

分散メモリシステムにおける大規模距離画像の並列同時位置合わせ手法

大石 岳 史^{†1} 佐川 立 昌^{†3} 中澤 篤 志^{†2}
倉 爪 亮^{†4} 池 内 克 史^{†1}

本論文では複数方向から測定された多数の距離画像を並列に同時位置合わせする手法を提案する。複数距離画像の同時位置合わせ手法はこれまでに幾つか提案されているが、それらを大規模距離画像にそのまま適用することは計算時間及びメモリ使用量の問題のため現実的ではない。そこで我々は同時位置合わせ手法を並列化し、分散メモリシステム上で計算の高速化及びメモリ使用量の削減を可能にする手法を開発した。位置合わせ計算で最も計算コストが高いのは対応点探索及び誤差計算であるが、これらの計算は各距離画像間で独立に行う事ができる。そこで、これらの計算を各距離画像の組合せ毎にプロセッサに割当てる事によって並列処理を行った。このとき、位置合わせ計算に不要、冗長な組合せを除去する事により、計算時間の短縮及び効率的なデータ分散を図った。また負荷分散とデータ分散を同時に実現するためにはグラフ分割の手法を用いた。提案した手法の有効性を確認するために、16 プロセッサの PC クラスタを用いて大規模距離画像群に対して同時位置合わせ実験を行い、収束性能と位置合わせ精度、計算時間、メモリ使用量に関して評価した。

Parallel Simultaneous Alignment of a Large Number of Range Images on Distributed Memory System

TAKESHI OISHI,^{†1} RYUSUKE SAGAWA,^{†3} ATSUSHI NAKAZAWA,^{†2}
RYO KURAZUME^{†4} and KATSUSHI IKEUCHI^{†1}

This paper describes a method for parallel alignment of multiple range images. Since it is difficult to align a large number of range images simultaneously, we developed a parallel method to accelerate and reduce the memory requirement of the process. Although a general simultaneous alignment algorithm searches correspondences for all pairs of all range images, by rejecting redundant dependencies, our method makes it possible to accelerate computation time and reduce the amount of memory used on each node. The correspondence search is performed independently for each pair of range images. Accordingly, the computations between the pairs are preformed in parallel on multiple processors. All relations between range images are described as a pair node hyper-graph. Then, an optimal pair assignment is computed by partitioning the graph properly. The method was tested on a 16 processor PC cluster, where it demonstrated the high extendibility and the performance improvement in time and memory.

1. はじめに

近年、レーザレンジファインダの発達により、文化遺産や美術品の形状を 3 次元モデルとして保存する試み

が世界各地で行われている^{1)~4)}。これらの計測に用いられるレーザレンジファインダは、光切断法や Time-of-flight 方式などの技術によって、近距離では 1mm 以下の精度、また 50m から 100m 程度の長距離でも 1cm 以下という高い精度で物体表面の形状を測定することが可能となっている。

一方で、対象が大規模構造物になると、その全周 3 次元モデルを生成する事は難しくなる。レーザレンジファインダの測定可能箇所はレーザが届く範囲に限られるため、一回の測定では物体全体をモデル化することはできない。そのため、完全な全周 3 次元モデルを得るためには複数方向から多数の距離画像を取得しなければならない。この複数方向から取得された距離画

†1 東京大学 生産技術研究所
Institute of Industrial Science, The University of Tokyo

†2 大阪大学 サイバーメディアセンター
Cybermedia Center, Osaka University

†3 大阪大学 産業科学研究所
Institute of Scientific and Industrial Research,
Osaka University

†4 九州大学 大学院 システム情報科学研究院
Graduate School of Information Science and Electrical
Engineering, Kyushu University

像は、それぞれの測定位置に対する異なった座標系で記述されているため、各々の相対位置姿勢を求め座標系を統一する位置合わせ処理が必要となる。測定対象が比較的小さい場合は、ターンテーブルなどを用いてレンジファインダと対象物体の相対位置関係を記録し、各距離画像間の相対位置関係を求める事も可能である。しかし対象物体が大規模な場合には、この相対位置関係を正確に測定することは難しい。

この距離画像の位置合わせ問題を解決するために、これまでに多くの手法が提案されている。特によく知られている手法は Besl が提案した ICP (Iterative Closest Point) 法⁵⁾ や Chen の手法⁶⁾ である。ICP では 2 つの距離画像中の最近傍点を対応点として、この対応点間距離を最小化するように各距離画像の変換行列を求め、繰返し計算によって相対位置を求めていく。一方、Chen の手法では面の法線を計算し、点と面の距離を最小化するように相対姿勢を求めていく。この他にも視線方向に点を投影して他方の距離画像との対応点を求める方法などもある⁷⁾。また ICP は誤対応やノイズによる影響を受け易いため、ランダムサンプリングと LMedS (Least Median Squares Estimation) 法によってロバスト性を高める手法も提案されている⁸⁾。

これらの位置合わせ手法は 2 枚の距離画像の相対位置姿勢を求めるものであるが、距離画像枚数が多い場合はこれを順次繰返ししていくと誤差の蓄積が問題となる。これを避けるためには、全ての距離画像間の相対位置姿勢を同時に推定する同時位置合わせ手法が有効である。Neugebauer は視線方向に対応点を探索し、全ての距離画像の組合せにおける点と面の距離の二乗和を誤差として、最小二乗問題を線形化して解くことによって全体位置合わせを行う手法を提案している⁹⁾。Benjemaa¹⁰⁾ は法線方向による頂点の分割と z-buffer を使ってペアワイズな位置合わせを高速化するとともに、Bergevin の手法¹¹⁾ を拡張して重なり合う距離画像間で複数枚の同時位置合わせする手法を提案している。また Nishino は M 推定を用いたロバストな同時位置合わせ手法を提案している¹²⁾。

