

球近似と符号付き球面距離関数による動的シーンのリアルタイムレンダリング

古家 互¹ 岩崎 慶^{1,a)} 土橋 宜典² 西田 友是³

受付日 2011年8月28日, 採録日 2012年4月2日

概要: 本研究では, 環境照明下における動的なシーンをリアルタイムにレンダリングする手法を提案する. 環境照明下の物体表面の輝度を計算するためには, 環境照明, 物体表面上の計算点における可視関数, および BRDF の積を積分計算する必要がある. 本研究では, 環境照明および BRDF を球面ガウス関数で表現し, 可視関数を符号付き球面距離関数 (Spherical Signed Distance Function: SSDF) で表現する. これにより, 高速な輝度計算が可能となるが, SSDF の計算は計算コストが高いため, 従来手法では前計算時に処理しており動的なシーンへの適用ができなかった. 本研究では, 物体を球の集合で近似することにより, 高速に SSDF を計算する手法を提案する. これにより, 視点, 照明, BRDF および物体の移動や回転, 変形を含む動的なシーンをリアルタイムにレンダリングすることができる.

キーワード: リアルタイムレンダリング, GPU, イメージベースライティング, 動的シーン, 球面ガウス関数

Real-time Rendering of Dynamic Scenes Using Sphere Set Approximation and Spherical Signed Distance Function

WATARU FURUYA¹ KEI IWASAKI^{1,a)} YOSHINORI DOBASHI² TOMOYUKI NISHITA³

Received: August 28, 2011, Accepted: April 2, 2012

Abstract: We propose a real-time rendering method for dynamic scenes under all-frequency environmental lighting. To render the object surfaces illuminated by environmental lighting, the product of the environmental lighting, the visibility function at each shading point on the object surfaces, and the BRDF is integrated. Our method represents the environmental lighting and the BRDFs with spherical Gaussians and represents the visibility functions with spherical signed distance functions (SSDFs). Although these representations enable us to render scenes under all-frequency lighting in real-time, the previous method is limited to static scenes since the computational costs of SSDFs are quite high. To address this problem, we propose an efficient calculation method of SSDFs by approximating the object geometries with a set of spheres. This makes it possible to render fully dynamic scenes where the viewpoints, lighting, and BRDFs can be changed at run-time and objects can move, rotate and be deformed in real-time.

Keywords: real-time rendering, GPU, image-based lighting, dynamic scene, spherical Gaussian

1. はじめに

コンピュータグラフィックスの分野において, 高精細で

フォトリアルな画像を高速にレンダリングする研究は, 重要な研究課題の1つとして, 現在まで多くの研究がなされてきた. 近年, 光源として実際の画像 (環境マップ) を利用し, 現実世界の複雑な環境照明下で現実感のあるシーンをレンダリングする Image-based Lighting 手法の研究がさかに行われている. 環境マップで表現された照明を考慮したレンダリングでは, 全方位からの入射光を考慮する必要があるが, 計算量が非常に多いため, 高速にレンダリングすることは困難であった.

¹ 和歌山大学
Wakayama University, Wakayama 640-8510, Japan
² 北海道大学
Hokkaido University, Sapporo, Hokkaido 060-8628, Japan
³ 東京大学
The University of Tokyo, Kashiwa, Chiba 277-8561, Japan
a) iwasaki@sys.wakayama-u.ac.jp

Sloan らは、物体表面での光の遮蔽や反射といった伝達関数を前計算しておく Pre-computed Radiance Transfer (PRT) を提案し、環境照明下のシーンのリアルタイムレンダリングを可能にした [1]。この手法を拡張する手法が多く提案されている。Wang らは、環境照明を球面放射基底関数 (Spherical Radial Basis Function, SRBF) の一種である球面ガウス関数を用いて近似することにより、レンダリング時に視点・物体表面の反射特性 (BRDF)・照明を変更可能な手法を提案した [2]。この手法では、物体による遮蔽情報を符号付き球面距離関数 (Spherical Signed Distance Function, SSDF) として表現する。球面ガウス関数と SSDF との演算を効率的に行うことによってリアルタイムレンダリングを実現しているが、SSDF の計算には時間を要するため、シーンを固定し SSDF を前計算している。そのため、Wang らの手法 [2] は動的シーンには対応できない。

本研究では、SSDF をレンダリング時に計算することにより、動的なシーンをリアルタイムレンダリングする手法を提案する。提案法の特徴としては以下の点があげられる。

- 物体を球で近似することにより、SSDF をリアルタイムに計算。
- 視点・照明・物体の位置・物体の形状・BRDF をリアルタイムに変更可能。
- 複雑な環境照明および鋭い鏡面反射材質に対応することにより、高精細な画像の生成が可能。

2. 関連研究

PRT 法は、物体表面における光の伝達を前計算し、球面調和関数 [1], [3], [4] や Haar Wavelet [5] といった基底関数で伝達関数を近似することにより、環境照明下のシーンをリアルタイムにレンダリングする手法である。これらの手法では、基本的にシーンは固定されており、前処理の時点で遮蔽情報を計算するため、レンダリング時にシーン内の物体を動的に動かすことができなかった。その後、PRT 手法は拡張され、環境照明下における動的シーンの高速レンダリング手法がいくつか提案されてきた。

Zhou らは物体による遮蔽情報を、物体を中心とした同心球上で記録することにより、剛体の移動を考慮した動的シーンを高速にレンダリングする手法を提案した [3]。Ren らは、球面調和関数の指数関数演算である SHEXP 演算を導入し、物体を球の集合で近似することによって、剛体だけでなく、変形物体をリアルタイムにレンダリングする手法を提案した [4]。しかしながらこれらの手法は低周波環境照明にのみ対応しており、非常にぼやけた影しか表現することができなかった。

Green らは鏡面反射成分を球面ガウス関数で近似することにより全周波環境照明下におけるリアルタイムレンダリング手法を提案した [6]。Tsai らは PRT データを SRBF

に射影し Clustered Tensor Approximation (CTA) を利用して圧縮する手法を提案した [7]。この手法は環境照明や BRDF、可視関数を SRBF の集合により近似する。SRBF は高周波信号を表現可能であり、全周波環境照明によるリアルタイムレンダリングを実現した。しかしながらこれらの手法は静的なシーンに限定されている。

Xu らは Spherical Piecewise Constant Basis Function (SPCBF) を用いる PRT 手法を提案した [8]。この手法は環境照明や BRDF、可視関数を SPCBF の集合により近似する。SRBF や球面ガウス関数と同様、SPCBF も高周波信号を表現可能であり、動的なシーンの全周波環境照明によるリアルタイムレンダリングが可能である。しかしながら、この手法は剛体物の移動や回転のみを考慮しており、変形物体には対応していない。

