

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

(福山大学工学部) (広島大学工学部)

1. はじめに

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り、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、いずれの方法が良いかは決論付けられないが、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があるものではpriority algorithmが多く採用されている²⁾。本論文は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到適した一手法を提案する。

今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(視点に近い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っており、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ーとすると、クラスター内の各面およびクラスター間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ー内の各面の優先順位が一意に決まるようなクラスター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っており、面同志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(例えば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挟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志の交点において前後判定を行う方法である。したがって、前後判定の回数が減少できる。

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志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

同志の前後判定をどの方法を用いて行うかを予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の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。なお、凸多面体の構成面の一つの面を分割面とする方法であり、移動する物体の処理を可能とするために、グループ処理による優先順位の決定法を採用している。提案する方法は次の特徴をもっている。

(1) Schumacker, Newell, Fuchs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の方法と同様に判定回数が減少する。

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

(3) 他の方法(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) 物体データおよび優先順位テーブルを作成し、これらをディスクファイルする。

(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(優先順位の決定)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) グループ：必要に応じて(2.4(1)参照)、幾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と言い、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ものとする。

(3) 凸多面体の構成面：構成面は凸多面体外部から見て右廻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。

2.2 優先順位の設定手順

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予め除外し、除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志については、優先順位テーブルを用い、互いに他のグループに属す凸多面体同志については、ステップ1)の結果（グループ間の優先順位が決定できないものは、この時点で優先順位テーブルに代わるものを求める）を用いて優先順位を決定する。

ステップ4) 優先順位が決定されていないものについては、凸多面体同志の輪郭線の交点を用いて前後判定を行う。

2.3 グループ内の凸多面体の優先順位の設定法

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 および点 $P_u(x_u, y_u, z_u)$ に対して、次の関数を定義する。

$$F_S(P_u) = (P_3 - P_1) \times (P_2 - P_1) \cdot (P_u - P_1) = aX_u + bY_u + cZ_u + d \quad (1)$$

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、一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。図1(b)のように V_i に分割面が存在しない場合もあるが、この場合 $\bar{S}_{i,j}=0$ 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(当然 $\bar{S}_{i,i}=0$)を物体データ作成時に作成する。

2) 凸多面体同志の前後判定

前述のように、凸多面体同志の前後判定は、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極めて少ない)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、視点がある領域に存在すれば 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に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、輪郭線同志の交点を用いる方法²⁾で前後判定を行う必要がある。

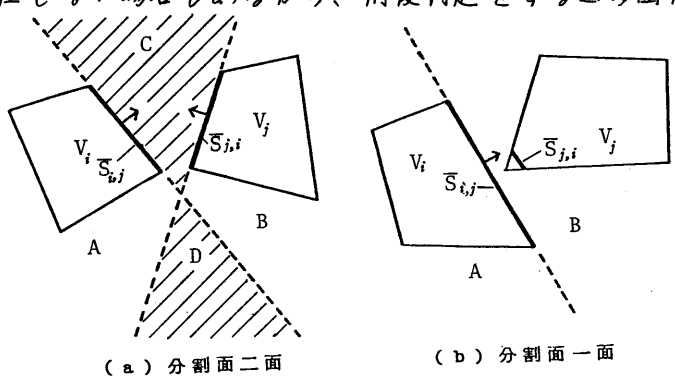


図1 凸多面体同志の優先順位

表1 優先順位決定テーブル

$HS_{i,j}(P_v)$	$HS_{j,i}(P_v)$	関係
1	1	独立
-1	-1	
1	-1	V_j が手前
0	-1	
1	0	V_i が手前
-1	1	
-1	0	
0	1	不定
0	0	

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($FS_{i,j}(P_v) \geq 0$)となるか、裏の面($FS_{i,j}(P_v) < 0$)となるかの組み合わせにより決定される(例えば、 $\bar{S}_{i,j}$, $\bar{S}_{j,i}$ が共に表の面なら図1(a)ではCの領域)。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 $HS_{i,j}(P_v)$ を定義すると、優先順位の関係は表1のように表わされる。したがって次式の符号で判定できる。

$$HS_{i,j}(P_v) - HS_{j,i}(P_v) \begin{cases} > 0 : V_j \text{が手前} \\ = 0 : \text{独立} \\ < 0 : V_i \text{が手前} \end{cases} \quad (3)$$

