修士論文

異種のレーザレンジセンサを用いた大規模距離画像の並列統合手法

指導教官　　池内　克史　教授

東京大学大学院　情報理工学系研究科

電子情報学専攻

086425　浜崎佑樹

提出日 2 月 9 日

内容概要

近年，実物体の 3 次元モデル化の技術の発展に伴い，文化遺産のモデル化に注目が集まっている．文化遺産をモデル化することで，保存や修復，解析，展示などに利用することができると考えられている．しかし，現在までに提案されている手法を文化遺産に適用するには，様々な問題がある．

文化遺産はその複雑性から，用途に応じて様々な種類のレーザレンジセンサを併用して計測を行う必要がある．一方で，従来の手法は，同種のレーザレンジセンサによって得られた距離画像を対象としていた．そのため，文化遺産に既存の手法を適用すると，得られた距離画像同士の整合性が取れず，不均整な 3 次元モデルが生成され，膨大な計算時間がかかってしまう．

そこで本研究では，従来の手法を拡張し，異種のレーザレンジセンサによって得られたデータに適用できるような手法を提案する．

まず，均整の取れた 3 次元モデルを生成するため，信頼度付き合致表面法を提案する．従来の合致表面法は，精度の高いデータと精度の低いデータを同等に扱っているため，精度の低いデータの影響により，不均整な 3 次元モデルが生成されていた．そこで，レーザレンジセンサの信頼度を導入し，精度に応じて統合における優先度を設定することで，様々な精度のレーザレンジセンサによって得られたデータから，均整の取れた 3 次元モデルを生成することが可能になったことを示す．

次に，計算時間を短縮するため，Resolution Grid Kd-tree と適応的空間分割法を提案する．現在，大規模な距離画像を統合するために，PC クラスタを用いて並列計算を行う手法として，Grid Kd-tree や八分木を用いた並列計算手法が提案されている．しかし，データを均等に分割して PC クラスタに割り当てているため，異なる解像度が混在するデータに適用すると，分割されたデータの大きさに偏りが生じ，効率的な並列計算が行われない．そこで，含まれるデータの大きさを考慮した適応的な分割を行うことで，様々な解像度のデータが混在していても，効率的な並列計算を行うことが可能となったことを示す．

目次

図目次

表目次

1 序論

1.1 はじめに

近年，歴史的建造物などの文化遺産を 3 次元モデル化するデジタルアーカイブの研究が注目を集めている[1] [2] [3][4]．カラーカメラやレーザレンジセンサなどのセンシング技術の発達に伴い，現実世界の物体を 3 次元モデルとしてデジタル化する研究が，盛んに行われるようになってきている．この実物体の 3 次元モデル化技術は，学術，芸術，エンターテイメントなどの様々な分野において活用することができるため，その発展が期待されている．そうした中で，デジタルアーカイブは最も注目される分野の一つとして，近年多くの関心が寄せられている．

文化遺産を 3 次元モデル化する技術は，保存修復や解析に非常に有用であるため，その必要性が高まっている．建造物などの屋外に存在する文化遺産は，外気にさらされているため劣化の問題や，事故や意図的な破壊によって，崩壊の危機に瀕しているものも少なくない．そこで，これらの文化遺産の 3 次元形状をデジタルデータとして保存しておくことで，後の修復の手助けとすることができるようになる．また，文化遺産はその重要性から，慎重な扱いが必要であり，解析手段が制限されている．しかし，デジタル化されたデータを用いることによって，様々な解析をソフトウェア的に行うことができる．一方で，得られた文化遺産の 3 次元モデルは，保存修復や解析以外にも，教育や観光にも活用することができる．例えば，3 次元モデル化された文化遺産を複合現実感と呼ばれる方法によって公開することによって，映像などで見るよりもリアルに体験することができるようになる[5][6][7].

大規模な文化遺産の 3 次元モデルを作成する研究は，これまでにも広く行われてきた．大規模な建造物を対象とする場合，その 3 次元モデルを作成するためには多量のデータを必要とする. この大規模なデータを処理するために, PC クラスタを用いて並列計算を行い，文化遺産を 3 次元モデル化する研究なども行われている[36][38]．しかし，これらの研究では，取得されたデータの精度や解像度が均一であると仮定して処理を行っていたため，生成される 3 次元モデルの精度に問題があり，また処理に膨大な計算時間がかかってしまうという問題があった．

我々が対象とするような大規模で複雑な建造物を 3 次元モデル化する場合，様々な種類のレーザレンジセンサを用いて計測を行う必要がある．本研究で対象とするバイヨン寺院は，カンボジアアンコール遺跡の 1 つであり，150m×150m 程の敷地に建てられた巨大な建造物である．中央にある塔は高い部分で役 40m の高さがあり，51 本の塔には尊顔と呼ばれ

る彫刻がなされている．寺院全体は二重の回廊で囲まれており，その回廊には当時の人々の様子が描かれた精細な浮き彫りがされている．このように，バイヨン寺院は屋外に存在する大規模な建造物でありながら，そのモデリングには高精細な表面形状の情報が必要とされる．そのため，用途に応じて様々な種類のレーザレンジセンサが用いられ，得られたデータは異なる精度や解像度のデータが混在している．

そこで本研究では，様々な種類のレーザレンジセンサによって得られた大規模データを効率的に処理し，高精度な 3 次元モデルを作成する手法を開発する．前述のように，これまでの手法では同種のレーザレンジセンサから得られたデータ群を対象としていたため，異種のレーザレンジセンサから得られたデータ群に適用すると，以下のような 2 つの問題が生じていた．まず 1 つ目は精度の問題である．精度の異なるレーザレンジセンサによって得られたデータを対等に扱っていたため，これらが混在する部分では，精度が低いデータに影響されてしまうという問題があった．2 つ目は計算時間の問題である．大規模データを処理するために並列化手法が提案されていたが，この手法では，解像度の異なるデータに対して均一に分割を行い, PC クラスタに割り当てていたため, 計算時間が増大していた．そこで本研究では，異種のレーザレンジセンサによって得られたデータ群において，高精度な 3 次元モデルを生成し，計算時間を短縮するため，レーザレンジセンサの特性を考慮した効率的な処理を行う手法の開発を目的とする．

図 1.1 バイヨン寺院

1.2 実物体の 3 次元モデル化

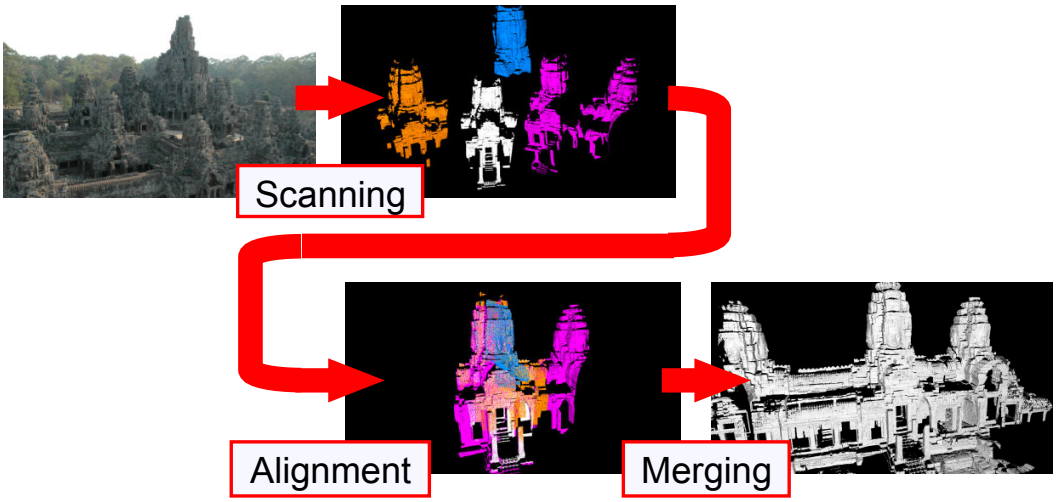

図 1.2 実物体の 3 次元モデル化

実物体から 3 次元モデルを生成するには，以下の 3 ステップを行う．

1) 計測

2) 位置合わせ

3) 統合

1)ではレーザレンジセンサを用いて実物体の計測を行い，距離画像を取得する．レーザレンジセンサは，可視領域しか一度に計測できないため，様々な位置や方向から複数回に分けて計測を行う必要がある．また，文化遺産のように複雑な対象物を計測する場合，様々な種類のレーザレンジセンサを用途に応じて使い分けるため，取得される距離画像はそれぞれ異なった精度や解像度となる．

2)では取得された距離画像を同一の座標空間に統一し，位置を合わせる．取得された距離画像はそれぞれ異なる座標系に保持されているため，その位置関係が明確ではない．そのため，距離画像の位置を合わせる必要がある．

3)では位置あわせされた距離画像群をまとめて，1 つの 3 次元モデルとして再構築を行う．2)で位置あわせされた距離画像は，あくまで複数の距離画像の集合体に過ぎないため，距離画像同士の重なった部分の冗長性を取り除き，1 つの 3 次元モデルに統合する必要がある．

以下では，それぞれのステップについて，関連手法を取り上げながら，説明する．

1.2.1 計測

まず始めに，実物体から距離画像と呼ばれる，物体表面の部分形状を取得する．本研究では，レーザレンジセンサを用いて計測された部分形状を用いる．レーザレンジセンサとは，レーザを用いてセンサ本体と物体の表面との距離を計測するもので，レーザが反射して返ってくる時間によって計算する Time-of-flight 法や，照射地点と対象物とカメラの 3 点で三角測量によって計算する光切断法などが用いられている．レーザレンジセンサは，非常に高解像度かつ高精度での計測が可能である．本研究ではバイヨン寺院を初めとする文化遺産を対象としているため，高解像度かつ高精度での 3 次元モデル化が要求される．一方で，レーザが届く範囲までしか計測が行えないという特性上，一度に取得できる範囲が限られているため，位置や方向を変えながら複数回に分けて計測を行う必要がある．その主な目的は文化遺産の保存や修復であるため，計測に時間がかかることも大きな問題とはならない．また可能な限り，高精度かつ高解像度な計測を行うことが望まれる．これらの理由から，レーザレンジセンサを用いた計測は目的に合致していると言える．

本研究では以下のレーザレンジセンサによって取得された距離画像を対象としている．

- Leica Geosystems, Cyrax 2500
- Z+F, IMAGER5003
- KONICA MINOLTA, VIVID910
- 気球センサ (Floating Laser Range Sensor)
- 木登りセンサ

Cyrax2500 は Leica Geosystems の製品であり，高解像度で計測を行うことができる．計測原理としては，レーザが対象物に反射して返ってくるまでの時間によって距離を計算する Time-of-flight 法を用いている．測定距離は 1.5 – 50m，分解能は 50m 地点で 0.25mm，精度は ±4mm である．

IMAGER5003 は Zoller+Fröhlich の製品であり，水平方向は本体を回転させ，垂直方向は内部に存在する鏡を回転させることで，球状に幅広い範囲を計測することができる．最大で，水平方向は 360°，垂直方向は 310°計測することができる．計測時間は中程度の解像度において 100 秒程度である．計測原理としては，レーザを照射した際の対象物からの反射波の位相のずれから距離を計算する位相差方式を用いている．精度は±5mm である．

VIVID910 は KONICA MINOLTA の製品であり，測定距離が限られるものの，非常に高精度かつ高解像度で計測を行うことができる．そのため，彫刻などの精細な計測が必要な箇所に用いる．測定距離に応じて，レンズを付け替えることができるようになっている．計測時間は 2.5 秒程度である．スリット状のレーザでスキャンを行う光切断方式を用いており，

三角測量の原理で距離計算を行う．測定距離は 0.6 – 1.2m，精度は±0.008mm である．

気球センサは阪野らによって提案されたセンサであり[8]，気球でスキャナ部分を吊ることによって上空から計測を行うことができる．精度は低いものの，計測を行うことが困難な建造物の上部などを計測することができる．気球が運動することによって生じる歪みに対して，気球に取り付けたビデオカメラから得られる画像列を用いて運動を推定し，データの補正を行っている．レーザ光源には Z+F LARA25200 を用いている．計測原理は Time-of-flight 法である．精度はおよそ±20mm 程度であることが知られている．

木登りセンサとは松井らによって提案されたセンサであり[9], 2 台のラインスキャナが装着された基盤を梯子型のリフトに取り付け，地面に対して垂直方向にセンサを移動させながら計測を行うことができる．センサの寸法や視野角の問題によって計測が困難な，狭い部分の計測に用いる．センサユニットには SICK LMS200 を用いている．計測原理は Time-of-flight 法で，分解能は 10mm，精度は±15mm である．

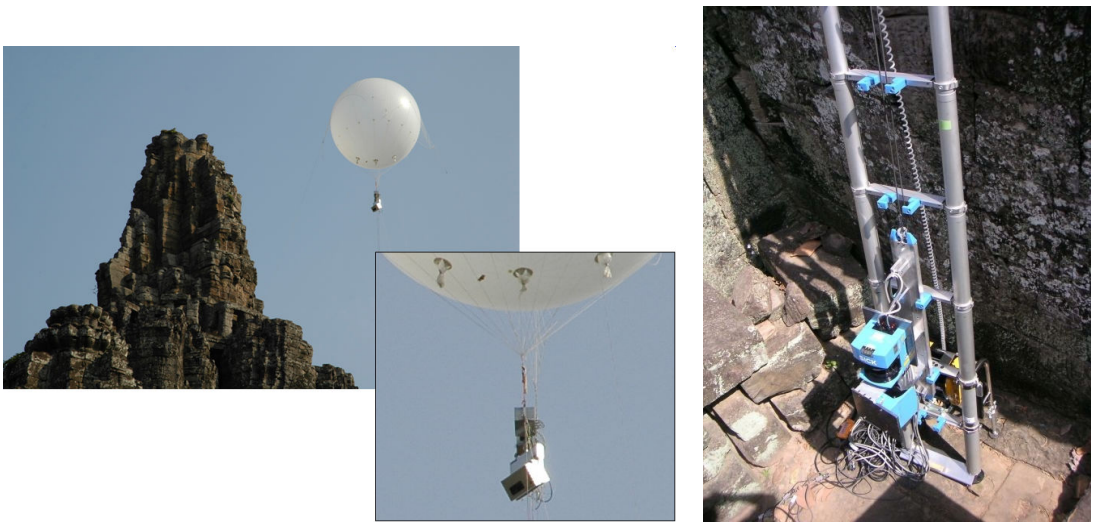

図 1.3 気球センサ(左)と木登りセンサ(右)

以上の 5 種のレーザレンジセンサによって取得された距離画像を用いる．このように，本研究で計測対象とするバイヨン寺院のように複雑で大規模な場合，様々なレーザレンジセンサを用途に合わせて用いるため，様々な精度のデータが混在することになる．本研究ではこれらのレーザレンジセンサによって得られたデータを効果的に処理し，3 次元モデルを生成する．

1.2.2 位置合わせ

1.2.1 において述べたとおり，レーザレンジセンサで実物体の表面形状のデータを取得するためには，様々な位置や方向において，多数の計測を行う必要がある．こうして計測された距離画像は，それぞれの計測位置における座標系で取得される．すなわち，どの距離画像がどの距離画像と隣り合っているか等といった情報が含まれていない．言い換えれば，距離画像間において，どの点とどの点が対応するかといった明確な対応関係は既知ではない．そのため，それぞれの相対的な位置関係を求め，座標系を統一する位置合わせの処理が必要となる．

位置合わせの手法としてよく知られているものに ICP(Iterative Closest Point)法[10]がある．2 枚の距離画像において，それらの間で最も近い点同士の対応付けと，その対応点間距離を最小にするような変換行列の推定を，交互に繰り返し計算を行うことによって，対応付けと位置合わせを同時に行う手法である．これに対して Chen らが提案した手法[11]では，ある距離画像における点を制御点にし，その法線と他の距離画像との交点を対応付けることで，位置あわせを行った．面上に対応点を求めた方が，解の収束性が良いことが知られている[12]．一般的に距離画像は一部しか重ならないように取得されているため，ICP 法の極小解が正しい位置となる保障は無い．すなわち，重なっていない部分のデータに不要な対応関係を作らないように処理を行う必要がある．そのための閾値や拘束条件を導入した様々な手法が提案されている[13][14][15][16].

以上に述べたこれらの手法は，2 枚の距離画像の相対関係を求めるもので，距離画像が多い場合には，前述の処理を順次繰り返し行う必要があり，誤差の蓄積が問題となる．そこで必要となるのが同時位置合わせの手法であり，その一つとして Neugebauer らの手法[17]がある．これは，視線方向に対応点を探索し，距離画像間における点と面の距離の二乗和を誤差として定義し，最小二乗問題を線形化して取り扱って解くことで，同時位置合わせを行う手法である．

いずれの手法においても，対応点探索における計算量が問題となってくる．ICP 法においては，距離画像におけるすべての頂点に対して最近傍点探索を行うため，計算量が膨大となってしまう．最近傍点探索を高速化するためのデータ構造として Kd-tree を用いる手法[18]が提案されている．しかしながら，大規模な物体を対象とした場合のように，多数の距離画像を位置合わせするには，計算量やメモリ使用量の点において困難となる．そこで，本研究では大石らが提案した並列同時位置合わせ手法[19]を用いる．この手法は，距離画像に対して前処理を行うことで，PC クラスタへ効率的に距離画像を割り当てることでメモリの無駄をなくし，大規模な距離画像を高速に位置合わせすることが可能である．

1.2.3 統合

位置合わせされた距離画像は，実物体と同じような形状になるように，相対的な位置関係が求められたに過ぎず，この時点では複数の距離画像の集合でしかない．すなわち，距離画像同士が重なり合う部分が存在し，頂点密度が場所によって異なる状態となっている．この距離画像群をそのまま 3 次元モデルとして取り扱うには向いていない．したがって，この重なり合った部分の冗長さを取り除き，一つの 3 次元モデルとしてメッシュを再構築する必要がある．これが統合処理である．同時にこのステップで，重なり合う部分を比較することによって，誤差の修正を行う．これは，レーザレンジセンサの計測誤差や，位置あわせの際に生じる誤差によって，実物体と生成された 3 次元モデルとの間に乖離を引き起こす可能性があるためである．

統合の手法は，以下の 2 つに大別することができる．

- メッシュベースの統合手法
- ボリュームベースの統合手法

以下では，それぞれの手法について，関連手法を取り上げながら説明する．

1.2.3.1 メッシュベースの統合手法

メッシュベースの統合手法では距離画像のメッシュ同士を直接繋ぎ合わせるといった処理を行うことで，統合を行う．

Turk らの手法

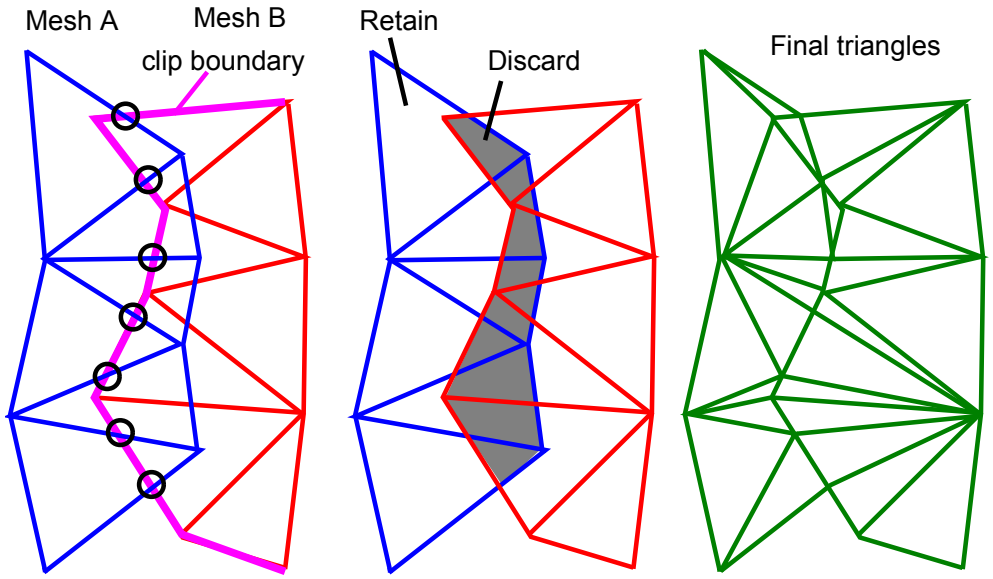

図 1.4 メッシュ縫合手法

メッシュベースの統合手法としてよく知られているものに，Turk らによるメッシュ縫合(Mesh Zippering)手法[20]がある．この手法では 3 次元メッシュをジッパーのように縫い合わせていくことによって，統合を行っている．具体的には，以下の 3 ステップで構成される．

1) 重なったメッシュから冗長なパッチを取り除く．
2) 2 つのメッシュを繋ぎ合わせる．
3) 各距離画像との一貫性のあるメッシュにする．

まず，オーバーラップ部分を除去する．メッシュ A とメッシュ B があるとしたとき，メッシュ A のあるパッチ T を考える．T の頂点 V_1，V_2，V_3 のそれぞれから一番近いメッシュ B の頂点を探す．それぞれの距離 d が閾値内でかつ，メッシュ B の境界上にあるパッチではないとき，T を除去する．

次に，二つのメッシュを繋ぎ合わせる．メッシュ A の縁がメッシュ B の境界に交わるとき，新しい頂点 Q を生成する．頂点 Q とメッシュ B の境界上の頂点を用いて両メッシュ共通のパッチを形成する．これらの Q をまとめるために，メッシュ B を新たな頂点 1 つにつき一度分割する．そしてメッシュ B を 2 つのパートに分割し，メッシュ B の内部にあるものは除去する(図 1.4).

