

# コンピュータ理工学特別研究報告書

## 題目

Lytro Illum を用いたマルチビューステレオに関する研究

学生証番号 1244226

氏名 小川 真司

提出日 平成 28 年 2 月 2 日

指導教員 蚊野 浩

京都産業大学  
コンピュータ理工学部

## 要約

Lytro Illum で撮影した生画像を多数のマルチビューステレオ画像に分解できる。本研究の目的は、このマルチビューステレオ画像から、被写体の実空間における定量的な 3 次元座標を推定することである。

マルチビューステレオ画像に対するステレオマッチングアルゴリズムとしてマルチベースライン法がある。Illum から生成したマルチビューステレオ画像を標準的なステレオ画像と仮定して、マルチベースライン法を適用することは容易である。一方、その計算結果を定量的な 3 次元座標に変換するためには、事前に、ステレオカメラキャリブレーションが必要である。

Illum のマルチビューステレオ画像にステレオカメラキャリブレーションを適用すると、ステレオ画像ごとに画像面の位置、光軸の方向、焦点距離、レンズ歪みなどが算出される。これらの数値には誤差が生じるため、標準的なステレオ画像とは異なるものになる。従って、キャリブレーション済みのマルチビューステレオ画像に、単純にマルチベースライン法を適用することができない。また、本論文の研究を実施するために、OpenCV に実装されている関数を利用したが、その中のステレオマッチングアルゴリズムは、水平方向に基線を持つ標準ステレオ画像だけを処理可能である。

以上のこと考慮して、本研究では、次の手順でマルチビューステレオ画像から、定量的な 3 次元座標の元になる視差画像を推定した。

- (1) ステレオ画像対ごとにステレオカメラキャリブレーションを行う。
- (2) ステレオ画像対ごとに平行化を行う。
- (3) 平行化したステレオ画像から視差画像を推定する。
- (4) 視差画像に逆平行化を行う。
- (5) 逆平行化した視差画像の値を基線長でスケーリングした後、平均化する。

マルチベースライン法で推定した視差画像と、提案手法で推定した視差画像を比較した。マルチベースライン法で得る視差画像は物理的な距離との関係が不明である。提案手法で得る視差画像は、物理的な距離との関係が明確である。一方、提案手法では視差情報を統合するために、全ての視差画像を平均化した。それぞれの視差画像の値には外れ値が含まれるため、平均化処理によって、誤差が大きくなつたと考えられる。また、平行化と逆平行化の処理によって、斜め方向のステレオ画像は画像の端の部分が切れる。これらの影響を受けて、統合された視差画像の有効範囲は円形になった。

## 目次

|                              |          |
|------------------------------|----------|
| 1 章 序論                       | · · · 1  |
| 2 章 Lytro Illum とマルチベースライン法  | · · · 3  |
| 2.1 Illum の特徴と機能             | · · · 3  |
| 2.2 Illum からのマルチビューステレオ画像の生成 | · · · 4  |
| 2.3 マルチベースライン法               | · · · 5  |
| 3 章 Illum を用いたマルチビューステレオ     | · · · 7  |
| 3.1 生画像から多眼ステレオカメラ画像への変換     | · · · 7  |
| 3.2 多眼ステレオカメラ画像のキャリブレーション    | · · · 7  |
| 3.3 ステレオ画像の平行化と逆平行化          | · · · 9  |
| 3.4 複数の視差画像を統合するための手法        | · · · 10 |
| 4 章 実験結果と考察                  | · · · 12 |
| 4.1 ステレオ画像の平行化と逆平行化の検証       | · · · 12 |
| 4.2 複数の視差画像の統合               | · · · 21 |
| 4.3 考察                       | · · · 22 |
| 5 章 結論                       | · · · 24 |
| 参考文献                         | · · · 25 |
| 謝辞                           | · · · 25 |
| 付録                           | · · · 26 |

## 1章 序論

ステレオ三次元計測（ステレオビジョン）は、一つの被写体を異なる位置から撮影した複数の画像を使って、三角測量の原理で被写体までの距離を測定する技術である。2台のカメラを水平に配置したステレオカメラを用いることが多い。2台のカメラが同じ性能（画像センサの大きさや画素数、焦点距離などが同じ）で、画像面が同一平面上にあり、お互いの光軸が平行な場合を標準的なステレオカメラとよぶ。標準ステレオで、お互いのカメラが真横（画像センサの水平走査線が重なる）の場合を平行ステレオとよぶ。平行ステレオで撮影されたステレオ画像を平行化されたステレオ画像とよぶ。

ステレオビジョンは自動車に搭載して前方障害物までの距離を測定するために利用されている[2]。また、人が立ち入り難い場所の現状把握や、建物・ビルの屋上などを再現すること、および自然の景観や地形の評価・シミュレーションに活用されている[3]。

図 1.1 はステレオカメラで撮影した画像の例である。2枚の画像を重ねて比較すると、近くにある被写体の像のずれは小さく、遠くにある被写体の像のずれは大きい。この像のずれ量を視差とよぶ。被写体までの距離と視差は反比例の関係にあることから、被写体までの距離を求めることができる。また、画像内の全ての画素に対して視差を求め、それを画像化したものを見差画像(図 1.2)とよぶ。

