社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－AL－111 (10)
2007／3／9

窓のある単純多角形間の交差判定最適アルゴリズム

仁尾 都
明星大学経済学部
nio@mi.meisei-u.ac.jp

窓のない多角形 P,Q 間の交差判定は複雑な手順を持つ多角形の三角形分割アルゴリズムを利用することにより $O(n_P+n_Q)$の最適手間でできることが知られている．今回，窓のある単純多角形間 P,Q の交差判定も $O(n_P+n_Q)$の最適手間でできることを示す．また，Q が星状の窓を持つ星状多角形の場合は，多角形を三角形分割しなくとも，$O(n_P+n_Q)$の最適手間で交差判定できることを示す．

Optimal algorithms for intersection detection between two simple polygons with windows

Misato Nio
School of Economics, Meisei University
nio@mi.meisei-u.ac.jp

An algorithm using triangulating polygons is proposed for the intersection detection between simple polygons with windows. Furthermore, another algorithm without triangulating polygons is proposed for the intersection detection between a simple polygon with windows and a star-shaped polygon with star-shaped windows. The processing time of both is $O(n_P+n_Q)$ and optimum.

１．はじめに

本論文は多角形 P,Q が与えられた時，$P \cap Q = \phi$ を判定する問題に関する．交差判定アルゴリズムは図形最適配置やロボット制御問題，シミューションなどの多くの応用分野で重要な役割を果たしている．多角形の三角形分割アルゴリズム[1] を用いることによって，窓のない二つの単純多角形 P,Q の交差を $O(n_P+n_Q)$の最適手間で判定できることが知られている．今回，〔1〕を利用して，窓のある単純多角形間の交差判定を $O(n_P+n_Q)$の最適手間で判定できることを示す．更に窓のある単純多角形と星状の窓のある星状の多角形との場合は，〔1〕を用いずとも $O(n_P+n_Q)$の最適手間で交差判定できることが簡単に証明できること示す．

２．窓のある単純多角形 P，Q の交差判定アルゴリズム

２．１．準備

定義１．窓のある単純多角形 $P \equiv P_0 - \cup_{i=1\sim k_P} P_i$，$P_i$:単純多角形（$0 \leqq i \leqq k_P$），$P_0 \supset P_i (i \neq 0), P_i \cap$

$P_j = \phi$ $(1 \leqq i \leqq k_P, 1 \leqq j \leqq k_P, i \neq j)$.

定義２．$n_{Pi} \equiv |P_i|$, $n_P \equiv \Sigma n_{pi}$

定義３．$\partial P \equiv$ 単純多角形 P の境界.

定義４. 関数 closed_line_not_intersect(P,Q) ≡ 単純多角形 P,Q に対し，$\partial P \cap \partial Q = \phi$ なら true, そうでないなら false の値をとる関数.

なお，窓のある単純多角形 P，Q に対し，$P \cap Q \neq \phi$ のとき，P,Q は交差するとする.

図１．窓のある単純多角形 P,Q の交差状態と非交差状態

窓のない単純多角形の三角形分割のための最適アルゴリズム〔1〕により，次の定理１が成立することが知られている.

〔定理１〕P，Q を窓のない単純多角形とするとき，$\partial P \cap \partial Q = \phi$ の判定は $O(n_P + n_Q)$ の最適手間で可能である.

〔定理１の系〕closed_line_not_intersect(P,Q) の処理は $O(n_P + n_Q)$ の最適手間で可能である.

〔定理２〕単純多角形 $P \ni$ 点 q の判定は $O(n_P)$ の最適手間で可能である[2].

〔定理２の系〕窓のある単純多角形 $P \ni$ 点 q の判定は $O(n_P)$ の最適手間で可能である.

〔補題１〕P,Q は窓のある単純多角形とし，$p \in P$，$\exists i, p \in Q_i$，$P \not\subset Q_i$ ならば $P \cap Q \neq \phi$.

