

## 確率モデルによるパーティクルシステムの描画法

7P-2

黄 國彰<sup>†</sup> 乃万 司<sup>‡</sup><sup>†</sup>奈良先端科学技術大学院大学 <sup>‡</sup>九州工業大学

## 1 はじめに

近年 CG において利用が盛んなパーティクルシステム [1][2] では一般に、パーティクル (粒子) の数が多いほど実感的な画像が生成できるが、それだけ高い計算コストを必要とする。しかし、オブジェクトによっては、本来存在するはずの粒子のごく一部しか目に見え (視点の方向に光を反射し) ない場合がある。この場合、目に見える粒子のみを選択的にモデル化することができれば、実感性はそのままで計算コストを大幅に減らすことができる。

本稿では、鏡面反射する微小オブジェクトを対象に、確率モデルを用いて目に見える粒子のみをモデル化する手法を提案する。また、具体的な対象として、ダイヤモンドダストと葉のきらめきをとりあげ、本手法の有効性を検証する。

## 2 確率モデルによるパーティクルシステム

鏡面反射では特定の方向のみに光が反射するため、各粒子について鏡面反射のみを考慮するパーティクルシステムでは、粒子自身による遮蔽効果を除けば、目に見える粒子の数は目に見えない粒子の数に比べてかなり少ない場合がある。

例えば、あるパーティクルシステムの粒子を微小平面とし、その法線方向が全方向に均一に分布しているとする。また、光源を単一の点光源 (単位方向ベクトル  $\vec{L}$ ) とし、視点を固定 (単位方向ベクトル  $\vec{V}$ ) する。halfway ベクトル  $\vec{H} = \frac{\vec{L} + \vec{V}}{|\vec{L} + \vec{V}|}$  による Phong モデルの鏡面反射項  $I_s = k_s I_l (\vec{N} \cdot \vec{H})^n$  (ただし、 $k_s$ : 鏡面反射係数、 $I_l$ : 光源の強さ、 $\vec{N}$ : 法線ベクトル) が閾値  $I_e$  以上である条件  $I_s \geq I_e$  から、閾値以上の鏡面反射光が目が届く法線方向は、 $\theta = \arccos\left(\frac{I_e}{k_s I_l}\right)^{\frac{1}{n}}$  によって定まる図 1 の単位球面上の円形部分に限られる。(単位球面の法線方向とパーティクルシステムの

法線方向とを同一視する。) そのため、目に見える粒子だけを扱えば、本来扱うべき粒子の  $\frac{A}{W} \approx \frac{\sin^2 \theta}{4}$  分の粒子だけを扱えばよいことになる。(A: 円形部分の面積, W: 単位球全体の面積)

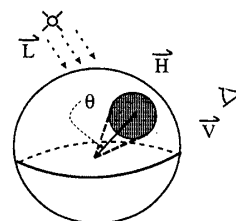


図 1. 単位球と法線方向

このとき、法線方向が上の円形部分に入る粒子だけを生成し描画しても、すべての粒子を生成して描画した場合と同じ画像が得られるはずである。

また、アニメーションの場合はフレーム毎に、光源方向、視点、ならびに各粒子の法線方向などが変化する。この場合はフレーム毎に、現在扱っている各粒子の法線方向が図 1 の円形部分に入るかどうかを調べ ( $\vec{N} \cdot \vec{H}$  と  $\cos \theta$  を比較)、円形部分から外れた場合にはその粒子のデータを削除し、一方では、光源方向、視点、ならびに各粒子の位置や法線方向の変化から、新たに上の円形部分に入るであろう粒子を確率的に生成すればよい。もちろん、フレーム間の変化が少ないほど、削除/生成する粒子の数は少なくてすむ。

## 3 モデルの適用

上記の確率モデルによるパーティクルシステムを、ダイヤモンドダスト現象と葉のきらめき現象に適用する方法について説明する。

## 3.1 ダイヤモンドダスト

実際のダイヤモンドダストは多面体の氷晶であるが、ここでは、ダイヤモンドダストの粒子を微小三角形でモデル化し、粒子群に次のような属性を与える。

**粒子の分布:** パーティクルシステムをある有限の空間領域内で考え、その領域内で一定の粒子の密度を持つものとする。

**粒子の運動:** ダイヤモンドダストのような非常に小さな粒子は、特に風などの影響がない場合はゆっ

Rendering of Particle Systems for Specular Reflection Objects by Stochastic Model

Kuo-chang Hwang<sup>†</sup>, Tsukasa Noma<sup>‡</sup>

<sup>†</sup>Nara Institute of Science and Technology (NAIST)

8916-5 Takayama, Ikoma, Nara 630-01, Japan.

<sup>‡</sup>Kyushu Institute of Technology

680-4 Kawazu, Iizuka, Fukuoka 820, Japan.

