

# 格子点連結法と局所操作を用いた点群データからのソリッドモデル自動再構成

松岡 司<sup>†</sup> 植田 健治<sup>†</sup> 早野 勝之<sup>†</sup>

3次元表現の利用が拡大するにつれて、現実の3次元形状から幾何的に正確かつ位相構造を再現したソリッドモデルを平易な手段により生成することの重要性が高まっている。3次元形状から面モデルを生成する前の中間データとして、3次元測定器などで測定した点群データを一般に用いる。しかし、測定器の性質により、多くの場合点群密度のばらつきが発生するため、いかなる点群データからも幾何的再現性と位相的再現性を両立したモデルを生成することは非常に困難であった。そこで我々は、これら幾何、位相両面にわたる再現性の問題を2ステップのアルゴリズムにより解決した、構造化されていない点群からのポリゴンメッシュの再構成手法を提案する。

## Solid Model Reconstruction from Unorganized Points with a Marching Lattice Points Method and Local Operations

TSUKASA MATSUOKA,<sup>†</sup> KENJI UEDA<sup>†</sup> and MASAYUKI HAYANO<sup>†</sup>

As using 3D models in computer environments is getting popular, the needs of an easy method which can construct geometrically exact shape and topological structure of solid models from actual 3D objects is increased. To construct 3D models from actual 3D objects, we generally use a set of points data taken from various 3D scanner devices. However, it is very difficult to construct exact shape of solid models from these points data because of the nonuniform density property of scanner devices. Therefore, we propose a new polygon mesh reconstruction method from unorganized points set which can solve geometric and topologic fidelity problems with our two-step algorithm.

### 1. はじめに

近年、インターネット上でのマルチメディア利用の広がりとともに3次元表現の利用は拡大しており、平易な手段による3次元モデリングの重要性が高まっている。従来、3次元モデルを得るにはCADなどの形状入力ツールを用いて手動でモデリングを行っていた。そのため、同じ形状入力ツールを用いてもモデリングを行う人の技量、経験により生成されるモデルは異なっていた。そこで現在では、モデリングを行う人の技量や経験に依存しない、3次元測定器などを用いて3次元形状の表面に分布する点群データを取得した後、それを用いて自動的にモデルを生成する手法が注目されている。しかし、点群データを取得する手法により点群密度のばらつきが発生することが多いため、いかなる点群データからも幾何的正確さと位相的再現性を両立したモデルを生成することは非常に困難であった。

そこで、本論文では位相と幾何の再現性を両立した次のような2ステップのアルゴリズムを提案する。

**step1** 格子点連結法による、貫通穴や中空などの位相構造を再現した基本メッシュ (basic mesh) の生成

**step2** 局所操作を用いた詳細化による、点群の点をすべて頂点とする細密メッシュ (dense mesh) の生成

本手法は、位相再現処理、幾何再現処理を別ステップにより分離して処理することにより、位相と幾何の再現性を両立した点群の性質に依存しない汎用的手法であり応用範囲は広い。従来手法は、点群の性質に前提があるか、もしくは位相・幾何いずれか1つの再現性に重きを置いた処理であるため、汎用性は低く応用範囲も限定される。

本章の残りでは、点群からのモデリング手法の基礎となる点群の性質および点群からのモデリング手法に求められる要件について述べる。以降、2章では従来手法の問題点について述べる。3章で step1 の基本メッシュ生成について述べ、4章で step2 の細密メッ

<sup>†</sup> 株式会社リコー研究開発本部ソフトウェア研究所  
Software Research Center, Ricoh Company, Ltd.

シユ生成について述べる。5章で本手法を適用した例と考察を示す。

### 1.1 点群の性質

点群データは装置に依存したさまざまな性質を持つが、共通する性質として密度の非一様性がある。また、方式によっては密度の低いものもある。レーザーレンジスキャナを用いて取得した点群データは密度が高く一様 (dense and uniform) である。しかし、複数の方向からとった点群データをマージして全体を表す点群データとする際に重なりや欠落が生じる場合があり、点群全体としては一様な密度ではない。また、ビデオカメラなど普及している装置を用いることにより、点群を得る方式<sup>1)~3)</sup>の研究も行われている。shape from silhouette 手法<sup>1)</sup>は形状を回転させたときのシルエットを観測することにより、点群を取得する。形状の部位により回転半径が異なるので、得られる点群の密度も異なる。shape from motion<sup>2),3)</sup>手法はビデオなど単眼カメラにより取り込んだ連続画像をもとに点群を取得する。この手法は特徴点を抽出し追尾することによりその座標を得るので、密度は低く一様ではない。

### 1.2 点群からのモデリング手法の要件

点群を基にモデリングを行う手法に必要な条件を示す。

#### (1) 幾何再現性

点群から生成したポリゴンメッシュに対して、雑音除去、特徴抽出、最適化、単純化などアプリケーションで扱うためにさまざまな後処理が行われるが、精度の悪いポリゴンメッシュはいずれの処理においても不利なので、精度の良いポリゴンメッシュが必要である。

#### (2) 位相再現性

位相構造を失ったメッシュは視覚的にはさほど問題がないが、工学的には構造解析できないという問題がある。面積計算、体積計算、強度計算など構造を解析するアプリケーションには位相的再現性が必要である。3次元表現利用の拡大につれて視覚利用以外の利用方法も増加しており、この問題は拡大する傾向にある。

#### (3) 点群密度の非一様性

点群データは装置に依存したさまざまな性質を持つが、総じて、密度の非一様性がある。したがって、密度が高く一様な点群 (dense and uniform points) だけではなく、密度がまばらで一様でない点群からも形状復元できることが望ましい。

#### (4) 点群データ形式の一般性

点群を基にモデリングを行う手法が普及するにつれて、今後は点群データの交換がさかんになり、さまざまな形式や付加情報を持った点群が流通すると思われるので、点群の形式や付加情報などに依存しない手法が利便性が高い。したがって、構造化されていない点群 (unorganized points) から形状復元できることが望ましい。