〔証明〕$P \not\subset Q_i$ であるから，$\exists r \in P$，$r \notin Q_i$. したがって，p と r はジョルダン曲線 ∂Q_i が２分する空間に分かれて存在する．一方，$p \in P$ と $r \in P$ の間には P に含まれる道が存在する．その道は必ず ∂Q_i と交差するのでその交点の１点を s とすれば，$s \in Q$. 一方，$s \in P$ でもある．したがって，$P \cap Q \neq \phi$.■

２．２．窓のある単純多角形 P，Q の交差判定アルゴリズム

(1)p=any vertex of P_0 ; q=any vertex of Q_0 　　//P_0 と Q_0 の任意の頂点を p,q とする

(2)if $p \in Q_0$ then

(3) 　　for i=1 to k_Q 　　//p を内点とする Q の窓 Q_i を探す

(4) 　　　　if $p \in Q_i - \partial Q_i$ then

(5) 　　　　　　if closed_line_not_intersect(Q_i,P_0)= true then

```
(6)                 print "P∩Q=φ" ;break          //なぜなら Qi⊃P0
(7)             else print "P∩Q≠φ" ;break         //なぜなら補題 1
(8)     print "P∩Q≠φ"          //なぜなら p∈Q
(9)elseif q∈P0 then
(10)    for i=1 to kP              //q を内点とする P の窓 Pi を探す
(11)        if q∈Pi - ∂Pi then
(12)            if closed_line_not_intersect(Pi,Q0)=true then
(13)                print "P∩Q=φ" ;break          //なぜなら Pi⊃Q0
(14)            else print "P∩Q≠φ" ;break         //なぜなら補題 1
(15)    print "P∩Q≠φ"          //なぜなら q∈P
(16)elseif closed_line_not_intersect(P0 ,Q0)=true then
(17)     print "P∩Q=φ"         //なぜなら∂P∩∂Q=φ ,P⊄Q,Q⊄P であるから
(18)else print "P∩Q≠φ"
```

定理２より(2)，(4)は手間が $O(n_Q)$の，(9)，(11)は手間が $O(n_P)$の最適処理が可能であり，closed_line_not_intersect は定理１の系より手間が $O(n_P+n_Q)$の最適処理が可能であるので上記アルゴリズムの手間は $O(n_P+n_Q)$である．

〔定理３〕P，Q を窓のある単純多角形とするとき，P，Q の交差判定は $O(n_P+n_Q)$の最適手間で可能である．

３．窓のある単純多角形 P と星状の窓のある星状多角形 Q との交差判定アルゴリズム

３．１．準備

定義５．$N(q,P) \equiv \{p \mid p$ は，点 q を光源とし偏角を $\forall\theta$ $(0 \leqq \theta \leqq 2\pi)$ とする光線が最初に単純多角形 P の辺に交差する点．ただし，交差しない場合，q を p に加える$\}$を偏角順に結んで作る多角形（図２）．可視多角形とも呼ぶ．

定義６．$F(q,P) \equiv \{p \mid p$ は，点 q を光源とし偏角を $\forall\theta$ $(0 \leqq \theta \leqq 2\pi)$ とする光線が最後に単純多角形 P の辺に交差する点．ただし，交差しない場合，q を p に加える$\}$を偏角順に結んで作る多角形（図２）．

定義７．$\partial' P(k) \equiv P$ は星状多角形で，P のカーネル $k \in \partial P$ の場合，k を含む P の辺を ∂P から除いた折れ線とし，その他の場合は $\partial' P(k) = \partial P$ とする．

定義８．関数 not_intersect(k,P,Q)≡点 k をカーネルとして共有する窓を持たない星状多角形 P,Q に対し，$\partial' P(k) \cap Q = \phi$ なら true，そうでないなら false の値をとる関数．

