アルゴリズム 11－2
（1989. 10. 24）

組合せ構造を優先した線分ボロノイ図の構成

今井敏行　杉原厚吉
東京大学　工学部　計数工学科

線分を生成元とするボロノイ図を構成する算法を示す．図形のもつ組合せ構造を優先的に扱っている．組合せ構造を優先しない一般の算法で正常な出力が得られるような入力および計算精度のもとでは、この算法でも正常な出力が得られ，一般の算法では暴走するような悪い計算精度のもとでもこの算法では暴走しない．この意味でこの算法は計算誤差に対して強い。さらに誤差を前提とする算法であるので入力が退化している状況でも算法は正常に働き，退化状態のための例外処理が不要である．

A COMBINATORIAL-STRUCTURE ORIENTED ALGORITHM FOR VORONOI DIAGRAMS OF LINE SEGMENTS

Toshiyuki Imai　Kokichi Sugihara
Faculty of Engineering, University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan

An algorithm to coustruct Voronoi diagrams whose generators are line segments is shown. In this algorithm combinatorial structures of the diagrams are considered as higher priority information than numerical values. If the precision in computation is enough, this algorithm constructs the correct Voronoi diagrams. Moreover, even when precision in computation is so poor that conventional algorithms do not work well, this algorithm ends up with some output that is topologically consistent. In this sense, this algorithm is robust against numerical error. This algorithm is designed on the assumption that numerical error exists; hence it has not any exceptional rules for degeneracy but still works for degenerate cases.

1．まえがき

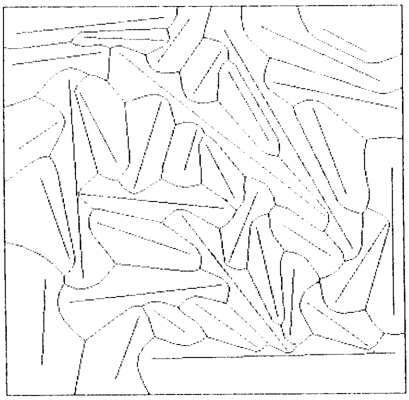
平面上にn本の直線分s_i(i=1,...,n) が与えられたとき最も近い直線分がs_iである平面上の点の集合は一つの領域をなす。これをR_iとかくとR_i(i=1,...,n)は平面の分割を与える。この分割によって得られる図を直線分s_i(i=1,...,n)を生成元とする線分ボロノイ図という（図１）。生成元が点であるときには同様にしてよく知られた点ボロノイ図が得られる。

ボロノイ図の構成算法としては、逐次添加法や再帰二分法が知られている。逐次添加法は生成元を一つずつ添加しながらボロノイ図を変更していく方法で、再帰二分法は生成元を二等分してボロノイ図を二つつくり合併することを再帰的に行う方法である。従来、これらの方法では、計算誤差や退化状態は生じないものとして論じられることが普通であった。そのため、実際にプログラムにしてみると生成元が規則正しく並んでいるだけで暴走するような非実用的な面があった。しかし最近、図形のもつ組合せ構造を保つことを優先的に考えることによって、計算誤差や退化のある場合でも暴走が回避できるという発想のもと、点ボロノイ図に関しては実用的な算法が実現している（[3]，[4]）。本稿では、同様な考えに基づいた線分ボロノイ図の構成算法とその実現例を示す。

