

適応的前計算遮蔽データを用いた 動的シーンのリアルタイムレンダリング

田村 尚希[†] ヘンリー・ジョハン^{†,††} 西田 友是[†]

近年、点光源、面光源などのローカル光源や画像で表現された照明環境（環境光源）で照らされた動的なシーンを高速にレンダリングできる Precomputed Shadow Fields 法 (PSF 法) が提案された。PSF 法では、物体周囲の空間中のサンプル点において予め遮蔽情報を保存し、実行時に物体上の各頂点において、前計算した遮蔽情報を統合することで動的なシーンを高速にレンダリングできた。しかし、既存の手法では、遮蔽情報を格納するサンプリング点数が全ての同心球において均一であり必要以上の情報を保持していた。また、グラフィクスハードウェアを活用した高速なレンダリングには対応していなかった。そこで本研究では、画質を損なうことなくサンプリング数を削減する手法を提案する。本手法では、隣接サンプル点間における遮蔽情報の変化が比較的緩やかであることに着目しサンプリング数を削減する。またグラフィクスハードウェアを用いて輝度分布の変化が滑らかな低周波照明下のレンダリングを高速化する手法を提案する。

Real-Time Rendering of Dynamic Scenes Using Adaptive Shadow Fields

NAOKI TAMURA,[†] HENRY JOHAN^{†,††} and TOMOYUKI NISHITA[†]

Recently, the Precomputed Shadow Fields method, which achieves fast rendering of dynamic scenes under environment illumination and local light sources, was proposed. This method can render shadows fast by precomputing the occlusion information at many sample points arranged on concentric shells around each object and combining multiple precomputed occlusion information rapidly in the rendering step. However, this method employs the same sampling rates on all shells and does not utilize graphics hardware for rendering. In this paper, we propose a method for reducing the amount of sample points without degrading the image quality. We reduce the number of sample points adaptively by considering the difference of the occlusion information between adjacent sample points. Additionally, we propose a method for fast rendering under low-frequency illuminations using shadow fields on graphics hardware.

1. はじめに

写実的な画像を生成する技術は、コンピュータグラフィックスの研究の中で最も重要なテーマの一つである。一般に写実的な画像の生成には、ライティングが重要な役割を占める。これまで平行光線、点光源、面光源のようなローカル光源をシーン中に複数配置することでライティング環境を設定し写実的な画像を計算する例が多かった。近年、シーンを覆うドーム状の光源環境（環境光源）を画像によって入力し、写実的な画像を生成する研究が数多くされている。環境光源を用いたレ

ンダリングは、一般にレイトレーシング法がよく用いられる。しかし、レイトレーシング法を用いた場合には、多くの計算時間を必要としていた。

Zhouらは、前計算放射輝度伝達法¹⁾を拡張し、ローカル光源と環境光源が並存する照明下の動的なシーンをレンダリングできる Precomputed Shadow Fields 法²⁾(PSF 法)を提案した。PSF 法では、移動・回転したい物体毎に、物体周囲の空間中のサンプル点で遮蔽情報を格納したデータ構造 (*shadow fields*) を前計算する。PSF 法の *shadow fields* は物体を中心とした同心球上にサンプル点を配置している。実行時に各 *shadow fields* に格納された遮蔽情報を高速に組み合わせることで、任意の場所での遮蔽情報を高速に構築し輝度を計算する。しかし、PSF 法では、遮蔽情報を格納するサンプリング点数が全ての同心球において均

[†] 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

^{††} 現在は Nanyang Technological University, Singapore

一であり必要以上の情報を保持していた。また、グラフィックスハードウェア (GPU) を活用した高速なレンダリングには対応していなかった。

本研究の目的は、前述した PSF 法の欠点を解決し、ヴァーチャルリアリティやゲームといった一般のアプリケーションにおいて、より汎用的に利用できるアルゴリズムを提案することである。提案法では、shadow fields 内のデータが近傍のサンプル点間において緩やかに変化することに着目してサンプル点を削減する。また、レンダリング速度は、GPU 上でレンダリング計算の大半を処理することにより改善する。

