

三角形面素生成による3次元物体の陰影画表示

王来生* 横井茂樹**
三重大学 工学部*

鶴岡信治* 三宅康二**
名古屋大学 工学部**

1. まえがき
最近、ラスタ型G Dの普及に伴い、物体の陰影画表示技術に対する要求が高まっている。物体を陰影画表示する場合、まず物体の形状を記述することが必要となる。その方法としては、基本立体(球、円柱、楕円体など)の合成、曲面パッチの接続による表現、陰関数表現など種々の方法があるが、物体表面を三角形面素の貼り合わせにより表現する方法は、汎用性のある有効な方法である⁽¹⁾。
ところで、物体を表示する場合、表示目的に応じていくつかの選択が必要となる。たとえば、中心投影と平行投影、点光源と平行光源、隠れ面処理の必要性、材質感(金属状、ゴム状、ガラスのツク状、透明状等)などである。本研究では、上述の諸条件をパラメータ等で選択することにより、適当な条件の基で、物体上の三角形面素を生成

成する汎用ルーチン群の開発を試み、これまで、三角形面素を基本とする汎用の物体表示システムとしては木村⁽²⁾が開発したものがあるが、本研究では、投影の種類、光源の種類、クリッピング処理の導入、透明、半透明表示機能の付加など、より広汎な機能を含んだシステムの開発を意図している。本ルーチン群により、利用者が、物体を三角形面素のはり合わせの形で表現すれば、容易に任意の条件で物体を表示することが可能となる。本文では、諸条件に対し、三角形面素を生成する基本アルゴリズムについて述べる。また、これらを、実際に利用して、多面体の表示及び地形図、二変数関数データの表示を行った結果についても示す。

2. 処理手順の概要

図1に処理手順の流れ図を示す、各

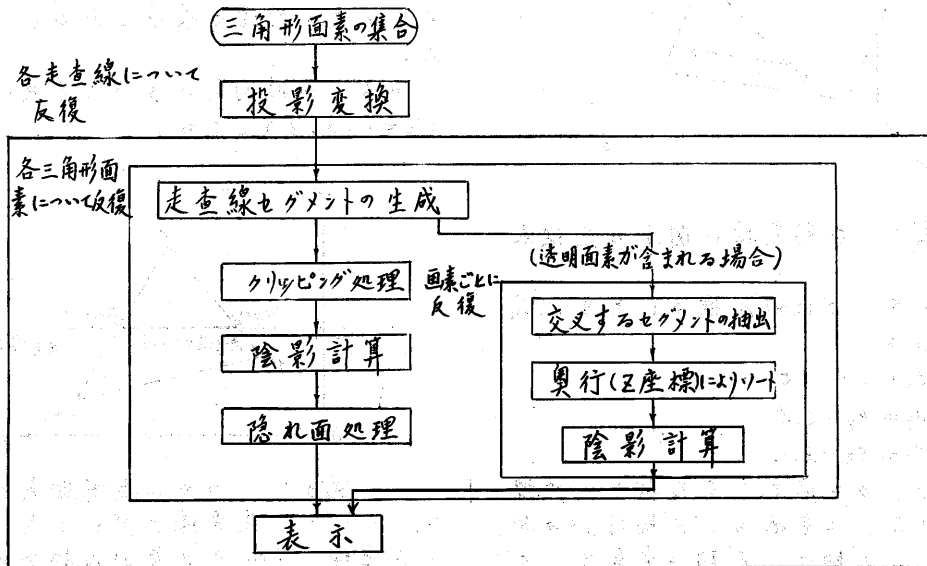


図1. 処理手順の概要

三角形面素の三頂点の空間座標が与えられると、最初にこれらの座標値すべてを投影変換する。以後、各走査線ごとに、すべての三角形面素に対し、走査線セグメントの生成から、隠れ面処理までが施され、一走査線ずつ表示される。

3. 処理内容

本節では、各ステップでの処理の内容の詳細について説明する。

3.1 投影変換

物体の定義される座標系(旧座標系と呼ぶ)で表現された点 $P = (x, y, z)$ をある視点 $E = (e_x, e_y, e_z)$ から眺めたときのその点の像は、視線方向ベクトルに垂直な面に投影して得られる。

