画像符号化のための線形変換

正会員 山下 幸彦[†]

1. ま え が き

静止画や動画などの画像データはテキストデータなどと比べて、データ量が非常に多い。したがって、画像符号化では、画像をある一定の画質に保ちながら、保存や伝送のためのデータ量が少ない、すなわち、符号化効率が高い方式を開発することが重要である。そのため画像符号化では、数値をビット列に変換するときにデータ量を削減するエントロピー符号化（ハフマン符号化や算術符号化）の他に、予測や、画像の輝度信号や色差信号または予測誤差の集まりに対して、線形変換を行うことによってデータ量を削減する。画像符号化のための線形変換としては、変換の重要性を広く知らしめた楨本らによるスラント変換¹⁾、現在の標準的変換である離散コサイン変換 (Discrete Cosine Transform: DCT)²⁾、ブロック歪みが原理的に生じない離散ウェーブレット変換 (Discrete Wavelet Transform: DWT)^{3) 4)}などが知られている。DCTは、画像符号化の標準方式であるJPEG, MPEG-1, MPEG-2, MPEG-4, H.264/MPEG-4AVCに採用されており、現在の画像符号化の中心的な変換となっている。DWTは、静止画像を最も高い効率で符号化できるJPEG2000で採用されている。本稿では、第2章で線形変換によってデータ量を削減できる理由とKarhunen-Loève変換を説明し、第3章で離散フーリエ変換 (DFT) からDCTを導出するとともにDCTを説明し、第4章でDWTを解説する。そして、第5章で本稿をまとめる。

2. 線形変換によるデータ量の削減

2.1 平均情報量の削減

一つの連続的な値 x を符号化することを考える。ここでは、 x は平均 μ 、分散 σ^2 の正規分布に従うものとする。確率密度関数は、

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)} \quad (1)$$

で与えられる。連続的な値は無限の情報量を持っているため、符号化することはできない。そこで、本講座第4回で解説する量子化を行う。ここでは最も単純な量子化である、量子化ステップ q に対して、 x が区間

$$I_k = \left(\left(k - \frac{1}{2}\right)q + \mu, \left(k + \frac{1}{2}\right)q + \mu \right] \\ (k = 0, \pm 1, \pm 2, \dots)$$

に含まれている場合、量子化値を k とするものを考える。そして、この k をビット列に変換する。復号器で、ビット列から量子化値 k を得ることはできるが、元の値 x を厳密に得ることはできない。ここでは、 x の復号値を $qk + \mu$ とする。この操作を逆量子化と呼ぶ。このとき、量子化・逆量子化による誤差は、 $\pm \frac{q}{2}$ 程度であるので、 q を充分小さくすれば、実用的に x を符号化することができる。

いま、量子化値が k となる確率を $P(k)$ とすれば、

$$P(k) = \int_{I_k} p(x) dx$$

となる。 q が充分小さい場合、 $x \in I_k$ ならば $p(x)$ を $p(qk + \mu)$ で一定と近似できるため、

$$P(k) \simeq qp(qk + \mu) \quad (2)$$

となる。次に、このビット列の平均的な長さ、すなわち、平均的データ量を最小にすることを考える。そのための条件は、シャノンの情報源符号化定理によって与えられており、量子化値 k に対して、 $-\log_2 P(k)$ 程度の長さのビット列を割当てればよい。したがって、量子化値の平均的なデータ量 (ビット) は、

$$H = - \sum_{k=-\infty}^{\infty} P(k) \log_2 P(k) \quad (3)$$

となる。この H は、エントロピー (平均情報量) と呼ばれる。ここでは、量子化値に対する最適な符号化が行われるものとし、 H を符号化によって出力されるデータ量と考える。式 (2) より、

$$H \simeq - \sum_{k=-\infty}^{\infty} qp(qk + \mu) \log_2 (qp(qk + \mu)) \quad (4)$$

が成立する。 q が充分小さいとして、この和を積分で近似する。 $y = qk$ とおき、式 (4) の最初の q を dy と考えることができるので、 H は次式で近似できる。

