

〈論文〉

三次元物体の陰影表示の一方法*

西田友是** 中前栄八郎***

Abstract

The authors describe the algorithms for the half toned presentation of objects projected by a parallel light, and show the usefulness of the algorithms by some examples.

The input data are represented by a set of the convex polyhedra constructing a object. The algorithms are as follows. The light source is assumed the first observation point. The back faces to the light source are the shading planes. Then whether the contour lines on the projection plane intersect or not is decided. The intersecting area are shadows on to the faces of a convex polyhedron hidden by another.

The observers observation point is treated the second observation point. Then the relation of intersection of each polyhedron on the projection plane is decided. The last step is scanning to make the figure on the projection plane like the scanning of television.

1. まえがき

三次元図形を二次元平面上に写實的に表示するためには、物体の隠れた部分を消去して表示する問題(hidden line elimination)¹⁾⁻⁸⁾ および陰影表示問題(half toned presentation)⁹⁾⁻¹⁵⁾を解決する必要がある、これらに関して多くの論文が発表されてきた。しかし、一般にこれらの問題は相当複雑な処理とかなりの処理時間を必要とし、入力方法も面倒であり人間-機械系の立場からみた場合、必ずしもじゅうぶん解決されたとはいえない。筆者らは多面体の隠線消去の一手法として、多面体を適当に分割して凸多面体の集合として入力することにより入力データの簡易化を計り、同時に処理時間の高速化を可能にするアルゴリズムを開発し発表した⁸⁾。本論文では、この手法を拡張して、三次元物体の陰影表示を行なうアルゴリズムについて述べる。

先に述べたように陰影表示に関して多くの論文が既に発表されているが、それらの主要な特徴について述べると次のごとくである。Romney, Watkins¹¹⁾等は

面を三角形に分割し、走査線と三角形の境界の交差を調べる方法を取り、図の濃さについては物体の方向と距離の関数で表示した。また Warnock¹²⁾は面を最小区分まで分割し“key square”の集合として取り扱い走査線ごとに計算を行ない、各走査線に対して隠れた部分を消去し、各面の可視部分の濃度を距離と方向から求めた。なお物体の反射率をも考慮している。以上はいずれも視点と光源は同一であり、したがって物体による影の投影は考慮していない。また Appel¹⁰⁾はdigital plotterを使用し、影の境界は影を作る面の境に沿って順に point by point で影になる面に投影して計算を行なった。また複数個の光源を考えることを可能にし、面の濃度はその面の方向の関数のみによって求め、輪郭を描くことによって図形の混乱を避けた。なお三色によるカラー表現を実現した。Bouknight, Kelly¹³⁾は、Warnockの方法を拡張し、光源の移動を可能にし、投影された影の処理を解決した。また投影の計算を少なくするため、多面体の交差の判定の簡易化を行ない、各交点の計算は最後の段階で行なうようにしている。また Goldstein, Nagel¹⁴⁾は曲面を表現するため三次元幾何を表現する比較的簡単なモデル(例えば球、円柱、円錐、だ円体等)を考え、光の反射成分を求める方法を開発した。さらに Gouraud¹⁵⁾は自由曲面を細分して平面で近似した入力に対してス

* An Algorithm for Half Toned Representation of Three Dimensional Objects, by Tomoyuki Nishita (Toyo Kogyo Co., Ltd.) and Eiichiro Nakamae (Faculty of Engineering, Hiroshima University)

** 東洋工業株式会社

*** 広島大学工学部

ムージングのアルゴリズムを開発し、高速化のためのハード化を行ない、曲面の表示に成功した。

本論文では平行光線で写し出された物体の陰影表示について述べ適用例をあげてその有用性を示す。本アルゴリズムの特徴は次のようである。(1) 任意の形状の多面体を凸多面体に分割し、使用頻度の多い形状(平行六面体, 円柱等)は標準化し、必要なパラメータのみを入力することで従来の方法に比べデータの作成が早く、かつデータ・エラーも減少できる。(2) 検討の対象となる凸多面体の大きさ、位置の変更が簡単でかつ同形のを複数個自動作成できるのでアレンジが容易である。(3) 凸多面体の輪郭線の性質を利用することにより陰影処理のアルゴリズムが簡単になり、処理時間が短縮される。(4) 平面上の図柄や任意の線分を含めた隠線消去も同時に行なうことで複雑な図形の処理が可能で、広範な実用面に応用できる。

