

# 微小面ビルボーディングを用いた複雑なシーンの表示手法

## Rendering intricate scenes using Microfacet Billboarding

### 概要

樹木や毛髪などの極端に複雑な物体の幾何形状を正確に計測することは難しく、計測によって得られた情報からこれらの物体を正確に表示することは困難である。我々は、物体の幾何形状を視線方向に依存して方向の変化する微小面の集合で近似し、実画像をテクスチャとして貼り付けることにより、複雑な形状の物体や風景を効率的に表示するための、微小面ビルボーディング法を提案する。微小面は視点の移動に伴って回転し、面と視線が常に直交するように配置される。微小面に張られるテクスチャ画像はカメラと視線方向、微小面の位置から決定され、物体までの距離に応じて切り抜かれる。微小面のサイズは計測された幾何形状の精度から決定され、その精度が良い場合にはサイズを小さくすることで形状モデルに基づく描画法 (MBR) に近づき、幾何形状が信頼できない場合にはサイズを大きくすることで画像に基づく描画手法 (IBR) に近づく。以上の処理はグラフィクスハードウェアの機能を使って効率的に実装することが可能であり、複雑な物体や風景を実時間で描画できることを示す。

Rendering of intricately shaped objects such as those that are soft or cluttered is difficult because we cannot acquire their geometry completely. Since their geometry varies volumetrically, modeling them by fixed facets exceeds the tolerances from some singular view directions. In this paper, we propose a novel modeling method which uses view-dependent “micro” facets with view-dependent textures. The facets are positioned vertically to the view direction and the texture of each facet is constrained by alpha and depth clipping techniques. Thus, volumetric surface can render fine geometry from various view directions. The size of each facets changes according to the approximate geometry of the model and the resolution of rendering. We also describe how we have developed a real-time renderer that has a hardware-accelerated clipping technique.

## 1 はじめに

実世界の物体や風景を計測し、コンピュータグラフィクスとして再現することは、仮想現実感、拡張現実感といった現実感モデルを構築する上で非常に重要な処理である。しかし、複雑な形状を持つ物体を正確に計測、または表現することは困難であり、厳密に処理するためのコストが極めて高い。この問題に対して、幾何モデル、光学モデルの点から多くの研究がなされているが [3, 4, 5]、樹木の葉や毛髪などの極端に複雑な物体の表示は依然として困難である。

ある程度の大きさを持つ物体の形状を測定する最も精度の高い方法として、レーザーレンジファインダーによる計測が挙げられる。この方法では物体の形状を 0.1–10.0mm の精度で計測することが可能であるが、毛髪などの細かく入り組んだ形状を正確に計測するにはこの精度では不十分である。またこの手法は計測精度が撮影対象の表面特性に影響されやすく、また撮影できない対象も多い。さらに、複雑な幾何形状を正確に表現するために必要なデータ量は膨大であり、これらのデータを全て保持することは現実的ではない。

一方で、幾何形状を取得せずに、2次元画像から見えを合成する手法 (Image Based Rendering, IBR) がある。この手法は、入力画像を増やすことにより原理的にはあらゆる風景を再現することが可能であるが、そのために必要なデータ量はしばしば膨大であり、撮影に要するコストとデータを保持するコストが高いことから利用できる状況は限られている。また、幾何形状を持たないことから、合成された形状を他の環境に持ち込んだり、視点を大きく変えるなど

の変更が困難である。

これらの問題を解決するために、我々は、計測によって得られた幾何形状を最大限に活用しつつ、画像に基づく見えの合成を行って任意の対象物体を表示する手法を提案する。そのために、微小面ビルボーディングと呼ばれる新しいモデル化を提案し、それを使って計測物体を表示する方法を説明する。

本手法の概要は以下のとおりである。入力として与えられるものはセンサーなどを用いた計測によって得られた物体の幾何形状、および複数視点から撮影された対象物の2次元画像列である。両者は位置の校正が来ているものとする。物体の幾何形状を用いて、計測対象物の表面を微小面の集合で表現する。この微小面を観測者の視線方向に応じて回転させ、面と視線が常に直交するように配置する。これらの微小面に対し、カメラで撮影された物体の実画像が投影され、テクスチャ表示される。カメラ画像は、カメラ視線に沿った物体までの距離に応じて、必要な部分だけが切り抜かれて表示される。微小面のサイズは計測された幾何形状の精度から決定され、精度が良い場合にはサイズを小さくすることで形状モデルに基づく描画法 (MBR) に近づき、逆に幾何形状が全く信頼できない場合にはサイズを大きくすることで画像に基づく描画手法 (IBR) に近づく。