[†] 東京工業大学 大学院理工学研究科

"Basic Technologies of Image Coding (3): Arithmetic Coder" by Yukihiro Yamashita (Graduate School of Science and Engineering, Tokyo Institute of Technology, Tokyo)

$$\begin{aligned}
 H &\simeq - \int_{-\infty}^{\infty} p(y+\mu) \log_2(qp(y+\mu)) dy \\
 &= \int_{-\infty}^{\infty} p(y+\mu) \left(\frac{y^2}{2\sigma^2} - \log_2 q + \log_2 \sigma \right. \\
 &\quad \left. + \frac{1}{2} \log_2(2\pi) \right) dy \\
 &= \frac{1}{2} \log_2 \sigma^2 - \log_2 q + \frac{1}{2} (\log_2(2\pi) + 1)
 \end{aligned} \tag{5}$$

すなわち、一つの連続値を符号化したデータ量は、分散の対数の1/2から量子化ステップの対数を引き、ある定数を加算したものになる。画像は、画素における明るさなどの値(画素値)が複数集まったものである。カメラの方向は任意であるため、画素の位置によって、画素値の確率的な構造は変わらないと仮定できる。画素値をそのまま並べて符号化すれば、データ量は式(5)に画素数をかけたものになる。画像を見ると、隣接する二つの画素値はあまり変わらないことが多いことがわかる。図1は、この状況を図示したものである。 s_1 と s_2 が二つの画素値を表し、図1の黒点が1組のデータ(s_1, s_2)を表している。そして、図1の黒点の分布は、 s_1 と s_2 が近い値をとることが多いことを示している。このとき、(s_1, s_2) - 座標系を使って表す場合と、それを回転させた(t_1, t_2) - 座標系を使って表す場合とでは、データ量はどちらが少なくなるであろうか。以下では、その優劣に関して数学的に説明する。

画像の扱いを容易にするために、画像の N 個の画素値を一列に並べて、 N 次元列ベクトル $\mathbf{f}$ で表す。そして、 $\mathbf{f}$ を変換することを考える。変換として、ここでは線形変換だけを考える。線形変換とは、あるベクトルに定数をかけたものの変換結果が、もとのベクトルの変換結果に同じ定数をかけたものになり、二つのベクトルの和の変換結果がそれぞれの変換結果の和になるものである。図1の座標回転も線形変換によって表すことができる。ベクトルの要素の2乗や異なる要素の乗算をするような変換は、非線形変換と呼ばれる。一般に、非線形変換を使えば線形変換に限るよりもデータ量を少なくすることができるが、その最適な非線形変換を求めることも実装することも難しいため、本稿では扱わない。

線形変換は、 $\mathbf{f}$ に行列 U をかけることによって実現できる。変換後のベクトルの各成分を量子化するときの量子化

ステップはすべて同じ値 q とする。量子化ステップの値を変えることは、変換後のベクトルの各成分にそれぞれ定数をかけることと同じであり、一つの線形変換で表すことができるため、このように仮定しても一般性を失わない。画像を復号するときは、逆量子化で得たベクトルを逆変換して画像を得る。このとき、逆量子化で得たベクトルの各要素には、原画像を変換したベクトルの要素に対して0から $\pm q/2$ の誤差が生じる。ここでは、簡単のため、すべての逆量子化で得たベクトルが一定の誤差を持っていると仮定する。画像符号化の目的は、復号画像の原画像に対する誤差を抑えながら、データ量を削減することである。そのためには、誤差が原画像によって変化しない方がよい。したがって、原画像によって復号画像の誤差が変わらない線形変換を考える。

実際の画像符号化では、下に述べるように、人間の視覚特性を考慮するため、原画像によって誤差は異なる。しかしながら例えば、前処理を施すことによって許容される誤差が原画像によって変わらないようにできるため、原画像によって誤差が変わらない変換に限ることができる。

誤差が原画像によって変わらないようにするためには、 $U\mathbf{f}$ のノルムが $\mathbf{f}$ のノルムの定数倍である必要がある。この条件を満たす U は、直交(実ユニタリ)行列の定数倍となるが、定数倍は量子化係数を変えること等価であるので無視して、変換行列 U は直交行列であるものとする。単位行列を I で表すとき、直交行列 U は、 $UU^T=U^TU=I$ を満たす行列として定義され、ベクトルの回転(画像の回転ではなく)や対称操作を表す。

