

修士論文

高精度レンジセンサーを用いた  
3次元モデルの構築

平成12年2月4日提出

指導教官

池内 克史 教授

東京大学大学院 工学系研究科 電子情報工学専攻

86453 西川 拓

# 目次

<b>1</b>	<b>序論</b>	<b>1</b>
1.1	はじめに . . . . .	2
1.2	本論文の目的 . . . . .	3
1.3	本論文の位置づけ . . . . .	6
1.3.1	およその位置関係の推定 . . . . .	7
1.3.2	正確な位置関係の推定 . . . . .	7
1.4	本論文の構成 . . . . .	8
<b>2</b>	<b>以前のアルゴリズムとの比較</b>	<b>9</b>
2.1	はじめに . . . . .	10
2.2	おおまかな位置あわせ . . . . .	10
2.3	正確な位置あわせ . . . . .	11
<b>3</b>	<b>Simultaneous Registration の概要</b>	<b>13</b>
3.1	レンジセンサとその実際 . . . . .	14
3.2	位置あわせの流れ . . . . .	14
<b>4</b>	<b>おおまかな位置あわせ</b>	<b>20</b>
4.1	はじめに . . . . .	21
4.2	おおまかな位置あわせの限界 . . . . .	22
4.2.1	測定誤差 . . . . .	22
4.2.2	誤差の伝搬 . . . . .	23
4.3	おおまかな位置あわせの流れ . . . . .	25

---

4.4	変換行列の導出 . . . . .	26
4.4.1	回転行列の導出 . . . . .	28
4.4.2	並進行列の導出 . . . . .	29
4.5	実験結果 . . . . .	29
4.6	まとめ . . . . .	30
<b>5</b>	<b>全体位置あわせ</b>	<b>33</b>
5.1	はじめに . . . . .	34
5.2	全体位置あわせの流れ . . . . .	35
5.3	対応点探索 . . . . .	37
5.4	全体位置あわせの解法 . . . . .	37
5.5	非線型最小二乗解の解法 . . . . .	39
5.6	誤差式の線形化 . . . . .	40
5.7	実験結果 . . . . .	41
<b>6</b>	<b>結論</b>	<b>48</b>
6.1	まとめ . . . . .	49
6.2	今後の課題 . . . . .	50
	参考文献	51

## 目 次

1.1	位置あわせ後の画像と統合後の画像 . . . . .	4
3.1	Cyrax レンジセンサの外観 . . . . .	15
3.2	Cyrax センサから得られるデータ例 . . . . .	16
3.3	CCD 画像と Cyrax から得られる輝度情報を含んだ画像の例 . . . . .	19
4.1	初期位置あわせ変換行列の伝搬 . . . . .	24
4.2	初期位置あわせ変換行列の導出 . . . . .	25
4.3	初期位置あわせの変換行列の導出 . . . . .	27
4.4	座標系の変換 . . . . .	28
4.5	初期位置あわせ前の画像 . . . . .	30
4.6	初期位置あわせ後の画像 . . . . .	31
4.7	12 枚の距離画像をおおまかな位置あわせした後の画像 . . . . .	32
5.1	対応点探索 . . . . .	38
5.2	対応点の処理 . . . . .	39
5.3	全体位置あわせ後の画像 (正面) . . . . .	42
5.4	全体位置あわせ後の画像 (背面) . . . . .	43
5.5	全体位置あわせ後の画像 (上部) . . . . .	44
5.6	全体位置あわせ後の画像 (背面) . . . . .	45
5.7	全体位置あわせ後の画像 (背面) . . . . .	46
5.8	全体位置あわせ後の画像 (背面) . . . . .	47

## 表 目 次

1.1	位置あわせの手法の比較 . . . . .	5
3.1	Cyrax レンジセンサの仕様 . . . . .	14
4.1	おおまかな位置あわせと全体位置あわせの相違点 . . . . .	22

# 第 1 章

## 序論

この章では、本論文の目的と位置づけ、そして以下の構成を述べる。

## 1.1 はじめに

本論文は、高精度な 3 次元スキャナを用いて、一般的に大型でそして複雑な形状をした物体の 3 次元形状を得ることをその目的としている。

日本の古い文化財をはじめ、世界の文化遺産は、常に自然災害、風化の危険にさらされている。こうした文化財をなんらかの形で保護する活動は必要であるが、本論文は、そういった活動に一つの解を与えるものだと考える。万が一の場合に備えて文化財の形状を保存しておけば、形状を復元することが可能になる。

また近年、歴史的建造物に加えて、直線あるいは平面の集合では表せない複雑な形状をした建築物のデジタル保存の需要が高まっている。一般のビルや家屋だとその形状の単純さゆえ、人手による測定の結果を CAD システムに落とし込んでその形状を電子保存することもできるが、複雑な形状をした建築物ではそれは難しい。本論文では高精度なレンジセンサを用いて、こうした複雑な建造物の 3 次元モデルを作成することをテーマとしている。以上のような用途のためには通常、物体を一方向から見た 3 次元データだけでなく、全周囲から見た完全な 3 次元データが必要とされる。対象とする物体の形状が複雑な場合には、そのような完全な 3 次元データを得ることは難しく、それを克服することが課題となる。

このような実世界にある物体の 3 次元形状を得るためのセンサとして、レンジセンサが開発され、数多く実用化されてきた。レンジセンサとは広くセンサから対象物体表面までの距離を測定するためのセンサのことをいい、様々な種類のものが含まれる。例えば、火星の表面を無人で移動するロボットにおいて、視覚の役割を果たし、衝突を回避したり、外界の地図を作成する目的で利用されている。また産業用のロボットの視覚として働き、工場において機械部品を識別するために利用されている。さらに人の姿勢を矯正するため、人体形状を計測し、これを検診に役立てるためにも利用されている。このような様々な用途に応じて、様々な種類のレンジセンサが使われている。

最初に述べたような用途で使われるレンジセンサは、その多くが画像センサが組み込まれたもので、画像の各画素に対象物体までの距離を格納するようにして、3 次元形状のデータを得るようなものである。「画像」とは一般に、そこに写る情景の輝度が各画素に格納されたものである。これとの対比から、レンジセンサが得る物体までの距離を各画素に格納したような画像を距離画像と呼ぶ。

このような一般的なレンジセンサを用いて物体形状を計測したとき、得られる形状データ、すなわち距離画像は、当然ある 1 方向から見た部分的な形状のデータである。また、複雑な形をした物体では、形状のうち凸な部分の陰になって計測できない部分があることもある。そのため、全周囲から見て完全な 3 次元形状データを得るためには、物体に対して様々な角度・視点からレンジセンサで計測することを繰り返し、得られた複数の形状データを 1 つにまとめることで、全周囲からの完全なデータを得ることになる。

複数の視点からの距離画像を 1 つにまとめ、完全な形状データを得るためには、次のような 2 つの問題を解決しなければならない。

1. 物体を計測した複数の視点間の位置関係を正確に計算する
2. 重複する形状の同一部分の 3 次元データを一元化して、過不足のない形状データを得る

それぞれ順に、距離画像の、位置あわせ (alignment) の問題、統合 (merging) の問題、と呼ばれる。図 1.1 に示すのは、対象物体として「まねき猫」を様々な角度から計測した距離画像を位置あわせした画像 (同左)、とその同一部分のデータを一元化した画像 (同右) である。

第 2 節では本論文の目的、すなわち本論文の意図するところ、を述べる。

第 3 節では本論文の位置づけを述べる。ここでは、本論文に関連した研究で、以前から行われているものとの相違点を示す。

第 4 節では本論文の以下の構成を述べる。

## 1.2 本論文の目的

この 2 つの問題、すなわち位置あわせ、統合のうち、本論文では、位置あわせの問題を考える。これは上述のように、複数の視点からそれぞれに得た物体の 3 次元形状データを重ね合わせるために、個々の視点の位置関係を求めることである。距離画像とは、各視点固有の座標系で表された物体形状の数値データである。この、視点に固有の座標系で表された距離画像を、統一された 1 つの座標系において表すことで、それぞれの視点で得た 3 次元形状データを重ね合わせることができる。

もし、全ての距離画像が 6 自由度 ( $x, y, z$  の並進と  $x, y, z$  軸回りの回転) の正確なモーショ

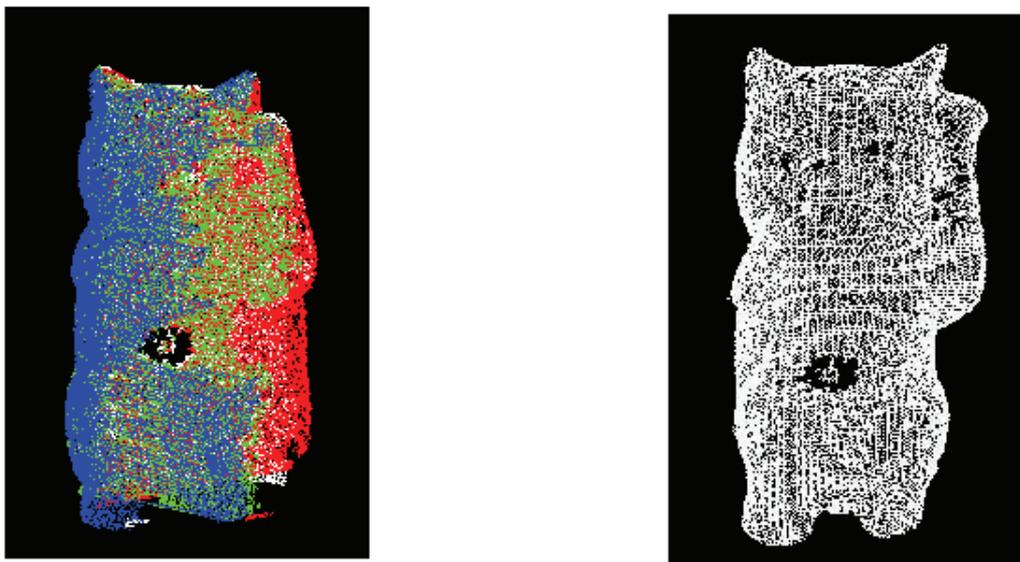


図 1.1: 位置あわせ後の画像 (左) と統合後の画像 (右)

ンデバイスで取得されたなら、画像情報を全く用いることなく位置あわせに必要な情報を全て得ることができる。しかし、本論文の目的は、こうした高価なデバイスを使用することなく、観測点やスキャンの方向に依存しないシステムを作り出すことである。

一般に複数枚の距離画像の位置あわせを行うためには、次のような2つの手法があり得る。

1. 正確に位置決めのできる装置を用い、センサないし物体を回転させながら距離画像を取得することで、視点間の位置関係を画像の取得時に得てしまう
2. レンジセンサを3次元空間中で自由に動かし、物体形状を多数の方向から計測し、得た3次元データだけから計測した方向自体を知る

まず1つ目の方法を考える。これは形状データ取得時に、視点の位置情報を外部の装置で得てしまおうとするものである。例えば、ターンテーブルを用い、その上に物体を載せて、これを回転させながら固定したレンジセンサから形状データを取得することが考えられる。この時、たとえターンテーブルの回転角を十分よく制御できても、センサとターンテーブルの位置関係が正確に分らなければ、回転軸が正確には算出できない。この場合

には、センサとターンテーブルの正確な位置を求める面倒な校正を行わなければならない。また他の手段として、レンジセンサをロボットハンドの先端に搭載し、ロボットハンドを 3 次元空間で動かしながらデータを得ることも考えられる。このような場合にも、ロボットの位置決め精度が十分でないために、ロボットの動きから得られた座標系間の位置関係を用いて距離画像を単一の座標系で表しても、うまく重ならないことがある。このように、ターンテーブルやロボットハンドなどで視点の位置を得ようとするときでも、精度の面で満足できないことがある。

次に 2 つ目の方法について考える。これは、視点間の位置関係を、得た形状データ自体から推定しようとするものである。それには、異なる視点から物体表面の同じ部分に関するデータを取得しておき、その同一部分のデータを利用して、視点間の位置関係を求めることになる。この方法は、1 つ目の方法に比べてターンテーブルなど、精度の求められる装置が必要ないなど、自由度が高いことが魅力である。また 1 つ目の方法でも、上述のように精度の面で満足できないこともあり、その場合には、結局この 2 番目の方法を適用した方がよいこともある。

上述の 2 つの位置あわせの手法の比較を表 1.1 に示す。

表 1.1: 位置あわせの手法の比較

	モーションデバイスの使用	画像情報から位置あわせ
位置あわせに要する時間	スキャンと同時	時間を要する
位置あわせの精度	あまり精密でない	手法により精密にできる
費用	モーションデバイスが高価	レンジセンサのみ

前述のように、本論文の目的は、高精度な 3 次元スキャナを用いて、複雑な形状をした物体のできるだけ正確な 3 次元形状を構築することである。すなわち、できる限り精度の高いデータを用いてできる限り精度の高い位置あわせを行うことが必要となる。そこで本論文では、この 2 つ目の方法を中心に考える。つまり、視点間の位置関係が全く分からない距離画像について、位置を正確に合わせることを目的とする。ここでは位置あわせの処理を次のような 2 つの段階に分ける。

1. 視点の位置が全く分からない複数の距離画像からおよその視点間の位置関係を概算する
2. そのおおよその位置関係の分かった距離画像を、さらに正確に合わせる

の 2 段階である。特に 2. の段階は上述のターンテーブルやロボットハンドを利用して視点の位置情報を獲得した場合の、さらなる精度の向上にも用いることができる。

1. では、異なる視点から共通に観測されるような対象物体の大域的な特徴を取り出し、これを用いる。ここでは、以下の 2 点が重要になる。

- 対象の形状をできるだけ限定しない
- そのような大域的な特徴を選ぶ

もちろん、「おおよそ」といっても可能な限り正確な位置関係が得られることが望ましい。

2. では、より精度の良い位置あわせのために、1. のように何か特徴を取り出すのではなく、距離画像の形状データ自体を最大限有効に活用することを考える。このとき問題になるのは、