このように、これまでに様々な位置合わせ手法が提案されているが、いずれの手法でも問題となるのが対応点探索の計算量である。基本的な ICP では最近傍点を全ての頂点に対して計算するため、対応する 2 枚の距離画像の頂点数を等しく n とした場合、対応点探索の計算量は $O(n^2)$ である。そこで探索を高速化する手法としては探索木 (kd-tree) を使う手法や¹³⁾、kd-tree に加えて近傍点をキャッシュしておく事により、探索範囲を狭める手法なども提案されている¹⁴⁾。また距離

画像を複数に分割し、PC クラスタを用いて並列に計算させることによって計算効率を高める方法も提案されている¹⁵⁾。

一方で複数距離画像の同時位置合わせは、全ての組合せに対して対応点探索を行う必要があるため、画像枚数が多くなると計算量が急激に増加するという問題がある。すなわち対象となる距離画像枚数を m とすると、全組合せ数は $m \times (m - 1)$ であるため、計算量のオーダーは $O(m^2 n^2)$ となる。また、大規模距離画像を位置合わせする際のもう一つの大きな問題点はメモリ使用量である。通常の同時位置合わせ手法では、全ての組合せに対して繰返し計算を行うために、全距離画像をメモリ中に読込まなければならない。15) のような並列化 ICP でも各プロセッサは全ての距離画像をメモリ中に保持する必要がある。

そこで、本論文で我々は計算の高速化及びデータ分散によるメモリ削減を実現する並列同時位置合わせ手法を提案する。また、本手法を分散メモリ型の PC クラスタ上で実装し、大規模距離画像群の同時位置合わせ実験によりその有用性を示す。まず 2 章では本手法で用いる基本的な位置合わせ手法について説明し、3 章では計算の並列化手法を示す。そして 4 章では本手法の効率を評価し、5 章で大規模な距離画像の同時位置合わせ結果を示す。6 章はまとめである。

2. 同時位置合わせ手法

本章ではまず Neugebauer によって提案された同時位置合わせ手法⁹⁾ を説明する。本論文で提案する手法は、この同時位置合わせ手法を拡張したものである。まず前提条件として各距離画像は予めおおまかに位置合わせされているものとする。位置合わせ手順は ICP と同様であるが、全ての距離画像に対して誤差関数が一つ定義され、全ての距離画像の位置が同時に推定される。手順の概略を以下に示す。

- (1) 全ての距離画像間で以下の計算を行う
 - (a) 全頂点の対応点を探索する
 - (b) 全対応点間の誤差を計算する
- (2) 計算された全ての誤差を最小化するように各距離画像の変換行列を求める
- (3) 終了条件を満たすまで (1) (2) を繰り返す

以下、これらの手順についてそれぞれ説明する。

2.1 対応点探索

まず、全距離画像から 2 つの画像を選択しそれぞれモデル画像、シーン画像と呼ぶことにする。モデル画像上の頂点とそれに対応するシーン画像のメッシュ上の点に対応付けられる。図 1 に示す で示されたデー

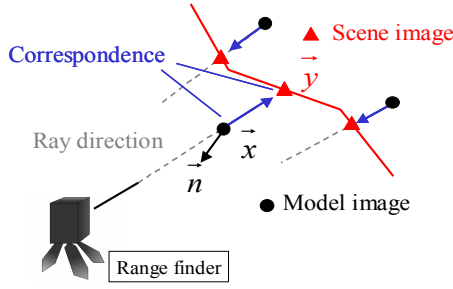


図 1 対応点探索

Fig. 1 Search correspondences

タがモデル画像であり、で表現されたデータがシーン画像である。対応点はモデル画像上の頂点からモデル画像取得時のレンジセンサの視線方向に直線を延ばし、シーン画像のメッシュとの交点である。このときモデル画像の視線方向に対して、対応するシーン画像のメッシュが裏面方向になる場合や、対応点間距離が、与えられた閾値 l_{th} 以上の場合には誤対応として除去する。この対応点探索をモデル画像上の全ての頂点に対して行う。

2.2 誤差計算

対応する2点間の距離には点と面の距離を用いる。モデル画像上の頂点 \vec{x} とそれに対応するシーン画像メッシュ上の点 \vec{y} とし、 \vec{n} をモデル画像上の頂点 \vec{x} の法線とすると(図1)、対応点間誤差 ε は以下になる。

$$\varepsilon = \mathbf{R}_M \vec{n} \cdot \{(\mathbf{R}_S \vec{y} + \vec{t}_S) - (\mathbf{R}_M \vec{x} + \vec{t}_M)\} \quad (1)$$

ここで \mathbf{R}_M, \vec{t}_M はモデル画像の回転、平行移動行列であり \mathbf{R}_S, \vec{t}_S はシーン画像の回転、平行移動行列である。これを全ての距離画像の組合せ (i, j) が含む頂点 k に対して計算し二乗誤差の和 $\bar{\varepsilon}$ が最小となるような変換行列を求める。この式は以下になる。

$$\bar{\varepsilon} = \min_{\mathbf{R}, \vec{t}} \sum_{i \neq j, k} (\mathbf{R}_i \vec{n}_{ik} \cdot \{(\mathbf{R}_j \vec{y}_{ijk} + \vec{t}_j) - (\mathbf{R}_i \vec{x}_{ik} + \vec{t}_i)\})^2 \quad (2)$$

2.3 変換行列の計算

式(2)に示した誤差評価式はこのままでは非線形であり最小化計算が難しいため、これを線形化する。求める変換の回転角を微小角と仮定すると、回転行列 \mathbf{R} は以下のように書くことができる。

$$\mathbf{R} = \begin{pmatrix} 1 & -c_3 & c_2 \\ c_3 & 1 & -c_1 \\ -c_2 & c_1 & 1 \end{pmatrix} \quad (3)$$

