

マルチスキャンニング法によるスムーズエッジ処理 を施した三次元物体の陰影表示[†]

西田 友 是^{††} 中前 栄 八 郎^{†††}

計算機により創成された三次元物体をラスタスキャンディスプレイ上に表示する場合、面の境界部が階段状となったり、小さい物体が見え隠れするエリアジグを生じる。これは画素数に対応した有限な離散点で画像をサンプリングするために生じる。この問題を解決する方法としては、サンプリング点数を増加させる方法、各画素内に占める図形の面積を詳細に求める方法等が開発されている。また、これらの方法は隠面消去の方法と大きな係わりをもつ。ここで提案する方法は、隠面消去をスキャンライン単位で行う方法に適した方法であり、各画素の色は走査線を増加することにより求める。すなわち、増加した走査線上の図形の存在区間および輝度を求め、それらに適当なフィルタをかけることにより画素の色を決定する。なお、フィルタはフーリエウインドとガウシアンフィルタを用いた。提案する手法は、画素内の面積を詳細に求める方法よりもアルゴリズムが簡単であり、水平方向に連続的にサンプリングすることから、サンプリング点数を増加させる方法よりもよい画質の得られる方法である。

1. ま え が き

コンピュータにより生成された三次元物体の陰影表示は、CAD、アニメーション、飛行訓練用シミュレータ等の広範囲な分野において非常に有用な手段として注目され、また活用されている。しかし、コンピュータにより創成された三次元物体の陰影表示画像は、一般にその面の境界部に階段状のぎざぎざ (jagged edge) を生じる。これは CRT 上に表示したい図形を有限の点 (一般には、スクリーン上の画素数に対応) でサンプリングして表示するために起こるもの (aliasing) として知られている。

サンプリング点数が十分でない場合、次のような種の問題が生じる。すなわち、①前述のように面の境界が階段状に表示される、②小さい物体あるいは細長い物体が表示されないことがある、③周期的に分布する画像ではモアレパターンを生じる、④動画の場合、面の境界が波打ち現象 (境界が揺れて見える) を示し、スムーズな動きを示さない、⑤動画の場合、小さい物体が見え隠れすることがある。

上述の諸問題を解決し、高品質の画像を得るには、各画素の色 (単色の場合は輝度) を、その画素内の一点の色 (たとえば画素の中心) で決定すべきではなく、画素内およびその近傍の図形の情報をも考慮して

決定する必要がある。このため、各画素に係わる面 (図柄および影も面として扱うことができる) の色を精度よく求め、それらを平均 (アベレージング) あるいは荷重平均 (フィルタリング) することにより画素の色を決定する必要がある。なお、このための処理方法は、ここで取り扱うような三次元物体の陰影表示の場合、隠面消去の方法に大きな係わりをもつ。

ここで提案する方法は、隠面消去をスキャンラインごとに行う場合に適した方法で、各画素に係わる情報をより多く得るために、各走査線間に仮想的な走査線を設け、かつ、フィルタリングを能率よく行って画素の色を決定するものである。

2. スムーズエッジ処理の基本的な考え方

2.1 スムーズエッジ処理の分類と提案手法の特徴

提案する手法の位置付けを明確にするために、いままでに開発されている方法について述べる。これらの方法の分類は、以下の二つの観点から行うことができる。

第一に、画像の生成順からみると、次の三つに分類できる。

(A-1) 画像を生成した後、表示された面の境界に沿って進みながら、境界部の色をこの境界を構成する二つの面の中間色に修正する方法¹⁾。

(A-2) 表示すべき画素数の数倍の画素数について色の計算を行い、それらをアベレージングあるいはフィルタリングにより、表示すべき CRT の画素数に応じてデータを圧縮する方法 (たとえば、 $2,048 \times 2,048$ の色から 512×512 の画像を求める)²⁾。

[†] Half-Tone Representation of 3-D Objects with Smooth Edges by Using a Multi-Scanning Method by TOMOYUKI NISHITA (Faculty of Engineering, Fukuyama University) and EIHA-CHIRO 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)

(A-3) 画像生成時に同時にスムーズ処理を行うもので、物体空間上の面をフィルタリングして各画素の色を決定する方法(プリフィルタリングと呼ばれる)³⁾⁻⁸⁾。

ここで、A-1の方法は処理時間が早く、単純な図形に対しては画質がよい。しかし、一画素中に複数個の面の境界が通過するもの(たとえば、細長い面)には適用できない。A-2の方法では、高品質の画像を得るにはサンプル点数をかなり多くする必要があり、その場合記憶容量や処理時間が問題となる。したがって、複雑な図形に対してはA-3の方法が適している。A-1、A-2の方法は隠面消去処理とは関係なく独立に処理できるが、A-3の方法は隠面消去の方法との関連が大きい。したがって、それぞれの隠面消去法に適した方法が開発されている。すなわち、隠面消去の方法に関してさらに分類すると、

(A-3.1) 走査線単位に可視面を求めるスキャンラインアルゴリズムに適した方法³⁾⁻⁵⁾、

(A-3.2) 表示すべき面の走査の前に、あらかじめ面の可視部分を切断しておくポリゴンクリップ法に適した方法⁶⁾、

(A-3.3) 各画素単位ごとに可視面を求めるZ-バッファ法に適したもの⁷⁾およびレイ・トレーシング法に適した方法⁸⁾、

がある。なお、どの隠面消去法がよいかは対象図形の性質により異なるので、一概には決定できない。

第二に、各画素の色の決定方法の観点から分類すると、画素の色を決定するフィルタの形状と、フィルタの処理方法により次のように分類することができる。まず、前者についてはフィルタ形状が正方形かその他の形状かにより、

