２次元円筒ジオメトリ画像による
3次元モデルの形状表現と検索への応用

２００６年9月

白 井　啓 一 郎

目 次

第1章

序論

1.1 研究の背景

3次元コンピュータグラフィックス（以下3次元 CG）は，フォトリアリスティックな表現をコンピュータを用いて再現する手法として 1980 年代初頭から映画などにおいて使用され始め，マルチメディア時代と言われる今日では，コンピュータの高速化や家庭への普及にともない，ゲームなどのエンターテイメント，更にはインターネット上でのウェブショッピングなどにも使用され，ますます身近なものとなっている．

この3次元 CG において最も重要な役割を果たすのが，物体を表現する3次元モデル（3D モデル）であり，形状を表現するだけでなく，陰影の表現，テクスチャマッピングなど，フォトリアリスティックな表現を行うのに欠かせない処理がこの 3D モデルを介して行われる．また，映画やゲームなどの一つのシーンを作成するにあたっても，数十から数百体の 3D モデルが使用されることになる．近年では，より高品質な表現が要求され，より多くの緻密な 3D モデルが用いられる傾向がある．このようなモデルを扱う上で大きな問題となるのが，その膨大なデータ量であり，ローカルデバイスからの読み込みであっても数十秒を要し，更にネットワークを介した伝送を行う場合はその伝送時間も増大する．そのため，ネットワーク上で 3D モデルの共有を行う場合，動画像ストリーミングで行われているデータのプログレッシブ表現や圧縮符号化などの処理が，3D データに対しても必要となる．このような技術を必要とするものとしてはカーナビゲーションシステムなどがあり，DVD-ROM からの素早いデータの読み込みと表示が必要とされるだけでなく，年月の経過などに伴う風景の変化を正確に表示するためには，ネットワーク上のデータベースからの最新のデータを受信して表示することも必要とされる．

また近年では，計測技術の発展に伴い，建築物や彫刻などの重要文化財の３次元形状をレーザーレンジファインダなどの測定器で計測し，その３次元形状情報を修復や閲覧に役立てようとする試みが世界各国で行われている．例えば，ドイツのドレスデン聖母教会は 3D 技術

を利用して破片の状態から建物の復興が行われている．このような３次元形状の取り込みは通常，レンジファインダと呼ばれるレーザー測定器によって行われ，レンズから物体表面までの距離が格子状に取得される．ただし，その精度は通常の３次元 CG で扱われるモデルの精度に比べて遙かに高く，わずか数十センチ四方で数十メガバイトから数百メガバイトの情報量をもつ 3D モデルとなる．このようなモデルではデータ圧縮以前に表示そのものが困難となる．そのため，閲覧時の解像度に合せて読み込むデータ数を変え，メモリの消費を抑えたり，CPU および GPU の計算量を抑えるプログレッシブ表現は必須となる．このような理由から，3D モデルに対するプログレッシブ性を備えた圧縮符号化の研究が現在盛んに行われている．

上述した 3D モデルに対するプログレッシブ性を備えた圧縮を実現させる有効な手法としては，3D モデル上のジオメトリ情報を２次元画像を用いて表現するという手法が注目されており，３次元頂点座標情報を画素値に記録したこのジオメトリ画像と呼ばれる画像からは，もとの 3D モデルの形状に近いモデルを復元することができる（図 1.1，図 1.2 参照）．通常の 3D モデルを構成するメッシュは，実際には頂点座標情報の他に，隣接する頂点同士の接続を表す“ 接続情報 ”を必要とするため，このように頂点座標情報のみを記録した場合は，接続情報は失われてしまうことになる．しかし，ジオメトリ画像の画素は格子状に規則正しく並ぶため，格子上の画素を正方メッシュの格子と見なすことで格子状の接続情報を再構成することができ，正方メッシュとしてモデルを復元することが可能となる．これにより接続情報が一切不要となり，大幅に情報量を削減することができる．また，3D モデル上では不均一であった隣接頂点の配置が画像化により均一格子となり，正則性を持つため，時系列信号として扱えるようになる．これにより，既存の画像の圧縮および伝送技術の適用が可能となり，JPEG 画像のように 3D モデルを扱うことが可能となる．図 1.3 はジオメトリ画像を用いた 3D モデルのプログレッシブ表現の例を示したものであり，このように 3D モデルを一度画像化することで，単なる画像サイズの変更で，複雑な 3D モデルの形状をたやすく変更することが可能となる．

3D モデルをジオメトリ画像へと変換する手法としては，円筒投影法を用いる手法と，メッシュパラメータ化を用いる手法が知られている．円筒投影法を用いる手法としては [Tsai et al. 2002, Okuda et al. 2003, Lee et al. 1995] があり，この手法は，モデルを包含するように配置した円筒スクリーンに対して３次元情報を投影し，この円筒を切り開くことでジオメトリ画像（パノラマ状の距離マップなど）を取得する（図 1.1 参照）．したがって，得られる画像はグレースケールとなる．ただし，投影の際に，円筒面から見えないオクルージョン領域の情報は失われるため，軸対称の形状を得意とするなど，適応できるモデルは限定される可能性がある．しかし，顔形状などのように古くから円筒メッシュを用いて表現さ

れるモデルも多く，このようなモデルであれば良好な形状を復元できるジオメトリ画像へとモデルを高速に変換することができ，高速表示用の簡略表現に非常に適した手法といえる [Lee et al. 1995].

一方，メッシュパラメータ化法としては，[Gu et al. 2002, Sander et al. 2003, Yoshizawa et al. 2004, Hoppe and Praun 2005] があり，この手法は，3D メッシュを引き伸ばすことで平面メッシュへと変換し，このメッシュ上の３次元情報を均一格子でサンプリングして画像の画素値として保存することでジオメトリ画像を取得する（図 1.2 参照）．この手法では円筒投影法のように失われる領域はないため，その復元精度は円筒投影法のものよりも高くなる．しかし，パラメータ化の問題は線型方程式を解く問題に帰着されるため，大規模なモデルに適用した場合には膨大な時間を要するという問題がある．また，メッシュの分岐などの接続性に対する条件があり，3D モデルをより細かいそれぞれが１枚のメッシュと見なせるチャートに分けてから処理を行う必要がある．

1.2 本研究の目的と概要

前節 1.1 で述べた背景のもと，本研究では形状の復元精度と画像化速度の両面において良好な結果が得られる 3D モデルのジオメトリ画像化法の実現を目的とする．そのため，高速性をすでに備えており，ある程度の復元精度をもつ円筒投影法をここでは取り上げ，その復元精度の向上と更なる高速化のため，そのアルゴリズムの大幅な改善を行う．

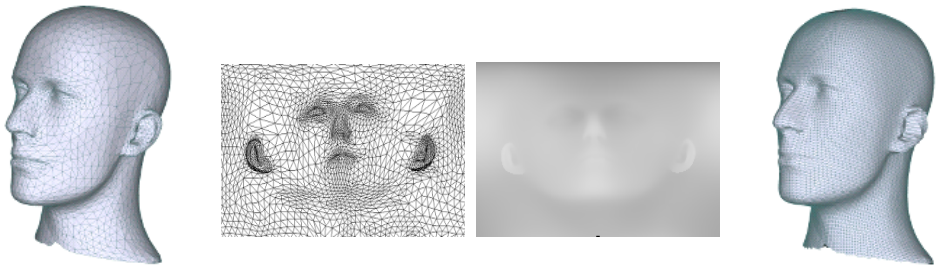
上述の目的のため，本研究では以下の異なるタイプの２種類の円筒投影法を新たに提案する．

階層的円筒レイキャスト法：形状の復元精度向上を目的とした円筒投影法 [Shirai et al. 2006a]

既存の手法で用いられる投影法である光線追跡法（以降レイキャスト法と呼ぶ）を用いるが，オクルージョン領域を捕捉するため，通常は１回のみ行われる投影処理，すなわち，円筒内部方向へ光線を放つことで行われる形状検出を,“ 階層的 ”に“ 角度 ”を変えつつ行うことで，オクルージョン領域に光線を回り込ませる方法を提案する．

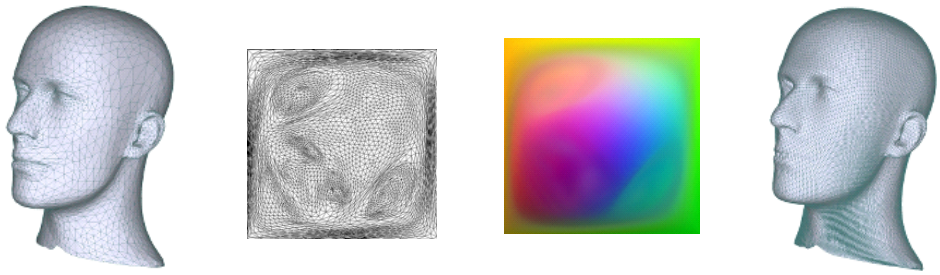
円筒 Z バッファ法：画像化速度の向上を目的とした円筒投影法 [Shirai et al. 2004, 白井 啓一郎 et al. 2006a]

レイキャスト法の問題として，アルゴリズムの計算量が挙げられる．そこで，より高速な投影法として知られる Z バッファ法の適用を行う．ただし，Z バッファ法は平面スクリーンに対する投影法であるため，今回，円筒ジオメトリ画像作成のため，円筒面に対して適用し



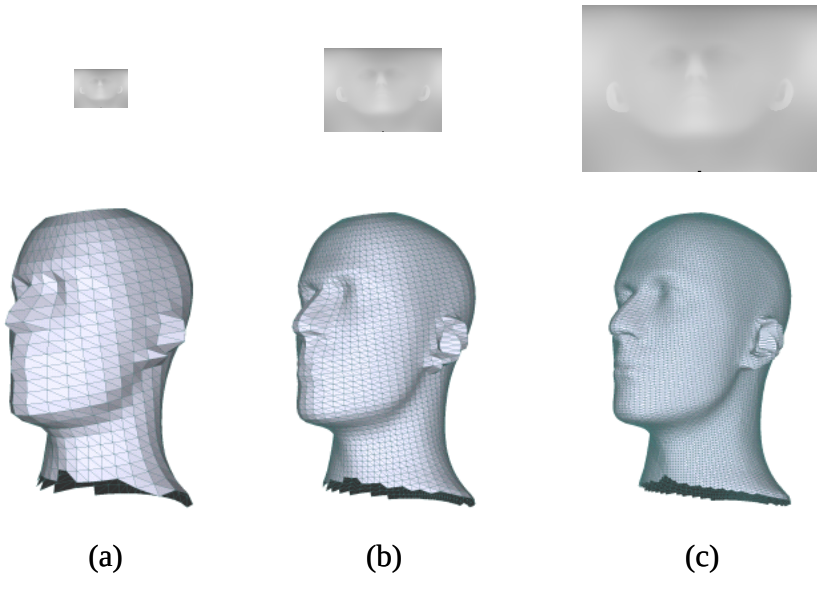
(a) (b) (c) (d)

図 1.1: 円筒投影法によるジオメトリ画像化．(a) オリジナルメッシュ．(b) 2 次元メッシュに変換された 3 次元メッシュ．(c) 円筒軸からの半径を記録したジオメトリ画像．(d) 復元モデル



(a) (b) (c) (d)

図 1.2: パラメータ化によるジオメトリ画像化．(a) オリジナルメッシュ．(b) 2 次元メッシュに変換された 3 次元メッシュ．(c) 3 次元頂点情報座標を記録したジオメトリ画像．(d) 復元モデル



(a) (b) (c)

図 1.3: ジオメトリ画像のサイズ変更による 3D モデルのプログレッシブ表現．(a) 32×32 のジオメトリ画像から復元したモデル．(b) 64×64 のジオメトリ画像から復元したモデル．(c) 128×128 のジオメトリ画像から復元したモデル

The model ”Mannequin head” is courtesy of Washington University.

た新しい円筒 Z バッファ法を提案する．

階層的円筒レイキャスト法はジオメトリ画像からの形状復元に重点をおく方法であり，円筒 Z バッファ法はジオメトリ画像の作成速度に重点をおく手法である．両手法の投影速度と形状復元精度はトレードオフとなるが，円筒ジオメトリ画像を利用するアプリケーションの用途に合わせて使い分けることでアプリケーションの性能を向上させることができると考えられる．

実際に，提案した円筒投影法が形状の復元以外の目的に対しても有効であることを示すため，円筒ジオメトリ画像を利用した 3D モデルの検索法を提案する．

円筒ジオメトリ画像を用いた 3D モデル検索法：[Shirai et al. 2006a]
類似した形状を持つモデルの円筒ジオメトリ画像は類似する傾向がある．そのため円筒ジオメトリ画像同士のマッチングにより，類似 3D モデルを検索することが可能となる．検索自体は画像検索となるが，円滑な検索を行うためにはユーザーが保持する 3D モデルを“ 高速 ”かつ“ 精度 ”よくジオメトリ画像へと変換することが要求される．この検索精度と速度を既存の検索法と比較することで，提案法の有効性を示す．

また，円筒投影法とは異なるが，同じく 3D モデルのプログレッシブ性を備えた圧縮を可能とする 3 次元メッシュ・パラメータ化法についての提案も行う．提案する手法は復元形状の精度を更に向上させるジオメトリ画像の作成を可能とし，上述した円筒投影法によって得られるジオメトリ画像と同様，この画像を用いた様々な応用が期待できる．

頂点密度均一化による 3D メッシュ・パラメータ化法：[白井 啓一郎 et al. 2006b, Shirai et al. 2006b]
3D メッシュ・パラメータ化をモデルの形状圧縮に使用する場合，パラメータ化後の 2D メッシュに疎密を生じさせないことが重要となる．そこで，各頂点周りの 2D メッシュの密度を 3D メッシュ形状の複雑さに合わせて柔軟に拡げる手法を提案する．3 次元形状が複雑な箇所に対応する 2D メッシュをより拡げることで，2D メッシュを均一サンプリングする際に，より多くのサンプルを取得することができ，復元時の形状が細部にわたり鮮明となる．

1.3 本論文の構成

本論文は，前節で述べた目的に基づいて２種類の円筒投影法の提案を行ったものであり，本章を含め，７章から構成される．また，付録を設け，本文中で省略した手法をここに示す．

第２章では，まず，本論文において必要となる 3D モデルや，その形状を実際に表すメッシュのデータ構造について説明する．次に，既存の円筒投影法の概要を述べる．まず，円筒投影法の概要を示し，座標系を定義する．次に，円筒面からのオクルージョン領域の欠落による劣化の度合いを示し，改善すべき問題点をあげる．

第３章では，形状の復元に重点をおいた円筒投影法を提案する．この方法はレイキャスト法を階層的に行うものであり，階層的円筒レイキャスト法と呼ぶ．ここではその光線の射出点の決定の仕方，および，射出の方向の求め方を示す．射出の方向には特に注意を払う必要がある．

第４章では，投影速度に重点を置いた円筒投影法を提案する．この方法はレイキャスト法の代わりに新たに Z バッファ法を円筒投影に適用するものであり，円筒 Z バッファ法と呼ぶ．適用の際に，投影面が円筒となることに起因する様々な問題が生じるため，この解決法を示していく．

第５章では，円筒ジオメトリ画像の応用例として，円筒ジオメトリ画像のマッチングを用いた高速な 3D モデルの検索法を提案する．この手法は円筒投影法の投影速度と主成分分析を利用した高速な画像マッチングの利点を組み合わせたものであり，既存の検索法と同程度の精度の検索を約 500 倍の速さで行うことができる．

第６章では，パラメータ化法を用いたジオメトリ画像化と形状復元について示す．前章までの手法は３次元モデルを単色の画像で保存するものであるが，この手法は x, y, z 座標を３色のカラー画像で保存するものである．この場合，情報量は増加してしまうが，より詳細な形状の復元が可能であり，この画像を用いた様々な応用も期待できる．

第７章では，本研究における全体的な成果をまとめる．

第2章

諸定義と従来法

2.1 はじめに

本章では，まず，本論文において必要となる数式表現，および，3D メッシュのデータ構造について説明を行う．次に，[Tsai et al. 2002，Khan et al. 2004，Okuda et al. 2004，Lee et al. 1995] で用いられている従来の円筒投影法について述べる．ここではまず，円筒投影法のアルゴリズムを座標系などの説明と共に順を追って示していく．次に，得られた円筒ジオメトリ画像からの形状復元例を示し，その復元形状の問題点と，円筒投影法の改善すべき点をまとめる．

2.2　記号表現

まず，本論文内で扱う主な記号表現を以下のように定義する．

- 変数は斜体で表し，行列やベクトルを太字で表す．特に行列を大文字を用いて表す．例えば頂点座標などを $\mathbf{p} = [x, y, z]$ と表す．
- 頂点座標やベクトルの各要素を指定する場合はピリオドを用いてベクトルの記号を含めて表す：$p.x$．
- 頂点の集合など，集合を $\{\cdot\}$ とアルファベットのカリグラフ体を用いて表す．例えば頂点座標の集合を $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ と表す．
- 辺の長さなどを表すためにノルム $\|\cdot\|$ を用いるが，基本的にはユークリッドノルム（L_2 ノルム）を表すものとする．すなわち，ベクトル $\mathbf{v} = [x, y, z]$ の長さを，簡単に $\|\mathbf{v}\| = \sqrt{v.x^2 + v.y^2 + v.z^2}$ と表す．

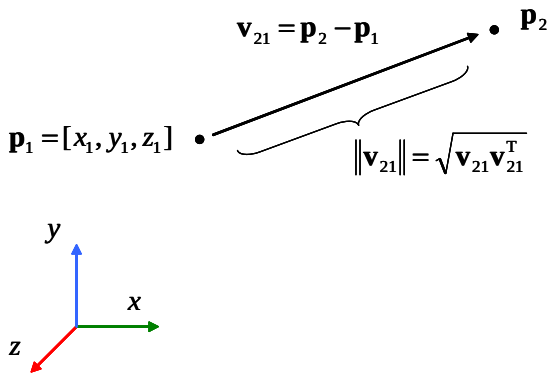


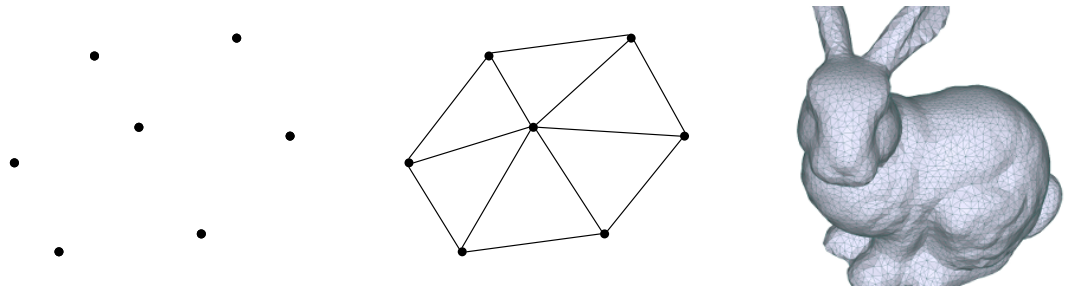
図 2.1: 座標やベクトルの数式表現

その他の数式記号についての諸定義は，各章の冒頭および本文中にて説明する．

2.3 メッシュの定義

2.3.1 データ構造

3D メッシュは 3 次元空間に任意に配置された頂点と，それらを接続してできる面によって形作られる．図は 2.2 その様子を示したものであり，(a) の頂点をつなぎ合わせることで，(b) の面が形成され，この集合として，(c) の 3D モデルが形作られる．



(a) (b) (c)

図 2.2: 3D メッシュ．(a) 頂点．(b) 面．(c) 3D モデル．

The model ”Stanford bunny” (c) is courtesy of Stanford University.

データとしてこれらを扱う際は，頂点の持つ座標情報や法線ベクトル情報を“ 頂点情報 ”として扱い，どの頂点同士を結ぶかという情報を“ 接続情報 ”として扱う．

- **頂点情報：** 頂点座標を $\mathbf{p} = [x, y, z]$，法線ベクトルを $\mathbf{n} = [x, y, z]$ と表す．x, y, z の各要素を指定する場合は $p.x$，$n.x$ と書き表して区別する．
- **接続情報：** 各頂点にインデックス番号を付け，メッシュの“ 辺 ”や三角形の“ 面 ”をこの番号の組合せで表す．例えば頂点番号 i, j, k などを用いて，辺を $\{i, j\}$，面を $\{i, j, k\}$ と表す．

表 2.1 は上述した表現法を用いて図 2.2 (c) に示したモデル（Stanford bunny）を表したものであり，この形式は 3D モデルの標準的なフォーマットの多くで採用されている[*1]．

2.3.2 トポロジー

前項では基本的なメッシュの構造を示した．しかし，実際には頂点の接続の仕方によってもメッシュが分類され，その処理も異なったものとなる．例えば，簡単なものであれば，すべての面が三角形で構成された三角形メッシュ，四角形やそれ以上の多角形が混在したポリ

[*1] Ply 形式（論文などのプログラムで主に用いられる），Obj 形式（Wavefront 社），Wrl 形式（VRML）などがある

表 2.1: 3D メッシュモデルのデータ構造の例．まず頂点座標情報が x, y, z の順に記述され，次に何番目の頂点座標を接続するかを指定した接続情報が記述される．

Bunny.obj

```
# x, y, z 座標
v -0.036872  0.127727  0.004409 # 1個目
v -0.045361  0.128854  0.001145 # 2個目
v -0.069007  0.151612  0.036602 # 3個目

similar...

# 面の持つ頂点番号
f  6195  6194  6118 # 1個目 <-- 三角形
f  5829  5118  5806 # 2個目
f  5806  5118  5771 # 3個目
f  6206  5234  3892 1392 # 4 個目 <-- 四角形

simillar...
```

ゴンメッシュがある．複雑なものであれば，CAD[*2]などで使用される 6 面体メッシュなどもある．また，球やトーラス[*3]のように空間的に閉じたメッシュもあれば，枝分かれをしたメッシュも存在する．このため，メッシュの定義が必要となる．

定義：　本研究が対象とするメッシュを以下のように定義する．

- メッシュは厚みをもたず，全ての面が三角形で構成される単純な“ 三角形メッシュ ”を扱う．なお，三角形面を以後，単に“ 三角形 ”と呼び，記号で表す場合は △ を用いる．また，以降，3 次元メッシュと 2 次元メッシュを扱うことになるが，それぞれ $\triangle_{3D}$，$\triangle_{2D}$ と書き表す．

- 空間的に閉じているか，枝分かれをしているかは基本的には問わない．

[*2] Computer Assisted Drawing：コンピュータを用いた製図システム
[*3] ドーナッツ型

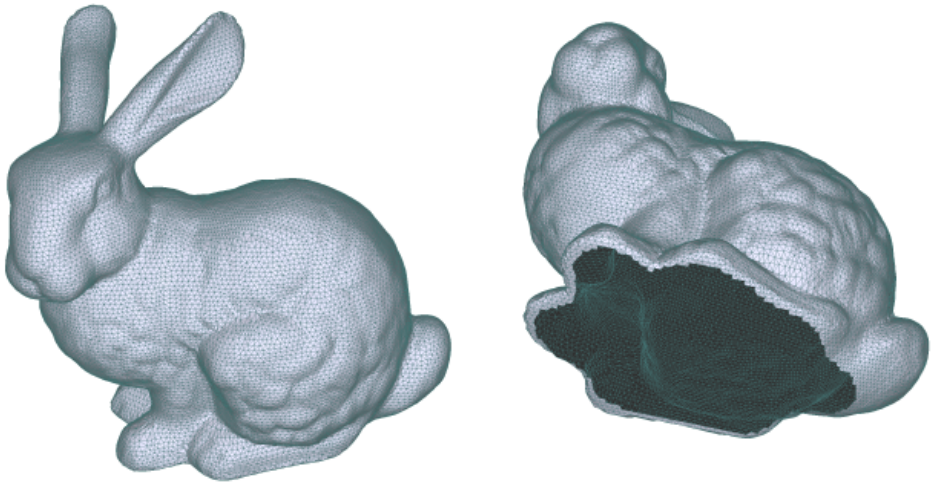
図 2.3 は取り扱うメッシュデータの例を示したものであり，すべての面が三角形で構成されているのが分かる．

2.3.3 正則性

本論文では，各頂点が上下左右，右上，左下の計 6 つの隣接頂点と規則正しく結ばれた正則メッシュ（接続価数が一定なメッシュ）と，このような規則性を持たない非正則メッシュの両方を扱う．実験に用いるオリジナルモデルは，マニュアルで作成されたり，光学測定器で取得したメッシュをつなぎ合わせて作成されるため，基本的には非正則メッシュとなる．一方，次節 2.4 で述べるジオメトリ画像から復元されるモデルは正則メッシュとなる．図 2.4 はこの 2 種類のメッシュを示したものであり，(a) が非正則メッシュ，(b) が正則メッシュを表している．

メッシュの特徴： 2 種類のメッシュはそれぞれ特徴を以下の特徴を持つ．

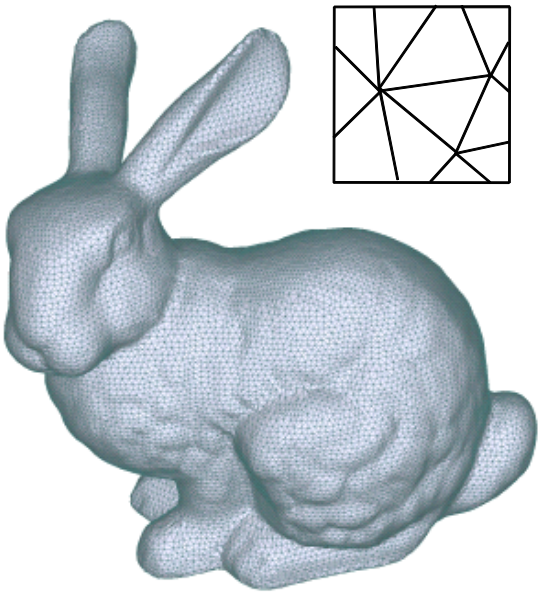
- **非正則メッシュ：** 頂点を自由に配置でき，モデルの詳細な形状を表現するのに適している．ただし，隣接頂点数が一定でないため，変形処理などの各種信号処理を施しにくいという欠点がある．
- **正則メッシュ：** 隣接頂点数が固定され，頂点配置も規制されるため，形状は幾分劣化し，丸みを帯びる．しかし，規則的に並んだ頂点に対して信号処理を行いやすいという大きな利点がある．



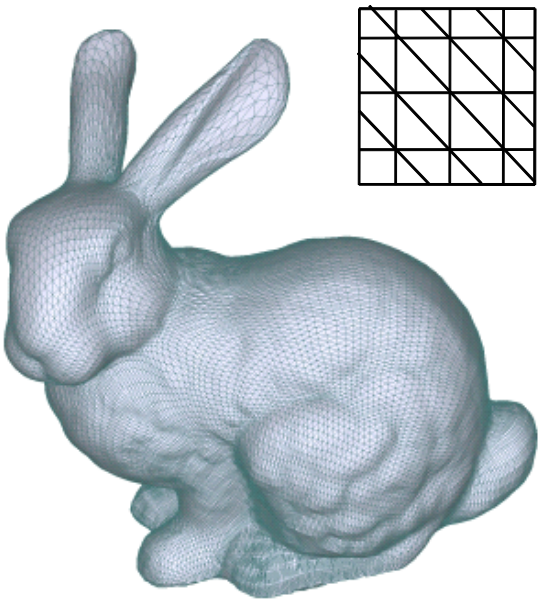
(a) (b)

図 2.3: 本論文内で主に扱う三角形 3D メッシュの例．メッシュはすべて三角形パッチの集まりから構成される．

これらの特徴からは，形状表現と信号処理を行う容易さの間にトレードオフの関係があるのが分かる．



(a)



(b)

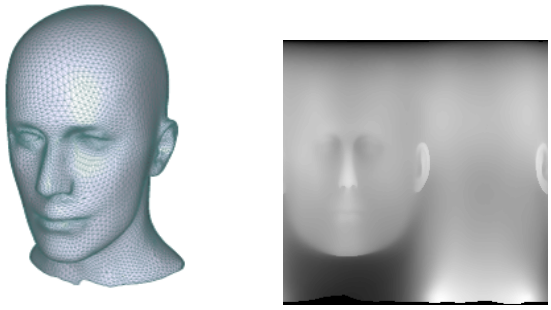
図 2.4: (a) 非正則メッシュと (b) 正則メッシュ．正則メッシュでは各頂点が上下左右，右上，左下の計 6 つの隣接頂点と規則正しく結ばれている．

2.4 円筒投影法

本節では円筒投影法について説明する．まず簡単に，円筒投影法を行う目的である 3D モデルの円筒ジオメトリ画像化と，得られたジオメトリ画像からの形状復元について述べた後，従来法のアルゴリズムを座標などの定義を行いながら説明していく．

2.4.1 円筒ジオメトリ画像化

円筒投影法は，3D モデルを包含するように配置した円筒スクリーンに対して，3D モデル表面のもつ頂点情報，法線情報，円筒軸からの半径などのジオメトリ情報（形状に関係する情報）を投影する手法であり，この円筒を切り開くことでジオメトリ画像を得ることができる．ここでは円筒投影法によって得られたジオメトリ画像を“ 円筒ジオメトリ画像 ”と呼び，このように 3D モデルをジオメトリ画像に変換することを“ ジオメトリ画像化 ”と呼ぶ．図 2.5 は 3D モデル Mannequin head(a) とその円筒ジオメトリ画像を示したものである (b)．このジオメトリ画像に対してウェーブレット変換などを施すことで，プログレッシブ性を備えたデータの圧縮や伝送が可能となる．また，このジオメトリ画像化により 3 次元メッシュの持つ頂点接続情報は完全に失われることになるが，次項で接続情報の再構築方法について述べる．



(a) (b)

図 2.5: オリジナルモデル (a) とそのジオメトリ画像 (b)（モデルの円筒軸からの半径を記録したもの）．

2.4.2 ジオメトリ画像からの形状復元

得られた円筒ジオメトリ画像を正方格子でサンプリングし，各画素を縦横斜めの近傍画素とつなげば，図 2.6 に示すような正則メッシュを生成することができる（左右はループしているので，円筒状のメッシュとなる）．一方，ジオメトリ画像に記録したモデル表面の円筒

