

# 携帯端末に適用可能なモーション ブラー警告システム

An Image Blur Alert System for Mobile Device Cameras

手塚 渉太<sup>1</sup> 浅井 洋樹<sup>2</sup>  
山名 早人<sup>3</sup>

Shota TEZUKA Hiroki ASAI  
Hayato YAMANA

スマートフォンのような小さな端末で撮影した写真を確認する際、写真にブレが生じていても小さな画面では確認が困難である。また、ブレの検出・補正に関する従来手法では、被写体が動くことに起因するモーションブラーには対応できていないか、対応できたとしても完全に検出・補正ができないという問題があった。そこで本研究では、携帯端末に適用可能なブレ領域をリアルタイムでユーザに警告するモーションブラー警告システムを提案する。警告するブレには、手ブレや被写体の動きによって生じるモーションブラーに加えて、ピンボケも含まれる。本システムでは、撮影した写真を縦横均等に分割した領域ごとにブレ領域を検出する。ユーザは、ブレが生じた領域を拡大することでブレを確認しやすくなり、撮り直すか否かの判断が容易になる。本システムを用いたユーザ評価実験では、5点満点で使いやすさの被験者平均が4.3点、満足度の被験者平均が4.0点という結果が得られた。

When we check photos taken by mobile devices such as smart phones, it is difficult to find blurs in the photos because of a small screen. Traditional researches about a blur detection and modification cannot detect motion blurs due to moving subjects. Even if they can detect them, they cannot remove all the motion blurs. In this paper, we propose a motion blur alert system for mobile devices that warns users to show motion blur areas in real time. The kind of the warning blurs include camera shake blurs, motion blurs and out of focus. Our proposed system detects the areas that include blurs from equally divided areas in a photo. By using the system, users can decide a retake of photos easily to zoom blurred areas. Our experiment shows that the score of ease of use is 4.3/5.0 on average, and the score of satisfaction is 4.0/5.0 on average.

## 1. はじめに

スマートフォンのような携帯端末のカメラで写真を撮影する際、ユーザの意図に反してブレが生じることがある。ブ

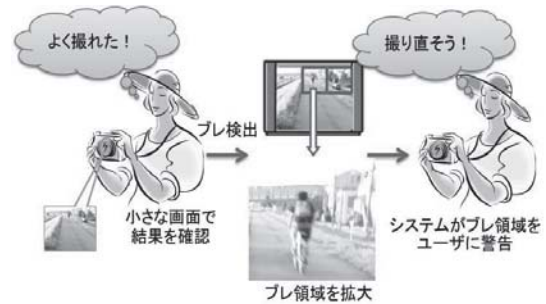


図1 モーションブラー警告システムの概要

レは、画像の品質を悪くさせるにもかかわらず、携帯端末上の小さなディスプレイでは確認が困難である。ブレには大きく分けてモーションブラー、手ブレ、ピンボケの3種類が存在する[1]。モーションブラーは被写体の動きによって生じ、手ブレは撮影者が撮影直前にカメラを動かすことによって生じる。また、ピンボケは被写体にピントが合わないことによって生じる。本研究では以上のいずれかのブレを含む画像をブレ画像とする。

これらのブレを低減するために、手ブレ補正機構やオートフォーカス機能を持つカメラ[2][3][4]が開発されている。しかし手ブレ補正機構は、あくまで手ブレを低減する技術であり、またモーションブラーには効果が無い。オートフォーカスにおいても使用者の意図したピントであったかどうかを撮影後に確認する必要がある。

これまでに提案されている携帯端末上に適用可能なブレの問題を解決する研究[5][6]では、加速度センサや画像処理によってブレ画像を検出する手法を提案している。これらの手法は手ブレやピンボケには対応可能であるが、モーションブラーの検出には適用することができない。そこで本稿では、手ブレとピンボケに加えてモーションブラーにも対応した、携帯端末のカメラに適用可能なモーションブラー警告システムを提案する。

本システムでは、モーションブラーに生じやすい部分的なブレにも対応するため、撮影画像を縦横n個の領域に分割して、それぞれの領域ごとにブレを検出する。ブレ検出は、領域ごとの画像から得られるエッジの情報を用いて、特徴量を抽出し、Support Vector Machine (SVM) によって検出する。本システムを用いることによって、ブレ領域を拡大して確認することで、ユーザは写真を撮り直すか否かの判断が容易になる。

