

複数特徴量の統合による画像劣化検知システム

手塚 渉太[†] 須藤 優介[†] 山名 早人^{‡§}

[†] 早稲田大学基幹理工学研究科 〒169-8555 東京都新宿区大久保 3-4-1

[‡] 早稲田大学理工学術院 〒169-8555 東京都新宿区大久保 3-4-1

[§] 国立情報学研究所 〒101-8430 東京都千代田区一ツ橋 2-1-2

E-mail: {ymtezuka, sutou, yamana}@yama.info.waseda.ac.jp

あらまし カメラで写真を撮影する際、ブレやぼけ等の理由で撮影された写真が劣化することがある。ブレやぼけは、画像の品質を劣化させるにもかかわらず、撮影機器上の小さなディスプレイでは確認が困難である。そこで本稿では、ブレやぼけによる画像の劣化を撮影者に伝え、撮り直しを促すような手法を提案する。撮り直しを促すことで、撮り直すか否かの判断は撮影者に委ねられる。そして、手ブレだけでなく、被写体の動きによるブレやぼけも改善でき、画質の改善に要する時間も画像復元などに比べ短くすることができる。また、特殊なハードウェアを用いないため、どのようなカメラにも適用可能である。実験では、画像を9等分したブロックごとに劣化検出精度と再現率を求めて評価する。実験の結果、既存の劣化検出手法と比較して検出精度は9.4%、再現率は4.8%の向上が確認できた。また処理時間においても、約400万画素の画像の特徴量抽出までにかかる処理時間を約5.3倍高速化できた。

キーワード 劣化画像, エッジ検出, 特徴抽出, SVM

1. はじめに

カメラで写真を撮影する際、撮影者の意図に反し、ブレやぼけの影響によって劣化した写真が撮影される場合がある。ブレやぼけは、画像の品質を劣化させるにもかかわらず、撮影機器上の小さなディスプレイでは確認が困難である。ブレやぼけによる写真の劣化には、大きく分けて以下の3種類が存在する[1]。

- 手ブレによって生じる劣化
- 被写体の動きによって生じる劣化
- ピントのずれによって生じる劣化

手ブレによって生じる劣化は、撮影者が露光時間中にカメラを動かすことによって生じる劣化であり、被写体の動きによって生じる劣化は、被写体が露光時間中に動くことによって生じる劣化である。また、ピントのずれによって生じる劣化は、被写体にピントが合わず、ぼけが発生することによって生じる劣化である。本稿では、以上の3種類の劣化のうち、いずれかの劣化を含む画像を劣化画像として定義する。

近年、カメラ技術の進歩により、露光時間を短くすることで手ブレや被写体の動きによる劣化を低減するカメラ[2]や、手ブレ補正機構を持つカメラ[3][4][5]が広く普及している。西ら[6]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、露光時間を短くするカメラを用いた場合、露光時間を短くするだけ入射光量が減るので、その分ノイズの影響が増えてしまう。一方、手ブレ補正機構は、あくまで手ブレによる劣化を“低減”する技術であり、完全

には防ぐことはできないという問題がある。

ブレによる画像劣化の問題を解決するもう一つの手法として、ブレによる劣化を除去[7]-[11]、あるいは検出する手法[11]-[14]がある。Shanら[9]は、ベイズの定理に基づいた確率論による手法を用いて、PSF(点拡がり関数)の推定と画像復元の処理を交互に繰り返すことにより、ブレの除去を行った。しかし、Shanらの手法は、ユーザが初期のPSFを設定する必要があり、除去結果がPSFの初期設定に依存するという問題がある。また、PSF推定と除去処理の2段階の処理を繰り返し行うため、処理時間が長くなる。

Raskarら[10]は、ブレ画像と画像中の高周波成分の損失に関係があることに着目し、露光時間中にシャッターの開閉を一定のパターンでコード化するflutter shutter cameraを提案した。flutter shutter cameraを用いることにより、高周波成分の減衰を防ぎ、単純な線形演算でブレを除去することができる。しかし、特殊なハードウェアを必要とし、全てのカメラに対し適用できるわけではないという問題がある。

一方、Koら[11]は、エッジの情報から画像全体における劣化検出を行い、さらに劣化が検出された画像に対して劣化除去まで行っている。しかし、Koらの手法は、部分的な画像の劣化は検知することが出来ない。

また、Liuら[14]は、機械学習を用いて、部分的な画像の劣化を検出し、検出された劣化が被写体の動きによるものか、ピンボケによるものかを分類する手法を提案した。高速にブレやぼけを検出できれば、撮り直しを促すことによって劣化の問題を解決することができるが、Liuらの手法は、色情報などを踏まえた全体

