

異方性 3 次元テクスチャのポイントベースレンダリング

伊藤 智也[†] 藤本 忠博^{††} 村岡 一信^{†††} 千葉 則茂^{††}八戸工業大学[†] 岩手大学^{††} 東北工業大学^{†††}

1. はじめに

コンピュータグラフィックスにおいて、微細な表現構造を持つ物体の表現は、データ量、計算時間および画像品質の点において非常に困難な問題である。3次元CGにおいて一般的に用いられるポリゴンによる幾何モデルを用いた場合には、データ量、計算量共に膨大となり、さらにレンダリングにおいてピクセル単位でサンプリングを行う場合、深刻なエイリアシングが発生し、画質を低下させてしまうことがある。これを解決する方法として、3次元構造をもつテクスチャを用いる手法が提案されており、画像の解像度に対して微細な表面構造をもつ物体の表現に有効である。

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、投影面積当たりの密度の異方性、微小面に対するシェーディング法などが必要である。それらの問題を解決する方法の一つとして、異方性3次元テクスチャ法[1]が村岡らによって開発されている。しかし、レンダリングにはレイトレーシング型のポリウムレンダリングを適用しているため、ボクセル内の複数の微小面に対してそれぞれ視線・光線等との計算を行う必要がある。したがって、与えられた幾何情報によっては、データ量と同様に計算量も膨大となる。

本論文では、異方性3次元テクスチャのレンダリング手法としてレイトレーシング型のレンダリング法に対して、原理的にも簡単で、さらにグラフィックハードウェアを活用することも可能なポイントベースによるレンダリング法を提案する。

2. 異方性 3 次元テクスチャ法

2.1 異方性 3 次元テクスチャの表現

本手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（図1）。ボクセルには微小面の集まりそのものを記述するのではなく、

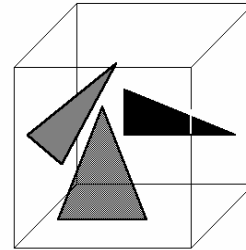


図1 ボクセルに含まれる微小面

その特徴を示す仮想的な球体を考え、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。

2.2 仮想球体の登録アルゴリズム

ポリゴンの微細な形状を表現するためには、ボクセル内の仮想球体の表面上に必要な情報を登録しなければならない。ここで仮想球体への情報の登録アルゴリズムについて述べる。

Step1: 微細な形状をポリゴンで作成する。

Step2: 作成したポリゴンをボクセルの範囲でクリッピングし、サブポリゴンと呼ぶクリッピングされたポリゴン(これがボクセル内の微小面となる)をそのボクセルに登録する。

Step3: 各ボクセルに密度球、陰影球を用意する。

Step4: 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。

Step3における情報登録の詳細を次に示す。

(1) 密度球、陰影球の表面上をランダムにサンプリングした点を生成し、それら各点 p をクリアする。

(2) ボクセルに含まれるサブポリゴンの全てにおいて、次の作業を行う。

(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 N 、相対面積を S 、密度球の表面上における点 p の単位法線ベクトルを N_p とすると、密度球上の全ての点 p について、次式で与えられる値 S_p を加える。ここで、 θ は N と N_p との成す角である。

“Point-based Rendering Method for An Anisotropic Three-dimensional Texture” by Tomoya ITO (Hachinohe Institute of Technology), Tadahiro FUJIMOTO (Iwate University), Kazunobu MURAOKA (Tohoku Institute of Technology) and Norishige CHIBA (Iwate University).

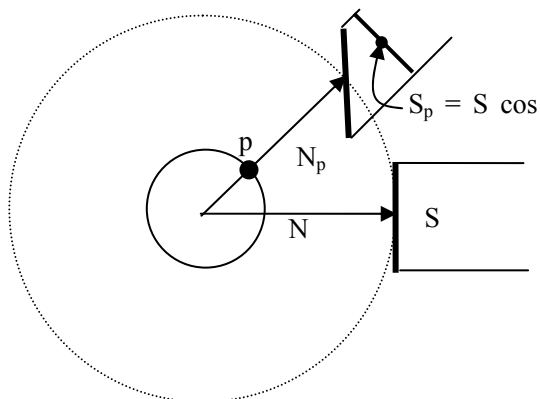


図2 密度球への登録

$$S_p = S |\cos \phi| = S |N \cdot N_p| \quad (1)$$

(2-3) サブポリゴンの N と同じ法線を持つ陰影球上の点 p (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、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。
 (1), (2) の作業により仮想球体に情報を登録した後、サブポリゴンは廃棄する。このように形状データを無くしてしまうことによって、大幅なデータ量の削減が期待できる。

3. 異方性3次元テクスチャのポイントベースレンダリング

3.1 ポイントデータの作成

仮想球体のポイントベースによるレンダリングのためのポイント(点群)データの作成アルゴリズムについて述べる。

ポイントデータは、レンダリングの最小単位となるために、サンプリング数が多いほど精度は向上すると考えられるが、ここでは仮想球体に登録された密度球上の点の中から、サブポリゴンの断片の面積に比例した面積の点を数個抽出する。また、それらの点にはそれぞれのサブポリゴンの法線を与える。

3.2 ポイントデータのレンダリング

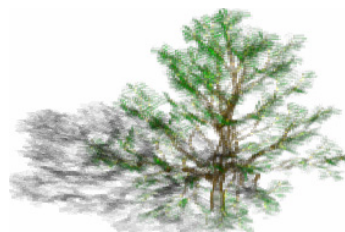
ポイントベースレンダリングの手法として、現在良く用いられている基本的な手法の一つにスプラッティング(splatting)[2]がある。スプラッティングとは、ポイントの描画において、各点に対し、スプラットと呼ばれる四角形や円、楕円を定義し、重複させることで点間の隙間を埋める方法である。本手法では、法線を持つポイントデータをスプラッティング法によってレンダリングを行った。



(a)幾何モデルによるレンダリング



(b)レイトレーシングによるボリュームレンダリング



(c)本手法によるレンダリング

図3 レンダリング例

4. 実行結果

図3に幾何モデルとして樹木を用いた場合のレンダリング結果を示す。画像のボクセル数は $128 \times 128 \times 128$ である。3次元テクスチャを用いた結果(b)はエイリアシングの少ない画像が得られていることが確認できる。(c)は、(b)のデータから点群データを生成し、スプラットとして四角形を用いてレンダリングした例である。

5. まとめ

ポリゴンによる幾何モデルから生成された3次元テクスチャをポイントベースレンダリングによって、データ量を削減してレンダリングを行った。

今後の課題としては、視点に依存した点群データの効率的な生成法や、スプラッティング以外のポイントレンダリング手法の適用などが挙げられる。

参考文献

- [1]村岡, 千葉: “異方性3次元テクスチャ法”, 画像電子学会誌, 28巻, 第2号, pp.131-139, 1999).
- [2]藤本, 今野, 千葉: “ポイントグラフィックス概説” 芸術科学会論文誌, Vol13, No.1: pp.8-21, 2004.