

計算誤差による暴走のない図形演算アルゴリズム： 4K-3 スペースモデルによる実現

(株) 日立製作所 京浜工業専門学院 大沢 晃

1. はじめに

計算誤差による図形演算プログラムの暴走は従来大きな未解決課題であった¹⁾。ここではスペースモデル²⁾のデータ構造SPF(Space Pointer File: 図1)を用いて、トポロジーの矛盾を防ぐ新しい対策法を提案する。

2. トポロジー矛盾と従来の問題点

図形の'OR'で頂点と辺が近接していると、計算誤差で図2(i)の如く図形のトポロジーが矛盾を起こし、プログラムが暴走することがある。従来「非常に接近した図形要素は同一点を占める」として対策されていたが、その方法では図3(i)の矛盾した図形を発生し、反って暴走する場合があった。

3. スペースモデルによる対策の原理

3.1 暴走防止の条件：矛盾のないSPF

SPFは交点を陽に持つため、交点計算と誤差により発生する矛盾の対策はSPF生成時で行う。集合演算などは生成済のSPF上で行うが、これらのプログラムは矛盾のないSPF上では暴走せぬように作る。またそこでは交叉計算はなく誤差で矛盾を作ることはない。従って計算誤差による矛盾が起らないSPF生成法が、暴走防止の条件になる。

3.2 トポロジー優先交叉判定法

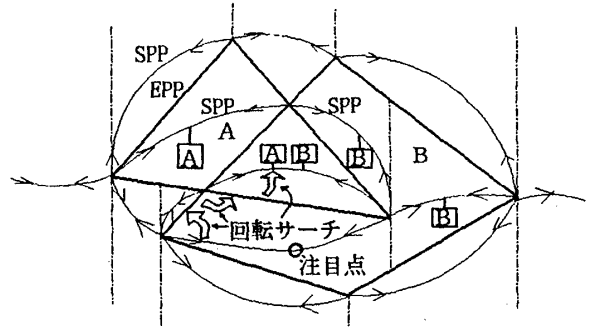
SPFでは回転サーチなどポインタ検索により注目点の周囲を局所的に調べられる。図2(ii)の A_1 入力で交点 c_1 を設置後 t 点の上側を調べると B が見つかる。このため a_2 が B より上側なら A_2 の交叉は交点計算なしで判明し矛盾しない。 c_2 の座標値計算は必ず答を出すため、 A_2 と B を含む二直線の交叉計算で行う。平行線では分母のゼロを検出して平行部中点を、また誤差で交点が線分範囲外になったら線分端点を交点とする。

3.3 トポジカルな近接点の表現

図3(ii) c_1, c_2 は同座標値でも区別し、長さゼロの辺で接続する。トポジカルな並び順はSPP(Space Pointer Pair), EPP(Edge Pointer Pair)の X^+ 方向属性で表現し矛盾を防ぐ。

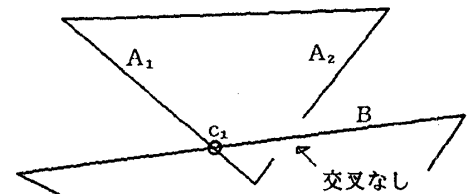
3.4 トポジカルな制約と次のイベント

多角形は始点 a_0 を入力後辺 $a_1 t$ を伸ばし、空間境界を通過するイベント毎にポインタを更新し、外形を一巡して入力する。図4で次のイベント E_{j+1} はSPPtで表される空間の外形辺 By^+ , By^- , Bx^+ と入力との各交点 C^+ , C^- , b , 又は次の頂点 a_{i+1} のうち入力先端 t に最も近い点となる。 Bx^+ は By^+ , By^- の端点

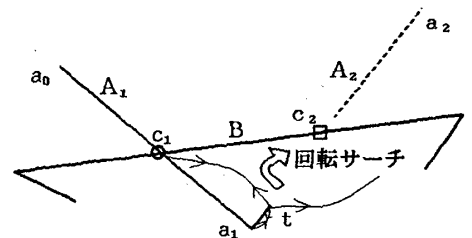


SPP(Space Pointer Pair)による空間の表現
□A□BはSPPに付加し、色などを表す空間属性

注目点の周囲はポインタで検索出来る
Y方向検索は図示の回転サーチによる
図1. スペースモデルのデータ構造



(i) トポロジーの矛盾：交叉なし



(ii) Bを検索：SPFによる正しい交叉判定
図2 計算誤差による矛盾とSPFによる対策

To Prevent Failure of Geometrical Operation Program Caused by Computational Error.

