

## 広域投影空間を持つシャドウ・アンチエイリアス

山口 直<sup>†1</sup> 池戸 恒雄<sup>†2</sup>

実時間レンダリング技術において、影のエイリアスとシャドウバッファ容量に制約された影の有効投影領域の問題は、今日においても現実感と実時間性が求められるシステムにおいては重要な研究課題である。本論文では影のシルエットラインの実時間アンチエイリアス技術として、シャドウポリゴンのシルエットライン検出とライン情報の保存、複数のシルエットラインが交差する画素内のシャドウポリゴンの内外判定、またシャドウバッファ空間を仮想領域化して投影面範囲を論理的に拡大するそれぞれの新しい手段を提案する。これら手段によって、本研究では、わずかなバッファ容量において 64~65,536 倍相当の空間におけるアンチエイリアス画質を得ることを、ハードウェアアルゴリズムのシミュレーションとともに確認し、またハードウェア回路化に関する考察も行った。

### Shadow Antialiasing with Virtual Projection Space

SUNAO YAMAGUCHI<sup>†1</sup> and TSUNEO IKEDO<sup>†2</sup>

The alias and narrow projection area for shadow-casting due to the physical restriction of buffer size are some of the most important issues for realistic and real-time rendering. This paper proposes an antialias technique on shadows in complex geometry involving detection of silhouette-line in shadow objects, store of multiple silhouette-line information per pixel in specific data formats at first pass, and restore of the silhouette-lines in order to determine a rendered point whether to inner or outer of shadow polygons at second pass. Formatting the silhouette-line information by mathematical expression and packaging these in specific buffer structures, a small-sized physical buffer enables the high quality shadow-casting in available ranges of 64-65,536 times wider than the ordinary 2-dimensional shadow buffer. We verified the availability of this algorithm with Java programming and investigated a hardware implementation for real-time rendering.

### 1. はじめに

コンピュータグラフィックスにおける影のレンダリング技術は、物体間の幾何学的関係を明示するうえで不可避であり、これまで数多くの研究が行われてきた。特に実時間性が必要な実用システムでは、シャドウマップ<sup>1)</sup> やステンシルシャドウボリューム法<sup>2)</sup> が知られている。しかし実時間レンダリングでは 2 次元配列画像メモリの 1 種であるシャドウバッファを用いることから、影のシルエットライン（輪郭線）上に現れるエイリアスと、影の有効投影領域が制限される 2 つの未解決な問題をかかえている。前者はバッファの解像度に依存したエイリアスがあり、また後者では光源と視点とのなす角が直交するに従って影の有効投影領域（視点からの深度）はシャドウバッファの持つ水平画素数（たとえば 1K）にしかない。

エイリアスのないソフトシャドウ技術はラジオシティやフォトンマッピング<sup>3)</sup> 等の大域照明アルゴリズムで解決されるが、これらは実時間システムには適していない。この問題に対して Sen ら<sup>4)</sup> はエイリアスを軽減する手段として、影のシルエットライン座標をバッファに保存し、視点座標系での描画時に、この保存値を用いてシルエットラインを再生し影の有無を判定するシルエットマップ方式を提案した。しかし画素内に複数のシルエットラインが含まれる場合や、複数のオブジェクトが隣接する場合の実用上の処理は検討されていない。一方、影の有効投影領域を拡大する技術としてアダプティブシャドウマップ<sup>5)</sup> や、パースペクティブシャドウマップ<sup>6)</sup> が近年提案された。これらは視点距離に対応してバッファ解像度を变化させ投影範囲の拡大には成功したが、シャドウ生成の処理時間や視点変化ごとにシャドウバッファの再描画が必要になるという点で計算負荷の増加がともなう。また遠方物体が視点近くにまで影を落とす場合は、遠方であるがための低解像度によるエイリアスを避けることができない。

前記それぞれの研究に加え、影のアンチエイリアスについて多数の推論的手法<sup>7)-10)</sup> の提案がある。多くはシャドウマップとフィルタリング技術を組み合わせたものであり本質的に前記それぞれの問題のすべてを解決するまでには至っていない。

本論文は以上の背景に鑑み、実時間システムにおいて多数の、また複雑に配置されたオブ

†1 法政大学情報科学研究科  
Graduate School of Computer and Information Sciences, Hosei University

†2 法政大学情報科学部  
Faculty of Computer and Information Sciences, Hosei University

ジェクトが生み出す影のシルエットライン上に現れるエイリアスを軽減するアンチエイリアスの手段と、限られた物理バッファ容量でありながら影の有効投影領域を光源位置に依存することなく従来の数百倍に拡大する新しい手段を、ハードウェア実装に適したアルゴリズムとしてそれぞれ提案するものである。

## 2. シルエットライン・アンチエイリアス

本研究の影生成は、光源を投影中心としてシャドウポリゴン（ここでは光源を遮蔽するポリゴンのデプス値を意味する）をシャドウバッファに記憶する第1パスと、視点を中心にポリゴンをレンダリングしながらこの過程で第1パスの情報を参照して影を生成する第2パスの、2パス法を基本とする。ここで第1パスにおいてシャドウポリゴンだけでなく、シルエットラインとなる画素（ピクセルグリッド：以下グリッドと呼ぶ）上の情報を、グリッド単位で記憶する。これをシャドウポリゴンとともに第2パスで読み出し、グリッドを横断するシルエットラインを再生しながらレンダリング点がラインの内外（シャドウポリゴンの内部・外部）のいずれに属するかで影の判定を行う。以上から本研究ではアンチエイリアスを行うための画像解像度を高める手段として、スーパーサンプリングのように、シャドウバッファの2次元配列画素の増設を図るのではなく、グリッド内におけるシルエットラインの交差情報の再生を用いるものである。従来のシルエットマップ方式<sup>4)</sup>とは以下の4つの点で異なる。(1)シルエットラインバッファは2次元構造ではなく、シルエットラインの通過グリッド数に対応した1次元バッファで構成、(2)ポリゴンどうしの接辺か、シルエットラインかの判別を第1パスのレンダリング過程で行い接辺は削除、(3)複数のシルエットラインが1つのグリッド内を横断・交差する場合、ラインごとに該当する情報をバッファに記憶、(4)シルエットライン情報はライン交差と属性の数値とし精度は拡張可能とする。

