

動的シーンにおけるソフトシャドウの高速計算

田村 尚希[†] ヘンリー・ジョハン[†] 西田 友是[†]

近年、グラフィックスハードウェアの進歩を背景として、写実的な画像をリアルタイムで生成する研究が盛んに行われている。そして、半影、直接光、間接光、集光効果をリアルタイムでレンダリングする手法として、前計算放射輝度伝達法がある。しかし、既存の手法では、前計算した状態から物体を移動・回転できないという点が最も大きな課題であった。そこで本稿では、既存の前計算放射輝度伝達法と比べて画質、計算速度を損なうことなく、リアルタイムで物体を移動・回転できる手法を提案する。提案法では、物体が移動・回転した際、他の物体による影が主に変化することに着目し、自分自身による影と他の物体による影に区別して扱い、両者をグラフィックスハードウェアを用いて高速に計算している。

Fast Calculation of Soft Shadows in Dynamic Scene

NAOKI TAMURA,[†] HENRY JOHAN,[†] and TOMOYUKI NISHITA[†]

Recently, due to the progress of graphics hardware, there have been many researches on real-time rendering of photorealistic images. In order to render photorealistic effects such as soft shadows, direct and indirect illuminations, and caustics, Pre-computed Radiance Transfer (PRT) methods were proposed. However, previous methods have a serious limitation, that is the objects in the scene cannot be moved or rotated from their states in the pre-computation. In this paper, we propose a method for real-time photorealistic rendering of dynamic scenes without reducing the quality and the speed compared to the previous PRT methods. We notice the fact that when objects are moved or rotated, changes occur mainly in the regions of shadows cast by other objects. Our method makes a distinction between self-shadow and shadows cast by other objects and computes these two kinds of shadows efficiently using the programmable graphics hardware.

1. はじめに

近年、グラフィックスハードウェア (GPU) の進歩を背景として、写実的な画像をリアルタイムで生成する研究が盛んに行われている。Sloan らが提案した前計算放射輝度伝達法¹⁷⁾ は、無限遠のドーム状の複雑な光源環境のもとで、半影、直接光、間接光、集光効果をリアルタイムでレンダリングできる手法として注目されている。現在では wavelet 変換を利用する手法¹³⁾¹⁴⁾、主成分分析を用いて前計算データを圧縮する手法¹⁸⁾ など、主に画質、計算速度を向上させる手法が数多く提案されている。

しかし、前計算時間、前計算結果データが膨大、前計算したシーン状態から物体を移動・回転できない、メッシュを高解像度にしなくてはならない、など一般

のアプリケーションで利用するためには課題が残されていた。特にその中で、物体を移動・回転でき、かつ高速に処理できる有効な手法はまだ提案されていない。そこで、本稿では、既存手法と比較して画質、計算速度を損なうことなく、リアルタイムで物体を移動・回転できる手法を提案する。

提案法では、物体が移動・回転した際、他の物体による影が主に変化することに着目し、自分自身による影と他の物体による影に区別して考え、両者を効率的に計算している。また、昨今著しく進化を続ける GPU で大半の処理を行えるように提案法を構築している。

提案法の特徴は以下の三点である。

- 動的なシーンへの対応
- GPU を用いた影の高速計算
- メッシュ解像度の緩和

本稿が想定するシーン条件は次の通りである。シーンは三角形メッシュで構成されており、拡散面を対象とする。また、扱う光源は環境マップによって表現された無限遠のドーム状の光源であり、これ以降環境マップ

[†] 東京大学大学院新領域創成科学研究科複雑理工学専攻
Department of Complexity Science and Engineering,
Graduate School of Frontier Sciences, The University
of Tokyo

ブ光源と呼ぶ。

2. 関連研究

本節では関連研究を述べる。提案法ではドーム状の光源が対象であり、光源が大きいためソフトシャドウが重要になる。またさらに、ドーム状の光源下においてリアルタイムでレンダリングする手法であるため前計算放射輝度伝達法が大きく関係する。そこで二種類の分野について関連研究を挙げる。

2.1 ソフトシャドウ

西田らはソフトシャドウの特徴を捉えレンダリングする手法¹⁵⁾を提案した。しかし、ソフトシャドウの生成には多くの計算量が必要であり、リアルタイムでレンダリングするのは難しかった。