- 得られるデータが、対象面の連続な形状データではなく、センサの視線方向特有の基準で量子化されたデータである
- 異なる視点からの距離画像には、両方の視点から得られたデータが全ての領域で重複しているわけではない

ことである。

### 1.3 本論文の位置づけ

位置あわせに関する研究は以前から行われているが、それは以下のように大きく 2 つに分類することができる。

2 枚の画像の位置あわせを行う際に、

- 2 枚の画像の対応点を数点求めて、その情報のみを用いて位置あわせを行う手法

- 2 枚の画像のオーバーラップ部分全ての点の情報を用いて、多数の対応点情報から位置あわせを行う手法

前者は、一般的には変換行列を求める際の方程式が線形となるため計算が容易である。しかし、2 枚の画像の視点位置情報を求める際に、数点の 3 次元情報しか用いないため、外乱に弱い。

一方、後者は多数の対応点の組を求めるため外乱に強くなる。しかし、一般には非線形方程式を解かねばならないため、計算が難しい。また、データ量が多くなるに従って計算時間の短縮とメモリ管理が重要になる。

上述のように、本論文では、視点の位置が全く分からない距離画像系列からおよその位置関係を得ることと、およその位置関係の分かった画像に対し正確にその位置を合わせより精度の高い形状データを得ること、を分けて考える。

### 1.3.1 およその位置関係の推定

視点間の位置関係が全く分からない距離画像系列から位置関係を全く自動で行うことは難しい。以前からよく行われているのは、マウスで画像を動かしておおまかに合わせるという手法である。具体的には、注目している 2 枚の画像を同じビューアに表示し、片一方の画像を止めた状態で、もう一方の画像をマウスで動かし近づける。しかし、こういったインターフェースに慣れるのは非常に時間がかかるので、この処理を短時間で行うには非常に熟練を要する。

本論文では、注目している画像間でのなんらかの特徴点を数点選びだし、その情報をもとにおおまかに位置あわせするという手法を採った。この手法は、2 枚の画像の対応点の情報を全て使うのではなく、数点の情報のみを使用することから、外乱には影響されやすいが、非常に短時間でおおまかな位置あわせができる。

### 1.3.2 正確な位置関係の推定

おおまかな位置あわせが施された後の画像系列に対して正確な位置関係を求めるための論文はこれまでも数多くなされてきた。しかし、それらの多くは、2 枚ずつの位置あわせの集積として全体の位置あわせを行うため、位置あわせの際の誤差が蓄積し、処理する画

像の枚数が多くなるに従って、位置あわせが非常に難しくなる。本論文では、3枚以上の画像系列に対しても一度に位置あわせを行うため、誤差が一ヶ所に集中するということがなく、非常にロバストな位置あわせが行える。

## 1.4 本論文の構成

第 2 章では、距離画像の位置あわせについて、これまでになされてきた研究について列挙する。

第 3 章では、実験で用いた Cyrax レンジセンサの仕様や Cyrax レンジセンサが与えるデータについて、また本論文で用いる位置あわせの手法についてのおおまかな概要を述べる。

第 4 章では、全体位置あわせの前段階としての、大まかな位置あわせの手法について述べる。全体位置あわせはそのアルゴリズムの性格上、ある程度の位置あわせが行われていることが前提条件となっているため、この大まかな位置あわせは必要不可欠なルーチンである。また全体位置あわせにおける収束時間の短縮のため、可能な限り正確な位置あわせを行うことが望ましい。

第 5 章では、本論文のメインとなる全体位置あわせのアルゴリズムについて述べる。以前のアルゴリズムでは、常に 2 枚ずつの位置あわせを集積することによって対象物体全体の位置あわせを行っていたため、誤差の蓄積が生じてしまい、最初と最後の距離画像がうまく張り合わなくなることが多かった。そこで、画像系列全体を一度に位置あわせするアルゴリズムを用いることによって、対象物体全体の位置あわせをする際にも誤差が一ヶ所に集中しないようになった。また実際に鎌倉の大仏の距離画像データを用いてその実験結果をまとめる。

最後に第 6 章では、本論文を総括し、結論を述べる。

## 第 2 章

### 以前のアルゴリズムとの比較

この章では、距離画像の位置合わせに関する以前のアルゴリズムを列挙し、その優劣を比較する。

## 2.1 はじめに

これまでも複数方向からスキャンされた 3 次元的距離画像を位置あわせするための様々な研究がなされてきた．本節では、これまでに発表されている位置あわせのための技術を列挙してみたい．

本論文では、視点の位置が全く分からない距離画像系列からおよその位置関係を得ることと、およその位置関係の分かった画像に対し正確にその位置を合わせより精度の高い形状データを得ること、を分けて考える．

## 2.2 おおまかな位置あわせ

視点間の位置関係が全く分からない距離画像系列から位置関係を推定する問題について、これまでにいくつかの研究がある．まずこれについてまとめる．

基本的な方針は、距離画像を領域分割し、各領域から得られる大域的な特徴量の対応から位置関係を求める、というものである．これを行うために、位置関係が分からなくても対応関係が確立できるような、よい特徴量を選ぶことが 1 つの課題となる．なるべく多くの対象に対して適用でき制約が少ないこと、またできるだけ正確な視点間の位置関係が求められること、の 2 つが重要である．

河井は [1]、複雑でオクルージョンが生じ易く、不連続の多い物体 (例えば紫陽花の花のような物体) を対象にした手法を提案している．各視点での距離画像において、それぞれ連結性をもとに領域分割し、各領域の大きさや形などの特徴をもとに、異なる視点での距離画像において対応する領域を決定し、対応するとされた領域の重心が一致するように 2 つの視点間の座標変換を求める．さらに、各領域を小領域に分割し、完全に一致する小領域を選んで二次曲面を当てはめ、先に求めた変換を初期値として、ずれが最小になるように反復計算して精度よく視点間の変換を求めようとした．

Shum[2] らは、多面体を対象として、平面の対応を用いて位置関係を求めている．他の手法では、隣り合う二視点間で順番に位置関係を求めるものがほとんどであるのに対し、この手法では、全ての視点から得られたデータ全てを用いて一度に位置関係を求めている．平面の対応を用いる方法にはいろいろな利点があり、例えば、対応付けが容易であること

が挙げられる．また、平面の法線ベクトルは並進に対して不変であることから、位置関係を回転の成分と並進の成分に分けて別々に求められることも利点である．

## 2.3 正確な位置あわせ

次に、上に述べた方法で位置関係を得た場合や、なんらかの方法で位置決めしながら距離画像を得た場合などのように、視点間のおよその位置関係があらかじめ分かっているときの距離画像の重ねあわせについて考える．このような場合の手法として、様々なものが提案されている [3] が、ほとんどが Besl[4] の提案した ICP (Iterative Closest Point) アルゴリズムを基本とする [5-10]．本来、ICP アルゴリズムとは、物体の完全な形状が分かっている、その部分形状が別に与えられたとき、部分形状と元の形状との位置関係 (座標変換) を推定するために考案された手法である．そのアルゴリズムは、点データで与えられた 2 つの形状の「距離」を最小化するものである．距離とは、一方の形状の各データ点がある座標変換によって変換したとき、それらの点と元の物体形状で対応する点との距離の総和である．最初は 2 つの形状の間で、点と点の対応関係は分からない．そこで、変換を一度定めるごとに対応を推定しなおし、このプロセスを繰り返すことで、正しい対応関係を得て、その結果正しい変換を定めようとする．

ICP アルゴリズムは、前述のように、本来部分形状の姿勢推定のための手法であるので、この問題、すなわち異なる視点から観測した距離画像の位置あわせの問題に応用してはうまくいかないことが多い．よい結果が得られないのは次に挙げるような原因による．

- オクルージョンなどによって、1 つの視点でのデータがもう一方の視点でのデータに完全に含まれることがない
- 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
- 異なる視点でのデータの間で点同士の対応関係が分からない．
- それぞれの計測点は測定誤差を含んでいる

Zhang[5] が提案した距離画像の位置あわせのアルゴリズムは、考え方は ICP アルゴリズムと同じであるが、上に挙げた問題を解決するための工夫がなされている．この手法では、

1 枚目の距離画像の 1 つ目の計測点に対し、2 枚目の距離画像の計測点の中で最も近い点までの距離を考え、また、その逆の 2 枚目から 1 枚目の距離画像への同様な距離を考え、全計測点にわたるこれらの和の平均として、2 枚の距離画像間の距離を定義している。オクルージョンに対応するため、一番近い点までの距離がある値より遠い場合には、その点には対応点はないとしている。計測点が離散的にしか得られないことに対処するため、計測点がある一定間隔以上離れている場合には、計測点の間を線形補間して仮想的な計測点を設けている。また、一番近い点を探索する時間を軽減するため、k-d tree という木構造を用いている。

Chen[6] は、次のようにして 2 つの距離画像から視点間の位置関係を求めている。まず 1 枚目の画像の滑らかな部分から選んだ計測点における対象物体表面の法線と、2 枚目の画像の形状面の交点を求め、その点における接平面を求める。その接平面と、対応する 1 枚目の画像の計測点との距離を考え、これを 1 枚目の画像の計測点について求めるた総和を最小にするように位置関係を求める。2 つのデータの位置関係を、法線と接平面を用いてうまく計算しているが、なめらかな対象の雑音の少ない画像でないと適用できない。

誤対応を減らすために、計測点の位置以外の情報を用いる方法が提案されている。Dorai [11] は、Chen の方法での誤対応を減らすため、2 つの制御点の組を考え、1 枚目の画像の 2 点と 2 枚目の画像の 2 点に対応しているとした場合、1 枚目の画像での 2 点間の距離と 2 枚目の画像での 2 点間の距離がおおよそ等しいことを利用している。また、Fledmar[12] は、各点での法線方向も等しくなるように視点間の変換を求めている。Bergervin[7] は、Chen の方法の多視点への拡張である。隣り合う 2 視点間で順番に視点間の座標変換を求めるのではなく、1 つの視点を基準に定め、その基準点に対して他の視点への変換を求める。

Masda[9] は、ロバスト推定の手法である Least Median Squares の考え方をもとに、2 つの点データの集合の中で、少数の点をランダムに選び、視点間の座標変換の推定を繰り返し、正しい変換を得ようとしている。同様の推定方法で、点と三角パッチの対応を用いたものも提案されている。[10]

また、上述のように、位置あわせの済んだ距離画像に対し、重複して得られた部分の形状データを一元化するなどして過不足のない完全な形状データを得る、距離画像の統合や、距離画像を統合して得られた形状データから、何らかの形状の記述を得ることについても、多数の方法が提案されている。[13-17]

## 第 3 章

# Simultaneous Registrationの概要

この章では , Simultaneous Registration の概要を述べる .

### 3.1 レンジセンサとその実際

以下に実験で用いたレンジセンサの仕様を表 3.1 に示す.

表 3.1: Cyrax レンジセンサの仕様

精度	6mm
使用可能距離	100m(50cm - 50m を推奨)
スキャン密度 (最大)(鉛直方向)	2mm(@50m)
スキャン密度 (最大)(水平方向)	2mm(@50m)
スポットサイズ	6mm (0-50m)
視野 (鉛直方向)	40 度
視野 (水平方向)	40 度
可動範囲 (水平回転)	360 度
可動範囲 (鉛直回転)	180 度

Cyrax センサから直接得られるデータ (図 3.2 参照) は、まず縦横の解像度を示し、それ以下に Cyrax センサを原点とする各点の 3 次元座標と輝度値が格納されている. 各距離画像を位置あわせするためには、この生データにいくつかの加工を施す必要がある.

まず、Cyrax センサはスキャン時に決定したスキャン範囲 (二次元的には長方形) に等間隔にレーザを照射するため、対象物体以外の物体 (背景や他物体) のデータを取り除く必要がある.

また、距離画像の位置あわせの過程で各点の法線方向の情報が必要となる. そこで、Cyrax センサから得られた点データから各点の連結関係を計算し、メッシュを作り出さなくてはならない.

### 3.2 位置あわせの流れ

各レンジ画像のデータ加工が終わると、実際に位置あわせの作業に移る. 以下では位置あわせの大まかな流れを示す.

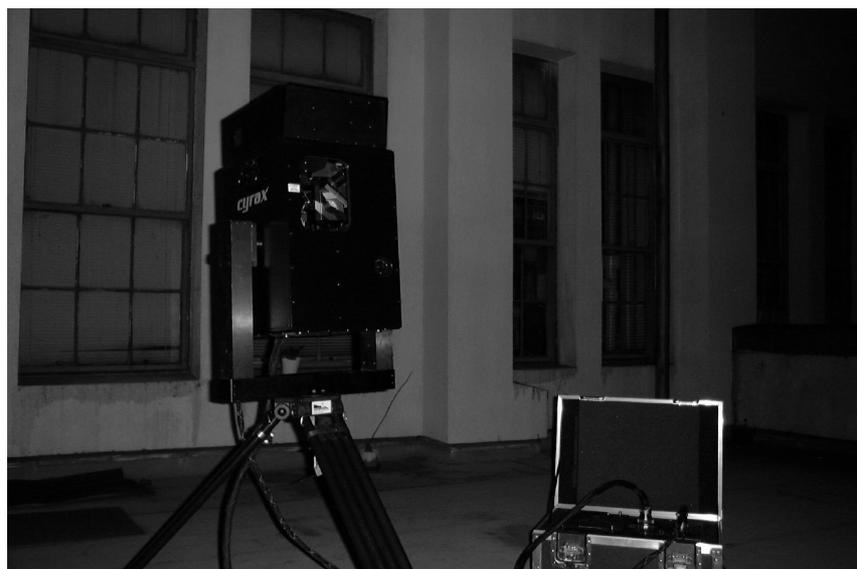


図 3.1: CyraX レンジセンサの外観

1. レンジセンサで様々な方向から物体をスキャンする
2. 得られた距離画像を適当な閾値を定めて背景除去する.
3. 点モデルをメッシュモデルにする.
4. 2 枚の距離画像間で、いくつかの対応点を見つけ、その対応点間の距離を最小化する. ここでの評価は 2 枚ずつの距離画像で行うため、全ての画像をこの評価だけで位置あわせするのは、あまりにも大まかな評価すぎる
5. 最小二乗法を用いた詳細な位置あわせを行う
6. 収束するまで 5. を繰り返す.