## 2. 関連研究

本節では、携帯端末に適用可能なブレ検出・補正手法に関する研究について述べる。

### 2.1 一般的なカメラ内臓機能を用いる手法

スマートフォンなどの携帯端末のカメラに内蔵されているブレの問題を解決する一般的な機能は、手ブレ補正機構とオートフォーカスの2種類が挙げられる。

西ら[8]は、手ブレ補正機構を搭載したカメラにおいて、手ブレ補正がOFFの場合に比べてONにした場合では、手ブレの影響が平均で1/2から1/3程度に低減することを確認している。しかし手ブレ補正機構は、あくまで手ブレを低減する技術であり、完全には防ぐことができなく、モーションブラーにも効果が無い。またオートフォーカスにおいても、使用者の意図したピントであったかどうかを撮影後に確認する

<sup>1</sup> 非会員 早稲田大学大学院基幹理工学研究科  
ymtezuka@yama.info.waseda.ac.jp

<sup>2</sup> 学生会員 早稲田大学大学院基幹理工学研究科  
asai@yama.info.waseda.ac.jp

<sup>3</sup> 正会員 早稲田大学理工学術院, 国立情報学研究所  
yamana@yama.info.waseda.ac.jp

必要がある。

### 2.2 Sanketiらの手法[5]

Sanketi ら[5]は、スマートフォンの加速度センサを用いてリアルタイムに手ブレを検出する手法を提案した。加速度センサで端末の動きを算出することによって、ユーザに手ブレが生じないように警告する。特に目の不自由なユーザは、ブレが生じていることに気づきにくいいため、リアルタイムで手ブレが生じているかのフィードバックを与えることによって、手ブレが生じているか否かの判断が明確になる。また、加速度センサにおける端末の動きを計算するだけであり処理が軽いため、性能が低い端末でも高速に処理することができる。

しかし、Sanketi らの手法は、手ブレに対しての検出には有効であるが、その他のブレに関しては対応していない。

### 2.3 Koらの手法[6]

Ko ら[6]は、画像全体におけるブレを検出・補正する手法を提案した。まず、撮影された入力画像が与えられた際に、その入力画像がブレ画像かブレのない画像であるかの判定を行う。入力画像がブレのない画像と判断された場合は、ブレのパラメータとしてブレの角度と長さが識別される。識別されたブレのパラメータによってPSF(点拡がり関数)が推定され、最終的にブレ画像の補正が行われる。

ブレ画像の検出は、100枚のブレのない画像と、100枚のブレ画像を学習画像として、それらの画像のエッジ量の平均と標準偏差を算出した分布を用いて行う。算出された分布を用いて、ブレ画像がブレのない画像よりもエッジ量の平均と標準偏差の値が小さくなるという性質を利用して、入力画像のブレ検出をおこなっている。

しかし、Ko らの手法はモーションブラーといった部分的なブレに対しては有効ではなく、エッジ量の平均と標準偏差のみの特徴しか用いていないため、精密なブレ判定が難しいという問題がある。

## 3. 提案手法

本節では、関連研究で上げた問題点を考慮しつつ、ブレの警告を目的とした提案手法について述べる。

### 3.1 提案手法の概要

提案手法では、まず撮影直前における端末の加速度を加速度センサで保存しておき、入力画像における手ブレを検出する。この時、端末が一定以上動いた場合は入力画像全体を手ブレ画像としてユーザに警告する。手ブレ画像と判定されなかった画像は、画像全体を縦横  $n$  個の領域に分割して、領域ごとに特徴量の抽出を行ってブレ検出を行う。

特徴量抽出では、ブレ検出を行うために、ブレ領域にはエッジが検出しにくいという特徴と、段階的にブレ領域を縮小するとエッジが検出しやすくなるという特徴を利用し、合計で6種類の特徴量を抽出する。その後、予め学習しておいたSVMのモデルを用いて6種類の特徴量を統合して、分割した領域ごとにブレ検出を行う。ブレ検出の結果から、最終的に入力画像のブレ領域をユーザに警告する。

