

優先順位テーブルを用いた三次元物体の 陰影表示の一手法†

西田 友 是** 藤 井 克 典** 中 前 栄 八 郎***

三次元物体を透視図として表示するには、隠面消去を行う必要がある。この隠面消去の方法には種々の方法が開発されており、その一つに、隠面消去のための図形走査に先だて、画面中のすべての面の前後関係、すなわち奥行に関する優先順位を求めておく方法がある。本論文で提案する方法もこれに属すもので、凸多面体の集合を対象としており、次の特徴をもっている。(1)凸多面体単位に優先順位を決定することにより、視点(または光源)に対するその判定回数が減少し、また、その判定も容易となる。(2)前処理として、凸多面体の構成面のうちからある条件を満たす分割面と呼ばれる面を選択する。これにより、視点が与えられた際、分割面の視点に対する表裏を用いて即座に凸多面体同士の優先順位を決定することができる。なお、これらの分割面は優先順位テーブルにあらかじめ記憶しておく。(3)凸多面体の集合をいくつかのグループに分割して入力し、優先順位テーブルをグループ単位に作成することにより、移動物体を含む図形に対する優先順位の設定が容易となる。適用例により、提案する手法の有用性を示している。

1. ま え が き

三次元物体の透視図を表示することは、機器および建築物等の CAD/CAM や、教育および訓練用のシミュレータに有用な手段である。

三次元物体を表示する際に必要な隠面消去(線画の場合は隠線消去)についてはすでに多くの方法がある。これらは面の前後判定の方法により、次のように分類される¹⁾。すなわち、前後判定を画素単位、走査線単位、または判定が容易となるまで細分された領域単位に行う方法(それぞれ、depth buffer algorithm, scan line coherence algorithm, area coherence algorithm と呼ばれる)と、図形の走査の前に全画面中の面の前後関係を決定する方法(priority algorithm)に分類される。これらの方法の優劣は、対象となる図形の性質に依存するため、いずれの方法がよいかは結論づけられないが、フライトシミュレータ等の曲面を含まない図形を高速に処理する必要のあるものでは priority algorithm が多く採用されている²⁾。本論文も後者の範疇に属するもので、凸多面体の集合で構成される三次元物体の隠面消去に適した一手法を提案する。

いままでに開発されている priority algorithm は、視点が与えられる前、すなわち物体データ作成時に前処理として優先順位(視点到近のものほど優先順位は高いものとする)の決定に必要な情報を求めておく Schumacker^{3),4)} や Fuches^{5),6)} の方法、視点が与えられてから優先順位を決定する Newell⁷⁾ や筆者ら^{8),9)} の方法、および両者を併用する Wein¹⁰⁾ の方法の三つに大別される。

Schumacker の方法は、物体を凸多角形の集合として取り扱い、三次元空間をいくつかの平面(これを分割面と呼ぶ)により分割し、分割された空間内に存在する凸多角形の集合を一つのクラスタとし、クラスタ間およびクラスタ内の各面の優先順位を物体データ作成時にあらかじめ与えておく方法である。ただし、この方法はクラスタ内の各面の優先順位が一意に決まるようなクラスタを人手を介して決定する必要があり、また図形の形状によっては、データを修正する必要がある。この人手による処理を改善したのが Fuches の方法であり、入力された面データのすべてについて、その面を含む平面で空間を順次分割する方法である。したがって、図形の複雑さに伴って分割される面の数が多くなる欠点がある。

Newell の方法は、物体を任意の多角形の集合として取り扱い、面同士の前後判定を一方の面の方程式と他方の面の頂点座標との関係で求める方法である。この方法においては、サイクリックな図形(たとえば、A, B, C 面があるとき、A が B より、B が C より、C が A より手前にある図形)または凹多角形が他の面を

† An Algorithm for Half-Tone Representation of Three-Dimensional Objects by Using a Priority Table by TOMOYUKI NISHITA, KATUNORI FUJII (Faculty of Engineering, Fukuyama University) and EIHACHIRO NAKAMAE (Faculty of Engineering, Hiroshima University).

