

距離関数を用いた未知視点から得た複数の距離画像の 重ね合わせ手法

清水 郁子 出口 光一郎
東京大学工学部

複数の視点から得た距離画像の重ね合わせ手法について述べる。あらかじめおよその位置関係が分かっているとき、より精度の高い視点間の位置関係を求めることにより重ね合わせを行う。本研究では、基準となる対象上の各点からの距離関数を求める。距離関数とは、空間の各点から基準となる対象点のなかで一番近い点までの距離を表すものである。距離関数をあらかじめ求めることにより、点の対応を探索することなく基準となる対象点までの距離が得られる。これを利用して、重ね合わせた距離画像の重なり方を容易に調べることができるようになるので、得られた視点間の位置関係が真の位置であるのかどうかを見分けることができる。また、距離関数の領域ごとにラベルをつけることによって、オクルージョン等によって生じる誤対応を減らすことができる。

A Method for Positional Registration of Multiple Range Images Using Distance Map

Ikuko Shimizu and Koichiro Deguchi

**Faculty of Engineering, University of Tokyo,
Bunkyo-ku, Tokyo, Japan**

A method for registration of multiple range images is proposed. A discrete distance map is used in the registration process. It is defined as a set of distances from each spatial point on the grid of the 3D space to the nearest point on the object. Once the distance map is calculated for a base object points, searching the nearest point becomes no need to obtain the distance from a given range image point to the object. Using this map, we easily examine how fine two range images are registered and find false registration. We also reduce the incorrect correspondence caused by occlusion by labeling the regions of the distance map.

1 はじめに

対象物体の表面までの距離を非接触に測定するセンサであるレンジファインダ [1] による測定では、センサから物体表面上の点までの距離データ、すなわち視点の位置に依存した座標系での物体表面上の三次元座標が得られる。得られた距離を各画素に並べてできる画像を、距離画像と呼ぶ。

一般に、一枚の距離画像では、物体の裏側や凸部に隠された部分、計測範囲外の領域など、計測できない部分がある。そこで、対象物体の全体形状を得るためには、センサ（または物体）を動かしていろいろな視点から計測し、得られた距離画像を統合することになる。

本研究の目的は、それぞれ視点の位置に依存する座標系で得られた複数の距離画像を統合することにより、物体の全表面の三次元形状を計測することである。

多数の視点から得られた距離画像を統合するためには、各視点の位置関係を決め、視点の位置に依存する座標系での値であるそれぞれの距離画像を基準となる座標系で表す必要がある。視点の位置に依存する座標系での値と、基準となる座標系での値との間の関係は、回転と並進の組み合わせとして表すことができる。これらをまとめて、「変換パラメータ」と呼ぶことにする。

本研究では、あらかじめなんらかの特徴量を用いるなどして視点間のおよその位置関係がわかっている場合に、より精度の高い変換パラメータを求め、複数の距離画像を重ね合わせる手法について述べる。精度の高い変換パラメータを求めることにより、たくさんの視点から計測したデータを効果的に統合し、より信頼性の高い計測結果を得ることが可能になる。

あらかじめおよその位置関係が分かっている場合の手法は様々なものが提案されている [2][3][4] が、変換パラメータを求める有効な方法として、ICP アルゴリズム [5] を応用したもの（および、同様の考え方を用いたもの [3]）が知られている。ICP アルゴリズムはもともとは物体の完全な形状が既知であるときの、部分形状の姿勢推定のための手法である。二つの形状の「距離」、すなわち、変換パラメータによって変換された片方の形状に含まれる各点と、その点に一番近いもう片方の形状中の点（＝対応点）との間の距離の和を最小化する変

換パラメータを求める。そのような変換パラメータは、変換パラメータを更新するたびに、対応点を更新することによって、探索される。

この方法を距離画像の重ね合わせに用いる場合、次に挙げるような原因から、良い結果が得られないことがある。

- オクルージョンなどによって、片方のデータがもう片方のデータに完全に含まれるわけではないため、誤対応が生じる。
- 距離画像は離散的にしか得られないために、違う視点から物体表面上の全く同じ点を測定することは一般にはない。
- それぞれの計測点は測定誤差を含んでいる。