### 3.2 加速度センサでの手ブレ検出

2.2で述べたSanketi らの手法から、端末の加速度センサを用いて手ブレの検出を行う。撮影直前に  $x$  軸、 $y$  軸、 $z$  軸のそれぞれの方向から加速度センサで端末の加速度を取得しておき、その3軸の加速度のいずれかが  $0.5[m/s^2]$  を超えた場合は、入力画像全体を手ブレ画像としてユーザに警告す

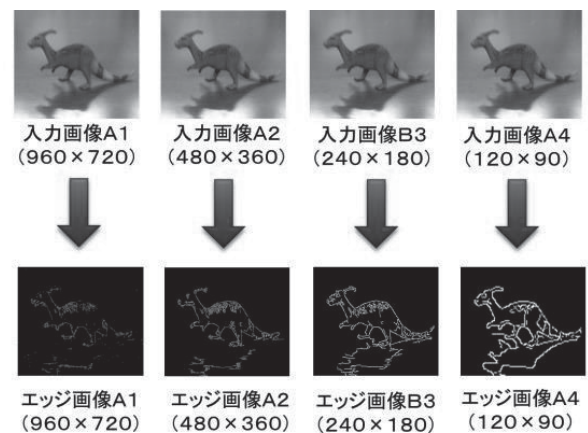


図2 画像サイズを変化させた場合のエッジ検出

る。また3軸の加速度が  $0.5[m/s^2]$  を下回る場合は、次の3.3節での領域分割の処理を行う。手ブレか否かの3軸の加速度の閾値は、経験的に定めた値である。

### 3.3 領域分割

入力された画像に関して、モーションブラーなどの部分領域のブレにも対応できるように、画像を縦横  $n$  個の領域に分割して、領域ごとに特徴量抽出を行う。分割数は、一般的に多いほど部分領域のブレにも対応できるが、多すぎると処理時間が増えたり、ユーザが検出結果を確認しづらくなったりする場合がある。

### 3.4 特徴量抽出

特徴量抽出では、エッジ情報を用いた6種類の特徴量を抽出してSVMでブレの予測を行う。6種類の特徴量をそれぞれ特徴量A, B, C, D, E, Fと表す。

#### 3.4.1 特徴量A, B, C, Dの算出

2.3で述べたKo らの手法から、エッジ検出における画素値平均と標準偏差を比較することは、ブレ領域とブレのない領域を区別するのに有効な手段であると考えられる。画素値の平均と標準偏差は、どちらの値も大きいほどブレのない領域である可能性が高い。反対に平均と標準偏差がどちらの値も小さいほど、ブレ領域である可能性が高い。ただし、平均と標準偏差の片方の値が大きくて、もう片方の値が小さい場合も考えられる。

そこで、提案手法においても領域分割された画像に対して、Canny エッジ検出[9]により画素値の平均と標準偏差をそれぞれ別の特徴量として用いる。領域分割された画像において、それぞれの領域ごとに画素値の平均と標準偏差を計算する。画像領域のサイズが  $W \times H$  画素の画像領域中の位置  $i, j$  の画素値を  $f(i, j)$  とすると、画素値の平均  $\mu$  (特徴量A) と標準偏差  $\sigma$  (特徴量B) は式(1)、式(2)のように計算できる。

$$\mu = \frac{1}{W \times H} \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} f(i, j) \quad (1)$$

$$\sigma = \sqrt{\frac{1}{W \times H} \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (f(i, j) - \mu)^2} \quad (2)$$

また、画像を縮小していくと、ブレによって検出できな

ったエッジが検出される場合がある。例えば、図2のような場合である。つまり、ブレ画像では、画像を縮小していくと、エッジ画像における画素値の平均と標準偏差が大きくなる傾向がある。また、もともとエッジがしっかり検出できているブレのない画像では、平均と標準偏差が大きく異なる傾向はあるが、ブレのある画像に比べて上昇率は低いと考えられる。

