

画像圧縮に適した特異値分解アルゴリズムの考察

小幡 雅彦¹ 岩崎 雅史^{2,1} 近藤 弘一³
守本 晃⁴ 芦野 隆一⁵ 中村 佳正^{1,2}

imasa@amp.i.kyoto-u.ac.jp

- ¹ 京都大学 情報学研究科 数理工学専攻 ² 科学技術振興機構 PRESTO
³ 同志社大学 工学部 電気工学科 ⁴ 大阪教育大学 教育学部 情報科学講座
⁵ 大阪教育大学 教育学部 数理科学講座

概要

これまでに画像のデータ量を圧縮する様々な方法が提案されている．ここでは，Kakarala-Ogunbona の画像圧縮アルゴリズムを取り上げる．この方法と離散ウェーブレット変換とを併用することで，すぐれた画像圧縮が可能となる．この方法には行列の多分割 (multiresolution) 特異値分解の計算過程が含まれるが，その数値的検証はこれまで十分には行われてはいない．本論文では，まず，画像のタイプによって特異値分布が大きく変動すること，ある場合には特異値がクラスタをなすことを数値的に示す．特異値の近接度が高いと既存の特異値分解ルーチンでは必ずしも特異ベクトルが高精度に求められるとは限らず，圧縮画像が原画像と大きく異なってしまう危険性がある．そこで，本論文において，特異値のクラスタを分散させ，特異値相互のギャップを拡大するアルゴリズムを提案し，数値実験によってその効果を明らかにする．

On a Singular Value Algorithm Suited to Image Compression

Masahiko Obata¹ Masashi Iwasaki^{2,1} Koichi Kondo³
Akira Morimoto⁴ Ryuichi Ashino⁵ Yoshimasa Nakamura^{1,2}

- ¹ Department of Applied Mathematics and Physics, Kyoto University
² PRESTO, Japan Science and Technology Agency
³ Department of Electrical Engineering, Doshisha University
⁴ Division of Information Science, Osaka Kyoiku University
⁵ Division of Mathematical Sciences, Osaka Kyoiku University

Abstract

Many image compression algorithms have been designed. Especially, we here consider an image compression algorithm by Kakarala-Ogunbona. A good image compression can be done by combining this algorithm with the discrete wavelet transform. The Kakarala-Ogunbona algorithm includes a process for multiresolution form of singular value decomposition (SVD). However, any numerical property of the algorithm has not been fully studied yet. In this paper first we see that the distribution of singular values depends sensitively on the type of images and singular values form clusters in some cases. It is a hard problem to compute singular vectors of such singular values with a sufficient accuracy by using the known SVD routines. This may cause a risk where a compressed image is rather different from the original one. We here propose a new algorithm which relaxes clusters and extend a gap of singular values and then show an efficiency of the algorithm by numerical experiments.



(a) 原画像



(b) 4分割画像

図 1 : ブロックアルゴリズムによる 4 分割画像

1 はじめに

ハードウェアの進歩に伴い、コンピュータやデジタルカメラ・プリンタなどの周辺機器で扱うことができる画像は、今や超精密アナログカメラの画質レベルに匹敵する。当然、画像がもつデータ量も増大するため、細部にわたる忠実さよりも表示速度が重視される場面では画像圧縮は必要不可欠な技術である。もちろん、画像圧縮によりデータ量を削減するだけでなく、原画像との違いは人間の視覚では、できる限り確認できないのが理想である。同時に、記憶領域を小さくできれば、メモリやハードディスクなどの計算機資源が有効に活用できる。

Web サイトでは、一般的に GIF や JPEG, PNG といった形式で圧縮された画像が利用されている。これらは、BMP 画像のような非圧縮画像に比べてファイルサイズが格段に抑えられるため、高いアクセス性が求められる Web サイトには適している。また、ネットワークを介して画像データを送受信する際も画像圧縮技術は重宝されることが多い。

画像は、コンピュータ内部でピクセルごとに色の濃淡が数値化される。例えば、縦 m ピクセル、横 n ピクセルの 256 階調グレースケール画像 X は、成分に $[0, 255]$ の整数値をもつ $m \times n$ 行列

