

大域照明におけるファーのリアルタイムレンダリング

Real-time Rendering of Fur with Global Illumination

山田 大輔† 山本 強† 土橋 宜典†
Daisuke Yamada Tsuyoshi Yamamoto Yoshinori Dobashi

1. 背景と目的

近年、映画やゲームなどの分野で写実的な CG 映像が用いられている。CG による様々な生物の描画も行われているが、中でも毛皮(ファー)は哺乳類等の動物が持つ大きな特徴である。一般に、3DCG におけるファーのレンダリングは計算コストが高いという問題点がある。そこで、ファーの断面図のテクスチャをマッピングした半透明ポリゴンに重ね描きすることで、写実的かつ高速なレンダリングを可能とした手法が提案されている[1]。しかし、輝度計算については直接光のみが考慮され、大域照明までは考慮されていない。そこで本研究では、Precomputed Radiance Transfer(PRT)法[2]を利用して大域照明を擬似的に考慮したファーのリアルタイムレンダリングを目的とする。基本的には、ファーのレンダリングに必要な輝度データを前もって計算することで高速化を達成する。しかし、前計算データが膨大になるという問題が発生するため、ファーを含む物体の輝度は皮膚に近いほど暗くなることを考慮し、データの圧縮を図る。

2. 既存研究

まず、提案法で利用しているファーのレンダリング手法[1]と PRT 法[2]について概説する。

2.1 ファーのレンダリング手法[1]

図1左に示すように、ポリゴン上に毛の断面図のテクスチャ(シェル・テクスチャ)をマッピングしたポリゴンを半透明表示してその輝度値をブレンディングすることでファーを高速に描画する。ただし、物体表面に相当する最下層のポリゴンの半透明表示は行わない。しかし、この方法だけでは、重ねたポリゴンを横から見た場合に隙間が生じてしまう。そこで、図1右に示すように、ファーに対し垂直方向に切り取ったテクスチャ(フィン・テクスチャ)を物体表面に垂直なポリゴンにマッピングして半透明表示することで隙間を埋め、視覚的クオリティが改善できる。

この方法により、任意の表面に対しファーをリアルタイムにレンダリングできるが、ファー部分のライティング処理は直接光しか考慮されていない。

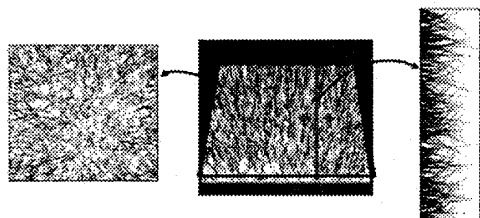


図1: シェル・テクスチャ、フィン・テクスチャ

† 北海道大学情報科学研究科情報メディア環境学講座

2.2 Precomputed Radiance Transfer(PRT)法

前処理において、視点位置や光源情報に依存しないデータを計算しておくことによって大域照明をリアルタイムに実現する手法である。物体のある点 x の輝度 $I(x)$ は、その物体の周囲の画像を表す環境マップの輝度値から計算する。

$$I(x) = \frac{1}{\pi} \int L(\omega) V(\omega) \max(\omega \cdot n, 0) d\omega \quad (1)$$

L は物体を取り巻く光源(環境マップ)、 V は遮蔽情報で方向 ω が遮られていたら 0、それ以外は 1 となる。 n は点 x の法線を表す。

一般に、上式の計算時間は非常に膨大になるという問題点がある。そこで、文献[2]では、物体は動かないものと仮定し、球面調和関数(Spherical Harmonics、以下、SHと略す)を用いることでリアルタイムレンダリングを実現している。まず、前処理において以下の式で表されるデータを各頂点ごとに計算する。

$$M_i^m = \frac{1}{\pi} \int Y_i^m(\omega) V(\omega) \max(\omega \cdot n, 0) d\omega \quad (2)$$

ここで Y_i^m は SH を表す。 M_i^m は伝達ベクトル(Transfer Vector)と呼ばれる。

レンダリング時においては、光源情報 L を SH により近似する。SH は直交性を持ち SH で近似したものの同士の掛け算はそれらの係数同士を掛け合わせるだけで求めることができる。したがって事前に計算して保持しておいたデータ M_i^m を実行時に光源情報 L を近似した結果 I を掛け合わせるだけで式(1)を計算できる。すなわち、

$$I(x) = \sum_{l,m} I_l^m M_l^m \quad (3)$$

これにより物体の輝度計算を高速に行える。

3. 提案手法

3.1 ファーの輝度計算

本研究では文献[1]の手法によって表現されるファーで覆われた物体に対し、PRT 法によってライティング処理を行う。以下、提案法について詳しく説明する。その際、考慮すべき点は重ねているファーのポリゴンの処理をどのように計算するかである。