本稿で示す算法が扱う問題は次のとおりである。

平面上の単位正方形$[0,1]^2$内に与えられた互いに共有点をもたないn本の直線分に関する線分ボロノイ図を単位正方形$[0,1]^2$内に構成せよ。

ボロノイ図で、生成元のもつ領域はボロノイ領域、ボロノイ領域の周の辺をボロノイ辺、ボロノイ辺どうしの交点をボロノイ点とよぶ。点ボロノイ図では、ボロノイ辺は直線分になるが、線分ボロノイ図では、ボロノイ辺は直線分と放物線分とが互いに接する曲線となり構造が複雑になる。ここで生成元の直線分s_i(i=1,...,n)を端点p_{2i-1}，p_{2i}と開線分e_iに分け、分割された生成元に関してボロノイ図を構成することにする。ただし、端点p_{2i-1}，p_{2i}と開線分e_iの間のボロノイ辺は、開線分e_iに垂直で、それぞれ端点p_{2i-1}，p_{2i}を通る直線分とする。こうすることによって全てのボロノイ辺が単一の直線分または放物線分になる（[5],[6]）（図１）。分割された生成元に関するボロノイ図から、はじめの線分ボロノイ図を求めることは簡単であるから、ここでは分割された生成元に関してボロノイ図を構成することにする。また、生成元はそれぞれボロノイ領域をただ一つもつので、以後ボロノイ領域と生成元を同じ名で呼ぶ。また、開線分e_iには端点p_{2i-1}からp_{2i}への向きがついているものとし、開線分e_iの左右というときにはこの向きによるものとする。



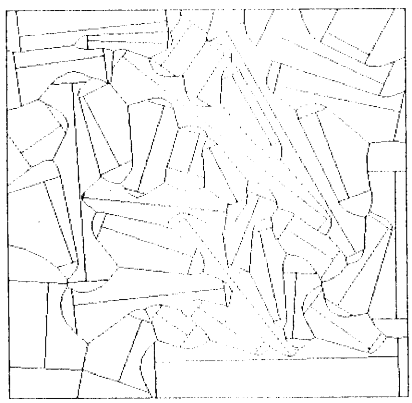


図１．線分ボロノイ図と生成元を分割したボロノイ図

2．従来の逐次添加法と暴走

本稿で示す算法は基本的には逐次添加法である。本節では、線分ボロノイ図の逐次添加法の概略と計算誤差や入力の退化によって生じる逐次添加法での暴走について述べる。線分ボロノイ図の逐次添加法の概略は次のとおりである。

1：正方形$[-1,2]^2$を内部に持つような三角形$a_0a_1a_2$を１つとる。
2：2n+3点$a_0, a_1, a_2, p_{2i-1}, p_{2i}$ $(i=1,...,n)$を生成元とする点ボロノイ図を構成する。
3：各$i=1,...,n$ について、開線分e_iを添加する。

ただし開線分e_iを添加する作業は次のとおりである（図２）。

3-1：ボロノイ領域p_{2i-1}からはじめて、ボロノイ辺を新しく作って既にあるボロノイ辺と交わったら隣のボロノイ領域へ移る、という操作を新しく生じるボロノイ辺が閉じるまで繰り返す。
3-2：新しく生じたボロノイ辺で確定したボロノイ領域の内部にある（古い）ボロノイ辺を消去する。

１で３点 a_0, a_1, a_2をとるのはこうすることによって無限にのびるボロノイ辺を３つのボロノイ領域a_0, a_1, a_2境界３本に限定するための便法で、こうしても単位正方形$[0,1]^2$の内部のボロノイ図には変化がない。また２で先に全端点を添加するのは既存の点ボロノイ図の構成算法を利用できるからである。

