投影面スケッチによるソリッドモデル入力法†

福井幸男†† 廣谷豊史†††
大平智弘†† 岸 義樹††

3次元形状を表現するソリッドモデルをコンピュータ上に定義する手法には，基本形状を用意してそれらに集合演算を行う方法，2次元形状を定義し空間的に移動させたときの軌跡として定義するスイープ，オイラー式を満たしながら面や稜線を変更してゆくオイラーオペレーション等の方法がある．これらは主としてコマンド言語によって実行されるが，形状と言語との対応はユーザにとって理解しにくい．本論文では，3次元形状を2次元投影面に投影する投影面上で，投影図を先ず定義し，逆投影を行って3次元空間内に最終的にソリッドモデルを定義する対話的手法を提案する．本手法の特徴は，タブレット等で描かれた2次元図形を一面ごとに逆投影して3次元空間内に，面の傾きや位置を入力するとともに，面・稜線・頂点等の形状要素間の接続関係も自動的に生成する．逆投影によって面の傾き等を一義的に決定させるために必要な条件は，既に定義済の隣接面から得るかまたは，投影方向（視線方向）のベクトルと面の法線ベクトルとの内積の絶対値を最大とすること，として得ている．実時間で認識処理し稜線が閉じた段階で面が認識され，面がすべて閉じた段階で形状全体のデータ構造が完成される．実験により，有効性・簡便さが確認された．

1. まえがき

3次元形状をコンピュータ上に表現するソリッドモデルの表現形式の一つに，境界表現（Boundary Representation）があり[1]，本論文はこの表現形式の多面体モデルの入力に関するものである．境界表現モデルは形状表面を覆う各面の和集合として3次元形状を定義している．多面体の場合データ構造上では，特定の面を指定すれば，この面を構成する各稜や各頂点が参照でき，また任意の頂点に接続された稜線や面が参照できる．面・稜・頂点等の各形状要素間では，検索効率上の差はあっても，特定の要素からそれに関連する他の要素を参照できる構造を保持している．形状要素間のこの接続構造を，トポロジと呼ぶことにする．また，各面は位置や傾きを定義するために平面方程式をもち，頂点は3次元座標値をもつ．座標系に直接依存したこれらの幾何学量をジオメトリと呼ぶことにする．境界表現モデルでは，トポロジとジオメトリの両方をもっている．

次に，このモデルの入力方法をみると，主として2通りの方法が使われている[2)-4)]．その一つは集合演算を用いる方法である．これはあらかじめ用意された基本立体を，集合演算で結合させたり削除を行い，目的の形状を生成させる．この過程は，同時に素材の加工工程のシミュレーションにもなりうる手法である．他の一つはいわゆるスイープと呼ばれる手法である．この典型的な例としては，多角柱を定義するときに，先ず底面となる多角形を定義し，次にこの底面を面と直角方向に平行移動させたときの，軌跡として多角柱を定義し，これに基づいてデータ構造を生成するものである．段差・突起等のある複雑な形状に対しては，面内の特定の領域にこの操作を複数回適用すれば生成可能である．このような入力操作は，主としてメニュー等を用いたコマンド言語により実行される場合が多いが，一般的には言語と形状との対応は人間になじみにくい．図形を自然言語のみで伝達する実験では，言語だけでは困難なことが確認されている[5)]．したがって，図形の直接入力を主としたコマンド言語との併用は，重要と考えられる．

一方，直接図形からの形状入力の方法としては，複数の2次元投影図をデジタイズしてゆく方法[6),7)]，あらかじめ2次元投影された座標軸に沿って描くことにより行う方法[8),9)]等があるが主としてジオメトリの入力が主でトポロジ入力は不十分である．また三面図を描く過程をオンラインでデータ入力に使い，同時に集合演算も図面上で指示してソリッドモデルを生成するコマンド列を自動生成させる手法もある[10)]．

ところで思考過程において形状を考察する際は，人

† An Input Method for Solid Models by Sketches on Projection Planes by YUKIO FUKUI (Perceptual Functions Division, Industrial Products Research Institute, M. I. T. I.), TOYOTOSHI HIROTANI (CadVenture Inc.), TOMOHIRO OHIRA and YOSHIKI KISHI (Perceptual Functions Division, Industrial Products Research Institute, M. I. T. I.).
†† 工業技術院製品科学研究所情報機能課
††† (株)キャドベンチャー

