

屋内環境における平面对応付けによる 複数距離画像の同時位置あわせ

藤本 敬介^{†1} 守屋 俊夫^{†2} 中山 泰一^{†1}

異なる位置で計測した複数の距離画像の統合は、各データ間で同じ部位を対応付け、対応間を合わせ込むことによって行われる。点对応を用いる多くの従来手法は、対象が大規模になると対応数が膨大になるために計算が困難となる。本研究では、複数の点を平面としてまとめることで、対応数を大幅に減らし、高速かつ省メモリでの実行を可能にした。平面を合わせこむために、対応毎に決められた位置からの平面への垂線の足をを用いた平面間の距離の定義、及び Marquardt 法を用いた収束計算法を提案する。

Registration of Multiple Range Images for Indoor Environment

KEISUKE FUJIMOTO,^{†1} TOSHIO MORIYA^{†2}
and YASUICHI NAKAYAMA^{†1}

In this paper, we describe the plane-to-plane registration method for multiple range images used in the registration. This method generates pairs of corresponding planes and repeatedly minimizes an error metric of the distance between the planes by Levenberg-Marquardt minimization. The plane-to-plane distance is defined by the distance between the feet of a perpendicular line.

^{†1} 電気通信大学 情報工学科
Department of Computer Science, The University of Electro-Communications
^{†2} 日立製作所基礎研究所
Hitachi Ltd.

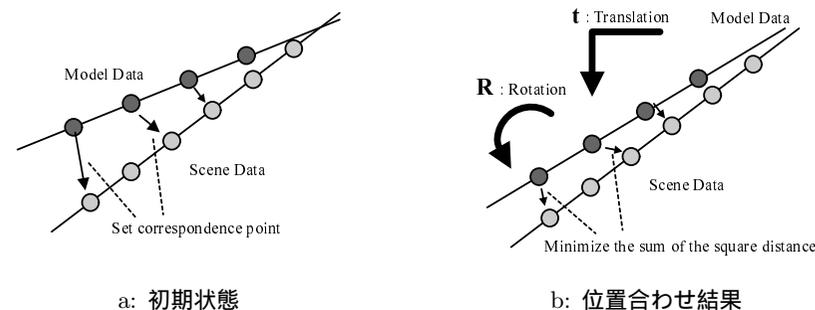


図1 点对応手法による位置合わせ

1. はじめに

本稿では、複数位置から計測した点群形式のデータセットに対して、平面特徴部を利用した統合手法を提案する。複数の点群データセットの統合手法に関する研究は多数存在する¹¹⁾。その中でも特に、Besl らの Iterative Closest Point (ICP) アルゴリズム³⁾ が有名である。最も基本的な ICP アルゴリズムは、データセット間の対応点の探索と対応点間の 2 乗距離和の最小化を、繰り返し実行することにより位置合わせを行う。また、ICP アルゴリズムの類似・改良手法も多数存在する¹⁰⁾。

ICP アルゴリズムをはじめとした、点群として得られる距離データからの位置合わせ手法の多くは、点と点 (point-to-point) もしくは点と平面 (point-to-plane) の対応を利用する。point-to-point の手法は図1に示すように、位置合わせによって、対応間の距離を最小にする回転・並進から成る変換パラメータを推定する。そのため、同じ部位の対応を厳密に取ることができれば精度の良い位置合わせが可能である。point-to-point の位置合わせの問題として、センサの分解能は有限であるため、厳密には正しい対応点が存在しないという点がある。そこで、一般的には精度面から point-to-plane の手法が用いられる。point-to-plane の手法は、片方のセンサの点に対し、もう片方の対応付けられた点の接平面に投影した位置を対応点とする。局所的に良い平面近似を与えることができる場合、point-to-plane の手法は point-to-point の手法よりも高精度となる。

もしもデータセットの計測位置の数が増えた場合、ICP アルゴリズムなどの 2 つのデータセットのための位置合わせを逐次的に用いると、位置合わせの誤差が蓄積されるという

問題がある。特に空間が大規模になるほど、この問題は顕著になる。そのため、大規模な空間の構築の際には、誤差の蓄積しない位置合わせ手法が必要となる。Chen ら⁴⁾、Masuda ら⁶⁾ はそれぞれ、逐次位置合わせ手法による複数データセットのための位置合わせ手法を提案した。Stoddart ら¹²⁾ は全データセット間での点対応を求め、繰り返し計算により対応間の距離を最小化することによる位置合わせ手法を提案した。Benjamaa ら¹⁾ は Bergevin らの手法²⁾ を改良し、z バッファを用いた同時位置合わせ手法を提案した。Neugebauer ら⁷⁾ は point-to-plane の対応付けにより、誤差の蓄積しない同時位置合わせ手法を提案した。Pulli⁹⁾ は、逐次位置合わせ手法に対し、後ろ向きにも位置合わせを行うことで、誤差を減らした。

