

# 光線微分とメッシュモデル単純化を用いたレイトレーシング効率化

## Efficient Ray Tracing using Ray Differential and Mesh Model Simplification

宮下 卓†  
Taku Miyashita

金井 理†  
Satoshi Kanai

岸浪 建史†  
Takeshi Kishinami

### 1. はじめに

工業製品の意匠デザインでは、高品位な製品モデル画像の高速生成技術が要求されている。高品位画像を得る代表的手法にレイトレーシングがある。この手法は、光線とモデル表面との交差テストが繰り返されるため処理時間を要する。特にモデルが、大規模な三角形メッシュモデルで表現されている場合、光線と三角形面分との交差テスト回数が莫大になり、高速な画像生成が困難になる。また高速化のためメッシュモデルを一様に単純化してしまうと、交点算出の精度低下を招き、生成画像の品質が下がる。そこで本研究では光線微分とメッシュモデル単純化を用いたレイトレーシングの交点計算処理の効率化手法を提案し、その有効性について報告する。

### 2. 提案手法の概要

処理概要を図1に示す。まず対象となるメッシュモデルの各面分を、オクトリー(八分木)の各ノードに関連づけて格納・管理する(A1)。次に、テスト光線を表示画素内からランダムに発生させ、モデルの三角形群との交差テストを行う。テスト光線とモデルとの交点算出を行い、反射・屈折で生ずる2次光線の方法を計算し、光線追跡を繰り返す(A2)。同時に交点における、光線微分値を算出し、交点を持つ三角形の属性値としてその値を付加する(A3)。メッシュ各面分に付加された光線微分値とオクトリーノードの寸法とを比較し、単純化が必要な三角形群を含むノードを決定する(A4)。ノードに含まれる三角形群を逐次統合し、局所的に面分数を減少させ出力する(A5)。単純化済メッシュモデルに対し、全画素においてレイトレーシング処理を行い、最終的な画像を生成する。

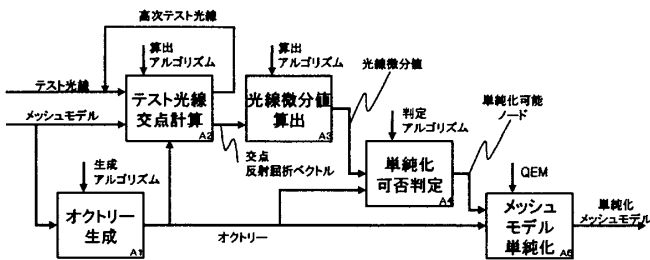


図1. 提案手法概要

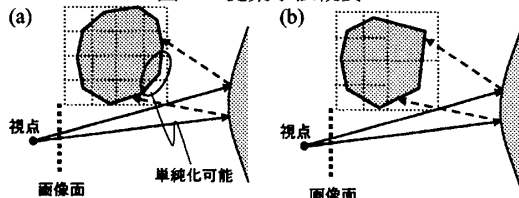


図2. 隣接光線の差分とモデル単純化の可否

### 3. 光線微分

#### 3.1 隣接光線の差分とモデル解像度の関係

図2のように、レイトレーシングにおいて画像面上の隣接2画素から照射される2本の光線について考える。この2本の光線は、一般に追跡距離が遠ざかるにつれ、または物体表面で反射・屈折する度に、光線間の位置および方向の差が増加する。この差が大きの場合、光線の空間密度は低くなり、図2(a)のように隣接2光線がモデルと交わる2交点の周辺付近に、光線との交差に寄与しない面分が多数存在することとなる。これらの面分は、画素の輝度計算に寄与しないため、図2(b)のように単純化して差し支えないと思われる。このように隣接2画素を通過する光線の位置・方向の差を追跡し評価できれば、交点周辺の面分に対する単純化の可否が推定できる。

隣接2画素を通過する光線の位置・方向の差を評価する方法として Igehy[1]らの光線微分を利用する。光線微分は、ある光線の位置・方向ベクトルの画素位置に関する導関数ベクトル値である。これを視点から逐次評価していくことで、隣接2光線の位置・方向の差分を推定することができる。本研究では表示対象となるメッシュモデル各部分の単純化評価指標として、この光線微分を評価し、その値に基づいてメッシュの面分数を減少させる。

