

## Improvement of Reliability of Geometrical Algorithm Based on Topological Relation Between a Point and a Line

Tadamitsu KANEKO\*, Toshinori TSUJI\*, Toshihiro KAWANO\*\*,  
and Yasuo NAKAYAMA\*\*\*

*\*Faculty of Humanities, Miyazaki Municipal University,  
Miyazaki-shi, 880 Japan*

*\*\*Kitakyusyu National College of Technology,  
Kitakyusyu-shi, 803 Japan*

*\*\*\*Faculty of Commercial Sciences, Kumamoto Gakuen University,  
Kumamoto-shi, 862 Japan*

### Abstract

To keep geometrical calculation from the failure caused by numerical errors, this paper proposes that the topological relation between a point and a line is fundamental, and that the algorithm for it is improved. Experiments for the set-theoretic operations show that the use of the proposed algorithm raised a little reliability but reduced the performance.

## 点と直線の位置関係に基づいた図形処理の信頼性の向上

金子 正光      辻 利則      河野 俊弘      中山 泰雄

### 1. まえがき

計算誤差による図形処理の破綻の問題では、図形の持つ位相構造の破壊をいかに防止するかが重要な課題であり、従来の方法に代わる斬新な方法が提案されている<sup>1)~4)</sup>。しかしその一方で、位相構造の破壊が生じる過程を分析し、従来の方法に対して改善案を示すことも、既存のプログラムの効率的な改良のために有用であると思われる。そこでこの観点から、2次元図形の集合演算における数値的安定性の向上に関して検討した結果を報告する。

一般に、図形処理では交差、包含、遠近などの図形相互間の位置関係が扱われるが、中でも点と直線の位置関係は基本的なものの一つである。本論文では、これに注目して位相構造の破壊の状況を考察し、その判定法の改善策を提案する。更にそれを集合演算に応用する方法を説明し、実験結果を報告する。

### 2. 位相構造の破壊状況と検討方針

#### 2.1 計算誤差による位相構造の破壊

集合演算でどのような図形が生成されるかは交点の有無およびその位置に依存すると考えられる。このことから図1のような位相構造の破壊状況を挙げる事ができる。(a)は交点数の過不足により矛盾が生じる場合である。2つの多角形が交わる時の交点総数は偶数でなければならない。(b),(c)はそれぞれ同一辺上および複数の辺上に検出された交点の並びの順序が本来の順序と入れ代わっている場合である。(d)は交点と既存の辺との位置関係が本来のものと異なる場合である。

これらの内で破壊の程度が最も大きいのは、交点数に矛盾がある(a)であり、交点数に矛盾がない(b),(c)がそれに続くと考えられる。また、交点数および交点の並びの順序が正しく求められたとしても(d)の可能性が考えられる。集合演算においては、交点が発見された後は、既存の図形の情報と交点の情報との検索だけで図形が生成され、交点と既存の辺との位置関係は必要とされない。従って(d)のような場合は自己交差などの矛盾を含んだ図形が生成されるが、それによって処理が中断されることはないであろう。ただし、更に別の図形との集合演算を続行する際にこのような矛盾が悪影響を及ぼす可能性は十分考えられる。

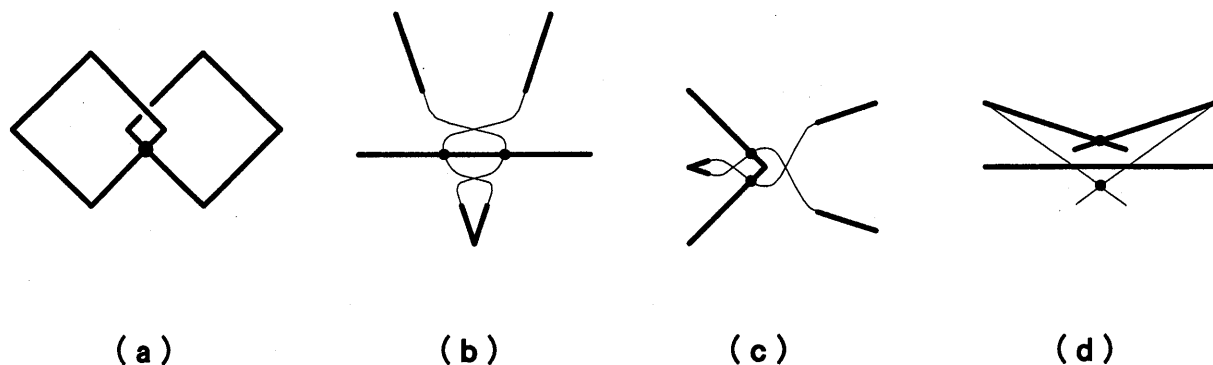


図1 位相構造の破壊の例

## 2.2 検討方針

後述のように、集合演算で必要とされる図形相互間の位置関係は点と直線の位置関係に帰着できる。従ってこの位置関係を全ての位置関係の基本として位置付け、その判定の際に位相構造が破壊される状況を詳しく調べた上で対策を考えることにする。但し今回の報告では、前述した破壊の程度の大きい(a)に関するもの、すなわち点と直線の位置関係と交差判定処理との関りを中心にした調査と対策にとどめる。