平行移動行列を、

$$\vec{t} = \begin{pmatrix} t_x & t_y & t_z \end{pmatrix}^T \quad (4)$$

とすると式(2)は以下のように変形することができる⁹⁾。

$$\bar{\varepsilon} = \min_{\vec{\delta}} \sum_{i \neq j, k} \|\mathbf{A}_{ijk} \vec{\delta} - s_{ijk}\|^2 \quad (5)$$

$$s_{ijk} = \vec{n}_{ik} \cdot (\vec{x}_{ik} - \vec{y}_{ijk}) \quad (6)$$

$$\mathbf{A}_{ijk} = \begin{pmatrix} \underbrace{0 \dots 0}_{6i \times 1} & \underbrace{\mathbf{C}_{ijk}}_{6 \times 1} & \underbrace{0 \dots 0}_{6(l-i-1) \times 1} \end{pmatrix} + \begin{pmatrix} \underbrace{0 \dots 0}_{6j \times 1} & \underbrace{-\mathbf{C}_{ijk}}_{6 \times 1} & \underbrace{0 \dots 0}_{6(l-j-1) \times 1} \end{pmatrix} \quad (7)$$

$$\mathbf{C}_{ijk} = \begin{pmatrix} \vec{n}_{ik} \times \vec{y}_{ijk} \\ -\vec{n}_{ik} \end{pmatrix}^T \quad (8)$$

$$\vec{\delta} = (\vec{m}_0^T \dots \vec{m}_{n-1}^T)^T \quad (9)$$

$$\vec{m}_i = \begin{pmatrix} c_{1i} & c_{2i} & c_{3i} & t_{xi} & t_{yi} & t_{zi} \end{pmatrix}^T \quad (10)$$

ただし全ての距離画像枚数は n である。式(5)より以下のように $\vec{\delta}$ が求められ、最適な \mathbf{R} と \vec{t} が得られる。

$$\left(\sum_{i \neq j, k} \mathbf{A}_{ijk}^T \mathbf{A}_{ijk} \right) \vec{\delta} = \sum_{i \neq j, k} \mathbf{A}_{ijk}^T s_{ijk} \quad (11)$$

3. 同時位置合わせの並列化

本章では1台のサーバと複数台のクライアントからなる分散メモリシステムにおいて、前章で示した同時位置合わせ手法を並列に実現する手法を提案する。

前章で説明した対応点探索及び誤差計算は各距離画像間で独立に計算することができるため、これらの計算を距離画像の組合せ毎にクライアントに割当てる。各距離画像の変換行列の計算は、対応点探索などに比べて計算時間が短い上にメモリ空間を必要としないため、サーバプログラムで行う。また、各クライアントが計算する組合せも同様にサーバプログラムが求める。計算手順を簡略化して示すと以下の様になる。

Algorithm Pseudo Code of Parallel Alignment

```
// Server process
AssignPairs();
while(error > threshold){
  for(i = 0; i < nImage; ++i)
    for(j = 0; j < nImage; ++j)
      if(List[i][j]){
        // Client process
        SearchCorrespondingPoints(i,j);
```

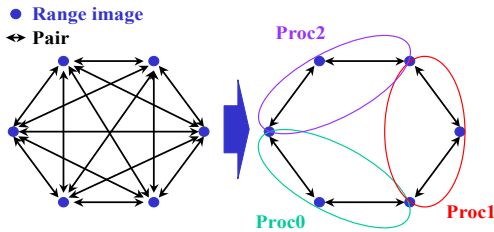


図 2 組合せの除去
Fig. 2 Rejection of redundant pairs

```

    ComputeError(i,j);
  }
  // Server process
  ComputeTransformationMatrices(all);
}

```

対応点探索及び誤差計算を距離画像の全ての組合せに対して行う場合、並列化によるデータ分散はほとんど期待できない。そこで図 2 に示すように予め不要、冗長な組合せを除去する事により、計算を高速化するとともにデータの分散効率を高める事を考える。また、負荷分散とデータ分散を同時に満たすためには、各プロセッサへの組合せの割当ては慎重に計画しなければならない。本章ではこれら 2 つの方法についてそれぞれ説明する。

3.1 組合せの除去

距離画像間の重なりが全く無い場合や重なる領域が小さい場合は、これらの距離画像間で位置合わせ計算を行う必要が無い。また逆に距離画像の重なり枚数が十分多い場合は、これらの距離画像間の計算は冗長とみなす事ができる。そこで本節では大域的、局所的、相対的な観点からこれらの組合せを除去する方法を説明する。

3.1.1 大域的な接続関係

前提条件として各距離画像はおおまかに位置合わせされているものとする。まず以下の条件を満たさない組合せを除去する。

- (1) 2 枚の距離画像の「バウンディングボックス」が重なっている
- (2) 2 枚の距離画像の「視線方向」がなす角度 θ が閾値以内である

(1) は、二つの距離画像間に十分な重なりがあり、初期位置がある程度正確に推定されていれば得られる条件である。(2) は対応点探索が視線方向探索であるため、誤対応を減らすためにも有効な方法である。ただし、対象物の同じ部分を全く異なる方向から測定す

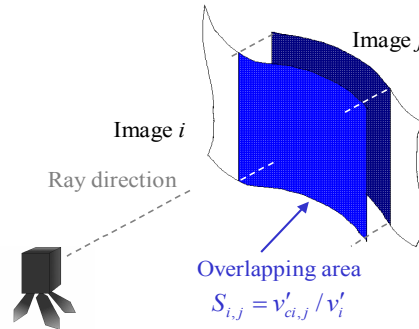


図 3 局所的重なり領域
Fig. 3 Local overlapping area

る場合も十分考えられることから、この条件は裏面除去 ($\theta = 90^\circ$) のために使用するのが望ましい。

3.1.2 局所的重なり領域

次に、各距離画像間の局所的な重なり領域の大きさを計算する。重なり領域の大きさは各距離画像の頂点数 v とそれに対する距離画像との対応点数の比 v_c で表される(図 3)。ただし、対応点探索の計算は多くの計算時間を必要とするため、ランダムに選択された少数の頂点に対して対応点を求める。距離画像 i のランダムに選択された頂点数を v'_i とし、これに対する距離画像 j との対応点数を $v'_{ci,j}$ とすると、距離画像 i からみた距離画像 j との重なり領域の大きさの推定値 $S_{i,j}$ は以下のように表される。

