二つの凸多角形の重なりを求める最適な並列アルゴリズム

5C−2

陳慰† 中野浩嗣† 増澤利光†† 辻野嘉宏† 都倉信樹†

†大阪大学基礎工学部情報工学科 ††大阪大学情報処理教育センター

1．まえがき

重なり問題とは，図形が与えられた時，互いに重なる(交わる，共通部分を持つ)図形が存在するかどうかを判定したり，重なりを列挙したりする問題である．重なり問題は計算幾何学における最も基本的な問題であるばかりでなく，地理情報処理，VLSIのCAD,コンピュータ・グラフィックスの分野でも重要な問題である．二つの凸多角形の重なり(共通部分)を求める問題は重なり問題の中でも基本的な問題の一つである．

従来の結果によると， 凸n角形Pと凸m角形Qとの共通部分を求める逐次アルゴリズムの時間計算量は$\Theta(n+m)$である[4]．さらに，CREW-PRAM（複数のプロセッサが１つの共有メモリに接続しており各プロセッサは任意のメモリセルにアクセスできる．但し，同時読み出し可,同時書き込み不可)の上でこの問題を解く,時間計算量$O(\log(n+m))$,プロセッサ数$O(n+m)$の並列アルゴリズムも提案されている[3]． 同問題に対して， 本稿では，CREW-PRAMの上で，計算時間$O(\log(n+m))$，プロセッサ数$O((n+m)/\log(n+m))$の並列アルゴリズムを提案する．この並列アルゴリズムは，計算時間とプロセッサ数の積が，既知の最高速の逐次アルゴリズムの計算時間とオーダ的に一致するので，最適な加速(optimal speed-up)[2]である．

2．アルゴリズム

2.1 逐次アルゴリズム

本稿で提案するアルゴリズムは，文献(4)の逐次アルゴリズムを並列化したものである． まず，文献(4)の逐次アルゴリズムを簡単に紹介する．

一般性を失うことなく，入力として与えられた２つの凸多角形P，Qの頂点のx座標は異なると仮定する．PとQの各頂点を通り，y軸に平行な直線で平面をm＋n＋１個の領域（スラブという）に分割する．各スラブと凸多角形P，Qそれぞれとの共通部分はともに台形（縮退して三角形のときもある）である（図1）．この二つの台形の共通部分は高々６頂点の凸多角形である．その内の高々２点がスラブの１つの垂直線上に，高々２点がもう一方のスラブの垂直線上に，高々２点がスラブ内にある．これらの点を各スラブごとに求める．アルゴリズムの概略は以下のようである．

（１）凸n角形Pと凸m角形Qの各頂点を通り，y軸に平行な直線で平面をm＋n－１個のスラブに分割する（右端と左端の無限領域は無視する）．

（２）左端のスラブから順に各スラブ内でのPとQの共通部分を求め，順に合併していく．

2.2 CREW-PRAM上の並列アルゴリズム

本節では，CREW-PRAMの上で，二つ凸多角形の重なりを求める最適並列アルゴリズムを提案する．

［入力］凸n角形Pと凸m角形Qのそれぞれの頂点を時計回りの順に列挙した頂点の列が配列に与えられる．

［出力］PとQの共有部分P∩Q（凸k角形とする）の頂点を時計回りの順に列挙した頂点の列を配列に求める．

［方法］（説明： 前節で述べたアイデアを並列化する．ただし，スラブはm＋n－１個あるので，各スラブに１個のプロセッサを割り当てると,$O(n+m)$個のプロセッサが必要となる.ここでは,連続した$\log(n+m)$個のスラブに１個のプロセッサを割当てることにより，プロセッサ数を$O((n+m)/\log(n+m))$にし，最適加速の並列アルゴリズムを得ている．）

［ステップ１］スラブを構成する．

(1) PとQそれぞれに対して，最も左の頂点及び最も右の頂点を求める．

(2) Pを，最も左の頂点及び最も右の頂点によって，上側境界u_Pと下側境界l_Pに分ける．Qについても同様に上側境界u_Qと下側境界l_Qに分ける． ただし，u_P，l_P，u_Q，l_Qはそれぞれ頂点の系列（配列である）で，x座標の昇順に並んでいるものとする（下側境界を逆時計回り順にする）．

(3) u_P，l_P，u_Q，l_Qをマージし，P，Qの全ての頂点をそのx座標の昇順に並びかえる．このとき，この頂点系列の連続する２頂点が一つのスラブを定める．

［ステップ2］各スラブ内でP∩Qを求める．

（説明：各スラブ内のP∩Q（高々６頂点の凸多角形である）を求めるために，まず，そのスラブの垂直線とu_P，l_P，u_Q，l_Qそれぞれとの交点を求める．垂直線（aとする）上には，その垂直線を定めた頂点（例えば，u_Pの頂点A（図2））があり，これは(3)で求めている．一方，aとl_Pとの交点A1は，aの左側（右側）でaに最も近いl_Pの点B（C）を求め，線分BCとaとの交点として求める（図2）．aとu_Q，l_Qそれぞれとの交

An Optimal Parallel Algorithm for Computing Two Convex Polygons
Wei CHEN†,Koji NAKANO†,Toshimitsu MASUZAWA†,
Yoshihiro TSUJINO†,Nobuki TOKURA†
†Faculty of Engineering Science,
Osaka University
††Education Center for Information Processing,
Osaka University