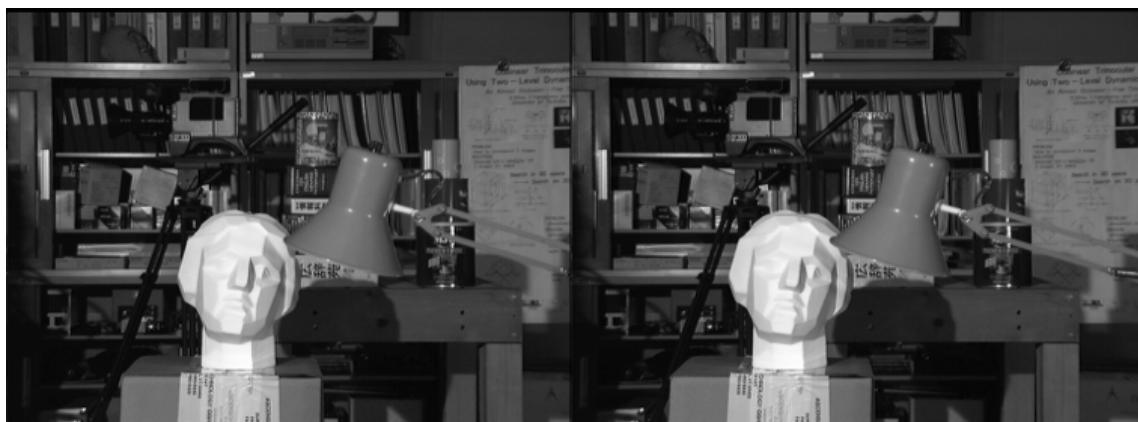


図 1.1 左カメラで撮影した画像(左)と右カメラで撮影した画像(右)

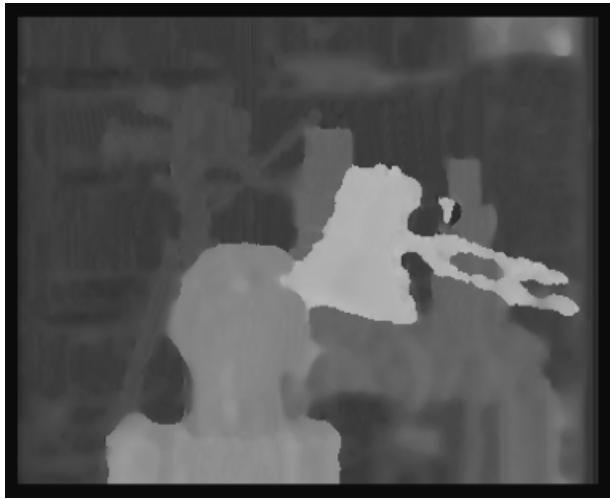


図 1.2 図 1.1 から計算した視差画像  
(引用元:<http://disparity.wikidot.com/>)

ステレオ 3 次元計測で 3 箇所以上の視点を用いるものをマルチビューステレオとよぶ。マルチビューステレオは、2 視点のステレオと比べて距離の測定精度が高く誤計測の可能性が小さい。

本論文はライトフィールドカメラである Illum[1]を使ったマルチビューステレオについて記述する。Illum は、画像センサ表面にマイクロレンズアレイを装着している。主レンズによってマイクロレンズアレイ表面に形成される光の像は、マイクロレンズの働きによって光の方向ごとに分解され画像センサに記録される。このようにライトフィールドカメラ Illum は、レンズに入射する光を光線に分解して記録するカメラである。そして、撮影したあとで、光線集合に対する演算によってピントの位置を変えたり、ぼけぐあいを調節することができる。

Illum が撮影する光線集合を処理することで、容易にマルチビューステレオ画像に変換することができる。本論文では、Illum の生画像から生成したマルチビューステレオ画像を処理することで 1 枚の視差画像を計算する方法について研究した。以下、2 章では Illum とマルチベースライнстレオ法について説明する。3 章では、Illum を用いたマルチビューステレオに関する提案手法を説明する。4 章で結果と考察、5 章で結論を述べる。

## 2章 Lytro Illum とマルチベースライン法

### 2.1 Illum の特徴と機能

一般的なデジタルカメラは、画像センサで被写体像の光の強さを記録する。一方、Illum(図 2.1)は画像センサ表面に「マイクロレンズアレイ」という非常に小さなレンズの配列が設置され、その働きによって、被写体像の光を方向ごとに分解して記録する。すなわち、デジタルカメラは被写体像を記録するカメラ、Illum は被写体像の元になる光線集合を記録するカメラである。



図 2.1 Lytro Illum の外観

(<https://store.lytro.com/products/lytro-illumin> より引用)

Illum のマイクロレンズアレイは、微小レンズが六角格子状に並んだ構造になっている。マクロレンズに入射する光は、マイクロレンズを通過する角度によって、複数の画素のうちのどの画素に集まるのか決まっている。つまり、各画素に集められた光がもともとどの方向から、カメラのレンズに入ってきたのかを逆算することができる。これにより、Lytro Illum で撮影した生画像を処理することで、自由にピントの位置を変えたり、ぼけぐあいを調整することが可能になる。

## 2.2 Illum からのマルチビューステレオ画像の生成

Illum で撮影した画像は, Lytro Desktop という専用アプリケーションを使って, 画像センサの生画像を含む lfr ファイルとして書き出すことができる. lfr ファイルからマルチビューステレオ画像に変換するには, 研究室で開発済みのプログラム IllumLfrRead.cpp と, IllumMultiviewImage.cpp を用いる.

IllumLfrRead.cpp は lfr ファイルを入力して, その中の画像センサの生画像を  $7728 \times 5368$  画素のカラーfiltrタアレイ画像 IllumRaw.tif として保存する. 続いて, IllumRaw.tif をカラー化したデモザイク画像を IllumRGB.tif として保存する. 処理の流れは図 2.2 になる.

