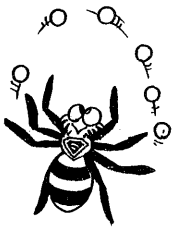


SPIDER開発を通して観たディジタル画像処理アルゴリズムの現状（1）
―― 直交変換とその画像処理への応用 ――

金子正秀　　尾上守夫
（東京大学　生産技術研究所）

[1] まえがき

離散的フーリエ変換（DFT），Walsh-Hadamard変換（WHT）、離散的COS変換（DCT）等の直交変換は、画像処理分野でも、特徴抽出、フィルタリング、画像圧縮等において、極めて重要な手法となっている。これらの直交変換技術の基本的性質及び従来の高速演算法については、Ahmed，Raoによる文献[1]を初めとして、多数の文献が発表されているので（例えば[2]～[5]）、ここでは、画像処理への応用を念頭に置いて、ソフトウェア面における最近の研究動向を述べることにする。

まず**2**で、直交変換の高速演算法における最近の研究の幾つかを概説する。次に**3**で、discrete linear base変換、Alpha変換等の新しい直交変換技術について紹介し、更に**4**で、変換技術における最近の話題を幾つか簡単に述べる。最後に**5**で、2次元直交変換の具体的計算手順と計算時間との関係についての実験結果を示し、先に提案した無転置演算法が、仮想記憶環境においても極めて有効であることを示す。

[2] 直交変換の高速演算法における最近の研究

直交変換の具体的応用には、高速演算法の存在が重要である。高速演算法としては、例えば、変換行列を疎行列（sparse matrix）の積に展開し、演算回数（特に乗算回数）を、直接計算に比して低減することに重点が置かれている。しかし、画像処理への応用を考える場合には、扱うデータ量が膨大であるので、単に演算回数だけでなく、必要なメモリ容量やデータ転送量、データの並べ換え等の種々の前・後処理、他についても、十分配慮する必要がある。

本節では、高速演算法における最近の研究として、WinogradによるDFTの高速演算法、任意サイズのDFT計算法、WHTに対する行列表現による統一的扱い、及び、slant変換、DCTの高速演算法について概説する。

2.1. Winogradによる高速フーリエ変換アルゴリズム（WFTA）[6],[7]

WFTAは複素数データのDFTに対する高速演算法であり、N点データの変換にO(N)の乗算回数で済むという大きな特徴を有している（Cooley-Tukey FFTアルゴリズムではO($N\log_2 N$)の乗算が必要）。しかし、必要なメモリ容量が、FFTの2.25Nに対し、最大7Nと増大するという問題がある。

以下、アルゴリズムの概要を簡単に紹介しておく。

(1) small-Nアルゴリズム

N点データ$\mathbf{x}$のDFT結果$\mathbf{X}$を行列表示すると、

$$\mathbf{X} = D_N \mathbf{x} \qquad D_N(m,k) = [W_N^{mk}] \qquad (1)$$
$$m,k \in [0,N-1],\ W_N = \exp(-j2\pi/N)$$

ここで、D_Nを式(2)の様にCanonical分解する。

$$S_N C_N T_N = D_N \qquad (2)$$

T_N, S_Nは各々J×N，N×J incidence行列であり、その要素は0，1，−1のみから成る。C_NはJ×J対角行列。

Winogradは$N=2,3,4,5,7,8,9,16$の8通りについて、式(2)に対し、J～Nである解を求めた。これは、乗算回数

が O(N) であることを意味し、small-N アルゴリズムと呼ばれている。なお、加算回数はFFTの場合と同程度である。

(2) Nが大きい場合

Nが互いに素なL個の因数の積

$$N = N_1 \times N_2 \times \cdots \times N_L \qquad (3)$$

に分解でき、各因数 N_ℓ が "small-N" 因数に対応するなら、式(1),(3)に対し、WFTAの基本式(4)が得られる。

$$\mathbf{X}' = (S_{N_L} \circledast \cdots \circledast S_{N_1}) \cdot (C_{N_L} \circledast \cdots \circledast C_{N_1}) \cdot (T_{N_L} \circledast \cdots \circledast T_{N_1})\, \mathbf{x}' \qquad (4)$$