点も同様に求める．u_P以外の頂点で定められるスラブの垂直線についても同様である．）

(4) P，Qの各頂点Aについて，Aの左側（右側）でAに最も近いu_P，l_P，u_Q，l_Qの頂点を以下のように求める．ただし，簡単のため，Aの左側でAに最も近いu_Pの頂点を求める方法についてのみ説明する（他の場合も同様である）：(3)で得られたP，Qのすべての頂点からなる系列の中で，i番目の頂点をAi(1≦i≦m+n)とする．各iについて，max{Ajの座標｜Ajはu_Pの頂点，かつ，j≦i}（ただし，大小関係は座標の辞書式順序による）を求めれば，Aiの左側でAiに最も近いu_Pの頂点が求まる．

(5) log(n＋m)個のスラブごとに一個のプロセッサを割り当てて，このlog(n＋m)個のスラブ中のP∩Qの部分上側境界と部分下境界を求める．

［ステップ３］P∩Qに合併する

(6) ステップ２で求められた(n＋m－1)／log(n＋m)個のP∩Qの部分上側境界と部分下側境界を連結してP∩Qの上側境界と下側境界（頂点のリスト）を得る．さらに，上側境界と下側境界を連結してP∩Qを得る．

2.3 アルゴリズムの評価

(1)は，x座標の最小値，最大値を求めればよいので，O((n＋m)／log(n＋m))プロセッサ，O(log(n＋m))時間で実現できる[2]．

(2)では，PとQの各頂点が最も右及び最も左の頂点のインデックスによってO(1)時間で自分のインデックスを修正する．従って，log(n＋m)個の頂点ごとに一個のプロセッサを割り当てれば，(2)はO((n＋m)／log(n＋m))プロセッサ，O(log(n＋m))時間で実現できる．

(3)は，　ソートされた２つの系列のマージなので，O((n＋m)／log(n＋m))プロセッサ，O(log(n＋m))時間で実現できる[1], [6]．

(4)は系列の接頭部最大値（prefix maxima)を求めればよいので，O((n＋m)／log(n＋m))プロセッサ，O(log(n＋m))時間で実現できる[2]．

(5)では，一個のスラブの垂直線とP，Qとの交点は，(4)の結果によって，O(1)時間で求められる．従って，このスラブのP∩Qの共有部分がO(1)時間で求められる．log(n＋m)個のスラブごとに一個のプロセッサを割当ると，O((n＋m)／log(n＋m))プロセッサ，O(log(n＋m))時間で終わる．

(6)では，まず，頂点系列をポインタで連結する．これは，O((n＋m)／log(n＋m))プロセッサ，O(1)時間で実現できる．この頂点系列を配列に格納するためには，リストランキングを行い，各頂点について系列中での順番を求めればよい．これは，O(n＋m)／log(n＋m))プロセッサ，O(log(n＋m))時間で実現できる[5]．

３．むすび

CREW-PRAMの上で，凸n角形と凸m角形の重なりを求める最適な並列アルゴリズムを提案した．

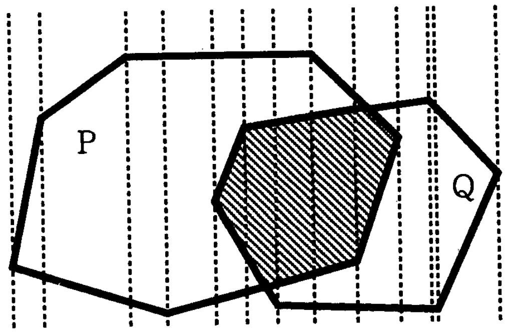

図1　二つの凸多角形の共通部分

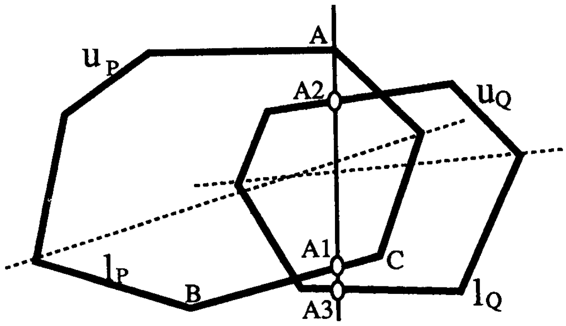

図２　Aを通る境界線とP,Qとの交点

参考文献

(1) A.Borodin and J.E.Hopcroft:" Routing,merging, and sorting on parallel models of computation", Journal of Computer and System Sciences , 30,pp. 130-145,1985.

(2) A.Gibbons and W.Rytter:" Efficient parallel algorithms",Cambridge University Press,p.1,pp.13-15,pp.1988.

(3) I.Stojmenovic:" Computational geometry on a hypercube",Proc.ICPP,Vol.3,pp.100-103,1988.

(4) M.I.Shamos and D.Hoey:"Geometric intersection problems",Proc.17th FOCS,pp.208-215,1976.

(5) R.J.Anderson and G.L.Miller:" Deterministic parallel list ranking",AWOC(LNCS 319),pp.81-90, 1988.

(6) S.G.Aki:"The design and analysis of parallel algorithms",Prentice Hall,1990 .