〔特別講演〕

# コンピュータ・グラフィックスとマン・マシン・システム

東京大学宇宙航空研究所 助教授　大須賀　節　雄

## 1　はじめに

コンピュータ・グラフィックス技術が華々しく登場したのはすでに十数年も前のことである。したがってもはや決して新技術ではない。また形式的な分類をするなら，コンピュータ・グラフィックスあるいはそれを具現化しているディスプレイ装置は数ある入出力装置の一形式に過ぎないとも言える。しかるに今なお多くの研究者がこの分野の研究に携さわり，しかも新鮮な魅力を持ちつづけている。この原泉は何だろうか。

魅力の原因を探るにはその特徴を知るのが近道である。コンピュータ・グラフィックスについて言えば，それがコンピュータ端末として二次元情報の表示が行なえること，単位時間あたりの情報転送量が大であること，表示のみでなく消去が可能なこと，カラーや濃淡など表示の質が多様で制御可能なこと，などが挙げられる。

われわれは三次元の世界に住んでいるが，すべての物体は網膜上に投影された二次元の像として感知される。また人間は情報の表現および記憶に紙という二次元の媒体を用いてきた。そのために三次元の情報を二次元の面上に表現する技術や，言語という一次元の表現形式で表わしにくい情報を二次元面上で表現する技術を育ててきた。これは特に科学技術の分野で顕著で，建造物・船・自動車・飛航機などの形状設計や機械設計，工場のレイアウトやプラントの配管，電気回路設計やＩＣマスク・パターン設計，各種関数や相関関係などのグラフ表示，雑誌・新聞などの編集など多様な二次元表示が用いられてきている。

コンピュータの世界は一次元であり，元来は入出力も処理も，一次元表現である言語で記述されるもののみに対象が限定されていた。したがって上述のような諸問題は極く一部を除き，この対象からはずされることになり，この意味で従来のコンピュータは人間の社会に発生する情報の一部のみしか扱かうことができなかった。もし二次元の情報を十分早く表示できるなら，すなわち一画面の表示を人間の思考の流れを著るしく停滞させない程度の速度で行なえるなら，そのようなシステムは現実の世界に発生する情報を非常に広い範囲にわたって処理の対象とすることができるであろう。コンピュータ・グラフィックスにはまさにこれを実現する事が期待されているのであり，その最大の魅力がここにあるといえるだろう。

コンピュータ・グラフィックスのもう一つの魅力は表示の実体感や美的要求を満足してくれる点にある。次節以下で述べるように，コンピュータ・グラフィックスの基礎技術が確立し，曲面の表示や陰影技法が開発され，カラー・グラフィックスが開発されるにつれ，より実際の像に近い表示を行おうとする努力がなされた。このことは，従来のコンピュータでは常に抽象化された情報のみが扱われ，表示自体は無味乾燥であったのに比べると雲泥の差であり，コンピュータ・グラフィックス分野に技術のみでない芸術的要素を付け加えることになった。

以下ではマン・マシン・システムという立場から，コンピュータ・グラフィックス技術を眺めてみることにする。なおコンピュータ・グラフィックスをハード面で実現するディスプレイ装置は当初は，リフレッシュ型ランダム・ポジショニングＣＲＴが主流であったが，近年カラー・ＣＲＴ，高分解能ＣＲＴ，カラー・ビデオ・プロジェクション・システムなどが開発され，さらに液晶，ＬＥＤ，ＥＬ，プラズマ・パネルなど固体表示素子が開発された，技術的に実用化の域に達しているものもある。これら各種ディプレイ素子あるいは装置はそれぞれ固有の特徴を有しているが，システムにおけるディスプレイ端末という立場からみると，具備すべき基本機能は同様である。すなわちディスプレイ端末として使用されるために最小限必要な条件は，ディスプレイ面上に二次元座標系を定義することができ，一組の座標値が外部から与えられた時に，その位

置に点を表示し得ることである。すべての図形は一定の規則のもとで点の集まりにより表わされるから，この条件を満たすすべての装置は図形端末となり得る。現実の端末装置には更に高度の機能を備えたものが多いが，本稿では上述の条件を満たすものとして代表的なＣＲＴディスプレイを想定する。

## 2　基礎技術

ディスプレイ装置は二次元の図形を表示するがコンピュータ自体は一次元的な処理を行う。したがって一次元情報と二次元情報の間の変換が必要であり，この変換アルゴリズムがコンピュータ・グラフィックスの基礎技術を構成する。以下これについて簡単に述べる。

### 2.1　ランダム・ポジショニングとラスタ・スキャン

表示装置としてのＣＲＴの基本は表示面内で互に直交する２方向（これをＸ軸，Ｙ軸とする）に，相互に独立に外部信号によって表示点を移動し得ることである。ＣＲＴディスプレイの場合，表示点はＣＲＴ内部で発生され，ビーム収束機構によって蛍光膜を塗布されたスクリーン上に焦点を結ぶ電子ビームによるスポットである。電子ビームは外部から加えられる電場や磁場の中を通るとき，場の強さに応じて偏向する。この特性を利用して外部に偏向用の電極あるいはコイルを用い，ここに加える電圧もしくは電流を制御することにより，表示面上の点の位置を制御することができる。