以下の各節では以上の各ステップの詳細を述べる.

物体のスキャン 3次元物体、特に凹凸のある形状、複雑な形状をした物体は、その全ての点をスキャンするためにはその対象物体を複数方向からスキャンしなければならない. しかも以下のように、その各々の距離画像における視点方向の相対位置を決定するためには、視点方向が隣り合う画像の間で、同じ点がある程度スキャンしている

300  
499  
-0.986192 -1.626142 -4.645115 -1649  
-0.987343 -1.624279 -4.642910 -1631  
-0.990014 -1.624116 -4.654691 -1687  
-0.989459 -1.617808 -4.652204 -1540  
-0.988476 -1.611276 -4.647734 -1603  
-0.992526 -1.613048 -4.666986 -1524  
-0.988908 -1.602317 -4.650339 -1633  
-0.991890 -1.602116 -4.664210 -1534

図 3.2: Cyrax センサから得られるデータ例

ことが必要である。これらのオーバーラップした点をもとにして、相対的位置関係を割り出すためである。隣り合う画像の間でのオーバーラップは多ければ多いほど位置関係の計算結果は正確となるが、その分、スキャンすべき枚数も多くなる。正確さと画像のデータ量とのトレードオフである。実際に用いたレンジセンサ、Cyrax はその画角<sup>1</sup> が約 40 度であることから、対象物体の重心を中心とし約 30 度おきにスキャンすればある程度のオーバーラップを隣り合った画像間で共有できると考えた。

背景除去 CyraX レンジセンサは、それに積まれた CCD カメラから撮影された画像 (正方形) の中で任意の四角形を選択してその四角形内をスキャンする仕組みになっている。すなわち、レンジセンサから発射されたレーザは、対象物体だけをスキャンするのではなく、その前や後ろに存在する物体をもスキャンしてしまう。従って対象物体以外の物体 (ここではそれを総称して背景と呼ぶことにする) を排除することが必要になってくる。対象物体よりも著しく後方あるいは前方にある物体はその depth をもとに背景除去することができる。しかし、視点方向からの depth が対象物体の depth とそれほど変わらない、あるいは対象物体の depth の最大値と最小値の間に含まれてしまうような物体を取り除くことは、以上の方法では無理である。本論文において

<sup>1</sup>写真レンズで撮影できる範囲がレンズ中心に張る角度

は、こういった点は、ビューアの上でそれらの点をマウスで選択して消すことができるようにした。

メッシュの生成 CyraX レンジセンサで得られるデータは離散点の 3 次元データである。しかし、以下の詳細な位置あわせの際に、各点の法線ベクトル情報が必要になる。従って、離散点の連結関係を考えてメッシュを生成しなければならない。本論文では、三角メッシュを生成した。

2 枚の画像の位置あわせ (おおまかな位置あわせ)  $n$  枚の距離画像  $S_0 \dots S_{n-1}$  の位置あわせの問題は、オーバーラップした部分における対応点間の距離を最小にする、いわゆる最小化の問題に帰着する。本論文で示す手法は、位置あわせの誤差を蓄積しないために全ての距離画像を同時に位置あわせする。そのため、全ての距離画像の大まかな相対的位置関係を始めに知っておかなければならない。

全ての距離画像は順々にワールド座標系に変換される。各距離画像とワールド座標系の大まかな位置関係は、新しい距離画像とそれまでに統合されてきた距離画像の少なくとも 3 つの対応点で定義される。必要な対応点はユーザが対話的に選択する。ここで、距離画像上の任意の点  $\vec{x}$  の、ワールド座標系での点  $\vec{x}'$  への (大まかな) 変換行列を  $T_i^{(\tau)}$  とする。すなわち、

$$T_i^{(\tau)} \vec{x} = \vec{x}' \quad (3.1)$$

複数枚の画像での同時位置あわせ (詳細な位置あわせ)  $\vec{\theta}_i$  を画像  $S_i$  のワールド座標系に対する相対的な方向ベクトルとする。方向ベクトル  $\vec{\theta}_i$  は 3 つの回転角と 3 つの平行移動パラメータの計 6 つのパラメータを持っている。回転行列  $R_i^{(\theta)}$  と平行移動ベクトル  $t_i^{(\theta)}$  によって方向ベクトル  $\vec{\theta}_i$  が定義できる。つまり、任意の点  $\vec{x} \in S_i$  のワールド座標系への変換は次のように  $T_i^{(\theta)}$  で定義できる。

$$T_i^{(\theta)} \vec{x} = R_i^{(\theta)} \vec{x} + t_i^{(\theta)} \quad (3.2)$$

さて、前のステップで定義された大まかな変換行列  $T_i^{(\tau)}$  と、それをより精度良く改良する変換行列  $T_i^{(\delta)}$  を用いて、

$$T_i^{(\theta)} = T_i^{(\delta)} T_i^{(\tau)} \quad (3.3)$$

と表されるとする. また、画像  $S_i$  と  $S_j$  の距離をマッチング距離  $D(S_i, S_j)$  と定義すると、問題は以下の  $\varepsilon$  の最小化問題に帰着する.

$$\begin{aligned}
\varepsilon^2 &= \min_{\theta} \sum_{i \neq j} D(T_i^{(\theta)} S_i, T_j^{(\theta)} S_j) \\
&= \min_{\delta} \sum_{i \neq j} D(T_i^{(\delta)} T_i^{(\tau)} S_i, T_j^{(\delta)} T_j^{(\tau)} S_j) \\
&= \min_{\delta} \sum_{i \neq j, k} (R_i^{(\delta)} \vec{n}_{ik}^{(\tau)} ((R_j^{(\delta)} \vec{y}_{ijk}^{(\tau)} + \vec{t}_j^{(\delta)}) - (R_i^{(\delta)} \vec{x}_{ik}^{(\tau)} + \vec{t}_i^{(\delta)})))^2 \\
&= \min_{\delta} \sum_{i \neq j, k} \|A_{ijk} \vec{\delta} - s_{ijk}\|^2
\end{aligned} \tag{3.4}$$

ここで、

$$s_{ijk} = \vec{n}_{ik}^{(\tau)} \cdot (\vec{x}_{ik}^{(\tau)} - \vec{y}_{ijk}^{(\tau)}) \tag{3.5}$$

$$A_{ijk} = \left( \underbrace{0 \dots 0}_{6i \times 1} \quad \underbrace{C_{ijk}}_{6 \times 1} \quad \underbrace{0 \dots 0}_{6(l-i-1) \times 1} \right) + \left( \underbrace{0 \dots 0}_{6j \times 1} \quad \underbrace{-C_{ijk}}_{6 \times 1} \quad \underbrace{0 \dots 0}_{6(l-j-1) \times 1} \right) \tag{3.6}$$

$$C_{ijk} = \begin{pmatrix} \vec{n}_{ik}^{(\tau)} \times \vec{y}_{ijk}^{(\tau)} \\ -\vec{n}_{ik}^{(\tau)} \end{pmatrix}^T \tag{3.7}$$

である.

これらの式変形及びその意味は、第 5 章で述べる.

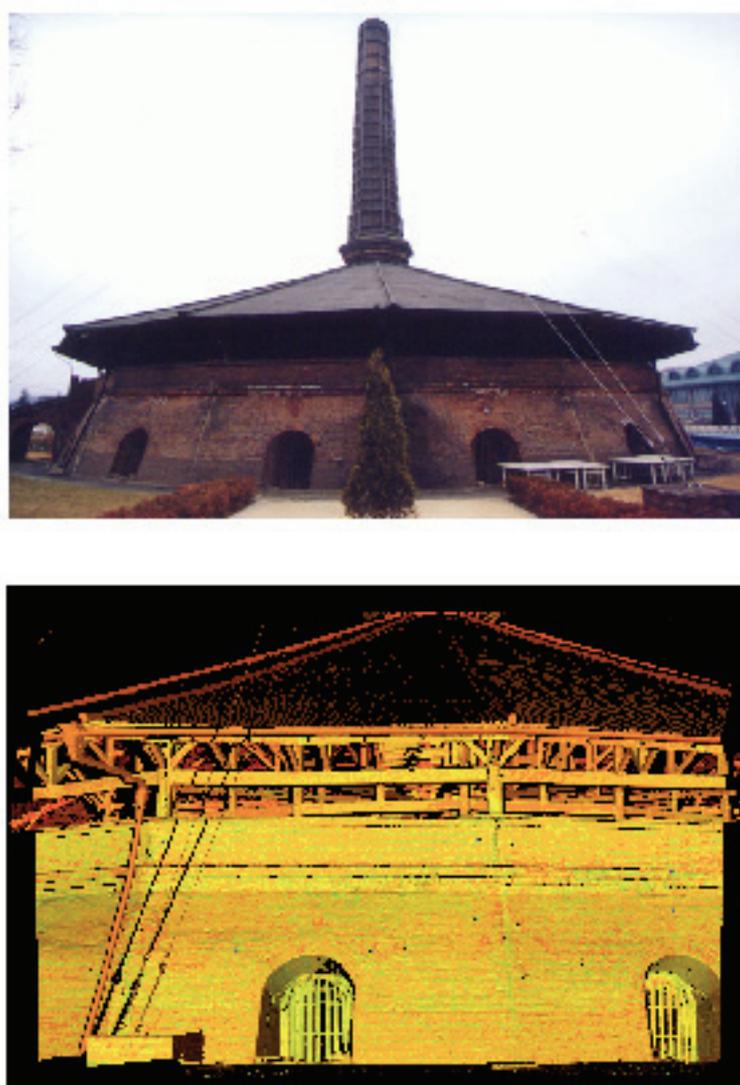


図 3.3: CCD 画像 (上) と Cyrax から得られる輝度情報を含んだ画像 (下) の例

## 第 4 章

### おおまかな位置あわせ

この章では、全体位置あわせの前段階としてのおおまかな位置あわせについて述べる。

## 4.1 はじめに

本章では、全体位置あわせの前段階としてのおおまかな位置あわせについて述べる。まず、いくつかの定義をしておく。2枚の画像を考えたとき、基準になり、移動させない画像を Model 画像、そして Model 画像の視点方向に合わせる画像を Scene 画像と定義する。

一般に、おおまかな位置あわせ、つまり初期位置あわせは、マウスを使って Scene 画像を Model 画像に近づけることによって行うことが多い。その場合には、6 自由度 ( $x, y, z$  方向の並進と  $x, y, z$  軸中心の回転) の動きをマウスで制御できるようにしておき、マウスの動きに対応する変換行列を常に保持しておく。したがって、各画像をどんな方向に動かしても、その変換行列を保持しているので、変換後の画像を保存できる。

しかし、マウスによって、おおまかな位置あわせを行うことは実際にはそれほど容易なことではない。おおまかな位置あわせを行うための巧みなインターフェイスを作るのはそれほど容易ではないし、そういったインターフェイスが実現されたとしても、かなりよい精度の初期位置あわせをするのには大変な時間がかかるであろう。

まずおおまかな位置あわせの必要性について述べる。それは、大きく以下の 2 点に要約される。つまり、

- 全体位置あわせのアルゴリズムは、大まかな位置あわせをした画像系列に対してのみ動作することが保証されているため、大まかな位置あわせを何らかの手法で行うことが必要
- 全体位置あわせの段階での収束時間は初期位置あわせの精度に比例する。したがってより精度のよい位置あわせをこの段階で行っておくことが望ましい

全体位置あわせの際には距離画像上の全ての点の情報を使うことによってロバスト性を高めるが、大まかな位置あわせにおいては、距離画像上の数点の情報を使い、その情報をもとに全ての点を変換する。もっとも、より精度のよい位置関係が得られることが望ましいが、精度よりも処理時間を優先する。(表 4.1 参照)

第 2 節では、おおまかな位置あわせの位置づけ、すなわち、位置あわせを 2 段階えにしなければならない理由について述べる。

第 3 節では、おおまかな位置あわせの概略と流れについて説明する。

表 4.1: おおまかな位置あわせと全体位置あわせの相違点

	おおまかな位置あわせ	全体位置あわせ
精度	おおまか	細かい
処理に用いるデータ	数点の 3 次元情報	距離画像全ての点の 3 次元情報
対応点探索	手動	自動
処理時間	速い	対応点探索に時間を要する

第 4 節では、2 つの距離画像間の視点方向を一致させるための変換行列を導出する手法を述べる。

第 5 節では、実際に鎌倉の大仏の 3 次元データを用いたおおまかな位置あわせの実験結果を述べる。

第 6 節では、本章のまとめとして、おおまかな位置あわせの問題点を述べ、それに続く全体位置あわせへの布石とする。

## 4.2 おおまかな位置あわせの限界

本節では、おおまかな位置あわせで正確な位置あわせが達成できない理由、すなわち、おおまかな位置あわせの限界について記述する。

おおまかな位置あわせで正確に位置関係が導出できない理由として、大きく

- レンジセンサ自体の限界、すなわち測定誤差に起因する不正確な変換行列の生成
- 上の誤差の伝播

の 2 つが考えられる。各々について詳しく見ていく。

### 4.2.1 測定誤差

まず、レンジセンサ自体の限界、すなわち距離データ自体の限界について考える。

レンジセンサから得られる 3 次元データは、センサの視線方向特有の基準で量子化されたデータである。従って、オーバーラップのある画像 2 枚の画像間においても、全く同一の点が 2 枚の画像間でスキャンされていることはほとんどありえない。

