

直線検出による像倍率変化の抑制についての一考察

北野 和彦¹ 小林 亜樹¹

概要 : DFF 法において, 合焦判定に使用する入力画像であるマルチフォーカス画像は, フォーカス位置変化に伴う像倍率の変化がない(少ない)ことが理想である. しかし, 広く普及しているカメラのレンズの多くは, 像倍率変化が生じる機構である. 本稿で筆者らは, マルチフォーカス画像の像倍率変化の抑制を目指した. マルチフォーカス画像のエッジ画像列において, 中心から周辺へ向かう半直線上のフォーカス位置変化によるエッジの変化画像を用いて, 直線検出を行った傾きを像倍率変化率として推定する手法を提案する. 変化率を基にマルチフォーカス画像のエッジ画像列に対してフォーカス位置毎にスケールパラメータを決めて像倍率変化を抑制する. 像倍率変化の抑制前と後のエッジ画像列から DFF 法を適用し, 評価する.

キーワード : Android, Hough 変換, フォーカス位置, 距離画像

KITANO KAZUHIKO¹ KOBAYASHI AKI¹

1. はじめに

物体までの距離を推定する手法の 1 つに DFF (Depth From Focus) 法がある. これは異なるフォーカス位置で撮影された画像列に対して合焦判定を行い, 合焦と判定したフォーカス位置を用いて奥行推定を行う [3] 手法である. DFF 法に関する多くの研究は, テレセントリック光学レンズ [4] を用いている. これは, DFF 法がフォーカス位置変化に対する像倍率が変化しない(少ない)ことが理想だからである.

しかしながら, 広く一般的に普及しているカメラのレンズの多くは, 像倍率変化が生じる機構であるため, テレセントリック光学レンズのようにフォーカス位置変化による像倍率変化が少ないレンズの使用が好まれていた. そのため, スマートフォンのカメラを用いて撮影された画像であっても遠近の違いが判別できる程度の距離値推定は可能であるが像倍率変化による影響により一部の画素において誤った距離が推定される [7].

一方, フォーカス位置変化による像倍率変化を SURF により特徴点抽出を行い, 画素間の対応付けから補償する手法 [6] や位相関数によって推定された画像特徴のシフトに基づいて像倍率変化を補償する手法 [5] などの像倍率変化

に対応した研究もなされている.

本稿では, 撮影物体上のエッジの変化を直線近似できるものとみなすことで, エッジ位置のフォーカス位置変化に伴う変化を表現するエッジの変化画像を生成し, この上で Hough 変換 [1][2] による直線検出を行い, そのパラメータから像倍率変化率を推定する手法を提案する. この推定変化率を用いて撮影画像列をスケールすることで, 像倍率変化の影響を抑制した DFF 法による奥行き推定を行う.

2. 像倍率変化率推定手法

2.1 用語

- フォーカス位置
Android OS 搭載の端末を制御する API の 1 つであるカメラのフォーカス制御を行うパラメータであり, 本稿ではフォーカス位置 d と表記する. フォーカス位置 $d \in [0.0, 10.0]$ である. このとき $d = 0.0$ は, 無限遠に相当し, $d = 10.0$ は最短撮影距離に相当する.
- マルチフォーカス画像
同一シーンについて, 適当なステップ幅を以て異なるフォーカス位置毎に撮影した画像列である. これら画像のことを撮影画像とする.
- エッジ強度
撮影画像について, 適当なエッジ抽出オペレータにて抽出した画素毎の値を表す. この値で再構成した画像

¹ 工学院大学
Kogakuin University, Japan

をエッジ画像とする。

- 距離値

マルチフォーカス画像より DFF 法によって推定した画素毎の奥行き距離である。この値で再構成した画像を距離画像とする。

2.2 概要

DFF 法により距離値を推定する場合、像倍率変化を伴うマルチフォーカス画像をそのまま用いたのでは、精度低下につながる。

そこで、本稿では像倍率変化を補償するために像倍率変化率を推定する手法を提案する。マルチフォーカス画像の撮影は、Android OS 搭載でフォーカス位置変化が可能であるスマートフォンのカメラを用いる。

マルチフォーカス画像の撮影から距離画像作成までの全体の処理手順を図 1 に示す。

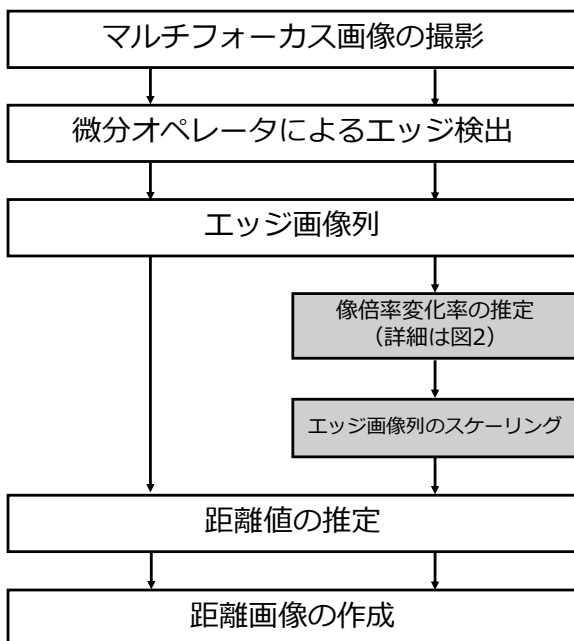


図 1 全体の手法の流れ

あるステップ幅で分割したフォーカス位置 d でマルチフォーカス画像の撮影を行う。その各画像に対してエッジ抽出を行い、エッジ画像列を生成する。これらのエッジ画像から像倍率変化率の推定を行う。像倍率変化率をもとに各フォーカス位置 d の画像に対してスケーリングを行い、既存手法から距離値を推定する。

