Parallel-Hierarchical Computing System for Multi-Level Transformation of Masked Digital Signals

Andriy A. Yarovyy[1], Leonid I. Timchenko[2], Nataliya I. Kokriatskaia[1,2]
[1]*Vinnytsia National Technical University, 21021, Ukraine*
[2]*State University for Transport Economy and Technologies, Kyiv, 03049, Ukraine*
axa@vinnitsa.com

***Abstract*—Paper deals with organization principles of parallel-hierarchical (PH) transform for multistage perception and processing, compression and recognition of information in computational systems which make use of computational scheme similar to neural.**

Unified methodological approach was developed for analysis of parallel processes. This approach considers influence of structural hierarchy in dynamics. It tracks processes of spatial areas transformation of correlated and generation of uncorrelated in time elements of generated PH network, at the time of transition of the network from one stable stage to another.

The main feature of the proposed method is the studying of the dynamics of spatial-correlated procedure for transformation between current and output units in PH network. This procedure gives an opportunity to present processing in PH network as the process of parallel-sequential transformation of diverse image components and consideration of temporal characteristics of transformation.

The features analysis of structural-functional organization of networking model of PH transformation of the information environments and corresponding masks generation methods is realized in the given research. On the base of research carried out the method of optimized masks formation while information coding in PH transformation is proposed. The software package, characterized by increased efficiency of direct and reverse PH transformation is developed.

***Index Terms*—codes, computer science, image processing, parallel processing, transform coding.**

I. Introduction

The solution of the problem dealing with rapid transformation of large information arrays to provide its efficient recording, storage, processing and reading is connected with the creation of fast- acting coding/decoding devices [1]-[2].

Rate of coding-decoding process of digital data array depends greatly on the realized algorithm of digital processing. Level of hardware development of multichannel digital coding/decoding devices of large information arrays is supported by serial algorithms of digital processing, this is connected with considerable losses of time, needed to perform serial in time coding/decoding process [1]-[6].

This problem becomes very actual in the field of image coding, where it is expedient to use parallel coding.

Unlike widely used types of video information coding, for instance, differential coding, code-pulse, delta modulation, based on the principle of serial coding of differential information, the given research suggests to use parallel-pyramidal principle of processing the results of data array coding, distributed in space-time area, that leads to considerable increase of algorithmic processing speed, compression of data arrays and provision of natural form of signals (images) redescription [7].

In previous research, certain mathematical models of networking method of parallel-hierarchical (PH) transformation and their applied usage were considered [8]-[11]. Here, we will define abstract model of networking structure of the transformation.

II. Basic Notions And Definitions Of Networking Model Of Multilevel Transformation Of Masked Signals

Theories of graphs, notions of linear lists and trees [12]-[13] used in the theory of information systems are structures with arranged connections. In this sense structural information (connections between data elements) while their description is predetermined by the type of the structure and is a structure with flexible hierarchy. One of the ways of parallelism realization in the process of handling of multi-connection structures is “regularization”, that provides their description by means of regular networking transformation structure. In this case, the information, regarding connections is included in networking transformation structure in explicit form, i.e., is presented by data elements. The abstract model of networking structure of transformation will be considered in details in the given paper.

Let us introduce some notions related to tree-like model of transformation networking structure with regular connections (Fig. 1) [7].

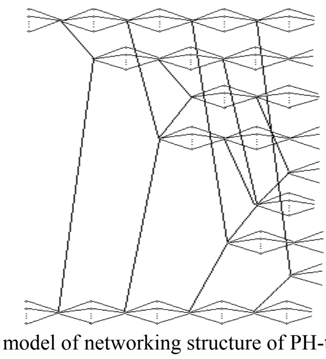


Figure 1. Tree-like model of networking structure of PH-transformation

Let graph $G = (V, E)$ be a structure of data processing, comprising the sets of V nodes and sets of E edges. Graph is directed, if edges are presented in the form of arranged pairs

Digital Object Identifier 10.4316/AECE.2012.03002

of nodes. Transformation tree is defined by directed graph that possesses the following properties: only root nodes do not have arcs making part of them; each node comprises the set of arcs, number of which is determined by non-linear structure of processed data; only one route, i.e., a single finite set of edges follows from each root to the node.

Structure of data processing is identified with directed graph; data elements corresponding to nodes, and directed arcs, connecting nodes describe various dependencies among elements and are marked correspondingly.

