

The Application of the GHM Multiwavelet Transform to Image Processing

同志社大学 工学部 溝畑 潔 (Kiyoshi Mizohata)
 Faculty of Engineerings,
 Doshisha University

1. Introduction

この論文ではコンパクトサポートを持つ対称なマルチウェーブレットの応用について論じる。一般にコンパクトサポートを持つウェーブレットは対称性を持たないが、1994年、Geronimo と Hardin 及び Massopust はコンパクトサポートを持つ対称なマルチウェーブレットを構成した [1]。(簡単のためにこれを GHM マルチウェーブレットと呼ぶ)。その後、Strela と Heller 及び Strang 達によって画像処理への応用について論じられた [3]。離散データに対するマルチウェーブレットの応用において問題になるのがデータの前処理についてである。画像データなどの離散データに対して Mallat 変換を行う時に各 scaling function の係数を決めるのを前処理と呼ぶが、いかなる方法が良いのかまだ未解決である。

この論文では新しい前処理を提案して、従来の前処理及び他のウェーブレットによる結果との比較を行う。

2. GHM マルチウェーブレットの構築

後の前処理の導入のために、GHM マルチウェーブレットの構築について簡単に述べておく。(くわしくは [1],[2] を参照)

Definition 2.1.

$$\{(\psi_\delta)_{j,k} = 2^{\frac{j}{2}} \psi_\delta(2^j x - k)_{\delta=\{1,2,\dots,n\} j,k \in \mathbf{Z}}\}$$

が $L^2(\mathbf{R})$ の正規直交基底になるとき、 $\psi_1, \psi_2, \dots, \psi_n$ をマルチウェーブレットという。

従来のウェーブレットはウェーブレット関数が 1 個であるが、マルチウェーブレットは複数のウェーブレット関数から構成される。GHM マルチウェーブレットは次の Hat 関数 $h(x)$ から構成される。

$$(2.1) \quad h(x) = \begin{cases} 0 & (|x| > 1) \\ x+1 & (-1 \leq x \leq 0) \\ -x+1 & (0 \leq x \leq 1) \end{cases}$$

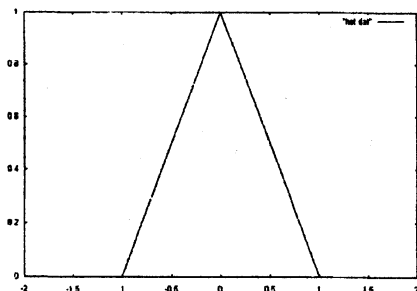


図 1 : hat 関数

まずはこの Hat 関数を分解する。 $w(x)$ は $[0, 1]$ にサポートをもつ 2 乗可積分関数として $\phi_1(x), \phi_2(x)$ を次のように定義する。

$$(2.2) \quad \begin{aligned} \phi_1(x) &= w(x) \\ \phi_2(x) &= h(x) - \frac{\langle h, w \rangle}{\langle w, w \rangle} w(x) - \frac{\langle h, w(\cdot+1) \rangle}{\langle w, w \rangle} w(\cdot+1). \end{aligned}$$

但し \langle, \rangle は内積を表す。サポートの大きさに注意すると、 $\{\phi_1(x-j)\}_{j \in \mathbb{Z}}$ 、 $\{\phi_2(x-j)\}_{j \in \mathbb{Z}}$ が直交系になるための必要十分条件は

$$(2.3) \quad \langle h, h(\cdot-1) \rangle = \frac{\langle h, w \rangle \langle w, h(\cdot-1) \rangle}{\langle w, w \rangle}$$

と与えられる。そこでこの $w(x)$ をうまくつくれば良いわけである。例えば $w(x)$ が $x = \frac{1}{2}$ に関して対称であれば ϕ_2 も対称になる。以上に着目して $w(x)$ を

$$(2.4) \quad w\left(\frac{x}{2}\right) = h(x-1) + s_0 w(x) + s_1 w(x-1)$$

を満たすように作る。すると s_0, s_1 が (2.3) を満たすための条件は

$$(2.5) \quad \begin{aligned} 8 + 12(s_0 + s_1) - 28(s_0^2 + s_1^2) - 14s_0s_1 + 6(s_0^3 + s_1^3) + 18(s_0^2s_1 + s_0s_1^2) \\ + 2(s_0^4 + s_1^4) + 2(s_0^4 + s_1^4) - 7(s_0^3s_1 + s_0s_1^3) = 0 \end{aligned}$$

