

## ■2 群 (画像・音・言語) - 5 編 (画像符号化)

# 5 章 周波数変換

(執筆著者: 岩橋政宏) [2013 年 4 月 受領]

### ■概要■

この章では、画像の信号波形をいくつかの成分に分け、それぞれの大きさを情報として符号化することで、より少ないデータ量で任意の画像を表現する方法を説明する。成分としては、緩やかな明暗の変化から線によるスケッチまで、いくつかの汎用的なものを用意しておき、符号化側と復号側で共有する。そして、これらの成分の大きさと位置を画像ごとに調べ、情報として符号化することで、画像の輝度値そのものを符号化する場合に比べ、より少ないデータ量で画像信号を表現する。

成分としては、いろいろな画像に対して汎用的に使えるものが望ましい。また、成分が相互に重複しないことが望ましい。更には、画像に含まれる細かい模様を正確に表現できることが望ましい。このような多くの要求を満たすように、できるだけ少ない種類の成分を用意することがこの章の課題である。

画像符号化に適する方法として、JPEG では離散コサイン変換 (DCT) が採用されたが、その後の JPEG 2000 では離散ウェーブレット変換 (DWT) に代わった。DCT は成分 (基底) として三角関数を用いる。成分間の重複性がなく (直交し)、任意の画像を成分の和として完全に表現 (再構成) できる。しかし、成分の空間的な広がり、ブロックサイズとして固定されている。これに対し、DWT は細かい模様のために局所的な成分 (高周波は短い基底) を、緩やかな明暗変化のためには大域的な成分 (低周波は長い基底) をそれぞれ備えている。このことが画像信号の性質と整合する。

基底としては、空間的に局在化し、相互に直交し、画像を再構成できる、必要最小限のものを揃える。更に、再生画像のひずみを低減するには対称性 (直線位相性) も必要となる。制約条件が多くなると自由度が減少し、信号への適合度が低下する。このため、正規直交性を双直交性に緩める方法 (リフティング構成) や、基底をフレームに一般化して、1 次独立性を緩和した冗長な (過完備な) 変換などが提案されている。また、画像特有の課題として、細かい模様の向きに対応できる (方向適応型) の変換も数多く研究されている。

### 【本章の構成】

本章では、周波数変換の原理 (5-1 節)、アダマール変換 (5-2 節)、KL 変換 (5-3 節)、離散コサイン変換 (5-4 節)、変換処理の高度化 (5-5 節)、サブバンド符号化とフィルタバンク (5-6 節)、ウェーブレット変換とブロック変換 (5-7 節)、ウェーブレット変換のリフティング構成 (5-8 節)、x-let (5-9 節) について述べる。

■2群 - 5編 - 5章

5-1 周波数変換の原理

(執筆著：岩橋政宏) [2013年4月 受領]

周波数変換を用いた信号処理では、周波数の異なる複数の三角関数を成分として、振幅と位相を調整して重ね合わせることで、与えられた任意の画像を表現する。例えば、画像の1ライン（縦あるいは横）について、位置  $n = 0, 1, 2, \dots, N-1$  における画素の明るさを  $x(n)$  として、この信号波形を、

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{jkn\omega_0}, \quad \omega_0 = \frac{2\pi}{N} \tag{5.1}$$

と表現する。ここで、 $X(k)$ はフーリエ係数と呼ばれ、

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-jkn\omega_0}, \quad \omega_0 = \frac{2\pi}{N} \tag{5.2}$$

と計算される。入力波形に対称性がなければ、結果は一般に複素数となる。この値を極座標表現することで、第  $k$  番目の成分である、

$$e^{jkn\omega_0} = \cos kn\omega_0 + j \sin kn\omega_0 \tag{5.3}$$

の振幅と位相が得られる。第1から第  $N-1$  までの  $N$  個の全成分を、式(5.1)により合成することで、もとの波形  $x(n)$  が完全に再構成される。この変換は、離散フーリエ変換 (DFT) と呼ばれ (整数  $k$  に関する和という意味では「級数展開」)、高速な計算法 (FFT) が存在することから、画像処理において広く利用されている。

一方、フーリエ係数の小さな成分を省き、大きな成分のみを合成しても、もとの波形は概ね再生される。特に画像の場合、画素間の相関が強く、輝度が緩やかに大きく変化する低い周波数成分 ( $k$  の値が小さい成分) が強い。このことから、高い周波数成分のフーリエ係数を捨てることで、再生される画像の画質を若干犠牲として、画像の表現に必要なデータ量を圧縮できる。ただし、整数である画素値を周波数変換すると、フーリエ係数は一般に複素数、すなわち実部と虚部 (ないしは振幅と位相) の2種類となるため、このままではデータ圧縮の用途には適さない。これに対し、次節以降で説明するように、値の総数が増加しない (冗長ではない) 変換が数多く提案されている。

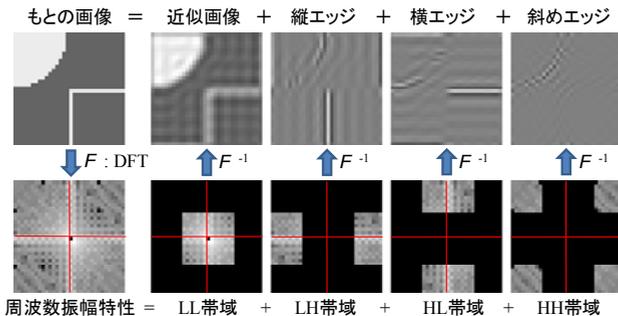


図 5.1 画像波形の周波数変換. いくつかの周波数帯域に分解して処理する

図 5・1 には、周波数変換の例を示す。図中の「周波数振幅特性」は、「もとの画像」の縦方向に、更には横方向に、式(5・2)の DFT を施して得たフーリエ係数の振幅値（複素数の絶対値）である。中心が全画素の平均値（最低周波数）を、中心から遠ざかるほど高い周波数の成分を意味する。このフーリエ係数を四つの帯域 {LL, LH, HL, HH} に分割し、それぞれを逆変換することで、同図に示すような {近似画像, 縦エッジ, 横エッジ, 斜めエッジ} が得られる。例えば、近似画像さえ閲覧できればよく、できる限りデータ量を低減したい場合は、LL 帯域のフーリエ係数のみを蓄積・転送する。また、高画質が要求される場合は、エッジを再生するのに必要な高周波帯域も併せて利用する、という考え方である。