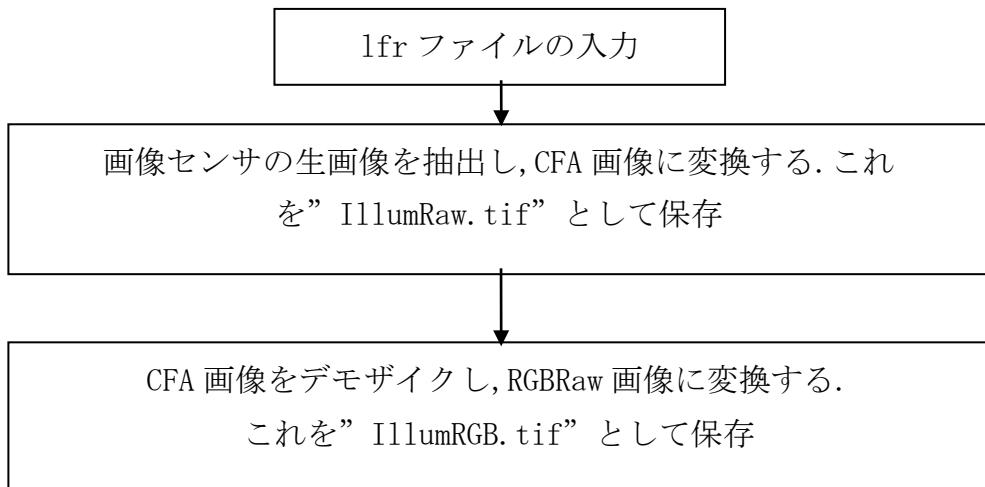


図 2.2 IllumLfrRead の流れ

IllumMultiviewImage.cpp は IllumRGB.tif を入力し, マイクロレンズに対して同じ位置にある画素を, マイクロレンズアレイの配列にしたがって再配置した画像を生成して保存する. 処理の流れは図 2.3 になる. 最終的に生成する一枚一枚の画像は  $1080 \times 752$  画素のカラー画像である.

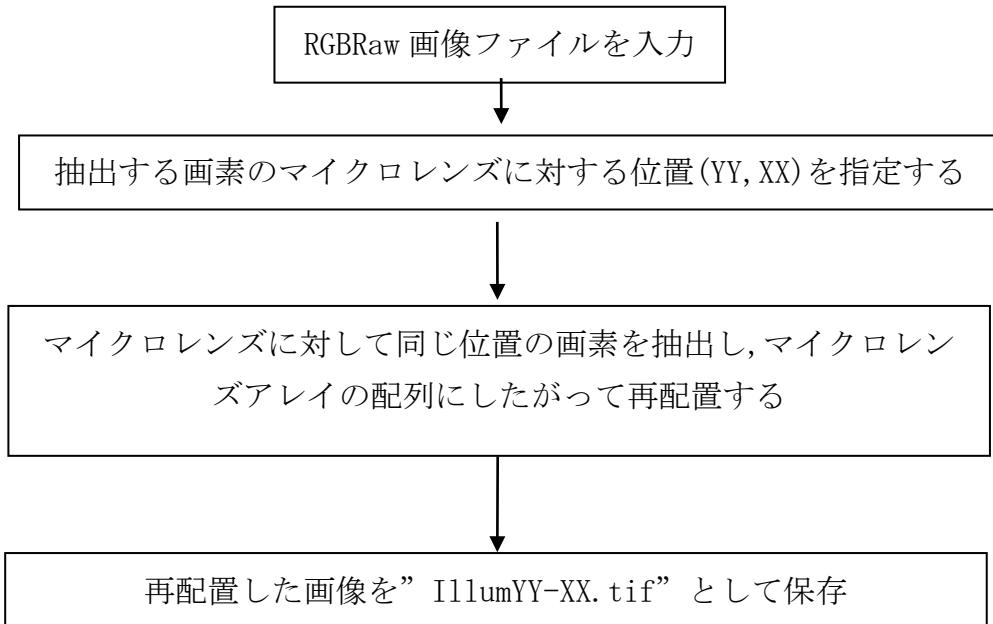


図 2.3 IllumMultiviewImage.cpp の流れ

### 2.3 マルチベースライン法

マルチビューステレオ画像を処理するステレオマッチングアルゴリズムとしてマルチベースライン法がある。この方法は、1つの基準となるカメラに対して、様々な方向や長さの基線のステレオペアから得る距離の手がかりを統合して、一つの視差画像を得る方法である。平行ステレオの場合、異なるステレオ画像対の視差を基線長で割ることで、被写体までの奥行き方向の距離に対して、正規化された視差値になる。各ステレオ画像から得る、正規化された視差に対する SSD(Sum of Squared Difference)を足し合わせ、そこから最も確からしい対応位置を決定する。ステレオ画像対の数が増加するに従い、正しい位置において、より明瞭な最小値が得られるため、ステレオマッチングの誤対応を減らすことができ、距離の推定精度が向上する。なお、SSD はマッチングウィンドウ間の相違度を表す指標の一つである。

マルチベースライン法のアルゴリズムを C 言語によるプログラムで実装し、Illum の生画像から生成したマルチビューステレオ画像に適用した。その結果を図 2.4 に示す。基準画像 1 枚、参照画像 105 枚の合計 106 枚の画像を使用した。ステレオマッチングを行うマッチングウィンドウの大きさは  $9 \times 9$ 、視差の探索範囲は最大視差 5、最小視差 -9 に設定した。また、すべての画像が平行化されてい