これらに共通する問題は、point-to-point や point-to-plane による位置合わせを基本にするために、対応点探索のために多くの計算時間が必要であることである。点の数を N とした場合、全部の対応点を単純に用いた場合、対応点探索の計算量は $O(N^2)$ となる。しかしながら、データセットの数 M が 3 以上になる場合、同時にこれらを位置合わせすることは非常に難しくなり、計算コストは $O(M^2 N^2)$ となる。更に、全データセットをメモリ上に蓄える必要があるため、大きなメモリが必要となる。

対応数を減らすには、点群を直接扱わず、特徴的な部位どうしの対応を用いる手法が考えられる。そこで、特徴部として平面部を直接合わせこむことができれば、複数の点をひとつの平面にまとめることで、対応数を減らすことができる。

本稿では、複数の大規模なデータセットからの平面と平面 (plane-to-plane) の対応を用いた同時位置合わせ手法を提案する。複数の点を平面にまとめることで、点対応を用いる手法に比べ対応の数が少ないため高速な実行が可能となる。更に、平面パラメータのみを用いるため、少ないメモリで実行することができる。位置合わせに平面の対応を用いるために、平面間の距離を定義する。通常、平面間の距離は、平面が平行以外の場合に定義することができないが、本章では垂足 (垂線の足) を用いて平面間の距離を定義する。本手法は、入力した複数の点群のセットから検出した平面に対し、データセット間での平面どうしの対応付けと、平面間の距離の最小化する変換パラメータの推定を繰り返すことで、位置合わせを行う。

2. 平面間の距離の定義

本手法は、各平面間の距離を最小にする回転と並進から成る変換パラメータを求めることによって、複数の点群データセットを統合する。平面对応の位置合わせのためには、まず平面間の距離について定義をする必要がある。そこで、平面間の距離として、空間上の任意の

点 (“参照点” と呼ぶ) から平面上に下ろした垂足間の距離を用いる。本節では、まず垂足を用いた理由について述べる。続いて平面間の距離について述べ、データに回転と並進を与えたときの距離を導出する。

2.1 垂線の足の使用

通常、平面は c と n の 2 つのパラメータによって定義される。 c は平面上に置かれた任意の点 (“中心点” と呼ぶ)、 n は平面の法線ベクトルである。以後、この 2 つのパラメータによって定められる平面 H を (c, n) とする。

従来手法のように、中心点と平面の距離を 2 平面間の距離とした場合、その距離は、それぞれの平面の中心点の位置関係に影響される。例えば、それぞれの平面上に置かれた中心点と同じ位置にあった場合、平面が平行でない場合でも、距離は零になってしまう。また、2 平面間の角度が同じ場合でも、中心点 c によって距離は変化する。これは、中心点 c が平面上の任意の位置に設定できることに起因する。

それに対し、平面に対して、ある点から下ろした垂足の位置は一意に決まる。そこで本手法では、平面間の距離として、垂足間の距離を用いることとする。それにより、もしひとつの無限平面上に両方の平面が乗るときは、 c の座標によらず垂足は同じ位置となる。 c が同じ位置にあったとしても、異なる平面であれば垂足も異なる位置となる。

2.2 平面間の距離の定義

本節では、平面間の距離の定義について述べる。 $\langle \cdot, \cdot \rangle$ をベクトル間の内積と定義すると、中心点 c と法線 n を持つ平面 H は

$$H = \{x | \langle n, x - c \rangle = 0\} \tag{1}$$

となる。参照点 q から平面 H に下ろした垂足 $s(q, H)$ は以下となる。

$$s(q, H) = q - \langle n, q - c \rangle n \tag{2}$$

本手法では、平面間の距離を、参照点 q からそれぞれの平面に下ろした垂足間の距離として定義する。すなわち、参照点 q を用いた、2 つの平面 H_1, H_2 間の距離 $d_q(H_1, H_2)$ を次のように定義する。

$$\begin{aligned} d_q(H_1, H_2) &= \|s(q, H_1) - s(q, H_2)\| \\ &= \|\langle n_1, q - c_1 \rangle n_1 - \langle n_2, q - c_2 \rangle n_2\| \end{aligned} \tag{3}$$

