NEW FAST TRIGONOMETRIC TRANSFORMS

[1] K. Egiazarian, [2] V. G. Labunets, [2] E. V. Labunets, and [1] J. Astola
[1] Tampere University of Technology
Signal Processing Laboratory, Tampere, Finland
e-mails: karen@cs.tut.fi, jta@cs.tut.fi
[2] Ural State Technical University
Department of Automation and Information Technologies
Ekaterinburg, Russia
e-mail: lab@ait.rcupi.e-burg.su

ABSTRACT

New fast trigonometric transforms formed as combinations of discrete parametric Pontryagin transforms and discrete conjugate parametric Pontryagin transforms are introduced in this paper. They include as special cases such known orthogonal trigonometric transforms as the Fourier, Hartley, Wang, cosine and sine, Walsh, Chrestenson transforms, and many others. Different methods of efficient computation of the introduced transforms are described.

1 Introduction

Discrete trigonometric (complex- and real-valued) transforms such as the discrete Fourier, cosine, sine, Wang, Chrestenson transforms and others, are widely used in many applications of digital signal processing [13]. Some of them are building blocks for modern unitary transforms [11, 16].

Much work has been done in generalisation of discrete trigonometric transforms, unification of the existing trigonometric transforms and synthesis of new such transforms possessing similar fast implementation. Among them we note the development of a sinusoidal family of unitary transforms by eigenvectors of tridiagonal matrices [9, 18], shifted discrete Fourier transforms [17], shifted trigonometric transforms [5, 6], lapped transforms and extensions [11], local trigonometric transforms [16], and others.

In this paper we introduce a general class of parametric trigonometric transforms, the combination of discrete parametric Pontryagin transforms (CDPPT).

In Section 2.1 we introduce a new class of transforms, the combination of shifted discrete Fourier transforms (CSDFT), and show that they unify a large class of discrete unitary trigonometric "Fourier-like" transforms.

In Section 2.2 we first generalize the family of Pontryagin transforms (transform matrix is formed as Kronecker products of the discrete Fourier transform matrices [14]) to parametric Pontryagin transforms. Then we introduce a new class of trigonometric transforms formed as a combination of discrete parametric Pontryagin transforms (CDPPT).

In Section 3 three methods for efficient implementation of introduced transforms are described, based on FFT-type algorithms, Good's technique, and Jacobi-Givens planar rotations for DCT-IV (or DST-IV) type transform matrices.

2 Combination of trigonometric transforms

2.1 Combination of Shifted Discrete Fourier Transforms (CSDFT)

Let $\{x(n),\ n=0,1,...,N-1\}$ be a real or complex data sequence of length N. The combination of shifted discrete Fourier transforms (CSDFT) $\{X(k),\ k=0,1,...,N-1\}$ is defined as

$$X(k)=\sum_{m=0}^{N-1}[\alpha_1\gamma_M^{a(k+b)(m+c)}+\alpha_2\gamma_M^{-a(k+b)(m+c)}]x(m), \quad (1)$$

$k=0,1,...,N-1$, or in the matrix form:

$$\mathbf{X}=(\alpha_1 F_N+\alpha_2\overline{F}_N)\mathbf{x}, \quad (2)$$

where $\gamma_M=exp(-i\frac{2\pi}{M})$ is a primitive Mth root of unity,

$$F_N=F_N(M,a,b,c)=[\gamma_M^{a(k+b)(m+c)}], m,n=0,1,...,N-1, \quad (3)$$

is the shifted (parametric) discrete Fourier transform matrix, $\overline{F}_N=\overline{F}_N(M,a,b,c)$ is the conjugate to F_N, a,b,c are real numbers, $\alpha_1=\alpha_1(m,k)$ and $\alpha_2=\alpha_2(m,k)$ are complex-valued functions of two variables.