Structure of data processing of transformation model $D=\{K, \Psi\}$ is defined by the set of K nodes and the set $\Psi=\{f_1 f_2 \dots\}$ of f_i functions: K$\rightarrow$1 and $1\rightarrow K$ which represent the set of nodes in one node and vice-versa. Two nodes K and K' (K, $K'\in$K) are connected by the arc f_i, if $K'=f_i(K)$, the arc being directed from K and K'. Thus, the tree of networking PH-transformation contains two kinds of sub-trees: convergent and divergent.

Structure of data processing $D=\{K, \Psi\}$ where a set of nodes is represented in one node, i.e., $K\rightarrow 1$; function Ψ is determined by F*-criterion, and creates convergent structure of a sub-tree.

Structure of data processing $D^*=\{K, \Psi^*\}$ where one node is represented by a set of nodes K, i.e., $1\rightarrow K$, and function Ψ^* is determined by Q^*-transformation function, and creates divergent structure of a sub-tree.

Property 1. Sequential in time formation of convergent (K) and divergent (D) sub-trees will create K-D tree.

Property 2. Neighboring K-D trees of one level are shifted in time relatively each other by one K and D of sub-trees. Non-linear structure K-D of trees (Fig. 1) forms generalized tree of the network.

Intersections of generalized tree are the nodes K and D of sub-trees, having identical routes to root nodes.

Tail nodes are single nodes K of sub-trees of generalized tree of networking transformation, one node K of sub-trees being in its intersections.

Networking tree is a finite set K-D of trees, neighboring trees of one level are shifted relatively each other by K-D tree, and the number of intersections is determined by a number of tail nodes, location of which in the sequence of intersections is described – ($2c$+3), where c – is the number of intersection, c=0, 1, ….

Branch of networking tree is any randomly formed, according to previous definition, K-D tree.

Property 3. A number of networking tree levels is determined by a number of its tail nodes, increased by one.

Neighboring K-D trees of the same level are those, shifted by K-D tree.

The suggested method of parallel transformation of large data arrays we will consider applying networking algorithm [7], [9], [14], basic properties of which are parallelism and hierarchy, synchronism and determinacy. The network comprises a set of finite sets Ω, set of elements A and is conventionally divided into a number of levels.

Network of PH-transformation is a totality of such characteristics:

$C\in\left(\Omega, A, Q^*, F^*\right)\in\left\{M_1^1(t_0); M_2^1(t_0);\dots;\right.$ $\left.M_h^1(t_0); M_1^2(t_1);\dots; M_n^u(t_s)\right\}$ – is finite sets totality, $h\geq 2$ – is a number of output sets, $u\geq 2$ – is consecutive number of level; $n\geq 2$ – is consecutive number of u-th level; t_s – is clock period or step, where corresponding set, $S\geq 1$ is formed, t_0 – is the first or initial step, where output sets of the first level are formed. $A=\left\{a_1^1(t_1); a_2^1(t_1);\dots; a_h^1(t_1); a_1^1(t_3);\dots; a_{n'}^{u'}(t_s)\right\}$ – is finite set of elements, $u'\geq 2$ – is consecutive number of the level; $n'\geq 1$ – is consecutive number of the set, the element belongs to; t_s' – is clock period, where corresponding element is formed.

Totality of sets and set of elements are intersected.

$\Omega\cap A=\varnothing,\ M_i^j(t_s)=\left\{a_1^{j-1}(t_s), a_2^{j-1}(t_s)\dots a_k^{j-1}(t_s)\right\}$,

$M_i^j(t_s)$ - is output set for j-th level.

F* - is criterion of element selection from the set $a_i^j(t_s)=F^*\left[M_i^j(t_{s-1})\right]$ (transition from the set $M_i^j(t_{s-1})$ to the element $a_i^j(t_s)$), Q^*– is function of set transformation, $Q^*_{a_i^j(t_s)}\left[M_i^j(t_{s-1})\right]=M(t_{s+1})$ (transition from element $a_i^j(t_s)$ to set $M(t_{s+1})$). Capacity of output sets $M_i^1(t_0)$ is number m, quantity of which is H:

$M_1^1(t_0)=\{a_{11}; a_{12};\dots a_{1m}\};\quad M_2^1(t_0)=\{a_{21}; a_{22};\dots a_{2m}\};\dots;$
$M_h^1(t_0)=\{a_{h1}; a_{h2};\dots a_{hm}\}$

Each of these sets will be transformed by a single networking algorithm, and all the sets are processed parallely. Element a_{ij} is chosen from the set M_i, and element $a_i^1(t_1)=a_{ij}$ is an element of the network C.

