

## 球面分割グラフにおける点位置決定アルゴリズムと 凸多面体の支持点計算への応用

仁尾 都  
明星大学経済学部  
nio@mi.meisei-u.ac.jp

球面分割グラフ位置決定問題のために、メルカトル写像を用いることを特徴とする、手間が  $O(\log n)$  の最適アルゴリズムを提案する。さらに凸多面体の支持点計算問題のためのロバストでかつ手間が  $O(\log n)$  の最適アルゴリズムを応用例として示す。

### An algorithm for the point location problem on a spherical geodesic line graph and its application for computing a supporting point of a polytope

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

Un optimal algorithm for the point location problem on a spherical geodesic line graph using Mercator mapping and a robust and optimal algorithm for computing the supporting point of a polytope are proposed.

#### 1. はじめに

本論文は枝が測地線よりなる球面分割グラフ  $G$  と質問点  $q$  が与えられたとき、 $q$  が属する  $G$  の面を決定する点位置決定問題、および質問ベクトル  $r$  と凸多面体  $P$  が与えられた時、 $r$  と  $P$  の頂点の位置ベクトルとの内積が最大となる  $P$  の頂点を求める支持点計算問題に関する。 $G$  の例として、球面上の母点に対する球面ボロノイ図が挙げられる。

2次元直交座標系上の点位置決定問題には多くの研究例がある。平面分割グラフの頂点数を  $n$  とすれば、必要メモリが  $O(n)$ 、前処理手間が  $O(n \log n)$ 、決定手間が  $O(\log n)$  の最適アルゴリズム [1,2] などが存在する。

今回、球面上の  $n$  個の頂点を持ち、測地線を枝とする球面分割グラフに関する点位置決定問題に対し、メルカトル写像の利用を特徴とし、計算手間が  $O(\log n)$  となる最適アルゴリズム (以降、メルカトル写像法と呼ぶ) を提案する。さらにメルカトル写像法の応用として、質問ベ

クトル  $r$  に対する頂点数が  $n_p$  の凸多面体  $P$  の支持点計算手間が  $O(\log n_p)$  となる最適アルゴリズムを示す。

支持点計算は凸多面体に関する計算幾何学では重要な要素技術となっている。例えば凸多面体間交差判定手法として有名な GJK 法 [3] は、収束計算時に支持点計算を繰り返し実行する。Dobkin らの Hierarchical representation [4] を用いて、 $P$  を前処理すれば凸多面体の支持点の計算の手間は理論的には  $O(\log n_p)$  となることが知られているが、残念ながら Dobkin らの技法はロバストでないといわれている [6,7]。この技術を GJK 法の支持点計算に用いた研究例 [5] もあるが、ロバスト性が劣るアルゴリズムになることは避け得ない。そのため GJK 法では一般に手間が  $O(n_p)$  の支持点計算アルゴリズムが用いられている。

本論文で提案するメルカトール写像法を利用した支持点計算アルゴリズムは、前処理を必要とはするものの、手間が  $O(\log n)$  の性能を持ちながら、しかもロバストであるため、GJK 法や貫通距離計算アルゴリズム [8] などの高速化に活用できる。

## 2. 球面測地線グラフに関する点位置決定問題

### 2. 1 用語の定義

定義 1. 球面測地線グラフ  $G$ : 単位球面  $S^2$  を測地線で 2 つ以上の領域に分割するグラフ。

定義 2. メルカトール写像  $m: S^2$  上の点  $v$  の極座標を  $(1, \phi_v, \theta_v)$  とするとき、 $v$  を  $(\phi_v, \theta_v) \in R$  ( $R = \{(\phi, \theta) | 0 \leq \phi \leq 2\pi, 0 \leq \theta \leq \pi\}$ ) に対応させる写像。

$m$  の写像は  $\phi$  方向には  $2\pi$  を法とする。すなわち、 $R$  の左端の垂直直線線分の点  $(0, \theta)$  とそれに対応した右端の垂直直線線分の点  $(2\pi, \theta)$  は同一点とみなす。また、 $R$  の上端の水平直線線分 (始点  $(0, \pi)$ , 終点  $(2\pi, \pi)$ ) の上の点はすべて同一点であり、また下端の水平直線線分 (始点  $(0, 0)$ , 終点  $(2\pi, 0)$ ) の上の点はすべて同一点であるとみなす。すると、地球上の図形をメルカトール地図に 1 対 1 に写像するのと同様に、 $m$  は  $S^2$  と  $R$  の間の 1 対 1 対応になる。