In general, the CSDFTs are multiparametric nonunitary transforms. Nevertheless, for some sets of parameters CSDFTs are unitary transforms. Among them are the following known trigonometric transforms [6, 11, 15]:

1) Discrete Fourier Transform (DFT):

$$M=N; a=1, b=c=0; \alpha_1=1, \alpha_2=0.$$

2) Discrete Hartley Transform (DHT):

$$M=N; a=1, b=c=0; \alpha_1=\frac{1+i}{2}, \alpha_2=\frac{1-i}{2}.$$

3) Discrete Wang Transforms (DWT):

$$M=N; a=1, b,c\in\{0,0.5,1\}; \alpha_1=\frac{1+i}{2}, \alpha_2=\frac{1-i}{2}.$$

4) Discrete cosine Transforms (DCTs):

$$I: \quad M=N-1; a=0.5, b=c=0; \alpha_1=\alpha_2=\frac{\beta(m)\beta(k)}{2}.$$

$$II: \quad M=N; a=0.5, b=0; c=0.5; \alpha_1=\alpha_2=\frac{\beta(k)}{2}.$$

$$III: \quad M=N; a=0.5, b=0.5; c=0; \alpha_1=\alpha_2=\frac{\beta(m)}{2}.$$

$$IV: \quad M=N; a=0.5, b=c=0.5; \alpha_1=\alpha_2=0.5.$$

5) Discrete sine Transforms (DSTs):

$$I: \quad M=N+1; a=0.5, b=c=1; \alpha_1=\alpha_2=\frac{i}{2}.$$

$$II: \quad M=N; a=0.5, b=1; c=0.5; \alpha_1=\alpha_2=i\frac{\beta(k)}{2}.$$

$$III: \quad M=N; a=0.5, b=0.5; c=1; \alpha_1=\alpha_2=i\frac{\beta(m)}{2}.$$

$$IV: \quad M=N; a=0.5, b=c=0.5; \alpha_1=\alpha_2=\frac{i}{2}.$$

Here $\beta(k)$ is defined by

$$\beta(k)=\begin{cases}\sqrt{(2/N)}, & k\neq 0\\ \sqrt{(1/N)}, & k=0.\end{cases}$$

Some other particular cases of CSDFTs can be found in [6].

2.2 Combination of Discrete parametric Pontryagin Transforms (CDPPT)

Let F_p be the matrix of the discrete Fourier transform of order p.

The discrete Pontryagin-(Vilenkin-Chrestenson) transform (DPT) matrix P_N is defined as the Kronecker product of the Fourier matrices [14]:

$$P_N=\otimes_{j=1}^{n}F_{p_j}, \tag{4}$$

where $N=p_1\cdot\ldots\cdot p_n$ is some decomposition of N. The DPT contains as the particular cases such known transforms, as the Walsh transform (when $p_j=2$, $j=1,2,\ldots,n$) and the Chrestenson-(Vilenkin) transform [3] (when $p_j=p$, $j=1,2,\ldots,n$).

First we will extend the DPTs to Discrete parametric Pontryagin-(Vilenkin-Chrestenson) Transforms (DPPT) simply by replacing the kernel $F_p=F_p(p,1,0,0)$ in the definition of DPT to arbitrary shifted Fourier transforms $F_p=F_p(q,a,b,c)$, where $F_p(q,a,b,c)$ is defined by (3).

Now, let us substitute P_N instead of F_N in (2) to obtain the combination of discrete parametric Pontryagin-(Vilenkin-Chrestenson) transforms (CDPPTs):

$$\mathbf{X}=(\alpha_1P_N+\alpha_2\overline{P}_N)\mathbf{x}. \tag{5}$$

Let us consider the following example. Let $N=9$ and $P_N=F_3(3,0.5,0.5,0.5)\otimes F_3(3,1,0,0)$, i.e. the Kronecker product of the DCT-IV and the DFT matrices of order 3. As the result we will get the following orthogonal matrix