前回の報告では、マルチフォーカス画像をエッジ抽出を経てエッジ画像列を生成し、距離値の推定を行った。そのため、像倍率変化の影響は考慮していない。

そこで、本稿では距離値を推定する前に像倍率変化率を推定し、エッジ画像列に対してスケーリングを行う。像倍率変化率推定の処理手順は図 2 の通りである。

像倍率変化によるエッジ位置の変化を測定し、エッジの

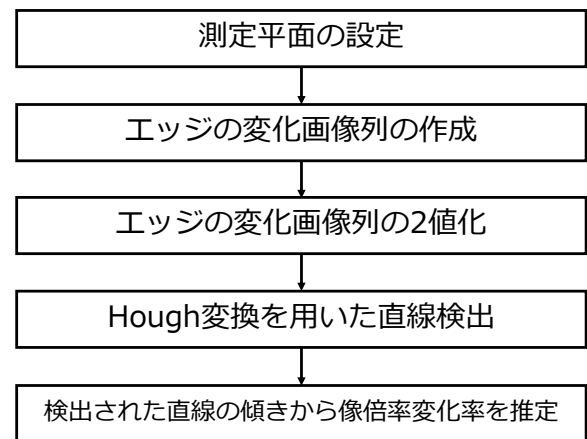


図 2 像倍率変化率推定手法の流れ

変化画像を作成する。像倍率が変化することによる画角の変化が生じるとフォーカス位置の異なるエッジ画像を比較したとき、そのエッジ画像を比較するとエッジ領域も異なる。本研究では、このエッジ領域の変化がエッジの変化画像において直線上に分布すると仮定している。そのため、エッジの変化画像に Hough 変換を行い、直線として検出し、その直線の傾きを用いて像倍率変化率を推定する。

2.3 マルチフォーカス画像の撮影

スマートフォンから 2 点それぞれ異なる位置に物体が置かれていることを想定する。ここで、フォーカス位置 $d = [0.0, 10.0]$ までを N 段階に分割して同一シーンで N 枚撮影する。この画像列をマルチフォーカス画像と呼ぶ。マルチフォーカス画像は、 x 軸、 y 軸に加えてフォーカス位置 d 軸の 3 次元の画像で表現される (図 3)。このとき、各フォーカス位置 d で撮影画像の位置 (x, y) は画素値 $i(x, y, d)$ である。

2.4 エッジ画像列

マルチフォーカス画像の画素値 $i(x, y, d)$ を用いて、エッジ抽出オペレータ $\mathcal{E}(\mathbf{I})$ によりエッジ抽出したエッジ画像列 $E(x, y, d)$ を生成する ((1) 式)。

$$E(x, y, d) = \mathcal{E}(i(x, y, d)) \quad (1)$$

2.5 エッジの変化を測定する平面の設定及び作成

フォーカス位置変化におけるエッジの変化画像を作成するにあたって、エッジの変化を測定する平面を設定する。 xy 平面上で画面中心から周辺方向へ向かう線分上 (図 3) であり、すべてのフォーカス位置において同様の画素とする。したがって、測定領域はこの線分を d 方向へ拡大した平面上が測定平面である。線分長は、エッジ画像の短辺方向の半分とすることで、線分方向には特段の制限を設けない。この平面上に展開されたエッジ強度分布をエッジの変化画像と呼ぶ。

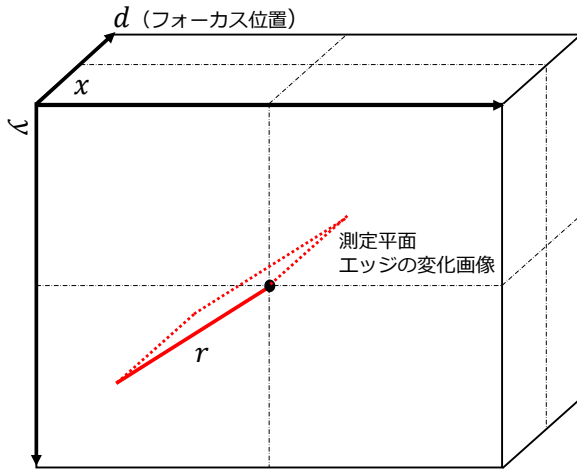


図 3 画像列の 3 次元表現とエッジ変化画像平面

図 4 のようにエッジ画像の中心を原点とみなして xy 平面を象限に分割すると、図 3 の例は、第 3 象限のエッジの変化画像と呼ぶ。これらのエッジの変化画像は画像中心から周辺へ向かう方向を r 軸と呼ぶ。また、エッジの変化画像を $Ec(d, r)$ で表現する。

