

アンチ・エリアシングのための直交スキャンライン法†

田中 敏光** 高橋 時市郎**

エリアシングは画質を劣化させる大きな要因であり、その除去はコンピュータグラフィックスの重要な課題である。エリアシングは、画素内に投影される物体の面積を精密に求めることで減少できる。本論文では、画面を画素の水平境界で走査した後で2本の隣合う水平走査線の間を1画素の長さだけ垂直方向に走査する直交スキャンライン法を提案する。直交スキャンライン法は1画素内のポリゴンの面積を正確に求めることができる。このため、高度のアンチ・エリアシングが可能である。画像生成実験により、エッジの方向に依存しないアンチ・エリアシングが可能であること、どのように細いポリゴンであっても適正に表示できることが示された。さらに、画像生成速度を比較することにより、1画素あたり4～7本のサブ・スキャンラインを走査するのと等価な時間で、より高精細にアンチ・エリアシングされた画像を生成できることが示された。

1. ま え が き

エリアシング (aliasing) は CG 画像の品質を著しく低下させる要因の1つである。このため、数多くのエリアシング削減 (アンチ・エリアシング) 手法が研究されてきた^{1)~6)}。

エリアシングが発生するのはポリゴンのエッジが画素を横切る場合である。これらの画素では、背景も含め、1つの画素に複数の物体が投影される。したがって、画素の輝度値をただ1つのポリゴンから決めたのでは、正確な値にはならない。各物体が画素内に占める面積を求め、その大きさに比例して物体の輝度値を混ぜ合わせれば、エリアシングを除去することができる^{1), 4)}。このためには、1画素内に投影されるポリゴンの面積を、精度良く求める必要がある。

代表的な隠れ面消去手法のうち、Zバッファ法とレイ・トレーシング法はポイントサンプリングを基本とする。このため、1画素内に投影されるポリゴンの面積を高い精度で求めるには、1画素あたりきわめて多くのサンプル点が必要となる。処理量はサンプル点の数に比例するため、膨大な計算時間を要する。

スキャンライン法は台形近似により1画素内のポリゴンの面積を近似的に求めることができる。さらに、スキャンライン間に複数本のサブ・スキャンラインを挿入することで面積の計算精度を向上させるマルチスキャンニング法²⁾がある。しかし、この手法でも、近似精度を高めるにはサブ・スキャンライン数を増やさなければならない。その結果、計算量が増加する。

これらの手法の問題点は、ポリゴンの面積が近似的

にし求められない点にある。これを解決するため、直交スキャンライン法を提案する。直交スキャンライン法は、画素の境界で画像を水平方向に走査した後、ポリゴンの頂点やポリゴンエッジ同士の交点を通るように垂直方向に走査する。垂直走査線は隣接する水平走査線の間だけで走査される。これにより、すべてのポリゴンが正しく台形分割されるので、計算精度の範囲でポリゴンの面積を正確に求めることができる。

本論文では、スキャンライン法での代表的なアンチ・エリアシング手法とその問題点、直交スキャンライン法の概要、ポリゴンが交差する場合のアンチ・エリアシングについて述べる。また、計算機実験により、本手法の有効性を示す。

2. スキャンライン法におけるアンチ・エリアシング手法

これまでに提案されたアンチ・エリアシング手法を概説し、問題点を指摘する。ポリゴンが1画素内に占める面積を求める方法は、次の2つに大別される。

(1) クリッピングにより1画素内のポリゴンの面積を正確に求める手法^{3), 5), 6)}。

(2) 隣接する2本のスキャンライン間で、ポリゴンの領域を台形近似で求める手法。

前者は、各画素ごとに、視点に近い順にポリゴンをクリッピングして画素内に占める面積を求めなければならないので、計算時間がかかる。高速化の手法も考案されてはいるが、処理が複雑になる割には計算時間はそれほど大幅に短縮されない。

一方、後者は、エッジが他のエッジと交差しないときには面積を正しく求めることができる。しかし、図1(a)のように、スキャンラインの間にエッジの端点

† Cross Scanline Algorithm for Anti-aliasing by TOSHIMITSU TANAKA and TOKIICHIRO TAKAHASHI (Autonomous Robot Systems Laboratory, NTT Human Interface Laboratories).