$$\frac{(P_9+\overline{P}_9)}{2}=\begin{pmatrix}
0.455 & 0.455 & 0.455 & 0.333 & 0.333 & 0.333 & 0.122 & 0.122 & 0.122\\
0.455 & -0.333 & -0.122 & 0.333 & -0.455 & 0.122 & 0.122 & -0.455 & 0.333\\
0.455 & -0.122 & -0.333 & 0.333 & 0.122 & -0.455 & 0.122 & 0.333 & -0.455\\
0.333 & 0.333 & 0.333 & -0.333 & -0.333 & -0.333 & -0.333 & -0.333 & -0.333\\
0.333 & -0.455 & 0.122 & -0.333 & -0.122 & 0.455 & -0.333 & 0.455 & -0.122\\
0.333 & 0.122 & -0.455 & -0.333 & 0.455 & -0.122 & -0.333 & -0.122 & 0.455\\
0.122 & 0.122 & 0.122 & -0.333 & -0.333 & -0.333 & 0.455 & 0.455 & 0.455\\
0.122 & -0.455 & 0.333 & -0.333 & 0.455 & -0.122 & 0.455 & -0.333 & -0.122\\
0.122 & 0.333 & -0.455 & -0.333 & -0.122 & 0.455 & 0.455 & -0.122 & -0.333
\end{pmatrix}.$$

Note that this matrix P_9 contains only $2\cdot 3=6$ different elements, whereas the DCT-IV matrix $F_9(9,0.5,0.5,0.5)$ of order 9 contains $2\cdot 9=18$ different elements, and the Kronecker product $F_3(3,0.5,0.5,0.5)\otimes F_3(3,0.5,0.5,0.5)$ of two DCT-IV matrices of orders 3 contains $2\cdot 6=12$ different elements. Moreover, the coding gains computed for the Markov I order process with the correlation parameter $\rho=0.95$ [11] give the following results: 5.1448, 3.5113 and 2.4178, respectively, for P_9 (from this example), DCT-IV of order 9, and the Kronecker product of two DCT-IV of order 3.

3 Fast Algorithms

There are two most popular methods of efficient computation of trigonometric transforms (based on the factorization of transform matrix) via fast Fourier transform (FFT), via fast algorithm using Good's [7] factorization technique and via fast DCT-IV (or DST-IV) [11, 13, 15].

3.1 Fast algorithms via FFT

He we describe fast CSDFT transform via FFT algorithms. This will give more flexibilities to use several efficient methods for computing DFT (see, e.g. [12]).

Let $M=N=r_1\cdot r_2\cdot\ldots\cdot r_n$, $\gamma=\gamma_M$. The CSDFT $\mathbf{X}=V\cdot\mathbf{x}$ of the vector $\mathbf{x}$ will be represented as the sum of two transforms, the shifted discrete Fourier transform and the shifted inverse discrete Fourier transform:

$$\mathbf{X}=\mathbf{X_1}+\mathbf{X_2}=V_1\cdot\mathbf{x}+V_2\cdot\mathbf{x}, \tag{6}$$

where

$$V_j=[\alpha_j\gamma^{(-1)^{j+1}a(k+b)(m+c)}],\ m,k=0,\ldots,N-1,\ j=1,2. \tag{7}$$

Each of the matrices V_j can be factorized by

$$V_j=\alpha_j\gamma^{abc}L_j\cdot\Phi_j\cdot R_j,\ j=1,2, \tag{8}$$

where

$$L_j=diag(1,\gamma^{(-1)^{j+1}ab},\ldots,\gamma^{(-1)^{j+1}(N-1)ab}), \tag{9}$$

$$R_j=diag(1,\gamma^{(-1)^{j+1}ac},\ldots,\gamma^{(-1)^{j+1}(N-1)ac}), \tag{10}$$

and Φ_j is the generalized Fourier matrix of order N, $\Phi_j=[\gamma^{(-1)^{j+1}amk}]$, $m,k=0,1,\ldots,N$.

