Algorithm of efficient computation of $DCT^{I\text{-}IV}$ using cyclic convolutions

Ihor Prots'ko

Abstract — The general method for efficient computation of discrete cosine transform (DCT) using cyclic convolutions is considered. Forming hashing arrays on the basis of simplified arguments on the basis of discrete cosine transform are analyzed. The examples of four types of discrete cosine transform using proposed method are analyzed.

Keywords — algorithm, cyclic convolution, discrete cosine transforms, hashing array, synthesis.

I. Introduction

Since the middle of 1960s fast computation of discrete Fourier transform (DFT) has been intensively developed. In many studies attention was paid to the prospects of applying only real computing. In 1974 a discrete cosine transform (DCT) [1], which recreates the real of basis function dependence similar to the DFT was proposed. There are 8 types of discrete cosine transform, which are in general examined in [2]. Cosine, sine transforms and Fourier transforms are interrelated with strict mathematical relations that allow us to find an effective technique to compute one transform via another [3].

DCT is widely applied for several reasons. Firstly, basis DCT functions are well approximated to Karhunen-Loyeva transfer function for a big number of stationary stochastic processes which allows presenting the signal of a given accuracy with a minimal number of components. Secondly, DCT is included as part of some efficient DFT algorithms such algorithm [4]. Thirdly, DCT contains a number of special properties, because the conversion is concentrated in the lower indices and more intense and zeroing the remaining output values does not lead to a significant loss of signal energy that prevents edge effects at the block encoded images [5]. DCT is used in many applications, especially in processing digital signals audio and video.

Further intensive development of information technology sets higher requirements before DCT and algorithmic, software and hardware with the performance and development of functional and specific opportunities transforms.

Efficient computing of one or two dimensional DCT, called fast discrete cosine transform (FDCT), has been investigated for more than four decades. A significant numbers of publications devoted of efficient computation of DCT [6] has been written. Multivariate effective computing algorithms can be divided into: size of radix two, split radix, mixed radix, odd size, and prime factors composite transform size.

For the synthesis of efficient algorithms of DCT the following approaches are used:

1) direct matrix factorization of DCT;

2) indirect calculation by FFT or through discrete Hartley transform;

3) algorithms based on the theory of complexity.

It is proved that two types of DCT are polynomial transforms, and in one case it was used to obtain a fast algorithm. It is shown that four types of DCT with group symmetry (properties pertaining to the theory of groups and their representations) and for each of them fast algorithm is derived purely algebraically. Works of fast DCT are summarized and systematized and final step in this direction is the theory of the creation of fast algorithms [7].

One approach of efficient algorithms is the possibility to compute DCT through the cyclic convolutions. Papers [8] use the following index mappings on base primitive root *g* of cyclic group. A lot of the papers [9]-[11] appeared about efficient hardware architectures connected with the computation through cyclic convolution.

The paper is structured as follows. Section 2 describes four types of discrete cosine transform (subsection A) and simplified arguments of basis $DCT^{I\text{-}IV}$ (subsection B) for general algorithm on the basis of hashing arrays for DCT using cyclic convolutions (subsection C). In Section 3 the performance of the proposed general algorithm on examples of four $DCT^{I\text{-}IV}$ types for concrete sizes are analyzed (subsection A,B,C,D) and Section 4 conclusions are presented.

II. Efficient computation of $DCT^{I\text{-}IV}$ using cyclic convolutions

Computation types DCT and IDCT (direct and inverse) are one of the most long-term procedures in information technology, such as for example the compression frame images. This procedure requires the greatest degree of improvements that will speed up the work of software and hardware. Efficient computation of DCT using cyclic convolutions is important and needs development.

Using the method of computation for each type of DCT based on cyclic convolutions has different specifics and is analyzed in this paper.

Ihor Prots'ko with the CAD Department, Lviv Polytechnic University, S. Bandery Str.,12, Lviv, 79046, Ukraine, (e-mail: ioprotsko@gmail.com).

A. Types of discrete cosine transform

Discrete cosine transform reflects the input of a linear combination of weighted basis functions. There are 8 types of discrete cosine transform discussed in [2]. These transforms are further improved by the DFT for real input data. DCTs operate on finite, discrete sequences. The axis of symmetry of continuous discrete cosine transform can be done on the sample or between two samples, which corresponds to a shift in the half interval sampling. This allows different options of transform under the boundary conditions for real input data. Continuation of the input data can be extended: whole sample symmetrically (WS), whole sample asymmetrically (WA), half interval sampling symmetrically (HS) and half interval sampling asymmetrically (HA). There are only two axes of symmetry for a limited sequence and, consequently, a possible set of alternatives ε-type extension that corresponds to 8 types of DCT (Table 1).

Table 1. Set options for ε-type expansion.

ε	WSWS	HSHS	WSWA	HSHA
DCT	DCT ^{I}	DCT ^{II}	DCT ^{III}	DCT ^{IV}

ε	WSHS	HSWS	WSHA	HSWA
DCT	DCT ^{V}	DCT ^{VI}	DCT ^{VII}	DCT ^{VIII}

The relations of direct and inverse computation of four types of DCT can be presented in the following form:

$$(DCT^{I}_{N})^{-1} = (DCT^{I}_{N})^{T} = (DCT^{I}_{N});$$
$$(DCT^{II}_{N})^{-1} = (DCT^{II}_{N})^{T} = (DCT^{III}_{N});$$
$$(DCT^{III}_{N})^{-1} = (DCT^{III}_{N})^{T} = (DCT^{II}_{N});$$
$$(DCT^{IV}_{N})^{-1} = (DCT^{IV}_{N})^{T} = (DCT^{IV}_{N});$$

DCT^{I}, DCT^{IV} specify symmetrical forward and backward transform, and transform of DCT^{II} and DCT^{III} type specify transfer in one second.