** NTT ヒューマンインタフェース研究所知能ロボット研究部

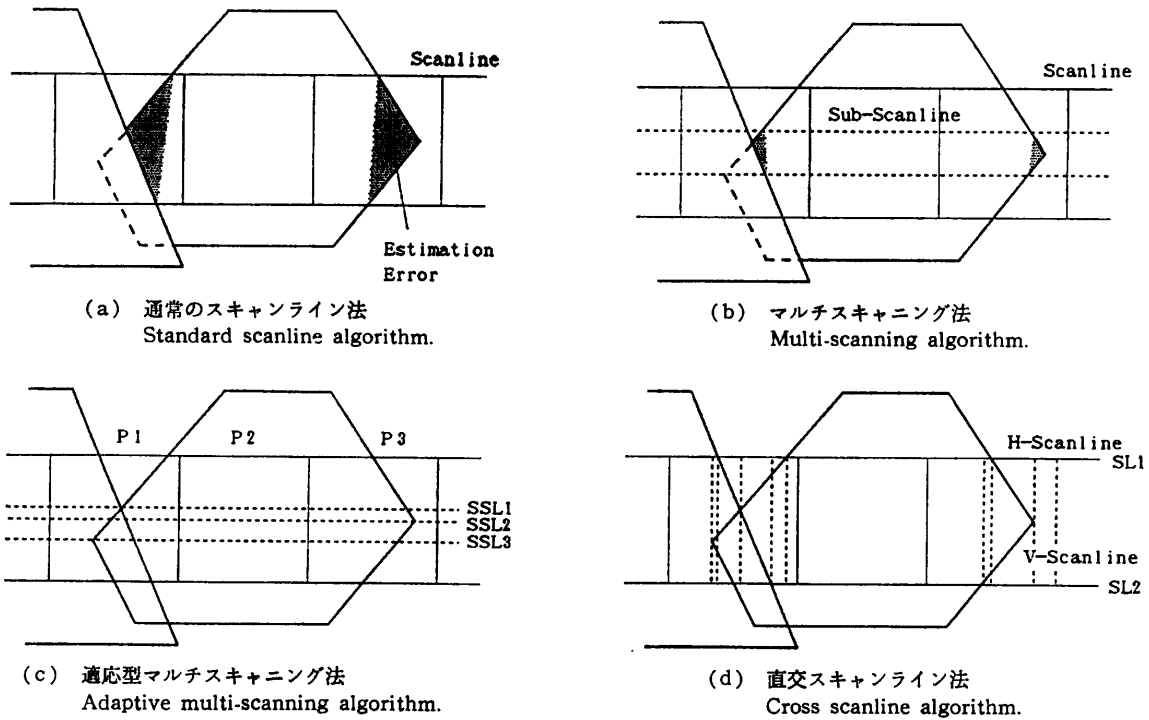


図 1 スキャンラインアルゴリズムとその面積近似誤差
Fig. 1 Scanline algorithms and their estimation errors.

が存在する場合やエッジが互いに交差する場合には台形近似による誤差が大きくなるため、エリアシングが生じる。

2.1 マルチスキャンニング法

台形近似法では、スキャンラインと垂直方向のサンプリング間隔を細かくしないと、エリアシングを目立たなくするために必要な近似精度は得られない。そこで、元々のスキャンラインの間に仮想的なサブ・スキャンラインを挿入し、1画素を複数の場所で走査するマルチスキャンニング法²⁾が提案されている。例えば、図1(b)に示すように、1スキャンラインあたり2本のサブ・スキャンラインを挿入する。これに、画素境界に配置される元々のスキャンライン2本を加え、4本のスキャンラインとポリゴンとの交線の長さからポリゴンの面積を台形近似で求める。その結果、全体として3倍、精度が向上する。マルチスキャンニング法をフィルタ処理と組み合わせ、より効果的なアンチ・エリアシングを行う手法も提案されている²⁾。元のアルゴリズムを大きく変更することなく面積の近似精度をあげられる点で、マルチスキャンニング法は優れた手法である。

しかし、実際の画像生成においては、以下の3点が問題となる。

(1) 図1(b)に示すような近似誤差が依然として残る。たとえ面積の近似誤差が5%であっても、256階調の輝度分解能を持つ標準的なグラフィックディスプレイでは最大12階調の差となり、エリアシングの原因となる。

(2) 水平・垂直方向で分解能が異なる。スキャンライン上の分解能は位置計算の精度で決まるが、垂直方向の分解能はサブ・スキャンライン数で決まってしまう。このため、水平分解能は垂直分解能に比べはるかに小さい。したがって、水平・垂直方向のエッジでジャギの度合いが異なることになる。

(3) 計算時間がサブ・スキャンライン数に比例して増大する。このため、効率よくアンチ・エリアシングを行うには、挿入するサブ・スキャンラインの本数を必要最小限にとどめなければならない。しかし、エリアシングは画面の複雑さやエッジの方向により異なるため、あらかじめこの本数を求めることは容易ではない。