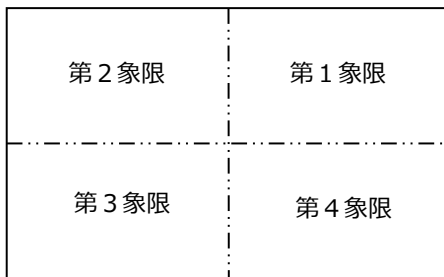


図 4 象限分割

2.6 エッジの変化画像に対する Hough 変換

エッジ変化画像 $Ec(d, r)$ は、エッジ画像の画面中心から周辺へ向かう直線上のエッジ強度分布を縦方向に、フォーカス位置 d の変化に従って横方向に並べた画像で、ここでは、その後の抽出処理での精度を考慮してアスペクト比をほぼ 1:1 になるように拡大率 α を用いてスケールした画像である (図 5)。

Hough 変換に必要な入力画像が 2 値画像である必要がある。エッジの変化画像の画素値は、エッジ強度自体であるため多値画像である。そこで、適当な閾値で 2 値化した 2 値化エッジ変化画像 $B(d, r)$ へと変換する ((2) 式)。

$$B(d, r) = \begin{cases} 0 & Ec(d, r) < t_h \\ 1 & Ec(d, r) \geq t_h \end{cases} \quad (2)$$

Hough 変換は画像の原点から検出された直線に向かう法線の長さ ρ とその法線の角度 θ が得られる。

したがって、2 値化エッジ変化画像上での直線の傾きは、 $1/\tan(\theta)$ である。ここで、計算精度上の問題のために拡大

した d 方向の拡大率 α を考慮すると、推定すべき直線のフォーカス位置 d の変化に対する変化率 $\Delta f(d)$ は、2 値化エッジ変化画像を生成する際の d 方向の拡大率 α を用いて、

$$\Delta f(d) = \frac{\alpha}{\tan(\theta)} \quad (3)$$

と示せる。

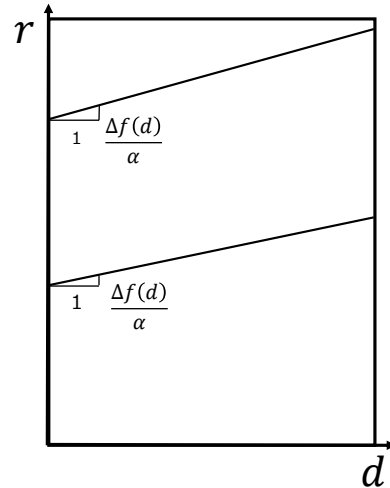


図 5 エッジ変化画像の模式図

本手法では撮影画像のフォーカス位置 d の変化による像倍率変化率は、フォーカス位置 d によらず一定で、画面内の位置に対して不変で、画像全体の単純な拡大縮小であると仮定している。

本手法での像倍率変化率に対する仮定の下では、任意のフォーカス位置 d における撮影画像のある座標に対応する他のフォーカス位置 d における撮影画像の画素位置の変化は、画面の中央で最小値 0 をとり、画面周辺に向かうにつれて単調増加となる。これは、エッジの変化画像上での (ほぼ) 直線状の高エッジ強度点列の直線の傾きが画像上部ほど大きい。すなわち、撮影画像の周辺部ほどフォーカス位置変化に対する、像に対応する画素位置の変化も大きくなる関係として現れる。そこで、この高エッジ強度点列が近似直線で表せるとするとき、その直線の傾きである変化率 $\Delta f(d)$ は画面中心からの距離 r に比例するものと考えられるため、

$$\Delta f(d)_N = \frac{\Delta f(d)}{r} \quad (4)$$

と (4) 式に示すように正規化した $\Delta f(d)_N$ を用いて像倍率変化率を推定することとする。

本手法には 2 値化処理に使用する閾値と ρ - θ パラメータ空間において直線とみなす投票数の閾値が必要である。そのため、本稿では 2 値化閾値 th_b の集合と投票数閾値 th_v の集合を用意し、2 つの集合の直積集合の各要素を用いてエッジの変化画像に対して Hough 変換を行う。

2.7 像倍率変化率の推定

Hough 変換による直線検出では、閾値によって検出される直線数が変化し、一般に複数の直線が検出される。そこで、その直線パラメータを正規化した $\Delta f(d)_N$ の列のうち、最大最小の2本を除いたデータによるトリム平均を推定値とする。ただし、検出できた直線数が4本未満の場合は、最大最小値も含めた算術平均とする ((5) 式)。また、検出された直線の本数が0本の場合は、計算に含まないこととする。

$$\overline{\Delta f(d)_N} = \begin{cases} \frac{\sum_{i=1}^n \frac{f(d)_{N_i}}{n}}{n} & n < 4 \\ \frac{\sum_{i=1}^{n-2} \frac{f(d)_{N_i}}{n-2}}{n-2} & n \geq 4 \end{cases} \quad (5)$$