Consider the efficient computation of DCT^{I-IV} using cyclic convolutions that will speed up the work of software and hardware for many applications.

B. Define simplified arguments of basis DCT^{I-IV}

Information technologies widely use DCT^{I-IV} types, which can be represented respectively by the following formula:

for DCT^{I}

$$X^{c1}(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N} t_k t_n x(n) \cos[\frac{kn\pi}{N}], \quad k = 0,1,...,N \tag{1}$$

for DCT^{II}

$$X^{c2}(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} t_k x(k) \cos[\frac{k(2n+1)\pi}{2N}], \quad k = 0,1,...,N-1 \tag{2}$$

for DCT^{III}

$$X^{c3}(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} t_n x(n) \cos[\frac{(2k+1)n\pi}{2N}], \quad k = 0,1,...,N-1 \tag{3}$$

for DCT^{IV}

$$X^{c4}(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(k) \cos[\frac{(2k+1)(2n+1)\pi}{4N}], \quad k = 0,1,...,N-1 \tag{4}$$

where

$$t_j = \begin{cases} 1, & j = 0, N \\ 1/\sqrt{2}, & j \neq 0, N \end{cases}.$$

Analyze the structure of the matrix basis for the types of DCT for arguments, where components $c_{k,n}$ are respectively:

for DCT ^{I} $$c_{k,n} = kn\pi / N, (k, n = 0,1,...N); \tag{5}$$

for DCT ^{II} $$c_{k,n} = k(2n+1)\pi / 2N, (k, n = 0,1,...N-1); \tag{6}$$

for DCT ^{III} $$c_{k,n} = (2k+1)n\pi / 2N, (k, n = 0,1,...N-1); \tag{7}$$

for DCT ^{IV} $$c_{k,n} = (2k+1)(2n+1)\pi / 4N, (k, n = 0,1,...N-1); \tag{8}$$

Basis of periodic (2π), symmetric (π) and asymmetric (π/2) functions for each type of DCT, are presented respectively in Table 2.

Table 2.Properties basis for types of DCT.

Types	periodic	symmetric	asymmetric
DCT ^{I}	2N	N	N/2
DCT ^{II}	4N	2N	N
DCT ^{III}	4N	2N	N
DCT ^{IV}	8N	4N	2N

Matrix of arguments C_a for each type of DCT for the property of period is equal respectively

$$C_a^I(k,n) = (k\,n) \bmod (2N), (k, n = 0,1,...N); \tag{9}$$
$$C_a^I(k,n) = [k(2n+1)] \bmod (4N), (k, n = 0,1,...N-1); \tag{10}$$
$$C_a^I(k,n) = [(2k+1)n] \bmod (4N), (k, n = 0,1,...N-1); \tag{11}$$
$$C_a^I(k,n) = [(2k+1)(2n+1)] \bmod (8N), (k, n = 0,1,...N-1); \tag{12}$$

Based on the substitutions of rows of data matrix (9-12) hashing arrays P(n) are formed that define block cyclic structures of basis matrix. Accordance of properties of simplified matrix elements of the arguments is determined by consistent performances:

for DCT ^{I}

$$c_{k,n} = 2N - [(c_{k,n}) \bmod (2N)], \quad if\ [(c_{k,n}) \bmod (2N)] > N; \tag{13}$$

$$\underline{c_{k,n}} = N - \{2N - [(c_{k,n}) \bmod (2N)\}, \quad if\ \{2N - [(c_{k,n}) \bmod (2N)] > N/2; \tag{14}$$
$$otherwize \quad \underline{c_{k,n}} = c_{k,n}$$

for DCT ^{II} or DCT ^{III}

$$c_{k,n} = 4N - [(c_{k,n}) \bmod (4N)], \quad if\ [(c_{k,n}) \bmod (4N)] > 2N; \tag{15}$$

$$\underline{c_{k,n}} = 2N - \{4N - [(c_{k,n}) \bmod (4N)]\},$$
$$if \ \{4N - [(c_{k,n}) \bmod (4N)] > N; \quad (16)$$
$$otherwize \quad \underline{c_{k,n}} = c_{k,n}$$

for DCT ^{IV}

$$c_{k,n} = 8N - [(c_{k,n}) \bmod (8N)],$$
$$if \ \ [(c_{k,n}) \bmod (8N)] > 4N; \quad (17)$$

$$\underline{c_{k,n}} = 4N - \{8N - [(c_{k,n}) \bmod (8N)]\},$$
$$if \ \{4N - [(c_{k,n}) \bmod (8N)] > 2N; \quad (18)$$
$$otherwize \quad \underline{c_{k,n}} = c_{k,n}$$

Simplified matrix arguments complemented matrices Sc of cosine sign, defined by the inequalities

for DCT^{I}

$$Sc[k,n] = \begin{cases} +1, & if \quad 3N/2 < c_{k,n} < N/2 \\ 0, & if \quad c_{k,n} = N/2, 3N/2 \\ -1, & if \quad N/2 < c_{k,n} < 3N/2 \end{cases}, \quad (19)$$

for DCT ^{II} and for DCT ^{III}

$$Sc[k,n] = \begin{cases} +1, & if \quad 3N < c_{k,n} < N \\ 0, & if \quad c_{k,n} = N, 3N \\ -1, & if \quad N < c_{k,n} < 3N \end{cases}, \quad (20)$$

for DCT ^{IV}

$$Sc[k,n] = \begin{cases} +1, & if \quad 6N < c_{k,n} < 2N \\ 0, & if \quad c_{k,n} = 2N, 6N \\ -1, & if \quad 2N < c_{k,n} < 6N \end{cases}. \quad (21)$$

Therefore, expressions (13,15,17) define elements and form hashing array P(n), and then with P(n) expressions (14,16,18) define elements of simplified hashing arrays P'(n) and matrix signs Sc (19,20,21) taking part in the efficient computation of DCT^{I-IV}.

