

# 多視点カメラ画像からの 対象物体モデリングと 任意視点画像生成

2005 年度

磯 大輔

# 論文要旨

本論文では、多視点カメラ画像を用いて対象物体をモデリングし、その任意視点画像を生成するための研究について述べる。

近年、コンピュータグラフィックスを取り入れた映像表現方法をあらゆるところで目にするようになった。しかし高品質なコンピュータグラフィックスを取り入れたゲームや映画では、その映像生成のための製作コストが莫大なものとなっている。そのような背景のもと、コンピュータビジョンの技術を用いて自動で映像を加工するための研究が活発に行われている。本論文では、このような自動映像加工を目的として近年盛んに研究が行われている、多視点カメラ画像からの対象物体形状のモデリングとそのモデルを用いた任意視点画像生成のために重要となる幾つかの要素技術に焦点を当て、従来の問題点を解決するための研究成果について述べるものである。

まず、任意背景下において影領域の誤抽出の問題を解決するために視差画像を用いた背景差分手法を提案する。本背景差分手法は視差画像を用いることで足元に生じる影領域の誤抽出を改善することができる。

次に、物体形状復元アルゴリズムとしてデータ構造として Octree を用いた視体積交差法を提案する。本手法は Octree の親子関係を利用することで物体形状復元にかかる計算量を大幅に削減することができる。これにより物体形状復元に必要な計算機環境を軽減することが可能となる。

そして、多視点カメラ画像からの対象物体モデリングにおいて最も煩雑な問題の一つである強キャリブレーションを行うことなく、仮想的な 3 次元空間とカメラ画像との対応関係を取ることができる擬似正射影グリッド空間を提案する。従来提案されていた射影グリッド空間でも、仮想空間とカメラ画像との対応関係をカメラ画像間の関係だけで求めることが可能であったが、定義される仮想空間がカメラ幾何の特性により、それをユークリッド空間において表現しようとするとき歪みが生じてしまっていた。しかし、本手法で提案する空間の考え方を導入することにより、カメラ画像間の対応関係のみで、ユークリッド空間のような仮想 3 次元空間を定義することが可能となる。また、定義された空間はユークリッド空間に限りなく近いため、既存のグラフィックス機能をそのまま利用できるという利点もある。本論文ではこの利点を利用して、ボクセルデータで表現される 3 次元モデルから高品質な任意視点画像を生成可能な手法として知られている Microfacet Billboarding 手法を擬似正射影グリッド空間で適用する実験結果も示す。

最後に、3 つの提案手法を組み合わせた物体形状モデリングと任意視点画像生成手法について説明する。3 つの提案手法により、任意背景下においてカメラキャリブレーションをすることなく高速に物体形状をモデリングし、任意視点画像が既存グラフィックスチップ機能を用いて高速に描画することが可能となる。

# 目次

第 1 章 序論	1
1.1 産業における映像・視覚表現の現状	2
1.2 関連研究	5
1.2.1 カメラキャリブレーションに関連する研究	5
1.2.2 シーン全体の任意視点画像生成手法	7
1.2.3 対象物体の任意視点画像生成手法	8
1.3 対象物体に限定した任意視点画像生成手法について	12
1.3.1 対象物体抽出処理について	13
1.3.2 3次元形状情報取得について	15
1.3.3 任意視点映像合成について	17
1.4 本研究の目的	19
1.5 本論文の構成	21
第 2 章 理論	22
2.1 カメラパラメータとカメラの校正	23
2.2 F-Matrix	26
2.2.1 エピポーラ幾何における F-Matrix	27
2.2.2 点对応による F-Matrix の推定	30
2.3 射影グリッド空間	32
2.4 視体積交差法	35
2.5 3次元情報表現形式	36

第 3 章	カラー画像と視差画像による背景差分手法	38
3.1	はじめに	39
3.1.1	影の誤抽出	39
3.2	提案する背景差分手法	41
3.3	評価実験	45
第 4 章	高速な物体形状復元手法	49
4.1	はじめに	50
4.2	提案する物体形状復元手法	51
4.2.1	物体形状復元手法の概要	51
4.2.2	一時状態の決定	53
4.2.3	立方体状態の決定	55
4.2.4	親の一時状態参照による計算量削減	55
4.2.5	内部 Voxel の削除	57
4.3	評価実験	58
第 5 章	画像情報からの座標系定義	60
5.1	はじめに	61
5.2	擬似正射影グリッド空間	62
5.3	着色方法	66
5.3.1	Voxel への着色	66
5.3.2	Microfacet Billboarding 法の適用	67
5.4	評価実験	70
5.4.1	実験 1 : 復元された物体形状の視覚的評価	70
5.4.2	実験 2 : 座標系の歪みの比較	73
5.4.3	2 つの実験結果より	76
第 6 章	提案手法による任意視点画像生成	78
6.1	はじめに	79
6.2	提案する任意視点画像生成手法について	80
6.3	任意視点画像生成実験	82

6.4	共有仮想空間通信システム	98
6.4.1	関連研究	99
6.4.2	本システムの目的	100
6.4.3	システムについて	101
6.4.4	提案システムに対する検討	106
第7章 結論		108
7.1	まとめ	109
7.2	本手法によって解決した問題点	111
7.3	結論	112
謝辞		114
参考文献		115

# 目次

1.1	射影グリッド空間 (PGS)	7
1.2	一般的な任意視点画像生成の流れ	12
1.3	RGB 表色系と HSV 表色系	13
2.1	エピポーラ幾何の概念図	27
2.2	Euclid 空間と Projective Grid Space	32
2.3	射影グリッド空間の定義	33
2.4	射影グリッド空間における視点座標	34
2.5	視体積交差法	35
2.6	様々な表現形式	37
3.1	理想環境と実環境における画素の分布	39
3.2	カラー画像と視差画像	41
3.3	背景差分手法の流れ	42
3.4	背景差分手法擬似コード	42
3.5	背景差分の比較と結果	44
3.6	評価実験対象画像	47
3.7	影領域指定画像と実験結果	48
4.1	物体形状復元手法の流れ	51
4.2	物体形状復元手法擬似コード	52
4.3	画像内立方体領域決定	53
4.4	一時状態決定の例	54
4.5	スタックの参照	56

4.6	内部 Voxel の削除 . . . . .	57
4.7	評価実験に用いたカメラ画像の一部 . . . . .	58
4.8	形状復元結果の一部 . . . . .	58
5.1	OPGS 定義時のカメラ配置例 . . . . .	62
5.2	OPGS とカメラ画像との対応関係 . . . . .	64
5.3	Voxel への着色 . . . . .	66
5.4	微小面と視点・視線方向の関係 . . . . .	67
5.5	微小面へのテクスチャマッピング . . . . .	68
5.6	Voxel モデルから微小面の集合への変換 . . . . .	69
5.7	撮影環境 . . . . .	70
5.8	モデル形状比較に用いたカメラ画像 . . . . .	71
5.9	擬似正射影グリッド空間内での復元モデル形状 . . . . .	71
5.10	射影グリッド空間内での復元モデル形状 . . . . .	71
5.11	ユークリッド空間内での復元モデル形状 . . . . .	71
5.12	PGS の歪み具合の評価 ( $p$ 軸) . . . . .	73
5.13	PGS の歪み具合の評価 ( $q$ 軸) . . . . .	74
5.14	PGS の歪み具合の評価 ( $r$ 軸) . . . . .	74
5.15	OPGS と PGS 内のある平面での誤差分布 . . . . .	75
6.1	任意視点画像生成手法の流れ . . . . .	80
6.2	実験時のカメラ配置 . . . . .	82
6.3	多視点入力画像と得られたシルエット画像 (シーン 1) . . . . .	83
6.4	任意視点画像 (シーン 1 - 1) . . . . .	84
6.5	任意視点画像 (シーン 1 - 2) . . . . .	85
6.6	多視点入力画像と得られたシルエット画像 (シーン 2) . . . . .	86
6.7	任意視点画像 (シーン 2 - 1) . . . . .	87
6.8	任意視点画像 (シーン 2 - 2) . . . . .	88
6.9	多視点入力画像と得られたシルエット画像 (シーン 3) . . . . .	89
6.10	任意視点画像 (シーン 3 - 1) . . . . .	90
6.11	任意視点画像 (シーン 3 - 2) . . . . .	91
6.12	多視点入力画像と得られたシルエット画像 (シーン 4) . . . . .	92
6.13	任意視点画像 (シーン 4 - 1) . . . . .	93

6.14	任意視点画像（シーン4 - 2）	94
6.15	多視点入力画像と得られたシルエット画像（シーン5）	95
6.16	任意視点画像（シーン5 - 1）	96
6.17	任意視点画像（シーン5 - 2）	97
6.18	共有仮想空間通信システム概念図	101
6.19	システム構成	102
6.20	ステレオカメラと同期ユニット	103
6.21	システムの処理分担	104
6.22	システム実行時の様子	106
6.23	本システム実行中のタイムライン	107

# 第 1 章

## 序論

## 1.1 産業における映像・視覚表現の現状

近年、コンピュータ性能の飛躍的な向上により、映像・視覚表現は多種多様になってきている。これは人工的に生成した映像の表現を向上させるための技術であるコンピュータグラフィックス（CG）分野の研究が日々行われ、さらにそれらを実現するためのコンピュータ性能が飛躍した結果、実際に産業分野へ応用することが可能となったためである。まず、CGによってどのように表現が多種・多様化されているかを簡単に説明することにする。

CGの分野での表現方法は大きく分けて写実的表現（photorealistic rendering）と非写実的表現（non-photorealistic rendering）の2つに分けられ、それぞれその表現力を向上させるべく研究が行われている。前者はCGによって人工的に作った映像を限りなく実映像のように見せるための手法の総称であり、後者はCGによって手描きの絵画風の映像やアニメのような見た目の映像を生成する手法の総称である。

写実的表現を用いた映像・視覚表現の産業への応用例として、まずテレビゲームへの応用があげられる。コンピュータ性能の向上とCG技術の発展はゲーム分野において劇的な描画内容の転換を図ることに成功した。従来のコンピュータ性能とCG技術では不可能だった写実的表現のリアルタイム処理が近年では可能となり、実写映像と見分けのつかないゲーム中のシーンや登場人物をユーザがインタラクティブに操ることができるようになった。例えば、高品質なCGモデルをあらかじめ生成しておき、それをゲーム中で用いることで、実在の人物が本当に登場しているようなゲーム映像を作ることが可能となる。具体的な例として、株式会社カプコンが製作した“鬼武者”[68]というゲームシリーズが上げられる。このシリーズでは金城武やジャン・レノといった実在する俳優をCGによるポリゴンモデルによってコンピュータ内にて表現し、それをゲームの主人公としてゲームプレイヤーが操作することが可能である。その結果、プレイヤーは映画の一場面のようなゲーム画面を見ながらゲームを楽しむことができる。また、写実的表現の応用例として実映像とCGによって作られた超現実的なシーンがある。近年の映画ではCGによって作り出された、実際に存在しない世界の風景や、現実には起こりえないシーンの映像と実際に撮影された映像とを組み合わせることで合成された超現実的なシーンを用いることが一般的になってきている。例えば、映画“STAR WARS”[64]シリーズでは実際には存在しない惑星や宇宙のシーンをCGによって作り出し、これを別に撮影した俳優の映像と組み合わせることで、現実には存在しない場所のシーンを実現している。また、映画“The Matrix”[66]において登場した、あたかも時間が止まった世界の中を自由に視点が動く映像は、俳優を取

## 1.1 産業における映像・視覚表現の現状

り囲むように配置した多数のカメラを高速に切り替えながら撮影した実映像を，CG 技術によって加工し，さらにそれを CG によって作られたシーンと合成することによって実現したシーンである。

一方，非写実的表現の応用例として CG によるアニメの作成があげられる。トゥーン・シェーディング (Toon Shading) [24] と呼ばれる非写実的表現の一手法は，影の階調をわざと粗くすることで従来のアニメに用いられているような手書きで加えられた影のような効果を与えることができる。CG によって作られたポリゴンモデルに対してこのトゥーン・シェーディングを適用することで，CG モデルの描画結果へアニメ中の登場人物のような効果を与えることができる。トゥーン・シェーディングが実際に映画に用いられた例として，“APPLESEED” [60] というアニメ映画がある。この映画では，登場人物の CG モデルに対して前述した方法によって手書き風の効果を与えることで，アニメ映画でありながら従来の手描きアニメでは実現が難しかったシーン内の滑らかな視点移動や，登場人物の動きを映像化している。また，もっと一般的に普及しているところでは，ビデオや画像の編集ソフトが備える絵画やステンドグラス風の効果を与える機能もこの非写実的表現の産業分野への応用例である。

表 1.1 に前述した CG 分野における 2 つの表現方法，そしてその特徴と具体例を示す。この様に，現在目にする映像・視覚表現の多くは，表現方法に沿ってコンピュータによって何かしらの加工が施されている。そして今後のコンピュータ技術の進歩や，CG の研究成果により，さらに豊富な表現を持つことが考えられる。ここで，これら進歩によって写実的表現，非写実的表現がそれぞれどのような発展を遂げるかを考えてみる。非写実的表現方法の最終的な目標は，絵画やアニメといった人間の作り出す文化的な側面の影響を強く受けた作品を CG によって表現することである。よって，コンピュータ性能の向上によって導かれる今後の発展はより様々な芸術手法を CG によって実現する，つまりさらに多種多様な表現，そして豊かな表現を実現することと考えられる。一方，写実的表現の目

表 1.1: CG 分野の 2 つの表現方法

表現方法	写実的表現	非写実的表現
特徴	実映像と区別のつかない映像	手描き風や絵画調の映像
具体例	ゲーム，“鬼武者”シリーズ 映画，“STAR WARS”シリーズ 映画，“The Matrix”シリーズ	”APPLESEED” ビデオ・画像編集ソフトの絵画調効果

## 1.1 産業における映像・視覚表現の現状

指すところは実映像と区別のつかない映像をCGによって実現すること、つまりより緻密に、そして実物らしく見える映像を作り出すことである。この特徴から考えると、非写実的手法はCGによって作り出される映像に何かしらの効果を与えるものと言うことができる。そして写実的表現は、実物と区別のつかないようなCGによる映像を、その表現が実現可能な環境において作り出す必要があるといえる。つまり、写実的表現の表現品質のさらなる発展には、より実物らしいCGを作り出すことができる「CGクリエイター」という決定的な要因が必要であるといえる。そして、現在のCGを用いた豊かな写実的表現はクリエイター達が持つ技術や表現力を、時間をかけて注いだ結果の産物であり、それはゲーム分野においても同値である。つまり、このような品質の高い写実的な映像をCGで作成する場合、経済的・時間的なコストは非常に膨大となり、それが映画やゲームの製作費の高騰、製作期間の長期化を招く原因のひとつとなっている。

## 1.2 関連研究

このような背景のもと，実映像から写実的な表現を持つ映像を自動的に加工・合成することができる仕組みに対する需要が高まっている．そして実映像から自動的に映像を合成する一つの解決方法としてコンピュータビジョン（CV）の技術を用いることが導き出され，従来から行われてきたロボット視覚システムやマシンビジョンにおける多視点のカメラ画像から対象物体の形状を復元する研究だけでなく，映像やゲーム分野などへの応用に向けて多視点カメラ画像から新しい映像を自動的に作る研究が活発に行われている．これら自動映像生成手法に関する研究は，大きく2つの方針に分類することができる．1つは撮影した映像全体を利用して新たな映像を合成する手法である．そしてもう1つは対象物体に限定して新たな映像を合成する手法である．本節ではまず，CVの分野で実映像から自動的に映像を合成するために必要とされるカメラキャリブレーションに関する説明について述べる．次に，実映像のシーン全体を利用して自由視点を自動生成する手法について説明し，その後に対象物体に限定してその物体の任意視点画像を自動生成する手法について説明する．

### 1.2.1 カメラキャリブレーションに関連する研究

3次元物体形状復元や任意視点画像生成において重要なカメラキャリブレーション自身や，それに関する研究について述べる．カメラキャリブレーションに関連する様々な研究は3次元物体形状復元手法の研究と同様に活発に行われている．その代表的な研究の1つがTsaiによって提案された手法[41]である．Tsaiは実世界の3次元座標と2次元画像平面間の対応点セットを6つ以上用いて，実世界の座標から画像内の座標へ変換する手法を提案した．現在，この手法はカメラキャリブレーションを行う際に広く用いられている．Tsaiは3次元空間内での物体の位置は回転と平行移動のみによって表現できることを利用して，6つ以上の対応点の組からこれらのパラメータを取得する手法を提案した．これらのパラメータを得ることができれば，3次元空間から2次元のカメラ平面への射影変換行列を求めることが可能となる．すなわち，カメラキャリブレーションが行われた状態となる．

その後，Zhang[50]はこのカメラキャリブレーションを簡易化する手法を提案した．この手法では，1台の固定カメラである平面上のマーカを動かして撮影する．この時，撮影された平面が3次元空間中でのX-Y平面であることと，平面が動くということはカメ

ラが相対的に動いているということ想定することで、カメラキャリブレーションを行う。この手法では撮影対象と撮影したカメラ画像がどちらも2次元であることを利用することで従来よりも容易にカメラキャリブレーションを行うことが可能となる。

また、カメラキャリブレーションを行わずに画像間の対応関係だけで対象物体の3次元形状を復元する手法も行われている。斎藤と金出ら [36] は2台のカメラのカメラ光線によって定義される射影グリッド空間 (Projective Grid Space, PGS) において対象物体の3次元形状を復元し、任意視点画像を生成する手法を提案している。この PGS とは図 1.1 に示すように、2台の基底カメラから発するカメラ光線によって座標系が定義される空間である。これらのカメラ光線は、他のカメラへエピポーラ線として投影されるため、仮想空間内のある1点をエピポーラ線の交点として画像内で知ることができる。よって、空間内の1点と複数台のカメラ画像内でのその対応点の関係をを用いることで、視体積交差法を用いて対象物体の形状を復元することができる。しかし、従来の PGS で用いられる3次元座標系はカメラ間のエピポーラ幾何により定義されているため各軸が互いに直交したものにはならない。そのため、PGS で復元した形状をポリゴンモデル等に変換して一般の3次元モデルビューワーを用いて表示しようとしても、それらのビューワーは各軸が直交したユークリッド座標系を想定しているため、歪んだモデル形状になってしまう。そこで従来の PGS に基づいた自由視点画像生成手法 [46] においては、PGS で復元した形状から入力画像間の対応関係を求め、この対応関係を用いてモーフィングにより自由視点画像を生成することを行っていた。

同じようにカメラキャリブレーションを行わずに画像間の対応関係だけで対象物体形状を復元する手法が Kimura ら [21] によって提案されている。この手法では3台の基底カメラからのカメラ光線によって定義される Projective Voxel Space (PVS) 内において対象物体形状を復元する。この手法ではまず各カメラ画像上に投影されたエピポーラ線を平行化し、その直線上で3つの画像間での対応点の探索を行う。そして視体積交差法によって対象物体形状の復元を行う。PGS と同様に、カメラ光線からなる PVS において復元された形状をユークリッド空間にてそのまま表現すると歪みが生じてしまっている。しかし、PVS で得られた復元情報にはカメラ画像間の対応点やオクルージョンに関する情報が含まれているため、これら情報を用いて任意視点の合成を行う。任意視点の合成には View Morphing を用いている。

### 1.2.2 シーン全体の任意視点画像生成手法

では次に、撮影した映像全体を使用して新たな映像を合成，生成する手法について述べる。

2001年にアメリカのCBSや日本のフジテレビがスポーツ中継で実験的に放送した“EyeVision”システム[62]は多視点カメラの映像から自動的に映像を生成する手法の実用例である。“EyeVision”システムとは、カーネギーメロン大学によって研究、実用化されたシステムで、スタジアムを取囲むように配置された多数のカメラを完全同期させて同一エリアを撮影し、それら各カメラからの映像を高速に切り替えることでそのシーンを時間が止まった世界の中で視点を動きながら見たような効果を作り出すことができるシステムである。しかし、この“EyeVision”システムはあくまでも撮影した映像を切り替えるだけで、撮影したカメラとカメラのちょうど中間の位置での視点の映像を作り出すことはできない。北原ら[22, 53]や、稲本ら[15, 52]はサッカーシーンを多視点カメラで撮影した映像全体を用いて、そのシーンの視点と視点の間に位置する視点での映像を自動生成する手法を提案している。これらの手法では対象物体の3次元的な形状を明示的に復元せず、撮影したカメラと生成する自由視点との間の3次元的な幾何情報を用いて実画像の変換・合成を行うことで自由視点を生成している。これらの手法は原画像の情報を最大限活用し

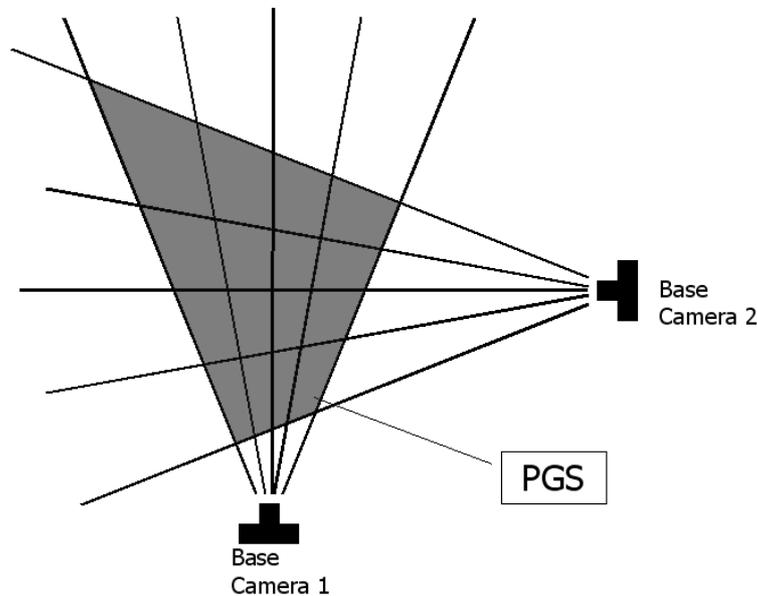


図 1.1: 射影グリッド空間 (PGS)

ているため、生成された自由視点映像は良好な結果となっている。これらの手法では、実際のカメラ同士の間位置での映像を作り出すための手法であるイメージベースト・レンダリング (Image Based Rendering: IBR)[45, 25, 10, 14, 38] を用いている。IBR とは実際に撮影された画像をもとにして新たな画像を作り出す手法全般を総称したものである。IBR は実画像間の対応関係に基づいて中間視点位置の画像を合成するが、この時に視点と視点の位置が離れていると合成される中間画像の精度が著しく悪化するという特徴がある。そのため、撮影に用いるカメラ間の間隔を狭くして多数のカメラを配置するか、狭い範囲のみでの中間視点位置画像を生成するに留まっているのが現状である。

### 1.2.3 対象物体の任意視点画像生成手法

次に、対象物体のみに限定してその任意視点画像を自動生成する手法について説明する。対象物体の任意視点画像生成手法は多視点カメラ画像を利用するものが主流であるが、単一カメラ画像から形状を復元する手法も存在する。そこで参考として単一カメラ画像からの3次元形状復元手法について説明し、次に多視点カメラ画像からの任意視点画像生成手法について説明する。

#### 1.2.3.1 単一カメラ画像からの3次元形状復元手法

単一カメラ画像から3次元形状を復元する手法として、カメラの焦点を利用した Nayar らの Shape from Focus 法 [33] がある。Shape from Focus 法とは単一カメラで対象物体をカメラ方向に沿って動かした画像を何枚か撮影し、それらの画像の焦点の合い具合から奥行き方向を推定する手法である。この手法では焦点が合っているかどうかの判定に sum-modified-Laplacian(SML) を用いる。ある画像において、焦点の合っている平面では画像中の周波数成分が高くなるが、逆に焦点の合っていない画像の場合は低周波成分が多くなる。これを利用してある点において焦点が合っているかどうかの判定を行う。まず、撮影された各画像に対してある大きさの窓を設定し、その窓内での周波数成分を求める。全ての連続画像において対象物体の同じ領域が撮影されている範囲の周波数成分が最も高い画像が最も焦点が合っているため、そのときの物体とカメラとの距離を保持しておく。これを全ての領域に対して行うことでカメラから見た奥行き方向の復元を行うことができる。この手法の問題点はカメラの視線方向のみの形状しか復元できないこと、また物体を移動させて撮影する際に、対象物体の同じ領域が画像内では大きさが変わってしまうことなどが上げられる。

同様に，単一カメラ画像から3次元形状を復元する手法として，単一カメラを動かしながら対象物体を撮影し，特徴点の移動量からその奥行き方向の形状を求める Shape from Motion[5, 1] がある．これは，カメラを動かしながら対象物体を撮影する際に，カメラから遠い位置にある物体の方がカメラから近い位置にある物体よりも撮影された画像内での移動距離が小さいことを利用する手法である．この手法では，連続画像内における特徴点の追跡が難しいという問題点がある．この特徴点追跡時の問題解決のために，エピポーラ幾何を利用した手法 [31] などが提案されている．

### 1.2.3.2 多視点カメラ画像からの任意視点画像自動生成手法

ここでは，多視点カメラ画像から対象物体の任意視点画像を自動生成する手法とそれに関連する手法について述べる．

金出ら [17, 18] は，時間と共に変化するイベントを多数のカメラからの画像を用いてモデリングして任意視点画像を生成する手法を提案している．これは”Virtualized Reality”と呼ばれ，金出らによってこのコンセプトが提案されて以来，多くの研究者達により活発に研究が行われている．

Vedula らはマルチベースラインステレオと視体積交差法を組み合わせた手法 [43] を用いて，50台のカメラによって得られた画像から復元された対象物体形状の表面ポリゴンモデルへ入力画像のテクスチャをマッピングして自由視点画像を生成した．まず，事前にカメラキャリブレーションを行っておき，校正済みの50台のカメラによって得られた画像に対してベースラインステレオを用いて視差画像を生成する．そして得られた視差画像を各カメラのシルエット画像へ再投影することで物体領域のみを切り出す．この処理を何度か繰り返して精度を上げ，最後に得られた表面ポリゴンモデルへテクスチャマッピングを行う．そしてテクスチャマッピング済みのモデルをあらゆる仮想視点へ投影することで，それら対象物体の任意視点画像を生成する．

また Vedula らは後に，視点間だけでなく時間軸方向にもモデル形状の内挿を行い [42]，任意の時刻における自由視点画像の生成に成功している．この手法では復元された Voxel モデルの Voxel 1つ1つの3次元空間中の軌跡を”scene flow”と名付け，その位置を補間することで，カメラで撮影していない時間のモデル生成を実現している．

また，多視点画像から対象物体を復元してリアルタイムで表示するシステムが Cheung ら [9] や，濱崎ら [59] によって提案されている．Cheung らの手法では対象物体形状を視体積交差法によって Voxel モデルとして復元している．さらに復元された Voxel モデルの全 Voxel に対して，表面判定処理を行う．これは物体を表すある Voxel の6近傍に物体

を表す Voxel が存在しなければ、その Voxel は物体表面に位置するという考えに基づいている。この処理により、視点からは見ることのできない内部の Voxel を削除し、復元モデルの実時間提示（毎秒 10 フレーム程度）を実現している。しかし彼らのシステムの目的は実時間での人体動作の解析であるため、復元した Voxel への着色は一切行っていない。また精度のよい復元結果を求めていないため、Voxel の解像度は  $64 \times 64 \times 64$  と比較的荒くなっている。

対象物体の 3 次元形状を明示的に復元することなく、自由視点を生成する手法も同様に研究が行われている。ここではそれらについても触れたいと思う。Matusik ら [30] によって多視点カメラ画像から任意視点画像をリアルタイムに出力するシステムが提案されている。このシステムでは、画像の見え方で対象物体形状を間接的に表現した hull を独自の高速アルゴリズムで求めて、任意視点画像を生成している。彼らのシステムは 4 つのカメラによって撮影された画像から hull を生成し、任意視点画像生成までをおよそ毎秒 8 フレームの速度で実現している。

Zitnick[51] らは多視点カメラ画像と、それらと同じ位置から得られる視差画像とを用いて、任意視点画像をリアルタイムで出力するシステムを提案しているが、彼らの手法も対象物体形状を復元しているわけではなく、3 次元形状を明示的に得ることはできず、さらに視点の移動位置も限られてしまう。

Yamazaki ら [47] は仮想視点と常に平行となる小さな板状の物体へ、その範囲に写るカメラ画像を貼り付けて描画する手法について提案している。この手法はレンジスキャナによって得られたレンジデータと、それとまったく同じ位置におかれたカメラ画像を用いている。まず複数のレンジスキャナによって得られたレンジデータを統合し、対象物体形状を復元する。そして復元モデルに Microfacet と呼ばれる小さな板状の物体を配置し、かつ仮想視点と常に垂直になるような向きにしておく。なお、この板は復元されたモデル表面に等間隔に配置されている。そして板へ各カメラから得られた画像の適切な領域をテクスチャマッピングすることで任意視点を合成する。この手法により、毛のようなポリゴンや Voxel によって表現することが難しい形状を実画像から取得したテクスチャを用いることで描画できるため、現実に近い任意視点画像を出力することができる。

この Microfacet Billboarding 法を用いて実時間での任意視点描画システムが Goldlücke ら [13] によって提案されている。彼らは低解像度 ( $64 \times 64 \times 64$ ) の Voxel 空間において復元した対象物体形状から Microfacet を配置して任意視点の描画を行っている。Microfacet を使用することにより、低解像度にもかかわらず比較的精度のよい任意視点画像を生成することを可能としている。

また Carranza ら [8] はカメラで撮影した人体動作をあらかじめ保持してある人体モデ

## 1.2 関連研究

ルへ割り当て，その人体モデルに多視点カメラから取得した実画像をマッピングすることで，撮影中の人物を任意視点から見ることができるシステムを提案している．テクスチャマッピングを行う際に，人体モデルの各頂点に対してカメラから隠れているかどうかの判定を行う．この判定により，誤ったテクスチャマッピングを回避してより誤差の少ない任意視点画像を生成することが可能となっている．

ここに述べるようにカメラ画像から対象物体の任意視点画像を生成する研究が活発に行われているが，ここに述べた手法は現実世界の3次元空間と，カメラ画像の2次元平面との対応関係を求めるために1.2.1にて述べたカメラキャリブレーションを行っている．

