

要約

本研究の目的は、ライトフィールドカメラ Lytro を用いた三次元計測を実現し、その距離測定精度を検証することである。Lytro はアプリケーション上で使用される最終フォーマットを生成する途中にステレオ画像を生成できる。これを利用すれば Lytro の内部に仮想的に二台のカメラがあると考えることができる。つまり、ステレオ計測ができるのではないかと考えたのである。この考えのもと、二つの実験を行った。第一の実験として Lytro を用いて実際に距離を測定する実験を行うとともに、第二の実験として、ステレオカメラキャリブレーションを行った。キャリブレーション結果からステレオカメラの内部パラメータ、外部パラメータを求め、理論的に距離測定精度を検証した。第一の実験結果から、Lytro の前面から約 20cm の地点でステレオ画像の視差が 0 になることと、無限遠での視差が約-6 画素に漸近することがわかった。また、最小二乗法を用いて 40 組のデータから近似式を求め距離推定に適用したが 60cm 以降では今回求めた近似式を用いての距離推定できなかった。第二の実験結果である Lytro のカメラキャリブレーションで得られたカメラパラメータからシミュレートされる結果は、34.5cm の地点で視差が 0 になり、無限遠での視差は約-6.7 画素に漸近するという結果が得られた。このように第一の実験と第二の実験の結果は、完全には一致しなかった。また、今回のキャリブレーションではステレオカメラの位置関係を示す回転行列の精度が不十分であることがわかった。

目次

第1章 序論	・・・ 3
1.1 本研究の背景と目的	・・・ 3
1.2 論文の構成	・・・ 4
第2章 ライトフィールドカメラの原理	・・・ 5
2.1 ライトフィールドと光線状態の計算理論	・・・ 5
2.2 ライトフィールドカメラ Lytro	・・・ 8
2.2.1 Lytro の概要	・・・ 8
2.2.2 Lytro の動作原理	・・・ 10
2.2.3 Lytro の生データとそのステレオ画像への分解	・・・ 11
第3章 ステレオ画像計測と Lytro を用いた実距離測定	・・・ 13
3.1 平行ステレオによる三次元計測の原理	・・・ 13
3.1.1 実距離測定	・・・ 16
3.1.2 提案手法の基本的な考え方	・・・ 17
3.3 Lytro を用いた実距離の測定	・・・ 17
3.2.1 実験用画像の取得	・・・ 17
3.2.2 Lytro の生画像からステレオ画像への再構成	・・・ 19
3.2.3 視差の計測	・・・ 19
3.3 実験結果	・・・ 20
3.4 最小二乗法による近似式	・・・ 26
第4章 Lytro のキャリブレーションと実距離測定の検証	・・・ 29
4.1 カメラキャリブレーション	・・・ 29
4.1.1 同次座標を用いた透視投影モデル	・・・ 29
4.1.2 カメラキャリブレーションの手順	・・・ 34
4.1.2 カメラキャリブレーションの実施	・・・ 35
4.2 実験結果	・・・ 37
第5章 結論	・・・ 42

謝辞

参考文献

付録

1. 序論

1.1 本研究の背景と目的

米国 Lytro 社が開発したライトフィールドカメラ[1]は撮影時にピントを合わせる必要がなく、撮影後のソフトウェア的な後処理によって任意の位置にピントを合わせることができるカメラである。ライトフィールドカメラの特徴は、従来のカメラと異なり、三次元空間の光線情報をライトフィールドとして取得することである。Lytro はリフォーカス画像を生成するために、ライトフィールドを用いている。

ライトフィールドは、三次元空間を満たす光線集合のことであるが、これをリフォーカス画像を生成すること以外に利用することも可能である。実際、Lytro の原型となった、1992 年に MIT の Adelson が研究したプレノプティック・カメラ[2]は、Lytro とほとんど同じ構成のカメラであり、ステレオ画像計測の理論にもとづいて、距離画像を計測するために利用された。

初期のライトフィールドカメラとして、スタンフォード大学の Marc Levoy らの研究があげることができる[3]。これは図 1.1 のように、128 台のカメラを横 16×縦 8 に構成したものである。また、株式会社ビュープラスから販売されてる、25 眼カメラアレイシステム ProFUSION25 があげられる。これは 25 台のカメラが 12 mm 間隔で並べられている。これら二つのカメラアレイ型ライトフィールドカメラは機能検証には有効であるが実用的ではない大きさである。Lytro 社のライトフィールドカメラはマイクロレンズ方式であり実用的なサイズに納まっている。

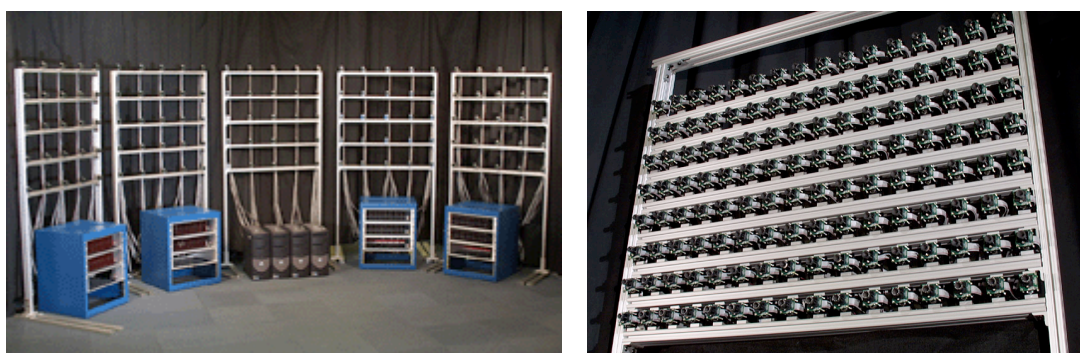


図 1.1 スタンフォード大学のマルチカメラアレイ



図 1.2 ビュープラス社の ProFUSION25

Lytro は上述したとおり、撮影後のソフトウェア的な処理で焦点位置(ピント)を変更した画像を生成することができる。これは今までのカメラ撮影方法では不可能なことであり、どのようにして画像を取得しているのか、また、実際どのような画像処理を行い最終画像を生成しているのかという疑問は誰しものが持つであろう。そして他に何か新たな機能を付加できないのかと考えた結果、Lytro を用いて実距離を測定しようと考えた。

1.2 論文の構成

本論文の第 2 章では Lytro の原理と、それを用いた実距離測定について説明する。第 3 章では実験方法の原理を数式を用いて説明する。第 4 章では実距離測定の結果を示すとともに、Lytro に対して行ったカメラキャリブレーションから理論的に予測される計測値を導き、それを実測値と比較する。第 5 章で本研究の意義と課題をまとめる。

2. ライトフィールドカメラ Lytro の原理

2.1 ライトフィールドと光線状態の計算理論

Lytro はライトフィールドを取得し、それに対する一種の画像処理によって写真画像を生成する装置である。本節では、Lytro の原理を説明する前に、その基礎になっているライトフィールドと光線状態の計算理論について、簡単に説明する。

ライトフィールドの訳語は光線空間である。その意味は、光で照射され、人間が見ることができる三次元空間を、無数の光線があらゆる方向に飛び交っている空間をとらえたものである。つまり、光線で埋め尽くされた空間＝ライトフィールドといえる。

写真は被写体から発せられた光線がレンズで屈折し撮像素子に到達することによって生成される。三次元空間上に、光源の位置（被写体表面からの反射光も、被写体が光を発していると考えられる）、光線方向、レンズの位置、撮像素子の位置を定義すれば、光線の経路を幾何光学的に計算することができる。以下では、光線行列を用いてレンズを含む光線経路を計算する手法を示す。

図2.1のように光軸方向を z 軸、光軸に垂直な方向に x 軸、 y 軸をとる三次元座標を設定する。光線は、三次元空間において進行方向を持つ一本の線によって表現される。光線状態を光線上の1点（光点）の座標とその位置における方向ベクトルによって定義する。光点の初期位置を p_0 、その方向ベクトルを、 z 軸方向の大きさを1に正規化したベクトル q_0 とする。この光点が、 z 軸方向に、距離 a 進むことによって位置 p_1 、に移動したとする。その時の方向ベクトルを q_1 とする。すると、光点は傾き q_0 で z 軸方向に d だけ進むので $p_1 = p_0 + q_0 d$ となる。光線方向は変わらないので $q_1 = q_0$ となる。よって、その時の変換は行列（光線行列と呼ばれる）を用いて、以下のように表される。

$$\begin{pmatrix} p_1 \\ q_1 \end{pmatrix} = \begin{pmatrix} 1 & d \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} \quad (2.1)$$

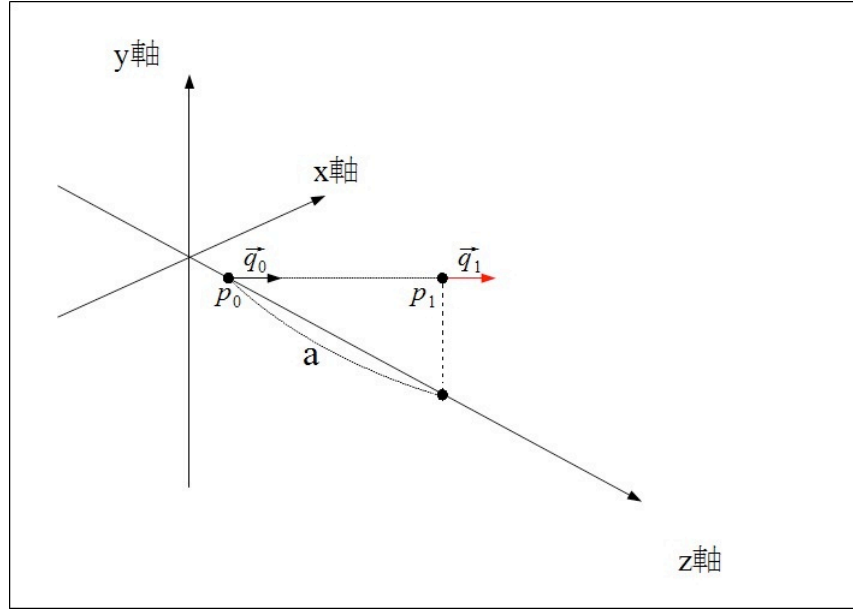


図 2.1 光線の直進する様子

次にレンズにおける光線の屈折を考える。レンズを、厚みのない薄肉レンズでモデル化する。図 2.2 より光点は $\mathbf{p}_0$ から $\mathbf{p}_1$ 、 $\mathbf{p}_2$ 、 $\mathbf{p}_3$ に進む。 $\mathbf{q}_1$ はレンズに入射する前の光線の方向ベクトル（進行方向（ Z 軸）の要素を 1 に正規化してある）、 $\mathbf{q}_2$ は屈折後の方向ベクトル。また、 $\mathbf{p}_0$ からレンズまでの距離が a 、 $\mathbf{p}_3$ からレンズまでの距離が b で、 $\mathbf{p}_0$ は $\mathbf{p}_3$ で結像する。このとき、以下の式が成り立つ。

$$\begin{aligned}\mathbf{p}_1 &= \mathbf{p}_0 + a\mathbf{q}_1 \\ \mathbf{p}_2 &= \mathbf{p}_1 \\ \mathbf{p}_3 &= \mathbf{p}_2 + b\mathbf{q}_2\end{aligned}\quad (2.2)$$

レンズの中心を座標の原点にとると、 $\mathbf{p}_0$ と $\mathbf{p}_3$ は原点に対して $a:b$ に位置するので、次の関係が成り立つ。

$$b\mathbf{p}_0 + a\mathbf{p}_3 = \mathbf{0} \quad (2.3)$$

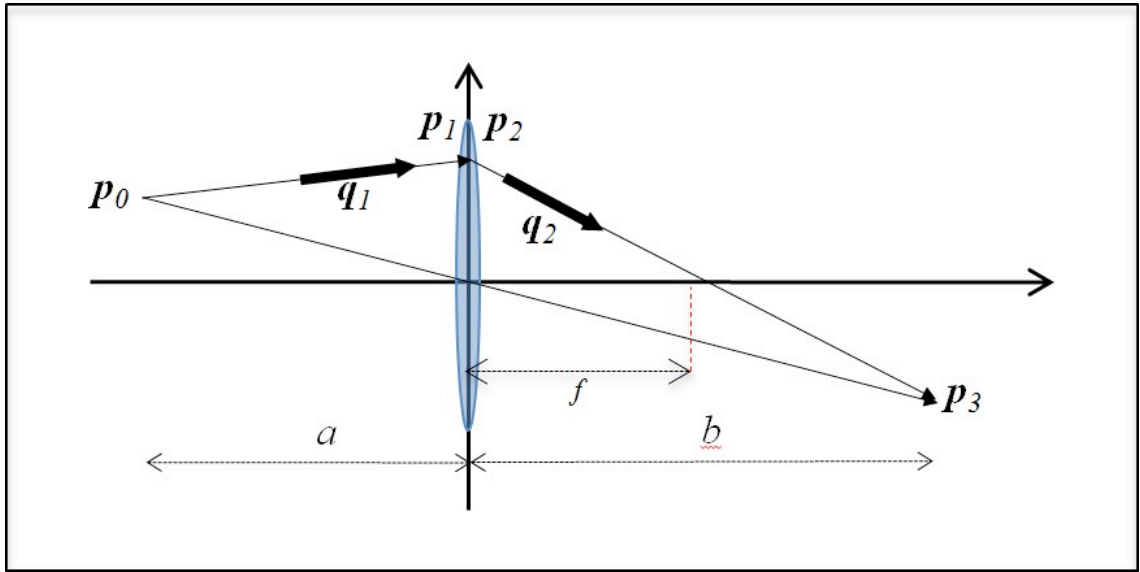


図 2.2 光線の屈折する様子

またレンズの公式

$$\boxed{\times} \quad (2.4)$$

が成り立つ。これらを整理すると

$$\begin{aligned} p_2 &= p_1 \\ q_2 &= -\frac{1}{f} p_1 + q_1 \end{aligned} \quad (2.5)$$