■2 群 - 5 編 - 5 章

5-2 アダマール変換

(執筆著：岩橋政宏) [2013 年 4 月 受領]

図 5・2 は、ウォルシュ・アダマール変換 (Walsh Hadamard Transform : WHT) の処理手順を示す。画像信号を  $2 \times 2$  画素のブロックに分け、各ブロックに対して行列  $\mathbf{W}_2$  を、

$$\begin{pmatrix} x'_{00} \\ x'_{10} \end{pmatrix} = \mathbf{W}_2 \begin{pmatrix} x_{00} \\ x_{10} \end{pmatrix}, \quad \begin{pmatrix} x'_{01} \\ x'_{11} \end{pmatrix} = \mathbf{W}_2 \begin{pmatrix} x_{01} \\ x_{11} \end{pmatrix}$$

$$\begin{pmatrix} y_{LL} \\ y_{HL} \end{pmatrix} = \mathbf{W}_2 \begin{pmatrix} x'_{00} \\ x'_{01} \end{pmatrix}, \quad \begin{pmatrix} y_{LH} \\ y_{HH} \end{pmatrix} = \mathbf{W}_2 \begin{pmatrix} x'_{10} \\ x'_{11} \end{pmatrix} \tag{5・4}$$

ただし、

$$\mathbf{W}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{5・5}$$

として縦と横に適用する。以上をまとめて、 $\mathbf{O}_2$  を  $2 \times 2$  のゼロ行列として、

$$\begin{pmatrix} y_{LL} \\ y_{HL} \\ y_{LH} \\ y_{HH} \end{pmatrix} = \begin{pmatrix} \mathbf{W}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{W}_2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{W}_2 & \mathbf{O}_2 \\ \mathbf{O}_2 & \mathbf{W}_2 \end{pmatrix} \begin{pmatrix} x_{00} \\ x_{10} \\ x_{01} \\ x_{11} \end{pmatrix} \tag{5・6}$$

と表すこともできる。式(5・5)の右辺全体をルート 2 で割っているのは、各行の成分の 2 乗和 (ノルム) を 1 に正規化するためである。

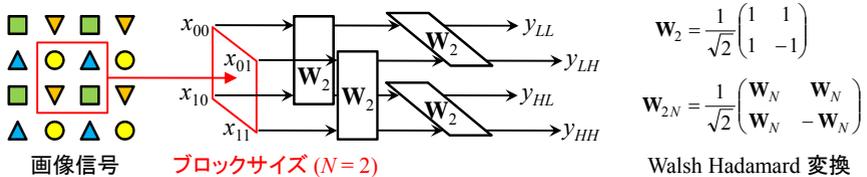


図 5・2 2次元のアダマール変換 (WHT) の処理手順。N=2 の例

この変換を整数の画素値に適用しても、結果は複素数にはならず、値の総数は増加しない (冗長ではない)。式(5・6)の左辺は「変換係数」と呼ばれ、隣接する画素間での「和」と「差」として簡単に計算できる。添え字の「L」と「H」は、低い周波数と高い周波数にそれぞれ対応することを意味する。高い周波数を削減することでデータ量が削減されるが、逆変換により再生される画像は、細かい模様がひずんでしまう。こうしたひずみを低減することが、変換を設計するうえでの課題となる。

図 5・3 には、アダマール変換の例を示す。図中の「もとの画像」に  $\mathbf{W}_2$  を適用して「変換係数」を得た。ただし、 $y_{LL}$  は左上、 $y_{LH}$  は右上、 $y_{HL}$  は左下、 $y_{HH}$  は右下にそれぞれ集めて表示してある。図 5・2 と同様、四つの帯域をそれぞれ逆変換することで、近似画像と 3 種類のエッジが得られる。DFT との違いとして、エッジ部分だけが綺麗に抽出できている。これは、

$W_2$ による処理が $2 \times 2$ 画素のブロック内に「コンパクト」に収まっていることによる。以上より、ブロックが小さい方が、小さく細かい模様（局所的な高周波）の位置を精度よく検出できることが分かる。一方、輝度の緩やかな変化を抽出するには、大きなブロックが適している。画像符号化では、残したい模様と、削っても構わない模様を明確に分離できるよう、用途に応じて適切に変換を選択あるいは設計する必要がある。

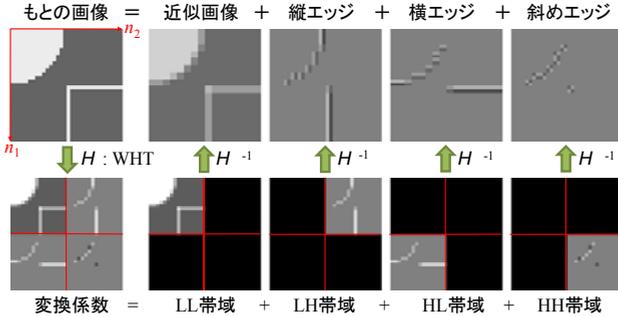


図 5・3 画像波形のアダマール変換。いくつかの基底に分解して符号化する

次に、変換を規定する「基底」について説明する。式(5・6)を展開して並び替えると、

$$\begin{pmatrix} y_{LL} & y_{LH} \\ y_{HL} & y_{HH} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} (x_{00} + x_{01} + x_{10} + x_{11}) & (x_{00} - x_{01} + x_{10} - x_{11}) \\ (x_{00} + x_{01} - x_{10} - x_{11}) & (x_{00} - x_{01} - x_{10} + x_{11}) \end{pmatrix} \quad (5 \cdot 7)$$

となる。右辺の要素は、画素値 $\{x_{00}, x_{01}, x_{10}, x_{11}\}$ に+1 ないしは-1 の重みを付けた和となっている。この重みを、 $(x_{00}, x_{01})$ は1行目、 $(x_{10}, x_{11})$ は2行目として行列で表現すると、

$$\begin{pmatrix} \phi_{LL} & \phi_{LH} \\ \phi_{HL} & \phi_{HH} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \begin{pmatrix} +1 & +1 \\ +1 & +1 \end{pmatrix} & \begin{pmatrix} +1 & -1 \\ +1 & -1 \end{pmatrix} \\ \begin{pmatrix} +1 & +1 \\ -1 & -1 \end{pmatrix} & \begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix} \end{pmatrix} \quad (5 \cdot 8)$$