## 3. 位置関係の判定法の改良

### 3.1 対象の図形について

集合演算の対象となる図形は多角形とする。それらの入力には慣習に従い頂点の座標値によって行われ、頂点は反時計回りに順序付けられているものとする。また、辺には識別番号を付けるものとする。辺番号は全対象を通して一意とする。

集合演算では、頂点の生成・消滅は起こるが、辺を延長した直線は与えられたもの以外に新しく生じることはない<sup>10)</sup>。従って頂点や交点よりも直線を基本的なものとし、数値計算では直線だけを用いる。

直線の定義データは直線の方程式の係数から成り、入力された多角形の頂点の座標から求められる。すなわち、辺  $i$  を延長した直線を

$$a_i x + b_i y + c_i = 0 \quad (1)$$

で表わすとすれば、各係数は

$$\begin{aligned} a_i &= y_1 - y_2, \\ b_i &= x_2 - x_1, \\ c_i &= x_1 y_2 - x_2 y_1 \end{aligned} \quad (2)$$

で与えられる。ここで  $(x_1, y_1), (x_2, y_2)$  は辺  $i$  の始点および終点の座標である。直線の定義データは処理全体を通して変更されることはない。

### 3.2 点と直線の位置関係の判定

辺番号  $i, j$  を持つ二直線の交点を  $V_{i,j}$  で表わす。また、 $V_{i,j}$  と直線  $k$  との位置関係を関数  $F(i, j, k)$  で表わす。 $F$  は次のように直線の定義データだけで定義される。

$$F(i, j, k) = \frac{\begin{vmatrix} a_i & b_i & c_i \\ a_j & b_j & c_j \\ a_k & b_k & c_k \end{vmatrix}}{\begin{vmatrix} a_i & b_i \\ a_j & b_j \end{vmatrix}} \quad (3)$$

$F$  の値は辺  $k$  の始点  $S_k$  から終点  $E_k$  に向かうベクトルと  $S_k$  から  $V_{i,j}$  に向かうベクトルとの外積に等しい。図2のように、辺  $k$  と同じ方向を向いた時左側を正領域、右側を負領域と呼ぶことにし、 $F$  の値が正なら  $V_{i,j}$  は  $k$  の正領域に、負なら負領域に、0 なら  $k$  上にあるとみなす。位置関係の判定に必要なのは  $F$  の符号だけであるから除算は不要である。辺  $i, j$  が交点  $V_{i,j}$  を形成するので、分母が0になることは通常あり得ない。分子が0になる場合は三直線  $i, j, k$  が一点で交わる、いわゆる退化の発生を意味する。退化発生

