3次元空間における有限個の点の 高速 Convex Hull アルゴリズム*

古 川 進**

An Efficient Convex Hull Algorithm
for Finite Point Sets in 3-D Space

Susumu FURUKAWA

An efficient convex hull algorithm is presented for finite sets of n points in 3-D space. The method is based on the idea of determining the outline loop of a convex polyhedron viewing from a point and connecting the point with each vertex of the outline loop. A convex hull polyhedron can be constructed

with $O(n)$ operations, i.e., the processing time is proportional to the number of given points. The algorithm was programmed in FORTRAN language and various computational results were obtained. They guarantee the justice of the theoretical considerations.

Key words: algorithm, spatial set of points, convex hull, computational geometry, geometric models, CAD

1. 緒 言

設計の初期構想の段階から生産に至るまでの多くの過程において、有効に計算機を利用し、作業の能率を高めようとする考えは総合設計生産 (CAD/CAM) システムといわれ、その実現を目指して、多くの努力が払われている。この総合設計生産システムにおいては、3次元形状モデル (あるいは幾何モデル) がつねにその中心に位置するものと考えられている。この形状モデルに個々の技術が統合されることにより、有効な CAD/CAM システムが構築されると考えられているのである¹⁾。それ故、形状モデリングに関しては比較的古くから研究が開始され、多くの努力がなされてきた。これらの努力の結果として、最近においては、幾何モデル生成処理システムとして構築されたものも種々報告されている²⁾³⁾。

著者らは、効率の良い幾何モデル生成処理システムを構築するための基礎として、先に木構造を利用した多面体の記述法に関する提案を行った⁴⁾。この方法においては以下に述べるような木構造によって多面体が記述される。すなわち、まず原立体 (記述の対象となっている多面体) の convex hull 多面体をその根に対応させる。次にその子供には、根に対応する凸多面体と原立体との差の部分の convex hull 多面体を配置するのである。以下同様な操作を繰り返すことによって、木構造の葉の部分に至るまでがすべて凸多面体に対応するようにする。

このような多面体の記述法における特徴としては、

- (1) 任意の節における凸多面体はその子供を必ず内部に含む。
- (2) 根を含めて、奇数次の節に対応する凸多面体は (近似的に) 多面体の実際に存在する部分を表し、また偶数次の凸多面体はその上位に位置する奇数次多面体より取り去るべき空間を意味している。

このような多面体の記述法を採用することによってどのような利点が得られるかについては現在検討中であるが、隠線消去処理⁵⁾、形状の部分的な変形操作⁶⁾などは比較的効率良く行い得るものと思われる。

ところで、実際の設計作業を進めて行く際には、始めに生成された立体の形状モデルは、検討され、追加、修正、削除等の処理が施され、最終的には当初の予想とは著しく異なるものになって行く。このとき計算機内に構成された形状モデルも、予測出来ない種々の変化を受け、その都度その構造が変化する。著者らの多面体モデルの場合には、木構造の各節に位置する凸多面体に変化を受け、したがって、convex hull 図形を作り直す作業が頻繁に行われるはずである。空間内に与えられた点の convex hull 図形を高速に求めるアルゴリズムがぜひとも必要となるゆえんである。

本論文においては、上述のような背景を前提として、3次元空間内に与えられた有限個の点の convex hull アルゴリズムの新しい方法について提案する。本アルゴリズムは各点に対しては (平均的な意味で) それぞれ一定の処理操作を行うだけで作業を終了することが可能であり、したがって演算に要する時間は与えられた点の数に比例する程度のものである。

* 原稿受付 昭和58年10月17日。昭和58年度精機学会春季大会学術講演会 (昭和58年3月27日) および昭和58年度精機学会秋季大会学術講演会 (昭和58年10月15日) にて発表