但し、$\circledast$ はKronecker積を表わす。また、$\mathbf{x}'$, $\mathbf{X}'$ は各々、データの並べ換え R_I, R_F を施した入、出力ベクトル。

N_ℓ に対するsmall-Nアルゴリズムの乗算回数を M_{N_ℓ} とすると、全体としての乗算回数 M_N は、

$$M_N = M_{N_L} \cdot M_{N_{L-1}} \cdot \cdots \cdot M_{N_1} \qquad (5)$$

で表わされる。

計算の便宜上から、入・出力データをL次元行列に書き替えたものに対するnested formも提案されている。

WFTAにおける加算回数は、Nが式(3)の形に分解される時、対応するsmall-Nアルゴリズムの適用順序に依存する。これに対しては、まず9、次に5, 7, 16 の順となっており、残り2,3,4, 8の順序は任意で良い。

(3) WFTAに関するその他の話題

WFTAプログラムの作成には、具体的手法をまとめたZoharの文献[8]が参考になる。また、T.W. Parsonsは実数データ用のWFTAについて検討している。[9]

fixed-point演算の場合についての誤差解析では、WFTAはFFTに比して誤差が多く、FFTと同程度の誤差となるためには、データ表現のために1～2ビット余分に必要であることが示されている。[10] また、WFTAではFFTに比べ、メモリ・レジスタ間等のデータ転送、データの並べ換え等が全体としての計算時間に影響してき、演算回数のみによる比較が危険である旨の指摘もある。[11]

2.2. 任意サイズDFTの高速演算法

N点データ $x(k)$ のDFT $X(m)$

<順変換> $$X(m) = \sqrt{\frac{1}{N}} \sum_{k=0}^{N-1} x(k)\, W_N^{mk} \qquad (6)$$

<逆変換> $$x(k) = \sqrt{\frac{1}{N}} \sum_{m=0}^{N-1} X(m)\, W_N^{-mk} \qquad (7)$$

順変換の場合はchirp z変換の計算にならって次の様に行なえる。[12]

$$mk = \{m^2 + k^2 - (m-k)^2\}/2 \qquad (8)$$

と書き直せば、式(6)は、

$$X(m) = \sqrt{\frac{1}{N}}\, W_N^{\frac{m^2}{2}} \left[\sum_{k=0}^{N-1} \left\{ x(k) W_N^{\frac{k^2}{2}} \right\} \cdot W_N^{-\frac{(m-k)^2}{2}} \right] \qquad (9)$$

となり、$\hat{x}(k) = x(k) W_N^{\frac{k^2}{2}}$ と $h(k) = W_N^{-\frac{k^2}{2}}$ の畳込みとして計算できる。このためには、$\hat{x}(k)$ には0を追加し、$h(k)$ には、$h(k) = h(N'-k+1)$ $k = N'-N+1, \cdots, N'-1$ なるデータを付加してN'点（$N' \geq 2N-1$, 2の冪とする）系列とし、通常の基数2FFTを介して高速に計算できる。

一方、逆変換の場合については、新しく、相関を用いて計算する手法について述べる。

$$-mk = \{m^2 + k^2 - (m+k)^2\}/2 \qquad (10)$$

と書き直せば、式(7)は、

$$x(k) = \sqrt{\frac{1}{N}}\, W_N^{\frac{k^2}{2}} \left[\sum_{m=0}^{N-1} \left\{ X(m) W_N^{\frac{m^2}{2}} \right\} W_N^{-\frac{(m+k)^2}{2}} \right] \qquad (11)$$

となり、$\hat{X}(m) = X(m) W_N^{\frac{m^2}{2}}$ と $H(m) = W_N^{-\frac{m^2}{2}}$ の相関として計算できる。このためには、$\hat{X}(m)$ には0を追加し、$H(m)$ には、$H(m) = W_N^{-m^2/2}$ $m = N, \cdots, 2N-2$, 及び、$H(m) = 0$, $m = 2N-1, \cdots, N'-1$ を追加してN'点系列とし、基数2FFTを介して高速に計算できる。