以上から本研究は、シルエットラインバッファ（以下SLBという）をシルエットライン上のグリッド数のみで構成すること、SLBを1次元動的配置のプログラマブルな拡張構造とすることでわずかなメモリ容量で複雑なシルエットライン情報を記憶し、またSLB情報を数値データ化して描画物体全体を圧縮描画することを可能とし、影の投影有効領域を拡大した場合にもデータ精度の選択により影の精度を保障するものである。一方、これらの実現には、シルエットライン情報の構成やSLBへのシルエットライン情報の高速な記憶と読み出しによる影判定がそれぞれ重要な課題となる。以下本研究の上記課題に対応した手法を説明する。

## 3. シルエットライン処理

### 3.1 シルエットライン情報

本研究では第1パスでのシルエットライン情報を基に第2パスでその交差状態を再生するために必要な下記の情報をSLBにそれぞれ保存する。また保存形式を図1に示す。

- (1) Fraction：グリッド交差座標（0～1の小数值）
- (2) Slope：直線の傾き（0～1の小数值）
- (3) Polygon ID：ポリゴン（遮光）識別子
- (4) Edge Flag：シルエットライン有無
- (5) End Flag：端点（始点か終点か）
- (6) L/R Flag：ポリゴン左右辺
- (7) Cross Flag：グリッドへの入力交差軸
- (8) Major：直線の長軸（X・Y軸のどちらに傾いているか）
- (9) Xsign：X増分値の正負

SLBは1次元構造とし画素内に含まれるシルエットライン数および「Fraction」と「Slope」の精度は任意に設定する。たとえばそれぞれ3ビットであれば、1つのグリッドは8×8のサブグリッドを持ったことに等しい。「Fraction」、「Slope」、「Cross Flag」、「Major」、「Xsign」の具体的な役割は図2に示す。「Fraction」は「Cross Flag」が0の場合グリッドへの入点の座標値、1の場合グリッドからの出力点の座標値を意味し、「Slope」とともに外郭線補間に使う。「Major」と「Xsign」は直線の種類を意味し、直線の進行方向を決定するために使う。「Polygon ID」は簡単なポリゴン識別子、「L/R Flag」はポリゴンの左右辺（ポリゴ

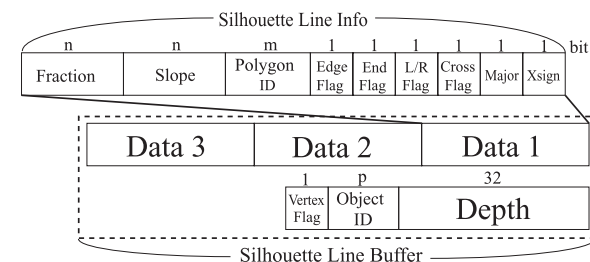


図1 シルエットラインのデータ構造とSLB  
Fig.1 Data structures of silhouette line and SLB.

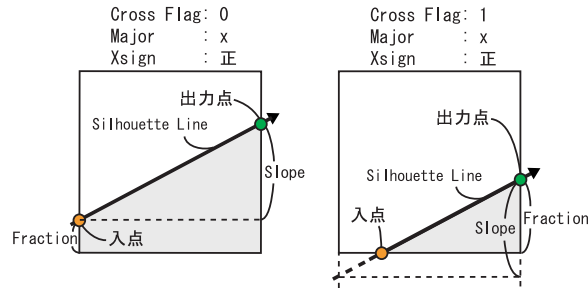


図 2 (左図) 入点が外郭線の短軸となる軸上, (右図) 入点が外郭線の長軸となる軸上にあるグリッド  
 Fig.2 (Left) Grid that input point is on minor axis of outline and (Right) major axis of one.

表 1 バッファの構成  
 Table 1 Buffer configurations.

バッファ名	次元	ビット幅	容量
シャドウバッファ	2	32-bit	1K*1K
IDバッファ	2	8-bit以上	1K*1K
エッジバッファ	2	1bit	1K*1K
1次SLB	1	任意	任意
2次SLB	1	任意	1次の半分程度
3次SLB	1	任意	2次の半分程度
⋮	⋮	⋮	⋮

ンの内部は左右のどちらなのか)を表し,ポリゴン接続線の決定(3.3.2項)で使われる。「End Flag」はグリッド内に頂点があるときのみに有効で,影の内外判定に使う。

また,複雑な交差条件を認識するため,SLBに「デプス値」,「オブジェクトID」,「Vertex Flag(頂点の有無)」も記憶する。以上の(1)から(9)までのシルエットライン単位の情報と,SLB単位の情報を基にシルエットラインを復元し,影の内外判定を行う。

### 3.2 バッファ構成

本研究では以下の4組のバッファで構成する。表1にそれぞれのバッファ特性を示す。

- (1) シャドウバッファ:光源最近傍デプス値
- (2) IDバッファ:オブジェクトID
- (3) エッジバッファ:シルエットラインの有無表示
- (4) シルエットラインバッファ(SLB):前記シルエットライン情報

表1において1組のシャドウバッファとIDバッファは従来から2パス方式で用いられる基

本的なバッファであり,エッジバッファとSLBが追加された構造となっている。エッジバッファはシャドウバッファと同じ2次元配列を持ち1ビット/1グリッドでシルエットラインのあるグリッドを示すフラグである。またSLBは増設型(リンクリスト構造)で可変長とする。表1を見る限り従来方式と比較し多量のバッファを必要とするように思われるが,物理的なバッファサイズはシルエットライン上の情報のみを扱う点で,後述するようにわずかである。

エッジバッファは,第2パスにおいてレンダリング点を視点から光源座標系に変換し影の有無をテストする際,変換点グリッドがシルエットライン上か否かを示すもので,このエッジフラグが立っていればSLBをアクセスし,立っていないれば影の有無を即決する情報である。この判定により,シャドウバッファに記憶される映像の大半のグリッドは,通常シルエットラインではないため,第2パスではSLB読み出しをとまなわない高速処理が可能となる。SLBへのアドレッシングは前記変換座標値を文字列とするハッシュ法等を用いる。

SLBはアドレスあたり3組のシルエットライン情報を保存する(図1参照)。これはポリゴン形状が三角形で与えられることと,メモリ効率から決定したものである。同一オブジェクトIDのシルエットラインが1つのグリッド内に複数交差する場合は,同一アドレスのSLBに保存し(3つ以上はリンク),異なるオブジェクトIDでは異なるアドレスに挿入する。ここでは光源に最も近いシルエットライン情報を持つSLBを1次SLBとし,リンクするSLBを光源から近い順に2次,3次と定義する。