(B-1.1) フーリエウィンド* (一般に一つの画素内のみを対象とする。ただし、文献1)の方法は境界に沿うようにウィンドが傾いている¹⁾⁻⁵⁾、

(B-1.2) 円錐フィルタ、ガウシアンフィルタ等の軸対称のもの(一般に、隣接する画素の領域も考慮)^{2),6),7)}、

に分類できる。前者はウィンド内に占める各面の面積を求め、それらの面積を考慮して画素の色を決定することに等しい。一般に後者のほうが若干画質がよいとされている。

フィルタの処理方式から分類すると、代表的な方法

として次の二つが挙げられる。

(B-2.1) 画素内にサブピクセルを発生し、各サブピクセルにおける色を平均(B-1.1の方法に対応)または荷重平均(B-1.2の方法に対応)することにより画素の色を決定するもの^{3),7),8)}。

(B-2.2) 各ウィンドの係わる面をウィンド形状で分割して、詳細におのおのの面積(B-1.2の場合は体積に相当)を求める方法⁴⁾⁻⁶⁾。

後者のほうが計算精度はよいが、その反面処理がかなり複雑となる。

以上、現在までに開発されている三次元物体の陰影表示に関するスムーズエッジ処理について述べた。本論文で提案する方法は以下の特徴をもつものである。すなわち、処理が簡単で、かつ精度をある程度確保できる方法(B-2.1とB-2.2の中間的な方法)で、プリフィルタリング(A-3の方法)に属し、フィルタ形状については、B-1.1とB-1.2の両方に適用でき、かつ一般によく用いられるスキャンラインアルゴリズム(A-3.1)に適した手法である。なお、最も画質のよい方法としては、B-2.2の方法を用いて精度よく色を求める方法⁶⁾があげられるが、この方法が適用できるのはポリゴンクリップ法(A-3.2)に限定されている。

2.2 提案手法の基本的な考え方

前述のように、本論文ではプリフィルタリングに属す一手法について提案するが、三次元物体のカラー陰影表示にスムーズエッジ処理を施す場合、次の点を考慮する必要がある。①対象図形に、物体構成面のほかに窓等の図柄や木目、花柄等のテクスチャが含まれているか。②物体上の影の処理を行うか。③隠面消去との関係、すなわち、隠面消去を物体空間上で行うか、または透視面上に座標変換後に行うか。④隠面消去を行う際に同時にスムーズエッジ処理をするか、または隠面消去処理後の図形にスムーズエッジ処理を行うか。以上の点である。

本論文では、図柄、テクスチャ、影の処理の可能な方法について述べるが、説明を簡単にするため、次の二段階に分けて説明する。まず、本節では、透視面上に投影された面のみで構成される図形を考え、それらの各面の可視部分はわかっているものとして提案手法の基本的な考え方を説明する。次に、3章において、上記の考え方に図柄および影の処理、ならびに隠面消去処理を付加した際の方法を説明する。したがって、本節では、すでに透視変換された二次元上の面を対象

* ここでは二次元フーリエウィンドを示し、サンプリング窓が正方形のものをいう。また、窓内の重みは一定である。

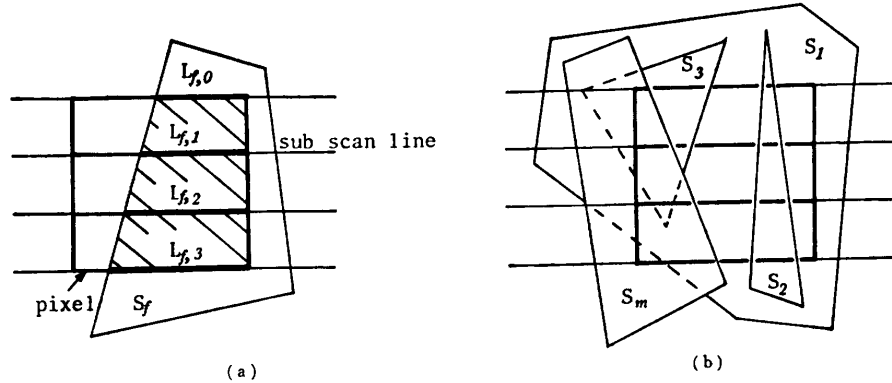


図 1 台形積分による面積計算
Fig. 1 Area calculation by using the trapezoidal integral.