2次元変換に対しても同様のアルゴリズムを用いることができる。N×N点変換を行なうのにN'×N'点変換を3回実行するため、計算時間やメモリ量は増大するが、任意の大きさの画像に適用できるという大きな利点がある。

2.3. WHTの高速演算法の行列表現による統一的扱い

WHTに関しては種々の定義がなされ、また多くの高速演算法が発表されてい

る。[1],[14] これに対し、FinoとAlgaziは、行列表現を用いて、WHTを統一的に扱う方法を示した。[13]

まず、order 2^nのHadamard orderのWalsh-Hadamard(WH)行列$[H_h(n)]$について考える。$[H_h(n)]$はorder 2のWH行列$[H_2]$を用いて次の様に書き表わせる。

$$[H_h(n)] = [H_2] \circledast [H_h(n-1)] = [H_2]^{\circledast n} \quad (12)$$

但し、$[H_2] = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

$\circledast n$はn回連続してKronecker積をとることを示す。

order 2^nのPaley及びWalsh orderのWH行列を各々$[H_p(n)]$, $[H_w(n)]$とすると、これらは次の様に$[H_h(n)]$から導くことができる。

$$[H_p(n)] = [H_h(n)][BR] = [BR][H_h(n)] \quad (13)$$

$$[H_w(n)] = [BI][BR][H_h(n)]$$
$$= [BI][H_h(n)][BR] \quad \text{etc.} \quad (14)$$

但し、$[BR]$, $[BI]$は各々、ビット逆順序並べ換え, ビット・インバージョン並べ換えを行なう行列である。

従って、$[H_h(n)]$による変換の前後で、適当なデータの並べ換えを施すことにより、他のorderingに対応したWHT結果を得ることができる。

2.4. slant変換の高速演算法[15],[16]

slant変換は離散的のこぎり波状の基底ベクトルを有し、画像ラインに沿った輝度変化を良く表現できる特徴がある。Pratt et al.はorder $N(=2^n)$のslant行列$[S_N]$をKronecker積を用いて、iterativeに構成する方法を示した。[15] しかし、(i)次数の異なる行列のKronecker積の分解を行なう必要がある、(ii)$[S_N]$はordered formではない、等の点で問題があり、特にNが大きい場合には計算が難しかった。これに対し、J.P.Crettezはordered formのslant変換の効率的計算法として2つのアルゴリズムを示している。[16] このうち、次に示すアルゴリズムは式(14)とのアナロジから得られており、実際に応用し易い。

$[S_N]$に適当な配置換えを施し、ordered formにしたslant行列を$[S_N^\circ]$とすると、これは式(15)の様に分解され、Hadamard-orderのWHTを基本にして効率的に計算できる。

$$[S_N^\circ] = \frac{1}{\sqrt{N}}[BI][\Delta_n][H_h(n)][BR] \quad (15)$$

但し、$[\Delta_n]$はslant係数に対応した行列であり、iterativeに構成される。

表1. 離散的COS変換(DCT)の高速演算法の比較 (1次元, N点データ)

項目 ＼ 提案者	N.Ahmed et al. [17] (1974)	W.H.Chen et al. [19] (1977)	M.J.Narasimha et al. [20] (1978)
アルゴリズムの概要	2N点FFTを使用	DCT行列のsin/cos行列と±1行列への分解	N点FFTを使用 (データの並べ換えが必要)
必要なデータ領域	4N (=2×2N)	N	2N
乗算回数(実数)	$4N(\log_2 2N+1)$	$N\log_2 N - \frac{3}{2}N + 4$ $(N \geq 4)$	$N\log_2 N - N + 2$
その他		・プログラムは複雑になる。 ・実数演算のみで可。	