最後に，各距離画像との一貫性のあるメッシュにする．これは，結合後のメッシュと元の距離画像を比較することで行われる．すなわち，結合後のメッシュにおけるある頂点の法線と，その頂点の周囲にある元の距離画像の頂点の法線を比較する．そして，その頂点の法線を，周囲の法線の重み付き平均とする．さらに，頂点の位置を，その法線近くの点の平均とする．

Soucy らの手法

Soucy らも同様にメッシュベースの統合手法を提案している[21]．まず，冗長に取得された部分を検出し，重複の状態によってデータを分割する．そして，それぞれの部分に対して以下の処理を行う．

1) 各部分を平面近似して新たな直交座標系を導入し，格子状のメッシュを定義する．
2) それぞれの距離画像をメッシュに投影し，メッシュの頂点での各距離画像の高さを求める．
3) 高さの重み付き平均を取り，メッシュの各頂点の位置を決定する．

このようにして部分的に正射影された距離画像のメッシュが複数生成されるが，まだ分割された状態であるので，最後にメッシュの隙間をドロネー三角形分割することによって接続し，複数の距離画像の統合を行っている．

これらメッシュベースの統合手法は，メッシュの境界部分の処理が難しく，距離画像に含まれる計測誤差や，残留している位置合わせ誤差の影響を受けやすいという問題がある．

1.2.3.2 ボリュームベースの統合手法

ボリュームベースの統合手法とは，距離画像をボリュームデータに落とし込むことで一つの 3 次元メッシュモデルへと再構築する方法である．ボリュームベースの統合手法は，元のメッシュ構造を考慮する必要性がなく，ロバスト性が高いため，現在はボリュームベースの統合手法が主流となっている．そこで本研究でもボリュームベースの統合手法を採用する．

以下に，ボリュームベースの統合手法の一般的な手順について述べる．ボリュームベースの統合手法においては，符号付距離場という一種の中間表現を用いる．すなわち，次の 2 ステップの変換処理を行うことで統合を行う．

1) 距離画像群から符号付距離場を生成する．
2) 符号付距離場を 3 次元モデルに変換する．

まず符号付距離場について説明し，符号付距離場を 3 次元モデルへと変換する手法について述べた後，関連手法を紹介する．

1.2.3.2.1 符号付距離場

符号付距離場とは，物体表面からの距離とその物体の内側にあるか外側にあるかを示す場である．この符号付距離場は次の 3 ステップによって生成される．

1) 3 次元空間上の各点において，その点から物体表面までの距離を絶対値で取得する．
2) 距離として参照した面の表裏より，その点が物体の内側にあるのか外側にあるのかを判別する．
3) その点が物体の内側にあれば負の符号，物体の外側であれば正の符号を距離に付与し，これをその点の符号付距離とする．

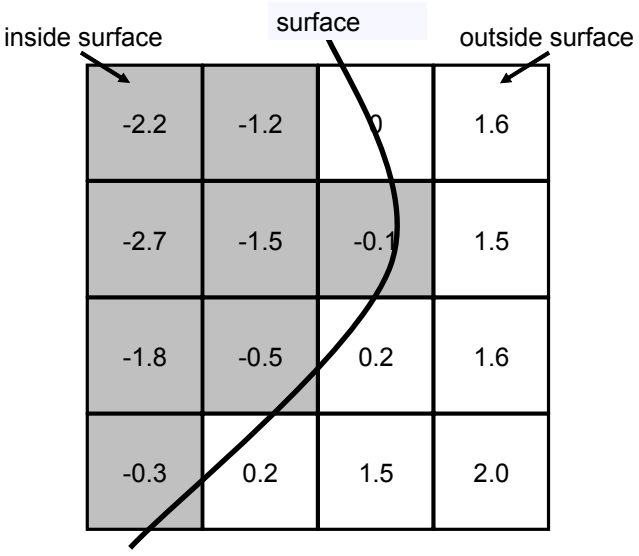

図 1.5 ボクセル表現における符号付距離場

これによって，符号付距離が空間に対して分布した，スカラー場が生成される．これが符号付距離場である．符号付距離場を $f(x)$ とすると，物体表面は $f(x)=0$ を満たす 2 次曲面の等値面として表され，0 等値面と呼ぶ．また，式の表現形式から，陰関数表面とも呼ばれる．

ボリュームベースの統合手法においては，空間を 3 次元格子状に区切ったボクセル表現を用いる．これは，3 次元空間を 3 次元格子状に区切ったものであり，2 次元画像におけるピクセル表現を 3 次元に拡張したものに相当する．ボクセル表現を用いた符号付距離場の生成は，各ボクセル中心からの符号付距離を求めていくことで行う．例えば図 1.5 では，空間を 16 のボクセルに区切り，各ボクセル中心から距離画像までの距離に応じて，符号付距離を求めて格納している．物体の内側に中心が存在するボクセルに関しては負の値が，外側にあるボクセルは正の値が格納されていることが分かる．こうすることで，各ボクセルに符号付距離が格納された形で離散的な符号付距離場を得ることができる．距離画像群が与えられた際に，ボクセル中心から各距離画像への最近傍点のうち，最も近いものをそのボクセルの符号付距離とするのが，一般的な方法である．

1.2.3.2.2　メッシュの生成

生成された符号付距離場から，3 次元モデルに変換するには，マーチングキューブ法[22]を用いる．マーチングキューブ法とは，隣接する 8 つのボクセル中心を頂点とした単位立方格子の各辺に対して，スカラー値が 0 となる位置に頂点を作り，それらを適当に結んで等値面を構成していくことで，全体のメッシュを生成する手法である．

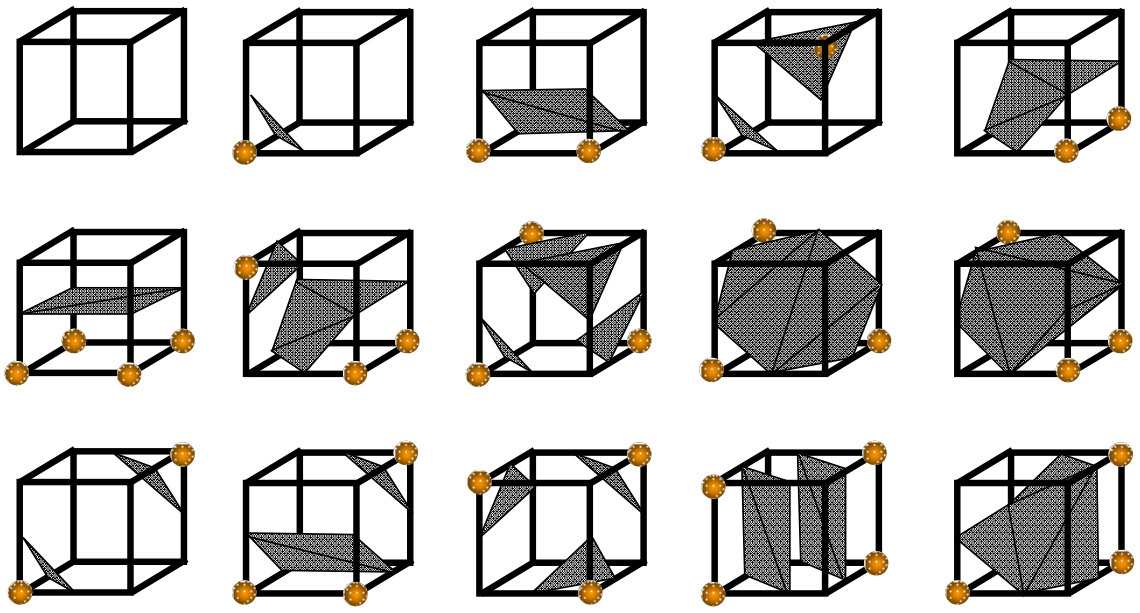

図 1.6 マーチングキューブ法のメッシュ生成パターン

言い換えれば，単位立方格子の各頂点が正か負の値を持ち，その符号が辺の両端で異なる際に，その値が 0 となる辺上の位置に頂点を作り，それらを結び，面を作ることとなる．この際，8 頂点の符号の取り方は 256 通りあることになるが，幾何学的に対称なものを除いていくと，実質的には図 1.6 に示した 15 通りに絞られる．単位立方格子の 8 頂点の符号を，この 15 種類のパターンに当てはめて，等値面を構成することで，効率的にメッシュを生成することができる．マーチングキューブ法は，その簡潔な結合方法ゆえに，ノイズとも取れる微小な三角形の生成や，微小な穴を生成してしまう問題がある．また，生成される面もボクセル内に一様な単一の面として簡略化されてしまうため，実際の物体より表現が省かれてしまう可能性もある．マーチングキューブ法を拡張させ，これらの問題に対応した手法もいくつか提案されている[23][24]．しかしながら，本研究では物体全体に対して非常に細かく分割を行うことを想定しているため，これらの問題は許容できると判断した．加えて，複雑な手法を採用することによる計算コストの増大よりも，大量にボクセルを分割することによる計算コストの増大の方が容易に削減できると見込み，マーチングキューブ法を採用した．

1.2.3.2.3 関連手法

以下に，ボリュームベースの統合手法をいくつか紹介する．

Hoppe らの手法

Hoppe らは，距離画像を 3 次元点の集合として見たときに，それに局所的な平面を当てはめることで，各ボクセルから平面までの距離を計算した[25]．最近傍点 $\mathbf{x}_i$ 付近の点群を接平面 $Nbhd(\mathbf{x}_i)$ としたときに，重心 $\mathbf{o}_i$ をその平面の代表点としている．すなわち，$\mathbf{o}_i$ までの距離を，符号付距離の絶対値として採用している．

符号の決定に関しては，平面の法線方向を用いている．法線は平面の主成分解析から求められている．

$$\mathrm{CV} = \sum_{\mathbf{y}\in Nbhd(\mathbf{x}_i)} (\mathbf{y}-\mathbf{o}_i)\otimes(\mathbf{y}-\mathbf{o}_i) \tag{1.1}$$

ここで，CV は共分散行列，⊗はベクトル外積を表している．この CV に対して，固有値 $\lambda_i^1 \geq \lambda_i^2 \geq \lambda_i^3$ と，それぞれに対応する固有ベクトル $\hat{\mathbf{v}}_i^1$，$\hat{\mathbf{v}}_i^2$，$\hat{\mathbf{v}}_i^3$ があるときに，法線として $\hat{\mathbf{v}}_i^3$，あるいは $-\hat{\mathbf{v}}_i^3$ を用いている．

以上より，点 $\mathbf{p}$ の符号付距離は，法線 $\hat{\mathbf{n}}_i$ を用いて以下のように表される．

$$\mathrm{dist}_i(\mathbf{p}) = (\mathbf{p}-\mathbf{o}_i)\cdot\hat{\mathbf{n}}_i \tag{1.2}$$

Hoppe らの手法は，単純な最近傍点を符号付距離に用いる場合に比べて，実際の物体に整合した計算になっていると言える．しかしながら，距離画像中のデータ全てを有効とみなしているため，距離画像の計測誤差や位置合わせ誤差の影響により，符号付距離を誤って

求めてしまう可能性がある．

Curless らの手法

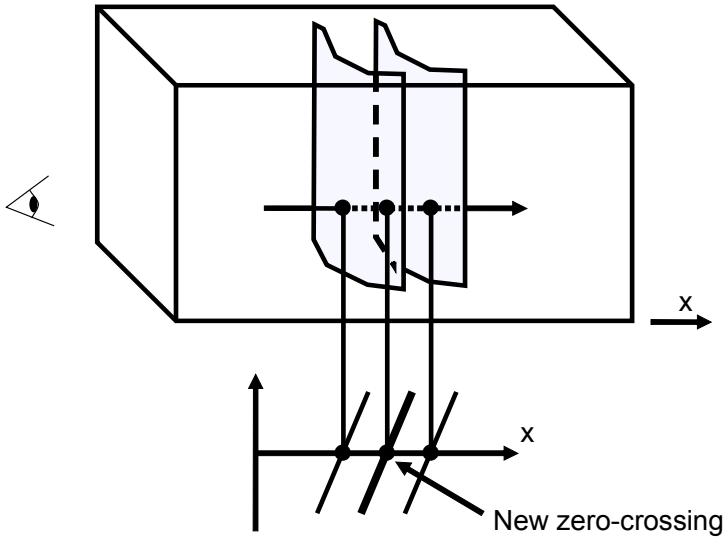

図 1.7 Curless らの手法

Curless らは，センサの視点を利用して符号付距離の計算を行った[26]．すなわち，ボクセルから距離画像の最近傍点として，ボクセルとセンサの視点を結ぶ視線と距離画像の交点を用いた．符号としては，ボクセルと距離画像とセンサの視点の位置関係を比べ，ボクセルが視点と距離画像の間なら正の符号を，距離画像がボクセルと視点の間にあるときに負の符号を付与した．

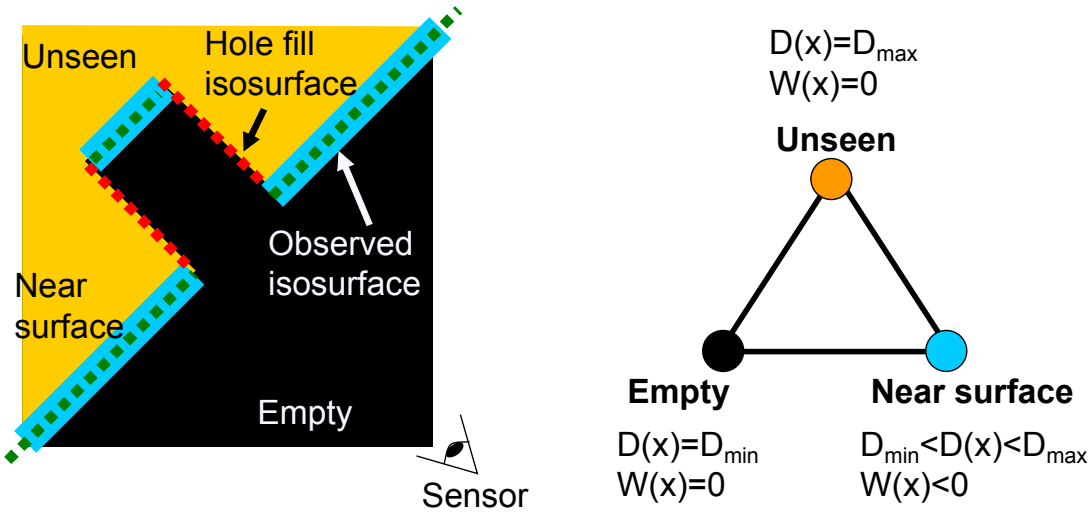

図 1.8 Curless らによる穴埋めの手法

また Curless らは，このセンサの視点を利用して，計測漏れによって生じる穴を埋める手法も提案している．これは各ボクセルを unseen，empty，near the surface に分類することで行っている．具体的には以下の手順で行う．

1) 全てのボクセルを unseen として初期化する．
2) 符号付距離の計算の際に表面付近として求まった部位を near the surface として分類する．
3) センサ位置と表面の間に位置するボクセルを empty に分類する．
4) unseen と empty が直接隣り合った部位に新たに表面を形成する．

Curless らの手法は，穴埋めを行うことで，よりなめらかなメッシュの生成を実現しているが，Hoppe らの手法と同様に，距離画像中のデータ全てを有効とみなしている．したがって，距離画像の計測誤差や位置合わせ誤差に影響されてしまうため，依然，符号付距離を誤って求めてしまう可能性が残っている．

増田らの手法

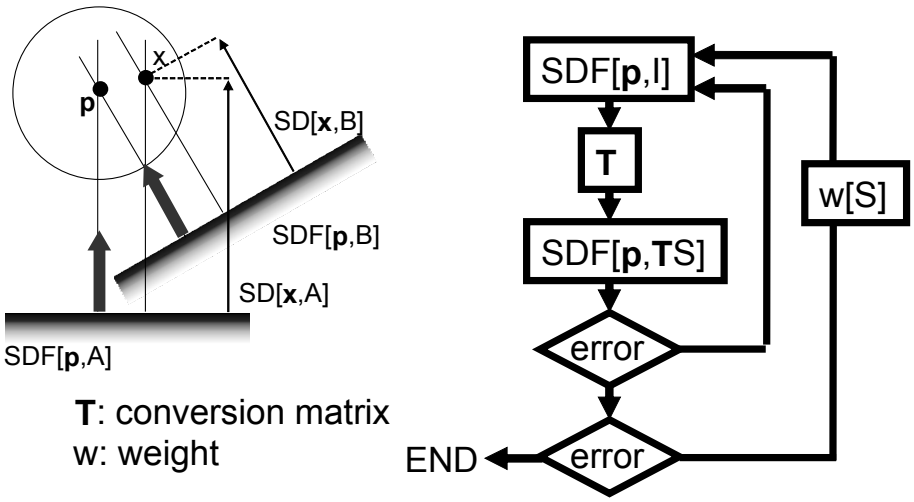

図 1.9 符号付距離場の計算と距離画像の位置あわせ

増田らは，各距離画像について求めた符号付距離場と法線場の値に，ロバスト統計手法を適用する統合手法を提案している[27]．近傍の場の値も一緒に用いることで，距離画像同士の重複が少ない場合に関しても，統計処理を容易に行えるようにしている．この手法では，符号付距離場による形状記述を用いて，統合した形状に距離画像を位置合わせし，ロバスト統計手法によって誤差の大きな計測点を除きながら処理を再帰的に繰り返すことで，位置合わせと符号付距離場の計算が同時に行えるようにしている．具体的には図 1.9 のように，距離画像ごとに符号付距離場を計算し，それらの合わせるような変換行列を再起計算によって求める．その変換行列をもって距離画像の位置を合わせることができると同時に，最終的な符号付距離場も求まるという仕組みである．

この手法は高精度かつロバスト性も高い手法であるが，複雑な計算のため処理に時間がかかり，大規模な距離画像に適用するには適していない.

Rocchini らの手法

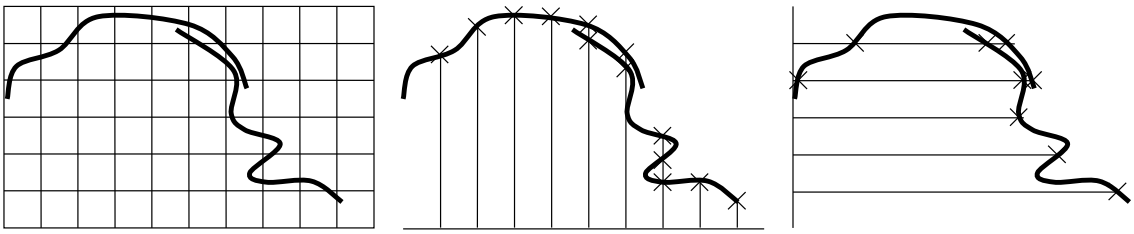

図 1.10 Marching Intersections における走査の様子

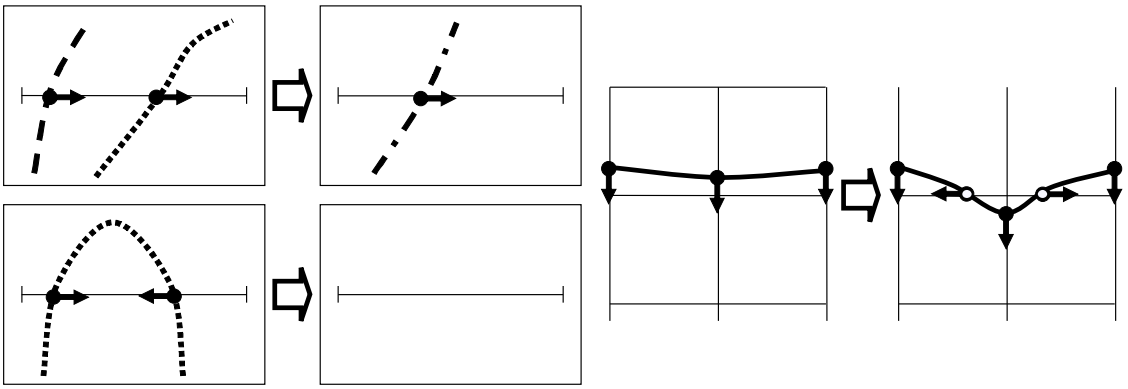

図 1.11 Marching Intersections における交点の移動と融合

また，符号付距離を用いないボリュームベースの統合手法として，Rocchini らが提案した Marching Intersections[28]を紹介する．この手法では，複数の距離画像が含まれる 3 次元空間に対して，縦方向と横方向において 3 次元格子状に走査線を走らせ，距離画像との交点を動かしながらパターンに合わせて融合させていくことで統合する手法である．それぞれの距離画像において，走査線と何回目に交わった点かでその移動させる向きを変更する．図 1.11 左上に示したように, 同一区画内に違う距離画像の交点が同じ向きに移動している場合には交点を融合する．図 1.11 左下に示したように，同一区画内に同じ距離画像の交点が向かい合う向きに移動している場合にも交点を融合する．図 1.11 右に示したように，交点の移動により区画をまたぐ際には，新たな交点を生成しながら移動を続ける．これを縦方向と横方向で行い，求まった交点を結び合わせることで，統合された 3 次元モデルを得ることができる．この手法は高精度な統合を行うことができるが，誤差に対するロバスト性が低いという問題点がある.