### 3.3 シルエットライン決定手段

#### 3.3.1 特殊な直線生成アルゴリズム

本研究では第1パスのポリゴン外郭線補間段階において,外郭線がグリッドと交差する情報(グリッド辺とその交点等)をSLBに記憶する。そして,これらがシルエットラインとして判定された場合,第2パスの影判定に用いる。このため従来からのBresenham<sup>11)</sup>やAken<sup>12)</sup>の一般的な直線補間アルゴリズムを用いることができない。これらはいずれも直線に最近傍となるグリッド点を決定する方式であり,グリッドが外郭線と交差していない場合であっても近傍のグリッドを選択したり,図3(左)のように外郭線が交差するグリッドであっても選択されない場合が生じたりする。本方式では,外郭線により近いグリッドを選択するのではなく,外郭線が通過するグリッドのみを選択するものであり最近傍法とは異なるアルゴリズムを用いる。よって本方式では左図の場合,右図の緑で示したグリッドも選択されることになる。

3.1節のシルエットライン情報を得ながら直線補間を高速に進めるためには外郭線の現在のグリッドから次のグリッドの交差軸の先読み判定を必要とする。本方式では外郭線はつねに

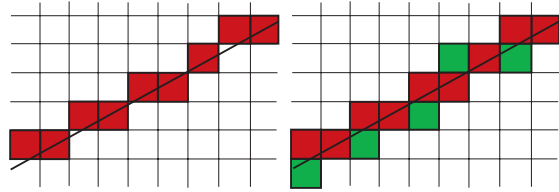


図 3 (左図) Bresenham 補間, (右図) 外郭線補間  
Fig. 3 (Left) Bresenham and (Right) our outline interpolations.

表 2 出力点の分類  
Table 2 Classification of output points.

直線種類		交差可能性点		
Major	Xsign	長軸上	頂点上	短軸上
x	正	4	3	2
	負	4	0	1
y	正	2	3	4
	負	1	0	4

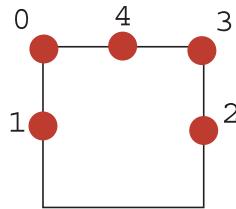


図 4 グリッド内における直線の出力点  
Fig. 4 Output points of crossed line with pixel grid.

y 軸に対して増分方向で補間することで外郭線の交差条件は限定できるため、この交差条件を外郭線の持つ傾き「Major」と x 軸に対する進行方向「Xsign」からなる 4 つの組合せと、これらに対する外郭線とグリッド辺との交差点を 5 点に分類する。表 2 および図 4 にこの関係を示す。表 2 において、たとえば外郭線が x 軸側に傾いており x 値が増加する場合はグリッドとの交差する可能性のある点は 4, 3, 2 となり「Slope」, 「Cross flag」, 「Fraction」情報を用いて次のグリッド交差軸が求まる。交差条件で交差点を分類することで計算量は 3/5 となる。一方、それぞれの出力点 0~4 は次に外郭線が交差するグリッドの入点となり

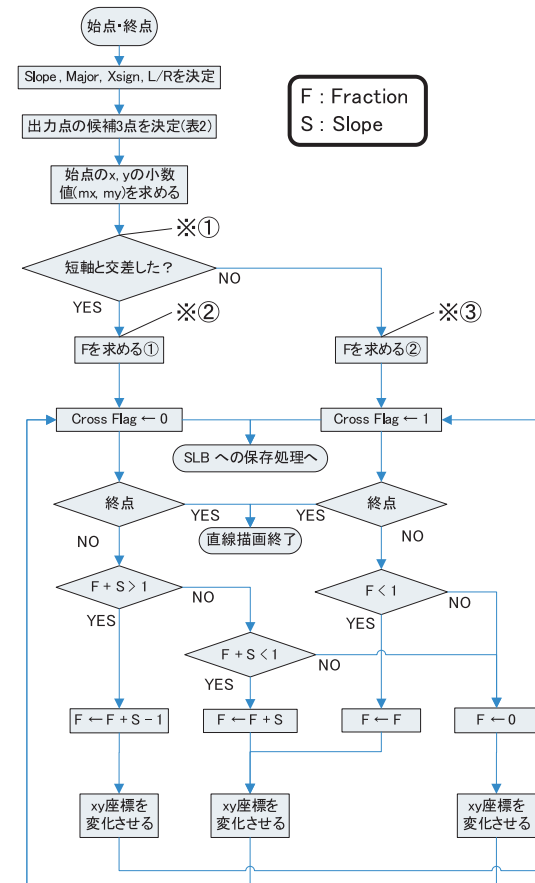


図 5 外郭線補間のフローチャート  
Fig. 5 Flowchart of our outline interpolation.

xy 座標の変化量 (「0: x - 1, y + 1」, 「1: x - 1, y」, 「2: x + 1, y」, 「3: x + 1, y + 1」, 「4: x, y + 1」) となる。表 2 内の「長軸上」は外郭線の長軸 (Major) となる軸上, 「短軸上」は外郭線の短軸 (Major でない方の軸) となる軸上に、出力点が存在することを意味する。

上記の分類に従って 3.1 節記載のシルエットライン情報を決定し SLB に保存するまでの全体のアルゴリズムの流れを図 5 に示す。

表 3 始点グリッドにおける次グリッド決定の式

Table 3 Equations to determine forward pixel grid in first grid.

