

## 3次元表示のプロセス

川井孝之 : 海洋情報課

3 Dimensional Processing.

Takayuki Kawai : Oceanographic Data and Information Div.

### 1. はじめに

コンピュータグラフィックス（以下略CG）は、大きな可能性を秘めた新しいメディアとしてさまざまな分野に浸透しつつある。そして、これらの背景には、CGにより出力された図の持つ情報量の多さに加え、ハード、ソフトウェア技術の進歩によるプログラミングの単純化が掲げられるであろう。

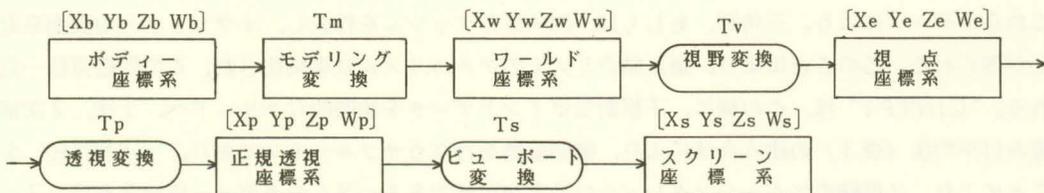
特に後者においては、ここ数年の目覚ましい進歩により、以前一部の専門家に任せられていた複雑なプログラミングを、パッケージ化されたサブルーチンを使用することにより、図形処理に関する知識をまったく持たないユーザーが代行することを可能とした。

現在、水路部で使用されているグラフィックパッケージ、“RASPAK”<sup>\*1</sup>、及び“GEOPAK”<sup>\*2</sup>は、この様なユーザー指向型のグラフィックパッケージであり、FORTRAN (V) モード環境下で動作する各々の図形処理サブルーチンが用意されている。これらのパッケージを使用することにより、豊富なカラー表示と共に、従来の2次元（以下略2D）を表現に加え、3次元（以下略3D）による新しい表現方法が可能となった。

ここでは、これらのグラフィックパッケージの特徴である3D表示を例に取り、“RASPAK”、“GEOPAK”の使用例を示すと共に、そのブラックボックスであるサブルーチンのアルゴリズムを、基本的な図形処理論と対応しながら紹介することにした。このことにより、これらサブルーチンの機能を理解し、又ソフトウェア開発の参考となれば幸いである。

### 2. 3次元表示のプロセス

3D表示は、我々が日常目にする3D空間の表現が可能のため、よりリアルな表示が期待出来る。



第1図 3D表示のプロセス

\* 1 “RASPAK”

基本的なラスターイメージを取り扱うパッケージであり、ベクター～ラスター変換、エリアシェイティング、ポイントプロット、カラー指定、テキストプロット等を行う。

\* 2 “GEOPAK”

“RASPAK”下で使用するマッピングパッケージであり、等高線図、透視図等の作成を行う。

このことにより、CGはCAD等に見られる様に、その適用範囲は、加速度的な拡がりを見せたといえよう。それでは、3D表示はどのようなアルゴリズムによって2D空間へ投影されるのであろうか。第1図を参照していただきたい……。この様に、3D表示は、2D表示とは異なるプロセスを遂て2D空間であるディスプレイ、プロッター上へ投影される。これは、“GTRANS”により行なわれるが、ここでは、それより下のレベルによるアルゴリズムを、3D表示における座標系、及びその座標変換理論を示しながら紹介することとした。

## 2-1 モデリング

2D表示においては、グローバルな座標系に直接図形を定義するのが普通である。しかし、3D表示で取り扱う3Dオブジェクトを対象とした場合、前記の様な座標系に、3Dオブジェクトを直接定義することは、大変困難な作業となる。そこで、3D表示においては、2D表示のマスター座標系に相等するローカルな(オブジェクトに固有な)座標系を設け、この座標上で3Dオブジェクトを定義し、その後モデリング変換を施して3Dシーンを定義するのが一般的な手順となる。それでは、このローカルな座標系であるボディー座標系によって、3Dオブジェクトはどの様に定義(モデリング)されるのであろうか。

このモデリング技術はCG分野よりはむしろ数値制御(Numerical Control, 略してNC)工作機分野においてつかわれ、形状処理工学に見られる様に数々の方法が考え出されている。