とするが、本来処理対象は三次元物体であるため、一画素中に占める物体は重なり合うこともある (A-3.2の方法では、重なり合う面がなくなるように前処理をする必要がある)。そのため物体の可視部分は当然のことながら種々の形状をしている。したがって、それらの形状に対処できる色計算の方法が必要となる。そこで、この色の簡単な計算法について提案する。すなわち、おのおのの走査線の間数本の仮定の走査線を考え、これにフーリエウィンドまたは軸対称のフィルタを適用して画素の色を決定する二つの方法を提案する。

2.2.1 フーリエウィンドを用いる方法

図1に示すように、スキャンラインを構成する各画素は正方形と仮定する。次に、画素を水平方向に等分割する仮想的な走査線を仮定し、画素の境界を通る走査線を含め、これらをサブスキャンラインと呼ぶことにする。ここで、一画素の幅を単位長さとし、サブスキャンラインによる画素の分割数を n (図1(a)では $n=3$) とする。また、画素は、左から右に番号付けされているものとし、 i 番目の画素を画素 i とする。

図1(a)において、面 S_f の画素内の面積は、台形積分法により簡単に求めることができる。ある画素内のサブスキャンラインと面 S_f との交差部分の長さを $L_{f,i}$ ($i=1, 2, \dots, n$, ここで、 $i=0, n$ は画素の境界に相当) とすると、面 S_f の面積 a_f は次式により求められる。

$$a_f = \frac{1}{2n}(L_{f,0} + L_{f,n}) + \frac{1}{n} \sum_{i=1}^{n-1} L_{f,i} \quad (1)$$

いま面 S_f の赤色成分 R の輝度を I_f 、背景の輝度を零とすると、画素の R 成分の輝度は $a_f \cdot I_f$ となる。一般には、図1(b)に示すように、対象とする画素内に複

数個の面が存在する。その面の個数を m とすると、画面 i の R 成分の輝度 C_i は次式により決定される。

$$C_i = \sum_{f=1}^m a_f \cdot I_f \quad (2)$$

緑(G)、青(B)の各成分についても同様に計算し、 R 、 G 、 B の各成分を合成すれば色として表現できるので、以下 C_i を色として取り扱う。また I_f も各色成分を含んだものとする。(2)式に(1)式を代入すると次式を得る。

$$C_i = \sum_{l=0}^n d_l \cdot C'_{i,l} \quad (3)$$

ただし、

$$d_l = \begin{cases} 1/(2n) & (l=0, n \text{ のとき}) \\ 1/n & (0 < l < n \text{ のとき}) \end{cases}$$

$$C'_{i,l} = \sum_{f=1}^m L_{f,l} \cdot I_f$$

(3)式は次のことを意味する。すなわち、まず、おのおののサブスキャンライン上で、一画素内に占める各面の可視部分の長さを求め、それらの長さからそのサブスキャンラインに対する色 $C'_{i,l}$ を求める。各サブスキャンラインに対する色に重み係数 d_l を乗じて加算することにより、一画素の色が決定される。なお、 $C'_{i,n}$ は、処理中の画素のすぐ下に隣接する画素の初期値として利用される。すなわち、画素の境界を通るサブスキャンライン上での色は、その上下に存在する画素の色に寄与する。したがって、一画素の色は $(n+1)$ 本のサブスキャンラインによって決定される。

2.2.2 軸対称形状のフィルタを用いる方法

前述の方法においては、一画素を n 等分するサブスキャンラインを用いたが、ここでは上下に隣接する各画素の中心間を n 等分するサブスキャンラインを考え

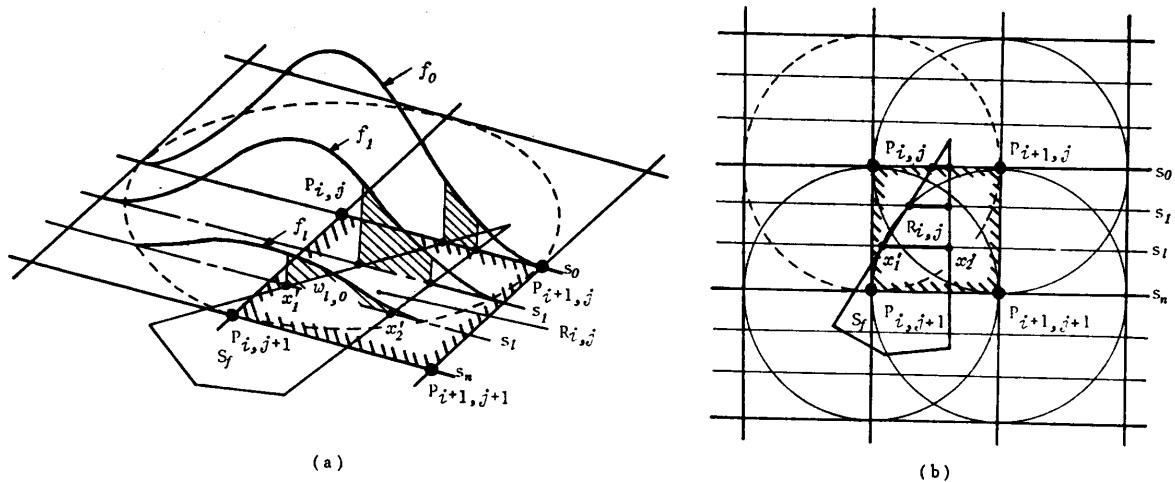


図2 軸対称フィルタを用いた色計算
Fig. 2 Color calculation by using an axis-symmetric filter.