更に、多数の距離画像を精度良く重ね合わせるためには、得られた多数の距離画像のデータを全体で整合のとれるような変換パラメータを求めるなければならない。そのために、得られたデータ全てを使って、すべての視点の運動パラメータを一度に求めることが望ましい。しかし、ICP アルゴリズムを応用する場合には、得られた全てのデータを一度に使うと最小化が複雑になってしまい困難であるため、順番にデータを重ね合わせていかなるを得ない。

多数の距離画像を順番に重ねていく場合には、上に挙げた3点に加えて、良い結果が得られない原因として、次のようなことも考えられる。

- 最小化によって得られた変換パラメータが、真の変換パラメータではない。

順番に重ね合わせられた距離画像の中に、真の変換パラメータ以外で重ね合わされているものがあると、最終的に得られた結果に大きく影響を与える。

真の変換パラメータが得られない場合は二つある。まず、変換パラメータの初期値が十分良いものでなかったために、ローカルミニマムに収束してしまい、真の最小値に到達できない場合、また、重なっている部分が、例えば球面や平面のようななめらかな形状であるとき、最小値になる変換パラメータが真の変換パラメータ以外にも（多数）ある場合である。

4 番目の問題点を解決するためには、最小化によって得られた変換パラメータが、真の変換パラ

メータであるのか、そうでない可能性があるのかを見分ける必要がある。

見分けるための方法として、最小値になった位置の周りでの「重なり方」、つまり、少しずつ重ね合わせた点を動かしたとき、重ねたものと重ねられたものの距離がどう変化するかを調べることが考えられる。これを実行する場合、少しずつ動かす度に点の対応を探索すると、計算時間が膨大になってしまい、現実的ではない。

本研究では、あらかじめ重ね合わせられる方のデータ点のまわりの各点で、データのなかで一番近い点までの距離を求めておく。これを重ね合わせられるデータに関する距離関数と呼ぶことにする。距離関数を求めてしまえば、対応点の探索することなく対応点までの距離を知ることができる。

距離関数をあらかじめ求めておくことにより、「重なり方」を現実的な実行時間で調べることができる。また、距離関数に簡単なラベル付けをすることにより、誤対応を減らす手法についても述べる。

第2章で処理の概要を述べ、それぞれの処理について第3, 4, 5章で説明する。第6章で実験結果を示す。

2 処理の概要

N 枚の距離画像を重ね合わせることを考える。 $i(i=1, \dots, N)$ 番目の距離画像中に含まれる物体の部分形状データの集合 X_i とする。各距離画像に含まれる部分形状は、 M_i 個のデータ点から成るものとする。

$$X_i = \{x_j^i\} (i=1, \dots, N, j=1, \dots, M_i) \quad (1)$$

基準となる距離画像に対し、以下のようにして他の距離画像を基準となる距離画像と同じ座標系で表し、順番に重ね合わせてゆく。 k 番目の距離画像を重ね合わせるとき、すでに基準となる座標系で表されている距離画像の集合を W_k とおく。まず一番目の距離画像を基準にすることになると、 $W_1 = \{X_1\}$ である。

1. k 番目の距離画像までの重ね合わせが終了して、それらの対象点の集合 W_k を得ているとする。基準となる距離画像の集合 W_k 内の各点に対する距離関数 $D_k(x)$ を計算する。(第3章)

2. $k+1$ 番目の距離画像 X_{k+1} を、あらかじめ与えられるおおよその変換パラメータの初期値 T_{k+1}^0 で変換する。 T_{k+1}^0 で変換された点の集合を $T_{k+1}^0(X_{k+1})$ と表す。

3. x_n を X_{k+1} 内の点を変換パラメータで変換した点として、距離の二乗和 $\sum_{n=1}^{M_{k+1}} D_k(x_n)$ を最小にする変換パラメータ \hat{T}_{k+1} を T_{k+1}^0 を初期値として反復法で求める。(第4章)

4. $\hat{T}_{k+1}(X_{k+1})$ の各点と W_k 内の点との重なり方を調べる。(第5章)

- もし、 \hat{T}_{k+1} による距離の二乗和がローカルミニマムなら、変換パラメータの初期値を変えて3へ。

- もし、同程度の重なり方をする位置が複数あったら、この時点での重ね合わせはあいまいさが残ると判定し、もう一度後で W_k の点の数が増えた段階で、この距離画像の重ね合わせを再度試みることにする(基準となる距離画像の集合はかわらない。 $W_{k+1} = W_k$.)