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

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。従って、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志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。従って、処理時間が優先される場合には、物体データを若干変更することで分割面を存在さすこ

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)のように、 V_i の形状を大きく変えない程度の微小の面で、かつ分割面となる S_{ij} を付加すればよい。

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とするためと、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毎に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ることにより、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計られる。

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、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 m^2 に減少する。

なお、このグループ処理は、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含包する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で述べたように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共に同数で、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である。例えば、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、

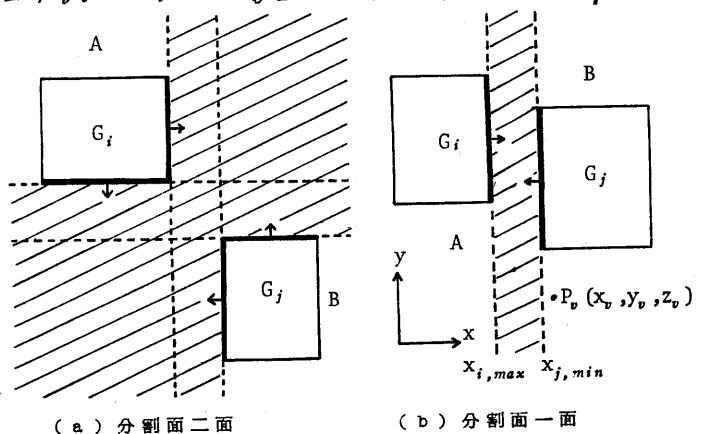


図2 グループ間の優先順位

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 $x = x_{j,max}, x = x_{j,min}$ の各平面が分割面であることが判かり、 $x_{j,min} < x$ より $x = x_{j,min}$ の平面は視点に対して裏であることが判かる。

なお、グループ同志が干渉する場合は、当然グループ間の優先順位は決定できない。したがって、これらのグループに属す凸多面体間の優先順位テーブルは与えられていないので、2.2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めることができる。ただし、判定しようとする凸多面体同志のうち、一方のみについて分割面を求めれば充分である。

3. 適用例

図3に適用例を示す。(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、プロッター装置、ライプリンター装置、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ガ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い、画面分割法¹⁾を用いて表示したもので、ほとんどのグループが干渉している例である。

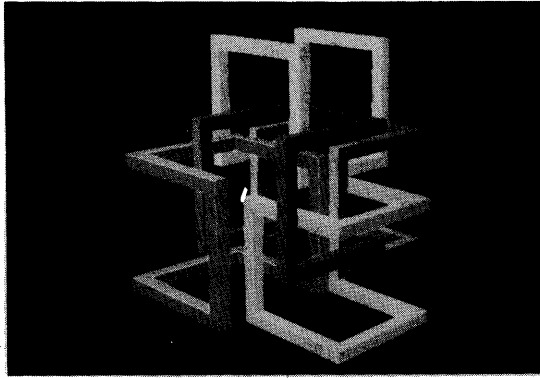
これらの例のデータ数および計算時間を表2に示す。表2において、Iは物体データ及び優先順位テーブルの作成（ディスクへの登録を含む）、IIは視野内の物体抽出、透視変換及び面の表裏判定、IIIは優先順位の決定（優先順位テーブルのディスクからの呼び込みも含む）、IVは影の領域の決定、Vは図形の走査の各時間を示す。また(c)内は優先順位テーブル法を用いない場合(文献¹⁾の方法)の時間を示す。表から判かるように、優先順位テーブルを作成するために、Iは多少時間は長くなるが、この処理はデータ作成時の一度のみでよく、与えられた視点に対する優先順位の決定は8~12倍早くなった。いずれの例も優先順位の決定に凸多面体の輪郭線を用いる方法を一度

表2 データ数および計算時間

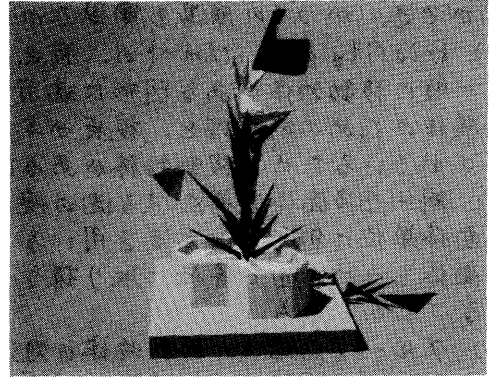
適用例	(a)	(b)	(c)	(d)	
グループ数	0	0	9	3	
物体数	40	44	36	56	
面数	240	200	205	372	
光源	平行光源	平行光源	点光源	平行光源	
計算時間 (秒)	I	9(5)	10(6)	9(8)	18(13)
	II	1.0	0.7	1.1	3.0
	III	0.4(5.1)	0.5(4.3)	0.3(3.7)	1.6(18.6)
	IV		4.6(5.8)	8.5(9.2)	
	V	190	125	425	329