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1,n} \\ x_{21} & x_{22} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix},$$

ただし、 $x_{i,j}$ は $[0, 255]$ の整数値、として表現される。以下では画像と行列を同一視して表す。RGB カラー画像は、赤 (Red)・緑 (Green)・青 (Blue) の 3 色について濃淡情報をそれぞれグレースケール画像と同様の行列形式で保持している。よって、簡単化のため、以下、グレースケール画像を対象とする。

め、以下、グレースケール画像を対象とする。

画像 X に対して 2 次元離散ウェーブレット変換、または、ブロック化 + 特異値分解を 1 度実行すれば、左上 $(m/2) \times (n/2)$ ピクセルには画像 X を $1/2$ 倍したような画像 X_1 が生成される。右上には縦方向、左下には横方向、右下には斜め方向のエッジを抽出したような画像が現れる。同様に、 X_1 の 4 分割、すなわち、 $X_1 \rightarrow$ (縮小近似画像 X_2) + (縦方向のエッジ抽出画像) + (横方向のエッジ抽出画像) + (斜め方向のエッジ抽出画像)、 X_2 を 4 分割、 \dots 、 X_k を 4 分割というように分割を繰り返せば多分割画像が作成できるが、これが画像圧縮における基本的な工程となる [1]。画像の多分割化後は、SPIHT [5] などでコーディングすればよい。なお、離散コサイン変換による画像圧縮でも X のブロック化は伴うが、ブロック特異値分解のような画像の 4 分割は行われない。量子化やハフマン符号によるコーディングが施され、データ量も小さく拡大縮小にも強い圧縮画像が得られるが、原画像を復元できない不可逆変換ということには注意が必要である [2]。

ブロック特異値分解による多分割アルゴリズムが、Kakarala-Ogunbona [3] によって提案されている。このアルゴリズムと離散ウェーブレット変換を併用したハイブリット型アルゴリズムも報告されている [1]。指紋のような特別な性質をもつ画像では、JPEG や JPEG2000 として実用化されている離散コサイン変換や離散ウェーブレット変換よりも、ハイブリット型アルゴリズムによって、より自然な画質で画像圧縮できることも知られている。ブロック特異値分解による画像圧縮には様々な可能性が秘められており、さらなる数値的な検証がなされるべきである。

本論文では、まず、Kakarala-Ogunbona アルゴリズムで多分割する際、特異値分解の対象となる行列の特異値の分布を調べる。画像によっては特異値が互いに重複および近接するが、そのような場合、既

存の特異値分解法では、精度よく特異ベクトルが計算できるとは限らない [6]。そこで、ブロック特異値分解において特異値の近接度を下げるアルゴリズムを新たに定式化する。

2章では、画像を多分割化するための Kakarala-Ogunbona のブロック特異値分解アルゴリズムについて概観する。3章において、縦、横、斜め縞模様、および、ランダム模様の画像に対して Kakarala-Ogunbona アルゴリズムを適用したとき、どのような特異値分布をもつ行列が現れるかを論じる。4章では、ブロック特異値分解アルゴリズムにおいて、特異値相互のギャップを拡大し近接度を下げる効果的な方法について提案し、その効果を数値実験で確認する。

2 Kakarala-Ogunbona のブロック特異値分解アルゴリズム

画像行列 X に対して、2次元ウェーブレット変換すれば図1のような4分割画像が得られる。縮小近似画像の4分割化を繰り返すと多分割画像が生成され、SPHITなどでコーディングすれば画像圧縮は完了する。画像の多分割化は、 X のブロック化 + 特異値分解でも実現できる。なお、離散コサイン変換も X のブロック化と併用されるが、この場合は多分割画像が生成されない。本章では、画像を多分割するための Kakarala-Ogunbona によって提案されたブロック特異値分解アルゴリズムについて概説する。Kakarala-Ogunbona アルゴリズムは任意のブロックサイズで任意サイズの長方形画像に適用できるが、その場合については [1, 3] を参照されたい。また、大きな画像は適当なサイズの複数個の画像に分けて、それぞれについて個別に多分割化してもよい。便宜上、以下ではブロックサイズを2として32ピクセル×32ピクセルの原画像 X から4分割画像 X_1 を作成する場合における Kakarala-Ogunbona アルゴリズムを説明する。