軸からの半径から，同図 (c) に示すようなオリジナルモデルの形状に近い円筒メッシュを復元することができる．

2.4.3　適切なジオメトリ画像の解像度とデータ量

モデルのもつ座標情報と接続情報（三角形情報）の比率は約 1 : 2 から 1 : 3 であり，ジオメトリ画像化による近似ではこの接続情報を切り捨てても形状を復元できるため，この時点で 1/3 の情報量となる．ただし，オリジナルモデルと同一頂点数で形状復元を行っても，実際には形状は劣化するため，元の 4 倍〜6 倍の頂点数を必要とする．この時点で情報量は元の $6 \times (1/3) = 2$ 倍となり増加する．しかし，ジオメトリ画像を圧縮する際には，接続情報を持たないため，圧縮をすべて非可逆圧縮を用いて行うことができる[*4]．また，時系列信号であるので近傍画素との相関を利用した圧縮を行うことができる．この結果，例えば図 2.3 の約 2MB の Stanford bunny を約 40KB のジオメトリ画像から復元できるようになる．この場合，情報量は元モデルの 10%〜20%となる．

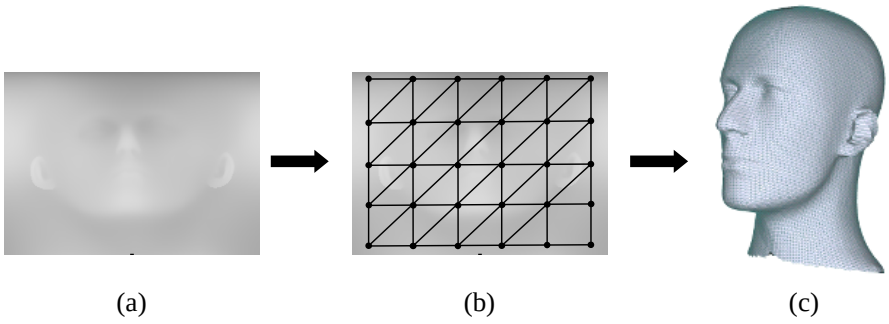


図 2.6: ジオメトリ画像からの正則メッシュの生成．(a) 円筒ジオメトリ画像．(b) 近傍画素を接続による正則メッシュの生成．(c) 復元された円筒状の正則メッシュ．

[*4]非可逆圧縮とは，圧縮前の状態にデータを完全に戻すことはできないが，高い圧縮率を得られる圧縮方式．一方，接続情報が欠落した場合，メッシュは破綻するため，この方法よりも圧縮率が劣る可逆圧縮で圧縮する必要がある．

2.5 投影アルゴリズム

投影処理は大きく分けて，(1) 円筒とモデルの位置合わせ（前処理），(2) レイキャスト法（光線追跡法）による投影，(3) ジオメトリ画像への 3 次元情報の記録の手順で行われる．

2.5.1 円筒とモデルの位置合わせ

円筒の座標系の定義： まず，図 2.7 (a) のように 3D モデルを円筒内に納めるために，3D モデルと円筒の座標合せとスケール合せが必要となる．ここでは簡単のため，円筒をワールド座標系の原点上に固定し，そのサイズを半径 1，高さ 1 とする．また，この円筒表面のスクリーン座標系を (θ, y) を用いて表し，その表面は図 2.7 (b) に示すように正方格子状に分割されているものとする（以降この格子を単に画素と呼ぶ）．

主成分分析によるモデルの位置合せ： 円筒内部にモデルが納まるように，モデルを移動し，リサイズすることで位置合わせを行う．また，モデルの姿勢を正すため，主成分分析を用いてモデルの主軸を求め，その主軸と円筒の y 軸が一致するようにモデルに回転処理を施す．この回転処理については付録 A.2 を参照されたい．

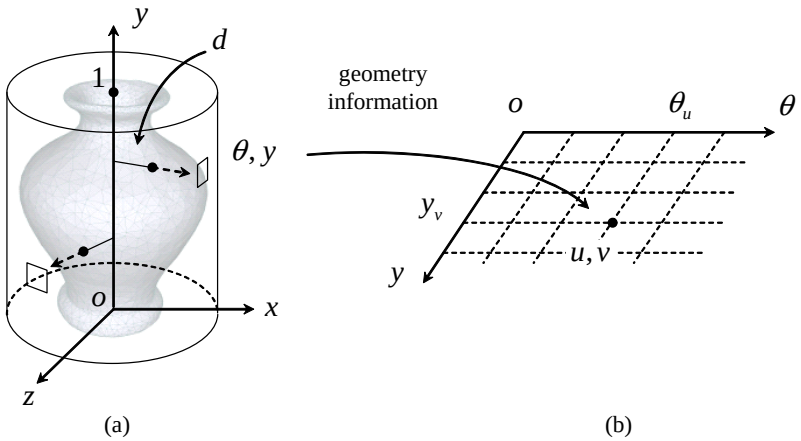


図 2.7: 円筒投影によるモデル上の点情報の投影の様子と，展開されたジオメトリ画像との座標の関係

2.5.2 レイキャスト法

円筒投影の定義： 円筒面への投影は，通常の平面ディスプレイへの投影のように放射状ではなく，円筒軸から“ 軸に対して垂直 ”に行うものとする．すなわち，この投影により空間

中の点の座標 (x, y, z) はスクリーン座標 (θ, y) へ

$$(\theta, y) = (\tan^{-1}(z/x), y) \tag{2.1}$$

と変換される（y 成分の値は変わらない）．実際には，円筒面の画素に対応するモデル上の点のみが必要なため，画素を垂直に通過する光線とモデルとの交点を求め，画素に対応付ければよいことになる（図 2.7 (a) 参照）．

レイキャスト法：　画素と対応するモデル上の交点を探索する手法がレイキャスト法であり，図 2.8 (r1〜r3) はレイキャスト法の手順を図示したものである．その手順は，

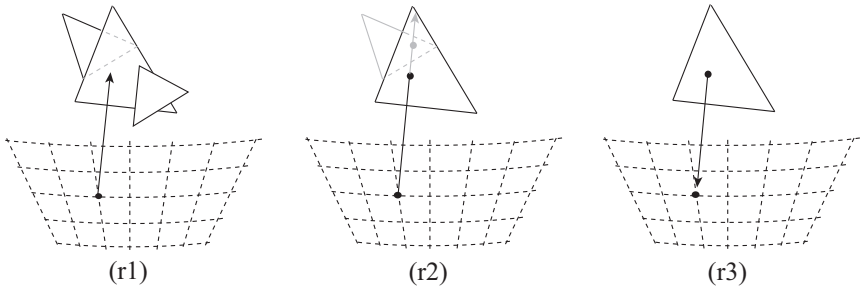


図 2.8: レイキャスト法によるスクリーン画素への交点情報の対応付け

(r1) 画素から円筒軸方向に光線を放ち，

(r2) 光線と交わるメッシュ上の三角形を探す．入り組んだモデルでは交差する三角形とその交点も複数個存在するが，

(r3) 交点の軸からの距離を比較し，最も距離の大きい交点，すなわち投影面から最も近い交点を探し出す．

この操作を全画素に対して行う．これにより 3D モデル全体の投影像が得られる．

ただし，光線がどのメッシュ上のどの三角形と交差するかは事前には分からないため，手順 (r2) での光線との交差判定を多数の三角形と行う必要があり，多くの計算量を必要とする．このため，通常は三角形を空間的にグループ化させることで交差判定の回数を減らす“ 空間分割法 ”を用いる必要がある [Glassner 1984, Quail]．分割方法は様々であるが，本研究ではモデルの存在する空間を扇形に空間分割した．

2.5.3 ジオメトリ画像への 3 次元情報の記録

円筒ジオメトリ画像： 表面が格子状に分割された円筒は，平面画像が湾曲したものとみなすことができる．この画像をここでは円筒ジオメトリ画像と呼び，その画素番号を (u, v)，その円筒面上での座標を (θ_u, y_v) として表す．この画素値として，レイキャスト法で得た交点のもつジオメトリ情報（ここでは交点座標の円筒軸からの半径）を記録する．その他にも，法線マッピング[*5]に用いる法線情報などを必要があれば記録する．

円筒軸からの半径の記録： 交点の円筒面からの距離 d を記録する場合，円筒と交点が存在するモデル表面の位置関係によっては符号を分ける必要が生じる．すなわち，図 2.9 のように入り組んだモデルでは，(a) モデルを円筒面から見た場合（ここでは赤線の切断面），(b) 位置によっては円筒面から直に見える場合（a,a' の関係）と円筒軸を介して見える場合がある（b,b' の関係）．(c) 円筒軸を介する場合はモデルの投影像が軸で反転するため，(d) それぞれ半径 d の符号を正 $d > 0$ と負 $d < 0$ に設定する必要がある．なお，円筒の半径を 1 と設定したため，距離 d の取り得る範囲は $-1 < d < 1$ となる．

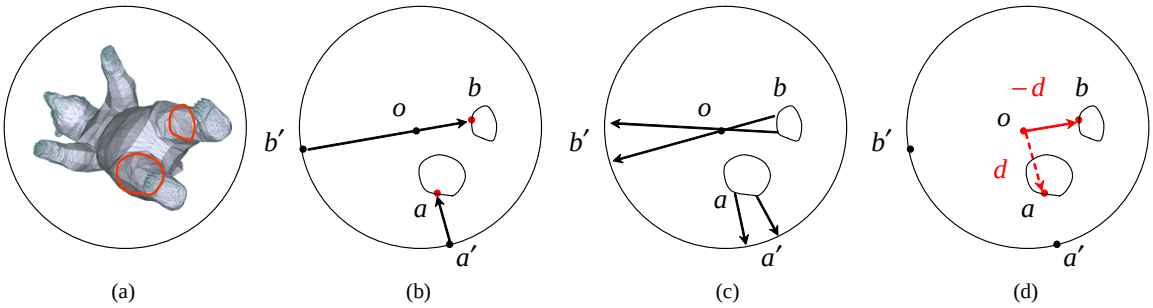


図 2.9: モデルの円筒軸に対する位置による半径のとり方．(a) 形状が複雑なモデルとその切断面（赤線）．(b) 可視面と円筒軸の関係．(c) 円筒軸によって反転する投影像．(d) 符号による位置関係の判別

ジオメトリ画像からの形状復元： 円筒ジオメトリ画像からもとの 3 次元形状を復元する場合，ジオメトリ画像に記録した距離 d と座標 (θ, y) を用いて，新たな頂点座標が

$$(x, y, z) = (d\cos\theta, y, d\sin\theta) \tag{2.2}$$

として得られ，各画素を近傍画素と接続することで，規則的な接続を持つ正則メッシュが形作られる．

[*5]バンプマッピング（bump-mapping）としても知られる．テクスチャマッピングがモデルに画像を貼り付けて色や模様を表現するのに対して，法線マッピングは形状の局所的な傾きを表す法線情報を貼り付けることで，モデルの細かい凹凸を表現する．

2.6　投影結果と問題点

円筒投影法を用いて 3D モデルをジオメトリ画像化した結果とジオメトリ画像からの形状を復元した結果を図 2.10 と図 2.11 に示す.

画像サイズの設定： ジオメトリ画像のサイズは，復元モデルの三角形数がオリジナルモデルの三角形数とほぼ同等となるように設定した．すなわち，上下左右の近傍画素を接続して正方メッシュとした後，対角画素を接続して三角形メッシュとするので，2× 縦 × 横 = 三角形数，となるように設定してある．得られるジオメトリ画像はモデルを周囲から見た図をパノラマ状に合成したような画像となっているのが分かる．このジオメトリ画像はモデルの円筒軸からの半径を記録したものであり，画素の色が明るくなるほど手前を表すことになる．また，色が完全に黒である部分はジオメトリが存在していないことを表している．

復元された形状であるが，図 2.10 のモデルは見た目が良好なのに対して，図 2.11 のモデルはそうでないことが分かる．これは主に円筒から見えないオクルージョン領域の欠落が影響しており，図 2.10 のように形状が筒状に近いモデルではオクルージョンによる欠落箇所も少ないが．図 2.11 のような形状の分岐や凹領域を多く含むモデルではオクルージョン領域が多いため，単純にメッシュを接続するとオクルージョンとなる領域に，レース状に見える実際には存在しないジオメトリが生じる．もし，オクルージョン領域を接続したくなければ，ジオメトリ画像の画素値が大きく変わる部分がオクルージョン領域の境界となるので，閾値を設けて画素の接続を制限することでオクルージョン領域を穴として表すこともできる．しかしいずれにせよ，このようなオクルージョン領域の欠落は，見た目にも，データの価値的にも望ましいものではない.

2.6.1　問題点

既存の円筒投影法を形状復元に用いる場合の問題点としては

- **オクルージョン領域の欠落：**存在しないジオメトリが生じたり，穴が開いたりする.
- **レイキャスト法の計算量：**光線とメッシュの交点を求めるために，単一光線に対して複数の三角形との余分な交点計算が必要.

があげられる.

まず，オクルージョン領域を捕捉し，復元するためには，光線の角度を変えていくことが有効であると考えられる．ただし，角度を一括して変えた場合，当然ながら，別の領域がオ

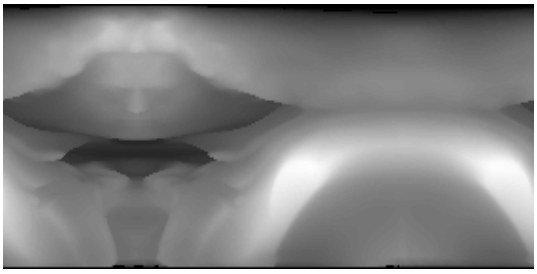
Igea

(a) Original (67K face)

Beethoven

(b) Original (80K face)

(c) Distance-map (260×130)



(d) Distance-map (280×140)

(e) Reconstruction (64K face)

(f) Reconstruction (76K face)

図 2.10: オリジナルモデルの円筒ジオメトリ画像と復元例 (1/2).

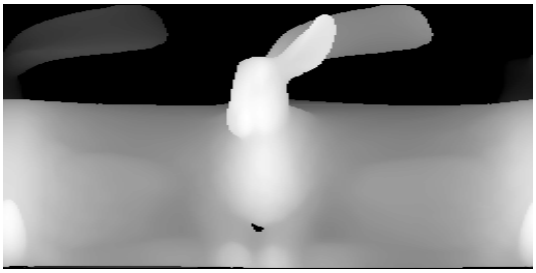
Models are courtesy of Cyberware Inc. (Igea), and Viewpoint Inc. (Beethoven).

Stanford bunny

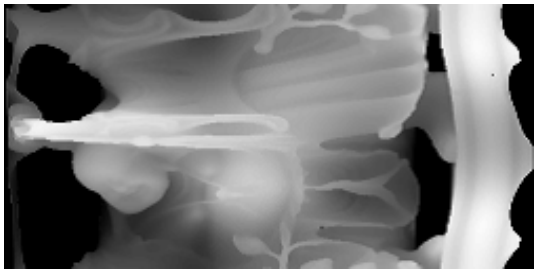
(a) Original (24K face)

Happy buddha

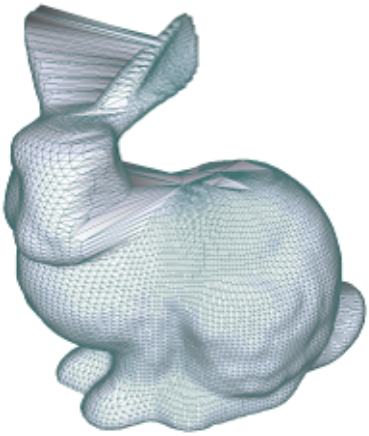
(b) Original (293K face)



(c) Distance-map (80×160)



(d) Distance-map (540×270)



(e) Reconstruction (25K face)

(f) Reconstruction (260K face)

図 2.11: オリジナルモデルの円筒ジオメトリ画像と復元例 (2/2).

Models are courtesy of Stanford University (Stanford bunny, Happy buddha).

クルージョンとなるため，本研究ではこの改善策として，階層的に徐々に光線の角度を変えて交点探索を行うレイキャスト法を提案する．この方法を次章 3 章にて述べる．

次に，レイキャスト法の計算量についてであるが，これは現在の 3 次元 CG で主に用いられている交点計算法や投影アルゴリズムを代用することが得策だと考えられる．本研究ではこの改善策として，Z バッファ法として知られる投影アルゴリズムの円筒への適用を提案する．この方法を 4 章にて述べる．

2.7　本章のまとめ

本章ではまず，メッシュのデータ構造や種類の定義を行い．次に，本論文がテーマとしている円筒投影法の基本的なアルゴリズムを示した．そのアルゴリズムは，円筒から垂直に光線を放ち，3D モデルとの交点を探すというレイキャスト法と呼ばれる投影法を用いるものであった．この手法の問題点として，円筒面から見えないオクルージョン領域の情報が欠落すること，レイキャスト法の計算量が多いことがあげられる．この 2 つの問題を解決するための提案手法を次章から述べていく．

第3章

オクルージョン領域の復元を目的とした階層的円筒レイキャスト法

3.1 はじめに

円筒投影法によって得られる円筒ジオメトリ画像から３次元形状を復元する場合，円筒面からオクルージョンとなるメッシュ領域が大きく欠落することが問題となる．円筒投影法が後述する 3D メッシュ・パラメータ化法（6 章参照）に比べ，高速に 3D モデルを画像化でき，大まかな形状を復元できるとはいえ，このオクルージョン領域の欠落は望ましいものではない．また，このオクルージョン領域の復元ができれば，円筒ジオメトリ画像を介した高速かつ精度のよい 3D モデルのデータ圧縮が実現できることになる．

本章では，このオクルージョン領域の復元を目的とし，従来法で用いられるレイキャスト法を“ 階層的 ”に光線の発射角度を変えながら行うことでオクルージョン領域を捕捉する手法を提案する [Shirai et al. 2006a].

3.2　概要と諸定義

レイキャスト法を用いて円筒上の画素から 3D モデル上の対応点を探す場合，図 3.1 のように窪んだ形状をもつ場所では，円筒から発射した光線が窪みよりも手前の表面で遮断されるため（左），窪み内部のオクルージョン領域の形状を捕捉することができず，復元形状が劣化することが問題であった（右）.

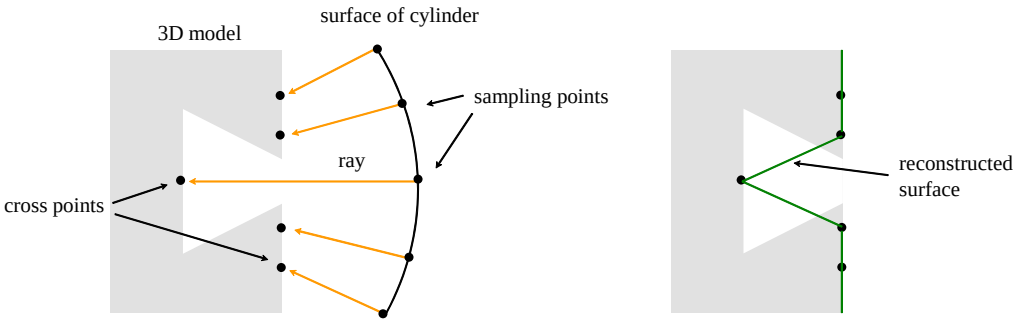


図 3.1: 光線追跡とその復元形状．(右) 灰色凹形状の図形がモデルの断面を表し，湾曲した線が円筒表面を表している．この円筒表面上に等間隔に並ぶ点は光線を発射する点（ジオメトリ情報をサンプルする画素）を表しており，このサンプリングポイントからモデル方向に垂直に光線が放たれ，モデル表面に交点が得られる．(左) 得られた交点座標をつないだ形状が復元時の形状となる．

3.2.1　光線の発射方法

オクルージョン領域に光線を回り込ませるため，ここでは図 3.2 に示すように，第一段階の光線追跡で得られた点において，隣接交点を結び，その中点から，更に 3D モデルの方向へ垂直に光線を飛ばし，交点があるかどうかを追加探索することを考える．このように，徐々に詳細な交点探索を繰り返し行うことで，図 3.1 ではクサビ型でしかなかった復元形状（実線部分）が，図 3.2 (a)(b) のように窪みの形状に近づいていくことになる．

3.2.2　画素値に保存するジオメトリ情報

今回ジオメトリ画像の画素に記録するジオメトリ情報は，光線の発射点から交点までの距離とした．復元の際には，半径を 1 とした円筒面上の格子位置から，この距離をもとに頂点を配置していくことになる．

3.2.3 階層とジオメトリ画像の解像度の関係

中点から光線を放っていくと，階層が上がるに従い，その解像度は 2 倍，4 倍と増えていくことになる．もし 256×256 の解像度のジオメトリ画像を得たい場合は，適当な初期解像度，例えば，32×32 の解像度から始め，階層を増やすことで，64×64，128×128 へと近づけていく．この間にオクルージョン領域が捕捉されていく．一方，従来法の場合，256×256 の解像度で 1 回の投影のみで行う．当然ながらオクルージョン領域は欠落することになる．

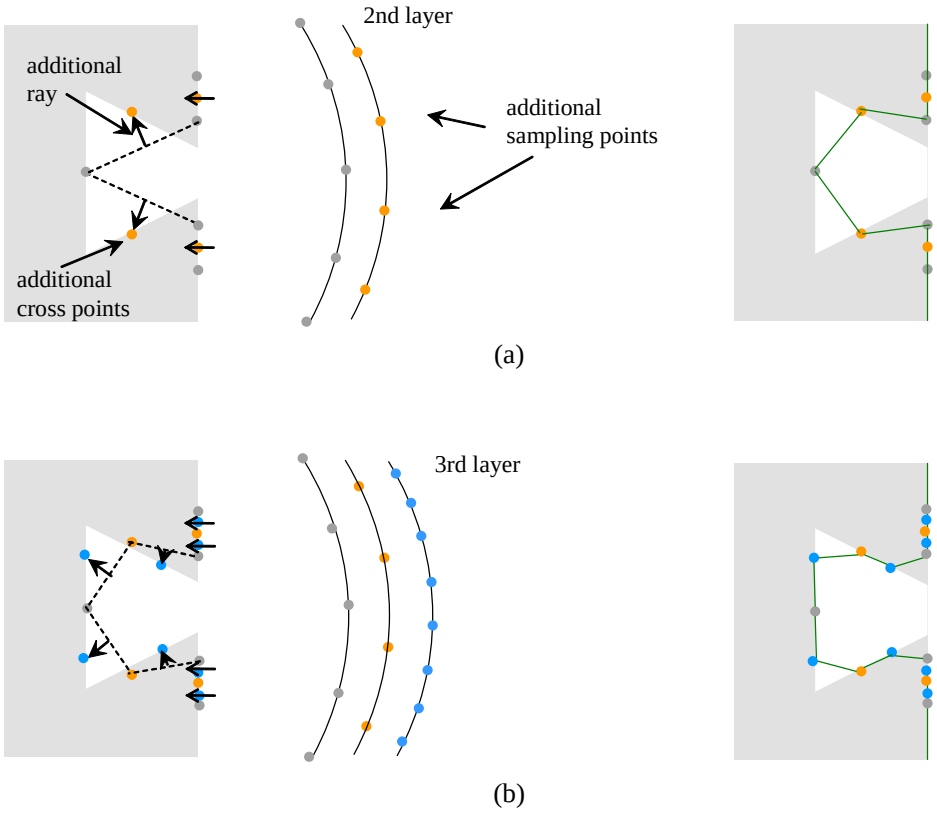


図 3.2: 階層的な光線追跡の様子．上から (a) 第二段階の光線追跡，(b) 第三段階の光線追跡．

3.2.4 多重解像度表現：データ構造の階層化

図 3.2 において，円筒面を 2 つ 3 つと増やしているが，これは各階層ごとにジオメトリ画像を分けて交点までの距離を記録していくということを表している．このように階層を重ねるごとに細かくなる変化を，倍の解像度で表していく表現方法は，信号処理の分野では多重解像度表現として知られている．このようなデータ構造にすることで，データにプログレッシブ性を持たせることが可能となる．

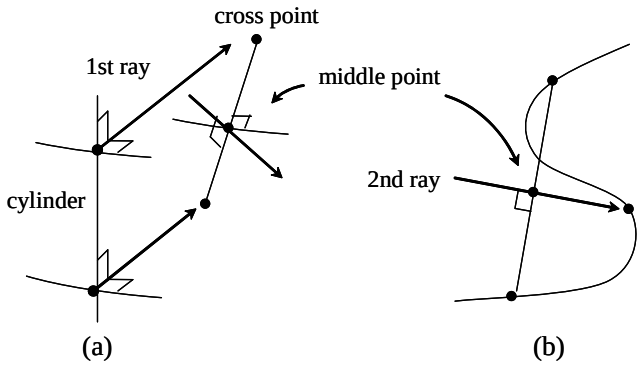


図 3.3: 垂直方向への階層的な光線追跡．(a) 3D 図．(b) 円筒を真横から見た図

3.3　階層的円筒レイキャスト法

本手法では交点計算の回数が多くなるため，まず，交点の計算に，高速な計算法として知られる Möller らの方法 [Möller and Trumbore 1997] を用いる．なお，C 言語のソースコードなども彼らの論文内に示されているので参照されたい．

3.3.1　光線の発射点と発射方向

ここでは著者による光線の分岐のさせ方について説明する．階層的に光線追跡を行う際，光線は前回求まった交点の中点から“ 垂直方向 ”に放たれるものとする（図 3.3 (b) 参照）．しかし，３次元で考えた場合，直線に対して垂直な方向は，この直線を法線とする平面上のすべての方向となる．このため，図 3.3 (a) のように水平方向に対しても垂直という拘束条件が必要となる．

しかし，２回目以降の光線追跡では交点の配置はこのようなグリッド状にはならないため，もう少し一般的な求め方を考える必要がある．また，分割を行う方向も，図 3.4 のように，水平方向 (a)，垂直方向 (b)，対角方向 (c) の３方向を考える必要がある．

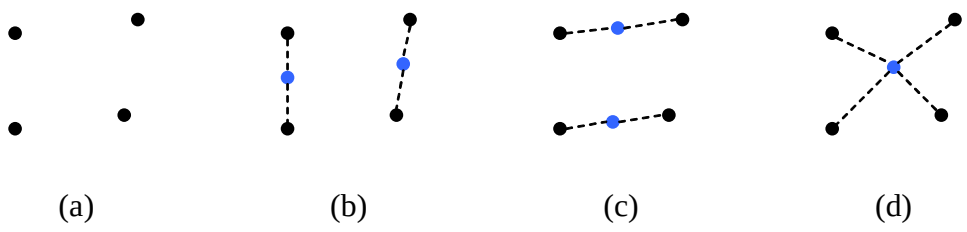


図 3.4: 分割方向．(a) 前回得られた交点．(b) 垂直方向への分割．(c) 水平方向への分割．(d) 対角方向への分割

光線の発射点と発射方向の求め方を説明するが，まず対角方向を求めた後に，水平方向と垂直方向を求めていく．図 3.5 と図 3.6 はこの手順を示したものである．

まず，対角方向の発射点と発射方向を次の手順で求める（図 3.5 参照）．

- **発射点：** ４つの交点の“ 平均点 ”[*1]を発射点とする．
- **発射方向：** この平均点と交点により４つの三角形の面を張ることができるので，各三角形の法線を求め，この法線を平均したベクトルの方向を発射方向とする．

次に，水平方向と垂直方向の発射点と発射方向を求める（図 3.6 参照）．手順は両方向とも同じであるため，ここでは垂直方向の求め方を示す．

- **発射点：** ２つの交点の“ 中点 ”を発射点とする．
- **発射方向：** 対角方向の処理で得られた発射点を２点加える（交点線分の両側の２点）．交点と加えた発射点を用いて２枚[*2]の三角形の面を張ることができるので，各三角形の法線を求め，この法線を平均したベクトルの方向を発射方向とする．

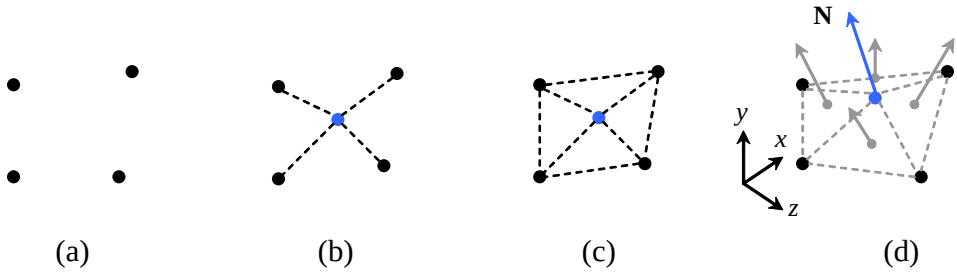


図 3.5: 対角方向への分割．まず，前回得られた交点を考え (a)，その平均点を発射点とする (b)．次に，交点と求めた平均点を用いて三角形を作成し (c)，これらの三角形の法線の平均を発射方向とする (d)．

3.3.2 発射点とモデルの位置関係による発射方向の変更

光線の発射方向に関しては更に条件をつける必要があり，新たな発射点が円筒から見てモデルの内側にあるか外側にあるかで，次のように発射方向を変える必要がある．

- 発射点がモデル**内部**の場合： 法線方向と**順方向**に光線を放つ．
- 発射点がモデル**外部**の場合： 法線方向と**逆方向**に光線を放つ．

[*1] 4 つの点であるので，ここでは中点，すなわち，線分上の中央の点という言葉を避けた

[*2] ４枚ではなく２枚なのは，発射点としてとった中点が交点を結んだ直線上に存在するため，図 3.6 (c) の場合，水平方向には折り目が生じないためである．

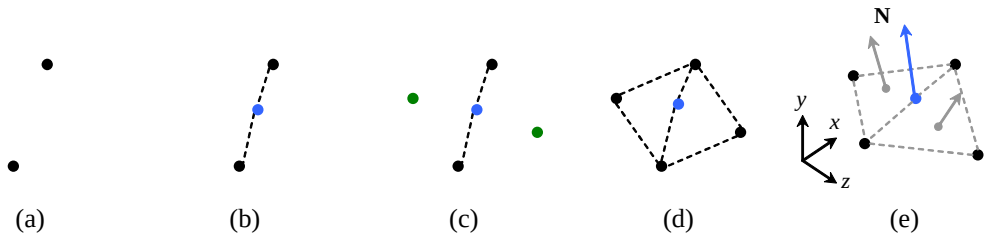


図 3.6: 水平・垂直方向への分割．まず，前回得られた交点を考え (a)，その平均点を発射点とする (b)．次に，法線を求めるが，このままでは三角形を作成できないため，対角方向への分割の際に求めた平均点を 2 点加え (c)，この平均点と交点を用いて三角形を作成し (d)，これらの三角形の法線の平均を発射方向とする (e)．

