3次元ウェーブレット変換・逆変換の計算式と計算例

小口, 雄康 / OGUCHI, Yuko

(出版者 / Publisher)
法政大学計算科学研究センター

(雑誌名 / Journal or Publication Title)
法政大学計算科学研究センター研究報告 / Bulletin of Computational Science Research Center, Hosei University

(巻 / Volume)
13

(開始ページ / Start Page)
233

(終了ページ / End Page)
240

(発行年 / Year)
2000-03-31

(URL)
https://doi.org/10.15002/00024900

3 次元ウェーブレット変換 · 逆変換の計算式と計算例

小口　雄康
計算科学研究センター

信号処理においてウェーブレット変換 · 逆変換は有力な手法である．1·2 次元データに関しては比較的簡単に計算できるし，市販のソフトウェアーも手軽に利用できる．しかし，3 次元となると事情が異なる．いくつかの教科書や報告に解説が見られるが簡単に理解できるようには説明されていない．この点を考慮して 3 次元ウェーブレット変換 · 逆変換の計算式を 2 次元からの直接の発展として整理して示した。特に、行列演算の解釈に重点をおいて計算手続きを説明してある。最後に、計算例を示し、計算の正しいことを確かめてある．

1. はじめに

離散値 · 正規直交ウェーブレット変換で 1 次元から 2 次元，3 次元と次元数を増して行くときの計算アルゴリズムを統一的に解り易く説明できないかと常々考えてきた．1·2 次元変換は簡単に計算できるようになった．3 次元についても，最近その応用が有効であることが示された (松山他 [1])．しかし，その計算手続きは十分理解できるようには説明されていない．もちろん，Daubechies の教科書にはきちんと説明されているが，それを理解し，実際の計算にまで発展できるか，はそう簡単ではない．今回，1 つの筋道で非常に簡単に解釈できるようになったので以下に報告する．ただし，1 次元ウェーブレット変換について計算手続きを十分理解している前提で書いてある．

2. 1 次元ウェーブレット変換

2.1 データを縦行列と考える場合

いま，対象とする 1 次元データを縦行列データであると考える．

$$M_{m1} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{m1} \end{bmatrix}$$

ウェーブレット変換行列を W_{mm} とすると、ウェーブレット変換 $\tilde{M}$ は以下のように表される．

$$\tilde{M}_{m1} = W_{mm} M_{m1} \tag{1}$$

(1) が普通なされる 1 次元ウェーブレット変換の定義である．

1 次元ウェーブレット変換で実際の演算は (1) を用いて行われる．変換の中身は，和をとる演算子 ϕ_m(スケーリング関数) と差をとる演算子 ψ_m(ウェーブレット) によって，$\tilde{M}_{m1}$ は以下のように分離される．

$$\tilde{M}_{m1} = \begin{pmatrix} \phi_m M_{m1} \\ \psi_m M_{m1} \end{pmatrix} \tag{2}$$

また、同じことであるが次のような表現もなされる．

$$\begin{aligned}\tilde{M}_{m1} &= \begin{pmatrix} S_m \\ D_m \end{pmatrix} \\ S_m &= \phi_m M_{m1} \\ D_m &= \psi_m M_{m1} \qquad (3)\end{aligned}$$

ここで、S_m は m についての和を、D_m は m についての差(これがウェーブレット変換である)をとったものである．これらは down sampling されてデータ数が $m/2$ となっている．