近年、HeckbertらはGPUを用いてソフトシャドウをレンダリングする手法⁴⁾を提案した。また、Heidrichらは、GPUを用いた影の計算で多用されるシャドウマップ法²¹⁾をソフトシャドウに拡張した手法⁵⁾を提案した。さらに、SolerらはFFTによる畳み込み演算を用いてソフトシャドウを計算する手法²⁰⁾を提案した。また、MollerらやAssarssonらは、シャドウボリューム法³⁾を拡張して、GPUを利用してソフトシャドウを描画する手法¹⁾¹²⁾を提案した。しかしこれらの手法は環境マップ光源によるソフトシャドウは考えていない。

2.2 前計算放射輝度伝達法

Sloanらが提案した前計算放射輝度伝達法¹⁷⁾は、半影、直接光、間接光、集光効果をリアルタイムでレンダリングできる手法として注目された。その後Kautzらが、任意のBRDFに対応した手法⁷⁾や、光沢面のレンダリングを効率化させた手法⁹⁾を提案した。またNgらは、Sloanの手法では環境マップ光源、遮蔽の情報を低周波帯でしか近似できない欠点をwavelet変換を利用し克服する手法¹³⁾¹⁴⁾を提案した。さらにSloanらは、主成分分析を用いてデータを圧縮する手法¹⁸⁾や、BTFと併用する手法¹⁹⁾を提案した。これらの手法では、前計算した状態から物体を移動・回転できない。また、Jamesらは、限定的な空間でのみ物体を変化できる手法⁶⁾を、Meiらは、計算が頂点単位、CPUベースであるが、物体を移動・回転できる手法¹¹⁾を提案した。しかしこれらの手法は、基本的に計算が頂点単位であるため、輝度の変化が急激な箇所ではメッシュ解像度を上げなくてはならない。

3. 提案法

本節では、提案法のアルゴリズムを説明する。

3.1 提案法の基本概念

物体を動かさないのは、予め求めた遮蔽の関係が変化してしまうからである。そこで、遮蔽情報を単一で扱うのではなく、シーンをいくつかのグループに分け遮蔽を区別して考える。グループ内は変形しないと仮定し、グループ単位で移動・回転する。これまで、自分自身・他の物体と曖昧に利用していた遮蔽を以下のように定義して区別する。

- 自己遮蔽：同じグループ内の物体による遮蔽
- 他者遮蔽：他のグループの物体による遮蔽

提案法では、まず自己遮蔽のみを考慮して輝度計算をした後、グループ間の距離・方向を利用して、あるグループが他のグループに及ぼす影を計算する。

ここで、グループ内は変形しないため、各グループが移動・回転しても自己遮蔽は変化しない。そのため輝度の変化が大きいと予想される部分を予め特定でき、局所的にメッシュ解像度を上げることができる。しかし他者遮蔽はグループ同士の相対的な位置によって決まるため、自己遮蔽のように予め局所的にメッシュ解像度を上げることはできない。そこで、前者を頂点単位で処理し、後者を可視できる部分のみピクセル単位で処理することを提案する。これにより、メッシュ解像度をシーン全体で均一に上げる必要がなくなり、効率良く計算できる。

3.2 自己遮蔽のみを考慮した輝度計算

図1(a)のようにグループ毎に自己遮蔽のみを考慮した輝度計算を行う。この過程は、Sloanらの手法¹⁷⁾と同様に球面調和関数¹⁰⁾を用いて頂点単位で計算する。ここで、 i 番目の頂点 x_i の自己遮蔽のみを考慮した拡散面の放射輝度 L_s は以下の式で与えられる。

$$L_s(x_i) = \frac{\rho_d}{\pi} \int_{\Omega} L_{ENV}(\vec{\omega}) v_s(x_i, \vec{\omega}) (\vec{\omega} \cdot \vec{n}) d\vec{\omega} \quad (1)$$

ここで L_{ENV} は環境マップ光源からの入射輝度、 v_s は二値の自己遮蔽情報、 \vec{n} は頂点 x_i の法線ベクトル、 ρ_d は拡散反射率、 Ω は単位球、 $\vec{\omega}$ は単位球面への方向ベクトルを表す。次に、光の輸送関数 T を以下のように定義する。