投票数閾値 th_v の各集合の直積集合を用いることで、要素数に応じた変化率 $\overline{\Delta f(y)_N}$ が算出される。この算術平均をとることで、1つの測定平面での変化率が算出される。

また、設定した測定平面がパラメータとして与えられるとパラメータの数に応じた測定平面での変化率が算出される。それらの変化率を算術平均をとることで像倍率変化率 R が推定される。

2.8 距離値推定

スケーリング後のエッジ画像列に対しての距離値推定は、エッジ強度比較手法 [7] と同様である。マルチフォーカス画像のエッジ画像列を入力とする。このエッジ画像列 $E(x, y, d)$ 、座標 (x, y) における距離値 $s(x, y)$ として (6) 式より算出する。

$$s(x, y) = \min_d \{ \arg \max E(x, y, d) \} \quad (6)$$

これは、エッジ画像列の各座標において、エッジ強度の比較を行う。このとき、エッジ強度が最も高いフォーカス位置 d を合焦フォーカス位置であると判断し、距離値としている。また、この手法ではエッジ強度が最も高いフォーカス位置 d が同 xy 座標において複数フォーカス位置で観測される可能性がある。そのような場合は、最も無限遠寄りの値を採用する。

3. 実験

3.1 目的

スマートフォン実機を用いて撮影したマルチフォーカス画像から、提案手法を用いてどの程度の距離画像を生成できるかを試みた。提案手法による像倍率変化の影響を抑制する効果がどの程度であるかを推定された距離画像を比較することで確認することを目的とする。

3.2 条件

エッジの変化を測定する領域は、画面 xy 平面の中心から4隅に向かう対角線上の線分によって規定される平面とした。Android OS を搭載したスマートフォンを用いてマルチフォーカス画像の撮影を行う。使用した端末は Nexus 5X であり、詳細を表1に示す。

表 1 撮影に使用した端末

Device	Nexus 5X
OS	Andorid 7.0
CPU 1.8GHz	Qualcomm Snapdragon 808 (MSM8992) + 1.4GHz ヘキサコア
Memory	2GB
画素数	約 1230 万画素
ピクセルサイズ	1.55[μm]
絞り	f/2.0

Nexus 5X は、Google 社と LG エレクトロニクス社が共同開発したスマートフォンである。スマートフォンではマルチフォーカス画像の撮影のみを行い、画像処理は PC で行う。処理に使用した PC の詳細を表2に示す。画像処理は python 上で OpenCV を使用した。

表 2 画像処理に使用した PC

OS	ubuntu 14.04 LTS
CPU	Intel Core i3-6100 3.70GHz
Memory	32GB

3.3 撮影アプリケーション

マルチフォーカス画像の撮影は、自作アプリケーションを使用した。このアプリケーションは次の機能をもつ。

- マニュアルフォーカスモード
- 任意のシャッター間隔による自動撮影

このアプリケーションは、API Level*121 以上が必須である。すなわち、Android 5.0 (Lollipop) 以上である必要があり、Android 4.4 (KitKat) 以下のバージョンでは動作できない。また、端末自体がマニュアルフォーカスに対応している必要がある。Nexus 5X はいずれの条件も満たしているため、マニュアルフォーカスが可能である端末である。

3.4 撮影環境

撮影は、白と黒の縞模様が印刷された紙を2つ用意し、図6のような位置に置く。スマートフォンは固定して撮影を行う。このとき、カメラからの距離を手前の紙 α から約 15[cm]、奥側の紙 β から約 30[cm] とした。

*1 Android プラットフォームのバージョンごとに提供されるフレームワーク API のリビジョンを一意に識別する整数値

フォーカス位置 d を, $d = [0.0, 10.0]$ (無限遠から最短撮影距離) である. フォーカス位置 d のステップ幅は 0.1 とし 101 枚撮影した.

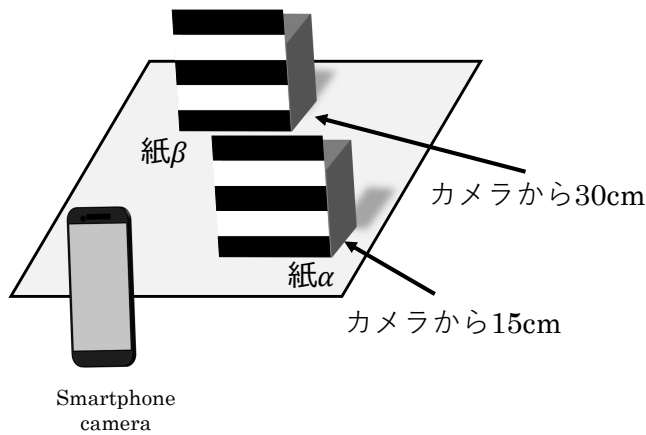


図 6 撮影環境

3.5 エッジ変化測定領域の設定

フォーカス位置 $d = 0.0$ (無限遠) における撮影画像を図 7 に, その画像を Sobel オペレータによりエッジ抽出したエッジ画像を図 8 に示す. ただし, 紙面上では図を見やすくするために画像処理をしてある. 以下のエッジ画像やエッジ画像から作成したエッジの変化画像などのすべてが同様である.