直線種類		グリッド交差座標を求める式		
Major	Xsign	長短軸判定式	Cross Flag: 0	Cross Flag: 1
x	正	$My - Mx * slope \geq 0$	$fraction = fr$	$fraction = fr + slope$
	負	$My - (1 - Mx) * slope \geq 0$	$fraction = fr$	$fraction = fr + slope$
y	正	$Mx - My * slope \geq 0$	$fraction = fr$	$fraction = fr + slope$
	負	$(1 - Mx) - My * slope \geq 0$	$fraction = fr$	$fraction = fr + slope$

Mx : x 値の小数部, My : y 値の小数部, fr : 長短軸判定式の左辺の値

図 5 でポリゴン頂点を補間する際、「Major」、「Xsign」、「Slope」、「L/R Flag」は頂点座標値から直線補間式を導出する過程で得られる。補間を開始して 1 グリッドごとに歩進し、グリッド辺との交差座標値となる「Fraction」と交差軸「Cross Flag」を求め、SLB 保存処理へと移る。「Fraction」は直線補間式から得る座標値の小数点以下の値となる。次グリッドの「Fraction」は xy 座標を進める前に求める。以下に次のグリッドへ進む際の交差軸「Cross Flag」による処理の違いを示す。この処理の結果、表 2 の交差可能性点のいずれかへ進み、xy 座標を更新する。

(1) 「Cross Flag」が 0 で「Fraction+Slope」が 1 を超える場合、表 2 の“長軸上”の点が出力であることを示し、次グリッドの「Cross Flag」に 1 をセットする。「Fraction+Slope」が 1 と等しい場合は“頂点上”、1 未満の場合は“短軸上”の点が出力点となり、どちらの場合も次グリッドの「Cross Flag」には 0 をセットする。

(2) 「Cross Flag」が 1 で、「Fraction」が 1 未満である場合は表 2 内の“短軸上”、1 と等しければ“頂点上”の点が出力点となり、次グリッドの「Cross Flag」には 0 をセットする。

図 5 の ①, ②, ③ はそれぞれ補間開始点における処理で、直線種類により異なる。

① はグリッドへの入点が存在しない始点グリッドの「Cross Flag」と「Fraction」を求める処理である。この処理には表 3 内の長短軸判定式を使用し、「補間開始点から直線進行方向逆に延長させグリッドと交差させたときに、外郭線の長軸あるいは短軸となる軸のどちらと交差しているか」を判定する。判定の結果「true」の場合「入点は短軸上」にあることを意味し、フロー中の「Yes」へ進み、一方、「false」の場合「入点は長軸上」となりフロー中の「No」へと進む。② は入点が外郭線の短軸となる軸上で交差する場合で、③ は入点が外郭線の長軸上で交差する場合のそれぞれの「Fraction」を求める処理である。計算の詳細は表 3 に示す。

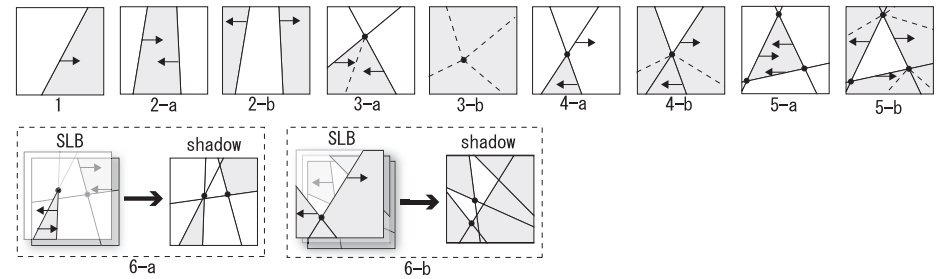


図 6 シルエットラインの交差パターン  
Fig. 6 Intersection pattern of Silhouette Line.

### 3.3.2 ポリゴン接続辺とシルエットライン検出

映像に依存してシルエットライン量は変動し、またレンダリング処理と並行してシルエットライン情報をバッファに記憶することで、シルエットラインとして記憶した外郭線が、レンダリング過程でポリゴンの接続辺となることが多発する。そのため、最終的にシルエットとなる外郭線のみを SLB に残す処理がバッファ容量の削減に不可避となる。接続辺であることが判明した場合、SLB から保存値を消去する。本研究では描画中のポリゴン外郭線のオブジェクト ID と、同一のオブジェクト ID を持つ SLB 内の情報とを比較し、「Polygon ID」と「L/R Flag」以外のすべてが一致する場合（接続辺では「L/R Flag」は相反する。また、同じ「Polygon ID」を持つ直線どうしのポリゴン接続辺は起きない）、接続辺と見なし該当するシルエットライン情報を SLB から消去する。

図 6 には複数のシルエットラインがグリッド内で交差するパターンと、それぞれのシルエットラインの検出例を示す。図内では各番号(1~6)が振られた正方形をグリッドと見なし、灰色の領域は影、白色の領域は照射を意味する。また、各直線から出ている矢印は、直線とグリッドに囲まれた左右どちらの領域が影であるか(L/R Flag)を示している。図 6 の 1 は 1 つのシルエットライン交差、2 から 5 は複数の交差を表す。2 は頂点を含まない交差であり、「L/R Flag」の値で 2-a と 2-b の 2 パターンの影領域に分類できる。3, 4, 5 は頂点を含む交差で、破線はポリゴン接続辺で消去されたシルエットラインを表す。3-b はすべてポリゴン接続辺のためシルエットラインは存在しない。4-a と 4-b はそれぞれ同じ「L/R Flag」の値を持つが、影領域が両シルエットラインの積の影領域(2 つ以上の直線でできた影領域の重なる部分のみ)と、和の影領域(2 つ以上の直線でできた影領域を合計した部分)の 2 つに分類できるパターンである。この判定には外郭線の終端を示す「End



Flag」と2つの外郭線が交わる頂点を示す「Vertex Flag」の情報を用いる。5-a, 5-bはすべてのシルエットラインがグリッドの端と交差せず、閉じた図形となっているパターンを表す。6-aは2つの異なるオブジェクトID, 6-bは3つの異なるオブジェクトIDのシルエットラインが交差したパターンで、各オブジェクトIDの影領域を判定した後、互いの影領域を合わせることでグリッド内の影領域を表す。

### 3.3.3 シルエットライン保存

前項の手段によってグリッド内の接続辺は検出されるが、さらに隠面除去テストにより不要なシルエットラインを削除する。この処理は、保存されているシルエットラインより光源に近いシャドウポリゴンがグリッドにオーバーラップした場合（ポリゴン内挿補間中のスパンラインである場合はポリゴン内部領域として通過するグリッド内全域がオーバーラップされることを意味する）を行う。これは、そのポリゴンの内側の遠方に位置するすべてのシルエットラインは新しいシャドウポリゴンの隠面側となり、影の投影上は不要となるためである。この判定はレンダリング中のシャドウポリゴンのデプス値と、すでに記憶されているシルエットラインのデプス値を比較すればよい。特に、光源に最も近いシャドウポリゴンがグリッドにオーバーラップした場合は、該当するグリッド内のすべてのシルエットライン情報とエッジバッファをリセットする。