Criterion, to be used for selection of element from the set, we will denote as F*-criterion, $a_i=F^*(M), a_i\in M$. Taking into account the selected element the transformation of the given set is obtained, as a result new set of the same capacity is formed, where all the elements, being equal to selected one (if there are any); are somehow marked, for instance, are brought to zero: $a_{i1}; a_{i2}; 0; a_{i4}; 0; 0; a_{i7};\dots 0; a_{im}$, in the given example $a_i^1(t_1)=a_{i3}=a_{i5}=a_{i6}=a_{ij}=a_{im-1}$. Such operation we will denote as Q*-transformation.

Further, from obtained set, applying F* creation such element as $a_{ij_1}, a_j^1(t_3)=a_{ij_1}$ is selected ($a_{ij_1}=a_{ij}$) and Q*-transformation is formed. As a result, the set is formed, in which all the elements, equal a_{ij_1}, are marked. Iteration transformation is performed until all the elements of the output set are marked. Such set is marked as zero, and further transformation is not performed.

Zero set is such a set, where all the elements are marked as a result of Q*-transformation.

Let us define process of transformation into zero set as matching process. Obviously, the fewer cycles necessary to sample a set of zero, the better the convergence of the process. We will denote such type of transformation as horizontal or branch transformation.

Transformation of initial set, taking into account the intermediate results prior to obtaining zero set is called a branch.

Considering all H of initial sets of the first level while selecting element $a_j^1(t_j), i = \{1,2,\ldots,h\}; \; j = \{1,3,5,\ldots\}$ from each set, new sets are formed, where i – is consecutive number of initial set, t_j – is clock period, where the element is selected. At first selection of elements from H input sets in clock period t_1 new set $M_1^2(t_1) = \{a_1^1(t_1), a_2^1(t_1), a_3^1(t_1), \ldots, a_n^1(t_1)\}$ is formed. At the second step of t_3 transformation from initial sets one more set from H elements is formed: $M_2^2(t_3) = \{a_1^1(t_3), a_2^1(t_3), \ldots a_h^1(t_3)\}$. Such transformation is performed until all output sets become zero.

Sets $M_1^2(t_1); M_2^2(t_3); \ldots; M_i^2(t_j)$ – are initial sets of the second level. They will also be transformed applying networking algorithm till complete convergence. Then H transformation of output sets, performed at the first level of transformations, elements of which are the elements of the first level, will be defined as the second level. While transformation of the second level set, the elements, that will form initial sets for the third level and up to the k-th level, the elements of which do not create a new set, are formed: all transformations are performed by clock periods $t_i, \; i = 1,2,3,\ldots$ In each clock period for any level, either the selection of elements from the sets by F* creation or Q*-transformations of sets according to previously selected elements occur, that shows the synchronism property of the given system.

To illustrate the notions of networking algorithm, graphical presentation of the network, the structure of which represents the totality of sets and elements is more convenient. In accordance with above-mentioned network graph contains three types of nodes. In Fig. 2 symbol of a rectangle ▢ – is an initial set, symbol of box □ – is an intermediate set (result of F*-transformation) and symbol of circle ○ – is an element. Oriented arcs connect sets and elements and some arcs are directed from the sets to elements, whereas others – from elements to sets. The arc, directed from the set $M_j^i(t_k)$ to element $a_j^i(t_{k+1})$ determines F* criterion of element selection, and the arc – from the element $a_j^i(t_{k+1})$ to the set $M_j^i(t_{k+2})$ indicates Q*-transformation of the set. Arcs are directed, thus it is oriented graph, zero set on this graph is denoted by symbol $\boxed{0}$.

It follows from Fig. 2 that there are elements such as $a_2^1(t_{s-1}); a_1^2(t_2); a_1^3(t_5)$ labeled by symbol ⊗ on the graph. Each of these elements does not belong to any set, because it is selected in the given clock period for its level as a single element, it does not take part in further processing of arrays and is its result. We will designate such elements as tail elements or elements forming the result. While Q*-transformation of sets, equal elements can be observed. Let us allocate information regarding all equal elements and their location in a set. For this purpose, for each Q*-transformation of the set, it is necessary to correspond binary code, where "1_S" are in those bits in which set positions equal elements are. All other positions of code, which correspond to other elements of the set, are filled with "0".

