

CSGモデルにおける可視面分抽出に関する研究

A STUDY ON EXTRACTION OF VISIBLE

BOUNDARY FACES FROM CSG MODEL

—高速面画作成のための基礎研究—

—A basic study for high-speed processing shaded pictures—

川上 敬*

嘉数 侑昇*

沖野 教郎**

Takasi KAWAKAMI

Yukinori KAKAZU

Norio OKINO

* 北海道大学工学部

** 京都大学工学部

Hokkaido University

Kyoto University

あらまし 本研究は3次元ソリッドモデルから可視面分情報を用いて高速に面画作成を実行し得るような手法を構築することを目的とする。本手法の大まかな流れとしては、まずCSG (Constructive Solid Geometry) モデルから境界、稜線情報や面情報を取り出し、それらの情報に視線を与えることによって2次元投影図を作成する。つぎにその図から隠線消去された2次元エッジ情報を抽出し、そのエッジをリンク、交点をノードとすることによって投影図自体を無向グラフとみなし、グラフ探索によって、可視面分を分割、抽出し、抽出された可視面分データに対してスムーズ・シェーディングの技法を用いて高速に面画を出力する。

Abstract This paper proposes a new approach of making shaded pictures in high-speed by generating the visible boundary faces from given 3D solid model. The new approach is as follows: (1) Generate boundary data which include a set of edges and surfaces information. (2) Make a 2D perspective data in which hidden lines must be removed. (3) Regard the perspective as an undirected geometric graph, and partition off the visible boundary faces by developed algorithms based on graph theory. (4) Apply a smooth shading technique to the extracted visible boundary faces, and make a shaded picture. In this study, TIPS-1 system is introduced.

1. まえがき

現在、CADの分野では、形状モデリング時における対話特性から見て形状処理の高速性が要求される。特に意匠設計などの場合においてその処理に時間を要している画面 (Shaded picture) 作成に関して、その高速処理への期待は大きいものがある。このために今日までも多くの研究が行なわれてきた。この問題に対して言えることは、形状情報のうちある視線方向から見える部分情報を扱っていることである。又、形状モデリングの一方法であるCSG (Constructive Solid Geometry) モデルは、3次元形状情報を完備しているが、前述の例のように目的によっては一部の情報しか必要としない場

合が多く存在する。そのなかでもコンピュータ・グラフィックスへの適用は特に顕著な例であり、ある視線方向を与えた時に可視となる部分の面分情報があればこの可視面分のみの処理を実行すれば、面画作成等の高速化処理が期待されよう。このような観点から期待に応えるべく筆者らは、コンピュータ・グラフィックスに適したデータである可視面分を抽出、分割するアルゴリズムを提唱し、画面の出力を行なうシステムの開発を行なってきた [1] が、このシステムにおいては、穴のある形状を扱うことができなかった。そこで本稿では、穴のある形状についても扱うことのできる可視面分抽出アルゴリズム、並びに画面出力システムの開発を行なっ

たので報告する。なお、3次元形状モデラーとしてTIPS-1[4]を用い、扱うセグメントを構成する曲面は、円柱面や球面などの2次曲面までとする。

2. CSGモデル

一般に形状は数学的に3次元空間中の点の集合として扱うことができる。形状を S 、その境界を S_B とし、直交座標系の任意点を $X=(x,y,z)$ とすると、形状は次式で表わすことができる。

$$S = \{ X | f_B(X) \geq 0 \} \quad (1)$$

$$S_B = \{ X | f_B(X) = 0 \} \quad (2)$$

ここで、 $f_B(x)$ は境界上で0、形状内部で正、外部で負となるような関数である。このようにして形状は不等式の与える空間として定義することができる。

CSG (Constructive Solid Geometry)モデルにおいて、形状 S は、いくつかのセグメントとよばれる部分集合 S_j の和集合によって定義される。

$$S = \bigcup_{j=1}^m S_j \quad (3)$$

また、このセグメントはいくつかのエレメント E_{ij} で構成される。ここで、領域を不等式群 $G_{ij}(X) \geq 0$ (これをエレメント E_{ij} とよぶ)の交わりとして定義される半空間とすると領域中のセグメント S_j は次の式で表現される。