Ritschel らは、シーン中に多くの Virtual Point Light (VPL) を設置し、数百もの Imperfect Shadow Maps (ISMs) を高速に生成することにより、全周波環境照明下の複雑で動的なシーンの高速なレンダリングと間接照明を実現した [9]。しかしながら、高フレームレートを保つには、低解像度の ISM を使用する必要がある。そのため、可視判定に誤りが生じる場合がある。また、インタラクティブな速度を実現しているとはいえ、リアルタイムでの実行は難しい。

Křivánek らは、インポートランスサンプリングを利用した環境照明下のレンダリング手法を提案した [10]。この手法では少数のシャドウマップを用いて可視関数を近似しているため、影の計算が正確ではない場合がある。Annen らは、全周波環境照明下における動的シーンのリアルタイムレンダリング手法を提案した [11]。この手法では、環境照明を複数の面光源の集合で近似し、各面光源による影をシャドウマップによってレンダリングする手法である。この方法は輝度計算において BRDF と照明の積分計算を行わず、面光源の中心方向での BRDF の値で計算しているため、大きな面光源で照射された場合に誤差が大きくなる。

3. 提案法の概要

本論文では、視点・物体の位置や形状・照明・BRDF をレンダリング時に変更可能な動的シーンのリアルタイムレンダリング手法を提案する。また、入射光として環境照明を考慮する。提案法では、BRDF を拡散反射成分と鏡面反射成分に分けて輝度計算を行う。拡散反射成分の計算では、光源を少数の球面ガウス関数で近似する。鏡面反射成分の計算では、異なるローブの鋭さを持つ球面ガウス関数について畳み込み演算した prefiltered environment maps [2] を利用する。拡散反射成分および鏡面反射成分のどちらの計算においても、球面ガウス関数と可視関数の積の積分が必要となる。Wang ら [2] はこの計算に SSDF を利用していたが、SSDF を事前に計算していたため動的シーンへの対

応はできなかった. 提案法では, 物体を球の集合で近似することによってレンダリング時に SSDF を計算する. GPU を用いて, 頂点単位で各球面ガウス関数の SSDF の値を計算し, ピクセル単位で SSDF の値を補間して輝度を計算する.

4. 提案法

物体表面上の位置 \mathbf{x} における射出方向 \mathbf{o} の輝度 $R(\mathbf{x}, \mathbf{o})$ は以下の式 (1) で計算される.

$$R(\mathbf{x}, \mathbf{o}) = \int_{\Omega} L(\mathbf{i})V(\mathbf{x}, \mathbf{i})\rho(\mathbf{i}, \mathbf{o}) \cos \theta d\mathbf{i} \quad (1)$$

ここで, L は環境照明, V は \mathbf{x} における可視関数, ρ は BRDF, θ は \mathbf{x} における法線 \mathbf{n} と入射方向 \mathbf{i} とのなす角とし, Ω は半球上の方向の集合とする. BRDF ρ を拡散反射成分 k_d と鏡面反射成分 $k_s\rho_s(\mathbf{i}, \mathbf{o})$ に分離する.

$$\rho(\mathbf{i}, \mathbf{o}) = k_d + k_s\rho_s(\mathbf{i}, \mathbf{o}) \quad (2)$$

ここで, 式 (2) を式 (1) に代入すると, 以下のように変形される.

$$R(\mathbf{x}, \mathbf{o}) = k_d R_d(\mathbf{x}) + k_s R_s(\mathbf{x}, \mathbf{o}) \quad (3)$$

$$R_d(\mathbf{x}) = \int_{\Omega} L(\mathbf{i})V(\mathbf{x}, \mathbf{i}) \cos \theta d\mathbf{i} \quad (4)$$

$$R_s(\mathbf{x}, \mathbf{o}) = \int_{\Omega} L(\mathbf{i})V(\mathbf{x}, \mathbf{i})\rho_s(\mathbf{i}, \mathbf{o}) \cos \theta d\mathbf{i} \quad (5)$$

本研究では, Wang ら [2] と同様に直接光の輝度 $R(\mathbf{x}, \mathbf{o})$ を式 (4), (5) を用いて拡散反射成分 $R_d(\mathbf{x})$ と鏡面反射成分 $R_s(\mathbf{x}, \mathbf{o})$ に分けて計算する.

4.1 球面ガウス関数

照明 L , BRDF ρ , 余弦項 $\cos \theta$ は, 球面放射基底関数 (Spherical Radial Basis Function) の一種である球面ガウス関数 $G(\mathbf{i}, \boldsymbol{\zeta}, \eta, \mu)$ で表現する. 球面ガウス関数は以下のように表される.

$$G(\mathbf{i}, \boldsymbol{\zeta}, \eta, \mu) = \mu e^{\eta(\mathbf{i} \cdot \boldsymbol{\zeta} - 1)} \quad (6)$$

ここで, $\boldsymbol{\zeta}$ は球面ガウス関数のローブの中心軸を表す単位ベクトル, η はローブの鋭さを表すパラメータ, μ は球面ガウス関数の振幅であり RGB 3 成分からなる.

球面ガウス関数は回転操作が容易であり, ローブ中心軸 $\boldsymbol{\zeta}$ のみを回転させることにより回転後の関数を表すことができる. また, 高周波信号を表現可能であり, 鋭い鏡面反射を表現することができる.

4.2 符号付き球面距離関数 SSDF

可視関数 $V(\mathbf{x}, \mathbf{i})$ は, 入射方向 \mathbf{i} に遮蔽物があるかどうかを返す球面二値関数である. 本研究では, 可視関数 $V(\mathbf{x}, \mathbf{i})$ を符号付き球面距離関数 SSDF $V^d(\mathbf{x}, \mathbf{i})$ へと変換する. SSDF $V^d(\mathbf{x}, \mathbf{i})$ は以下の式 (7) により与えられる.

$$V^d(\mathbf{x}, \mathbf{i}) = \begin{cases} + \min_{V(\mathbf{x}, \mathbf{t})=0} \cos^{-1}(\mathbf{i} \cdot \mathbf{t}) & \text{if } V(\mathbf{x}, \mathbf{i}) = 1 \\ - \min_{V(\mathbf{x}, \mathbf{t})=1} \cos^{-1}(\mathbf{i} \cdot \mathbf{t}) & \text{if } V(\mathbf{x}, \mathbf{i}) = 0 \end{cases} \quad (7)$$

ここで, \mathbf{t} は入射ベクトル \mathbf{i} に最も近い遮蔽境界へ方向ベクトルである (図 1 参照). SSDF は, 方向 \mathbf{i} に遮蔽物がない場合 (図 1 (a)), 遮蔽されている方向ベクトルの中で最近傍のベクトル \mathbf{t} とのなす角を返し, 方向 \mathbf{i} に遮蔽物がある場合 (図 1 (b)), 遮蔽されていない方向ベクトルの中で最近傍のベクトル \mathbf{t} とのなす角にマイナスをかけたものを返す. 図 1 (c) は Stanford Bunny モデルをシーン内のある位置 \mathbf{x} から見たときの SSDF を, 法線ベクトル \mathbf{n}_x を中心に上半球部分を 256×256 ピクセルのキューブマップとして可視化したものである. 画像中の各ピクセルは, SSDF を求める位置 \mathbf{x} から伸ばしたベクトル \mathbf{i} の半球面上の位置に対応している. また, ピクセルの色は以下のように描画した.