ここで、 H_1 は (c_1, n_1) 、 H_2 は (c_2, n_2) の平面とする。このように定義した距離が、距離の公理

$$d_q(H_1, H_2) + d_q(H_2, H_3) \geq d_q(H_1, H_3) \tag{4}$$

を満たすことは容易に確認できる。参照点 q の位置によって、 $d_q(H_1, H_2)$ の値は変化する

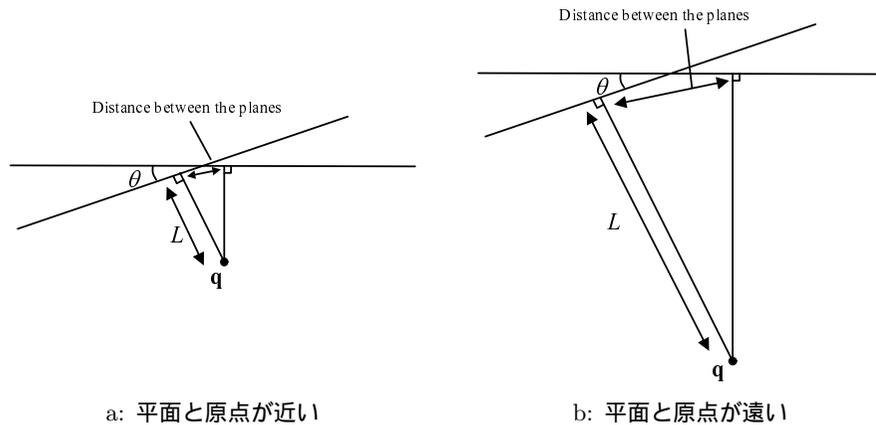


図 2 点对应手法による位置合わせ

ため、 q を固定することで平面間の距離を一意に定める必要がある。そこで、どこに参照点 q を設定するかが問題となる。図 2 に、平面間の幾何的な位置関係が同じ場合における、参照点からの距離が異なる場合の比較を示す。図 2a は参照点からの距離が、図 2b より平面の対に近い状態である。2 つの平面の関係が同じであるにも関わらず、図 2a における平面間の距離は図 2b における距離よりも小さい。このように、式 (3) の距離は、参照点 q から平面までの距離に応じて変化する。そこで、本手法では平面から一定の距離に参照点を設定する。

参照点は、片方の平面を基準として、中心点 c_1 から法線ベクトル n_1 方向に、パラメータ L だけ伸ばした位置を用いる (図 3)。そのとき参照点の位置は以下となる。

$$q = c_1 + Ln_1 \quad (5)$$

パラメータとして与えた L は、法線ベクトルの誤差に応じて決定する。図 2 に示すように、2 つの平面が平行でない場合、 L が大きいほど垂足間の距離は大きくなる。つまり、 L が大きいほど、法線ベクトルの誤差が位置合わせの誤差に影響する。よって、 L は法線ベクトルの誤差に応じて設定する。本手法では、 L の値は経験的に与えることとする。

2.3 回転・並進後の平面間の距離

位置合わせは、各データに対して回転と並進を与えることで、データに含まれる平面を合わせこむものである。そこで、それぞれの平面に回転と並進を与えたうえで、平面間の距

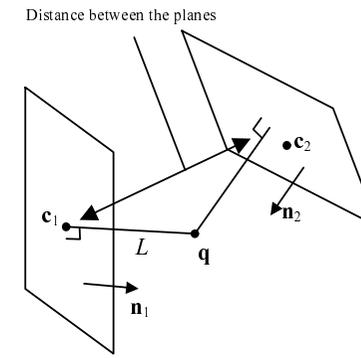


図 3 垂線の足を用いた平面間の距離

離を求める。回転行列 R と並進行列 t によって定められる変換量を T とする。また、 T による変換を $T(\cdot)$ とする。

平面 H に変換 T を与えたときの、変換後の平面は $T(H)$ と表され、中心点は $Rc + t$ 、法線ベクトルは Rn となる。よって、参照点 q から変換後の平面 $T(H)$ に下ろした垂線の足 $s(q, T(H))$ は以下となる。

$$s(q, T(H)) = q - \langle Rn, q - Rc - t \rangle Rn \quad (6)$$

これより、2 つの平面 H_1, H_2 に対し、それぞれ変換 T_1, T_2 を与えると、平面間の距離は以下となる。

$$\begin{aligned} d_q(T_1(H_1), T_2(H_2)) &= \|s(q, T_1(H_1)) - s(q, T_2(H_2))\| \\ &= \|\langle R_1 n_1, q - R_1 c_1 - t_1 \rangle R_1 n_1 \\ &\quad - \langle R_2 n_2, q - R_2 c_2 - t_2 \rangle R_2 n_2\| \end{aligned} \quad (7)$$