Kakarala-Ogunbona アルゴリズム

32×32行列 X を2×2小行列 $X(k, \ell)$, $1 \leq k \leq 16, 1 \leq \ell \leq 16$ に分割する。

$$X = \begin{pmatrix} X(1,1) & \cdots & X(1,16) \\ \vdots & \ddots & \vdots \\ X(16,1) & \cdots & X(16,16) \end{pmatrix},$$

$$X(k, \ell) \equiv \begin{pmatrix} x_{2k-1,2\ell-1} & x_{2k-1,2\ell} \\ x_{2k,2\ell-1} & x_{2k,2\ell} \end{pmatrix}.$$

$X(k, \ell)$ ごとに4×1の列ベクトルに変換する。

$$X(k, \ell) \mapsto x(k, \ell) \equiv \begin{pmatrix} x_{2k-1,2\ell-1} \\ x_{2k,2\ell-1} \\ x_{2k-1,2\ell} \\ x_{2k,2\ell} \end{pmatrix}.$$

列ベクトルを並べて4×256行列 T を作成する。

$$T = (T(1) \ T(2) \ \cdots \ T(16)),$$

$$T(\ell) \equiv (x(1, \ell) \ x(2, \ell) \ \cdots \ x(16, \ell))$$

T の特異値分解 $T = USV^T$ を求める。ただし、 $S = \text{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$, $\sigma_k, k = 1, 2, 3, 4$ は T の特異値、 U は4×4直交行列、 V は256×256直交行列とする。

4×256行列 $T_1 = U^T T$ を求める。

T_1 の k 行目ベクトルを16×16行列 $X_1(k)$ に変換する。

$$X_1(k) = \begin{pmatrix} t_1(k,1) & \cdots & t_1(k,241) \\ \vdots & \ddots & \vdots \\ t_1(k,16) & \cdots & t_1(k,256) \end{pmatrix},$$

$t_1(k, j) : T_1$ の (k, j) 成分。

$X_1(k)$ を並べて32×32行列 X_1 を作成する。

$$X_1 = \left(\begin{array}{c|c} X_1(1) & X_1(2) \\ \hline X_1(3) & X_1(4) \end{array} \right).$$

で得られる直交行列 U の第1列目ベクトルは、でブロック化された2×2行列 $X(k, \ell)$ の4成分を平均化するフィルタとなる。によって $X(k, \ell)$ の成分は T の列に配置されるので、のように T の左から U^T を作用させれば、 T_1 の第1行目にはブロックに含まれる4成分の平均値が並ぶ。よって、のように T_1 の第1行目を $X_1(1)$ に変換すれば、それが X の縮小近似画像となる。同様に、 $X_1(2), X_1(3), X_1(4)$ はそれぞれ X の縦、横、斜め方向のエッジが強調された画像と対応する。したがって、のように $X_1(k)$ を並べると、 X の4分割化が完了する。

3 画像の種類と特異値分布

Kakarala-Ogunbona アルゴリズムで多分割画像を作成する際、特異値分解の対象となる行列 T は画像ごとに異なり、ブロック特異値分解で求められる特

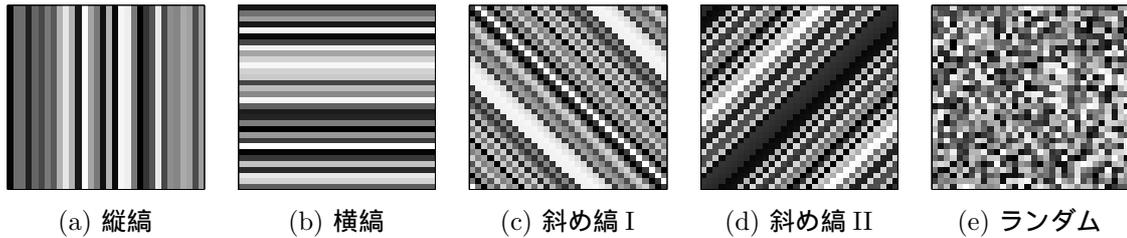


図 2 : 5 種類のテスト画像 (32 ピクセル × 32 ピクセル)