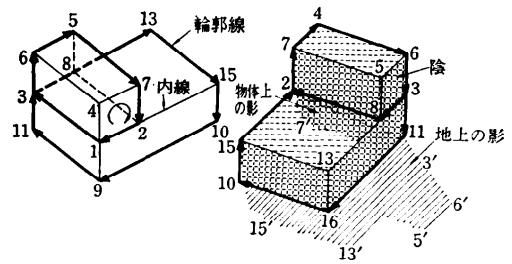
2. アルゴリズム

まず本論文において用いる用語を次のように定義する。すなわち観察者の目の位置を視点、観察者が注視している点を注視点とし、注視点を含み視線に垂直な平面を透視面とする。また面の法線ベクトルは物体の外部を向くものとする。ここで視点または光源に向いている面をそれぞれに対する表の面 (front face), その他の面を裏の面 (back face) と定義し、表の面と表の面の稜線を内線, 表と裏の面との稜線を輪郭線分, そのつながりを輪郭線とする。ここで輪郭線は透視面上において凸多角形であり、右回りに定義することができる。

アルゴリズムは次の仮定のもとで行なわれるものとする。すなわち光は太陽光線のような平行光線とし、物体の反射光の照度への影響は無視するものとする。

また出力装置(ラインプリンターまたはXYプロッター)の濃淡表示能力の制限を考慮して物体の反射率は一律と仮定し、また視点から物体までの遠近によるコントラストは考慮しない。なお曲面体および曲面は凸多面体と凸平面の集合で近似する。

陰影表示のアルゴリズムは次のごとくである。まず光源を第一の視点(ただし光源は無限遠点に存在するものとする)と仮定し影(shadow)を求める。すなわち凸多面体の各頂点を光源から見た透視面上へ座標変換し、光源に対する面の向きを求めると、裏の面が陰(shade)の面である。また各面の向きを利用して各凸多面体の内線と輪郭線を求め、透視面上で輪郭線が交



(a) 光源からの投影 (b) 視点からの投影

図 1 投影図

Fig. 1 Perspective drawings.

差するか否かを判定する。交差した部分は光源に対してより近い凸多面体による遠い凸多面上への影であるから、後者の各面上での影を求める(図1(a)参照)。またこれらの輪郭線が地上の影の輪郭線を構成するから、それらの頂点を座標変換して地上の影を求める。この際、輪郭線が他の輪郭線を囲む場合は、囲む方の輪郭線のみで地上の影を求めることができる。

次に観察者の視点を第2の視点と考え、透視面上での濃淡を求める。まず各頂点を透視面上へ座標変換し、面の向きを求め、内線および輪郭線を求める。次に透視面上での凸多面体の交差関係を求め、交差する場合は前後関係を求め記憶する。最後に透視面上でテレビの走査と同様に凸多面体および影を走査し、その走査線上の各点の濃淡を求め表示する。以下にその詳細について述べる。

2.1 座標変換

入力データの三次元座標系を (X_0, Y_0, Z_0) とし、注視点 P_r を原点とするように平行移動した後の座標系を (X, Y, Z) とする。視点 P_v は P_r からの距離 R , XZ 平面との角 θ および XY 平面との角 φ で与える。透視面を $X'Y'$ 平面とし視線を Z' 軸とする座標系からみると、頂点 $P(X, Y, Z)$ は

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} -\sin\theta & \cos\theta & 0 \\ -\cos\theta \cdot \sin\varphi & -\sin\theta \cdot \sin\varphi & \cos\varphi \\ \cos\theta \cdot \cos\varphi & \sin\theta \cdot \cos\varphi & \sin\varphi \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (1)$$

となり透視面上へ投影されたその頂点を $P'(x, y)$ とすると

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{R}{R-Z'} \cdot \begin{bmatrix} X' \\ Y' \end{bmatrix} \quad (2)$$

ただし、 $R > Z'$ の範囲で考え、光源から見た場合は $R = \infty$ である(図2参照)。また $Z_0 = Z$ の平面(通常は $Z_0 = 0$ とする)を地面と仮定し、光の方向を θ' ,

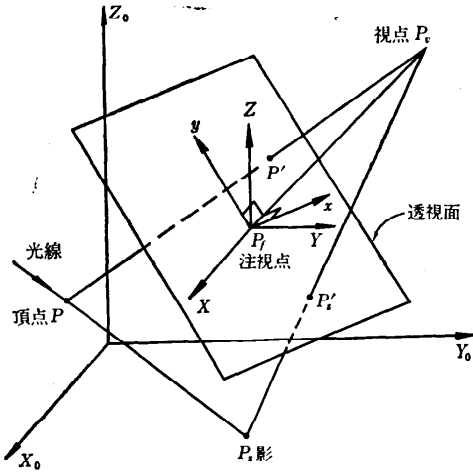


図 2 座標系
Fig. 2 Co-ordinates system.

