

博士論文

自由視点テレビのための
多視点映像処理の研究

平成21年3月25日

指導教授 谷本正幸

名古屋大学大学院 工学研究科
電子情報システム専攻

学生番号 480633029

福嶋慶繁

本論文は，カメラアレイシステムを用いた，自由視点テレビシステムの実現を目指し，問題になる以下の課題について，関連研究をまとめると同時に解決策を論じる．具体的には，

- カメラの色合いを補正する色補正
- カメラ配置誤差を補正する幾何補正
- 自由視点画像合成とその最適化
- 自由視点テレビのプロトタイプ構築

について述べたものである．

目次

第1章 序論	1
1.1 研究背景	1
1.1.1 自由視点テレビ (Free Viewpoint Television, FTV)	3
1.1.2 FTVのデータ取得	4
1.1.3 補正処理	12
1.1.4 FTVの圧縮処理	14
1.1.5 FTVの自由視点画像合成	14
1.1.6 FTVの自由視点画像表示	21
1.2 研究の目的	22
1.3 本論文の構成	23
第2章 多視点映像の色補正処理	25
2.1 緒言	25
2.2 関連研究	25
2.3 提案手法	28
2.3.1 カメラ特性	28
2.3.2 ゲイン調整	29
2.3.3 処理の流れ	30
2.4 実験および結果	31
2.4.1 実験	31
2.4.2 結果	32
2.5 結言	33

第 3 章	多視点映像の幾何補正処理	35
3.1	緒言	35
3.2	関連研究	36
3.3	カメラキャリブレーションによる画像補正	38
3.3.1	ピンホールカメラモデル	38
3.3.2	カメラキャリブレーション	40
3.3.3	回転による画像補正	42
3.3.4	提案手法	43
3.3.5	実験	46
3.3.6	キャリブレーションによる画像補正のまとめ	49
3.4	ノンメトリックな二次元カメラアレイ画像補正	51
3.4.1	理想状態と対応点軌跡	51
3.4.2	提案レクティフィケーション手法	54
3.4.3	実験および結果	63
3.4.4	自然特徴点による画像補正のまとめ	68
3.5	結言	70
第 4 章	多視点映像からの自由視点画像合成処理	71
4.1	緒言	71
4.2	関連研究	71
4.3	視点依存自由視点画像合成	74
4.4	自由視点画像合成最適化	75
4.4.1	ローカル最小化	76
4.4.2	グローバル最小化	77
4.5	提案手法	78
4.5.1	Dynamic Programming による視差推定	79
4.5.2	Multi-Pass Dynamic Programming	80
4.5.3	距離変換による高速化	82
4.6	実験	86
4.6.1	ローカル最小化	87

4.6.2	最大視差と PSNR	90
4.6.3	最大視差と計算時間	96
4.6.4	コストパフォーマンス	98
4.7	結言	98
第 5 章	FTV システムの構築	101
5.1	緒言	101
5.2	実時間 FTV システム	101
5.2.1	システム構成	102
5.2.2	実行結果	103
5.2.3	実時間 F T V システムのまとめ	104
5.3	自由視聴点システム	105
5.3.1	システム構成	106
5.3.2	撮影および自由視聴点生成生成	107
5.3.3	結果	108
5.3.4	自由視聴点のまとめ	109
第 6 章	結論	111
	研究業績	127

第1章 序論

1.1 研究背景

自由視点テレビ (FTV: Free Viewpoint TV) [1, 2, 3] とは, ユーザが自由に視点を選択することが可能であり, あたかもその場にいるかのように感じられる次世代のテレビのことであり, 本論文ではこの実現を目指している.

20世紀にテレビは開発されて以来, 1926年の高柳健次郎博士による世界初の電子式受像実験の成功, 1953年の商用放送開始, 1960年のカラー化, 2003年の地上デジタル放送開始などの大きな改革を経て, 現在に至っている. しかしながら, この発展の流れは, 色の3次元ベクトル化, 画像解像度の向上など, すでに決められたカメラワークで撮影された単一の視点の映像配信に限られている. 一方, 新たな試みとして, 試験的に立体ディスプレイ用の放送が行われたり, 立体映画等の放映が行われていたり, DVDなどのパッケージでユーザが, 数箇所視点の切り替えることが可能なものが販売されたりし始めている. このように, 現在のテレビは, より現実感の高い映像が放送可能となるように目指し始めている.

この立体映像を表示するためには, 最低, 二視点分の情報を, つまり人間の目の数だけの情報を送らなければならないが, より自然な表示をするためには, さらに多視点の映像が必要となるのだが, FTVには, 自由な視点での映像が生成可能な技術である, 自由視点画像合成機能を持っており, 従来のテレビにユーザが指示した映像を描画するだけでなく, その近傍の視点位置の映像生成をすることにより, 立体映像まで表示することが可能となっている.

このFTVが実現すれば、

- 文化遺産や伝統芸能のデジタル保存
- 教育番組
- エンターテインメント番組
- セキュリティシステム
- 自由視点テレビ

などの多数の応用が可能となる。

FTVのデータを取得する方法の中で最も有望なものは、多数のカメラを用いることで、ひとつのシーンを様々な方向から撮影する、マルチカメラシステムである。計算機的能力が劇的に向上し、加えてカメラが安価に製造できるようになったことから、100台を超えるようなカメラを使ったマルチカメラシステムが徐々に登場しはじめている[4, 5]。これにより、マルチカメラシステムで撮影した多視点映像から映像処理を行う技術が活発に研究されはじめている。

この多視点映像による映像処理は、自由視点画像生成[6, 7]や、3次元再構築[8, 9]、高解像処理[10]、可変焦点画像生成[11]など、さまざまな用途に適応可能である。これらの技術は開発途上の段階であり、IBR (Image Based Rendering) や MBR (Model Based Rendering)、コンピュータビジョンといった研究分野で盛んに研究されている。

次節以降ではこのFTVについて、データ入力である多視点映像からの情報取得を詳しく述べながら、全体概要を述べる。まず第1.1.1節にてFTVの概要を述べる。次に、第1.1.2節と第1.1.3節で光線情報取得と補正処理について述べる。また、1.1.4節にてFTVの情報圧縮処理について、さらに、第1.1.5節と第1.1.6節で、自由視点画像の生成および表示技術について述べる。

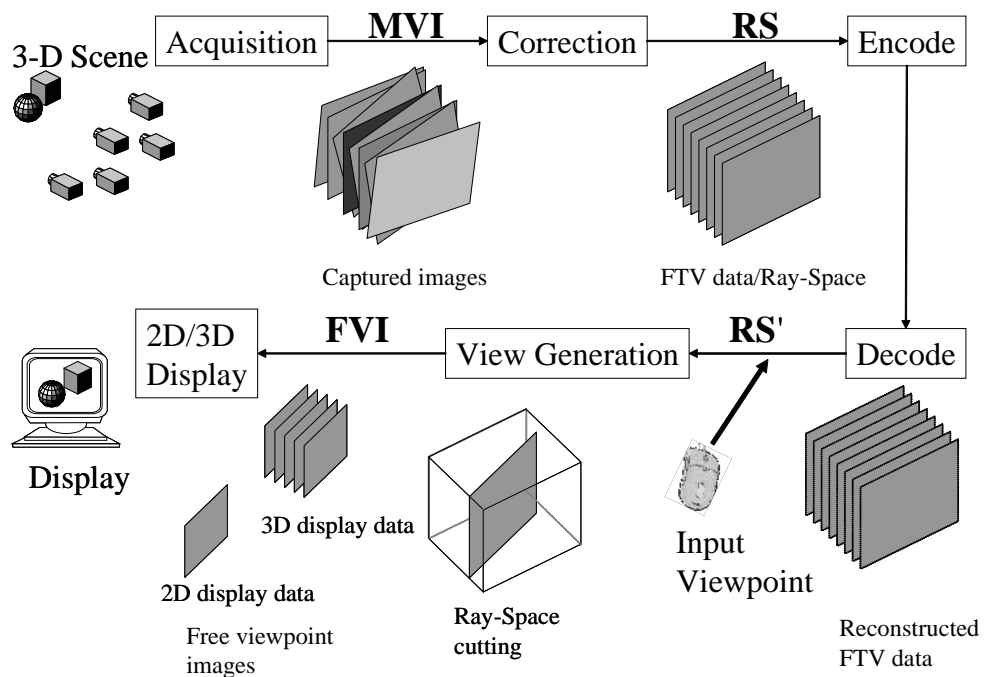


図 1.1 F T Vのプロセス

1.1.1 自由視点テレビ (Free Viewpoint Television, FTV)

ここでFTVのブロック図を図1.1に示す。図1.1にあるように、FTVでは、まずカメラなどにより、多視点からシーンを見た情報を取得する。この出力データMVI(Multi View Image)は三色RGBの画像色情報、画像解像度width×height、frame数、そしてその視点数Nの情報量、つまり、 $3 \times \text{width} \times \text{height} \times \text{frame} \times N$ という膨大な情報量となる。これらの取得処理については第1.1.2節で述べる。

次に、後段の自由視点画像の生成を高速に処理するためや、視点移動時に違和感を与えないようにするために、取得した多視点映像を補正処理する。これは、カメラなどには個体差があり、すべてのカメラで同じ色合いの画像を取得することが困難なことや、カメラの位置を設計通りに寸分の誤差もなく配置することが困難であることから、その補正が必

要となるからである。これらの処理については第 1.1.3 節で述べる。また、出力データ **RS** の情報量は未加工の入力データ **MVI** と同様である。これらの処理を終えると、次に光線情報を効果的に圧縮してユーザのいる場所へ送る。現在、最大の圧縮効率を持つ H.264/AVC エンコーダは動画データデータを 1/20 から 1/100 に圧縮することが可能だが、多視点画像の圧縮にそれを拡張すると AVC に比べて 10% から 20% 改善した圧縮が可能となっている。この圧縮については第 1.1.4 節で簡単に述べる。

最後に、受け取った圧縮符号化データを伸張して、自由視点画像を生成して表示する。この際、ユーザは視点を入力することで FTV データから所望の視点映像を取り出すことになる。取り出し方は出力するディスプレイに依存するが、たとえば一般的な平面ディスプレイであれば 1 視点分の動画データのデータ量が取り出され、また 2 眼立体ディスプレイならば 2 視点分の動画データが取り出される。この自由視点画像の合成については第 1.1.5 節で、表示については第 1.1.6 節で述べる。

1.1.2 FTV のデータ取得

(a) マルチカメラシステムによる FTV データの取得

～文献 [12] のシステム～

FTV のデータの取得装置としては、現状ではマルチカメラシステムが最も有力である。これは、カメラ間隔を自由に変えられるため、汎用の機材だけで目的に応じた密度の情報を取得しやすいと考えられているからである。

図 1.2 にあるシステムでは、カメラ 15 台と PC 15 台で映像を取得している。このシステムは本論文の前身となる FTV の生成装置である。各 PC には画像取得のための汎用キャプチャボードが装着されており、横 320 画素 x 縦 240 画素 (QVGA) の動画を 30 fps で取得できる。この装置の懸念点は、カメラおよびキャプチャボードの同期 (垂直同期、及び水平同期) をとる手段がないため、撮影された画像の同期が保証さ



図 1.2 F T Vの試作機のマルチカメラシステム

れないことである。さらには、カメラを設計通りの位置に配置しきれていないという懸念もある。つまり、受光素子は数 μm という精度で設計どおりに規則的に配置されているのに対して、カメラの位置というのは手でネジ止めする程度の精度である。

このことが原因で、設計通りに配置されていたときに撮影される画像と、実際に撮影される画像とでは無視できないずれが生じてしまう。この問題を解決するために後工程での補正処理が必要となるが、高精度かつ画角が十分な自由視点映像を得るためには、補正処理に頼るのではなく、なるべく設計通りに配置できることが望ましい。

補正処理としては、先述の通りカメラの配置誤差の補正に加えて、カメラ間の色合いの補正も必要になる。カメラの個体差のためにカメラ間の色合いは大きく異なるため、視点変更時（カメラの切り替え時や自由視点画像提示時）に視聴者に不自然さを感じさせなくするには、色合いを揃えるための補正は必要不可欠である。



図 1.3 多視点映像音声撮影システム (上：円弧配置，左下：直線配置，
右下：2次元アレイ配置)

(b) マルチカメラシステムによる FTV データの取得 ～ 21 世紀 COE 多視点映像音声撮影システム～

本論文で使用したシステムとして 100 視点からの映像を取得可能な多視点映像音声撮影システムがある。¹

本システムは、100 台の高精細カメラと 400 個のマイクロフォンを接続し、完全同期²で映像および音声のデータを連続取得することがで

¹名古屋大学では、文部科学省 21 世紀 COE プログラムにおいて平成 14 年に採択された情報系 COE (拠点名：「社会情報基盤のための音声・映像の知的統合」，拠点リーダー：末永康仁 情報科学研究科教授) [13] のコアプロジェクトとして、多視点映像音声撮影システム (100 眼 400 耳システム) を開発した [14]

²1 台のノードにカメラ 1 台とマイクロフォンを最大で 4 台を接続可能である。同期信号はデジチェーンで各ノードに送られ、この同期信号に基づいて各ノードは映像および音声を撮影する。デジチェーンでの同期信号の遅れは各ノード間で $1 \mu s$ 以下であるため、同期誤差は全体でも $99 \mu s$ 以下に抑えられる。この遅れを十分に小さいと判断し、ここでは完全同期と呼ぶ。



図 1.4 撮影した動画像の例. シーケンス *Akko&Kayo* のうち 9 カメラ分のみを表示

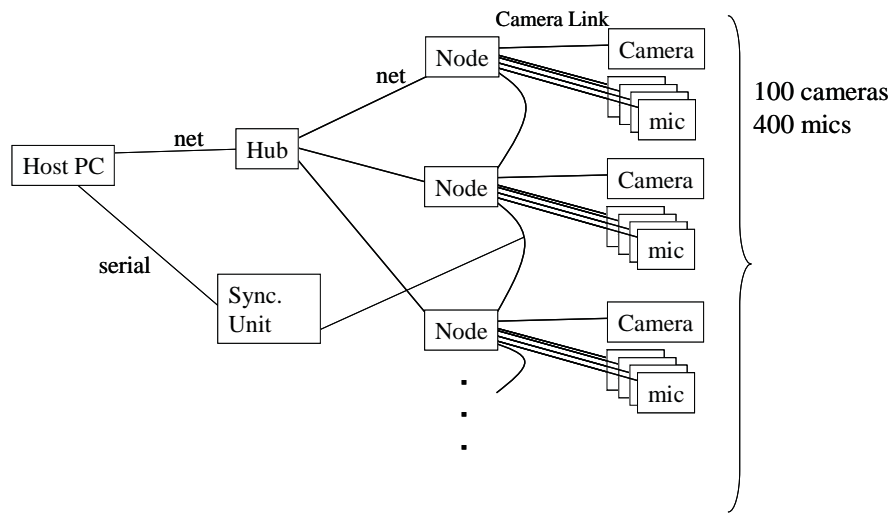


図 1.5 多視点映像音声撮影システムのブロック図

表 1.1 多視点映像音声撮影システムの仕様

基本映像センササイズ	1392Hx1040V	[pixel]
映像フレームレート最大値	29.4118	[fps]
映像出力解像度	8	[bits/pixel]
カラー	Bayer 方式	
映像データ記録時間	約 50	[min]
フレーム同期タイミングのばらつき	1 以内	[μ sec]
映像ピクセル クロックレート	50	[MHz]
音声帯域アナログ入力チャンネル数	最大 4	[ch]
音声帯域アナログ信号の A D変換サンプリングレート	96~8	[kS/sec]
音声帯域アナログ信号の A D変換分解能	16	[bits]

きる性能を有している。

システムの様子と撮影した動画像を図 1.3 と図 1.4 にそれぞれ示す。また、仕様とブロック図を、表 1.1 と図 1.5 にそれぞれ示す。

この装置の最大の特徴は、ハイビジョンサイズの完全同期した動画像を非圧縮で 50 分程度撮影できる点である。これらの特徴を実現している方法を図 1.5 を用いて以下に説明する。まず、ホスト P C が全ノードに "ready for capturing" コマンドを送信する。このコマンドを受けて、各ノードは "capturing status" 状態になる。次に、ホスト P C が同期信号発生装置 (Sync. Unit) に "start capturing" コマンドを送信する。同期信号発生装置はこのコマンドを受けて、同期信号を全ノードに送信し続ける。各ノードは同期信号によってカメラを駆動して画像を撮影し、ノード内にある H D D に画像を記録する。データの転送速度が H D D の書込み時間よりも速いため、各ノードには H D D を 2 台用意してあり、交互にデータを書かせている。撮影を修了するには、ホスト P C から同期信号発生装置に "stop capturing" コマンドを送信する。また同時に、ホス



図 1.6 25 眼コンパクトカメラアレイシステム

ト P C から全ノードに ”Ready” コマンドを送信する。これらのコマンドにより、同期信号発生装置は同期信号を止めて、各ノードは ”Ready” 状態になる。

このシステムの懸念点としては、多視点映像撮影システムも先述のマルチカメラシステムと同様に配置誤差の問題が挙げられる。また、補正処理も同様に、カメラの配置誤差の補正とカメラ間の色合いの補正が必要になることも挙げられる。

(c) マルチカメラシステムによる FTV データの取得

25 眼コンパクトカメラアレイシステム

本論文で利用したもうひとつのシステムとして 25 眼のコンパクトカメラアレイシステム (ProFUSION25)³があげられる。本カメラアレイの特徴は、12mm 間隔で 5×5 の 25 台のカメラがはじめからカメラアレイとして固定されていることであり、可搬性が高い。加えて、カメラの間隔が非

³株式会社 ViewPLUS の商品名

常に狭いことがあげられる。このカメラアレイでは、640×480のVGA解像度を非圧縮にて約25fpsで撮影可能である。また各カメラのシャッターのタイミング等は内部で同期されている。

本システムは出荷時にレンズの位置などは調整されてはいるものの、カメラの向く方向が完全に一致することはなく、幾何補正が必要となってくる。また、各カメラの色合いも、カメラごとに調整可能だが、たとえ同じ指令値を与えたとしても、同じ色合いになることはなく、色補正も必要となる。同期に関しては、すべてのカメラで十分に取れている。

(d) マルチカメラシステムによるFTVデータの取得

～その他のシステム～

文献[15, 16]のシステムでは、安価なWebカメラを多数用いた構成となっている。このシステムは、カメラとMPEG2（またはMPEG1やJPEG）符号化器、ネットワーク機能が付いているいわゆるWebカメラを複数台と、それらを制御するホストPCとで構成された装置であり、多視点動画像を取得できる。先述までのシステムと同様に同期や配置誤差が懸念されると同時に、符号化されたデータを高速に伸張しなければならないという懸念もこのシステムにはある。また、補正処理も同様に、カメラの配置誤差の補正とカメラ間の色合いの補正が必要になる。しかしながら、汎用ハードウェアをそのまま利用できるため、今後のFTVの発展形態として注目されている。

NTTやマイクロソフトでは、それぞれ9台および8台のIEEE1394カメラを用いたシステムを用いている[17, 18]。これらはPoint Gray Research社[19]のマルチカメラシステムを用いて構築されている。これらのシステムではIEEE1394の機能を用いて同期した画像が撮影できるのが特徴である。ただし、複数台のカメラを用いているため、先述までの配置誤差の問題は懸念点として同様に挙げられる。また、補正処理も先述までと同様に、カメラの配置誤差の補正とカメラ間の色合いの補正が必要になる。

(e) その他マルチカメラシステム以外での取得

光線取得のためのマルチカメラシステムについて述べてきたが、マルチカメラを用いずに取得する方法も試みられている。例えば、高解像度のカメラの前面にマイクロレンズアレイを配置するIP (Integral Photography) カメラ [21, 22, 70, 23] がある。動作原理を簡単に説明すれば、ひとつのCCD素子上に小さなレンズ（マイクロレンズ）をアレイ上に並べたマイクロレンズアレイを置くことで、小さいなカメラアレイを構成する方式である。IPカメラの懸念点としては、撮影する画像の視点をマルチカメラのように広げることができないことが挙げられる。また、カメラの解像度がまだ望ましいほど高くないことが挙げられる。しかしながら、今後もカメラが高解像度になっていくだろうことから、非常に有望な光線取得方法と期待されている。

そのほかには、1台の高速度カメラと鏡を組み合わせたシステムが挙げられる [24]。これは、被写体を取り囲むように鏡を配置し、その鏡に反射する光をカメラが撮影方向を変えながらデータを取得するシステムである。カメラの方向は、撮影のタイミングに合わせてガルバノミラーと呼ばれる小さなモータ駆動の鏡を高速に捜査することで変えることができる。このようにして、ガルバノミラーを動かすことで視点変更しながら撮影する装置となっている。撮影する視点位置を時間方向に分割して撮影するため動画像に対応するためには、より高フレームレートで撮影可能な高速度カメラが必要となってくる。

(f) 取得システムのまとめ

カメラアレイ以外のデバイスはひとつのCCD素子にて多視点映像を撮影するため、システムの規模は抑えられるものの、視点数と画像解像度がトレードオフの関係となったり、フレームレートと視点数がトレードオフの関係になると、多視点かつ高解像度の画像をビデオレートで取得することは比較的困難である。

表 1.2 各カメラアレイの比較. 画像サイズの略称は以下の通り.
QVGA(320 × 240), VGA(640 × 480), XGA(1024 × 768)

	(a)	(b)	(c)	文献 [15]	文献 [17]	文献 [18]
カメラ台数	15	100	25	64	9	8
解像度	QVGA	1392×1040	VGA	QVGA	XGA	XGA
同期	×	○	○	×	○	○
圧縮	なし	なし	なし	あり	なし	なし
カメラ間隔	自由	自由	固定, 最小	自由	自由	自由

一方でカメラアレイではシステムは大規模となるが, カメラを多数並べることで高解像度かつ多視点の情報取得が可能となる. 上記で列挙したシステムをまとめたものを表 1.2 に示す. もっとも大規模で高解像度な映像を取得可能なシステムは本論文で使用した本節 (b) の多視点映像音声撮影システムであることがわかる. また, もうひとつの 25 眼システム (c) はカメラ間隔が最小のもっともコンパクトなシステムとなっている.

1.1.3 補正処理

(a) 色補正

マルチカメラシステムの節で述べたように, 各カメラには個体差があり, 全く同じ位置から同じ被写体を撮影したとしても同じに様に写るとは限らない. なぜなら, カメラ内の CCD 素子の特性の違いや, CCD 素子の手前に置かれているカラーフィルタが同一位置に置かれていないからである. そのため, マルチカメラで撮影したシーンを, あるフレームで止め, 各カメラごとに切り替えて描画すると, 色相や輝度の違いにより映像は点滅したようにちらつく. この影響は, 視点切り替え時への視覚的な悪影響だけにとどまらず, 自由視点画像合成処理を行う際にも影

響する。なぜなら、画像合成処理では、近接するカメラ間で画像の相関を取るプロセスがあるからである。このプロセスでは被写体はランバート反射すると仮定され、同じ点から反射される光の特性はどの方向に対しても同一となる。つまり、各カメラに個体差があれば、このランバート反射の仮定が崩れ、自由視点画像合成処理が困難となる。

色補正はいくつか提案されているが、本論文で対象としている大規模(100台)なカメラアレイシステムでの色補正の試みは少ない。そこで、本論文では、大規模なカメラシステムに対して色補正処理を検討する。

(b) 幾何補正

各カメラを厳密に設計通りに配置しようとしても、その設置誤差を無くすことは困難である。カメラの方向の分解能は非常に高く、多視点映像音声撮影システムのカメラの場合、0.1度の回転をさせると、画像中央部では約5pixelほどの変化がある。この精度でカメラをコントロールすることは、ほぼ不可能である。加えてカメラのレンズのマウントにも個体差があり、CCD面に対して、同じ角度かつ同じ位置にレンズを設置することは困難であり、これもまた誤差として現れる。この誤差の補正処理は、撮影した画像を射影変換と呼ばれる処理で変形することで、望んだ位置で取得した画像に近づけることになる。この幾何補正に関しては、2台、3台のカメラシステムでは、古くからレクティフィケーション処理として行われているものの、それ以上のカメラ台数を持つシステムではあまり検討が行われていない。加えてこの場合も同様に、100台という大規模なシステムでの検討は、なおさら少ない。そこで、本研究では、2,3台ではなく4台以上の、最大で100台のカメラアレイシステムで成り立つ補正処理を提案する。

1.1.4 F T Vの圧縮処理

マルチカメラシステムを使ったF T Vでは、撮影した動画像のデータが膨大になるため、効果的なデータ圧縮方法を検討する必要がある。F T Vでは自由視点画像を合成するために、同じ場所を様々な方向から撮影するので、幸いにしてカメラ間の画像には強い相関がある。この相関も利用することで、単純にカメラ毎に圧縮するよりもデータ量を削減できることが期待できる。

複数台のカメラを同時に圧縮する方法は、現在、M V C (Multi-view Video Coding) という名称で盛んに研究されている。中でも、動画像圧縮の国際標準を決定するM P E G / J V T会合での研究は活発で、年4回の会合を定期的で開催して研究している。この会合では、幾つかの要求条件を設けて、その条件を満たす符号化方式を検討している。

このF T Vの圧縮処理は文献 [27] に詳細が記述されている。本論文では、この圧縮処理は考慮せず、入力から出力までの経路は十分に広帯域であるとして扱った。

1.1.5 F T Vの自由視点画像合成

F T Vにおける自由視点画像の合成に目を向けると、光線空間法 [25, 26] を基礎にして、今まで国内外の研究機関で研究されてきており、今なお盛んに研究されている。これらの研究は、Plenoptic Sampling [28] と呼ばれるサンプリング定理を満たすように光線情報を取得することが困難なことに起因している。自由視点画像を合成するとき、C G (Computer Graphics) のようにモデルを用いる場合では、レイトレーシングなどにより容易に合成できるが、F T Vのようにモデルを用いずに光線空間で表現し、かつ光線情報が不足している場合（言い換えると、Plenoptic Sampling を満たさない場合）では、光線空間を補間するための複雑な計算が必要なのである。

光線空間の補間は、多くの場合、撮影した画像データから奥行き情報

を作ることによって実現される。このような奥行き推定の研究はコンピュータビジョンで盛んに研究されているが⁴、この研究結果をそのまま利用することができないという点も、FTVのために自由視点画像の合成に関する研究 [30], [31] が盛んにされている理由として挙げられる。

なぜならば、コンピュータビジョンでは、撮影した位置での奥行き情報推定が研究目的であるのに対して、FTVでは撮影した位置ではない奥行き情報を推定する必要があるため、そのまま利用できないからである。

以下、FTVでの自由視点画像合成の基礎となる光線空間法⁴の概念、マルチカメラシステムで撮影した画像から直交座標系光線空間を構築する方法、自由視点画像の合成方法を順に述べる。

(a) 光線空間法の概念 [32]

光線空間法とは、空間中を伝播する光線の情報を用いて3次元空間情報を記述する手法である。この方法では、3次元空間中の光線情報を、位置 $(x; y; z)$ と $(\theta; \phi)$ の方向から通過する光線の光強度（輝度情報）を $f(x; y; z; \theta; \phi)$ という計5次元の定義域で記述する。ここでは、単純化するために値域である色情報は輝度の情報のみとする。時間変化も考えると、更に時間を加えた $f(x; y; z; \theta; \phi; t)$ という計6次元の空間で記述する。

今、点 $(x_0; y_0; z_0)$ にピンホールカメラを置き、3次元シーンを撮影したとする。これは、点 $(x_0; y_0; z_0)$ を通り、視角内に入る光線の輝度値を記録することに他ならない。このとき、この撮影により記録される光線群は $f(x; y; z; \theta; \phi)|_{x=x_0; y=y_0; z=z_0}$ と表すことができる。すなわち、1回の撮影では5次元空間のうち2次元の部分空間を記録している。全ての x, y, z を動かした視点位置で撮影することにより5次元の全データが記録される。

光の反射、屈折、減衰などの影響はないものとし、光線は直進すると仮定すると、先の5次元空間データを4次元に落とすことができる。4

⁴Middlebury College の Web Site [29] では、多くの奥行き推定法の比較がされ、情報が集積している。

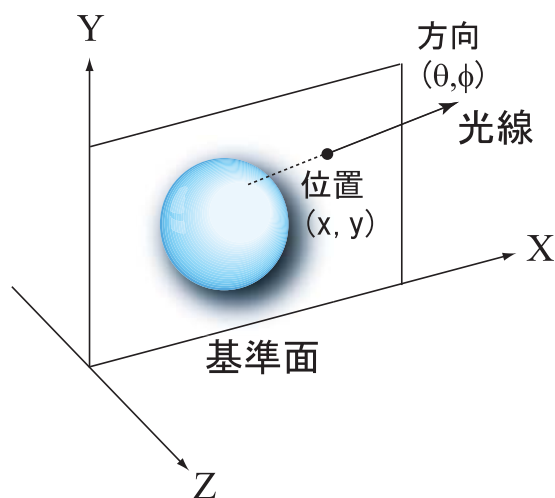


図 1.7 3次元実空間中の座標系

次元への縮退の方法としてはマルチカメラの配置方法により様々な選び方が考えられるが、直線カメラ配置での記録に適した直交座標系による定義、シーンを取り囲むカメラ配置での記録に適した極座標系による定義が代表的である。以下では、直線に並べたカメラ配置による光線空間の例を示す。

(b) マルチカメラシステムで撮影した画像から直交座標系光線空間を構築する方法 [32]

図1.7のような3次元空間及び座標系を考え、この3次元空間内の $x-y$ 平面を基準面として設定する。この基準面を通過する光線は、その通過位置 (x, y) と通過方向 (θ, ϕ) の4つのパラメータにより一意に表すことができる。すなわち、この空間中の光線は4次元空間 (x, y, θ, ϕ) 内の点に対応させることができる。次にこの空間内の各点にそれぞれの光線の輝度 f を対応させることで、この3次元空間内の光線情報を $f(x, y, \theta, \phi)$ と表すことができる。これにより、

$$f(x, y, \theta, \phi) \tag{1.1}$$

$$\begin{cases} -\pi \leq \theta < \pi \\ -\pi/2 \leq \phi < \pi/2 \end{cases} \quad (1.2)$$

というパラメータが得られる．さらに基準面上に物体がなく，基準面を通過する光が通過前後で同じであるとすれば， θ の定義域が狭まり，

$$\begin{cases} -\pi/2 \leq \theta < \pi/2 \\ -\pi/2 \leq \phi < \pi/2 \end{cases} \quad (1.3)$$

となる．これを直交座標光線空間と呼ぶ．このとき，5次元光線空間 $f(X, Y, Z, \theta, \phi)$ は4次元光線空間 $f(x, y, \theta, \phi)$ へ変換することができ，その幾何学的性質より変換式は次式で与えられる．

$$\begin{pmatrix} x \\ y \\ \theta \\ \phi \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\tan \theta & 0 & 0 \\ 0 & 1 & -\tan \phi / \cos \theta & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ \theta \\ \phi \end{pmatrix} \quad (1.4)$$