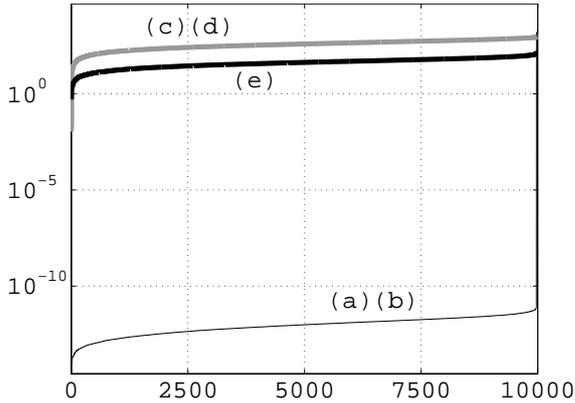


図 3 : 5 種類の画像 10000 個をそれぞれ 4 分割する際, 特異値分解の対象となる行列 T がもつ特異値相互の最小ギャップ gap_{\min} (横軸: 近接順位, 縦軸: gap_{\min})

表 1 : 5 種類の画像 10000 個をそれぞれ 4 分割する際, 特異値分解の対象となる行列 T がもつ特異値相互の最小ギャップ gap_{\min} (10000 個の画像ごとに gap_{\min} を求めた際の最大値と最小値)

	gap_{\min}	
	最大値	最小値
(a) 縦縞	2.0E + 03	1.4E - 16
(b) 横縞	2.2E + 03	2.7E - 15
(c) 斜め縞 I	1.1E + 03	0.0E + 00
(d) 斜め縞 II	9.9E + 02	0.0E + 00
(e) ランダム	1.5E + 02	5.2E - 01

異値 σ_k ($\sigma_1 \geq \sigma_2 \geq \dots$) も画像によって変動する. 特異値分解では, 特異値のギャップ

$$gap \equiv |\sigma_k - \sigma_{k+1}|$$

は特異ベクトルの精度に関する重要な指標である. すなわち, 一般に特異値の分布が密になるほど, 特異ベクトルの直交性が悪化する [6].

以下では, 図 2 の縦, 横, 斜め縞およびランダム模様の 32 ピクセル × 32 ピクセルグレースケール画像を作成して, Kakarala-Ogunbona アルゴリズムのもとで行列 X_* がもつ特異値の最小ギャップ $gap_{\min} \equiv \min_k |\sigma_k - \sigma_{k+1}|$ がどの程度になるかを調べた. 数値実験には CPU : Pentium M 1.2GHz, Memory : 504MB をもつ IBM Think Pad 上で Matlab 6.5 を使用した. なお, 行列の特異値分解は Matlab の組み込み関数 svd() によって倍精度で求めた. 図 2 のテスト画像 (a) ~ (e) は, それぞれ, 縞の濃淡と太さまたは点の濃淡をランダムに変化させて生成した. 画像である. 具体的には, 行列

$$X_a = \begin{pmatrix} x_1 & x_2 & \dots & x_{32} \\ x_1 & x_2 & \dots & x_{32} \\ \vdots & \vdots & & \vdots \\ x_1 & x_2 & \dots & x_{32} \end{pmatrix}, X_b = \begin{pmatrix} x_1 & x_1 & \dots & x_1 \\ x_2 & x_2 & \dots & x_2 \\ \vdots & \vdots & & \vdots \\ x_{32} & x_{32} & \dots & x_{32} \end{pmatrix},$$