** 福山大学工学部電子電気工学科

*** 広島大学工学部第2類(電気系)

扱むような場合はデータ作成時に面を分割しておく必要がある。筆者らの方法は凸多面体を単位として取り扱っており、視点から見た凸多面体の輪郭線同士の交点において前後判定を行う方法である。したがって前後判定の回数を減少できる。

Wein の方法も凸多面体を単位として取り扱っており、Schumacker, Newell および筆者らの方法に、さらに一点を共有する物体同士の処理を組み合わせることにより、前後判定を能率よく行おうとするものである。しかし、この方法は各物体同士の前後判定をどの方法を用いて行うかをあらかじめ人手によって指定しておく必要がある。

ここで提案する方法は、文献 9) の方法を改良したものである。すなわち、Wein の方法と同様に、優先順位決定のための前処理と、視点が与えられてから優先順位を決定する方法を併用することにより全体的に効率のよい優先順位決定を行うものであるが、これらの処理を計算機により処理し、人間の介入を不要にしたものである。具体的には、視点が与えられる前に優先順位を決定する方法として凸多面体の構成面の一つの面を分割面として選択する。なお、分割面の存在しないものおよびグループ（物体の集合）間の優先順位に関しては視点が与えられてから決定する方法である。これにより移動物体の処理が容易になる。提案する方法は次の特徴をもっている。

(1) Schumacker, Newell, Fuches の方法のような面単位の処理に比べ、物体単位の処理を行うので、Wein の方法と同様に優先順位の判定回数が減少する。

(2) Schumacker, Wein の方法のようにあらかじめ人手を介して、空間の分割および手法の指定をする必要がない。

(3) 他方法 (Wein の方法は不明) は、優先順位が一意に決定される必要から、面を再分割する必要があり、そのため記憶容量が増加する可能性があるが、本手法では再分割する必要はない。

2. 優先順位テーブルを用いた三次元物体の隠面消去

ここで提案する優先順位テーブルを用いた隠面消去法の手順は、次のステップからなる。

(1) 物体データを入力し、優先順位テーブルを自動作成し、これらをディスクへファイルする。

(2) 視野内の物体を抽出した後、視点（または光

源）に対する透視面上へ各頂点を座標変換し、不可視面を求めるために、面の表裏を判定する。

(3) 視点（または光源）に対して、物体の前後判定（優先順位決定）を行う。なお、光源に対しては、影の領域を求めておく。

(4) 図形を上から下へと走査し、優先順位を用いて走査線上の各点での可視面を求め、CRT に出力する。

なお(2),(3)の処理については、まず視点に対して処理を行い、次に光源を視点とみなして(2),(3)の処理を光源数だけ繰り返す。

本論文では(1)と(3)について記述する。なお(2)と(4)については、文献 9) の手法と同一である。

2.1 準備

準備として次の条件と定義を与える。

(1) 取り扱う図形：三次元物体は凸多面体（凸多角形も含む）の集合として取り扱う。なお、球、円柱等の凸の曲面体は多面体に近似して処理する。

(2) 凸多面体の構成面：構成面は凸多面体外部から見て右回りの頂点列で与えられているものとし、その法線ベクトルは凸多面体の外部を向くものとする。

(3) グループ：必要に応じて(2.4節(1)参照)、いくつかの凸多面体で構成されるグループを作り、これを単位として入力する。このグループとは、一般に一つの機能を構成する凸多面体の集合、位置的に近い凸多面体の集合、あるいは移動物体を構成する凸多面体の集合で、それらの相対位置関係が変化しないものをいい、ユーザが任意に指定することができる。なお、この場合はすべての凸多面体は必ずいずれかのグループに属するものとする。

2.2 優先順位決定手順

視点が与えられた時点で、優先順位を効率的に決定するには、視点（または光源）から見て互いに重なり合わないものは簡単なテストによりあらかじめ除外し、徐々に複雑なテストを行うのが好ましい。そのため、次の手順により優先順位を決定する。

ステップ1) グループ間の優先順位を決定する。

ステップ2) 透視面上において凸多面体を包含する最小の長方形を考え、これらの長方形同士が重ならないものは互いに独立と判定する（この判定は、 x あるいは y 成分の最大値および最小値の比較により簡単にテストでき、*minimax test*^{1),4)}と呼ばれる）。

ステップ3) 同一グループ内の凸多面体同士については、優先順位テーブルを用いて優先順位を決定す

る。ただし、判定する両凸多面体に分割面が存在しない場合はステップ4)の方法を用いる。また、互いに他のグループに属する凸多面体同士については、ステップ1)の結果を用いて優先順位を決定する。ただし、グループ間の優先順位が決定できないものは、2.4節(2)で後述するように、この時点で優先順位テーブルに代わるものを求める。なお、判定する両凸多面体に分割面が存在しない場合は同様にステップ4)の方法を用いる。