本手法は、画像を用いた表示手法の特徴として、任意の対象に適用可能であるという性質を持ち、同時に利用できる幾何形状を最大限に活用することにより遠近感のある表示実現できるほか、遮蔽や光源の影響など、他の物体との相互作用を考慮した表示を行うことが可能である。

## 2 微小面によるモデル化

計測から描画にいたるプロセスを大まかに分類すると以下ようになる。(1) 幾何形状を元に空間を小領域に分割し、各領域に描画対象が存在する場合は微小面を発生させる。(2) 視線方向と入力画像の撮影方向に応じて微小面に対するテクスチャ画像を決定する。(3) 描画対象までの距離に応じて微小面上の不要なテクスチャ画像を切り抜く。以下、個々の処理について詳しく述べる。

### 2.1 微小面の生成

まず計測結果の形状を微小面の集合で近似する(図1右図)。空間を小領域に分割し、計測によって得られた幾何形状を再サンプリングすることで粗い幾何形状を得る。この方法として我々はボクセル分割に基づく方法を採用した。まず、表示対象を完全に含むボリューム空間を考え、ボクセル上で幾何形状を再サンプリングする。もしボクセル内に何も存在しなければそのボクセルは空として以下の処理を全て省略し、何らかの形状が存在すればそこに微小面を発生させ、以下の節で述べる処理を行う。

微小面は空間内の四辺形として定義され、面の中心はボクセル中心と一致させる(図1左図)。任意の方向から見た時にこの微小面でボクセル全体を被覆するために、視線方向に変化に伴って微小面を回転させ、微小面が視線方向と常に垂直になるように拘束する。この処理はビルボーディングと呼ばれ、実時間指向のグラフィクスシステムで用いられる手法である。回転を行う際に、ボクセル全体が常に被覆されるためには、ボクセルの幅を  $w$  とすると、微小面の幅が  $\sqrt{3}w$  以上であれば良い。

### 2.2 テクスチャ画像

各微小面にはカメラで撮影された画像が投影され、テクスチャとして貼り付けられる。カメラ画像が複数ある場合には、視線方向とカメラの撮影方向のなす角  $\theta$  が最も小さ

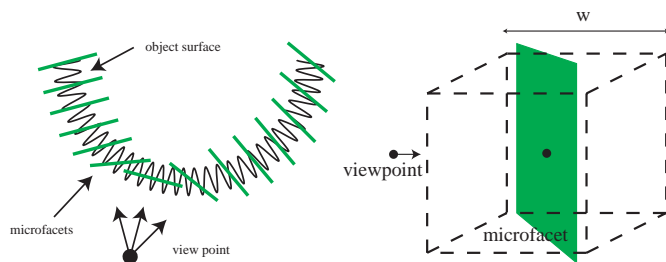


図1: 描画対象の幾何形状を微小面の集合により近似する(左図)。空間をボクセル分割し、描画対象を含むボクセル内に視線方向と垂直な微小面を発生させる(右図)。

いものから順に、その画像が微小面を完全に含むもの選択されて、投影される。選択されるカメラは視点の変更とともに描画時に動的に決定される。カメラの撮影間隔が疎な場合、視点移動の際にカメラ画像が切り替わることによる見えの変化が大きくなり、現実感を損なう。そのため、微小面に対して投影されるカメラ画像を  $\theta$  が小さい方から順に複数枚選択しておきそれらを補間することで、カメラ画像間の移動を滑らかにを行う。

### 2.3 距離情報を用いたテクスチャの切り抜き

微小面に対するテクスチャ画像の選択は、視線方向とカメラの撮影方向を元に決定される。そのため複数個の面に対して同一のテクスチャが選択されることがあり、視線がカメラの撮影地点から離れるにしたがって、テクスチャ上の一つの画素が複数回描画されることになる。一方、テクスチャの各画素は対象の一つの点から得られた情報であるから、これは描画時に同一の物体が複数現れる、ダブルイメージとして現れ、見えの不具合を生じる。