φ' で与えると頂点 P の地上の影 $P_i(X_i, Y_i, Z_i)$ は

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\cos\theta' \cdot \cot\varphi' \\ 0 & 1 & -\sin\theta' \cdot \cot\varphi' \\ 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z_0 - Z_i \end{pmatrix} \quad (3)$$

で求めることができる。次に (1), (2) 式を用いて透視面上での影の点 P_i' を求める必要があるが、この場合凸多面体の性質から、その輪郭線は 1 個でかつ凸多角形であるから、光源から見て輪郭線を構成する頂点のみ計算しておけばよい。

2.2 稜線の分類

面は外部から見て右回りにとっており、かつ凸多角形であるから、面 S が $\{P_1, P_2, P_3, \dots\}$ で構成されるなら、面の法線ベクトル N は $(P_1 - P_2) \times (P_3 - P_2)$ であるから、面が視点 $P_v(X_v, Y_v, Z_v)$ を向いているか否かは次式の符号で判定できる。

$$\begin{aligned} \Delta_{P_v} &= N \cdot (P_v - P_1) = - \begin{vmatrix} X_v & Y_v & Z_v & 1 \\ X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ X_3 & Y_3 & Z_3 & 1 \end{vmatrix} \\ &= \Delta_A X_v + \Delta_B Y_v + \Delta_C Z_v + \Delta \end{aligned} \quad (4)$$

ここで $(\Delta_A, \Delta_B, \Delta_C)$ は面の法線ベクトルの要素である。ただし光源から見たときは、光線ベクトル $L(X_L, Y_L, Z_L)$ とすると光源を無限遠点に仮定しているから、

$$\Delta_i = N \cdot L = \Delta_A \cdot X_i + \Delta_B \cdot Y_i + \Delta_C \cdot Z_i \quad (5)$$

を用いて判定できる。しかし光源から見た場合も、視点から見た場合もともに面の構成頂点 P_1, P_2, P_3 が透視面上で右回りなら表の面であるから、透視面上での座標から次式の符号で判定可能である。

$$D = - \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \quad \begin{cases} > 0 \text{ 表の面 (front face)} \\ \leq 0 \text{ 裏の面 (back face)} \end{cases} \quad (6)$$

この面の表裏の関係から各凸多面体の内線、輪郭線分を求め、輪郭線分の方角を利用して輪郭線 (頂点番号のつながり) を求める。

2.3 二つの凸多面体の前後関係

凸多面体が重なって見えるか否か、また影が生じるか否かは、視点あるいは光源から見た透視面上で輪郭線同志が交差しているか否かにより判定する。まず輪郭線の構成頂点の x, y 座標の最大値および最小値により独立な凸多面体を除いたのち、文献 8) の方法を適用して輪郭線同志の交差判定をする (付録参照)。

輪郭線が交差した場合、どちらの凸多面体が手前か判定する必要があるが、これは互の輪郭線の透視面上での交点の一つを用いて、この交点に対応する凸多面体上の点の視点あるいは光源に対する位置関係で判定することができる。

a) 視点から見た場合: 透視面上で、線分 $\{P_i, P_{i+1}\}$ と $\{P_j, P_{j+1}\}$ の交点 P_i' とし、それに対応する凸多面体 V_i, V_j 上の点を Q_i, Q_j とする (図 3 参照)。交点 $P_i'(x, y)$ を (1) 式の関係を用いて三次元上の点 $P_i(X, Y, Z)$ として求め、 $P_i, P_{i+1}, P_j, P_{j+1}, P_v, P_i'$ を任意な面 (たとえば $Z=0$ 平面) に投影し、投影面での線分 $\{P_i', P_{i+1}'\}$ に対する (付 1) 式を $f_{i,j}'$ とし、 $\overline{P_v P_i}$ と $\overline{P_v Q_i}$ の比 δ_i とすると、

$$\delta_i = \overline{P_v Q_i} / \overline{P_v P_i} = f_{i,j}' / (f_{i,i}' - f_{i,i}') \quad (7)$$

となる。ここで分母が 0 のときは他の面へ投影する。

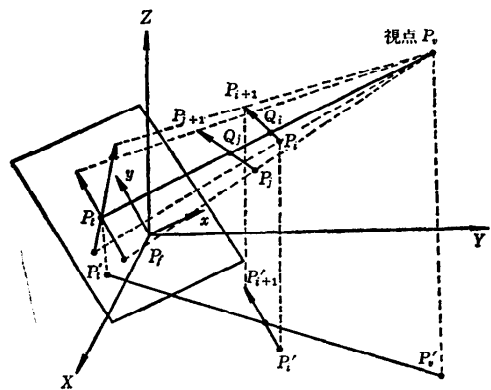


図 3 物体の前後判定
Fig. 3 Decision of distance from object to visual point.