この式より以下の式が得られる．

$$x = X - Z \tan \theta \quad (1.5)$$

$$y = Y - Z \tan \phi / \cos \theta \quad (1.6)$$

続いて，実空間と光線空間との関係についてさらに述べる．ここで簡単のため，ある y の位置の光線方向が $\phi = 0$ である光線空間の部分空間を考える．するとこのとき得られる光線空間は2次元部分空間 $f(x, \theta)$ となる．また，後の説明のため $u = \tan \theta$ として以降ではこの2次元部分空間を $f(x, u)$ とする．実空間と光線空間との対応関係を図 1.8 に示す．図 1.8 の右図はある一定の高さ y のときの $x - u$ 平面図を示している．したがって，実空間中の一点 (X, Y, Z) を通る光線群が光線空間 $f(x, u)$ で描く軌跡は，

$$\begin{cases} X = x + uZ \\ u = \tan \theta \end{cases} \quad (1.7)$$

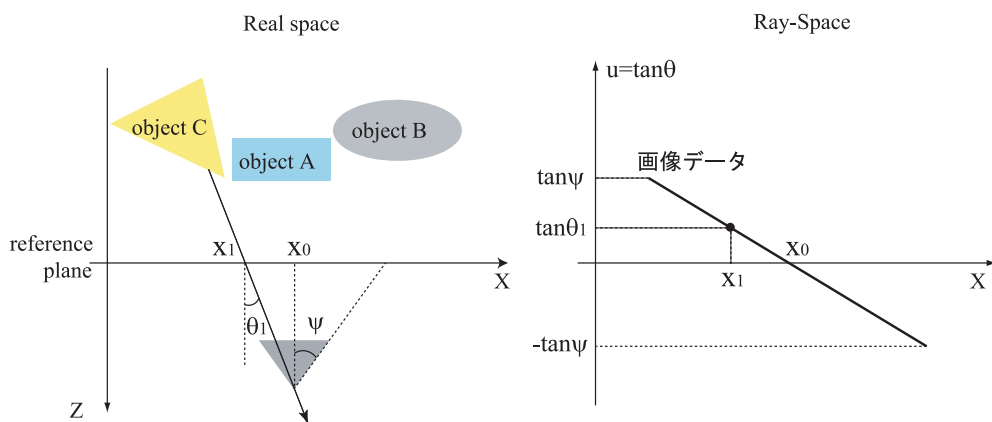


図 1.8 実空間と光線空間の対応



図 1.9 $x - u$ 平面の例，ただし値域は RGB の 3 次元ベクトル

と与えられる。これより、実空間中の一点を通る光線群は、光線空間 $x - u$ 内の平面上にあることが分かる。これはすなわち、ピンホールカメラモデルを仮定した場合の画像情報が、光線空間の平面による断面として与えられることを意味する。カメラを直線に並べた場合、光線空間は画像データを平行に並べることで構築されることになる。

図 1.9 に $x - u$ 平面の例を示す。ただし、この図では、色情報を輝度だけではなく RGB の 3 次元ベクトルで記述されている。この図のように、

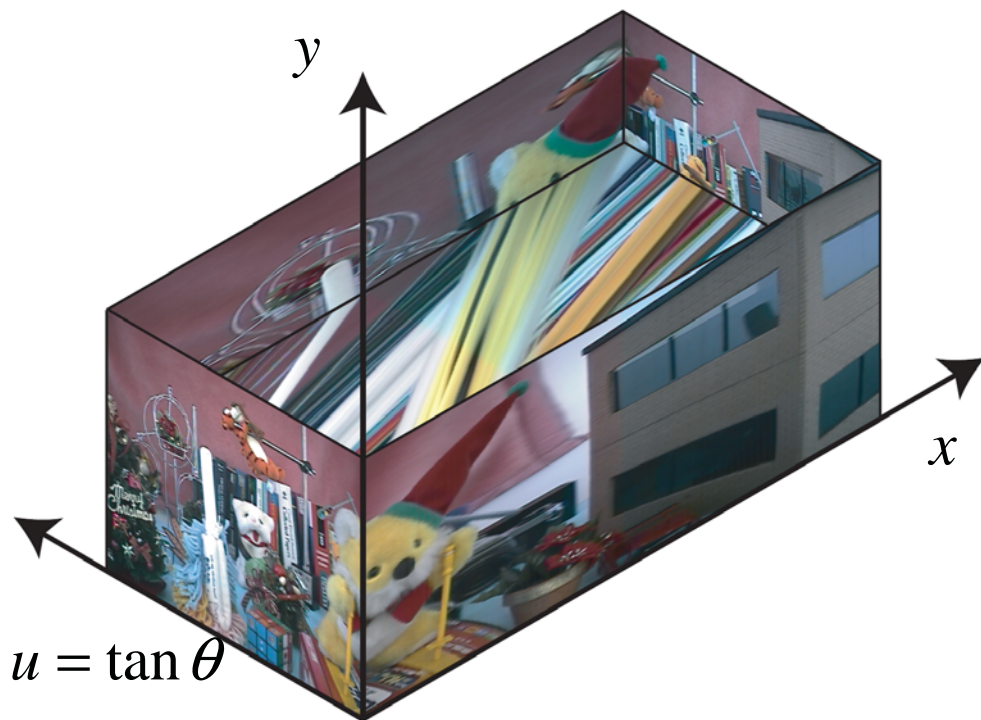


図 1.10 ϕ を無視した場合の 3 次元光線空間

一点の軌跡は直線を描く．この処理を y - ϕ 平面にも適応することで 4 次元の光線空間が記述される．また，光線空間の値域である f は輝度のスカラー値ではなく，RGB の 3 値を持つベクトルへ拡張することは容易である．ここで，4 次元空間は図示できないため， ϕ を無視した 3 次元のカラーの光線空間を図 1.10 に示す．

(c) 自由視点画像の合成方法 [32]

本節では，取得された光線空間を基に自由視点の画像を合成する方法を述べる．図 1.11 に自由視点画像合成における実空間と光線空間の関係を示す（左図：実空間，右図：光線空間）．ここでは，密な光線空間が取得されていると仮定し，図 1.11 の右図の灰色の領域を取得された光線空間とする．図 1.11 の左図で示す実空間中の仮想カメラの画像は，仮想カ

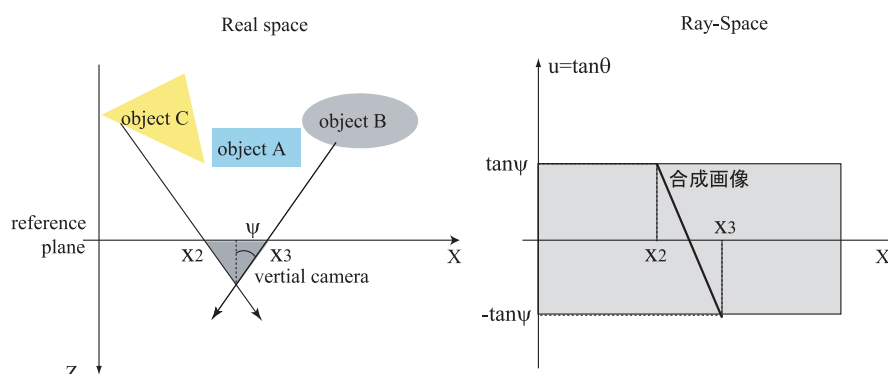


図 1.11 光線空間からの自由視点画像の合成

メラが集約する光線群を取得済みの光線空間から取り出すことで合成される。仮想カメラを実空間中 (X, Z) に仮定すると、この一点を通過する光線群は次式で表される。

$$x = X - Z \tan \theta \quad (1.8)$$

ただし、 $u = \tan \theta$ である。この式 1.8 より、仮想カメラの位置 (X, Z) と、カメラの画素に対応する角度 θ を与えれば、光線を一意に同定できる。また、実空間中の一点を通る光線群が光線空間では平面上にあるという性質から、仮想カメラが集光する光線情報も $x - u$ 平面上では右図のように直線となる。このように、所望のカメラ位置に相当する断面に沿って光線情報を取り出せば、自由視点の画像を合成することができる。また縦方向については、2次元アレイ上にカメラを配置し、縦方向にも同様に直線にカメラを並べることで、 y 及び ϕ も考慮した縦方向の視点移動も実現できる。以上より、光線空間とは空間の光線の情報を位置と方向をパラメータとして一意に表すものである。また、カメラで撮影された画像はそのカメラの位置に集約する光線方向と輝度値を記録するものだと考えると、多眼カメラでの撮影画像を規則的に並べることで光線空間が構築される。自由視点画像の合成は、構築された光線空間から所望の光線情報の切り出しによって実現される。図 1.12 に直線カメラ配置における光線空間取得の様子、図 1.13 に構築された光線空間を示す。さらに、

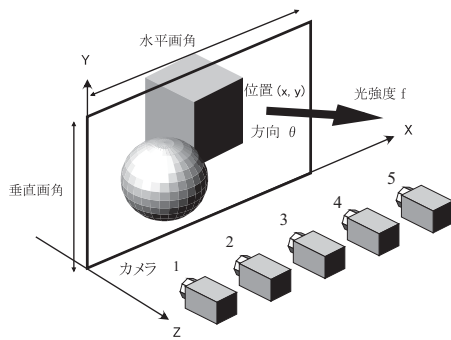


図 1.12 直線カメラ配置における
光線空間取得

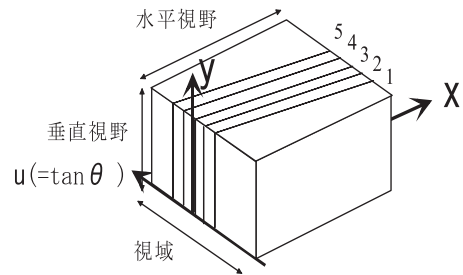


図 1.13 構築された光線空間

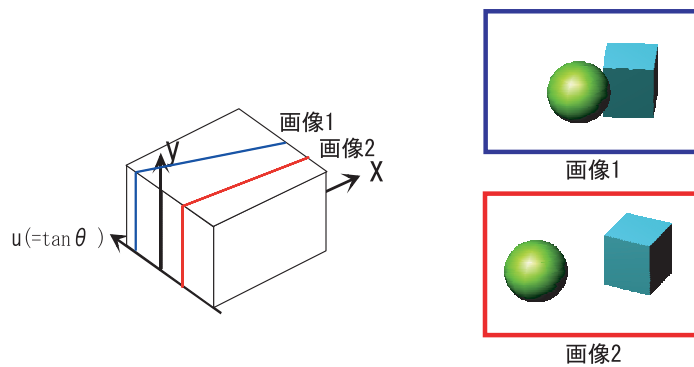


図 1.14 自由視点画像合成

構築された光線空間からの自由視点画像合成の様子を図 1.14 に示す。

1.1.6 FTVの自由視点画像表示

自由視点画像の表示方法としては、通常のディスプレイに1視点の画像を表示するのが現段階では一般的である。しかしながら、さらなる臨場感を求めてパララックスバリアやシンドリカルレンズ(図 1.15)、マイクロレンズアレイ [21] を用いた2視点画像または多視点画像のディス



図 1.15 4眼式立体ディスプレイ



図 1.16 SeeLinder

プレイへの出力も可能である．さらには，図 1.16 に示した SeeLinder[33] や D F D (Depth-Fused 3-D) ディスプレイ⁵[20] のようなディスプレイへの表示も可能である．また，別のアプローチとして，自由視点画像をホログラフィ表示する研究も行われている [34]．

これらのディスプレイやホログラフィにおいて，光線取得から表示までを実時間で行えたという報告は少ない．しかしながら，究極のテレビになりうる F T V を，より臨場感あるものにするためには，これらのディスプレイ技術の開発も非常に重要な課題である．

1.2 研究の目的

ここまでを振り返り，このカメラアレイを用いた自由視点テレビを実現するには，通信路が広帯域であり，圧縮処理と通信処理を考えないとし，出力が一般的なディスプレイであるとすれば以下の 3 つの課題がある．

⁵多くのエッジを持った 2 つの画像を透明な前後 2 面に重ね合わせて表示し，その前後 2 面の画像の輝度比を変化させることで 2 面の間の任意の位置に奥行き位置を知覚できる立体錯視現象を利用した裸眼立体ディスプレイ

1. 色補正
2. 幾何補正
3. 実時間自由視点画像合成

以上のFTVの各プロセスを振り返ると、100台以上の大規模なカメラアレイを用いて、実時間での自由視点画像合成を実現したものは今まで提案されていない。そのため、色補正や、幾何補正など、これだけの規模を対象とした研究も少なく、この規模に合わせて再考する必要がある。

もし、この3つが十分な精度で実現可能であれば、FTVの入力から出力までの一連の流れを達成したこととなる。最終的にFTVのプロトタイプを試作するためにも、これらの技術開発は必須である。そこで、これら3つの課題を解決し、実時間でのFTVシステムを構築することを本論文の目的とした。

1.3 本論文の構成

本論文の構成は以下の通りである。

まず第2章にて色補正処理について述べる。反射特性がランバート反射となるカラーパターンを用い、大規模なカメラアレイにおいても成立可能な手法を提案する。

次に、第3章にて幾何補正処理について述べる。この幾何補正では、カメラ配置に制限の少ないカメラキャリブレーションと呼ばれる手法を用いた手法と、より高精度な補正が可能となる、特徴点の対応関係だけから補正可能な手法を提案する。

そして第4章にて、自由視点画像合成手法について述べる。ここでは、まず、視点依存自由視点画像合成法を提案し、効率的な計算方法を提案する。次にその最適化手法である Multi Pass Dynamic Programming を提案する。

5章にて、これらの結果を用いた、FTVの試作システムを2つ提案する。ひとつは、入力から出力までをリアルタイムにつないだFTVシステムであり、もうひとつは音声を含めて自由視聴点が可能なオフライン処理でのアプリケーションである。最後に6章にて本論文の結論と今後の課題について述べる。

第2章 多視点映像の色補正処理

2.1 緒言

カメラレイシステムにおいて、カメラの色合いが均一となることは重要である。もし不均一なままでは、たとえ視点切替を出力とした場合でもフリッカとして表れ、不自然さを視聴者が感じるであろう。ましてや、自由視点画像合成を行うには、均一な色合いは、演算を簡略化することもでき、大きなメリットがある。

色補正は、各カメラのゲインやルックアップテーブル（ガンマ値など）の変更や絞り、シャッタースピード等をコントロールし、撮影前に調整する場合と撮影した後にカメラの色合いをデジタル処理にて補正する場合がある。もちろん、両者は排他的ではなく、デジタル処理の場合は元情報が劣化するため、出来る限り源信号での調整も望まれる。本論文の色補正では、両方の領域で処理を行った。

2.2 関連研究

マルチカメラシステム用の色補正は、すでに幾つかの手法が提案されており、大別すれば、以下2種類の手法に分類される。

1. 撮影後のデータから補正情報を求める手法
2. 撮影前にカラーパターンを写し、事前に補正情報を求める手法

まず撮影後のデータから補正する場合の例を紹介する。この手法は基本的には画像間対応を取り、その差を減らすアプローチとなる。

Sohnらは、H. 264/AVCを多視点画像の圧縮に使うために、輝度と色相の平均と分散を用いて、ブロック毎にそれらを補償する方法を提案している [39]. この方法は多視点画像の圧縮に特化しているため、各ブロックで色合いが異なるなどブロックノイズが出る. Kawaiらは、線形変換を行う際に、セグメント処理と反復計算を使って係数値を求める手法を提案している [40]. Chenらも、Kawaiらと同様に、線形変換の係数値を反復計算で求める方法を提案している [41]. KawaiらとChenらの両手法に共通しているのは、線形変換で補正する点である.

またFeckerらは、画像間のヒストグラムの差からルックアップテーブル(LUT)を求めて補正する方法を提案している [38]. 対応点を取る必要がなく、比較的簡単に実行可能だが、しかし、この方法では物体の隠れなどのオクルージョンに対応するのは根本的に不可能なため、補正の能力はオクルージョンの量に左右されてしまう. そこで、山本ら [35]により、対応点をSIFT[42]と呼ばれる手法でとることで上記問題を解決した手法が提案されている. この手法では、LUTをガンマ補正まで含めて補正することに成功している.

しかしながらこれらの手法はRGBを独立であるとして解いている. 実際には、各カメラのRGBの色空間自体は独立では無く非独立であり、各色は混ざり合っている. しかしながら、同質のカメラを用いる限り、その混ざり方は決して大きくなく、カメラの各色に混ざり画ないとしてとくことも十分近似解として成立する. より正確に、モデル化するために、各色を非独立に扱おうとすると、線形独立でといた場合に比べて結果が悪化する場合もある. なぜなら、よほどカラフルな物体が写っていない限り、広いRGBの3次元空間を参照する点が少ないため、十分な対応関係が得られず、色相等が失われる場合があるからである.. 色補正とは、対応する点同士の色合い、輝度値を同じものに変換する処理であり、極論すれば全て白や黒に変換する場合がもっとも優れた結果となる. そのため、多次元の行列などで処理をするためには、多くの参照点が必要となる. 山本らの手法はこの問題の解決のため、入力を動画像として多数の参照点を取得していたが、これもまた被写体に依存する.

次に撮影前に、基準となるデータを取り補正する場合の例を紹介する。この手法は基本的にはカラーチャート (図 2.1) を撮影し、その差を減らすアプローチとなる。

Ilieらは3種類の方法を提案している [36]。一つ目は Linear Least Squares Matching と呼んでいるもので、RGBの各チャンネルを独立に線形式で補正する方法である。二つ目は RGB to RGB Transformation と呼んでいるもので、 3×3 行列で補正する方法である。三つ目は General Polynomial Transformation と呼んでいるもので、RGB to RGB Transformation と似ているが、 3×3 行列の代わりに多項式を用いる方法である。彼らによると、「三つ目の手法が適切と考えられるが、二つ目のほうが良い結果がなぜか得られる」とのことである。Joshiらは、Ilieらと少し違い、オフセットも考慮できるように 3×4 行列で補正する方法を提案している [37]。

これらの手法は、カラーパターンを用いることで比較的多くの参照点を取ることで変換しているため、事後に補正するよりも高精度な結果が得られている。

しかしながら、これらの研究は全て2つのカメラでの処理を対象としており、マルチカメラへ適応する場合は1台ずつ2つのカメラのペアとして処理していくに留まる。そのため、大規模なカメラアレイには不向きである。加えて、事前段階からのカメラの補正については全ての処理にて言及されていない。そこで、以下の条件をすべて満たす色補正を行う。

1. RGBの各チャンネルを一緒に扱う
2. 線形変換で補正する
3. カラーパターンボードを使用いて処理する
4. 事前にカメラ間の差を減らす

本手法では、参照カメラと補正対象カメラというペアではなく、理想的な色を定義し、この色に近づくように補正行列を計算するため、多数のカメラをもつ大規模マルチカメラシステムでも有効に働く。

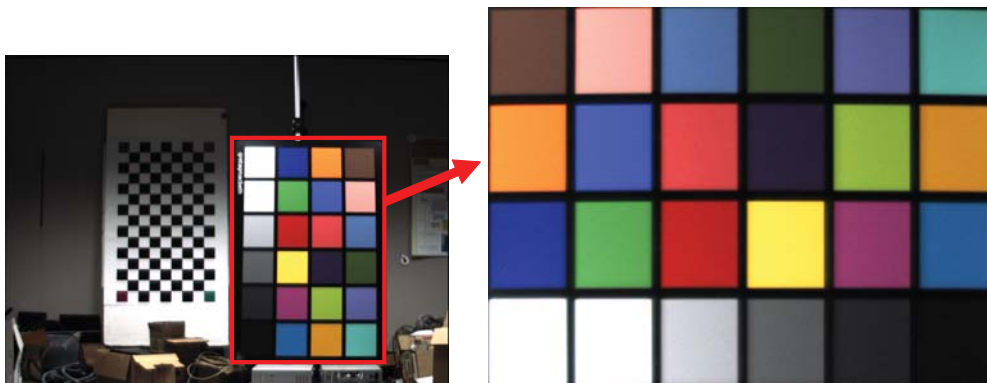


図 2.1 カラーチャート

2.3 提案手法

2.3.1 カメラ特性

本実験では、100眼システムのカメラ PULNiX TMC-1400CL を用いて実験を行う。用いたカメラでは、LUT の設定やゲインのコントロールが電子的に行うことが可能である。また絞りの調整は各個マニュアルで別々に行うことが可能である。

事前準備として、カメラアレイに使用するを 1m 以内に集め、カラーチャートを撮影した。これは、カメラに入射する光の特性が極力同じになるようにするためである。このカラーチャートは、反射特性がランバート反射となるように加工されたものである。チャート上の色は、一般的なカラーチャートの配色となっているが、本手法では、その色がどのような色であるか把握する必要は無く、RGB の各色の色が数種類、カメラに写っていればよい。次に、カラーチャートをカメラの前におき全てのカメラでチャートを抽出した。

まずこのカメラの特性を調べるため、カラーチャートを露光時間を変えながら写し、光電変換の特性を調べたものが図 2.2 である。この図では、カラーパタンのグレイを写したときの生信号である。

輝度が飽和した領域を除けば、ほぼ線形に輝度が変化していることが

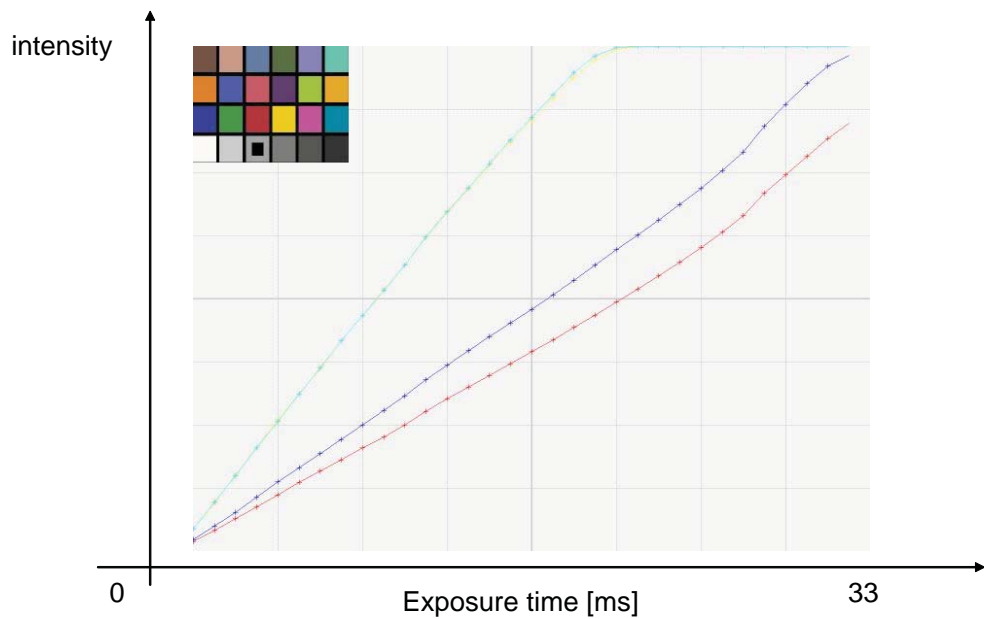


図 2.2 カメラの光電変換特性

わかる。現実に使われているカメラも飽和領域ではこのように異なる挙動をするが、飽和領域を用いないことで対応している。そのため、LUTにて飽和領域の輝度値を打ち切ることで、線形な領域のみを用いることとした。

2.3.2 ゲイン調整

ここでは全てのカメラの誤差が小さくなるように各カメラのゲインの調整を行った。ベクトル数 $size$ である 2 枚の RGB 画像中の対応領域は $(R_0, G_0, B_0), (R_1, G_1, B_1)$ の 2 つの誤差信号 E は単純に RGB 間の距離で定義される。

$$E = 1/size \sum_n \sqrt{(R_0(n) - R_1(n))^2 + (G_0(n) - G_1(n))^2 + (B_0(n) - B_1(n))^2} \quad (2.1)$$

まず，基準となるカメラの適当にえらび，比較対象とする．そして，この基準カメラとの誤差関数が小さくなるように他のカメラのゲインを変化させ，誤差が最小となるゲイン値を選択した．

2.3.3 処理の流れ

事前準備の後，再度カラーチャートを撮影する．この時も，露光時間を変えて各カメラで何枚ものチャートを撮影した．図 2.3 は抽出されたカラーチャートの生信号である．緑が強いのは，RGB 成分のうち緑のフィルタの透過性が高いためであり，このカラーバランスは後に適切な値へと調整される．この処理はホワイトバランス処理と呼ばれ，一般的な画像処理ソフトウェアで実行することが可能である．自由視点画像合成では，各画像の色合いさえ同一となれば実行が可能となるため，カラーバランスや，彩度の調整等の色作りは普段の写真撮影と同様である．

撮影した後画像補正用の行列演算を開始する．

まず各チャートを画像からベクトルへと変換する．1つのカメラごとに 24 色のチャート x 露光時間は変更した回数だけの色情報が得られる (図 2.4)．ここではチャートの各色は対応領域の平均色で代表した．

まず，RGB 画像の色変換を 3×4 の行列演算で定義する．入力色ベクトル V_i から理想的な色ベクトル V_{ideal} への変換は，

$$V_{ideal} = CV_i \quad (2.2)$$

となる．あるひとつの要素 (r, g, b) の射影を同次座標で表せば，

$$\begin{bmatrix} r_{ideal} \\ g_{ideal} \\ b_{ideal} \\ 1 \end{bmatrix} = \begin{bmatrix} c1 & c2 & c3 & c4 \\ c5 & c6 & c7 & c8 \\ c9 & c10 & c11 & c12 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_i \\ g_i \\ b_i \\ 1 \end{bmatrix} \quad (2.3)$$

で表現される．この行列を各カメラで求める事が目的となる．

まず，理想的な色を定めるために，適当な基準カメラを定めその色を理想チャートとする．次に，各カメラごとに変換行列 C を最小二乗法に

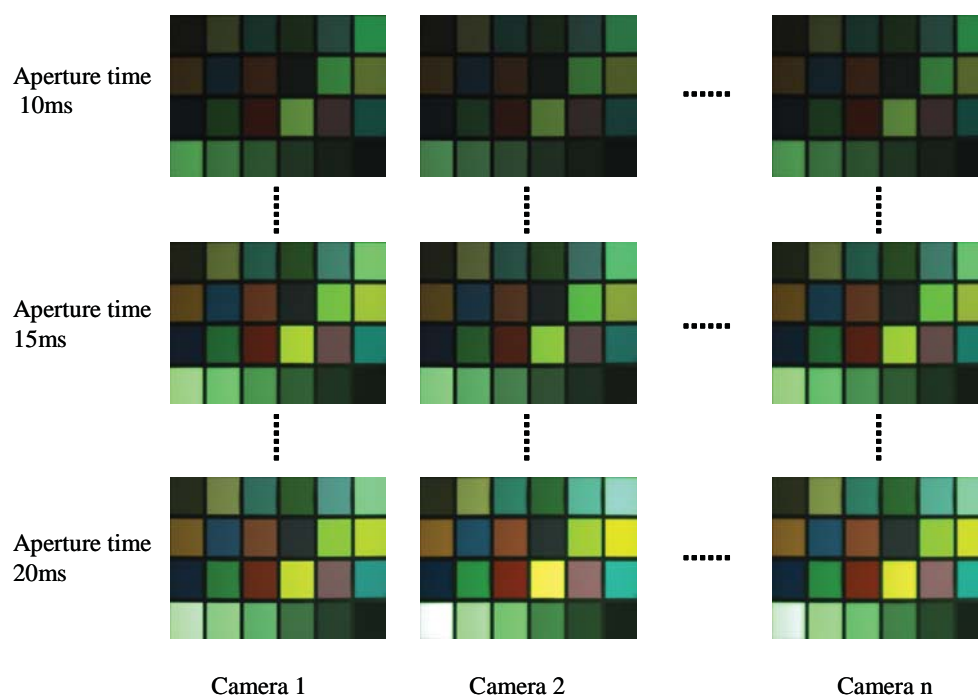


図 2.3 各露光時間でのカラーチャート

て求める。この行列は未知数 12 であるため R G B 3 色かつ 4 点以上の対応が必要となるが、取得点数はカラーチャートや異なる輝度値で撮影しているため十分である。そして、求めた補正行列を各画像に適応する。この次に、各カメラの画像を足し合わせ平均しまたこれを理想チャートとして更新する。この処理を何度も繰り返すことで、補正行列とした。

2.4 実験および結果

2.4.1 実験

CCD やその回路の特性は温度により変わることが知られており、カメラの使用が長くなるとノイズレベルが高くなることや、色合いが変化することが知られている。低ノイズの画像が求められる場合、冷却 CCD カメ

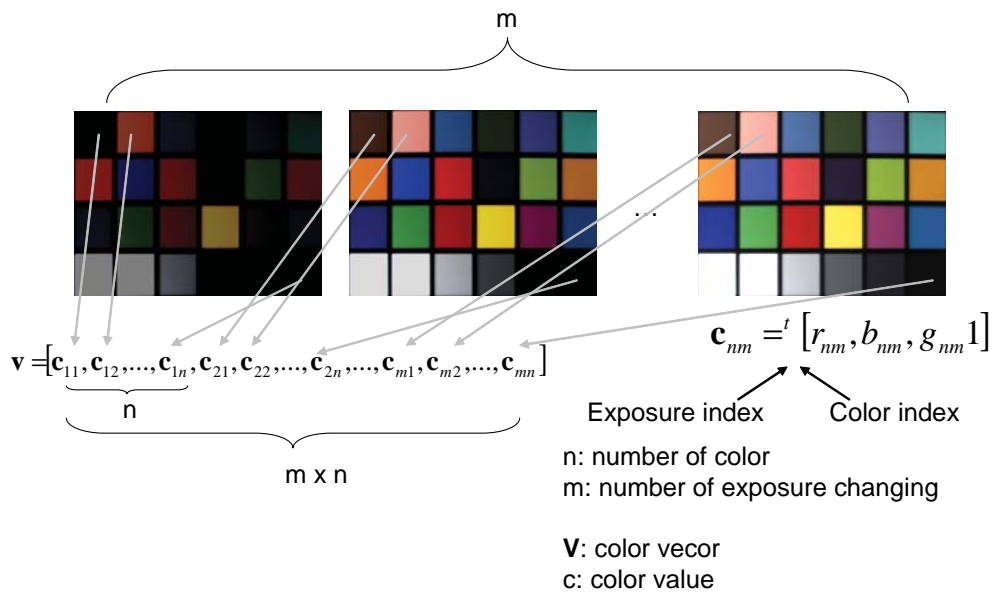


図 2.4 異なる露光時間で撮影したチャートのベクトル化

ラ等が用いられるが、本システムでは無数のカメラを用いるため全てのカメラを冷却することはコスト的に困難である。そのため、カメラを起動してから、10分ほど放置し平衡状態にしてから実験を行った。使用したカメラの台数は80台である。また繰り返し処理の回数は5回とした。また露光時間は3ms, 6ms, 9ms, 12ms, 15ms, 18msの6ポイントで行った。

2.4.2 結果

初期段階のゲイン値調整の結果、エラーは30.1となった。このゲイン値を用いて撮影した画像から補正行列求める。

図2.5は行列による補正の繰り返し回数によるエラーの減少を表している。このようになんども処理を繰り返すことによりエラーは小さくなる。最終的に5回の繰り返し処理にて平均誤差は4.96となり、この値は入力カメラのノイズレベルとほぼ同等である。しかしながら、この繰り返し処理を行うと彩度がなくなり、グレイスケールの画像に近くなっていく

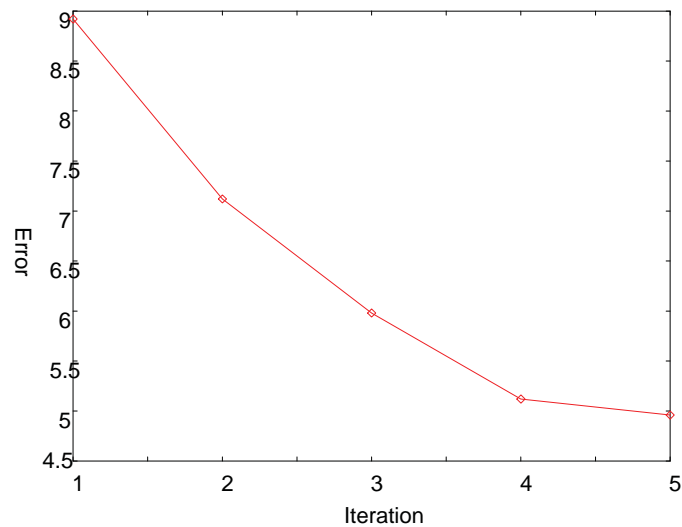


図 2.5 繰り返し処理によるエラーの減少

ことが確認された。そのため、出力画像を確認しながら適切な回数で打ち切る必要があるだろう。だが、カメラアレイの画像補正で要求されることは鮮やかに発色することではなく、各画像の輝度値が一致することである。この鮮やかさの調整は、最終的調整することも可能であり、問題ないとする。