そこで、画像中の各画素に対してあらかじめ描画対象までの距離を持たせておき、描画時に各画素がボクセル内に含まれる場合に限り微小面上に画素を書き出すことで、このダブルイメージを除去する。図2にその例を示す。カメラ画像の各画素には色情報の他に対象物体までの距離が記録されており(中央図)、描画時にその値を微小面の位置と比較して、不要な部分を切り抜いて除去する(右図)。画像に距離情報を付加する処理は、入力幾何形状を画像に射影することで行う。これは前処理としてあらかじめ行うことが可能である。

微小面の発生間隔を  $w$ 、微小面のカメラ方向の微小面上の点の奥行きを  $D$ 、画像に割り当てられている奥行きを  $d$  とする時、テクスチャの切り抜きは描画面上の画素単位で次の関係式に従って行う。

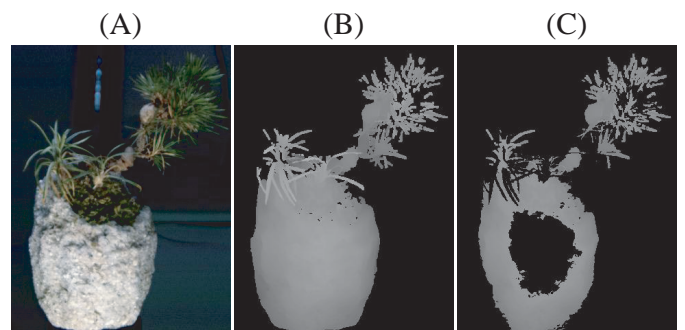


図2: (A) 入力画像 (B) 各画素に割り当てられた対象までの距離 (C) 距離に応じたテクスチャの切り抜きの例

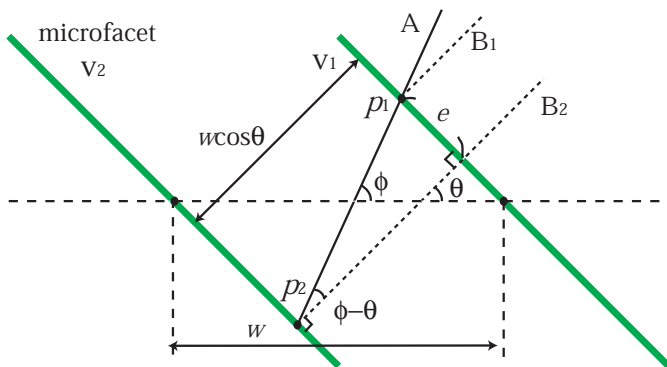


図 4: 表示されるテクスチャの不連続性

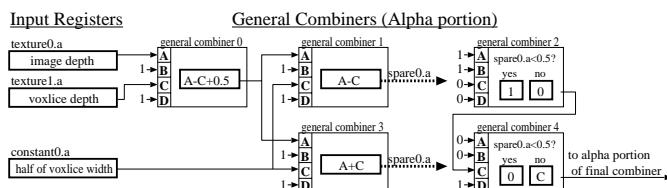


図 3: Register Combiners の設定

$$\begin{cases} |d - D| < \frac{w}{2} & \rightarrow \text{画素を現在の微小面上に描画} \\ \text{それ以外} & \rightarrow \text{画素を破棄} \end{cases} \quad (1)$$

この処理は nVidia GeForce3 グラフィクスカードの Register Combiners テクスチャ合成機能を利用して微小面の描画時に行う。具体的なテクスチャ合成の際のレジスタ割り当ては以下の通り。まずテクスチャとして使用される画像は、RGB チャンネルに撮影によって得られた色画像を、チャンネルに幾何形状を射影して得られた距離画像を割り当てる (texture unit 0 の  $\alpha$  portion に  $D$  をロード)。微小面は四辺形ポリゴンで表現し、視点の位置からポリゴンの向きを決定した後、4 つの頂点をカメラ方向に見た時の距離を 1 次元テクスチャ座標として各頂点に割り当てる。微小面上の各点を描画する際にはこれらのテクスチャ座標を補間した値を、テクスチャ画素値として使う (texture unit 1 に  $d$  をロード)。この時図 3 の通りに Register Combiners をプログラムすることで、式 (1) の判定が実現される。画素を描画する場合には  $\alpha = 1$  に、切り取る場合には  $\alpha = 0$  に置換され、画素の描画/破棄が行われる。