図 7 撮影画像 ($d = 0.0$)

このエッジ画像上にエッジの測定領域の xy 平面内の位置を示した画像を図 9 に示す. この画像の中心から 4 隅に向かう線分上に d 方向に拡大した測定平面でエッジの変化を測定して 4 枚のエッジの変化画像を作成する. また, 図 9 中にある数字は識別を行うために用いた各象限の数値であり, その象限で測定平面を区別することとする.

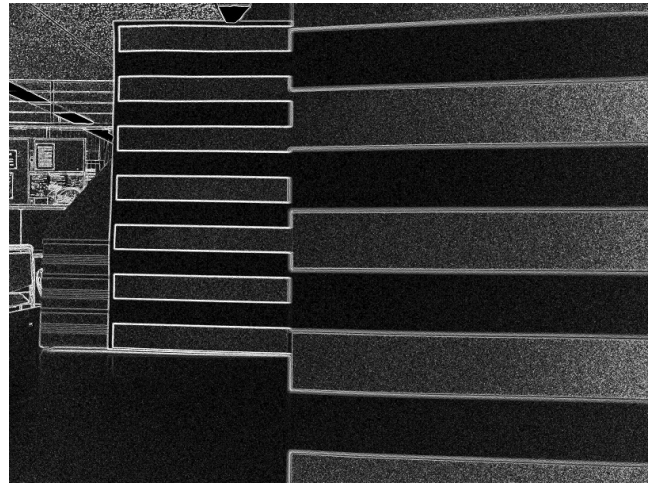


図 8 エッジ画像 ($d = 0.0$)

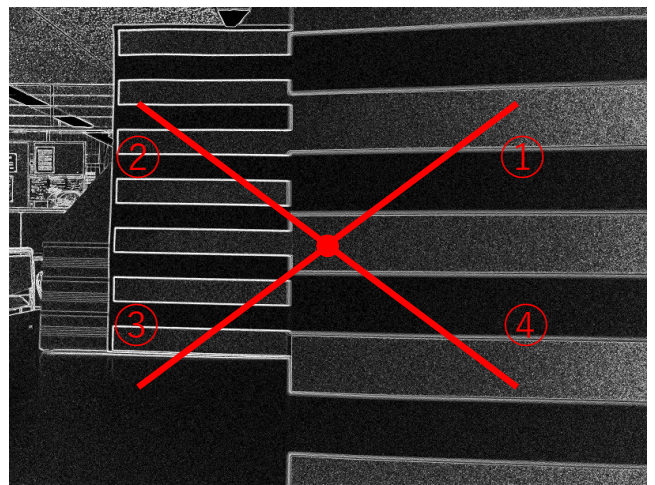


図 9 エッジ画像上の測定領域

3.6 エッジの変化画像

各象限に対応する線分による測定平面でエッジを測定することで作成したエッジの変化画像をそれぞれ図 10, 11, 12, 13 に示す.

図 9 において, 第 1 象限および第 4 象限に向かう測定平面では, 高いエッジ強度を横切るのは 2 か所である. 第 1 象限および第 2 象限を示す. 図 10, 13 のエッジの変化画像を見ると, 2 種類の画像それぞれにエッジの推移が 2 か所観測できる. また, 第 2 象限では 6 か所, 第 3 象限では 5 か所あり, それぞれの象限に向かう線分による測定平面に対応した図 11, 12 のエッジの変化画像に対して同数のエッジの推移が確認できる.

3.7 エッジ画像列に対する Hough 変換

Hough 変換を行うときは, 2 値画像が必要である. ここで図 14 のように 2 値化閾値が低いと, エッジ強度の弱い部分まで拾ったエッジの変化画像が得られ, 直線の領域のエッジ強度の弱い部分で広がったような形状になっている. この領域内では多数の直線が検出される. これにより, 多

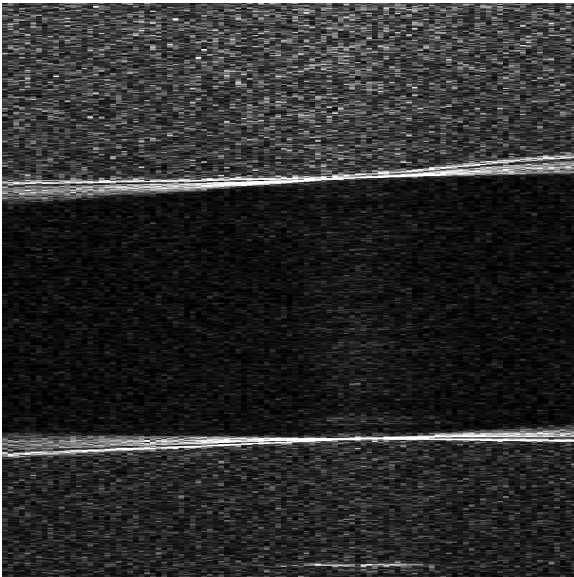


図 10 エッジの変化画像 (第 1 象限)

