

## 正則化を利用した距離画像からの三次元形状モデルの生成

天野 晃<sup>†</sup> 坂口嘉之<sup>‡</sup> 美濃導彦<sup>†</sup> 池田克夫<sup>†</sup>  
<sup>†</sup>京都大学工学部 <sup>‡</sup>東洋紡織総合研究所

あらまし

本稿では、正則化を利用して距離画像から四角形面要素であらわされる三次元形状モデルを生成する方法について述べる。距離画像から生成される三次元形状モデルは、誤差が少なく、面要素の数が少なく、表面が滑らかであるということが望ましい。これらの拘束をうまく満たすような三次元形状モデルを生成する方法として、正則化理論を利用した。また、パラメータを変更することによって、誤差を少なくする、あるいは面要素の数が少なくなるような三次元形状モデルを生成することができる。

和文キーワード 距離画像 三次元形状モデル 正則化

## 3D Model Generation from a Range Image by Regularization

Akira Amano<sup>†</sup> Yoshiyuki Sakaguchi<sup>‡</sup> Michihiko Minoh<sup>†</sup> Katsuo Ikeda<sup>†</sup>  
<sup>†</sup> Faculty of Engineering, Kyoto University <sup>‡</sup> TOYOBO, ltd.

Abstract

In this paper, we describe about a method of generating 3D model which is represented as square surface patch, from range image. 3D model which is generated from range image, is expected to have less error, and less surface elements, and surface is smooth. We used regularization theory to satisfy these constraints. By changing parameters, we can get model of less error, or model of less surface elements.

英文 key words “Range Image” “Regularization” “3D Model”

## 1 はじめに

計算機の低価格、高性能に伴い種々の計測データのコンピュータグラフィックス表示の需要は高くなってきている。コンピュータグラフィックスでは、三次元形状モデルの作成に非常に手間がかかり、その自動生成に対する要求は大きい。

Grimson[2]による曲面の再構成では、再構成される曲面は一樣な格子上の点で表現されるため、データの量が多くなりコンピュータグラフィックスでの表示に適さない。Linら[3]による曲面の再構成では、再構成に用いられるデータが断面像でなければならないという制約があり、距離画像に対しては適用できない。田中ら[4]による距離画像からの三次元形状モデルの生成では、曲面を表現する面要素の数を適応的に変化させることにより、データ量と誤差をうまくバランスさせている。しかしながら、この方法では、均一な格子を再分割していくことにより各面要素を得るため、各格子点は移動せず、同じ曲面を表現するのに冗長な数の面要素を使っているといえる。

本研究では、曲面を表現する面要素の形状と数を適応的に変化させることにより、再帰的に四角形に分割された格子構造において面要素の数と形状が最適になるような三次元形状モデルを生成する手法を提案する。

## 2 距離画像からの三次元形状モデルの生成

### 2.1 三次元形状モデルの生成

距離画像は、格子上の点の関数値として、格子面から物体面までの距離が与えられているものである。

生成される三次元形状モデルは面要素の集合で表される曲面であるが、これは以下のような拘束を満たすものであることが望ましい。

1. 距離画像の値と形状モデルの曲面との誤差が小さい
2. 形状モデルの曲面を表す面要素の数が少ない
3. 形状モデルの曲面は滑らかである

距離画像にノイズが含まれていない場合、形状モデルの曲面と距離画像との誤差が0になる三次元形状モデルが最適である。しかしながら、ノイズがある場合には、この誤差としてノイズ分が残るはずであり、このような解を得るためには、3の滑らかさの拘束が必要になる。さらに、形状モデルの曲面を表す面要素の数は、少ない方が良く、このために2の条件が付け加えられる。

このようにいくつかの条件をもっともよく満たすような解を得る方法として、理論的な根拠が明確である、正則化理論がある。

### 2.2 正則化理論

未知量  $z$  が、線形操作  $A$  を受けた観測量  $y$  として観測されるとする。

$$y = Az \quad (1)$$

このとき、観測量  $y$  から未知量  $z$  を推定する逆問題を次の関数の最小化問題として定式化する手法を正則化理論と呼ぶ。

$$E = \|Az - y\|^2 + \lambda \|Pz\|^2 \quad (2)$$

ペナルティ関数と呼ばれる右辺第一項は、観測量  $y$  が誤差を含まないときは、 $z$  の最適な推定量においてその値が0になるような関数である。また、観測量  $y$  が誤差を含むときは観測量と推定量との差を表す関数となる。安定化汎関数と呼ばれる右辺第二項はこの問題の中で未知量  $z$  に関して一般的に与えられる制約条件を表す。