そこで提案手法では、まず領域分割された画像の領域を、それぞれの領域が縦横 1/2 になるように段階的に縮小する。縮小回数は、縦もしくは横が 20 ピクセルを下回るまでの回数である。これにより、それぞれの領域ごとに様々なサイズの縮小画像が生成されるので、領域ごとにエッジの平均と標準偏差の上昇率を計算し、それぞれを別の特徴量とする。画像の縮小回数を  $n$ 、縮小  $i$  回目における画素値の平均を  $\mu_i$ 、標準偏差を  $\sigma_i$  とすると、平均の上昇率  $\alpha_\mu$  (特徴量 C)、標準偏差の上昇率  $\alpha_\sigma$  (特徴量 D) を、最小二乗法の考え方をを用いて式 (3)、式 (4) のように定義する。ただし、縮小 0 回目の場合は、元のサイズの画像における平均と標準偏差の値である。

$$\alpha_\mu = \frac{(n+1) \sum_{i=0}^n i \mu_i - \sum_{i=0}^n i \sum_{i=0}^n \mu_i}{(n+1) \sum_{i=0}^n i^2 - (\sum_{i=0}^n i)^2} \quad (3)$$

$$\alpha_\sigma = \frac{(n+1) \sum_{i=0}^n i \sigma_i - \sum_{i=0}^n i \sum_{i=0}^n \sigma_i}{(n+1) \sum_{i=0}^n i^2 - (\sum_{i=0}^n i)^2} \quad (4)$$

### 3.4.2 特徴量 E, F の算出

画像を一旦縮小させてから元の画像サイズに戻すように拡大すると、画像の高周波成分が失われ、全体的にボケた画像になる。もともとブレのない箇所は、縮小・拡大させると、画像のボケの影響でエッジが抽出されにくくなる特徴がある。しかし、もともとブレのある箇所は縮小・拡大させてもエッジが検出できる量が縮小前とあまり変わらないという特徴がある。これらの特性を利用して、入力画像を A とし、A を縦横 1/4 のサイズに縮小させてから縦横 4 倍のサイズに拡大した画像を画像 B とすると、画像 A, B をそれぞれエッジ検出して差分をとった画像における画素値平均や標準偏差の値を特徴量として用いる。これらの値を用いることによって、もともとブレがあった領域とブレのなかった領域を比較することが可能となる。

差分をとった画像における画素値平均や標準偏差の値の算出は、まず画像 A, B のそれぞれに対して Laplacian フィルタ [10] を用いてエッジ検出した画像を画像 A', B' とする。画像 A' と B' は同じ画像サイズであることから、画像領域のサイズが  $W \times H$  画素の画像 A' における位置  $i, j$  の画素値を  $f_a(i, j)$  とし、画像 B' における位置  $i, j$  の画素値を  $f_b(i, j)$  とすると、画像 A' と B' の差分画像  $I_{diff}$  における位置  $i, j$  の画素値  $f_{diff}(i, j)$  は式 (5) のように計算できる。また、式 (5) から、差分画像の画素値平均  $\mu_{diff}$  (特徴量 E) と標準偏差  $\sigma_{diff}$  (特徴量 F) を式 (1)、式 (2) と同様に計算すると、それぞれ式 (6)、式 (7) のように算出できる。

$$f_{diff}(i, j) = |f_a(i, j) - f_b(i, j)| \quad (5)$$

$$\mu_{diff} = \frac{1}{W \times H} \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} f_{diff}(i, j) \quad (6)$$

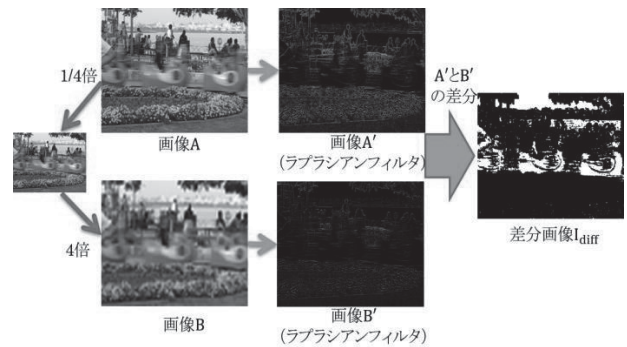


図3 特徴量 E, F の算出

$$\sigma_{diff} = \sqrt{\frac{1}{W \times H} \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (f_{diff}(i, j) - \mu)^2} \quad (7)$$