となるので、レンズで屈折するときの変換行列は以下のように表せる。

$$\begin{pmatrix} p_2 \\ q_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ q_1 \end{pmatrix} \quad (2.6)$$

これらの式を用いてカメラの主レンズに入射する光線が撮像素子に受光されるまでの軌跡をシミュレーションすることができる。ライトフィールドカメラで行われる写真画像を計算する処理は、原理的には、光線の軌跡を計算するこの理論に基づいて行われる。

2.2 ライトフィールドカメラ Lytro

本研究で用いるライトフィールドカメラ Lytro の概要、動作原理、実験に利用する Lytro の生データの構成と、生データをステレオ画像に再構成する方法について説明する。

2.2.1. Lytro の概要

Lytro は撮像素子の前面にマイクロレンズをハニカム構造状に配列し、光線方向を分離して記録する。図 2.4 に、その撮像部の構成を示す。Lytro のマイクロレンズサイズは $14\mu\text{m}$ 、撮像素子の画素ピッチは $1.4\mu\text{m}$ であり、マイクロレンズの直径は画素の 10 倍となる。また、イメージセンサの保護ガラスはマイクロレンズアレイの機能を兼ねておりガラスの上面は平坦であり、底面がマイクロレンズアレイとなっている。

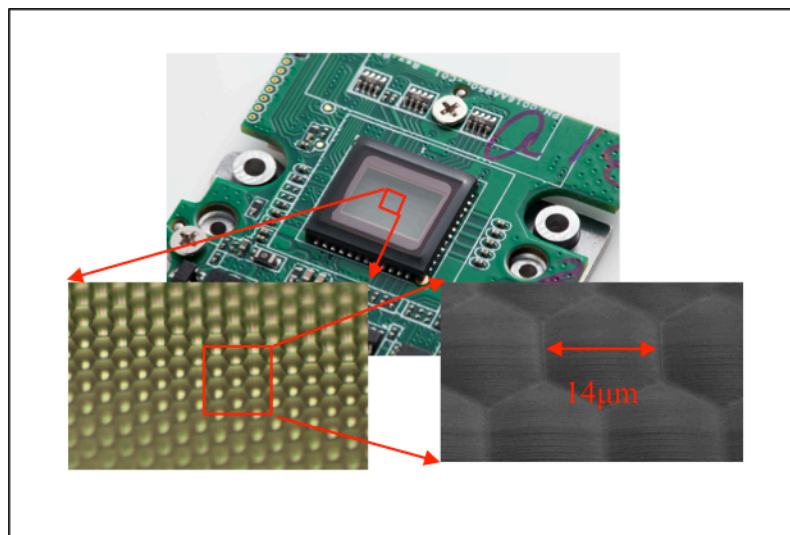


図 2.3 画像センサ部の拡大写真

(写真: 日経 BP 社、豊通エレクトロニクス ヴァン・パートナーズ)

Lytro は撮影後にアプリケーションを用い、撮像画像のピントを合わせたい位置にマウスポインタを合わせクリックすることで、その位置にリフォーカスすることができるカメラである。このアプリケーションはLytroの発売時点でMac版のみが利用可能であったが、現在は64bit版のみではあるがWindows7で利用可能である。左の画像では、奥の本に焦点が合っているが、カメラ手前側の物体をクリックすると、その物体に焦点が合うように画像が再構成される。



図 2.4 Lytro アプリケーションによる生成画像の例

2.2.2. Lytro の動作原理

Lytro は光線集合であるライトフィールドを取得し独自のレンダリングアルゴリズムを使用して、撮影後に写真画像の焦点位置を自由に変更できる処理を実現している。図 2.6 を用いてその原理を説明する。図 2.6 において A にある被写体は主レンズで屈折しマイクロレンズ位置で焦点を結ぶ。A の各点から発した光線はマイクロレンズを通過し、各撮像素子に記録される。マイクロレンズに対応する画素を重ねると A に焦点があった画像が生成される。また、B にある灰色の丸の位置を通過する光線はそれぞれ異なったマイクロレンズを通過し撮像素子に記録される。それらの画素値を平均化することで B に焦点があった画像を生成でき、A がボケて見えるといった現象を再現できる。

図 2.5 の主レンズ上の部分開口 1 を通過した光線は各マイクロレンズが覆っている撮像素子の下の画素に記録される。その位置の画素だけで画像を生成することができる。同様に部分開口 2 を通過した光線は各マイク

ロレンズが覆っている撮像素子の上の画素に記録される。このようにマイクロレンズが N 個の画素を覆うとき、マイクロレンズに対して同じ位置にある画素を集めてできる N 個の部分開口に分解して取得される N 個のステレオ画像を生成する。

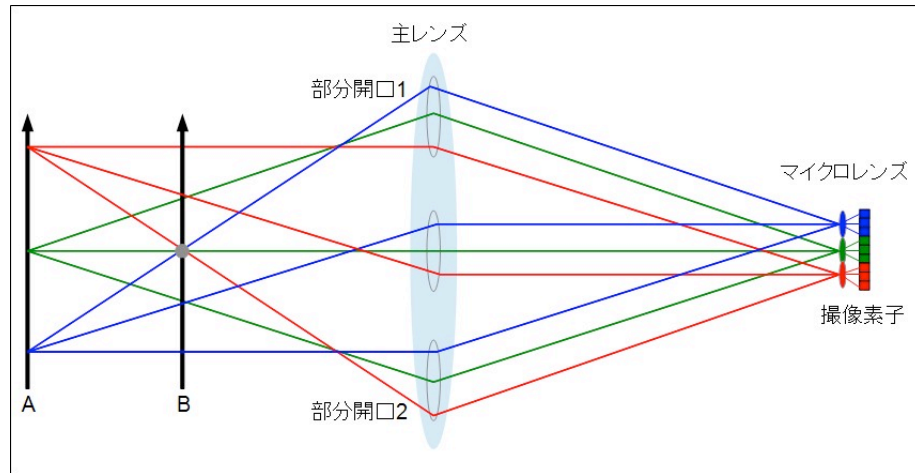


図 2.5 マイクロレンズ方式ライトフィールドカメラの概要

2.2.3. Lytro の生データとそのステレオ画像への分解

Lytro が取得する生データを画像化すると、図 2.7 のようになる。図 2.7 の左は 3280×3280 画素の撮像素子で取得される生画像である。その一部を拡大すると、直径 10 画素の円構造が観察される。このことから、 330×380 個のマイクロレンズがハニカム構造で画素を覆っていることがわかる。1 つのマイクロレンズが覆う画素は 78 個程度である。しかし、正しく光線を認識できているのは多く見積もっても 50 個程度である。

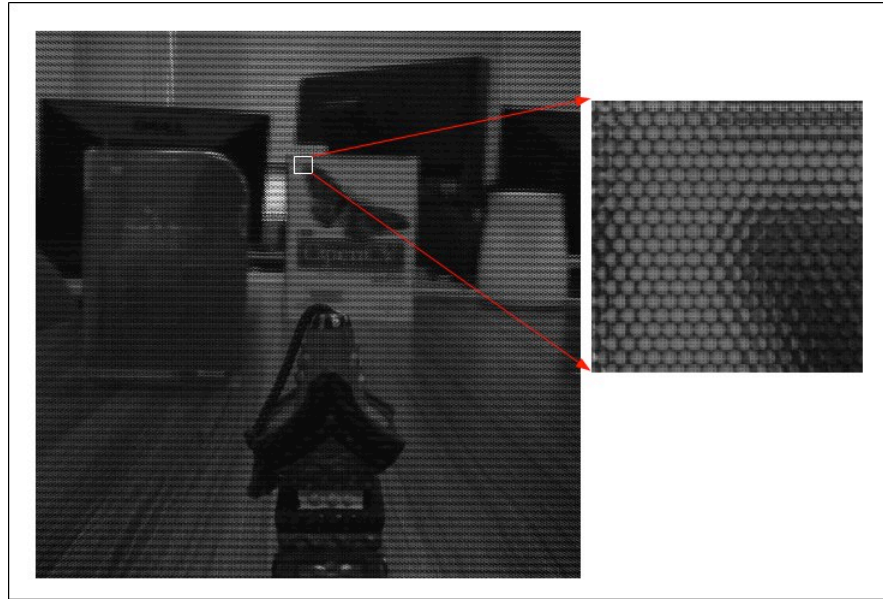


図 2.6 Lytro 社の LFC の生画像

異なったマイクロレンズの下にある同じ位置の画素に注目する。図 2.8 に示すように、左上のマイクロレンズからラスタ走査順に同じ位置の画素を取り出し、それらを 330×380 に並べると、1 つの小画像を再構成することができる。一つのマイクロレンズは概ね 50 個の画素をカバーするので、 330×380 の小画像を 50 枚取得することができる。このようにして再構成した画像群は、2.2.2 で述べたようにステレオ画像を構成している。図 2.9 にマイクロレンズの中央とそれに対して左右上下位置、右下位置に対応したステレオ画像を示す。