のピクセルを調べることにより劣化の検出を行うため、処理に時間がかかってしまう。

これらの問題を解決するため、提案手法では、カメラで撮影した際に撮影された写真の劣化を検知し、劣化が検知された場合には、撮影者に劣化している部分の通知を行い、撮り直しを促すようなシステムを提案する。ここで、撮影者が劣化部分を確認することによる「撮り直すか否かの判断」は、撮影者に任せることとする。撮り直しを促すことにより、手ブレだけでなく被写体の動きによるブレやぼけも改善することが可能となる。また、複雑な処理を必要としないため、画質の改善に要する時間も画像復元に比べ短くすることができ、特殊なハードウェアも用いないため、どのようなカメラにも適用可能である。本手法では、画像の部分的な劣化に対応するため、画像を N 個のブロックに分割して、それぞれのブロックごとに劣化の検出を行う。劣化の検知は、それぞれのブロックごとに、エッジなどの情報から複数の特徴量を抽出し、それぞれの特徴量に対して **Support Vector Machine (SVM)**により、劣化検出のための閾値を学習することで実現する。

本稿では、次の構成をとる。まず、2章で関連研究について述べ、3章で提案手法を説明する。次に4章で提案手法についての実験と評価を行い、最後に6章でまとめを述べる。

2. 関連研究

本節では、劣化の検出に関係する研究に焦点を当て、劣化検出手法の関連研究について述べる。

2.1. 劣化を検出・除去する手法[11]

Koら[11]は、画像全体における劣化の検出手法を提案した。まず、撮影された入力画像が与えられたときに、その入力画像が劣化画像か劣化の無い理想画像であるかの判定を行う。入力画像が理想画像と判断された場合は、処理を終了する。また、入力画像が劣化画像の場合は、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、最終的にブレによる画像劣化の除去が行われる。画像劣化の検出は、100枚の理想画像と100枚の劣化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、劣化画像が、理想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、入力画像が理想画像か劣化画像かの検出を行っている。あらかじめ学習されたエッジ量の平均と標準偏差の特徴のみを利用することで、携帯端末でも適応できるような処理の軽い検出が可能となる。

しかし、Koらの手法は、部分的なブレやぼけに対す

る劣化には有効ではなく、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、微妙なブレによる劣化判定が難しいという問題がある。

2.2. 劣化を検出・分類する手法[14]

Liuら[14]は、被写体の動きとピンボケによる部分領域の劣化に焦点を当て、単純ベイズによる機械学習を用いた劣化領域の検出と分類の手法を提案した。劣化画像の検出には、以下の3つの特徴量を用いている。

- パワースペクトル
- 勾配ヒストグラム
- 彩度

パワースペクトルは、画像の周波数成分の分布を表現したものであり、ローパスフィルタを通すと高周波成分だけを抽出することができる。劣化領域には、劣化のない領域と比較して高周波成分が少ないという特徴がある。勾配ヒストグラムは、エッジを検出し、検出されたエッジを局所領域ごとに勾配方向で区間分割してヒストグラムを取ったものである。劣化領域には、劣化のない領域と比較して、ブレやぼけの影響により、鋭いエッジが含まれることが少ないという特徴がある。彩度は、色の鮮やかさを表す尺度である。劣化領域には、劣化のない領域と比較して色鮮やかではない、つまり彩度が小さいという特徴がある。Liuらは、以上の特性を踏まえ、パワースペクトル、勾配ヒストグラム、彩度の3つの特徴量を組み合わせることによって、画像の劣化領域を検出した。

しかし、Liuらの手法は、色情報などから画像全体のピクセルを調べる必要があるため、画像サイズが大きくなると処理時間が長くなる。また、Liuらの用いている特徴量だけでは、雲のない青空のような平坦な領域を劣化領域であると判断してしまう可能性がある。

3. 提案手法

本節では、関連研究の問題点を考慮しつつ、画像劣化の検出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

携帯機器上の小さなディスプレイで画像を確認する場合、大きなディスプレイで確認する場合と比べて、画像の劣化具合が分かりにくいという問題がある。小さなディスプレイでも、画像を拡大することにより劣化具合を確認しやすくなるが、確認のための時間が必要となる。この時、ある程度どの部分に劣化が生じているのかがわかれば、ユーザが画像の劣化を確認する際に無駄な手間を削減することができる。

そこで提案手法では、画像全体を N 個のブロックに分割し、分割された画像のブロックごとに画像劣化の検出を行う。実験では、例として N を9とし、9つに等分したブロックを用いた。9等分したブロックのうち、

どのブロックで劣化が生じているかをユーザに提示することで、ユーザは劣化が生じているブロックだけを重点的に確認することが可能となる。なお、 N の値は変更可能である。

提案手法では、劣化検知を行うために、劣化している画像にはエッジが検出しにくいという特徴と、段階的に劣化画像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する。手法の流れは、学習のステップと、学習されたモデルを用いて劣化画像を検知するステップに分けられる。学習のステップでは、学習用の画像全体から7種類の特徴量を抽出し、SVMによりモデルを生成する。劣化画像の検知のステップでは、生成したモデルを用いて、テスト用の画像全体から生成したブロックごとに劣化ブロックか劣化のないブロックかを分類する。