### 1.3 対象物体に限定した任意視点画像生成手法について

前述した，多視点カメラ画像から対象物体に限定した任意視点画像生成手法は大まかには3つの処理部から構成される．図 6.1 はそのフローと各処理部での問題点，改善点を示したものである．まず，多視点入力カメラ画像から対象物体領域のみを抽出する処理を行う．次に，得られた対象物体領域の画像から，対象物体形状に関する3次元的な情報を取得する．そして最後に，得られた3次元的な情報から任意視点画像を描画するための処理を行う．

この3つの処理部にはそれぞれ問題点，改善すべき点があり，これらに関して現在も広く研究が行われている．対象物体抽出処理部において，より正確に対象物体のみを抽出するための問題について今でも研究が行われている．そして3次元形状情報取得処理に関しては，高速に物体形状情報を取得するための計算アルゴリズムやその他の解法，そしてより容易にカメラ画像と空間との位置関係を取得するための問題について議論されている．さらに任意視点映像合成処理部ではより精細な映像を出力するための解決法を導くべく，広く研究が行われている．ここでは，対象物体に限定した任意視点画像生成手法における各処理の細かい問題点についてさらに詳しく述べ，それらを解決するためにどのように取り組まれているかについて説明する．

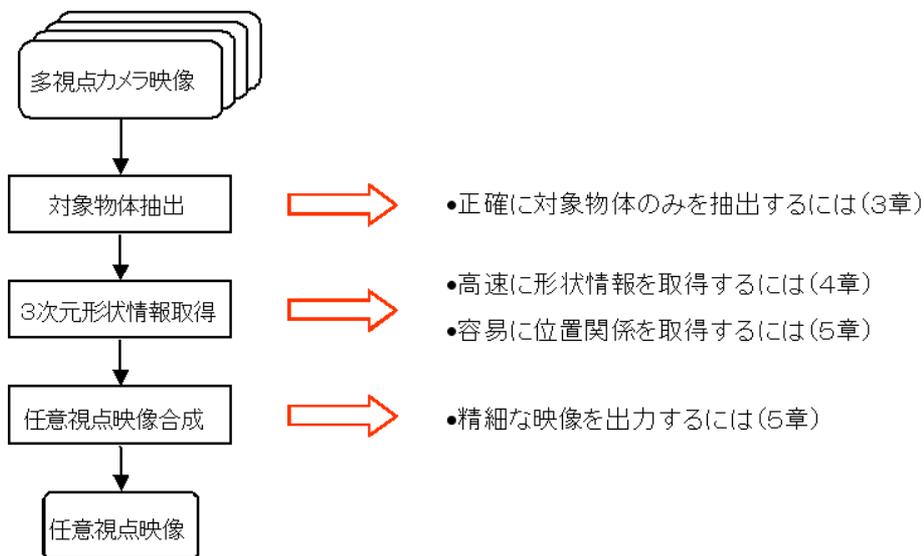


図 1.2: 一般的な任意視点画像生成の流れ

### 1.3.1 対象物体抽出処理について

カメラ画像から対象物体を抽出するための研究は監視カメラシステムや映像加工，また MPEG 符号化への応用を目指して活発に行われている。しかし光源や影の効果が画像に対して強く影響を与えてしまうため，あらゆる環境において完全な精度で対象物体のみを抽出するのは困難であるのが現状である。光源や影の影響，その他特定の環境に対して有効な手法を提案するために，一般的に用いられる RGB 形式だけではなく，YUV 形式や HSV 表色系に色を変換した背景差分手法の試みも行われている。

ここで，HSV 表色系と YUV 表色系について簡単に説明する。HSV 表色系とは色相 H，彩度 S，明度 V の 3 つの成分によって色を表す形式のことを指す。色相はいわゆる人間が視覚的に感じる赤や青といった色合いの度合いを表し，彩度はその色の鮮やかさを示す度合いである。また明度は色の明るさの度合いを示す。図 1.3 に RGB 表色系，HSV 表色系の概念図を示す。HSV 表色系では色合いを表す色相が独立しているため，色の分類が容易になるという特徴がある。また，明度 V を用いることで光源や影の影響による変化を調べやすいという特徴もある。次に，YUV 形式とは，輝度 Y と輝度成分と青色成分の差 U，輝度信号と赤色成分の差 V の 3 つの要素によって色を表す形式である。この形式では画素の輝度が Y 成分によって表されているため，光源や影によって変化した輝度の影響を Y 成分から抽出しやすいという特徴がある。RGB 形式から YUV 形式へは変換行列を用いることで一度の演算で変換することが可能である。

では，どのような背景差分手法が実際に研究され，提案されているか具体的に述べてい

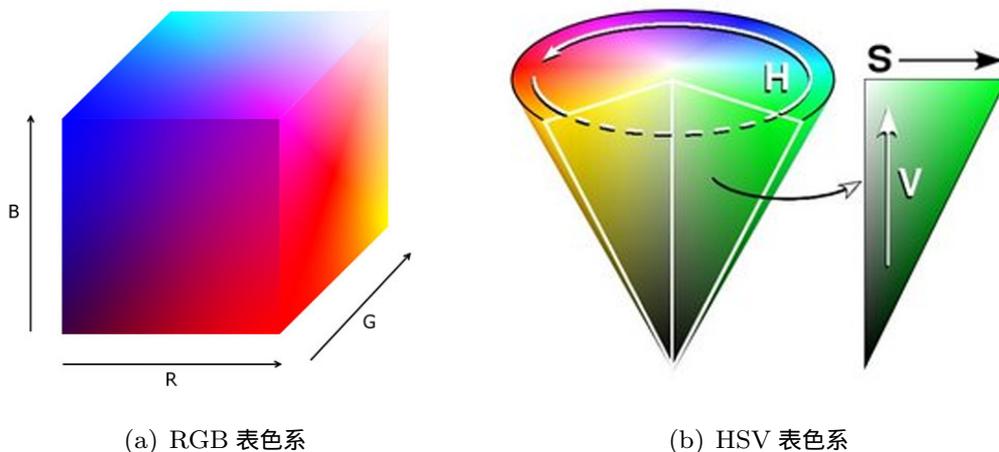


図 1.3: RGB 表色系と HSV 表色系

### 1.3 対象物体に限定した任意視点画像生成手法について

く、Yang ら [48] は動く物体の存在するシーンにおけるその物体領域の抽出、追跡手法についての研究を行い、実時間処理で物体領域を抽出する手法について提案している。この手法では、RGB 表色系をそのまま用いて物体領域抽出を行っている。まず、入力画像と背景画像から R, G, B 各要素の減算を行い、各要素の差の 1 つでも閾値以上である場合は前景領域と判定し、全ての要素の差が閾値より小さい場合は背景領域とする。この演算結果では光源の影響による小さなノイズが含まれているため、この結果画像を縮小し、縮小された画像において輪郭領域のラベリングを行い、連結領域が小さいラベル領域を取り除く。そして結果画像と縮小前の画像の論理積を取ることで最終的に前景領域の抽出を行う。なお、この手法では背景画像の更新を常時行うため、R, G, B 要素の単純な差分のみにもかかわらず良好な結果を得ることができている。しかし、この手法は物体の追跡が主目的であるため、得られた物体領域には背景と誤判定された小さな画素が残ってしまうという問題点がある。

Francois ら [12] は RGB 表色系ではなく HSV 表色系に変換して背景差分を行っている。この手法ではまず入力画像の全ての画素の色を RGB 形式から HSV 形式に変換し、何枚かの背景画像の各画素の平均と標準偏差を求めておく。そして背景画像と入力画像の各画素に対して背景か前景かの判定処理を行う。判定処理は、背景画像の平均画素値と入力画像の画素が標準偏差の 2 倍以内であればその画素は背景とし、それ以外の場合は前景とする、というものである。しかしこの手法では、RGB 形式から HSV 形式に色空間を変換する際に、画像内の暗い領域における色相の精度が落ちてしまう問題があり、そのため暗い画像では影の誤抽出が起こってしまうという問題がある。

Kamkar-Parsi ら [16] は YUV 形式の U と V 成分の分散、Y 成分のエッジの出現・消失確率、Y 成分の勾配からなるベクトルを用いた背景差分手法を提案している。まず、入力画像と背景画像の各画素における U と V 成分の分散を求め、閾値処理を行う。つまり色を表す U, V 成分の分散を閾値処理することで明らかに背景か物体であるかの判定をあらかじめ行っておく。この時に背景か前景かを明確に判定できない画素については不定の値として判断を行わずに次の処理に進む。次のステップでは、全ての画素において Y 成分のその画素での水平方向と鉛直方向の勾配を求め、背景画像、現在の画像でそれぞれエッジが出現・消失した確率を求める。背景でも現在の画像においてもエッジ出現・消失の起こった確率が低い場合は背景と判定し、エッジが出現・消失している確率が高い場合は前景とする。それ以外の中間の場合について不定と判定して次のステップに進む。3 つめのステップでは対象画像とその 4 近傍での Y 成分、つまり 5 つの Y 成分の値からなる 5 次元のベクトルを求め、背景と現在の画像とでそのベクトルのなす角を求める。この角度が閾値より小さい場合は背景とし、そうでない場合は前景とする。最後に各処理ブロックで

### 1.3 対象物体に限定した任意視点画像生成手法について

得られた背景，前景，不定の結果を統合して最終的にその画素が背景であるか前景であるかの判定を行う。この手法ではエッジ出現・消失の確率がガウス分布に依存しているため，この処理での判定が最終的な前景抽出結果に大きな影響を与えるとといった問題点がある。

ここまで述べたように，現在の背景差分処理ではあらゆる条件で完璧な物体領域を抽出できる手法がまだ確立できていないのが実状であり，「ある条件において良好な結果を得る」ことがまず第一の目的となっている。そして，それら手法を足がかりとして最終的に堅牢な手法を提案することを目的として現在でも研究が続けられている。

#### 1.3.2 3次元形状情報取得について

次に，3次元形状情報取得についての問題点とそれを解決している関連研究について述べる。

コンピュータビジョンの分野では古くからカメラ画像から対象物体の形状を求める研究が行われ，それに伴い3次元形状情報を自動で復元するための研究も行われている。しかし3次元形状を求めることは， $O(N^3)$  のオーダーの計算量が必要であり，常に計算量との戦いを強いられてきた。この問題を解決するために，高速に物体形状を復元するための研究が行われている。それら研究のアプローチとして大きく分けて2つの方法が取られている。1つは豊富なコンピュータ，CPU リソースを利用する方法で，もう1つは計算アルゴリズムを改善する方法である。

前者の具体例として，Wu ら [44] の提案した PC クラスタを用いたシステムがある。彼らは多視点カメラ画像から対象物体を撮影して PC クラスタを用いて身体動作の実時間3次元映像化を行っている。彼らは PC クラスタによる並列処理を活かすためのアルゴリズムとして，3次元空間中の各 Voxel を全てカメラへ投影するのではなく，空間を平面に分割して各平面ごとにカメラへ投影する方法を用いている。これにより3次元の計算が2次元の計算に集約され，さらに PC クラスタによる並列処理により高速に物体形状を復元することができる。この実験結果では1つの Voxel のサイズが  $3cm^3$  の環境下において，ほぼ実時間の処理を実現している。

また，後者の具体例として Potmesil[35] や Srivastava ら [39] の用いた Octree データ構造を用いた復元アルゴリズム，Niem[34] や Lyness ら [28] が用いた“pillar”と呼ばれる柱状の領域を1つの単位として行った方法などがある。Potmesil は空間を分割した立方体1つ1つを表す8分木データ構造を用いて計算量の削減を行う手法を提案した。この手法では，1つ1つの Voxel 単位ではなく，あるまとまりの空間単位で形状復元を行うため，計算量の削減を行うことが可能である。また，Srivastava らは透視投影カメラモデ

### 1.3 対象物体に限定した任意視点画像生成手法について

ルの環境で、凸形モデルを対象とした交差判定を画像平面上ではなく3次元空間中で行う手法を提案した。この手法での形状復元処理は4つの処理に分かれている。1つめの処理では、1つのシルエット画像によって得られる角錐形モデルを囲む円錐形モデルを考え、そして処理対象となる立方体を囲む最小の球との交差判定を行う。この大まかで単純な処理によって、明らかに角錐形モデルの外部に立方体が存在するかどうかの判定を行う。2つめの処理では、立方体の8頂点が角錐形モデルの内部にあるかどうかを調べる。全ての頂点が角錐形モデルの内部にあればその立方体は物体とし、頂点が角錐形モデルの内部と外部に存在する場合は分割対象の立方体と考える。全ての頂点が角錐形モデルの外部にある場合は、3つめの処理に移る。3つめの処理では、角錐形モデルの面に対し、立方体の8頂点がどのような位置にあるかを調べる。8つの頂点が外側にくる面が1つでも存在すれば、立方体は角錐形モデルの外部にあると判定し、そうでない場合は角錐形モデルと立方体は交差していると判定する。

Niem や Lyness らの手法では“pillar”と名付けた柱状の領域を1つの単位として対象物体復元を行っている。この復元手法は、対象空間となる3次元仮想空間を鉛直方向に伸びる柱の集合として扱い、その柱の上端と下端の2点をシルエット画像へ投影する。するとシルエット画像上に2点を結ぶ直線が現れる。シルエット画像とその直線の交差している点を探し、その直線のシルエット上と背景領域上を通過している領域を求める。そしてシルエットとの交点を3次元空間へ再投影し、シルエット上を通過した直線領域に該当する部分のみを3次元空間内に残す。これを全ての柱状領域に対して行うことで、3次元空間内に物体形状を復元する。この手法においても、ある領域ごとに物体形状復元をおこなうため、計算量の削減が可能となる。

ここまで述べたこれら2つのアプローチにはそれぞれメリット、デメリットが存在する。前者の豊富なコンピュータ、CPUリソースを使用する方法ではその圧倒的な計算量が劇的な計算量削減が期待できる。しかし、豊富なリソースを使用するためにはその設備のためのコストがかかってしまうという根本的なデメリットがある。また、計算アルゴリズムの改善では基本的にコストをかけずに計算速度向上が見込めるといったメリットがある一方、前者ほどの劇的な計算速度向上は見込めないという側面もある。しかし、高速な計算アルゴリズムによって設備に必要なコストを小さくすることが可能であり、最良の計算方法の探究は常に行われるべきであると考えられる。

### 1.3.3 任意視点映像合成について

多視点カメラ画像から任意視点画像を生成する最後のステップとして3次元物体形状情報を元にその物体の任意視点画像を生成する処理を行う。ここでの処理では物体形状情報から、復元された物体の位置に応じてカメラ画像から取得した色を任意視点画像内の適切な位置へ着色を行う。その際、物体復元形状に生じる誤差や3次元仮想空間とカメラ画像間での座標変換で生じる誤差などにより、実画像と同じような精度の任意視点画像を生成することが困難であるのが現状である。そしてこの問題を解決するために、より精細で実画像に近い出力結果を得るために様々な研究が行われている。

第1.2節にて前述した Yamazaki らの提案した Microfacet Billboarding 法は、この問題を解決するための1つのアプローチである。詳細は前述した通りだが、ある Microfacet 内ではテクスチャの連続性が保たれているため出力される任意視点画像の精度は良好である。また、表面が毛で覆われているような対象物体の場合、復元された物体形状でその毛を表現するのは非常に難しいが、この手法では物体表面の毛の見え方を実画像のテクスチャを用いて表現できるため、たとえ物体形状でその毛を表現することができなくても出力画像では毛の部分まで表現することが可能である。

撮影しながら任意視点画像を生成しない場合では、さらに計算時間をかけたアルゴリズムを用いることで任意視点画像の精度を向上させることが可能な手法が提案されている。

西野ら [54] は「Eigen-Texture 法」名付けた手法を提案している。この手法では様々な明るさで撮影された多視点入力画像列を、セルと呼ばれる小さな画像内領域ごとに固有空間に圧縮して保持し、距離画像によって復元された対象物体へテクスチャマッピングを行う。画像情報を圧縮して保持しているにもかかわらずその劣化の度合いは小さく、表面が毛で覆われているような場合でもセルごとにテクスチャの連続性が保たれているため、その毛の状態を表現することが可能である。また、様々な照明環境で撮影されているため、複合現実感のように仮想的な環境内で照明条件を変化させても、その照明状態に応じた対象物体の任意視点画像を生成することが可能である。だが、この手法はそもそもが撮影しながらの任意視点画像提示を目的としていないため、固有空間に圧縮する際の計算コストが非常に膨大であるという欠点がある。

Eisert ら [11] は、多視点カメラ画像から静止物体を高精度に復元し、その任意視点画像を自動生成する手法を提案している。この手法の特徴はカメラキャリブレーションの誤差によって生じる物体形状の誤差を補正する点である。まず、初期モデルとして多面体を用意しておき2つのカメラ画像からその初期モデルを物体形状へと変形させる。初期モデ

### 1.3 対象物体に限定した任意視点画像生成手法について

ルへの変形はカメラ画像と、2台のカメラ画像に対してステレオ法を行って取得した視差画像とを用いる。初期モデルの各頂点を制御点とし、これをカメラ画像へと投影して制御点がカメラ画像の物体領域の境界部に隣接するように動かす。この際、視差画像を用いて制御点移動のスケーリングを行う。この時点で得られたモデルにはキャリブレーションと視差画像での誤差が含まれた状態である。これら誤差の影響を取り除くため、対象物体の位置と形状の推定を行い、物体形状の補正を行う。位置の推定では2つのカメラ画像間の位置関係を用いる。3次元空間中の1点をその画像への射影変換することで直接的に求めた画像内の1点の画素と、片方の視点の1点を相対的な2つのカメラの相対的な位置関係によって求めたもう片方の1点の画素は等しくならなければならない。しかし、実際にはキャリブレーションの誤差の影響により同じ位置にならないことが多い。そこで、相対的な位置関係の式と前述した2通りで得られた画像内の2点間のズレから、物体位置の推定を行う。次に、制御点を放射線状に動かすことで物体形状の補正を行う。物体形状の補正も、前述した位置推定と同様に2つのカメラ画像間の位置関係を用いる。ある制御点を放射線状に動かすと、必要なパラメータは放射線状にどの程度移動させるかという係数になる。この係数を2つの視点間の相対的な位置関係と、それらから導かれた画像内の2つの投影点のズレから算出する。この位置と物体形状の補正を行うことで、この手法で得られる物体形状は精度が高くなり、その結果物体へ貼り付けられるテクスチャのずれがなくなる。つまり、物体形状表面にほぼ正確にテクスチャが貼り付けられるため、最終的に得られる任意視点画像の精度が高くなる。この手法も静止物体を対象としているため、位置や形状の誤差推定にかかる計算時間は膨大であるという欠点がある。

以上、ここまで述べたように、より精度の高い任意視点映像合成のための研究がなされている。そしてこれら手法に共通しているのはカメラキャリブレーションが必要であるという点である。第1.2.1節で述べたようにカメラキャリブレーションは非常な煩雑な作業を伴い、その煩雑さゆえにスタジオなどの大規模な空間での実験が困難となっている。

## 1.4 本研究の目的

第 1.3 節までに述べたように，任意視点画像生成手法のおおまかな 3 つの処理には様々な問題点があり，それらを解決するための研究が現在でも行われている．そしてそれら研究の最終目的は，現在の映像技術への応用や新しい映像表現技術の生成である．そこで，各処理部におけるこれら問題のうち以下に記述するものを解決するための手法をまず提案する．そして本論文に述べるこれら 3 つの手法を組み合わせることで多視点カメラ画像から対象物体形状のモデリングを行い，そのモデリングによって得られた物体形状を用いて任意視点画像を生成する手法を実現する．

まず，背景差分における影領域の誤抽出の問題を解決する．現在提案されている任意視点画像生成手法では，対象物体領域抽出のための手法を検討し提案している例は少ない．前述した関連研究での目的はビデオ監視システムなどの動物体追跡が主である．しかし，画像の撮影から任意視点画像生成までの一連の処理の中での物体領域抽出は重要な役割を持つ．そこで任意視点画像生成手法の中の 1 つの手法として，任意背景において影領域を取り除くことを第一の目的とした対象物体領域抽出手法を提案する．

次に，対象物体形状復元アルゴリズムにおける計算量の問題を解決する．前述した通り，様々な復元アルゴリズムについて研究が進められているが，それらと同等に形状復元にかかる計算量を削減することが可能な独自のアルゴリズムを提案する．

最後に，カメラキャリブレーションの煩雑さを解決することと，任意視点画像精度向上を目的として，画像情報だけから仮想空間を定義する手法について提案し，その空間内で任意視点画像の精度を向上するために Microfacet Billboard 法を適用する．従来は対象物体の形状復元にはカメラキャリブレーションが必要とされていた．また，カメラキャリブレーションを行わない手法では，対象物体形状の復元は行わず，中間画像を生成するための 3 次元的な情報しか用いていなかった．しかし，ここで提案する仮想空間定義手法を用いれば，カメラキャリブレーションを行わなくても対象物体形状の復元を行うことが可能となり，現在では非常に煩雑な大規模な空間での応用も可能となる．

まず，第 3 章において任意背景から対象物体を抽出するための背景差分手法について述べる．そして任意背景においても影の誤抽出の問題に対しても有効であることを評価実験の結果を元に示す．

次に，第 4 章において視体積交差法と Octree を用いた高速な対象物体復元アルゴリズムについて述べる．このアルゴリズムは対象物体を撮影しながら任意視点を自動生成するための実時間処理を目標とする．そして本手法がどれだけ高速であるかを評価実験を元に

その有効性を示す。

そして、第 5 章では画像情報だけから定義される仮想空間を定義する手法、そしてこの仮想空間内での任意視点生成のための着色処理について述べる。擬似正射影グリッド空間 (Orthogonal Projective Grid Space, OPGS) と名づけたこの仮想空間が PGS と比較してどの程度歪みが改善されているかを実験を通して示し、ユークリッド空間との比較も行う。着色処理は Voxel への直接的な着色方法、Microfacet Billboarding の OPGS への適用について述べる。

また、これら 3 つの手法を組み合わせる任意視点画像生成手法について第 6 章にて述べ、この手法を用いて得られた任意視点画像を示す。そしてこれら手法を用いて構築したシステムの応用例として、2 つの遠隔地において対象物体を撮影した多視点カメラ画像からそれぞれの場所での対象物体のモデリングを行い、それを同じ仮想空間内で配置する仮想空間通信システムについて述べる。

## 1.5 本論文の構成

本論文の構成を述べる。

まず、第 2 章では理論として用いるカメラ校正・F-Matrix・エピポラ幾何、3次元データ構造等について述べる。次に、第 3 章では提案する背景差分手法について述べ、その有効性について記す。第 4 章では高速な物体形状復元アルゴリズムについて記述し、第 5 章では画像情報のみで仮想空間を定義する手法とその有効性について述べる。そして第 6 章では前述した 3 つの手法を用いた、対象物体形状のモデリングを行って任意視点画像を生成する手法について述べ、その結果についても示す。また、この手法を用いた具体例として共有仮想空間通信システムについて説明する。最後に、第 7 章にて本論文の結論を述べる。

## 第 2 章

### 理論

本章では、これから述べる提案手法に関連する理論について説明する。

## 2.1 カメラパラメータとカメラの校正

画像面を2次元射影平面とみなし画像面上の点  $x = [x, y]^T$  をとり、その斉次座標を  $\tilde{x} = [x_1, x_2, x_3]^T$  とする。同様に3次元空間は射影空間であるとみなし、3次元点  $X = [X, Y, Z]^T$  に対し、斉次座標を  $\tilde{X} = [X_1, X_2, X_3, X_4]^T$  とする。

このとき透視投影は式 (2.1) であらわされる。

$$\lambda \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} \quad (2.1)$$

ここで  $\lambda$  は任意の実数である。式 (2.1) は、

$$P_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.2)$$

とおくと、

$$\lambda \tilde{x} = P_f \tilde{X} \quad (2.3)$$

と表すことができる。

実際のカメラで得られる画像はアナログ信号であるが、コンピュータでは一般的に離散化されたデジタル画像で処理され、画素 (Pixel) と呼ばれる単位で記録される。したがって物理的座標  $x$  から画像座標  $m$  へ変換する必要がある。この変換は平行移動、縦横のスケール変換および焦点距離に応じたスケール変換を組み合わせたものであり、次のように記述することができる。

$$\tilde{m} = A \tilde{x} \quad (2.4)$$

## 2.1 カメラパラメータとカメラの校正

ここで  $\tilde{m} = [m_1, m_2, m_3]^T$  は画像座標  $m$  の斉次座標である。また  $A$  は  $x$  を  $m$  に変換する行列であり、焦点距離  $f$ 、画像中心座標  $(u_0, v_0)$ 、 $u$  および  $v$  方向スケールファクター  $k_u, k_v$  およびせん断係数  $k_s$  によって次のように表される。

$$A = \begin{bmatrix} fk_u & fk_v & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

式 (2.3) および式 (2.4) より 3次元の点  $X$  は画素  $m$  に以下のように投影されることがわかる。

$$\lambda \tilde{m} = AP_f \tilde{X} \quad (2.6)$$

$A$  はカメラ校正行列とよばれる。またこれは式から明らかなようにカメラの内部パラメータによって構成され、内部パラメータ行列とも呼ばれる。

カメラを中心としたカメラ座標系  $X$  からワールド座標系  $X_w$  への変換は 3次元の回転  $R$  および並進  $T$  によって次のように変換される。

$$X = RX_w + T \quad (2.7)$$

式 (2.7) は斉次座標を使えば次のように表せる。

$$\tilde{X} = M\tilde{X}_w \quad (2.8)$$

ここで、 $M$  は回転  $R$  と並進  $T$  を組み合わせた  $4 \times 4$  行列であり、次のとおりである。

$$M = \begin{bmatrix} R & T \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2.9)$$

$M$  はワールド座標に対するカメラの姿勢 (回転) および位置 (並進) によって決まる行列である。回転や並進はカメラの外部パラメータと呼ばれるので  $M$  のことをカメラの外部パラメータ行列と呼ぶ。

## 2.1 カメラパラメータとカメラの校正

式 (2.6) および式 (2.8) より，ワールド座標系における 3 次元空間の点  $X$  は 2 次元画像上の点  $m$  に次のように投影されることがわかる．

$$\lambda \tilde{m} = P \tilde{X} \quad (2.10)$$

ここで，式 (2.10) の  $P$  は，

$$P = AP_f M = A \begin{bmatrix} R & T \end{bmatrix} \quad (2.11)$$

で表せる  $3 \times 4$  行列である． $P$  は透視カメラ行列と呼び，内部パラメータおよび外部パラメータにより構成される． $P$  は  $A, R, T$  に分解することができ， $P$  が既知であるときそのカメラは校正済み，未知の場合は未校正であるという．また， $A, R, T$  を求めることをカメラの校正という．

$P$  を一般化し， $3 \times 4$  の行列  $P_p$  で表すと，

$$\lambda \tilde{m} = P_p \tilde{X} \quad (2.12)$$

$$P_p = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \quad (2.13)$$

となる．この  $P_p$  を射影カメラ行列とよぶ．

## 2.2 F-Matrix

複数の画像を用いて対象物体の3次元情報を得ようとする場合、画像中の点の各画像間の対応づけを利用する。そのためにはカメラの3次元座標、レンズの向きといったカメラパラメータが必要である。

ステレオ視のような3次元計測の場合、通常カメラパラメータは撮影時に測定し、既知として後の処理に与えることが多い。システムが固定系であればそれがかまわないが簡易で移動可能なカメラシステムを考える場合、撮影した画像からカメラパラメータを得られることが望ましい。

そこで本手法では画像間の対応点から比較的精度が高くカメラパラメータを求められる Fundamental Matrix(以下 F-Matrix)[19] を用いたキャリブレーション法を用いるものとする。F-Matrix とはステレオ視の対応づけの問題から考えられたエピポーラ幾何から生じたものであり、 $3 \times 3$  の行列で表され、画像間の関係を示すものである。実際には F-Matrix を計算することではカメラパラメータは得ることはできないが、画像間の対応関係が明らかになるため対応点の探索が可能になり、カメラパラメータを得たとき同じように扱うことができる。以下でエピポーラ幾何における F-Matrix の理論をのべる。

## 2.2.1 エピポーラ幾何における F-Matrix

エピポーラ幾何 [56] はステレオ視の対応づけの問題を解くための幾何である。図 2.1 のように 2 つのカメラで 3 次元空間の同じ点を見ているとすると、その点と 2 つのカメラの中心、そして両方の画像におけるその点の投影が 1 つの平面にあることになる。この平面をエピポーラ平面とよび、エピポーラ平面とそれぞれの画像との交点線をエピポーラ線、エピポーラ線と各画像の交点をエピポールとよぶ。

両カメラの幾何関係が既知の場合、一方の画像で 1 点が与えられるとエピポーラ平面と各エピポーラ線が決定する。他方の画像上での対応点はエピポーラ線上に限定されるためエピポーラ線上の 1 次元の探索でよいことになる。