$$S_j = \bigcap_{i=1}^n E_{ij} \quad (4)$$

ただし、 E_{ij} は $G_{ij}(X) \geq 0$ によって定義される空間領域を意味する(図1)。

したがって式(3),(4)より、集合演算による形状幾何の一般式は次のようになる。

$$S = \bigcup_{j=1}^m \left\{ \bigcap_{i=1}^n E_{ij} \right\} \quad (5)$$

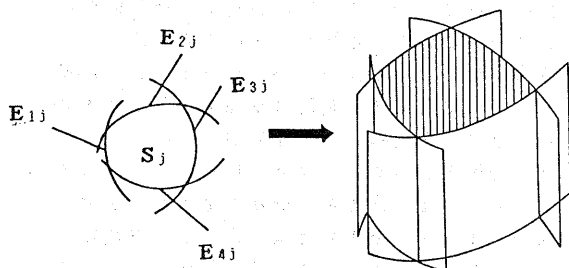


図1. セグメントの定義

ここで、正のエレメント(存在領域)を P_{ij} 負のエレメント(除去領域)を Q_{ij} として(5)式を書き直すと、

$$S = \bigcup_{j=1}^m \left\{ \bigcap_{i=1}^n P_{ij} \right\} - \bigcup_{j=1}^m \left\{ \bigcap_{i=1}^n Q_{ij} \right\} \quad (6)$$

となる。つまりCSGモデルにおいて形状 S は、基本形状(プリミティブ)の和、差、積集合演算に定義できる。この集合演算の例を図2に示す。

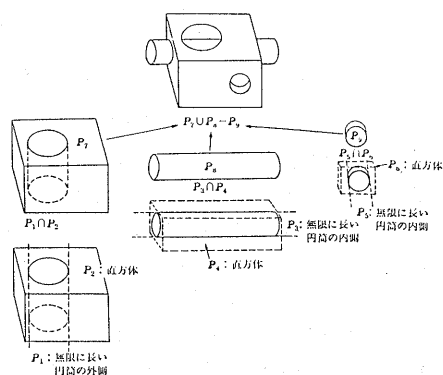


図2. CSGモデルの集合演算例

前述のように、CSGモデルで形状は基本形状の集合演算によって定義できるために以下のような利点がある。

- (1) 構造簡明
- (2) データ少量
- (3) 入力、修正が容易

しかし、形状から可視面分を抽出するためには境界、稜線情報が必要となるが、CSGモデルにはそのような情報が陽には表現されていない。その点で有利になると思われる形状モデルはB-Repである。以上の考察から形状の可視面分抽出を行なうためのモデルとしては、CSG、B-Repのデュアルモデルを採用するのが最適であろう。すなわち、最初にCSGモデルが定義されており、そこからCSGに対応づけられたB-Repが自動生成される形状モデル[3]を採用することにする。

3. 可視面分抽出の基本アルゴリズム

ある形状に対して視線を与えた時の可視面分を抽出するために必要となる情報は、以下のようなものであると思われる(図3)。

- (1) 3次元曲面情報(面の方方程式を含む)
- (2) 3次元稜線、輪郭線情報

(3) トポロジー情報

(4) ある視線に対する隠線消去された2次元線分情報（線分間の結合情報を含む）

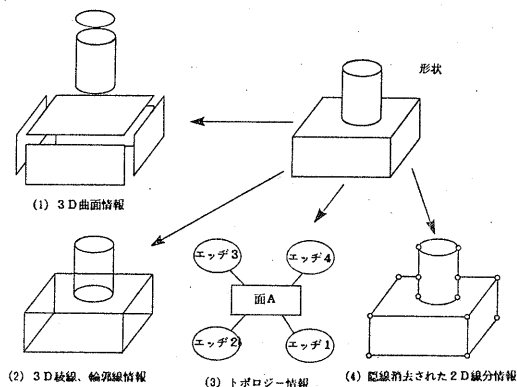


図3. 可視面分抽出に必要な情報

したがって可視面分抽出を行なう処理はCSGモデルから上述の情報を得ることから始まる。そこで2節で述べたように、最初に定義されたCSGモデルから対応づけられたB-Repが自動生成される形状モデルにより、(1), (2), (3)の情報を生成する。次にこの情報にある視線を与え2次元投影図を作成し、その図に対して隠線消去を行ない、(4)の情報を得る。この(4)の情報というのは、つまり隠線消去をほどこしたワイヤー図となっているので、このデータにおけるエッジをリンク、交点をノードとみなすと、2次元投影図はそれ自体幾何情報を持った無向グラフと考えることができる。よってエッジ間の結合状態を利用したグラフ探索により面分抽出を行なうことができる(図4)。