3.2. 提案手法の詳細

提案手法の説明は、ブロックの分割数 N を9として述べる。提案手法の流れを以下に示す。

- Step1. 入力画像を9等分のブロックに分割
- Step2. ブロックごとに7種類の特徴量を抽出
- Step3. 学習済みのSVMモデルを用いて、複数の特徴量を統合
- Step4. 9等分したブロックごとに劣化を検知

本節では、3.2.1項でブロック分割について述べ、3.2.2, 3.2.3, 3.2.4項で7種類の特徴量についての説明を行う。次に、3.2.5項でSVMによる学習について述べる。

3.2.1. ブロック分割

まず、入力された画像に関して、部分領域の劣化にも対応できるように画像を N 個のブロックに分割する。ここでブロックの分割数は、画像中に劣化部分がある場合に撮影者がどの部分を拡大したらよいかわかれば十分なので、実験では例として N を9とし、均等に9つに分割した。特徴量は、9等分されたブロックごとに算出する。

3.2.2. ハフ変換における直線個数

画像劣化が生じている領域は、直線のエッジが検出されにくい特徴がある。そこで、分割された画像に対して、ハフ変換[15]を用いることによりエッジ画像から直線を検出し、検出された直線の本数を1つの特徴量として用いる。検出された直線の本数が少なければ劣化領域である可能性が高く、本数が多ければ劣化のない領域である可能性が高いという特徴がある。

3.2.3. Canny エッジ検出結果画像における画素平均と標準偏差

2.1節で述べたKoらの手法[11]から、エッジ検出における画素平均と標準偏差の値を比較することは、劣化領域と劣化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劣化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劣化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ブロック分割された画像に対して、Canny エッジ検出[16]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。ブロック分割された画像において、ブロックごとに画素値の平均と標準偏差を計算する。画像サイズが $M \times N$ 画素の画像中の位置 i, j の画素値を $f(i, j)$ とすると、画素値の平均 μ と標準偏差 σ は式(1)、式(2)のように計算できる。

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

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

また、画像を縮小していくと、劣化によって検出できなかったエッジが検出される場合がある。例えば、図1のような場合である。

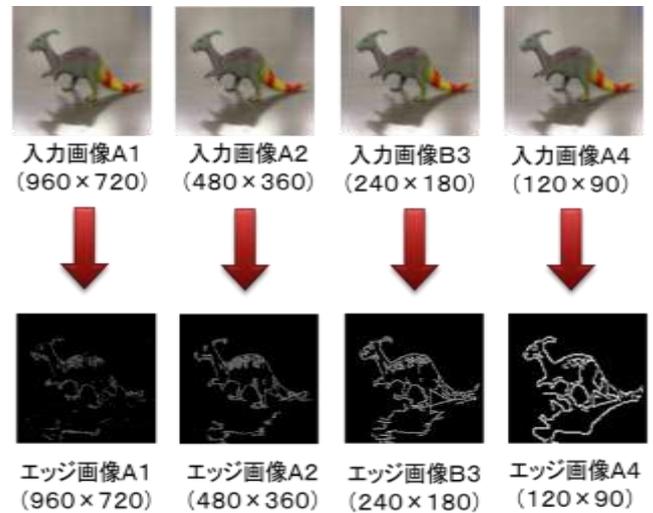


図1 画像サイズを変化させた場合のエッジ検出結果

つまり、劣化画像では、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、元々エッジがしっかり検出できている劣化のない画像では、平均と標準偏差が大きくなる傾向はあるが、劣化のある画像に比べて上昇率は低いと考えられる。さらに、テクスチャのない平坦部は、縮小しても平均と標準偏差は全く上昇しないと考えられるので、平坦部と劣化のある部分との区別も、

平均と標準偏差の上昇率を比較することで可能となる

そこで提案手法では、まずブロック分割された画像のブロックを、それぞれのブロックが縦横 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 50 ピクセルを下回るまでの回数である。これにより、それぞれのブロックごとに様々なサイズの縮小画像が生成されるので、ブロックごとにエッジの平均と標準偏差の上昇率を計算し、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 μ_i 、標準偏差を σ_i とすると、平均の上昇率 α_μ 、標準偏差の上昇率 α_σ を、最小二乗法の考え方をういて式(3)式、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。