る。また、画素の位置に関して、スクリーン左上の画素の中心を原点とし、水平方向を i 、垂直方向に j を割当て、画素 (i, j) と表現するものとする。

いま、図2に示すように、四つの画素の中心点 $P_{i,j}$ 、 $P_{i+1,j}$ 、 $P_{i,j+1}$ 、 $P_{i+1,j+1}$ によって囲まれる正方形の領域を $R_{i,j}$ とし、 $P_{i,j}$ を中心とする任意の形状（ただし、 $P_{i,j}$ を通りスクリーンに垂直な軸に対して軸対称）の画素幅を半径とするフィルタを考え、その体積は1となるように正規化する。

図2において、画素 (i, j) の色に対して面 S_r が影響を及ぼす割合は、面 S_r の上部に存在するフィルタの体積に依存する。この体積は区分別積法により、以下のようにして求めることができる。すなわち、各サブスキャンラインにおいて、面 S_r が存在する区間上のフィルタの面積（サブスキャンライン上の面 S_r を含む線分と、この線分を含み面 S_r に垂直な面上に描かれるフィルタ特性曲線により囲まれる面積、図2(a)中の斜線部の面積）を求め、これらの総和にサブスキャンライン間隔 $(1/n)$ を乗じることにより得られる。そこで、おのおのサブスキャンライン上の上述の面積を、それぞれのサブスキャンライン上で面 S_r の寄与率と呼ぶことにする（この寄与率は各画素の色計算をする際の重み係数となる）。

色計算のために次の準備を与える。①おのおのサブスキャンライン上において、領域 $R_{i,j}$ の左端を原点とする x' 座標系を考える $(0 \leq x' \leq 1)$ 。②各サブスキャンライン s_l ($l=1, 2, \dots, n$) 上のフィルタ形状を示す関数を $f_l(x')$ とし、それを零から x' まで積分したものを $F_l(x')$ とする。なお、本手法は基本的には任

意の形状のフィルタを取り扱うことができるが、特性のよいフィルタはガウシアンフィルタや円錐フィルタのように、一般に中央部が高く周辺になるほど低くなる。したがって、フィルタの有効半径が画素幅の場合、すべての x' に対して $f_n(x')$ は零または微小となるから、処理時間を考慮して $f_n(x')$ は無視する。③フィルタの体積が1となるように F_l を正規化する。なお、この関数 $F_l(x')$ はあらかじめ計算しておく、ルックアップテーブルに記憶しておく（本論文では、 x' を64等分してテーブル化した）。

領域 $R_{i,j}$ における色は次のようにして求まる。サブスキャンライン s_l と面 S_r との交点を x'_1 、 x'_2 ($x'_1 < x'_2$) とすると、 s_l 上における S_r の画素 (i, j) への寄与率は $F_l(x'_2) - F_l(x'_1)$ で求まる。ここで、図2(b)に示すように、領域 $R_{i,j}$ は $P_{i+1,j}$ 、 $P_{i+1,j+1}$ 、 $P_{i,j+1}$ を中心とするそれぞれのフィルタの領域にも含まれる。したがって、 $R_{i,j}$ を構成する4頂点 $P_{i,j}$ 、 $P_{i+1,j}$ 、 $P_{i+1,j+1}$ 、 $P_{i,j+1}$ の各画素に対する面 S_r (サブスキャンライン s_l 上で、かつ $R_{i,j}$ 領域の面 S_r) の寄与率 $W_{l,0}$ 、 $W_{l,1}$ 、 $W_{l,2}$ 、 $W_{l,3}$ は次式で与えられる。

$$\begin{aligned} W_{l,0}(x'_1, x'_2) &= F_l(x'_2) - F_l(x'_1) \\ W_{l,1}(x'_1, x'_2) &= F_l(1-x'_1) - F_l(1-x'_2) \\ W_{l,2}(x'_1, x'_2) &= F_{n-l}(1-x'_1) - F_{n-l}(1-x'_2) \\ W_{l,3}(x'_1, x'_2) &= F_{n-l}(x'_2) - F_{n-l}(x'_1) \end{aligned}$$

(4)

このように、一つの領域 $R_{i,j}$ 処理時に近傍の四つの画素の色に対する $R_{i,j}$ での寄与率が求まり、他の領域 $R_{i-1,j}$ 、 $R_{i-1,j-1}$ 、 $R_{i,j-1}$ について寄与率の計算

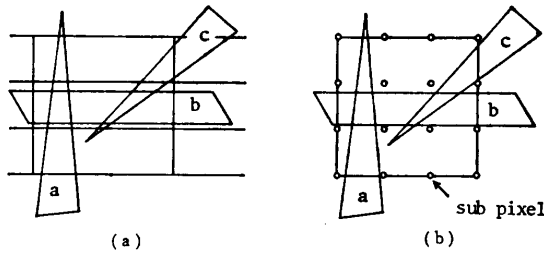


図3 サンプルング誤差
Fig. 3 Sampling error.

終了時に画素 (i, j) の色が決定される。すなわち、 $(2n-1)$ 本のサブスキャンラインにより各画素の色が決定される。

2.3 サンプルング誤差

提案した手法はいずれも水平方向に対するサンプルングは連続であるが、垂直方向に対しては離散的であるため、図3 (a)に示すように、垂直方向に細長い面 (たとえば a および c) に対しては精度よく計算できるが、水平方向に細長い面 (たとえば b) に対してはサブスキャンラインが交差せず、面が検出されない場合が生ずる。一方、従来のサンプルング点を増加させる方法 (B-2.1) では、垂直、水平方向ともに離散的にサンプルングするため、図3 (b)に示すように、垂直あるいは斜めに細長いものについてもサンプルングできない場合が生じる。このように、従来のサンプルング点を増加させる方法に比べて、提案手法は水平方向には連続的にサンプルングすること、およびサンプルングできないケースが減少することにより、全体として精度のよい画像を得ることができる。