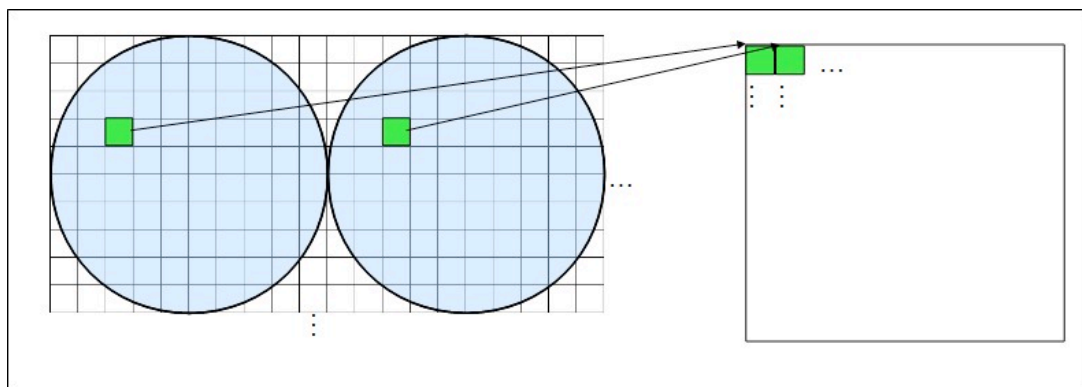


図 2.7 ステレオ画像生成の概要

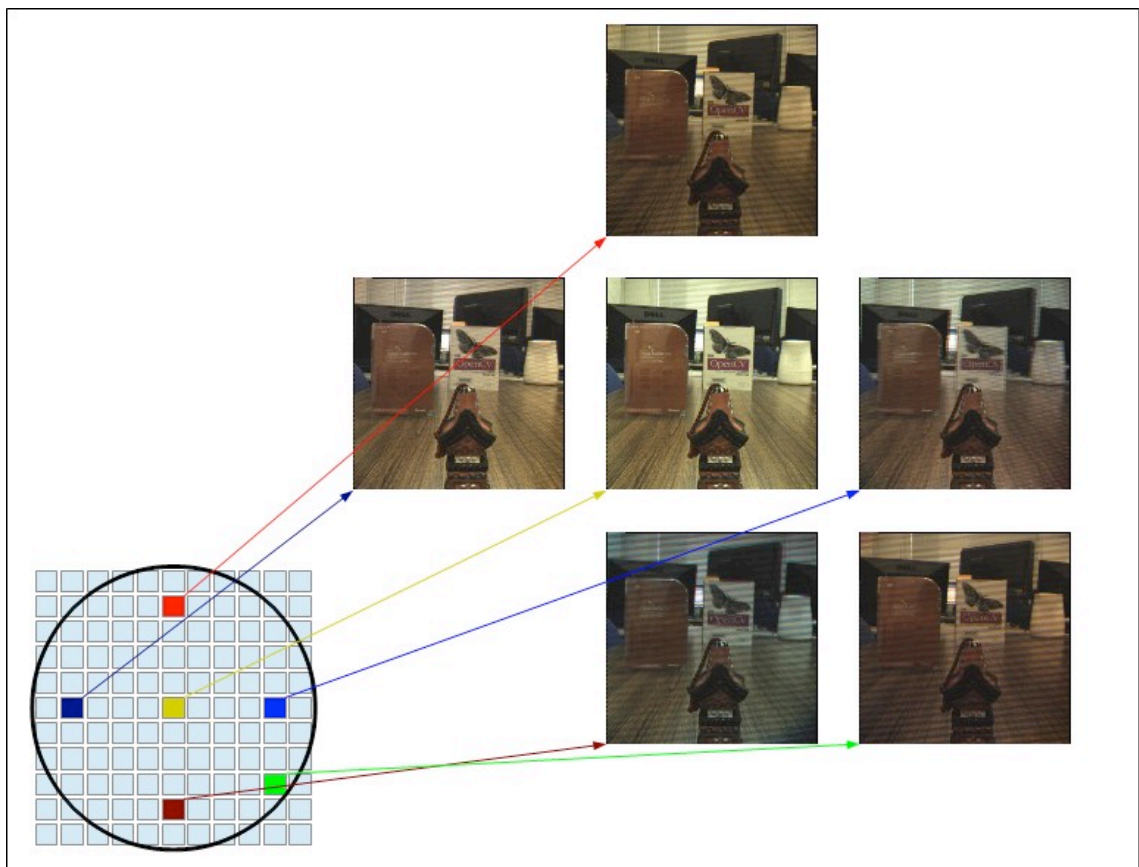


図 2.9 マイクロレンズに対する画素位置ごとのステレオ画像

マイクロレンズの中央位置の画素から生成されるステレオ画像は鮮明な画像であるが、周辺位置の画素から生成されるステレオ画像は劣化したものになってしまう。

3. ステレオ画像計測と Lytro を用いた実距離測定

本章では、ステレオ画像計測の原理について述べた後、Lytro の生データを用いたステレオ計測の実験結果を示す。

3.1 平行ステレオによる三次元計測の原理

ステレオ画像計測は、異なった位置で撮影した 2 枚以上の画像を用いて、三角測量の原理によって被写体までの距離を測定する技術である。最も簡単なステレオ画像計測システムの構成は、特性が同じ 2 台のカメラを水平方向に並べて撮影する場合である。これを平行ステレオあるいは標準ステレオとよぶ。平行ステレオにおける 2 台のカメラの投影中心間の長さを基線長とよぶ。平行ステレオでは、図 3.1 に示すように、三次元空間の測定点 (X,Y,Z) を左カメラで観察した位置 (x_l, y_l) と右カメラで観察した位置 (x_r, y_r) において、 $y_l = y_r$ となる。したがって、被写体までの距離によって生じる視差は x_l と x_r の差である。観察した 2 枚の画像から視差を計算し、それに基線長と焦点距離を用いることで被写体までの距離を計ることができる。

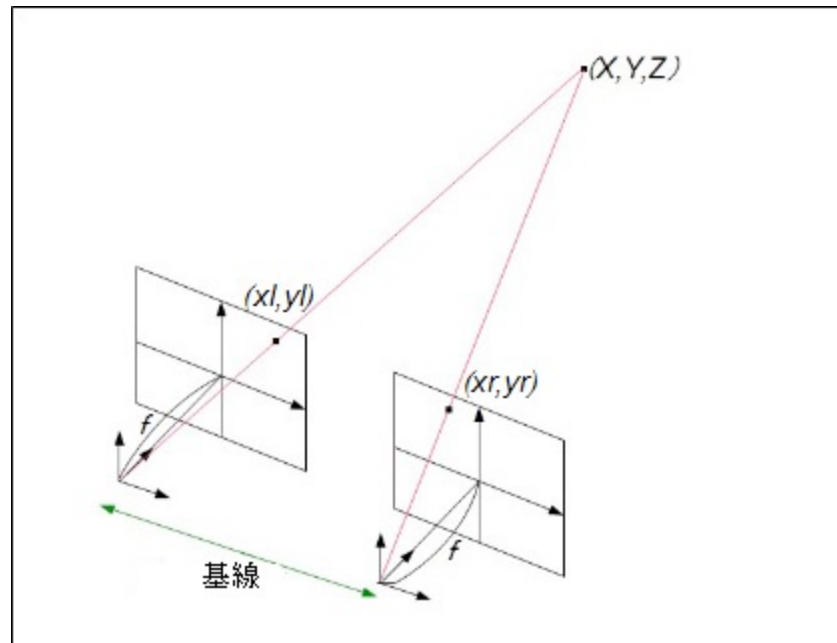


図 3.1 平行ステレオ

図 3.1 より、左側の画像座標を (x_l, y_l) 、右側の画像座標を (x_r, y_r) 、2 台のカメラの間隔(基線)を b 、カメラの焦点距離を f とした時、三次元空間の座標 (X, Y, Z) は二つの画像上の座標 (x_l, y_l) 、 (x_r, y_r) を用いて次のように表され

る。

$$\begin{aligned} X &= \frac{bx_l}{x_l - x_r} \\ Y &= \frac{by_l}{x_l - x_r} \\ Z &= \frac{bf}{x_l - x_r} \end{aligned} \quad (3.1)$$

この式群の Z についての式を変形すればカメラの焦点距離を求める式に変形することができる。先に結果の式を記述する。

$$f = \frac{Z(x_l - x_r)}{b} \quad (3.2)$$

式(3.2)の導出方法について説明する。まずステレオビジョンにおける2台のカメラの透視投影行列は以下のように表せる。

$$\begin{bmatrix} \times \\ \times \\ \times \end{bmatrix} \quad (3.3)$$

$\mathbf{P}$ は基準となるカメラの透視投影行列（3×4 行列）、 $\mathbf{A}$ はその内部パラメータ行列（3×3 行列）である。 $\mathbf{P}'$ は他方のカメラの透視投影行列、 $\mathbf{A}'$ はその内部パラメータ行列、 $\mathbf{R}$ と $\mathbf{t}$ は基準カメラに対する姿勢と位置を表す回転行列（3×3 行列）と並進ベクトル（3×1 行列）である。この2台のカメラの透視投影行列を平行ステレオに適用したとき、 $\mathbf{A}$ 、 $\mathbf{A}'$ 、 $\mathbf{R}$ 、 $\mathbf{t}$ は以下のようになる。

$$\mathbf{A} = \mathbf{A}' = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3.4)$$

$$\begin{bmatrix} \times \\ \times \\ \times \end{bmatrix} \quad (3.5)$$

$$\begin{bmatrix} \times \\ \times \\ \times \end{bmatrix} \quad (3.6)$$

ここで、 f はカメラ焦点距離、 (c_x, c_y) は画像中心（光軸と画像面の交点）の座標。 $\mathbf{R}$ は他方のカメラの回転行列を表すが、今回は平行ステレオを前提にしているので、左右のカメラは回転はしていないので基準カメラと同様に

単位行列となる。 b は基線長である。

また、その内部パラメータ、外部パラメータを用いてワールド座標 (X,Y,Z) をカメラ座標 (x_l, y_l) 、 (x_r, y_r) に変換する式は以下のようになる。

$$\begin{pmatrix} x_l \\ y_l \\ 1 \end{pmatrix} \sim P \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.7)$$

$$\begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} \sim P' \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

式(3.7)は同次式であるため等式に変換し P を計算すると次のようになる。

$$\begin{pmatrix} x_l \\ y_l \\ 1 \end{pmatrix} = \alpha \begin{pmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (3.8)$$

$$\begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} = \beta \begin{pmatrix} f & 0 & c_x & -fb \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\begin{aligned} x_l &= \alpha(f + c_x Z) \\ y_l &= \alpha(f + c_y Z) \\ 1 &= \alpha Z \\ x_r &= \beta(f + c_x Z - fb) \\ y_r &= \beta(f + c_y Z) \\ 1 &= \beta Z \end{aligned} \quad (3.9)$$

(3.9)の式から α , β を消去すると次のようになる。

$$\begin{aligned}x_l &= \frac{fX + c_x Z}{Z} \\y_l &= \frac{fY + c_y Z}{Z} \\x_r &= \frac{fX + c_x Z - fb}{Z} \\y_r &= \frac{fY + c_y Z}{Z}\end{aligned}\quad (3.10)$$

式(3.10)の一番目の式から三番目の式を引くと、

$$\begin{aligned}x_l - x_r &= \frac{fX + c_x Z - (fX + c_x Z - fb)}{Z} \\x_l - x_r &= \frac{fb}{Z} \\f &= \frac{Z(x_l - x_r)}{b}\end{aligned}\quad (3.11)$$

となり(3.2)が導出できる。

3.1.1 実距離測定

2.2.3 節の手順で生成したステレオ画像は、一台のカメラで撮影した生画像から再構成したものである。したがって、平行ステレオの条件になっていると考えられる。これらの画像間の視差を求めることで、距離に換算できる。二枚の画像の視差を d 、基線長を b 、レンズの焦点距離 f と定義する。今回の実験では、視差を被写体の重心位置から計算する。この方法を採用した理由は、サブピクセルの精度で被写体の位置を計測することができると考えたからである。

2.2.3 節で述べたように Lytro で撮影した画像は複数のステレオ画像に再構成することができる。これらのうち、マイクロレンズの中央から左2画素、右に3画素の箇所から再構成した2枚のステレオ画像を用意する。この位置を選択した理由は、再構成された画像がはっきりとしており、視差を明確に視認できたからである。マイクロレンズに対して、周辺に位置する画素から再構成された画像は、ぼけやモアレが発生しており、ステレオ計測には向かないと判断した。図 3.2 に生画像から再構成したステレオ画像の例を示す。



図 3.2 Lytro の生画像から生成したステレオ画像の例

3.1.2 提案手法の基本的な考え方

2.2.3 節で述べたように Lytro の生画像を再構成することで、多数のステレオ画像に分解できる。これは、Lytro が複数のカメラを内蔵しており、ステレオ撮影可能であることと等価である。これを利用して被写体までの実距離を計測する。Lytro を用いる利点は 1 ショットでステレオ画像を撮影することである。一般的なカメラを用いる場合、二台を平行に配置して撮影するか、一台のカメラを平行移動させる。これでは手間がかかり面倒である。Lytro であれば一般的なカメラよりも容易に距離計測ができる。

また 4 章では、Lytro で得たステレオ画像対に対してカメラキャリブレーションを行い内部パラメータ、外部パラメータを求め、それぞれのカメラの計算上の焦点距離と基線長などから本章の実験結果を検証する。

3.2 Lytro を用いた実距離の測定

Lytro から抽出した 2 枚の画像が平行ステレオ（標準ステレオ）で取得した 2 枚の画像になっていると仮定して、実距離を計測した。

3.2.1 実験用画像の取得

図 3.3 のように Lytro をカメラ台に固定する。Lytro から一定の距離の位置に A4 サイズの黒色画用紙を配置し、そこまでの距離を計測する。黒色画用紙は、Lytro の筐体の先端から 60.6cm を最も近い位置として、そこから

30.3cm ごとに遠ざけてゆき、484.8cm まで 15 回撮影を行った。被写体を黒色にした理由は、画像中の被写体の重心を求めるために画像を容易に二値化できるからである。被写体と Lytro の距離は、床面に貼られたパネルの一边の長さを目安とし、巻き尺で距離を確認した。したがって、mm オーダーでの誤差は生じていると考えられる。



図 3.3 実験風景

上記と同様の実験を撮影距離を 1cm～25cm、被写体を黒色の 1cm 角、2cm 角、3cm 角の画用紙として撮影した。1cm～4cm までは 1cm 角の画用紙、5cm～10cm までは 2cm 角の画用紙、11cm～25cm までは 3cm 角の画用紙をそれぞれ用いた。背景には白色の板を設置し被写体を二値化しやすいようにしている。

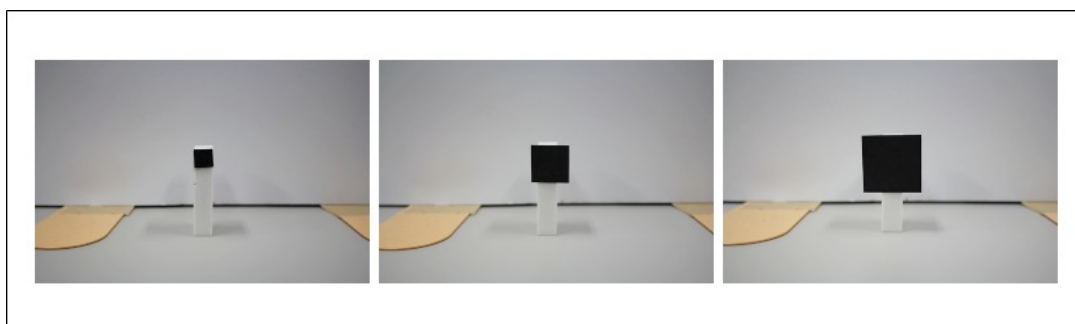


図 3.4 1cm 角、2cm 角、3cm 角の被写体



図 3.5 1cm～25cm の実験風景

3.2.2 Lytro の生画像からステレオ画像への再構成

2.2.3 で述べたように、Lytro の生画像から、マイクロレンズに対して同じ位置にある画素を抽出して並べ直した複数の小画像はステレオ画像を構成する。一つのマイクロレンズは 50 画素程度をカバーするので、50 枚のステレオ画像を再構成することができる。今回の実験では、その中からマイクロレンズ中心から右に 2 つ目、左に 3 つ目の画素から再構成されるステレオ画像 2 枚を利用した。その一方を右画像、他方を左画像と呼ぶ。この二枚を選択した理由は目で見た場合比較的明るく、状態の良い画像であったためである。