観測量  $y$  が誤差を含むとき、ペナルティ関数は最適な推定量においてもその値が0にならないが、解に関する一般的な性質を表す安定化汎関数とのバランスをとることによって最適な推定量を求めることができる。このバランスを決めるのが正則化パラメータ  $\lambda$  である。このとき、 $\lambda$  を小さくすると推定量  $z$  は観測量  $y$  に強く依存した解になり、大きくすると一般的な性質を重視した解になる。

式(2)の解は、最小化問題の解に対する必要条件であるオイラー方程式の形にして、連立方程式の解を求める問題として解くことができる。この連立方程式の係数行列は大規模かつ疎であることが多いため、数値解法として種々の反復法が用いられる。

## 3 正則化を利用した定式化

### 3.1 曲面の表現

曲面の表現方法には色々あるが、ここではデータの扱いやすさから、再帰的に四角形に分割された格子による面要素を用いる。また面要素を構成する各四角形は、不等辺四角形であるとする。これは、同じ面要素数で表される曲面でも、各面要素が不等辺四角形であるほうが表現できる曲面の自由度は大きいからである。四角形の面要素にはロフト曲面、線形 Coons 曲面、Bezier 曲面など様々なものがあるが、ここでは四角形の4頂点の座標から高速に計算できる双一次曲面を用いる。

面要素の4頂点の高さを  $f(x_0^f, y_0^f)$   $f(x_1^f, y_1^f)$   $f(x_2^f, y_2^f)$   $f(x_3^f, y_3^f)$  とする。これらの点による双一次曲面は  $f(x, y)$  は以下のように表される。

$$\begin{aligned} f(x, y) = & f(x_0^f, y_0^f)(1-u)(1-w) \\ & + f(x_1^f, y_1^f)(1-u)w \\ & + f(x_2^f, y_2^f)u(1-w) \\ & + f(x_3^f, y_3^f)uw \end{aligned} \quad (3)$$

なお、 $u$ 、 $w$ は以下の式を解くことによって $x$ 、 $y$ の関数として得られる。

$$x = x_0^f(1-u)(1-w) + x_1^f(1-u)w + x_2^f u(1-w) + x_3^f uw \quad (4)$$

$$y = y_0^f(1-u)(1-w) + y_1^f(1-u)w + y_2^f u(1-w) + y_3^f uw \quad (5)$$

ところで、各面要素は同じ大きさである必要はなく、複雑な形状のところには、より多くの面要素を配置し、形状が単純なところは少ない面要素で表現するほうが効率良く曲面を表現することができる。点を任意に配置して、それを四角形で覆っていく方法を用いた場合、点の数が増減することにヴォロノイ図などを作り、面要素を生成する必要がある。このような方法では、適応的に面要素の数を変化させる場合、計算量が多くなり過ぎるという問題がある。そこで、本研究では格子状の点を再帰的に四角形に分割していく。すなわち、それぞれの面要素を分割する場合、格子の中心に点を一つ増やすことによって四つの面要素に分割するという方法を用いた(図1参照)。こうすることによって点の増減による面要素の生成を容易に行うことができる。ところで、このような分割方法を用いると図1における点 $r$ のような点に、曲面の不連続が生じる。そこで、このような三つの面要素が隣接している点の高さは、点 $q$ 及び点 $s$ の線形補間により拘束し、自由に動く点は点 $q$ や点 $s$ のような四つの面要素が隣接している点のみとしている。

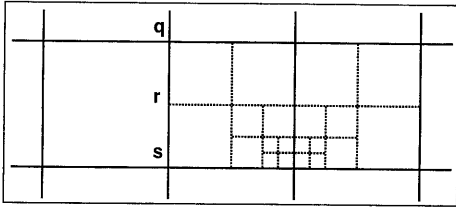


図1: 面要素の再分割

分割によって値が決まる関数である式(7)は、本来必要な面要素の数や、表示しやすい構造などを反映した関数にするべきであるが、これは表示装置などによって変化する関数である。ここでは一般的に、表示には面要素の数は少ないほうが良いという性質を用いて関数の値として面要素の数をを用いた。

