

PC クラスタによる複数距離画像の並列同時位置合わせ

大石岳史^{†1} 佐川立昌^{†2} 中澤篤志^{†3} 倉爪亮^{†4} 池内克史^{†1}

†1 東京大学生産技術研究所 †2 大阪大学産業科学研究所
†3 大阪大学サイバーメディアセンター †4 九州大学大学院システム情報科学研究院

E-mail: †1 {oishi, ki}@cvtl.iis.u-tokyo.ac.jp, †2 sagawa@am.sanken.osaka-u.ac.jp,
†3 nakazawa@ime.cmc.osaka-u.ac.jp, †4 kurazume@is.kyushu-u.ac.jp

あらまし 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。大規模距離画像の同時位置合わせを実現するために、我々は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並列化手法を開発した。位置合わせ計算で最も計算コストが高い対応点探索及び誤差計算は、各距離画像間で独立に行う事ができる。そこで、これらの計算を距離画像の組合せ毎にプロセッサに割り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グラフ分割の手法を用いた。そして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の同時位置合わせ実験を行った。

キーワード 距離画像, 位置合わせ, 並列計算, PC クラスタ

Parallel Simultaneous Alignment of Multiple Range Images on PC Cluster

Takeshi Oishi^{†1} Ryusuke Sagawa^{†2} Atsushi Nakazawa^{†3}
Ryo Kurazume^{†4} Katsushi Ikeuchi^{†1}

†1 Institute of Industrial Science, The University of Tokyo

†2 Institute of Scientific and Industrial Research, Osaka University †3 Cybermedia Center, Osaka University

†4 Graduate School of Information Science and Electrical Engineering, Kyushu University

E-mail: †1 {oishi, ki}@cvtl.iis.u-tokyo.ac.jp, †2 sagawa@am.sanken.osaka-u.ac.jp,
†3 nakazawa@ime.cmc.osaka-u.ac.jp, †4 kurazume@is.kyushu-u.ac.jp

Abstract This paper describes a method for parallel alignment of multiple range images. For aligning a large number of range images simultaneously, we developed the parallel method that improves the time and memory performances of the process. Since the computation between two range images can be performed independently, the computation of each correspondence pair of range images is assigned to each processor. By rejecting redundant dependencies, our method makes it possible to accelerate computation time and reduce the amount of memory used on each node. The graph partitioning algorithms are applied to this problem in order to obtain an optimal solution for pair assignment. The method was tested on a 16 processor PC cluster, where it demonstrated the high extensibility and the performance improvement in time and memory.

Keyword Range Image, Alignment, Parallel Computation, PC Cluster

1. はじめに

近年、レーザレンジセンサの発達により、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[1,2]。これらの計測に用いられるセンサの測定範囲はレーザの届く範囲に限られるため、一回の測定で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。

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[3,4,5]。特によく知られている手法はBeslが提案したICP (Iterative Closest Point)法[6]で、ほとんどの手法はこのICPの拡張である。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な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。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り返していくと誤差の蓄積が問題となる。そこで、これを避けるために、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も多く提案されている[7,8,9,10]。

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。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[11], 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[12]。距離画像を複数に分割し、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[13]。しかし、繰り返し計算を行うためには全距離画像をメモリ中に読込まなければならないため、メモリ使用量も大きな問題となる。[13]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、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する事によりその有用性を示す。まず2章で同時位置合わせの並列化手法を示す。そして3章では本手法の効率を評価し、4章で大規模な距離画像の同時位置合わせ結果を示す。5章はまとめである。

2. 並列同時位置合わせ手法

本章では同時位置合わせ処理を並列化する手法について説明する。対象となるシステムは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である。基本となる同時位置合わせ手法には、我々はNeugebauerの手法を用いた[7]。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。

2.1. 処理の並列化

位置合わせ処理で最も計算コストが高い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。また、それ以外の計算は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。計算手順を簡略化して示すと以下の様になる。

```

Algorithm Procedure Code of Parallel Alignment
Pre-Process(); // Server
while(error > threshold){
  for(i = 0; i < nImage; ++i)
    for(j = 0; j < nImage; ++j)
      if(List[i][j]){
        SearchCorrespondingPoints(i, j); // Clients
        CalculateErrorMatrix(i, j); // Clients
      }
  CalculateTransformationMatrices(all); // Server
}

```

ただし、対応点探索及び誤差計算を全ての距離画像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。そこで予め効率的な組合せを選択する事により、計算を高速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、各クライアントへの組合せの割当ては慎重に計画しなければならない。次節以降でこれらの方法について説明する。