間のもつ視覚機能に大きく影響されると考えられる．すなわち，眼球網膜に映るのは2次元投影像であり，3次元に組み立てるのは大脳である．また新たに3次元形状を創造する過程は大脳と視覚によるインタラクションのくり返しである．大脳からの指令は視覚で2次元的に，そして様々な方向からの確認により行われ，形状認識の段階では視覚から大脳へ2次元投影像を伝えて大脳で3次元処理している．

以上の考察から，2次元投影像を与えて3次元構造を組み立てる入力法は，人間の思考形態に類似した，換言すればなじみやすい手法になるのではないかと考えて本論文の手法による実験を行った．ここでは一般多面体形状を投影面上で線画で入力することによりソリッドモデルのデータ構造を生成する手法について述べる．本手法によるデータ生成過程ではデータ構造は完全ではないが，拡張オイラー式を満たす変形のみで操作を行うため[11]，最終的には完全なトポロジが得られる．ジオメトリのチェック，例えば物体自身の面同士の干渉チェック等は自動的には行っていない．以下第2章では3次元変換，第3章ではトポロジの生成手法を説明し，第4章で実験結果を示す．

2. 逆投影によるジオメトリの変換

2.1 変換原理

本手法は3次元空間を正射影した2次元平面上に3次元形状の投影図を与え，これを3次元空間に「逆投影」させて3次元形状を定義するのが基本原理である．使う表示装置は通常のグラフィックディスプレイで表示面を投影面かつ入力面とし，カーソルまたはタブレットによって入力する．入力は投影面上で稜線を描いて与えてゆき，稜線が閉じた段階で，面の平面性のチェックが行われ，面が登録される．隣接面に沿って入力を行い，必要に応じて視線方向（投影方向）を変更させ同様に稜線を入力してゆく．フリーハンド入力のため例えば正確に平行な面等の入力は困難であるが，ある閾値以下の傾きや，座標値の値は自動修正できる．稜線によって物体表面の面を入力してゆき，面が定義されていない領域がなくなった段階で3次元形状のデータが完成される．なお，面内に存在する穴等の作成に関しては，3.2 節で概略を述べる．

一般に投影面上に描かれた閉じた領域に対応する3次元空間内の面は，**図1**に示すように傾きに関し2，位置に関し1の計3の自由度をもち，一義的に定義できない．これに対し次に示す基準で自由度をなくして逆投影による一義的な定義を行うことにする．

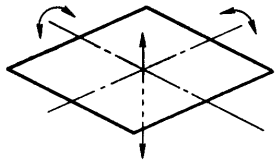

図1 面の自由度
Fig. 1 Degrees of freedom of a plane.

（1） 位置に関する自由度について

既に3次元空間内に定義されている面に隣接する面を原則的に入力するものとする．隣接する面がなければ，これは最初の面を意味することになり，絶対位置は形状定義に無関係なので，システムのもつ既定値を与えている．

（2） 傾きに関する自由度について

新しく定義する面に隣接する面の数によって，傾きの自由度は変化する．傾きの自由度がある場合は，視線方向ベクトル $\boldsymbol{v}$ と法線ベクトル $\boldsymbol{n}$ との内積の絶対値 $|\boldsymbol{v}\cdot\boldsymbol{n}|$ が最大となる方向に設定する．ここで，$\boldsymbol{v}$ は投影面に垂直な方向（投影方向）で「向き」が投影面から形状を定義する空間へ向いたベクトルで，$\boldsymbol{n}$ は新しく定義される面の外向き法線ベクトルである．したがって自由度のある面の入力では，形状に対する視線方向 $\boldsymbol{v}$ が面の傾きを決定させることになるので，$\boldsymbol{v}$ を描画とは独立に与える必要がある．換言すれば定義しようとする形状の投影方向を与えなければならない．$\boldsymbol{n}$ は具体的にはシステムによって次のいずれかが選択される．

（イ） 隣接面がない（自由度 3）の場合

$\boldsymbol{n}=-\boldsymbol{v}$ とする（**図2**（a）参照）．

（ロ） 隣接面が一つ（自由度1）の場合

既に定義されている面と新しく定義される面が接する位置の稜線（これを共通稜線と以降呼ぶことにする）は空間内に固定されているので，この稜線が拘束条件となり自由度は1となる．この場合，図2（b）に示すように逆投影面積が最小となる方向，すなわち投影面積が最大となる方向に面の傾きが設定される．