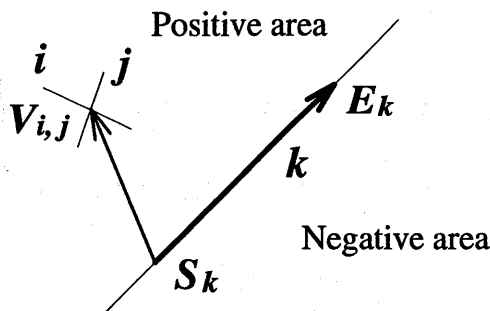


図2 点  $V_{i,j}$  と直線  $k$  の位置関係

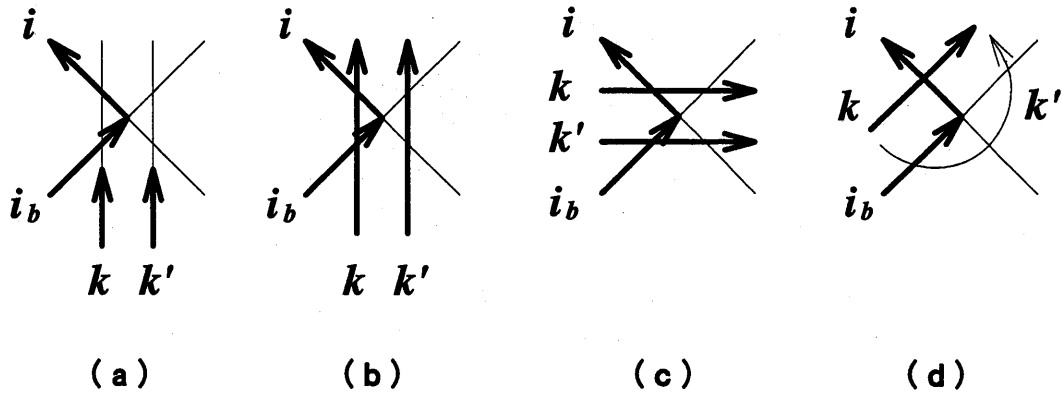


図3 交差判定の配置例

時の対策として記号摂動法<sup>9)</sup>と呼ばれる便利な方法が考案されているが、これは直線の定義データを整数形式に置き換えて数値計算を行ういわゆる整数着法<sup>10)</sup>を採用した場合に限られる。本報告ではまえがきで述べた主旨に従い、直線の定義データを通常の実数形式のまま扱うことにするので、退化発生時には当該辺を実際に摂動させて退化を回避する。

### 3.3 交差判定時の矛盾と対策

点と直線の位置関係に帰着できる交差判定法を取り上げる。すなわち、一方の辺の二端点が他方の辺に対して互いに反対側にあり、他方の辺の端点についても同様のことが成り立つ時、二辺は交差すると判定される。

多角形の連続する二辺  $i_b, i$  と別の多角形の辺  $k$  との交差判定を考える。図3は、辺  $k$  の二端点が直線  $i_b$  と  $i$  の作る4つの領域のいずれに属すかによって分類したものである。ここでは直線  $k$  は  $V_{i_b, i}$  の近傍を通り、辺  $i_b$  の始点と辺  $i$  の終点は遠方にあるものとする。

(a)は辺  $k$  の二端点が同じ領域に属すと判定される場合の一例である。このとき  $V_{i_b, i}$  と直線  $k$  の位置関係は、計算誤差のために二通りの可能性があり、それを図の  $k, k'$  で表わす。いずれにしても交点数は0であり矛盾は生じない。(b), (c)は辺  $k$  の二端点が  $V_{i_b, i}$  に関して点対称の領域に属すと判定される場合の例である。どちらの場合も上記と同様に、計算誤差による位置関係の不定性のために、図に示す  $k, k'$  の二つの可能性が考えられる。従って(b)の場合、交点数は0または2となるが、いずれにしても矛盾は生じない。(c)の場合、辺  $k$  は辺  $i_b$  または  $i$  と交わるが、いずれにしても交点数の矛盾は生じない。(d)は、辺  $k$  の二端点が直線  $i_b$  に関して同じ側にあると判定される場合の一例である。この場合は  $V_{i_b, i}$  と直線  $k$  との位置関係から交差の有無を求めると矛盾を生じる恐れがある。例えば  $V_{i_b, i}$  が直線  $k$  の正領域にあると判定される可能性がある。これを図では  $k'$  で表わす。その場合辺  $k$  は、辺  $i_b, i$  のいずれとも交差しないという矛盾した判定が下される。従って今の配置では数値計算に頼らず、辺  $k$  の二端点と直線  $i_b$  との位置関係から、 $V_{i_b, i}$  は直線  $k$  の負領域にあると判定すべきである。

なお、検討の対象をより多くの辺を含む配置にまで広げる必要があるが、今回は以上のものに限定した。

### 3.4 位置関係の判定法の改良

前節(d)の対処に従った、 $V_{i_b, i}$  と直線  $k$  の位置関係を表わす修正された関数  $F'(i_b, i, k)$  を提案する。仕様は次のとおりである。

1. 次の三条件が全て満たされる時  $V_{i_b, i}$  は直線  $k$  の正領域にあるとみなし、 $F'$  の関数値を1とする。

[条件1]  $F(i, i_b, i_b) > 0$

[条件2]  $F(k_b, k, i_b) > 0, F(k, k_b, i_b) > 0$

[条件3]  $F(k_b, k, i) < 0, F(k, k_b, i) > 0$

ここで辺番号の添え字  $b$  および  $f$  は多角形における当該辺の前および次の辺を意味する。この状況

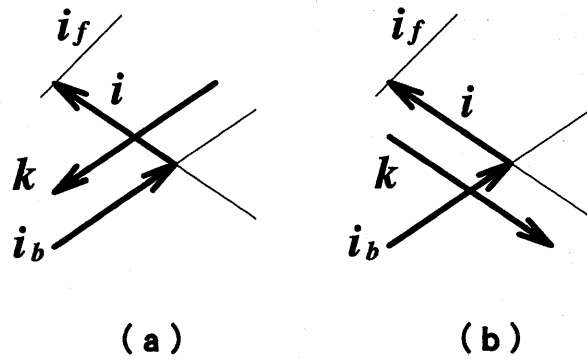


図4 点  $V_{i_b, i}$  と直線  $k$  の位置関係判定の配置例

に対応する配置を図4(a)に示す。条件1は直線  $i_b$  と  $i$  の交差の仕方を、条件2は辺  $k$  の二端点と直線  $i_b$  の位置関係を、条件3は辺  $k$  の二端点と直線  $i$  の位置関係を定める。