視線方向ベクトルを $l = (-l_1, -l_2, -l_3)$ とする。投影面は視点から距離 t だけ離れた位置にあり、視線に垂直な面とする。ここで、視線と投影面の交点 $O' = (o_x, o_y, o_z)$ を新しい座標系の原点とする。

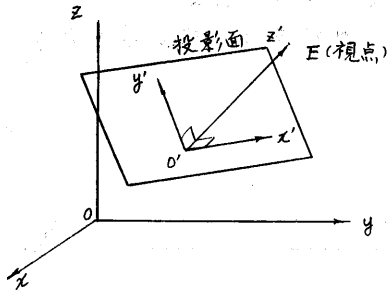


図2 投影変換に関する座標系

このとき、 O' は旧座標系により次式で表される。

$$\begin{cases} o_x = e_x - l_1 t \\ o_y = e_y - l_2 t \\ o_z = e_z - l_3 t \end{cases}$$

ここで、 O' を原点とする新しい座標系を次のように定める。 z' 軸は $O'E$ の向きにとり、 x' 軸は、 z' 軸に垂直で、 $O-x-y$ 平面に平行とし、 y' 軸は z' 軸

と x' 軸に垂直にとする。

このとき、旧座標系での点 $P(x, y, z)$ は新座標系では次の $P' = (x', y', z')$ により表される。

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} -l_1/t & l_1/t & 0 \\ -l_2/t & -l_2/t & t \\ l_1 & l_2 & l_3 \end{pmatrix} \begin{pmatrix} x - o_x \\ y - o_y \\ z - o_z \end{pmatrix} \quad (1)$$

また、視点は、新座標系では $(0, 0, t)$ で表される。

ここで、 x' 軸 y' 軸を表示画面の X 軸、 Y 軸に対応させると、各点 $P' = (x', y', z')$ は、視点 $E' = (0, 0, t)$ と P' を結ぶ直線と $O'-x'-y'$ 平面との交点に投影される。これを求めると

$$\begin{cases} X = \frac{t}{t-z'} x' \\ Y = \frac{t}{t-z'} y' \end{cases} \quad (2)$$

の点に投影されることになる。

上式は中心投影の変換式を表している。ここで、平行投影の場合は、 $t \rightarrow \infty$ とおくと、 $X = x'$, $Y = y'$ となり、(1)式で求められた x' 座標値、 y' 座標値そのものを左用いればよい。

3.2 走査線セグメントの生成

各三角形面素の3頂点は投影変換により2次元座標で表現されることになる。

ある三角形面素の3頂点を $P_i = (x_i, y_i)$ ($i = 1, 2, 3$) とする。

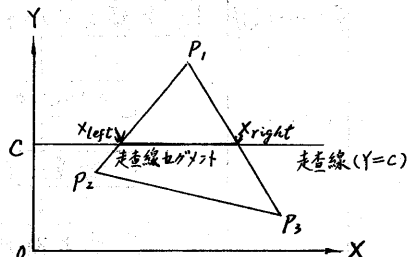


図3 一つの三角形面素の走査線セグメント

走査線 $Y = C$ と交わる部分のセグメントを求めると、次のようにして

行われる

三角形の3つの辺を $\overline{P_1P_2}$, $\overline{P_2P_3}$, $\overline{P_3P_1}$ をそれぞれ L_1 , L_2 , L_3 で表す。

走査線ごとの処理に入る前に、予め、各辺及び3辺全体のY座標の最小値と最大値を求めておく。

$$L_1: \begin{cases} y_{\min}^{(1)} = \min(Y_1, Y_2) \\ y_{\max}^{(1)} = \max(Y_1, Y_2) \end{cases} \quad (3)$$

$$L_2: \begin{cases} y_{\min}^{(2)} = \min(Y_2, Y_3) \\ y_{\max}^{(2)} = \max(Y_2, Y_3) \end{cases} \quad (4)$$

$$L_3: \begin{cases} y_{\min}^{(3)} = \min(Y_3, Y_1) \\ y_{\max}^{(3)} = \max(Y_3, Y_1) \end{cases} \quad (5)$$

辺全体:

$$\begin{cases} y_{\min}^T = \min(Y_1, Y_2, Y_3) \\ y_{\max}^T = \max(Y_1, Y_2, Y_3) \end{cases} \quad (6)$$