### 3.2 問題の定式化

距離画像は、格子点 $x_{ij}^o$ 、 $y_{ij}^o$  ( $i = 1 \cdots N_x, j = 1 \cdots N_y$ ) 上の高さ $z_{ij}^o = s(x_{ij}^o, y_{ij}^o)$ と表現できる。また、形状モデルの曲面は再帰的に四角形に分割された格子上の点 $x_q^f$ 、 $y_q^f$  ( $q = 1 \cdots N_q$ ) 上の高さ $z_q^f = f(x_q^f, y_q^f)$ の点から生成される、四角形の面要素によって表される。また、 $\Delta$ は形状モデルの分割を表し、 $\phi(\Delta)$ は分割によって値が決まる関数を表す。

生成する形状モデルの曲面が満たすべき条件を式で表現すると、以下ようになる。

$$E_{error}(x, y) = (s(x, y) - f(x, y))^2 \quad (6)$$

$$E_{data} = \phi(\Delta) \quad (7)$$

$$E_{bend}(x, y) = f_{xx}^2 + f_{yy}^2 \quad (8)$$

$$E = \iint E_{error} dx dy + w_{data} E_{data} + w_{bend} \iint E_{bend} dx dy \quad (9)$$

各面要素を $Q_p$  ( $p = 1 \cdots N_p$ ) とし、式(9)を離散的に書き直すと以下のように表される。

$$E_{error}^s(Q_p) = \int_{Q_p} E_{error} dx dy \quad (10)$$

$$E = \sum_{p=1}^{N_p} E_{error}^s(Q_p) / (N_x N_y) + w_{data} E_{data} + w_{bend} \sum_{q=1}^{N_q} E_{bend}(x_q^f, y_q^f) / (N_x N_y) \quad (11)$$

関数 $E$ を最小化するような $x_q^f$ 、 $y_q^f$ 、 $z_q^f$ を求めることにより、三次元形状モデルが得られる。ただし、 $x_q^f$ 、 $y_q^f$ は格子の構造を変化させない範囲で移動させるものとする。

ところで、生成する三次元形状モデルには、目的によってある精度の曲面、あるいはある面要素数の曲面、あるいは面要素の形状が一定な曲面が必要である場合が考えられる。ここでは、これらの拘束を持たせた収束計算方法についても考える。

### 3.3 正則化を利用した三次元形状モデルの生成

生成しようとしている三次元形状モデルは式(9)を最小にするような形状モデルの曲面を持つものである。この式を正則化理論の枠組で捉えたと、ペナルティ関数にあたる式は式(6)にあたり、安定化汎関数にあたる式は式(7)及び式(8)にあたる。それぞれの関数が最適な解においてその値が最小になるという条件を満たせば、この式を最小化することによって最適な三次元形状モデルが得られることが保証される。

## 4 関数の収束計算

式11の関数を最小化するとき、変化させるものは $x_q^f$ 、 $y_q^f$ 、 $z_q^f$ 、及び分割の四つがある。

変化させる変数が面要素の位置を表す $x_q^f$ 、 $y_q^f$ 、 $z_q^f$ だけである場合、式(11)からオイラー方程式を導き、最急降下法などの方法で解くことができる。分割を変化させる場合、分割による各関数の値の変化を予測し、ある面要素を分割するかどうかを判断していくことによって最小化を行う。

ここでは、 $x_q^f$ 、 $y_q^f$ 及び $z_q^f$ のみを変化させる場合、分割を変化させる場合について考察し、次に両方を同時に変化させて最小化を行う方法について説明する。

#### 4.1 面要素形状変化による収束計算

分割を変化させないで、形状のみを変化させる場合、 $E_{data}$  は定数となるので、収束計算は以下の式の最小化となる。

$$E = E_{error} + w_{bend} E_{bend} \quad (12)$$

上式からオイラー方程式を導き、これを以下の式による反復解法で解くことによって上式を最小化することができる。なお、 $\gamma$  は反復計算におけるステップ幅である。

$$\begin{aligned} x_q^{f\ t+1} &= x_q^f - \gamma \left( \frac{\partial E_{error}}{\partial x} + w_{bend} \frac{\partial E_{bend}}{\partial x} \right) \\ y_q^{f\ t+1} &= y_q^f - \gamma \left( \frac{\partial E_{error}}{\partial y} + w_{bend} \frac{\partial E_{bend}}{\partial y} \right) \\ z_q^{f\ t+1} &= z_q^f - \gamma \left( \frac{\partial E_{error}}{\partial z} + w_{bend} \frac{\partial E_{bend}}{\partial z} \right) \end{aligned} \quad (13)$$

