

## 第5章

### 降雪雪片のレンダリング法

#### 5.1 まえがき

降雪雪片のような微細な表現構造を持つ物体を，単純に幾何形状でモデリングするとエイリアシングの問題が発生し柔らかい質感を表現することが困難である．これを解決する方法としてテクセル[14]に代表される手法が提案されている．これは，表面構造と反射特性を3次元テクスチャによって表現し，ボリュームレンダリングと同様な手法でレンダリングするもので，エイリアシングの少ない良好な結果が報告されている．3次元テクスチャの各要素にはレンダリング時に必要な情報（光の反射分布，密度，色等）が記憶されるが，これらは異方性を持つ．しかしながら，従来は簡単化のため，光の反射分布の他は異方性を考慮しないか[14][15][16][18][17]，大幅な簡略化が行われている例が多い[20][21][19][22]．

本章では，まず，テクセルに代表される従来の3次元テクスチャ法について説明し，次に，降雪雪片に特化した3次元テクスチャ法を示す．さらに，従来の3次元テクスチャ法の異方性に関する問題点を指摘し，それを解決する異方性3次元テクスチャ法を示す．

#### 5.2 従来の3次元テクスチャ法

ここでは，Kajiya ら[14]のテクセル(texel)を例に従来の3次元テクスチャ法について説明する．

## 5.2.1 テクセルの要素

テクセル法では、物体は微小面から構成されていると考え、これを3次元の部分領域(ボクセル)に分解して記述する。ボクセルには微小面の集まりそのものではなく、その特徴を表す次の3要素を格納する。

第1要素：微小面の集まりの投影面積当たりの密度

第2要素：微小面の集まりの構造

第3要素：微小面の集まりのシェーディング関数

第1要素は、定義すべき物体を構成する微小面の集まりの投影面積当たりの密度であり、ボクセルの不透明度と微小面の集まりの光の反射強度を表している。投影面積当たりの密度とは、単位空間をある平面に平行投影した場合の、単位空間の投影面積に対する単位空間に含まれる微小面の集まりの投影面積の比率を意味する(図14参照)。

第2要素は、定義すべき物体の表面構造を表現している法線、接線などであり、第3要素を適用するために用いられる。

第3要素は、ボクセル中の微小面の集まりの反射特性を表す関数(シェーディング関数)である。

### 5.2.2 立方体の体積と密度

立方体の体積は、長さ、幅、高さの積で求められます。密度は、質量を体積で割った値です。この図は、立方体の体積と密度の関係を説明しています。

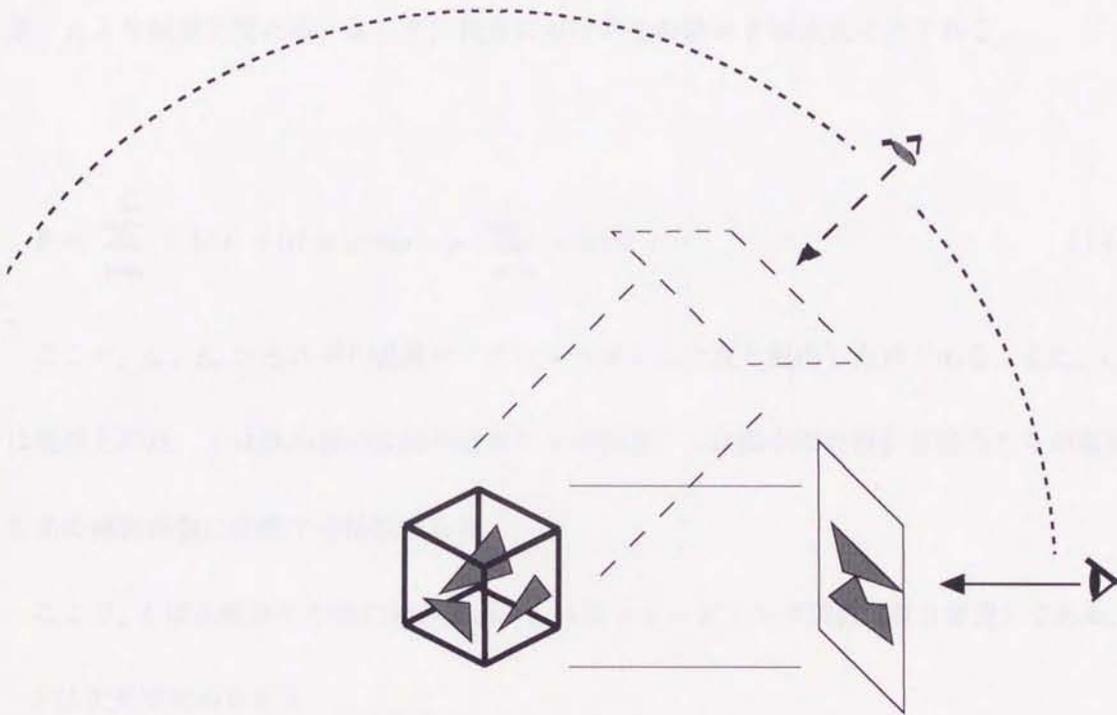


図14 投影面積当たりの密度

## 5.2.2 テクセルのレンダリング

テクセルのレンダリングはレイトレーシングにおけるボリュームレンダリングとほぼ同様の方法で行う。すなわち、図15に示すように視線上の各点  $s$  の微小面が視線方向へ反射する光を累積する。なお、光源光が各微小面に到達するまで、また、各微小面の反射光が視点に到達するまでの間に、それらの光量はセル中の投影面積当たりの密度（第1要素）により減衰を受ける。よって、視点における光の強さ  $B$  は次式で表される。

$$B = \sum_{t=t_{in}}^{t_{out}} | I_s(t) \psi(t) \rho(t) \exp(-\gamma \sum_{s=t_{in}}^t \rho(s)) | \quad (14)$$

ここで、 $t_{in}$ 、 $t_{out}$  はそれぞれ視線がテクセルへ突入した点と脱出した点である。また、 $t, s$  は視線上の点、 $\rho$  は微小面の投影面積当たりの密度、 $\gamma$  は微小面の投影面積当たりの密度を光の減衰係数に変換する係数である。

