NAIST-IS-MT0151027

修士論文

多重スケール環境マップによる
粗さが不均一な材質への写り込みの高速レンダリング

奥村 文洋

2003年2月7日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士 (工学) 授与の要件として提出した修士論文である。

奥村 文洋

審査委員： 横矢 直和 教授
千原 國宏 教授
山澤 一誠 助教授

多重スケール環境マップによる
粗さが不均一な材質への写り込みの高速レンダリング*

奥村 文洋

内容梗概

近年，コンピュータグラフィックスの分野において物体の表現能力を高めるための研究がなされている．これらの技術は，仮想環境内に仮想物体を挿入した場合の物体の見え方を写実的に表現するためなど，様々な用途に用いられている．中でも，周囲の環境の写り込みを表現する環境マッピングとよばれる手法は幅広い分野で利用されている．この手法は高速に周囲の写りこみを表現するために開発されたものであるが，従来，表面の粗さが不均一な材質への写り込みの高速なレンダリングは困難であった．そこで本研究では，オフライン処理とハードウェア処理を組み合わせた写り込みの高速レンダリング手法を提案する．本手法は，入力となる環境マップに対してフィルタ処理を行うことで，複数の粗さに応じた環境マップ（多重スケール環境マップ）を予め作成する．このとき，フィルタ処理を単純に実行すると膨大な計算量になる．そこで計算時間を短縮するために，ピラミッド画像と頂点数の異なる測地ドーム (Geodesic Dome) を用いて計算の高速化を行う．さらに，フィルタの影響が大きい画素のみを計算対象とし，フィルタの重みをテーブルとして保持することで高速化を行う．そして，多重スケール環境マップをテクスチャとして用い，粗さが不均一な物体への写り込みを描画する．このとき，グラフィクスハードウェアに搭載されている 3D テクスチャマッピング機能を利用することで高速な描画を行う．実験では，提案手法による描画結果と Phong モデルによる描画結果を比較し，提案手法において粗さが不均一な

* 奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, NAIST-IS-MT0151027, 2003 年 2 月 7 日.

材質に対する写り込みを良く近似できていることを示す．さらに，高速化を行わない多重スケール環境マップと高速化を行った多重スケール環境マップを比較することで高速化の効果を検証する．最後に，全方位マルチカメラシステムで取得した全天球画像を環境マップとして用いて仮想環境内に仮想物体を配置した場合と環境を動画に拡張した場合の写り込みのレンダリング結果を示し，提案手法の有効性を示す．

キーワード

コンピュータグラフィックス，実時間レンダリング，環境マッピング，材質の粗さ，Phong モデル，全方位実写画像

Fast Rendering of Reflections of Non-uniform Surfaces Using a Multi-Scale Environment Map*

Bunyou Okumura

Abstract

Various techniques are used for rendering virtual objects in a virtualized real scene under arbitrary illumination conditions. Especially, an environment mapping technique was developed to render glossy objects in real time. However, it is difficult to render reflections in non-uniform roughness surfaces in real time. In this paper, we propose a real time rendering technique for such materials. The proposed technique is based on combining off-line processing and hardware rendering. In our method, a filtering of an omnidirectional image is performed to create an environment map (multi-scale environment map) according to multiple roughness of surfaces. However, there is a problem that the filter processing requires a huge amount of calculation. Therefore, we employ the pyramidal image data structure and multi-scale geodesic domes to accelerate computation. Furthermore, the filter processing is performed only on pixels which significantly influence rendering results within omnidirectional images. Calculations are also accelerated by using a table which holds weights of the filter. Reflections in surfaces are then rendered in real time using the multi-scale environment map as a 3D texture with support of graphics hardware. In experiments, we show that the proposed method can well approximate Phong model, and that the technique

* Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT0151027, February 7, 2003.

for acceleration of filter processing is effective. Finally, we show some rendering results for glossy objects in virtualized environments which are constructed of omnidirectional images captured by an omnidirectional multi-camera system.

Keywords:

computer graphics, real-time rendering, environment mapping, roughness of surface, phong model, omnidirectional real image

目 次

図 目 次

表 目 次

1. はじめに

近年，コンピュータグラフィックス (CG) の分野において写実的な画像を作成するためのレンダリング手法に関する研究が多くなされている [1, 2]．一般に，CG の写実性を向上させるためには，現実環境で生じる影 [3, 4]，屈折 [5]，反射 [6, 7, 8] などの再現が重要である．中でも，写り込みは金属やプラスティック等の材質で生じ，現実環境の中で多数見られるため，写り込みを再現することは写実性を高める上で重要である．しかし，写り込みは相互反射によって生じる現象であるため，写り込みのレンダリングには計算時間が膨大になるという問題があった．

従来，写り込みを高速にレンダリングするための手法として環境マッピング [9] が提案されている．この手法は，相互反射を簡略化して扱うことで写り込みの高速なレンダリングを可能とした．しかし，材質の反射特性が考慮されていないため，対象とする材質が鏡のような滑らかな材質に限定され，様々な材質に応じた写り込みの表現が困難であるという問題があった．この問題を解決するため，様々な改良が環境マッピングに関して行われた [10, 11, 12, 13]．これらの従来研究において，反射特性を考慮した写り込みのレンダリングが可能になったが，いずれも単一の反射特性を想定している．このため，反射特性が不均一な材質から構成される物体のレンダリングには計算時間がかかるという問題がある．

本研究では，材質の反射特性のうち写り込みが特に影響を受ける物体表面の粗さに着目し，表面の粗さが不均一な材質に対して，粗さを考慮した写り込みを高速にレンダリングすることを目的とする．アプローチとして，オフライン処理と実時間レンダリングを組み合わせる．まず，オフライン処理として環境マップに対してあらかじめフィルタ処理を行い，材質の粗さに応じた複数の環境マップを作成し，それらをボクセルデータとしてまとめて保持する．このとき，ボクセル空間の中心からの距離が材質の粗さに，中心からの方向が材質表面での正反射ベクトルの方向に対応するように格納する．こうすることで複数の粗さに応じた写り込み情報を保持した多重スケール環境マップが作成される．そして，レンダリングの際は多重スケール環境マップを 3D テクスチャとして利用することで，写り込みをテクスチャマッピングによってレンダリングする．このとき，粗さに応じたテクスチャ座標を設定することで，粗さが不均一な材質であっても写り込み

を一度にレンダリングすることが可能である．

以下，2 章において写り込みのレンダリングに関する従来研究と本研究の目的について述べ，3 章において多重スケール環境マップの詳細とその計算方法，写り込みのレンダリング手法，さらに多重スケール環境マップの計算の高速化手法について述べる．4 章では，提案手法の有効性を確認するために，写り込みのレンダリング結果，高速化の影響の検証，Phong モデルとの比較実験，動画像を利用したレンダリング結果を示す．そして最後に考察とまとめを述べる．

2. 写り込みのレンダリング手法に関する従来研究と本研究の方針

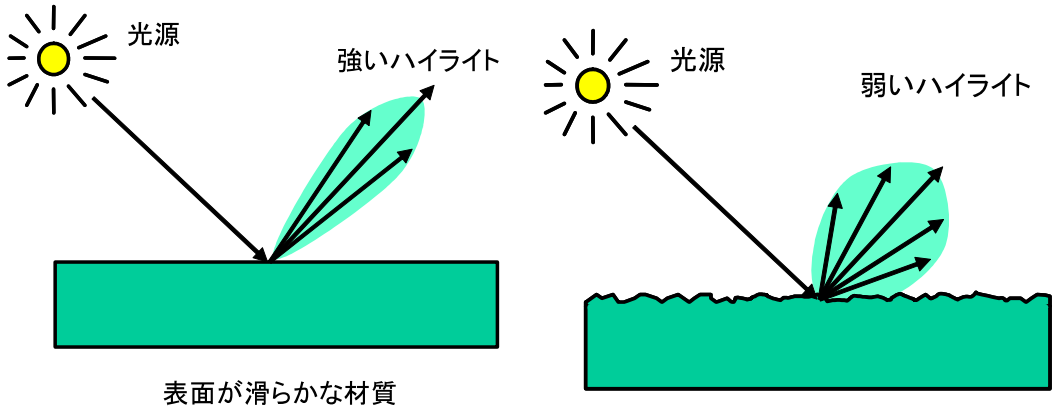
本章では，レンダリング手法に関する従来研究を写り込みの描画に重点を置いて概観する．まず，CG をレンダリングする上で重要な材質の反射特性に関する従来研究を示し，本研究で取り扱う物体表面の粗さに関して説明する．そして，写り込みが表現可能なレンダリング手法について従来研究を示し，最後に本研究の方針を述べる．

2.1 写り込みのレンダリング手法に関する従来研究

CG をレンダリングする際に重要となる要素として，材質の反射モデルの設定とレンダリング手法が挙げられる．これら 2 つはお互い密接に関係しているが，ここではまず，材質の反射特性と反射モデルについて示し，材質表面の粗さと写り込みの関係について述べる．そして，写り込みが表現可能なレンダリング手法について従来研究を示す．写り込みのレンダリング手法は，設定された反射モデルを元に相互反射の計算を行い写り込みを正確に計算する手法と，何らかの近似によって写り込みを簡易的に再現する手法との 2 つに分けることができる．前者の例としてはレイトレーシング法や様々な大域照明レンダリング手法があり，写実性が重視される場合に用いられる．そして，後者の例としては環境マッピング手法があり，レンダリング時間が重視される場合に用いられる．以下，それぞれについて詳細を述べる．

2.1.1 材質の反射特性と反射モデル

一般に材質によって生じる反射は拡散反射成分と鏡面反射成分に分けることができる．前者は反射輝度強度が視点位置に依存しない特徴があり，材質にいったん吸収された後，放射される光として扱われることが多い．後者は反射輝度強度が視点位置に依存する特徴があり，材質表面で反射された光として扱われることが多い．写り込みとは視点位置に依存する鏡面反射成分によって生じる現象であ



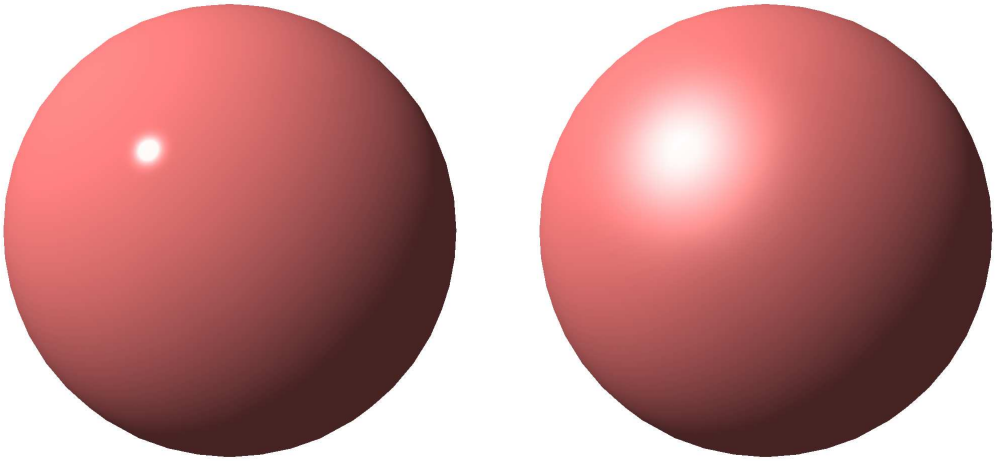
(a) 表面が滑らかな材質　　(b) 表面が粗い材質

図 1　滑らかな材質と粗い材質で生じる光の反射の違い

るため，写り込みのレンダリングを行う際には鏡面反射成分を取り扱うことが必要となる．鏡面反射成分に影響を与える要素として，材質表面の粗さが挙げられる．この場合の粗さとは，材質表面が微少面で構成されていると仮定した場合の，微少面の法線方向のばらつきとして考えられる．

図 1 に表面が表面が滑らかな材質 (a) と粗い材質 (b) で生じる光の反射の違いを示す．表面が滑らかな材質，すなわち微少面の法線方向のばらつきが少ない材質では，光の反射があまり広がらない．対して，表面が粗い材質では，ばらつきが大きいため光の反射が大きく広がっていることが分かる．図 2 は，粗さの異なる球体に対して平行光源によるハイライトをレンダリングした結果である．表面が滑らかな材質ではくっきりとしたハイライトが生じ，粗い材質ではぼやけたハイライトが生じていることが分かる．

このように，材質の粗さによる物体表面の光学的な表現として，物体表面上のある一点での光の入射と放射の関係を記述する方法がある．これは BRDF(Bidirectional Reflectance Distribution Function：双方向反射関数) モデル [14] と呼ばれ，この関数は，光源の方向と視点の方向が決定した際の材質の反射率を表現し，透過や屈折などを生じない材質の反射特性を表現することが可能となっている．BRDF では表現可能な材質の範囲が広い反面，一般には BRDF を簡単な式などで表現で



(a) 滑らかな材質でのハイライト　　(b) 粗い材質でのハイライト

図 2　滑らかな材質と粗い材質でのハイライトの違い

きない．このため，BRDF による反射特性の記述，保持や BRDF に基づいたレンダリングなどは困難である．この問題を解決するため，材質に対して様々な仮定をおくことでより計算が容易な反射モデルが提案された．

Lambertian モデル [15] は，材質が完全拡散反射面である場合をモデル化したものである．これは，光源から照らされた場合に，どの方向から見ても同じ輝度値を持つ材質を表現可能であり，最も単純な反射モデルとして利用されている．また，Phong モデル [16] は拡散反射成分と鏡面反射成分の反射特性を経験的に決められた計算式で記述したものである．Phong モデルは計算が比較的簡単なため，OpenGL 等で用いられている反射モデルである．Torrance-Sparrow モデル [17] は，材質の物理的な特性に基づいた反射モデルである．材質の表面が微少面で構成されていると仮定し，微少面の法線ベクトルの分布が正規分布に基づくとして，材質の粗さなどを定義した反射モデルとなっている．

このような材質の反射モデルを用いて写り込みのレンダリングを利用した研究例として，Debevec ら [18] の研究がある．これは，金属球で構成されるライトプローブを用いて環境の光源情報を推定し，推定された光源情報を元に仮想物体に