1.3 本研究の位置づけ

従来の手法を用いて文化遺産の 3 次元モデルを作成すると，統合の過程において，不均整な 3 次元モデルが生成され，膨大な計算時間がかかってしまう．文化遺産のように大規模で複雑な対象物の 3 次元モデルを作成するためには，様々な種類のレーザレンジセンサを用いる必要があり，それらの整合が上手く取れていないからだと考えられる．

そこで，本研究では従来の手法を拡張し，異種のレーザレンジセンサによって得られた距離画像を効率的に統合する手法を提案する．

1.2.3.2.3 で取り上げた Hoppe らの手法，Curless らの手法，Rochinni らの手法は誤差に対するロバスト性に問題がある．これらの手法は，距離画像中のデータ全てを有効とみなしているため，距離画像の計測誤差や位置合わせ誤差に影響されてしまうからである．そこで次章では，大規模な距離画像に適したロバスト性の高い符号付距離場の計算手法として，合致表面法を取り上げる．そして，異種のレーザレンジセンサによって得られた距離画像から均整の取れた 3 次元モデルを作成するために，合致表面法を拡張した，信頼度付き合致表面法を提案する．合致表面法では，全ての距離画像を同等に扱っていたため，精度の高いレーザレンジセンサによって得られた距離画像が，精度の低いレーザレンジセンサによって得られた距離画像に影響されてしまっていた．そこで，レーザレンジセンサの信頼度を導入することで，距離画像の重み付けをし，適合的な統合を行えるようにする．

また，文化遺産のような大規模な距離画像に対して符号付距離場の計算を行うと，非常に時間がかかってしまうという問題がある．例えば，1.2.3.2.3 で取り上げた増田らの手法は再帰計算を行っているため計算コストが高く，大規模な距離画像に対して用いることはできない．そこで，3 章では最近傍点探索の手法を，4 章では空間分割の手法を取り上げ，それらを拡張し，異種のレーザレンジセンサによって得られた距離画像の統合において，計算時間の短縮を実現する手法の提案を行う．

まず，最近傍点探索に対して，Grid Kd-tree を拡張した，Resolution based Grid Kd-tree を提案する．Grid Kd-tree は，距離画像を分割して PC クラスタに割り当てることで，最近傍点の探索空間を狭め，効率的な並列計算を行い，計算時間を短縮していた．しかし，解像度の異なる距離画像に対しても均一に分割していたため，距離画像によって分割されたサイズが異なり，効率的な探索が行われていなかった．そこで，データの存在する部分のみを，その距離画像の解像度に応じて細かさを変えて分割することで，効率的な並列計算が行えるようにする．

次に，空間分割に関して，八分木を用いた並列計算手法を拡張した，適応的空間分割法を提案する．従来の手法では，空間を均一に分割して PC クラスタに割り当てていたため，含まれる距離画像の数やサイズによって計算時間にばらつきが出て効率的な計算が行われ

ていなかった．そこで，距離画像の数やサイズを考慮し，符号付距離場の計算モデルを設定することで，PC クラスタへの割り当てに無駄が生じないような分割を実現する．

1.4 本論文の構成

第 1 章では，本研究の背景と目的，実物体から 3 次元モデルを生成するための 3 ステップについて説明した．その中でも，本研究が対象とするところを統合のステップと定めた．従来の統合手法では同種のレーザレンジセンサによって得られた距離画像を対象としていたために，異種のレーザレンジセンサによって得られた距離画像が混在する場合に適用すると問題が起こる．そこで，それに対応した手法を考案することを本研究の位置づけとした．その問題とは，精度の問題と，計算速度の問題である．

第 2 章では，精度の問題に関して述べる．本研究の基盤となる合致表面法の説明をした後，異種のレーザレンジセンサに対応した手法を提案する．そして提案手法の実験とその結果を示す．

第 3 章では，計算速度の問題のうち，最近傍点探索について述べる．まず本研究の基盤となる Kd-tree と Grid Kd-tree について説明をした後, Grid Kd-tree の評価実験の結果を示す．その後，異種のレーザレンジセンサに対応した手法を提案する．

第 4 章では，計算速度の問題のうち，空間分割について述べる．まず本研究の基盤となる八分木とその並列計算手法について説明をした後，異種のレーザレンジセンサに対応した手法を提案する．最後に，提案手法の実験結果を示す．

第 5 章では，本論文で述べた研究結果をまとめる．

2 信頼度付き合致表面法

統合における課題の 1 つは，生成される 3 次元モデルの精度である．レーザレンジセンサの計測誤差や位置合わせの誤差によって，均整の取れた 3 次元モデルが生成されなくなる可能性があるという問題である．本研究では精度の問題を解消する符号付距離場の計算手法として，信頼度付き合致表面法を提案する．これは同種のレーザレンジセンサによって得られた距離画像群を対象としていた合致表面法を，異種のレーザレンジセンサによって得られた距離画像群へと拡張した手法である．そこでまず以下において，合致表面法の説明を行う．その後，信頼度付き合致表面法について述べる．

2.1 合致表面法

本研究では合致表面法を基盤として用いる．まず符号付距離の誤推定がいかにして起こるかを説明した後，合致表面法について説明する．

2.1.1 符号付距離の誤推定

1.2.3.2.3 において述べたように，Hoppe らの手法や Curless らの手法を初めとする多くの手法は，距離画像中のデータ全てを有効とみなしているため，距離画像の計測誤差や位置合わせ誤差の影響により，符号付距離を誤って求めてしまう可能性がある．

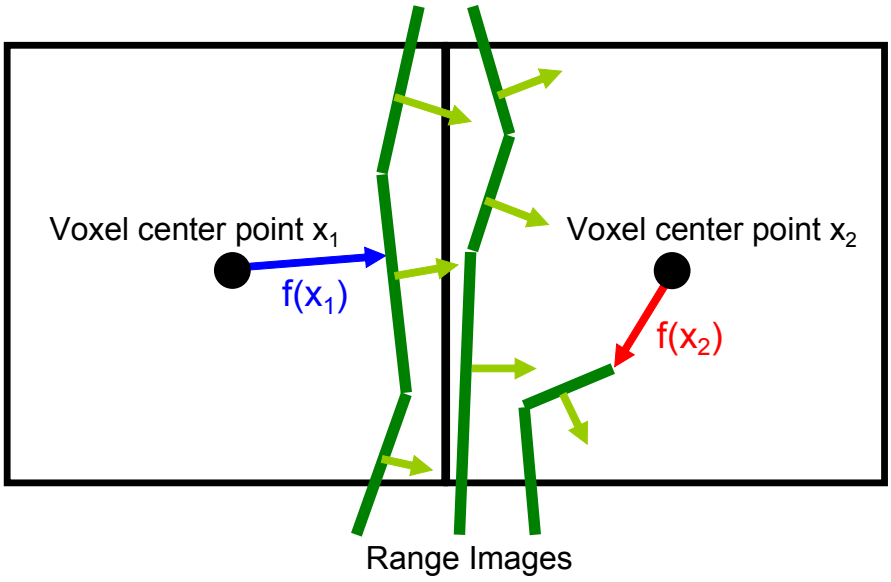

図 2.1 符号付距離の誤推定

誤推定がどのように起こるかを，図 2.1 に示した．図 2.1 は，ある隣接する 2 つのボクセルにおいて，3 枚の距離画像から，それぞれ符号付距離を計算する様子を示している．まず，左側のボクセルにおいては，3 枚の距離画像への最近傍点のうち，最も近かったものが青い矢印として示されている．また距離画像の法線の向きより，物体の内側にあることも分かる．従って，左側のボクセルの符号付距離は青い矢印で示された距離に，負の符号を付与したものとなる．これは推定が正常に行われた場合の例である．次に右側のボクセルにおいては，3 枚の距離画像への最近傍点のうち，最も近かったものが赤い矢印と示されている．しかしながら，ここで選択されている最近傍点は明らかにノイズであり，本来得られるべき距離よりも小さい値が得られてしまっていると言える．これは距離画像の端点が最近傍点として選ばれてしまっているからと考えることもできる．

このように，距離画像の端点やエッジの影響により，符号付距離の誤推定が起こりうる可能性がある．言い換えれば，入力距離画像の誤差に強く影響されてしまっていることが分かる．

2.1.2 合致表面法

符号付距離場のノイズに対するロバスト性を高める手法として，合致表面法という手法が提案されている[29]．合致表面法とは，閾値を満たす距離画像の中で，符号付距離の平均を取得することで，ノイズによる外れ値を除外するという手法である．本研究ではこの合致表面法を基盤として採用する．

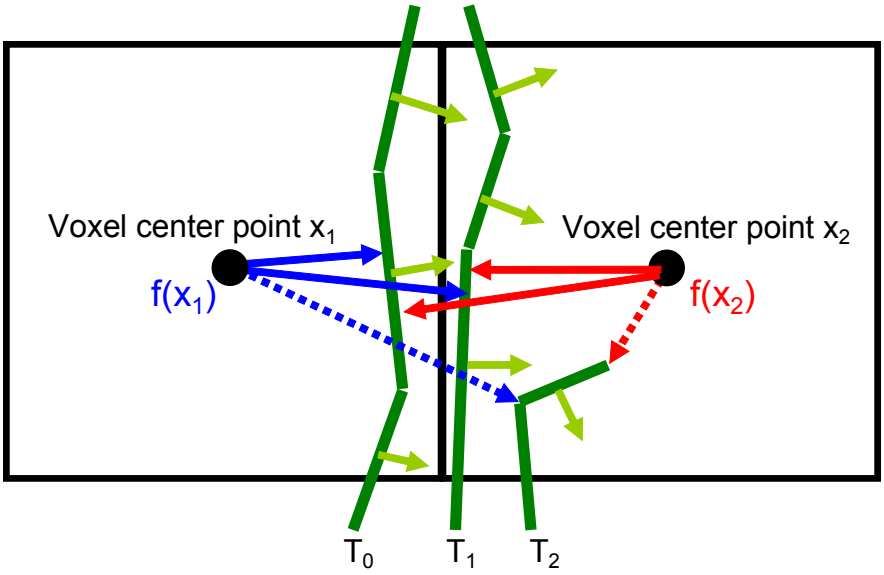

図 2.2 合致表面法

図 2.2 は合致表面法の様子を示した図である．まずボクセル中心 $\boldsymbol{x}$ から距離画像 T_0 の最近傍点 p_0 を探索する．次に，p_0 から他の距離画像 T_1，T_2，…の最近傍点 p_1，p_2，…を探索する．ここで，それらのが p_0 から充分近く，法線方向がほぼ同じであると判断される場合に, 同一の平面を表していると考えることができる. すなわち, 2 枚の距離画像において，次のような判別を行う．

$$\begin{aligned}&\mathrm{SameSurface}\left(\langle p_0, n_0\rangle, \langle p_1, n_1\rangle\right) = \\ &\begin{cases}\mathrm{True}\left(\|p_0 - p_1\| \le \delta_d\right) \wedge \left(n_0 \cdot n_1 \ge \cos\theta_n\right) \\ \mathrm{False\ otherwise}\end{cases}\end{aligned} \tag{2.1}$$

ここで，δ_d は距離の差の閾値，θ_n は法線方向の差の閾値である．これを満たす距離画像の表面を合致表面(consensus surface)と呼び, この合致表面に属する距離画像への最近傍点の平均を取得し，これを d_0 とする．

$$d_0 = \frac{\sum_{k=0}^{m} \mathrm{distance}(x_1, p_k)}{m} \tag{2.2}$$

ここで，m は合致表面に属する距離画像の枚数である. そして, 基準とする距離画像を T_0 から T_1，T_2，…と変えていき，同様の処理を行う．こうして求まった d_0，d_1，d_2，…の中で最も小さいものをそのボクセルの符号付距離として採用する．図 2.2 では実線の矢印が合致表面と判断され，点線の矢印が閾値を満たさず，ノイズとして除外されている．

以上のように, 合致表面法により，ノイズの影響が大きい点を除外できるようになった.

2.2 信頼度付き合致表面法

2.1.2 においてノイズの影響を加味しながら統合を行う合致表面法を紹介した．しかしながら, 合致表面法では, 異なる精度の距離画像が混在する場合に対応することができない. 精度の高いレーザレンジセンサによって得られた距離画像と，精度の低いレーザレンジセンサによって得られた距離画像を同等に扱っているため，精度の低い距離画像の影響によって，最終的に不均整な 3 次元モデルを生成してしまう可能性がある．そこでこの問題に対応した，信頼度付き合致表面法を提案する.

レーザレンジセンサの種類によって距離画像を区別するために，レーザレンジセンサの信頼度を導入する．これは，より解像度が高く精密に取得できるレーザレンジセンサによって計測され距離画像ほど，高い値となる．実際は，レーザレンジセンサとの距離等に応じて距離画像に含まれる各頂点にそれぞれ信頼度を設定することができるが，簡単のため距離画像 1 枚につき 1 つの信頼度として説明する.

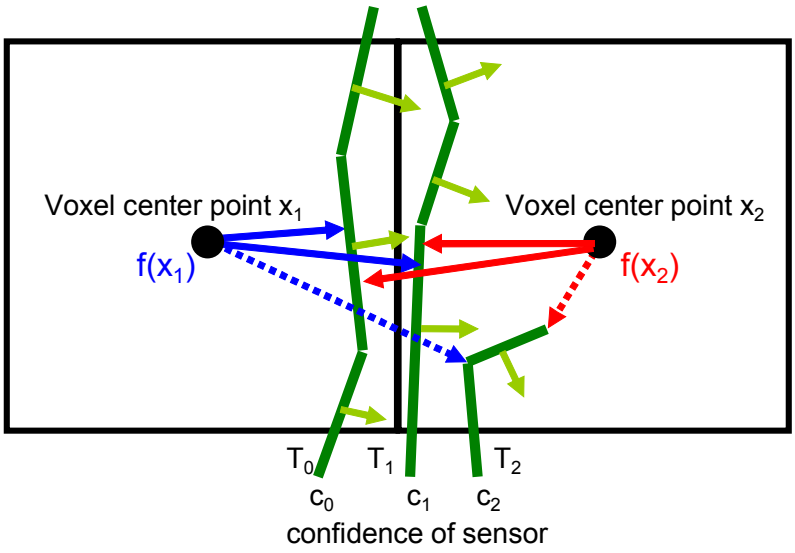

図 2.3 信頼度付き合致表面法

まず，2.1.2 と同様に(2.1)式を用いて，合致表面を求める．ただし，その判定条件の拡張を行う．すなわち，

$$\mathrm{SameSurface}\left(\langle p_0, n_0\rangle, \langle p_1, n_1\rangle\right) = \begin{cases} \mathrm{True}\ \left(\|p_0 - p_1\| \le f_d(c_0)\right) \wedge \left(n_0 \cdot n_1 \ge f_n(c_0)\right) \\ \mathrm{False\ otherwise} \end{cases} \tag{2.3}$$

ここで，c_0 とは距離画像 T_0 の信頼度であり，$f_d(c_0)$ ，$f_n(c_0)$ は c_0 に基づく可変閾値である．c_0 が大きいほど $f_d(c_0)$ は小さく，$f_n(c_0)$ は大きい．c_0 が小さいほど $f_d(c_0)$ は大きく，$f_n(c_0)$ は小さい．これにより，信頼度の低い距離画像が基準となる場合，合致表面とみなす範囲を広げることができるようになる．

そして，この合致表面に属する距離画像への最近傍点の信頼度による重み付き平均を求める．距離画像 T_0 を基準とした場合の式は以下のようになる．

$$d_0 = \frac{\sum_{k=0}^{m} c_k \cdot \mathrm{distance}(x_1, p_k)}{\sum_{k=0}^{m} c_k} \tag{2.4}$$

ここで，c_k とは距離画像 T_k の信頼度である．こうして求まった d_0，d_1，d_2，…の中から最小のものを符号付距離として採用する．なお，距離画像の信頼度は，レーザレンジセンサの精度などを考慮して決定する．

2.3 実験

合致表面法と，信頼度付き合致表面法との比較実験を行った．

2.3.1 実験データ

用いた実験データは以下のとおりである．尊顔と呼ばれるバイヨン寺院の各塔にある彫刻を対象としている．

	Data set 1	Data set 2	Data set 3
File size	94.8MB	61.9MB	32.1MB
Vertices	329177	4292232	2235984
Cyrax	2	3	1
FLRS	1	3	4

表 2.1 実験データ

ここで Cyrax とは Cyrax によって得られた距離画像を，FLRS とは，気球センサによって得られた距離画像の枚数を示している．(2.4)式における c_k として，FLRS によって得られた距離画像を 1.0，Cyrax によって得られた距離画像を 10.0 とした．これは Cyrax の精度が±4mm，FLRS の精度が±20mm 程度であることから，やや大きめに 10 倍の差をつけて設定している．

図 2.4，図 2.5，図 2.6 が実際に統合に用いた距離画像である．尊顔は塔の上部に位置しているため，Cyrax のデータでは物体の上面の一部が計測できず穴となっている．この部分を FLRS のデータによって補完するのが目的である．

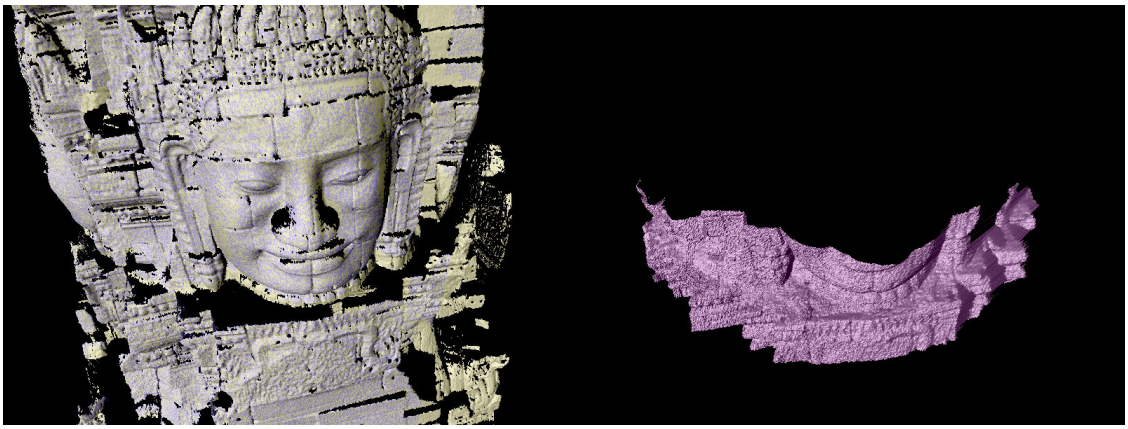

図 2.4 Data set 1 : Cyrax のデータ(左)と FLRS のデータ(右)

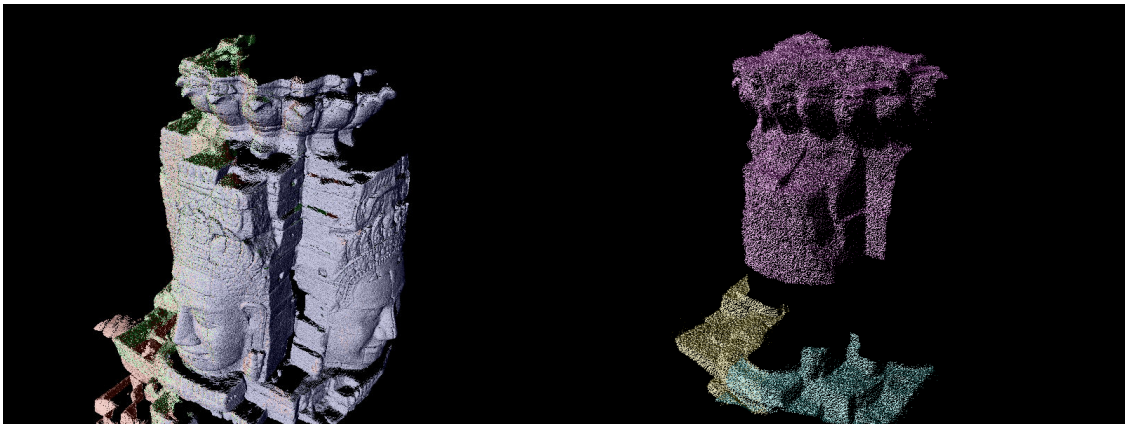

図 2.5 Data set 2 : Cyrax のデータ(左)と FLRS のデータ(右)