となる。これは、式(5・1)の $\exp(jkn \omega_0)$ に相当し、変換においては「基底」(広い意味では「フレーム」)と呼ばれる。一般に、一次変換は長さ $N$ の信号波形を $M$ 種類の基底の線形結合として表現する。画像のデータ圧縮では、 $N=M$ の場合が広く用いられるが、信号の解析では $N < M$ となる冗長な場合も用いられる。最近では、冗長性をもたせて基底の自由度を上げ、全体としてはできるだけ少ない数の基底で、残したい模様を近似する方法(スパースコーディング)も研究され、低ビットレート符号化での効果が報告されている。

図 5・4 の上段には、式(5・8)の基底を+1 は白、-1 は黒として図示した。DFT の基底が画像全体に広がっているのとは異なり、すべて $2 \times 2$ 画素のブロック内に収まっている。つまり、基底が空間的に局在化されている。変換係数は基底が画像中のどの位置に存在するかをブロック単位で示す。このため、ブロックサイズが小さいほど、空間分解能が高い。一方、図 5・4 の下段には、これらの基底に DFT を施した (WHT の基底を DFT の基底で表現した) 結果を示す。図 5・1 の下段と同様、四つの帯域{LL, LH, HL, HH}に分割されている。

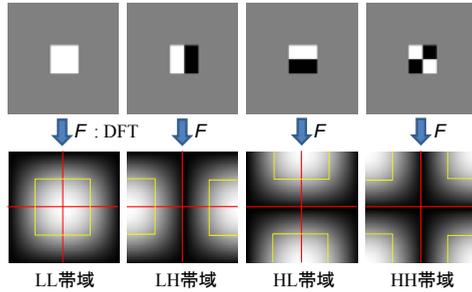


図 5・4 アダマール変換の基底とその周波数振幅特性

一般に、基底が空間的に局在化すると、その周波数成分の帯域幅は広がる。すなわち、空間分解能が高くなると、周波数分解能は低くなる（両者の積は一定）。例えば、

$$\mathbf{W}_4 = \begin{pmatrix} \mathbf{W}_2 & \mathbf{W}_2 \\ \mathbf{W}_2 & -\mathbf{W}_2 \end{pmatrix} \quad (5 \cdot 9)$$

として基底の長さを倍にすると帯域幅は狭くなる。これは、より詳細に周波数の種類（模様の違い）を弁別できることを意味する。反面、その成分が画像中のどの位置に存在するかという、空間的な位置に関する精度は低下する。画像符号化には、細かい模様には局所的な基底を、全体的なグラデーションの変化には広域的な基底を、それぞれ使い分けることが有効である。これを効果的に実現する手段が、後述するウェーブレット変換である。 $\mathbf{W}_2$ はハール基底としてウェーブレット変換でも利用される。

■2群 - 5編 - 5章

5-3 KL 変換

(執筆者：岩橋政宏) [2013年4月 受領]

フーリエ変換やアダマール変換では、基底の形は三角関数や矩形波としてあらかじめ定まっている。これに対し、カルーネン・レーベ変換 (Karhunen-Loève Transform : KLT) では、基底の形が個々の入力に応じて最適に決定される。結果、より少ない基底により画像を忠実に表現できる。このため、ロッシー符号化において画像データをより小さく圧縮できる。また、リフティング構成を応用することで、KLT によるロスレス符号化も可能となっており、衛星画像のようなマルチスペクトル画像の圧縮にも応用されている。以下、ロスレス符号化を含む一般的な場合について、KLT によるデータ圧縮の原理を説明する。

図 5・5 には、ブロックサイズが  $2 \times 1$  画素の場合を例に、KLT の設計の指針と手順をまとめる。この場合、 $2 \times 2$  行列  $\mathbf{K}$  により、入力信号  $\mathbf{X} = (x_0 \ x_1)^T$  から 2 種類の変換係数  $\mathbf{Y} = (y_L \ y_H)^T$  がブロックごとに計算される。一般に、信号の符号量はその分散  $\sigma^2$  の対数に比例するため、変換後の総符号量  $H_y$  は、2 種類の変換係数の分散の、対数の相加平均、すなわち、分散の相乗平均 (GM) の対数として表現される。これが KLT による圧縮後のデータ量である。一方、変換前の分散は、2 種類の変換係数それぞれの分散の相加平均であるため、圧縮前の符号量は、分散の相加平均 (AM) の対数となる。つまり、相乗平均と相加平均の比が大きくなるように、行列  $\mathbf{K}$  を設計すればよい。ただし、変換の前後でエネルギー (分散の総和) が保存される (Parseval の定理が成り立つ) と仮定している。

<p>1. ブロック変換</p> $\begin{pmatrix} y_L \\ y_H \end{pmatrix} = \mathbf{K} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \quad \mathbf{K} = \begin{pmatrix} t_{00} & t_{01} \\ t_{10} & t_{11} \end{pmatrix}$ <p>任意の入力に対し基底 <math>t_{ij}</math> を最適化</p>	<p>5. 符号化利得 (歪みの比、同じ符号量)</p> $G = 10 \log_{10} \frac{\sigma_{ex}^2}{\sigma_{ey}^2} = 6.02(H_x - H_y)$ $= 3.01 \cdot \log_2 \frac{AM}{GM} \quad [\text{dB}] \quad \text{を最大化}$
<p>2. データ圧縮に適した変換は?</p> $H_x = \frac{1}{2} \log_2 \gamma \sigma_x^2 \quad [\text{bit}] \rightarrow H_y = \frac{H_{y_L} + H_{y_H}}{2} \quad \text{を最小化}$	<p>6. カルーネン・レーベ変換 (KLT)</p> <p><math>GM^2 \geq \det \mathbf{R}_{yy}</math> なので、 GM を最小化 <math>\Rightarrow \mathbf{R}_{yy} = \mathbf{K}^T \mathbf{R}_{xx} \mathbf{K}</math> を対角化</p>
<p>3. ロスレス符号化 (符号量)</p> <p>変換前: <math>H_x = \frac{1}{2} \log_2 \gamma (AM), \quad AM = \frac{\sigma_{y_L}^2 + \sigma_{y_H}^2}{2}</math></p> <p>変換後: <math>H_y = \frac{1}{2} \log_2 \gamma (GM), \quad GM = \sqrt{\sigma_{y_L}^2 \sigma_{y_H}^2}</math></p>	<p>7. 入力信号の共分散行列 <math>\mathbf{R}_{xx}</math> に対する固有ベクトルとして基底行列 <math>\mathbf{K}</math> を得る</p> <p>但し、<math>\mathbf{R}_{yy} := E[\mathbf{Y}\mathbf{Y}^T], \quad \mathbf{R}_{xx} := E[\mathbf{X}\mathbf{X}^T]</math></p> $\sigma_{y_L}^2 := E[x^2], \quad \sigma_{ex}^2 := E[(x'-x)^2]$ $\sigma_{ey}^2 = \frac{\sigma_{y_L}^2 + \sigma_{y_H}^2}{2}, \quad \sigma_x^2 = \frac{\sigma_{y_L}^2 + \sigma_{y_H}^2}{2}$
<p>4. ロッシー符号化 (符号量と符号化歪み)</p> <p>変換前: <math>H'_x = H_x - \log_2 q_x, \quad \sigma_{ex}^2 = \frac{1}{12} q_x^2</math></p> <p>変換後: <math>H'_y = H_y - \log_2 q_y, \quad \sigma_{ey}^2 = \frac{1}{12} q_y^2</math></p>	