### 3 誤差に基づくパラメータ決定

微小面の発生密度は、画質と描画速度に大きく影響する。一般に発生密度が高ければ高いほど画質は向上するが、描画速度は低下する。そこで画質と描画速度の関係式を導き出し、最適な発生密度を決定することを考える。

微小面には入力画像がテクスチャとして投影されるため、視点位置がカメラ位置にある場合は微小面の解像度によらず結果画像は入力画像と一致する。この時、描画結果の画像には、幾何形状を近似したことによる誤差は全く無いと考えることができる。一方、視点位置がカメラ位置から外れるにしたがって、もともと連続である幾何形状を不連続な微小面の集合で近似したことによる見えの不連続が生じる。

図 4 のように、隣り合う微小面  $v_1, v_2$  に対し、 $A$  方向のカメラが選択され、 $B$  方向から観測しているとする。この時、微小面の発生間隔を  $w$ 、視線の角度を図のように  $\theta$ 、カメラの撮影方向を  $\phi$  とすると、視線方向の微小面の間隔は  $w \cos \theta$  と表せる。 $p_1, p_2$  はカメラ画像上の同一点であるので、これらの近傍は描画時にも連続していることが望ましいが、微小面が不連続であるためにこれらの点は

$$e = w \cos \theta \tan |\phi - \theta| \quad (2)$$

だけ離れて観測される。ある入力カメラ列に対して、 $\delta = \max |\phi - \theta|$  を考えると、 $l \cos \theta < w$  より次式が成立。

$$e < w \tan \delta \quad (3)$$

式 (3) より、テクスチャの不連続を抑制するためには (1)  $w$  を減少させる、即ちボクセル分割を密に行って微小面の発生密度を上げる、もしくは (2)  $\delta$  を減少させる、即ち入力画像を密に撮影することが有効である。多くの場合、入力画像列をある程度以上密にすることは困難であるため、与えられた入力画像列に対し、可能な限りボクセル分割数を変更することで制御を行う。例えば、計測物体を設置し、周囲から  $30^\circ$  間隔でカメラ撮影した画像をテクスチャとする場合、 $\delta < 15^\circ$  より  $e < 0.13w$  を得る。 $e$  をスクリーン上のずれと考えると、 $e < 1$ 、即ち  $w < 7.4$  (単位はスクリーン上の画素) であればテクスチャは連続的に表示されることが分かる。この式により、視点移動した際の微小面の発生密度を、閾値処理によって決定することが出来る (図 5)。

### 4 実験結果

本手法の有効性を検証するため、(A) 物体を世界座標系の原点付近に設置し、周囲から中心方向に計測する場合、(B) カメラを世界座標系の原点付近に設置し、周囲に向かって計測する場合、の 2 種類のカメラ配置で実験を行った。実験に用いた幾何形状、カメラ画像は VIVID900 [7] を用いて撮

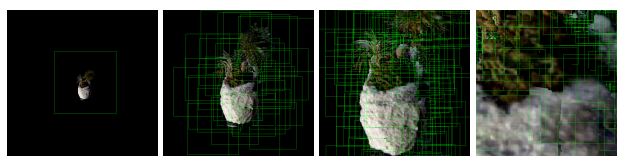


図 5: 視点移動に応じた解像度の変更

影した．この計測系では，カメラ画像と幾何形状が同時に計測できるため，両者の位置合わせを行う必要が無い．撮影時のカメラ位置は測定せず，得られた点群を Neugebauer の手法 [8] を用いて位置合わせした結果を利用してカメラ位置を校正した．