前述のように、おおまかな位置あわせでは、Model 画像と Scene 画像から対応する 3 点を手動で選び、その 3 点で作られる 3 角形を一致させるための変換行列を求めるが、対応する点を選択する場合のインターフェースとして、大きく以下の二つが考えられる。すなわち、

1. マウスをクリックしたピクセルで点を決定する
2. 3 次元データ上の”点”をピックできるようにする

1. の選択方法の場合、3 次元座標値 (一般に浮動小数点数) を整数値として認識してしまうため、値の丸め、が行われてしまい、正確な座標値を選択することができない。

2. の選択方法の場合は選択した点が正しければ、正しい 3 次元座標値を返してくれる。しかし、上の理由により、選択された点はほとんどの場合、正確な対応点を選んではいない。

前述のように、全体位置あわせの場合は、基本的に距離画像の全ての 3 次元座標データを用いて位置関係を求めるため、比較的外乱に強い。しかし、本章のおおまかな位置あわせでは、全データのうち、3 点の 3 次元データしか用いないため、そのデータに含まれる誤差は、それによって得られる結果に非常に大きな影響を及ぼす。

#### 4.2.2 誤差の伝搬

ここでは、上の誤差が複数回のおおまかな位置あわせの段階で伝搬していく問題について説明する。

大まかな位置あわせでは、異なる視点から共通に観測される部分の情報を用いて位置情報を得るため、観測部分に全く共通部分を持たないような視点間の画像を位置あわせすることはできない。例えば、大仏の正面からスキャンした距離画像と背面からスキャンした距離画像では共通部分を全く持たないため、位置あわせを行うことができない。

しかし、観測部分に全く共通部分を持たない視点間の画像であっても、それぞれの画像

と共通部分を持つ距離画像との位置関係を計算することによって、すなわち間に距離画像を仲介することによって、それらの位置関係を計算することができる。(図 4.1 参照)

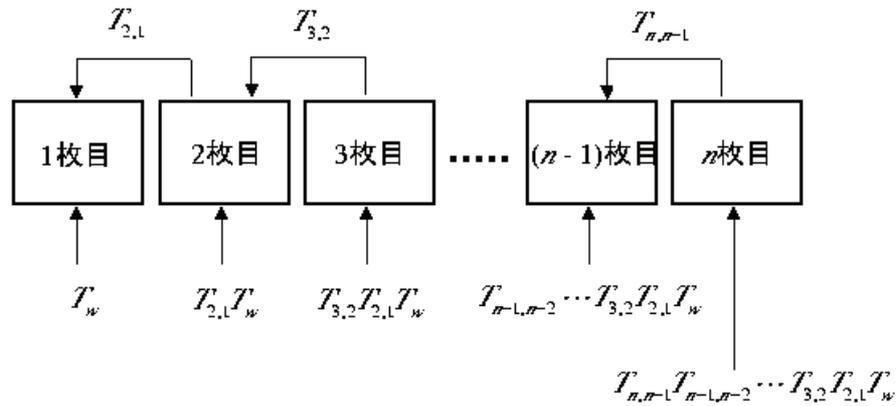


図 4.1: 初期位置変換行列の伝搬

図 4.1 のように  $n$  枚の画像  $S_1, S_2, \dots, S_n$  の添え字の若い方から順にワールド座標に変換するものとする。すなわち、1 枚目の画像  $S_1$  をワールド座標に変換するための変換行列を  $T_w$ 、また、 $i$  枚目の画像  $S_i$  の視点方向を  $i - 1$  枚目の画像  $S_{i-1}$  の視点方向に一致させるための変換行列を  $T_{i,i-1}$ 、またワールド座標系に変換後の画像をそれぞれ  $S'_1, S'_2, \dots, S'_n$  と表すとすると、 $S_1$  から  $S_n$  までの全ての画像の座標系をワールド座標系に一致させるための変換  $T_w, T_2, \dots, T_n$  は以下のように記述できる。(図 4.2 参照)

$$\left\{ \begin{array}{l} S'_1 = T_w S_1 \\ S'_2 = T_2 S_2 = T_{2,1} T_w S_1 \\ S'_3 = T_3 S_3 = T_{3,2} T_{2,1} T_w S_1 \\ \vdots \\ S'_{n-1} = T_{n-1} S_{n-1} = T_{n-1,n-2} T_{n-2,n-3} \dots T_{2,1} T_w S_1 \\ S'_n = T_n S_n = T_{n,n-1} T_{n-1,n-2} \dots T_{2,1} T_w S_1 \end{array} \right. \quad (4.1)$$

ここで、誤差の伝搬という問題がおこる。

すなわち、 $i$  枚目の画像  $S_i$  の視点方向を  $i-1$  枚目の画像  $S_{i-1}$  視点方向に一致させるための正確な変換行列を  $\Delta T_{i,i-1} T_{i,i-1}$ 、また 1 枚目の距離画像をワールド座標系に正確に変換するための行列を  $\Delta T_w T_w$  とすると、 $i$  枚目の画像における変換行列の誤差  $\Delta T_i$  は、

$$\Delta T_i = \Delta T_{i,i-1} \Delta T_{i-1,i-2} \dots \Delta T_{2,1} \Delta T_w \quad (4.2)$$

となり誤差が蓄積することが分かる。また、式を見ると自明のように、誤差の蓄積は初期位置あわせをする画像の順序に因らない。

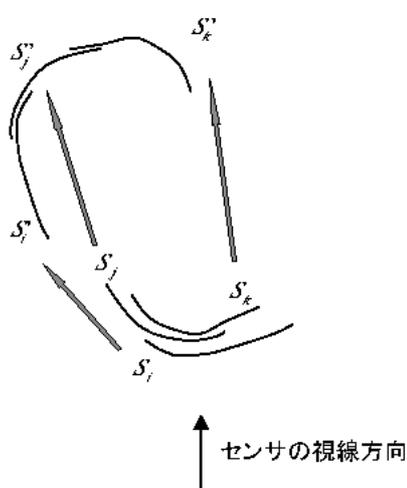


図 4.2: 初期位置あわせ変換行列の導出

### 4.3 おおまかな位置あわせの流れ

以下に、おおまかな位置あわせから全体位置あわせに至るまでの処理手順を簡単に示す。

1. 基準となる画像  $S_1$  をワールド座標に変換する。
2. オーバーラップを十分に持つ 2 枚の距離画像  $S'_i, S_j$  を選択する (ただし、 $S'_i$  は既にワールド座標に変換された画像、 $S_j$  はまだワールド座標に変換されていない画像とする)
3. 2 枚の画像  $S'_i, S_j$  の重なり部分に含まれる共通の特徴点を選択する

4. 3. で選択した 3 点から変換行列を求める
5. 4. で求めた変換行列を  $S_j$  上の全ての点に掛け合わせる
6. オーバーラップを有する全ての画像の組に対して 1. から 4. を繰り返す
7. 全ての距離画像のおおまかな位置あわせを終了する.
8. 以上の初期位置変換行列の情報をもとに全体位置あわせを行う

以上の説明からも分かるように、隣り合う視点からの距離画像の重なり部分は少なすぎでは特徴点をうまく採ることができないが、しかし重なり部分が多すぎても対象物体を一周スキャンするための回数、すなわち距離画像の枚数が多くなり、処理するデータ量が必要以上に多くなってしまふ。そこで経験的には、隣り合う視点の距離画像のオーバーラップは各々の画像の 30% から 40% 程度が適していると思われる。

おおまかな位置あわせでは、オーバーラップ部分を持つ 2 枚の距離画像において、そのオーバーラップ部分の、すなわち両方の距離画像に含まれる、明らかな特徴点を 3 点選びだし、その 3 点で作られる三角形がピッタリと張り合うような変換を求め、その変換を他の全ての点に掛け合わせることによって、おおまかに位置合わせを行おうというものである。

もちろん、手動で選択してきた 3 点が、並進運動と回転移動でピッタリと張り合うこと、つまり、選択してきた 3 点で張られる三角形が合同になることはないであろうから、ここで求める変換は近似的なものである。

その変換行列の導出法を次に示す。

## 4.4 変換行列の導出

図 4.3 にあるように、Model 画像上の選択した 3 点を  $M_1, M_2, M_3$ 、そしてその 3 点にそれぞれ対応する Scene 画像上の選択した 3 つの対応点を  $S_1, S_2, S_3$  とする。以下、 $\triangle S_1 S_2 S_3$  を  $\triangle M_1 M_2 M_3$  に変換する行列  $T$  を求める。まず Model 画像について考える。

$M_1$  から  $M_2$  へ向かうベクトルを  $\mathbf{u}_{12}$ 、 $M_1$  から  $M_3$  へ向かうベクトルを  $\mathbf{u}_{13}$ 、また 2 つのベクトル  $\mathbf{u}_{12}$  と  $\mathbf{u}_{13}$  のなす角を  $\theta_m$  とする。更に、 $\mathbf{u}_{12}$  と  $\mathbf{u}_{13}$  方向の単位ベクトルをそれぞれ  $\mathbf{n}_{12}, \mathbf{n}_{13}$

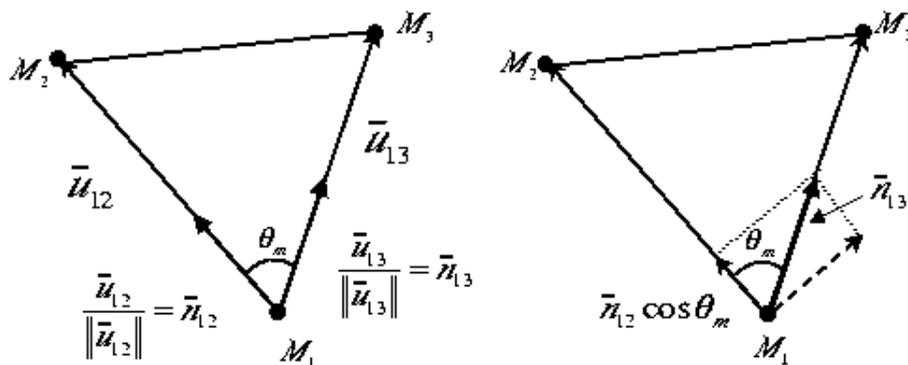


図 4.3: 初期位置あわせの変換行列の導出

とする. すなわち、

$$\mathbf{n}_{12} = \frac{\mathbf{u}_{12}}{\|\mathbf{u}_{12}\|} \quad (4.3)$$

$$\mathbf{n}_{13} = \frac{\mathbf{u}_{13}}{\|\mathbf{u}_{13}\|} \quad (4.4)$$

次に、2つのベクトル  $\mathbf{u}_{12}$  と  $\mathbf{u}_{13}$  で張られる平面内で、線形独立な2つの単位ベクトル  $\mathbf{m}_1, \mathbf{m}_2$  を定める. その1つを  $\mathbf{m}_1 = \mathbf{n}_{12}$  とすると、 $\mathbf{m}_2$  は、

$$\mathbf{m}_2 = \frac{\mathbf{n}_{13} - \mathbf{n}_{12} \times \cos \theta_m}{\|\mathbf{n}_{13} - \mathbf{n}_{12} \times \cos \theta_m\|} \quad (4.5)$$

となる. 更に、2つのベクトル  $\mathbf{m}_1, \mathbf{m}_2$  の両方に垂直な単位ベクトルを  $\mathbf{m}_3$  とすると、

$$\mathbf{m}_3 = \mathbf{m}_1 \times \mathbf{m}_2 \quad (4.6)$$

ここで、生成された3つのベクトル  $\mathbf{m}_1 = (m_{1x}, m_{1y}, m_{1z})^T$ ,  $\mathbf{m}_2 = (m_{2x}, m_{2y}, m_{2z})^T$ ,  $\mathbf{m}_3 = (m_{3x}, m_{3y}, m_{3z})^T$  は正規直交規定である.