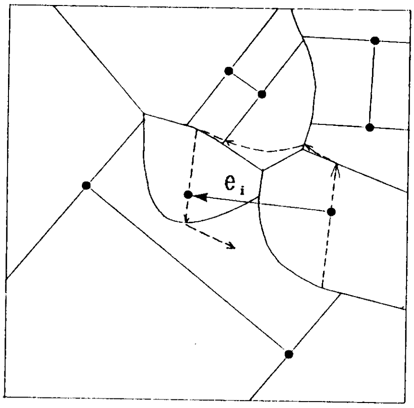


図２．開線分の添加

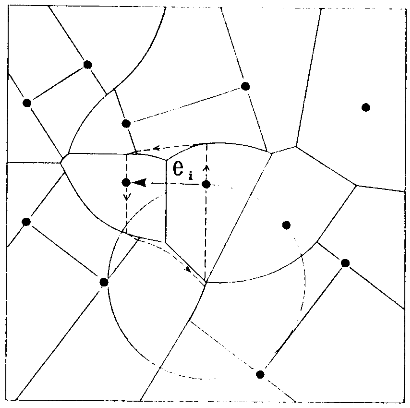


図３．逐次添加法の暴走

次に逐次添加法での暴走について述べる。普通はボロノイ点には３本のボロノイ辺が集まるが、ボロノイ点に４本以上のボロノイ辺が集まっている状態を退化している状態とよぶことにする。退化が生じるのは４つ以上の生成元が同一円周上にのっているときである。ただし、開線分が円周にのるとは円周に接することとする。開線分e_iを添加するときには、既にあるボロノイ辺と新しく生じるボロノイ辺の交差判定が必要であるが、もし退化が生じると同一の交点が別のボロノイ辺のうえにあるとして判定され、ボロノイ辺が閉じないで暴走することがある（図３）。また、誤差のある計算のもとでは、たとえ退化していなくても交差の誤判定から同じように暴走することがある。

３．線分ボロノイ図の組合せ構造

退化や誤差によって暴走するのは計算結果がボロノイ図のもつ構造を破壊するからと考えることができる。したがって、ボロノイ図がもつべき組合せ構造を保つことを優先し、これに反する計算結果を排除することにより、暴走を回避することができる。この方法で点ボロノイ図に関しては退化や計算誤差のある場合でも暴走しない構成算法が実現している（[3], [4]）。そこでは次のような組合せ構造を優先している。

（１）生成元の添加により既にあるボロノイ辺の一部が消去されるが、消去されるボロノイ辺は木構造をなす。
（２）生成元の添加により既にあるボロノイ領域の一部が消去されるが、各ボロノイ領域で消去され

る部分は単連結である。

これらの性質を保つことによって、ボロノイ図が平面グラフであるという性質も保たれる。保つべき組合せ構造として何をとるかは問題ごとに異なり、また同一の問題でも任意性がある。線分ボロノイ図では上の性質（1）は成立するが（2）は成立しない。実際一つのボロノイ領域から何か所もの部分が消去されることがある（図4）。
本稿で採用した性質は次の3つである（図5）。

（3）生成元の開線分の添加により消去されるボロノイ辺は木構造をもつ。
（4）この木構造は両端点のボロノイ領域を結ぶ道を含み、その道はただ一つである。
（5）生成元の開線分の添加により、その両端点のボロノイ領域はそれぞれ1カ所だけが消去される。

性質（3）は各生成元が単連結なボロノイ領域をただ一つもつことからわかる。（4）は開線分が両端点を結ぶことと、道が木の一部であることからわかる。（5）は端点のボロノイ領域が凸であることと、この領域内にできるボロノイ辺が直線分であることからわかる。これらの性質を保つことによって生成元を分割した線分ボロノイ図がもつべき平面グラフであるという組合せ構造が保たれるだけでなく、分割前の生成元の線分ボロノイ図がもつはずの、各生成元がそれぞれただ一つ単連結なボロノイ領域をもつ平面グラフであるという組合せ構造も保たれる。

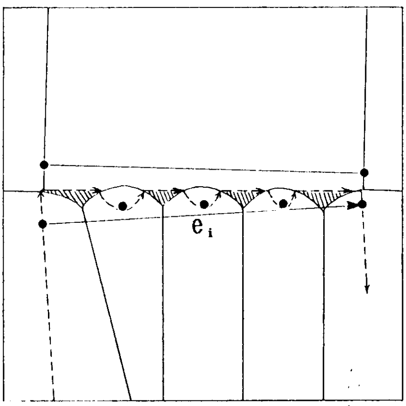


図4．開線分の添加によりボロノイ領域の
複数の部分が消去される例

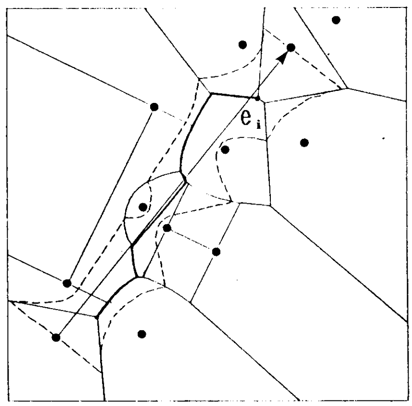


図5．開線分の添加により消去される
木構造と道

4．新しい算法の内容

前節で示した性質（3），（4），（5）を保つようにして生成元を分割した線分ボロノイ図を構成する算法を示す。1，2，3は第2節と同じである。

1：正方形$[-1,2]^2$を内部に持つような三角形$a_0a_1a_2$を1つとる。
2：2n+3点$a_0, a_1, a_2, p_{2i-1}, p_{2i}$ $(i=1,...,n)$を生成元とする点ボロノイ図を構成する。
3：各$i=1,...,n$ について、開線分e_iを添加する。