C. Synthesis of efficient algorithms of DCT^{I-IV} on the basis of cyclic convolutions

One approach of efficient algorithms gives the possibility to compute DCT through the cyclic convolutions. Most papers use a transition from discrete transform to compute cyclic convolutions mapping for simple size by Raiders [8] or split composite size on prime factors by Agarwal and Cooley [12] or the combination of these approaches.

Analysis of expressions respectively (9-12) of arguments of DCT basis can be presented the multiplications of integer number by modulo of period. Therefore matrix of arguments (N×N) conforms the table algebraic operation (*= (n x k)mod N) of multiplication by modulo N in general. The algebraic system <N-1,*> with operation on set (1,2…N-1) corresponds equivalent basis matrix of discrete cosine transform in case N the prime size of transform. These algebraic systems <N-1,*> are Abel groups. Besides algebraic system <N-1,*> with prime N presents cyclic group and table of operation is Hankel circular matrix. Elements of cyclic group are equal natural powers of generate element. Generate element of cyclic group is primitive and is not singular. Therefore all elements of cyclic group can be determined the powers of primitive element [13]. Results of analysis are used for forming matrix with circular correlations or cyclic convolutions of the basis matrix DCT.

The proposed approach for efficient computation of discrete cosine transforms is based on decomposition of basis matrix in block-cyclic structures via forming hashing arrays [14]. As a result, block cyclic structure of basis matrix can be describes as a hashing array

$$P(n) = P(L_1)P(L_2)\ldots P(L_k) = (n_{11}, n_1, ..., n_{1L1}) \quad (22)$$
$$(n_{21}, n_{22}, ..., n_{2L2})...(n_{kL1}, n_{kL2}, ..., n_{kLk})$$

where k – the number of subarrays, n_{ij} - element of a subarray; L_i - the number of elements in the subarray $P(n_i)$; n - size of the total array for P(n), which is determined by:

$$n = (L_1 + L_2 + \ldots + L_k) \quad (23)$$

The k number of subarrays in P(n) is determined by the value of N size (simple, easy power, composite) of transform and types of DCT. Hashing array P(n) specifies the order of the input data for performance of the discrete transform.

The properties of symmetry and periodicity of DCT basis lead to lower values representation of elements of subarrays $P'(n_i)$ with supplement respective subarrays of cosine signs $Sc(n_i)$. Submatrices $Sc(n_i)$ contain elements that can be equal to +1,-1,0 (indicate short +,-,0).

Hashing array P(n) of transform defines specific structure of DCT basis matrix reduced to cyclic submatrices. Therefore are the specific parameters P(n) that characterize accordingly modified basis matrix:

- the number of subarrays in hashing array k;
- the size for each of subarrays (L_1, L_2, ..., L_k);
- first element of each subarray n_{i1}, i = 1 (1) k .

The next step in the synthesis algorithm for computation of DCT is to determine cyclic submatrices identity. That finding of identical and quasi identical submatrices (with the same index, but opposite signs) is based on the values of parameters of hashing array P(n) and hashing array P'(n), supplemented array Sc(n) of cosine signs. The parameters of hashing arrays are

$$P'(n) = P'(n_1)P'(n_2)\ldots P'(n_k),$$
$$Sc(n) = Sc(n_1)Sc(n_2)\ldots Sc(n_k) \quad (24)$$

for a given N size and DCT type determined by a simplified matrix, respectively (14,16,18) with elements $\underline{C}_a$, and signs Sc(n) are defined by (19,20,21).

To find identity cyclic submatrices among value elements of basis matrix may in advance, but a large search of all elements requires significant amount of memory to store and associated time costs. More effectively identify only the first elements of submatrices in the analysis of the structure of the basis of coordinates for placement submatrices. That is calculated using the coordinates of the row and column values of the first elements of submatrices and analyzed value of the first elements.

In compliance with coordinates (i,j) of the elements, hashing arrays $P(n_i)$ and $P'(n_i)$ are:

(i\ j) 1 2 3 4 5 6 7 8 9 10 11 12 … n
$P(n_i)(n_{11},n_{12}, n_{13, \dots}, n_{1L1}, n_{21}, n_{22}, n_{23,\dots}, n_{2L2}, \dots, n_{kL1}, n_{kL2}, \dots, n_{kLk})$
$P'(n_i)(n'_{11},n'_{12},n'_{13,\dots},n'_{1L1},n'_{21},n'_{22},n'_{23,\dots},n'_{2L2},\dots,n'_{kL1},n'_{kL2},\dots,n'_{kLk})$,

where integers $0 < n_{ij} < T$; $0 \le n'_{ij} \le n_{ij}$, T – period of DCT basis.

Coordinates of the first elements of submatrices are determined by (i+Li), (j+Li), where Li - is the size of hashing subarrays, which are chosen for condition of membership value of the first elements of the submatrices in the matrix structure to the element of hashing subarrays. The first elements are calculated by relevant coordinates (i, j) and match the element P'(n) of hashing arrays defined by expressions (14,16,18) in accordance with the type of DCT transform. The example of general matrix structure for coordinates is presented in Table 3.

Table 3. Table of coordinates of the first elements of submatrix and their values.

($i+L_i$, $j+L_i$) – $s_{i,j}$ $\underline{c}_{i,j}$;		
(1,1) – $s_{i,j}$ $\underline{c}_{i,j}$;		…
(1+ L_1,1) – $s_{i,j}$ $\underline{c}_{i,j}$;		
(1+ L_1+ L_2,1) – $s_{i,j}$ $\underline{c}_{i,j}$;		…
...		
(1+$L_{1\dots}$+L_k,1) – $s_{i,j}$ $\underline{c}_{i,j}$;	(1+$L_{1\dots}$+L_k, 1+L_k) – $s_{i,j}$ $\underline{c}_{i,j}$;	…

Definition of identity and quasi identity of cyclic submatrices is performed by selecting the coordinates equal to $\underline{c}_{i,j}$ first elements of identical submatrices horizontally. Combining element-wise addition values of input data correspond to the coordinates of first elements of identical submatrices. These values will be used to compute cyclic convolutions. Computation of single cyclic convolutions is performed when analyzing identity submatrices vertically in the matrix structure. For the non-identity submatrices cyclic convolution for specified coordinates is conducted in the case of analyzing process of the whole matrix structure.