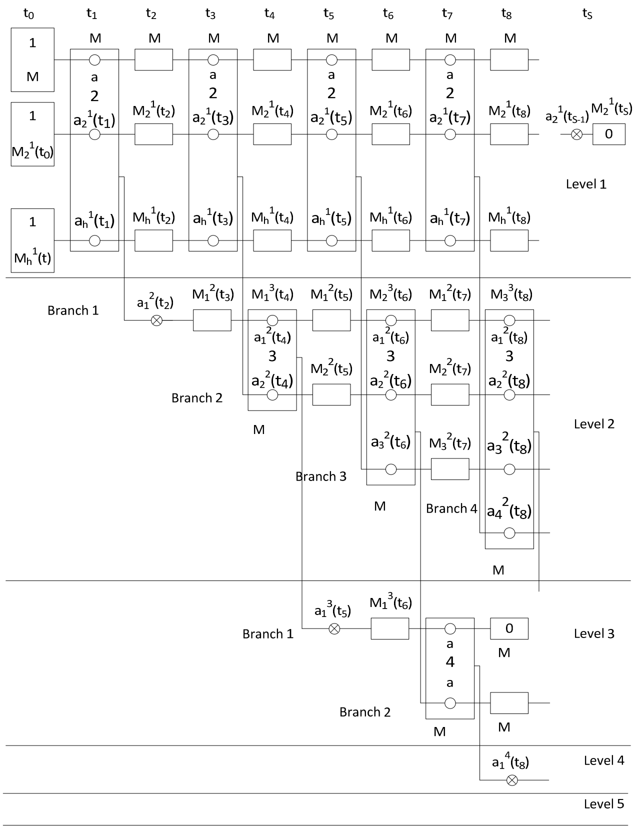


Figure 2. Graph-networks of PH-transformation

The term shadow mask (further simply mask) of Q*-transformation of the set means binary code, word length of which equals the capacity of the set, and "1_S" are located in those code digits, that correspond to location of labeled at the given step elements of a set. Such masks are formed for all intermediate and zero sets in all branches and levels.

III. Methods of Masks Presentation for Realization of Multilevel Transformation

Let us consider several methods of masks presentation and their properties while realization of PH-transformation, influencing its characteristics [15]. For restoration of initial information, transformed, in accordance with a particular technique of PH-transformation, while arrays processing it is necessary at each step of transformation t_i, to store at what positions in the array $A_j^v(t_i)$ (j – is the number of the array, v – is the number of level); elements, equal element $a_j^v(t_{i-1})$ are located. For this purpose we will form binary word, word length of which equals the dimensionality of the array $A_j^v(t_{i-2})$ and "ones" are in those positions of the code, in which positions of the array is the element, equals the selected one, is located. All other positions of binary code are filled with "zeros". This binary code, formed at each step of Q*-transformation of the array, will be called mask. $F_j^v(t_i)\left(F_j^v(t_i)\right)$ – is the mask of the array $A_j^v(t_i)$ by the element $a_j^v(t_{i-1})$.

Masks are formed in the process of array transformation till its complete convergence at all levels and for all branches. Masks are needed for decoding process and contain information regarding at what position (positions) in the array the selected element must be. As the masks are the result of array transformation, then the amount of masks for any array must not exceed the number of elements in the array:

$$f_j^v \le \omega_j^v, \quad (1)$$

where f_j^v – is the number of masks, formed while transformation of j-th array at v-th level; ω_j^v – is dimensionality of initial j-th array at v-th level.

Actually, the number of masks coincides with number of selected elements. Let m be dimensionality of the array, r – is the number of identical elements, l – is the number of groups with identical elements. Then the number of masks while transformation of such array equals:

$$f = m - r + l. \quad (2)$$

Masks of each array possess three properties:

1. Mask at any step of array transformation has not less than one "1"

$$F_j^v(t_i) \ne 0. \quad (3)$$

2. "1" in each digit of the mask in all masks while array processing comes only once:

$$F_j^v(t_i) \wedge F_j^v(t_k), \quad (4)$$

where $k \ne i,\ i = \{\omega, \omega+2, ..., z-2, z\},\ k = \{\omega, \omega+2, ..., z\}$.

3. Disjunction of all masks of the array, if the array does not contain zero elements, equals the code with "1" in all digits.

$$F_j^v(t_\omega) \vee F_j^v(t_{\omega+2}) \vee \ldots \vee F_j^v(t_{z-2}) \vee F_j^v(t_z) = 2^n - 1, \quad (5)$$

where n – is the word length of the mask.

