2 - p_3| \quad (2)$$

前者は母点 p_1, p_2, p_3 で構成されるボロノイ点 p_4 が, 新たに添加された母点 p_4 の領域の内部に

あるか外部にあるかの判定を行なうために用いる (内外判定).

後者は母点 p_3 が母点 p_1 と p_2 のいずれに近いかを判定するために用いられる (最近点判定).

逐次添加法 [6] では, 遠方に 3 点を配置する事によって凸包の内部に母点を追加するだけでボロノイ図を構成する事が出来る. この 3 点を配置することによって, 母点がすべて同一直線上に配置される状態が避けられる. このとき退化して正確には値が 0 になる判定式の符号は, 正負いずれに判定されても誤りは生じないという性質を持つ. そのため, 計算の精度を向上させることによって誤りの少ないボロノイ図を得ることができる.

2.1 内外判定

表記を簡単にするため, 記号を以下のように定義する.

$$\Delta x_{ij} = x_i - x_j$$

$$\Delta y_{ij} = y_i - y_j$$

$$\Delta r_{ij} = (\Delta x_{ij}^2 + \Delta y_{ij}^2) / 2$$

内外判定式は以下のように変形することができる.

$$\det H = \begin{vmatrix} \Delta x_{21} & \Delta y_{21} & \Delta r_{21} \\ \Delta x_{31} & \Delta y_{31} & \Delta r_{31} \\ \Delta x_{41} & \Delta y_{41} & \Delta r_{41} \end{vmatrix} \quad (3)$$

$$= \begin{vmatrix} \Delta y_{21} & \Delta r_{21} \\ \Delta y_{31} & \Delta r_{31} \end{vmatrix} \times \Delta x_{41}$$

$$- \begin{vmatrix} \Delta x_{21} & \Delta r_{21} \\ \Delta x_{31} & \Delta r_{31} \end{vmatrix} \times \Delta y_{41}$$

$$+ \begin{vmatrix} \Delta x_{21} & \Delta y_{21} \\ \Delta x_{31} & \Delta y_{31} \end{vmatrix} \times \Delta r_{41} \quad (4)$$

計算式 (4) が従来から提案されている計算式である [6]. p_1 をピボットとしてその他の母点に対応する行から引いているが, この計算式には, どの母点をピボットに選ぶか, という任意性があ

る。ここでピボットをボロノイ点に対応している p_1, p_2, p_3 から選ぶことにすると、各行列式はボロノイ点が構成された時点で予め計算しておくことによって計算量を軽減する事ができる。このとき接近している2点間の距離をより正確に評価するために、最も距離の長い辺を対辺に持つ点をピボットにする。

ここでピボットの選択に対して、 p_4 が考慮されていないため、 p_4 がピボット以外の点 p_k に非常に近い場合、 Δx_{4k} などの情報は桁落ちなどによって失われ、 p_4 と p_k の垂直二等分線とは離れたボロノイ点に対しても判定を誤ることがある。また、4点が2点ずつ接近して配置されている場合、どの要素をピボットに選んでも、いずれか一方の精密な情報は失われる。このために同一円周上にはない母点に対する判定を誤る事がある。

これを回避するために以下のように変形する。

$$\begin{aligned} & \det H \\ = & \begin{vmatrix} \Delta y_{21} & \Delta r_{21} \\ \Delta y_{31} & \Delta r_{31} \end{vmatrix} \times \Delta x_{4k} \\ - & \begin{vmatrix} \Delta x_{21} & \Delta r_{21} \\ \Delta x_{31} & \Delta r_{31} \end{vmatrix} \times \Delta y_{4k} \\ + & \begin{vmatrix} \Delta x_{21} & \Delta y_{21} \\ \Delta x_{31} & \Delta y_{31} \end{vmatrix} \times \Delta r_{4k} \\ + & \begin{vmatrix} \Delta x_{21} & \Delta y_{21} \\ \Delta x_{31} & \Delta y_{31} \end{vmatrix} \\ & \times (\Delta x_{4k} \Delta x_{k1} + \Delta y_{4k} \Delta y_{k1}) \quad (5) \end{aligned}$$