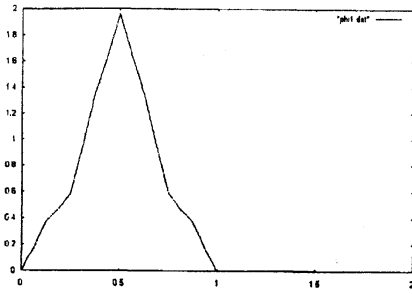
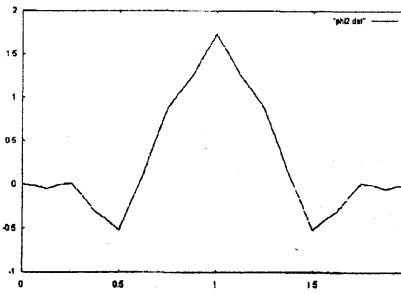
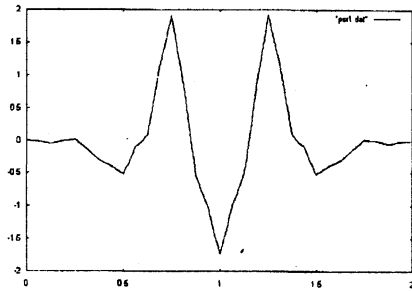
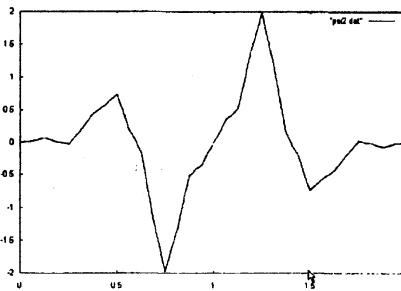
となる。 $w(x)$ を対称にするために $s_0 = s_1$ とおくと (2.5) より $s_0 = s_1 = -\frac{1}{5}$ となる。次に normalize をすると GHM マルチウエーブレットのスケーリング関数 ϕ_1, ϕ_2 が構成できる。更に (2.2) より次の関係式が出る。

$$(2.6) \quad \frac{1}{2}\phi_1(x) + \frac{1}{2}\phi_1(x+1) + \phi_2(x) = \sqrt{3}h(x)$$

以上の議論を進めると GHM マルチウエーブレットのスケーリング関数 ϕ_1, ϕ_2 と対応するウエーブレット関数 ψ_1, ψ_2 のみたすべき dilation equation が次のように導き出される。

$$(2.7) \quad \begin{aligned} \begin{bmatrix} \phi_1(x) \\ \phi_2(x) \end{bmatrix} &= \begin{bmatrix} \frac{3}{5} & \frac{4\sqrt{2}}{5} \\ -\frac{1}{10\sqrt{2}} & -\frac{3}{10} \end{bmatrix} \begin{bmatrix} \phi_1(2x) \\ \phi_2(2x) \end{bmatrix} + \begin{bmatrix} \frac{3}{5} & 0 \\ \frac{1}{10\sqrt{2}} & 1 \end{bmatrix} \begin{bmatrix} \phi_1(2x-1) \\ \phi_2(2x-1) \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 \\ \frac{9}{10\sqrt{2}} & -\frac{3}{10} \end{bmatrix} \begin{bmatrix} \phi_1(2x-2) \\ \phi_2(2x-2) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -\frac{1}{10\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} \phi_1(2x-3) \\ \phi_2(2x-3) \end{bmatrix} \\ \begin{bmatrix} \psi_1(x) \\ \psi_2(x) \end{bmatrix} &= \begin{bmatrix} -\frac{1}{10\sqrt{2}} & -\frac{3}{10} \\ \frac{1}{10} & \frac{3\sqrt{2}}{10} \end{bmatrix} \begin{bmatrix} \psi_1(2x) \\ \psi_2(2x) \end{bmatrix} + \begin{bmatrix} \frac{9}{10\sqrt{2}} & -1 \\ -\frac{9}{10} & 0 \end{bmatrix} \begin{bmatrix} \psi_1(2x-1) \\ \psi_2(2x-1) \end{bmatrix} \\ &+ \begin{bmatrix} \frac{9}{10\sqrt{2}} & -\frac{3}{10} \\ \frac{9}{10} & -\frac{3\sqrt{2}}{10} \end{bmatrix} \begin{bmatrix} \psi_1(2x-2) \\ \psi_2(2x-2) \end{bmatrix} + \begin{bmatrix} -\frac{1}{10\sqrt{2}} & 0 \\ -\frac{1}{10} & 0 \end{bmatrix} \begin{bmatrix} \psi_1(2x-3) \\ \psi_2(2x-3) \end{bmatrix} \end{aligned}$$