また $\overline{P_i P_i}$ と $\overline{P_i Q_j}$ の比 δ_j を同様に求めると次式の関係がある。

$$\delta = \delta_i - \delta_j \quad \begin{cases} > 0 & V_j \text{ が手前} \\ < 0 & V_i \text{ が手前} \end{cases} \quad (8)$$

ここで $\delta = 0$ の場合は、その交点を含む面と線分の端点との関係から (4) 式を用いて前後関係を求める。

b) 光源から見た場合：光は平行光線と仮定しているから、頂点間の交点までの比は三次元上でも保持されるから、透視面上の座標に (付1), (付3) 式を適用し、

$$t_j = \overline{P_j P_i} / \overline{P_j P_{j+1}} = -f_{i,j} / \Delta_b \quad (9)$$

とすると、三次元上での交点 Q_j は次式から求まる。

$$Q_j = P_j(1-t_j) + P_{j+1}t_j \quad (10)$$

凸多面体 V_i に属す面で、かつ交点を含む面を S とすると、(4) 式の Δ_{Q_j} の符号より Q_j が S のどちら側にあるか判る。すなわち正なら V_j が手前であり、負なら V_i が手前である。

透視面上で輪郭線が他の輪郭線に囲まれる場合には光源から見たときも、視点から見たときも次の方法で前後判定を行なうことができる。囲まれる凸多面体 V_B の任意な頂点 P と透視面上で P を囲む凸多面体 V_A の表の面 S との関係は (4) 式の Δ_P を用いて判定する。光から見た場合には地上への影は V_A の輪郭線で決まり、 V_B が隠されるなら V_B はすべて影である。

2.4 面上の影の領域と照度の計算

光源から見た透視面上において輪郭線が交差すると、隠される側の凸多面体上の面に影を生じる。いま隠す側の凸多面体を V_i 、隠される側を V_j とすると、 V_i の輪郭線が V_j の面上への影を構成する。これを求めるには次の方法で行なう。 V_i, V_j の輪郭線の交点において、 V_j の領域内へ入る V_i の輪郭線分から、 V_j を出る輪郭線分へと進んで各面上の影の位置を求める。ここで交点は必ず偶数個であり、輪郭線分の V_j に対する出入りは (付3) 式の Δ_b で判る。

図4によって説明すると、まず入る輪郭線分 $[P_b, P_{b+1}]$ とその交点を含む面 F_b に属する内線との交差判定をする。交点がない場合には、交点が存在するまで V_i の輪郭線分を一つずつ進め、交差した場合には V_j のその内線を含む他のもう一つの面 F_j に進み、その面に属す内線と交差判定をし、次々とこれを繰返し、 V_i の輪郭線が V_j を出るまで、すなわち F_i まで続行する。

内線と輪郭線の交差判定は、判定中の面の向きに方向づけられた内線 $[P_k, P_{k+1}]$ と輪郭線分 $[P_i, P_{i+1}]$

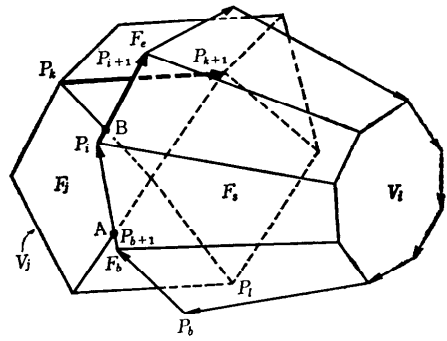


図4 影の領域の判定

Fig. 4 Search for shadow on plane.

が (付1) 式を用いた次式を満足するなら交差する。

$$f_{i,k} \leq 0 \cap f_{i,k+1} \geq 0 \cap f_{k,i+1} < 0 \quad (11)$$

面上の影、すなわちこれらの輪郭線の頂点、または内線との交点の記憶は、各面に対して次式を満たす t_1, t_2 によりその後の処理を簡便にすることができる。すなわち V_j の影の面を $F[P_1, P_2, P_3, \dots]$ とすると、

$$P_i = (P_1 - P_2) \cdot t_1 + (P_3 - P_2) \cdot t_2 + P_2 \quad (12)$$

したがって

$$\begin{aligned} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} &= \begin{bmatrix} x_1 - x_2 & x_3 - x_2 \\ y_1 - y_2 & y_2 - y_2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} x_i - x_2 \\ y_i - y_2 \end{bmatrix} \\ &= \frac{1}{f_{1,3}} \begin{bmatrix} f_{2,i} \\ -f_{1,i} \end{bmatrix}. \end{aligned} \quad (13)$$

視点から見たときの面上の影の座標は、(12) 式により三次元上の点 P_i を求め、(1), (2) 式から透視面上での影 P_i' を求める。なお図4に示すように、視点から見た透視面上において、 V_i の輪郭線が面 F_j から出る交点 B から面 F_j に入る交点 A までの右回りにみた F_j の頂点も、 F_j 上の影を構成する頂点であり、この影は右回りで凸多角形である。

