

等角投影法を用いた 簡易な立体形状入力システムの試作第2報

————— 曲面の入力アルゴリズム —————

中嶋 正之 安居院 猛 羽生田 千春

東京工業大学・工学部・像情報工学研究施設

3次元のコンピュータ・グラフィクスにおいては，そのデータの
入力方法が大きな問題となる．筆者らは先に，3次元空間を2次元
平面上に表現するのに最も簡便な方法である，等角投影法を用いて，
主として多面体近似された立体を入力する方法について報告した．
等角投影法の特徴としては，主軸方向の長さが一定であり，計測が
できること，主軸によって作られる平面に平行な面では，楕円が同
じ縦横比で表現できることなどが挙げられ，不等角投影法などに比
較して作成が容易である．ここでは，さらに第2報として，等角投
影法を用いて曲面形状を入力するアルゴリズムおよび立体入力例に
ついて報告する．

Solid Object Input System by Conformal Projection Method
Part II

————— Curved Surface input algorithm —————

Masayuki NAKAJIMA Takeshi AGUI Chiharu HANYUDA
Imaging Science and Engineering Laboratory,
Tokyo Institute of Technology.

One of the important themes in computer graphics is the method of designing an easy data-input system of 3D objects. We proposed an input system of polyhedron using the projective plane by a conformal projection method. In this paper, we propose a new input algorithm of curved solid using the projective plane.

First, a basic data-input algorithm of 3D objects by this projection and input examples are shown. Many points on 3D objects are interactively constructed on computer by this algorithm. Next, an input algorithm of curved surface are described. Finally, data-input examples and display techniques are shown.

1. はじめに

3次元物体を対象とするコンピュータ・グラフィックスの分野においては、対象立体データをどのように効率よく入力するかが大きなテーマとなっている。

まず、データを入力しようとする物体が、実際に存在している場合、データ入力の方法としては、その寸法を計測して、キーボードから入力する方法や、3次元ディジタイザ（座標読み取り装置）による方法があげられる¹⁾。

また、正視投影図によって物体が表現されている場合、キーボードによる入力や、平面ごとに2次元のディジタイザによって入力する方法をとることが多い。

しかし、いずれの方式も入力する人間側の労力の負担が大きい。他の方法としては、3面図を入力し、人工知能的に3次元立体を再構成する方法も試みられているが、実用化レベルには達していない²⁾。

そこで、筆者らは先に、比較的ユーザに負担をかけないマンマシンインターフェースの改善を目的として、軸測投影法の1種である等角投影法を用いて2次元のディジタイザから直接3次元立体データを入力する方法について提案した³⁾。

本システムの利用より極めて複雑な形状を有する建築物も比較的容易に計算機へ入力することが可能となることが明らかとなった。

しかし文献3では、対象とする立体が主として、多角形近似された平面から構成されたものに限定されていた。そこで、本報告では、第2報として、等角投影法を用いた曲面の入力アルゴリズムについて提案する。なお使用する記号、用語は全て文献3に示されているものをそのまま使用する。

2. 等角投影法

等角投影法では、三次元空間上の、直交軸 X, Y, Z 軸は、投影後にお互いに等しい角度で交わる。これらを主軸とよび、ここでは、それぞれを X_p, Y_p, Z_p 軸とする。

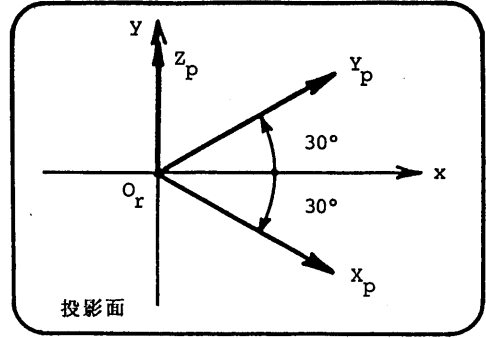
図1に示すように、投影後の X_p, Y_p 軸が、投影面の x 軸に対して、それぞれ $30[\text{deg}]$ の角をなすことから、投影面に対する Z 軸の傾き ψ を求める。