ると仮定して、プログラムを作成した。

マルチベースライン法の処理は次のように行われる。

- (1) 基準画像の画素位置ごとに、マッチングウィンドウ内の画素値の標準偏差を求める。標準偏差が閾値を下回るならば、その画素位置の視差は計算できないと判定する。これは、特徴がない領域の計算を打ち切るための処理である。
- (2) 基準画像のある画素に対応する参照画像中の画素を、あらかじめ決めた視差の範囲内で探索する。探索における類似度は、マッチングウィンドウ間でのNCC(正規化相互関係数)を使う。マルチベースライン法は、視差をパラメータとする類似度を、複数のステレオ画像で足し合わせる。ただし、ステレオ対ごとの視差は、あらかじめ、基線長で正規化しておく。今回の実験では、105組のステレオ画像を使い、視差の刻みは0.1に設定した。



図 2.4 マルチベースライン法で生成した視差画像の例

マルチベースライン法で生成した視差画像の画素値は、画素を単位とする視差値である。これを実距離に換算するためには、カメラの焦点距離などの情報が必要である。したがって、このままでは、実空間における定量的な値を求めることができない。

### 3章 Illum を用いたマルチビューステレオ

3章では、マルチビューステレオ画像にステレオカメラキャリブレーションを行い、ステレオマッチングを実施する方法について述べる。

#### 3.1 生画像からマルチビューステレオ画像への変換

2.2に述べた方法で、生画像をマルチビューステレオ画像に変換した。マイクロレンズ内は、概ね、縦・横 13 画素に分割されており、マルチビューステレオ画像は約 169 組生成できる。しかし、変換した画像の中で不鮮明な画像もあるため、実際に使用した画像は基準画像 1 枚と、参照画像 81 枚の合計 82 枚である。

#### 3.2 マルチビューステレオ画像のキャリブレーション

マルチビューステレオ画像から、3 次元空間に関する定量的な距離値を計算するために、カメラキャリブレーションを行う。カメラキャリブレーションは、位置が既知である多数の三次元座標と、それをカメラで撮影した画像座標の組から、カメラの内部パラメータ（焦点距離、画像中心、レンズ歪み係数）と外部パラメータ（位置と姿勢）を求めることである。また、ステレオカメラキャリブレーションは、一組のステレオ画像（ステレオカメラで撮影した 2 枚の画像）を同時にカメラキャリブレーションすることで、2 台のカメラの内部パラメータとカメラ間の外部パラメータを求めることがある。本研究では、マルチビューステレオ画像の中の一枚の画像を基準画像として、その他の全ての画像との間でステレオカメラキャリブレーションを行った。基準画像は、マイクロレンズ中央に位置する画素を再配列した画像である。キャリブレーションには OpenCV の関数を利用した。

キャリブレーションに使ったチェスボードパターンの画像を図 3.1 に示す。これは  $10 \times 7$  の交点数、一マス 93.5mm 四方のパターンである。これをさまざまに位置と方向から 8 枚の画像を撮影し、キャリブレーションを行った。注意すべき事項として、キャリブレーションに用いる画像は、撮影の位置と方向をさまざまに変えることが重要である。例えば、正面方向から撮影した画像だけを用いるとキャリブレーションの精度が悪くなり、画像の平行化や視差画像の計算結果が不安定になる。