ところで，(3) の計算はどうなされるかを考える．

例えば，実際には，

$$S_m = \phi_m M_{m1} = \sum_{m=1}^{m} \phi_m M_{m1} \qquad (4)$$

のように，単に内積をとる，すなわち，掛け算をして和をとるだけである．

なお，式を見易くするために混乱が生じないときは，添字を引数と見なしたり，最大値と見なしたりして特別に区別しない．

2.2 データを横行列と考える場合

次に，n 個のデータ列を横行列とした場合の 1 次元ウェーブレット変換を考えよう．

$M_{1n} = [a_{11}, a_{12}, a_{13}, \ldots, a_{1n}]$ と置く．

ウェーブレット変換するためにはデータを転置して縦行列としなければならない．すなわち，

$$\tilde{M}_{1n}^T = W_{nn} M_{1n}^T \qquad (5)$$

でウェーブレット変換が表現できる．

ただし，^{T} は 2 次元行列の転置を表し，一般に，$M_{mn}^T = M_{nm}$ である．

(5) の右辺から得られた 1 次元ウェーブレット変換は縦行列であるから，与えられたデータと同じ横行列に戻すには，転置をとらなければならない．すなわち，1 次元ウェーブレット変換は以下で完成する ((5) の左辺に転置を用いた理由である)．

$$\begin{aligned}\tilde{M}_{1n} &= (W_{nn} M_{1n}^T)^T \qquad (6) \\ &= M_{1n} W_{nn}^T \qquad (7)\end{aligned}$$

ここで，(6) から (7) への計算で 2 次元行列の転置公式

$$\begin{aligned}(A_{lm} B_{mn})^T &= B_{mn}^T A_{lm}^T \\ &= B_{nm} A_{ml} \quad (n \times l \text{ 行列}) \qquad (8)\end{aligned}$$

を用いている．

(3)，(4) に対応する式は以下となる．

$$\begin{aligned}\tilde{M}_{1n} &= \begin{pmatrix} S_n \\ D_n \end{pmatrix}^T \\ S_n &= \phi_n M_{1n}^T = \phi_n M_{n1} = \sum_{n=1}^{n} \phi_n M_{1n} \\ D_n &= \psi_n M_{1n}^T = \psi_n M_{n1} = \sum_{n=1}^{n} \psi_n M_{1n} \qquad (9)\end{aligned}$$

(1) と比べて，転置はデータを縦行列に直す形式上の表現であって，実質上は同じ計算をしているだけのことであること，が分かる．

データが横か縦かは本質的な問題ではないことを認識することは，この後の多次元への拡張で重要なことである．

式の表現が簡単であるから，ここでは縦行列と考えることにしよう．

1 次元ウェーブレット変換の定義式は非常に簡単であるが，多次元の変換も実際の計算は 1 次元変換を組み合わせて行われることが以下に示される．したがって，1 次元ウェーブレット変換の計算式をきちんと作ることが重要である．

1. ϕ，ψ とデータとの積は，データ数が足りない場合どう処理されているか，
2. 和，差の部分がきちんと分離されて所定の位置に書き込まれているか，

は特に重要である．

3. 2 次元ウェーブレット変換

3.1 定義

データが M_{mn} と与えられているとしよう．すなわち，m 個の縦行列データが独立に横に n 組与えられているとみなす．各縦行列を 1 次元ウェーブレット変換するとしてこれを以下のように書く．

$$W_{mm}M_{mn} \qquad (\ m \times n\ \text{行列})$$

ここで，M の左右の添字 m が等しくないと行列の掛け算ができないので，ウェーブレット変換もできないことに注意．これが転置をとる必要を生むのである．

次に，n 個のデータからなる横行列を 1 次元ウェーブレット変換するとしよう．変換できるためには転置をとらなければならない．その結果 $n \times m$ 行列が得られるので元の配列に戻すためにふたたび転置をとることにしよう．この結果を 2 次元ウェーブレット変換と定義する．

すなわち，以下で 2 次元ウェーブレット変換を定義する．

$$\begin{aligned} \tilde{M}_{mn} &= (W_{nn}(W_{mm}M_{mn})^T)^T & (10)\\ &= W_{mm}M_{mn}W_{nn}^T & (11)\\ &= W_{mm}(W_{nn}M_{mn}^T)^T & (12) \end{aligned}$$