図 5・5 カルーネン・レーベ変換 (KLT) の概要 (設計の指針と手順)

このように、KLT では基底のセットである  $\mathbf{K}$  が、入力信号の共分散行列  $\mathbf{R}_{xx}$  に対する固有ベクトルとして決まる。入力信号が白色性のスペクトルをもつ場合は、 $\mathbf{K}$  をどのように選んでも、2 種類の変換係数の分散は同じである。すなわち、 $GM = AM$  なので、KLT によるデー

タ圧縮は期待できない。一方、画像のように画素間の相関が強い、有色性の入力信号に対しては、変換係数の分散が互いに大きく異なる。このため、 $GM < AM$  となるように  $\mathbf{K}$  を決定することで、入力に応じた効果的なデータ圧縮が、ロスレス符号化においても可能となる。また、ロッシェ符号化では、KLTの変換係数を量子化した値（ステップサイズで割ってから整数に丸めた値）をエントロピー符号化する。量子化により値の小さい変換係数はゼロになるが、これによる再生画像の画質への影響は、 $L_2$  ノルムの意味で最小に抑えられている。

■2群 - 5編 - 5章

5-4 離散コサイン変換

(執筆者：岩橋政宏) [2013年4月 受領]

KLTは、符号化利得の意味で最適な正規直交変換となっている。しかし、入力ごとに固有ベクトルを計算し、結果をオーバーヘッドとして圧縮データに付加する必要がある。これに対し、離散コサイン変換 (Discrete Cosine Transform : DCT) は、入力ごとの最適化が不要であり、様々な画像に対して平均的な圧縮性能をもつ。特に  $N=8$  の DCT は、JPEG や MPEG などの国際標準に採用されており、画像圧縮においては非常に効果的な変換である。

多くの画像波形は、「近隣の画素は、互いに値が似ており、その度合いは距離が離れるほど、指数関数的に減少する」という性質をもつ。そのような波形は、AR(1)モデルとして、白色雑音  $w(n)$  と相関係数  $\rho$  により (ただし、 $\rho$  は 1 よりやや小さい実数)、

$$x(n) = w(n) + \rho x(n-1) \tag{5 \cdot 10}$$

と記述される。そのスペクトルは周波数に対する単調減少関数であり、エネルギーが低域に偏っている。DCT は AR(1)モデルに対する KLT に相当する。DCT にはいくつかのタイプがあるが、順変換としては DCT-II が広く使われており、

$$X(k) = \sum_{n=0}^{N-1} x(n) \left( C_N^{II} \right)_{kn} \tag{5 \cdot 11}$$

により変換係数が計算される。これに対する逆変換は DCT-III と呼ばれる。

図 5・6 には、 $N=4$  の DCT-II を、 $N=2$  の回転行列に因数分解する例を示す。このように、大きな行列を小さな行列に因数分解することで、使用する一次記憶メモリの容量を小さくできる。また、DFT に対する FFT のように、計算に要する時間を大幅に短縮できる。DCT には数々の高速計算アルゴリズムが存在するため、リアルタイム処理に適した様々な LSI が開発されている。

**離散コサイン変換 (DCT)**

$$\left( C_{N+1}^I \right)_{kn} = \sqrt{\frac{2}{N}} \varepsilon_k \varepsilon_n \cos\left(\frac{kn}{N} \pi\right), \varepsilon_p = \begin{cases} 1/\sqrt{2} & p=0, N \\ 1 & \text{上記以外} \end{cases}$$

$$\left( C_N^{II} \right)_{kn} = \sqrt{\frac{2}{N}} \varepsilon_k \cos\left(\frac{k(2n+1)}{2N} \pi\right)$$

**DCTとKLTの関係**

$(R_{xx})_{nm} = \sigma_x^2 \rho^{|n-m|}$  である Markov Model に対し  
 $\rho \rightarrow 1, N \rightarrow \infty$  の場合の KLT が、DCT-I であり  
 $\rho \rightarrow 1$  かつ  $N$  が  $\rho$  と独立の場合が、DCT-II である

**サイズ  $N$  の DCT は、 $N/2$  に分解できる**

$$C_N^{II} = B_N \begin{pmatrix} \hat{C}_{N/2}^{II} & 0 \\ 0 & \hat{C}_{N/2}^{IV} \cdot J_{N/2} \end{pmatrix} \begin{pmatrix} I_{N/2} & J_{N/2} \\ J_{N/2} & -I_{N/2} \end{pmatrix}$$

**$N=4$  の DCT-II を、 $N=2$  に分解した例**

$R_1 = R_2 = R_3 = H_{\pi/4}, \quad R_4 = G_{-3\pi/8}$

$H_{\theta} = \begin{pmatrix} \cos \theta & \sin \theta \\ \sin \theta & -\cos \theta \end{pmatrix}, \quad G_{\theta} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$

Householder Reflection                  Givens-Jacobi Rotation

図 5・6 離散コサイン変換 (DCT) の概要 (KLT との関係, 高速アルゴリズム)