本節の処理の経過を図3に示す。

### 3.5 SVM による学習

3.4.1 と 3.4.2 で定義した 6 種類の特徴量 A, B, C, D, E, F を使用し、ブレ検出を行う。学習には Chang ら [11] が開発した LIBSVM を利用した。6 種類の特徴量をそれぞれ 6 次元の特徴ベクトルとし、すべての次元ごとに正規化を行い、ブレ検出のためのモデルを生成する。SVM のカーネルには RBF カーネルを用いた。

### 3.6 後処理

後処理として SVM で誤検出するような特定パターンの例外処理や、ユーザが結果を確認しやすいような修正を行う。

#### 3.6.1 例外処理

図4にあるような単色の壁や青空などもともとエッジがとれない領域を強いブレと判定し、SVM の予測結果が誤った予測をする問題がある。そこで、提案手法では、SVM でブレと判定した領域の中から単色の壁や青空などの領域を例外処理としてブレではない領域として検出結果を反映させる。これらの領域は、周辺が外の色の変化がほとんどないという特徴がある。つまり、カラー画像における画素値の分散を算出することによって、図4にあるような領域の誤検出を防ぐ。

画素値の分散の算出は、まず画像領域のサイズが  $W \times H$  画素のカラー画像領域中の位置  $i, j$  の R 成分の画素値を  $f_r(i, j)$ 、G 成分の画素値を  $f_g(i, j)$ 、B 成分の画素値を  $f_b(i, j)$  とすると、RGB 成分の画素値  $f_{rgb}(i, j)$  は式 (8) のように算出される。式 (8) を用いて、RGB 成分の画素値の平均を  $\mu_{rgb}$  とすると、その分散  $\sigma^2$  は式 (9) のように算出できる。この時、SVM でブレと判定した領域の中から  $\sigma^2$  の値が 10 以下の領域をブレのない領域として検出結果を反映させた。



図4 誤検出する領域例(左:単色の壁, 右:青空)



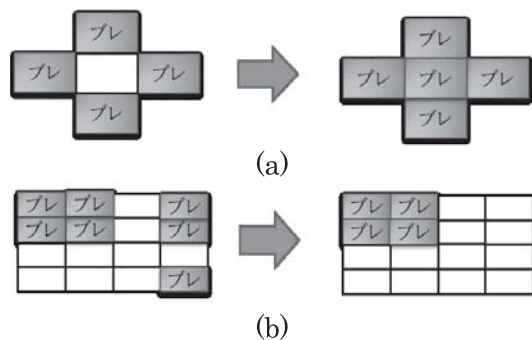


図5 ブレ検出結果の修正例

$$f_{rgb}(i, j) = \frac{f_r(i, j) + f_g(i, j) + f_b(i, j)}{3} \quad (8)$$

$$\sigma^2 = \frac{1}{W \times H} \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (f_{rgb}(i, j) - \mu_{rgb})^2 \quad (9)$$

3.6.2 検出結果の修正

3.3で述べたように、入力画像の領域分割数は、一般的に多いほど部分領域のブレにも対応できるが、多すぎると様々な箇所にブレ領域が存在するような画像では、ブレの警告を行ってもユーザが結果を確認しづらくなってしまいう問題がある。そこで、まずブレ領域が縦横4近傍で連続した領域を同一のグループとみなすラベリング処理を行う。ラベリング処理は、グループごとにそれぞれ別のグループ番号を付与することによってグループ分けを行うことで、画像内に複数のブレ領域がある場合に有効である。ラベリング処理でグループ分けされたブレ領域から、本システムでは次のような修正を行った。

1. 4近傍がブレ領域である場合は、検出結果にかかわらずブレ領域とし、同一のグループとする(図5-a)。
2. グループのブレ領域が4領域未満の場合は警告しないようにする(図5-b)。

上記のような条件を加えることで、ブレ領域のグループが少なくなり、ユーザが結果を確認しやすくなる。ただし、これらの条件は、様々な画像におけるブレの警告結果を確認した中で、経験的に設定したものである。

4. モーションブラー警告システムの実装

本節では、3節で述べた提案手法を用いて実際にAndroid端末のカメラアプリケーションへの実装について述べる。図6にもとづいてユーザが写真の撮影からモーションブラーを発見するまでの流れを以下に示す。