## 2. 従来手法

点群からポリゴンメッシュを生成する手法としては、過去にさまざまな手法が提案されているが、点どうしの隣接関係が明確な構造化された点群を扱う手法<sup>6),9)</sup>と構造が未知の点群を扱う手法<sup>4),5),7),8)</sup>の2つに分類できる。主なポリゴンメッシュ生成アルゴリズムの性質を表1にまとめた。

隣接関係が明確な整列された点群データからポリゴンメッシュを自動生成する手法は、Turkらの方法<sup>6)</sup>に代表される。彼らの手法では点群の点をすべて頂点とする精度の良いポリゴンメッシュを生成できるが、物体表面を覆いつくすような複数方向からの整列された点群データを前提としているので点群データ形式の一般性に欠ける。

整列されていない点群データからポリゴンメッシュを自動生成する手法としては、Hoppeらの方法<sup>7)</sup>がある。彼らの手法では密度が高く一定している (dense and uniform) 測定器で取り込んだ点群を想定している。Marching Cubes法を用いる彼らの手法ではまばらで密度が一定していない点群では最も密度が低いところにあわせて格子間隔を広げることを余儀なくされ、精度の良いメッシュを作成できない。芹田らの方法<sup>8)</sup>では Marching Cubes法<sup>10)</sup>によりポリゴンメッシュを生成した後、メッシュの頂点を最も距離の近い点に移動させることで精度を向上させている。しかし、

表1 点群からのポリゴンメッシュ生成アルゴリズム  
Table 1 Polygon mesh reconstruction algorithms from points set.

手法	点群属性	位相制限	誤差
ZP法	隣接関係、一様	なし	なし
MC法1	細密、一様	中空	あり
MC法2	細密、一様	なし	あり
$\gamma$ -graph <sup>4)</sup>	不要	貫通穴、中空	なし
$\alpha$ -solid <sup>5)</sup>	細密、一様	中空	なし
本手法	不要	なし	なし

ZP法: Zippered Polygon<sup>6)</sup>

MC法1: Marching Cubes with Signed Distance Function<sup>7)</sup>

MC法2: Marching Cubes with Volumetric Method<sup>8)</sup>

まばらで密度が一定していない点群では多くの点が頂点にはならないので、精度の良いメッシュを作成できない。Veltkampの方法<sup>4)</sup>では与えられた点群の密度に関係なく点をすべてポリゴンメッシュの頂点にすることができる。すなわち、密度が非一様な点群からでも精度の良いメッシュを作成できる。しかし、再現できる位相構造に制限があり、貫通穴を再現できない。Bernardiniらの方法<sup>5)</sup>では与えられた点群の点をすべてポリゴンメッシュの頂点にすることができ、位相構造も再現できる。しかし、密度が高く一様な点群という前提がある。また、中空構造の再現性は明示されていない。

### 3. 基本メッシュ生成

基本メッシュ生成のアルゴリズムの概要を示す。

- (1) 最小点群密度計算  
点群の最も疎な部分の密度である最小の点群密度を計算する。
- (2) ボクセルサイズ計算  
輪郭ボクセルにより物体の表面を覆いつくせるように、最小点群密度を基にボクセルサイズを計算する。
- (3) ボリュームモデル生成  
輪郭ボクセルに挟まれるボクセルを内部ボリュームとするボリュームモデルを生成する。
- (4) 基本メッシュ生成  
輪郭ボクセル内の代表点を頂点とするポリゴンメッシュを生成し、基本メッシュとする。

以上により、貫通穴や中空などの位相構造を再現した基本メッシュを生成できる。

#### 3.1 点群最小密度

点群は物体の表面に沿って分布していると考えられるので、物体表面の単位面積あたりの点の個数を表面密度  $\rho$  と定義する。 $\rho$  の計算方法を示す。

- (1) 点群  $P = \{p_1, p_2, \dots, p_n\}$  内のある点  $p_i$  について  $k$ -neighborhoods

$$N_{bhd}(p_i) = \{n_{bhd}(1), n_{bhd}(2), \dots, n_{bhd}(k)\}$$

を求める。 $n_{bhd}(j)$  とは  $N_{bhd}(p_i)$  の中で点  $p_i$  から  $j$  番目に近い点を示す。

$k$ -neighborhoods はある点からの距離が近い順に  $k$  個の点を取得したものである。 $k$  は点群の性質により決定する。密度が一定の点群では比較的小さな数、密度がまばらで一定していない点群では大きな数を設定する (図 1)。

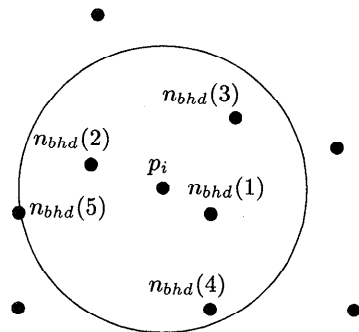


図 1  $k$ -neighborhoods ( $k=5$ ) と包含球  
Fig. 1  $k$ -neighborhoods ( $k=5$ ) and including sphere.