$$X_c = \begin{pmatrix} x_{32} & \dots & x_2 & x_1 \\ \vdots & \ddots & \ddots & x_2 \\ x_{62} & \ddots & \ddots & \vdots \\ x_{63} & x_{62} & \dots & x_{32} \end{pmatrix}, X_d = \begin{pmatrix} x_1 & x_2 & \dots & x_{32} \\ x_2 & x_3 & \dots & x_{33} \\ \vdots & \vdots & \ddots & \vdots \\ x_{32} & x_{33} & \dots & x_{63} \end{pmatrix},$$

$$X_e = \begin{pmatrix} x_1 & x_2 & \dots & x_{32} \\ x_{33} & x_{34} & \dots & x_{64} \\ \vdots & \vdots & \ddots & \vdots \\ x_{993} & x_{994} & \dots & x_{1024} \end{pmatrix}$$

の成分 x_k , $k = 1, 2, \dots, 1024$ を $[0, 255]$ のランダムな整数とし, このような X_a, X_b, X_c, X_d, X_e をそれぞれ 10000 個ずつ準備した.

画像 (a) と (b), それぞれ, 10000 個の多分割化したところ, gap_{\min} の値は同じような傾向を示していて, ほとんどの場合 10^{-10} 以下となった. 表 1 のように, 最も特異値が近接する場合で gap_{\min} がマシン精度程度の極めて小さい値になることも分かった. 一方, 画像 (c) と (d) では図 3 のように, ほとんどが $gap_{\min} > 1$ であったが, 表 1 に示すように, gap_{\min} がゼロとなり行列 X_* の特異値が重複することもあった. 画像 (e) では, 概ね $gap_{\min} > 1$, かつ, 最も特異値が近接する場合でも $gap_{\min} = 0.52$ と他の画像の場合と比較して特異値が分散していることが分かった.

倍精度計算において, $gap_{\min} > 10^{-3}$ 程度ならば

直交性のよい特異ベクトルが計算できるとされている [6] . この中で、常にこの条件が満たされるのは画像 (e) のみである . つまり、画像 (a) ~ (d) のような何らかの規則性をもつ画像の場合は、Kakarala-Ogunbona アルゴリズムでは行列 X_* を精度よく特異値分解できず、その結果、美しい圧縮画像が得られない可能性がある .

4 拡大画像

原画像 X に何らかの規則性があると、Kakarala-Ogunbona アルゴリズムにおける特異値分解が高精度に実現できるかは定かでない . つまり、原画像 X より生成される行列 T の特異値相互の最小ギャップ gap_{\min} が小さければ、必ずしも美しい 4 分割画像が得られるとは限らない . 本章では、原画像 X のまわりにランダム模様のふちを追加することで規則性を崩した \bar{X} について検討する . なお、原画像サイズは 32 ピクセル \times 32 ピクセル、Kakarala-Ogunbona アルゴリズムにおけるブロックサイズを $b = 2$ とする .

まず、左側のみに 2 ピクセルのふちが付いた拡大画像を表す 32×34 行列

$$\bar{X} = \left(\begin{array}{cc|c} d_1 & d_2 & \\ d_3 & d_4 & X \\ \vdots & \vdots & \\ d_{63} & d_{64} & \end{array} \right)$$

について数学的に考察する . ただし、 d_k 、 $k = 1, 2, \dots, 64$ は $[0, 255]$ のランダムな整数とする . Kakarala-Ogunbona アルゴリズムの \sim において、 X が 4×256 行列 T に、 \bar{X} が 4×272 行列 \bar{T} に変換されるならば、 \bar{T} は T とふちの画像に対応する 4×16 行列 D によって

$$\bar{T} = (D \quad T)$$

と表現できる . ただし、 D の成分は d_k 、 $k = 1, 2, \dots, 64$ からなる . このとき、

$$\bar{T}\bar{T}^\top = DD^\top + TT^\top$$

なので、 D を変化させると $\bar{T}\bar{T}^\top$ は TT^\top と異なる固有値分布をもつことが分かる . つまり、 D をうまく制御すれば、 \bar{T} は近接した特異値をもたない行列にできる .

