Vol.2017-UBI-54 No.23
Vol.2017-CDS-19 No.23
2017/5/26

スマートフォンのカメラを用いたDFF法の実装

北野 和彦[1]　小林 亜樹[1]

概要：距離を推定する手法に DFF(Depth From Focus) 法が知られている．しかし，スマートフォンのカメラによって撮影された複数画像による DFF 法の検討はあまり進められていない．本稿で筆者らは Android OS 搭載のスマートフォンのカメラから撮影した複数画像を用いた DFF 法による距離推定プログラムを作成した．実機を用いて撮影した複数画像から距離を推定し，距離画像を作成した．距離画像を作成するにあたって，フォーカス変位による画角変化，撮影時に要した時間及び撮影時のスマートフォンの熱等の問題について議論する．

キーワード：Android，Camera2 API，フォーカス位置，ぼけ，距離画像

KITANO Kazuhiko[1]　KOBAYASHI Aki[1]

1. はじめに

同一場面の撮影では，フォーカス位置の変位によってぼけのない画像とぼけが含まれる画像を得ることができる．この原理を応用して，フォーカス位置を順々に変位させながら撮影して得られた画像群（多重フォーカス画像) に対して合焦しているか否かの評価を行い，合焦するフォーカス位置を用いてカメラのレンズから物体までの距離や奥行の情報を求めるものは DFF(Depth From Focus) 法 [1][2][3][4][5] と呼ばれ，研究が進められてきた．また，DFF 法は，撮影する枚数が多いため，フォーカス位置が異なる数枚の画像とあらかじめ作成したぼけモデルから距離を求める DFD（Depth From DeFocus）法 [1][6][7][8] も提案されている．

しかし，コンシューマデバイスであるスマートフォンのカメラによって撮影された画像群による DFF 法や DFD 法の検討はあまり進められていない．広く普及したスマートフォンに搭載されたカメラを用いて奥行き情報を取得できれば応用機会は広い．固定的な位置からの撮影で取得可能な方法での実用化が望まれる中で研究が活発とは言えない状況は，端末の種類が豊富で様々な機器が存在するだけでなく，光学系などのパラメータも非公開であることで実現が難しいからであると考えられる．特に光学系の詳細なぼけに関するパラメータを要する DFD 法の実装は困難であることから DFF 法を採用することにした．

そこで本稿では，Android OS のスマートフォン上で DFF 法の原理に基づく実装を行い，基本的な特性の分析を行ったため報告する．方式の実現可能性，撮影時間，距離推定精度，その他の機器に及ぼす影響等についてまとめる．

2. Android OS 上での DFF 法

2.1 定義

- フォーカス位置
  Android OS 搭載端末を制御する API の 1 つである android.hardware.camera[*1]（以降 Camera2 API と記述する) に含まれるカメラを制御するパラメータである．本稿では，フォーカス位置を d と記述し，d の値域は [0.0,10.0] である．これは，端末毎に異なる実距離対応を持つ相対距離である．制御に用いるパラメータは，浮動小数点であるため $d = [0.0f, 10.0f]$ と記述する．ここで，$d = 0.0f$ は無限遠相当であり，$d = 10.0f$ は最近接撮影距離相当を示す相対距離である．
- エッジ強度
  微分フィルタによって得られた隣接画素との差分から計算される勾配の大きさとする．
- 距離値
  フォーカス位置 d の変位により得られる多重フォーカス画像から画素毎に合焦判定を行い，合焦している

[1] 工学院大学
Kogakuin University

[*1] https://developer.android.com/reference/android/hardware/camera2/package-summary.html

Vol.2017-UBI-54 No.23
Vol.2017-CDS-19 No.23
2017/5/26

フォーカス位置 d（合焦フォーカス位置）を距離値とする．

- 距離画像
  通常の二次元画像のような，色や濃淡を画素値とする画像ではなく，距離値を画素値として持つ画像である．また，一般的に距離画像は，色や濃淡の情報を持っていないため，本稿では各距離値に対応した色を割り当て可視化を行う．

2.2 概要

DFF 法は，フォーカス位置 d の変位により得られる多重フォーカス画像から画素毎に合焦判定を行い，合焦するフォーカス位置 d から距離値を推定する手法である．

本稿では多重フォーカス画像の撮影をコンシューマデバイスであるスマートフォンに搭載されているカメラから行う．DFF 法によって推定した距離値から距離画像を作成する手法を提案する．本手法の流れを図 1 に示す．距離値の推定には，エッジ抽出によって得られるエッジ強度画像の画素値（エッジ強度）を比較することによって合焦しているか否かを判定する．