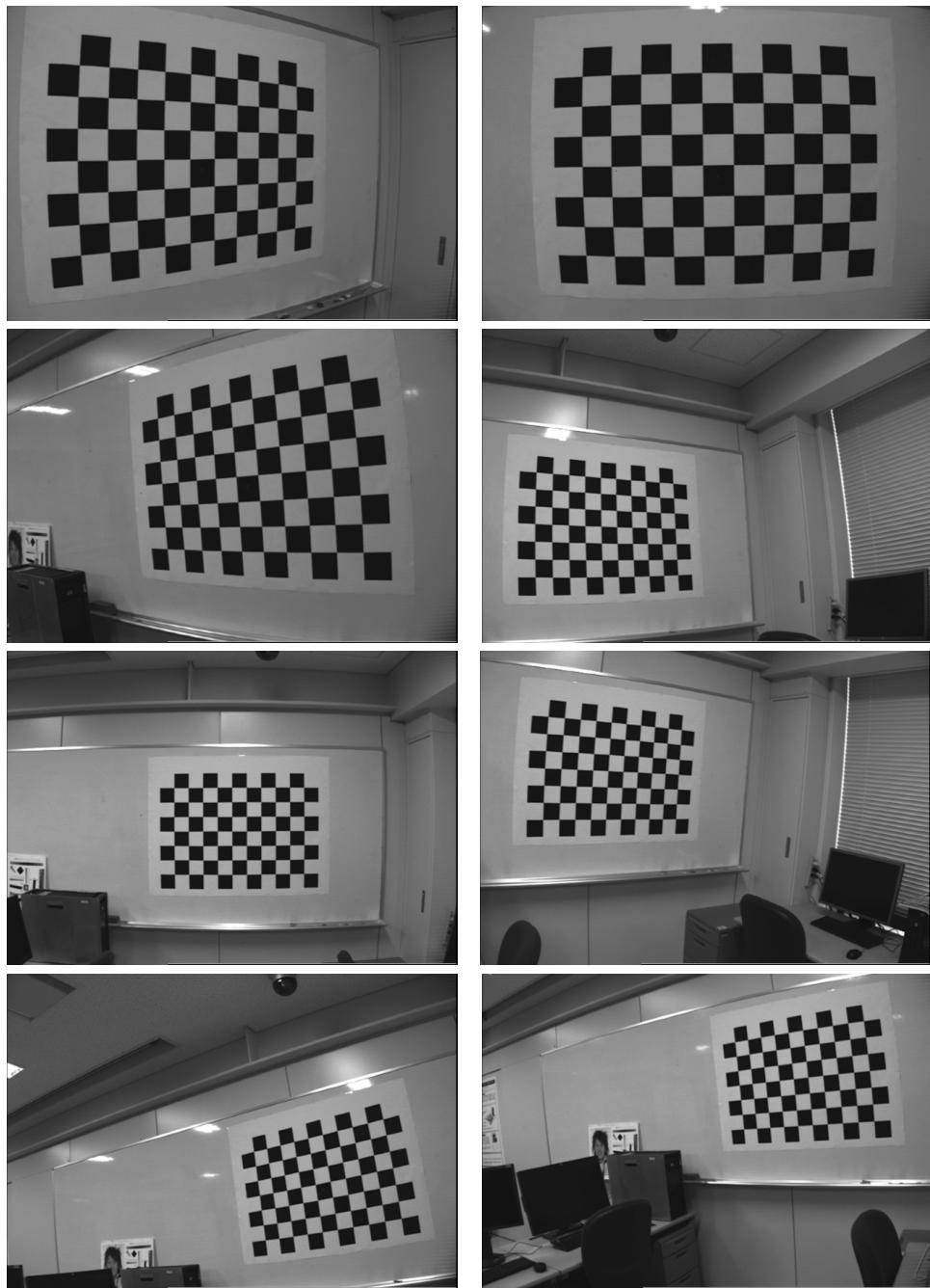


図 3.1 キャリブレーションに利用した 8 枚のチェスボード画像

キュリブレーションの結果から次のことが明らかになった.

- (1) それぞれの画像は  $1080 \times 752$  画素のカラー画像である. これに対応する画像センサは画素サイズが  $0.7 \mu\text{m}$  四方, センササイズが  $0.756\text{mm} \times 0.5264\text{mm}$  と換算できる.
- (2) それぞれの画像を撮影したカメラの焦点距離はおよそ  $0.7\text{mm}$  と換算できる.

(3) 縦横に並ぶカメラの間隔はおよそ 0.35mm と換算できる。  
 すなわち, Illum は画像センサの面積が 0.756mm×0.526mm, レンズの焦点距離が 0.7mm, 画素数 1080×752 画素の微小カメラが, 縦横に 0.35mm の間隔で 150 度並んだカメラアレイとみなすことができる。

### 3.3 ステレオ画像の平行化と逆平行化

最も単純なステレオ画像は, 同じ仕様の 2 台のカメラを水平方向に並べたステレオカメラで撮影した一組の画像である。この時, 2 つの被写体像は, 被写体までの距離に応じた視差だけ平行移動した関係になる。この条件を厳密に述べると, 2 台のカメラの画像センサ面が共通平面の上にあり, 画像センサの同じ水平ラインが同じ直線上にある, という条件になる。この関係にある一組のステレオ画像を平行化されたステレオ画像とよぶ。

平行化されたステレオ画像は, 視差画像を推定するためのステレオ画像処理が単純になる。本研究で視差画像を計算するために用いるOpenCVの関数 (stereoBM) はステレオ画像が平行化されていることを前提としている。一般的な位置関係にあるステレオ画像は平行化されていないが, それらを平行化することが可能である。OpenCVでは, 関数 (stereoRectify, initUndistortRectifyMap, remap) を使って平行化することが可能である。

ステレオ画像の平行化について説明する。撮影されたステレオ画像は, 通常, 平行化されていない。第一にレンズ歪みが存在する。第二に 2 つの画像センサの位置関係を厳密に平行化（画像センサ面を共面にし, スキャンラインを合わせる）することは不可能である。第三にカメラの位置関係が縦や斜めの場合がある。縦方向にずれた画像間では, 縦方向に視差があり, 斜め方向にずれた画像間では, 斜め方向に視差がある。そのため, これらの場合は, 画像を回転させて, 横方向に視差があるように変換する必要がある。

平行化の式は(3-1)で表される。なお, 平行化を行うための関数として OpenCV の remap を用いた。

$$dst(x, y) = src(mapX(x, y), mapY(x, y)) \quad \dots \quad (3-1)$$

$dst(x, y)$  は出力画像の座標  $(x, y)$  における画素値,  $src(x, y)$  は入力画像の座標  $(x, y)$  における画素値である。 $mapX, mapY$  は平行化を行う画像変換の関数 stereoRectify, initUndistortRectifyMap で得られる画像変換マップである。変換によっては,  $mapX, mapY$  の値が画像の外側にはみ出してしまうことがある。こ

の場合は何らかの外挿手法が必要になる。本研究では、あらかじめ設定した値 0 になるようにした。OpenCV の関数 `stereoRectify` を用いて平行化すると、水平方向に視差があるステレオ画像か、垂直方向に視差があるステレオ画像のいずれかとして平行化される。垂直ステレオとして平行化されると、視差が縦方向でるように変換されるため、この場合さらに、90 度回転させる必要がある。

マルチビューステレオ画像を平行化すると、一組のステレオ画像ごとに平行化が実行される。次いで OpenCV の関数 `stereoBM` で視差画像を計算する。このように計算する視差画像集合を統合したいのであるが、平行化された状態の視差画像を、そのまま統合することはできない。なぜならば、視差画像は基準画像に対する視差を表す画像であるが、ステレオ画像対ごとの平行化によって、そのつど、基準画像は異なった形に変形されるからである。そのため、平行化された視差画像を、元の基準画像の状態に戻すことを考える。この処理を逆平行化とよぶことにする。