3.2.3 視差の計測

ステレオ三次元計測は、被写体が右画像で観察される位置と左画像で観察される位置の差、すなわち視差を距離に換算する。視差を求めるためには、通常、ステレオマッチングによって左右画像における被写体の位置を同定する必要がある。しかし、今回の実験のように、既知の被写体が一っただけ撮影されているような場合には、ステレオマッチングを行う必要はない。

画像中の被写体位置を高精度に求めることは重要である。そのため、ある程度の面積を有する被写体を撮影し、被写体像を2値化した後、その重心を求めることで、被写体位置を高精度化した。

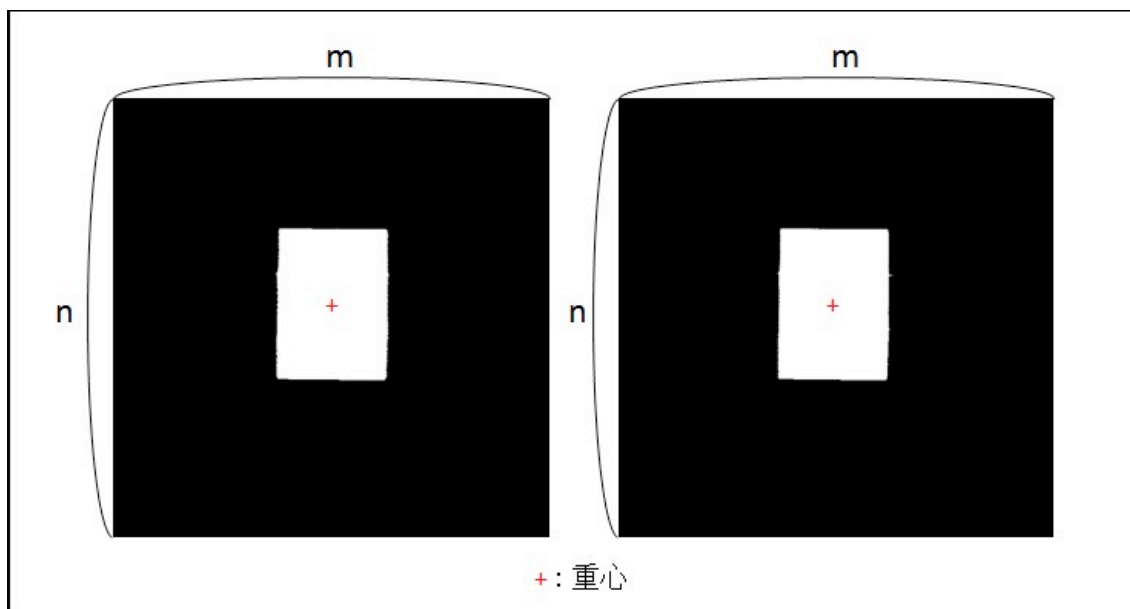


図 3.6 2 値化された領域の重心

$f(x,y)$ を 2 値画像の全体とし、その中で被写体領域の重心の x 座標は式 (3.12) で求めることができる。

$$x_g = \frac{\sum_{y=1}^n \sum_{x=1}^m x f(x,y)}{\sum_{y=1}^n \sum_{x=1}^m f(x,y)} \quad (3.12)$$

今回は、平行ステレオであることを仮定しているので、 x 軸方向の重心だけに注目し視差を求め、 y 軸方向の重心は考慮しない。

3.3 実験結果

図 3.3 で示したように、被写体を Lytro の筐体前端から 60.6cm の地点から 484.8cm まで 30.3cm 刻みで計 15 枚撮影した。その生画像の一枚を図 3.7 に示す。

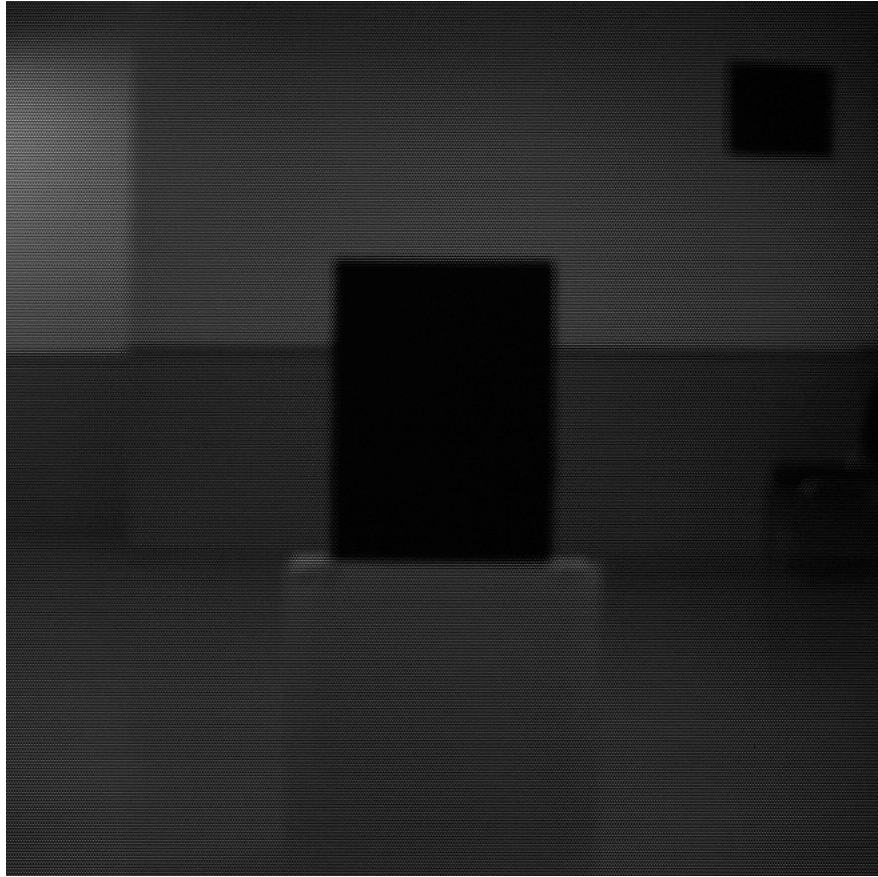


図 3.7 カメラから 121.2cm 地点の被写体の生画像

そして、各距離における生画像からステレオ画像を再構成する。図 3.8 に再構成されたステレオ画像の例を示す。

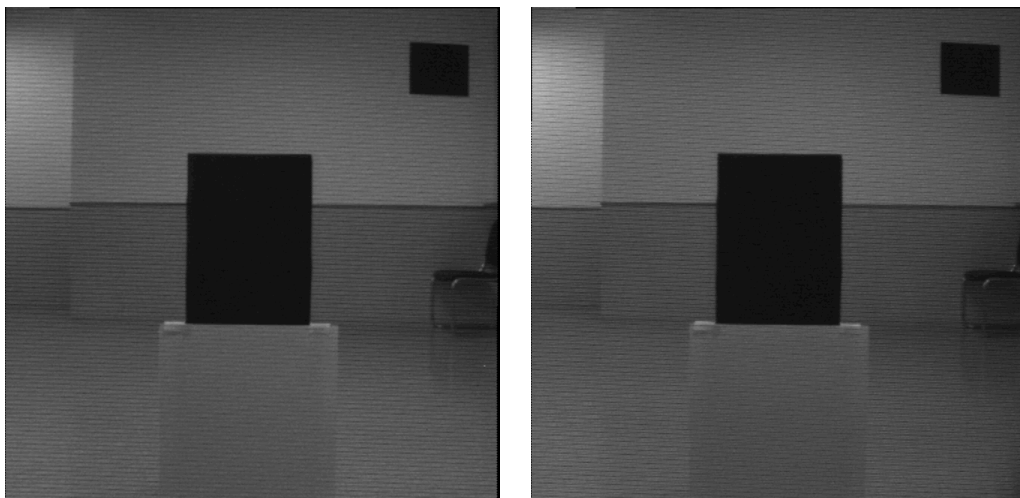


図 3.8 左ステレオ画像と右ステレオ画像

ステレオ画像に対して被写体部分を二値化する。図 3.9 に二値化された画像の例を示す。

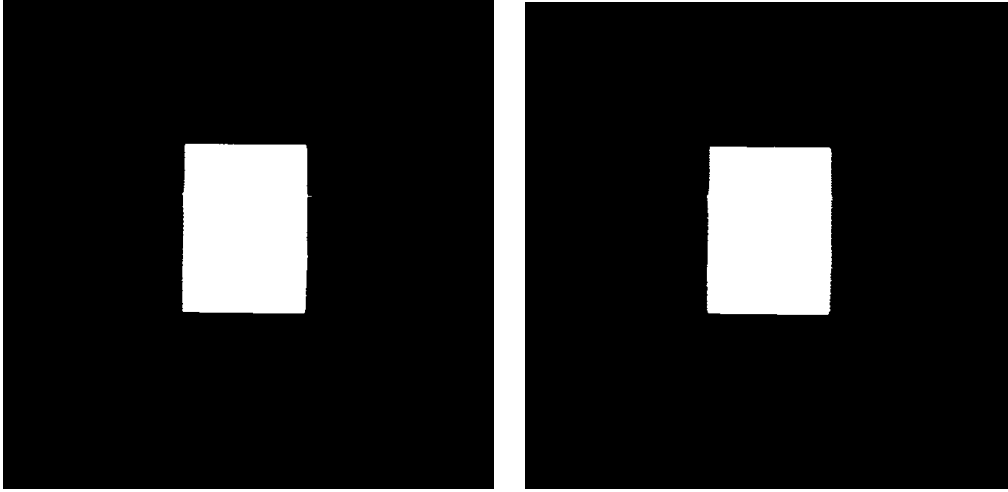


図 3.9 左右のステレオ画像を二値化した画像

二値画像から、被写体の x 軸方向の重心を求める。左画像の重心値－右画像の重心値が視差となる。

式(3.2)を変形すると

$$fb = Z(x_l - x_r) \quad (3.13)$$

となり、ある距離 Z における視差 $(x_l - x_r)$ を用いてカメラの焦点距離と基線の積を定数として求める。この定数を用いて、別のステレオ画像から求めた視差を利用して距離を推定する。

例えば、被写体と Lytro の距離が 60.6cm で撮影して求めた定数を用いて被写体と Lytro の距離が未知である画像の距離 d を求める方法は次のようになる。

初めに、60.6cm で撮影した時の定数 $fb_{60.6}$ を求める。

$$fb_{60.6} = 60.6(x_{l60.6} - x_{r60.6}) \quad (3.14)$$

次に、(3.13)の式の左辺に ×、右辺の $(x_l - x_r)$ に d における視差 $(x_{ld} - x_{rd})$ を代入する。

$$\begin{aligned} fb_{60.6} &= Z_d(x_{ld} - x_{rd}) \\ Z_d &= \frac{fb_{60.6}}{(x_{ld} - x_{rd})} \end{aligned} \quad (3.15)$$

となり Z_d を推定することが出来る。今回は 60.6cm から 484.8cm まで 30.3cm ごとに 15 の異なる距離で求めた定数を利用し、15 個の定数を用いて画像中の被写体と Lytro の距離を推定した。

視差－距離の関係のグラフを以下に示す。

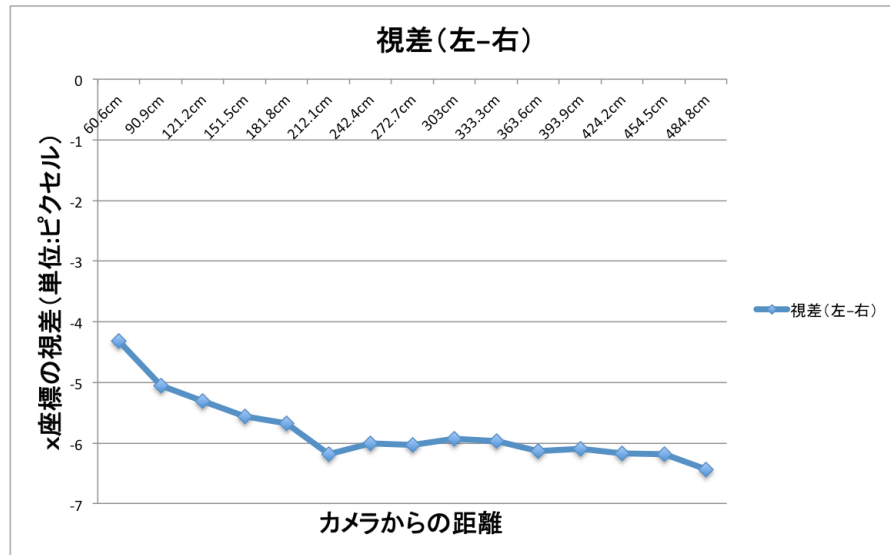


図 3.10 視差－距離グラフ

平行ステレオにおける視差は距離の逆数に比例し、被写体が遠くなれば視差 0 に漸近する。それに対して、図 3.10 のグラフは、視差が距離の逆数に比例するという傾向はあるが、無限遠の被写体に対する視差は -6 画素～-7 画素の間に漸近している。この結果から考えるに、Lytro で撮影して得られるステレオ画像は 6 画素～7 画素分位置をずらして生成していると思われる。

