コンピュータビジョンと
イメージメディア 137－4
（２００３．３．２７）

PCクラスタによる大規模距離画像の並列アライメント

大石岳史　佐川立昌　中澤篤志　倉爪亮　池内克史

東京大学生産技術研究所

oishi@cvl.iis.u-tokyo.ac.jp　sagawa@cvl.iis.u-tokyo.ac.jp

nakazawa@cvl.iis.u-tokyo.ac.jp　kurazume@is.kyushu-u.ac.jp

ki@cvl.iis.u-tokyo.ac.jp

概要

本論文では複数方向から測定された距離画像を並列にアライメントする手法を提案する．大規模な距離画像群の同時アライメントは，計算時間及びメモリ使用量から考えてあまり現実的ではない．そこでアライメント計算を並列化し，データを分散する事によってこれまで難しかった大規模な距離画像の同時アライメントを可能にする手法を提案する．また並列処理をPCクラスタ上で行い，高い拡張性と，時間効率とメモリ使用効率という点に関して本手法の有効性を確かめる．

キーワード：距離画像，アライメント，PCクラスタ，並列計算

Parallel Alignment of a Large Number of Range Images on PC Cluster

Takeshi Oishi　Ryusuke Sagawa　Atsushi Nakazawa
Ryo Kurazume　Katsushi Ikeushi

Institute of Industrial Science, University of Tokyo

oishi@cvl.iis.u-tokyo.ac.jp　sagawa@cvl.iis.u-tokyo.ac.jp

nakazawa@cvl.iis.u-tokyo.ac.jp　kurazume@is.kyushu-u.ac.jp

ki@cvl.iis.u-tokyo.ac.jp

Abstract

This paper describes a parallel alignment method of multiple range images. It is difficult to align a large number of range images simultaneously. Therefore, we developed the parallel method to improve the time performance and memory performance. The method was tested on 16 processor PC cluster and shown the high extendibility and the performance improvement in time and memory.

Key word : range image, alignment, PC cluster, parallel processing

1. はじめに

近年，レーザレンジファインダの発展により，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といった長距離でも1cm以下の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．

しかし大規模な距離画像から3次元モデルを生成するためには様々な問題がある．レーザレンジファインダの測定可能箇所はレーザの届く範囲に限られるため，一度の測定では物体全体をモデル化することは不可能である．そのため，完全な3次元形状を得るためには複数方向から距離画像を測定し，その後いくつかの計算処理を行わなければならない．まず行われる各距離画像の相対位置姿勢を求めるアライメント処理は，それぞれ異なる座標系にある複数距離画像の座標系を統一する処理である．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て，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きな建造物などの場合は，レンジファインダの位置や姿勢を正確に記録することは非常に難しく，各距離画像間の相対位置関係は後処理として計算で求めなければならない．大規模な距離画像のアライメント処理は計算時間やメモリ使用量の点から難しい処理となっている．アライメント処理の次には，各距離画像を一つのモデルに統一するマージングと呼ばれる処理が行われる．マージングにはデータをつなぎ合わせるZipper 法[13]やボクセルを用いた手法[14, 15]などがあり，大規模距離画像に対しても並列計算を行う手法が提案されている[16]．そこで本論文では問題となっている大規模距離画像のアライメントについて考える．

距離画像のアライメント手法はBesl が提案したICP (Iterative Closest Point)法や[3]，Chen の手法[4]を基本として多くの手法が提案されている．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．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[5]．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[6]．このように，様々なアライメント手法が提案されてきているが，いずれの手法にも問題となるのが，対応点探索の計算コストである．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，二つの距離画像の頂点数を等しくnとすると，計算量は$O(n^2)$である．そのため頂点数が多くなるほど，非常に多くの計算時間を必要とする事がわかる．そこでICP を高速化する手法としては探索木(kd-tree)を使う手法や[7]，kd-tree に加えて近傍点をキャッシュしておく事によって，探索範囲を狭める手法などもある[8]．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[9]．