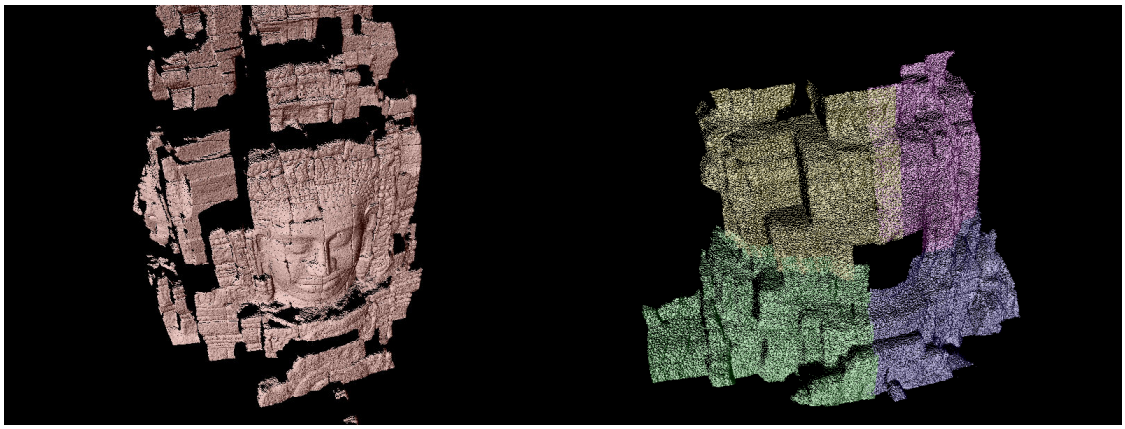

図 2.6 Data set 3 : Cyrax のデータ(左)と FLRS のデータ(右)

2.3.2 実験結果と評価

各手法を用いて統合を行った結果，Data set 1 については図 2.7，図 2.8，Data set 2 については図 2.9，図 2.10，Data set 3 については図 2.11，図 2.12 のようになった．図を見て分かるとおり，従来の合致表面法ではモデルの表面にノイズが乗っているのに対し，信頼度付き合致表面法では均整の取れたモデルが生成されていることが分かる．

図 2.7，図 2.9，図 2.11 を図 2.4，図 2.5，図 2.6 と比較すると，合致表面法を用いた場合では，FLRS のデータが使われている部分において，不均整な表面が形成されてしまっていることが分かる．一方で，図 2.8，図 2.10，図 2.12 を図 2.4，図 2.5，図 2.6 と比較すると，信頼度付き合致表面法では，Cyrax のデータとの整合性をうまく取ることで，均整の取れた表面が生成されていることが分かる．

図 2.7 Data set 1 について合致表面法を用いて統合を行った結果

図 2.8 Data set 1 について信頼度付き合致表面法を用いて統合を行った結果

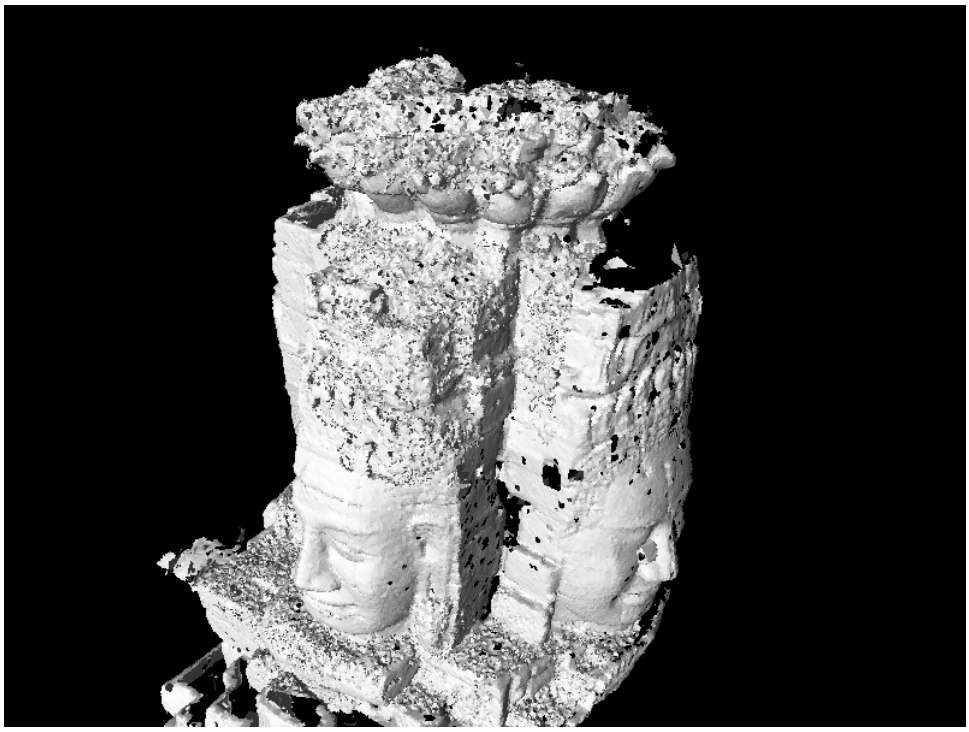

図 2.9 Data set 2 について合致表面法を用いて統合を行った結果

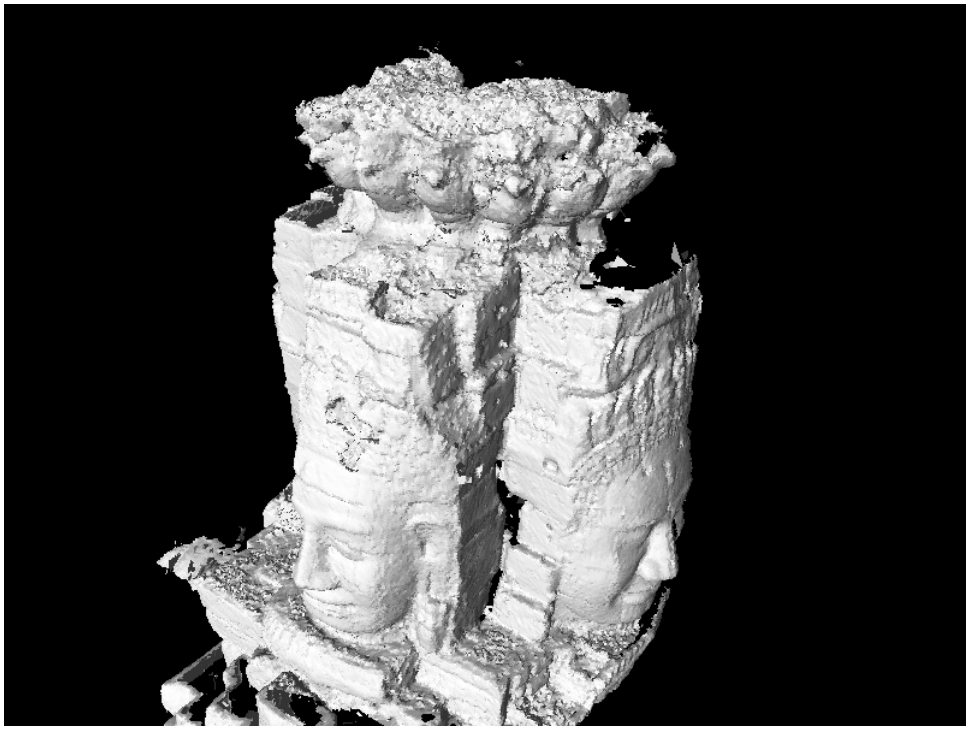

図 2.10 Data set 2 について信頼度付き合致表面法を用いて統合を行った結果

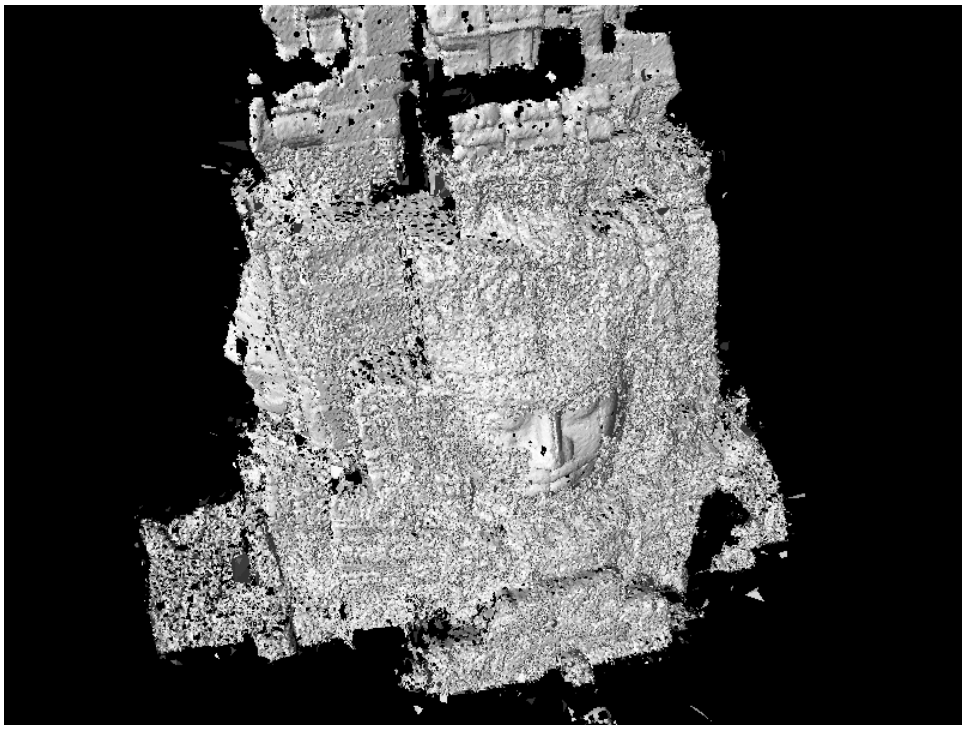

図 2.11 Data set 3 について信頼度付き合致表面法を用いて統合を行った結果

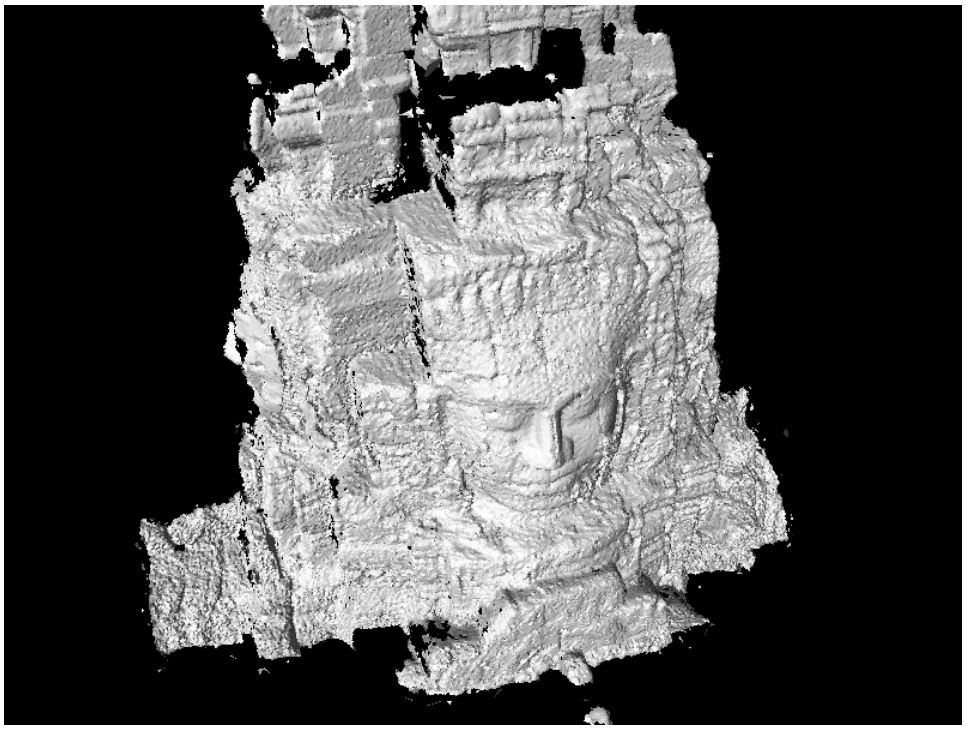

図 2.12 Data set 3 について信頼度付き合致表面法を用いて統合を行った結果

3 Resolution based Grid Kd-tree

統合における課題の 1 つは，符号付距離場の計算時間である．符号付距離の計算には多くの時間が必要となるため，PC クラスタを用いて並列計算を行う様々な手法が提案されている．本研究においても，PC クラスタを用いた並列計算を行う．

符号付距離の計算に多くの時間が必要となる原因の 1 つは，最近傍点探索である．合致表面法では全ての距離画像に対して最近傍点を探索しないといけないため，効率的に探索が行える手法が必要となってくる．

最近傍点探索を効率的に行う手法として，ハッシュやインデックスを作る手法が提案されている[30][31]が，これらの手法では計算コストを抑えることができるものの，膨大なインデックス表を作るためにメモリ空間を多く必要とする．本研究においては，大規模な距離画像を対象としているため，これらの手法を使ってインデックス表を作るのはメモリ空間の制約から現実的ではない．

その他の手法として，階層的なデータ構造を用いた手法が提案されている．Kd-tree[32]，quadtree[33]，k-d B tree[34]， R-tree[35]などが存在する．いずれも木構造に細かな差異はあるものの，探索方法は似通っている．一方で，大規模な距離画像に適用するために，PC クラスタを用いた並列計算を行う際に，効率的な探索が行えるように Kd-tree を拡張した Grid Kd-tree が提案されている[36]．Grid Kd-tree は，距離画像を均等に分割し，分割されたそれぞれの距離画像に対して Kd-tree を形成することで，PC クラスタへの割り当てを行っている．しかし，異なる解像度の距離画像が混在する際には，この Kd-tree の大きさに偏りが生じ，計算時間が増大してしまうという問題がある．そこで本研究では，異種のレーザレンジセンサによって得られた解像度の異なる距離画像が混在するデータ群に対して，効率的な探索を行えるように Grid Kd-tree を拡張した Resolution based Grid Kd-tree を提案する．

3.1 Kd-tree を用いた最近傍点探索

符号付距離の計算において，全ての頂点を探索するのは現実的でない．そこで，よく用いられているのが Kd-tree という 2 分木のデータ構造である．空間を再帰的に分割していくことで木構造を形成し，ボクセル中心付近の部分空間を探索範囲として木構造を辿ることで効率的な探索を行う．

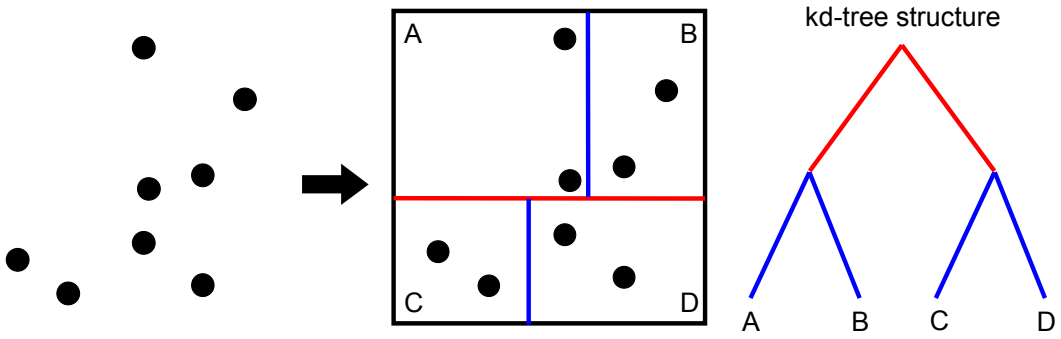

図 3.1 Kd-tree の生成

Kd-tree は以下のように作成される．

1) 空間全体を均等に2つに分割し，それぞれの部分空間を子ノードとする根ノードを作る．
2) 分割された部分空間をさらに 2 つに分割し，それぞれの部分空間を子ノードとしてこのノードに加える．
3) 部分空間内に含まれる頂点数が閾値以下になるまで 2)を再帰的に繰り返す．

このようにして，末端の葉ノードには点群が，枝ノードには空間を分割する敷居の情報が書かれたインデックスが格納された木構造が作成される．例えば図 3.1 左のような頂点群からなる距離画像が与えられた場合を考える．まず空間を上下 2 つに分割し，それぞれの部分空間に頂点が半分ずつ含まれるようにする(図 3.1 真中，赤線)．そして，木構造を作り進める(図 3.1 右，赤線)．次に，分割によってできたそれぞれの部分空間を，含まれる頂点数ができるだけ均等になるように分割し(図 3.1 真中，青線)，木構造を作り進める(図 3.1 右，青線)．仮に部分空間に含まれる頂点数の閾値が 2 だとすれば，それぞれの部分空間の分割はここでストップする．それぞれの部分空間のインデックスを得た上で，葉ノードに格納することで，図 3.1 右のような木構造ができあがる．

Kd-tree における最近傍点探索は次のような手順で行われる．

1) 入力された頂点座標がどの葉ノードの範囲に属するか，インデックスを参照しながら辿っていく．
2) 葉ノードが特定できたら，その葉ノードに含まれる頂点に対して総当りで最近傍点探索を行う．

例えば図 3.1 の Kd-tree で探索を行うとする．仮に最近傍点探索を行うボクセル中心の頂点座標が A に含まれるとしたら，まず木構造を根ノードから辿り A を探索する．そして，A に含まれる頂点 2 つから最も近いものを探すことなる．頂点数を n としたとき，単純な全探索の計算コストは $O(n)$ となるのに対して，Kd-tree では木の深さが $\log(n)$ になると考えれば $O(\log(n))$ となる．

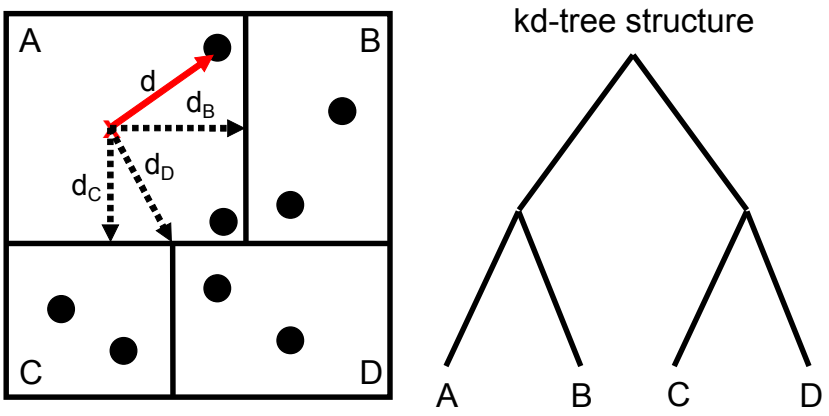

図 3.2 Kd-tree の境界問題

しかしながら，Kd-tree の特性上，1 つの葉ノード内で探索が終わらない可能性がある．それは図 3.2 のような場合である．前述した手順により，最近傍点が葉ノード A 内で見つかり，その距離が d だったとする．この時，A に隣り合う他の葉ノード B，C，D それぞれまでの最短距離 d_B，d_C，d_D と d を比較して，d の方が大きかった場合，その葉ノード内についても探索しないといけない．これは，他の葉ノードに含まれる頂点までの最短距離が d より小さくなる可能性があるからである．このようにして，最悪の場合には探索コストが $O(n)$ にまでなってしまう可能性がある．