逆平行化のアルゴリズムは、式(3-1)を利用したものである。

まず、平行化された画像における任意の座標  $(x, y)$  の画素値を  $V$  とすると、式(3-2)となる。

$$V = \text{dst}(x, y) \quad \dots \quad (3-2)$$

次に  $(x, y)$  における `mapX`, `mapY` の値を  $XX$ ,  $YY$  とすると、式(3-3), (3-4)を得る。

$$XX = \text{mapX}(x, y) \quad \dots \quad (3-3)$$

$$YY = \text{mapY}(x, y) \quad \dots \quad (3-4)$$

このとき、 $XX$  および  $YY$  の値がともに画像の画素数の範囲内であれば、逆平行化画像の画素値を  $\text{dst}'$  とすると、式(3-5)になる。

$$\text{dst}'(XX, YY) = V \quad \dots \quad (3-5)$$

これを画素数だけ繰り返すことで逆平行化を行う。

### 3.4 複数の視差画像を統合するための手法

平行化されたステレオ画像に対して、OpenCV の関数 `StereoBM` を使ってステレオマッチング処理を行う。これによって一つ一つの視差画像が推定される。その視差の値は、ステレオ画像ごとの画素単位での値である。それぞれの視差画像から、対応する座標の視差値を取り出し、基線長で正規化する。そこから、0 を除く視差値の平均をとる方法を試みた。ここで、視差値 0 は、視差が計算できなかった場合に `StereoBM` が返す値である。`StereoBM` はステレオマッチングの

成否を判定する閾値を設定することができる。マッチングの類似度がこの閾値よりも小さい場合、視差が計算できないと判定する。

## 4章 実験結果と考察

### 4.1 ステレオ画像の平行化と逆平行化の検証

3.3 で述べたステレオ画像の平行化を実行し、視差画像を求めた。作成した視差画像は、横方向 10 枚、縦方向 8 枚、斜め方向 64 枚の合計 82 枚である。図 4.1 ～図 4.4 に基準画像と、これに対して縦・横・斜めに視差がある画像を示す。



図 4.1 使用した基準画像



図 4.2 横方向に視差がある参照画像の例



図 4.3 縦方向に視差がある参照画像の例



図 4.4 斜め方向に視差がある参照画像の例

図 4.5 から図 4.10 に, 平行化した結果の画像を示す. いずれも, レンズ歪みの補正がかかった変換と, それぞれの方向にあわせて, 平行化されていることがわかる. また, 縦方向と斜め方向の平行化による変換(図 4.7 から図 4.10)は, 画像サイズの都合上, 変換した際にはみ出す範囲が出てしまう. 逆平行化の変換を行うとき, この範囲の画素値は 0 としてとり扱うことにする.



図 4.5 基準画像を平行化した例（対の画像の視差が横方向の場合）



図 4.6 横方向に視差がある参照画像を平行化した例



図 4.7 基準画像を平行化した例（対の画像の視差が縦方向の場合）



図 4.8 縦方向に視差がある参照画像を平行化した例



図 4.9 基準画像を平行化した例（対の画像の視差が斜め方向の場合）



図 4.10 斜め方向に視差がある参照画像を平行化した例

図 4.11 から図 4.13 は、平行化した方向ごとに求めた視差画像である。本実験では、視差値が大きくなるほど白く表示し、視差値が小さくなるほど黒く表示している。ただし、StereoBM はステレオマッチングの成否を判定する閾値を設定する

ことができる。そのため、マッチングの類似度が、この閾値よりも小さい場合、視差が計算できないと判定する。その場合の視差値は0である。

図4.11から図4.13のカラーチェッカーパターンがある場所に赤い円の印をつけた。その部分に注目すると、横方向の視差画像(図4.11)は、縦に視差が不安定な領域が続いている。縦方向の視差画像(図4.12)は、横に視差が不安定な領域が続いている。斜め方向の視差画像(図4.13)は、エッジが縦、横とともに有効に機能するため、視差が計算されている範囲が広く出ている。StereoBMは、ブロックマッチングを利用している。チェッカーパターン領域の中で、各色の小領域内は特徴がないので、マッチングが不安定になる。横方向のステレオ画像(図4.6)の場合には、上下の色を区切る横長の黒い領域においてもマッチングが不安定になる。このようにマッチングが不安定になる領域が縦につながるので、図4.11の赤丸で示したような状態になる。図4.12は方向が90度異なるが、同様の現象である。図4.12は各色の小領域でのみマッチングが不安定になっていることを示している。



