

# 視点とオブジェクトの位置関係を考慮した シャドウマップの動的生成法

岩尾 友秀<sup>†</sup> 神原 誠之<sup>†</sup> 横矢 直和<sup>†</sup>

<sup>†</sup> 奈良先端科学技術大学院大学 〒630-0192 奈良県生駒市高山町 8916-5

E-mail: † {tomo-i, kanbara, yokoya}@is.naist.jp

**あらまし** コンピュータグラフィックスの分野において影を描画するアルゴリズムは、現在まで多く研究されているが、そのひとつにシャドウマップ法がある。シャドウマップ法は、光源からオブジェクトまでの距離をあらかじめ計算し、シャドウマップと呼ばれるバッファに保存しておき、これを後に行う影付けに利用することで高速かつ正確な影付けを行うことが可能である。しかし、視点とオブジェクトの位置関係により、シーン中のオブジェクトに割り当てられるシャドウマップの解像度が不足し、影の境界にエイリアシングが発生するという問題点がある。そこで本論文では、視点とオブジェクトの位置関係に応じて動的にシャドウマップを生成することで、影のエイリアシングを軽減させる手法を提案する。また、従来のシャドウマップ法と、視点からオブジェクトまでの距離を考慮してシャドウマップを割り当てるパースペクティブシャドウマップ法に提案手法を適用して実験を行い、処理速度およびレンダリング結果について、考察を行う。

**キーワード** シャドウマップ法、動的シャドウマップ生成、パースペクティブシャドウマップ法、リアルタイムレンダリング

## Dynamic Shadow Map Generation Based on the Relationship between Viewpoint and Object

Tomohide IWAO<sup>†</sup> Msayuki KANBARA<sup>†</sup> and Naokazu YOKOYA<sup>†</sup>

<sup>†</sup> Graduate School of Information Science, Nara Institute of Science and Technology

Takayama 8916-5, Ikoma, Nara, 630-0192, Japan

E-mail: † {tomo-i, kanbara, yokoya}@is.naist.jp

**Abstract** Many algorithms which draw a shadow in real time have been proposed in computer graphics field. The most famous method among them is the shadow map method which calculates the distance from a light source to objects in advance, and stores the results in the buffer called shadow map. However, there is an aliasing problem caused by the low resolution of the shadow map allocated to the object in the scene, which depends on the relationship between viewpoint and objects. This paper describes a technique which can cope with the aliasing problem by generating a shadow map dynamically according to the relationship between viewpoint and objects. In experiments, the proposed method is applied to the conventional shadow map method and the perspective shadow map method which assigns a shadow map in consideration of the distance from a viewpoint to objects. Finally we discuss the usefulness of the proposed method based on the results of rendering images and processing time.

**Keyword** shadow map method, dynamic shadow map generation, perspective shadow map method, real time rendering

### 1. はじめに

近年、リアルタイムグラフィックス分野の発展は著しく、写実的なシーンが描画できるようになってきた。その要因のひとつとして挙げられるのが、コンピュータの高性能化であり、その演算速度が向上するにつれ

て、従来までは軽視されていたような、写実性を高めるためのエフェクトに注目が集まっている。その中でも、影は写実性を高めるためには欠くことのできない要素である。個々の物体の影を描画することで、シーンに現実感を与えることが可能である。

影を描画するアルゴリズムは現在まで多く研究さ

れており[1、2、3]、そのひとつにシャドウマップ法がある[4]。シャドウマップ法は、シーンの幾何学的計算を必要としないにもかかわらず、セルフシャドウも含めた正確な影が描画できるという利点を持つ。そのため、ゲーム等のリアルタイムアプリケーションを中心に広く利用されている。しかし、視点と光源の位置関係によっては、影の境界にエイリアシングが発生するという問題点がある。シャドウマップ法は、光源から見える領域内における光源からの距離をあらかじめ計算し、シャドウマップと呼ばれるテクスチャに保存しておき、これを後に行う影付けに利用する。エイリアシングは、主にシャドウマップの解像度不足によって引き起こされる。