- そうでなければ、基準となる距離画像の集合に $k+1$ 番目の距離画像を加える。 $W_{k+1} += X_{k+1}$ 。

5. $k++$; 1へ。

以下にそれぞれの処理について説明する。

3 距離関数の計算

基準となる点の集合 W_k が与えられたとき、 W_k の距離関数 $D_k(x)$ を、点 x から W_k への最短距離の二乗として、以下のように定義する。

$$D_k(x) = \min_{v \in W_k} d^2(x, v) \quad (2)$$

ただし、 $d^2(x, v)$ は x と v の距離の二乗であるとする。

空間をボクセルに区切り、各ボクセル x において、 $D_k(x)$ をあらかじめ計算する。ここでは、高速算法である Fast Marching 法 [6] を用いる。

Fast Marching 法では、式 (2) で定義された距離関数の平方根 $u(x)$ が $|\nabla u| = 1$ を満たすことから、この偏微分方程式を解く。基準となる点の集

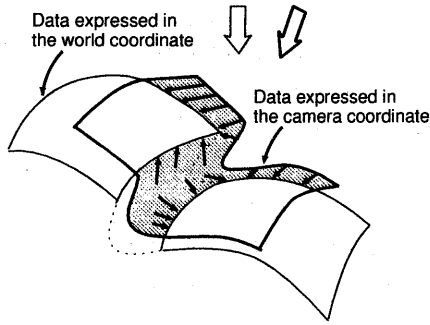


図1. 誤対応のおこる様子。斜線部分は基準となる形状に含まれていない領域。斜線部分に含まれる点は、基準となる形状の端の点に対応付けられてしまう。

合 $v \in W_k$ では、 $u(v) = 0$ であるからこれを初期値として、すべての領域で $|\nabla u| = 1$ を満たすように $u(x)$ の値の小さい方から順に決定する。

あらかじめおよその位置関係は分かっているので、ある程度以上離れたものは対応点がないと考えられる [3]。そこで、距離関数は W_k から距離 D_k の範囲だけ求めれば良い。

ここで、オクルージョン等によって起る誤対応について考える。重ね合わせる方の距離画像に含まれているが、基準となる距離画像には含まれていない領域 (基準となる距離画像では計測されていない領域) にあるような点は、対応点はないが、その点が基準となる距離画像に含まれているかどうかは分からないため、誤った対応点に対応付けられてしまう。

図1では、斜線部分が、基準となる距離画像中に対応する領域のない部分である。斜線部分に含まれる点の対応点を矢印で示す。これらの点の対応点は、曲面上に得られている基準となる点の集合の端の点に対応付けられている可能性が高い。

そこで、端点が最近点であるような領域に「端点」のラベルをつける。これは、Fast Marching 法で距離を計算するときに簡単な処理を追加することで可能である。

データ点が「端点である」とは、距離画像中で、その点の画像上で隣り合う点を持たないか、 xy 方向の4近傍の点との奥行きの差が閾値以上である点の一つでもあるような点であるとする。距離画像を重ね合わせたデータ点の集合では、距離画像一枚毎に端点をラベルづけし、各端点について、すべての他の距離画像中のデータ点のうち端点でない点との距離が閾値以内であれば、その点は「端

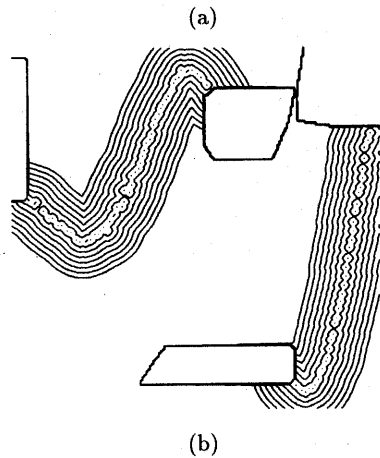
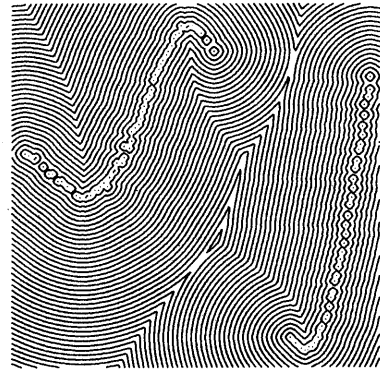


