コンピュータビジョンと
イメージメディア 113－7
（1998. 11. 19）

# Calibrated Computer Graphics のための高速画像生成

馬場 雅志　天野 晃　青山 正人　浅田 尚紀

広島市立大学 情報科学部 知能情報システム工学科

E-mail: {baba, a-amano, masa, asada}@its.hiroshima-cu.ac.jp

**概要：** Calibrated Computer Graphics のための画像生成手法の高速化について述べる．レンズのぼけ効果を忠実にモデル化した逆投影ぼけモデルに基づく多重Zバッファアルゴリズムは高品質なぼけ画像生成が行えるが，生成画像の1点につき1～2回のZバッファ法を適用するため，その計算コストは高いものとなっている．そこで，多重Zバッファアルゴリズムの高速化について検討し，高速化の効果と画質への影響について評価を行った．さらに，多数のピンホール画像を加算することによって同様のぼけ画像を生成する方法についても検討を加え，多重Zバッファアルゴリズムと比較した結果についても述べる．

**キーワード：** Calibrated Computer Graphics，画像生成，多重Zバッファアルゴリズム，多重ピンホールアルゴリズム

# Improvement of Image Generation Algorithms for Calibrated Computer Graphics

Masashi Baba, Akira Amano, Masahito Aoyama, Naoki Asada

Department of Intelligent Systems, Hiroshima City University

**Abstract:** For the Calibrated Computer Graphics, an approach to realistic image synthesis based on camera calibration, we have developed the multiple Z-buffer algorithm which enables us to produce high quality images including blurred objects. However, its computation is rather heavy because the standard Z-buffer algorithm is applied on every pixel of an image to be generated. In this paper, we introduce a multiple pinhole algorithm in which images are generated by accumulating a set of shifted pinhole images, and discuss the improvement of both algorithms by evaluating the trade-off between the reduction of computation time and the degradation of image quality.

**Keywords:** Calibrated Computer Graphics, Image Generation, Multiple Z-buffer Algorithm, Multiple pinhole algorithm

## 1 はじめに

仮想現実 ( Virtual Reality ) や拡張現実 ( Augmented Reality ) では，仮想空間と実空間をシームレスに融合した画像を生成する技術が求められている．仮想空間と実空間を融合する方法として，

(1) 仮想空間への実写画像の埋め込み

(2) 実写画像への仮想物体の埋め込み

の2種類の方法が考えられるが，いずれの場合にも実写画像と同等の品質の画像を生成する技術が必要となる．これは，実写画像を撮影した実カメラの特性に基づいて画像を生成し，実写画像と合成する必要があることを意味している．

実カメラの特性を忠実に再現するには，レンズによるぼけの効果を考慮する必要がある．従来のCGでは，ピンホールカメラモデルを用いて作成した画像に距離に応じたぼけを付加する手法 [1]，レンズを通過する複数の光線で画像を生成することによってレンズ効果を表現する Distributed Ray Tracing [2]，レンズの設計データに基づいてズームレンズの効果を表現する手法 [3] などが提案されている．しかし，これらの研究では実カメラとの対応付け，すなわちカメラキャリブレーションを行っていないため，実写画像との比較や合成は行われていない．

これに対して我々は，CV と CG に共通する技術要素であるカメラモデルに着目し，カメラキャリブレーションに基づいて CG 画像を生成し，実写画像並みのリアルな合成画像を作成する “Calibrated Computer Graphics”(CCG) の研究を進めている [4, 5, 6]．文献 [4] では，レンズのぼけ効果を忠実にモデル化した逆投影ぼけモデル [7] に基づいて，Z バッファを利用した高品質なぼけ画像生成アルゴリズム (多重 Z バッファアルゴリズム) を示した．さらに，文献 [5, 6] では，ズーム・フォーカス・アイリス統合カメラモデルを提案し，実カメラのキャリブレーションに基づいて3種類のレンズパラメータを自由に変更した合成画像を作成する方法を示した．