ここで、 H_1 は (c_1, n_1) 、 H_2 は (c_2, n_2) の平面とする。 T_1 は (R_1, t_1) 、 T_2 は (R_2, t_2) の変換とする。

ただし、参照点については変換を与えず、式 (5) で求めた参照点 $q = c_1 + Ln_1$ を用いる。この理由は、後述する最小化中に参照点が動いてしまうと、距離の基準も変化してしまうからである。

3. Marquardt 法による最小化

本節では、収束計算を用いた複数平面間の距離の最小化について述べる。各データセット

は、それぞれ複数の平面を持っているものとする。位置合わせのためには、データ間で対応付けられた平面間の距離を最小にする回転と並進から成る変換パラメータを求める必要がある。しかしながら、平面間の距離の評価式は解析的に解くことができない。そこで、勾配ベクトルとヘッセ行列を算出し、Marquardt 法によって最小化する。

3.1 変換パラメータの定義

はじめに回転量、並進量に関する変換パラメータを定義する。データセット数を M とし、データセット $i \in \{1, \dots, M\}$ の変換を $T^{(i)}$ とする。 $T^{(i)}$ は $\mathbf{R}^{(i)}$ と $\mathbf{t}^{(i)}$ から成る。回転について、それぞれ $p_1^{(i)}$ を X 軸に関する回転量、 $p_2^{(i)}$ を Y 軸に関する回転量、 $p_3^{(i)}$ を Z 軸に関する回転量とすると、回転行列 $\mathbf{R}^{(i)}$ は以下となる。

$$\mathbf{R}^{(i)} = \mathbf{R}(Z, p_3^{(i)})\mathbf{R}(Y, p_2^{(i)})\mathbf{R}(X, p_1^{(i)}) \quad (8)$$

$$\mathbf{R}(X, p_1^{(i)}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(p_1^{(i)}) & -\sin(p_1^{(i)}) \\ 0 & \sin(p_1^{(i)}) & \cos(p_1^{(i)}) \end{pmatrix} \quad (9)$$

$$\mathbf{R}(Y, p_2^{(i)}) = \begin{pmatrix} \cos(p_2^{(i)}) & 0 & \sin(p_2^{(i)}) \\ 0 & 1 & 0 \\ -\sin(p_2^{(i)}) & 0 & \cos(p_2^{(i)}) \end{pmatrix} \quad (10)$$

$$\mathbf{R}(Z, p_3^{(i)}) = \begin{pmatrix} \cos(p_3^{(i)}) & -\sin(p_3^{(i)}) & 0 \\ \sin(p_3^{(i)}) & \cos(p_3^{(i)}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (11)$$

また並進行列 $\mathbf{t}^{(i)}$ は

$$\mathbf{t}^{(i)} = \begin{pmatrix} p_4^{(i)} & p_5^{(i)} & p_6^{(i)} \end{pmatrix}^T \quad (12)$$

となる。データセット i の回転と並進に関する変換パラメータをひとつのベクトル $\mathbf{p}^{(i)}$ にまとめ、以下のように表す。

$$\mathbf{p}^{(i)} = \begin{pmatrix} p_1^{(i)} & p_2^{(i)} & p_3^{(i)} & p_4^{(i)} & p_5^{(i)} & p_6^{(i)} \end{pmatrix}^T \quad (13)$$

すべてのデータセットの変換パラメータ \mathbf{p} は $6M$ 次元のベクトルとなり、以下のように

表す。

$$\mathbf{p} = \begin{pmatrix} \mathbf{p}^{(1)T} & \dots & \mathbf{p}^{(M)T} \end{pmatrix}^T \quad (14)$$

3.2 複数のデータセット間における評価値

複数のデータセット間の評価値を定義する。評価値は、データセット間で対応付けられた平面間の距離の 2 乗和とする。各データセットは、それぞれが複数の平面を持ち、各データセット間の平面が対応付けられる。すべてのデータセット間で、計 J 個の平面の対があるとする（対応付けに関しては第 4 節で述べる）。対応のインデックスを $k \in \{1, \dots, J\}$ とする。平面の対 k を含むデータセット i の平面を $H_k^{(i)}$ とする。図 4 に、データセット数が 3、対応数が 3 の例を示す。ここで、平面の対 k を含む 2 つのデータセット $i, j \in \{1, \dots, M\}$ の組を考える。この平面の対の参照点は

$$\mathbf{q}_k^{(i)} = \mathbf{c}_k^{(i)} + L\mathbf{n}_k^{(i)} \quad (15)$$

となる。ここで $\mathbf{q}_k^{(i)}$ をデータセット i に含まれる平面の対 k についての参照点とする。これらの関係を図 5 に示す。 $H_k^{(i)}$ のパラメータを $\mathbf{c}_k^{(i)}, \mathbf{n}_k^{(i)}$ とし、データセット i に含まれる平面の対 k の中心点、法線ベクトルとする。