ここまでの手法は2枚の距離画像の位置合わせを行う手法であるが，距離画像枚数が多い場合にはペアワイズな手法では誤差の蓄積が問題となる．そこで，多くの枚数を必要とする大規模な距離画像群に対しては全ての距離画像間の相対位置姿勢を同時に推定する全体位置合わせ手法が必要である．Neugebauer は視線方向に対応点を探索し，点と面の距離を誤差関数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[10]．この手法は本研究のアルゴリズムの基礎となっている．Benjamaa 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アライメント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を行っている[12]．しかし，このような手法を用いても我々が扱うような大規模な距離画像を同時にアライメントする事は非常に難しい．なぜならばペアワイズな対応点探索に加えて距離画像の組合せ分の計算量が増加する上に，全ての距離画像をメモリ中に読込む必要があるためである．また今後のレンジファインダの性能向上や計測技術の発展によって，更に扱うデータ量は増えていくと考えられる．そのため計算時間が短くメモリ使用量が少ない，かつ拡張性の高い手法が要求されると考えられる．そこで我々は安価で拡張性の高いPC クラスタを用いて，データ分散，並列計算を行うことにより大規模距離画像の同時位置合わせを行う手法を提案する．

まず2章では本手法で用いる基本的なアライメントアルゴリズムについて説明し，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章で大規模距離画像のアライメント結果を示す．6 章はまとめである．

2. アライメントアルゴリズム

本章ではまず基礎となるアライメントアルゴリズムの概要を説明する．基本的なアライメント手順は以下の通りである．

1. 2枚の距離画像の頂点同士の対応点を求める．
2. 対応点間の誤差を求める．
3. 求めた誤差を最小化するように各距離画像に対する変換行列を求める．

4. 求めた変換行列を各距離画像に適用する.
5. 誤差が十分小さくなるまで 1〜4 を繰り返す.

2.1 対応点探索

ICP アルゴリズムでは 2 つの距離画像間の誤差として対応する頂点間の距離を用いるが，本手法ではモデル画像上の頂点とシーン画像のメッシュ上の点を対応付ける. Figure 2.1 に示す点群で示されたデータがモデル画像で面で表現されたデータがシーン画像である.

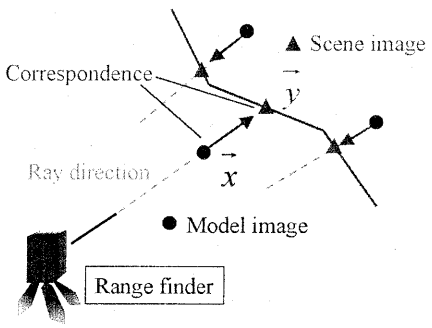

Figure 2.1 Search Correspondence

対応点はモデル画像上の頂点からモデル画像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を対応点としている. これをモデル画像上の全ての頂点に対して探索を行う. このときモデル画像の視線方向と，対応するメッシュの法線方向が異なる場合と，対応点間距離が与えられた閾値以上の場合は誤対応として除去する.

2.2 誤差評価

対応する 2 点間の距離は点と面の距離を用いる. Figure 2.2 に示すように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 $\vec{y}$ とすると，この 2 点間の誤差は式 (1) で表される.

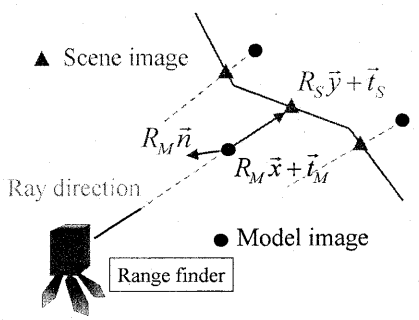

Figure 2.2 Error Evaluation

$$\vec{n}\cdot(\vec{y}-\vec{x}) \tag{1}$$

ここで $\vec{n}$ はモデル画像上の頂点 $\vec{x}$ の法線であり，予め計算しておく. アライメントはこの誤差を最小化するようにモデル画像及びシーン画像の変換行列を求める. この変換行列を含めた誤差評価式は (2) 式のようになる.

$$R_M\vec{n}\cdot\{(R_S\vec{y}+\vec{t}_S)-(R_M\vec{x}+\vec{t}_M)\} \tag{2}$$

ここで $R_M \cdot \vec{t}_M$ はモデル画像の回転・平行移動行列であり $R_S \cdot \vec{t}_S$ 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に対して計算し二乗誤差が最小となるような変換行列を求める. この式は (3) のようになる.