- (2)  $k$ -neighborhoods の中で元の点  $p_i$  から最も遠い点  $n_{bhd}(k)$  と元の点  $p_i$  の間の距離を半径とする円の面積  $A_{disc}$  を計算する。

$$A_{disc} = 2\pi \cdot \|p_i - n_{bhd}(k)\|^2 \quad (1)$$

点  $n_{bhd}(k)$  と点  $p_i$  の間の距離を半径とし、点  $p_i$  を中心とする球の内部に  $k$ -neighborhoods のすべての点は含まれる。また、点は物体の表面に分布するため、局所的には平面にのるものと考えられる。したがって、点  $p_i$  における表面密度  $\rho$  は

$$\rho = \frac{k+1}{A_{disc}} \quad (2)$$

となる。

点群の局所密度  $\rho$  の計算をすべての点または代表的な点について行い、その中で得られた最小の表面密度  $\rho_{min}$  を点群の最小表面密度と推定する。

#### 3.2 ボクセルサイズ

ボリュームモデルを生成するには点群の点を含む輪郭ボクセルが物体の表面をすべて覆う必要がある。したがって、点群の最小表面密度の箇所にあわせてボクセルの 1 辺の長さ  $L$  を式 (3) のように決定する。

$$L = \frac{C}{\sqrt{\rho_{min}}} \quad (3)$$

$C$  はマージンを表す 1 より大きな定数である。輪郭ボクセルの欠損がある場合ボリュームモデルは生成されない。芹田らの方法<sup>8)</sup>では輪郭ボクセルの欠損をなくするため、さらに欠損箇所の補間を行っている。我々は最小表面密度に基づいてボクセルサイズを決定しており、輪郭ボクセルの欠損は少ない。さらに、係数  $C$  の値を位相構造を失わない範囲内で調整することにより、欠損のない輪郭ボクセルを実現できる。

$C$  の値は、位相構造の再現性を高めるため、できるだけ 1 に近い小さな値にすべきである。これはボクセ

ルサイズが小さいければ、細かな形状をボリュームモデルで再現することができるからである。細い貫通穴や小さな中空などはボクセルサイズが十分に小さくないと、貫通穴や中空として表現されなくなり位相構造が失われることになる。本手法ではボリュームモデルで位相構造が再現されないと、最終的な詳細メッシュでも位相構造は失われたままである。一方、点群密度計算では  $k$ -neighborhoods を用いるので、 $(k+1)$  個の点による平均化が行われるため、 $C$  の値を 1 にしてしまうと、点をいっさい含まない欠落穴の大きさがボクセルサイズを上回り、輪郭ボクセルの欠落が生じる可能性がある。したがって、1 から  $(k+1)$  の間で  $C$  の値を設定するとよい。通常は  $C$  の値は、1 に近い値ということで、1 から 2 の間ぐらいで点群取得装置や方式により経験的に  $C$  の値を決めていくことができると思われる。大きな欠落穴が予想される点群では  $C$  の値を  $(k+1)$  に近づける。ただし、 $C$  の値を大きくしていくと、位相構造が失われる可能性がある。

### 3.3 ボリュームモデル生成

後述の格子点連結法を実施し物体の境界面を生成するには、格子点における内外判定が必要である。そのためにボリュームモデルを生成する。

ボリュームモデルの作成には芹田らの方法<sup>8)</sup>を用いる。その結果、属性 0 のボクセルを外部、属性 1 のボクセルを輪郭、属性 2 を持ったボクセルを内部とするボリュームモデルが生成される。芹田らの方法を示す。

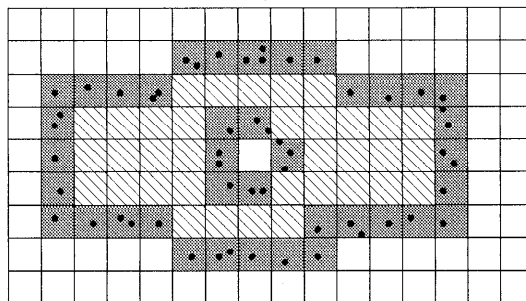
- (1) 点群を包含するだけの数と位置にボクセルを生成する。ただし、端のボクセルが必ず点を含まない外部ボクセルになるように多めに生成する。
- (2) ボクセル内に点群の点を含むなら輪郭ボクセルと見なし、属性 1 を設定する。その他のボクセルは 0 と設定する。境界ボクセルの属性は 0 になる。
- (3) ある方向に境界ボクセルから開始してボクセルを走査し、輪郭ボクセルに挟まれるボクセルに内部ボクセル属性 2 を設定する。これらの走査を 6 方向 (X, Y, Z の各軸の正および負) 行い、4 方向以上で内部ボクセルとなったものを採用する。

図 2 に点群と各ボクセルの関係を示す。

### 3.4 基本メッシュ生成

格子点連結法を実施しボリュームモデルから基本メッシュを生成する。

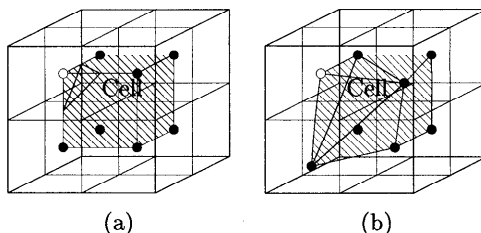
我々は、基本メッシュの頂点を点群内の点にのせるため、Marching Cubes 法<sup>10)</sup>を修正した格子点連結法 (Marching Lattice Points Method) を開発した。



■:輪郭ボクセル □:内部ボクセル □:外部ボクセル

図 2 点群とボクセル

Fig. 2 Points and voxels.



格子点の属性: ○=属性 0, ●=属性 1

図 3 ボクセル, 格子点, Cell: (a) Marching Cubes 法, (b) 格子点連結法

Fig. 3 Voxels, Lattice Points and Cell: (a) Marching Cubes Method, (b) Marching Lattice Points Method.