(2.7) で定まる GHM マルチウエーブレットのスケーリング関数 ϕ_1, ϕ_2 と対応するウエーブレット関数 ψ_1, ψ_2 のグラフを描くと次のようになる $\phi_1(x), \phi_2(x), \psi_1(x)$ は対称だが、 $\psi_2(x)$ は反対称である。

図2 : $\phi_1(x)$ 図3 : $\phi_2(x)$ 図4 : $\psi_1(x)$ 図5 : $\psi_2(x)$

3. 離散データの前処理

離散データ $\{x[k]\}$ が与えられているとする。Mallat 変換ではその離散データに対して

$$(3.1) \quad f(x) = \sum_k x[k] \phi(x - k) \quad (\phi(x) : \text{scaling function}).$$

を元のデータと考慮して wavelet 変換を行い、その係数を対応する離散データとみなす。マルチウェーブレットの場合、scaling function が複数個あるために注意が必要である。例えば今回あつまっている GHM マルチウェーブレットの場合、 ϕ_1, ϕ_2 と 2 個の scaling function がある。離散データが与えられた場合に、これから 2 個の scaling function の係数を決めるのを前処理と呼ぶ。即ち離散データ $\{x[k]\}$ に対して

$$(3.2) \quad f(x) = \sum_k \alpha[k] \phi_1(x - k) + \sum_k \beta[k] \phi_2(x - k)$$

の係数 $\alpha[k], \beta[k]$ を決定するのが前処理という。もっとも単純な方法は Repeated row と呼ばれる方法で、単純に

$$(3.3) \quad \alpha[k] = \beta[k] = x[k]$$

とおく。ただしこの方法ではデータ量が 2 倍になるために画像圧縮などでは使えない。

Strela 達が論文 [3] で提案しているのは

$$(3.4) \quad f(n) = x[2n], \quad f\left(n + \frac{1}{2}\right) = x[2n + 1]$$

を満たすようにとる方法である。(これを前処理 I と呼ぶ) これは 2 個の scaling function のサポートの長さが 1 と 2 であるのを巧みに用いている。係数 $\alpha[k], \beta[k]$ は次のように決

定される。

$$(3.5) \quad \begin{aligned} \alpha[n] &= \frac{\phi_2(1)x[2n+1] - \phi_1(\frac{1}{2})(x[2n] + x[2n+2])}{\phi_2(1)\phi_1(\frac{1}{2})} \\ \beta[n] &= \frac{x[2n+2]}{\phi_2(1)} \end{aligned}$$

この方法では $\alpha[n]$ を決定するのに $x[2n], x[2n+1], x[2n+2]$ の3つのデータが必要となる。これは画像圧縮において平均二乗誤差が増える要因になると思われる。

本研究で提案したいのは(2.6)に着目した”重みつき”の前処理である。(2.6)より次の関係式が導き出される。

$$(3.6) \quad \sum_{k \in \mathbf{Z}} \sqrt{2}\phi_1(x-k) + \sum_{k \in \mathbf{Z}} \phi_2(x-k) \equiv \text{constant}$$

この(3.6)を用いて次のような前処理を提案する。(これを前処理IIと呼ぶ)

$$(3.7) \quad \begin{aligned} \alpha[n] &= \sqrt{2}x[2n+1] \\ \beta[n] &= x[2n] \end{aligned}$$

(3.7)の前処理の場合、離散データが全て同じ値であるならば(3.2)で定義される $f(x)$ は定数関数になる。

4. シミュレーションの結果

本研究ではGHM マルチウェーブレットを用いて画像圧縮した結果を、前処理Iと今回提案した前処理IIを用いた場合で比較した。また従来のウェーブレットと比較するために D_2 -wavelet(Daubechies, $N=2$)と、GHM マルチウェーブレットと同じ精度($=2$)をもつ D_4 -wavelet(Daubechies, $N=4$)とも比較した。量子化の方法は色々考えられるが、ウェーブレット係数の影響を知りたいので次のような量子化を行った。画像を2段階、Mallat変換してから、ウェーブレット係数の絶対値の大きいのを一定割合ピックアップして他はすべて0に置き換え、逆変換で画像を復元してから元の画像との誤差を評価した。使用した画像は以下の4例である。いずれもサイズが 512×512 の白黒画像であり、画素は0から255までの整数値となっている。復元画像の質を評価するために、平均自乗誤差、ピークSN比、最大誤差という代表的な3つの誤差を用いた。平均自乗誤差と最大誤差は低い方が画像の質が良く、逆にピークSN比は高いほど画像の質が良い。表のrateは選んだウェーブレット係数の割合で、例えばrate=0.01の場合はウェーブレット係数の絶対値の大きいのを全体の1%選び、残りは全て0にした事を意味する。