対して実環境に基づいた照明を施すため．この研究では，推定された光源情報と仮想物体の形状から BRDF に基づいた反射輝度を計算している．

また，Ramamoorthi ら [19] は環境中におかれた物体に対する拡散反射を効率的に表現するための手法を提案している．この研究では，環境中に配置された物体に対する BRDF に基づいた拡散反射の輝度は，球面調和関数の低次の項のみで表現可能であることを示している．この際の計算にも BRDF が用いられている．

2.1.2 相互反射を考慮した写り込みのレンダリング手法

以上に示した反射モデルは，物体表面での局所的な反射のみを扱っている．実環境では，局所的な反射だけでなく物体間などで相互反射が生じている．このため，写実的な CG を作成するためには相互反射も考慮する必要がある．このような相互反射を考慮した写り込みのレンダリング手法しては，レイトレーシング法 [20, 21] とパストレーシング法 [22] が有名である．以下にそれぞれの詳細を示す．

レイトレーシング法

材質に反射モデル設定し，物体間の相互反射を考慮した写り込みのレンダリング手法の代表的な例としてレイトレーシング法がある．特徴として，相互反射，影，屈折などを一括して取り扱うことが可能であり，特に写り込みに関しては，物体間で複数回反射する光も取り扱うことが可能である．しかし，拡散反射で生じる相互反射は取り扱うことができないという欠点があり，実用の点では拡散反射成分の相互反射をレンダリング可能なラジオシティ法 [23, 24] などと組み合わせられることが多い．

図 3 にレイトレーシング法における概略図を示す．まず，視点からの仮想的な光線 (レイ) を考え，物体とレイが衝突した点において，シーン中に定義された光源からの影響，反射による影響，屈折による影響をそれぞれ計算することで輝度値を決定する．また，反射や屈折の影響を考慮する際に再起的なレイを考えることで鏡のような物体に対する写り込みを表現することが可能である．

しかし，レイトレーシング法でレンダリングを行う際には，視点から少なくとも画素数分のレイを発生させ，それぞれに対して上記の処理を行う必

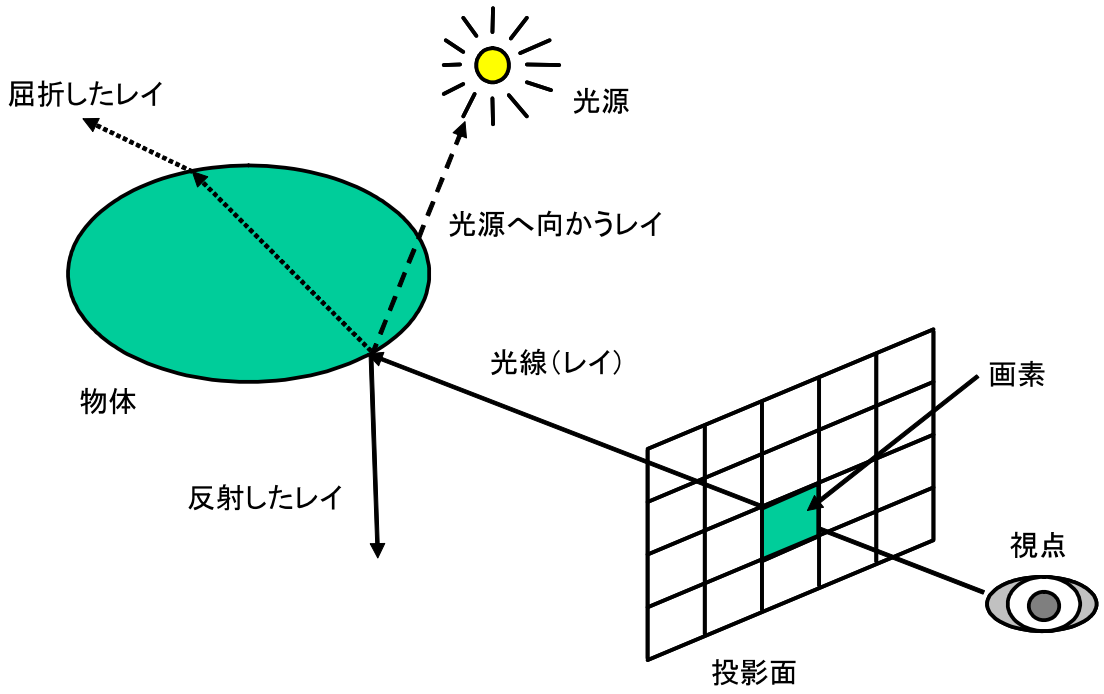


図 3 レイトレーシング法によるレンダリング

要がある．このとき，それぞれのレイは物体との衝突判定や再起的な反射の計算を行うため計算量が膨大となる．また，レイトレーシング法では反射の際に発生する再起的なレイを正反射方向にしか発生させない．このため，表面が粗い材質に特有の，光が正反射方向以外にも多数散乱することによって生じるぼやけたハイライトなどを再現することが困難である．この問題を解決する手法として，分散レイトレーシング法 [25] などが提案されている．分散レイトレーシング法では，サブピクセル単位でのレイトレースを行うため，ぼやけたハイライトを再現しやすいという特徴がある．

パストレーシング法

レイトレーシング法では拡散反射成分や鏡面反射成分の相互反射を完全に取り扱うことはできない．この問題を解決するための手法としてパストレーシング法がある．この手法はレイトレーシング法と似ているが視点に

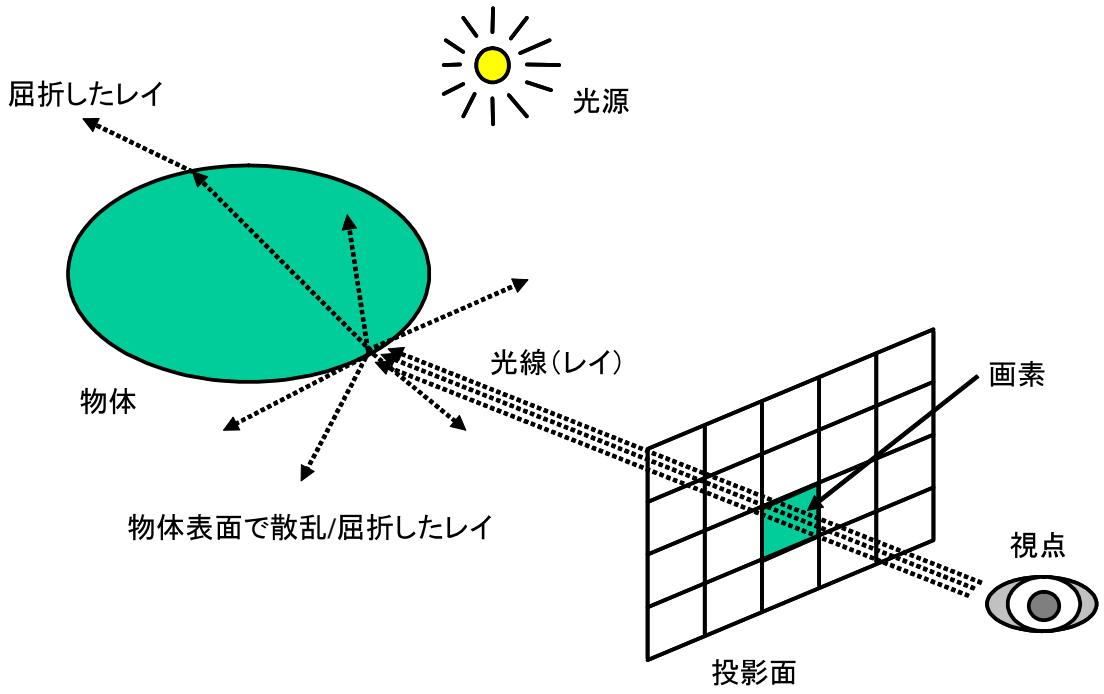


図 4 パストレーシング法によるレンダリング

到達するレイの経路を辿るという考えに基づいている．すなわち，レイが物体表面に衝突した際に，レイが散乱される現象を確率を用いて表現し，最終的に光源に到達したレイの総和が結果画像の輝度値となるレンダリング手法である．

図 4 にパストレーシング法において画像の輝度値を決定する場合の概略図を示す．パストレーシング法では視点に到達したレイの経路を逆方向に辿ることでレンダリングを行う．このとき，物体表面での散乱を確率的に扱うことで材質の反射特性を表現している．そして，レンダリングの際に多数のレイを計算することで拡散反射や，表面が粗い材質における鏡面反射などの相互反射を正確に表現することが可能になる．しかし，パストレーシング法では確率を用いてレイの散乱を記述しているため，レイの数が少ない場合には結果画像にノイズが乗りやすい欠点がある．これは，確率を用いて散乱を記述しているため，計算したレイの数が十分でない場合には

計算が安定に行われず，計算結果である画像の輝度値にばらつきが生じてしまうためである．このため，ノイズの少ない画像を生成するためには，非常に多数のレイを計算する必要があり，計算量がレイトレーシング法以上に膨大であるという欠点がある．また，レイの経路を辿る際に，パストレーシング法では光源に到達しないレイも追跡している．しかし，このようなレイは結果画像に影響を与えないため，計算が効率的に行われないという問題がある．

そこでこの問題を解決するための手法として，フォトンマッピング [26] が提案されている．この手法では光源から放射される光子 (フォトン) をいったんフォトンマップとして物体表面上に保持しておくことで有効でないレイの経路を辿ることを減らし，パストレーシング法の欠点を改善する手法となっている．

上で述べたように，いずれの手法も計算量の多いレンダリング手法であり実時間処理は困難である．このため，実時間レンダリングの分野においては相互反射を擬似的に再現するための手法が研究されている．

2.1.3 環境マッピング

相互反射を擬似的に再現するための手法として，環境マッピング [9] がある．この手法は，相互反射を簡略化して取り扱うことで写り込みを高速にレンダリングする手法として Blinn らによって提案された．特徴として，写り込みは物体を取り巻く仮想球から入射する光のみによって生じるとして近似を行うことで高速なレンダリングを可能とした点が挙げられる．図 5 に環境マッピングによる写り込みのレンダリングの際の概念図を示す．環境マッピングでは物体を取り巻く仮想的な球を考え，シーンから入射する光をいったん仮想球面上へ投影し，光の情報を画像 (環境マップ) として保持する．そして，写り込みのレンダリングを行う際には，まず物体の表面上での視点ベクトルの正反射ベクトルを計算し，正反射ベクトルと対応したテクスチャ座標を計算する．そして，このテクスチャ座標を用いて環境マップをテクスチャマッピングすることで，写り込みのレンダリングを行う．このとき，環境マップを保持する場合の球面の展開の方式によって様々な

手法が提案されている．

Spherical 環境マッピング

Spherical 環境マッピング [9] では，球面座標を用いて環境マップを展開する (図 6(a) 参照)．1 枚の画像で環境マップを表すことができるが，周辺部分が大きく歪む欠点が存在する．また，環境マップの周辺部分は球面上の一点にマッピングされるため，この点において歪みが生じる欠点がある．

Dual – Paraboloid 環境マッピング

Dual – Paraboloid 環境マッピング [27] では 2 つの放物面を用いて球面を展開する (図 6(b) 参照)．このため，Spherical 環境マップのような大きな歪みはないが，テクスチャを 2 枚用いる必要がある．また，2 つのテクスチャの境界部分の処理をうまく行わないと 2 枚のテクスチャのつなぎ目が見えてしまうといった問題がある．

Cube 環境マッピング

Cube 環境マッピング [28] では立方体の 6 面を用いて展開する (図 6(c) 参照)．ぞれの環境マップの歪みが小さいことや，複雑な座標計算が必要ではないといった点が特徴として挙げられる．しかし，Dual – Paraboloid 環境マッピングと同様につなぎ目の処理の問題などがある．

Spherical 環境マッピングと Dual – Paraboloid 環境マッピングは，反射ベクトルからテクスチャ座標への変換が複雑な式となるという欠点や，環境マップを保持する際の画像周辺での歪みが問題となった．また，Cube 環境マップではテクスチャを 6 枚用いるためメモリの使用量が他の手法より多いという欠点がある．しかし，現在では座標計算の容易性などからグラフィックハードウェアの機能として，Cube 環境マップが実装されていることが多い．しかし，いずれの環境マップを用いても鏡のような反射特性を持つ材質に対する写り込みしか再現できない．そこで，次に様々な材質に対する写り込みを実現した環境マッピングについて述べる．