ステップ4) ステップ3)において優先順位が決定されていない凸多面体については、凸多面体同士の輪郭線の交点を用いる方法により前後判定を行う。

2.3 グループ内の凸多面体の優先順位の決定法

(1) 優先順位テーブル

準備として頂点 $[P_1, P_2, P_3, \dots]$ よりなる面 S および任意の点 $P_w(X_w, Y_w, Z_w)$ に対して次の関数を定義する。

$$F_S(P_w) = (P_3 - P_1) \times (P_2 - P_1) \cdot (P_w - P_1) = aX_w + bY_w + cZ_w + d \quad (1)$$

ここで、 (a, b, c) は面 S の法線ベクトルである。

提案する方法は、凸多面体の構成面のうちから以下に述べる条件を満たす分割面を選択しておくことにより、容易に優先順位を決定する方法である。

いま、 n_f 個の面からなる凸多面体 V_i に属する面 $S_{i,l} (l=1, 2, \dots, n_f)$ と、 n_p 個の頂点 $P_k (k=1, 2, \dots, n_p)$ からなる凸多面体 V_j を考えると、次式を満たす面が V_j に対する V_i の分割面である。

$$F_{S_{i,l}}(P_k) \geq 0 \quad \text{for all } P_k \in V_j \quad (2)$$

ここで、(2)式は、面 $S_{i,l}$ を含む平面の正領域に、 V_j の全頂点が存在することを意味する。すなわち、 V_i と V_j が、 $S_{i,l}$ を含む平面に対して互いに反対側の半空間に存在する。なお(2)式を満たす面は複数個存在することもあるが、1個の分割面のみ求めておけばよい。この分割面を $\bar{S}_{i,j}$ と記述することにする。

この $\bar{S}_{i,j}$ を要素とする $n \times n$ (n は凸多面体数) の優先順位テーブルを視点の位置が与えられる前に求めておく。ここで、 $\bar{S}_{i,j}$ は面を示す記号であるが、優先順位テーブルを取り扱う際には面番号を示すものとする。また、図1(b)の V_j のように分割面が存在しない場合もあるが、この場合 $\bar{S}_{j,i} = 0$ と記述することにする。したがって、 $\bar{S}_{i,i} = 0$ である。

(2) 凸多面体同士の前後判定

前述のように、凸多面体同士の前後判定は、隠面消去および影の処理の場合に必要となる。視点(または

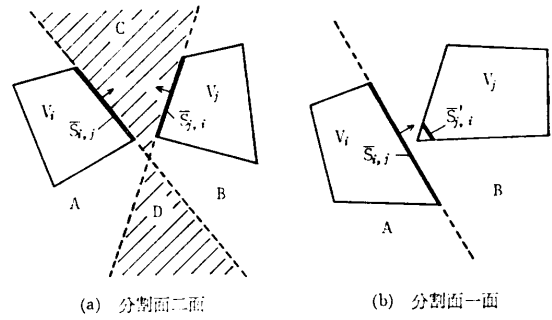


図1 凸多面体同士の優先順位
Fig.1 Priority between two convex polyhedra.

光源) が与えられた場合、凸多面体 V_i と V_j のいずれが他方を隠すかの判定方法について述べる。

前述のように、分割面が存在しない場合もあるから、前後判定をする凸多面体に対して、両方に分割面が存在する場合、一方にのみ存在する場合、両方ともに存在しない場合(このケースは一般の図形ではきわめて少ない)がある。両者に分割面が存在する場合には、図1(a)に示すように、 V_i の分割面 $\bar{S}_{i,j}$ と V_j の分割面 $\bar{S}_{j,i}$ を含む平面により切断される空間 A, B, C, D が生じる。この場合、視点がAの領域に存在すれば V_i が V_j を隠す可能性があり、Bの領域に存在すれば V_j が V_i を隠す可能性があり、CまたはDの領域内に存在すれば V_i と V_j は互いに他を隠さない性質がある。また一方の凸多面体のみ分割面が存在する場合は、図1(b)に示すように、Aの領域に視点が存在する場合は V_i が V_j を隠す可能性があり、Bの領域に視点が存在する場合には V_j が V_i を隠す可能性がある。なお、両方の凸多面体に分割面が存在しない場合は、当然ながら分割面を用いる方法は適用できないので、輪郭線同士の交点を用いる方法⁸⁾で前後判定を行う必要がある。

表1 優先順位決定テーブル
Table 1 Decision table of priority.