ここで、 X 軸上の点 $X_1(x_1, 0, 0)$ 、および Y 軸上の点 $Y_1(0, y_1, 0)$ を、まず Z 軸回りに θ [deg] 回転し、さらに X 軸回りに ψ [deg] 回転させて、 $X-Y$ 平面に投影すると考える。

この変換により、 X_1, Y_1 は、次のように投影される。

$$X_{p1} = (X_1 \cos \theta, X_1 \sin \theta \sin \psi) \quad (1)$$

$$Y_{p1} = (Y_1 \sin \theta, Y_1 \cos \theta \sin \psi)$$



x, y : 投影面上の座標
 X_p, Y_p, Z_p : 三次元空間上の座標軸を投影したもの
 O : 原点

図1 等角投影法による投影面

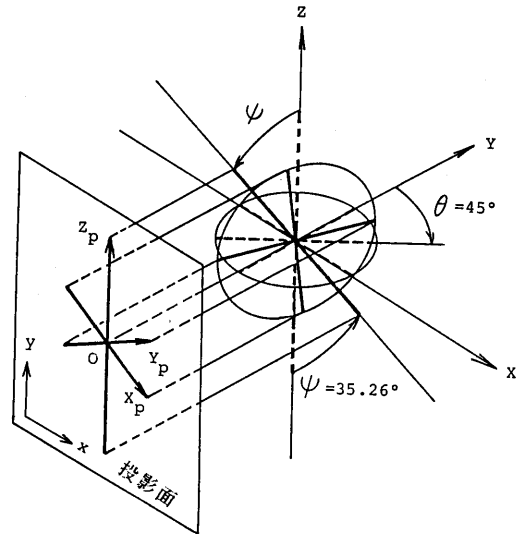


図2 等角投影法

投影面の x 軸に対して、投影後の座標 X_{p1} が $-30[\text{deg}]$ 、 Y_{p1} が $30[\text{deg}]$ の角をなすことから次の結果を得る。

$$\tan \theta = -1 \quad (2)$$

$$\theta = -45 \text{deg}$$

$$\sin \psi = \tan(30 \text{deg}) = 1/\sqrt{3} \quad (3)$$

$$\psi = 35.26 \text{deg}$$

すなわち、等角投影法は図2に示すように、三次元空間上の X, Y 軸を、 Z 軸回りに $-45[\text{deg}]$ 回転し、さらに Z 軸を投影面に対して、 $35.26[\text{deg}]$ 傾けて、投影したものといえることができる。

したがって、主軸方向の寸法は、以下のようになる。
 $\cos(35.26 \text{deg}) = 0.816 \quad (4)$

これにより、実寸に対しての縮尺率が 0.816 となり、主軸方向の寸法が等しい。このため、等測投影法ともよばれる。また、主軸方向では、寸法の測定が可能で

あり、正確に描くことができる。

本報告では、次の第3章で、文献3で紹介した頂点および直線の対話形式による入力方法について簡単に紹介し、第4章では第3章で示す外形形状をガイドのデータとして曲面を入力する方法について示す。

3. 等角投影法による立体外形形状の基本的な入力方法

3.1 3次元空間上の頂点の入力

等角投影法により立体形状を計算機に入力する基本として、まず、3次元空間上のある1点 $P(X, Y, Z)$ を、2次元平面であるタブレットから入力することを考える。なお、本節以降、3次元座標と2次元座標を区別するために、3次元空間上の座標を大文字で表し、2次元の投影面上の座標は小文字で表す。

等角投影法による投影面上のある1点を選ぶと、この点に対応する3次元空間上の座標は、図3における点 P_{7a}, P_{7b} のように、1点には定まらない。したがって、このように投影面から3次元データを入力する場合、座標値を決定するためには、まず3次元座標が既知である基準点を定め、これに対する投影面上での相対的な位置によって、座標値を求める。

