

## テクスチャマッピングを用いた球の描画法

7P-8

中村 英史 川瀬 桂

日本アイ・ビー・エム株式会社 東京基礎研究所

nakamu@trl.ibm.co.jp, kawase@trl.ibm.co.jp

## 1 概説

インタラクティブな3次元グラフィックスシステムにおいて、球を描画することはコストの高い操作である。分子CADなど多数の球を描画するインタラクティブなアプリケーションでは高速な球の描画が極めて重要である。

本稿は、テクスチャマッピングの手法を用いてハードウェアによりLightingのされた球を高速に描画する機構である。

## 2 従来の方法と問題点

インタラクティブな3次元グラフィックスにおいて複数の球をz-Buffer<sup>1</sup>法を用いて描画するには、まず、球を十分滑らかになるまで多数の細かい三角形に分割する。三角形内部の各点の色を決定するには各三角形の各頂点でLightingの計算を行ない色を決定し、内部の点の色は頂点の色を補間(Gouroud Shading<sup>1</sup>)したり、各頂点に与えられた法線方向を補間しながら三角形の内部の各点でLightingの計算を行なってその点の色を決定(Phong Shading<sup>1</sup>)する、などの方法がある。こうして各球面上の点の色が決定したらz-Buffer法によりスクリーン上の点の色を決定する。

Lightingの計算自体演算量が多いのに加え、通常1つの球は100~1000くらいの三角形に分割する必要があるので、Scene全体での三角形の量は極めて多くなってしまい、多大な計算量が要求される。

## 3 テクスチャマッピングを用いた球描画法

本稿では、従来の方法がある仮定の基で簡略化し、分子CADでDNAやタンパク質の分子の3次元構造のシミュレーション結果の表示で必要とされる多数の球の高速な描画に必要な色(Lighting)と奥行き(z-Buffer)の情報を2次元配列に格納(Image化)して、多数の三角形を描画する代わりに2次元の表をLookupすることにより重なりをも正しく処理して複数の球を描画できることを報告する。

## 3.1 色情報のImage化

分子CADにおいてDNAやタンパク質の分子模型の全体像を見る場合は、視点から分子模型までの距離が分子模型の大きさよりもはるかに長いので、その結果視野が狭くなり各球を見る方向はほぼ一定とみなせる。また視野が狭いために各球の投影変換は平衡投影に近似できる。すなわち、分子を構成する各球の中心座標に関する座標変換は透視投影変換を行なうが、各球毎には

- 視点は無限遠点にある → 並行光線。
- 全ての光源は無限遠点にある → 一定の視線ベクトル。

と仮定して処理しても事実上問題はない。

このような簡略化を行なうと3次元空間中の任意の物体はその位置に依存することなく、表面の材質と各光源の色の情報の他に各頂点の法線の方向のみで原理的に色を決定することができる。従って球のようにどの方向から見ても形が相似になるものに関しては上記の仮定の基では複雑なLightingの計算を一度だけしておけばその計算結果を他の球のLightingの計算に流用することができる。

さらに視点とスクリーンの中心を結ぶ視線がスクリーンに垂直とみなせるときは、各球の平衡投影も直軸測投影となる。結局、スクリーンの中心に見える球の反射光を計算してImage化しておく。球が複数種類あるときは各球の種類毎に反射光の値をImage化しておく。

## 3.2 奥行き情報のImage化

3次元の前後関係を描画順によらず忠実に表現するためにz-Buffer法を行なう必要がある。前述の仮定の基では各球の表面の点の奥行き情報(z-value)は球の中心点のz-valueと、球の中心を含み中心と原点に垂直な平面からの球面上の各点までの距離が分かれば簡単に計算できる。全ての球は単位球の半径倍だから単位球のz-valueを前述の色情報と同様にImage化しておき、各球のz-valueを求める時には球の中心のz-valueと球の半径を用いて適当な変換(後述)を行えば良い。

## 3.3 テクスチャマッピングによる球の描画

上記2項で用意した色情報と奥行き情報のImageを用いて視点座標系において中心が $P_0(x_0, y_0, z_0)$ で半径が $R$ の球を描画する。先ず1辺が $2R$ の長さの正方形