Kd-tree を用いずに符号付距離場を計算した場合のコストについて考える．まず，単純な最近傍点探索について考える．符号付距離を計算するボクセル数を N_v，頂点数 N の距離画像 M 枚が与えられたと考える．すると，それぞれのボクセル中心から，入力された距離画像全てに対して最近傍点探索を行うことになるため，計算コストは次のようになる．

$$N_v \times M \times N \tag{3.1}$$

次に，合致表面法を用いる場合を考える．合致表面法では，それぞれのボクセル中心から，距離画像までの最近傍点を求め，その最近傍点から他の距離画像までの最近傍点を求める処理を行う．したがって計算コストは次のようになる．

$$N_v \times M^2 \times N \tag{3.2}$$

今度は，Kd-tree を用いて合致表面法を行った場合の計算コストについて考える．最近傍点探索においては，前述したとおり $O(\log(n))$ になるため，(3.1)式は次のようになる．

$$N_v \times M \times \log N \tag{3.3}$$

合致表面法では，それぞれのボクセル中心から距離画像までの最近傍点を求め，その最近傍点から他の距離画像までの最近傍点を求める処理を行うので，(3.2)式は次のようになる．

$$N_v \times M^2 \times \log N \tag{3.4}$$

3.2 Grid Kd-tree

大規模な距離画像を統合するには Kd-tree を用いたとしても非常に時間がかかる．また，距離画像全体を一度にメモリに乗せきることができないため，そもそも探索が行えないという問題点があった．そこで中尾らは，これらの問題を解決するために，Grid Kd-tree という Kd-tree を拡張したデータ構造を提案した．符号付距離場の計算に Grid Kd-tree を使うことによって， PC クラスタを用いた並列計算を行うことができる．その具体的な手順は以下のとおりである．

1) 空間をいくつかの部分空間に分割し，PC クラスタに割り当てる．
2) その部分空間に対応する Grid Kd-tree のデータを割り当てる．
3) その部分空間の符号付距離場を，割り当てられたデータを用いて計算する．
4) 全ての部分空間について求まった符号付距離場を統合する．

すなわち，符号付距離場の計算を分担して行うという手法である．Grid Kd-tree は，距離画像のうち符号付距離場の計算の際に必要となる部分を，PC に割り当てるためのデータ構造である．

Grid Kd-tree は以下の手順によって作成される．

1) PC クラスタに渡されるサブボクセルの空間に応じて，距離画像を分割する．
2) 分割された各々の距離画像に対して独立に Kd-tree を作成する．
3) サブボクセル空間の位置と各々の Kd-tree を対応させたインデックスを作成する．この時，距離画像が含まれていないサブボクセル空間には Kd-tree を生成せず，Kd-tree が存在しないことを表す値を与える．
4) 3)で作成したインデックスに，それぞれの Kd-tree の根を対応付ける．

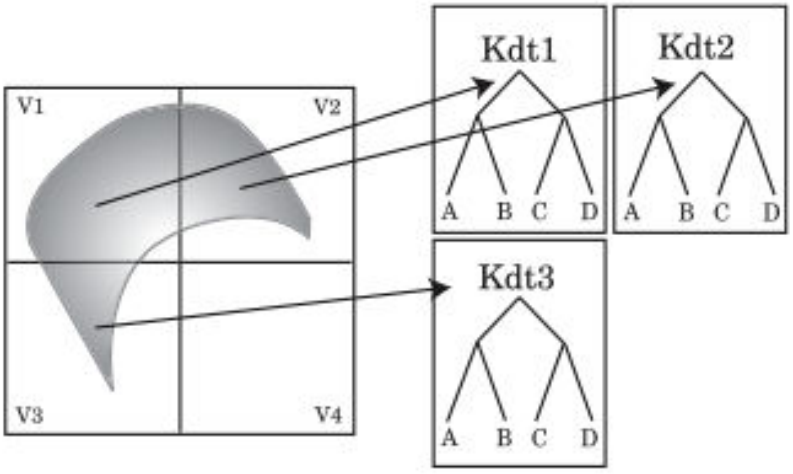

図 3.3 Grid Kd-tree における距離画像の分割と Kd-tree の生成

ファイル構造としては，分割された Kd-tree の 3 次元空間上の位置とファイル上の位置が記述されたヘッダ部, 分割された Kd-tree が連続して記述されるデータ部で構成されている．

3.2.1 計算量削減

Grid Kd-tree は 2 つ点において，計算量を削減している．1 つは，最近傍点探索において探索対象となる頂点数の削減である．頂点数が N の距離画像に対して，サブボクセル V_1，V_2，V_3，V_4 において最近傍点探索を行う場合を考える．Kd-tree を用いた場合，それぞれのサブボクセルから距離画像全体を探索する必要があるため，計算コストは，

$$4\log N \tag{3.5}$$

となる．Grid Kd-tree を用いた場合，分割によってサブボクセル V_1，V_2，V_3，V_4 に割り当てられる部分空間に含まれる頂点数をそれぞれ n_1，n_2，n_3，n_4 とすると，

$$\log n_1 + \log n_2 + \log n_3 + \log n_4 + 4k \tag{3.6}$$

となる．ここで k は分割された Kd-tree の根を見つ出すコストであり，このコストは N が十分大きい場合無視することができるため，(3.5)式より小さいと考えることができる．

もう 1 つは，探索候補となる距離画像の枚数の削減である．M 枚の距離画像から生成された M 枚の Grid Kd-tree のインデックスを探索し，サブボクセル空間に Kd-tree が含まれる場合はその Kd-tree を読み込み，含まれない場合は何も読み込まない．これにより，サブボクセルに読み込まれる Kd-tree の枚数は M 枚より少なくなる．

3.2.2 メモリ消費量の削減

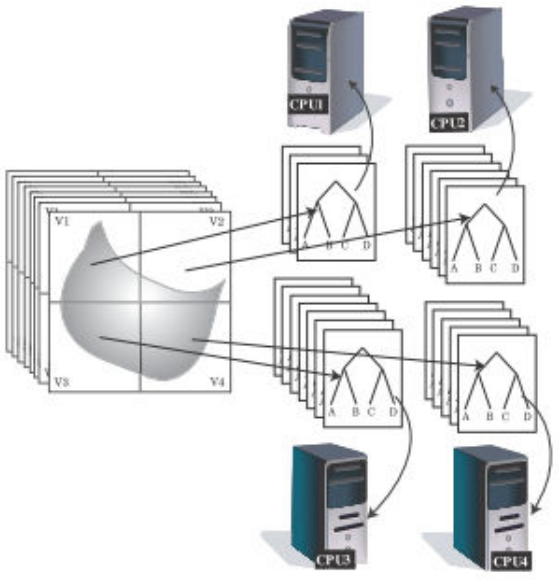

図 3.4 Grid Kd-tree における分割された Kd-tree の割り当て

合致表面法では，互いの距離画像を比較する処理が含まれるため，全ての距離画像をメモリに読み込む必要があるため，非常にメモリの消費量が大きくなる．Grid Kd-tree では，サブボクセル空間ごとに分割された Kd-tree を読み込むことで，メモリ消費量の削減を実現している．また，サブボクセル空間ごとに分割するため，サブボクセル単位で PC クラスタに割り当てる並列計算においても，用いることができる．

3.2.3 境界問題

あるボクセル中心からの最近傍点が，それを含む部分空間内にあるとは限らない．3.1 において説明した Kd-tree における現象もこの 1 つであり, Grid Kd-tree にもこの問題は発生する．部分空間の幅を w とすると，得られた最近傍点までの距離の絶対値 d が，

$$\frac{1}{2}w < d < \frac{\sqrt{3}}{2}w \tag{3.7}$$

を満たす場合，より近い点が部分空間外にある可能性があると言える．

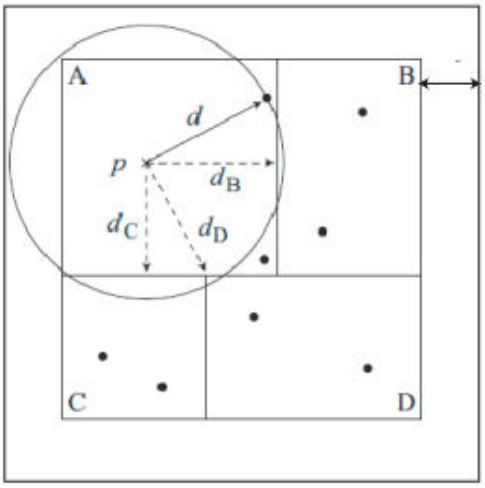

図 3.5 Grid Kd-tree における分割された Kd-tree の境界線の拡張

Kd-tree ならば木を辿って他のノードを探索すればよいが，Grid Kd-tree においては，そもそも部分空間外については，メモリ上に読み込まれていない．そこで中尾らは，分割を行う際に，Kd-tree 作成の対象とする部分空間の大きさを拡張した．その拡張幅は(4.7)式に従うと $\sqrt{3}w$ となるが，マーチングキューブ法で符号付距離場からメッシュに変換するため，最小単位となるボクセルで考えればよい. そのボクセル幅を w' とすれば, 拡張幅は $\sqrt{3}w'$ となる．しかしながら，計測に用いたレーザレンジセンサの誤差 ε の 2 倍の拡張幅も必要であり，これは w' より十分大きいと考えられる．したがって，2ε の拡張を行えば十分ということになる．

3.3 Grid Kd-tree の評価実験

3.3.1 実験環境

本研究は東京大学情報基盤センターのスーパーコンピュータである HA8000 クラスタシステムを利用して実験を行った．今回実験を行った環境は表 3.1 のとおりである．Linux オペレーティングシステム上で動作し，ネットワーク通信インターフェースとして MPI1.2 を用いることができる．

図 3.6 HA8000 クラスタシステム

CPU	AMD Quad-Core Opteron 8356 (2.3GHz)
node	Max 16 node
Memory	4 [GB/node]
OS	RedHat Enterprise Linux 5

表 3.1 実験環境

3.3.2 実験データ

File size	26GB
Vertices	1929847215
Range images	2235

表 3.2 実験データ

本研究で対象としているバイヨン寺院に対して実験を行った．[36]では小規模な実験デー

タに対しての考察は行われているが，バイヨン寺院に対する実験データが不十分であり，評価がなされていない．そこで表 3.2 のような実験データに対して実験を行った．

ここでは，ボクセルの解像度として 2 通り設定し(表 3.3)，それぞれにおいて PC の台数，Grid Kd-tree における分割深度，計算時間を測定した．ボクセルの解像度とは符号付距離を格納するボクセルの細かさのことであり，最終的な出力の解像度に影響する．また，分割深度とは Grid Kd-tree における距離画像の分割の度合いを示す値である．分割深度が D の場合，それぞれの距離画像は 1 辺につき 2^D の分割がなされ，PC クラスタに割り当てるサブボクセルの総数は $2^{3D} = 8^D$ となる．なお，計算時間としては，Grid Kd-tree データ構造に変換された距離画像に対して，符号付距離場を計算する時間を測定した．すなわち，Grid Kd-tree データ構造に変換する時間や，符号付距離場から 3 次元メッシュモデルへと変換する時間は含まれない．

Resolution 1	26.8 cm on a side of voxel
Resolution 2	6.7 cm on a side of voxel

表 3.3 ボクセルの細かさ

また，異種のレーザレンジセンサによって得られた距離画像として以下のようなデータに対して実験を行った．

Data size	2.98GB
Range images	410
VIVID	171

表 3.4 異種のレーザレンジセンサによるデータ

ここで，VIVID とは VIVID によって得られた距離画像の枚数を意味する．

3.3.3 実験結果と評価

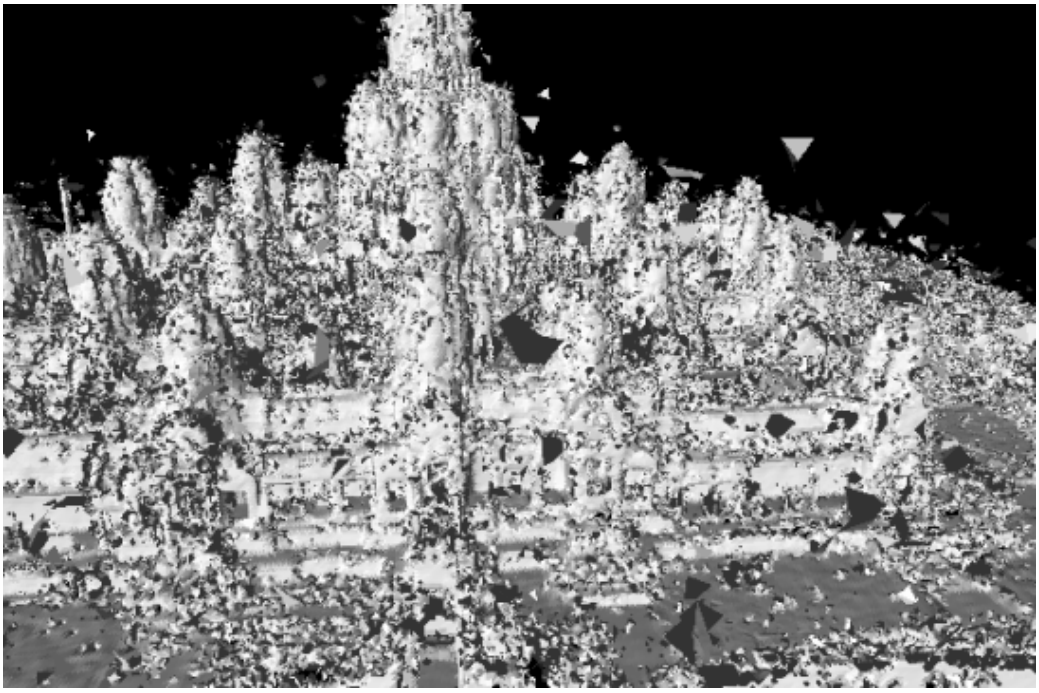

図 3.7 Resolution 1 の統合結果

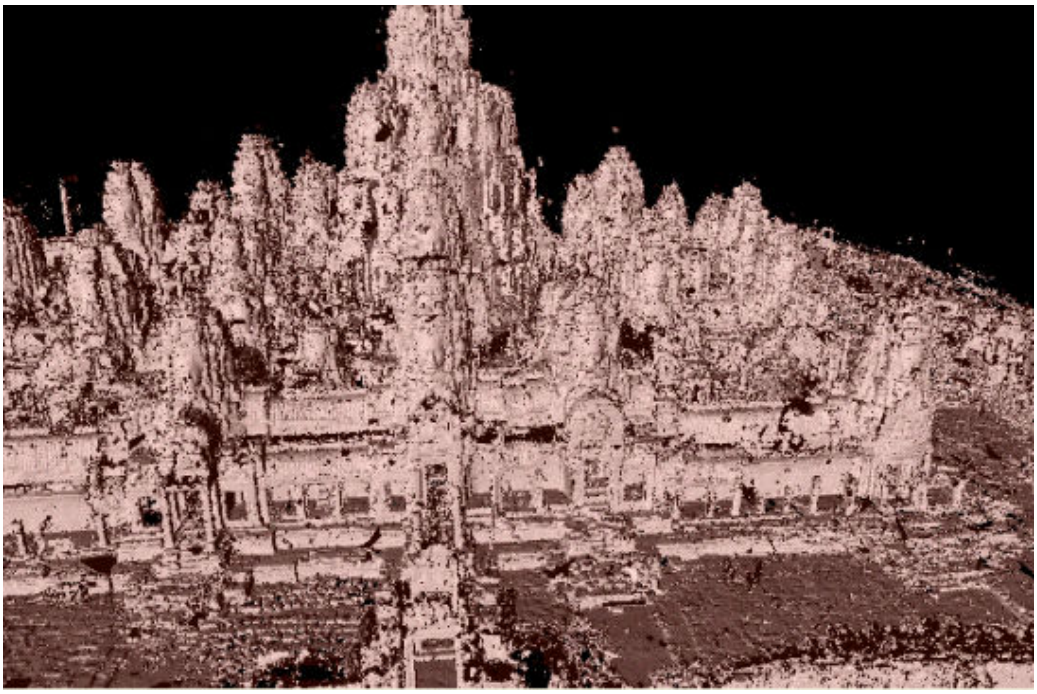

図 3.8 Resolution 2 の統合結果

各ボクセル解像度における統合の結果は図 3.7，図 3.8 のようになった．空間にノイズと見られる 3 角パッチが浮遊しているが，レーザレンジセンサの精度に対してボクセルの解像度を大きく設定して実験を行ったためであると考えられる．今回は計算時間の実験のためこのような設定にしたが，実際に統合を行う際には，レーザレンジセンサの精度に合わせてより細かく 3 次元モデルを生成することになるため，このような問題は生じない．

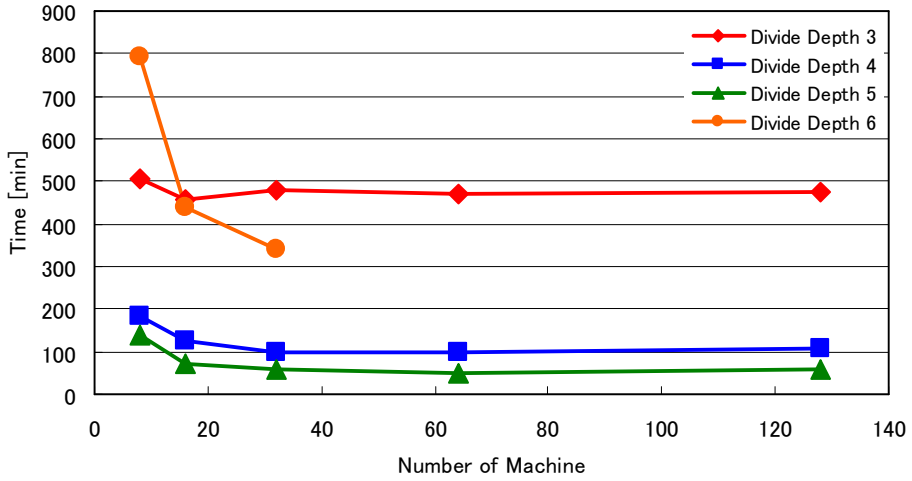

図 3.9 Resolution 1 における計算時間

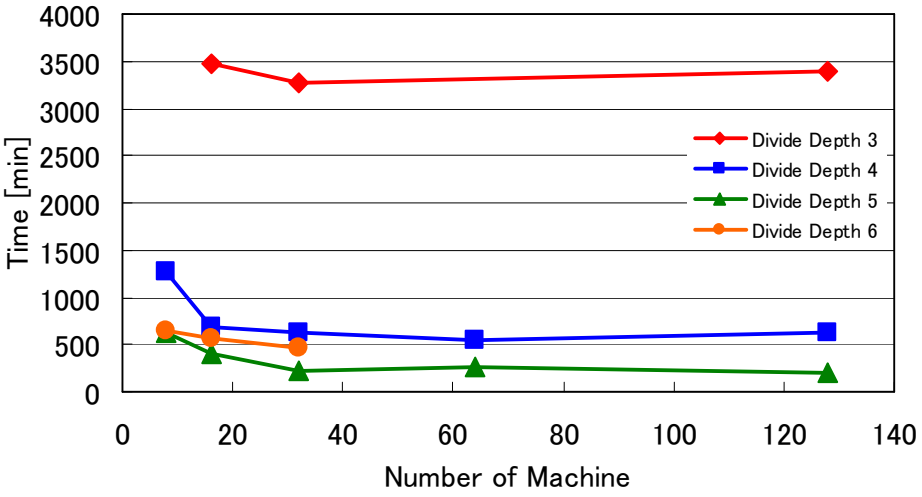

図 3.10 Resolution 2 における計算時間

計算時間の実験結果は図 3.9，図 3.10 のようになった．ここで Divide Depth とは Grid Kd-tree の分割深度のことである．図を見て分かるとおり，分割深度を上げていくと，計算時間は短くなるが，図 3.9 においては分割深度 5 以降で，図 3.10 においては分割深度 6 で

計算時間が上昇してしまっていることが分かる．サブボクセルの大きさの差異により，特定の PC のみ計算が終わっていないということが起きうるため，通常，サブボクセルを細かく分割することで計算時間は短縮される．しかしながら，分割深度を深くすると必ずしも計算時間が短縮されるわけではない．これは，サブボクセルの数が増えると，ネットワークコストを初めとするオーバーヘッドが増大してしまうからであると考えられる．これらの理由により，データによって最適な分割深度が存在することとなる．