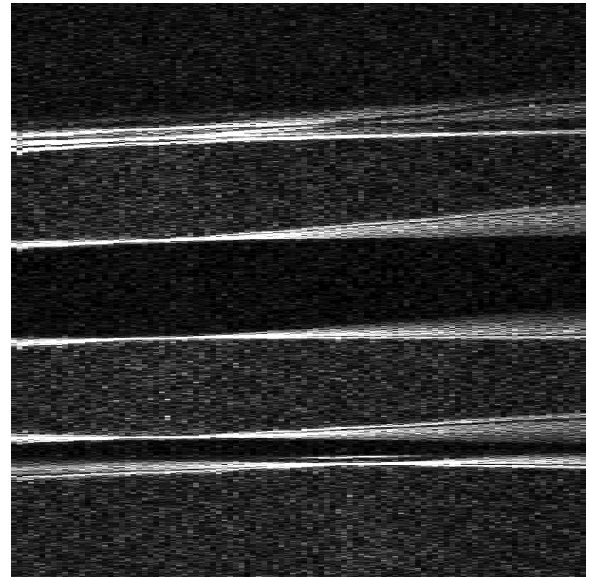


図 12 エッジの変化画像 (第 3 象限)

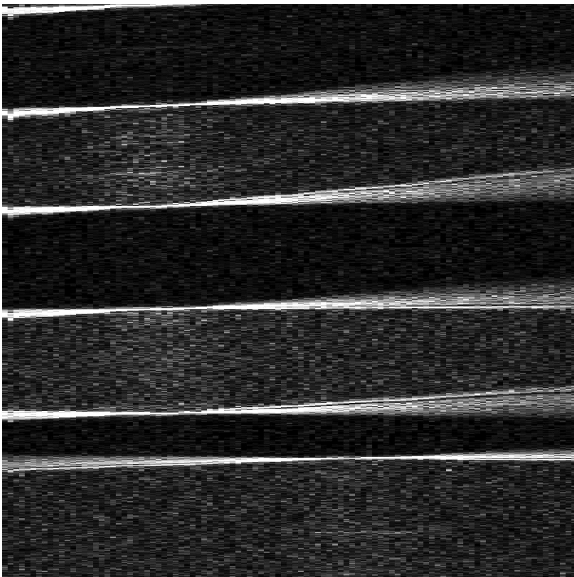


図 11 エッジの変化画像 (第 2 象限)

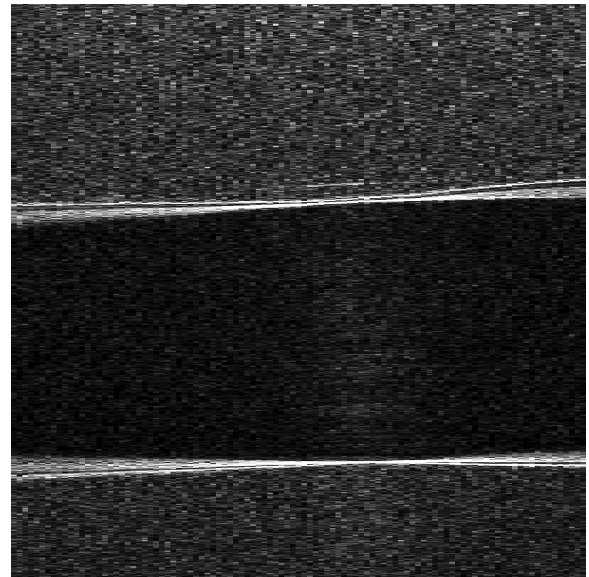


図 13 エッジの変化画像 (第 4 象限)

数の傾きが混ざることによって推定精度の低下が考えられる。

一方、図 15 のように 2 値化閾値を高く設定するとエッジの強い領域だけがしか残らず、低投票数閾値でなければ、直線を検出が困難になる。また、投票数閾値が低くなることでノイズによる影響を受けやすく誤った直線が検出され推定精度の低下が考えられる。

そこで本実験において、経験的に 2 値化閾値 $th_b = \{80, 90, 100, 110, 120\}$ 、また Hough 変換に用いる投票数閾値 $th_v = \{250, 350, 450, 550\}$ とした。

Hough 変換によって得られる直線の傾きの正負は一般的にどちらかである。しかしながら、二値化閾値によって直線の領域のエッジ強度の弱い部分で広がったような形状になることや、ノイズ等の影響により直線の傾きの正負が混

在することが想定される。傾きの正負が混在したときは、傾きが正である集合と負である集合に分類して正の集合と負の集合の要素数を比較する。その結果要素数が少ない傾きの集合は破棄する。

4. 結果と考察

各象限方向の測定平面および 2 つの各閾値の集合の要素によって算出した変化率を表 3, 4, 5, 6 に示す。

各象限方向の測定平面および二値化閾値 th_b と投票数閾値 th_v の直積集合における変化率 $\overline{\Delta f(y)}_N$ の算術平均した値が各表に示す平均である。これら 4 つの値をさらに算術平均をとることで像倍率変化率 R を算出し、 $R = 5.00 \times 10^{-4}$ であった。一方、目視でもっとも適当な像倍率変化率を求めた結果 R_T は 5.36×10^{-4} であった。