ここで、 $I_s$  は光源からの光の強さであり、 $\psi$  はシェーディング関数（第3要素）である。

$I_s$  は次式で求められる。

$$I_s = I \exp(-\gamma \sum_{r=L_m}^{L_{in}} \rho(r)) \quad (15)$$

ここで、 $L_m$  は光源の光線がテクセルに突入する点であり、 $I$  はその点での光の強さである。

5.3 傾斜線に垂直な次元テクセルのレンダリング

傾斜線に垂直な次元テクセルのレンダリングは、傾斜線に垂直な次元テクセルのレンダリングに似ている。

傾斜線に垂直な次元テクセルのレンダリングは、傾斜線に垂直な次元テクセルのレンダリングに似ている。

傾斜線に垂直な次元テクセルのレンダリングは、傾斜線に垂直な次元テクセルのレンダリングに似ている。

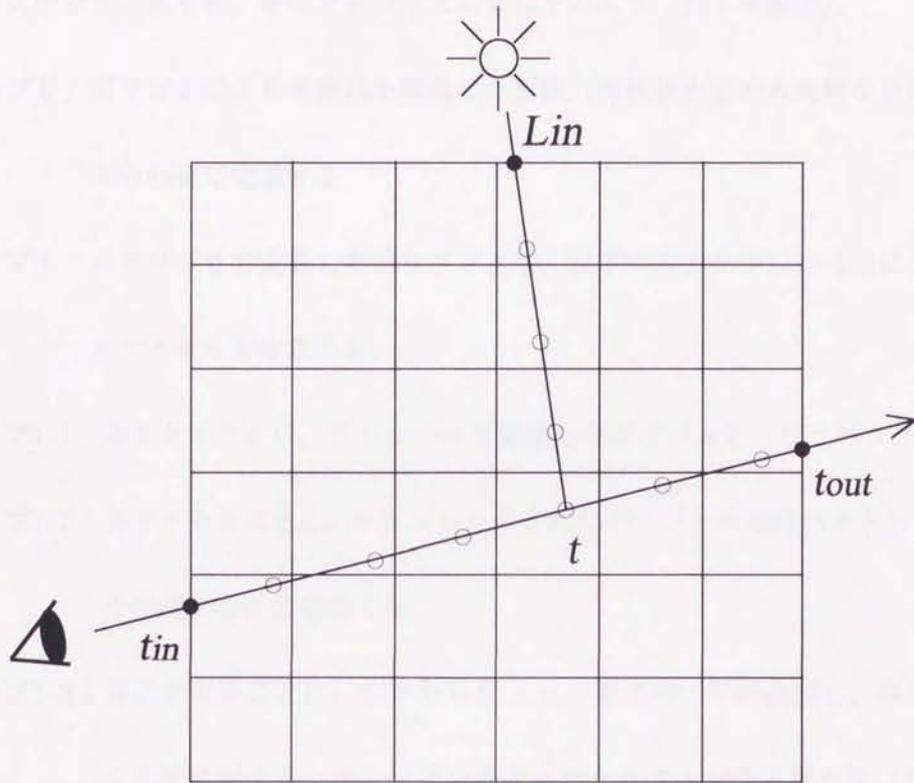


図15 テクセルのレンダリング

### 5. 3 降雪雪片専用3次元テクスチャ法

降雪雪片専用3次元テクスチャ法では、降雪雪片を模倣したポリゴン形状から直接3次元テクスチャを生成する。そのアルゴリズムを以下に示す(図16参照)。

ステップ0: ポリゴンにより雪結晶を模倣した形状(樹枝状六花や六角板など)を複数組み合わせて定義する。

ステップ1: ステップ0で定義したポリゴンより、以下のステップ1-1~1-3により3次元テクスチャを生成する。

ステップ1-1: ボクセルごとに、ステップ0で定義したポリゴンをクリッピングする。

ステップ1-2: 各ボクセルごとに、クリッピングされたポリゴンの法線ベクトルを合成し、そのボクセルに登録する。

ステップ1-3: 各ボクセルごとに、ボクセルを  $x, y, z$  軸方向へ平行投影し、ボクセルがつくる影に対する、クリッピングされたポリゴンが作る影の比(投影面積密度)を求め、それらの平均をそのボクセルの密度として登録する。

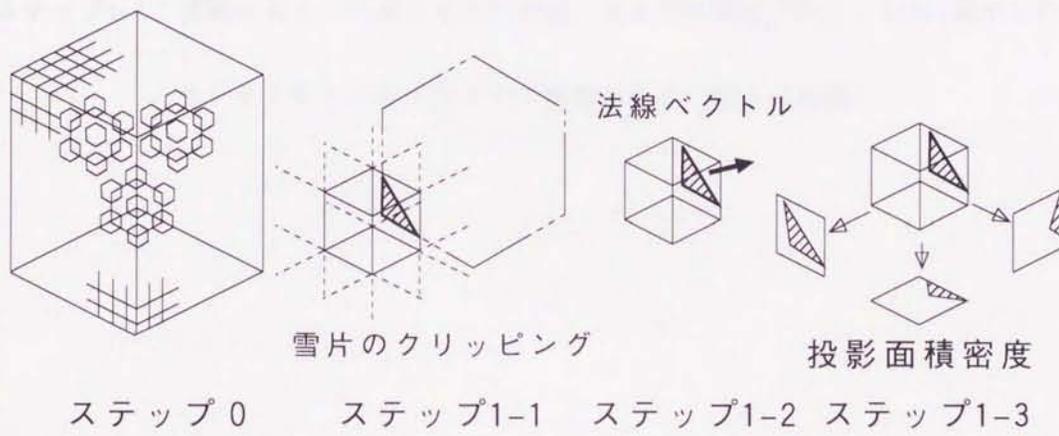


図16 雪片の3次元テクスチャの生成

雪片の3次元テクスチャは、レンダリング時間短縮とアンタイエイリアシングのため、視点からの距離に応じて最適の解像度のものを使用できるように、次のステップにより複数解像度用意する。

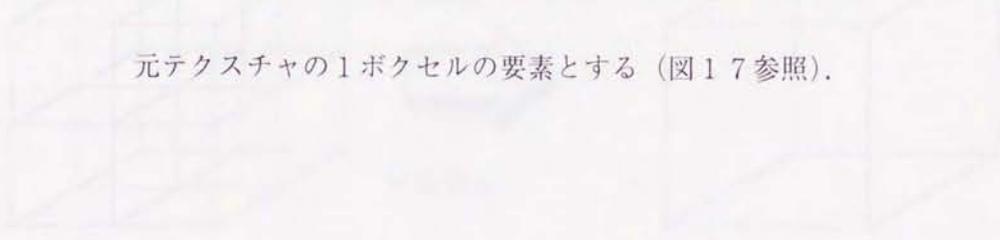
ステップ0：最大解像度の3次元テクスチャを用意する。

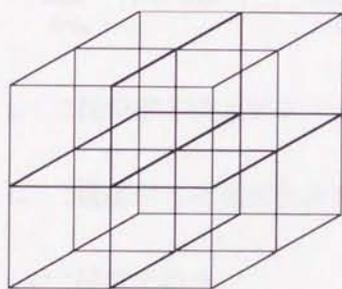
ステップ1：次のステップ1-1を、1ボクセルからなる3次元テクスチャが得られるまで

繰り返し、解像度の異なる複数の3次元テクスチャを生成する。

ステップ1-1：隣接する8つのボクセルの密度、および法線を平均し、1/8に縮小した3次

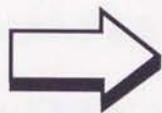
元テクスチャの1ボクセルの要素とする（図17参照）。



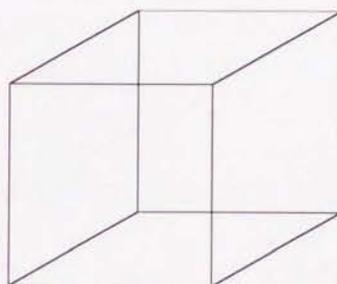


8 ボクセル

平均法線ベクトル



平均密度



1 ボクセル

図17 1/8の縮小

雪片を表すボリュームデータに対するレンダリングは図15に示したテクセル法と同様に、視線上の各点  $t$  が視線方向へ反射する光を累積することにより反射光の輝度を求める。光源光が点  $t$  へ達するまで、また各点の反射光が視点に到達するまでの間に、それらの光量はボクセル中の密度により減衰させられるため、視点に到達する光の強さ  $B$  は次式で表される。

$$B = \sum_{t=t_{in}}^{t_{out}} \left\{ I_r(t) \exp \left( -\gamma \sum_{s=t_{in}}^t \rho(s) \right) \right\} \quad (16)$$