ただし開線分e_iを添加する作業は次のとおりである。
3-1：ボロノイ領域p_{2i-1}からボロノイ領域p_{2i}にいたる道をみつける。
3-2：3-1で求めた道に消去されるべきボロノイ辺の部分を性質（3），（5）を保ちながら付け加えていき、消去される木構造をを決定する。
3-3：決定した木構造を消去し新しいボロノイ辺を付け加え新しいボロノイ領域を確定する。

この算法で１については第２節と同様である。２は組合せ構造を優先させた点ボロノイ図の構成算法を利用する。３のうち、３-2はボロノイ辺の交差判定を繰り返すだけであり、３-3は単にボロノイ辺の接続関係の更新だけですむ。そこで、３-1について説明する。

３-1では、基本的には、添加開線分e_iに関して右側と左側のボロノイ領域の間のボロノイ辺をたどることによって道を見つけようとする。しかし、この方法では計算誤差による誤判定によって、道が生成元p_{2i}のボロノイ領域に達しない可能性がある。そのために、添加開線分e_iに関して、ボロノイ点v_jを周上にもつボロノイ領域の生成元gの右らしさを表す実数値関数$f(g, v_j)$を適当に定めておく。ただし$f(g, v_j)$は、計算誤差なく値が求められるなら、gが添加開線分e_iに関して右側にあるときに正の値をとり、左側にあるときは負の値をとり、０に近いほど添加開線分e_iに近い、という性質を満たすようにつくる。この$f(g, v_j)$の作り方と問題点については後で述べる。

３-1で道は次のように探す。

３-1-1：ボロノイ領域p_{2i-1}，p_{2i}が、あるボロノイ辺の両側で隣接しているならば、そのボロノイ辺を道として登録して終了、隣接していなければ３-1-2へ行く。

３-1-2：ボロノイ領域p_{2i-1}の周上のボロノイ点v_jから出るボロノイ辺b_kで、その左右のボロノイ領域それぞれg_l，g_mが$f(g_l, v_j) < 0$，$f(g_m, v_j) > 0$をみたすものをとる。ないときはボロノイ領域p_{2i-1}から出る任意のボロノイ辺b_kをとる。
ボロノイ辺b_kを道の候補として登録し、かつb_kをpathに代入する。
pathの両端のうち生成元p_{2i-1}のボロノイ領域に接続していない側のボロノイ点をvtxに代入する（vtxからpathを探せるようにしておく）。
pathにvtxで接続する左右のボロノイ辺をそれぞれlpath，rpathに代入する。
ボロノイ辺b_kの左右の生成元をそれぞれlgen，rgenに代入する。
lgen，rgenとともにvtxを取り囲む生成元をngenに代入する。

３-1-3：ngenが生成元p_{2i}になるまで次の操作Ａを続ける。

操作Ａ：pathが無限にのびるボロノイ辺になるか閉ループをつくるか、ngenが生成元p_{2i}になるまで次の操作A1を続ける。

操作A1：f(ngen, vtx)＞０のときはrgenにngenを、pathにlpathを代入し、f(ngen, vtx)＜０のときはlgenにngenを、pathにrpathを代入する。
pathを道の候補として登録する（pathからvtxを探せるようにしておく）。
vtxをf(ngen, vtx)の絶対値をキーとしてヒープに積む。
pathの両端のうちvtxでないボロノイ点を改めてvtxに代入する。（vtxからpathを探せるようにしておく）。
pathにvtxで接続する左右のボロノイ辺をそれぞれlpath，rpathに代入する。
lgen，rgenとともにvtxを取り囲む生成元をngenに代入する。