データセット i の変換パラメータ \mathbf{p}_i による変換を $T^{(i)}$ とする。回転・並進後の複数データセット間での評価値を、各平面の対の距離の 2 乗和として、以下のように定義する。

$$\begin{aligned} E(\mathbf{p}) &= \sum_{i \neq j, k} d_{\mathbf{q}_k^{(i)}}^2 \left(T^{(i)}(H_k^{(i)}), T^{(j)}(H_k^{(j)}) \right) \\ &= \sum_{i \neq j, k} \left\| \left\langle \mathbf{R}^{(i)} \mathbf{n}_k^{(i)}, \mathbf{q}_k^{(i)} - \mathbf{R}^{(i)} \mathbf{c}_k^{(i)} - \mathbf{t}^{(i)} \right\rangle \mathbf{R}^{(i)} \mathbf{n}_k^{(i)} \right. \\ &\quad \left. - \left\langle \mathbf{R}^{(j)} \mathbf{n}_k^{(j)}, \mathbf{q}_k^{(i)} - \mathbf{R}^{(j)} \mathbf{c}_k^{(j)} - \mathbf{t}^{(j)} \right\rangle \mathbf{R}^{(j)} \mathbf{n}_k^{(j)} \right\|^2 \end{aligned} \quad (16)$$

3.3 評価値の最小化

提案する位置合わせ法は、平面間の距離を最小化する変換パラメータを求めるものである。すなわち、以下を満たすような、評価式 (16) を最小にする変換パラメータ \mathbf{p} を求める。

$$\min_{\mathbf{p}} \sum_{i \neq j, k} d_{\mathbf{q}_k^{(i)}}^2 \left(T^{(i)}(H_k^{(i)}), T^{(j)}(H_k^{(j)}) \right) \quad (17)$$

式 (16) は非線形であるために、解析的に最小化できない。そこで、収束計算によってパラメータを求める。収束計算の手法としては、最急降下法やニュートン法、Marquardt 法など

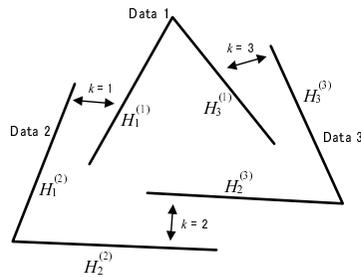


図 4 複数データセット間の対応関係

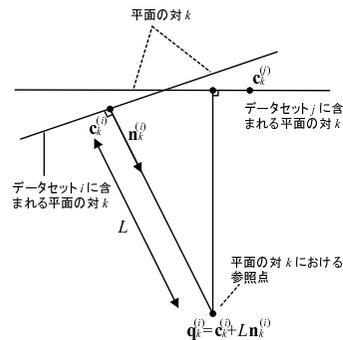


図 5 平面と参照点の距離を L とした場合の平面間の距離

がよく用いられる。これらの収束計算法は、初期条件が最適解に十分近い場合に、最適解に収束する。Marquardt 法は微分が可能である場合に高速に実行することが可能である。本手法では、高速かつ精度良い結果となる Marquardt 法を使用する。本節では、Marquardt 法に用いる、勾配ベクトルとヘッセ行列を計算し、評価値を最小にする p を求める。

データセット i の変換パラメータに関する勾配ベクトルを $g^{(i)}$ 、ヘッセ行列を $H^{(i)}$ として、Marquardt 法を用いてデータセット i の変換パラメータ $p^{(i)}$ を以下の式により更新する。

$$(H^{(i)} + \lambda I)\Delta p^{(i)} = -g^{(i)} \quad (18)$$

ここで $\Delta p^{(i)}$ はデータセット i の変換パラメータの更新量である。パラメータ λ は $H^{(i)} + \lambda I$ が正則となるように選ばれる。最小化のアルゴリズムの詳細は文献⁵⁾などを参照されたい。

4. 位置合わせ手法

本節では、点群からの平面を用いた位置合わせ手順について述べる。本手法は各データセット間で対応付けられた平面間の距離を最小化することによって位置合わせを行う。本手法の手順は以下とする。

- (1) 平面検出
- (2) 対応付け
- (3) 最小化

手順 1. において、各データセットに対して BSP 木を用いて平面クラスタを検出する。手順 2. において、データセット間で、近傍に位置する平面クラスタどうしを探索し、対応付けを行う。手順 3. において、すべての平面の対に対して、同時に最小化する。本手法は ICP アルゴリズムと同様に、手順 2. の対応付けと手順 3. の最小化を繰り返すことで、位置合わせを行う。