$\mathbf{f}$ に関する期待値操作を E_f で表す。 $\mathbf{f}$ の平均 $E_f[\mathbf{f}] = \bar{\mathbf{f}}$ で表す。また、ベクトルや行列に対する転置を $\cdot^T$ で表す。 $\mathbf{f}$ と $\bar{\mathbf{f}}$ の第 i 成分を、それぞれ f_i と $\bar{f}_i$ ($i=1, 2, \dots, n$)で表せば、 $\mathbf{f}$ の共分散行列 R の第 (i, j) -成分 R_{ij} は、

$$R_{ij} = E_f(f_i - \bar{f}_i)(f_j - \bar{f}_j) \tag{6}$$

で定義される。 R_{ii} は $\mathbf{f}_i$ の分散であり、 R_{ij} ($i \neq j$)は $\mathbf{f}_i$ と $\mathbf{f}_j$ の共分散である。共分散 R_{ij} は $\mathbf{f}_i$ と $\mathbf{f}_j$ が一緒に大きくなったり小さくなるとき正の大きな値をとり、 $\mathbf{f}_i$ と $\mathbf{f}_j$ がばらばらに変動するとき0に近い値をとる。したがって、画像上で二つの画素が近い場合、それらの画素値の共分散は正の大きな値をとり、離れると0に近い値になる。

分散 R_{ii} を使えば、ベクトル $\mathbf{f}$ を符号化したデータ量は、それぞれ要素の分散に対する式(5)の N 個の和

$$\frac{1}{2} \sum_{i=1}^N \log_2 R_{ii} - N \log_2 q + \frac{N}{2} (\log_2(2\pi) + 1)$$

となる。行列 U で変換したベクトルの共分散行列は、 $R' = URU^T$ となる。したがって、変換しない場合を基準にした変換を行ったことによるデータ量の変化は、

$$\frac{1}{2} \sum_{i=1}^N \log_2 (R')_{ii} - \frac{1}{2} \sum_{i=1}^N \log_2 R_{ii} \tag{7}$$

となる。したがって、この値が小さくなるように U を選べばよい。また、 tr でトレース(対角和)を表せば、トレース

図1 平均情報量の削減

は可換であるから

$$\sum_{i=1}^N (R')_{ii} = \text{tr}[URU^T] = \text{tr}[RU^T U] = \text{tr}[R] = \sum_{i=1}^N R_{ii} \quad (8)$$

となり、分散の和は直交変換によって変わらない。式(8)の条件のもとに、式(7)を小さくするためには、 $\log_2$ 関数は上に凸な関数であるから、変換 U によって得られるベクトルの各成分の分散が均一であるよりも、「できるだけ少数の成分の分散を大きくし、残りの成分の分散を小さくした方が良い。」すなわち、図1の場合は、 (t_1, t_2) で表す方が、 (s_1, s_2) で表すよりもデータ量を小さくすることができる。

直交行列 U の中で、式(7)を最小にするものは、Karhunen-Loève変換で与えられる。これは、 U の列ベクトルが、共分散行列 R の固有ベクトルとなっている場合である。ここで、Karhunen-Loève変換を使った場合のデータ量の変化を計算する。先に述べたように、画像の近隣の画素間の共分散は大きい。ここでは、1次元信号を考え、共分散行列が

$$R_{ij} = \rho^{|i-j|} \quad (9)$$

の形で与えられるとし、 $\rho=0.95$ とする。これは、隣の画素値との共分散が0.95、その隣の画素値とは0.95²で、遠くなるほどべき乗で共分散が小さくなることを意味している。 $N=64$ とすれば、式(7)の値は、-50.4ビットとなり、データ量を1画素あたりの平均で50.4/64=0.79ビット程度削減することができる。

2.2 視覚特性上重要でない情報の削減

線形変換を使えば、人間視覚に対して重要な部分と重要でない部分を切り分けることができる。一般に、画像におけるゆっくりとした変化を表す画像の低周波成分の誤差は、すぐに知覚されるが、激しい変化を表す高周波成分の誤差は知覚されにくい。また、量子化係数を小さくすれば誤差が小さくなるが、量子化後の値やその分散が大きくなり、式(5)よりデータ量が増加する。逆に、量子化係数を大きくすれば誤差は大きくなるが、データ量は減少する。したがって、低周波成分に対して量子化係数を小さく、高周波成分に対して量子化係数を大きくすれば、データ量を保ちながらも人間が見たときの画質を向上させることが可能になり、符号化効率が向上する。

分散の偏りと視覚特性とは別のものであり、そのための線形変換も異なるものであるべきであるが、現在の画像符号化では、空間周波数をもとにした同じ変換を用いている。そのため、先に述べた前処理による許容誤差を原画像によらなくすることと、量子化ステップを変えることが等価になる。

3. 離散コサイン変換 (DCT)

3.1 離散フーリエ変換 (DFT)