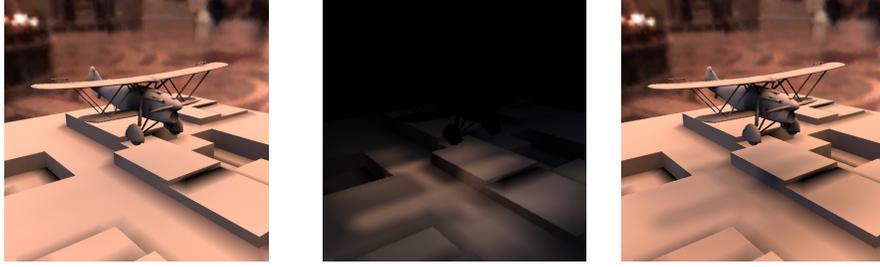
$$T(x, \vec{\omega}) = \frac{\rho_d}{\pi} v_s(x, \vec{\omega}) (\vec{\omega} \cdot \vec{n}) \quad (2)$$

これを(1)式に代入する。

$$L_s(x_i) = \int_{\Omega} L_{ENV}(\vec{\omega}) T(x_i, \vec{\omega}) d\vec{\omega} \quad (3)$$

ここで、 L_{ENV}, T を球面調和関数を用いて展開し各基底に対する係数を次式のように計算する。

$$L_l^m = \int_{\Omega} L_{ENV}(\vec{\omega}) y_l^m(\vec{\omega}) d\vec{\omega} \quad (4)$$



(a) 自己遮蔽のみを考慮した輝度計算 (b) 他者遮蔽による輝度計算 (影) (c) 最終的な輝度

図 1 提案法の計算過程を示した図

$$T_l^m(x_i) = \int_{\Omega} T(x_i, \vec{\omega}) y_l^m(\vec{\omega}) d\vec{\omega} \quad (5)$$

ここで y は球面調和関数の基底、 l は正の整数、 m は $|m| \leq l$ を満たす整数である。また、球面調和関数の正規直交性から、(3) 式は

$$L_s(x_i) = \sum_{l=0}^{d-1} \sum_{m=-l}^{m=l} L_l^m T_l^m(x_i) \quad (6)$$

と変形できる。ここで d は任意の正の整数。この計算は、物体もしくは光源が回転し、物体と光源の相対的な関係が変化した場合のみ再計算が必要である。

3.3 他者遮蔽を考慮した輝度計算

自己遮蔽のみを考慮した輝度を計算した後、他者遮蔽を考慮した輝度を計算する。図 1(b) のように本来他者遮蔽によって遮られる輝度を求めた後、図 1(c) のように自己遮蔽のみを考慮した輝度から減算し、全体のレンダリングを完了する。

$$L(p) = L_s(p) - L_o(p) \quad (7)$$

ここで p は任意のピクセル位置、 L はシーンの最終的な輝度、 L_o は他者遮蔽で遮られる輝度である。

3.3.1 環境マップ光源の分解

環境マップ光源を N 個の領域に分解し $L_o(p)$ を離散的に計算する。

$$L_o(p) \approx \frac{\rho^d}{\pi} \sum_{i=1}^N B_i(\vec{\omega}_i) (1 - v_o(p, \vec{\omega}_i)) (\vec{\omega}_i \cdot \vec{n}) \quad (8)$$

$$B_i(\vec{\omega}_i) = \int_{U_i} L_{ENV}(\vec{\omega}) d\vec{\omega} \quad (9)$$

ここで、 B は放射発散度、 v_o は二値の他者遮蔽情報、 $U_i (i = 1, 2, \dots, N)$ は i 番目の領域を表す。

この過程で問題となるのは領域 U_i の分け方と方向 $\vec{\omega}_i$ の決め方である。提案法では、Kollig らの手法⁸⁾ を利用し、図 2(a) のように環境マップ光源をいくつかのポロノイ領域に分ける。各領域内の照度の総和を B_i に、重心方向を方向 $\vec{\omega}_i$ としている。つまり、環境マップ光源を平行光線の集まりで近似する。