$$\varepsilon^2=\min_{R,t}\sum_{i\neq j,k}\left(R_M\vec{n}\cdot\{(R_S\vec{y}+\vec{t}_S)-(R_M\vec{x}+\vec{t}_M)\}\right)^2 \tag{3}$$

2.3 変換行列の計算

(3) に示した誤差評価式はこのままでは非線形なので線形化する. 求める変換の回転角を微小角と仮定すると，回転行列 R は式 (4) のように書くことが出来る.

$$R=\begin{pmatrix}1 & -c_3 & c_2\\ c_3 & 1 & -c_1\\ -c_2 & c_1 & 1\end{pmatrix} \tag{4}$$

平行移動行列を,

$$t=\begin{pmatrix}t_x & t_y & t_z\end{pmatrix} \tag{5}$$

とすると誤差評価式 (3) は以下のように変形する事ができる.

$$\varepsilon^2=\min_{\delta}\sum_{i\neq j,k}\left\|A_{ijk}\vec{\delta}-s_{ijk}\right\|^2 \tag{6}$$

$$s_{ijk}=\vec{n}_{ik}\cdot(\vec{x}_{ik}-\overrightarrow{y_{ijk}}) \tag{7}$$

$$A_{ijk}=\left(\underbrace{0\ldots0}_{6i\times1}\underbrace{C_{ijk}}_{6\times1}\underbrace{0\ldots0}_{6(l-i-1)\times1}\right)+\left(\underbrace{0\ldots0}_{6j\times1}\underbrace{-C_{ijk}}_{6\times1}\underbrace{0\ldots0}_{6(l-j-1)\times1}\right) \tag{8}$$

$$C_{ijk}=\begin{pmatrix}\overrightarrow{n_{ik}\times y_{ijk}}\\ -\vec{n}_{ik}\end{pmatrix} \tag{9}$$

$$\vec{\delta}=\begin{pmatrix}m_0\cdots m_{n-1}\end{pmatrix} \tag{10}$$

$$m_i=\begin{pmatrix}c_{1i} & c_{2i} & c_{3i} & t_{xi} & t_{yi} & t_{zi}\end{pmatrix} \tag{11}$$

このとき全ての距離画像枚数は n である. 変換行列 R と t を求めるためには (6) 式中の $\vec{\delta}$ を求めれば良いことが分かる. 式 (6) より,

$$\vec{\delta} = \left(\sum_{i \neq j,k} A^T{}_{ijk} A_{ijk} \right)^{-1} \sum_{i \neq j,k} A^T{}_{ijk} s_{ijk} \qquad (12)$$

が得られ，最適な変換行列を求めることが出来る．

ここで，収束を早めるためにモデル画像上の頂点法線とシーン画像上のメッシュの法線を平均したベクトル$\overrightarrow{n_{xy}}$を定義し，モデル画像上の頂点$\vec{x}$の法線$\overrightarrow{n_x}$の代わりに用いる．

$$\overrightarrow{n_{xy}} = \frac{\overrightarrow{n_x} + \overrightarrow{n_y}}{\left\| \overrightarrow{n_x} + \overrightarrow{n_y} \right\|} \qquad (13)$$

$\overrightarrow{n_x}$はシーン画像上の対応点（面）法線である．この平均法線ベクトルを用いると誤差評価式は以下のように書き直される．

$$\overrightarrow{n_{xy}} \cdot (\vec{y} - \vec{x}) \qquad (14)$$

$$R_M \vec{n}_{xy} \cdot \{(R_S \vec{y} + \vec{t}_S) - (R_M \vec{x} + \vec{t}_M)\} \qquad (15)$$

$$\overrightarrow{n_{xy}}' = R_M \overrightarrow{n_{xy}} \qquad (16)$$

ただし簡単化のために$\overrightarrow{n_{xy}}$に掛かる回転行列は，モデル画像に対する回転行列R_Mとした．平均誤差を用いる方法では，より正確に誤差が評価できるとともに方程式の形そのものは変わらないため，線形化及び変換行列は同様の計算式で求めることが出来る．