### 2. 2 球面測地線グラフに関する点位置決定アルゴリズム

[補題]  $S^2$  上の測地線  $L$  の写像  $m(L)$  は  $\phi$  軸に関し単調である。

[証明] 始点  $(1, \phi_s, \theta_s)$  と終点  $(1, \phi_e, \theta_e)$  を結ぶ  $S^2$  上の測地線上の点  $(1, \phi, \theta)$  の写像  $m(1, \phi, \theta)$  の座標  $(\phi, \theta)$  との間には、 $0 \leq \phi \leq 2\pi$  の範囲で下記の関係が成り立つ。

$$\theta = \tan^{-1}(\tan \theta_e \tan \theta_s \sin(\phi_e - \phi_s) / (\tan \theta_e \sin(\phi_e - \phi) + \tan \theta_s \sin(\phi - \phi_s))) \cdots (1)$$

$0 \leq \phi \leq 2\pi$  の範囲で  $\sin \phi$  は一価関数であり、 $\tan^{-1} \phi$  も一価関数であるので、(1) は  $\phi$  の一価関数である。■

[定理 1] 質問点  $q \in S^2$  の  $S^2$  上の測地線よりなる分割グラフ  $G$  における点位置決定問題は、 $n$  を  $G$  の頂点数とすれば、手間が  $O(n \log n)$ 、必要メモリが  $O(n)$  の前処理を行い、かつ質問処理が  $O(\log n)$  で最適に解決可能である。

[証明] 平面曲線グラフ  $m(G)$  の枝は  $\phi$ - $\theta$  座標系の領域  $R$  を分割し, 補題より  $\phi$  軸に関し単調であり, かつ任意の 2 つの枝は頂点以外では交差しないのでスラブ法を利用する. 点位置決定アルゴリズムが適用できる. 従って, Sarnak-Tarjan の点位置決定アルゴリズム [1] を質問点  $m(q)$  とグラフ  $m(G)$  に適用すれば, 必要メモリが  $O(n)$  で, 前処理が  $O(n \log n)$  で, かつ質問処理が  $O(\log n)$  で最適に  $m(G)$  の面を探索できるので, この面に対応する  $G$  の面を質問に対する解とすればよい. ■

$S^2$  上の測地線グラフに関する点位置決定のための概略手順は下記のように Sarnak-Tarjan の点位置決定アルゴリズム [1] に沿う.

[ $S^2$  上の測地線グラフ  $G$  についての点位置決定アルゴリズム]

入力:  $S^2$  上の  $n$  個の頂点を持つ測地線グラフ  $G$ , 質問点  $q \in S^2$ , 出力:  $q$  を含む  $G$  の面

(1) 質問応答用のパーシステントなデータ構造  $M$  を前処理で作成する.

- (1.1)  $m(G)$  の頂点の  $\phi$  座標をソートし, 各スラブ  $s_i (1 \leq i \leq n+1)$  の区間  $[\phi_i, \phi_{i+1}]$  を決定.
- (1.2)  $s_i$  毎に  $[\phi_i, \phi_{i+1}]$  の間に存在する  $m(G)$  の枝  $e_j$  と  $\phi = \phi_i$  との交点の  $\theta$  座標値  $\theta_{ij}$  を (1) 式で計算し,  $\theta_{ij}$  をソートすることによりスラブ毎の枝による分割領域を決定.
- (1.3) 各スラブ, 及びスラブ毎の分割領域を Sarnak-Tarjan の点位置決定アルゴリズムが指定するパーシステントなデータ構造  $M$  に変換する.

(2)  $m(q) = (\phi_q, \theta_q)$  を質問点とする  $M$  についての点位置決定問い合わせに対する処理を行う.

- (2.1)  $\phi_q$  を含む  $s_i$  を 2 分法で決定.
- (2.2)  $s_i$  内の  $m(q)$  を含む分割領域を  $M$  の 2 分木に従って根から検索して決定する. この際,  $M$  のノードが管理する枝の上方に  $m(q)$  があることの判定は (1) 式の  $\phi$  に  $\phi_q$  を代入して得られる  $\theta$  より  $\theta_q$  が大きい事により行う.
- (2.3) 最終的に得られた  $s_i$  の分割領域に対応する  $G$  の面を出力する.