Combining the results of convolutions is performed horizontally on the basis of according coordinates of the first elements of submatrices. The resulting output data of transform corresponds to rows with the order according to the elements of hashing array P(n).

III. Algorithms DCT$^{I\text{-}IV}$ using cyclic convolutions in examples

Distribution of cyclic submatrices in basis matrix structures and characteristics of hashing array P(n) determine the complexity of the algorithm for efficient computation of DCT$^{I\text{-}IV}$ types. This approach of efficient computation of DCT uses availability of the fast convolution algorithms [15]. Many implementations and our examples have sequences with reiterative identical groups of elements of cyclic convolution. Consider some cases of identical sequences. In the following case, $h(n) = (h_1, h_2,\dots, h_m, h_1, h_2,\dots, h_m)$, n=2m, the cyclic convolution is equal (25)

$$\begin{pmatrix} h(m)\, h(m) \\ h(m)\, h(m) \end{pmatrix} \otimes \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} h(m) \otimes (x_0 + x_1) \\ h(m) \otimes (x_0 + x_1) \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} ?, \quad (25)$$

if identical group of elements has inverse sign $h(n) = (h_1, h_2,\dots, h_m, -h_1, -h_2,\dots, -h_m)$, the cyclic convolution is equal (26)

$$\begin{pmatrix} h(m) - h(m) \\ -h(m)\, h(m) \end{pmatrix} \otimes \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} h(m) \otimes (x_0 - x_1) \\ -h(m) \otimes (x_0 - x_1) \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}, \quad (26)$$

where $h(m)=(h_1, h_2,\dots, h_m)$, $x_0=(x_1, x_2,\dots, x_m)$, $x_1=(x_{m+1}, x_{m+2},\dots, x_{2m})$, $\otimes$ -operation of cyclic convolution.
Therefore, n/m – number of reiterations of identical group of elements of the sequence of cyclic convolution for size n, which reduces computational complexities by n/m times.

A. *Specific algorithm for DCT^I of size N=15*

Consider the example of synthesis of algorithm and computation of DCT^I of size N=15. The hashing array P(n) and P'(n) are:

P(14)= P(4)P(4)P(4)P(2) =
= (1,2,4,8)(14,13,11,7)(6,12, 9,3)(10,5);
P'(14) = (1,2,4,7)(1,2,4,7)(6,3,6,3)(5,5),
Sc(14) = (+, +, +, -)(-, -, -, +)(+, -, -, +)(+, -).

Definition of identity cyclic submatrices is performed by selecting the coordinates of the first elements of identical submatrices without signs in the basis matrix. In correspondence with coordinates (i,j), hashing array elements $P(n_i)$ and $P'(n_i)$ are:

(i, j) 1 2 3 4 5 6 7 8 9 10 11 12 13 14
P(14)= (1, 2, 4, 8)(14,13,11,7) (6, 12, 9, 3) (10, 5);
P'(14)= (1, 2, 4, 7) (1, 2, 4, 7) (6, 3, 6, 3) (5, 5),
Sc(14)= (+, +, +, -) (-, -, -, +) (+, -, -, +) (+, -).

Coordinates of the first elements of submatrices are determined by (i + Li), (j + Li), starting with i = 1, j = 1. The value of the first elements of submatrices are calculated by matching the coordinates (i, j) and the elements of P(n) hashing array using formula [(n_i x n_j) mod 2N], in the case of a value greater than N using simplified expression (13,14).

Table 4. Table of coordinates and first elements with signs of submatrix DCT^I, N = 15

($i+L_i$, $j+L_i$) - $s_{ij}n_{ij}$ (sign and first elements of submatrices)			
(1,1) – +1;	(1,5) – -1;	(1,9) – +6;	(1,13)– +5;
(5,1) – -1;	(5,5) – +1;	(5,9) – -6;	
(9,1) – +6;	(9,5) – -6;	(9,9) – +6;	(9,13) – 0;

(13,1) – +5;	(13,9) – 0;	(9,13) – -5;

Table 4 summarizes the basis matrix of arguments of dimension (14x14), where the number of horizontal and vertical elements is indicated. According to hashing array P(14) for DCT^I, N=15 to compute the output values via convolution with identical group. With hashing arrays P'(14) basis matrix can be reproduced, whose arguments resulted in a form of cyclic submatrices without signs. The matrix of simplified argument and of cosine signs of basis DCT^I type for N=15 is presented in Table 5, which corresponds to the generalized Table 4.

Table 5.Tables of values of simplified elements and signs of matrix DCT^I, N=15

n \ k	1:	2:	4:	8:	14:	13:	11:	7:	.
1:	1	2	4	7	1	2	4	7	.
2:	2	4	7	1	2	4	7	1	.
4:	4	7	1	2	4	7	1	2	.
8:	7	1	2	4	7	1	2	4	.
14:	1	2	4	7	1	2	4	7	.
13:	2	4	7	1	2	4	7	1	.
11:	4	7	1	2	4	7	1	2	.
7:	7	1	2	4	7	1	2	4	.
6:	6	3	6	3	6	3	6	3	.
12:	3	6	3	6	3	6	3	6	.
9:	6	3	6	3	6	3	6	3	.
3:	3	6	3	6	3	6	3	6	.
5:	5	5	5	5	5	5	5	5	.
10:	5	5	5	5	5	5	5	5	.