本稿では、従来までは経験的に設定されてきたシャドウマップの範囲を、視点から見える領域に応じて動的に生成することで、影のエイリアシングを軽減する手法を提案する。フレームごとに描画されるシーンを解析し、視点から見えるオブジェクトを考慮して光源側の射影行列を決定することで、シーンに応じた効率的なシャドウマップを生成する。提案手法では、各シーンを解析する際、個々のオブジェクトの頂点計算のみを行うため、別の描画パスを必要とせず、最小のオーバーヘッドで影のエイリアシングを軽減することが可能である。

以降、2章では、従来影付けに一般的に使われているシャドウマップ法と関連研究について述べる。3章では提案手法の詳細について述べ、4章では有用性を示すための実験および考察を行う。最後に5章でまとめと今後の課題を述べる。

## 2. シャドウマップ法

本章ではシャドウマップ法の原理と問題点、および関連研究について順に述べる。

### 2.1 シャドウマップ法の原理

シャドウマップ法の原案は、Williams[4]により提案された。シャドウマップ法では、はじめに、光源から見たシーンを描画し、光源からの距離をシャドウマップと呼ばれるテクスチャに格納しておく。次に、描画対象のシーンをレンダリングする際に、シャドウマップを参照しながら、現在描画しようとしているピクセルが影であるかを判定することで、影付けを行う。

図1に、シャドウマップ法で影付けを行う具体的な例を示す。例として点Aを描画しようとする場合、まず点Aから光源までの距離を計算する。次にその値とシャドウマップに格納されている値を比較する。この場合では、シャドウマップには点Cまでの距離が格納されている。そのため、点Aから光源までの距離の方

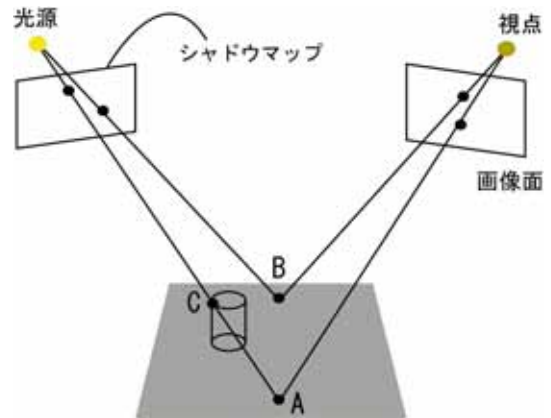


図1 シャドウマップ法の原理[5]

が長くなり、点Aは影であると判定される。一方、点Bについても同様の計算を行うが、この場合では点Bから光源までの距離とシャドウマップに格納されている値が等しいので、影ではないと判定される。

このように、シャドウマップ法ではピクセルごとに光源までの距離を計算する必要がある。そこで、Segal[6]らは、既存のハードウェアテクスチャマッピング機能を利用することで、この処理を単純化する手法を提案した。近年では、ハードウェアが投影テクスチャマッピング[7]、および影の判定をサポートするケースが増えており、商品レベルのグラフィックスボードでも容易に実時間でシャドウマッピングを行うことができるようになってきている[8、9]。

### 2.2 シャドウマップ法の問題点および関連研究

シャドウマップ法は有用な影付け手法であるが、視点が影に接近した場合、影の境界にエイリアシングが起きやすいという問題点がある。主な原因はシャドウマップの解像度不足であり、もし仮にバッファのメモリ容量が無限にあるとすれば、理論的には解決することが可能である。しかし実際は、ハードウェアの制約があるため、解決することが難しい問題であるとされてきた。

上記の問題を解決することを目的とした主なシャドウマップ法の改良手法として、Stammingerらのパースペクティブシャドウマップ(以下、PSM) [10]が挙げられる。この手法では、シャドウマップを従来のワールド座標系ではなく、透視変換後の空間でシャドウマップを生成している。例として図2にワールド空間と透視変換後の空間のビューボリュームを示す。透視変換により、視点に近い位置に存在するオブジェクトは大きくなるため、この状態でシャドウマップを生成すると、視点から近い位置に存在するオブジェクトに割り当てられるシャドウマップの解像度を高めることができる。