$t_{in}$  : 視線が3次元テクスチャに突入する点

$t_{out}$  : 視線が3次元テクスチャから脱出する点

$t, s$  : 視線上の点

$\rho$  : 密度

$\gamma$  : 密度を光の減衰係数に変換する係数

$I_r(t)$  : 雪のシェーディング関数

光源光が視点へ反射する強さを表すシェーディング関数  $I_r(t)$  は次式で求め、式中の  $\cos$  は、値が負の場合は0とみなす。

$$I_r(t) = \rho(t) \{ I(t) (C_i + C_d \cos \phi + C_r \cos^n \zeta) + I_a C_d \} \quad (17)$$

$\phi$  : 法線ベクトルと光源ベクトルとの成す角

$\zeta$  : 光源光の正反射ベクトルと視線ベクトルの成す角

$n$  : Phong のモデルの係数

$C_i$  : 雪の等方散乱係数

$C_d$  : 雪の拡散反射係数

$C_r$  : 雪の鏡面反射係数

$I_a$  : 環境光の強さ

$I(t)$  : 点  $t$  への光源からの入射光の強さ

$$= I_a \exp \left[ -\gamma \left\{ \sum_{r=t}^{L_m} \rho(r) \right\} \right]$$

$I_0$  : 光源の光の強さ

$L_m$  : 光源の光線が 3 次元テクスチャへ突入する点

なお、選択した点の密度、法線は、3 次元テクスチャのその点の近傍のボクセルの値から 3 次元線形補間により求める。

雪片のボリュームレンダリングは雪片の 3 次元テクスチャを視点からの距離に応じて解像度の異なるものに切り替えて使用する、マルチ解像度レンダリングを行う (図 1 8 参照)。

図 3、図 4 の降雪シミュレーションの画像で示した雪片は、本手法により表現したものである。

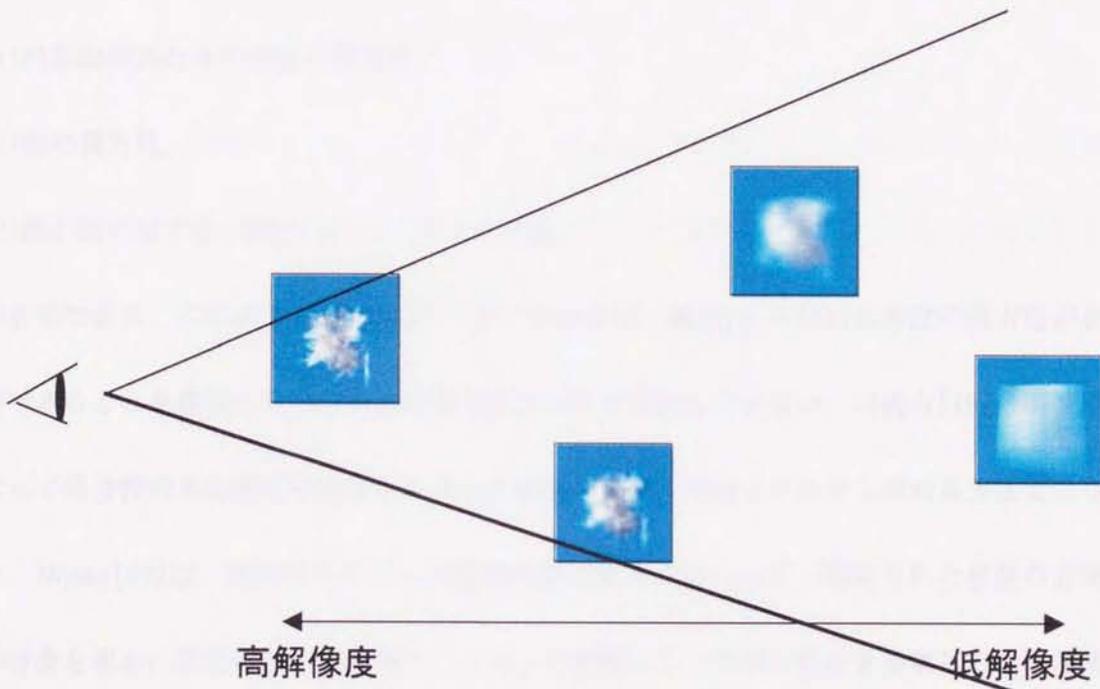


図18 雪片のマルチ解像度レンダリング

## 5. 4 3次元テクスチャの異方性に関するこれまでの研究

3次元テクスチャの1ボクセルに含まれるような微小な物体が、見る方向によって色が変化したり、見かけの大きさ（投影面積）が変化するような場合（例えば、葉の表裏で色が異なる草木など）には、異方性を考慮しなければうまく表現することはできない。

幾何モデルで表現された微細な構造を正確に3次元テクスチャで再現するためには、

(1) 投影面積当たりの密度の異方性、

(2) 色の異方性、

(3) 微小面に対する一般的なシェーディング法

が必要である。これまでの報告では、(1)については、Kajiyaら[14]は密度に異方性が必要であることを指摘したが、具体的な方法については示していない。村岡ら[19]は関数によって異方性のある密度を定義する方法を提案したが、幾何モデルから求める方法ではない。Noma[22]は、幾何モデルによる樹木の葉の集まりについて、固定された複数の方向の密度を求め、任意の方向の密度は、これらの補間により求める方法を提案したが、補間の詳細は不明である。Neyret[20][21]は密度の異方性を楕円体で近似したが、楕円体で表現しうる単純な異方性に限定されている。

(2)は、見る方向で色が変化する微小な物体（例えば表裏で色が異なる葉など）を表現するために必要であるが、ここで対象としている3次元テクスチャ法としては考慮した例は見あたらない。

(3)については、Noma[22]は樹木の葉の集まりのためのシェーディング関数を示した

が、枝やその他の形状に対しても適用可能か明らかではない。Neyret[20][21]は反射分布を楕円体で近似する方法を提案したが、微細構造が複雑なほど誤差が大きくなると思われる。

## 5.5 異方性を考慮した3次元テクスチャ法

ここでは、5.4の(1)~(3)を考慮した3次元テクスチャ法について、基本型とその改良型の2方式を示す。

本手法は、(1)の密度の異方性については、Neyretの手法のように楕円体のような単純な異方性に限定されず、Nomaの手法のように特定の形状(葉)に限定されない。(2)の色の異方性については、表裏で色の異なる葉のような薄い物体でも表現可能である。(3)のシェーディング関数については、Neyretの手法では楕円体で近似可能なほど単純な形状になるまでボクセルのサイズを小さく高精細にしなければシェーディング関数を定められないが、本手法ではボクセルのサイズは任意でよいという特徴を持つ。

一方、本手法では、入力する幾何形状としてポリゴンのみを対象としているが、Neyretの手法では、ポリゴンのほか、CSGや自由曲面、ポリウムデータなど多様な形状モデルから生成することができる。これについては、ポリゴン以外の方法で表現された幾何形状をポリゴンに変換する手法がこれまで開発されており([13][25]など)、これらを利用することで対応可能であると考えている。