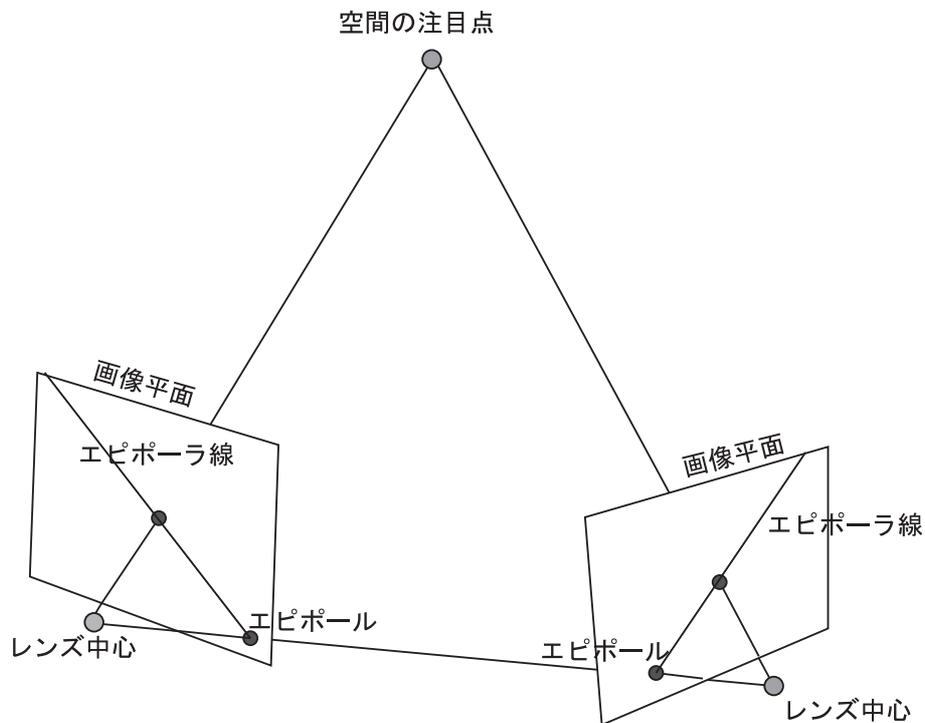


図 2.1: エピポーラ幾何の概念図

正規化カメラを仮定する．画像 1 上の点を  $x$  画像 2 における対応点を  $x'$  とし，それぞれの 3 次元座標を  $\tilde{x}, \tilde{x}'$  とする．また，一方のカメラを他方のカメラの位置と姿勢にする移動は回転行列  $R$  と並進ベクトル  $t$  によって表されるとする．このとき，両画像における対応点と両カメラのレンズの中心が共平面なので，ベクトル  $\tilde{x}$  と  $R\tilde{x}' + t$  が共平面である．つまり

$$\tilde{x}^T (t \times (R\tilde{x}' + t)) = 0 \quad (2.14)$$

ここで，変換

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{\times} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \quad (2.15)$$

を用いると，式 (2.14) は

$$\tilde{x}^T (t \times (R\tilde{x}' + t)) = \tilde{x}^T [t]_{\times} (R\tilde{x}' + t) = \tilde{x}^T E \tilde{x}' = 0 \quad (2.16)$$

と書ける．ここで， $E = [t]_{\times} R$  と式 (2.16) は 3 次元座標と関係なく 2 次元画像座標とカメラ運動のみを変数とした方程式で，エピポーラ方程式と呼ばれる．行列  $E$  を Essential Matrix(以下 E-Matrix) と呼ぶ．

通常画像を扱う場合はデジタル画像を用いるため， $\tilde{x}, \tilde{x}'$  のデジタル画像の座標をそれぞれ  $\tilde{m}, \tilde{m}'$  とすると，式 (2.16) は

$$\tilde{x}^T E \tilde{x}' = \tilde{m}^T E \tilde{m}' = 0 \quad (2.17)$$

と書ける．ここでカメラ 1 の内部パラメータを表す行列を  $A$ ，カメラ 2 の内部パラメータを表す行列を  $A'$  とおくと，

$$F = A^{-T} E (A')^{-1} \quad (2.18)$$

となり，この  $F$  が F-Matrix にあたる．F-Matrix はカメラの内部パラメータと，外部パラメータの両方を含む．また画像 1 におけるエピポール  $e$  は画像 2 の全ての点に対応するので

$$e^T F m' = 0, \quad \forall m' \quad (2.19)$$

が成り立つ．そのためには次式

$$F^T e = 0 \quad (2.20)$$

が成り立つ必要がある．同様に  $e'$  は

$$F e' = 0 \quad (2.21)$$

が成り立つ．よって F-Matrix が与えられると， $e$  と  $e'$  は  $F^T F$  の最も小さい固有値に対応する固有ベクトルとして求められることになる．

## 2.2.2 点对応による F-Matrix の推定

ここでは画像間の点对応が与えられたときの，中心射影の F-Matrix 推定法について述べる．画像 1 における点  $m_i = [u_i, v_i]^T$  と画像 2 における対応点  $m'_i = [u'_i, v'_i]^T$  はエピポーラ方程式  $\tilde{m}_i E \tilde{m}'_i$  をみたす．F-Matrix の要素をベクトルに並べ，この方程式を変形すると，

$$u_i^T f = 0 \quad (2.22)$$

となる．ここで，

$$u_i = [u_i u'_i, u_i v'_i, u_i, v_i u'_i, v_i v'_i, v_i, u'_i, v'_i, 1]^T$$

$$f = [f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}]^T$$

$f_{ij}$  は F-Matrix の  $i$  行  $j$  列の要素である．

対応が  $n$  点が与えられたとき，式 (2.22) を積み重ねて，次式

$$Uf = 0 \quad (2.23)$$

を解けばよいことがわかる．ここで，

$$U = \begin{bmatrix} u_1^T \\ \vdots \\ u_n^T \end{bmatrix}$$

中心射影の場合，F-Matrix のランクは 2 で，スケールは任意であることから，自由度は 7 となる．よって最低 7 点の対応点が得られれば解けることになる．8 点以上与えられれば，線形解法が存在するといえる．最小自乗法に基づいて

$$\min_F \sum_i (\tilde{m}_i E \tilde{m}'_i)^2 \quad (2.24)$$

あるいは等価的に

$$\min_f f^T U^T U f \quad (2.25)$$

を解けば良いことになる．ここで， $f$  のスケールは任意である． $\|f\|=1$  を拘束条件として加え，式 (2.25) を解くと， $f$  は  $U^T U$  の最小の固有値に対応する固有ベクトルとして求められる．

## 2.3 射影グリッド空間

射影グリッド空間 (Projective Grid Space : PGS) は、基準となる2台のカメラの中心射影によって構成される3次元空間である。3次元空間を離散的に扱う場合、2次元画像における単位面積である画素 (Pixel) と同様に3次元の単位体積は Voxel と呼ばれる。またここでは Voxel を区切る直線をグリッドと呼ぶことにする。

図 2.2 はカメラシステムと、一般的に3次元空間を離散的に扱う場合のユークリッドグリッド空間 (Euclidian Grid Space : EGS), 射影グリッド空間との関係を示したものである。EGS はカメラとは関係なく、ある1点を原点として定義された直交3次元空間であり各 Voxel の大きさも等しい。これに対し、射影グリッド空間は複数台あるカメラのうち2台を基底カメラとし、そのカメラ2台のカメラによって中心射影的に決定される3次元空間である。この図のように射影グリッド空間は一般のユークリッド空間から見ると各3軸及び各グリッドは直交するとは限らない。また各 Voxel の大きさも一定ではなく、基底カメラに近いほど小さく、遠いほど大きくなるという性質を持つ。

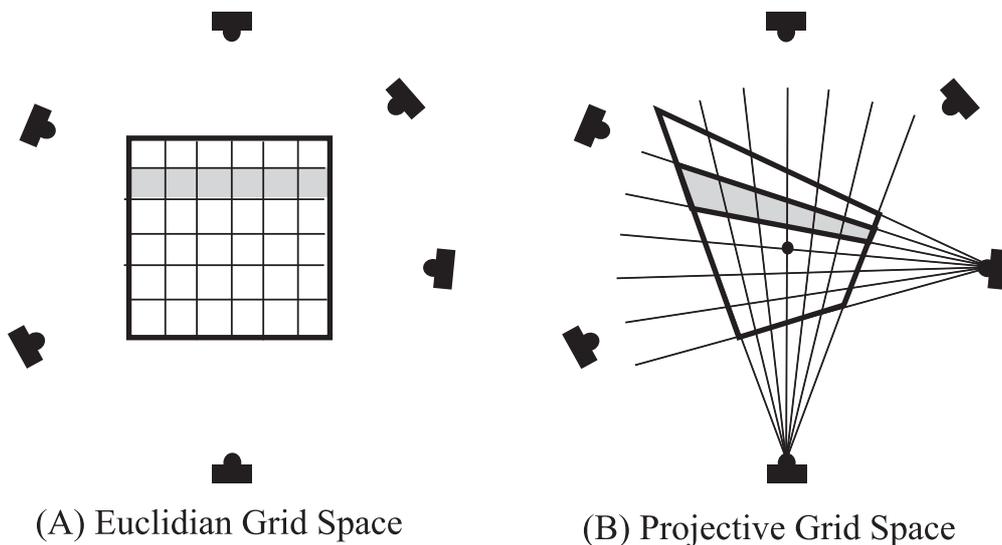


図 2.2: Euclid 空間と Projective Grid Space

## 2.3 射影グリッド空間

次に実際の射影グリッド空間の定義について述べる．まず複数台のカメラシステムのうち2台を選択し，基底カメラ1，基底カメラ2とする．基底カメラ1から得られる画像の $X$ 軸， $Y$ 軸をカメラの視点から中心射影的に空間に投影したものを射影グリッド空間の $P$ 軸， $Q$ 軸とする．同様に基底カメラ2から得られる画像の $X$ から射影グリッド空間の $R$ 軸を決定する．これら $P, Q, R$ 軸を3軸として定義される3次元空間が射影グリッド空間である．また射影グリッド空間における座標は各画像の画素とその視点とを結ぶ直線によって定義されるものとする．つまり射影グリッド空間の $P$ 座標 $Q$ 座標がそれぞれ基底カメラ1の画像の $X$ 座標 $Y$ 座標に対応し， $R$ 座標が基底カメラ2の $X$ 座標に対応する．基底カメラ以外の他の全てのカメラもこの座標系を用いるものとする．

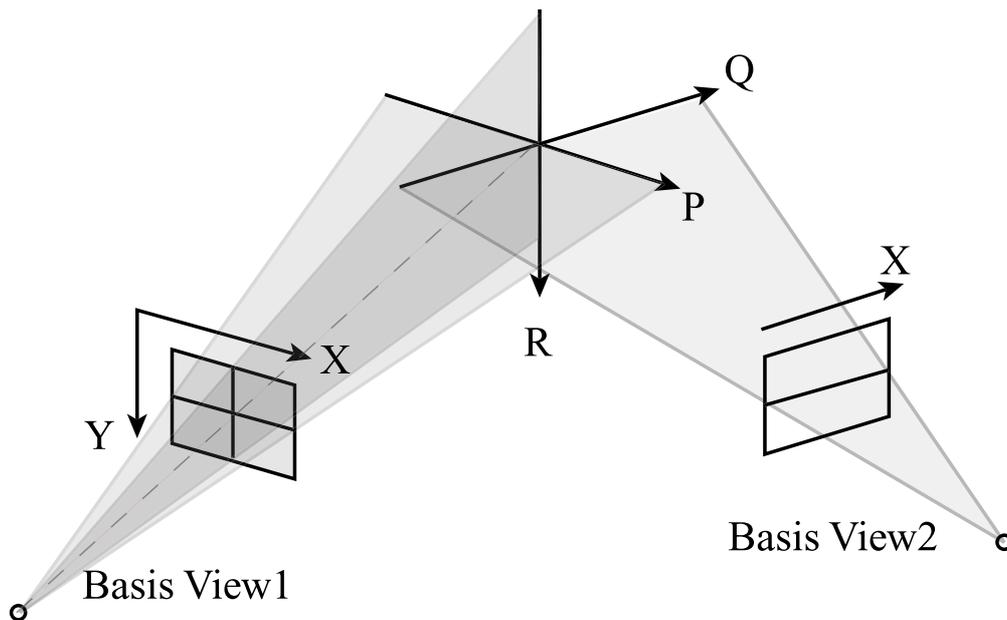


図 2.3: 射影グリッド空間の定義

### 2.3 射影グリッド空間

次に射影グリッド空間における視点位置の3次元座標の定義を示す．基底カメラ1, 2の各視点を  $C_1, C_2$  としその他カメラの視点を  $C_i$  とする (図 2.4)．射影グリッド空間の定義より基底カメラ1から得られる画像の  $X$  軸,  $Y$  軸によってカメラ1の視点  $C_1$  から中心射影的に  $P$  軸,  $Q$  軸が定められているから,  $C_1$  の  $P$  座標,  $Q$  座標は一意には定まらずあらゆる値を取りうるが, ここでは処理の汎用性を考えて基底カメラ1の画像中央の  $X$  座標  $X1_c, Y$  座標  $Y1_c$  をそれぞれ  $C_1$  の  $P$  座標,  $Q$  座標とする．射影グリッド空間の  $R$  軸は画像2の  $X$  軸として定義されることから, 画像2に対するエピポールとなる  $C_1$  の  $R$  座標は  $e_{21}$  の  $X$  座標 ( $e_{21_x}$ ) となる．つまり,  $C_1$  の座標は  $C_1(X1_c, X2_c, e_{21_x})$  になる．同様に,  $C_2$  の射影グリッド空間における  $R$  座標も一意には定まらないが, 基底カメラ2の画像中央の  $X$  座標  $X2_c$  を用いて定義されるものとする． $C_2$  は画像1に対するエピポールとなるので, その  $P$  座標,  $Q$  座標はそれぞれ  $e_{12_x}, e_{12_y}$  となる．つまり,  $C_2$  の座標は  $(e_{12_x}, e_{12_y}, X2_c)$  と表すことができる．また, 基底カメラ1, 2以外のカメラの視点  $C_i$  の座標は, 基底カメラ1, 2のエピポールの座標を用いて,  $C_i(e_{1i_x}, e_{1i_y}, e_{2i_x})$  であることがわかる．

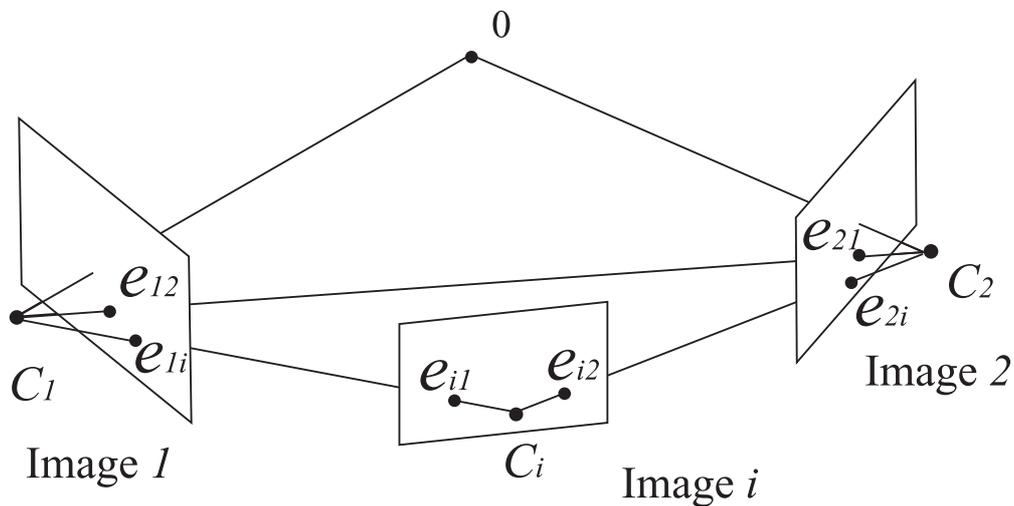


図 2.4: 射影グリッド空間における視点座標

## 2.4 視体積交差法

多視点カメラで撮影した被写体の3次元形状を復元するための手法の1つとして、視体積交差法 [23] がある。この手法は多視点カメラで撮影した被写体のシルエットを利用する。

透視投影モデルのある1つのカメラ画像から得られたシルエットから、その対象物体の角錐形の物体形状を得ることができる。これをすべてのカメラ画像において行い、各カメラ画像から得られた角錐の積を取ることで対象物体の3次元形状を復元する。図にその概念図を示す。今、 $I$  を全てのシルエット画像の集合とし、 $M_i$  を  $i$  番目のシルエット画像から復元された角錐形の復元モデルとすると、視体積交差法は次のような数式で表現することができる。

$$M_I = \bigcap_{i \in I} M_i \quad (2.26)$$

$$\lim_{I \rightarrow \infty} M_I = M \quad (2.27)$$

$M$  は対象物体と等しい形状の復元モデルを意味する。つまり、カメラの数が無限であれば実際の形状と等しくなることを意味する。そのため、視体積交差法ではより多くの多視点カメラ画像を用いれば、復元モデルがより実際の形状に近づくようになる。

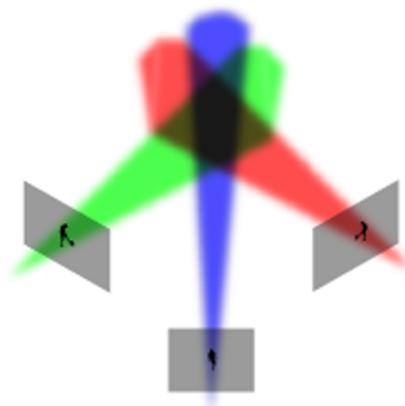


図 2.5: 視体積交差法

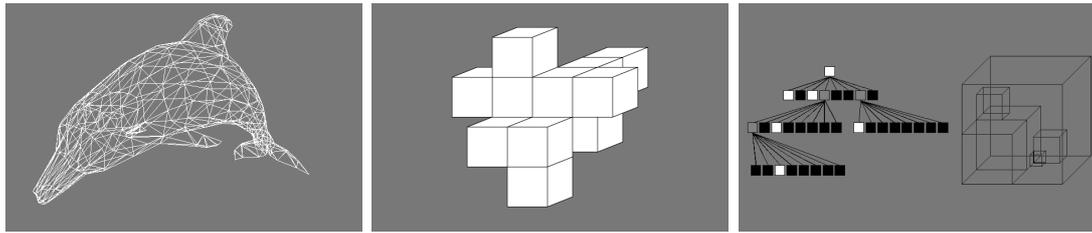
## 2.5 3次元情報表現形式

3次元空間や3次元形状の表現について様々な表現形式が提案されている。Baker[2]は多視点シルエット画像からワイヤーステイクモデルを構築する手法を提案した。ワイヤーステイクモデルは制御点情報のみで構築できるため、情報量が少なくてすむ利点がある。このワイヤーステイクモデルは現在ゲーム等で使用されている最も一般的な表現方法で、頂点情報とポリゴンへマッピングするテクスチャ情報を使用して3次元形状が表現されている。近年は一度に多数の頂点やポリゴンを扱えるようになり、3次元モデルの形状は高精細になり実物と区別がつかないようなモデルが実時間処理で扱えるようになってきている。

Martin と Aggarwal[29] は3次元形状を Voxel という単位体積を用いて表現した。Voxel モデルは高い精度で対象物体を表現できるが、情報量が多くなってしまいうため、全ての Voxel を操作する場合の計算量は膨大である。Voxel の用途としてボリュームデータや形状の表現がある。ボリュームデータの表現としてレイキャスティング法 [7] がある。この手法はある視点からの視線の延長上にある Voxel の数を濃度として計算し、それを2次元画像上に投影することで対象物体全体の分布を把握する手法である。視線がある Voxel を通る場合はその Voxel の値をそのまま用い、Voxel どうしの間を視線が通過する場合は周囲の Voxel 値で補間することで自然な画像を生成することが可能となる。そのため CT 画像など内部情報を可視化する場合には Voxel を用いたレイキャスティングがよく用いられる。また、形状の表現として Voxel が用いられる場合、各 Voxel に色や属性情報を割り当ててそれをレンダリングする。Seitz らの Voxel coloring[37] はそれらの手法の1つである。Voxel coloring ではカメラからの距離ごとに空間を区切ったレイヤー順に Voxel に対して走査を行う。まず、あるレイヤー内の Voxel を全てのレイヤーの外にあるカメラ画像へ投影し、投影された画素の色を取得する。異なる画像において得られた画素の色が閾値より小さい場合、その Voxel は全てのカメラから見る事ができるものとして物体を構成 Voxel とし、そしてその画像の色を割り当てる。そのため、このアルゴリズムでは異なるカメラ間での画素の色の比較が重要となる。この画素の色の比較・判定を改善するために、Broadhurst ら [6] は画素の一致判定に確率を導入して誤判定を減らす手法を提案した。

また、Potmesil[35] や Szeliski ら [40] は Octree と呼ばれる八分木による空間表現を利用した対象物体復元アルゴリズムを提案している。これらの手法における八分木では、対象空間全体を再帰的に分割していったその各々の状態が保存される。このデータ構造を用

## 2.5 3次元情報表現形式



(a) ワイヤースケルトン

(b) Voxel

(c) Octree

図 2.6: 様々な表現形式

いることにより処理時間を短縮することができる。

## 第3章

# カラー画像と視差画像による背景差分手法

## 3.1 はじめに

1.3.1 章にて述べたように現在の背景差分手法には様々な問題点がある。そこで、この問題点のうちの一つ、影領域の誤抽出を自然背景下において解決することを第一の目標とする背景差分手法を提案する。

本章では、まず影領域の誤抽出がどのようにして起こるのかについて説明し、それを解決するために用いた本手法について説明する。また、本手法の評価実験を行い、影領域の誤抽出がどの程度改善されたかについて説明する。

### 3.1.1 影の誤抽出

まずはじめに、影領域の誤抽出の原因について説明する。理想的な白色光源下において、その光源が画素の色に与える影響は RGB 形式の色の要素からなる色空間内のその色を表すベクトルの延長線上への変化のみである。図 3.1(a) にその概念図を示す。対象物体を含む画像（以下、原画像と記述）内で影領域となっている画素の色  $c_c = (r_c, g_c, b_c)$  と、背景画像内での同じ位置の画素の色  $c_b = (r_b, g_b, b_b)$  との関係は、原点と  $c_b$  を結ぶ直線方向だけの变化に留まる。つまり、

$$c_c = \alpha c_b \quad (3.1)$$

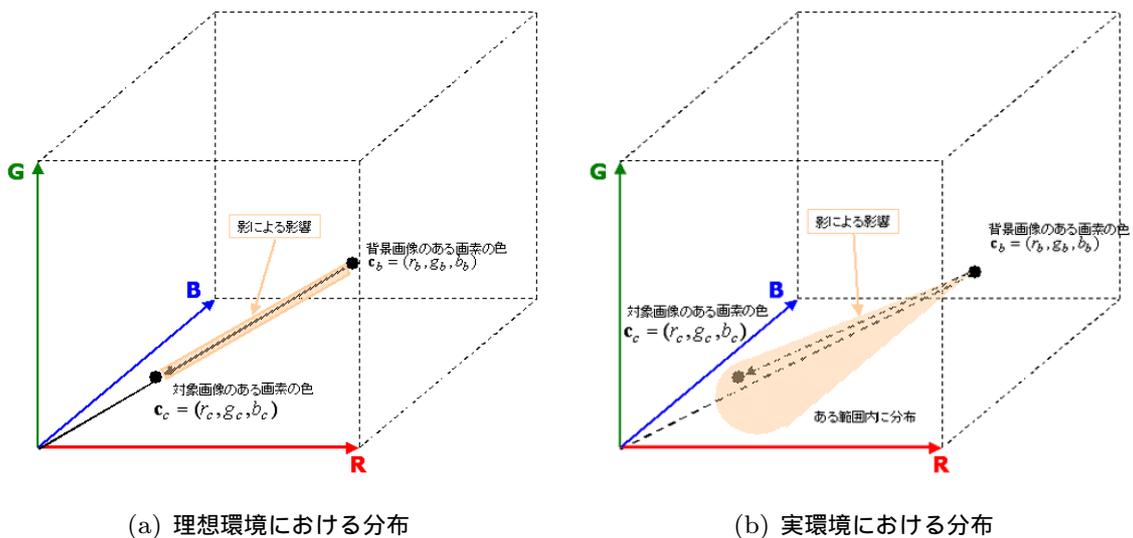


図 3.1: 理想環境と実環境における画素の分布

### 3.1 はじめに

の関係が成り立つ。なお、 $\alpha$  は光源の影響を表す係数である。これは、原画像、または背景画像内においてその画素にハイライトが起こらない限り成り立つ。よって理想的な環境ではその直線方向のみの色に条件を絞って影領域かどうかの判定を行えばよい。しかし、実際の環境では図 3.1(b) に示すように、白色ノイズの影響によって式 3.1 の関係が成り立たず、直線の延長方向のみならず、ある範囲内に分布することがわかっている。そして、白色ノイズが影響を与えているその分布には規則性がない。この不規則な分布によって影領域の誤抽出が現在でも解決困難な問題として存在している。

## 3.2 提案する背景差分手法

本手法では前述した影領域誤抽出の理由から、カラー画像のみならず視差画像を用いることで解決を図る。視差画像は2つのカメラ画像へステレオ法を用いることによって得られる距離情報を持った画像である。図 3.2 に示すように、影は常にある平面上に存在するため、視差画像内には影の影響が現れない。そして、対象物体領域に対しては物体が存在することで視差画像内でも物体を表す領域が存在する。よって視差画像の背景差分を行うことで影領域のみを取り除くことが可能であると考えられる。しかし、視差画像はステレオ法を適用した際の窓領域の大きさに応じて解像度が荒くなってしまうという側面も持つ。そこで、視差画像は影領域を取り除くためだけに用い、対象物体領域を抽出するためにカラー画像を用いる。

まず、本手法の流れを図 3.3 に示し、図 3.4 に背景差分アルゴリズムを表した擬似コードを示す。ある座標におけるカラー背景画像の画素値を  $c_b$ 、視差背景画像の画素値を  $d_b$  とし、現在対象としているカラー画像の画素値を  $c_c$ 、視差画像の画素値を  $d_c$  とする。

ここで、 $p$  は判定処理を行った画素の座標における得られたシルエット画像の状態を表す。また、 $th_d, th_U, th_L, th$  はそれぞれ視差の閾値、色の上限閾値、色の下限閾値、角度の閾値を表す。本手法ではこれらの閾値を経験的に求めた値に手動設定して背景差分を行う。これは、自動で閾値を求めるよりも環境に応じて手動で値を設定した方がより対象物体の抽出精度を上げることができるためである。

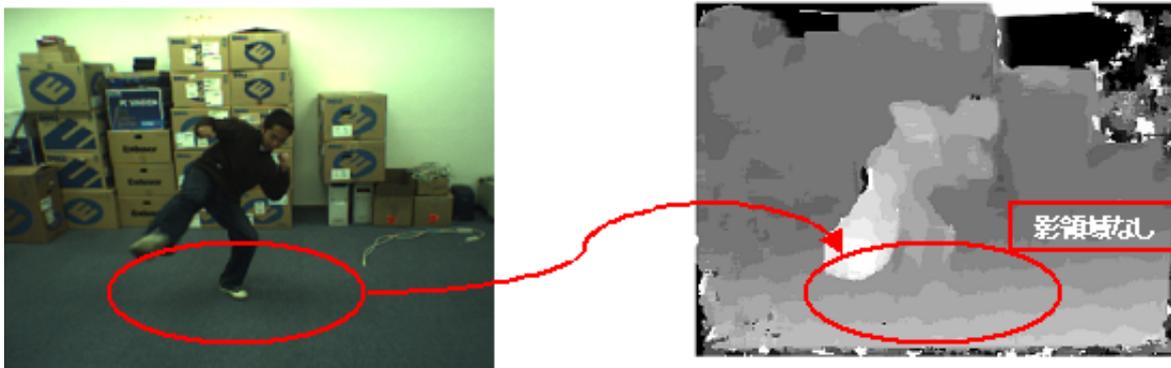


図 3.2: カラー画像と視差画像

### 3.2 提案する背景差分手法

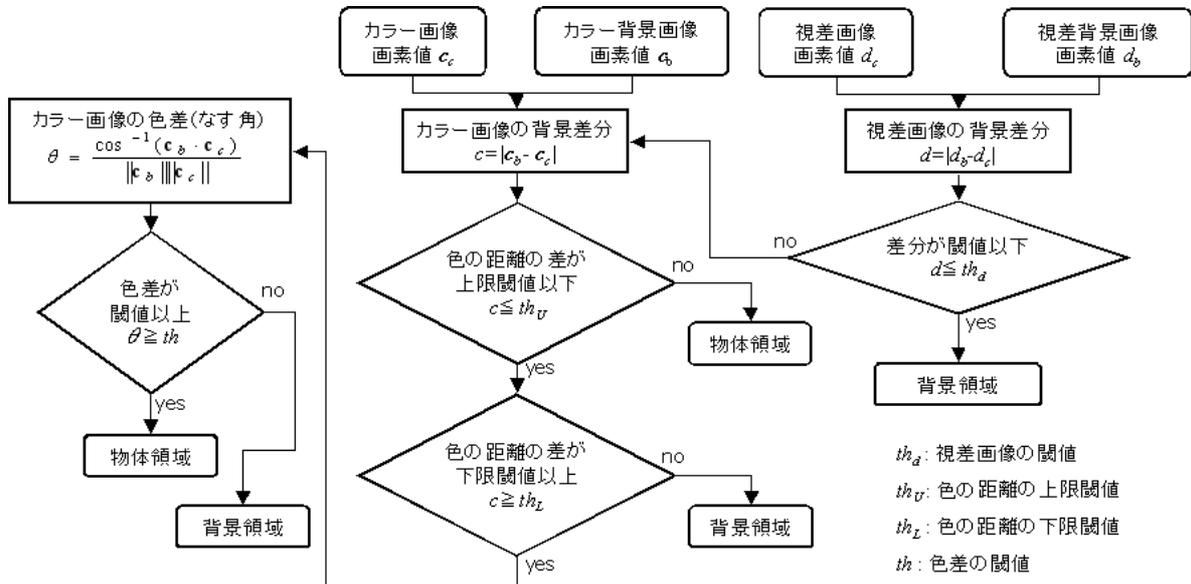


図 3.3: 背景差分手法の流れ

```

if  $|d_b - d_c| > th_d$  then
  if  $\|\mathbf{c}_b - \mathbf{c}_c\| > th_U$  then
     $p = SILHOUETTE$ 
  else
    if  $\|\mathbf{c}_b - \mathbf{c}_c\| < th_L$  then
       $p = BACKGROUND$ 
    else
       $\theta = \cos^{-1} \left( \frac{\mathbf{c}_b \cdot \mathbf{c}_c}{\|\mathbf{c}_b(x,y)\| \|\mathbf{c}_c\|} \right)$ 
      if  $\theta > th$  then
         $p = SILHOUETTE$ 
      else
         $p = BACKGROUND$ 
  else
     $p = BACKGROUND$ 
  
```

図 3.4: 背景差分手法擬似コード

### 3.2 提案する背景差分手法

本手法では、まず視差の差分を取り閾値以下ならばその画素は背景領域とする。この判定により影領域でも背景領域として検出することができる。視差画像において影領域は視差の値が床と同値になるため、視差背景画像との差分を取ることで影領域を削除することが可能となる。このように、このステップにおいて大部分の背景領域を決定することが可能となる。