すなわち、図4に示すように、投影面上のある点 $p_0(i_0, j_0)$ を基準点としたとき、次に指定できる点の方向を6方向に限定する。これにより、例えば図4の投影面において、次に指定する点の基準点からの方向を1で示す向きとするならば、対応する3次元空間上ではY座標値が増加する。

タブレット上での2次元座標の指定に必要な初期値は、3次元空間上の基準点の座標 (i_0, j_0) である。2次元平面上で、同じ点をオペレータがタブレットから2度指定したときには、この点をデータとし、その他の場合のうち、図4に示した方向の限定を満たしているときには、指定された点をパスとして、各方向に応じて、3次元空間上の座標値を変化させる。

3.2 稜線の基本入力

3.1で述べた、3次元空間上の頂点を入力するアルゴリズムを用いて、稜線の入力を行う。

データ入力の前段階として、補助線となるメッシュを描いておく。次に、図5(a)のようにCRT画面のどの位置を3次元空間上での原点とするかを指定して、主軸を描く。データ入力における初期状態は、原点 O を $(0, 0, 0)$ を基準点とする。例えば、原点を始点として、点 $P(5, 4, 3)$ を終点とする稜線を作成したいときは、図5の(b)(c)(d)のような手順でデータを入力する。すなわち、図5(a)において、投影面上の原点の位置を再

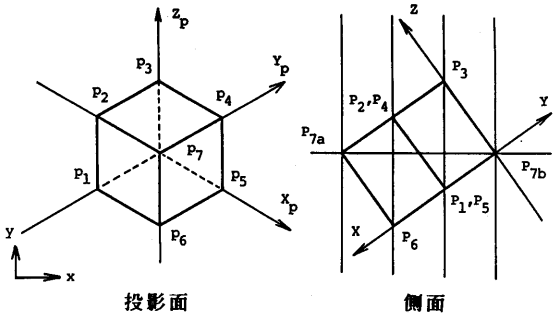


図3 投影面上の座標と3次元座標との対応

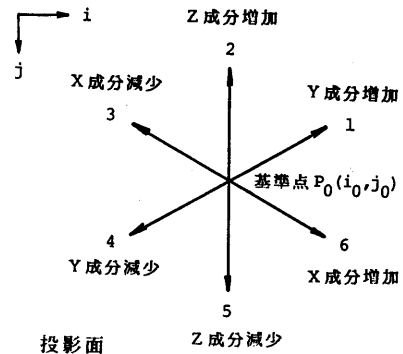
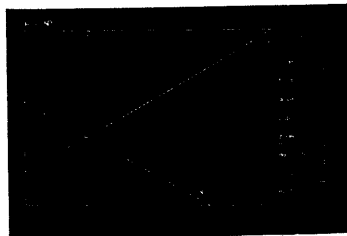


図4 投影面上での方向限定

び指定すると、稜線の始点座標として $(0, 0, 0)$ が登録され、投影面上のこの位置が基準点になる。次に、基準点から、図4に示した限定方向のうち2で示す方向に、補助メッシュの座標で5マス移動した位置を指定すると、図5(b)のように、3次元座標はX座標が5増加して、パス・データ $(5, 0, 0)$ が得られる。さらに、 $(5, 0, 0)$ を基準点として、図4の1で示す方向へメッシュの4マス分だけ移動した位置を指定すると、図5(c)のパス・データ $(5, 4, 0)$ が得られる。同様に、Z方向の座標を変化させて図5(d)のような $(5, 4, 3)$ のパス・データを得て、最後に投影面上のこの位置を再び指定すると、稜線の終点となる3次元座標 $(5, 4, 3)$ が得られる。

稜線データを入力する際に、最初の基準点として常に原点を用いると、稜線入力のために稜線の始点位置まで上記の手順で座標値を移動しなければならず、操作が冗長になる。これを避けるために、ある稜線を入力するとき、最初に指定した投影面上の位置 p が、既に入力されているある3次元データ P を投影した位置と一致する場合、 p および対応する3次元データ P を次の基準点とする。これによりワイヤフレーム・モデルが容易に入力されることになる。