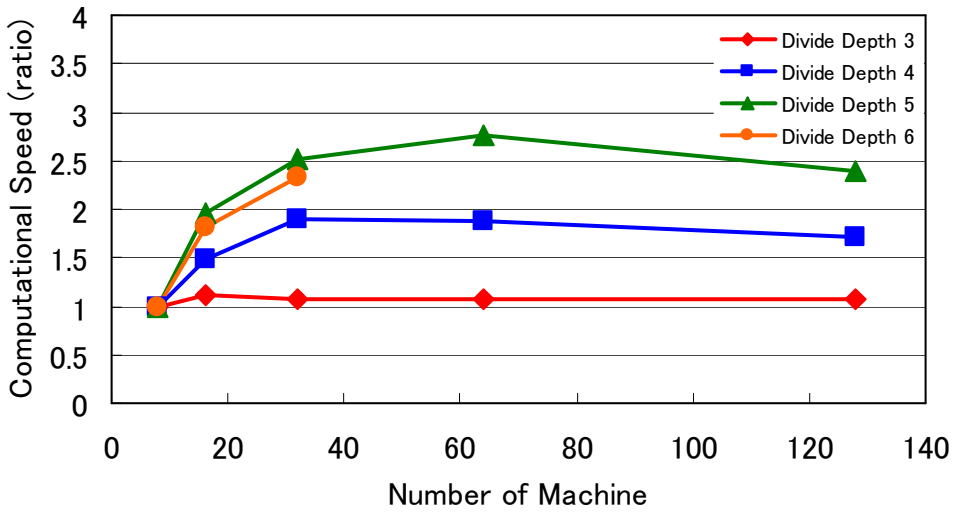

図 3.11 Resolution 1 における計算速度

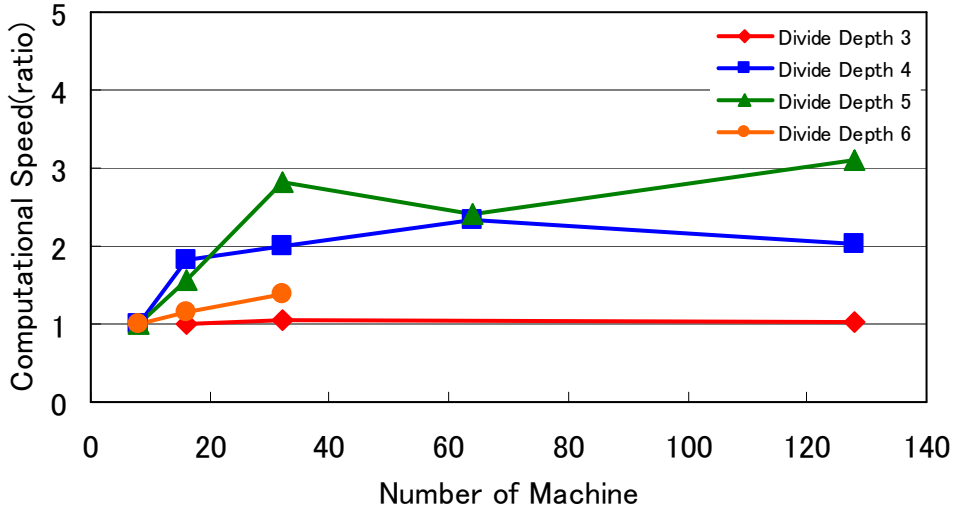

図 3.12 Resolution 2 における計算速度

前述したオーバーヘッドの影響を見るために，図 3.11，図 3.12 のようなグラフを作成した．これは，それぞれの分割深度において，PC が 8 台の時の計算時間を基準としたときの計算時間の比率である．理想的な環境において，このグラフは直線となるはずである．しかしながら, その上昇は緩やかであり , PC32 台あたりで頭打ちとなっていることが分かる．これらが，ネットワークコスト等のオーバーヘッドの影響と考えられる．PC の台数を増やすにつれて，符号付距離の計算時間は短くなる．しかしながら，サブボクセル 1 つあたりのオーバーヘッドは一定であると考えられるため，相対的にオーバーヘッドの影響が大きくなっていく．これらの理由により，緩やかなカーブを描く結果となったと想定される．

次に，表 3.4 のデータに対して実験を行った．

図 3.13 はサブボクセルに含まれる頂点が多い順にソートした，サブボクセルごとの計算時間である．グラフの左端を見ると，距離画像を多く含むサブボクセルの計算時間が大きくなっていることが分かる．単純にサブボクセルを細かく分割することでは，計算時間を効率的に短縮することはできないことが分かる．したがって，様々な種類のレーザレンジセンサを用いて得られた大規模な距離画像に適用するためには，効率的な分割を行う手法が必要となる．

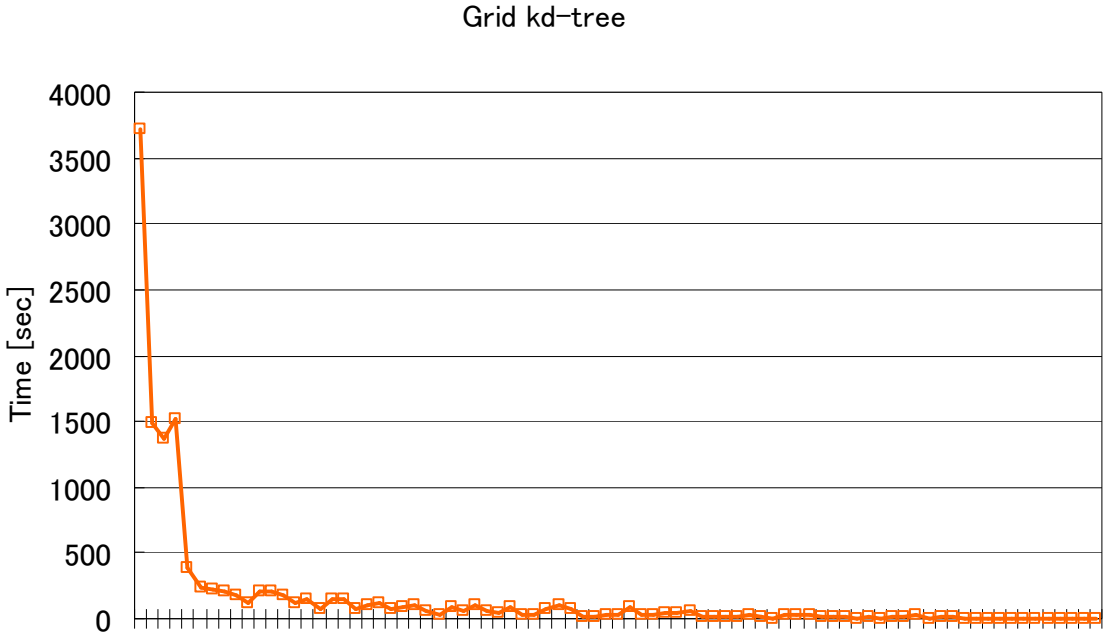

図 3.13 サブボクセルごとの計算時間

3.4 Resolution based Grid Kd-tree

大規模な物体を対象とした場合，距離画像 1 枚 1 枚に含まれる頂点群は，空間全体に対して非常に小さく局在している．Grid Kd-tree では，これに合わせて距離画像を細かく分割するとしても，空間全体に対して等間隔に分割する必要があり，冗長である．また，解像度の異なる距離画像が与えられた場合，片方に合わせて分割を行ったとしても，他方が適当に分割されない場合があり，計算コストをうまく削減できない可能性がある．本研究で主な対象としている文化遺産のように複雑で大規模な物体の場合，様々なレーザレンジセンサによって取得された解像度の異なる距離画像が混在しており，Grid Kd-tree だけではうまく計算コストを削減できず，また冗長なデータ構造となってしまう可能性が考えられる．そこで, 本研究ではこれらの問題に対応すべく, Grid Kd-tree を拡張した Resolution based Grid Kd-tree を提案する．

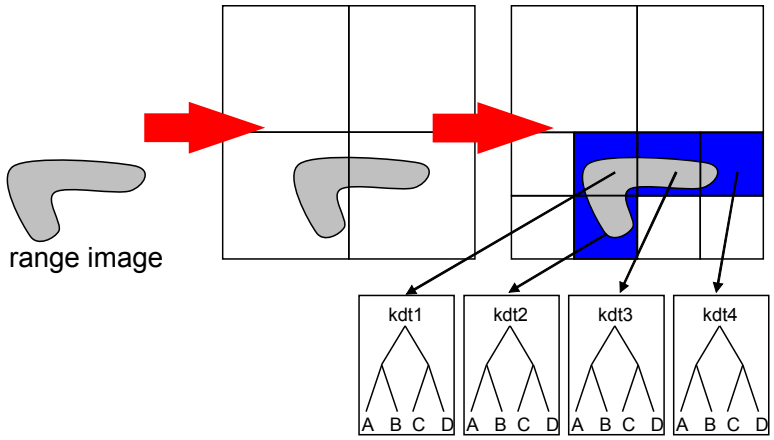

図 3.14 Resolution based Grid Kd-tree の作成手順

Resolution based Grid Kd-tree は以下の手順によって作成される．

1) Grid Kd-tree と同様に，PC クラスタに渡されるサブボクセル空間に応じて距離画像を分割する．
2) 1)で分割された各々の距離画像に対して，そこに頂点が存在するならば，その距離画像の解像度に応じて再度分割を行う．
3) 2)で分割された各々の距離画像に対して，Kd-tree を生成する．
4) 2)で分割された距離画像の 3 次元空間上の位置と各々の Kd-tree を対応付けたインデックスを作成する．
5) 1)で分割された距離画像の 3 次元空間上の位置，すなわち PC クラスタに渡すサブボクセル空間の位置と，4)で作成されたインデックスを対応付けたインデックスを作成する．
6) 4)で作成したインデックスに，それぞれの Kd-tree の根を対応付ける．

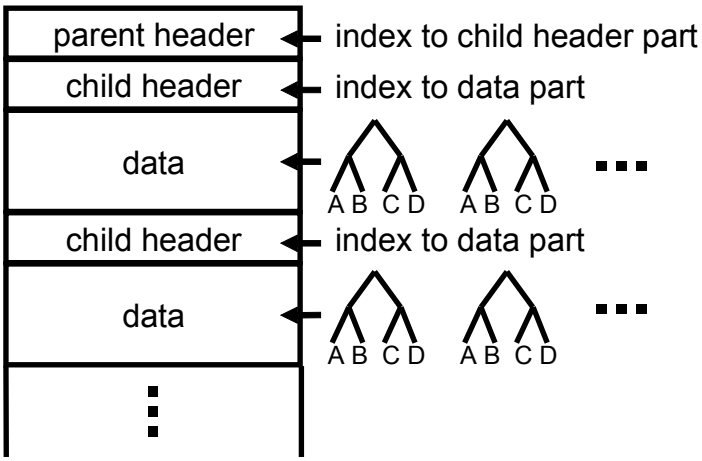

図 3.15 Resolution based Grid Kd-tree のデータ構造

ファイル構造としては，PC クラスタに渡すサブボクセルの 3 次元空間上の位置とファイルの位置が記述された親ヘッダ部と，分割された Kd-tree の 3 次元空間上の位置とファイルの位置が記述された子ヘッダ部と，分割された Kd-tree が連続して記述されるデータ部で構成されている．

Resolution based Grid Kd-tree では，2 段階で分割を行うことにより，空間全体に対して距離画像を均等に細かく分割することなく，距離画像が存在する部分空間のみを細かく分割することが可能になった．1)で PC クラスタに渡すサブボクセル空間に応じて分割を行っているため，図 3.16 のように，これを単位に PC クラスタに割り当てることで，3.2.2 で述べたような並列計算が行える．ひとたび割り当てが行われれば，3)で生成された Kd-tree は同一 PC 内に存在するため，距離画像によってその分割の細かさが異なったとしても，必要とする部位をその都度参照することができる．

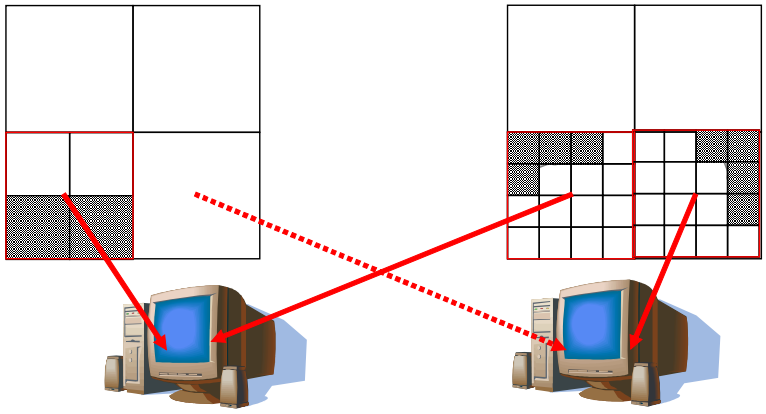

図 3.16 Resolution based Grid Kd-tree における PC クラスタへの割り当て

3.4.1 計算量削減

この手法では，2 つの点において，並列計算時の計算時間を削減できると考えられる．1 つは，距離画像の分割である．Grid Kd-tree では解像度が高い距離画像を効率的に分割するためには，空間全体を細かく分割する必要があった．これでは，ヘッダ部のインデックスが冗長になるだけでなく，割り当てる部分空間の数が多くなりすぎてしまうため，ネットワークコストが大きくなってしまう可能性がある．それに対して Resolution based Grid Kd-tree では，距離画像のうち，PC クラスタに割り当てるサブボクセル空間に含まれる部分も，解像度に応じて分割することができるため，ネットワークコストにとらわれることなく，探索範囲を狭めることができる．

もう 1 つは，異種のレーザレンジセンサによって得られた距離画像が混在する場合である．Grid Kd-tree では，複数の解像度が混在する距離画像群が与えられた際に，荒い解像度の距離画像に合わせて分割を行うと，細かい解像度の距離画像に対して生成される Kd-tree が大きくなってしまう．逆に細かい解像度の距離画像に合わせて分割を行うと，荒い解像度の距離画像を分割しすぎてしまい，3.2.3 における境界線の拡張等に起因するオーバーヘッドが増大し，計算時間が増えてしまう可能性がある．それに対して Resolution based Grid Kd-tree では，解像度に応じて分割の細かさを変えることができるため，分割された Kd-tree のサイズを異なる距離画像間でもほぼ均等に保つことができ，オーバーヘッドとの兼ね合いを適当に保つことができる．

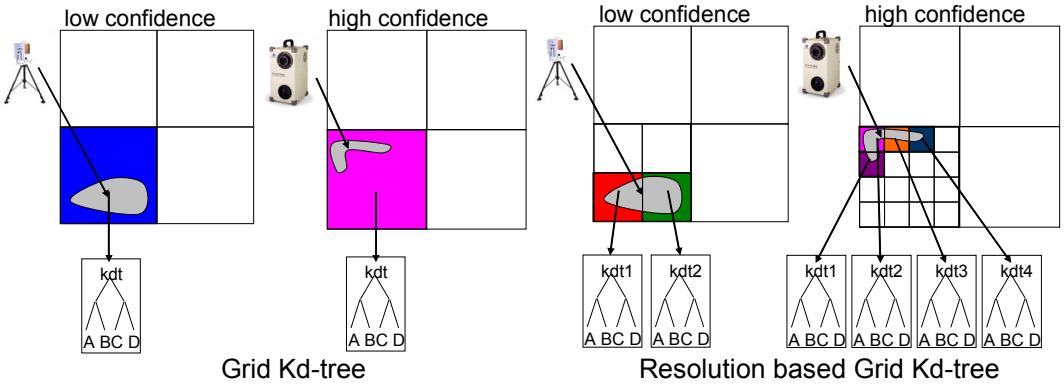

図 3.17 Grid Kd-tree と Resolution based Grid Kd-tree の比較

3.5 実験

計算時間の評価を行うため実験を行った．提案手法である Resolution based Grid Kd-tree を用いて実験を行い，Grid Kd-tree と比較する．

なお，実験は 3.3.1 に示した環境で行った．

3.5.1 実験データ

計算時間の評価を行うため，以下のデータに対して実験を行った．Data set 1 と Data set2 はバイヨン寺院の回廊の一部を切り出したものである．Data set 3 はペディメントと呼ばれる彫刻のデータであり，Data set4 は尊顔のデータである．

	Data set 1	Data set 2	Data set 3	Data set 4
Data size	2.98GB	15.0GB	2.65GB	0.15GB
Imager, Cyrax	239	1309	0	5
VIVID	171	159	208	0

表 3.5 実験データ

ここで，Imager，Cyrax，VIVID とはそれぞれ Imager，Cyrax，VIVID によって得られた距離画像の枚数を意味する．VIVID によって得られた距離画像は，非常に高解像度の距離画像となっているため，二段階目の分割深度に差をつける．その基準として，分割された距離画像に含まれる頂点数が大体同じ大きさとなるように定める．これは距離画像の空間的な大きさと頂点数を比較することで容易に行える．ここで，二段階目の分割深度とは図 3.14 右で行われる分割の細かさのことであり，距離画像によってその細かさを変えることができる．その定義は 3.3.2 と同様であり，二段階目の分割深度が D の場合，一段階目の分割によって作られたサブボクセルに対して，1 辺につき 2^D の分割がなされ，分割数は $2^{3D} = 8^D$ となる．また計算時間の定義も同様であり，Resolution Grid Kd-tree データ構造に変換された距離画像について，符号付距離場を計算する時間を測定した．

比較対象としては Grid Kd-tree を用いる．PC の台数を 64 に固定し，それぞれの手法において分割深度を変えながら実験を行った．

3.5.2 実験結果と評価

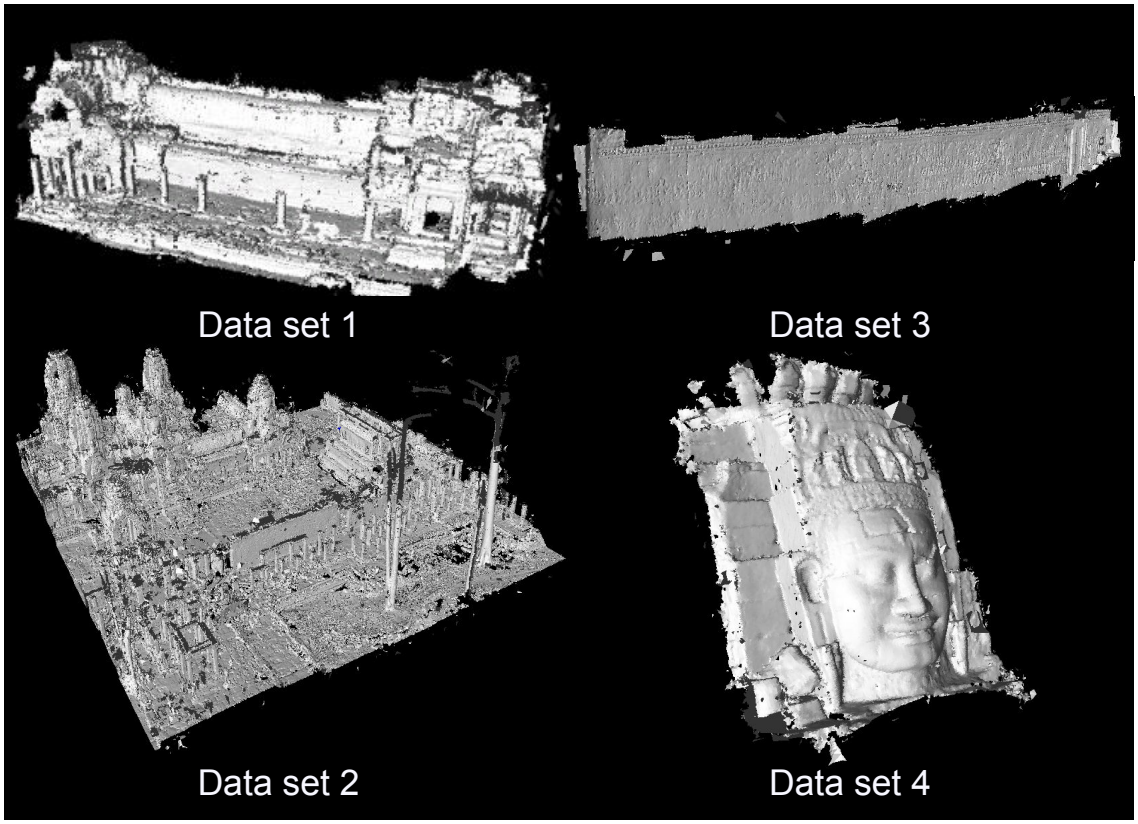

図 3.18 Data set 1 ~ Data set 4 の統合結果

Grid Kd-tree		Resolution based Grid Kd-tree			
Divide depth	Time [min]	Divide Depth	High Res	Low Res	Time [min]
3	390	3	2	1	154
4	72	4	2	1	42
5	120	5	2	1	56
6	1008	6	2	1	530

表 3.6 Data set 1 における分割深度と計算時間

統合によって得られた 3 次元モデルを図 3.18 に示した．表 3.6 は，Data set 1 における Grid Kd-tree と Resolution based Grid Kd-tree の計算時間の比較である．High Res とは解像度の高い距離画像，すなわち VIVID によって得られた距離画像の二段階目の分割深度を意味し，Low Res はそれ以外の距離画像における二段階目の分割深度を示している．ここで High Res を 2，Low Res を 1 とした理由は，距離画像に含まれる頂点数にある．VIVID によって

得られた距離画像において，距離画像 1 枚あたりの頂点数は約8.96×10^5であり，それ以外のレーザレンジセンサによって得られた距離画像に関しては約2.46×10^5となっており，およそ 3.6 倍の差がついている．3.3.2 で述べたように，分割深度 1 につき，分割数は 8 倍になるため，分割深度の差は 1 で十分である．また，図 3.16 で示したように二段階目に分割された距離画像は全て同じ PC に割り当てられるので，ネットワークコストは考慮しなくていいが， 3.2.3 で述べた Grid Kd-tree の境界線の拡張に影響されるため，二段階目の分割深度を大きくしすぎても意味が無い．これらの理由により，表 3.6 のような分割深度において実験を行った．