しかし，多重 Z バッファアルゴリズムは，生成画像の1点につき1～2回の Z バッファ法を適用する必要があるため，その計算コストは高いものとなっている．したがって，CCG を仮想現実や拡張現実に適用するには，画像生成アルゴリズムを高速化する必要がある．そこで本論文では，多重 Z バッファアルゴリズムの高速化について検討し，高速化の効果と画質への影響について評価を行った．さらに，多数のピンホール画像を加算することによって同様のぼけ画像を生成する方法 [8](以下では，多重ピンホールアルゴリズムと呼ぶ) についても検討を加え，多重 Z バッファアルゴリズムと比較した結果についても述べる．

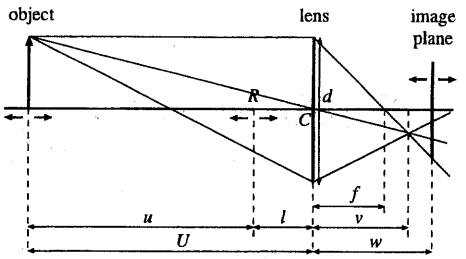



図 1: 統合カメラモデル

## 2 カメラキャリブレーション

カメラモデルとして，図1に示すズーム・フォーカス・アイリス統合カメラモデルを使用する．モデルパラメータとして焦点距離 $f$ [mm]，合焦距離 $U$ [mm]，レンズの開口径 $d$ [mm]，撮像面距離 $w$ [mm]，1画素のサイズ $s$ [mm] を用いる．

カメラキャリブレーションには，カメラ (SONY XC-007) とズームレンズ (FUJINON A16×9BRM-28) を使用し，モデルパラメータを求めた [6]．

## 3 多重 Z バッファアルゴリズム

多重 Z バッファアルゴリズムは，撮像面上の1点に結像する光はシーン中の1点 (合焦点 $O$) を必ず通過するという特徴に着目し，その合焦点からレンズ方向とレンズと反対方向に Z バッファ法を用いることにより撮像面上の1点の明度を求める方法である．

図2に多重 Z バッファアルゴリズムの概略を示す．まず，レンズから光束円錐 $OC_{near}$ を見た $OC_{near}$ 画像と，合焦点 $O$ から光束円錐 $OC_{far}$ を見た $OC_{far}$ 画像を作成し，物体の遮へい効果を考慮して {$OC_{near}$ 画像 > 合焦画像 > $OC_{far}$ 画像 } の優先順位で合成 $OC$ 画像を作成する．そして，合成 $OC$ 画像の平均明度を最終生成画像上の1点の明度とする．$OC_{near}$ 画像はレンズ位置に，$OC_{far}$ 画像は点 $O$ に関してレンズと対称の位置に，それぞれレンズ形状の投影面を設定し Z バッファ法[1]を用いて画像を作成する．

[1]Z 値の判定方法が異なる3種類の Z バッファ法を用いる．通常の Z バッファ法で用いられる判定方法は，投影すべき物

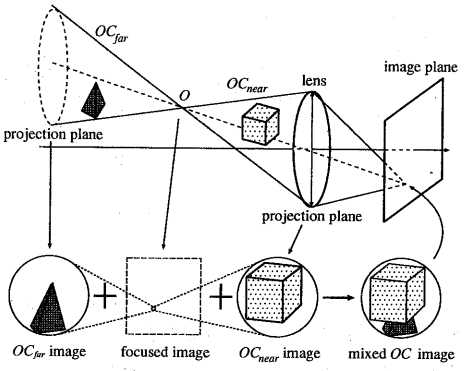



図 2: 多重 Z バッファアルゴリズム

## アルゴリズム

画像生成に必要なパラメータとして，焦点距離 $f$，撮像面距離 $w$，レンズの口径 $d$，1 画素の大きさ $s$ を与える．また，最終生成画像の画素数を $N$，$OC$ 画像の画素数を $N_{OC}$ とする．