2.2 適応型マルチスキャンニング法

しかしながら、次のようにマルチスキャンニング法を改良すれば、1画素内にポリゴンが占有する面積を正しく求めることができる。

図1(c)のようにポリゴンの頂点やエッジ同士の交

点を通るようにサブ・スキャンラインを配置し、ポリゴンをサブ・スキャンラインで分割すれば、台形法により、面積を正しく求めることができる。この方法を適応型マルチスキャンニング法と呼ぶことにする。この方法は、図形が複雑になるとサブ・スキャンライン数が多くなり、計算量が増大するのが弱点である。

3. 直交スキャンライン法

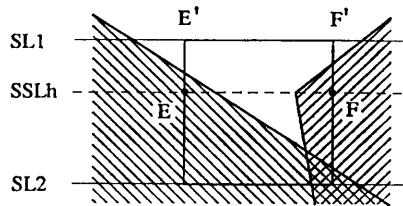
2章の考察に基づき、適応型マルチスキャンニング法を基本にして、計算量の増加を抑えて1画素内に占めるポリゴンの面積を正確に求める手法を提案する。

3.1 基本的なアイデア

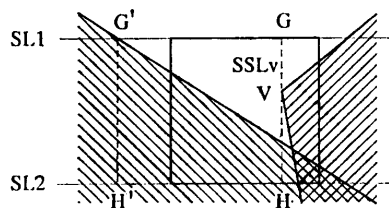
適応型マルチスキャンニング法の問題点は、サブ・スキャンラインによるポリゴンの分割が、本来分割する必要のない画素でも発生することである。たとえば、図1(c)の中で最も上にあるサブ・スキャンラインSSL1による分割は、画素P1内では必要であるが、画素P2, P3内では不要である。そこで、ポリゴンの頂点やエッジ同士の交点を含む画素内だけ、サブ・スキャンラインを走査するように改良すれば、不要な処理の増加を抑えることができる。

1画素の内だけでサブ・スキャンラインを走査するには、

- 水平走査：図2(a)に示すように、水平サブ・スキャンラインSSLhを点Eから点Fまでの間だけ走査する、
- 垂直走査：図2(b)のように、点Gから点Hまで



(a) 水平走査
Horizontal scanning.



(b) 垂直走査
Vertical scanning.

図2 1画素のみのサブ・スキャンライン走査
Fig. 2 Limited sub-scanlines in one pixel.

垂直方向にサブ・スキャンラインSSLvを走査する、

の2つの方法が考えられる。このように画面の途中からスキャンラインを走査するには、

- (1) 走査を開始する点に投影されるすべてのポリゴン（いちばん手前のポリゴンだけではなく、覆い隠されているポリゴンも含む）。

(2) 走査する区間と交差するポリゴンのエッジを求める必要がある。ここでは、垂直走査の方が水平走査よりも手間が少なく、優れていることを説明する。

まず、(1)を求める手間を比較する。

垂直走査では、点Gに投影されるポリゴンが必要であるが、これはSL1の走査結果から簡単に求められる。つまり、通常のスキャンライン走査では、SL1を走査するとき、これと交差するエッジを求め、交点のx座標値でそれらのエッジをソートしたリスト（エッジリスト）を作る。次に、エッジリストから2本のエッジに挟まれた走査線上に投影されるポリゴンのリストを求め、それらの前後判定を行う。したがって、このポリゴンのリストを蓄えておけば良い。

一方、水平走査では点Eに投影されるポリゴンが必要である。通常のスキャンライン走査では、画面の左端から順次ポリゴンを追加・削除するため、平均すると画面の半分の長さだけ処理しなければならない。したがって走査範囲を限定してもそれほど大幅な計算量の削減は望めない。また、SL1上の点E'に投影されるポリゴンから、Eに投影されるポリゴンを求めることもできるが、そのためには、まず、点E'に投影されるポリゴンを求めなければならない。この手間は、点Gに投影されるポリゴンを求める手間と同じである。それに加えて、線分E'Eを走査してこれと交差するエッジを見つける処理が必要となる。したがって、水平走査のほうが垂直走査より処理量が大きくなる。

次に(2)を求める手間を比較する。

垂直走査では、GHと交差するポリゴンのエッジはたやすく求められる。これは、①直前の垂直スキャンラインG'H'と交差するエッジは既に求まっている、また、②G'とGがSL1上、H'とHがSL2上の点であるため、G'GおよびH'Hと交差するエッジはSL1, SL2のエッジリストから簡単に選び出すことができる、からである。すなわち、SL1, SL2それぞれを走査する過程で、交差するエッジを交点のx座標値