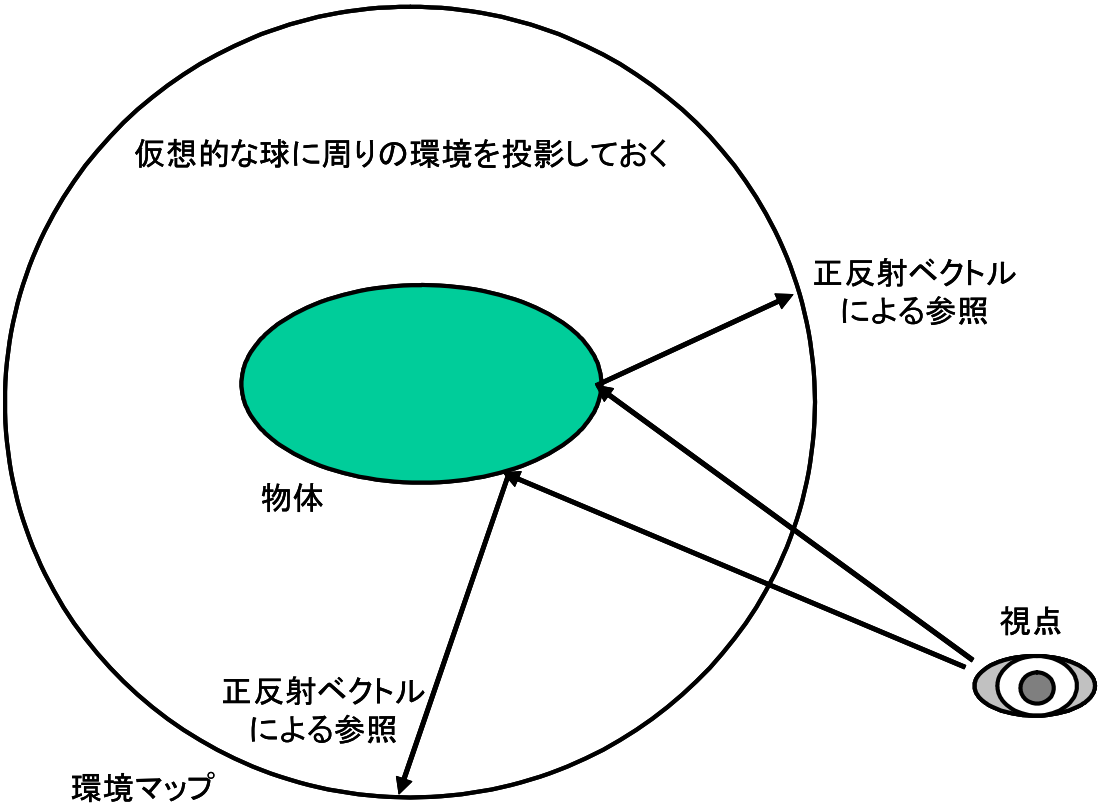
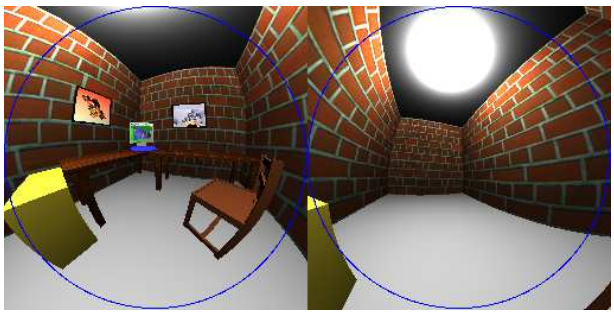
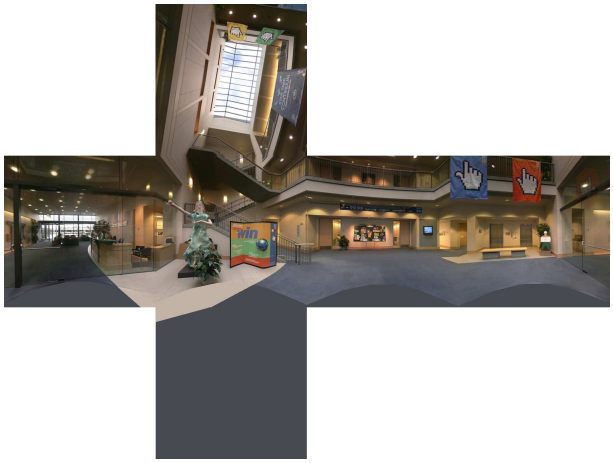


図 5 環境マッピングによる写り込みのレンダリング

(a) Spherical Environment Mapping(出典 [29])



(b) Dual – Paraboloid Environment Mapping(出典 [27])



(c) Cube Environment Mapping(出典 [29])

図 6 異なる展開方式による環境マップ

2.1.4 材質の反射特性を考慮した環境マッピング

環境マッピングでは物体表面の反射特性を考慮することができなかった．そこで，環境マッピングを拡張し物体表面の反射特性を考慮することで，レンダリング時のコストを引き上げることなく CG の表現能力を高めることが可能となった [30, 31, 32]．以下に，反射特性を考慮した環境マッピングに関する従来手法の代表的なものについて示す．

Miller ら [10] による研究では，環境マップに対してフィルタ処理を施すことでランバート面に対する拡散反射成分のレンダリング手法を提案している．この手法では拡散反射成分は物体の法線ベクトルにのみ依存することを用いて，環境マップに拡散反射の輝度値を格納している．そして，通常の環境マップは視点からの正反射ベクトルで環境マップをサンプリングするが，この手法では法線ベクトルによってサンプリングすることで，拡散反射成分のレンダリングを高速に行うことを可能としている．

Cabral ら [11] による研究では，実在する材質の反射特性に基づいた写り込みを高速にレンダリングする手法を提案している．この手法は，レンダリング対象となる材質と同じ材質で構成された球体を用意し，その球面上に生じる写り込みをあらかじめ撮影する．そして，撮影された画像を用いて仮想物体にテクスチャマッピングを施すことで材質の反射特性に基づいた写り込みと拡散反射のレンダリングを行う手法となっている．しかし，あらかじめ画像を撮影しておかなければならない点や，レンダリングする材質に応じた球体用意しなければならないといった問題がある．また，シーンを変更して写り込みをレンダリングする際には実環境の異なるシーンで撮影し直す必要がある．

Heidrich ら [12] は，環境マップに対してあらかじめフィルタ処理を行うことで材質の反射特性に応じた写り込みのレンダリング手法を提案している．この研究では写り込みの強度に影響を与える Fresnel 項について考慮している．Fresnel 項を考慮することで，光の入射角に応じて写り込みの強度が変わるという現象が再現されたため，より現実に近い写り込みの表現が可能となった．また，物体表面の法線方向をテクスチャとして保持することで，細かいメッシュを持つことなく細かな凹凸を表現することが可能な Normal Map を提案している．この Normal

Mapによって，より精細な写り込みをレンダリングすることが可能となった．しかし，あらかじめフィルタ処理を行わなければならず，物体表面の反射特性に対する自由度が低いため，あらかじめ設定した反射特性でしか写り込みをレンダリングできないという問題がある．

Kautzら [13] による研究では，物体表面の反射特性を考慮した写り込みを高速にレンダリングする手法を提案している．この手法では，環境マップに対して物体表面の反射特性に基づいたフィルタ処理を行うことで，反射特性に応じた写り込み画像を作成し，その画像を環境マップとして用いることで写り込みのレンダリングを行う手法である．また，入力となる環境マップと反射特性に基づいたフィルタを適度に縮小することで計算量の削減を図っている．さらに，グラフィックハードウェアのMipmap生成機能を用い，縮小画像の高速な生成を行うことで高速なフィルタ処理を可能としている．以上の処理は実時間で行われるため，材質の反射特性が変更された場合であっても写り込みのレンダリングが可能となる．しかし，Kautzらの手法ではシーン内に単一の反射特性を持つ材質を想定している．このため，反射特性の異なる材質が同時に複数存在するシーンをレンダリングする場合には，それぞれの材質の反射特性に応じた写り込み画像を複数生成する必要があり，計算時間が増大するという問題がある．

2.2 本研究の目的と方針

以上をふまえて，本研究の目的とアプローチについて述べる．研究の目的は材質の反射特性に対応した写り込みを高速にレンダリングすることである．このとき，材質の反射特性の中でも写り込みに対して最も影響を与える粗さに着目する．つまり，シーン内に複数の粗さが存在する場合や，粗さが不均一な材質から構成される物体の高速レンダリングを可能とするため，粗さに対して自由度を持ったレンダリング手法を提案する．

提案手法では，表面が粗い材質で生じるボケた写り込みの情報をあらかじめ計算しておく．これは，入力となる環境マップに対して反射特性に応じたフィルタ処理を施すことで計算される．この計算はオフライン処理として行われ，結果はボクセルデータとして保持される．本論文ではこのボクセルデータを多重スケール

環境マップと呼ぶ．レンダリングの際には多重スケール環境マップをテクスチャとして用いることで不均一材質への写り込みのレンダリングを行う．多重スケール環境マップには複数の粗さに対応した写り込み情報が格納されているため，シーン内に異なる粗さを持つ材質が複数存在してもレンダリング速度は一定である．また，近年のグラフィックハードウェアには 3D テクスチャマッピング機能が搭載されているため，ボクセルデータをテクスチャとして利用する際にハードウェアの支援が得られ実時間でのレンダリングが可能となる．

しかし，多重スケール環境マップの計算は単純に実行すると膨大な計算量となる．そこで，複数の解像度の異なる画像 (ピラミッド画像) と頂点数の異なる測地ドーム (Geodesic Dome) を用いることで計算の効率化を図る．さらに，多重スケール環境マップの計算の際に用いられるフィルタの重みをあらかじめ計算し，テーブルとして保持しておくことで高速化を図る．

3. 多重スケール環境マップを用いた写り込みのレンダリング

本章では提案手法の詳細について述べる．まず，提案手法で用いた材質の反射モデルである Phong モデルについて詳細を示し，提案手法で用いた Phong モデルの簡略化について述べる．そして，簡略化された Phong モデルを用い，粗さに応じた写り込み情報の保持と計算方法を示す．このとき，多重スケール環境マップに写り込み情報を格納する際のボクセルの解像度と表現可能な粗さの対応が問題となるため，提案手法における粗さの表現と Phong モデルにおける対応関係について述べる．最後に，多重スケール環境マップを用いた写り込みのレンダリング方法について述べ，多重スケール環境マップの作成の際に用いた高速化手法の詳細を示す．

3.1 材質の反射モデル

本研究では材質の反射モデルとして Phong モデル [16] を用いる．Phong モデルは材質の粗さを表現可能な照明モデルであり，図 7 のように物体表面上のある点 $\mathbf{x}$ に対して，視点方向を $\vec{v}$，光源の方向を $\vec{l}$，物体表面の法線を $\vec{n}$，視点ベクトルの正反射ベクトルを $\vec{r}$，$\vec{r}$ と $\vec{l}$ のなす角を θ，$\vec{n}$ と $\vec{l}$ のなす角を ϕ としたとき，写り込みの輝度が $\cos^{\Lambda}(\theta)$ となる照明モデルである．なお，各ベクトルは単位ベクトルとする．Λ は粗さ係数となり，Λ が大きいほど表面が滑らかな材質となる．

一般に Phong モデルによって環境内に配置された物体に対する写り込みをレンダリングする場合，入力となる環境を Ω，Ω 上の光源 i の輝度を L_i とすると輝度値の計算は式 (1) を用いて行われる．

$$L(\mathbf{x};\vec{\mathbf{v}},\vec{\mathbf{n}}) = K_r \int_{\Omega} f(\cos(\theta),\cos(\phi),\Lambda) L_i \, d\vec{l} \tag{1}$$

ただし，

$$f(\cos(\theta),\cos(\phi),\Lambda) = \begin{cases} \cos^{\Lambda}(\theta) & ; \text{if } \cos(\theta) \geq 0 \\ & \quad \text{and } \cos(\phi) \geq 0 \\ 0 & ; \text{otherwise} \end{cases} \tag{2}$$

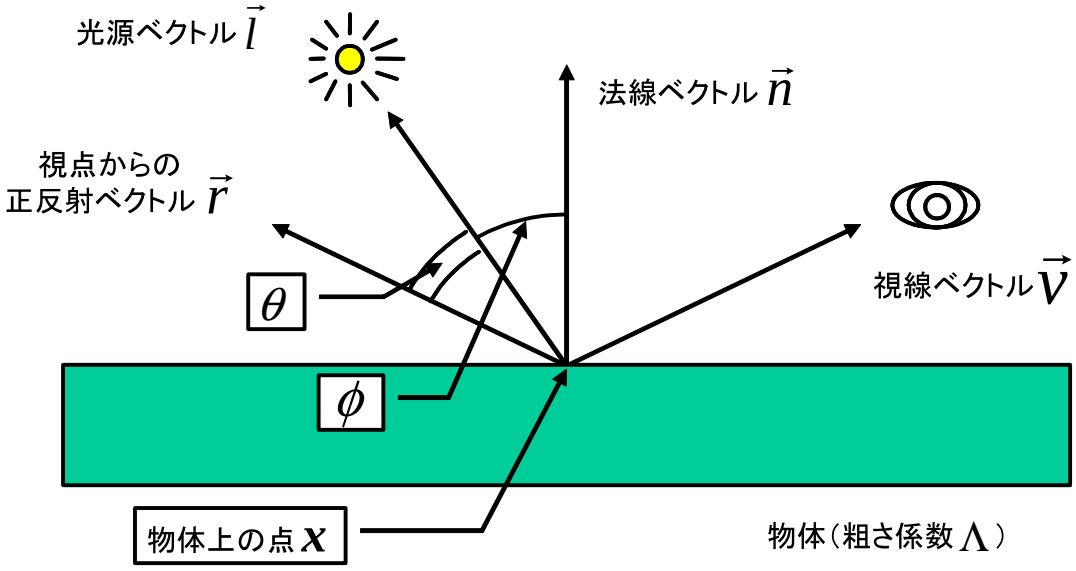


図 7　物体と視点と光源の関係

ここで，関数 $f(\cos(\theta), \cos(\phi), \Lambda)$ は Phong モデルに基づいた反射輝度を決定する式となっており，θ によって写り込みの輝度強度が決定され，ϕ によって光源が物体を照らすかどうかの判定を行っている．

3.2　多重スケール環境マップ

Phong モデルを用いて自然な照明計算を行うためには Ω 上に多数の光源を配置しなければならない．しかし実時間でのレンダリングを行う場合，多数の光源を用いた照明計算は計算時間の点から困難である．

そこで本研究では式 (2) に着目する．一般に表面が粗い反射特性を持つ材質であっても Λ はあまり小さくならないことが経験的に知られており，θ がある程度大きくなると $\cos^{\Lambda}(\theta)$ の値は非常に小さくなる．そこで，写り込みの計算を行う場合に光源が物体を照らすかどうかの判定，すなわち ϕ による判定を省くことで式 (1) を式 (3) のように変形する．