(5) 式の計算法でも (4) 式同様に、行列式の部分は予め計算しておくことが出来る。行列式のピボットと、 p_4 に対する最近点とで2つの自由度を持つため、精密な情報を保った、より安定な計算法であると考えられる。

2.2 最近点判定

$D(p_1, p_2, p_3)$ の判定のために $|p_1 - p_3|^2$, $|p_2 - p_3|^2$ を求めてその差を計算した場合、 p_1, p_2 が非常に接近していて p_3 が遠く離れているとき、丸めにより2つの距離が一致して、 p_1, p_2 の等距

離線上にない母点 p_3 に対して $D = 0$ となることがある。このときに最近点ではない点を最近点として出力することがある。この母点が新しい母点とは領域が隣接しない母点であるときに誤りを生じる。

これを回避するためには、距離の比較ではなく、距離の2乗差を計算して評価する。

$$\begin{aligned} & D(p_1, p_2, p_3) \\ = & (\Delta x_{13}^2 - \Delta x_{23}^2) + (\Delta y_{13}^2 - \Delta y_{23}^2) \\ = & (\Delta x_{13} + \Delta x_{23}) \times \Delta x_{12} \\ & + (\Delta y_{13} + \Delta y_{23}) \times \Delta y_{12} \quad (6) \end{aligned}$$

2.3 距離計算

各母点が、曲線の関数と曲線上の位置を示すパラメタで与えられている状況を考える。この場合曲線上で隣り合う母点の差ベクトルや距離などの計算に、直接座標の差を計算する方法の他に、曲線の関数を用いて差ベクトルの関数を陽に求めて、2点のパラメタから計算する方法がある。

座標値は曲線上の点の真の座標値に対して丸め誤差を含んだ情報であるため、曲線の関数と、パラメタの差を用いることによってより正確に差ベクトルが計算できる。特に、厳密には異なる点であるが、丸めによって同じ座標値が与えられた点を区別することが出来る、という効果がある。

2.4 改善例

ここで、今回提案した計算法によって誤りが改善された例を示す。図1は、従来の計算法で初めて誤りが生じた時点で停止したものである(18点/12795点)。また図2は改良した算法を使って同じところまで計算した結果である。図1の中央下の辺りで生じている誤りが、図2では解消されている。最後まで算法を実行すると、従来の算法では図3、新しい算法では図4のような結果が得られる(12795点)。この図では、異なる図形間の境界のみを表示している。

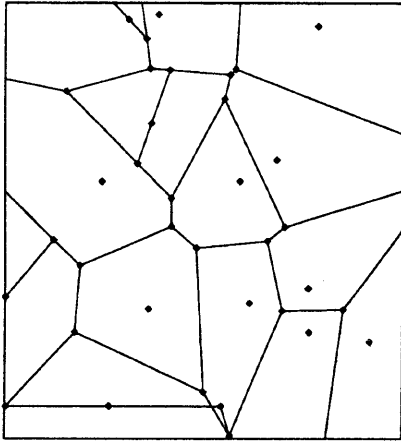


図 1. 従来の算法による誤り (18 点 / 12975 点)

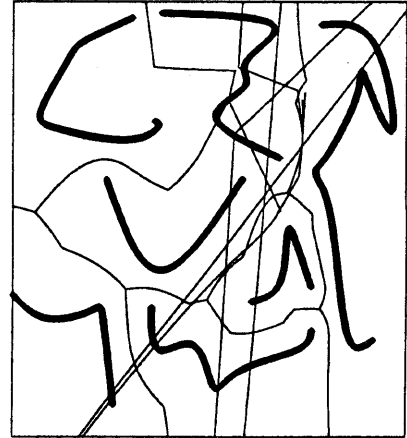


図 3. 従来の算法による結果 (12795 点)