本節では、最初に格子点連結法の基となった Marching Cubes 法について説明し、次に格子点連結法によるポリゴンメッシュ生成方法について説明する。

#### 3.4.1 Marching Cubes 法

Marching Cubes 法<sup>10)</sup>では隣接する 8 つのボクセルの中心点を格子点とする規則格子を構成し、隣接する 8 つの格子点を頂点とする立方体を Cell とする。図 3 (a) にボクセル, 格子点および Cell を示す。その後、格子点の中間点を結んでポリゴンメッシュを生成する。

点群上の点を頂点に採用したポリゴンメッシュを生成する場合、Marching Cubes 法ではポリゴンメッシュの頂点はつねに格子点の中間点上にあるため、ポリゴンメッシュを生成した後、頂点移動、稜線削除、面削除などのアジャストメント<sup>8)</sup>を行い、頂点の座標を点群の点まで移動させる必要がある。

#### 3.4.2 格子点連結法

ここでは格子点連結法によるポリゴンメッシュ生成について説明する。

- (1) 格子点および値の決定

ボクセル内に含まれる点の中からボクセルの中

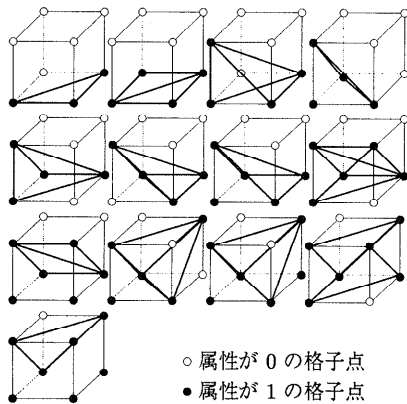


図 4 格子点連結法のポリゴン生成パターン  
 Fig. 4 Polygon generation table of Marching Lattice Points Method.

心に最も近い点を選び、ボクセルの代表点とする。隣接する 8 つのボクセルの代表点を格子点とする不規則格子を構成し、隣接する 8 つの格子点を頂点とする 6 面体を Cell とする。格子点の値はボクセルの属性値 0 の場合はそのまま利用し、ボクセルの属性値 1 または 2 の場合は格子点の値を 1 に設定する。図 3 (b) に格子点連結法のボクセル、格子点および Cell の関係を示す。

(2) ポリゴン生成

格子点の値は属性値 0 が立体の外部、属性値 1 が立体の内部としてポリゴンを生成する。格子点の属性の組合せは 256 通りがあるが、回転により同じ位相になるものを省くと、23 通りになる。このなかでポリゴンを生成するものは、図 4 に示す 13 通りである。これらに従ってポリゴン生成を行う。

格子点連結法では、頂点座標に点群内の点座標を採用したポリゴンメッシュをボリュームモデルからアジャストメントを行うことなく直接生成できる。

4. 細密メッシュ生成

基本メッシュを詳細化し、細密メッシュを生成するアルゴリズムの概要を示す。

細密メッシュ生成アルゴリズム：最初に与えられた点群のうち、基本メッシュから最も遠く離れた点から順に頂点分割操作を適用することによりメッシュの頂点に加えていく。

$$M^0 \xrightarrow{vsplit_1} M^1 \xrightarrow{vsplit_2} \dots \xrightarrow{vsplit_n} M^n$$

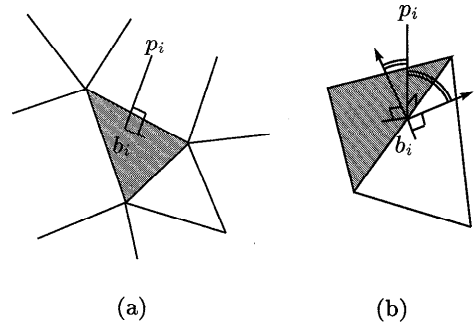


図 5 点の最も近い面への射影：(a) 面の法線ベクトル方向に射影した例、(b) 稜線の垂直方向に射影した例  
 Fig. 5 A point projection to the nearest face: (a) in the direction of the face normal vector, (b) in the perpendicular direction of the edge.

ここで、 $M^0$  は基本メッシュ、 $M^n$  は細密メッシュである。遠くの点から頂点に加えていくことにより、後述するメッシュエネルギーの減少が行われる。

また、個々の頂点操作の際に以下に示すエネルギー評価に基づいた操作を行い、さらにエネルギー減少を図る。

- 頂点分割操作には候補が複数存在するが、候補の中からポリゴンメッシュのエネルギーが最小になるものを選ぶ。
- 頂点分割操作により変化する稜線について、稜線交換操作<sup>11)</sup>を行い、さらにポリゴンメッシュのエネルギーを減少させる。

このように、エネルギーをつねに評価しながら、点群の点を頂点としてポリゴンメッシュに加える頂点分割操作を点がなくなるまで続ければ、点群のすべての点を頂点とする細密メッシュが得られる。また、頂点分割操作と稜線交換操作は位相の変化をとまなわない局所操作であるため、基本メッシュの段階で再現された位相構造を保ったまま詳細化できる。

4.1 点群の面属性登録

詳細化操作を行う前準備として、点群の各点を基本メッシュを構成する面に射影することで面と関係づけ、点の座標と面との間の距離を面の属性として登録する。

点群内の点  $p_i$  を面の法線ベクトル方向に射影し、面上の最も近い点  $b_i$  を求める。そのように点  $p_i$  をすべての面に射影し、点  $p_i$  と点  $b_i$  との距離が最も短い距離になる面に点を登録する。その様子を図 5 (a) に示す。どの面にも射影できない点に関しては、面に射影できない点と最短距離になる稜線上の点または頂点を求め、その点と元の点を結ぶベクトルと、稜線または頂点に隣接する面の法線ベクトルのなす角度が最も

小さな面に属性として登録する。その様子を図 5(b) に示す。

このようにすることで、点群内の点はすべて、基本メッシュを構成するいずれかの面の属性になる。すべての点の中で面との距離が最大である点  $P_f$ 、および  $P_f$  が属する面、すなわち点との距離が最大である面  $F_f$  については、後述する頂点分割操作や稜線交換操作が行われてもつねに最新のものが参照できるように保持・更新できるようにしておく必要がある。しかし、操作によりメッシュの変更が起こるたびにすべての点と面の距離計算を行うのは非常に計算コストが高い。そこで、本論文では計算量の削減のため、メッシュが変形される範囲内でのみ点の面属性登録の更新を行ったところ、5章で述べるように、ほとんどの点を頂点とする詳細なメッシュを生成することができた。