#### 3.2 光線微分値の定義と算出方法

光線微分算出手法について述べる。画像面上の画素位置  $(x, y)$  を通る光線を、位置ベクトル  $\mathbf{P}(x, y)$  と方向ベクトル  $\mathbf{D}(x, y)$  により、 $\mathbf{R}(x, y) = \langle \mathbf{P}(x, y), \mathbf{D}(x, y) \rangle$  と定義する。

画像面上の  $x, y$  方向に微小距離移動した画素位置を通過する光線  $\mathbf{R}(x+dx, y+dy)$  は、式(1)で近似される。

$$\mathbf{R}(x+dx, y+dy) = \langle \mathbf{P}(x, y), \mathbf{D}(x, y) \rangle + \left\langle \frac{\partial \mathbf{P}}{\partial x}, \frac{\partial \mathbf{D}}{\partial x} \right\rangle dx + \left\langle \frac{\partial \mathbf{P}}{\partial y}, \frac{\partial \mathbf{D}}{\partial y} \right\rangle dy \quad (1)$$

式(1)右辺の導関数の組  $\langle \frac{\partial \mathbf{P}}{\partial x}, \frac{\partial \mathbf{D}}{\partial x} \rangle$ ,  $\langle \frac{\partial \mathbf{P}}{\partial y}, \frac{\partial \mathbf{D}}{\partial y} \rangle$  を光線  $\mathbf{R}$  の光線微分と定義する。光線微分の値は、視点と画素位置により初期化された後、1) 一様媒質中の伝播、2) 反射、3) 屈折に応じて変化する。光線微分値は、ある光線の追跡時にモデルとの交点が求まる度に[1]で示された方法に従い再評価される。

### 4. 光線微分に基づくメッシュモデル単純化手法

#### 4.1 オクトリーを用いた空間管理と単純化処理

本研究では、1)光線微分評価用テスト光線とモデル面分との交点探索処理、2)光線微分の値に基づくモデルの局所的な単純化処理、の効率化を目的として、Luebke[2]らが提案したオクトリーによるメッシュモデルの管理と単純化手法を利用する。この手法ではオクトリーの各ノード内で、①木構造内におけるノードの深さ・幅方向の情報、②分

† 北海道大学大学院情報科学研究科

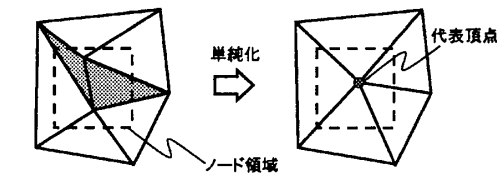


図3. 代表頂点によるモデル単純化処理

割領域の寸法, ③単純化の際に三角形面分の頂点群の移動先となる「代表頂点」の座標値, ④ノード内に頂点が少なくとも1個含まれる三角形のリスト, に関する情報を管理する. 初期化時にはメッシュモデル全体と視点を包含する立方体を生成し, ノード内に三角形が含まれていない状態, もしくは1ノードに三角形の1頂点のみが含まれる状態になるまで空間分割を再帰的に行い, 上述のオクトリーを構築する. 三角形の単純化処理時には, オクトリーのあるノードに含まれる三角形頂点の位置を, そのノードの代表頂点に統合する処理を行う(図3).

4.2 光線微分値を用いた単純化可否判定

これまでに述べた光線微分とオクトリーを用いてメッシュモデルの単純化判定を行う. ある画素を通過する光線  $R(x,y)$  とモデルとの交点  $P$  が求まっている場合, 隣接画素の通過光線は下式(2)により光線微分から推定できる.

$$P'(x+dx, y+dy) = P(x,y) + dP(x,y) = P(x,y) + (\partial P/\partial x)dx + (\partial P/\partial y)dy \quad (2)$$

すなわち, 隣接する2本の光線により生ずる交点間の距離は  $\|dP\|$  で推定される. 従って,  $\|dP\|$  の値が交点  $P$  を含むノード分割領域の対角線寸法より大きい場合, 交点周辺にはレンダリングに寄与しない微小三角形群が存在すると推定され, 単純化が可能と判定する. この判定をすべてのノードについて行った後, 単純化を行う.

