

修士論文

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

2005年 2月 3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。

岩尾 友秀

審査委員：

横矢 直和教授

千原 國宏教授

山澤 一誠助教授

# 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

### 内容梗概

リアルタイムグラフィックス分野において、写実性を高めるために影は欠くことのできない要素である。影を描画するアルゴリズムは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、光源から物体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。まず、フレーム毎にオブジェクトの可視判定を行い、可視頂点のみを包含する境界球を作成する。次に、その境界球を包含するビューボリュームが生成できるように、光源側の投影パラメータを決定することで、視点から見えない無駄な領域を除いた効率的なシャドウマップを動的に生成する。本論文では、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。

---

\*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-ISMT0351016、 2005 年2 月3日

## キーワード

コンピュータグラフィックス、シャドウマップ法、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、動的シャドウマップ生成

# Dynamic Shadow Map Generation Using Relationship between Positions of Viewpoint and Object \*

Tomohide Iwao

## Abstract

In real time computer graphics, a shadow is one of the most important elements. Many algorithms which draw a shadow in real time have been proposed. 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 a buffer called the shadow map. The shadow map method is computationally efficient and can draw exact shadows. 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 new technique which can cope with the aliasing problem by generating a shadow map dynamically according to the relationship between viewpoint and objects. First, the visibility of objects is examined at every frame, and a boundary ball containing the visible vertices is created. Next, a shadow map is dynamically generated by determining the projection parameters so that the boundary ball is included in a view-volume. 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.

---

\*Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT0351016, February 3, 2005.

Keywords:

computer graphics, shadow map method, perspective shadow map method,  
real time rendering, dynamic shadow map generation

# 目次

1. はじめに	1
2. 研究背景および本研究の位置付け	3
2.1 コンピュータグラフィックスで扱う影の種類	3
2.2 実時間影付けの従来手法	4
2.3 シャドウマップ法の原理および問題点	8
2.4 シャドウマップ法の改良および本研究の位置付け	11
3. 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	15
3.1 提案手法の概要	15
3.2 オブジェクトの可視判定	16
3.3 可視頂点を包含する境界球の生成	18
3.4 光源側の射影行列の決定	19
3.5 レンダリング	22
3.6 提案手法とPSMの融合	22
4. 実験結果	26
4.1 実験環境	26
4.2 通常シャドウマップ法と提案手法の比較	27
4.3 シャドウマップ法および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た比較実験	30
4.4 レンダリング速度の比較	38
5. まとめ	40
謝辞	
参考文献	

## 目次

1. ハードシャドウとソフトシャドウ	3
2. 平面投影シャドウ	5
3. 投影テクスチャマッピング	6
4. シャドウボリューム法	7
5. シャドウマップ法の原理	9
6. エイリアシングが発生する場合	10
7. ビューボリュームとシャドウマップの関係	14
8. 提案手法の流れ	15
9. ローカル座標系からカメラ座標系への変換過程	17
10. 射影座標系	17
11. 可視頂点のみで生成する境界球	18
12. 生成された境界球を可視化した例	19
13. ビューボリュームの決定方法	20
14. シャドウマップに投影された境界球の例	21
15. オブジェクト間の距離が接近している場合の境界球の例	21
16. オブジェクト間の距離が離れている場合の境界球の例	22
17. PSMの原理	24
18. 通常のシャドウマップ法によるレンダリング	24
19. パースペクティブシャドウマップ法によるレンダリング	25



20. 提案手法とPSMの融合手法の流れ	25
21. SMとSM+提案手法によるレンダリング結果の比較	28
22. SMによるセルフシャドウのレンダリング	29
23. SM+提案手法によるセルフシャドウのレンダリング	29
24. シーン1 : SM	32
25. シーン1 : PSM	32
26. シーン1 : SM+提案手法	33
27. シーン1 : PSM+提案手法	33
28. シーン2 : SM	34
29. シーン2 : PSM	34
30. シーン2 : SM+提案手法	35
31. シーン2 : PSM+提案手法	35
32. シーン3 : SM	36
33. シーン3 : PSM	36
34. シーン3 : SM+提案手法	37
35. シーン3 : PSM+提案手法	37
36 SMとSM+提案手法の処理速度の比較	39
37 PSMとPSM+提案手法の処理速度の比較	39

## 表目次

1. 影付け手法の特徴	4
2. 実験環境	26
3. フレームレートの比較	38

## 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、演算速度が向上するにつれて、従来は軽視されていた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[1、2、3]。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。近年では、現実世界とコンピュータグラフィックスの融合分野である拡張現実感に関する研究分野においても、正確な影付けを導入する試み[4、5、6]がなされており、影の精度はもとより処理速度に関しても優れたアルゴリズムが求められている。

リアルタイムに影付けを行うアルゴリズムは現在までも多く研究されており[7、6]、そのひとつにシャドウマップ法がある[9]。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ことなど利点が多い。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。具体的には、シャドウマップをシーン中にテクスチャマッピングし、影であるか否かを判定する。このため、影の精度はシャドウマップの解像度に依存する。エイリアシングは、主に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する場合に発生する。

本論文では、従来まではシーン全体を包含するように設定されてきたシャドウマップがカバーする有効範囲を、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、視点から見える領域を考慮してシャドウマップの設定範囲を表す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い、別の描画パスを必要としないため、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、従来研究を概観し、シャドウマップ法とその問題点について述べる。3 章では、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の動的生成手法について詳細に述べ、4 章では、提案手法を従来のシャドウマップ法およびパースペクティブシャドウマップ法に適用した実験結果を示し、レンダリング結果と処理速度に関して考察を述べる。そして最後に 5 章で本論文をまとめる。

## 2. 研究背景および本研究の位置付け

本章では、コンピュータグラフィックス分野における影付けの研究背景および本研究の位置付けについて述べる。まず 2.1 節では、影の種類や特徴について説明し、2.2 節では、リアルタイムの影付けの従来手法を紹介する。2.3 節では、本研究で扱う影付け手法であるシャドウマップ法について詳述し、最後に、2.4 節では本研究の位置付けについて述べる。

### 2.1 コンピュータグラフィックスで扱う影の種類

一般的に、CG で描画される影の種類は、大きく分けて 2 つある。1 つはハードシャドウと呼ばれるもので、影の境界がくっきりした影である。主にリアルタイムの影付けに用いられることが多い。もう 1 つは、ソフトシャドウと呼ばれるもので、ハードシャドウとは対照的に影の境界はぼやけたもので、一般的に実環境で多く見られる影である。図 1 にハードシャドウとソフトシャドウの違いを示す。ハードシャドウは、点光源によって作られる影であり、物体にさえぎられて光が全くあたらない領域(本影)に生じる。一方、ソフトシャドウは、大きさを持つ光源(面光源)などによって作られる影であり、光が全くあたらない本影領域と、光源の一部分に照らされているため本影より少し明るい領域(半影)の 2 つの領域から成る。

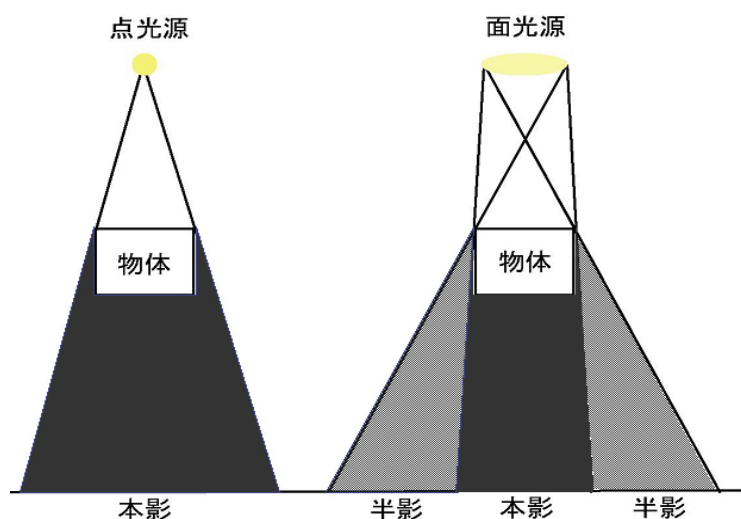


図 1 ハードシャドウとソフトシャドウ

現実世界において、室内では電球など大きさのある光源が存在している場合が多く、また屋外であっても太陽光線がとどかない領域も生じるため、結果として影は、境界がぼやけたソフトシャドウになる場合が多い。したがって、現実世界を忠実に再現しようとするれば、ソフトシャドウとして影を表現することが理想的である。そのため、実時間でソフトシャドウを生成する試みも多く研究されている[10、11、12、13]。しかし、現実世界の光源環境は非常に複雑であり、正確なソフトシャドウをリアルタイム生成することは難しい。また、リアルタイムグラフィックス分野では、異なる光源位置で描画されたハードシャドウを組み合わせることによってソフトシャドウを近似的に生成することが多いため、精度の高いハードシャドウを高速に描画するアルゴリズムの研究が依然として主流となっている。

## 2.2 実時間影付けの従来手法

リアルタイムの影付け手法には、多くの様々な手法が存在するが、ここでは現在主に使用されているものについて紹介する。代表的な影付け手法の特徴を表1にまとめた。それぞれ処理速度、任意形状物体に対する影付けが可能かどうか、セルフシャドウが表現可能かどうかについて記した。ここでいうセルフシャドウとは、物体が自分自身に落とす影のことを言う。以下、平面投影シャドウ、投影テクスチャシャドウ、シャドウボリューム法について簡単に述べる。その後、次節で本研究と深く関連するシャドウマップ法について詳述する。

表1 影付け手法の特徴

影付け手法名	処理速度	任意形状	セルフシャドウ
平面投影 シャドウ	速い	×	×
投影テクスチャ シャドウ	中	○	×
シャドウ ボリューム法	遅い	○	○
シャドウマップ 法	遅い	○	○

### 2.2.1 平面投影シャドウ

平面投影シャドウは、Swan[14]によって提案されたリアルタイムの影付け手法である。光源に照らされたオブジェクトが平面に影を落とす場合、物体の各頂点は平面上に対応する点を持っている。すなわち、**図 2**に示すように光源と頂点 A を通る直線を延長し平面と交差した点 P が、対応する影の位置になる。**図 2**で示した頂点 A から平面上の対応点 P への変換は、光源と平面の位置が求まれば一意に決定できる。このため、全ての頂点についてこの変換を行った後にオブジェクトを描画すれば、影の領域を描画することができる[15]。この手法では、影を落とす対象は平面に限られており、任意形状のオブジェクトへの影付けには対応していない。また、シーン中のオブジェクトの個数が増加するにつれて描画回数も増えるという問題点もある。しかし、オブジェクトが少なくかつ平面に影を落とす場合には、非常に高速に影付けができるため、現在でも使用される機会が多い手法である。