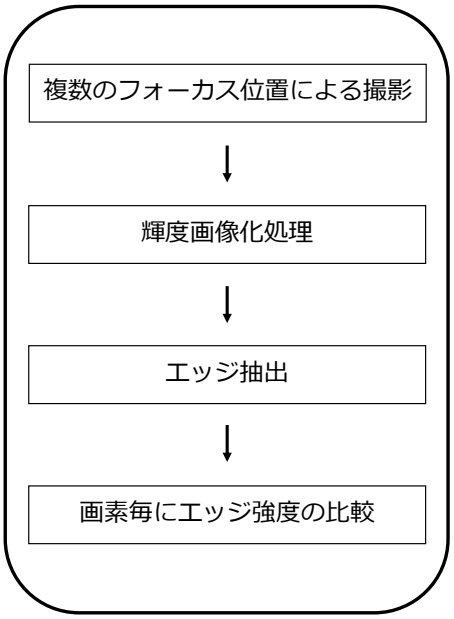

図 1 手法の流れ

2.3 撮影画像

撮影は図 2 のように，スマートフォンから異なる位置に 2 つの物体が置かれていることを想定する．

フォーカス位置 $d = [0.0f, 10.0f]$ までの N 段階に分けて同一場面上で撮影を行うことで得られた N 枚の撮影画像群を入力として処理を行う．このとき，撮影対象である 2 つの物体は，フォーカス位置 d の範囲内で必ず合焦するものであるとする．

Android OS 上で実装するには，API を通じてカメラの制御を行い，撮影を行う．

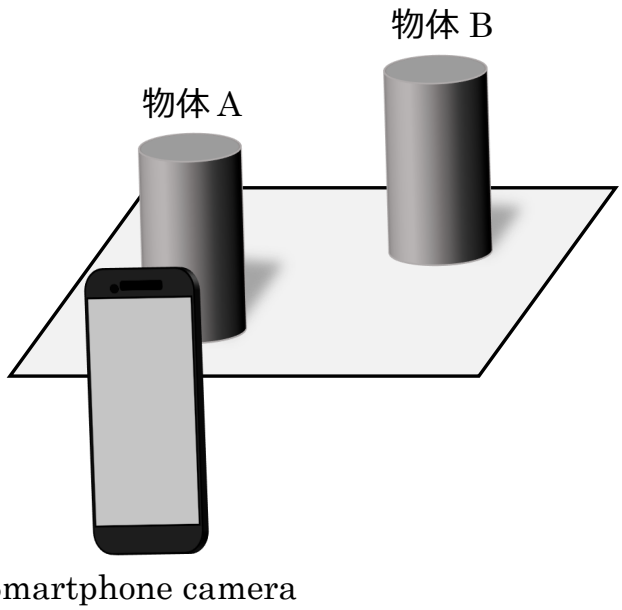

図 2 撮影環境

2.4 輝度画像化処理

多重フォーカス画像から，合焦している画素を検出する際の合焦判定にコントラストを用いる場合は，一般に輝度画像を用いる．

Android OS 搭載のスマートフォンのカメラによる撮影画像は，一般的な用途に用いられることから RGB 出力によるカラー画像である．輝度値に変換するためには，(1) 式により算出する．RGB 値と Y 値の対応図を図 3 に示す．

$$Y = 0.299R + 0.587G + 0.114B \tag{1}$$

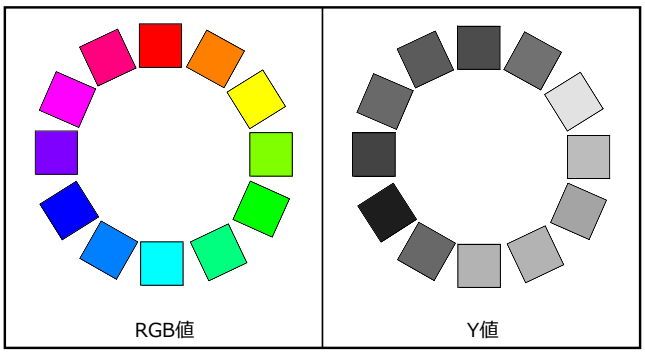

図 3 RGB 値と Y 値の対応図

2.5 エッジ抽出

エッジは，画像処理において色や濃淡変化が大きい物体の境界付近に存在する．ここで，フォーカス位置 d を変位させながら撮影した画像群には，何らかの物体に対して合焦しているか，または合焦せずにぼけているかの画像が存在する．

図 2 の物体 A に対して合焦している画像とぼけている画像を想定すると，物体の境界付近の濃度変化は，ぼけている画像のほうが緩やかである．ここでエッジは，隣接画素に濃淡変化が存在していれば，点として存在している．また，濃淡変化（隣接画素との差分）が大きければエッジ強度は高くなり，小さければエッジ強度も低くなる．

このことから合焦している画像とぼけている画像の濃度変化の違いからエッジ強度も変化すると予想できるためエッジ強度の比較を合焦判定に用いた．

本稿では，Sobel フィルタを用いてエッジの抽出を行う．これによって算出されたエッジ強度を画素値として持つエッジ強度画像から合焦判定を行う．Sobel フィルタを図 4 に示す．

