# 自己反射を考慮した鏡面反射物体のGPUレイトレーシング法

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

鏡や金属などの鏡面反射物体のレンダリングは重要な研究課題の一つ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する必要がある．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，Hakuraらは鏡面反射物体を様々な視点からレンダリングした画像PEMs (Parameterized Environment Maps)を作成し，PEMsを補間することで，視点を変えても高速に鏡面反射物体を描画する手法を提案した[3]．しかしながらこの方法は鏡面反射物体を移動させることや回転させることができない．

そこで本稿は鏡面反射物体自身が映り込む複数回の反射（自己反射と呼ぶ）を考慮した高速なレンダリング法を提案する．提案手法は，GPUを用いてフラグメント単位でレイトレーシングを行うことにより，自己反射を考慮した鏡面反射物体の高精細な画像を生成することができる．

## 2　提案手法

提案手法は前計算処理とレンダリング処理の二つの処理で構成される．自己反射を考慮して鏡面反射物体をレンダリングするためには，視線の反射レイと物体表面との交点を計算する必要がある．提案手法は，視線が物体表面で反射した点からレイを少しずつ反射方向に進め交点を反復的に探索する．しかし，交点と反射した点との距離が長い場合，多くの反復回数が必要となり，交点探索の計算コストが大きい．そこで，離散的な方向において頂点と物体表面との距離を前計算しておく．その結果を用いて探索の開始点を決定することにより，探索回数を減少させる．ここで，映り込む周囲の環境は環境マップで与えることとする．

### 2.1　前計算処理

前計算では，離散的な方向について物体表面と各頂点との距離を計算し，前計算テーブルとして保存する．離散的な方向のレイをサンプルレイと呼ぶ．

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，半球上の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．生成した各サンプルレイに対して物体表面との交点を計算し，頂点と交点の距離を計算する．距離計算は全頂点，全サンプルレイで行われる．

計算した距離値は前計算テーブルに保存する．ここで，距離値を全て保存するとデータ量が非常に大きくなる．探索開始点の決定には厳密な距離値を必要としないので，サンプルレイの仮想半球を方位角と仰角において等間隔に分割し，分割した範囲内に含まれるサンプルレイに対応する距離値の最小値と最大値を保存する．これにより，前計算テーブルのデータ量を削減した．

### 2.2　レンダリング処理

レンダリング時の処理では，反射レイを少しずつ進めて交点を探索することで，GPUでレイトレーシングを行う．処理の流れは，(1)浮動小数点テクスチャに各種データを格納し，(2)そのテクスチャを用いて反射レイと物体表面との交点を探索する．(1)は，FBO(Framebuffer Object)を使用し，スクリーンの各ピクセルに対応した物体表面上の点の座標，法線，前計算データをスクリーンサイズの浮動小数点テクスチャに格納する．このテクスチャを用いて交点を探索する．ここで，視点から見えない物体表面の情報がないので，交点探索に失敗する可能性がある（図1）．そこで，提案手法ではこの問題を解決するためにDepth peeling[4]を導入する．Depth peelingを用いると，深度別に複数枚の浮動小数点テクスチャを作成することができるので，視点から見えない部分の反射計算を行うことができる．

---

GPU ray-tracing method of specular object considering 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

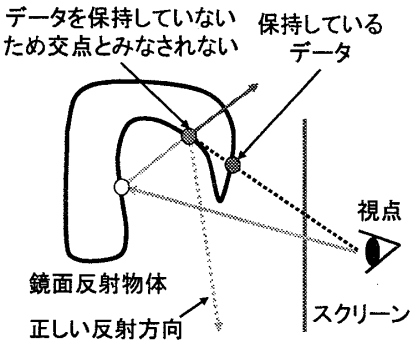


図１：交点探索の失敗例

(2)の交点探索は，まず視線に対する反射方向を計算し，前計算データを用いて探索開始点を決定する．そして探索開始点から反射方向にレイをδ分進める．交点の判定は，レイの指す位置と物体表面との距離dと閾値εとの比較で行う．物体表面の位置は座標情報を格納しているテクスチャを参照すれば取得できるので，距離dが計算できる．閾値εよりdの方が大きければδ分レイを進めて再び交点判定を行う．閾値εより距離dの方が小さければその位置を交点とし，再び反射方向を計算し交点を探索する（図２）．

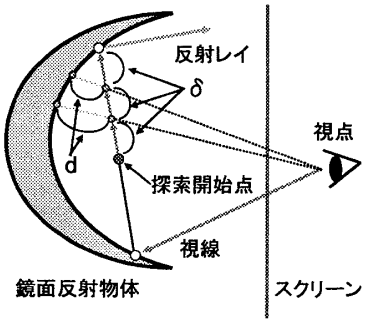
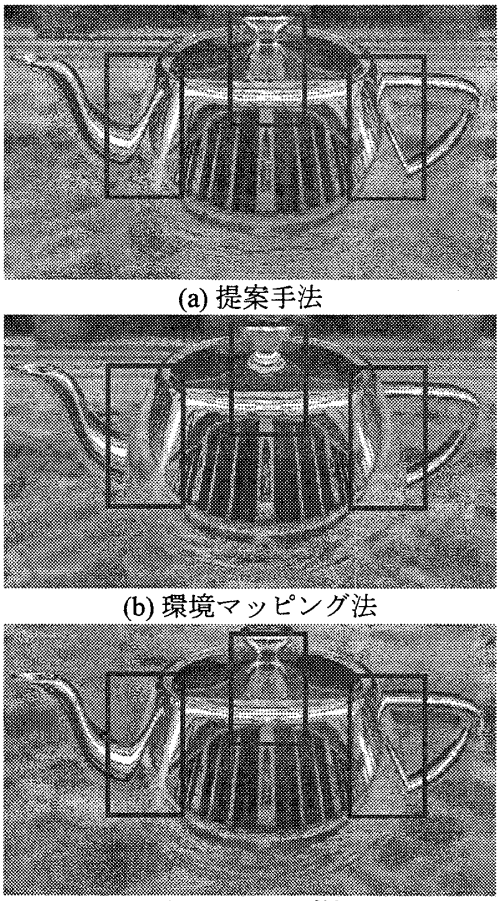


図２：交点探索

交点探索により物体表面から最終的に放射した反射方向を計算する．その反射方向を用いて環境マップを参照して色を取得し，スクリーンに描画する．

## 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，環境マッピング法，CPU でのレイトレーシング法による結果画像を図３に示す．サンプルレイの数は 1,264 とし，方位角と仰角の分割数はともに４とした．前計算に要した時間は 17 分 24 秒，前計算テーブルの容量は 1.56MB であった．スクリーンサイズは 640×480 とし，描画時間は 0.083 秒であった．計算環境は CPU Pentium4 3.40GHz, GPU GeForce8800GTX である．


(a) 提案手法

(b) 環境マッピング法

(c)レイトレーシング法 (CPU)

図３：結果画像例

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などで見ることができ，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，相互反射への拡張が考えられる．

## 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] Ziyad S. Hakura et al., Parameterized Environment Maps, Proc. Symposium on Interactive 3D graphics, pp.203-208, 2001.

[4] Cass Everitt, Interactive order-independent transparency, Technical report, NVIDIA, 2001.