

応用力学研究所研究集会報告 No.19ME-S2  
「戸田格子 40 周年 非線形波動研究の歩みと展望」 (研究代表者 西成 活裕)

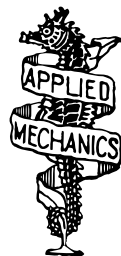
Reports of RIAM Symposium No.19ME-S2  
*40 years Anniversary of Toda lattice - history and perspective of nonlinear wave research*  
Proceedings of a symposium held at Chikushi Campus, Kyushu University,  
Kasuga, Fukuoka, Japan, November 7 - 9, 2007

Article No. 27

# 特異値分解を用いた画像圧縮方法に 関する研究

笹田 昇平 (SASADA Shohei), 近藤 弘一 (KONDO Koichi),  
岩崎 雅史 (IWASAKI Masashi)

(Received February 19, 2008)



Research Institute for Applied Mechanics  
Kyushu University  
April, 2008

# 特異値分解を用いた画像圧縮方法に関する研究

同志社大学工学部

笹田 昇平 (SASADA Shohei)

近藤 弘一 (KONDO Koichi)

京都府立大学人間環境学部

岩崎 雅史 (IWASAKI Masashi)

## 概要

特異値分解を用いた Kakarala-Ogunbona の画像分解アルゴリズムは個々の画像から分解フィルタを抽出する。そのため、画像の性質を含んだフィルタの生成が可能である。本手法の画像圧縮への応用について考察を行う。

## 1 研究背景

画像圧縮方式の業界標準に JPEG (Joint Photographic Experts Group) がある。JPEG の問題点であるブロックノイズの発生などが改良された次世代規格に JPEG2000 がある。JPEG2000 は画像に対してウェーブレット変換 (2DWT) で画像分解を行う。2DWT は窓フーリエの一種で決まった基底関数で信号の展開を行う。分解画像は 1 つの低周波画像と 3 つの高周波画像で構成される。その低周波画像に対して再度 2DWT で画像分解を行う。分解を繰り返し多重分解画像を得る。多重分解画像に対して、SPIHT (Set Partitioning in Hierarchical Trees), EBCOT (Embedded Block Coding with Optimal Truncation) などの圧縮符号化を用いることで画像圧縮が可能となる。JPEG2000 は JPEG に比べて 30% から 50% 圧縮率が高い。一方、特異値を用いて画像分解を行うものに Kakarala-Ogunbona(KO) 分解がある。KO 分解はまず、元画像を並べ替えて横長行列を作る。横長行列に対して SVD 計算を行う。得られた左直交行列を横長行列の左から作用させる。変換された横長行列を逆の操作で並べ替えて分解画像を得る。分解画像に対して再度 KO 分解を行う事で、多重分解画像解析が可能となる。KO 分解は特異値分解 (SVD) の主成分分析を用い、決まった基底関数を持たず SVD のスペクトル分解で展開を行う。また、2 以上の任意の整数の二乗のサブバンドに分解が可能である。そのため、詳細な画像分解が可能であることが期待される。非可逆画像圧縮の手順は図 1 のように分解画像に対して量子化、符号化を行うことである。しかし、KO 分解で生成された分解画像に対する量子化、符号化はまだ開発されていない。本論では、KO アルゴリズムに対する量子化についての考察を行う。数値実験にて、2DWT と KO 分解に対する量子化の精度の比較を行う。また、KO 分解でのフィルタの性質はまだ理解されていない。そのため、数値実験によりフィルタに対する考察を行う。実験に用いる評価式は PSNR (Peak Signal-to-Noise Ratio) と MSSIM (Mean Structural SIMilarity) を用いる。PSNR は、画像の誤差を基に画像評価を行う。しかし、PSNR の値は人間の視覚とあわないときがある。そのため、近年開発された MSSIM という評価式を用いる。MSSIM は画像の誤差に加え、輝度、構造、コントラストで評価を行う。MSSIM の結果は人間の視覚に近いと言われている。より正確な画像評価をするために性質の違う評価式を用いる。