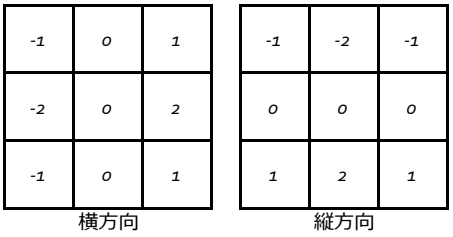

図 4 Sobel フィルタ

2.6 画素毎にエッジ強度の比較

N 段階のフォーカス位置（$d = d_1, d_2, \cdots, d_N$）で撮影された多重フォーカス画像をエッジの抽出を行い，N 枚のエッジ強度画像の画素値を $E(x, y, d)$ とする．また，座標 (x, y) における距離値 $s(x, y)$ として (2) 式より算出する．

$$s(x, y) = \min\{\arg\max_{d} E(x, y, d)\} \tag{2}$$

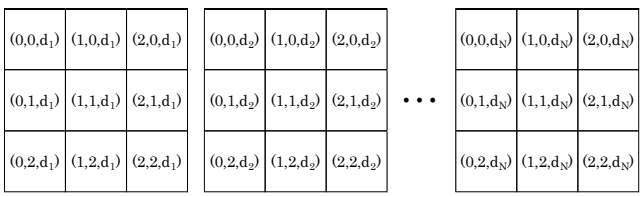

図 5 エッジ強度画像の画素値（エッジ強度）

図 5 のような 3*3 画素の N 枚のエッジ強度画像があるとき，例として座標 $(1, 1, d_1), (1, 1, d_2), \cdots (1, 1, d_N)$ の画素値 (エッジ強度) を比較する．このときに，最もエッジ強度が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て距離値とする．

一般的にフォーカス位置 d は，連続的ではなく離散的な変化であるため若干合焦フォーカス位置に誤差が生じる可能性があるが，ここでは考えないものとする．

また，エッジ強度が複数のフォーカス位置 d で最大となることが考えられる．このときは，最も無限遠寄りの値をとる．

3. Android 端末上での実装

Android 端末上で実装するにあたって 3 種類の方法が考えられる．

(1) 端末上で撮影及び処理を行う
(2) 画像のデータの一部を PC に転送し，処理を行う
(3) 画像のデータを PC に転送し，処理を行う

(1) は，撮影から処理までの一連の流れを端末で行うことで，PC を経由する必要がないが，端末によってメモリ容量等に制約がある．(2)(3) は，撮影は端末で行い，画像のデータを PC に転送を行うことで，メモリ等の制約が減り，実装が容易になる．また，画像データの一部のみを転送することによって，通信量を減らすことが可能である．

本稿では，本手法で実装で Android 端末から得られる画像のデータから，どの程度の距離画像が得られるかの確認を行うため，実装が容易な画像データを PC に転送して処理を行う．ここで，画像のデータは USB 経由で PC に転送するため，(3) を採用する．

3.1 試作アプリケーション

本稿で使用したカメラアプリケーションは，Andorid アプリケーションの統合開発環境である Android Studio[*2] を用いて開発した．使用したプログラミング言語は Java である．Google 社が提供している Camera2 API のサンプルソースである android-Camera2Basic[*3] を参考にしてマニュアルフォーカスモード搭載で，自動撮影が可能なカメラアプリケーションを開発した．

3.1.1 端末の対応条件

このアプリケーションは，現状必ずしもすべてのスマートフォンで対応しているわけではないため，対応する条件について説明する．

まず，Camera2 API は，API Level[*4] 21 であり， すなわち Android 5.0（Lollipop）以上である必要がある．そのため Android 4.4（KitKat）以下のバージョンでは動作対象外であり，使用することができない．

次にスマートフォンのカメラが Camera2 API に対応し

*2 https://developer.android.com/studio/index.html?hl=ja
*3 https://github.com/googlesamples/android-Camera2Basic
*4 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

表 1　デバイスのサポートレベル

0	INFO_SUPPORTED_HARDWARE_LEVEL_LIMITED
1	INFO_SUPPORTED_HARDWARE_LEVEL_FULL
2	INFO_SUPPORTED_HARDWARE_LEVEL_LEGACY
3	INFO_SUPPORTED_HARDWARE_LEVEL_3

ているか否かを確認する必要がある．CameraCharacteristics クラスの INFO_SUPPORTED_HARDWARE_LEVEL を使用する．INFO_SUPPORTED_HARDWARE_LEVEL は，int 型の定数（satatic final のフィールド）として定義されており，[0,3] の値が格納されている．格納されている値に対応したサポート状況一覧を表 1 に示す．

Android 5.0 以上のすべてのデバイスに対して INFO_SUPPORTED_HARDWARE_LEVEL_LEGACY はサポートされており，Camera2 API の下位互換である Camere API と同程度の機能を有している．しかし，マニュアルフォーカスを実装するには，最低でも INFO_SUPPORTED_HARDWARE_LEVEL_FULL がサポートされている必要がある．

3.1.2　マニュアルフォーカスモードの実装

マニュアルフォーカスモードを実装するにあたってまず，AF（オートフォーカス）を OFF にする必要がある．カメラからの入力画像キャプチャを要求する際には CaptureRequest.Builder クラスの set() を通じて適当な設定パラメータをセットする．ここで，AF を無効化するためには，CONTROL_AF_MODE に CONTROL_AF_MODE_OFF を設定する．(e.g. set(CaptureRequest.CONTROL_AF_MODE, CameraMetadata.CONTROL_AF_MODE_OFF))．これらのシンボルは，CaptureRequest クラスと CameraMetadata クラスの int 型の定数（satatic final のフィールド）として定義されている．