本手法は平面を用いた位置合わせであるため、各データセットは複数の平面を持つ必要がある。例えば屋内などの人工的な環境に対して、適用することができる。

4.1 平面検出

まず、点群を複数の平面クラスタに分割する。平面部を直接得ることはできないため、点群から得られる情報を元に平面を推定する。

本手法では BSP 木をベースとした階層クラスタ手法⁸⁾を用いる。木の各ノードが点群のサブセット $P = \{x_1, \dots, x_n\}$ から成るとする。点群 P から得られる共分散行列 C を用いて、ノードを分割するかどうかを決定する。以下に C を示す。

$$C = \begin{bmatrix} x_1 - \bar{x} \\ x_2 - \bar{x} \\ \vdots \\ x_n - \bar{x} \end{bmatrix}^T \begin{bmatrix} x_1 - \bar{x} \\ x_2 - \bar{x} \\ \vdots \\ x_n - \bar{x} \end{bmatrix}, \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (19)$$

C は 3×3 の対称の半正定値行列であるため、行列の 3 つの固有値は非負となる。ここで、固有値を $\lambda_1 \leq \lambda_2 \leq \lambda_3$ で表す。各固有値はそれぞれに対応する固有ベクトル v_1, v_2, v_3 の方向における分散となる。

最小の固有値 λ_1 に対応する固有ベクトル v_1 は、点群サブセット P を平面近似した場合の法線ベクトルとなり、固有値 λ_1 がその大きさととなる。そのため λ_1 が他の固有値に対し

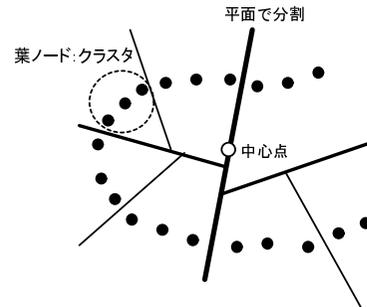


図 6 階層構造を用いた点群の分割

て相対的に小さい場合に平面であるといえる． λ_1 の割合 σ は以下のように求めることができる．

$$\sigma = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \quad (20)$$

点群サブセット P が平面に近いほど σ は小さくなる．

各固有値の大きさは，対応する固有ベクトル方向の大きさとなる．そこで，クラスタ全体の大きさを以下として決定する．

$$\tau = \frac{\lambda_1 + \lambda_2 + \lambda_3}{n} \quad (21)$$

本手法では，この平面の度合い σ とクラスタの大きさ τ を基準にして，再分割するかどうかを決定する．各ノードは以下の条件のどちらかを満たすときに再分割される．

- σ と τ が設定された閾値よりも大きい．
- サブセット P の点数が一定数以上である．

最も大きい固有値に対応する固有ベクトルを法線とする平面を H とする．ノードの持つ点群サブセット P を，平面 H を用いて再分割する．平面に対し同じ側にある点を子のサブセットとし，2つのサブセットとなる．このように階層的な BSP 木の葉を用い，図 6 のようにクラスタを作成する．

4.2 平面クラスタの対応付け

続いて，前節で作成した平面クラスタの対応付けを行う．本手法では対応付けは2つの条件に基づく．まず，2平面の中心点間のユークリッド距離が閾値より小さいことを条件とする．もうひとつは，本手法で定義した平面間の距離が最も近いことを条件とする．

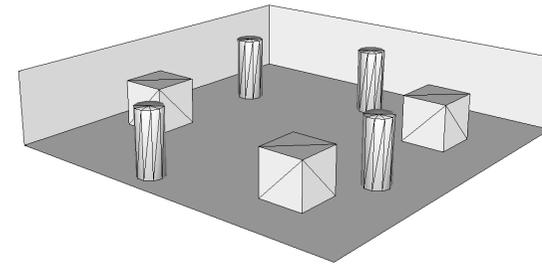


図 7 シミュレーション環境

平面の対における中心点間のユークリッド距離が大きくなると，異なる平面である可能性が高くなる．もし同じ平面であったとしても，中心点間の距離に応じて，法線ベクトルの推定誤差が位置合わせに与える誤差も大きくなる．よって，対の設定のためには中心点間が近い必要がある．また，2つの平面クラスタは，同じ平面上に存在する必要があるために，平面間の距離が最も近いものを選ぶこととする．各平面クラスタは多数の点から構成されているため，点对応の手法に比べて，平面对応手法を用いると対の数が少なくなる．そのため，高速に対対応付けを行うことができる．