球の様な定形的なオブジェクトの場合、簡単な数式により、その表面上に位置する点は無限個求めることが出来る。そこで、これらの点群データを、後に紹介する座標変換を用い、プロッタ上に作画したとしよう。我々は、この出力された図を見た場合、そこに“球”を見ることが出来る。だが、これは球をモデリングしたことにはならない。なぜならば、我々はその図を見て、経験的に“球”として見ているのに対し、コンピュータにとっては、球の表面上に分布した、単なる点群として見ているからである。つまり、これらのデータは、点の位置情報( $P_x, P_y, P_z$ )のみであり、体積を有する3Dオブジェクトを、点だけでは定義出来ないからである。そこで、3Dオブジェクトをモデリングする場合、これらの点群データを、何らかの方法により結合することで面を作り、これらの面の集合体として3Dオブジェクトを定義するのが、一般的な方法である。つまり、3Dオブジェクトをモデリングするためには、その大きさ(座標値)  $70i$ 、形状(頂点結合リスト)が新しい情報として必要となる。

我々が、取り扱うデータは、水深データ等に見られる様に、ランダムなポイントデータである場合が多く、又そのデータの表わすオブジェクトの形状(海底地形)も不定形な場合が殆んどである。しかし、この様なデータを用い、前述した様な頂点結合リストを作るのはかなり複雑なアルゴリズムを用する。従って、通常は、これらのデータにより、三角形、もしくは、四角形のメッシュを作成し、オブジェクトの表面を近似する方法が取られる。このことにより、頂点結合リストのアルゴリズムは単純化され、この問題点は一気に解消される。“GINTP1”は、この様に、不規則なポイントデータを規則的なグリッドへ、1次、2次補間、及び重み付平均法(※3)の組み合わせにより、補間変換を行なうサブルーチンであり、“GSMTH”を使用することにより、3段階のスムージングレベルの設定が可能である。又このボディー座標系を用いることにより、3Dオブジェクトの体積を計算することが可能である。

### ※3) 重み付平均法

既知とする何箇所かのメッシュ  $[x_1, y_1], [x_2, y_2] \dots [x_n, y_n]$  におけるデータ  $f_1, f_2 \dots f_n$  を使用し、未知であるメッシュ  $[x, y]$  におけるデータ  $f$  を、滑らかな曲面で補間する方法であり、(1)式が用

いられている。

$$f(x, y) = \frac{\sum_{i=1}^n (f_i / \sqrt{(x_i - x)^2 + (y_i - y)^2})}{\sum_{i=1}^n (1 / \sqrt{(x_i - x)^2 + (y_i - y)^2})} \dots\dots\dots (1)$$

ただし、データ  $f_i$  が与えられたメッシュにおいては、(1)式によらず  $f_i$  とする。式(1)の分子は、点  $[x, y]$  での高さを求めるとき、既知点に近い点であるほど大きな重みを付けるようにしたものであり、分母は、そのデータに対する重みの総和が1となる様に、正規化するための項である。この式が定義域内で連続であり、滑らかな曲面を創成することは容易に想像出来る。しかし、この式の大きな欠点として既知点における曲面の接平面がすべて水平になってしまうことが掲げられる。つまり

$$\begin{aligned} & \frac{\partial}{\partial x} f(x, y) \Big|_{x=x_i, y=y_i} \\ &= \frac{\partial}{\partial y} f(x, y) \Big|_{x=x_i, y=y_i} = 0 \end{aligned}$$

となり、このことは既知点では、すべてが滑らかな頂上や、滑らかな凹みの底であるか、あるいは坂の途中の水平な場所(階段における舞い場の様な所)になってしまうことを意味する。このことにより、起伏に富んだ海底地形などの場合、丸められてしまい、実際のイメージとは異なる図形が表示されてしまう可能性があることに留意されたい。

## 2-2 モデリング変換

3D オブジェクトの形状を定義したボディー座標系から3D ワールド座標系への変換は、モデリング変換と呼ばれる変換により行なわれる。これは、オブジェクトの動きや向きに対しての変換であり、回転、平行移動、及びスケールングにより、以下に示すアフェイン変換により定義される。ここで、3D ワールド座標系とは、ボディー座標系に対して、グローバルな座標系であり、ユーザーの使用するワークエリアの様なものである。