2. 次の三条件が全て満たされる時  $V_{i_b, i}$  は直線  $k$  の正領域にあるとみなし、 $F'$  の関数値を1とする。

[条件4]  $F(i, i_f, i_b) > 0$

[条件5]  $F(k_b, k, i_b) > 0, F(k, k_f, i_b) < 0$

[条件6]  $F(k_b, k, i) > 0, F(k, k_f, i) > 0$

この状況に対応する配置を図4(b)に示す。

3. 条件1～3のうちの偶数個の条件に対して不等号を逆向きにした場合に、三条件が全て満たされるならば  $V_{i_b, i}$  は直線  $k$  の正領域にあるとみなし、 $F'$  の関数値を1とする。奇数個の条件に対して同様のことをした場合に、三条件が全て満たされるならば  $V_{i_b, i}$  は直線  $k$  の負領域にあるとみなし、 $F'$  の関数値を-1とする。また条件4～6についても同様とする。
4. 以上に該当しない場合は  $F(i_b, i, k)$  の値をそのまま  $F'$  の関数値とする。

上記の仕様からわかるように、関数  $F'(i_b, i, k)$  では直線  $k$  に対する点の位置関係よりも、直線  $i_b, i$  に対する点の位置関係が優先する。そのため全ての位置関係に  $F'$  を用いるとすれば、 $k$  と  $i$  を交換した判定  $F'(k_b, k, i)$  の必要がある場合は上記の優先順位が入れ代わり、交換前の判定と整合が取れなくなる。従って  $F$  と  $F'$  を使い分けて優先順位を統一する必要がある。

本論文では辺番号の小さいほうを優先させ、 $k < i$  の時は  $F(i_b, i, k)$  を、 $k > i$  の時は  $F'(i_b, i, k)$  を用いる。また多角形相互間でもこの優先順位が統一されるように、新たに集合演算の対象となる多角形は、処理済みの多角形よりも大きい辺番号を持つようにする。

#### 4. 多角形の集合演算

##### 4.1 データ構造と処理概要

頂点や辺の接続関係を表わすトポロジカルなデータを図5に示す。辺番号のリストには反時計回りの順序で辺番号が格納される。交差情報のリストには、辺  $i$  に関する交差情報として  $t(i, k) \times k$  の値が格納される。但し  $k$  は交差相手の辺番号である。また  $t(i, k)$  は相手の多角形との交差状況を表わすもので、 $\pm 1$  の値を持つ。すなわち辺  $i$  上を進むと考えた場合、 $t(i, k) = 1$  ならば辺  $k$  との交差後に相手多角形の内側に入り、 $t(i, k) = -1$  の場合は外側に出る。 $t(i, k)$  の値は後述のように交差判定時に求める。なお、一辺上に複数の交差が存在する場合は、その辺の始点に近い順に交差情報を並べかえる。

処理前の辺番号のリストと交差情報のリストを検索することにより、生成図形の辺番号のリストを作成する。その際、別の多角形に完全に含まれた多角形は生成図形に含めない。図5は和集合演算の例である。

最後に生成図形の辺番号のリストから表示データを作る。このとき初めて、必要な交点位置が計算される。その際の誤差は表示図形の若干の歪みをもたらすが、致命的な問題ではない。