図 3.7 は交点がモデルの形状に収束する様子を表したものであり，(a) は上述した条件に従って光線を発射したもの，(b) はこれとは逆の操作を行ったものである．この図から分かるように，発射方向が正しくない場合は復元される形状が破綻することになる．

3.3.2.1 発射位置の判断方法

発射点がモデルの内部にあるかどうかは，発射点を通る円筒軸から円筒面までの線分とモデルとの交差回数で判断できる．図 3.8 はこれを示したものであり，図中 (b) のように発射点がモデル外部にある場合は，円筒面から発射点までのモデルとの交点数が偶数となる．一方，図中 (c) のように発射点がモデル内部にある場合は，交点数が奇数となる．

以上の手順を繰り返すことで，オクルージョン領域を捕捉することができる．

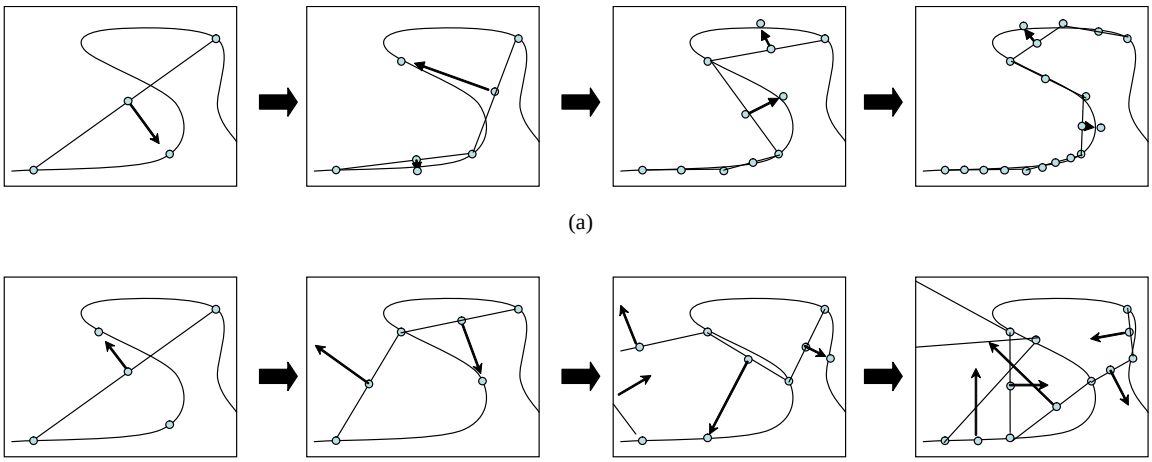


図 3.7: 発射点とモデルの位置関係による光線の発射方向の変更．(a) モデル外部のときは法線方向に，内部のときは法線とは逆方向に発射した場合．(b) (a) とは全く逆方向に発射した場合

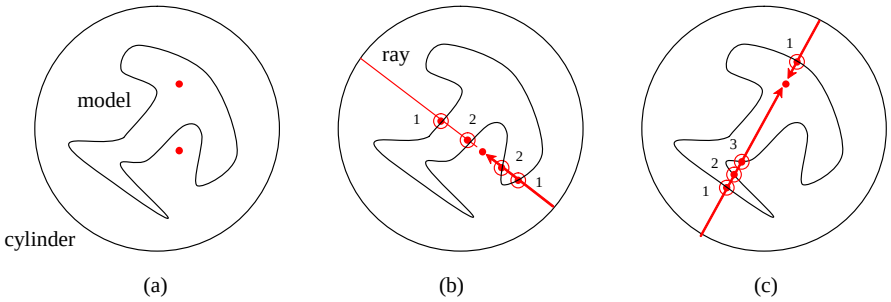


図 3.8: 発射点がモデルの外部にあるか内部にあるかの判定法．(a) のモデルの外部と内部にある 2 点の発射点に向けて，円筒面から光線を垂直に放った場合，(b) のように発射点がモデル外部の場合は，光線は発射点に到達するまでにモデル表面と“ 偶数 ”回交差する．一方，(c) の発射点がモデル内部にある場合には，この交差回数は“ 奇数 ”となる．

3.4　結果

提案した階層的な光線追跡法を実際のモデルを用いて行った結果を示す．実験には図 3.9 (a) の Mannequin head モデルと，より形状が複雑な図 3.10 (b) の Beethoven モデルを用いた．これらのモデルを円筒投影することで得られたジオメトリ画像が同図 (b) の黒い画像となる．ここで，取得画像がウェーブレット変換によって得られる多重解像度画像に似ているが，この画像は提案法の階層的な光線追跡で得られた距離画像であり，ウェーブレット変換などは行っていないことに注意されたい．また，(c) のモデルは 1 回目の光線追跡の段階での復元モデルを表したものとなる．1 回目の復元結果は従来法も提案法も同一のものが得られる．

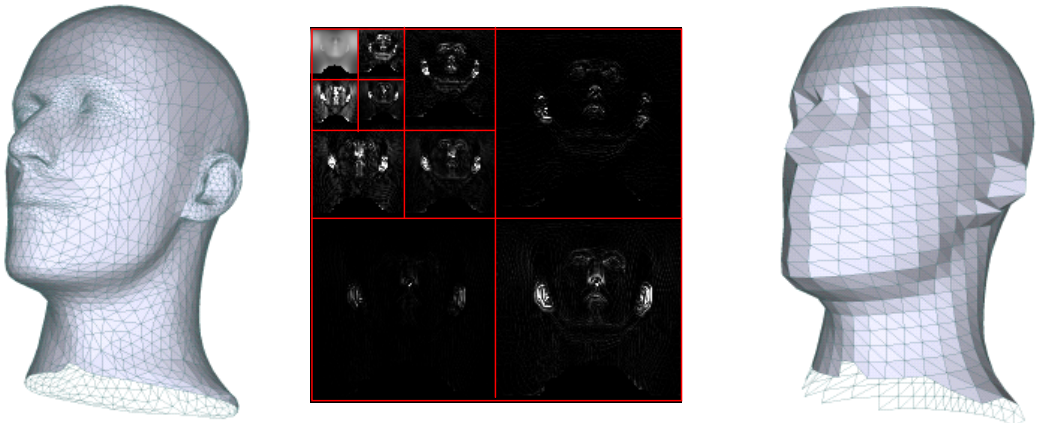
諸設定とジオメトリ画像についての説明：　ジオメトリ画像の画素に記録するジオメトリ情報は，3.2.2 項で定義したように光線の射出点から交点までの距離とする．この距離の長さを輝度で表したものが (b) に示した画像となる．すなわち，色が白い画素ほど距離が長いことを意味し，このような画素はモデルの形状に特徴のある部分に対応するように存在していることが分かる．

次に，光線追跡の回数や解像度であるが，最初の円筒からの光線追跡を 32×32 の解像度で行い，その後 3 回の光線追跡を行った．まず，1 回目の光線追跡の結果がジオメトリ画像 (b) の左上の輝度の明るい領域となる．次に，2 回目の光線追跡は水平・垂直・対角の 3 方向に対して行うことになる．この結果を表したものが 1 回目の画像の右（水平）・下（垂直）・右下（対角）の領域となる．また，2 回目以降は光線の射出点から交点までの距離も短くなるため，画像として表すと，このように全体的に黒く見えることになる[*3]．次に，3 回目以降の光線追跡からは，前回求めた交点が新しい射出点を求めるために加えられるため，解像度が前回の解像度の倍となる．この結果を表したものが，2 回目の結果の外側の領域となる．また，以降，光線追跡の回数を増やすごとに倍の解像度の領域が同様にして追加されていくことになる．このようにして，合計 4 回の光線追跡より画像サイズ 256×256 のジオメトリ画像 (b) が得られることになる．

3.4.1　2 回目以降の光線追跡の結果を加えた復元形状

2 回目以降の光線追跡結果を加えた場合のモデルの復元形状と，この結果との比較のため，円筒面から行う初回の光線追跡の解像度を単純に倍にしていった従来法の結果を図 3.11，図 3.12 に示す．

[*3] ここでは色の変化をより分かりやすくするために，各領域の画素値のスケールを変えたものを示した．実際にはほとんど黒色となる．

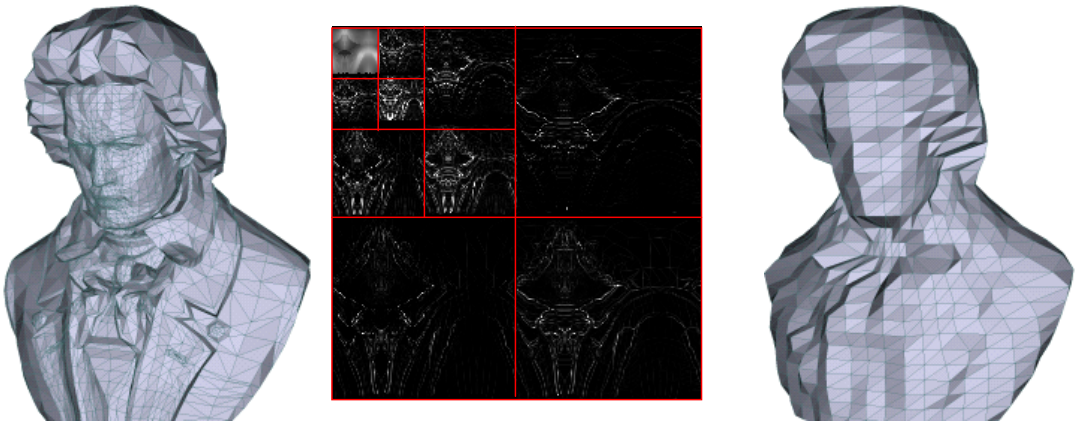


(a) Original model Mannequin-Head

(b) Geometry image of 3 time iterations (256x256)

(c) Reconstruction from 32x32 geometry image

図 3.9: Mannequin head モデルに対して階層的な光線追跡法を行った結果得られたジオメトリ画像とその初期復元モデル．(a) オリジナルモデル．(b) 合計 3 回の光線追跡によって得られた 256×256 の円筒ジオメトリ画像．(c) 1 回の追跡結果である 32×32 のジオメトリ画像から復元されたモデル



(a) Original model Beethoven

(b) Geometry image of 3 time iterations (128x128)

(c) Reconstruction from 32x32 geometry image

図 3.10: Beethoven モデルに対して階層的な光線追跡法を行った結果得られたジオメトリ画像とその初期復元モデル．(a) オリジナルモデル．(b) 合計 3 回の光線追跡によって得られた 256×256 の円筒ジオメトリ画像．(c) 1 回の追跡結果である 32×32 のジオメトリ画像から復元されたモデル

Models are courtesy of Washington University (Mannequin head) and Viewpoint Inc. (Beethoven).

復元形状であるが，一見それほど変わりはない．しかし，結果をよく見た場合，提案法ではモデルの輪郭などの特徴が，従来法よりも早い段階で復元されているのが分かる．例えば Mannequin head では，３回目の光線追跡結果を加えた段階で目線や口元が表現されており，Beethoven では，２回目の光線追跡結果を加えた段階で顔の表情や服の襟の部分が表れ始めているのが分かる．更に，提案法の場合はジオメトリ画像に記録される距離も小さくなるため，１画素の情報量に割り当てるビット数も少なくて済む．また，その画素の大部分は値が 0 に非常に近く（画像中の黒色領域），省略しても結果に支障をきたさないため，これらの画素値を省略することで大幅に情報量を削減することができる．

次に，提案法によるオクルージョン領域の復元の効果を示すため，モデルの一部を拡大したものを図 3.13 と図 3.14 に示す．まず，Mannequin head を拡大した図 3.13 では，耳の裏側の部分のオクルージョンとなり，従来法ではこの部分が短絡されているのが分かる．これに対して，提案法では裏側にまでメッシュが回りこみ，しっかりと耳たぶが表現されている．次に，より形状が複雑な Beethoven を拡大した図 3.14 では，襟元や襟首など，モデルを正面から見て目立つ部分ががオクルージョンとなり，またその領域も広いため，従来法ではこの部分にレース状のギザギザが顕著に表れているのが分かる．これに対して，提案法では襟の裏側にもメッシュが入り込み，襟元がしっかりと表現されている．

3.4.2　形状の数値的評価

従来法と提案法によるオクルージョンの復元度を３次元形状誤差評価ツール“ Metro ”[Cignoni et al. 1998] を用いて計測した結果を表 3.1 に示す．この結果からは，解像度を上げるにつれて両手法ともオリジナルモデルとの形状誤差は小さくなるが，オクルージョンの復元がなされている提案法の方が，すべての解像度において形状誤差が通常法よりも少なく，復元形状の精度が上回っていることがわかる．

上述のように，3D モデルを主観的に見た際に目立つ箇所を復元できることは，情報的にも非常に価値があると考えられる．また，形状を表現するのに必要な射出点から交点までの距離も徐々に小さくなっていくため，情報量もそれほど増加しない．そのため，解像度に合せた復元形状を高速に得ることができる．

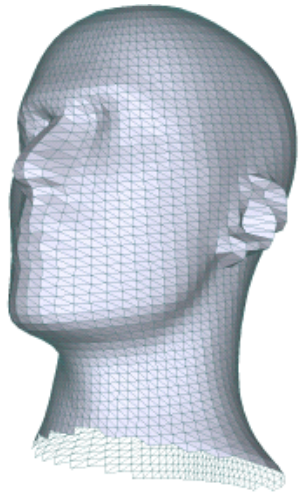
表 3.1: 3 次元形状誤差評価ツール“ Metro ” [Cignoni et al. 1998] による従来法との形状誤差の比較．初期解像度においては光線の射出点が両手法ともに円筒からであるため形状誤差の値は変わらない．計測設定はサンプリング法: similar triangles sampling, 計測法: Reast Mean Square error.

Mannequin head

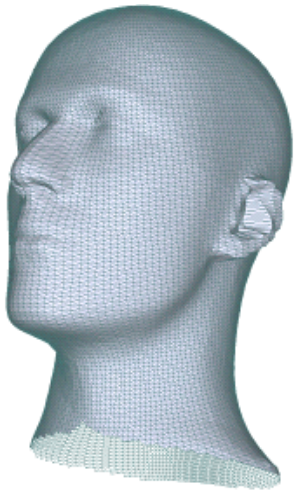
	Geometry image size			
Method	32×32	64×64	128×128	256×256
Normal	0.005109	0.002518	0.001536	0.001136
Hierarchical	same	**0.002465**	**0.001171**	**0.000482**

Beethoven

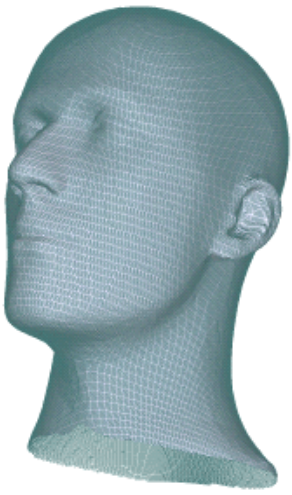
	Geometry image size			
Method	32×32	64×64	128×128	256×256
Normal	0.048876	0.025215	0.015591	0.012998
Hierarchical	same	**0.027519**	**0.01424**	**0.010859**



(a) Reconstruction from 64x64 geometry image

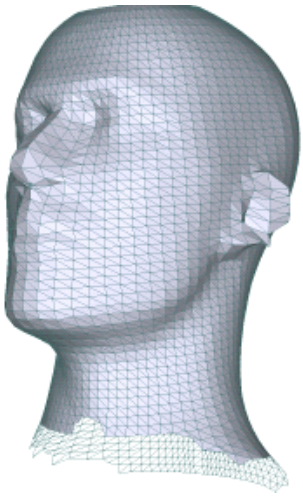


(b) Reconstruction from 128x128 geometry image

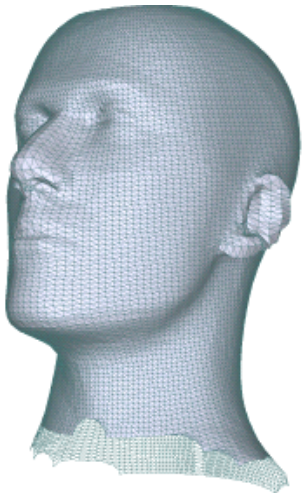


(c) Reconstruction from 256x256 geometry image

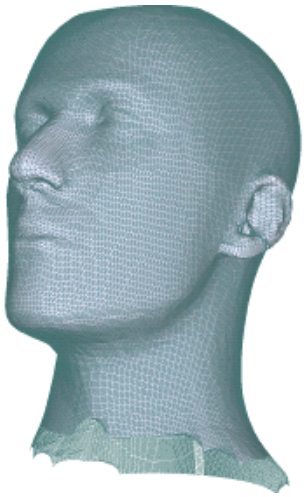
Reconstructions of the conventional method



(d) Reconstruction from 64x64 geometry image



(e) Reconstruction from 128x128 geometry image



(f) Reconstruction from 256x256 geometry image

Reconstructions of the proposed method

図 3.11: 従来法（上段）と提案法（下段）の各解像度での復元モデル．(a)(d) 解像度 64×64 の復元モデル．(b)(e) 解像度 128×128 の復元モデル．(c)(f) 解像度 256×256 の復元モデル

(a) Reconstruction from 64x64 geometry image

(b) Reconstruction from 128x128 geometry image

(c) Reconstruction from 256x256 geometry image

Reconstructions of the conventional method

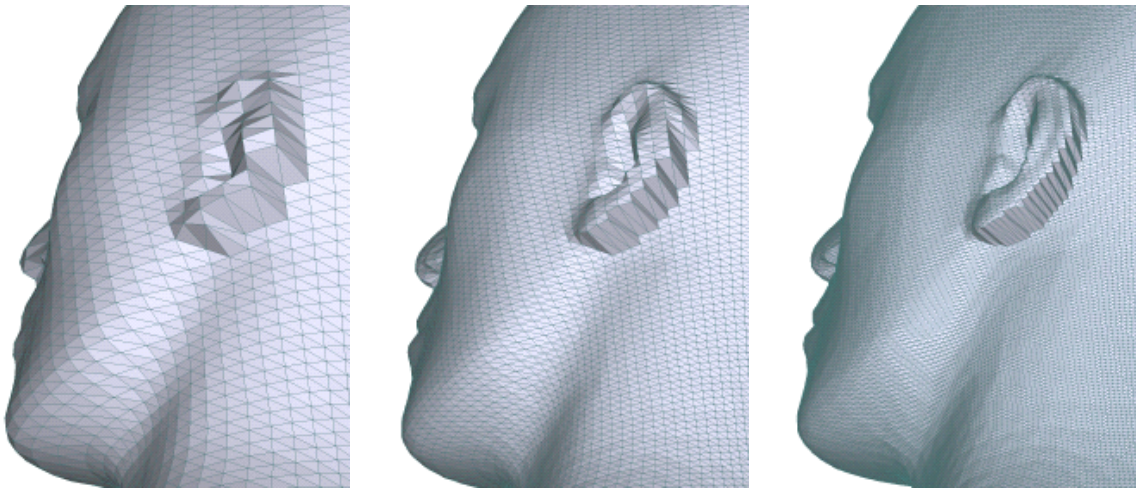
(d) Reconstruction from 64x64 geometry image

(e) Reconstruction from 128x128 geometry image

(f) Reconstruction from 256x256 geometry image

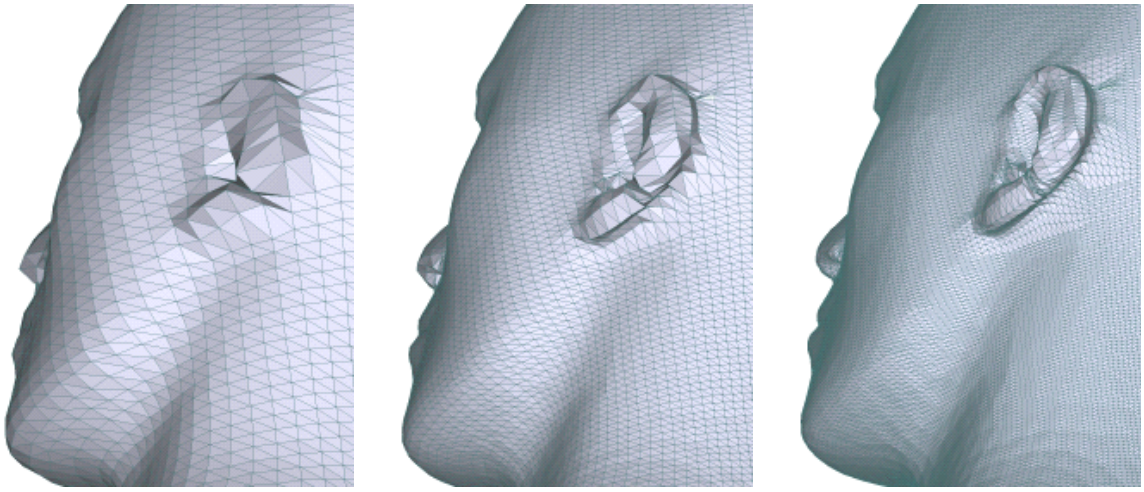
Reconstructions of the proposed method

図 3.12: 従来法（上段）と提案法（下段）の各解像度での復元モデル．(a)(d) 解像度 64×64 の復元モデル．(b)(e) 解像度 128×128 の復元モデル．(c)(f) 解像度 256×256 の復元モデル



(a) Reconstruction from 64x64 geometry image　(b) Reconstruction from 128x128 geometry image　(c) Reconstruction from 256x256 geometry image

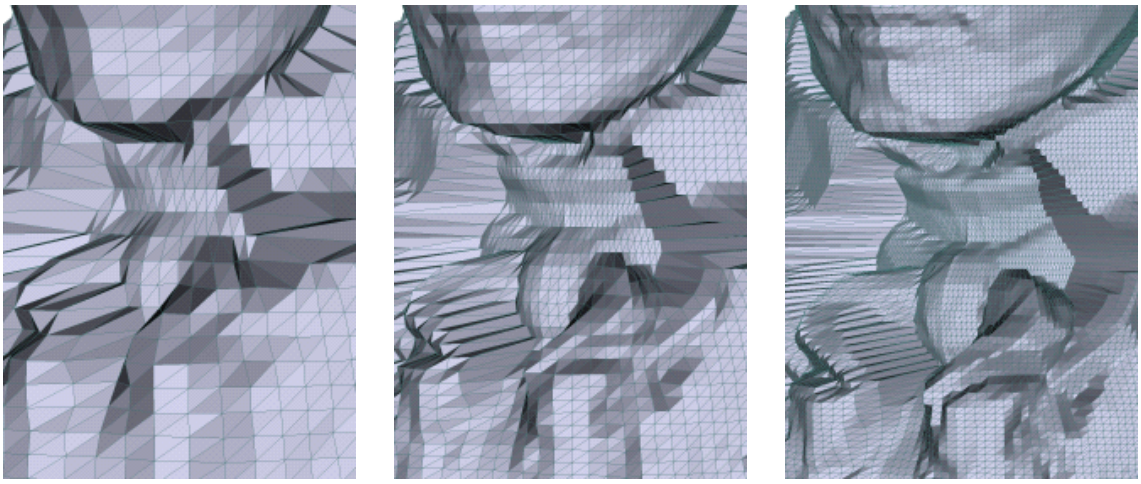
Reconstructions of the conventional method



(d) Reconstruction from 64x64 geometry image　(e) Reconstruction from 128x128 geometry image　(f) Reconstruction from 256x256 geometry image

Reconstructions of the proposed method

図 3.13: 従来法（上段）と提案法（下段）の各解像度での復元モデル．(a)(d) 解像度 64×64 の復元モデル．(b)(e) 解像度 128×128 の復元モデル．(c)(f) 解像度 256×256 の復元モデル

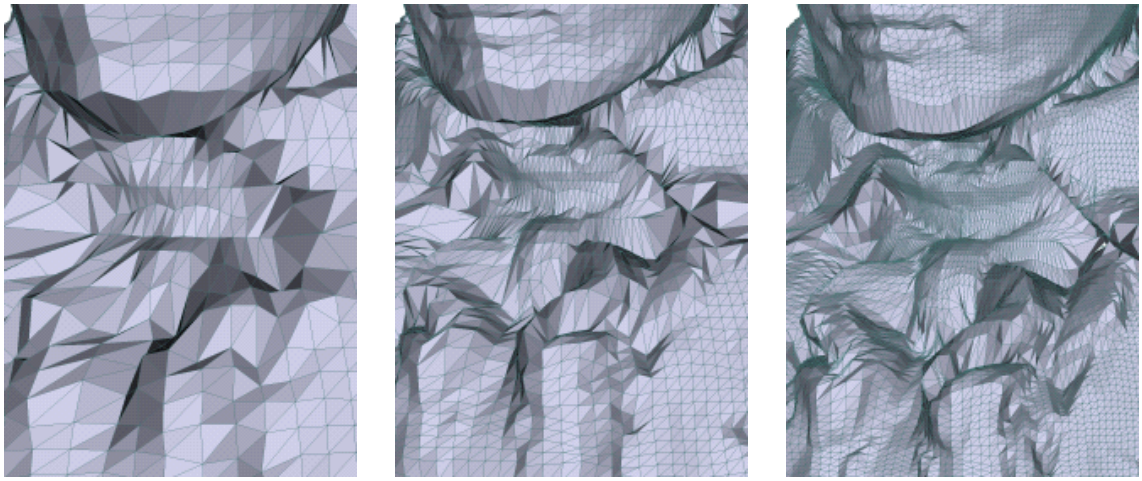


(a) Reconstruction from 64x64 geometry image

(b) Reconstruction from 128x128 geometry image

(c) Reconstruction from 256x256 geometry image

Reconstructions of the conventional method



(d) Reconstruction from 64x64 geometry image

(e) Reconstruction from 128x128 geometry image

(f) Reconstruction from 256x256 geometry image

Reconstructions of the proposed method

図 3.14: 従来法（上段）と提案法（下段）の各解像度での復元モデル．(a)(d) 解像度 64×64 の復元モデル．(b)(e) 解像度 128×128 の復元モデル．(c)(f) 解像度 256×256 の復元モデル

3.5 本章のまとめ

本章では，階層的に光線追跡を行う階層的円筒レイキャスト法を提案し，円筒投影で欠落するオクルージョン領域の復元が可能であることを示した．この手法では交点探索で得られた交点同士の中点から新たな交点探索を行い，これを繰り返す．その際に注すべき点として，光線を射出する法線方向を３次元で考える必要があること，射出点がモデルの外部にあるか内部にあるかで射出方向を反転させる必要があることがあげられる．この手法を用いることで円筒ジオメトリ画像からも非常に良好な復元形状を得ることができる．

第4章

投影速度の高速化を目的とした円筒Zバッファ法

4.1 はじめに

前章では，オクルージョン領域を捕捉し，ジオメトリ画像から復元される形状の精度を向上させることを目的とした円筒投影法について述べた．一方，3D モデルの円筒投影を行う目的としては，このような他にも 3D モデルの位置合せに用いる 2 次元投影像を得る目的もある [Pito 1996, Winkelbach et al. 2003]．このような目的に使用する場合は，オクルージョン領域の捕捉よりも，3D モデルの移動に伴う投影像をそのつど取得するための画像化速度が重要となる．しかしながら，円筒投影法で用いられているレイキャスト法は，現在の 3 次元 CG においては低速な投影手法であり，その画像化速度には数秒を要してしまう．このため，リアルタイムで処理を行うことは困難であった．

本章では円筒投影法の高速化を目的とし，従来法で用いられていたレイキャスト法の代わりに，同じく CG の投影法として知られる“ Z バッファ法 ”を適用させることを考える．すなわち，円筒ジオメトリ画像の作成のため,“ 平面 ”スクリーンへの投影法として知られる Z バッファ法を改良し，新たに,“ 円筒 ”スクリーンへの投影法として“ 円筒 Z バッファ法”を提案する [白井 啓一郎 et al. 2006a]．ただし，Z バッファ法は平面スクリーンへの投影法であるので，投影面が円筒になることに起因するいくつかの問題が生じることになる．

本章では，まず，Z バッファ法のアルゴリズムについて述べ，次に，投影面が平面から円筒に変わることで生じる問題点を示す．この問題を解決するための手法を Z バッファ法の手順に沿って示し，最後に，実際にモデルを投影した際の速度を従来の円筒投影法（レイキャスト法）と比較し，有効性を示す．

4.2　Z バッファ法の概要

Z バッファ法は，３次元 CG における投影法の一つで，メッシュを構成する三角形単位で投影処理を行う．このとき，投影面を構成する各画素に色情報のほかに奥行きに関係する情報を持たせ，描画の際に同じ座標の画素の奥行き情報を比較することで，最も手前にある色情報を投影面に書き込む．この奥行きに関係する情報のためのメモリ領域を Z バッファと呼ぶことから Z バッファ法と呼ばれる．

本章では，まず Z バッファ法の投影アルゴリズムについて簡単に述べ，次に，投影面が円筒になることで生じる問題点をあげる．

4.2.1　アルゴリズム

図 4.1 (z1-z4) は Z バッファ法 による投影面上の画素への交点の対応付けを示したものである．Z バッファ法では三角形を特定し，その投影面上の投影像が占める範囲内の画素に対して一括して交点情報を記録する（走査変換：Polygon Scan Conversion と呼ぶ）．これを全三角形に対して繰り返す [CG 標準テキストブック].