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

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

3.2.4. Laplacian フィルタにおける輝度値が最大時の画素の割合

以下、輝度値が最大の部分、つまり輝度値が 255 の場合の画素の割合を L_{max} として定義し、説明を行う。まず、3.2.3 項と同様に画像を縦横 1/2 ずつ段階的に縮小した画像を生成する。ここで、それぞれの画像に対して Laplacian フィルタ[17]をかけてエッジを抽出し、輝度ヒストグラムとして表すと、縮小するごとに L_{max} が大きくなる傾向がある。このとき、劣化のない画像の方が、劣化を含む画像と比較して、縮小していくと L_{max} の値が増え、上昇率も高い傾向がある。そこで、段階的に縮小した画像に対して、 L_{max} の平均値と上昇率をそれぞれ特徴量とする。 L_{max} の平均値 σ_L と上昇率 α_L は、画像の縮小回数を n 、縮小 i 回目における L_{max} の値を L_i とすると、平均値 σ_L 、上昇率 α_L を式(5)、式(6)のように定義する。

$$\sigma_L = \frac{\sum_{i=0}^n L_i}{n+1} \quad (5)$$

$$\alpha_L = \frac{(n+1) \sum_{i=0}^n i L_i - \sum_{i=0}^n i \sum_{i=0}^n L_i}{(n+1) \sum_{i=0}^n i^2 - \left(\sum_{i=0}^n i\right)^2} \quad (6)$$

例として、劣化のない画像と劣化を含む画像における、 L_{max} の値を比較した場合をそれぞれ図 2、図 3 に

示す。ただし、縦軸の画素数は、最大が 1 になるように正規化を行っており、実際は 9 分割されたブロックごとに L_{max} の平均値 σ_L と上昇率 α_L を算出する。

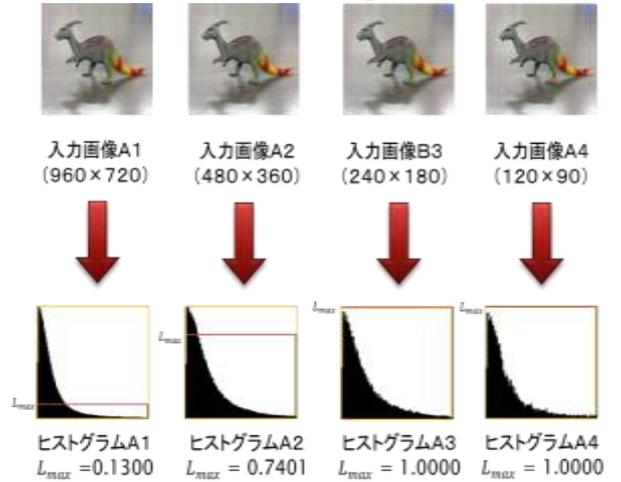


図2 劣化のない画像における輝度ヒストグラム結果 (縦軸：画素数，横軸：輝度値)

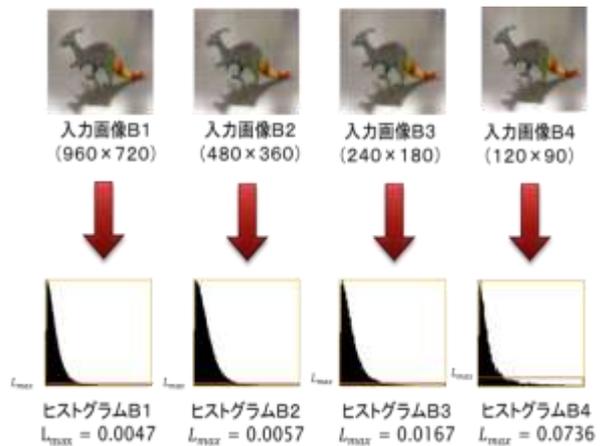


図3 劣化を含む画像における輝度ヒストグラム結果 (縦軸：画素数，横軸：輝度値)

3.2.5. SVM による学習

3.2.2 項、3.2.3 項、3.2.4 項で定義した 7 種類の特徴量を使用し、劣化画像の検知を行う。学習には、Chang ら[18]が開発した LIBSVM を利用した。7 種類の特徴量をそれぞれ 7 次元の特徴ベクトルとし、全ての次元ごとに正規化を行い、画像劣化検知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。RBF カーネルのパラメータは、適当に数値を設定して実験を行い、最も精度の高いものを用いた。

4. 実験・評価

本節では、劣化画像検出の実験について述べる。まず、4.1 節で実験に用いるデータと実験環境について述べ、4.2 節で実験内容について述べる。最後に、4.3

節で実験結果を示す。

4.1. 実験データ・実験環境

まず、実験データとして、人手で 100 枚の劣化画像と 100 枚の劣化のない画像を用意し、劣化画像 50 枚、劣化のない画像 50 枚を学習用の画像とする。そして、残りの劣化画像 50 枚、劣化のない画像 50 枚をテスト用の画像とする。学習用・テスト用の画像は、著者が撮影した画像や、Flickr[19]、Pbase.com[20]のような Web 上の写真共有サイトから収集したものである。実験に用いた PC の実行環境を表 1 に記載する。

