6Y-4

自己反射を考慮した鏡面反射物体の高速レンダリング

中谷　聡志[*1]　　岩崎　慶[*2]　　高木　佐恵子[*2]　　吉本　富士市[*2]

和歌山大学大学院システム工学研究科[*1]　　和歌山大学システム工学部[*2]

1　はじめに

鏡や金属に代表される鏡面反射物体のレンダリングは重要な研究課題として広く研究されている．鏡面反射物体をレンダリングするためには，周囲の環境の映り込みを考慮しなければならない．映り込みを表現するためには，物体の各頂点で視線方向に対する反射方向を計算する必要がある．Whitted が提案したレイトレーシング法[1]は鏡面反射物体の写実的な画像を生成できるが，計算コストが高い．Blinn らが提案した環境マッピング法[2]は高速にレンダリングできるが，近隣物体や物体自身の映り込みは表現できない．また，Kalos らは近隣の物体の映り込みを考慮した環境マッピング法を提案した[3]．しかしながらこの手法では物体自身が映り込む効果は表現できない．そこで本稿は自己反射を考慮した高速なレンダリング法を提案する．

2　提案手法

提案手法は，大きく分けて前計算処理，レンダリング処理の二つの処理で構成される．映り込みを計算するためには視線に沿ってレイを追跡する必要があるが，これは計算コストが高い．提案手法では，前計算で離散的な方向においてあらかじめレイを追跡しておいて，その結果を用いて補間し反射方向を高速に求める．ここで，映り込む周囲の環境は環境マップで与えられるものとする．

2.1　前計算処理

前計算では，各頂点の離散的な入射方向に対し，自己反射を考慮して最終的に物体から反射して出る光の方向を求める．これをレイトレーシングで行う．処理の流れは以下の通りである．(1)物体の各頂点で離散的な入射方向(サンプルレイと呼ぶ)を生成，(2)各サンプルレイを用いてレイトレーシング，(3)レイトレーシングによって算出した，物体を最終的に放出したレイ(サンプル反射レイと呼ぶ，図１参照)をテーブルに保存する．

2.1.1　サンプルレイの生成

サンプルレイの生成はまず物体上の全頂点で仮想的な半球を設定し，半球内で頂点に向かう様々な方向のレイを生成する．サンプルレイはこの頂点における法線・接線・従法線を用いてこの頂点を中心とする局所座標系で表現する．

2.1.2　サンプル反射レイの計算

生成した各サンプルレイに対してレイトレーシングを行い，サンプル反射レイを計算する．(1)レイが物体と衝突しない場合，(2)反射回数が閾値を超えた場合の２つをレイトレーシングの終了条件とする．レイトレーシングの計算は全頂点，全サンプルレイで行われる．

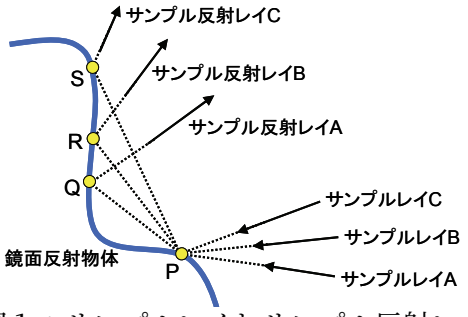


図１：サンプルレイとサンプル反射レイ

2.1.3　前計算テーブルへの保存

レイトレーシングで計算したサンプル反射レイは前計算テーブルに保存する．しかしながら，全サンプル反射レイのデータを保存するとデータ量が非常に大きくなる．ここで，反射回数が１回のサンプル反射レイに関しては，視線方向と法線方向からレンダリング時に簡単に計算できることから，前計算テーブルに保存するサンプル反射レイのデータは，反射回数が２回以上のデータとした．また，方向を離散化して，離散化した方向ベクトルにインデックス付けを行う．そして，サンプル反射レイの方向を前計算テーブルに保存する代わりに，サンプル反射レイに最も近い離散化した方向ベクトルのインデックスを保存することで，前計算テーブルのデータ量を削減した．

Fast Rendering of specular object taking into account self-reflection

*1 Satoshi Nakatani, Graduate School of Systems Engineering, Wakayama University

*2 Kei Iwasaki, Saeko Takagi and Fujiichi Yoshimoto, Faculty of Systems Engineering, Wakayama University

2.2　レンダリング処理

レンダリング処理では前計算によって求めたデータを用いて，物体から最終的に反射して放出する反射方向を線形補間によって算出する．処理の流れは以下の通りである．(1)視線方向に最も近い４本のサンプルレイ(近傍サンプルレイと呼ぶ)を算出，(2)前計算テーブルを参照して，近傍サンプルレイに対応するサンプル反射レイを取得，(3)サンプル反射レイを線形補間，(4)線形補間して求まったレイ(視線反射レイ)を用いてレンダリングを行う．