N 個の値からなる1次元信号 $x(m)$ ($m=0, 1, \dots, N-1$)を考える。この信号をDFTした結果を $X(u)$ ($u=0, 1, \dots, N-1$)とおけば、DFTと逆DFTは、

$$X(u) = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} x(m) e^{-\frac{2\pi i}{N} mu} \quad (10)$$

$$x(m) = \frac{1}{\sqrt{N}} \sum_{u=0}^{N-1} X(u) e^{\frac{2\pi i}{N} mu} \quad (11)$$

と表される。オイラーの公式 $e^{i\theta} = \cos\theta + i\sin\theta$ より、DFTは原信号を周期が N の整数分の1の $\sin$ 関数と $\cos$ 関数で分解していることがわかる。

上の式で、 u が小さい/大きい場合の $X(u)$ は、信号の低/高周波成分の大きさを表している。一般に、画像の低周波成分は大きく、その分散も大きい。逆に、高周波成分は小さく、その分散も小さい。したがって、DFTによって、信号の分散を低周波成分に集中することができるため、もとの画素値を符号化するよりもデータ量を小さくすることができる。

DFTは信号を周波数成分に分解することができるが、画像符号化の線形変換に使われることはほとんどない。その理由を説明する。信号の性質が画像のどの部分でも同じ、すなわち、ある点の値の、その周囲の点の値に対する条件付き確率密度が一定、あるいは、信号が定常である場合、変換する信号の長さ N はできるだけ大きくした方が符号化効率が高くなる。しかしながら、一般に画像は定常ではなく、また N を大きくすると計算量が大きくなるという問題のため、画像符号化では N を比較的に小さな値に設定する。例えば、 N として4や8という値が使われる。この場合長い信号から長さ N の部分の切出して、線型変換を施すことになる。例えば、図2(a)から切出した、図2(b)の信号を変換していることになる。このとき、DFTは図2(c)の周期信号を変換していると考えることができる。この信号には、元の信号には存在しなかった、大きな段差が生じている。この原因は、切出した信号の右端と左端における値が違うためである。したがって、DFTを用いると切出しによって実際には存在しない高周波成分が生じ、低周波分に信号の分散が集中しなくなるため、符号化効率が低下する。

3.2 DFTからDCTへ

図2(d)のように、切出した $x(m)$ を線対称にしたものを隣に置いた、長さ $2N$ の信号(周期 $2N$ の周期信号)を考える。

図2 DFTとDCT

$$y(m) = \begin{cases} x(m) & (0 \leq m \leq N-1) \\ x(2N-1-m) & (N \leq m \leq 2N-1) \end{cases} \quad (12)$$

$y(m)$ から長さ $2N$ の DFT によって得られる $Y(u)$ は、

$$Y(u) = \frac{1}{\sqrt{2N}} \sum_{m=0}^{2N-1} y(m) e^{-\frac{2\pi i}{2N} mu} \quad (13)$$

$$y(m) = \frac{1}{\sqrt{2N}} \sum_{u=0}^{2N-1} Y(u) e^{\frac{2\pi i}{2N} mu} \quad (14)$$

となる。ここで、 $C(l)$ ($l=0, 1, \dots, N-1$) を、

$$C(l) = \begin{cases} 1 & (l=0) \\ \sqrt{2} & (l \neq 0) \end{cases} \quad (15)$$

とおき、

$$Y'(u) = Y(u) \frac{C(u)}{\sqrt{2}} e^{-\frac{\pi i}{2N} u}$$

とおけば、

$$Y'(u) = \frac{C(u)}{2\sqrt{N}} \sum_{m=0}^{2N-1} y(m) e^{-\left[\frac{\pi i(2m+1)u}{2N}\right]} \quad (16)$$

$$y(m) = \frac{1}{\sqrt{N}} \sum_{u=0}^{2N-1} \frac{Y'(u)}{C(u)} e^{\left[\frac{\pi i(2m+1)u}{2N}\right]} \quad (17)$$

という変換と逆変換を考えることができる。ここで、式 (16) で加算される項の m に $2N-1-m$ を代入すれば

$$\begin{aligned} y(2N-1-m) e^{\left[-\frac{\pi i\{2(2N-1-m)+1\}u}{2N}\right]} \\ = y(m) e^{\left[\frac{\pi i(2m+1)u}{2N}\right]} \end{aligned} \quad (18)$$