表1 実験に用いた PC のスペック

CPU	Intel(R) Core™ i7 960 (3.20GHz)
メモリ	12GB

4.2. 実験内容

実験では、Liu らの手法を対抗手法として、提案手法と同一の実験データで画像劣化検出を行い、結果を比較する。実験内容は、画像を分割して得たブロックごとに、そのブロックが劣化しているかそうでないかの 2 値で判定を行い、それぞれの特徴量ごとの検出の精度、再現率、F 値と、全ての特徴量を統合したときの検出精度、再現率、F 値を算出・比較するというものである。さらに、入力画像分割から特徴量抽出までにかかる処理時間も計測・比較する。ただし、1 つのブロックの中に劣化している箇所と劣化していない箇所が共に存在する場合、人手による劣化画像か否かの判断が難しいため、実験ではこのようなブロックは除外している。また、テクスチャのない平坦部分だけのブロックも検出対象として使用し、平坦部は劣化が生じていないと判断する。実験に用いるブロックの選択基準として、次の 3 つが挙げられる。

1. 劣化している箇所と劣化していない箇所が共に存在するブロックは使用しない
2. 劣化ブロックは、手ブレ、被写体の動き、ピンボケによる劣化のいずれかを含む
3. なるべく類似しているブロックが多くなるようにする

ブロックの選択基準 1 について、人手による劣化か否かの判断が難しいブロックを含めると、提案手法が劣化部分に対して有効な検出ができていないのが曖昧になる可能性がある。手法の目的としては、手ブレか被写体の動きの劣化を判断できることが重要なのではなく、劣化部分に対して正確に劣化検出できることが重要な目的である。よって、そのような判断が難しいブロックを除外した実験によって、提案手法が劣化部分に対して有効であることを示す。

実際に実験に用いたブロックの枚数は、学習用の画

像から 788 枚、テスト用の画像から 500 枚である。テスト用の 500 枚のブロックのうち、半分の 250 枚が劣化のブロックであり、もう半分の 250 枚が劣化のないブロックである。ただし、Web 上の画像は自然に劣化が発生した画像を用いているが、著者が撮影した画像においては、故意に劣化を与えた画像も含まれる。

分類するブロックの総枚数を N 、正しく判定できた枚数を t としたとき、精度 A_c を式(7)のように定義する。また、分類する劣化ブロックの総枚数を N_b 、 N_b の中で正しく判定できた枚数を t_b としたとき、再現率 R_e を式(8)のように定義する。F 値 F_m は、精度 A_c と再現率 R_e を用いて、式(9)のように表される。

$$A_c = \frac{t}{N} \quad (7)$$

$$R_e = \frac{t_b}{N_b} \quad (8)$$

$$F_m = \frac{2 \times A_c \times R_e}{A_c + R_e} \quad (9)$$

4.3. 実験結果

実験結果として、4.3.1 項にブロック検出における精度と再現率の結果を示し、4.3.2 項に処理時間の算出結果を示す。

4.3.1. ブロック検出における精度と再現率の結果

まず、3.2 節で説明した 7 種類の特徴量を簡単のため、以下の表 2 にまとめる。

表2 7種類の特徴量について

特徴量 A	ハフ変換により検出された直線の個数 (3.2.2)
特徴量 B	Canny エッジ画像の画素値平均 (3.2.3 式(1))
特徴量 C	段階的に縮小した特徴量 B の上昇率 (3.2.3 式(3))
特徴量 D	Canny エッジ画像の画素値の標準偏差 (3.2.3 式(2))
特徴量 E	段階的に縮小した特徴量 D の上昇率 (3.2.3 式(4))
特徴量 F	段階的に縮小した L_{max} の平均値 (3.2.4 式(5))
特徴量 G	段階的に縮小した L_{max} の上昇率 (3.2.4 式(6))