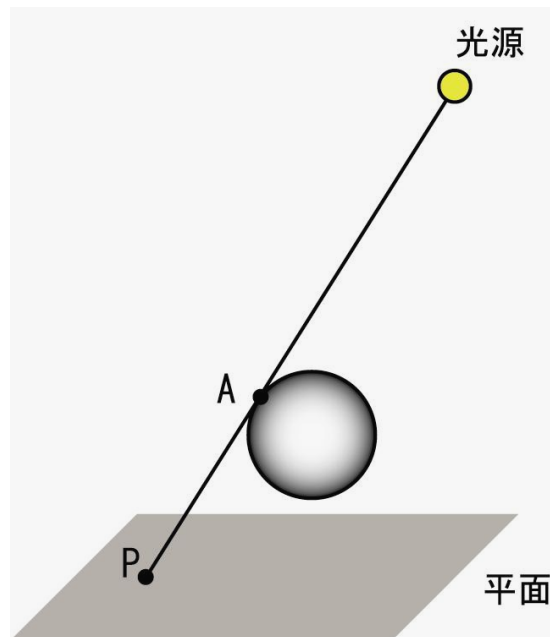


図 2 平面投影シャドウ

### 2.2.2 投影テクスチャシャドウ

投影テクスチャシャドウは、任意形状のオブジェクトに対しても影付けを行うことができることに加え、処理速度も比較的速いため、現在でもゲーム等を中心としたリアルタイムアプリケーションで多く採用されている手法である。この手法は、投影テクスチャマッピング[16]と呼ばれる技術を利用して影付けを行う。投影テクスチャマッピングは、テクスチャ画像を光源方向から3次元シーンに投影するマッピング手法である。投影方法は、大きく分けて透視投影と平行投影の二種類があり、一般的に点光源では透視投影が使用される。図3に示すように、基本的には、テクスチャのどのピクセルが3次元シーンに投影されるかを考えるのではなく、逆に3次元シーン中のオブジェクトの各頂点がテクスチャ画像のどのピクセルに対応するかを計算する。

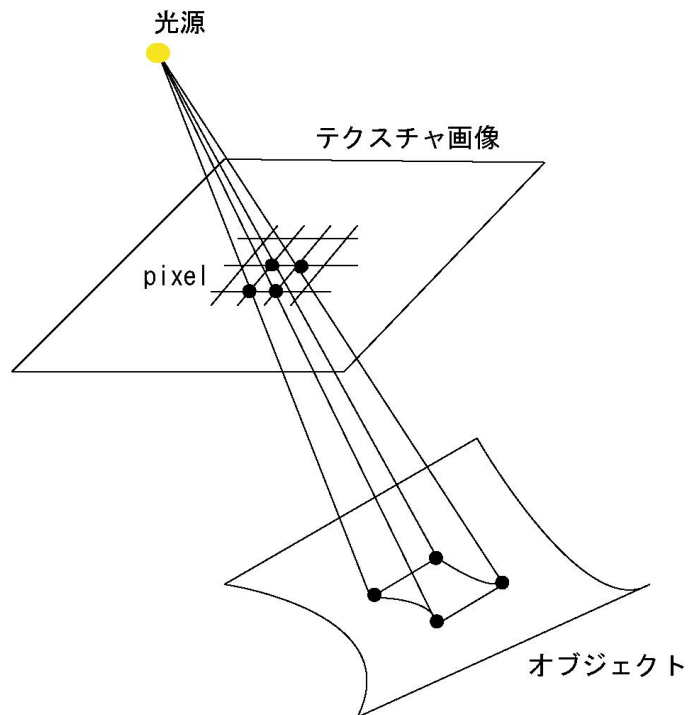


図3 投影テクスチャマッピング



原理的には透視変換と同じであるため、3次元シーン中のオブジェクトの各頂点と光源側の透視変換行列により、各頂点のテクスチャ座標を得ることができる。実際には、このテクスチャ画像にオブジェクトのシルエットをあらかじめレンダリングしておくことで影付けを行う。ただし、この手法も前述の平面投影シャドウと同様に、影の投射はオブジェクト単位であるため、多数のオブジェクトが存在する複雑なシーンには不向きである。また、自身の影が自身に投射されるセルフシャドウを表現することも不可能である。

### 2.2.3 シャドウボリューム法

シャドウボリューム法は、Crow[17]によって提案されたリアルタイムの影付け手法である。この手法は、セルフシャドウまで含めた正確な影付けを行うことができる等、利点が多く、現在では次節で紹介するシャドウマップ法と並んで広く利用されている影付け手法のひとつである。

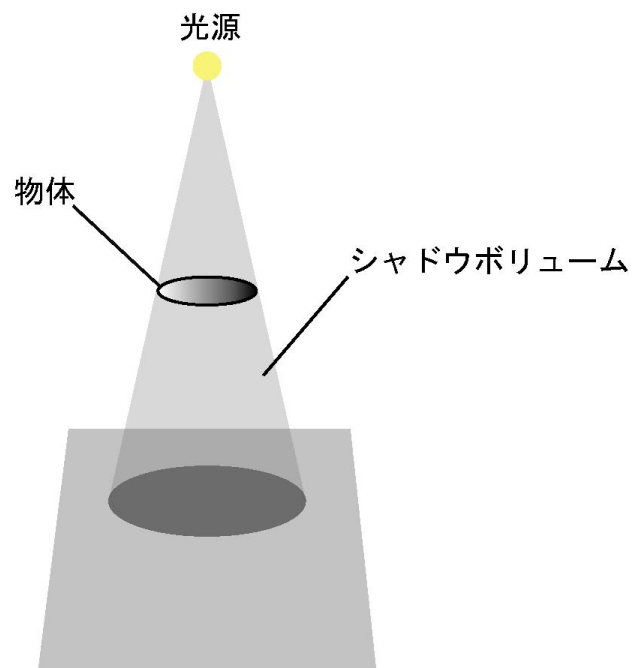


図4 シャドウボリューム法

シャドウボリューム法は、**図 4** に示すように光源方向から見えるシーン中の物体の頂点を引きのぼし、光が当たらない領域(シャドウボリューム)を生成する。実際の影付けは、描画するピクセルがシャドウボリュームの中に入っているかどうかで判断する。シャドウボリュームの生成は、モデルの各頂点に引き延ばすためのダミーポリゴンを用意するのが一般的である。このため、遮蔽物体のポリゴン数が増加するにつれて、計算量も増加するため、複雑なシーンをレンダリングする場合には不向きである。シャドウボリューム法を高速化する手法も多く研究されているが[18、19、20]、シーンの幾何学的計算が必須であるという特性上、どの手法も劇的に処理速度を改善するまでには至っていない。

## 2.3 シャドウマップ法の原理および問題点

本研究は、リアルタイムの影付け手法であるシャドウマップ法の問題点を解決することを目指したものである。本節では、研究対象であるシャドウマップ法の原理および問題点について述べる。

### 2.3.1 シャドウマップ法の原理

シャドウマップ法の原案は、Williams[9]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、光源からオブジェクトまでの距離をシャドウマップと呼ばれるテクスチャバッファに格納しておく。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否かを判定することで、影付けを行う。

**図 5** 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。

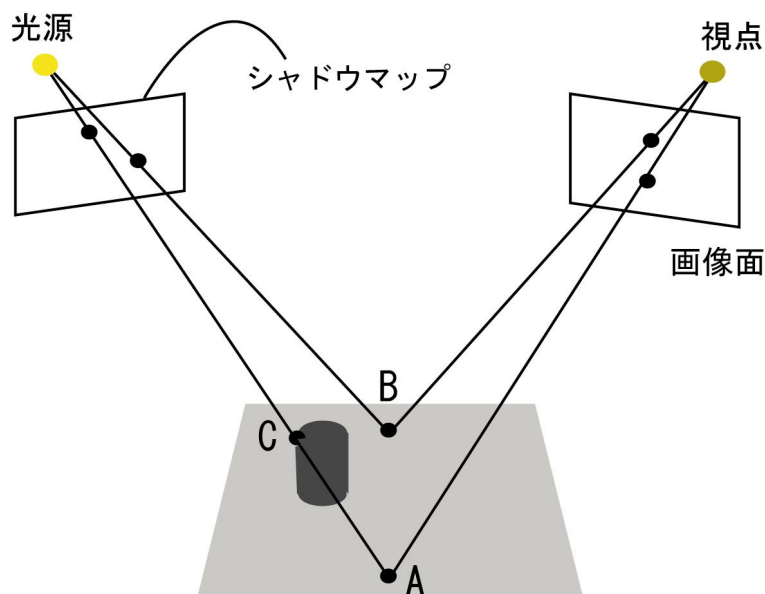


図5 シャドウマップ法の原理[21]

Segal[22]らは、2.1節で紹介した投影テクスチャマッピングを利用することで、影であるか否かの判定処理を単純化する手法を提案した。投影テクスチャマッピングでは、3次元シーン中のオブジェクトの各頂点に光源側の透視変換行列を掛けることで、各頂点のテクスチャ座標を得ることができ、テクスチャ座標を計算する際にその頂点から光源までの距離も分かるため、非常に効率が良い。近年では、ハードウェアが投影テクスチャマッピング[16]、および影の判定をサポートする場合[23、24]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ことから、シャドウマップ法の実装ではこの機能を使用することが一般的である。

### 2.3.2 シャドウマップ法の問題点

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。エイリアシングが起きるケースを図6に示す。3次元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1ピクセルに対し、画像面では2ピクセル以上の大きさとなるため、オーバーサンプリングが起きている。そのため、シャドウマップの情

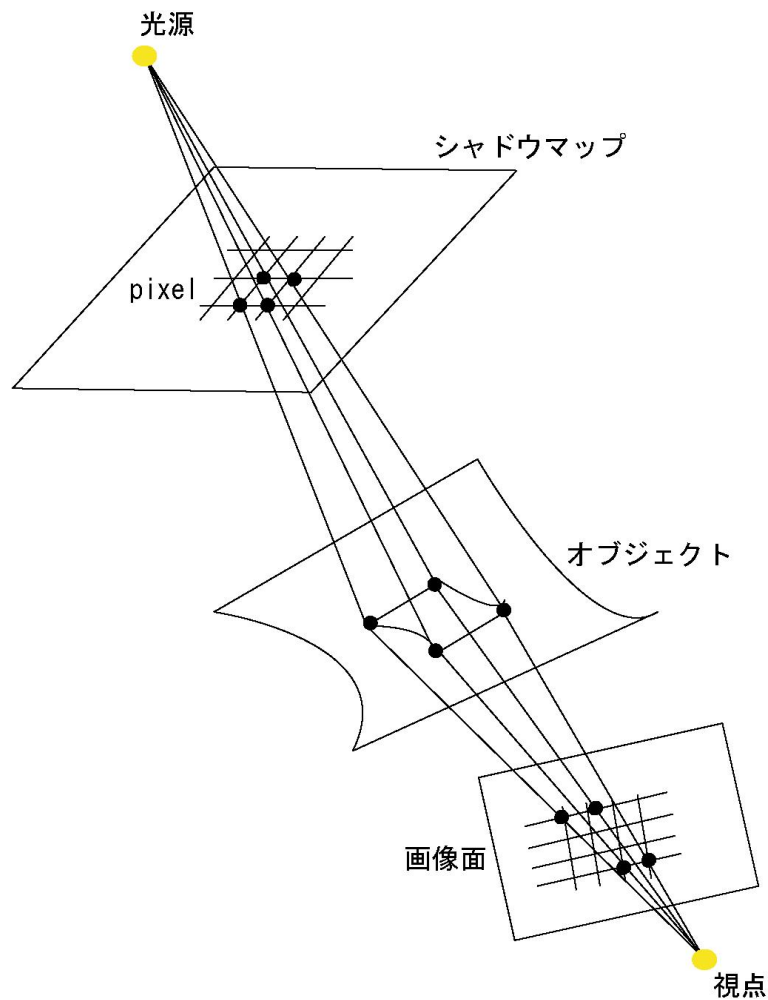


図 6 エイリアシングが発生する場合 [25]

