ソリッドモデル形状設計システムFREEDOM-Ⅱにおける形状処理

山口富士夫　時枝敏也
九州芸術工科大学

1. まえがき

近年，厳密に形状をコンピュータ記憶に表現し，コンピュータディスプレイを用いて形状設計を行えるシステム（ソリッドモデル形状設計システム）が，いくつか現れてきた[1),2)]．これらのシステムは，設計，生産に関連する多くの形状処理を自動化する可能性を有するという点できわめて重要な意味を持つものと考えられる．しかしながら現在の所，これらのシステムの多くには次の問題点がある．すなわち，

(1) 処理アルゴリズムが非常に複雑である．従って，プログラムが大規模となり，システムの信頼性と保守に問題があり得る．
(2) 処理速度が遅いため，対話的作業に不向きである．
(3) 処理の対象形状に制約がある．すなわち，殆んどのシステムでは，平面および円筒面，円錐面などの初等幾何曲面のみを有する形状を対象とし，自由形状の含まれた形状を統一的プロセッサで処理するまでには至っていない．

本システムは，上記の問題点を解決する目的で設計し，計画された．現在，平面および初等幾何曲面を含む対象形状に対しての処理プロセッサの開発をほぼ完了した．

本システムの特色は，処理に関連する可能性のあるポリゴン面を三角形分割している点にある．このため，本質的な処理は，二つの三角形相互の干渉問題に帰すことができ，処理アルゴリズムを簡単化できた．また，三角形相互の結合の連続性により，交線ループを求める際，前の演算結果を次の演算に有効利用して，処理の高速化を計った．

初等幾何曲面に対する処理演算も，基本的には，多面体処理を行う多面体ブール演算プロセッサにより行っている．現在のところ，自由形状は処理対象としていないが，本アルゴリズムの基本は，三角形表現された面を対象としているので，自由形状を含む形状に対しても利用できると考えている．

2. 形状モデルのデータ構造

基本的には，Baumgartの提案している *Winged Edge Data Structure* を利用している[3)]（図1参照）．このデータ構造は，形状要素間の結合関係をコンピュータ処理に便利に表現しているため，形状の持つ連続性の性質を有効に利用し得る．本システムでは，Baumgartの提案のものに対し，円筒面，円錐面，球面などの初等幾何曲面を含め，さらに面の中に穴をも許した．穴付きの面を許すために，通常の実稜線（Real Edge）に対し，Bridge稜線を導入した．Bridge稜線とは穴に至る橋渡しの役目の稜線である（図2の破線）．Bridge稜線は，E@.PFACE = E@.NFACE であるため，Baumgartのデータ構造操作関数のいくつかは変更を要する．例えば，面Fの稜線の反時計回転プロシージャは，中間の頂点も引数とする関数 ECCWFS を用いてつぎのように行う．すなわち