1. 撮像面距離 $w$ から合焦距離 $U$ を求める．
2. 合焦画像の画角を撮像面距離 $w$，画像の 1 辺の画素数 $\sqrt{N}$，1 画素の大きさ $s$ から求める．
3. 画素数 $N$ の等値 Z バッファ法を用いて，Z バッファの値が合焦距離 $U$ に等しい物体の明度で構成した合焦画像を作成する．
4. $OC$ 画像の投影面の画角を，合焦距離 $U$，レンズ口径 $d$ から決定する．
5. 最終生成画像の画素位置 $(x, y)$ を与える．

以下の処理 (6から 14) を画素数 $N$ だけ繰り返す．

6. 最終生成画像の全画素の明度が求まれば終了．
7. 画素位置 $(x, y)$ と撮像面距離 $w$ から合焦点 $O$ を求める．
8. 画素数 $N_{OC}$ の順 Z バッファ法を用いてレンズ形状画像 ($OC$ マスク画像) を作成する．合焦点 $O$ からレンズ中心を投影中心としてレンズ形状の円形物体を投影し，その物体の内部領域を $OC$ 画像とする．
9. 画素位置 $(x, y)$ において，3で作成した合焦画像の値が得られている場合 (合焦点 $O$ に物体が存在する場合) は，10を行い 11をスキップする．それ以外は 10以下を実行する．
10. 逆 Z バッファ法を用いて画素数 $N_{OC}$ の $OC_{near}$ 画像を作成する．合焦点 $O$ からレンズ中心を投影中心として，$O$ とレンズ間に存在するすべての物体を投影する．
11. 順 Z バッファ法を用いて画素数 $N_{OC}$ の $OC_{far}$ 画像を作成する．合焦点 $O$ からレンズと合焦点に対して対称な点を投影中心として，$O$ と投影面間に存在するすべての物体を投影する．

次の 12の処理を $OC$ 画像の画素数 $N_{OC}$ だけ行う．

12. $OC_{near}$ 画像と $OC_{far}$ 画像から合成 $OC$ 画像を作成する．レンズに近い $OC_{near}$ 画像を優先し，$OC_{near}$ 画像の値が得られていない画素については，対応する $OC_{far}$ 画像の画素の値を用いる．
13. 合成 $OC$ 画像の画素値を平均し，最終生成画像の 1 画素 $(x, y)$ の明度とする．
14. 新たな画素位置を $(x, y)$ とし 6 に戻る．

このアルゴリズムにおいて，計算コストが高いのは Z バッファ法を用いるステップ 3，8，10，11 である．ただし，ステップ 11の実行は，合焦点 $O$ に物体が存在するか否かに依存する．そこで，画素数 $N$ に対して $O$ に物体が存在しない比率を $\varepsilon(0 \leq \varepsilon \leq 1)$ とすると，Z バッファ法を適用する回数は次式で表される．

$$(2+\varepsilon)N + 1 \qquad (1)$$

Z バッファ法の計算時間は生成する画像の画素数に比例すると仮定すると，全体の計算時間 $T_{MZ}$ は次式で表される．

$$T_{MZ} = kk_{OC}(2+\varepsilon)NN_{OC} + kN + \delta \qquad (2)$$

ただし，$\delta$ は Z バッファ法以外の処理に要する時間，$k$，$k_{OC}$ は比例定数とする．

### 3.1 高速化の検討

多重 Z バッファアルゴリズムを高速化する方法として，以下の 2 つの方法を検討した．

体の奥行き値がすでに蓄えられているものよりも小さければ，物体を描画し奥行き値を入れ替える方法である．これを「順 Z バッファ法」と呼ぶ．これに対して，奥行き値が大きい物体を描画し奥行き値を入れ替える方法を「逆 Z バッファ法」と呼ぶ．また，奥行き値が等しい時だけ描画する方法を「等値 Z バッファ法」と呼ぶ．