(continuation of Table 5)

n \ k	6:	12:	9:	3:	5:	10:
1:	6	3	6	3	5	5
2:	3	6	3	6	5	5
4:	6	3	6	3	5	5
8:	3	6	3	6	5	5
14:	6	3	6	3	5	5
13:	3	6	3	6	5	5
11:	6	3	6	3	5	5
7:	3	6	3	6	5	5
6:	6	3	6	3	0	0
12:	3	6	3	6	0	0
9:	6	3	6	3	0	0
3:	3	6	3	6	0	0
5:	0	0	0	0	5	5
10:	0	0	0	0	5	5

(continuation of Table 5 of signs of simplified elements)

1:	+	+	+	-	-	-	-	+	+	-	-	+	+	-
2:	+	+	-	+	-	-	+	-	-	-	+	+	-	+
4:	+	-	+	+	-	+	-	-	-	+	+	-	+	-
8:	-	+	+	+	+	-	-	-	+	+	-	-	-	+
14:	-	-	-	+	+	+	+	-	-	+	+	-	-	+
13:	-	-	+	-	+	+	-	+	+	+	-	-	+	-
11:	-	+	-	-	+	-	+	+	+	-	-	+	-	+
7:	+	-	-	-	-	+	+	+	-	-	+	+	+	-
6:	+	-	-	+	-	+	+	-	+	-	-	+	0	0
12:	-	-	+	+	+	+	-	-	-	-	+	+	0	0
9:	-	+	+	-	+	-	-	+	-	+	+	-	0	0
3:	+	+	-	-	-	-	+	+	+	+	-	-	0	0
5:	+	-	+	-	-	+	-	+	0	0	0	0	-	+
10:	-	+	-	+	+	-	+	-	0	0	0	0	+	-

Hashing array P(n) → (1,2,4,8,14,13,11,7,6,12,9,3,10,5) specifies the order of elements of input data of the discrete cosine transform using cyclic convolutions.

Computation of cyclic convolution is performed for combined input data for identity and quasi identity submatrices selected for analysis horizontally and vertically. The number of cyclic convolutions DCT^I of size N=15 is four of 4-point convolution with identical sequences and two of one points.

The resulting structure for DCT^I of size N=15 consists such components (Fig.1): ±/U – element-wise addition/subtraction units, n-point CCU – cyclic convolution units.

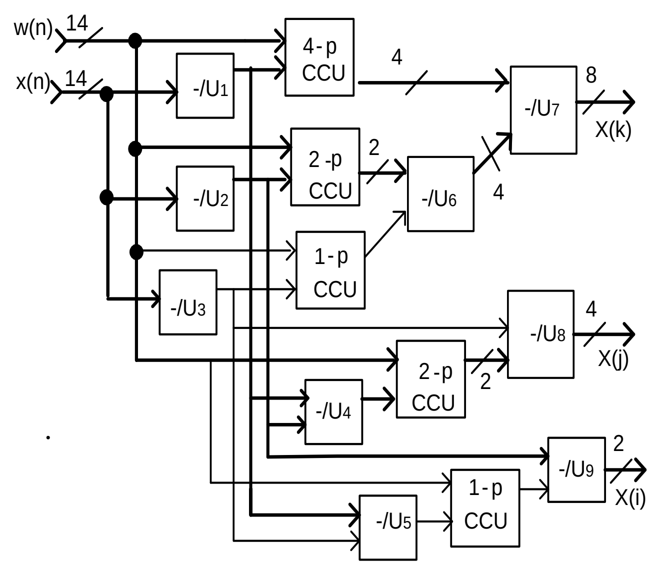

Fig.1 The stucture for DCT^I of size N=15.

The order of sequence of coefficients:
w(n)={cos (φ), cos (2φ), cos(4φ), cos(8φ), cos(14φ), cos(13φ), cos(11φ), cos (7φ), cos (6φ), cos(12φ), cos(9φ), cos(3φ), cos(10φ), cos(5φ)}, where φ=π/15.
The order of sequence of input data :
x(n)={x(1), x(2), x(4), x(8), x(14), x(13), x(11), x(7), x(6), x(12), x(9), x(3), x(10), x(5)},
which are combined in corresponding element-wise addition/subtraction by consistent performances in the $\pm/U_i$ (i=1,2,…,9)

Combining without values x(0), x(N) in $\pm/U_{6,7,8}$ units the results of convolutions is performed horizontally on the basic coordinates according to the first elements of submatrices. Output data without values X(0), X(N) of transform correspond the order of hashing array in a result of computation.

B. Specific algorithm for DCT^{II} of size N=8

Consider the example of a generalized scheme for synthesis of algorithm and computation of DCT^{II} for size N = 8. Basis matrix of DCT^{II} arguments with elements for (10) contains the values (2n +1) of elements in the first row quantity – 16, which cover the entire period equal to 4N, and k of the elements in the first column quantity – 16, which cover the entire half of period equal to 4N.

A difference of values in rows k and columns (2n+1) requires transition from hashing array P(n) to the appropriate hashing array indices of the rows Pr(n) and columns Pc(n):

Pr(16) = (1, 3, 9, 5, 15, 13, 7, 11)(2, 6, 14, 10) (4, 12)(8)(0);
Pr'(16) = (1, 3, 7, 5, 1, 3, 7, 5) (2, 6, 2, 6,) (4, 4) (8) (0);
Sc(16) = (+,+, -, +, -, -, +, -) (+, +, -, -) (+,-) (0) (+1);

Pc(16) = (1,3,9,27,17,19,25,11)(31,29,23,5,15,13,7,21);
Pc'(16) = (1,3,7,5,1,3,7,5) (1,3,7,5,1,3,7,5);
Sc(16) = (+,+, -, +, -, -, +, -) (-,-,+,-,+,+,-,+).