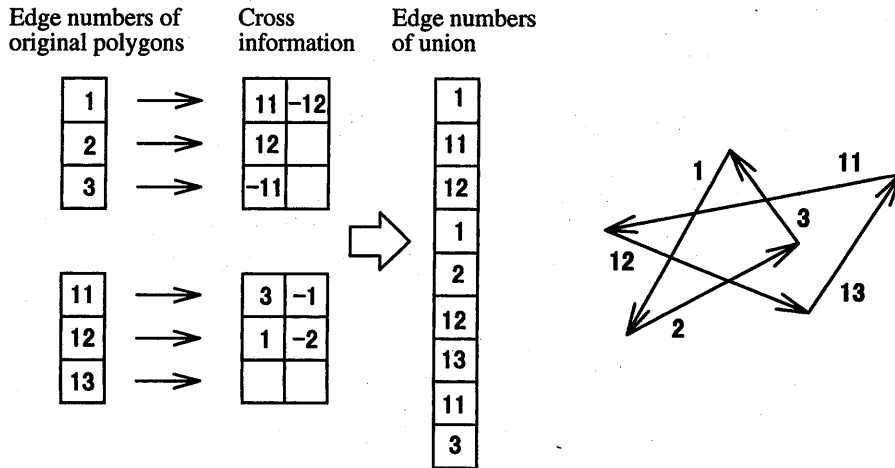


図5 集合演算で使われるトポロジカルなデータ

4.2 数値計算

集合演算の中で行う数値計算について説明する。

4.2.1 辺と辺の交差判定

異なる多角形にそれぞれ属す二辺  $i, k$  の交差条件は次のようになる。

$k < i$  の場合

$$F(i_b, i, k) \cdot F(i, i_p, k) < 0, \tag{4}$$

$$F'(k_b, k, i) \cdot F'(k, k_p, i) < 0 \tag{5}$$

が同時に成り立つこと。

$k > i$  の場合

$$F'(i_b, i, k) \cdot F'(i, i_p, k) < 0, \tag{6}$$

$$F(k_b, k, i) \cdot F(k, k_p, i) < 0 \tag{7}$$

が同時に成り立つこと。

なお、(4)または(6)の左辺の因数の符号から  $t(i, k)$  の値が求められる。(4)から求める場合は、

$$F(i_b, i, k) < 0, \quad F(i, i_p, k) > 0, \quad \text{ならば} \quad t(i, k) = 1,$$

$$F(i_b, i, k) > 0, \quad F(i, i_p, k) < 0, \quad \text{ならば} \quad t(i, k) = -1$$

とする。(6)から求める場合も同様である。

4.2.2 交差情報の並べ替え

この処理では一方の交点を作る直線と他方の交点との位置関係判定を応用する。例として辺  $i$  上の二交点  $V_{i, j_1}, V_{i, j_2}$  の並べ替えを考える。交差情報のリストには辺  $i$  に関する交差情報が  $t(i, j_1) \times j_1, t(i, j_2) \times j_2$  の順に格納されていると仮定する。この時、

$j_1 < j_2$  の場合

$$t(i, j_2) \times F'(j_1, i, j_2) > 0 \tag{8}$$

$j_1 > j_2$  の場合

$$t(i, j_1) \times F'(j_2, i, j_1) < 0 \tag{9}$$

ならば  $V_{i, j_2}$  の方が  $V_{i, j_1}$  より辺  $i$  の始点に近いので、交差情報を交換する。図6は交換を要する配置の例である。この判定は配置の特性を利用して、 $F$  より信頼性の高い  $F'$  だけで行うことができる。

なお、辺  $j_1$  と  $j_2$  が連結されている場合は、連結されていないほうの端点だけを使って判定を行うように、 $F'$  を修正しておけば無駄な退化の発生を避けることができる。

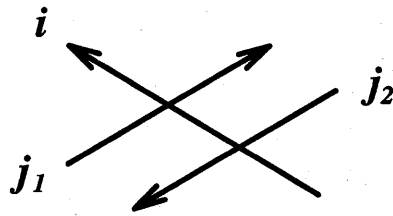


図6 交差情報の並べ替えを要する配置

#### 4.2.3 多角形同士の包含判定

二つの多角形の包含関係を判定するために、一方の多角形の任意の一頂点と他方の多角形との包含関係を調べる。この判定は、調査対象の頂点を始点に持つ辺をその向きに伸ばした半直線と、他方の多角形の全ての辺との交点総数の奇数/偶数判定により行う。この処理は辺と辺の交差判定処理と同様の過程であるので、両処理間の整合性は保たれる。

### 5. 実験

#### 5.1 プログラム

プログラムはC言語で作成し、数値データは全て倍精度実数型とした。

#### 5.2 実験モデル

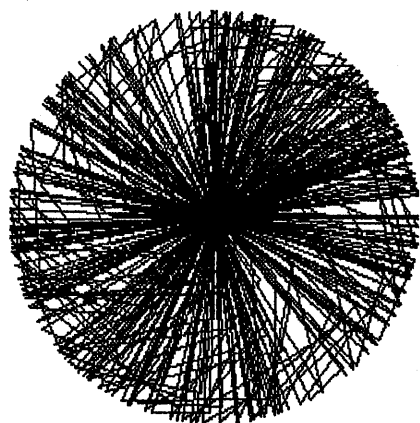
和集合演算の対象となるモデルを次のように二通り用意した。

【モデルA】 (1,1)を中心とする半径  $R_1$  (値は可変),  $R_2 = 1$ , ( $R_1 < R_2$ ) の二つの円に挟まれた領域内にランダムに配置した三角形(その二頂点はそれぞれ内側および外側の円周上にある)を、更に任意方向に  $R_1$  以下の任意の距離だけ平行移動させたもの。三角形の数は200である。