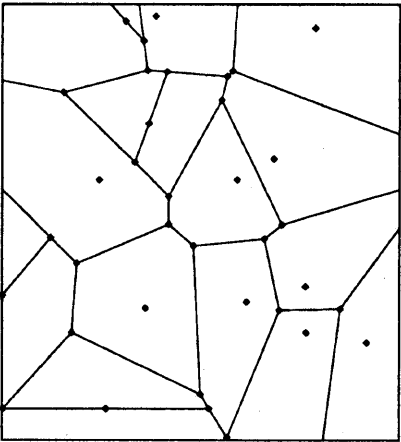


図 2. 新しい算法による改善 (18 点 / 12975 点)

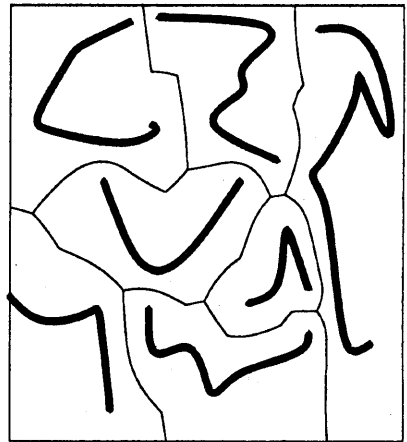


図 4. 新しい算法による結果 (12795 点)

3 高速化

逐次添加法を用いた時の計算時間のオーダは与える図形に大きく依存する。最悪の場合は $O(n^2)$ の時間を必要とする。この原因は、開曲線の端点など、 $O(n)$ 角形の領域を持つ母点が存在することである。従来の算法はすべての母点の領域が(約)6角形であることを前提にしているため、高速化のためには $O(n)$ 角形の領域の影響を少なくする工夫が必要である。

3.1 母点添加

各母点の添加は次のような手順で行なわれる。

1. 新しい母点にもっとも近い母点を探索する。
2. 新しい母点のボロノイ領域によって削除されるべき位相構造を求める。
3. 新しい母点の領域を構成する。

ここで削除される位相構造を求めるためには、新しい母点のボロノイ領域に含まれるボロノイ点を求めればよい。従来の算法 [6] では、ある母点

の領域から、新しい母点によって切り取られるボロノイ点を求める際に、以下のように領域の周囲のすべてのボロノイ点に対して内外判定を行なって印をつけている。

絶対値優先法 (abs と略す)

- 領域のまわりのすべてのボロノイ点に対して数値判定を行ない、絶対値の大きいボロノイ点の符号を優先して位相的に無矛盾になるように内外を決定する。

$O(n)$ 角形の領域をもつ母点が存在する場合、この領域のボロノイ点だけで $O(n^2)$ 回の内外判定を行なっている。しかし、判定するボロノイ点の多くは新しい領域には含まれないボロノイ点であり、冗長な判定となっているため改善の余地がある。

実際に、母点の添加順序を工夫することによって新しい母点の領域の角数を平均定数オーダーにすることができる。このとき、実際に削除されるボロノイ点の数も定数オーダーである。よって、新しい領域に含まれないボロノイ点に対する判定を減らすために、以下のように変更する。

隣接点優先法 1 (next1 と略す)

- 新しい母点に含まれるボロノイ点から、順に判定していき、新しい母点の外にある、と判定されたところで停止する。判定を行なわなかったボロノイ点はすべて新しい母点の外であるという印をつける。

位相的に矛盾が生じないための歯止めとして、領域の周りのすべてのボロノイ点に対して印をつける。この算法では $O(n)$ 角形の領域が存在する場合、すべてのボロノイ点に対して印をつけているため、数値判定の回数は減少するものの、印をつける手続きは減少していないために冗長な計算時間を要する。ボロノイ点に印をつける手続きも減少させて、さらに高速化するために以下のように変更する。

隣接点優先法 2 (next2 と略す)

- 新しい母点に含まれるボロノイ点から、順に判定していき、新しい母点のボロノイ領域の外にある、と判定されたら停止する。さらに、判定した領域の母点に判定済の印をつける。以後、位相的歯止めとして、ボロノイ点を構成する母点の中に判定済の母点があれば、そのボロノイ点は自動的に新しい母点のボロノイ領域の外にある、という印をつける。