次に、視差の閾値より大きい場合には、カラー画像の画素値を RGB 成分を持つ 3 次元ベクトルと考えると距離の差分を取る。このベクトル空間での光源の影響は、ベクトルの大きさが変化するだけで方向はほとんど変化しないため、もし対象としている画素が対象物体領域に含まれているのならこの差は大きな値となる。そこで、この差分が上限閾値より大きければシルエット領域とする。しかし、光源の影響によるノイズが含まれてベクトルの大きさが変化している可能性があるため、上限閾値より小さい場合には、下限閾値との比較を行う。下限閾値よりも小さければノイズがのった対象物体領域ではないため、背景領域とする。もし下限閾値よりも大きければカラー背景画像の画素値ベクトルと現在対象としているカラー画像の画素値ベクトルとのなす角を求め、角度の閾値との比較を行う。光源の影響によるベクトルの角度の変化は小さいと考えられるため、この比較により光源の影響による画素値の変化を考慮した最終的な判定を行うことができる。

なお、本手法では視差画像の判定処理を先に行っているが、カラー画像の判定処理を先に行っても同じ結果を得ることができる。しかし、視差画像での判定処理は 1 つのみであり、かつ単純な除算のみである。そしてこの判定処理によって、続くカラー画像への処理を行わないケースが多々存在する。よって視差画像の判定を先に行うことで全体的な計算時間の向上が見込めるため、視差画像の判定を先に行っている。

図 3.5 に背景差分に用いた背景画像、カラー原画像と視差画像、上記のアルゴリズムによって得られたシルエット画像を示す。この図からもシルエット領域が視差画像を用いない手法よりも正確に抽出でき、足元の影領域を取り除けていることがわかる。

### 3.2 提案する背景差分手法



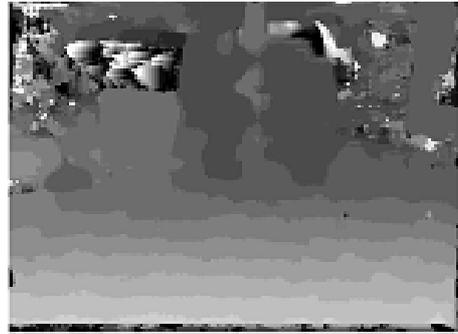
(a) カラー原画像



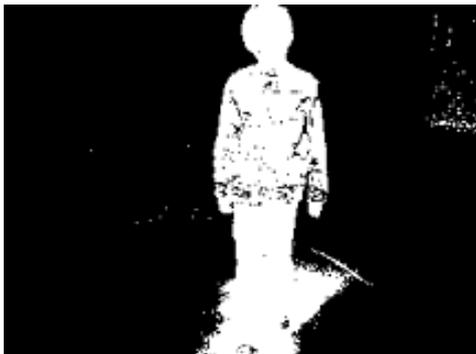
(b) カラー背景画像



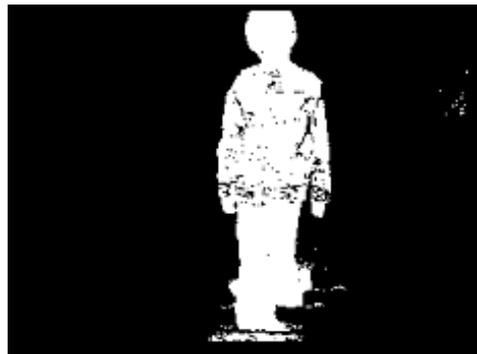
(c) 視差原画像



(d) 視差背景画像



(e) カラー画像のみの差分結果



(f) 本手法による差分結果

図 3.5: 背景差分の比較と結果

### 3.3 評価実験

本手法による影領域誤抽出の割合を評価するための実験について述べる。

本手法では視差画像を用いたことで影領域の誤抽出を軽減することが可能である。そこで、視差画像を用いた場合と用いない場合とでの影領域の誤抽出の度合いを比較する。影領域は対象とする画像を見て、影領域と推測される領域を手動で指定する。また、視差画像を用いない場合の背景差分手法は、本手法のカラー画像のみの処理の流れに沿うものとする。

評価実験における条件は次の通りである。実験時に用いた各種閾値もこの条件で示す。これら値は対象画像に対して良好な結果が得られた時の手動で選択した値である。また、実験に用いたカラー画像と視差画像を図 3.6 に示す。左列がカラー画像，右列が視差画像である。

- 撮影条件：300msec おきに撮影した 4 枚の画像
- 画像解像度：320 × 240
- カラー画像色形式：24bit RGB フォーマット
- 視差画像色形式：8bit グレースケールフォーマット
- 視差閾値 ( $th_d$ )：16
- 色下限閾値 ( $th_U$ )：5
- 色上限閾値 ( $th_L$ )：15
- 色差閾値 ( $th$ )：0.2

影領域を指定したカラー画像と実験結果を図 3.7 に示す。左列の図中の赤い領域が影領域を指定した領域である。また、中列と右列の画像中の青い領域が正しく影領域と判定された領域，赤い領域は物体領域と誤判定された領域である。また、視差画像を用いた場合と用いなかった場合とでの誤判定率を表 3.1 に示す。

まず、図 3.7 を見ると視差画像を用いた場合の方が赤い領域が少ないことが確認できる。

表 3.1: 誤り率の比較 (誤り率は%)

差分手法	画像 1	画像 2	画像 3	画像 4	画像 5	平均
視差画像あり	13.1	16.7	19.0	9.8	3.07	12.3
視差画像なし	83.9	73.9	60.9	61.0	43.9	64.7

### 3.3 評価実験

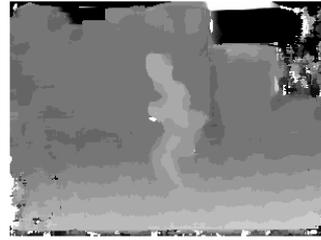
これは、視差画像を用いたことで影領域と指定した範囲での背景差分がより正確に行えていることを示している。また表 3.1 から、視差画像を用いることで誤り率が平均で 12%程度に抑えられていることが確認できる。視差画像を用いない場合との比較でも、視差画像を用いた場合の方が視差画像を用いない場合よりも平均値で 5 倍程度誤り率が改善されていることが確認できる。

以上の実験から、本手法によって影領域誤抽出の問題を改善することが可能であると結論付けられる。しかし、問題点も同時に確認できる。本手法では視差画像を用いることで影領域の誤抽出の改善を図っているが、視差画像では足元と床とが同化している領域が存在している。そして視差画像の差分時に、その床領域と同化した足領域が床と判定されてしまい、足元部分が背景と誤判定されてしまっている。この問題は、ステレオ法の窓領域を変更して視差画像の荒さを小さくすることで改善が期待される。本手法の第一の目的は影領域の誤判定であり、また任意視点画像生成手法へ用いることであるが、ステレオ法の窓領域を小さくすることで、背景差分にかかる計算時間が増加してしまうと処理全体での計算時間も増加してしまう。よって実際の任意視点画像生成手法へ適用する際は、窓領域の大きさも含めたパラメータの最適化を環境に応じて行うことが必要であると考えられる。

### 3.3 評価実験



(a) カラー画像 1



(b) 視差画像 1



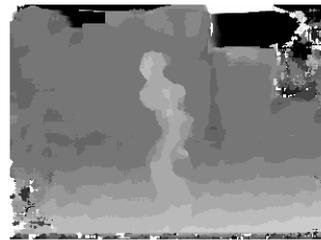
(c) カラー画像 2



(d) 視差画像 2



(e) カラー画像 3



(f) 視差画像 3



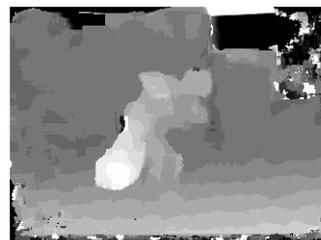
(g) カラー画像 4



(h) 視差画像 4



(i) カラー画像 5



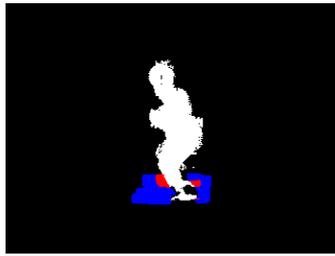
(j) 視差画像 5

図 3.6: 評価実験対象画像

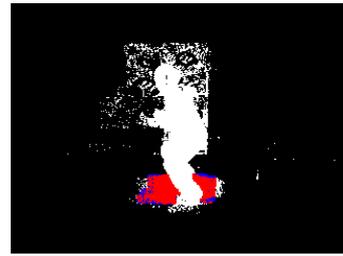
### 3.3 評価実験



(a) 影領域指定画像 1



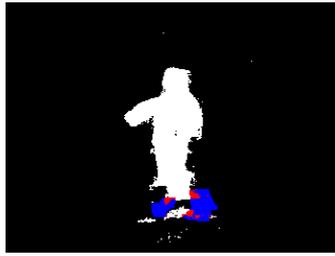
(b) 視差画像あり 1



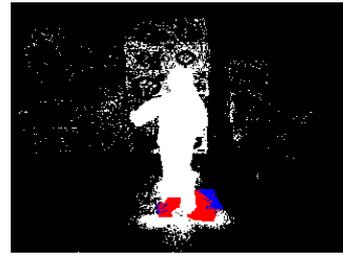
(c) 視差画像なし 1



(d) 影領域指定画像 2



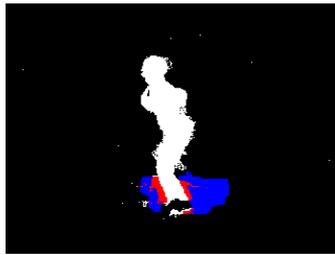
(e) 視差画像あり 2



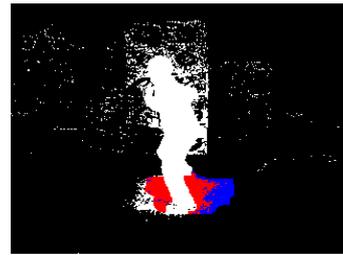
(f) 視差画像なし 2



(g) 影領域指定画像 3



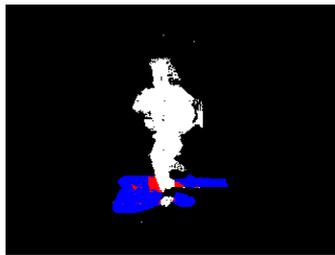
(h) 視差画像あり 3



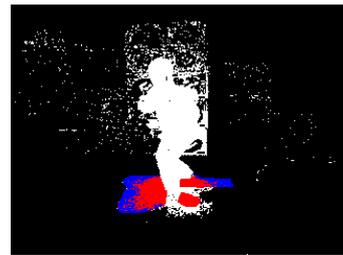
(i) 視差画像なし 3



(j) 影領域指定画像 4



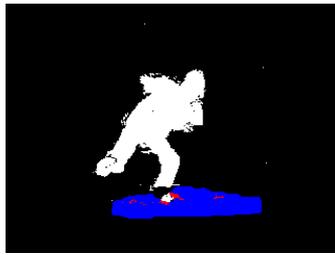
(k) 視差画像あり 4



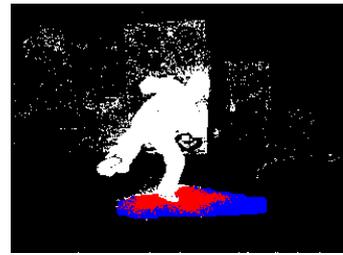
(l) 視差画像なし 4



(m) 影領域指定画像 5



(n) 視差画像あり 5



(o) 視差画像なし 5

図 3.7: 影領域指定画像と実験結果

## 第 4 章

### 高速な物体形状復元手法

## 4.1 はじめに

本章では第 1.3 節にて述べた，3次元物体形状復元における  $O(N^3)$  の計算量を削減するために提案する復元手法について述べる。

前述したように，3次元的な物体形状を復元するには  $O(N^3)$  の計算量が必然的に必要となり，それを解決するための手法が提案されている。そこで，この計算量を削減するための高速な復元手法の1つを提案し，その手法について述べる。本手法はデータ構造として Octree 構造を利用する。そして基本的な Octree 生成アルゴリズムに独自の計算量削減処理を加えた復元手法について提案し，その内容について説明する。

本章ではまず提案する物体形状復元アルゴリズムについて説明する。その後，本手法を用いた場合と全 Voxel を均等に調べた場合の計算時間の比較を行い，本手法の優位性を示す。

## 4.2 提案する物体形状復元手法

ここでは提案する物体形状復元手法について示す。まず、手法全体の概要について説明し、その後、その詳細について説明する。

### 4.2.1 物体形状復元手法の概要

Octree 構造は前述した通り、対象とする立方体の状態に応じて木構造を生成する手法である。ここで、立方体の3つの状態を次のように定義する。

- *BLACK* : 立方体は物体領域に含まれない
- *GRAY* : 立方体は物体領域に部分的に含まれる
- *WHITE* : 立方体は物体領域に含まれる

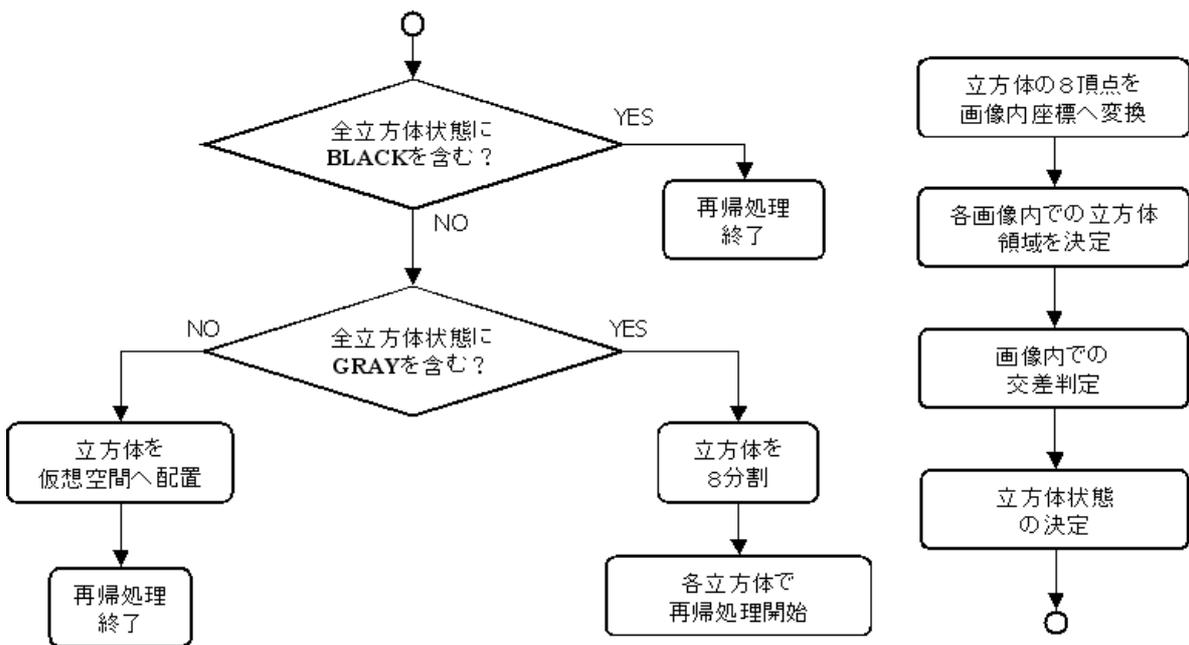


図 4.1: 物体形状復元手法の流れ

```

for(i=1;i ≤ cameraNumber;i++){
    refer the parent temporary cube type;
    if referred type is GRAY then
        project 8 vertices of the cube into image;
        decide search rectangle;
        search the rectangle;
        decide the temporary cube type;
        if temporary cube type is BLACK then
            store the temporary type as BLACK;
            exit this loop;
        else if current cube type is not BLACK then
            store the temporary type;
        else
            decide the temporary cube type as WHITE;
    }
    refer the all temporary cube types;
    if one of the temporary cube types is BLACK then
        decide the node type as BLACK;
    else if all temporary cube types are WHITE then
        decide the node type as WHITE;
    else
        decide the node type as GRAY;
        make 8 children node;
        for (j=1;j ≤ 8;j++) {
            iterate same process at each child node recursively;
        }
}

```

図 4.2: 物体形状復元手法擬似コード

## 4.2 提案する物体形状復元手法

図 4.1 に物体形状復元手法全体の流れを示す。また、処理を表す擬似コードを図 4.2 に示す。まず、対象としている立方体の 8 頂点を、各物体領域を抽出した画像（以下、「シルエット画像」と呼ぶ）内の対応する座標へと変換する。そして画像内の立方体領域を決定し、この立方体領域にシルエットがどのように含まれているかを交差判定によって求める。この交差判定によって、ある視点のシルエット画像における立方体の状態（以下、「一時状態」と呼ぶ）が決定するが、ここまでの処理を全ての視点のシルエット画像に対して行い、全視点で判定された一時状態の集合を求める。一時属性の集合に 1 つでも *BLACK* 状態が含まれる場合は、その立方体は物体領域に含まれないことになるため、立方体の最終的な状態を *BLACK* として、この立方体での再帰処理を終え、次の立方体へと処理を移す。そうでない場合は一時状態の集合に *GRAY* 状態が含まれるかどうかを調べる。もし *GRAY* 状態が含まれていれば、その立方体の最終的な状態は *GRAY* となり、分割対象となる。そして分割された 8 個の立方体へと再帰的に処理を続ける。そうでない場合、つまり全ての一時状態が *WHITE* の場合、その立方体は物体領域に含まれるため、最終的な立方体の状態は *WHITE* となり、この立方体での再帰処理を終了し、次の立方体へと処理を移す。なお、本手法では親の立方体の一時状態を参照することで、その子の代にあたる立方体での座標変換、交差判定処理を省略し、計算量を削減することが可能となる。

### 4.2.2 一時状態の決定

まず、対象とする立方体がそれぞれの視点のシルエット画像においてどのような状態であるか調べる必要がある。そこで、処理対象の立方体の 8 頂点を各シルエット画像平面上の対応する座標へと変換する。ユークリッド空間を想定している場合であれば、カメラキャリブレーションによって得られた射影変換行列を用いて座標変換を行う。また、後述

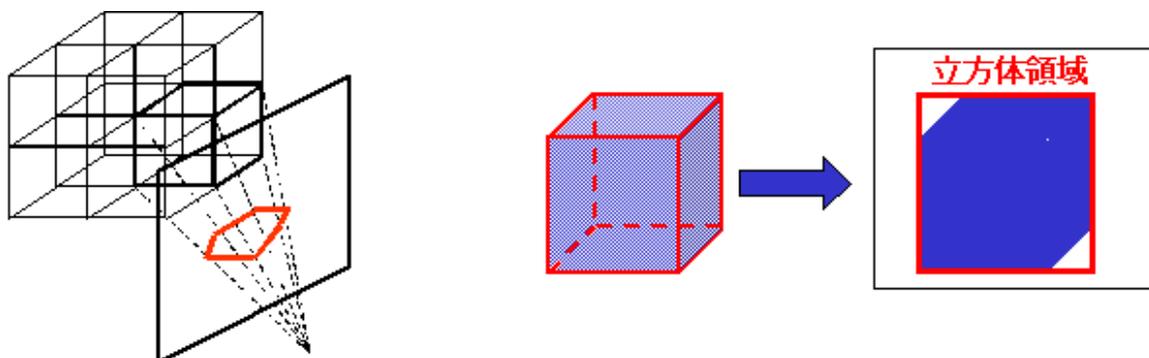


図 4.3: 画像内立方体領域決定

## 4.2 提案する物体形状復元手法

する OPGS 座標系であれば基底カメラと撮影カメラ間の Fundamental Matrix を用いて画像平面上の対応点へと座標変換を行う．OPGS とカメラ画像間での対応関係取得方法については第 5 章にて詳しく述べる．

座標変換後，画像内における立方体領域を決定する．画像内で立方体が現れる領域はカメラの位置関係により 6 角形になりえてしまう．しかし，この 6 角形の領域を正確に求めることは，計算時間のロスとなってしまう．そこで本手法では，投影された 8 頂点を囲む最小の矩形を画像内での立方体領域とみなして処理を行う．なお，ここで生じる形状の誤差は後の任意視点画像生成処理において吸収されるため見た目の影響はない．こうして得られた画像内立方体領域に対して交差判定を行う．図 4.3 に画像内立方体領域決定の概要図を示す．

次に，交差判定を行う．ここでの処理は，画像内立方体領域内にシルエット領域がどのように存在しているかを調べることである．まず，画像内立方体領域を 1 画素ずつ順にスキャンしていく．この時に画像内立方体領域に含まれる画素が全て背景領域の場合，現在対象としている立方体はその視点から物体領域に含まれないことが判明する．よってこの時は一時状態を *BLACK* とする．逆に，全ての画素がシルエット領域である場合，その立方体は 3 次元的な物体領域に含まれることがわかる．よってこの場合は一時状態を *WHITE* とする．画像内立方体領域に背景領域，シルエット領域どちらも含まれる場合は，その視点から見た場合の立方体は物体領域を部分的に含むことがわかる．よって，この時の一時状態は *GRAY* と決定する．図 4.4 に，この 3 つの場合の例を示す．図 4.4(a) のように画像内立方体領域が完全にシルエット外部に存在する場合は，一時状態は *BLACK* となる．逆に図 4.4(c) のように完全にシルエット内部に存在する場合は，一時状態は *WHITE* となる．そして図 4.4(b) に示すように，部分的に含まれる場合は一時状態は *GRAY* となる．

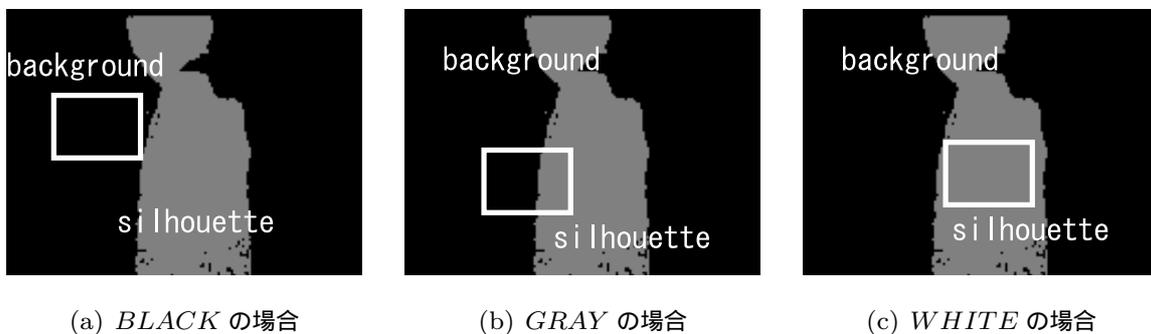


図 4.4: 一時状態決定の例

### 4.2.3 立方体状態の決定

前述した一時状態を全ての視点でのシルエット画像を元に求めた後、最終的な立方体の状態を決定する。この最終的な立方体状態の決定には視体積交差法の考え方を利用する。視体積交差法では各視点から得られる物体形状の共通領域がその多視点画像から得られる物体形状であると考え、これを Octree 構造を用いた視体積交差法として適用することは、各視点において判定された一時状態の集合を調べることと等価である。つまり、各視点のシルエット画像にその投影された立方体の一時状態が全て *WHITE* であるならば、その立方体は物体を構成すると判断する。逆に、1 つでも *BLACK* の一時状態が存在している場合、その立方体は物体を構成しないと判断する。最終的な立方体の状態が *BLACK* と *WHITE* の場合、この時点で立方体の状態が「あいまいな」状態でないことを表すため、これら状態の場合は再帰処理を終了し、次の立方体へと処理を移す。そして、それ以外の場合、つまり一時状態の集合に *BLACK* が含まれず、かつ *GRAY* の一時状態が 1 つ以上含まれる場合はその立方体はさらに分割可能と判断して 8 個の等しい大きさの立方体へ分割する。そして分割した 8 個の立方体に対し、前述した立方体の状態判定処理を再帰的に繰り返す。

ここまで述べた処理を、空間全体を表す 1 つの立方体の時点から開始し、あらかじめ定めた木の深さに達するまで再帰的に繰り返す。これにより、物体形状が大小様々な大きさの立方体によって表現することができる。

### 4.2.4 親の一時状態参照による計算量削減

本手法では親の節での各シルエット画像へ立方体を座標変換した際の、一時的な属性は子の節においても継承される性質を利用して、スタックに親の節での情報を蓄えておきそれを参照することで小立方体の探索回数を減らし、さらに計算量を減らしている。まず、対象となる立方体の画像内での探索領域を求め、小立方体の属性を調べる。そしてこの時の属性を画像の枚数分存在するスタックへプッシュしてゆく。再帰的に処理を繰り返すため、対象となる立方体の属性が決定すると子の節へと移り同様の処理を行うが、子の節では親の節での属性を保持するスタックを参照する。親の節における、各シルエット画像での探索領域の属性が対象物体を表す場合、子の節においても必ず物体を表す属性となる。結果として、シルエット画像において探索領域を求めることなく対象となる立方体の属性を決定することができる。

図 4.5 にその具体例を示す．例えば 4 つのカメラ画像を用いた時，ある階層の節での属性が

- カメラ 1 に投影した結果は *WHITE*
- カメラ 2 に投影した結果は *WHITE*
- カメラ 3 に投影した結果は *GRAY*
- カメラ 4 に投影した結果は *GRAY*

の条件のため，最終的に *GRAY* に決定したとする．この場合，Octree の分岐条件から子の階層に移り再帰処理を行うが，子の階層においてカメラ 1 とカメラ 2 へ分割した立方体を座標変換しても結果は必ず *WHITE* となる．つまりある階層において *WHITE* 属性が出現した場合，以後は必ず *WHITE* 属性になるといえる．そこで，親の階層における全てのカメラにおける一時的な立方体属性の結果をスタックに確保しておき，親の階層に相当するデータを参照することで無駄な座標変換計算やラスタスキャンをすることなく再帰処理を進めることができる．

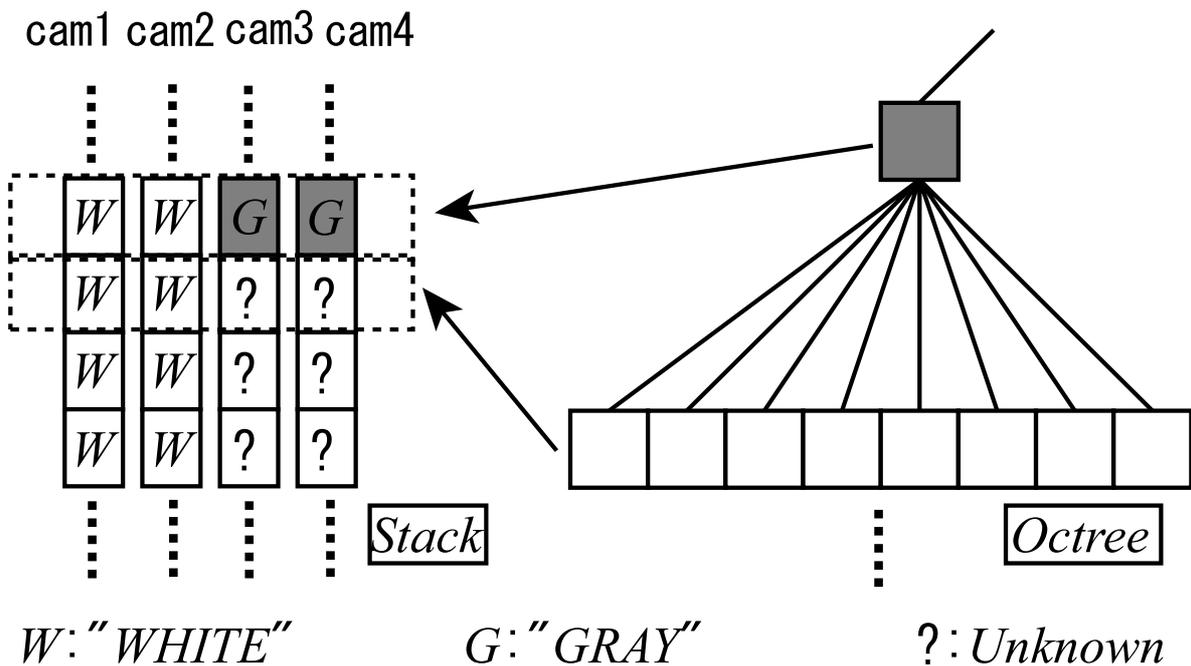


図 4.5: スタックの参照

## 4.2.5 内部 Voxel の削除

ここまでのステップで Octree 構造によって物体形状を得ることができる。しかしながら大小様々な大きさの立方体による表現は、後に行われる任意視点画像生成のための処理にとって不都合である。そこで、ここまでのステップで得られた Octree 構造を最終的な任意視点画像生成手法へ適用するため、図 4.6 に示すように Octree 構造で表現されている大きさが異なる様々な立方体の集合から最小単位の大きさの Voxel モデルへと変換し、さらに任意視点画像生成に不必要なモデル内部の Voxel 削除を行う。各 Octree の葉は  $2^{3n}$  個分の Voxel サイズと同値の立方体 1 つを表しているため、各立方体から Voxel の集合への変換は容易に行うことができる。また、Octree 構造のままでは連結関係の特定が難しいため困難な物体内部判定も、Voxel モデルに変換することで容易に行うことができる。

内部 Voxel の削除には 6 近傍の Voxel を参照することにより行う。対象となる Voxel の 6 近傍に 1 つでも背景を表す Voxel が存在する場合、その Voxel は表面 Voxel とする。逆に 6 近傍の Voxel が全て物体を表す Voxel の場合は、その Voxel は内部 Voxel とし任意視点画像生成には用いない。