上記の方法では V_i の輪郭線に囲まれて面全体が影となる面 (例えば図4の F_i) が求まっていない。この場合輪郭線と交差していない面の任意な頂点 P_i が V_i の輪郭線に囲まれる場合は影の面となる。すなわち V_j の領域内に存在する V_i のすべての輪郭線に対して、(付1) 式の $f_{i,j}$ が正なら、輪郭線は凸の性質より P_i は V_i に囲まれ、 F_s は影の面となる。

つぎに視点から見て表の面は照度を求める必要がある。いま単位の光線ベクトルを $L(X_i, Y_i, Z_i)$ とすると、各面の照度は、光 A と面の法線ベクトル N の余弦に比例するから

$$\begin{aligned} \cos \theta &= (L \cdot N) / (|L| \cdot |N|) \\ &= (\Delta_A X_i + \Delta_B Y_i + \Delta_C Z_i) \end{aligned}$$

$$\begin{aligned} & \div (\Delta A^2 + \Delta B^2 + \Delta C^2)^{1/2} \\ & = D / (\Delta A^2 + \Delta B^2 + \Delta C^2)^{1/2} \end{aligned} \quad (14)$$

ここで D は(6)式の結果を用いる。したがって面の濃さを H とすると、

$$H = H_{\max}(1 - k \cos \theta) \quad (15)$$

ここで H_{\max} は最大濃さ、 k は面の反射係数とする。また陰の面では当然 $H = H_{\max}$ である。

2.5 Scanning Algorithm

各点の濃さを求めるには、視点から見た透視面上で上から下へ scan を行なう。いま scan をする水平線を scan line とすると、その scan line 上にある凸多面体と地上の影を求め、さらにその scan line と交差している線分を求め、凸多面体ではさらに scan line 上の面および面上の影を求める。

個々の輪郭線、地上の影および面上の影はすべて凸多角形であるから scan line との交差判定は次の手順で容易に求まる。凸多角形 S の頂点の中で y 座標の最大値を与える点を P_{\max} 、最小値を与える点を P_{\min} とし、その y 座標をそれぞれ y_{\max} 、 y_{\min} とする。いま scan line を $y = y_s$ とすると (図5(a)参照)、凸多角形 S は $y_{\min} \leq y_s \leq y_{\max}$ のとき scan line 上に存在し、scan line と S との交点は2個で (P_{\max} 、 P_{\min} では2交点とする)、それらは P_{\max} から P_{\min} に右回りの線分と、左回りの線分上にある。したがって S のどの線分と交差するかは左右独立に求めることができるから、左右それぞれ下向きに方向づけした頂点のつながりとして記憶しておく。また内線も同様に下向きにしておく。 $y_{i+1} < y_s \leq y_i$ を満たす線分 (P_i 、 P_{i+1}) が scan line と交差するが、scan line は上から下へと移動するから $y_{i+1} < y_s$ なら scan line と交差する。 $y_{i+1} \geq y_s$ ならば $i = i+1$ とし次の線分に進み、同様な手順で P_{\min} まで続行する。また交点をそのつど求めるには乗除算に時間を要するので、 $y_i < y_s$ から $y_i \geq y_s$ と

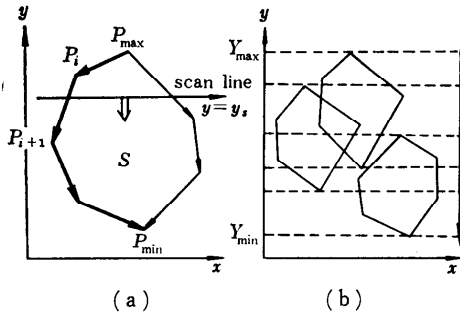


図5 図形の走査

Fig. 5 Scanning motion.

なったときの x を求め、次からは交差するたびに $\Delta x / \Delta y = (x_i - x_{i+1}) / (y_i - y_{i+1})$ を引いて交点 x の値を求める。

また走査中の scan line 上にある凸多角形の y_{\min} か、他の凸角形の y_{\max} が現われるまでは scan line 上にある凸多角形は変化しないから、どの凸多角形が scan line 上にあるかの判定は $2n_0 - 1$ 回以内でよく、scan line は y_{\max} の最大 Y_{\max} から y_{\min} の最小 Y_{\min} まで動かせばよい (図5(b)参照)。