次に、視差を 6.44 画素上にシフトし距離推定を行った。例として 272.7cm 地点で求めた定数 $f_{272.7}$ と、ほかの地点で求めた視差を利用して距離を推定したグラフを以下に示す。

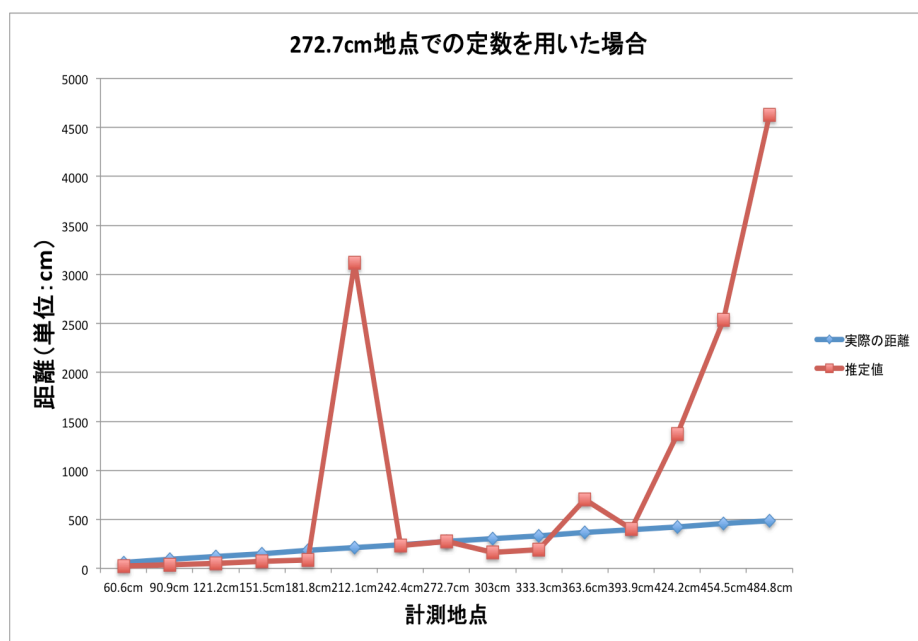


図 3.11 距離推定グラフ

図 3.11 の距離推定のグラフは実際の距離と異なる結果となった。212.1cm 地点、363.6cm 地点、424.2 地点以降では極端に推定値が実際の距離と異なった。これは 212.1cm 地点より遠い位置では、視差による誤差が大きく、推定される距離への影響が大きいためこのような結果になったと考えられる。

次に 1cm~25cm での視差と視差が 0 になる位置を求めた。その時の視差-距離グラフは以下ようになった。

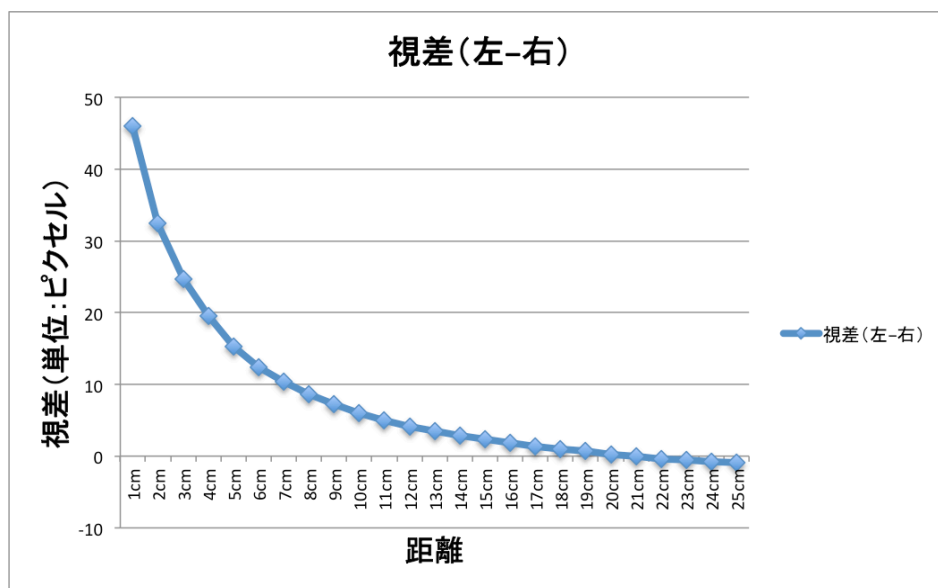


図 3.12 視差-距離グラフ

距離が近ければ視差は大きく、距離が遠ければ視差は小さくなった。こちらのグラフは、距離が視差の逆数に比例するという関係を、より明確に読み取ることができる。また、視差が 21cm と 22cm の間で 0 になっていることが分かった。つまり、Lytro は視差が 21cm~22cm で 0 になるように調整されている。

また、このグラフを 6.44 画素分上にシフトし先ほどと同様の方法で実距離推定を行った。定数は 12cm 地点のものを使用した。そのグラフを次に示す。

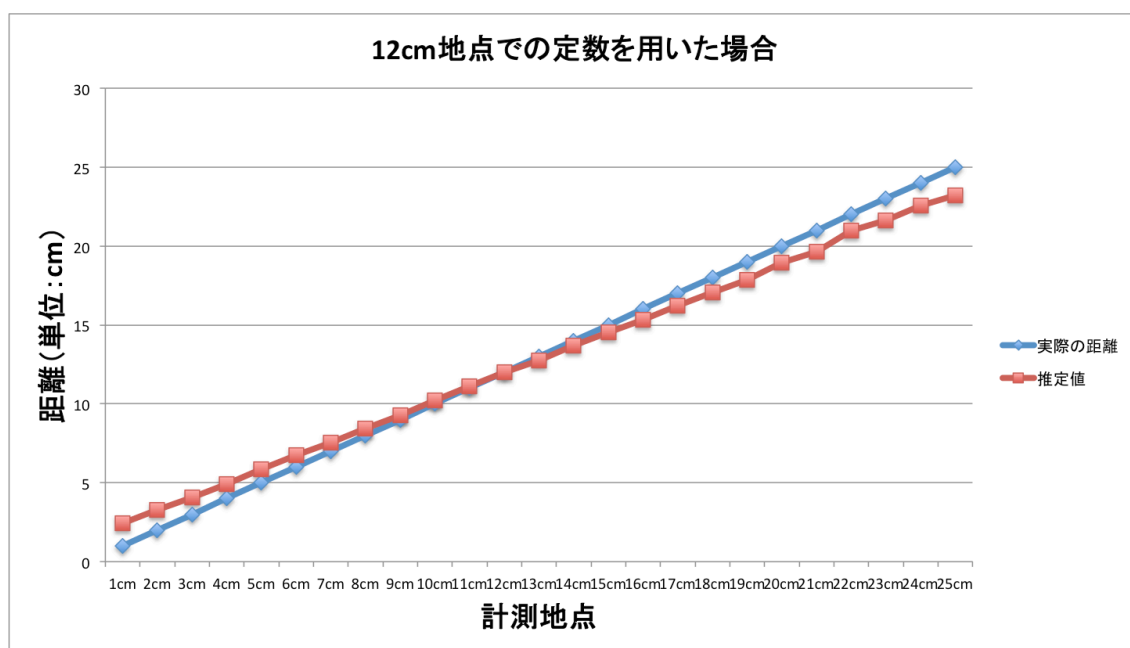


図 3.13 1cm~25cm での距離推定グラフ

1cm 地点での誤差はプラス 1.4cm、25cm 地点での誤差はマイナス 1.8cm であった。定数を用いた地点付近は比較的良好な結果が得られた。

3.4 最小二乗法による近似式

3章で計測したデータを用いて、最小二乗法で近似式を求める。距離と視差の関係は一般的には反比例で表せる。しかし、Lytro は無限遠での視差は 0 ではないことはこれまでの実験で明らかである。図 3.10、図 3.12 から推測するに近似式は

$$y = \frac{a}{x} + b \quad (3.16)$$

となる。 x 軸は距離、 y 軸は視差とする。測定で得られた数値の組の集合を

$$(x, y) = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \quad (3.17)$$

とすると、この式の a 、 b は次の式で求められる。

$$\begin{aligned} a &= \frac{\left(\sum_{k=1}^n \frac{1}{x_k} \right) \left(\sum_{k=1}^n y_k \right) - n \left(\sum_{k=1}^n \frac{y_k}{x_k} \right)}{\left(\sum_{k=1}^n \frac{1}{x_k} \right)^2 - n \left(\sum_{k=1}^n \frac{1}{x_k^2} \right)} \\ b &= \frac{\left(\sum_{k=1}^n \frac{1}{x_k} \right) \left(\sum_{k=1}^n \frac{y_k}{x_k} \right) - \left(\sum_{k=1}^n y_k \right) \left(\sum_{k=1}^n \frac{1}{x_k^2} \right)}{\left(\sum_{k=1}^n \frac{1}{x_k} \right)^2 - n \left(\sum_{k=1}^n \frac{1}{x_k^2} \right)} \end{aligned} \quad (3.18)$$

入力データ 40 組と式(3.18)より、 $a=60.56$ 、 $b=-3.00$ を得た。したがって、距離と視差の近似式は次のように表すことができる。

$$y = \frac{60.56}{x} - 3 \quad (3.19)$$

式(3.19)の近似式では無限遠で視差は -3 に収束し、20.1cm～20.2cm の間で視差が 0 になる結果になった。

また距離を求める場合は式(3.19)を変形した

$$y = \frac{60.56}{y+3} \quad (3.20)$$

を用いる。

1cm～25cm における、式(3.19)で求めた視差の近似値と実測値のグラフを次に示す。

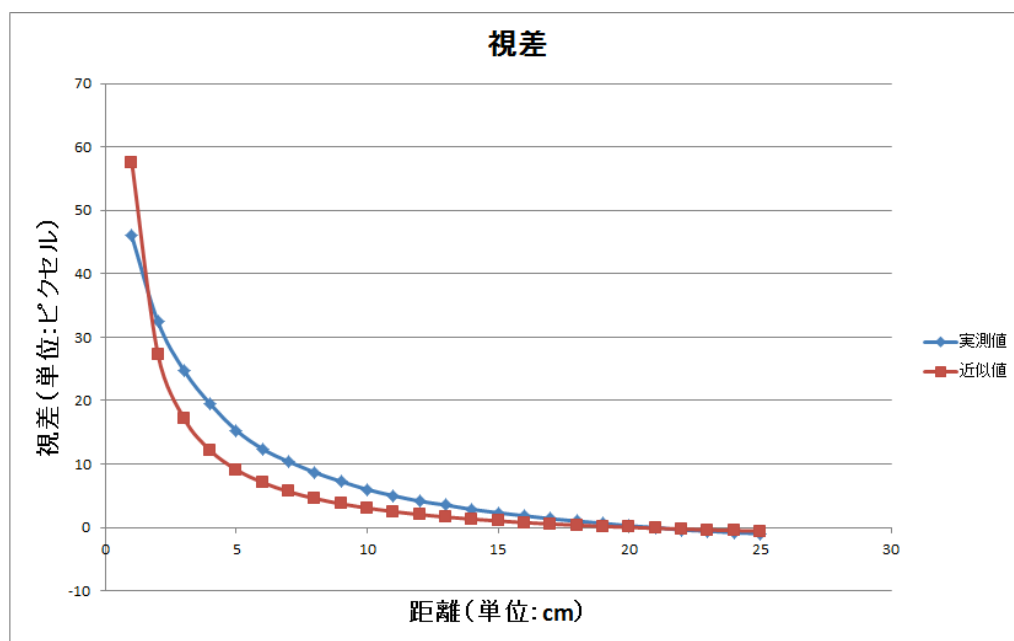


図 3.14 1cm～25cm における視差の近似値と実測値の比較

図 3.14 の 1cm～25cm における視差では、近似値は 2cm～20cm の区間では実測値よりも小さい値をとっている。視差の最大誤差は 1cm 地点における 11.57 ピクセルで最小誤差は 22cm 地点における 0.13 ピクセルであった。この近似式を用いて距離を推定した。実際の距離と推定値の比較のグラフを次に示す。