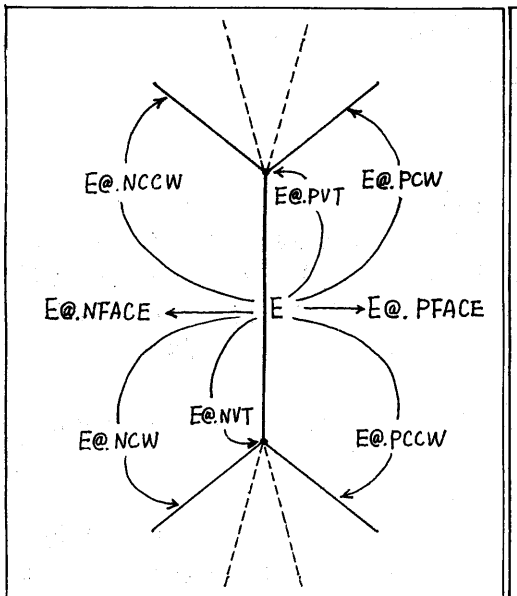

図1 Winged Edge Data Structure

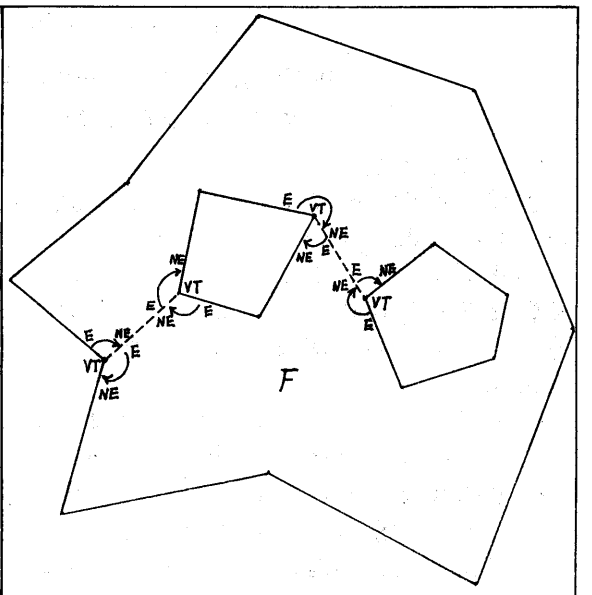

図2 Bridge稜線と面の反時計向き回転

```
E0:=F@.FPED;   E:=E0;   VT:=VCCWF(E,F);
REPEAT   NE:=ECCWFS(VT,E,F); VT:=φTHER(NE,VT); E:=NE UNTIL E=E0;
```

上のPASCALによるプログラムにおいて，VTは頂点，E,E0,NEは稜線，Fは面を表すいずれもポインタ変数である．FPEDは，面を構成する一つの稜線を指すポインタのためのフィールド，VCCWFは，面F上，稜線Eから反時計回転向きの頂点を求める関数，ECCWFSは，面F上，反時計回転向きに，頂点VTを介して接続する稜線を求める関数，またφTHERは，稜線NEに対し，頂点VTと異る他端の頂点を求める関数である(上のプログラムについては図2参照)．

3. 形状の標準化——三角形分割

処理を行う前に形状を何らかの方法で標準化しておくことは，プログラムを簡易化する点で一般に効果的である．

このための方法として，形状を凸多面体の集合に分割するやり方があるが，凸多面体に分割することそれ自体が相当複雑な処理となること，また二つの凸多面体同志でも複数個の交線ループが存在する場合があり，これらを残らず処理することは簡単ではないなどの難点がある．

本システムでは，多面体を構成する面を三角形分割する方法を用いた．三角形分割方式により得られる利点を次に列挙する．

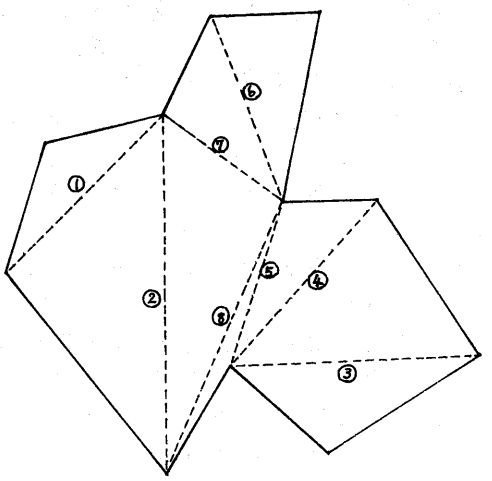

図4　ポリゴン面の三角形分割

(1) 下記に示すように，ポリゴン面を三角形分割することは比較的簡単である．
(2) 交線ループ上の1点を発見する処理において，多面体相互，ポリゴン面相互，三角形相互の概略チェックを行い，最後に厳密な計算が行える．すなわち，粗から密への処理が可能となる．
(3) 二つの三角形相互の交線は唯一つに限られるので，三角形相互の干渉を基本とする方法により，交線ループを残りなく求めることができる．
(4) 4. に示すように，交線ループは，交差三角形を順次たどることにより局所的に決定でき，かつこの場合，前の演算結果を次の演算に有効利用できる．

ポリゴン面は，穴が含まれている場合も，Bridge稜線を介して，一つながりのループを構成している(図2参照)．したがって三角形分割は，穴の有無にかかわらず同一の手続きによって行う．その手続きは次の通りである．まずポリゴン面を構成する各頂点の凹凸を判定する．一つの凹頂点に注目する．それに引続いて隣接する二つの稜線により実三角形ができる場合には，その三角内に他の頂点が入っているかを調べる．もし，いずれの頂点も入っていなければ，ポリゴンをその三角形の某三稜線で分割しポリゴンから除外する．各凹頂点に関し，それが凸頂点になるまで上述の手続きを繰返す．凹頂点がなくなったら，一頂点より放射状に各頂点を結んでポリゴンを分割し，三角形分割を完了する．

4. 多面体ブール演算プロセッサのアルゴリズム

二つの互いに交差する多面体a，bの空間をA，Bで表す．以下において，スター記号＊は，対象とする図形に外接し，座標軸に直交する面を有する直方体空間を表すものとする．

(1) 初期処理

(1-1) コモンボックス c の作成

その空間が $C = A^* \cap B^*$ で表されるコモンボックス c を作る(図5の①).