$H(\bar{S}_{i,j}, P_e)$	$H(\bar{S}_{j,i}, P_e)$	関係
1	1	互いに独立
-1	-1	
1	-1	V_j が手前
0	-1	
1	0	
-1	1	V_i が手前
-1	0	
0	1	
0	0	不定

分割面により決定される空間のいずれの領域に視点 P_v が存在するかは、 $\bar{S}_{i,j}$ (または $\bar{S}_{j,i}$) が視点に対して表の面 ($F\bar{S}_{i,j}(P_v) \geq 0$) となるか、裏の面 ($F\bar{S}_{i,j}(P_v) < 0$) となるかの組合せにより決定される (たとえば、 $\bar{S}_{i,j}, \bar{S}_{j,i}$ がともに表の面なら図1(a)では視点はCの領域に存在). したがって、 $\bar{S}_{i,j}$ が表なら1, 裏なら-1, $\bar{S}_{i,j}=0$ なら0とする関数 $H(\bar{S}_{i,j}, P_v)$ を定義すると、優先順位の関係は表1のように表される. したがって次式の符号で前後関係を判定できる.

$$H(\bar{S}_{i,j}, P_v) - H(\bar{S}_{j,i}, P_v) \begin{cases} > 0: V_j \text{ が手前} \\ = 0: V_i, V_j \text{ は互いに独立} \\ < 0: V_i \text{ が手前} \end{cases} \quad (3)$$

$$\bar{S}_{i,j} = \bar{S}_{j,i} = 0 \quad : \text{不定}$$

ここで、分割面が表か否かは、不可視面(裏の面は必ず不可視)を前後判定の対象から除外するための面の表裏の決定時にすでに求められているから、(3)式のみの計算で優先順位が決定できる. したがって、かなりの処理時間の節減が期待できる.

両方の凸多面体に対して分割面が存在しない場合は、前述のように輪郭線同士の交点を用いる方法により判定するので、多少計算時間が増加する. したがって、処理時間が優先される場合には、物体データの作成時に、若干データを変更することにより分割面を人為的に存在させることができる. すなわち、どちらか一方の凸多面体に対して図1(b)に示すように、凸多面体の形状を実際には変化させないで、かつ分割面となる仮想的な面 $\bar{S}'_{j,i}$ を付加すればよい. 具体的な作業としては、図1(b)の V_j に仮想面を付加する場合、 V_j の微小部分を削り取るか、または V_i と V_j との間に存在する面で、かつその面の正領域に V_i が存在する任意の面を考え、その面の係数((1)式参照)を付加する.

2.4 グループ間の優先順位

(1) グループ処理の意義

移動物体の処理を可能にするためと、記憶容量の節減のためにグループ処理を行う.

物体が時間的に移動するような図形を取り扱う場合、物体相互の位置関係が変化すると優先順位テーブルを作成するのでは効率が悪い. したがって、相対位置関係が不変なものを同一グループとして定義すれば、同一グループ内の優先順位テーブルは不変であるから、優先順位テーブルの有効利用が図られる.

一方、優先順位テーブルは凸多面体数の二乗の記憶容量を必要とする. したがって、物体数が多い場合には、グループ単位での優先順位テーブルを作成することにより、この記憶容量を節減できる. グループ処理をしない場合とする場合を比較すると、 n 個の凸多面体を m グループに等分割したと仮定すると、必要な記憶容量は n^2 から n^2/m に減少する.

なお、このグループ処理は、ディスク中の物体データベースから視野内の物体を抽出する処理にも有効である(文献11)参照.

(2) グループ間の優先順位の決定

グループ内の全凸多面体を包含する xy, yz, zx 平面にそれぞれ平行な平面より構成される最小の直方体を考え、この直方体を bounding box とする. ここで、各グループの bounding box は互いに干渉することを許すものとし、bounding box の構成面の法線ベクトルは外部を向くものとする.

いま、二つのグループ G_i と G_j を考えると、グループ間の優先順位は、両者が互いに干渉し合わない限り、 G_i と G_j の bounding box の構成面と視点との位置関係により、次のように求まる.