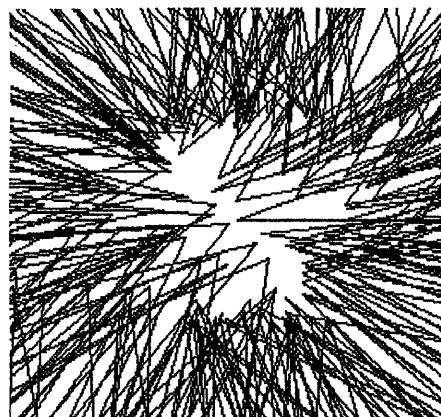
【モデルB】 モデルAの領域を中心の開き角約  $90^\circ$  の範囲にせばめて150個の三角形を同様に配置し、更に中心付近でそれらと  $90^\circ$  前後の角度で交わるように50個の三角形を配置したもの。

集合演算にあたっては、直前の演算でできた図形に三角形を一つずつ加えていく方式をとった。

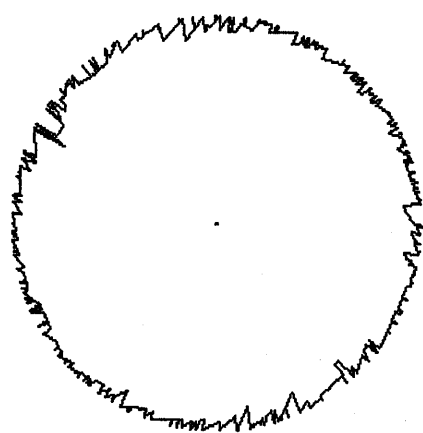
$R_1 = 10^{-14}$  のときの各モデルの配置と、4章で提案した改良アルゴリズムによる生成図形を、図7および図8に示す。それぞれの図において、(a)はモデルの全体の配置、(b)はその中心付近を  $5 \times 10^{13}$  倍に拡大したものであり、(c)および(d)は生成図形である。この時の  $R_1$  の値では、いずれのモデルにおいても計算途中で暴走やエラーが発生することなく、かつ生成図形にも異常が見られないなど、計算誤差の影響は表面化しなかった。



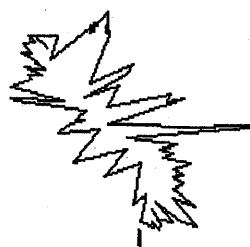
(a) Mutually close  
200 triangles.



(b) Central part of (a)  
magnified by  $5 \times 10^{13}$ .



(c) Union of (a).



(d) Central part of (c)  
magnified by  $5 \times 10^{13}$ .

図7 モデルAの集合演算の例

### 5.3 アルゴリズム改良の効果

アルゴリズム改良の効果を見るために、 $R_1$ を変えた場合の実験結果を表1に示す。比較のために、位置関係の関数  $F$  だけを用いた改良前のアルゴリズムによる結果も併せて掲げる。「成功した三角形の個数」欄は、何個の三角形の和集合演算ができたかを表す。この欄が200の場合は処理が中断されずに最後まで実行したことを示し、200未満の数値の場合は、更にもう1個の三角形を加えようとした時に矛盾が生じて処理が中断したことを示す。「図形の正しさ」欄では、○は生成図形が正常であることを、×は中央の穴ができなかったことを、△は中央の穴が自己交差など矛盾した図形になったことを示す。またアルゴリズム改良後の計算時間を、改良前を1とした場合の比で表す。

この表から、 $R_1$ が  $10^{-14}$ 以下になると計算誤差の影響が現れはじめ、 $10^{-16}$ 以下では影響が顕著になることがわかる。改良の効果は、 $R_1$ が  $10^{-14} \sim 10^{-15}$ 程度の時に若干見られるが、 $10^{-16}$ 以下ではほとんど見られない。また、アルゴリズムの改良は計算時間の大幅な増加を伴うことがわかる。