4.3 法線を考慮した代表頂点位置決定手法

4.1節での単純化では, 単純化後の代表頂点を定める必要がある. この決定手法として Lindstrom[4]が提案した QEM(Quadric Error Metric)によるメッシュモデル単純化を修正し用いた. 本手法ではモデル単純化にオクトリーを用いているため, Lindstrom と同様ノード領域内に属する三角形群との2乗距離の和の誤差が最小となる代表点  $v$  を, 以下の式(3),(4)で算出する.

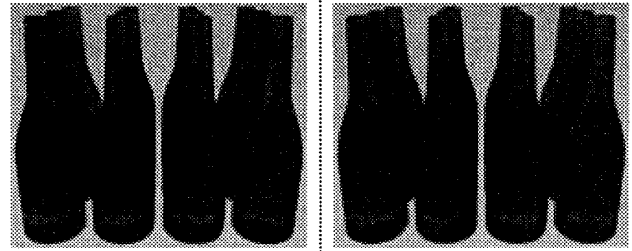
$$v = A^{-1}b \quad (3)$$

$$Q = \left( \sum_{p \in \text{planes}(v)} pp^T \right) = \begin{pmatrix} A & -b \\ -b^T & c \end{pmatrix} \quad (4)$$

ここで,  $\text{planes}(v)$  はノード領域内で  $v$  に統合される三角形集合,  $p$  は各三角形を含む平面の係数ベクトルであり,  $p = [n_x, n_y, n_z, (x_a \cdot n)]^T$  で表現される.  $n = (n_x, n_y, n_z)$  は三角形の頂点  $x_a$  における単位法線ベクトルである.

本手法がレンダリング対象とするモデルは NURBS モデルを三角形分割して作られるため, 三角形には頂点と法線が独立して与えられている. したがって, 本手法では3頂点をもつ法線の平均値を(4)式の  $p$  に用いる  $n$  とする.

(a)従来手法 (面分数 1,053,984枚) (b)提案手法 (面分数 44,574枚)



(c)a,bのワイヤーフレーム表示

(a) (b)



(d)aとbの差分画像

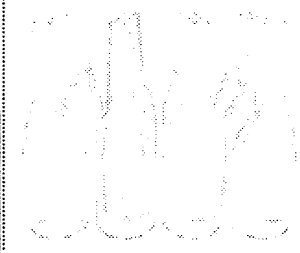


図4. 提案手法レンダリング画像

表1. 提案手法レンダリング時間

	(a)従来手法	(b)提案手法
画面解像度	1280 x 960	
テスト光線本数	—	5000本
面分数	1,053,984	44,574
前処理時間	—	60秒
レンダリング時間	600秒	420秒

5. 実装結果

提案手法を C++言語により実装し, 市販の3次元 CAD 向けレンダリングソフト(EXA 社, Surf-Ray)内にその処理を組み込み, 実験を行った. 図4のように入力モデルを192枚からなる瓶の NURBS モデルとし, これを一様に高密度分割したモデル(a), 本手法による光線微分に基づいた局所単純化モデル(b), それぞれに対してレイトレーシングを行った. また(a), (b)の背面から見たワイヤーフレーム表示を(c), 差分画像を(d)に示す.

このように本手法により(a)にくらべ 1/20 の面分数で画像の品質がさほど劣化せずにレイトレーシングが行われていることがわかる. また, (a)(b)におけるレンダリング時間, ポリゴン数などについて表1に記す. このように面分数が減少し, レンダリング時間も約 80%程度短縮していることが確認できた.

参考文献

[1] Homan Igehy : Tracing Ray Differentials , Proc. SIGGRAPH 1999, pp.179-186.  
 [2] David Luebke et al. : View-Dependent Simplification of Arbitrary Polygonal Environments, Proc. SIGGRAPH 1997 pp.199-208.  
 [3] Michael Garland et al. : Surface Simplification Using Quadric Error Metrics, Proc.SIGGRAPH 1997, pp.209-216  
 [4] Peter Lindstrom : Out-of-Core Simplification of Large Polygonal Models, Proc.SIGGRAPH 2002, pp.259-262