図 1: 非可逆画像圧縮のフローチャート

## 2 Kakarala-Ogunbona の画像分解アルゴリズム

画像を縦  $M$  ピクセル, 横  $N$  ピクセルの 256 階調グレースケールとすると, 画像は  $M \times N$  型行列  $X = (X_{i,j})$  ( $i = 0, 1, \dots, M-1; j = 0, 1, \dots, N-1$ ) で表される. ただし, 各成分は  $[0, 255]$  の整数値である.  $X$  に対する Kakarala-Ogunbona の画像分解 (KO 分解) はブロックアルゴリズムに基づき構成される [1,2]. まず, 行列  $X = (X_{i,j})$  を  $b \times b$  小ブロック行列に分割する. ただし,  $M, N$  は  $b$  で割り切れるとする. 各  $b \times b$  型小ブロックの第  $(k, l)$  成分をすべて並べて  $M' \times N'$  型ブロック行列  $X^{(k,l)} = (X_{m,n}^{(k,l)})$  ( $m = 0, 1, \dots, M'-1; n = 0, 1, \dots, N'-1$ ) とする. ここで  $M' = M/b, N' = N/b$  とおく. このとき  $X$  から  $X^{(k,l)}$  への変換は  $k = 0, 1, \dots, b-1; l = 0, 1, \dots, b-1$  に対して

$$\mathcal{B}_b^{(k,l)} : X \mapsto X^{(k,l)}; \quad X_{m,n}^{(k,l)} = X_{bm+k,bn+l} \quad (1)$$

と表される. ブロック行列  $X^{(k,l)}$  をすべて並べて  $M \times N$  型行列  $Y = (Y_{i,j})$  とする. 元の行列  $X$  から  $Y$  への変換は

$$\mathcal{B} : X \mapsto Y = (X^{(k,l)})_{0 \leq k,l < b}; \quad Y_{M'k+m,N'l+n} = X_{m,n}^{(k,l)} = X_{bm+k,bn+l}$$

である. また,  $M' \times N'$  型ブロック行列  $X^{(k,l)}$  のすべての要素を並び換えて  $1 \times M'N'$  型行ベクトル  $\mathbf{a}^{(bk+l)} = (a_i^{(bk+l)})$  ( $t = 0, 1, \dots, M'N'-1$ ) とする. この変換を

$$\mathcal{R} : X^{(k,l)} \mapsto \mathbf{a}^{(bk+l)}; \quad a_{N'm+n}^{(bk+l)} = X_{m,n}^{(k,l)} \quad (2)$$

とおく. すべての行ベクトル  $\mathbf{a}^{(s)}$  ( $s = 0, 1, \dots, b^2-1$ ) を縦に積み上げて  $b^2 \times M'N'$  型行列  $A = (A_{s,t})$  とする. このとき  $Y$  から  $A$  への変換は

$$\mathcal{P} : Y \mapsto A = (\mathbf{a}^{(s)})_{0 \leq s < b^2}; \quad A_{bk+l,N'm+n} = a_{N'm+n}^{(bk+l)} = Y_{M'k+m,N'l+n}$$

である. 元の行列  $X$  から  $A$  への変換は

$$\mathcal{K} = \mathcal{P} \circ \mathcal{B} : X \mapsto A; \quad A_{bk+l,N'm+n} = X_{bm+k,bn+l}$$

となる. 得られた横長行列  $A$  に対して特異値分解 (SVD) を行う.  $A$  の特異値を  $\sigma_0, \sigma_1, \dots, \sigma_{b^2-1}$  ( $\sigma_s \geq \sigma_{s+1}$ ), 左特異ベクトルを  $\mathbf{u}_s$  ( $s = 0, \dots, b^2-1$ ), 右特異ベクトルを  $\mathbf{v}_t$  ( $t = 0, 1, \dots, M'N'-1$ ) とおく. このとき,

$$\begin{aligned} A &= USV^T, \\ S &= \begin{pmatrix} \Sigma & O \end{pmatrix}, \Sigma = \text{diag}(\sigma_0, \sigma_1, \dots, \sigma_{b^2-1}), \\ U &= \begin{pmatrix} \mathbf{u}_0 & \cdots & \mathbf{u}_{b^2-1} \end{pmatrix}, V = \begin{pmatrix} \mathbf{v}_0 & \cdots & \mathbf{v}_{M'N'-1} \end{pmatrix} \end{aligned} \quad (3)$$