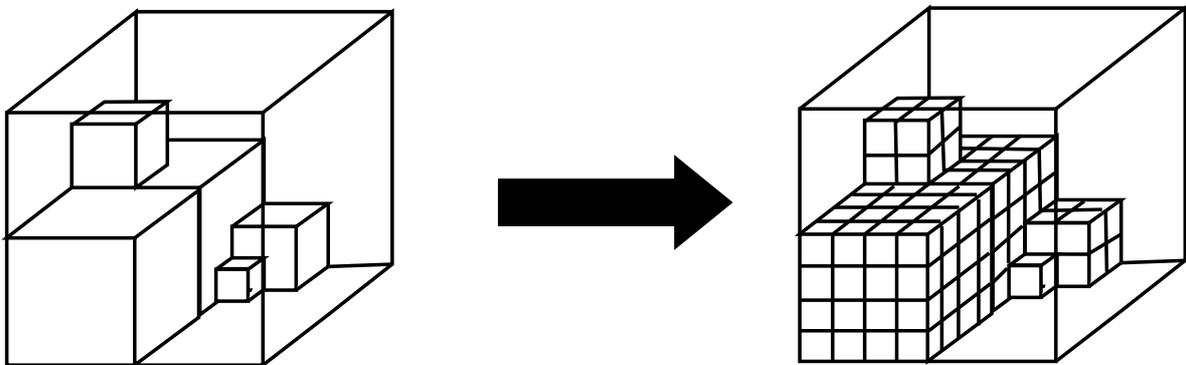


図 4.6: 内部 Voxel の削除

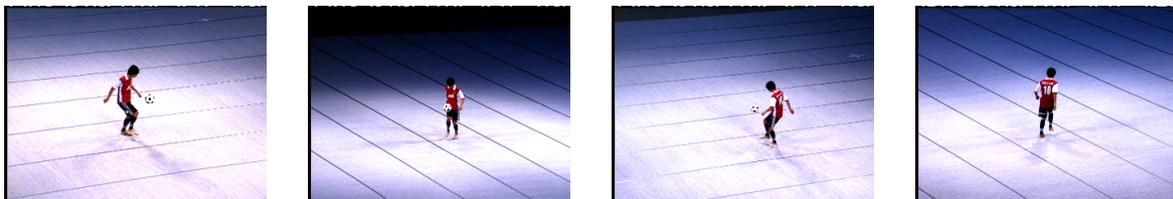
## 4.3 評価実験

ここまでで述べた物体形状復元手法の有効性を示すために行った評価実験について説明する。

本手法による物体形状復元にかかる計算時間と、全ての Voxel を調べて物体形状を復元するのにかかる計算時間との比較を行った。実験条件は以下の通りである。図 4.7 には実験に用いたカメラ画像の一部を示す。

- 画像解像度：320 × 240
- 視点数：4
- シーン数：10
- Voxel 解像度：256 × 256 × 256
- Octree 最大深度：8
- CPU：Intel Pentium3 1.0GHz

なお、Octree の深さが 1 の時は 1 つの立方体の 1 辺は 256Voxel に相当し、深さが 8 の時は立方体の 1 辺は 1Voxel に相当する。図 4.8 に復元結果の図の一部を示す。また、表 4.1 には各シーン復元時の立方体のサイズとその立方体の個数を示す。



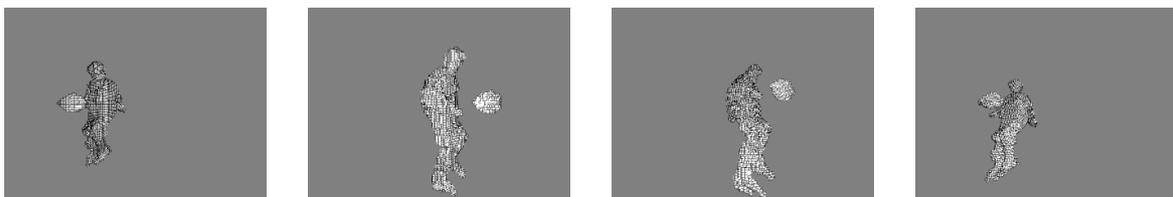
(a) 視点 1

(b) 視点 2

(c) 視点 3

(d) 視点 4

図 4.7: 評価実験に用いたカメラ画像の一部



(a) 視点 1

(b) 視点 2

(c) 視点 3

(d) 視点 4

図 4.8: 形状復元結果の一部

表 4.1: 立方体サイズとその数

シーン	1 voxel	2 <sup>3</sup> voxels	4 <sup>3</sup> voxels	8 <sup>3</sup> voxels	16 <sup>3</sup> voxels	計
1	6403	411	9	0	0	10267
2	6472	356	18	0	0	10472
3	6184	365	18	0	0	10256
4	6346	398	11	0	0	10234
5	6863	368	15	0	0	10767
6	6600	293	19	0	0	10160
7	6695	326	17	0	0	10391
8	7139	379	17	0	0	11259
9	7697	429	19	0	0	12345
10	7864	387	24	0	0	12496

物体形状復元にかかった時間は本手法では平均で 0.121 秒であったのに対し、全ての Voxel を調べる方法では 5.372 秒と約 42 倍もの差となった。この理由はまず、本手法が大きな立方体の時点で物体かどうかの判定を終えることができたためである。表 4.1 を見ると 4<sup>3</sup>Voxel 分の立方体の段階で処理が終わったケースがあったことが確認できる。つまり、単純に考えても 4<sup>3</sup> 回の計算を省略したということが言える。また子の代における復元処理を省くことができたことも計算量削減理由の一つである。表 4.1 から 4<sup>3</sup>Voxel 分の立方体の段階で立方体の状態が *WHITE* と判断できている場合があったことが確認できるが、この時はさらに大きな立方体の段階で一時状態に *WHITE* があったと推測できる。よって、一時状態に *WHITE* が出現した段階から子の代における計算を省略できているため、計算量を削減できたといえる。

この実験での測定時間には画像の読み込みや物体領域抽出処理、そして内部 Voxel 削除処理を含んでいない。よって本手法を用いて任意視点画像を生成した場合、フレームレートは最大でも毎秒 8 フレーム程度であると考えられる。毎秒 30 フレーム以上の出力を実時間処理とすると、この計算速度は実時間画像生成を実現できていない。しかし全 Voxel を評価する手法と比較すると 42 倍の高速化に成功しており、将来的に計算機性能向上した際、全 Voxel を評価する手法では実時間処理が難しい環境でも本手法では実時間処理を実現できることになる。このように計算量削減を実現できたという面において本手法は有効であるといえる。

## 第5章

### 画像情報からの座標系定義

## 5.1 はじめに

本章では第 1.2.1 節にて述べたカメラキャリブレーションの煩雑さを解決し、かつ PGS よりもユークリッド空間に近い座標系を画像情報からだけで定義する手法を提案する。本手法では焦点距離をほぼ無限大にした 2 台の正射影とみなせるカメラを用意し、これらのカメラ光線を空間の軸とする擬似正射影グリッド空間 (Orthogonal Projective Grid Space, OPGS) を定義する。カメラ光線の広がり小さいため、従来の PGS よりもユークリッド空間に近い座標系を持つ 3 次元空間を定義することができる。つまり、本手法によって従来はカメラキャリブレーションで必要だった実世界における 3 次元位置の実測が不要となり、また PGS にて生じていた空間の歪みを本手法によって改善することができる。しかし、本手法ではユークリッド空間と同等の直交座標系を定義することはできず、定義された仮想空間には若干の歪みが生じてしまう。そこで、物体形状に生じる歪みを出力画像にて吸収するために、第 1.3 節にて述べた任意視点画像生成手法の 1 つである Microfacet Billboarding 法を本手法によって定義される空間へ適用する。

はじめに、OPGS の定義方法について述べる。次にそこで生じる物体形状の歪みを吸収し、任意視点画像生成結果を向上するために取り組んだ Microfacet Billboarding 法の適用について説明する。Microfacet Billboarding 法の適用に関して、処理速度よりも誤差のある環境下でよりよい精度の結果画像を作ることに重点を置いている。

## 5.2 擬似正射影グリッド空間

では,OPGS を定義するための方法について説明する. 図 5.1 に空間定義時のカメラ配置例を真上から見た図を載せる. まず,OPGS を定義するために正射影とみなす 2 台カメラを別途用意する. このカメラを基底カメラと呼び,このカメラから発するカメラ光線によって OPGS 座標系を定義する. そして図に示すようにこれら 2 台の基底カメラから発するカメラ中心軸同士が約  $90^\circ$  で交わるように配置する. ただし完全に  $90^\circ$  の位置に配置することは困難であるためおおよそでかまわない. また,基底カメラは正射影とみなせるほど画角が非常に狭くなっているため,なるべく広い対象範囲を撮影できるように対象となる空間から離れた位置に配置する. 次に対象物体を撮影するためのカメラ(以下,「撮影用カメラ」と呼ぶ.)を配置する. これらのカメラ配置に制限はなく,任意の位置へ置いてかまわない. 全てのカメラを配置し終わったら基底カメラと撮影用カメラ,全てのカメラを用いて座標系定義のための撮影を行う. OPGS の定義には基底カメラから発するカ

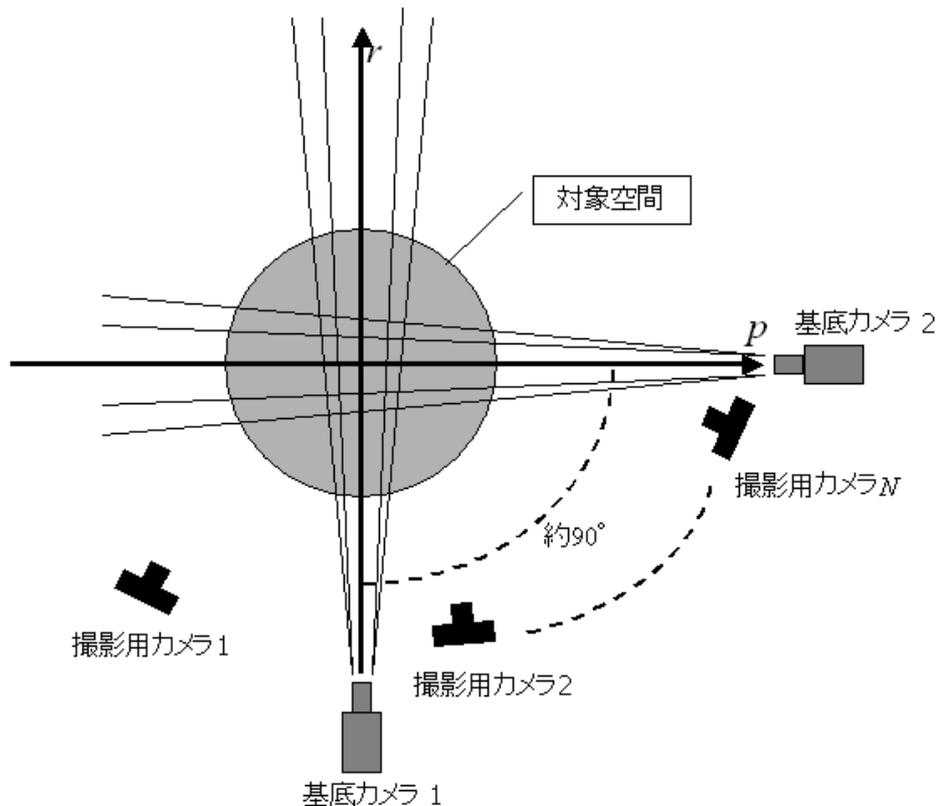


図 5.1: OPGS 定義時のカメラ配置例

## 5.2 擬似正射影グリッド空間

メラ光線が他の基底カメラ画像や撮影用カメラ画像へ投影されるエピポーラ線が必要である．そのため，基底カメラで撮影した画像と撮影用カメラで撮影した画像間の対応点群を求め，それら対応点群から Fundamental Matrix を算出する．本手法では Fundamental Matrix を Zhang[49] の手法によって算出している．OPGS では基底カメラ 1 内を通るカメラ光線の画像内での座標からユークリッド座標系での  $x$  軸， $y$  軸に相当する  $p$  軸， $q$  軸が定義され，基底カメラ 2 へ投影されたエピポーラ線によってユークリッド座標系での  $z$  軸に相当する  $r$  軸が決定される．なお，本手法では基底カメラは座標系を定義するためだけに必要なカメラであるため，対象物体の撮影には用いない．あくまで空間定義時のみに撮影を行い，一度 OPGS 座標系を定義してしまえば，それ以降はまったく使用しない．

次に，OPGS 内の座標と撮影用カメラ画像内の座標との対応関係を求める方法について説明する．OPGS 内のある 1 点  $P(p, q, r)$  は基底カメラで撮影した画像内の座標  $p_1(p, q)$  と  $p_1$  を通るカメラ光線が基底カメラ 2 で撮影した画像内に投影されたエピポーラ線上の 1 点  $p_2(r, y)$  の 2 点によって決定される． $y$  は式 (5.1) に示すエピポーラ線の方程式  $l_0$  によって求められる． $F_{12}$  は基底カメラ 1 から基底カメラ 2 への Fundamental Matrix である．

$$l_0 = F_{12} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} \quad (5.1)$$

そして，基底カメラ 1 内の点  $p_1(p, q)$  を通るカメラ光線はエピポーラ線  $l_1$  としてカメラ  $i$  に式 (5.2) によって投影される．

$$l_1 = F_{1i} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix} \quad (5.2)$$

ここで  $F_{1i}$  は基底カメラ 1 からカメラ  $i$  への Fundamental Matrix である．同様に基底カメラ 2 内の点を通るカメラ光線は式 (5.3) に示すように Fundamental Matrix,  $F_{2i}$  によってカメラ  $i$  へエピポーラ線として投影される．

$$l_2 = F_{2i} \begin{bmatrix} r \\ y \\ 1 \end{bmatrix} \quad (5.3)$$

この  $l_1$  と  $l_2$  の交点を求めることで OPGS 内の 1 点  $P$  とカメラ  $i$  内の 1 点の対応関係を求める．これを基底カメラ以外の全カメラ画像間で行うことで OPGS 内の 1 点と各カメラ画像間の対応関係を求めることができる．図 5.2 にその様子を示す．

## 5.2 擬似正射影グリッド空間

ここまでで述べたように，OPGS とは正射影とみなせる 2 台の基底カメラを用いて定義された PGS であるといえる．第 1.2.1 節にて説明した PGS は 2 台の基底カメラから発するカメラ光線を用いて定義した空間全般を指し，PGS の定義に際して基底カメラの選択選択には一切の制限はなく，任意に選んだ 2 台のカメラを基底カメラとして用いることができる．またカメラの画角に関する制限もなく，どのような画角のカメラを用いてもよい．実際，前述した斎藤らの手法では撮影に使用したカメラを基底カメラとして利用している．一方，OPGS では基底カメラの配置，画角に制限を加えている．可能な限り正射影として扱えるカメラを準備し，それを基底カメラとして用いなくてはならない．また配置に関しても，2 台の基底カメラのカメラ中心を通る光線が互いに  $90^\circ$  で交わるような位置関係にしなくてはならない．さらに，仮想空間を可能な限り大きく定義するためにカメラの位置を対象範囲からなるべく離して配置しなくてはならない．つまり，OPGS とは PGS に含まれる特殊な一部の形態であるといえる．しかしこの制限によって，基底カメラによって定義された空間はユークリッド空間のような直交座標系に限りなく近づき，かつ仮想 3 次元空間とカメラ画像間の対応関係を撮影した画像からだけで求めることが可能である．この利点はカメラへ制限を加える際に生じる不利益に比べ，非常に大きいものであるといえる．

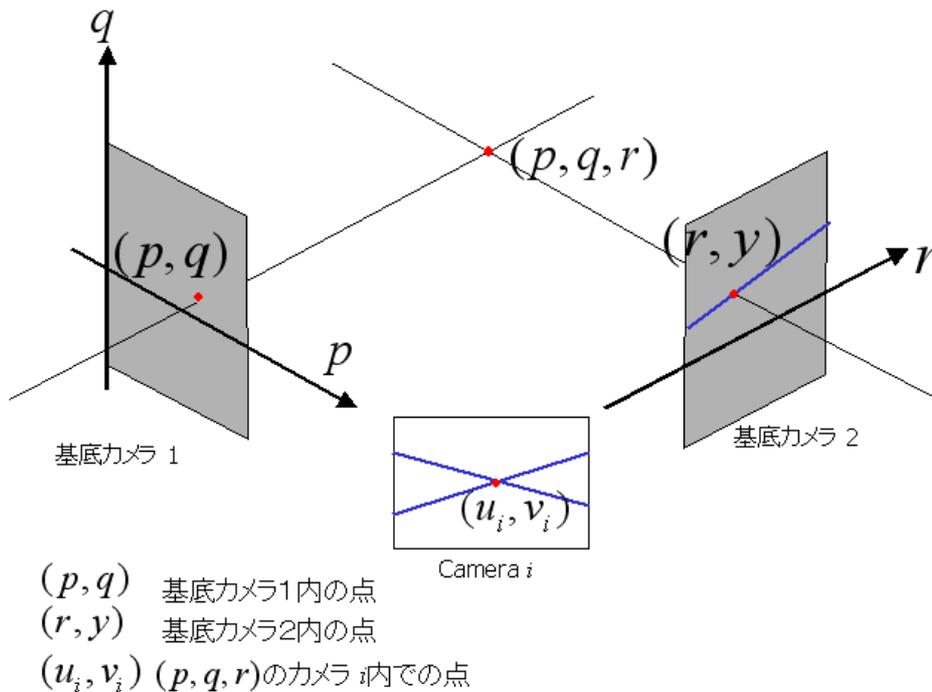


図 5.2: OPGS とカメラ画像との対応関係

## 5.2 擬似正射影グリッド空間

一般的には実世界の3次元空間内の点とそれが投影された2次元画像平面内の点との既知の対応点を6点以上使って各カメラの外部・内部パラメータを算出することで透視投影行列を求め、3次元空間と2次元平面間の1対1の対応関係を求める。このカメラキャリブレーションの作業は実世界の多数の点を実測しなくてはならないため、非常に煩雑な作業となってしまう。特に大規模な環境ではそれが顕著である。しかし本手法で定義されるOPGSでは3次元空間の定義に必要なのはエピポーラ線のみのためカメラ画像間の対応点のみを求めればよい。撮影環境の規模によらず、仮想3次元空間と画像平面間の対応関係を求める手法は変わらない。さらに、本手法では正射影とみなせるカメラの光線によって仮想空間を定義するため、PGSよりもユークリッド空間により近い座標系を持つ空間としてOPGSを定義することができる。

限りなくユークリッド空間のような直交座標系に近いOPGSではあるが、実在するカメラに対して焦点距離を無限大にすることは不可能であるため、実際に得られたOPGSの座標系にはカメラ幾何特性に応じた歪みが生じ、ユークリッド空間の座標系との誤差が生じてしまう。このOPGSで生じてしまうユークリッド座標系の違いと、従来の射影グリッド空間での生じる座標系の違いの差についての比較実験、およびその検討については第5.4.1節、および第5.4.2節にて詳しく述べるものとする。

## 5.3 着色方法

### 5.3.1 Voxel への着色

任意視点画像を生成する際、Voxel へ直接着色をして、着色済みの Voxel の集合として復元されたモデル形状を表示する方法が考えられる。まず、この方法について本節で述べる。この方法は対応する画素の色を取得するだけでよいため、計算量があまりかからないというメリットがある。

本手法では Voxel に割り当てる色を、仮想視点の位置に応じて最も近い 2 台の撮影用カメラ画像から取得した 2 つの色をブレンドすることによって求める。図 5.3 にその様子を示し、式 (5.4) に着色時の計算式を示す。

$$c(\mathbf{p}) = \frac{\phi}{\theta + \phi} c(\mathbf{p}_i) + \frac{\theta}{\theta + \phi} c(\mathbf{p}_{i+1}) \quad (5.4)$$

今、仮想視点に最も近い 2 台のカメラをカメラ  $i$ 、カメラ  $i+1$  とする。そして  $\theta$ 、 $\phi$  はそれぞれ仮想視点とカメラ  $i$ 、仮想視点とカメラ  $i+1$  との水平方向のなす角である。ブレンドの重み付けはこれらの角度によって決定する。ある Voxel の座標  $\mathbf{P}$  が決定すると、カメラ  $i$  とカメラ  $i+1$  内でその Voxel と対応する点  $\mathbf{p}_i$ 、 $\mathbf{p}_{i+1}$  が OPGS 座標と撮影用カメラ座標との対応関係から求めることができる。そしてこれら 2 点のピクセルの色  $c(\mathbf{p}_i)$  と  $c(\mathbf{p}_{i+1})$  に対して式 5.4 を適用することで対象となる Voxel の色が決定される。

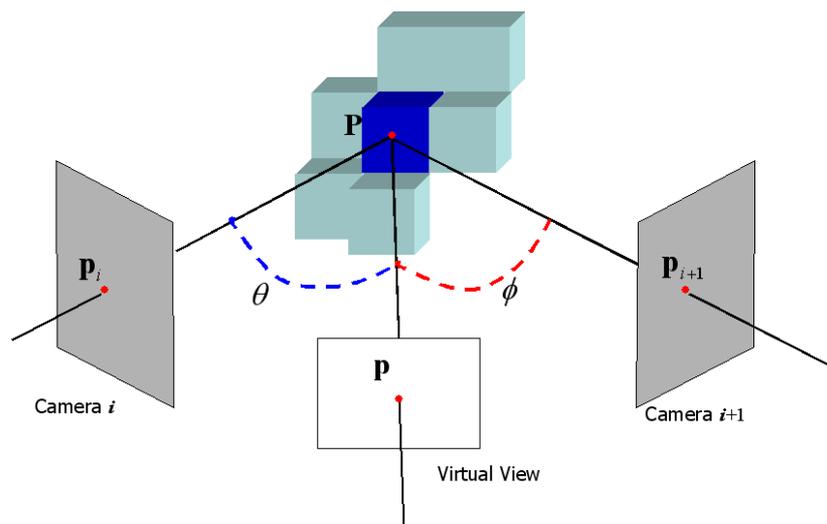


図 5.3: Voxel への着色

## 5.3.2 Microfacet Billboarding 法の適用

次に，OPGS への Microfacet Billboarding 法の適用について述べる．Microfacet Billboarding 法とは，点群によって復元されたモデルの各点に，仮想視点に対して常に垂直に位置する微小面を配置し，その微小面に対応するテクスチャをカメラ画像から取得し，マッピングしてレンダリングを行う手法である．この手法の最大の特徴は，ある1つの微小面上でのテクスチャの連続性は常に保たれるため，仮想空間3次元座標と，実カメラ画像平面座標間の対応関係やモデル形状に誤差が生じている環境においても，結果画像の表示精度を高く保てることである．よって，本手法で定義したOPGS内で復元された若干の歪みをもつ物体形状を用いて任意視点画像を生成する手法に適しているため，Microfacet Billboarding 法をレンダリング手法として取り入れることとした．

まず，Voxel モデルの各表面 Voxel の位置に仮想視点に対して常に垂直に位置する微小面を配置する．ここでの垂直とは仮想視点位置からの視線ベクトル  $v$  と，微小面の法線ベクトル  $n$  の水平方向のなす角が0になることを意味する．そして，微小面のサイズは全ての視点位置に対して Voxel 1つをカバーするように，Voxel サイズの  $\sqrt{3}$  倍とする．図 5.4 にその様子を示す．

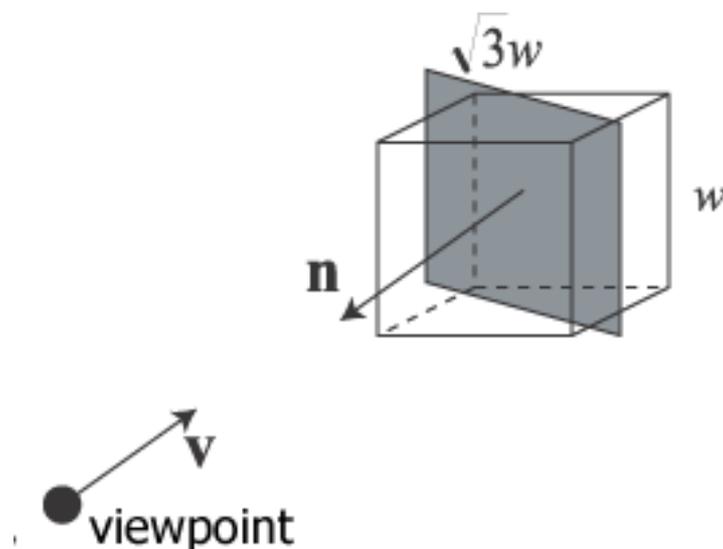


図 5.4: 微小面と視点・視線方向の関係

### 5.3 着色方法

次に、この微小面に実カメラ画像から取得したテクスチャをマッピングする。この時、微小面の法線と微小面と OPGS 内での実カメラ位置とのなす角を求め、このなす角が最小となるカメラを選択する。使用するカメラ画像が決定したら、微小面の4頂点を OPGS と各画像平面間の対応関係から、画像内での座標に変換し、微小面へマッピングするテクスチャ領域を決定する。図 5.5 にテクスチャマッピング決定の例を示す。図 5.5 内の Microfacet  $i$  で示される微小面の法線  $n_i$  とカメラ中心座標  $O_1$  と  $O_2$  のなす角、 $\theta_1$  と  $\theta_2$  とでは  $\theta_1$  の方が小さい。よってこの微小面へマッピングするテクスチャは Image1 から取得する。一方、Microfacet  $j$  で示される微小面の法線  $n_j$  とカメラ中心座標  $O_1$  と  $O_2$  のなす角、 $\phi_1$  と  $\phi_2$  とでは  $\phi_2$  の方が小さい。そのため、微小面  $j$  へマッピングするテクスチャは Image2 から取得することになる。なお、カメラ画像内でシルエット外の領域は全て透明のテクスチャとしておく。これにより、対象物体領域外の画像領域が微小面にマッピングされることを防ぐことができる。

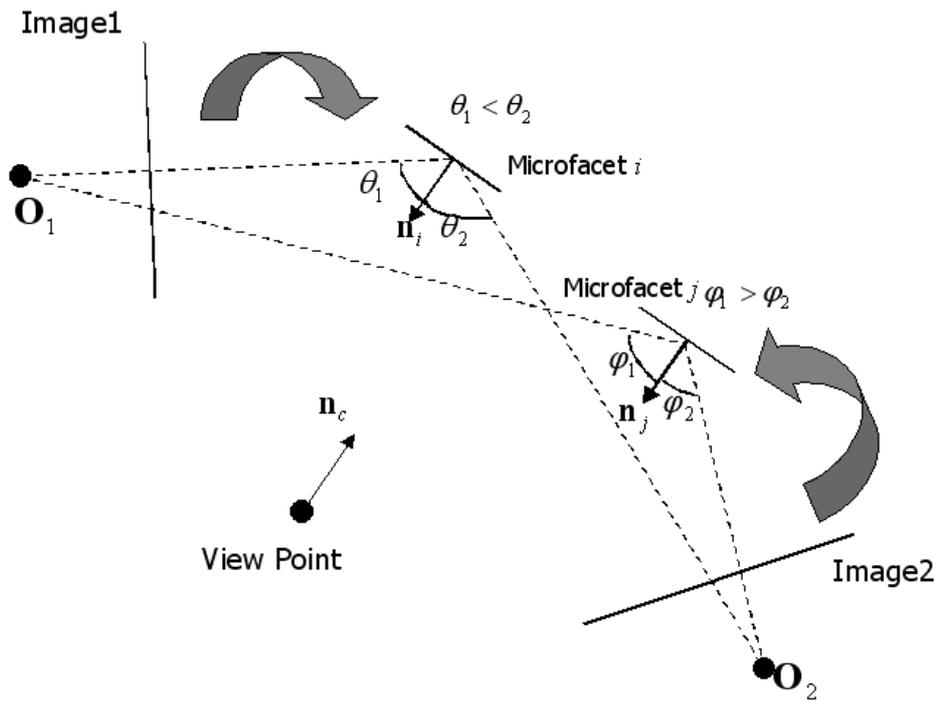


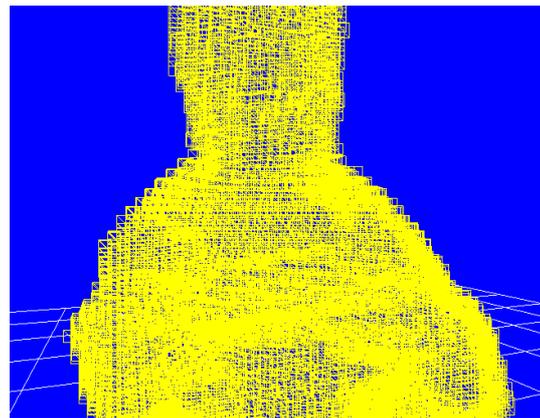
図 5.5: 微小面へのテクスチャマッピング

図 5.6 に Voxel モデルへ着色した場合と Voxel モデルから微小面の集合への変換，そしてそれにより得られるレンダリング結果を示す．

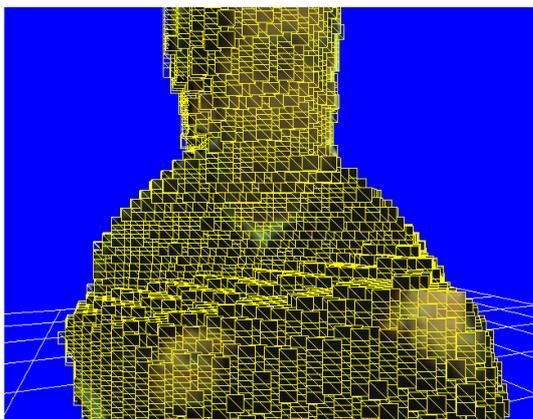
前述した，Voxel 1 つに対してブレンドした 1 色だけを割り当てる手法では，Voxel 1 つがカバーする領域が 1 色でしか表現できないため，結果画像がぼやけた印象を与えてしまう結果となっている．さらに隣り合う Voxel への着色結果に誤差が生じた場合，実カメラ画像内での隣り合う位置での画像領域との差が大きくなってしまう．しかしこの手法では，仮想空間内での Voxel 1 つ分の大きさの領域をそれと対応する大きさのテクスチャによって表現することができるため，多少の誤差が生じても微小面内での画素の連続性は保たれることになる．