本論文では、シーンは三角形メッシュで構成されており拡散面または光沢面を有しているとする。また、一般に急激な輝度変化を含む全周波近似を利用するレンダリング³⁾⁴⁾は、メモリ消費量・計算速度ともに一般のアプリケーションにおいて現実的ではないため、GPU による高速化は低周波近似のみを対象とする。

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 ソフトシャドウの描画

西田らは、線光源、面光源に照らされてできるソフトシャドウをレンダリングする手法⁵⁾を提案した。また、環境光源に性質の良く似た、ドーム状の天空光によってできるソフトシャドウを計算する手法⁶⁾を提案した。しかし、彼らの手法では、ソフトシャドウを生成するために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、GPU を利用して高速にソフトシャドウを計算する手法が数多く提案されている。GPU を利用してソフトシャドウを計算する手法は、シャドウマップ法をベースにした手法⁷⁾⁸⁾とシャドウボリューム法をベースにした手法⁹⁾¹⁰⁾に大別できる。

これらの手法では環境光源によるソフトシャドウは考慮されていないか、適用できたとしてもリアルタイムにソフトシャドウを生成することは難しい。

2.2 前計算放射輝度伝達法

Sloan らは環境光源下の半影、直接光、間接光、集光効果をリアルタイムにレンダリングできる前計算放射輝度伝達法¹⁾を提案した。その後 Sloan らの前計算放射輝度伝達法を改良して、任意の BRDF に対応した手

法¹¹⁾、主成分分析を用いて前計算データを圧縮した手法¹²⁾、wavelet 変換を用いて全周波情報まで考慮した手法³⁾⁴⁾などが提案された。しかし、上記の手法では前計算した遮蔽情報を変更できないため、シーン中の物体を移動・回転できない。James らは、複数のシーン状態に対して前計算輝度伝達法を適用し、それらを補間することで物体を移動、回転、変形できる手法¹³⁾を提案した。しかし彼らの手法では、前計算した範囲内の限定的な空間でのみしか物体を変化できない。Mei らは、SRTM (Spherical Radiance Transport Maps) を用いて、物体を自由に移動・回転できる手法¹⁴⁾を提案した。Kautz らは、環境光源下のシーンにおいて、球面ラスタライズ計算を毎フレーム、全頂点行うことにより物体形状を自由に変形できる手法¹⁵⁾を提案した。しかしこれらの手法では、複雑で大規模なシーンにおいて計算量が膨大になり、高速に描画することは難しい。また、ローカル光源との併用も難しい。

Zhou らは、物体周囲の多数の点で遮蔽情報を格納した shadow fields を前計算し、実行時に shadow fields に保存されているデータを組み合わせることで動的なシーンをレンダリングできる PSF 法²⁾を提案した。しかし彼らの手法では、shadow fields のサンプル点が全ての同心球で均一であり、レンダリング速度もそれほど高速ではない。上記の手法をスケーラビリティ、メモリ消費量、レンダリング速度の三つの観点から比較すると PSF 法が最も優れている。

3. Precomputed Shadow Fields 法

本節では、PSF 法を概略するとともに、既存アルゴリズムの問題点とその原因について述べる。

3.1 アルゴリズムの概要

PSF 法では、移動・回転したい物体、ローカル光源毎に shadow fields を前計算する。shadow fields の計算には、まず物体を中心とした同心球を考える。次に各球上にサンプル点を取り、各サンプル点からみた対象物体による遮蔽の情報及び対象物体から到達する輝度の情報を取得し直交変換を施す。この時、直交変換に球面調和関数展開 (SH)¹⁾を用いて低周波近似するか、もしくは、wavelet 変換³⁾を用いて全周波近似するかによって影の質やレンダリング速度・メモリ消費量が異なる。本論文では、遮蔽情報に対して直交変換を施して得られた基底関数の係数ベクトルを遮蔽係数ベクトル (OCV) と呼ぶ。同様に物体から到達する輝度情報に関しては輝度係数ベクトル (RCV) と呼ぶ。また、各頂点における自己遮蔽も同様に前計算する。