図6 Barbara



図7 Lena

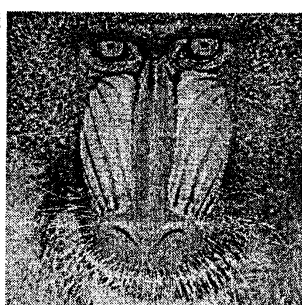


図8 Mandrill

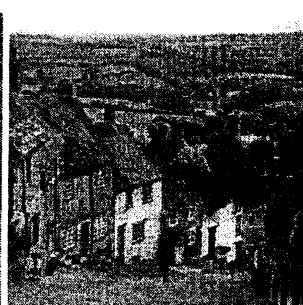


図9 Gold

画像 : Barbara

画像 : Barbara	D_2	D_4	GHM(前処理I)	GHM(前処理II)
平均自乗誤差 (rate=0.01)	205.7	201.2	281.8	225.5
平均自乗誤差 (rate=0.03)	144.5	134.7	209.4	156.7
平均自乗誤差 (rate=0.05)	105.5	97.7	165.4	116.5
平均自乗誤差 (rate=0.1)	54.4	49.6	98.4	63.3
ピーク SN 比 (dB)(rate=0.01)	25.0	25.1	23.6	24.6
ピーク SN 比 (dB)(rate=0.03)	26.5	26.8	24.9	26.2
ピーク SN 比 (dB)(rate=0.05)	27.9	28.2	25.9	27.5
ピーク SN 比 (dB)(rate=0.1)	30.1	31.2	28.2	30.1
最大誤差 (rate=0.01)	101.1	103.41	115.8	121.0
最大誤差 (rate=0.03)	73.6	79.0	106.3	99.3
最大誤差 (rate=0.05)	68.0	58.5	90.1	80.6
最大誤差 (rate=0.1)	51.0	46.4	62.7	55.5

画像 : Lena

画像 : Lena	D_2	D_4	GHM(前処理I)	GHM(前処理II)
平均自乗誤差 (rate=0.01)	59.9	56.7	142.1	91.0
平均自乗誤差 (rate=0.03)	37.8	36.4	97.8	60.2
平均自乗誤差 (rate=0.05)	26.1	25.4	74.4	42.2
平均自乗誤差 (rate=0.1)	13.8	13.0	38.7	21.8
ピーク SN 比 (dB)(rate=0.01)	30.4	30.6	26.6	28.5
ピーク SN 比 (dB)(rate=0.03)	32.4	32.5	28.2	30.3
ピーク SN 比 (dB)(rate=0.05)	34.0	34.1	29.4	31.9
ピーク SN 比 (dB)(rate=0.1)	36.7	37.0	32.3	34.7
最大誤差 (rate=0.01)	73.8	74.3	115.3	98.3
最大誤差 (rate=0.03)	54.2	44.6	80.0	63.6
最大誤差 (rate=0.05)	41.7	34.2	65.7	54.9
最大誤差 (rate=0.1)	25.0	20.9	37.2	32.6

画像 : Mandrill

画像 : Mandrill	D_2	D_4	GHM(前処理I)	GHM(前処理II)
平均自乗誤差 (rate=0.01)	407.3	396.9	538.1	455.5
平均自乗誤差 (rate=0.03)	316.1	306.6	454.6	369.3
平均自乗誤差 (rate=0.05)	257.5	251.5	393.6	309.1
平均自乗誤差 (rate=0.1)	164.7	158.8	279.5	205.4
ピーク SN 比 (dB)(rate=0.01)	22.0	22.1	20.8	21.5
ピーク SN 比 (dB)(rate=0.03)	23.1	23.3	21.6	22.5
ピーク SN 比 (dB)(rate=0.05)	24.0	24.1	22.2	23.2
ピーク SN 比 (dB)(rate=0.1)	26.0	26.1	23.7	25.0
最大誤差 (rate=0.01)	113.1	127.4	150.7	142.2
最大誤差 (rate=0.03)	104.1	94.2	150.8	133.3
最大誤差 (rate=0.05)	90.1	89.8	111.5	139.6
最大誤差 (rate=0.1)	69.7	67.2	93.7	93.0