According to hashing array Pr(n) for DCT^{II} horizontal 14 rows are selected that allow computing 8 output data through cyclic convolutions without (8)(0). The convolution of two identical elements (4, 4) is replaced by convolution with a single point, respectively, for lines 4, 12. Table 6 summarizes the basis matrix of arguments (14x16), where the numbers of values of horizontal and vertical are indicated.

Table 6. Table of coordinates and first elements of matrix DCT^{II}, N=8.

<table>
<tr><td colspan="4">($i+L_i$, $j+L_i$) - $s_{ij}n_{ij}$
(sign and first elements of submatrices)</td></tr>
<tr><td colspan="2">(1,1) – +1;</td><td colspan="2">(1,9) – -1;</td></tr>
<tr><td>(9,1)– +2;</td><td>(9,5)– +2;</td><td>(9,9) – +2;</td><td>(9,13) – +2;</td></tr>
<tr><td colspan="4">(13,1) – +4;</td></tr>
</table>

With hashing arrays Pr(16), Pc(16) arguments of the basis matrix can be reproduced, which result in a form of cyclic submatrices without signs. Part matrix of simplified arguments without signs of cosine basis transform N= 8 is presented in Table 7, which corresponds to the generalized Table 6.

Table 7. Table of values of simplified elements without signs of matrix DCT^{II}, N=8.

0:	1:	4:	13:	8:	9:	12:	5:	…	7:	6:	3:	10:
1	3	7	5	1	3	7	5	…	1	3	7	5
3	7	5	1	3	7	5	1	…	3	7	5	1
7	5	1	3	7	5	1	3	…	7	5	1	3
5	1	3	7	5	1	3	7	…	5	1	3	7
1	3	7	5	1	3	7	5	…	1	3	7	5
3	7	5	1	3	7	5	1	…	3	7	5	1
7	5	1	3	7	5	1	3	…	7	5	1	3
5	1	3	7	5	1	3	7	…	5	1	3	7
2	6	2	6	2	6	2	6	…	2	6	2	6
6	2	6	2	6	2	6	2	…	6	2	6	2
2	6	2	6	2	6	2	6	…	2	6	2	6
6	2	6	2	6	2	6	2	…	6	2	6	2
4	4	4	4	4	4	4	4	…	4	4	4	4
4	4	4	4	4	4	4	4	…	4	4	4	4

The values of columns require transition from hashing array to the appropriate hashing array indices (n-1)/2 of Pc(n):

(1,3,9,27,17,19,25,11)(31,29,23,5,15,13,7,21) →
(0, 1, 4, 13, 8, 9.12, 5) (15, 14, 11, 2, 7, 6, 3, 10).

Hashing array of Pc(n) specifies the order of input data [x, x'], where initial order is:
x(n)={x(0),x(1),x(2),x(3),x(4),x(5),x(6),x(7)} and
x'(n)={x(7),x(6),x(5),x(4),x(3),x(2),x(1),x(0)}.

Performance of element-wise additions of input data will be used for analysis of identity and quasi identity cyclic submatrices placed horizontally. In result analysis of submatrices for Table 6 horizontally and vertically, the number of cyclic symmetric convolutions of DCT^{II} of size N=8 are the 8-point cyclic convolution with identical sequences, which resulted in four defined output data, and 4-point cyclic convolution with identical sequences, which resulted in two output data. The remaining output datum is determined through one point products.

The resulting structure for DCT^{II} of size N=8 consists such components (Fig.2): BRC– buffer register of coefficients, BRD – buffer register of input data, ±/U – element-wise addition/subtraction units, n-point CCU – cyclic convolution units.

The order of sequences:
w(n)={cos (φ), cos (3φ), cos(7φ), cos(5φ), cos(2φ), cos(6φ), cos(4 φ)}, where φ=π/16;
x(n)=2{x(0),x(1),x(4),x(2),x(7),x(6),x(3),x(5)},
which are combined in corresponding element-wise addition/subtraction by consistent performances in the $\pm/U_1$, $\pm/U_2$, $-/U_3$, $+/U_4$ units.

The CCU units perform 4-point and 2-point cyclic convolutions (Fig.1). There are different implementations of the CCU among them suitable the systolic arrays or memory-based unit of cyclic convolutions.

The output data of cyclic convolutions are performed on the basis of accordance coordinates of the first elements of submatrices horizontally (Table 6). Output data of transform in the result of computation are scaled by two.

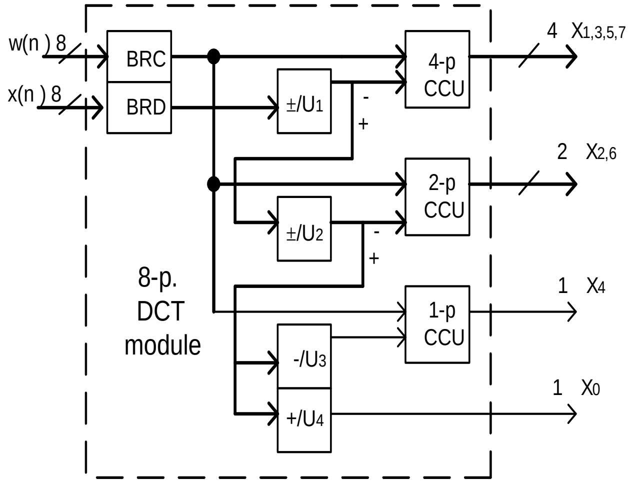

Fig.2. The stucture for DCT^{II} of size N=8.

C. *Specific algorithm for DCT^{III} of size N=8*

Consider the example of a synthesis of algorithm and computation of DCT^{III} N = 8. In contrary with DCT^{II}, basis matrix of DCT^{III} arguments with elements for (11) contains values (2k +1) of the elements in the first column of quantity - 16 and covers the entire period equal to 4N. The first row contains values n of quantity - 16 and cover the entire interval equal to 2N. Hashing array for rows is:

Pr(16) =(1,3,9,27,17,19,25,11)(31,29,23,5,15,13,7,21).