4.2 メッシュエネルギー

ここでは、細密メッシュ生成に用いるメッシュのエネルギーを説明する。メッシュエネルギーは Hoppe ら<sup>11)</sup>により、次のように定義されている。

ポリゴンメッシュ  $M$  のエネルギー  $E(M)$  は、点とポリゴンメッシュの間の距離エネルギー  $E_{dist}$  と稜線の長さエネルギー  $E_{spring}$  の和で表される。

$$E(M) = E_{dist}(M) + E_{spring}(M) \quad (4)$$

距離エネルギー  $E_{dist}$  は、点群の各点から最も近いメッシュの面への距離の 2 乗和で表される。

$$E_{dist}(M) = \sum_i d^2(\mathbf{x}_i, \phi_V(|K|)) \quad (5)$$

ここで  $\mathbf{x}_i$  は点群の点、 $\phi_V(|K|)$  は面の位相構造に相当する単体的複体  $K$  に対応するユークリッド空間の領域、すなわち面そのものを表している。

稜線の長さエネルギー  $E_{spring}$  は、稜線の長さの 2 乗和にばね定数  $\kappa$  を乗じたものとして表される。

$$E_{spring}(M) = \sum_{\{j,k\} \in K} \kappa \|V_j - V_k\|^2 \quad (6)$$

4.3 頂点分割操作

基本メッシュの位相を保ちながら詳細化操作を行うために、我々は頂点分割操作 (vertex split)<sup>12)</sup> を順方向操作に使えるようにポリゴンメッシュ変形機能として拡張した。もともと、頂点分割操作 (vertex split) は、Progressive Mesh<sup>12)</sup> における稜線分割操作 (edge collapse) に対する逆操作であるが、次のような関数仕様に変更し、詳細化の基本操作として用いる。

$$vsplit(\downarrow V_s, \downarrow C_{V_t}, \downarrow C_{V_u}, \downarrow F_{lt}, \downarrow F_{lu}, \downarrow F_{rt}, \downarrow F_{ru}, \uparrow V_t, \uparrow V_u, \uparrow F_l, \uparrow F_r, \uparrow C_{V_s})$$

ここで  $\downarrow$  は入力引数を、 $\uparrow$  は出力引数を表す。  $V_s, V_t,$

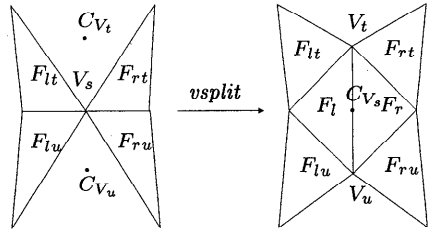
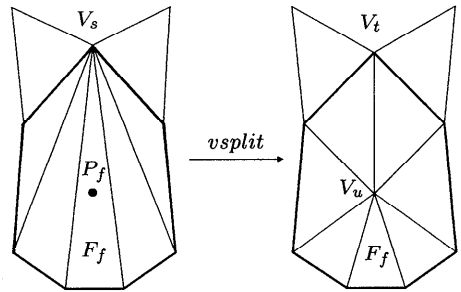


図 6 頂点分割操作  
Fig. 6 Vertex split operation.



$$V_s = V_t = V_{F_f}$$

図 7 頂点分割操作による詳細化  
Fig. 7 Detailization with vertex split operation.

$V_u$  は頂点、 $F_l, F_r, F_{lt}, F_{lu}, F_{rt}, F_{ru}$  は面、 $C_{V_t}, C_{V_u}, C_{V_s}$  は座標をそれぞれ表しており、要素間の関係を図 6 に示す。

この頂点分割操作関数  $vsplit$  に対し、 $V_s$  に点との距離が最大である面  $F_f$  内の頂点  $V_{F_f}$  を、 $C_{V_t}$  に頂点  $V_{F_f}$  の座標を、 $C_{V_u}$  に面との距離が最大である点  $P_f$  の座標を、それぞれ与える。その結果、ポリゴンメッシュから最も遠い点  $P_f$  をポリゴンメッシュの新たな頂点  $V_u$  とする詳細化ができる。図 7 にその様子を示す。

点  $P_f$  が射影される面  $F_f$  の頂点  $V_{F_f}$  は 3 点存在する。また、 $F_{lt}, F_{lu}, F_{rt}, F_{ru}$  の候補も  $V_{F_f}$  のまわりの面の組合せにより複数存在する。このように、点に対する頂点分割操作にいくつか候補があるが、頂点分割操作を行った際に式 (4) に示すポリゴンメッシュのエネルギーが最小になるような候補を選択する。

4.4 頂点分割操作のエネルギー計算

頂点分割操作前後のエネルギーの差分  $\Delta E_{vsplit}$  を、頂点分割操作により変化および追加された面と稜線のエネルギー差分とする。

$$\begin{aligned} \Delta E_{vsplit} = & E_{dist}(\{(V_t|V_u), *, *\}) \\ & + E_{spring}(\{(V_t|V_u), *\}) \\ & - [E_{dist}(\{V_s, *, *\}) \\ & \quad + E_{spring}(\{V_s, *\})] \end{aligned} \quad (7)$$

ここで、 $\{(V_t|V_u), *, *\}$  は頂点  $V_t, V_u$  のいずれか

表 2 本手法の実施結果  
Table 2 Performance statistics of our method.