また，和, 差の表現は以下のようになる．

$$\begin{aligned} \tilde{M}_{mn} &= \begin{pmatrix} S_mS_n & S_mD_n \\ D_mS_n & D_mD_n \end{pmatrix} & (13)\\ S_mS_n &= \phi_m\phi_nM_{mn} \\ S_mD_n &= \phi_m\psi_nM_{mn} \\ D_mS_n &= \psi_m\phi_nM_{mn} \\ D_mD_n &= \psi_m\psi_nM_{mn} & (14) \end{aligned}$$

ここで，例えば，

$$S_mS_n = \sum_{m=1}^{m}\sum_{n=1}^{n}\phi_m\phi_nM_{mn} \qquad (15)$$

と計算されるから，(14) で添字の等しい項はその添字について和をとると解釈する．こう考えると，変換で m，n の順序を考える必要がなくなり，転置をとる，とらないには関係なくなってしまう．これも，1 次元と同じである．

3.2 システム方程式への適用としての定義

改めて，2 次元ウェーブレット変換の定義をシステム方程式を解くという観点から見てみよう．

線形系で n 個の未知数 x_n が m 個の観測値 y_m とシステム行列 M_{mn} から得られるとしよう．

$$M_{mn}x_n = y_m \qquad (16)$$

両辺にウェーブレット変換行列を掛けて，

$$W_{mm}M_{mn}x_n = W_{mm}y_m \qquad (17)$$

一方，

$$\tilde{X}_n = W_{nn}x_n,\ \tilde{Y}_m = W_{mm}y_m \qquad [\ (1)\ \text{参照}\ ]$$

と置くと，(17) は以下のように書き直される．

$$W_{mm}M_{mn}W_{nn}^T\tilde{X}_n = \tilde{Y}_m$$

(16) と比べて，2 次元ウェーブレット変換を以下のように定義することができる．

$$\tilde{M}_{mn} = W_{mm}M_{mn}W_{nn}^T \qquad (18)$$

(18) は (11) と一致し，考え方にはよらない．この点を明確にしないと 2 次元ウェーブレット変換の導入で混乱を生ずる原因となる．

4. 3 次元ウェーブレット変換

3 次元ウェーブレット変換への拡張は (10)，(12) のどちらかによると理解し易い．

4.1 (10) 式の拡張

初めに (10) を参照して 3 次元へ拡張する．

3 次元データが M_{lmn} と与えられているとする．

まず，(10) の添字を書き直す．

$$\tilde{M}_{lm} = (W_{mm}(W_{ll}M_{lm})^T)^T$$

これから推定し，3 次元ウェーブレット変換へ以下

のように拡張する．

$$\tilde{M}_{lmn} = (W_{nn}(W_{mm}(W_{ll}M_{lmn})^T)^T)^T \tag{19}$$

ここで，3 次元ウェーブレット変換における行列の転置則は決められていないから変換が可能となるよう明確にして置こう．

変換が合理的に行われるためには隣り合った添字が等しくなければならないことを原則とする．

(19) において，

$$\begin{array}{rcl} W_{ll}M_{lmn} & \Rightarrow & M^{(1)}_{lmn} \qquad \text{から，} \\ (W_{ll}M_{lmn})^T & = & M^{(1)T}_{lmn} \;\Rightarrow\; M^{(1)}_{mnl} \end{array}$$

となると考えれば，次に W_{mm} を掛け、かつ転置をとることができる．

まとめて，以下のように転置則を定める．

$$M^T_{lmn} = M_{mnl} \tag{20}$$
$$M^T_{mnl} = M_{nlm} \tag{21}$$
$$M^T_{nlm} = M_{lmn} \tag{22}$$

すなわち，3 次元ウェーブレット変換における転置は先頭の添字を最後へ持って行く操作である．

また，3 回の転置で添字が元の順序へ戻るので 3 次元ウェーブレット変換の定義も合理的であると考えられる．

一方，逆変換を考えるときは (20)—(22) の M を変換行列 $\tilde{M}$ と見なして右辺の転置が左辺の転置と等しくなければなければならないと考える．すなわち，逆変換では，変換の場合の逆操作をして最後の添字を先頭に持って行かねばならないことが解る．