$$S_{i,j} = v'_{ci,j} / v'_i \quad (12)$$

この $S_{i,j}$ と $S_{j,i}$ の大きい方の値が閾値以下の場合、接続関係が弱いとみなして組合せを除去する。実験では閾値を 0.03 ~ 0.05 としている。また、重なり領域の計算は距離画像毎に独立に行う事ができるため、並列化も容易である。この計算は位置合わせ計算の前処理として一度だけ行われるため、順次計算する事が可能であり、メモリ使用量も問題にならない。

3.1.3 相対画像間距離

最後に各距離画像間の距離が相対的に大きい組合せを除去する。ここで、距離画像を内包する最小バウンディングボックスが局所座標系で $(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$ によって表現されるものとする。このとき、距離画像の中心は、 $z = z_{min}$ 上のバウンディングボックス側面の中心とする。そして各距離画像間の距離を、その中心間のユークリッド距離として定義する。図 4 に示すように各距離画像間の距離を計算し、この距離が隣接する距離画像からの距離より大きい場合は組合せを除去する。隣接する距離画像とは、注目している二つの距離画像の両方に接続されている距離画像である。

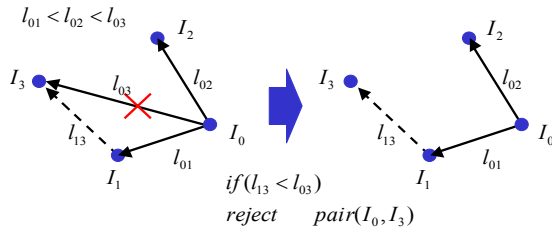


図 4 相対的な距離による組合せの除去
Fig. 4 Rejection pairs by relative distances

例えば図 4 の中で距離画像 I_0 と I_3 に注目すると、 I_0 と I_3 の距離 l_{03} は、 I_1 と I_3 間の距離 l_{13} より相対的に大きいため、 I_0 と I_3 の組合せを冗長な組合せとして除去する。この処理を順次繰り返す事によって、最も近い距離画像の組合せだけを選択する事ができる。またこの方法により誤差の伝播をより均等な方向に分散する事ができる。

3.2 組合せの割当て

以上の手法によって得られた距離画像の組合せを各プロセッサに割当てる。計算時間のみを考慮して並列化する場合は、距離画像の各組合せを順次各プロセッサに割当てていけばよい。しかしメモリ使用量を最小化する場合、問題は NP-困難であるため最適解を得る事は難しい¹⁶⁾。そこで我々はこの問題をグラフ分割問題に帰着し、経験的解法を応用する事によって各プロセッサに割当てる組合せを決定する手法を提案する。

3.2.1 組合せグラフ

まず、分割すべき組合せグラフを定義する。図 5 の左図は距離画像 I_i とそれらの接続関係を示している。この接続グラフから、それぞれの組合せ $P_{i,j}$ をノードとし、各距離画像をネットとしたグラフを作成する(図 5 右)。この場合、各ネットは 2 つ以上のノードに接続されたハイパーグラフとなる。ここで、グラフに含まれるノード重みの和が計算時間を表し、ネット重みの和がメモリ使用量を表すことにする。計算時間は計算すべき総頂点数に比例することから、各ノード重み $W_{i,j}^{node}$ を計算すべき総頂点数、すなわち対応する距離画像 i 及び j の頂点数 v_i と v_j の和とする。またメモリ使用量は各距離画像 i に含まれる頂点数に比例する事から、ネット重み W_i^{net} として頂点数 v_i で表す。

$$W_{i,j}^{node} = v_i + v_j \quad (13)$$

$$W_i^{net} = v_i \quad (14)$$

このハイパーグラフを、分割されたグラフに含まれるノード重みの和が均等でネット重みの和が最小となるように分割する。図 5 右の場合、部分グラフ 0 はペア $(P_{0,3}, P_{1,3}, P_{2,3})$ と距離画像 (I_1, I_2, I_3) を含み、部分

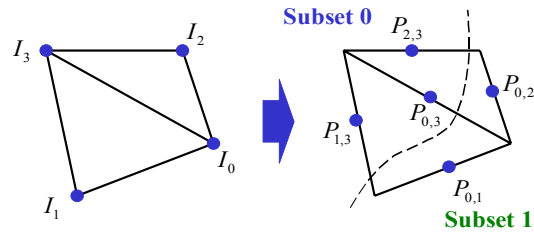


図 5 組合せハイパーグラフ
Fig. 5 Correspondence pair node hyper-graph

グラフ 1 は同様に $(P_{0,1}, P_{0,2})$ と (I_0, I_1, I_2) を含んでいる。最適な分割を求めるために、本論文では

- (1) ノード重みの和が均等になる初期分割を求める
- (2) ネット重みの和が最小になるよう分割を改良するという手法を採用する。

3.2.2 初期分割

組合せハイパーグラフをノード重みの和が均等になるように初期分割する。ここで並列化数を k とし、作成されたグラフは k 個の部分グラフに分割されるものとする。基本となる分割処理ではグラフは 2 分割される。3 個以上の部分グラフに分割する場合には再帰的に 2 分割を繰り返す。再帰的 2 分割法は単純な手法であるが、 $\log k$ 回の 2 分割により比較的良好な k 分割された部分グラフを得ることができる¹⁷⁾。