2.2. 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。

2.2.1. 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」がなす角度が閾値以内である

ここで、バウンディングボックスとは距離画像を包含する最小の直方体であり、視線方向とはセンサ座標系で-z軸方向である。条件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なければならない条件である。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。

2.2.2. 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_i で表される(図1)。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。

$$S_{i,j} = v'_{ci,j} / v'_i \quad (1)$$

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並列化も容易であり、位置合わせ計算の前処理として一度だけ行われるため、順次計算する事によってメモリ使用量が問題になる事もない。

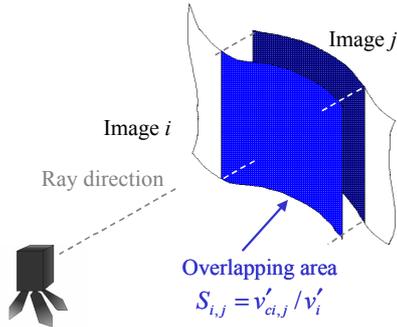


図1 局所的重なり領域
Fig.1 Local overlapping area

2.2.3. 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。ただし各距離画像間の距離は、各画像のバウンディングボックスの視線方向側面を表面中心とし、その表面中心間のユークリッド距離と定義する。図2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対応関係を除去する。

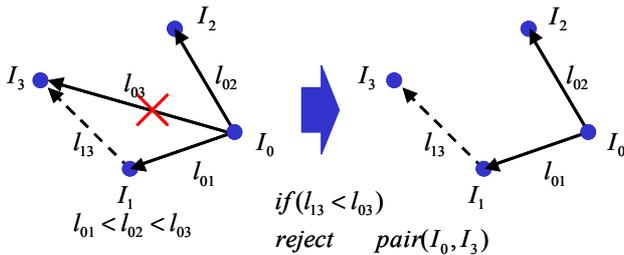


図2 相対的な距離による組合せの除去
Fig.2 Rejection pairs by relative distances

例えば図2の中で距離画像 I_0 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。この処理後の一つの距離画像に対する組合せ数は 12 以下となるため、組合せ数 m に対する計算量は $O(m^2)$ から $O(12m)$ となる。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させる事ができる。

2.3. 組合せの割当て

以上の処理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[14]。そこで我々はこの問題をグラフ分割問題に帰着し、経験的解法を適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。

2.3.1. 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3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3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 $W^{node}_{i,j}$ 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^{net}_i として頂点数 v_i で表す。

$$W^{node}_{i,j} = v_i + v_j \quad (2)$$

$$W^{net}_i = v_i \quad (3)$$

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3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_0, I_2) を含み、部分グラフ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。

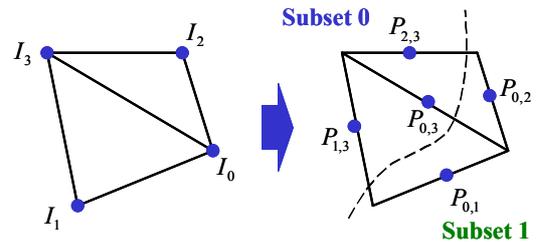


図3 組合せハイパーグラフ
Fig.3 Correspondence pair node hyper-graph

2.3.2. 初期分割

作成された組合せハイパーグラフは、まずノード重みの和が均等になるように初期分割される。切断されるエッジ数が最小になるように2分割する手法には、固有ベクトルを用いたスペクトル分割手法[15,16]などがあるが、これらを本問題に適用する事は難しい。そこで我々は、ランダムに選択されたノードから、ノード重みの和が均等になるまで幅優先探索によってグラフを成長させていく手法を用いた[17,18]。この手法は局所極小解に陥りにくく、比較的短時間で効率の良い分割が得られる(図4)。

分割されたグラフは最初に選択されたノードによって変化するため、ランダムに選択された複数のノ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。また k 個のグラフに分割するためには再帰的に2分割を繰り返す。この場合 k はプロセッサ数を表す。この手法は単純であるが、 $\log k$ 回の2分割により比較的良好な k 分割されたグラフを得ることができる[19]。

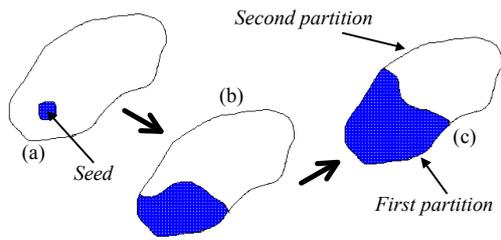


図4 ランダムシードを用いた幅優先探索
Fig.4 Random seeded breath-first search

2.3.3. 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。本手法では反復改良法である KLFM アルゴリズム[20,21]を拡張する。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る事によって最適な解を探索する。本手法では、あるネットに接続された全てのノードを移動させた場合のゲインによって移動させるノードを決定する。また、 k 個に分割されたグラフを改良するためには、各部分グラフに含まれるネット重みの和が最大のもので、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。

$$g_{i,j,k} = D_{i,j,k}^{int} - D_{i,j,k}^{ext} \quad (4)$$

逆に G_i 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。

$$g_{j,i,k} = D_{j,i,k}^{int} - D_{j,i,k}^{ext} \quad (5)$$

境界に属する全てのネットに対して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。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、移動され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%以下)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。