なお凸多面体のどの面が scan line 上にあるかは次の手順で求める。図6に示すように scan line が凸多面体へ入る交点を x_e 、出る交点を x_i とし、 x_e, x_i を含む面をそれぞれ F_e, F_i とすると、 $F_e \neq F_i$ のときは F_e に含まれる内線と scan line との交差を求め、この内線から次の面 F_i が判るから、その面を含む内線と scan line との交差を判定し、面 F_i に到達するまで続行して scan line 上の面を求めることができる。面の照度はすでに述べた方法で求められているから各点の濃さが求まる。なおその面上に影がある場合には、さらに影の輪郭との交差を求め、影の領域を求める必要がある。

scan line 上で多くの凸多面体や影が重なった際、どの凸多面体および影のどの部分が見えるか判定する必要がある。この手順として視点から遠いものから順に記憶していく方法を用いた。このようにすると最後には可視部分のみが記憶されることになる。すなわち地上の影をすべて記憶した後、遠い凸多面体から順に面の濃さ、面上の影を求め記憶する。図7(a)にその例を示す。ここで高さの高いものほど手前のものとする。ところで凸多面体の前後関係が図7(b)のように順序をつけにくいものが存在するが、scan line 上で

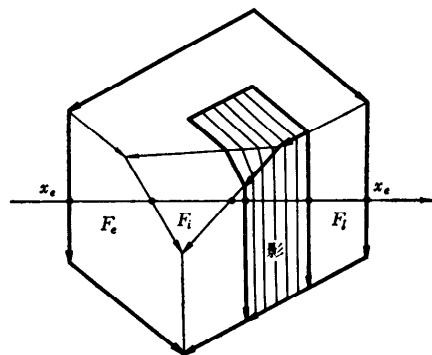


図6 Scan line 上の濃度

Fig. 6 Density on scan line.

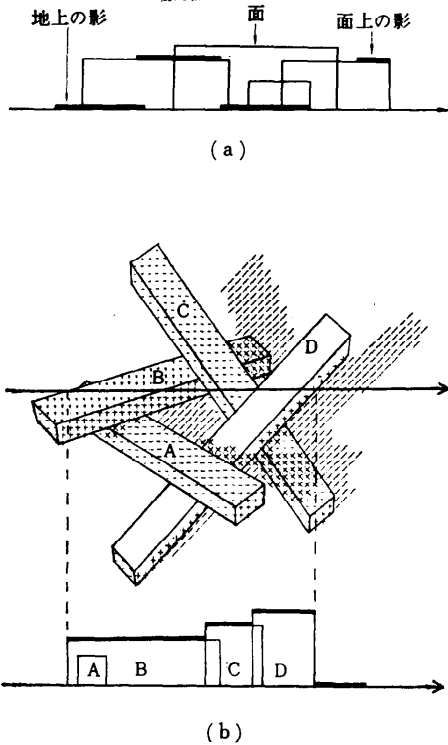


図 7 Scan line 上での前後関係
Fig. 7 Relation of objects on scan line.

重なっているものだけに順序づけをすると、凸多面体の性質から scan line 上では一意的に決まる。つまり A, B, C, D の順に記憶していくと図 7 (b) のように太線の部分が残るからそれを表示すればよい。

以上の計算手順の概略を図 8 に示す。ここで n_0 は凸多面体数で、 $k=1$ のときは光源から見た場合であり、 $k=2$ のときは視点から見た場合である。

3. 適用例

前記のアルゴリズムによる適用例を図 9 に示す。図中 (a), (b) は本プログラムで用意している標準型の凸多面体のみで構成した例であり、(a)~(c) はラインプリンターで表示した。紙面の制限から格子の数は 135×180 とした。なお活字の縦横比が異なるため出力の際これを補正している。また濃淡は最高 7 回重ね印字、20 段階とした。(d)~(f) は同一照度の面の境を示すためと面上の図柄および空間中の線分も処理するために XY プロッターを用いて表示した。濃淡