となり、式 (16) の和を 0 から $N-1$ と N から $2N-1$ に分け、式 (18) の対称性を後者の和に用いれば、 0 から $N-1$ の和になるので、

$$Y'(u) = \frac{C(u)}{\sqrt{N}} \sum_{m=0}^{N-1} x(m) \cos \left[\frac{\pi(2m+1)u}{2N} \right] \quad (19)$$

が成立する。式 (19) より、 $Y'(u)$ はすべて実数である。

式 (17) で、 $y(m)$ も $Y'(u)$ も実数であるため、複素数の指数関数項の $\sin$ 関数の項は消えるはずであるから、

$$y(m) = \frac{1}{\sqrt{N}} \sum_{u=0}^{N-1} \frac{Y'(u)}{C(u)} \cos \left[\frac{\pi(2m+1)u}{2N} \right] \quad (20)$$

が成立する。式 (19) より、 $Y'(2N-u) = Y'(u)$ と $Y'(N)=0$ が成立するから、式 (20) の和を、 $m=0, 1$ から $N-1$, $N, N+1$ から $2N-1$ の四つの部分に分け、まとめれば、

$$x(m) = y(m) = \frac{1}{\sqrt{N}} \sum_{u=0}^{N-1} C(u) Y'(u) \cos \left[\frac{\pi(2m+1)u}{2N} \right] \quad (21)$$

が成立する。式 (19) と (21) が、DCT とその逆変換を表している。もとの信号を対称におく方法が複数通りあるため、DCT も複数通り存在するが、画像符号化では一般に上に示した DCT が使われる。

3.3 2次元 DCT

今までは、値が一列に並んでいる 1 次元信号の変換を論じてきたが、画像は値が縦横に並んでいる 2 次元信号である。画像符号化では、一般に画像の縦と横に対して可分な変換を用いる。すなわち、すべての行に対して 1 次元の変

図3 2次元 DCT

換を行った後、その結果のすべての列に対して 1 次元の変換を行って得られたものを、2 次元の変換の結果とするものである。DCT の場合に、この処理を一つの式で表す。2 次元信号 $x(m, n)$ ($m, n=0, 1, \dots, N-1$) から 2 次元 DCT (2D DCT-II) によって得られる結果を $X(u, v)$ ($u, v=0, 1, \dots, N-1$) とすれば、

$$\begin{aligned} X(u, v) = \frac{C(u)C(v)}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) \\ \cos \left[\frac{\pi(2m+1)u}{2N} \right] \cos \left[\frac{\pi(2n+1)v}{2N} \right] \end{aligned} \quad (22)$$

となり、2 次元逆 DCT (2D IDCT-II) は、

$$\begin{aligned} x(m, n) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v) X(u, v) \\ \cos \left[\frac{\pi(2m+1)u}{2N} \right] \cos \left[\frac{\pi(2n+1)v}{2N} \right] \end{aligned} \quad (23)$$

で表される。

式 (22) を見ると、 N^2 個の変換の結果 $X(u, v)$ は、入力信号と $\cos$ 関数の積で構成される N^2 種類の基底関数との内積を計算して得られる形になっている。図 3 (a) は、 $N=8$ の場合の、この 64 個の基底関数を図示したものである。

DCT の画像符号化における有効性を見るために、図 3 (b) の画像を、 8×8 画素のブロックに分割し、それぞれのブロックを DCT で変換し、低周波成分を表す 8 個の変換の結果 $X(u, v)$ ($(u, v) = (0, 0), (0, 1), (1, 0), (2, 0), (1, 1), (0, 2), (3, 0), (2, 1)$) はそのままにし、残りの 56 個の変換係数を 0 した後に逆変換して得られた画像が、図 3 (c) である。この図より、少数の変換係数で画像を十分な精度で表していることがわかる。

4. 離散ウェーブレット変換 (DWT)

DCT ではブロックごとに線形変換を行っており、データ量を小さくするために量子化ステップを大きくすると、復号画像のブロック境界に不連続な線が見える。その問題を解決できるブロックによらない変換が、JPEG2000 などで行われている DWT である。

DWT では、入力信号 $x(m)$ に対して、ローパスフィルタ ($h(n)$) とハイパスフィルタ ($g(n)$) を施し、低周波成分と高周波成分の二つの信号に分解する。そして、それぞれの成分から値を一つおきに取出すというダウンサンプリングを