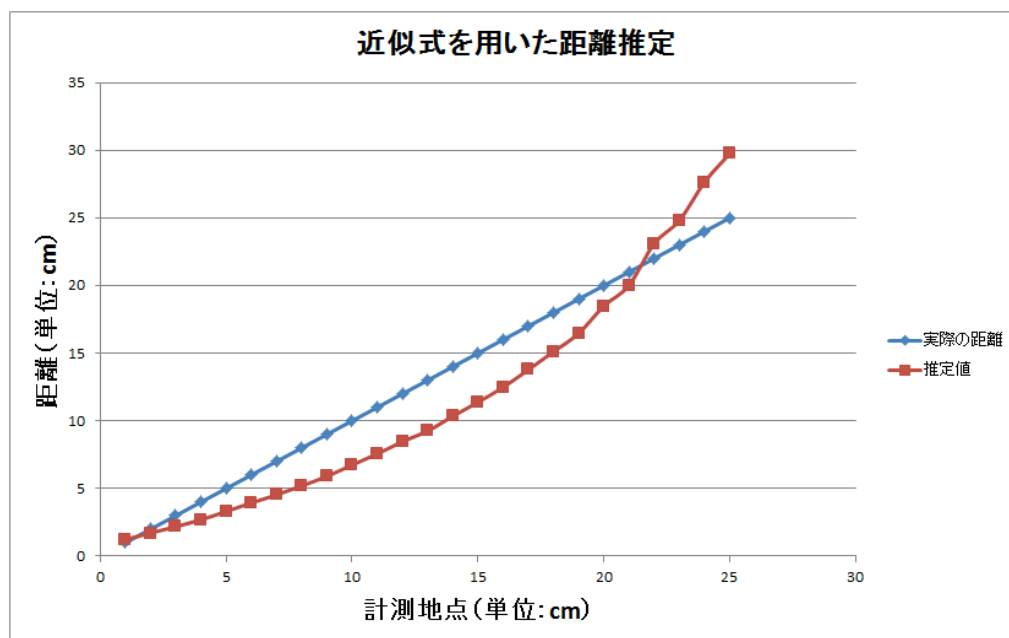


図 3.16 1 cm～25cm における実際の距離と推定値の比較

60.6cm～484.8cm における、式(3.19)で求めた視差の近似値と実測値のグラフを以下に示す。

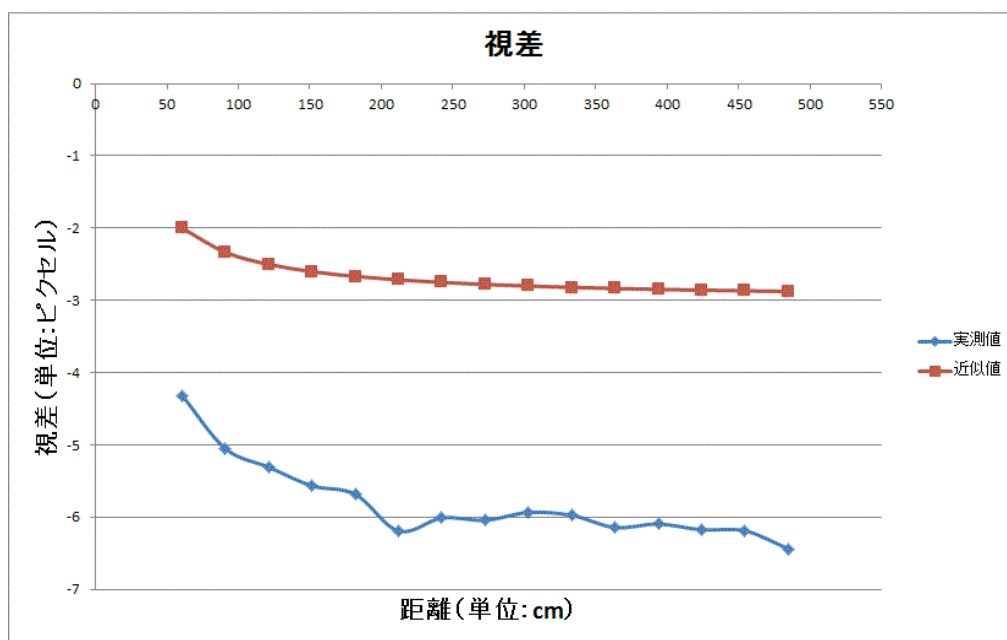


図 3.15 60.6～484.cm における視差の近似式と実測値の比較

距離の最大誤差は 25cm 地点でのプラス 4.75cm で最小誤差は 1cm 地点でのマイナス 0.24cm であった。

図 3.15 の 60.6cm～484.8cm における視差は、実測値は近似値より約 3 ピクセル視差が大きいという結果になった。このまま式(3.19)の近似式を適用して距離を推定した場合、近似式の y 軸方向のシフト量より実測値の視差の量が大きく式(3.20)の $y+3$ の値がマイナスになってしまい距離を正しく推定することができない。つまり、この距離で求めた近似式を用いての距離測定は適用できない。60.6cm 以降のデータ量を増やし最小二乗法で近似式を求めればさらにフィットした関数を求められたであろう。

4. Lytro のキャリブレーションと実距離測定の実証

3章で述べたように Lytro の生画像から再構成する右画像、左画像は、Lytro が内蔵する仮想的な右カメラ、左カメラで撮影した画像であると考えられる。第4章では、右カメラと左カメラを同時にカメラキャリブレーションすることで、ステレオカメラとしての内部パラメータ、外部パラメータを求める。これらのカメラパラメータを用いて、3章で得た実験データを検証する。3章では焦点距離を f としていたが、4章ではカメラキャリブレーションによる出力では x 軸方向、 y 軸方向の焦点距離を区別しているので f_x , f_y と記述する。

4.1 カメラキャリブレーション

カメラキャリブレーションについて説明する。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、それをカメラで観察した画像座標の組から、カメラの内部パラメータと外部パラメータを求める計算手順のことである。OpenCV に Zhang の手法によるカメラキャリブレーションが実装されているため、今回はこれを利用した。

4.1.1 同次座標を用いた透視投影モデル

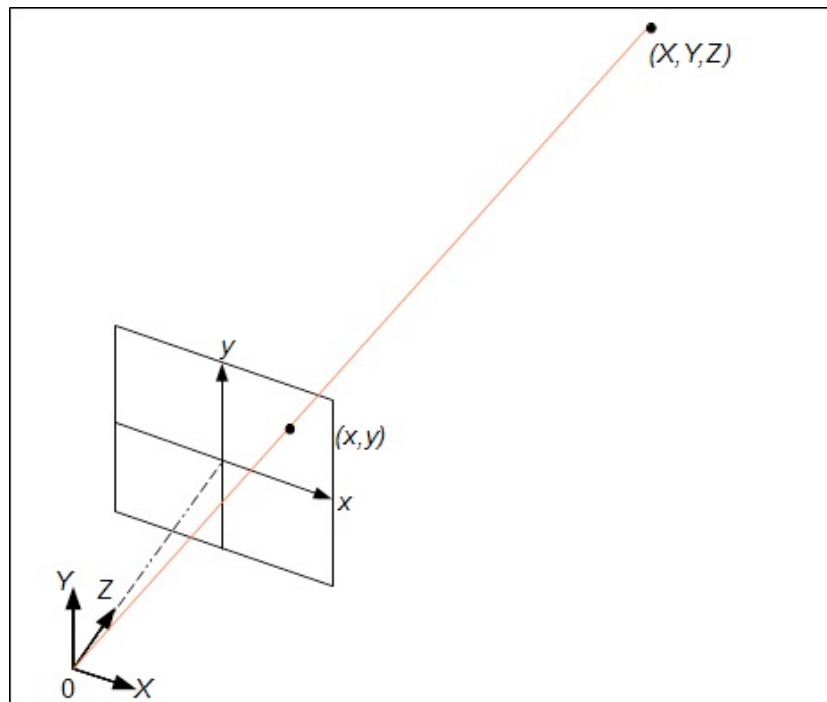


図 4.1 透視投影モデル

カメラによる透視投影は図 4.1 のようにモデル化できる。透視投影とは三次元物体を、三次元座標と投影中心を結ぶ直線が投影面と交わる位置に投影することである。人間の視覚は透視投影を行っていると考えられるので、見えたままを二次元平面に描画することと言える。三次元空間を定義するために、カメラの光学中心を原点とし、 z 軸をカメラの光軸と一致させ x 軸、 y 軸は画像の横方向と縦方向にとる。このような座標系をカメラ座標系と呼ぶ。一方、画像平面上の位置を表す座標系としては、カメラの光軸と画像平面が交わる交点を原点とし、画像の横方向と縦方向に x 軸、 y 軸を決める。

このとき、空間座標 $\boxed{\times}$ と画像平面上の座標 (x,y) との間には次の関係式が成立する。

$$\boxed{\times} \quad (4.1)$$

ただしここでは焦点距離を 1 に正規化している。

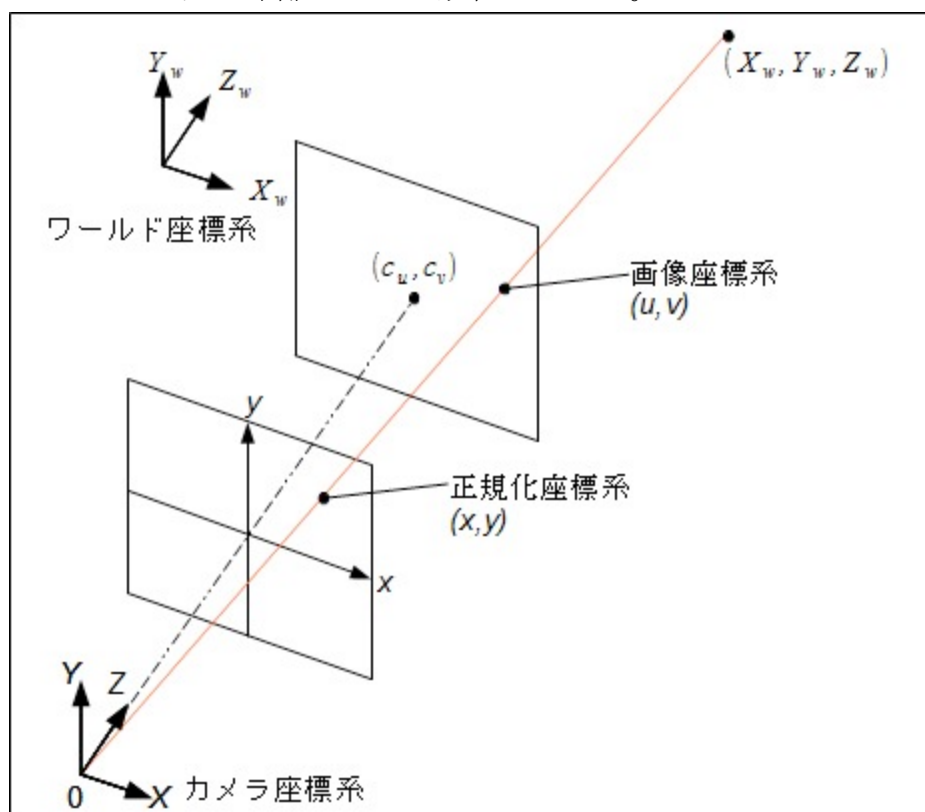


図 4.2 透視投影モデルの一般化

次に空間と画像平面との関係を一般化する。一般に空間表す座標系は空間中のある適当な位置に固定されたワールド座標系を用いる。ワールド座標で表された空間座標を $\boxed{\times}$ とすると同じ点をカメラ座標系で表した座標との間には次の関係式が成立する。

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = R \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} + t \quad (4.2)$$

ここで R, t は以下に示すような、 3×3 の回転行列と三次元の平行移動ベクトルである。

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (4.3)$$

$$t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}$$

一方、画像中の位置は、原点を画像中の適当な位置に定め、長さの単位は画素(pixel)を用いる。このような画像中の位置を表す座標は画像座標 (u, v) と呼ばれている。また、すでに出てきた座標 (x, y) は正規化座標と呼ばれている。この時、正規化座標系と画像座標の関係は以下の式で表される。

$$x = \frac{\delta_u(u - c_u)}{f_x} \quad (4.4)$$

$$y = \frac{\delta_v(v - c_v)}{f_y}$$

ここで f_x, f_y はカメラの焦点距離、 $\boxed{\times}$ はそれぞれ横方向と縦方向の画素の物理的な間隔、 $\boxed{\times}$ は画像座標における光軸と画像平面との交点の位置(画像中心)を表す。

(4.1)から(4.4)の式よりワールド座標 $\boxed{\times}$ と画像座標 (u, v) の間は、(4.2)と(4.3)より

$$\begin{aligned} X &= r_{11}X_w + r_{12}Y_w + r_{13}Z_w + t_1 \\ Y &= r_{21}X_w + r_{22}Y_w + r_{23}Z_w + t_2 \\ Z &= r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_3 \end{aligned} \quad (4.5)$$

(4.5)と(4.1)より

$$\begin{aligned} x &= \frac{r_{11}X_w + r_{12}Y_w + r_{13}Z_w + t_1}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_3} \\ y &= \frac{r_{21}X_w + r_{22}Y_w + r_{23}Z_w + t_2}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_3} \end{aligned} \quad (4.6)$$

(4.6)と(4.4)より

$$\begin{aligned} \frac{\delta_u(u - c_u)}{f_x} &= \frac{r_{11}X_w + r_{12}Y_w + r_{13}Z_w + t_1}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_3} \\ \frac{\delta_v(v - c_v)}{f_y} &= \frac{r_{21}X_w + r_{22}Y_w + r_{23}Z_w + t_2}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_3} \end{aligned} \quad (4.7)$$

$\Leftrightarrow$

$$\begin{aligned} u &= \frac{f_x}{\delta_u} \frac{r_{11}X_w + r_{12}Y_w + r_{13}Z_w + t_1}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_3} + c_u \\ v &= \frac{f_y}{\delta_v} \frac{r_{21}X_w + r_{22}Y_w + r_{23}Z_w + t_2}{r_{31}X_w + r_{32}Y_w + r_{33}Z_w + t_3} + c_v \end{aligned} \quad (4.8)$$