3. 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 AthlonMP2400+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。

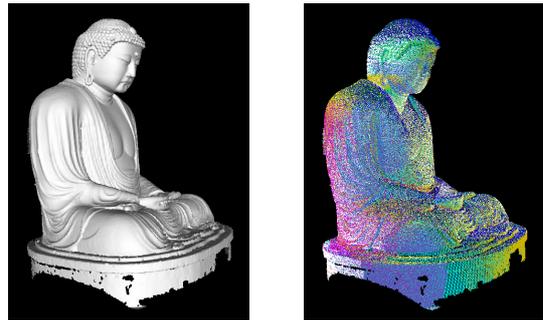


図5 評価に用いた距離画像
(左:単一幾何モデル, 右:生成された距離画像列)
Fig.5 Range images used for evaluations
(left: original model, right: range images)

使用したデータはCyraX2400[22]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5左)から、人工的に作成した50枚の距離画像群(図5右)である。各距離画像の頂点数は最大122,029点、最小12,197点、平均83,288点、また平均メッシュ数は158,376である。

3.1. 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2,450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。

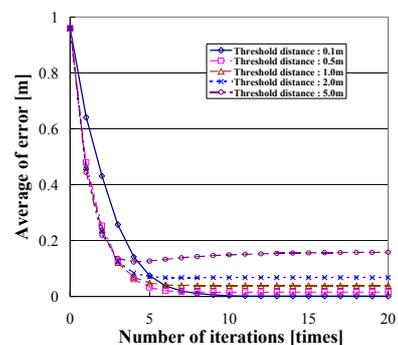


図6 収束結果(全ての組合せ)
Fig.6 Convergence with general method

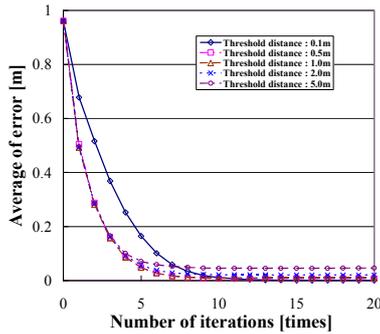


図7 収束結果 (組合せ除去)
Fig.7 Convergence with our method

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、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 6 と図 7 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ことができるためであると考えられる。

3.2. 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 8 に示す。

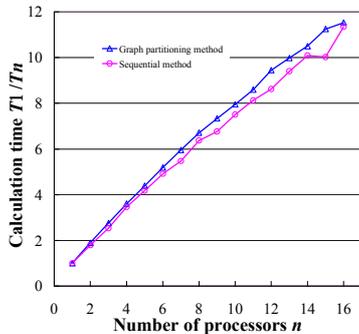


図8 計算時間の変化
Fig.8 Computational Efficiency

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。また組合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 5 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 20,560ms、16 プロセッサの場合が 1,784ms であり、約 11.5 倍の時間短縮が可能となっている。

3.3. 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、各プロセッサに割当てら

れる最大メモリ使用量を図 9 に示す。ただし各値 M_n は 1 プロセッサ時の最大メモリ使用量 M_1 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当てた場合のメモリ使用量も表す。この図から、順次割当てより本手法の方が非常に良い効率が見られる事が分かる。また相対的に並列化数が少ないほど本手法の効果が大きい事も分かる。この場合、2 並列の場合は順次割当てが 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 5 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。

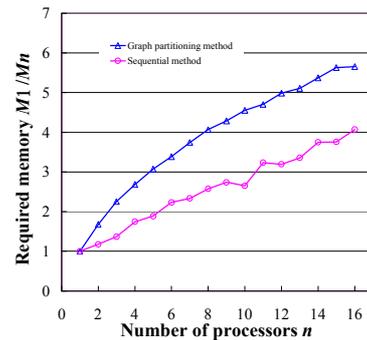


図9 メモリ使用量の変化
Fig.9 Required memory

4. 大規模距離画像の同時位置合わせ

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 433,785 点、平均メッシュ数 798,890) である。メモリ空間の制約から、計算可能な最小並列化数は 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) を表 1 に示す。

表 1 実験結果 (バイヨン寺院)

Table 1 Experimental results (Bayon Temple)