(1-2) 交差ポリゴンスタックの作成

多面体 a, b を構成する各ポリゴンのうち, その外接直方体と C とが交差するポリゴン群 $\{f_{a1}, f_{a2}, \cdots\cdots\}$ 及び $\{f_{b1}, f_{b2}, \cdots\cdots\}$ について, $F_{ai}^* \cap F_{bj}^*$ (時Φ)なるポリゴンペアを IPSTACK にプッシュする. IPSTACK は交差の可能性のあるポリゴンペアを含んでいる(図5の②).

(1-3) IPSTACK 中のポリゴンペアをそれぞれ三角形分割し, 各ポリゴンの TLIST1 のフィールドに, 分割によって生じた三角形のリストを作る(図5の③).

(1-4) 同一ポリゴンペアに対し, TLIST1 よりそれぞれ三角形を取り出し, それらの外接直方体の交差する三角形ペアのリスト(ITLIST)を作る(図5の④). この時, それぞれの交差三角形ブロックの TLIST2 のフィールドに, 交差する相手の三角形と ITLIST 中のブロックに対するポインタも記入しておく(図8参照).

(2) 交線ループ上の1点の発見

ITLIST より三角形を取り出し, 実稜線による交差があるかを厳密に調べる. その結果, 交差していないことがわかれば ITLIST よりそのペアを除去する. 交差があれば, 交差実稜線 e_r, e_r を含む三角形 t_r および相手の三角形 t_s を記録する. 一対の三角形ペア t_r, t_s の交差は次のように簡単に求められる. まず一方の三角形より順次有向実稜線 e_r (始点, 終点の位置ベクトルを V_a, V_b とする) を取り出して, 次の5条件をすべて満足するものがあるかを調べる. なお本報告で"頂点 V に関し, ある平面に対する S"とは, 頂点 $V=[x_0\ y_0\ z_0]$ と相手の平面式のパラメータ a, b, c, d (平面式 $ax+by+cz+d=0$) による $S = ax_0+by_0+cz_0+d$ を表すものとする($N=[a\ b\ c]$ は三角形の表向き法線ベクトルに一致させてある). また V_1, V_2, V_3 は相手の三角形頂点の位置ベクトルである.

(第1条件) $T_r^* \cap T_s^* \neq \phi$

(第2条件) $(S_a \geqq 0)$ AND $(S_b \leqq 0)$ または $(S_a \leqq 0)$ AND $(S_b \geqq 0)$

次に $Q = V_a + (S_a/(S_a - S_b)) \cdot (V_b - V_a)$ ……(1)

とするとき

(第3条件) $\{(V_1 - Q) \times (V_2 - Q)\} \cdot N \geqq 0$

(第4条件) $\{(V_2 - Q) \times (V_3 - Q)\} \cdot N \geqq 0$

(第5条件) $\{(V_3 - Q) \times (V_1 - Q)\} \cdot N \geqq 0$

もし, 上の5条件を満足する実稜線が見つからなければ, 二つの三角形の関係を逆にして, 同様のテストを再度行う(図5の⑤).

(3) 交線ループの処理

互いに交差するポリゴンを f_r, f_s, f_r において実稜線を含む交差三角形を t_r, e_r に交差する f_s 中の三角形を t_s とする. また t_r を表(おて)から見て反時計回転の順序に e_r の端点位置ベクトルを V_a, V_b, V_a, V_b に関し t_s に対する S を S_a, S_b とし e_r は点 Q_i で $f_s(t_r)$ と交差しているとする. すでに三角形分割されている f_r, f_s 中の三角形を次のように追跡して, 交線ループ上の次の実交点 Q_{i+1} が求まる(図5の⑥)

(3-1) t_r の第3頂点 V_c に関し, t_s に対する S_c を求め, e_a と e_b のうち, 両端点の S が異符号となる稜線が t_s と交差するかを調べる(図6).