The **first** method of masks presentation – is presentation of the masks by definition (3). Masks of the array represent binary words, dimensionality of which equals the dimensionality of the array. This method of masks formation can be used in any algorithms of PH-transformation. The drawback of this method is cumbersome way of masks presentation, but the algorithm of masks formation is rather simple. Due to the properties of masks formation (3), (4) and (5) there is a good opportunity to perform control while storage or transfer over communication lines of such masks. The property of masks (3) allows to reveal mistakes of 1→0 type, if the mask contains one "1" or group mistakes of 1→0 type, which lead to zeroing of the mask. The property of masks (4) allows to reveal single and group mistakes of 0→1 type. The property of masks (5) allows to reveal mistakes of 0→1 type. The only case that is not revealed as a mistake – is double mistake in identical digits of the masks, i.e., double mistakes of 1→0 and 0→1-types. To reveal malfunctions of 1→0 type it is sufficient to perform the operation of disjunction over all the masks of the array, and to analyze the result to verify if "1" are available in all the digits, i.e. to perform the operation of conjunction over all the digits of the result. Such control does not require considerable hardware expenditures and can be performed while array processing. Control of malfunctions of 0→1 type in accordance with (4) is time-consuming, and that is why is not efficient. However such malfunctions are easily detected while performing operation of summation by modulus 2 of all masks. If there are no malfunctions, then the result presents the known code of "1", i.e., unit-normal code or code 1 out of N [16].

$$F_j^v(t_\omega) \oplus F_j^v(t_{\omega+2}) \oplus \ldots \oplus F_j^v(t_{z-2}) \oplus F_j^v(t_z) = 2^n - 1. \quad (6)$$

Operation of summation by mod2 allows to detect malfunctions of 1→0, 0→1 types and combination 1→0, 0→1, if their total number in identical digits of masks is odd.

The **second** method of masks presentation – is stack method. This method of masks formation is the masks are initial addresses (number of position) of the subsets with identical elements or directly address of the element, the mask is formed for. This method allows to reduce the volume of masks presentation, but requires additional transformations while coding and decoding of the array. This method is called stack because while coding of the array forming addressed of the selected elements are written by stack principle, widely used in memory devices [17]. Decoding of the array with stack masks assumes step-by-step transformation of the array according to the following rule, upper address and values from this address is selected from the stack, and the array is filled with the elements, equal to the element with the given address. Filling of the information is carried out till the position, the address of which is selected from the stack at previous steps of decoding. If such address is missing, then the filling is carried out till the last element of the array. When the last address is selected from the stack, decoding process is completed.

The process of array filling can be controlled, analyzing the value of the next and replaced elements. If these values are equal then the next position is filled, otherwise, when previously filled elements in the array are single or represent single group.

The characteristic feature of stack mask structure is that their codes are not repeated. This property can be used to reveal errors and malfunctions while storage and transfer of the information. If while comparison of stack masks of one array identical masks are revealed, this testified that the mistake has been made.

The **third** method of masks presentation is based on optimization of the redundant first method of masks formation. The redundancy of mask presentation, capacity of which equals the dimensionality of the array, is that those digits of the masks, containing "1", in all other masks are filled with "0". Such digits from the next masks can be excluded; the third method of masks presentation is based on this concept. In this case, while array coding, each subsequent mask has dimensionality which is smaller than the previous one by the number of "1" in previous mask.

We will illustrate this statement by the example of array coding, consisting of eight elements (Fig. 3). In the given example, for masks storage the memory of 18 bits volume is required, where as for storage of complete masks we require 5x8=40 bits. The last mask may not be stored, as it consists of all "1". While decoding in the last but one mask instead of "1" the last but one selected element must be placed, and instead of "0" – last selected elements.

The given method of masks presentation also possesses good properties of error detection. At such mask presentation the word length of the next mask equals the quantity of "0" in previous mask, and the last mask consists of all "1". This property of masks allows to reveal all single and group errors of 1→0 or 0→1 type only multiple errors of 1→0 and 0→1 in one mask cannot be revealed. This method of masks representation was realized in section 3 of the given paper.

Element number	Array	Masks				
1	3	1	0	0	1	1
2	3	1	1	0	0	
3	10	0	0	1		
4	3	1	0			
5	3	1				
6	4	0				
7	12	0				
8	7	0				
Selected elements		3	4	7	10	12

Figure 3. The example of array coding

The **fourth** method is presentation of masks by logic-time code (LTC) [7], [18]. Here, each element is corresponded by its time section (quantum of time) – LTC. If while coding of the array element, equal to selected one, is met, then corresponding quantum of time is filled with the pulse, otherwise it is missing. Coding process of masks is carried out parallely for all various elements of the array.