さらに、各辺を直線の方程式で表現したときの係数を求めておく。

($x = a_k y + b_k$ として表現)

$$L_1: a_1 = \frac{X_2 - X_1}{Y_2 - Y_1}, b_1 = X_1 - Y_1 a_1 \quad (7)$$

$$L_2: a_2 = \frac{X_3 - X_2}{Y_3 - Y_2}, b_2 = X_2 - Y_2 a_2 \quad (8)$$

$$L_3: a_3 = \frac{X_1 - X_3}{Y_1 - Y_3}, b_3 = X_3 - Y_3 a_3 \quad (9)$$

ここで、 $|Y_{i+1} - Y_i| < \epsilon$ のときは上記計算は行わず、 a_k に特別な値を入れておく。

以上の数値を各三角形面素について求めておいたとき、走査線 $Y = c$ の中で、三角形面素に含まれる部分(走査線セグメント)は次のようにして求められる。

まず、 $c < y_{\min}^T$ または $y_{\max}^T < c$ のときは、三角形が走査線と交わらない。

走査線と交わる場合、走査線と交わる辺は、 $y_{\min}^{(i)} \leq c \leq y_{\max}^{(i)}$ を満たすものであり、これは通常2つ存在する。上の関係を満たす2つの辺 L_k, L_l に対し、

$Y = c$ との交点のX座標を計算する。

$$X^k = a_k \cdot c + b_k \quad (10)$$

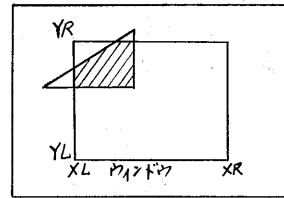
$$X^l = a_l \cdot c + b_l \quad (11)$$

但し、 $|Y_{i+1} - Y_i| < \epsilon$ の関係を満たすとして a_k に特別な値が与られている辺は上記計算の対象としない。

このとき、セグメントの左端と右端のX座標は次のように求められる。

$$X_{\text{left}} = \min(X^k, X^l) \quad (12)$$

$$X_{\text{right}} = \max(X^k, X^l) \quad (13)$$



スクリーン

図4. クリッピング処理

3.3 クリッピング処理

物体を表示する場合、スクリーン上の指定された範囲(ウィンドウ)内に入る部分のみを表示することが必要となる。ウィンドウ外の部分を消去する(表示しない)処理はクリッピング処理といわれるが、ここでは、セグメントの座標値を修正する方法により行った。

ウィンドウ(表示領域)を $X_L \leq X \leq X_R$, $Y_L \leq Y \leq Y_R$ で定められる領域とする。まず、Y方向のクリッピングは、上述のセグメントの計算において、 $Y_L \leq Y \leq Y_R$ の範囲の走査線のみについて行えば、当然それ以外の部分は表示されない。一方、X方向のクリッピング処理は、求められた各三角形面素のセグメントの座標値を次のように修正することにより行える。

三角形面素において既に求められているセグメントの左端 X_{left} と右端 X_{right} に対し、

①、 $X_{\text{left}} > X_R$ あるいは $X_{\text{right}} < X_L$ のとき、

このセグメントは、ウィンドウ外に存

在するので、表示対象としない。

②、上記以外するとき

$$X'_{left} \leftarrow \max(X_{left}, X_L)$$

$$X'_{right} \leftarrow \min(X_{right}, X_R) \text{ とし,}$$

X'_{left} , X'_{right} を新たにセグメントの左端と右端にする。

3.4 陰影計算

陰影値の計算は、面素の法線ベクトルと視線方向ベクトル、光源方向ベクトルを用いて行われる。

まず、これらのベクトルの導出法を示す。なお、これらのベクトルはすべて新座標系を用いて行う

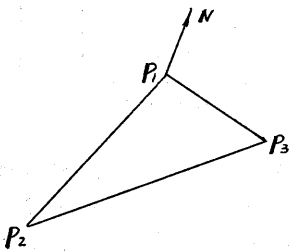


図5. 三角形面素の法線ベクトル

①、面素の法線ベクトル

三角形面素の3頂点を $P_1=(x_1, y_1, z_1)$, $P_2=(x_2, y_2, z_2)$, $P_3=(x_3, y_3, z_3)$ とすると明らかに、面素の法線ベクトル $N=(n_x, n_y, n_z)$ は