左側に加えて右側、上側、下側に任意サイズのふちを付けた拡大画像 \bar{X} は、Kakarala-Ogunbona アルゴリズムの \sim によって

$$\bar{T} = (D_1 \ T_1 \ D_2 \ \cdots \ D_{32} \ T_{32} \ D_{33})$$

に変換される . ただし、 D_k 、 $k = 1, 2, \dots, 32$ は上下左右いずれかのふちの画像に対応する行数 4 の行列

またはベクトルであり、 $T = (T_1 \ T_2 \ \cdots \ T_{32})$ となる . ふちが左側のみの場合と同じように

$$\begin{aligned} \bar{T}\bar{T}^\top &= DD^\top + TT^\top \\ DD^\top &\equiv \sum_{k=1}^{33} D_k D_k^\top \end{aligned}$$

が成り立つ . したがって、ふちあり画像を 4 分割の対象とすれば、 gap_{\min} の値が小さくなるような行列の特異値分解は避けられる . また、 \bar{T} の行を入れかえても特異値は変化しないことにも注意したい .

通常用いられる特異値分解アルゴリズムでは、原点シフトによって特異値の相対ギャップが拡大されるため、得られる特異値の精度は gap_{\min} の値にほとんど依存しない . dqds アルゴリズム [7] や mdLVs アルゴリズム [4] のように、高い相対精度をもつ特異値計算法もある .

画像圧縮に必要なのは特異値ではなく特異ベクトルのいくつかの成分である . このことから、特異値の相対ギャップを拡大して特異ベクトルを高精度に計算する新しいアルゴリズムが定式化できる . 以下では、画像 X の高精度な 4 分割アルゴリズムを提案する .

高精度な 4 分割アルゴリズム

Step 1 Kakarala-Ogunbona アルゴリズムの \sim によって X から T に変換する . また、 $\bar{T} \leftarrow T$ とする .

Step 2 \bar{T} の特異値を求める .

$gap_{\min} \geq \epsilon$ ならば
 \bar{T} の特異値分解 $\bar{T} = USV^\top$ を求める .

$gap_{\min} < \epsilon$ ならば
 $\bar{T} \leftarrow (D \ \bar{T})$ として Step 2 に戻る . ただし、 D は成分が $[0, 255]$ のランダムな整数で与えられた $4 \times n'$ 行列、 n' はふちの大きさに対応する .

どの特異値分解アルゴリズムを利用するかによって ϵ の値を設定すべきであるが、マシンイプシロン付近の小さな値は望ましくない .

Step 3 ふちの情報を含まない T に左から U^\top を作用させる、つまり、 $A \leftarrow U^\top T$ とする .

Step 4 Kakarala-Ogunbona アルゴリズムの \sim によって 4 分割画像 X_1 を得る .

このアルゴリズムは、特異値に近接しているものがあれば一時的にふちを付けて、特異値分解終了後には直ちにふちを取り除いて 4 分割画像を作成するという仕組みである .

最後に、提案アルゴリズム、特に Step 2 の効果を実証するための数値実験を行った . 原画像は 3 章と

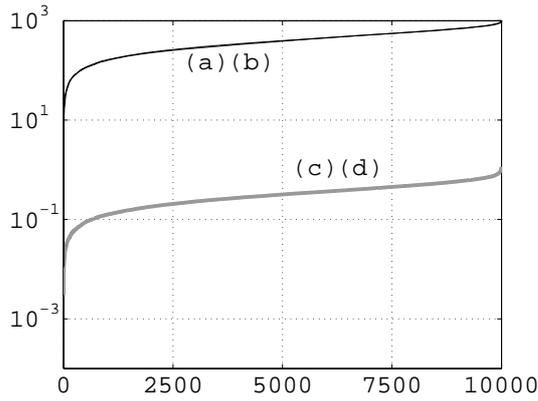


図 4：4 種類のふちあり画像 10000 個をそれぞれ 4 分割する際，特異値分解の対象となる行列 T がもつ特異値相互の最小ギャップ gap_{\min} （横軸：近接順位，縦軸： gap_{\min} ）