5. 実 験

本節では，シミュレーション環境と実環境において，それぞれ実験する．シミュレーションに用いた環境は，広さが 100m^2 となる平面が多く存在する部屋とする．図 7 に，その環境を示す．計測データは，垂直方向に線状に取得できる距離センサを水平に回転させることとして，レイトレーシングによって得る．計測したデータには，標準偏差が 0.01m となる正規分布に従いノイズを与える．まず，シミュレーション環境において計測した 30 箇所からのデータセットの同時位置合わせを行う．point-to-plane による手法に対し，位置合わせ後のデータセットの位置誤差と法線誤差について比較する．続いて，実データに対する実験として，8 個の位置から得た大規模なデータの同時位置合わせを行う．

5.1 シミュレーション環境での同時位置合わせ

続いて，多数のデータセットからの同時位置合わせについて実験する．実験に利用した計算機は CPU が Intel Core i7 CPU 2.67GHz (2CPU)，主記憶容量が 6GB の PC，OS は

Windows Vista である。前節で用いたシミュレーション環境において、30 のランダムな位置からデータを取得した。各データセットの初期位置としては、正しい初期位置から $\pm 0.1\text{m}$ 、 ± 10 度の範囲で一様分布の乱数を与えた。図 8 にシミュレーション環境での実験結果を示す。図 8a に初期状態を示し、図 8b に位置合わせ結果を示す。表 1 に、視点数、合計の取得点数、検出した平面、必要メモリ、計算時間、精度について示す。

本手法の結果として、合計の回転誤差量は 2.17×10^{-5} となり、並進誤差は 2.13×10^{-3} となった。収束に必要な回数は、10 回であった。合計の計算時間は 6.1 秒となった。

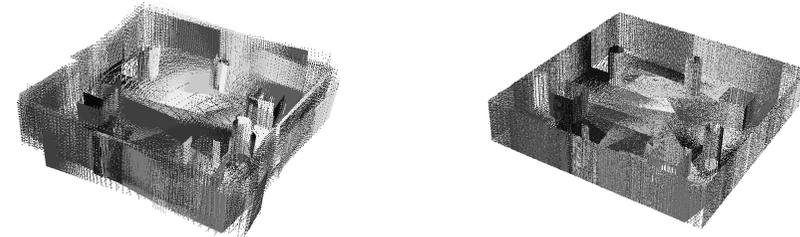
比較手法として Chen ら⁴⁾ の逐次位置合わせの手法を用いる。Chen ら手法は逐次的な位置合わせであるために、高い精度が得られない反面、各繰り返しにおける対応の探索数少なく、同時位置合わせ手法に比べて高速である。実験の結果、位置合わせによる回転誤差量は 4.48×10^{-4} 、並進誤差は 1.03×10^{-1} となった。また、各データセットの平均の収束回数は 94 回、合計の計算時間は 129.3 秒となった。

本手法は、必要メモリが非常に少なく、高速な実行が可能であった。これは、本手法では対応を作成する際に平面クラスタのみが必要であり、比較手法で対応付けに利用する点の数が 100 万以上であるのに対し、本手法で対応付けに利用する平面の数は約 2,000 と、非常に少ないためである。

5.2 実環境での位置合わせ

続いて実環境において実験する。実験に利用した計算機は CPU が Intel Core 2 Duo CPU 2GHz (2CPU)、主記憶容量が 2GB の PC、OS は Windows XP である。実験結果を図 9 に示す。初期位置の状態を図 9a に、位置合わせ後の状態を図 9b に示す。ここでデータ取得に用いた計測装置は RIEGL 社の LMS-Z420i とする。このセンサの誤差の標準偏差は約 0.008m である。実験では、8 つの位置から取得したデータを用いる。合計の取得点数は 22,722,347 となった。また初期位置の誤差は並進誤差として $\pm 0.1\text{m}$ 、回転誤差として ± 10 度を与える。表 2 に実験結果を示す。本手法は、平面部以外も多く含む実環境に対しても適用可能であることが確認できる。

表 2 に視点数、合計の取得点数、検出した平面数、使用メモリ、計算時間を示す。結果として、2 千万点以上という点数の多さに関わらず、必要なメモリとして 200KB での実行が可能になった。この理由としては、検出した合計の平面数が約 4,000 と少数であったためと考えられる。



a: 初期状態

b: 位置合わせ結果

図 8 シミュレーション環境における実験

表 1 シミュレーション環境の同時位置合わせ結果

手法	提案手法	Chen
視点数	30	
点数	1,421,843	
検出した平面数	2,275	—
使用メモリ	150 [KB]	240 [MB]
実行時間	6.1 [s]	129.3[s]
収束回数	10	94
回転誤差	2.17×10^{-5}	4.48×10^{-4}
並進誤差	2.13×10^{-3}	1.03×10^{-1}