図2のように、点  $p_q = (x_q^f, y_q^f, z_q^f)$  に隣接する四つの面要素を  $Q_a, Q_b, Q_c, Q_d$  とし、点  $p_q$  に隣接する点を  $p_k = (x_k^f, y_k^f, z_k^f)$ 、 $p_l = (x_l^f, y_l^f, z_l^f)$ 、 $p_m = (x_m^f, y_m^f, z_m^f)$ 、 $p_n = (x_n^f, y_n^f, z_n^f)$  とする。また、 $Q_a, Q_b, Q_c, Q_d$  の重心の座標を  $(x_a, y_a)$ 、 $(x_b, y_b)$ 、 $(x_c, y_c)$ 、 $(x_d, y_d)$  とし、 $p_q$  からの距離をそれぞれ  $l_a, l_b, l_c, l_d$  とする。同様に  $p_k, p_l, p_m, p_n$  から  $p_q$  までの距離をそれぞれ  $r_k, r_l, r_m, r_n$  とする。このとき点  $p_q$  における  $E_{error}$ 、 $E_{bend}$  の偏微分はそれぞれ以下のように近似できる。

$$\begin{aligned} \frac{\partial E_{error}}{\partial x} &= \\ & \frac{1}{4} (E_{error}^s(Q_a) \frac{x_a - x_q^f}{l_a} + E_{error}^s(Q_b) \frac{x_b - x_q^f}{l_b} \\ & + E_{error}^s(Q_c) \frac{x_c - x_q^f}{l_c} + E_{error}^s(Q_d) \frac{x_d - x_q^f}{l_d}) \end{aligned} \quad (14)$$

$$\begin{aligned} \frac{\partial E_{bend}}{\partial x} &= \\ & \frac{1}{4} \left( (z_k^f - z_q^f) \frac{x_k^f - x_q^f}{r_k} + (z_l^f - z_q^f) \frac{x_l^f - x_q^f}{r_l} \right. \\ & \left. + (z_m^f - z_q^f) \frac{x_m^f - x_q^f}{r_m} + (z_n^f - z_q^f) \frac{x_n^f - x_q^f}{r_n} \right) \end{aligned} \quad (15)$$

$$\begin{aligned} \frac{\partial E_{error}}{\partial y} &= \\ & \frac{1}{4} (E_{error}^s(Q_a) \frac{y_a - y_q^f}{l_a} + E_{error}^s(Q_b) \frac{y_b - y_q^f}{l_b} \\ & + E_{error}^s(Q_c) \frac{y_c - y_q^f}{l_c} + E_{error}^s(Q_d) \frac{y_d - y_q^f}{l_d}) \end{aligned} \quad (16)$$

$$\begin{aligned} \frac{\partial E_{bend}}{\partial y} &= \\ & \frac{1}{4} \left( (z_k^f - z_q^f) \frac{y_k^f - y_q^f}{r_k} + (z_l^f - z_q^f) \frac{y_l^f - y_q^f}{r_l} \right. \\ & \left. + (z_m^f - z_q^f) \frac{y_m^f - y_q^f}{r_m} + (z_n^f - z_q^f) \frac{y_n^f - y_q^f}{r_n} \right) \end{aligned} \quad (17)$$

$$\begin{aligned} \frac{\partial E_{error}}{\partial z} &= \frac{1}{4} (E_{error}^s(Q_a) + E_{error}^s(Q_b) \\ & + E_{error}^s(Q_c) + E_{error}^s(Q_d)) \end{aligned} \quad (18)$$

$$\begin{aligned} \frac{\partial E_{bend}}{\partial z} &= \frac{1}{4} \left( (z_k^f - z_q^f) + (z_l^f - z_q^f) \right. \\ & \left. + (z_m^f - z_q^f) + (z_n^f - z_q^f) \right) \end{aligned} \quad (19)$$