報量が不足することによりエイリアシングが発生し、これは視点が影に近づくにつれて顕著になる。影を落とすオブジェクトが複雑な形状を有している場合には、影そのものが描画されない場合もある。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、解決することは困難である。

もう一つの問題点として、バイアス問題が知られている。例を挙げて説明すると、図 5 において影にならないピクセルは、シャドウマップに格納されている値と、描画対象のピクセルから光源までの距離が一致しなければならない。しかし

実際は、シャドウマップに格納されている距離値は、光源方向から量子化された離散的な値であるため、視点方向から量子化されたピクセルから算出された光源までの距離値とは微妙に異なる[24]。そのため、実際は影でないピクセルが影であると判定されるような現象が起きる。しかしこの問題は、シャドウマップに格納する距離値にバイアスを加えておく等の工夫によって、実装上は解決することができる。このため、近年のシャドウマップ法の研究対象としては、影の精度を高めてエイリアシングを軽減することや、あるいはエイリアシングが目立たないようにソフトシャドウを導入するといったことに焦点を当てているものが多い。

## 2.4 シャドウマップ法の改良および本研究の位置付け

本節では、シャドウマップ法の改良に関する従来研究を紹介した後に、本研究の位置付けについて述べる。

### 2.4.1 従来研究

シャドウマップ法は有用な影付け手法であるが、影の境界にエイリアシングが起りやすいという特性から多くの改良研究がなされてきた。改良研究のアプローチは、大きく分けると二種類である。ひとつは、エイリアシングを目立たなくするためにソフトシャドウを導入するアプローチである。二つ目は、影の輪郭を明確に描画することでエイリアシングを軽減するというアプローチである。

#### [ソフトシャドウを導入するアプローチ]

はじめに、主にソフトシャドウを導入することを基本方針としている従来研究について紹介する。Reeves[26]らは、シャドウマップにフィルタを適用することによってアンチエイリアシングを施し、擬似ソフトシャドウを生成することで、エイリアシングを目立たなくする手法を提案した。これがシャドウマップ法に関する最初の改良手法である。Zhang[27]らは、ハードウェアテクスチャマッピング機能が処理速度に関してボトルネックになることが多いことに着目し、シャドウマップに格納されている距離値を視点側のシーンに逆投影することで影付けを判定し、テクスチャマッピング機能を使用しない影付け手法を提案した。ただし、予め視点側からシーンをレンダリングしておく必要があるため、テクスチャマッピング機能がボトルネックとならない場合では利点

を生かすことができない。また Locovic[28]らは、光源からの光の減衰率や透過率を算出することで、本来は高解像度のシャドウマップを必要とする煙や髪の毛などの影付けを容易に行うことができる手法を提案した。さらに Dachsbacher[29]らは、拡散反射光等も考慮することで、より現実感のあるソフトシャドウをリアルタイムで生成することに成功した。

#### [影の輪郭を明確に描画するアプローチ]

次に、影の輪郭を明確に描画することを基本方針としている従来研究について紹介する。はじめに、Segal[22]らは投影テクスチャマッピングを利用することで、シャドウマップに格納されている距離値と視点側のピクセルから光源までの距離との比較処理の過程を、グラフィックスハードウェア化できることを示した。この研究によって、シャドウマップ法は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が一般的となり、シャドウマップ法に関する改良研究が盛んに行われるようになった。Fernando ら[30]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、多くの描画パスが必要となり、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。また Sen[31]らは、従来までは別個の研究対象とされていたシャドウボリューム法[17]をシャドウマップ法に適用したハイブリッド手法を提案した。シャドウボリューム法を使用してオブジェクトのシルエット座標を格納するシルエットマップと呼ばれるテクスチャを生成し、それと従来のシャドウマップの距離値を利用することでエイリアシングを軽減することに成功した。しかし、シルエットマップを生成する際に使用するシャドウボリューム法は、ポリゴン数が多くなるにつれて計算量が多くなるため、複雑なシーンを対象とした場合には課題が残った。

近年の注目すべき改良研究としては、Stamminger らのパースペクティブシャドウマップ(以下、PSM) [32]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。シャドウマップ生成時にシーン全体を透視変換することにより、視点に近い位置に存在するオブジェクトは大きくなり、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に成功した。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、視点に近い位置に存在するオブジェクトにより多くの解像度を割り当てることは非常に効率的である。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、光源側のビューボリュームを小さくするなどシャドウマップの設定範囲には微調整が必要となる。

#### 2.4.2 本研究の位置付け

本研究は、従来のシャドウマップ法に見られるエイリアシングを、比較的少ないオーバーヘッドで軽減することを目的としている。そこで、エイリアシングを引き起こす主な原因となっているシャドウマップの生成過程に注目した。図7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ってしまう。そこで本研究では、動的なシーン(視点移動も含む)を描画する際に、フレームごとに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、動的に効率的なシャドウマップを生成することを目指した。提案手法では、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し、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。そのため、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き、動的に効率的なシャドウマップを生成することが可能となる。さらに、提案手法は従来のシャドウマップ法だけでなく、前項で紹介したパースペクティブシャドウマップ法(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できる。

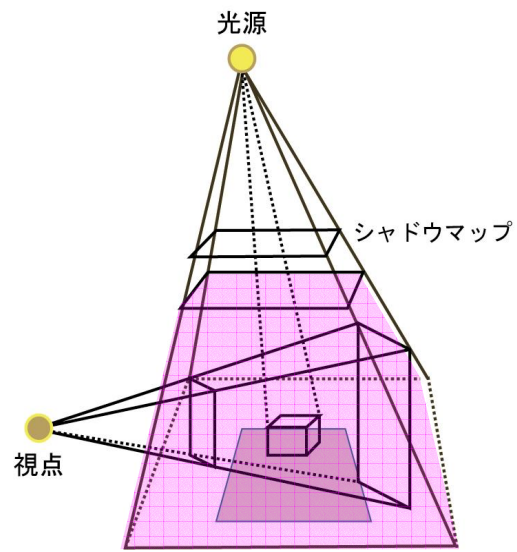


図7 ビューボリュームとシャドウマップの関係



### 3. 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

本章では、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の動的生成法[33、34]について述べる。3.1節では、提案手法の概要について説明し、3.2節から3.4節では提案手法の処理過程の詳細について述べる。最後の3.5節では、提案手法とPSMの融合手法について述べる。

#### 3.1 提案手法の概要

本節では、提案手法の概要について述べる。フレームごとに描画されるシーンを解析し、視点から見える領域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。図8に提案手法の流れを示す。

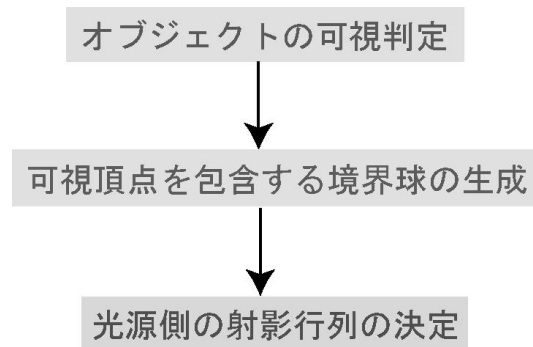


図8 提案手法の流れ

はじめにフレームごとに視点からどのオブジェクトが見えるかの可視判定を行う。次に、可視頂点のみを包含する境界球を生成する。最後に、光源側のビューボリュームが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マップを生成する。一般的に、フレームごとにシーンを解析するためには、あらかじめ視点側から一回レンダリングを行い、後にこのデータを使用する場合が多い。しかしこの場合、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 の処理を一時中断する必要があるため、GPU の性能を十分に発揮することができないためである[35]。提案手法では CPU 演算のみ行うことで、処理速度の向上を目指した。次節からは、提案手法の詳細な処理について説明する。

### 3.2 オブジェクトの可視判定

本節では、オブジェクトの可視判定について述べる。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。図 9 に、ローカル座標系からカメラ座標系への変換過程を示す。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 10 に示すような半立方体の射影座標系に変換する。射影変換は、一般的なグラフィックスボードのレンダリング過程においてハードウェア内部で可視判定に使用されていることが多い。

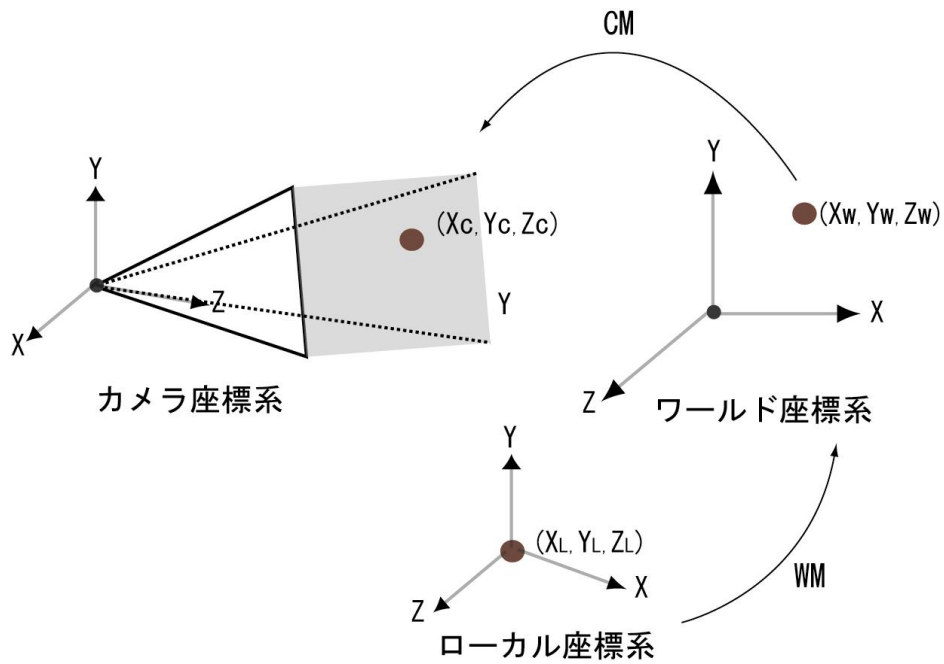


図 9 ローカル座標系からカメラ座標系への変換過程

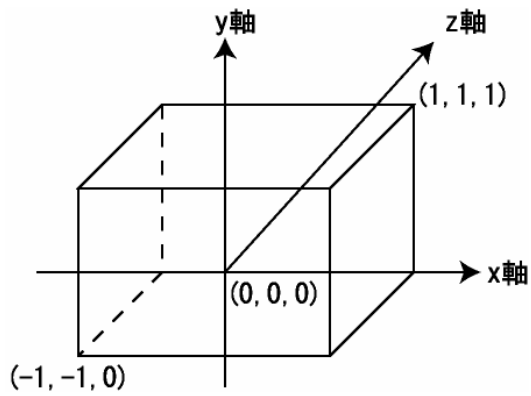


図 10 射影座標系

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。逆に、ビューリウム内に存在しない頂点はこの範囲外となる。なお、射影座標系への変換は可視判定のためのみに使用するので、可視頂点はワールド座標系の値で保持しておく。