$$N = (P_2 - P_1) \times (P_3 - P_1) \quad (\text{外積})$$

すなわち、

$$n_x = \sum_{i=1}^3 (y_i - y_j)(z_i + z_j) \quad (14)$$

$$n_y = \sum_{i=1}^3 (z_i - z_j)(x_i + x_j) \quad (15)$$

$$n_z = \sum_{i=1}^3 (x_i - x_j)(y_i + y_j) \quad (16)$$

但し、

$$j = \begin{cases} i+1 & (i \neq 3) \\ 1 & (i = 3) \end{cases}$$

として求められる。

視線方向ベクトルと光源方向ベクトルは、それぞれ投影、光源の性質によって計算方法が異なる。

②、視線方向ベクトル

平行投影のときは、視線方向ベクトルはどこでも一定で、与えられたベクトル L (または $-L$) を用いる。

中心投影のときは、三角形面素の各点ごとに視線方向ベクトルが異なってくる、このときはセグメントの1点(画素)ごとに視線方向ベクトルを計算する必要がある。

今走査線 $Y=C$ に対する三角形面素のセグメントの左端と右端 X_{left} , X_{right} であるとする。ここで、表示画面上の (X_{left}, C) から (X_{right}, C) までの各画素 (X_n, C) に対して、視線方向ベクトルは、 $(X_n, C, 0)$ から $(0, 0, t)$ へ向かうベクトルとして、 $(-X_n, -C, t)$ により、与えられる。

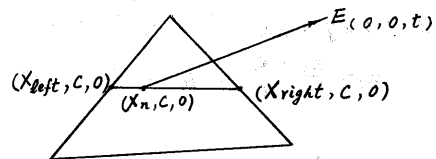


図6. セグメント上のその視線ベクトル

なお、三角形面素の大きさが視点からの距離に比べて小さいときは、3つの頂点の重心位置 $G = \frac{1}{3}(P_1 + P_2 + P_3)$ における視線方向ベクトル $E-G$ を用いて、三角形面素中のすべての点で一定として、このベクトルを用いて計算することもある。($E=(0, 0, t)$)

③、光源方向ベクトル

平行光源の場合、どの点でも一定の光源ベクトルをもつから、与えられたベクトルを用いる。

点光源の場合、まず、走査線 $Y=C$ のセグメントの中の各点 (X_n, C) に対して、これらの空間座標値 $(X_n, C, 0)$

と視点の座標値 $(0, 0, t)$ を結ぶ線分と三角形面素との交点の座標 (P_x^n, P_y^n, P_z^n) を求める, これは次の式で与えられる.

$$T = \frac{n_x(X_n - x'_i) + n_y(C - y'_i) - n_z z'_i}{-n_x x'_i - n_y y'_i + n_z(t - z'_i)} \quad (17)$$

$$P_x^n = \frac{X_n}{1 - T} \quad (18)$$

$$P_y^n = \frac{C}{1 - T} \quad (19)$$

$$P_z^n = \frac{-t \cdot T}{1 - T} \quad (20)$$

ここで, (x'_i, y'_i, z'_i) は, 三角形面素の任意の頂点の座標である.

これと, 光源 (l_x, l_y, l_z) を結ぶベクトル $(l_x - P_x^n, l_y - P_y^n, l_z - P_z^n)$ を各点の光源ベクトルとする.

上記3つのベクトルを用いて表面の陰影計算をする, これは以下の式による.

①. ゴム状またはプラスチック状(拡散反射+鏡面反射)

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} R_1 \\ G_1 \\ B_1 \end{pmatrix} + w(\cos \alpha)^n \begin{pmatrix} 1 - R_1 \\ 1 - G_1 \\ 1 - B_1 \end{pmatrix} \quad (21)$$

$$\begin{pmatrix} R_1 \\ G_1 \\ B_1 \end{pmatrix} = \left\{ d + (1 - d) \cos \theta \right\} \begin{pmatrix} R_0 \\ G_0 \\ B_0 \end{pmatrix}$$

式(21)において, $0 \leq R_0, G_0, B_0 \leq 1$ は物体の色, d は背景光の割合 ($d \ll 1$), θ は光源と法線の間の角度, α は視線と反射光の間の角度, w はハイライト強さ ($0 \leq w \leq 1$), n は表面の光沢度 (n は15程度) である. ②. 金属状の材質(鏡面反射のみ, 通常は複数光源).