物体を中心に設置した場合 (A) の実験結果を図 6 に示す．実験では毛で覆われた人形をターンテーブル上に設置し，周囲 360 度から 10° 以上の間隔で計測を行った．(a) に計測によって得られた幾何形状，およびテクスチャ画像の一つを示す．得られた幾何形状列に符号付距離変換を行いボリューム空間で再サンプリングしたすると (b) のようなボリュームデータを得る．ここから Wheeler[10] の方法を使って表面形状を再構成した結果が図 (c) であるが，物体表面は毛で覆われているため，計測は不完全であり，正確な幾何形状を復元できない．そこで図 (d) にあるように  $64 \times 64 \times 64$  サイズのボクセル集合から微小面を発生させて幾何形状を近似した．(d) の微小面の色は，選択されているカメラの番号に対応している．これをもとにテクスチャマップを行い，距離に応じて切り抜くことで (e) の結果を得る．この手法を使うと，物体境界付近の幾何形状が曖昧な部分も正確に再現できていることがわかる．

カメラを中心に設置した場合 (B) の実験結果を図 7 に示す．実験に用いた VIVID では一度に計測可能な範囲が狭いため，シーン全体を撮影するために必要な撮影回数は 52 枚であった．図の上段に描画された結果を，下段に微小面ごとに選択されているカメラ位置を示す．幾何形状を利用することで遠近感のある描画を行うことが可能であるが，遮蔽のためテクスチャ画像が存在しない領域には穴があく．

PentiumIII 1GHz, 主記憶 500MByte, グラフィクスカード GeForce3, ビデオメモリ 64MByte の PC で表示実験を行った結果，(A)(B) どちらの結果も，本論文の図を出力した再解像度において，実時間で表示できる事を確認した．

## 5 比較と考察

描画対象の幾何形状を粗く近似し，テクスチャを用いて高精度に表示する手法として，Sprite を利用する方法 [9]，Visual Hull を利用する方法 [6]，Lumigraph Rendering を利用する方法 [2] などが挙げられるが，本手法は描画対象の特徴抽出や背景抽出を一切必要とせず，単純に幾何要素の集合を利用している点で，2次元テクスチャの重ね合わせを利用したボリュームレンダリング [1] にも近い．本手法とこれら手法の最も大きな相違点は幾何形状が視点移動に伴って変化する (view dependent) ことである．そこで微小面をビルボーディングさせることによる描画結果への影響を考察する．

本手法では，幾何形状を微小面で表現するため，微小面の大きさや視線の方向によっては，実際の幾何形状と微小面との乖離幅が大きくなり，歪みが発生する．また，テク

スチャを view dependent にマッピングするため，サンプリングの周期も生成画像に影響を与える．今，注目微小面を一つ選び，図 8 のような配置を考える．図の記号を用いて，仮想視点におけるピクセル位置と実際のピクセル位置とのずれを計算すると，固定微小面の乖離幅  $D_a$ ，ビルボード微小面の乖離幅  $D_b$  はそれぞれ以下の式で計算される．

$$D_a = \left( \frac{\Delta u - \Delta d \cdot \tan(\theta - \phi)}{1 + \tan^2 \theta} - \frac{\Delta u - \Delta d \cdot \tan \theta}{1 + \tan^2 \theta} \right) \frac{f}{dv} \quad (4)$$

$$D_b = \left( \frac{\Delta u - \Delta d \cdot \tan(\theta - \phi)}{1 + \tan \theta \cdot \tan(\theta - \phi)} - \frac{\Delta u - \Delta d \cdot \tan \theta}{1 + \tan^2 \theta} \right) \frac{f}{dv} \quad (5)$$

ただし，仮想視点の焦点距離を  $f$  とし，微小面の大きさに比べて，視点・カメラ位置が十分に物体から離れていることを用いて式を簡略化している．それぞれの式をプロットしたものが，図 9 の上段である．ここでは  $f = dv$  とし，微小面サイズを  $16 \times 16$  pixel としている．また，下段は，左図を 2次元グラフ上に射影したものである．歪みが最も大きくなる状況で計算するため， $\Delta u$  は微小面端の位置， $\phi$  はサンプリング間隔  $\div 2$  に固定した．図 9 の左列から順に，サンプリング周期の上昇に伴い，歪みが減少していることが分かる．また， $f = dv$  としていること，距離に応じたテクスチャ切り抜きにより，ある閾値以上に幾何形状と微小面が乖離しないことを勘案すれば，現在行なっている実験環境においては，発生するピクセル誤差は非常に小さいことが分かる．図の右列より，固定微小面とビルボード微小面の歪みを比較すると，視線方向の変化が大きくなると，固定微小面においては急激に歪みが拡大するのに対して，ビルボード微小面においては歪みが安定していることが分かる．