** 正 会 員 山梨大学工学部 (甲府市武田 4-3-11)

本論文においては、さらに実際にプログラムを作成し、種々の例について処理時間を測定したところ、理論的考察と一致する結果も得られたので、合わせて報告する。

2. Convex Hull アルゴリズムに関する 従来の研究

convex hull を求める問題は単に著者らの幾何モデルへの応用のみならず、計算幾何学、パターン認識⁷⁾、線形計画法などの分野においても基本的な課題の1つであり、したがって、従来より関心が持たれてきた。

convex hull アルゴリズムに対する従来のアプローチはおもに2次元および3次元空間の点に対するものである。

2次元空間に与えられた点のconvex hull アルゴリズムはGraham⁸⁾およびJarvis⁹⁾によって論じられている。

3次元空間の場合に対しては、Appel ら¹⁰⁾およびPreparata ら¹¹⁾¹²⁾によって議論されている。Appel らの方法は原理的には凹部を見つけて、その部分をconvex に直すという操作を繰り返すものである。この方法においては、対象となる点はある多面体の頂点であり、かつその多面体に対して、すでにそのデータ構造が作られていることが前提条件となっている。演算時間の評価式については何も言及されていないが、1つの目安として、頂点数15~20程度の場合に対してIBM 360/168による計算例が報告されている。このときの演算時間は0.2~0.8s程度である。

Preparata らは最初、通常考えられる二分法の考え方を応用して、演算時間が $n \log_2 n$ に比例する程度で済む方法を提案した¹¹⁾が、その後近似的なconvex hull 図形を得るだけで良い場合には、演算時間が

$$T = O(n + k^2 \log_2 k)$$

で済む方法を提案した¹²⁾。ここで T, n, k はそれぞれ、演算時間、与えられる点の数および近似の度合を示すパラメータである。上式において、パラメータをたとえば $k = \sqrt{n}$ とおくと、 n が小さい範囲ではほぼ $O(n)$ と考えても良いものと思われる。しかし、この方法を利用して厳密なconvex hull 図形を求めようとする場合には、 $k \geq n$ とおくことが必要であり、通常の方法に比較してより多くの処理時間が要求される。

3. Convex Hull アルゴリズム

3.1 凸多面体の面の可視性テスト

図1に示されるような凸多面体 P のどの面がその外部の一点 p より見えるかについて考察する。いま多面体 P の各面 S_i ($i=1, 2, \dots, m$)に対してすべて外向きの法線

Fig. 1 Test of face visibility

Fig. 2 The convex hull of a convex polyhedron and a point
—; outline loop

ベクトル n_i が与えられているものとする。ここで m は多面体 P の面の数を表すものとする。面 S_i 上の一点 r_i より点 p に向かうベクトルを v_i で表すことにすると、これら2つのベクトルの間に $n_i \cdot v_i > 0$ なる関係式が成り立つ場合に面 S_i は点 p より見えるといえる。

視点 p より見える面を以下においては可視面、そうでない面を不可視面と呼ぶことにする。また、可視面同志によって共有される稜線を可視内線、不可視面同志によって共有される稜線を不可視内線、可視面と不可視面によって共有される稜線を輪郭線と呼ぶことにする。

3.2 凸多面体とその外部の一点とのConvex Hull

図2に示されるように、凸多面体 P とその外部の一点 p が与えられたとき、これらのconvex hull 図形は次の2つの手続きを順次実行することによって求めることができる。

- ① まず前節において述べられた方法によって、 P のすべての可視面を抽出する。次にこれらの可視面を利用して多面体 P の輪郭線を求める。図2の例においては、面 S_i の稜線(太線で示されている)が求める輪郭線となる。
- ② 上述の手続きによって求められた輪郭線の構成頂点 V_j ($j=1, 2, \dots, l$)と点 p を結ぶ線分を新しく作られるべきconvex hull 多面体 P' の稜線として生成する。ここで l は輪郭線ループの構成稜線(あるいは頂点)の数を表すものとする。この作業に伴って、新たに発生する面も生成することが必要であるが、この操作は簡単である。