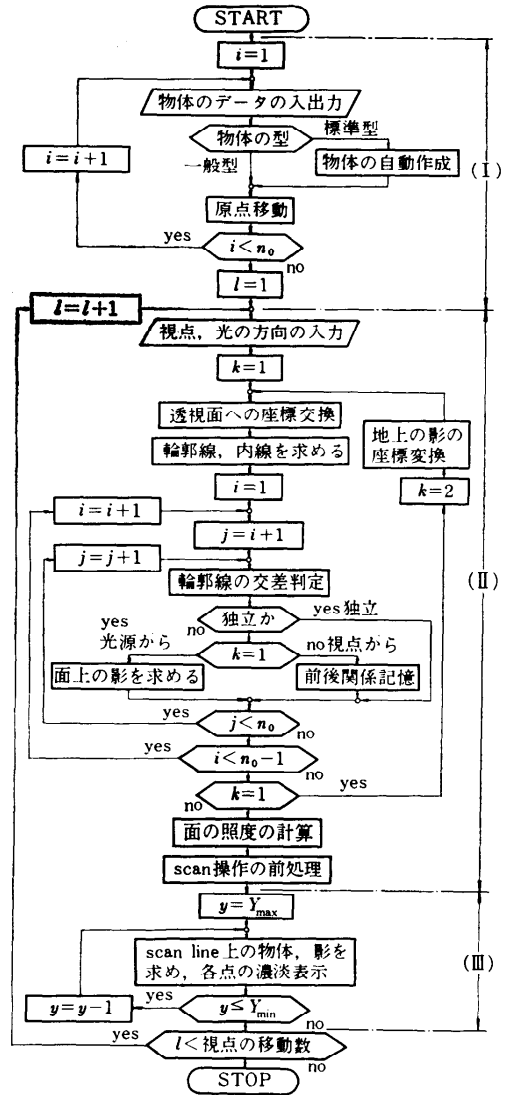


図 8 フローチャート
Fig. 8 Flow-chart.

のレベルは 10 段階とした。(e), (f) は実際の病院の鳥瞰図と仰視図である。なおこの場合の隠線消去の方法は文献 8) を用いた。本例の計算には FACOM 230-60 とプロッターを使用し、プログラムは FORTRAN を使用した。各例の処理時間を表 1 に示す。なお同表の計算区分は図 8 による。

4. むすび

筆者らの使用した出力装置はラインプリンター、

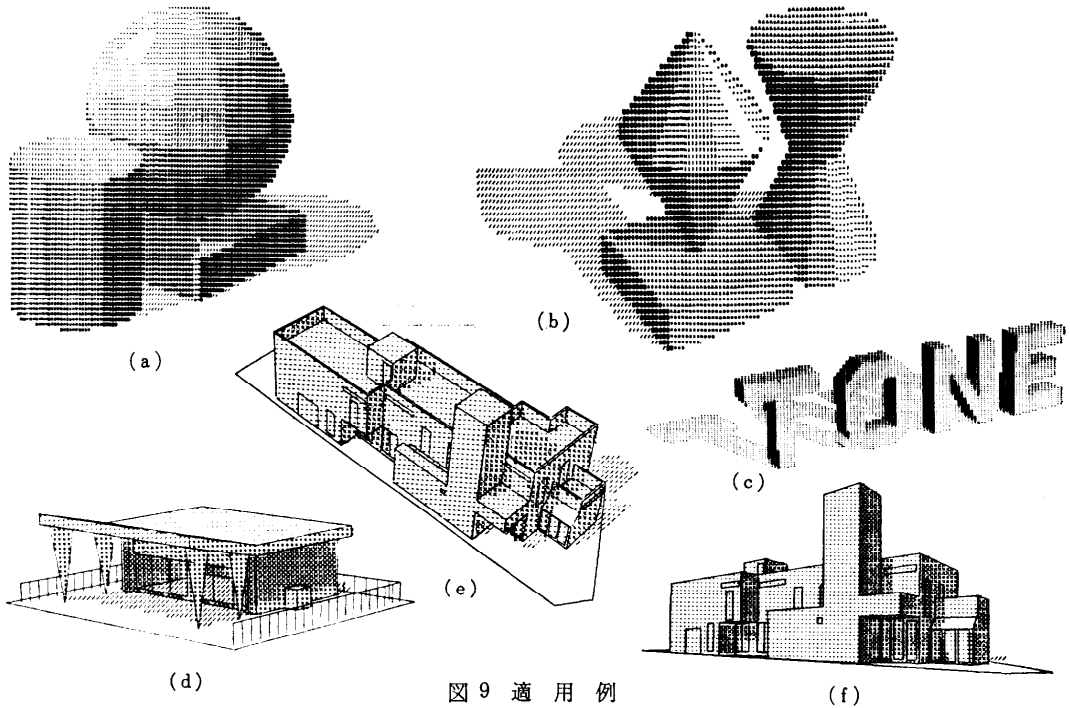


図9 適用例
Fig. 9 Example.

表1 計算時間
Table 1 Computation time.