The values of rows require transition from hashing array to the appropriate hashing array indices (2k+1)→k the rows Pr(n):
Pr(16) = (1, 3, 9, 27, 17, 19, 25, 11)(31, 29, 23, 5, 15, 13,7,21)
→ Pr(16) = (0, 1, 4, 13, 8, 9,12, 5)(15, 14, 11, 2, 7, 6, 3,10).
The first eight rows define all 8 output data, where four output values correspond to 13→2, 8→7, 9→6, 12→3.
Hashing array for column is defined (15) and has values of elements less than 2N :

Pc(16) =(1, 3, 9, 5, 15, 13, 7, 11)(2, 6, 14, 10) (4, 12)(8)(0).

Simplified hashing array for column and array of cosine signs have the following form:

Pc'(16) = (1, 3, 7, 5, 1, 3, 7, 5) (2, 6, 2, 6)(4, 4)(8)(0),
Sc(16)=(+, +, -, +, -, -, +, -) (+,+, -, -) (+, -) (0)(0).

Simplified hashing array of row and array of cosine signs have the following form:

Pr'(16) = (1, 3, 7, 5, 1, 3, 7, 5) (1, 3, 7, 5, 1, 3, 7, 5),
Sc(16)= (+,+,-,+,-,-,+,-) (+,+,-,+,-,-,+,-).

Identity cyclic submatrices are defined via analysis of submatrices distribution in the structure of the basis matrix for Table 8. The following Table 8 summarizes basis matrix of arguments of the dimension (8x16).

Table 8. Table of coordinates and first elements of submatrix DCT^{III}, N=8.

<table>
<tr><th colspan="5">($i+L_i$, $j+L_i$) - $s_{ij}n_{ij}$ (sign and value and first elements of submatrices)</th></tr>
<tr><td rowspan="4">(1,1) – +1;</td><td rowspan="2">(1,9) – +2;</td><td>(1,13) – +4;</td><td rowspan="4">(1,15) – +8;</td><td rowspan="4">(1,16) – 0;</td></tr>
<tr><td>(5,13) – +4;</td></tr>
<tr><td rowspan="2">(5,9) – +2;</td><td>(9,13) – +4;</td></tr>
<tr><td>(13,13) – +4;</td></tr>
</table>

Hashing array of Pc(n) specify the order of the input data when we conduct the discrete cosine transform through cyclic convolutions. The matrix of simplified arguments without cosine signs of basis transform N= 8 is presented in Table 9, which corresponds to the generalized Table 8.

Table 9. Table of values of simplified elements without signs of matrix DCT^{III}, N=8.

k \ n	1:	3:	9:	5:	15:	13:	7:	11:	.
0:	1	3	7	5	1	3	7	5	.
1:	3	7	5	1	3	7	5	1	.
4:	7	5	1	3	7	5	1	3	.
13:	5	1	3	7	5	1	3	9	.
8:	1	3	7	5	1	3	7	5	.
9:	3	7	5	1	3	7	5	1	.
12:	7	5	1	3	7	5	1	3	.
5:	5	1	3	7	5	1	3	7	.

(continuation of Table 9)

k \ n	*2:*	*6:*	*14:*	*10:*	*4:*	*12:*	*8:*
0:	2	6	2	6	4	4	8
1:	6	2	6	2	4	4	8
4:	2	6	2	6	4	4	8
13:	6	2	6	2	4	4	8
8:	2	6	2	6	4	4	8
9:	6	2	6	2	4	4	8
12:	4	6	2	6	4	4	8
5:	6	2	6	2	4	4	8

Performance of element-wise additions of input data will be used for identity and quasi identity cyclic submatrices placed horizontally. Computation of cyclic convolution is performed once for combined input data for identity and quasi identity submatrices selected for analysis vertically. The number of cyclic convolutions of DCT^{III} of size N = 8 are the 8-point cyclic convolution with identical sequences and one 4-point cyclic convolution with identical sequences also. The remaining values are determined by one point products.

Combining the results of cyclic convolutions is performed horizontally at the base coordinates of the first elements of submatrices. Output data of transform of computation are scaled by two in result. Therefore horizontal 8 rows of respective hashing array Pr(n) =(0, 1, 4, 13, 8, 9, 12, 5) are selected, allowing to compute output data values via cyclic convolutions.

D. Specific algorithm for DCT^{IV} of size N=8

Consider the example of algorithm and computation of DCT^{IV} with the values of the arguments [(2k+1)(2n+1)] for size N = 8. Basis matrix DCT^{IV} for size N =8 form the hashing array of arguments with elements smaller than 4N:

P(16)=(1,3,9,27,17,13,25,11,31,29,23,5,15,19,7,21).

The values of elements require transition from hashing array to the appropriate hashing array with values of elements (2k+1)→k of P(n):

P(16)= (0,1,4,13,8,6,12,5,15,14,11,2,7,9,3,10).

Simplified hashing array and array of cosine signs have the following form:

P'(16)= (1,3,9,5,15,13,7,11, 1,3,9,5,15,13,7,11);
Sc(16)= (+, +, +, -, -, +, -, +, -, -, -, +, +, -, +, -).

Respective 8 rows of hashing array P(16) for DCT^{IV} are selected which allows computing the 8 output data through cyclic convolutions. Part of matrix of simplified arguments and cosine signs of basis transform N= 8 is presented in Table 10.