$$(R, G, B) = \left(\max \left(\frac{255 \times V^d(\mathbf{x}, \mathbf{i})}{0.5\pi}, 0 \right), \max \left(\frac{255 \times (-V^d(\mathbf{x}, \mathbf{i}))}{0.5\pi}, 0 \right), 0 \right) \quad (8)$$

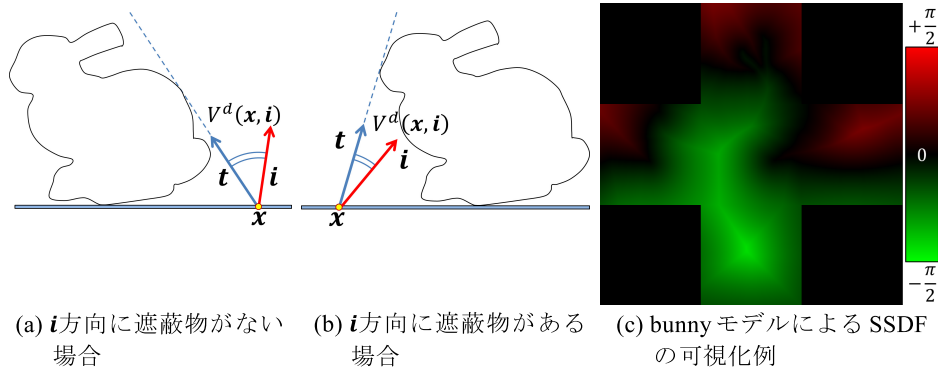


図 1 符号付き球面距離関数 SSDF
Fig. 1 Spherical Signed Distance Function SSDF.

すなわち、赤は遮蔽されていない領域、緑は遮蔽されている領域をそれぞれ表しており、各色の濃淡と角度の関係は図 1 (c) のカラーバーのとおりである。遮蔽境界から離れるに従い、赤、あるいは緑の濃度が濃くなっていることが確認できる。Wang らの手法 [2] では SSDF を事前計算処理で求めていたが、本研究では実行時に高速に求めることにより、動的なシーンに対応する。

4.3 環境照明の表現方法

環境照明は、拡散反射成分の計算に用いられる場合と、鏡面反射成分の計算に用いられる場合で、その表現方法が異なる。

拡散反射成分計算に対しては、Tsai らの手法 [7] を用い、環境マップ画像を複数の球面ガウス関数により近似する。本研究では 10 個の球面ガウス関数を用いて近似している。なお、環境マップ画像内の輝度の高い部分を中心に近似するため、このように少ない球面ガウス関数でも近似可能であり、Wang らは近似に使用する球面ガウス関数の数を 10 以下としている [2]。

鏡面反射成分の計算での環境照明は、異なるパラメータの球面ガウス関数の畳み込み演算によりフィルタリングされた複数の環境マップ画像として表現される。これは、レンダリング時に高速に環境照明と球面ガウス関数の積の積分を行うためである。事前計算処理において、異なる η の球面ガウス関数を用いて環境マップ画像に対して畳み込み演算を行い、結果をキューブマップのミップマップピラミッドに格納する。本研究では、 $\eta = 1, 10, 100, 1,000, 10,000$ の 5 つを用いて畳み込みを行った。

4.4 球近似による高速な SSDF の計算

本研究では、シーン内の遮蔽物となる剛体や変形物体を球の集合で近似することによって、物体の移動や回転、さらには、物体自体も変形するような動的なシーンで、リアルタイムに SSDF を計算する方法を提案する。

SSDF は最近傍の遮蔽境界からどの程度離れているかを角度により表現したものであり、空間的に滑らかに変化

することが知られているため [2]、本研究では頂点単位で SSDF を計算し、各ピクセルの SSDF の値は補間によって求める。

遮蔽物は n_s 個の球で近似するものとし、 j 番目の球 S_j は中心座標 \mathbf{c}_j および半径 r_j で定義される。頂点 \mathbf{v} において球 S_j を用いて計算される SSDF は、球面ガウス関数の中心軸 ζ と遮蔽境界ベクトル (図 2 (a) のベクトル \mathbf{t}) とのなす角である。球の中心座標 \mathbf{c}_j と半径 r_j を用いて、以下の式 (9) により SSDF の値 $V^d(\mathbf{v}, \zeta)$ は計算される (図 2 参照)。

$$V^d(\mathbf{v}, \zeta) = \cos^{-1} \left(\frac{(\mathbf{c}_j - \mathbf{v})}{\|\mathbf{c}_j - \mathbf{v}\|} \cdot \zeta \right) - \sin^{-1} \left(\frac{r_j}{\|\mathbf{c}_j - \mathbf{v}\|} \right) \tag{9}$$

ここで、式 (9) の第 1 項は頂点 \mathbf{v} から球の中心座標 \mathbf{c}_j へ方向ベクトルと球面ガウス関数の中心軸 ζ とのなす角、第 2 項は頂点 \mathbf{v} から球の中心座標 \mathbf{c}_j へ方向ベクトルと球の接線方向とのなす角である。中心軸 ζ が球と交差する場合、つまり、SSDF が負値である場合でも式 (9) は成り立つ。

遮蔽物を近似する n_s 個の球について式 (9) を計算し、最小値を求めることによって遮蔽物による頂点 \mathbf{v} の SSDF を計算することができる。Algorithm 1 に本研究での SSDF の計算アルゴリズムを示す。なお、Wang ら [2] はここで示した計算方法ではなく、各物体の各頂点から、法線方向を上向きとする半球面に向かってレイトレースを行うことにより SSDF を算出している。

Algorithm 1: SSDF($\mathbf{v}, \mathbf{n}, \zeta$)

Input: \mathbf{v} : 頂点, \mathbf{n} : 法線, ζ : 球面ガウス関数の中心軸, \mathbf{c}_j : 球座標, r_j : 球半径

Output: 頂点 \mathbf{v} における SSDF

```

ssdf ← π/2
for-each  $\mathbf{c}_j, r_j \in \text{spheres}$   $j \leftarrow 1 \sim n_s$ 
    if (  $\|\mathbf{c}_j - \mathbf{v}\| < r_j$  ) continue
     $\mathbf{c}'_j \leftarrow \mathbf{c}_j + \mathbf{n} \times r_j$ 
    if (  $\mathbf{n} \cdot ((\mathbf{c}'_j - \mathbf{v}) / \|\mathbf{c}'_j - \mathbf{v}\|) < 0$  ) continue
    value ←  $\cos^{-1}(((\mathbf{c}_j - \mathbf{v}) / \|\mathbf{c}_j - \mathbf{v}\|) \cdot \zeta) - \sin^{-1}(r_j / \|\mathbf{c}_j - \mathbf{v}\|)$ 
    if ( value < ssdf ) ssdf ← value
end for-each
return ssdf