類似問題の解決法として、切断されるエッジ数を最小化する分割手法はいくつか存在する。一般的な手法としては、固有ベクトルを用いたスペクトル分割手法^{18),19)}がある。しかし問題の性質の違いのため、この手法を本問題に應用する事は難しい。また應用が容易であると考えられるグラフ成長型分割手法としては、KL (Kernighan and Lin) アルゴリズム²⁰⁾を拡張した手法²¹⁾などもある。このような手法は短時間で良好な解が得られるが、局所極小解に陥りやすいという欠点もある。

そこで我々は、ランダムシードを用いた幅優先探索によるグラフ成長手法を用いた^{17),22)}。まず、組合せグラフの全てのノードの中からランダムに一つのノード(シード)を選択する。そして、ノード重みの和が目標とする比率に達するまで、シードから幅優先探索によってグラフを成長させていく。分割結果は選択されるシードによって異なってくる。そこで、複数個の異なるシードを用いて分割を行い、部分グラフに含まれるネット重みの和が最小になる分割を採用する。この手法は局所極小解に陥りにくく、効率の良い初期分割を得る事ができる(図 6)。

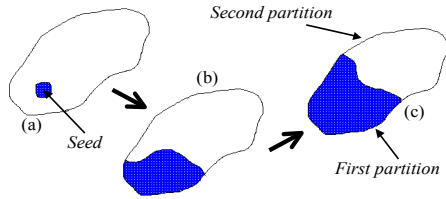


図 6 ランダムシードを用いた幅優先探索
Fig. 6 Random seeded breath-first search

3.2.3 グラフ改良

最後に、各グラフに含まれるネット重みの和が最小になるようグラフの改良を行う。2分割されたグラフを改良する手法としては KL アルゴリズムや、これを拡張した FM (Fiduccia and Mattheyses) アルゴリズム²³⁾があり、KLFM アルゴリズムと呼ばれている。本手法ではこのアルゴリズムを拡張して初期分割されたグラフを改良する。改良は反復計算によって行われる。KLFM アルゴリズムでは片方の分割されたグラフからもう一方のグラフにノードを移動させた場合のエッジカットの変化量をゲインとして定義し、最大のゲインを持つノードを移動させていく事によって最適な解を探索する。通常の KLFM アルゴリズムでは、一つのノードを移動させた場合のゲインを考えるが、本手法では一つのネットを移動させた場合について考える。つまり、あるネットに接続されたノード全てを移動させた場合のゲインを計算する。この方法は先読み法²⁴⁾やクラスタリング²²⁾といった手法のように局所極小解に陥り難い。また、KLFM アルゴリズムは2分割されたグラフの改良についてしか考えていない。そこで k 個に分割されたグラフを改良するために、各部分グラフに含まれるネット重みの和が最大のものと、残された全ての部分グラフと1対1で改良を行い、それ以上改良が進まなくなるまでこの処理を繰り返す。

以下に1対1でのグラフ改良方法を示す。まず部分グラフ G_i と G_j の境界にあるネット $N_{(i,j),k}$ に着目し、このネットに接続された全てのノードがそれぞれの部分グラフに移動した場合のネット重みの変化量を求める。 $N_{(i,j),k}$ が G_i 側に移動した場合の G_i のネット重みの和の変化量を $D_{i,j,k}^{int}$ とし、 G_j の変化量を $D_{i,j,k}^{ext}$ とすると、ゲインは以下のように表される。

$$g_{i,j,k} = D_{i,j,k}^{int} - D_{i,j,k}^{ext} \quad (15)$$

逆に G_j 側に移動した場合のゲイン $g_{j,i,k}$ は内部変化量 $D_{j,i,k}^{int}$ と外部変化量 $D_{j,i,k}^{ext}$ から以下のように表される。



図 7 評価に用いた距離画像 (左:単一幾何モデル, 右:生成された距離画像)

Fig. 7 Range images used for evaluations
(left: original model, right: range images)

$$g_{j,i,k} = D_{j,i,k}^{int} - D_{j,i,k}^{ext} \quad (16)$$

境界に属する全てのネットに対してこれら2つのゲインを計算し、 G_i 側に移動した場合と G_j 側に移動した場合の2つのリストを作成する。そして、これらのリストをそれぞれゲインが小さい順に並び替えてネットを移動させていく。移動させる側のリストは、内包するノード重みの和が大きい方を選択する。一つのネットを移動させる度に、それに関係するネットの情報を更新し、移動させたネットは固定する。また、ネットを移動させた際に最大となるネット重みの和と移動させたネットを記録しておき、一回の反復が終了したときにその最小値が開始時に比べて小さく且つバランスが取れている(誤差1%以内)場合は、開始時からその時点まで移動を進める。またその最小値が開始時より大きい場合は、移動させるネットは無いものとして状態を開始時に戻す。一回の反復は動かすネットが無くなるか、定めた回数に達したら終了する。一回の反復で、最大となるネット重みの和が改良された場合は再度リストを作成して改良を繰り返す。それ以外の場合は改良を終了する。

4. 評価実験

本章では、まず単一の幾何モデルから人為的に合成された距離画像群を用いて、我々の手法の収束性能と精度、計算時間、メモリ使用量を評価する。実験には2つの ArhlonMP2400+プロセッサと4GBのメモリを搭載した、8台PCからなるPCクラスタを使用した。また各PC間は100Base-TXで接続されている。

使用したデータはCyrax2400²⁵⁾によって測定した高さ約13mの鎌倉大仏の完全な3次元モデル(図7左)から、人工的に作成した50枚の距離画像群(図7右)である。各距離画像の頂点数は最大122029点、最

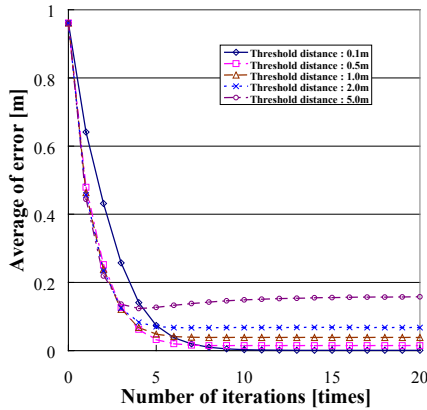


図 8 収束結果 (全ての組合せ)

