セグメントチェイン法による自由曲面の 隠れ面消去処理*

村 松 正 司** 山 口 富 士 夫***

A Visible Surface Algorithm by Using the Segment Chain Method

Masashi MURAMATSU and Fujio YAMAGUCHI

This paper presents a fast visible surface algorithm for free form bodies. Conventional algorithms for such objects require a lot of computing time as well as much memory capacity. The proposed algorithm approximates the objects with many small triangles. It makes use of a special triangle data structure which preserves topological information between neighboring triangles and requires comparatively small memory storage. By employing the segment chain method, the algorithm exhibits much faster characteristics compared with the Watkins' algorithm.

Key words : visible surface algorithm, computer graphics

1. 緒 言

不規則な変化を持つ自由曲面の表示には隠れ面消去処理が不可欠である。この処理は自由曲面を多くの多角形（ポリゴン）で近似すると、多面体の場合と同一のプロセスで行うことができる。また、自由曲面上の点の座標値を算出するには多くの演算時間を要するので、多面体近似の方法によれば点の座標計算の重複を避けることもできて都合がよい。ところが、自由曲面を多面体で近似するには多くの面が必要となり、隠れ面判定処理に多量の時間を要することが大きな問題点となる。

本研究では、セグメントチェイン法¹⁾を利用することによりこの問題に対処した。セグメントチェイン法とはスキャンラインアルゴリズムの一種であるが、一続きのセグメントを一まとめとして扱うもので、これにより高速な処理を行うことができる。

セグメントチェイン法の利用により、自由曲面に対する高速な隠れ面消去アルゴリズムを開発した。

2. セグメントチェイン法

セグメントチェイン法について簡単に説明する。

* 原稿受付 昭和60年9月17日、昭和59年度精機学会春季大会学術講演会（昭和59年3月25日）にて発表

** 正 会 員 セイコー電子工業(株)（習志野市茜浜1-1-1）

*** 正 会 員 早稲田大学理工学部（東京都新宿区大久保3-4-1）

Watkinsの隠れ面消去アルゴリズム²⁾は、演算時間の特性が、対象シーンの複雑さではなくその可視図形の複雑さにのみ依存する。この演算時間特性の魅力は、Warnockのアルゴリズム³⁾の考え方を利用したことにより達成されている。Warnockのアルゴリズムは、画面にウィンドウを設定し、ウィンドウ内における物体の可視性をTHINKERによって判断する。ウィンドウ内が複雑で可視性の判断がTHINKERの能力を越える場合には、ウィンドウを分割し再帰的に処理を行う。この考え方は非常にエレガントであるが、ウィンドウの分割回数が多く、アルゴリズムの処理速度は高速ではない。Watkinsはウィンドウをスキャンラインまたはその一部であるサンプルスパンにとり、スキャンラインコヒーレンスを利用して分割点を予測することにより分割回数を減らし、高速なアルゴリズムとすることに成功した。

しかし、Watkinsのアルゴリズムでも多面体近似された曲面のように可視図形が複雑になると、多量の演算時間を要する。この理由は、ポリゴンとスキャンライン平面の交線であるセグメント単位に処理をするためである。つまり、セグメントが多い場合には演算時間も長くなるのである。

しかし、一続きのセグメント（これをセグメントチェインと呼ぶ）単位に処理を実行すれば、処理の高速化が期待できる（図1）。一般にはセグメントが単独で現れることは珍しい。問題はいかに高速にセグメン

Fig. 1 A segment chain

トチェーンを作成するからである。セグメントチェーンの作成は、三角形分割の利用により y 値の比較のみで簡単に達成できることがわかっている¹⁾。したがって、曲面の場合には特にセグメントチェーンの手法は有効なのである。

セグメントチェーン法は、Watkins のアルゴリズムにおいてセグメント単位で行っていた処理をセグメントチェーン単位で行うものといえる。したがって、セグメントチェーン法は、曲面を多面体で近似した場合のように、輪郭稜線に対して前方稜線の多い形状に特に有効である。