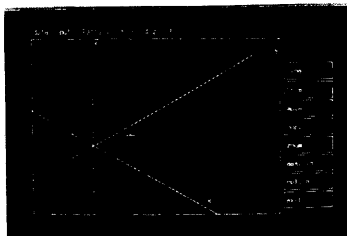
(a) 原点の指定



(b) X座標を変化させた状態



(c) Y座標を変化させた状態



(d) Z座標を変化させた状態

図5 稜線の入力手順

4. 等角投影法を用いた曲面の入力

4.1 曲面の表現

本章では、等角投影法を用いて曲面を入力する方法について述べる。

三次元空間上の曲面を計算機内部で表現することについては、多くの研究が行われている⁴⁾。曲面の表現は、計算機内部での曲面データの持ち方により、曲面を数式で表す方法と、多面体近似あるいは離散的な数値データとして表現する方法とに大別できる。

曲面を数式によって表現する方法は、曲面の表面を連続的に定義するもので、球面や二次曲面などの陰数表示によるものと、媒介変数で表現された複数の空

間曲線により曲面の一部を切り出し、その内部を曲線の混成 (blending) によって補間するものがある。後者において、切り出した曲面の一部はパッチ面 (patch) とよばれる⁵⁾。

製造工程に利用することを前提としたCADでは、模型を実測した点列データを数式化して、曲面形状を表現することが多い。たとえば、航空機製造におけるCADでは、機体の形状データを数式で定義したMDシステム (master Dimension System) とよばれるデータベースが作成され、これによる形状データが設計から製造まで一貫して利用される⁶⁾。

一方、多面体あるいは離散的なデータ列として曲面形状を表す方法は、コンピュータ・グラフィクスなどの、形状データに精度が要求されない分野で用いられている。この方法には地形図の表示に用いられる、 $Z_j = f(X_i, Y_j)$ の形をした二変数一価関数によるもの、物体の各断面ごとに二次元データを持つ形式のもの、空間曲線の補間で得られた曲面上の点を、離散的に持つものなどがある。

それぞれの特徴をみると、二変数一価関数によるものでは、変数の間隔が一定で、データの変更がしやすいが、データが急激に変化するところでは、データの間隔が大きくなってしまふ。さらに、一価関数で表せない場合は使えない。

断面図による方法は、実際に存在する物体の形状を計測する場合に有効である。しかし、滑らかな曲面を多面体近似で表現するには、多くの断面形状が必要になる。

空間曲線の補間による方法は、滑らかな曲面を多面体近似する場合に有効である。また、基準となる空間曲線とこれに対する断面形状を指定して、これらを補間して曲面形状を生成する手法は、人間にとって理解しやすいと考えられる。

本報告では、オペレータが考えた曲面形状を、表示された投影図を見ながらタブレットから直接入力するために、曲面の生成は指定データが少なくすむ空間曲線を補間する手法によって行う。タブレットからは、第3章で述べた方法を用いて基準となる曲線の制御点と、これに対する断面の制御点を入力する。入力された制御点を補間して直線近似された複数の空間曲線を生成し、さらに空間曲線を補間することにより、最終的な曲面の多面体近似データを得る。

4.2 空間曲線の生成

本節では、空間曲線を生成するための補間方法について述べる。本報告で用いた空間曲線は、指定された制御点列を補間することによって生成される。一般に、補間手法に対して必要とされる条件としては、補間によって生成される曲線が制御点列から容易に想像でき

ること、逆に滑らかな曲線を生成するとき、制御点の指定が簡便であることなどがあげられる。本報告では、これらの条件をふまえて、図6に示すような補間手法を用いた⁷⁾。すなわち、制御点列として P_0, P_1, P_2 を指定したとき、 P_0 と P_1 の中点を P_{01} 、 P_1 と P_2 の中点を P_{12} とする。このとき、 P_{01} において線分 P_0P_1 に接し、 P_{12} において線分 P_1P_2 に接するような楕円曲線を、補間された曲線とする。