なお、 $E_{error}^s(Q_p)$  は以下のような関数である。

$$E_{error}^s(Q_p) = \int_{Q_p} 2(s(x, y) - f(x, y)) dx dy \quad (20)$$

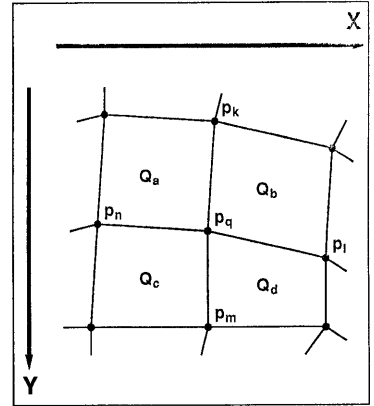


図2: 関数の計算

#### 4.2 再分割による収束計算

面要素の形状を変化させず、面要素の再分割によって関数を最小化する方法について説明する。

ある面要素を再分割するかどうかは、反復計算の途中で式(11)が再分割によって減少するかどうかで判断する。これには各面要素において、分割しなかった場合の  $E_{error}$  及び  $E_{bend}$  の減少と分割した場合の  $E_{error}$ 、 $E_{bend}$ 、 $E_{data}$  の増減を計算すれば良い。反復計算のあるステップで、ある面要素が分割されなかった場合、式(9)の減少  $\delta_n$  は、式(20)及び式(19)を用いて以下のように近似できる。面要素  $Q_p$  の四つの角を  $p_k, p_l, p_m, p_n$  とする。

$$\delta_{error}(Q_p) = E_{error}^s(Q_p) \quad (21)$$

$$\begin{aligned} \delta_{bend}(Q_p) &= \frac{1}{4} \left( \frac{\partial E_{bend}}{\partial z} \Big|_{p_k} + \frac{\partial E_{bend}}{\partial z} \Big|_{p_l} \right. \\ & \left. + \frac{\partial E_{bend}}{\partial z} \Big|_{p_m} + \frac{\partial E_{bend}}{\partial z} \Big|_{p_n} \right) \end{aligned} \quad (22)$$

$$\delta_n = \gamma (\delta_{error}(Q_p) + w_{bend} \delta_{bend}(Q_p)) \quad (23)$$

面要素  $Q_p$  において、分割を行った場合の  $E_{data}$  の減少は、 $\delta_{data} (= -3)$  と書くことができる。ところで、分割を行った場合の  $E_{error}$  及び  $E_{bend}$  の減少  $\delta_d$  は、実際に分割を行なうことにより計算できる。

このとき、面要素  $Q_p$  において、以下の式が満たされると分割が行われる。

$$\delta_n < \delta_d + w_{data}\delta_{data} \quad (24)$$

一つの面要素が再分割されると、その面要素の重心に新たに一つの点が必要になる。また、その面要素の各辺の中心に四つの点が必要となる。

### 4.3 形状変化・再分割による収束計算

前節までで、形状のみを変化させた場合と再分割を行っていく場合の収束計算方法を述べた。ここではこれらを同時に変化させた場合の収束計算について説明する。

ところで、形状を変化させる収束計算方法と再分割による収束計算は変化させるものが独立であるから、基本的にはこれを同時に行えばよい。この場合、式 13 は以下になる。

$$\begin{aligned} x_q^{t+1} &= x_q^t - \gamma \left( \frac{\partial E_{error}}{\partial x} + w_{data} \frac{\partial E_{data}}{\partial x} + w_{bend} \frac{\partial E_{bend}}{\partial x} \right) \\ y_q^{t+1} &= y_q^t - \gamma \left( \frac{\partial E_{error}}{\partial y} + w_{data} \frac{\partial E_{data}}{\partial y} + w_{bend} \frac{\partial E_{bend}}{\partial y} \right) \\ z_q^{t+1} &= z_q^t - \gamma \left( \frac{\partial E_{error}}{\partial z} + w_{data} \frac{\partial E_{data}}{\partial z} + w_{bend} \frac{\partial E_{bend}}{\partial z} \right) \end{aligned}$$

$E_{data}$  は分割によって値が決まる関数であるから、 $x$ 、 $y$ 、 $z$  方向への偏微分はそれぞれ 0 と考えることができる。

このような方法を用いると、距離画像と生成する三次元形状モデルとの誤差がある値になるまで収束計算を行う、あるいは面要素の数がある値のときに最適な面要素の配置を計算することが容易に行える。そのため、コンピュータグラフィックスに要求される三次元形状モデルとして、処理時間の問題から面要素の数を制限したり、正確さを要求されるために誤差を一定値以下にするといった条件を満たすモデルを生成することが可能である。

## 5 実験

以下の実験では距離画像として、図 3 を用意し、三次元形状モデルを生成する実験を行った。なお、距離画像は  $128 \times 128$  の点から構成され、ノイズを含まないとする。また、距離画像の値は最大値が 1 に正規化されているとする。