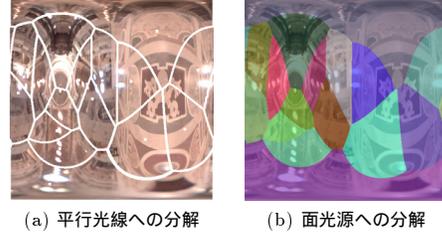


図 2 環境マップ光源の分解

3.3.2 畳み込み演算によるソフトシャドウ

平行光線の集まりでソフトシャドウを表現するには、図 3 から理解できるように多くの光源数を必要とする。しかし、光線数の増加は直接的に計算負荷の増加を意味しており、十分な速度が期待できない。



(a)32 サンプル (b)64 サンプル (c)128 サンプル

図 3 平行光線の集まりによるソフトシャドウの描画

そこで、少ない数の光源でソフトシャドウをレンダリングする手法を提案する。各領域を前節で示した平行光線ではなく、図 2(b) のように面光源 W の集合として考える。

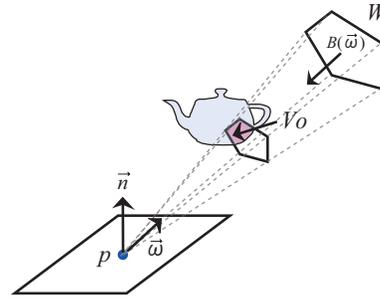


図 4 他者遮蔽による遮蔽割合 V_o の概念

さらに、図 4 のように他者遮蔽によって各光源を遮る割合 V_o を求めることで、少ない光源数でソフトシャ

ドウを表現できる。

$$L_o(p) \approx \frac{\rho_d}{\pi} \sum_{i=1}^q B_i(\vec{\omega}_i) V_o(p, \vec{\omega}_i) (\vec{\omega}_i \cdot \vec{n}) \quad (10)$$

ここで q は $q \ll N$ を満たす整数、 V_o は実数で表された他者遮蔽による割合である。

光源 W は無限遠に存在し、そのままでは遮蔽割合 V_o は単純には求められない。そこで図 5 のようにグループの境界球の接平面を考え、光源 W を投影し有限の大きさの面光源 W_{proj} とする。同様に他者遮蔽も接平面に投影し、面光源 W_{proj} と重なる割合を求める。

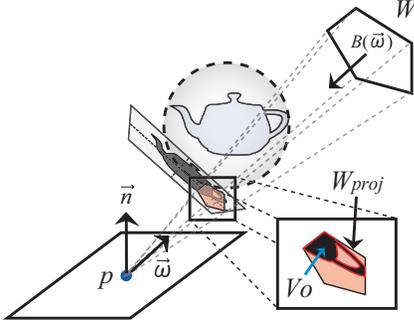


図 5 無限遠光源の接平面への投影と遮蔽割合 V_o の求め方

遮蔽割合 V_o を全ピクセルで正確に計算するには負荷が高すぎるため、グループ間の計算で近似する。グループの中心座標から図 5 のように V_o を求める。さらに図 6 のように接平面内で W_{proj} を平行移動させて他の位置の V_o を近似する。この計算は、Soler らの手法²⁰⁾ でも示されているように畳み込み演算に相当する。各グループで、他のグループと各面光源の一組毎に畳み込みを用いて V_o を計算し、テクスチャに保存する。さらに、これを投影して影を求める。

$$V_o(p, \vec{\omega}_i) \approx W_{proj_i} \otimes (1 - v_o(p, \vec{\omega}_i)) \quad (11)$$