■2群 - 5編 - 5章

5-5 変換処理の高度化

(執筆著：岩橋政宏) [2013年4月 受領]

変換処理をより少ないメモリで、より高速に実現するには、乗算器の総数を減らすことが効果的である。例えば、 $N=2$ の回転行列には、 $\cos \theta$  と  $\sin \theta$  による実数乗算が四つ含まれる。特に、回転角度が  $\pi/4$  [rad] の場合はアダマール変換と等しく、 $\sqrt{2}$  を除けば係数は  $\pm 1$  のみとなり、実数乗算に比べて演算負荷が大幅に軽くなる。また、 $\sqrt{2}$  による除算を2回分まとめて実施すれば、1/2倍すなわち1ビット右へのシフト演算で実現できる。更に、乗算器の係数値や信号値を短い語長の2進数で表現することも効果がある。ただし、短語長化による誤差が生じ、一般には再生画像の画質が劣化する。

図5・7(右側)には、短語長化にもかかわらず、再生画像に全く誤差が生じない(ロスレスとなる)処理方法を示す。説明を簡単にするため、 $p=0$  の場合を考える。順変換では  $y_0 = x_0 - ux_1$  なる演算を行い、逆変換では  $y_0 + ux_1$  なる演算を行うことで  $x_0$  が再生される。ここで、丸め処理  $R[\cdot]$  を導入すると、順変換では  $y_0 = x_0 - R[ux_1]$  となり丸め誤差が生じる。しかし、逆変換では  $y_0 + R[ux_1]$  なる演算を行うので、 $-R[ux_1]$  と  $R[ux_1]$  のそれぞれで生じた丸め誤差がキャンセルし、 $x_0$  がロスレスに再生される。この原理を応用することで、ロスレス再生を保障したうえで、任意の回転変換を短語長化できる。更には、整数入力に対して整数を出力する整数変換を、順方向と逆方向のそれぞれについて構成できる。このため、KLT や DCT によるロスレス符号化が可能となる。

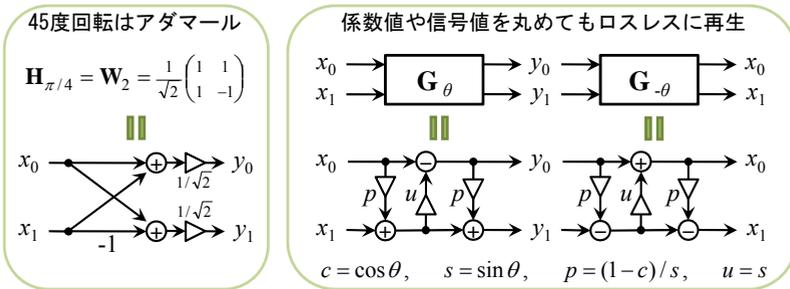


図5・7 変換処理の高度化 (乗算器数の低減, 整数変換の実現)

■2群 - 5編 - 5章

5-6 サブバンド符号化とフィルタバンク

(執筆著者：岩橋政宏) [2013年4月 受領]

サブバンド符号化は、画像信号を幾つかの周波数帯域（サブバンド）に分割してから、それぞれを量子化してエントロピー符号化する。図5・8（左側）には、2ステージのオクターブ分割の処理例を示す。4種類の2次元デジタルフィルタにより、画像信号 $X$ を四つの帯域{LL, LH, HL, HH}に分割する。ただし、このままでは符号化すべき画素の総数が4倍になる（冗長になる）ので、サブサンプリング（ $\downarrow D$ ）により画素数を4:1に間引く。画像信号の場合、低周波帯域（LL）のエネルギーが特に大きいことから、この帯域分割処理をLLに対して繰り返し適用する。結果、図5・8（右側）に示すように帯域がオクターブ分割され、低域については周波数の違い（緩やかな輝度変化の相違）を高い精度で分析できる。

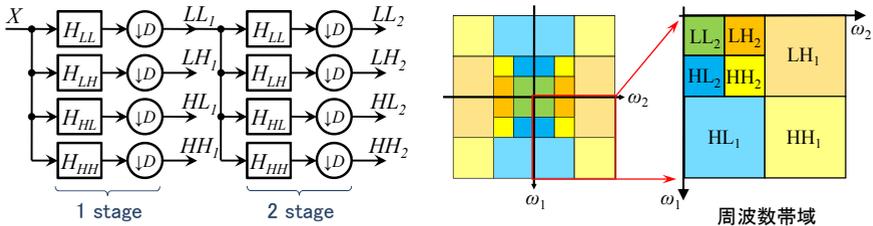


図5・8 フィルタバンクの連続接続と、オクターブ分割された周波数帯域

サブバンド符号化では、信号値の総数（信号列の構成要素である値の総数）を、変換により増加させない条件下で、信号列を複数の帯域信号に分割し、再度、もとの信号列を再構成できるように、フィルタの集合（フィルタバンク）を設計することが課題となる。例えば、帯域分割におけるサブサンプリングにより、エイリアシングの発生は避けられない。そこで、帯域信号を合成したときに、帯域間でエイリアシングがキャンセルするようにフィルタを設計する。これにより完全再構成が実現される。また、特に画像の場合、スケッチ線のような細かい模様を精度よく抽出できるように、フィルタのインパルス応答を決定する必要がある。例えば、1次元フィルタを縦と横に適用した場合、斜め方向に対する自由度はない。これに対し、斜め方向の角度の分解能を高めた「方向性 (Directional)」フィルタバンクが提案され、任意の曲線模様に対する表現力が高められている。

■2 群 - 5 編 - 5 章

5-7 ウェーブレット変換とブロック変換

(執筆者：岩橋政宏) [2013 年 4 月 受領]

離散ウェーブレット変換 (Discrete Wavelet Transform : DWT) は、信号を複数の「基底」の線形結合として表現する点では、DFT, WHT, KLT, DCT といったブロック変換と同じである。ただし、ブロック変換ではすべての基底が同じ長さをもっているのに対し、DWT では周波数ごとに異なる。具体的には、高い周波数を表す基底は短く、低い周波数を表す基底は長い。前者は空間的に局在化したエッジの表現に、後者はエネルギーが集中する低域を細かく帯域分割するのに適している。例えば、局所的な細かい模様 (パルスの波形) の周波数成分は広帯域に及ぶため、DFT や DCT では一般に多数の基底が必要となる。