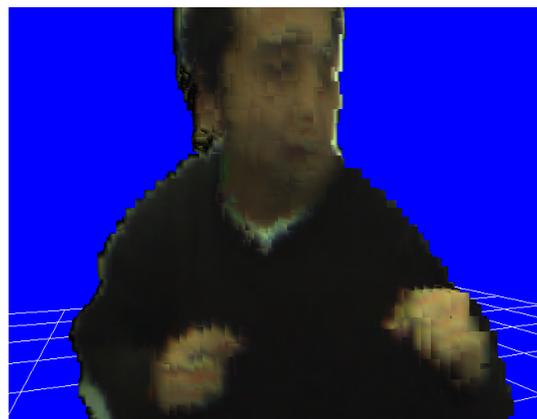
(a) Voxel モデル



(b) 微小面集合



(c) 微小面集合へのテクスチャマッピング



(d) レンダリング結果

図 5.6: Voxel モデルから微小面の集合への変換

## 5.4 評価実験

本節では、本手法の有効性を示すために行った2つの実験について述べる。

### 5.4.1 実験1：復元された物体形状の視覚的評価

まず、本手法によって定義される OPGS，そして PGS とユークリッド空間の3つの空間において物体形状を復元してそのモデル形状を見た目で評価する。実験時の条件を以下に示し、図 5.7 には撮影環境を示す。正射影とみなせるカメラは画角が約  $1^\circ$ ，撮影用カメラの画角は約  $90^\circ$  である。また図 5.8 に物体形状比較評価に用いたカメラ画像を示す。

- 画像サイズ：320 × 240
- 撮影カメラ数：4
- Voxel 解像度：256 × 256 × 256
- 基底カメラ
  - OPGS：正射影とみなせる座標系定義用カメラを基底カメラとして使用
  - PGS：カメラ1，カメラ3を基底カメラとして使用
  - ユークリッド空間：なし（6つの対応点から各画像への射影変換行列を算出）

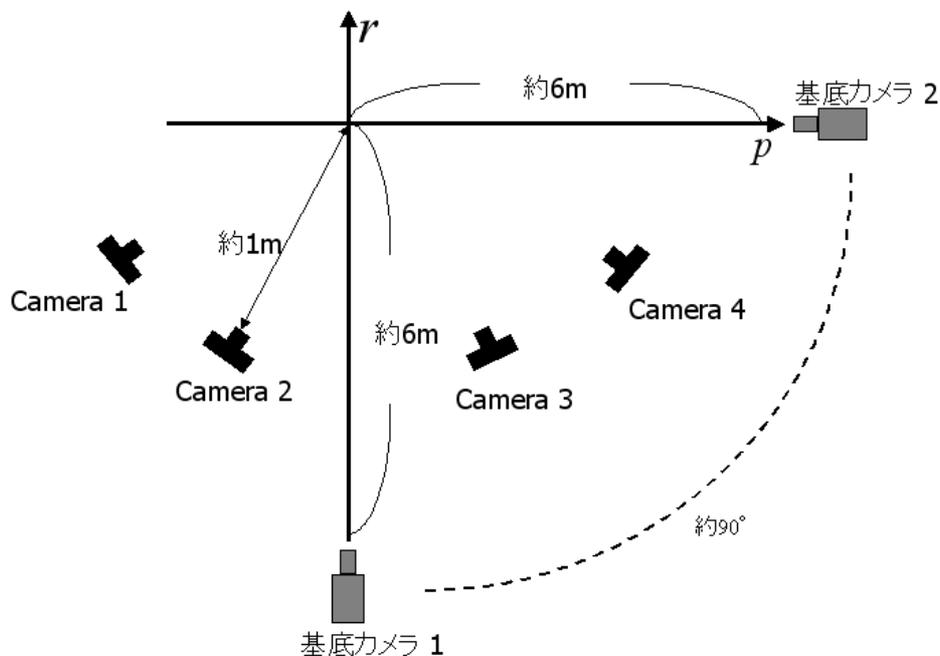


図 5.7: 撮影環境

## 5.4 評価実験



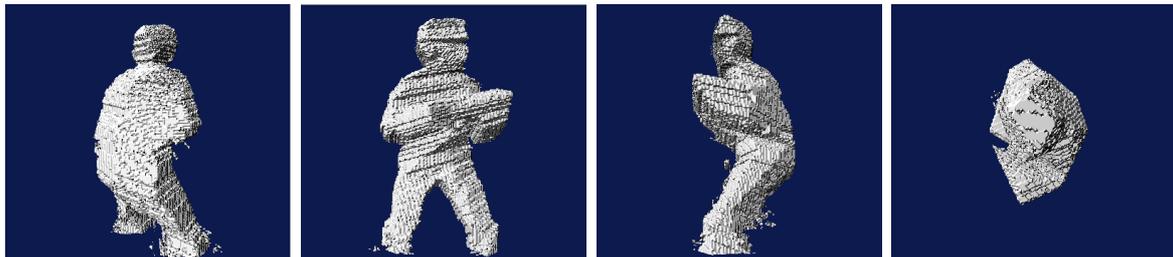
(a) カメラ 1

(b) カメラ 2

(c) カメラ 3

(d) カメラ 4

図 5.8: モデル形状比較に用いたカメラ画像



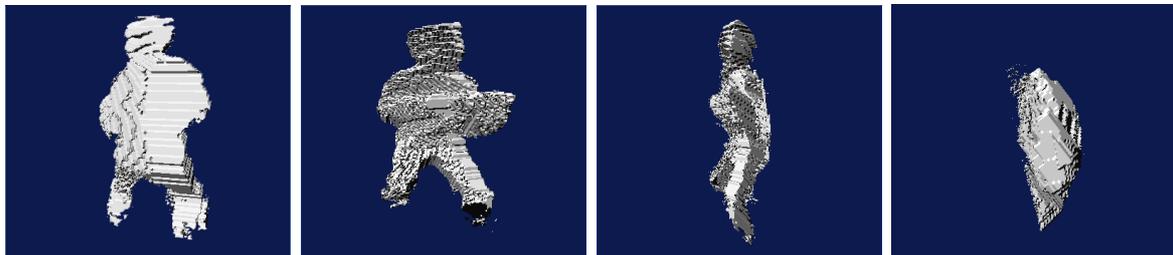
(a) 視点 1

(b) 視点 2

(c) 視点 3

(d) 視点 4

図 5.9: 擬似正射影グリッド空間内での復元モデル形状



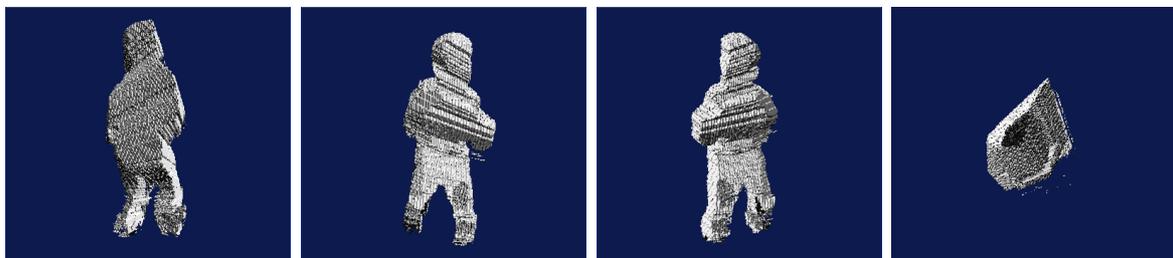
(a) 視点 1

(b) 視点 2

(c) 視点 3

(d) 視点 4

図 5.10: 射影グリッド空間内での復元モデル形状



(a) 視点 1

(b) 視点 2

(c) 視点 3

(d) 視点 4

図 5.11: ユークリッド空間内での復元モデル形状

まず OPGS 内で復元されたモデルを図 5.9 に示す．この図に示すモデルは，Voxel 形状から MC 法 [27] によってポリゴンモデルに変換したものである．前述したように，空間の定義には復元に用いたカメラとは別に用意した正射影とみなせるカメラだけを用いている．次に，図 5.8 のうちカメラ 1，カメラ 3 を基底カメラとして用いて定義した PGS にて対象物体形状を復元したモデルを図 5.10 に示す．このモデルも同様にポリゴンモデルで表現されている．このモデルを復元する際に座標系定義の目的で用いたカメラ 1，カメラ 3 は焦点についての設定は特に行っておらず，一般的な透視投影モデルに基づくカメラとなっている．そのため，このカメラから発するカメラ光線は OPGS 定義時に用いたカメラよりも広がり具合が大きくなっている．さらに比較のために，フルキャリブレーションを行ってユークリッド空間内で対象物体を復元したモデルを図 5.11 に示す．カメラキャリブレーションは実世界での 6 点を実測し，その対応点をカメラ画像内にて測定して行った．前述した 3 つのモデルはすべて同じ対象物体を同じカメラ画像で復元している．復元に用いた画像の数もすべて同じである．そして各図で示すモデル形状は，それぞれの空間で復元したモデルのユークリッド空間における形状である．

まず，図 5.9 の OPGS での復元モデルと図 5.10 に示す PGS のモデル形状を比較すると，OPGS でのモデルの方が PGS のモデルに比べて形状がより自然に見えることが確認できる．特に，図 5.9(d) と図 5.10(d) に示される真上からの視点の画像で比べると人の体の厚みが図 5.10(d) では不自然であることが確認できるが，図 5.9(d) ではそれがより自然な厚みに見える．同様に，図 5.11 のユークリッド空間内で復元したモデルと OPGS 内でのモデルを比較しても物体形状が不自然には見えないことがわかる．この実験より，視覚的には物体形状の歪みはほとんど確認できないといえる．

### 5.4.2 実験 2 : 座標系の歪みの比較

次に、OPGS と PGS での座標系の歪み具合の比較を行った。今回の実験では OPGS と PGS の歪みの度合いを評価する指標としてエピポーラ線の広がり具合を用いた。エピポーラ幾何の性質によりカメラから発するカメラ光線は放射状に広がっているが、理想的な正射影カメラから発するエピポーラ線はすべて平行となる。そして OPGS も含んだ広い意味での PGS は基底カメラ間で投影されるエピポーラ線によって定義されるため、エピポーラ線の広がり具合が PGS の歪みに反映される。よって、基底カメラに投影されるエピポーラ線同士の平行性について調べることで PGS の歪み具合を評価することができる。評価実験ではエピポーラ線同士の平行性の尺度としてそれぞれの直線の垂直方向の差を用いた。つまり、この値を PGS 座標系のユークリッド空間座標系に対する誤差とする。そして PGS 内の  $p, q, r$  軸それぞれの誤差は、各軸方向に隣接するエピポーラ線同士の垂直方向の差から求めた。この誤差を OPGS, PGS それぞれに対して求め、OPGS と PGS の歪みの比較を行った。各軸での評価方法の詳細を以下に述べる。

#### 5.4.2.1 $p$ 軸の評価

測定方法の概要を図 5.12 に示す。この図に示すように、まず基底カメラ 1 内での 1 点  $(p, q)$  と  $p$  軸方向に隣接する  $(p+1, q)$  を通るカメラ光線を基底カメラ 2 へ投影する。すると、基底カメラ 2 内に 2 本のエピポーラ線が投影される。次に、基底カメラ 2 内でこれらエピポーラ線上の 1 点  $(r, y1)$ ,  $(r, y2)$  をそれぞれ指定する。 $y1, y2$  はエピポーラ線の方程式によって求められる値である。ここで投影された 2 本のエピポーラ線同士の垂直方

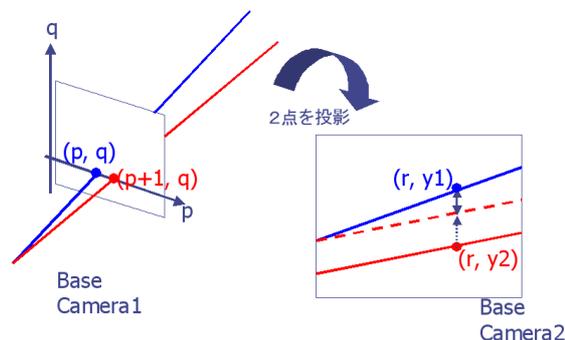


図 5.12: PGS の歪み具合の評価 ( $p$  軸)

向の差  $|y_1 - y_2 - b|$  ( $b$  は点  $(p + 1, q)$  を投影したエピポーラ線の  $y$  切片) を求め、そしてこれを PGS 内の 1 点  $(p, q, r)$  での  $p$  軸方向の誤差とする。

#### 5.4.2.2 $q$ 軸の評価

$q$  軸方向の測定方法の概要を図 5.13 に示す。 $p$  軸方向の時と同様に  $q$  軸方向に隣接する 2 点  $(p, q)$ 、 $(p, q + 1)$  のエピポーラ線を基底カメラ 2 へ投影し、これらの垂直方向の差を  $(r, y_1)$ 、 $(r, y_2)$  において求める。そして求めた垂直方向の差を射影グリッド空間内の 1 点  $(p, q, r)$  での  $q$  軸方向の誤差とする。

#### 5.4.2.3 $r$ 軸の評価

$r$  軸方向の測定方法は、図 5.14 に示すようにまず基底カメラ内での 1 点  $(p, q)$  を通るカメラ光線を基底カメラ 2 へ投影する。次に、基底カメラ 2 内でのエピポーラ線上の  $r$  軸方向に隣接する 2 点  $(r, y_1)$ 、 $(r + 1, y_2)$  を求め、これらの点を通るカメラ光線を基底カメラ 1 へ再投影する。そして、基底カメラ 1 内での 2 本のエピポーラ線上の点  $(p, y_3)$ 、 $(p, y_4)$  を求め、この点での垂直方向の差を求める。求めた垂直方向の差を PGS 内の 1 点  $(p, q, r)$  での  $r$  軸方向の誤差とする。

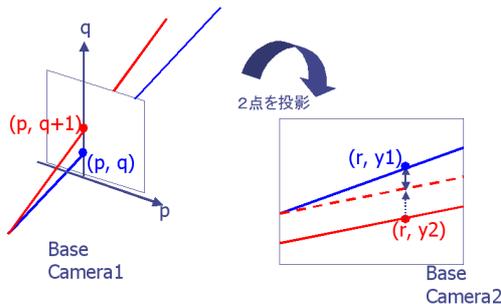


図 5.13: PGS の歪み具合の評価 ( $q$  軸)

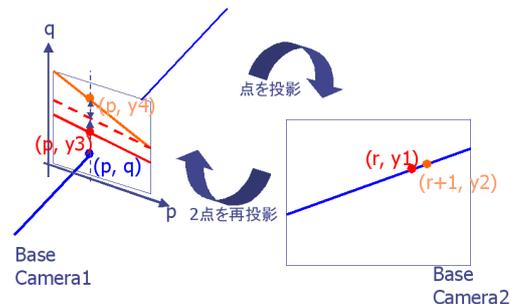


図 5.14: PGS の歪み具合の評価 ( $r$  軸)

## 5.4 評価実験

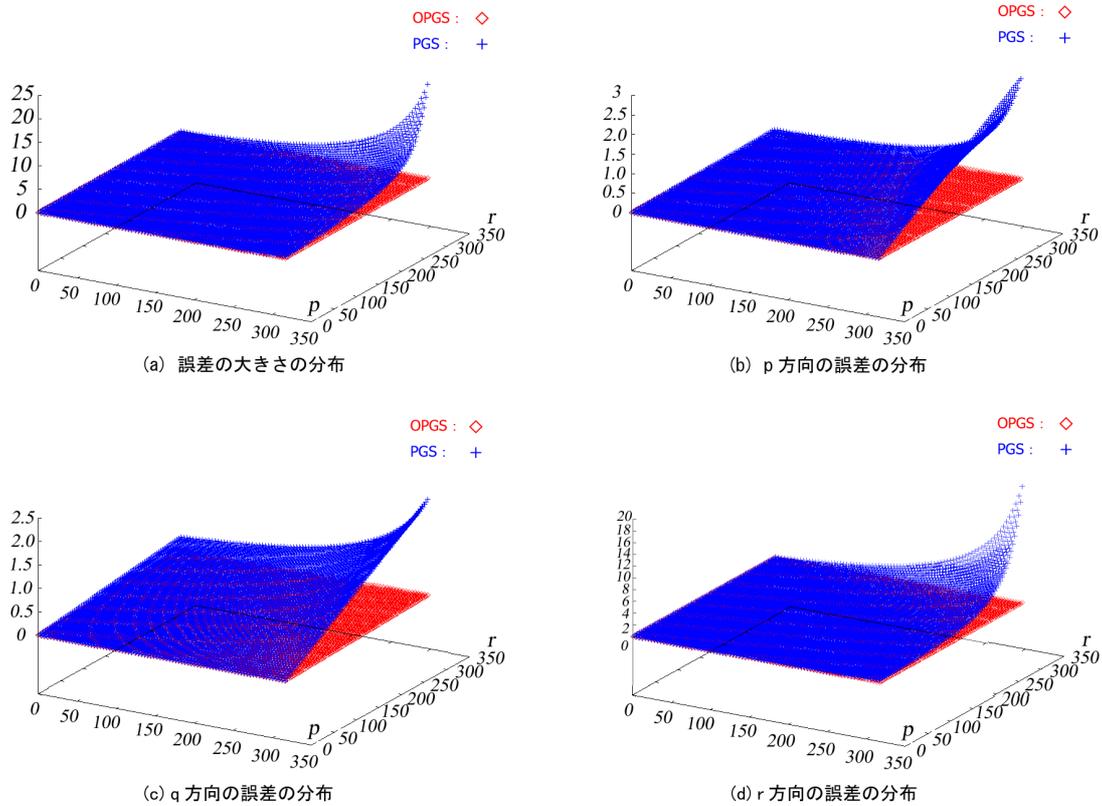


図 5.15: OPGS と PGS 内のある平面での誤差分布

### 5.4.2.4 評価実験

前述した歪み具合の度数を用いて OPGS, PGS それぞれの誤差の分布を同時に求めた。この評価実験も実験 1 と同じ条件で行った。ただし、この評価実験は OPGS と PGS に対して行っている。表 5.1 には各軸方向の誤差の平均値、表 5.2 にはその分散を同じく各軸方向について算出したものを示す。平均の単位はピクセル、分散の単位はピクセルの 2 乗である。さらに、図 5.15(a) から図 5.15(d) に OPGS, PGS 空間の  $q = 127$  の平面における全格子点の誤差をプロットした分布図を示す。この  $q = 127$  の平面とは、定義した仮想空間内の  $q$  軸方向の中間に位置する平面である。図 5.15(a) は各軸方向の誤差を 1 つのベクトルと考えたときの大きさの分布を、そして図 5.15(b) から図 5.15(d) はそれぞれ、 $p, q, r$  軸方向の分布を示している。分布の値の単位はピクセルである。

まず、表 5.1 に示す誤差の平均値から、PGS での誤差の方が OPGS の誤差よりもすべ

表 5.1: 各空間での直線の垂直方向の差の絶対値の平均 (pixel)

	$p$	$q$	$r$
OPGS	$2.81 \times 10^{-3}$	$6.48 \times 10^{-2}$	$3.30 \times 10^{-3}$
PGS	$2.88 \times 10^{-1}$	$4.24 \times 10^{-1}$	$8.89 \times 10^{-1}$

表 5.2: 各空間での直線の垂直方向の差の分散 (pixel<sup>2</sup>)

	$p$	$q$	$r$
OPGS	$3.70 \times 10^{-5}$	$1.75 \times 10^{-2}$	$5.30 \times 10^{-5}$
PGS	$3.04 \times 10^{-1}$	$8.46 \times 10^{-1}$	9.59

ての方向において約 10 倍以上も大きくなっていることが確認することができる。さらに、表 5.2 の各軸方向の分散の値からも OPGS の方が誤差のばらつきが小さいことがわかる。また、図 5.15(a) から図 5.15(d) の分布を見ても、図 5.9 に示したモデルの範囲だけでなく、空間内のある平面全体での誤差の分布についても OPGS の方が PGS よりも誤差が小さく、その分布も一様であることがわかる。図 5.15 の分布を見ると、 $p, r$  が大きくなればなるほど、つまり基底カメラから離れた空間位置にあればあるほど、PGS ではその誤差が大きくなり、空間にはより大きな誤差が生じていることが確認できる。しかし、OPGS の場合は基底カメラから離れてもその誤差の変化は小さいことがこの図から確認できる。OPGS の誤差の絶対量を見ても、表 5.1 より平均で 1 ピクセル以下と小さく、さらに表 5.2 の分散値から空間全体としてわずかな歪みのみが生じているだけということが確認できる。各軸方向の誤差分布についても、図 5.15(b) から図 5.15(b) を見ると同様のことが言えることが確認できる。これは本手法で提案した OPGS では空間全体の平均の誤差が 1 ピクセル以下であり、誤差の分布のばらつきも小さいため、空間全体としての誤差が小さく、結果として OPGS 内で復元したモデル形状をユークリッド空間内で表現しても両者の違いを小さく抑えることができていることを意味する。

### 5.4.3 2つの実験結果より

以上、2つの評価実験より次のことが言える。まず、OPGS 内で復元したモデルは見た目に関してはごく自然な形状を取るといえる。これは図 5.9 に示す OPGS 内で復元したモデルと図 5.11 を比較すれば明らかである。また、その歪みの度合いも表 5.1 から OPGS は PGS に比べて平均で 10 分の 1 から 100 分の 1 程度に抑えられていることがわ

かり，歪みの分布に関しても表 5.2 から 5 分の 1 から  $1/50000$  程度となっていることがわかる．つまり PGS は基底カメラから離れた位置であればあるほど歪みが大きくなっていくのに対し，OPGS では非常に小さな歪みが空間全体に一様に分布しているだけということがわかる．以上のことから OPGS はカメラキャリブレーションの代替手段として仮想空間を構築するのに有効であると言える．

## 第6章

### 提案手法による任意視点画像生成

## 6.1 はじめに

本章ではここまで提案した以下に示す3つの手法，

- 背景差分手法
- 高速物体形状復元手法
- 画像からの情報のみによる空間定義手法

を組み合わせた任意視点画像生成の手法と，この手法を用いて構築した共有仮想空間通信システムについて述べる．

任意視点画像生成手法は前述した3つの手法を組み合わせ，多視点画像の入力から任意視点画像の生成までを通して行う．本手法によって任意背景環境下においても影領域を取り除いた対象物体領域の抽出を行うことができる．また，提案した高速復元手法により，物体形状復元にかかる時間は短時間となっている．さらに仮想空間の座標系をカメラキャリブレーションを行わずとも定義することが可能である．なお，任意視点画像生成手法では Microfacet Billboarding 法を適用してよりよい精度の結果画像を作ることに重点を置いている．

さらにこの手法を用いた応用例として構築した共有仮想空間通信システムについて述べる．このシステムは離れた2地点における多視点カメラシステムによって撮影した多視点画像列から，2地点での対象物体を共有される同じ仮想空間内で描画し，あらかじめ作成しておいたCGモデルと合成するものである．本章では共有仮想空間通信に関連する研究についても説明し，その後システムについて述べる．また提案したシステムについての検討も行う．

## 6.2 提案する任意視点画像生成手法について

まず、提案する任意視点画像生成手法について説明する。図 6.1 に多視点カメラ画像の入力から任意視点画像の出力までの流れを示す。本手法では、はじめに正射影とみなせる 2 台の基底カメラと対象物体の撮影に用いる撮影用カメラで Fundamental Matrix 算出用の多視点画像列を撮影しておく。これらの画像を用いて各カメラ間の画像内での対応点群を求め、その点群からカメラ間の Fundamental Matrix を求める。Fundamental Matrix を算出したらこの行列によって OPGS 座標系を定義する。なお、基底カメラは座標系定義のためだけに用いているので、座標系定義の処理を終えたら基底カメラは使用しない。

OPGS 座標系を定義したら、カラー背景画像と視差背景画像を撮影しておく。これは背景差分に用いるための背景画像である。背景画像を撮影し終わったら対象物体を撮影した多視点画像を入力する。多視点画像はカラー画像と視差画像を用いる。また同時に入力す

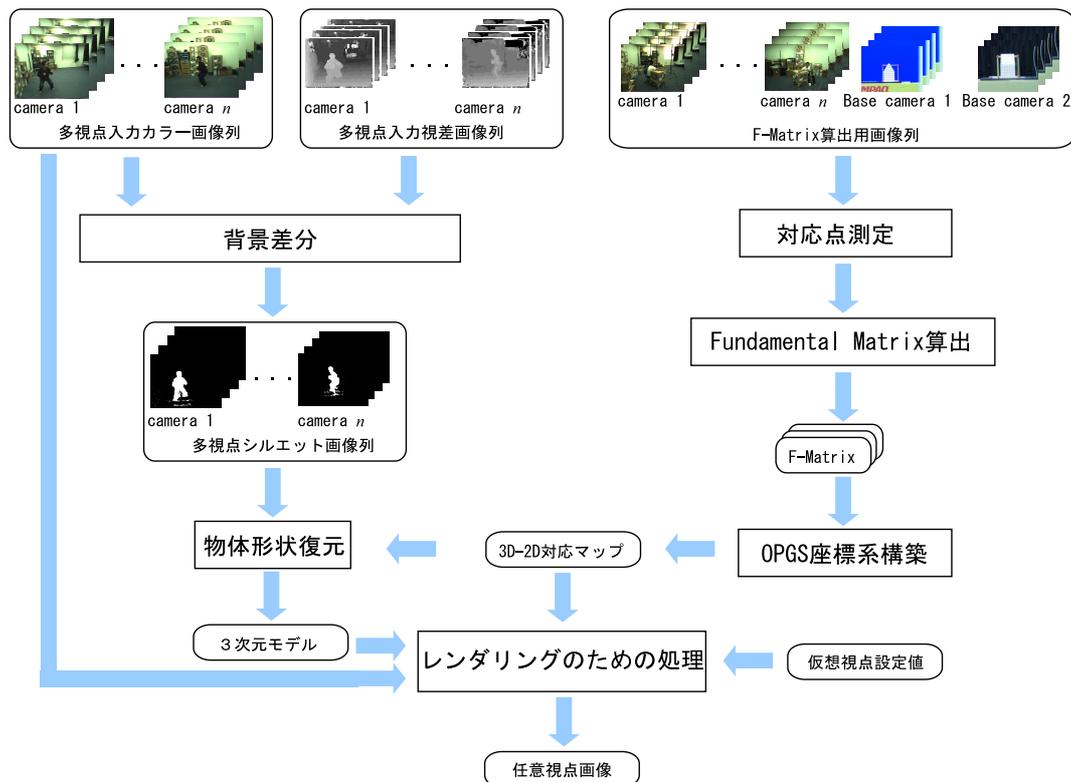


図 6.1: 任意視点画像生成手法の流れ

## 6.2 提案する任意視点画像生成手法について

るのは同期を取って撮影した同時刻の多視点カメラ画像である．これら画像を入力したらカラー背景画像，視差背景画像とカラー原画像，視差原画像から各視点における対象物体領域の抽出を背景差分によって行う．抽出された対象物体領域はシルエット画像として次の処理へと渡される．

各視点のシルエット画像を求め終えたら，これら多視点シルエット画像と事前に定義した OPGS 座標系と各カメラ画像座標への対応関係から対象物体形状の復元を行う．復元処理はまず Octree 形式で物体形状を求め，それを Voxel 形式へと変換する．その後，復元モデルの表示に不必要な物体形状内部に属する Voxel を全て取り除き，表面 Voxel のみで構成される物体形状を次のレンダリングへの処理へと渡す．

表面 Voxel モデル形状を取得したら，これら Voxel を用いて任意視点画像出力のための処理を行う．まず，Voxel へ直接着色する場合は，仮想空間内でのカメラ位置と仮想視点の位置関係から該当する色情報を2つのカメラ画像から取得し，その色をブレンドして対象とする Voxel へと着色する．また，Microfacet Billboard 法を適用する場合，前述したように仮想視点を挟む2台のカメラ画像のうち，適するカメラ画像を選択して該当領域のテクスチャを Microfacet へとマッピングする．

ここまでの処理を終えたら，仮想視点の位置や向いている方向をパラメータとした任意視点画像を出力する．このような流れで多視点カメラ画像からカメラキャリブレーションを行わずに任意視点画像を生成することができる．

## 6.3 任意視点画像生成実験

では、本手法を用いて任意視点画像を生成した実験について説明する．実験条件は以下の通りである．また、図 6.2 に撮影条件を示す．この撮影条件は第 5.4.1 節で行った評価実験と同じ条件である．

- カメラ数：4
- Voxel 数： $256 \times 256 \times 256$
- 画像解像度： $320 \times 240$  画素
- カラー画像の色深度：24bit RGB カラー画像 (R, G, B 各 8bit)
- 視差画像の色深度：8bit グレースケール画像

図 6.3 以降に実験に用いた多視点画像列と生成されたシルエット画像、本手法によって得られた任意視点画像をシーン毎に順に載せる．結果画像ページの左側には Microfacet Billboarding 法を用いて生成した自由視点画像を、また右側には比較のために Voxel に直接着色をして生成した結果画像を載せている．これらの結果画像から、実際のカメラ画像で撮影していない視点での画像が生成できていることがわかる．また、得られた画像での対象物体の形状は、結果画像を見る限りでは原画像での形状と近いことがわかる．さらに、Voxel に着色した場合、対象物体への着色が全体的にぼやけているのがわかるが、Microfacet Billboarding 法を使用することにより、より実画像に近い結果になっていることが視覚的に判断できる．

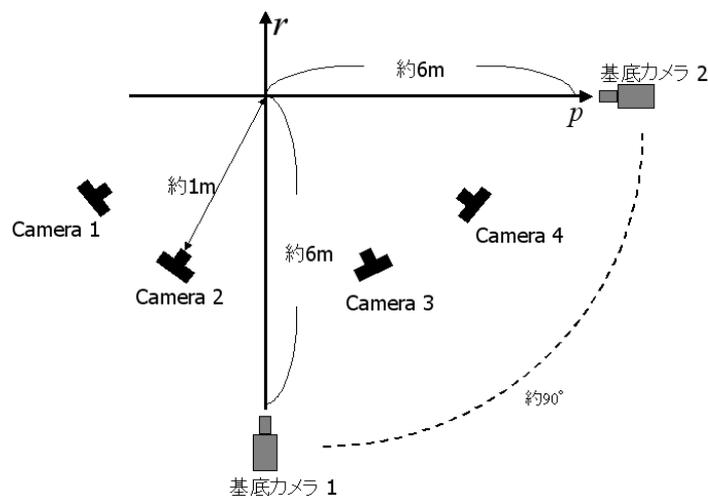


図 6.2: 実験時のカメラ配置

### 6.3 任意視点画像生成実験



(a) カラー画像 (カメラ 1)



(b) カラー画像 (カメラ 2)



(c) カラー画像 (カメラ 3)