変換の転置と逆変換の転置は添字の並べ替え方が異なるので，両者を区別するために逆変換の場合を τ で表そう．以下が逆変換の転置則である．

$$\tilde{M}^\tau_{mnl} = \tilde{M}_{lmn} \tag{23}$$
$$\tilde{M}^\tau_{nlm} = \tilde{M}_{mnl} \tag{24}$$
$$\tilde{M}^\tau_{lmn} = \tilde{M}_{nlm} \tag{25}$$

3 次元ウェーブレット変換の和，差の表現は (13), (14) を参照して以下のようにまとめられる．ただし，3 次元行列であるから、2 次元行列のようには表せない．$\tilde{M}$ を一応以下のような表示をして置くが各項の順序は適宜示しただけである．

$$\begin{pmatrix} S_lS_mS_n & S_lS_mD_n & S_lD_mS_n & S_lD_mD_n \\ D_lS_mS_n & D_lS_mD_n & D_lD_mS_n & D_lD_mD_n \end{pmatrix}$$

1·2 次元と同様に，実際の計算は例えば以下でなされる．

$$S_lS_mS_n = \sum_{l=1}^{l}\sum_{m=1}^{m}\sum_{n=1}^{n}\phi_l\phi_m\phi_n M_{lmn} \tag{26}$$

3 次元でも転置は実質的には意味を持たなくなる．

4.2 (12) 式の拡張

(12) を基にして 3 次元へ拡張すると以下のようになる．

$$\tilde{M}_{lmn} = W_{ll}(W_{mm}(W_{nn}M^T_{lmn})^T)^T \tag{27}$$

この場合の転置則は，明らかに，最後の添字を最初へ持って行くとすればよい．しかし，和 · 差で表すと順序が違うだけで，計算結果は (19) と同じになることは明白である．

(27) に形式的に (8) を適用して書き直してみる．

$$\begin{array}{rcl} \tilde{M}_{lmn} & = & W_{ll}(W_{mm}(W_{nn}M^T)^T \\ & = & W_{ll}(W_{nn}M^T_{lmn})^T W^T_{mm} \\ & = & W_{ll}M_{lmn}W^T_{nn}W^T_{mm} \end{array}$$

同様に，(19) を書き直すと，

$$\begin{array}{rcl} \tilde{M}_{lmn} & = & (W_{nn}(W_{mm}(W_{ll}M_{lmn})^T)^T)^T \\ & = & W_{mm}(W_{ll}M_{lmn})^T W^T_{nn} \\ & = & W_{mm}M^T_{lmn}W^T_{ll}W^T_{nn} \end{array}$$

となる．

この両者は，表現が異なるが同じ答を与える．見掛け上の差は転置則の違いによると考えられる．3 次元変換の定義はこのどちらか（またはその変形) で与えられるが，行列で定義する場合はその転置則に注意しなければならない．

最も簡単には，

$$\tilde{M}_{lmn} = W_{ll}W_{mm}W_{nn}M_{lmn} \tag{28}$$

と書いてもよい．

ただし，変換の計算は添字ごとになされるものとするが，変換の順序は任意である．

5. ウェーブレット逆変換

ウェーブレット逆変換は変換の逆演算をする．そのため，使い易い形の式を用いる．

5.1 1 次元逆変換

ここで，W_{mm} は正規直交行列であるから $W^{-1} = W^T$ であることを用いて，(1) の左側から M_{mm}^T を掛けて，

$$M_{m1} = W_{mm}^T \tilde{M}_{m1} \tag{29}$$

を得る．

この計算も多次元で使われるからタフなプログラムを作って置かねばならない．

5.2 2 次元逆変換

(10) を以下のように逆変換する．

$$\begin{aligned}
\tilde{M}_{mn}^T &= W_{nn}(W_{mm}M_{mn})^T \\
W_{nn}^T\tilde{M}_{mn}^T &= (W_{mm}M_{mn})^T \\
(W_{nn}^T\tilde{M}_{mn}^T)^T &= W_{mm}M_{mn} \\
W_{mm}^T(W_{nn}^T\tilde{M}_{mn}^T)^T &= M_{mn}
\end{aligned}$$