3. アライメントの並列化

通常の全体位置合わせでは，全ての距離画像の組合せに対して計算を実行する必要がある．しかし全ての組合せを計算する場合には，全ての距離画像データをメモリ中に読込む必要があるため，並列化処理によってデータを分散することはできない．また距離画像間の重なりが無い場合や，重なる部分が小さい場合は位置推定計算に影響を及ぼさないため，冗長な計算になる．そこで，予めこれらの冗長な組合せを取り除く事によって，計算の高効率化及びデータの分散を行う事ができる．本手法では以下の 3 つの条件によって冗長な組合せを除去した．

1. 二つの距離画像の「バウンディングボックス」が重なっている
2. 二つの距離画像の「視線方向」がなす角度が閾値以内である
3. 二つの距離画像が「隣接状態」にある

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．3 は Figure3.1 に示すように，二つの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破棄するという処理によって得られる条件である．Figure 3.1 中での各頂点は各距離画像の表面中心を表しており，この表面中心間のユークリッド距離を各距離画像間の距離としている．表面中心はバウンディングボックスの視線方向側面の中心である．

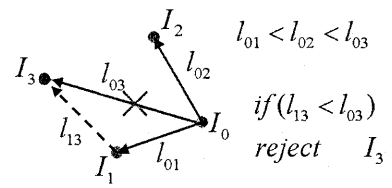

Figure 3.1 Closest Range Images

3.1 計算時間の効率化

アライメント計算を最も単純に並列化する方法は各距離画像間の計算を各ノードに割り振ることである．各距離画像間の計算はそれぞれ独立に行う事ができるため，これらの計算を順にノードに割り当てていく．

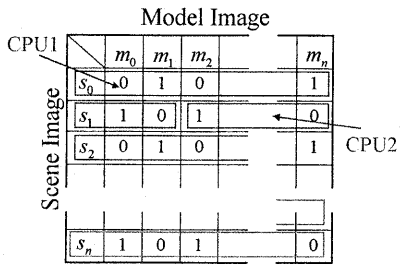

Figure3.2 Parallel Processing

実装上の理由によってシーン画像を基準として計算することで効率良く計算することができるため，Figure 3.2 に示すように各ノードに対してシーン画像を$s_0, s_1, \ldots, s_n$という順に割り当て，更にそのシーン画像に対応するモデル画像を割当てていく．本手法では，各距離画像中の頂点数及びメッシュ数はほぼ等しいものとして扱っているため，それぞれのノードに割当てる対応数c_iは全対応数c_{all}を全ノード数（CPU 数）n_{cpu}で割った値となる．

$$c_i = c_{all} / n_{cpu} \qquad (17)$$

また各ノードiの計算時間はこの対応数c_iに比例する．実際にはシーン画像の前処理計算が必要なため，割当てられたシーン画像数に比例した時間も加算されるが，

モデル画像数に対してシーン画像数は少ない上に，各ノードに割当てられるシーン画像数はほぼ等しいと考えられるため考慮する必要は無い．

また，この方法ではノード数に対して距離画像数が少ない場合は，各ノードに均等に対応を割当てる事ができないため，計算時間の効率化が図れない場合がある．しかし本手法が対象としているのは大規模な距離画像群であるため，この方法で十分な効果が得られるものと考えられる．

3.2 メモリ使用効率

前述のように通常の全体位置合わせでは全ての距離画像をメモリ中に読込む必要があるが，冗長な対応関係を除去して並列化を行った場合は計算する距離画像のみをメモリ中に読込めばよい．各距離画像の頂点数，メッシュ数がほぼ等しいものとすると，各ノードが必要とするメモリ量 M_i は割当てられた距離画像数 r_i に比例する．

$$M_i = r_i \times M_r + M_c \tag{18}$$

このとき M_r は各距離画像の平均データ量，M_c は計算に必要なメモリ量で，距離画像枚数によらず一定とみなす事ができる．各ノードが読込む距離画像 r_i は，使用するモデル画像 $r_{model,i}$ とシーン画像 $r_{scene,i}$ の和集合で表される．