- step1. 画面右下のシャッターボタンを押すことによって、ユーザが写真を撮影する。
- step2. 画面左下のプレビューボタンを押すことによって、写真をプレビュー画面に表示する。ブレと判定された領域は赤枠で囲まれる。
- step3. 赤枠部分をダブルタップすると周囲が拡大され、ブレの状態を確認する。
- step4. ユーザは確認した結果によって撮り直すか否かの判断を行う。

Step2では、撮影後、プレビューボタンは図6(b)のように

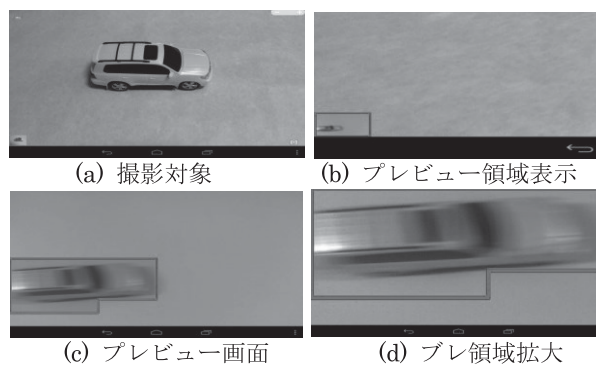


図6 撮影からモーションブラー発見までの流れ

ブレ領域が存在すれば赤枠、存在しなければ白枠で囲まれる。プレビューボタンを押すことにより、システムが自動的にブレ領域を表示する。本システムではnの値を5から10までの好きな分割数に設定できる。

Step3では、図6(c)に示されるように、プレビュー画面においてブレが生じている領域は赤枠で囲まれ、赤枠内の領域全体は赤で透過させている。ユーザが画面にタッチしている間や拡大している間は、透過状態を解除する。また、拡大・縮小に関しては、画面をダブルタップすると図6(d)に示されるように、タップした箇所の周辺を2倍に拡大することができ、再びダブルタップを行うと本のサイズに戻る。

Step4では、ユーザがシステムの警告したブレ領域を確認した後、ユーザが写真を撮り直すか否かの判断をする。システムは、背景領域などユーザにとってブレが生じていても構わない箇所に関しても警告する可能性があるため、最終的な撮り直しの判断はユーザに委ねられる。

5. 評価実験

本節では、本システムの実験について述べる。まず、5.1で本システムを実装した端末の実行環境について述べ、5.2で実験の内容について述べる。最後に、5.3で実験結果を示す。

5.1 端末の実行環境

本システムをAndroidのカメラアプリケーションとしてAndroid端末に実装し、本システムを評価する。本システムを実装したAndroid端末はNEXUS7とXperia acro HD(S0-03D)の2機種である。これらの端末の実行環境を表1に記載する。実験では、NEXUS7を主に使用し、S0-03DはNEXUS7との処理時間の比較に使用した。

表1 実験に用いた端末のスペック

	NEXUS 7	SO-03D
CPU	APQ 8064 Quad-core (1.5GHz)	MSM 8260 Dual-core (1.5GHz)
メモリ	2GB RAM	1GB RAM
OS	Android 4.4.2	Android 4.0.4

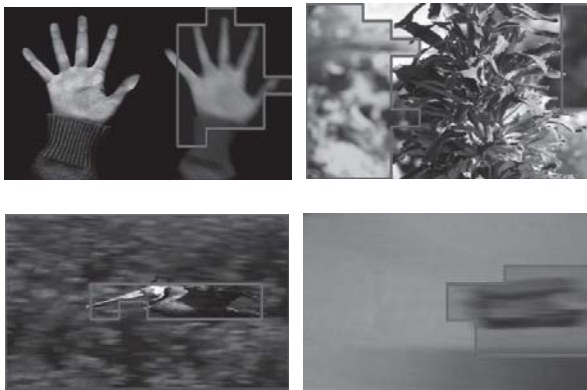


図7 プレ検出例

5.2 実験内容

本節では、本システムを組み込んだ Android 端末を用いて以下の3種類の実験を行う。

1. プレ検知の処理時間
2. 部分領域にブレのある画像に対するブレ検知
3. システムの使用感を評価するユーザ評価

1. の処理時間を算出する実験では、NEXUS7 と SO-03D の2機種を用いて、領域分割からブレ警告するまでの処理時間を測定する。2. の部分領域にブレのある画像に対してブレ検知を行う実験では、システムがブレ領域のある画像に対してどのような検知を行うかの具体例を示す。3. のユーザ評価実験では、実際に被験者が本システムを使うことでシステムの使い勝手や満足度を評価する。