次に最近接撮影距離の値を取得する．
CaptureRequest.Builder クラスの get() を通じて，LENS_INFO_MINIMUM_FOCUS_DISTANCE を設定する．
(e.g. get(CameraCharacteristics.LENS_INFO_MINIMUM_FOCUS_DISTANCE))．このシンボルは，Float 型の定数（satatic final のフィールド）として定義されている．

最後にフォーカス位置 d の制御を行う．
CaptureRequest.Builder クラスの set() を通じて，LENS_FOCUS_DISTANCE とフォーカス位置 d を設定する．
(e.g.set(CaptureRequest.LENS_FOCUS_DISTANCE,0.0f))．このシンボルは，CaptureRequest クラスの Float 型の定数（satatic final のフィールド）として定義されている．

3.1.3　DFF 用画像取得アプリケーションの試作

DFF 法による処理速度等に関する検討は行わないことから，DFF のための多数枚の画像取得のみを行うアプリケーションプログラムを試作した．3.1.2 節で説明したようなマニュアルフォーカス設定を用いて，端末搭載カメラ上で API 制御可能な全域でフォーカス位置を変化させながら画像取得を行う．利用者からのアクションに基づく画像取得のほか，Timer ならびに TimerTask を利用して一定間隔で自動取得する機能などを実装した．

表 2　撮影に使用した端末

Device	Nexus 5X
OS	Andorid 7.0
CPU	Qualcomm Snapdragon 808 (MSM8992) 1.8GHz ＋ 1.4GHz ヘキサコア
Memory	2GB

4.　実験

4.1　目的

試作アプリケーションを用いて DFF 法による奥行き情報取得のための画像取得とその処理結果に関する性能などについて検討することを目的として，撮影ならびに処理実験を行った．端末上の機能検証では，100 枚程度の連続画像取得について，可能な時間やその他の機器への影響などを確認する．得られた画像群による性能検証では，距離画像を生成し，端末カメラによる奥行き情報取得精度の基礎的な検討を行う．

4.2　実験環境

撮影に使用した Android OS 搭載のスマートフォンを表 2，画像処理に使用した PC を表 3 に示す．

表 3　処理に使用した PC

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

Nexus 5X は，Google と LG によって共同開発されたスマートフォンであり，カメラは 1230 万画素である．また，ピクセルサイズ 1.55[μm]，絞り $f/2.0$ である．

Nexus 5X の HARDWARE_LEVEL は，LEVEL_3 であり，マニュアルフォーカスモードの搭載が可能である．

画像処理に使用したプログラミング言語は python であり，OpenCV で提供されているライブラリーをインポートして処理を行った．

4.3　撮影環境

撮影環境を図 6 に示す．まず，白と黒の縞模様が印刷された紙を 2 つ用意する．スマートフォンを固定してから，カメラから距離が異なる位置に縞模様が印刷された紙を置く．このとき，カメラからの距離を右の紙から約 15[cm], 30[cm] として撮影を行った．

以降本稿では，カメラから約 15[cm] の位置に置かれた紙を紙 α, 30[cm] の位置に置かれた紙を紙 β と記述する．また，紙 α 及び紙 β 以外の領域（2 つの紙が置かれている

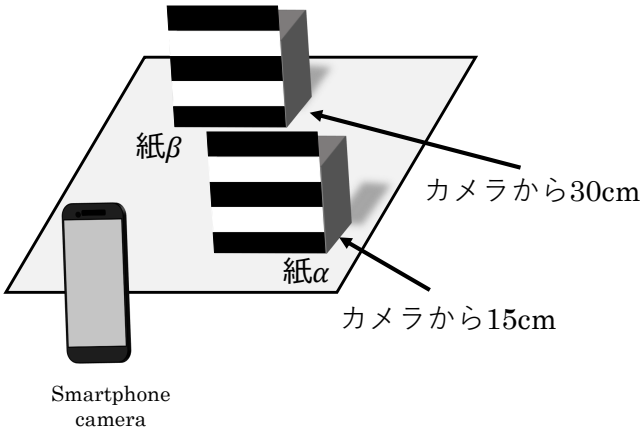

図 6 撮影環境

机を除く）をまとめて背景と記述する．

今回使用したフォーカス位置は，無限遠から最近接撮影距離までの $d = [0.0f, 10.0f]$ をステップ幅 0.1 の 101 段階とした．すなわち，フォーカス位置集合 D は，D $= 0.1if$ $|0 \leq i \leq 100$ である．フォーカス位置集合 D 内のすべてのフォーカス位置に設定した画像を 1 枚ずつ取得する．

撮影する際のシャッター間隔は 0.5, 1, 2, 4 秒の計 4 通りで撮影を行った．

5. 結果と考察

5.1 撮影時間と端末の熱

まず，シャッター間隔を 4 秒，計 405 秒では，端末に発熱が確認されたが問題なく撮影できることを確認した．続いてシャッター間隔を 2 秒，計 203 秒では，4 秒と同様に発熱が確認されたが問題なく撮影できることを確認した．

しかし，シャッター間隔を 1 秒，計 102 秒では，撮影自体はできたが，端末の発熱がひどく連続使用は避けた方が良い結果となった．最後に 0.5 秒，計 51.5 秒では，撮影が途中で止まり，使用した Nexus 5X が故障する結果となった．PC から，撮影時のログを確認したところ，35 枚目の撮影完了後に電源が落ちていたことが確認できた．