次に、抽出された面分ごとについて輝度情報を付加し、スムーズ・シェーディングによって面画(Shaded Picture)の出力を行なうシステムを開発し、計算結果の検証を行なう。

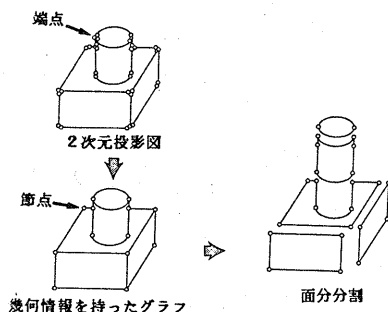


図4. 面分抽出例

4. 可視面分抽出の定義

ここで、ある形状から可視面分を抽出するという事は、どのようなことかを考察する。特に3節で述べた幾何情報を持った無向グラフから面分を分割、抽出することについて考える。

まず、この探索空間である幾何情報を持ったグラフ全体を G とすると、 G は以下の式で書ける。

$$G = (V, E) \quad (7)$$

$$V = \{v_i | i \in I_v\} \quad (8)$$

$$E = \{(v_i, v_j) | i, j \in I_v, i \neq j\} \quad (9)$$

G : 探索空間であるグラフ全体

V : グラフ中のVertexの集合

E : グラフ中のEdgeの集合

I_v : Nodeの番号の集合

また、 v_i は2次元座標における幾何データを持っているので次式のようにも表現できる。

$$v_i = (x_i, y_i) \quad (10)$$

さらに、グラフ G は形状の性質上、必ず閉じたグラフとなるので孤立点や孤立辺は存在しない。

ここで可視面分の抽出をグラフ G から以下の条件を満たす部分グラフ SG (Sub Graph)を取り出すことと定義する。

(G-1) SG は1つもしくは複数のループで構成される。

(G-2) ループの内側にリンクが存在しないか、もしくは内側に存在するリンクを使っても分割不可能な最小単位のループ(以下単位ループと呼ぶ)である。

面分が複数のループで構成される場合というのは、貫通穴の開いた形状の時などで、穴の開いている面に対して、外側のループで囲まれる領域から、内側のループが囲む領域を除いた部分を面分とする。

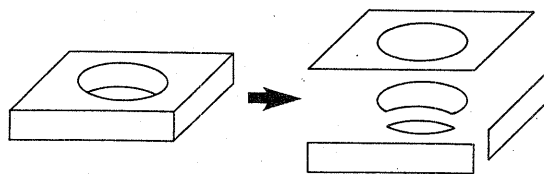


図5. 穴の開いた形状の面分抽出例

5. システム構成

以上の考察を用いて可視面分を抽出するための各種アルゴリズムと、それらを統合しCSGモデルから面画を出力するシステムについて以下に述べる。このシステム構成を図6に示す。各ステップについて以下の各章で詳述する。

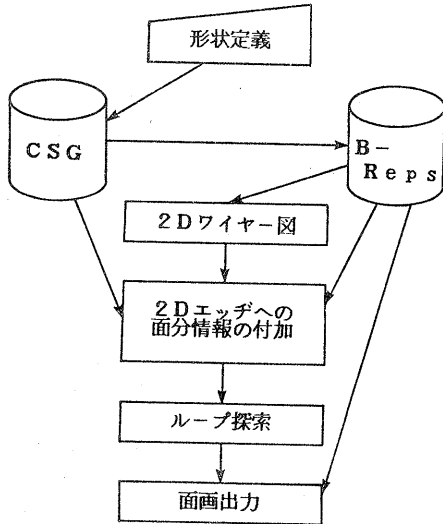


図6. システム構成

6. CSGから境界・稜線情報(B-Rep s)の生成

B-Rep s (Boundary Representation)モデルは、形状の表面を構成している面、稜線、頂点の幾何データと、それらの結合関係(トポロジー・データ)によって形状表面および内部まで定義するものである。CSGからB-Rep sへの変換アルゴリズムは図7に示すものである。