平行移動: T

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{pmatrix}$$

スケール変換: S

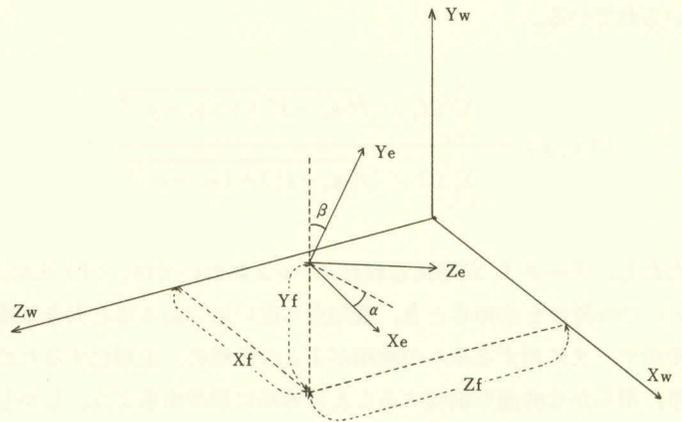
$$S = \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$x, y, z$  各軸に対する変換:  $R_x, R_y, R_z$

$$R_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x & 0 \\ 0 & -\sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R_y = \begin{pmatrix} \cos\theta_y & 0 & -\sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R_z = \begin{pmatrix} \cos\theta_z & \sin\theta_z & 0 & 0 \\ -\sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

3D ワールド座標系を  $[x_w, y_w, z_w, 1]$ , ボディ座標系を  $[x_b, y_b, z_b, 1]$  とすると,

$$[x_w, y_w, z_w, 1] = [x_b, y_b, z_b, 1] \cdot \underbrace{S \cdot R_x \cdot R_y \cdot R_z \cdot T}_{T_m} \dots\dots \text{モデリング変換}$$



第2図

### 2-3 視野変換

3D ワールド座標系で定義した3D シーンを見るための変換であり, 前述した各座標系と異なり, 視点を原点とし, 視線をZ軸の正方向に向けた左手系の座標系である。3D ワールド座標系との関係を第2図に示す。xe軸はxw-zw平面に平行にとるため, 回転は2回で済ませ, 最後にZ軸の向きを反対にすることにより, 右手系→左手系へと変換する。

視線をZ軸の正方向に向けた左手系の座標系である。3D ワールド座標系との関係を第2図に示す。xe軸はxw-zw平面に平行にとるため, 回転は2回で済ませ, 最後にZ軸の向きを反対にすることにより, 右手系→左手系へと変換する。

$x_w, y_w, z_w$  軸方向にそれぞれ  $x_f, y_f, z_f$  だけの平行移動:  $T_1$

$$T_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -x_f & -y_f & -z_f & 1 \end{pmatrix}$$

視点の回りに  $-\alpha$  だけ回転:  $T_2$

$$T_2 = \begin{pmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

さらに視点  $x_e$  軸の回りに  $-\beta$  だけ回転:  $T_3$

$$T_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & \sin\beta & 0 \\ 0 & -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

右手系→左手系への変換:  $T_h$

$$T_h = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

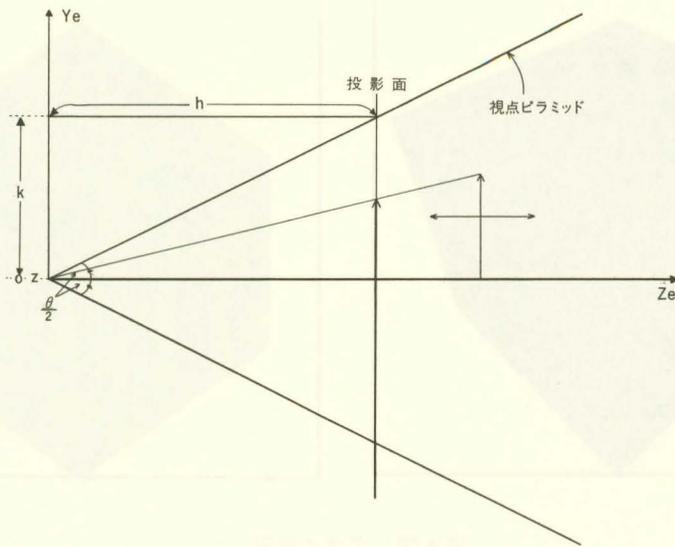
視点座標系を  $[x_e, y_e, z_e, 1]$  とすると,

$$[x_e, y_e, z_e, 1] = [x_w, y_w, z_w, 1] \cdot \underbrace{T_1 \cdot T_2 \cdot T_3 \cdot T_h}_{T_r} \dots\dots \text{視野変換}$$

これらの機能は, サブルーチン "GVUANG" 及び "GEYE" に相当する。

### 2-4 透視変換

視点を中心として記述されている視点座標系を2D空間である表示装置へ投影することを透視投影といい, この変換を行なうのが透視変換である。これらの変換式は, 第3図からわかる様に, 表示装置上の座標を,  $[x_p, y_p, z_p]$  とすると次式となる。



