可視多角形アルゴリズム

3L-2

伊藤　栄治　　吉澤　恵資　　大森　健児
法政大学　工学部

1.はじめに

可視多角形とは、多角形の内部の点から見える点の集合で星状多角形を成しているものである。ここでは、単純な多角形の内点からの可視多角形を求めるアルゴリズムを述べる。

単純な多角形において、内点からの可視多角形を求める問題は、一般にO(n)の処理時間を有する算法が知られている[1][2]。それらの算法の中で、アルゴリズムが単純でかつ、効率がよいとされるLEEの算法では、多角形を一巡しながら、その時点までの可視多角形を求める方式をとっている。そのため、最終的には隠れてしまう、見える領域と見えない領域の境界点（視線と辺との交点）をも求めてしまう。また、可視点の探索は、視線と辺との交差判定により行っている。

一方、本アルゴリズムでは、前処理により、各頂点に多角形の構造的な情報を与えることによって、可視点の探索を外積計算のみで行えるようにした。そのため、手間はO(n)であるが、LEEの算法と比較した結果、多角形の頂点数が増加していった場合の処理時間の増分をLEEの算法より少なくすることができた。

本論文では、はじめにLEEの算法とその問題点を示し、次に本アルゴリズムを述べる。

2．ＬＥＥの算法と問題点

LEEの算法では、単純な多角形の内点からの可視多角形を頂点のリストを一回りしながら、それまでに辿った時点での視点から見える部分をスタックに積み込んでいく。そして、一回りし終えた時点でスタックの要素が可視多角形に頂点列になっているようにする。それぞれの頂点が可視点かどうかは、辿っていく頂点の推移と、視線と多角形を構成している辺との交差判定により判断している。

•問題点

LEEの算法の非効率的な点は、無駄となる、可視領域と不可視領域の境界点を求めてしまうことである。LEEの算法では、反時計廻りに頂点を辿るため、視点に対して左からのくい込みが存在する場合は、特に非効率的になる。この例を図-1に示す。

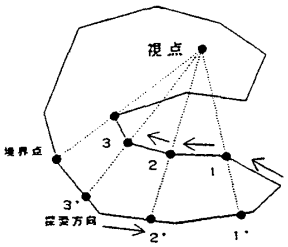

図-1 LEEの算法の非効率な探索

この場合、頂点1,2,3と辿って行くと、最終的に見えなくなるこれらに対する境界点(頂点1',2',3')をそれぞれ求めてしまう。これは、頂点列を辿っていく際に、多角形の構造に対して、何も情報を持っていないためである。そこで、本アルゴリズムは、各頂点の象限や視線の推移方向やくい込み部分などの多角形の構造に関しての情報を前処理で得ることにより、可視点の探索を外積計算のみで行えるようにした。さらに、境界点の算出(線分同士の交点計算)は非常に処理負担がかかるので、本アルゴリズムでは、最終的に境界となる点のみを最後に求めることにした。

A visualizing algorithm for a simple polygon
Eiji Ito,Keiji Yosizawa,kenji Ohmori
Hosei University

3.外積計算による可視多角形を求める手法

可視多角形は、元の多角形の頂点のうち視点から見える頂点列に、可視領域と不可視領域の境界となる可視境界点を加えた頂点列である(図-2)。単純な多角形の場合、視点から見えない頂点は、視点に対してのくい込みにより、その頂点への視線が妨げられるために見えなくなる。そこで、本アルゴリズムでは、他の頂点への視線を妨げる作用をするくい込み部分(可視不可視境界点)を検出し、そのくい込みにより見えなくなる点を元の頂点列から取り除くことによって可視頂点列を求め、最後にその頂点列に可視境界点を加えて可視多角形を求める。

次に、この手法の手順を示す。

(1)前処理として、各頂点の視線の推移方向と象限をそれぞれ求め、可視不可視境界点を求める。
(2)それぞれの可視不可視境界点に対する不可視頂点の探索を行なう。
(3)くい込みの重なりを解消する。
(4)可視境界点を求め可視頂点列に加える。

(1)前処理

はじめに、各頂点に対して視線の推移方向と象限を求める。視線の推移方向は、4.で示す外積計算により求める。象限は、x軸(正と負)とy軸(正と負)を含めて象限を8つに分け、頂点が時計廻り方向に推移すると減少し、反時計廻り方向に推移すると増加するようにつける。一つ前の頂点をP0(x0,y0)、求める頂点をP1(x1,y1)とすると象限を求める式は、以下のようになる。