3. スムーズエッジ処理を施した三次元物体の陰影表示

三次元物体の陰影表示におけるスムーズエッジ処理は、前述のように使用する隠面消去アルゴリズムと深い係わりをもつ。本章では、2.2 節で述べた方法を拡張し、隠面消去とスムーズエッジ処理を同時に行う方法を提案する。ここで取り扱う対象は、凸多角形で構成されている三次元物体であり、物体上の図柄 (多角形で表現されているもの、たとえば、建築物の窓等) や複数個の光源 (平行光線および点光源) により生じる物体上の影も考慮して処理できるものである。

3.1 概略手順

陰影表示処理全体の手順は、概略次のステップから成る。

- (1) 物体データの入力。

- (2) 視点、注視点および光源の位置の入力、各頂点の透視面への座標変換。

- (3) 面の前後判定、面上の影の抽出。

- (4) 透視面の走査、隠面消去、色計算およびスムーズエッジ処理。

ここでは、(4)の処理についてのみ説明する (他は文献 9) 参照)。なお、(3)の処理を(4)の処理中に行う方法に対しても、提案手法は適用可能である。しかし、各走査線ごとに(3)の処理を行うと判定回数が増加するため、ここでは上述の手順を採用している*。

おのおののサブスキャンラインにおける処理は次のステップから成る。

- (1) 図形 (面、図柄、影) とサブスキャンラインとの交点を求める。

- (2) 交点を座標値の小さい順 (左から順) に並べ替える。

- (3) それぞれのサブスキャンラインについて、可視面を左から右へ順次求め、新しい可視面が現われると、その面のサブスキャンラインとの交点および輝度を用いて、各画素の色を決定する。

ここで、(3)の隠面消去および画素の色計算と、スムーズ処理との関係が重要となる。ここでは、紙面の都合上、処理のやや複雑な 2.2.2 項で述べた軸対称形状のフィルタを用いる方法に、隠面消去処理を付加する方法についておもに説明する。

3.2 隠面消去

図形を走査する際には、図柄、影はそれらが属している面上に密着している多角形と考え、面と同じ扱いをする (同一扱いをする場合、本論文ではこれらを総称してポリゴンと呼ぶことにする)。ただし、同じ面に属している図柄および影については、影、図柄の順で手前に存在するものと考え。したがって、サブスキャンラインとポリゴンとの交点を小さい順に並べ替える際、影、図柄、面の交点が同一座標を占める場合 (たとえば、隣接する面上に影や図柄がある場合) は、それらの座標は、上記ポリゴンの前後関係に準じて、影の右交点、図柄の右交点、次の面の左交点、その面上の図柄の左交点、影の左交点の順に小さいものとする。また、処理中のポリゴンに隠されるポリゴンを記憶しておくスタックメモリ NEXT (図4 (b)参照)

* 本手法は凸多角形を対象としているが、凹多角形を含む図形の場合、面の前後判定が一意に決定できない形状が存在する。その場合は図形を分割するか、(3)の処理を(4)の処理中に行う方法を採用する必要がある。なお、後述の色計算については凹多角形に対しても適用できる。

を準備する。

隠面消去は次の方法で行う。サブスキャンライン上を左から右へと進み、交点通過時に次に可視となるポリゴンを抽出する。その際、その交点の種類により以下の三つの方法により処理する。

(1) 交点がポリゴンの左交点の場合、その交点を含むポリゴンが、現在処理中のポリゴン（すなわち可視ポリゴン）より手前なら、NEXT中に現在処理中のポリゴンを記憶し、その交点を境界とするポリゴンをこれから処理するポリゴンとする（図4(a)中 x_2' の場合）。一方その交点を境界とするポリゴンより現在処理中のポリゴンが手前なら、NEXT中に交点を境界とするポリゴンを記憶し右へ進む。

(2) 交点が現在処理中のポリゴンの右交点の場合、NEXT中で最も手前のポリゴンをこれから処理するポリゴンとする（図4(a)中 x_5' の場合）。

(3) 交点が現在処理中でない他のポリゴンの右交点の場合、その交点を含むポリゴンをNEXT中から消去する（図4(a)中 x_4' の場合）。この場合、現在処理中のポリゴンがそのまま可視であるから、そのま

まの状態を右へ進む。

以上の方法により、左から右に順に可視ポリゴンが抽出される。可視ポリゴンが変化する際、あるいは計算領域 $R_{i,j}$ の境界通過時に次節で述べる色計算を行う。

3.3 画素の色の決定

説明の前に次の準備を与える。

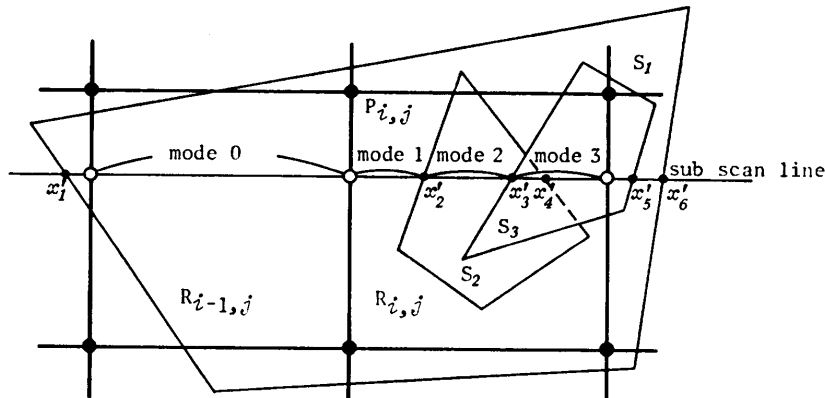
(1) 画素 (i, j) の色を $C_{i,j}$ 、ポリゴン S の色を $I(S)$ とする。 $I(S)$ は S の反射係数と照度により決まる。ここで、2.2.1項で述べたように、 $C_{i,j}$ 、 $I(S)$ 、および反射係数はそれぞれ赤、緑、青の各要素について与えられるものとする。ここで、画素内のサブスキャンラインでの輝度は一定と考え、ある1点（(5)式の x_1, x_2 の中点）で代表させるものとする。また、花柄等の任意のパターンのテクスチャ¹⁰⁾が面上に存在する場合には、そのパターンに応じて反射係数が変化するものとする。