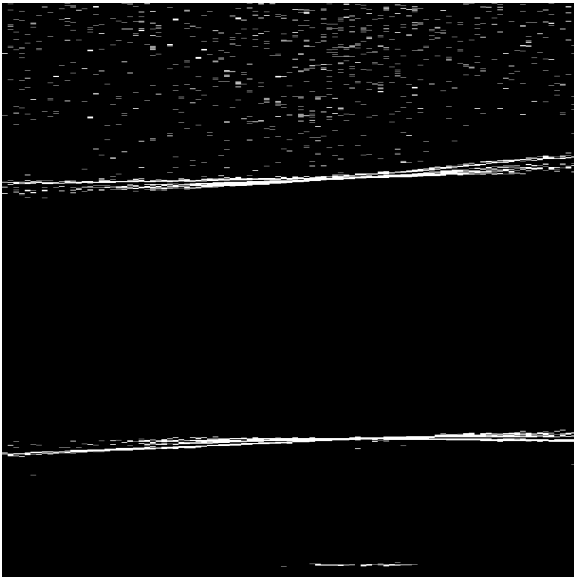


図 14 低 2 値化閾値による 2 値化

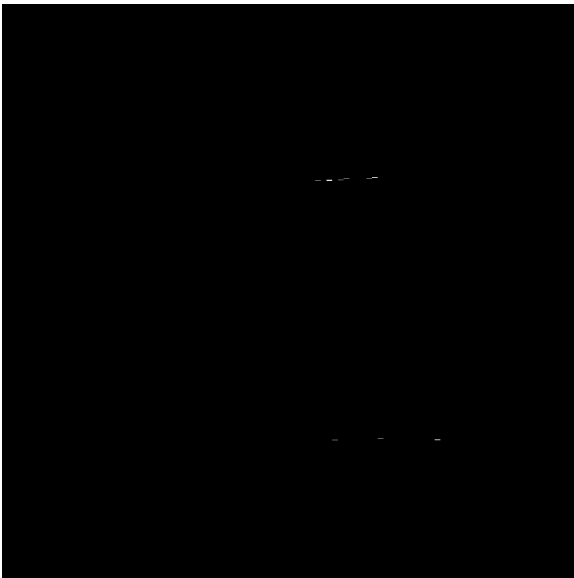


図 15 高 2 値化閾値による 2 値化

表 3 第 1 象限方向の測定平面における変化率 $\overline{\Delta f(y)}_N$

閾値 th_b	閾値 th_v	検出本数 n	変化率 $\overline{\Delta f(y)}_N \times 10^{-4}$
120	250	1	5.31
110	250	3	7.34
100	350	1	5.29
100	250	4	5.42
90	350	2	5.42
90	250	5	7.06
80	350	4	5.42
80	250	5	6.99
平均			6.03

スケーリング前のマルチフォーカス画像のエッジ画像列で作成した距離画像を図 16, スケーリング後のマルチ

表 4 第 2 象限方向の測定平面における変化率 $\overline{\Delta f(y)}_N$

閾値 th_b	閾値 th_v	検出本数 n	変化率 $\overline{\Delta f(y)}_N \times 10^{-4}$
120	350	5	4.24
120	250	10	4.04
110	350	6	4.51
110	250	12	3.92
100	450	2	4.83
100	350	7	4.21
100	250	12	4.36
90	450	3	4.50
90	350	8	3.95
90	250	15	5.30
80	550	1	5.34
80	450	6	4.51
80	350	13	4.93
80	250	19	5.55
平均			4.59

表 5 第 3 象限方向の測定平面における変化率 $\overline{\Delta f(y)}_N$

閾値 th_b	閾値 th_v	検出本数 n	変化率 $\overline{\Delta f(y)}_N \times 10^{-4}$
120	450	1	3.28
120	350	7	4.17
120	250	10	4.59
110	550	1	3.28
110	450	1	3.28
110	350	7	4.17
110	250	11	4.42
100	550	1	3.28
100	450	3	3.50
100	350	9	4.54
100	250	11	4.44
90	550	1	3.28
90	450	6	4.39
90	350	11	4.59
90	250	16	5.45
80	550	3	4.51
80	450	9	4.10
80	350	12	3.95
80	250	18	5.06
平均			4.12

フォーカス画像のエッジ画像列で作成した距離画像を図 17 に示す. スケーリング適応前の距離画像では, 大まかに手前の紙, 奥の紙, その左側の 3 つの領域に対して, 実際の遠近に相当する位置関係で距離値が推定されている. しかしながら, 実際には近い位置にあるはずの手前の紙の領域

表 6 第 4 象限方向の測定平面における変化率 $\overline{\Delta f(y)}_N$

閾値 th_b	閾値 th_v	検出本数 n	変化率 $\overline{\Delta f(y)}_N \times 10^{-4}$
120	250	3	5.39
110	250	3	5.39
100	350	2	5.14
100	250	3	5.39
90	350	2	5.14
90	250	3	5.39
80	450	2	5.14
80	350	2	5.14
80	250	5	5.41
平均			5.28

に無限遠側を示す距離値（青色）に相当する値が推定されている。また奥の紙には、最短撮影距離側の距離値（橙色）が推定されている。縮小処理後では、3つの領域に対して実際の遠近とは位置関係がことなる距離値が推定されている割合が低減していることが確認できる。以上の理由からエッジの変化画像に対して Hough 変換から像倍率変化率の推定において有用性を示せた。