注1. FFTにはradix-2アルゴリズムを用いるものとする。

注2. 2次元画像に適用する場合、FFTを用いるアルゴリズムについては、各行,列に対して1次元FFTを計算する。従って、データ領域としては、2次元実数配列の他に、1次元FFT用の作業領域を用意しておけば、計算を行なうことができる。

注3. 上記の他に、R.M.Haralickによる、N点FFTを2つ用いる手法[21]もあるが、Narasimha et al.の手法の方が優れていると言われている。[20]

計算はin-placeで行なえ、従来、利用し易い高速演算法がなかっただけに、画像処理への応用からも、本アルゴリズムは貴重である。

2.5. 離散的COS変換(DCT)の高速演算法

DCTはKarhunen-Loève変換を良く近似する変換であり[17][18]しかも高速演算法が存在するということで、信号処理・画像処理の分野で着目され、活発な研究がなされてきている。

DCTの高速演算法をまとめると表1.の様になる。Ahmed et al.の方法は計算時間、及び必要なメモリ量の増大という点で問題があったが、Chen, Narasimha et al.の方法により、改善が図られ、DCTの有用性が更に高まるものと思われる。

[3] 新しい直交変換手法

DCTとは別に、Karhunen-Loève変換を近似し、かつ高速演算法を有する手法として、discrete linear base変換やAlpha変換(AT)が発表されている。いずれも、比較的小さい(4x4, 8x8程度)画像の圧縮用に使われているが、他の変換との比較等に関しては、今後の検討が必要である。

また、WHTを基本にして、他の直交変換を導く手法についての研究も進められている。これは、理論的にも興味ある問題である。但し、実際の画像処理への応用に対しては、WHTからの変換用の行列の蓄積、計算時間等の点から、比較的小さい画像への適用に限られると考えられる。

3.1. discrete linear base変換(DLBT)[22]

DLBTの基底ベクトルは、次の条件を満たす様に生成される。

(i) N個のベクトル $V_1, \cdots, V_N \in R^N$ は互いに直交。

(ii) ベクトルの成分は整数値であり、共通因数を持たない。

(iii) 各ベクトルはevenまたはodd。

但し $V=(a_1, \cdots, a_N)$ が even, if $a_i = a_{N+1-i}$

odd, if $a_i = -a_{N+1-i}$

$(i=1, \cdots, N)$

生成されたベクトルはseguency順に並べ換えられる。N=4の場合の例を次に示す。

[例] r=1, S=-2 (文献[22]を参照)

$$\begin{pmatrix}1\\1\\1\\1\end{pmatrix} \quad \begin{pmatrix}3\\1\\-1\\-3\end{pmatrix} \quad \begin{pmatrix}1\\-1\\-1\\1\end{pmatrix} \quad \begin{pmatrix}1\\-3\\3\\-1\end{pmatrix}$$

なお、R^Nのベクトルが2つの低次元部分空間R^{N_1}, R^{N_2} $(N=N_1, N_2)$の基底ベクトルを用いて表現できるという原理に基づく、高速演算法が存在する。

3.2. Alpha変換(AT)[23]

Alpha変換は、信号のWHTを計算した後に、信号の統計的性質に適応して選ばれたパラメータαを含む行列を乗じて、KLTに似た結果を得ようとする手法である。

N=2の場合について説明する。入力データのWHT結果を(h_1, h_2)とすると、Alpha変換は式(16)で与えられる。

$$\begin{pmatrix}g_1\\g_2\end{pmatrix} = \begin{pmatrix}h_1\\h_2\end{pmatrix}\left(A_2\right), \quad A_2 = \frac{1}{\sqrt{1+\alpha^2}}\begin{bmatrix}1 & \alpha\\ -\alpha & 1\end{bmatrix} \qquad (16)$$

Alpha行列(ユニタリ)

パラメータαは、$F(\alpha) = \bar{g}_1^2/(\bar{g}_1^2+\bar{g}_2^2)$を最大とするように選ばれる。なお、order 2のAlpha変換はKLTと厳密に一致する。