1度のカメラアレイの調整には約5時間ほどかかった。しかしこの補正データは状態が安定した後は恒久的なものであり、1度の手間で済む。

2.5 結言

本章では、カラーチャートを用いて、撮影前からカメラ間の色合い補正を行う手法を提案した。

この手法では、事前処理にて、カメラがの光電変換特性がほぼ線形となるように調整し、またカメラの色合いがもっとも近くなるように、各カメラ別々にゲインをセットした。その処理の後、全てのカメラが、平均色の画像にもっとも最小となる距離へと変換されるように 3×4 の行列

にて線形変換される。カラーチャートはシャッタースピードを変化させ、数種類の画像を撮影することで多くの参照点とした。

実験の結果、ゲイン調整等を行った後の平均誤差は 30.15 であったが、補正行列を適応した結果 4.96 まで抑えることが可能であった。これは使用したカメラのノイズレベルとほぼ同等であり、多視点映像を順番に視点切り替えをして見たときも、画面が点滅することなく、視覚的に良好な結果となっている。また、後述する自由視点画像合成もこの色補正で十分動作することが確認された。

しかしながら、平均色を求める繰り返し処理は、白色ノイズを仮定すれば、無限に繰り返せば色相等の情報欠落していく。そのため、現在では、手動で適切な繰り返し回数を決定していた。この問題を解決するためには、理想的な色である平均色に色相が一定値である、平均輝度値が決まっているなど、なんらかの拘束条件を加える必要があるだろう。だが、この補正行列を適応すれば、たとえ各カメラの色合いがなくなろうと輝度値はほぼ同等であり、全て同じ色合いを持つカメラとしてみなせる。つまり、この補正行列にさらに彩度などをコントロールする行列を全てのカメラに共通に適応することで色合いがつぶれる問題は解消可能である。また、本論文では検討しなかったが、画像の周辺部はレンズによる効果により光量が減衰し暗くなっている。今後は、これらの情報を考慮し問題解決を行っていくことが必要であると考えられる。

第3章 多視点映像の幾何補正 処理

3.1 緒言

光線空間 [26] や Light Field Rendering[43] による自由視点画像合成を行うためには，入力多視点映像を撮影するカメラレイには，カメラの個体差や設置誤差等がない理想的な設置が望ましい．もし，理想的な設置がされていない場合は，カメラの光軸の位置やレンズ歪みなどの内部変数や位置・方向の外部変数を用いて自由視点画像合成を行う手法 [31] と，なんらかの手法により多視点映像を理想的な光線空間へ補正したものを入力として合成する手法がある [44]．前者は合成処理を直接行える一方で後者は，いったん，補正というプロセスを必要とする．しかし，補正後は入力データ構造が整列されるため，補正処理にかかる時間以上に自由視点画像合成の計算時間が向上し，また画質等が向上する利点がある．本論文では，補正した入力画像から自由視点画像合成を行うことを目的としている．

この補正には，2通りのアプローチがある．一つ目は，カメラの内部変数や位置・方向等のカメラパラメータを求めるカメラキャリブレーションを行い，その結果が理想的な配置となるように投影する手法である．二つ目は，カメラパラメータを用いずに，カメラレイに写る対応点の動きが理想的な形になるように変形する手法である．前者はメトリックキャリブレーション・レクティフィケーション，後者はノンメトリックキャリブレーション・レクティフィケーションと呼ばれている．

メトリックキャリブレーションの特徴は以下の用である．

- カメラのパラメータを求めることが可能
- カメラの物理位置からフィッティングするため、投影後の誤差の最小化ではない.
- さまざまなカメラ配置について一般的に適応可能
- 精度限界がキャリブレーション精度に依存するため、カメラ間隔が狭く高精度なキャリブレーションが必須な場合は不向き

つまり、精度は低いですが、大規模カメラアレイに限れば柔軟な適応が可能な方法である。また、ノンメトリックキャリブレーションの特徴は以下の用である。

- カメラのパラメータを求めることが不可能
- 対応する特徴点の位置を最適化するため、画像上の誤差はメトリックベースのものよりも高精度
- 全てのカメラ配置に対して一般的に当てはまる補正方法がない
- 精度限界が特徴点の座標取得精度に依存するため、キャリブレーションベースのものよりもカメラ間隔が狭い場合も適応可能

つまり、使用法が限定されるが、どんなにカメラアレイの間隔が狭くても精度の高い補正が可能な方法である。

本論文では、さまざまなカメラ配置、カメラパラメータの有無、精度の要求にこたえるため、両フレームワークについてアルゴリズムを提案し、画像を補正する。

3.2 関連研究

キャリブレーションを用いた補正は古くから研究されており [54]、二眼ステレオの平行化（レクティフィケーション）が代表的な研究である。図

3.1は二眼ステレオの例であり、任意に配置されたカメラにおいて、二つのカメラの画像面が光軸を結ぶ直線（ベースライン）に対して平行かつ、回転がベースラインと同様の向きを持つ状態である。また、この処理は、三つのカメラの場合も適応可能である（図3.2）。任意におかれた3台のカメラを二眼ステレオペアが直行しておかれたように変換する処理である。このような補正はノンメトリックな手法でも研究されており、ステレオの場合 [56] や三眼ステレオの場合 [63] など高精度に補正が可能となっている。加えて、レンズディストーションの除去 [55] も可能である。

さて、ここでカメラの配置を直線やアレイなどの4台以上から構成されるカメラアレイの場合はどうであろうか？もしカメラが4台以上ある場合、カメラの位置に制約が生まれ、設置誤差がある場合は理想的な状態へ完全に変換することは不可能である。なぜなら、画像補正で修正可能な条件はカメラの方向と内部の状態だけだからである。二眼、三眼の補正には位置の補正は含まれていない。今までは、三次元構造を復元するためには、カメラのパラメータさえそろっていれば可能であったため、このような問題は主には取り扱われていなかった。そのため、Image based rendering のように入力画像の位置までも必要となると、カメラキャリブレーションをベースとした補正はカメラの位置に誤差はほとんどないと仮定し、回転と内部変数の誤差のみを打消す処理を行っていた。位置の補正はそれぞれ自由視点画像合成となり、自由視点画像合成のための補正のために自由視点画像合成を行うとなり本末転倒となる。

しかしながら、カメラの位置が微小にずれただけであればほぼ正しい画像に補正することが可能である。そのため、ノンメトリックな手法は研究されてきた。まず、直線のカメラアレイの場合は、松本らにより提案されている [58]。この手法では、基準となるカメラを3台のカメラを選択して補正し、残りを3つの対応関係から解決する。二次元カメラアレイは、VaishらのPlane+Parallaxを用いた手法 [57] が提案されているが、これはアフィンカメラモデルで近似するため、カメラアレイと既知パターンが完全に平行に設置された状態でなければ成立せず、ずれた場合は近似となる。そのため、精度の高いレクティフィケーションを行うために

は、レクティフィケーション用のパターンを厳密に配置する必要がある。(パターンがピッチ、ヨーの角度に1度のオーダーで外れてもいけない。)その他の手法として、Dengらにより2次元カメラアレイの画像をレクティフィケーションする手法が提案されている。この手法はノンメトリックな手法に属し、精度の高いレクティフィケーションが可能一方で、特殊な条件下でしか成立しない。その条件とは、「1台のカメラ」を、「機械」を用いて、「アレイ状に厳密に動かして」シーンを撮影することである。つまり、この手法を用いる限り静止画像しか撮影することが出来ない。

そこで本論文では、さまざまなカメラ配置への適応するために、キャリブレーション結果を用いる補正方法を、また、二次元カメラアレイを対象としたノンメトリックな補正方法を提案する。

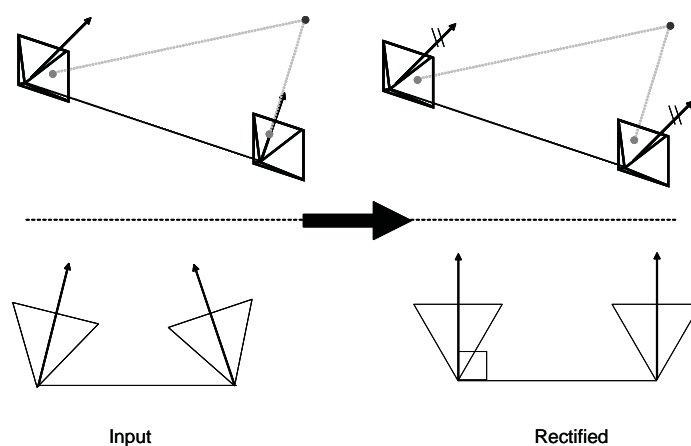


図 3.1 二眼ステレオの補正

3.3 カメラキャリブレーションによる画像補正

3.3.1 ピンホールカメラモデル

本論文では、カメラをピンホールカメラモデルで仮定する [61].

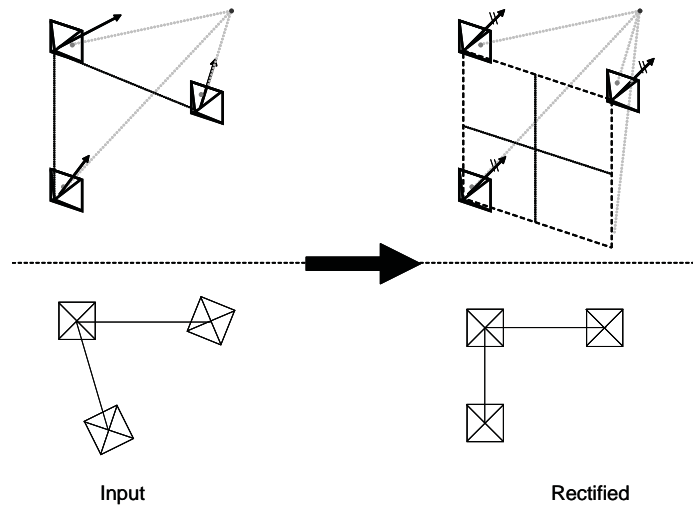


図 3.2 三眼ステレオの補正

$\tilde{M} = [X, Y, Z, 1]^T$ を世界座標での点とし、 $\tilde{m} = [u, v, 1]^T$ をその点が射影行列 P で画像に射影される点とする。ピンホールカメラモデル（図 3.3）では \tilde{M} と \tilde{m} は次式で関係付けられる。

$$\tilde{m} \simeq P\tilde{M} \quad (3.1)$$

ここで、 \simeq は定数倍の不定性を表す。また、 3×4 行列である P は射影行列と呼ばれる。射影行列 P は次式で表される。

$$P = K \begin{bmatrix} R & -RT \end{bmatrix} \quad (3.2)$$

ここで、 3×3 行列 R はカメラの回転を表し、3次元ベクトル T はカメラの位置を表す。これらは、カメラ座標系と世界座標系を関係付ける外部パラメータと呼ばれる。また、 K はカメラ座標系から画像座標系へ変換する内部パラメータと呼ばれる。内部パラメータ K は焦点距離 f_u, f_v 、せん断パラメータ γ 、画像中心 (u_0, v_0) で構成される上三角行列であり、

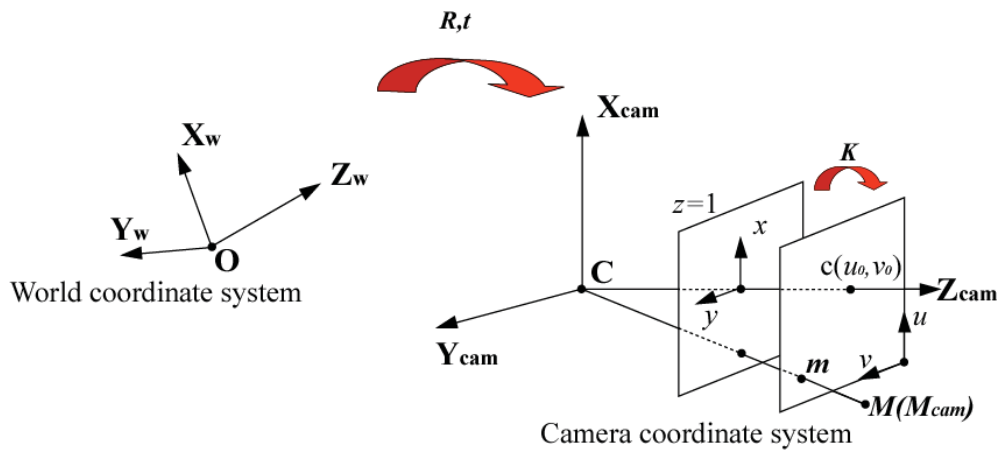


図 3.3 ピンホールカメラモデル

次式で与えられる.

$$K = \begin{bmatrix} f_u & \gamma & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

なお、本研究では、レンズディストーションは除去済みであるとして考慮しない [55].

3.3.2 カメラキャリブレーション

カメラキャリブレーションとはカメラの内部変数や位置、方向を求める手法である.

カメラキャリブレーションの古典的なアプローチは、三次元位置が既知の参照点を多数観測し、それらの投影像を用いてキャリブレーションを行う方法である [46]. 通常は、そのような参照点が配置されたキャリブレーション用の 3 次元参照物体を用いる. この方法では参照点の三次元座標と投影像の 2 次元座標を数十点对応付けることでカメラパラメータを推定する. 参照点の三次元座標を正確に設定できる場合、この手法はカメラパラメータを高精度かつ安定に推定できる. しかし、多視点カメ

ラにおいて、すべてのカメラが観測できるように参照点を三次元空間中で高精度に設定することは大きさの制約などから困難である。

このような古典的手法の改善策として、二次元位置が既知の参照点を描いた平面を用いて簡便にキャリブレーションする方法も提案された [45],[47]。この方法では平面を複数の異なる位置に動かして参照点を撮影し、平面と画像面との射影変換を計算してカメラパラメータを推定する。平面は自由に動かすことができ、その運動を知る必要はなく、参照点の位置を2次元平面上のみで指定すればよいので、参照点の3次元位置を与えなければならない古典的キャリブレーション方法に比べて作業の手間は大幅に削減される。その簡便さから単眼カメラに関しては広く用いられており、近接配置の多視点カメラにおいてその精度を向上させる手法もある [48]。この他にも平行線群を印刷した平面を用いて多視点カメラをキャリブレーションする手法 [49] や円図形を印刷した平面を利用する手法 [50] も提案されているが、基本的には [45] の手法を基本としており、精度に大きな違いはない。

これらのなんらかの既知物体を必要とする手法に対して、既知の参照点や参照物体を必要としないセルフキャリブレーションといった方法が近年研究されている [51],[52],[53]。これは、三次元位置が未知の参照点を多視点カメラで撮影し、その投影像のみからカメラの内部パラメータと外部パラメータの両方を求めるものである。セルフキャリブレーションは極めて制約の少ない柔軟な手法であり、運用に際しては、例えば参照点としてLEDを使用し、LEDを光らせたと同時に各カメラがその画像を取得するといった作業を繰り返せば良い。しかし、セルフキャリブレーションでは2次元の画像情報のみから3次元情報を記述するカメラパラメータを推定する作業であるので、利用できる制約に比べて推定すべきパラメータが多くなり、ノイズに関してしばしば数値的な安定性に欠けることも知られている。

キャリブレーションに要求することは以下3件である。

1. 特別な装置を必要としない

2. 高精度な結果

3. カメラ台数が多いため簡便な手法が望ましい

3つの要件を考えたとき、まず、精度の低いセルフキャリブレーションは除外される。また、既知物体を用いるキャリブレーションの精度は測定したい空間に占める既知物体の面積や体積に依存するため極力大きなものが望ましい。しかし、大きな体積のものを3次元空間に設置することは困難であり、また、全てのカメラに共通に写る空間は多くはないため、物体を大きく写すことも困難である。そのため、三次元パターンを用いるものは条件から除外される。以上から二次元パターンを用いる手法 [45] 採用する。この手法では、外部変数の推定には全てのカメラに共通に写る既知物体が必要であるが、内部変数の推定は各個に行うことが可能である。また、複数毎のパターンをさまざまな角度でカメラに写すことで精度が向上する。そのため、大きな平面パターンと小さな平面パターンを併用してキャリブレーションを行うことにした。比較的遠方には大きな平面パターンを固定して撮影し、また近影には、小さなパターンの角度を変えながら複数毎撮影を行いカメラキャリブレーションを行う。

3.3.3 回転による画像補正

二次元画像の射影は以下の行列の射影で表現できる。

$$\mathbf{m}_2 \simeq \mathbf{H}\mathbf{m}_1 \quad (3.4)$$

ここで、 $\mathbf{m}_1, \mathbf{m}_2$ は各カメラの画像座標を $\mathbf{m}_1 = (u_1 \ v_1 \ 1)^\top, \mathbf{m}_2 = (u_2 \ v_2 \ 1)^\top$ と同時座標で表現したものである。行列 \mathbf{H} は Homography(射影変換行列) と呼ばれる 3×3 行列である。

カメラの回転、内部変数を変更するためには、この射影変換行列を適応すればよい。射影変換行列 \mathbf{H} は、回転行列 \mathbf{R} 、並進ベクトル \mathbf{t} および変更前後の内部パラメータ $\mathbf{K}_1, \mathbf{K}_2$ により求められる(式 3.5)。

$$\mathbf{H} = \mathbf{K}_2 \mathbf{R} \mathbf{K}_1^{-1} \quad (3.5)$$

古典的なカメラキャリブレーションによる画像補正では、カメラの全てのパラメータが計測されているため、全てのカメラの内部変数を均一にし、目的の方向を向けることで完了する。

3.3.4 提案手法

提案手法は、理想カメラアレイの位置フィッティング、理想内部変数の導出、実対応点と理想対応点の投影という流れで行う。今論文では直線カメラ配置を例に説明するが、二次元構造や円形配置、ドーム状、しいては任意なカメラの配置でも可能である。これは、初期ステップである、カメラアレイの当てはめ問題は、3次元の点列を理想的な点列もしくは関数に近似する問題であるため、その後段の処理はカメラ配置に依存しない処理であるためである。

まず直線配置の理想カメラアレイの条件は、

- カメラ焦点がベースライン上にある
- 光軸はベースラインに垂直に配置
- 全てのカメラが等間隔に並ぶ

である。この条件を満たすため、カメラキャリブレーションでえられた n 台のカメラ ($i = 1, 2, \dots, n$) の位置 \mathbf{T}_i^{real} からベースラインを主成分分析で計算する。

$$\mathbf{T}_g = \begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \frac{1}{n} \begin{bmatrix} \sum_{k=1}^n x_k^{real} \\ \sum_{k=1}^n y_k^{real} \\ \sum_{k=1}^n z_k^{real} \end{bmatrix} \quad (3.6)$$

カメラアレイの重心 \mathbf{T}_g は各位置の平均で計算され、この共分散行列 \mathbf{V} は以下であらわされる。

$$\boldsymbol{\tau} = \begin{bmatrix} {}^t\mathbf{T}_0^{real} - {}^t\mathbf{T}_g \\ {}^t\mathbf{T}_1^{real} - {}^t\mathbf{T}_g \\ \vdots \\ {}^t\mathbf{T}_n^{real} - {}^t\mathbf{T}_g \end{bmatrix} \quad (3.7)$$

$$\mathbf{V} = \boldsymbol{\tau}^t \boldsymbol{\tau} \quad (3.8)$$

カメラアレイの主軸を見つけるために、 \mathbf{V} を固有値分解すると、固有値 λ 固有ベクトル \mathbf{w} の第一、第二、第三固有ベクトル $\mathbf{w}_i : i = 1, 2, 3$ は以下になる。

$$\mathbf{V}\mathbf{w} = \lambda\mathbf{w} \quad (3.9)$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \mathbf{w}_3 \end{bmatrix} \quad (3.10)$$

この行列 \mathbf{W} はカメラアレイの回転を表すことになる。加えて、理想的なカメラアレイでは各カメラは同じ方向を向くため、この値が各カメラの理想的な回転行列 \mathbf{R}_{ideal} となる。

$$\mathbf{R}_{ideal} = \mathbf{W} \quad (3.11)$$

ここで理想的なカメラ配置では位置ベクトルはある間隔 $step$ で中心座標 $center$ から分布している。まずこの理想的な位置ベクトルを仮に $\mathbf{T}'_i : i = 1, 2, \dots, n$ とすると

$$\mathbf{T}'_i = \begin{bmatrix} step(i - center) \\ 0 \\ 0 \end{bmatrix} \quad (3.12)$$

で表現される。実際には、回転や重心があるため、行列 \mathbf{R}_{ideal} と、重心 \mathbf{T}_g を用いて理想的な各カメラの位置を表すと

$$\mathbf{T}_i^{ideal} = \mathbf{W}\mathbf{T}'_i + \mathbf{T}_g \quad (3.13)$$

となる。

しかし、直線のカメラアレイの場合、カメラの yaw 軸に関する自由度を持っており、どのように回転であっても補正の条件を満たしている。たとえば、理想的なカメラアレイ上のカメラが全てが真上を見ていたと仮定しても条件を満たすのである。それでは都合が悪いため、yaw 軸に関しては、各カメラのキャリブレーション結果の yaw の回転軸の平均値を用いた。ただし、二次元アレイ、円形や、ドーム型のカメラ配置では回転の自由度がないため一意に定まっている。

上記で配置されたカメラアレイは、どのような内部変数を持っていても、全てのカメラにおいて同じであれば補正された条件を満たす。そこで、本論文では理想的な内部変数 \mathbf{K}_i を、各カメラでキャリブレーション結果の焦点距離の平均 f_u, f_v と、光軸中心の平均値 u_0, v_0 で表す。また、せん断成分は 0 となるように設定した。

$$\mathbf{K}_i = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

これで、理想的なカメラの内部変数、回転、位置が全て求まったことになる。

ここで初めて、実カメラと理想カメラへと変換が可能となる。

i 番目のカメラの理想的な投影行列 \mathbf{P}_i^{ideal} は、

$$\mathbf{P}_i^{ideal} = \mathbf{K}_i^{ideal} \begin{bmatrix} \mathbf{R}^{ideal} & -\mathbf{R}^{ideal} \mathbf{T}_i^{ideal} \end{bmatrix} \quad (3.15)$$

であり、カメラキャリブレーションで用いた二次元パターンを用いれば、既知の 3 次元座標 \mathbf{M} と、理想投影行列により、

$$\mathbf{m}^{ideal} = \mathbf{P}_i^{ideal} \mathbf{M} \quad (3.16)$$

と、理想的な画像座標 \mathbf{m}^{ideal} が定まる。ここで、 (i, j) 番目にある格子模様の二次元パターンの 3 次元座標は、そのパターン上の平面が $Z = 0$ であ

る平面とすれば、各点は間隔 $cstep$ で等しくなるため、

$$M = \begin{bmatrix} cstep \times i \\ cstep \times j \\ 0 \\ 1 \end{bmatrix} \quad (3.17)$$

となる。

実際に撮影した実カメラの画像座標 m^{real} は理想的には座標 m^{ideal} に対応するので、補正行列 H は

$$m^{ideal} = Hm^{real} \quad (3.18)$$

で求まる。これは、4点以上の対応関係があれば線形に解く事が可能である。

この行列を全てのカメラで求めることで補正処理は完了する。

3.3.5 実験

80台のカメラ5cm間隔で直線にならべ、4mのカメラアレイ作成した。図3.4に実際に設置したカメラアレイを示す。各カメラは水平方向に約26度の画角を持ち、解像度は1392×1040のものを用いた。カメラの設置は金属性の剛体に5cm間隔で穴をあけ、ねじで完全に固定することで行った。

まず各カメラでカメラキャリブレーションを行った。カメラアレイから、8.2m先に水平方向4m、垂直方向2mのパターンを設置し、そのパターンの格子は10cm四方とした。また、近接撮影用のパターンは、水平方向1.5m、垂直方向0.8mのパターンを20回異なる角度で撮影した。実際に用いたパターンを図3.6、図3.5に示す。

キャリブレーションの結果、各カメラはほぼ直線上に配置され、カメラ間距離の平均値は4.96cmとなった。実際にカメラキャリブレーションを行い、その結果を図示したものは図3.7であり、赤が実カメラ、紫が理



図 3.4 実際に用いた直線のカメラアレイ (80 台)

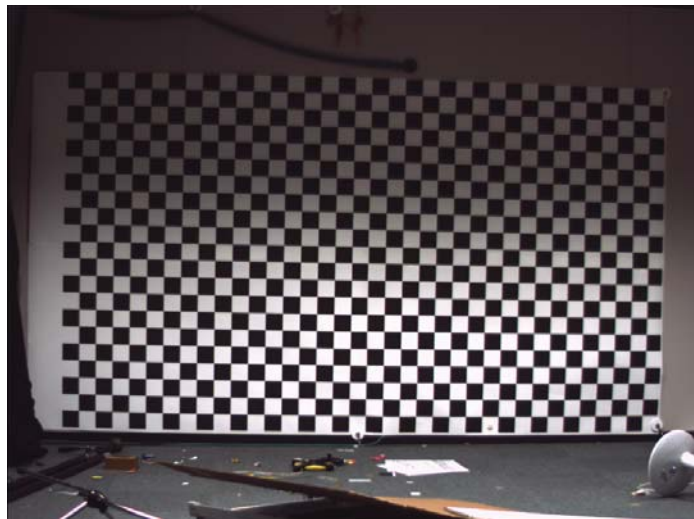


図 3.5 大きな平面パターン

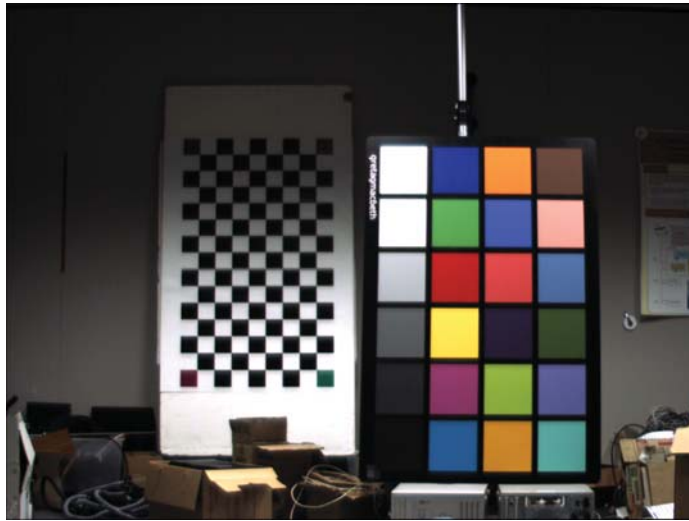


図 3.6 小さな平面パターン

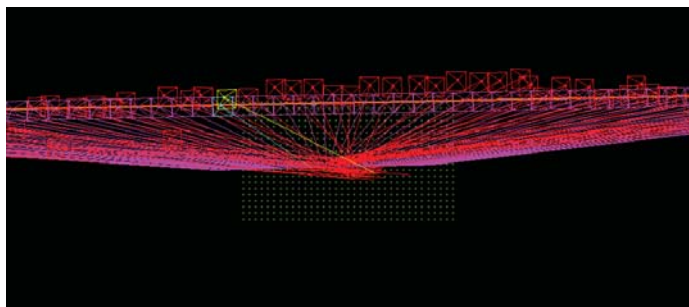


図 3.7 キャリブレーション結果

想的な配置のカメラである。また、一部の黄色は特定の実カメラであり、水色は、それに対応する理想カメラとなっている。緑は、遠方に配置した大きなパタンの格子点である。このように若干のずれはあるものの、カメラキャリブレーションを行うことが可能であった。理想的なカメラ位置からの距離は平均で、0.25cm であり、最大のずれは 3.5cm であった。

この結果を用いて画像補正を行った。入力画像を図 3.8、補正画像を図 3.9 に示す。ここではカメラアレイ中ほどのカメラ 20, 40, 60 を図示した。左の人物の指、下のボタンに着目すると、補正前は、まったく水平

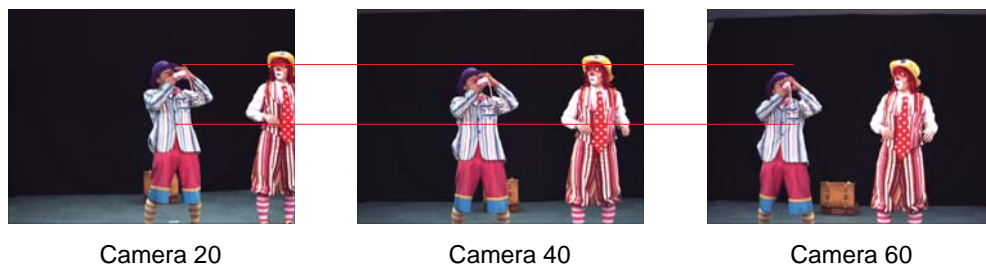


図 3.8 入力画像

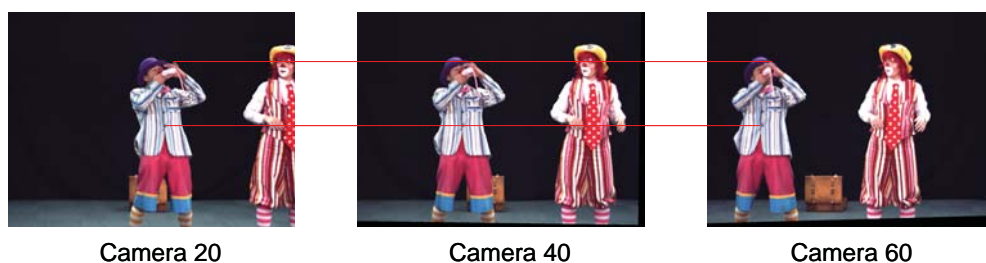


図 3.9 補正画像

線上に乗っていなかったのが、補正後に水平線上に乗っていることがわかる。また、3つのカメラ全てで水平線上に対応点が乗っており、補正が出来ていることがわかる。

また、全てのカメラに写る、ある対応点の v 座標の平均値からのずれの絶対値は平均で 0.42pixel であった。

しかしながら、文献 [58] のノンメトリックな手法で補正したところ、ずれの絶対値は平均で 0.35pixel であった。この直線のカメラ配置に特化した、ノンメトリックな補正手法を用いたほうが、誤差は小さくなった。

3.3.6 キャリブレーションによる画像補正のまとめ

ここまでは、カメラキャリブレーションを用いた幾何補正の手法を提案した。例として、直線カメラ配置の場合の補正を示した。二次元パターンを用いたキャリブレーション精度を向上させるために、大きなパターンと

小さなパターンを用いてキャリブレーションを行い，補正を行った結果，誤差は 0.42pixel となった．誤差は 1pixel 以内に対応点が納まっており，本手法は実用に耐えうるものである．

一方で，文献 [58] のノンメトリックな補正手法を用いれば，その誤差は 0.35pixel となり，本キャリブレーションを用いた手法よりも高精度なものとなっている．これは，特徴点を直線上に乗るように直接最適化するノンメトリックな手法であるからだと考えられる．しかしながら，ノンメトリックな手法は補正に特化するため，カメラの焦点距離や光学中心などカメラパラメータは未知のままである．この場合，物体の奥行き方向に自由度が残り物体の 3 次元復元は完全には行うことが出来ない [61]．そのため実際の物体の奥行きが必要な場合にはこの手法は適さない．

以上から，光線空間による自由視点画像合成に特化した場合はカメラパラメータ等は必要なく，ノンメトリックな手法が有利ではあるが，物体の実際の大きさ等が必要なときは本手法を用いるなど，状況に応じて使い分ける必要があるであろう．また，本手法では，カメラアレイのフィッティングを変えることで直線ではなく，その他さまざまな手法に適応可能である．いまだノンメトリックな手法は未検討である，円形配置，ドーム配置でのカメラアレイでは本手法を適応することがよいだろう．