Besides four basic methods of masks presentation specific cases are possible, these cases are defined by transformation algorithm of PH-transformation. While selecting elements from the array according to serial number, if we apply the third method of masks presentation, in the first digit all the masks "1" will always be. The result of array decomposition, according to such algorithm is shown in Fig. 4a. Array, shown in Fig. 3, is taken at initial.

Element number	Array	Masks				
1	3	1	1	1	1	1
2	3	1	0	0	0	
3	10	0	0	0		
4	3	1	0			
5	3	1				
6	4	0				
7	12	0				
8	7	0				
Selected elements		3	4	7	10	12

a)

Element number	Array	Masks				
1	3	1				
2	3	1				
3	10	0				
4	3	1				
5	3	1				
6	4	0				
7	12	0				
8	7	0				
Selected elements		3	10	4	12	7

b)

Figure 4. Decomposition of the array, based on the selection of elements according its serial number

It is shown in Fig. 4a, that the first digits of masks contain "1". If the selected elements are single in the array, then in all other digits masks are "0". Such data contain little information, and it may not be used while coding. Fig. 4b shows the result of array decomposition, taking into account the above-mentioned; as a result, masks of such array are reduced to one eight-digit mask.

For transformation algorithm of the array, the situation is possible, when all the array consists of identical elements. It means that the mask of such array is only one and contains "1" in all digits. Such mask may not be stored, only dimensionality of the array and the element, this array consists of, should be stored. While the decoding of the array it is necessary to take into account, that if the mask is missing, and then the array consists of identical elements.

IV. Analysis of Computational Efficiency of Multilevel Parallel-Hierarchical Transformation

Let us perform comparative analysis of the efficiency by the number of the operations, used for multilevel parallel-hierarchical transformation. Let there be k arrays. We denote by n the number of the elements in the array number i (it is obvious, that $n_i \geq 1$), $i = \overline{1,k}$, $\sum_{i=1}^{k} n_i = N$ – is total number of the elements, being processed; m_i – is the number of various elements in i-th array (it is obvious, that $m_i \geq 1$), $\sum_{i=1}^{k} m_i = M$ – is total number of various elements, being processed; p_i^j – is the probability of j-th element emergence in i-th array, where $j = \overline{1, m_i}$, and $m = \max\limits_{i=\overline{1,k}} m_i$.

To obtain the first set of the elements, branch of parallel-hierarchical transformation, comparison operation must be performed N times, the operation of general part selection must be performed $\sum_{i=1}^{k} n_i p_i^1$ times.

That is why, for processing $N_1 = \sum_{i=1}^{k} n_i \left(1 - p_i^1\right)$ elements remain.

To obtain the second set of the elements, branch of parallel-hierarchical transformation, comparison operation must be performed N_1 times, the operation of general part selection must be performed $\sum_{i=1}^{k} n_i \left(1 - p_i^1\right) p_i^2$ times.

That is why, for processing $N_2 = \sum_{i=1}^{k} n_i \left(1 - p_i^1\right)\left(1 - p_i^2\right)$ elements remain.

To obtain m-th set of the elements of the branch, it is necessary to perform comparison operation $N_{m-1} = \sum_{i=1}^{k} \left[n_i \prod_{j=1}^{m-1} \left(1 - p_i^j\right) \right]$ times and selection operation – $\sum_{i=1}^{k} \left[n_i \prod_{j=1}^{m-1} \left(1 - p_i^j\right) p_i^m \right]$ times.

Thus, the number of operation will be:

$$N + \sum_{i=1}^{k} n_i p_i^1 + \sum_{i=1}^{k} n_i \left(1 - p_i^1\right) + \sum_{i=1}^{k} n_i \left(1 - p_i^1\right) p_i^2 +$$

$$+ \sum_{i=1}^{k} n_i \left(1 - p_i^1\right)\left(1 - p_i^2\right) + \cdots + \sum_{i=1}^{k} \left[n_i \prod_{j=1}^{m-1} \left(1 - p_i^j\right) p_i^m \right] +$$

$$+ \sum_{i=1}^{k} \left[n_i \prod_{j=1}^{m-1} \left(1 - p_i^j\right) \right] = N +$$

$$+ \sum_{i=1}^{k} n_i \ (p_i^1 + 1 - p_i^1 \ + \left(1 - p_i^1\right) p_i^2 \ + \left(1 - p_i^1\right)\left(1 - p_i^2\right) + \ \ldots \ +$$

$$+ \prod_{j=1}^{m-1} \left(1 - p_i^j\right) p_i^m \ + \prod_{j=1}^{m-1} \left(1 - p_i^j\right)\Big) \ =$$

$$= N + \sum_{i=1}^{k}[n_i\,(1+(1-p_i^1)+(1-p_i^1)(1-p_i^2)+\ldots+$$

$$+\prod_{j=1}^{m-1}(1-p_i^j)] = N+\sum_{i=1}^{k}\left[n_i\left(1+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right)\right]. \quad (7)$$

Let us prove the formula (7) applying the method of mathematical induction.