2.2.1　視線反射レイの計算

視点から頂点へ向かう視線方向に最も近い４つの近傍サンプルレイを計算する．その近傍サンプルレイに対応するサンプル反射レイのデータを前計算テーブルから取得し，線形補間を行う．ここで，４つのサンプル反射レイの反射回数が全て一致しているとは限らない．

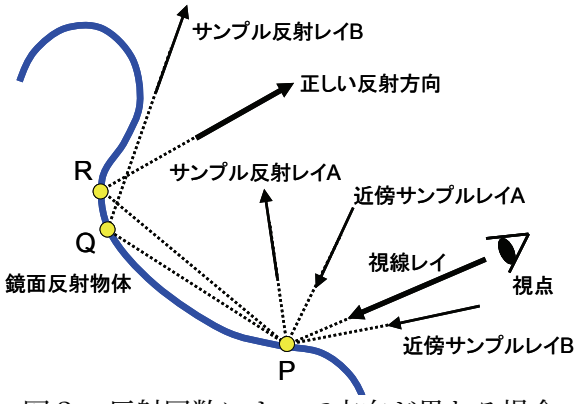


図２：反射回数によって方向が異なる場合

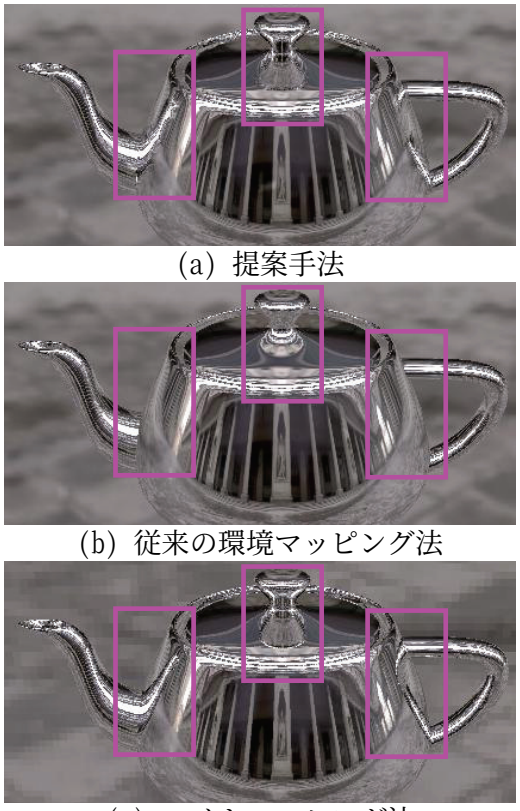
図２に示すように，反射回数が異なるサンプル反射レイは方向が大きく異なることがあり，誤った方向に補間される．よって本稿では，４つのサンプル反射レイの反射回数が同じ場合と異なる場合で視線反射レイを区別して計算する．

2.2.2　レンダリング

各頂点の視線反射レイのデータを用いて周囲の環境をマッピングする．１つの頂点における視線反射レイの反射回数が全て同じならばその視線反射レイの方向を用いてマッピングする．もし異なる場合，それぞれの反射回数での重みと視線反射レイの方向の環境マップの輝度データとの積をそれぞれ足し合わせる．

３　結果とまとめ

頂点数が 12,801 のティーポットでの，提案手法，従来の環境マッピング法，レイトレーシング法による結果画像を図３に示す．



(a) 提案手法

(b) 従来の環境マッピング法

(c) レイトレーシング法

図３：結果画像例

サンプルレイの数は 257 とし，方位角と仰角をそれぞれ等間隔に分割して離散化したサンプル反射レイの方向数は 32,514 とした．前計算テーブルの容量は 4.90MB，描画時間は 0.06 秒であった．計算環境は CPU Pentium4 3.40GHz，GPU GeForce 7800GTX である．

従来の環境マッピングでは表現できなかった自己反射がティーポットのふたや注ぎ口付近で見ることができ，またインタラクティブな速度で描画することができた．今後の課題として，自己反射の部分に視点を近づけたときに生じるエリアシングの改善が挙げられる．

参考文献

[1] Turner Whitted, An Improved Illumination Model for Shaded Display, Communications of the ACM, Vol. 23, No. 6, pp.343-349, 1980.

[2] James F.Blinn, et.al, Texture and Reflection in Computer Generated Images, Communications of the ACM, Vol.19, No.10, pp.542-547, 1976.

[3] Laszlo Szirmay-Kalos, et.al, Approximate Ray-Tracing on the GPU with Distance Impostors, Computer Graphics Forum, Vol. 24, No. 3, pp.685-704, 2005.