ただし、制御点列が閉じておらず、 P_0 または P_2 が制御点列の端点である場合、補間曲線は中点で接する代わりに、 P_0 または P_2 それ自体を接点とする。また、制御点列が閉じているときは、補間曲線も閉曲線とする。この曲線をパラメータ t を用いて表示すると、以下のように表せる。

$$X(t) = (A \cdot t^2 + B \cdot t + C) / (1 + t^2) \quad \{0 \leq t \leq 1\} \quad (5)$$

ただし、

$$\begin{aligned} X(t) &= (X(t), Y(t), Z(t)) \\ A &= P_{01} + 2(P_{12} - P_1) \\ B &= 2(P_1 - P_{01}) \\ C &= P_{01} \end{aligned} \quad (6)$$

マトリクス形式で表現すると、式(5)は次のようになる。

$$X(t) = \begin{bmatrix} (1-t^2) & 2 \cdot t(1-t) & 2 \cdot t^2 \end{bmatrix} \begin{bmatrix} P_{01} \\ P_1 \\ P_{12} \end{bmatrix} \quad (7)$$

本報告では、補間した曲線を直線近似して点列データとして持つために、パラメータ t は離散的な値とするが、 t を等分割して点列を得ると、点列間距離が一定にならない。このため、 t は以下のように分割する。

$$t = \tan((\pi \cdot n) / (4N)) \quad (8)$$

ただし、

$$\begin{aligned} N &: \text{分割数} \\ n &= 1, 2, \dots, N \end{aligned}$$

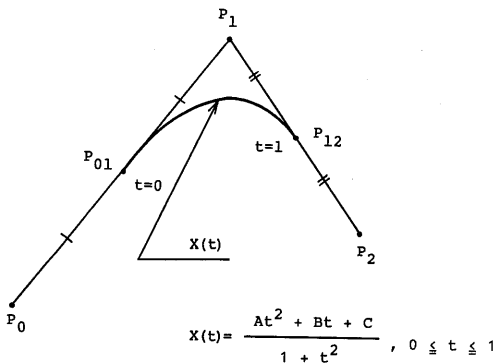


図6 曲線の補間

4.3 曲面の生成

本節では、曲面の境界を表す空間曲線を補間して曲面を生成する手法について述べる。

図7に示すように、4本の空間曲線が与えられたとき、これらの曲線で囲まれた部分の曲面を、曲線を混成して生成する。混成を行う関数は、混成関数(blending function)とよばれ、双 n 次式 $B(u, v)$ の形をもつものや、パラメータ u に対して独立な $B(v)$ の形をとるものがある。

双 n 次式によるものは、実測して得られたデータのよう、各パッチ面が小面積で、しかも各パッチ面での曲率変化が小さいような場合の補間に適すると考えられる⁸⁾。

一方、パラメータ u に対して独立な混成関数は、図7のうち、パラメータ u 方向の曲線 $G(u, 0), G(u, 1)$ を断面形状とし、 v 方向の曲線 $G(0, v), G(1, v)$ を断面形状の径路とみて、断面形状の形が優先するようにしたもので、任意の v における断面形状が容易に想像でき、設計者が意図する形状を入力するのに適していると考えられる⁹⁾。

本報告では、スケッチなどをもとに対話形式でデータを入力することを前提とするので、後者の方法を用いた。断面形状の空間曲線と基準径路の空間曲線が与えられたとき、これらを補間して曲面を生成する方法を以下に示す。

図8に示すように、三次元空間上で基準径路を生成するための2本の制御点列から、それぞれ対応する三点を取り出し補間した曲線を、パラメータ v によって、以下のように表す。

$$\left. \begin{aligned} BS_1(v) \\ BS_2(v) \end{aligned} \right\} \quad \{0 \leq v \leq 1\} \quad (9)$$

さらに、基準径路曲線の始点と終点の組 $BS_1(k), BS_2(k)$ ($k=0, 1$)に対応して指定された断面形状の制御点列のうち、ある三点を選び、これを補間した曲線をパラメータ u によって表し次のように表す。

$$\left. \begin{aligned} SEC_1(u) \\ SEC_2(u) \end{aligned} \right\} \quad \{0 \leq u \leq 1\} \quad (10)$$