の小さい順に並べたエッジリストが既に作られている。これらのリストを垂直スキャンラインの走査位置で分割するだけで、すべての垂直スキャンラインについて、必要なエッジが求められる。

一方、水平走査では、同様の手法で EF と交差するエッジを直前の結果である E'F' 上のエッジから求めるためには、E'E, F'F と交差するエッジがあらかじめわかっている必要がある。しかし、E'E, F'F が SL 1, SL 2 上にないので、すべてのエッジの中から交差するエッジを見つける処理が新たに必要となる。E'E, F'F 上のエッジは垂直スキャンライン上のエッジと同様の手法で求めることができるが、これは、E'E, F'F の 2 か所で垂直走査をしてから、さらに EF を水平走査をすることにはかならない。

以上の点から、垂直に 1 画素スキャンライン走査の方が水平に走査するよりも効率がよい。

3.2 直交スキャンライン法

3.1 節の考察の結果から、縦方向にサブ・スキャンラインを配置する手法を採用する。縦横、直交する向きにスキャンライン走査することから、この手法を直交スキャンライン法 (Cross Scanline Algorithm) と名付ける。混同を避けるため、画素の水平境界に配置された従来どおりのスキャンラインを H スキャンライン、2 本の H スキャンラインの間をこれと垂直に 1 画素の高さだけ走査するサブ・スキャンラインを V スキャンラインと呼ぶことにする。

直交スキャンライン法では、各画素内でのポリゴンの面積計算を簡単にするため、ポリゴンの頂点とエッジの交点に加え、上下の H スキャンラインとエッジとの交点でも V スキャンラインを走査する。この結果、図 1 (d) に示すように、V スキャンラインはポリゴンを台形または三角形に分割する。このため、画素ごとのポリゴンの面積は比例分配によりたやすく求められる。

3.3 処理の流れ

直交スキャンライン法の処理の流れを以下に示す。

[開始]

- (1-1) 透視変換、裏面の削除、画面でのクリッピング、等の前処理を行う。
- (1-2) ポリゴンのエッジを端点の y 座標でソートしたリスト (HE リスト) を作成する。
- (1-3) H スキャンラインを画面下端に設定する。

[Hループ]

- (2-1) 通常のスキャンライン法と同様の手法で、H スキャンラインとポリゴンとの交差を求め、視点か

ら見えるポリゴンとそのポリゴンが H スキャンライン上に占める領域を決定する。

- (2-2) (2-1) の処理の過程で用いられる、H スキャンラインと交差するエッジをその交点の x 座標でソートしたリスト (HAE リスト) を保存する。

- (2-3) H スキャンラインが画面下端なら、[H スキャンラインの更新] に進む。

- (2-4) V スキャンラインの走査位置を決定する。

- (1) 現在の HAE リスト上の交点
- (2) 1 本前の HAE リスト上の交点
- (3) 現在および直前の H スキャンラインの間に存在するポリゴンの頂点
- (4) 画面の左右端

を通る V スキャンラインが走査される。

- (2-5) V スキャンラインを画面左端に設定する。

[Vループ]

- (3-1) 通常のスキャンライン法と同様の手法で垂直方向に走査し、V スキャンラインとポリゴンとの交差を求める。

- (3-2) 2 本の V スキャンライン間で画面上に投影されたエッジが互いに交差すれば、その交点を通るように新たに V スキャンラインを生成する。

- (3-3) 各画素のポリゴンの面積を台形法で求める。
- (3-4) V スキャンラインが画面右端でないなら、V スキャンラインの位置を更新し、[Vループ] に戻る。

[H スキャンラインの更新]

- (4-1) H スキャンラインが画面上端でないなら、H スキャンラインの位置を 1 つ上の画素境界に移動して、[Hループ] に進む。

[終了]