(d) カラー画像 (カメラ 4)



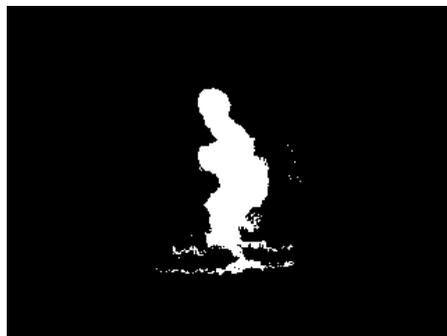
(e) シルエット画像 (カメラ 1)



(f) シルエット画像 (カメラ 2)



(g) シルエット画像 (カメラ 3)



(h) シルエット画像 (カメラ 4)

図 6.3: 多視点入力画像と得られたシルエット画像 (シーン 1)

### 6.3 任意視点画像生成実験



(a) 視点 1 (Facet)



(b) 視点 1 (Voxel)



(c) 視点 2 (Facet)



(d) 視点 2 (Voxel)



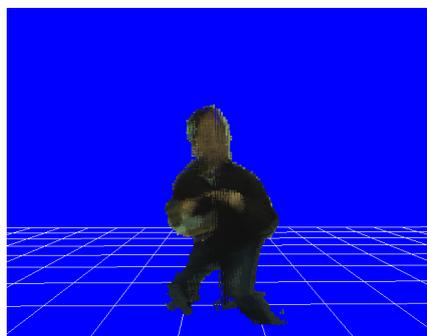
(e) 視点 3 (Facet)



(f) 視点 3 (Voxel)



(g) 視点 4 (Facet)



(h) 視点 4 (Voxel)

図 6.4: 任意視点画像 (シーン 1 - 1)

### 6.3 任意視点画像生成実験



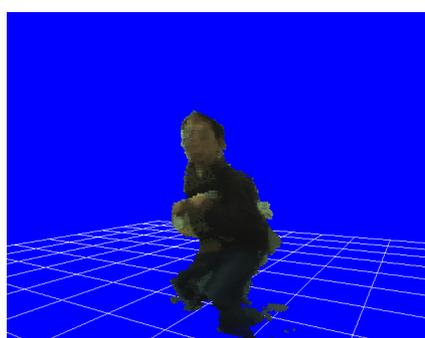
(a) 視点 5 (Facet)



(b) 視点 5 (Voxel)



(c) 視点 6 (Facet)



(d) 視点 6 (Voxel)



(e) 視点 7 (Facet)



(f) 視点 7 (Voxel)



(g) 視点 8 (Facet)



(h) 視点 8 (Voxel)

図 6.5: 任意視点画像 (シーン 1 - 2)

### 6.3 任意視点画像生成実験



(a) カラー画像 (カメラ 1)



(b) カラー画像 (カメラ 2)



(c) カラー画像 (カメラ 3)



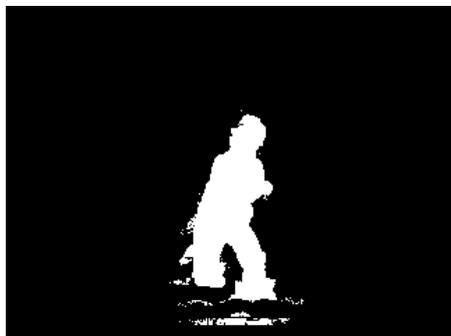
(d) カラー画像 (カメラ 4)



(e) シルエット画像 (カメラ 1)



(f) シルエット画像 (カメラ 2)



(g) シルエット画像 (カメラ 3)



(h) シルエット画像 (カメラ 4)

図 6.6: 多視点入力画像と得られたシルエット画像 (シーン 2)

### 6.3 任意視点画像生成実験



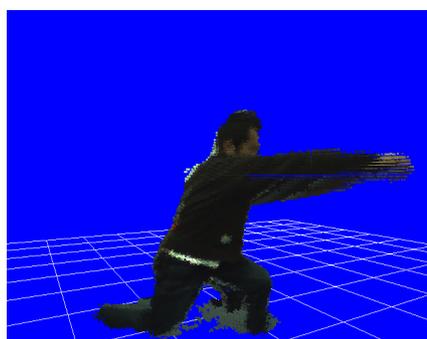
(a) 視点 1 (Facet)



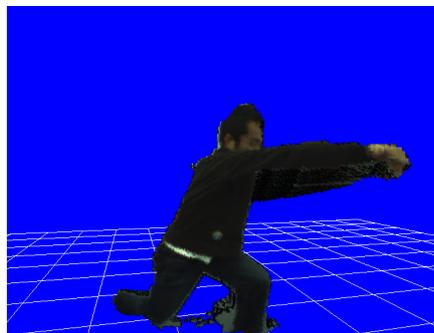
(b) 視点 1 (Voxel)



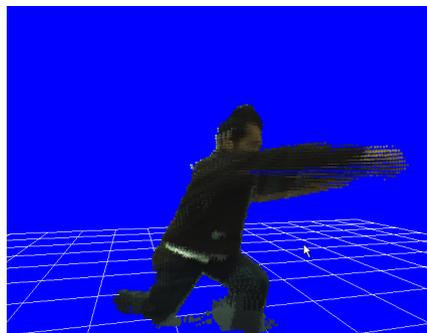
(c) 視点 2 (Facet)



(d) 視点 2 (Voxel)



(e) 視点 3 (Facet)



(f) 視点 3 (Voxel)



(g) 視点 4 (Facet)



(h) 視点 4 (Voxel)

図 6.7: 任意視点画像 (シーン 2 - 1)

### 6.3 任意視点画像生成実験



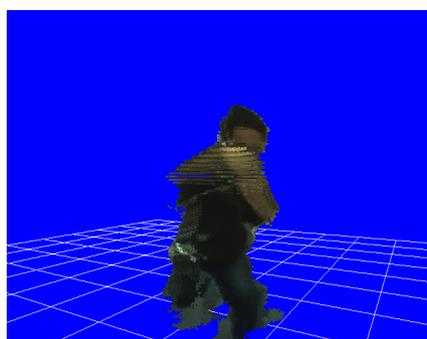
(a) 視点 5 (Facet)



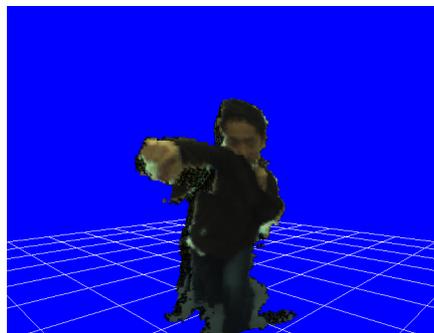
(b) 視点 5 (Voxel)



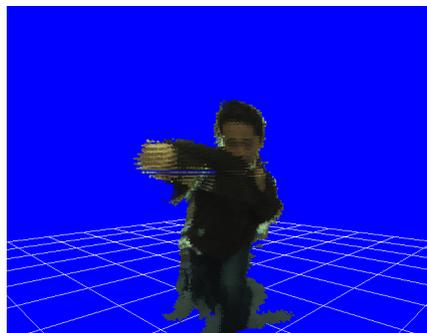
(c) 視点 6 (Facet)



(d) 視点 6 (Voxel)



(e) 視点 7 (Facet)



(f) 視点 7 (Voxel)



(g) 視点 8 (Facet)



(h) 視点 8 (Voxel)

図 6.8: 任意視点画像 (シーン 2 - 2)

### 6.3 任意視点画像生成実験



(a) カラー画像 (カメラ 1)



(b) カラー画像 (カメラ 2)



(c) カラー画像 (カメラ 3)



(d) カラー画像 (カメラ 4)



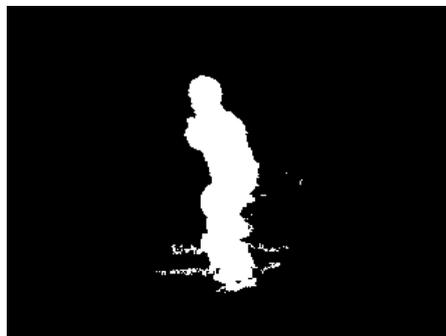
(e) シルエット画像 (カメラ 1)



(f) シルエット画像 (カメラ 2)



(g) シルエット画像 (カメラ 3)



(h) シルエット画像 (カメラ 4)

図 6.9: 多視点入力画像と得られたシルエット画像 (シーン 3)

### 6.3 任意視点画像生成実験



(a) 視点 1 (Facet)



(b) 視点 1 (Voxel)



(c) 視点 2 (Facet)



(d) 視点 2 (Voxel)



(e) 視点 3 (Facet)



(f) 視点 3 (Voxel)



(g) 視点 4 (Facet)



(h) 視点 4 (Voxel)

図 6.10: 任意視点画像 (シーン 3 - 1)

### 6.3 任意視点画像生成実験



(a) 視点 5 (Facet)



(b) 視点 5 (Voxel)



(c) 視点 6 (Facet)



(d) 視点 6 (Voxel)



(e) 視点 7 (Facet)



(f) 視点 7 (Voxel)



(g) 視点 8 (Facet)



(h) 視点 8 (Voxel)

図 6.11: 任意視点画像 (シーン 3 - 2)

### 6.3 任意視点画像生成実験



(a) カラー画像 (カメラ 1)



(b) カラー画像 (カメラ 2)



(c) カラー画像 (カメラ 3)



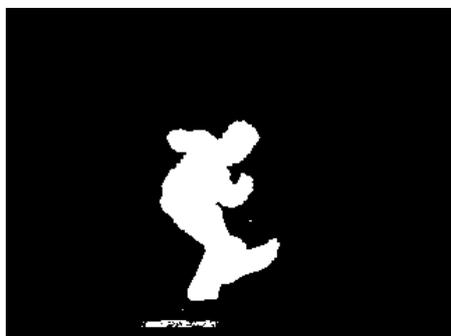
(d) カラー画像 (カメラ 4)



(e) シルエット画像 (カメラ 1)



(f) シルエット画像 (カメラ 2)



(g) シルエット画像 (カメラ 3)



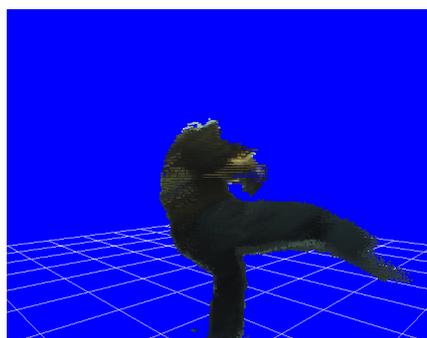
(h) シルエット画像 (カメラ 4)

図 6.12: 多視点入力画像と得られたシルエット画像 (シーン 4)

### 6.3 任意視点画像生成実験



(a) 視点 1 (Facet)



(b) 視点 1 (Voxel)



(c) 視点 2 (Facet)



(d) 視点 2 (Voxel)



(e) 視点 3 (Facet)



(f) 視点 3 (Voxel)



(g) 視点 4 (Facet)



(h) 視点 4 (Voxel)

図 6.13: 任意視点画像 (シーン 4 - 1)

### 6.3 任意視点画像生成実験



(a) 視点 5 (Facet)



(b) 視点 5 (Voxel)



(c) 視点 6 (Facet)



(d) 視点 6 (Voxel)



(e) 視点 7 (Facet)



(f) 視点 7 (Voxel)



(g) 視点 8 (Facet)



(h) 視点 8 (Voxel)

図 6.14: 任意視点画像 (シーン 4 - 2)

### 6.3 任意視点画像生成実験



(a) カラー画像 (カメラ 1)



(b) カラー画像 (カメラ 2)



(c) カラー画像 (カメラ 3)



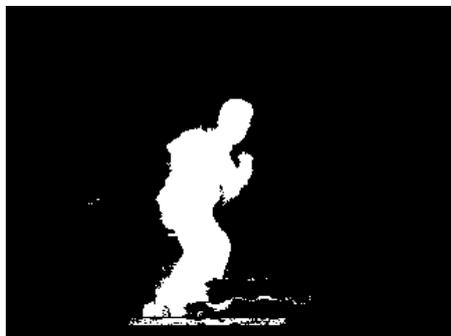
(d) カラー画像 (カメラ 4)



(e) シルエット画像 (カメラ 1)



(f) シルエット画像 (カメラ 2)



(g) シルエット画像 (カメラ 3)



(h) シルエット画像 (カメラ 4)

図 6.15: 多視点入力画像と得られたシルエット画像 (シーン 5)

### 6.3 任意視点画像生成実験



(a) 視点 1 (Facet)



(b) 視点 1 (Voxel)



(c) 視点 2 (Facet)



(d) 視点 2 (Voxel)



(e) 視点 3 (Facet)



(f) 視点 3 (Voxel)



(g) 視点 4 (Facet)



(h) 視点 4 (Voxel)

図 6.16: 任意視点画像 (シーン 5 - 1)

### 6.3 任意視点画像生成実験



(a) 視点 5 (Facet)



(b) 視点 5 (Voxel)



(c) 視点 6 (Facet)



(d) 視点 6 (Voxel)



(e) 視点 7 (Facet)



(f) 視点 7 (Voxel)



(g) 視点 8 (Facet)



(h) 視点 8 (Voxel)

図 6.17: 任意視点画像 (シーン 5 - 2)

## 6.4 共有仮想空間通信システム

本章で提案した任意視点画像合成手法を用いた応用例として、離れた2地点における多視点カメラシステムによって撮影した多視点画像列から、2地点での対象物体を共有される同じ仮想空間内で描画し、あらかじめ作成しておいたCGモデルと合成するシステムを構築した。本システムはコンピュータビジョンの手法を用いて対象物体を復元し、それを離れた場所で互いに見ることができるシステムである。本節ではこのシステムについて述べる。

### 6.4.1 関連研究

提案するシステムを説明する前に、3次元的な物体形状を仮想空間内で用いるシステムに関してどのような研究が行われているかについて説明する。

Lok[26] はユーザ本人の3次元形状を仮想環境内で反映し、その形状に応じて仮想環境内で物体を仮想的に触って動かしたりすることができるシステムを提案している。このシステムではまずユーザの多視点カメラ画像からそのユーザの3次元形状情報を Visual Hull によって取得する。このシステムではユーザの3次元形状を正確に復元せず、Visual Hull によっておおよその3次元的な形状情報のみを取得し、その形状情報から仮想環境内の物体との衝突判定や物体操作などを行っている。このシステムでは3次元的な物体形状情報を Visual Hull のみで取得しているため仮想環境を用いるシステム内でもリアルタイムで形状情報の取得、表示を行うことが可能である。このシステムは教育現場や作業環境において仮想環境が必要な場合を想定して提案されているため、複数人数での空間共有については特に触れられていない。

Beran[4] は複数ユーザで仮想空間を共有して作業を行えるシステムを提案している。このシステムではユーザは HMC(Head Mounted Camera) を装着し、そのカメラによって写したマーカー付きの物体を仮想空間で操作する。またユーザ自身の画像を2台の WMC(Wall Mounted Camera) によって撮影してこの画像を元にユーザ自身を仮想空間内へ配置する。このシステムでは2台の WMC によって得られた視差情報を元にユーザの奥行き方向を算出している。これはユーザがリアルタイムに仮想物体を操作することができるように複雑な物体形状処理を行わずに簡易的に物体形状を取得するためである。よってユーザ前面の視点以外ではユーザの形状が正確に表すことができない。

Baker[3] らは“Coliseum”と呼ぶ、ビデオ会議のための共有仮想空間システムを提案している。このシステムではビデオ会議参加者が使用する PC に5台のカメラを配置し、このカメラ画像から IBVH[30] と同じ手法を用いてビデオ会議参加者の Visual Hull を生成する。そしてこれを会議参加者が共有する仮想空間に配置して会議を行う。前述したように IBVH は独自のアルゴリズムを使用しているため非常に高速に3次元的な物体形状情報を取得できる。このアルゴリズムを用いているのは、提案するシステムの目的がビデオ会議に用いられることであり、リアルタイム性を重視しているからである。よってこのシステムでも参加者の明示的な3次元形状を取得することはできない。

Kauff[20] らも3次元空間を共有するビデオ会議システムを提案している。このシステムでは会議参加者を2台のカメラによって撮影し、彼ら独自のステレオ法によって参加者

## 6.4 共有仮想空間通信システム

の視差画像を取得する。この方法ではユーザが会議中によく動かすと考えられる手の領域を抜き出すことができ、手領域の隠れに関しては非常に高い精度で補間することが可能となっている。そして、視差画像から得られたユーザの物体形状は共有仮想空間内の位置に応じて会議参加者達へ提示される。このシステムでもやはりビデオ会議での臨場感を出すことに主眼が置かれ、ユーザの3次元形状の扱いは正確な形状を求めることではなく、参加者に対して適切な形状で表示されることを目的とされている。

Nakashima[32]らは2D, 3D環境での作業を複数ユーザで共有することができるシステムを作り提案した。彼らのシステムでは複数のユーザ環境を同一の IllusionHole という作業スペースに投影する。個別の作業環境は VNC[67] によって共有され、ユーザ操作は2D, 3Dともに同じスティック付きのマウスで行うことができる。この共有空間は3D空間を基本とし、その中で2Dのウィンドウを表示したり移動したりすることができる。彼らのシステムは3次元空間をユーザで共有しているが、実世界でのマウス位置以外の3次元情報は利用せずに3次元形状のデータをユーザ同士で協調作業を行えるのみである。そのため、実世界の3次元物体形状をこの空間内で共有することはシステム使用の目的範囲外となっている。

### 6.4.2 本システムの目的

前節で述べたように、仮想空間を共有してその中で3次元的な物体形状を利用するシステムはビデオ会議や協調作業においてユーザの存在を3次元空間内にて適切に表示することが主な目的となっており、ユーザの明示的な形状を利用していない。そこでユーザ形状を明示的に復元してそれを仮想空間で共有するシステムを提案することを目的とする。本システムでは、ユーザ全体を撮影してそれを仮想空間にて共有することができるため、離れた場所において自分を含めた参加者全員の全身を見ながら協調作業をすることができる。例えば、離れた場所にいる参加者がゴルフやテニスのレッスン時にこのシステムを利用すれば同じ空間内で視点を変えながら、参加している全ユーザの動きを観測することができる。また、別途用意したCGモデルを配置した仮想空間に入ることができるため、現在はアバターを用いた3次元空間内でのチャットやゲームなどにおいて自分自身を登場させて、それらサービスへ参加することが可能となる。

## 6.4.3 システムについて

本システムは、離れた2地点における多視点カメラシステムによって撮影した多視点画像列から、2地点での対象物体を共有される同じ仮想空間内で描画し、あらかじめ作成しておいたCGモデルと合成する。対象物体の撮影から、各復元モデルの出力までの流れは、第6章に示した手法を用いている。図6.18に本システムの概念図を示す。また図6.19にはシステムの構成を示した図を載せる。

図6.19に示すように、本システムは8台のステレオカメラと、9台のPCから成り立つ。9台のPCはすべてギガビットイーサネットで構成されるLANに接続されている。9台のうち8台は撮影に用いるPCで、このPCのCPUはPentium III 750MHz、メモリは512MBである。また1台は表示用PCで、CPUはXeon 1.8GHz × 2、メモリは2GBである。これらのPCのOSはWindows2000である。8台の撮影用PCは4台

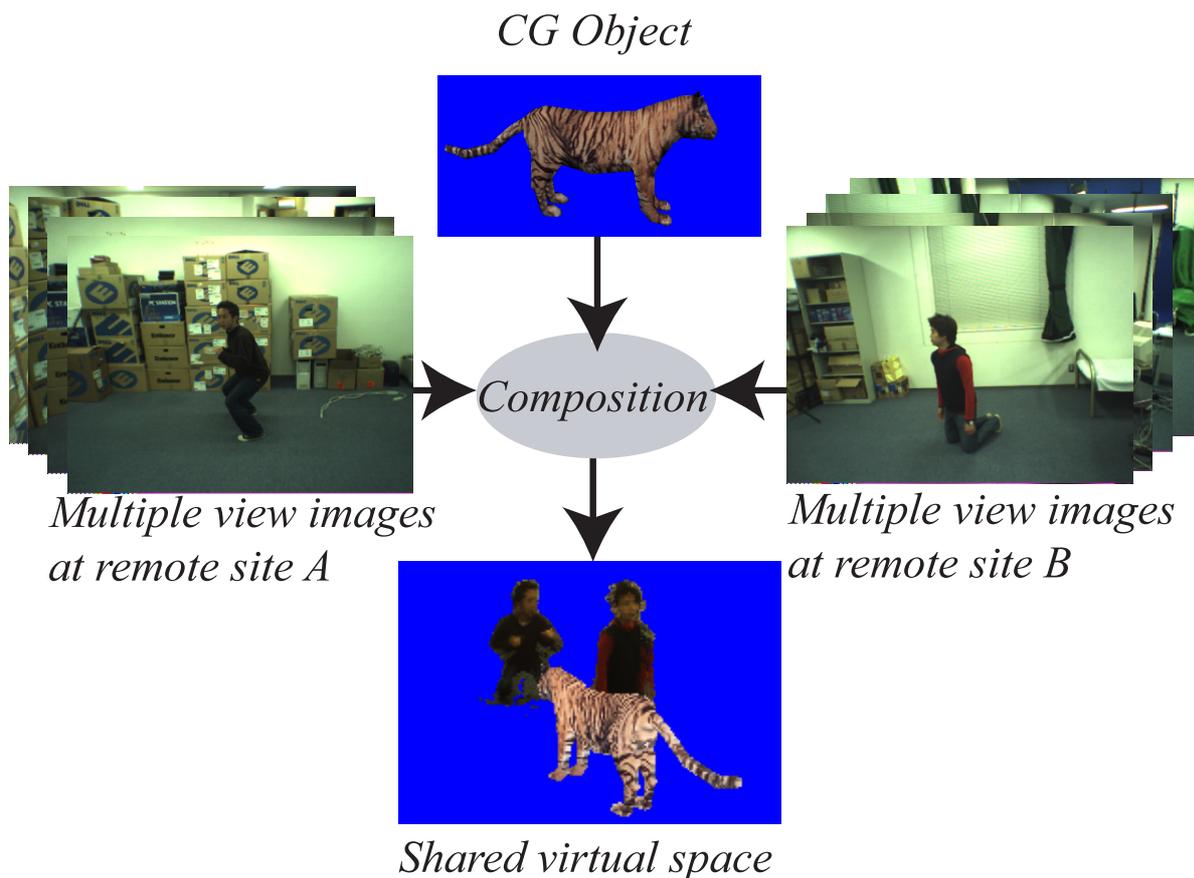


図 6.18: 共有仮想空間通信システム概念図

## 6.4 共有仮想空間通信システム

ずつ撮影地点に配置する。つまり、1地点の撮影には4つのPCを用いる。また、ステレオカメラは各撮影用PCとIEEE1394で接続されており、カメラとPC間で高速なデータ転送を行うことができる。なお、ステレオカメラ間の同期を取るための同期ユニットもIEEE1394で接続されている。

本システムで用いるステレオカメラはPoint Grey Research社の”Color Digiclops”[61]で、1つのステレオカメラは3台のCCDカメラによって構成されるため、カラー画像のキャプチャと同時に視差画像を生成することができる。この視差画像は2眼ステレオ法の組み合わせによって生成される。各ステレオカメラはPCとIEEE1394で接続することができる。このステレオカメラは640×480画素、320×240画素、160×120画素の、3つのサイズの画像を撮影することができる。本システムではこのカメラを用いて、320×240画素、色深度24bitRGBカラー画像と320×240画素、色深度8bitグレースケールの視差画像を撮影する。シルエット画像も320×240画素、色深度8bitである。また、本システムでは同社の”SYNC Unit”[65]を使うことで、異なるPCに接続されているステレオカメラ間の同期を取っている。ステレオカメラと同期ユニットの接続もIEEE1394である。

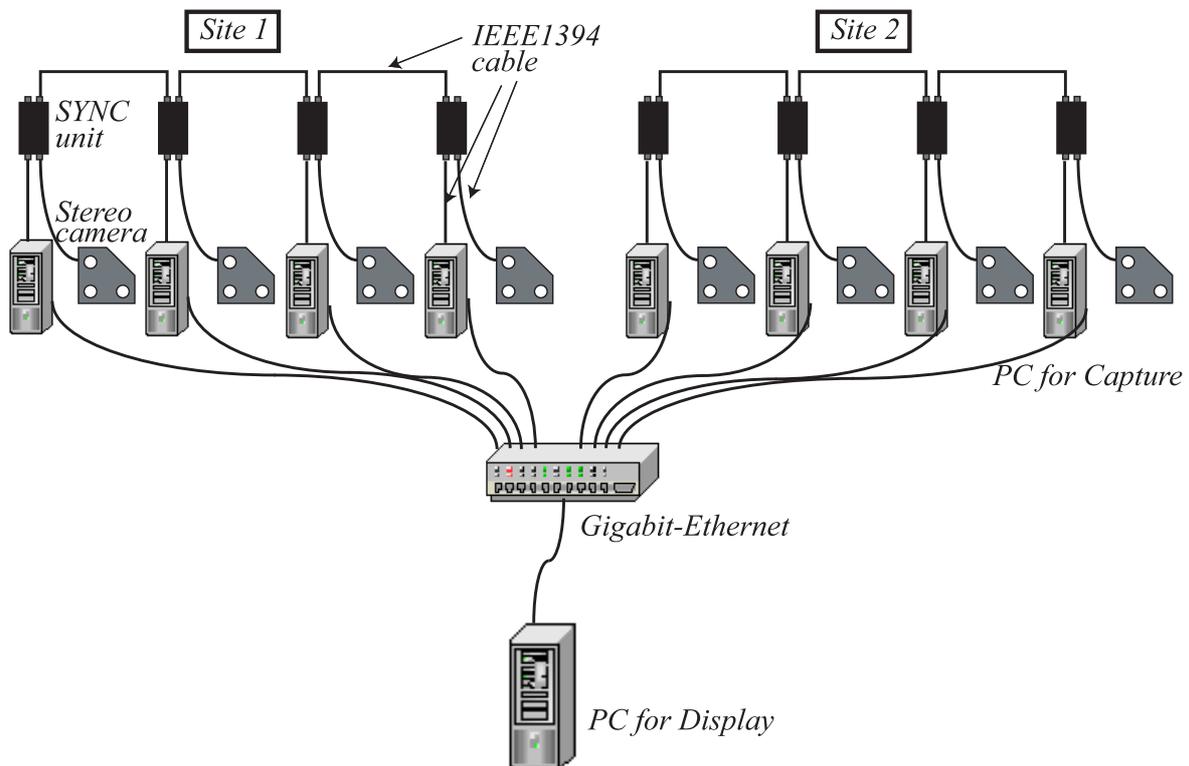


図 6.19: システム構成

## 6.4 共有仮想空間通信システム

図 6.20 にこのステレオカメラと同期ユニットの図を載せる .