Let the number of sets k be equal 1, then the number of elements N will be equal n and

$$n+\sum_{i=1}^{1}\left[n_i\left(1+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right)\right]=$$

$$=2n+n\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)=n\left(2+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right).$$

Let the number of sets k be equal 2, then

$$n_1+n_2+\sum_{i=1}^{1}\left[n_i\left(1+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right)\right]=2n_1+2n_2+$$

$$+n_1\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)+n_2\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)=2(n_1+n_2)+$$

$$+(n_1+n_2)\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)=(n_1+n_2)\left(2+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right).$$

From assumption, that formula (7) is valid for k sets, we will prove; that it is valid for $k+1$ sets.

$$N+\sum_{i=1}^{k}\left[n_i\left(1+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right)\right]+n_{k+1}\left(2+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right)=$$

$$=n_1+n_2+\ldots+n_k+\sum_{i=1}^{k}\left[n_i\left(1+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right)\right]+n_{k+1}+$$

$$+n_{k+1}\left(1+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right)=n_1+n_2+\ldots+n_{k+1}+$$

$$+\sum_{i=1}^{k+1}\left[n_i\left(1+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right)\right]=N+\sum_{i=1}^{k+1}\left[n_i\left(1+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right)\right],$$

where $N=\sum_{i=1}^{k+1}n_i$. Thus, the formula of the form (7) is obtained, that was to be proved.

We will compare the efficiency of parallel-hierarchical transformation and already known transformations, for instance, FFT, Hadamard transform and Haar transform [1], [19] – [22], by the number of operations, used in them.

Let the emergence of the elements in the array will be the event equiprobable with the probability p, where as for N elements being processed from k of n-elements sets. It is valid that:

$$N+\sum_{i=1}^{k}\left[n_i\left(1+\sum_{z=1}^{m-1}\prod_{j=1}^{z}(1-p_i^j)\right)\right]=N+\sum_{i=1}^{k}\left[n_i\left(1+\sum_{z=1}^{m-1}(1-p)^z\right)\right]=$$

$$=N+\left(1+\sum_{z=1}^{m-1}(1-p)^z\right)\sum_{i=1}^{k}n_i=N+N\left(1+\sum_{z=1}^{m-1}(1-p)^z\right)=$$

$$=N\left(2+\sum_{z=1}^{m-1}(1-p)^z\right)=N\left(2+\frac{1-p}{p}\right)=\left(1+\frac{1}{p}\right)N,$$

if $n=m$, then $p=1/N$.

Hence, the number of operation for parallel-hierarchical transformation equals $N(N+1)$. For comparison, it is should be noted, that the number of operations used for widely applied in practice transformations, for instance, orthogonal transformations, will be: for FFT – $4N^2\log_2 N$, for Hadamard transform – $2N^2\log_2 N^2$, and for Haar transform – $4N(N+1)$.

The lack of labor consuming operations of multiplications and division prove that computations algorithm, performing parallel-hierarchical transformation is rather simple, that makes it an efficient method to be used in various applied sphere where combination of high degree of parallelism and compact form of data presentation is required.

V. The Results of Experimental Research of the Method of Multilevel Transformation of Masked Signals

Software complex intended for realization of direct and reverse PH-transformation of information environments, containing two separate software products was developed [23] - [25]:

1. Software, intended for realization of direct PH-transformation with optimization of masks formation procedure. The characteristic feature of the realized algorithm is application of masks formation that reduces the volume of the memory, required for their storage as compared with non-optimized algorithm, as well as application of minimum element multiplication by the capacity in G-transformation operator that is one of the stages of PH-transformation (Fig. 5).