なお、新稜線、面の生成に伴って、可視内線および可

視面はすべて不要となる．もちろん，可視内線同志に共有されている頂点も不要となる．

アルゴリズムの正当性の証明

凸多面体 P の可視面の1つ S_i 上の（輪郭線上の点を除く）点 p' を考える．いま点 p および p' を通る直線 R を考えると，線 R は p, p' のほかに P のいずれか1つの不可視面との交点 p'' を持つ．ところで，点 p は新しく作られる凸多面体 P' 上の一点となるはずであるから，線 R と P' の各面との交点は (P' は凸多面体であるから) p 以外には1か所だけでなければならない．これは明らかに，点 p'' であるはずである．したがって，点 p' は多面体 P' の内部の点になるといえる．

同様な議論は，線分 $\overline{pp'}$ 上の (p を除く) すべての点に対して成立する．したがって，線分 $\overline{pp'}$ 上の点はすべて凸多面体 P' の内部にあるといえる．

上述の議論は可視面上の任意の点に対して成り立つ．それ故，すべての可視面および可視内線は convex hull 多面体 P' の内部に入ることになる．

3.3 与えられた点がすべて Convex Hull 多面体の境界面上に存在する場合

与えられた点のすべてが，convex hull 図形の境界面上に位置することがあらかじめわかっている場合には，前節の convex hull アルゴリズムを利用することによってすべての点に対する convex hull 図形を求めることができる．

まず，与えられた点の中より，同一平面上にはない任意の4点を選び，4面体 P_0 を生成する．次に残りの点の中から任意の点 p_1 を選び，多面体 P_0 および頂点 p_1 に前節の結果を適用して新たな凸多面体 P_1 を作る．これは一般には，点 p_i ($i=1, 2, \dots, n-4$) および多面体 P_{i-1} より新たな凸多面体 P_i を生成する操作であるといえる．漸化式で表現すると，

$$P_i = \Phi(p_i, P_{i-1}) \quad (i=1, 2, \dots, n-4) \quad (1)$$

と書ける．ここで Φ は点 p_i および多面体 P_{i-1} より新たな convex hull 多面体を生成することを表す関数であるとする．また n は与えられた点の数である． P_{n-4} が求める convex hull 多面体となる．

3.4 点の角度分布表

式(1)における点列 p_i がその位置に無関係に現れる場合には，その都度 p_i のすべての構成面の可視性を調べることが必要である．本節においてはこのような事態を避けるための方法について考察する．

図3に示されるように，点 p_{i-1} の付近に点 p_i が存在する場合について考える．いま点 p_{i-1} と多面体 P_{i-2} の convex hull 図形を生成する際に求められた輪郭線を f_{i-1} で表すと，点 p_i は点 p_{i-1} の付近に位置しているの

Fig. 3 The case where the places of two points p_{i-1} and p_i are near
—; outline loop f_{i-1}

Fig. 4 A distribution diagram of points on (θ, φ) plane

であるから，輪郭線 f_{i-1} の一部がやはり新たな輪郭線の構成線となる可能性が高い．

したがって，この場合には点 p_i より見たときの新たな輪郭線 f_i は容易に求められる．すなわち，求められた輪郭線分に接続されている稜線をその可視性を調べながらたどって行くことにより，輪郭線のループが得られるのである．

式(1)において，点 p_{i-1} に引き続いて現れる点 p_i が点 p_{i-1} の近くであるようにするためには，あらかじめ点の分布状態についてある程度調べられていることが必要である．このために，各頂点のおおまかな位置関係を示す点の分布図を作成する．

具体的には，まず4面体 P_0 の図心の位置に原点を移し，その後，各点の経度，緯度をそれぞれ計算する．これらの経度，緯度に対して図4に示されるようなあらかじめ用意された点の角度分布表の該当位置にその点を対応させる．この操作がすべての点に対して終了すれば，図4のような2次元の点の角度分布表が得られる．図4において， θ および φ はそれぞれ経度，緯度を表すパラメータである．パラメータ θ, φ はそれぞれ区間 $[0, 2\pi]$ および $[0, \pi]$ の間で変化する．また図4の分布表の各要素の占める領域は，パラメータの定義域をそれぞれ $\sqrt{n}$ 等分したときに得られるものとする．ただし， $\sqrt{n}$ が整数にならない場合には，これを四捨五入して整数値に直した