ユーザ評価実験では、本システムを組み込んだ NEXUS7 を用いて、次の手順で評価を行った。

1. 自由に写真を撮影する
2. プレ検知結果を確認する
3. 結果に関して、本システムをユーザインタフェース・精度・処理時間の3つの観点から評価する
4. 1から4までを20回繰り返す
5. 本システムの全体的な使いやすさ・満足度の評価に加え、使ってみた感想・意見などを自由に記入する。

撮影する写真は被験者が自由に決めるが、モーションブラーを中心としたブレが生じた写真と、ブレのない写真をバランスよく撮影するように事前に説明を行った。被験者一人に対して写真を撮影する枚数は20枚とし、撮影した写真1枚ごとにユーザインタフェース・精度・処理時間の3つの観点からそれぞれ評価する。評価の選択肢として1~5の5段階を与えた。数値が大きいほど評価が高いことを示している。

また、20枚撮影し終わった後、被験者はアンケート用紙に本システムの全体的な使いやすさ・満足度を同様に5段階で評価し、感想や意見を自由に記入する。

5.3 実験結果

本節では3種類の評価実験の結果について述べる。

5.3.1 プレ検知の処理時間

本節ではブレ検知の処理時間を計測した結果について述べる。撮影画像の画像サイズは SO-03D の画面サイズと同様の 1280×720 のサイズを用いて、分割数は 5×5 の 25 等分から 10×10 の 100 等分までの処理時間をそれぞれ計測する。処理時間は、撮影画像が領域分割されてからブレ警告をするまでの時間を示す。各端末における処理時間の計測結果を表2に示す。表2の結果から NEXUS7 のスペックであれば 100

分割未満を1秒以内に処理可能であることがわかる。

表2 各端末における処理時間の計測結果

分割数	NEXUS7 (秒)	SO-03D(秒)
5×5 (25 等分)	0.546	1.012
6×6 (36 等分)	0.557	1.115
7×7 (49 等分)	0.611	1.172
8×8 (64 等分)	0.663	1.201
9×9 (81 等分)	0.746	1.352
10×10(100 等分)	0.823	1.428

5.3.2 プレ検知の例

図7に本システムにおけるブレ検知の例を示す。検知結果より、背景など被写体以外にブレが生じている場合でも、システムはブレと判断していることがわかる。つまり、被写体以外がブレと判断している場合は、写真の取り直しを行う必要がないなどの、最終判断はユーザの主観的な判断に委ねられることになる。

5.3.3 ユーザ評価

著者らの所属する大学の学生、男性5名、女性2名の合計7名に評価実験への参加を依頼し、ユーザ評価を行った。実験結果について、まず被験者ごとの撮影した20枚における UI・精度・処理時間の3つの観点での評価の平均点、標準偏差を示した結果を表3、表4に示す。表3から被験者平均をみると、すべての観点において3点以上の評価を得ていることがわかった。特に、処理時間の点数では、被験者平均が4.64点となり、満足している人が多い傾向にあった。また表4から、処理時間の標準偏差は小さい値となったが、精度の標準偏差は大きく、点数にばらつきのある結果となった。このことから、ブレ検知結果の精度面において、写真のシーンによって満足できるか否かの差が大きいことがわかる。

次に、被験者ごとのシステム全体の使い勝手・満足度を評価した結果を表5に示す。表5から、全体的なシステムの使いやすさ・満足度の被験者平均が4点以上という結果が得られた。また被験者の感想の中で、良い点では処理時間について述べられている意見が多かった。一方、悪い点では、ブレ検知の精度面について述べている意見が多かった。

表3 各評価における被験者ごとの平均点 (小数点以下第3位は四捨五入)

	UI(点)	精度(点)	処理時間(点)
被験者 A	3.65	3.20	3.85
被験者 B	3.95	4.10	5.00
被験者 C	4.15	4.25	4.50
被験者 D	4.20	3.55	4.90
被験者 E	4.95	3.95	5.00
被験者 F	4.10	3.40	4.25
被験者 G	4.65	3.65	4.95
被験者平均	4.24	3.73	4.64