(3-2) もし交差すれば(図7-(a)), その稜線が新 e_r となる. 現 t_r に対し新 e_r を共有

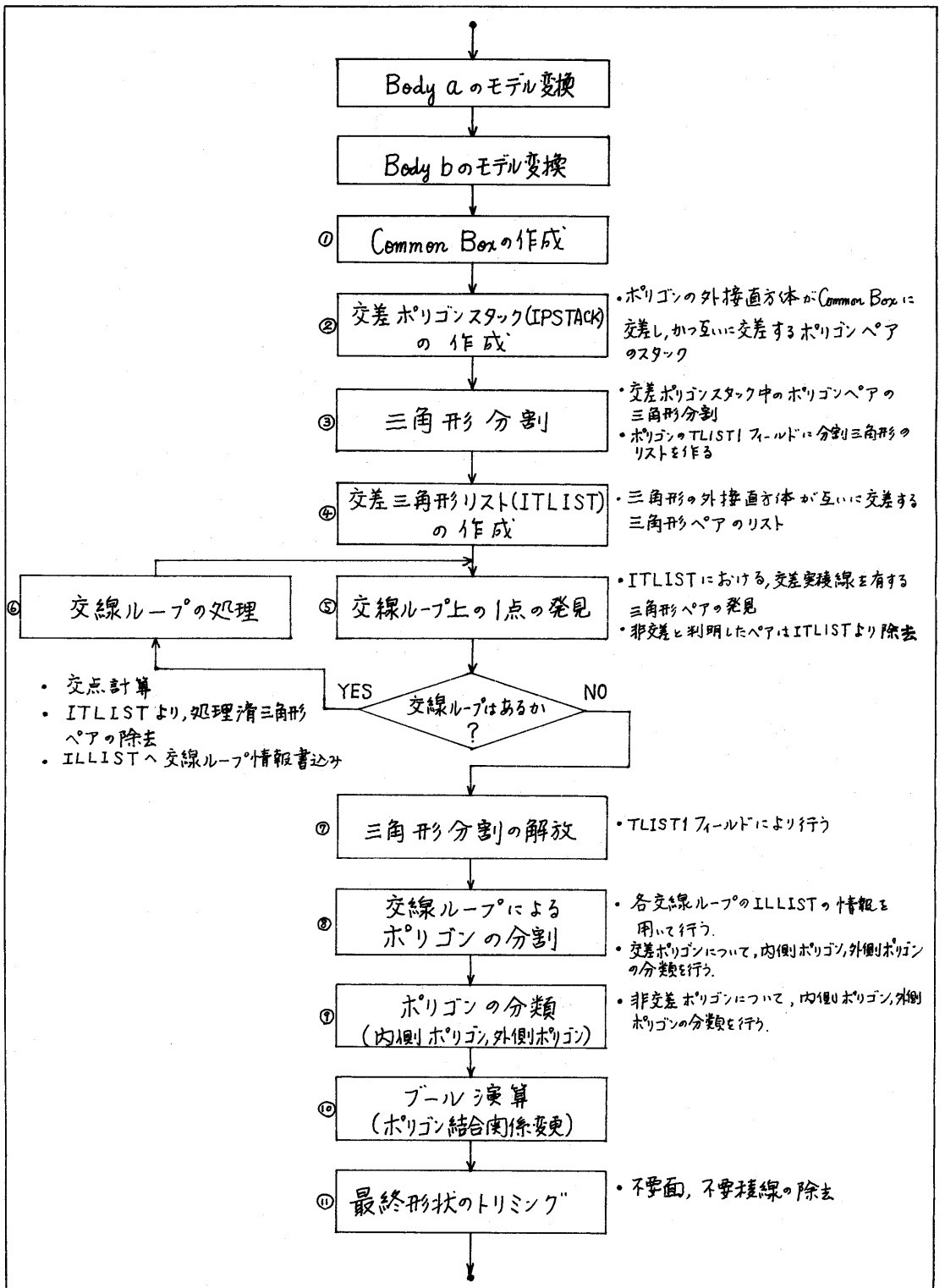

図5　多面体ブール演算プロセッサのアルゴリズム

する他方の三角形が新t_r, また現t_sはそのまゝ新t_sとなる.