と表される. ただし,  $U$  は  $b^2 \times b^2$  型直交行列,  $V$  は  $M'N' \times M'N'$  型直交行列であり,  $\Sigma$  は  $b^2 \times b^2$  型対角行列,  $S$  は  $b^2 \times M'N'$  型長方形行列である. 得られた左直交行列  $U$  の転置を横長行列  $A$  に左から掛けて  $\hat{A} = U^T A$  を求める.  $\hat{A}$  は  $A$  と同じ型の長方形行列である. 得られた  $\hat{A}$  は (3) より

$$\hat{A} = SV^T = \left( \sigma_0 \mathbf{v}_0 \quad \sigma_1 \mathbf{v}_1 \quad \cdots \quad \sigma_{b^2-1} \mathbf{v}_{b^2-1} \right)^T \quad (4)$$

とも表される.  $\hat{A}$  を  $\mathcal{P}$  と逆の操作で並び換えて

$$\begin{aligned} \hat{X} &= \mathcal{P}^{-1}(\hat{A}) = \left( \sigma_{bk+l} \mathcal{R}^{-1}(\mathbf{v}_{bk+l}^T) \right)_{0 \leq k, l < b} \\ &= \begin{pmatrix} \sigma_0 \mathcal{R}^{-1}(\mathbf{v}_0^T) & \cdots & \sigma_{b-1} \mathcal{R}^{-1}(\mathbf{v}_{b-1}^T) \\ \sigma_b \mathcal{R}^{-1}(\mathbf{v}_b^T) & \cdots & \sigma_{2b-1} \mathcal{R}^{-1}(\mathbf{v}_{2b-1}^T) \\ \vdots & \ddots & \vdots \\ \sigma_{b^2-b} \mathcal{R}^{-1}(\mathbf{v}_{b^2-b}^T) & \cdots & \sigma_{b^2-1} \mathcal{R}^{-1}(\mathbf{v}_{b^2-1}^T) \end{pmatrix} \end{aligned} \quad (5)$$

となる.  $\hat{X}$  は  $M' \times N'$  型の小ブロック行列  $\sigma_s \mathcal{R}^{-1}(\mathbf{v}_s^T)$  により分割された行列である. 左上のブロック  $\sigma_0 \mathcal{R}^{-1}(\mathbf{v}_0^T)$  は第 1 主成分であり低周波画像となる. 残りの  $b^2 - 1$  個のブロック行列は第 2 主成分以降であり高周波画像となる.  $\hat{X}$  が KO アルゴリズムによる分解画像となる. これらの操作をまとめると KO 分解は次のアルゴリズム 1 と定義される. また, KO 分解の再構成は  $X = \mathcal{K}^{-1}(U\mathcal{P}(\hat{X}))$  である.

#### アルゴリズム 1 (Kakarala-Ogunbona の画像分解)

**Input.** 元画像  $X$ .

**Output.** 分解画像  $\hat{X}$ .

**Step 1.**  $A = \mathcal{K}(X)$  を求める.

**Step 2.**  $A$  を特異値分解  $A = USV^T$  する.

**Step 3.**  $\hat{A} = U^T A$  を求める.

**Step 4.**  $\hat{X} = \mathcal{P}^{-1}(\hat{A})$  を求める.

分解画像の低周波画像に対してさらに繰り返し KO 分解を行うと多重分解画像を得る. 得られた多重分解画像に対して圧縮符合化を行えばデータサイズの圧縮が行われる. 圧縮画像を復元する場合は逆の操作を行う. KO 分解専用の圧縮符合化はまだ提案されていない.

例えば図 2 (b) の画像 lena に対して  $b = 2$  の KO 分解をする. 特異値は

$$\sigma_0 = 6.7 \times 10^4, \quad \sigma_1 = 2.7 \times 10^3, \quad \sigma_2 = 1.8 \times 10^3, \quad \sigma_3 = 1.0 \times 10^3 \quad (6)$$

であり, 左直交行列  $U$  は

$$U = \begin{pmatrix} 0.4998 & 0.5805 & 0.4035 & -0.5004 \\ 0.5003 & -0.4054 & 0.5810 & 0.4978 \\ 0.4998 & 0.4045 & -0.5784 & 0.5021 \\ 0.5001 & -0.5788 & -0.4064 & -0.4997 \end{pmatrix} \quad (7)$$

となる. 第 1 列目の左特異ベクトル  $\mathbf{u}_0$  は平均化フィルタ  $\sigma_0 \mathbf{v}_0^T = (\mathbf{a}^{(0)} + \cdots + \mathbf{a}^{(3)})/2$  として機能する. 第 2 列目以降の左特異ベクトルは高周波フィルタとして機能する.