(2) サブスキャンラインとポリゴンとの交点の位置関係により、色計算の区間を次のように分類する。

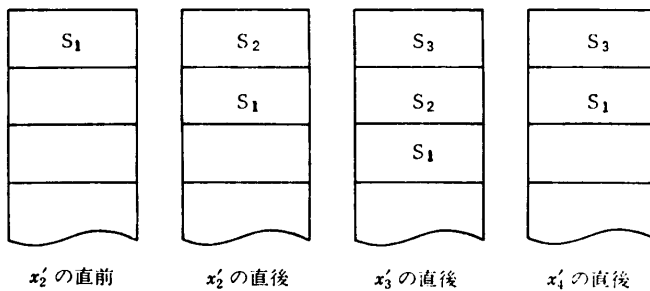
すなわち、計算領域 $R_{i,j}$ 中にポリゴンとの交点がない場合をモード0、 $R_{i,j}$ の左端から最も左の可視交点までの区間をモード1、 $R_{i,j}$ 中の最も左の可視交点から最も右の可視交点までの区間をモード2、最も右の交点から右端までの区間をモード3とする（図4(a)参照）。

画素の色の決定は次のようにする。最初にスクリーン上のすべての画素に対する $C_{i,j}$ ($i=1, 2, \dots, N, j=1, 2, \dots, M; N, M$ は画面サイズ) を零とする。次に、サブスキャンライン上を左から右へと進んで可視面を求め、その際に色計算を行う。すなわち、可視面が変わる点および $R_{i,j}$ の境界通過時に、 $R_{i,j}$ に関係をもつ4個の画素それぞれに対して、(4)式で得られる寄与率と面の色の積をとり、これを順次加算することにより最終的な画素の色を決定する。

$R_{i,j}$ 中に存在するある可視交点 P_k を通過するとき（モード1、モード2のとき）、あるいは



(a) 画素の色の計算



(b) スタックメモリNEXTの遷移

図4 サブスキャンライン上での色計算

Fig. 4 Color calculation on a sub scan line.

$R_{i,j}$ の境界通過時 (モード0, モード3のとき) の色は次式により求まる.

$$\begin{aligned}
 C_{i,j} &\leftarrow C_{i,j} + W_{i,0}(x'_1, x'_2)I(S) \\
 C_{i+1,j} &\leftarrow C_{i+1,j} + W_{i,1}(x'_1, x'_2)I(S) \\
 C_{i+1,j+1} &\leftarrow C_{i+1,j+1} + W_{i,2}(x'_1, x'_2)I(S) \\
 C_{i,j+1} &\leftarrow C_{i,j+1} + W_{i,3}(x'_1, x'_2)I(S)
 \end{aligned} \tag{5}$$

ただし,

$$\begin{aligned}
 x'_1=0, x'_2=1 & \quad (\text{モード0の場合}) \\
 x'_1=0, x'_2=x'_k & \quad (\text{モード1の場合}) \\
 x'_1=x'_s, x'_2=x'_k & \quad (\text{モード2の場合}) \\
 x'_1=x'_k, x'_2=1 & \quad (\text{モード3の場合})
 \end{aligned}$$

ここで, 右辺の $C_{i,j}$, $C_{i+1,j}$, $C_{i+1,j+1}$, $C_{i,j+1}$ はそれぞれ可視交点 P_k を通過する以前に求められている色である. また, x'_k は P_k を $R_{i,j}$ の座標系で表現した座標値, x'_s は一つ左の可視交点の座標値 (必ずしも x'_{k-1} ではない), S は x'_1 と x'_2 の間に存在するポリゴンである.

図4のように, S_1, S_2, S_3 のポリゴンが $R_{i,j}$ に係わる場合を例にとると, $R_{i,j}$ 中のサブスキニングラインでの $C_{i,j}$ に加算される色は次式で示される.

$$\begin{aligned}
 C'_{i,j} &= F_i(x'_2)I(S_1) + (F_i(x'_3) - F_i(x'_2))I(S_2) \\
 &\quad + (F_i(1) - F_i(x'_3))I(S_3)
 \end{aligned} \tag{6}$$

また, $R_{i,j}$ 中を左から右に処理する際のスタックメモリ NEXT の変化を図4(b)に示す.

なお, 提案する手法は一走査線ごと (n 本のサブスキニングラインが終了すること) に CRT に色情報を出力する方法であるから, 実際には $C_{i,j}$ に関して全面面分のメモリは必要ではなく, $C_{i,j}$ と $C_{i,j+1}$ に相当する上下2行分のメモリを準備するだけで十分である.