第3図

$$\left. \begin{aligned} x_p &= \frac{h}{k} \cdot \frac{x_e}{z_e} \\ y_p &= \frac{h}{k} \cdot \frac{y_e}{z_e} \\ z_p &= 1 - \frac{h}{z_e} \end{aligned} \right\} \dots\dots\dots (2)$$

ここで、(2)式、及び第3図より、透視図における視角を $\theta$ とすると、

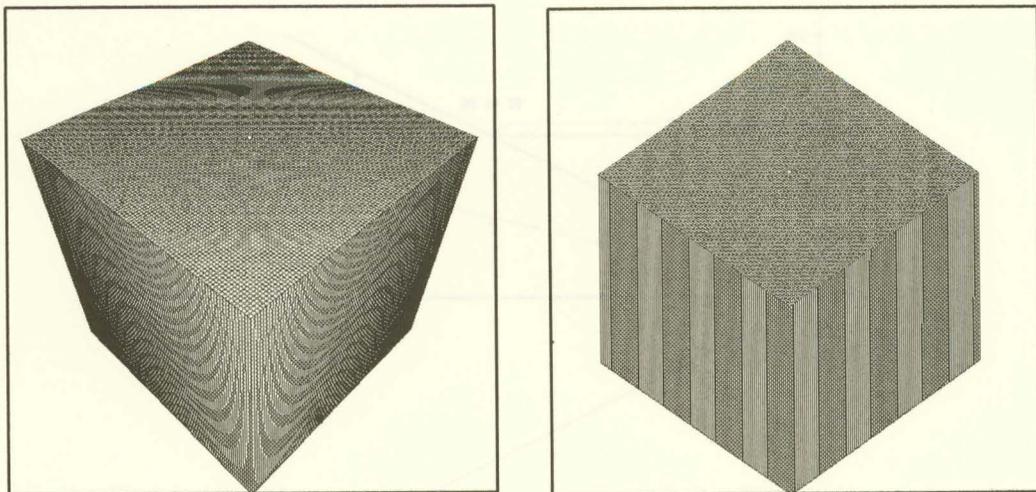
$$\tan(\theta/2) = k/h \text{ となる。}$$

そこで、(2)式の第2式を、 $z_e$  に関して微分すると、

$$\begin{aligned} \frac{dy_p}{dz_e} &= -\frac{h}{k} \cdot \frac{y_e}{z_e^2} \\ &= -\frac{k}{h} \cdot \frac{y_p^2}{y_e} = -\tan \frac{\theta}{2} \cdot \frac{y_p^2}{y_e} \end{aligned} \dots\dots\dots (3)$$

となる。これは、 $z_e$  に対する  $y_p$  ( $x_p$  についても同様に) の変化を示す。つまり投影図の奥行きの強調度を表わしている。この機能はサブルーチン "GDIST" により行うことが出来る。しかし、(3)式より、 $k/h$  の値を  $1/2$  倍にすると投影図の大きさは2倍になってしまい、このため、投影図の大きさを一定に保つためには、視点の位置を、2倍だけ視点から遠ざけなければならない。これは、3Dワールド座標系における視点の位置  $z_e$  値を変化させるのが簡単な方法である。

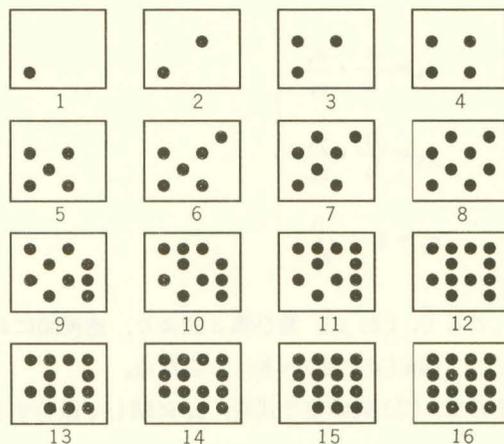
第4図に、"GDIST" によるデプス効果を示す。(a)図はオブジェクトに目を近づけた場合に見られる図(広角レンズにより得られる像)に相当し、(b)図はオブジェクトから遠ざかって見た場合(望遠レンズにより得られる像)に相当する。