7 種類の特徴量 A, B, C, D, E, F を単独で検出を行った精度と再現率をそれぞれ図 4 図 5 示す。

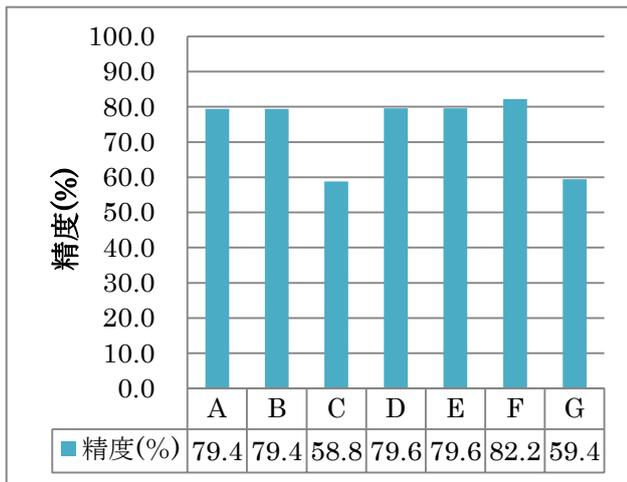


図4 各特徴量における精度

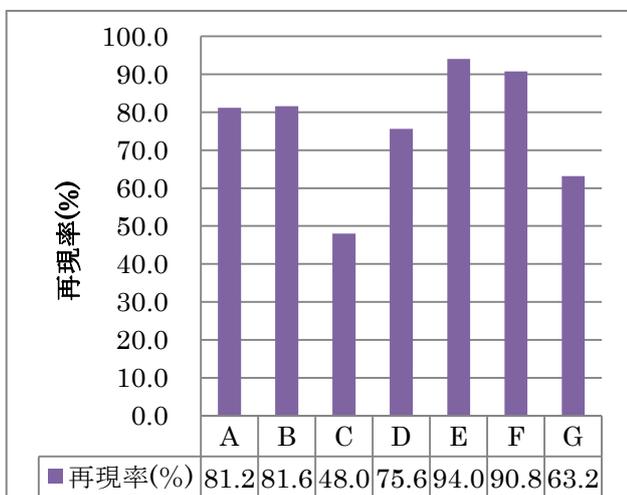


図5 各特徴量における再現率

次に、7種類の特徴量をそれぞれの特徴ベクトルの次元とし、SVMを用いて全ての特徴量を統合させた提案手法における検出結果の算出を行った。また、図4と図5の結果から、精度の面において特徴量Fが82.4%、再現率の面において特徴量Eが94.0%で最も良い結果となった。一方、特徴量CとGは、精度と再現率のどちらも検出精度が65%を下回る低い値となった。よって、精度と再現率が低かった特徴量CとGを除外した検出結果とLiuらの手法での検出結果の算出も行い、提案手法、特徴量CとGを除外した手法、Liuらの手法における検出結果の精度、再現率、F値をまとめた結果を表3に示す。最後に、提案手法、特徴量CとGを除外した手法、Liuらの手法それぞれの精度と再現率を比較した検出結果を図6に示す。ただし、Liuらの手法は、[14]に基づき、著者がプログラムしたものである。

表3 各手法における検出結果

	精度 (%)	再現率 (%)	F 値 (%)
提案手法	87.0	94.4	90.5
特徴量 C,G を除外した手法	84.4	84.0	84.2
Liuらの手法	77.6	89.6	83.2

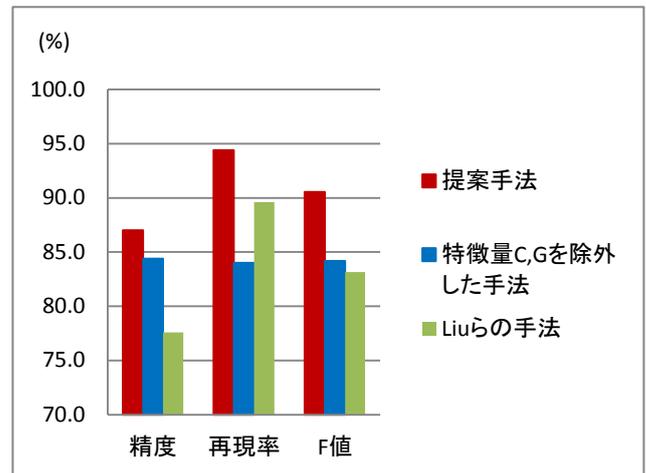


図6 各手法における検出結果の比較

表3と図6から、既存手法であるLiuらの手法と提案手法を比較すると、検出における精度は、Liuらの手法が77.6%、提案手法が87.0%となり、提案手法の方が9.4%高かった。また、再現率は、Liuらの手法が89.6%、提案手法が94.4%となり、提案手法の方が4.8%高い結果となった。F値についても、提案手法の方が7.3%高い結果となった。Liuらの手法よりも特に精度の面で大きく上回っている。つまり、Liuらの手法は、提案手法よりも劣化していないブロックに対して、劣化であると誤って判定した枚数が多いことがわかる。実際、Liuらの手法は、関連研究の問題点として挙げた平坦部のブロックに対して、誤って劣化であると判定してしまうことが多く見られた。特徴量CとGを除外した手法は、再現率に注目すると、提案手法よりも再現率が約10%低かった。つまり、提案手法よりも劣化のブロックを正しく劣化であると判定できた枚数が少ないことがわかる。

また、提案手法で正しく劣化であると判定できたブロックの中で、特徴量CとGを除外した手法が誤って劣化のないブロックであると判定した例を図7に示す。図7に示すブロックの共通点として、テクスチャが複雑な点が挙げられる。テクスチャが複雑だと、劣化が生じていてもエッジが多く検出される場合が多い。特徴量CとGを除外した手法は、検出結果がエッジの検出されやすさに大きく依存してしまう傾向がある。反対に提案手法は、段階的に縮小したブロックの情報