シルエットライン情報をSLBに保存する際には、描画中の外郭線を持つオブジェクトIDとSLBの各オブジェクトIDを比べ、同じオブジェクトIDがある場合はその保存域に描画中のシルエットライン情報を追加記憶する。また、オブジェクトIDが一致しない場合は、新しいアドレスに描画中のシルエットライン情報を保存する。この結果SLBサイズは動的に変化するとともに、最終的に記憶される情報は視覚的に感知できるシルエットラインにほぼ等しいわずかな量となる。

以上の処理によって第1パスにおいて複雑に交差するシルエットライン情報であっても、効率的にSLBに記憶することができる。

## 4. 影付け

### 4.1 視点座標レンダリングとシャドウバッファ

第1パスで記憶したSLB情報を第2パスではレンダリングと並行して高速に読み出し、影の判定を行う必要がある。このために本研究では以下の手法を用いる。第2パスでは、レンダリング点を視点系から光源系座標値に変換するが、本研究では変換値を整数部と、小数部に分け、整数部はシャドウバッファ, IDバッファ, エッジバッファ, SLB(ハッシュ文字

表4 影の判定条件

Table 4 Conditions to determine shadow.

各種2次元バッファ			SLB			影判定
ID	シャドウ	エッジ	1次	2次	3次	
等しい	-	-	-	-	-	照射
	奥※1	-	-	-	-	影
異なる	手前※2	エッジでない	-	-	-	照射
		エッジである	影領域内	-	-	影
			影領域外	影領域内	-	影
			影領域外	影領域外	影領域内	影領域外

※1: レンダリング点がシャドウバッファのデプス値よりも光源から遠いことを意味する

※2: レンダリング点がシャドウバッファのデプス値よりも光源に近いことを意味する

列として)のアドレスに、また小数部はSLBのシルエットライン判定式に用いる。変換値がシャドウバッファのデプス値より光源から遠い場合、影と判定し、一方、光源に近い場合はエッジバッファでレンダリング点がシルエットラインか否かを判定する。シルエットラインの場合はSLBを用いた影判定を行い、シルエットラインでない場合は照射部分と判定する。シャドウバッファの低解像度により発生するセルフシャドウ・アライアスを防ぐため、本研究ではIDバッファの値とレンダリング点のオブジェクトIDが同じ場合は影付けを行わない。この影判定の条件を表4にまとめる。

表4において、レンダリング点のオブジェクトIDとIDバッファの値が等しい場合は無条件に照射、シャドウバッファの値より奥に位置する場合は無条件に影を表し、判定の対象はオブジェクトIDが異なり、エッジバッファが有効で、シャドウバッファのデプス値に対し光源寄りの場合の条件を示している。複数のシルエットラインが交差し、1次SLBで影と判定されない場合は、2次SLBと進み影と判定されるまでか、記憶された最後の次数のテストが終了するまで続ける。

### 4.2 シルエットライン内外判定

ここでは表4での“SLB”を使った影領域判定の方法を示す。また、4.1節の変換値の小数部をx座標値「Mx」、y座標値「My」として表す。表4の条件で、シルエットラインが1つの場合は1つの直線式で影判定を行うが、複数の場合は複数の直線式で影判定を行い、それぞれの結果の論理積(積の影領域)または論理和(和の影領域)をとる(図6の2~5)。図7ではグリッド内での内外判定の一例を示す。この例で、レンダリング点は、ポリゴン左辺のシルエットラインよりも右にあり影領域内(図7内の灰色の領域)と判定される。各直線式は表5に示すように「Major」、「Xsign」、「Cross Flag」で決定される。表5では直線

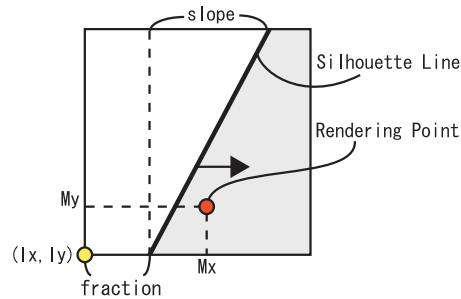


図 7 SLB を使った影判定の例  
Fig. 7 A sample to determine shadow by using SLB.

表 5 各シルエットラインの直線式  
Table 5 Line equations of each Silhouette Lines.

Major	Xsign	Cross Flag	直線式
x	正	0	$x = (1 / \text{slope}) * (\text{My} - \text{frac})$
		1	$x = (1 / \text{slope}) * (\text{My} - \text{frac}) + 1$
	負	0	$x = -(1 / \text{slope}) * (\text{My} - \text{frac}) + 1$
		1	$x = -(1 / \text{slope}) * (\text{My} - \text{frac})$
y	正	0	$x = \text{slope} * \text{My} + \text{frac}$
		1	$x = \text{slope} * (\text{My} - 1) + \text{frac}$
	負	0	$x = -\text{slope} * \text{My} + (1 - \text{frac})$
		1	$x = -\text{slope} * (\text{My} - 1) + (1 - \text{frac})$

表 6 表 5 の直線式を使った影判定  
Table 6 Determination of shadow area by using line equations of Table 5.

L/R Flag	Mx < x	影判定
Left	TRUE	影領域外
	FALSE	影領域内
Right	TRUE	影領域内
	FALSE	影領域外

式を y の式 ( $x = \text{slope} \times y + \text{Fraction}$ ) で表し, 表 6 ではその直線式と「Mx」を比較し「L/R Flag」を考慮して影判定することを示す. 同一アドレスの SLB 内で複数のシルエットライン内外判定をする際は, グリッド内の頂点の有無を判断してから影領域を再現する. 以下に具体的な判定方法を記述する.

表 7 2 直線間の影領域の判定条件  
Table 7 Condition to determine shadow between two lines.

A		B		二直線の「傾き」比較	積和決定式※3
Major	Xsign	Major	Xsign	A Slope <= B Slope	
x	正	x	負	-	Exp1 ※1
		y	-	-	Exp1
	負	x	正	-	Exp2 ※2
		y	-	-	Exp2
y	正	x	負	-	Exp1
		y	負	-	Exp1
	負	x	正	-	Exp2
		y	正	-	Exp2
AとBの「Major」が等しくかつ				TRUE	Exp1
AとBの「Xsign」が等しい				FALSE	Exp2

※1 : Exp1 = [L/R flag of B] nxor [End flag of A]  
 ※2 : Exp2 = [L/R flag of B] xor [End flag of A]  
 ※3 : Exp1およびExp2が[0(false)]の時は「積の影領域」, [1(true)]の時は「和の影領域」を意味する

#### 4.2.1 グリッド内に頂点を含まない場合

多くの場合, 各シルエットラインの影判定結果の論理積で積の影領域を再現すればよいが, トーラス型のような内側に穴を持つ物体は, 和の影領域を再現する必要がある. グリッド内に頂点は含まないためシルエットラインの交差はないとして, シルエットラインの「Fraction」の座標が他方のシルエットラインの内側にあるか否かで積および和の影領域の判定を行う (図 6 の 2-a, b).