ウェーブレット変換もブロック変換も共にフィルタバンクとして表現できるため、統一理論の下、用途に応じた様々な制約条件下で基底が設計される。JPEG のロシーン符号化で利用される DCT では、変換係数を粗く量子化するとブロックひずみが生じる。これを低減するため、基底をブロックよりも広くして、ブロック間で重畳させる変換 (Lapped Orthogonal Transform : LOT) が提案された。最近では、非分離型構成による方向分解能の向上、リフティング構成によるロスレス符号化への拡張など、多方面にわたり高度化されている。

図 5・9 には、アダマール変換 (WHT) と DWT の関係を示す。WHT は、入力信号をサブサンブラ (↓2) で間引いた後、基底行列  $W_2$  を掛けることに等しい。これは、低域通過フィルタ  $H_L(z)$  と高域通過フィルタ  $H_H(z)$  により帯域分割した後、2:1 に間引く処理として表現できる。これらのフィルタのインパルス応答が、WHT の基底に他ならない。どちらも 2 タップであり、基底の長さは同じである。これは、ハール基底によるウェーブレット変換とも呼ばれる。図 5・10 は、図 5・2 の 2 次元の WHT を、2 次元の DWT として等価表現している。ここでは、すべての帯域で基底の広がりには  $2 \times 2$  画素であるが、図 5・8 のオクターブ分割に適用すると、高域の  $\{LH_1, HL_1, HH_1\}$  は  $2 \times 2$ 、低域の  $\{LL_2, LH_2, HL_2, HH_2\}$  は  $4 \times 4$  となる。すなわち、高い周波数は局所的な基底、低い周波数は広域的な基底となっている。

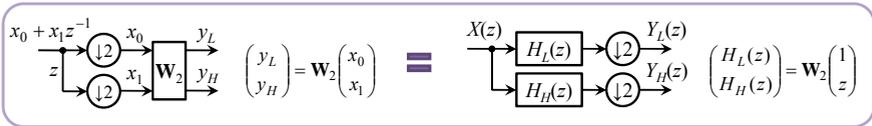


図 5・9 1次元アダマール変換のウェーブレット変換としての表現

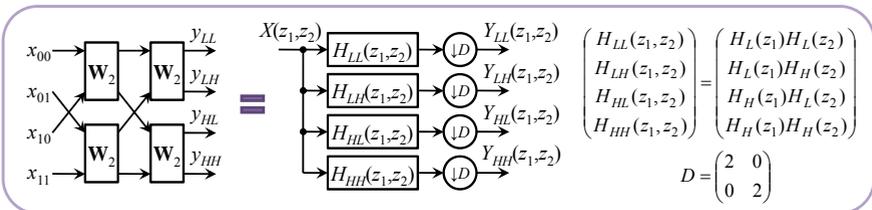


図 5・10 2次元アダマール変換のウェーブレット変換としての表現

■2群 - 5編 - 5章

5-8 ウェーブレット変換のリフティング構成

(執筆著: 岩橋政宏) [2013年4月 受領]

画像符号化の国際標準である JPEG や MPEG では、DCT が採用された。その後の国際標準である JPEG 2000 では、リフティング構成された双直交な離散ウェーブレット変換 (DWT) が採用された。JPEG 2000 には、ロッキー符号化のための 9/7 DWT と、ロスレス符号化のための 5/3 DWT がある。後者の場合、低域通過フィルタは 5 タップ、高域通過フィルタは 3 タップであり、低域は長い基底、高域は短い基底となっている。

図 5・11 (右側) には、5/3 DWT の処理手順を示す。整数として与えられる入力信号列  $\{x(0), x(1), x(2), \dots, x(N-1)\}$  は、サブサンブラ ( $\downarrow 2$ ) と遅延器  $z$  により、偶数番号  $\{x(0), x(2), \dots\}$  と奇数番号  $\{x(1), x(3), \dots\}$  に分けられる。次に、整数への丸め処理を  $R[\cdot]$ 、フィルタ係数を  $(a, b) = (-1/2, 1/4)$ 、サブサンプル後の位置を  $m = 1, 2, \dots, N/2 - 1$  として、

$$\begin{pmatrix} y_H(m) \\ y_L(m) \end{pmatrix} = \begin{pmatrix} x(2m+1) \\ x(2m) \end{pmatrix} + \begin{pmatrix} R[a(x(2m) + x(2m+2))] \\ R[b(y_H(m) + y_H(m-1))] \end{pmatrix} \quad (5 \cdot 12)$$

により、整数の帯域信号が生成される。これを EBCOT (Embedded Block Coding with Optimized Truncation) によりエントロピー符号化することで、圧縮データが生成される。このとき、図中のフィルタ  $P(z)$  と  $U(z)$  は、

$$\begin{pmatrix} P(z) \\ U(z) \end{pmatrix} = \begin{pmatrix} a(1+z^{-1}) \\ b(1+z^{-1}) \end{pmatrix} \quad (5 \cdot 13)$$

と表される。もとの信号を再生するには、EBCOT で復号した後、5/3 DWT の逆変換を適用する。この際、図 5・7 (右側) に示した原理により、もとの信号がロスレスに再生される。

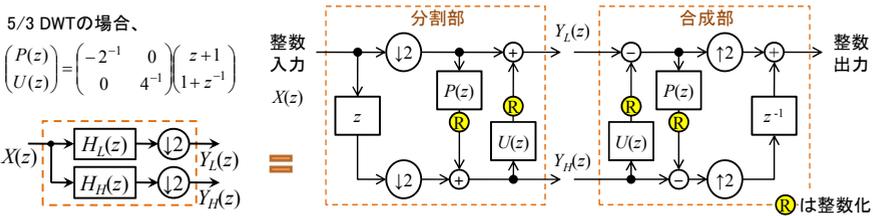


図 5・11 整数型のリフティング・ウェーブレット変換による 1 次元 2 帯域分割

図 5・11 (左側) に示す低域通過フィルタ  $H_L(z)$  と高域通過フィルタ  $H_H(z)$  はそれぞれ、

$$\begin{pmatrix} H_L(z) \\ H_H(z) \end{pmatrix} = \begin{pmatrix} 1 & U(z^2) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ P(z^2) & 1 \end{pmatrix} \begin{pmatrix} 1 \\ z \end{pmatrix} \quad (5 \cdot 14)$$