熱問題に関して，ステップ幅を大きくして少ない枚数で検証することや，静止画像ではなく，動画を用いる方法が考えられる．

5.2 撮影画像

撮影画像は 4032*3024 画素であり，解像度は水平，垂直方向ともに 72dpi である．また，撮影画像は Jpeg 圧縮として，圧縮パラメータはデフォルトの設定である．1 枚あたりの容量は，950[KB] であり，全体で約 100[MB] である．

図 7 は，フォーカス位置 $d = 1.5f$ のときの撮影画像である．これは，紙 β に対しては合焦しているが，紙 α は，ぼけているときの撮影画像である．また，背景は若干ぼやけている．

続いて図 8 は，フォーカス位置 $d = 6.4f$ のときの撮影画像である．これは，紙 α に対して合焦しているが，紙 β 及び背景は，ぼやけている撮影画像である．

5.3 エッジ強度画像

この 2 枚の撮影画像に対して，Sobel フィルタによってエッジ抽出して作成したエッジ強度画像を図 9, 10 に示す．図 9 は，$d = 1.5f$ のときの撮影画像から作成したエッジ強度画像である．紙 α と紙 β を比較してみると，合焦している紙 β のエッジの方がエッジ強度高いことが確認できる．

図 10 は，$d = 6.4f$ のときの撮影画像から作成したエッジ強度画像である．$d = 1.5f$ のときのエッジ強度画像と同様に紙 α と紙 β を比較してみると，合焦している紙 α のエッジのほうがエッジ強度高い．

背景を $d = 1.5f$ 及び $d = 6.9f$ のときのエッジ強度画像のエッジ強度を比較すると，$d = 1.5f$ では，はっきりとエッジとして検出され，エッジ強度も高くなっている．しかし，$d = 6.9f$ では，エッジ強度は，低い値である．しかし，$d = 6.4f$ の背景にある蛍光灯に対してはっきりとしたエッジが確認できる．これは，蛍光灯の光が強く，ピントずれによるぼけから画像の濃淡の変化が緩やかになったとしても天井と十分に濃淡の変化であったからではないかと考える．

これらのことから，$d = 1.5f$, $6.4f$ のときの 2 枚の撮影画像からエッジ強度画像を作成して同一の 3 つの領域 (紙 α，紙 β，背景) でエッジ強度を比較すると，あきらかな違いが確認できた．

図 7 撮影画像 ($d = 1.5f$)

ここで，$d = 1.8f$ のときのエッジ強度画像から紙 β においてエッジ強度（エッジ部）が 250 以上の座標を 1 点選び座標 a，$d = 6.5f$ のときのエッジ強度画像から紙 α において，エッジ強度（エッジ部）250 以上の座標を一点選び座標 a2 とする．また，$d = 1.8f$ のときのエッジ強度画像から紙 β の平担部（縞模様の黒部分）から一点選び座標 b1,

Vol.2017-UBI-54 No.23
Vol.2017-CDS-19 No.23
2017/5/26

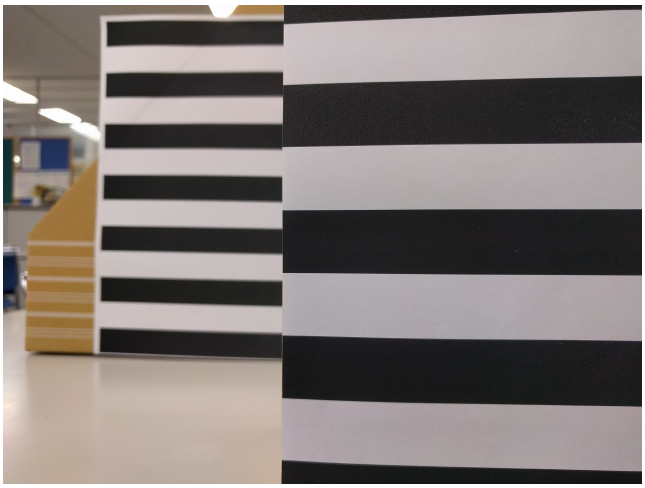

図 8　撮影画像 ($d = 6.4f$)

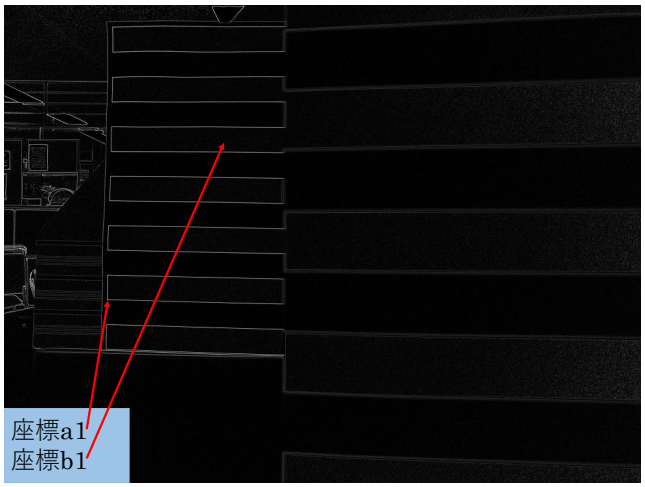

図 9　エッジ強度画像 ($d = 1.5f$)