Fig. 8 Convergence with general method

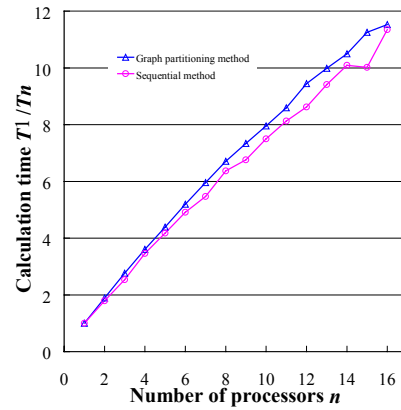


図 10 計算時間の変化

Fig. 10 Computational Efficiency

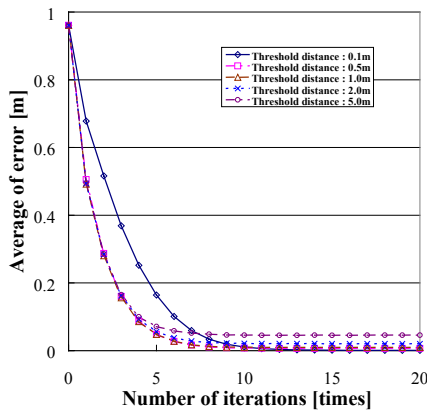


図 9 収束結果 (組合せ除去)

Fig. 9 Convergence with our method

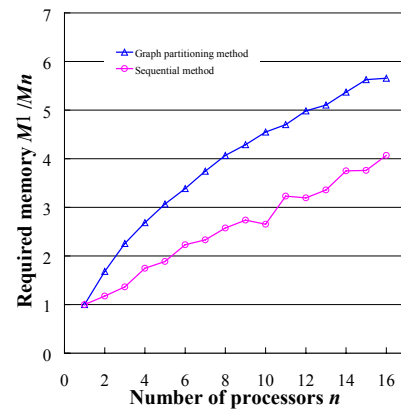


図 11 メモリ使用量の変化

Fig. 11 Required memory

小 12197 点, 平均 83,288 点, また平均メッシュ数は 158,376 である。

4.1 収束と精度

まず, 組合せの除去によって, 収束性能と精度にどのような影響を与えるかを検証する. 使用した距離画像数は 50 枚であるため全組合せ数は 2450 通りであるが, 組合せ選択により 160 通りとなった. ある姿勢における距離画像の位置合わせ精度は, 正確に位置合わせされた距離画像と同一頂点間距離の平均を誤差として評価する. 作成した各距離画像には実測の距離画像に近づけるために, Cyrax で測定した距離画像同様に視線方向に最大 1cm の誤差を与えた. また初期位置は x , y , z 方向にそれぞれ最大距離 10cm でランダムに移動させ, x 軸, y 軸, z 軸に対してそれぞれ最大角度 0.05 ラジアンでランダムに回転させてある. これらの距離画像を対応点探索時の距離閾値 l_{th} をいくつか変えて位置合わせした場合の, 全ての組合せを使用した場合と組合せを除去した場合の収束結果を図 8 と図 9

にそれぞれ示す.

これらの図に示されるように, 組合せの除去を行った場合は, しない場合に比べてより正しい位置に収束している事が分かる. これは不要, 冗長な組合せを除去することによって, 誤対応をより効果的に削減する事ができるためであると考えられる.

4.2 計算時間

次に本手法の計算時間を評価する. ただし計算時間は 1 回の反復計算に要する時間で評価し, 複数回の反復計算を行った場合の平均時間を計算時間とした. プロセッサ数に対する計算時間の変化を図 10 に示す. 図中でプロセッサ数 n における計算時間 T_n は, 1 プロセッサの計算時間 T_1 に対する比で表されている. この図から, 並列化数の増加とともに計算時間がほぼ線形に改善されている事が分かる. また組み合わせグラフを用いないで順次割当てを行った場合より常に高い効率を得られている. 図 7 のモデルに対する本手法の実際の計算時間は 1 プロセッサの場合に 20560ms, 16

表 1 実験結果 (奈良)

Table 1 Experimental results (Nara)

nProc	Time[s]	Max Mem.[MB]	Min Mem.[MB]
2	76.0	1287	1275
16	13.2	292	254

表 2 実験結果 (バイヨン寺院)

Table 2 Experimental results (Bayon Temple)

nProc	Time[s]	Max Mem.[MB]	Min Mem.[MB]
4	103.9	1608	1456
16	40.2	559	472

プロセッサの場合が 1784ms であり、約 11.5 倍の時間短縮が可能となっている。

4.3 メモリ使用量

最後にメモリ使用量についての評価を行う。1~16 並列で計算を行った場合の、各プロセッサに割り当てられる最大メモリ使用量を図 11 に示す。ただし各値は 1 プロセッサ時の最大メモリ使用量との比で表されている。比較のために距離画像を測定した順に割り当てた場合のメモリ使用量も表す。距離画像枚数に対して十分な並列化数が得られる場合は順次割当てと本手法の差は小さいが、相対的に並列化数が少ないほど本手法の効果が大きくなる事が分かる。この場合、2 並列のときに順次割当てが 1.2 倍の効率しか得られないのに対して、本手法は 1.7 倍という高い効果が得られている。図 7 のモデルに対する実際のメモリ使用量は 1 プロセッサの場合に 269Mbyte であるのに対して、16 プロセッサの場合は 48Mbyte となっており、1 プロセッサ時に比べて 17% までメモリ使用量を減少させる事が可能となった。

5. 大規模距離画像の同時位置合わせ

最後にメモリ空間の制約により 1 プロセッサでは計算することができない大規模な距離画像を、本手法によって同時位置合わせした結果を示す。用いたデータは以下の 2 組の距離画像である。