H スキャンラインは画素境界に配置されるため、等間隔である。V スキャンラインは (2-4) の条件を満たす場所に配置されるため、不定間隔となる。また、

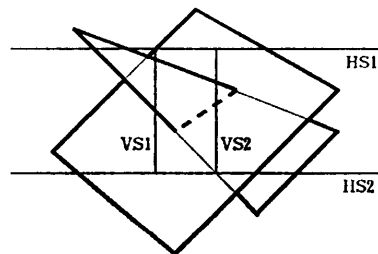
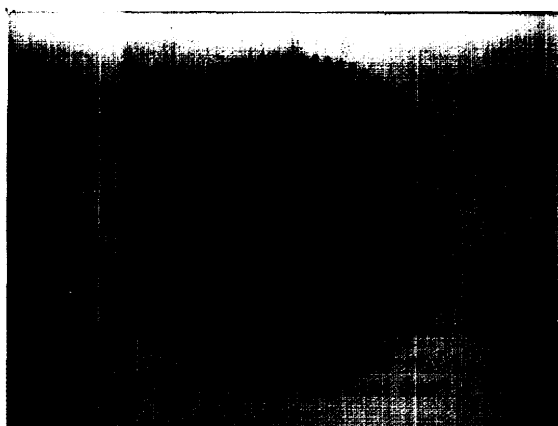
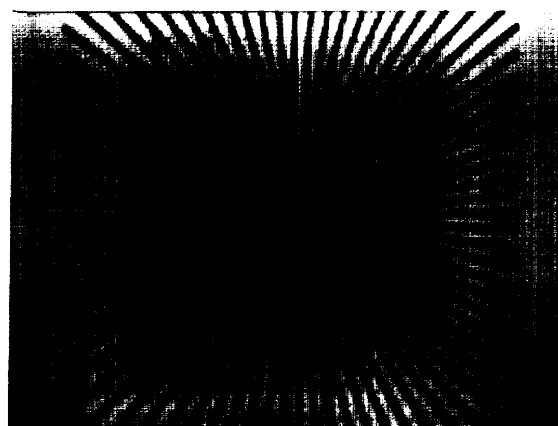


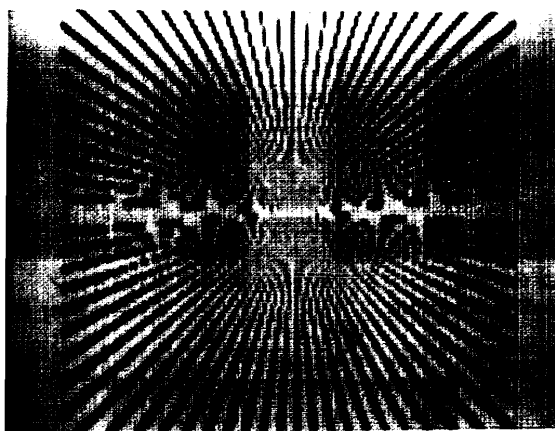
図 3 ポリゴン交線の作る仮想エッジ
Fig. 3 Temporary edge generated by a polygon intersection.



(a) 直交スキャンライン法
Cross scanline algorithm.



(b) (a)の中央部を4倍に拡大
4 times enlarged image of (a).



(c) 通常のスキャンライン法
Standard scanline algorithm.

図 4 放射状パターン
Fig. 4 Radial pattern.



(a) 直交スキャンライン法
Cross scanline algorithm.

(b) 通常のスキャンライン法
Standard scanline algorithm.

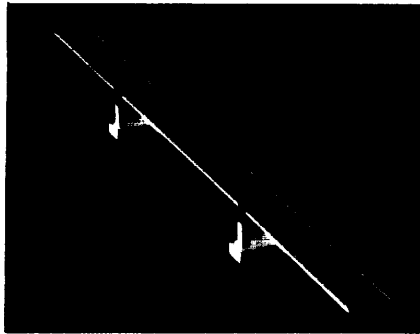
図 5 傾き5度のメッシュパターン
Fig. 5 Mesh pattern rotated 5 degrees.



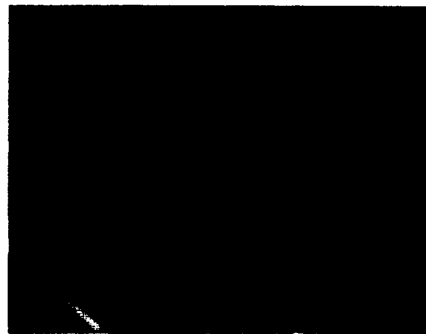
(a) 直交スキャンライン法
Cross scanline algorithm.

(b) 8倍オーバーサンプリング
8 times over scanning.

図 6 XY 軸と平行なメッシュパターン
Fig. 6 Mesh pattern parallel to the axes.



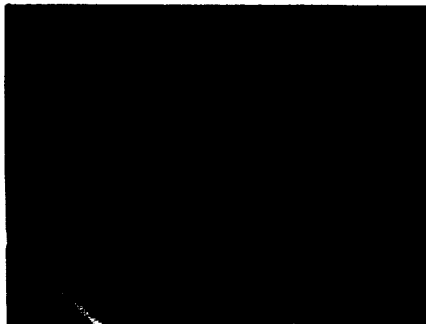
(a) 直交スキャンライン法による全景
Whole view created by the cross scanline algorithm.



(b) (a)の一部を4倍に拡大
4 times enlarged image of (a).