Table 10. Table of values of simplified elements and signs of matrix DCT^{IV}, N=8.

k \ n	0:	1:	4:	13:	8:	6:	12:	5:	…
0:	1	3	9	5	15	13	7	11	…
1:	3	9	5	15	13	7	11	1	…
4:	9	5	15	13	7	11	1	3	…
13:	5	15	13	7	11	1	3	9	…
8:	15	13	7	11	1	3	9	5	…
6:	13	7	11	1	3	9	5	15	…
12:	7	11	1	3	9	5	15	13	…
5:	11	1	3	9	5	15	13	7	…
									…
0:	+	+	+	-	-	+	-	+	…
1:	+	+	-	-	+	-	+	-	…
4:	+	-	-	+	-	+	-	-	…
13:	-	-	+	-	+	-	-	-	…
8:	-	+	-	+	-	-	-	+	…
6:	+	-	+	-	-	-	+	+	…
12:	-	+	-	-	-	+	+	-	…
5:	+	-	-	-	+	+	-	+	…

Hashing array of P(16) specifies the order of input data when conducting the discrete cosine transform on base cyclic convolutions. Performance of element-wise subtractions of input data will be used for cyclic convolution with submatrices placed horizontally. The number of cyclic convolutions DCT^{IV} of size N = 8 are only the 16-point cyclic convolution with identical sequences.

Output data of transform as a result of computation is scaled by two and determined for:

X(0), X(1), X(4), -X(2), -X(7), X(6), -X(3), X(5).

Three output values must be taken with the opposite sign according to P(n)=(0,1,4,13,8,6,12,5), where 2→-13, 7→-8, 3→-12 correspond to one another. Therefore horizontal 8 rows of according hashing array P(n) are selected allowing to compute output data values via cyclic convolutions.

IV. Conclusions

Efficient computation of each of the four types of DCT can be performed on the basis of response of hashing arrays and then using fast cyclic convolution algorithms. Functions of arguments for DCT^{II} and DCT^{III} have different forms of rows and columns. Consequently a separate hashing array for row and column has to be applied according to this approach. Hashing arrays for direct DCT^{II} and hashing arrays for inverse $IDCT^{II} = DCT^{III}$ or vice versa differ number of indices for Pr(n) and Pc(n) and corresponding structure of basis matrix with cyclic submatrices. As a result, main characteristics of algorithm that specifies the types of DCT^{I-IV} are: function of basis arguments; initial dimension of basis matrix; order of sequences of input data; sequence of output data; convolution with identical sequences; version of hashing arrays; axes of symmetry for size of transform.

In general computational expenses of proposed technique for DCT^{I-IV} on base cyclic convolutions on the steps of performance can present in following form

$$C = C_I^{+} + \sum_i Cp_i^{+,*} + C_{III}^{+} \quad , \qquad (27)$$

where C^{+}_{I} – addition/subtraction on the stage of unions identity and quasi identity cyclic submatrices placed horizontally; $Cp_i^{+,*}$– arithmetic operations for computation of *p*-point cyclic convolution, *i*- number of cyclic convolution; C^{+}_{III} – addition/subtraction on the stage of unions of the results of cyclic convolutions and some input data. Computation of cyclic convolutions includes all multiplications of algorithm for DCT^{I-IV}.

Separate computations of cyclic convolutions, which are structured according to this approach to basis matrix, and the combinations of convolutions results make the proposed technique important for concurrent programming and for its implementation in parallel systems.

References

[1] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform”, *IEEE Trans. Comput.*, C-23, pp. 90-93, 1974.

[2] G. Strang, “The Discrete cosine transform,” *SIAM Review*, vol.41, no.1, p.135-147, 1999.

[3] R. Gluth, “Regular FFT-related transform kernels for DCT/DST-based polyphase filter banks,” in *Proc. IEEE ICASSP 1991*, Toronto, Canada, May 1991, pp.2205-2208.

[4] H. V. Sorensen, D. L. Jones, M. T. Heideman, C. S. Burrus, “Real Valued Fast Fourier Transform Algorithms,” *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 35, no. 6, pp.849–863, June 1987.

[5] A. V. Oppenheim, R. W. Schafer, *Discrete-time signal processing*. Prentice Hall, NJ, 1999.

[6] V. Britanak, P. Yip, and K. R. Rao, *Discrete Cosine and Sine Transforms*. New York, NY: Academic Press, 2007.

[7] S. Egner, M. Pueschel, "Automatic generation of fast discrete signal transforms," *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 1992–2002, September 2001.

[8] C. M. Rader, "Discrete Fourier transforms when the number of data samples is prime," *Proc. IEEE*, vol. 56, pp. 1107-1108, 1968.

[9] Y.-H. Chan, W.-C. Siu, "A cyclic correlated structure for the realization of discrete cosine transform," *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, vol. 39, no. 2, pp. 109–113, February 1992.

[10] P. K. Meher, "Systolic designs for DCT using a lowcomplexity concurrent convolutional formulation," *IEEE Trans. Circuits & Systems for Video Technology*, vol. 16, no. 9, pp. 1041–1050, Sept. 2006.

[11] Cheng C., Parhi K.K. "Hardware efficient fast DCT based on novel cyclic convolution structures," *IEEE Trans. Signal Proc.*, vol. 54, no.11, pp. 4419-4434. 2006,

[12] Agarwal R. C, and J. W. Cooley, New algorithms for digital convolution." *IEEE Trans. on Acoustics, Speech, and Signal Processing ASSP-25*, No. 5 (October 1977): 392 - 409.

[13] I. Prots'ko, "Analysis algebraic system of arguments for prime size DHT," *in Proceeding of the XI International Conference TCSET'2012*, Lviv-Slavske, february 2012. –p.428.

[14] I.Prots'ko, Ukraine Patent 96540, G06F 17/16 (2006.01), H03M 7/30 (2006.01).

[15] A. F. Breitzman, "Automatic Derivation and Implementation of Fast Convolution Algorithms", PHD thesis, Drexel University, January 2003.

Ihor Prots'ko was born in Sambir, Ukraine, in 1958. He received the engineer degree and Ph.D. degree from Lviv Polytechnic University in 1981and 1999, where he is currently Assoc. Prof. CAD Department. His current research interests include fast algorithms of discrete transforms and special-purpose digital means.