この数式に出てくる $f_x, f_y, \delta_u, \delta_v, c_u, c_v$ はカメラにより決定される内部パラメータである。一方 $\mathbf{R}, \mathbf{t}$ はワールド座標に対して位置と姿勢で決定される関数であり外部パラメータとなる。

(4.8)の式を見通しよく記述するために同次座標を導入する。まず、それぞれの座標を以下のように同次座標でベクトル表記する。

$$\tilde{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \tilde{m} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \tilde{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tilde{X}_w = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

このとき(4.1)の式は次のように表現できる。

$$\boxed{\times} \quad (4.9)$$

ベクトル記号の $\sim$ は同次座標を、 $\sim$ は両辺が定数倍の違いを許して等しいことを意味する。 $\mathbf{I}$ は 3×3 の単位行列を、 $\mathbf{0}$ は三次元のゼロベクトルを表し、 $(\mathbf{I}|\mathbf{0})$ は 3×4 の行列である。

また、(4.2)の関係式は次のように表現される。

$$\boxed{\times} \quad (4.10)$$

ここで、 $\mathbf{M}$ はカメラの外部パラメータから構成される。4×4 の行列である。

$$\mathbf{M} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.11)$$

さらに、(4.4)の関係式は次のように表される。

$$\boxed{\times} \quad (4.12)$$

$\mathbf{A}$ は次のようなカメラの内部パラメータから成る 3×3 の行列である。

$$\mathbf{A} = \begin{pmatrix} \frac{f_x}{\delta_u} & 0 & c_u \\ 0 & \frac{f_y}{\delta_v} & c_v \\ 0 & 0 & 1 \end{pmatrix} \quad (4.13)$$

以上の各座標間の関係式をまとめ、(4.8)に対応する関係式は次のようになる。

$$\begin{aligned} \tilde{m} &\sim A \tilde{x} \\ \tilde{m} &\sim A(I | 0) \tilde{X} \\ \tilde{m} &\sim A(I | 0) \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \tilde{X}_w \\ \tilde{m} &\sim A(\mathbf{R} | \mathbf{t}) \tilde{X}_w \end{aligned} \quad (4.14)$$

ここで、 $\mathbf{A}(\mathbf{R} | \mathbf{t})$ はまとめて 3×4 の行列となるので、これを $\mathbf{P}$ とおくと次のような式で表現される。

$$\boxed{\times} \quad (4.15)$$

この $\mathbf{P}$ は透視投影行列と呼ばれている。

4.1.2 カメラキャリブレーションの手順

4.1.1 節で述べたことを利用してカメラキャリブレーションの方法について説明する。

まず透視投影行列 $\mathbf{P}$ の各要素を次のように置く。

$$\mathbf{P} = \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{pmatrix} \quad (4.16)$$

この時、(4.15)の式を要素で書くと以下のようにになる。

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \quad (4.17)$$

上の式より座標(u,v)を求めると次のようになる。

$$\begin{aligned} u &= \frac{p_{11}X_w + p_{12}Y_w + p_{13}Z_w + p_{14}}{p_{31}X_w + p_{32}Y_w + p_{33}Z_w + p_{34}} \\ v &= \frac{p_{21}X_w + p_{22}Y_w + p_{23}Z_w + p_{24}}{p_{31}X_w + p_{32}Y_w + p_{33}Z_w + p_{34}} \end{aligned} \quad (4.18)$$

$\boxed{\times}$ と置き、残りの $\mathbf{P}$ の各要素について整理すると次のようになる。

$$\begin{aligned} X_w p_{11} + Y_w p_{12} + Z_w p_{13} + p_{14} - u X_w p_{31} - u Y_w p_{32} - u Z_w p_{33} &= u \\ X_w p_{21} + Y_w p_{22} + Z_w p_{23} + p_{24} - u X_w p_{31} - u Y_w p_{32} - u Z_w p_{33} &= v \end{aligned} \quad (4.19)$$

空間中の座標 $\boxed{\times}$ とその投影点の画像上の座標(u,v)が得られれば、上の式のように $\mathbf{P}$ に関する二つの方程式が得られる。未知数は全部で 11 個あるので、既知の位置座標の組が最低 6 組あれば透視投影行列 $\mathbf{P}$ を求めることができる。

ここで、既知の空間座標と画像座標をそれぞれ (X_{w1}, Y_{w1}, Z_{w1}) 、 (X_{w2}, Y_{w2}, Z_{w2}) 、... と $\boxed{\times}$ 、 $\boxed{\times}$ 、... として(4.19)から得られる $\mathbf{P}$ の各要素に関する連立方程式を、行列を用いて表すと以下のようになる。

$$\begin{pmatrix} X_{w1} & Y_{w1} & Z_{w1} & 1 & 0 & 0 & 0 & 0 & -u_1 X_{w1} & -u_1 Y_{w1} & -u_1 Z_{w1} \\ 0 & 0 & 0 & 0 & X_{w1} & Y_{w1} & Z_{w1} & 1 & -v_1 X_{w1} & -v_1 Y_{w1} & -v_1 Z_{w1} \\ X_{w2} & Y_{w2} & Z_{w2} & 1 & 0 & 0 & 0 & 0 & -u_2 X_{w2} & -u_2 Y_{w2} & -u_2 Z_{w2} \\ 0 & 0 & 0 & 0 & X_{w2} & Y_{w2} & Z_{w2} & 1 & -v_2 X_{w2} & -v_2 Y_{w2} & -v_2 Z_{w2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \end{pmatrix}$$

(4.20)

式(4.20)を最小二乗法を用いて解くことによって透視投影行列の係数 $p_{11} \sim p_{33}$ を求めることができる。このように推定した透視投影行列を 3×3 の内部パラメータ行列と 3×4 の外部パラメータ行列に分けることで、焦点距離、画像中心、回転行列、並進ベクトルなどを求めることができる。なお、ここで述べた方法は線形解法による一般的なカメラキャリブレーションの方法である。OpenCV に実装されている Zhang の方法は、これよりも高度・高精度なものであり、同時にレンズ歪み係数を求めることができる。

4.1.3 カメラキャリブレーションの実施

Lytro の内部パラメータと外部パラメータを調べるためにチェスボードパターンを用いてカメラキャリブレーションを行った。カメラキャリブレーションとはカメラ固有の内部パラメータと、ある地点においてワールド座標系におけるカメラの位置姿勢(回転量、平行移動量)を表現する外部パラメータを求める処理である。カメラキャリブレーションを行うことでワールド座標上の三次元座標点がカメラ画像のどこに投影されるのか計算できる。また、カメラ特有のレンズ歪みを補正することもできる。

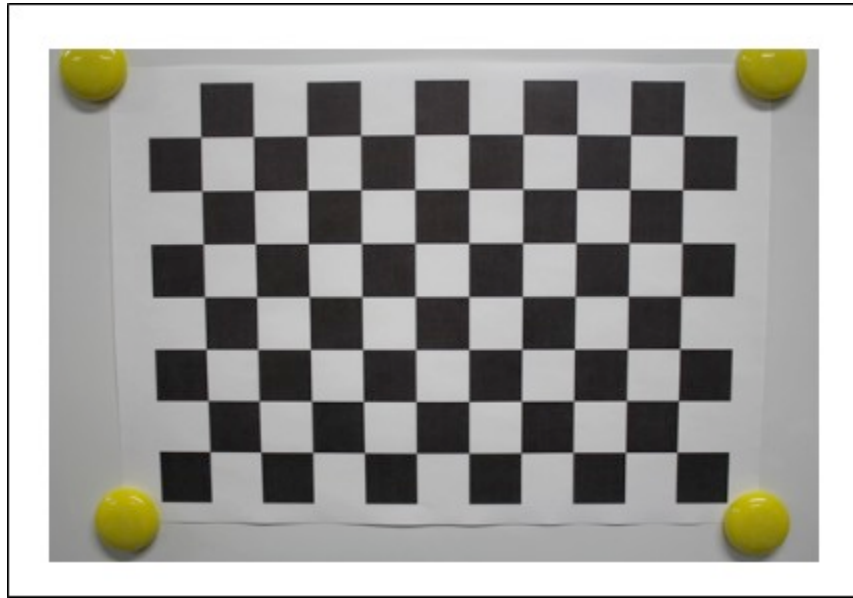


図 4.3 10×7 チェスボードパターン

使用したものは交点数が 10×7 の一マスが 24mm 四方のチェスボードパターンで、右カメラ、左カメラそれぞれ 44 枚の画像を使用してキャリブレーションを行った。各画像ごとにチェスボードパターンの交点位置を画像処理で抽出するとともに、対応する三次元座標として mm を単位とする (X,Y) 座標を与える。Z 座標は 0 を設定する。

カメラキャリブレーションを行うにあたって満たさなければ成らない条件がある。先ほど述べたようにカメラキャリブレーションはカメラの内部パラメータ、外部パラメータを求めることである。内部パラメータの内容は x, y 軸方向の焦点距離、画像中心の座標の 4 つで構成される。また、外部パラメータは x, y, z 軸方向の 3 つの回転パラメータ、 x, y, z 軸方向の 3 つの平行移動パラメータで構成される。つまり、4 つの内部パラメータと 6 つの外部パラメータの合計 10 個のパラメータを求めることになる。ここで、N 個のコーナーを持つ K 枚のチェスボード画像があるとする。そうすると、K 枚の画像は $2NK$ 個の拘束条件を与える(2 が掛かっているのは、画像上の各点が x と y 座標を持っているから)。未知数は、4 つの内部パラメータと $6K$ 個の外部パラメータである($6K$ とは K 枚の画像についてそれぞれに外部パラメータが存在するから)。この時、解が求まるためには

$$2NK \geq 6K + 4$$

が成立する必要がある。今回は 10×7 交点のチェスボードを 44 枚使用してい

るのでこの式に代入すると

$$2 \times 70 \times 44 \geq 6 \times 44 + 4$$

$$6160 \geq 268$$

となり拘束条件を満たしていることがわかる。

4.2 実験結果

3章の実験ではカメラの焦点距離と基線長がわからないため、ある距離で計測したカメラの焦点距離と基線長の積を定数として、それを他の距離で求めた視差を利用して距離を推定していた。ここでは、ステレオカメラキャリブレーションでこれらの定数を求め、実験結果を検証する。

OpenCV の関数を用いてキャリブレーションを行いカメラの焦点距離、基線長を求めた。キャリブレーションにあたって、チェスボードパターン1辺24mmの10×7で異なる方向から撮影した44枚を使用した。その時の内部パラメータ、外部パラメータは以下のようになった。

表 4.1 右カメラ内部パラメータ

右内部パラメータ	x 軸	y 軸
焦点距離（画素単位の）	940.20	941.45
画像座標中心	322.98	308.47

表 4.2 左カメラ内部パラメータ

左内部パラメータ	x 軸	y 軸
焦点距離（画素単位の）	939.59	941.65
画像座標中心	312.59	308.53

表 4.3 右カメラ外部パラメータ

右外部パラメータ	x 軸	y 軸	z 軸
平行移動量（mm）	110.45	89.10	502.36
回転量（ラジアン）	-2.1779	2.2226	0.2100

表 4.4 左カメラ外部パラメータ

左外部パラメータ	x 軸	y 軸	z 軸
平行移動量	114.84	89.06	501.69
回転量	-2.1812	2.2259	0.2115

平行移動量の単位はmmである。表 4.3、表 4.4 の回転量は各軸まわりの回転角をラジアン単位で表記したものである。これらの数値から、左カメラと右カメラの相対的な回転量を求めるために、まず、各軸周りの回転角による表現（オイラー角による表現）を 3×3 の回転行列による表現に変換する。これは、Rodrigues 変換と呼ばれるもので行うことができる。回転量は OpenCV の Rodrigues 変換で回転行列に変換することができる。回転行列と x, y, z 軸方向の平行移動量は以下ようになる。

$$\begin{aligned}
 R_l &= \begin{pmatrix} -0.9967 & -0.0246 & 0.0779 \\ 0.0158 & -0.9936 & -0.1118 \\ 0.0802 & -0.1102 & 0.9907 \end{pmatrix} \\
 R_r &= \begin{pmatrix} -0.9963 & -0.0247 & 0.0818 \\ 0.0157 & -0.9939 & -0.1090 \\ 0.0840 & -0.1074 & 0.9907 \end{pmatrix}
 \end{aligned} \quad (4.21)$$

表 4.5 左右カメラの距離

	差
x 軸方向	4.39208984
y 軸方向	0.04309082
z 軸方向	0.66696167

左右のカメラ回転行列 R_l 、 R_r から 2 台のカメラ間の相対的な回転行列を求め、それを回転角度に逆変換すると 0.275047 度になった。

そして、これらの内部パラメータ、外部パラメータを用いて計算上で(視差-距離)の関係をグラフとして表した。距離のスケールは 3 章の条件と同様に 60.6cm から 484.8cm までは 30.3cm 刻みで計測した場合と、1cm から 25cm までは 1cm 刻みで計測した場合、30cm から 40cm までは 1cm 刻みで計測した場合でグラフにした。