次節では，二次元カメラアレイに対する，ノンメトリックな補正手法を提案する．

3.4 ノンメトリックな二次元カメラアレイ画像補正

キャリブレーション結果を用いた画像補正は、カメラの主点位置を関数にフィッティングすることで行われるのに対して、ノンメトリックな画像補正は投影された対応点列の動き（軌跡）を関数にフィッティングすることになる。結果、多くの場合においてノンメトリックな補正方法のほうが精度は高くなる。しかしながら、カメラアレイに対するノンメトリックな補正手法は広くは検討されておらず、特に二次元カメラアレイに対する補正手法の検討は、いまだ十分とは言えない。そこで、以下では、光線空間法による自由視点画像合成には必須である二次元カメラアレイへのノンメトリックな補正手法を提案する。

3.4.1 理想状態と対応点軌跡

整列状態での三次元空間中の性質

図 3.10 は、 w_c 行 h_c 列である $w_c \times h_c$ (この場合 $w_c = 3, h_c = 3$) のアレイ状に配置された、 $w_c \times h_c$ 台の理想的なカメラアレイを 3 次元図示したものである。本稿では、カメラの位置（カメラ座標）を指定してカメラを指し示す必要がある時には、カメラ行 $i : i = 1, \dots, w_c$, カメラ列 $j : j = 1, \dots, h_c$ の添え字を示して、 $Camera_{i,j}$ のように記す。

さて、二眼ステレオカメラの平行化では、二つのカメラの光軸が同方向かつ、ステレオカメラのベースラインと直交する軸となり、その内部変数が同一となる。それに対して、本稿の目的とするカメラアレイレクティフィケーションは、 n 台全てが上記を満たし、加えて、全カメラの光軸の向く方向ベクトル f が全カメラの焦点が乗るアレイ平面 P の法線ベクトル p に水平でなくてはならない。つまり、各カメラの内部変数 $K_{i,j}$ が全て同一で、全てのカメラの姿勢 $R_{i,j}$ の方向ベクトルがカメラアレイ平面の法線ベクトル p と一致した状態である。

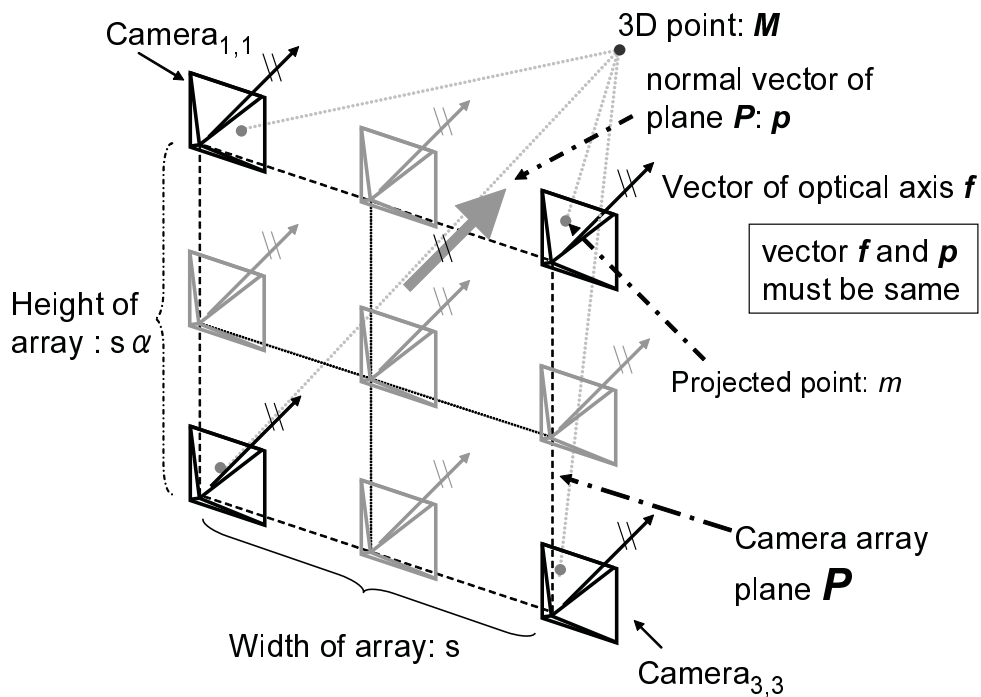


図 3.10 3次元空間中の3×3カメラアレイの理想状態

多視点画像の表記方法

本稿中では、2次元のアレイ状に配列されたカメラの2次元画像群である、4次元の多視点画像情報を紙面に記述するとき2つの方式で記述する。ひとつは、図3.11のように、各画像を各カメラの位置に合わせて、敷き詰めて並べる方式である。もう一方は、図3.12のように、各画像の画像座標が一致するように重ね合わせ、各画像の特徴点やその特徴点軌跡を表示する方式である。前者は、画像座標が実際とは異なる点や、スケールが異なる点により、細かな関係を把握しづらい一方、多視点画像全てを表示できる。後者は、どこかひとつの表示画像のみに限られるが、画像座標は正確と同じとなり、各画像の特徴点やその特徴点軌跡の対応関係を把握しやすい。そのため本稿では適時都合の良い形式で記述する。

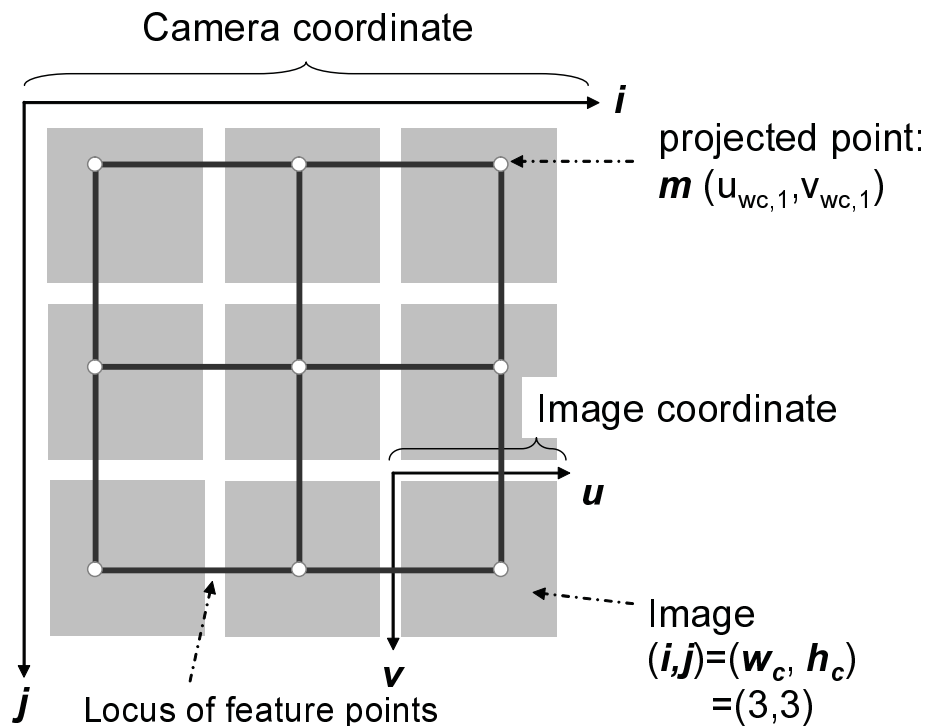


図 3.11 2次元画像中での 3×3 カメラアレイの理想状態 1

整列状態での二次元画像上での性質

三次元空間中での条件を満たしたカメラアレイで取得された画像群中の対応点は以下の性質を取る。ある三次元点 M が写る、 i 行 j 列目のカメラの画像座標 $(u_{i,j}, v_{i,j})$ において、同じカメラ行内では、対応点は全て同じ画像座標 v 行で等しくなり、また同じカメラ列内では、対応点は全て同じ画像座標 u 列に存在する (図 3.11)。

$$u_{i,1} = u_{i,2} = \dots = u_{i,h_c}$$

$$v_{1,j} = v_{2,j} = \dots = v_{w_c,j} \quad (3.19)$$

さらに、このカメラアレイが完全に格子状に並べられているとすれば、その格子の縦横比が $\alpha : 1$ (縦 : 横) としたとき、隣接する対応点同士の間隔

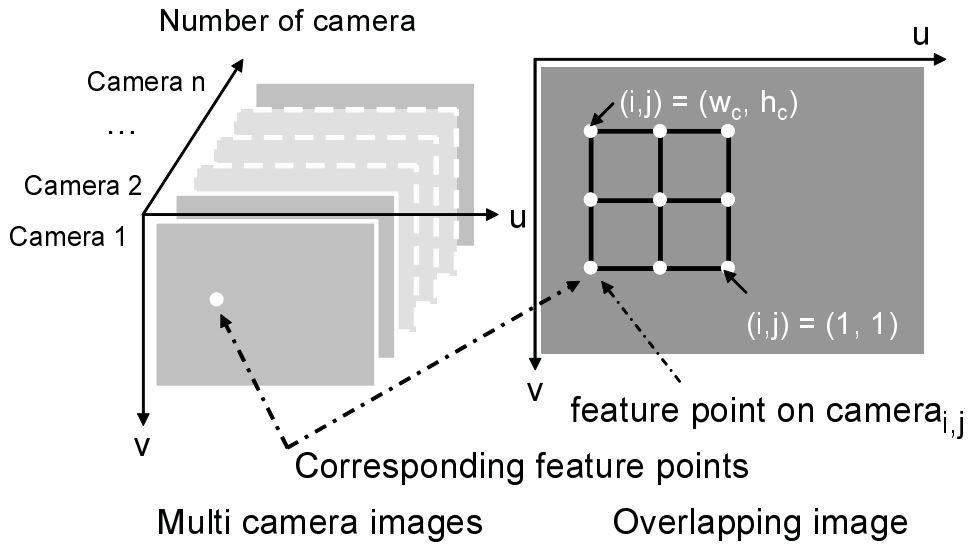


図 3.12 2次元画像中での3×3カメラアレイの理想状態2

d は等しくなり,

$$d = u_{1,j} - u_{2,j} = u_{2,j} - u_{3,j} = \dots = u_{w_c-1,j} - u_{w_c,j}$$

$$d\alpha = v_{i,1} - v_{i,2} = v_{i,2} - v_{i,3} = \dots = v_{i,h_c-1} - v_{i,h_c} \quad (3.20)$$

となり, 対応する点がカメラ番号とその間隔 d により順にアクセス可能な状態となる.

$$(u_{i,j}, v_{i,j}) = (u_{1,1} - id, v_{1,1} - jd\alpha) \quad (3.21)$$

本手法では, この状態に直接変換するようにレクティフィケーションを行う.

3.4.2 提案レクティフィケーション手法

本論文のレクティフィケーション手法は, 2段階のプロセスを経て達成される. まず, 全てのカメラの内部変数と, 姿勢を同一のものへ変換することで個体差をキャンセルする. 次に, 同一の方向を向く全てのカメ

ラの光軸をカメラの焦点が並ぶ平面 (図 3.10 中の平面 P) と直交する軸に変換する. これらの工程を 3次元空間上ではなく, 取得された特徴点の状態から直接演算する.

内部変数と姿勢の均一化

3.1 章, 3.2 章のプロセスは, カメラの配置位置には非依存な, 2台のカメラで成り立つ関係であるため, 添え字を置き換えることで表記を簡略化する. カメラ座標 (i_1, j_1) , カメラ座標 (i_2, j_2) の2台のカメラを指し示すために $(i_1, j_1) \rightarrow a, (i_2, j_2) \rightarrow b$ (ただし a, b は異なるカメラ), と表記する.

3次元空間中の平面 Π 上の点 M が2台のカメラ a, b に写る点を m_a, m_b とすると, その射影関係は式 (3.22) による 3×3 のホモグラフィ H_{ab} 行列で表される.

$$m_a = H_{ab} m_b \quad (3.22)$$

この行列 H_{ab} は平面 Π と各カメラの相対関係により式 (3.23) のように分解される [61].

$$H_{ab} = K_a \left(R_{ab} + \frac{tn^T}{l} \right) K_b^{-1} \quad (3.23)$$

ここで, R_{ab} は a, b のカメラの姿勢を R_a, R_b としたときのカメラ a に対する b の相対姿勢 $R_{ab} = R_a R_b^{-1}$, t は並進, K_a, K_b は2台のカメラの内部変数, l が平面までの距離, n が平面の法線ベクトルを表している (図 3.13). 式 (3.23) は, まずカメラ b に内部変数 K_b の逆行列を掛ける事で正規化し, 次に, 平面と各カメラの関係を用いて座標変換をし, 最後にカメラ a の内部変数を持つ画像面へ射影する事で, $a \rightarrow b$ への射影する事を意味する.

ここで, 平面 Π が無限遠にあると仮定したとき, 平面までの距離が $l \rightarrow \infty$ となるため, 式 (3.23) の並進の項が消え, カメラの内部変数と相対的な姿勢だけの関係のみを表すようになる. つまり, 無限遠平面のホ

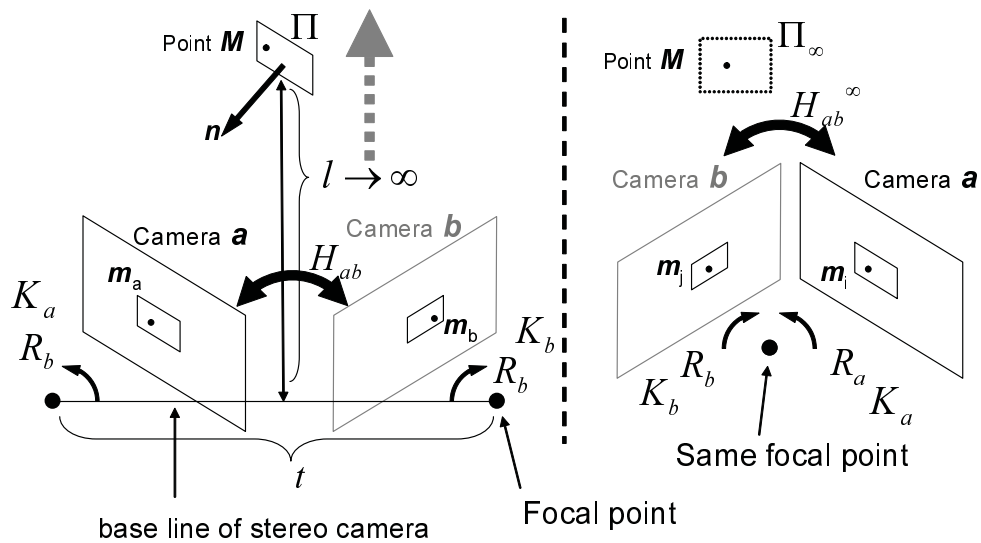


図 3.13 ホモグラフィ H_{ab} と各画像，平面 Π の関係

モグラフィ

$$H_{ab}^{\infty} = K_a R_a K_b^{-1} \quad (3.24)$$

となる．この時式 (3.23)，式 (3.24) より，無限遠平面に対し並進の項が無視できる点から，二つのカメラの焦点は一致している状態ともみなせる．

式 (3.24) に左から順に K_b, R_b を掛けて式を変形すると

$$K_a R_a = H_{ab}^{\infty} K_b R_b \quad (3.25)$$

となる．これは， H_{ab}^{∞} をカメラ b の姿勢，内部変数の行列積に掛けることで，カメラ a のものと同じとなることを示している．つまり，無限遠のホモグラフィさえ得ることが可能であれば，カメラの内部変数と姿勢の状態を暗に含んだ状態で，それらのパラメータを直接，均一にそろえることが可能となる．

消失点と無限遠点

3.1 章の無限遠平面のホモグラフィ H_{ab}^{∞} を求めるためには，無限遠に存在する物体の対応点をとる必要がある．この無限遠平面とは，運動視差

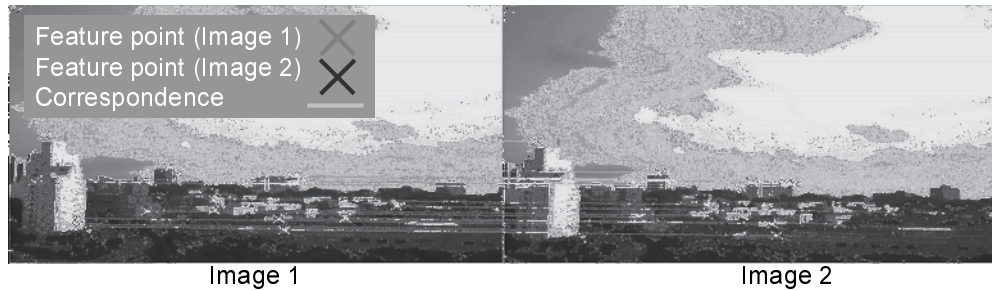


図 3.14 無限遠対応点

が無い，遠方にある平面であり，パノラマモザイク [62] に広く用いられている．図 3.14 中では遠くの風景が視差がほぼ零となり，無限遠と仮定することが可能である．この無限遠平面 Π^∞ 上の対応関係が求まれば，所望のホモグラフィを導くことが出来る．しかし，この遠方風景がレクティフィケーション時に常に存在するわけではなく，室内等では，無限遠点を直接得ることが比較的難しい場合もある．そこで本稿では，平面パターンによる消失点により代用可能な方法も示す．

消失点とは，3次元空間中で平行な二直線をピンホールカメラに射影したときの交点であり，その交点は無限遠にあるとみなせる．図 3.15 は格子パターンから消失点を得る例である．誤差のない環境下では，画像中の格子パターンを抽出できれば，直線群の交点として直ちに消失点は線形に演算可能である．しかしながら，実際はノイズ環境下での特徴点抽出となるため，実際には，格子点（全ての点間の距離が均一な点）にはなりえない．そこで，本稿では，抽出した特徴点列に最も近い格子を求める．この手順を図 3.16 に示す．実際に取得した特徴点を p_f ，理想的な格子（単純に全ての点間の距離が一定になるように与えた点列）を p_g とし， p_g が p_f に写るホモグラフィ H_g を誤差が最小となるように，非線形最適化手法である，Levenberg-Marquardt(LM) 法により最適化する [61]．

$$p_f = H_g p_g \quad (3.26)$$

上記で求めた格子点列 $H_g p_g$ からは単純な直線群の交点として消失点を得

られ、各カメラ i に写る消失点が無限遠点 p_i^∞ として利用可能となる。¹

次に、各カメラの方向と内部変数を均一化するため、これらの消失点ないし無限遠点を用いて、無限遠でのホモグラフィを計算する。あるカメラを基準カメラ a として²、カメラ b をそのカメラ a のパラメータにそろえる対象としたとき、無限遠平面で射影するホモグラフィ H_{ab}^∞ は、対応点が四点以上あれば、

$$p_a^\infty = H_{ab}^\infty p_b^\infty \quad (3.27)$$

となり、式 (3.27) を満たす H_{ab}^∞ を最小二乗的に求めることができる。この式 (3.27) の行列において、カメラ b を全てのカメラ i, j として、それぞれのカメラごとに H_{ab}^∞ を計算する ($a = \text{const. } b \in \text{camera}_{i,j}, 1 \leq i \leq w_c, 1 \leq j \leq h_c$)。ここで、基準カメラとなるカメラ a はどのカメラを選択してもよく、またカメラ b はカメラ座標を示せば、 (i, j) のカメラをさすため、

$$H_{ab}^\infty =: H_{i,j}^\infty \quad (3.28)$$

として記述する。 a の情報は不必要なため省略した。

このとき、風景などから直接無限遠点を自動的に抽出した場合には、全て正対応である対応点を抽出することは困難である。また、格子パターンから消失点を得る場合も間接的にしか得られないため、真値からの誤差が大きくなりアウトライアになる場合がある。そこで、ここでのホモグラフィ演算はアウトライア除去のため、投票による LMedS 法 (Least MEDian Square)[64] を用いてホモグラフィを求めた。

平面とカメラの直交化

無限遠平面での射影を各カメラに適用した段階で、カメラの内部変数、姿勢は同一の状態であり、このときの点の射影を、エピポーラ幾何では、

¹ p_g からなる直線の外積として導出する、もしくは格子の無限遠での交点 $(1, 0, 0)^T$ ないし $(0, 1, 0)^T$ に逆行列 H_g^{-1} を掛ければ導出可能。

²全てのカメラにおいて、姿勢、内部変数がそろっていることが目的であるから、 (i, j) のカメラのうちのどれかに合わせればよく、基準カメラとしてどのカメラを選んでもよい。

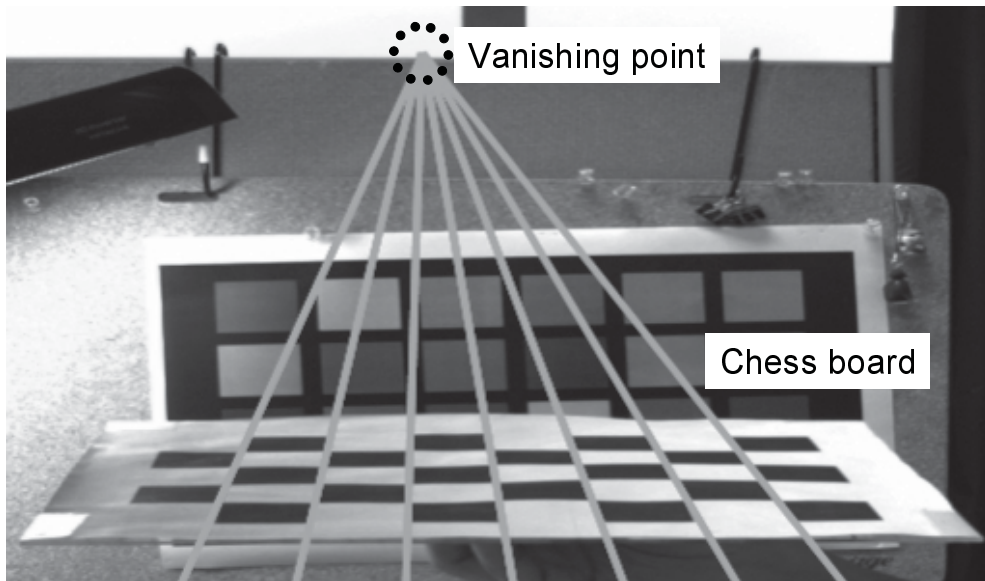


図 3.15 格子パターンからの消失点

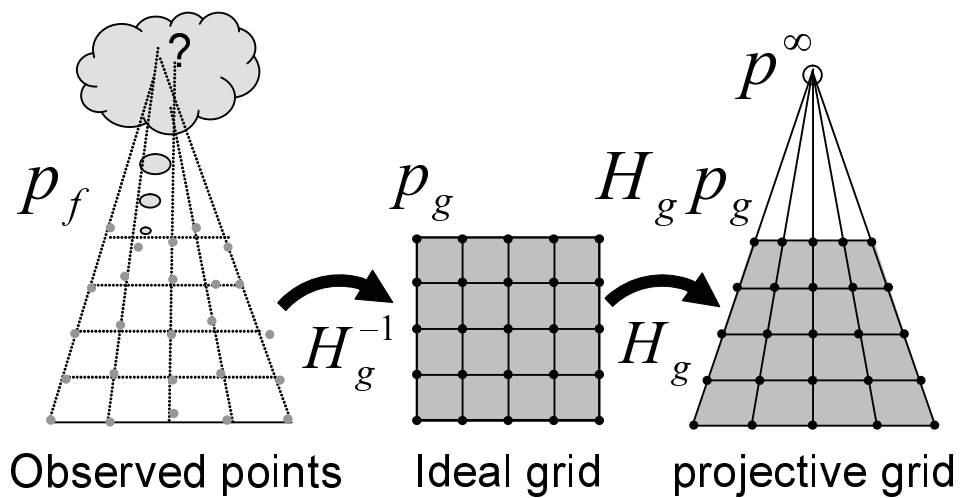


図 3.16 消失点の最適化

pure translation と呼ぶ [61]. pure translation 状態であつ、カメラの並進が平面で表されるとき（今回想定するカメラアレイは 2 次元平面），全てのカメラの全てのエピポーラ線が一致するという特性がある（図 3.17）.

具体例を示すため、無限遠平面上に乗らない特徴点、つまり、有限の距離にある点が n_{\max} 個あり、各カメラ (i,j) 上で対応する n 個目の特徴点を $p_{i,j}^n = (u_{i,j}^n, v_{i,j}^n)$ とする。そのとき、 n 番目の特徴点において、 i を固定して、全ての列 j を走査した特徴点列 $\{p_{i,1}^n, p_{i,2}^n, \dots, p_{i,h_c}^n\}$ 、また j を固定して、全ての行 i を走査特徴点列 $\{p_{1,j}^n, p_{2,j}^n, \dots, p_{w_c,j}^n\}$ は各自直線上に乗る。加えてこの縦・横の直線は、どの n 番目の特徴点においても一点に交わる。

この全ての縦・横のエピポーラ線の交わる点をエピポールと呼ぶ。このとき、図 3.17 のように、エピポールを作る特徴点の軌跡は格子を射影した形となり、その軌跡が交わる点がエピポールとなる。この点列から、エピポールを求める問題は、3.2 章の格子パターンを映したものから消失点を求める問題とまったく同じとなるため、同様の方法にてエピポールを計算可能である。³ただし、特徴点軌跡は取得した特徴点の数だけ得られ、ノイズ環境下では、各軌跡に対応するエピポールは一致しない。そのため、各水平、垂直方向のエピポールは各軌跡で得られたエピポールの重心を用いる。

この状態であるとき、本節では、全てのカメラに対して共通の射影行列をかけることでレクティフィケーションを行う。本節の処理の段階では、ただひとつの全てのカメラ間の対応点、つまり特徴点軌跡があればよい。

各エピポーラ線が平行となるとき、エピポールは無限遠に位置するようになる。水平方向のエピポールを $e_u = (e_{u0}, e_{u1}, 1)^T$ とし、無限遠に飛ばしたエピポールを $e'_u = (1, 0, 0)^T$ また、垂直方向のエピポールを $e_v = (e_{v0}, e_{v1}, 1)^T$ とし、無限遠に飛ばしたエピポールを $e'_v = (0, 1, 0)^T$ とすると、各エピポールを無限遠へと射影する行列 P は、

$$e'_w = Pe_w, w \in u, v \quad (3.29)$$

を満たす。式 (3.29) において、拡大縮小および、平行移動を許した、射

³格子パターン上の点から求める点と、内部・姿勢の変数がそろったカメラから得られたある 3 次元点の対応点列から求める点で異なるが幾何学的には同じものであり計算の過程は同じになる。

影成分のみを考慮した各エピポールを無限遠に射影する射影行列を P^* とすると,

$$[e'_u e'_v] = P^*[e_u e_v] \quad (3.30)$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & p_1 & 0 \\ p_2 & 1 & 0 \\ p_3 & p_4 & 1 \end{bmatrix} \begin{bmatrix} e_{u0} & e_{v0} \\ e_{u1} & e_{v1} \\ 1 & 1 \end{bmatrix} \quad (3.31)$$

となり, 未知数 p_1, p_2, p_3, p_4 は 2 つのエピポールの座標により線形に求まる.

この状態では, すでに全ての特徴点が水平・垂直に平行となる. しかし, 軌跡の縦横比は自由であり, またミラーリングを許している. まず, ミラーリングを修正する行列 A を定める.

$$A = \begin{bmatrix} m & 0 & 0 \\ 0 & n & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.32)$$

ただし, $m, n \in \{1, -1\}$. m, n はフリップがおきないように設定した.