上の計算から 2 次元ウェーブレット逆変換は以下で計算される．

$$\begin{aligned}
M_{mn} &= W_{mm}^T(W_{nn}^T\tilde{M}_{mn}^T)^T \qquad (30) \\
&\quad (= W_{mm}^T\tilde{M}_{mn}W_{nn})
\end{aligned}$$

5.3 3 次元逆変換

(19) を以下のように逆変換する．

$$\begin{aligned}
\tilde{M}_{lmn}^\tau &= W_{nn}(W_{mm}(W_{ll}M_{lmn})^T)^T \\
W_{nn}^T\tilde{M}_{lmn}^\tau &= (W_{mm}(W_{ll}M_{lmn})^T)^T \\
(W_{nn}^T\tilde{M}_{lmn}^\tau)^\tau &= W_{mm}(W_{ll}M_{lmn})^T \\
W_{mm}^T(W_{nn}^T\tilde{M}_{lmn}^\tau)^\tau &= (W_{ll}M_{lmn})^T \\
(W_{mm}^T(W_{nn}^T\tilde{M}_{lmn}^\tau)^\tau)^\tau &= W_{ll}M_{lmn} \\
W_{ll}^T(W_{mm}^T(W_{nn}^T\tilde{M}_{lmn}^\tau)^\tau)^\tau &= M_{lmn}
\end{aligned}$$

以下が，3 次元ウェーブレット逆変換の計算式である．

$$M_{lmn} = W_{ll}^T(W_{mm}^T(W_{nn}^T\tilde{M}_{lmn}^\tau)^\tau)^\tau \tag{31}$$

6. 行列の転置則

以下にウェーブレット変換における行列の転置則についてまとめておく．

6.1 変換

1. 1 次元

$$\tilde{M}_{m1} = W_{mm}M_{m1}$$

2. 2 次元

$$\begin{aligned}
(1) \quad \tilde{M}_{mn} &= W_{mm}(W_{nn}M_{mn}^T)^T \\
(2) \quad \tilde{M}_{mn} &= (W_{nn}(W_{mm}M_{mn})^T)^T \\
M_{mn}^T &= M_{nm} \quad (\text{変換則 } 1)
\end{aligned}$$

3. 3 次元

$$\begin{aligned}
(1) \quad \tilde{M}_{lmn} &= W_{ll}(W_{mm}(W_{nn}M_{lmn}^T)^T)^T \\
M_{lmn}^T &= M_{nlm} \quad (\text{変換則 } 2) \\
(2) \quad \tilde{M}_{lmn} &= (W_{nn}(W_{mm}(W_{ll}M_{lmn})^T)^T)^T \\
M_{lmn}^T &= M_{mnl} \quad (\text{変換則 } 3)
\end{aligned}$$

6.2 逆変換

1. 1 次元

$$M_{m1} = W_{mm}^T\tilde{M}_{m1}$$

2. 2 次元

$$
\begin{aligned}
(1)\quad M_{mn} &= (W_{nn}^T(W_{mm}^T\tilde{M}_{mn})^T)^T \\
(2)\quad M_{mn} &= W_{mm}^T(W_{nn}^T\tilde{M}_{mn}^T)^T \\
\tilde{M}_{mn}^T &= \tilde{M}_{nm} \quad (\text{変換則 } 4)
\end{aligned}
$$