〔定理４〕N(q,P)の計算は $O(n_P)$の最適手間で可能である[3]．

次の定理５は定理４の証明とまったく同様である．

〔定理５〕$F(q,P)$の計算は$O(n_P)$の最適手間で可能である．

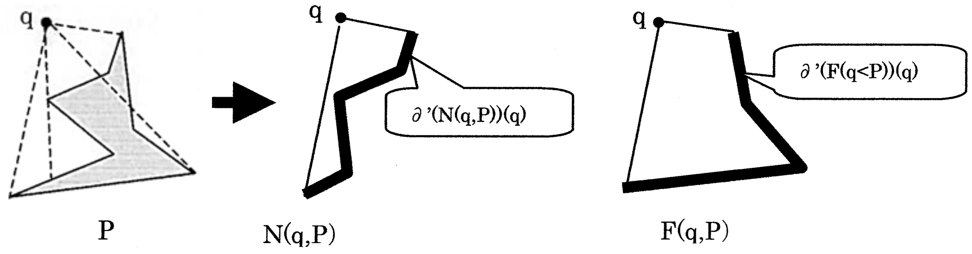

図２．$q \notin P$の場合の可視多角形$N(q,P)$と$F(q,P)$

次の補題２，３，４は自明である．

〔補題２〕$N(q,P)$と$F(q,P)$は点qをカーネルとして共有する星状多角形である．

〔補題３〕$q \in P$なら$N(q,P) \subseteq P$.

〔補題４〕$F(q,P) \supseteq P$.

〔補題５〕not_intersect(k,P,Q)は$O(n_P+n_Q)$の最適手間で処理可能である．
〔証明〕not_intersect(k,P,Q)を処理するアルゴリズムを下記に示す．

(1)k を含む高々二つの P の辺を除く P のすべての辺（r_i,r_{i+1}）（$1 \leqq i \leqq n_P$）と，k を含む高々二つの Q の辺を除く Q のすべての辺とに対し，k を基準点とする偏角によりそれらの頂点をマージしてソートし，辺を対応させる．この結果，（r_i,r_{i+1}）に対応する Q の辺が全くなければ true として全手順を終了．

(2)すべての（r_i,r_{i+1}）（$1 \leqq i \leqq n_P$）について，下記を繰り返す．

　(2.1)k を頂点とし（r_i,r_{i+1}）に対応する Q の辺のうち，r_i と r_{i+1} を通る凸錐に含まれる∂Qの部分集合である折れ線の一部分が（r_i,r_{i+1}）の右半平面内に含まれれば，false として全手順を終了．

(3)true として全手順を終了．

(1)で生成する対応組の数は$O(n_P+n_Q)$であり，その手間も$O(n_P+n_Q)$である．(2)の手間は，対応組の数が$O(n_P+n_Q)$であるため$O(n_P+n_Q)$となる．結果，全手順に要する手間は$O(n_P+n_Q)$であり，

またこの計算量の下限も明らかに $O(n_P+n_Q)$ であるので上記アルゴリズムは最適である．■

〔補題６〕P,Q は窓のある単純多角形とし，$p\in P, p\notin Q_0, q\in Q_0, q\in P$ ならば $P\cap Q\neq\phi$.

〔証明〕p と q はジョルダン曲線 ∂Q_0 が２分する空間に分かれて存在する．一方，$p\in P$ と $q\in P$ の間には P に含まれる道が存在する．その道は必ず ∂Q_0 と交差するのでその交点の１点を s とすれば，$s\in Q$．一方，$s\in P$ でもある．したがって，$P\cap Q\neq\phi$．