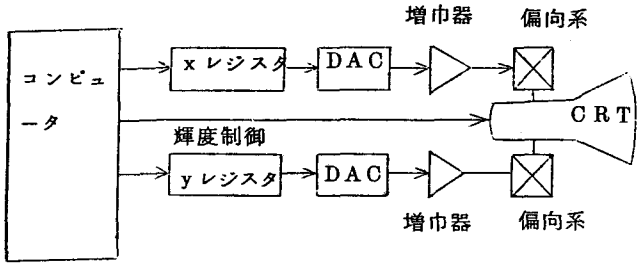
このビームの制御方式は目的により異なるが，コンピュータ・グラフィックスでは２通りの方式が用いられる。第一はＸ，Ｙ軸にそれぞれ同じ方式の一組の偏向機構を用い，各偏向機構は外部からのＸ方向またはＹ方向の制御信号に比例して輝点が移動する。Ｘ方向，Ｙ方向の任意の位置に輝点を移すことができるということから，これをランダム・ポジショニング方式と呼ぶことにする。第二は家庭用テレビジョンと同じ方式で，電子ビームをＸ軸方向に左端か右端まで，Ｙ方向位置を少しづつずらしながら走査する。これをラスタ・スキャン方式と呼ぶ。この場合，ビームの走査と同時にビームの輝度を制御し，ある座標位置を表示する時はビームがこの位置にきた時，輝度を増す。

第一の方式はコンピュータで図形を構成する時，特に線図形の場合に適している。図形が点のつながりによって表現されるからである。第二の方式は面を表示するのに適している。走査するということは画面全面を塗りつぶすことに相当するからである。反面，コンピュータで図形を定義しながら表示するのには適しない。したがって，まず，コンピュータ内メモリに図形・画像情報を作りあげ，記憶する。メモリの各要素は表示面の各座標点に対応し，図形・画像情報は各点の輝度情報を表わす。カラーの場合は赤，緑，青の三色の輝度情報が必要である。ディスプレイ装置はこのメモリから一定の周期で情報を読み出し，これと同じ速度でＣＲＴの電子ビームを振って表示面上を走査しながら，読み出した情報で輝度を制御する。表示方式によりディスプレイ装置のハードウエアおよびそれを駆動するための基本プログラムは異なるが，コンピュータ・グラフィックスの基本技術である。コンピュータによる図形生生技術は共通であり，これがコンピュータ・グラフィックスのソフト面での基本技術を構成する。したがって以下ではランダム・ポジショニングの場合を基本として述べる。

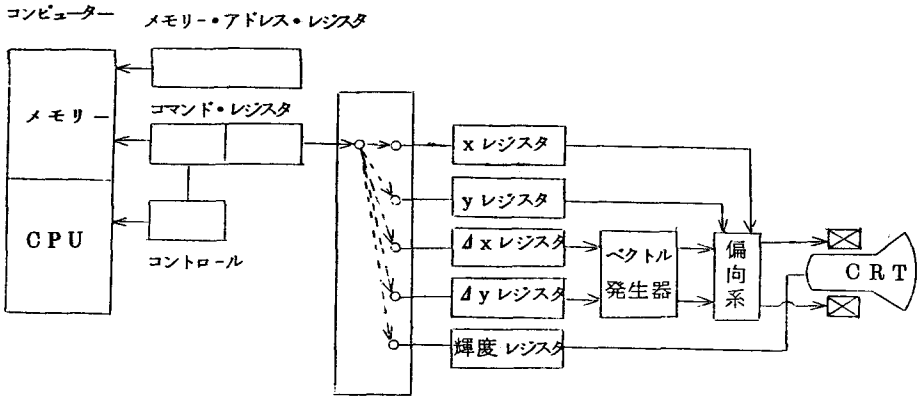
### 2.2　ディスプレイ装置の制御方式

最も単純なＣＲＴディスプレイ装置の構成は**図1**(a)のようになっている。この装置にディジタル入力が与えられると，Ｄ－Ａ変換器を通してアナログ信号に変えられ，それに比例した位置に輝点を発する。

このような装置では所要の形状，例えば直線を表示するには，プログラムにより直線を表わす点列を生成せねばならない。すなわち，各点の座標位置を計算し，それを順次装置に送ることになる。直線を点列に変




(a) 点表示型




(b) ベクトル表示型

図1 ディスプレイ装置ブロック図

換するアルゴリズムは比較的単純であり，表示速度をあげるためにこれをハードウエア化して制御部に組込んだ図1(b)のような装置も多い。更に直線のみでなく，円すい曲線，文字パターンなどを備えたものもある。計算機のプログラムにはこれらの図形要素を用いて複雑な形状を表示することが要求されるが，このようなプログラムは問題解決のためのプログラムそのものではなく，この補助として働く部分であり，ディスプレイ装置自体は人間と機械とのインタフェースとすると，このような基本の図形生成プログラム群はディスプレイ装置と，問題解決プログラム間の，システム内部のインタフェースと言ってもよいだろう。