### 5.5.1 基本型異方性3次元テクスチャ法

基本型は、各ボクセルごとに仮想的な球体を置き、この球体の表面上の点に異方性に必

要な情報を記憶させるというものである。

### 5.5.1.1 異方性のための仮想球体

密度の異方性については、“密度球”と呼ぶ投影面積当たりの密度を記憶するための仮想的な球体を考える。この球体の表面の点  $p$  に、点  $p$  から球の中心（すなわちボクセルの中心）方向に見た場合の投影面積当たりの密度を、そのボクセルに含まれるポリゴンすべてから求め登録する。

色の異方性と反射特性については、“陰影球”と呼ぶ仮想の球体を考え、ポリゴンと同じ法線を持つ陰影球の表面の1点に、ポリゴンの面積の重み付け加算で拡散反射係数と鏡面反射係数および Phong のモデルの係数を登録する（ポリゴンは Phong の反射モデルを用いているものとする）。

### 5.5.1.2 密度球・陰影球の実現方法

密度球、陰影球は、球体の表面をサンプリングした点の集合として表す。球体の表面上の1点は中心からその点へ向かう方向ベクトル（すなわち法線ベクトル）によって表せる。

この表面の点を表す法線ベクトルを求めるアルゴリズムを次に示す。

法線ベクトルを求めるアルゴリズム：

ステップ0：法線数を  $n$  とし、仮想球体の中心を  $O$  とする。

ステップ1： $O$  から乱数により方向を定められた  $n$  本の単位ベクトルを発生する（図19

(a)参照）。

ステップ2：任意の単位ベクトルを1つ選び、このベクトルを固定とする。

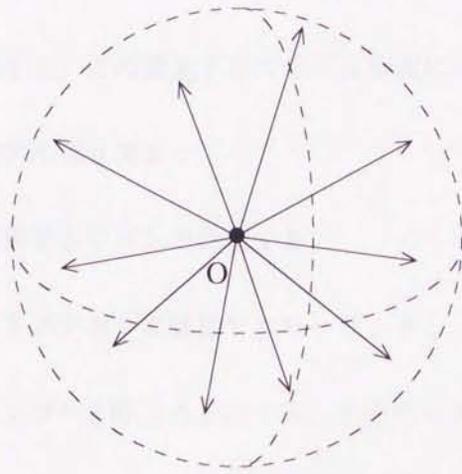
ステップ3：各々の単位ベクトルの先端に反発力を仮定する（図19(b)参照）。

ステップ4：反発力に従い，単位ベクトルの方向を修正する操作を，すべての単位ベクトルの修正角度が閾値以下になるまで繰り返す。

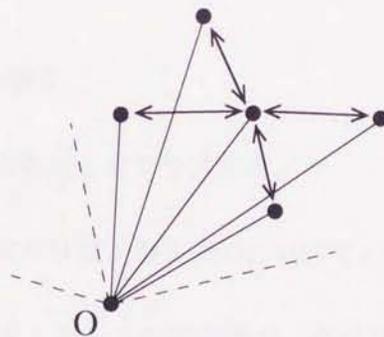
3.5.1.3 任意空間への法線ベクトル

任意空間への法線ベクトルを求めるには、まず任意空間の基底ベクトルを求め、それらの外積を計算する。

基底ベクトルを求め、それらの外積を計算する。



(a) n本の単位ベクトル



(b) ベクトル先端の反発力

図19 法線ベクトルを求めるアルゴリズム

### 5.5.1.3 仮想球体への登録アルゴリズム

ポリゴンで表現した微細な形状を用意し、これから各ボクセルの濃度球、陰影球の表面上に必要な情報を登録する。この設定アルゴリズムを次に示す。

濃度球・陰影球の登録アルゴリズム：

ステップ0：微細な形状をポリゴンで作成する。

ステップ1：ポリゴンをボクセルの範囲でクリッピングし、クリッピングされたポリゴン（サブポリゴンと呼ぶことにする）をそのボクセルに登録する。

ステップ2：各ボクセルごとに密度球、陰影球を用意する。

ステップ3：各ボクセルごとに、サブポリゴンの法線、面積、表面反射率などから、密度球、陰影球の表面上にそれらの情報を登録する。登録後、サブポリゴンは破棄する。

ステップ3の詳細を次に示す。

(1)密度、陰影球の表面上の各点  $p$  をクリアする。

(2)ボクセルに含まれるサブポリゴンのすべてについて、次の(2-1)～(2-3)を行う。

(2-1)ボクセルの投影面積を1とした相対面積で、各サブポリゴンの面積を正規化する。

(2-2)サブポリゴンの単位法線ベクトルを  $\mathbf{N}$ 、相対面積を  $S$ 、密度球の表面の点  $p$  の単位法線ベクトルを  $\mathbf{N}_0$  とすると（以後、ベクトルはすべて単位ベクトルとする）、密度球上のすべての点  $p$  について、次式で与えられる値  $S_p$  を加える（図20参照）。

$$S_p = S |\cos \phi|$$

$$= S |\mathbf{N} \cdot \mathbf{N}_p| \quad (18)$$

ここで、 $\phi$  は  $\mathbf{N}$  の  $\mathbf{N}_p$  との成す角、演算  $\cdot$  は内積である。

(2-3) サブポリゴンの  $\mathbf{N}$  と同じ法線を持つ陰影球上の点  $p$  ( $\mathbf{N}$  に一致する点が存在しない場合は、最も近い法線を持つ点) に、そのサブポリゴンの相対面積と表面反射係数 (拡散および鏡面) および *Phong* の係数を加える。なお、表面反射係数、*Phong* の係数はサブポリゴンの相対面積による重み付け加算とする。



#### 5.5.1.4 異方性3次元テクスチャのレンダリング法

本手法では、シェーディング関数  $\psi$  の中に密度  $\rho$  の効果を含むため、(14)式から2番目の  $\rho$  を除き、点  $s$  上での光源光の光の強さ  $I_s$  を  $\psi$  の引数とする次式によりレンダリングを行う。

$$B = \sum_{t=t_{in}}^{t_{out}} \left\{ \exp(-\gamma \sum_{s=t_{in}}^t \rho(s)) \psi(t, I_s(t)) \right\} \quad (19)$$

$\rho$  の計算には密度球を使用し、シェーディング関数  $\psi$  には陰影球を使用する。

$\rho$  の計算は、ボクセルに登録されている密度球の中心から光源の方向、または視点の方向へ半直線をのばし、その半直線と密度球の表面との交点を求める。その交点の密度を5.5.1.5に示す補間によって求め、それを投影面積当たりの密度とする。

シェーディング関数  $\psi$  には陰影球を使用する。視点から可視の半球の表面すべてをスキャンし、各点の視点方向への反射光を求め、その総和  $I_e$  を  $\psi$  の値とする（ただし、内積の結果が負の場合は0とみなす）（図2.1参照）。

$$\begin{aligned} I_e &= \sum_i S_i \cos \beta (I_s (Kd_i \cos \theta + Ks_i \cos^{\alpha} \alpha) + I_a Kd_i) \\ &= \sum_i S_i (-\mathbf{E} \cdot \mathbf{N}_i) (I_s (Kd_i (-\mathbf{L} \cdot \mathbf{N}_i) + Ks_i (-\mathbf{E} \cdot \mathbf{R})^{\alpha}) + I_a Kd_i) \end{aligned} \quad (20)$$