３．２．窓のある単純多角形 P と星状の窓のある星状多角形 Q との交差判定アルゴリズム

(1)p=any vertex of P_0 //P_0 の任意の頂点を p とする
(2)if $p\in Q_0$ then
(3) for i=1 to k_Q //p を内点とする Q の窓 Q_i を探す
(4) if $p\in Q_i - \partial Q_i$ then
(5) if not_intersect(q_i,Q_i,F(q_i,P_0))= true then //q_i は Q_i のカーネル
(6) print "$P\cap Q=\phi$" ;break //なぜなら補題４より $Q_i\supset F(q_i,P_0)\supseteq P_0$
(7) else print "$P\cap Q\neq\phi$" ;break //自明
(8) print "$P\cap Q\neq\phi$" //なぜなら $p\in Q$
(9)elseif $q_0\in P_0$ then //q_0 は Q_0 のカーネルで，$q_0\in Q_0$
(10) for i=1 to k_P //q_0 を内点とする P の窓 P_i を探す
(11) if $q_0\in P_i - \partial P_i$ then
(12) if not_intersect(q_0,N(q_0,P_i) ,Q_0)=true then
(13) print "$P\cap Q=\phi$" ;break //なぜなら補題３より $Q_0\subset N(q_0,P_i)\subseteq P_i$
(14) else print "$P\cap Q\neq\phi$" ;break //自明
(15) print "$P\cap Q\neq\phi$" //なぜなら $q_0\in P$ であるため q_0 を q と見なせば補題６が成立
(16)elseif not_intersect(q_0,N(q_0,P_0) ,Q_0)=true then
(17) print "$P\cap Q=\phi$" //なぜなら $\partial P\cap\partial Q=\phi, P\not\subset Q, Q\not\subset P$ であるから
(18)else print "$P\cap Q\neq\phi$"

本アルゴリズムが正しい理由は定理２より(2)，(4)は手間が $O(n_Q)$ の，(9)，(11)は手間が $O(n_P)$ の最適処理が可能であり，not_intersect は補題５より $O(n_P+n_Q)$ の最適処理が可能であるので上記アルゴリズムの処理時間は $O(n_P+n_Q)$ である．

〔定理６〕窓のある単純多角形 P と星状の窓のある星状多角形 Q の交差判定は多角形の三角形分割手段を用いないで $O(n_P+n_Q)$ の最適手間でできる．

〔定理６の系〕窓のある単純多角形 P と星状多角形 Q の交差判定は多角形の三角形分割手段を用いないで $O(n_P+n_Q)$ の最適手間でできる．

4．結果の検討

定理２の系の証明は定理２の証明と同じように，q を光源とし，x 軸の正方向に平行に進む光線が単純多角形多角形 ∂P_i（$0 \leqq i \leqq k_P$）と交差する回数を $O(n_P)$の最適手間で求め，回数が偶数であれば $P \ni q$ である事を利用する．

一方，定理２において，点 q を窓のある単純多角形 Q に拡張したのが定理３であると見なすことができる．従って，第２章のアルゴリズムにおいて，Q を点 q と見なせば，$O(n_P)$の最適手間の $P \ni q$ の判定アルゴリズムが得られ，定理２の系を証明できる．

第２，３章の２つのアルゴリズムは，closed_line_not_intersect と not_intersect の違いを除いて殆ど同じである．３章では，窓のある単純多角形と星状の窓のある星状多角形の場合は手間が線形であることの証明が複雑な多角形の三角形分割手段[1]を用いず，証明が容易な定理４[3]を用いて交差判定が $O(n_P+n_Q)$の最適手間でできることを示した．

頂点数の少ない星状多角形の少数の組み合わせで表現されたロボットに回転運動を許す場合の経路探索問題に対して，本アルゴリズムを用いれば有効である可能性がある．この場合，任意の点位置から複数の障害物に関する可視多角形を高速に作成するために障害物に対し適当な前処理を行う必要があるので，この前処理を開発することが今後の課題となる．

また，板取問題では，部材が穴を持つ場合，穴の中に他の部材を配置することが許される場合がある．そのような場合に本アルゴリズムが適用できる可能性がある．

5．終わりに

多角形の三角形分割手段を用いれば，窓のある単純多角形 P,Q 間の交差判定が $O(n_P+n_Q)$の最適手間で可能であることを示し，さらに，多角形の三角形分割手段を用いなくとも，窓のある単純多角形 P と星状の窓のある星状多角形 Q の交差判定が $O(n_P+n_Q)$の最適手間で可能であることできることを示した．

[1] B. Chazelle, Triangulating a simple polygon in linear time, Discrete Computational Geometry, 6:485-524, 1991

[2]F.P.プレパレータ，M.I.シェーモス，計算幾何学入門，総研出版，pp47, 1992

[3] D.T. Lee, Visibility of a simple polygon, Computer Vision, Graphics and Image Processing, vol.22, pp207-221, 1983