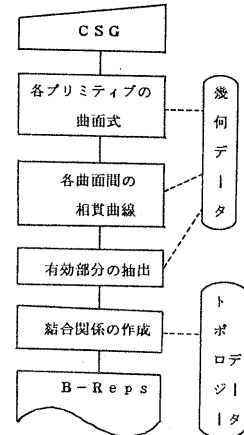
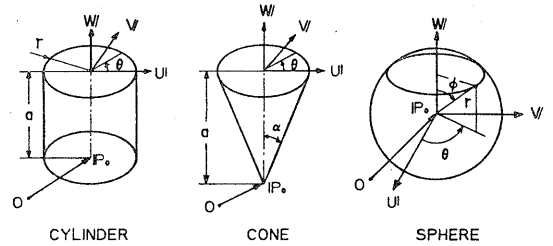


図7. CSGからB-Rep sへの変換アルゴリズム

ここで扱うセグメントを構成する曲面は平面と2次曲面までとしているので、各プリミティブの曲面式はCSGデータから比較的容易に求めることができる。平面と2次曲面の一般式を



$$\begin{aligned}
 IP &= IP_0 + aw + r(u \cos \theta + v \sin \theta) \\
 IP &= IP_0 + aw + a \tan \alpha (u \cos \theta + v \sin \theta) \\
 IP &= IP_0 + rk \cos \theta + r \sin \theta (i \cos \phi + j \sin \phi)
 \end{aligned}$$

図8. 2次曲面のパラメータ表現

それぞれ右手系直交座標表現(x,y,z)を用いて

$$b_1 x + b_2 y + b_3 z + b_4 = 0 \quad (11)$$

$$\begin{aligned}
 &a_{11} x^2 + a_{22} y^2 + a_{33} z^2 + 2a_{12} xy + 2a_{23} yz + \\
 &2a_{13} zx + 2a_{14} x + 2a_{24} y + 2a_{34} z + a_{44} = 0 \quad (12)
 \end{aligned}$$

とおき、各プリミティブごとに係数 b_i, a_{ij} の値を求める。

曲面間の相関曲線は、曲面の組み合わせにより次の3つの場合がある。

- (1) 平面と平面
- (2) 平面と2次曲面
- (3) 2次曲面と2次曲面

(1),(2)については一般式を用いて相関曲線を求めるが、(3)の場合そのままでは困難である。そこで2次曲面を2個のパラメータを使って表わす(図8)。パラメータ表現された曲面の相関曲線は、4個のパラメータを3本の関係式を使って1個にすれば求まる。ただしこの場合B-Rep sデータ中では、パラメトリック曲線を点群曲線として格納する。

一般には相関曲線の一部分が実際の形状稜線となるので、相関曲線から有効部分を抽出する必要がある。半空間 S_{ij} の境界を B_{ij} 、 S_i をセグメント、 P_i, Q_i をそれぞれ正負のセグメントとすれば、相関曲線上の点Xが有効部分に含まれるための条件は次の(1),(2)を満足することである。ここで点Xは、

$$X \in (B_{i1, j1} \cap B_{i2, j2}) \quad (13)$$

従って相関曲線の有効部分は、条件(1),(2)を満たす点Xの集合と定義できる。

$$(1) x \in (S_{i1} \cap S_{i2}) \quad (14)$$

(点Xがセグメント S_{i1} , S_{i2} の両方に含まれる。)

(2) S_{i1} , S_{i2} が共に除去モードセグメントのとき、

$$x \in \left(\bigcup_{i=1,2} Q_i \right) \cap \left(\bigcup_{i=1} P_i \right) \quad (15)$$

(点Xが S_{i1} , S_{i2} 以外の除去モードセグメントに含まれず、かつ、ある存在モードセグメントに含まれる)

S_{i1} , S_{i2} の少なくとも一方が存在モードセグメントのとき、

$$x \in \left(\bigcup_{i=1,2} S_i \right) \quad (16)$$

(点Xが S_{i1} , S_{i2} 以外のセグメントに含まれない)