### 3.3 可視頂点を包含する境界球の生成

本節では、可視頂点を包含する境界球の生成過程について述べる。可視判定により、ワールド座標系の値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生成する。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。これらの頂点の座標系における  $x$  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。図 11 に可視頂点のみで生成する境界球の例を示す。このように、1つのオブジェクトでも頂点単位で可視判定しているため、効率的な境界球を生成できる。さらに、ワイヤフレーム表示で可視化した境界球の例を図 12 に示す。ただし、この場合では、床が影を落とす場合はないため、可視判定する対象からは除外している。オブジェクトが 1つの場合(同図(a))と比較すると、オブジェクトが複数存在する場合(同図(b))の方が広い領域を包含しなければならないため、境界球は比較的大きくなっていることが分かる。

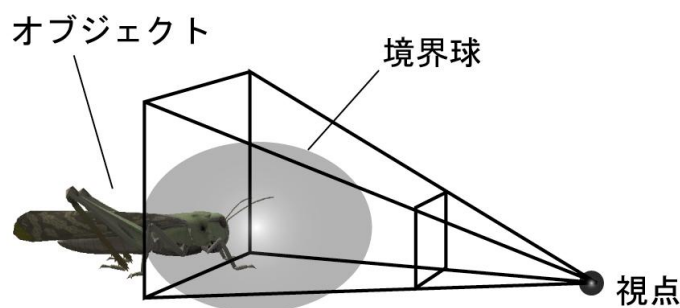
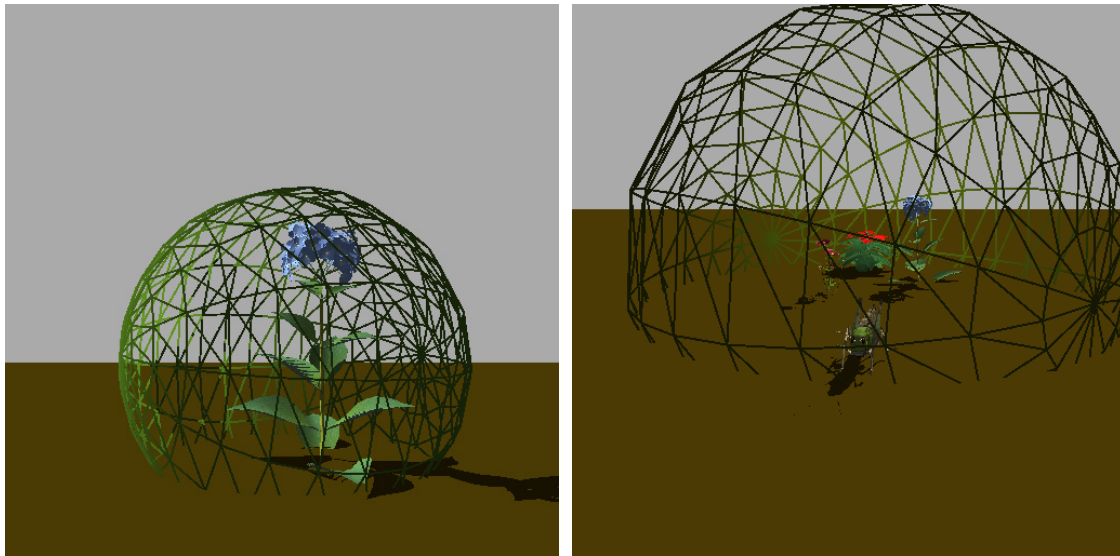


図 11 可視頂点のみで生成する境界球



(a) オブジェクトが1つの場合

(b) オブジェクトが複数存在する場合

図 12 生成された境界球を可視化した例

### 3.4 光源側の射影行列の決定

前のステップで求めた光源側のビューボリュームが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。そのため、視点から見えない無駄な領域を除いた部分を、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、光源側のビューボリュームの **near** 面、**far** 面、画角を決定する必要がある。図 13 に示すように、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位置に **far** 面を設定する。また、**near** 面の位置は一般的なビューボリュームと同じ要領で設定する。本手法ではシャドウマップのカバーする範囲を境界球で設定されているという特性から画角も一意に求めることができる。これにより、光源側のビューボリュームが境界球を包含できるような射影行列を容易に決定することができる。また、シャドウマップに投影された境界球を可視化した例を図 14 に示す。同図 (a) に示した設定された境界球をシャドウマップに投影したものが同図 (b) である。ワイヤーフレーム表示で可視した境界球のみがシャドウマップに投影されていることが分かる。

以下に、オブジェクト間の距離間隔を変化させて提案手法で二つのオブジェクトをレンダリングした結果をそれぞれ図 15、16 に示す。図 15 に示す結果では、二つのオブジェクトが接近しているため、境界球は小さく、かつシャドウマップに投影された境界球(同図(b))においても二つのオブジェクトが大きく投影されていることが分かる。一方、図 16 では、二つのオブジェクト間の距離が離れているため、境界球は大きく、二つ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、図 15 と比べると低いことが分かる。

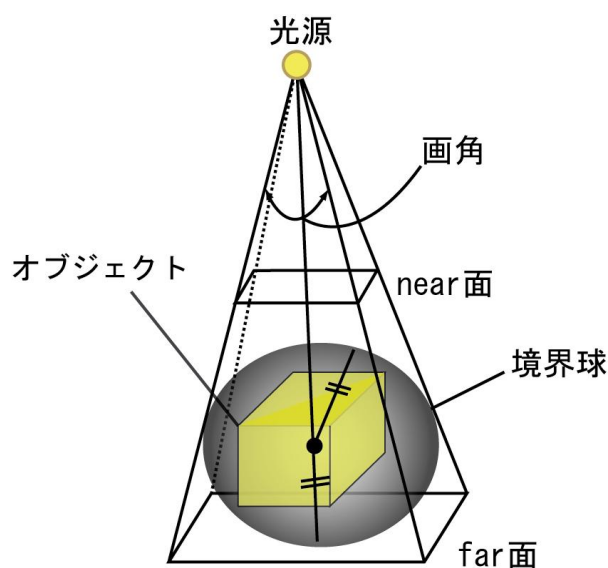
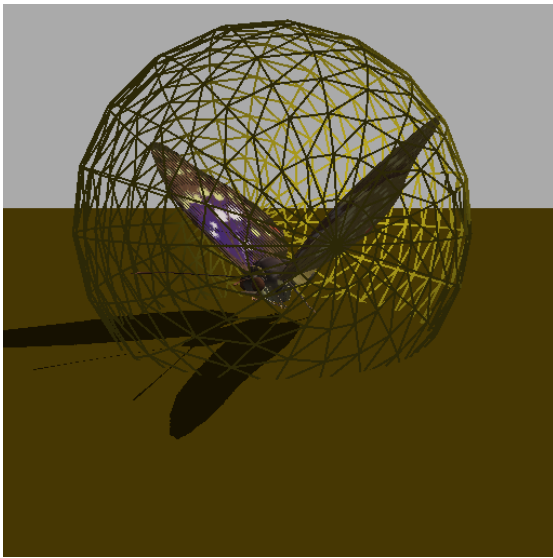
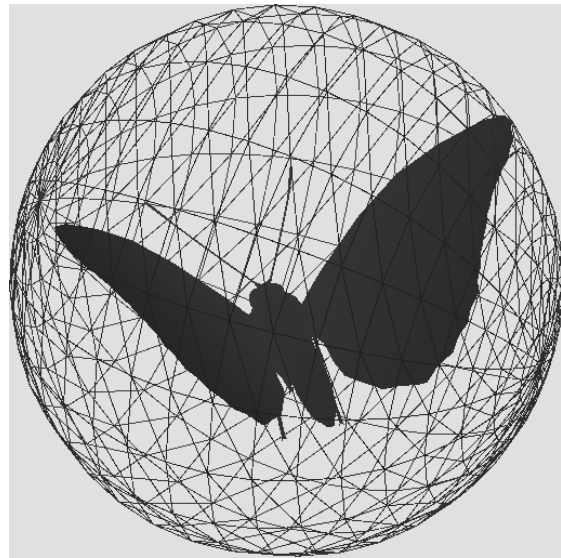