同様にして、Scene 画像の正規直交規定ベクトル  $\mathbf{s}_1 = (s_{1x}, s_{1y}, s_{1z})^T$ ,  $\mathbf{s}_2 = (s_{2x}, s_{2y}, s_{2z})^T$ ,  $\mathbf{s}_3 = (s_{3x}, s_{3y}, s_{3z})^T$  を求める.(図 4.4 参照)

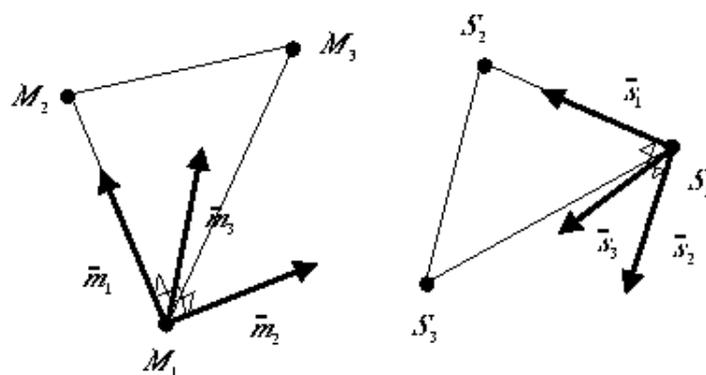


図 4.4: 座標系の変換

#### 4.4.1 回転行列の導出

次に、 $s_1, s_2, s_3$ からなる正規直交規定を  $m_1, m_2, m_3$ からなる正規直交規定に変換する変換行列  $R'$ を求めることを考える。ここで、

$$(s_1 s_2 s_3) = R' (m_1 m_2 m_3) \quad (4.7)$$

したがって、

$$R' = (s_1 s_2 s_3) (m_1 m_2 m_3)^{-1} \quad (4.8)$$

となるが、 $3 \times 3$  行列  $M = (m_1 m_2 m_3)$  は、その列成分が正規直交規定になっていることから、

$$M^{-1} = M^T \quad (4.9)$$

となるので、

$$R' = (s_1 s_2 s_3) (m_1 m_2 m_3)^T \quad (4.10)$$

さて、求める変換行列  $T$  は、並進行列  $t$  と、回転行列  $R$  に分解できる。すなわち、Scene 画像上の任意の点  $P_S$  が、 $T$  によって Model 画像上の点  $P_M$  に変換されたとすると、

$$P_M = R P_S + t \quad (4.11)$$

と表されるが、そのとき

$$R = R' \quad (4.12)$$

となることは自明である。

#### 4.4.2 並進行列の導出

次に並進行列を求める。回転行列  $R$  が求まっているので、Model 画像と Scene 画像上の対応する 1 点から並進行列  $t$  を求めることができる。すなわち、

$$P_M = RP_S + t \quad (4.13)$$

より

$$t = P_M - RP_S \quad (4.14)$$

ところが、前述のように、対応する 1 点のみの情報を用いることにより、非常に誤差の大きな並進行列  $t$  が求まる危険があるため、対応する 3 点の重心の並進を考えることにより、少しでも外乱に強くすることができる。すなわち、

$$G_M = \frac{M_1 + M_2 + M_3}{3} \quad (4.15)$$

$$G_S = \frac{S_1 + S_2 + S_3}{3} \quad (4.16)$$

により  $\triangle M_1 M_2 M_3$ 、 $\triangle S_1 S_2 S_3$  の重心を定義すると、

$$t = G_M - RG_S \quad (4.17)$$

により、よりロバストな並進行列  $t$  を求めることができる。

以上により、求めるべき回転行列  $R$  と並進行列  $t$ 、すなわち変換行列  $T$  が導出された。

以上の式変形を見れば分かるように、手動で選択する対応点 3 点は、できる限り一直線上にない方が望ましい。

## 4.5 実験結果

以下に鎌倉の大仏の距離画像を 2 枚用いて、おおまかな位置あわせを行った実験結果を示す。

図 4.5 に示すのは、大仏を真正面からスキャンした画像 (白色) と、向かって左に約 30 度の地点からスキャンした画像 (灰色) である。ここで、白色の画像を Model 画像とし、灰色の画像を Scene 画像としておおまかな位置あわせを行った。

図 4.6 に示すのは、図 4.5 に示した 2 枚の距離画像をおおまかに位置あわせした後の画像である。



図 4.5: 初期位置あわせ前の画像

一方、大仏を一周した画像系列 (12 枚) をおおまかな位置あわせした結果を図 4.7 に示す。

## 4.6 まとめ

最後に、おおまかな位置あわせについてまとめる。

おおまかな位置あわせは、次章で述べる全体位置あわせの前段階として位置づけられる。従って、できるだけ正確な位置関係を求めるのが好ましいが、実際には全く正しい位置関係は求まらない。それは以下のような理由からによる。

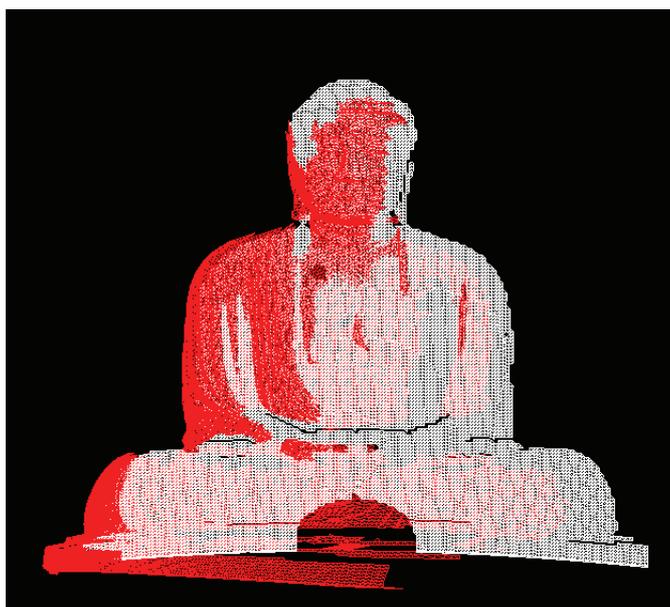


図 4.6: 初期位置あわせ後の画像

- 2枚の画像において得られる3次元データは、センサの視線方向特有の基準で量子化されたデータである。従って、全く同一の点が2枚の画像間でスキャンされていることはほとんどありえない。
- 重なりを持った距離画像において、正確な対応点を選択するインターフェースを作るのは難しい。
- 2枚の距離画像間の位置関係は比較的正確に求まったとしても、3枚目の画像の位置合わせの際に求まる変換行列は、以前に求まった変換行列に依存するため、徐々に誤差が蓄積していく。

実際、1番目の問題はセンサの性質上どうすることもできない。

2番目の問題に対しては更なるインターフェースの向上は十分に考えられるが、十分という精度は難しい。

次章で述べる全体位置あわせは、3番目の問題に対する解決策である。最初にも述べたように、全体位置あわせは一連の距離画像が、ある程度おおまかな位置あわせが施されていることを前提にしている。ある程度、という量を定量的に表現することは難しいが、全体位

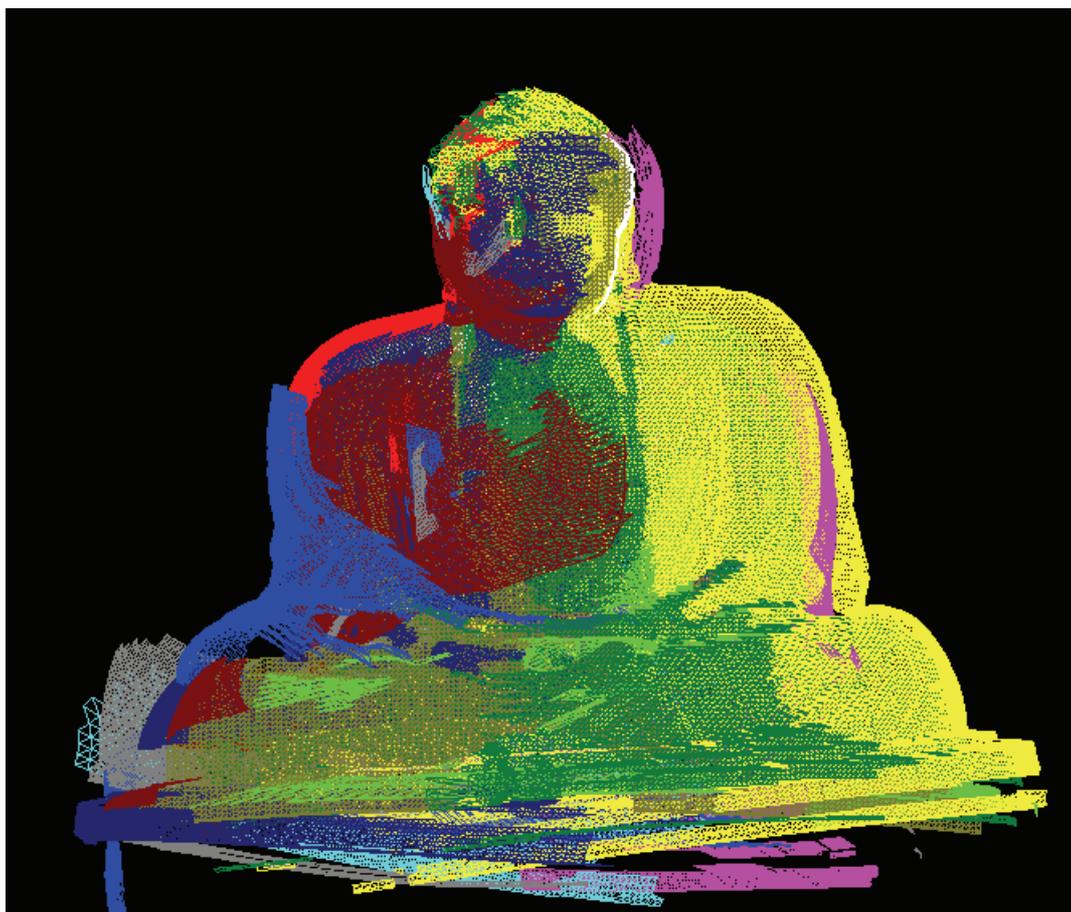


図 4.7: 12 枚の初期位置あわせ

置あわせで収束にそれほど時間を要しないほどのおおまかな位置あわせが、本章の課題であった。

## 第 5 章

### 全体位置あわせ

この章では、2枚ずつ初期位置あわせした画像系列に対して、全体で位置あわせを行う Simultaneous Registration の詳細について述べる。

## 5.1 はじめに

$n$  枚の距離画像  $S_0 \dots S_{n-1}$  の位置あわせの問題は、オーバーラップした部分における対応点間の距離を最小にする、いわゆる最小化の問題に帰着する。本章では、前章でおおよその位置あわせをしたあとの一連の距離画像に対して、更なる位置あわせを試みる。

これまでになされてきた位置あわせの多くは、3 枚以上の距離画像系列を位置あわせする場合においても、2 枚ずつのローカル位置あわせの集積として全体位置あわせを位置づけてきた。しかし前述のように、2 枚ずつのローカル位置あわせは、それ自体に少なからずの誤差を含むため、その結果の集積は非常に大きな誤差を生む。したがって、本章で行う全体位置あわせにおいては、3 枚以上の位置あわせにおいても、2 枚ずつではなく、3 枚以上同時に位置あわせを行えるアルゴリズムを用いている。

前章におけるおおまかな位置あわせにおいては、対応点は手動で選択していた。しかし手動で対応点を選択するとなると、その数には限りがある。従って、距離画像のうち限られた情報のみを用いて位置あわせを行うことになるので、非常に外乱に弱い。そこで、本章で行う全体位置あわせにおいては、距離画像の全ての情報を用いて位置あわせを行っている。すなわち 2 枚の画像間の対応点を手動ではなく、ある決まった手法に基づき、自動で選択することになっている。

第 2 節では、全体位置あわせの流れについて概説する。

第 3 節では、対応点探索の手法を具体的に説明する。この対応点探索については、

- 探索時間を優先するもの
- 探索の精度を優先するもの
- その後の全体位置あわせの収束時間を優先するもの

など非常に様々な手法が存在するので、目的に合った手法を選択することが重要である。

第 4 節では、2 枚の画像間で対応点が見つかったあと、画像を移動すべき方向を決定すべき方程式とその方程式の解法について説明する。

第 5 節では、実験結果として、鎌倉の大仏を一周した距離画像系列 (12 枚) に対して前節まで述べた全体位置あわせのアルゴリズムを適用した実験結果を述べる。またその収束

の様子をグラフに示すと同時に、得られた位置あわせ後のモデルを様々な断面で輪切りにして、その精度を検証した。

第 6 節では、本章のまとめとして、本論文で用いた全体位置あわせのアルゴリズムについて考察する。

## 5.2 全体位置あわせの流れ

以下に全体位置あわせの処理手順を簡単に示す。

全体位置あわせでは、前章のおおまかな位置あわせの結果を用いるため、おおまかな位置あわせからの流れを示す。

1. おおまかな位置あわせを行う
2. おおまかに位置あわせされた全ての画像系列に対して、2 枚ずつの画像の組をつくり、その全ての組において対応点を探索する
3. 2. の全ての結果を用いて全ての画像の変換行列を一度に求める
4. 変換された新たな画像系列に対して、収束するまで 2.3. の処理を繰り返す

以下にそれぞれの手順を具体的に示す。

2 枚の画像の位置あわせ (おおまかな位置あわせ) これは前章で述べた処理である。本論文で示す全体位置あわせの手法は、前述のように位置あわせの誤差を蓄積しないために全ての距離画像系列を同時に位置あわせする。複数の距離画像の変換行列を同時に求める方程式は一般に非線形方程式となるため、方程式を線形に近似するためには、全ての距離画像の大まかな相対的位置関係を始めに知っておかなければならない。

全ての距離画像は順々にワールド座標系に変換される。各距離画像とワールド座標系の大まかな位置関係は、新しい距離画像とそれまでにおおまかに位置あわせされた距離画像の間の少なくとも 3 つの対応点で定義される。必要な対応点はユーザが対話的に選択する。

ここで、距離画像上の任意の点 $\vec{x}$ の、ワールド座標系での点 $\vec{x}^I$ への(大まかな)変換行列を $T_i^{(\tau)}$ とする。すなわち、

$$T_i^{(\tau)} \vec{x} = \vec{x}^I \quad (5.1)$$

当然、 $T_i$ は回転行列 $R_i$ と並進ベクトル $\vec{t}_i$ に分解される。すなわち、

$$R_i^{(\tau)} \vec{x} + \vec{t}_i^{(\tau)} = \vec{x}^I \quad (5.2)$$

ここでの結果は、変換された後の画像だけでなく、その変換行列も後の処理で用いるため保持しておかねばならない。

対応点探索 本論文は、より精密なデータすなわち、より大規模なデータをより精密に位置あわせすることを目的としているため、ここでの対応点探索は処理時間を重視する。非常に単純な対応点探索は最近傍点探索であるが、この探索方法は非常に探索時間がかかる。k-d tree 法など、対応点探索を高速化するアルゴリズムを用いた手法もあるが、根本的な解決策にはならない。