(c) 通常のスキャンライン法
Standard scanline algorithm.



(d) 8倍スキャン
8 times over scanning.

図7 吊り橋

Fig. 7 Suspension bridge.

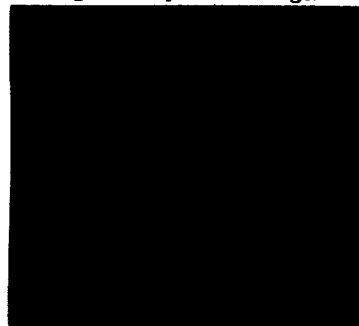


図8 実験画像

Fig. 8 Buildings.

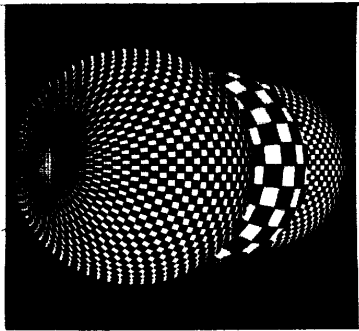


図9 実験画像
Fig. 9 Three balls.

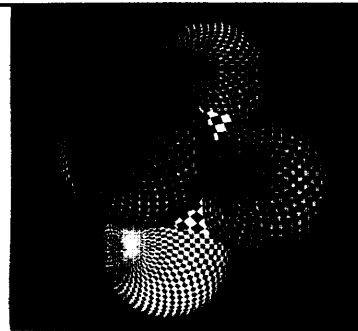


図10 実験画像
Fig. 10 Seven balls.

(3-2)においてVスキャンラインが追加されることがあるので、Vスキャンラインの走査位置は x が増大する方向のみ移動するとは限らない。しかし、これらの点をのぞけば、Vスキャンライン走査も通常のスキャンライン走査と同じである。

4. ポリゴン同士の交線のアンチ・エリアシング

3.2節で示したアルゴリズムは、画素がエッジのみで領域分割されることを前提としている。物体がポリゴンで記述されているときには、ポリゴンが他のポリゴンの一部を隠す場合も含め、この前提条件が成り立つ。しかし、ポリゴンが他のポリゴンを貫通する場合には、ポリゴンの交線で画素が分割されることがあるので、上記の前提条件が成立しない。

ポリゴン同士が交差しないように前処理でポリゴンを分割することは、計算量を増大させるため現実的ではない。そこで、3.2節の(2-1)と(3-1)でポリゴン同士の交差が見つかった場合にのみ、交差するポリゴンを分割し、その交線を仮想的なエッジとしてエッジリストに追加するように、アルゴリズムを改良する。仮想エッジを加えることで、画素は常にエッジで分割されることになる。

図3に示すように、各1組のH、VスキャンラインHS1、HS2、VS1、VS2で区切られる矩形領域を考える。実線で示される真のポリゴンエッジの場合、エッジがある点で領域の境界と交差するならば、そのエッジは他の点でも領域の境界と交差する。しかし、破線で示される仮想エッジでは、図3のようにこの性質が成立しない場合がある。そこで、仮想エッジと矩形との交点が1つしか見つからない場合には、仮想エッジを延長してもう1つの交点を求め、その位置もリストに追加する。このような処理を導入すれば、仮想エッジも通常のエッジと同様に扱うことができる。ただし、仮想エッジはポリゴンの輪郭線ではないので、スキャンラインが横切ってもポリゴンの追加・削除を行わない。

5. 画像生成実験

本手法の有効性を検証するために、種々の画像を生成した。実験には浮動小数点演算機構付きのVAX 8840を用いた。

5.1 放射状パターンの表示

ポリゴンエッジの方向によるジャギ発生のを

調べるために、放射状パターンの隠面消去を行った。

図4(a)は直交スキャンライン法により生成した画像で、図4(b)はその中心部を縦横各4倍に拡大したものである。図4(c)は(b)と同じ部分を通常のスキャンライン法(サブ・スキャンラインなし)で作成したものである。これらの画像生成ではフィルタ処理を行っていない。図からわかるように、通常のスキャンライン走査では水平エッジでジャギが著しく目立つのに対し、直交スキャンライン法では、どの方向のエッジでも目だたないジャギが発生しない。この実験結果は、直交スキャンライン法を用いることでエッジの方向に依存しないアンチ・エリアシングが可能であることを示している。

5.2 細い線分の表示

直交スキャンライン法を用いると、きわめて細い線分も適正に表示できる。この性質を確認するため、細い線分で構成されたメッシュを隠面消去した。線分の幅と間隔は、表示画面上でそれぞれ0.05画素、20画素程度である。