操作A1終了

pathが無限に延びるボロノイ辺になるか閉ループをつくれば、そうならなくなるまで次の操作A2を続ける。

操作A2：ヒープから（f(ngen, vtx)の絶対値が最小の）点を取り出しvtxに代入する。
vtxに接続するボロノイ辺で、まだ道の候補として登録されてないものをとりpathに代入する。

操作A2終了

pathの両端のうちvtxでないボロノイ点を改めてvtxに代入する。
pathの左右の生成元をそれぞれlgen，rgenに代入する。
pathにvtxで接続する左右のボロノイ辺をそれぞれlpath，rpathに代入する。
lgen，rgenとともにvtxを取り囲むボロノイ領域の生成元をngenに代入する。

操作Ａ終了

３-1-4：現在のpathを道の一部として登録する。
pathがボロノイ領域p_{2i-1}に接続するまで操作Ｂを続ける。
操作Ｂ：pathから（操作A1で探せるようにした）ボロノイ点を探しvtxに代入する。

vtxから（操作A1で探せるようにした）ボロノイ辺を探しpathに代入する。
現在のpathを道の一部として登録する。
操作B終了

上の算法の考え方は、まず生成元e_iに関して右側と左側のボロノイ領域の間を道の候補としてたどれる限りたどり、もし途中で無限にのびるボロノイ辺にたどり着くか閉ループを作りこれ以上たどれなくなったら、どこかで操作A1の生成元ngenの開線分e_iに関する左右判定（f(ngen, vtx)の符号判定）を誤っているとみなして、f(ngen, vtx)の絶対値が最小の（すなわち符号判定が最もあやしい）ボロノイ点から別のボロノイ辺をたどって行くことを、ボロノイ領域p_{2i}にたどり着くまで続けるというものである。この探索でたどったボロノイ辺はボロノイ領域p_{2i-1}の周上のボロノイ点を根とする木構造になるから、ボロノイ領域p_{2i}にたどり着いたら、木構造をなしているボロノイ辺を根に向かってたどれば必ずボロノイ領域p_{2i-1}, p_{2i}を結ぶ道が得られる（図6）。

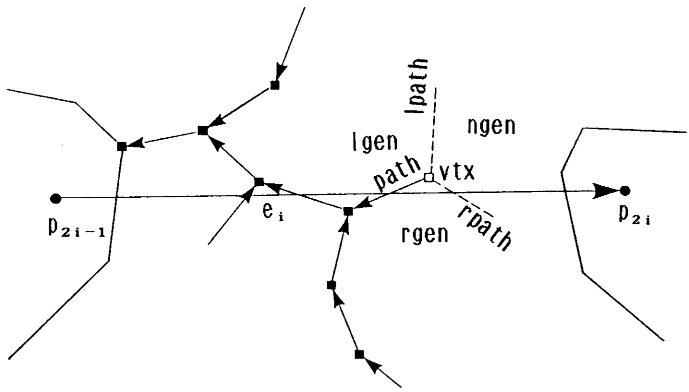


図6．道の探索

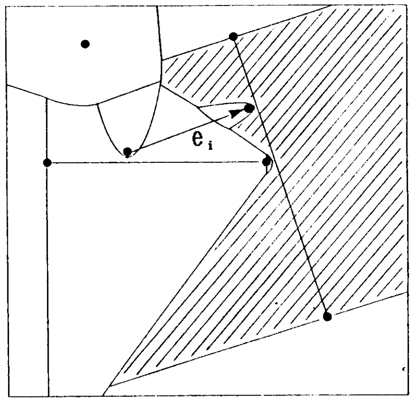


図7．左右判定のむずかしい生成元

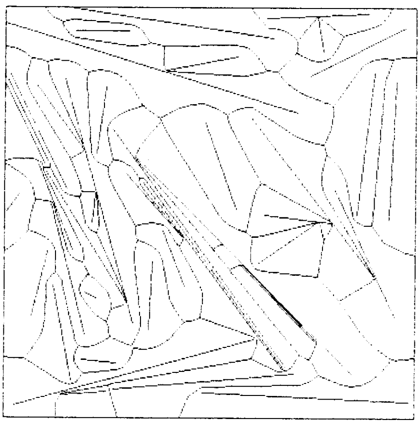
計算の手間は最悪の場合にはほとんどすべてのボロノイ辺をたどるから、ボロノイ辺の総数をmとするとO(mlogm)となる（logmはヒープを使っているためである）。しかしたいていの場合添加開線分e_iの付近のみをたどるので最悪の手間からはずっと少ない手間で済むと思われる。実際の手間は生成元の配置と計算精度により、ヒープの扱い以外は従来の逐次添加法と（計算精度が十分ならば）手間は同程度であるので非実用的には手間は増大しないと思われる。