最後に, 拡大縮小, 平行移動に関して自由度が残るが, 本論文では, 上記自由度を下記のように行列 S を定めることで制限した. 射影後の画像を囲む四角形のうち, 長い辺の長さが入力解像度と等しくなるように s 倍拡大・縮小し, 描画範囲内に収まるように平行移動 ($\text{shift}_u, \text{shift}_v$) させた. また, 所望の状態はカメラアレイの縦横比 α と軌跡の縦横比は等しく, 設置したときの理想的なカメラの比 α 値を代入することで拡大の縦横比を定めた,

$$S = \begin{bmatrix} s & 0 & \text{shift}_u \\ 0 & s\alpha & \text{shift}_v \\ 0 & 0 & 1 \end{bmatrix} \quad (3.33)$$

最終的に乗算する行列は

$$P = SAP^* \quad (3.34)$$

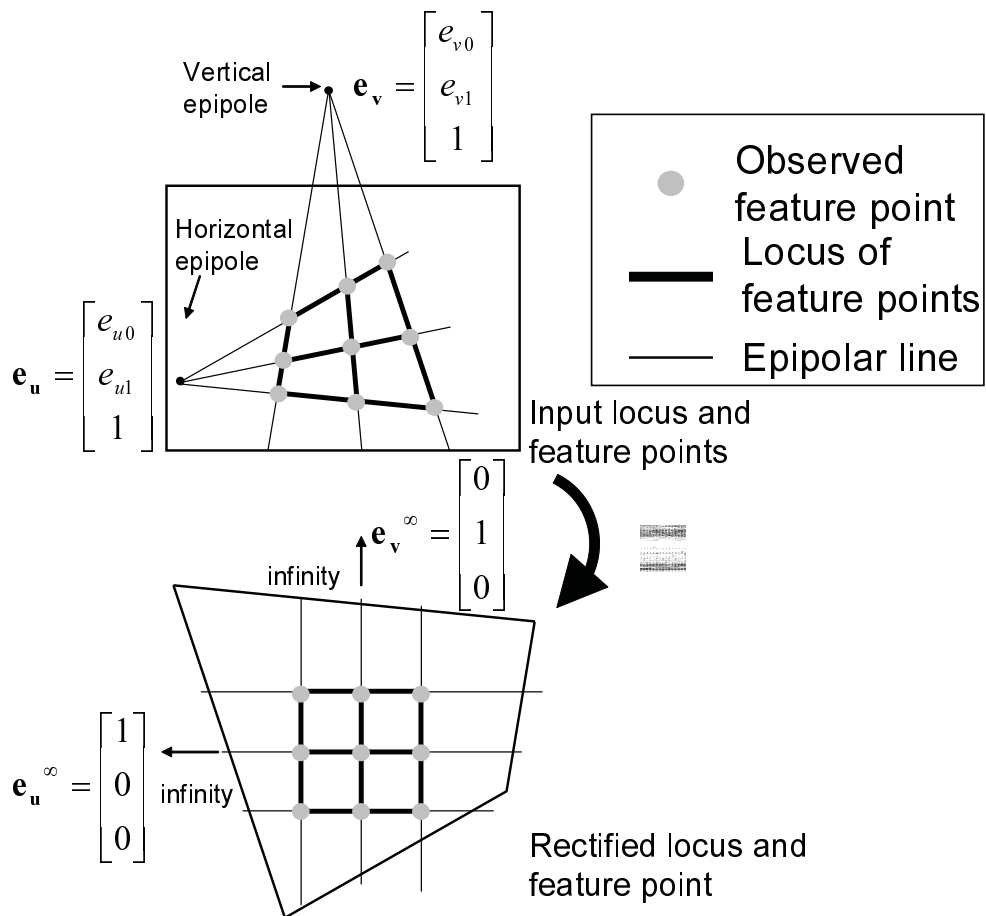


図 3.17 H^∞ により整列された特徴点軌跡

であり、この行列を共通に全てのカメラに適応する。

つまり、全てのカメラ (i,j) に対する固有のレクティフィケーション行列 $H_{i,j}$ は、

$$H_{i,j} = PH_{i,j}^\infty \quad (3.35)$$

となり、最終的に、式 (3.35) の行列で各画像を射影することでレクティフィケーションが完了する。

3.4.3 実験および結果

実験環境

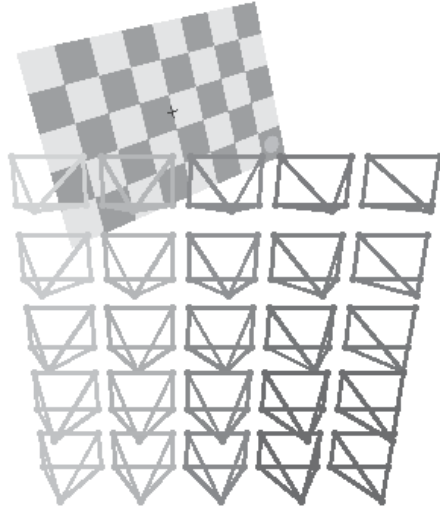
本手法を CG によるシミュレーション実験および実写実験により評価した。CG および実写両方とも、解像度 640×480 、画角 36.8 度を持つカメラで、 $5 \times 5 (i_{\max} \times j_{\max})$ の 2 次元カメラアレイを設置し、各カメラの間隔は縦横ともに 1.25cm である。図 3.18(a) は用いたシミュレーション環境であり、また、図 3.18(b) は実写実験で使用したカメラアレイである。各実験では、格子パターンから計算した消失点により無限遠平面の射影を求める方法と、直接風景などから取得可能な無限遠点を用いる二つの方式で実験を行った。

CG による実験では、 1m 離れた場所に $35\text{cm} \times 20\text{cm}$ の格子を写したシーンを想定して実験を行った。各カメラの姿勢は、 ± 2 度の範囲で、ピッチ・ヨー・ロールの各軸でランダムに設置し、また内部の焦点距離を ± 50 pixel の範囲でランダムに設置した。観測される特徴点の誤差はガウスノイズを仮定し、それぞれ u, v 軸方向へ標準偏差 σ_u, σ_v で付加した。また、無限遠点を得るためのパターンは 7×4 の格子を 20 枚撮影した ($7 \times 4 \times 20 = n_{\max}$ 点)。

実写実験でも、CG の場合と同じ、無限遠点を得るためのパターンは 7×4 の格子を 1m 先に設置し、角度を動かしながら 20 枚撮影した。ただし、実カメラの場合はレンズディストーションが生じるために、Zhang のキャリブレーション手法 [45] により歪み係数は既知として、たる型歪みの除去を行った。ただし、このときに同時に出てくるカメラの位置、内部変数の情報はまったく用いていない。

誤差関数

レクティフィケーション誤差を、実際の射影された各特徴点の u, v 軸に対する標準偏差で定義し評価を行う。図 3.19 中にある、ある特徴点軌跡において、 i カメラ行、 j カメラ列目に写る n 個目の特徴点の画像座標を



(a) CGシミュレーションで用いた5×5カメラアレイ



(b) 実カメラの5×5カメラアレイ

図 3.18 実験環境

$(u_{i,j}^n, v_{i,j}^n)$ として, j 行目の平均 u 座標を $\bar{u}_{ave,j}^n$, i 行目の平均 v 座標を $\bar{v}_{i,ave}^n$ とし, 各値の画像中 u 軸の誤差を E_u , v 軸の誤差を E_v として, 各々の平均的な格子からの距離を誤差とする. また E_u, E_v の二乗和の平方根を距離 E とする.

$$E_u = \sqrt{\frac{1}{\Omega_{\max} \times i_{\max} \times j_{\max}} \sum_n \sum_j \sum_i (\bar{u}_{ave,j}^n - u_{i,j}^n)^2} \quad (3.36)$$

$$E_v = \sqrt{\frac{1}{\Omega_{\max} \times i_{\max} \times j_{\max}} \sum_n \sum_j \sum_i (\bar{v}_{i,ave}^n - v_{i,j}^n)^2} \quad (3.37)$$

$$E = \sqrt{E_u^2 + E_v^2} \quad (3.38)$$

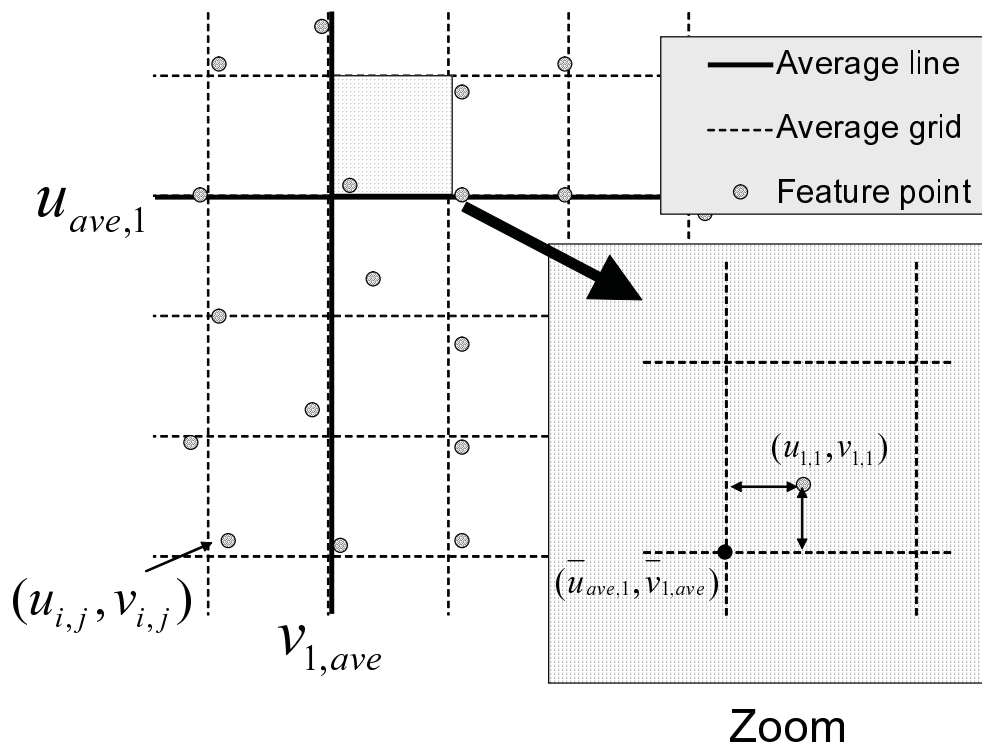


図 3.19 レクティフィケーションの誤差関数

結果

まず CG によるシミュレーション実験結果を示す。図 3.20 は観測される格子パターンの特徴点に標準偏差 $\sigma_u = \sigma_v = \sigma$ のノイズを加えたときの u, v 軸エラー距離 E を示しており、理論値、無限遠点を用いた本手法、格子パターンによる消失点を用いた本手法についてそれぞれプロットした結果である。理論値はシミュレーション時の理想的なパラメータを既知として入力した。また無限遠点による実験は、無限遠点に対して誤差を直接与えた。

Plenoptic Sampling[28] を考慮すると、誤差は u, v の各軸で、 ± 0.5 pixel の範囲内に収めたい。そのため、図 3.20 中では、 E は $\sqrt{2} \times 0.5$ pixel 以内の誤差に収めることとすると、既知パターンを使う場合、標準偏差が 0.3 pixel 以内に収めればよく、直接無限遠点を得る場合には、0.55 pixel で十分で

あることがわかる。これは、消失点演算などにより間接的に得る点に比べて、無限遠点を直接得たほうが無限遠点の真値からの誤差が小さかったためと考えられる。

次に実写による実験画像をレクティフィケーションした結果を示す。図 3.21, 3.22 は、その実写実験の結果画像である。図 3.21 は、2.3 節中の図 3.12 の方式で示したものであり、 $(i, j) = (1, 5)$ のカメラ画像を表示し、また、他のカメラの特徴点およびその軌跡をその上に描画したものである。描画されている特徴点は、格子パタンの左上内側のはじめのコーナーである。図中、左がレクティフィケーション前、右がレクティフィケーション後の画像であるが、平面パターン上の左上の乱雑であった特徴点軌跡が処理前後で、格子状へと変換されている事がわかる。また、図 3.22 は、2.3 節中の図 3.11 の方式でレクティフィケーション済みの全多視点画像とその軌跡を表示したものである。図 3.22 中では、描かれている軌跡以外の特徴点もおおむねレクティフィケーションされている事がわかる。

次にこの実写実験の数値的な誤差の値を示す。まず、入力画像そのままの誤差の値は 21.52 pixel であった。また、20 枚の格子パターンから消失点を用いてレクティフィケーションしたところ、誤差は、0.91 pixel であった。最後に実際に写る背景映像の対応を SIFT(Scale Invariant Feature Transform)[42] でとり⁴、無限遠での対応としたとき、0.25 pixel となり、十分な結果であった。

また、カメラキャリブレーションによる画像補正を試みたところ、カメラアレイ平面を推定することが出来ず適応不可能であった。実際のキャリブレーション結果では、カメラの順列が左右に入れ替わることや、前後に大きくふれてるなどしていた。これはカメラ間隔が 1.25cm と非常に近いため、カメラキャリブレーション精度限界に達しているからであると考えられる。カメラキャリブレーションの精度は、手法はもちろんのこ

⁴SIFT とは、特徴点とその点の特徴量の抽出方法である。その特徴量は画像の拡大縮小・アフィン回転にロバストであるため、画像間の自動対応付けが可能となる。ただし、アウトライアの混入は避けられないため、統計的な手法である LMedS 法 [64] 等を組み合わせて用いる。

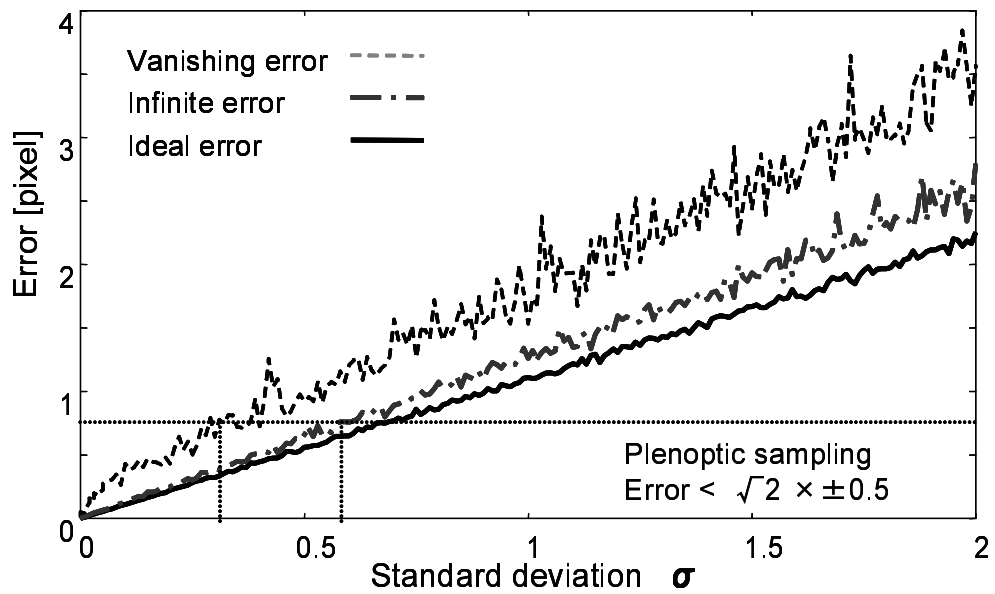


図 3.20 理想状態と提案手法の標準偏差と誤差の関係

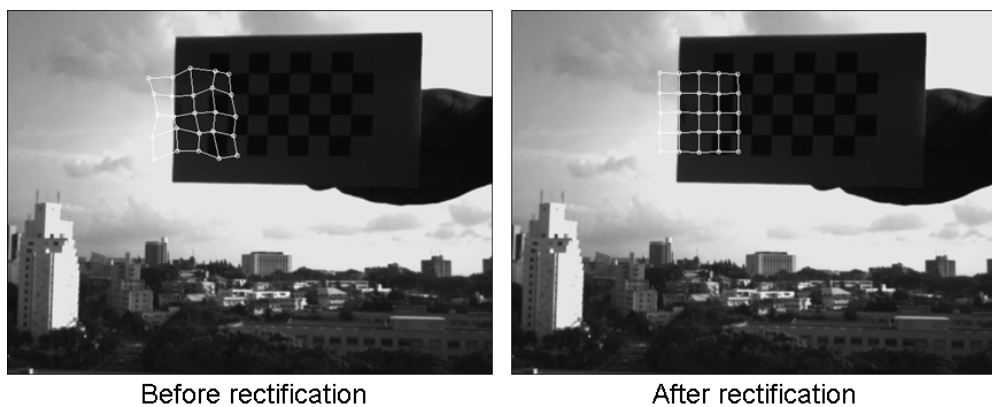


図 3.21 レクティフィケーション前・後の特徴点軌跡

と、特徴点の抽出精度，さらには，もともとのカメラの解像度や焦点距離にも依存する事が知られている [45]. これらの技術の向上とともに，カメラレイに特化したカメラキャリブレーションの検討も必要であろう.



図 3.22 レクティフィケーションされた全多視点画像とその対応点軌跡

3.4.4 自然特徴点による画像補正のまとめ

本節では、光線空間等で用いられる、マルチカメラを用いた二次元カメラアレイで取得される画像を全て平行化するレクティフィケーション方法を提案した。最低、四点の無限遠点と一点の有限距離の点の対応関係から、ノンメトリックな手法により、レクティフィケーションを可能とした。つまり、カメラの位置、方向等を測量するカメラキャリブレーションを必要としない手法である。実写実験により、実写画像のレクティフィケーション誤差が 0.25pixel となり、実用的な方法であることを示した。本手法は無限遠点が直接取得可能であれば、既知パターンなどを必要とせず、自然特徴点だけからも可能となる。

提案手法の問題点は、無限遠平面を求めるときの消失点がノイズに非常に影響されやすい点や、無限遠、有限距離と 2 段階に分離した結果、各

プロセス間で誤差が蓄積した点である。今後の課題として、上記の無限遠状態を経由しないような、有限距離のみで成立する関係を今後検討する予定である。

3.5 結言

自由視点テレビ実現のため、カメラの幾何補正について検討を行った。幾何補正は、後に述べる高速な自由視点画像生成には必須な技術であり、立体テレビ等に多視点画像を入力するためにも必須の技術である。

本章では、さまざまなカメラ配置に対して適応可能な、カメラパラメータを用いる手法と、二次元カメラアレイに特化した、特徴点を最適化する、ノンメトリックな手法を提案した。前者は、大規模なカメラアレイでは実用に耐える精度かつ、カメラパラメータも同時に算出することが可能であった。また、後者は、カメラパラメータは出ないものの、小規模なカメラアレイであっても十分適応可能であり、また高精度な補正手法であることが示された。具体的には、前者の手法を用いて80台のカメラで構成されるカメラアレイの映像を補正したところ、その補正誤差は0.42pixelとなった。自由視点画像合成に必要な精度である1pixel以内の精度を持ちこの補正により画像合成は十分に働く。後者の手法を用いれば、その誤差は0.35pixelとなり、本キャリブレーションを用いた手法よりも高精度なものとなっている。しかしながら、ノンメトリックな手法は補正に特化するため、カメラの焦点距離や光学中心などカメラパラメータは未知のままである。そのため、パラメータが必要な場合は前者を、必要ない場合は後者のような、より精度の高い物を使用するなど選択する必要があるだろう。

今後は、円形やドーム型のノンメトリックなレクティフィケーション手法や、カメラアレイに特化したカメラキャリブレーション手法を検討していく必要があるであろう。

第4章 多視点映像からの自由視点画像合成処理

4.1 緒言

Image-Based Rendering (IBR) とは、撮影画像を直接使い、様々な映像を描画する手法である。IBR は、ポリゴンを用いるコンピュータグラフィクスによる手法に比べ、容易に写実的な映像合成が可能である。

本節は、IBR である光線空間法 [25] を使い、自由視点画像合成を行うことを目的としている。光線空間法による自由視点画像合成を行うためには、Plenoptic Sampling[28] を満たす高密度な光線情報が必要となるが、この情報の取得は困難である。たとえば、大規模カメラアレイ (100 台程度、図 4.1 参照) を用いても十分な光線取得は不可能である。つまり、自由視点画像を出力するためには、不足光線を補間する事が必須となる。

この際、3次元シーンの完全復元を目指すとは負荷が高く、また、その精度には疑問が残る。本稿では、3次元モデルを介さず、直接的に仮想視点の合成を目指す。これは、視点依存視差推定を正確かつ高速に行うことが鍵となる [73]。本稿は、その視差推定を、被写体の奥行きが滑らかに変化するという拘束条件のもとで、高速に行う手法を提案する。

4.2 関連研究

自由視点画像合成手法にはさまざまなものが提案されている。モデリングに属する Model based rendering(MBR) と Image Based rendering(IBR) である。MBR では形状モデルとそのテクスチャが必要であり、そのモデ

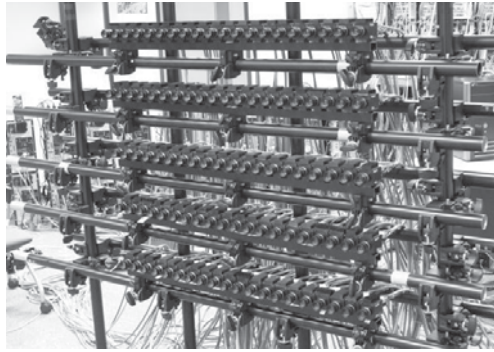


図 4.1 カメラアレイ

ル生成は実時間で行うのはコスト的に困難である。IBR による手法は、写実的な描画が可能である一方、サンプリング密度が十分でない場合、奥行き推定をともなう光線補間が必要となってくる。

MBR では、これを高速に行うために、被写体をひとつに限り、背景差分などで得たシルエットを投影することで物体形状を求める視体積交差法 [65] で求める手法がある。この手法は複雑な背景等を再現することは不可能であるため、背景をあらかじめ生成しておき、被写体をその上に合成する形式をとる。この実現例としては、PC クラスタによる実時間処理を実現した、九州大学 [66] のシステムや、京都大学のシステムがあげられる。

IBR による手法では、まず、東京大学による自由視点合成法が挙げられる。この手法は、未補正の画像をそのまま自由視点画像合成する枠組みとなっており、奥行き推定の際に全てのカメラを毎回射影変換する必要があるが、処理を GPU(Graphics Processing Unit) に任せることで並列化し、高速化することに成功している。現在では 8×8 の 64 台のネットワークカメラにて QVGA の解像度を 10fps 程度で描画することが可能となっている [71]。またこの手法は IP を入力としても適応されており、ほぼ同様な処理でリアルタイムな実行が可能となっている [70]。この手法では、二次元カメラアレイを入力として色補正や幾何補正を一切行わず、前者は一致度比較に色合い変化に強い、分散や正規化相関を使い後者の

対策には、毎奥行きごとに適切な射影変換を行う。一方、この手法は計算コストが高いが、代わりに、GPUにより計算を並列化、また、探索する奥行きを減らすことで高速化している。

他には、多視点画像を3次元フーリエ変換することで自由視点画像合成やフォーカス位置などを変更可能とした手法が東京工業大学や国立情報学研究所から提案されている [11]。この手法はいったん周波数領域に変換してしまえば、結果を利用して高速に描画可能だがその変換に計算時間がかかる。

IBR と MBR の中間的な手法が KDDI から提案されている [72]。この手法は、円周上に配置したカメラからの映像をいったんオフラインにて視体積交差法などであらいモデルを計算し、画像合成の時に、IBR 手法のように合成する視点にあわせて奥行き計算を行う手法である。この手法では、あらい奥行き情報等を利用し、被写体の隙間に入っていくなどウォークスルーが可能となっている。他には、サッカーに特化した自由視点画像合成法も慶応大学から提案されている [68]。フィールドや観客席などは事前のモデリングを用い、選手のみを IBR の手法で合成するものである。

以上を踏まえ、下記条件を全て満たす手法は提案されていないことがわかる。

1. IBR ベースの手法
2. 奥行き推定を含む、実時間での合成可能な手法
3. 高品質な自由視点画像合成

本章では、MBR などのモデリングではなく IBR に従う手法にて、高精度かつ高速な自由視点画像合成法を提案する。

4.3 視点依存自由視点画像合成

光線空間法により自由視点画像合成を行うことは、光線空間の切り出しだけでよく高速な自由視点画像合成が可能である。一方このためには、事前に多くの視点の映像を内挿しなければならず、この計算コストは非常に大きい。

そこで、直接的に自由視点画像を合成するには、生成視点を通る光線群とその光源奥行きに対応する画像中の点がわかれば出来ることを以下で示す。設置した2次元配置カメラアレイを真上から見たものである、図4.2中の Ray_A を補間する時、仮想視点 $V_{x,y,z}$ を通り、 (u,v) の方向を持つ補間対象の光線は、カメラアレイ上に張られる光線空間の参照平面座標 (X,Y) を以下の式で通過する。

$$X = x + uz, Y = y + vz$$

実カメラがその座標中整数値にある時、通過光線とその近傍カメラLU,LD,RU,RDの座標を、内分比 α, β , ($0 \leq \alpha, \beta < 1$)で表すと、

$$LU_{u,v}:([\![X]\!], [\![Y]\!]) \quad , \quad RU_{u,v}:([\![X+1]\!], [\![Y]\!]) \quad (4.1)$$

$$LD_{u,v}:([\![X]\!], [\![Y+1]\!]) \quad , \quad RD_{u,v}:([\![X+1]\!], [\![Y+1]\!]) \quad (4.2)$$

$$V_{x,y,z}(u,v) : ([\![X]\!] + \alpha, [\![Y]\!] + \beta) \quad (4.3)$$

となる(図4.3)。次に、光源奥行きと近傍カメラの画像対応をつけるために、図4.4のように奥行きを視差 d により関係付ける。視差は $d = \frac{Lf}{z}$ で与えられ(Lはカメラ間隔)、内分関係を用いれば、各画像対応は以下となる。(ただし、 $\bar{\alpha} = 1.0 - \alpha, \bar{\beta} = 1.0 - \beta$)

$$LU_{u,v}(d):(u+\alpha d, v+\beta d) \quad , \quad RU_{u,v}(d):(u-\bar{\alpha}d, v+\beta d) \quad (4.4)$$

$$LD_{u,v}(d):(u+\alpha d, v-\bar{\beta}d) \quad , \quad RD_{u,v}(d):(u-\bar{\alpha}d, v-\bar{\beta}d) \quad (4.5)$$

視差 d を与えた時、近傍カメラ中の対応光線群が、

$$LU_{u,v}(d) \simeq RU_{u,v}(d) \simeq LD_{u,v}(d) \simeq RD_{u,v}(d) \quad (4.6)$$

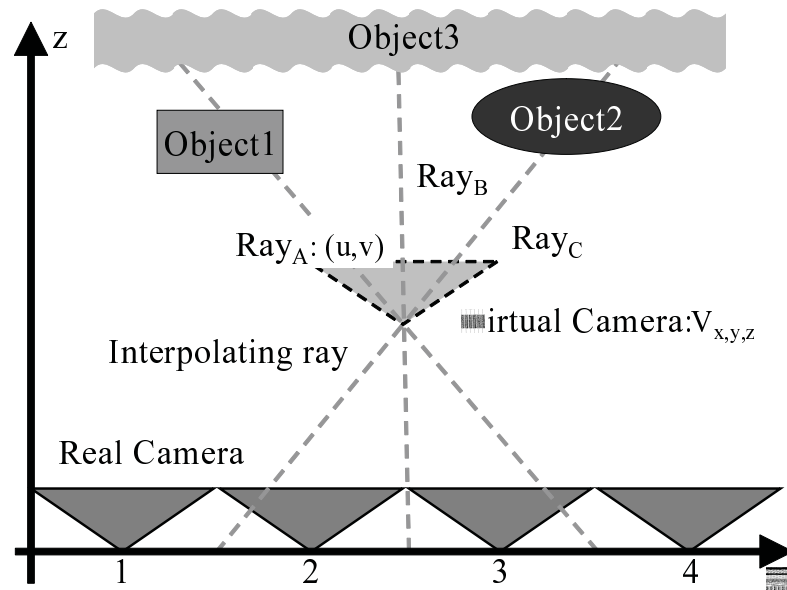


図 4.2 視点依存視差推定

となる Lambert 反射特性を満たせば，参照平面上での距離に応じて線形補間で光線補間が出来る．この視差決定の詳細は 2.2 節以降に説明する．

$$V_{x,y,z}(u, v) = \bar{\alpha}\bar{\beta}LU + \bar{\alpha}\bar{\beta}RU + \bar{\alpha}\beta LD + \alpha\beta RD \quad (4.7)$$

上記の，1) 生成視点決定，2) 必要近傍光線群の計算，3) 対応点からの生成視点の視差計算，という流れで推定される仮想視点中のシーン構造は，生成視点ごとに結果が異なるため，視点依存視差推定と呼ぶ．これらの処理を生成視点中の光線 $Ray_{B,C}$ など全光線で行うと，視点依存視差画像 $VDD(u, v)$ (View dependent disparity map) が生成される．このようにして，視点ごとに視差推定を伴い，直接的に画像合成を行う．

4.4 自由視点画像合成最適化

視点依存奥行き画像は，周囲の画像の一致度から推定することが出来る．本論文では，推定方法を二つの方法に大別する．一つ目の手法はマッ

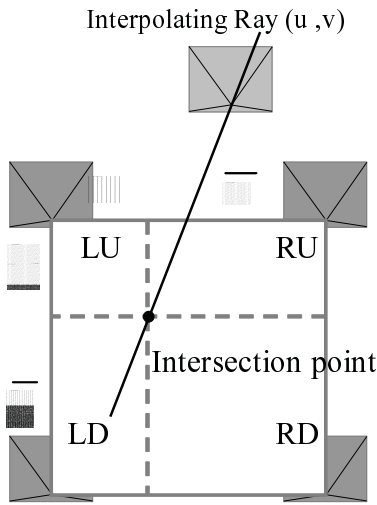


図 4.3 内分

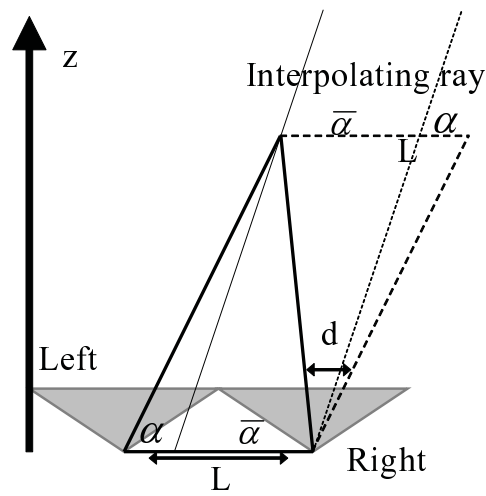


図 4.4 視差

チに用いる領域を広げ，ブロックなどの広い範囲でマッチングする方法である．二つ目の方法は，奥行きが滑らかに変化するとし，周囲の pixel の状況に応じて奥行きを決定する手法である．前者をローカル最小化，後者をグローバル最小化とよび，それぞれ 4.4.1 節，4.4.2 節で述べる．

4.4.1 ローカル最小化

式 4.6 は対応する各画像の画素値の比較から求められる．関数 Φ には SAD(Sum of Absolute Difference)，SSD(Sum of Square Difference)，正規化相関値，分散値などがあるが，本論文では SAD を例にして論ずる¹．まず，合成画像上 (u, v) の位置の一致度 Φ を測る．補間光線の近傍カメラを通る光線を $Ray_{p,q} \in Ray = \{LU, LD, RU, RD\} (p \neq q)$ とし全組み合わせの和を取れば．

$$\Phi_{pixel}^{u,v}(d) = \sum_{Ray_{p,q} \in Ray} |Ray_p^{u,v}(d) - Ray_q^{u,v}(d)|$$

¹SSD,SAD, 正規化相関の順に入力映像の色補正や幾何補正の結果に敏感となり，SSD は CG では最も結果が良いが，実写では，ノイズの影響で SAD に比べて結果が悪いため，今回は SAD を用いた．また，分散は比較映像が 3 枚以上の時に用いる場合がある．

となる。この関数 Φ_{pixel} の最小値を与える奥行き d が求める視点依存奥行き画像の奥行き値となる。

$$VDD(u, v) = \operatorname{argmin}_d \Phi_{pixel}^{u,v}(d)$$

しかし、この条件だけでは多くの誤対応がおきる。これは、画像上のノイズの影響や、テクスチャが一様な領域でどの画素を選んでも同様な一致度になるという対応の曖昧さによる。そのため、ひとつの画素だけでなくその周囲の画素を用いることで安定化させる。

$$\Phi^{u,v}(d) = \sum_{i=-bx}^{bx} \sum_{j=-by}^{by} \Phi_{pixel}^{u+i,v+j}$$

ここで、 bx, by はブロックのサイズである。このように pixel ではなくブロックでマッチングを取ることでノイズなどの影響から逃れることが可能となる。一方でこのブロック化にはデメリットがあり、それは、物体の境界がブロックの大きさに応じて鈍り、ミスマッチ領域が広がることである。このミスマッチがおきたで光線を補間すると物体境界は背景物体と被写体の色が不自然に混ざったものとなり視覚的な劣化が激しい。加えて、テクスチャが一様な領域での不安定さを解決するためには、比較的大きなブロックを用いざるを得ないため状況は悪化する。そこで、本論文では、グローバル最小化という概念を用いてこの問題を解決する。

4.4.2 グローバル最小化

グローバル最小化とは、滑らかな奥行き画像が合成されるように、周囲の奥行き状態との相違を拘束条件として奥行きを導出する手法であり、テクスチャが一様な領域での曖昧さの問題を解決する。この手法は、画像の一致度に加え、その合成される奥行き画像の微分値も同時に加えエネルギー E_I として最小化する。

$$E_I = \sum_u \sum_v (\Phi^{u,v} + \lambda \|\nabla VDD(u, v)\|) \quad (4.8)$$

ここで、 λ は両コストのバランス定数である。この式では、奥行きが不連続であれば、 λ 以降がペナルティとして働くため、極力滑らかな奥行きを導出する。

このエネルギー E_I のモデルは Morkov Random Field[75] と呼ばれる、その最小化にかかる計算オーダーは $O(d^{wh})$ 、つまり、奥行き候補値の解像度乗となり、NP 問題であることが知られている。

この最小化問題を近似解をもとめる手法は多数提案されている。グラフカット、ビリーフプロパゲーション [29]、動的計画法 (Dynamic programming) などである。グラフカットは、奥行き候補が2つの場合の最適解を解く手法であり、それ以上の候補値がある場合はイタレーション処理により近似的な解法となる。ビリーフプロパゲーションは、1度の処理につき周囲4連結に自身の一致度を伝播させていく手法であり、イタレーションすることにより解く。これらアルゴリズムは精度よく問題を解くことが可能だが、計算時間が多いため実時間での自由視点画像合成には不向きである。

よりリアルタイム性の強いとして、動的計画法があげられる。この手法は周囲の影響を微分を1方向に限定することで効率的に解く手法であるが上記二つのアルゴリズムに比べると精度は落ちる。

本稿では、動的計画法をベースとして、より精度の高い最適化手法を提案する。

4.5 提案手法

本節では、Dynamic Programming(DP)について説明し、それを多パス化した Multi Pass Dynamic Programming(MPDP)について述べる。またその MPDP の高速化手法である、距離変換について述べる。

4.5.1 Dynamic Programming による視差推定

式 4.8 の右辺左項の滑らかさの成分を 1 方向に限定すると二次元画像中の最適化となり，Dynamic Programming (DP) により効率的に解ける [74]。本章中の DP も [74] と同様に，左画素の視差と自分との差分 E_S と，自身の画素の一致度 E_D の和をエネルギー関数 E として総和を最小化する。しかし文献 [74] はステレオ画像上の視差推定であり，本手法は，視点依存視差画像の視差推定である。そこで，画像ライン y ごとのエネルギー関数 E_{line}^y を最小化する視差 $d(x)$ の組み合わせを視点依存視差画像とする。 E_S 中の γ は大きな差がある際の打ち切り値である。

$$E^{x,y}(d(x)) = E_D^{x,y}(d(x)) + \lambda E_S^{x,y}(d(x))$$

$$E_S = \min(|d(x-1) - d(x)|, \gamma)$$

$$E_{line}^y = \min \sum_x (E^{x,y}(d(x)))$$

$$VDD^y(x) = \operatorname{argmin}_{d(x)} E_{line}^y$$

E_{line} の最小値は，通常は視差の組み合わせを総当りで求めるしかないが，DP を用いれば，効率的に解ける。まず，画像中 x 列目までのエネルギー関数の総和を各視差 d ごとに表すと $F_x(d) = \min \sum_0^x (E^{x,y})$ であり， x が終端なら最小化対象の E_{line} と同じとなる。最適性の原理から，全体が最適であるとき，その部分も最適でなければならぬため， F_x は直前の最適解の F_{x-1} とその解を選択する直前の奥行き d_{pre} との漸化式で表される。 x 列目の視差を $d=d(x)$ とすると，

$$F_x(d) = \min(F_{x-1}(d_{pre}) + E^{x,y}(d)) \quad (4.9)$$

求めるものは，この F を最小とする視差の組み合わせであるので，左画素の奥行き d_{pre} を全ての奥行き状態において $x = 1, 2, \dots$ と順次計算し $pd_x(d)$ へ格納する。そして，終端まで演算し，終端で最小値を持つ視差から， $pd(x)$ をトラックバックすることで視差 VDD の組み合わせを選択

表 4.1 F_x とその直前の最小視差

	pd: $F_1(d)$	pd: $F_2(d)$	pd: $F_3(d)$	pd: $F_4(d)$
d=0	N/A : 6	1 : 13	1 : 6	1 : 9
d=1	N/A : 2	1 : 4	1 : 5	1 : 7
d=2	N/A : 2	0 : 2	0 : 7	1 : 6

する.

$$pd_x(d) = \underset{d_{pre}}{\operatorname{argmin}}(F_{x-1}(d_{pre}) + E^{x,y}(d))$$

図 4.5 は, この漸化式を模式化したもので各 Node は E_D のコストを持ち, 異なる視差状態へ遷移をする edge には相応のコスト E_S を与える. また表 4.1 は各々の視差状態における, 累積最小コスト F と直前の最適視差を記憶したものである. 上記の問題はこのネットワークの最小経路を導出する問題と等価である. まず, 初期値 F_1 に一致度 E_D^1 を各奥行きごとに代入する. 次に F_2 の全ての奥行き状態に対し, F_1 , と edge の和が最小となる直前の pass を決定・記憶し, 自身の一致度 E_D^2 をたし合わせ F_2 とする. この処理を終端まで行い, 最小値を持つノードを見つけ, それらの pass がつながる経路を, 最小コストを与える視差の組とする. しかしながら, DP は 1次元画像の最適化に留まるため, 行間が飛び飛びになる, 次節はこれを解決する手法を述べる.