Processor	Ave. Time(sec)	Max. Mem(MB)	Min. Mem(MB)
4	103.9	1608	1456
16	40.2	559	472

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% まで効率が改善されている。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。こ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、今後距離画像枚数の増加につれて大きな問題になると考えられる。図 10 に 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 812 秒であった。

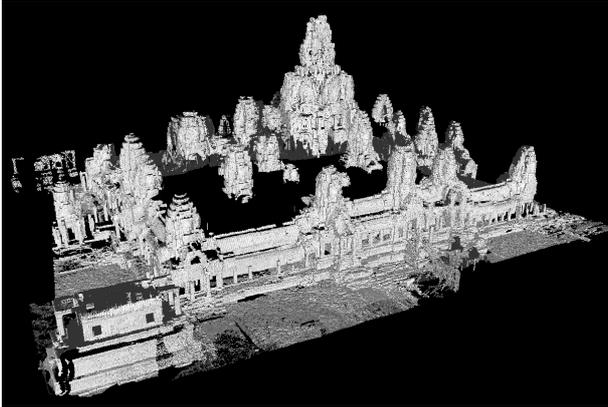


図 10 位置合わせ結果 (パイヨン寺院)

Fig.10 Alignment result(Bayon Temple)

5. 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、冗長な組合せを除去し、さらに計算時間とメモリ使用量を最適化するために、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。さらに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。

謝辞

本研究は科学技術振興機構・戦略的創造研究推進事業 (CREST) 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。

文 献

- [1] K. Ikeuchi and Y. Sato, *Modeling from Reality*, Kluwer Academic Press, 2001.
- [2] M. Levoy et. al. The Digital Michelangelo Project. In *Proc. SIGGRAPH 2000*, pp.131-144.
- [3] Y. Chen and G. Medioni, Object modeling by registration of multiple range images, *Image and Vision Computing* 10(3), 1992, 145-155.
- [4] G. Blais and M. Levine. "Registering Multiview Range Data to Create 3D Computer Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 8, 1995.
- [5] T. Masuda, K. Sakaue, and N. Yokoya, Registration and Integration of Multiple Range Images for 3-D Model Construction. In *Proc. Computer Society Conference on Computer Vision and Pattern Recognition*, 1996.
- [6] P. J. Besl and N. D. McKay, A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2) 1992, 239-256.
- [7] P. J. Neugebauer. Reconstruction of Real-World Objects via Simultaneous Registration and Robust Combination of Multiple Range Images. *International Journal of Shape Modeling*, 3(1&2):71-90, 1997.
- [8] R. Benjema and F. Schmitt. Fast global registration of 3d sampled surfaces using a multi-z-buffer technique. In *Proc. Int. Conf. on Recent Advances in 3-D Digital Imaging and Modeling*, pages 113-120, May 1997.
- [9] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):540-547, May 1996.
- [10] K. Nishino and K. Ikeuchi, Robust Simultaneous Registration of Multiple Range Images, In *Proc. Fifth Asian Conference on Computer Vision*, pp454-461, Jan., 2002.
- [11] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119-152, 1994.
- [12] D. Simon, M. Hebert and T. Kanade. Realtime 3-D pose estimation using a high-speed range sensor. In *IEEE Intl. Conf. Robotics and Automation*, pages 2235-2241, San Diego, California, May 8-13 1994.
- [13] C. Langis, M. Greenspan and G. Godin. The parallel iterative closest point algorithm. In *Proc. International Conference on 3D Digital Imaging and Modeling (3DIM)*, Quebec City, Quebec. May 28-June 1, 2001.
- [14] M. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco, CA: Freeman, 1979.
- [15] Alex Pothén, Horst D. Simon, and Kang-Pu Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal of Matrix Analysis and Applications*, 11(3):430-452, 1990.
- [16] L. Hagen, A. B. Kahng, "New Spectral Methods for Ratio Cut Partitioning and Clustering", *IEEE Transactions on Computer-Aided Design*, Vol. 11, No. 9, pp. 1074-1085, September, 1992.
- [17] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359-392, 1998.
- [18] S. Hauck and G. Borriello, "An Evaluation of Bipartitioning Techniques", *IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems*, Vol. 16, No. 8, pp. 849-866, August 1997.
- [19] A. George and J. W.-H. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [20] B. W. Kernighan, S. Lin, "An Efficient Heuristic Procedure for Partitioning of Electrical Circuits", *Bell Systems Technical Journal*, Vol. 49, No. 2, pp. 291- 307, February 1970.
- [21] C. M. Fiduccia, R. M. Mattheyses, "A Linear-Time Heuristic for Improved Network Partitions", *Design Automation Conference*, pp. 241-247, 1982.
- [22] <http://www.cyra.com>.