図2. 2次元の曲線の距離関数の等高線 (a) 領域全体の距離関数 (b) 曲線からの距離が閾値以下でかつ対応点が端点でない領域

点ではない」とする。

以上の説明のため、二次元の場合での例を図2に示す。(a)は曲線に対する領域全体の距離関数の等高線、(b)は曲線からの距離が閾値以下でかつ対応点が端点でない領域を示したものである。

4 位置合わせ

4.1 最小化する誤差関数

W_k と X_i の距離がもっとも近くなるような変換パラメータ T を求めるために、

$$J = \frac{1}{\sum_{m=1}^M w(T(x_m))} \sum_{m=1}^M w(T(x_m)) D_k(T(x_m)) \quad (3)$$

を最小化する。

ただし、

$$w(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \text{のラベルが「端点」} \\ 0 & D_k(\mathbf{x}) > D_t \\ 1 & \text{それ以外} \end{cases} \quad (4)$$

である。

4.2 最小化の方法

変換パラメータは、回転と並進の組み合わせであるが、回転を表すのに四元数、並進は並進ベクトルによって表すものとする。

並進ベクトルを \mathbf{t} とおき、回転を表す四元数を

$$\mathbf{q} = \begin{pmatrix} q_0 \\ \mathbf{q} \end{pmatrix} \quad (5)$$

とおくと、

$$T(\mathbf{x}_k) = \mathbf{x} + 2q_0\mathbf{q} \times \mathbf{x}_k - 2(\mathbf{q} \times \mathbf{x}) \times \mathbf{q} + \mathbf{t} \quad (6)$$

と表すことができる。ただし、 \mathbf{q} は3次元ベクトルである。

式(3)を $\|\mathbf{q}\| = 1$ の条件の元で J を最小化する。ラグランジュ未定乗数法により、

$$F = J + \lambda(q_0^2 + \mathbf{q} \cdot \mathbf{q} - 1) \quad (7)$$

として、

$$\frac{\partial F}{\partial \mathbf{t}} = \sum_{k=1}^N w(T(\mathbf{x}_k)) \nabla \tau_k = \mathbf{0} \quad (8)$$

$$\begin{aligned} \frac{\partial F}{\partial q_0} &= 2 \sum_{k=1}^N w(T(\mathbf{x}_k)) (\nabla \tau_k)^t (\mathbf{q} \times \mathbf{x}_k) + 2q_0\lambda \\ &= 0 \end{aligned} \quad (9)$$

$$\begin{aligned} \frac{\partial F}{\partial \mathbf{q}} &= \sum_{i=1}^N w(T(\mathbf{x}_k)) \left(\frac{\partial T(\mathbf{x}_k)}{\partial \mathbf{q}} \right)^t \nabla \tau_k + 2\lambda \mathbf{q} \\ &= \mathbf{0} \end{aligned} \quad (10)$$

$$\frac{\partial F}{\partial \lambda} = q_0^2 + \mathbf{q} \cdot \mathbf{q} - 1 = 0 \quad (11)$$

を満たす \mathbf{q} , q_0 , \mathbf{t} が求める解である。ただし、

$$\nabla \tau_k = \left(\begin{array}{c} \frac{\partial \tau}{\partial x} \\ \frac{\partial \tau}{\partial y} \\ \frac{\partial \tau}{\partial z} \end{array} \right) \Bigg|_{\mathbf{x}=T(\mathbf{x}_k)} \quad (12)$$

である。まとめると、

$$A_k = \left(\frac{\partial T(\mathbf{x}_k)}{\partial \mathbf{q}} \right)^t, \quad b_k = 2\mathbf{q} \times \mathbf{x}_k \quad (13)$$

として、

$$\begin{aligned} f &= \left[\begin{array}{c} \sum_{k=1}^N A_k \nabla \tau_k - \frac{1}{\sqrt{1-\mathbf{q} \cdot \mathbf{q}}} \left(\sum_{k=1}^N b_k^t \nabla \tau_k \right) \mathbf{q} \\ \sum_{k=1}^N \nabla \tau_k \end{array} \right] \\ &= \mathbf{0} \end{aligned} \quad (14)$$

を満たす解を求めればよい。

式(14)の解はNewton法を用いて解く。未知数 \mathbf{q} , \mathbf{t} をまとめて

$$\mathbf{a} = \begin{bmatrix} \mathbf{q} \\ \mathbf{t} \end{bmatrix} \quad (15)$$