このとき、補間された曲面は、 $S(u, v)$ ($0 \leq u \leq 1, 0 \leq v \leq 1$)として次のように表される。

$$\begin{aligned} S(u, v) &= BS_1(v) + T(v) \cdot R(v) \{ a \cdot (1-v) \cdot SEC_1(u) \\ &\quad + b \cdot v \cdot SEC_2(u) \} \end{aligned} \quad (11)$$

ただし、 $a = BS_2(0) - BS_1(0)$

$$b = BS_2(1) - BS_1(1) \quad (12)$$

$T(v)$ は次式に示すように、 $SEC_1(u)$ と $SEC_2(u)$ から定

まる v での断面形状の大きさを、基準径路により、規定する相似変換である。

$$T(v) = (BS_2(v) - BS_1(v)) / (a \cdot (1-v) + b \cdot v) \quad (13)$$

また、 $R(v)$ は、 a から b への回転をパラメータ v により表した 3×3 のマトリクスである。

すなわち、 a と b のなす角度を Θ としたとき、その間の角度 θ をパラメータ v で表す。

$$\theta(v) = \Theta \cdot v \quad (14)$$

a と b により作られる平面の単位法線ベクトルを $n(l, m, n)$ とすると、 $R(v)$ は、以下のように表される。

$$R(v) = \begin{bmatrix} \ell^2 + (1 - \ell^2)C & \ell \cdot m(1 - C) + n \cdot S & \ell \cdot n(1 - C) - m \cdot S \\ \ell \cdot m(1 - C) - n \cdot S & m^2 + (1 - m^2)C & m \cdot n(1 - C) + \ell \cdot S \\ \ell \cdot n(1 - C) + m \cdot S & m \cdot n(1 - C) - \ell \cdot S & n^2 + (1 - n^2)C \end{bmatrix} \quad (15)$$

$$\left. \begin{aligned} \text{ただし、} C &= \cos(\theta(v)) \\ S &= \sin(\theta(v)) \end{aligned} \right\} \quad (16)$$

プログラム上では、パラメータ u, v は、式(8)による離散的な数値をとり、式(11)を各断面形状の制御点列および基準径路の制御点列に対して実行する。

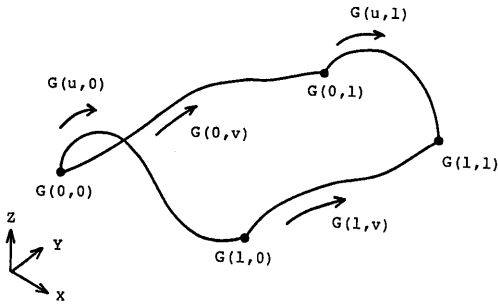


図7 空間曲線の補間による曲面生成

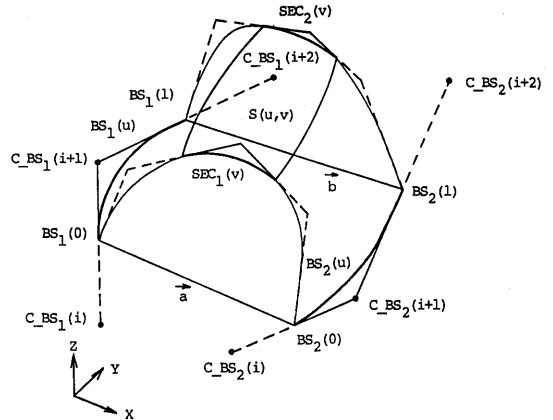


図8 断面形状と径路の補間による曲面生成

4.4.4 曲面入力アルゴリズム

図9に曲面入力アルゴリズムを示す。まず、基準径路となる曲線の制御点列をタブレットから入力する。図10(a)に基準径路の制御点列を入力して補間し、曲線を生成したときのCRT画面を示す。