#### 4.2.2 グリッド内に頂点を含む場合

シルエットライン情報の「Xsign」, 「Major」, 「Slope」, 「L/R Flag」, 「Vertex Flag」, 「End flag」を用い表 7 の規則性に従って積および和の影領域の選択をする (図 6 の 4-a, b). 複数直線が交差した場合も, それぞれ 2 本の直線 (A および B とする) の交差とし, A, B の交差判定条件を表 7 にまとめることができる. 直線 A, B の「Major」, 「Xsign」がそれぞれ等しい場合は, 直線 A, B の「Slope」を比べることにより, 異なる場合は「Major」, 「Xsign」の組合せのみにより「Exp1」, 「Exp2」の選択をする. ここで「Exp1」は直線 B の「L/R Flag」と直線 A の「End Flag」の否定排他的論理和演算を, 「Exp2」は排他的論

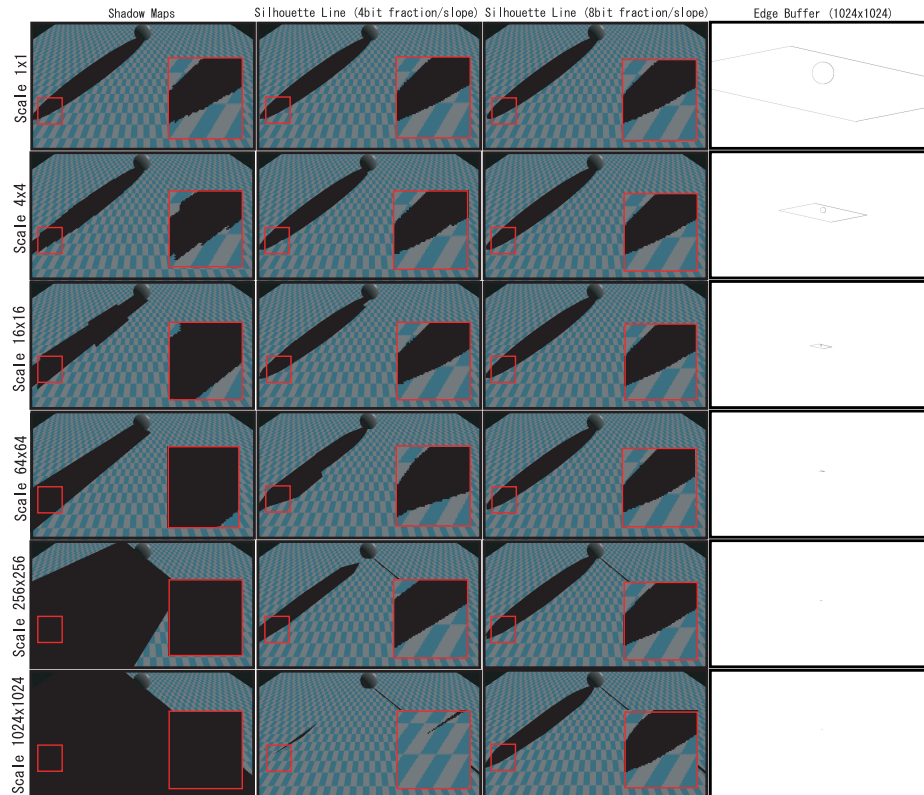


図8 シャドウマップとシルエットライン・アンチエイリアスの比較  
Fig. 8 Comparing Shadow Maps to Silhouette Line antialiasing.

理和演算を示す。この演算結果の0は直線A,Bの積の影領域,1は和の影領域の再現を意味する。

#### 4.3 アルゴリズム検証

本研究は実時間性を重視したアルゴリズムを課題としており、機能だけでなく計算速度およびハードウェアコストのそれぞれを考慮したものである。アルゴリズムを検証する第1ステップとしてJava言語を用い視覚検証テストを行った。図8に従来のシャドウマップと、シルエットライン・アンチエイリアスの比較した映像を示す。本研究ではシルエットライン

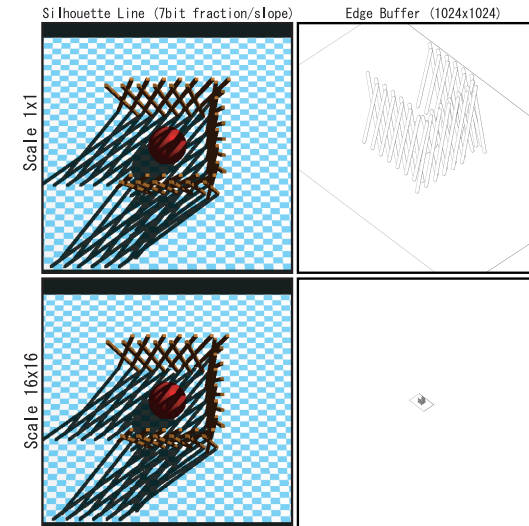


図9 複数のオブジェクトの影付け  
Fig.9 Shadowing with some objects.