表 4 各評価における被験者ごとの標準偏差  
(小数点以下第3位は四捨五入)

	UI (点)	精度 (点)	処理時間 (点)
被験者 A	0.57	0.98	0.35
被験者 B	0.38	0.83	0.00
被験者 C	0.79	0.83	0.50
被験者 D	0.75	0.97	0.30
被験者 E	0.22	1.07	0.00
被験者 F	0.44	0.73	0.43
被験者 G	0.65	1.24	0.22
被験者平均	0.54	0.95	0.26

表 5 被験者ごとのシステム全体の評価

	使いやすさ (点)	満足度 (点)
被験者 A	3	3
被験者 B	4	4
被験者 C	4	4
被験者 D	5	5
被験者 E	5	5
被験者 F	5	3
被験者 G	4	4
被験者平均	4.3	4.0

6. まとめ

本稿では、携帯端末に定期用可能なブレ領域をリアルタイムでユーザに警告するモーションブラー警告システムを提案した。警告するブレには、手ブレや被写体の動きによって生じるモーションブラーに加えて、ピンボケも含まれる。モーションブラーに多い部分領域によるブレに対応するために、本システムでは撮影した写真を縦横均等に分割した領域ごとに、ブレ領域を検出した。本システムは、Android 端末のカメラアプリケーションとして実装を行った。

ユーザ評価実験では使いやすさに関しては7人中6人が4点以上で、被験者平均が4.3点、満足度に関しては7人中5人が4点以上で被験者平均が4.0点となり、どちらの評価も7人全員が3点以上という結果が得られた。

今後の課題としては、ブレ検知結果の精度向上のため、写真のシーンによって特徴量の重みを変動させるなど画像の特性に合わせた特徴量の重み付けの検討や、ユーザがより簡単にブレを発見できるような画像の領域分割手法の改善があげられる。

[文献]

[1] DIGITAL PRINT LAND, “写真と画質について”, <http://www.digipir.ne.jp/photo/photo006.html>  
 [2] All About, “手ブレ補正機構とは?”, <http://allabout.co.jp/gm/gc/54430>  
 [3] PhotoSolid, <http://www.morphonic.com/products/PhotoSolid.htm>

[4] 加藤聰, “カメラ高画質化”, NEC 技報, Vol.61, No.2, 2008.  
 [5] P. R. Sanketi and J. M. Coughlan, “Anti-Blur Feedback for Visually Impaired Users of Smartphone Cameras”, In Proc. of ASSETS, pp.233-234, 2010.  
 [6] J. Ko and C. Kim, “Low cost blur image detection and estimation for mobile devices,” In Proc. of the 11<sup>th</sup> ICACT, Vol.3, pp. 1605-1610, 2009.  
 [7] 手塚渉太, 須藤優介, 山名早人, “複数特徴量の統合による画像劣化検知システム”, DEIM2013, E10-6, 2013.  
 [8] 西一樹, 萩野龍一, 政木康生, 追田真也, 高橋裕, 吉村秀人, “手ブレの3D計測と定量化”, IEICE Technical Report, Vol.106, No.470, pp.49-54, 2006.  
 [9] J. Canny, “A Computational Approach to Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.8, No.6, pp. 679-698, 1986.  
 [10] 奥富正敏, “デジタル画像処理”, CG-ARTS 協会, pp. 114-121, 2006.  
 [11] C.C. Chang and C.J. Lin, “LIBSVM: A Library for Support Vector Machines,” Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm/2001>.

手塚 渉太 Shota TEZUKA

2014 早稲田大学大学院基幹理工学研究科修士課程修了。

浅井 洋樹 Hiroki ASAI

早稲田大学グローバルエデュケーションセンター助手。同大大学院基幹理工学研究科博士後期課程（在学中）。デジタル手書き環境を中心に HCI や知的 UI に関する研究に従事。ACM, IEICE, IPSJ, HIS 各会員。

山名 早人 Hayato YAMANA

1993 早稲田大学大学院理工学研究科博士後期課程修了。博士（工学）。1993-2000 電子技術総合研究所。2000 早稲田大学理工学部助教授。2005 同大学理工学術院教授、国立情報学研究所客員教授。IEEE, ACM, AAI, IEICE. IPSJ 各会員。