(a) ステレオカメラ



(b) 同期ユニット

図 6.20: ステレオカメラと同期ユニット

## 6.4 共有仮想空間通信システム

次に、システムの処理分担を図 6.21 に示す。撮影用 PC では対象物体のカラー画像と視差画像を同時に撮影する。得られたカラー画像、視差画像とあらかじめ撮影しておいた背景カラー画像と背景視差画像を用いて、背景差分を行う。この背景差分によって、対象物体のシルエット画像を得ることができる。なお、本システムでは視差画像を物体形状復元には利用せず、背景差分のみに用いる。これは、得られる視差の値が物体形状復元には荒すぎるからである。

シルエット画像の生成後、撮影用 PC においてカラー画像とシルエット画像に対して JPEG 圧縮を行う。これはデータ転送量削減のためである。8 台の PC から同時に無圧縮のカラー画像とシルエット画像を転送してしまうと、ネットワークトラフィックが増大し、データ転送の遅延につながってしまうが、JPEG 圧縮することで、ネットワークトラフィックを軽減することができ、データ転送の遅延を小さくすることができる。なお、本システムでは JPEG の品質を 90% にしている。その結果、平均でデータ量を約 10% にまで圧縮している。

表示用 PC において、撮影用 PC から転送されたカラー画像とシルエット画像を受信した後、対象物体形状復元を行う。この処理では、2 地点で撮影された対象物体の形状

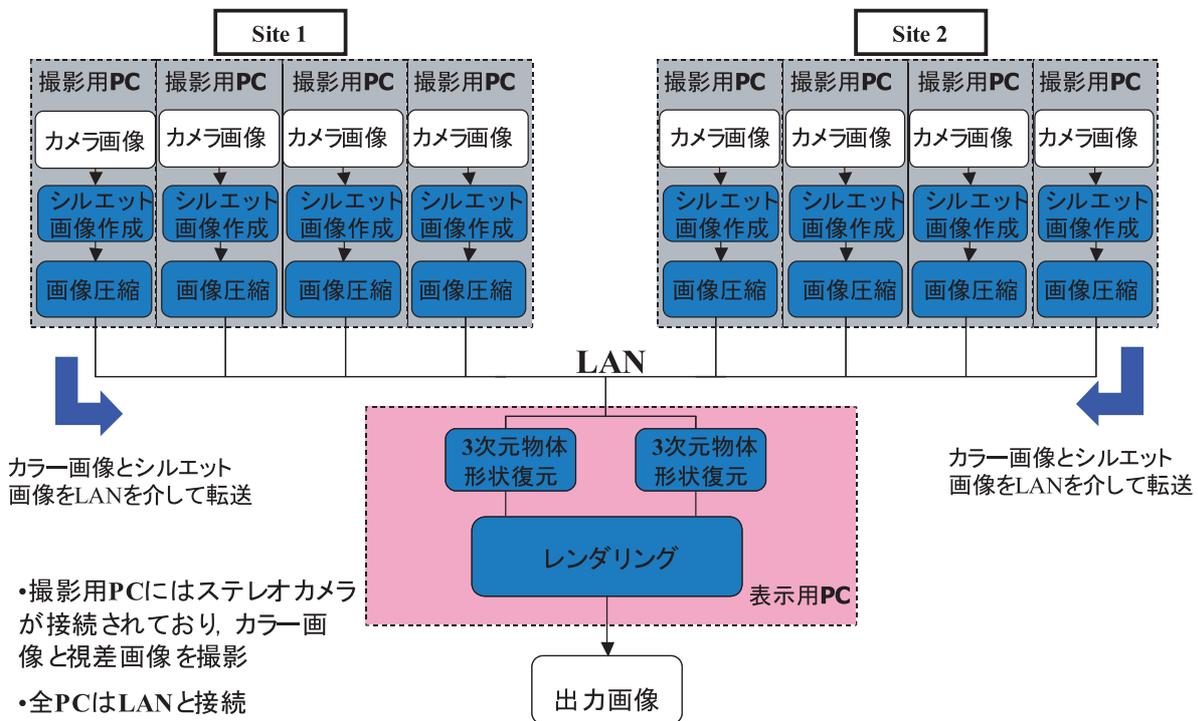


図 6.21: システムの処理分担

## 6.4 共有仮想空間通信システム

を、それぞれ Octree モデルとしてまず復元する。そして表面 Voxel モデルへと変換して、Voxel の着色を行う。本システム構築時には Microfacet Billboarding 法によるレンダリングは行っていない。これは可能な限り処理速度を上げるためである。

その後、指定された仮想視点の位置に応じて、任意視点画像として着色された Voxel モデルのレンダリングを行う。本システムでは高速なレンダリングを行うために、DirectX アーキテクチャの Direct3D[63] を使用する。Direct3D を使用することで、ハードウェアのレベルで、射影変換などの行列を用いた幾何変換の計算や、Z-Buffer の計算などを行うことができるため、システムのさらなる高速化を図ることができる。

## 6.4.4 提案システムに対する検討

ここでは、提案した共有仮想空間通信システムについての検討を行う。図 6.22 に本システムの実行時の様子を示す。図中右側の 8 つの画像のうち、上の 4 つは 1 つめの撮影地点において対象物体を撮影した画像で、下の 4 つはもう 1 地点において対象物体を撮影した画像である。また、図中左側に描画されている画像が共有仮想空間を描画したものとなっている。このようにあらかじめ用意した同一仮想空間内に 2 地点で撮影した対象物体を配置することができる。ことがわかる。

図 6.23 に対象物体の撮影から任意視点画像生成までのおおよそのタイムラインを示す。これにより平均で毎秒 3 フレームを実現していることがわかる。3 次元物体形状復元をおよそ 120msec で行っていることが大きく寄与している。Octree 構造を用いずに全 Voxel を用いる視体積交差法の計算時間は、同じ計算機環境で行った場合、約 5200msec であり、計算時間を大幅に削減できていることがわかる。また、本システムでは背景差分に視差画像を用いて時間をかけて行っているが、表示用 PC 側の処理が終わるまで撮影用 PC

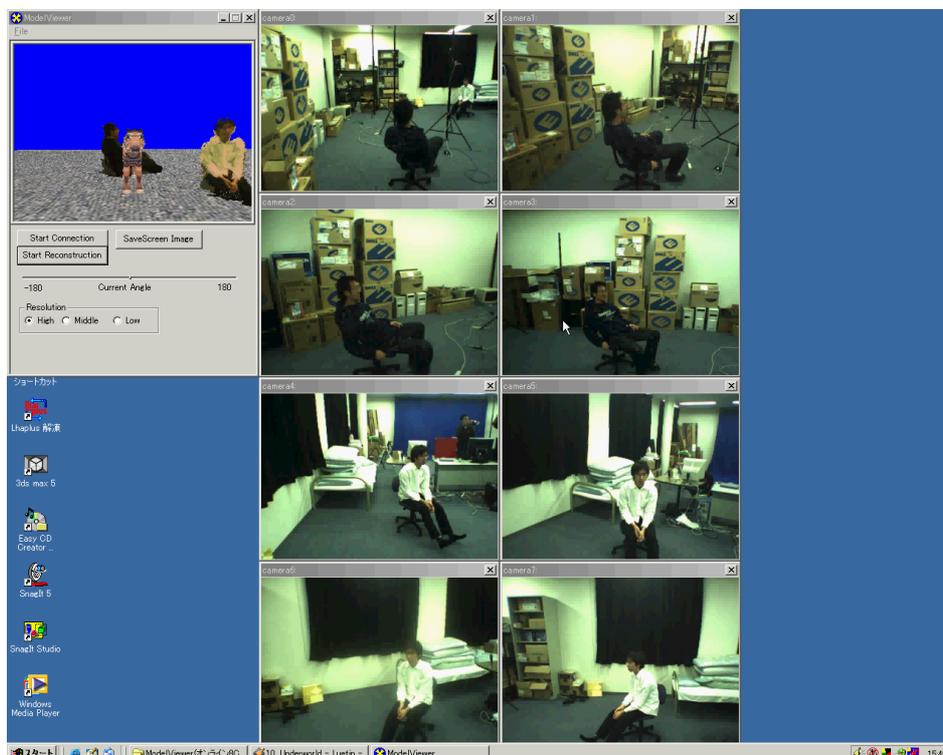


図 6.22: システム実行時の様子

## 6.4 共有仮想空間通信システム

では待機するため、背景差分にかけている時間がシステム全体の処理速度に影響を与えないことも、図 6.23 よりわかる。しかし、まだコマ送りのような出力しか行うことができず、システムの処理速度向上を行う必要があるものと思われる。現在、仮想空間に描かれる対象物体の描画のために Octree 形式のモデルから、最大解像度が  $256 \times 256 \times 256$  Voxel のモデルへと変換し、モデルを構成する Voxel に対して着色を行っているが、表面 Voxel のみでも約万 Voxel 分の着色を行う必要がある。この、レンダリングのための着色処理の計算速度を向上させることができれば、さらに出力速度を向上させることができると思われる。さらに、最近主流になりつつあるシェーダーレベルでのプログラミングを実施することにより、劇的な処理速度向上が期待できる。

なお、本システムでは解像度を可変にして、 $256 \times 256 \times 256$ 、 $128 \times 128 \times 128$ 、 $64 \times 64 \times 64$  の 3 通りで動作することが可能である。システム全体の処理速度は解像度を下げれば下げるほど高速化し、解像度に応じて、最大解像度時の速度に比べて 8 分の 1、64 分の 1 となるが、結果画像の質は解像度を落とせば落とすほど荒くなる。

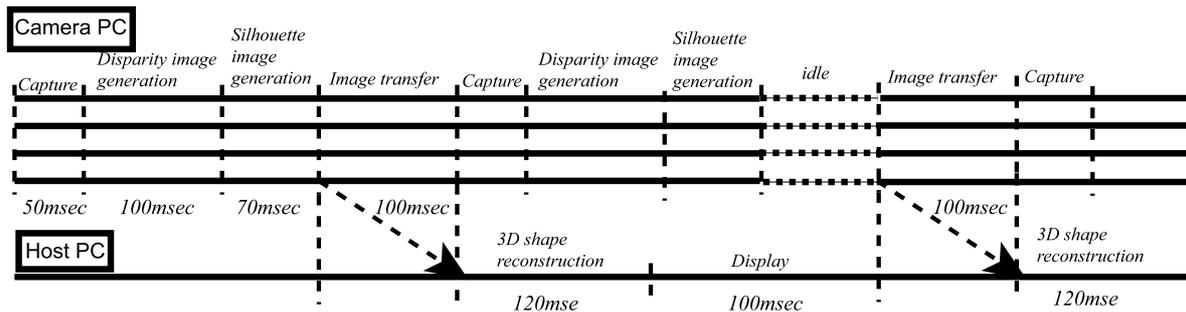


図 6.23: 本システム実行中のタイムライン

## 第 7 章

### 結論

## 7.1 まとめ

本論文は画像情報のみで仮想空間を定義し、その空間内において対象物体のモデリングを行い、その任意視点画像を生成するために行った研究について述べたものである。まずはじめに本手法の概要をまとめる。

第1章では本研究を行うにあたって、その動機となった産業界の現状や、現状を打開するために行われている関連研究とそれらの問題点について述べた。そしてそれらに対する本研究の位置付けを明確に述べた。

また、第2章では本研究の説明に必要と考えられる理論について説明した。

第3章では任意背景下において影領域の誤抽出を改善するための背景差分手法を提案した。本手法は、カラー画像と視差画像それぞれで背景差分を行う。まず、視差画像を用いた背景差分を行う。視差画像は影の影響を受けないため、この処理では対象物体のみを抽出することができる。しかし視差画像の解像度は粗いため、次に視差画像の背景差分結果に対してカラー画像を用いた背景差分を行う。カラー画像の背景差分では背景画像と原画像間の色の距離と角度を閾値処理して背景か前景かの判定を行う。この2つの処理を各画素に対して行い、画像内での対象物体領域の抽出を行う。本手法では視差画像を用いることで任意背景下における影領域の誤抽出を改善することが可能となる。

次に、第4章では Octree 構造を用いた視体積交差法の独自アルゴリズムを提案した。Octree 構造を用いた視体積交差法はすでにいくつか研究が行われているが、これらの手法は仮想3次元空間を8等分した立方体の集合であると考え、各立方体がカメラ画像の物体領域内に存在するかどうかの判定を行う。立方体が物体の内部、もしくは外部に完全に含まれる場合は立方体の状態を決定してしまうが、立方体が物体の内部と外部にまたがって存在する場合は、その立方体をさらに分割して再帰的にこの処理を繰り返す。そしてこの立方体の状態を木構造で表現すると、この8分木が物体形状を表すデータ構造となる。ここでさらに本手法では8分木の親の階層を参照することで、子の代での立方体への処理を省略することを可能とした。つまり、親の階層を参照するという木構造での概念を導入した結果、コンピュータビジョンにおける投影処理の計算を省くアルゴリズムを実現できたことになる。

また、第5章ではカメラキャリブレーションを行わなくても仮想空間とカメラ画像間の対応関係を得ることができる手法を提案した。まず2台の正射影とみなせる基底カメラを別途用意し、このカメラと撮影用カメラ間での Fundamental Matrix を算出する。Fundamental Matrix の算出には各画像内での対応点群だけでよい。Fundamental

Matrix を算出したら，基底カメラから発するカメラ光線が各カメラ画像へ投影されることによって出現するエピポーラ線を用いて OPGS の定義を行う．従来は，そのカメラ幾何の特性によってユークリッド空間のような直交座標系でなかった PGS だが，カメラ配置と画角に制限を加えることでユークリッド空間と遜色のない座標系を持つ OPGS を定義することを可能とした．

そして第 6 章では前述した 3 つの手法を組み合わせることで，対象物体形状モデリングを行い，任意視点画像を生成する手法を提案した．この手法ではまずカメラ画像を入力する前に OPGS の定義を行う．OPGS が定義されたら，あらかじめ撮影しておいたカラー背景画像，視差背景画像と現在撮影したカラー画像，視差画像を入力する．これら画像から対象物体領域の抽出を行い，その情報から物体形状のモデリングを行う．モデリング後に仮想視点位置に適切な色を割り当てて任意視点画像を生成する．

## 7.2 本手法によって解決した問題点

次に、提案した3つの手法によって解決した問題点について説明する。

第3章において述べた背景差分手法によって従来は困難であった任意背景下での影領域誤抽出の問題を改善することができた。従来研究では、ある特定の環境において適切な物体領域抽出を行うことが多かった。また、ある特定の問題に対しては安定した物体領域抽出を行っても、他の問題には弱いという特徴もあった。しかし本手法では任意背景下で影領域を適切に取り除くことができ、これによって従来は映画撮影等でよく用いられているブルーバック設備が必要なくなる。背景差分に用いる各種パラメータを適切に設定すれば本手法によっていかなる環境でも対象物体領域を精度良く抽出することができる。

第4章で述べた高速な物体形状復元手法により、カメラ画像から対象物体形状をモデリングする計算量を削減することが可能となった。これにより物体形状モデリングに必要な計算機環境を簡易化することができ、かつ、将来において計算機性能が向上すればカメラ画像を撮影しながら物体形状モデリングを行うことも可能となる。映像効果製作現場での制作期間はより短縮するものと考えられる。

また、第5章にて説明したOPGS定義手法により、コンピュータビジョンを用いた映像表現方法の産業へ応用がより現実的になった。従来は対象物体モデリングにカメラキャリブレーションが必須であり、大規模環境でのカメラキャリブレーションはその煩雑さゆえに非現実的であった。しかし本手法を用いれば、一度撮影だけしてしまえば撮影現場でなくても仮想空間定義のための情報は整っており、撮影期間の短縮化を図ることができる。かつ、OPGSはユークリッド空間と同等に扱っても見た目への影響は非常に小さいため、その後の映像加工の際に特殊な処理や加工を加える必要がない。よって、この手法も映像製作現場での制作期間短縮化に大きく貢献できるものと考えられる。

そして第6章での手法により、映像の撮影から対象物体形状のモデリング、任意視点画像生成までを行えることを示した。よって本手法を組み合わせることで映像加工方法の1つとして撮影した物体の形状を利用したり、任意視点画像を生成したりすることが可能となる。本手法を直接、もしくは部分的に用いるだけでも映像製作分野への貢献が可能であると考えられる。

## 7.3 結論

最後に結論を述べる．

本論文では視差画像を用いた背景差分手法，独自の Octree アルゴリズムによる高速な物体形状復元手法，そしてカメラ画像情報からの仮想空間定義手法について述べた．

まず，提案した視差画像を用いた背景差分手法によって，従来のカラー画像だけからでは困難であった影領域の影響を最小限に抑えた対象物体抽出が可能となった．この手法では視差画像を用いることで影領域の誤抽出が改善され，自由背景下での背景差分を容易に行うことが可能となった．これにより対象物体抽出に必要な撮影環境をより小規模にすることができると考えられる．

また，提案した独自の Octree アルゴリズムにより高速な物体形状復元が可能となった．本手法は数ある高速な物体形状復元手法の中の1つとして，従来は  $O(N^3)$  のオーダーが必要だった処理の高速化を実現した．本手法により対象物体モデリングに必要な計算機環境が簡易化され，将来のより高速な計算機環境において実時間処理が期待できると思われる．

さらに，実用化において問題となっていたカメラキャリブレーションの煩雑さを解消することができ，かつ，射影グリッド空間の簡易性を損なうことなくその座標系のゆがみを解消することが可能な擬似正射影グリッド空間を提案した．本手法により，射影グリッド空間において復元されたモデル形状がユークリッド空間内において歪んでしまっていた問題点を解決することができ，復元形状をユークリッド空間内でそのまま任意視点画像生成に利用することが可能となった．つまり，カメラキャリブレーションを行わなくても，対象物体形状を視覚的に捉えるには十分の精度で復元するための仮想空間を定義することができた．提案した仮想空間を用いることでカメラキャリブレーションの煩雑さをなくし，かつ現状の CG で用いられているグラフィックスプロセッサに含まれる機能をそのまま利用することが可能であるため，既存のゲームや映像製作への応用が簡易化される．その結果，これら産業への貢献を果たせるものと推測できる．

そして，前述した3つの手法を組み合わせることで対象物体のモデリングを行い任意視点画像を生成する手法について提案し，その結果画像を示した．さらに，応用例として本手法を用いて2つの遠隔地において撮影した複数物体を共有仮想空間において復元し，自由視点として出力するシステムを提案した．システムは2地点における対象物体形状を復元して同一空間にて描画することを可能とし，離れた位置からでも互いの全身を見ることが可能である．本手法は任意背景下において画像内から対象物体を抽出でき，対象物体モデリン

グを高速に行える．そして空間定義は簡易化されているため，将来的に遠隔地において全身を眺めることに価値のあるアプリケーションへの応用が期待できる．

以上，本論文では多視点カメラ画像から対象物体領域の抽出を行い，その画像から物体形状のモデリングをカメラキャリブレーションなしで行い，その任意視点位置における画像を生成する手法について述べた．そして応用例として仮想空間共有システムについても提案した．本手法を利用することにより，従来は実用化に向けて大きな足かせとなっていたカメラキャリブレーションを行うことなく，対象物体の任意視点映像を生成することができるようになった．これを応用することで，現在多大なコストがかかっているゲームや映画分野での利用が可能であると考えられ，映像や視覚表現分野の産業への貢献が期待できる．

## 謝 辞

本研究は、著者が慶應義塾大学理工学部、および慶應義塾大学大学院理工学研究科在学中に、斎藤英雄助教授のもとで行なったものである。本研究を遂行するにあたり、終始御指導くださり、本研究の内容の詳細にわたり貴重な御助言を賜りました斎藤英雄助教授に心から感謝します。また、本研究の詳細にわたり終始熱心に丁寧な御指導、御助言を頂きました小沢慎治教授に深く感謝致します。また、本論文執筆にあたり慶應義塾大学の岡田謙一教授、池原雅章教授には多大なる御教示、御助言をいただきました。心より感謝いたします。

また、本研究を進めるにあたり様々な面で支えて下さった小沢・斎藤研究室 59 期の皆様に心から感謝いたします。そしてさまざまな点においてご指導頂いた産業技術総合研究所 木村誠博士に、また研究室配属時からお世話になった斎藤英雄研究室、および小沢・佐藤研究室の皆様に深く感謝いたします。

そして、コンピュータサイエンスのおもしろさを働きながら学ぶ機会を下さったソニーコンピュータサイエンス研究所 暦本純一博士、飛田博章博士に深く感謝いたします。また、社会人学生として後期博士課程在籍中に業務と学業の両立を暖かく見守り、そしてご支援して下さいましたソニー株式会社 山本巖氏に心よりお礼申し上げます。

最後に、大学入学から現在に至るまでに支えて下さった全ての方に深く感謝いたします。

## 参考文献

- [1] Adiv, G. : “Determining Three-Dimensional Motion and Structure From Optical Flow Generated by Several Moving Objects”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 7, no. 4, pp.384-401, 1985.
- [2] Baker, H. : “Three-dimensional modelling,” 5th Int. Joint Conf. artif. Intell. '77, pp.649-655, 1977.
- [3] Baker, H. H., Bhatti, N., Tanguay, D., Sobel, I., Gelb, D., Goss, M. E., Culbertson, W. B., Malzbender, T. : “Understanding Performance in Coliseum, An Immersive Videoconferencing System”, ACM TOMCCAP Vol.2, Issue 1, pp.190-210, 2005.
- [4] Beran, V. : “Augmented Multi-User Communication System,” Proc. of the working conference on Advanced visual interfaces, pp.257-260, 2004.
- [5] Boufama, B., Weinshall, D., Werman, M. : “Shape from motion algorithms: A comparative analysis of scaled orthography and perspective”, J.-O. Eklundh, Computer Vision-ECCV '94, pp.199-204. Berlin-Heidelberg: Springer-Verlag, 1994.
- [6] Broadhurst, A., Drummond, T.W., Cipolla, R. : “A Probabilistic Framework for Space Carving” Proc. ICCV01, Vol.I, pp.388-393, 2001.
- [7] Bullitt, E., Aylward, S. R. : “Volume Rendering of Segmented Image Objects”, IEEE Transactions on Medical Imaging, vol. 21, Issue 8, pp.998-1002, August, 2002.
- [8] Carranza, J., Theobalt, C. Magnor, M., Seidel, H.P. : “Free-Viewpoint Video of Human Actors,” Proc. of ACM SIGGRAPH 2003, p.569-577, 2003.
- [9] Cheung, G. K. M., Kanade, T., Bouguet, J. Y., Holler, M. : “A real time system for robust 3D voxel reconstruction of human motions,” Proc. Computer Vision

- and Pattern Recognition Conf 2000. IEEE Comput. Soc, Los Alamitos, CA, USA, Vol.2, pp.714-729, 2000.
- [10] Debevec, P. E., Taylor, C. J., Malik, J. : “Modeling and Rendering Architecture from Photographs: A Hybrid Geometry and Image-Based Approach,” Proc. of ACM SIGGRAPH '96, pp.43-54, August, 1996.
- [11] Eisert, P., Steinbach, E., Girod, B. : “Automatic Reconstruction of Stationary 3-D Objects from Multiple Uncalibrated Camera Views,” IEEE Transactions on Circuits and Systems for Video Technology, Vol.10, No.2, pp.261-277, March, 2000.
- [12] Francois, A. R. J., Medioni, G. G. : “adaptive color background modeling for real-time segmentation of video streams”, Proc. of International on Imaging Science, System and Technology, pp.227-232, 1997.
- [13] Goldlücke, B., Magnor, M. : “Real-time, Free-viewpoint Video Rendering from Volumetric Geometry,” Proc. SPIE Conference on Visual Communications and Image Processing, Vol.5150, pp.152-1158, June, 2003.
- [14] Gortler, S. J., Grzeszczuk, R., Szeliski, R., Cohen, M. F. : “The lumigraph,” Proc. of ACM SIGGRAPH '96, pp.43-54, August, 1996.
- [15] Inamoto, N., Saito, H. : “Fly Through View Video Generation of Soccer Scene,” International Workshop on Entertainment Computing(IWEC2002), pp.109-116, May, 2002.
- [16] Kamkar-Parsi, A. H., Laganiere, R., Bouchard, M. : “A Multi-Criteria Model for Robust Foreground Extraction”, Proc. of VSSN 2005, pp.67-70, , November, 2005.
- [17] Kanade, T., Rander, P. W., Vedula, S., Narayanan, P. J. : “Virtualized Reality: Concepts and early results,” IEEE Workshop on Representation of Visual Scenes, pp.69-76
- [18] Kanade, T., Rander, P. W., Vedula, S., Saito, H. : “Virtualized reality: digitizing a 3D time-varying event as is and in real time,” International Symposium on Mixed Reality(ISMR99), pp.41-57, 1999.
- [19] Kanatani, K., Ohta, N. : “Optimal Fundamental Matrix Computation:Algorithm and Reliability Analysis, ” Proc. of the 6th Symposium on Sensing via Imaging Information (SSII 2000), 14-16 pp.291-296 June. 2000.
- [20] Kauff, P., Schreer, O. : “An Immersive 3D Video-Conferencing System Using

- Shared Virtual Team User Environments,” Proc. of the 4th international conference on Collaborative virtual environments, pp.105-112, 2002.
- [21] Kimura, M., Saito, H. : “Stereo Matching between Three Images by Iterative Refinement in PVS,” IEICE Transactions on Information and Systems, Vol.E86-D, No.1, pp.89-100, January, 2003.
- [22] Kitahara, I., Ohta, Y., Saito, H., Akimichi, S., Ono, T., Kanade, T. : “Recording Multiple Videos in a Large-scale Space for Large-scale Virtualized Reality,” Proc. of International Display Workshop(AD/IDW’01), pp.1377-1380, 2001.
- [23] Laurentini, A. : “How many 2D silhouettes does it takes to reconstruct a 3D object?”, Computer Vision and Image Understanding, Vol.67, pp.81-87, 1997.
- [24] Lander, J. : “Shades of disney: Opaquing a 3d world,” Game Developers Magazine, March, 2000.
- [25] Levoy, M., Hanrahan, P. : “Light Field Rendering,” Proc. of ACM SIGGRAPH ’96, pp.31-42, August, 1996.
- [26] Lok, B. : “Online Model Reconstruction for Interactive Virtual Environments,” Proc. of the 2001 symposium on Interactive 3D graphics, pp.69-72, 2001.
- [27] Lorensen, W. E., Cline, H. E. : “Marching Cubes: A High Resolution 3D Surface Construction Algorithm”, Proc. of ACM SIGGRAPH 1987, Vol.21, pp.163-169, 1987.
- [28] Lyness, C., Marte, O., Wong, B., Marais, P. : “Low-cost model reconstruction from image sequences,” Proc. of the 1st international conference on Computer graphics, virtual reality and visualisation, pp.131-135, 2001.
- [29] Martin, W. N., Aggarwal, J. K. : “Volumetric descriptions of objects from multiple views,” IEEE Trans. Pattern Anal. Mach. Intell. PAMI-5, Vol.2, pp.150-158, 1983.
- [30] Matusik, W., Buehler, C., Raskar, R., Gortler, S. J., McMillan, L. : “Image-Based Visual Hulls,” Proc. of ACM SIGGRAPH 2000, pp.369-374, 2000.
- [31] Moons, T., Gool, L. V., Proesmans, M., Pauwels, E. : “Affine reconstruction from perspective image pairs with a relative object-camera translation in between”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, pp.77-83, 1996.
- [32] Nakashima, K., Machida, T., Kiyokawa, K., Takemura, H. : “A 2D-3D Integrated Environment for Cooperative Work, ” Proc. ACM Symp. on Virtual Reality

- Software and Technology (VRST), pp.16-22, November, 2005.
- [33] Nayar, S. K., Nakagawa, Y. : "Shape from Focus," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, Issue 8, pp.824-831, August, 1994.
- [34] Niem, W. : "Robust and Fast Modelling of 3D Natural Objects From Multiple Views," Proc. of SPIE Image and Video Processing II, Vol.2182, pp.388-394, 1994.
- [35] Potmesil, M. : "Generating octree models of 3D objects from their silhouettes in a sequence of images," Computer Vision, Graphics, and Image Processing , Vol.40, pp.1-29, 1987.
- [36] Saito, H., Kanade, T. : "Shape reconstruction in Projective Grid Space from a large number of images," Proc. Computer Vision and Pattern Recognition Conf., 1999.
- [37] Seitz, S. M., Dyer, C. R. : "Photorealistic Scene Reconstruction by Voxel Coloring," Proc. Computer Vision and Pattern Recognition Conf., pp.1067-1073, 1997.
- [38] Seitz, S. M., Dyer, C. R. : "View Morphing", Proc. of ACM SIGGRAPH '96, pp.21-30, August, 1996.
- [39] Srivastava, S., Ahuja, N. : "Octree Generation from Object Silhouettes in Perspective Views," CVGIP, vol.49, pp.68-84, 1990.
- [40] Szeliski, R. : "Rapid octree construction from image sequences," CVGIP: Image Understanding , Vol.58, pp.23-32, 1993.
- [41] Tsai, R. Y. : "A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses," IEEE Journal of Robotics and Automation, Vol.RA-3, No.4, pp.323-344, August, 1987.
- [42] Vedula, S., Baker, S., Kanade, T. : "Spatio-temporal view interpolation," Proceedings of the 13th Eurographics workshop on rendering, pp.1-12, 2002.
- [43] Vedula, S., Rander, P. W., Saito, H., Kanade, T. : "Modeling, Combining and Rendering Dynamic Real World Events From Image Sequences," Proc. 4th Conf. Virtual Systems and Multimedia, Vol.1, pp.326-332, 1998.
- [44] Wada, T., Xiaojun, W., Tokai, S., Matsuyama, T. : "Homography based parallel vol. intersection: toward real-time volume reconstruction using active cameras," Proceedings Fifth IEEE International Workshop on Computer Architectures for Machine Perception , pp.331-339, 2000.

- [45] Wolberg, G. : “Image Morphing: A Survey,” Visual Computer, Vol. 14, pp. 360-372, 1998.
- [46] Yaguchi, S., Saito, H. : “Arbitrary Viewpoint Video Synthesis From Multiple Uncalibrated Cameras”,IEEE Trans. SMC. PART-B., Vol.34, No.1 pp.430-439, 2004.
- [47] Yamazaki, S., Sagawa, R., Kawasaki, H., Ikeuchi, K., Sakauchi, M. : “Microfacet Billboarding” Proceedings of the 13th Eurographics workshop on rendering, pp.169-179, vol.13, 2002.
- [48] Yang, T., Li, S. Z., Pan, Q., Li, J. : “Real-Time and Accurate Segmentation of Moving Objects in Dynamic Scene,”Proc. of ACM 2nd international workshop on Video surveillance and sensor networks, pp.136-143, 2004.
- [49] Zhang, Z. : “Determining the Epipolar Geometry and its Uncertainty:A Review,” INRIA research report, 2927, 1996.
- [50] Zhang, Z. : “A Flexible New Technique for Camera Calibration,” Technical Report MSR-TR-98-71, Microsoft Research, 1998.
- [51] Zitnick, C. L., Kang, S. B., Uyttendaele, M., Winder, S. A. J., Szeliski, R. : “High-quality video view interpolation using a layered representation”, ACM Transactions on Graphics, Vol.23, Issue 3, pp.600-608, August, 2004.
- [52] 稲本奈穂, 斎藤英雄 : “サッカーシーンにおける自由視点映像生成のための視点内挿法”, 日本バーチャルリアリティー学会論文誌 TVRSJ, Vol.7, No.4, pp.513-520
- [53] 北原格, 石川寛享, 渡辺真生, 太田友一 : “大規模空間の多視点映像を用いた運動視差の再現可能な自由視点映像の生成・提示方式”, 画像電子学会論文誌, Vol.31, No.4, pp.477-486, 2002.
- [54] 西野恒, 佐藤いまり, 佐藤洋一, 池内克史 : “Eigen-Texture 法 : 複合現実感のための3次元モデルに基づく見栄の圧縮と合成,” 電子情報通信学会論文誌 D-II, Vol.J82J-D-II, No.10, pp.1793-1803, 1999.
- [55] 佐藤淳 : “コンピュータビジョン - 視覚の幾何学 - ”, ISBN 4-339-02363-9, 株式会社コロナ社, 1999.
- [56] 除剛, 辻三郎 : “3次元ビジョン”, ISBN 4-320-08522-1, 共立出版株式会社, 1998
- [57] 除剛 : “写真から作る3時限CG - イメージ・ベースド・モデリング&レンダリング - ”, ISBN 4-7649-0286-9, 近代科学社, 2001
- [58] 谷内田正彦 : “コンピュータビジョン”, ISBN 4-621-03460-X C3355, 丸善株式会社, 1990

- [59] 濱崎省吾, 吉田裕之, 重永信一: “多視点シルエット画像からの高速な3次元形状復元手法”, 第7回画像センシングシンポジウム講演論文集, pp.59-64, June, 2001
- [60] “APPLESEED”, <http://www.appleseedthemovie.com/>
- [61] “Color Digiclops”, <http://www.ptgrey.com/products/digiclops/index.html>
- [62] “EyeVision”, [http://www.ri.cmu.edu/projects/project\\_449.html](http://www.ri.cmu.edu/projects/project_449.html)
- [63] “Microsoft DirectX”, <http://www.microsoft.com/directx/>
- [64] “STAR WARS”, <http://www.starwars.com/>
- [65] “SYNC UNIT”, <http://www.ptgrey.com/products/sync/index.html>
- [66] “The Matrix”, <http://japan.whatisthematrix.com/>
- [67] “VNC”, <http://www.realvnc.com/>
- [68] “鬼武者(株式会社カプコン)”, <http://www.oni-musha.com>