Fig. 5 A method of detecting a point being inside the convex polyhedron
—: outline loop f_{i-1}

値を用いることにする。

また、図4に示される点の角度分布表の作成の際に、同一の領域に属する2つ以上の点がある場合には、原点(P_0 の図心)からの距離の最大のもを最初を選び得るようにしておくものとする。

3.5 点がConvex Hull多面体の内部にも存在する場合

点 p_i が凸多面体 P_{i-1} の内部にある場合について考察する。この場合には、原理的には P_{i-1} のすべての面の可視性テストを行い、可視面が存在しないことを確かめることが必要である。しかしこの方法では凸多面体の内部に点が存在する場合にその処理効率が著しく低下することになる。

図5に示されるように、第 i ステップの処理において、点 p_i が凸多面体 P_{i-1} の内部にある場合の識別法について考察する。いま多面体 P_{i-1} を生成する際、すなわち、点 p_{i-1} の処理の際に求められた輪郭線を f_{i-1} で、また多面体 P_0 の図心を G_c で表すことにする。

点 p_i に対しては、輪郭線 f_{i-1} に囲まれた領域には可視面が存在しなかったものと仮定する。このときには、点 G_c を1つの頂点として、それと輪郭線 f_{i-1} の各頂点を結んで得られる底のない角すいの内部に点 p_i があるか否かについて調査する。

点 p_i がこの角すいの内部にある場合には点 p_i は多面体 P_{i-1} の内部にあるといえる。このときは新たに得られる多面体 P_i は、 P_{i-1} と同一のものである。

3.6 輪郭線ループの置換え

点 p_i の処理において、輪郭線ループ f_{i-1} のどの構成稜線も不可視であり、かつ、点 p_i が図5に示される角すいの外部にある場合には、前節までに述べられた方法では、その位置を確定できない。

このような場合には、以下の手順に従って、輪郭線ループ f_{i-1} を別の(仮の)輪郭線ループ f'_{i-1} で置き換えることにする。すなわち、 P_0 の図心 G_c 、点 p_{i-1} および p_i によって形成される面と、輪郭線ループ f_{i-1} との交わりがどこにあるかを調べる。求められた交点が輪郭線ループ

Fig. 6 Transposition of outline loop f_{i-1} to f'_{i-1}

プ f_{i-1} の k 番目のセグメント f_{i-1}^k 上にある場合には、輪郭線ループ f_{i-1} を稜線 f_{i-1}^k を共有する面で、かつ f_{i-1} の外部に位置する面の構成稜線 f'_{i-1} で置き換えるのである(図6)。このとき点 p_{i-1} も新たな輪郭線ループ f'_{i-1} の内部の点 p'_{i-1} に置き換えておくことが必要である。具体的には、 f_{i-1} の図心の位置を p'_{i-1} とするのが適当であろう。

上述の手続きによって、新たに得られた輪郭線ループ f'_{i-1} を利用して、再度3.4節および3.5節に述べられた方法に従って輪郭線ループ f_i を求め、あるいは点 p_i が多面体 P_{i-1} の内部にあるか否かを確認するのである。

輪郭線ループ f'_{i-1} を利用して、点 p_i の位置が確定できない場合には、さらに別の(仮の)輪郭線ループ f''_{i-1} を前と同様にして作り、それにより f_i を求め、あるいは多面体 P_{i-1} の内部であることを確定するのである。 f''_{i-1} によっても依然として、点 p_i の位置が確定できないときには、その位置が確定できるまで同様の操作を繰り返すことが必要である。

以上の考察により、3次元空間に有限個の点が与えられた場合の凸閉包は比較的効率良く求めることができると思われる。次節においては、凸閉包生成のために必要な演算時間の評価に関する考察を行う。