図4 離散ウェーブレット変換の階層化

図5 離散ウェーブレット変換

行う。ダウンサンプリングすることによって、二つの成分のデータ数と原信号のデータ数が同じになる。

逆DWTによって、二つの成分から元の信号に戻すときには、まず、各成分の値の間に0を挿入することによって2倍の長さとするアップサンプリングを行う。そして、一般には分解時とは別のローパスフィルタ $\tilde{h}(n)$ とハイパスフィルタ $\tilde{g}(n)$ を施し、両者を加算すればよい。

そしてDWTでは、この1回の分解によって得られた高周波成分はそのまま利用するが、低周波成分はさらにその低周波成分と高周波成分に分解していく(図4)。画像符号化では、この分解を5から6回程度行うことが多い。

2次元信号の場合、DCTと同様に横と縦に対してそれぞれ変換する。すなわち、すべての行に対して分解した結果のすべての列に対して分解を行う。

図5(a)はDWTして得られた結果であり、画像の左上が縦横とも低周波成分、左下が横が低周波成分で縦が高周波成分、右上が横が高周波成分で縦が低周波成分、右下が縦横とも高周波成分を表している。左上は両方とも低周波成分であるため、さらに分解している。この図は3回の分解を行った結果である。DWTを使った画像符号化では、基本的には図5(a)の各成分の値を量子化して、ビット列に変換する。

DWTによって、データ量が削減できる理由を説明する。DCTと同様に周波数で分割するため、低周波成分が大きく、高周波成分は小さくなる。このことは、図5(a)において、高周波成分は暗いことからわかる。したがって、少数の成分に分散を集中させることができるため、データ量を小さくすることができる。図5(b)は、横に関しては1段、縦に関しては2段分解した時の低周波成分はそのままにし、他の成分の値は0にした後に逆変換して得られた画像である。DCTの場合の図3(c)に対応して、全体の1/8の値だけを残している。DWTの低周波成分でも、原画像を十分な精度で表していることがわかる。

さらに、DWTでは階層表現により、DWTの変換結果に残る空間的冗長性を利用して、データ量を削減することも可能である。DCTを用いる符号化では、 $(u, v) = (0, 0)$ の成分に関してだけ、隣接ブロックとの差分を取ることで、データ量を削減しているが、他の成分は完全に別々

に値を符号化してしまっている。一方、図5(a)を見ると、DWTの高周波成分の値が小さい(暗い)領域はかたまっていること、縮尺が異なるが、高周波成分は分解の段数異なる高周波成分と、その領域が対応していることがわかる。すなわち、分解の段数が多い高周波成分の量子化値が0ならば、分解の段数が少ない高周波成分の対応する位置の量子化値も0である可能性が高い。したがって、段数が多い高周波成分の量子化値からビット列にしていくときに、段数が少ない高周波成分の量子化値に対する符号を適応的に変えることによって、データ量を削減することができる。

DWTのためのフィルタ $h(n)$ 、 $g(n)$ 、 $\tilde{h}(n)$ 、 $\tilde{g}(n)$ は、多数提案されている。JPEG2000では、フィルタ関数の点数(タップ数)が、7または9であるDaubechiesの9-7フィルタが用いられる。また、画像符号化にDWTを適用するためには、信号が有限長であるための端点処理やデータ数が奇数である場合の処理も必要となるが、いずれの場合も変換前後でデータ数を変えずに、DWTと逆DWTが可能である。

5. む す び

画像符号化で用いられる線型変換に関して、それを用いる理由と変換の概要に関して解説した。さらに詳しく知りたい場合は、参考文献を参照すると良い。

(2012年10月29日受付)

〔文 献〕

- 1) H. Enomoto and K. Shibata: "Orthogonal Transform Coding System for Television Signals", IEEE Trans. on, Electromagnetic Compatibility, 13, 3, pp.11-17 (Aug. 1971)
- 2) K.R. Rao and P. Yip, (訳) 安田浩, 藤原洋: "画像符号化技術 [DCTとその国際標準]", オーム社 (July 1992)
- 3) I. Daubechies: "Ten Lectures on Wavelets", Society for Industrial and Applied Mathematics (June 1992)
- 4) S. Mallat: "A Wavelet Tour of Signal Processing, Third Edition: the Sparse Way", Academic Press (Dec. 2008)

やました 幸彦 1985年、東京工業大学大学院理工学研究科情報工学専攻修士課程修了。同年、日本原子力研究所入所。(株)アイザックを経て、現在、東京工業大学大学院理工学研究科国際開発工学専攻にて、パターン認識、画像処理、集合知利用に関する研究に従事。准教授、博士(工学)。正会員。