（ハ） 隣接面が2以上の場合

定義される面と，複数個の隣接面との共通稜線が，許容誤差範囲内で同一平面上に含まれる場合，定義される面をこの平面に一致させる（図2（c））．

2.2 面の変換

前節で示した面の傾き，位置の設定等を方程式で表現する．3次元空間に定義しようとする形状に固有の

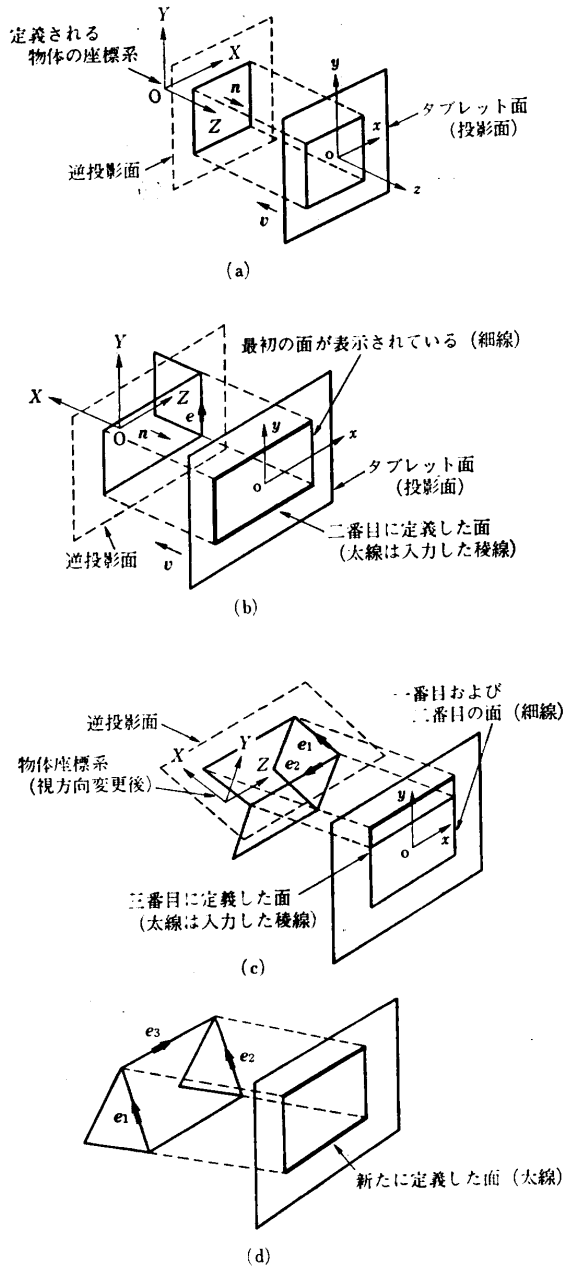

図 2　面の定義方法
（a） 自由度3の場合，（b） 自由度1の場合，（c），（d） 自由度0の場合
Fig. 2　Definition method of a face.
(a) in case of 3 degrees of freedom, (b) in case of 1 degree of freedom, (c), (d) in case of 0 degree of freedom.

右手座標系 O-XYZ（これを物体座標系と呼ぶ）を設定し，別の右手座標系（視座標系とよぶ）o-xyz を投影面を xy 平面に一致させ視線方向が $-z$ 方向と一致するように設定する（図 2(a)参照）．物体座標系の原点Oを，視座標系 o-xyz の z 軸上で原点から離れた負の位置に初期設定する．ユーザは表示装置の画面をみるが，この面は常に視座標系の xy 平面に固定されている．したがって投影方向を変化させる場合，見かけ上画面内の物体座標系が回転することになる．このような関連をもたせた二つの座標系，物体座標系 O-XYZ および視座標系 o-xyz のもとで表現された同一平面の方程式をそれぞれ，

$$\left.\begin{aligned} &AX+BY+CZ+D=0,\\ &A^2+B^2+C^2=1\\ &ax+by+cz+d=0,\\ &a^2+b^2+c^2=1 \end{aligned}\right\} \quad (1)$$

とすると次の関係がある[12]．

$$\left.\begin{aligned} &(x'y'z'w')=(XYZW)M\\ &(xyz)=(x'y'z')W/w'\\ &(a\ b\ c\ d/W)=(A\ B\ C\ D/W)(M^{-1})^{\mathrm{T}} \end{aligned}\right\} \quad (2)$$