図4.11 図4.5と図4.6から求めた視差画像

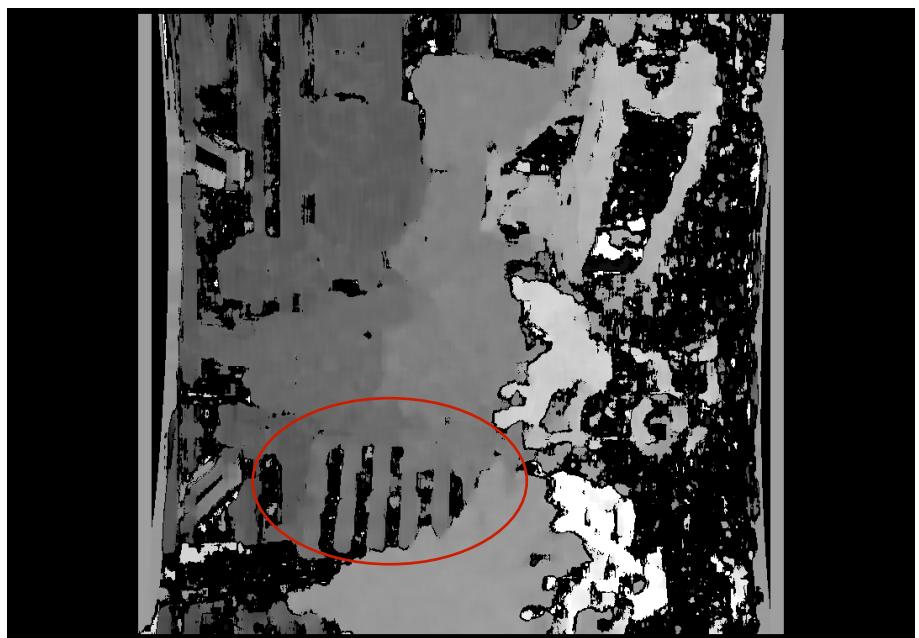


図 4.12 図 4.7 と図 4.8 から求めた視差画像

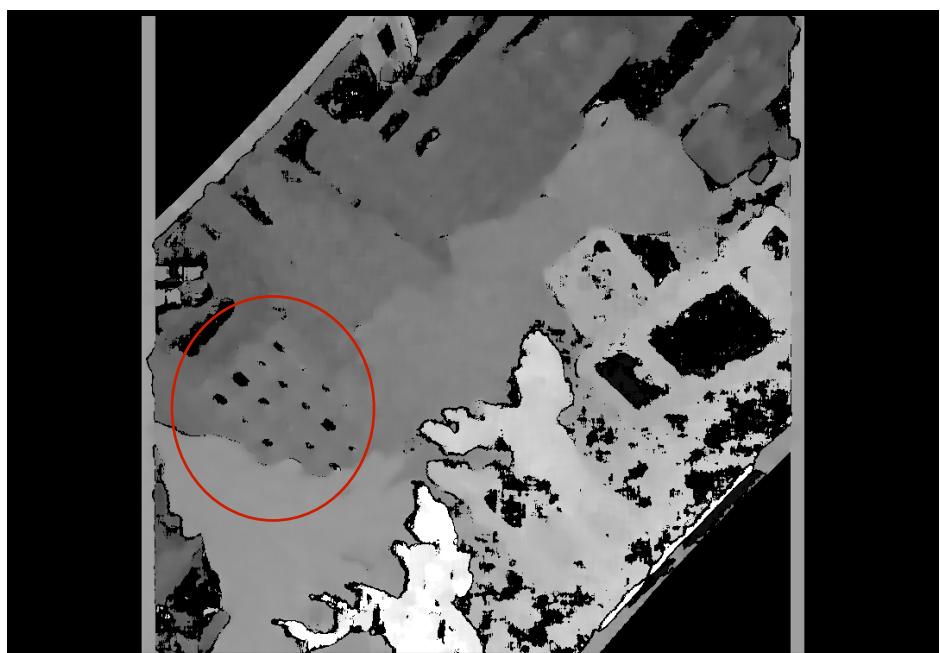


図 4.13 図 4.9 と図 4.10 から求めた視差画像

図 4.14 から図 4.16 はそれぞれ、図 4.11 から図 4.13 に対して、逆平行化の処理を施した画像である。



図 4.14 図 4.11 を逆平行化した視差画像



図 4.15 図 4.12 を逆平行化した視差画像

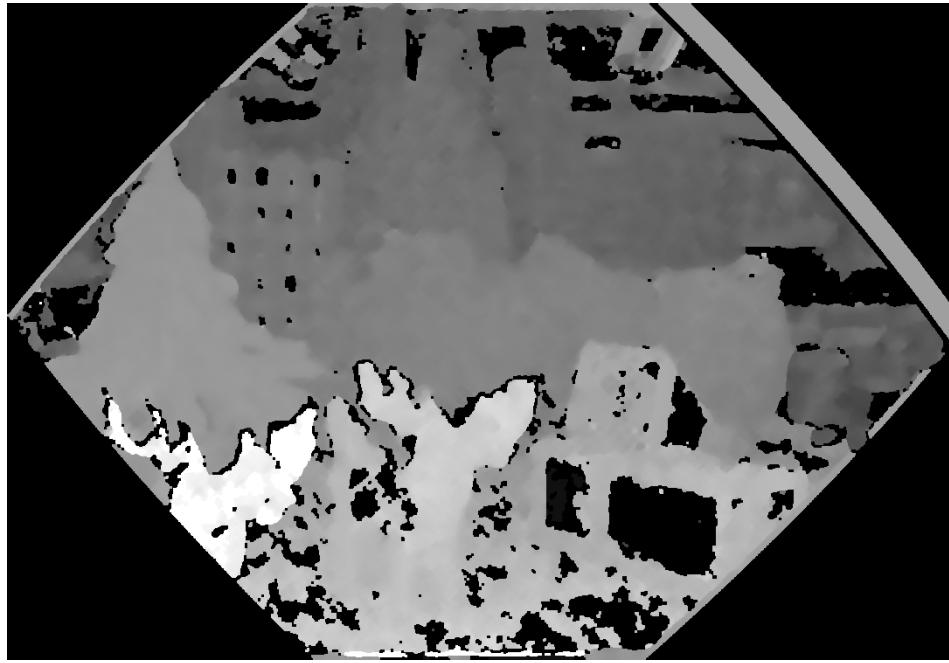


図 4.16 図 4.13 を逆平行化した視差画像

## 4.2 複数の視差画像の統合

基準画像を固定し、これに対する多数の参照画像ごとに平行化・ステレオマッチング・逆平行化の処理を行った。その結果、同じ基準画像に対して多数の視差画像を得た。そして、これらの視差画像の画素値をステレオ対ごとの基線長で正規化（基線長で割る）した。