とおき、

$$\mathbf{a}_{i+1} = \mathbf{a}_i - \left(\frac{\partial f}{\partial \mathbf{a}} \Big|_{\mathbf{a}=\mathbf{a}_i} \right)^{-1} f(\mathbf{a}_i) \quad (16)$$

としてパラメータを更新する。

5 距離画像の重なり方の判定

最小化によって得られた変換パラメータが、真の変換パラメータ \hat{T} であるのか、そうでない可能性があるのかを見分けるために、求めた変換パラメータ \hat{T} のまわりで、距離画像の重なり方、すなわち式(3)で示した誤差関数 J の値を調べる。

真の変換パラメータでない原因は次の2つが考えられる。

1. 変換パラメータの初期値が良いものでなかったために、ローカルミニマムに収束してしまった場合。(図3(a))
2. 重なっている部分が、例えば球面や平面のようななめらかな形状であるとき、最小値になる変換パラメータが真の変換パラメータ以外にも(多数)ある場合。(図3(b))

ここでは、 J の値の変化から2つを区別して、それぞれに対して別の処理をする。

具体的には、変換パラメータ \mathbf{q} と \mathbf{t} をそれぞれ $\mathbf{q} \pm \delta \mathbf{q}$, $\mathbf{t} \pm \delta \mathbf{t}$ の範囲で刻み幅 Δq_i , Δt_i づつ6つのパラメータを独立に変化させ、 J の値(距離の平均)と、 $M' = \sum_{m=1}^M p(\mathbf{x})$ (重なっている点の数)を評価する。

$$J_0 = \frac{1}{\sum_{m=1}^M p(\hat{T}(\mathbf{x}_m))} \sum_{m=1}^M p(\hat{T}(\mathbf{x}_m)) D_k(T(\mathbf{x}_m)) \quad (17)$$

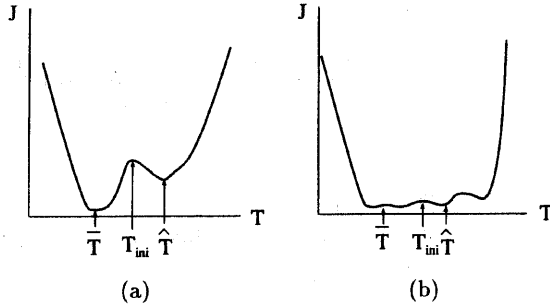


図3. 真の変換パラメータが得られないときのJの様子。(a) 変換パラメータの初期値がよいものでなかったためにローカルミニマムに収束した場合。(b) 重なっている部分がなめらかな形状であるとき、真の変換パラメータ以外にも最小値をとる変換パラメータが存在し、見分けがつかない場合。

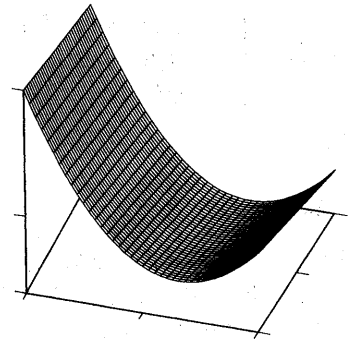


図4. 実験に用いた合成データ。なめらかな曲面。

$$M'_0 = \sum_{m=1}^M p(\hat{T}(x_m)) \quad (18)$$

とおく。

1. $J(T') < J_0$ かつ $M'(T') > M'_0$ であるような T' が存在したら、1の場合であると考えられるので、 T' を初期値として位置合せをしない。
2. $J(T') - J_0 < J_t$ かつ $M'(T') - |M'_0 - M_t|$ であるような T' が存在したら2の場合であると考えられるので、そのデータは重ね合わせしないで、重ね合わせる距離画像のリストの最後に加える。