ここで M は座標系間の変換マトリックス，$(\ )^{\mathrm{T}}$ は転置，W は座標系の基準値* を表す．

定義される面の方程式は次のように定まる．

（1） 自由度が3のとき

$$a=b=0,\ c=1,\ d=(\text{既定値}) \quad (3)$$

（2） 自由度が1のとき

定義される面内に既に定義済の面に含まれる共通稜線があるため，係数 A, B, C が定まれば残りの係数 D は求まる．共通稜線に沿った単位ベクトルおよび視線方向ベクトルを，物体座標系でそれぞれ $\boldsymbol{e}$, $\boldsymbol{v}$ で表すと法線ベクトル $\boldsymbol{n}$，係数 D は

$$\left.\begin{aligned} &\boldsymbol{n}=(ABC)=k\boldsymbol{e}\times(\boldsymbol{e}\times\boldsymbol{v}),\ k=\pm1\\ &D=-\boldsymbol{n}\cdot\boldsymbol{x}_0 \end{aligned}\right\} \quad (4)$$

ここで複号は次章で説明する方法により $\boldsymbol{n}$ が物体の外側を向くようにいずれかに決定される．また $\boldsymbol{x}_0$ は共通稜線上の一点の物体座標系での座標ベクトルである．

（3） 自由度が0のとき

定義される面内に既に共通稜線が2個以上存在する場合でいま2個の場合について次の2通りに分けて考える．3個以上の場合でも同様な処理を行う．

（イ） 共通稜線2個が接続されているとき（図 2（c）参照）

これらの稜線に沿った単位ベクトルをそれぞれ $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ とすると

$$\boldsymbol{n}=k(\boldsymbol{e}_1\times\boldsymbol{e}_2),\ k=\pm1 \quad (5)$$

* 座標値を実数表現する場合 $W=1$，整数表現のとき最大数を W として X, Y, Z が最大数を越える場合に W とともに値を小さくしてオーバフローを防ぐ．

複号については(4)と同様にいずれかに決定される.

(ロ) 共通稜線2個が離れているとき (図 2(d)参照)

この場合は許容誤差範囲内で同一平面上にあるか,あるいは互いにねじれの位置にあるかをスカラ三重積を用いてチェックする. 2個の稜線の端点同士を結ぶ方向の単位ベクトルを e_3 とおくと

$$e_1 \cdot (e_2 \times e_3) \leqq (\text{許容範囲}) \qquad (6)$$

ならば n を(5)式より求める. Dは(4)式で求まる.

稜線が閉じて領域ができても,その内部の複数個の共通稜線がいずれか一組でも(6)式を満たさなければ新しい面を定義しない. 面内に存在する穴等の生成では共通稜線はない.

2.3 頂点の変換

前節で稜線の連なりから面の式を求めたが(ただし面の向きについては後述),ここではその面内に存在する頂点の物体座標系での座標値 (XYZ) を求める. 視座標系での座標値 (xyz) のうち x, y は与えられたもので既知であり,面の方程式の係数 A, B, C, D (または a, b, c, d) が前節で求まっており,視線方向が決まれば座標系間の変換マトリックス M が定まる. したがって(1)式および(2)式から未知の (XYZ) を次式で求める.

$$\left.\begin{aligned} &(XYZ)=(X'Y'Z')W/W' \\ &(X'Y'Z'W')=(xyzW)M^{-1} \\ &z=(A\ B\ C\ D/W)(M^{-1})^{\mathrm{T}}(x\ y\ 0\ W)^{\mathrm{T}} \\ &\quad /(A\ B\ C\ D/W)(M^{-1})^{\mathrm{T}}(0\ 0\ -1\ 0)^{\mathrm{T}} \end{aligned}\right\} \qquad (7)$$

3. トポロジの生成

データ構造上,一つの面の外周を構成する連続した稜線の連なりが容易に検索できるが,この稜線の連なりをループと呼ぶことにする. また面内に穴または突起が存在する場合,この周囲にも稜線の連なりができるが,これを面の外周のループと区別して特に穴ループと呼ぶことにする. ループの方向として,面の外側から見たときに反時計方向に面の外周を一周する方向と定義した. 穴ループに関しては逆方向にとる. 前章で投影面上での領域から3次元空間内の平面を生成する方法について述べたが,面の向き(係数 k で表現した)は未定のままで,これはループの方向で決定される. したがってループを正しく抽出する必要があり,抽出について述べる.