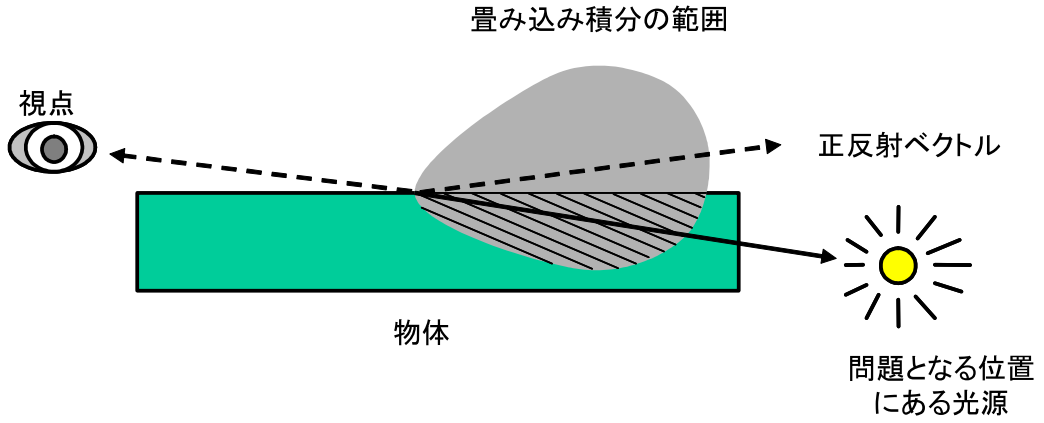


図 8 物体に隠される光源からの影響

$$L(\mathbf{x};\vec{\mathbf{v}},\vec{\mathbf{n}}) = K_r \int_\Omega f'(\cos(\theta),\Lambda) L_i \, d\vec{l} \tag{3}$$

$$= K_r F(\vec{r},\Lambda;\Omega) \tag{4}$$

ただし，

$$f'(\cos(\theta),\Lambda) = \begin{cases} \cos^\Lambda(\theta) & ;\ \text{if } \cos(\theta) \geq 0 \\ 0 & ;\ \text{otherwise} \end{cases} \tag{5}$$

この変形によって，物体に隠されている光源からの影響を受けるようになるため (図 8 参照)，表面が粗い材質 (Λ が小さい場合) を物体に対して浅い角度で観測する場合（$\vec{n}$ と $\vec{v}$ のなす角が $\pi/2$ に近い場合）に誤差が生じるが，そのような状況はまれであるためその影響は無視できると仮定する．

また，式 (3) の右辺を式 (4) のように表すと F は，表面の粗さが Λ であり視線からの正反射ベクトルが $\vec{r}$ となる物体上の点を観測した場合の写り込みの輝度値を得るための関数となる．このとき，F は環境 Ω にのみ依存するため，環境に変化がないと仮定すればあらかじめ計算することができ，正反射ベクトルの方向 $\vec{r}$ と表面粗さ係数 Λ の 3 次元のパラメータで記述可能なボクセル空間で表現される．そこで写り込みの輝度値をボクセルデータとして記録しておき，レンダリング時にボクセルを参照することで高速なレンダリングが可能となる．本研究では

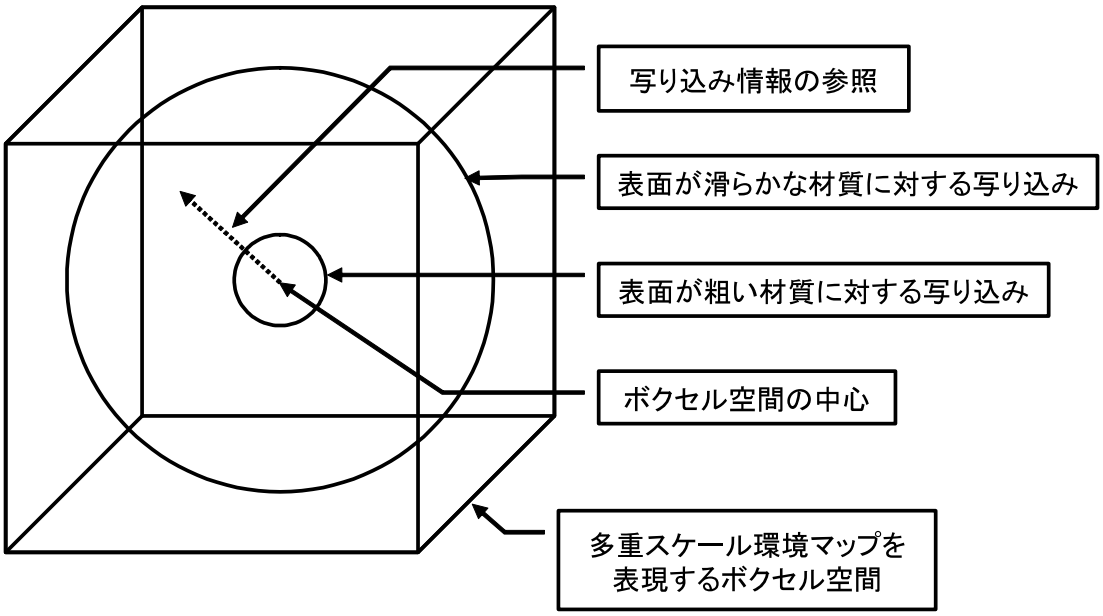


図 9 ボクセル空間内に配置される写り込み情報

この環境の写り込み情報を特に多重スケール環境マップと呼ぶ.

3.2.1 材質の粗さに応じた写り込み画像のボクセルデータ化

保持される写り込み情報のうち，表面が粗い材質に対する写り込みは強いボケが生じていると考えられるため，写り込み情報を保持するために高い解像度を割り当てることは無駄である．そこで，表面が粗い材質に対応する写り込み情報はボクセル空間の中心付近に，表面が滑らかな材質に対応する写り込み情報はボクセル空間の外周付近に格納する（図 9 参照）．こうすることで，粗い材質に対しては低解像度で，滑らかな材質に対しては高解像度で写り込みの情報を保持することができる.

多重スケール環境マップを表現するボクセル空間の中心からの方向が反射ベクトル，ボクセル空間の中心からの距離が材質の滑らかさに対応し，写り込み情報の参照が容易になる．つまり，ボクセル空間の中心からあるボクセル $\mathbf{t}$ へのベク

トルを $\vec{t}$ とすると，以下の関係が成立する．

$$\vec{t} = \vec{r} \cdot \lambda \tag{6}$$

$$\Lambda = g(|\vec{t}|) = g(|\vec{r}| \cdot \lambda) \tag{7}$$

ただし，λ は提案手法における粗さの表現であり，Phong モデルにおける粗さ Λ との対応は関数 g によって与えられる．この関数については，3.2.2 節で詳しく述べる．

入力となる単位球面上に定義された環境マップを Ω，環境マップの中心から環境マップ上のある画素へのベクトルを $\vec{\omega}$ とすると，多重スケール環境マップを計算するための式は以下の通りである．

$$F(\mathbf{t}; \vec{\mathbf{t}}) = \sum_{\vec{\omega}: 0 \leq \vec{t} \cdot \vec{\omega}} \Omega(\vec{\omega}) \cdot w(\vec{t}, \vec{\omega}) \tag{8}$$

ただし，

$$w(\vec{t}, \vec{\omega}) = \frac{(\vec{t} \cdot \vec{\omega})^{\Lambda}}{\sum_{\vec{\rho}: 0 \leq \vec{t} \cdot \vec{\rho}} (\vec{t} \cdot \vec{\rho})^{\Lambda}} \tag{9}$$

$$\Lambda = g(|\vec{t}|) \tag{10}$$

各ボクセルの値は，入力とする環境マップの各画素を光源とし，各光源に対して Phong モデルによる重み $w(\vec{t}, \vec{\omega})$ との重み付き加算を行い，その総和をとることで得られる．この計算は $\vec{t}$ と $\vec{\omega}$ のなす角を ψ とおけば，入力となる環境マップに対して $\cos^{\Lambda}(\psi)$ で表現されるフィルタの畳み込み積分を行うのと等価である（図 10 参照）．

3.2.2 材質の粗さの表現

多重スケール環境マップではボクセル空間の中心からの距離が材質の粗さと対応しているため，ボクセル空間内で粗さが等しい点を選ぶとある半径の球面を構成する．このとき球面を構成するボクセルの数によって，ある粗さの写り込みを保持するのに用いられるボクセル数，すなわちある粗さの環境マップを保持する解像度が決定する．ここで，滑らかな写り込みを保持する場合に解像度が低いと写り込みを正確に保持できない問題が発生する．

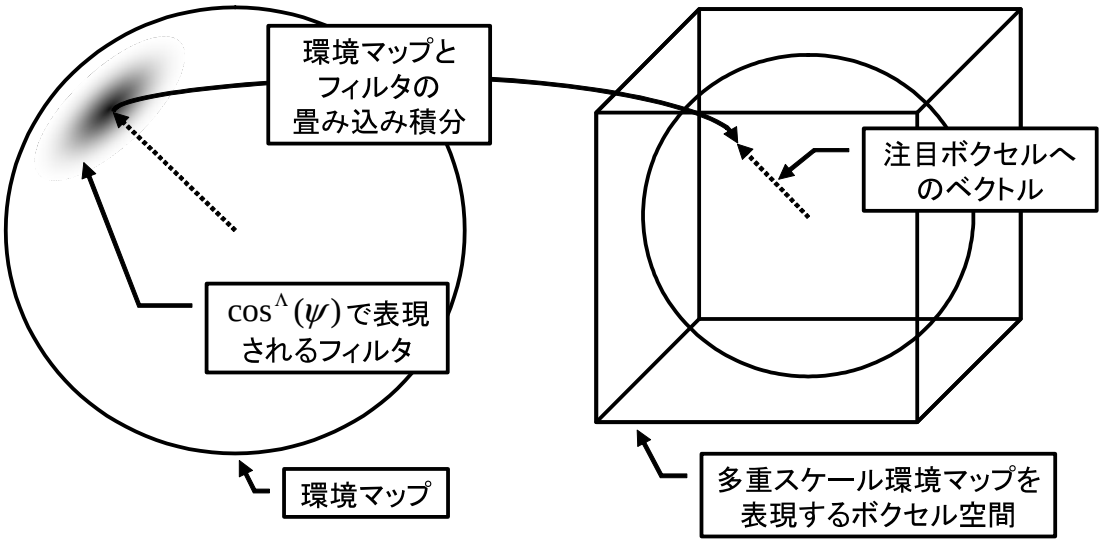


図 10 多重スケール環境マップの計算

そこで，材質への写り込みがボクセルデータとして正しく保持できる条件が必要となる．これは，ある粗さの写り込みを保持する環境マップの空間分解能が $\cos^{\Lambda}(\psi)$ で示されるフィルタのサイズを上回っていればよい．本研究では簡単のため，中心からの距離が同じである隣り合った 2 つのボクセルへのベクトルのなす角を ψ_{voxel}，ボクセルデータの輝度値の分解能を I，ボクセル空間の解像度を s とした時，条件式 (11) を満たす Λ を表現可能な粗さとする．

$$\frac{1}{I} \geq \cos^{\Lambda}(\psi_{voxel}) \tag{11}$$

ただし，

$$\psi_{voxel} = \arctan(2/(|\vec{t}| \times s)) \tag{12}$$

また，以上より式 (7) の $g(|\vec{t}|)$ は

$$\Lambda = g(|\vec{t}|) = \frac{-\log(I)}{\log(\cos(\arctan(2/(|\vec{t}| \times s))))} \tag{13}$$

となる．

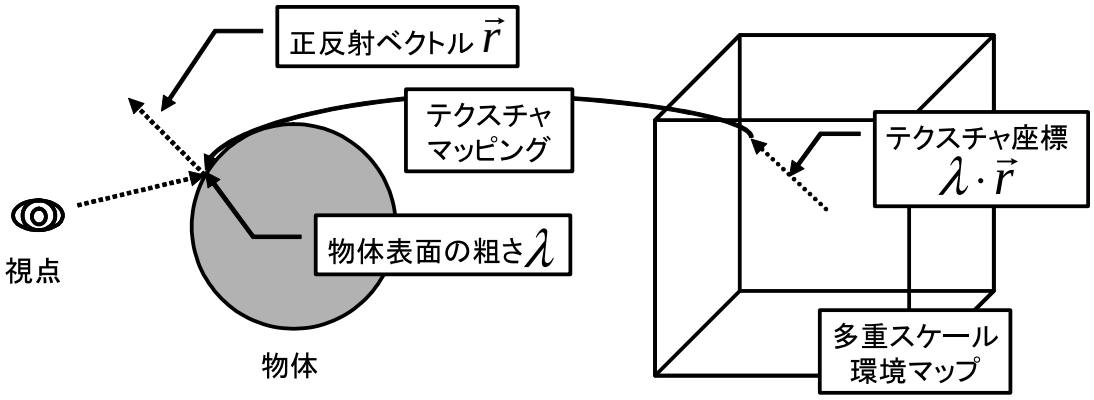


図 11 テクスチャマッピングによる写り込みのレンダリング

3.3 多重スケール環境マップを用いたレンダリング

写り込みのレンダリングは多重スケール環境マップとして計算されたボクセルデータを 3D テクスチャとしてテクスチャマッピングすることで行う．図 11 は，写り込みをレンダリングする際のテクスチャ座標の決定方法を示している．ボクセルデータのサンプリングに用いられるテクスチャ座標 $\vec{t}$ は，ユーザが設定する粗さ係数 $\lambda(0 < \lambda \leq 1)$ と正規化された視点からの反射ベクトル $\vec{r}$ より式 (6) で計算される．

このとき，粗さ係数を描画するポリゴンの頂点単位に設定することで，粗さが一様でない材質に対する写り込みを一度にレンダリングすることが可能である．また，正反射ベクトルの計算には近年のグラフィックハードウェアの機能であるテクスチャ座標生成を用い，さらにハードウェアの 3D テクスチャマッピング機能を利用することで高速なレンダリングを行う．

3.4 多重スケール環境マップの作成の高速化

式 (8) は畳み込み積分を行う計算であるため，計算量は環境マップ上の画素数と多重スケール環境マップのボクセル数に依存し，単純に実行すれば膨大な計算量となる．そこで，入力となる環境マップに解像度の異なる複数の画像を用いる

ことで計算の簡略化を行い，さらに重みをテーブル化することでレンダリング結果に影響の少ない計算を省く．

3.4.1 ピラミッド画像と測地ドームの利用

多重スケール環境マップにおいてボクセル空間の中心付近の情報，すなわち表面が粗い材質に対応する写り込み情報は，計算結果を格納する際の解像度が低く割り当てられている．しかし，式 (8) の計算ではボクセル空間の中心付近に存在するボクセルの輝度値を決定する際にも，入力となる環境マップの画素を利用して計算している．これは，粗い材質に対する写り込みを計算するために高い解像度の環境マップ使用することになり計算が膨大になる原因である．

そこで，表面が粗い材質に対応する写り込み情報の計算にはあらかじめ線形フィルタで縮小された環境マップ（ピラミッド型画像）を利用して式 (8) の計算を行う．また，本手法では入力となる環境マップからいったん細分化された測地ドームを用いて再サンプリングを行う．これを行うことで，入力となる環境マップの形式に依存することが無くなり，さらに環境マップの形式による解像度の偏りが無くなる．図 12 は複数の解像度の画像を用いてフィルタ計算を行う場合の概念図を示す.