### 2.3 線分の生成

線分の生成方法には基本的に2種の方法がある。1つは線分を点列で表わし，各点はディジタル量として計算するもの，もう1つはアナログ回路を用い，線分の両端点座標が与えられると，その間を等比率で変化するアナログ信号を生成し，それによって点の位置を制御するものである。後者は線分の発生回路を装置内に持つ場合のみ用いられるが，前者はそれ以外に計算機内のプログラムによる点列発生アルゴリズムとしても用いられる。装置に内蔵された線分の発生回路をベクトル発生器と呼ぶ。ベクトル発生器は初期のころはディジタル方式が用いられたが，その後発生速度の速いアナログ方式が増えてきた。いずれの方式も直線の方程式を解くという点では変りがないがディジタル方式はこれを離散的な点列すなわち二次元面内の各格子点のうち，与えられた2点を結ぶ直線に最も近いものを選ぶのに対し，アナログ方式は連続的なアナログ量で直線上を忠実になぞる。従って，ディジタル方式ではDDA方式，対称形DDA方式，レート掛算器方式などがあるが，いずれも点列が階段状になることは避け難く，表示が遅い反面，任意の点にたいし制御が可

能である。これに対し，アナログ方式は連続的な直線で得られるが，途中の制御は困難である。実際の方式に対する詳細は省くが，以後，ハードウエアによるにしろソフトウエアによるものにしろ，ディスプレイ装置は線分の発生回路を有しているものとする。

### 2.4 ディスプレイファイルとグラフィック・データベース

ディスプレイ装置に直線の表示機能があれば，プログラムにより線分データを連続的に発生し，送出することにより，任意図形を表示することができる。画面そのものが情報を記憶する蓄積形のディスプレイならこれでも良いが，リフレッシュ形ディスプレイでは静止した図形を表示するためには毎秒30回程度，同一図形を繰返し表示を行うことにより，画面のちらつきを避ける方式となっており，プログラムが計算しながら表示するのでは複雑な図形が描けなくなる。そこで図形を表わす直線列を前もって一定の順序に並べて記憶装置に入れておき，ディスプレイ装置の制御部が順次これにアクセスし，読取りながら表示する形式がとられる。このように図形表示のためのデータの集まりをディスプレイファイルという。

ディスプレイファイル内の各データの形式は前もって定められている。これには点や線を表わすデータのほか上記のリフレッシュのためにディスプレイファイナルの最後に達したら再び先頭にもどらねばならず，このために，あるいはこれ以外に図形表示の際の柔軟性を増すためにも，普通の計算機の命令語におけるジャンプ命令と同様の機能をもつものがあり，更に同じ図形パターンが幾つも同一図形の中に現われるときには，そのパターン用のデータ列を１つだけ作っておき，必要なときにそこに跳び，終れば元の列にもどるという，サブルーチンリングと同じ働きのものも準備されている。このようにディスプレイファイルにおける各図形データは計算機における命令と同様であり，ディスプレイファイル自身はプログラムと類似している。この意味で図形表示のための情報をディスプレイコマンドと呼ぶ。

ディスプレイファイルは通常はシステム内で扱われている全図形情報の一部である。グラフィックシステムを用いて問題解決を図る時には，モデルはグラフィックデータベースの形で存在する。これは形式も応用プログラムに適したものであり，ディスプレイファイルとは必ずしも同形ではない。ディスプレイファイルはグラフィックデータベースの中で表示部分のみが取り出され，一定の処理を受けて生成されるものである。例えばグラフィックデータベースは自動車の設計情報のとき，ディスプレイファイルにはそのフロントグリル部分が表示データとして作成されているという具合である。

データベースの中で表示される部分を窓と呼ぶ。窓に入る部分を取り出す操作，すなわち，どの図形要素を指定された窓用のディスプレイファイルに入れるかを決定することをウインドウイングと呼ぶ。この操作のもとで，ユーザーは任意の窓を指示することができる。

データベースは設計の進行に伴って更新はされるが，これ自体は静的なものである。これに対し，ディスプレイファイルは最終的な表示図形を表わすから，同じモデルの部分でも表現が異なるときには作り変えられる。例えば，三次元形状の場合，視点が変れば表示図形は変化する。ディスプレイファイルの編成には時間がかかるので，動的な変化を許すためにはこの変換を早く行わねばならない。このために，拡大・縮少とか座標変換のように，一定の規則の下で行われる変換をハードウエア化したものもある。

### 2.5 点および線の表現と変換

グラフィックシステムにおいてはプログラムにより表示図形を自由に操作できねばならない。図形を操作することは，基本的な図形要素である点または線分を操作することであり，更に，それは点の座標値に数学的な変換を施すことによりなされる。すなわち，データベース内の図形情報はウインドウイングにより表示される部分が取り出され，それに各種の変換が行われた後，ディスプレイファイルが生成される。変換とし

ては移動，拡大・縮少，剪断変形，回転，鏡像などがあり，更に，三次元形状では各種投影や透視変換などがある。

これらの個々の変換手順を求めることは困難ではない。しかし，個々の変換ごとに異った手順を用いることは効率的ではない。もしこれらを同一の方式で処理できれば体系としてすっきりするばかりでなく，変換手順をハードウエア化により高速化することも可能である。このようにして開発されたのが同次座標系と変換マトリックスを用いる方式である。同次座標系について述べる前に，変換とマトリックス演算の関係を見ておこう。これにはまず点の表現法を定めておく必要がある。以下ではまず二次元空間内での変換を考え，点の座標をベクトル形式で〔$x$ ・ $y$〕と表わす。

これに対し，2×2の変換マトリックスを用いてこのベクトルに次のような線彩変換を施す。

$$〔x \cdot y〕\begin{bmatrix} a & b \\ c & d \end{bmatrix} = 〔(ax+cy)\ (bx+dy)〕 = 〔x^* \ y^*〕$$