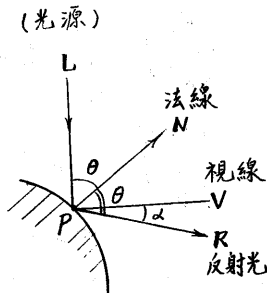


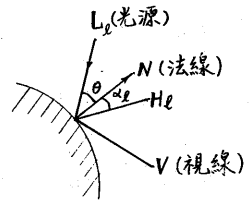
図7. 物体表面の基本ベクトル

ような材質ではほぼ完全拡散面とみなされ, この場合は $w=0$ とすればよい. 一方, プラスチックでは拡散反射のほか白いハイライトがつく, この場合は $w=1$ とすればよい.

②. 金属状の材質(鏡面反射のみ, 通常は複数光源).

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \left\{ d + (1 - d) \sum_{l=1}^L a_l (\cos \alpha_l)^{n_l} \right\} \begin{pmatrix} R_0 \\ G_0 \\ B_0 \end{pmatrix} \quad (22)$$

式(22)において, $0 \leq R_0, G_0, B_0 \leq 1$ は物体(金属)の色, d は背景光の割合 ($d \ll 1$), a_l は l 番目の光源の強さ ($0 \leq a_l \leq 1$), α_l は l 番目の光源ベクトルと視線ベクトルの二等分ベクトルと



$$H_l = \frac{L_l + V_l}{\|L_l + V_l\|}$$

図8. 金属表面の基本ベクトル

法線ベクトルの間の角度, n_l は表面の光沢度, n_l は10程度である.

3.5 隠れ面処理

予め, 三角形面素の奥行の順序が定められる場合(たとえば, 後述の地形図表示の場合等), 遠くの面素から順に画面上に重ね書きすればよい, しかし, 一般的な場合には, スキャンラインZバッファ法⁽⁵⁾を用いる.

最初一本の走査線上の画素数に対応する陰影値(輝度値)用の配列(カーの場合には3つ)の値に, 奥行値(Z座標値)を記憶する配列(Z-バッファと呼ばれる)を用意しておく, 各走査線ごとに以下の処理を行う.

初期値として, Z-バッファに十分小さい値(遠い奥行値に対応する)を代入しておき, 陰影値には背景の値を与えておく. 各三角形面素ごとに求められたセグメントに対し, セグメント上の各

画素について、その画素のZ座標値が既に記憶されているZ座標値より大きい(視点に近い)場合、Z座標値及び輝度値を、そのセグメントの値におきかえ、という作業をくり返し、全面素のセグメントについて行う。その結果、配列中の各画素には視点に最も近い画素(または背景)の輝度値が与えられる。

ここで、各セグメントのZ座標値は、各画素ごとに式(20)を用いて計算する必要がある。しかし、平行投影の場合には、セグメントの両端点でのZ座標値を用いて、次のように線形補間すればよい(走査線 $Y=C$)。

左端点 (X_{left}, C, Z_{left})

右端点 $(X_{right}, C, Z_{right})$

セグメント内の画素 (X_n, C) に対するZ座標値は:

$$Z_n = \frac{Z_{right} - Z_{left}}{X_{right} - X_{left}} (X_n - X_{left}) + Z_{left} \quad (23)$$

により計算される。

ここで、 X_n は隣接する点 (X_{n+1}, C) ($X_{n+1} = X_n + 1$) において、Z座標値 Z_{n+1} は、

$$\left. \begin{aligned} Z_{n+1} &= Z_n + \alpha, \\ \alpha &= \frac{Z_{right} - Z_{left}}{X_{right} - X_{left}} \end{aligned} \right\} \quad (24)$$

で与えられ、セグメント内の各座標のZ座標値を逐次的に計算すれば簡単に求められる。

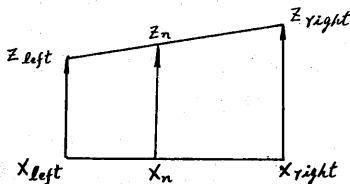


図9. 平行投影の場合のZ座標値の補間

3.6 透明・半透明面素を含む場合

透明・半透明な面素を含む場合、Newellらによって考案された簡単な透明計算法を利用して陰影計算を行う。この際、上述の簡単な隠れ面処理では実行できない、走査線中の画素ごとに陰影値を決定することになる。