6. おわりに

本稿では、複数の大規模点群における plane-to-plane の位置合わせ手法を提案した。本手法では、平面間の距離を垂足間の距離として定義し、各データセット間で平面クラスタの対応付けと最小化を繰り返すことにより、位置合わせを行った。従来は点として対象を扱っていたのに対し、平面特徴の当てはめを用いることにより、高速かつ省メモリでの位置合わせが可能となった。本手法は平面の対応を用いるために、平面部を多く持つ屋内などの人工的な状況において適用することができる。

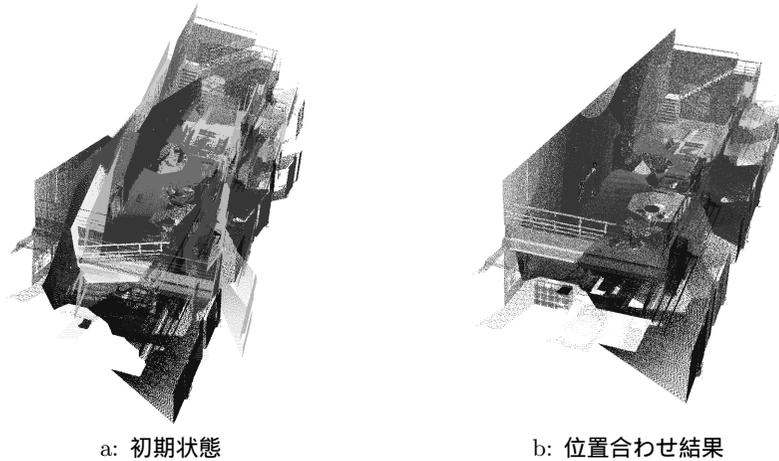


図9 実環境における実験
表2 実環境の位置合わせ結果

視点数	8
点数	22,722,347
検出した平面数	3,984
使用メモリ [KB]	200
実行時間 [s]	16.7
収束回数	13

参 考 文 献

- 1) Benjemaa, R. and Schmitt, F.: Fast Global Registration of 3D Sampled Surfaces using a Multi-z-buffer Technique, *Proc.Int'l. Conf. Recent Advances in 3D Digital Imaging and Modeling*, pp.113-120 (1997).
- 2) Bergevin, R., Soucy, M., Gagnon, H. and Laurendeau, D.: Towards a General Multi-View Registration Technique, *IEEE Trans.Pattern Analysis and Machine Intelligence*, Vol.18, No.5, pp.540-547 (1996).
- 3) Besl, P. J. and McKay, N. D.: A Method for Registration of 3-D Shapes, *IEEE Trans.Pattern Analysis and Machine Intelligence*, Vol.14, No.2, pp.239-256 (1992).

- 4) Chen, Y. and Medioni, G. G.: Object Modeling by Registration of Multiple Range Images, *Image and Vision Computing*, Vol.10, No.3, pp.145-155 (1992).
- 5) Dennis, J. E. and Schnabel, R. B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations, *SIAM Classics in Applied Mathematics*, (1996).
- 6) Masuda, T. and Yokoya, N.: A Robust Method for Registration and Segmentation of Multiple Range Images, *Computer Vision and Image Understanding*, Vol.61, No. 3, pp.295-307 (1995).
- 7) Neugebauer, P. J., Reconstruction of Real-World Objects via Simultaneous Registration and Robust Combination of Multiple Range Images, *Int'l. Journal of Shape Modeling*, Vol.3, No.1-2, pp.71-90 (1997).
- 8) Pauly, M., Gross, M. and Kobbelt, L.: Efficient Simplification of Point-Sampled Surfaces, *Proc.IEEE Conf. Visualization*, pp.163-170 (2002).
- 9) Pulli, K.: Multiview Registration for Large Data Sets, *Proc.Int'l. Conf. 3D Digital Imaging and Modeling*, pp.160-168 (1999).
- 10) Rusinkiewicz, S. and Levoy, M.: Efficient Variants of the ICP Algorithm, *Proc.3rd Int'l. Conf. 3D Imaging and Modeling*, pp.145-152 (2001).
- 11) Salvi, J., Matabosch, C., Fofi, D. and Forest, J.: A Review of Recent Range Image Registration Methods with Accuracy Evaluation, *Image Vision Computing*, Vol.25, No.5, pp.578-596 (2007).
- 12) Stoddart, A. J. and Hilton, A.: Registration of Multiple Point Sets, *Proc.Int'l. Conf. Pattern Recognition*, pp.40-44 (1996).