3. アルゴリズムの概要

本アルゴリズムは、大きく3つのステップに分けられる。

まず、自由曲面を多面体で近似する。この時、面を三角形にすると以下のセグメントチェーン作成が容易となる。したがって、近似多面体はすべての面を三角形とする。この近似多面体を記述するには新しく考案したトライアングルデータ構造を用いる。トライアングルデータ構造は三角形面と頂点のノードのみを持ち、稜線ノードは持たない。

次に、近似多面体に対して隠れ面消去処理を行う。この処理はセグメントチェーン法を用いる。近似多面体はトライアングルデータ構造によって記述されているので、稜線の取扱いに対し若干の工夫を必要とする。

最後に、スムーズシェイディングを利用して表示を行う。ここでは Phong⁴⁾ の式を用いてなめらかな表示を行う。

3.1 トライアングルデータ構造

セグメントチェーン法の利用には、データ構造に次の2つの条件が必要である。

① 面が三角形であること

② 位相構造を持つこと

したがって、ウィングドエッジデータ構造⁵⁾などによって記述されている一般の多面体に対しては、面を三角形分割することにより処理を行う。また、自由曲面を多面体に近似する場合には、面の数が多くなるので次の条件が必要である。

③ 少ないメモリーで記述可能なデータ構造であること

③の条件からウィングドエッジデータ構造は、曲面近似多面体の記述には不向きである。トライアングルデータ構造は、①、②、③の条件を満たすように考案されたデータ構造である。

3.1.1 基本構造

トライアングルデータ構造はすべての面を三角形とする。したがって、稜線ノードを省略できメモリーの節約を図れる。以下、混乱を避けるためにトライアングルデータ構造においては稜線という用語は用いない。位相構造を付与するために三角形 T および頂点 Q にポインタを設ける。つまり、 T の3個の頂点へのポインタ IVT , JVT , KVT , および T の3辺を介して接する3個の三角形へのポインタ ITR , JTR , KTR である。これら6個のポインタにより、 T から出発していかなる場所へも行くことができる。また、 Q の持つポインタ PTR は Q を共有する三角形のうち任意の一つの三角形に向ける。このポインタにより、 Q から T へたどることができる。つまり、 Q から出発していかなる場所へも行くことができる(図2)。

Fig. 2 The triangle data structure

Fig. 3 Data structure manipulating functions

3.1.2 データ構造操作関数

データ構造をたどる場合には、ポインタをそのまま利用するのではなくデータ構造操作関数を用意しておくことと便利である。図3にデータ構造操作関数を示す。TCCWQ(T, Q)およびTCWQ(T, Q)は、頂点QおよびQを共有する三角形Tを引数とする。Qのまわりの三角形のうち、Qを中心としてTから左まわり側に接する三角形がTCCWQ(T, Q)であり、右まわり側に接する三角形がTCWQ(T, Q)である。TCCWT(T₁, T₀)は、隣接する2つの三角形T₁およびT₀を引数とする。T₀に着目するとT₀の辺を介して接する三角形は3つあり、このうちの1つがT₁である。残る2つの三角形のうち、T₀を中心としてT₁から左まわり側の三角形がTCCWT(T₁, T₀)であり、右まわり側の三角形がTCWT(T₁, T₀)である。QCCWT(Q, T)は、三角形TおよびTの頂点Qを引数とする。Tを中心としてQから左まわり側の頂点がQCCWT(Q, T)であり、右まわり側の頂点がQCWT(Q, T)である。

データ構造操作関数はウィングドエッジデータ構造になぞらえて作成した。したがって、他の関数も作成可能である。しかし、稜線ノードがないことおよび面が三角形のみであることから、多種にわたる関数の必要性は低い。ここには基本的な関数3種を示すにとどめる。なお、ここにあげたデータ構造操作関数は2つの引数を持つが、2番目の引数が中心形状要素となっている。