表 2：5 種類のふちあり画像 10000 個をそれぞれ 4 分割する際，特異値分解の対象となる行列 T がもつ特異値相互の最小ギャップ gap_{\min} の最小値

	gap_{\min} の最小値
(a) 縦縞	$7.8E - 04$
(b) 横縞	$1.5E - 03$
(c) 斜め縞 I	$5.8E - 03$
(d) 斜め縞 II	$4.1E - 03$
(e) ランダム	$3.2E - 01$

同じく図 2 の画像 (a) ~ (e)，拡大画像はそれぞれに 2 ピクセルのランダムなふちを上側と左側につけた画像とした。

画像 (a) と (b) の拡大画像それぞれ 10000 個から生成される \bar{T} では，特異値相互のギャップが $gap_{\min} > 10$ となった。行列 \bar{T} の特異値が最も近接する場合でも， gap_{\min} の値は表 2 のように (a) で 7.8×10^{-4} ，(b) で 1.5×10^{-3} となり，3 章に示したふちなしよりも特異値のクラスが分散されているのが分かる。画像 (c) と (d) の拡大画像については，多くの場合で \bar{T} の gap_{\min} が 0.1 以上となり，ふちなし画像と比較すれば行列 \bar{T} の特異値が若干近接するが，高精度に特異値分解するには全く支障のないレベルである。最低でも gap_{\min} は $O(10^{-3})$ に抑えられ， \bar{T} が重複特異値をもつ状況は回避できた。画像 (e) では，原画像に規則性がないため，ふちを付けても gap_{\min} の値が大きく変化することはなかった。

5 まとめ

原画像をもとに縮小近似画像と縦，横，斜め方向のエッジ抽出画像の 4 分割画像を作成する画像圧縮方式において，有用な手段の 1 つである Kakarala-Ogunbona のブロック特異値分解について説明した。5 種類のテスト画像を Kakarala-Ogunbona アルゴリズムによって 4 分割化したところ，規則性のある画像についてはアルゴリズムの途中で近接または重複特異値をもつ行列が現れることがあった。特異値相互のギャップが小さくなれば，特異ベクトルの精度が低下するとの報告もあるため，本論文では特異値の相対ギャップを拡大し，クラスを分散させる 4 分割アルゴリズムを定式化した。原画像にランダム模様のあるふちを追加して 4 分割化するというシンプルなアイデアではあるが，理論的にも実験的にも高精度な特異値分解が困難な状況を回避できることを示した。

提案した 4 分割アルゴリズムと SPIHT によるコーディングを組み合わせると，原画像の情報がどれほど圧縮されたか（圧縮率），原画像にどれほど忠実に圧縮されたか（PSNR, SNR 値）などは今後の別機会に報告する予定である。

参考文献

- [1] Ashino, R., Morimoto, A., Nagase, M., and Vaillancourt, R.: *Image compression with multiresolution singular value decomposition and other methods*, Math. Comput. Model., Vol.41, pp. 773–790 (2005).
- [2] 越智宏, 黒田秀夫: JPEG & MPEG 図解でわかる画像圧縮技術, 日本実業出版社 (2006)
- [3] Kakarala, R., and Ogunbona, O. P.: *Signal analysis using a multiresolution form of the singular value decomposition*, IEEE Trans. Image Process., Vol.10, pp. 724–735 (2001).
- [4] Iwasaki, M., and Nakamura, Y.: *Accurate computation of singular values in terms of shifted integrable schemes*, (submitted).
- [5] Said, A., and Pearlman, A. W.: *A new fast and efficient image codec based on set partitioning in hierarchical trees*, IEEE Trans. on Circuits and Systems for Video Technology, Vol.6, pp. 243–250 (1996).
- [6] 高田雅美, 木村欣司, 岩崎雅史, 中村佳正: 高速特異値分解のためのライブラリ開発, 投稿中.
- [7] Parlett, B. N., and Marques, O. A.: *An implementation of the dqds algorithm (positive case)*, Lin. Alg. Appl., Vol.309, pp. 217–259 (2000).