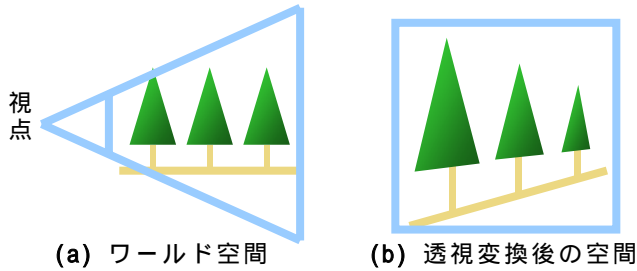


図2 PSMの原理[10]

PSMではフレームごとにシャドウマップを生成する必要はあるものの、エイリアシングを大幅に軽減することができる。一般的に影の解像度は、視点からの距離が近くなるにつれて不足するため、非常に効率的な手法である。しかし、従来手法と同様に、常にシャドウマップの解像度を高めるためには、視点側のビューボリュームを小さくするなどの微調整が必要となる。

また Fernando ら[11]は、視点に近い影の境界付近には解像度の高いシャドウマップを適用する手法を提案した。シャドウマップの階層化により、多くのシャドウマップを生成することになる。これにより、エイリアシング問題はほぼ解決することができるものの、解像度の高いシャドウマップの生成に、多くの描画パスを必要とするため、処理速度およびメモリの消費量に関しては課題が残った。そこで我々は、視点から見えない領域をシャドウマップの投影対象から除くことで、比較的少ないオーバーヘッドで動的に効率的なシャドウマップを生成する手法を提案する。また、提案手法は従来のシャドウマップ法だけでなく、PSMと組み合わせることでより効率的なシャドウマップを生成することが可能である。

### 3. 動的シャドウマップ生成

本章では、視点とオブジェクトの位置関係を考慮することでシャドウマップ法において生じるエイリアシングを軽減する手法について述べる。図3に示すように、従来のシャドウマップ法は、光源側のビューボリュームが視点からのビューボリュームを包含できるように、光源側の射影行列を決定する。しかしこの場合、実際には経験的に光源側のビューボリュームを決定することが一般的であるため、オブジェクトが描画されない領域もシャドウマップの投影対象となる可能性が生じ、個々のオブジェクトに割り当てられるシャドウマップの解像度が低くなるといった問題がある。そこで本稿では、動的なシーン(視点移動も含む)を描画する場合にフレームごとに視点からどの頂点が見えるかの可視判定を行い、可視頂点のみを包含する境界球を生成する。最後に、その境界球を包含するように、光源側の射影行列を決定することで、動的に効率的なシャドウ

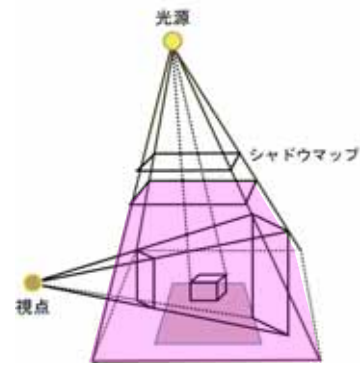


図3 ビューボリュームとシャドウマップの関係

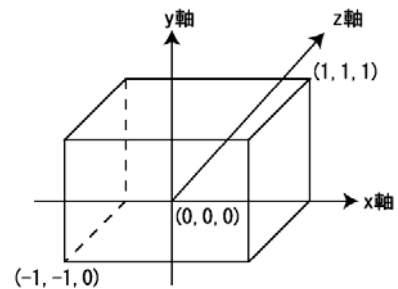


図4 射影座標系

マップを生成する。それぞれの処理の詳細を以下で説明する。

#### 3.1 オブジェクトの可視判定

フレームごとに視点から見えるオブジェクトの頂点を求めるために、可視判定を行う。はじめに、すべてのオブジェクトの頂点位置を取得し、それぞれの頂点に射影行列を適用し、射影座標系に変換する。