$$r_i = r_{model,i} \cup r_{scene,i} \tag{19}$$

アライメントの対象となる距離画像群は任意に並んでいるので，3.1 節で説明したように順に振り分けていくだけでは効率良くデータを分散することはできない．そこで Figure 3.2 に示されるようなテーブルを，それぞれのノードが読込む距離画像が少なくなるように最適化する．各ノードが読込む距離画像は使用するモデル画像とシーン画像の和集合になるため，並び合う二つのシーン画像に対応するモデル画像が共通している方がよい．そこで二つのシーン画像 (i, j) 間の距離 $l_{i,j}$ を以下のように定義して，この距離が小さい順に並び替えていく．

$$l_{i,j} = \sum_{0 \le k \le n} m_{i,k} \cup m_{j,k} - m_{i,k}(xor)m_{j,k} \tag{20}$$

$$m_{i,k} = \begin{cases} 1 & (\text{画像}i\text{と画像}k\text{が対応している}) \\ 0 & (\text{画像}i\text{と画像}k\text{が対応していない}) \end{cases}$$

n は総距離画像数である．

3.3 実装

プログラムはサーバークライアントの形で実装した．以下に計算手順を説明する．サーバ側では全ての距離画像のバウンディングボックスと初期状態から推定された状態への変換行列だけを保持し，各クライアント(ノード)が受け持つ距離画像対を計算してクライアントに伝える．各クライアントはサーバから受け取った距離画像対のリストを元にして必要な距離画像をメモリ中に読込み，式(12)の ${A^T}_{ijk} A_{ijk}$ と ${A^T}_{ijk} s_{ijk}$ の項をそれぞれ独立に計算してサーバに送る．サーバは全てのクライアントから受け取った行列から，全ての距離画像に対する変換行列を計算し適用するとともに，各クライアントにこの変換行列を伝える．そしてこれらの処理を誤差が十分小さくなるまで繰り返す．

Algorithm *Procedure of Parallel Alignment*

```
/* Check correspondence of all pair of */
/* the range images */
CreatePairTable;
/* Create the list of the calculation */
/* for each processor */
CreateCalculationList:
while(error > threshold){
  /* Client Process*/
  for(i = 0; i < nImage; ++i)
    for(j = 0; j < nImage; ++j)
      if(List[i][j]){
        CorrespondenceSearch(i, j);
        CalculationEachMatrix(i, j);
      }
  /* Server Process */
  CalculationMatrix(all);
  /* Server & client process */
  UpdatePosition;
}
```

各クライアントが計算する距離画像対リストは，反復計算の初めに一度だけ計算するため，各クライアントの計算量と必要とするメモリ量が反復計算によって大きく変化することは無い．

4. 効率評価

本研究では 8 台構成の PC クラスタを使用した．各 PC は 2 つの PentiumIII800MHz プロセッサと 1GB のメモリを搭載している．使用したデータは鎌倉大佛を測定した 15 枚の距離画像と，奈良大佛を測定した 114 枚の距離画像である．これらの距離画像はともに Cyrax2400[17]によって測定されたものである．本章では，この PC クラスタ上での本手法の効率を計算時間とメモリ使用量という点から評価する．

4.1 計算時間効率

まずは本手法の計算時間効率を評価する．ここで計算時間とは 1 回の反復計算に必要な時間として評価する．評価に用いたデータは鎌倉大佛を Cyrax2400 で測定した 15 枚の距離画像である．これらの距離画像の平

均頂点数は300,000, 平均メッシュ数は600,000である. Figure 4.1 は初期姿勢とアライメント後の距離画像を示している.

Figure 4.2 はプロセッサ数に対する計算時間効率を表している. 横軸はプロセッサ数で, 各プロセッサ数に対する計算時間は1プロセッサによる計算時間との比で表されている. このグラフにより, プロセッサ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. ちなみに 1プロセッサによる計算時間は平均55.1秒, 10プロセッサによる計算時間は8.0秒である.