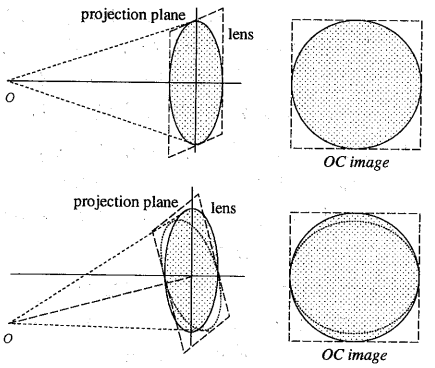



図 3: $OC$ マスク画像の作成

### 3.1.1 $OC$ マスク画像の形状

$OC$ 画像の形状は，レンズ形状の円形物体を投影してできた楕円形の内部領域として求めている (ステップ 8)．この $OC$ マスク画像の形状は，合焦点が光軸から離れるにしたがって楕円形に変形していくが，画角が大きくない場合には，$OC$ マスク画像の形状はほぼ円形とみなすことができる．

そこで，光軸上の一点で円形の $OC$ マスク画像を求め，合焦点が光軸から離れても同じ $OC$ マスク画像を使用することで高速化を図る．この計算は最終生成画像の画素数 $N$ だけ繰り返されるため，この処理を省くことにより計算時間が短縮し，次式になると期待できる．

$$T_{MZ} = kk_{OC}(1+\varepsilon)NN_{OC} + kN + \delta \qquad (3)$$

### 3.1.2 $OC$ 画像の画素数 $N_{OC}$

$OC$ 画像の画素数 $N_{OC}$ を少なくすることにより，1 回の Z バッファ法の計算時間の短縮が行える．上記アルゴリズムの処理時間のほとんどは，Z バッファ法による物体の投影に費やされるため，画素数

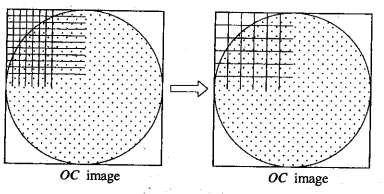



図 4: $OC$ 画像の画素数の削減

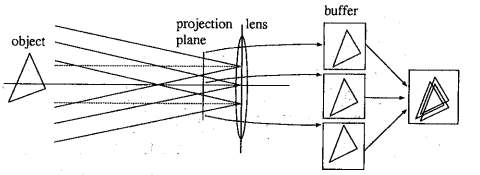



図 5: 多重ピンホールアルゴリズム

$N_{OC}$ を削減することによって $OC$ 画像の作成時間を大幅に短縮することが期待できる．ただし，画素数を少なくすると，$OC$ 画像においてエリアシングなどの問題が発生し，最終生成画像の画質が低下する可能性がある．

## 4 多重ピンホールアルゴリズム

図 5 に示すように，視点位置を少し変えた投影軸が平行なピンホール画像を複数枚作成し加算平均することにより，ぼけを含んだ画像を作成することが可能である．この方法では，投影面上に物体を投影するときに，合焦平面での物体が同じ領域に投影される様に，画像の描画領域を平行移動する．これによって，合焦平面にある物体はぼけないで，それ以外にある物体はぼける効果を出すことが可能である．

これは，Haeberli が提案した Accumulation Buffer を用いたアンチエリアシング，モーションブラー，被写界深度を表現する方法 [8] の応用として知られるが，実際のカメラモデルとの対比や画像生成時の画角の決定法については議論されていない．以下では，多重ピンホールアルゴリズムによって多重 Z バッファアルゴリズムと等価なぼけ画像が生成できることを示し，画像生成のための画角の決定法について述べる．