3.2 セグメントチェインアルゴリズム

```

procedure segment_chain_method;
  var y:integer; (scan line)
begin
  {we assume
  (1)the free form surfaces are approximated with many small triangles,
  and
  (2)are already transformed into 3D screen coordinates.}
  Y-sort list is created with respect to contour edges;
  for y:=scanline_min to scanline_max do
  begin
    Create active contour edge list(ACEL) using Y-sort list;
    Create segment chains using ACEL;
    THINKER tests visibility;
  end
end;

```

Fig. 4 Segment chain algorithm

```

procedure segment_chain_method;
  var y:integer; (scan line)
begin
  {we assume
  (1)the free form surfaces are approximated with many small triangles,
  and
  (2)are already transformed into 3D screen coordinates.}
  Y-sort list is created with respect to contour vertices;
  for y:=scanline_min to scanline_max do
  begin
    Create active contour triangle list(CTL) using Y-sort list;
    Create segment chains using CTL;
    THINKER tests visibility;
  end
end;

```

Fig. 5 Segment chain algorithm in the case of triangle data structure

一般の多面体を対象としたセグメントチェイン法における処理の流れを図4に示す。トライアングルデータ構造には稜線ノードがないため、稜線に関する部分に変更が必要である。変更を行った場合の処理の流れを図5に示す。以下、このアルゴリズムを順次説明す

る。

3.2.1 y ソートリストの作成

トライアングルデータ構造には稜線ノードがないので、y ソートリスト作成のためには、輪郭となる辺を持つ三角形（以下、輪郭三角形と呼ぶ）または輪郭頂点をソートすることになる。輪郭三角形は、複数の輪郭線を持つ場合があり得るが、輪郭頂点は唯一つの y 値を持つ。処理の容易さから、輪郭頂点を y 値によってソートして y ソートリストとした。

3.2.2 ACTL の作成

セグメントチェインを作成するには、左側の輪郭線からスキャンラインと交差している（アクティブな）三角形を順次たどる。したがって、セグメントチェイン作成にあたっては、現在のスキャンラインと交差している左側の輪郭線を求めなくてはならない。この輪郭線は、一定のスキャンラインの繰返しの間変化しないので、リスト化しておくと便利である。このリストを ACTL（Active Contour Triangle List）と呼ぶ。ACTL はスキャンラインごとに必要に応じ更新する。すなわち、現在のスキャンラインからアクティブとなる左側の輪郭三角形を ACTL に挿入し、アクティブではなくなる三角形を ACTL から削除する。

ACTL に挿入する三角形 T_0 は y ソートリストを利用して求める。すなわち、 T_0 は現在のスキャンライン上の頂点 Q_0 を持つ三角形のうち次の①、②の条件のいずれか一方を満足する三角形である（図 6）。

① 次の 4 つの条件をすべて満たす場合

- T_0 が前方面
- T_1 が後方面
- Q_2 の y 座標値がスキャンラインの y 値以上
- Q_1 の y 座標値がスキャンラインの y 値以上

② 次の 4 つの条件をすべて満たす場合

- T_0 が前方面
- T_1 が後方面
- Q_2 の y 座標値がスキャンラインの y 値以上
- T_2 が前方面

ここに、

$$T_1 = \text{TCCWQ}(T_0, Q_0), \quad Q_1 = \text{QCCWT}(Q_0, T_0)$$

$$T_2 = \text{TCWQ}(T_0, Q_0), \quad Q_2 = \text{QCWT}(Q_0, T_0)$$

とする。y ソートリストにある現在のスキャンライン上の頂点すべてについてこの処理を行い、条件に合う三角形は ACTL に挿入する。

ACTL から削除する三角形 T は、次の①、②の条件のいずれか一方を満足する三角形である（図 7）。

① T の 3 頂点の y 座標値が、すべてスキャンラインの y 値以下となる場合

Fig. 6 Triangles to be inserted into ACTL

Fig. 7 Triangles to be deleted from ACTL

② T の左側に三角形が入り込み、T が輪郭部ではなくなる場合

ACTL 中のすべての三角形に対してこの判定を行い、条件に合う三角形は ACTL から削除する。

このようにしてセグメントチェインの左端となる三角形のリストである ACTL を作成する。

Fig. 8 Data representation of segment chains

Fig. 9 Creation of segment chains

3.2.3 セグメントチェーンの作成

セグメントチェーンは二つのテーブルを用いて表現される(図8)。一つはセグメントチェーンの座標値および輝度値を書き込むセグメントテーブルである。もう一つは、セグメントチェーンの両端点に相当するセグメントテーブルの指標値を書き込むチェーンテーブルである。