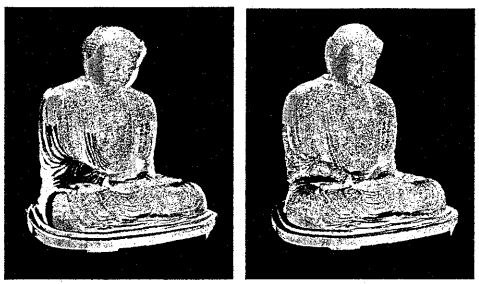

Figure 4.1 Initial position and alignment result

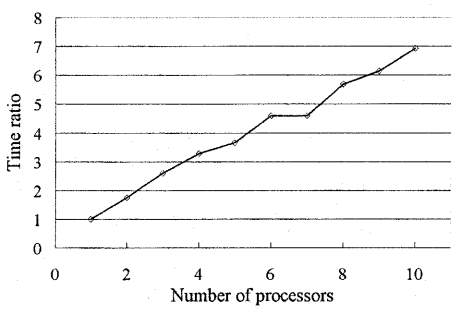

Figure 4.2 Time performance

4.2 メモリ使用効率

次にメモリ使用量についての評価を行う. 使用したデータは鎌倉大佛の距離画像 15 枚と奈良大佛の距離画像114枚である. 3.2節で示したように, 使用するメモリ量は読込む距離画像枚数に大きく影響される. そこでデータ分散の効率評価は, 各プロセッサに対して割当てられる最大距離画像枚数によって考える. この場合, 実際の位置推定の計算を行うことなく距離画像対のリストからメモリ使用量の効率化に関する評価を行うことができる.

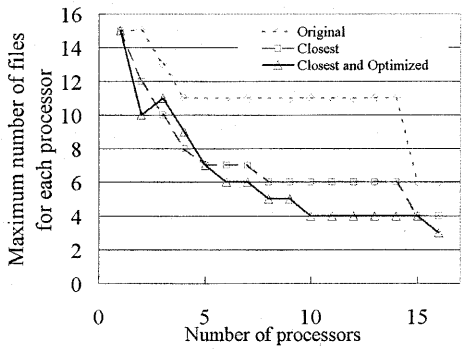

Figure 4.3 Memory performance(Kamakura)

Figure 4.3 は鎌倉大佛の距離画像を並列化した場合の, 各プロセッサ数に対する読込み最大画像枚数である. この時の角度閾値は60°とした.「Original」はバウンディングボックスの重なりと角度閾値の条件のみで冗長する距離画像対を除去した場合である. また「Closest」は更に隣接状態を考えた場合である. そして「Closest and Optimized」は隣接状態を条件に入れた上で, シーン画像の並びを最適化した場合である. このグラフより「Closest and Optimized」が, もっとも効率良くデータを分散させられる事がわかる.

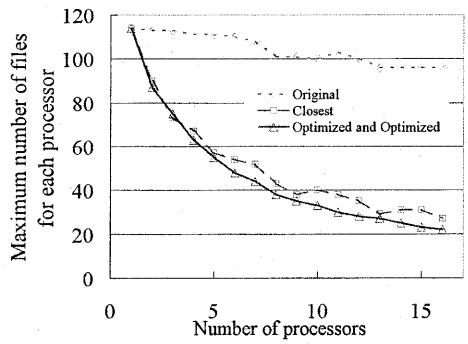

Figure 4.4 Memory performance(Nara)

また Figure 4.4 は奈良大佛を測定した距離画像に適用した場合であるが, 鎌倉大佛の場合と同様に, 隣接状態の条件を考え, シーン画像の並びを最適化した場合に最も安定して効率化が図れることが分かった.

5. 実験結果

では実際に奈良大佛を Cyrax2400 で測定した距離画像群をアライメントした結果を示す. この奈良大佛を測定した距離画像の総画像枚数は114枚であり, 平均頂点数は約320,000, 平均メッシュ数は約607,000である. 全頂点数は36,299,982点であり, 通常の計算機で

は表示することも難しい．これらの距離画像を 8 台構成の PC クラスタを用いて，8 並列及び 16 並列で同時アライメントを行った．Table 5.1 はこのときの反復計算一回に掛かった時間の平均と，全プロセッサ中の最大メモリ使用量及び最小メモリ使用量を表している．