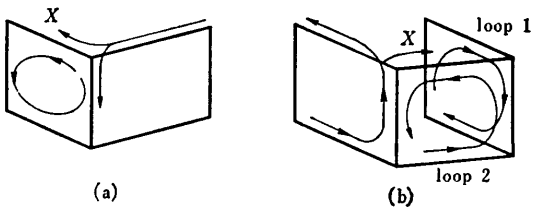

図 3 分岐点での追跡方向選択
(a) 二度同一方向への追跡禁止, (b) 分岐点前後で逆方向ループは同一であってはならない
Fig. 3 Selection of tracking direction at a junction. (a) Not allowed to track twice, (b) Loop of opposite directions must not be the same before and after the junction.

3.1 ループの生成

ループ抽出アルゴリズムは,描いてゆく稜線が閉じた段階で起動される. ループ抽出の過程は以下のとおりである.

(1) 新しい稜線から始まって稜線に沿って双方向に追跡開始する.

(2) 追跡が分岐点にくると次の規則に合致するすべての稜線が次の追跡対象候補となり,一つずつ追跡してゆく.

(イ) 稜線は同じ方向に以前に追跡されていてはならない (図 3(a)参照).

(ロ) 分岐点の前後の稜線について現在と逆方向の追跡ループが,分岐点の前後で同一であってはならない (図 3(b)参照).

(3) 最初の稜線にもどるとループ候補として登録する.

(4) すべてのループ候補が得られた後,各ループ候補を構成する稜線のうち,共通稜線が何個あるか調べて以下の処理を行う.

(イ) 共通稜線がない場合

視線方向から見て反時計まわりのループを登録する.

(ロ) 共通稜線が2個以上の場合

共通稜線がすべて許容誤差範囲内で同一平面上[13]にあるループを登録する.

穴ループの場合は,どの面に含まれるかが既知である(次節で説明する)から面の外周のループと逆方向にとる. k の値としては,ループに沿って一周したときに右ねじの進む向きのベクトル u と,面の外向き法線ベクトル n との内積 $u \cdot n$ が正の値になるように, ± 1 のいずれかを選択する.

右ねじの進む方向ベクトル u は,次式で求める.

$$\boldsymbol{u}=\sum_{i=1}^{n}(\boldsymbol{x}_i\times\boldsymbol{x}_j) \qquad (8)$$

ここで $\boldsymbol{x}_i$ ($i=1,2,\cdots,n$; n は一つのループを構成する稜線の数) はループに沿って付番された頂点の位置ベクトルとし，$j=i+1\,(i\neq n)$，$j=1\,(i=n)$ とする．

3.2 オイラーオペレータ駆動

本システムではユーザが多面体の稜線を一つずつ各面ごとに与えてゆくと，局所変形オペレータ（オイラーオペレータ）が起動されて，面その他のデータを自動生成してゆく．いわば，オイラーオペレーションをグラフィカルに実行するシステムとみなすことができる．

オイラーオペレーションは次に示す拡張オイラー式を満たすように各変数を操作する[14]．ただし物体内部の空洞はつくらないことにする．

$$v-e+f-l+2p-2b=0 \qquad (9)$$

v, e, f, l, p, b はそれぞれ頂点 (Vertex)，稜線 (Edge)，面 (Face)，穴ループ (Loop)，貫通穴 (Pass)，物体 (Body) の数を表す．