Further factorizations of V_j matrices can be done by the following decomposition of the matrices Φ_j:

$$\Phi_j=S^TW_j^{(1)}D_j^{(1)}W_j^{(2)}\ldots D_j^{(n-1)}W_j^{(n)}, \tag{11}$$

where

$$D_j^{(k)}=I_{r_{i+1,n+1}}\otimes D_{r_{0,k-1}r_k}, \tag{12}$$

$$W_j^{(k)}=I_{r_{i+1,n+1}}\otimes[\Phi_j]_{r_j}\otimes I_{r_{0,k-1}}, k=1,\ldots,n. \tag{13}$$

$r_{k+1,n+1}=r_{k+1}r_{k+2}\ldots r_{n+1}$, $r_0=r_{n+1}=1$, I_m is the identity matrix of order m, $[\Phi_j]_{r_j}$ is the generalized Fourier matrix of order r_j, $j=1,2$, S^T is a transposed matrix to the generalized bit-reversal matrix [12].

Let us show an example of the fast algorithm for the CSDFT with parameters $N=M=6, a=0.5, b=0, c=$

$0.5, \alpha_1 = \alpha_2 = \frac{\beta(k)}{2}$ (which is corresponding to the case of DCT-II). By the formulas (2)-(9) we have

$$V = \frac{\beta}{2} L_1 \Phi_1 R_1 + L_2 \Phi_2 R_2.$$

Substituting parameters to the formulas we obtain, $L_1 = L_2 = I$,

$$R_1 = diag(1, \gamma^{1/4}, \gamma^{1/2}, \gamma^{3/4}, \gamma, \gamma^{5/4}),\ \ R_2 = R_1^*,$$

$$\Phi_1 = S^T([I]_3 \otimes [\Phi_j]_2) diag([I]_2, (1,\gamma), (1,\gamma^2))([\Phi_j]_3 \otimes [I]_2),$$

$\Phi_2 = \Phi_1^*$, where $[I]_k$ is the identity matrix of order k,

$$[\Phi_1]_2 = \begin{pmatrix} 1 & 1 \\ 1 & i \end{pmatrix},\ \ [\Phi_2]_2 = ([\Phi_1]_2)^*,$$

$$[\Phi_j]_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \gamma & \gamma^2 \\ 1 & \gamma^2 & \gamma \end{pmatrix}.$$

For factorization of the matrices Φ_j (and, therefore, the matrices V_1 and V_2) one can use an algorithm of Winograd [12].

3.2 Fast algorithms via Good's technique

The Good's theorem plays an important role in efficient computation of many orthogonal transforms. Let a square matrix $\mathbf{H}$ of order $N = N_1 N_2 ... N_k$ be represented as the Kronecker product of k matrices $\mathbf{V}^{(j)}$ of order N_j, $j = 1, 2, ..., k$, i.e. $\mathbf{H} = \mathbf{V}^{(1)} \otimes \mathbf{V}^{(2)} \otimes ... \otimes \mathbf{V}^{(k)}$. Then, by Good's theorem there exists a way of representing $\mathbf{H}$ as a product of k sparse matrices:

$$\mathbf{H} = \prod_{j=1}^{k} \mathbf{H}^{(j)},$$

where

$$\mathbf{H}^{(j)} = \mathbf{I}_{(M(j))} \otimes \mathbf{V}^{(j)} \otimes \mathbf{I}_{(L(j))},\ j = 1, 2, ..., k,$$

$\mathbf{I}_{(n)}$ is the identity matrix of order n, $M(j) = N_{j+1} N_{j+2} \cdots N_k$ and $L(j) = N_1 N_2 \cdots N_{j-1}$.

Note, that the planar mapping of the kernel of n-dimensional separable Fourier transform have the form of the Chrestenson matrix, and, therefore, can be computed efficiently by applying Good's theorem as well as the decomposition of the Fourier transform kernel. This idea was used in the derivation of fast multidimensional discrete Fourier transform algorithms [2].

We can use the Good's technique with the FFT-type algorithms (introduced in the previous section) for fast implementation of CDPPTs.

3.3 Fast algorithms via fast DST-IV or DCT-IV

Most of the discrete trigonometric transforms can be decomposed to DCT-IV or DST-IV or similar transforms [11, 13, 15]. Therefore, we need to have fast algorithms for one of this transformations.