も特徴量として多く取り入れているため、検出結果がエッジの検出されやすさに大きく依存せず、図7のような例は正しく検出することができた。



図7 特徴量 C と G を除外した手法での判定を誤ったブロック例

4.3.2. 分割から特徴量抽出までにかかる処理時間の計測結果

入力画像分割から特徴量抽出までにかかる処理時間の計測結果を示す。画素数の異なる画像を用意し、それぞれの画像に対し、提案手法と既存手法を用いたときの処理時間の計測をおこなった。提案手法と Liu らの手法における処理時間の計測結果を表 4 に示す。そして、最後に提案手法と Liu らの手法における処理時間を比較した結果を図 8 に示す。処理時間は、それぞれの画素数ごとに 10 回実行したときの平均値を処理時間とする。ただし、キャッシュの影響がないことを考慮するために、画像の読み込みにかかる時間は除外し、実行は 1 回ごとに処理を終了させて計測する。

表4 各手法における処理時間の計測結果

画素数 (幅×高さ)	提案手法 (秒)	Liu らの手法 (秒)
640×480	0.048	0.124
960×720	0.073	0.261
1280×960	0.110	0.464
1600×1200	0.164	0.735
1920×1440	0.219	1.070
2240×1680	0.288	1.539

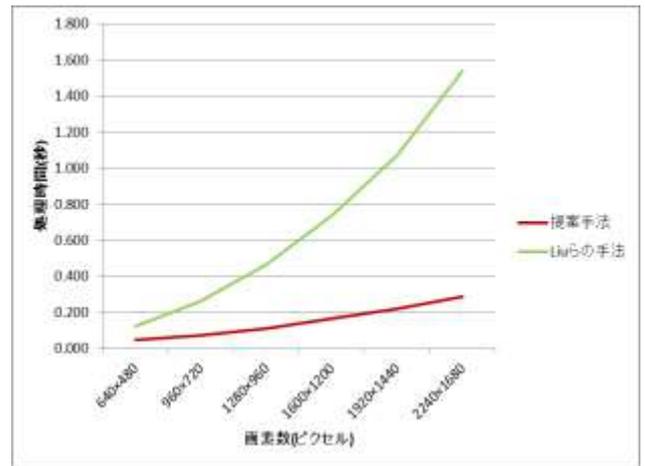


図8 各手法における処理時間の比較

図 8 より、どのサイズの画像においても、提案手法の処理時間が Liu らの手法よりも短い結果となった。特にサイズが 2240×1680 画像、つまり約 400 万画素の画像における処理時間を比較すると、提案手法の方が約 5.3 倍高速だった。また、提案手法は、画像サイズが 2240×1680 と 640×480 の処理時間の差が 0.238 秒となり、画像が大きくなって処理時間が増える割合が少ない傾向が見られた。一方、Liu らの手法は、画像サイズが 2240×1680 と 640×480 の処理時間の差が 1.434 秒となり、提案手法よりも処理時間が増える割合が多い傾向が見られた。つまり、Liu らの手法は、関連研究の問題点としても挙げたとおり、画像のサイズが大きくなるほど処理に時間がかかっている。提案手法において、画像サイズが大きくなって処理時間が短い理由としては、色情報など原画像をそのまま扱った特徴量は用いず、データ量を削減したエッジの情報などを特徴量として用いていることが考えられる。

5. PC 上と携帯端末上での処理時間の比較

本節では、提案した手法を組み込んだスマートフォン向けのカメラアプリケーション開発を行った際の処理時間について述べる。アプリケーションは、現在開発途中であり、処理時間は仮実装の段階での結果を示す。処理時間は、4.3.2 項と同様に入力画像分割から特徴量抽出までにかかる時間を PC 上で行った場合とスマートフォン上で行った場合とで比較を行う。実験に用いたスマートフォンとして Android 端末の Xperia acro HD(SO-03D)を使用した。処理時間の計測結果を表 5 に示し、計測結果を PC 上とスマートフォン上での比較した結果を図 9 に示す。

図 9 より、PC 上での処理時間と比較するとスマートフォン上での処理時間は CPU の性能が低いなどの影響により全体的に遅くなっていることがわかる。し

かし、スマートフォンでも約 400 万画素の画像に対して 2 秒程度で処理ができ、実際には撮影画像を画面サイズなどにリサイズした画像に対して処理を行うので、撮り直しを促すには十分高速であるといえる。

表5 各手法における処理時間の計測結果

画素数 (幅×高さ)	PC (秒)	SO-03D (秒)
640×480	0.048	0.317
960×720	0.073	0.451
1280×960	0.110	0.722
1600×1200	0.164	1.125
1920×1440	0.219	1.554
2240×1680	0.288	2.075