```

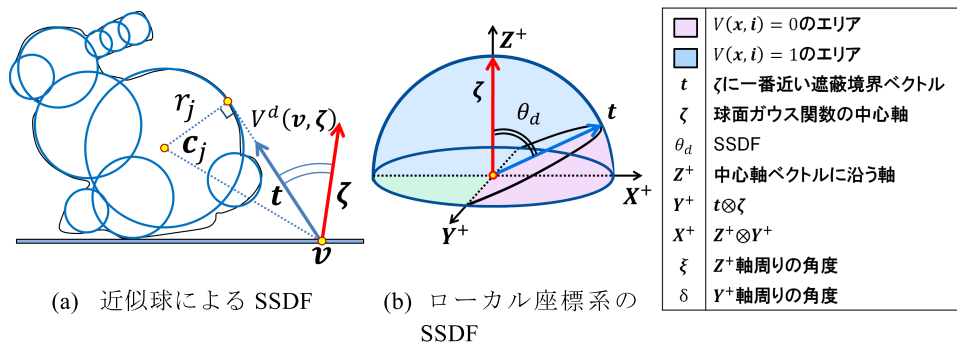


図 2 球近似を用いた SSDF の計算

Fig. 2 Calculation of SSDF using Sphere Set Approximation.

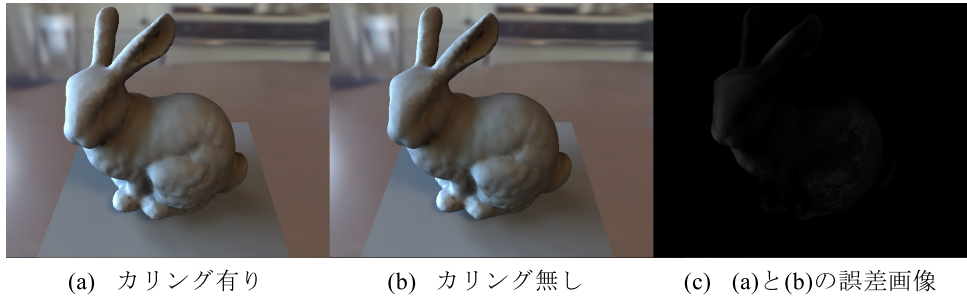


図 3 接平面以下の近似球のカリング処理をした場合としない場合との比較
Fig. 3 Comparison of images rendered with culling and without culling.

本研究では、遮蔽物を近似する球の数に比例した計算コストを要するため、球の数が多いシーンでは速度低下が予想される。速度低下への対処として、各頂点において頂点座標と法線により定義される接平面以下あるいは交差している球をカリングする。

図 3 はこの処理を行った場合 (図 3(a)) と行わなかった場合 (図 3(b)) の画像である。図 3(c) は図 3(a), (b) 2つの画像の誤差を可視化したものである。カリング処理を行った場合、図 3 に示すように誤差は視覚的にほぼ識別できないが、16%ほどの高速化が達成されている。また、頂点単位の SSDF の計算は並列計算可能であるため、GPU による並列計算を行った。これにより、Wang らの手法 [2] では 20~40 分程度必要としていた SSDF の計算時間をリアルタイムで実行することが可能となった。

4.5 輝度計算

提案法での直接光の輝度計算について説明する。以下の説明では、照明 $L(\mathbf{i})$ を $L(\mathbf{i}) \approx \sum_{k=1}^K G(\mathbf{i}, \boldsymbol{\zeta}_k, \eta_k, \mu_k) = \sum_{k=1}^K G_k(\mathbf{i})$, 鏡面反射成分の BRDF $\rho_s(\mathbf{i}, \mathbf{o})$ を $\rho_s(\mathbf{i}, \mathbf{o}) \approx \sum_{m=1}^M G(\mathbf{i}, \boldsymbol{\zeta}_m, \eta_m, \mu_m) = \sum_{m=1}^M G_m(\mathbf{i})$ として球面ガウス関数の線形和で表す。また、余弦項 $\cos \theta$ も $\cos \theta \approx G(\mathbf{i}, \mathbf{n}, \eta_c, \mu_c)$ として球面ガウス関数 1 つで表現する [2] ($\eta_c = 2.133$, $\mu_c = 1.170$)。ここで K , M はそれぞれ照明および BRDF を近似する球面ガウス関数の項数とする。

4.5.1 拡散反射成分

$L(\mathbf{i})$ および余弦項を近似する球面ガウス関数を式 (4) に代入すると、以下の式 (10) になり、これを計算することにより、位置 \mathbf{x} における拡散反射成分の輝度 $R_d(\mathbf{x})$ を求めることができる。

$$R_d(\mathbf{x}) = \sum_{k=1}^K \int_{\Omega} G_k(\mathbf{i}) G(\mathbf{i}, \mathbf{n}, \eta_c, \mu_c) \cdot V(\mathbf{x}, \mathbf{i}) d\mathbf{i} \quad (10)$$

まず照明項 $G_k(\mathbf{i})$ と余弦項 $G(\mathbf{i}, \mathbf{n}, \eta_c, \mu_c)$ の球面ガウス関数どうしの積を計算する。任意の 2 つの球面ガウス関数を $G_j(\mathbf{i}) = G(\mathbf{i}, \boldsymbol{\zeta}_j, \eta_j, \mu_j)$ $j = 1, 2$ と定義すると、球面ガウス関数どうしの積は、以下のように球面ガウス関数として計算される。

$$G_1(\mathbf{i}) G_2(\mathbf{i}) = G\left(\mathbf{i}, \frac{\boldsymbol{\zeta}_m}{\|\boldsymbol{\zeta}_m\|}, \eta_m \|\boldsymbol{\zeta}_m\|, \mu_1 \mu_2 e^{\eta_m (\|\boldsymbol{\zeta}_m\| - 1)}\right) \quad (11)$$

ここで $\boldsymbol{\zeta}_m = (\eta_1 \boldsymbol{\zeta}_1 + \eta_2 \boldsymbol{\zeta}_2) / (\eta_1 + \eta_2)$, $\eta_m = \eta_1 + \eta_2$ である。余弦項 $G(\mathbf{i}, \mathbf{n}, \eta_c, \mu_c)$ と照明項 $G_k(\mathbf{i})$ との積も球面ガウス関数として表現され、その球面ガウス関数を $G_k^{Lc}(\mathbf{i})$ とおく。