3.3. WHTを基本にして、他の直交変換を求める手法

WHTを介してDFTを計算する手法が以前から検討されている。[24]

また、最近、HeinとAhmedは、WH行列$[H_w(n)]$の性質を利用して、入力データのDCTが、Walsh-order WHTの結果に、次の変換行列$[T(n)]$を乗ずることにより求められることを示している。[25]

$$[T(n)] = [\hat{\Lambda}(n)][\hat{H}_w(n)]^t \qquad (17)$$

但し、$[\hat{\Lambda}(n)], [\hat{H}_w(n)]$は、各々、ビット逆順序に並べ換えられたDCT行列, WH行列である。^{t}は転置を表わす。

なお、$[T(n)]$はorthonormalであり、block diagonal構造を有する。

更に、JonesとHeinは、Fourier, slant, DCT, WHTを含めて、一般に用いられている直交変換がEven/Odd Transform[*1,*2]であり、Even/Odd Transform同志

では、疎行列を乗ずることによって、相互変換できることを指摘している。[26]

(注) *1. Even/Odd Transform では、変換ベクトルの半数が even ベクトルであり、残り半分が odd ベクトルである。

*2. Even/Odd ベクトル係数が無相関となるような相関行列を有するデータ・クラスに対しては、KLT も Even/Odd Transform となる。

[4] 変換技術における最近の話題

整数環上で、ある整数(Fermat数など)を法とした整数演算を行ない、有限離散的畳込みを高速かつ誤差なく計算する手法として、数論的変換(Number Theoretic Transform)が提案され研究が進められている。[27][28] 語長とデータ長の依存性、あふれ等の点での問題もあり、画像処理分野での利用に関しては報告が少ない。

2次元離散的 Hilbert 変換の研究[29]や、WHT と Haar 変換を組合せ、両者の特徴を持たせた、Hadamard-Haar 変換の研究[30]等も進められている。また、N×N 画像マトリクスを、高々N個の非零要素を有する対角行列に変換する singular value decomposition(SVD)についても研究が進められ、[32],[33]画像圧縮や復元などへの応用で成果を挙げているが、固有値計算を伴うため、大きな画像への適用は難しい。

直交変換の画像解析への応用に関する話題としては、S.T. Tanimoto による、N×N点画像に対し、1画素ずつ中心をずらしながら、M×M点窓内で局所的な2次元DFTを順次計算するための効率的手法が挙げられる。[31] 各窓ごとに、単純に2次元DFTを行なうのでは、$O(N^2M^2\log_2 M)$の計算が必要である。これに対し、窓の端部分での行方向または列方向の1次元DFT結果に着目し、計算手順を工夫することにより、$O(M^2N^2)$の計算量で済ませることができ、texture の記述などへの応用に有用と思われる。

[5] 計算手順による、2次元直交変換に要する計算時間の差異

2次元直交変換の場合、変換のための基本的な乗算、加算回数が同一であっても、行方向及び列方向に対する計算手順によって、全体としての計算時間には差異が生じ、特に画像サイズが大きい場合には、これによる影響が目立ってくる。

ここでは、4通りの計算手順を取り上げ、FFT と Hadamard-order WHT に対して計算時間の測定をし、比較を行なった結果を述べる。

5.1. 2次元直交変換の計算手順

基本的計算手順として、図1.に示す4通りを取り上げ、実験を行なった。

5.2. 計算時間測定用プログラムの構成

(1) Hadamard-order の WHT

2次元ランダム・データを用いて、タイプ1～4の手順について、butterfly 演算(実数加減算のみ)に要する時間の測定を行なった。

(2) 高速フーリエ変換(FFT)

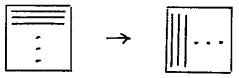
基数2の、Sande-Tukey FFT アルゴリズムを用いた。変換用原画としては、実部に乱数を、虚部に0を入れたものを使用した。また、W_Nの冪、従って、sin, cos の計算、及びビット逆順序については、前もってテーブルを作成しておき、これを参照しながら処理するものとした。計算時間としては、butterfly 演算とビット逆順序並べ換えに要した時間を測定した。