走査線 $Y=C$ 中のある画素を X とする、三角形面素の中で、 $Y=C$ の走査線に対するセグメント (X_{left}, X_{right}) が $X_{left} \leq X \leq X_{right}$ の関係を満たすものが画素 X と交わる。

画素 X と交わるすべての三角形面素の番号とZ座標値を記憶しておき、これをZ座標値の大きい順(視点に近い順)にソートする、その結果に対し、大きい順に調べ、最初の透明度 $t=1$ (不透明)な面の陰影値を記憶する(この陰影計算は3.4で述べた方法による)。次に、順次Z座標値の大きいものをとり出し、式(25)により、陰影値を更新する。これを最も大きいZ座標値のものまで行う。

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = (1-t) \begin{pmatrix} R_0 \\ G_0 \\ B_0 \end{pmatrix} + t \begin{pmatrix} R_1 \\ G_1 \\ B_1 \end{pmatrix} \quad (25)$$

但し、

$\begin{pmatrix} R_0 \\ G_0 \\ B_0 \end{pmatrix}$: 既に書かれている陰影値

$\begin{pmatrix} R_1 \\ G_1 \\ B_1 \end{pmatrix}$: 現在の面素の陰影値

t : 現在の面素の透明度 ($0 \leq t \leq 1$)

$\begin{pmatrix} R \\ G \\ B \end{pmatrix}$: 更新された陰影値

4. 応用

4.1 多面体の表示

一般に、多面体の各面は、三角形面素の集合として分解されるので、1個または、複数個の多面体は明らかに本システムにより表示可能である。

図10~16に本システムにより表示された多面体の例を示す。

ここで、注意すべきことは、平行投影及び平行光源を用いた場合、多面体上の各面において、陰影値(輝度値)は一定となるため、この条件を用いると物体の材質感が出にくい点である、材質感表現には、特に点光源を用いた方が良くと考えられる。

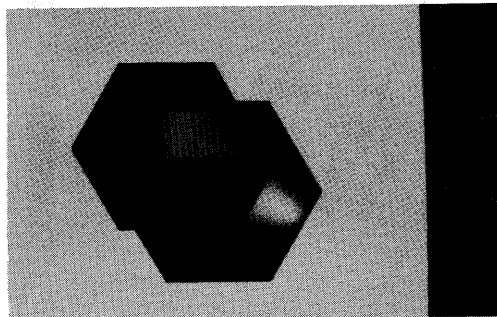


図12. 立方8面体の金属材料質感の表示(式(22)に f_3)

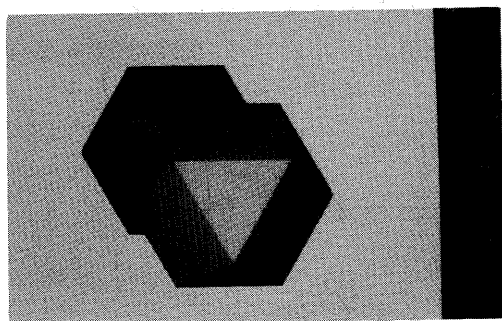


図10. 立方8面体の表示例(式(21)に f_3)

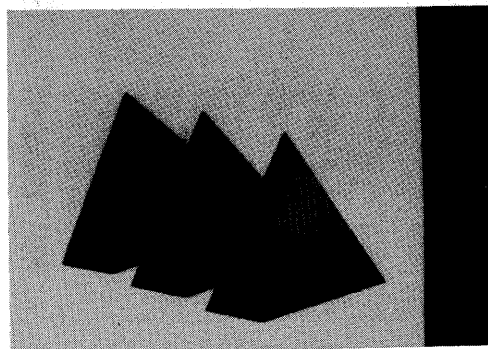


図13. 4面体の金属材料質感の表示(式(22)に f_3)

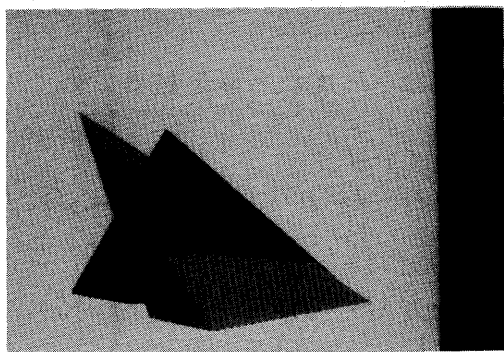


図11. 4面体の合成表示例(式(21)に f_3)