これにより，元の点〔$x$, $y$〕が〔$x^*$, $y^*$〕に変換される。このとき，変換マトリックス内の各要素の値に，以下の各種の変換が対応する。

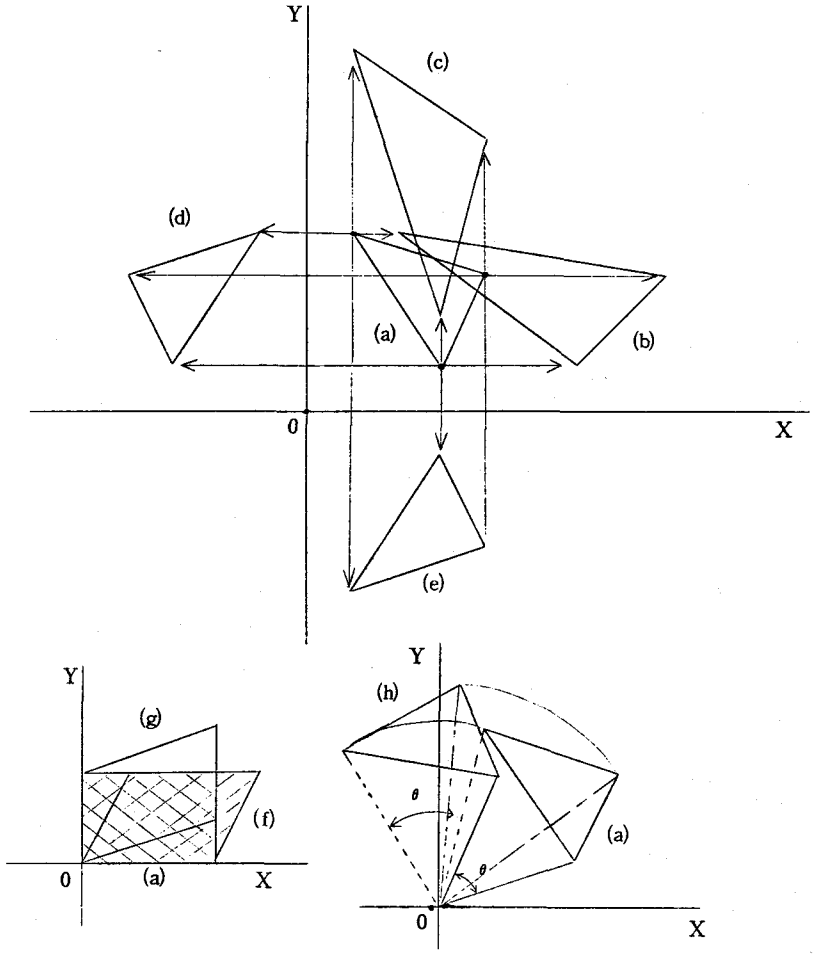



図2　各種変換

| $a$ | $b$ | $c$ | $d$ | 変換の内容 | |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 不 変 | (a) |
| $\gtrless 0$ | 0 | 0 | 1 | X軸方向に$a$倍の拡大（縮少） | (b) |
| 1 | 0 | 0 | $\gtrless 0$ | Y軸方向に$d$倍の拡大（縮少） | (c) |
| $-1$ | 0 | 0 | 1 | Y軸に関する鏡像 | (d) |
| 1 | 0 | 0 | $-1$ | X軸に関する鏡像 | (e) |
| 1 | $\neq 0$ | 0 | 1 | X軸方向への剪断変形 | (f) |
| 1 | 0 | $\neq 0$ | 1 | Y軸方向への剪断変形 | (g) |
| $\cos\theta$ | $\sin\theta$ | $-\sin\theta$ | $\cos\theta$ | 角度$\theta$の回転 | (h) |

この変換では原点〔0　0〕は変換により不変である。

一方，直線は両端点座標を与えれば定まるから，これを端点座標の組で表す。例えば〔$x_1$　$y_1$〕と〔$x_2$　$y_2$〕を結ぶ線分を $\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}$ で表わす。するとこの端点に上記の変換を施して得られる座標値の組を端点とする直線が次のように得られる。

$$\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix}=\begin{bmatrix} ax_1+cy_1, & bx_1+dy_1 \\ ax_2+cy_2, & bx_2+dy_2 \end{bmatrix}=\begin{bmatrix} x_1^* & y_1^* \\ x_2^* & y_2^* \end{bmatrix}$$

これが線分の変換となっていることは，元の線分で両端点間を$\lambda:1-\lambda$で分配する点〔$x$　$y$〕の変換点〔$x^*$　$y^*$〕が変換後の直線を同じ割合で配分することにより示される。〔$x_1$　$y_1$〕$=\boldsymbol{x}_1$，〔$x_2$　$y_2$〕$=\boldsymbol{x}_2$，〔$x$　$y$〕$=\boldsymbol{x}$とすると，$\boldsymbol{x}=\lambda\boldsymbol{x}_1+(1-\lambda)\boldsymbol{x}_2$であるとする。

$$〔\lambda x_1+(1-\lambda)x_2,\ \lambda y_1+(1-\lambda)y_2〕\begin{bmatrix} a & b \\ c & d \end{bmatrix}=〔x^*\ y^*〕$$

$$=〔a\{\lambda x_1+(1-\lambda)x_2\}+c\{\lambda y_1+(1-\lambda)y_2\},\ b\{\lambda x_1+(1-\lambda)x_2\}+d\{\lambda y_1+(1-\lambda)y_2\}〕$$