3. 3 次元

$$
\begin{aligned}
(1)\quad M_{lmn} &= (W_{nn}^T(W_{mm}^T(W_{ll}^T\tilde{M}_{lmn})^\tau)^\tau)^\tau \\
\tilde{M}_{lmn}^\tau &= \tilde{M}_{mnl} \quad (\text{変換則 } 5) \\
(2)\quad M_{lmn} &= W_{ll}^T(W_{mm}^T(W_{nn}^T\tilde{M}_{lmn}^\tau)^\tau)^\tau \\
\tilde{M}_{lmn}^\tau &= \tilde{M}_{nlm} \quad (\text{変換則 } 6)
\end{aligned}
$$

以上の転置則にこだわらず，次のように簡単に考えると混乱を避けることができる．

ウェーブレット変換は以下で定義する．ここで，添え字の等しいもので和をとると解釈する (アインシュタインの総和則).

$$1\text{ 次元：}\quad \tilde{M}_{m1} = W_{mm}M_{m1} \tag{32}$$
$$2\text{ 次元：}\quad \tilde{M}_{mn} = W_{nn}W_{mm}M_{mn} \tag{33}$$
$$3\text{ 次元：}\quad \tilde{M}_{lmn} = W_{nn}W_{mm}W_{ll}M_{lmn} \tag{34}$$

逆変換は，次のように書かれる．

$$1\text{ 次元：}\quad M_{m1} = W_{mm}^T\tilde{M}_{m1} \tag{35}$$
$$2\text{ 次元：}\quad M_{mn} = W_{nn}^TW_{mm}^T\tilde{M}_{mn} \tag{36}$$
$$3\text{ 次元：}\quad M_{lmn} = W_{ll}^TW_{mm}^TW_{nn}^T\tilde{M}_{lmn} \tag{37}$$

7. 計算例

3 次元ウェーブレット変換の計算を手元のコンピュータで行えるように，および，会話形で計算を進行できるように，計算には Fortran，図示には Matlab を用い，かつ，MEX プログラムで両者を結びつけた．コンピュータの能力によって取り扱えるデータは $256 \times 256 \times 32$ が最大であり，1 回の計算に 30 分以上掛かった．まず，大きさ $16 \times 32 \times 16$ の立体の中に稜だけからなる正六面体を埋め込んだモデルを作り，1 回変換し，それを逆変換した結果を以下に示す．ただし，表示の制限から明細な図を示すことはできなかった．

7.1 Daubechies 1 次 (Harr 関数) による変換

変換が見易いように Daubechies 1 次を用いた．

Fig.1. は変換した結果を 3 次元表示してある．変換全体の概略を知ることができる．

Fig.2. は Fig.1. を各セクション (SSS など) ごとに分離して 3 次元表示してある．立体内部の隠された図形が検出されている．なお，右下の INV はセクションごとに逆変換し，それらを全て加え合わせたもので内部図形を復元していることが確認できる．

Fig.3. は Fig.2. を $n = 1, 2, \ldots, 8$ ごとに 2 次元面で示した．縦，横軸を，それぞれ，n 方向，セクションにとってある．変換が正しく行われていることが明瞭に示されている．

Fig.4. は n，セクションごとに独立に逆変換した結果を示してある．右端の列は各 n ごとにセクション全ての逆変換の和を示してある．モデルの n についての断面図とみなすことができる．

7.2 Coiflet 4 次による変換

変換が複雑になる例として Coiflet 4 次を用いた．結果を Fig.5.—Fig.8. に示してある．各図は前節と同様である．前節と比べて複雑になっていて，もともと実体がないセクションに値が染み出していて aliasing がおこっている．ウェーブレットに何を用いるべきかを目的に応じて使い分けねばならないことがここに現れている．しかし，Fig.5. の逆変換 INV が Fig.2. と同じであることから計算が正しいことが分かる．Fig.9. に参考として Matlab で与えられている脳断面図のウェーブレット変換の SSS セクションを示してある．元は 5 枚からなっているが，ウェーブレット変換可能にするため n 方向に 0 を加えたために変換で aliasing をおこし，見かけ上 6 枚の断面が与えられている．もちろん，逆変換は元に戻るが詳しい議論は省略する．