5.3. 実験

使用計算機は FACOM M-160AD(多重仮想記憶方式、OS IV/F4)であり、使用言語は FORTRAN, 使用コンパイラは HE(最適化機能を有する)である。

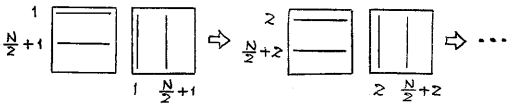
測定結果を表2.に示す。

［タイプ1］行方向演算 → 列方向演算



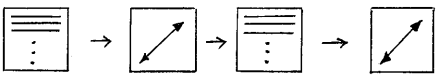
・基本的考え方
・1次元直交変換を単に、行方向、列方向に順に適用

［タイプ2］行方向と列方向、交互にbutterfly演算を進める。



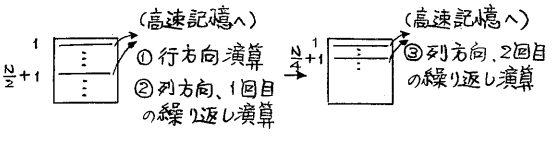
・画像配列、正方形
・どの画素とどの画素でbutterfly演算を行なうかを決めるための計算量（インデックス計算量と名付ける）が少なくて済む。

［タイプ3］
行方向演算 → 転置 → 行方向演算 → 転置



・転置処理が必要
正方形配列でないと、転置は面倒。
正方形配列であれば、高速転置アルゴリズムを使うこともできる。
（但し、ここでは単純な転置を採用）

［タイプ4］無転置演算法[34]



→ ・・・

・高速記憶の容量が少なくても可。

図1. 2次元直交変換の計算手順

表2. 計算手順による、2次元直交変換に要する計算時間の差異 （単位、秒）

画像サイズ ＼ 直交変換 ＼ タイプ ＼ 時間		radix 2 FFT (butterfly演算とビット逆順序) 1	2	3	4	Hadamard-order WHT (butterfly演算のみ) 1	2	3	4
128×128	a.	5.84	4.57	5.85	5.07	1.64	1.04	1.60	1.27
	r.	5.79	4.45	5.81	5.05	1.63	1.04	1.59	1.27
256×256	a.	27.5	20.5	26.3	23.1	7.28	4.73	7.14	5.81
	r.	27.2	20.3	25.9	22.8	7.10	4.71	6.94	5.64
512×512	a.	—	—	—	—	*1	*2	1082	76.3
	r.	—	—	—	—			32.6	25.7

［補足］
*1, *2 は途中打切り
（*1. a. 50分に対し r. 24秒
*2. a. 15分に対し r. 5秒）

［注］
注1. a. disc I/O等を含めた処理に要する全時間。
r. CPU time
注2. テストプログラムのみがactiveとなるようにして使用。

5.4. 検討

表2.の結果より、次のことが言える。

M-160ADの場合、データ領域が500kB（1語＝4バイト）程度までは、タイプ2が計算時間の点で有利である。これは、インデックス計算量が少ないためと考えられる。

但し、無転置演算法（タイプ4）を用いても、タイプ2より、1〜2割余分に計算時間がかかる程度である。これに対し、データ領域が1MB程度と大きくなった場合には、タイプ4が、ページング回数を比較的低く押さえることができ、極めて良い結果

を与える。これに対し、他の手順ではページング操作の回数、従って、時間が著しく増大し、計算機の使用効率の大幅な低下が生じてしまうのである。この様な傾向は、仮想記憶方式を用いた計算機に共通に言えることである(但し、ある程度、使用計算機に依存する)。

無転置演算法[34]は、高速記憶の容量が限られている場合に、外部記憶とのデータ転送を工夫して、効率的に2次元変換を計算しようとする手法であるが、仮想記憶方式の計算機においても、計算機の使用効率を挙げるために有効な手段となることが明らかになった。