次に、図10(b)のように、基準径路の各制御点ごとに断面形状の制御点列を入力して、断面形状の生成を行う。

図10(c)のように断面形状の入力が終了したならば、これらの曲線の直線近似点列から多面体近似された曲面が図10(d)のように生成できる。

以上、等角投影法を用いて、タブレットから制御点列を入力し、設計者が意図する曲面を生成する手法について述べた。

本手法により、自由曲面の入力が、直接画面で確認しながら行うことができた。しかし、曲面間の滑らかな接続については考慮していないので、複数の曲面を滑らかに接続しようとする場合は、新たに入力する制御点列が、接続される境界で連続となるようにする必要がある。

図11に、本手法により入力した曲面の例を示す。ただし、図11は曲面を多角形近似しワイヤフレームにより表現したものである。

《曲面の入力》

```

{
  [基準径路となる曲線の制御点列 C_BSk[i] (k=1,2) の
  入力。      制御点の数 : 2·M ]

  for(i=1 → M){
    [ C_BSk[i] を始点とする断面形状の制御点列
    C_SEC[i][j] の入力。      制御点の数 : N ]
  }

  for(i=1 → M-1){
    --- C_SEC[i] と C_SEC[i+1] の中間位置
    での断面形状の制御点列 M_SEC[i]
    の生成。

    if(i=1){
      [ M_SEC[i] ← C_SEC[1] ]
    }
    else if(i=M-1){
      [ M_SEC[i] ← C_SEC[M] ]
    }
    else{
      [ M_SEC[i] ← (C_SEC[i]+C_SEC[i+1])/2 ]
    }
  }

  for(i=1 → M-2){
    [C_BSk[i], C_BSk[i+1], C_BSk[i+2] を補間し
    た BSk[v] (k=1,2; 0 ≤ v ≤ 1) を生成する.]

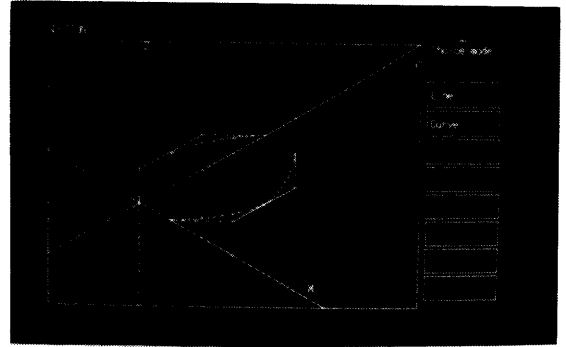
    for(j=1 → N-2){
      [ M_SEC[i+k-1][j], M_SEC[i+k-1][j+1],
      M_SEC[i+k-1][j+2] を補間した
      SECk[u] (k=1,2; 0 ≤ u ≤ 1) を生成する。

      S(u,v) の生成。 ]
    }
  }

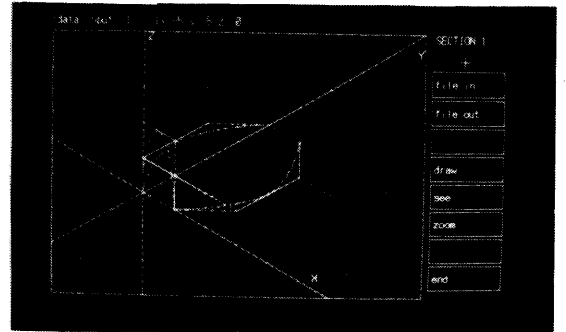
  [ S(u,v) から面ループを生成する。
  曲面データの生成。 ]
}