(3-3) もし交差しなければ, (図7-(b)), t_sの稜線のいずれか一つがt_rに交差する. そこでt_sの3頂点V_1, V_2, V_3に関し, t_rに対するS_1, S_2, S_3を求め, t_rに対し交点を持つ稜線を新e_rとする. 現t_sに対し, 新e_rを共有する他方の三角形が新t_r. また現t_rが新t_sとなる.

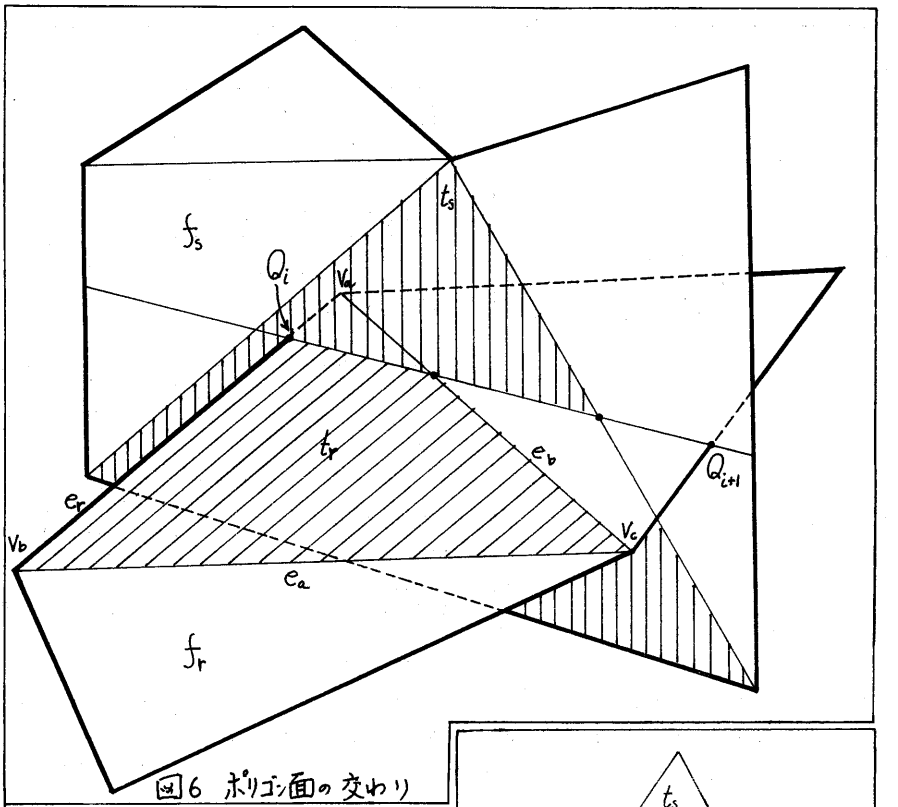

図6 ポリゴン面の交わり

(3-2), (3-3)において, 新e_rを求める際に得られたSを, 新e_rのS_a, S_bとして利用する.

(3-4) 新e_rが仮想稜線(三角形分割により生じた稜線)であれば, (3-1)に戻って同上の処理を繰返す.

(3-5) 新e_rが実稜線であれば, 式(1)により, これまで得られているSを用いて交点Q_{i+1}が求まる.

(3-6) 以上の処理を始点Q_1に戻るまで繰返す.

上述の交線ループの処理アルゴリズムにおいて,

- 交線上の三角形ペアは, 処理の都度, ITLISTより除外しておく.
- 交線ループを構成する, 二つの多面体のポリゴン, 稜線, 頂点, 交点座標値等は交線情報リスト(ILLIST)に格納しておく.

(4) 残存交線ループの発見

ITLISTに三角形ペアが存在しなければ, 未処理の交線ループはない. もし三角形ペアが残っていて, 交差実稜線を有する三角形ペアがあれば未処理の交線ループが存在することになる.