shadow fields を利用したレンダリングでは、まず物

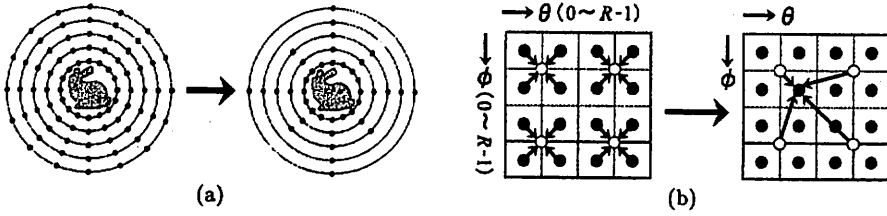


図 1 (a) 適応的サンプリングの概念図. 左: 既存手法²⁾(均一サンプリング), 右: 提案法 (適応的サンプリング). (b) サンプリング数最適化アルゴリズムの概念図. 図の θ, ϕ は各同心球を極座標表現した時の緯度, 経度方向を表す.

体の各頂点において輝度を計算する. その後メッシュの各三角形内で, 頂点の輝度を補間しながらシーンを描画する. 各頂点の座標 p における輝度は, 自己遮蔽と他の物体による OCV を統合したベクトルと, 環境光源を直交変換したベクトルおよび他の物体による RCV との内積により求める. 他の物体による OCV は, まず各物体による OCV を計算し, 次にそれらを統合することで求められる. ある物体 O による OCV(RCV) は, p を O のローカル空間の座標 p' に変換し, O の shadow fields の内 p' の近傍点の OCV(RCV) を線形補間して計算される. また, 各物体の OCV は triple product(三重積)⁴⁾ を反復することで統合できる.

3.2 問題点と原因

PSF 法には以下の二点の問題がある.

- 全同心球において均一なサンプリング数. C 個の同心球上に $R \times R$ 個のサンプル点 (極座標の ϕ, θ 方向にそれぞれサンプリング解像度 R) を配置し, 各サンプル点において D byte の係数を E 個を保持すると仮定すると, 均一サンプリングでは, 各 shadow fields 毎に $C \times R^2 \times E \times D$ byte のメモリ容量を必要とする.
 - CPU を用いた比較的低いレンダリング速度
- 4, 5 節において上記問題の解決手法を提案する.

4. Shadow Fields のサンプリング最適化

本節では, shadow fields のサンプリング数を最適化する手法について述べる. 一般に shadow fields 内の情報は, 近傍のサンプル点間において緩やかに変化する. 提案法では, 図 1(a) のように, 均一にサンプリングされた shadow fields から不必要なサンプル点を除去しサンプリング数を最適化する. また, GPU 実装の効率化を考えて, サンプリング数を常に 2 の乗数に保つ. c 番目の同心球の最適化されたサンプリング解像度 R_c は以下のように各同心球毎独立に計算する (図 1(b) 参照).

- (1) 対象同心球にサンプリング解像度 R (サンプリング点数 $R \times R$) で初期サンプル点を配置する (黒点).
- (2) 全サンプル点で遮蔽情報を計算し, OCV O に変換

する (低周波近似では球面調和関数, 全周波近似では wavelet 変換を用いる).