4.5.2 Multi-Pass Dynamic Programming

行間の不連続性の問題を解決するためには, エネルギー関数 E_I において上下左右の微分値を考慮する必要があるが, DP では 1次元画像の最適化が限界である. そのため, 本稿では, 他方向からの視差の滑らかさを伝播させた一致度 E_D 新たに合成し DP を適応する. ここでは DP を順方向, 逆方向, 縦方向の 3 方向から行い, エネルギー最小化する.

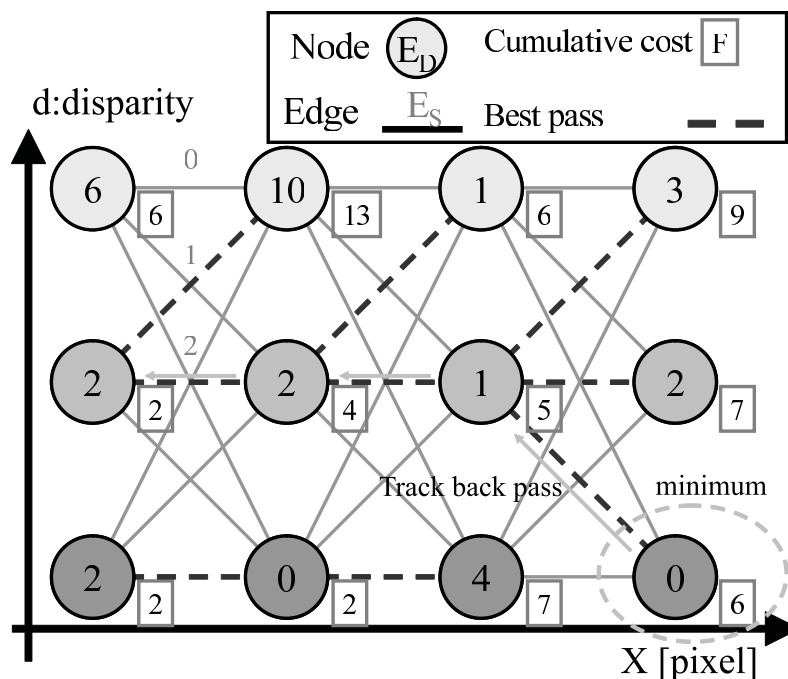


図 4.5 Dynamic Programming のネットワーク

さて、先の DP で求めた各々の行 y の部分最小コスト $F_x^y(d)$ を一致度 $E_D^{x,y}(d)$ とし、各々独立に最小値を選択すると、たとえトラックバック経路を記憶しなくとも、ほぼ DP と同様な準最適解の視差が得られる。

$$VDD(x, y) = \operatorname{argmin}_d F_x^y(d(x))$$

これは、DP のコスト算出時にイレギュラーな経路に大きなコストが計上されるためである。図 4.6 は準最適解、最適解の視差を示したものである。つまり、一度目の DP の結果である F に対し、別方向からの 2 度目の DP を適応すると、行間の不連続性を解決することが可能である。しかし F は、 x が進行すると累積的に増加する (Fig. 4.7)。対して、SAD の一致度はほぼ一定値である。そのため、 F をコストとして用いるには正規化の必要がある。ここでは、単純に各 x ごとに正規化するのではなく、DP を順方向に対し、逆向きの進行方向 (つまり $x, x+1$ の漸化式を逆方向から解く) から適応したものを足し合わせることで正規化する。Fig. 4.7 は

y 行目の順方向のコスト F_{x+}^y , 逆方向のコスト F_{x-}^y , その和 $F_{x_{total}}^y$ を示したものである.

$$E_D^{x,y}(d) = F_{x_{total}}^y(d) = F_{x+}^y(d) + F_{x-}^y(d)$$

このように, 順方向と逆方向からの DP 結果の和を取ると, 正規化すると同時に左右の影響を考慮した一致度が合成可能となる. この正規化したコスト $F_{x_{total}}^y$ に対して, 最終的に縦方向から DP を適応する.

縦方向の漸化式を, x 行目の y 列目までの縦の部分エネルギー $G_x^y(d(y))$, 直前の視差を $d_{pre} = d_x(y-1)$, 現在の視差を $d = d_x(y)$ を用いると

$$E^{x,y}(d_x(y)) = F_{x_{total}}^y + \lambda E_S^{x,y}(d_x(y))$$

$$E_S = \min(|d_x(y-1) - d_x(y)|, \gamma)$$

$$E_{line}^x = \min \sum_y (E^{x,y}(d_x(y)))$$

$$VDD^x(y) = \operatorname{argmin}_{d_x(y)} E_{line}^y$$

$$G_y^x(d) = \min(G_{y-1}^x(d_{pre}) + E^{x,y}(d))$$

$$pd_y(d) = \operatorname{argmin}_{d_{pre}} (G_{y-1}^x(d_{pre}) + E^{x,y}(d))$$

となり, 縦方向にたどることで最小のコストを与える奥行きが求まる. このように多方向の DP を統合する, Multi-Pass DP により, 行間の相違が少ない推定が可能となる. この3つの多方向への DP を本稿では Multi-Pass Dynamic Programming(MPDP) と呼ぶ.

4.5.3 距離変換による高速化

距離変換

Dynamic Programming は最適経路を求める手法であり, その導出には $O(d^2)$ の計算コストがかかるが, 準最適コストのみを求めることに限れば $O(d)$ で解くことが可能である. この高速化は距離変換 [76] を用いること

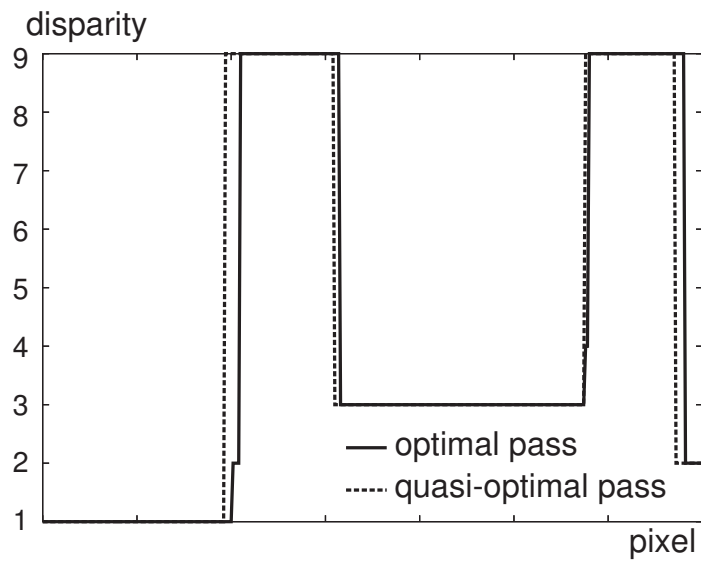


図 4.6 最適解と準最適解の視差

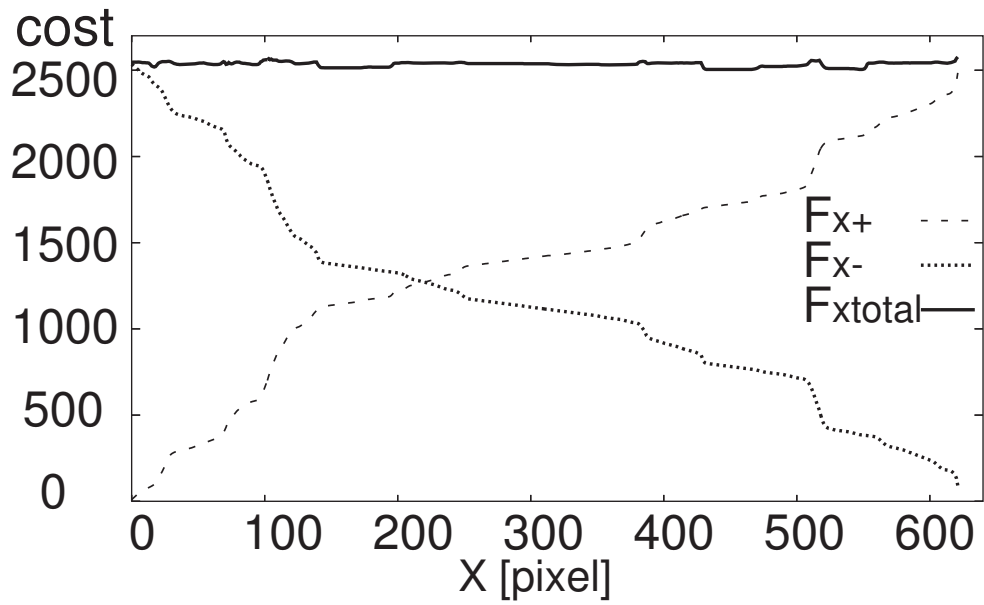


図 4.7 順方向, 逆方向の和をとり DP コストを正規化

で行う。図 4.8 は距離変換による準最適コストを求める図であり、点が一つ前の状態の F をあらわしている。また、その点から伸びる直線は式 4.9 右辺の \min 内部を表しており、前の F と今の F との奥行きが離れれば離れるほどエラーが大きくなる。この図中で、すべての奥行き状態においてそれぞれ最小のエラー値を求めることが課題であり、つまり、図中の赤線を求めることである。

まず、各状態において、奥行きの状態がひとつ手前のものに λ を加えたものと自身とを \min をとり、それを現在の準最適コスト F_x^{cur} として更新する。

$$F_x^{cur}(d) = \min(F_{x-1}(d), F_{x-1} + \lambda) \quad (4.10)$$

図 4.9 (左) がこの処理を行った後の状態である。次は反対に、各状態において、奥行きの状態がひとつ奥のものに λ を加えたものと更新したコスト F^{cur} とを \min をとると、それが最小と値となる。つまり、状態数 d 個の \min 演算を 2 度するだけで、最小値が計算できオーダが $O(d)$ となる。

この計算は、 F を近似することなく効率化することが可能となる。

より高速な奥行き推定

4.5.2 節の手法は、2 度の準最適解演算と 1 度の最適解演算を行うため、結局オーダは $O(d^2)$ となる。そこですべてが準最適化演算で収まるように視点依存奥行き画像を以下の式で導出する。

$$d = \operatorname{argmin}_d (F_f + F_b + F_p + F_n) \quad (4.11)$$

ここで F_f, F_b, F_p, F_n はそれぞれ、順方向、逆方向、上方向、下方向へ計算したものである。これらのコストの和の \min をとることで奥行き画像とした。

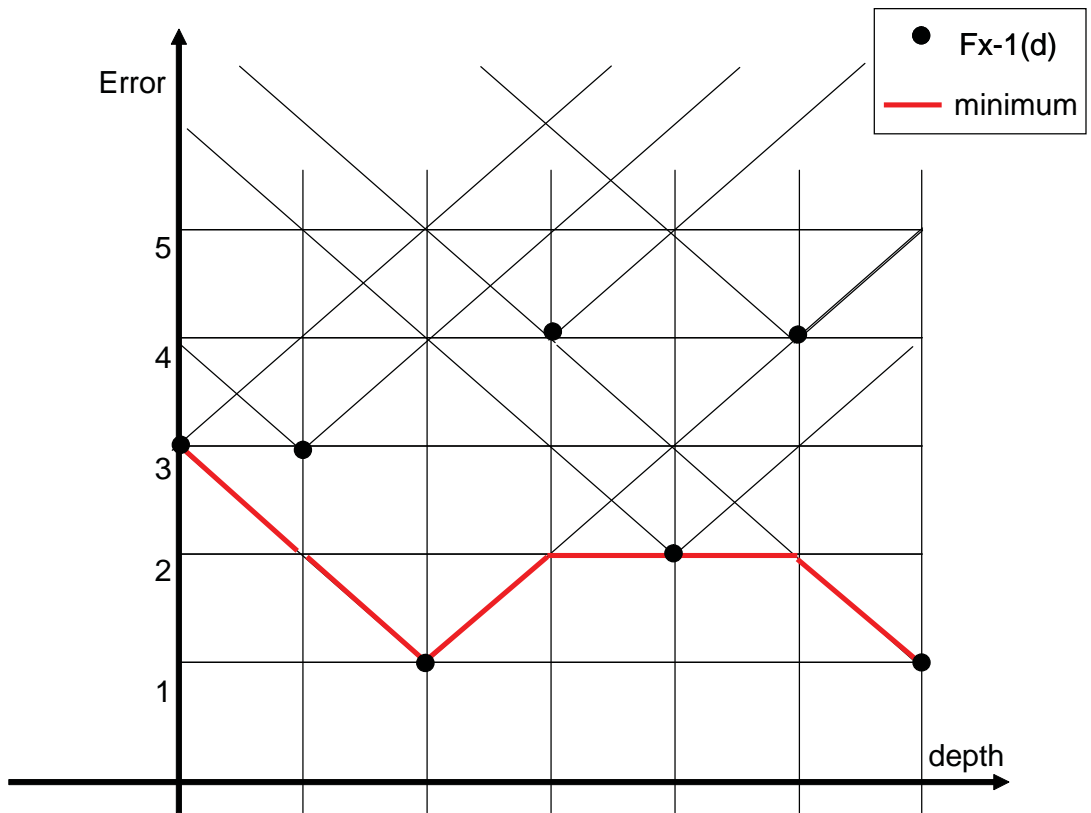


図 4.8 距離変換による準最適コストの導出

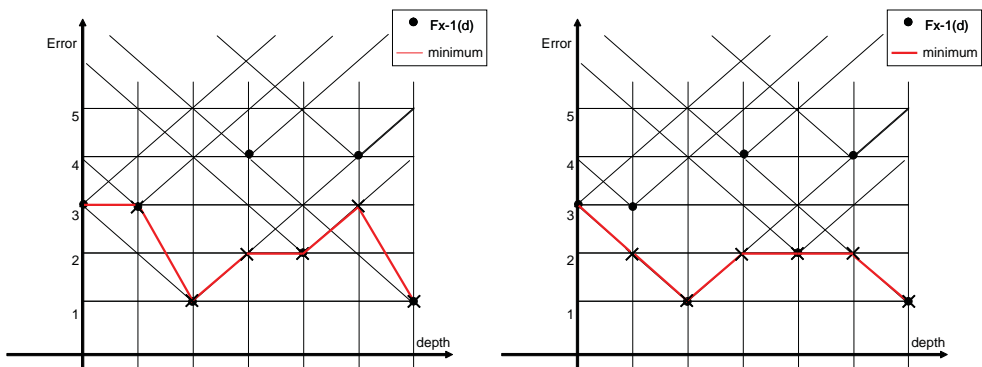


図 4.9 距離変換による準最適コストの導出の流れ

4.6 実験

ピクセルマッチング (Pixel Matching:PM), ブロックマッチング (Block Matching:BM), ビリーフプロパゲーション (Belief Propagation:BP), グラフカット (Graph Cut:GC), 動的計画法 (Dynamic Programming:DP), Multi Pass Dynamic Programming(MPDP) の各手法において精度, 計算時間の比較実験を行った. 精度測定に用いる参照画像は, 密に撮影した多視点データを間引き, 補間した画像 $int(u, v)$ と実撮影画像 $ref(u, v)$ とで測定した. また画質は PSNR(Peak Signal-to-Noise Ratio) で測定を行った. PSNR は式 4.12, 式 4.13 で定義され, 入力画像解像度 $w \times h$ 中, 探索範囲から外れるバウンダリーボックスを最大視差 md とブロックサイズの半径 br で定義し計算から除外した. 図 4.10 は PSNR を計算する領域を可視化したものである. また, 色空間を RGB 空間ではなく YUV の輝度信号 Y を用いた. RGB から YUV への変換は式 4.14 による.

$$PSNR = 20 \log\left(\frac{255}{\sqrt{MSE}}\right) \quad (4.12)$$

$$MSE = \frac{\sum_{i=md+br}^{w-1-md-br} \sum_{j=br}^{h-1-br} (ref(i, j) - inter(i, j))^2}{(w - 2br - 2md)(h - 2br)} \quad (4.13)$$

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (4.14)$$

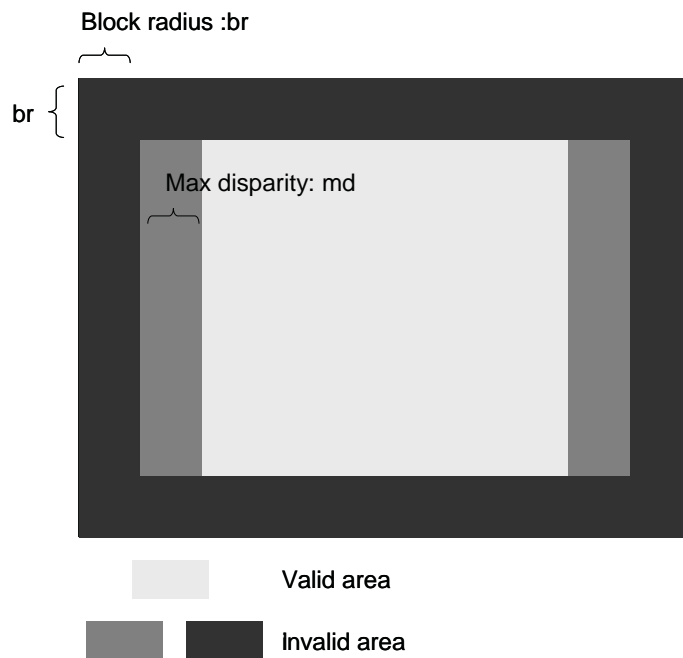


図 4.10 有効計算領域

実験は 4 種類行った。一つ目は、ローカル最小化においてブロックサイズが PSNR にどのように影響するかを測定し、二つ目は、最大視差を変化させたときの各手法の PSNR を測定した。三つ目は、最大視差を変化させたときの各手法の計算時間を、最後に計算時間と画質のコストパフォーマンスを計測した。

4.6.1 ローカル最小化

まず、ローカル最小化において、ブロックサイズを広げることで画質がどう変化するかを測定した。図 4.12 は x 軸にブロックサイズ、y 軸に PSNR をとったグラフである。このとき、画像のペアは最大視差は 22,44,66,88 の 4 種類を用い、また PSNR の測定関数は最小の有効領域にあわせ、 $br = 30, md = 88$ と設定した。またこのときの補間画像と奥行き画像はそれぞれ図 4.13, 図 4.14, 図 4.15, 図 4.16 となっている。図 4.12 より、ブロッ

クサイズが20付近までPSNRが増加し飽和した後、減少している。これは、図4.13、図4.14、図4.15の奥行き画像を比較すると、テクスチャの薄い領域において誤対応が減少していることがわかる。特に、帽子や背景などの領域で穴が開いているような状態だったものが正しく対応付けられ、その効果が目立つ。これが、PSNRを増加させた主要因である。一方で、一定の大きさ以上では画質は向上するどころか劣化する。これは、物体境界の奥行きが鈍化しているためである。比較のため、手前の人形と奥の人形の境界を縁取った画像を奥行き画像に重ね合わせたものを図4.17に示す。ピクセルマッチングでは、誤対応は多いものの、境界際の奥行きはおおむね正しく推定されている一方で、大きなブロックマッチングでは輪郭に収まることはない。これは合成画像の品質に大きく影響し、ゴースト像として現れる(図4.18)。

このようにローカル最小化による視点依存奥行き推定は、テクスチャ多きな領域では有効に働く大きなブロックマッチングも境界際では悪く働くという背反した特性を持つ。次の実験のグローバル最小化は上記の欠点を解消する。



図 4.11 合成対象の源画像

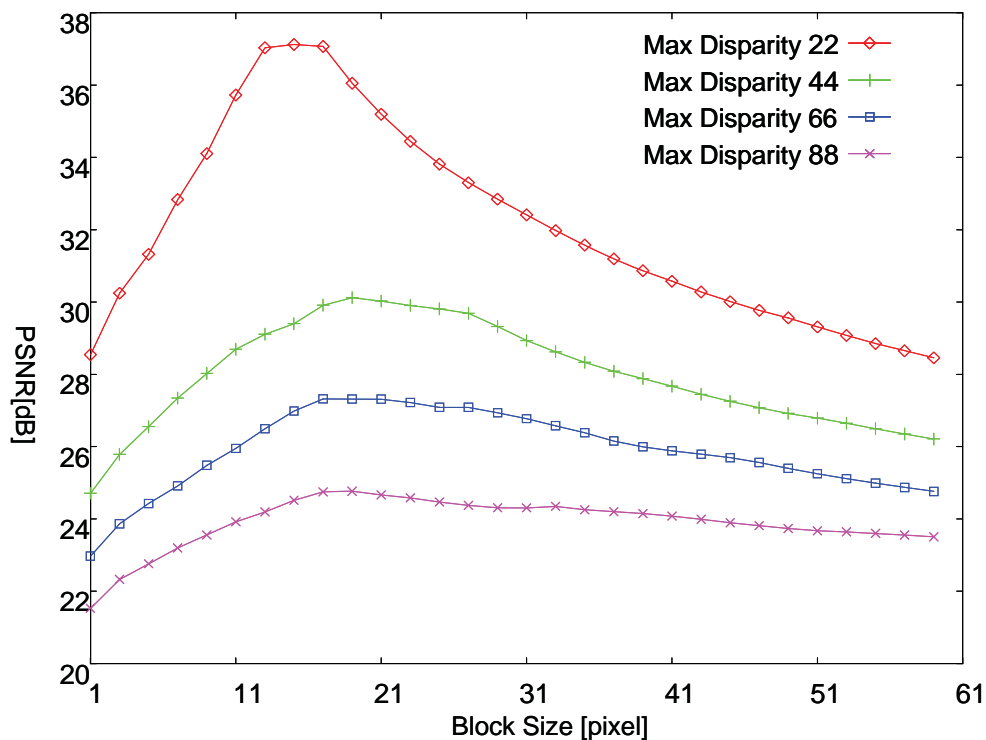


図 4.12 ブロックサイズと PSNR

4.6.2 最大視差と PSNR

本実験では、PSNR を、Multi Pass Dynamic Programming(MPDP) とその高速化(MPDP Fast), Belief Propagation(BP), Graph Cut(GC), Dynamic Programming(DP) と Pixel Matching, Block Matching(BM) の各手法を比較したその際、最大視差を変化させて測定した。

結果は、図 4.19 である。また最大視差 22pixel のときの奥行き画像、合成画像を図 4.20 から図 4.24 まで列挙する。

すべての最大視差の範囲で、精度は、GC, BP, MPDP, MPDP Fast という状態となっている。それ以降は、もっとも近い領域において、順序が前後するが、グローバル最小化の DP, ブロックサイズの大きい BM と



図 4.13 ピクセルマッチングの結果 右) 補間画像, 左) 奥行き画像



図 4.14 ブロックマッチング (5×5) の結果 右) 補間画像, 左) 奥行き画像



図 4.15 ブロックマッチング (9×9) の結果 右) 補間画像, 左) 奥行き画像



図 4.16 ブロックマッチング (19×19) の結果 右) 補間画像, 左) 奥行き画像

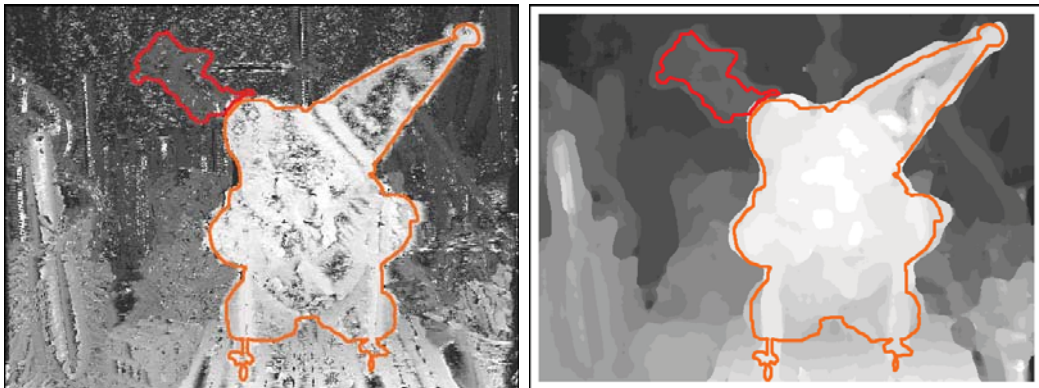


図 4.17 左) ピクセルマッチング 右) ブロックマッチング (19 × 19) の奥行き画像. 赤, 橙はオブジェクトの輪郭

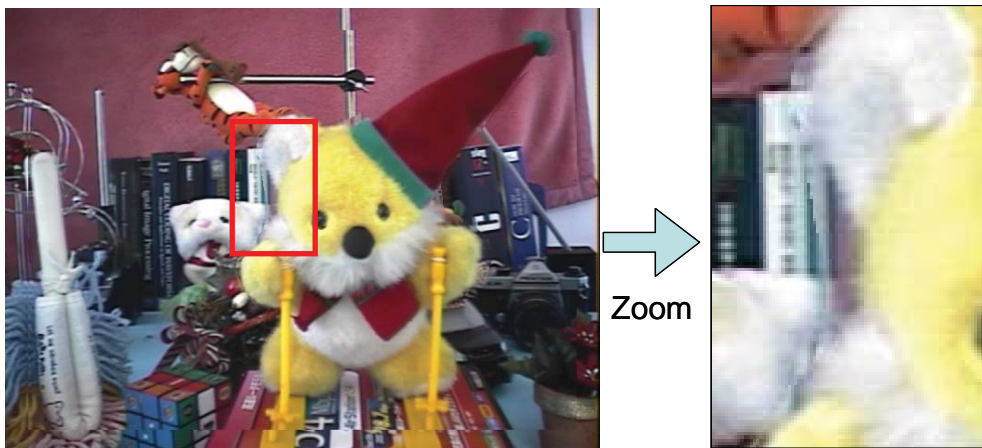


図 4.18 境界の鈍化によるゴースト像

続く. ここからわかる事は, グローバル最小化はローカル最小化に比べ高い画質を得やすいことがわかる. BP, GC はほぼ同レベルであり, 最高品質の画像を得ることが可能であり, また, MPDP もそれにほぼ順じた結果となった. また高速化した MPDP は若干精度が劣るものの他の手法よりもよい結果となっている.