3.4.2 重みテーブルの利用

提案手法では，さらに計算の高速化のために式 (8) の重みをあらかじめ計算することでテーブル化を行った．このとき，閾値 th を用いて $\cos^{\Lambda}(\psi) > th$ を満たす重みのみをテーブルに記録する．これにより，計算結果に大きな影響を与える環境マップの画素のみを計算対象とすることができ，計算量の削減が行える．なお，ボクセルの総数は N 個，i 番目のボクセルを計算する時に計算対象となる環境マップの画素数は M_i 個であるとする．

表 1 は重みテーブルに保存されるデータ形式を示す．左端の列は計算結果を格納するボクセルを示し，各ボクセルに対応した参照すべき環境マップ上の画素と重みの組である．各ボクセルの値はこのテーブルを用いて以下の式を計算するこ

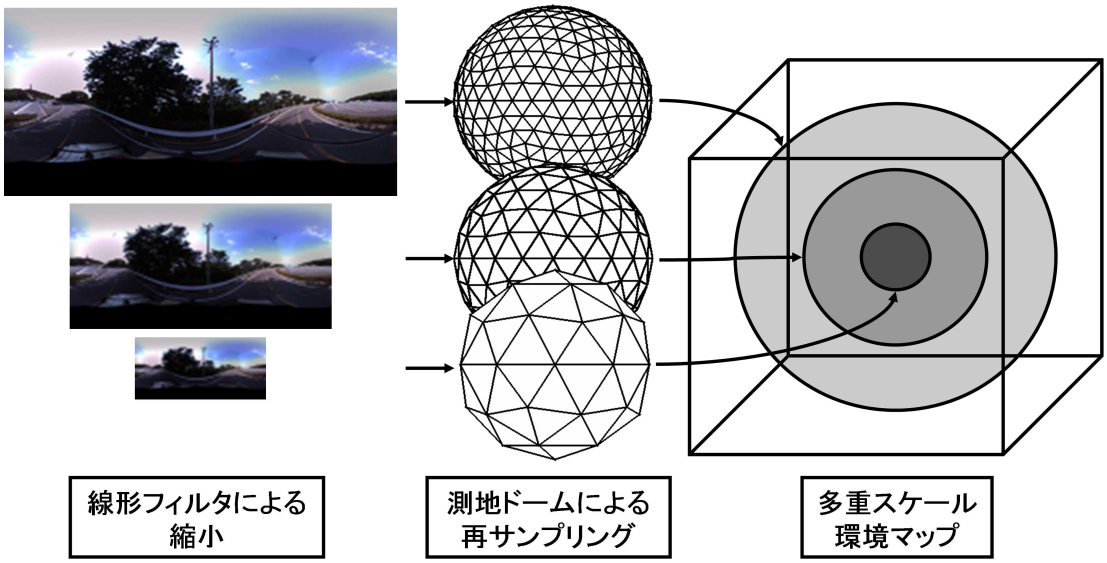


図 12　複数の解像度の画像を用いたフィルタ計算

表 1　重みテーブルの例

対象ボクセル	環境マップ上の参照画素と重みのリスト
$F(\vec{t}_0)$	$(\vec{\omega}_{0,0}, w_{0,0}),\quad \ldots \quad ,(\vec{\omega}_{0,M_0}, w_{0,M_0})$
$F(\vec{t}_1)$	$(\vec{\omega}_{1,0}, w_{1,0}),\quad \ldots \quad ,(\vec{\omega}_{1,M_1}, w_{1,M_1})$
$\vdots$	$\vdots$
$F(\vec{t}_N)$	$(\vec{\omega}_{N,0}, w_{N,0}),\quad \ldots \quad ,(\vec{\omega}_{N,M_N}, w_{N,M_N})$

とで決定される.

$$F(\vec{t_i}) = \sum_{j=0}^{M_i} \Omega(\vec{\omega}_{i,j}) \times w_{i,j} \tag{14}$$

4. 実験

提案手法の妥当性を評価するための実験を行った．まず，提案手法を用いた粗さが不均一な材質への写り込みのレンダリングを行い，粗さが不均一な材質に対しても高速にレンダリングすることが可能なことを確認する．そして，3.4 節で示した高速化手法の有効性を確認する．さらに，提案手法によるレンダリング結果と Phong モデルによるレンダリング結果の比較を行い，提案手法が Phong モデルを近似できていることを確認する．最後に，環境として動画像を用い，動きのあるシーン内での写り込みのレンダリング結果を示す．

また，計算には Pentium4 Xeon 1.7GHz Dual，メモリ 2GByte，グラフィックカードとして NVIDIA 社 GeForce3(VRAM 64MByte) を搭載した計算機を用いた．なお，4.1 節から 4.3 節については並列処理を行わず，最後の 4.4 節については並列処理を行った．

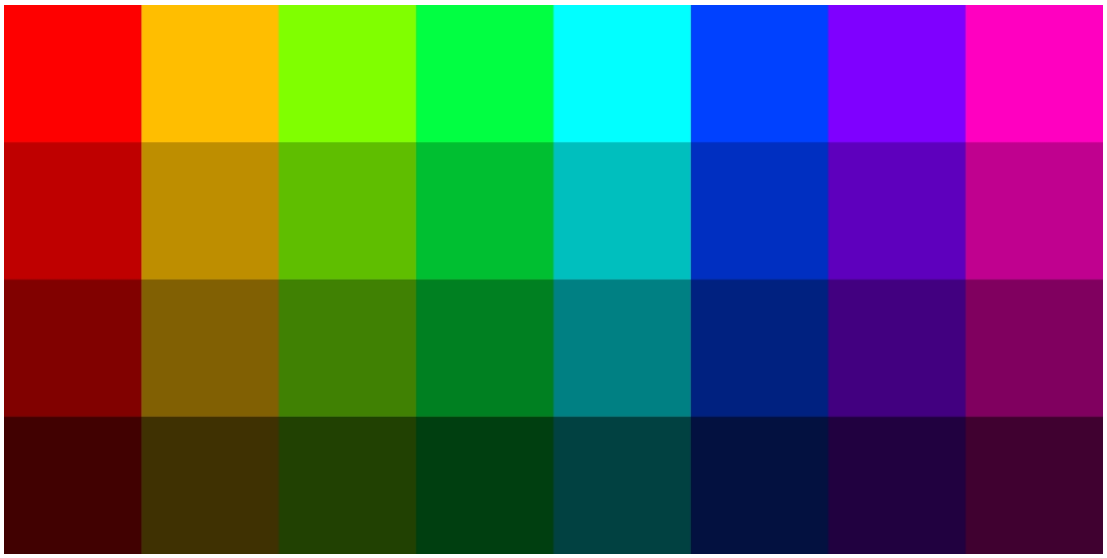
4.1 粗さが不均一な材質へのレンダリング結果

本節では粗さの異なる材質に対する写り込みを高速にレンダリング可能なことを示す．入力として図 13 に示す環境マップを用いた．図 13(a) の画像は横方向に色相，縦方向に彩度が変化する人工カラーパターンであり，図 13(b) の画像は全方位マルチカメラシステム Ladybug[33] を用いて撮影された高山サイエンスプラザ内の全天球パノラマ画像である．それぞれの画像の解像度は 512×256 である．そして，鏡面反射成分のみを持つ仮想物体 (球体とティーポット) に対する写り込みのレンダリングをそれぞれ行った．表面粗さ係数は，提案手法における粗さ表現において，$\lambda = 1, 0.5, 0.25$ の 3 通りと，不均一な粗さを持つ場合の計 4 通りである．不均一な粗さを持つ場合，球体は粗さが 0〜1 まで連続的に変わるように設定した．これは，向かって 6 時の方向が $\lambda = 0$ であり，9 時の方向が 0.25，12 時の方向が 0.5，3 時の方向が 0.75，6 時の方向が 1 となるように設定した．また，ティーポットは下半分の部分は $\lambda = 0.2$ 程度の粗さを持ち，上半分は $\lambda = 0.9$ 程度の反射特性を持つように設定した．なお，このとき利用した多重スケール環境マップの解像度は $128 \times 128 \times 128$ であり，多重スケール環境マップ計算の際の

高速化は行っていない．

図 13(a) の環境中での写り込みのレンダリング結果を図 14，図 15 に，図 13(b) での結果を図 16，図 17 に示す．図 14，図 16 は，粗さが均一な材質からなる物体の粗さを変えてレンダリングした結果であり，それぞれ，球体に対するレンダリング結果を (a),(c),(e) に，ティーポットに対するレンダリング結果を (b),(d),(f) に示す．また，図 15，図 17 粗さが不均一な材質へのレンダリング結果であり，それぞれ球体に対するレンダリング結果を (a)，ティーポットに対するレンダリング結果を (b) に示す．なお，多重スケール環境マップの作成は 73 時間を要した．また，多重スケール環境マップのサイズは 8MByte である．レンダリングは 640×480 の解像度においていずれも 60fps 以上で実行可能であった．

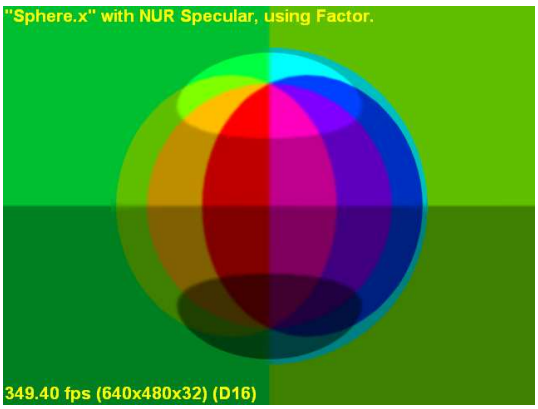
実験より，あらかじめ計算された多重スケール環境マップを用いて，粗さが不均一な材質に対する写り込みのレンダリングが高速に行えることを確認した．特に図 17(a) では粗さに応じて写り込む画像がぼやけており，写り込みが粗さを反映している．しかし，粗さが急激に変化している部分 (図 17(b) ティーポット中央付近など) で写り込みが歪んでいるのが分かる．本手法では粗さを頂点単位で設定しているため，粗さ係数が大きく異なる頂点で構成されるポリゴンをレンダリングする際に，テクスチャマッピングによる線形補間による影響であると考えられる．



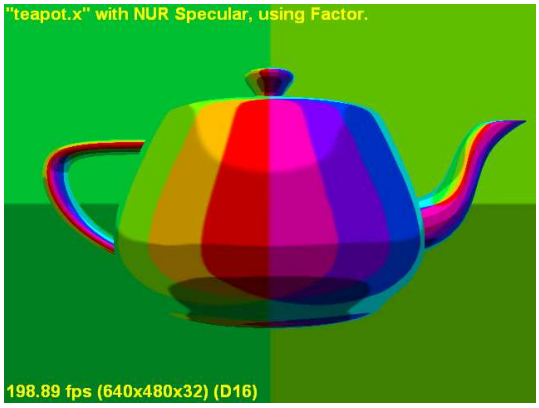
(a) 人工カラーパターン

(b) 高山サイエンスプラザの全方位画像

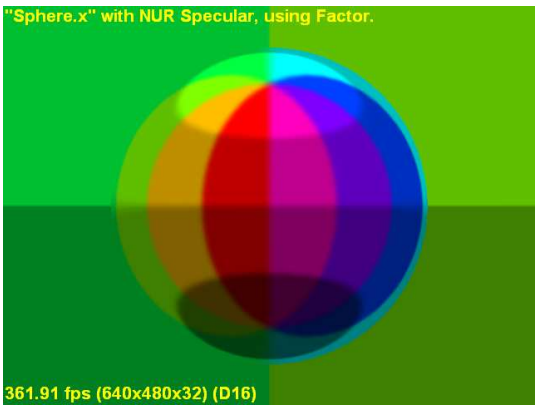
図 13　実験に用いた 2 つの環境マップ



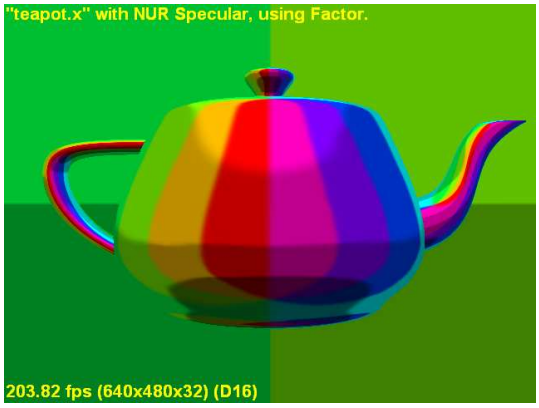
(a) 球体 ($\lambda = 1.0$)



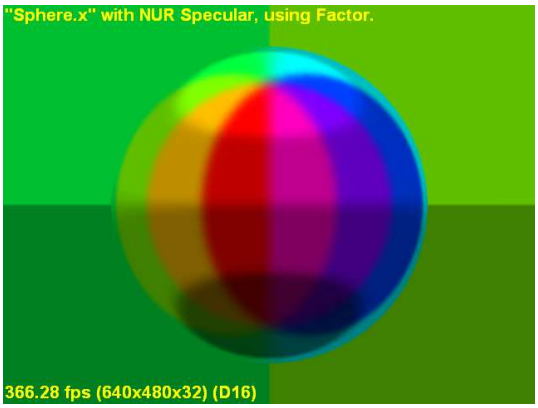
(b) ティーポット ($\lambda = 1.0$)