図 13 ビューボリュームの決定方法

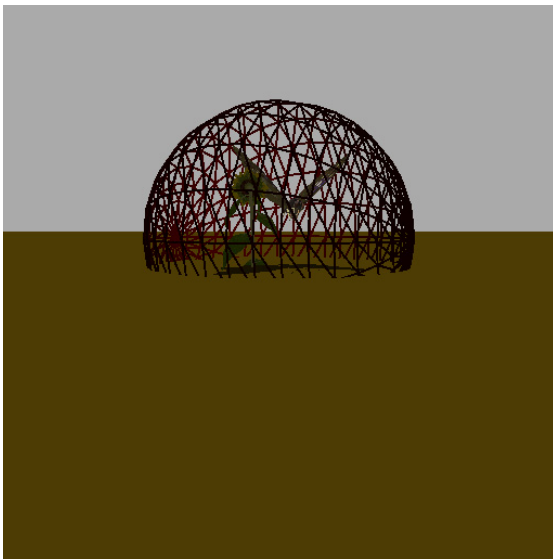


(a) 設定された境界球

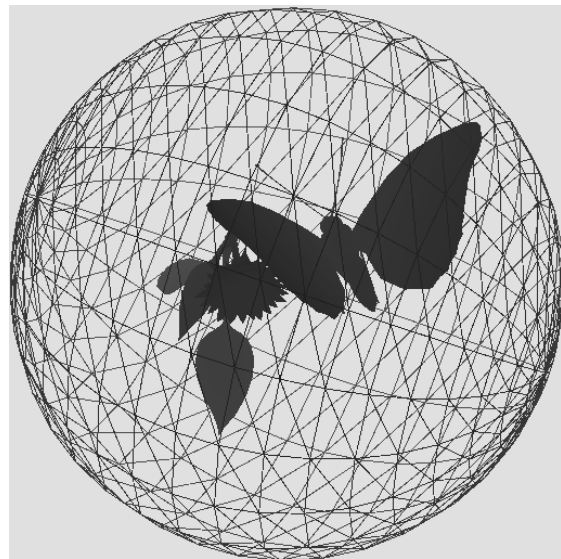


(b) シャドウマップ

図 14 シャドウマップに投影された境界球の例

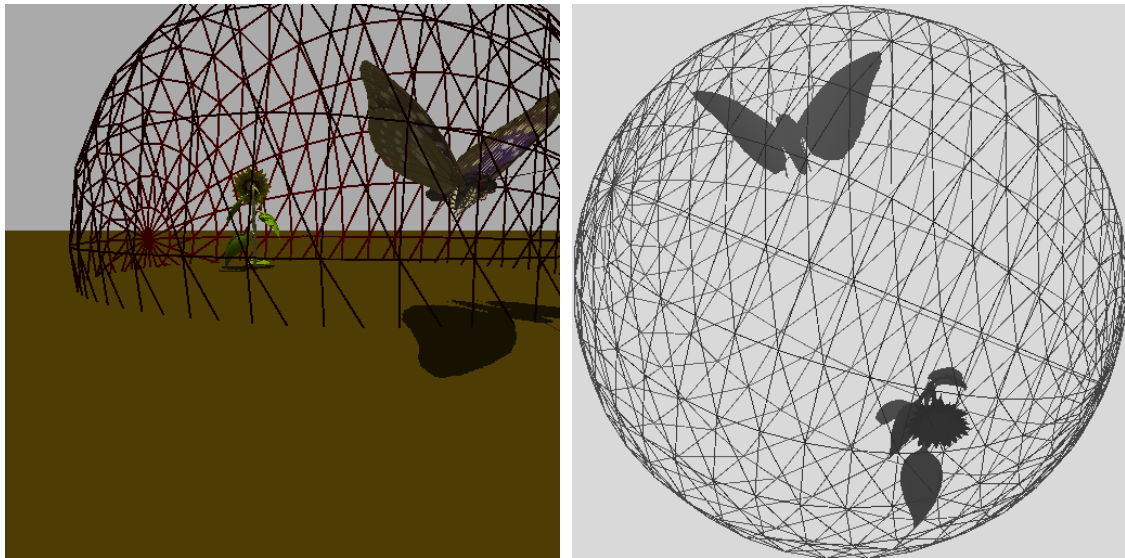


(a) 設定された境界球



(b) シャドウマップ

図 15 オブジェクト間の距離が接近している場合の境界球の例



(a) 設定された境界球

(b) シャドウマップ

図 16 オブジェクト間の距離が離れている場合の境界球の例

### 3.5 レンダリング

本節では、提案手法を従来のシャドウマップ法に適用したレンダリングについて説明する。前述したように、本手法では、はじめにオブジェクトの可視判定を行い、可視頂点を包含するような境界球を生成する。最後に境界球を包含するように、光源側の射影行列(パラメータ)を決定する。そして通常シャドウマップ法と同様に、ハードウェアテクスチャマッピングにより、シャドウマップをシーンに投影テクスチャマッピングし、視点側のピクセルが影であるか否かを判定する。影であると判定されたピクセルは、環境光のみで色付けし、光で照らされているピクセルは拡散光を考慮して色付けした。なお、境界球は説明のために可視化したものであり、実際のレンダリング結果では表示されない。

### 3.6 提案手法とパースペクティブシャドウマップの融合



本節では、提案手法とパースペクティブシャドウマップ(PSM)の融合手法について述べる。

### 3.6.1 提案手法の問題点

提案手法は、通常のシャドウマップ法と比較すると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に成功しているものの、複数のオブジェクトが存在し、かつオブジェクト間の距離が離れているようなシーンでは、**図16**に示したようにシャドウマップに使用されていない無駄な領域が増えるという問題点がある。

そこで、2章で概観した **Stamminger[32]**らのパースペクティブシャドウマップ法(PSM)に提案手法を適用することで、より効率的なシャドウマップが生成できると考えた。次項では、パースペクティブシャドウマップ(PSM)について説明する。

### 3.6.2 パースペクティブシャドウマップ(PSM)

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、透視変換後の座標系でシャドウマップを生成している。例として**図 17**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。全シーンを透視変換することより、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、この座標系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。そのため、複数のオブジェクトが存在し、かつオブジェクト間の距離がかなり離れているようなシーンにおいても、視点から近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を多くできるため、効率的にシャドウマップを利用することができる。**図 18、19**に、通常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でレンダリングした結果をそれぞれ示す。通常のシャドウマップと比較すると、パースペクティブシャドウマップ法では、透視変換後のゆがんだ空間でシャドウマップを生成するため、視点に近い位置に存在するオブジェクトのシャドウマップに占める面積が大きく、シャドウマップを効率的に利用できていることが分かる。このため、視点からのレンダリング結果においても影の精度が改善されている。

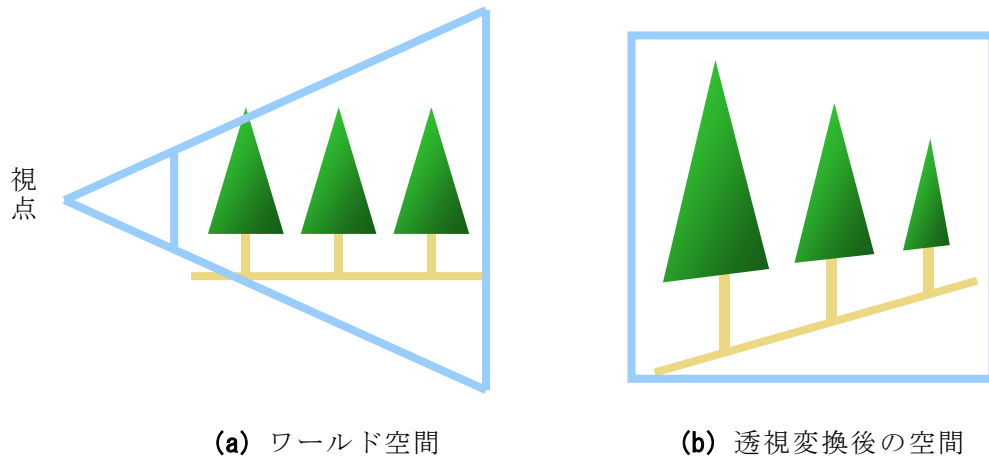


図 17 PSM の原理[32]

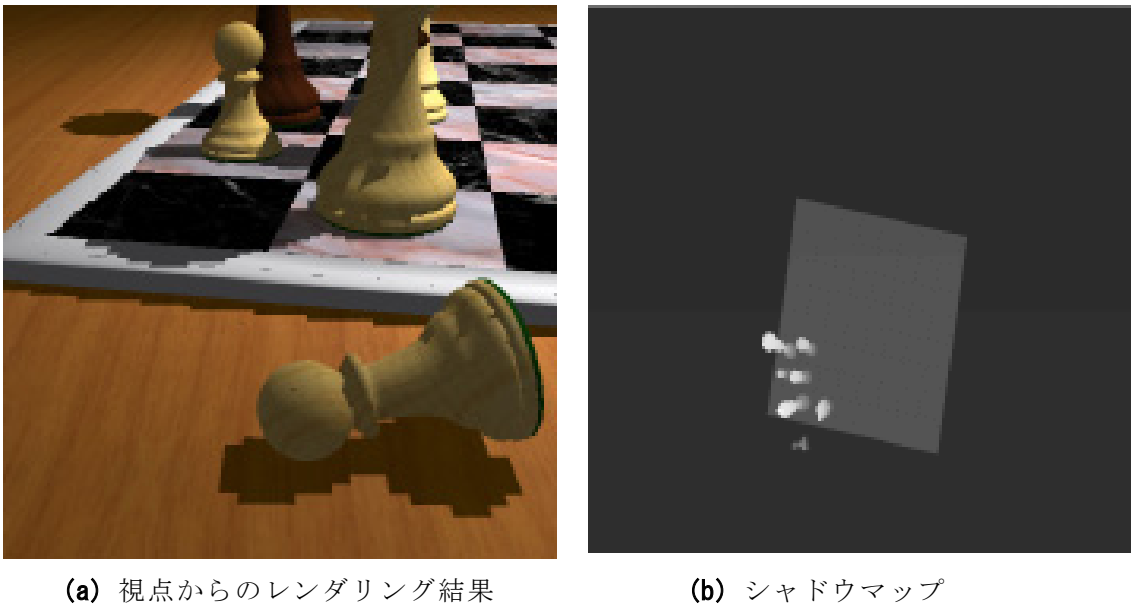
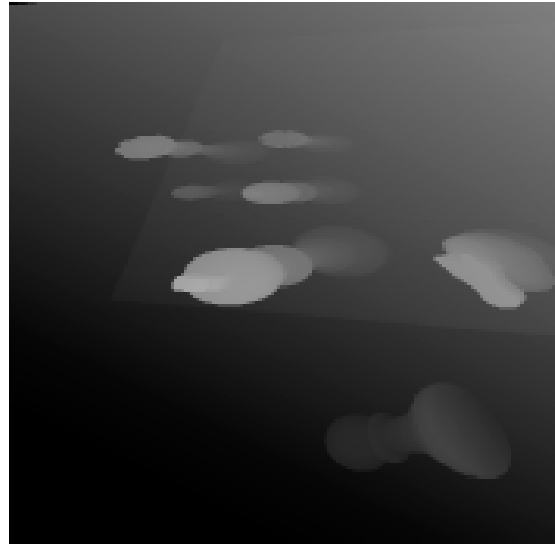


図 18 通常のシャドウマップ法によるレンダリング[32]



(a) 視点からのレンダリング結果



(b) シャドウマップ

図 19 パースペクティブシャドウマップ法によるレンダリング[32]

### 3.6.3 パースペクティブシャドウマップ法の枠組みでのシャドウマップの動的生成

提案手法は比較的容易に PSM と融合することが可能である。提案手法と PSM の融合手法の流れを図 20 に示す。はじめにオブジェクトの可視判定を行い、可視頂点を算出する。次に、すべての可視頂点を透視変換し、透視変換後の可視頂点を包含する境界球を生成する。最後に、光源側のビューボリュームが境界球を包含するように、光源側の射影行列を決定する。このように、従来はワールド座標系で生成していた境界球を、透視変換後の座標系で生成することで、提案手法と PSM を融合することができる。