MPDP とその高速化したものを比較すると, 高速化後の MPDP が, 最大視差が広がると性能が劣化することがわかった. また合成されが, 奥

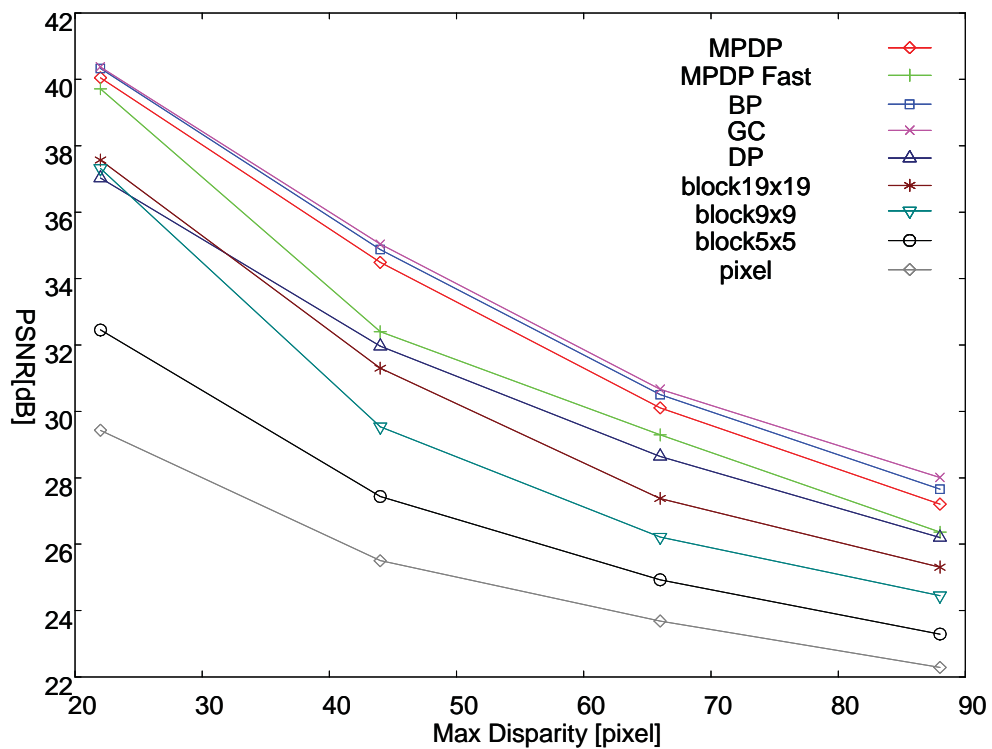


図 4.19 最大視差と PSNR

行き画像を見ると，高速化後のものは前のものに比べてざらつきが多く，滑らかさが弱かった。

しかし，DP と比べ，縦方向の不連続問題は高速化前・後ともに解決されており，PSNR も常時高い結果となった。

グローバル最小化は，前章の背反する問題も解決している．図 4.25，図 4.26 は MPDP と BP の物体境界を重ね合わせたものである．両者ともに，大きなサイズのブロックマッチングのように滑らかに奥行きはつながっているが，境界際のエッジのふくらみが少なく抑えられている．グローバル最小化自体も周囲の連続性を仮定するため，境界領域の奥行きは誤対応が増えるが，大きなブロックマッチングに比べてると影響が少ないと言える。



図 4.20 MPDP の結果 右) 補間画像, 左) 奥行き画像



図 4.21 MPDP (高速化) の結果 右) 補間画像, 左) 奥行き画像



図 4.22 Dynamic Programming の結果 右) 補間画像, 左) 奥行き画像

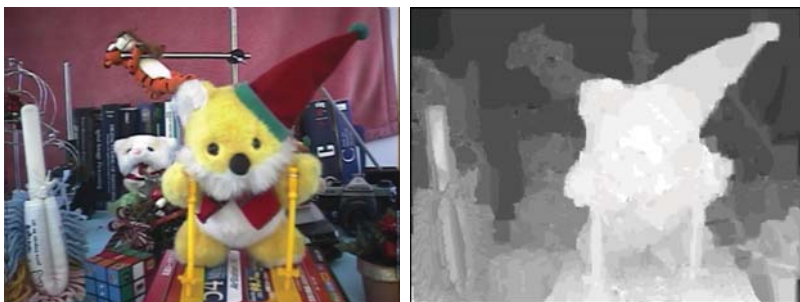


図 4.23 Belief Propagation の結果 右) 補間画像, 左) 奥行き画像



図 4.24 Graph Cut の結果 右) 補間画像, 左) 奥行き画像

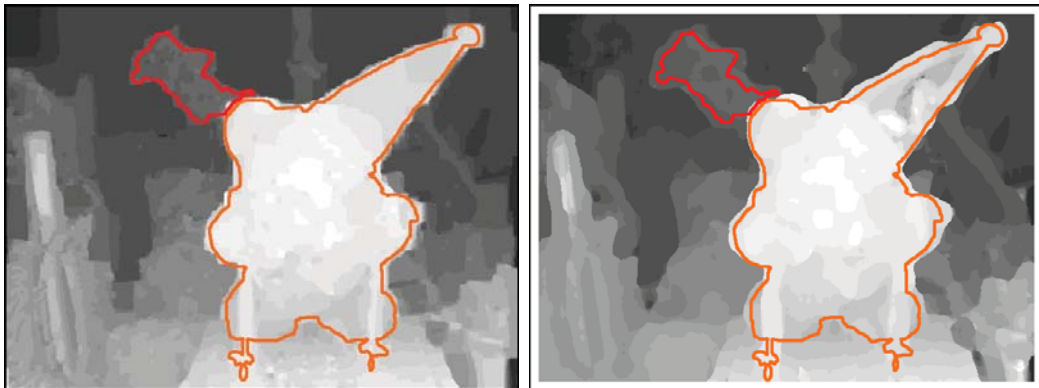


図 4.25 左)BP 右) ブロックマッチング (19 × 19) の奥行き画像. 赤, 橙は
オブジェクトの輪郭

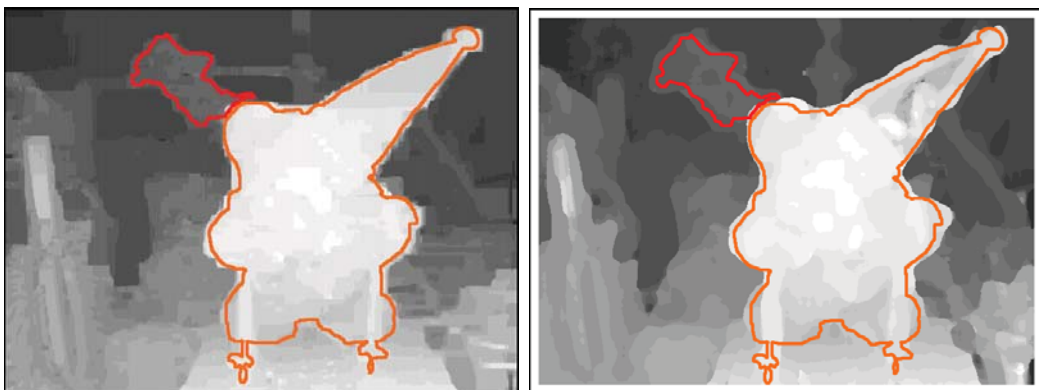


図 4.26 左)MPDP 右) ブロックマッチング (19 × 19) の奥行き画像. 赤,
橙はオブジェクトの輪郭

4.6.3 最大視差と計算時間

前章と同様な条件で, 計算時間を測定した. 結果は, 図4.27である. た
だし, BM19 × 19 と GC は多に比べ計算時間が格段に多いため, 図からは
除外した. また, 最大視差 95 の時の各手法の計算時間を GC, BM19 × 19
を含めて, 図 4.28 に示す.

高速な MPDP は 5 × 5 のブロックマッチング相当, MPDP は 9 × 9 のブ
ロックマッチング相当の計算時間となっている. 一方で, 画質は MPDP

のほうが大幅に高く、最大視差 44 の領域で、5.0dB 高い画像が合成可能となっている。

また、高速化前・後の MPDP は計算コストがオーダーで違うことがわかる。高速化前は $O(d^2)$ の曲線となっており、高速化後は $O(d)$ の線形増加である。

BP の計算 $O(d)$ のコストではあるが、単位あたりのコストが高いため、図 4.27 の範囲では、MPDP のほうが高速となっている。実際使用する場合も、最大視差が 100 を超えるような画像合成は品質も悪いため実用的ではないため、MPDP が有効である。

GC の計算コストは $O(d \log d)$ であるが、BP 以上にコストが大きく、実時間処理には完全に不向きなアルゴリズムである。加えてこのアルゴリズムは並列化が困難でありさらに実行時間が長くなる結果となった。

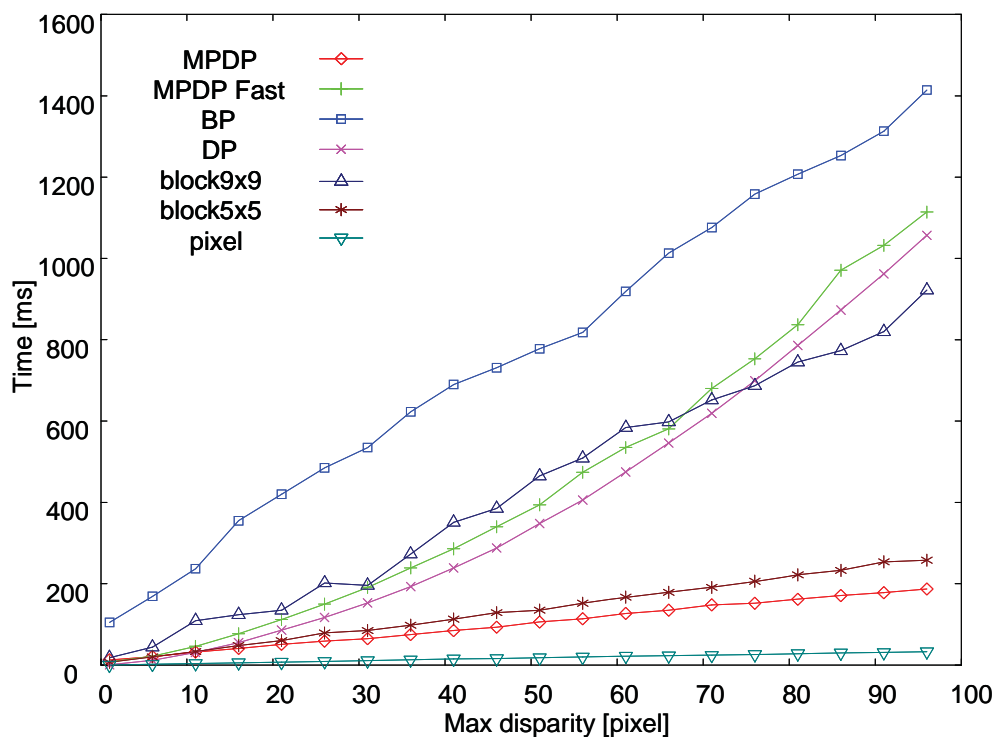


図 4.27 最大視差と計算時間

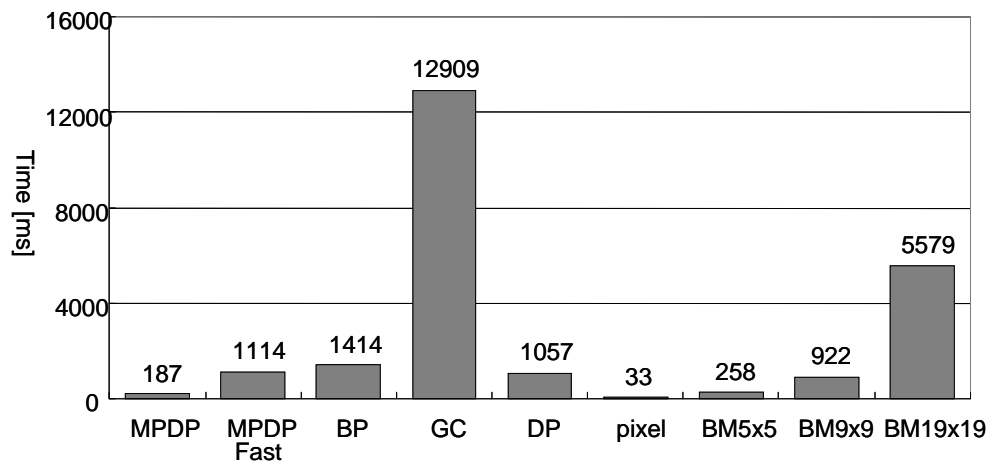


図 4.28 最大視差 95pixel の時の計算時間

4.6.4 コストパフォーマンス

計算時間と精度の計測をふまえ、各手法でコストパフォーマンスを計算した。コストパフォーマンスは式 4.15 で定義し、実験条件は、最大視差 22pixel とした。

$$CP = MSE\sqrt{Time} \quad (4.15)$$

図 4.29 はその結果である。MPDP の高速な場合がもっとも効率がよいことがわかる。また、MPDP が二番目に有効であることがわかった。一方 GC は MSE はもっとも最小化できたが、それに対する計算時間が多くなり、結果が悪くなった。以上から、精度と速度をバランスよく採用したい場合は提案手法が優れていることが明らかとなった。

4.7 結言

本章では、効率的な光線選択のための視点依存奥行き推定法のフレームワークを、また高速かつ高精度な奥行き推定法として、Dynamic Programming を様々な方向から適応する Multi-Pass Dynamic Programming により、視点依存視差推定をより正確に行う手法を提案した。

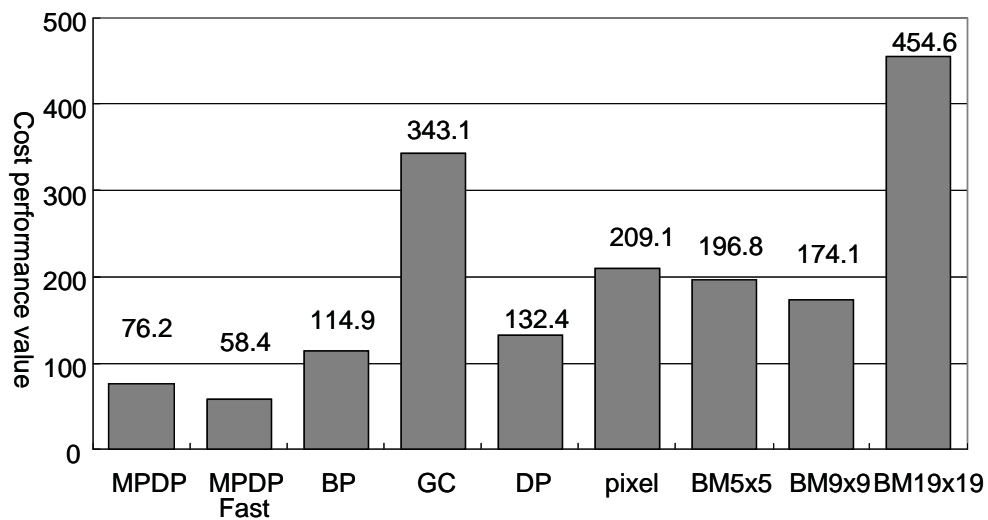


図 4.29 コストパフォーマンス値

前者により、入力のカメラアレイの台数に依存せず、一定の計算コストで自由視点画像合成が可能となり、また後者の手法は、古典的なブロックマッチング手法よりも高精度であり、計算コストの高い他の最適化アルゴリズムに比べて、十分高速な手法であった。具体的には、ブロックマッチングのサイズを 5×5 とした場合（これは、MPDPの高速な場合とほぼ同等な計算速度となる）、PSNRで比較すると最大で7.5dB向上した。また、最大のPSNRを得る事が可能な最適化アルゴリズムであるGraph cutに対して100倍以上速く、またGCとほぼ同等な精度が出る手法であるBPに対して、約10倍高速であった。その際、画質の劣化はGraph Cut, Belief Propagationに対して、物体の隠れなどの影響の小さい、カメラ間隔の近い場合には0.5dB程度であり、離れた場合には約2dB程度であった。また、自由視点画像合成を30fpsのビデオレートで合成するためには、VGAの解像度にて、奥行きが10レベルまでは対応可能であった。もし再現する奥行きがより広い場合は、画像解像度を減らすことや、奥行きレベルを間引くなどの対策が必要であろう。

今後はこのアルゴリズムをより高速に動作させるためにGPUによる

実装を行うことや、いまだ未対応であるオクルージョンへの対応等が必要であろう。

第5章 FTVシステムの構築

5.1 緒言

この章では、2つのFTVシステムの実現例を示す。

1. 実時間で撮影から自由視点画像を出力をするまでを行うFTVシステム
2. 映像である自由視点映像生成に加えて、音声の自由聴点含む自由視点システム

前者は入力画像に色補正，幾何補正，自由視点画像合成を全てリアルタイムに適應することで画像合成するものである。後者は，いったん計算機に保存したデータを用いるものの，音声を含めて自由にユーザが自由に視点，聴点を選ぶことを可能にしたシステムである。

5.2 実時間FTVシステム

実時間でのに自由視点画像合成を行うシステムは，いくつか報告されている。文献 [31] のシステムは，QVGA の 64 眼の web カメラの映像を 1 台のカメラに入力し，リアルタイムに自由視点画像合成が可能である。このシステムでは，色補正，幾何補正等を行っておらず，事前に極力カメラ間の違いをあわせこみ，画像合成側で全てのずれを含めて計算している。全ての処理をグラフィック用の計算ユニットに投げているため高い並列化が行われ，比較的高速に動作はしているが，画質を多少あきらめることでリアルタイム性を実現している。また，この研究室では，カメラ

アレイからではなくレンズアレイによるシステムも提案されている。入力デバイスこそこのなるものの、後段の自由視点画像生成処理はほぼ同じである。

他には、世界初のFTVシステムとなった、谷本研究室の15眼PCクラスタシステムがあげられる。このシステムでは、一対一でカメラとPCを接続し、加えて各PCを光ファイバで結ぶことで分散計算を行えるようになっていて、直線、二次元アレイの場合と提案されているが、必要な光線の選択が非効率であり、また通信のオーバーヘッドなどから、QVGAでの画像合成でも十分なフレームレートが出ていなかった [12]。

ここでは、本論文で提案したアルゴリズムを用いた新たな実時間FTVシステムを示す。

5.2.1 システム構成

本システムは3章で使用した、25眼のカメラアレイシステムを用いる。このカメラはPCIカードにて接続され、理論的な最大フレームレートは約25frame per second(fps)だが、実測値はおおよそ20fpsである。計算機は、Central Processing Unit(CPU)にXEON3.2Gz(Quad Core Dual)を用い、Graphics Processing Unit(GPU)にGeForce 8800GTAを用いた。ここでは、事前に、2章の手法で色変換行列を、3章のノンメトリックな手法で幾何変換行列が計算済みである。

本システムのフローチャートを図5.1に示す。まず、25眼の画像が撮影された後、その未加工画像はグラフィック用の計算ユニットGPUに送られる。このGPU上で各補正行列が適応された後、CPUがアクセス可能な領域にデータが転送され、自由視点画像合成が行われ、最終的な自由視点画像が出力される。自由視点画像合成の最適化アルゴリズムは自由に選択可能となっているが、このシステムでは主には、もっともコストパフォーマンスが高い、高速なMPDPを用いている。

この際、視点位置の入力は毎フレーム受け付けている。GPU、CPU間の同期は1フレーム遅れで行い、色補正・幾何補正、自由視点画像合成

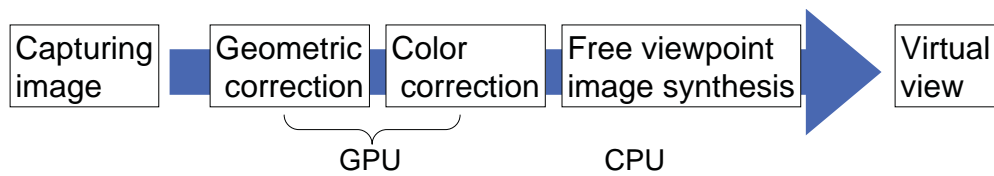


図 5.1 フローチャート



図 5.2 アナグリフ

は、それぞれのユニットで並列に行った。

また、立体出力を可能とするために、1つのフレームに対してステレオペアとなる2つの視点の自由視点画像を生成した。その際は一般的な赤青メガネで確認できる様にアナグリフ(図5.2)にて画像を1枚に統合した。

5.2.2 実行結果

本アプリケーションの Graphical User Interface(GUI) を図 5.3 に示す。このアプリケーションでは、トラックバーにて、 x,y,z 軸のコントロールが可能となっている。また、残るインターフェイスはアナグリフと、普通の自由視点画像出力とアナグリフの切り替えスイッチとなっている。

自由視点画像合成の探索する奥行き範囲を10階調に設定した結果、



図 5.3 グラフィカルユーザインターフェース

本システムでは約 60ms で一枚の自由視点画像を出力可能となった。また、アナグリフの場合、単純に自由視点画像合成処理を 2 度行っているため、倍の 120ms での合成となった。アナグリフへの変換時間は、単純な画像の重ね合わせであり、その計算時間は 1ms 以下となり無視出来る。

このシステムでは、色補正、幾何補正にかかる時間は自由視点画像合成に比べ短く、メモリの転送時間も含めて、全体の計算時間には影響がなかった。

5.2.3 実時間 F T V システムのまとめ

本システムではライブな FTV を鑑賞するためのシステムを提案した。25 眼の二次元カメラアレイを用いて、色補正・幾何補正、補正自由視点画像合成を含む全ての処理を 1 枚の自由視点画像合成を 60ms つまり 16.7fps で、またアナグリフの場合、120ms(8.3fps) で自由視点画像合成が可能と



図 5.4 カメラアレイとマイクロフォンアレイ

なった。

立体ディスプレイなどへ出力する場合，1枚ではなく，2枚以上の画像合成を行う必要がある．今回は，2眼のステレオ画像出力しか行わなかったが，4眼,8眼といった多眼立体ディスプレイに対しては，それぞれに対応した視点数の自由視点画像が必要となる．そのため，線形に増加する本システムでは，多眼ディスプレイに対応するのは困難であろう．今後は，実時間で多眼ディスプレイ用の自由視点画像合成を行うアルゴリズムを考えていく必要があるだろう．

5.3 自由視聴点システム

近年，多数のカメラから3次元情報を蓄積し，その自由視点画像生成の研究も多数行われている [10] しかし，その3次元シーンの情報を取得・再現するときに，音声データまでを再現したものは少ない．ここでは，多数のカメラとマイクからなる，カメラアレイとマイクロフォンアレイを用いて，3次元情報を取得し，そのデータからユーザーが任意の視点・聴点を選ぶことができる，自由視聴点映像生成を行う．

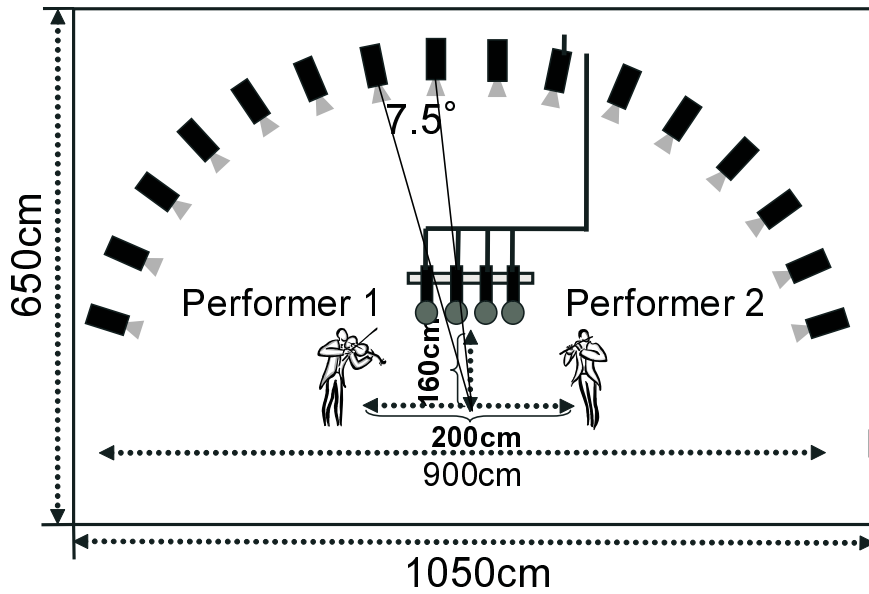


図 5.5 カメラ配置

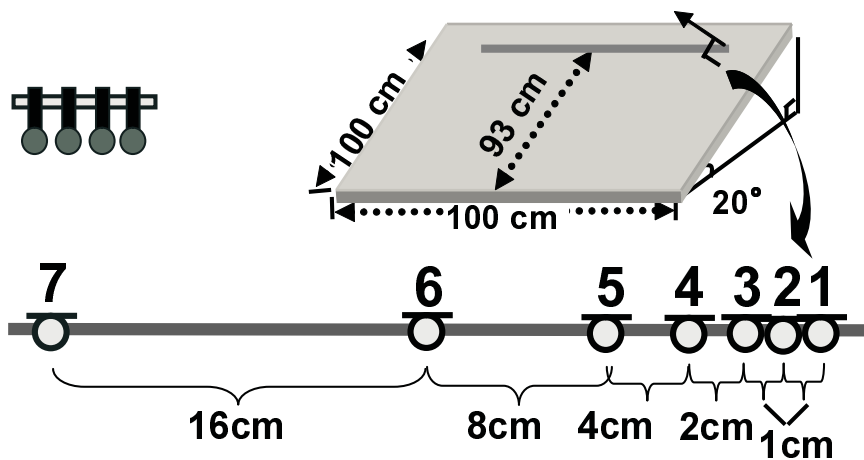


図 5.6 マイク配置

5.3.1 システム構成

本システムは、制御用のサーバと、画像、音声の取得用ノードから構成される。各ノードは1台のカメラ(1392×1040の解像度、30fps)と4台

のマイクを接続可能である。本システムの主な特徴として、全映像・音声データ間の同期が $1 \mu\text{s}$ 以内で取れる点にある。もう一点として、音声・映像ともに非圧縮データが取得可能な点である。前者は、すべてのノードを同期線で直列に結ぶことで達成される。また、後者はシステム構成として、各カメラごとに1台のノードを割り当てることで取得可能としている。両性能ともに、信号処理を研究上で、圧縮された汎用的なものよりも有用なデータとなる。

これらのシステムにより、図5.4中にあるカメラアレイ、マイクロフォンアレイを用いて、音声、映像を同期させて撮影を行った。カメラおよびマイクの配置図を、図5.5,5.6に示す。カメラアレイは、半径4.5mの円周上に7.5度間隔に17台配置し、円の中心を向く形で設置した。また、等比的な間隔で設置したマイクロフォンアレイを円の中心1m手前に設置した。

5.3.2 撮影および自由視聴点生成生成

本システムを用い、楽器演奏を撮影した。2~3人の演奏者は円の中心から半径1m以内かつ、マイクロフォンアレイから1.6m手前に位置して演奏を行った。

この撮影データに対してオフライン処理で、文献[44]の自由視点生成法と、文献[78]の自由聴点生成法により、自由視聴点映像の生成を行った。音声・映像の両処理のフローを図5.7に示す。まずはじめに、カメラやマイクの位置を測定するキャリブレーションを行う。次に、各画像の色合いや歪み、マイクの音量などを補正する。そして、シーンの奥行き情報の計算や音源位置の特定を行ったあとで、自由視点・聴点を生成する。

文献[44]は、Image Based Rendering に属する手法に基づき、物体の奥行き推定結果を用いて画像を3次元的にワーピングさせることで自由視点画像を生成する。文献[78]は、独立成分分析により音源分離を行い、頭部伝達関数を生成することで任意の位置の音源を再現可能となり、自由聴点を生成する。

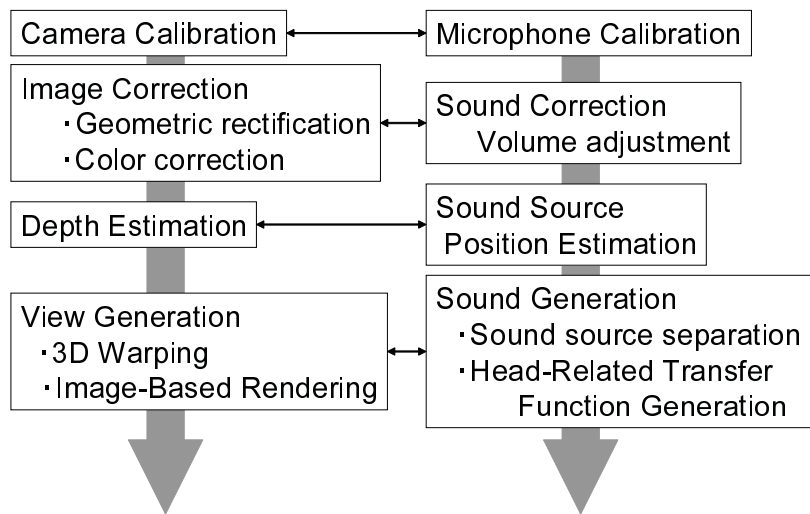


図 5.7 自由視聴点の処理フロー

両者ともに確立されつつある技術ではあるが、2つのメディアを統合するためには、各メディアで用いた時間軸と3次元座標を一致させる必要がある。前者は、音声映像の同期は取得時に完全に取れているシステムを用いたため解決される。また後者は、映像の3次元座標に音声の座標を合わせることで解決させる。実際には、音声・映像ともに別の基準をもって測量をするため、2つの座標系には差異が存在している。しかし、人間の空間認知能力を考慮すると、音声から3次元位置を特定する能力が、映像から特定する能力に対して著しく低いことから、両者の差異はほぼ影響がないと考えられるからである。

5.3.3 結果

本実験での幾何補正は、キャリブレーションベースの手法を用いて行った。音声との統合のためには、実際の3次元座標が必要となるからである。自由視聴点の生成結果を示す。図5.8の左図は生成した自由視点画像である。また、その音声と同期したビデオは、ディスプレイとヘッド

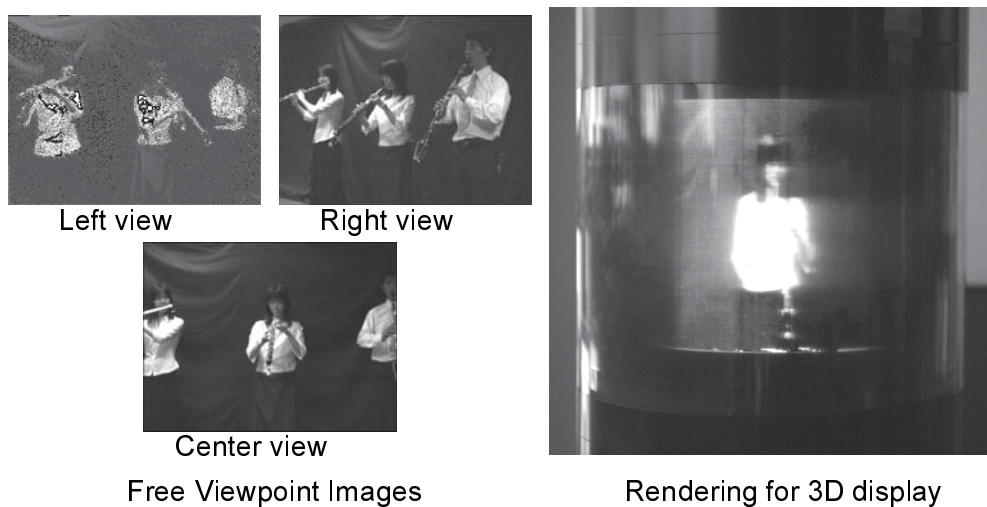


図 5.8 生成画像結果

フォンで再生可能であり，下記 URL[79] からダウンロード可能となっている．また，図 5.8 の右図は文献 [33] の円周立体ディスプレイへ生成した自由視点データを出力した結果である．今後は，これらと立体音源の出力を組み合わせることで臨場感のある再生を目指していく予定である．

5.3.4 自由視聴点のまとめ

ここでは，カメラアレイ，マイクロフォンアレイを用いて，3次元シーンの映像・音声を完全に同期して取得するシステムを構築した．それらを用いて，ユーザが自由に視点や聴点を自由に選択可能となる自由視聴点生成をオフライン処理にて行った．

今後の予定として，今回の自由視聴点生成では音声・映像の各処理は独立に行ったが，同じ3次元シーンから取得される音声・映像間には大きく情報に相関性があるはずである．この相関性を使う，両データにわたる横断的な処理により，物体の位置推定やシーンの状況解析などの精度向上を目指す．

第6章 結論

本論文では、自由視点テレビの実現のために必須となる基盤技術である、

1. 多視点画像の色合いを補正する技術
2. カメラアレイの配置誤差を補正する技術
3. 高速かつ高精細に自由視点画像合成を行う技術

について述べ、それらを用いて自由視点テレビのプロトタイプを作成した。

まず始めに、カメラの固体差により、カメラアレイで撮影した画像の輝度値、色合いが不均一となる問題を解決するために、各画像を色補正する技術について第2章で述べた。反射特性がランバートであるカラーチャートを用いて、撮影の前段階からカメラ間の色合い補正を行った。具体的には、事前処理にて、カメラの光電変換特性がほぼ線形となるように調整し、またカメラの色合いがもっとも近くなるように、各カメラ別々にゲインをセットした。また、その処理の後、各カメラに写るカラーパターンが、カラーパタンの平均色の画像と最小の距離を持つように 3×4 の行列にて線形変換した。

実験の結果、ゲイン調整等を行った後の平均誤差は30.15であったが、補正行列を適応した結果4.96まで抑えることが可能であった。これは使用したカメラのノイズレベルとほぼ同等であり、多視点映像を順番に視点切り替えをして見たときも、画面が点滅することなく、視覚的に良好な結果となっている。また、後述する自由視点画像合成もこの色補正で十分動作することが確認された。

色補正で残る課題は、まず非線形変換への対応が挙げられる。本論文では、入力となる信号自体をルックアップテーブルを用いて極力線形信

号となるように実験を行っていた。しかしながら、多視点映像を撮影する時に、ガンマ関数などのテーブルを用いて撮影する場合がある。これは、光のダイナミックレンジが広い場合、つまり、シーンの明暗のコントラストが強い場合に、色の256階調に、人の目の特性に合わせて有効にビットを割り当てる場合である。その場合、本手法のような線形変換では十分な対応ができない。そのため、RGBが独立かつ、非線形な変換にて、各カメラを補正する手法が必要となるだろう。

また、もうひとつの課題としては、レンズによる収差の補正がある。レンズ中央部に入る光はその周辺部に比べて強くなり、光の強度は周辺部ほど弱くなる。その結果、全く同一の発色をする物体であったとしても、カメラに写る位置によって強度が異なる。これは、マルチカメラだけの問題ではないが、同一の物体を多方向から見るマルチカメラシステムにおいて、この問題は顕著にあわられる。より高精度な色補正のためには、この収差の影響を考慮する必要があるだろう。