- (3) 対象同心球にサンプリング解像度 $R/2$ で新サンプル点を配置する (白点).
- (4) 全新サンプル点において近傍 4 個の初期サンプル点の O を線形補間して OCV \tilde{O}_{new} を計算する.
- (5) 全初期サンプル点において近傍 4 個の新サンプル点の \tilde{O}_{new} を線形補間して OCV \tilde{O}_{init} を計算する (ただし, 四隅には最近傍 1 個のサンプル点を, 境界には近傍 2 個のサンプル点を用いる).
- (6) 全初期サンプル点において OCV \tilde{O}_{init} と OCV O を逆変換し, 逆変換後の情報間の差を測定する.
- (7) 全初期サンプル点において, 差が設定しきい値以下ならば, 初期サンプル点を新サンプル点に置き換え, R の値を半分にする. さらに手順 3 に戻り再帰的に処理を繰り返す. しきい値を最初に超えた R の値を R_c とする. なお, 提案法で用いるしきい値は 6.1 節で実験を踏まえて述べる.

上記では特に遮蔽情報の場合を記述しているが, 輝度情報の場合も OCV を RCV に置き換え同様にを行う.

5. GPU による高速化

本節では, GPU を用いてレンダリング速度を向上させる手法について述べる. 提案法の GPU 実装では, 低周波近似のみを対象とし, shadow fields の各サンプル点の OCV, RCV は 4 次の球面調和関数 (16 基底) を用いて表現され, さらに各係数が 8bit に量子化されていると仮定する. 図 2 は shadow fields を用いた輝度計算の概要である. 図中の輝度計算処理の内頂点 v 毎の計算は, 互いに独立であり, 並列処理が可能である. 提案法ではこれらの計算に並列処理能力に特化した GPU を利用することで, CPU で計算するよりも高速に計算を行える. 図中の下線は実際に GPU 実装した箇所を示している.

提案法では, まず, 物体毎に $N \times N$ サイズ (ただし $N^2 >$ 物体の頂点数) の輝度テクスチャ T_B , 頂点配列テクスチャ T_L を用意する. その後 CPU でカリング (不可視頂点の除去) を行い可視頂点の配列 L を計算し, L を T_L に格納する. 次に, L の k 番目の頂点 v_k を T_B の $(x, y : k \bmod N, k/N)$ のピクセルと 1 対 1 の対応関係を作る. この操作により, 頂点毎の独立し

た計算がピクセル毎の計算に置き換わり、輝度計算の大半を GPU に任せることができる。図 2 に従って輝度の計算を行い、最終的に T_B には L 内の頂点での輝度が格納される。シーン描画時には vertex shader 内で頂点毎に T_B のうちでその頂点と対応するピクセルを参照し、取得した値を頂点色としてメッシュを描画する。

図 2 の下線部分は大きく分けて以下の 4 つの項目から成り立っている

- (1) 任意の頂点 p での適応的にサンプリングされた shadow fields からの OCV (RCV) の復元。
- (2) ローカル座標系からグローバル座標系へ座標軸を変換したときの OCV (RCV) の再計算 (SH Rotation)。
- (3) 三重積を用いた 2 つの OCV の統合。
- (4) 係数ベクトル同士の内積。

p における OCV の復元は、まず p を遮蔽物体のローカル座標 p' に変換する。その後 p' の最近の 2 つの同心球を計算し、各同心球において OCV を復元する。その後距離方向で距離に応じた補間を行う。なお、shadow fields のデータは予めテクスチャに格納しておく。SH Rotation は、基本的な ZXZXZ Rotation¹¹⁾ を fragment shader 内で行う。この時物体の回転量は予め CPU で計算し fragment shader の定数として設定する。三重積は、Ng らの手法⁴⁾、PSF 法²⁾ に示されているように、4 次の球面調和関数の非 0 の組み合わせは比較的少ない。提案法では、非 0 の組み合わせを総当りで fragment shader 上で計算する。係数ベクトル同士の内積は、16 係数のうち 4 係数毎に fragment shader の dot 命令 (4 次元ベクトルの内積計算命令) を用い、さらにそれらの和を計算し求めている。