$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = PM \cdot CM \cdot WM \cdot \begin{pmatrix} x_L \\ y_L \\ z_L \end{pmatrix} \quad (1)$$

PM:射影行列

CM:カメラ行列

WM:ワールド行列

式(1)は、オブジェクトをローカル座標系 $(x_L, y_L, z_L)$ から、射影座標系 $(x_p, y_p, z_p)$ に変換する過程を示している。はじめにローカル座標系のオブジェクトにワールド行列を適用してオブジェクトをワールド空間に配置する。次に、カメラ行列を適用して視点から見た座標系に変換する。最後に、カメラ座標系において設定したビューボリュームを図4に示すような半立方体の射影座標系に変換する。射影座標系における可視頂点の $x, y, z$ 座標はそれぞれ、 $-1 < x < 1$ 、 $-1 < y < 1$ 、 $0 < z < 1$ の範囲内に変換される。なお、射影座標系への変換は可視判定のために使用するのみで、可視頂点はワールド座標系の値で保持しておく。

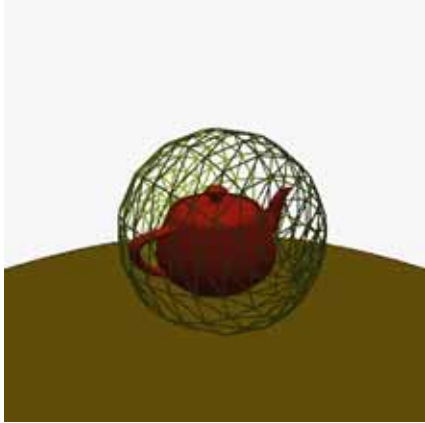


図5 設定された境界球を可視化した例

### 3.2 境界球の生成

可視判定により、ワールド座標系で保持されている頂点をすべて包含するような境界球を作成する。はじめに可視頂点をすべて包含するようなバウンディングボックスの8個の頂点を算出する。これらの頂点のx軸、y軸、z軸方向それぞれの最小値と最大値の中点を境界球の中心とする。最後に、中心からすべての可視頂点までの距離を計算し、その最大値を境界球の半径とする。ワイヤフレーム表示で可視化した境界球の例を図5に示す。

### 3.3 光源側の射影行列の決定

前のステップで求めた境界球を包含できるような光源側の射影行列を決定する。これにより、視点から見えるオブジェクトとほぼ同じ領域を、シャドウマップの投影対象とすることができる。そのためには、near面、far面、画角を決定する必要がある。はじめに、光源位置から境界球の中心までの距離を求め、その距離に境界球の半径を加えた距離をfar面として設定する。また、near面の位置は光源にできるだけ近い位置に手動で設定する(任意である)。対象範囲が境界球で設定されているという特性から画角も一意を求めることができる。これにより、境界球を包含できるような射影行列を決定することができる。この射影行列を用いて決定された光源側のビューボリュームを図6に示す。

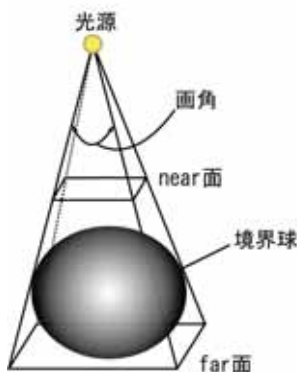


図6 ビューボリュームの決定方法

## 4. 実験

### 4.1 実験環境

本章では、提案手法の有効性を示すために、従来のシャドウマップ法(以下、SM)と2章で紹介したStamminger[10]らのパースペクティブシャドウマップ法(PSM)に提案手法を適用し、レンダリング結果および処理速度について考察する。なお、ここに示す実験結果はすべて、デスクトップPC(CPU Pentium4 2.54GHz、メモリ容量 512MB、グラフィックチップ Nvidia Geforce FX 5800 Ultra)で行った。リアルタイムCGの描画には3DライブラリDirectX9、およびそれぞれがサポートするプログラマブルシェーダ機能を、ハードウェアテクスチャマッチングおよび影の判定に利用した[12]。