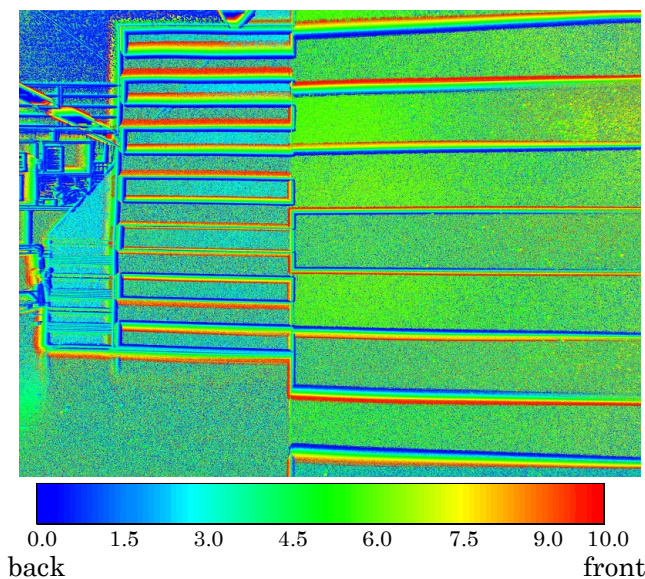


図 16 距離画像（スケール前）

5. おわりに

本稿では、Android OS 搭載のスマートフォンのカメラを用いてマルチフォーカス画像を撮影し、エッジ画像列によるエッジの変化画像を作成し、Hough 変換を行うことで像倍率変化率を推定した。

本手法では撮影画像のフォーカス位置の変化による像倍率変化率は、フォーカス位置によらず一定で、画面内の位置に対して不変で、単純な拡大縮小であると仮定して、像倍率変化率から、フォーカス位置毎にエッジ画像列に対し

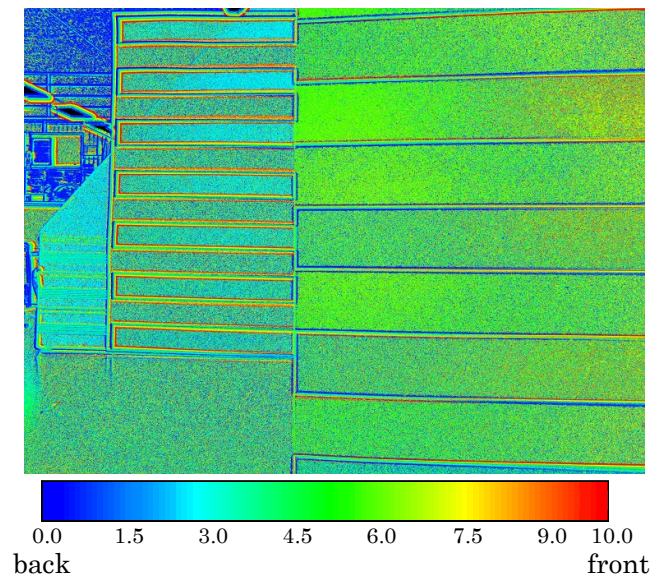


図 17 距離画像（スケール後）

てスケールを行い、距離画像を作成した。スケール前の距離画像と比較し3つの領域に対して実際の遠近とは位置関係が異なる距離値が推定されている割合が低減していることが確認できた。そのため、エッジの変化画像に対して Hough 変換から像倍率変化率に推定において有用性を示せた。

今後は、2 値化閾値および投票数閾値などのパラメータを自動で設定できるアルゴリズムの提案及び実装を行い、定量的な評価を目指す。

参考文献

- [1] Hough, P. V. C. , “Method and means for recognizing complex patterns,” U. S. Patent No. 3069654, 1962.
- [2] Ballard, Dana H, “Generalizing the Hough transform to detect arbitrary shapes,” Pattern recognition, Vol. 13, No. 2, pp.111–122, 1981.
- [3] T. Darrell and K. Wohn, “Pyramid based depth from focus,” Computer Vision and Pattern Recognition, 1988, Proceedings CVPR '88, Computer Society Conference, pp. 504–509, 1988.
- [4] Watanabe, Masahiro and Nayar, Shree K, “Telecentric optics for computational vision,” European Conference on Computer Vision, pp. 439–451, 1996.
- [5] S. Pertuz, D. Puig, and M. A. Garcia, “Improving Shape-from-Focus by Compensating for Image Magnification Shift,” 2010 20th International Conference on Pattern Recognition, pp. 802–805, 2010.
- [6] R. Senthilnathan, P. Subhasree, R. Sivaramakrishnan, and P. Karthikeyan, “Estimation of sparse depth based on an inspiration from SFF,” 2016 International Conference on Recent Trends in Information Technology (ICRITIT), pp. 1–6, 2016.
- [7] Kitano, Kazuhiko and Kobayashi, Aki, “Implementation of DFF Using a Smartphone Camera,” Consumer Electronics (GCCE), 2017 IEEE 6th Global Conference on, pp. 532–534, 2017.