6. 結果

本節では、提案法による結果を示す。結果に用いた shadow fields は、低周波近似、全周波近似共に 32 同心球 ($C = 32$) で構成され、初期サンプル点は、各同心球とも 64×64 ($R = 64$) である。また、各サンプル点の遮蔽情報は、低周波近似では 4 次の球面調和関数 (16 係数)、全周波近似では、wavelet 変換後の絶対値の大きな上位 5% の係数を用いて近似されている。両近似とも係数は 8bit に量子化されている。同心球の中心は、各物体の中心に設定され、 c 番目 ($c = 0, \dots, 31$) の同心球の半径は $0.2V_c(1+c)$ (V_c は物体のバウンディングスフィアの半径) である。計算には、CPU: Pentium D 3.0GHz, GPU: GeForce 7800 GTX を用いた。

6.1 しきい値の決定

Shadow fields のサンプリング数の最適化に用いる適切なしきい値を決定するため、以下のような実験を行った。まず、球に近い物体 (ティーポット)、細長い物

```

// TB : the exitant radiance texture //
// Ov : the self-occlusion information at vertex v //
//  $\hat{p}$  : the product of the BRDF and a cosine term //
rotate distant lighting Sd to align with global coordinate frame
For each entity I that is an object do
  L = visible vertices of I that are visible from camera
  compute distance from center of I to each scene entity
  sort entities in order of increasing distance
  For each visible vertex v in L do
    TB(v) = 0
    Uv = TripleProduct(Ov,  $\hat{p}$ )
    rotate Uv to align with global coordinate frame
    For each entity J do
      If J is a light source
        calculate RCV Sj(v)
        rotate Sj(v) to align with global coordinate frame
        Bv+ = DoubleProduct(Sj(v), Uv)
      Else
        calculate OCV Oj(v)
        rotate Oj(v) to align with global coordinate frame
        Uv = TripleProduct(Oj(v), Uv)
    End If
  End For
  TB(v)+ = DoubleProduct(Sd, Uv)
End For
End For

```

図 2 レンダリング処理の疑似コード。下線部は GPU 処理をしている箇所を示す。

体 (像)、平面に近い物体 (板) の 3 種類の異なるタイプの物体からなるシーンを用意し、次にしきい値を機々に変化させて画質の変化を測定した。図 3 は実験結果である。図の差分画像より、均一サンプリングの場合としきい値を 0.020 に設定した場合には視覚的に差異が観察できる。このような実験を繰り返し、低周波近似、全周波近似ともにしきい値を 0.010 に設定すれば、視覚的に十分な画質を得られることが分かった。これ以降の結果では、しきい値を 0.010 に設定している。

6.2 サンプリング数の削減結果

前節の 3 種類の物体の最適化されたサンプリング解像度 R_c を、表 1 に示した。表中の各物体の上の段が低周波近似の場合 (L)、下の段が全周波近似の場合 (A) の結果を示している。表 2 は提案法適用前後のメモリ消費量を示している。適応的サンプリングにより、低周波近似の場合で 60% ~ 70%、全周波近似の場合で 40% ~ 45% のメモリ消費量の削減が実現できる。低周波近似における均一サンプリングの shadow fields では、32 同心球上の 4096 サンプル点毎に 16byte 消費するため、 $32 \times 4096 \times 16 = 2\text{MB}$ 必要とする。

6.3 GPU による高速化

図 4, 5 は、提案法における低周波近似の場合のレンダリング結果である。図 4 のシーンは 15 物体、41,340 頂点、図 5 のシーンは 20 物体、88,739 頂点である。表 3 は提案法における shadow fields のメモリ消費量及びレンダリング速度である。提案法による GPU を用いた高速化の効果は非常に顕著に現れている。

表 1 提案法適用後の各同心球のサンプリング解像度 R_c .