### 4.1 画像のシフト加算による画像生成

図 6 に示すように，レンズ中心を原点とし，物体空間を正方向とする光軸上に $\alpha$ 軸，レンズの半径方向に $\beta$ 軸とする $\alpha-\beta$ 座標を定める．レンズの焦点距離を $f$ とし，$\alpha = -w$ に撮像面が位置するとする．光軸上の点 $A(a,0)$ からレンズ上の点 $P(0,d)$ に至る光線は，レンズによって屈折し，

$$\frac{1}{f} = \frac{1}{a} + \frac{1}{b} \qquad (4)$$

を満たす光軸上の点 $B(-b,0)$ を通過する．この光

線は，撮像面上の点 $W_0(-w, -\frac{d}{b}w + d)$ に像を結ぶ．

ここで，点 $P$ の位置にピンホールを置き，撮像面上にピンホール画像を生成することを考える．点 $A$ から点 $P$ に入射した光線は直進し，撮像面上の点 $W_1(-w, \frac{d}{a}w + d)$ に像を結ぶ．

したがって，$W_0W_1$ 間の距離は

$$\overline{W_0W_1} = dw\left(\frac{1}{a} + \frac{1}{b}\right) = \frac{dw}{f} \qquad (5)$$

となり，ピンホール位置 $d$ と撮像面距離 $w$，そして焦点距離 $f$ のみで決まり，点 $A$ の位置に依存しないことが分かる．このことは，シーン中の任意の点からピンホール $P$ に入射する光線が撮像面上に結ぶ像は，一定距離 $\overline{W_0W_1}$ シフトすることによって，レンズによって屈折した光線が撮像面上に結像する位置に一致することを示している．

### 4.2 ピンホール画像の画角

多重ピンホールアルゴリズムにおける各ピンホール画像の画角は，レンズの中心での画角 (図7の $\theta$) ではなく，図7の $\varphi$ の画角を使用する必要がある．これは，合焦点とレンズから作られる円錐の内部を通過する光線が，レンズに入射し撮像面上で結像するからである．

#### アルゴリズム

画像生成に必要なパラメータとして，焦点距離 $f$，撮像面距離 $w$，レンズの口径 $d$，1画素の大きさ $s$ を与える．また，最終生成画像の画素数を $N$，ピンホール画像の画素数を $N_Z$，中間生成画像の数 (レンズ面のサンプル数) を $L$ とする．

1. 最終生成画像の1辺の画素数 $\sqrt{N}$，1画素の大きさ $s$，撮像面距離 $w$ からピンホール画像の画角を求める．
2. レンズ面上のピンホール位置を $(\xi, \eta)$ とする．

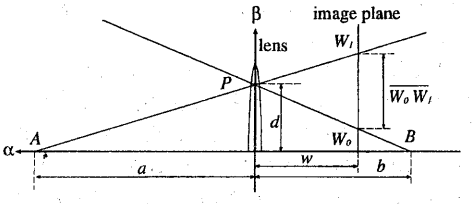



図 6: ピンホール画像のシフト加算によるぼけ画像生成

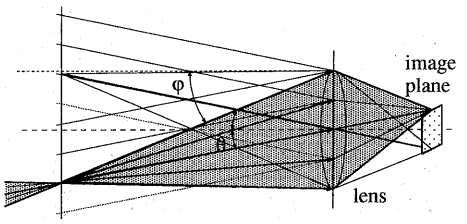



図 7: ピンホール画像の画角

以下の処理をレンズ面のサンプル数 $L$ だけ繰り返す．

3. 順Zバッファ法を用いてピンホール位置 $(\xi, \eta)$ から光軸に平行な方向を見た画素数 $N_z$ の画像を作成する．
4. 合焦点での画像が一致するようにピンホール画像を平行移動し加算する．
5. 新たな画素位置 $(\xi, \eta)$ を与え3に戻る．
6. 加算した画像を画像数 $L$ で割り，最終生成画像を得る．