以上によって新しい母点の領域が定数オーダーの角数であるとき、定数オーダーの手間で新しい母点が添加できることが期待される。

ただし、最初の領域のボロノイ点に印をつける作業において、新しい母点の領域の内側にある点と外側にある点とを一つづつ定める必要がある。これは、すべて内側(外側)と判定されることを避けるためである。この際に領域のすべてのボロノイ点に対して判定を行なうため、新しい母点が定数オーダーより大きい複雑さを持つ領域内に存在した場合、この母点を定数時間で添加することは困難である。

3.2 添加順序

母点の添加に対しては、各母点の領域の(添加時点での)複雑さが定数オーダーになるように、できるだけ一様に分布するように添加順序を決めることが望まれる。また、添加した母点の属する領域を探索する際に、探索領域数を少なくするための探索開始点の設定も必要となる。

従来の算法では、バケットと四分木を用いた以下のような方式を用いている。

バケット法 (bucket1,2)

- 領域を $\sqrt{n} \times \sqrt{n}$ 個の小領域に分割して、母点を小領域ごとに分ける。
- さらに隣り合う 2×2 の小領域をまとめていき、領域内の母点を1つ割り当てる。これを再帰的に繰り返して四分木を構成する。

- 添加順序は四分木を用いて各小領域に定められた順序に従って添加順序を定める。
- 母点の属する領域の探索は、そのバケットを含む小領域に属する点を探索開始点とする。

小領域に複数の母点が含まれている場合に、同じ小領域に属する点を続けて添加していく方法 (bucket1 と名付ける) と、同じ小領域に属する点は1つだけ添加してすべての母点を添加するまで反復する方法 (bucket2 と名付ける) とが考えられる。

一様分布では各小領域には定数個の母点しか存在しないため、このような算法によって1つの母点を定数時間で添加することが可能になる。

しかし、母点が曲線上に分布する場合は小領域に含まれる母点数は $O(1)$ ではなく $O(\sqrt{n})$ であるため、最近点の探索に $O(\sqrt{n})$ の時間を要する。

この時間を減少させるために、バケットを用いずに曲線上での隣接情報を利用して添加順序と探索開始点とを定める次の算法を考える。

log step 法

1. 曲線の端点をバケット法で定まる順序で添加する。
2. 端점에隣合う母点を添加する。探索開始点は隣り合う母点の端点とする。
3. k を $\lfloor \log n \rfloor$ とする。
4. k 番目の母点から $2k$ 個おきに母点を添加する。領域探索の開始点はすでに添加された点で曲線上で隣り合う点 (k 離れた母点) とする。
5. $k = 1$ ならばすべての母点を添加したので終了。それ以外は $k = k/2$ として4に戻る。

この算法では、およそ $n/2^m$ 個の母点を添加したとき、各曲線上では 2^m 個おきに母点が添加されているように母点の添加順序を定めている。

曲線の端点を始めに添加するのは、端点が $O(n)$ 角形の領域を持つ可能性が高いので、領域が定数オーダーの複雑さであるうちに添加してしまうためである。次いで端점에隣り合う点を添加するのは、母点が $O(n)$ 角形の領域内に存在するという状況をなるべく避けるためである。

十分多くの母点を添加した後では、曲線上で隣接する点との距離がその他に対して短くなっているため、新しい母点によって取り除かれるボロノイ点は両隣りの領域の点のみであることが期待される¹。これを認めれば、後半の $n/2$ 個の母点の添加は、 $1/2$ 添加時点での全体の構造の複雑さと同程度の手間、すなわち $O(n/2)$ の手間で追加できる。よって、全体が $O(n)$ の手間で追加できることが期待される。

3.3 実験結果

今回の実験は、入力図形として従来の算法で最も計算時間を要したデータを扱う (図5左)。このデータでは中心点のボロノイ領域が $O(n)$ 角形になるために、従来の算法では $O(n^2)$ の計算時間を要している。ここで計算時間はボロノイ図構成に要した時間のみを扱い、入出力などは時間を含めない。結果が正しいと見なせるか否かは不明確であるが、少なくともこれまでのような見た目ではっきり認識できる誤りは生じていない。