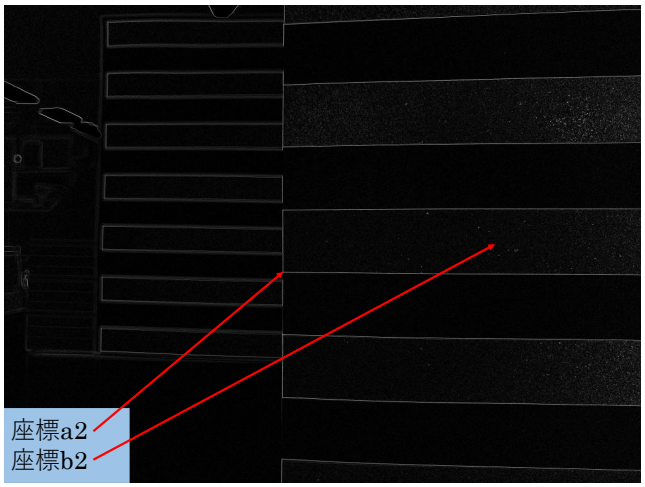

図 10　エッジ強度画像 ($d = 6.4f$)

$d = 6.4f$ のときのエッジ強度画像から紙 α の平坦部（縞模様の黒部分）から一点選び座標 b2 とする．

この 4 点の座標に対して，フォーカス位置 d におけるエッジ強度の関係を図 11 に示す．

座標 a1 及び座標 a2 ともに 1 つの山が確認できる．座標 a1 では，$d = 1.3f$ からエッジ強度が急激に高くなり，$d = 1.5f$ でエッジ強度が最大値を迎えている．座標 a1 で，エッジ強度が最大となる d は 1 点であった．座標 a2 では，$d = 5.7f$ でエッジ強度が高くなり，$d = 6.0f$ から $d = 6.4f$ の間の複数の d で最大値を迎えている．

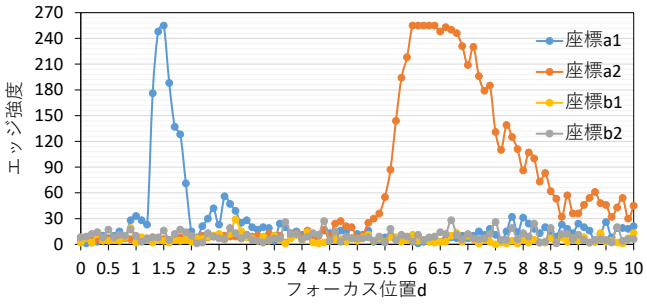

図 11　フォーカス位置 d におけるエッジ強度

座標 a1 では，距離値が $d = 1.5f$ でエッジ強度が最大を迎えていることから 1.5 となる．また，座標 a2 では，エッジ強度が最大となる d が複数ある．この場合は，最も無限遠よりの値をとるため，距離値は 6.0 になる．

エッジ部では，エッジ強度の最大値が 200 以上である画素が存在するが，平坦部では，座標 b1, b2 ともにエッジ強度が 30 以下の値を推移しているため，平坦部の座標 b1, b2 のみでフォーカス位置 d とエッジ強度の関係を図 12 に示す．

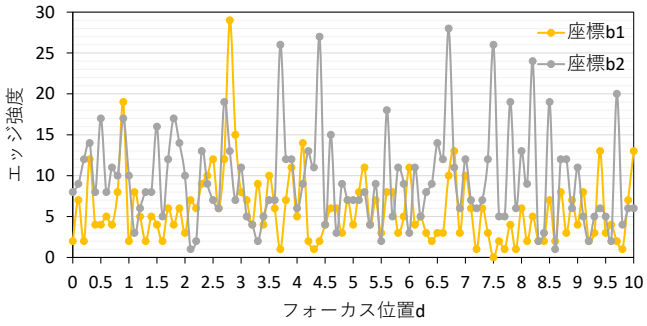

図 12　フォーカス位置 d におけるエッジ強度 (平坦部)

紙 β のエッジ部である座標 a1 と平坦部である座標 b1 と合焦するフォーカス位置を d を比較すると，$1.1f$ ずれが生じている．また，紙 α のエッジ部である座標 a2 と平坦部である座標 b2 と合焦するフォーカス位置 d を比較すると，$0.8f$ ずれが生じている．しかし，座標 a1, a2 の差が $4.5f$ であり，座標 a1, b1 と座標 a2, b2 の差と比較すると，ともに 4 分の 1 程度のずれであるため許容範囲であると考える．

5.4　距離画像

(2) 式を適用して推定された距離値から作成した距離画像を図 13 に示す．距離画像を作成する際に無限遠 ($d = 0.0f$) を青，最近接 ($d = 10.0f$) を赤とし，101 色を用いて色分