[6] むすび

直交変換における最近の話題について、画像処理への応用を念頭に置いて、高速演算法や新しい手法を中心に紹介した。紙数が限られているため、詳細は略さざるを得なかったので、詳しくは文献を参照されたい。また、本稿で取り上げたもの以外にも幾つか(特にFFTに関連して)報告があることを付け加えておく。

直交変換に関しては、高速演算法の開発、新しい変換技術の開発、ともに、活発に研究が進められている。本稿では触れなかったが、ハードウェア面での進歩にも著しいものがある。具体的応用も含めて、この分野における今後の研究成果が期待される。

<付録> SPIDERで作成した2次元直交変換サブルーチンの計算時間 (単位.秒)

サブルーチン名 \ 画像サイズ	64×64	128×128	256×256
FFTS2(基数2 FFT)	1.24	5.54	24.8
FFTA2(任意サイズ FFT)	18.5	81.6	—
WHT2 (WHT: Hadamard)	0.44	1.96	8.64
WHT2 (WHT: walsh)	0.50	2.16	9.39
HAR2 (Haar変換)	0.20	0.76	3.01
SLT2 (slant変換)	0.49	2.10	9.34
DCB2(DCT: butterfly型)	2.03	8.68	36.9
DCF2(DCT: 2倍長FFT)	3.15	13.6	—

注1. 使用計算機、FACOM, M-160AD, FORTRAN, HEコンパイラ使用

注2. サブルーチンをcallした時点から、メインプログラムにもどってくるまでの時間。従って、直交変換自体以外の前・後処理等も含めた全実行時間である。

注3. 各サブルーチンのアルゴリズム等詳細は、SPIDERマニュアルを参照のこと。

[参考文献]

■ 直交変換、全般

(1) N.Ahmed and K.R.Rao : "Orthogonal Transforms for Digital Signal Processing," p.263, Springer-Verlag, 1975.
(2) E.O.Brigham : "The Fast Fourier Transform," p.252, Prentice-Hall, 1974. (宮川, 今井 訳 : "高速フーリエ変換" p.262, 科学技術出版社, 1978.)
(3) 尾上守夫 : "直交変換技術 ── 高速フーリエ変換を中心に ──", テレビジョン, Vol.31, pp.712-721, 1979.9.
(4) 宮川、他 : "ディジタル信号処理," p.270, 電子通信学会, 1975.
(5) 直交変換関係の重要な論文はIEEE Selected Paper Seriesにも良く載っているので参照されたい.

■ Winogradのフーリエ変換アルゴリズム(WFTA)

(6) H.F.Silverman : "An introduction to programming the Winograd's Fourier transform algorithm (WFTA)," IEEE Trans. Vol.ASSP-25, pp.152-165, 1977.4.
(7) 飯塚、田中 : "フーリエ変換を高速に実行する最近の技術," 日経エレクトロニクス, pp.60-91, 1978.1.9.
(8) S.Zohar : "A prescription of Winograd's discrete Fourier transform algorithm," IEEE Trans., Vol.ASSP-27, pp.409-421, 1979.8.
(9) T.W.Parsons : "A Winograd-Fourier transform algorithm for real-valued data," ibid., pp.398-402.
(10) R.W.Patterson and J.H.McClellan : "Fixed-point error analysis of Winograd Fourier transform algorithms," IEEE Trans., Vol.ASSP-26, pp.447-455, 1978.10.
(11) L.R.Morris : "A comparative study of time efficient FFT and WFTA programming for general purpose computers," ibid., pp.141-150, 1978.4.

■ 任意サイズDFTの計算

(12) L.R.Rabiner, R.W.Schafer C.M.Rader : "The chirp z-transform algorithm and its application," Bell Syst. Tech. J., Vol.48, pp.1249-1292, 1969.5-6.

■ Walsh-Hadamard 変換

(13) B.J.Fino and V.R.Algazi : "Unified matrix treatment of the fast Walsh-Hadamard transform," IEEE Trans., Vol.C-25, pp.1142-1146, 1976.11.
(14) J.W.Manz : "A sequency-ordered fast Walsh transform," IEEE Trans., Vol.AU-20, pp.204-205, 1972.8.