本論文では、対応点を探索すべき 2 枚の距離画像が、 $z$  方向に

- 表と裏の対応 (法線ベクトルの内積が負となるような対応)
- その対応点間の距離がある閾値を超えるような対応

複数枚の画像での全体位置あわせ  $\vec{\theta}_i$ を画像 $S_i$ のワールド座標系に対する相対的な方向ベクトルとする。方向ベクトル $\vec{\theta}_i$ は 3 つの回転角と 3 つの平行移動パラメータの計 6 つのパラメータを持っている。回転行列 $R_i^{(\theta)}$ と平行移動ベクトル $\vec{t}_i^{(\theta)}$ によって方向ベクトル $\vec{\theta}_i$ が定義できる。つまり、任意の点 $\vec{x} \in S_i$ のワールド座標系への変換は次のように $T_i^{(\theta)}$ で定義できる。

位置あわせの収束 位置あわせの収束判定は非常に難しい。おおまかに位置あわせされた画像系列に対し、以上のルーチンが一度回れば、1 度全体位置あわせされた画像系列が一度に計算される。同じルーチンをもう一度回れば、2 度全体位置あわせされた画像系列が一度に計算される。したがって、収束判定は以下のようなものが考えられる。

- 各ルーチンで計算される変換行列が限りなく単位行列に近くなる
- 各ループでとってくる対応点間の距離の平均値が限りなく 0 に近くなる
- 直近のループでとってきた対応点間の距離の平均値に対する、一回前のループのそのの比が限りなく 1 に近くなる

本論文では 2 番目の収束を検証した。

### 5.3 対応点探索

本論文は、より精密なデータすなわち、より大規模なデータをより精密に位置あわせすることを目的としているため、ここでの対応点探索は処理時間を重視する。

非常に単純な対応点探索は最近傍点探索であるが、この探索方法は全点探索のため非常に探索時間がかかる。k-d tree 法など、対応点探索を高速化するアルゴリズムを用いた手法もあるが、根本的な解決策にはならない。本論文では、対応点を以下のように探索する。

まず、Model 画像を点で考え、Scene 画像をポリゴンで考える。対応点を探索すべき Model 画像上の点から、Model 画像の視線方向と平行な直線と Scene 画像の面との交点が見つかれば、その点を対応点とみなす。(図 5.1)

このように探索することによって、正面から向かって側面に位置する 2 枚の画像の対応点も正確に求めることができる。

ただし、以下のような対応点は取り除く。

- 表と裏の対応 (法線ベクトルの内積が負となるような対応)
- その対応点間の距離がある閾値を超えるような対応

### 5.4 全体位置あわせの解法

$$T_i^{(\theta)} \vec{x} = R_i^{(\theta)} \vec{x} + t_i^{(\theta)} \quad (5.3)$$

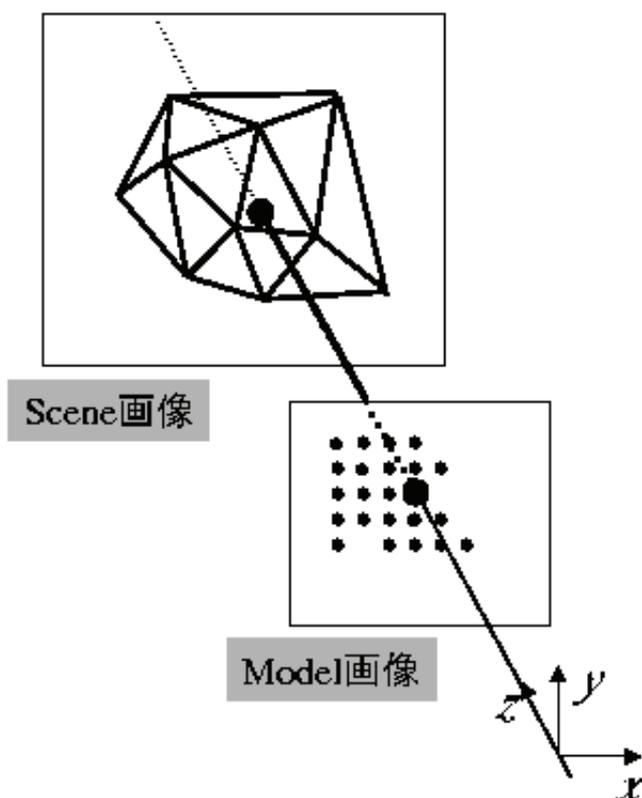


図 5.1: 対応点探索

さて、前のステップで定義された大まかな変換行列  $T_i^{(\tau)}$  と、それをより精度良く改良する変換行列  $T_i^{(\delta)}$  を用いて、

$$T_i^{(\theta)} = T_i^{(\delta)} T_i^{(\tau)} \quad (5.4)$$

と表されるとする。また、画像  $S_i$  と  $S_j$  の距離をマッチング距離  $D(S_i, S_j)$  と定義すると、問題は以下の  $\varepsilon$  の最小化問題に帰着する。

$$\begin{aligned} \varepsilon^2 &= \min_{\theta} \sum_{i \neq j} D(T_i^{(\theta)} S_i, T_j^{(\theta)} S_j) \\ &= \min_{\delta} \sum_{i \neq j} D(T_i^{(\delta)} T_i^{(\tau)} S_i, T_j^{(\delta)} T_j^{(\tau)} S_j) \\ &= \min_{\delta} \sum_{i \neq j, k} (R_i^{(\delta)} \vec{n}_{ik}^{(\tau)} ((R_j^{(\delta)} \vec{y}_{ijk}^{(\tau)} + \vec{t}_j^{(\delta)}) - (R_i^{(\delta)} \vec{x}_{ik}^{(\tau)} + \vec{t}_i^{(\delta)})))^2 \end{aligned}$$

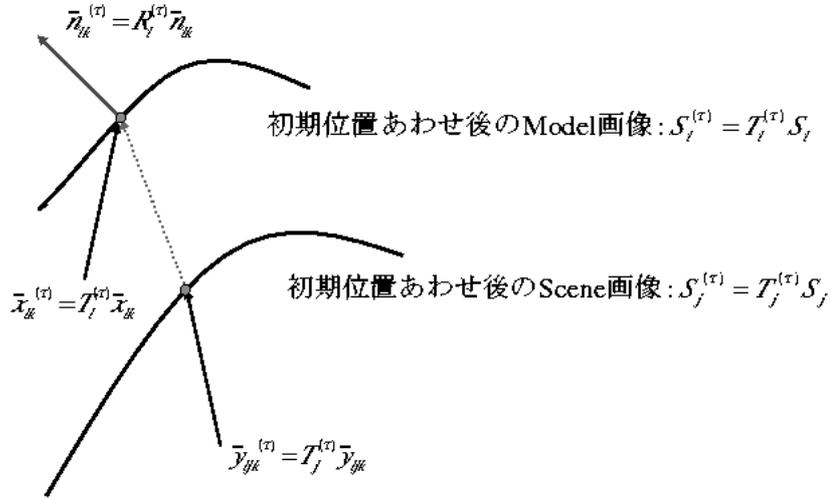


図 5.2: 対応点の処理

$$= \min_{\vec{\delta}} \sum_{i \neq j, k} \|A_{ijk} \vec{\delta} - s_{ijk}\|^2 \quad (5.5)$$

ここで、

$$s_{ijk} = \vec{n}_{ik}^{(\tau)} \cdot (\vec{x}_{ik}^{(\tau)} - \vec{y}_{jk}^{(\tau)}) \quad (5.6)$$

$$A_{ijk} = \begin{pmatrix} \underbrace{0 \dots 0}_{6i \times 1} & \underbrace{C_{ijk}}_{6 \times 1} & \underbrace{0 \dots 0}_{6(l-i-1) \times 1} \end{pmatrix} + \begin{pmatrix} \underbrace{0 \dots 0}_{6j \times 1} & \underbrace{-C_{ijk}}_{6 \times 1} & \underbrace{0 \dots 0}_{6(l-j-1) \times 1} \end{pmatrix} \quad (5.7)$$

$$C_{ijk} = \begin{pmatrix} \vec{n}_{ik}^{(\tau)} \times \vec{y}_{jk}^{(\tau)} \\ -\vec{n}_{ik}^{(\tau)} \end{pmatrix}^T \quad (5.8)$$

である。

## 5.5 非線型最小二乗解の解法

ワールド座標系は自由に定義できるので、未知のパラメータの間には線型従属性がある。この従属性は最初のレンジ画像の相対方向をワールド座標系に一致させることで避けられる。この仮定のもとでは、線形方程式 (4) は簡単に解くことができる。Newton-Taylor アルゴリズムまたは Levenberg-Marquardt 法を用いると誤差ベクトル  $\vec{\delta}$  は式 (8) のように計算で

きる.

$$\vec{\delta} = \left( \sum_{i \neq j, k} A_{ijk}^T A_{ijk} \right)^{-1} \sum_{i \neq j, k} A_{ijk}^T s_{ijk} \quad (5.9)$$

## 5.6 誤差式の線形化

誤差 $\varepsilon$ はこのままでは非線型な式なので、取り扱いが非常に難しい. そこで、一つ前のステップでほとんど正しい解が求まっていることから、回転を微小角回転とみなすことによつて回転行列を式 (9) のように表す.

$$R = \begin{pmatrix} 1 & -c_3 & c_2 \\ c_3 & 1 & -c_1 \\ -c_2 & c_1 & 1 \end{pmatrix} \quad (5.10)$$

ここで誤差の式を変形すると、

$$\begin{aligned} \varepsilon &= R_m n \cdot ((R_s y + t_s) - (R_m x + t_m)) \\ &= R_m n \cdot ((R_s y + R_m y - R_m y + t_s) - (R_m x + t_m)) \\ &= R_m n \cdot ((R_s - R_m) y + (t_s - t_m) - R_m (x - y)) \\ &= R_m n \cdot ((R_s - R_m) y + (t_s - t_m)) - R_m n \cdot (R_m n (x - y)) \\ &= R_m n \cdot ((R_s - R_m) y + (t_s - t_m)) - n \cdot (x - y) \end{aligned} \quad (5.11)$$

となり、第 2 項は論文中の  $s$  に等しい.

第 1 項を変形すると、

$$\begin{aligned} \text{第 1 項} &= R_m n \cdot ((R_s - R_m) y + (t_s - t_m)) \\ &= n \cdot ((R_s - R_m) y + (t_s - t_m)) \\ &= n \cdot \left( \begin{pmatrix} 0 & -c_{3s} & c_{2s} \\ c_{3s} & 0 & -c_{1s} \\ -c_{2s} & c_{1s} & 0 \end{pmatrix} - \begin{pmatrix} 0 & -c_{3m} & c_{2m} \\ c_{3m} & 0 & -c_{1m} \\ -c_{2m} & c_{1m} & 0 \end{pmatrix} \right) y + (t_s - t_m) \\ &= \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \cdot \begin{pmatrix} (-c_{3s} y_y + c_{2s} y_z) - (-c_{3m} y_y + c_{2m} y_z) + t_{sx} - t_{mx} \\ (-c_{1s} y_z + c_{3s} y_x) - (-c_{1m} y_z + c_{3m} y_x) + t_{sy} - t_{my} \\ (-c_{2s} y_x + c_{1s} y_y) - (-c_{2m} y_x + c_{1m} y_y) + t_{sz} - t_{mz} \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
 &= \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \cdot \begin{pmatrix} -c_{3s}y_y + c_{2s}y_z \\ -c_{1s}y_z + c_{3s}y_x \\ -c_{2s}y_x + c_{1s}y_y \end{pmatrix} - \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \cdot \begin{pmatrix} -c_{3m}y_y + c_{2m}y_z \\ -c_{1m}y_z + c_{3m}y_x \\ -c_{2m}y_x + c_{1m}y_y \end{pmatrix} \\
 &+ \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \cdot \begin{pmatrix} t_{sx} \\ t_{sy} \\ t_{sz} \end{pmatrix} + \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \cdot \begin{pmatrix} t_{mx} \\ t_{my} \\ t_{mz} \end{pmatrix} \\
 &= \begin{pmatrix} n_y y_z - n_z y_y \\ n_z y_x - n_x y_z \\ n_x y_y - n_y y_x \end{pmatrix} \cdot \begin{pmatrix} c_{1s} \\ c_{2s} \\ c_{3s} \end{pmatrix} + \begin{pmatrix} n_y y_z - n_z y_y \\ n_z y_x - n_x y_z \\ n_x y_y - n_y y_x \end{pmatrix} \cdot \begin{pmatrix} c_{1m} \\ c_{2m} \\ c_{3m} \end{pmatrix} \\
 &+ \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \cdot \begin{pmatrix} t_{sx} \\ t_{sy} \\ t_{sz} \end{pmatrix} + \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \cdot \begin{pmatrix} t_{mx} \\ t_{my} \\ t_{mz} \end{pmatrix} \\
 &= (n \times y^T \quad -n \quad -n \times y^T \quad n) \begin{pmatrix} c_{1m} \\ c_{1m} \\ c_{1m} \\ t_{mx} \\ t_{my} \\ t_{mz} \\ c_{1s} \\ c_{1s} \\ c_{1s} \\ t_{sx} \\ t_{sy} \\ t_{sz} \end{pmatrix} \tag{5.12}
 \end{aligned}$$