Teapot (L)	64 64 64 64 64 64 64 64 32 32 32 16 16 16 16 16 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
Teapot (A)	64 64 64 64 64 64 64 64 64 64 64 64 32 32 32 16 16 16 8 8 8 8 8 8 8 8 8 8 8 8 8
Statue (L)	64 64 64 64 64 64 64 64 32 16 16 16 16 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
Statue (A)	64 64 64 64 64 64 64 64 64 32 32 16 16 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
Plane (L)	64 64 64 64 64 64 64 64 64 32 16 16 16 16 16 16 8 8 8 8 8 8 8 8 8 8 8 8 8 8
Plane (A)	64 64 64 64 64 64 64 64 64 64 64 32 32 32 16 16 8 8 8 8 8 8 8 8 8 8 8 8 8 8

表 2 提案法適用前後のメモリ消費量の比較.

	均一サンプリング (MB)	適応的サンプリング (MB)
Teapot (L)	2.0	0.59
Teapot (A)	21.9	13.0
Statue (L)	2.0	0.55
Statue (A)	21.3	11.7
Plane (L)	2.0	0.75
Plane (A)	19.5	11.9

表 3 提案法における shadow fields のメモリ消費量とレンダリング速度.

	Sizes of SF (MB)	CPU (fps)	GPU (fps)
ボーリングシーン	4.2	1 ~ 3	30 ~ 40
物体落下シーン	12.9	0.5 ~ 1	12 ~ 16

7. まとめと今後の課題

本論文では、適応的にサンプル点を削減した shadow fields を用いて、ローカル光源と環境光源が並存する照明下の動的なシーンを高速にレンダリングできる手法を提案した。提案法では、shadow fields 内のデータが近傍のサンプル点間において緩やかに変化することに着目してサンプル点を削減した。また、GPU を用いて低周波近似をした場合のレンダリングを高速化する手法を提案した。よって提案法は、PSF 法の持つ問題を克服できた。

今後の課題として、全周波近似をした場合のレンダリング速度の向上や、同心球の数や間隔の最適化が考えられる。

参考文献

- 1) P. P. Sloan, J. Kautz, and J. Snyder. Pre-computed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proc. SIGGRAPH 2002*, pages 527-536, 2002.
- 2) K. Zhou, Y. Hu, S. Lin, B. Guo, and H. Y. Shum.

Precomputed shadow fields for dynamic scenes. *ACM Transactions on Graphics*, 24(3):1196-1201, 2005.

- 3) R. Ng, R. Ramamoorthi, and P. Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics*, 22(3):376-381, 2003.
- 4) R. Ng, R. Ramamoorthi, and P. Hanrahan. Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics*, 23(3):477-487, 2004.
- 5) T. Nishita, I. Okamura, and E. Nakamae. Shading models for point and linear sources. *ACM Transactions on Graphics*, 4(2):124-146, 1985.
- 6) T. Nishita and E. Nakamae. Continuous tone representation of three-dimensional objects illuminated by sky light. In *Proc. SIGGRAPH 86*, pages 125-132, 1986.
- 7) W. Heidrich, S. Brabec, and H. P. Seidel. Soft shadow maps for linear lights. In *Proc. 11th Eurographics Workshop on Rendering*, pages 269-280, 2000.
- 8) C. Soler and F. X. Sillion. Fast calculation of soft shadow textures using convolution. In *Proc. SIGGRAPH 98*, pages 321-332, 1998.
- 9) T. Akenine-Moller and U. Assarsson. Shading and shadows: Approximate soft shadows on arbitrary surfaces using penumbra wedges. In *Proc. 13th Eurographics Workshop on Rendering*, pages 297-306, 2002.
- 10) U. Assarsson and T. Akenine-Moller. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Transactions on Graphics*, 22(3):511-520, 2003.
- 11) J. Kautz, P. P. Sloan, and J. Snyder. Shading and shadows: Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. In *Proc. 13th Eurographics Workshop on Rendering*, pages 291-296, 2002.
- 12) P. P. Sloan, J. Hall, J. Hart, and J. Snyder. Clustered principal components for pre-computed radiance transfer. *ACM Transactions on Graphics*, 22(3):382-391, 2003.
- 13) D. L. James and K. Fatahalian. Precomputing interactive dynamic deformable scenes. *ACM Transactions on Graphics*, 22(3):879-887, 2003.
- 14) C. Mei, J. Shi, and F. Wu. Rendering with spherical radiance transport maps. In *Proc. Eurographics 2004*, pages 281-290, 2004.
- 15) J. Kautz, J. Lehtinen, and T. Aila. Hemispherical rasterization for self-shadowing of dynamic objects. In *Proc. Eurographics Symposium on Rendering 2004*, pages 179-184, 2004.