表 3.6 を見ると，Grid Kd-tree，Resolution Grid Kd-tree の両方において分割深度 4 が最短の計算時間となっており，前者は 72 分，後者は 42 分，と 30 分の短縮が見られる．その他の分割深度においても，Resolution based Grid Kd-tree の計算時間が短くなっている．最短の計算時間となった分割深度が同じ理由として，割り当てるサブボクセルの大きさがその分割深度において最適となったことが考えられる．すなわち，割り当てたサブボクセル内での符号付距離計算の差が結果に表れたと言える．

	Data set 1	Data set 2	Data set 3	Data set 4
Grid Kd-tree	72	82	9	2.28
Resolution based Grid Kd-tree	42	60	10	2

表 3.7 Grid Kd-tree と Resolution based Grid Kd-tree の計算時間 (min)

図 3.19 Grid Kd-tree と Resolution based Grid Kd-tree の計算時間

それぞれのデータにおける計算時間の結果は表 3.7，図 3.19 のようになった．Data set 3 以外において計算時間の短縮が見られる．Data set 3 は表 3.5 のとおり，全て VIVID によって得られた距離画像である．したがって，解像度の差が表れず，データ構造が冗長になってしまっていたことが原因となり，計算時間が Grid Kd-tree よりも長くなってしまったことが考えられる．したがって，複数のレーザレンジセンサによって得られた距離画像に対しては，Resolution based Grid Kd-tree を用いることで計算時間を短縮することができた．

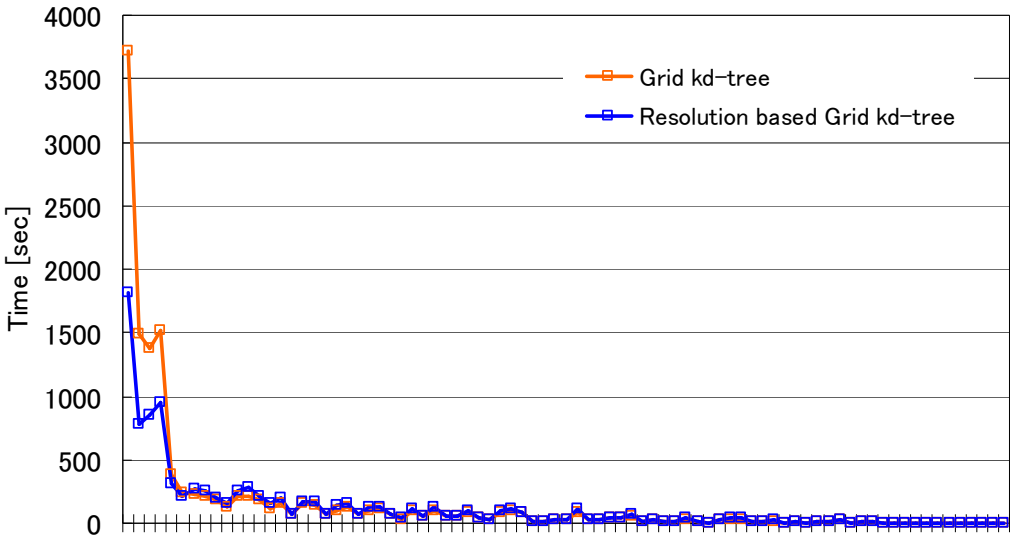

図 3.20 Data set 1 におけるサブボクセルごとの計算時間

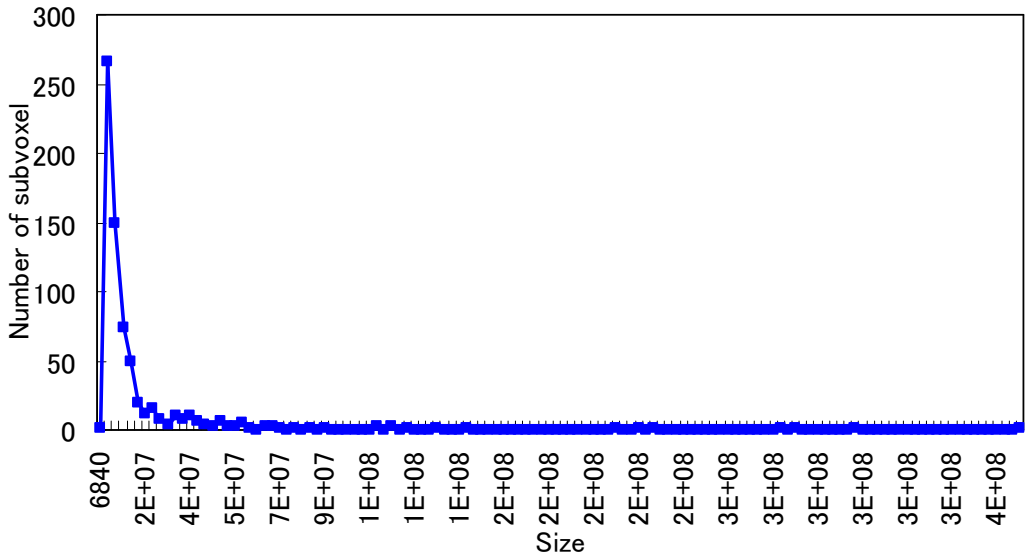

図 3.21 Data set 2 におけるサブボクセルに含まれる距離画像の大きさ

図 3.20 は Data set 1 において，サブボクセルに含まれる頂点が多い順にソートした，サブボクセルごとの計算時間である．グラフの左端を見ると，距離画像を多く含むサブボクセルの計算時間が大きくなっていることが分かる．Grid Kd-tree と比較して，Resolution based Grid Kd-tree では，サイズの大きなサブボクセルの計算時間が短縮されていることが分かる．しかしながら，他のサブボクセルと比較するといまだ計算時間が突出して大きいことが見て取れる．

図 3.21 は Data set 2 における，サブボクセルに含まれる距離画像の大きさのヒストグラムである．多くのサブボクセルが小さい大きさであるが，非常に多くの頂点を含むサブボクセルがいくつか存在していることが分かる．

図 3.20，図 3.21 より，サブボクセルによって含まれる頂点数に差ができ，計算時間に大きなばらつきができることがわかる．Resolution Grid Kd-tree を用いるだけでは，これらの大きさの差異を完全には吸収することはできず，特定の PC のみ計算が残っているという状況を引き起こす危険性がある．

4 適応的空間分割法

ボクセル表現において符号付距離場を作るには，全てのボクセルにおいて符号付距離の計算，すなわち最近傍点探索を行うことになる．そのため，効率的に空間を分割しなければ，計算時間が増大してしまう．そこで，空間を分割してボクセルを作る際に，再帰的な分割に置き換えて木構造とし，符号付距離の計算と並行してボクセルを作っていくことで，表面付近のみを細かく分割することができる八分木がよく用いられている．またこれを PC クラスタに割り当てて並列化する手法も提案されている．しかし，異なる解像度の距離画像が混在する際には，サブボクセルに含まれるデータ量に偏りが生じる．そのため，均等に分割を行うと，オーバーヘッドの影響で計算時間が増大してしまうという問題がある．そこで本研究では，サブボクセルに含まれるデータ量に応じて分割深度を決定する適応的空間分割法を提案する．

4.1 八分木を用いた空間分割

空間全体を均等に 3 次元格子状に分割してボクセルを作成し，その各々のボクセルに対して符号付距離を計算するのは非常に効率が悪い．マーチングキューブ法でメッシュを生成するためには，表面付近の符号付距離さえ分かれば十分である．そこで，符号付距離場の演算コスト削減のためのデータ構造として八分木がある[37]．八分木の生成方法は次のとおりである．

1) 根ノードを作成する．
2) 空間全体を 8 分割し，それぞれのボクセルの符号付距離を計算する．ボクセルの位置と符号付距離を格納したノードを作成し，子ノードとして根ノードに追加する．
3) 2)で求まった符号付距離が閾値内であるボクセルを 8 分割し，それぞれのボクセルの符号付距離を計算する．ボクセルの位置と符号付距離を格納したノードを作成し，子ノードとして 2)のノードに追加する．
4) 同様に処理を繰り返す．

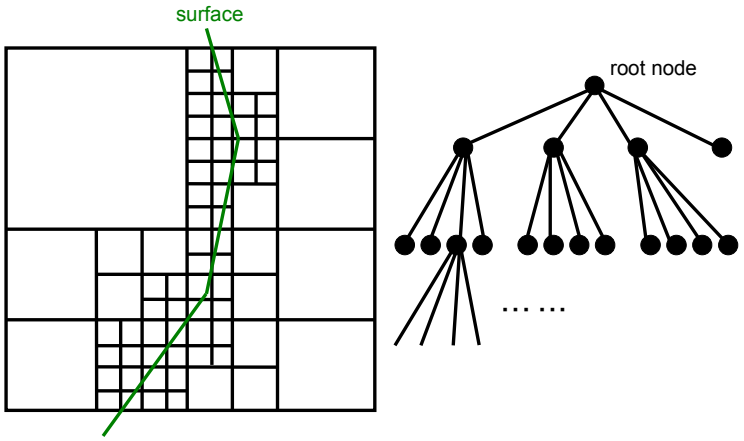

図 4.1 八分木の生成

これにより, 陰関数表面が 0 付近のボクセルのみ細かく分割された 8 分木が生成される．ここで，3)の閾値との比較は次の式に従って行う．

$$|f(x)| < \frac{3\sqrt{3}}{2} w \tag{4.1}$$

ここで$|f(x)|$は符号付距離の絶対値，wはボクセルの幅である．この式を満たすとき，陰関数表面がボクセル内にあることが考えられるため，ボクセルを分割する．このように再帰的に符号付距離を求めながら分割を行っていくことで, 8 分木に対応付けられた符号付距離場を得ることができる．表面を含む全てのボクセルのサイズは等しくなるため，あとはそのままマーチングキューブ法を適用するだけで，3 次元モデルを得ることができる．

4.2 八分木を用いた並列計算

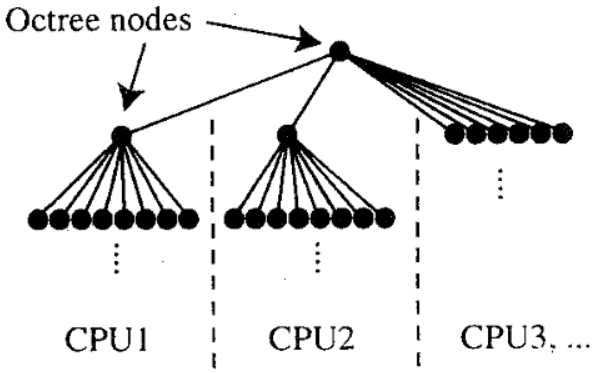

図 4.2 八分木を用いた並列計算

佐川らは，八分木を用いた符号付距離場の計算を並列化する手法を提案している[38]．八分木を用いて符号付距離場を計算する際には，ボクセルを再帰的に分割しながら木を生成していくため，各々の兄弟ノードの子ノード以降の分割や計算はそれぞれ互いに独立に行うことが可能である．したがって，分割されたサブボクセルをそれぞれ別の PC クラスタに割り当てることで，並列処理を行うことができる．その具体的な手順は以下のとおりである．

1) 一定の分割深度で分割されたサブボクセルをキューに格納する．
2) 各クライアント PC は順次キューを参照してサブボクセルを読み込み，符号付距離を計算する．
3) 処理が終わったクライアント PC は，引き続きキューを参照して次のサブボクセルを処理する．
4) キューが空になったら，各サブボクセルを木構造として繋ぎ合わせて，最終的な符号付距離場を得る．

4.3 適応的空間分割法

4.2 の手法では，サブボクセル単位に均等に分割を行い，PC クラスタに割り当てることで並列化を実現している．しかしながら，この手法では，サブボクセルごとに含まれる頂点数にばらつきが出てしまう可能性がある．異種のレーザレンジセンサによって得られた複数の解像度が混在する距離画像群においては，いくつかのサブボクセルに大量の頂点が含まれてしまうことがある．これは，非常に解像度の高いレーザレンジセンサを用いて計測を行う場所というのは，対象物全体に対して限られているため，空間的に偏ってしまう可能性があるからである．このような場合に均等な分割を行うと，1 つの PC に 1 つのサブボクセルの計算が残り，他の PC はその計算が終わるのを待つという状態になってしまうことが起こりうる．

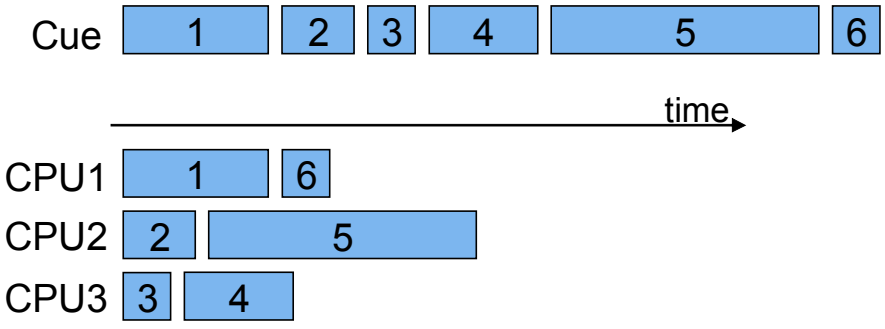

図 4.3 サブボクセルの大きさのばらつきによる計算時間の増大

例えば図 4.3 のようにキューに 6 つのサブボクセルが格納されていて，これを 3 つの PC で並列計算を行うことを考える．ボックスの横幅はそのサブボクセルの計算にかかる時間を示している．ここでは，それぞれのサブボクセルに 1~6 の番号を割り振っている．図を見ると，5 番のサブボクセルが他に比べて大きすぎるために，他の PC クラスタが空き状態になっているにも関らず，CPU2 が長い計算を行っている．このようなことを防ぐために，割り当てるサブボクセルの単位を小さく，すなわち八分木を細かく分割して割り当てることも考えられるが，これではネットワークコストをはじめとするオーバーヘッドが大きくなってしまう．そこで，本研究では八分木の効率的な分割を行うための手法を提案する．

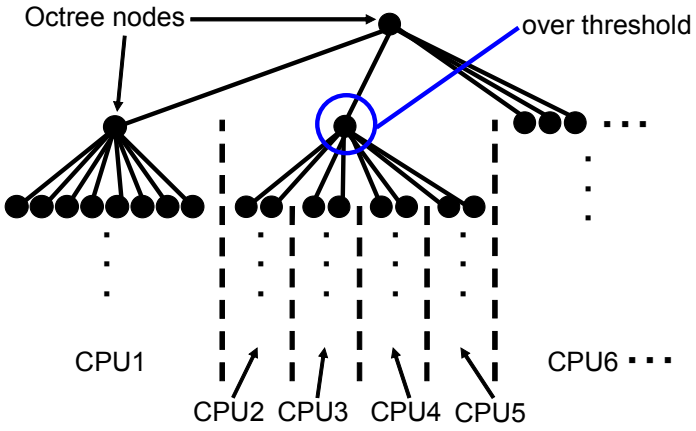

図 4.4 適応的空間分割法によるサブボクセルの割り当て

適応的空間分割法を使用した符号付距離場の計算の手順は以下のとおりである．

1) 一定の分割深度で分割されたサブボクセルについて，そのサブボクセルが閾値内ならそのままキューに格納する．
2) サブボクセルが閾値外なら閾値内になるように分割を行い，新たに生成されたサブボクセルを全てキューに配置する．
3) 各クライアント PC は順次キューを参照してサブボクセルを読み込み，符号付距離を計算する．
4) 処理が終わったクライアント PC は，引き続きキューを参照して次のサブボクセルを処理する．
5) キューが空になったら，各サブボクセルを木構造として繋ぎ合わせて，最終的な符号付距離場を得る．

この際，2)で分割を行ったサブボクセルを記憶しておくことで，5)において適切に繋ぎ合わせることができるようになる．ここで問題となるのは，サブボクセル分割の基準となる閾値の決め方である．時間がかかりそうなサブボクセルに対して分割を行うが，不必要に分割を行いすぎると，オーバーヘッドが増大してしまう．本研究では 2 通りの方法を考えた．

平均と標準偏差による手法

まず 1 つ目は，サブボクセルに含まれる頂点数の平均と標準偏差を用いる方法である．サブボクセルに含まれる頂点数にばらつきが少ないほど，計算時間もばらつきが少なくなるはずである．言い換えれば，PC クラスタに割り当てる各々のサブボクセルに含まれる頂点数が全て等しければ，無駄なく計算が行われるはずである．そこで，サブボクセルに含まれる頂点数の平均と標準偏差を算出し，平均から標準偏差の何倍離れているかを基準に分割を行なった．これにより，データセットによって閾値を変えることなく，サブボクセル 1 つあたりの頂点数を揃えることができると考えた．

符号付距離の計算コストモデル化による手法

2 つ目は計算コストのモデル化による方法である．(3.4)式に，Kd-tree を用いた際の最近傍点探索の計算コストをモデル化した．これを，実際に即した式に展開すると以下のようになる．

$$T_v = M\sum_{k=1}^{M}\log N_k \tag{4.2}$$

ここで，T_v はボクセル v における最近傍点探索にかかる時間，M は探索対象となる距離画像の枚数，N_k は距離画像 k に含まれる頂点の数である．したがって，最近傍点探索にかかる時間の総和 T_{total} は以下のようになる．

$$T_{\text{total}} = \sum_{v=1}^{N_v} T_v \tag{4.3}$$

ここで，N_v は最近傍点探索を行うボクセルの数である．これを用いることで，PC1 台あたりの計算時間は以下のように示すことができる．

$$T_{\text{PC}} = \frac{T_{\text{total}}}{N_{\text{PC}}} \tag{4.4}$$

ここで，N_{PC} は並列計算に用いる PC の台数である．

図 4.3 に示した例のように，重要となるのは PC1 台あたりにかかる時間である．それが等しくなるようにサブボクセルを分割することができれば，無駄が生じない．そこで，それぞれのボクセルに関して T_v が T_{PC} を越える際に，その大きさに応じて，T_{PC} より小さくなるようにサブボクセルの分割を行った．

4.4 実験

計算時間の評価を行うため実験を行った．本研究で提案した適応的空間分割法を用いて実験を行い，4.2 の八分木を用いた並列計算との比較を行った．

なお，実験は 3.3.1 に示した環境で行った．

4.4.1 実験データ

3.5.1 と同じ実験データに対して実験を行った．4.2 で述べた並列化手法を用いて分割深度を大きくしていく手法，適応的空間分割法を 4.3 で述べた 2 通りの分割基準で分割する手法で比較を行った．

4.4.2 実験結果と評価

	Data set 1	Data set 2	Data set 3	Data set 4
Previous method	42	60	10	2
Using average and deviation	11	70	5	2
Using model of time	13	59	10	2

表 4.1 従来手法と適応的空間分割法の計算時間 (min)

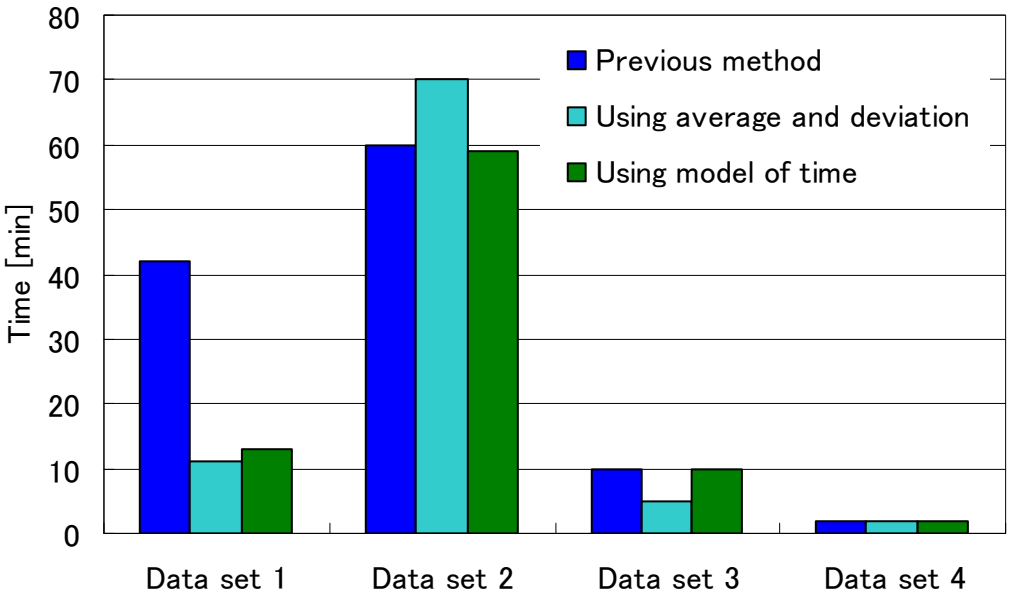

図 4.5 従来手法と適応的空間分割法の計算時間

3 通りの手法で計算を行った結果，表 4.1，図 4.5 のようになった．1 行目が従来の並列化手法を用いた結果，2 行目が平均と標準偏差を用いて適応的空間分割法を行った結果，3 行目が符号付距離の計算時間モデルを用いて適応的空間分割法を行った結果である．従来の手法と比較して，おおむね適応的空間分割法を用いた場合の方が計算時間が短縮されていることが分かる．表 3.7 と比較すると，Data set 4 においては短縮が全くなされていないことが分かる．これは，入力データが小さく，分割を行う必要のあるサブボクセルが存在しなかったためである．