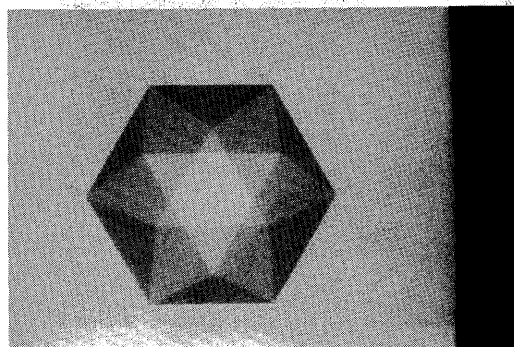


図14. 立方8面体の透明表示(式(25)に f_3)

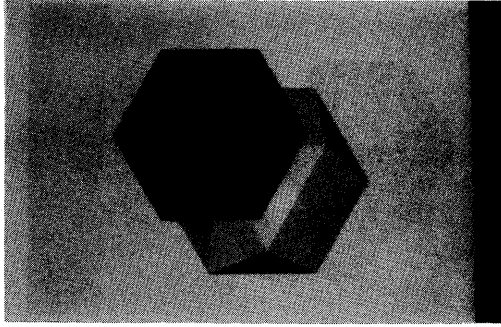


図15. 透明及び不透明な立方体面体の合成表示例

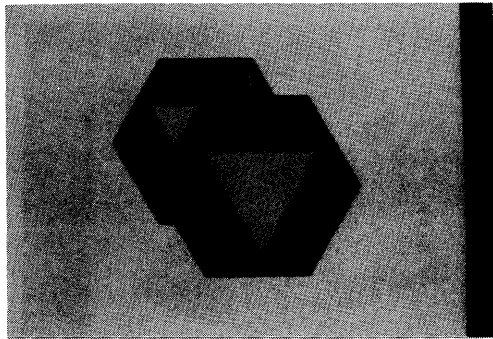


図16. 透明及び不透明な立方体面体の合成表示例

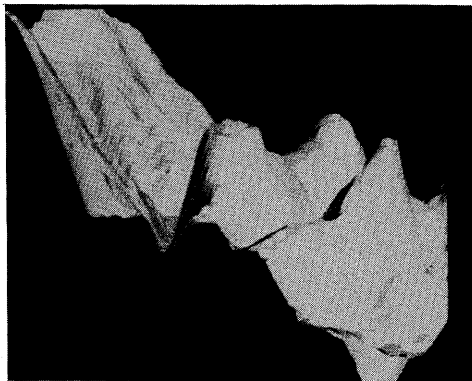


図17. 地形の陰影表示
(51×100格子点にて区切る)

4.2 地形図2変数関数データの表示

地形図を含む2変数1価関数のデータで線画表示することはよく行われている。ここでは、このデータを面画表示する方法について述べる。

(1) 入力データの形式

入力データは、2変数1価関数 $Z = f(x, y)$ の形で与えられ、各格子点 (x_i, y_j) ($i=1, \dots, M, j=1, \dots, N$) の値(標高値) Z_{x_i, y_j} が与えられる。

(2) 面画表示

上述の2変数関数で与えられた物体データから隣接する3点を順次に取り出し、三角形面素を構成する。

各三角形面素は空間座標で次のように表される2種のものが存在する。

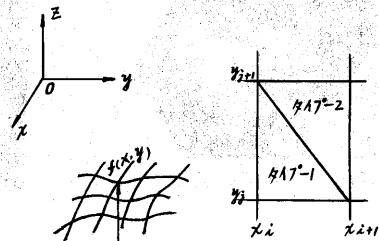


図18. 三角形面素の構成

図19. 二変数一価関数データ

(タイプ-1)

$((x_i, y_j, z_{x_i, y_j}), (x_{i+1}, y_j, z_{x_{i+1}, y_j}), (x_i, y_{j+1}, z_{x_i, y_{j+1}}))$

(タイプ-2)

$((x_{i+1}, y_j, z_{x_{i+1}, y_j}), (x_i, y_{j+1}, z_{x_i, y_{j+1}}), (x_{i+1}, y_{j+1}, z_{x_{i+1}, y_{j+1}}))$