Here we will present a decomposition of the DST-IV matrix $S_N^{IV} = F_N(N, \frac{i}{2}, 0.5, 0.5)$ using the Jacobi-Givens rotations.

Let $\mathrm{rot}\Big(\varphi_m \mid k_m, l_m\Big) := \begin{bmatrix} \cos\varphi_m & \sin\varphi_m \\ -\sin\varphi_m & \cos\varphi_m \end{bmatrix}_{k_m, l_m}$ be the elementary Jacobi-Givens rotation in 2D coordinate plane (e_{k_m}, e_{k_l}) of the signal space $\mathbf{V}(e_0, e_1, ..., e_{N-1})$. We use sequential method of decomposition of S_N^{IY} using finite sequence of elementary Jacobi-Givens rotations:

$$[S_N^{IY}]_m = \mathrm{rot}\Big(\varphi_m \mid k_m, l_m\Big) \cdot [S_N^{IY}]_{m-1},$$

where $m = 0, 1, ..., t$, $[S_N^{IY}]_0 := S_N^{IY}$, $[S_N^{IY}]_t = I_N$. The angles φ_m are determined so that $S_{k_m, l_m} = S_{l_m, k_m} = 0$, where $[S_N^{IY}]_m; = [S_{k_m, l_m}]$.

We start with S_2^{IY}. Obviously,

$$S_2^{IY} = \begin{bmatrix} s_{1,1} & s_{3,1} \\ s_{3,1} & -s_{3,1} \end{bmatrix},$$

where $s_{i,j} := \sin(i\pi/2^{j+2})$. For the decomposition to S_2^{IY} we need to make one rotation $\mathrm{rot}(\varphi_1|1,2)$, where $\varphi_1 = \mathrm{arctg}(s_{1,1}/s_{3,1}) = -\pi/8$.

Let us move to 4×4-transform matrix

$$S_4^{IY} := \begin{bmatrix} s_{1,2} & s_{3,2} & s_{5,2} & s_{7,2} \\ s_{3,2} & s_{7,2} & s_{1,2} & -s_{5,2} \\ s_{5,2} & s_{1,2} & -s_{7,2} & s_{3,2} \\ s_{7,2} & -s_{5,2} & s_{3,2} & -s_{1,2} \end{bmatrix}.$$

Taking pair of rotations $\mathrm{ROT}(\varphi_1, \varphi_2) := \mathrm{rot}(\varphi_1|1,4)\mathrm{rot}(\varphi_2|2,3)$ as the two first iterations of decomposition and defining angles by the first basis vector: $\varphi_1 = -\mathrm{arctg}(s_{1,2}/s_{7,2}) = -\pi/16$, $\varphi_2 = -\mathrm{arctg}(s_{3,2}/s_{5,2}) = -3\pi/16$, we obtain

$$[S_4^{IY}]_2 = \mathrm{rot}(\varphi_1|1,4)\mathrm{rot}(\varphi_2|2,3)[S_4^{IY}] =$$

$$= \frac{\sqrt{2}}{2} \begin{bmatrix} & & 1 & 1 \\ d & d & d & -d \\ d & d & -d & d \\ 1 & -1 & & \end{bmatrix}^T,$$

where $d := \sqrt{2}/2$. Next two pair of rotations give

$$[S_4^{IY}]_4 = \mathrm{rot}(\varphi_3|1,2)\mathrm{rot}(\varphi_4|3,4)[S_2^{IY}] =$$

$$= \frac{\sqrt{2}}{2} \begin{bmatrix} & & 2 & \\ \sqrt{2} & & & -\sqrt{2} \\ \sqrt{2} & & & \sqrt{2} \\ & -2 & & \end{bmatrix}^T.$$

After rotation $\mathrm{rot}(\varphi_5|1,4)$ we will have a permutation matrix $[S_5^{IY}]_4 = \mathrm{rot}(\varphi_5|1,4)[S_4^{IY}] =$

$$= \begin{bmatrix} & & 1 & \\ & & & -1 \\ 1 & & & \\ & -1 & & \end{bmatrix} = P_4.$$