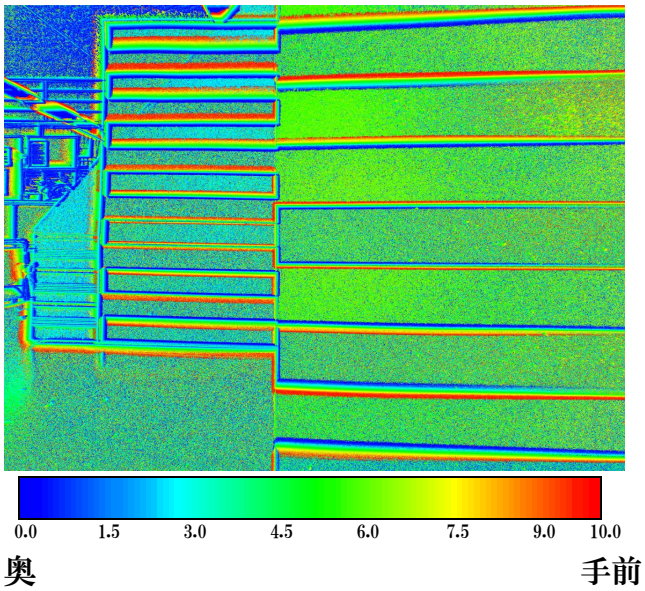

図 13　距離画像

けを行った．

大きく分けて 3 つの距離領域として推定されていることが確認できる．1 つ目の領域は紙 α であり，平坦部に着目すると，$d = 6.5f$ 付近の距離値と対応した緑と黄の割合が多いことが確認できる．2 つ目の領域は紙 β であり，同じく平坦部に着目すると，$d = 2.0f$ 付近の距離値と対応したみずいろの割合が多いことが確認できる．最後の領域は背景であり，$d = 0.5f$ 付近の距離値と対応した青の割合が多いことが確認できる．このことから平坦部及び背景において実際に遠近に対応した距離値が推定できていることが確認できる．

次にエッジ部を着目すると，実際の遠近に対応した距離値とそれに反した距離値が推定されていることが確認できる．これは，スマートフォンのカメラのレンズでフォーカス位置 d を変位させると画角が変化することが影響していると考えられる．

5.5　フォーカス位置変位による画角変化

無限遠（$d = 0.0f$）から最近接 ($d = 10.0f$) までのフォーカス位置 d の変位でどの程度画角が変化しているのかを確認するために，輝度画像化処理を施した $d = 0.0f$ と $d = 10.0f$ の輝度画像をそれぞれ重み 0.5 で合成することで得られた合成画像を図 14 に示す．

図 14 から本研究で使用した端末のカメラの最大の画角ずれを読み取ることができる．紙 β の左上端に注目すると画角変化は水平方向と鉛直方向にそれぞれ同程度の変化が生じていることが確認できる．また，撮影画像の中心付近では縞模様のずれが少なく，中心から外側に行くにつれてずれ幅が大きくなっていることがわかる．

次に，画角変化の影響受けている画素とそうでない画素では，エッジ強度が異なることが考えられるため，画素値

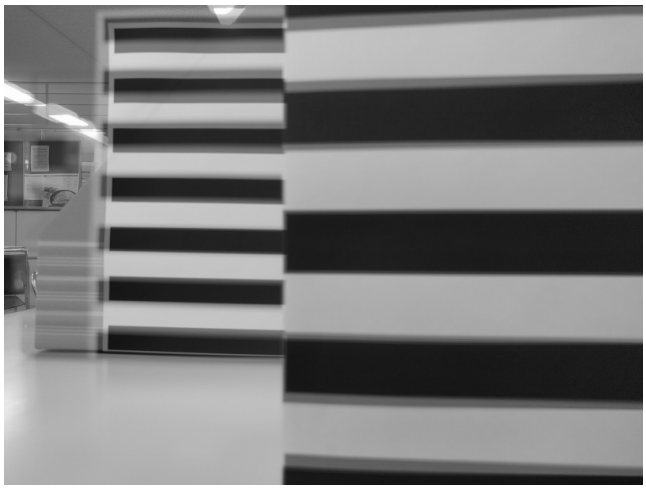

図 14　合成画像

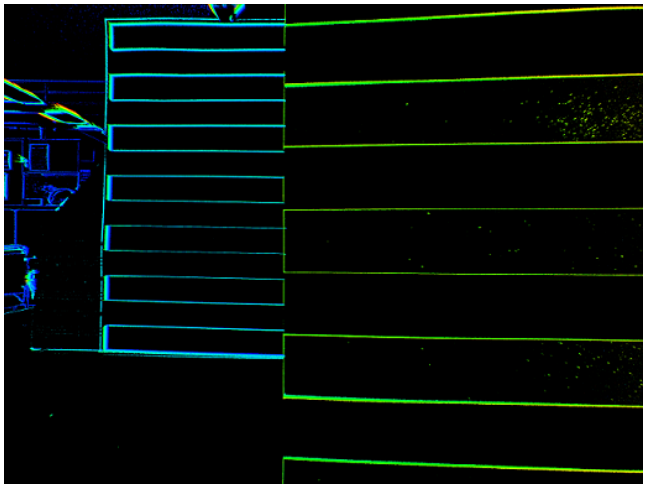

図 15　エッジ強度 127 以上の画素で作成した距離画像

が 127 以上の画素のみで作成した距離画像を図 15 に示す．閾値を設けることで，画角変化による影響の大部分を取り除くことが確認できたが，同時にエッジ強度が低い平坦部まで距離値を失ってしまうため単純な閾値処理では，画角変化の問題が解決できないことがわかる．