ただし、 $S_{i1} = S_{i2}$ なら S_i , P_i , Q_i はその境界を含み、 $S_{i1} \neq S_{i2}$ なら S_i , P_i , Q_i はその境界を含まない。

以上で $B-R e p s$ の幾何データが求まるがこれだけでは面、稜線、点が空間中にバラバラに存在しているだけなので、3次元のソリッド形状を表わしているとはいえない。そこでトポロジー・データを作成し、面、稜線、点の結合関係を作成する。

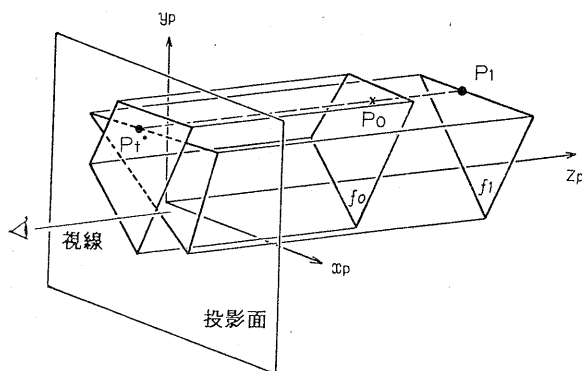


図9. 2D投影図の隠線消去

7. 2次元ワイヤー図の作成

6節で得られた $B-R e p s$ データに対して視線を与えることで2次元ワイヤー図を作成することができる。円柱面などの2次曲面に視線を与えると輪郭線が見えるが、これは $B-R e p s$ データ中には含まれていないので、まずこ

の輪郭線を求める。次に輪郭線も含めてすべての3次元エッジを視線方向に投影し、すべてのエッジ間の交点を求め、エッジ上に交点を持たない小エッジ群に分割する。それらの小エッジ間の中点から視線方向に直線を発生させたときに、その直線が中点より視線側において形状と交わればそのエッジは不可視となる(図9)。

1 2 3 4 5

NO.	PS	PE	pointer	ICASE
-----	----	----	---------	-------

NO. ; エッジ NO. PS ; スタート ポイント

PE ; エンド ポイント pointer ; 幾何データ ポインタ

ICASE ; =1; 直線 =2; 楕円 =3; そう曲線

=5; 方物線 =99; 点群曲線

図10. 2D可視エッジリスト

この隠線処理により可視となったエッジが2次元エッジ情報として図10の型で格納される。以上のようにして作られたワイヤー図はデータとしては両端を持ったエッジが投影面上にバラバラに存在しているだけなので、両端の座標を利用してエッジ間、交点間の結合関係を作成し、ワイヤー図を幾何情報を持ったグラフへと変換する。

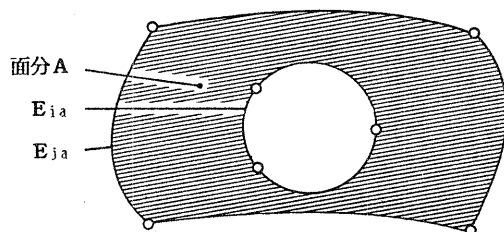


図11. 複数ループで構成されている面分例

8. 2次元エッジへの面分情報の付加

筆者等が行ってきた研究において文献[1]のアルゴリズムでは7節で求められたグラフに対して無作為にループを探索し、そのループが面分を表わす単位ループであるかの判定を行ない、違った場合にはエッジの幾何データを利用したルールにより探索空間を縮退し、探索を繰り返すといったものだが、前述のようにこの方法では面分が複数のループで構成されている場合(面に穴が開いているような場合)の処理に難点があった(図11)。つまり、2本のエッジ

E_{ia} と E_{ja} は同じ面分 A を構成するエッジであるが、直接の結合関係は存在しないので判定が困難となる。又、ループ探索時のルールがあまり有効でなかったため、エッジの数が増大すると組合せ爆発を起こす危険性さえあった。そこで今回のアルゴリズムではこの部分に変更を加えた。すなわち2次元ワイヤー図において、1本のエッジは必ず異なる2つの面分領域の境界線の一部である。なぜなら、一般の形状において1本のエッジをはさんで左右の面が同一面であるということはいえない。ただし形状外領域(背景)も1つの面分とする。