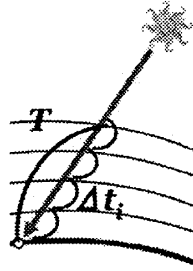
2.1 節で述べたように、ファーは物体表面の法線方向に一定間隔に配置した半透明ポリゴンを用いて表現されている。光源からの光は、これらの半透明ポリゴンの各頂点に到達するまでに、光源と頂点の間に存在するファーによって減衰する。これは次式によって表現される。

$$L \cdot \exp(-\alpha \tau)$$

ただし、 L は光源の輝度値、 α は減衰係数である。 τ は以下の式によって求められる。

$$\tau = \int_0^T \rho(t) dt = \sum_{i=0}^m \rho_i(t_i) \cdot \Delta t_i$$

ただし、図2に示すように、 T は各頂点から最上層のポリゴンまでの距離、 m は各頂点と光源の間に存在する半透明ポリゴンの枚数、 Δt_i は各ポリゴンとの距離、 ρ は密度を表す。本研究では、 ρ はポリゴンとの交点が毛の部分であれば1、それ以外は0としている。



ファアの形状が変化しなければ、 τ も変化しない。したがって τ も前処理として計算することが可能となり、次式のように、2.2節で述べた M を求める式(2)の中に組み込むことができる。

$$M_i^m = \frac{1}{\pi} \int Y_i^m(\omega) V(\omega) \max(\omega \cdot n, 0) \exp(-\alpha \tau) d\omega$$

3.2 多項式近似によるデータ圧縮

ファアを考慮することでポリゴン数が増加するため、3.1で述べた方法では前処理で保持したデータや事前計算の時間が膨大になるという問題が生じる。そこで多項式近似を用いることで、データの圧縮を図る。

図3は各層(1~16層)における M_i^m の値の散布図を多項式(3次)によって近似したグラフである。多項式によって近似することで各層の M_i^m の値を保持するのではなく、近似式の各係数を保持しておく。近似式の係数は最小二乗法により算出する。このとき、我々の実験によれば、全ての層での M_i^m を用いる必要はなく、ある程度少ない層(例えば1,4,7,10,13,16層目など)の値のみを用いても十分な精度が得られることを確認している。これにより、事前計算の高速化も期待できる。

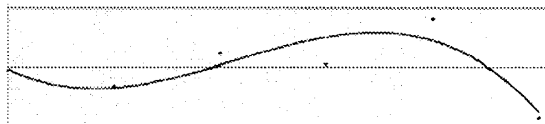


図3：最小二乗法(多項式近似)による近似

4. 実装結果

4.1 多項式近似の妥当性

3.2節で述べた多項式近似を用いた場合と用いない場合で画像を比較する。比較に用いた物体は単純な平面で表1のとおりである。表1において、サンプル数は式(2)を数値積分するためのサンプル点数でありこれが多いほど正確な計算が可能になるが、その分計算時間が膨大する。

表1：実験データ

	図4	図5
頂点数	121	1000
面数	100	1960
サンプル数	200	800
層の数	16	16

表2から多項式近似を用いた場合のほうがデータ量は約1/4まで減り、前処理の計算時間が大幅に削減されていることがわかる。さらに図4から視覚的なクオリティもほとんど変化がないとみられ、多項式近似を用いたデータ量の圧縮とある程度の少ない層による事前計算は非常に有効性が高いといえる。

4.2 Stanford Bunnyによる出力結果

提案手法を用いてBunnyの表示を行った。実験環境は表1のとおりである。データ量や事前計算時間は表2のよう

表2：データ量、計算時間

図4	多項式近似なし	多項式近似	図5	多項式近似
データ量	3MB	788KB	データ量	11.5MB
計算時間	2.3h	70s	計算時間	13.5h

になり、頂点数が増加することでデータ量や計算時間は膨大になっており、今後改善する必要がある。出力結果は図5のようになった。また、環境マップを変更した場合やファアや物体の表面の色を変化させた場合でも、前計算をやり直すことなくリアルタイムに画像の生成を行うことが可能である(図5右参照)。

5. まとめと今後の課題

本研究は環境光源によって照射されたファアをリアルタイムにレンダリングする手法を提案した。このとき、前計算データが膨大になるという問題点を多項式近似によって圧縮することで解決する手法を提案した。最小二乗法で求めた係数を保持しておくことでデータ量を圧縮することができた。提案法では光源の変更や回転、物体やファアの色は自由に操作することができるが、ファアを含めたポリゴンの位置情報を変更することができないため、ファアの長さやシェル・テクスチャの変更はできない。今後の課題としてはインタラクティブ性の向上や、ファアの毛の向きの考慮、剛体の物体のアニメーションなどが挙げられる。

参考文献

- [1] J.Lengyel et al, Real-time Fur over Arbitrary Surfaces, Symposium on Interactive 3D Graphics 2001, 227-232.
- [2] Peter-Pike Sloan et al, Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments, SIGGRAPH2002, 527-536.

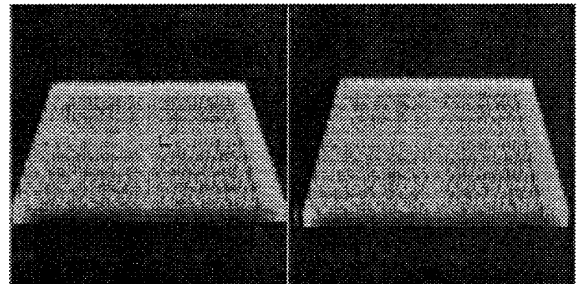


図4：多項式近似を用いた場合(右)と用いない場合(左)による出力結果



図5：Stanford Bunnyによる出力結果