画像 : Gold

画像 : Gold	D_2	D_4	GHM(前処理I)	GHM(前処理II)
平均自乗誤差 (rate=0.01)	85.0	85.3	161.0	115.1
平均自乗誤差 (rate=0.03)	61.64	60.6	125.2	84.1
平均自乗誤差 (rate=0.05)	48.0	41.2	102.4	68.2
平均自乗誤差 (rate=0.1)	31.0	30.5	66.0	42.8
ピーク SN 比 (dB)(rate=0.01)	28.8	28.8	26.1	27.5
ピーク SN 比 (dB)(rate=0.03)	30.2	30.3	27.2	28.9
ピーク SN 比 (dB)(rate=0.05)	31.3	31.4	28.0	29.8
ピーク SN 比 (dB)(rate=0.1)	33.2	33.3	29.9	31.8
最大誤差 (rate=0.01)	64.0	69.5	103.0	97.8
最大誤差 (rate=0.03)	54.4	54.8	81.1	76.9
最大誤差 (rate=0.05)	38.8	41.2	65.9	60.9
最大誤差 (rate=0.1)	34.6	30.5	48.2	37.3

これらの結果から、 D_2 と D_4 ではあまり違いはないが、平均自乗誤差やピーク SN 比 (dB)、最大誤差では GHM マルチウェーブレットを用いたよりいい結果がでている。更に前処理 I と前処理 II では、前処理 II の方がいい結果を出しているのがわかる。特に平均自乗誤差やピーク SN 比 (dB) では前処理 II の方が完全に良い結果を出しているが、残念ながら D_2 や D_4 より劣っている。しかし画像の質 (数値には表れないが) は GHM の方が良いように思える。その中でも前処理 II の方が前処理 I よりも画像の質が高いと判断できる。画像の質の差がでやすいように rate=0.01 の場合の Lena の復元画像を顔の部分で比較したのが次の画像である。



図10 Lena, D_2 , rate=0.01



図11 Lena, D_4 , rate=0.01



図12 Lena, GHM(前処理I), rate=0.01



図13 Lena, GHM(前処理II), rate=0.01

図10、図11の画像は口元や目のあたりが大きなゆがみが生じているが、図12、図13はの画像は比較的良く復元されている。更に前処理IIを用いた図13の画像の方が図12の画像より画質がシャープで良質である。

5. 結論

復元した画像と元の画像との誤差では、今回提案した前処理IIの方が前処理Iよりいい結果がでている。特に誤差を示す代表的な数値ピーク SN 比 (dB) に関しては全ての場合で前処理IIの方が高い数値となった。ただし D_2 や D_4 と比べると数値の上では悪い結果が出た。原因はおそらく2つの異なる scaling function と wavelet function を用いるためと予想され、おそらく他の量子化を用いても同じ結果がでると思われる。但し復元画像の質(数値には表れないが)は前に述べたように前処理IIが一番良さそうである。

復元画像の画質について考察すると D_2 と D_4 の場合、rate が小さい場合大きなゆがみが生じるのに対してGHM マルチウェーブレットの場合、ゆがみは小さいが輪郭に粒状のノイズが表れる。この粒状のノイズはGHM マルチウェーブレットの scaling function, wavelet function が複数で異なる形状(特に wavelet function の場合、一方が対称なのに対してもう一方が反対称)なのに全く同じ量子化を行ったためと思われる。通常のウェーブレットの場合、一回 Mallat 変換すると4つにデータが分解されるが、GHM マルチウェーブレットの場合16に分解される。そこで wavelet function の形状の違いに注意して各パートの量子化を決めるともっと良い結果が出ると思われる。

References

- [1] J. Geronimo, D. Hardin, and P. R. Massopust, *Fractal functions and wavelet expansions based on several scaling functions*, J. Approx. Theory., 78 (1994), 373-401.
- [2] J. Geronimo, D. Hardin, P. R. Massopust, and G. Donovan, *Construction of orthogonal wavelets using fractal interpolation functions*, SIAM J. Math. Anal., 27 (1996), 1158-1192.
- [3] V. Strela, P. Heller, G. Strang, et al., *The Application of multiwavelet filter banks to image processing*, IEEE Trans. Image. Processing., 8 (1999), 548-563.