情報の数値化により、描画範囲の拡大や縮小を任意のデータ精度で保障できる。またその特性をそれぞれの描画範囲での画質の違いも合わせて比較する。ここで図8, 図9, 図10内で使われているScale値は、第1パスでの描画範囲の拡大率(物体の縮小率)を示しており、各Scale値における第2パスの描画領域は同一のものとする。図8において、Scale1×1では4ビットの「Slope」、「Fraction」でもシャドウマップと比較して高品質な映像を得られるが、Scale16×16では4ビットデータの場合、エッジの欠ける現象が精度不足として生じる。この問題は「Slope」、「Fraction」の精度を増すことで解決できる。8ビット精度の場合、Scale64×64までほぼ一定にアンチエイリアスされることが示されている。図9では、複数のオブジェクトでの影付けを示しており、第1パス時に物体を縮小描画してもシルエットラインを数値情報として保存しているため影の精度がほとんど変化しないことが分かる。

#### 4.4 仮想シャドウバッファ

シャドウバッファを用いた影生成の深刻な問題の1つに、バッファの解像度に制約されて影の有効投影領域が狭くなることを指摘した。特に光源と視点方向が直交する場合、影の表示範囲はシャドウバッファの水平軸画素数に制限される。本研究は、シルエットラインを数



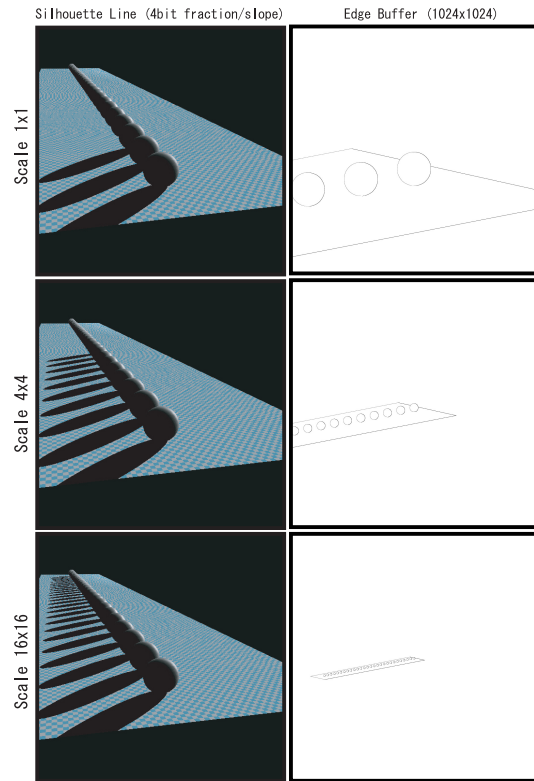


図 10 有効投影領域の比較  
Fig. 10 Comparing Shadowing in valid projection area.

値情報として持ち、第 2 パスでシルエットラインを再生することで解像度を仮想的に上げることができる。シルエットラインの精度は「Fraction」と「Slope」のデータがそれぞれ 4 ビットで、シャドウマップの 256 倍、8 ビットで 65,536 倍の解像度と等価である。これは物理グリッド内にその倍数のサブグリッドを設けたスーパーサンプリングと同等の機能を意味する。本研究ではシルエットライン上のサブグリッド情報を数値や属性データとした点が従来のグリッド情報と異なる。この結果、有効投影領域は、たとえば Scale16 × 16 で描画した場合、影の精度を保証しつつ 16 倍に拡大することができる。各 Scale 値における有効投影領域の比較を図 10 に示す。有効投影領域を拡大すると、オブジェクトあたりの面積が

表 8 物体サイズと SLB のメモリ使用量  
Table 8 Examples of SLB capacities and object sizes.

SL: Silhouette Line	Fig 8						Fig 9		Fig 10		
	Scale	1x1	4x4	16x16	64x64	256x256	1024x1024	1x1	16x16	1x1	4x4
SL交差グリッド数	3272	1104	274	68	14	4	23606	1159	4503	3112	2145
SL数(本)	3324	1154	342	154	106	96	24882	2421	4733	3620	3753
SL情報量(KByte)	31.4	10.7	2.9	1.0	0.5	0.4	235.3	20.1	44.2	31.9	27.3
シャドウバッファメモリ使用量	4096										

縮小され、そのシルエットラインが交差するグリッド数が減少するため SLB のメモリ使用量が減少する場合 (表 8 参照) と、一方有効投影領域に新たなオブジェクトが多数入り込み、シルエットラインの交差グリッドが増加するためメモリが増加する場合がある。

本研究の影の領域判定は、従来のサブピクセルやフラグメントバッファ等の物理バッファ上の処理ではなく、すべて数値演算で処理するためメモリの縮小化が図られる。

#### 4.5 今後の課題

本研究のアルゴリズムには以下の問題点がある。その 1 つに、オブジェクトを縮小したときに発生する黒い線がある (図 8 の Scale256 × 256 以降)。これは物体の縮小描画時に、XY 小数部の精度は「Fraction」、「Slope」により保証されるが、デプス値は SLB あたり 1 つの定義でありシルエットラインとシャドウバッファのデプス値を比較したときに誤判定され、消去されたことに起因する (シルエットラインごとのデプス値およびデプス値傾き保存で解決可能)。次に、トーラス型の物体の場合、トーラスの内側の曲線 (頂点が存在するグリッド) は項目 4.2.2 項の方法で解決できるが、内円の左右のグリッド (同一オブジェクト ID のシルエットラインが頂点でないが交差するグリッド) の影で一部欠ける等の可能性がある。ほかにも凹凸のあるオブジェクトは上記と同様の問題が生じるが、凸多角形の集まりとして考えオブジェクト ID を分ければ解決できる。

またソフトシャドウ技術においても、ハードシャドウを基本としてフィルタリングする方法が多くとられていることから、本方式を基により精度の高いソフトシャドウを表現することが今後の研究課題である。

#### 5. ハードウェア実装の検証

本方式はハードウェア実装に適応したアルゴリズムであり、ハードウェア実装を検証した。図 11 において映像描画の通常パスは、ポリゴン内挿補間、テクスチャマッピング、シェーダを経てオブジェクトを画素化し、画素を一時的に Pixel cache にバッファしたのち画像メ

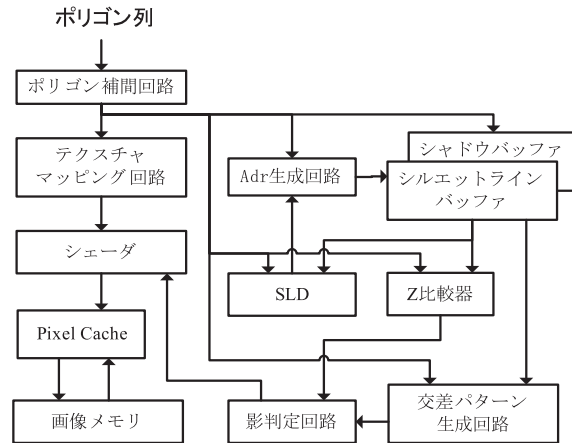


図 11 ハードウェア実装のプロセス

Fig. 11 The process of hardware implementation.