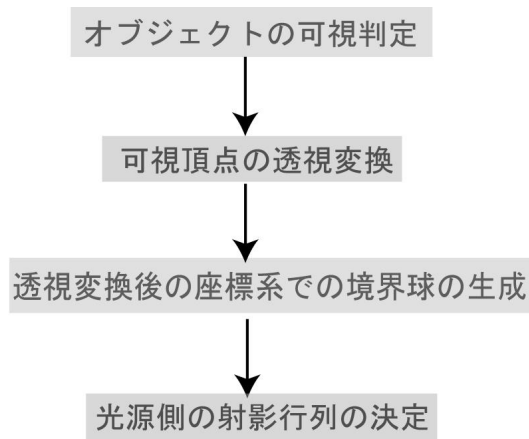


図 20 提案手法と PSM の融合手法の流れ

## 4. 実験結果

本章では、提案手法を通常のシャドウマップ法(SM)、およびパースペクティブシャドウマップ法(PSM)に適用してレンダリングした結果をそれぞれ示し、考察を行う。4.1節では、実験環境について述べ、4.2節ではSMとSM+提案手法のレンダリング実験結果を示し、提案手法を適用することの有効性を示す。さらに4.3節では、PSMに提案手法を適用した場合も含めて、4種類の手法による実験結果を比較し、レンダリング結果の画質について考察を行う。最後に、4.4節では、処理速度について考察する。

### 4.1 実験環境

本章では、提案手法の有効性を示すために、通常のシャドウマップ法(以下、SM)と2章で紹介したStamminger[32]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。実験環境を表2に示す。リアルタイムCGの描画には、3DライブラリDirectX9およびそれら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[36]。

表2 実験環境

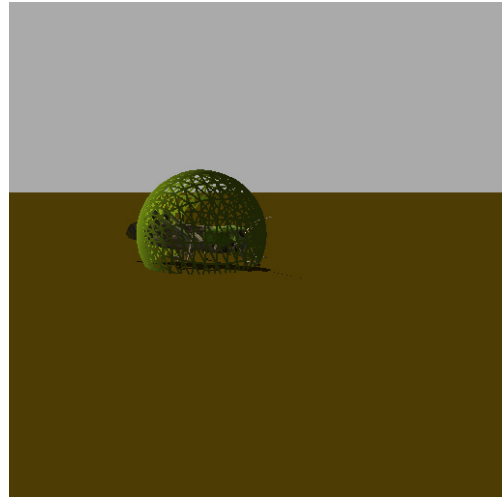
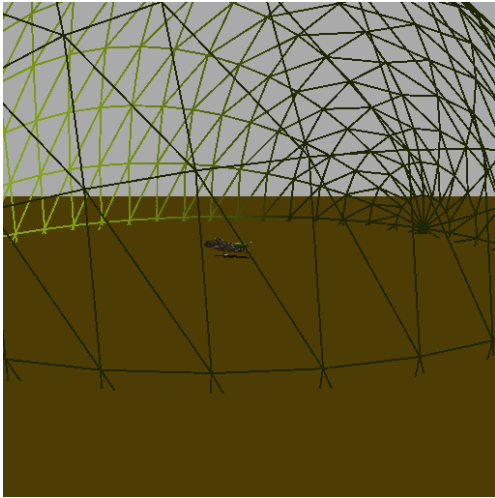
CPU	Pentium4 2.54GHz
メモリ容量	512MB
グラフィックスチップ	Nvidia Geforce FX 5800 Ultra
3D描画ライブラリ	Microsoft DirectX9

## 4.2 通常のシャドウマップ法と提案手法の比較

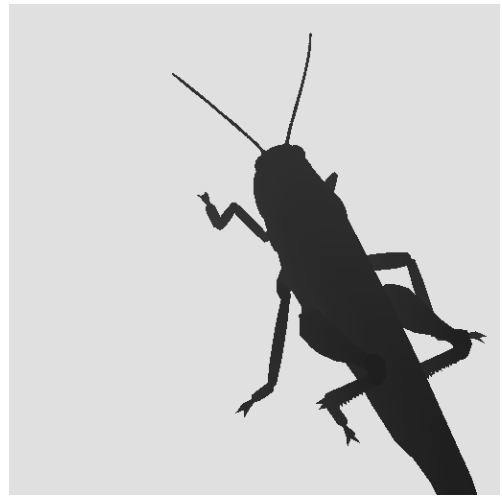
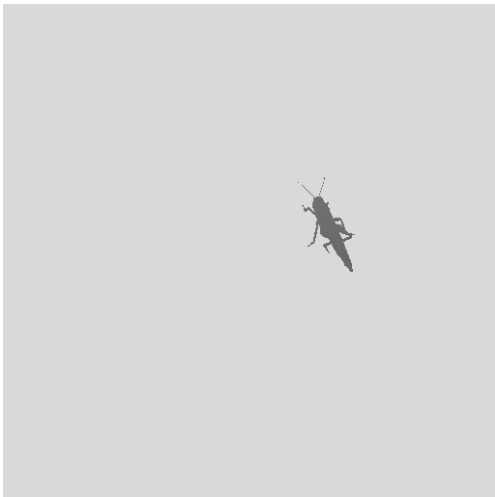
本節では、提案手法の有効性を示すために、通常のシャドウマップ法(以下、SM)と、SMに提案手法を適用したSM+提案手法のレンダリング結果を示し、考察を加える。SMおよびSM+提案手法でレンダリングした結果とその際に生成されたシャドウマップ、境界球をそれぞれ **図 21** に示す。同**図 (a) (b)** から分かるように、SMでは、経験的にすべてのオブジェクトが光源側のビューボリュームに含まれるようにシャドウマップの投影範囲が設定されているため、シーン中のオブジェクトに割り当てられるシャドウマップの解像度は低くなり、シャドウマップには利用されていない無駄な領域が多い。一方、SM+提案手法では、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の領域を効率良くシーン中のオブジェクトに割り当てることができている。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では表現しきれなかったバッタの触覚等も描画されていること等から、影の解像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、本来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設定した。

次に、シーン中のオブジェクトが自分自身に落とす影であるセルフシャドウが描画されるケースについて検討する。セルフシャドウでは、影を落とすのはオブジェクトの一部であり、シャドウマップに割り当てられる解像度は、オブジェクト全体と比較すると非常に低い。このため、正確にセルフシャドウを描画するには、高解像度のシャドウマップが必要である。セルフシャドウに視点を近づけたシーンを、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を**図 22, 23** にそれぞれ示す。**図 22** に示したSMのレンダリング結果は、オブジェクトの一部に割り当てられるシャドウマップの解像度が非常に低いため、視点からのレンダリング結果もエイリアシングが発生しており、影の精度が低い。一方、**図 23** に示したSM+提案手法のレンダリング結果は、SMと比較するとオブジェクトの一部に割り当てられるシャドウマップの解像度が高いため、影の精度が改善されていることが分かる。これらの実験から、提案手法をSMに適用することで、シーン中のオブジェクトを効率的にシャドウマップの投影対象とすることができ、影の精度を改善できることが確認できた。

設定された境界球



生成されたシャドウマップ



レンダリング結果



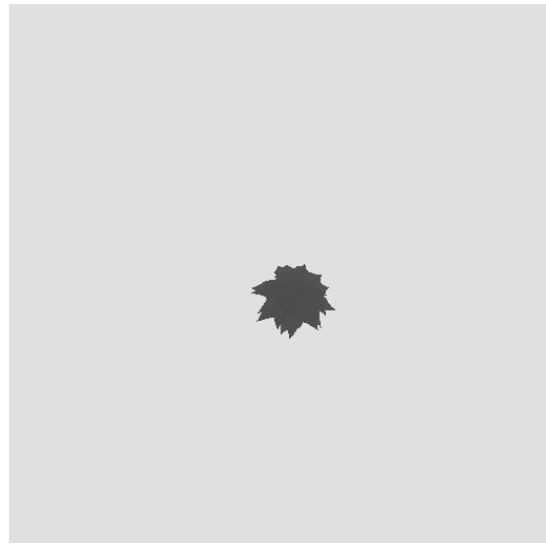
(a) SM

(b) SM+提案手法

図 21 SM と SM+提案手法によるレンダリング結果の比較



(a) レンダリング結果

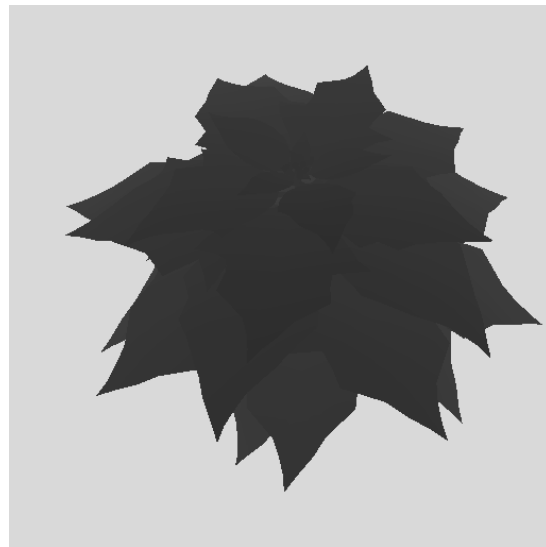


(b) 生成されたシャドウマップ

図 22 SM によるセルフシャドウのレンダリング



(a) レンダリング結果



(b) 生成されたシャドウマップ

図 23 SM+提案手法によるセルフシャドウのレンダリング

### 4.3 通常のシャドウマップ法および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た比較実験

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場合では、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。そこで、3.6節で述べたように、提案手法をパースペクティブシャドウマップ(PSM)に適用することを試みた。本節では、

- ・ SM
- ・ PSM
- ・ SM+提案手法
- ・ PSM+提案手法

の4種類のレンダリング手法についての比較実験を行い、個々の手法に考察を加える。

はじめに、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、SM+提案手法、PSM、PSM+提案手法の計4種類の方法でレンダリングを行った。レンダリング結果とその際に生成されたシャドウマップをそれぞれ図24~27に示す。SMと比較するとPSMの方が、影のエイリアシングは軽減されていることが図25(a)を見れば分かる。これは、視点から近い位置に存在するオブジェクトに対して多くのシャドウマップを割り当てているためである。しかし光源側のシャドウマップの投影範囲はシーン全体を包含するように決定するのが一般的であるため、シャドウマップには使用されていない無駄な領域が多い。一方、図27に示したPSMに提案手法を適用した(PSM+提案手法)では注視しているオブジェクトのみをシャドウマップの投影対象とすることで、視点から見える領域に応じて効率的なシャドウマップが生成されていることが分かる。これにより、視点からのレンダリング結果における影の精度が大幅に改善されていることが図27(a)を見れば分かる。

次に、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シーンを、上記と同じSM、SM+提案手法、PSM、PSM+提案手法の計4種類の方法でレンダリングを行った。レンダリング結果とその際に生成されたシャドウマ



ップをそれぞれ図 28～31 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の領域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 30(a) に示すように視点に近い位置に描画されているバッタの触覚の影は正確に描画できていない。一方、図 31 に示した PSM+提案手法では、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。