6 実験結果

6.1 収束の様子

図4、図5に示した曲面の重ね合わせを行ったときの、収束の様子を図6に示す。いずれも、 40×40 の合成した距離画像である。実線が図4、破線が図5の例である。

図6を見ると分かるように、どちらの例でも、数回の繰り返して収束している。

6.2 計算時間の比較

距離関数を求めて式(3)を最小にする本研究の手法と、点の対応の探索を3-d tree[3][7]を用いて

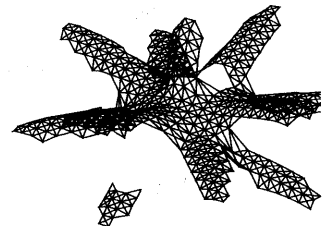


図5. 実験に用いた合成データ。複雑な形状のもの。

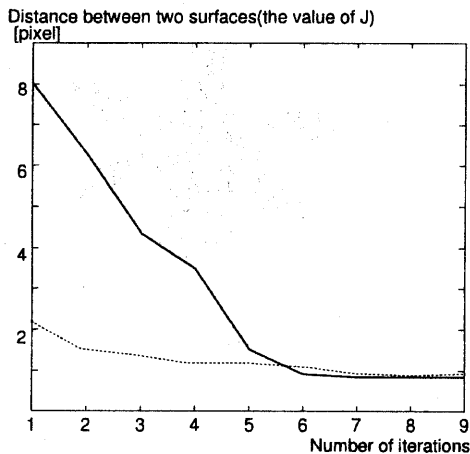


図 6. 収束の様子

探索する手法との計算時間の比較を行った結果を示す。両方とも式(3)で示した J を最小にした。

図4の重ね合わせを行った。データ数は1600点ずつである。

最小化の繰り返しに関係ない前処理として、本手法では距離関数を計算し、3-d treeを用いる場合はtreeを構成する。

最小化の各繰り返しでは、本手法では式(16)によりパラメータを更新する。一方、3-d treeの手法では、各点に対する対応点を、距離を計算しながらtreeを探索して求め、回転、並進のパラメータを計算する。

また、得られた運動パラメータの回りの重なり方を調べるためには、本手法では $D_k(x)$ の値を足すだけであるが、3-d treeを用いる場合は、距離を計算しながら各点の対応点を探索する。

それぞれUltra sparcで実行し、timeコマンドにより測定したものである。

表 1. 実行時間の比較

単位 [秒]	本手法	3-d tree
前処理	67.1	0.3
各繰り返し (10 回)	8.8	7.1
J の計算 (20 回)	0.16	339.5

表1を見て分かるように、前処理は本手法の方が遅い。各繰り返しでも、若干本手法の方が遅い。これは、式(14)の偏微分の計算が複雑で、各点に対して距離関数の勾配などを求めているためであ

と思われる。しかし、重なり方を調べるための計算は、3-d treeの方法に比べ、本手法では非常に高速である。

6.3 重なり方の判定

第5章で述べたように、最小化によって得られた変換パラメータが、真の変換パラメータ T であるのか、そうでない可能性があるのかを見分けるために、求めた変換パラメータ T のまわりで、距離画像の重なり方を調べた結果を示す。

図7に示した曲面(b)を曲面(a)に重ね合わせたとき、(b)を最小値のまわりで x 軸方向と y 軸方向に動かしたときの

J の値を図8に示す。両方の曲面はそれぞれ6400点から成る。

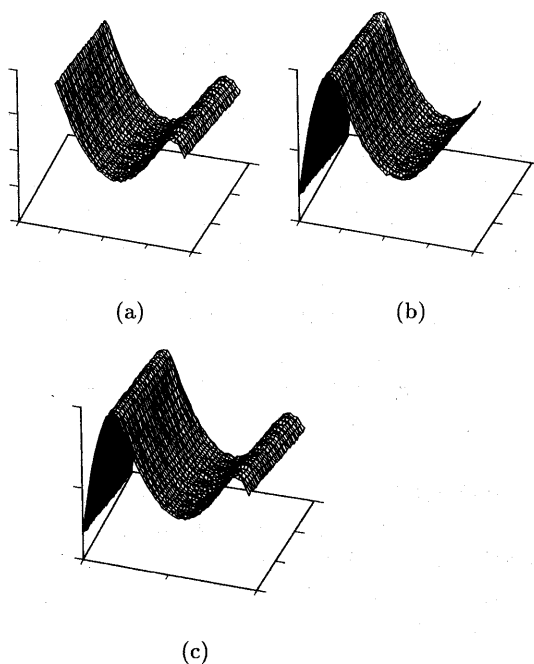


図 7. 重ね合わせた曲面. (a) 基準となる曲面. (b) 重ね合わせた曲面. (c) 重ね合わせた結果.