### 4.2 レンダリング実験

はじめに、提案手法の有効性の比較を示すために、ひとつのオブジェクトのみを注視している同一シーンを、SMおよびSM+提案手法でレンダリングした結果とその際に生成されたシャドウマップ、境界球をそれぞれ図7に示す。同図(b)(c)から分かるように、SMでは、経験的に投影範囲が設定されているため、シーン中のオブジェクトに割り当てられるシャドウマップの解像度は低くなり、シャドウマップには利用されていない無駄な領域が多い。一方、SM+提案手法では、視点から見えている領域を効率良く境界球で包含し、その境界球をシャドウマップの投影対象としているため、シャドウマップのほとんどをシーン中のオブジェクトに割り当てることが可能である。このため、視点からのレンダリング結果(同図(a))においても、SMと比較してSM+提案手法の影は、SMでは表現できなかったバットの触覚も描画されていることから、精度が高いことが分かる。なお、ここに示した境界球に関しては、SMでは境界球は生成しないが、SM+提案手法と比較するために手動で生成したものである。

SM+提案手法の課題として、比較的広域かつ複数のオブジェクトが存在するシーンを注視しているケースにおいては、図5で示した境界球が大きくなる傾向があるため、SMと同様に視点に近い位置に存在するオブジェクトに割り当てられるシャドウマップが不足し、エイリアシングが発生する可能性がある。そのようなシーンに対して、SM+提案手法とPSM+提案手法の融合手法の適用を試みた。SMも加えた3種類の手法でレンダリングした結果とそのシャドウマップをそれぞれ図8に示す。SM+提案手法のシャドウマップは、SMと比較するとシャドウマップを有効利用しており影の精度は高いものの、すべてのオブジェクトに対して均等に解像度を割り当てているため、図8(b)に示すように視点に近い位置に描画されているバットの触覚の影

は正確に描画できていない。一方、提案手法と PSM の融合手法では、**図 8(c)**から視点に近い位置に存在するオブジェクトにより多くのシャドウマップを割り当てられている。これにより、視点に近い位置に存在するオブジェクトの影の精度が高くなっており、提案手法のみでは表現しきれなかったバツタの触覚が落とす影も描画できている(**図 8(c)**)。ただし、PSM を適用すると視点から遠い位置に存在するオブジェクトに割り当てるシャドウマップの解像度が逆に減少するため、形状が複雑なオブジェクトを描画する際には状況に応じて使い分ける必要があると考えられる。そのためには、カメラ座標系におけるオブジェクトの Z 値の分散やオブジェクトの複雑さ等を目安にしてこれら二つの手法を切り替えるという方法も考えられる。

#### 4.3 レンダリング速度の比較

本節では、それぞれの手法の処理速度について検討する。それぞれ 4 種類(SM、PSM、SM+提案手法、PSM+提案)の手法でポリゴン数を変化させてレンダリングした秒間フレームレートの比較を**表 1**に示す。このとき、処理速度に関しては、シャドウマップに投影されるオブジェクトの大きさによって処理速度は変化する。シャドウマップにアクセスして影の判定を行う処理は、GPU が行っており、シーンが変化すると逐一シャドウマップも変化し、結果的に GPU に与える計算量も変化するためである。これらの結果から提案手法を適用した場合でも、光源、視点、オブジェクトがフレームごとに変化するような動的な環境においても、比較的少ないオーバーヘッドでリアルタイム描画することが可能であることが確認できた。