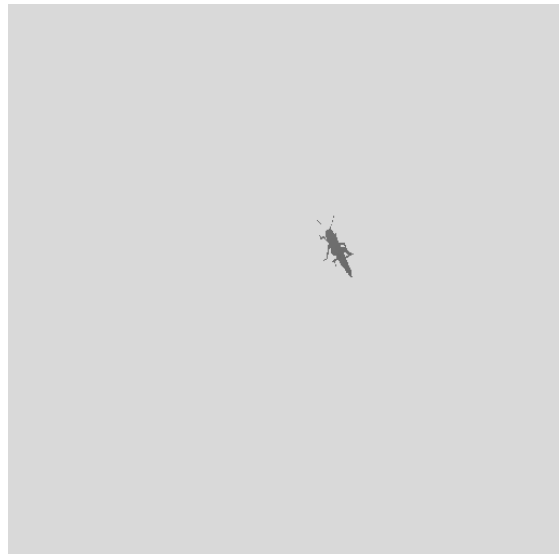
これにより、視点に近い位置に存在するオブジェクトの影の精度が高くなっており、SM+提案手法のみでは表現しきれなかったバッタの触覚が落とす影も描画できている。また、PSM(図 29)と PSM+提案手法(図 31)を比較すると、双方とも視点に近い位置に存在するオブジェクト(バッタ)の影の精度が高くなっているものの、図 29 に示した PSM のみのレンダリング結果では、視点から遠い位置に存在するオブジェクト(ここでは草花)のセルフシャドウが消えてしまっている。これは、PSM を適用すると視点から遠い位置に存在するオブジェクトに割り当てるシャドウマップの解像度は逆に減少するため、草花に割り当てるシャドウマップの解像度が不足したためである。一方、図 31 の PSM+提案手法のレンダリング結果では、図 29 に示した PSM と比較するとシャドウマップに無駄な領域が少ないため、セルフシャドウも消えていない。

さらに別のシーンを対象として、上記と同じ内容の実験を行ったレンダリング結果を合わせて Fig. 32～35 に示す。こちらも先ほどの実験結果とほぼ同じようなレンダリング結果であることが分かる。

これらの実験結果から、提案手法は SM と PSM どちらの手法に適用した場合でも、影の精度を高めることが可能であると言える。また、SM+提案手法と、PSM+提案手法に関しては、視点からの視野が広範囲を対象としている場合出は、PSM+提案手法が有効であると言える。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は逆に減少するため、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奥行き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。



(a) レンダリング結果

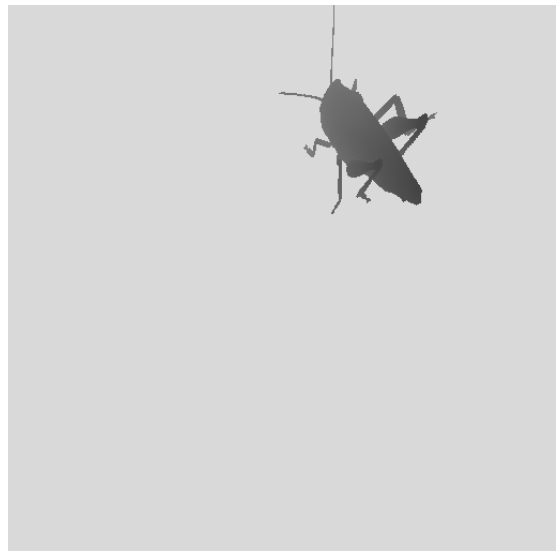


(b) 生成されたシャドウマップ

図 24 シーン 1 : SM



(a) レンダリング結果

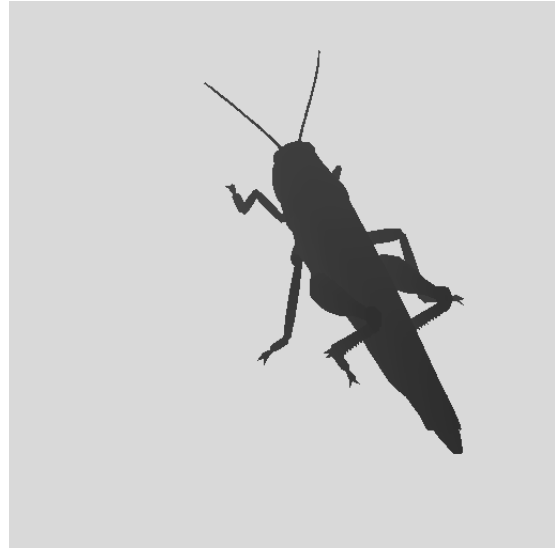


(b) 生成されたシャドウマップ

図 25 シーン 1 : PSM



(a) レンダリング結果

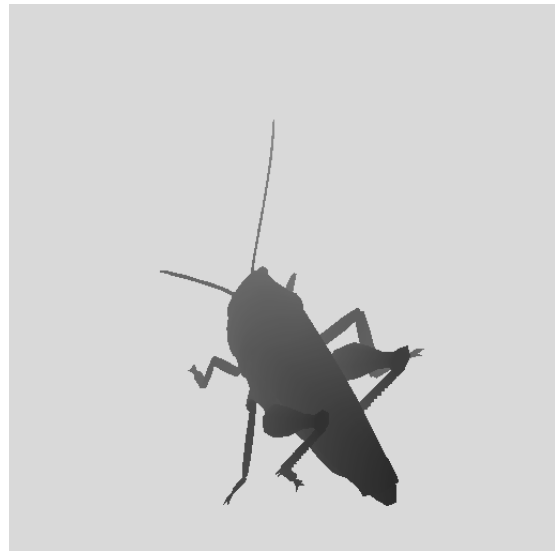


(b) 生成されたシャドウマップ

図 26 シーン 1 : SM+提案手法



(a) レンダリング結果

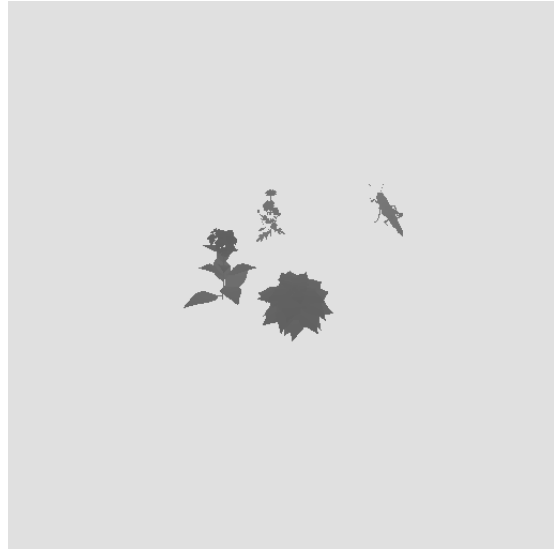


(b) 生成されたシャドウマップ

図 27 シーン 1 : PSM+提案手法



(a) レンダリング結果



(b) 生成されたシャドウマップ

図 28 シーン 2 : SM



(a) レンダリング結果

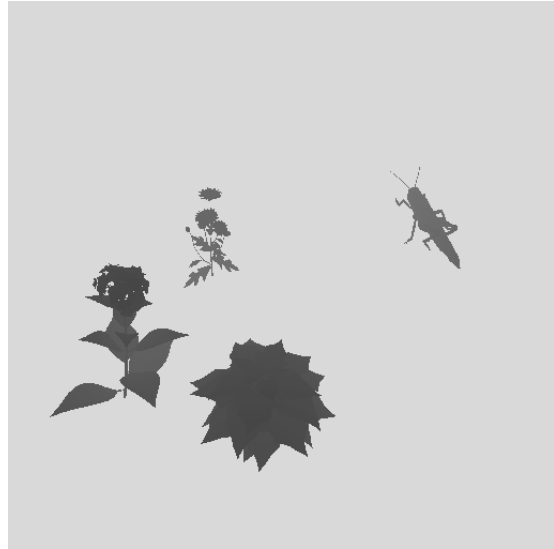


(b) 生成されたシャドウマップ

図 29 シーン 2 : PSM



(a) レンダリング結果

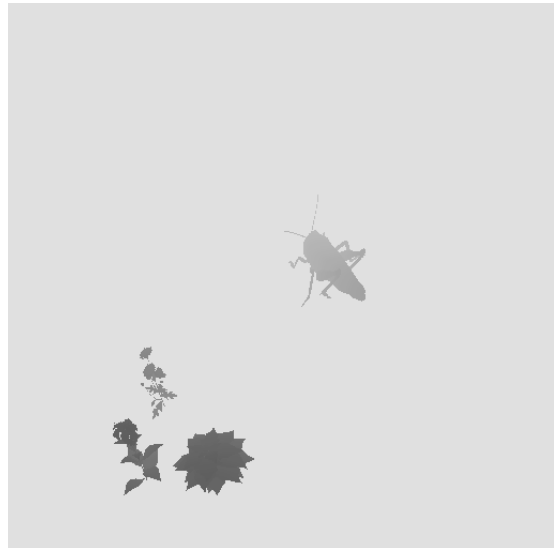


(b) 生成されたシャドウマップ

図 30 シーン 2 : SM+提案手法



(a) レンダリング結果



(b) 生成されたシャドウマップ

図 31 シーン 2 : PSM+提案手法



(a) レンダリング結果



(b) 生成されたシャドウマップ

図 32 シーン 3 : SM



(a) レンダリング結果

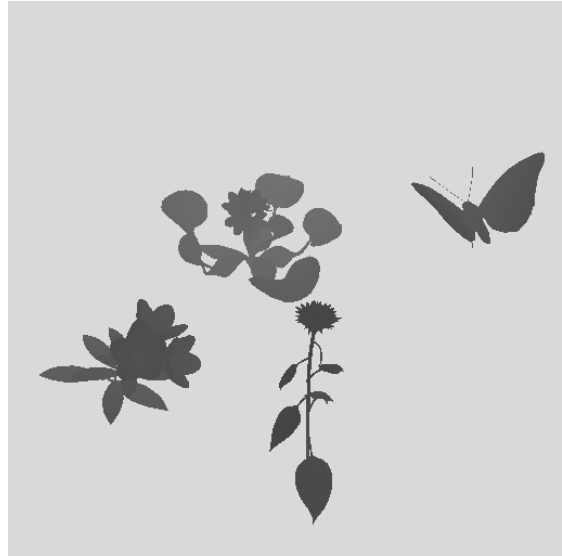


(b) 生成されたシャドウマップ

図 33 シーン 3 : PSM



(a) レンダリング結果



(b) 生成されたシャドウマップ

図 34 シーン 3 : SM+提案手法



(a) レンダリング結果



(b) 生成されたシャドウマップ

図 35 シーン 3 : PSM+提案手法

## 4.4 レンダリング速度の比較

本節では、4種類(SM、PSM、SM+提案手法、PSM+提案)の手法の処理速度について検討する。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 3 示す。さらに、SM と PSM に提案手法を適用した際のオーバーヘッドをそれぞれ図 36、37 に示した。ただし、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。というのも、シャドウマップにアクセスして影の判定を行う処理は、グラフィックスボード上の GPU が行っており(ハードウェアテクスチャマッピング)、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、およそ 10%前後の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 3 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
20,000	144fps	131fps	129fps	124fps
37,000	92fps	88fps	86fps	83fps
67,000	57fps	55fps	54fps	53fps
100,000	39fps	38fps	36fps	32fps