$I_s$  : ボクセルに到達する光源の光の強さ

$I_a$  : 環境光の強さ

$S_i$  : 陰影球の  $i$  番目の点の相対面積

- $K_d$  :         $\kappa$         拡散反射係数
- $K_s$  :         $\kappa$         鏡面反射係数
- $n_i$  :         $\kappa$         Phong のモデルの係数
- $\theta$  :         $-L$  と  $N_p$  との成す角
- $\alpha$  :         $-E$  と  $R$  との成す角
- $\beta$  :         $-E$  と  $N_p$  との成す角
- $R$  : 光源ベクトルの正反射ベクトル
- $E$  : 視線ベクトル
- $N_i$  : 陰影球の  $i$  番目の点の法線ベクトル
- $L$  : 光源ベクトル

光源ベクトル  $L$

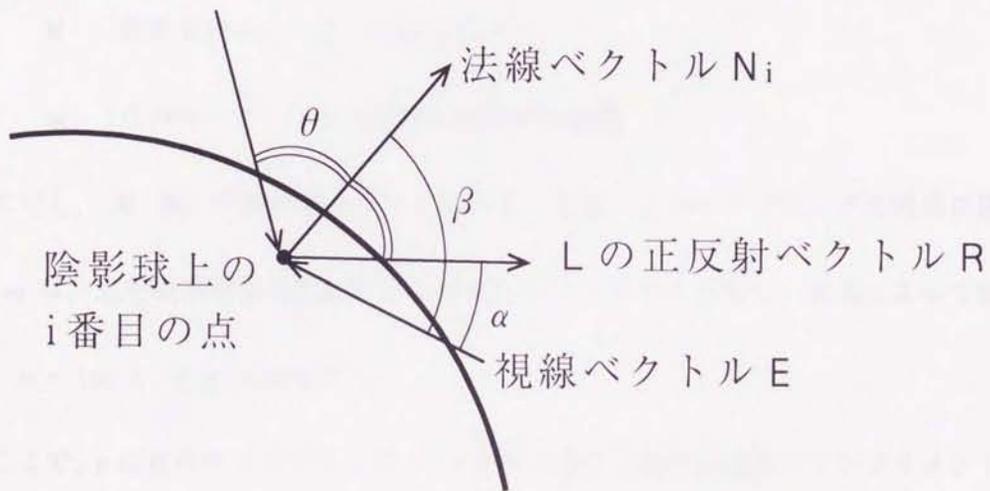


図 2 1 陰影球によるシェーディング

### 5.5.1.5 密度の補間法

密度球表面の任意の点の密度  $\rho$  は次式によって補間する (図 2.2 参照).

$$\rho = \sum_i \rho_i (\mathbf{N} \cdot \mathbf{N}_i)^m \quad (21)$$

$\rho_i$  : 密度球の  $i$  番目の点の密度

$\mathbf{N}_i$  :  $i$  番目の点の法線ベクトル

$\mathbf{N}$  : 密度を求めたい点の法線ベクトル

$m$  : 点のサンプリングの粗密に関する係数

ただし,  $(\mathbf{N} \cdot \mathbf{N}_i)$  が負の場合は 0 とみなす. なお, 点のサンプリングの粗密に関する係数  $m$  は, 点が仮想球体表面に均等に分布しているものとみなし, 次式によって定める.

$$m = \log k / \log \cos(4/\sqrt{n}) \quad (22)$$

ここで,  $n$  は点のサンプリング数,  $k$  は補間の重みに関する定数でインプリメントでは 0.5 とした.

球体表面上のサンプリング  
されている点の値

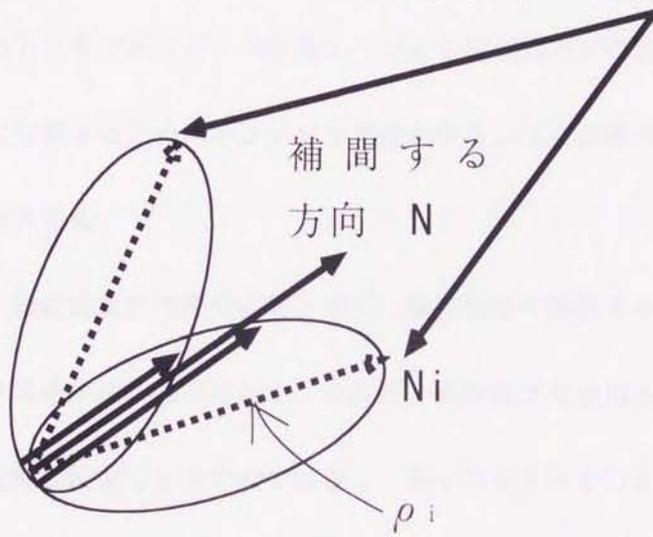


図 2 2 密度の補間

## 5.5.2 改良型異方性3次元テクスチャ法

基本型では、密度球・陰影球を球体表面をほぼ均等にサンプリングした有限な点の集合で表している。サンプリング数は多いほど精度は向上するが、それに比例して全体のデータ量も増大する。また、密度球から密度を求める際に補間を行っているが、サンプリング数が少ないほど、補間結果の信頼性は低くなる。さらに、陰影球にサブポリゴンの情報を登録する場合も、サブポリゴンの法線に一致する表面の点が存在しなければ、最も近い法線を持つ点に登録するため、サンプリング数が少ないほど法線の差が大きくなり、陰影付けの誤差は増大する。

このため、密度計算に補間を必要とせず、陰影付けの誤差を小さくできる改良型の異方性3次元テクスチャ法をここで示す。これは、仮想球体の表面上の点を、基本型のように前もって固定的に決定しておくのではなく、個々のボクセルごとに、それに含まれるサブポリゴンの法線と同じ法線を持つ球体表面の点を、仮想球体を構成する点として採用するというものである。

### 5.5.2.1 改良型の仮想球体

改良型で使用する仮想球体は1種類のみである。球体は表面の点の集合で表し、各点には、その点の法線ベクトル、正規化されたサブポリゴンの面積の和、重み付け加算された拡散・鏡面反射、Phongのモデルの係数が記憶される。

### 5.5.2.2 仮想球体への登録法

仮想球体への登録は、次のように基本型とほぼ同じように行う。

仮想球体の登録アルゴリズム：

ステップ0：微細な形状をポリゴンで作成する。

ステップ1：ポリゴンをボクセルの範囲でクリッピングし、クリッピングで得られたサブポリゴンを、そのボクセルに登録する。

ステップ2：各ボクセルごとに仮想球体を用意し、表面の点pの数を0とする。

ステップ3：各ボクセルに登録されたサブポリゴンについて、その法線に一致する仮想球体の表面の点pが存在すれば、そこにサブポリゴンの情報を登録し、存在しなければ、サブポリゴンの法線と同じ法線を持つ点pを仮想球体に追加し情報を登録する。登録後、サブポリゴンは破棄する。

ステップ3の詳細を次に示す。

(1)ボクセルの投影面積を1とした相対面積で、各サブポリゴンの面積を正規化する。

(2)サブポリゴンの法線Nと同じ法線を持つ仮想球体上の点pに、サブポリゴンの相対面積を加算し、拡散・鏡面反射係数、Phongの係数を、相対面積で重み付け加算する。