### 3 Kakarala-Ogunbona アルゴリズムに対する量子化

非可逆画像圧縮をするとき図 1 より，分解画像に対して，量子化，符号化をする必要がある．しかし，KO 分解に対する量子化，符号化は十分研究されていない．そのため，本節では KO 分解に対する量子化方法の提案を行う．KO 分解ではサブバンド数を自由に変えることができる．KO 分解で生成されたサブバンドはエネルギーが大きい順に並んでおり，エネルギーの大きいものに多くのビットを与えることで量子化が可能である．量子化の手順は次の通りである．

#### KO 分解に対する量子化方法

**Input.** 分解画像  $\hat{X}$ .

**Output.** 量子化画像  $\bar{X}$ .

**Step 1.**  $\hat{X}$  のサブバンドの中で最大のエネルギーのものの画素ごとに 8 ビット与える．他のサブバンドは 0 ビットとする．

**Step 2.**  $\hat{X}$  のサブバンドに順に任意の bpp になるまで同じ操作を繰り返す．

**Step 3.** 再構成を行い， $\bar{X}$  を得る．

以上が KO 分解に対する量子化の手順である．本手法の精度を実験にて確認する．実験に用いる画像は図 2 (a) である．本画像に対して，3 種類の量子化方法を行う．まず 1 つめに，提案手法である．2 つめに，全画素に均等にビットを与える手法である．3 つめに，ビットをサブバンド毎で差をつける手法である．3 種類の量子化の精度を実験にて確認する．実験結果を図 4, 5 に示す．図 4, 5 より MSSIM, PSNR 共に提案手法が最も精度が良い．KO 分解は特異値分解の主成分分析を用いているためにサブバンド数を多くするとエネルギーが低周波部分にかたよる．そのため，エネルギーが多いところに多くのビットを与える提案手法が最も良くなると考えられる．



(a) mandrill

(b) lena

図 2: 実験に用いる画像

### 4 数値実験 1 (量子化実験)

KO 分解で生成された分解画像を画像圧縮に用いるとき，量子化，符号化をする必要がある．KO 分解は特異値分解を用いて画像分解を行う．特異値分解はある種の主成分分析であるため，画像のエネルギーの順にサブバンドが並ぶ．本節では KO 分解で生成された多重分解画像に対して量子化を行う．提案手法の効果を数値実験にて確認する．実験の手順は次の通り

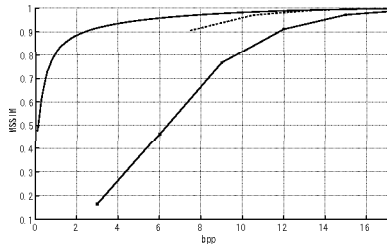


図 4：各手法の量子化の精度  
 実線：提案手法，点線：2 の手法，  
 四角：3 の手法

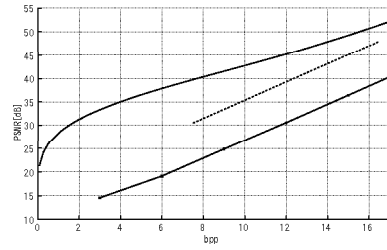


図 5：各手法の量子化の精度  
 実線：提案手法，点線：2 の手法，  
 四角：3 の手法

である．元画像に対して KO 分解を用いて多重分解画像を得る．多重分解画像に対して提案手法を用いて量子化を行う．量子化された画像を逆の操作で復元し，復元画像を得る．画像の精度の判断として，PSNR 及び MSSIM を用いる．また，2DWT で生成された分解画像に対する量子化と精度の比較を行う．量子化の結果を図 6 から図 9 に示す．PSNR，MSSIM では 2DWT より  $b$  が大きいとき KO 分解での量子化のほうが精度がよい．サブバンド単位でビットを決めているために， $b$  の値を大きくすることで量子化が有効になる．KO 分解はサブバンド数が多くなるにつれてエネルギーの分布が集中する．エネルギーの分布が集中するとき，提案手法は効果が高い．KO 分解に対する量子化は容易であり，高い効果があることを示した．

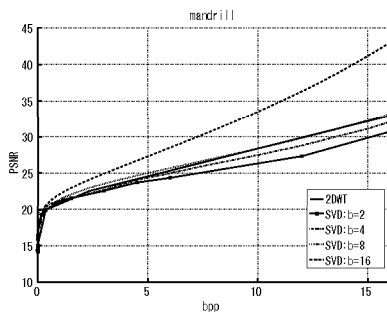


図 6：量子化実験 (mandrill PSNR)