### 5.1 面要素形状変化による収束

面要素の形状を変化させる方法による、三次元形状モデルの収束計算について実験を行った。

初期面要素として、 $6 \times 6$  の格子上の正方形を用いて収束計算を行った。実験は、 $w_{bend}$  が 0.0 と 10.0 のときについて行った。 $E_{error}$ 、 $E_{bend}$ 、及び式 (11) の変化、生成した三

次元形状モデル、距離画像との誤差を濃淡で表した画像を図??及び図??に示す。なお、このときの  $E_{error}$  の値はそれぞれ 0.0047、0.0051 であった。

生成される三次元形状モデルは、稜線などの複雑な形状の部分に面要素が少なく、正確には再現されていない。このような部分に対してはより多くの面要素が必要である。また  $w_{bend}$  を大きくすることによって、誤差は大きい表面が滑らかな三次元形状モデルを得ることができることがわかる。

### 5.2 面要素数変化による収束

面要素の数のみを変化させて三次元形状モデルを生成する方法について実験を行った。

初期面要素として、 $4 \times 4$  の格子上の正方形を用いて収束計算を行った。 $w_{bend}$  は 0 とし、 $w_{data}$  が 0.0001 と 0.00001 の場合について実験を行った。 $E_{error}$ 、 $E_{bend}$ 、及び式 (11) の変化、生成した三次元形状モデル、距離画像との誤差を濃淡で表した画像を図5に示す。このときの  $E_{error}$  の値はそれぞれ 0.0056、0.0005 であり、面要素の数はそれぞれ 88 個、239 個であった。

この結果からわかるように、面要素の数を増やしていけば、単調に誤差は減少する。また、式 (11) の値が最小になる時の面要素の数は、 $w_{data}$  が小さいほど多くなる。生成された三次元形状モデルは入力となる距離画像を忠実に再現しており、良好な結果が得られている。しかしながら実験では、同じ数の面要素を使った場合、形状を変化させた収束方法の方が誤差の少ないモデルが生成できた。

### 5.3 面要素数・形状変化による収束

最後に面要素数及びその形状を変化させて三次元形状モデルを生成する収束計算を行ったときの結果を示す。

はじめに、初期面要素として、 $4 \times 4$  の格子上の正方形を用いて、図3の距離画像に対して、 $w_{data} = 0.0$ 、 $w_{bend} = 10.0$  の場合と、 $w_{data} = 0.00001$ 、 $w_{bend} = 0.0$  の場合の二通りについて実験を行った。 $E_{error}$ 、 $E_{bend}$ 、及び式 (11) の変化、生成した三次元形状モデル、距離画像との誤差を濃淡で表した画像を図??に示す。

生成した三次元形状モデルの面要素数はそれぞれ、48 個 160 個であった。また、 $w_{data} = 0.0$ 、 $w_{bend} = 10.0$  の場合の  $E_{error}$  は 0.0150、 $w_{data} = 0.00001$ 、 $w_{bend} = 0.0$  の場合の面要素数が 88 個の時の  $E_{error}$  の値は 0.0038 であった。

結果からわかるように、同じ面要素数の場合、分割のみを変化させた場合に比べて、形状と分割を変化させた場合の方が、 $E_{error}$  は小さくなる。なお、面要素数が 88 のときの分割のみを変化させた場合と、形状と分割を変化させた場合の結果を図??に示す。これに対して、同じ  $E_{error}$  の値に対する、形状のみを変化させた場合と形状と分割を変化させた場合を単純に比較することはできないが、形状と分割を変化させる場合に、収束したときの  $E_{error}$  の値が形状のみを変化させた場合に収束した時の  $E_{error}$  と一致するようにパラメータを調整すれば、形状と分割を変化させた場合の方が面要素数が少ないと考えられる。

## 6 おわりに

本研究では、距離画像を用いた三次元形状モデルの生成に正則化理論を応用することによって、三次元形状モデルの精度または、面要素数がある値にしたときに最適なモデルを生成する方法について提案した。この方法の特徴は以下の通りである。

- 入力となる距離画像と生成した三次元形状モデルの表面である曲面との誤差がある値以下になるような拘束を持たせたモデル生成ができる。
- 生成する三次元形状モデルの表面である曲面を表すのに必要な面要素数が決まっているときに、最適な面要素の配置になるようにモデルの生成を行うことができる。