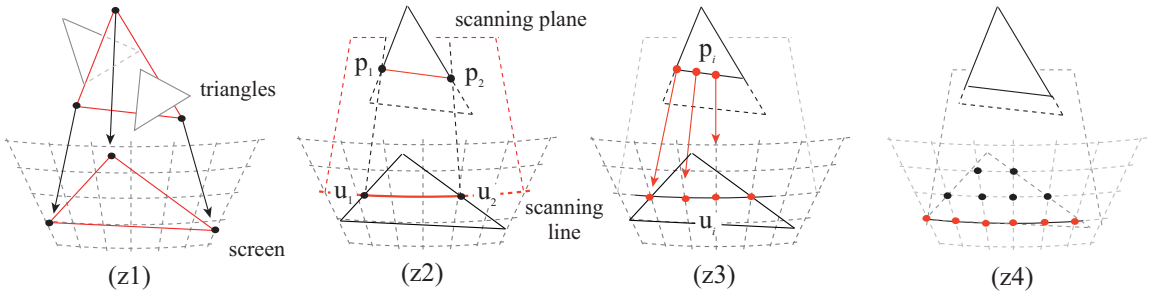


図 4.1: Z バッファ法による投影面への交点情報の対応付け．手前の破線の格子は投影面を表し，各格子点は画素の中心を表している．

Z バッファ法の手順は以下のようになる．

(z1) ある三角形（ここでは赤線枠）の投影面に対する投影像を考え，

(z2) 投影後の三角形に対して，ある水平スキャンラインとの交点（線分の両端の点 $\mathbf{u}_1, \mathbf{u}_2$ の座標）を求め，これを走査範囲とする[*1]．

[*1]投影法（より詳しくは陰面消去法）には，Z バッファ法の他にもスキャンライン法と呼ばれる方法があり，スキャンライン法では，あるスキャン平面に対して，これと交わる全ての三角形の交線とその投影を考え，画面の一ラインごとに描画を行う．順序と名称が紛らわしいため注意されたい．

(z3) 走査範囲内の各画素の座標 $\mathbf{u}_i$ と，対応する実際の三角形内の座標 $\mathbf{p}_i$ との距離を求め，その値を画素に記録する．通常は高速化のため，両端の点 $\mathbf{u}_1, \mathbf{u}_2$ 上で求めた値を線形補間して $\mathbf{u}_i$ を求める．投影前の三角形 $\triangle_{3D}$ 上も線形性を持つため，このように投影面上で三角形 $\triangle_{2D}$ を線形補間しても問題ない．

(z4) 以上の水平走査 (z2,z3) を繰り返すことで一つの三角形を投影面に投影する．

この操作を全三角形に対して行う．

ただし，任意の順序の投影では，手前の三角形が奥の三角形によって上書きされる場合があるため，奥行き判定用に投影面と同じ解像度の Z バッファを用意して三角形の円筒軸からの距離を蓄え，画素ごとにその距離を比較することで投影面に近い交点情報を上書していく．そのため，

(z3-a) 得られた交点の軸からの距離を，既に画素に記録されている値と比較し，投影面に近い交点であれば，この距離を含めたすべてのジオメトリ情報を上書きする．

Z バッファ法では，交点計算を行う三角形に無駄がなく，その交点計算も手順 (z3) に示したスキャン平面上の二次元的な計算で済むため，2.5.2 で示したレイキャスト法と比べ，高速に投影できる．

4.2.2 曲面の円筒面に適応する際に考えられる問題点

Z バッファ法は 3D モデル側から投影面側に対して投影を行うため，投影面が曲面になることで次のような問題が発生する[*2]．

(1) 投影面がモデルを完全に取り囲むため，ある方向からは見えないモデル表面も別の方向からは見える場合があり，この方向への投影が必要となる．三角形を円筒面側と円筒軸側の両方向に対して投影すれば解決可能であるが，この場合，上述した走査変換に余分な計算を必要とし，高速化の目的から外れることになる．

(2) 投影面が曲面となることで三角形の投影像は湾曲する．例えば，投影前の三角形 $\triangle_{3D}$ に含まれる座標 $\mathbf{p} = [x, y, z]$ は，投影により $\mathbf{u} = [tan^{-1}(z/x), y]$ に写される．得られた三角関数を含むため，座標群 $\mathbf{p}_i$ が持っていた線形性は失われることになる．そのため，投影後の三角形 $\triangle_{2D}$ の辺は歪み，内部の座標間隔も歪むことになる．このため，通常用いられる 2 次元の線形走査変換では正しい補間を行うことができなくなる．

提案アルゴリズムでは，主にこの二つの問題の解決を目的とする．

[*2] レイキャスト法の場合，投影面側から 3D モデル側へ交点探索を行うため，投影面が平面であっても円筒面であっても，投影面から放つ視線の角度さえ変更すれば，問題なく投影を行うことができる

4.3　円筒 Z バッファ法

4.3.1　投影方向の決定（z1）

スクリーンが円筒形であるため，投影方向は円筒軸から全方向となる．しかし，図 4.2 のようにモデルが円筒軸からずれる場合，図中の影の部分の三角形が軸方向を向くようになり，投影方向も図中 (a) の方向から (b) の方向へと逆転する．ここでは (a) の方向を円筒面から真正面に見える方向という意味で“ 正面方向 ”，(b) の方向をそれとは逆の円筒軸方向を向くということで“ 背面方向 ”と呼ぶ．この背面方向への投影を，投影式

$$(\theta, y) = (\tan^{-1}(z/x), y) \tag{4.1}$$

のみで扱うことができないことが問題となる．このため，ここでは三角形のスクリーンに対する向きから投影方向を判断し，式 (4.1) で投影を行えるよう三角形の頂点座標に補正を行う．

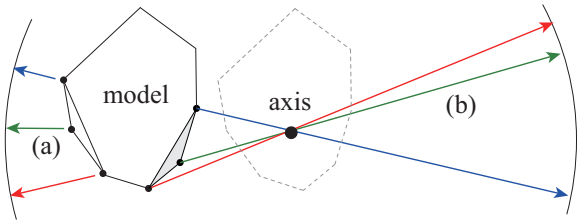


図 4.2: モデルの円筒軸からのずれによる投影方向の正面方向 (a) から背面方向 (b) への変化

4.3.1.1 正面投影と背面投影の判定

三角形の円筒スクリーンに対する向きは，三角形の 3 頂点を通る視線と三角形の法線のなす交差角，すなわち，3 つの内積の符号から知ることができる．

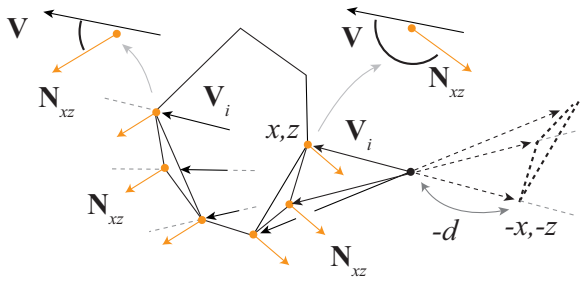


図 4.3: 正面投影と背面投影での視線と三角形法線の関係

図 4.3 は正面投影と背面投影での視線 $\mathbf{V}_i$ と三角形の法線 $\mathbf{N}_{xz}$ の関係を表したものであり[*3]，

[*3] この投影では y の値は常に一定なため，y の値は内積計算には関係しない．そのため，視線は円筒軸から頂点 (x_i, z_i), $i \in \{1, 2, 3\}$ を結ぶベクトル $\mathbf{V}_i = (x_i, z_i)$ とし，三角形の法線は，その x, z 成分 $\mathbf{N}_{xz}$ のみを必要とする．

その投影方向と内積の間の関係は,

(z1.a) 正面投影：3 つの視線と法線がともに円筒面方向を向き，その交差角は 90° 以下，すなわち，すべての内積が正 $\mathbf{V}_i \cdot \mathbf{N}_{xz} \geq 0$ となる場合,

(z1.b) 背面投影：3 つの視線と法線が反対方向を向き，その交差角は 90° 以上，すなわち，すべての内積が負 $\mathbf{V}_i \cdot \mathbf{N}_{xz} \leq 0$ となる場合,

となる．ただし背面投影は式 (2.1) では扱えないため，図中の破線で示した三角形のように,

(z1.b2) 背面投影となる三角形の x, z 座標の符号を反転させる．また，以降この三角形から得られる距離 d の符号を反転させる．

しかし，異符号を同時にもつ例外的な三角形も存在し,

(z1.c) 視線と法線の内積に異符号を同時にもつ場合，三角形は領域によって投影方向が異なる．このため，例外三角形として処理する．

4.3.2 例外三角形

前項 4.3.1 の投影方向の判断基準で，投影方向を一概に決定することができない三角形を，ここでは“ 例外三角形 ”と呼ぶ．図 4.4 は例外三角形において数字で示した各領域が円筒面にどのように映るのかを示したものである．ここではこのように投影像が分かれたり，引き伸ばされる理由と，その処理法について述べる．

この説明のため，まず三角形表面を点（微小な面）の集合として考える．このように考えたとき，三角形の向きの判定は，三角形上の任意点での向きの判定，すなわち，任意点での視線と法線の交差角の判定に帰着される[*4]．

この考えのもと図 4.4 (a) の三角形上の点を通過する視線と法線の交差角を見た場合，三角形中央においてその交差角が鋭角から鈍角に変化する（つまり内積の符号が反転する）境界線が直線的に存在し，交差角が鋭角となる領域 1 と交差角が鈍角となる領域 2,3 に分けることができる（領域 2,3 はスクリーンに映る位置関係を理解しやすくするために分けた）．これは投影が y 軸に対して常に垂直に，かつ，外側のスクリーンに向けて行われるために起きる特殊な現象であるが，領域 1 と領域 2, 3 のもつ交差角は明らかに異なるため，それぞれ正面投影と背面投影となる．このとき，投影像がずれ，軸の通過に伴い像が反転する．

次に，図 4.4 (b) のように三角形の内部に円筒軸を含む場合，図 4.4 (a) の三角形同様に投影方向の境界が円筒軸を通るように存在し，背面投影の領域 1, 2 と正面投影の領域 3, 4, 5 に

[*4] 前項 4.3.1.1 において頂点のみで三角形の向きを判定したが，これは頂点上での交差角がすべて鋭角（鈍角）の場合は頂点に囲まれるすべての点上の交差角も鋭角（鈍角）になるためである．

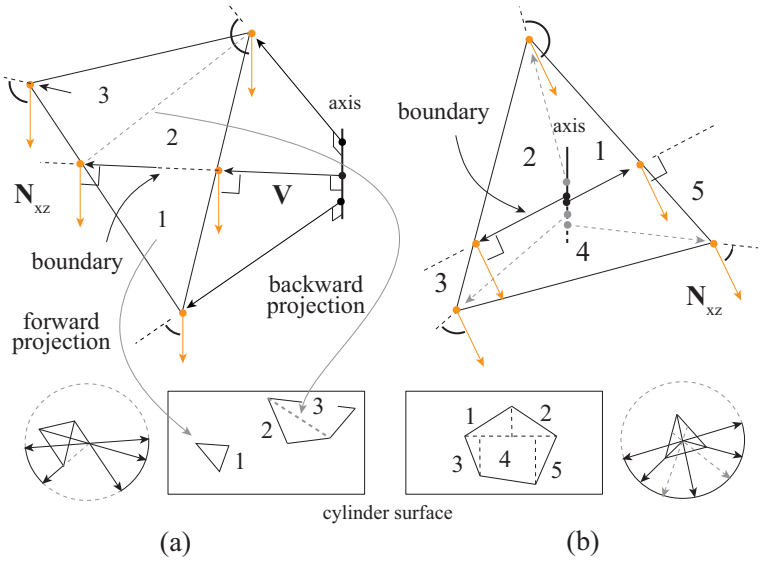


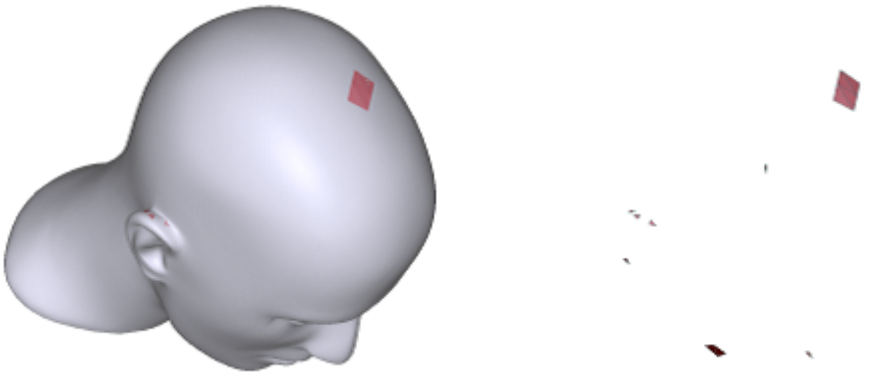
図 4.4: 内部領域の投影方向が異なる特殊な三角形と，その細分化された三角形領域のスクリーン上の投影像

分けることができる．ここで図 4.4 (a) の投影像と異なるのは投影像の中央が広がることである．これは円筒軸と三角形の交点の投影像であり，円筒図法を用いた世界地図の極部が広がるように，極点となる軸上の点はどの方向からも見え，引き伸ばされることになる．ただし，通常は三角形の裏面は不可視として扱われるため，投影像は半円周長となる．

処理方法： 例外三角形を図 4.4 に示した各領域ごとに三角形を細分化すれば，個々の三角形となり，各方向への投影が可能となる．実際にこのような条件を満たす可能性のある例外的な三角形は，円筒の中心角に対する三角形の占める範囲が大きく，多くの異なる角度の視線が内部を通過する三角形となるが，これは通常は円筒軸に相当近接した三角形となる．図 4.5 はモデル Mannequin head に含まれる例外三角形を示したものであり，例外三角形は円筒軸に近接した頭頂部に数個，その他まばらにしか存在していないことがわかる．様々なモデルで試した結果では，円筒軸周りに全体の三角形数の 0.02〜0.03%ほど含まれる程度で，この三角形の分割処理自体にはほとんど時間はかからなかった．

4.3.3　円周方向への走査範囲の限定（z2）

通常スキャンラインごとの走査変換では，まずスクリーンに投影された三角形の辺とスキャンラインの交点を求め，その大小関係を比較し，走査変換の開始点 $\mathbf{u}_{min}$ と終了点 $\mathbf{u}_{max}$ を定めることで水平方向の走査範囲の限定を行う（図 4.6 (a) 参照）．スクリーンが平面の場合（つまり通常の Z バッファ法）は，この交点計算は簡略化することが可能であり，t 回目の水



(a) On a surface (b) Patches

図 4.5: モデルに含まれる例外三角形

平走査に用いる交点を $\mathbf{u}^t$ とするとき，$t+1$ 回目の交点は $\mathbf{u}^{t+1} = \mathbf{u}^t + d\mathbf{u}$ として常に一定な差分 $d\mathbf{u} = (dx, dy)$ を足し込む加算のみで得られる（図 4.6 (b) 参照）．

しかし円筒スクリーンへの投影では，三角形の投影像はひずみ，その辺も湾曲する．したがって，新たな交点を差分 $d\mathbf{u} = (d\theta, dy)$ の加算により取得できないことが問題となる（図 4.6 (c) 参照）．

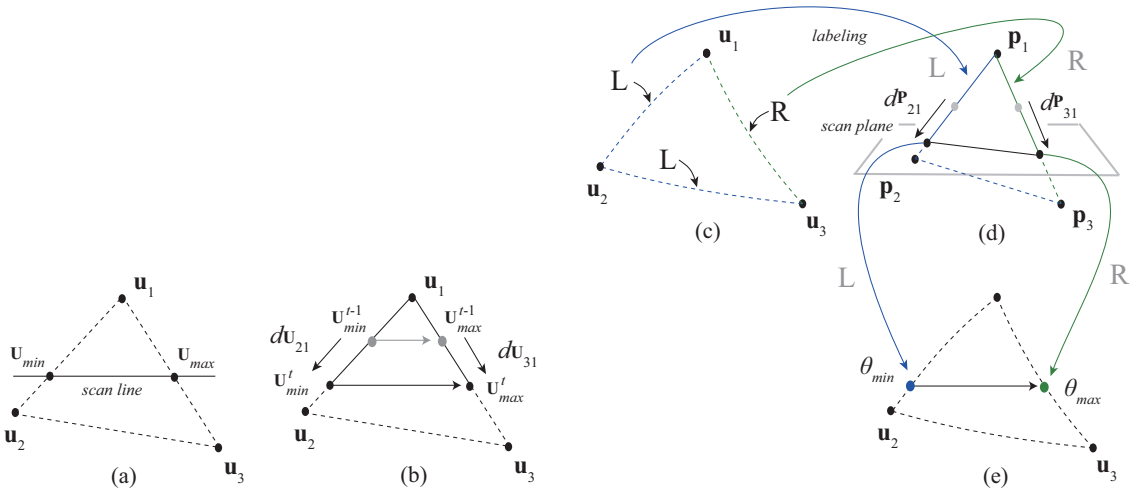
4.3.3.1 三次元での交点更新と辺のラベル付け

湾曲した三角形 $\triangle_{2D}$ 内部の補間を行うため，ここでは，図 4.6 (d) に示すように，投影前の三角形 $\triangle_{3D}$ の辺とスキャン平面との交点を求め，三次元ベクトルである差分 $d\mathbf{p} = (dx, dy, dz)$ の加算により更新を行い，これら更新された交点を式 (2.1) で順次スクリーンに投影することで，開始点 θ^t_{min} と終了点 θ^t_{max} を求めることを考える．

ただし，計算から求まる θ の値は $[0, 2\pi)$ の範囲でループするため，単純に大小比較して開始点 θ_{min} と終了点 θ_{max} とすると円筒の区切れ目にまたがる三角形（*split_triangle*）では誤った走査範囲を与えてしまう（図 4.7 (a) 参照）．この回避のため，図 4.6 (b) に示すように三角形の辺に L と R の記号のラベル付けを行う．すなわち，投影前の三角形の辺がスクリーンに投影された際に“ 左側 ”の辺となるか“ 右側 ”の辺となるかを明示的に定め，それぞれの辺上の交点が投影される際に左辺上の交点が θ_{min}，右辺上の交点が θ_{max} となると定義する（図 4.6 (d) 参照）．このようにすれば円筒の区切れ目にまたがる三角形ではループが生じた時点で $\theta_{max} < \theta_{min}$ となり，

$$\begin{aligned} &\text{if } (\ \mathit{split\_triangle} \cap \theta_{\max} < \theta_{\min}) \\ &\qquad \theta_{\max} = \theta_{\max} + 2\pi \end{aligned} \tag{4.2}$$

として範囲を補正することが可能となる（図 4.7 (b) 参照）．



Case of the flat screen　　Case of the cylindrical screen

図 4.6: 湾曲した三角形の辺上での，水平走査の開始点と終了点を求めるための，三角形の辺への左辺と右辺のラベル付けと三次元での交点座標の更新

4.3.3.2 ラベル付けのための前処理 （円筒の区切れ目にまたがる三角形の補正）

上述した三角形のラベル付けを実現するためには，以下の補正処理が更に必要となる．ラベル付けはスクリーン上に映った頂点座標 $\mathbf{u}_i = (\theta_i, y_i)$ の大小関係を見て行うが，この三角形では頂点 $\mathbf{p}_i$ を投影した時点で既にループが生じるため，このままでは図 4.7 (c) の中央の三角形に対してラベル付けを行うことになる．このため，まず三角形が円筒の区切れ目にまたがるかどうかを判断し，図 4.7 (d) に示すように頂点を移動させ，正しい三角形を作るようにする必要がある．

ここで円筒の区切れ目にまたがる三角形では，その誤った投影像の θ 方向の長さが半円周長よりも長くなる性質がある．なぜなら投影像の最大長は円筒軸上の極点で半円周長であり（4.3.2 参照），通常はその長さは半円周長よりも短い．ループした場合は長さが逆転するので半円周長よりも長くなる．このため，その投影像の長さ，すなわち，それぞれの頂点間の距離 $d\theta$ を比べることで他の三角形と区別することができる．

結局 4.2 の手順 (z2) はラベル付け (z-a) と交点の更新手順 (z2-b) とで構成され，上記判別方法と補正処理を加えたラベル付けの手順は以下のようになる．

(z2.a) 三角形の頂点 $\mathbf{p}_i$ を式 (2.1) によりスクリーン座標 $\mathbf{u}_i$ に変換する．

(z2.a2) ある頂点 $\mathbf{u}_i$ と他の頂点 $\mathbf{u}_j, \mathbf{u}_k$ の差分 $d\theta_{ij}, d\theta_{ik}$ を求め，この値がともに $d\theta > \pi$ または $d\theta < -\pi$ ならば，円筒の区切れ目にまたがる．よって，それぞれ $\mathbf{u}_i+ = 2\pi$, $\mathbf{u}_i- = 2\pi$ と補正する．

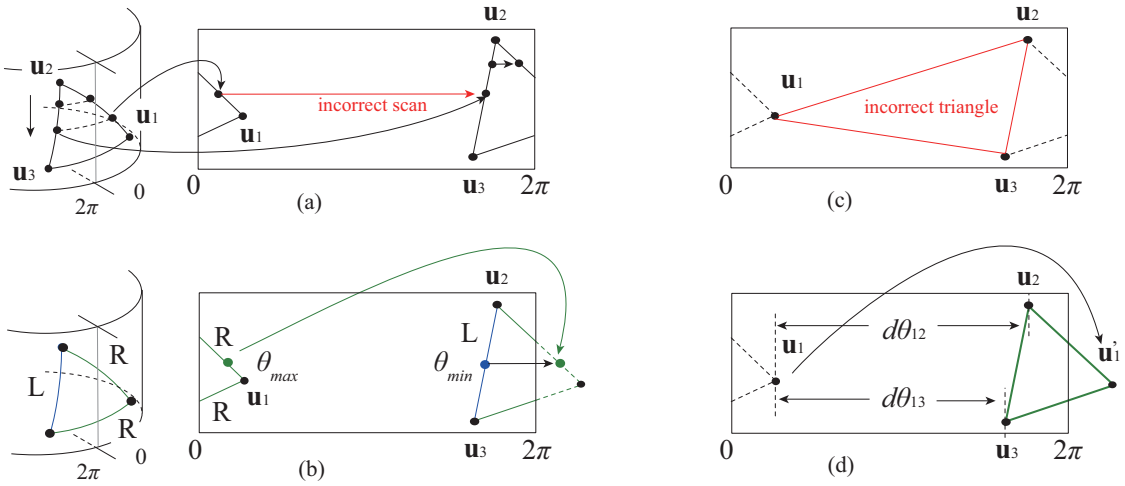


図 4.7: 円筒の区切れ目にまたがる三角形で生じる誤った走査変換と，ラベル付けによる訂正の様子．また，このラベル付けの前処理に必要な頂点座標の補正

(z2.a3) 補正された頂点 $\mathbf{u}_i$ の大小関係を比較し，もとの三角形の辺が投影後に左辺となるか右辺となるかをラベル付けする．

また，走査範囲の開始点と終了点を与える交点の更新手順は，

(z2.b) 加算用に用いる $d\mathbf{p}_{ij} = (\mathbf{p}_i - \mathbf{p}_j)/|\mathbf{p}_i.y - \mathbf{p}_j.y|$ を計算する．

(z2.b2) 三角形の辺とスキャン平面との交点を $d\mathbf{p}$ の加算により更新する．この交点を順次投影するが，左辺の交点が水平走査変換の開始点 θ_{min} を与え，右辺の交点が終了点 θ_{max} を与えるとする．

(z2.b3) ただし，$\theta_{max} < \theta_{min}$ となる場合，円筒の区切れ目にまたがる三角形で θ の値にループが生じているため，式 (4.2) のように補正する．

以上により，湾曲した三角形の走査変換を行う準備が整う．

4.3.4 水平方向への走査変換（z3）

円筒スクリーンへの走査変換では，スキャン平面上で光線と三角形（スキャン平面との交線）の交点を求め，交点情報を画素値に記録していく（図 4.8 (a) 参照）．ここでは交点計算の手順と計算式の簡略化について説明する．

4.3.4.1 交点座標と奥行き判定用の距離の取得

まず三角形とスキャン平面との交線と光線を式で表し，次にこの交点を求める（図 4.8 (b) 参照）．また y の値は一定であるため，$y = y_v$ のスキャン平面上を扱うものとして以後は省略する．

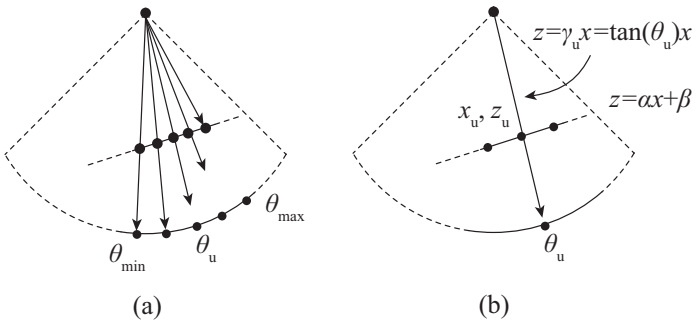


図 4.8: 円周方向への走査変換とその交点座標の取得の様子

交線の両端の点の座標をそれぞれ (x_1, z_1), (x_2, z_2) とすれば，交線の式は，

$$z = \alpha x + \beta, \quad \alpha = \frac{z_1 - z_2}{x_1 - x_2}, \ \beta = z_1 - \alpha x_1 \tag{4.3}$$

と与えられる．次に，円筒面上の画素位置 θ_u と円筒軸を通る光線を考えると，この直線式は，

$$z = \gamma_u x, \quad \gamma_u = \tan(\theta_u) \tag{4.4}$$

と与えられる．この交線と光線から交点 (x_u, z_u) が，

$$(x_u, z_u) = \left(\frac{\beta}{\gamma_u - \alpha}, \frac{\gamma_u \beta}{\gamma_u - \alpha} \right) \tag{4.5}$$

として求まる．この交点 (x_u, y_v, z_u) のもつジオメトリ情報を画素 (u, v) の値として記録することになる．

手順 (z3) での奥行き判定を行うため，円筒軸からこの交点までの距離 d_u を求める．交点座標 (x_u, z_u) を用いれば，

$$d_u = \sqrt{x_u^2 + z_u^2}$$

と与えられ，これに式 (4.5) を代入すれば，

$$d_u = \left| \frac{\beta}{\gamma_u - \alpha} \right| \sqrt{1 + \gamma_u^2} \tag{4.6}$$

として求まる．ただし，背面投影された三角形の場合はその符号を負とする．

計算の簡略化：　式 (4.4) の光線の傾き γ_u は三角関数より求まるが，画素数 u が有限個であるため，円筒画像のサイズを決める際にあらかじめ計算し，ルックアップテーブルに保存しておくことで計算を簡略化することができる．同様に式 (4.6) の $\sqrt{1 + \gamma_u^2}$ の項もテーブル

化する．また，交線の傾き α も円筒軸に垂直なスキャンではそれぞれの三角形で一定であるため，一度求めるだけで済むことになる．

4.4　結果

提案した円筒 Z バッファ法を用いて，3D モデルを円筒ジオメトリ画像化する実験を行った．また，円筒ジオメトリ画像から復元した形状を合わせて示す．実験は三角形数が 50K 以上の三角形数を持つ数体のモデルを用いて行った．図 4.9 と図 4.10 は 50K 以上の場合の結果の一部を示したものであり，各行がそれぞれ，オリジナルモデル，距離マップ，復元モデルを表している．ただし，復元モデルは表示のため, オクルージョンによる形崩れを補正している．

復元メッシュの補正法：　図 4.9 のモデルでは，円筒からは見えない頭頂部に本来穴が生じるが，穴領域の多角形ポリゴンを三角形分割（Triangle Tessellate）[document] することで穴を埋めている．一方，図 4.10 の凹凸の多いモデルの場合，(a) の Happy buddha では，右手の部分がオクルージョンとなり欠落している．このようなことを防ぐため，(b) の Horse では高速なメッシュ分割法 [Yan et al. 2005] を用いて，大まかにメッシュを切り分けて処理を行った，この処理は数秒で終了する．

4.4.1　画像化速度の比較

画像サイズの設定：　画像化速度を量るための画像サイズは，２章の実験と同様に，復元モデルの三角形数がオリジナルモデルの三角形数と同等となるように設定した．

従来法の設定：　提案法との比較対象にはレイキャスト法に高速化を施した従来法を用いた．高速化の手段としては，3 次元の交差判定法に Möller ら [Möller and Trumbore 1997] の手法を用い，交差判定を行う三角形を限定するため,“ 空間分割法 ”を用いて三角形を空間的にグループ化させた．なお，空間分割の仕方であるが，モデルの存在する空間を扇形に分割し，分割数は画素数と同じとした．すなわち，画像サイズが 400×200 ならば 400×200 の扇形空間に分割されることになる．

ただし，この分割法は必ずしも最適とはいえないため，提案法の交差判定を 3 次元に置き換えたものも示した．すなわち，手順 (z2.a2) で求まる三角形の投影像が占める範囲の四角形領域において，投影前の三角形に対してレイキャスト法と同様の処理を行った．これは手順こそ違うが，極力無駄な交差判定を省いたレイキャスト法を想定していることになる．

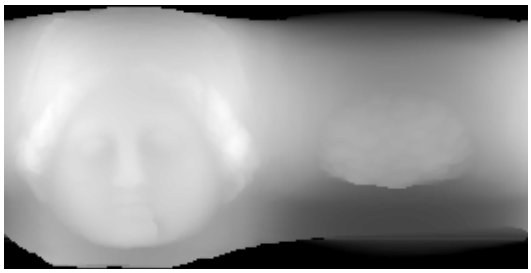
表 4.11 は三角形数と投影速度の関係を示したものであり，横軸が三角形数，縦軸が投影速

Igea

(a) Original (67K face)

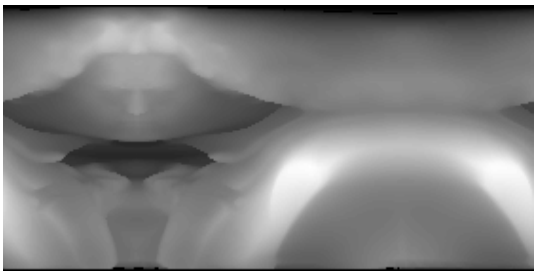
Beethoven

(b) Original (80K face)



(c) Distance-map (260×130)

Processing-time 0.125 s



(d) Distance-map (280×140)

Processing-time 0.141 s

(e) Reconstruction (64K face)

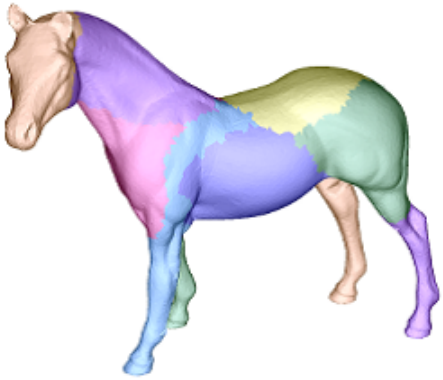
(f) Reconstruction (76K face)

図 4.9: オリジナルモデルの円筒ジオメトリ画像と復元例 (1/2).