このように計算した複数の視差画像を統合する手法には、さまざまな方法が考えられる。今回は、最も単純な方法ではあるが、0以外の視差値を平均化したものを統合した視差画像とした。図 4.17 にその結果を示す。図 4.17 を観察すると、次のことがわかる。

- (1) 統合した結果、円形の領域内に有効な視差値が現われた。これは、平行化・逆平行化の処理によって、画像の周辺部が計算不可能な領域と判定されたことが影響している。
- (2) 画像中の上の領域（遠方の領域）は概ね正しく計算できているように見える。下の領域（手前の領域）は計算が不安的になっている領域が多い。これは図 4.15 の視差画像に見られるように、個々の視差画像レベルで、手前の領域の視差が不安的であったことが影響していると考えられる。
- (3) 視差値を平均したものを見ると、視差の計算結果が不

安定な領域において、誤差が大きくなっていると考えられる。

これらの結果はそれぞれに課題を含んでいる。次のような対策が考えられる。統合した視差画像の有効領域が円形になることは、OpenCV の関数を利用して平行化したことが原因である。これを解決するには、平行化の関数を自作する必要がある。手前の領域の視差が不安定になる一つの理由は、ステレオマッチングの手がかりになる画像特徴が少ないとされる。しかし、図 4.15 を見ると、トナカイの胴体や足の部分でも視差が不安定になっている。その理由として、キャリブレーションの精度が十分でないことが考えられる。平均値によって視差を統合したが、他の方法として画素ごとの視差値の中で最頻値を統合した視差値とする方法などが考えられる。



図 4.17 統合した視差画像

### 4.3 考察

Illum から生成したマルチビューステレオ画像をキャリブレーションし、その結果を用いて、マルチビューステレオによる 3 次元計測を行った。OpenCV の関数を利用して、マルチビューステレオ画像に対するキャリブレーション、平行化、ステレオマッチング、逆平行化、視差画像の統合という一連の処理を完成させることができた。これによって、定量的な測定値を得ることができる 3 次元計測の基礎技術を確認することができた。

課題として次のことがある。

- (1) マルチベースライン法による統合結果と比較して、視差画像のレベルでは明確に良くなつたとは言えない。キャリブレーションのノウハウを蓄積してステレオマッチングの精度を高めるであるとか、マッチングウィンドウサイズなどのパラメータを最適化する、など実験手順を高度化して、今回のシステムとしての最高性能を発揮させることが望ましい。
- (2) 視差画像の計算にとどまっており、3次元直交座標における mm 単位での計測値を得るまでには到っていない。また、測定精度の評価を実施できなかつたことも課題である。
- (3) マルチビューステレオにおける距離情報の統合手法としてマルチベースライン法と提案手法を実装して、比較した。その他の手法についても、さまざまな研究があると思われるが、調査・検討することができなかつた。

## 5 章 結論

Illum で撮影した生画像からマルチビューステレオ画像を生成し、これにステレオビジョンのアルゴリズムを適用した。まず、生成したマルチビューステレオ画像が標準的なステレオ画像であると仮定し、マルチベースライン法を適用して 1 枚の視差画像を計算した。この視差画像は被写体までの距離を表現するものではあるが、このままでは、定量的な 3 次元計測結果に換算することができない。

そこで、生成したマルチビューステレオ画像にステレオカメラキャリブレーションを施したのち、ステレオマッチングを行った。OpenCV の関数を利用してこれを実装する場合、ステレオ画像ごとに平行化・ステレオマッチング・逆平行化を行ったのち、視差画像を統合する必要があった。視差画像の統合手法としては、有効な視差値を平均化する方法を採用した。この一連の処理を完成させることができ、これによって、定量的な測定値を得ることができる 3 次元計測の基礎技術を確認した。

課題として次のことが残った。

- (1) マルチベースライン法による視差画像と比較して、明確に良くなったとは言えない。
- (2) 視差画像の計算にとどまっており、測定精度の評価ができていない。
- (3) 距離情報の統合手段として、有効な視差値を平均する方法を採用したが、初步的な方法であり、改善の余地がある。

## 参考文献

- [1] 蚊野浩, 中島類, 「ぼけ具合の調整や3D画像の作成も自由自在 最新テクノロジーの研究・・・あとからピント合わせ」, Interface CQ出版, 2015年6月号, pp.156-165, 2015年.
- [2] 服部寛, 「車載向けステレオ画像処理技術」, 東芝レビュー, Vol.63, No.5, pp.48-51, 2008.
- [3] 畠周平, 早川賢司, 山口由美子, 真屋学, 「マルチビューステレオ手法による三次元モデル作成サービス」, For the Future 2014, アジア航測, 2014年.

## 謝辞

本論文を作成にあたり, 丁寧な御指導を賜りました蚊野浩教授に感謝いたします.

## 付録 本研究で開発したプログラムの説明

### プログラム 1

checker\_patern.cpp

チェスボードパターンの画像を読み込み、チェスボードの交点座標を推定して、記録するプログラム。

rectify.cpp

2枚のステレオ画像からカメラパラメータを求め、平行化、視差画像、逆平行化を行うプログラム。

main.cpp

上記のプログラムから得た結果から、視差画像を提案手法によって、統合するプログラム。

### プログラム 2

main.cpp

標準的なマルチビューステレオ画像から、マルチベースライン法で視差画像を計算するプログラム。