一方，〔$x_1^*$　$y_1^*$〕，〔$x_2^*$　$y_2^*$〕を結んだ点を$\lambda:1-\lambda$で配分する点の座標を〔$\tilde{x}$　$\tilde{y}$〕とすると，

$$\tilde{x}=\lambda x_1^*+(1-\lambda)x_2^*$$
$$=\lambda(ax_1+cy_1)+(1-\lambda)(ax_2+cy_2)=x^*$$
$$\tilde{y}=\lambda y_1^*+(1-\lambda)y_2^*$$
$$=\lambda(bx_1+dy_1)+(1-\lambda)(bx_2+dy_2)=y^*$$

となり，両方が一致する。

更に，2本の線間の関係については，2直線の交点が正しく変換後の直線間の交点に変換されることは上述のことから明らかであり，更に，平行な2直線は変換後も平行であることは，元の直線のこう配を$m_1$とすると

$$m_1=(y_2-y_1)/(x_2-x_1)$$

であるが，変換後の直線のこう配は

$$m_2=\frac{\{(bx_2+dy_2)-(bx_1+dy_1)\}}{\{(ax_2+cy_2)-(ax_1+cx_1)\}}=f(m_1)$$

のようになり，こう配は変化するが，元のこう配$m_1$にのみ依存して定まり，座標値によらないことから示される。

このように変換マトリックス法は具合の良い性質を有しているが，移動を扱えないという欠点がある。

この問題は点の位置を表すのに〔$x$　$y$　1〕のように3番目の座標を導入した表現法を用い，変換マトリ

ックスも3×3マトリックスに拡張するという方法で解決された[(2), (4), (5)]。これによると変換は

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a & b & p \\ c & d & g \\ e & f & r \end{bmatrix} = \begin{bmatrix} ax+cy+e, & bx+dy+f, & px+qy+r \end{bmatrix} = \begin{bmatrix} \overline{x} & \overline{y} & h \end{bmatrix}$$

のように，変換後に3番目の座標に$h$なる値が生ずるが，これから

$$\begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} \overline{x}/h & \overline{y}/h & 1 \end{bmatrix}$$

を作ることにより，変換前後が同形となる。

このように$n$次元空間の座標を$n+1$成分ベクトルで表すことを同次座標表現と呼ぶ。この表現のもとで，変換マトリックスの9個の成分のうち，$a$，$b$，$c$，$d$は2×2マトリックスと同様に拡大・縮小，剪断変形，回転，鏡像などの変換を表し，$e$，$f$は平行移動を，$p$，$q$は投影を，$r$は全体的な（座標軸によらない）拡大・縮小率を表す。

同次座標系により変換がこのように統一的に扱われるが，更に，変換マトリックスを用いる方法の利点は一連の変換が引き続いて行われる場合は，個々の点に個々の変換を順次施す替りに，複合変換形をマトリックス積として先に求めた後，これを点座標に施すことにより，多数の点の変換を高速化できるという点にある。すなわち，変換$T_1$，$T_2$，$\cdots T_n$，が点$x$に施される場合

$$x \cdot T_1 \cdot T_2 \cdots\cdots \cdot T_n = x \cdot T^*, \quad T^* = T_1 \cdot T_2 \cdots\cdots T_n$$

とすることができる。例えば，任意軸回りの回転を行う場合，$T_1$：回転位置への平行移動，$T_2$：原点回りの回転，$T_3$：元の座標への平行移動とし，

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -x_0 & -y_0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_0 & y_0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ -x_0(\cos\theta - 1) + y_0 \sin\theta & -x_0 \sin\theta - y_0(\cos\theta 1 - 1) & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \overline{x} & \overline{y} & h \end{bmatrix}$$

$$= \begin{bmatrix} x^* & y^* & 1 \end{bmatrix} = \begin{bmatrix} \overline{x}/h & \overline{y}/h & 1 \end{bmatrix}$$

となる。

この方法はそのまま三次元空間内の変換に拡張される。この場合，点の座標は$\begin{bmatrix} x & y & z & 1 \end{bmatrix}$で表され，変換マトリックスは4×4のマトリックスである。変換により$\begin{bmatrix} \overline{x} & \overline{y} & \overline{z} & h \end{bmatrix}$なるベクトルが求まるが，これから$\begin{bmatrix} x^* & y^* & z^* & h \end{bmatrix}\begin{bmatrix} \overline{x}/h & \overline{y}/h & \overline{z}/h & 1 \end{bmatrix}$とする。

変換マトリックスを**図3**のように表すと，$a \sim i$は拡大・縮小，剪断変形，回転，鏡像などを表す。これは二次元の場合と同様であるが，それぞれの変換の定義は次元が増したことにより複雑である。$j \sim \ell$は平行移動を表し，$p$，$q$，$r$は透視変換を，$s$は全体的な拡大率を表す。

$$\left[\begin{array}{ccc:c} a & b & c & p \\ d & e & f & q \\ g & h & i & r \\ \hdashline j & k & l & s \end{array}\right]$$