Models are courtesy of Cyberware Inc. (Igea), and Viewpoint Inc. (Beethoven).

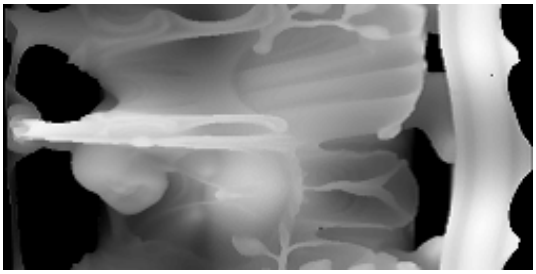
Happy buddha

(a) Original (293K face)



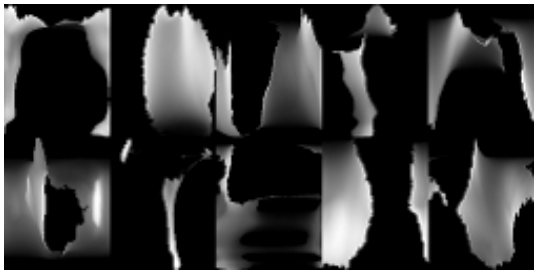
Horse

(b) Original (158K face)



(c) Distance-map (540 × 270)

Processing-time 0.656 s

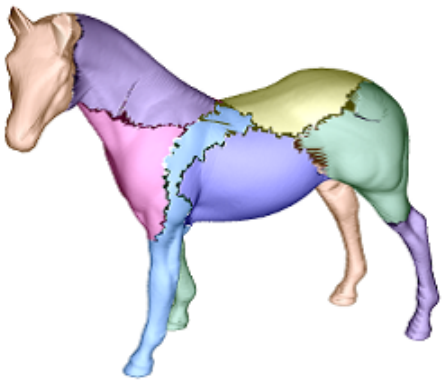


(d) Distance-map (400 × 200)

Processing-time 0.266 s

with segmentation +1.469 s

(e) Reconstruction (260K face)



(f) Reconstruction (70K face)

図 4.10: オリジナルモデルの円筒ジオメトリ画像と復元例 (2/2)．(d) についてはメッシュ分割にかかる追加時間も示した．

Models are courtesy of Stanford University (Stanford bunny, Happy buddha).

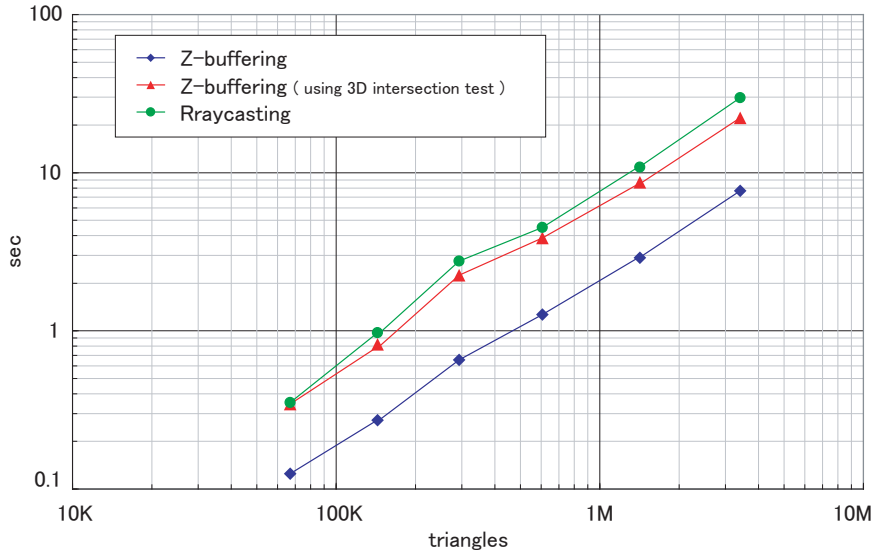


図 4.11: モデルのジオメトリ画像化に要した時間．Pentium4 3.4GHz MSVC7 C++で計測，CPU (G7) &Win (GA) 最適化済み

度を表している．まず，提案法と比較する従来法であるが，レイキャスト法の空間分割をかなり細かく区切ったため，Z バッファ法の交差判定を 3 次元に置き換えたものの変換速度がわずかに上回る程度である．これらに対し，提案した Z バッファ法では，ジオメトリ画像化速度が 40%まで短縮され，大幅な改善がなされているといえる．また，レイキャスト法と提案法とは手順のみが異なるため，得られるジオメトリ画像はレイキャスト法のものと同等となり，復元される形状は両手法とも同一のものが得られる．

4.5　本章のまとめ

本章では円筒投影法の高速化を目的とし，従来の円筒投影で用いられているレイキャスト法の代わりに，より高速な投影法である Z バッファ法の円筒投影への適用を試みた．Z バッファ法は平面スクリーンへの投影法であるため，投影面が円筒になることにより，取り囲む投影面のどの方向に三角形を投影すべきか考える必要があり，円筒面に投影される三角形の辺や内部が歪むという問題も生じることになる．本章では，まず，投影方向を判断するめ，円筒軸から放った光線と三角形の法線の内積を用いた判断式を示した．次に，投影面上で歪む三角形内部を補間するため，投影前の三角形の辺を用いて，投影像の範囲を絞りつつ交点計算を行う補間方法を示した．この円筒 Z バッファ法により円筒ジオメトリ画像の作成速度が飛躍的に向上することになる．

第5章

3Dモデル検索への応用

5.1 はじめに

円筒投影により3Dモデルは円筒ジオメトリ画像へと変換されるが，類似した形状のモデルを変換した場合，そのジオメトリ画像も類似する（図5.1 (a)(c)参照）．形状が類似したモデルでは，モデルの軸からの半径の変化も似ており，この半径を記録した円筒ジオメトリ画像にもその影響が表れていると考えられる．このことを利用することで，円筒ジオメトリ画像の類似性を用いて3Dモデルの検索が可能になると考えられる．

本章では，まず，検索法の概要について述べ，次に，近年の画像検索で用いられる主成分分析を用いたマッチング法について説明する．最後に，既存の3Dモデル検索法と検索精度と速度を比較し，その有効性を示す．

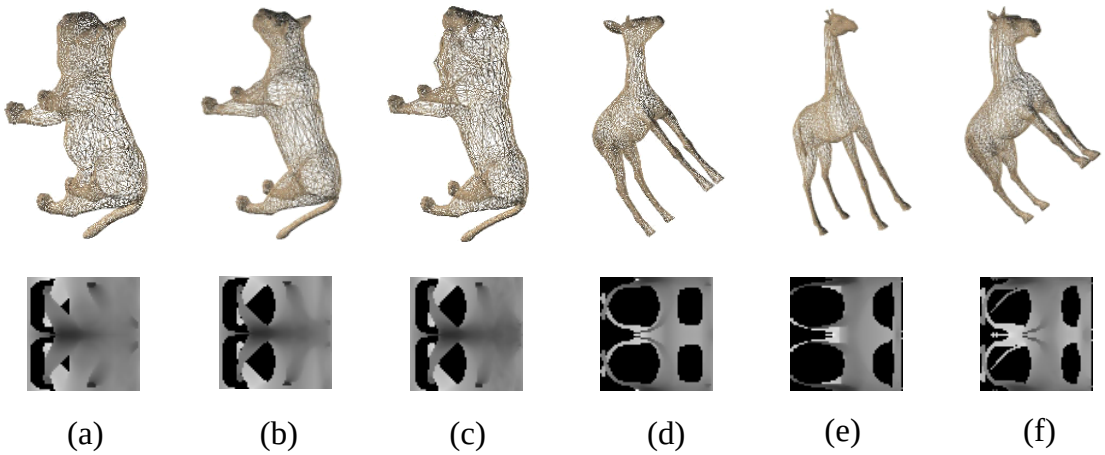


図 5.1: 形状の類似した3Dモデルの円筒ジオメトリ画像．(a)(b)(c)や(d)(e)(f)のモデルのように形状が類似したモデルでは，その円筒ジオメトリ画像も似たものとなる

5.1.1　関連研究

3D モデルの検索法としては，モデル間の形状誤差を測ることで類似モデルを探す方法 [Vranic and Saupe 2002, Elad et al. 2000, Novotni and Klein 2001, Osada et al. 2001] などが提案されているが，これらの手法では 3 次元での誤差の計算が必要となるため，計算量が多く，検索速度の面で問題があった．このため，３次元形状に関係する情報を持つ 2 次元画像を用いて検索を行う手法が提案されており [Johnson 1997, Johnson and Hebert 1999, Chen et al. 2003]，次元数を減らすことで計算量の削減と検索の高速化を実現している．[Johnson 1997, Johnson and Hebert 1999] は“ Spin images ”と呼ばれる画像を用いて 3D モデルの位置合わせなどを行う方法であり，ある注目頂点とその周囲の頂点の局所的な位置関係を 2 つのパラメータで近似し，これを記録してパラメータ画像（スピン・マップ）を作成する．この画像はモデルの回転や移動に対して不変なため，画像マッチングにより，３次元形状の類似性を測ることができる．[Chen et al. 2003] は 3D モデルをある平面に投影して得られる 2 次元シルエット（輪郭）画像を用いる方法であり，形状が類似したモデルではそのシルエットも類似することを利用し，シルエット画像のマッチングにより，３次元形状の類似性を測定している．ただし，両手法とも，局所的な方向から得られた画像を用いて類似性を比べるため，モデル全体の形状が類似しているかどうかを知るためには，10 枚から 20 枚の他視点からの画像を必要とし，各画像間で類似性を比べる必要がある．当然ながら，使用する枚数が増えれば，平面に対するモデルの投影処理や，画像マッチングの計算量は増大し，前処理と画像検索を合わせた検索処理に時間を要する．このような理由から，円筒画像一枚でモデルの検索が可能であれば，検索速度の更なる時間短縮を実現することができると考えられる [Shirai et al. 2006a].

5.2 アルゴリズム概要

提案する検索法の大まかなアルゴリズムは以下のとおりである．

円筒投影によるジオメトリ画像化

画像マッチングに用いる円筒画像を 4 章で提案した高速な円筒 Z バッファ法を用いて作成する．作成する円筒画像のサイズであるが，ここでは形状復元時に元のモデル形状を十分に表現できる画像サイズとして 144×144 とした．合計 20736 の形状情報のサンプルを得ることになるので，マッチングを取るには十分と考えられる．

画像サイズの縮小

得られた円筒画像のサイズを 48×48 に縮小する．画像マッチングでは主に画素値の大まかな変化を表す低周波数成分の類似度が大きく影響するため，高解像度の画像を用いても検索結果に違いは現れないためである．画像サイズは主観的に見ても十分に違いを判断できる大きさということで 48×48 とした．なお，縮小は単純に画素を間引くのではなく，ローパスフィルタをかけてから画素を間引くことで縮小した．初期画像サイズを大きめに設定してから縮小する理由であるが，円筒投影法は単に円筒状の画素に対応するモデル上のある 1 点をサンプルする手法であるため，サンプリング間隔を粗くすると，画素間の値が不連続となる．また，局所的に得た値に全体的な結果が左右されることになる．このため，３次補間を用いて画像を縮小させることで，画素値に周囲の画素の影響を持たせ，ロバストなマッチングを行えるようにしている．

この段階でマッチングに使用する成分の数は $48 \times 48 = 2304$ 個となる．図 5.2 (a)(b) はここまでの処理を示したものである．同図 (c) での画素の並び変えは主成分分析用の準備手順を表している．

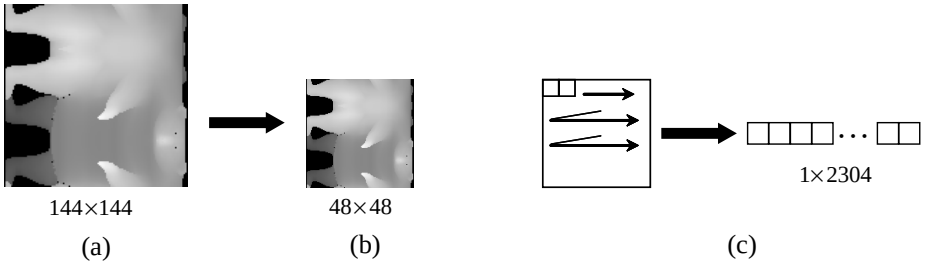


図 5.2: 主成分分析のための検索画像の変換．(a) 円筒投影で得られた画像．(b) 縮小による冗長度削減．(c) 列ベクトルへの変形

主成分分析によるデータベース内の画像特徴の学習

得られた 2304 個の成分同士の二乗誤差の比較によるマッチングも可能であるが，これを数百体，数千体のモデルに対して行と計算量が多くなり現実的でない．そのため，主成分分析を用いてデータベース内の画像の特徴を学習させることで，最も検索に影響を与える成分を 2304 個の成分から抜き出す（主成分へと変換させる）．

主成分分析を用いた画像マッチング法を扱った論文 [Ke and Sukthankar 2004, Fan et al. 2005] などでは，主成分数は 100 から 200 個ほどでも十分にマッチングを取れることが示されている．この場合，単純計算で 20 倍以上の速度の向上が考えられる．また，マッチングに悪影響を及ぼす成分も除外されると考えられるので，検索精度の向上も期待することができる．

主成分を用いたマッチング

主成分分析を行うと，負荷行列（Loading Matrix）と呼ばれるデータから主成分を取り出すための変換行列が得られる．この変換行列を用いて少数の主成分を画像から抜き出し，マッチングを行うことになる．

以上が大まかな流れである．ただし，主成分分析において負荷行列による変換の仕方や，主成分数の決定法などの説明が必要だと考えられるため，次章にてその詳細を述べる．

5.3 主成分の取得法

ここでは前節で説明した画像サイズ縮小などの手順により得られた 2304 個の成分から，データベースを検索するための少数の主成分を抜き出す手順について説明する．

5.3.1 主成分変換行列の算出

まず，主成分分析を行い，データからデータベースの検索に対して有効な成分を抜き出すための変換行列（一般的には負荷行列と呼ばれる）を求める．主成分分析はデータを表している変数のばらつきを分析するものであり，画像に用いる場合は，各画素位置での画素値のばらつきなどが分析対象となる．図 5.3 は主成分分析を行うための手順を，データ解析ソフトウェアの多くがサポートする行列形式で表したものである．その手順は簡単であり，まず，入力データ行列を，(a) データベースに登録されている各画像の画素を，(b) のように一列に並び替え，これを (c) のように縦に積み重ねることで作成する．これを主成分分析（図中 PCA: Principal Components Analysis) にかけることで (d) の負荷行列（Loading matrix）と呼ばれる変換行列が得られる．

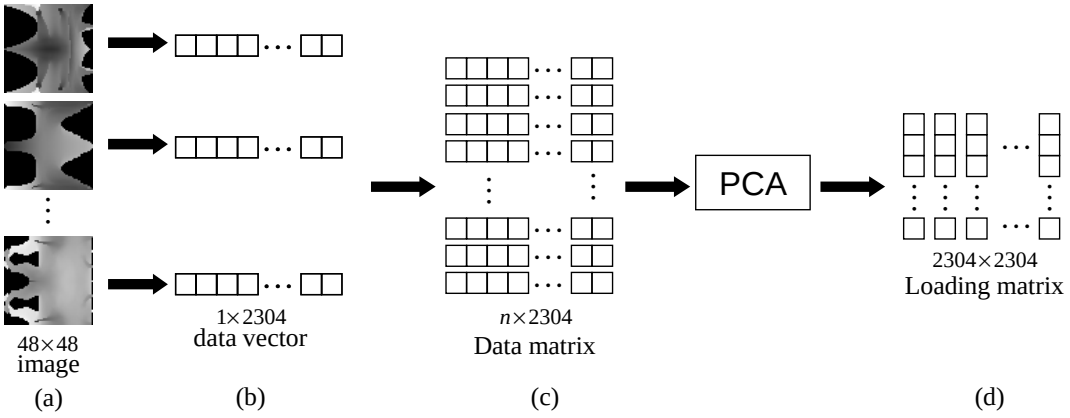


図 5.3: 主成分分析による負荷行列の取得．(a) データベースに登録されている画像．(b) 列ベクトル変換．(c) データ行列．(d) 負荷行列（Loading matrix）

主成分分析内部で行われる処理については詳しくは付録 A.2 で述べるが，まず，データ行列の各列ごとの共分散がとられ（各画素位置における画素値の変化の度合いが各画像間で比べられ），(2) この共分散行列の固有値分解が行われる．これにより固有値と固有ベクトルが求まるが，値の大きい固有値に対応する固有ベクトルから順に，データから主要な主成分を抜き出すための変換ベクトルとなる（固有値は主成分の分散の大きさを表す）．(3) 固有値の大きいものから順に固有ベクトルが並べ，行列として表したものが変換行列となる．

5.3.2　負荷行列によるデータの再構成

得られた負荷行列の各列ベクトル（固有ベクトル）が，データを主成分に変換する変換ベクトルの役割をするため，もしデータを主要な k 個の成分で表したければ，負荷行列の 1 列目から k 列目までを取り出し，データにかけ合せればよいことになる．図 5.4 はこの手順を示したものであり，(a) の負荷行列から (b) のように主要な変換ベクトルの組を取り出し，これを (c) のようにもとのデータベクトルにかけ合わせることで，(d) の検索に対して主要な成分のみが得られることになる．

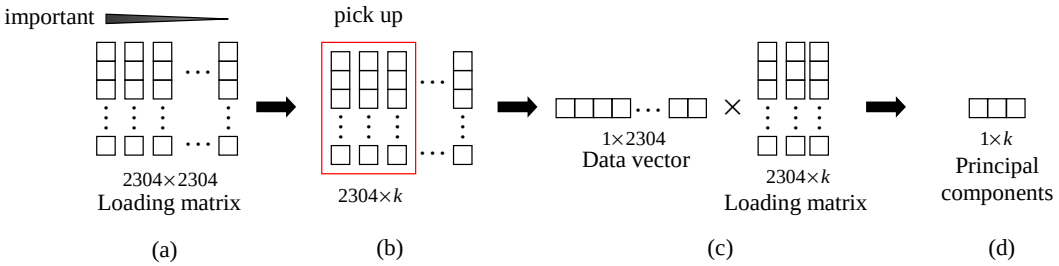


図 5.4: 負荷行列によるデータの次元の削減．(a) 負荷行列．(b) 主要変換ベクトル．(c) データベクトルの変換．(d) 得られた検索用の主成分

5.3.3　累積寄与率による主成分数の決定

ここで，どの程度の主成分を用いればよいのかということが問題となる[*1]．主成分分析では，通常，この判断のために寄与率と累積寄与率を用いる．寄与率とは各主成分がもとのデータに含まれる特徴をどの程度表現するかを表したもので，累積寄与率はこれを累積させたものとなる（計算方法については付録 A.2.2 を参照されたい）．ここではこの累積寄与率が 95%を越えるかどうかをその目安にした．

今回の実験では 162 体のモデル（Animal Kingdom[*2]）を用いる．このモデルを用いて累積寄与率を測定したものが図 5.5 であり，横軸が主成分数，縦軸が累積寄与率を表している．この図からは 95%の累積寄与率を得るには 46 個の主成分を用いればよいことが分かる．主成分数を 46 個とした場合，この個数は全成分数 2304 個の 20%であるので，マッチングにかかる時間は単純計算で 20%まで短縮されることになる．

[*1] 2304 個の列数を持つデータベクトを主成分分析した場合，2304 個の主成分が得られることになる．
[*2] Models ”Animal kingdom” are available for purchase from ”3D CAFE”: url:www.3dcafe.com/

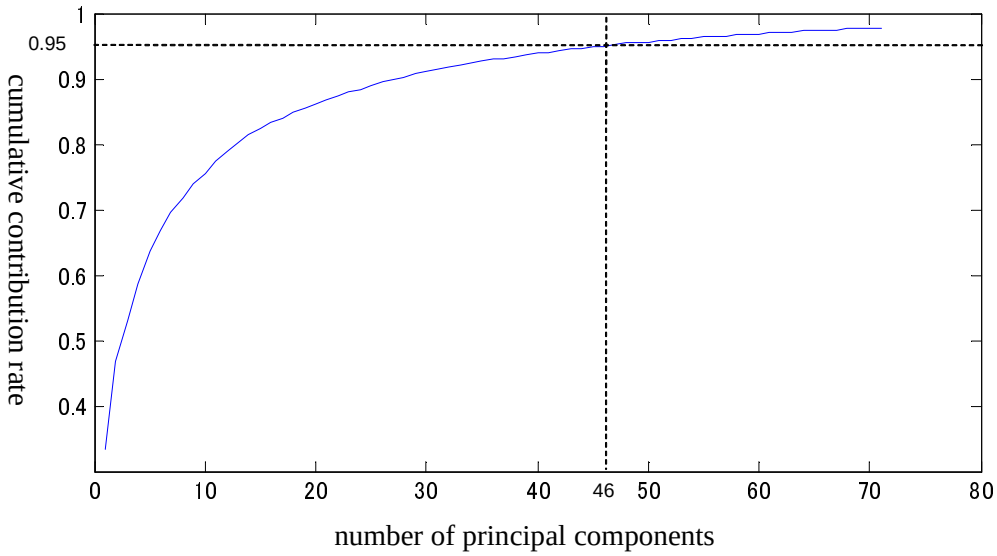


図 5.5: 累積寄与率と主成分数の関係

5.4　検索結果

3D モデル 162 体（Animal kingdom[*3]）を用いて検索実験を行った．図 5.6 はその一部のモデルを示したものである．また既存の検索法との検索精度と速度の比較のため，シルエット画像を用いた検索法 [Chen et al. 2003] を用いた検索も行った．

5.4.1　検索精度の比較

図 5.7 と図 5.8 に検索結果の一部を示す．各ブロックの一番左側の 3D モデルが検索キーで，その右側が検索されたモデルを表している．左側に近づくにつれ，類似度が高いと見なされたモデルとなる．なお，検索結果が上段と下段に分かれているが，上段が提案法のもので，下段が従来法の検索結果を示したものである．

検索結果であるが，従来法と提案法ともに第 1 候補，第 2 候補は検索キーに近いモデルを検索できていると思われる．また，両手法とも，主観的に見て形状が異なると思われるモデルを同程度含んでいるため，検出精度的にはさほど変わらないと考えられる．

累積寄与率と検索精度の関係　図 5.9 に異なる累積寄与率で検索を行った結果を示す．各行は上から累積寄与率 99%，95%，80%，60%での検索結果を表しており，各行が左から，検索キーのモデル，類似モデルの第一候補，第二候補，第三候補となる．実験の結果からは累積寄与率が 95%あれば主観的にも形状の類似したモデルを得ることができた．

5.4.2　検索速度の比較

表 5.1 に 1 体のモデルに対する類似モデルを検索するのに要した処理時間（投影処理とマッチング処理に要した合計時間）を表す．また，3D モデルの形状による検索では，検索キーとなる 2 次元画像を作り出す時間が，モデルの頂点数や三角形数によって変化すると考えられるため，頂点数の異なる十数体のモデルを用いた場合の検索速度を示した．

この表から，従来法が 1 体のモデルの類似モデルを検索するのに平均して 1 秒近くかかるのに対し，提案法はその約 500 分の 1 と大幅に検索速度が短縮されているのが分かる．これは従来法 [Chen et al. 2003] では多視点からの十数枚の画像を用いて，類似度を測る必要があるが，円筒ジオメトリ画像を用いた場合は，円筒画像 1 枚のみで済むためであると考えられる．

[*3]Models ”Animal kingdom” are available for purchase from ”3D CAFE”: url:www.3dcafe.com/

提案法の検索時間の内訳は，投影の要する時間が平均 0.015 秒，主成分のマッチング処理に要する時間が平均 0.001 秒であった．対象モデルの数が百体程度と少ないこともあるが，円筒投影法の投影速度が投影速度に与える影響は大きいといえる．

結果として，本論文で提案した円筒投影法を使用することで，シルエット法と同等の精度の検索を，数百倍の速度で行うことができる方法を示すことができた．

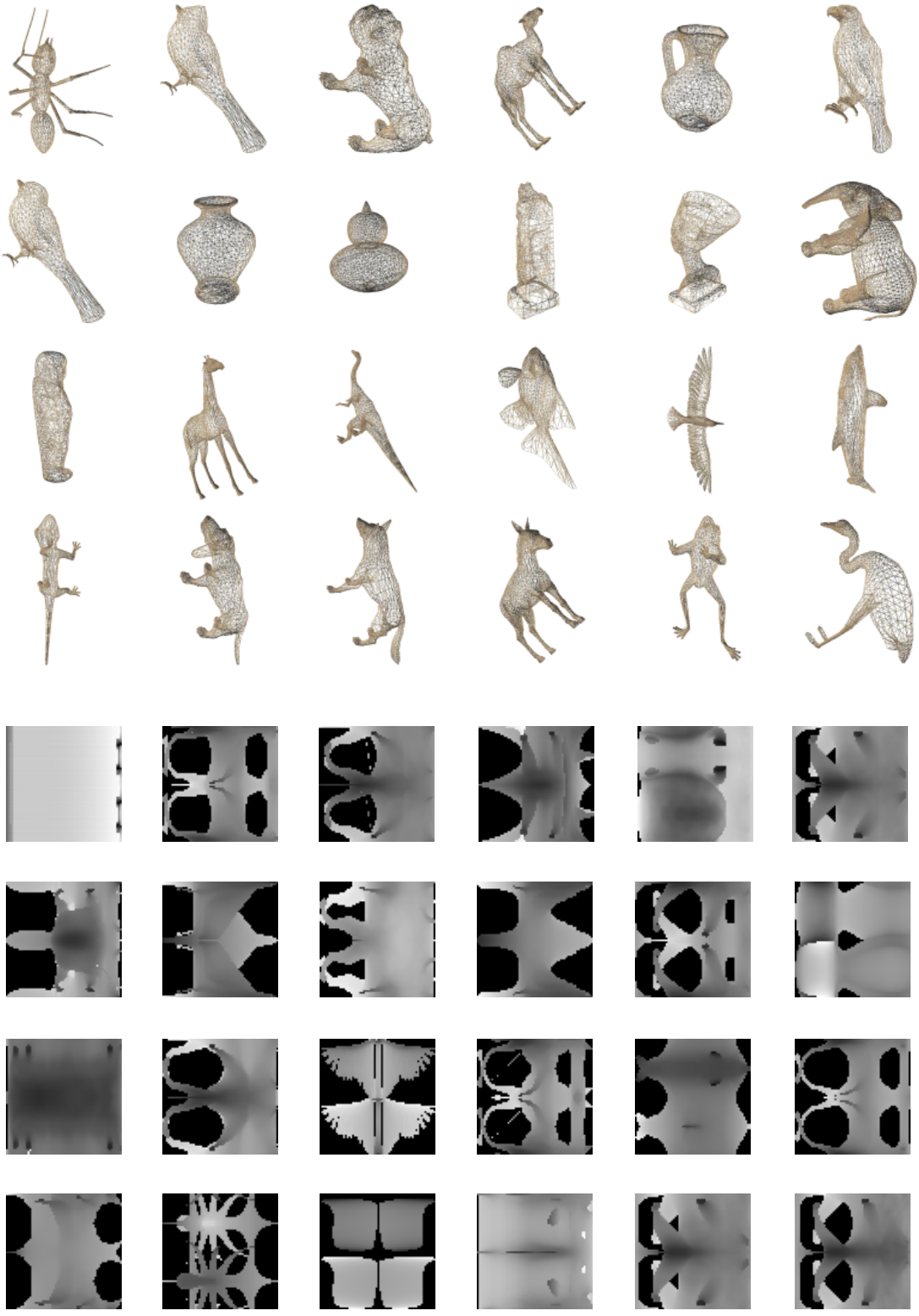


図 5.6: データベースに登録されている 3D モデルの一部．データベースには下段の円筒ジオメトリ画像が登録されおり，画像マッチングに用いられる

Models ”Animal kingdom” are available for purchase from ”3D CAFE”: url:www.3dcafe.com/