### 5.5.2.3 密度と輝度の計算

光の減衰を計算するための密度 $\rho$ は次式で求める。

$$\rho = \sum_i S_i |\mathbf{V} \cdot \mathbf{N}_i| \quad (23)$$

$S_i$ ：仮想球体のi番目の点の相対面積

$\mathbf{N}_i$ ：　　　　　　　　　法線ベクトル

V : 光線ベクトル (光源ベクトルや視線ベクトル)

密度はどのようなVに対しても補間なしで求められる。

視点方向へ反射する光の強度は陰影球における計算と同様に式(20)で求めるが、サブポリゴンの正確な法線を使用できるため基本型より正確な結果が得られる。

#### 5.5.2.4 データ量の削減方法

与えられた幾何形状によっては仮想球体の点が膨大となるボクセルも生じると考えられる。このため、異方性に大きく寄与する相対面積の大きなサブポリゴンのみを登録し、残りのサブポリゴンの情報はスカラ量として1つにまとめることでデータ量を削減する(図2.3参照)。相対面積の大きなサブポリゴンの情報を優先的に残すことで、陰影付けにさほど影響を与えないデータ量削減が期待できる。

インプリメントでは、異方性を考慮するサブポリゴンの最大枚数と、そのボクセルに含まれるサブポリゴンの全面積に対する、異方性を考慮するポリゴンの面積の割合を定めて、面積の大きなサブポリゴンから処理し、最大枚数か面積の割合のどちらかを越えたとき、残りのサブポリゴンをスカラ量としている。スカラ量は、残りのサブポリゴンの相対面積の合計 $S_a$ と、それらの拡散反射係数の和(サブポリゴンの相対面積による重み付け加算) $Kd_a$ の2つである。

スカラ量を考慮するため(20)、(23)式を次式のように変更する。

$$I_e = \sum_i^{n_i} S_i (-\mathbf{E} \cdot \mathbf{N}_i) (I_s (Kd_i (-\mathbf{L} \cdot \mathbf{N}_i) + Ks_i (-\mathbf{E} \cdot \mathbf{R})) + C_i S_i Kd_a (C_2 I_s + I_a)) \quad (24)$$

$S_a$  : スカラ面積

$K_{d_i}$  : スカラ拡散反射係数

$C_1, C_2$  : スカラ用係数(0~1)

$$\rho = \sum_i S_{v_i} |\mathbf{V} \cdot \mathbf{N}_i| + C_3 S_a \quad (25)$$

$C_3$  : スカラ用係数(0~1)

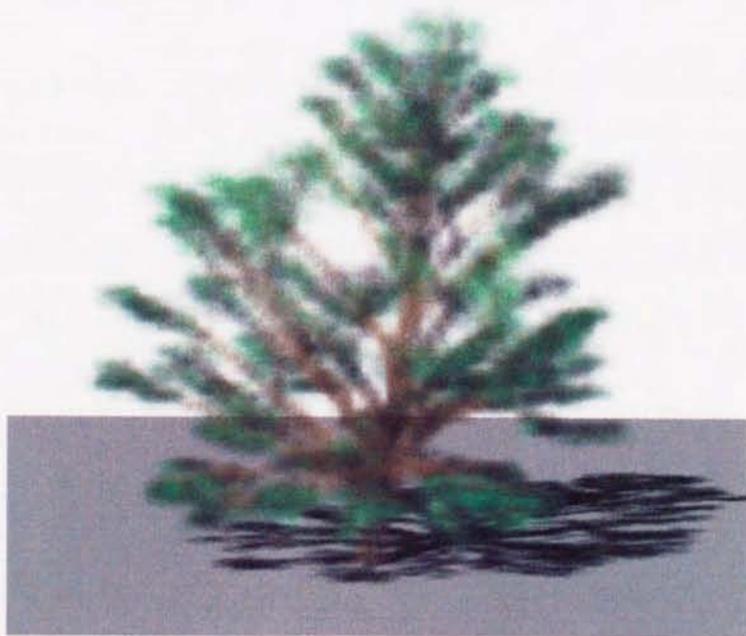


### 5.5.3 異方性3次元テクスチャの生成例

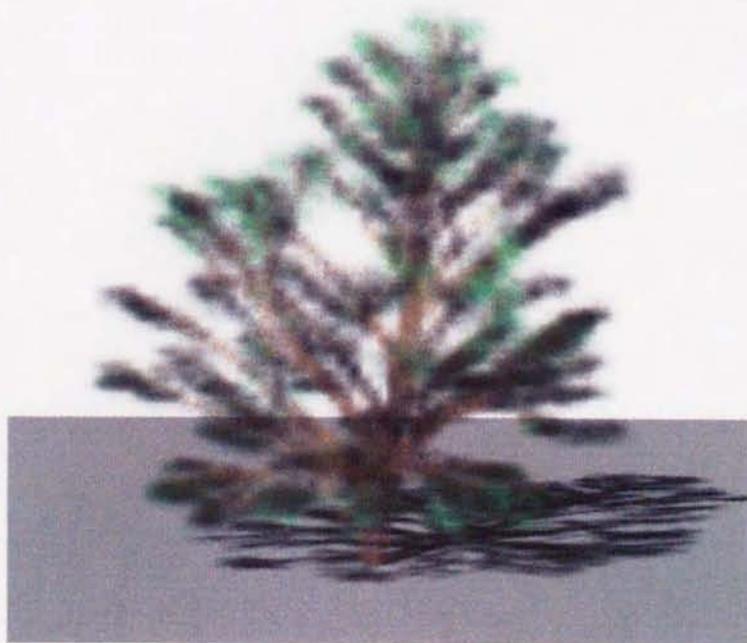
幾何モデルにより表現された樹木のレンダリング結果を図24に、それを $64^3$ のサイズの3次元テクスチャに変換し、拡大表示したものを図25と図26に示す。図25は基本型、図26は改良型によるものである。基本型、改良型とも仮想球体表面の点数（方位のサンプリング数と呼ぶことにする）は24と6であり、改良型では異方性を考慮する面積の割合を70%とした。方位のサンプリング数が24の場合は基本型、改良型ともレンダリングの結果に大差はない。6点の場合では、改良型ではあまり変化は感じられないが、基本型では密度計算に補間を用いていることによる誤差、および、サブポリゴンの法線とそれが実際に登録された点の法線との差による影響が大きくなり、正確な陰影付けが行えず、全体的に暗い画像になってしまっている。方位のサンプリング数減少によるデータ量削減の画像への影響は改良型の方が有利であるといえる。