Consequently up to the permutation matrix $S_4^{IY} :=$

$$= P \begin{bmatrix} c_{1,2} & & & s_{1,2} \\ & s_{3,2} & s_{3,2} & \\ & \overline{s}_{3,2} & s_{3,2} & \\ \overline{s}_{1,2} & & & c_{1,2} \end{bmatrix} \begin{bmatrix} 1 & -1 & & \\ 1 & 1 & & \\ & & 1 & 1 \\ & & 1 & 1 \end{bmatrix} \times$$

$$\times \begin{bmatrix} 1 & & & 1 \\ & 1 & & \\ & & 1 & \\ -1 & & & 1 \end{bmatrix} \begin{bmatrix} d^2 & & & \\ & d & & \\ & & d & \\ & & & d^2 \end{bmatrix}.$$

Similarly, decompositions are found for any $N = 2^n$. For example, for $N = 8$ we have $S_N^{IY} =$

$$= \begin{bmatrix} c_{1,3} & & & & & & & s_{1,3} \\ & c_{3,3} & & & & & s_{3,3} & \\ & & c_{5,3} & & & s_{5,3} & & \\ & & & c_{7,3} & s_{7,3} & & & \\ & & & \overline{s}_{7,3} & c_{7,3} & & & \\ & & \overline{s}_{5,3} & & & c_{5,3} & & \\ & \overline{s}_{3,3} & & & & & c_{3,3} & \\ \overline{s}_{1,3} & & & & & & & cs_{1,3} \end{bmatrix} \times$$

$$\times \begin{bmatrix} 1 & & & -1 & & & & \\ & 1 & -1 & & & & & \\ & 1 & 1 & & & & & \\ 1 & & & 1 & & & & \\ & & & & 1 & & & -1 \\ & & & & & 1 & -1 & \\ & & & & & 1 & 1 & \\ & & & & 1 & & & -1 \end{bmatrix} \times$$

$$\times \begin{bmatrix} c_{1,1} & s_{1,1} & & & & & & \\ \overline{s}_{1,1} & c_{1,1} & & & & & & \\ & & 1 & -1 & & & & \\ & & 1 & 1 & & & & \\ & & & & 1 & -1 & & \\ & & & & 1 & 1 & & \\ & & & & & & c_{1,1} & s_{1,1} \\ & & & & & & \overline{s}_{1,1} & c_{1,1} \end{bmatrix} \times$$

$$\times \begin{bmatrix} 1 & & & & & & -1 & \\ & 1 & & & & & & 1 \\ & & 1 & & & 1 & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & -1 & & & 1 & & \\ 1 & & & & & & 1 & \\ & -1 & & & & & & 1 \end{bmatrix} \times$$

$$\times \mathrm{diag}(d^2, d^2, d^3, d^2, d^2, d^3, d^2, d^2).$$

In the general case, decomposition of S_N^{IY} is formed as a product of n X_N-type matrices (e.g. X-shape matrices in the decompositions of S_4^{IV} and S_8^{IV}) and $]n/2[$ $\mathcal{H}_n$-type matrices (e.g. seconds from the last matrices in the decompositions of S_4^{IV} and S_8^{IV}):

$$S_N^{IY} = \prod_{i=1}^{n} X_{2^n}^i \times \prod_{i=1}^{]n/2[} \mathcal{H}_{2^n}^i,$$

where $]x[$ is the integer part of the number x,

$$X_{2^n}^i =$$

$$\begin{cases} \prod_{k=1}^{2^{n-1}} rot\left(\frac{2k-1}{2^{n+1}}\pi \middle| k, 2^n - k + 1\right), & \text{if } i = n, \\ \prod_{k=1}^{2^{n-2}} rot\left(\frac{-pi}{4}\pi \middle| k, 2^{n-1} - k + 1\right) \oplus & \\ \oplus \prod_{k=1}^{2^{n-2}} \left(rot\frac{-pi}{4}\pi \middle| k, 2^{n-1} - k + 1\right), & \text{if } i = n-1, \\ X_{2^{n-2}}^i \oplus X_{2^{n-1}}^i \oplus X_{2^{n-2}}^i & \text{if } i < n-1, \end{cases}$$