モデル	$n$	$k$	$\rho_{min}$	$C$	$L$	$m$	$T_m/T_0$ (sec.)
parts (図 9)	5052	5	1.95e-2	1.7	20.02	5051	106.94/5.46
moai (図 10)	3042	5	2.26e-3	1.6	33.95	3032	187.46/36.44
bunny (図 11)	35947	5	2.80e+6	4.0	1.41e-2	35688	1586.88/386.42

$n$ : number of points,  $m$ : number of vertices  
 $T_0$ : process time from points set to basic mesh  
 $T_m$ : process time from points set to detail mesh

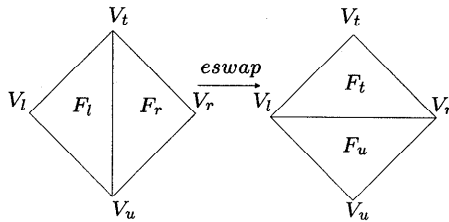


図 8 稜線交換操作  
Fig. 8 Edge swap operation.

を共有する面を,  $\{(V_t|V_u),*\}$  は頂点  $V_t, V_u$  のいずれかを共有する稜線をそれぞれ表す。また,  $\{V_s,*,*\}$  は頂点  $V_s$  を共有する面を,  $\{V_s,*\}$  は頂点  $V_s$  を共有する稜線をそれぞれ表す。

すべての頂点分割操作の候補について  $\Delta E_{vsplit}$  を計算し, それが最小になる頂点分割操作を選択する。

式 (7) の距離エネルギーの項に関しては, メッシュが変形される範囲内だけで点の面属性登録の更新を行っているため正確ではないが, 計算量を削減することができる。頂点分割操作による詳細化を適用した際に変化する面と稜線の範囲を図 7 の太線内に示す。

#### 4.5 稜線交換操作

頂点分割操作ではエネルギー差分を最小にすることによりエネルギー最小化を行っているが, 既存の頂点を残す頂点分割操作では細長い面が生成されやすい。そこで, 頂点分割操作を行った後, 稜線交換操作 (edge swap)<sup>11)</sup> によりエネルギーが減少するものについては, 稜線交換操作を実施する。稜線交換操作関数を示す。

$$eswap(\downarrow V_t, \downarrow V_u, \downarrow F_l, \downarrow F_r, \uparrow V_l, \uparrow V_r, \uparrow F_t, \uparrow F_u)$$

ここで  $\downarrow$  は入力引数を,  $\uparrow$  は出力引数を表す。  $V_t, V_u, V_l, V_r$  は頂点,  $F_l, F_r, F_t, F_u$  は面をそれぞれ表しており, 要素間の関係を図 8 に示す。

頂点分割操作を行ってできる頂点 (図 7 の  $V_u$ ) を共有する面 (図 7 の太線内の面) に属する稜線すべてについて, 稜線交換操作後のメッシュエネルギーが減少する場合, 稜線交換操作を行うことによりポリゴンメッシュのエネルギーの最小化を図る。

#### 4.6 稜線交換操作のエネルギー計算

稜線交換操作前後のメッシュエネルギーの差分  $\Delta E_{eswap}$  を式 (8) に示す。

$$\begin{aligned} \Delta E_{eswap} = & E_{dist}(F_l) + E_{dist}(F_r) \\ & + E_{spring}(\{V_t, V_u\}) \\ & - [E_{dist}(F_t) + E_{dist}(F_u) \\ & + E_{spring}(\{V_l, V_r\})] \end{aligned} \quad (8)$$

$\Delta E_{eswap}$  は, 稜線交換操作により削除される要素 (図 8 の面  $F_l, F_r$ , 稜線  $\{V_t, V_u\}$ ) のエネルギー和から, 稜線交換操作により追加される要素 (図 8 の面  $F_t, F_u$ , 稜線  $\{V_l, V_r\}$ ) のエネルギー和を減じたものになる。

式 (8) の距離エネルギーの項に関しては, メッシュが変形される範囲内だけで点の面属性登録の更新を行っているため正確ではないが, 計算量を削減することができる。

### 5. 結 果

本手法の実施結果を表 2 に示す。すべて AT 互換機 (PentiumII 300 MHz, 256 MB RAM) で実施した。

#### 5.1 CAD データから作成した点群の実施例

3次元ソリッドモデル デSIGNBASE<sup>13)</sup> により作成したソリッドモデルの表面にランダムに点を生成して得られた点群に本手法を適用した例を図 9(a)~(d) に示す。(c) の基本メッシュで位相再現が, (d) の詳細メッシュで幾何再現が行われていることが分かる。

図 9(e)~(h) は頂点分割操作により幾何再現を行っている様子を示す。(e)~(h) にかけて, 点を頂点としてとりこむにつれ精度が向上していくのが分かる。

図 9(j)~(l) は, パラメータを変更した例である。比較のために (i) を示す。(i) はパラメータ設定を表 2 のとおり行い, 736 回頂点分割を行ったところで止めたメッシュである。(j) は稜線交換を行わなかった例である。(i) と比較すると細長い三角形が非常に多く稜線の長さエネルギーが大きいことが分かる。(k) は稜線の長さエネルギーを考慮せずに詳細化を行っていた例である。(i) と比較するとやや細長い三角形が多いことが分かる。(l) は, 大きな欠落穴が予想されない