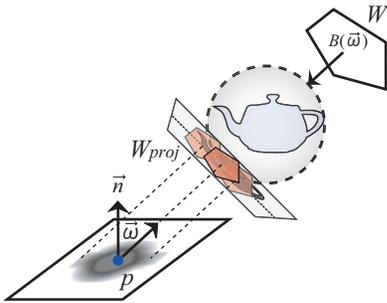


図 6 提案法における遮蔽割合 V_o の求め方

また、複雑な形をした畳み込みのカーネル関数では、GPU において高速に処理できないため、提案法ではボックスフィルタに単純化している。ここでボックス

フィルタの幅 s は以下のように計算する。

$$Area(W_{proj_i}) \approx \Omega_{W_i} * R_i * R_i \quad (12)$$

$$s(\vec{\omega}_i) = \sqrt{Area(W_{proj_i})} \quad (13)$$

$Area$ は面積、 Ω_{W_i} は光源 W_i の立体角、 R はグループの中心座標から接平面までの距離を表す。

3.3.3 影計算の問題点の克服

ある位置からある方向 $\vec{\omega}_i$ を見たとき、自己遮蔽と他者遮蔽が同時に存在する場合がある。この場合に提案法を直接適用すると二回遮蔽され、正しい輝度よりも小さくなってしまふ。そこで (14) 式のように、各ピクセル p での自己遮蔽の値を前節で求めた遮蔽割合に乘算し、自己遮蔽が存在する場合は他者遮蔽が影響しないようにする。

$$L_o(p) \approx \frac{\rho_d}{\pi} \sum_{i=1}^q B_i(\vec{\omega}_i) v_s(p, \vec{\omega}_i) V_o(p, \vec{\omega}_i) (\vec{\omega}_i \cdot \vec{n}) \quad (14)$$

ここで (2) 式を用いて (14) 式を変形する。

$$L_o(p) \approx \sum_{i=1}^q B_i(\vec{\omega}_i) T(p, \vec{\omega}_i) V_o(p, \vec{\omega}_i) \quad (15)$$

各ピクセルでの T の値は、(5) 式で求められる各頂点の球面調和関数の係数を、対象とするピクセルを占める三角形内で補間して算出する。具体的な算出方法は 4.3 節で述べる。

4. 実装

本節では、実装における要点を述べる。

4.1 自己遮蔽のみを考慮した輝度計算

3.2 節で示した自己遮蔽のみを考慮した輝度を GPU を用いて高速に計算する。図 7 のように各頂点に対してテクスチャの 1 ピクセルを割り当て、各頂点を対応する位置に点として描画する。このとき、(5) 式で求められた各頂点をもつ係数を頂点属性値として与える。さらに、(4) 式で求められた環境マップ光源の係数を予めプログラブルシェーダに与えておき、シェーダ内で (6) 式のように積和計算を行う。計算されたテクスチャは、シーン描画時に、頂点シェーダでテクスチャ参照し利用する。また、3.2 節で述べたように再計算の必要がない限り同一テクスチャを参照し続ける。

4.2 畳み込み演算

GPU を用いた畳み込み演算は、基本的に、ガウシアンフィルタを用いたグレアエフェクト²⁾ と同一である。二次元の畳み込みを 1 回で処理するのではなく、一次元の畳み込み 2 回で処理している。

4.3 影計算の問題点の克服

3.3.3 節で述べたように、まず近傍の 3 頂点において任意方向の T の値を、(5) 式で求めた係数から再構

築する。そして3頂点で求められた T の値を三角形内で補間する。補間処理は、GPUで行えるため、予め各頂点での T の値を再構築すればよい。そして、プリミティブ描画時に各頂点の T の値を頂点属性値として与え、プログラマブルシェーダで補間後の値を利用する。各頂点で値を再構築する時、球面調和関数の値を予めテーブルに保存し評価の高速化を図ることも可能であるが、この場合再構築される値の精度が、作成するテーブルの解像度に依存してしまう。よって、提案法では、デカルト座標系の実数球面調和関数¹⁰⁾を利用し離散化させずに評価を行っている。

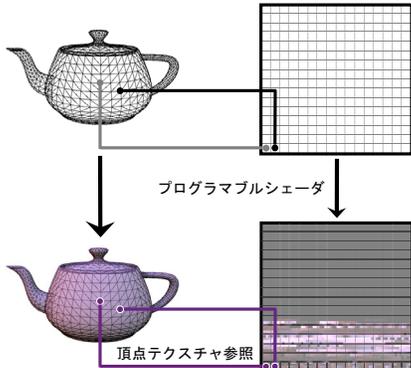


図7 GPUを利用した自己遮蔽による輝度計算方法

5. 結果

提案法による結果と既存手法との比較について述べる。提案法における環境マップ光源の分割数は32、球面調和関数の次数 l は、提案法、Sloanらの手法¹⁷⁾ともに3である。また、図8はレイトレーシング法で作成した参考画像である。各ピクセルのサンプル数は16384サンプルである。