### 3. 支持点計算問題

ここで扱う 3 次元ベクトル  $v$  はすべて極座標で成分表示された単位ベクトルとし,  $(\phi, \theta)$  ( $\phi, \theta$  は  $v$  の偏角, 天頂角.  $0 \leq \phi \leq 2\pi, 0 \leq \theta \leq \pi$ .) で表すとする.

#### 3. 1 用語の定義

定義 3.  $P$ : 頂点数を  $n_p$  とする凸多面体.

定義 4. ベクトル  $r$  に対する  $P$  の支持点:

$r$  と点位置ベクトルの内積が最大となる  $P$  の頂点(図 1).

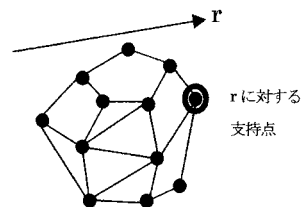


図 1 ベクトル  $r$  に対する凸多面体  $P$  の支持点

定義 5.  $P$  の頂点  $v_k$  の支持ベクトル : 支持点が  $v_k$  となるようなベクトル .

定義 6.  $P$  の頂点  $v_k$  の支持ベクトル集合  $nv_k$  : 頂点  $v_k$  の支持ベクトルの集合(図 2).

定義 7.  $P$  の面  $f_i$  の法線ベクトル  $nf_i$  : 面  $f_i$  の法線ベクトル.

定義 8.  $P$  の辺  $e_{i,j}$  の法線ベクトル集合  $ne_{i,j}$  :  $\{(1-t)nf_i+tnf_j \mid f_i \text{ と } f_j \text{ は辺 } e_{i,j} \text{ を共有, } 0 \leq t \leq 1\}$  (図 2).

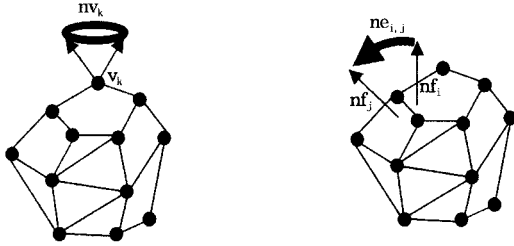


図 2. 頂点法線ベクトル集合  $nv_k$  と辺ベクトル集合  $ne_{i,j}$

定義 9. ガウス写像  $g$  : ベクトル  $(\phi, \theta)$  を  $S^2$  の極座標点  $(1, \phi, \theta)$  に対応させる関数

定義 10.  $P$  の法線ベクトルの  $S^2$  へのガウス写像  $g(P) : \{g(nf_i), g(ne_{i,j}), g(nv_k)\}$

定義 11.  $P$  の法線ベクトル地図  $NVM : m(g(P))$ ,  $NVM$  は Normal Vector Map の略.

$ne_{i,j}$  の定義によれば  $g(ne_{i,j})$  は  $S^2$  の測地線となるので  $g$  は結果的に  $P$  の辺,面を  $S^2$  上の測地線グラフ  $G$  の枝と頂点に 1 対 1 に対応させ,結局は  $P$  の頂点を  $G$  の面に 1 対 1 に対応させることにもなる.  $m(g(ne_{i,j}))$  は凸多面体  $P$  上に定義されたベクトル  $ne_{i,j}$  の 2 次元  $\phi - \theta$  直交座標系への写像となり,  $R$  内のグラフの枝となる.  $P$  が 4 面体である場合の  $NVM$  の例を図 3 に示す.