Akira OHSAWA
HITACHI, Ltd.

b^+ , b^- , 又はSPPtの先端点 p のうち X 座標値が最も t に近い点を通る境界線である。 E_{j+1} はこれらの内、入力空間形状のトポロジカルな制約から、誤差で順序を誤っても矛盾を起こさぬ点のみを引き数とし、 X 値を比較する関数 $NXT()$ により求める。このため入力空間形状を図4(i)~(iii)など X^+ , X^- 方向入力各36種、合計72種に分類した。

3.5 多角形を閉じる場合の矛盾

始点付近の既設辺のため始点終点が別空間と見做されて閉じない場合には、終点通過後も入力を外形沿いに伸ばし、入力図形の既入力部と接触した点で両者を接続、始点からそこまでを消去する。

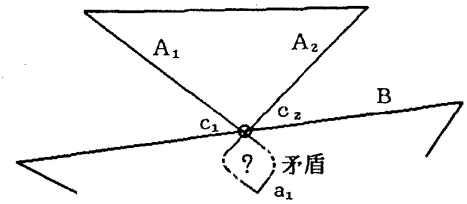
4. 実験結果: 写真1.半径10(整数)の小円の周りで長辺10,000の三角形200個を乱数角度で回転し重ねた。写真2.その中心部の1,000倍表示。中心付近多数の交点で切り捨て誤差が出るが暴走しない。

5. まとめ

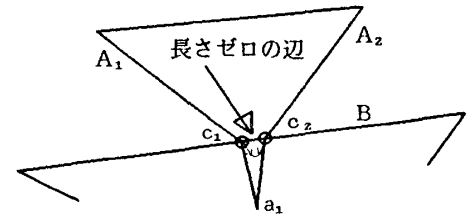
スペースモデルは二次元・三次元、注目、局所集合演算、運動図形の衝突検出、等が $O(1)$ の対話速度で実現できる新しい方式である。今回の高信頼度化により実用化の基礎が出来たと考えられる。

★参考文献

- 1)大沢, 他: スペースモデルによる二次元図形処理システムの試作, 情報処理学会論文誌, Vol.29, No.10, pp.933-943(1988)
- 2)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資料, アルゴリズム1-1(1988)

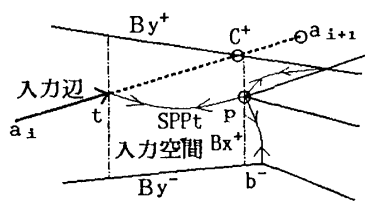


(i) c_1, c_2 を同一点とみなす: 矛盾

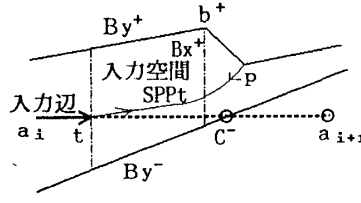


(ii) c_1, c_2 は同座標でも別々とする: SPf対策

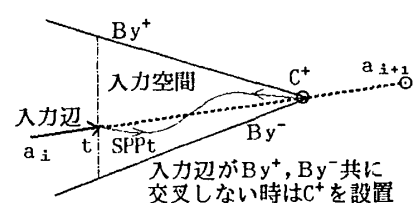
図3 近接点を同一点とみなす矛盾と対策



(i) $E_{j+1} = NXT(C^+, p, a_{i+1})$



(ii) $E_{j+1} = NXT(b^+, C^-, a_{i+1})$



(iii) $E_{j+1} = NXT(C^+, a_{i+1})$

図4. 入力空間形状と次のイベントを求める関数 $NXT()$ の例: 順序を誤っても矛盾を起こさぬ点のみを引き数とする

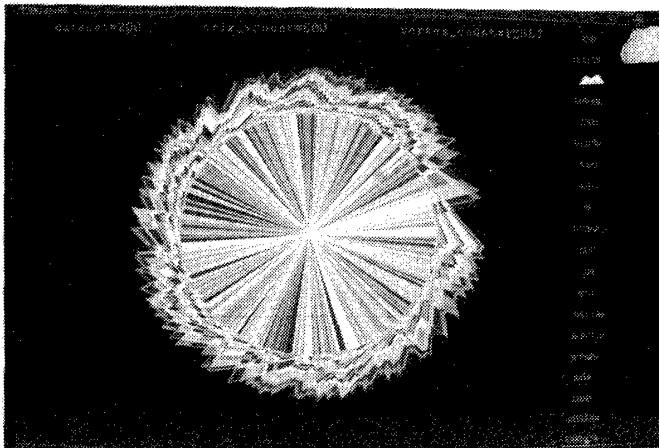


写真1. 三角形200個を乱数角度で回転し重ね表示

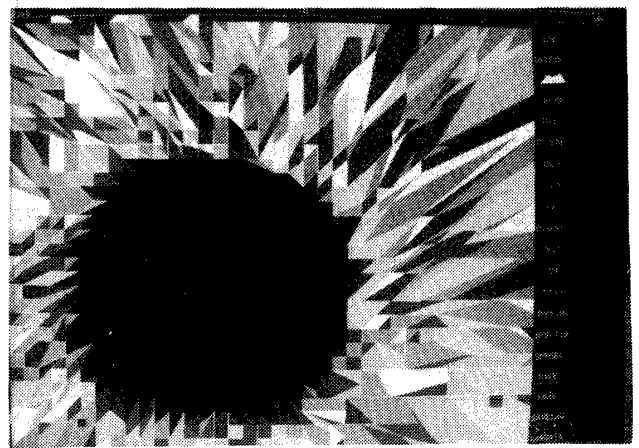


写真2. 写真1の中心部1,000倍拡大表示