## 5.7 実験結果

図 5.3、5.4、5.5 に鎌倉の大仏 12 枚を全体位置あわせした結果を示す。

また、全体位置あわせの精度を検証するため、全体位置あわせされた大仏に対する切断面を図 5.6、5.7 に示す。

最後に、12 枚の位置あわせの過程で、ループ回数に対する、対応点間の誤差 (Model 画

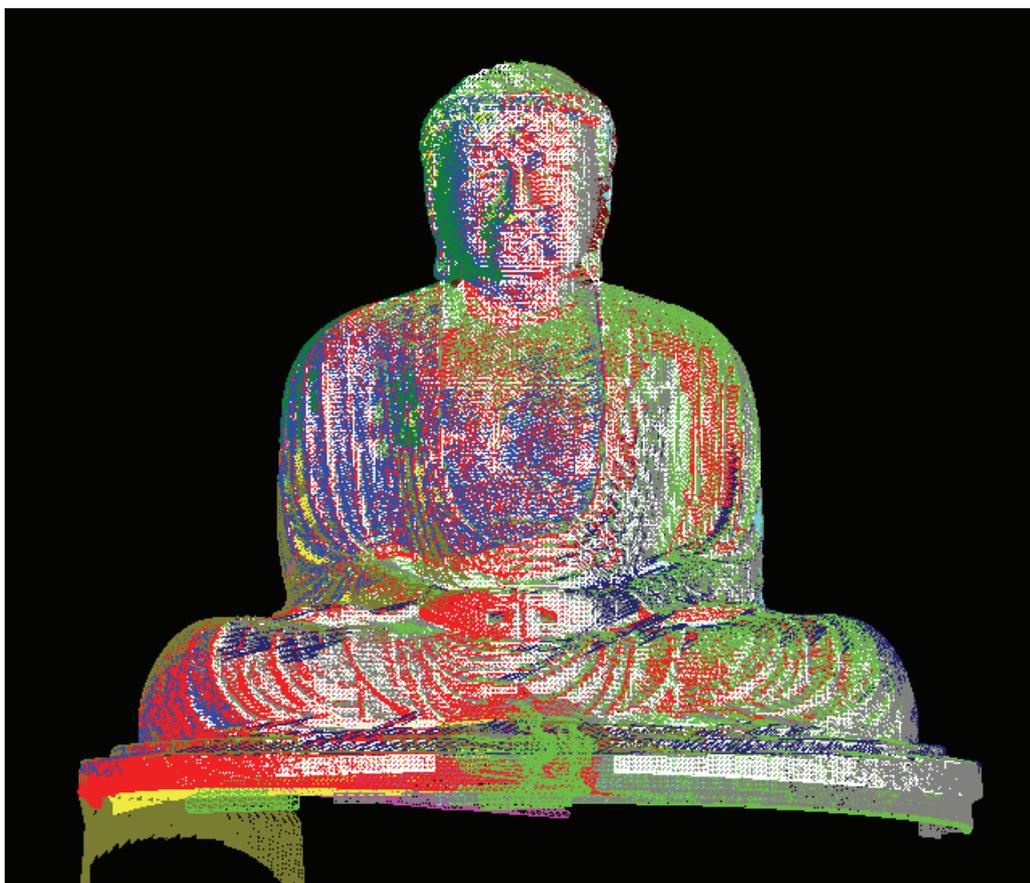


図 5.3: 全体位置あわせ後の画像 (正面)

像上の接平面と Scene 画像上の点の距離) の平均の推移をグラフ 5.8 に示す.

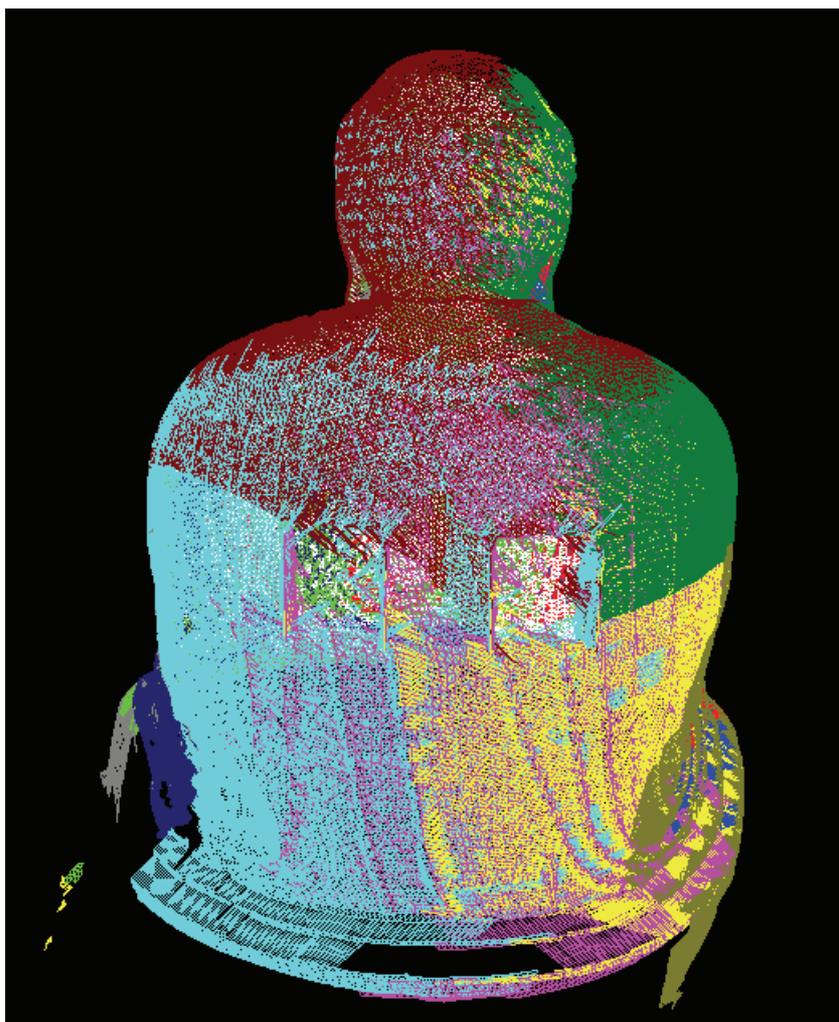


図 5.4: 全体位置あわせ後の画像 (背面)

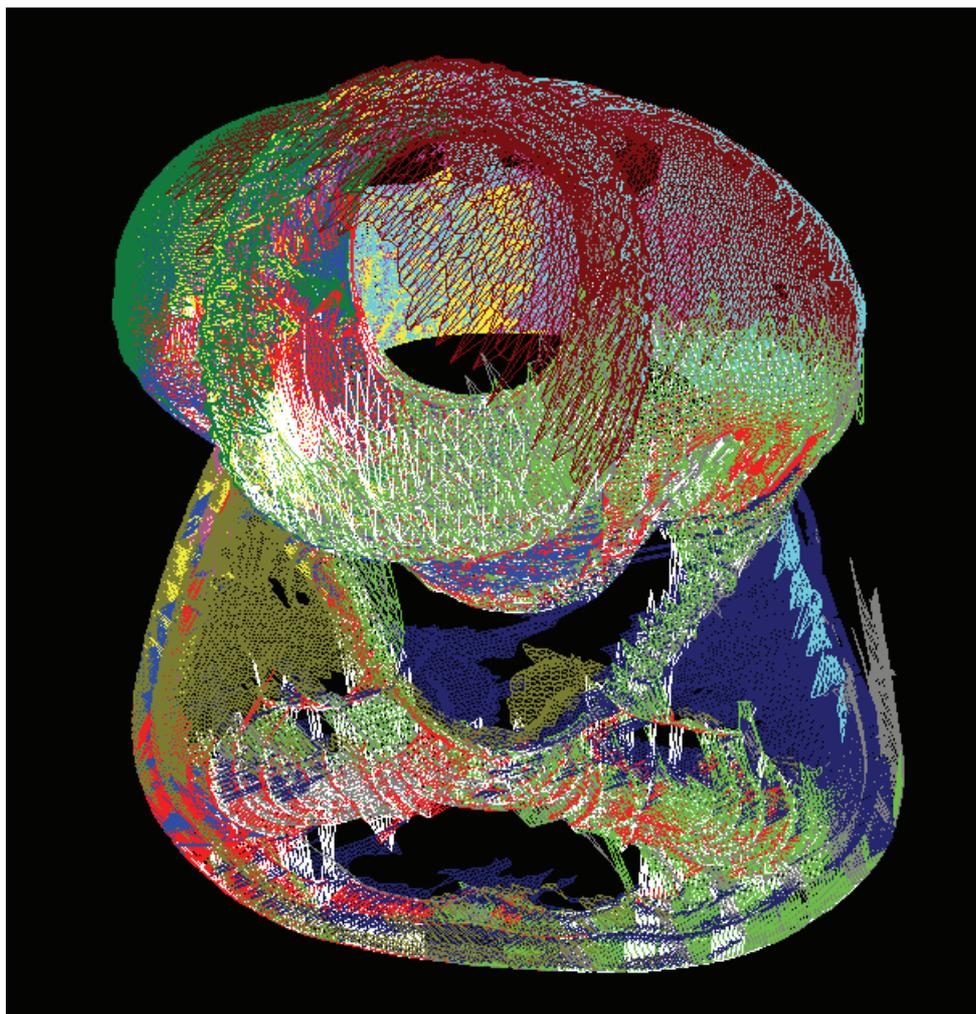
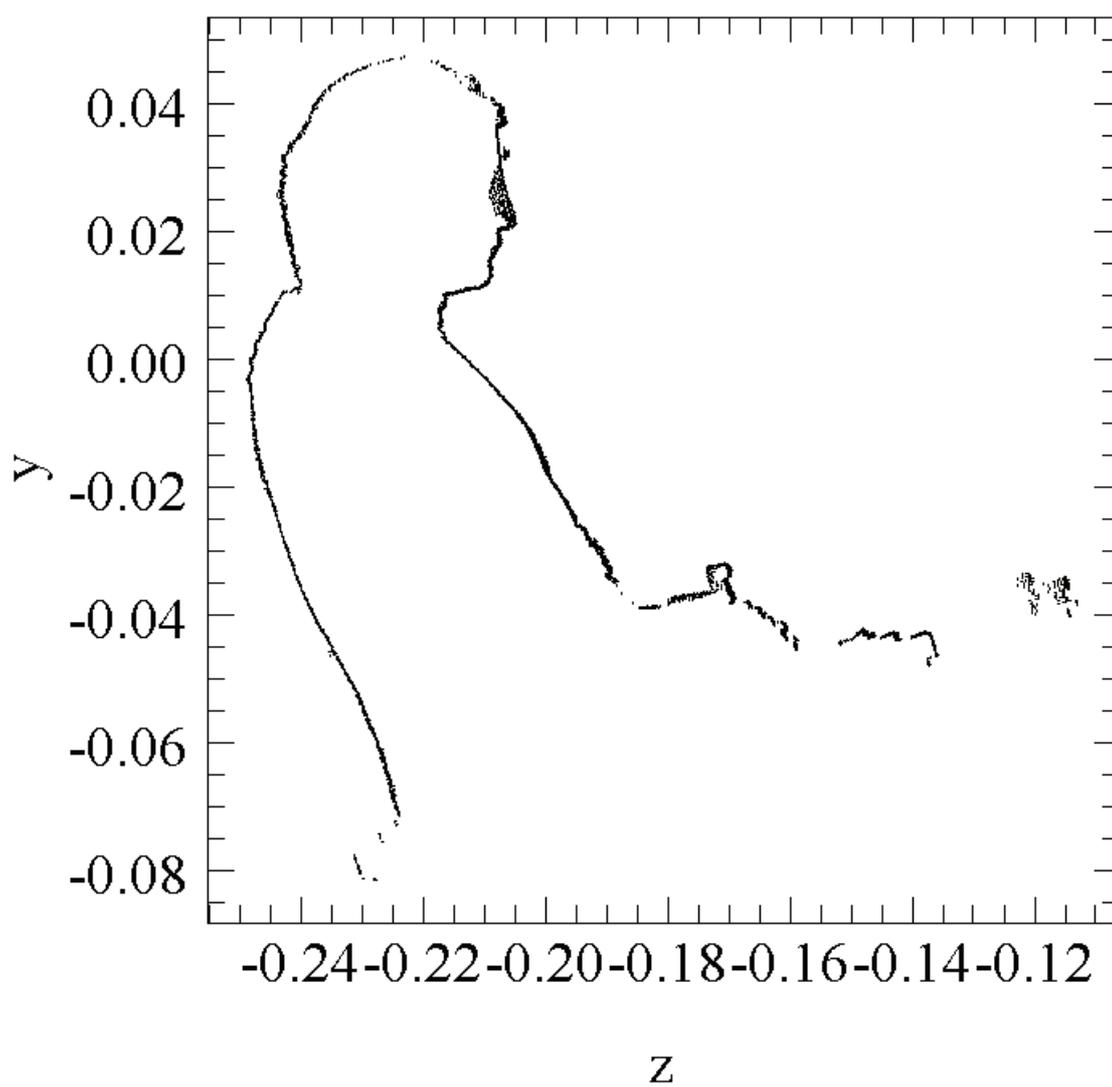
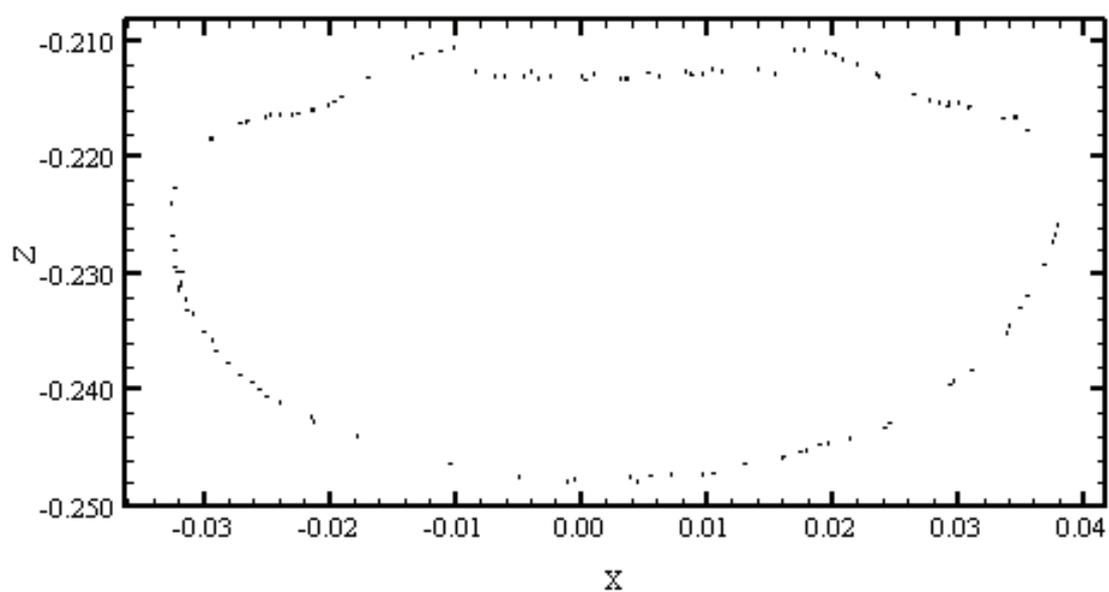