Quadrant(P1)=Quadrant(P0)+(-1or1)*Diff
Diff=|signum(x0)-signum(x1)|+|signum(y0)-signum(y1)|

•可視不可視境界点の算出

可視不可視境界点は、視点に対して内側にくい込んでいる部分の頂点(凹頂点)であり、この点を境に視線の推移方向が変化する点(以下変凹頂点と呼ぶ)である。しかし、図-2の多角形のようにくい込み部分に曲がりが生じている場合には、変凹頂点であっても可視不可視境界点とならない点がある(図-2の頂点4)。くい込み部分の曲がりは、多角形の頂点列を辿っていった場合に、変凹頂点または変凸頂点が2つ以上連続しているときに起こる。また、この曲がりは、変凹凸頂点が連続して出てきた数だけ必ず凹凸が反対の変凹頂点がでてきて元にもどる。そこで、多角形の頂点列を辿りながら可視不可視境界点を求めていく際に、変凹頂点と変凸頂点のカウントを取ることにより、図-2の頂点4のような点を可視不可視境界点と判断しないようにする。

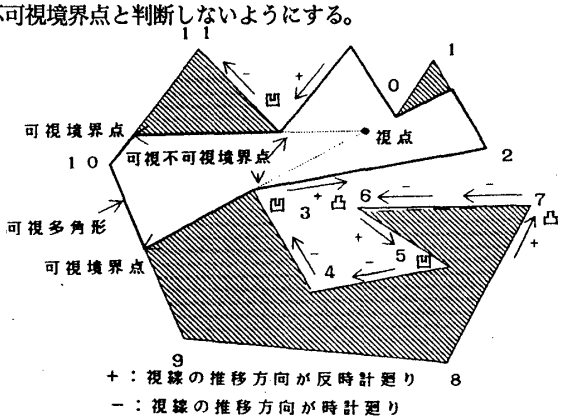

図-2　可視多角形

(2)外積計算による不可視頂点の探索

本アルゴリズムでは、反時計廻りを基準としているので、不可視頂点の探索は、可視不可視境界点から視線の推移が時計廻りの方向への頂点について行う。探索には、処理負担の少ない外積と

前処理で求めた象限を用いる。例えば、図-3において可視不可視境界点Aについての探索は、OA×OB,OA×OC...と外積の符号を調べていき、符号が変化するまでの頂点を不可視頂点とする(頂点B,C,D)。不可視頂点の探索において各頂点の象限を必要とする理由は、外積のみで探索を行なうと、図-4のように多角形がとぐろを巻いている場合には、180度の整数倍で符号が変化する外積の性質により探索が途中で終ってしまうためである。そこで、多角形がとぐろを巻いているかどうか判断できるように、探索の際に、前に述べた象限を外積計算とともに用いる。また、外積の符号が変化し探索が終わった時点の線分を、可視境界点を求める際に用いるため、可視不可視境界点と対にして記憶しておく。

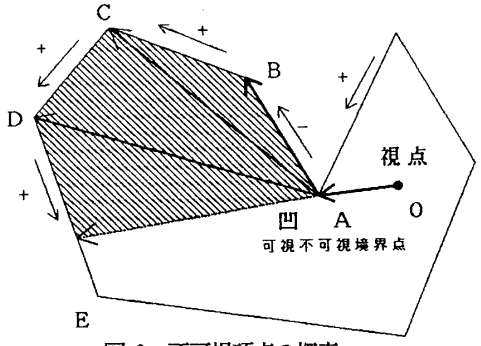

図-3 不可視頂点の探索

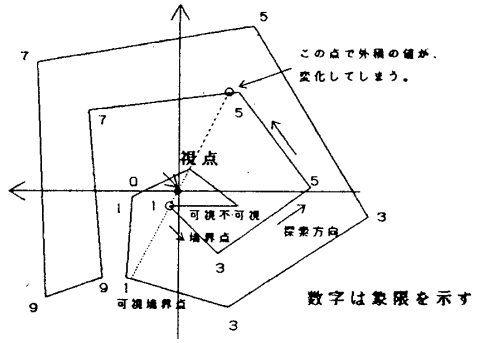

図-4 多角形のとぐろ

(3)くい込みの重なりの解消

多角形が複雑になっている場合には、幾つかのくい込み部分が重なりあい、あるくい込みが他のくい込み部分を隠してしまうことがある。これまでは、一つのくい込みに着目して不可視頂点の探索を述べてきたが、くい込みが2つ以上の場合には、それぞれのくい込みの関係を調べる必要がある。くい込みの関係は、可視頂点を頂点列順に並べると、必ず元の頂点列と同じ方向に推移することと、前に述べた象限を用いることにより容易に求めることができる。求まったくい込みの関係から、新たにくい込の重なりによる不可視頂点を探索する。