2.3節では、凸多面体の分割面は一面に限定したが、bounding box については複数個の分割面を考える. すなわち、二つのグループについて考えると、グループ同士が互いに干渉しない場合には、分割面は相方のグループに対してともに同数で、1, 2 または 3 面存在する. いずれの場合にしろ、同一 bounding box のすべての分割面が裏となる領域に視点が存在するなら、そのグループが他方のグループより手前にある. たとえば、図2の場合、視点がA (G_i の内部も含む)の領域に存在するなら G_i が手前で、B (G_j の内部も含む)の領域に存在するなら G_j が手前であり、他の

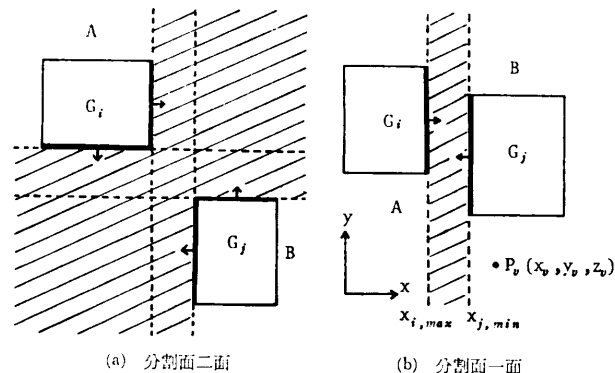


図2 グループ間の優先順位
Fig. 2 Priority between two groups of objects.

領域(図中斜線部)に視点が存在するなら G_i と G_j は互いに独立である。ここで, bounding box は x , y あるいは z の各成分におけるグループ内の頂点の最小値および最大値(6変数)によって与えられているから, 分割面の表裏判定は, これらの最小値あるいは最大値と視点座標の x, y, z 成分を比較することにより簡単に求まる。

図2(b)の例では, $x_{i,max} < x_{j,min}$ より $x = x_{i,max}$, $x = x_{j,min}$ の各平面が分割面であることがわかり, $x_{j,min} < x_0$ より $x = x_{j,min}$ の平面は視点に対して裏であることがわかる。

なお, グループの bounding box 同士が互いに干渉する場合は, 当然グループ間の優先順位は決定できない。したがって, グループ単位の処理は行わず, この二つのグループ内の物体については同一グループと考え, 2.2節で述べたステップ3)において(2)式を用いて分割面(すなわち, 優先順位テーブルの要素)を求め, この分割面の表裏で優先順位を決定する。ここで, 判定しようとする凸多面体同士のうち, 一方のみについて分割面を求めれば十分である。

3. 適用例

図3に応用例を示す。(a)は直方体が複雑に組み合わされた例である。(b)は任意の方向を向いた多角形が多く存在する例である。(c)は計算機室の例であり, 計算機, プロッタ装置, ラインプリンタ装置, 机, 天井, 床等をそれぞれ一つのグループとして入力してある。本例は各装置を移動して適切な配置の検討をするためのものである。(d)は単相変圧器の内部構造を示すためのカットモデルの例であり, コイル部の円筒は, 大きい円柱を小さい円柱で自動的にくり抜いて作成したものである。さらに断面を得るための自動切断も行われている。本例はデータ量が多いためグループ処理を行っており, ほとんどのグループが干渉している例である。

これらの例のデータ数および計算時間を表2に示す。表において, Iは物体データおよび優先順位テーブルの作成(ディスクへの登録を含む), IIは視野内の物体抽出(ディスクからの読み込みを含む), 透視変換および面の表裏判定, IIIは優先順位の決定(優先順位テーブルのディスクからの呼び込みも含む), IVは影の領域