図 5.5: 全体位置あわせ後の画像 (上部)

図 5.6: 全体位置あわせ後の  $yz$  平面切断面

図 5.7: 全体位置あわせ後の  $zx$  平面切断面

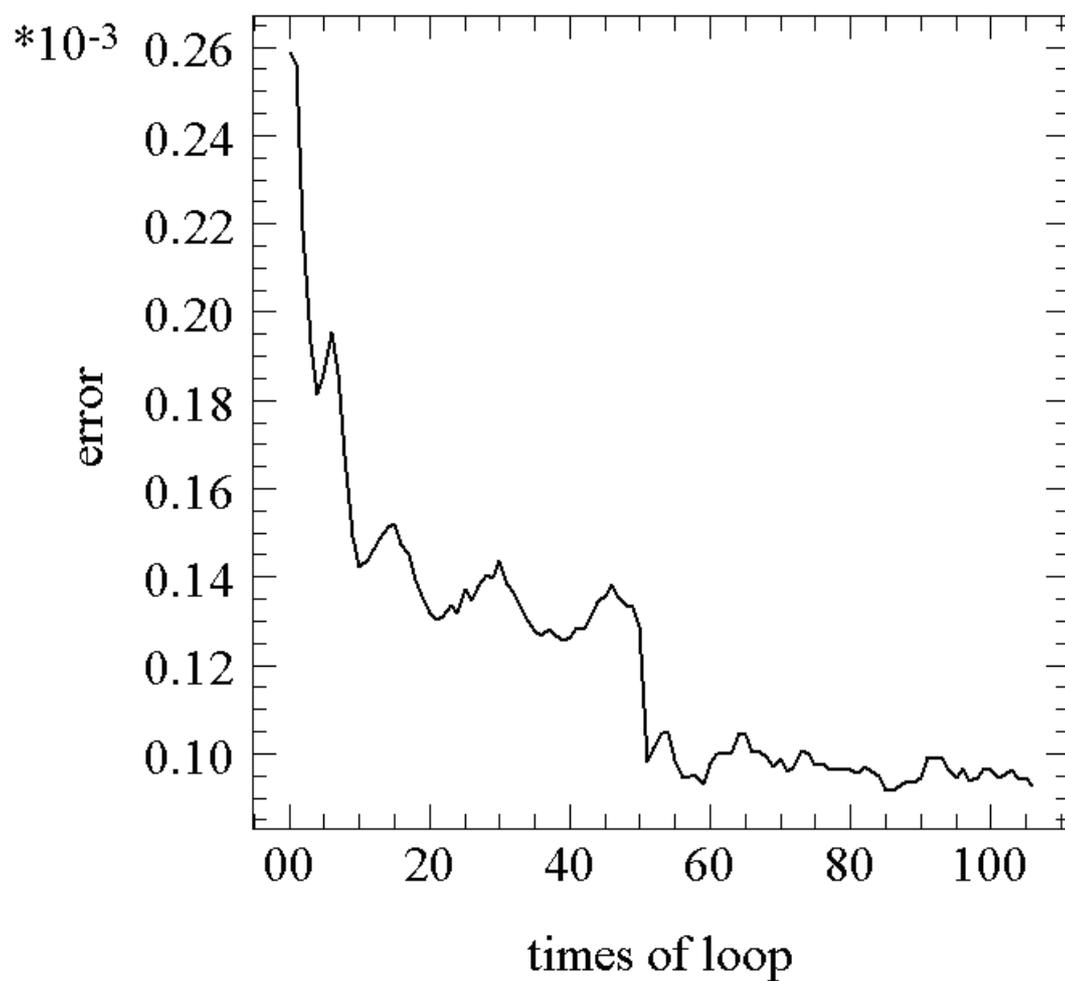


図 5.8: ループ回数と誤差の関係

## 第 6 章

### 結論

この章では、本論文を総括し結論を述べる。

## 6.1 まとめ

おおまかな位置あわせは、全体位置あわせの前段階として位置づけられる。従って、できるだけ正確な位置関係を求めるのが好ましいが、実際には全く正しい位置関係は求まらない。それは以下のような理由からによる。

- 2枚の画像において得られる3次元データは、センサの視線方向特有の基準で量子化されたデータである。従って、全く同一の点が2枚の画像間でスキャンされていることはほとんどありえない。
- 2枚の距離画像間の位置関係は比較的正確に求まったとしても、3枚目の画像の位置合わせの際に求まる変換行列は、以前に求まった変換行列に依存するため、徐々に誤差が蓄積していく。

実際、1番目の問題はセンサの精度や測定誤差につながる部分である。本論文では、対応点探索の際に、点と面を対応させることによって、すなわち、点を補間することによってこの問題を回避している。

全体位置あわせは、2番目の問題に対する解決策である。最初にも述べたように、全体位置あわせは一連の距離画像が、ある程度おおまかな位置あわせが施されていることを前提にしている。ある程度、という量を定量的に表現することは難しいが、全体位置あわせで収束にそれほど時間を要しないほどのおおまかな位置あわせが、“おおまかな位置あわせ”に要求される。

一般的に初期位置あわせはマウスを使って行う場合が多いが、微妙な位置あわせは非常に熟達したテクニックを必要とする。したがって、位置あわせに非常に時間を要することが多い。しかし本論文の手法では、非常に短い時間で位置あわせを行うことができる。

本論文での全体位置あわせは、以前の2枚の逐次位置あわせの集積としての全体位置あわせを改良したものである。これにより、誤差が集積することなく全体に分散し、ロバストな位置あわせが行えた。

位置あわせの結果がどの程度厳密かを判断するのは非常に難しい。1つの方法として、対応点間の点と面の距離がどの程度小さくなったかを図5.8に示した。この結果により、1cmのオーダーまでの位置あわせができていることが読み取れる。また、図5.6、図5.7に位置

あわせされた画像系列を断面した結果を示している。これらの図により、位置あわせが非常に精巧に行われたことを示している。

本論文での全体位置あわせは、最小二乗法として、Newton-Taylor 法を採用している。Newton-Taylor 法の欠点は局所解にはまり易いこと、そして最終解が初期位置に非常に左右されやすいことである。局所解の問題は、対応点探索の場合の閾値をうまく設定することによって回避している。

## 6.2 今後の課題

上に示した全体位置あわせにおける Newton-Taylor 法の 2 つの欠点を根本的に回避するためにはマーカート法を実装しなければならない。

また、レンジセンサから得られる輝度情報を用いてよりロバストな位置あわせを行いたい。

最後に、位置あわせされたデータを用いてマージングを行い、一元化されたデータを作成したい。

## 参考文献

- [1] 河井 良浩, 植芝 俊夫, 吉見 隆, 大島 正隆. 多視点レンジデータからの3次元形状復元. 電子情報通信学会誌, Vol.J75-D, No.4, pp. 737-748, 1992.
- [2] H.Shum, K.Ikeuchi and R.Reddy. Principal Component Analysis with missing Data and Its Application to Polyhedral Object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 9, pp. 854 - 867, 1995.
- [3] 増田 健, P.Boulangier. 多視点距離画像の空間統合による全周計測 (ミニ特集 リアルタイムに近づいた3D形状計測). 計測と制御, Vol.34,No.6,pp. 449 - 452, 1995.
- [4] P.J.Besl, N.D.McKay. A Method of Registration of 3-D Shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence* Vol. 14,No.2 pp. 239 - 256, 1992.
- [5] Z.Zhang. Iterative Point Matching for Registration of Free-Form Curves and Surfaces. *International Journal of Computer Vision*, Vol. 13, No. 2, pp. 119-152, 1994.
- [6] Yang Chen, Gerard Medioni. Object modelling by registration of Multiple range Images. *Image and Vision Computing* Vol. 10, No.3 pp. 145 - 155, 1992.
- [7] R.Bergevin, M.Soucy, H.Gagnon and D.Laurendeau. Toward a General Multi-View Registration Technique. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.18, No. 5, pp. 540-547, 1996.
- [8] G.Turk, M.Levoy. Zippered Polygon Meshes from Range Images. In *ACM SIGGRAPH Computer Graphics*, pp. 311 - 318, 1994.

- [9] T.Masuda and N.Yokoya. A Robust Method for registration and Segmentation of Multiple Range Images. *Computer Vision and Image Understanding*, Vol. 61, No. 3, pp. 295 - 397 , 1995.
- [10] 増田 健, 横矢 直和 . モデル生成のための複数の距離画像の位置あわせと統合 . 画像の認識・理解シンポジウム 講演論文集, pp. 247-252, 1996.
- [11] C.Dorai, G.Wang, A.K.Jain and C.Mercer. Registration and Integration of Multiple Object Views for 3D Model Construction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, pp. 83 - 89, 1998.
- [12] J.Feldmar, N.Ayache and F.Berring. 3D-2D Projective Registration of Free-Form Curves and Surfaces. *Computer Vision and Image Understanding*, Vol. 65, No. 3, pp 403 - 424, 1997.
- [13] M.Rutishauser, M.Stricker and M.Trobina. Merging Range Images of Arbitrary. In *Proceedings of IEEE Computer Vision and Pattern recognition*, pp. 573 - 580, 1994.
- [14] A.Hilton, A.J.Stoddart, J.Illingworth and T.Windeatt. Reliable Surface Reconstruction from Multiple Range Images. In *Proceedings of IEEE Computer Vision and Pattern recognition*, pp. 117 - 126, 1996.
- [15] Y.Chen and G.Medioni. Description of Complex Objects from Multiple Range Images Using an inflating Ballon Model. *Computer Vision and Image Understanding*, Vol. 61, No. 3, pp. 325 - 334 , 1995.
- [16] H.Hoppe, T.DeRose and T.Duchamp. Surface Reconstruction from Unorganized points. In *ACM SIGGRAPH Computer Graphics*, Vol. 26-2, pp. 71 - 78, 1992.
- [17] P.Hebert, D.Laurendeau and D.Poussart. Surface Profile Reconstruction: Reliable Geometric Primitive Extraction. In *Proceedings of IAPR International Conference on Pattern recognition* , pp. A-258 - 263, 1994.

- 
- [18] Peter J. Neugebauer. Reconstruction of Real-World Objects Via Simultaneous Registration and Robust Combination of Multiple Range Images. *International Journal of Shape Modeling*, Vol. 3 No.1 & 2, pp. 71 - 90, June 1997.
- [19] Mark D. Wheeler, Yoichi Sato, Katsushi Ikeuchi. Consensus Surfaces for Modeling 3D Objects from Multiple Range Images. *Proceedings of DARPA Image Understanding Workshop '97*.
- [20] Berthold K.P.Horn. Closed-form Solution of Absolute Orientation Using Unit Quaternions. *Optical Society of America*, Vol. 4 No.4/April 1987.
- [21] O.D.Faugeras, M.Hebert. The Representation, Recognition, and Locating of 3-D Objects. *The International Journal of Robotics Research*, pp. 27 - 53, May 1986
- [22] R.Bergevin, D.Laurendeau and D.Poussart. Registrating Range Views of Multipart Objects. *Computer Vision and Image Understanding* 61 pp. 1 - 16, 1995.
- [23] K.Higuchi, M.Hebert and K.Ikeuchi Building 3-d models from Unregistered Range Images. *Graphical Models and Image Processing* 57 pp.315 - 333, 1995.
- [24] B.Curless and M.Levoy. A Volumetric Method for Building Complex Models from Range Images. *Proc. of SIGGRAPH '96 New Orleans, Louisiana* pp. 303 - 312, 1996.
- [25] G.Borgefors. Chamfering: A Fast Method for Obtaining Approximations of the Euclidean Distance in N Dimensions. *3rd Scand. Conf. on Image Analysis*, Copenhagen, Denmark 1983.
- [26] W.E.Lorensen and H.E. Cline. Marching Cubes: A High Resolution 3d Surface Construction Algorithm. *Proc. of SIGGRAPH '87 Anaheim, California* pp. 163 - 169, 1987.

## 発表文献

1. 西川 拓, 池内克史. 高精度レンジセンサを用いた3次元モデルの構築 (Constructing a 3D Modeling Using a High Resolution Range Sensor) 日本バーチャルリアリティ学会 第4回大会 pp. 199 - 202, 1999.
2. 西川 拓, 西野 恒, 佐藤洋一, 池内克史. 高精度レンジセンサによる3次元モデルの構築. 情報処理学会第59回全国大会 pp. 4-165 - 4-166, 1999

## 謝辞

この論文を書くにあたって、多くの方々にお世話になりました。

池内克史教授には、論文のテーマ決めから始まり、毎週のミーティングでのこと細かなご指導や、で大変お世話になりました。また佐藤洋一講師には研究の方向性について、色々ご指導いただきました。一方、影澤政隆助手、長谷川仁則技官、秘書の元木恵子さん、松浦芳子さんには日常の細かなお手伝いをして頂き本当にありがとうございました。研究室の先輩、同輩、後輩には、研究に関する議論を交わし合い、貴重な意見を沢山いただきました。

最後に、鎌倉の大仏のデータを取得するにあたり、後徳院の方々を始め、CADセンターの方々にも大変お世話になりました。ここに感謝の意を述べさせていただきます。