なお、上述の理論は平面図形の場合に対しても応用可能であり、かつ3次元空間の場合に比較して著しく容易なものである。

4. 演算時間の評価

本アルゴリズムの主要な部分は、①点の角度分布表を作る、②輪郭線ループを利用して新しい面を生成する、の2つの部分に分けて考えることができる。①の分布表の作成は各点に対して1回の処理で済むので、演算時間は与えられた点の数に比例する。

②の処理の演算時間の評価について考察する。 k 個の点が空間に一樣に分布していると仮定する。このとき、これらのすべてを内部に含む球を考えると、この球上に描かれた弧の長さは最大 $2\pi r$ であり、したがって、生成された凸多面体のある点から見た場合の輪郭線ループの

Fig. 7 The case where there exist many visible faces and visible inner points

長さ（あるいは構成稜線の数）もほぼ r に比例すると考えられる。ところで球の半径 r は、

$$r = \alpha \left(\frac{3k}{4\pi} \right)^{1/3} \quad (2)$$

で表されるから、輪郭線の構成稜線数は $k^{1/3}$ に比例する程度であるといえる。ここで α は比例定数である。

与えられた点がすべて convex hull 多面体の構成頂点であることがわかっている場合について考察する。この場合には、図 7 に示されるような輪郭線ループの内部に頂点が含まれるという状態は起こり得ない。それは、与えられた点がすべて convex hull 多面体の構成頂点であるという仮定に反するからである。したがって、この場合には非常に多くの可視面が存在することは不可能である。言い換えると可視面の数は与えられた点の数には無関係であり、ある定数を越えないはずであるといえる。したがってこの場合の輪郭線ループを求めるための手続きは（平均的な意味で）一定の処理操作で終了するといえる。

点 p_i の調査の際に、輪郭線ループの置換えが行われる場合の評価についても考察しておく必要があるであろう。引き続いて調べられる 2 点、たとえば p_{i-1} および p_i の処理を考えると、これら 2 点の位置が（点の角度分布表の上で）遠く離れている場合には、輪郭線ループの置換えが、頻繁に行われるはずであり、この場合には、置換えの回数が半径 r に比例する程度となるであろう。しかし、実際には点の角度分布表を利用する限り、引き続いて現れる 2 点の位置の差はほぼ一定であると考えても差し支えないであろう。したがって、輪郭線ループの置換えの起こる回数もある一定の平均値を持つと考えられる。

以上の議論より、与えられた点がすべて convex hull 多面体の構成頂点であることがわかっている場合には、点 p_i および多面体 P_{i-1} より多面体 P_i を生成する過程はほぼ一定の時間で行われ得るものと考えられる。したがって、 n 個の点の convex hull 多面体を生成するための処理時間 T は、

$$T = \beta n \quad (3)$$

Fig. 8 The relation between the number of points and the processing time obtained by numerical experiments

と表される。ここで β は比例定数である。

与えられた点が convex hull 多面体の内部にも有り得る場合について考察する。点の角度分布表の各小領域において、最初に選出される点は原点より最も遠い位置にあるものであるから、これらの点については、前述の点がすべて convex hull 多面体の頂点である場合とほぼ同様な議論が成り立つ。また同じ領域に属するほかの点は convex hull 図形の内部に属する場合が多いと考えられ、これらについては、3.5 節の方法によって識別される。輪郭線ループの構成稜線数はほぼ一定と考えられているので、この場合にもやはり、一定の時間で、識別が可能である。

以上の議論より、与えられた点が convex hull 多面体の内部に存在し得る場合にも、その処理時間は点の数に比例する程度であるといえる。すなわち、いずれの場合においても総演算時間 T は、

$$T = \alpha n + \beta n = (\alpha + \beta)n = An \quad (4)$$

で表される。ここで α, β はそれぞれ、点の角度分布表を作成するための一点あたりの処理時間および各段階における凸多面体の（平均的な）生成時間を表すものである。また A はそれらの和を表す比例定数である。