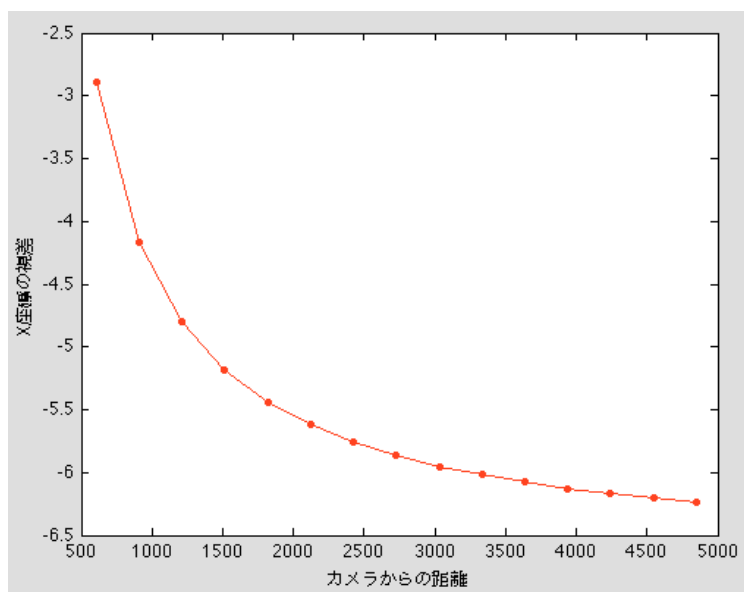


図 4.4 60.6～484.8cm の視差-距離グラフ

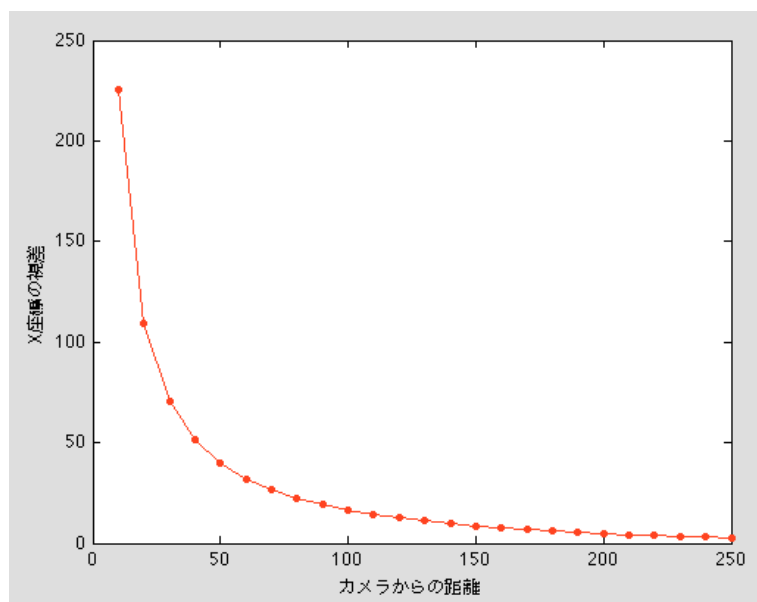


図 4.5 1～25cm の視差-距離グラフ

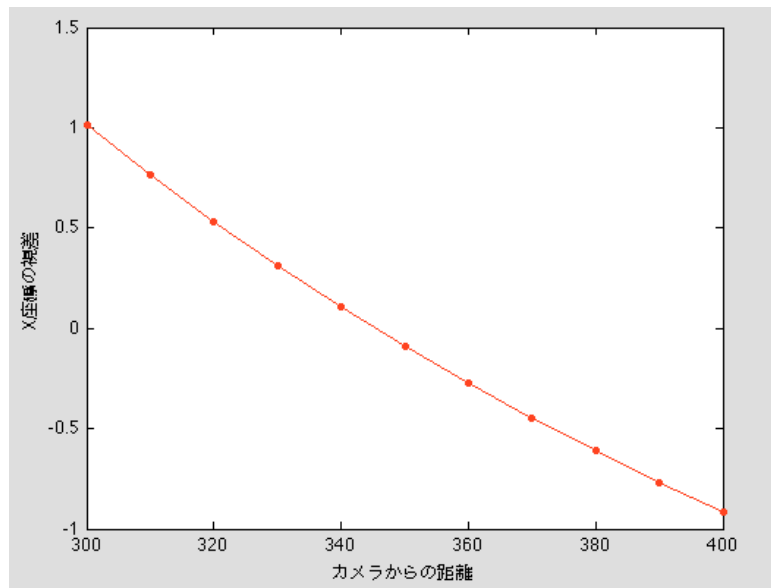


図 4.6 30～40cm の視差-距離グラフ

3 章で行った実験結果とグラフを比較してみると図 4.5 のグラフの形状はほぼ同じになった。図 4.6 より視差が 0 になる距離は 34.5cm で実測値の 20cm との差が 14.5cm となった。そして、視差は-6.7 画素に漸近することがわかり、実測で得られた結果は-6.11 画素であったのでこちらは近い値が得られた。

また、右カメラの光軸上の点を、左カメラで観察した位置をプロットすると図 4.7 を得た。

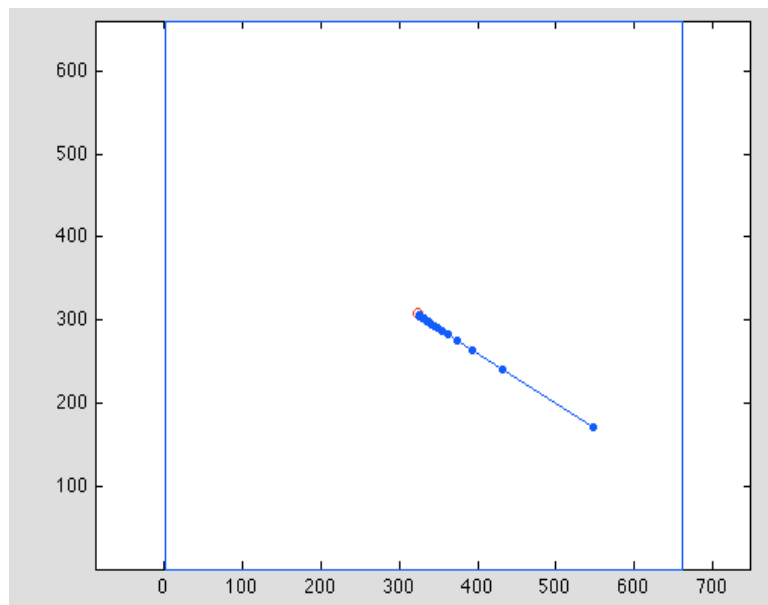


図 4.7 右カメラ光軸上の点を左カメラで観察した位置の変化

図 4.7 は 1cm~25cm での中心のずれを表したグラフである(無限遠から、ある距離までのグラフ)。外の正方形の枠は画像の領域を現している。 x 方向への視差があることが確認できたのに加え、 y 軸方向への視差も出るという結果が得られた。これは、キャリブレーションで得た外部パラメータの中で、回転行列の誤差が原因であると考えた。

図 4.7 の結果を踏まえた上、次に 2 台のカメラの回転行列を右カメラの回転行列にそろえて画像中心のずれを出力した。その結果を以下に示す。

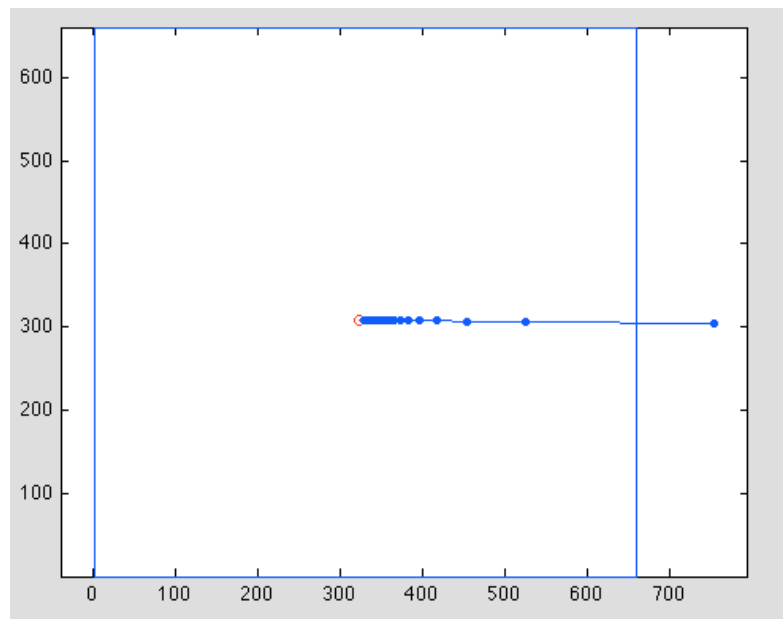


図 4.8 右カメラ光軸上の点を左カメラで観察した位置の変化

この結果より、カメラキャリブレーションによって求められた、本来ならば平行ステレオであるはずの 2 台のカメラの回転量が若干異なっていることによって y 軸方向へのずれが生じるということがわかった。

5. 結論

第3章で行った実験の実距離測定は、Lytro での実距離測定は 200cm よりも遠い位置にある被写体を高精度に推定することは困難であった。今回行った計測方法は適さず、さらに高精度に Lytro、被写体を設置し撮影を行う必要があった。1cm～25cm での距離推定は比較的良好な結果が得られた。この距離では撮影による誤差が視差への影響が少ないと考えられる。平行ステレオは無限遠の被写体に対して視差が 0 になるが、Lytro では 21～22cm で視差 0 になった。これは 21cm～22cm で視差が 0 になるように意図的に平行移動されているということである。つまり、Lytro は 21cm～22cm にある被写体が最も撮影しやすいように設定されている。また視差は無限遠になるにしたがって -6.11 画素に収束した。また、測定した距離・視差のデータ 40 組を利用して最小二乗法で近似式を求めて、距離推定を行ったが 60cm 以降のデータが少なく、60cm 以降では正しく推定できないという結果になった。

4 章の実験で、Lytro でキャリブレーションを行ったときに得られる外部パラメータの回転量を表す回転行列が、仮想的に考えられる左右のカメラでは僅かながら異なっており、その結果、計算上 y 軸方向への視差が生じてしまった。この原因は回転行列が 2 台のカメラ間で異なっていたことであり、一方のカメラに回転行列をそろえたら y 軸方向のずれは生じなかった。つまり、カメラキャリブレーションの回転行列には大きな誤差が含まれるので検証には至らず、そのような傾向があることが確認できる程度であった。

Lytro を用いての実距離測定は新たなアプリケーションを付加するという側面をもっており、そのさらなる発展への礎となったと考えられる。また、本研究の実距離測定の実験における改善すべき点として、被写体と Lytro の距離は Lytro の先端から被写体までの距離で計測していたが、この距離を被写体と Lytro のカメラ中心で測ればさらに精度の良い結果が得られたかもしれない。それに付け加え、無限遠での視差が -6.11 画素に漸近したことも加味し、数式に変更を加え実距離を測定すれば、別の結果が得られるはずである。

謝辞

本研究を進めるにあたり、担当教授の蚊野先生に感謝致します。昼夜を問わず私の質問に対して親身になり考えて下さったり、熱の入ったご指導本当にありがとうございました。また、学生生活の中で私とかかわりをもってくださった皆様に感謝します。

参考文献

- [1] Lytro 社の Web サイト, <https://www.lytro.com/>
- [2] Adelson, E. H., and Wang J. Y. A., **Single Lens Stereo with a Plenoptic Camera**, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2), pp.99-106, 1992.
- [3] Marc Levoy の Web サイト, <http://graphics.stanford.edu/~levoy/>
- [4] 蚊野浩, 「あらゆるところに賢いカメラ」, 日経エレクトロニクス, 8 月 20 日号, pp.42-47, 2012 年度.
- [5] 蚊野浩, 「デジカメの画像処理」, オーム社, 画像情報メディア学会, 2011 年.
- [6] デジタル画像処理編集委員会, 「デジタル画像処理」, CG-ARTS 協会, 2004 年.

付録

本研究を進めるにあたり、以下の **MatLab** プログラムを開発した。

LytroRawRead2.m

Lytro の .lfp ファイルからその生画像を tif ファイルに変換するプログラム。

MakeStereo2.m

生画像からマイクロレンズの任意の位置におけるステレオ画像を生成するプログラム。

Binarization.m

ステレオ画像における被写体部分を白色にした二値画像を生成するプログラム。

C_of_G.m

二値画像の被写体部分の x 軸方向の重心を求めるプログラム。

rodrigues.m

キャリブレーションで得たカメラの回転量を回転行列に変換しカメラ間の傾きを度数法で求めるプログラム。

ichijihenkan.m

内部パラメータ・外部パラメータを用いて x 軸方向・ y 軸方向の(視差-距離)グラフをと右カメラ光軸上の点を左カメラで観察した位置の変化をプロットするプログラム。

OpenCV において

CvCalibrateCamera_NEW.cpp

複数毎のチェスボードパターンを入力画像としてカメラキャリブレーションを行い、内部パラメータ・外部パラメータを出力するプログラム。