**図3　4×4変換マトリックス**

三次元変換の固有の問題として重要なのは投影である。一般に投影は$n$次元空間内の点をそれより低い次元の面上への変換として表される。この変換の方法により様々な投影図が可能である。図形処理では三次元の形状を扱う場合，表示は二次元のディスプレイ上になされるから，投影は不可欠である。

座標系の各平面に直角に投影を行う正投影や，ある角度をつけて立体感を出す斜投影はいずれも無限遠の光源からの投影に対応するもので比較的単純であるが，透視変換は有限距離の視点からの視線による変換である[(3)~(5)]，すなわち，三次元空間内に視点$E$と，$E$を含まない平面$S$を定め，物体上の各点から$E$に集まる直線を$S$に投影した図形として三次元の形状を表現する（**図4**）。

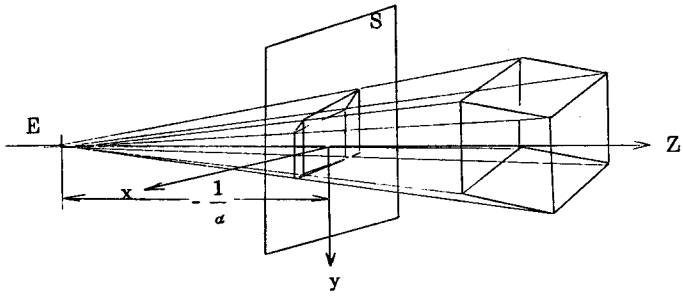



**図4　透視変換**

視線方向が$Z$軸と一致している場合，この変換は下記のような変換マトリックスを通して行われる。

$$\begin{bmatrix} x & y & Z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0^* & \alpha \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & 0 & (\alpha z+1) \end{bmatrix}$$

$$\begin{bmatrix} x^* & y^* & z^* & 1 \end{bmatrix} = \begin{bmatrix} x/(\alpha z+1), & y/(\alpha z+1), & 0, & 1 \end{bmatrix}$$

なお，変換した結果をディプレイ装置で表示する場合，$z$方向座標によって輝度を制御することにより現実感を出すという方法が用いられることがある。このためには変換後も$z$座標を残す必要があるが，これにはマトリックス内の$^*$の付いた要素を1とすれば良い。

この透視変換により原点の位置は不変である。従って，任意の点および方向に視点および視線を定めるときには，移動や回転との組合せにより複合変換を行う。

この方法はすでに述べたものと同じである。

### 2.6　曲線および曲面の表示

図形処理技術の中で曲線や曲面の扱いは年々増しており重要な技術となっている。一般に曲線を表示するときは点または直線の列で行うにしても，システム内の表現はこれでは不便で，正確さ，性質のは握しやすさ，必要記憶容量の少ないことなどの点で数学的な表現が望ましい。しかし，実際の問題で生ずる一般曲線は数式で表されない複雑なものが多い。そこで全曲線を幾つかの部分に分け，各部分を数学的な曲線で表すという方法がとられる。

数学的曲線の表現には非パラメータ表現とパラメータ表現があり，更に，前者は $y=f(x)$ の形の陽関数表現と $f(x\ 0)=0$ の形の陰関表現がある。これらは共に座標系に依存した表現である。

これに対しパラメータ表現は$t$をパラメータ，$P(t)$を座標位置とすると $P(t)=[f(t), g(t)]$の形で曲線を表す。この形式は閉曲線や多値関数などを表すにも具合が良いし，座標系に依存しない。パラメータの範囲は $t_1 \leq t \leq t_2$ のように有限であり，多くの場合，$0 \leq t \leq 1$ のように正規化されている。

上記の表現は二次元平面曲線の場合で示したが，表現形式は空間曲線の場合も全く同様である。このうち，

一般の曲線表示で重要なものにスプライン曲線と呼ばれるものがある。これは離散的な点列を与えてこれから全体の曲線を定義する。一般に数学的スプラインは各曲線部分を$m$次の多項式で近似し，隣接部分間の接続点では位置および $k-1$ 次の微分値の連続性を条件とする。一般に多項式が高次になると数値的に不安定となる。一方，低次のものでは多数の点にわたって曲線を表現することができないため，連続的に多数の多項式を接続する必要がある。3次曲線は変曲点とねじれとを表すことのできる最低次多項式として最も多く用いられている[5]。

一般式でこれを

$$P(t)=\sum_{i=1}^{4} B_i t^{i-1}, \qquad t_1 \leq t \leq t_2$$

$$P(t)=\{x(t),\ y(t),\ z(t)\}$$

と表す。この式の係数 $B_i$（$i=1\sim4$）は境界条件から定まり，両端点では位置の条件とこう配で，曲線の接続点では両方の曲線の2次の微係数の一致条件から求める。

この方法は与えられたすべての点を通る曲線を与えるが，設計などには必ずしも適さない。これは曲線を与えるために1次，2次の微係数までを指定することは直感的に行いにくいからである。これに代るものとして，幾つかの手法が開発された。