図	凸多面体	頂点	面	CPU-TIME (秒)			隠線消去
				I	II	III*	
(a)	3	122	118	0.19	0.48	0.46(7.41)	
(b)	5	120	70	0.30	0.80	0.33(7.72)	
(c)	13	104	78	0.62	0.63	0.31(7.59)	
(d)	15	76	50	0.65	0.80	0.44(2.55)	0.78
(e)	27	218	163	1.14	2.80	0.99(6.15)	1.75
(f)	27	218	163	1.14	2.69	0.59(5.40)	1.52

* () 内はプリンター出力時の印字命令時間

XYプロッターであるため、表現能力はかなり制限される。したがって視点から物体までの距離による対照の強弱、物体の反射率、周囲の明るさおよび曲面のスムージングを考慮しなかった。また図形としての見易さを得るため影は陰と同じ濃さにすべきであるが、地上の影はやや淡い値とした。したがって濃淡表示能力の良い装置の使用が可能であれば、上記の諸問題を考慮に入れたより良い画質で表示できると推察する。

本アルゴリズムではほとんどの計算式が(付1)や(4)式のように簡単で、かつ一度求めた値を重複して利用できるため高速化が期待できる。また凸多面体の性質を利用して前処理することにより、図形の scann-

ing が早くなり、出力図形の格子数にそれほど影響されない。

謝辞

本論文の作成にあたり、計算機の使用に種々便宜を計っていただいた広島大学計算センター、名古屋大学および九州大学大型計算センターの各位、および貴重な資料の提供をいただいた中電技術コンサルタント株式会社の日高氏に感謝いたします。

〈付録〉 二つの輪郭線の交差判定

線分 (P_i, P_{i+1}) の右半平面上に頂点 P_j が存在する場合、(付1)式の $f_{i,j}$ が正であることを利用する。すなわち輪郭線 S_i の線分 $[P_i, P_{i+1}]$ と輪郭線 S_j の頂点 P_j の関係を示す $f_{i,j}$ を要素とする行列と、 S_j の線分に対する S_i の頂点の関係を示す $f_{j,i}$ を要素とする行列から、簡単に輪郭線が交差するか、独立か、または囲まれるか判定できる。

$$f_{i,j} = \begin{vmatrix} x_j & y_j & 1 \\ x_i & y_i & 1 \\ x_{i+1} & y_{i+1} & 1 \end{vmatrix} = a_i x_j + b_i y_j + c_i \quad (\text{付1})$$

なお (P_i, P_{i+1}) , (P_j, P_{j+1}) が交差した場合交点は次

式で求まる.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{\Delta_b} \begin{bmatrix} -b_j & b_i \\ a_j & a_i \end{bmatrix} \cdot \begin{bmatrix} c_i \\ c_j \end{bmatrix} \quad (\text{付2})$$

ただし

$$\Delta_b = \begin{vmatrix} x_{i+1} - x_i & y_{i+1} - y_i \\ x_{j+1} - x_j & y_{j+1} - y_j \end{vmatrix} = f_{i,j+1} - f_{i,j} \quad (\text{付3})$$

参考文献

- 1) I. E. Sutherland: Computer Graphics-Ten Unsolved Problems, *Datamation*, 12, 5, p. 22 (May. 1966).
- 2) P. P. Loutrel: A solution to the hidden-line problem for computer drawn polyhedra, *IEEE Trans. Computers*, C-19 (March 1970).
- 3) 大森: 多面体の隠線消去問題, *情報処理*, 12, 9, p. 534 (1971).
- 4) 高橋栄: Hidden Line Algorithm とデータ構造, *情報処理学会, グラフィクス・ディスプレイ報告集*, p. 51 (1970).
- 5) 中前・西田: 三次元物体の Hidden Line の処理の一手法, *電全大*, p. 1919 (1971).
- 6) F. J. Mcgrath: A method for eliminating hidden lines with polyhedra, *Simulation*, p. 37 (Jan. 1971).
- 7) C. B. Jones: A new approach to the hidden line problem, *The Computer Journal*, 14, 4, p. 232.
- 8) 中前・西田: 多面体の隠線消去の一手法, *情報処理* 13, 4, p. 239 (1972).
- 9) C. Wylie, G. Romney, D. Evans, R. Erdahl: Half-tone perspective drawings by computer, *Fall Joint Computer Conference*, (1967).
- 10) A. Appel: On calculating the illusion of reality, *Hardware*, p. 945 (1969).
- 11) G. W. Romney, G. S. Watkins, D. C. Evans: Real-time display of computer generated half-tone perspective pictures, *Hardware*, p. 973 (1969).
- 12) J. E. Warnock: A hidden surface algorithm for computer generated halftone pictures, *Dep. Comput. Sci., Univ. of Utah, Tech. Rep.*, 4-15, (Jun. 1969).
- 13) W. J. Bouknight: A Procedure for Generation of Three Dimensional Half-toned Computer Graphics Presentation, *Comm. ACM*, 13, 9, (Sep. 1970).
- 14) R. Goldstein, R. Nagel: 3-D visual simulation, *Simulation*, p. 25 (Jan. 1971).
- 15) H. Gouraud: Continuous shading of curved surfaces, *IEEE Trans. Computers*, p. 623 (June 1971).

(昭和48年8月24日 受付)

(昭和48年10月24日再受付)