そこで各々の2次元エッジがどの面分間の境界線であるかという情報を付加することにする。これによって、ループ探索の際に1つの面分に関わっているエッジを集めてくればよいので探索空間が非常に小さくなり、探索速度が速くなる。又、組合せ爆発の心配もなくなる。従って図11の場合においても、直接結合関係を持っていない E_{ia} 、 E_{ja} が、実は同じ面分 A を構成するエッジだという判定を簡単に行なうことができる。

以下に面分情報付加の方法について述べる

8.1. 面分内代表点の作成

まず、そのエッジが境界となっている2面分の代表点を求める。

そのために、エッジと中点において垂直に交わるような直線を発生させる。この直線はエッジの線種により異なる。

a) エッジが直線の場合

エッジの両端点を $(x_1, y_1), (x_2, y_2)$ とすると、直線は

$$Y = -\{(x_2 - x_1)/(y_2 - y_1)\} X + A \quad (17)$$

$$[A = \{(y_1 + y_2)(y_2 - y_1) - (x_1 + x_2)(x_2 - x_1)\} / 2(y_1 - y_2)]$$

b) エッジが2次曲線の場合

2次曲線方程式が

$$aX^2 + 2hXY + bY^2 + 2gX + 2fY + c = 0 \quad (18)$$

また、中点の座標を (x_0, y_0) とすると直線は

$$BY - CX + (Cx_0 - By_0) = 0 \quad (19)$$

$$[B = hx_0 + by_0 + f]$$

$$[C = ax_0 + hy_0 + g]$$

c) エッジが点群曲線の場合

点群曲線は直線分の連鎖と考えられるので中点を内部で含んでいる直線分の両端点を

$(x_1, y_1), (x_2, y_2)$ とすると直線は(17)式になる。

このようにして求めた直線と他のエッジとの交点をすべて求め、エッジの両側における最近交点を探し、エッジの中点との中間に位置する点を面分内代表点とする(図12)。

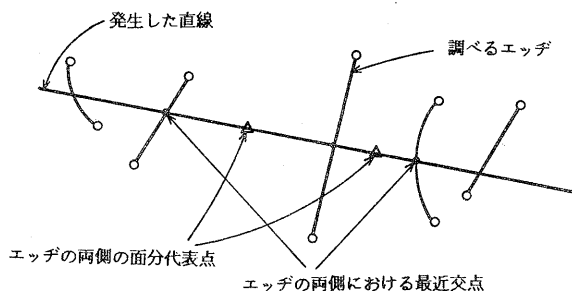


図12. 面分代表点の作成

8.2. 面分内代表点の3次元形状における従属面

次に、面分内代表点が3次元形状においてどの面に属するのかを判定する。

面分内代表点を3次元に変換(視線方向の深さはある一定値にとり)し、その点を通る視線方向の直線を発生させる。図13に示すような視線の場合、直線の方程式は次のようになる。

$$\frac{X - x_0}{\cos \theta \cdot \sin \gamma} = \frac{Y - y_0}{\sin \theta \cdot \sin \gamma} = \frac{Z - z_0}{\cos \gamma} \quad (20)$$

更に、この直線と形状を構成している全セグメントとの交点を求め、形状の境界で0、内部で正、外部で負となるような関数を利用して一番手前側に見える形状表面を判

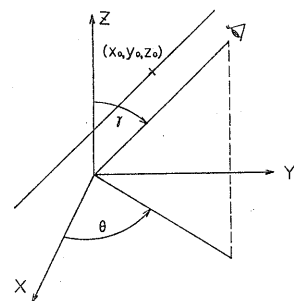


図13. 視線方向の直線

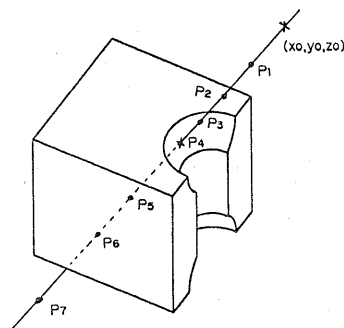


図14. 形状表面の判定

定する (図14)。

このようにして求まった面分情報をエッジ～面分関係リストとして記憶する。このエッジ～面分関係リストのデータ構造は、図15のようになっている。

1	2	3	4
SEG1	SUF1	SEG2	SUF2

SEG1 : 面分1が示すセグメントNO.