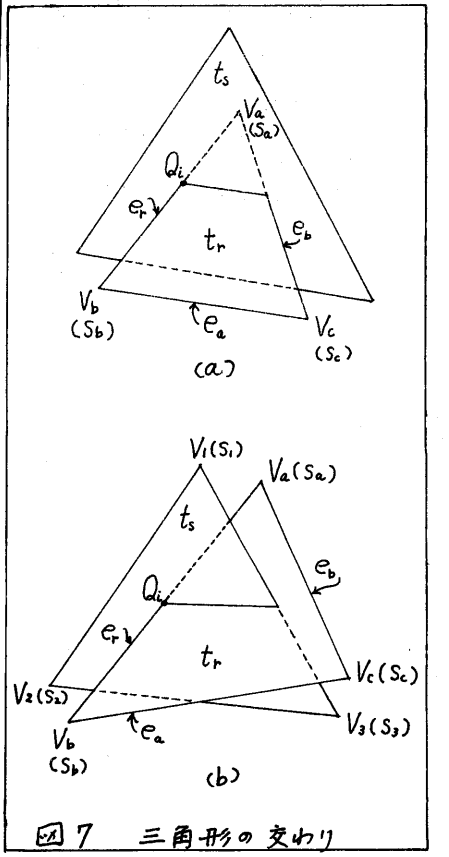

図7 三角形の交わり

(5) 三角形分割の解放
交線ループにより各ポリゴンを分割するための情報はすべて，ILLISTに書き込まれている．そこで，この段階で三角形分割を解放する．これはTLISTリストの情報により行う(図5の⑪)．

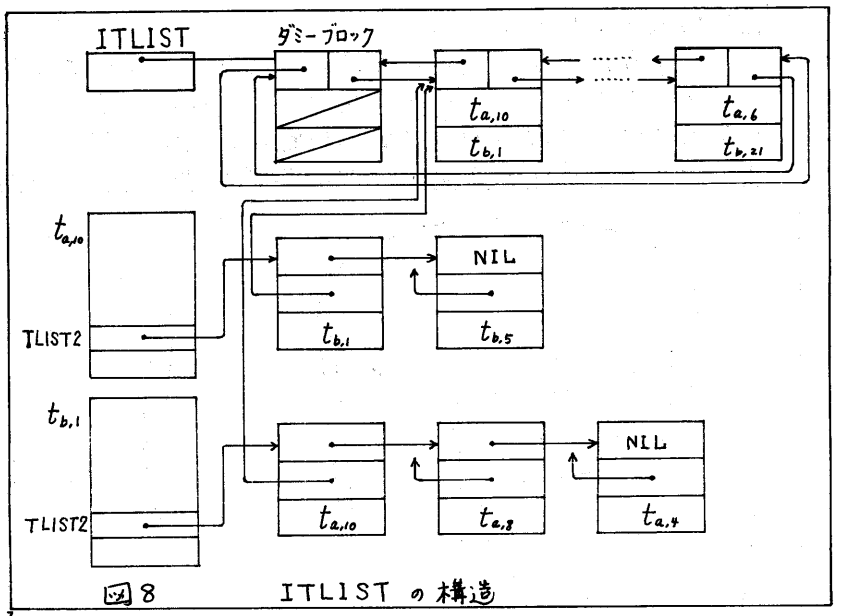

図8　ITLIST の構造

(6) 交線ループによるポリゴンの分割

ILLISTに書き込まれている情報により，交差ポリゴンを分割する．また交差ポリゴンについては，内側ポリゴンか，外側ポリゴンかは分割過程で判別できるのでこの段階で処理する(図5の⑫)．

ポリゴンの種類＼演算		Body a ∪ Body b	Body a ∩ Body b	Body a − Body b
Body a	外側ポリゴン	残す	除く	残す
	内側ポリゴン	除く	残す	除く
Body b	外側ポリゴン	残す	除く	除く
	内側ポリゴン	除く	残す	ポリゴンの向きを反転して残す

表1　ブール演算による各ポリゴンの処理

(7) ポリゴンの分類
交差に関与しないポリゴンについて，内側ポリゴンか，外側ポリゴンかを判定する．この判定は，ポリゴン面上の1点より無限半直線を作り，これと相手の多面体のポリゴンとの交差回数を調べることにより行える．すなわち，交差回数が偶数の場合(零の場合も含める)は外側ポリゴン，また奇数の場合は内側ポリゴンである(図5の⑬)．