4. 画質の検討および適用例

図5に画質を検討するためのパターンの表示例を示す. (a)はスムーズ処理をまったくしない場合で, (b), (c)は提案手法を適用したもので, (b)はフーリエウィンドを, (c)はガウシアンフィルタを用いた場合である ((b), (c)ともに画素分割数 n は3). (d)は従来のサンプリング点を増加させる方法によるフィルタリング (ガウシアンフィルタで, 画素内に 3×3 点のサンプル点を設け, 隣接画素も含め 5×5 点で一

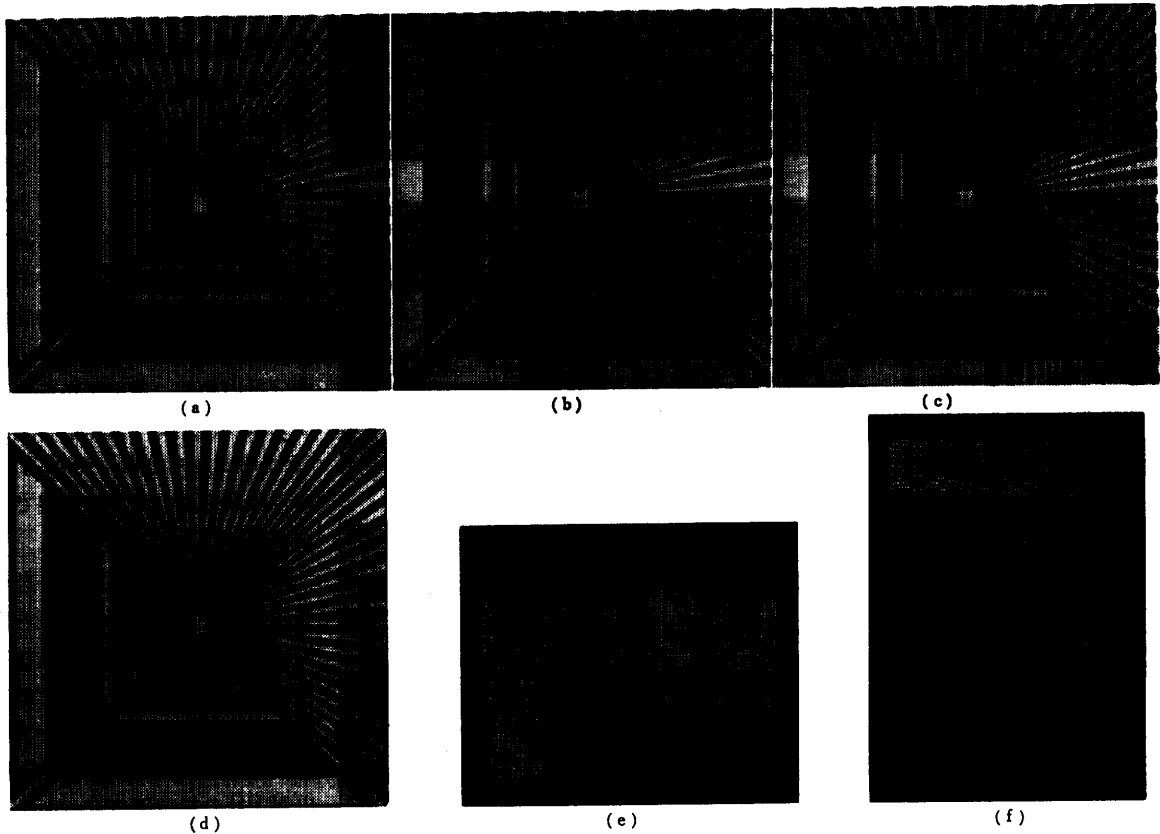


図5 手法の比較
Fig. 5 Comparison of methods.

画素の色を決定) の場合である。以後, (a), (b), (c), (d) に対する方法を方法 A, B, C, D とする。これらの例からわかるように, 提案手法は, 図形の境界部のぎざぎざ, モアレパターンおよび細い図形の表示の改善に有効であることがわかる。(e), (f) は手法の比較をより明確にするためのものである。(e) では左から方法 A, B, D, C の順に示してある。また, (f) では上から順に方法 A, D, C($n=3$), C($n=6$), C($n=12$) が示されている。なお, (f) において, 方法 B は D とまったく同じ画質となったので省略した。(c), (d), (e) 図から明らかなように提案手法のほうが, 従来よく用いられるサンプリング点を増加させる方法より優れた画質を与えることがわかる。ただし, 水平に近い細い図形の場合は両者の優劣差はほとんどない。画素分割数 n に関しては, (b), (c),

(e) 図でわかるように通常 $n=3$ でほぼ十分であるが, (f) のように細い図形が水平方向に並ぶ場合には n を 6 以上とすることが望ましい。

図 6 に適用例を示す。(a), (c), (e) はスムーズ処理を施さない場合であり, (b), (d), (f) はフーリエウィンドを用いた提案手法 ($n=3$) を適用したものである。(a), (b) は古代出雲大社の想像図である。(c), (d) は本例を計算した計算機室である。(e), (f) は日本間を表示したもので, ふすまの花柄, 掛軸の文字はテクスチャ処理を行っている。

これらの適用例は, 提案手法によりかなり画質が改善されることを示しており, 512×512 の画素数の CRT を用いて十分実用に堪え得る画像が得られることを示している。