8. おわりに

3 次元ウェーブレット変換 · 逆変換を分かり易い計算式に統一して示した．特に注意すべきことは転

置行列の取り扱い方である．本報ではこの点を明確にすることを目的とし説明してある．また，計算例を示したが，データ数が多量であるから 1 回だけ変換 · 逆変換して示してある．

9. 参考文献

[1] 松山　左和　他, ”ウェーブレット変換による動的カラー画像のハンドリング”, 可視化情報，Vol.19 Suppl No.1, 1999

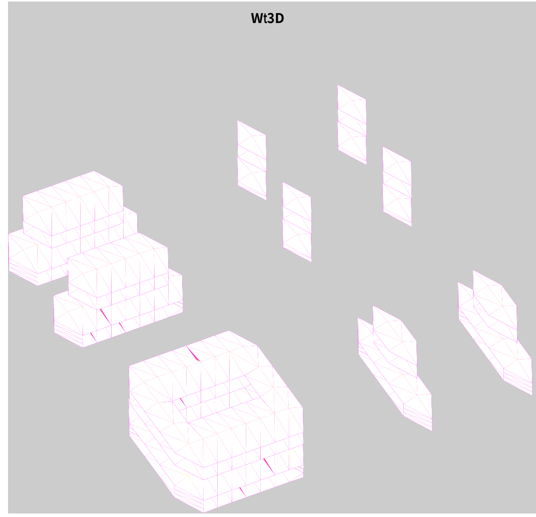

Fig. 1. 3D-graph of Daubechies' 1st order wavelet transform

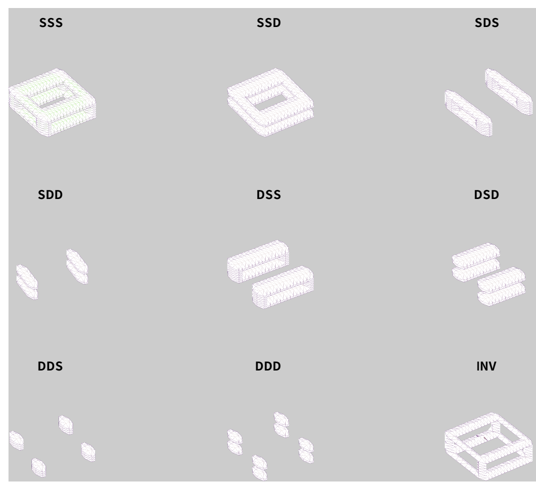

Fig. 2. 3D-graph of each section of Fig.1.

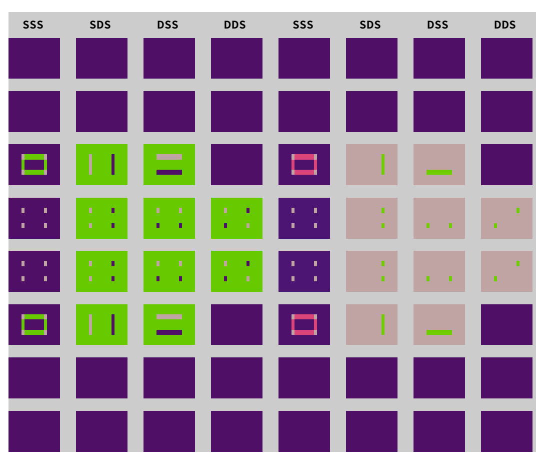

Fig. 3. 2D-graph of Fig.2.

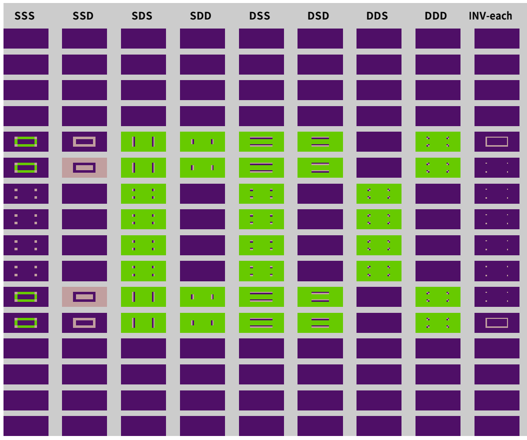

Fig. 4. 2D-graph of inverse wavelet transform of Fig.3.