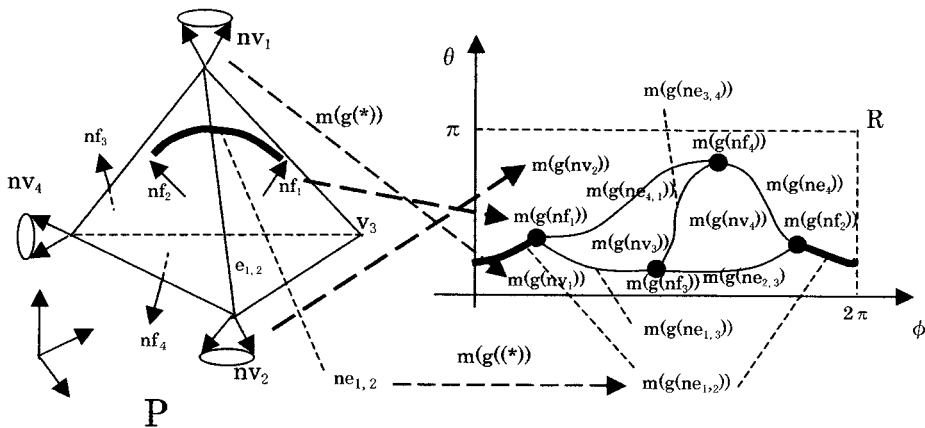


図 3. 4 面体  $P$  の  $NVM$  の例

図3では  $g$  による  $P$  の法線ベクトルの  $S^2$  への写像は省略されている。従って、補題より  $m(g(P))$  は  $\phi$  軸に関し単調である。図中、辺単位法線集合  $ne_{1,2}$  が  $R$  内の 2 本の太線で示された  $m(G)$  の単調な枝に写像している様子が示される。2 本の太線は  $\phi$  に関し  $2\pi$  を法とするので連続している。その他のすべての枝も  $\phi$  軸に単調である様子も示されている。

### 3. 2 球面点位置決定問題と支持点計算問題との関係

質問ベクトル  $r$  に対応する凸多面体  $P$  の支持点を求める問題は、 $P$  の表面点に定義された法線ベクトルの中から  $r$  に等しい法線ベクトルを探索し、得た法線ベクトルが所属する  $P$  の頂点を決定する問題となり、後者の問題は、単位球面上の質問点  $g(r)$  に対する  $g(P)$  における点位置決定問題となり、更にこの問題は第 2 章で述べたように 2 次元直交座標系における質問点  $m(g(r))$  に対する  $m(g(P))$ 、すなわち NVM における点位置決定問題となる。その為、結局定理 1 により、次の定理が得られる。

[定理 2] 与えられた質問ベクトル  $r$  に対する頂点数が  $n_p$  の凸多面体  $P$  の支持点計算問題は、前処理手間が  $O(n_p \log n_p)$  で、必要メモリが  $O(n_p)$  で、かつ支持点計算手間が  $O(\log n_p)$  である。

### 4. 結果の検討

Dobkin らの Hierarchical representation [4] は凸多面体  $P$  の独立頂点集合を抽出しそれらを削除し、残った頂点に対し凸包を作成し頂点数のより少ない凸多面体を再構築することを繰り返し、最後には単体に至る  $O(\log n_p)$  個の凸多角形系列を得るものである。これを用いれば支持点計算問題は理論的には手間が  $O(\log n_p)$  で解決可能である。しかし、実際には  $n_p$  が大きくなるにつれ、系列図形には同一平面上に殆ど存在するような 4 頂点のグループが出現する確率が高くなり、ロバストな図形計算処理が困難になることが指摘されている [6]。しかも、 $P$  を階層表現した場合の図形系列数  $O(\log n_p)$  の係数は大きく、したがって前処理の手間と必要メモリ容量、および計算処理時の手間は実用上大きい。特に、系列図形は 3 次元図形であるからそのメモリ負担は大きい。これらの理由から Dobkin らの理論をロバストなプログラムとして実現することに成功した例はないといわれている [5]。

今回提案した支持点計算アルゴリズムは、3 次元支持点計算問題を 2 次元直交座標系上の点位置決定問題に変換して解き、 $P$  のトポロジーを変更しないので 4 点以上の頂点の同一平面性を判定する必要がないためロバストであり、またデータ構造は 2 次元であるのでメモリも計算量も少ない。

本方式による計算誤差は  $m(g(r))$  を含むスラブが特定された後に、そのスラブ内の  $G$  の枝と  $m(g(r))$  の上下関係を (1) 式で求める際に発生する発生する可能性が残る。このようなことが起これば本来の  $G$  の面とは異なる面、つまり、本来の  $P$  の支持点とは異なる頂点を選ぶということが起こりえる。しかし求めた頂点が支持点であるかどうかはその頂点に接続する頂点を調べれば

確認が容易にできる。たとえそれが支持点でなくとも、山登り法で連結頂点を辿って支持点を求めればよく、Dobkin の階層表現を用いた場合のような計算手順進行時の論理破綻を生じさせるような深刻な問題点を内包しない。

平面点位置決定アルゴリズムとして採用可能なものは Sarnak-Tarjan のアルゴリズムの他に、対象とするグラフの枝が単調であることを必要とするアルゴリズム(例えば鎖分解法 [2])に限定される。