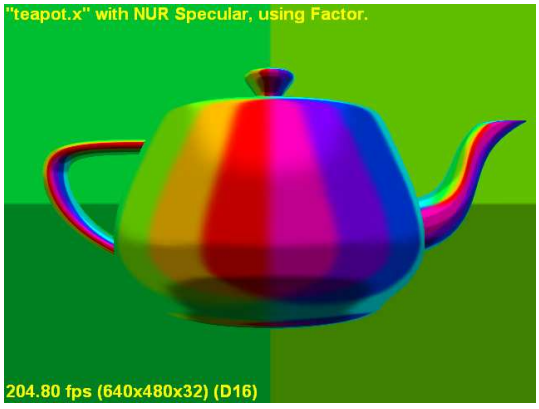
(c) 球体 ($\lambda = 0.5$)



(d) ティーポット ($\lambda = 0.5$)

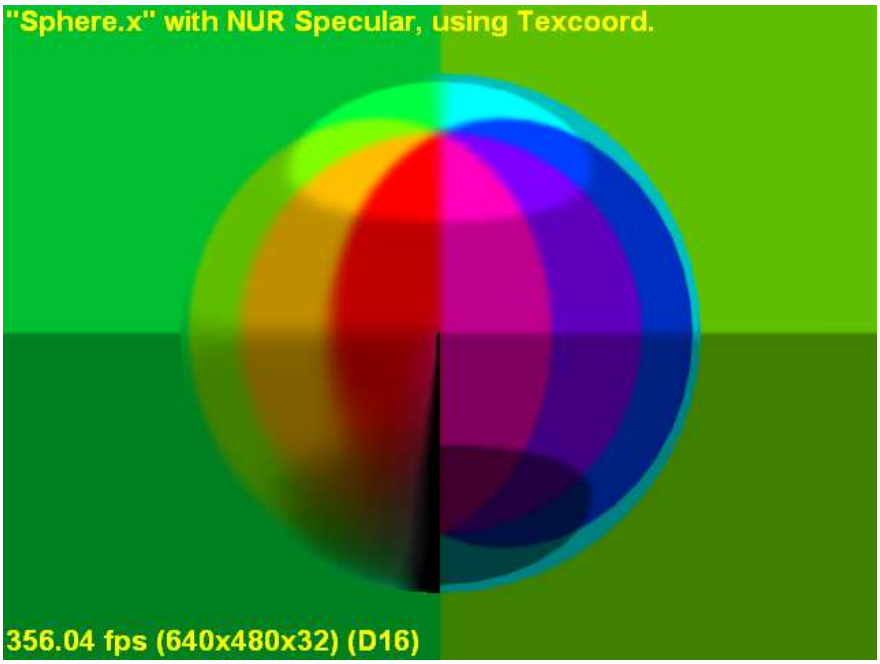


(e) 球体 ($\lambda = 0.25$)

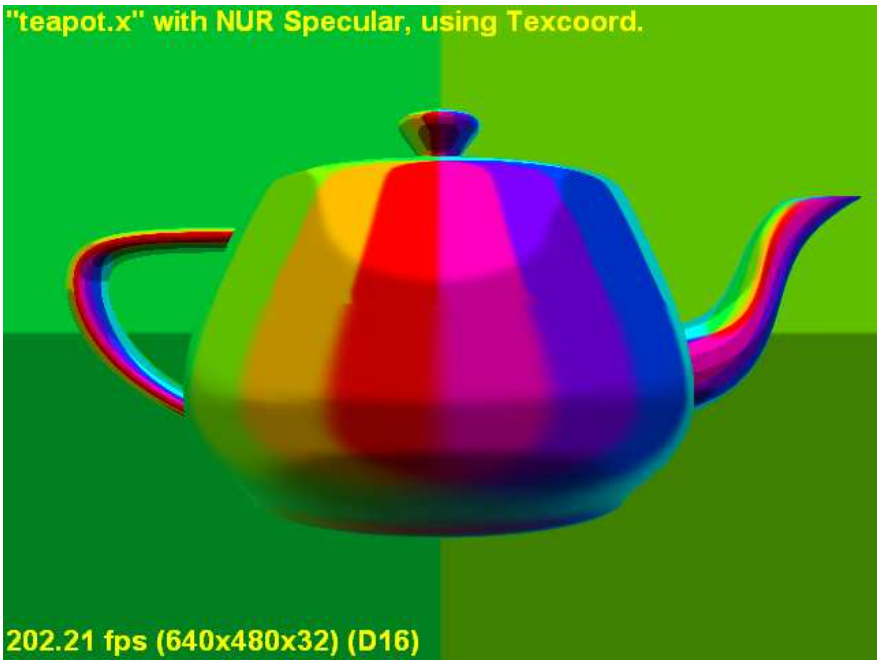


(f) ティーポット ($\lambda = 0.25$)

図 14 カラーパターンを用いたレンダリング結果 (均一な材質)



(a) 球体 (不均一な粗さ)



(b) ティーポット (不均一な粗さ)

図 15　カラーパターンを用いたレンダリング結果 (不均一な材質)

(a) 球体 ($\lambda = 1.0$)

(b) ティーポット ($\lambda = 1.0$)

(c) 球体 ($\lambda = 0.5$)

(d) ティーポット ($\lambda = 0.5$)

(e) 球体 ($\lambda = 0.25$)

(f) ティーポット ($\lambda = 0.25$)

図 16　実写画像を用いたレンダリング結果

(b) 球体 (不均一な粗さ)

(b) ティーポット (不均一な粗さ)

図 17 実写画像を用いたレンダリング結果 (不均一な材質)

表 2 画像の解像度と測地ドームの頂点数の対応

画像の解像度	測地ドームの頂点数
512×256	51842
256×128	12962
128×64	3242
64×32	812

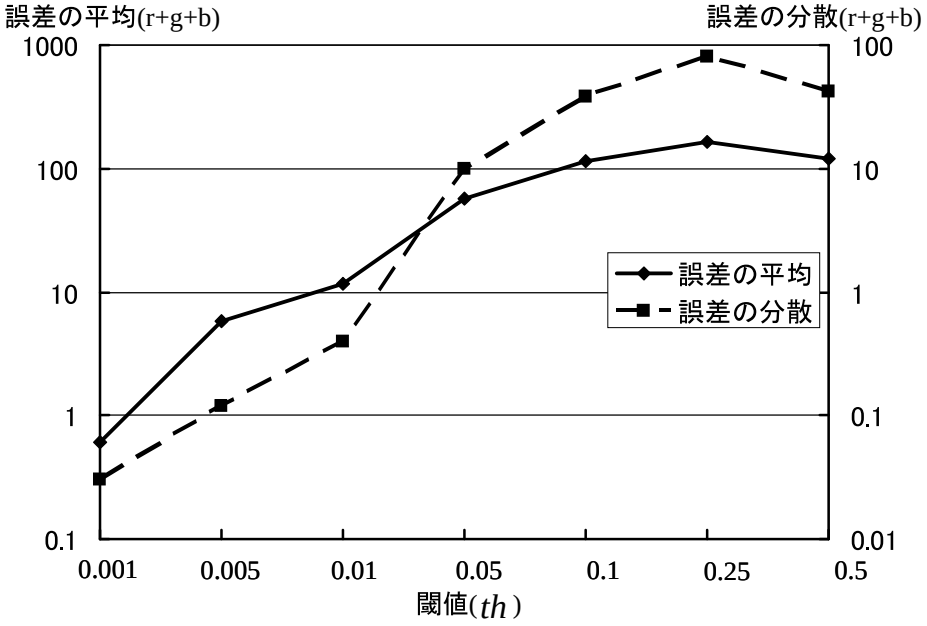
4.2 高速化の影響の検証

3.4 節において，多重スケール環境マップの計算を高速化する手法を示した．しかし，高速化手法を適用する場合と適用しない場合で生成される多重スケール環境マップが異なる可能性がある．そこで，高速化を行わなかった多重スケール環境マップのボクセルの輝度値を真値として，高速化に伴いどの程度の誤差が生じるかを検証する．ここで，誤差は入力となる環境マップに依存するため，4.1 節で用いた図 13(a)，(b) の両方の画像について調べた．さらに，高速化のときに用いた閾値は $th = 0.001, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5$ の 7 通りで計算を行い，計 14 通りの環境マップを作成し比較を行った．また，生成した多重スケール環境マップの解像度は $128 \times 128 \times 128$ である．また，高速化の際には 64×32 までの 4 段階の縮小画像を作成し，測地ドームも頂点数 51842 から 812 までの 4 段階を用意した．画像の解像度と頂点数の対応関係を表 2 に示す．

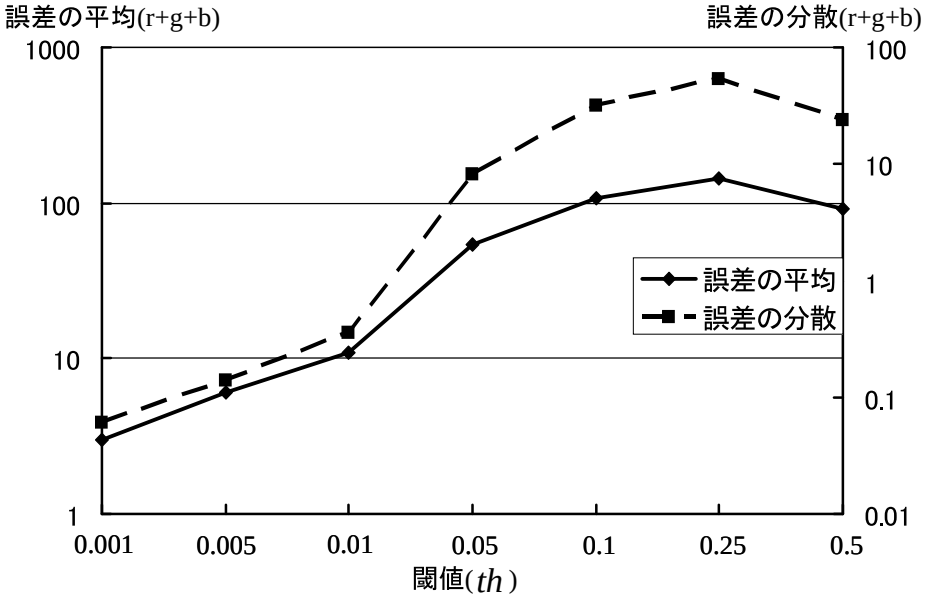
表 3 に，図 13(a),(b) それぞれの環境マップを用いた際の閾値と多重スケール環境マップの誤差の平均および分散の関係を示す．また，図 18 に各色成分 (red, $green$, $blue$) の 256 階調での誤差の和と閾値の関係を示している．また，図 19 に閾値と計算されたボクセル数の関係を示す．これは，全てのボクセルが計算された場合を 1 とし，計算されたボクセル数の割合が 1 でないものは，閾値処理によって計算されなかったボクセルが存在したことを示す．また，それぞれの場合の計算時間と計算に使用した重みテーブルのサイズを表 4 に示す．計算時間は入力となる環境マップに依存しないため図 13(a) の結果のみを示す．

表 3　高速化に伴うボクセルの輝度値の誤差 (256 階調)

入力画像	閾値 (th)	平均 (r, g, b)	分散 (r, g, b)
図 13(a)	0.001	(0.1, 0.2, 0.3)	(0.01, 0.01, 0.01)
	0.005	(1.9, 1.9, 2.0)	(0.03, 0.04, 0.05)
	0.01	(3.8, 3.8, 4.0)	(0.13, 0.13, 0.14)
	0.05	(19, 19, 19)	(3.24, 3.35, 3.37)
	0.1	(38, 38, 38)	(12, 13, 13)
	0.25	(54, 55, 54)	(32, 24, 24)
	0.5	(34, 46, 41)	(10, 18, 14)
図 13(b)	0.001	(1.0, 1.0, 1.0)	(0.02, 0.02, 0.02)
	0.005	(2.0, 2.0, 2.1)	(0.04, 0.05, 0.05)
	0.01	(3.6, 3.6, 3.6)	(0.12, 0.11, 0.13)
	0.05	(17.8, 17.4, 19.2)	(2.6, 2.6, 3.0)
	0.1	(35, 35, 38)	(10, 10, 12)
	0.25	(47, 46, 51)	(17, 17, 20)
	0.5	(30, 29, 32)	(7.6, 7.5, 8.8)



(a) 図 13(a) から作成した多重スケール環境マップの誤差 (256 階調)



(b) 図 13(b) から作成した多重スケール環境マップの誤差 (256 階調)

図 18 高速化に伴うボクセルの輝度値の差と閾値の関係

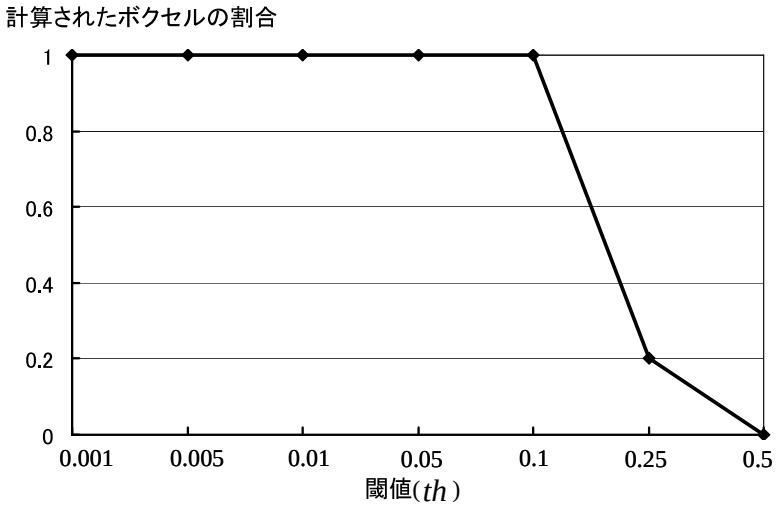


図 19 正しく計算されたボクセル数の割合と閾値の関係

表 4 計算時間の比較

計算手法	閾値 (th)	テーブル作成時間 [h:m:s]	フィルタ計算時間 [h:m:s]	合計時間 [h:m:s]	テーブルサイズ [MByte]
高速化なし	—	—	73:51:54	73:51:54	—
高速化あり	0.001	21:23:41	0:00:09	21:23:50	459
	0.005		0:00:09	21:23:50	258
	0.01		0:00:08	21:23:49	216
	0.05		0:00:08	21:23:49	116
	0.1		0:00:07	21:23:48	74.1
	0.25		0:00:07	21:23:48	35.7
	0.5		0:00:07	21:23:48	32.7