を中心を  $P_0$  に重ね、 $OP_0$  ( $O$ : 視点) に垂直になるよう  
におく。次にこの正方形に前述の仮定の基で適当な座標  
変換を行ない、スクリーン座標系内の正方形 (Skelton)  
に変換する。正方形の1つの頂点を原点として各々の  
辺に沿って  $[0,0, 1,0]$  で Parametrize して正方形内の点  
を2次元座標  $(s, t)$  で表す。この  $(s, t)$  座標に対応する  
色情報と奥行き情報を、各々が格納されている Image  
からテクスチャマッピング<sup>1</sup> によって Lookup する。こ  
のように Lookup された色情報と奥行き情報を  $C(s,t)$ 、  
 $D(s,t)$  と書くことにする。さらに各々についての次の  
処理が施される。

- 色の決定。  
色情報に関しては Image が2次元の配列なので  
正面から見た球の外部の点も含んでいることに注  
意しなければならない。この点については描画時  
に描いてはいけなないので、球の内部と外部を区別  
するための「型抜き」のための1ビットの表も持っ  
ておく。これは色情報の Image の1成分として  
含めることができ、この型抜き情報を考慮すれば  
 $C(s,t)$  が描画すべき色となる。

- z-Buffer 法。  
z-Buffer 法に必要なデータは球面上の点の z-value:  
 $Z_p$  である。これは球の中心の z-value:  $z_0$ 、球の  
半径  $R$ 、視点からの距離  $d_0$  と対応する  $D(s,t)$  を  
用いて

$$z_{sphere} = z_0 + \frac{R \times D(s,t)}{d_0} \equiv z_0 + D(s,t) \times \left(\frac{R}{d_0}\right)$$

で求められる。1つの Scene について  $R/d_0$  は  
各球毎に1回ずつ計算しておけばよい。

#### 4 ハードウェアへの実装とパフォーマンスの 向上

本稿の方法によるパフォーマンスの向上は、球の三角  
形分割とテクスチャマッピングの各々の処理に要するコ  
ストから概算できる。テクスチャマッピングで1つの球  
を描画するには、正方形に、色情報と z-value の2枚の  
Image をテクスチャマッピングするので必要な三角形は  
2つである。よってもしハードウェアがテクスチャマッ  
ピングによって得られた z-value を一次変換できる機構  
を持っていれば、劇的なパフォーマンスの向上が期待で  
きる。通常テクスチャマッピングは Image から Lookup  
された複数のデータ (texel value) に適当な Filtering  
を施して最終的な色決定をするので、Filtering の計算も  
含んだテクスチャマッピングの機構がハードウェアに実  
装されていれば、z-value を得るのに必要な加算器と乗  
算器は既に持っている。大幅にリソースを追加する必要  
はなく、 $z_0$  と  $R/d_0$  を通せるようにデータパスを変更  
するだけでよい。

#### 5 実行例

原子の個数がおおよそ 300 の分子を本稿の方法によっ  
て描画した例を図1に示す。

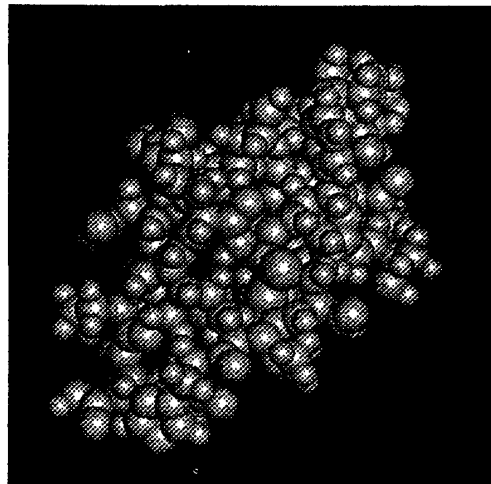


図1:

#### 6 まとめと今後の展望

本稿は従来の方法に比べて

- 球を十分に細かい三角形に分割するオーバーヘッ  
ドがない。
- 各ピクセル毎の Lighting の計算:  $C(s,t)$  はシー  
ンで一度だけ行なえばよい。
- z-Buffer 法で使われる z-value:  $D(s,t)$  はシー  
ンに関係なく一度だけ計算すればよい。
- 各球の相互位置関係は透視投影を施し、各球はそ  
の z-value の逆数によってスケールリングする。
- 球の描画はテクスチャマッピングによって実行す  
る。

という処理によって多数の球を立体感を失うことなく、  
はるかに高速に描画することを実現している。

本稿で報告した方法は他の形、例えばパイプ状の物体  
にも適用できる。

#### 参考文献

- [1] Computer Graphics, PRINCIPLES AND PRAC-  
TICE, Foley, van Dam, Feiner and Hughes, SEC-  
OND EDITION, ADDISON WESLEY, 1992