モデル 1. 奈良大仏を Cyrax2400²⁵⁾ で測定した 114 枚の距離画像 (平均頂点数 327470 点, 平均メッシュ数 606072)

モデル 2. バイヨン遺跡を Cyrax2500 によって測定した 210 枚の距離画像 (平均頂点数 433785 点, 平均メッシュ数 798890)

メモリ空間の制約から、計算可能な最小並列化数はモデル 1 の場合は 2 並列, モデル 2 の場合は 4 並列であった。最小並列化数及び最大並列化数 (16) での計算時間及びメモリ使用量 (最大値, 最小値) を、それぞれ表 1 と表 2 に示す。

モデル 1 の場合は並列化数 8 倍に対して計算時間は 5.75 倍, メモリ使用量は 22.6% まで効率が改善されている。またモデル 2 の場合は並列化数 4 倍に対して計算時間は 2.58 倍, メモリ使用量は 34.8% となっている。計算時間が並列化数に比例していないのは、サーバ側の行列演算に多くの計算時間を必要とするためである。例えばモデル 2 の場合はサーバ側での計算時間は約 14 秒であり, 16 並列の場合には 35% もの計算時間が行列演算に使われている事になる。このサーバ側の行列演算に費やされる計算時間は、距離画像枚数の増加につれて大きくなるため今後の課題の一つである。

図 12 にそれぞれ 20 回の反復計算を行った結果, 得られた距離画像を示す。全体の位置合わせ計算時間はモデル 1 が 269 秒, モデル 2 が 812 秒であった。

6. おわりに

本論文では複数枚の距離画像を同時に並列位置合わせする手法を提案した。本手法はまず様々な評価基準を用いて不要, 冗長な組み合わせを除去し, 計算時間の短縮と効率的なデータ分散を図った。さらに計算時間とメモリ使用量を最適化するために, 組合せグラフを定義してグラフ分割アルゴリズムを適用した。また, 本手法を PC クラスタ上で実装し, 収束性能と精度, 計算時間, メモリ使用量についての評価を行った。そして実際に大規模な距離画像を同時位置合わせすることによって本手法の有効性を示した。なお本並列化手法は他の多くの同時位置合わせ手法に適用する事が可能である。今後の課題としては変換行列計算部分を高速化する事が挙げられる。

謝辞 本研究の一部は, 科学技術振興機構戦略的基礎研究事業 (CREST) 高度メディア社会の生活情報技術, 又他の一部は, 文部科学省科学技術振興調整費リーディングプロジェクト「大型有形・無形文化財の高精度デジタル化ソフトウェアの開発」の支援を受けて行われました。また測定にご協力頂いた鎌倉高德院, 奈良東大寺の方々及び, 日本国政府アンコール遺跡救済チーム (JSA) の方々に感謝の意を申し上げます。

参考文献

- 1) Ikeuchi, K. and Sato, Y.: *Modeling from Reality*, Kluwer Academic Press (2001).
- 2) Ikeuchi, K.: *Modeling from Reality*, *Proc. Third International Conference on 3D Digital Imaging and Modeling (3DIM)* (2001).
- 3) Levoy, M.: *The Digital Michelangelo Project*, *Proc. SIGGRAPH 2000*, pp. 131-144 (2000).
- 4) Sagawa, R., Nishino, K. and Ikeuchi, K.:

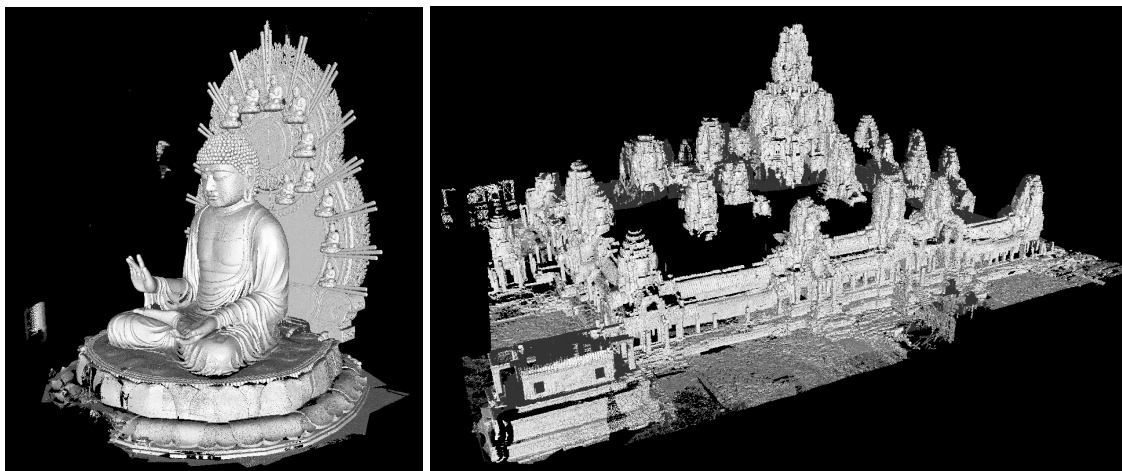


図 12 位置合わせ結果 (左 : 奈良大仏, 右 : バイヨン寺院)

Fig. 12 Alignment result (left : Nara Buddha, right : Bayon Temple)