第4図 デプス効果

### 3. “RASPAK”におけるカラーモデル

“RASPAK”では、3つの2桁の数字によって構成されるカラーコードにより、豊富なカラー表示が出来る。3つの2桁の数は各々0～16のレンジを有し、対応する基本3原色のシェイドやパターンの違いにより、 $17 \times 17 \times 17 = 4,913$ 色の相合せが可能である。これは、ハード的に見ると、第5図に示す様な干渉縞が少なくなる様に工夫された $4 \times 4 = 16$ ピクセル（画素）によるビットプレーンを設け、カラープロッターにより、シアン、マゼンダ、イエローの各インクを吹き付けることにより、中間色を出している。これらのカラーテーブルは、サブルーチン“GCOLOR”により設定され、又サブルーチン“GCMODE”の使用により、4種類のカラーモデルの選択が可能である。



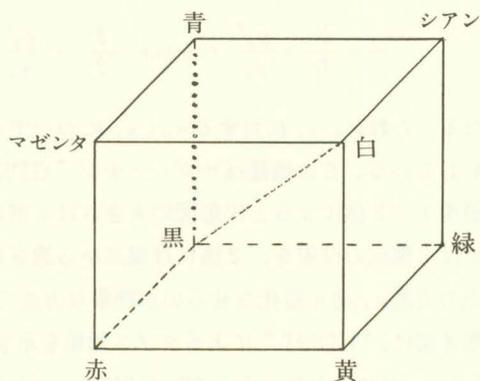
第5図

#### 3-1 RGB カラーモデル

赤、緑、青よりなるカラーモデルは、直交座標系を用いて表わすと、第6図のようになる。RGBでは、各原色の個々が寄与して一諸に加えられた結果を作るために、加法原色と呼ばれる。

#### 3-2 CMY カラーモデル

赤、緑、青の補色であるシアン、マゼンダ、イエローは、白色光からあるカラーを減ずることにより、減法原色と呼ばれる。これは、面がシアン色のインクで塗られ



第6図 RGB カラー立方体

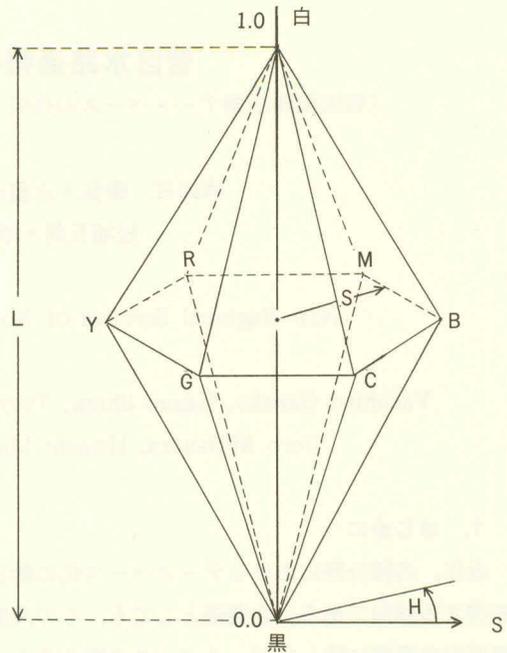
ている場合、その面からは、赤、緑、青の白色光より、補色である赤を減じ、従って加法原色の観点からは、青プラス緑となる。RGBとCMYとの関係は次式によって表わされる。

$$\begin{pmatrix} C \\ M \\ Y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

### 3-3 HLS カラーモデル

Tektronix で使用された Ostwald [OSTW 31] カラーシステムに基づく HLS は、第 7 図で示される様な双六角錐部分空間によって表わされる。色相は、青を 0° とする垂直軸の回りの角度、明るさは垂直軸、又飽和は中心軸から双六角錐の側面までの割合で表わされる。

HLS カラーモデルでは、反対側 (180°) の色相では、補色を表わし、又中心軸上では、グレイスケールとなる。RGB、CMY モデルは、ハードウェア向きであるのに対し、HLS モデルは直感に訴えることに基づいて、ユーザー向きといえる。



第 7 図 双六角錐 HLS カラーモデル

## 4. むすび

ここで紹介したサブルーチンは、“RASPAK”及び、“GEOPAK”のほんの一部にすぎない。紙面の都合上各サブルーチンの使用方法を紹介することが出来なくて残念である。興味のある方は、海洋情報課にマニュアルがあるので参照されたい。これら豊富な図形処理サブルーチンを使用し、海底地形鳥瞰図にとらわれない、新しい表現を期待する次第である。

## 報告者紹介



Takayuki Kawai

川井孝之 昭和62年3月現在、  
本庁水路部海洋情報課海洋情報官付