$$\mathcal{H}_{2^n}^i := \mathcal{H}_{2^{n-2}}^{i-1} \oplus \mathcal{H}_{2^{n-1}}^i \oplus \mathcal{H}_{2^{n-2}}^{i-1},$$

where $i = 2, 3, \ldots\ ]n/2[$, $\mathcal{H}_{2^n}^i = I_{2^n}$, if $i >\ ]n/2[\ +1$, i.e. $\mathcal{H}_{2^0}^1 = 1$, $\mathcal{H}_{2^1}^1 = I_{2^1}$, $\mathcal{H}_{2^2}^2 = I_{2^2}$, etc.

The complexity of the described algorithm is the same as the complexity of fast DCT-IV algorithm using the FFT [11] ($N(n+2)/2$ multiplications and $3Nn/2$ additions) which gives the lowest achivable complexity for this transform [11].

References

[1] N. Ahmed and K.R. Rao, *Orthogonal Transforms for Digital Signal Processing*, Springer-Verlag, 1975.

[2] L. Auslander, E. Feig, and S. Winograd, New algorithms for the multidimensional discrete Fourier transform, *IEEE Trans. on ASSP*, vol.31, No. 2, 1983, pp. 388-403.

[3] H.E. Chrestenson, A class of generalized Walsh functions, *Pacific J. Math.*, vol. 5, 1955, pp. 17-31.

[4] P. Duhamel, M. Vetterli, *New 2^n DCT algorithms suitable for VLSI implementation.* Proc. of ICASSP'87, vol. 3, pp. 1805-1808, 1987

[5] K.O. Egiazarian *Discrete Trigonometric Transforms,* R. Traple (ed.), Cybernetics and System'88, pp.1212-1215, 1988, by Kluwer Academic Publishers.

[6] K.O. Egiazarian, S. Agaian, J. Astola, *On a parametric family of discrete trigonometric transforms.* Proc. of IS&T/SPIE's Symp. on Electronic Imaging, Sc. and Technology, pp. 42053, 28, Jan.-2 Feb., 1996

[7] I.J. Good, The interaction algorithm and practical Fourier analysis, *J. Royal Statist. Soc.*, vol. B20, 1958, pp. 361-372.

[8] H. Hou, *The fast Hartley transform algorithm.* IEEE Trans. on Computers, vol. 36, no.2, 1987

[9] A.K. Jain, "A sinusoidal family of unitary transforms", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 1, No. 4, 1979, pp. 356-365.

[10] P. Lee, F.-Y Huang. *Restructured recursive DCT and DST algorithms.* IEEE Trans. on Sign. Processing, vol. 42, n. 7, July, 1994

[11] H.S. Malvar, *Signal Processing with Lapped Transforms*, Artech House, Norwood, Massachusetts, 1992.

[12] J.H. McClellan and C.M. Rader, *Number Theory in Digital Signal Processing*, Englewood Cliffs, New Jersey, 1979.

[13] K.R. Rao and P. Yip, *Discrete Cosine Transform. Algorithms, Advantages, Applications*, Academic Press, 1990.

[14] A.M. Trakhtman and V.A. Trakhtman, *Basics of Theory of Discrete Signals on Finite Intervals*, (in Russian), Moscow, 1975.

[15] Z.Wang, "Fast Algorithms for the discrete W transform and for the discrete Fourier transform", *IEEE Trans. on ASSP*, vol. 32, No. 4, 1984, pp. 803-816.

[16] M.V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, IEEE Press, A.K.Peters, Massachusetts, 1994.

[17] L.P. Yaroslavsky, *Digital Picture Processing. An Introduction*, Springer, Berlin, 1985.

[18] Y. Yoshida, "A Convolution Property for Sinusoidal Unitary Transforms", *IEICE Transactions Fundamentals*, vol. E//-A, No. 5, May 1994, pp. 856-863.