次に，8 近傍で画素値から画角変化による影響を受けている画素か否かの判定を行う．距離値 $s(x, y)$ は，各座標でエッジ強度が最大となるフォーカス位置 d であるため，この d におけるエッジ強度画像の画素値を $K_i(x, y)$ とする．ここで，注目画素 K_c，8 近傍画素の画素値 $\{K_0, \ldots, K_8\}$ とし，以下の式で判定を行う．

$$K_{avg} = \frac{1}{8}\sum_{i=0}^{8} K_i \tag{3}$$

$$s(x, y) = \begin{cases} k_c > k_{ave} & s(x, y) \\ k_c < k_{ave} & null \end{cases} \tag{4}$$

判定後の $s(x, y)$ から作成した距離画像を図 16 に示す．黒の画素が今回の判定で距離値でないと判定された画素で

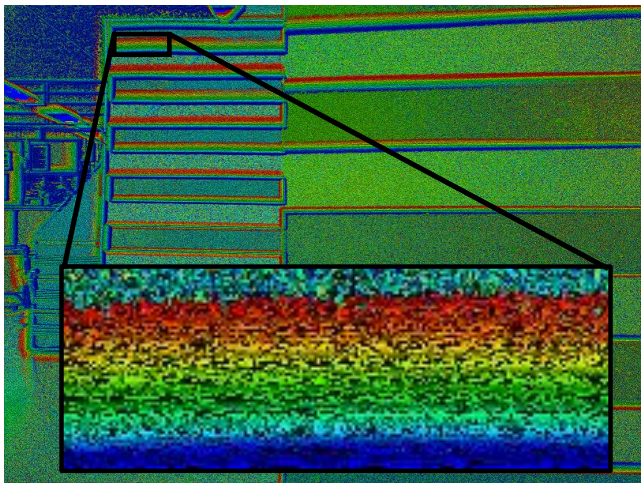

図 16 判定後の距離値で作成した距離画像

ある．単純に注目画素の 8 近傍画素の画素値の平均から，判定を行うとエッジ強度の低い高い関係なく距離値の情報が除去されてしまうのが確認できる．これは，約 1200 万画素に対して 3*3 の 9 画素で判定を行っているため，エッジ強度が高い画素でもその周辺画素も同様に高ければ距離値ではないと判定されてしまうので画角変化の影響にあるエッジ強度の低い画素のみを削除できなかったのではないかと考える．閾値と同じく，単純は近傍画素との比較では，画角変化の問題が解決できないことがわかる．

6. 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フォーカス位置 d を変位させながら撮影した画像から距離画像を作成した．概ね実際の遠近に対応した距離値が推定できていることを確認した．しかし，エッジ部では，フォーカス位置 d の変位による画角変化の影響で，実際の遠近にあった距離値のほかにそれに反した距離値が推定されていることを確認した．これの解決策として，閾値によるフィルタリング及び注目画素とその周辺の画素の比較を行ったが，これだけでは解決に至らなかった．また，撮影の際に端末の発熱が問題になることを確認した．

今後は，フォーカス位置 d の変位でどの程度画角が変化するのかを調査するとともにエッジ部と平坦部に異なる閾値を設定して検証を行う．また，注目画素との比較にも閾値を設定して再度検証を行う．

参考文献

[1] 浅田 尚紀，藤原 久永，松山 隆司，“多重フォーカス画像を用いたエッジ検出と距離計測”，電子情報通信学会論文誌（D-II），vol.J77-D-2，no6，pp.1048-1058，June 1994.
[2] 松山 隆司，竹村 岳，“多重フォーカス画像を用いた実時間 3 次元距離計測”，情報処理学会論文誌，vol.39，no.7，pp.2149-2158，Jul 1998.
[3] 池岡 宏，柏山 英輝，浜本 隆之，児玉 和也，“多重フォーカス画像を用いたスマートイメージセンサによる距離計測”，映像情報メディア学会誌，vol.62，no.3，pp.384-391，2008.
[4] 服部 寛，小野口 一則，渡辺 睦“焦点変動画像系列からの領域分割 ”，全国大会講演論文集，vol.45，pp265-266，Sep 1992.
[5] 工藤 朋之，三池 秀敏，“カメラフォーカスを連続変化させた動画像を用いた奥行き分布検出”，情報処理学会研究報告コンピュータビジョンとイメージメディア（CVIM），vol.1996，no.5，PP.7-12，Jan 1996.
[6] 日浦 慎作，松山 隆司，“構造化瞳をもつ多重フォーカス距離画像センサ”，電子情報通信学会論文誌（D-II），vol.J82-D-II，no11，PP.1912-1920，Nov 1999.
[7] 河村 岳，大原 正満，“単眼カメラで空間認識，「ぼけ」から距離を推定”，日経エレクトロニクス，no.1137，pp.59-67，June 2014.
[8] Schechner, Yoav Y and Kiryati, Nahum，“Depth from defocus vs. stereo: How different really are they?.”, In International Conference on Pattern Recognition，pp.1784-1786, 1998.