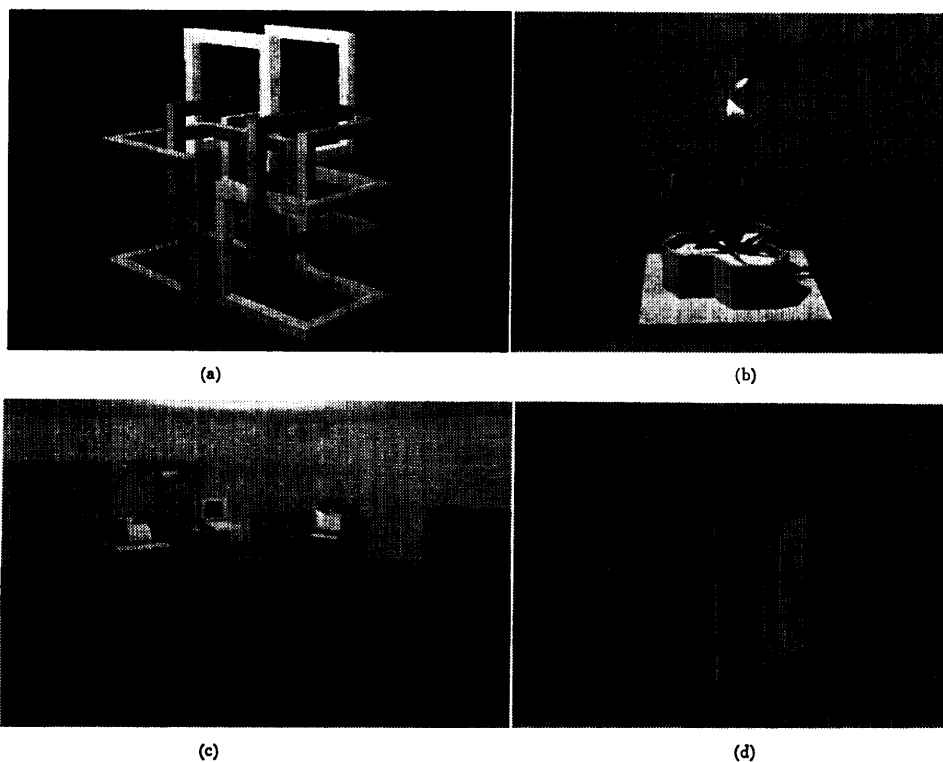


図3 応用例
Fig. 3 Examples.

表 2 データ数および計算時間
Table 2 Data and computation time.

適用例		(a)	(b)	(c)	(d)
グループ数		0	0	9	3
物体数		40	44	36	56
面数		240	200	205	372
光源		平行光源	平行光源	点光源	平行光源
計算時間 (秒)	I	3.41(2.38)	5.27(4.18)	4.73(4.45)	9.87(8.70)
	II	1.11	1.14	1.26	1.73
	III	0.11(1.70)	0.13(1.14)	0.08(0.99)	0.40(4.52)
	IV	—	1.06(1.76)	1.33(1.50)	—
	V	66.75	43.91	165.35	132.65

の決定, Vは図形の走査および濃淡計算の各時間を示す. また () 内は優先順位テーブル法を用いない場合(文献9)の方法の時間を示す. 表からわかるように, 優先順位テーブルを作成するために, Iは多少時間は長くなるが, この処理はデータ作成時の一度のみでよく, 与えられた視点に対する優先順位の決定は8~15倍早くなった. いずれの例も優先順位の決定に凸多面体の輪郭線を用いる方法を一度も使用する必要がなかったことから, 計算時間がかなり短縮された. 図形の走査時間(表中V)に多くの時間を要しているが, フライトシミュレータ²⁾では一般にVの図形走査はハードウェアが使用されており, 短時間で処理できる. また, 隠線消去問題では, IV, Vの処理は不要であることから, IIIの処理時間の短縮は重要な意味をもつものである.

優先順位の決定に要する処理時間について, 他の方法との比較をする.

Weinの方法は, 処理時間が短縮するように, 人間があらかじめ各物体間に適用する方法(Schumacker, Newellの方法等)を指定するものである. したがって, 人間の介入に要する時間を除外すれば, 当然処理時間は早いことが推測される. また, Fuchesの方法はSchumackerの方法を発展させたものであるから, 以下にFuches, Newellおよび本手法についての比較を行う.

FuchesおよびNewellの方法は面を単位として優先順位を決定する方法である. したがって, 多面体を単位とする筆者らの方法と比較すると, 最も少ない面で構成される物体(四面体)の集合の場合でも, 判定対象の数は4倍となる. 一方, Fuchesの場合, 面の位置関係の記憶形式は木構造であるから, Newellや本手法のような面数および多面体数の二乗に比例する方法より処理時間が短い. しかし, 木構造化のための面の分割が必要であり, 物体が複雑になるに従って,

面の数が加速度的に増加する欠点がある(文献6)によると最悪時には3乗に比例). したがって, 複雑な図形の場合には, 筆者らの方法が優位であると推測できる.

なお, 本適用例の計算にはTOSBACデータシステム600およびグラフィカM508R(512×212画素, R, G, B各256レベル)を用いた.