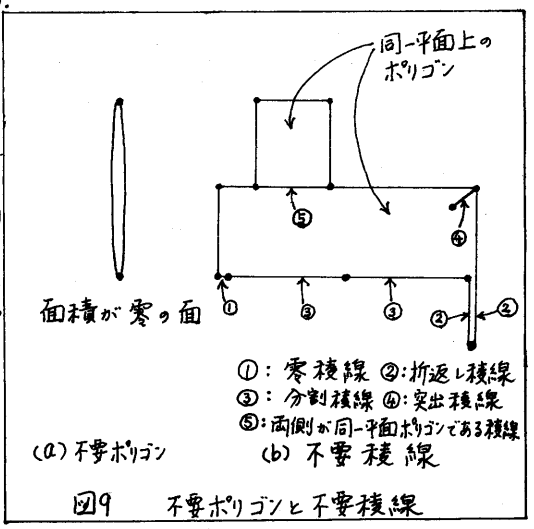

図9　不要ポリゴンと不要稜線

(8) ブール演算
指定された演算が，Body a ∪ Body b，Body a ∩ Body b，Body a − Body b かに従い，表1に示す通りに各ポリゴンの結合を変更し，不必要なポリゴンは除去する(図5の⑭)．

(9) 最終形状のトリミング

多面体相互のブール演算処理後図9に示すような不必要なポリゴンや稜線が含まれることがある.図9(a)は,面積が零のポリゴン,図9(b)は,長さが零の稜線,折返し稜線,分割稜線,突出稜線及び両側のポリゴンが同一の平面上のポリゴンである稜線である(図5の⑪). これらの不要ポリゴン,不要稜線を除去する.

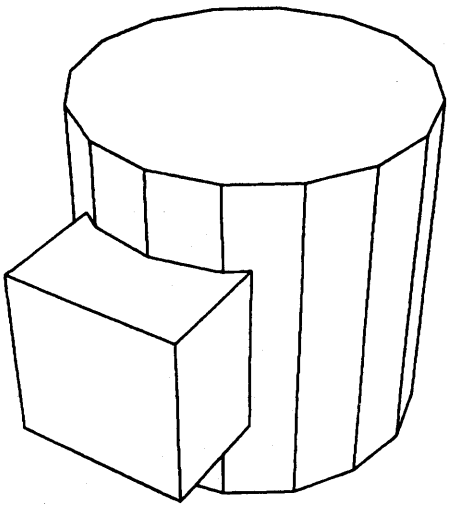

図10-(a) ∪演算

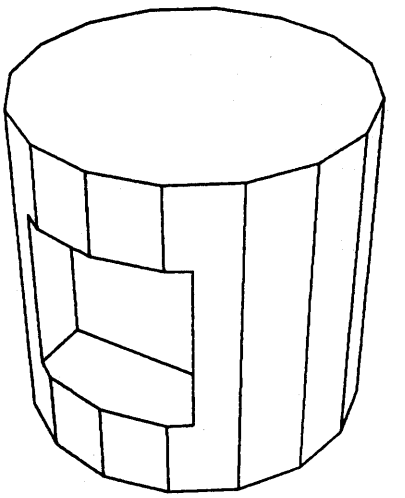

図10-(b) −演算

以上に,FREEDOM-IIにおける形状処理の基本部分について述べた.

今後は,三角形分割を用いたブール演算プロセッサを自由曲面に対して応用してみたい.

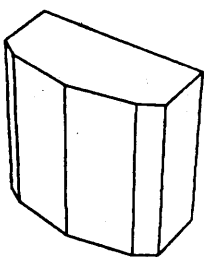

図10-(c) ∩演算

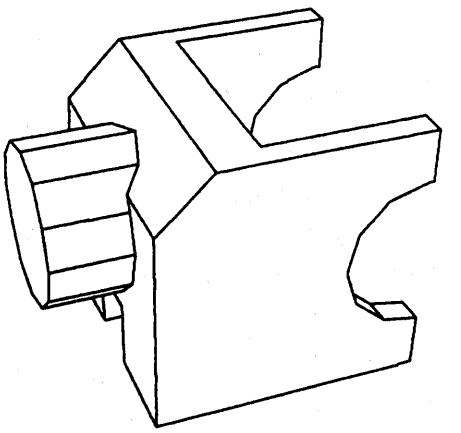

図11 I設計例

参考文献

1) Braid, I.C., "The Synthesis of Solids Bounded by Many Faces," Comm. of ACM, April 1975.

2) Hosaka, et al., "A Unified Method for processing Polyhedra," Proc. of 1974 IFIP Conference, 1975.

3) Baumgart, B.G., "Geometric Modeling for Computer Vision," Computer Science, Stanford Univ. 1974.