SUF1 : 面分1が示すサーフェイスNO.

SEG2 : 面分2が示すセグメントNO.

SUF2 : 面分2が示すサーフェイスNO.

図15. エッジ～面分関係リスト

9. ループ探索

8節で作成したエッジ～面分関係リストから1つの面分に関わっているエッジを取り出してきて、それらの端点の座標によりエッジをつなぎあわせてループを作成する。

又、1つの面分が複数のループで構成されている場合、ループ間の包含関係を調べ、その関係をトリー構造で表わす (図16)。

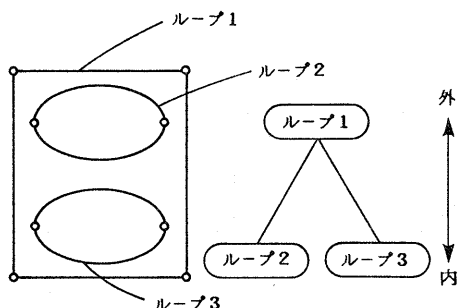


図16. ループの包含関係

ここでループ間の包含関係は以下の式で表わすことができる。

ループ LP_i とループ LP_j が

$$LP_i = \bigcup_{k=1}^n E_{ki} \quad (21)$$

$$LP_j = \bigcup_{k=1}^m E_{kj} \quad (22)$$

(E_{ki} : LP_i を構成しているエッジ) で表わされるとき、

$$\bigcup_{k=1}^m E_{kj} \subset LP_i \quad (23)$$

ならば

$$LP_j \subset LP_i \quad (24)$$

が成り立つ。(ただし $\subset$ の記号は幾何学的な意味で内側に存在する事を意味している。) 何

故なら、図17のようにループ同志が交差するような事はありえない。

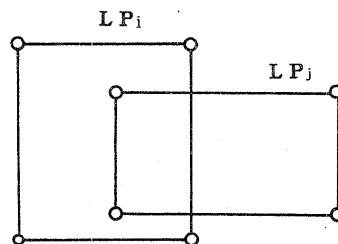


図17.

10. 面画作成

以上までの処理によって抽出された面分情報により面画の出力を行なう。ある視線から見える形状が面分に分割されているので、その面分1つずつをスムーズ・シェーディングを利用して塗りつぶしていくことができる。

スムーズ・シェーディングとはポリゴンの頂点に輝度を与え、それによってポリゴン内部の輝度を線形補間により求めるものである。その為にはまず面分をポリゴンに近似する必要がある。面分を構成しているあるエッジが直線の場合には、両端点でよいが曲線の場合にはある程度細かく分割し、ポリゴンに近似する。

次に面分 (ポリゴン) の頂点それぞれに輝度を与える。各頂点をその面が示す曲面上に投影し、その点における法線ベクトルを求めて光線ベクトルとの内積により輝度を算出する (図18)。

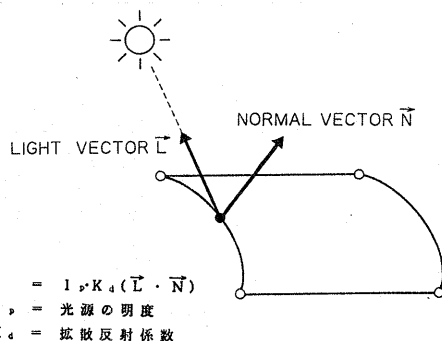


図18. 面上の1点における輝度の算出

このような手順で面分の塗りつぶしは行なわれるが、問題となるのは (図19) のようにループの中にループが存在する場合で、このときには重ね塗りの方法を用いることにする。すなわちループの包含関係が、