```

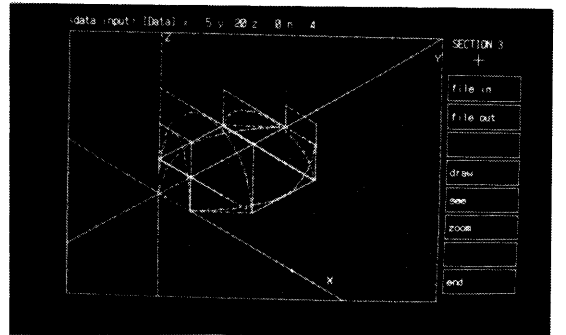
図9 曲面入力アルゴリズム



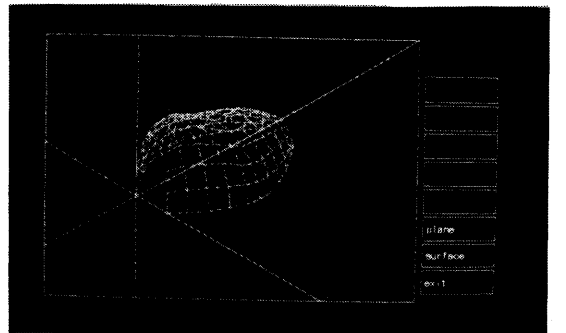
(a) 基準径路の指定



(b) 断面形状制御点の指定



(c) 断面形状の生成



(d) 曲面形状の生成

図10 曲面形状の入力

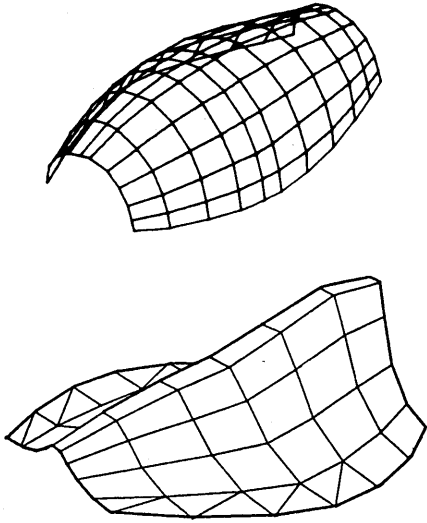


図 1.1 曲面の入力例

《文献》

- 1) 安居院, 中嶋: “画像工学の基礎”, pp.164-165, 昭晃堂
- 2) HIROSHI SAKURAI, David C.Gossard: “Solid Model Input through Orthographic Views”, ACM-SIGGRAPH, Vol.17, No3, pp.243-252(1983)
- 3) 中嶋, 安居院, 羽生田: “等角投影法を用いた簡易な立体形状入力システムの試作”, 情報処理学会「グラフィクスとCAD研究会昭和60年度」シンポジウム, 1985.
- 4) B.A Barsky: “A Description and Evaluation of Various 3-D Models”, IEEE CG&A, Jan. pp.38-52, 1984.
- 5) R.N.Goldman: “An Useful of Blending Functions”, IEEE CG&A, Oct. pp.49-54, 1983.
- 6) 佐田登志夫: “CAD/CAMガイドブック”, 工業調査会, 1983.
- 7) 芳田剛, 小野博宜, 田嶋太郎: “コンピュータ図学”, コロナ社, pp.81-111, 1976.
- 8) D.F.Rogers, J.A.Adams: “Mathematical Elements for Computer Graphics”, McGraw-Hill, 1976.
- 9) 津田順司, 上西博文, 菊池純男: “3次元図形処理システムMDM-Iの開発”, 情報処理学会グラフィクスとCAD研究会資料5-3, 1982.

5. まとめ

本報告は、計算機に対する立体形状データの入力を、より簡便に行うことを目的として、等角投影法を用いた立体形状データ入力システムの試作について述べた第2報である。

第1報では、主として直線から構成される面を入力する方法について示し、さらに本報の第2報では曲面の入力アルゴリズムについて述べ、等角投影法による曲面データの作成を対話的に行う技法について示した。

以上に述べた各手法により、等角投影法による立体形状データの入力が充分可能であることが確認できた。

今後に残された課題としては、タブレットとCRTディスプレイというデータ入出力装置の使用を前提とした場合、本データ入力システムを、より統合されたシステムとして見直し、マン・マシン・インタフェースの改善や、他の入力方式との併用などを考慮する必要があると考えられる。