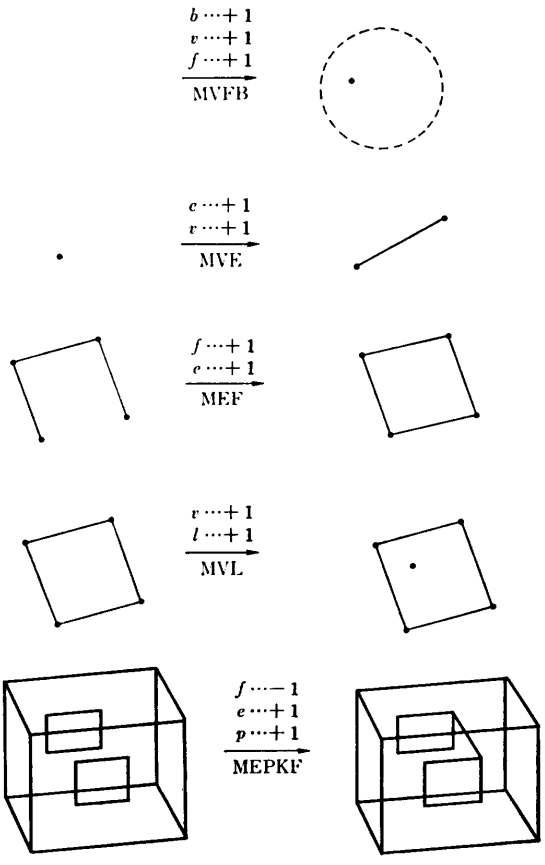

図 4 基本変形操作
Fig. 4 Fundamental set of operation.

ここでは次の五つの基本操作およびその逆操作を用いている (**図 4** 参照)．(+は増，−は減を表す．)

MVFB (Make Vertex Face Body)…$+v$, $+f$, $+b$
MVE (Make Vertex Edge) ……$+v$, $+e$
MEF (Make Edge Face) ……$+e$, $+f$
MVL (Make Vertex Loop) ……$+v$, $+l$
MEPKF (Make Edge Pass Kill Face)…$+e$, $+p$, $-f$

投影面上でユーザが入力した稜線が入力済の稜線に接続されていれば，逆投影された3次元空間内でも接続されているとみなすことにする．ただし複数個の接続候補があれば，投影面に最も近い，ユーザから見て手前の候補を採用することにする．これにより，新しく稜線が入力されたとき，接続関係から上記のどの基本操作を起動させるかがシステム側で判断できることになる．ただし効率上，面内に穴または突起のための穴ループをつくるとき (MVL が起動されるとき) のみ，ユーザが直前にこのことをシステムに知らせることにしている．なぜならば，このときは次に入力される稜線がどの面内に含まれるかシステムが直ちに見つける必要があるからだ．面の探索範囲は投影されている面のうち（奥行方向を含め左右上下に視座標系にウィンドウを設け，このウィンドウ内部の形状のみが表示されることにする)，ユーザ側を向いた面すなわち $\boldsymbol{n}\cdot\boldsymbol{v}<0$ を満足する面に絞っている．稜線がどの面内に含まれるかは，投影面上で両端点がどの面に含まれるか2次元的に調べている．

4. 実験結果

本入力手法によれば，ユーザは3次元空間内に稜線を描いてゆく感覚で多面体の定義ができる．ただしタブレットの2次元から3次元空間に意図したとおりに逆投影を行うために，次のことを常に念頭に置かなければならない．

(1) 基本的に1面ずつ，既に定義されている面に隣接した面を追加しながら，描かなければならない．一つの面を完成させないで，途中で他の面を描くことはできない．これは面単位で描き終った時点に，逆投影で3次元座標値を求めるためである．

(2) 入力しようとする面の自由度を考慮しなければならない．特に自由度が1の場合，すなわち既に入力済の面に一辺だけ接する新しい面を描くときには，次の要領で視線方向を設定する必要がある．先ず隣接