凸多面体間の交差判定アルゴリズムとして代表的な GJK 法では、手間が  $O(n_p)$  の支持点計算を用いているためにその計算時間の大部分が支持点計算に消費され、その平均収束回数を  $w$  とすれば、平均手間は  $\theta(w n_p)$  となる。一般には  $w$  が  $n_p$  に比べて無視できるほどの小さいオーダーであるとされているようであるが、その証明の存在を筆者は知らない。メルカトール写像法を代わりに用いれば、メモリが  $O(n_p)$  で、手間が  $O(n_p \log n_p)$  の前処理が新しく必要とはなるものの、GJK 法の平均手間を  $\theta(w \log n_p)$  とすることが期待できる。

凸多角形を対象にした種々の計算幾何学問題においては頂点は互いに一次的に連結するため、2 分法が強力な道具になる。また、凸多角形の少数の頂点が初期に任意に与えられれば、凸多角形を詳細化してゆく際の足がかりとして、残った頂点の中から代表的な頂点を取り出すことがしばしば必要になるが、この様な場合でも、代表頂点として 2 分点を採用することは極容易である。言葉を変えれば、凸多角形は明示的に階層表現しなくとも、いつでも簡単に 2 分木階層表現化できる。

一方、凸多面体  $P$  では頂点が 2 次的に連結しているため、2 分法を容易に利用できない。本アルゴリズムは、 $P$  の代表的な頂点を取り出す道具として利用可能である。この際、代表点は支持点を用いることになる。つまり、本アルゴリズムは階層表現化の道具として利用できる。ただし、パーステントなデータ構造  $M$  を前処理で作成して置く必要があり、それを用いて表現された階層は平衡化されておらず、その深さは  $O(n_p)$  となる。しかし、一般の凸多角形では  $S^2$  上の頂点が一様に分布することが期待できるので、平均的な深さは  $\theta(\log n_p)$  となるのではないか。

支持点を凸多面体の代表点として利用している例として、Bergen の凸多角形間貫通距離計算 [8] がある。2 つの凸多面体の支持点を計算することで、それらの Minkowski 和の支持点を求めることができるという性質を利用している。このため、ただ単に支持点を代表点に使っているという単純な例とは必ずしもいえないが、このアルゴリズムにもメルカトール写像法を利用すれば高速化できる。このように、メルカトール写像法は凸多面体計算幾何学の要素技術として活用できる。

3 次元  $n$  連立線形計画法においても、一旦前処理を施せば、以降、線形目的関数を任意に変更しても、メルカトール写像法を利用すれば、その都度  $O(\log n)$  でロバストに最適解を求めることができる。したがって、線形計画法における感度分析の高速化に利用できる。

## 5. 終わりに

$n$  個の測地線を枝とする球面分割グラフに関する位置決定問題のための決定手間が  $O(\log n)$  の最適アルゴリズム (メルカトル写像法) を提案した. さらに応用例として  $n$  頂点の凸多面体の支持点計算問題のための計算手間が  $O(\log n)$  のロバストな最適アルゴリズムを示した. これを利用すれば, 凸多角形間の交差判定や貫通距離計算の高速化が図れる.

- [1] N. Sarnak and R. E. Tarjan, Planar point location using persistent search trees, Comm. ACM. 29(1986),669-679
- [2] E. Edelsbrunner, L. J. Guibas and J. Stolfi, Optimal point location in a monotone subdivision, SIAM J. Comput., 15(2), 317-340(1986)
- [3] E. G. Gilbert, D. W. Johnson and S. S. Keerthi, Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space, IEEE Journal of Robotics and Automation, Vol.4, No.2, 193-203, April (1988)
- [4] D. Dobkin, D. Kirkpatrick, Determining the separation of preprocessed polyhedra – a unified approach, Proc. of ICLALP '90,400-413,1990, Lecture Notes in Computer Science 443.
- [5] M. C. Lin and S. Gottschalk, Collision detection between geometric models: a survey, (1998)
- [6] G. Barequet, B. Chazelle, L. J. Guibas, Joseph S. B. Mitchell and Ayellet Tal, BOXTREE: A Hierarchical Representation for Surfaces in 3D, Eurographics Association, 1996, Volume 15, (1996), Number 3
- [7] Chung Tat Leung, Kelvin, An Efficient collision Algorithm for Polytopes in Virtual Environments, M. Phil THESIS(1996), The University of Hong Kong
- [8] G. Bergen, Proximity queries and penetration depth computation on 3d game objects, Game Developers Conference, 2001
- [8] Edelsbrunner, L.J. Guibas and J. Stolfi, The University of Hong Kong