Bezier[7] の方法は多角形の頂点が与えられると，それによって一意に曲線が定まる形式のもので，始点と終点では曲線は与えられた頂点と一致するが，中間の点は必ずしも通過しない。Bezier 曲線は Bernstein 多項式を用いて

$$P(t)=\sum_{i=0}^{n} P_i J_{n,i(t)}, \qquad 0 \leq t \leq 1$$

$$J_{n,i(t)}=\binom{n}{i} t^i (1-t)^{n-i}$$

により曲線を定義し，$P_i$ が各頂点のベクトル成分を含むように決定される（**図5**）。

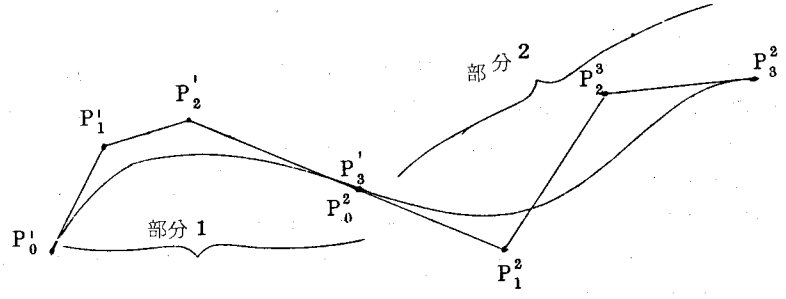



**図5　Bezier 曲線とその一次接続**

この方法は有用な性質を有しているが，曲線の定義において柔軟性が不十分であり，又，一部の変更の影響が全曲線に及ぶため設計には使い難い点がある。これを改善したものにB－スプライン法[8]および穂坂[10,11]の方法がある。実用的にはこれらは非常に有用な方法であるが，限られた紙面で要点すら述べるのが困難である。興味のある方は是非末尾文献を参照していただきたい。

曲線表現の方法を拡張し，$u$，$w$，の二つのパラメータを用い，$P(u,w)$ で $x$ 方向に $u$，$y$ 方向に $w$ のときの $z$ の値を与える方法で曲面を表すことが可能である。最初は Coons[9] により，各格子点の座標と微分値が与えられたとき $x$，$y$ 両方向の小格子部分を，まず四辺の曲線を与え，次いでその中を補間するという方法で曲面を求め，これを接続してゆくという方法が用いられた。各格子内の曲面をパッチという。補間には両方向の3次式が初期に用いられたが，曲線の場合と同様 Bezier 曲面[7]，Bスプライン[10]曲面，穂坂[11]

の曲面などがより優れた方法として開発されている。

## 3　図形入力

図形情報を扱かうには単に表示のみでなく，図形入力が必要である。コンピュータ・グラフィックスの分野でも，頭初から図形入力装置として様々なものが利用されてきた。基本的には図形入力は出力の逆に2次元上の任意の座標点をコンピュータに指示し得ることであるが，この他に，コンピュータ・グラフィックスにおいては，表示された多くの図形要素のうちの一つを指すことによって，それを他から識別する機能が要求された。前者をポジショニング，後者をポインティングと呼ぶ。現在用いられている入力装置には，CRTが図形を電子ビームの移動によって描くこと，したがって画面上では瞬間的には常に一点のみが光っていることを利用し，感光機能を持つ装置によってこの光を捕え，信号をコンピュータに送り，コンピュータ側で，表示中の図形が指示されたことを知るというポインティング機能に基づくライトペン，二次元面内の指示した点の位置を検知し，コード化して送るタブレット，操作部は極く単純で$X$，$Y$いずれの方向にも人間の操作により信号を出すが，操作量と信号との関係は厳密なものではなく，人間が表示面を見ながら所要の位置に点が達した時停止させることにより，結果的に所要の精度で点の位置を制御し，移動量がコンピュータ内で取り出せるようにしたマウスとかジンバルなどが用いられている。なおタブレットにも各種の方式があり，電位差，磁場，超音波など用いる媒体により方式も異なる。

一方，表示の場合と同様に，これら任意の位置の指定とは別に画像等の入力ではこれを走査して光の強弱を電気信号に変え，コンピュータがこれを読み込むスキャン方式の入力がある。

いずれの場合にもコンピュータが受け取るのは単なる数値の組である。図形処理の場合，どのように入力情報に意味づけするか，画像情報の場合，そのデータからどのようにして，また何に基づいて対象を認識するかは全く別の問題である。

## 4　マン・マシン・システム

コンピュータ・グラフィックスとして世に出た最初のシステムはSKETCHPADと呼ばれるものである。このシステムでは，CRT画面に橋梁のモデルを描きながら，その強度計算を行い，結果を表示するような例によって，グラフィック・システムの威力を示した。それを追いかけるように，当時コンピュータ・グラフィックスの先駆者といわれる人達が次々に将来への明るいビジョンを述べた。それは多くの人に，“図形によって問題を入力することによりコンピュータからの解析結果がディスプレイ上に表示される”と主張しているように受け取られた。しかし現実はその通りにはゆかなかった。図形を入力し，表示することと，問題を解くこととは全く別の問題だからである。コンピュータ・グラフィックス技術は二次元情報によって人と情報交換する可能性を示したのであって，コンピュータの問題解決能力を増したのではない。

では先駆者たちの主張は誤まっていたのだろうか。そう早急に結論するのも危険である。もしコンピュータ・グラフィックスがなければどうしてもコンピュータの処理範囲に入りにくい種類の問題が多数あることはすでに述べた通りであり，また，もしコンピュータによる問題解決能力が（必ずしもコンピュータ・グラフィックス技術との関わりなしに）向上したなら，これと組み合わせることにより，確かに，かつて述べられたことが実現するからである。