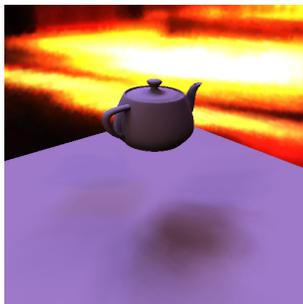


図8 レイトレーシング法による参考画像

- 既存手法との画質の比較

図9、図10から、提案法では図8のレイトレーシング法による画像と視覚的に近い画像を生成できることが分かる。また、数値的に差分を取ってみてもほとんど差がなく、提案法は精度の高い手

法であると言える。同様の条件でSloanらの手法¹⁷⁾では緩やかな影しか生成できず、差分値も大きい。

- メッシュ解像度

図9(c)、図10(c)を比較して、提案法ではメッシュの解像度に依存せずに精度の高い影を生成できていることが分かる。

- 実行速度

Sloanらの手法¹⁷⁾の実行速度は60fpsである。また、提案法でも実行速度は、物体を移動・回転して60fpsである。

- 前計算時間

前計算時間は、提案法、Sloanらの手法¹⁷⁾ともに、低解像度メッシュにおいて10秒、高解像度メッシュにおいて88秒である。

- 前計算データ量

前計算データ量は、提案法、Sloanらの手法¹⁷⁾ともに、低解像度メッシュにおいて151KB、高解像度メッシュにおいて861KBである。

計算時間の計測には、CPU: Pentium4 3.0GHz、メインメモリ: 1024MB、GPU: GeForce 6600GT (グラフィックスメモリ 128MB))を用いた。また、実装にはOpenGLとGLSL¹⁶⁾を用いた。

6. まとめ

本稿では、無限遠のドーム状光源下の動的シーンにおいてリアルタイムにレンダリングできる手法を提案した。提案したアルゴリズムは、自己遮蔽と他者遮蔽による影を区別して考え、両者を効率良く計算している。また、処理の大半をGPUのプログラマブルシェーダを用いて高速に処理できる。

今後の課題は、人間の三次元モデルのアニメーションのような複雑に形状が変化するアニメーションへの対応である。

参考文献

- 1) Ulf Assarsson and Tomas Akenine-Moller. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Transactions on Graphics (TOG)*, 22(3):511-520, 2003.
- 2) ATI Research. Real-Time 3D Scene Post-processing. *Game Developer's Conference Europe 2003*, 2003.
- 3) Franklin C. Crow. Shadow algorithms for computer graphics. *Proc. of SIGGRAPH 77*, pages 242-248, 1977.
- 4) Paul S. Heckbert and Michael Herf. Simulating soft shadows with graphics hardware. *Technical report TR CMU-CS-97-104, Carnegie Mellon Uni-*