次に球面ガウス関数 $G_k^{Lc}(\mathbf{i})$ と可視関数 $V(\mathbf{x}, \mathbf{i})$ との積の積分を計算する。任意の球面ガウス関数 $G(\mathbf{i}, \boldsymbol{\zeta}, \eta, \mu)$ と可視関数 $V(\mathbf{x}, \mathbf{i})$ の積の積分は、球面ガウス関数の中心軸 $\boldsymbol{\zeta}$ を SSDF $V^d(\mathbf{x}, \mathbf{i})$ に代入して得られる角度 $\theta_d = V^d(\mathbf{x}, \boldsymbol{\zeta})$ を用いて以下の式 (12) で計算される。

$$\begin{aligned} & \int_{\Omega} G(\mathbf{i}, \boldsymbol{\zeta}, \eta, \mu) \cdot V(\mathbf{x}, \mathbf{i}) d\mathbf{i} \\ & \approx \mu \int_{\delta_0}^{\pi} \int_0^{\pi} G(\mathbf{i}, \mathbf{z}^+, \eta, 1) \sin \xi d\xi d\delta \\ & = \mu f_h(\theta_d, \eta) \end{aligned} \quad (12)$$

ここで、球面ガウス関数の中心軸 $\boldsymbol{\zeta}$ を \mathbf{z}^+ , 遮蔽境界ベクトル \mathbf{t} と中心軸 $\boldsymbol{\zeta}$ の外積を \mathbf{y}^+ としたローカル座標系内において、 $\xi \in [0, \pi]$ は \mathbf{i} と \mathbf{y}^+ とのなす角であり、 $\delta \in [0, \pi]$ は \mathbf{y}^+ 周りの角度と定義される。また、 $\delta_0 = (\pi/2) - \theta_d$ である (図 2(b) 参照)。式 (12) の積分式は解析的に解くことができないため最小二乗法を用いて近似して、 f_h とおくことにする。 f_h は以下のシグモイド関数で計算される。

$$\begin{aligned} f_h(\theta_d, \eta) & \approx \frac{1}{1 + e^{(k(\eta)\theta_d)}} \frac{2\pi}{\eta} (1 - e^{-\eta}) \quad (13) \\ k(\eta) & \approx 0.0079 \times \left(\frac{\eta}{10}\right)^2 - 0.9088 \times \left(\frac{\eta}{10}\right) + 0.79119 \quad (14) \end{aligned}$$

以上の計算により、拡散反射成分の輝度 $R_d(\mathbf{x})$ は以下の式 (15) により計算される。

$$R_d(\mathbf{x}) = \sum_{k=1}^K \mu_k^{Lc} f_h(\theta_{d_k}, \eta_k^{Lc}) \quad (15)$$

4.5.2 鏡面反射成分

環境照明下の鏡面反射成分の輝度は、以下の式 (16) により求められる。

$$R_s(\mathbf{x}, \mathbf{o}) = \sum_{m=1}^M \int_{\Omega} G_m(\mathbf{i}) G(\mathbf{i}, \mathbf{n}, \eta_c, \mu_c) V(\mathbf{x}, \mathbf{i}) \cdot L(\mathbf{i}) d\mathbf{i} \quad (16)$$

式 (16) の計算では、まず余弦項 $G(\mathbf{i}, \mathbf{n}, \eta_c, \mu_c)$ と BRDF $G_m(\mathbf{i})$ の球面ガウス関数どうしの積を式 (11) より求める。ここで、余弦項 $G(\mathbf{i}, \mathbf{n}, \eta_c, \mu_c)$ と BRDF $G_m(\mathbf{i})$ との積の結果である球面ガウス関数を $G_m^{pc}(\mathbf{i})$ とおく。次に球面ガウス関数 $G_m^{pc}(\mathbf{i})$ と可視関数 $V(\mathbf{x}, \mathbf{i})$ との積を計算する。任意の球面ガウス関数 $G(\mathbf{i}, \zeta, \eta, \mu)$ と可視関数 $V(\mathbf{x}, \mathbf{i})$ の積は、角度 $\theta_d = V^d(\mathbf{x}, \zeta)$ を用いて以下の式 (17) で近似される [2].

$$\begin{aligned} G(\mathbf{i}, \zeta, \eta, \mu) V(\mathbf{x}, \mathbf{i}) &\approx \frac{\int_{\Omega} G(\mathbf{i}) V(\mathbf{x}, \mathbf{i}) d\mathbf{i}}{\int_{\Omega} G(\mathbf{i}) d\mathbf{i}} G(\mathbf{i}, \zeta, \eta, \mu) \\ &\approx \frac{f_h(\theta_d, \eta)}{f_h(\frac{\pi}{2}, \eta)} G(\mathbf{i}, \zeta, \eta, \mu) \quad (17) \\ &= G\left(\mathbf{i}, \zeta, \eta, \frac{f_h(\theta_d, \eta)}{f_h(\frac{\pi}{2}, \eta)} \mu\right) \end{aligned}$$

式中の $f_h(\pi/2, \eta)$ は以下のとおりである。

$$f_h\left(\frac{\pi}{2}, \eta\right) = \int_{\Omega} G(\mathbf{i}, \mathbf{z}^+, \eta, 1) d\mathbf{i} = \frac{2\pi}{\eta} (1 - e^{-\eta}) \quad (18)$$

ここで、演算結果の球面ガウス関数を $G_m^{pcV}(\mathbf{i})$ とおく。最後に球面ガウス関数 $G_m^{pcV}(\mathbf{i})$ と環境照明 $L(\mathbf{i})$ との積の積分を計算する。任意の球面ガウス関数 $G(\mathbf{i}, \zeta, \eta, \mu)$ を用いた環境照明 $L(\mathbf{i})$ との積の積分は以下のように与えられる。

$$\begin{aligned} \int_{\Omega} G(\mathbf{i}, \zeta, \eta, \mu) \cdot L(\mathbf{i}) d\mathbf{i} &= \mu \int_{\Omega} G(\mathbf{i}, \zeta, \eta, 1) L(\mathbf{i}) d\mathbf{i} \\ &= \mu \Gamma(\zeta, \eta) \quad (19) \end{aligned}$$

ここで、関数 $\Gamma(\zeta, \eta)$ は、事前に畳み込み演算によりフィルタリングされた環境マップ画像のミップマップピラミッドから計算される。ミップマップ内の各階層は鋭さ η により選択され、画像内のピクセルは中心軸 ζ を用いてアクセスされる。よって、鏡面反射成分計算における照明と任意の球面ガウス関数との積の積分は、ミップマップピラミッドへのテクスチャフェッチ処理に置き換えることができる。また、 η により離散化された関数 $\Gamma(\zeta, \eta)$ は、テクスチャフェッチ処理の際、バイリニア補間により補間される。その結果、レンダリング処理では、畳み込み演算処理に用いた η の値の範囲 (適用例では $\eta = 1, 10, 100, 1,000, 10,000$) で任意の η を引数にとることができる。この η をもとにテクスチャの LOD (Level of Detail) 機能を用いて補間処理を実現する。以上の計算により、鏡面反射成分の輝度 $R_s(\mathbf{x}, \mathbf{o})$ は以下の式 (20) により計算される。

$$R_s(\mathbf{x}, \mathbf{o}) = \sum_{m=1}^M \mu_m^{pcV} \Gamma(\zeta_m^{pcV}, \eta_m^{pcV}) \quad (20)$$