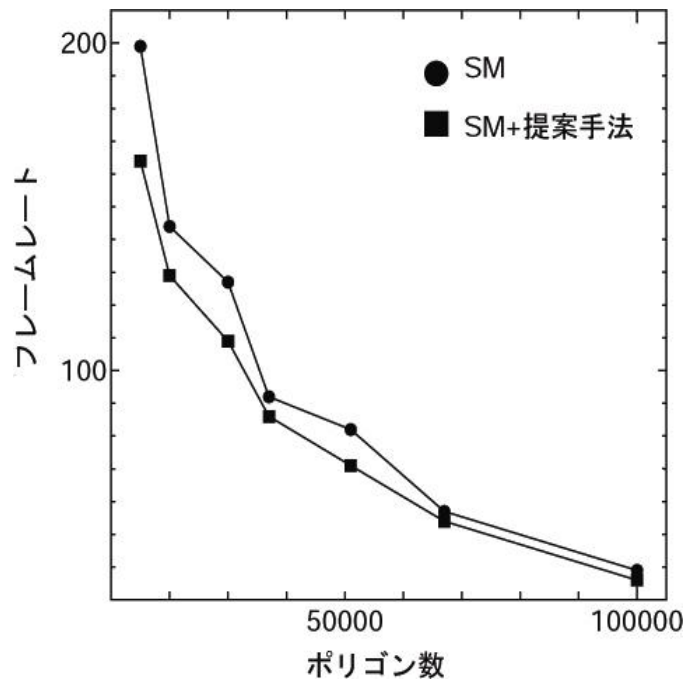


図 36 SM と SM+提案手法の処理速度の比較

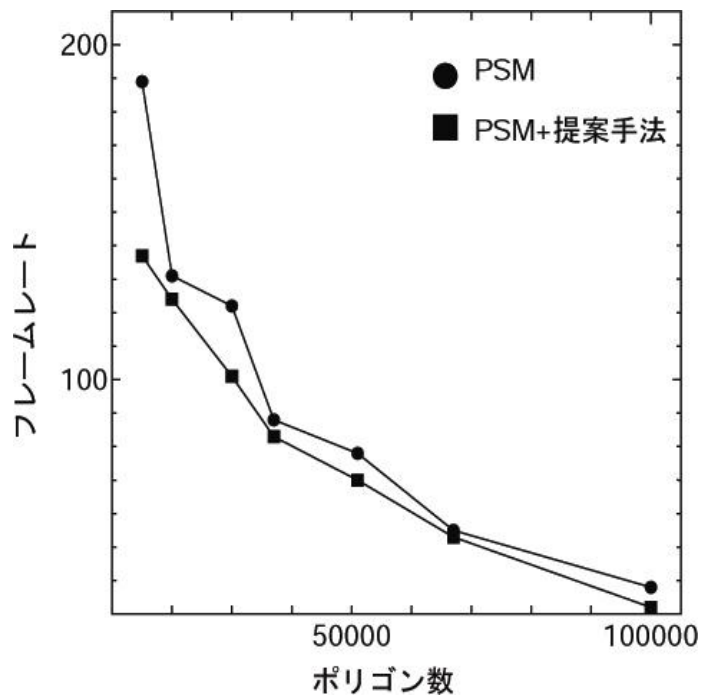


図 37 PSM と PSM+提案手法の処理速度の比較

## 5. まとめ

本論文では、従来のシャドウマップ法で視点が影に近づいた際に見られるエイリアシングを軽減することを目的として、視点とオブジェクトの位置関係を考慮してシャドウマップを動的に生成する手法を提案した。提案手法では、はじめに動的なシーン(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。次に、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マップを生成する。

実験では、提案手法を通常シャドウマップ法(SM)と、シャドウマップ法の改良手法であるパースペクティブシャドウマップ法(PSM)に適用し、レンダリング実験を行った。提案手法は、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き、シャドウマップを有効利用することがきるため、結果として双方ともエイリアシングを軽減することができた。ただし、PSMに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は逆に減少するため、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、SM+提案手法とPSM+提案手法を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奥行き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が考えられる。

処理速度に関しては、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フレームレートを計測した。その結果、提案手法を適用した場合でも、通常シャドウマップ法(SM)のリアルタイム性を失うことなく、ポリゴン数にもよるが、およそ10%前後のオーバーヘッドでレンダリング可能であることが確認できた。今後は、処理速度の向上についての検討を行う。

## 謝辞

本研究の全過程において、常に適切かつ有益なご指導を賜りました視覚情報メディア講座横矢直和教授に心から感謝致します。本研究を遂行するにあたり、有益な御助言、ご指導を賜りました像情報処理学講座千原國宏教授に謹んで御礼申し上げます。本研究を進めるにあたり、常に暖かい姿勢で適切なご指導を頂きました視覚情報メディア講座山澤一誠助教授に深謝致します。

そして、研究の方向付けから論文執筆に至るまで一貫して終始適切な御助言、御示唆を賜りました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。さらに、有益な御助言をいただきました視覚情報メディア講座佐藤智和助手に心から御礼申し上げます。また研究活動を行うにあたり、ご指導、御協力頂いた視覚情報メディア講座の学生の皆様、ならびに守屋知代様に厚く感謝致します。

## 参考文献

- [1] T. Moller and E. Haines : Real-time rendering, A.K.Peters Ltd, 1999.
- [2] D. Foley, V. Dam, K. Feiner and F. Hughes : コンピュータグラフィックス理論と実践, オーム社, 2001.
- [3] M. Deroula : Game programming gems, ボーンデジタル, 2001.
- [4] 神原 誠之, 横矢 直和 : 現実環境の照明条件と奥行きの実時間推定による仮想物体の陰影表現が可能な拡張現実感, 画像の認識・理解シンポジウム講演論文集, pp.247-252, 2004.
- [5] T. Naemura, T. Nitta, A. Mimura and H. Harashima : Virtual shadows in mixed reality environment using flashlight-like devices, Trans. Virtual Reality Society of Japan, pp.227-237, 2002.
- [6] 苗村 健, 金子 正秀, 原島 博 : 光線記述に基づく仮想光源画像の合成, 映像学誌, pp.1328-1335, 1998.
- [7] M. Hasenfratz, M. Lapierre, N. Holzschuch and F.X. Sillion : A survey of real-time soft shadows algorithms, Computer Graphics Forum ,Vol.22, No.4, pp.753-774, 2003.
- [8] A. Woo and P. Poulin : A. Fournier : A survey of shadow algorithms, IEEE Computer Graphics and Applications, Vol.10, No.6, pp.13-32, 1990.
- [9] L. Williams : Casting curved shadows on curved surfaces, Proc. SIGGRAPH'78, pp. 270-274, 1978.
- [10] T.Möller and U.Assarsson : Approximate soft shadows on arbitrary surfaces using penumbra wedges, Proc. Rendering Techniques2002, pp.297-306, 2002.
- [11] U. Assarsson and T. Möller : Geometry-based soft shadow volume algorithm using graphics hardware, Proc. SIGGRAPH 2003, pp.511-520, 2003.
- [12] S. Brabec and P. Seidel : Single sample soft shadows using depth maps,. Proc. Graphics Interface, pp.219-228, 2002.
- [13] E. Chan and F. Durand : Rendering fake soft shadows with smoothies. Proc. Rendering Techniques2003, pp.25-27, 2003.
- [14] THE BURROW Director 8.5 Site <http://www.theburrow.co.uk/d85/>

- [15] Planar Projected Shadow における変換行列の導出  
<http://cgi3.tky.3web.ne.jp/~tkano/columns/20000716.shtml>
- [16] 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)
- [17] F. Crow : Shadow algorithms for computer graphics, Proc. SIGGRAPH'77, pp242-248, 1977.
- [18] S. Brabec and P. Seidel : Shadow volumes on programmable graphics hardware, Eurographics Symposium on Rendering, Vol.25, No.3, 2003.
- [19] D. McCool : Shadow volume reconstruction from depth maps, ACM Transactions on Graphics, Vol.19, No.1, pp.1-26, 2000.
- [20] S. Roettger, A. Irion and T. Ertl : Shadow volumes revisited, Winter School on Computer Graphics, pp.373-379, 2002.
- [21] 荒屋 真二 : 明解3次元コンピュータグラフィックス, 共立出版, 2003.
- [22] M. Segal, C. Korobkin, V. Widenfelt, J. Foran and P. Haeberli : Fast shadow and lighting effects using texture mapping, Proc. SIGGRAPH'92, pp.249-252, 1992.
- [23] C. Everitt, A. Rege, C. Cebenoyan : Hardware shadow mapping, Nvidia technical report , 2001.  
[http://developer.nvidia.com/object/hwshadowmap\\_paper.html](http://developer.nvidia.com/object/hwshadowmap_paper.html)
- [24] J. Kilgard : Shadow mapping with today's OpenGL hardware, GDC2000 programming session, 2000.  
[http://developer.nvidia.com/object/ogl\\_shadowmap.html](http://developer.nvidia.com/object/ogl_shadowmap.html)
- [25] A. Watt : 3D computer graphics, pp.274, 1999.
- [26] T. Reeves, D. Salesin and R. Cook : Rendering Antialiased shadows with depth maps, Proc. SIGGRAPH'87, pp.283-291, 1987.
- [27] H. Zhang : Forward shadow mapping, Proc. Rendering Techniques'98, pp.131-138, 1998.
- [28] T. Lokovic and E. Veach : Deep shadow maps, Proc. SIGGRAPH2000, pp.385-392, 2000.
- [29] C. Dachsbacher and M. Stamminger : Translucent shadow maps, Proc. Eurogr

- aphics Workshop on Rendering2003, pp.197-201, 2003
- [30] R. Fernando, S. Fernandez, K. Bala and D. P. Greenberg : Adaptive shadow maps, Proc. SIGGRAPH 2001, pp.387-390, 2001.
- [31] P. Sen, M. Cammarano and P. Hanrahan : Shadow silhouette maps, Proc. SIGGRAPH 2003, pp.521-526, 2003.
- [32] M. Stamminger and G. Drettakis : Perspective shadow maps, Proc. SIGGRAPH2002, pp.557-562, 2002.
- [33] 岩尾 友秀, 神原 誠之, 横矢 直和 : 視点と光源の位置関係を考慮したシャドウマップの動的生成手法, 情報処理学会関西支部支部大会講演論文集, pp.45-48, 2004.
- [34] 岩尾 友秀, 神原 誠之, 横矢 直和 : 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, 電子情報通信学会 技術研究報告, 2005
- [35] 森 眞一郎, 五島 正裕, 中島 康彦, 富田 眞治 : 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, pp.179-182, 2004.
- [36] 今黎 隆 : DirectX9シェーダプログラミングブック, 毎日コミュニケーションズ, 2004.