■ Slant変換

(15) W.K.Pratt, W.H.Chen and L.R.Welch : "Slant transform image coding," IEEE Trans., Vol.COM-22, pp.1075-1093, 1974.8.
(16) J.P.Crettez :"The relationship between two fast slant transforms in ordered form," Proc. of the 4th Int. Joint Conf. on Pattern Recog., pp.445-448, 1978.

■ 離散的COS変換（DCT）

(17) N.Ahmed, T.Natarajan and K.R.Rao : "Discrete cosine transform," IEEE Trans., Vol.C-23, pp.90-93, 1974.1.
(18) K.S.Shanmugam : "Comments on 'Discrete cosine transform' ," IEEE Trans., Vol.C-24, p.759, 1975.7.
(19) W.H.Chen, C.H.Smith and S.C.Fralick : "A fast computational algorithm for the discrete cosine transform," IEEE Trans, Vol.COM-25, pp.1004-1009, 1977.9.
(20) M.J.Narasimha and A.M.Peterson : "On the computation of the discrete cosine transform," IEEE Trans., Vol.COM-26, pp.934-936, 1978.6.
(21) R.M.Haralick : "A storage efficient way to implement the discrete cosine transform," IEEE Trans., Vol.C-25, pp.764-765, 1976.7.

■ 新しい直交変換手法

(22) R.M.Haralick and K.Shanmugam : "Comparative study of a discrete linear basis for image data compression," IEEE Trans, Vol.SMC-4, pp.16-27, 1974.1.
(23) G.E.Lowitz : "The Alpha transform : a Walsh-Hadamard generalization which adapts to image statistics," Proc. of the 4th Int. Joint Conf. on Pattern Recog., pp.441-444, 1978.11.
(24) R.Kitai and K.H.Siemens : "Discrete Fourier transform via Walsh transform," IEEE Trans., Vol.ASSP-27, p.288, 1979.6.
（この分野の研究動向及び文献リストが載っている。）
(25) D.Hein and N.Ahmed : "On a real-time Walsh-Hadamard / cosine transform image processor," IEEE Trans., Vol.EMC-20, pp.453-457, 1978.8.
(26) H.W.Jones and D.N.Hein : "The Karhunen-Loeve, discrete cosine, and related transforms obtained via the Hadamard transform," International Telemetering Conference / USA, pp.87-98, 1978.

■ 変換技術における最近の話題、他

(27) R.C.Agarwal and C.S.Burrus : "Number theoretic transforms to implement fast digital convolution," Proc.IEEE, Vol.63, pp.550-560, 1975.4.
(28) R.C.Agarwal and C.S.Burrus : "Fast convolution using Fermat number transforms with applications to digital filtering," IEEE Trans., Vol.ASSP-22, pp.87-97, 1974.4.
(29) N.K.Bose and K.A.Prebhu : "Two-dimensional discrete Hilbert transform and computational complexity aspects in its implementation," IEEE Trans. Vol. ASSP-27, pp.356-360, 1979.8.
(30) K.R.Rao, M.A.Narasimaha and K.Revuluri : "Image data processing by Hadamard-Haar transform," IEEE Trans., Vol.C-24, pp.888-896, 1975.9.
(31) S.T.Tanimoto : "An optimal algorithm for computing Fourier texture descriptors," IEEE Trans., Vol.C-27, pp.81-84, 1978.1.
(32) H.C.Andrews and C.L.Patterson : "Singular value decomposition (SVD) image coding," IEEE Trans., Vol.COM-24, pp.425-432, 1976.4.
(33) H.C.Andrews and C.L.Patterson : "Outer product expansions and their uses in digital image processing," IEEE Trans. Vol.C-25, pp.140-148, 1976.2.
(34) 尾上守夫："大規模画像データの無転置2次元変換法," テレビジョン, Vol.30, pp.672-677, 1976.8.