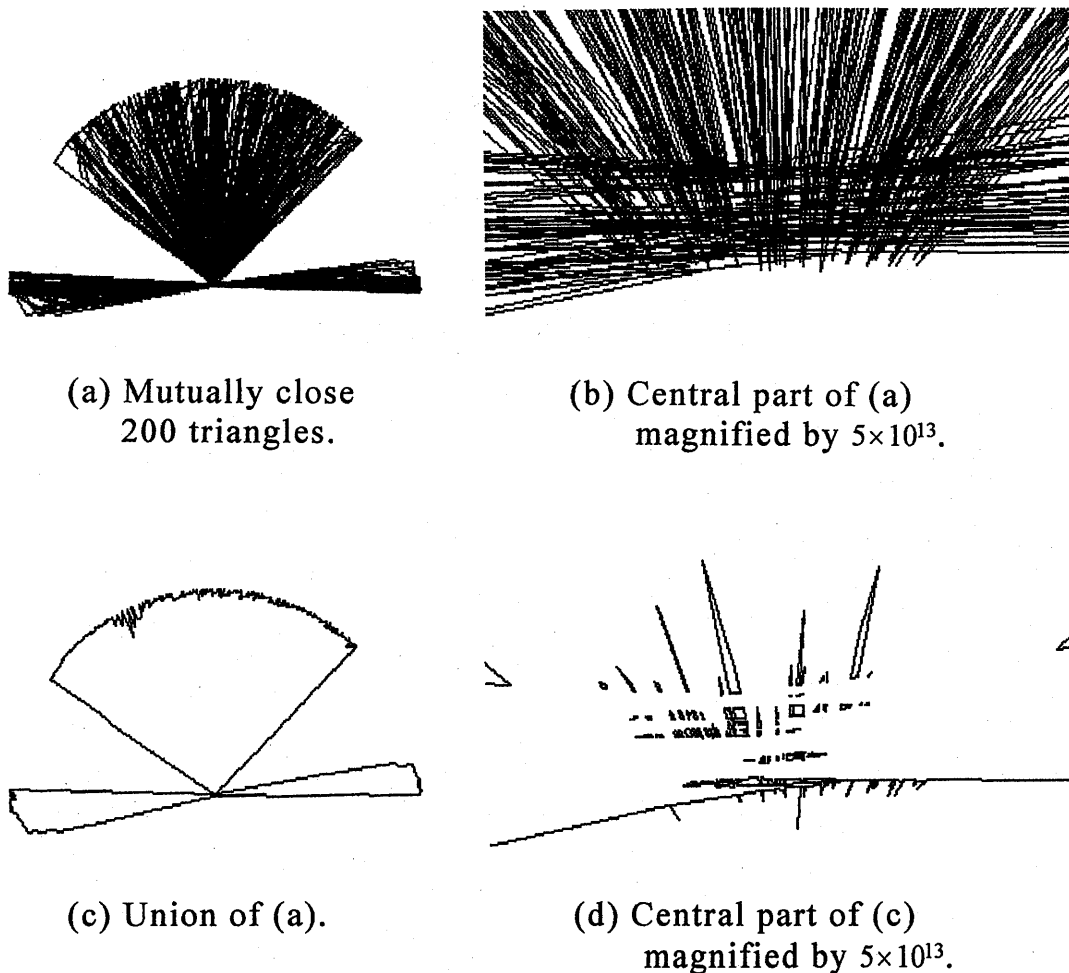


図8 モデルBの集合演算の例

表1 アルゴリズム改良の効果

モデル	$R_1$	アルゴリズム改良前		アルゴリズム改良後		
		成功した三角形の個数	図形の正しさ	成功した三角形の個数	図形の正しさ	計算時間
A	$10^{-13}$	200	○	200	○	1.68倍
	$10^{-14}$	200	×	200	○	1.80倍
	$10^{-15}$	200	△	200	△	1.75倍
	$10^{-16}$	5	—	5	—	—
	$10^{-17}$	3	—	6	—	—
B	$10^{-14}$	200	○	200	○	1.68倍
	$10^{-15}$	35	—	166	—	—
	$10^{-16}$	2	—	2	—	—
	$10^{-17}$	2	—	2	—	—

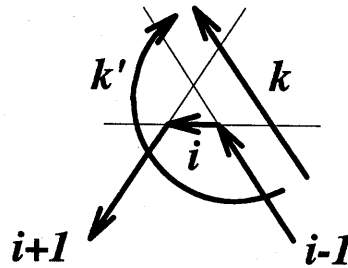


図9 辺*i*が極端に短い場合の誤判定

#### 5.4 処理中断の原因

処理が中断された理由のほとんどは、交差判定時に矛盾が生じたためである。すなわち、(4)および(5)式の左辺の第一因数が共に正、第二因数が共に負となった。この状況は交差条件を満たすが、交差の具体的な描写を試みても実現できないことからわかるように、矛盾を生じている。これは3.3での検討の範囲外にあるもので、調査の結果次のことがわかった。