この手法において，Zバッファ法を適用する回数は $L$ 回となる．これに，Zバッファ法の計算時間が生成する画像の画素数 $N_z$ に比例すると仮定すると，全体の計算時間 $T_{MP}$ は次式で表される．

$$T_{MP} = k_Z L N_Z + \delta \qquad (6)$$

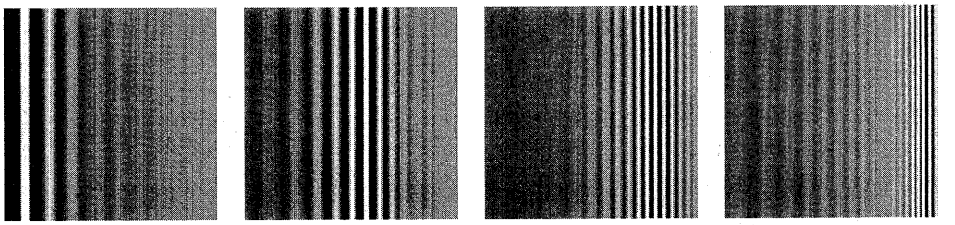
## 5 計算時間と画質の評価

画像生成には，完全拡散面で構成された短冊状の物体を左から右に順番に距離が遠くなるように配置したシーンを設定した．ズーム，絞りは固定とし，フォーカスを段階的に変化させた画像を作成した (図8)．画像生成には，SGI社 O2 (CPU R10000, クロック 250MHZ, メモリ 256MB, OS IRIX 6.3) を使用した．

### 5.1 多重Zバッファアルゴリズム

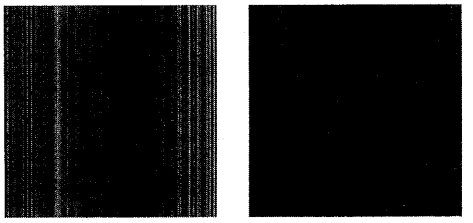
#### 5.1.1 $OC$ マスク画像の形状

$OC$ マスク画像の形状を円形に固定し作成した画像は，1点ごとに $OC$ マスク画像の形状を求める手法と比較して，画像の輝度値のレベルでは全く変化がなかった．これは，作成した画像の画角が小さいため，合焦点が比較的光軸に近く歪みが小さかったことが考えられる．$OC$ マスク画像を円形に固



(a) 合焦距離 $U = 220$　(b) 合焦距離 $U = 320$　(c) 合焦距離 $U = 420$　(d) 合焦距離 $U = 520$

図 8: 生成画像の例



(a) 画素数 $N_{OC} = 20^2$　(d) 画素数 $N_{OC} = 80^2$

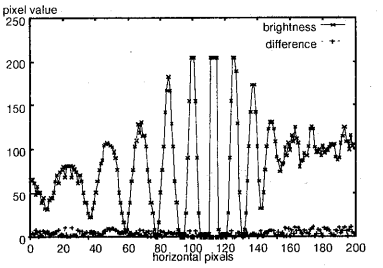
図 9: 画素数 $N_{OC} = 100^2$ の画像との差分

定することによって最終画像の生成時間は，投影面の画素が $N_{OC} = 80^2$ のとき 156 秒から 119 秒に短縮した．

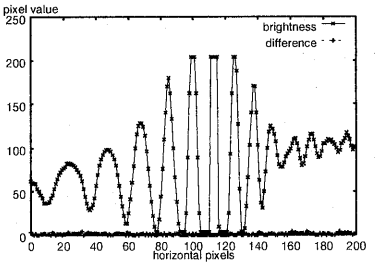
### 5.1.2 *OC* 画像の画素数 $N_{OC}$

*OC* 画像の画素数 $N_{OC}$ を変えたときの画像と $N_{OC} = 100^2$ の時の画像との差分をとった例を図 9に示す．また，画像の横方向での明度プロファイルと $N_{OC} = 100^2$ の時の画像との差を図 10に示す．ここで，横軸は画像の水平画素位置を表す．画質に関しては，$N_{OC}$ を少なくすると，生成画像上では縦の縞のようなものが見える．これを明度プロファイルで見ると，$N_{OC}$ が多い時は明度はなだらかに変化しているが，$N_{OC}$ を少なくすると凹凸が目立つようになり明度差も大きくなっている．