今後は、面積がなるべく均一になる、面要素の各辺の長さがなるべく均一になる、などの新たな条件を加えたときの収束計算の方法について研究を進める予定である。

## 参考文献

- [1] 吉本 富士市, 市田 浩三, 清野 武, “区分的3次関数を用いた2次元データの平滑化の自動的方法,” 情報処理, vol. 18, no. 2, pp. 128-134, 1977.
- [2] W.E.L.Grimson, “An Implementation of a Computational Theory of Visual Surface Interpolation,” CVGIP, 22, pp. 39-69, 1983.
- [3] W.C.Lin, S.Y.Chen, “A New Surface Interpolation Technique for Reconstructing 3D Objects from Serial Cross-Sections,” CVGIP, 48, pp. 123-143, 1989.
- [4] 田中 弘美, 岸野 文朗, “3D 曲率を用いた適応的形状復元,” 画像の認識・理解シンポジウム予稿集, pp. 397-404, 1992.
- [5] Poggio, T., Torre, V., Koch, C., “Computational vision and regularization theory,” Nature, 317, 6035, pp. 314-319, 1985.

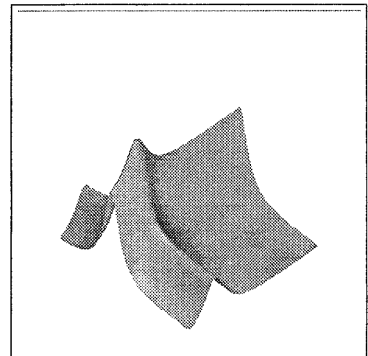
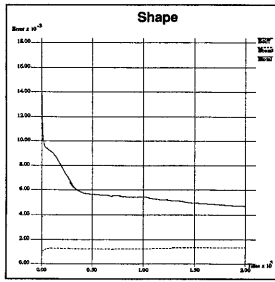
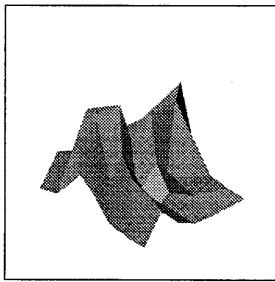


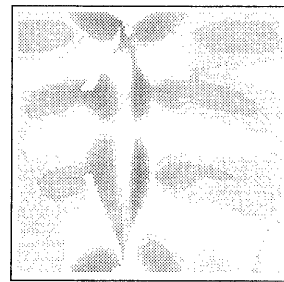
図 3: 距離画像の三次元表示



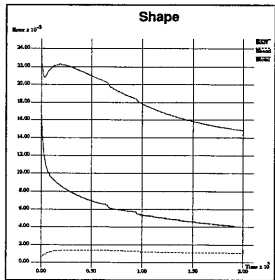
(a) 各関数の変化 ( $w_{bend} = 0.0$ )



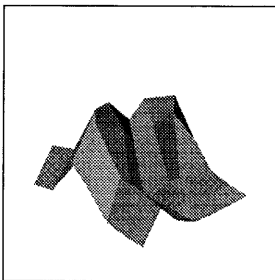
(b) 三次元表示 ( $w_{bend} = 0.0$ )



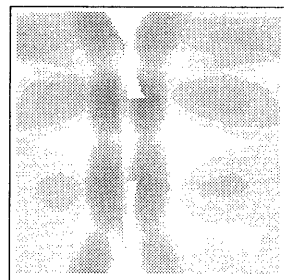
(c) 誤差 ( $w_{bend} = 0.0$ )



(d) 各関数の変化 ( $w_{bend} = 10.0$ )

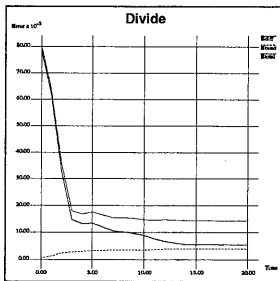


(e) 三次元表示 ( $w_{bend} = 10.0$ )

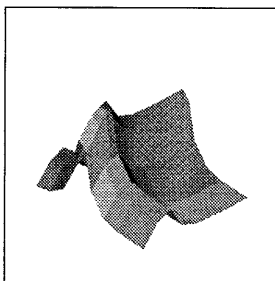


(f) 誤差 ( $w_{bend} = 10.0$ )