各隣接格子点に対し、上記2種類の面素が構成されるので、これらに対する表示処理を行う。

ここで、隠れ面処理に関しては、各面素は視点から遠い行の格子点に関する

るものから順に投影(セグメント生成)していき、順次、近い方の面素を重ね書きしていけばよく、Z-バッファ法のような特別な隠れ面処理は必要としない。

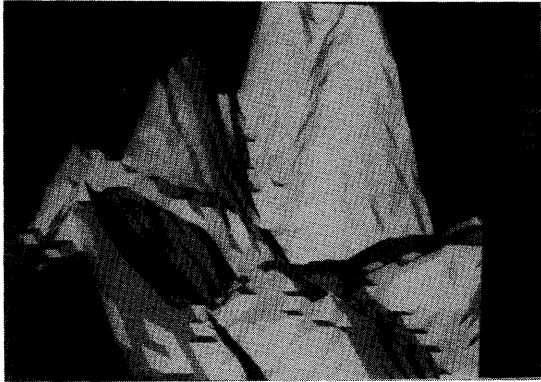


図20・地形の陰影表示
(51×100格子点に区切る)



図21・地形の陰影表示
(51×100格子点に区切る)

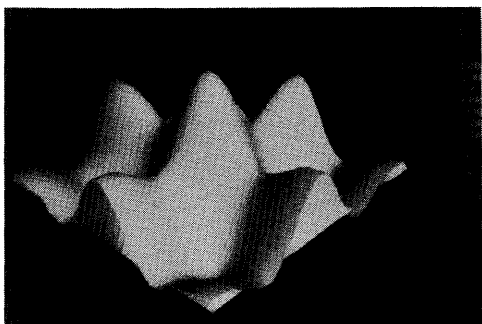


図22・関数の陰影表示
(関数 $Z = (\cos(x) \cdot e^{-tx} + \sin(y) \cdot e^{-ty})R$, $R=10$, $t=1/100$,
60×60格子点1に区切る)

4. おすび

三角形面素の集合を種々の条件のもとで生成(表示)する汎用的システムを開発した。本システムにおいて問題となったことがいくつかある。一つは、多面体の場合、平面で構成されているため表面の陰影値の変化が少なく、物体の材質感が出にくい点である。この点は、点光源を用いればある程度解決されるが、曲面に比べて材質感が出にくいことには変わらない。次に、透明面素を含む場合、Newellの簡易透明表示法をそのまま利用したが、本文中で述べたように、走査線中の画素ごとにZ座標値によるソートを行う必要があり、非常に計算時間がかかる。また、透明感も十分でない。これらの点については今後の問題として検討したい。

現在、本システムにスムーズシェーディング機能と影の表示機能の付加を検討中である。これらを組み込んだシステムが完成するとかなり広汎な目的での利用が可能であろう。

謝辞

日頃有益な御助言を頂く、名古屋大学福村教授・鳥脇教授に感謝する。また、本システム作成に対して有益な御指摘を頂いた日本電気株式会社の萬淳一氏に感謝する。

文献

- (1) N. Badler and R. Bajcsy: "Three-dimensional representations for computer graphics and computer vision", *Computer Graphics*, 12, p.p. 153~160 (1978)
- (2) 岡村、原因、中前、西田: "三角形近似による自由曲面のカラー陰影表示法", *電子通信学会信学技報 IE 82-92* (1982)
- (3) 木村滋: "3次元物体のカラーハーフトーン表示の基本要素としての三角形タイル表示" *情報処理学会*

第24回全国大会, 3J-6 PP.1129-1130
(1982)

- (4) J. Blinn: "Clipping using homogeneous Coordinates", *Computer Graphics*, 12, 3
PP. 281~287 (1978)
- (5) F. C. Crow: "Shaded computer graphics in the entertainment industry", *Computer*,
11 PP. 11~22 (1978)
- (6) 横井, 伊藤, 鶴岡, 三宅: "カラー
3次元物体の反射、透過モデルに
関する基礎的検討", 情報処理学会
グラフィクスとCAD研資, 7-1 (1982)
- (7) M. E. Newell, R. G. Newell and
T. L. Sancha: "A solution to the
hidden line problem", *Proc. ACM
National Meeting*, (1972)