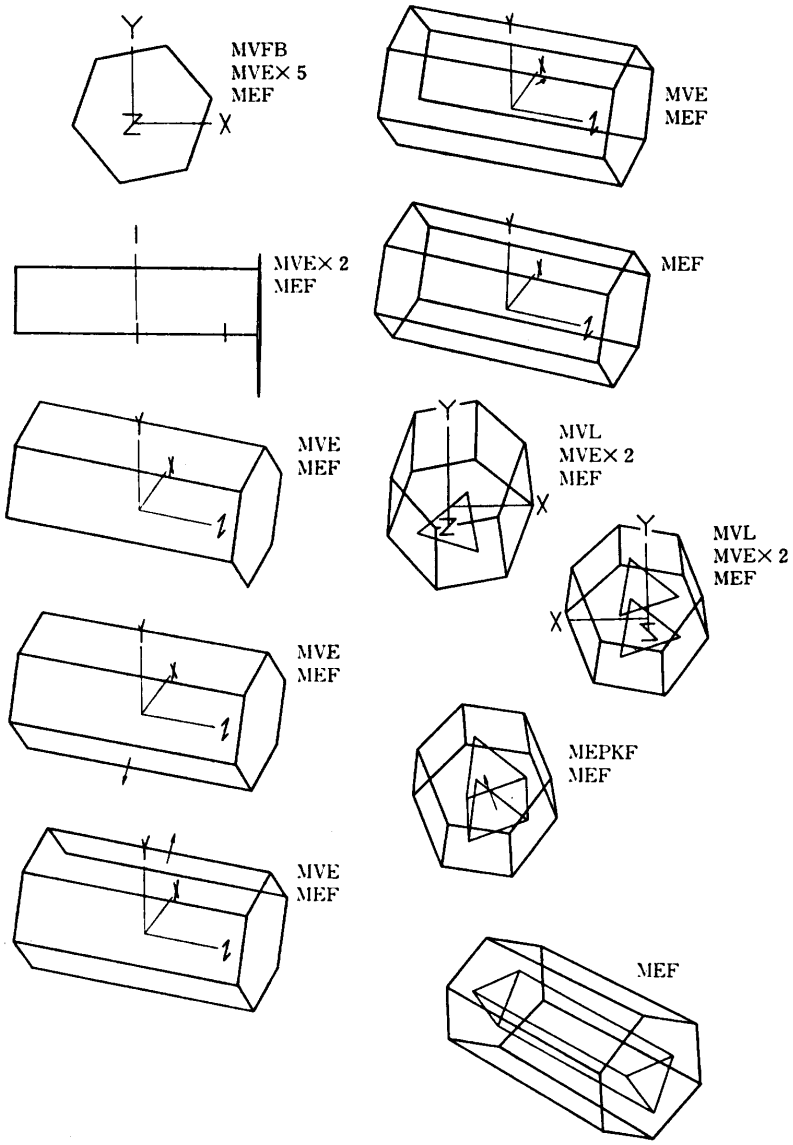

図 5 穴のある形状の作成ステップと駆動されるオイラーオペレータ
Fig. 5 Stepwise construction of a model with a hole by Euler operators.

面に接する辺を回転軸として，新しく入力する面を仮想的に揺動させる．このとき，画面上への投影面積が最大になる傾きが目的の傾きとなるように回転軸に対する視線方向（投影方向）を設定してから入力を行う．自由度3の場合は，常に視線方向に垂直な平面として入力され，自由度0の場合は，視線方向は入力面の傾きに影響を及ぼさない．

図5に本手法による六角柱定義の経過を，一つの面の入力終了時点での投影方向を保ったまま，投影面の状態と起動されるオイラーオペレータの種類を示す．矢印は新入力面とその向きを表示して，ユーザに再確認を促す．**図6**は不規則図形の入力経過の途中の一部を任意の方向からみたものと，入力完了後隠れ線処理した形状を示す．

5. むすび

本論文では穴・突起を持つ多面体を，投影面上で定義することにより，逆に3次元空間に逆投影してソリッドモデルを生成する手法について述べた．重要な点は以下のとおりである．