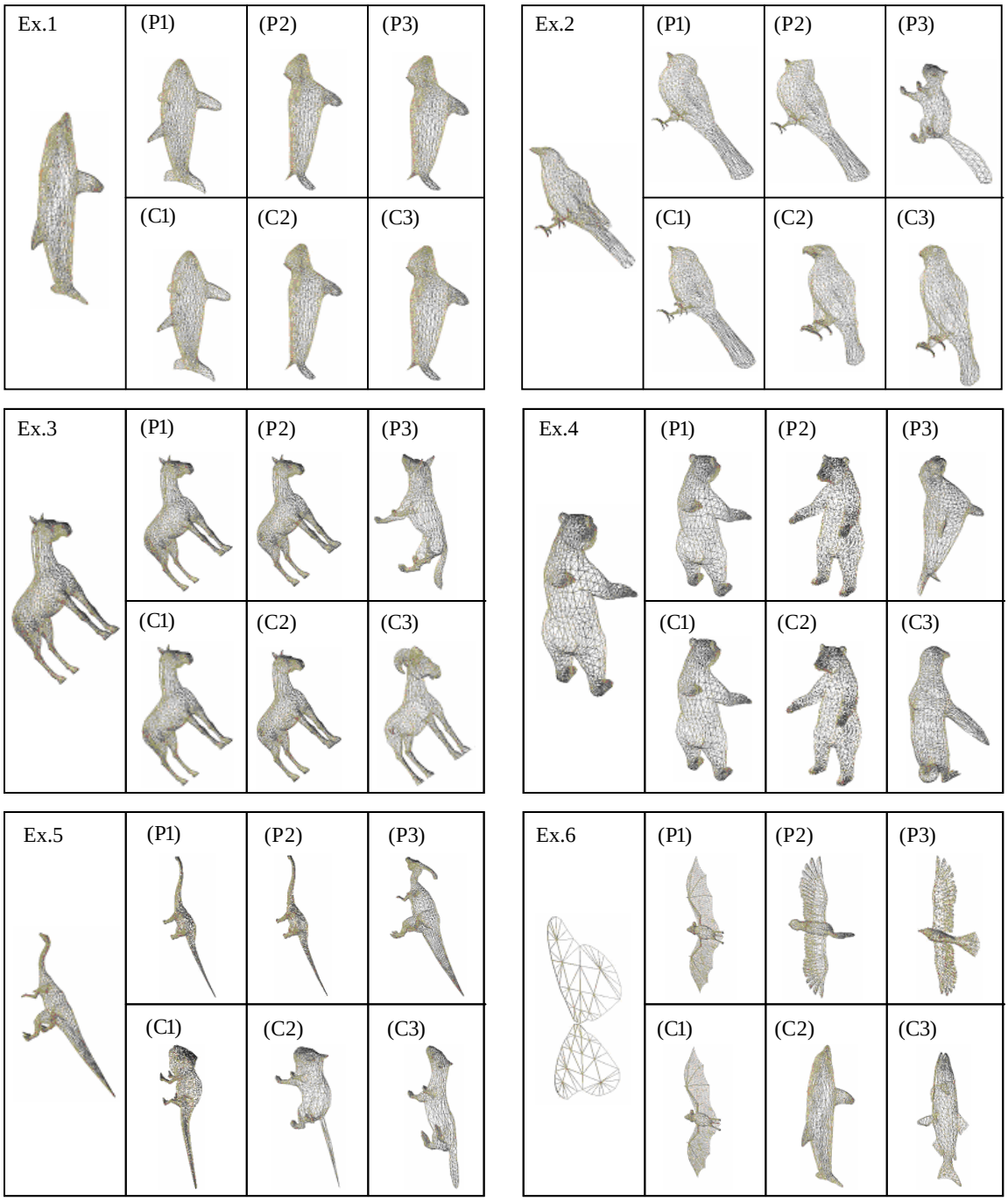


図 5.7: 検索結果 (1/2)．最も左の 3 次元モデルが検索キー．その右が検索されたモデルを表している．検索結果が上段と下段に分かれているが，上段が提案法のもので，下段が従来法 [Chen et al. 2003] の検索結果を示しめしている

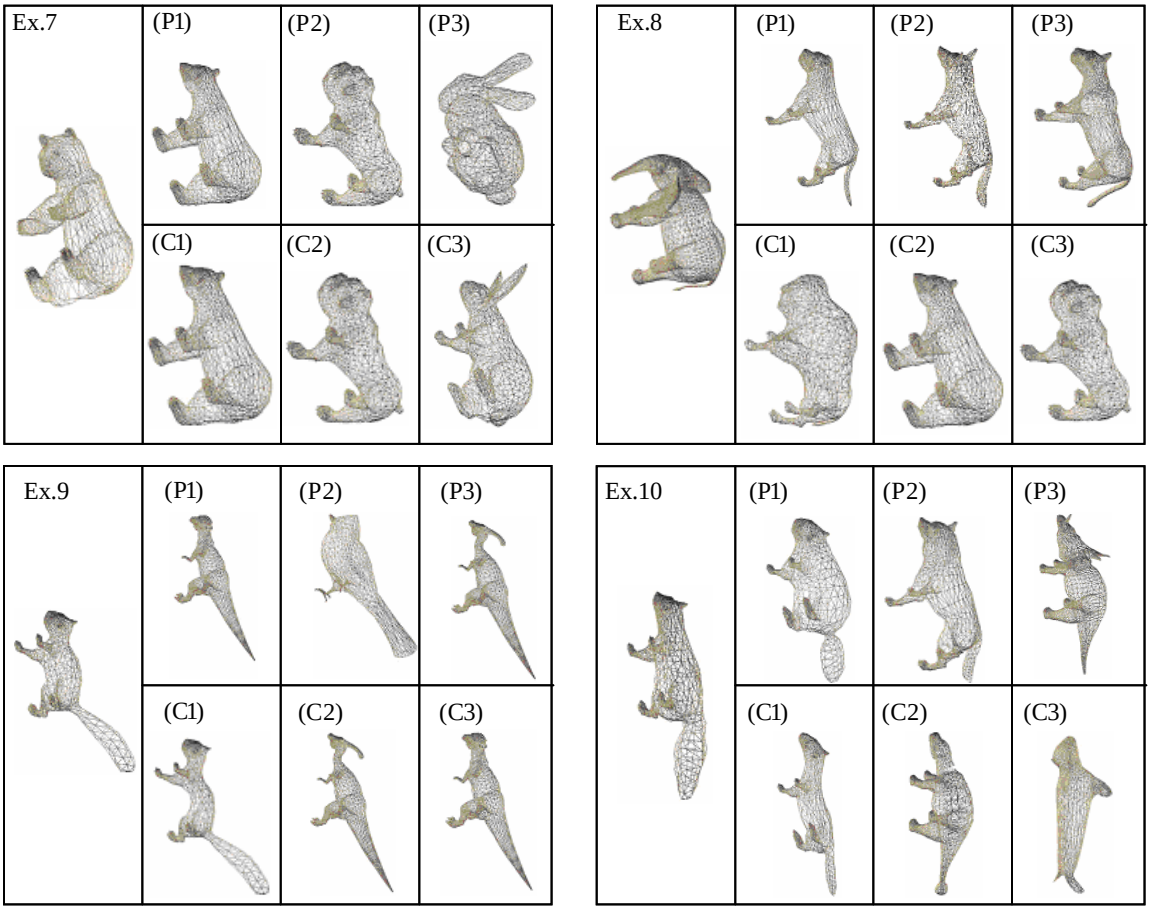


図 5.8: 検索結果 (2/2)．最も左の 3 次元モデルが検索キー．その右が検索されたモデルを表している．検索結果が上段と下段に分かれているが，上段が提案法のもので，下段が従来法 [Chen et al. 2003] の検索結果を示しめしている

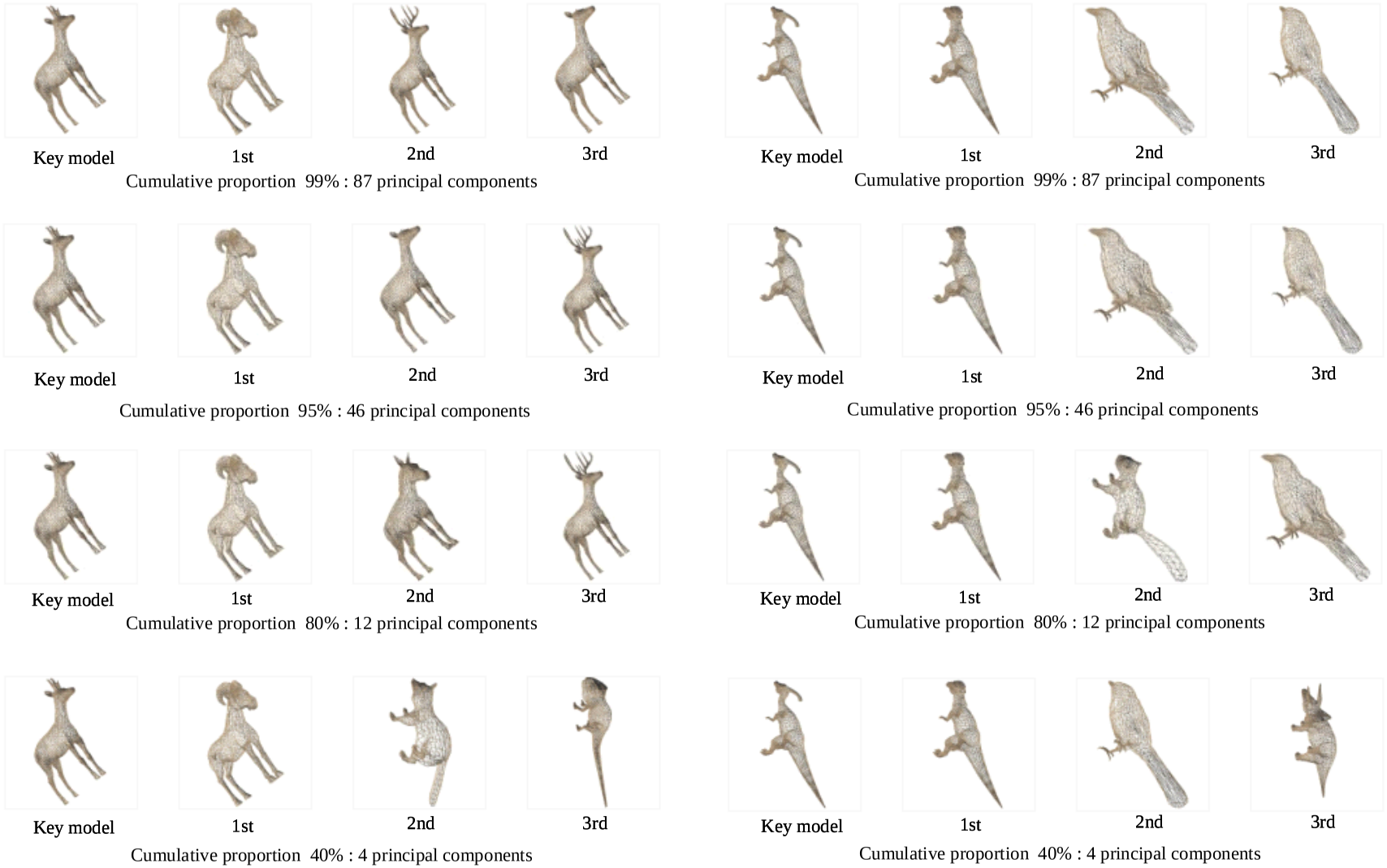


図 5.9: 累積寄与率と検索精度の関係．各行は上から累積寄与率（主成分数）が 99%(87)，95%(46)，80%(12)，60%(4) の結果を表しており，各行は左から，検索キーのモデル，類似モデルの第一候補，第二候補，第三候補となる．

表 5.1: 従来法 [Chen et al. 2003] との検索速度の比較

Numberof vertices	Conventional method (sec)	Our method (sec)
911	7.50	0.007
1501	5.78	0.008
2001	8.28	0.009
2580	7.65	0.009
3001	11.88	0.012
3539	7.49	0.013
3999	8.91	0.013
4504	6.88	0.015
4992	9.06	0.016
5422	8.13	0.017
5869	11.56	0.020
6431	8.38	0.021
6997	8.44	0.021
Average		
3520	8.14	0.015

5.5 本章のまとめ

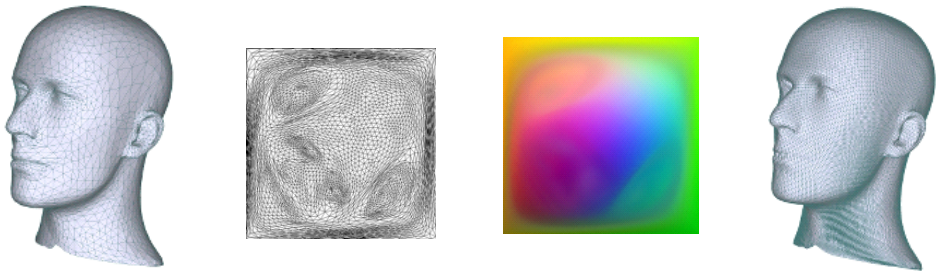
本章では，２次元円筒ジオメトリ画像を用いた 3D モデルの検索法を提案した．この手法は形状の類似したモデルではそのジオメトリ画像も似たものになることを利用したものであり，円筒 Z バッファ法による高速な画像化と，主成分分析を利用してマッチングに用いる成分数を減らすことで，シルエット画像を用いる 3D モデル検索法に比べて大幅な検索速度の短縮を可能とした．この他にも円筒ジオメトリ画像の利用法は多いと考えられる．

第6章

３次元メッシュ・パラメータ化によるジオメトリ画像化と形状復元

6.1　はじめに

本章では，今後の展望として，３次元メッシュ・パラメータ化を用いた形状復元法を示す．パラメータ化[*1]は，入り組んだ 3D メッシュを，一枚の重なりの無い２次元メッシュ（以下 2D メッシュ）へと変換する手法であり，近年の３次元 CG では，“ テクスチャマッピング ” のためのメッシュとテクスチャ画像の対応づけに用いられたり，前章で述べてきたメッシュのジオメトリ画像化に用いられ，基盤技術となっている．



(a)　(b)　(c)　(d)

図 6.1: パラメータ化によるジオメトリ画像化．(a) オリジナルメッシュ．(b) ２次元メッシュに変換された３次元メッシュ．(c) ３次元頂点情報座標を記録したジオメトリ画像．(d) 復元モデル

図 6.1 はパラメータ化による２次元化から形状復元までの処理を示したものであり，まず (a) のオリジナルモデルをパラメータ化したものが (b) の 2D メッシュとなる．ここで (a) と (b) の 3D メッシュと 2D メッシュ上の点は１対１の対応関係を持つため，(b) のメッシュを均一格子サンプリングすると，対応関係をもとに 3D メッシュ上の点の座標を得ることができる．この x, y, z の座標情報を画像の画素値に（カラーとして）記録することで (c) のジオ

[*1] ３次元の頂点座標を２次元パラメータ空間の媒介頂点座標へ変換するため，パラメータ化と呼ばれる．

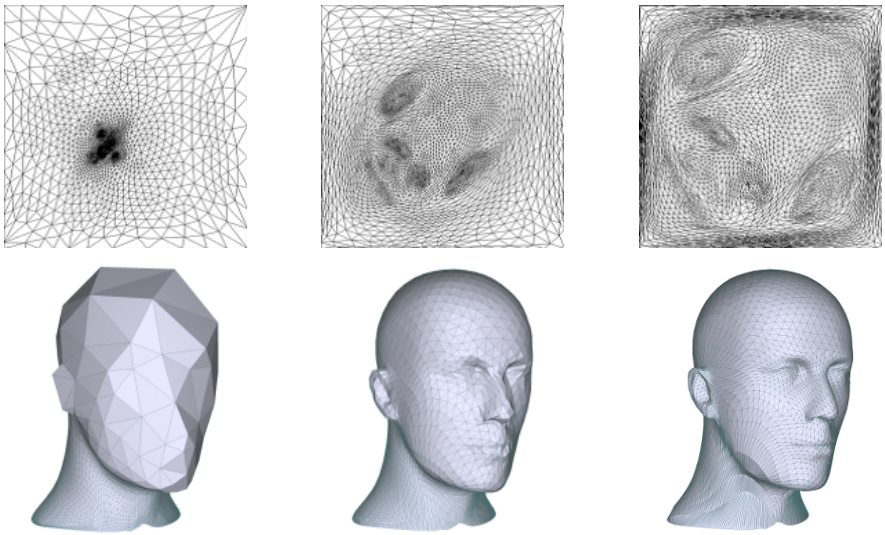
メトリ画像を作成することができる（ここでは表示用に x, y, z の値を R,G,B カラーで表した）．このジオメトリ画像からは本論文で述べてきた円筒ジオメトリ画像同様，もとのメッシュ形状に近いメッシュを復元することができる．パラメータ化を用いて作成したジオメトリ画像には，円筒ジオメトリ画像では欠落するモデルのオクルージョン領域の情報も含まれるため，円筒ジオメトリ画像から復元したモデルよりも精度のよい形状を得ることができる．

近年，パラメータ化に関する研究成果も充実し，その精度も向上している．ここで提案する手法は更にその復元形状の精度を向上させるものである [白井 啓一郎 et al. 2006b, Shirai et al. 2006b].

6.1.1 関連研究

パラメータ化は，メッシュの各頂点間を結ぶ辺の長さを最小化することで行われ，最小化の際に辺に重みをかけることでパラメータ化後の 2 次元メッシュを制御する．この重みのかけ方により図 6.2 のようにメッシュの疎密や頂点間を結ぶ辺の角度が変化する．パラメータ化の基礎的な手法としては，面積比を用いる [Floater 1997] をはじめとし，メッシュの角度を保存させる重みを用いる [Eck et al. 1995, Desbrun et al. 2002] などが知られている．しかし，これらは単純なメッシュを対象としているため，複雑なモデルに用いるとパラメータ化後の 2D メッシュに図 6.2 (a) のような極端な疎密が生じる．この場合，形状復元時に均一格子でサンプリングしても，密な部分が再現できないという問題が生じる．

この問題を解決するため，モデルをいくつかのチャートに分けてからパラメータ化する手法 [Gu et al. 2002, Sander et al. 2003, Zhang et al. 2003] や，メッシュの密な部分を拡げていくストレッチ法 [Sander et al. 2001, Sander et al. 2002, Yoshizawa et al. 2004] が提案されており，特にこのストレッチ法を用いてメッシュの密な部分を拡げていくことで，テクスチャマッピングと復元形状の精度を大きく向上させることができる（図 6.2 (b)(c) 参照）．



(a) none　(b) 3 time　(c) 20 time

図 6.2: 2D メッシュのストレッチと，得られる復元形状．(a) は初期パラメータ化，(b) は 3 回反復ストレッチを行ったもの，(c) は 20 回反復ストレッチを行ったものを表している

6.2 従来法の概要と問題点

[Sander et al. 2001, Sander et al. 2002, Yoshizawa et al. 2004] の手法ではメッシュを引き伸ばす際の目安として，方向や距離を考慮したメッシュの伸びを表す量を用いている．図 6.3 はこれを簡単に示したもので，パラメータ化で得られた図中 (a) の 2D メッシュ上では正方形である領域が，実際には 3D メッシュ上では図中 (b) のように長方領域領域に相当し，この変換では縦横の長さが Γ，γ へと変化することになる．従来法ではこの大きさ $\sqrt{\Gamma^2+\gamma^2}$ を最小化させることを目的とし，パラメータ化の際に用いる重みを変更して，パラメータ化を解きなおしていく．重みや計算方法については次項 6.3 にて述べる．

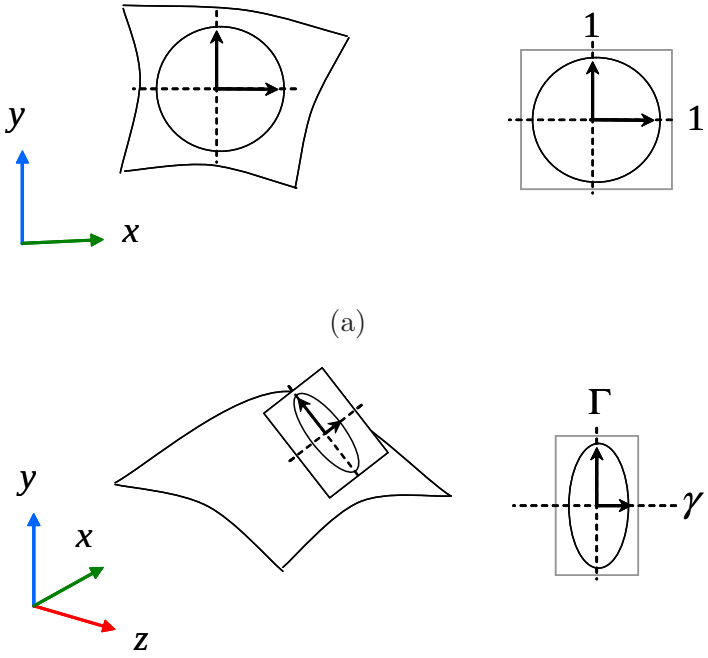
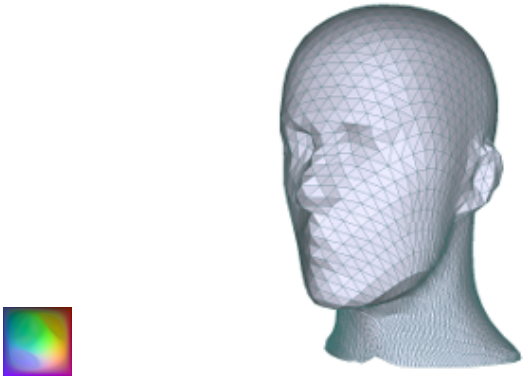
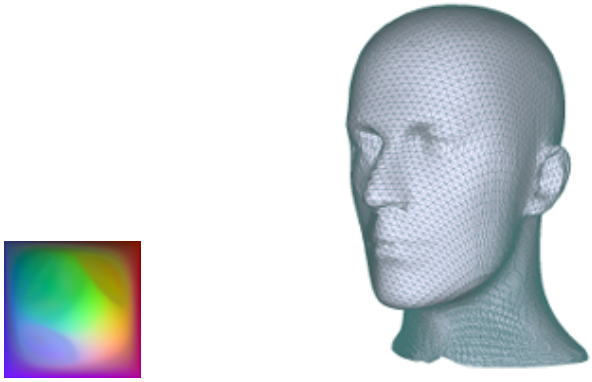


図 6.3: 2D メッシュと 3D メッシュ上でのメッシュの伸び

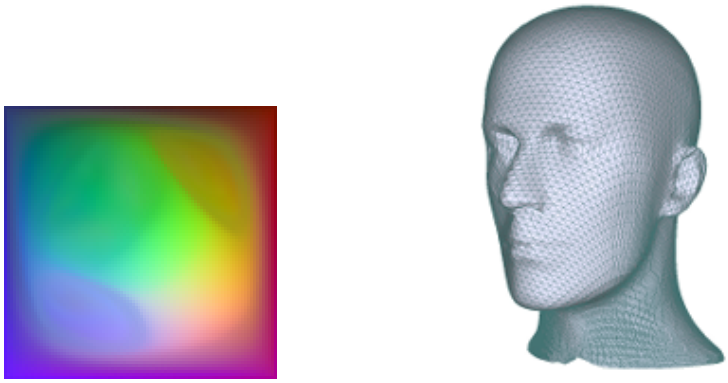
しかし，これらのストレッチ手法がメッシュを拡げる際に各辺にかける重みの方向性に対する拘束力が大きいため，場所によってはこの拘束によってメッシュの拡がりが制限され，この部分のリメッシュ時の形状が劣化してしまう問題がある．これを解決する単純な方法として，ジオメトリ画像の解像度を上げる方法があるが，この場合，情報量が増加するため，転送に時間を要してしまう．また，オリジナルモデルと同数の頂点数で，同様の形状を表現することは困難であり，通常，数倍の頂点数を必要とする．図 6.4 はジオメトリ画像サイズと復元されたメッシュの精度の関係を示したものである．



(a) Reconstruction from the 54×54 geometry image. 3K vertices.



(b) Reconstruction from the 108×108 geometry image. 11K vertices.



(c) Reconstruction from the 216×216 geometry image. 46K vertices.

図 6.4: ジオメトリ画像サイズと復元時の形状の精度の関係．(a) オリジナルモデルと同数の頂点数で復元したもの．(b)：(a) の縦横 2 倍の解像度で復元したもの．(c)：(a) の縦横 4 倍の解像度で復元したもの

6.3 パラメータ化のアルゴリズム

従来法の問題点を改善する方法の説明のために，パラメータ化のアルゴリズムや重みの更新法についての若干の知識が必要となるため，ここではパラメータ化の基本的なアルゴリズムについて説明する．

6.3.1 諸定義

まず，使用するメッシュや記号を定義する．3D モデルは円筒投影のときとは異なり，空間的に閉じておらず，分岐などもない 1 枚のメッシュと見なせるものとする．次に，メッシュの任意頂点を i で表し，この頂点に接続する他の頂点を $j \in \mathcal{N}(i)$ で表す．

6.3.2 問題の定式化：線形スパース方程式による解法

パラメータ化の多くは，頂点同士を結ぶ変の重み付き長さを最小化することで行われる．図 6.5 は簡単なメッシュのパラメータ化を示したものであり，重みが等しい場合には，パラメータ化によりすべての辺の長さが等しくなる．メッシュ全体のエネルギーを式で表せば，

$$E = \sum_i \sum_{j \in \mathcal{N}(i)} w_{i,j} \|\mathbf{u}_i - \mathbf{u}_j\|^2 \tag{6.1}$$

となる．ここで $\mathbf{u}_i$ は 2 次元頂点座標，$\|\cdot\|$ はユークリッドノルム，$w_{i,j}$ は各辺にかける重みを表している．この代表的な重みとして面積比などがある [Floater 1997]．このエネルギーを最小化する $\{\mathbf{u}_i\}$ を得るため，$\mathbf{u}_i$ で微分してゼロベクトルを与える時の $\{\mathbf{u}_i\}$ を考えれば，

$$\frac{\partial E}{\partial \mathbf{u}_i} = \sum_{j \in \mathcal{N}(i)} w_{i,j}(\mathbf{u}_i - \mathbf{u}_j) = \mathbf{0} \tag{6.2}$$

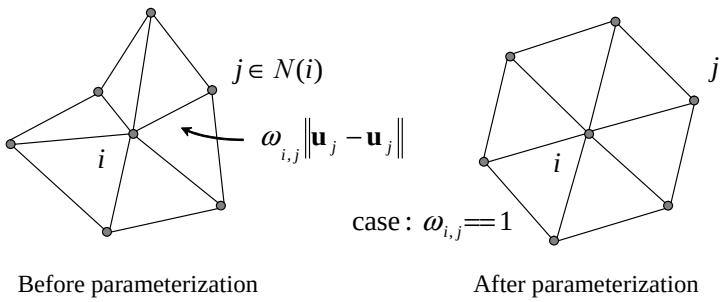


図 6.5: 簡単なメッシュのパラメータ化

が得られる．また，上式はスパース行列を用いれば，

$$\mathbf{WU} = \mathbf{0}, \qquad \text{where} \qquad W_{i,j} = \begin{cases} \sum_j w_{i,j}, & i = j \\ -w_{i,j}, & j \in \mathcal{N}(i) \\ 0, & \text{otherwise} \end{cases} \qquad \mathbf{U}_i = \mathbf{u}_i \tag{6.3}$$

と簡単に表すことができる．ただしこのままでは拘束が足りず，解は $\{\mathbf{u}_i\} = \mathbf{0}$ に収束してしまう．

拘束条件としては，メッシュ端の境界上の頂点をあらかじめ二次元平面上に固定することが一般的で，式 (6.3) に対して拘束条件を加えた場合，境界の頂点座標を縦に並べた行列 $\mathbf{B}$ と，$\mathbf{W}$ の境界の頂点に対応する行を書き換えた $\hat{\mathbf{W}}$ を用いて，

$$\hat{\mathbf{W}}\mathbf{U} = \mathbf{B} \qquad \text{where} \qquad \hat{W}_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \\ W_{i,j} & \text{otherwise} \end{cases} \quad i \in \text{boundary} \qquad \mathbf{B}_i = \begin{cases} \mathbf{u}_i & i \in \text{boundary} \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{6.4}$$

と書き直される．これによりパラメータ化された頂点座標を，

$$\mathbf{U} = \hat{\mathbf{W}}^{-1}\mathbf{B} \tag{6.5}$$

として得ることができる．なお，境界の頂点の配置 $\mathbf{u}_i$ は，基本的には境界の形状が凸型を満たしていれば配置は自由で，一般に円形や長方形が用いられる．

6.3.3　メッシュストレッチ

単にパラメータ化を行っただけでは，図 6.2 (a) のような粗密がメッシュに生じるため，パラメータ化されたメッシュを再度，重み $w_{i,j}$ を変えてパラメータ化することで，メッシュを拡げていく [Sander et al. 2001, Sander et al. 2002, Yoshizawa et al. 2004]．ここではこのことをメッシュストレッチと呼ぶ．

反復時の重みの更新は，以前の重み w^{t-1} に対して，現在の 2D メッシュの拡がり具合を表す量 $\sigma_{i,j}$ を乗算することで行われ，

$$\begin{aligned} w^t_{i,j} &= \sigma_{i,j} w^{t-1}_{i,j} \\ w^t_{i,j} &= w^t_{i,j} / \textstyle\sum_j w^t_{i,j} \end{aligned} \tag{6.6}$$

これによりメッシュの疎密が緩和され，図6.2 (b)(c) のように，リメッシュの結果が向上する．

6.3.3.1 反復ストレッチの終了条件

メッシュの拡がりが収束したかどうかは，$\sigma_{i,j}$ の値が 1 に収束したかどうかを調べることで行うことができる．

$$\frac{1}{n}\sum_{\{i,j\}}|\sigma_{i,j}-1|^2 < threthold \tag{6.7}$$

ここで $\{i,j\}$ はメッシュに含まれるすべての辺を表し，n はその総数を表している．

6.3.4 問題点

[Sander et al. 2001] や [Yoshizawa et al. 2004] では，$\sigma_{i,j}$ として，方向や距離を考慮したメッシュの伸びを表す量が用いられているが，この方向性に対する拘束力が強いため，場所によっては十分にメッシュを拡げることができず，このような部分のリメッシュ形状を劣化させてしまう．また，このように $\sigma_{i,j}$ の拘束力が強かったり，変化が大きいと，パラメータ化を繰り返す際に $\sigma_{i,j}$ が単調に収束せず，振動するなどの問題が生じる．

6.4　復元形状の精度向上のための提案法

6.4.1　頂点の周辺密度の均一化

形状復元のためのパラメータ化では，まず，2D メッシュを無駄なく均一に拡げ，ジオメトリ画像作成時の均一格子でのサンプリングで 3D モデル全体を表現することが要求される．更に，前述のように，メッシュを拡げる際に無駄な振動を生じさせないためには，メッシュの形状を徐々に変形させていく必要があり [Baraff and Witkin 1998]，メッシュの各辺にかかる拘束力も弱く，流動的に変化する $\sigma_{i,j}$ を用いる必要がある．