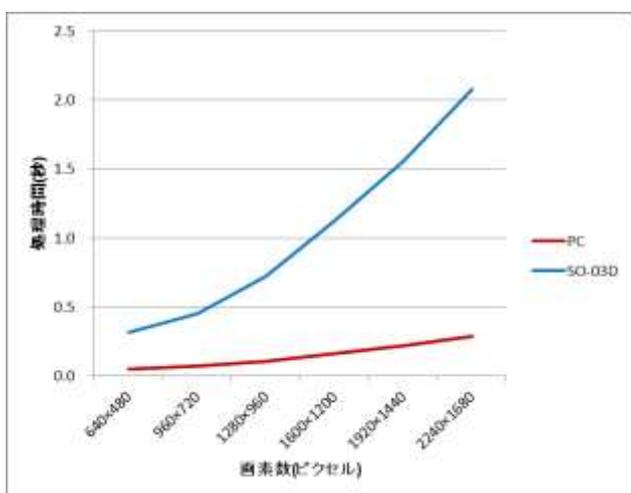


図9 PC とスマートフォン上での処理時間の比較

6. まとめ

本稿では、複数の特徴量を用いて、画像劣化検知する手法を提案した。提案手法では、画像全体の中で劣化している部分を撮影者に通知し、撮り直しを促す。撮り直すか否かの判断は撮影者に委ねられるが、画像復元などの技術に比べ、撮り直しをすることで容易に劣化の改善することができる。実験では劣化画像と劣化のない画像から生成したブロックにおける検出の精度、再現率、処理時間を求めて評価した。結果として、検出の精度は、既存の劣化検出手法である Liu らの手法が 77.6%であるのに対し、提案手法では 87.0%であった。また、再現率では、Liu らの手法が 89.6%であるのに対し、提案手法では 94.4%であった。処理時間では、約 400 万画素の画像の特徴量抽出までにかかる処理時間を約 5.3 倍高速化できた。

今後、実用可能なスマートフォンでのアプリケーション開発を行い、ユーザに使用した満足度を評価する

ような実験を行う予定である。また、劣化の検出に有効なエッジ情報以外の特徴量の検討や、画像の特性に合わせた複数特徴量の重み付け統合の検討により、更なる精度向上を目指す。

参考文献

- [1] DIGITAL PRINT LAND, “写真と画質について”, <http://www.digipri.ne.jp/photo/photo006.html>
- [2] 神崎洋治, 西井美鷹, “体系的に学ぶデジタルカメラのしくみ”, 日経 BP ソフトプレス, pp. 211-221, 2009.
- [3] All About, “手ブレ補正機構とは?”, <http://allabout.co.jp/gm/gc/54430/>
- [4] PhotoSolid, <http://www.morphoinc.com/products/PhotoSolid.html>
- [5] 加藤聡, “カメラ高画質化”, NEC 技報, Vol.61, No.2, 2008.
- [6] 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, pp.49-54, 2006.
- [7] R. Fergus, B. Singh, A. hertzman, S. T. Roweis and W. T. Freeman, “Removing camera shake from a single photograph,” ACM SIGGRAPH, Vol.25, No.3, pp.787-794, 2006.
- [8] A. Levin, “Blind motion deblurring using image statistics,” In NIPS, Vol.19, No.3, pp. 841-848, 2007.
- [9] Q. Shan, J. Jia and A. Agarwala, “High-quality motion deblurring from a single image,” ACM SIGGRAPH, Vol.27, No.3, pp.1-10, 2008.
- [10] R. Raskar, A. Agrawal and J. Tumblin, “Coded exposure photography: Motion deblurring using fluttered shutter,” ACM SIGGRAPH, Vol.25, No.3, pp.795-804, 2006.
- [11] J. Ko and C. Kim, “Low cost blur image detection and estimation for mobile devices,” In Proc. Of the 11th ICACT, Vol.3, pp. 1605-1610, 2009.
- [12] W. H. Richardson, “Bayesian-based iterative method of image restoration,” Journal of the Optical Society of America, Vol.62, pp.52-59, 1972.
- [13] L. B. Lucy, “An iterative technique for the rectification of observed distributions,” Astronomical Journal, Vol.79, pp. 745-754, 1974.
- [14] R. Liu, Z. Li and J. Jia, “Image partial blur detection and classification,” In CVPR, pp. 1-8, 2008.
- [15] D. H. Ballard, “Generalizing the Hough Transform to Detect Arbitrary Shapes,” Pattern Recognition, Vol.13, No.2, pp. 111-122, 1981.
- [16] J. Canny, “A Computational Approach to Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.8, No.6, pp. 679-698, 1986.
- [17] 奥富正敏, “デジタル画像処理”, CG-ARTS 協会, pp. 114-121, 2006.
- [18] 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>.
- [19] Flickr, <http://www.flickr.com/>
- [20] PBase.com, <http://www.pbase.com/>