なお、各々の計算を見れば明らかのように、位置 \mathbf{x} における輝度を求めるためには、拡散反射成分、鏡面反射成分それぞれの球面ガウス関数の中心軸 ζ の SSDF の値 $V^d(\mathbf{x}, \zeta)$ を求める必要がある。

4.6 レンダリング

レンダリングは、バーテックスシェーダで SSDF を頂点単位で並列に計算し、フラグメントシェーダにおいて、補間された SSDF と球面ガウス関数を用いてピクセル単位に直接光の輝度を計算する。

4.6.1 バーテックスシェーダ

バーテックスシェーダへの入力は、頂点、法線のほかに近似球データと BRDF パラメータや拡散反射成分の環境光源データ等がある。SSDF の計算はまず拡散反射成分、鏡面反射成分それぞれの球面ガウス関数の中心軸を計算する。拡散反射成分の中心軸は式 (10), (11) より余弦項と照明項の積で、鏡面反射成分の中心軸は式 (11), (16) より余弦項と BRDF 項の積で求めることができる。次に、求めた中心軸ごとに近似球データから Algorithm 1 を用いてそれぞれの中心軸に対する SSDF を計算する。なお、拡散反射成分の環境照明や鏡面反射成分の BRDF は、複数の球面ガウス関数の線形和で表現されるので、ここで必要になる中心軸や、それを用いて計算される SSDF もその個数分計算することになる。

4.6.2 フラグメントシェーダ

バーテックスシェーダで計算された SSDF はフラグメントシェーダに渡され、直接光の輝度計算に用いられる。拡散反射成分は式 (15) により、鏡面反射成分は式 (20) によりそれぞれ計算される。それぞれの計算結果を式 (3) により合算し、直接光の輝度を求めることによりレンダリングを行う。

5. 結果

図 4, 図 5 に本研究による結果例を示す。計算環境はいずれも CPU が Core i7 3.33 GHz, GPU は GeForce GTX480 を搭載した PC である。球の集合による遮蔽物の近似は文献 [12] の手法を用いた。

5.1 適用例

図 4 は剛体物の移動アニメーションシーンに提案法を適用した結果画像である。モデルは Stanford Bunny を使用した。シーン全体の頂点数は 14,468 頂点である。また、近似に用いた球の数 n_s は 512、環境照明を近似する球面ガウス関数の数 K は 10 である。また、図 4 のシーンの更新速度はいずれも平均 30 fps である。図 4(a) は、物体表面全体に拡散反射と鈍い鏡面反射 BRDF を設定し描画した結果である。BRDF は Ward モデル [13] を用いた。鏡面反射成分の反射率 k_s は 0.7、拡散反射成分の反射率 k_d は 0.3 である。BRDF を近似する球面ガウス関数の数 M は 1 である。図 4(b) は、図 4(a) と同様のシーンを用いて描画した。BRDF は Ward モデル [13] を赤の領域に、Blinn-Phong モデル [14] を緑の領域で用いた。なお、それ以外の部分の BRDF を拡散反射として描画した。提案法では、このよう

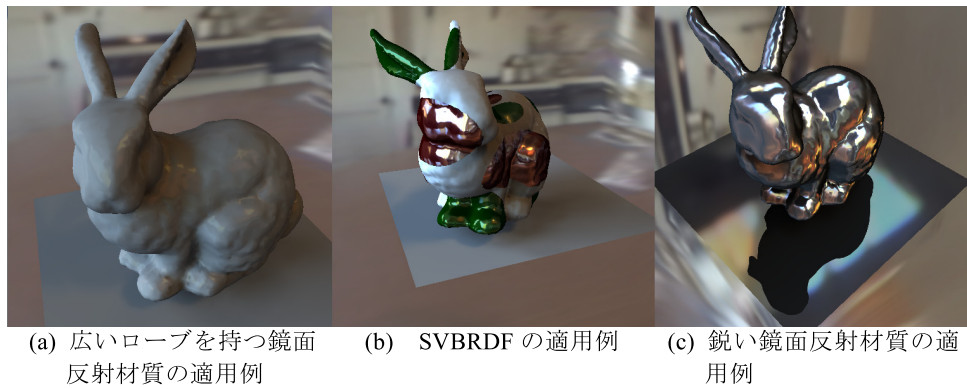


図 4 bunny モデルのレンダリング結果

Fig. 4 Rendering results of bunny model with moderate glossy BRDF, spatially-varying BRDF (SVBRDF) and highly glossy BRDF.

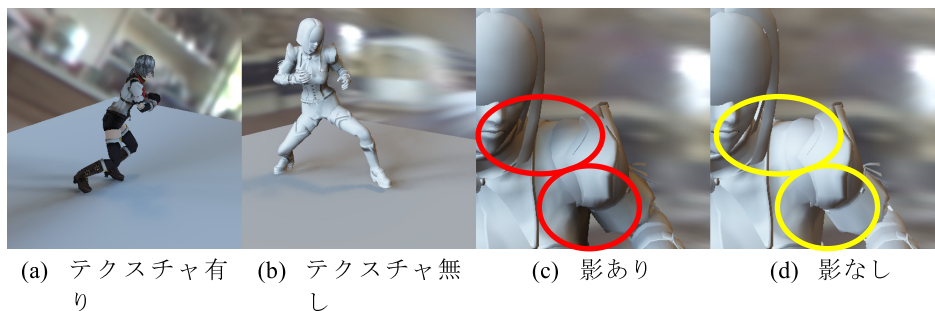


図 5 形状変形するキャラクターのレンダリング結果

Fig. 5 Rendering results of deformable character.

なピクセル単位で異なる BRDF を適用することも可能である。BRDF を近似する球面ガウス関数の数 M はどちらも 1 である。図 4(c) は、図 4(a) と同様のシーンを用いて、物体表面全体に鋭い鏡面反射 BRDF を設定し描画した結果である。BRDF は Ward モデル [13] を用いた。環境照明からの光が、床面やウサギの表面において、鏡のように反射していることが確認できる。なお、図 4(c) は環境照明の光源として用いた環境マップを回転させることにより、環境光の入射方向を動的に変えてレンダリングしている。なお、このような照明の回転や変更もリアルタイムで実行可能である。

図 5 は、キャラクターアニメーションへ提案法を適用した例である。キャラクターオブジェクトは変形可能な物体であり、提案法は剛体だけでなく変形物体へも適用可能である。なお、提案法は複雑な物体上にも適用可能である。図 5(c), (d) はそれぞれ、図 5(c) は影を考慮して描画し、図 5(d) は影を考慮しないで描画したものである。図 5(c) の赤丸で囲まれた部分にキャラクターの肩や脇に影が SSDF を考慮した結果として描画されている。SSDF を考慮しないで描画した図 5(d) では肩や脇に影が図 5(c) と同じ場所（黄色の丸で囲まれた部分）に描画されていないことが確認できる。シーンの全頂点数は 21,918 で、球の数 n_s は 105、環境照明を近似する球面ガウス関数の数 K は 10 である。レンダリング速度は平均 45 fps であり、リアルタイ