4. む す び

優先順位の決定のための前処理と, 視点を与えられてから優先順位を決定する方法を併用することにより, 効率よく優先順位を決定する方法を提案した. すなわち, 前処理として, 凸多面体の構成面のうちから分割面を選択しておけば, 与えられた視点に対してその分割面が表か裏かを知るのみで即座に凸多面体同士の前後判定が可能となる方法を述べ, また, その方法を適用できないものは凸多面体の輪郭線を利用して前後判定する方法について述べた. さらに, 移動物体の処理記憶容量の節減に適したグループ処理についても論じた.

提案した手法は, 次の利点を有することが明らかになった.

(1) 視点(または光源)に対する凸多面体の構成面の表裏を求めることは, 不可視面(あるいは陰の面)の抽出と同時に凸多面体同士の前後判定にも利用できるので有効な方法である.

(2) 二つの凸多面体同士が一つも分割面をもたない場合を除き(グループ同士が干渉した場合も若干の処理が追加されるが), 簡単な方法で前後判定ができるから, 計算時間がかなり短縮できる. また, データの修正が許されるなら, すべて簡単な方法で処理できる. この場合, (1)式, (3)式のように簡単な関数を用いることができ, かつその演算も簡単なので, ハード化が容易である.

(3) priority algorithm中の, 視点を与えられる前に優先順位を決定する方法は, 一般に移動物体がある図形に適用することが困難である. しかし, 本手法は, 優先順位のための前処理と, 視点を与えられた後に優先順位を決める方法が組み合わされているため, 移動物体がある図形にも適用が可能である.

(4) 一つの凸多面体を構成する面同士の優先順位は不要であり, 他の凸多面体に対する優先順位はすべて同一とみなせることから, 凸多面体単位に処理する方法を用いた. その結果, 隠面消去や影の処理を凸多

面体を単位として取り扱うことが可能となり（多くの隠面消去法では面を単位としている）、これらの判定回数を減少することができる。

(5) グループ処理は、移動物体の処理を可能にすると同時に、優先順位テーブルの作成時間や視野内の物体抽出時間の短縮にも効果がある。

謝辞 本計算例のデータの作成は、広島大学電気機器工学研究室の学生諸君の協力をいただいたことを付記し、感謝の意を表す。

参 考 文 献

- 1) Newman, W. M. and Sproull, R. F.: *Principles of Interactive Computer Graphics*, pp. 367-388, McGraw-Hill, New York, (1979).
- 2) Schachter, B. J.: Computer Image Generation for Flight Simulator, *IEEE CG & A*, Vol. 1, No. 4, pp. 29-68 (1981).
- 3) Schumacker, R. A. et al.: Study for Applying Computer-Generated Images to Visual Simulation, AFHRL-TR-69-14 (1969).
- 4) Sutherland, I. E., Sproull, R. F. and Schumacker, R. A.: A Characterization of Ten Hidden-Surface Algorithm, *ACM Comput. Surv.* Vol. 6, No. 1, pp. 1-55 (1974).
- 5) Fuchs, H. and Kedem, Z. M.: Predetermining Visibility in 3-D Scene, *Comput. Gr.*, Vol. 14, No. 2, pp. 175-181 (1979).
- 6) Fuchs, H. and Kedem, Z. M.: On Surface Generating by a Priority Tree Structures, *Comput. Gr.*, Vol. 14, No. 3, pp. 124-133 (1980).
- 7) Newell, M. E., Newell, R. G. and Sancha, T. L.: A New Approach to the Shaded Picture Problem, *Proc. ACM, Nat. Conf.*, p. 443 (1972).
- 8) 中前, 西田: 多面体の隠面消去の一手法, *情報処理*, Vol. 13, No. 4, pp. 239-246 (1972).
- 9) 西田, 中前: 三次元物体の陰影表示の方法, *情報処理*, Vol. 15, No. 1, pp. 18-25 (1974).
- 10) Wein, M., Tanner, P. et al.: Hidden Line Removal for Vector Graphics, *Comput. Gr.*, Vol. 12, No. 3, pp. 173-180 (1978).
- 11) 西田, 中前: 画面分割法による三次元物体表示および遠近処理, *情報処理学会, コンピュータビジョン研究会* 5-4 (1980).

(昭和 57 年 8 月 20 日受付)

(昭和 58 年 1 月 17 日採録)