表 1 フレームレートの比較

ポリゴン数	SM	PSM	SM+提案手法	PSM+提案手法
20000	144fps	131fps	129fps	124fps
37000	92fps	88fps	86fps	83fps
67000	57fps	55fps	54fps	53fps
100000	39fps	38fps	36fps	32fps

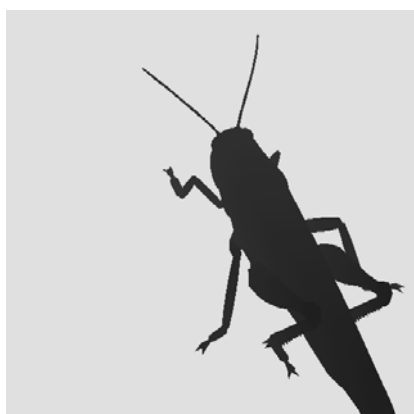
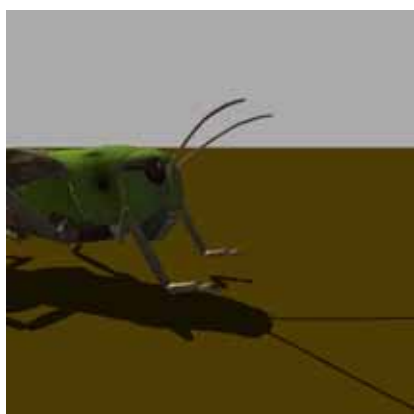
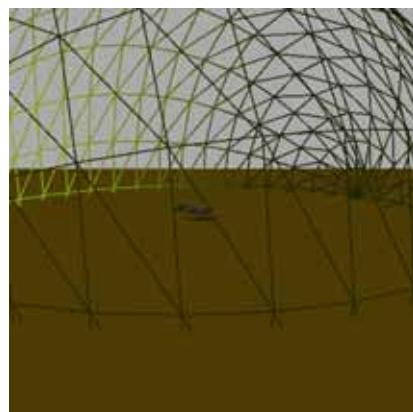
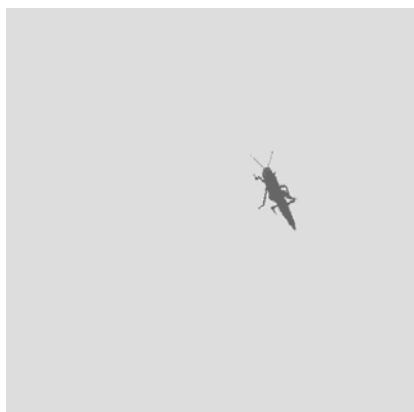
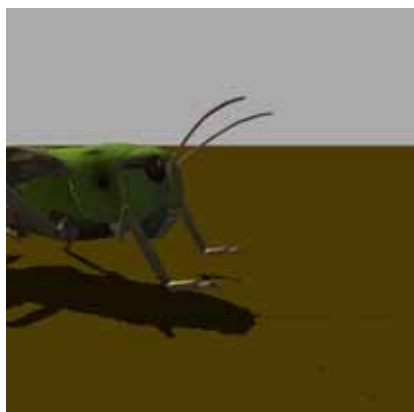
## 5. まとめ

本稿では、視点とオブジェクトの位置関係を考慮したシャドウマップを生成し、視点から見えない無駄な領域をシャドウマップの投影対象から除くことで、影の

エイリアシングを軽減する手法を提案した。さらにこの提案手法を従来のシャドウマップ法と、パースペクティブシャドウマップ法に適用し、実験・考察を行った。一般的に、フレームごとにシーンを解析するためにレンダリングを行うと処理速度は遅くなる。これは、解析する際に GPU から CPU へのデータ転送が頻繁に起こり、AGP バスのオーバーヘッドと GPU のパイプラインフラッシュのため、GPU の性能を十分に発揮することができないためである[13]。本稿では処理速度の向上を目指し、CPU 演算のみ行うことで、比較的少ないオーバーヘッドでエイリアシングを軽減することができた。今後は、処理速度の向上についての検討を行う。