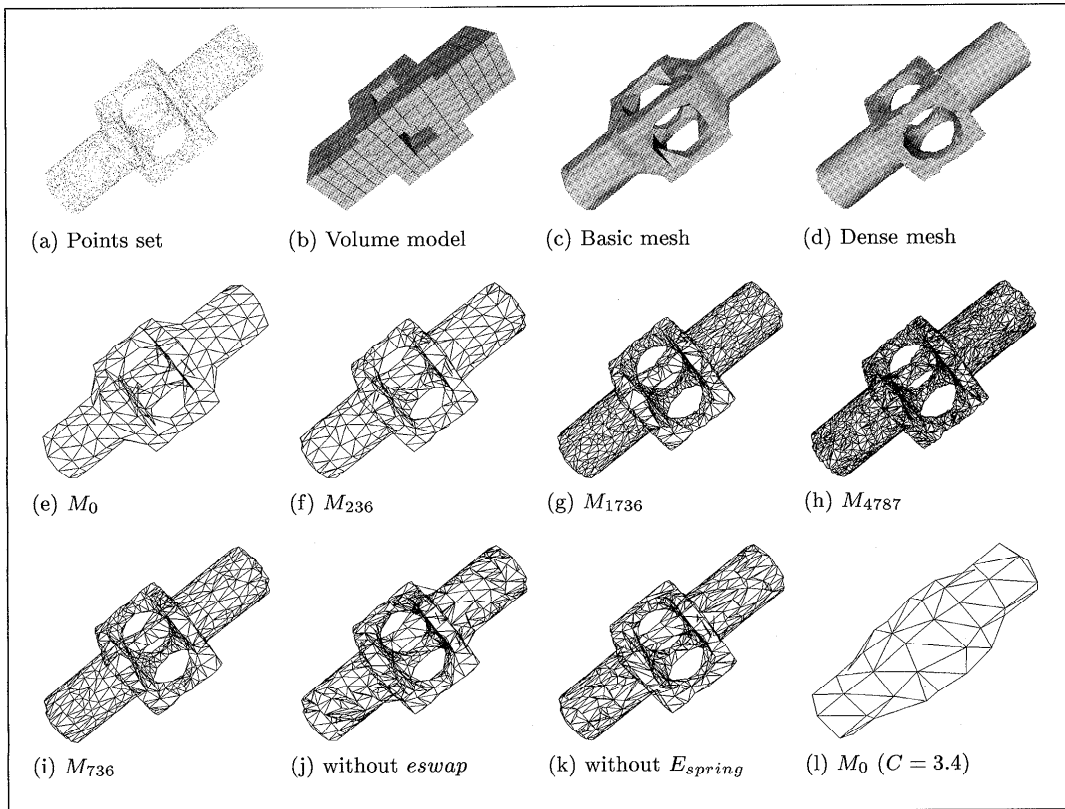


図9 CADデータから自動生成した点群への実施例  
Fig. 9 Reconstruction of CAD object from random sampled point data.

点群であるにもかかわらず、 $C$  を1と  $(k+1)$  の中間的な値である3.7に設定した。基本メッシュの段階で貫通穴は再現されていないので、位相構造が失われているのが分かる。図9の他の例については、 $C = 1.7$  という1に近い値で実施し、すべて貫通穴を再現することができた。

### 5.2 密度や規則性が異なる点群への適用

低密度・非一様点群への適用例を図10に示す。密度の低いまばらな点群の例として、shape from motion手法<sup>3)</sup>により得られた点群(株式会社リコー情報通信研究所提供)に適用した。この点群は密度はまばらだが、大きな欠落穴は予想されなかったため、 $C$  の値は1に近い1.6に設定した。

高密度・一様点群への適用例を図11に示す。密度の高い規則的な点群の例として、レーザーレンジスキャナによりとりこんだ点群<sup>14)</sup>に適用した。この点群は密度は一定だが、底に大きな欠落穴があるので、 $C$  の値を  $(k+1)$  に近い4.0に設定した。

## 6. まとめ

本論文では、位相的再現性と幾何的再現性を両立した、未整列の点群からのポリゴン自動生成手法を提案した。本手法を貫通穴の位相構造を持ったデータに適用した結果、貫通穴が再現されたポリゴンモデルを作成することができた。また、密度や規則性が異なる点群に適用した結果、点群内のほとんどの点を頂点にした精度の良いポリゴンモデルを作成することができ、意図した効果が得られた。

一方、図11のbunnyの耳のつけねなど凹んだ部分の再現性や耳の先端部分など薄い部分の再現性に問題のあることが実施例からうかがえる。今後の課題としては、このような問題を改善していくことが考えられる。

謝辞 本研究の機会を与えていただいた株式会社リコーソフトウェア研究所國井秀子所長と同第3研究室佐藤敏明室長に感謝します。また、点群データを提供していただいた株式会社リコー情報通信研究所江尻広一技師長と同黄英傑係長研究員に感謝します。最後に



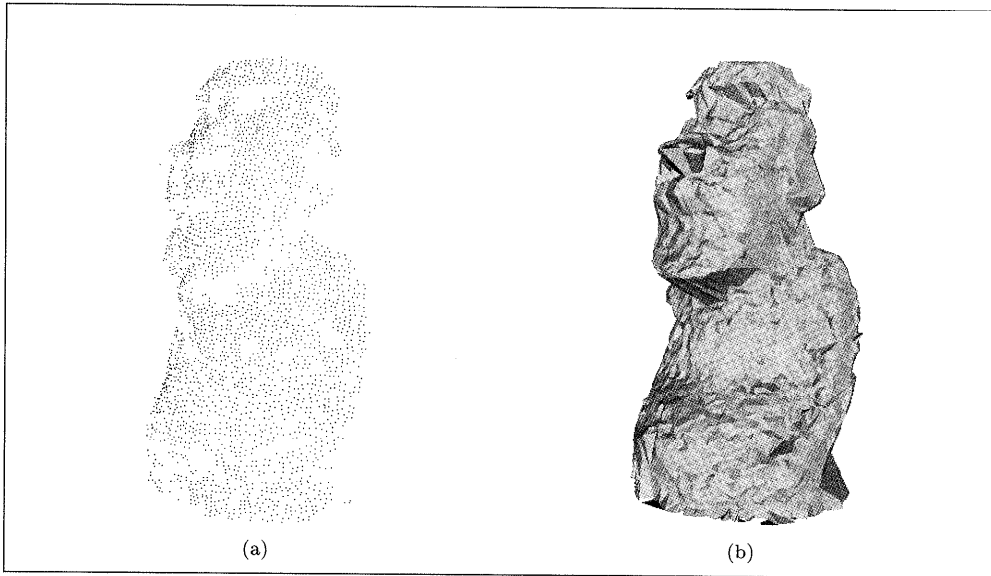


図 10 低密度・非一様点群への適用例：(a) shape from motion 手法による点群（株式会社リコー情報通信研究所提供），(b) 本手法による再構成モデル

Fig. 10 Reconstruction from non dense and non uniform points set: (a) points set from *shape from motion* (Courtesy of Information and Communication R&D Center of RICOH Company), (b) reconstructed model with our method.