として式(5・13)と関係づけられる。したがって、式(5・13)を式(5・14)に代入することで、

$$\begin{pmatrix} H_L(z) \\ H_H(z) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & z \end{pmatrix} \begin{pmatrix} 1+2ab & b & ab \\ 1 & a & 0 \end{pmatrix} \begin{pmatrix} 1 \\ z+z^{-1} \\ z^2+z^{-2} \end{pmatrix} \quad (5 \cdot 15)$$

が得られ、 $H_L(z)$ は5タップ、 $H_H(z)$ は3タップであることが確認できる。

次に、2次元である画像処理に特有の問題について説明する。画像の縦（または横）方向に1次元処理を適用した後、横（または縦）方向に適用することで、画像信号は四つの帯域{LL, LH, HL, HH}に分割される。この際に用いられるフィルタは、式(5・15)から、

$$\begin{cases} G_H(z_1) = z_1^{-1}H_H(z_1) = 1+a(z_1+z_1^{-1}) \\ G_H(z_2) = z_2^{-1}H_H(z_2) = 1+b(z_2+z_2^{-1}) \end{cases} \quad (5 \cdot 16)$$

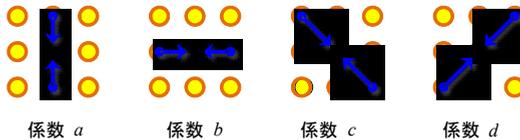
を定義すると、例えば、HH帯域については、

$$\begin{aligned} G_{HH}(z_1, z_2) &= G_H(z_1)G_H(z_2) \\ &= 1+a(z_1+z_1^{-1})+b(z_2+z_2^{-1})+ab(z_1z_2+z_1^{-1}z_2^{-1}+z_1^{-1}z_2+z_1z_2^{-1}) \end{aligned} \quad (5 \cdot 17)$$

を経た後、縦と横、共に2:1にサブサンプルした値と等しくなる。このような処理は、式(5・17)の1行目が二つの伝達関数、 $G_H(z_1)$ と $G_H(z_2)$ に分離できることから、「分離型」の2次元処理と呼ばれる。この場合、同式の2行目からわかるように、係数値を縦方向は $a$ 、横方向は $b$ と、それぞれ個別に指定できるが、斜め方向は $ab$ として自動的に決まり、選択の自由度がない。これに対し実際の画像には、縦エッジ、横エッジ、斜めエッジはそれぞれ独立に存在する。このため、

$$G_{HH}(z_1, z_2) = 1+a(z_1+z_1^{-1})+b(z_2+z_2^{-1})+c(z_1z_2+z_1^{-1}z_2^{-1})+d(z_1z_2^{-1}+z_1^{-1}z_2) \quad (5 \cdot 18)$$

として、**図 5・12**に示すように、縦90度は $a$ 、横0度は $b$ 、斜め-45度は $c$ 、+45度は $d$ として、各方向を独立に指定できる方が望ましい。このようなフィルタは「非分離型」と呼ばれる。ただし、1次元のラインメモリで構成される分離型とは異なり、2次元のメモリアクセスが必要となるため演算負荷は高くなる。



**図 5・12** HH帯域におけるフィルタ処理の方向。分離型には $c=d=ab$ なる制約がある

■2群 - 5編 - 5章

5-9 x-let

(執筆著：岩橋政宏) [2013年4月 受領]

離散ウェーブレット変換 (DWT) は、信号をいくつかの成分に分解する際、広域的に緩やかに変化する成分 (空間的に広域な、低い周波数の基底) と、局所的で細かい模様成分 (空間的に局在化した、高い周波数の基底) をもつため、デジタル画像の表現に適している。更なる課題として、模様の「方向」を更に細かく弁別することがあげられる。これに対し、変換係数の総数 (基底の種類) が入力画素の総数 (信号の次元) よりも大きく冗長な、「過完備 (Over Complete)」な変換を用いる方法が提案されている。この場合、基底に相当するものはフレームである。基底のように 1 次独立性や正規直交性を満たすとは限らないが、スパースコーディングなど、設計の自由度を信号解析に活用する研究が注目されている。

図 5・13 (左側) には、分離型 DWT (9/7 DWT の 3 Stage 目) の基底と、これに対応する周波数帯域を示す。LH 帯域に対応する基底は縦長なので 90 度、HL 帯域は横長で 0 度の方向をもつエッジをそれぞれ検出する。また、HH 帯域の基底は+45 度と-45 度の両方を検出する。したがって、分離型 DWT が検出できる方向は 3 種類に限られる。これに対し、デュアルツリー複素 DWT の場合、図 5・13 (右側) に示すように{-75, -45, -15, 15, 45, 75}度の 6 種類を区別できる。これは、実部と虚部が 90 度の位相差をもつヒルベルト変換対を生成し、片側帯域信号 (図中の①と②) を得て両者を掛け合わせ (図中の③)、結果の実部をとることで (図中の④)、-15 度方向の基底を作り出している。しかし、冗長度が高いため、画像符号化における適用範囲はある程度限定的となる。

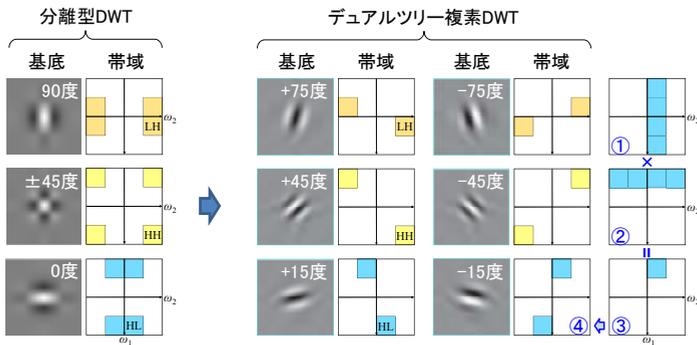


図 5・13 Dual Tree Complex Wavelet は 6 種類の方向を弁別できる

方向弁別性を高める別の方法としては、図 5・14 に示す Ridgelet 変換がある。まず、画像に 2 次元のフーリエ変換を適用した後、斜め方向に (図中の角度  $\theta$  で) スライスしてラインを得て、それぞれの角度におけるラインに対し、1 次元の逆フーリエ変換を適用する。これは Radon 変換と呼ばれ、CT スキャンなどに応用されている。次に、各角度において 1 次元のウェーブレット変換を適用することで、Ridgelet の変換係数を得る。