表 3 より閾値に 0.001 程度設定した場合で，誤差の平均値が図 13(a) を用いた際で 0.3 未満，図 13(b) においても 1.0 程度の値で収まっている．一般の表示装置では輝度値の差が 256 階調で 1 未満である場合はその違いを表示することができず，さらに 1 程度の輝度値の違いは人間がその差を知覚することは難しい．よって，高速化に伴って生じた計算誤差は小さく，高速化手法が有効である．また，図 18 より閾値を大きくすることでボクセルの誤差が大きくなっていることが分かる．なお，図 19 において計算されたボクセル数の割合が 1 に満たない閾値が存在しているが，これは計算されなかったボクセルが存在していることを示している．つまり，設定した閾値が大きすぎることによって，閾値よりも大きい重みを持つボクセルと測地ドームの頂点の組が存在しなかったことを示している．これは，閾値の選択が不適切であったことを示し，今回の場合では閾値は 0.1 以下が望ましいといえる．

多重スケール環境マップの計算において，閾値を大きくするに従って誤差が大きくなるが，表 4 のテーブルサイズに着目するとテーブルサイズが減少している．これは，利用する計算機の性能に応じて適切な重みを選択すること可能であることを示している．しかし，今回の表 4 の計算時間からも分かるとおり，500MByte 程度の重みテーブルを利用して多重スケール環境マップを作成することは十分可能である．

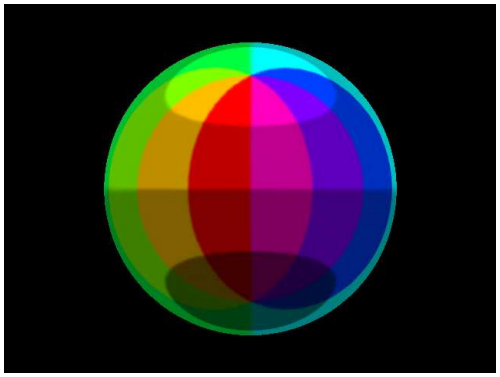
また，表 4 より高速化を行うことによって計算が約 3.5 倍程度早くなっていることが分かる．さらに，重みデータのテーブル化を行うことで，複数の多重スケール環境マップを作成する場合にもフィルタ計算のみを行えば良く，一つの入力環境マップに対して数秒程度の高速な計算が可能である．

4.3 Phongモデルとの比較

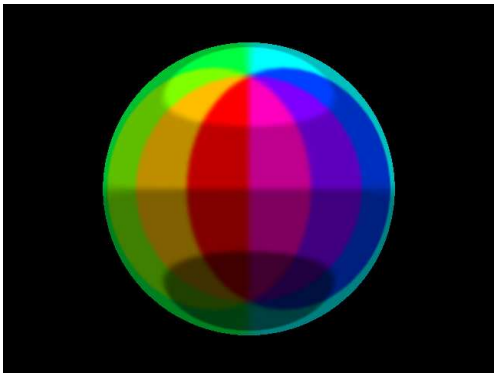
本節では，提案手法によるレンダリング結果がPhongモデルによるレンダリング結果と比べてどの程度の差があるかの検証実験の結果を示す．実験方法として，4.1節で用いた図13の2つの環境マップを用い，粗さの異なる球体とティーポットに対して提案手法とPhongモデルによるレンダリングを行った．そして，Phongモデルによるレンダリングを真値として結果画像の輝度値の平均，分散を示す．また，計算時間の比較を行った．このとき，高速化を行わなかった多重スケール環境マップと，高速化を行った多重スケールマップのそれぞれと比較を行う．また，粗さは$\lambda = 1.0, 0.75, 0.5, 0.25$，不均一なものの合計5通りの実験を行った．なお，提案手法に用いた多重スケール環境マップの解像度は$128 \times 128 \times 128$とし，Phongモデルによるレンダリングは提案手法で指定した粗さ係数λと対応する粗さ係数を与えた．

図20，図21にレンダリング結果と差分画像を示す．図20では図13(a)を用いて$\lambda = 0.5$の粗さを持つ球体に対してレンダリングを行った結果を示し，図21では図13(b)を用いて粗さが不均一なティーポットに対してレンダリングを行った結果を示している．なお，両図において，Phongモデルによるレンダリング結果を(a)，高速化を行わない提案手法によるレンダリング結果を(b)，(a)と(b)の差分画像を(c)，高速化を行った提案手法によるレンダリング結果を(d)，(a)と(d)の差分画像を(e)に示している．

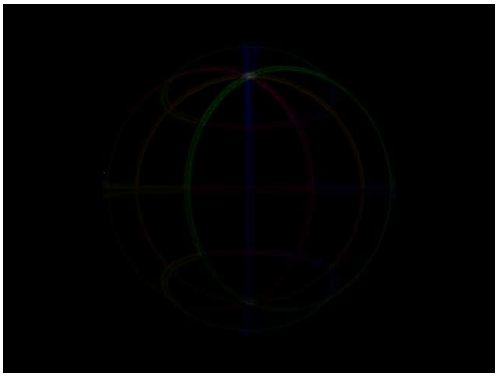
各レンダリング結果について輝度値の比較を行った結果を表5に，提案手法(高速化無し，高速化有り)とPhongモデルの前処理およびレンダリングに要した計算時間について表6に示す．



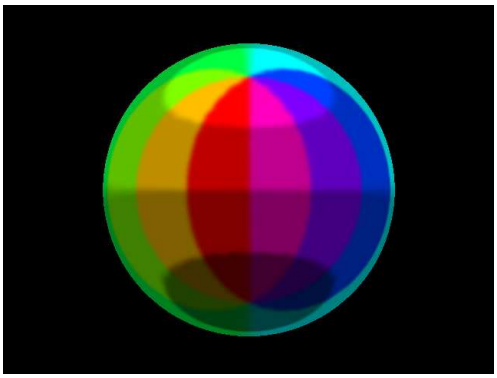
(a) Phong モデルによる結果



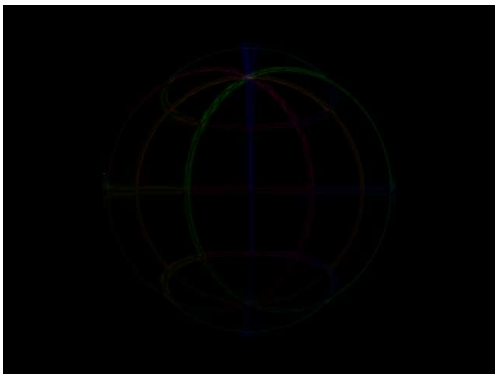
(b) 提案手法 (高速化なし)



(c) Phong モデルとの差分画像



(d) 提案手法 (高速化あり)



(e) Phong モデルとの差分画像

図 20 レンダリング結果の比較 (均一な粗さを持つ球体, $\lambda = 0.5$, カラーパターン使用)

(a) Phong モデルによる結果

(b) 提案手法 (高速化なし)

(c) Phong モデルとの差分画像

(d) 提案手法 (高速化あり)

(e) Phong モデルとの差分画像

図 21　レンダリング結果の比較　(不均一な粗さを持つティーポット，実写画像使用)

表 5 Phongモデルによるレンダリング結果との比較

(a) 球体を用いた比較

高速化の有無	粗さ	図 13(a)		図 13(b)	
		誤差の平均	誤差の分散	誤差の平均	誤差の分散
無し	$\lambda = 1.0$	(0.4, 0.3, 0.4)	(0.02, 0.03, 0.03)	(2.0, 2.0, 2.0)	(0.18, 0.18, 0.17)
	$\lambda = 0.75$	(0.5, 0.4, 0.5)	(0.02, 0.03, 0.03)	(2.4, 2.4, 2.4)	(0.21, 0.21, 0.20)
	$\lambda = 0.5$	(0.7, 0.6, 0.6)	(0.03, 0.03, 0.04)	(2.9, 2.9, 2.9)	(0.28, 0.28, 0.27)
	$\lambda = 0.25$	(1.2, 1.1, 1.1)	(0.06, 0.06, 0.07)	(3.4, 3.4, 3.5)	(0.33, 0.34, 0.34)
	不均一	(1.0, 1.0, 0.9)	(0.13, 0.07, 0.07)	(3.3, 3.4, 3.4)	(0.34, 0.34, 0.35)
有り	$\lambda = 1.0$	(0.4, 0.4, 0.3)	(0.02, 0.03, 0.02)	(2.2, 2.2, 2.2)	(0.20, 0.20, 0.19)
	$\lambda = 0.75$	(0.5, 0.4, 0.4)	(0.02, 0.03, 0.02)	(2.5, 2.5, 2.5)	(0.23, 0.23, 0.22)
	$\lambda = 0.5$	(0.7, 0.6, 0.6)	(0.03, 0.03, 0.03)	(3.0, 3.0, 3.0)	(0.30, 0.30, 0.29)
	$\lambda = 0.25$	(1.3, 1.1, 1.1)	(0.06, 0.06, 0.07)	(3.5, 3.6, 3.6)	(0.37, 0.38, 0.37)
	不均一	(1.3, 1.1, 1.1)	(0.10, 0.07, 0.06)	(3.7, 3.7, 3.7)	(0.39, 0.39, 0.40)

(b) ティーポットを用いた比較

高速化の有無	粗さ	図 13(a)		図 13(b)	
		誤差の平均	誤差の分散	誤差の平均	誤差の分散
無し	$\lambda = 1.0$	(0.3, 0.3, 0.3)	(0.02, 0.02, 0.03)	(2.0, 2.0, 2.0)	(0.18, 0.18, 0.17)
	$\lambda = 0.75$	(0.4, 0.3, 0.4)	(0.02, 0.02, 0.03)	(2.4, 2.4, 2.4)	(0.21, 0.21, 0.21)
	$\lambda = 0.5$	(0.6, 0.5, 0.7)	(0.02, 0.03, 0.03)	(2.9, 2.9, 2.9)	(0.28, 0.28, 0.28)
	$\lambda = 0.25$	(1.1, 0.9, 1.0)	(0.05, 0.05, 0.06)	(3.3, 3.3, 3.4)	(0.32, 0.34, 0.33)
	不均一	(1.0, 0.8, 0.9)	(0.04, 0.04, 0.05)	(3.0, 3.1, 3.1)	(0.30, 0.31, 0.30)
有り	$\lambda = 1.0$	(0.4, 0.3, 0.3)	(0.02, 0.02, 0.02)	(2.2, 2.2, 2.2)	(0.20, 0.20, 0.19)
	$\lambda = 0.75$	(0.5, 0.4, 0.4)	(0.02, 0.02, 0.02)	(2.4, 2.5, 2.5)	(0.23, 0.23, 0.22)
	$\lambda = 0.5$	(0.6, 0.5, 0.5)	(0.03, 0.03, 0.03)	(3.0, 3.0, 3.0)	(0.30, 0.31, 0.30)
	$\lambda = 0.25$	(1.1, 0.9, 1.0)	(0.05, 0.05, 0.06)	(3.5, 3.5, 3.6)	(0.36, 0.38, 0.37)
	不均一	(1.0, 0.8, 0.9)	(0.04, 0.04, 0.05)	(3.3, 3.3, 3.4)	(0.34, 0.35, 0.35)

表 6 Phong モデルとの計算時間の比較

手法	多重スケール環境マップの作成時間 [h:m:s]	レンダリング時間 [h:m:s]
提案手法 (高速化無し)	73:51:54	< 0:00:0.017(60fps)
提案手法 (高速化有り)	21:23:49	< 0:00:0.017(60fps)
Phong モデル	—	1:03:05

Phong モデルとの比較実験では表 5 より，提案手法が Phong モデルと比較して輝度値にして 1 程度で近似できていることが分かる．この誤差は，人間が見た場合にはほとんど違和感がないと考えることができる．そして，2 つの物体 (球体，ティーポット) に対してレンダリングを行ったが，表 5(a),(b) よりどちらの場合も似たような結果が得られていることが分かる．これによって，本手法が様々な物体の写り込みのレンダリングの際に適用可能であると考えられる．

しかし，図 21(c),(e) のティーポットの上部 (滑らか) と下部 (粗い) を比較すれば分かるが，材質の粗さが粗いほど (λ が小さいほど)，誤差が大きくなる傾向がある．また，表 5 からも，λ が小さいほど誤差が大きくなっていることが分かる．これは，多重スケール環境マップに粗さに応じた写り込み情報を保持する際の解像度不足が原因であると考えられる．特に $\lambda = 0.2$ 程度の非常に粗い材質に対応する写り込みは，全天球の写り込みの保持に十数ボクセル程度で保存されているため，解像度不足による影響が大きいと考えられる．

誤差が生じている部分としては，まず，物体の法線が視線方向と 90 ° に近い角度になっている部分 (たとえば球体の外周付近) が挙げられる．これは，提案手法で用いた Phong モデルに対する近似による影響であると考えられ，図 8 のような状況が発生していると考えられる．しかしながら，画像全体の誤差を見ても人が目で見ても気づけないような小さな値であるため，影響は無視できると言える．さらに，写り込み画像のエッジ付近でも誤差が生じているが，これは多重スケール環境マップを用いてテクスチャマッピングする際の線形フィルタによる影響と

考えられる．

計算時間に関しては，提案手法では前処理として，多重スケール環境マップの作成に高速化を行わない場合で 73 時間，高速化を行って 21 時間程度かかる．しかし，いったん多重スケール環境マップを作成すれば 60fps 以上でレンダリングを行うことが可能である．対して Phong モデルによるレンダリングでは，1 フレームあたり 1 時間程度を要するため，写り込みの高速なレンダリングは困難である．

4.4 動画像を利用した写り込みのレンダリング

全方位マルチカメラシステム Ladybug[33] を用いて撮影された全周パノラマ動画像を用いて，動きのあるシーン内での仮想物体に対する写り込みのレンダリングを行った実験結果を示す．動画に拡張する際には，多重スケール環境マップの計算をあらかじめ行い蓄積しておき，レンダリングの際に多重スケール環境マップを動的に更新することで，動画となったシーンのレンダリングを行った．シーケンスは 2 つ用意し，それぞれ 15fps の動画であり，長さは 695 フレーム (46 秒程度) である．仮想物体として，時刻によって粗さが変化する球体と 4.1 節で利用したティーポットを用意した．なお，多重スケール環境マップの解像度は $64 \times 64 \times 64$ であり，背景画像の解像度は 512×256 とした．また，2 並列処理を行っており多重スケール環境マップおよび背景をディスクからロードするスレッドと，レンダリングするスレッドを実行している．

図 22 に粗さが時間に応じて変化する球体に対するレンダリング結果を示す．453 フレームの時点から徐々に表面が粗くなり，再び滑らかになる様子を示している．また，図 23 に粗さが不均一なティーポットに対する写り込みの結果を示す．動きのあるシーンに対しても写り込みのレンダリングを行うことが可能であった．動画像の更新はキャプチャ時と同じ 15fps で行い，視点変更は 60fps 以上で操作可能であった．なお，695 フレームの多重スケール環境マップを作成するのに要した時間は約 45 分程度であった．

本実験で，動きのあるシーンに対して写り込みのレンダリングが可能であることを確認した．しかし，ディスクからの読みとり帯域の制限によって，高解像度の背景，および多重スケール環境マップを利用することができなかった．そこで，

既存の動画像の圧縮方法などを利用することでディスクの帯域を削減すること等が考えられる.