（1） 投影面上で物体を面単位で稜線を描いて定義することにより，面単位で3次元変換する．

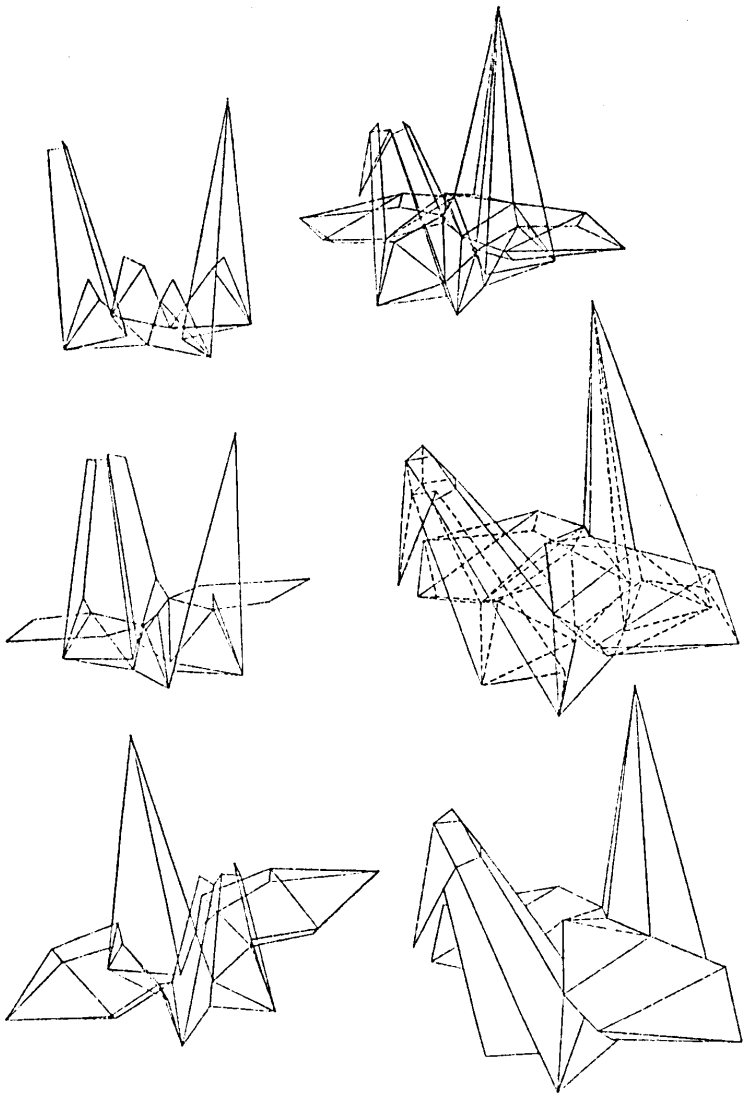

図 6 不規則形状定義過程の一部分
Fig. 6 A part of defining steps of irregular shape model.

（2） 一義的に3次元変換するための手掛りは，投影方向と，定義済の隣接面の保持する情報を使う．

（3） 稜線が定義されるたびに，適切なオイラーオペレータが駆動されてデータを更新してゆく．

（4） すべての面の定義が終了した時点で，境界表現形式のソリッドモデルが生成される．

なお本手法では，正確な寸法をどの時点で与えるかについては言及していない．これは別途系統的な検討を要する課題と考えられる．

参 考 文 献

1) Requicha, A. A. G. and Voelcker, H. B.: Solid Modeling—A Historical Summary and Contemporary Assessment, *IEEE Computer Graphics and Applications*, Vol. 2, No. 2, pp. 9-24 (1982).
2) 穂坂 衛，木村文彦：機械設計自動化のための幾何モデル生成処理システム，日本機械学会論文集（第3部），Vol. 44, No. 378, pp. 661-669 (1978).
3) 沖野教郎：形状モデリング，精密機械，Vol. 47, No. 11, pp. 1334-1342 (1981).
4) 穂坂 衛：情報処理技術と CAD/CAM，精密機械，Vol. 47, No. 11, pp. 1319-1324 (1981).
5) 廣谷豊史，渥美浩章：線図形と言語表現との関係についての一考察，日本建築学会論文報告集，Vol. 307, pp. 93-101 (1981).

6) Thornton, R. W.: Interactive Modeling in Three Dimensions through Two Dimensional Windows, *Proceedings CAD 78*, IPC, pp. 204-211, Business Press, Surrey, U. K. (1978).
7) Lafue, G.: A Theorem Prover for Recognizing 2-D Representations of 3-D Objects, in Latombe, J.C. (ed.) *Artificial Intelligence and Pattern Recognition in Computer Aided Design*, pp. 391-401, North-Holland, Amsterdam (1978).
8) Liardet, M., Holmes, C. and Rosenthal, D.: Input to CAD Systems—Two Practical Examples, in ibid, pp. 403-427.
9) 廣谷豊史，福井幸男，大平智弘：3次元図形の入力の研究，日本建築学会第3回電算機利用シンポジウム梗概集，pp. 343-347 (1981).
10) 穂坂 衛，木村文彦：手書き図面入力とその幾何モデル生成への応用について，日本機械学会論文集C編，Vol. 45, No. 389, pp. 75-82 (1979).
11) 沖野教郎：自動設計の方法論，p. 86，養賢堂，東京 (1982).
12) Newman, W. H. and Sproull, R. F.: *Principles of Interactive Computer Graphics*, p. 339, McGraw-Hill Kogakusha, Tokyo (1979).
13) Faux, I. D. and Pratt, M. J.: *Computational Geometry for Design and Manufacture*, p. 65, Ellis Horwood, New York (1981).
14) Braid, I. C., Hillyard, R. C. and Stroud, I. A.: Stepwise Construction of Polyhedra in Geometric Modelling, in Brodlie, K. W. (ed.) *Mathematical Methods in Computer Graphics and Design*, pp. 123-141, Academic Press, London (1980).

(昭和58年7月 1 日受付)
(昭和60年6月20日採録)