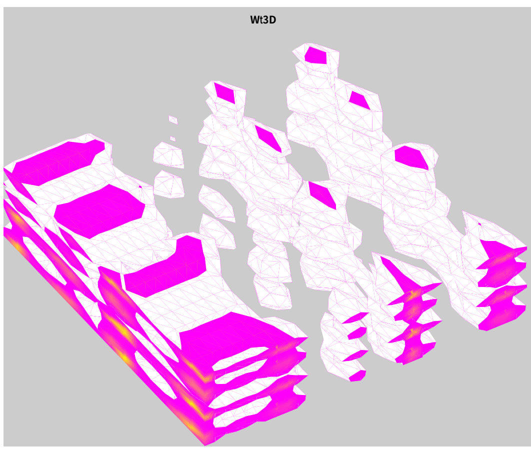

Fig. 5. 3D-graph of Coiflet 4th order wavelet transform

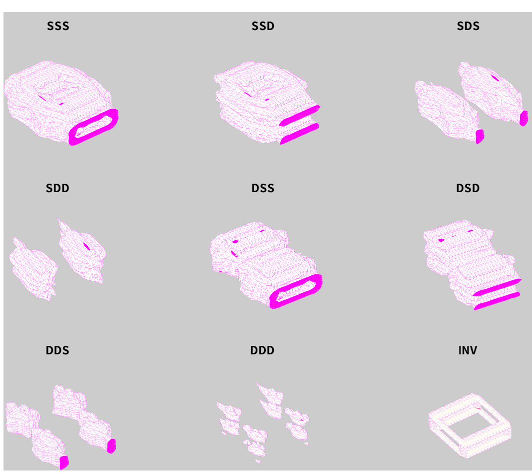

Fig. 6. 3D-graph of each section of Fig.5.

Fig. 7. 2D-graph of Fig.6.

Fig. 8. 2D-draw of inverse wavelet transform of Fig.7.

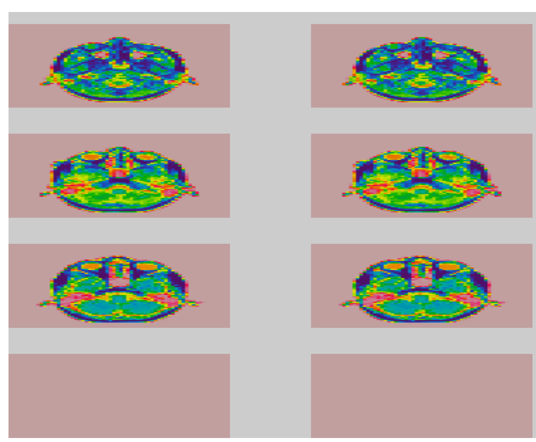

Fig. 9. SSS section of Wavelet transforms of brain sections

キーワード

信号処理，行列の転置，３次元ウェーブレット変換・逆変換

....

Summary

Formulations of 3D wavelet transforms

Yuko Oguchi

Computational Science Research Center, Hosei University

Wavelet transform has become one of the most useful methods of signal processings. Especillay, descrete orthogonal wavelets are more available than others. Until now, 1- and/or 2-dimensional transforms have become to be calculated quite easily as shown by the author among others. 3-dimensional transform, however, can not be understood so easily because of the difficulty of algorithm though it is described in some text books how to calculate them. In this report, the orthor has shown that 3D transform is derived reasonabley from 1D transform on the viewpoint of matrix multiplication and it's transpose. Some examples of 3D transforms and their inversions are shown in detail.

Keywords

Signal processing, Matrix transpose, 3D wavelet trnsform and it's inverse