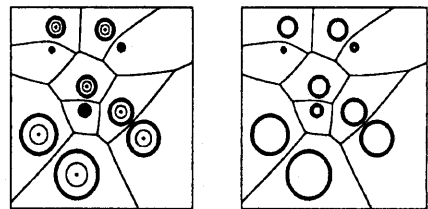


図5. 計算時間を計測したデータ

まず、図6に母点添加算法の変化による計算時間の変化をしめす。添加順序は log step 法を用いて定めている。計算時間は入力図形に依存しており、この例は絶対値優先法で最も時間を要するものである。

¹ただし、円周上の点などには成り立たない

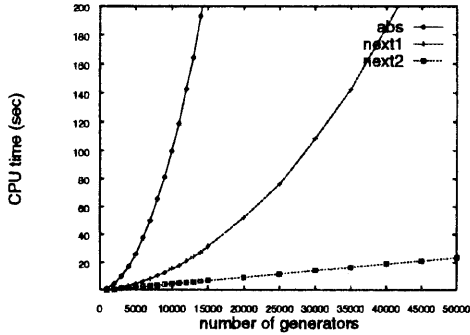


図 6. 母点添加算法の違いによる計算時間の変化

絶対値優先法, 隣接点優先法 1, 隣接点優先法 2 の計算時間のオーダーはそれぞれ $O(n^{1.9})$, $O(n^{1.6})$, $O(n^{1.0})$ である。 $O(n^2)$ に近いオーダーであった計算時間が $O(n)$ に減少している。

次いで図 7 に添加順序の変化による計算時間の変化をしめす。母点添加算法は隣接点優先法 2 を用いている。

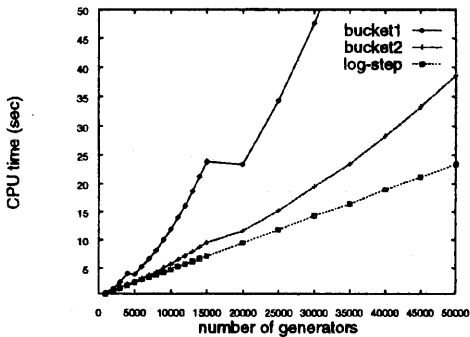


図 7. 添加順序の違いによる計算時間の変化

計算時間のオーダーは bucket1, bucket2, log step 法の順にそれぞれ $O(n^{1.3})$, $O(n^{1.1})$, $O(n^{1.0})$ である。 bucket1 でグラフが平坦になっている箇所は, おそらく小領域の数が増加したことに対応するものであると考えられる。添加順序の変更は母点添加算法の変更ほど大きな変化は見られないが, $O(n)$ を僅かに上回っていた計算時間が $O(n)$ で抑えられるようになった。

さらに, 判定などの実行回数のオーダーを表 1, 2 に示す。

表 1. 添加算法による計算時間の変化 1

| | abs | next1 | next2 |
|-------|-----|-------|-------|
| 時間 | 1.9 | 1.6 | 1.0 |
| 枝数 | 1.0 | 1.0 | 1.0 |
| 内外判定 | 1.9 | 1.0 | 1.0 |
| 印 | 1.9 | 1.9 | 1.0 |
| 最近点判定 | 1.0 | 1.0 | 1.0 |
| 探索母点 | 1.0 | 1.0 | 1.0 |

表 2. 添加順序による計算時間の変化 1

| | bucket1 | bucket2 | log |
|-------|---------|---------|-----|
| 時間 | 1.3 | 1.1 | 1.0 |
| 枝数 | 1.2 | 1.1 | 1.0 |
| 内外判定 | 1.2 | 1.1 | 1.0 |
| 印 | 1.2 | 1.1 | 1.0 |
| 最近点判定 | 1.5 | 1.0 | 1.0 |
| 探索母点 | 1.5 | 1.1 | 1.0 |

枝数は新しく生成した枝の総数で, 内外判定を行なった領域の数に等しい。内外判定は内外判定を行なった回数, 印はポロノイ点に印をつけた回数である。最近点判定は最近点判定を行なった回数, 探索母点は, 最近点を求めるために探索した母点の数である。