$N_{OC}$ を少なくすると計算時間は短くなるが，最終生成画像の画質が低下する．$N_{OC}$ を変えた時の計算時間の変化と差の平均値を図 11に示す．ただし，縦軸は左側が明度値の差の平均値，右側が計算時間を表し，横軸は $N_{OC}$ を表す．計算時間は，$N_{OC}$ に対してほぼ線形に増加しているが，差の平均値は $N_{OC} = 50^2$ までは急激に減少し，その後なだらかに変化してほぼ 1 に収束している．




(a) 画素数 $N_{OC} = 20^2$




(d) 画素数 $N_{OC} = 80^2$

図 10: $U = 320$ の時の $N_{OC}$ の変化による画質の変化

## 5.2 多重ピンホールアルゴリズム

フォーカスを変化させて複数の画像を作成し，多重 Z バッファアルゴリズムと同様な画像が作成できることを確認した．

加算画像の数 $L$ を少なくすると画像作成に要する時間は短くなるが，最終生成画像の画質が低下する．図 12に示すように，特に合焦距離から離れた場所の誤差が大きくなる傾向がある．$L$ を変えた時の計算時間の変化と誤差の平均値の変化を図 13に示す．ただし，縦軸は左側が明度値の差の平均値，右側が計算時間を表し，横軸は加算画像の数 $L$ を

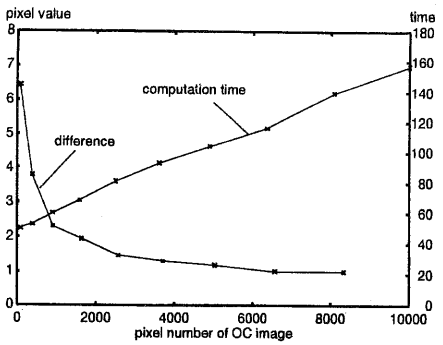



図 11: 計算時間と誤差の変化 (多重 Z バッファアルゴリズム)

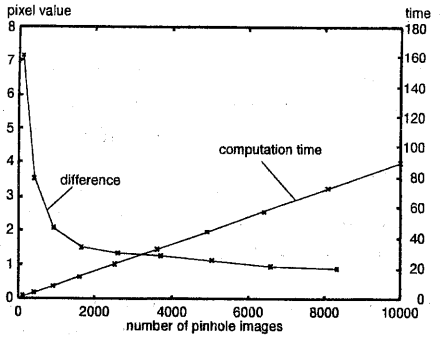



図 13: 計算時間と誤差の変化 (多重ピンホールアルゴリズム)

表す．画像生成に要する時間は，$L$ に対してほぼ線形に増加している．差の平均値は $L = 50^2$ までは急激に減少し，その後なだらかに変化してほぼ 1 に収束している．

また，ハードウェアの accumulation buffer による効果を確認するために，加算枚数を変化させたときの計算時間を調べた．使用した計算機の accumulation buffer では，126 枚の加算までハードウェアで行うことが可能であり，それ以上の加算はソフトで行っている．図 14の横軸はハードウェアによる加算枚数を表し，縦軸は計算時間を表している。グラフから画像の加算枚数が 16 枚以上になれば，4 倍以上の高速化が可能であることが分かる．