図9は一方の多角形の三辺*i-1*, *i*, *i+1*と他方の多角形の一辺*k*との交差判定の状況を示している。但し辺*i*は極端に短く、辺*i*と直線*k*とは非常に接近している。実際の計算では、 $V_{i-1,i}$ と直線*k*との位置関係および $V_{i,i+1}$ と直線*k*との位置関係は、いずれも3.4節で定義された関数*F'*により判定された。前者の場合、3.4節の1.項目の条件1、および条件2と条件3の不等号を逆向きにした二条件が全て満たされ、 $V_{i-1,i}$ は直線*k*の正領域にあると判定された。後者の場合、3.4節の1., 2., 3.のいずれの項の条件も満たされず結局4.項すなわち関数*F*により判定されたが、この時計算誤差のために本来とは逆の位置関係が決定された。この状況を図では曲線*k'*で示す。以上により、上に述べた矛盾した状況が生じた。

処理中断の残りの理由は、交差情報の並べ替えが失敗したためであるが、その根本的な原因については現在調査中である。

### 6. 改良案の問題点

現在までに明らかになったアルゴリズム上の問題を示す。

#### 6.1 一方の多角形を基準にした位置関係の判定の問題

一般に、一方の多角形に属す直線を基準にして、他方の多角形の頂点の位置関係を求めるだけでは矛盾は生じない。なぜなら、複数の直線によって分割される領域のいずれに点があるかを示すにすぎないからである。これとは逆の立場すなわち後者の多角形を基準にした位置関係判定を行ってはじめて、その結果と上記の位置関係とが噛み合わず矛盾が生じる。この問題に対して、改良案は一方を基準にした位置関係だけを使って逆の立場の位置関係を間接的に引き出すことにより、その矛盾を防止しようとする。しかし前節で述べたように、3.4節の1., 2., 3.のいずれの項の条件も満たされず結局4.項すなわち関数*F*により判定(逆の立場の判定)を行わざるをえない場合が発生し、その時に矛盾が生じる可能性がある。これが改良案の大きな制約となっている。今後は逆の立場の判定を避ける確率をいかに高くするかが改良のポイントとなろう。

#### 6.2 集合演算をくり返し行う場合の問題

3.4節の1., 2.における条件1と条件4の関数*F*の値は、一方の多角形の情報だけを用いて得られる。このことから、多角形は自己交差などの矛盾を含んではならないことが要求される。しかし改良案を使って集合演算で作られた多角形は、2.1節で述べたようにこの要求を満たさない。従って更に別の図形との集合演算を行おうとすると判定を誤り、処理が破綻する恐れがある。

また、集合演算の結果できた図形には古い図形の頂点が含まれていないことがある。そこで更に集合演算を続けると、上記の関数*F*の値として使われる、以前の交差判定で使用した点と直線の位置関係が、別

の点と直線の位置関係にすりかえられる恐れがある。両者が異なる場合は矛盾が生じる可能性がある。このように、改良案では点と直線の位置関係について時系列的な意味での整合性がとられていない。

## 7. むすび

計算誤差による集合演算の破綻を防止するために、図形相互間の全ての位置関係を点と直線の位置関係に帰着させると共にその判定法を改良する提案を行った。実験の結果、本報告で示した改良案ではわずかに効果が見られたが十分とはいえない。また処理時間の大幅な増加も問題である。今後は数値的安定性を更に高めると共に、高速化に向けて検討する予定である。

## 謝 辞

本研究を進めるにあたりご支援頂いた熊本工業大学費田 寛教授に感謝します。最後に、本研究は、宮崎学術振興財団助成金の援助を得たことを記す。

## 参考文献

- (1)大沢 晃, “計算誤差による暴走のない図形演算アルゴリズム: スペースモデルによる実現,” 情処学論, 31,1, pp.42~55, Jan.1990.
- (2)Sugihara K. and Iri M., “A Solid Modelling System Free from Topological Inconsistency,” J.Inf.Process.,12,4, pp.380~393,1989.
- (3)稲垣 宏, 杉原 厚吉, 杉江 昇, “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,” 情処学論, 35, 1, pp.1~10, Jan.1994.
- (4)今井 敏行, 杉原 厚吉, “誤差による破綻の心配のない線分Voronoi図構成算法,” 情処学論,35,10, pp.1966~1977, Oct.1994.
- (5)杉原 厚吉, “計算幾何工学,” 培風館, pp.83~108,1994.

## 所属・氏名

金子 正光 辻 利則  
(宮崎公立大学)  
河野 俊弘  
(北九州高専・非常勤講師)  
中山 泰雄  
(熊本学園大学)