## 文 献

- [1] A. Woo, P. Poulin, and A. Fournier : A survey of Shadow Algorithms. IEEE Computer Graphics and Applications, pp13-32, 1990.
- [2] F. Crow : Shadow algorithms for computer graphics, Proc. SIGGRAPH'77, pp242-248, 1977.
- [3] M. Hasenfratz, M. Lapierre, N. Holzschuch and F.X. Sillion,: A survey of real-time soft shadows algorithms.Computer Graphics Forum , pp753-774, 2003.
- [4] L. Williams : Casting curved shadows on curved surfaces. Proc. SIGGRAPH'78, pp270-274, 1978
- [5] 荒屋 真二 : 明解 3 次元コンピュータグラフィックス. pp116-118, 2003.
- [6] M. Segal, C. KorobkinR, V. Widenfelt, J. Foran, and P. Haerberli : Fast shadow and lighting effects using texture mapping. Proc. SIGGRAPH'92, pp249-252, 1992.
- [7] C. Everitt : Projective texture mapping. Nvidia technical report, 2001.  
[http://developer.nvidia.com/object/Projective\\_Texture\\_Mapping.html](http://developer.nvidia.com/object/Projective_Texture_Mapping.html)
- [8] C. Everitt, A. Rege, and C. Cebenoyan : Hardware shadow mapping. Nvidia technical report , 2001.  
[http://developer.nvidia.com/object/hwshadowmap\\_per.html](http://developer.nvidia.com/object/hwshadowmap_per.html)
- [9] J. Kilgard : Shadow mapping with today's OpenGL hardware. GDC2000 programming session, 2000.  
[http://developer.nvidia.com/object/ogl\\_shadowmaph.html](http://developer.nvidia.com/object/ogl_shadowmaph.html)
- [10] M. Stamminger, and G. Drettakis : Perspective shadow maps. Proc. SIGGRAPH2002, pp557-562, 2002.
- [11] R. Fernando, S. Fernandez, K. Bala, and D. P. Greenberg : Adaptive shadow maps. Proc. SIGGRAPH 2001, pp387-390 , 2001.
- [12] 今給黎 隆 : DirectX9 シェーダプログラミングブック, 2004.
- [13] 森, 五島, 中島, 富田 : 並列ベクトルプロセッサとしてのグラフィックスプロセッサ. 情報処理学会関西支部支部大会講演論文集, pp179-182, 2004.



(a) レンダリング結果

(b) 生成されたシャドウマップ

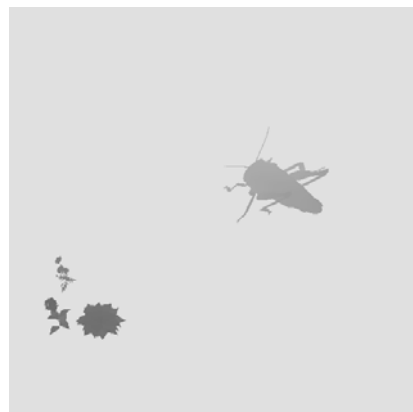
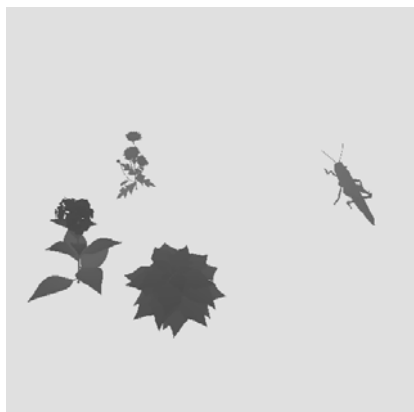
(c) 設定された境界球

図7 SM(上)とSM+提案手法(下)のレンダリング結果 (SMでは手動で設定)

レンダリング結果



生成されたシャドウマップ



(a) SM

(b) SM+提案手法

(c) PSM+提案手法

図8 複数のオブジェクトが存在するシーンにおけるレンダリング結果とシャドウマップの比較