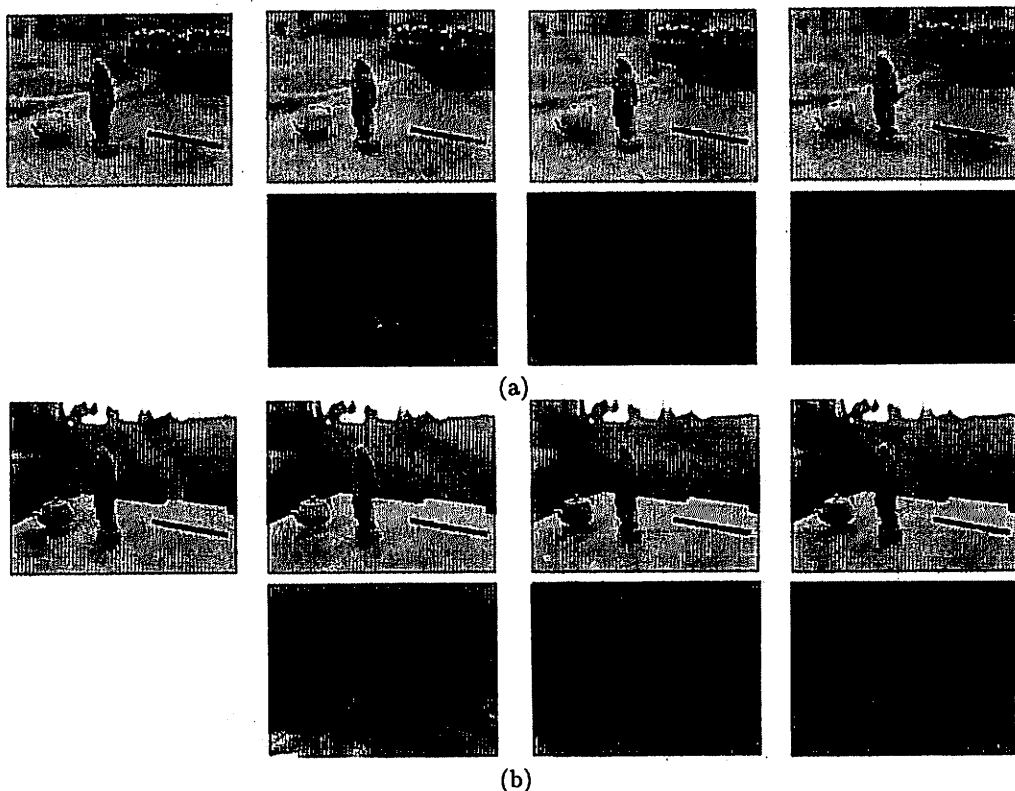


図 3 しきい値を変えてレンダリングした結果. (a): 低周波近似 (b): 全周波近似. (a), (b) において左: 均一サンプリングの結果 (リファレンス画像), 左 2 番目から順にしきい値を, 0.020, 0.010, 0.005 に設定した場合の結果. 下段はリファレンス画像との差分画像 (15 倍にスケール).



図 4 ボーリングシーン (15 物体, 41,340 頂点).



図 5 物体落下シーン (20 物体, 88,739 頂点).