(4)可視境界点の算出

可視境界点は、不可視頂点の探索がすべて終わった時点で残った可視不可視境界点について、視点から可視不可視境界点を通る半直線と、その可視不可視境界点に対する不可視頂点の探索で記憶した線分との交点を求めればよい。

4. 比較と結果

LEEの算法と本アルゴリズムの違いは、LEEの算法では、頂点列を順に辿ってその時点までの可視領域を求めているので、無駄な交点計算が多くなり、また頂点が可視点かどうかの判定に交差判定を用いている。一方、本アルゴリズムでは、前処理を行うことによりすべて外積計算のみで可視点を求めている。また可視境界点を求めるための交点計算も最小限に抑えている。次に、外積計算の式と交点計算式を示す。

2直線の交点計算式

2直線の方程式 (y2-y1)x-(x2-x1)y=y2*x1-x2*y1
(y4-y3)x-(x4-x3)y=y4*x3-x4*y3

交点(x0,y0)は

x0=D1/D　　y0=D2/D

D1={(x4-x3)(x2*y1-y2*x1)-(x2-x1)(x4*y3-y4*x3)}

D2={(y4-y3)(x2*y1-y2*x1)-(y2-y1)(x4*y3-y4*x3)}

D=(y4-y3)(x2-x1)-(x4-x3)(y2-y1)

外積計算の式

(x1,y1)と(x2,y2)の外積は

y2*x1-y1*x2

また交差判定は、外積計算を4回行うことによりできる。

これらの計算式から、外積計算は、交点計算や交差判定より処理負担が少ないことがわかる。従って、外積計算により可視多角形を求める本アルゴリズムは、LEEの算法より処理負担が少ない。一方、記憶空間に関しては、LEEの算法では、n(頂点数)、本アルゴリズムでは、3nの記憶空間を必要としている。本アルゴリズムは、XEROX 1161AI上でSmalltalk-80により開発した。

次に、これら2つの手法の処理時間を比較をするため、2つの測定を行った。

(測定1)各頂点数(10,20...)の多角形をランダムにそれぞれ10個発生させて、それぞれの頂点数の多角形について平均処理時間を測定した。表-1に各頂点数の平均処理時間を図-5に頂点数と平均処理時間の関係を示す。

頂点数／ms	10	20	30	40	50	100
LEEの算法	284	836	1437	2108	2873	5605
本アルゴリズム	250	528	794	1010	1274	2352

表-1 各頂点数の平均処理時間

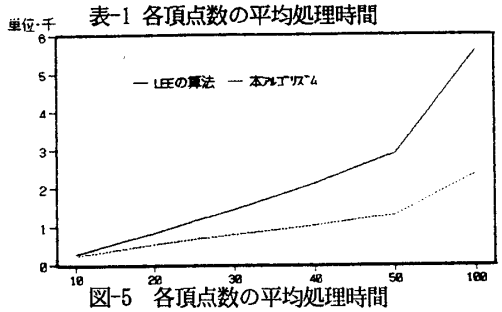

図-5 各頂点数の平均処理時間

(測定2)くい込みがない簡単な多角形とくい込みが幾つかある複雑な多角形について処理時間を測定した。その結果を表-2に示す。

頂点数30個	簡単な多角形	複雑な多角形
LEEの算法	510 /ms	1420 /ms
本アルゴリズム	610 /ms	860 /ms

表-2 各多角形の平均処理時間

5. 考察

測定1より、本アルゴリズムはLEEの算法に比べ、頂点数が増加した場合の処理時間の増加が抑えられていることがわかる。これは、外積計算と交差判定,交点計算との処理負担の違いからきていると考えられる。また、測定2より、くい込みのない簡単な多角形では、前処理のため本アルゴリズムの方が処理時間が多いが、複雑な多角形では、非常に処理時間が少ないことがわかる。従って、本アルゴリズムは、多角形の頂点数が増加するほどまた、多角形が複雑になるほど有益になる。

参考文献

[1]D.T.LEE:Visibility of a simple polygon,1983
[2]El-Gindy and D.Avis:A linear algorithm for computing the visibility polygon from a point,1981
[3]伊理 正夫他:bit別冊:計算幾何学と地理情報処理,1986.9