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。

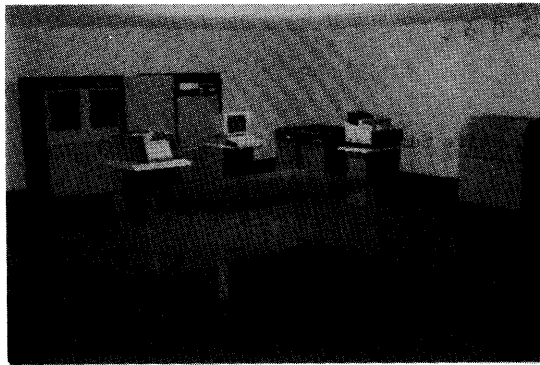
なお、本適用例の計算には



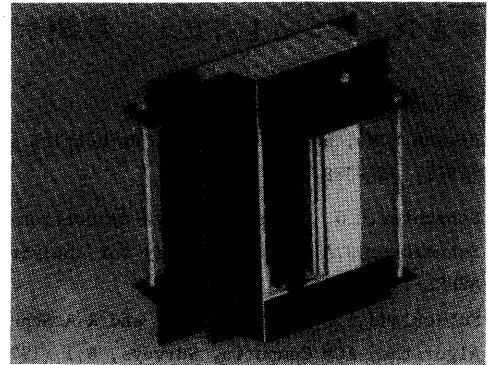
(a)



(b)



(c)



(d)

図3 適用例

OKITACシステム 50/40で(主記憶装置のユーザーエリアは約48k \times W)およびグラフィカ M508R(512 \times 512画素, R, G, B各256レベル)を用いた。

4. おわりに

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、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志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志の前後判定にも利用できるのが有効的な方法である。

(2) 二つの凸多面体同志が一つも分割面をもたない場合を除き(グループ同志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、視点が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、視点が与えられた後に優先順位を決める方法が組み合わせられているため、移動物体がある図形にも適用が可能である。

(4) 同一凸多面体を構成する面の優先順位は、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ることにより、種々の判定に対して、一つの凸多面体を一つの面と等価に取り扱うことが可能となり、これらの判定回数を削減することができる。

(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には、広島大学電気機器工学研究室の学生諸君の協力を頂いたことを付記し、感謝の意を表す。

[参考文献]

- 1) Newman, W.M., and R.F. Sproull, Principles of Interactive Computer Graphics, McGraw-Hill (1979) 367-388
- 2) Schachter, B.J., Computer Image Generation for Flight Simulator, IEEE CG&A, Vol.1, No.4 (1981) 29-68
- 3) Schumacker, R.A., et al., Study for Applying Computer-Generated Images to Visual Simulation, AFHRL-TR-69-14 (1969)
- 4) Sutherland, I.E., R.F. Sproull, and R.A. Schumacker, A Characterization of Ten Hidden-Surface Algorithm, ACM Computing Surveys, 6,1 (1974) 1-55
- 5) Fuchs, H., and Z.M. Kedem, Predetermining Visibility in 3-D Scene, Computer Graphics, Vol.14, No.2 (1979) 175-181
- 6) Fuchs, H., and Z.M. Kedem, On Surface Generation By a Priori Tree Structures, Computer Graphics, Vol.14, No.3 (1980) 124-133
- 7) Newell, M.E., Newell, R.G., and Sanch, T.L., A New Approach to the Shaded Picture Problem, Proc. ACM, Nat. Conf. (1972) 443
- 8) 中前・西田：多面体の隠線消去の一手法, 情報処理, Vol.13, No.4 (1972) 239-246
- 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 (1974) 18-25
- 10) Wein, M., P. Tanner, et al., Hidden Line Removal for Vector Graphics, Computer Graphics, Vol.12, No.3 (1978) 173-180
- 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)