図5はメッシュがX軸から5度傾いた場合である。図5(a)は直交スキャンライン法による結果で、水平・垂直とも線分がきちんと表示される。一方、図5(b)は通常のスキャンライン法による生成画像で、水平方向の線分が切れ切れに表示される。これは、スキャンラインの間隔で決まる垂直分解能が、横線を連続に表示するためには不足しているからである。1, 3, 7, 15本とサブ・スキャンラインの本数を変えて画像生成実験を行った。その結果、水平線分が連続して表示されるには、7本以上のサブ・スキャンラインが必要であった。しかし、この場合でも縦線と横線の輝度が明らかに違っており、適正な表示とはいえない。

次に、メッシュが完全にXY軸と一致した場合を調べた。この場合も、直交スキャンライン法は妥当な画像を生成できる。この結果を図6(a)に示す。図からわかるように、水平・垂直線とも同じ輝度で表示される。図6(b)は7本のサブ・スキャンラインを挿入した結果である。7本もサブ・スキャンラインを挿入しても、メッシュを構成する線分はサンプリング間隔よりも細いため、横線がスキャンラインの谷間に入ってしまう。このため、横線がまったく表示されない場所がある。

横線が表示されても、その輝度は縦線より明るくなる。これは、垂直方向の分解能が1/8画素であるた

め、横線がスキャンラインと交差する場合には $1/8$ 画素の面積を持つと判断されるからである。これに対し、垂直方向では、正しい面積 ($1/20$ 画素) が求まる。この結果、縦線と横線で輝度が異なることになる。これが、マルチスキャンニング法の原理的な限界である。

5.3 詳細な構造物の表示

細い線分を含む構造物の例として、図7に示す吊り橋の画像を生成した。図7(a)は直交スキャンライン法で生成した画像で、(b)はその一部を縦横各4倍に拡大表示したものである。フィルタ処理を行っていないため橋を吊るワイヤが階段状になってはいるが、連続した線として表示されている。図7(c)は(b)と同じ部分を通常のスキャンライン法で生成した画像である。この場合、ワイヤは点列としてしか表示されない。図7(d)は8倍サンプリングした画像であるが、依然としてワイヤが点列として表示される。このように、直交スキャンライン法は細いポリゴンを含む物体の表示に効果的である。

図8~10も直交スキャンライン法で生成した画像である。これらの例から、直交スキャンライン法により高品質の画像が生成できることがわかる。

5.4 画像生成時間の比較

直交スキャンライン法とマルチスキャンニング法の画像生成時間を比較する。生成時間を比較して、直交スキャンライン法と同じ計算時間で走査できるサブ・スキャンライン数 (N_{ET} : 等価サブ・スキャンライン数) を実験的に求める。

実験に用いるデータは5.1節で用いた放射状パターン(図4)とBuildings(図8), Three balls(図9), Seven balls(図10)である。各画像に含まれるポリゴンの数を表1に示す。生成する画像の大きさは 512×512 画素, $0 \sim 255$ 階調である。

4種のデータについて、直交スキャンライン法と、サブ・スキャンライン数を0から10まで変えたときのマルチスキャンニング法で画像を生成し、計算時間を測定した。図11の斜め線がマルチスキャンニング法による生成時間を、水平線が直交スキャンライン法による生成時間を示す。これらの交点から定まる等価サブ・スキャンライン数 N_{ET} を表1に示す。この結果から、直交スキャンライン法は4~7本のサブ・スキャンラインを挿入するのと同程度の計算時間で高品質の画像を生成できることがわかる。

マルチスキャンニング法では、サブ・スキャンライ

表1 形状データ
Table 1 Object data sets.

Data set name	Number of polygons	Figure	N_{ET}
Radial pattern	80	Fig. 4	4
Buildings	151	Fig. 8	4
Three balls	10,450	Fig. 9	6
Seven balls	16,150	Fig. 10	7

N_{ET} : Number of sub-scanlines that yields equivalent computing time to the cross scanline algorithm.

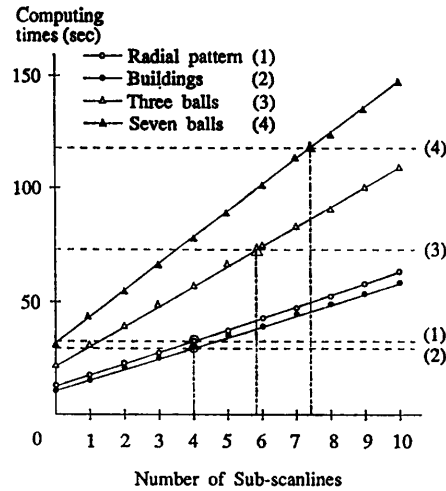


図11 画像生成時間
Fig. 11 Computing times.