適応的空間分割法の 2 手法を比較すると，標準偏差を用いた手法では，Data set 1 や Data set 3 のように大きく短縮できるものや，Data set 2 のように短縮が上手くいかない場合もある．一方，計算時間モデルを用いた手法は，大幅な短縮は無いものの，どのようなデータに対しても時間を短縮できていることが分かる．

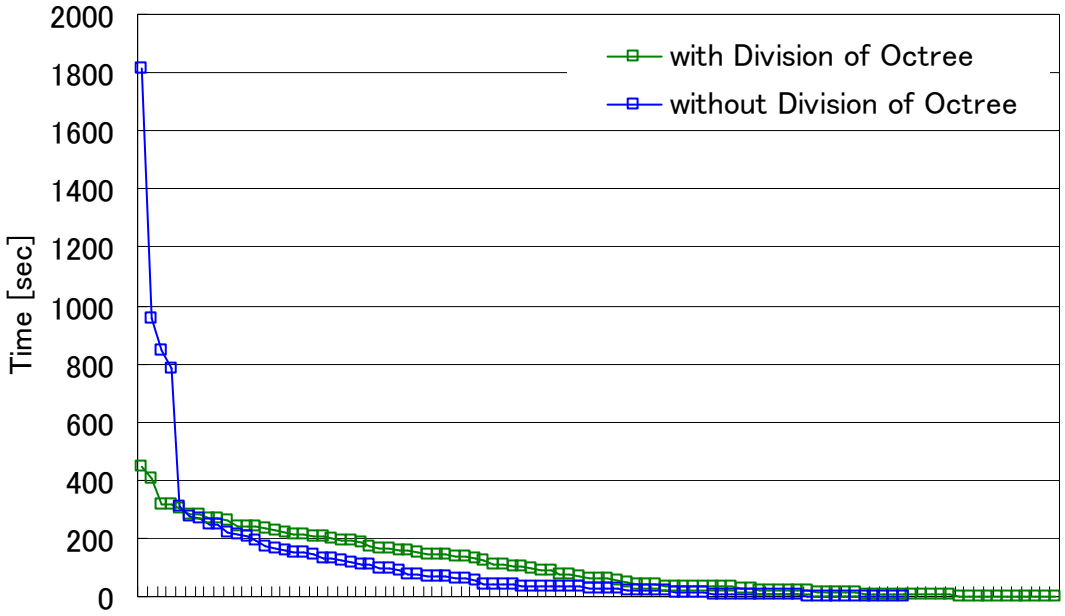

図 4.6 Data set 1 におけるサブボクセルごとの計算時間の分布の違い

図 4.6 は Data set1 におけるサブボクセルごとの計算時間でソートした，適応的空間分割法を用いた場合と用いなかった場合のサブボクセルの計算時間のグラフである．適応的空間分割法を用いた場合，計算するサブボクセルの数が増えてはいるものの，適応的空間分割法を用いなかった場合と比較すると突出して計算時間のかかるサブボクセルが無くなっていることが分かる．すなわち，図 3.20 において完全には解決できていなかったサブボクセルによる計算時間の差異を無くすことができたと言える．これにより図 4.3 のようなサブボクセルの大きさのばらつきによる計算時間の増大を防ぐことができ，計算時間を短縮することができるようになった．

σ ~2 σ	2 σ ~3 σ	3 σ ~	Time [min]
0	8	8	28
0	4	8	11
0	2	8	15
0	4	4	13
2	4	4	14

表 4.2 Data set 1 における平均と標準偏差を用いた分割数と計算時間の関係

表 4.2 は Data set 1 において，平均と標準偏差を用いて適応的空間分割法を行った場合の計算時間の結果である．この手法では，サブボクセルに含まれる頂点数の平均と標準偏差 σ を求め，それぞれのサブボクセルにおいて平均から σ の何倍離れているかで，サブボクセルの再分割の細かさを決定した．その分割数を様々に変えて実験を行った結果が表 4.2 である．表を見ると，分割数を大きく設定すると計算時間が長くなってしまっていることが分かる．つまり，分割数を上手く定めないと，計算時間の短縮が見込めないと言える．平均と標準偏差を用いているので，サブボクセルの大きさの分布が同じようなデータに対しては同様な分割数に設定できると考えられるが，そうでない場合に計算時間が短縮される保障はない．

それに対して計算時間モデルを用いた手法では, PC1 台あたりの計算時間を基準に比較を行うので，データの分布に対する依存性は少ない．すなわち，図 4.3 のように PC1 台の計算時間をはみ出してしまうようなサブボクセルのみを分割すればいい．表 4.1 において，どのデータに対しても計算時間が短くなったのは，以上の理由によると考えられる．実際に統合を行う際には，サブボクセルの大きさの分布がどのようになっているか分からない以上，計算時間モデルを利用する方が好ましいと言える．

	Data set 1	Data set 2	Data set 3	Data set 4
Grid Kd-tree	72	82	9	2.28
Resolution based Grid Kd-tree + Division of Octree	13	59	10	2

表 4.3 従来手法と提案手法の比較 (min)

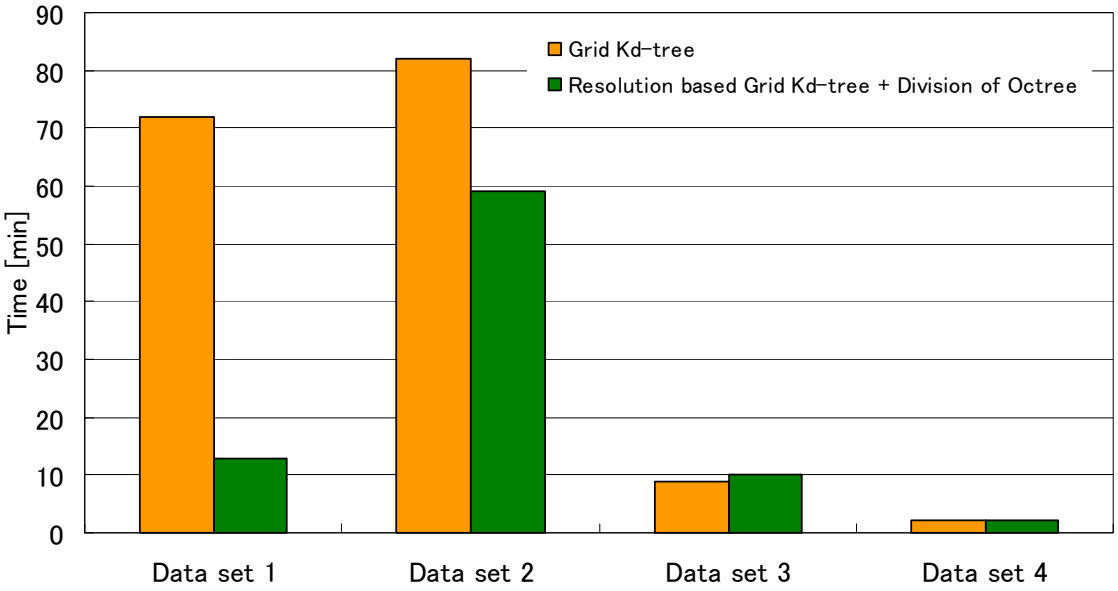

図 4.7 従来手法と提案手法の比較

表 4.3，図 4.7 は従来の Grid Kd-tree と八分木の並列計算手法を用いた場合と，本研究で提案あした Resolution based Grid Kd-tree と適応的空間分割法を用いた場合の比較結果である．Data set 1 や Data set 2 にように，異種のレーザレンジセンサによって得られた距離画像が混在する場合に計算時間が大幅に短縮できていることが分かる．Data set 3 や Data set 2 のような同種のレーザレンジセンサによって得られた距離画像を対象とする場合は，従来どおりの手法を用いれば良いことが分かる．

5 結論

5.1 結論

文化遺産の 3 次元モデルを作成するためには，様々な種類のレーザレンジセンサを用途に応じて使い分けて計測を行う必要がある．そうして得られた精度や解像度の異なる距離画像を，従来の同種のレーザレンジセンサによって得られた距離画像を対象とした手法を用いて統合すると，2 つの問題が生じる．それは，精度の問題と計算速度の問題である．そこで本研究では，同種のレーザレンジセンサを対象としていた既存の手法を拡張し，異種のレーザレンジセンサによって得られた大規模な距離画像の統合手法を提案した．

精度に関しては，合致表面法を基盤にした信頼度付き合致表面法を提案した．従来の合致表面法では，全ての距離画像を均等に扱っていた．ゆえに，精度の異なるレーザレンジセンサによって得られた距離画像が混在していると，精度の低い距離画像に影響され，不均整なモデルが生成されていた．そこで，計測に用いたレーザレンジセンサに応じた信頼度を導入し，合致表面法の計算を拡張することで，均整の取れた 3 次元モデルを生成することができるようになった．

計算速度に関しては，異種のレーザレンジセンサによって得られた距離画像に対して，並列計算を行うための 2 つの提案を行った．

まず，最近傍点探索の計算速度を向上させるため，Grid Kd-tree を基盤にした Resolution based Grid Kd-tree を提案した．Grid Kd-tree では全て距離画像を均等に分割してデータ構造を構築していた．そのため，並列計算を行う際に，解像度の異なる距離画像が混在する場合に，PC クラスタに割り当てるサブボクセルの大きさに偏りが生じ，最近傍点の探索時間が増大していた．そこで，距離画像の解像度に応じて分割深度を変えることができるようにし，また，距離画像 1 枚 1 枚が空間全体に対してほんの一部に存在していることを利用し，分割を 2 段階に分けた．こうすることで，距離画像の解像度に応じて分割深度を設定し，並列計算において PC クラスタへ効率的な割り当てができるようになり，計算時間を短縮することができた．

次に，符号付距離場の計算速度を向上させるため，既存の八分木の並列計算手法を基盤に適応的空間分割法を提案した．八分木の並列計算手法では，空間を均一に分割して PC クラスタへの割り当てを行っていた．そのため，空間に含まれるデータの大きさに偏りが生じ，効率的な計算が行われていなかった．そこで，PC クラスタへの割り当てが不均等にならないように，含まれるデータの大きさを考慮した適応的な分割を行った．その方法として 2 つ提案した．1 つ目の方法はサブボクセルに含まれるデータの大きさの平均と標準偏差

を取り，それぞれのサブボクセルと比較し，分割を決定した．2 つ目の方法は，符号付距離の計算コストをモデル化し, PC1 台あたりの推定計算時間とそれぞれのサブボクセルの推定計算時間を比較することで，分割を決定した．実験の結果，後者の計算モデルの方法が入力データに対する依存性が低く，効率的な分割が行われることが分かった．これにより，並列計算において待ち状態になっている PC を減らすことができ，計算時間を短縮することができるようになった．

5.2 今後の展望

信頼度付き合致表面法では，従来の合致表面法が合致表面に属するか否かの閾値を定数としていたのに対し，可変にすることで信頼度の高いレーザレンジセンサによって得られた距離画像を優先して合致表面とみなすことができるようになった．しかし現状では，この閾値式は経験的な値に基づいて設定している．今後はこの評価を行い，レーザレンジセンサの精度に基づき，最適な閾値式を設定することが可能なようにしたいと考えている．

Resolution based Grid Kd-tree においては，解像度に応じて分割深度を変化させることで，PC クラスタへの効率的な割り当てを実現し，計算時間を短縮した．しかし，解像度に応じた分割深度として，それぞれのサブボクセルに含まれる頂点数がおおよそ同じになるように手動で設定していた．そこで，解像度を考慮した上で，最適な分割深度を自動的に決定する手法を考える必要がある．

適応的空間分割法において，符号付距離の計算モデルを利用することで，空間分割におけるサブボクセルの大きさの偏りを軽減することができ，計算時間を短縮することができた．現在，PC クラスタには空間的な位置の並び順で割り当てている．そこで，計算モデルを利用し，計算時間が長く推定されたサブボクセルを優先的に割り当てることで，さらに計算時間を短縮することが必要であると考えられる．

6 謝辞

本研究を進めるにあたり，ご指導をいただいた池内克史教授に深い感謝の意を表したいと思います．素晴らしい研究環境を整えていただき，的確なアドバイスとともに終始ご支援いただいたことに心からお礼を申し上げます．

研究の遂行に関して，始終研究の面倒を見て下さった大石岳史特任講師には非常にお世話になりました．研究の基本的な指導をはじめ，方向性やプログラムの相談，細かなアドバイスなど多大なご助力をいただきました．多くのご迷惑をおかけしたことをお詫びするとともに，この場を借りて深くお礼を申し上げます．

また，些細な相談にも親身になって聞いて下さった岡本泰英さんに大変感謝しております．プログラムの相談を多くさせていただきましたが，その技術や知識の多さに尊敬の念を抱いています．

同じ修士 2 年の松久亮太さん，工藤雷太さんを始め，池内研究室全員の方には大変お世話になりました．

また本研究を行うにあたって利用させていただいた東京大学情報基盤センターに感謝いたします．多くの相談メールにて助けていただき，研究が捗りました．

最後に学生生活を支えてくれた多くの友達，そして家族に感謝の意を表します．

7 参考文献

[1] K. Ikeuchi, and T. Sato, ”Modeling from Reality” Kluwer Academic Press, 2001.

[2] K. Ikeuchi and D. Miyazaki, "Digitally Archiving Cultural Objects", Springer, Boston, 2007.

[3] M. Levoy, ”The Digital Michelangelo Project” In Proc. SIGGRAPH 2000, pages 131-144, 2000.

[4] R. Sagawa, K. Nishino, and 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 2001, Vol. 2, pages 172-179, 2001.

[5] 天目隆平, 神原誠之, 横矢直和, “「平城宮跡ナビ」マルチメディアコンテンツを利用したモバイル型観光案内システム”, 第 1 回デジタルコンテンツシンポジウム, no.S3-6, May. 2005.

[6] G. Papagiannakis, S. Schertenleib, B. O'Kennedy, M.Arevalo-Poizat, N. Magnenat-Thalmann, A. Stoddart and D. Thalmann, “Mixing virtual and real scenes in the site of ancient Pompeii”, Computer Animation and Virtual Worlds, vol.16, no.1, pp.11-24, Feb. 2005.

[7] T. Kakuta, T. Oishi, K. Ikeuchi, "Development and Evaluation of Asuka-Kyo MR Contents with Fast Shading and Shadowing",Proc. Int. Society on Virtual Systems and MultiMedia (VSMM 2008), Oct. 2008, pp.254-260.

[8] A. Banno, T. Masuda, T. Oishi, K. Ikeuchi, "Flying Laser Range Sensor for Large-Scale Site-Modeling and Its Applications in Bayon Digital Archival Project," International Journal of Computer Vision, Vol.78, No.2-3, pp.207-222, July, 2008.

[9] K. Matsui, S. Ono, K. Ikeuchi, ”The Climbing Sensor:3-D Modeling of a Narrow and Vertically Stalky Space by Using Spatio-Temporal Range Image,” 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, August 2005.

[10] P.J Besl and N.D. McKey, “A method for registration of 3-d shapes,” IEEE Trans. Pattern Anal. & Mach. Intell., vol.14, no.2, pp.239-256, Feb. 1992.

[11] Y. Chen and G. Medioni ”Object modeling by registration of multiple range images.” Image and Vision Computing, Vol.10, No. 3, pages 145-155, 1992.

[12] S. Rusinkiewicz and M. Levoy, ”Efficient Variants of the ICP Algorithm” In Third International Conference on 3D Digital Imaging and Modeling, 2001.

[13] G. Blais and M.D. Levine, “Registering multiview range data to create 3d computer objects,” IEEE Trans. Pattern Anal. Mach. Intell., vol.17, no.8, pp.820-824, 1995.

[14] K. Pulli, “Multiview registration for large data sets,” Proc. Second International Conference on 3D Digital Imaging and Modeling, pp.160-168, 1999.

[15] E. Bittar, S. Lavallee and R. Szeliski, “A method for registering overlapping range images of arbitrary shaped surfaces for 3-D object reconstruction,” Proceedings of SPIE, volume 2059, Sensor Fusion VI, pages 384-395, 1993.

[16] C. Dorai, G. Wang and A. Jain, ”Registration and integration of multiple object views for 3d model construction,” IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 20, number 1, pages 83-89, 1998.

[17] 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 and 2) pages 71-90, 1997.

[18] Z. Zhang, “Iterative point matching for registration of free-form curved surfaces,” International Journal of Computer Vision, volume 13, number 2, pages 119-152, 1994.

[19] 大石岳史，佐川立昌，中澤篤志，倉爪亮，池内克史, “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,” 情報処理学会論文誌：コンピュータビジョンとイメージメディア, Vol. 46 No. 9, pages, 2369-2378, 2005.

[20] G. Turk and M. Levoy, “Zippered polygon meshes from range images,” Proc. SIGGRAPH’94, pp.311-318, July 1994.

[21] M. Soucy, D. Laurendeau,“A General Surface Approach to the Integration of a Set of Range Views”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.17, no.4, pages,344-358, 1995.

[22] W. Lorensen and H. Cline, ”Marching cubes: a high resolution 3d surface construction algorithm.” In Proc. SIGGRAPH’87. ACM, pages, 163-170 1987.

[23] Leif P. Kobbelt, M. Botsch, U. Schewanecke and H.Seidel, ”Fearure sensitive surface extraction

from volume data.” In Proc SIGGRAPH ’01, ACM, pages, 47-56, 2001.

[24] T. Ju, F. Losasso, S. Schaefer, J. Warren, ”Dual contouring of hermite data.” In Proc SIGGRAPH ’02, pages, 339-346, 2002.

[25] H. Hoppe, T. DeRose, T. Duchamp, J.A. McDonald, and W. Stuetzle, “Surface reconstruction from unorganized points,” Proc. SIGGRAPH’92, pp.71-78, ACM, 1992.

[26] B. Curless and M. Levoy, ”A Volumetric method for building complex models from range images,” in Proc. SIGGRAPH ’96. ACM, pages, 303-312, 1996.

[27] 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.295-307, 1995.

[28] C. Rocchini, P. Cignoni, F. Ganovelli, C. Montani, P. Pingi, R. Scopigno, “The Marching Intersections algorithm for merging range images,” The Visual Computer, Vol.20, No.2-3, pp.149-164, May 2004.

[29] M. Wheeler and K. Ikeuchi, “Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition,” IEEE Trans. Patt. Anal. Machine Intell., 17(3):252–265, March 1995.

[30] Haim J. Wolfson and Isidore Rigoutsos, ”Geometric hashing: An overview,” IEEE Computational Science and Engineering, 4(4):10-21, 1997.

[31] A. Califano and R. Mohan, “Multidimensional indexing for recognizing visual shapes,” IEEE Trans. Pattern Analysis and Machine Intelligence, 16:373-392, 1994.

[32] J.L. Bentley, “Multidimensional binary search trees used for associative searching,” Journal of Communications of the ACM, vol.18, no.9, pp.509-517, 1975.

[33] J.H. Friedman, J. Bentley, and R. Finkel, “An algorithm for finding best matches in logarithmic expected time,” ACM Transactions on Mathematical Software, 3(3):209-226, 1977.

[34] J. T. Robinson, “The k-d-b tree: A search structure for large multidimensional dynamic indexes,” In Proc. ACM SIGMOD Int. Conf. on Management of Data, pages 10-18, 1984.

[35] A. Guttman, ” R-trees: A dynamic index structure for spatial searching,” In Proc. ACM SIGMOD Int. Conf. on Management of Data, pages 47-54, 1984.

[36] K. Nakao, Y. Okamoto, T. Oishi, K. Ikeuchi, "Efficient Parallel Integration of a Large Number of Range Images on PC Cluster," MIRU2008, July 2008.

[37] M. D. Wheeler, ”Automatic modeling and localization for object recognition,” PhD. dissertation, School of Computer Science, Carnegie Mellon University, 1996.

[38] R. Sagawa, K. Nishino, M.D. Wheeler, K. Ikeuchi, "Parallel Processing of Range Data Merging", Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, vol.1, pp.577-583, 2001.