以上の考察により，ある程度の大きさを持った微小面であっても，幾何形状とのずれによるピクセル誤差は小さい

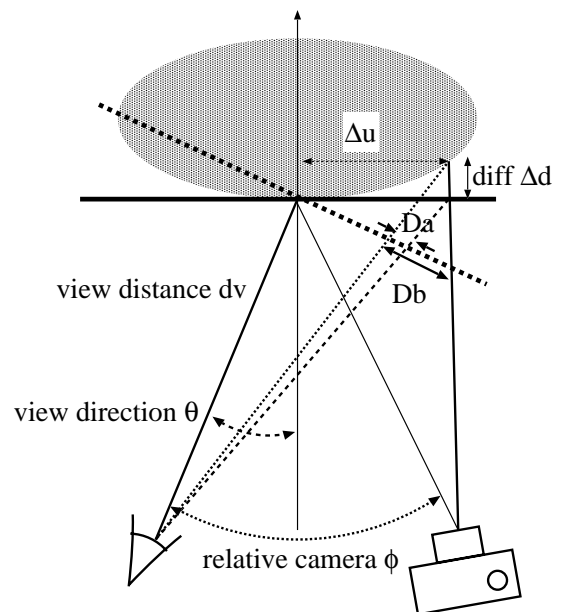


図 8: ファセット形状と実際の形状とのずれによる歪み



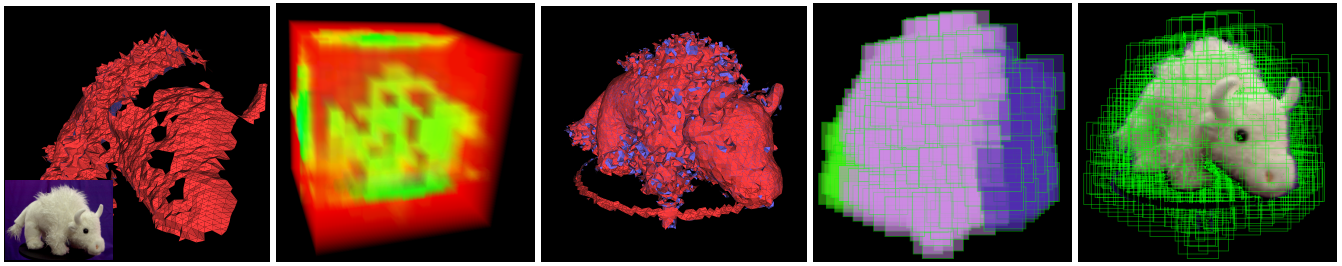


図 6: 物体を中心に設置した場合の実験結果: 左から順に (a) 計測結果の幾何形状とテクスチャの一つ (b) 幾何情報を統合したボリュームデータ (c) 再構成された表面形状 (d) 微小面ビルボードイング. 微小面は選択されたテクスチャで色分けしている. (e) テクスチャマップされた結果

ため, ビルボードイングを行う微小面を利用する本手法が, 現実感ある表示に有効であることが分かる.

## 6 まとめ

我々は, 微小面ビルボードイングを利用した複雑なシーンの描画手法を提案した. この手法は正確な幾何形状を取得, または表現することが困難な物体に対しても適用可能であり, 毛, 樹木, などが従来手法では扱えなかった対象物を精度良く描画できる. 本手法は描画対象を中心に視点が移動する場合にも, 視点を中心として描画対象が周囲に分布する場合にも利用可能であり, 幾何形状の取得が容易な物体と困難な物体が混在する環境でも有効であることを示した. 幾何形状を微小面の集合で近似することによる見えの劣化を, ハードウェアを利用したテクスチャの切り抜きを行うことで抑制し, また描画時に必要な解像度を適応的に選択することで画質を損なうことなく効率よく描画することが可能であり, 複雑なシーンを実時間で描画することが出来た.

本手法では入力 of 幾何形状の精度が高い場合には, 微小面の発生密度を上げることでより正確な描画結果を得ることが可能である. そこで入力形状の各部分毎に精度を評価し, 微小面の発生密度を適応的に変化させることでさらに画質や描画効率を上げることが可能であり, 今後の課題である.