セグメントチェーンの作成は次のように行う。まず、ACTLから一つの三角形Tを取り出す。Tのスキャンラインと交差している二つの辺のうち、左側の辺を左側辺と呼ぶ。左側辺とスキャンライン平面との交点の座標値および輝度をセグメントテーブルに書き込む。左側辺の対向頂点のy座標値がスキャンラインのy値より大きければ、次の三角形は下側頂点と対向頂点を結ぶ辺を介して接する三角形である。また、対向頂点のy座標値がスキャンラインのy値より小さければ、次の三角形は上側頂点と対向頂点を結ぶ辺を介して接する三角形である(図9)。この操作を左側辺が右側の輪郭線と一致するまで繰り返すと、一つのセグメントチェーンが作成される。セグメントチェーンはACTL中の三角形の個数と同数存在するので、ACTLが空になるまでこの操作を行う。

Fig. 10 Processing time characteristics

4. アルゴリズムの特性

アルゴリズムの処理時間特性を調べるために、Watkinsのアルゴリズムと比較した。対象形状は、自由曲面を多面体に近似し、トライアングルデータ構造によって記述した形状である。この形状に関して、近似の程度を変化させて三角形の数を換え、それぞれの場合の処理時間を計測し、グラフとした(図10,11)。なお、このグラフには自由曲面の定義式から曲面上の点を計算しトライアングルデータ構造を作成する時間、およびシェイディングに要する時間は含まれていない。二つのアルゴリズムはともに、トライアングルデータ構造を用いたので、これらの時間には二つのア

Fig. 11 The scene about which processing time was measured

Fig. 12 A display example

ルゴリズム間で違いはない。

図 12 に他の表示例を示す。

5. 結 言

本研究では自由曲面の隠れ面消去処理の手法として

セグメントチェイン法を用いた。また、セグメントチェイン法を用いるにあたって新しくトライアングルデータ構造を考案した。

トライアングルデータ構造の特徴として次の 2 点があげられる。

- (1) 三角形と頂点のみのノードからなるシンプルなデータ構造である。
- (2) ウィングドエッジデータ構造と同様に、物体の表面を自由にたどることができる。

トライアングルデータ構造は隠れ面消去処理以外にも自由曲面の干渉問題のラフチェック⁶⁾、また一般の多面体の集合演算などにも利用することができる。

セグメントチェイン法は輪郭稜線に着目して処理を行うため、曲面を近似したような形状には特に効果的である。本アルゴリズムによって特に高い有用性が示された。

参 考 文 献

- 1) F. Yamaguchi, T. Tokieda and M. Muramatsu: A Visible Line and Surface Detection Algorithm Created through a Taxonomical Study, Proc. CAD/CAM, Robotics and Automation Conf., Arizona, (1985) 79.
- 2) G. S. Watkins: A Real-time Visible Surface Algorithm, Univ. Utah Comput. Sci. Dept., UTES-CSc-71-101, (June 1970).
- 3) G. S. Warnock: A Hidden-surface Algorithm for Computer Generated Half-tone Pictures, Univ. Utah Comput. Sci. Dept., TR 4-15, (1969).
- 4) B. T. Phong: Illumination for Computer Generated Pictures, Communications of ACM, (June 1975) 311.
- 5) R. G. Baumgart: Geometric Modeling for Computer Vision, Stanford Univ. Comput. Sci., (1974).
- 6) 米田欣也, 村松正司, 山口富士夫: トライアングルプロセッサによる曲面相互の交差問題について, 昭和 59 年度精機学会春季大会学術講演会講演論文集 (1984) 227.