ムレンダリングが実現できている。

5.2 提案法とレイトレーシング法との比較

図 6(a) は提案法によるレンダリング結果、図 6(b) はレイトレーシング法によりレンダリングした参照画像である。それぞれ上段は拡散反射成分のみ、下段は鏡面反射成分のみでレンダリングした結果である。また、図 6(c) は、提案法とレイトレーシング法により生成された図 6(a), (b) の画像の誤差を可視化したものである。各画素の誤差は、RGB 値の差分の二乗和の平均をとったものである。図 6(c) 上段は図 6(a), (b) 上段の拡散反射成分の誤差を、図 6(c) 下段は図 6(a), (b) 下段の鏡面反射成分の誤差をそれぞれ表している。誤差が大きくなるに従い誤差画像の画素は明るくなる。図 6(b) に比べて、図 6(a) は影の領域が大きく、また、薄くなっているが、視覚的に妥当なライティング結果をリアルタイムに計算できている（図 6(b) はレイトレーシングに 135 秒要した）。影が大きくなっている原因は、球が計算点と離れている場合、SSDF を用いると遮蔽領域が大きく見積もられてしまうことが考えられる。そこで、球と計算点との位置関係に応じて補正を行う方法を検討している。

5.3 近似球数に関する考察

図 7 は近似球の数を変えて描画した画像（図 7 上段が

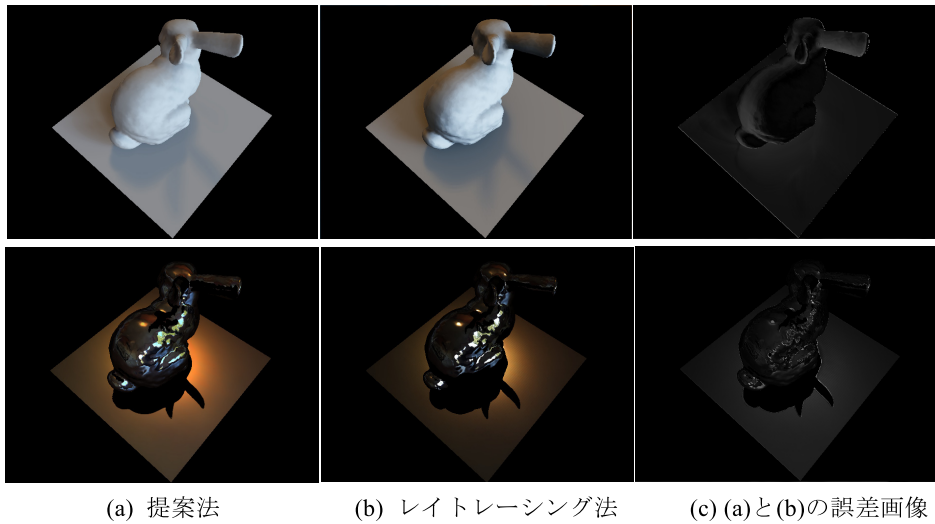


図 6 提案法とレイトレーシング法との画像比較

Fig. 6 Comparison of images rendered by using our method and ray tracing method.

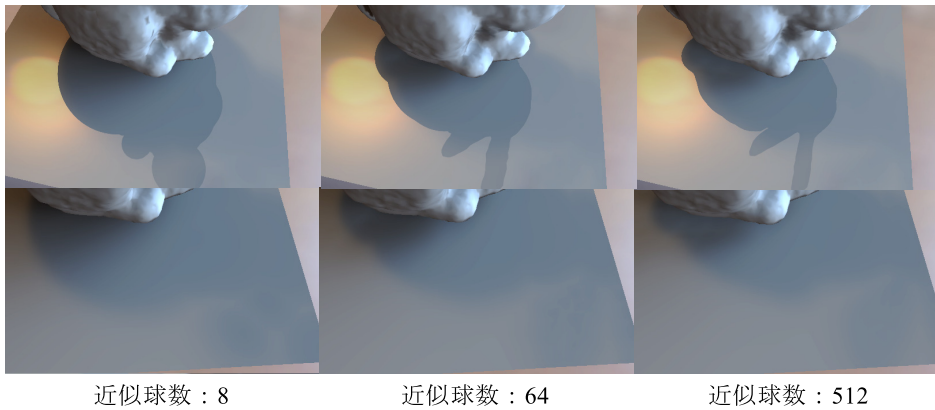


図 7 近似球数による影の比較画像

Fig. 7 Comparison of shadows with different numbers of spheres.

鏡面反射成分、図 7 下段が拡散反射成分)である。どちらも、近似球数が 8 個の場合 (左端) は 512 個の場合 (右端) と比較して全体的に影が大きくなり、細部、特に耳や尾の部分で詳細な形状が失われていることが分かる。さらに球形状が影にはっきりと現れている。近似球数が 64 個の場合 (中央)、拡散反射では、512 個の場合と比較して差異は顕著に現れてはいないが、やはり影領域が大きくなり細部が失われるというところは 8 個の場合と同様である。しかし、鏡面反射の場合、64 個でも 512 個と比べて差異がはっきりと識別できる。よって、近似球数を減らすことによる処理負荷の軽減を考える場合、影境界がぼやけている拡散反射成分計算には有効であるが、影境界がはっきりと現れる鏡面反射成分には適用しにくいと考えられる。

5.4 球近似手法について

本研究では、遮蔽情報を高速に取得するため、シーン内の遮蔽物となる物体を複数の球で近似する。物体の球近似手法には文献 [12] の手法を用いた。なお図 4, 図 5 に示した本研究の適用例のうち、図 4 で使用した頂点数 10,243

の Stanford Bunny を球近似した際の計算時間は CPU に Core i7 3.33 GHz を使用した場合、43 分 54 秒である。また、図 5 のキャラクタモデルに対しては、モデリングツールを使用し手動で近似球を設定した。

物体の変形をとともうアニメーションでは、剛体物体のアニメーションとは異なり、近似球の配置を物体の変形に合わせて動的に変更しなければならない。これを実現する手法として、ここでは 2 つを取り上げて説明する。

まず、アニメーションのフレームごとの変形形状すべてを事前に球近似するという方法がある。この手法の利点として、各変形形状に最適な近似球データを得ることができるといことがあげられる。しかし、欠点として、変形が限定されてしまうことや、フレーム数が膨大になると、近似計算に必要な時間や、近似球のデータ容量が膨大になること等の問題がある。

2 つ目の方法として、物体の基本形状に対してのみ球近似を行い、アニメーションでの物体変形の際には、変形箇所に合わせて近似球を回転、移動させるという方法である。たとえば、キャラクタの腕がアニメーションにより動く場