くりと地面に落ちてくる。そこで、重力による下向きのわずかな加速度と風などによるランダムな加速度とを与える。

**粒子の法線:** 粒子の法線方向は全方向均一に分布していると考えられる。フレーム毎に微小な回転を加え、法線方向を変化させる。

2節で述べたように、パーティクルシステムの領域内で、粒子の密度を満たすように、目に見える粒子のみを、その初期位置、速度、法線とともに生成する。フレーム毎に、粒子の位置や法線方向が変化するのに伴い、目に見えなくなった粒子のデータの削除と、目に見えるようになる粒子の確率的な発生を行なう。以上のようにして、ダイヤモンドダストを表現した例を図2に示す。

### 3.2 葉のきらめき

直射日光下で樹木を観察すると、木の葉の鏡面反射成分によって、葉の茂みの部分がきらきらと輝いて見えることがある。従来このような現象を描画する際は、葉のレベルまで形状モデリングを行ない、反射計算を行なう必要があった。しかし、ある程度離れた距離から樹木を見る場合は、確率的に「きらめき」を表現すれば十分である。

そこで本項では、輝く木の葉をパーティクルシステムの粒子とみなし、鏡面反射で輝く木の葉を確率的に生成して、木の葉のきらめきを表現する。前項のダイヤモンドダストでは、粒子自身による遮蔽効果を除けば、パーティクルシステムがオブジェクトのすべての反射すなわち見え方を表していたのに対し、本項では、鏡面反射成分のみを扱っている。通常の樹木のモデリングとは別にこのようなきらめき表現を行なっても、適当な形状との重ね合わせを行なうことによって、自然な木の葉のきらめきが表現できる。

**粒子の分布:** 粒子すなわち木の葉は、葉の茂み全体に一様に分布しているものとする。ここでは、木の茂みを半球状の形状に重ね合わせ、その半球上に葉が分布するものとする。

**粒子の運動:** 一般に木の葉は風などによって動き、きらめきが変わるが、本稿では静止しているものとする。

**粒子の法線:** 葉の法線方向は様々であるが、なるべく光を得ようという葉の性質から、平均すると、茂みに重ね合わせた半球上のそれぞれの場所で、その法線方向と一致するものとみなせる。そこで、半球の法

線と一致する場合に最も多く、半球の法線から離れるにしたがって少なくなるような葉の法線分布を設けた。

また、太陽の向きを変化させるアニメーションを作成した。太陽の移動につれて、フレーム毎に粒子の削除と生成を行なうことによって、光源方向の変化に伴って葉の輝きが変化する様子を表現できた。(図3)。

## 4 実験結果と評価

実験には Sun SS20 上の C 言語を用いた。背景を加えた結果は、図2および図3に示されている。また、約 10,000 個の粒子を想定した場合のダイヤモンドダストの生成では、実際にモデル化された粒子は、従来の手法の 1.2% である 120 個程度であるにもかかわらず、ほぼ同じ画像を得ることができた。レイトラシング法を用いた場合の、計算時間とメモリ使用量の比較を表1に示す。

表1. 従来の手法と確率モデル法との比較

	従来の手法	確率モデル法
CPU 時間 (hh:mm:ss)	2:30:45	15:40
メモリ使用量 (KByte)	2,547.50	29.60

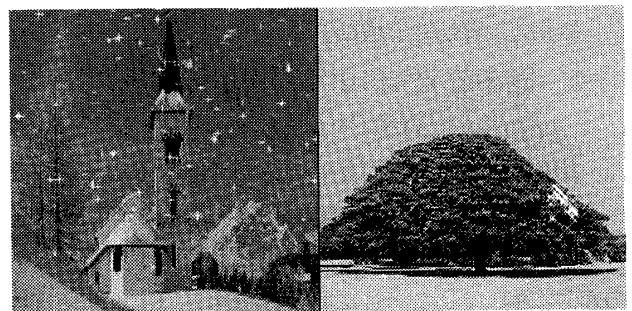


図2. ダイヤモンドダスト

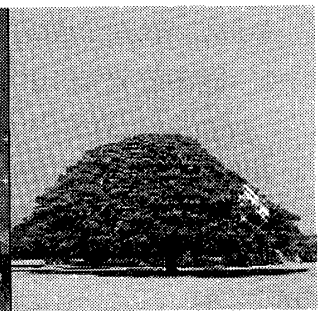


図3. 葉のきらめき

## 5 むすび

本稿では、確率モデルによるパーティクルシステムの新しい描画法を提案し、従来のシステムと比べて、計算コストとメモリ容量が大幅に削減できることを示した。今後の課題としては、日没時の海面のきらめき [3] などへの適用があげられる。

### 参考文献

- [1] William T. Reeves: Particles Systems - A Technique for Modeling a Class of Fuzzy Objects, *Comput. Graph.*, 17(3) 359-376, 1983.
- [2] William T. Reeves and Ricki Blau: Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems, *Comput. Graph.*, 19(3) 313-322, 1985.
- [3] Wolfgang Krueger: Intensity Fluctuations and Natural Texturing, *Comput. Graph.*, 22(4) 213-220, 1988.