5. 計算機による実行結果

前節までに述べられた方法に従って、FORTRAN プログラムを作成し、マイクロコンピュータ（CPU: MC 68000, RAM: 512 KB）を利用して演算時間の測定を行った。この数値実験においては、点のデータとして、 x, y, z 座標値に対してそれぞれ区間 $[0, 1]$ において得られる擬似一様乱数が使用された。また使用計算機は CRO MEMCO CS 3 D 5 E であった。

得られた処理時間の測定結果を図 8 に示す。図 8 は前

節において示された「演算時間は与えられた点の数に比例する」という結論を裏付ける結果を示している。

図中、破線は演算時間の上限および下限を示すもので補助的に書き加えたものであるが、すべての結果は、この放射状の帯の中に位置している。演算時間が上限に近い位置にある場合は多面体を形成する頂点の数が多かったことを意味している。与えられた点の数が比較的少ない所で上限を示す直線に近い場合が多いことはこのような意味によるものである。

なお、3.4節において述べられたように、点の角度分布表を作成する際に、各小領域においては、原点（多面体 P_0 の図心）よりの距離がそれぞれ最大のものを最初を選び出すことができるような工夫を施してあるが、著者の数値実験においては、このような操作を行わなくても演算時間が与えられた点の数に比例するという結論にはかわりがなかった。ただ、この場合、すなわち各小領域からランダムに点を選び出す場合には演算時間が約2%増加することが認められた。

6. 結 言

本論文においては、新しい convex hull アルゴリズムが提案され、さらに実際の計算機による実行結果も合わせて報告されたが、その結果、「演算時間は与えられた点の数に比例する程度で済むものである」という結論が得られた。それ故、本アルゴリズムは先に著者らの提案した多面体の階層構造の生成および処理を行う際に強力な手段を提供するばかりでなく、パターン認識、計算幾何学などの分野においても寄与し得るものであると思われる。

終わりに 日ごろ 御指導賜る 東京電機大学・穂坂衛教授、東京大学・佐田登志夫教授、同・木村文彦助教授に

感謝の意を表す。また本研究に対して、種々の御援助を与えられた、明星大学・前沢成一郎教授、山梨大学・松本道男教授、著者の身辺にあって種々の討論および援助をいただいた日産自動車・石塚譲二氏（当時、山梨大学大学院生）に感謝の意を表す。

参 考 文 献

- 1) 沖野教郎：三次元形状処理技術と CAD/CAM の新展開 (1), 機械の研究, **33**, 11 (1981) 1223.
- 2) A. A. G. Requicha: Representations for Rigid Solids, Theory, Method and Systems, Computing Surveys, **12**, 4, (1980) 437.
- 3) 中川 茂, 井上高志：3次元図形処理と形状モデリング技術の調査報告書, 情報処理振興事業協会技術センター (1983).
- 4) 古川 進ほか：立体の簡略構造とその構成に関する考察, 日本機械学会講演論文集, No. 821-3 (1982) 379.
- 5) S. Furukawa et al.: Succinct Representation of 3D Object and its Application to Hidden-line Problem, *Computer Applications in Production and Engineering*, North-Holland, Amsterdam, (1983) 631-642.
- 6) 石塚譲二, 古川 進：簡略構造による立体の生成について, 昭和 57 年度 精機学会秋季大会 学術講演会講演論文集 (1982) 621.
- 7) 加藤英二ほか：多面体モデルの安定な姿勢の決定, 情報処理学会第 25 回全国大会予稿集 (1982) 897.
- 8) R. L. Graham: An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set, Information Processing Letter, **1**, (1972) 132.
- 9) R. A. Jarvis: On the Identification of the Convex Hull of a Finite Set of Points in the Plane, Information Processing Letter, **2**, (1973) 18.
- 10) A. Appel et al.: Determining the Three-dimensional Convex Hull of a Polyhedron, IBM J. Res. Develop., **11**, (1977) 590.
- 11) F. P. Preparata et al.: Convex Hulls of Finite Sets of Points in Two and Three Dimensions, Communications of the ACM, **20**, 2, (1977) 87.
- 12) J. L. Bentley et al.: Approximation Algorithms for Convex Hulls, Communications of the ACM, **25**, 1, (1982) 64.