まず，このように柔軟かつ完全にメッシュを拡げるストレッチを実現させるため，ここでは各頂点の周囲の面積を徐々に均一化させていくことを考える．すなわち，ある頂点 i に接する三角形の合計面積 A_i を頂点の周辺密度とみなし，この周辺密度が等しくなるようにメッシュを変形させていき，最終的にすべての頂点の周辺密度を均一化する．しかし，各頂点に接続する三角形数や面積が異なるため，一括してこれらを近づけることは困難となる．このため，反復毎に，隣合う頂点のもつ周辺密度 A_i と $A_{j\in\mathcal{N}(i)}$ の平均化を繰り返すことで，全周辺密度を近づけていく．この平均化は，各頂点同士を結ぶ辺の長さ $r_{i,j}$ を $\sigma_{i,j}r_{i,j}$ へと変えることで行い，これにより面積が等方的に拡がるものと仮定する（図 6.6 参照）．

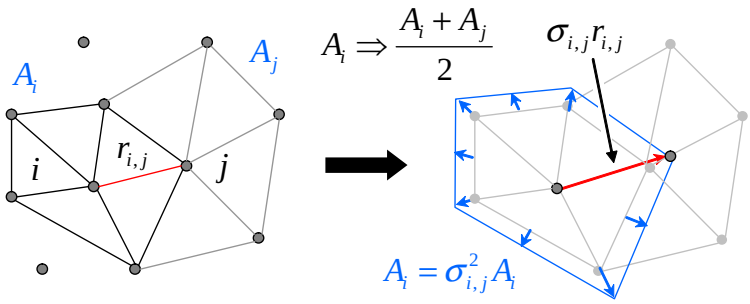


図 6.6: 辺の長さの比の変更による頂点周囲の三角形の合計面積の均一化

6.4.2　曲率の大きさに合わせた周辺密度の調節

更に，このように単にメッシュを拡げるだけでなく，3D モデルの形状の複雑さに合わせて 2D メッシュの拡げ具合を調節することを考える．これは，モデルの形状が平坦であれば，形状を復元するために多くのサンプルは必要なく，逆にそのような部分のメッシュの拡がりを抑えることで，形状の複雑な部分のメッシュを拡げるためのスペースができ，多くのサンプルを形状の詳細な表現のために回すことができるからである．

形状の変化の度合いを知るための尺度としては“ 主曲率 ”の二乗和を用いる．主曲率は図 6.7 のように，曲面の勾配の方向とその大きさを，その接平面上において直交する二つの主軸方向への大きさ k_1, k_2 を用いて表したもので，この二乗和 $k_1^2 + k_2^2$ は曲率の大きさを知るための一つの尺度として用いることができる．

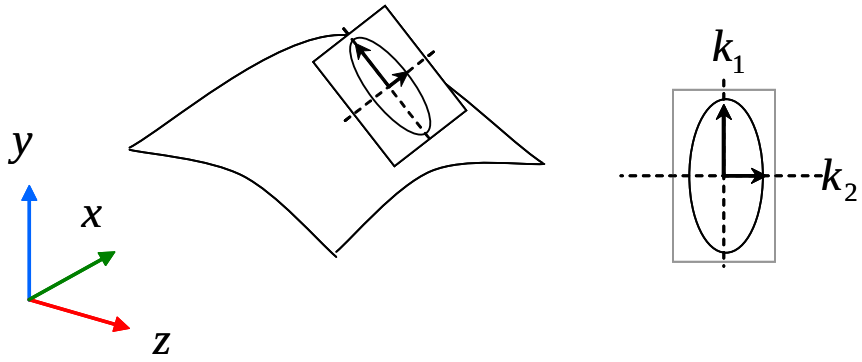


図 6.7: 主曲率

6.5　提案法アルゴリズム

6.5.1　周辺密度均一化のための反復用重みの導出

図 6.6 において，頂点 i に接続するある辺の長さ $r_{i,j}$ を $\sigma_{i,j}r_{i,j}$ へと変化させたとき，頂点周りの面積が等方的に拡がるとすると，面積 A_i は $\sigma_{i,j}^2 A_i$ へと変化する．このようにして面積 A_i を隣接頂点の面積 A_j との平均値 $(A_i + A_j)/2$ へと近づける場合，必要となる $\sigma_{i,j}$ は，

$$(A_i + A_j)/2 = \sigma_{i,j}^2 A_i \tag{6.8}$$

$$\sigma_{i,j} = \sqrt{(A_i + A_j)/2A_i}$$

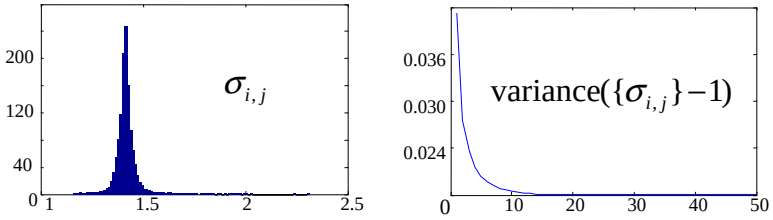
と求まる．これを頂点 i の周りの各頂点 $j \in \mathcal{N}(i)$ との間で求め，各辺に対する重み $\sigma_{i,j}$ とする．

ただし，境界の頂点は境界によって周囲の面積を削られているため，面積 A_j を次のように補正し，メッシュが拡がりすぎるのを抑える必要がある．

$$A_j = \begin{cases} 4A_j, & j = \text{corner} \\ 2A_j, & j = \text{boundary} \\ A_j, & \text{otherwise} \end{cases} \tag{6.9}$$

図 6.10 (a) はストレッチの結果を示したものであり，復元されたメッシュ（左），周辺密度を濃淡で表したもの（中央），もとのジオメトリがどのように拡がったかを示すための法線マップ（右）を表している．中央の図の面積密度を示した図の濃淡が一定なことから，提案法は十分にメッシュを拡げることができるといえる．

図 6.8 は反復時における重み $\sigma_{i,j}$ のヒストグラムと，収束の様子を示したものであり，この図からは $\sigma_{i,j}$ の変動も少なく，単調に収束していることが分かる．収束までの反復回数は，3D モデルの形状の複雑さにもよるが，多くのモデルでは 10 ～ 15 回ほどであった．



(a)　　(b)

図 6.8: 反復時に各辺にかかる重み $\sigma_{i,j}$ のヒストグラム (a) とその値が 1 へと収束する様子 (b)

6.5.2 形状復元時の頂点分布の正規化

特徴量を付加する前に，まず，形状復元時の頂点分布を均一化させるということを行う．図 6.10 (b) はその様子を示したものであり，このように，結果として得られる頂点分布，すなわち，“ 3D メッシュ上での面積密度 ”が均一となる状態に 2D メッシュの面積密度を一度正規化しておくことで，曲率の大きさに合わせて 2D メッシュを拡げる際，同程度の特徴量で，どの頂点周囲のメッシュも同じだけ拡がるようにする．

3D メッシュ上での頂点分布の均一化は，簡単に行うことができ，3D メッシュのもつ頂点の周辺密度の大きさに合わせて 2D メッシュを拡げ，形状復元時のサンプル数を調節することでできる．実際には 2 次元の面積 A_i と，もとの 3 次元の面積 $A_{i\ \mathrm{3D}}$ の比を一定にすればよく，パラメータ化の反復毎にこの面積比を均一化することで行うことができる．

$$A^t_{i\ \mathrm{uniform}} = \frac{A^t_i}{A_{i\ \mathrm{3D}}} \tag{6.10}$$

この $A^t_{i\ \mathrm{uniform}}$ を式 (6.8) と式 (6.9) で A^t_i の代わりに用いることで，形状復元時に頂点が一様に分布するようになる．

6.5.3 曲率の考慮

次に主曲率の大きさ $\sqrt{k_1^2 + k_2^2}$ を考慮していく．この主曲率 k_1，k_2 は，面の折れ具合を示す平均曲率 H と，面の凹凸の度合いを示すガウス曲率 K から求められ，次のような関係があり，

$$\begin{aligned} H_i &= \frac{k_{i1} + k_{i2}}{2}, \qquad K_i = k_{i1}k_{i2} \\ k_{i1}^2 + k_{i2}^2 &= 4H_i^2 - 2K_i \end{aligned} \tag{6.11}$$

平均曲率 H とガウス曲率 K は，離散曲面における近似曲率 [Garimella and Swartz 2003, Desbrun et al. 1999] を用いた場合，次式で与えられる．

$$\begin{aligned} H_i &= \frac{1}{4A_{i\ \mathrm{3D}}} \| \sum_{j \in \mathcal{N}(i)} (\cot \alpha_j + \cot \beta_j)(\mathbf{x}_i - \mathbf{x}_j) \| \\ K_i &= \frac{1}{A_{i\ \mathrm{3D}}} (2\pi - \sum_{j \in \mathcal{N}(i)} \theta_j) \end{aligned} \tag{6.12}$$

ここで $\mathbf{x}$ は頂点座標，α_j, β_j は頂点 i, j を結ぶ稜線の両側の角度（図 6.9 参照），θ_j は頂点 i に接している部分の三角形の角度を意味している（同図参照）．

ただし，メッシュ端の頂点では，上記の式からは曲率を正しく求められないため，提案法では単純に周囲の頂点 $j \in \mathcal{N}(i)$ のもつ $k_{j1}^2 + k_{j2}^2$ の平均値を用いた．境界の頂点はパラメー

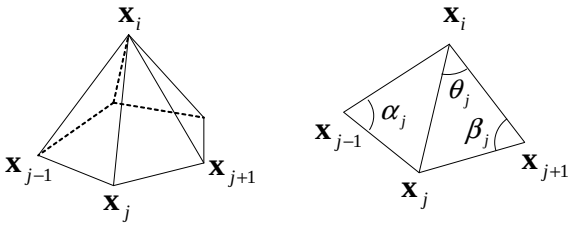


図 6.9: 曲率を求める際に必要な変数

タ化の際にはあらかじめ固定されているため，このように平均値を用いても結果に特に支障は見られなかった．

6.5.3.1 曲率に対する補正

式 (6.11) と式 (6.12) から $k_1^2 + k_2^2$ が求まるが，実際にはこの値の変化が大きすぎるため，ストレッチに直接用いると，メッシュの拡がりに大きな偏りが出てしまう．そこで経験的に N 乗根 ($N = 3 \sim 4$) をとることで値を調節し，最適値とする．

$$C_{i\ \mathrm{optim}} = (k_{i1}^2 + k_{i2}^2)^{\frac{1}{N}} \tag{6.13}$$

また，この値がもつ局所的な誤差を減らすため，以下に示す特徴領域を保護した平滑化（diffusion）[Tomasi and Manduchi 1998] を行った．

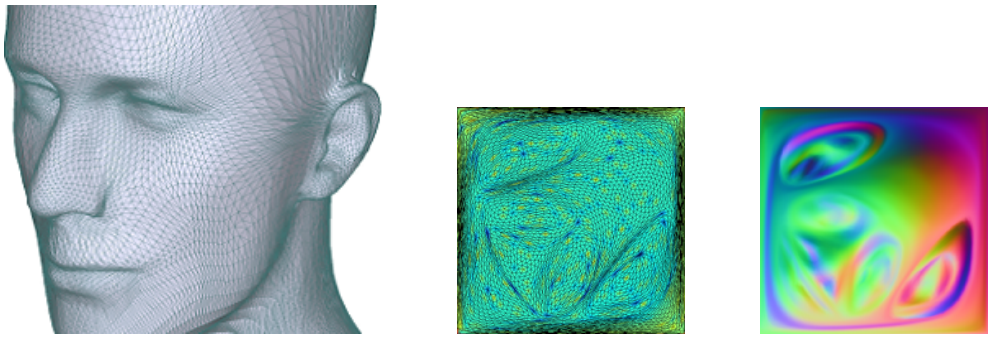
$$\begin{aligned} C_{i\ \mathrm{smooth}} &= C_i + \frac{1}{n_i} \sum_{j \in \mathcal{N}(i)} L(\|\mathbf{x}_i - \mathbf{x}_j\|) D(C_i - C_j)(C_i - C_j) \\ L(x) &= \exp(-\frac{x^2}{2\sigma_L^2}) \\ D(x) &= \exp(-\frac{x^2}{2\sigma_D^2}) \end{aligned} \tag{6.14}$$

ここで n_i は頂点 $j \in \mathcal{N}(i)$ の数を表し，L と D は頂点間の距離と差分に対する重み関数を表している．分母の σ_L，σ_D を大きくするほど平滑化が早まることになる．提案法で用いた 3D モデルの場合，この値を $\sigma_L = 0.1$，$\sigma_D = 2.0$ に設定し，2 回この平滑化を行った．

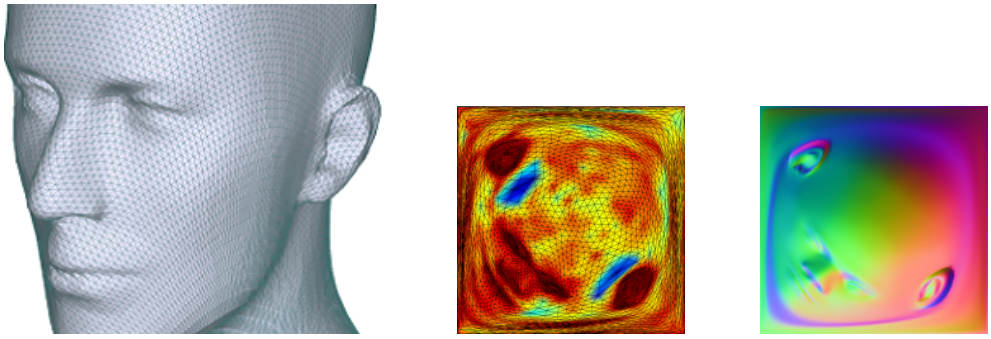
このようにして求めた曲率をストレッチに影響させるには，

$$A_{i\ \mathrm{optim}}^t = \frac{A_{i\ \mathrm{uniform}}^t}{C_{i\ \mathrm{optim}}} \tag{6.15}$$

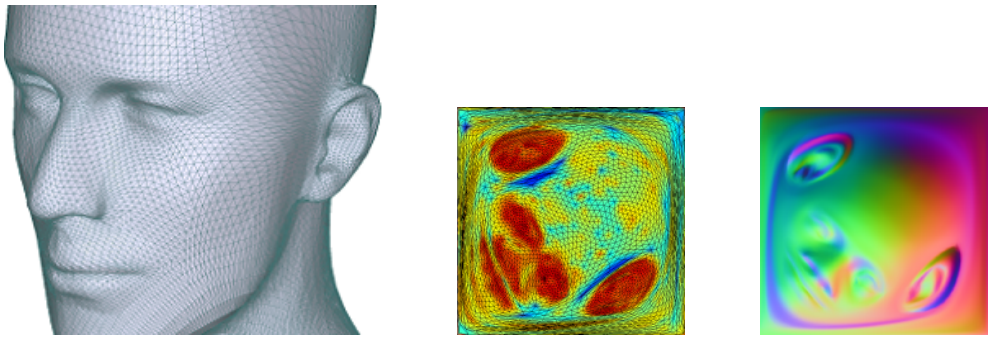
とすればよく，$C_{i\ \mathrm{optim}}$ の値が大きいほど，頂点の周辺密度 A_i は小さいとみなされるので，周囲からは面積を大きくする力がかかり，メッシュが拡がるようになる．この $A_{i\ \mathrm{optim}}^t$ を用いることで，図 6.10 (c) のように形状に特徴のある部分に適度に頂点が集まり，細部が鮮明な復元形状を得ることができる．



(a) 2D mesh density uniformization



(b) 3D mesh density uniformization.



(c) With the curvature consideration.

図 6.10: 提案法の各段階での復元形状．各列はそれぞれ，復元形状（左），パラメータ化時の 2 次元メッシュのメッシュの密度（中央），ジオメトリ情報（法線）の分布（右）を表している．中央の図の濃淡は周辺密度 A_i の大きさを表しており，青に近いほど密度が低く，赤に近いほど密度が高いことを意味する．各行がそれぞれ，2D メッシュの面積密度を均一化したもの（上段），復元時の 3 D メッシュの面積密度を均一化したもの（中段），曲率の大きさにあわせて面積密度を最適化したもの（下段）を表している．復元のために用いたジオメトリ画像のサイズは 128×128，復元モデルの三角形数は 32K 個.

6.6　結果

他の 3D モデルを使用した場合の，提案法と従来法との比較結果を図 6.11 と図 6.12 に示す．左からオリジナルモデル，従来法，提案法の順となっている．従来法としては，高い形状復元度をもつ [Yoshizawa et al. 2004] の手法を用いた．復元モデルの頂点数はオリジナルモデルとほぼ同一とし，メッシュの復元は，格子状に再配置した頂点を上下左右の頂点と結んで正方メッシュとした後に，3D メッシュの誤差を“ Metro ”[Cignoni et al. 1998] を用いて計測するため，3 次元での距離が短い対角頂点同士を結んで準正則メッシュ[*2]とした．

結果は，従来法に比べ形状に特徴のある部分に頂点が集まり，もとの 3D モデルの特徴が十分表現されていることが分かる．特に従来法では欠けてしまっているメッシュの角の部分なども，しっかりと表現されている．

6.6.1　形状の数値的評価

表 6.1 は Metro を用いて測定したオリジナルモデルと復元モデルの間の二乗平均誤差を示しており，全体として提案法の方が誤差が低く，精度が向上するという結果となった．なお，従来法の最適回数は形状誤差が最も小さいときのものとした．

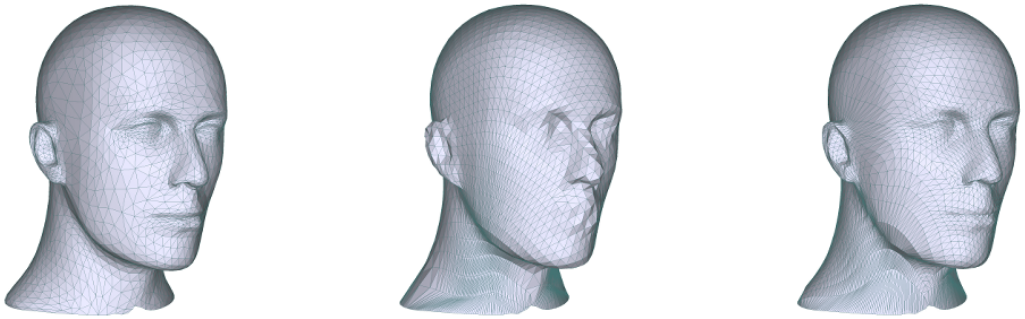
表 6.1: Metro [Cignoni et al. 1998] による従来法 [Yoshizawa et al. 2004] との形状誤差の比較．計測設定はサンプリング法：similar triangles sampling，計測法：Reast Mean Square error．表の数値はすべて $\times 10^{-3}$ 倍してある．

	Mannequin 5K face	Horse 20K face	Triceratops 8K face	Gargoyle 43K face
Conventional	3.768	0.387	0.913	1.822
Ours	3.466	0.335	0.712	1.252

6.6.2　処理時間

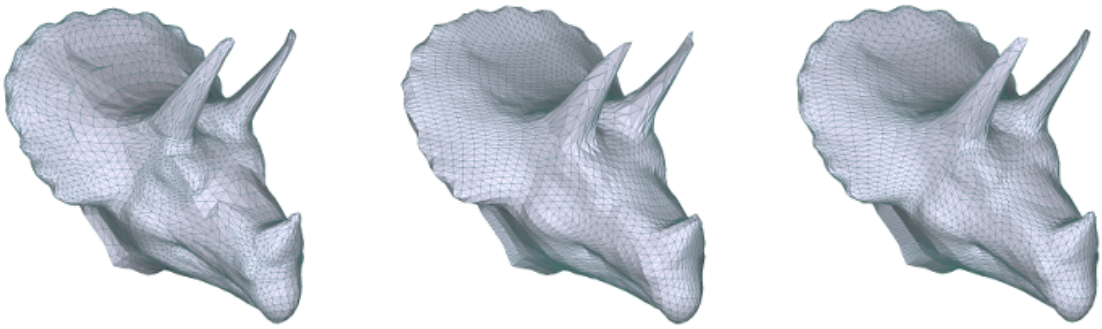
最適状態（定常状態）へ収束するための反復回数は，従来法が 5 回程度であり，これに対し提案法はメッシュの三角形数や面積のばらつきに依存し，三角形数 5K 個の Mannequin で 14 回，8K 個の Triceratops で 6 回，20K 個の Horse で 19 回，40K 個の Gargoyle で 9 回であった．1 回あたりの反復に必要な時間は，MATLAB 上での計測であるが，従来法とほとんど変わらず，Pentium4 3.8GHz の PC において，Mannequin で 0.85 秒，Triceratops で 1.20 秒，Horse で 3.28 秒，Gargoyle で 6.80 秒であった．

[*2]基本的には頂点の周囲の頂点との接続価数が一定の正則なメッシュであるが，一部，接続価数が異なる．



(a) Original (b) Conventional $\mathcal{U}^3$ (c) Our method $\mathcal{U}^{14}$

Mannequin head (3K vertex) and the reconstructed models (3K vertex) from 40×40 geometry maps.

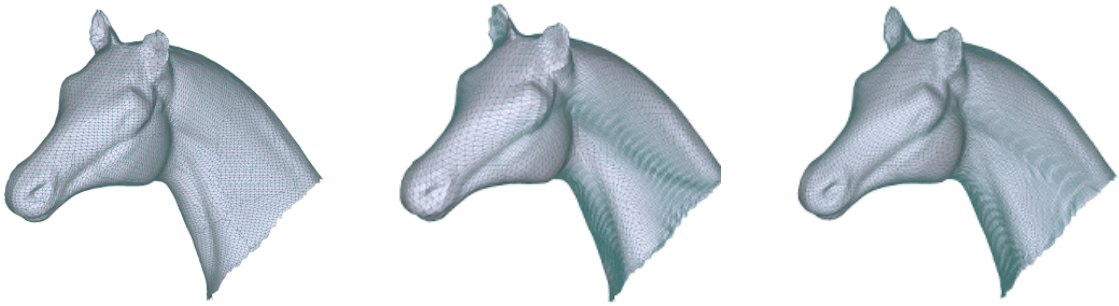


(a) Original (b) Conventional $\mathcal{U}^5$ (c) Our method $\mathcal{U}^6$

Triceratops head (4K vertex) and the reconstructed models (4K vertex) from 64×64 geometry maps.

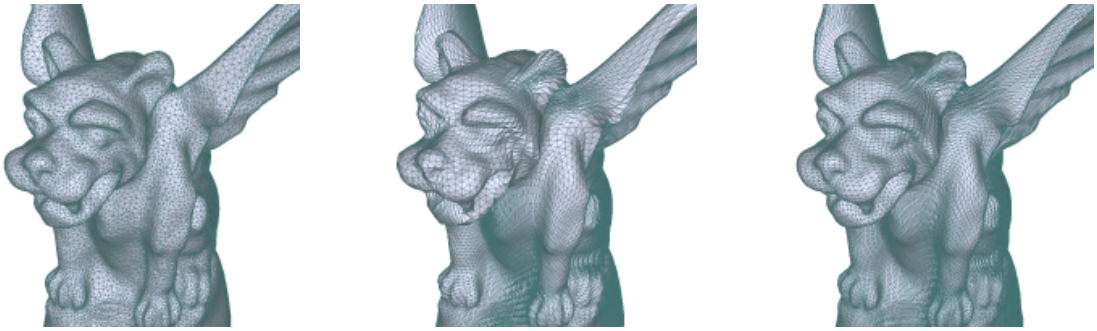
図 6.11: 従来法 [Yoshizawa et al. 2004] との復元形状の比較と，座標変換マップによるテクスチャマッピング結果の改善の様子 (1/2)．(a) オリジナルモデル．(b) 従来法による復元形状．(c) 提案法によって得られた復元形状

Models are courtesy of Washington University (Mannequin head) and Viewpoint Inc. (Triceratops).



(a) Original　(b) Conventional $\mathcal{U}^7$　(c) Our method $\mathcal{U}^{19}$

Horse head (10K vertex) and the reconstructed models (10K vertex) from 100×100 geometry maps.



(a) Original　(b) Conventional $\mathcal{U}^3$　(c) Our method $\mathcal{U}^9$

Gargoyle (21K vertex) and the reconstructed models (21K vertex) from 146×146 geometry maps.

図 6.12: 従来法 [Yoshizawa et al. 2004] との復元形状の比較と，座標変換マップによるテクスチャマッピング結果の改善の様子 (2/2)．(a) オリジナルモデル．(b) 従来法による復元形状．(c) 提案法によって得られた復元形状

Models are courtesy of Cyberware (Horse) and Microsoft Inc. (Gargoyle).

6.7 本章のまとめ

本章では，パラメータ化を用いたジオメトリ画像の生成法について述べ，精度向上のための改善法を示した．通常のパラメータ化を用いた場合にパラメータ化後の 2D メッシュに生じる疎密をなくすため，まず，各頂点周りの周辺密度が平均化するようにパラメータ化を繰り返し行う手法を示し，十分にメッシュの疎密を均一化できることを示した．次に，3D メッシュのもつ曲率の大きさを考慮して，周辺密度を調節することで，曲面を表すのに適した頂点数をリメッシュ時に得られるようにした．

なお，近年では平面へのパラメータ化の他にも，球面へのパラメータ化などが研究されている．これは 3D モデルの多くは球面的に閉じたメッシュである場合が多く，球面で表すのが自然であると考えられているためである．このような多様なパラメータ化に対してメッシュの拡がりをどのように制御するかということが今後の課題といえる．

第7章

結論

本論文では，２次元円筒ジオメトリ画像の作成法について述べ，ジオメトリ画像から復元される形状精度の向上を目的とした円筒投影法と，ジオメトリ画像の作成速度の高速化を目的とした２つの新しい円筒投影法の提案を行った．また，円筒ジオメトリ画像を利用した3Dモデルの検索法も新たに提案した．以下に本研究で行った提案法と得られた成果をまとめる．

まず，復元形状の精度向上を目的とした円筒投影法として，円筒面に対するオクルージョン領域を捕捉するために，通常は１回のみ行う光線追跡を階層的に行う階層的円筒レイキャスト法を提案した．この方法は画素から放たれた光線と3Dモデルとの交点を求め，この交点同士の中点から更にまた光線追跡を行うという簡単なものであったが，中点から新たに光線を放つ際に，中点がモデル外部にあるか内部にあるかによってその方向を変更する必要があったため，交点間を結ぶ直線と3Dモデルの表面の交差回数を用いてこれを判断する方法を示した．

この階層的な光線追跡を用いることで，オクルージョン領域の復元精度が改善され，全体としても復元精度が大幅に向上する結果となった．また，この手法により得られるジオメトリ画像は，画像符号化で用いられる多重解像度表現を行った画像の状態に近くなるため，既存の画像符号化法を用いることで効率的に圧縮することが可能である．

次に，ジオメトリ画像化速度の高速化を目的とした円筒投影法として，従来の計算コストの高い光線追跡法の代わりに，Ｚバッファ法を用いる円筒Ｚバッファ法を提案した．この方法では，三角形を特定して投影処理を行うため，光線追跡法のように不特定多数の三角形との交差判定が必要なく，高速に投影処理を行えることが利点となる．ただし，Ｚバッファ法は平面スクリーンへの投影法であるため，円筒スクリーンに対して適用する場合には，投影方向に関する問題と，投影面が湾曲することによってひずむ三角形内部の補間に関する問題が生じる．このため，これを判断するための式と，改善のするための手順を示した．

この円筒 Z バッファ法を用いることにより，ジオメトリ画像の作成速度は従来法の約 40%まで短縮されることになる．また，比較に用いた従来法も高速化を施したものであったので，実際には 40%よりも更に短縮されており，大幅な速度面での改善が実現できたといえる．

また，円筒ジオメトリ画像を用いたアプリケーションとして 3D モデルの検索法を示した．類似した形状を持つ 3D モデルでは，中心軸からの半径の変化も近くなり，円筒ジオメトリ画像も類似するため，ジオメトリ画像同士の画像マッチングにより，3D モデルの検索が行える．ここでは本研究で提案した高速な円筒投影法に加え，主成分分析を用いた画像マッチング法により高速化を図った．

この検索精度と速度を測定するため，実際に百体以上の 3D モデルを用いた実験を行った．また，既存のソフトウェアとして存在する手法との比較実験も行った．結果は既存の検索法と同等の精度であったが，検索速度は劇的に上回っており，約 500 倍の検索速度を得ることができた．これらの結果からも，本論文で示した円筒投影法の有効性は十分に示されているといえる．

最後に，3 次元メッシュパラメータ化法によるジオメトリ画像化法を示した．形状の複雑さに合わせ，パラメータ化された 2 次元メッシュの密度を調節することで，より小さいサイズのジオメトリ画像から形状を復元することが可能となる．この手法で得られたジオメトリ画像を 3D モデル検索結果の出力画像とすることで，検索結果の 3D 表示などが可能となる．これによりジオメトリ画像のみで構成された新たな検索システムを構築することができると考えられる．

提案した円筒投影法は基礎的なものであるが，他の諸研究と組み合わせることで新たなシステムを構築できる期待は大きい．近年では円筒投影法とポイントレンダリングを組み合わせて 3D モデルのリアルタイム伝送を行う手法 [梶谷 将治 et al. 2005] も提案されおり，本手法がこれらの手法の精度向上に貢献し，技術の向上につながることが期待できる．

付 録A

本論文で引用した手法の補足説明

A.1　はじめに

本章では，本論文内で引用した手法で，本文内では詳細に説明ができなかったものについての補足説明を行い，他の参考文献を読まずに論文内で示した手法の実装を行えるようにする．

A.2　主成分分析

観測されたサンプルのデータは通常，いくつかの変数を用いて $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ と表されることが多い．例えば，頂点座標などは $\mathbf{p} = \{x, y, z\}$ として 3 変数で表される．主成分分析（PCA：Principal Component Analysis）はこのようなサンプルに含まれる特徴のある変数を知るために便利であり，これを用いてサンプルの分布の傾きなどを知ることができる（図 A.1 参照）．また，影響力の強い変数のみを用いてデータを再構成するなどといったことにも用いられる．