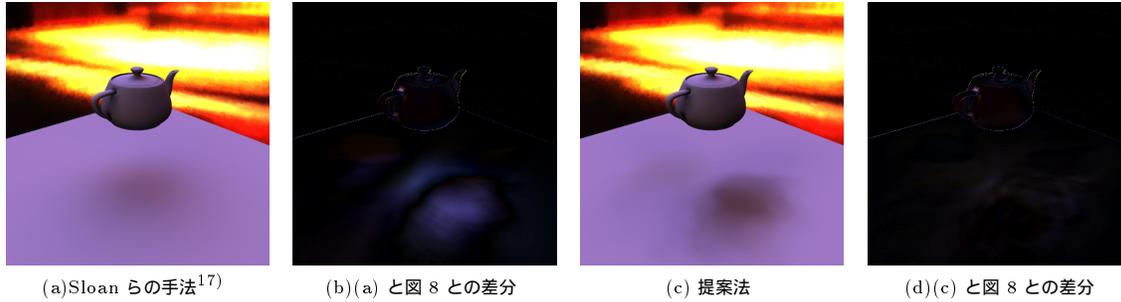


図 9 低解像度メッシュ(ティーポット 1,461 頂点、床 961 頂点)における比較例

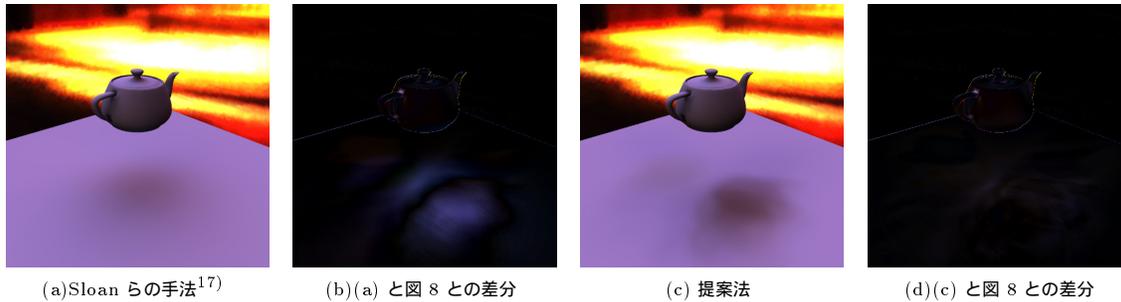


図 10 高解像度メッシュ(ティーポット 1,461 頂点、床 12,321 頂点)における比較例

- versity, January 1997, 1997.
- 5) Wolfgang Heidrich, Stefan Brabec, and Hans-Peter Seidel. Soft Shadow Maps for Linear Lights. *Proc. of 11th Eurographics Workshop on Rendering*, pages 269–280, 2000.
 - 6) Doug L. James and Kayvon Fatahalian. Precomputing interactive dynamic deformable scenes. *ACM Transactions on Graphics (TOG)*, 22(3):879–887, 2003.
 - 7) Jan Kautz, Peter-Pike Sloan, and John Snyder. Shading and shadows: Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. *Proc. of 13th Eurographics Workshop on Rendering*, pages 291–296, 2002.
 - 8) Thomas Kollig and Alexander Keller. Efficient illumination by high dynamic range images. *Proc. of 14th Eurographics Workshop on Rendering*, pages 45–50, 2003.
 - 9) Jaakko Lehtinen and Jan Kautz. Matrix radiance transfer. *Proc. of 2003 Symposium on Interactive 3D graphics*, pages 59–64, 2003.
 - 10) T. M. MacRobert. Spherical harmonics: An elementary treatise on harmonic functions, with applications. Dover Publications, 1948.
 - 11) Chunhui Mei, Jiaoying Shi and Fuli Wu. Rendering with spherical radiance transport maps. *Proc. of Eurographics 2004*, pages 281–290, 2004.
 - 12) Tomas Akenine-Moller and Ulf Assarsson. Shading and shadows: Approximate soft shadows on arbitrary surfaces using penumbra wedges. *Proc. of 13th Eurographics Workshop on Rendering*, pages 297–306, 2002.
 - 13) Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics (TOG)*, 22(3):376–381, 2003.
 - 14) Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics (TOG)*, 23(3):477–487, 2004.
 - 15) T. Nishita, I. Okamura, and E. Nakamae. Shading Models for Point and Linear Sources. *ACM Transactions on Graphics (TOG)*, 4(2):124–146, 1985.
 - 16) Randi J. Rost, John M. Kessenich, Barthold Lichtenbelt, and Marc Olano. OpenGL Shading Language. Addison Wesley, 2003.
 - 17) Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *Proc. of SIGGRAPH 2002*, pages 527–536, 2002.
 - 18) Peter-Pike Sloan, Jesse Hall, John Hart, and John Snyder. Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics (TOG)*, 22(3):382–391, 2003.
 - 19) Peter-Pike Sloan, Xinguo Liu, Heung-Yeung Shum, and John Snyder. Bi-scale radiance transfer. *ACM Transactions on Graphics (TOG)*, 22(3):370–375, 2003.
 - 20) Cyril Soler and Francois X. Sillion. Fast calculation of soft shadow textures using convolution. *Proc. of SIGGRAPH 98*, pages 321–332, 1998.
 - 21) Lance Williams. Casting curved shadows on curved surfaces. *Proc. of SIGGRAPH 78*, pages 270–274, 1978.