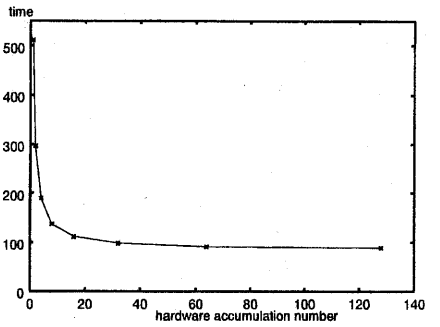
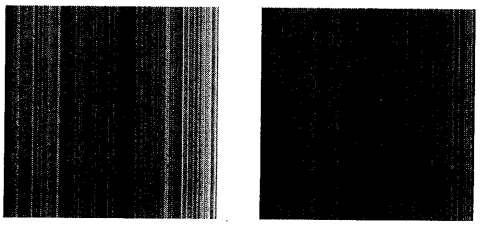



図 14: Accumulation Buffer の効果

## 6 まとめ

本論文では，Calibrated Computer Graphics のための画像生成手法の高速化について述べた．まず，多重 Z バッファアルゴリズムの高速化に関する検討を行い，$OC$ マスク画像の形状の固定と，$OC$ 画像の画素数 $N_{OC}$ を変化させる効果について述べた．また，多重ピンホールアルゴリズムを用いることに関して，その妥当性を検討し，画像作成の時間に関係する加算画像数に関する検討を行った．



(a) 加算画像数 $L = 20^2$　(d) 加算画像数 $L = 80^2$

図 12: 加算画像数 $L = 100^2$ の画像との差分

それぞれの手法においては計算時間と画質に関する検討を行った結果，多重 Z バッファアルゴリズムの $OC$ 画像の画素数 $N_{OC}$ と，多重ピンホールアルゴリズムのレンズのサンプル数 (画像の加算画像数)$L$ が同じ値の場合には，多重ピンホールアルゴリズムの方が高速であることが分かった．しかし，両手法で全く同じ画質の画像を作成することが困難であったため，画質を基準として両手法間を比較するまでには至らなかった．

今後は，両手法を比較する画質の評価基準について検討を加え，多様なシーン設定 (表示する仮想物体数やテクスチャの有無) における性能比較を行う予定である．

## 参考文献


[1] M. Potmesil, I. Chakravaty: "A lens and aperture camera model for systhetic image generation", Proceedings of SIGGRAPH '81, pp.297–305, 1981.

[2] R.L.Cook: "Distributed ray tracing", Proceedings of SIGGRAPH '84, pp.137–145, 1984.

[3] C. Kolb, D. Mitchell, P. Hanrahan: "A realistic camera model for computer graphics", Proceedings of SIGGRAPH '95, pp.317–324, 1995.

[4] 馬場 雅志, 浅田 尚紀, 天野 晃: "Calibrated Computer Graphics による画像合成の試み – カメラ キャリブレーションに基づく 任意フォーカス画像の生成と検証 – ", 情報処理学会論文誌, Vol.39, No.7, pp.2180–2188, 1998.

[5] 馬場 雅志, 浅田 尚紀, 迫田 肇, 天野 晃, 青山 正人: "ズーム・フォーカス・アイリス統合カメラモデルを用いた Calibrated Computer Graphics", Visual Computing / グラフィクスと CAD 合同シンポジウム, pp.67–72, 1998.

[6] 馬場 雅志, 天野 晃, 青山 正人, 浅田 尚紀: "Calibrated Computer Graphics のためのズーム・フォーカス・アイリス統合カメラモデル", 画像の認識・理解シンポジウム (MIRU '98) 論文集, pp.II47–II52, 1998.

[7] 浅田 尚紀, 藤原 久永, 松山 隆司: " 逆投影ぼけモデルを用いた遮へいエッジの光学的性質の解析 ", 電子情報通信学会論文誌 D–II, Vol.J78–D–II, No.2, pp.248–262, 1995.

[8] P. Haeberli, K. Akeley: "The Accumulation Buffer: Hardware Support for High-Quality Rendering", Proceedings of SIGGRAPH '90, pp.309–318, 1990.