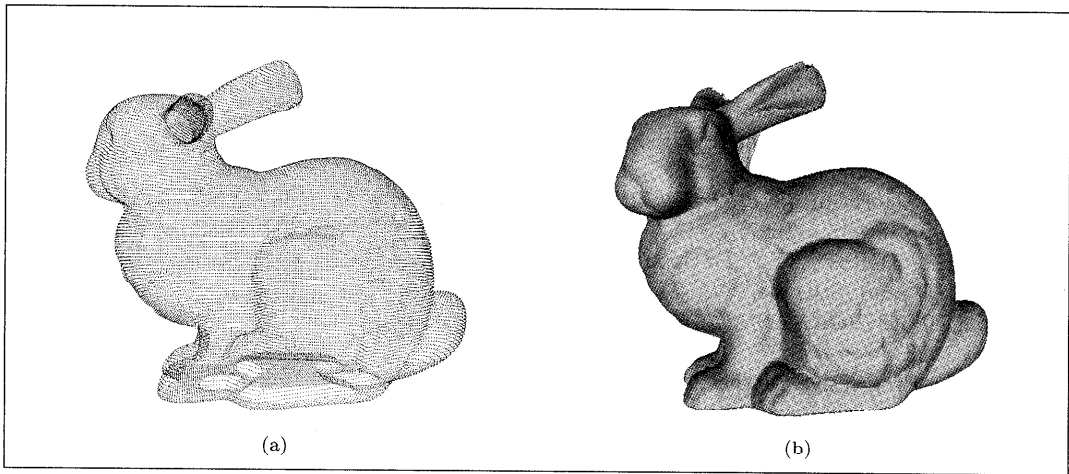


図 11 高密度・一様点群への適用例：(a) laser range scanner で計測された点群（Stanford University 公開データ），(b) 本手法による再構成モデル

Fig. 11 Reconstruction from dense and uniform points set: (a) points set from laser range scanner (Courtesy of Stanford University), (b) reconstructed model with our method.

点群データを一般に広く公開している The Stanford 3D Scanning Repository のメンバーに敬意を表するとともに感謝します。

参 考 文 献

- 1) Giblin, P. and Weiss, R.: Reconstruction of Surface form Profiles, *International Conference*

- on *Computer Vision*, pp.136-144 (1987).

- 2) Poleman, C.J. and Kanade, T.: A Paraperpective Factorization Method for Shape and Motion Recovery, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.19, No.3, pp.206-218 (1997).
- 3) Tomasi, C. and Kanade, T.: Shape and Motion from Image Streams under Orthography

- a Factorization Method, *International Journal of Computer Vision*, Vol.9, No.2, pp.137-154 (1992).
- 4) Veltkamp, R.C.: Boundaries through Scattered Points of Unknown Density, *Graphical Models and Image Processing*, Vol.57, No.6, pp.441-452 (1995).
  - 5) Bernardini, F., Bajaj, C.L., Chen, J. and Schikore, D.R.: A Triangulation-Based Object Reconstruction Method, *13th ACM Symposium on Computational Geometry*, pp.481-484 (1997).
  - 6) Turk, G. and Levoy, M.: Zippered Polygon Meshes from Range Images, *Proc. SIGGRAPH '94 Conference*, pp.311-318 (1994).
  - 7) Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W.: Surface Reconstruction from Unorganized Points, *Proc. SIGGRAPH '92 Conference*, pp.71-78 (1992).
  - 8) 芹田陽一郎, 渡邊大地, 千代倉弘明: ポリユースティング手法を用いた点群データからのポリゴンおよび曲面モデル自動再生成, 第13回 NICOGRAPH/MULTIMEDIA 論文コンテスト論文集, Vol.3, pp.70-79 (1997).
  - 9) Sourcy, M. and Laurendeau, D.: Multi-Resolution Surface Modeling from Multiple Range View, *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.348-353 (1992).
  - 10) Lorensen, W.E. and Cline, H.E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *Proc. SIGGRAPH '87 Conference*, pp.163-169 (1987).
  - 11) Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W.: Mesh Optimization, *Proc. SIGGRAPH '93 Conference*, pp.19-26 (1993).
  - 12) Hoppe, H.: Progressive Meshes, *Proc. SIGGRAPH '96 Conference*, pp.99-108 (1996).

- 13) DESIGNBASE: <http://www.ricoh.co.jp/sbc/soft/designbase>.
- 14) The Stanford 3D Scanning Repository: <http://www-graphics.stanford.edu/data/3Dscanrep>.

(平成10年9月3日受付)

(平成11年2月8日採録)



松岡 司 (正会員)

1984年大阪大学基礎工学部制御工学科卒業。同年三菱プレジジョン(株)入社。フライトシュミレータのビジュアルシステムの開発に従事。シャープ(株)を経て、1990年(株)リコー入社。以来、ソリッドモデラの研究開発に従事。曲面の連続性制御、特異点に興味を持つ。



植田 健治 (正会員)

慶應義塾大学大学院工学研究科修士課程修了。1983年(株)リコー入社、同社ソフトウェア研究所にて、CADシステム、ソリッドモデリングシステムの研究開発に従事。曲線、曲面等の形状処理に興味を持つ。



早野 勝之 (正会員)

1994年慶應義塾大学環境情報学部環境情報学科卒業。1996年慶應義塾大学大学院政策・メディア研究科修士課程修了。同年(株)リコー入社。ソリッドモデラの研究開発に従事。曲面のレンダリングに興味を持つ。