図8を見ると、 y 軸方向に動かしてもあまり J の値が変化していない様子が分かる。

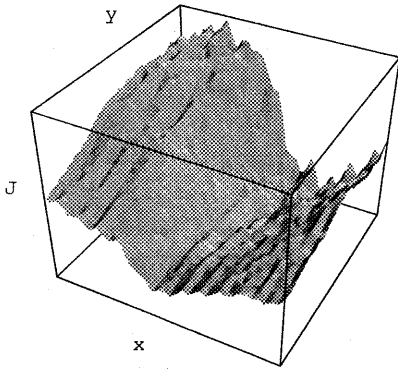


図 8. 図 7 の (a)(b) に示した二つの曲面の重なり方の様子. (b) の曲面を x 軸方向, y 軸方向にそれぞれ微量ずつ動かし, J をプロットしたもの.

6.4 重ね合わせ結果

図 5 に示した複雑な形状を 11 の視点から測定したものを重ね合わせた結果を図 9 に示す.

7 まとめと今後の課題

あらかじめおよその位置関係が分かっている場合の, 複数の距離画像を重ね合わせる手法を示した.

あらかじめ距離関数を求めておくことにより, 点の対応の探索の必要がなくなるため, 得られた運動パラメータのまわりでの距離画像の重なり方を簡単に調べることができる. これにより, 得られた変換パラメータが真の変換パラメータであるか否かを判定することができる.

また, 距離関数を求めるときに対応点が端点かどうかのラベルを付けておくことにより, 誤対応を減らすことができる.

距離画像を重ね合わせたものの距離関数を計算するとき, 多数の視点からの距離画像を, ここでは区別なしに基準点に加えて行ったが, これらを有効に用いて測定誤差の影響を減らすことを今後考える必要がある.

参考文献

- [1] 井口 征士, 佐藤 宏介. 三次元画像計測. 昭晃堂, 1990.
- [2] Y.Chen and G.Medioni. Object Modeling by Registration of Multiple Range Images. In *Proc.*

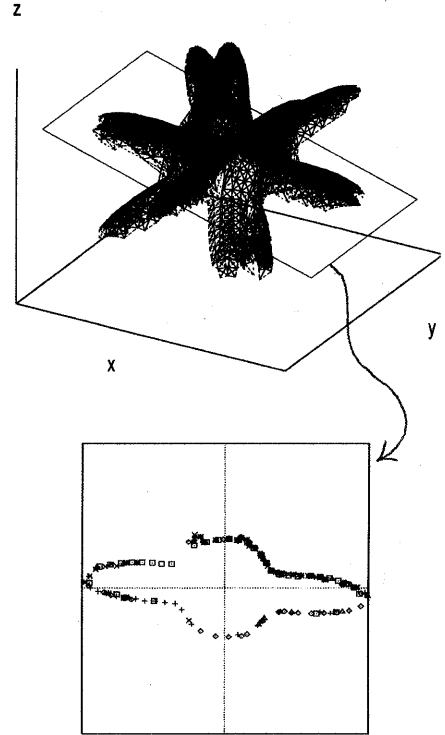


図 9. 11 の視点から観測したものを重ね合わせた結果 (上). ある平面で切ったもの (下). 各視点ごとに異なる印で表示し

IEEE Int. Conf. on Robotics and Automation, pp. 2724-2729, 1991.

- [3] Z.Zhang. Iterative Point Matching for Registration of Free-Form Curves. Technical Report 1658, INRIA, Sophia Antipolis B.P.109 06561 Valbonne Cedex France, March 1992.
- [4] T.Masuda and N.Yokoya. A Robust Method for Registration and Segmentation of Multiple Range Images. *CVIU*, Vol. 61, No. 3, pp. 295-397, 1995.
- [5] P.J.Besl and N.D.McKay. A Method for Registration 3-D Shapes. *IEEE Trans. PAMI*, Vol. 14, No. 2, pp. 239-256, 1992.
- [6] R.Kimmel and J.A.Sethian. Fast Marching Methods for Computing Distance Maps and Shortest Paths. Technical Report LBL-38451, University of California, Lawrence Berkeley Laboratory, Berkeley, CA 94720, February 1996.
- [7] D.A.Simon, M.Hebert and T.Kanade. Real-time 3-D Pose Estimation Using a High-Speed Range Sensor. Technical Report CMU-RI-TR-93-24, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, November 1993.