- Robust and Adaptive Integration of Multiple Range Images with Photometric Attributes, *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2001*, Vol. 2, pp. 172–179 (2001).
- 5) Besl, P. J. and McKay, N. D.: A method for registration of 3-D shapes, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 14, No. 2, pp. 239–256 (1992).
 - 6) Chen, Y. and Medioni, G.: Object modelling by registration of multiple range images, *Image and Vision Computing*, Vol. 10, No. 3, pp. 145–155 (1992).
 - 7) Blais, G. and Levine, M.: Registering Multi-view Range Data to Create 3D Computer Objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 8 (1995).
 - 8) Masuda, T., Sakaue, K. and Yokoya, N.: Registration and Integration of Multiple Range Images for 3-D Model Construction, *Proc. Computer Society Conference on Computer Vision and Pattern Recognition* (1996).
 - 9) Neugebauer, P. J.: Geometrical Cloning of 3D Objects via Simultaneous Registration of Multiple Range Image, *Proc. of the 1997 Int. Conf. on Shape Modeling and Application (SMA'97)*, pp. 130–139 (1997).
 - 10) Benjemaa, R. and Schmitt, F.: Fast global registration of 3d sampled surfaces using a multi-z-buffer technique, *Proc. Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling*, pp. 113–120 (1997).
 - 11) Bergevin, R., M. Soucy, H. G. and Laurendeau, D.: To-wards a general multi-view registration technique, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 5, pp. 540–547 (1996).
 - 12) Nishino, K. and Ikeuchi, K.: Robust Simultaneous Registration of Multiple Range Images, *Proc. of Fifth Asian Conference on Computer Vision ACCV '02*, pp. 454–461 (2002).
 - 13) Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces, *International Journal of Computer Vision*, Vol. 13, No. 2, pp. 119–152 (1994).
 - 14) Simon, D. A., Hebert, M. and Kanade, T.: Realtime 3-D pose estimation using a high-speed range sensor, *Proc. IEEE Intl. Conf. Robotics and Automation*, pp. 2235–2241 (1994).
 - 15) Langis, C., Greenspan, M. and Godin, G.: The parallel iterative closest point algorithm, *Proc. International Conference on 3D Digital Imaging and Modeling (3DIM)* (2001).
 - 16) Garey, M. R. and Johnson, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman & Cie (1979).
 - 17) George, A. and Liu, J. W.-H.: *Computer Solution of Large Sparse Positive Definite Systems*, Englewood Cliffs, New Jersey: Prentice-Hall (1981).
 - 18) Pothén, A., Simon, H.D. and Liou, K.-P.: Partitioning sparse matrices with eigenvectors of graphs, *SIAM Journal of Matrix Analysis and Applications*, Vol. 11, No. 3, pp. 430–452 (1990).
 - 19) Hagen, L. and Kahng, A. B.: New Spectral Methods for Ratio Cut Partitioning and Clustering, *IEEE Transactions on Computer-Aided Design*, Vol. 11, No. 9, pp. 1074–1085 (1992).
 - 20) Kernighan, B. W. and Lin, S.: An Efficient Heuristic Procedure for Partitioning of Electri-

- cal Circuits, *Bell Systems Technical Journal*, Vol. 49, No. 2, pp. 291–307 (1970).
- 21) Karypis, G. and Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs, *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, pp. 359–392 (1998).
- 22) Hauck, S. and Borriello, G.: An Evaluation of Bipartitioning Techniques, *IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems*, Vol. 16, No. 8, pp. 849–866 (1997).
- 23) Fiduccia, C. M. and Mattheyses, R. M.: A Linear-Time Heuristic for Improved Network Partitions, *Proc. Design Automation Conference*, pp. 241–247 (1982).
- 24) Krishnamurthy, B.: An Improved Min-Cut Algorithm for Partitioning VLSI Networks, *IEEE Transactions on Computers*, Vol. C-33, No. 5, pp. 438–446 (1984).
- 25) : <http://www.cyra.com>.

(平成 ? 年 ? 月 ? 日受付)

(平成 ? 年 ? 月 ? 日採録)



大石 岳史

1999年慶應義塾大学理工学部電気工学科卒業。2002年東京大学大学院学際情報学府学際情報学専攻修士課程修了。2005年東京大学大学院学際情報学府学際情報学専攻博士課程修了。現在、東京大学生産技術研究所特任助手。博士(学際情報学)。実物体の形状モデリングの研究に従事。



佐川 立昌

1998年京都大学工学部情報工学科卒業。2000年東京大学大学院工学系研究科情報工学専攻修士課程修了。2003年東京大学大学院工学系研究科電子情報工学専攻博士課程修了。現在、大阪大学産業科学研究所助手。実時間視覚処理と物体のモデリングの研究に従事。



中澤 篤志(正会員)

1974年生。1997年大阪大学基礎工学部システム工学科卒業,1999年同大学院基礎工学研究科修士課程,2001年同大学院博士課程修了。同年科学技術振興事業団研究員(東京大学生産技術研究所)。2003年より大阪大学サイバーメディアセンター講師。博士(工学)。画像計測,分散視覚システム,人体動作解析および生成の研究に従事。日本ロボット学会,ヒューマンインタフェース学会,IEEE各会員。



倉爪 亮(正会員)

1967年2月4日生。1991年東京工業大学機械物理工学専攻修士課程修了。同年(株)富士通研究所入社。1995年同学機械宇宙学科助手。2000年スタンフォード大客員研究員。同年東京大学生産技術研究所博士研究員。2002年より九州大学システム情報科学研究院助教授,現在に至る。博士(工学)。群ロボット,歩行機械,レーザ計測,医用ロボットの研究に従事。



池内 克史(正会員)

1973年京都大学工学部機械工学科卒業。1978年東京大学大学院工学系研究科情報工学専攻博士課程修了。MIT人工知能研究所,電総研,CMU計算機科学科を経て,1996年より東京大学生産技術研究所教授。2000年より東京大学大学院情報学環教授兼任。人間の視覚機能,明るさ解析,物体認識,人間行動観察学習ロボット,高度交通システムなどの研究に従事。工学博士。D.Marr賞(ICCV:1990年),IEEE優秀論文賞(CVPR:1991年),最多引用論文賞(AI Journal:1992年)Fu記念優秀論文賞(IEEE Trans. R&A,1998年)等受賞。IEEE Distinguished Lecturer(SPS 2000-2001,CS 2004-2006),IEEE Fellow。