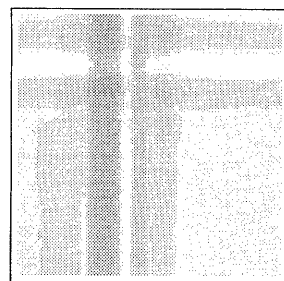
図 4: 形状変化によるモデルの生成



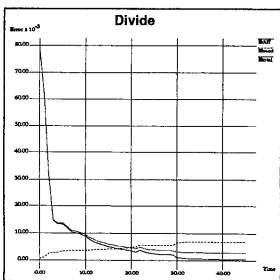
(a) 誤差の変化 ( $w_{data} = 0.0001$ )



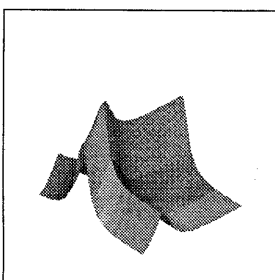
(b) 三次元表示 ( $w_{data} = 0.0001$ )



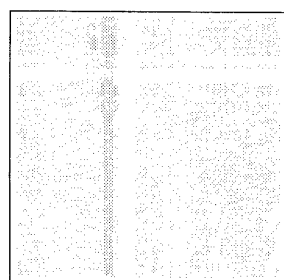
(c) 誤差 ( $w_{data} = 0.0001$ )



(d) 誤差の変化 ( $w_{data} = 0.00001$ )

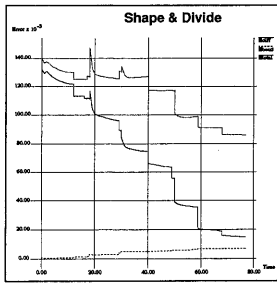


(e) 三次元表示 ( $w_{data} = 0.00001$ )

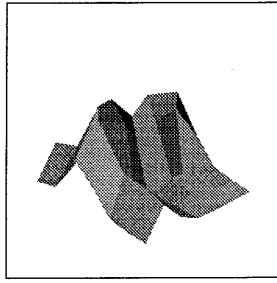


(f) 誤差 ( $w_{data} = 0.00001$ )

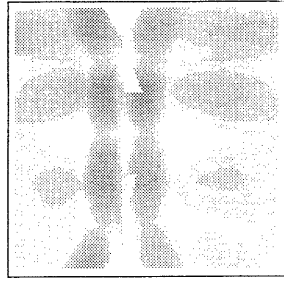
図 5: 再分割によるモデルの生成



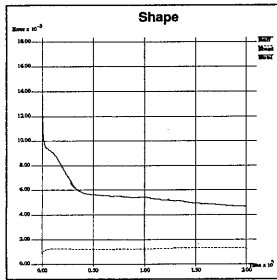
(a) 各関数の変化 ( $w_{bend} = 10.0$ ,  $w_{data} = 0.0$ )



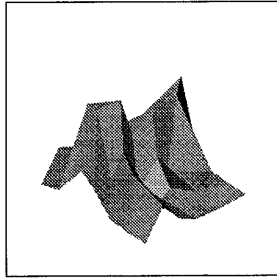
(b) 三次元表示 ( $w_{bend} = 10.0$ ,  $w_{data} = 0.0$ )



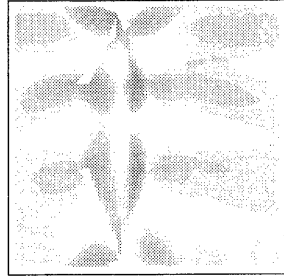
(c) 誤差 ( $w_{bend} = 10.0$ ,  $w_{data} = 0.0$ )



(d) 各関数の変化 ( $w_{bend} = 0.0$ ,  $w_{data} = 0.00001$ )

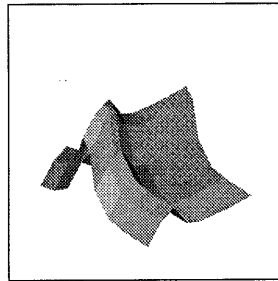


(e) 三次元表示 ( $w_{bend} = 0.0$ ,  $w_{data} = 0.00001$ )

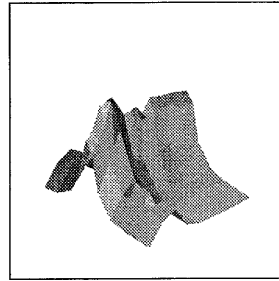


(f) 誤差 ( $w_{bend} = 0.0$ ,  $w_{data} = 0.00001$ )

図 6: 形状変化と分割によるモデルの生成



(a) 分割のみ



(b) 分割と形状変化

図 7: 同面要素数の時の比較