(a) 第 255 フレーム ($\lambda = 1.0$)

(b) 第 260 フレーム ($\lambda = 0.75$)

(c) 第 265 フレーム ($\lambda = 0.5$)

(d) 第 270 フレーム ($\lambda = 0.25$)

(e) 第 275 フレーム ($\lambda = 0.5$)

(f) 第 280 フレーム ($\lambda = 0.9$)

図 22　シーン 2 内でのレンダリング結果 (粗さの変化する球体)

(a) 第 90 フレーム

(b) 第 95 フレーム

(c) 第 100 フレーム

(d) 第 105 フレーム

(e) 第 110 フレーム

(f) 第 115 フレーム

図 23　シーン 2 内でのレンダリング結果 (粗さが不均一なティーポット)

4.5 実験結果のまとめと考察

実験では，提案手法によるレンダリング結果を示し，粗さの異なる材質に対して写り込みが高速にレンダリングできていることを確認した．粗さ係数を変化させた材質に対してレンダリングを行うことで，提案手法が材質の粗さに応じた写り込みをレンダリングすることが可能であることを確認した．また，粗さが不均一な材質に対する写り込みのレンダリングにより，シーン中に複数の粗さが存在するような場合であっても高速にレンダリング可能なことを確認した．また，高速化手法の有効性を確認した．高速化の手法を用いることで，多重スケール環境マップの作成を高速化を行わない場合に比べておよそ 3.5 倍高速に計算することが可能であった．また，重みをテーブルとして保持しておくことで，複数の多重スケール環境マップを作成する際，高速に多重スケール環境マップを作成することが可能であった．そして，提案手法が Phong モデルのレンダリング結果と比較して輝度値にして 1 程度で近似できていることが確認できた．また，計算時間において，Phong モデルが 1 フレームあたり 1 時間かかるのに対し，提案手法では実時間でのレンダリング可能であった．さらに，動画像を用いた写り込みのレンダリング結果を示し，提案手法の応用例を示した．あらかじめ多重スケール環境マップを作成しておくことで，動きのある環境内に置かれた仮想物体に対して写り込みをレンダリングすることが可能であった．

しかし，提案手法を用いて粗さが不均一な材質をレンダリングする場合，粗さ情報を頂点単位でしか設定できない．これは，粗さが複雑に変化する物体を違和感なく表現する際に，細密なモデルを用意する必要があることにつながる．また，頂点単位での粗さの設定は仮想物体に対して粗さ情報を与える際の困難さつながる．この問題を解決するためのアイディアとして，テクスチャに対して粗さ情報を埋め込み，画素単位での粗さの計算を行う方法が考えられる．これによって仮想物体の編集がより容易になると考えられる．また，多重スケール環境マップはボクセルデータとして保持される．このため，高い解像度の写り込みをレンダリングするためには，高解像度のボクセルデータが必要となる．しかし，高解像度のボクセルデータを持つことはメモリ使用量の拡大につながる．このため，メモリの使用量を発散させることなく高解像度の写り込みを実現することが課題となる．

5. むすび

本論文では，あらかじめ粗さに応じた写り込み情報を計算することで，粗さが不均一である材質に対する写り込みを高速にレンダリングする手法を示した．提案手法では，まず環境マップに対してあらかじめ Phong モデルによる材質の粗さに応じたフィルタ処理を行うことで，複数の粗さが表現可能な多重スケール環境マップを作成する．そして，多重スケール環境マップをテクスチャとして用いることで写り込みの実時間レンダリング手法を提案した．さらに，ピラミッド型画像を用いることでフィルタ計算の簡略化を行い，計算に利用する重みをテーブル化することによる計算の高速化を行った．実験では表面の粗さが不均一な材質に対しても実時間でレンダリングが可能であることを確認し，高速化の手法が多重スケール環境マップの作成結果に大きな影響を与えることなく計算の高速化を行えることを確認した．そして，提案手法におけるレンダリング結果と Phong モデルによるレンダリング結果との比較を行うことで，提案手法が Phong モデルによる写り込みを輝度値の誤差が 256 階調で 1 程度で近似できていることを確認した．また，多重スケール環境マップを動画に拡張することで，動きのあるシーン内での写り込みのレンダリングが可能であることを確認した．

本手法を用いることで，インタラクティブ性が要求される CG において，表面が滑らかな材質だけでなく，表面が粗い材質も表現することが可能になる．実際の応用例としては，ヴァーチャルリアリティによる仮想空間のウォークスルーや，ゲームなどのアミューズメント等において，仮想物体の表示に本手法を適用することが挙げられる．このような用途では，あらかじめ用意されたシーンのみを扱うことが多いため，各シーンごとにあらかじめ多重スケール環境マップを作成しておけばよい．提案手法を用いることで，より写実的な仮想環境を構築することが可能になると考える．

今後の課題としては，多重スケール環境マップを用いて詳細な画像の写り込みをレンダリングする際には，ボクセルデータ形式のメモリ消費量の大きさが問題として挙げられる．そこで，多重スケール環境マップに対して効果的な圧縮を行う方法や，他の環境マッピングと組み合わせることでより鮮明な写り込みのレンダリングを行うことなどが考えられる．また，提案手法においては，頂点単位で

しか粗さの設定が行えない．そこで，テクスチャに粗さ情報を埋め込むことで画素単位で粗さの設定を行えるようにする必要があると考えられる．さらに，多重スケール環境マップによる粗さの表現は直感的では無いという点がある．たとえば，表面粗さ係数を 1 から 0.75 に変化させた場合と，0.5 から 0.25 に変化させた場合で，写り込みがぼやける度合いが違う．このため，実際に本手法を用いてレンダリングに用いる際に粗さ係数を決定するために試行錯誤を繰り返す必要があり，利用しづらいといえる．そこで，今後は多重スケール環境マップにおける粗さ表現を利用しやすい形に表現し直すなどの改良も必要である．

謝辞

本研究を進めるにあたり，多大なご助言，御指導を頂きました 視覚情報メディア講座 横矢 直和 教授に厚く御礼申し上げます．副指導教官として御助言を頂いた像情報処理講座 千原 國宏 教授，並びに 視覚情報メディア講座 山澤 一誠 助教授，視覚情報メディア講座 神原 誠之 助手に深く感謝致します．

また，本研究に関する貴重な助言や指摘を頂きました 大阪大学サイバーメディアセンタ 町田 貴史 助手，並びに 視覚情報メディア講座 佐藤 智和 氏 に深く感謝致します．日々の研究室の活動を支えて下さった北川 知代 女史に心より感謝申し上げます．視覚情報メディア講座の諸氏には，本研究を進めるにあたり多大なる御協力を頂きました．ここに感謝の意を表します．

参考文献

[1] E. Lindholm, M. J. Kilgard and H. Moreton: "A User-Programable Vertex Engine," Proc. SIGGRAPH '01, pp. 149-158, 2001.

[2] H. W. Jansen, S. R. Marschner, M. Levoy and P. Hanrahan: "A Practical Model for Subsurface Light Transport," Proc. SIGGRAPH '01, pp. 511-518, 2001.

[3] T. Lokovic and E. Veach: "Deep Shadow Maps," Proc. SIGGRAPH '00, pp. 385-392, 2000.

[4] R. Fernando, S. Fernandez, K. Bara and D. P. Greenberg: "Adaptive Shadow Maps," Proc. SIGGRAPH '01, pp. 387-390, 2001.

[5] Y. Y. Chuang, D. E. Zongker, J. Hindorff, B. Curless, D. H. Salesin and R. Szeliski: "Environment Matting Extentions: Twords Higher Accuracy and Real-Time Capture," Proc. SIGGRAPH '00, pp. 121-130, 2000.

[6] M. D. McCool, J. Ang and A. Ahmad: "Homomorphic Factorization of BRDFs for High-Performance Rendering," Proc. SIGGRAPH '01, pp. 171-178, 2001.

[7] 奥村 文洋, 町田 貴史, 横矢 直和: "多重スケール環境マップを用いた粗さの異なる材質を表現可能な実時間レンダリング手法", 日本バーチャルリアリティ学会第 7 回大会論文集, pp. 381-382, 2002.

[8] 奥村 文洋, 町田 貴史, 横矢 直和: "実写に基づく全周多重スケール環境マップを用いた不均一物体への写りこみの高速レンダリング", 電子情報通信学会技術研究報告, PRMU2002-137, 2002.

[9] J. Blinn and M. Newell: "Texture and Reflection in Computer Generated Images," Commun. of the ACM 19, pp. 542-546, 1976.

[10] G. Miller and R. Hoffman: "Illumination and Reflection Maps: Simulated Objects in Simulated and Real Environments," Proc. SIGGRAPH '84, Course Notes – Advanced Computer Graphics Animation, 1984.

[11] B. Cabral, M. Olano and P. Nemec: "Reflection Space Image Based Rendering," Proc. SIGGRAPH '99, pp. 165-170, 1999.

[12] W. Heidrich and H. P. Seidel: "Realistic, Hardware-accelerated Shading and Lighting," Proc. SIGGRAPH '99, pp. 171-178, 1999.

[13] J. Kautz, P. Vázquez, W. Heidrich and H. P. Seidel: "A Unified Approach to Prefiltered Environment Maps," Proc. Euro Graphics Rendering Workshop '00, pp. 185-196, 2000.

[14] G. J. Ward: "Measuring and Modeling Anisotropic Reflection," Proc. SIGGRAPH '92, Vol. 26, pp. 256-272, 1992.

[15] J. D. Foley, A. V. Dam, S. K. Feiner and J. F. Hughes: *Computer Graphics: Principles and Practice - second edition*, Addison-Wesley Publishing Company, Massachusetts, 1993.

[16] B. T. Phong: "Illumination for Computer Generated Pictures," Commun. of the ACM, Vol. 18, No. 6, pp. 311-317, 1975.

[17] K. E. Torrance and E. M. Sparrow: "Theory for Off-specular Reflection from Roughened Surfaces," Jour. Optical Society of America, Vol. 57, pp. 1105-1114, 1967.

[18] P. Debevec: "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography," Proc. SIGGRAPH '98, pp.189-198, 1998.

[19] R. Ramamoorthi and P. Hanrahan: "An Efficient Representation for Irradiance Environment Maps," Proc. SIGGRAPH '01, pp.497-500, 2001.

[20] A. Appel: "Some Techniques for ShadingMachine Rendering of Solids," Proc. AFIPS Joint Computer Conference, pp. 37-45, 1968.

[21] Mathematical Applications Group, Inc.: "3-D Simulated Graphics Offered by Service Bureau," *Datamation*, 1968.

[22] J. T. Kajiya: "The Rendering Equation," Proc. SIGGRAPH '86, pp. 143-150, 1986.

[23] M. C. Goral, K. E. Torrance, D. P. Greenberg and B. Battaile: "Modeling the Interaction of Light Between Diffuse Surfaces," Proc. SIGGRAPH '84, pp. 213-222, 1984.

[24] T. Nishita, and E. Nakamae: "Continuous Tone Representation of Three Dimensional Objects Taking Account of Shadows and Interreflection," Proc. SIGGRAPH '85, pp. 23-30, 1985.

[25] R. Cook and K. Torrance: "A Reflectance Model for Computer Graphics," ACM Transactions on Graphics, Vol. 1, No. 1, pp. 51-72, 1986.

[26] H. W. Jansen: "Realistic Image Synthesis Using Photon Mapping," *A K Peters, Ltd.*, 2001.

[27] W. Heidrich and H. P. Seidel: "View-independent Environment Maps," Proc. Eurographics/Siggraph Workshop on Graphics Hardware '98, pp. 39-45, 1998.

[28] Douglas Voorhies and Jim Foran: "Reflection vector shading hardware," Proc. SIGGRAPH '94, pp. 163-166, 1994.

[29] Microsoft DirectX 9 ソフトウェア開発キット, http://www.microsoft.com/japan/msdn/directx/default.asp.

[30] J. Kautz and M Mccool: "Approximation of Glossy Reflection with Prefiltered Environment Maps," Proc. Graphics Interface '00, pp. 119-126, 2000.

[31] P. Diefenbach and N. Badler: “Multi-Pass Pipeline Rendering: Realism For Dynamic Environments,” Proc. ACM Symposium on Interactive 3D Graphics, pp. 59-70, 1997.

[32] N. Greene: “Environment Mapping and Other Applications of World Projections”, IEEE Computer Graphics & Applications , Vol. 6, No. 11, pp. 21-29, 1986.

[33] 池田 聖, 佐藤 智和, 横矢 直和: “全方位型マルチカメラシステムによるパノラマ動画像の生成”, 電子情報通信学会 技術研究報告, PRMU2002-154, 2002.