添加算法において, 隣接点優先法 1 の場合は内外判定の数は減少したものの, 印を付けるポロノイ点の数は絶対値優先法の場合と変わらないためより多くの母点数で実験すると, $O(n^2)$ 時間を要するものと思われる。

添加順序において, bucket1, bucket2 で大きな差が生じているのは両者における最近点探索の探索開始点が異なるためであると思われる。この結果は曲線上での隣接条件を全く用いない場合, $O(n^{1.5})$ の計算時間を必要とすることを示唆している。

最後に従来方式で一番早かったデータ (図 5 左) に対する結果を表 3, 4 に示す。従来方式で一番早かった図形が, 今回提案した方式では一番遅いという逆転現象が起きている。従来方式では入力図形によって計算時間のオーダーが大きく

変化しているが、今回提案した方式では入力図形に依らず計算時間が線形オーダーであることが分かる。

表 3. 添加算法による計算時間の変化 2

| | abs | next1 | next2 |
|-------|-----|-------|-------|
| 時間 | 1.2 | 1.1 | 1.0 |
| 枝数 | 1.0 | 1.0 | 1.0 |
| 内外判定 | 1.2 | 1.0 | 1.0 |
| 印 | 1.5 | 1.6 | 1.0 |
| 最近点判定 | 1.0 | 1.0 | 1.0 |
| 探索母点 | 1.0 | 1.0 | 1.0 |

表 4. 添加順序による計算時間の変化 2

| | bucket1 | bucket2 | log |
|-------|---------|---------|-----|
| 時間 | 1.2 | 1.1 | 1.0 |
| 枝数 | 1.2 | 1.1 | 1.0 |
| 内外判定 | 1.2 | 1.1 | 1.0 |
| 印 | 1.2 | 1.1 | 1.0 |
| 最近点判定 | 1.2 | 1.0 | 1.0 |
| 探索母点 | 1.3 | 1.1 | 1.0 |

4 まとめ

本稿で提案した方法によって、ボロノイ図の構成における数値誤差による誤判定の悪影響を軽減することに成功した。また、曲線上での隣接情報を用いることによって、曲線上の母点に対するボロノイ図の構成を実際に線形時間で行うことに成功した。

この計算法を用いた場合も、より母点を密に配置したとき、誤判定によって誤った結果を得ているものと思われるが、従来のような明らかな誤りが生じることは確認されていない。

今後の課題としては、直線上の4点のように、真に退化しており、かつその判定のために結果が乱れてしまうような場合に対する、有効な数値判定法の確立が挙げられる。

参考文献

- [1] Guibas, L., and Stolfi, J.: Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, Vol. 4 (1985), pp. 74-123.
- [2] 大石泰章, 杉原厚吉: 数値的に安定な分割統治型 Voronoi 図構成算法, 情報処理学会論文誌, Vol. 32 (1991), pp. 709-720.
- [3] Ohya, T., Iri, M., and Murota, K.: Improvement of the incremental method for the Voronoi diagram with computational comparison of various algorithms. *Journal of the Operations Research Society of Japan*, Vol. 27 (1984), pp. 306-336.
- [4] 杉原厚吉: 計算幾何工学, アドバンストエレクトロニクス シリーズ II-2, 培風館, 1994.
- [5] Sugihara, K.: Approximation of generalized Voronoi diagrams by ordinary Voronoi diagrams. *Computer Vision, Graphics, and Image Processing - Graphical Models and Image Processing*, Vol. 55 (1993), pp. 522-531.
- [6] Sugihara, K., and Iri, M.: A robust topology-oriented incremental algorithm for Voronoi diagrams. *International Journal of Computational Geometry and Applications*, Vol. 4 (1994), pp. 179-228.
- [7] 山田和公, 杉原厚吉: 一般図形ボロノイ図の近似構成法とその改良, 情報処理学会研究報告(アルゴリズム研究会), 96 - AL - 51 - 2, No. 57, Vol. 96 (1996), pp. 9-16