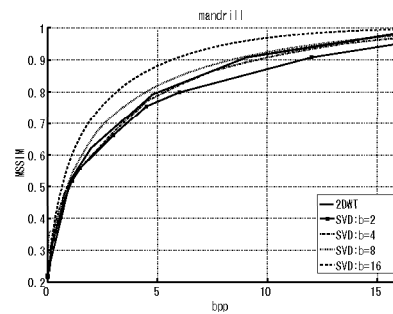


図 7：量子化実験 (mandrill MSSIM)

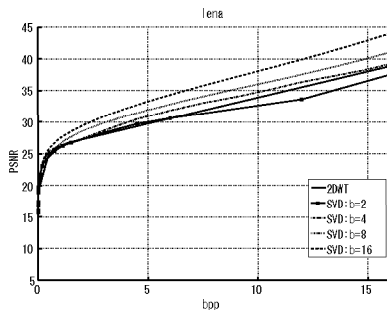


図 8：量子化実験 (lena PSNR)

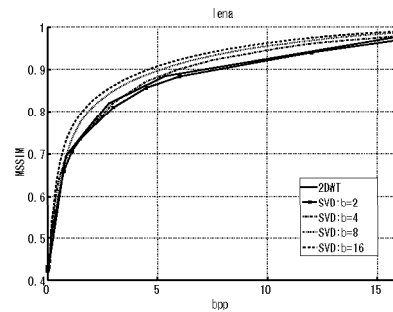


図 9：量子化実験 (lena MSSIM)

## 5 数値実験 2 (フィルタの保存バイト数に関する実験)

本節ではフィルタの性質について考察を行う。KO 分解で生成されるフィルタの性質に関して研究はされていない。そのため、本節では数値実験によりフィルタの性質を検証する。フィルタ  $U$  の量子化を行う。本論のすべての計算は倍精度浮動小数点 (double 型, 8 バイト) で計算している。画像の再構成の際に  $U$  を double 型から 4 つの整数型 (int32, int16, int8, それぞれ 4, 2, 1 バイト) に型変換し量子化する。また,  $b$  を 2, 4, 8, 16, 24, 32 と変化させて比較する。このとき圧縮画像に関する誤差評価を行う。実験には図 2(a), 2(b) の画像を用いる。結果を表 1, 2 に示す。int32, int16 は PSNR, MSSIM で画像の劣化はない。一方, int8 では PSNR, MSSIM 共に  $b$  が大きくなるにつれて悪化する。よって, フィルタ係数  $U$  を整数型 2 バイト (int16) に型変換することで  $U$  のバイト数は 25% に削減することが可能である。また, この結果は  $U$  の結果精度は少なくとも 2 バイトあればよいことを示している。各種のチップは浮動小数点 4 バイトを用いているため, 本手法をチップ化する時にも有益な結果である。

表 1: フィルタの保存バイト数 (mandrill, MSSIM)

mandrill/b	2	4	8	16	24	32
int8	1.0	1.0	0.98	0.95	0.95	0.85
int16	1.0	1.0	1.0	1.0	1.0	1.0
int32	1.0	1.0	1.0	1.0	1.0	1.0

表 2: フィルタの保存バイト数 (mandrill, PSNR)

mandrill/b	2	4	8	16	24	32
int8	45	40	34	28	26	22
int16	誤差なし	誤差なし	誤差なし	誤差なし	誤差なし	誤差なし
int32	誤差なし	誤差なし	誤差なし	誤差なし	誤差なし	誤差なし

## 6 まとめ

本論では, KO 分解の量子化の精度の高さを数値実験にて示した。量子化は 2DWT に比べて 5~10dB 良くなる。2DWT はフィルタが固定であるため画像の性質を含んだ画像分解が不可能であるが, KO 分解はフィルタを画像から生成するため高精度な分解が可能である。また, KO 分解は  $b^2$  の多くのサブバンドに分解可能なため, 2DWT に比べ詳細な分解が可能である。誤差なしでのフィルタの保存は 2 バイトが適切である。

## 参考文献

- [1] R. Ashino, A. Morimoto, M. Nagase, and R. Vaillancourt, Image compression with multiresolution singular value decomposition and other methods, *Math. Comput. Model.* **41** (2005), 773–790.
- [2] R. Kakarala and O. P. Ogunbona, Signal analysis using a multiresolution form of the singular value decomposition, *IEEE Trans. Image Process.* **10** (2001), 724–735.