図 2 4 幾何モデル（ポリゴン）による樹木

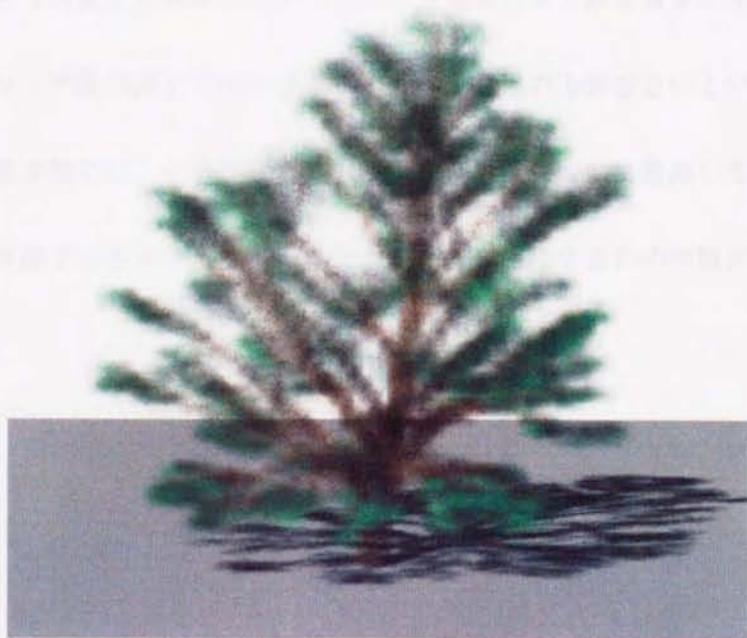


24方位

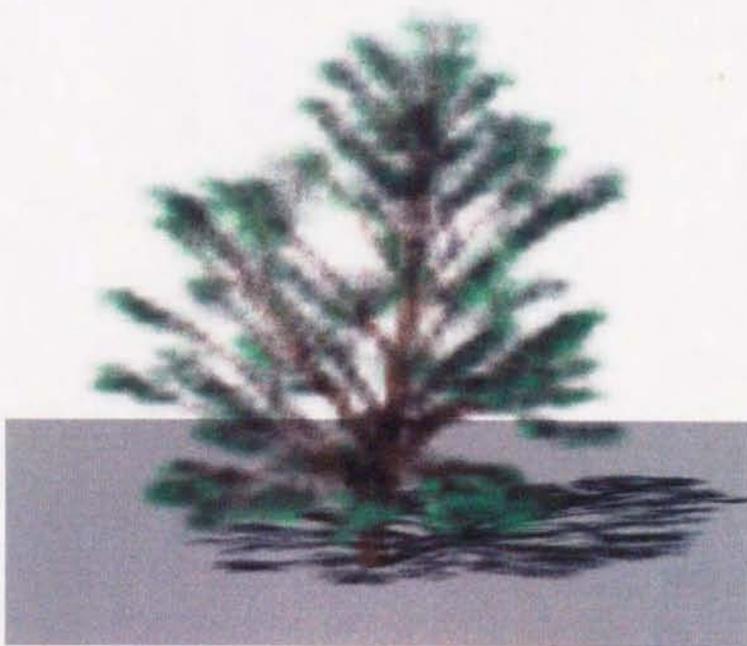


6方位

図25 基本型異方性テクスチャ



24 方位



6 方位

図 2.6 改良型異方性 3 次元テクスチャ

ポリゴン表現と3次元テクスチャのデータファイルのバイト数を表2に示す。データ量は方位のサンプリング数が同じであれば基本型より改良型の方が少ないという結果が得られた。これは、基本型では、それぞれのボクセルが全方位について保持しなければならないのに対し、改良型では各ボクセルに必要な方位のみを保持するため無駄が少ないためであると思われる。

方位数	基本型 (バイト数)	改良型 (バイト数)
4	1,048,576	1,048,576
6	1,572,864	1,572,864
8	2,097,152	1,572,864
12	3,121,440	1,572,864
16	4,145,728	1,572,864

表2 データ量

データファイル	方位のサンプリング数	データ量 (byte)
ポリゴンモデル	—	10,712,301
基本型異方性3次元テクスチャ	24	9,915,117
	6	2,763,551
改良型異方性3次元テクスチャ	24	2,792,180
	6	1,853,815

以後の生成例は改良型によるものを示す。

図27は、密度と色の異方性を確認するための幾何形状の画像であり、表裏で色が異なる（上面：緑、下面：赤）三角形板を螺旋状に、かつ下に向かうほど螺旋の半径が大きくなるように配置してある。この形状では、画像の中心付近では三角形の面が視線と平行に近くなるため、三角形の見かけの大きさが小さくなり、中心では完全に見えなくなる。また、中心より上に配置された三角形板はその下面が、また下に配置された三角形板ではその上面が見えている。図28に、これを3次元テクスチャ（ $32^3$ ）に変換した例を示す。画像の中心付近で密度が薄く、画面の上部では三角形板の下面の色（赤）、下部では上面の色（緑）になっており、色と密度の異方性が実現されていることが確認できる。比較のため、図29に密度と色の異方性を考慮していない従来の3次元テクスチャ法によるレンダリング結果を示す。全体が黄色（三角形板の両面の色（緑と赤）が混合した色）になってしまい色の变化や密度の変化も見られず、元の幾何形状の特徴を表現できないことがわかる。

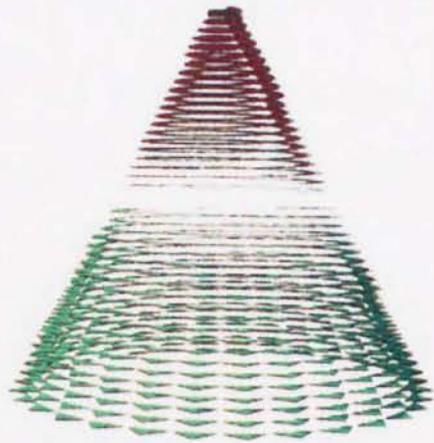


図 2 7 三角形板による形状のレンダリング



図 2 8 異方性 3 次元テクスチャ法によるレンダリング

Figure 29 shows a 3D cone rendered using the traditional texture mapping method. The cone is rendered with a yellowish-green color and a smooth, shaded surface. The rendering is done using a perspective projection, showing the cone from a slightly elevated angle. The background is plain white, and the cone is centered in the frame.

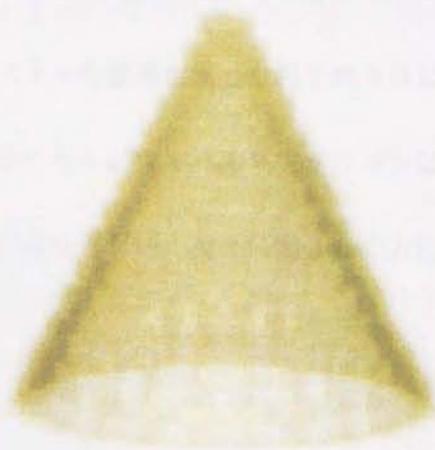
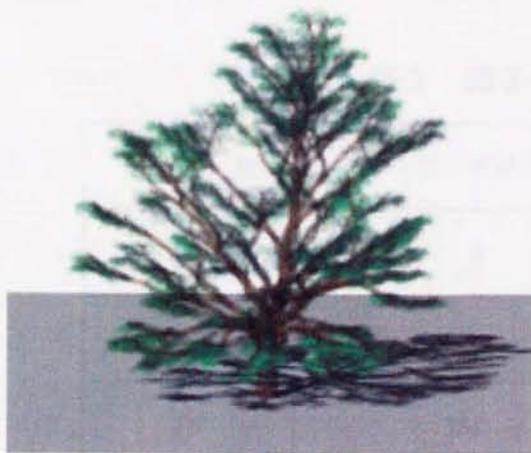