次に、カメラを設計通りに配置したつもりでも実際には配置しきれず、加えて、レンズなどの内部構造の不均一さにより、理想的な多視点映像が取得できない問題に対して、所望の位置で撮影した画像へと幾何補正する技術について第3章で述べた。本論文では、比較的カメラ配置に拘束のないカメラキャリブレーションを用いた手法と、精度の高い二次元カメラアレイに特化したノンメトリックな補正手法を提案した。

前者の手法を用いて80台のカメラで構成されるカメラアレイの映像を補正したところ、その補正誤差は0.42pixelとなった。自由視点画像合成に必要な精度である1pixel以内の精度を持ちこの補正により画像合成は十分に働く。後者の手法を用いれば、その誤差は0.35pixelとなり、本キャリブレーションを用いた手法よりも高精度なものとなっている。しかしながら、ノンメトリックな手法は補正に特化するため、カメラの焦点距離や光学中心などカメラパラメータは未知のままである。そのため、パラメータが必要な場合は前者を、必要ない場合は後者のような、より精度の高い物を使用するなど、適時選択が必要であろう。

残る課題として、カメラキャリブレーションを用いる手法の場合、より高精度なカメラキャリブレーションの手法を検討する必要があるだろう。今回はマルチカメラであるにもかかわらず、カメラのキャリブレーション自体は各カメラごとに単体で行っていたためである。各カメラ間での関係性を考慮したカメラアレイに特化したカメラキャリブレーション手法を検討することで、より高精度な補正が可能であると考えられる。また別のアプローチとして、3次元物体を用いる高精度なキャリブレーション手法を適用することが挙げられる。なぜなら今回使用したカメラキャリブレーション手法は簡便に行うことが可能な手法であり、利便性を考慮しなければ、カメラアレイの大きさに合わせた3次元物体を構築することで、今回よりも精度の高いキャリブレーションが可能となるからである。今回の実験の場合、幅4m、高さ2m奥行き10mほどの、カメラアレイに写る領域を占める3次元物体が用意可能ならば、高精度なキャリブレーションが可能となるであろう。

後者のノンメトリックな補正手法は、2次元カメラアレイに特化した補正手法であったため、円形やドーム型など、その他のカメラ配置で適応可能な補正手法を検討していく必要があるだろう。また、その際、カメラアレイ上で成り立つエピポーラ幾何学をより詳しく検討していくことにより、より一般的な手法に発展させる必要があるだろう。

また、4章では、二次元カメラアレイの映像を、上記の2つの技術により補正した画像を用いて、高速かつ高精度に自由視点画像合成を行う手法を提案した。視点依存視差推定を用いて効率的に情報選択するだけでなく、視差推定の段階で、もっとも時間に対する精度の効率の高い最適化手法である、Multi-Pass Dynamic Programming(MPDP)手法を新たに提案した。前者により、入力のカメラアレイの台数に依存せず、一定の計算コストで自由視点画像合成が可能となり、また後者の手法は、古典的なブロックマッチング手法よりも高精度であり、計算コストの高い、他の最適化アルゴリズムに比べて高速計算が可能となった。具体的には、ブロックマッチングのサイズを 5×5 とした場合（これは、MPDPの高速な場合とほぼ同等な計算速度となる）、PSNRで比較すると最大で7.5dB向

上した。また、最大の PSNR を得る事が可能な最適化アルゴリズムである Graph Cut(GC) に対して 100 倍以上速く、また GC とほぼ同等な精度が出る手法である Belief Propagation(BP) に対して、約 10 倍高速であった。その際、画質の劣化は GC, BP に対して、物体の隠れなどの影響の小さい、カメラ間隔の近い場合には 0.5dB 程度であり、離れた場合には約 2dB 程度であった。また、自由視点画像合成を 30fps のビデオレートで合成するためには、VGA の解像度にて、奥行きが 10 レベルまでは対応可能であった。もし再現する奥行きがより広い場合は、画像解像度を減らすことや、奥行きのレベルを間引くなどの対策が必要であろう。

本手法をより高速に動作させる方法として、画像合成を GPU などの並列計算に強いプロセッサ上で実行ことがあげられる。本論文の手法は画像の縦もしくは横の解像度分だけ並列化可能であるにもかかわらず、CPU の並列化は現在は 8 から 16 プロセスほどにとどまる。より多くの並列化が可能な演算装置を用いることで、より高速に動作させることが可能となるだろう。

また、本論文の画像合成では対応していない点は、物体の隠れと鏡面反射などのランバート反射をしない反射特性を持つ物体の合成である。これらの問題に対応するためには、本手法のように近接するカメラの情報だけでは不足しており、入力画像をすべて使う、もしくは、時間方向の相関性を使うなど、より多くの情報を用いなければならない。そのため、多視点映像の時間、空間のすべての情報を使い、画像補間を行うフレームワークを検討していく必要があるだろう。

最後に 5 章では全ての研究をふまえ、F T V のプロトタイプを 2 つ提案した。ひとつはカメラアレイを入力とした実時間自由視点画像合成システムであり、もうひとつは、オフライン処理ながらも音声までも自由に選択可能な自由視聴点システムである。

一つ目のアプリケーションにより、マシンとカメラが直結するような、通信路を考慮しない場合に限り、F T V システムは 60ms というほぼ実時

間¹での実行が可能となっている。このアプリケーションの実装は二次元カメラレイに限定しているため、今後は、よりフレキシブルなカメラ配置を実現するために、円形配置やドーム型配置、さらには自由なカメラ配置による実時間画像合成法を検討していく必要があるだろう。

また二つ目のアプリケーションにより、オフライン計算ながら、映像だけにとどまらず音声までも自由な視点、聴点で合成可能となった。今回の自由視聴点生成では音声・映像の各処理は独立に行ったが、同じ3次元シーンから取得される音声・映像間には大きく情報に相関性があるはずである。この相関性を使う、両データにわたる横断的な処理により、物体の位置推定やシーンの状況解析などの精度向上ができると考えられる。

本研究にて、実時間でのFTVがおおむね完成されており、広帯域な場合のFTVシステムは、実現されてるといえよう。計算機のマルチコア化や高速化が進む中、ビデオレートでかつ、ハイビジョン映像を描画できるような時期は近いであろう。しかしながら、放送や実用化のためには、本研究では深くは立ち入らなかった、圧縮、通信などにも研究課題が多く残る。

圧縮に関しては、現在のMVCなどの符号化形式では、実時間の自由視点画像合成には不向きである。なぜなら、画像合成は合成する視点に応じて必要な視点の情報が増えるため、圧縮されたデータに局所的なアクセスが必要となるからである。今の符号化形式では、部分的なデコードは不可能であり、すべての時間、視点でデコードすることとなり、高い計算能力が必要となる。そのため、自由視点画像合成に適したデータフォーマットやその圧縮方式を、特にローカルデコード可能な形式を検討する必要があるだろう。

また、データ伝送に関しては、QoS(Quality of Service)などを考慮したデータストリーミングについても検討する必要があるだろう。多視点映像を送る上で、どのような情報が臨場感を得るために必要なのかを検討しながら、通信路の環境に応じた制御方法を考えていく必要があるだ

¹VGAサイズの解像度で奥行きを10階層で仮定した場合の、幾何・色補正を含めた処理速度

ろう。

以上、残された課題を解決することにより、自由視点テレビなどの3次元映像通信の実現が促進されると期待される。

参考文献

- [1] 谷本正幸, “自由視点テレビ”, 3次元映像フォーラム, vol. 13, no. 3 pp.17–22, 2001.
- [2] 谷本正幸, “自由視点テレビ F T V”, 映像情報メディア学会誌, vol. 58, no. 7 pp.898–901, 2004.
- [3] 谷本正幸, 藤井俊彰, “自由視点映像技術”, 映像情報メディア学会誌, vol. 60, no. 1, pp.29–34, 2006.
- [4] Bennett Wilburn, Michael Smulski, Kelin Lee, and Mark A. Horowitz, “The Light Field Video Camera”, SPIE Electronic Imaging Conference on Media Processors, 2002.
- [5] Masayuki Tanimoto, Toshiaki Fujii, and Terumasa Aoki, “Test Sequences for Call for Proposals on Multiview Video Coding”, ISO/IEC JTC1/SC29/WG11 M12022, 2005.
- [6] Yaser Jazouane, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “Ray-Space Interpolation for Free Viewpoint Generation”, SIGGRAPH2005, 2005.
- [7] Koji Yamanaka, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “Multiview Interpolation for Ray-Space Coding”, SPIE IT-Com, vol. 6016, pp.60160D-1 – 60160D-10, 2005.
- [8] Jonathan Starck, Gregor Miller, and Adrian Hilton, “Volumetric Stereo with Silhouette and Feature Constraints”, British Machine Vision Conference (BMVC), vol. 3, pp.1189–1198, 2006.

- [9] 韓昇龍, 山崎俊彦, 相澤清晴, “拡張ブロックマッチングによる3次元ビデオの圧縮に関する検討”, 電子情報通信学会 画像工学研究会, 2005.
- [10] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy, “High Performance Imaging Using Large Camera Arrays”, Proc. SIGGRAPH2005, 2005.
- [11] 久保田彰, 児玉和也, 羽鳥好律, “円形多眼画像からの逆フィルタによる中心視点画像の生成手法”, 電子情報通信学会論文誌, vol. J90-D, no. 7, pp. 1714–1715, 2007.
- [12] Purim Na Bangchang, Toshiaki Fujii, and Masayuki Tanimoto, “Experimental System of Free Viewpoint Television”, Proc. SPIE-IST & Electron. Imaging, vol. 5006, 66, pp. 554–563, 2003.
- [13] 文部科学省 21世紀COEプログラム「社会情報基盤のための音声・映像の知的統合」
<http://www.imi.coe.nagoya-u.ac.jp/>
- [14] 藤井俊彰, 森健策, 武田一哉, 間瀬健二, 谷本正幸, 末永康仁, “大規模実世界データベース構築のための多元多点計測装置の開発”, 3次元画像コンファレンス 2006, 3-1, 2006.
- [15] 田口裕一, 苗村健, “自由視点画像合成に基づく光線空間符号化 —カメラアレイ入力への適用 —”, 画像符号化シンポジウム 2005, pp. 31–32, 2005.
- [16] D. Crispell, G. Taubin, and J. Jannotti, “Image Based Routing for Image Based Rendering”, the 6th Workshop on Omnidirectional Vision, Camera Networks, and Non-classical Cameras OMNIVIS2005, 2005.

- [17] 國田豊, 上野雅浩, 田中敬二, “遮蔽領域の再現性を向上させた視点依存信頼度マッピング法”, 第5回情報科学技術フォーラム FIT2006, 2006.
- [18] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski, “High-quality video view interpolation using a layered representation”, *ACM Transactions on Graphics (TOG)*, vol. 23, issue 3, 2004.
- [19] Point Gray Research
<http://www.ptgrey.com/>
- [20] 高田英明, 陶山史朗, 伊達宗和, 中沢憲二, “前後2面のLCDを積層した小型DFDディスプレイ”, *映像情報メディア学会誌*, vol. 58, no. 6, pp.807–810, 2004.
- [21] Fumio Okano, Haruo Hoshino, Jun Arai, and Ichiro Yuyama, “Real-time Pickup Method for a Three-dimensional Image based on Integral Photography”, *Applied Optics*, vol. 36, no. 7, pp.1598–1603, 1997.
- [22] 河宗玄, 小島將, 高橋桂太, 苗村健, “レンズアレイを用いた実時間全焦点自由視点画像合成システム”, *映像情報メディア学会誌*, vol. 59, no. 10, pp.1483–1487, 2005.
- [23] Ren Ng, Marc Levoy, Mathieu Bredif, Gene Duval, Duval Design, Mark Horowitz, and Pat Hanrahan, “Light Field Photography with a Hand-Held Plenoptic Camera”, *Stanford University Computer Science Tech Report CSTR 2005-02*, 2005.
- [24] 眞能啓輔, 圓道知博, 藤井俊彰, 谷本正幸, “ミラー走査による全周囲光線取得実験”, *電子情報通信学会技術研究報告*, Vol.107, No.380 pp. 123–127, 2007.

- [25] 藤井俊彰, 原島博, “構造抽出に基づく三次元画像符号化”, 画像符号化シンポジウム, pp.23-24, 1994.
- [26] 藤井俊彰, 金子正秀, 原島博, “光線群による3次元空間情報の表現とその応用”, テレビジョン学会誌, vol. 50, 9, pp. 1312-1318, 1996.
- [27] Kenji Yamamoto, Masaki Kitahara, Tomohiro Yendo, Toshiaki Fujii, Masayuki Tanimoto, Shinya Shimizu, Kazuto Kamikura, and Yoshiyuki Yashima, “Multiview Video Coding Using View Interpolation and Color Correction”, IEEE Trans. on Circuits and Systems for Video Technology, Vol.17, No.11, pp.1436–1449, 2007.
- [28] J.X. Chai, X Tong, S.C. Chan, and H.Y. Shum, “Plenoptic sampling”, Proc. SIGGRAPH’00, pp.307–318, 2000.
- [29] Middlebury College Stereo Vision Research Page,
<http://cat.middlebury.edu/stereo/>
- [30] 福嶋慶繁, 圓道知博, 藤井俊彰, 谷本正幸, “光線空間の効率的な部分補間法”, 3次元画像コンファレンス 2006, pp.85–88, 2006.
- [31] 高橋桂太, 苗村健, “視点依存奥行きマップ実時間推定に基づく多眼画像からの自由視点画像合成”, 映像情報メディア学会誌, vol. 60, no. 10, pp.1611–1622, 2006.
- [32] 山中康司, “修士論文:グローバルエネルギー最小化による光線空間補間に関する研究”, 名古屋大学大学院, 2006.
- [33] 圓道知博, “周囲360度から観察できる3次元ディスプレイ Seelinder”, 映像情報メディア学会誌 vol.59, pp. 1506–1509, 2005.
- [34] 大井隆太郎, 奥井誠人, “実写を入力としたフレネルホログラムの空間表示”, 映像情報メディア学会技術報告, AIT2006-124, vol. 30, no. 43, pp.3–6, 2006.

- [35] 山本健詞, 圓道知博, 藤井俊彰, 谷本正幸, David Suter, “Colour Correction for Multiple-camera System by Using Correspondences”, 映像情報メディア学会誌, vol. 61 no. 2 pp. 213 – pp. 222 2007.
- [36] Adrian Ilie, and Greg Welch, “Ensuring Color Consistency across Multiple Cameras”, Proc, ICCV05, pp. II:1268–1275, 2005.
- [37] Neel Joshi, Bennett Wilburn, Vaibhav Vaish, Marc Levoy, and Mark Horowitz, “Automatic Color Calibration for Large Camera Arrays”, UCSD CSE Technical Report CS2005-0821, 2005.
- [38] Ulrich Fecker, Marcus Barkowsky, and Andre Kaup, “Improving the Prediction Efficiency for Multi-View Video Coding Using Histogram Matching”, Proc. PCS2006, P2-16, 2006.
- [39] Kwanghoon Sohn, Yongtae Kim, Jungdong Seo, Jaewon Yoon, Changseob Park, and Junyong Lee, “H.264/ AVC-compatible Multi-view Video Coding”, ISO/IEC JTC1/SC29/WG11 M12874, 2006.
- [40] Yoshihiro Kawai and Fumiaki Tomita, “Intensity Calibration for Stereo Images Based on Segment Correspondence”, IAPR Workshop on Machine Vision Applications, MVA’98, pp. 331–334, 1998.
- [41] Yushan Chen, Jianle Chen, and Canhui Cai, “Luminance and Chrominance Correction for Multi-View Video Using Simplified Color Error Model”, PCS2006, P2-17, 2006.
- [42] David G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” International Journal of Computer Vision, vol.60, no.2, pp. 91-110, 2004.
- [43] M Levoy, and P Hanrahan, “Light Field Rendering”, Proc. ACM SIGGRAPH ’96, pp. 31–42, 1996.

- [44] Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “Free Viewpoint Image Generation Using MultiPass Dynamic Programming”, Proc.SPIE SD&A XIV, vol.6490, pp. 460–470, 2007.
- [45] Z. Zhang, “A Flexible New Technique for Camera Calibration”, IEEE Trans. PAMI, Vol.22, No.11, pp.1330–1334, 2000.
- [46] R.Y. Tsai, “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.3, pp.323–344, 1987.
- [47] P.F. Sturm, and S.J. Maybank, “On Plane-based Camera Calibration: A General Algorithm, Singularities, Applications”, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol.1, pp.432–437, 1999.
- [48] 植芝俊夫, 富田文明, “平面パターンを用いた複数カメラシステムのキャリブレーション”, 情報処理学会論文誌: コンピュータビジョンとイメージメディア, vol.44, no.SIG 17(CVIM 8), pp.89–99, 2003.
- [49] P. Baker, and Y. Aloimonos, “Calibration of A Multicamera Network”, In Proceedings of the IEEE Workshop on Omnidirectional Vision and Camera Networks, 2003.
- [50] X. Meng, and Z. Hu, “A New Easy Camera Calibration Technique Based on Circular Points”, Pattern Recognition, vol.36, issue 5, pp.1155–1164, 2003.
- [51] M. Pollefeys, R. Koch, and L. Van Gool, “Self-calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters”, International Journal of Computer Vision, vol.32, no.1, pp.7–25, 1999.

- [52] T. Svoboda, H. Hug, and L. Van Gool, "ViRoom - Low Cost Synchronized Multicamera System and Its Self-calibration", In Proceedings of the Pattern Recognition 24th DAGM Symposium, no.2449 in LNCS, pp. 515–522, 2002.
- [53] 岡谷貴之, 久保田俊作, 出口光一郎, "カメラモデルの段階的切換えによる多視点カメラシステムのセルフキャリブレーション", 情報処理学会論文誌: コンピュータビジョンとイメージメディア, vol.43, no.SIG 11(CVIM 5), pp.33–42, 2002.
- [54] N. Ayache, C. Hansen, "Rectification of Images for Binocular and Trinocular Stereovision", 9th International Conference Pattern Recognition 1988, vol.1, pp. 11–16, 1988.
- [55] R Swaminathan, SK Nayar, "Nonmetric Calibration of Wide-angle Lenses and Polycameras", IEEE Trans. PAMI, Vol. 22, No. 10, pp. 1172 –1178, 2000.
- [56] C. Loop and Z. Zhang, "Computing Rectifying Homographies for Stereo Vision", Proc. IEEE CVPR, Vol. 1, pp. 125 –131, 1999.
- [57] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy, "Using Plane + Parallax for Calibrating Dense Camera Arrays" proc. CVPR 2004, val. 1, pp. 1-2 – 1-9, 2004.
- [58] 松本健太郎, 圓道知博, 藤井俊彰, 谷本正幸, "自然特徴点を用いたマルチカメラ画像の幾何補正", 電子情報通信学会論文誌 D, vol. J90, no. 7, pp. 1718–1720, 2007.
- [59] 松本健太郎, 圓道知博, 藤井俊彰, 谷本正幸, "3次元画像処理のためのマルチカメラ画像の幾何補正", Proc. of The 10th Image Media Processing Symposium, pp. 89–90, 2005.

- [60] K. Deng, L. Wang, Z. Lin, T. Feng, and Z. Deng, "Correction and Rectification of Light Fields", In *Computers & Graphics.* , 27(2), pages 169–177, 2003.
- [61] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision Second Edition", Cambridge University Press, Cambridge, UK, 2003.
- [62] R. Szeliski, "Video Mosaics for Virtual Environments", *IEEE Computer Graphics and Applications*, pp. 22–30, 1996.
- [63] M. Heinrichs, and O. Rodehorst, "Trinocular rectification for various camera setups" *Proc. Photogrammetric Computer Vision'06*, pp. 43–48, 2006.
- [64] P.J. Rousseeuw, "Least Median of Squares Regression" *J. American Stat. Assoc.*, Vol.79, pp. 871–880. 1984.
- [65] A. Laurentini, "The Visual Hull Concept for Silhouette-based Image Understanding", *IEEE Trans. PAMI*, vol. 16, issue. 2, pp. 150–162, 1994.
- [66] 上田恵, 有田大作, 谷口倫一郎, "多視点動画画像処理による3次元モデル復元に基づく自由視点画像生成のオンライン化: PC クラスタを用いた実現法", *情報処理学会論文誌*, vol. 46, no. 11, pp. 2768–2778, 2005.
- [67] T. Matsuyama, X. Wu, T. Takai, and T. Wada, "Real-Time Dynamic 3D Object Shape Reconstruction and High-Fidelity Texture Mapping for 3D Video", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. CSVT-14, no. 3, pp. 357–369, 2004.
- [68] N. Inamoto, and H. Saito, "Free Viewpoint Video Synthesis and Presentation from Multiple Sporting Videos," *Electronics and Com-*

- munications in Japan (Part III: Fundamental Electronic Science), vol. 90, issue 2, pp. 40–49, 2007.
- [69] 高橋桂太, 苗村健, “視点依存奥行きマップ実時間推定に基づく多眼画像からの自由視点画像合成”, 映像情報メディア誌, vol. 60, no. 10, pp.1611–1622, 2006.
- [70] 小島將, 苗村健, “レンズアレイを用いた自由視点画像合成システムにおける撮像光学系の空間周波数特性,” 映像情報メディア学会誌, vol. 59, no. 4, pp.569–580, 2005.
- [71] 田口裕一, 高橋桂太, 苗村健, “ネットワークカメラアレイを用いた実時間全焦点自由視点映像合成システム”, 信学技報, Vol. 107, No. 539, PRMU2007-258, pp. 79–86, 2008.
- [72] A. Ishikawa, M. P. Tehrani, S. Naito, S. Sakazawa, and A. Koike, “Free Viewpoint Video Generation for Walk-through Experience Using Image-based Rendering,” Proc. of the 16th ACM international conference on Multimedia, pp. 1007-1008, 2008.
- [73] Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, ”Real-time Arbitrary View Interpolation and Rendering System Using Ray-Space”, Proc. SPIE vol.58 6016, 2005.
- [74] Y. Ohta, and T. Kanade, “Stereo by Intra- and Inter-scanline Search Using Dynamic Programming”, IEEE Trans. PAMI, Vol. 7, No. 2, pp. 139–154, 1985.
- [75] S. Li, “Markov Random Field Modeling in Computer Vision“, Springer Verlag, 1995.
- [76] G. Borgefors, “Distance Transformations in Digital Images”, Computer Vision, Graphics and Image Processing,34(3), pp. 344–371, 1986.

- [77] 金時煥, 河宗玄, 田口裕一, 高橋桂太, 苗村健, ”自由視点画像合成における見え方を考慮したレンズアレイ撮像系の設定”, 映像情報メディア学会誌, Vol. 60, No. 10, pp. 1658–1663, 2006.
- [78] 丹羽健太, 西野隆典, 武田一哉, ”ブラインド音源分離と頭部伝達関数を用いた音楽演奏を題材とした自由視聴点コンテンツの制作”, 信学技報 (EA2007-32), vol.107, no.120, pp.31–36, 2007.
- [79] <http://www.sp.m.is.nagoya-u.ac.jp/~niwa/fvlp.html>

研究業績

学術論文

1. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, “Multi-Pass Dynamic Programming による光線空間補間”, 電子情報通信学会論文誌 D, vol. J90-D, No. 7, pp. 1721–1725, (2007 年 7 月)
2. 福嶋 慶繁, 松本 健太郎, 圓道 知博, 藤井 俊彰, 谷本 正幸, “特徴点軌跡並行化による 2 次元カメラアレイレクティフィケーション”, 映像情報メディア学会誌, vol. 61, no. 4, pp. 564–571, (2008 年 4 月)
3. 福嶋 慶繁, 丹羽 健太, 圓道 知博, 藤井 俊彰, 谷本 正幸, 西野 隆典, 武田 一哉, “多視点・多聴点データ取得システムを用いた自由視聴点映像生成”, 電子情報通信学会論文誌 D, vol. J91-D, No. 8, pp. 2039–2041, (2008 年 8 月)
4. Yuji Mori, Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “View Generation with 3D Warping Using Depth Information for FTV” Signal Processing: Image Communication (January 2009)

国際会議

1. Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “Real-time Arbitrary View Interpolation and Rendering

- System using Ray-Space”, Proc. of SPIE OpticsEast ITCOM, vol. 6016, pp. 250-261, (January 2005)
2. Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “Synthesizing Wide-angle and Arbitrary View-point Images from a Circular Camera Array”, Proc. of SPIE Stereoscopic Displays and Virtual Reality Systems XIII, 6055N-31 (January 2005)
 3. Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “Free View-Point Image Generation Using Multi-Pass Dynamic Programming”, Proc. of Fourth Symposium on “Intelligent Media Integration for Social Information Infrastructure”, pp. 117–118, (December 2006)
 4. Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “Free Viewpoint Image Generation Using Multi-Pass Dynamic Programming”, Proc. of SPIE Stereoscopic Displays and Virtual Reality Systems XIII, 6490A-59, (January 2007)
 5. Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “An Optimization Method For Free View-Point Image Generation Using Ray-Space”, Proc. of IWAIT2007, P6-26, pp. 896-901, (January 2007)
 6. Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “Free Viewpoint Image Generation Synchronized with Free Listeningpoint Audio for 3-D Real Space Navigation”, Proc. of 3DTV-CON2007, 121, (May 2007)
 7. Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto, “A Novel Rectification Method for Two-Dimensional Camera Array by Parallelizing Locus of Feature Points”, Proc. of IWAIT2008, B5-1, (January 2008)

8. Yuji Mori, Norishige Fukushima, Toshiaki Fujii, and Masayuki Tanimoto, “View Generation with 3D Warping Using Depth Information for FTV”, Proc. of 3DTV-CON2008, pp. 229-232, (May 2008)
9. Mehrdad Panahpour Tehrani, Kenta Niwa, Norishige Fukushima, Yasushi Hirano, Toshiaki Fujii, Masayuki Tanimoto, Kazuya Takeda, Kenji Mase, Akio Ishikawa, Shigeyuki Sakazawa, and Atsushi Koike, “3DAV Integrated System Featuring Arbitrary Listening-point and Viewpoint Generation”, Proc. of MMPS2008 PID-213, (October 2008) (The best paper of the MMSP 2008 from IEEE signal processing society.)
10. Makoto Ota, Norishige Fukushima, Tomohiro Yendo, Masayuki Tanimoto, and Toshiaki Fujii, “Rectification of Pure Translation 2D Camera Array”, Proc. of IWAIT2009, 0044 (January 2009)

研究会／大会

1. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”100眼システムのための高精細リアルタイム自由視点画像生成システムの実装”, 電気学会東海支部「若手セミナー」(2005年3月)
2. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”光線に基づく実時間自由視点画像生成システム”, 3次元画像コンファレンス2006講演論文集, pp. 25-28 (2005年7月)
3. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”光線空間法に基づく広角自由視点画像生成”, 平成18年度電気関係学会東海支部連合大会講演論文集, O-263 (2005年9月)
4. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”光線空間法に基づく実時間広角自由視点画像合成”, 信学技報, SIP2005-84, vol.105, no. 292, pp. 61-66 (2005年9月)

5. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”光線空間法に基づく実時間広角自由視点画像合成”, 日本バーチャルリアリティ学会第10回大会予稿集, 2B2-2 (2005年9月)
6. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”光線空間法による自由視点画像生成の高速化”, 第10回映像メディア処理シンポジウム講演論文集, pp. 57-58 (2005年11月)
7. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”光線空間の効率的な部分補間法”, 3次元画像コンファレンス2006講演論文集, pp. 85-88 (2006年7月)
8. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”高能率自由視点画像生成アルゴリズム”, 第1回イメージクオリティとその応用ワークショップ論文集, A2-3, pp. 145-150 (2006年9月)
9. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”Multi-Pass Dynamic Programming による自由視点画像合成”, 平成18年度電気関係学会東海支部連合大会講演論文集, O-418 (2006年9月) (連合大会奨励賞受賞)
10. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”Multi-Pass Dynamic Programming による光線空間補間”, 第11回映像メディア処理シンポジウム講演論文集, I-4.12, pp. 81-82 (2006年11月)
11. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”特徴点軌跡並行化による2次元カメラアレイレクティフィケーション”, 3次元画像コンファレンス2007講演論文集, pp. 85-88 (2007年7月)(3次元画像コンファレンス2007優秀論文賞)
12. 太田 真人, 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, ”2次元配置カメラによる光線空間構築のための幾何補正”, 電気学会東海支部「若手セミナー」(2007年8月)

13. 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, "無限遠点と自然特徴点によるカメラアレイ Rectification", 平成 19 年度電気関係学会東海支部連合大会講演論文集, O-358 (2007 年 9 月)
14. 太田 真人, 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, "機械制御二次元カメラアレイにより取得された光線空間データの幾何補正", 平成 19 年度電気関係学会東海支部連合大会講演論文集, O-359 (2007 年 9 月)
15. 福嶋 慶繁, 丹羽 健太, 圓道 知博, 藤井 俊彰, 谷本 正幸, 西野 隆典, 武田 一哉, "多視点・多聴点データからの自由視聴点映像生成" 第 12 回映像メディア処理シンポジウム講演論文集, I-4.14, pp. 95-96 (2007 年 11 月)
16. 森 勇二, 福嶋 慶繁, 藤井 俊彰, 谷本 正幸, "奥行き画像を用いた 3D warping による自由視点画像生成", 電子情報通信学会 2008 年総合大会, D-11-7 (2008 年 3 月)
17. 太田 真人, 福嶋 慶繁, 藤井 俊彰, 谷本 正幸, "機械制御カメラアレイの幾何補正", 電子情報通信学会 2008 年総合大会, D-11-9 (2008 年 3 月)
18. 太田 真人, 福嶋 慶繁, 圓道 知博, 藤井 俊彰, 谷本 正幸, "並進移動の機械制御によるカメラアレイの幾何補正", 3 次元画像コンファレンス 2008 講演論文集, 2-1, pp. 13-16 (2008 年 7 月)
19. 鈴木 和真, 福嶋 慶繁, 圓道 知博, 谷本 正幸, 藤井 俊彰, "並列処理を用いた自由視点画像合成", 平成 20 年度電気関係学会東海支部連合大会講演論文集, O-225 (2008 年 9 月)
20. 太田 真人, 福嶋 慶繁, 圓道 知博, 谷本 正幸, 藤井 俊彰, "並進移動カメラによる二次元カメラアレイの幾何補正", 平成 20 年度電気関係学会東海支部連合大会講演論文集, O-230 (2008 年 9 月)

21. 太田 真人, 福嶋 慶繁, 圓道 知博, 谷本 正幸, 藤井 俊彰, ”並進移動カメラで構成した二次元カメラレイの幾何補正”, 第13回映像メディア処理シンポジウム講演論文集, I-4.09, pp. 87–88 (2008年10月)

MPEG/JVT 標準化活動

1. Norishige Fukushima, Toshiaki Fujii, and Masayuki Tanimoto, “1D Parallel Test Sequences for MPEG-FTV”, ISO/IEC JTC1/SC29/WG11, M15378, (April 2008)
2. Norishige Fukushima, Toshiaki Fujii, and Masayuki Tanimoto, “Adjusting method for Multi View Image, Color and Geometry Correction for MPEG-FTV Test Sequences”, ISO/IEC JTC1/SC29/WG11, M15379, (April 2008)
3. Norishige Fukushima, Toshiaki Fujii, and Masayuki Tanimoto, “Depth Estimation using Stereo Camera Rectification for Non-parallel Camera Setup”, ISO/IEC JTC1/SC29/WG11, M15380, (April 2008)
4. Kazuyoshi Suzuki, Norishige Fukushima, Toshiaki Fujii and Masayuki Tanimoto, “Reference Softwares for Depth Estimation and View Synthesis”, ISO/IEC JTC1/SC29/WG11, M15377, (April 2008)

表彰

1. 2006年度IEEE名古屋支部学生奨励賞
2. 2006年度電子情報通信学会東海支部学生研究奨励賞
3. 3次元画像コンファレンス2007 優秀論文賞
4. The best paper of the MMSP 2008 from IEEE signal processing society