次に周上にボロノイ点v_jをもつボロノイ領域の生成元gの右らしさを表す実数値関数$f(g, v_j)$について述べる。生成元gが端点のときは$f(g, v_j)$として開線分e_iを延長した直線と点gとの距離に符号をつけたものにすればよい。この時ボロノイ点v_jは関数値に関係しない。生成元gが開線分であるときには簡単ではない。開線分gの両端点が開線分e_iを延長した直線に関して同じ側にないときには左右をすぐには決められないからである。例えば、開線分e_iを延長した直線に関して、開線分gの多くの部分が右側にあるのに左側と判定したい場合がある（図7）。左右判定に生成元gだけでなくボロノイ点v_jも必要な理由はここにある。本稿で示す実現例においてもこの問題は解決されていない。現在のところ考えられる方策を示す。

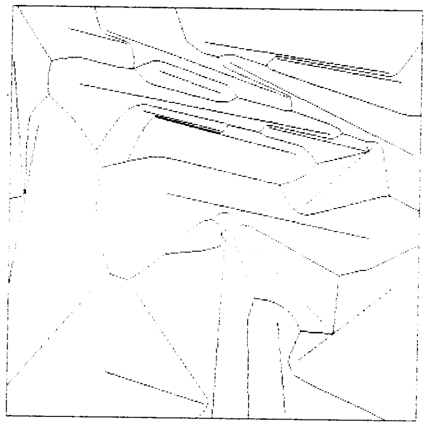
これまでに算法の操作Aで示したように$f(g, v_j)$の絶対値は道の探索を誤ったとき再出発するボロノイ点を決定するために用いるだけであり、道の探索には左右の判定が必要なだけである。計算が正確に行われていれば道は開線分e_iの両端からひいた２本の垂線の間から外へはでない。したがってこの２本の垂線の間の領域はボロノイ領域p_{2i-1}, p_{2i}と道の右側、左側に分けられる。道の上の任意のボロノイ点v_jとv_jを周上にもつボロノイ領域gに関して生成元gが開線分のときv_jからgにおろした垂線の足をwとすると２点w, v_jを結んだ線分はボロノイ辺および開線分e_iを延長した直線と交わらない。点wが開線分e_iの右側にあれば生成元gは開線分e_iの右側にあるといえる。　この方策は算法に使えそうであるがその正当性は現在検討中である。

５．実現例

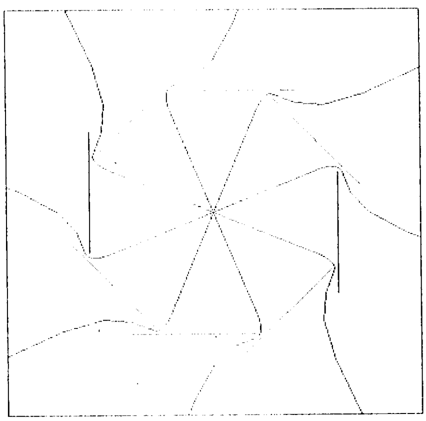
以下に不完全な$f(g,v_j)$用いたものであるが、実現例を図８に示す。(1),(2)は生成元が交差しない限り端点をランダムに選んだ場合で、生成元の数はそれぞれ４０，２０である。(3)は生成元が同一円周に接し退化が起きている場合で、この状態でも暴走せず出力が得られている。(4)は３９本の生成元が規則正しく並んでいる場合で、多くの退化が起こっている。(5)はAの文字の輪郭に適用した場合で、輪郭の各線分は、わずかに端点を離してある。(6)は道を選ぶときにngenを無条件に添加開線分e_iの左にあると判定した場合で、この状態でも暴走せず何らかの出力が得られていることがわかる。