## 参考文献

- [1] M. Brady, K. Jung, H.T. Nguyen, and T. Nguyen. Two-phase perspective ray casting for interactive volume navigation. In *Visualization 97*, 1997.
- [2] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen. Unstructured lumigraph rendering. *ACM SIGGRAPH*, 2001.
- [3] Tim Hawkins, Jonathan Cohen, and Paul Debevec. A photometric approach to digitizing cultural artifacts. In *2nd International Symposium on Virtual Reality, Archaeology, and Cultural Heritage*, Glyfada, Greece, November 2001.
- [4] K. Ikeuchi, Y. Sato, K. Nishino, R. Sagawa, T. Nishikawa, T. Oishi, I. Sato, J. Takamatsu, and D. Miyazaki. Modeling cultural heritage through observation. In *Proc. of IEEE First Pacific-Rim Conference on Multimedia*, Dec. 2000.
- [5] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. The digital michelangelo project: 3D scanning of large statues. In *Proc. SIGGRAPH 2000*, pages 131–144, 2000.
- [6] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. In *Eurographics Workshop on Rendering*, 2001.
- [7] Minolta. Vivid 900 non-contact digitizer. <http://www.minoltausa.com/vivid/>.
- [8] P.J. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. In *Shape Modeling and Applications*, 1997.
- [9] R. Szeliski, P. Anandan, and S. Baker. From 2d images to 2.5d sprites: A layered approach to modeling 3d scenes. In *IEEE International Conference on Multimedia Computing and Systems*, 1999.
- [10] M.D. Wheeler, Y. Sato, and K. Ikeuchi. Consensus surfaces for modeling 3d objects from multiple range images. In *Proc. International Conference on Computer Vision*, January 1998.