$$LP_1 \supset LP_2 \supset LP_3$$

であるという判定がなされた場合、外側から順に塗り重ねていく事ができる。

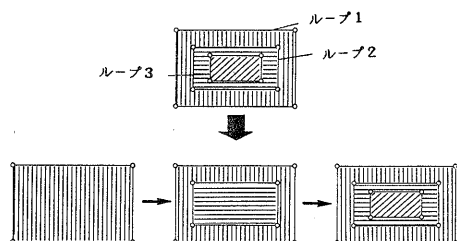


図19.重ね塗りによる面塗り

11. 実験結果

計算機実験の結果を以下に示す。

図20は面画出力の結果である。又、表1にはそれぞれの各処理ごとのCPU-TIMEを示した。表1の結果をグラフにしたものが図21である。このグラフから解るように本アルゴリズムによる面画出力は、2Dエッジへの面分情報付加に要する処理時間に大きく影響を受ける。しかし、この処理はセグメント数ではなく、可視エッジの数に影響を受けるので急に処理時間が増大するような事はない。又、表1においてShadingにかかる処理時間が急激に増大しているが実際には、グラフに示すような緩やかな曲線になるものと思われる。なぜなら、この処理は塗りつづす面分数に比例するものである。(C)のデータはセグメント数の割に面分数が多い形状なのでこのような結果が得られたのであろう。

表2は面画作成に要した時間を他のアルゴリズムと比較したものである。これにより本アルゴリズムの高速性が検証できた。なおCPUはHITAC M-680H、Graphic DisplayはJRC-NWX237を用いた。

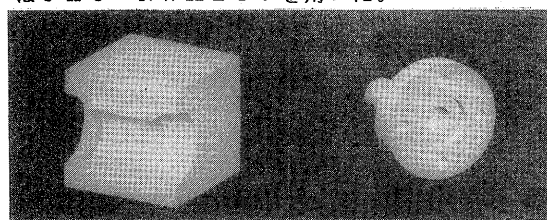
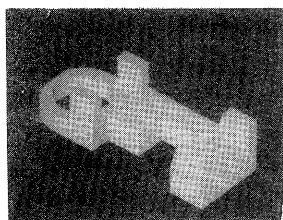
TEST DATA A
セグメント数4TEST DATA B
セグメント数11TEST DATA C
セグメント数13

図20.面画出力例

DATA	B-Reps DATA 作成	ワイヤー図 作成	2Dエッジへの 面分情報付加	ループ 探索	Shading	TOTAL
A	0.02	0.11	0.43	0.01	0.21	0.78
B	0.07	0.15	0.54	0.01	0.24	0.96
C	0.15	0.24	0.62	0.03	0.38	1.42

表1.各処理ごとのCPU-TIMEの比較

単位(sec)

DATA	Reg-tracing アルゴリズム	スキャンライン アルゴリズム	本アルゴリズム
A	4.54	3.86	0.78
B	5.36	4.94	0.96
C	6.17	5.02	1.42

表2.他のアルゴリズムとの面画作成に要するCPU-TIMEの比較

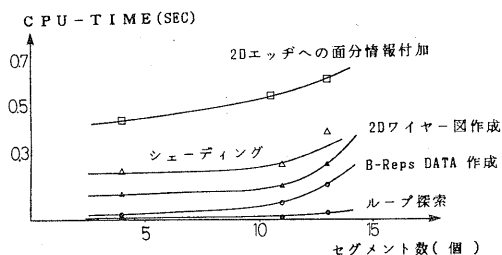


図21.各処理ごとのCPU-TIMEの比較

12. まとめ

最後に以上のことを次の3点にまとめて、本稿の結言とする。

- (1) CSGモデルにおける可視面分抽出の定義をおこなった。
- (2) 穴のあいた形状についても扱える可視面分抽出アルゴリズムを示した。
- (3) 抽出された可視面分を利用して面画を作成するシステムを開発し、その高速性と有効性を示した。

参考文献

- [1] 川上、嘉数、沖野：CSGモデルにおける可視面分抽出に関する研究(昭和61年度精密工学会北海道支部学術講演会講演論文集)
- [2] 渡部、嘉数、沖野：ソリッド形状モデリングにおけるCSGからB-REPSへの解析的変換の研究(精密工学会誌、VOL 53、2/1987)
- [3] 沖野教郎：自動設計の方法論(養賢堂)
- [4] 沖野、嘉数、久保：自動プロセッサTIPS-1の開発(精密機械、VOL 44、3/1978)
- [5] 山口富士夫：コンピュータ・ディスプレイによる図形処理工学(日刊工業)
- [6] JAMES D. FOLEY/ANDRIES VAN DAN: Fundamentals of Interactive Computer Graphics