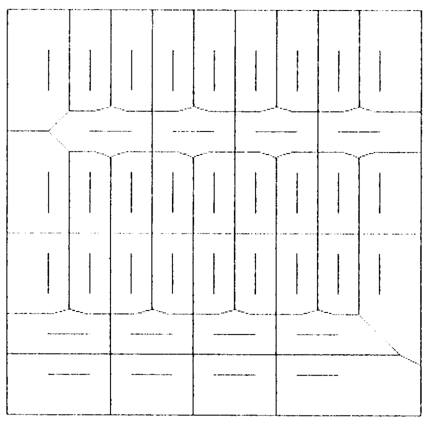
(1)



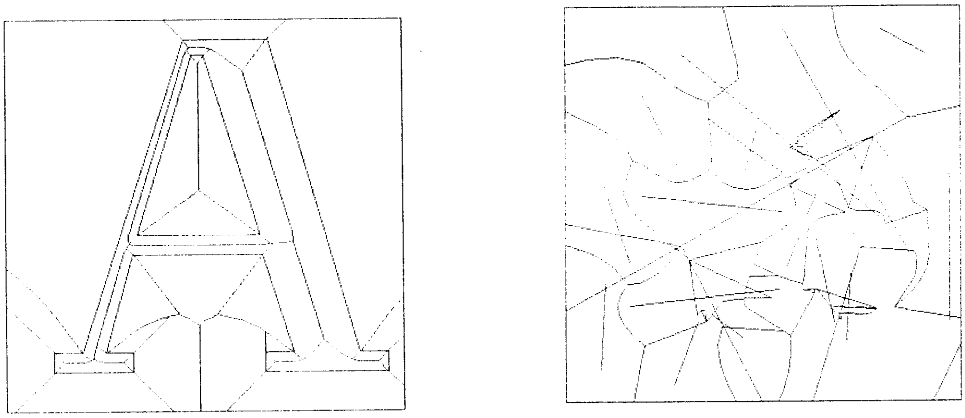
(2)



(3)



(4)



(5)　　　　　　　　(6)

図８．線分ボロノイ図の構成算法の実現例

６．今後の課題

今後の課題としては、まず$f(g, v_j)$を完全なものにすることが第一である。そのほかには、生成元の線分が端点を共有した折れ線にも適用できるよう算法を改良することも今後の課題である。これは生成元のデータ構造を変えることと扱うボロノイ辺の種類を増やすことで対応できそうである。また、あとから生成元を追加できるようにもしたい。これは、あとから端点を添加できるようにすればよいのであるが、これが実現すると点ボロノイ図の生成算法を用いる必要がなくなる。計算の手間の解析もほとんどがこれからの課題であり、暴走しなくても数値計算は正確な方が望ましいので、数値計算方法の改良も今後の課題である。

７．まとめ

組合せ構造を優先した線分ボロノイ図の構成算法とその問題点を示した。未完成部分が残るが暴走状態は回避できている。

８．謝辞

本研究に際し多くのご御助言をいただいた東京大学工学部伊理正夫教授に感謝致します。本研究は文部省科学研究費補助金（一般研究（C），No. 01550279）の援助を受けている。

参考文献

[1]今井敏行: 線分Voronoi図の構成算法における計算誤差対策，東京大学工学系研究科計数工学専攻修士論文，1989.

[2]伊理正夫(監修)，腰塚武志(編): 計算幾何学と地理情報処理，ｂｉｔ別冊，共立出版，東京，1986.

[3]伊理正夫，杉原厚吉: 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研究報告，88-AL-1-1，1988.

[4]K. Sugihara, M. Iri: Two Design Principles of Geometric Algorithms in Finite-Precision Arithmetic , Applied Mathematical letters, Vol. 2, No.2, pp. 203-206, 1989.

[5]小久保岩生: 一般化Voronoi線図の構成算法の研究－－特に線分に対するVoronoi線図について，東京大学工学系研究科計数工学専門課程修士論文，1985.

[6]C. K. Yap: An O(nlogn) algorithm for the Voronoi diagram of a set of simple curve segments, Discrete & Computational Geometry, 2:365-393, Springer-Verlag, New York, 1987.