単に数学的に表現された問題を解くばかりでなく，与えられた仕様を満たす製品を設計するといった一般的な意味で問題を解決することは，コンピュータになし得るものではない。このような環境において，現在最も有力な方法は人間とコンピュータがチームを作り互いにその長所（人間における創造力，総合評価能力，認識能力など，コンピュータにおける高速度情報記憶能力，大量記憶能力など）を最大限に活用できるよう

にすることである。このようなマン・マシン・システムによる問題解決技術は今後のコンピュータ利用の主流となってゆくことが予想される。このようなシステムを実現するためには人とコンピュータの間の情報交換が重要なことは言うまでもないが，グラフィックス技術はこの面でも重要なものになることは明らかである。

## 5　グラフィックス・ソフトウエアとその標準化

マン・マシン・システムに要求されるのは人と機械の協力のもとでの問題解決能力であるが，通常，このために高度の応用プログラムが必要である。応用プログラムは単にデータの変換を高速に行うだけでなく，複雑な問題の表現問題すなわちモデル化の問題を含むようになる。これを簡略化して示すと，図形を媒介とするインタラクティブな問題解決の手順は

①　　②　　③　　④

図形表示⇄形状の具体的表現⇄形状の抽象化表現⇄抽象化された問題の解析

のように表わされる。①はディスプレイ装置による実際の図形の表示，②は具体的な形状，たとえば直線や円弧などのコンピュータ内表現，③は外的な形状が表わす問題の意味を表わす情報の表現，④は抽象化された（コンピュータ・プログラムで処理できる形式の）問題の表現とその解決の手続きを表わす。一般に③は論理的なデータ構造という形式をとり，しばしばデータベースである。④はそのデータ構造の変換を含む処理手続きで，応用プログラムと呼ばれる。②が2節で述べた基礎技術で作られる形状表現であることは言うまでもない。

③，④は目的ごとに異なるのにたいし，②は多くの応用プログラムにたいして共通である。この部分は基礎的な部分であるとはいえ，実際にはプログラムが複雑で量も多い。従来，この部分を含めた応用ソフトウエアが各種開発されたが，ソフトウエアの標準がないため，可搬性が乏しく，広く利用されるまでにいたらなかった。最近，グラフィックス・システムの標準化の動きが漸く活発になってきたが，グラフィック・システムの標準化に際しての問題点はディスプレイ装置というハードウエアの構造により影響されることである。この問題にたいしては②の部分を個々のデバイスと密接に関連するデバイス・ドライバと標準グラフィックス・パッケージに分け，デバイスの影響はこれに付随するデバイス・トライバのみにとどめるという形式が取られるようになりつつある。いずれにしろユーザにとって，コンピュータ・グラフィックスの利用上欠くことのできないグラフィックス・パッケージが標準化され，使い易くなることは極めて重要なことである。

## 6　むすび

コンピュータ・グラフィックスについて駆け走で述べてきた。コンピュータ・グラフィックス技術が今後，設計などの分野で一層重要な役割を示すことは明らかである。拙稿が多くの関連諸分野で研究・設計・開発などに従事している方々に，更に深い勉学のための手引きともなれば望外の幸である。

### 参考文献


(1) Ahuja, D.V. and Coons, S.A. : "Geometry for Construction and Display" IBM・Syst. J.7, 3/4 p.185 (1968)

(2) Forrest, A.R. : "Computational Geometry", Proc. R. Soc. London A 321, pp.187～195 (1971)

(3) Kubert, B., Szabo, J. and Giulieri, S. : "The Perspective Representation of Functions of Two Variables", J. Assoc, Comp. Mach, 15, 2, pp.193～204 (April. 1968)

(4) Newman, W.M. and Sproull, R. : " Principles of Interactive Computer Graphics ", Mc Graw－Hill Book Co. 2nd ed. (1979)

(5) Rogers, D.F. and Adams, J.A. : " Mathematical Elements for Computer Graphics " Mc Graw－Hill Book Co. (1976)

(6) Adams, J. Al : " A Comparison of Methods for Cubic Splime Curve Fitting ", Comput. Aided Des., 6, pp. 1～9 (1974)

(7) Bezier, P.E. : " Numerical Control Mathematics and Applications ", John Wiley & Sons, Inc. (1972)

(8) Gordon, W.J. and Riesenfeld, R.F. : " Bernstein－Bezier Methods for the Computer－Aided Design of Free Form Curves and Surfaces ", J. ACM, 21, 2, pp. 293～310 (1974)

(9) Coons, S.A. : " Surfaces for Computer Aided Design of Space Forms ", MIT Project MAC TR－41 (1967)

(10) 穂坂 衛：" 自由形状曲面の理論と設計 ", 情報処理, 8, 3 pp. 65～72 (昭42)

(11) 穂坂 衛, 黒田 満：" CADにおける曲線曲面の創成について ", 情報処理, 17, 12, pp. 1120～1127 (1976)

(12) 穂坂 衛：" コンピュータ・グラフィックス ", 産業図書, (昭49)

(13) 大須賀節雄, 編者：" インタラクティブ・ディスプレイ・システム ", システム社 (昭49)

(14) Barnhill, R.E. and Riesenfeld, R.F. : " Computer Aided Geometric Design ", Academic Press, New York (1974)

(15) 電子通信学会誌：" ディスプレイ－最近の進歩－ 特集号 ", 61, 11, (1968)