ン数に比例して品質が向上するが、同時に計算時間も増大する。直交スキャンライン法では通常のスキャンライン法の数倍の時間がかかるが、厳密なエリアシング除去を行った画像が得られる。多少のエリアシングを問題としないならばマルチスキャンニング法が速い場合もある。しかし、何本のサブ・スキャンラインを挿入すれば要求される画像品質を満たすかは、物体の複雑さや視点の位置により異なる。このため、あらかじめ最適なサブ・スキャンライン数を決めることができず、試行錯誤を伴う。サブ・スキャンラインの本数が不足した場合、生成した画像は全く無駄になり、再度画像生成しなければならない。直交スキャンライン法では常に最良の画質が得られ、上記のような無駄や判断のわずらわしさは生じない。この点でも直交スキャンライン法が優れている。

6. むすび

HスキャンラインとVスキャンラインを用い、画像を縦横2方向に走査する直交スキャンライン法を提案した。直交スキャンライン法は、計算精度の範囲で1

画素内に投影されるポリゴンの面積を正確に求めることができるので、精密なアンチ・エリアシングが可能である。実験により、直交スキャンライン法がエッジの方向によらずエリアシングを除去できること、また、極めて細い線分をも適正に表示できることを示した。

マルチスキャンニング法によりサブ・スキャンラインをスキャンライン間に挿入することでエリアシングを削減できるが、計算量はサブ・スキャンライン数に比例して増大する。また、画像に応じた適切なサブ・スキャンライン数を試行錯誤で選ばなければならない。直交スキャンライン法は正確にポリゴンの面積を求めることができるので、本手法を用いればサブ・スキャンライン数を試行錯誤で選ぶ必要はない。実験の結果、直交スキャンライン法はサブ・スキャンライン数本を走査するのと同じ計算時間で、より高品質な画像を生成できることが示された。

文献3)のような軸対称フィルタと組み合わせることで、いっそう効果的なアンチ・エリアシングができる。マッピングされるテクスチャのエリアシングも除去することができる。また、本手法は極めて細いポリゴンも適正に表示できるので、エッジや頂点の作るハイライトを表示するのにも適している。これらについては稿を改めて報告したい。

謝辞 日頃ご指導賜る、高野陸男知能ロボット研究部長、滝川啓主幹研究員に感謝いたします。熱心にご討論いただいた研究グループの皆様に感謝いたします。本研究を進める上で貴重なご意見・ご教授をいただいた福山大学西田友是教授に深謝いたします。

参 考 文 献

- 1) 西田, 中前: カラーディスプレイにおけるスムーズな線分の発生方法, 情報処理学会論文誌, Vol. 22, No. 6, pp. 505-511 (1981).
- 2) 西田, 中前: マルチスキャンニング法によるス

ムズエッジ処理を施した三次元物体の陰影表示, 情報処理学会論文誌, Vol. 25, No. 5, pp. 703-711 (1984).

- 3) Feibush, E. A. et al.: Synthetic Texture Using Digital Filters, *Comput. Gr.*, Vol. 14, No. 3, pp. 294-301 (1980).
- 4) Crow, F. C.: The Use of Grayscale for Improved Raster Display of Vectors and Characters, *Comput. Gr.*, Vol. 12, No. 3, pp. 1-5 (1978).
- 5) Catmull, E.: A Hidden-Surface Algorithm with Anti-aliasing, *Comput. Gr.*, Vol. 12, No. 3, pp. 6-11 (1978).
- 6) Barros, J. and Funch, H.: Generating Smooth 2-D Monocolor Line Drawings on Video Display, *Comput. Gr.*, Vol. 13, No. 3, pp. 260-269 (1979).

(平成2年5月7日受付)

(平成2年11月13日採録)



田中 敏光 (正会員)

昭和57年名古屋大学工学部電気工学科卒業。昭和59年同大学院修士課程情報工学専攻修了。同年日本電信電話公社入社。以来、3次元形状記述手法、レンダリング手法等、コンピュータグラフィックスの研究に従事。現在 NTT ヒューマンインタフェース研究所知能ロボット研究部研究主任。電子情報通信学会会員。



高橋時市郎

1954年生。1977年新潟大学工学部電子工学科卒業。同年 NTT 電気通信研究所入所。以来、文字認識、画像生成の研究に従事。現在、ヒューマンインタフェース研究所知能ロボット研究部主幹研究員。電子情報通信学会、IEEE, AVIRG 各会員。