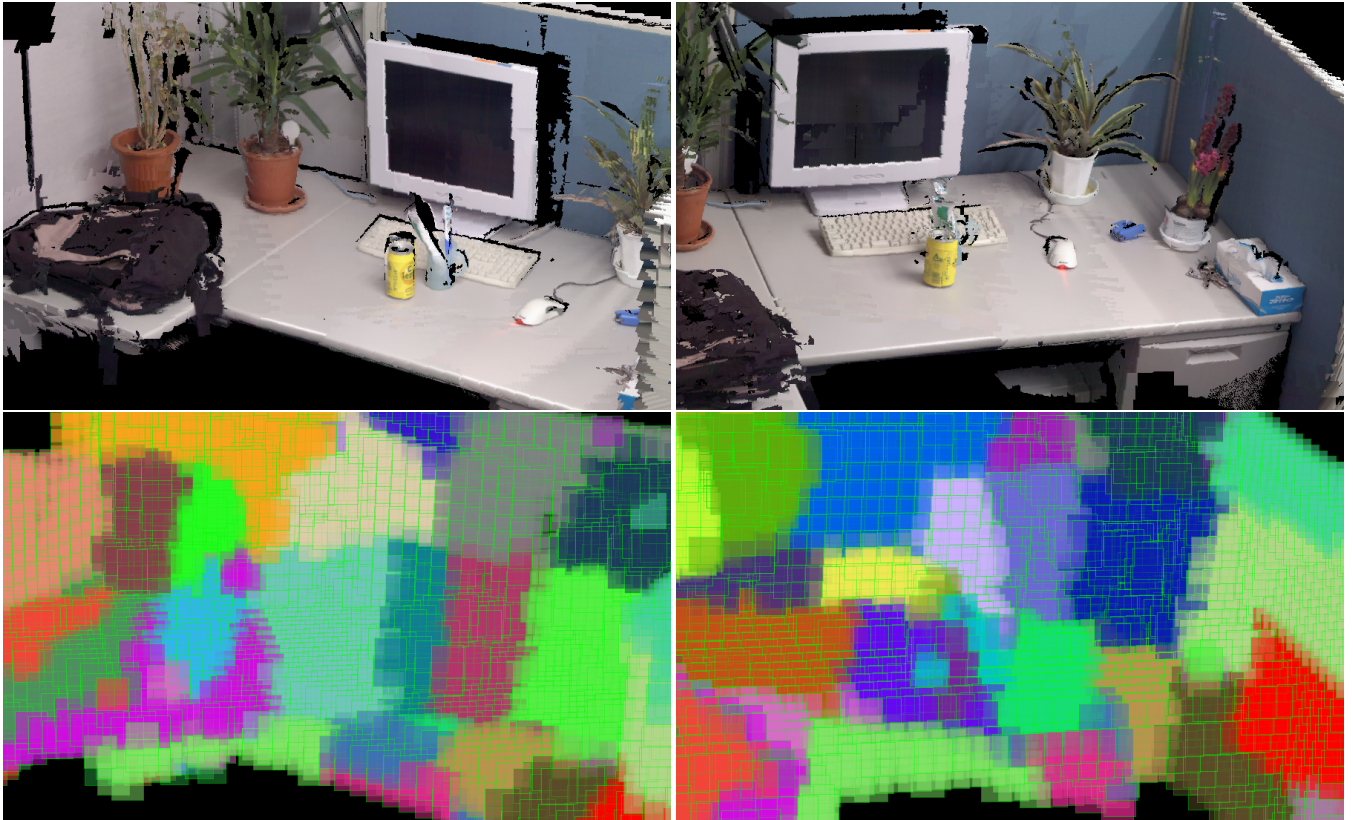


図 7: カメラを中心に設置した場合の実験結果

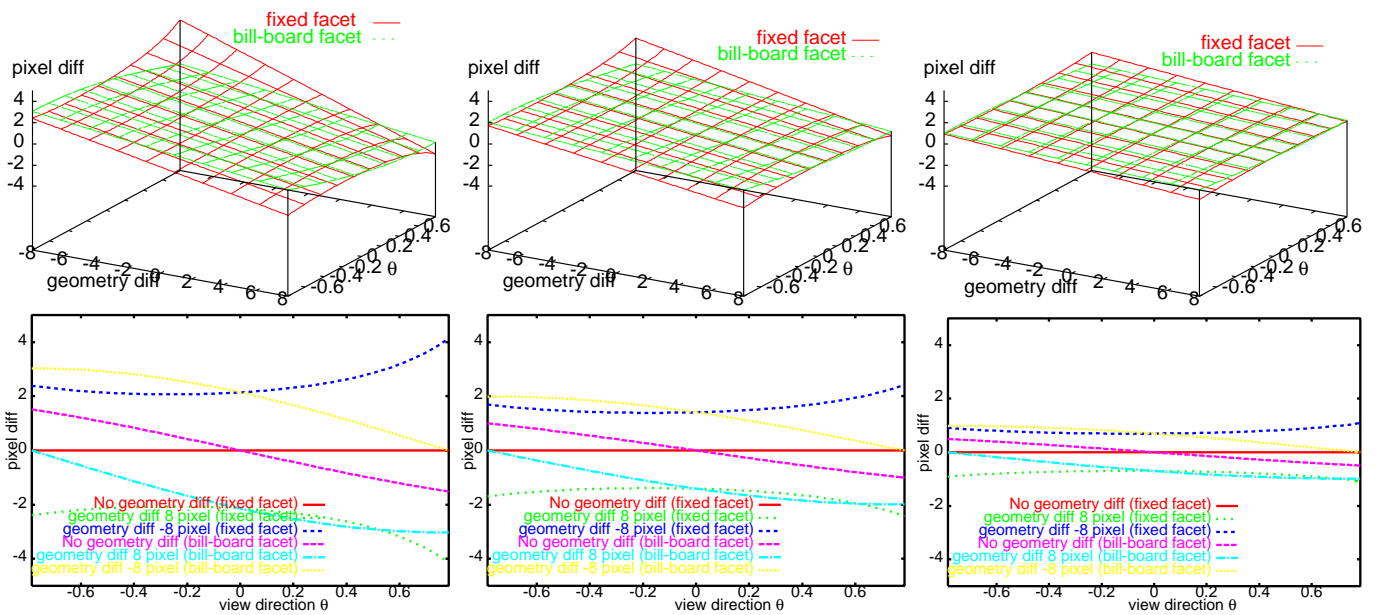


図 9: 左から順にカメラの撮影間隔 30°, 20°, 10°の時の画素の歪みの変化