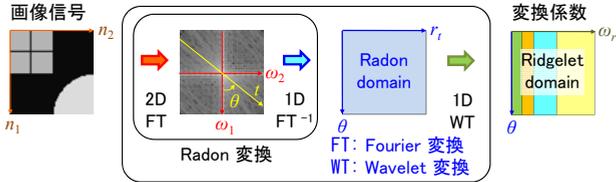


図 5・14 Ridgelet 変換

Ridgelet 変換をマルチスケールに拡張することで生じる冗長性を削減するため、図 5・15 の Curvelet 変換が提案された。画像に 2 次元のウェーブレット変換を適用し、方向性の弁別にも利用できる高域を取り出し、これらをブロックに細分割したうえで Ridgelet 変換を適用し、方向を更に細分化する。しかし、Radon 変換における斜め方向のスライスを、離散化された  $(\omega_1, \omega_2)$  平面上で実施すると誤差が生じるため、デジタル化には必ずしも適さない。

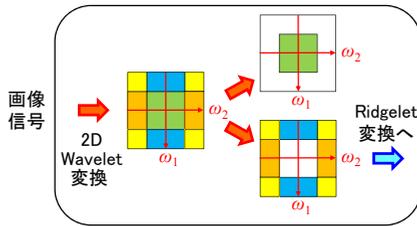


図 5・15 Curvelet 変換

離散化された 2 次元信号に対して完全再構成を保証する方法として、図 5・16 に示す Contourlet 変換が提案された。まず、画像に Laplacian Pyramid を適用し、低い周波数帯域と高い周波数帯域に分離する。この際、低域の信号は間引くが、高域は間引かない。更に、方向をより詳細に弁別するため、高域には方向性フィルタバンク (Directional Filter Bank) を適用する。このフィルタバンクは、サイコロの五の目状の間引き (Quincunx Subsampling) とファンフィルタにより方向を細分化する。また、冗長性もない。しかし、Laplacian Pyramid において高域は間引かないため、Contourlet 変換全体としては冗長となっている。

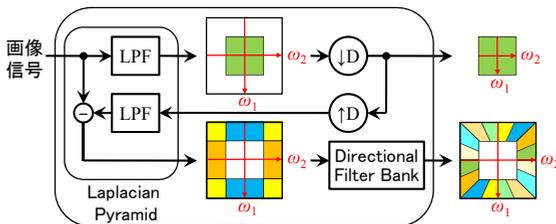


図 5・16 Contourlet 変換の処理手順

以上は、方向を細かく弁別するための基底をあらかじめ規定しておくアプローチである。これに対し、1 段目で入力画像ごとに方向を調査・抽出し、2 段目で規定の変換を適用するア

プローチも提案されている。1 段目の調査結果は画像ごとに異なるため、この情報はオーバーヘッドとして別途符号化する必要があるが、2 段目の変換には簡易で安定なものを利用できる。例えば、Curved Wavelet や Adaptive Directional Lifting Wavelet では、小さなブロックごとに最も強い方向を検出し、その線に沿って 1 次元の DWT を適用する。また、分離型 2 次元 DWT を適用しておよその方向を検出した後、この方向に沿って 1 次元 DWT を適用する Combined Directional Filter Bank も提案されている。

x-let には、Bandlet, Wedgelet, Beamlet, Brushlet, Grouplet など、ほかにも多くの種類がある。

#### ■参考文献

- 1) N. S. Jayant, Peter Noll, "Digital Coding of Waveforms: Principles and Applications to Speech and Video," Prentice-Hall, 1984.
- 2) P. P. Vaidyanathan, "Multirate Systems and Filter Bank," Prentice-Hall, 1993.
- 3) V. Britanak, P. Yip, R. Rao, "Discrete Cosine and Sine Transform: General Properties, Fast Algorithms and Integer Approximations," Elsevier, 2007.
- 4) H. Kiya, M. Yae, M. Iwahashi, "Linear phase two channel filter bank allowing perfect reconstruction," IEEE Proc. International Symposium, Circuits, Systems (ISCAS), no.2, pp.951-954, May 1992.
- 5) W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," Technical report, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1994.
- 6) ISO/IEC FCD 15444-1, Joint Photographic Experts Group, "JPEG2000 image coding system," March 2000.
- 7) S. Muramatsu, D. Han, T. Kobayashi, H. Kikuchi, "Directional lapped orthogonal transform: theory and design," IEEE Trans. Image Processing, vol.21, no.5, pp.2434-2448, May 2012.
- 8) I. W. Selesnick, R. G. Baraniuk, N. G. Kingsbury, "The dual-tree complex wavelet transform," IEEE Signal Processing Magazine, vol.22, no.6, pp.123-151, 2005.
- 9) M. N. Do, M. Vetterli, "The finite ridgelet transform for image representation," IEEE Trans. Image Processing, vol.12, no.1, pp.16-28, Jan. 2003.
- 10) J. L. Starck, E. J. Candès and D. L. Donoho, "The curvelet transform for image denoising," IEEE Trans. Image Processing, vol.11, no.6, pp. 670?684, Jun. 2002.
- 11) M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," IEEE Trans. Image Processing, vol.14, no.12, Dec. 2005.
- 12) R. H. Bamberger, M. J. T. Smith, "A filter bank for the directional decomposition of images: Theory and design," IEEE Trans. Signal Process., vol.40, no.4, pp.882-893, Apr. 1992.
- 13) D. Wang, L. Zhang, A. Vincent, F. Speranza, "Curved wavelet transform for image coding," IEEE Trans. Image Processing vol.15, no.8, pp.2413-2424, 2006.
- 14) W. Ding, F. Wu, X. Wu, S. Li, H. Li, "Adaptive directional lifting-based wavelet transform for image coding," IEEE Trans. Image Processing, vol.16, no.2, pp.416-427, 2006.
- 15) Y. Tanaka, M. Ikehara, T. Q. Nguyen, "Multiresolution image representation using combined 2-D and 1-D directional filter banks," IEEE Trans. Image Processing, vol.18, no.2, pp.269-280, Feb. 2009.