合、肩部、上腕、下腕を構成する頂点が、それぞれ別の変換行列により回転、移動の座標変換を行う。これらの頂点変換行列に、それぞれの部分を近似する球を関連付けて座標変換することにより、変形物体の球近似を実現する。本手法の適用例である図5はこの方法を用いている。この手法の利点は近似球データが1つでよいので近似計算時間や容量が少ないことや、変形が限定されないこと等があげられる。欠点として、各変形形状に合わせて最適な球近似データを得ることはできないことがあげられる。たとえば、モデルの一部に対する拡大縮小による変形では良い結果が得られない場合があると考えられる。

なお、平面を構成する四線分と球面ガウス関数の中心軸方向の半直線との最短距離を求めることで、平面とのSSDFは計算可能であるため、適用例中の床に対して球近似は行っていない。

5.5 球近似の制限

本研究が球近似の対象とする物体は、その形状を少ない数の球でよく近似できる物体である。これは、近似球の数に処理速度が比例するためである。よって、極端に薄い物体、たとえば、布や木の葉等の物体、あるいは、極端に細い物体、たとえば、髪の毛等の物体は、球により近似することが難しいか、多くの球を使用しなければならず、本研究が提案した手法ではリアルタイムに処理することが難しい。

6. まとめ

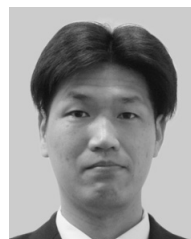
物体を球の集合で近似することにより、遮蔽情報であるSSDFを高速に計算する手法を提案した。従来手法では数十分要していたSSDFの計算を、GPUを利用することによってリアルタイムに計算することが可能となった。提案法により、全周波環境照明下における動的なシーンをリアルタイムにレンダリングすることができた。動的なシーンにおいてもリアルタイムに視点、照明およびBRDFが実行時に変更可能である。提案法は剛体物に限らず、変形物体にも適用可能である。また、鋭い鏡面反射やSVBRDFを表現することができる。

今後の課題として、球体以外のプリミティブ、たとえば楕円球や円柱、カプセル型等での近似によるSSDFの計算や、任意形状によるSSDFの計算手法の開発があげられる。また、さらなる高速化のため、近似球の階層構造の構築や、球と計算点との距離に基づいたLODを考慮すること等があげられる。

参考文献

- [1] Sloan, P.P., Kautz, J. and Snyder, J.: Precomputed Radiance Transfer for Real-time Rendering in Dynamic, Low-Frequency Lighting Environments, *ACM Trans. Graph.*, Vol.21, No.3, pp.527–536 (2002).
- [2] Wang, J., Ren, P., Gong, M., et al.: All-Frequency Ren-

- dering of Dynamic, Spatially-Varying Reflectance, *ACM Trans. Graph.*, Vol.28, No.5, pp.133:1–133:10 (2009).
- [3] Zhou, K., Hu, Y., Liu, S., et al.: Precomputed Shadow Fields for Dynamic Scenes, *ACM Trans. Graph.*, Vol.24, No.3, pp.1196–1201 (2005).
- [4] Ren, Z., Wang, R., Snyder, J., et al.: Real-time Soft Shadows in Dynamic Scenes using Spherical Harmonic Exponentiation, *ACM Trans. Graph.*, Vol.25, No.3, pp.977–986 (2006).
- [5] Ng, R., Ramamoorthi, R. and Hanrahan, P.: All-Frequency Shadows using Non-Linear Wavelet Lighting Approximation, *ACM Trans. Graph.*, Vol.22, No.3, pp.376–381 (2003).
- [6] Green, P., Kautz, J., Matusik, W., et al.: View-dependent Precomputed Light Transport using Nonlinear Gaussian Function Approximations, *Proc. Symposium on Interactive 3D Graphics and Games (I3D 2006)*, pp.7–14, ACM Press (2006).
- [7] Tsai, T.-Y. and Shih, C.-Z.: All-frequency Precomputed Radiance Transfer using Spherical Radial Basis Functions and Clustered Tensor Approximation, *ACM Trans. Graph.*, Vol.25, No.3, pp.967–976 (2006).
- [8] Xu, K., Jia, T.-Y., Fu, H., et al.: Spherical Piecewise Constant Basis Functions for All-Frequency Precomputed Radiance Transfer, *IEEE Trans. Visualization and Computer Graphics*, Vol.14, No.2, pp.454–467 (2008).
- [9] Ritschel, T., Grosch, T., Kim, H.M., et al.: Imperfect Shadow Maps for Efficient Computation of Indirect Illumination, *ACM Trans. Graph.*, Vol.27, No.5, p.129 (2008).
- [10] Krivánek, J. and Colbert, M.: Real-time Shading with Filtered Importance Sampling, *Computer Graphics Forum*, Vol.27, No.4, pp.1147–1154 (2008).
- [11] Annen, T., Dong, Z., Mertens, T., et al.: Real-time All-Frequency Shadows in Dynamic Scenes, *ACM Trans. Graph.*, Vol.27, No.3, pp.34:1–34:8 (2008).
- [12] Bradshaw, G. and O'Sullivan, C.: Adaptive Medial-axis Approximation for Sphere-tree Construction, *ACM Trans. Graph.*, Vol.23, No.1, pp.1–26 (2004).
- [13] Ward, J.G.: Measuring and Modeling Anisotropic Reflection, *Computer Graphics*, Vol.26, pp.265–272 (1992).
- [14] Blinn, J.F.: Models of Light Reflection for Computer Synthesized Pictures, *Computer Graphics*, Vol.11, No.2, pp.192–198 (1977).



古家 亙

2010年和歌山大学システム工学部飛び級。2012年同大学大学院システム工学研究科博士前期課程修了。現在、シリコンスタジオ株式会社に勤務。



岩崎 慶 (正会員)

1999年東京大学理学部卒業。2001年同大学大学院新領域創成科学研究科博士前期課程修了。2004年同大学大学院新領域創成科学研究科博士後期課程修了。同年和歌山大学システム工学部助手。2007年同講師。2009年同准教授。主としてコンピュータグラフィクスに関する研究に従事。科学博士。



土橋 宜典 (正会員)

1992年広島大学工学部第二類(電気系)卒業。1994年同大学大学院工学研究科システム工学専攻博士課程前期修了。1997年同専攻博士課程後期修了。同年広島市立大学情報科学部助手。2000年北海道大学大学院工学研究科助教授。主としてコンピュータグラフィクスに関して、照明シミュレーション、景観予測等の研究に従事。工学博士。



西田 友是 (正会員)

1971年広島大学工学部卒業。1973年同大学大学院工学研究科修了。同年マツダ(株)に入社。1979年福山大学工学部講師。1984年同助教授。1990年同教授。1998年東京大学理学部教授。1999年同大学大学院新領域創成科学研究科教授となり、現在に至る。コンピュータグラフィックスの研究に従事。工学博士。