本節では主成分分析を用いた“ 3D モデルの軸あわせ ”（本文 2.5.1 節参照）と“ 代表的な変数によるデータの再構築 ”（本文 5.3.1 節参照）を行うための計算法について簡単に説明する．ただし，ラグランジェ乗数法などを用いた計算過程の証明についてはここでは省略する．

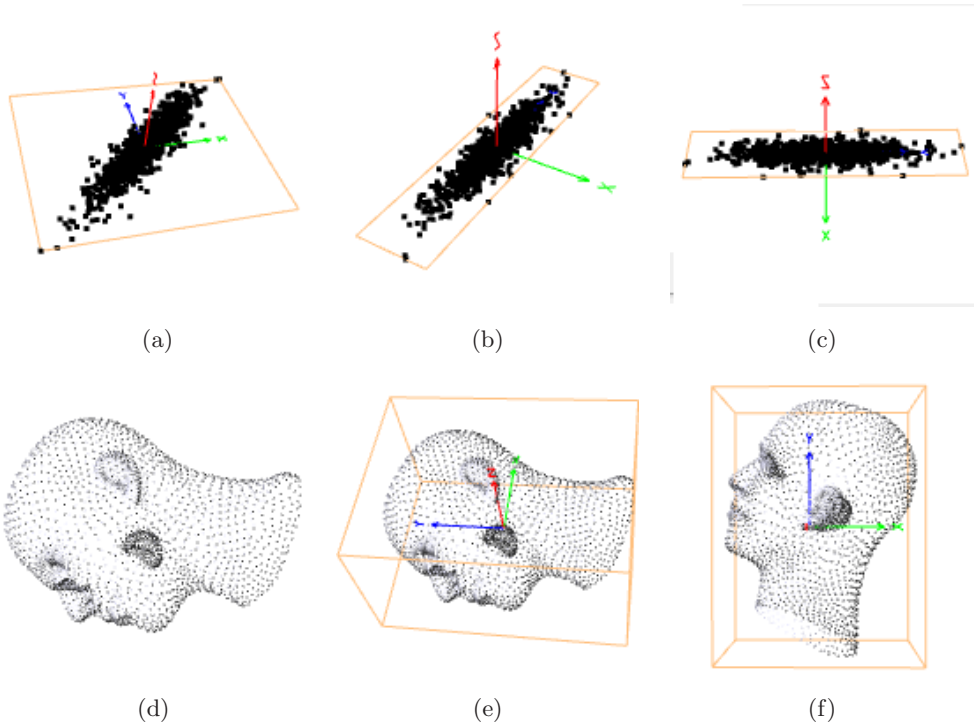


図 A.1: 主成分分析を用いたサンプルの軸合わせ．観測されたサンプルは通常 (a) のように分布に特徴をもつ場合が多い．主成分分析を用いるとこのようなンプルの傾きの主軸を得ることができる (b)．また，この主軸に現在の軸を合せる変換行列も得られるため，(c) のようにサンプルの分布を最も強く表す座標系へとデータの座標を変換させることができる．(d)(e)(f) は 3D モデルの頂点分布を用いた場合の例を示したものである．

A.2.1 共分散行列の取得

まず，サンプルの分散の方向とその度合いを知るため，観測データの各変数間の共分散を求めることが必要になる．共分散行列は次の手順で簡単に求められる．

観測データを n 個の変数をもつ列ベクトル $\mathbf{x} = [x_1, x_2, ..., x_j, ..., x_n]$ で表し，m 個のすべてのサンプルを行方向に積み重ねて行列で表現する．

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_i \\ \vdots \\ \mathbf{x}_m \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,j} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & & x_{2,j} & & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ x_{i,1} & x_{i,2} & \dots & x_{i,j} & \dots & x_{i,n} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \dots & x_{m,j} & \dots & x_{m,n} \end{bmatrix} \tag{A.1}$$

次に，各変数の平均値からの残差を求めるため，各列の行ベクトルから，その平均値 $\bar{x}_j$ を間引く操作を行う．

$$x_{i,j} = x_{i,j} - \bar{x}_j, \quad i \in \{1, 2, ..., m\} \tag{A.2}$$

ここで，

$$\bar{x}_j = \frac{1}{m} \sum_i x_{i,j}, \quad i \in \{1, 2, ..., m\}$$

この $\mathbf{X}$ の左側から $\mathbf{X}$ を転置したものを掛け合わせたものが，共分散行列となる．

$$\mathbf{C} = \mathbf{X}^\top \mathbf{X} \tag{A.3}$$

A.2.2 共分散行列の固有値分解による主成分軸ベクトル（負荷行列）の取得

サンプルの傾きの主軸は，共分散行列の固有値問題を解くことで得られる．すなわち，共分散行列 $\mathbf{C}$ の固有値に対応する固有ベクトル

$$\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\} = \text{eigenvector}(\mathbf{C}) \tag{A.4}$$

として主軸ベクトルが求まり，固有値の大きいほうから順に，第 1 主成分軸 $\mathbf{v}_1$，第 2 主成分軸 $\mathbf{v}_2$ といったように，サンプルの分散を大きく表す主要な軸となる．また，すべての軸は互いに直交する．図 A.1 (b) においてはサンプルを囲うように長方形が示してあるが，対角右上方向が第 1 主成分軸となり，これに直交する軸が第 2 主成分軸となる．

また，これらの主成分軸ベクトルを横１列に並べてできる行列

$$\mathbf{L} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n] \tag{A.5}$$

は負荷行列（Loading Matrix）と呼ばれ，この行列が座標変換やデータの再構築を行うための重要な変換行列となる．

A.2.3 負荷行列を変換行列としたデータの再構成

主成分軸などの軸ベクトルは軸方向を表すだけでなく，その方向への成分の大きさを求めるための変換行列（ベクトル）として使用することができる．例えば，座標系 x, y, z の 3 次元空間において，軸ベクトルが $\mathbf{v} = [1, 0, 0]^\top$，サンプルが $\mathbf{p} = [x, y, z]$ がとして与えられた場合，$x = \mathbf{pv}$ として軸方向への成分の大きさが得られる．

以上は１つの軸ベクトルのみを用いた場合であるが，複数個を用いて行うこともでき，例えば３つの互いに直交する軸ベクトル $\mathbf{v}_i$，$\mathbf{v}_j$，$\mathbf{v}_k$ を横に並べた行列 $\mathbf{V} = [\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k]$ は，成分の各軸方向への大きさを抜き出すための変換行列となる．

$$\mathbf{p}_{i,j,k} = \mathbf{pV} = \mathbf{p}[\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k] \tag{A.6}$$

このことを利用し，負荷行列 $\mathbf{L}$ の第１列から第 i 列までの主成分軸ベクトルをまとめ，変換行列として使用すれば，サンプルを主要な i 個の変数によって代表させることができる．

1.2.3.1 累積寄与率による必要な主成分数の決定

もとの変数の数よりも少ない主成分で，データを表す場合に問題となるのが，どの程度の数の主成分を用いれば，もとのデータに含まれる特徴を十分に表すことができるかということである．通常，主成分分析ではこれを判断するための指標として寄与率および累積寄与率を用いる．

寄与率は第 k 主成分が元のデータに含まれる特徴をどの程度表現するかを表すものであり，第 k 主成分の分散が，主成分の分散の総和に占める割合で与えられる．主成分分析においては第 k 主成分の分散は k 番目に大きな固有値と等しいので，寄与率は

$$p_k = \frac{\lambda_k}{\sum_{j=1}^{n} \lambda_j} \tag{A.7}$$

として与えられる．累積寄与率は第１主成分からの寄与率を累積したものであり，

$$p_{k\ \mathrm{accum}} = \sum_{j=1}^{k} p_j = \sum_{j=1}^{k} \lambda_j / \sum_{j=1}^{n} \lambda_j \tag{A.8}$$

として与えられる．

上述の累積寄与率を用いて主成分数を決定する場合，累積寄与率が 80%程度となるように主成分数が決められる場合が多い．

A.2.4 サンプルの向きを主成分軸に合せるための回転行列の作成

負荷行列を変換行列として座標変換を行った場合，サンプルの座標系は主成分軸を新しい軸とした座標系へと移ることになる．式 (A.6) においては，変換後のサンプル $\mathbf{p}_{i,j,k}$ の座標系は $\mathbf{v}_i$，$\mathbf{v}_j$，$\mathbf{v}_k$ を軸とする座標系となる．

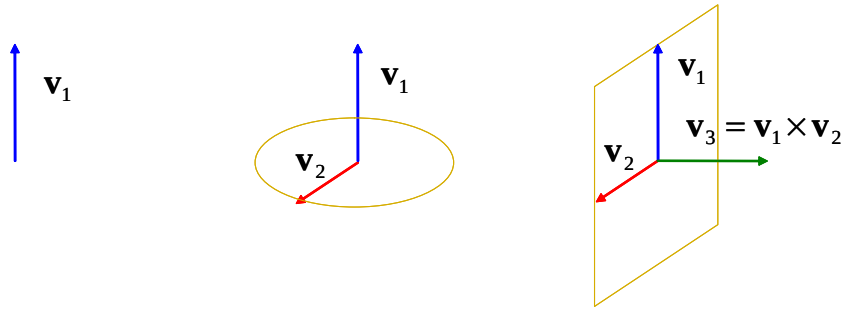
このことを利用し，負荷行列を回転行列のようなものと見なして 3D モデルの頂点の座標変換を行えば，本節最初に示した図 A.1 のように 3D モデルの向きを主軸に合せることができる．

しかし実際には，この変換では軸上の正と負の符号が入れ替わる場合があるため，3D モデルの回転に負荷行列をそのまま使用した場合，モデルによっては座標の符号が反転し，メッシュが裏返えるということが起こる．

これを防ぐため，第 3 主成分軸を表すベクトルを，第 1 主成分軸と第 2 主成分軸のベクトルの外積によって求めなおす必要がある．すなわち，

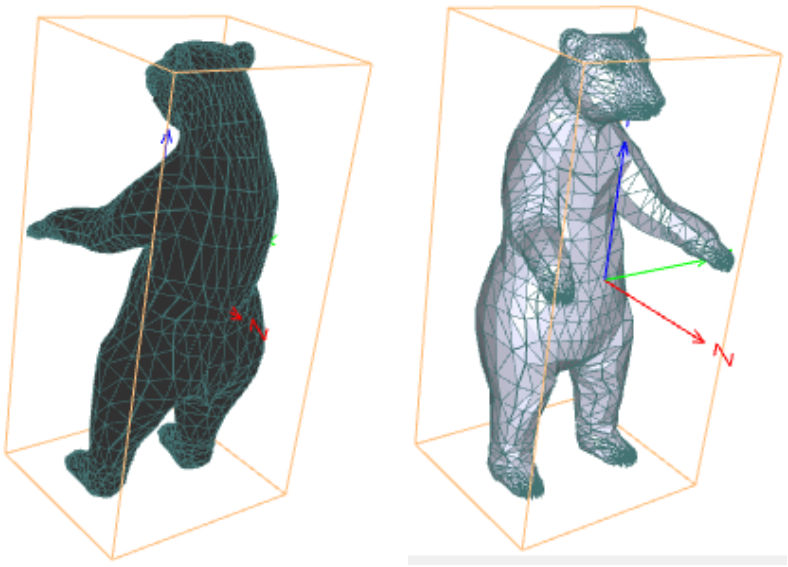
$$\mathbf{R} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_2 \times \mathbf{v}_1] \tag{A.9}$$

として，第 1 の軸と第 2 の軸によって張られた面に対し，第 3 の軸が必ず決まった方向を向くように設定しなおす必要がある．図 A.2 はこの様子を示したものである．また，図 A.3 が回転行列にすべての固有ベクトルをそのまま用いた場合と，第 3 主成分軸に外積を用いた場合の 3D モデルの位置合わせの結果を示したものである．



(a) (b) (c)

図 A.2: まず第 1 主成分軸が任意の方向にとられ (a)，次に，この第 1 主成分軸に垂直ないずれかの方向に第 2 主成分軸がとられる (b)．ここまでは特に座標の符号は特に関係しない．ただし，第 3 主成分軸をとる際には，すでに二つのベクトルにより面が張られているため，第 3 主成分軸の方向によっては面が裏返ってしまう．そのため，面を張る二つのベクトルの外積（法線）ベクトルを新たな第 3 主成分軸とし，方向を固定する．



(a) (b)

図 A.3: 負荷行列を用いて座標変換を行った場合に生じる 3D メッシュの反転．(a) は得られた負荷行列 $\mathbf{R} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$ をそのまま座標変換に用いた場合を示したもので，(b) は第 3 主成分軸を，第 1 主成分軸と第 2 主成分軸の外積として求めなおし，$\mathbf{R} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_2 \times \mathbf{v}_1]$ としたものを使用した場合を示したものである．

A.3 重心座標による面の補間

メッシュを画像として表示する場合，各画素で色や陰影などを計算する必要があるが，メッシュは頂点上にしかデータを持たないため，画素に対応しているメッシュの面上の任意の点で，3 次元頂点座標や法線ベクトルといったデータを求めるためには，頂点座標からデータを補間する必要がある．このような補間方法は，ポリゴンの走査変換という名で知られており [CG 標準テキストブック]．ここでは，その中の一つである三角形の重心座標を用いた補間方法（Barycentric Interpolation）について説明する[*1].

A.3.1 重心座標表現

三角形内部に点を取り，この点と三角形の各頂点を結べば，三角形を 3 つの小さな三角形に分けることができる．この 3 つの三角形の面積比をとり，各頂点の座標に掛け合わせ，これらを足し合わせると，三角形内部の座標を表すことができる．これを重心座標（Barycentric Coordinates）表現と呼ぶ.

図 A.4 は面積の取り方を示したものであり，三角形の各頂点の座標を $\mathbf{u}_1$，$\mathbf{u}_2$，$\mathbf{u}_3$ とし，この頂点の対角の三角形の面積比を，それぞれ S_1，S_2，S_3 と置けば，

$$\mathbf{u} = S_1\mathbf{u}_1 + S_2\mathbf{u}_2 + S_3\mathbf{u}_3, \quad S_1 + S_2 + S_3 = 1 \tag{A.10}$$

として任意の座標 $\mathbf{u}$ を表すことができる．面積比 S_i はベクトルの外積を用いて，

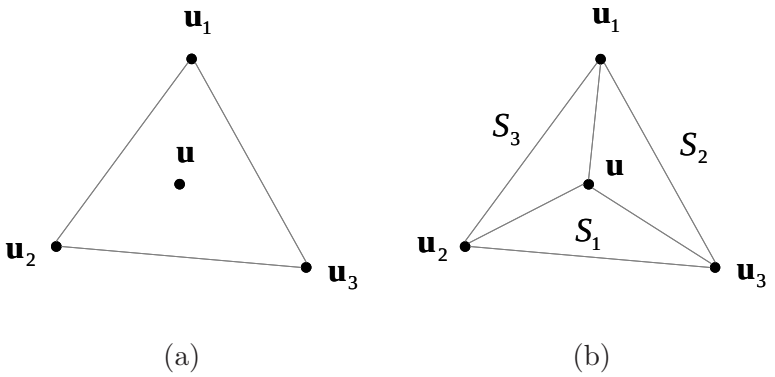


図 A.4: 重心座標による頂点座標の補間．三角形内部の頂点座標は，この頂点によって分割される三角形の面積比と，各頂点座標の線形結合によって $\mathbf{u} = S_1\mathbf{u}_1 + S_2\mathbf{u}_2 + S_3\mathbf{u}_3$ と表される．

[*1] 多角形ポリゴンの場合でも，多角形を三角形に分けることで同様の処理を行うことができる

$$
\begin{aligned}
S_1 &= \frac{1}{2} \|(\mathbf{u}_2 - \mathbf{u}) \times (\mathbf{u}_3 - \mathbf{u})\| \\
S_2 &= \frac{1}{2} \|(\mathbf{u}_3 - \mathbf{u}) \times (\mathbf{u}_1 - \mathbf{u})\| \\
S_3 &= \frac{1}{2} \|(\mathbf{u}_1 - \mathbf{u}) \times (\mathbf{u}_2 - \mathbf{u})\|
\end{aligned}
\tag{A.11}
$$

として求めた三角形の面積を，次のように正規化することで得ることができる．

$$
S_i = S_i \ / \ \sum_i S_i, \quad i \in \{1, 2, 3\} \tag{A.12}
$$

A.3.2 データの補間

次に，投影された 2 次元頂点の持つもとの 3 次元頂点座標の補間を行う．これは簡単に行え，式 (A.10) の 2 次元頂点座標 $\mathbf{u}$ を単に 3 次元頂点座標 $\mathbf{p}$ に置き換えるだけで済み，

$$
\mathbf{p} = S_1\mathbf{p}_1 + S_2\mathbf{p}_2 + S_3\mathbf{p}_3 \tag{A.13}
$$

面積比を計算しなおす必要はない．これは，頂点座標 $\mathbf{u}$ が $\mathbf{p}$ をある平面に投影して得られたものである場合，すなわち，$\mathbf{u} = \mathbf{P}\mathbf{p}$ として投影行列 $\mathbf{P}$ などによって線形的に変換されたものである場合，面積比は一定に保たれるためである．

同様にして，本来 3 次元の面に属するデータを 2 次元で求めた面積比を用いて線形補間することができ，法線ベクトルなども

$$
\mathbf{n} = S_1\mathbf{n}_1 + S_2\mathbf{n}_2 + S_3\mathbf{n}_3 \tag{A.14}
$$

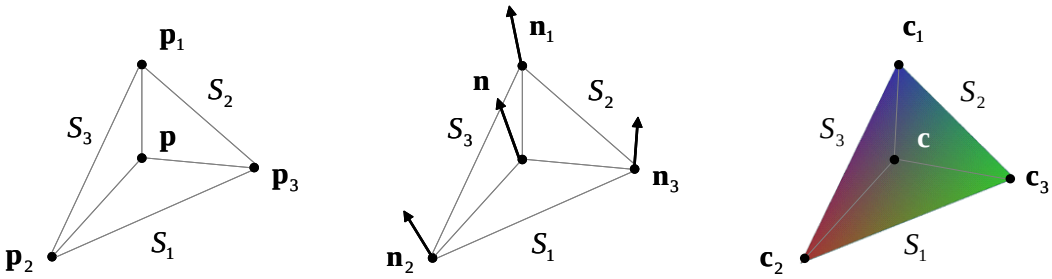
として補間することができる．また，頂点の持つ色なども同様にして補間することができ，

$$
\mathbf{c} = S_1\mathbf{c}_1 + S_2\mathbf{c}_2 + S_3\mathbf{c}_3, \quad \mathbf{c} = [R, G, B] \tag{A.15}
$$

として求められる．図 A.5 はこの補間の様子を示したものである．

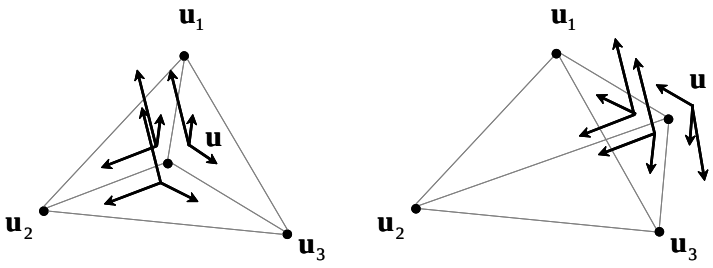
A.3.3 三角形の内外判定

なお，点 $\mathbf{u}$ が三角形内部に含まれるかどうかをこの方法では判定する必要があるが，式 (A.11) で用いた外積ベクトルの向きによって判断することができる．すなわち，点 $\mathbf{u}$ が三角形の内部にある場合には外積ベクトルがすべて同じ方向を向き，一方，点 $\mathbf{u}$ が三角形の外部ある場合にはいずれかのベクトルが反対方向を向く．図 A.6 はこの判定における外積ベクトルの向きを表したものである．



(a) (b) (c)

図 A.5: 重心座標による頂点のもつデータの補間．2 次元で求めた面積比との線形結合により，3D 頂点座標 **p**，法線ベクトル **n**，色 **c** などの補間も行うことができる．



(a) (b)

図 A.6: 三角形ポリゴンでの内外判定．点が三角形内部にあれば，外積ベクトルはすべて同じ方向を向く (a)．一方，点が外部にある場合は，いずれかのベクトルが反対方向を向く (b)．この向きにより点が三角形に含まれるかどうかを判断できる．

参考文献

[Baraff and Witkin 1998] Baraff, D., and Witkin, A. 1998. Large steps in coth simulation. *In Proceedings of ACM SIGGRAPH*, 43–53.

[CG 標準テキストブック] CG 標準テキストブック. Computer graphics 技術編 CG 標準テキストブック.

[Chen et al. 2003] Chen, D. Y., Tian, X. P., Shen, Y. T., and Ouhyoung, M. 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum (EUROGRAPHICS'03) 22*, 3, 223–232.

[Cignoni et al. 1998] Cignoni, P., Rocchini, C., and Scopigno, R. 1998. Metro: measuring error on simplified surfaces. *Computer Graphics Forum, Blackwell Publishers 17*, 2, 167–174.

[Desbrun et al. 1999] Desbrun, M., Meyer, M., Schröder, P., and Barr, A. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. *In Proceedings of ACM SIGGRAPH*.

[Desbrun et al. 2002] Desbrun, M., Meyer, M., and Alliez, P. 2002. Intrinsic parameterizations of surface meshes. *Computer Graphics Forum (Proc. Eurographics 2002)*, 209–218.

[document] document, O. Opengl -the opengl graphics system utility library-.

[Eck et al. 1995] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzl, W. 1995. Multiresolution analysis of arbitrary meshes. *In Proceedings of ACM SIGGRAPH*, 173–182.

[Elad et al. 2000] Elad, M., Tal, A., and Ar, S. 2000. Directed search in a 3D objects database using svm. Tech. Rep. HPL-2000-20(R.1), HP Laboratories Israel.

[Fan et al. 2005] Fan, X., Xie, X., Li, Z., Li, M., and Ma, W.-Y. 2005. Photo-to-search: Using multimodal queries to search the web from mobile devices. *the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR 2005)*.

[Floater 1997] Floater, M. S. 1997. Parameterization and smooth approximation of surface triangulations. *Computer Aided Geometric Design 14*, 3, 231–250.

[Garimella and Swartz 2003] Garimella, R. V., and Swartz, B. K. 2003. Curvature estimation for unstructured triangulations of surfaces. Tech. rep., Los Alamos National Laboratory.

[Glassner 1984] Glassner, A. S. 1984. Space subdivision for fast ray tracing. *IEEE Computer Graphics and Applications 4*, 10, 15–22.

[Gu et al. 2002] Gu, X., Gortler, S., and Hoppe, H. 2002. Geometry images. *In Proceedings of ACM SIGGRAPH*, 355–361.

[Hoppe and Praun 2005] Hoppe, H., and Praun, E. 2005. Shape compression using spherical geometry images. *in Advances in Multiresolution for Geometric Modelling, N. Dodgson, M. Floater, M. Sabin (eds.), Springer-Verlag 2005*, 27–46.

[Johnson and Hebert 1999] Johnson, A. E., and Hebert, M. 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Johnson 1997] Johnson, A. E. 1997. *Spin-Images: A Representation for 3-D surface matching*. PhD thesis, doctoral dissertation, Robotics Institute, Carnegie Mellon University.

[Ke and Sukthankar 2004] Ke, Y., and Sukthankar, R. 2004. Pca-sift: A more distinctive representation for local image descriptors. *In Proc. of*

the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR).

[Khan et al. 2004] Khan, I., Okuda, M., and Takahashi, S. 2004.

[Lee et al. 1995] Lee, Y., Terzopoulos, D., and Walters, K. 1995. Realistic modeling for facial animation. *in Proc. of the 22nd Annual Conf. on Computer Graphics and Interactive Techniques*, 55–62.

[Möller and Trumbore 1997] Möller, T., and Trumbore, B. 1997. Fast, minimum storage ray-triangle intersection. *Journal of graphics tools 2*, 1, 21–28.

[Novotni and Klein 2001] Novotni, M., and Klein, R. 2001. A geometric approach to 3D object comparison. *In Proceedings of the International Conference on Shape Modeling and Applications*, 167–175.

[Okuda et al. 2003] Okuda, M., Nagatomo, K., Ikehara, M., and Takahashi, S. 2003. Compression of 3D models by remesh on texture images. *IEICE Trans. on Information and Systems E86-D*, 6, 1110–1115.

[Okuda et al. 2004] Okuda, M., Nagatomo, K., Ikehara, M., and Takahashi, S. 2004. Similarity detection of 3D meshes using 2d herarchical regular grids. *IEEE Inter. Conf. on Multimedia and Expo TP1-6-5*.

[Osada et al. 2001] Osada, R., Funkhouser, T., Chazelle, B., and Dobkkin, D. 2001. Matching 3D models with shape distributions. *in Proc. Intr. Conf. on Shape Modeling and Applications*, 154–166.

[Pito 1996] Pito, R. 1996. A sensor-based solution to the“ next best view ” problem. *Pattern Recognition, 1996, Proc. of 13th Inter. Conf 1*, 941–945.

[Quail] Quail, M. Space time ray tracing using ray classification.

[Sander et al. 2001] Sander, P., Snyder, J., Gortler, S. J., and Hoppe, H. 2001. Texture mapping progressive meshes. *In Proceedings of ACM SIGGRAPH*, 409–416.

[Sander et al. 2002] Sander, P., Gortler, S., Snyder, J., and Hoppe, H. 2002. Signal-specialized parametrization. *Eurographics Workshop on Rendering*, 87–100.

[Sander et al. 2003] Sander, P., Wood, Z., Gortler, S., Snyder, J., and Hoppe, H. 2003. Multi-chart geometry images. *Eurographics Symposium on Geometry Processing*, 146–155.

[Shirai et al. 2004] Shirai, K., Okuda, M., and Ikehara, M. 2004. Fast regular mesh approximation of 3D models by cylindrical mapping. *IEEE International Symposium on Communications and Information Technologies 2004*.

[Shirai et al. 2006a] Shirai, K., Nagatomo, K., Okuda, M., Ikehara, M., and Takahashi, S. 2006. Cylindrical approximation of 3D meshes and its application to similarity detection. *Journal of Signal Processing*.

[Shirai et al. 2006b] Shirai, K., Okuda, M., and Ikehara, M. 2006. Area-distribution uniformizing parameterization for preserving features of 3D meshes. *12th Digital Signal Processing Workshop 4th Signal Processing Education Workshop*.

[Tomasi and Manduchi 1998] Tomasi, C., and Manduchi, R. 1998. Bilateral filtering for gray and color images. *in Proc. of the 1998 IEEE Inter. Conf. on Computer Vision*.

[Tsai et al. 2002] Tsai, C., Chang, W., Chen, C., and Tang, G. Y. 2002. Compression of 3D objects with multistage color-depth panoramic maps. *in Proc. of IEEE Data Compression Conf.*.

[Vranic and Saupe 2002] Vranic, D. V., and Saupe, D. 2002. Description of 3D-shape using a complex function on the sphere. *In Proceedings of the IEEE International Conference on Multimedia and Expo*, 177–180.

[Winkelbach et al. 2003] Winkelbach, S., Westphal, R., and Goesling, T. 2003. Pose estimation of cylindrical fragments for semi-automatic bone

fracture reduction. *Pattern Recognition (DAGM 2003), Lecture Notes in Computer Science*, 566–573.

[Yan et al. 2005] Yan, Z., Kumar, S., and Kuo, C.-C. J. 2005. Mesh segmentation schemes for error resilient coding of 3-D graphic models. *IEEE Trans. on Circuits and Systems for Video Technology 15*, 1, 138–144.

[Yoshizawa et al. 2004] Yoshizawa, S., Belyaev, A. G., and Seidel, H. P. 2004. A fast and simple stretch-minimizing mesh parameterization. *in Proc. Shape Modeling and Applications*, 200–208.

[Zhang et al. 2003] Zhang, E., Mischaikow, K., and Turk, G. 2003. Feature based surface parameterization and texture mapping. Tech. rep., Georgia Institute of Technology.

[梶谷 将治 et al. 2005] 梶谷 将治, 中川 直子, and 奥田 正浩. 2005. 複数の距離画像を用いた 3 次元映像の実時間伝送に関する検討. *Visual Computing/* **グラフィクスと** *CAD* **合同シンポジウム**.

[白井 啓一郎 et al. 2006a] 白井 啓一郎, 奥田 正浩, and 池原 雅章. 2006. 3D モデルの高速な円筒ジオメトリ画像化法. **電子情報通信学会論文誌** *A J89-A*, 7, 629–638.

[白井 啓一郎 et al. 2006b] 白井 啓一郎, 奥田 正浩, and 池原 雅章. 2006. 3D モデル形状復元のための頂点周辺密度均一化によるパラメータ化. **電子情報通信学会論文誌** *A*.

謝辞

本研究は著者が慶應義塾大学 大学院 理工学研究科 後期博士課程在学中に行ったものである．本研究を遂行し，本論文をまとめるにあたり，御意見，御指導を賜った，慶應義塾大学理工学部 **池原 雅章** 教授に心から感謝申し上げます．また，本研究の内容の詳細にわたり御助言を賜りました，北九州市立大学国際環境工学部 **奥田 正浩** 助教授に心から感謝申し上げます．本研究は信号処理とコンピュータグラフィックスを基盤とするものであり，このように多種の分野にわたる研究を遂行できたのも，各々の分野に精通する池原教授と奥田助教授の強い支えがあったこと，また，両氏に非常に良い機会にめぐり合えたことが大きいと思っております．

また，慶應義塾大学理工学部 **浜田 望** 教授，**岡田 英史** 教授，**斎藤 英雄** 教授には，本論文の審査をしていただくと共に，本論文の内容に関して多くの貴重な意見を賜りました，深く感謝申し上げます．

最後に，研究室配属時から様々な面で支えて頂いた池原研究室の皆様に感謝致します．