メモリに記憶する。一方、影による輝度の調整は影判定回路から得る信号をシェーダに与えることにより行う。このため、ポリゴンがシェーダの出力段階に至ると同じタイミング（遅延）で影の決定も行われなければならない。第 1 パスにおいて光源座標系のポリゴン列がポリゴン内挿補間回路に与えられ 3.3.1 項で定義した外郭線補間を行うとともに外郭線情報をシルエットラインバッファ (SLB) に記憶する。このバッファのアドレスは補間座標値を文字列としたハッシュ除算で構成した Adr 生成回路 (ADRG) で求める。並行してシャドウバッファには隠面除去処理を経たデプス値を記憶する。補間回路のシルエットライン情報と SLB に記憶されている情報を用いて、シルエットライン検出回路 (SLD) で 3.3.2 項の判定を行い、ポリゴン内部の共有線かシルエットラインかが決定される。SLD は z 比較器や外郭線情報一致検出回路等で構成する。本論文には多くの表や数式があるが、ほとんどがメモリテーブル化でき回路は小さくなる。SLD の検出結果を受けて新規外郭線となる場合は SLB に記憶するが、アドレスが重複する場合は新たなバッファエリアのアドレス生成は ADRG で行う。

第 2 パスでは視点座標系でポリゴンをレンダリングし、ADRG で補間値を光源座標系に変換し、エッジバッファのフラグが立っている場合、レンダリング点 (RP) でのシルエットライン情報を SLB から読み出す。SLB から読み出した情報から交差パターンの生成と影

判定テーブルを内蔵した交差パターン生成回路と影判定回路とから、4 章の各節に基づく処理によって影の有無を決定する。この信号はシェーダに与えられ、テクスチャや照明モデルから決定された RP の輝度のスケール値となる。シャドウバッファは 2 次元配列からなる画素単位のデプスバッファとエッジバッファからなるが、SLB は容量可変型の 1 次元バッファである。このバッファはハッシュ法によるアドレッシングにより、シルエットラインの図 1 に示す情報を記憶する。シルエットライン情報から交差パターンを生成し、RP のシルエットライン内外判定はテーブルを多用化することで、論理回路規模 25 万ゲート相当で概算 4 クロックサイクル/画素の決定が可能となる。このクロックサイクルは 1 次 SLB とした場合 (リンクレベルが増えれば速度は低下する) である。これはテクスチャマッピングの処理 (ヒットのミスのない) とほぼ同じであり、影の判定とレンダリングのフローとの同期が得やすい。

## 6. 結 論

これまで影生成アルゴリズムの中で問題とされてきたシルエットライン上でのアンチエイリアスと、影描画有効範囲の両問題は本研究の方法を用いることにより同時に解決することができた。描画オブジェクトが論理的に縮小され、グリッド内に多数のシルエットラインを含む場合であっても、SLB のリンク構造によりデータ保存が可能のため、メモリや計算負荷の増加があっても、スーパーサンプリング同様にアルゴリズムとしての限界はない。しかしスーパーサンプリングとは異なり、アンチエイリアスや影投下の有効領域の拡大効果は、シルエットライン画素のみを対象とするため、演算負荷を含めて通常メモリ容量は比較にならないほど少ない。今後の課題としては本アルゴリズムの IP 化 (LSI モジュール) に向けた検証等がある。

## 参 考 文 献

- 1) Williams, L.: Casting curved shadows on curved surfaces, *Proc. SIGGRAPH 1978*, Vol.12, No.3, pp.270-274 (1978).
- 2) Everitt, C. and Kilgard, M.J.: Practical and Robust Stencil Shadow Volumes for Hardware-Accelerated Rendering, Austin, Texas, NVIDIA (2002).
- 3) Jensen, H.W.: フォトンマッピング, Ohmsha, 第 5 章, pp.69-81.
- 4) Sen, P., Cammarano, M. and Hanrahan, P.: Shadow Silhouette Maps, *ACM Trans. Graphics (TOG)*, Vol.22, No.3, pp.521-526 (2003).
- 5) Fernando, R., Fernandez, S., Bala, K. and Greenberg, D.P.: Adaptive Shadow Maps,

*Proc. SIGGRAPH 2001*, pp.387–390 (2001).

- 6) Stamminger, M. and Drettakis, G.: Perspective Shadow Maps, *Proc. SIGGRAPH 2002*, Vol.3, pp.557–562 (2002).
- 7) Heidrich, W., Brabec, S. and Seidel, H.P.: Soft Shadow Maps for Linear Lights, *Proc. 11th Eurographics Workshop on Rendering*, pp.269–280 (2000).
- 8) Chan, E. and Durand, F.: Rendering Fake Soft Shadows with Smoothies, *Proc. 14th Eurographics Workshop on Rendering*, pp.208–218 (2003).
- 9) Laine, S., Aila, T., Assarsson, U., Lehtinen, J. and Akenine-Möller, T.: Soft Shadow Volumes for Ray Tracing, *Proc. SIGGRAPH 2005*, Vol.24, pp.1156–1165 (2005).
- 10) Assarson, U., Dougherty, M., Mounier, M. and Akenine-Möller, T.: An Optimized Soft Shadow Volume Algorithm with Real-Time Performance, *Graphics Hardware 2003*, pp.33–40 (2003).
- 11) Bresebham, J.E.: Algorithm for computer control of a digital plotter, *IBM Systems Journal*, Vol.4, No.1, pp.25–30 (1965).
- 12) Aken, J.R. and Novak, M.: Curve-Drawing Algorithms for Raster Displays, *ACM TOG*, Vol.4., No.2, pp.147–169 (1985).

(平成 19 年 11 月 9 日受付)

(平成 20 年 6 月 3 日採録)



山口 直

2007 年法政大学情報科学部デジタルメディア学科卒業。現在、法政大学大学院情報科学研究科情報科学専攻修士課程在学。



池戸 恒雄

法政大学情報科学部教授。工学博士。CG 実時間レンダリングアルゴリズムおよび GPU を教育研究領域とする。高速 GPU 商品化のため 2002 年 20 億円の投資により大学発ベンチャー企業デジタルメディアプロフェッショナルを設立。2004 年まで研究科長。2008 年から法政大学リエゾンオフィスを兼務。現在、シースルー眼鏡による仮想現実システムおよびデジタル TV 用 GPU の研究を進めている。