図 29 従来の 3 次元テクスチャ法によるレンダリング

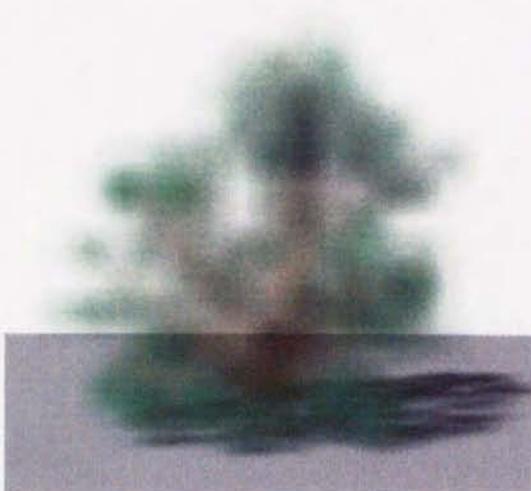
Neyret の手法では 1 ボクセルに含まれる幾何形状が楕円体で近似可能な程度に単純でなければならないという制限があり、任意の解像度の 3 次元テクスチャは、最も高解像度の 3 次元テクスチャを一度生成した後、隣接するボクセルを統合する操作を繰り返し適用して得なければならない (ただし、Neyret はメモリ節約のためオクトリーを採用している)。また、降雪雪片専用 3 次元テクスチャ法も同様に最も高解像度な 3 次元テクスチャを生成してから低解像度のテクスチャを生成することを基本としている。しかしながら、異方性 3 次元テクスチャ法では 1 ボクセルに含まれる幾何形状にこのような制限はなく、任意の解像度の 3 次元テクスチャを直接生成することができる。異なる解像度 ( $128^3$ ,  $32^3$ ,  $16^3$ ,  $8^3$ ) の 3 次元テクスチャを直接生成した例を図 30 に示す。なお、3 次元テクスチャの解像度が低いほど 1 ボクセルに含まれるサブポリゴンは一般に増加するため、方位のサンプリングは多くしなければならない。表 3 に図 30 の方位のサンプリング数とデータ量を示す。



128<sup>3</sup>



32<sup>3</sup>



16<sup>3</sup>



8<sup>3</sup>

図30 異なる解像度の3次元テクスチャ

表3 図30のデータ

ボクセル数	方位のサンプリング数	データ量(byte)
128 <sup>3</sup>	6	5,706,738
32 <sup>3</sup>	96	1,487,149
16 <sup>3</sup>	384	946,533
8 <sup>3</sup>	1536	728,063

3次元テクスチャ法はエイリアシングの少ない画像を生成することができる。図31に、複数の画像サイズで幾何モデルを直接レンダリングした結果と、本手法による3次元テクスチャ法でレンダリングした結果を比較して示す。両者とも1ピクセル当たり1本の視線でレンダリングを行い、3次元テクスチャは図30のものを使用した。幾何モデルによる結果では、どれもエイリアシングが発生しギザギザした画像になっているのに対し、3次元テクスチャによる画像ではエイリアシングのない滑らかな画像が得られていることが確認できる。

複数の3次元テクスチャを配置した例として、図32に樹木の3次元テクスチャを配置した並木のシミュレーション例、図33に樹林のシミュレーション例を示す。

降雪雪片への適用例として、図34に降雪切片を模倣した幾何モデルと、その3次元テクスチャ ( $32^3$ ,  $16^3$ ,  $8^3$ ,  $4^3$ ) を、図35にそれによる降雪風景のシミュレーションを示す。



図 3 1 幾何モデルと 3 次元テクスチャのレンダリング結果の比較

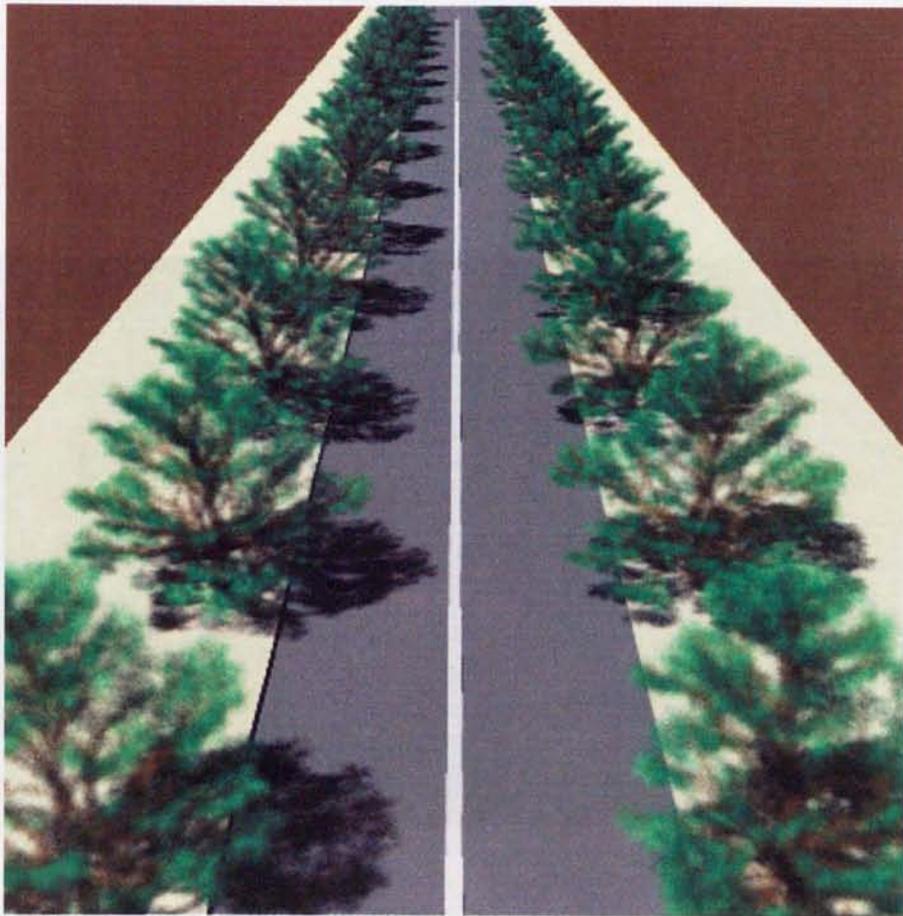


図32 並木のシミュレーション



図 3 3 樹林のシミュレーション



32 × 32 × 32    16 × 16 × 16    8 × 8 × 8    4 × 4 × 4

(a) 幾何モデル

(b) 3次元テクスチャ

図 3 4 降雪雪片の幾何モデルと複数解像度



図 3 5 異方性 3 次元テクスチャによる降雪風景のシミュレーション

## 5. 6 むすび

降雪雪片のような微細な表面構造を持つ物体の表現には、テクセルに代表される3次元テクスチャ法が向いている。このため、降雪雪片を模倣した幾何モデルより直接3次元テクスチャを生成する手法、およびそのレンダリング法を示した。

また、従来の3次元テクスチャ法には密度、色の異方性に問題があることを指摘し、それを解決する基本型異方性3次元テクスチャ法および改良型異方性3次元テクスチャ法を示した。画像生成実験より密度、色の異方性が実現されることを示し、基本型より改良型の方がデータ量が少なく、生成画像も良好であることを示した。