5. あとがき

三次元物体のカラー陰影表示におけるスムーズエッジ処理について論じた。

提案手法は, 水平に近い傾きをもった図形の境界に対しては若干画質の改善効果は少ないが, 適用例からわかるように画素分割数を 3 以上にすればかなり画質のよい画像を得ることができる。フーリエウィンドとガウシアンフィルタの画質に及ぼす差はわずかであるから, 処理の簡単さ, あるいは処理時間が優先する場合は, 前者の方法で十分といえる。適用例からも明らかなように, 物体上の図形 (テクスチャも含む) や影も含めて画像を高い画質で表現できるようになった。

謝辞 本研究を進めるに当たり, プログラムの製作などで協力いただいた広島大学福田直樹 (現シャープ), 大野逸郎 (現住友重機) 両君および福山大学計算機室藤井克典氏, および古代出雲大社の想像図のデータをご提供いただいた広島大学鈴木充教授に感謝します。

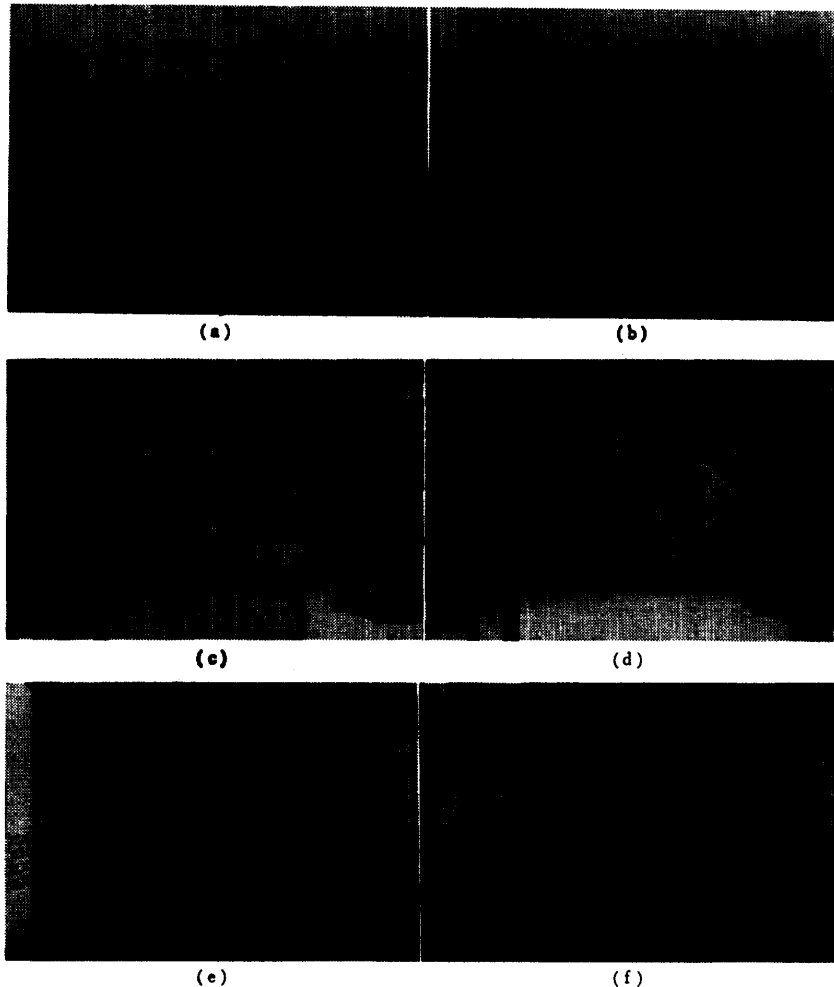


図 6 適用例
Fig. 6 Examples.

参 考 文 献

- 1) Romanowski, W.: Jag Free Images on a Raster CRT's, Inter Graphics '83, Tokyo, B-1 (1983).
- 2) Crow, F.C.: A Comparison of Anti-Aliasing Techniques, *IEEE Comput. Gr. Appl.*, Vol. 1, No. 1, pp. 40-48 (1981).
- 3) Crow, F.C.: The Aliasing Problem in Computer-Generated Shaded Images, *Comm. ACM*, Vol. 20, No. 11, pp. 779-805 (1977).
- 4) Catmull, E.: A Hidden-Surface Algorithm with Anti-Aliasing, *Comput. Gr.*, Vol. 12, No. 3, pp. 6-11 (1978).
- 5) Barros, J. and Fuchs, H.: Generating Smooth 2-D Monocolor Line Drawings on Video Displays, *Comput. Gr.*, Vol. 13, No. 3 pp. 260-267 (1979).
- 6) Feibush, E. A., Levoy, M. and Cook, R. L.: Synthetic Texturing Using Digital Filters, *Comput. Gr.*, Vol. 14, No. 3, pp. 294-301 (1980).
- 7) 柴本: モーショングラフィックスにおけるエリアスの除去, 情報処理学会グラフィックスとCAD研究会, 9-1 (1983).
- 8) Whitted, T.: An Improved Illumination Model for Shaded Display, *Comm. ACM*, Vol. 23, No. 6, pp. 343-349 (1980).
- 9) 西田, 中前: 三次元物体の陰影表示の一手法, 情報処理, Vol. 15, No. 1, pp. 18-25 (1974).
- 10) 西田, 中前: 三次元物体のカラー陰影表示の一手法, 情報処理学会コンピュータビジョン研究会, 4-1 (1980).

(昭和58年9月19日受付)

(昭和59年2月14日採録)