Mach/Proc	Time(s)	Max Mem(MB)	Min Mem(MB)
8/8	104	833	583
8/16	82	871	653

Table 5.1 Total performance

8 プロセッサと 16 プロセッサの場合でメモリ使用量にあまり差がないのは，複数 CPU を持つ PC ではマルチスレッドによって並列計算を行っているために，ともに 1 台の PC で使われる最大メモリと最小メモリを表しているためである．またプロセッサ数が 2 倍になっているにも関わらず，計算時間がほとんど変わらないのは，サーバ側で行う変換行列の計算時間が大きくなっているためである．実際にサーバ側の計算に要する時間は平均で 42 秒であり，各クライアントの計算時間の方が小さくなっている．最大メモリ使用量と最小メモリ使用量に大きな差が見られるのは，並列化が各ノードに対して均等に行われていないためだと考えられる．

Figure 5.1 に奈良大佛のアライメント結果を示す．全ての距離画像を表示する事はできないので，一部のデータを示す．20 回の反復計算に約 30 分の計算時間を要した．

Figure 5.1

6. おわりに

本論文では複数枚の距離画像を並列にアライメントする手法を提案した．計算時間とメモリ使用量という二つの点を考慮して，これらの点に関して効率の良い並列化手法を示した．また本手法を用いて PC クラスタ上で大規模な距離画像を同時アライメントすることで，本手法が有効性を確認した．

今後の課題としては以下の点が挙げられる．

1. 各距離画像の頂点数，メッシュ数を考慮して並列化を行う．本論文中では各距離画像の頂点数及びメッシュ数はほぼ等しいものと仮定しているが，実際には大きく異なる場合もあるので，これらを考慮してより効率の良いデータ分散を行う必要がある．
2. 本手法適用による収束や精度への影響について調べる．並列化に際して，各距離画像間の冗長と思われる関係を除去しているので，この事によるアライメントの収束及速度や精度にどのような影響があるのかを調べる必要がある．
3. 変換行列計算部分を高速化する．実験結果からも分かるように，距離画像枚数が多く，十分な並列化数で計算した場合は，相対的にクライアント側よりもサーバ側で行っている変換行列計算に多くの計算時間が必要となる．そこで，この変換行列計算を高速行う手法を考える必要があると考えられる．

謝辞

本研究は科学技術振興事業団戦略的基礎研究推進事業（CREST）高度メディア社会の生活情報技術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徳院，奈良東大寺の方々に感謝の意を申し上げます．

References

[1] D. Miyazaki, T. Ooishi, T. Nishikawa, R. Sagawa, K. Nishino, T. Tomomatsu, Y. Takase, K. Ikeuchi, The Great Buddha Project: Modelling Cultural Heritage through Observation, In *Proc. International Conference on Virtual Systems and Multimedia (VSMM)*, pp.138-145, 2000.

[2] M. Levoy et. al. The Digital Michelangelo Project. In *Proc. SIGGRAPH 2000*, pp.131-144.

[3] 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.

[4] Y. Chen and G. Medioni, Object modeling by registration of multiple range images, *Image and Vision Computing* 10(3), 1992, 145-155.

[5] Blais, G. and Levine, M. "Registering Multiview Range Data to Create 3D Computer Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 8, 1995.

[6] 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.

[7] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[8] David A. Simon, Martial Hebert, and Takeo 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.

[9] Langis, C., Greenspan, M., and Godin, G.. 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.

[10] 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.

[11] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. To-wards a general multi-view registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):540–547, May 1996.

[12] R. Benjemaa 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.

[13] Greg Turk and Marc Levoy. Zippered polygon meshes from range images. In *Proc. SIGGRAPH* '94, pp. 311-318. ACM, 1994.

[14] Curless, B., Levoy,M., A Volumetric Method for Building Complex Models from Range Images, In *Proc. SIGGRAPH* '96, ACM, 1996, pp. 303-312.

[15] M. Wheeler, Y. Sato, and K. Ikeuchi, Consensus surfaces for modeling 3D objects from multiple range images, In *Proc. IEEE International Conference on Computer Vision*, pp.917-923, Jan, 1997.

[16] R. Sagawa, K. Nishino, K. Ikeuchi, "Robust and Adaptive Integration of Multiple Range Images with Photometric Attributes", In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.2, pp.172-179, 2001

[17] http://www.cyra.com.