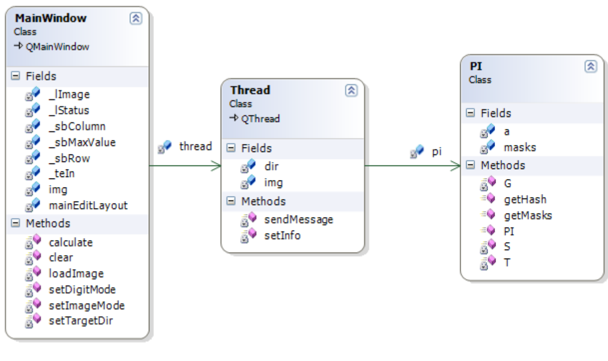


Figure 5. Class diagram of software library for realization of direct PH-transformation with optimization of masks formation procedure

2. Software, intended for realization of reverse PH-transformation, based on optimized mask method for restoration of transformed (applying the method of direct PH-transformation with optimization of masks formation procedure) information environments (in the form of 2D matrix of data or image). The characteristic feature of the considered algorithm is modification of decoding process on the basis of optimized algorithm of masks formation that increases the decoding rate (Fig. 6).

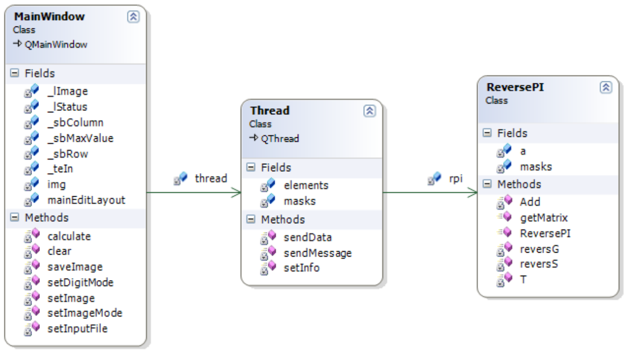


Figure 6. Class diagram of software library for realization of reverse PH-transformation, based on optimized mask method

Programming language of software complex realization is "C++". Functions of software library, after recompilation, operate correctly with various operation systems: MS Windows, GNU/Linux, Mac OS.

Realization of direct PH-transformation with optimization of masks formation procedure comprises such basic stages:

1. Loading of information array in the form of image or 2D matrix of data of the dimensionality, set by user.

2. Realization of direct PH-transformation with optimization of masks formation procedure while data / image processing (PH method in Fig. 5) – serial application of three operators $\Phi(M)=T\left[S\left(G\left(M\right)\right)\right]$:

2.1. Transposition (T method in Fig. 5);

2.2. G-transformation (T method in Fig. 5);

2.3. Shift (S method in Fig. 5);

and storage of one-dimensional matrix of tail elements of the transformed information array (getHash method in Fig. 5).

3. Formation of one-dimensional matrix of optimized masks (getMasks method in Fig. 5).

4. Introduction of file-protocol.

Realization of reverse PH-transformation, based on optimized mask method comprises such basic stages:

1. Loading of the array of tail elements in the form of one-dimensional data set, obtained while coding, applying the method of direct PH-transformation with optimization of masks formation procedure;

2. Loading of optimized masks array in the form of one-dimensional data set (0 and 1), obtained while coding applying the method of direct PH-transformation with optimization of masks formation procedure;

3. Realization of reverse PH-transformation, based on optimized mask method over data (ReversePI method in Fig. 6);

4. Restoration of reverse data array in the form of 2D matrix of data or image (getMatrix method in Fig. 6).

The developed software enables to load images or set matrix dimensions for direct and reverse PH-transformation and fill in the set matrix personally or by means of random-number generator. After that PH-transformation is carried out for the given data array, tail elements values their sum and the sum of input matrix (the given sums, according to the theory of PH-transformation must coincide), as well as time of processing and generalized dimension of masks are deducted.

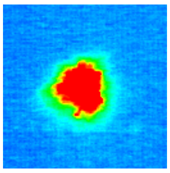
The suggested software complex for realization of direct and reverse PH-transformation of information environments was tested using various data set in the form of 2D data matrix of different dimensionality, as well as spot images of laser beam profile of various dimensionality.

In the given test example realization of direct PH-transformation with optimization of masks while processing of color spot image of laser beam profile was performed (Fig. 7), in RGB format, dimensionality 128x128 pixels. As well as reverse PH-transformation, based on optimized mask method for its restoration was performed. The results of experimental research, carried out (and test example too) demonstrate the increase of the efficiency of PH-transformation (in the context of coding / decoding information and images) by criteria:

1) direct PH-transformation processing rate: 984 msec – for non-optimized (Fig. 8, column "Time") and 782 msec – for optimized method (Fig. 9, column "Time");

2) reverse PH-transformation processing rate: 1250 msec – for non-optimized (Fig. 10, column "Time") and 422 msec – for optimized method (Fig. 11, column "Time");

3) reduction of memory volume redundancy while formation of masks array: 25057800 bits – for non-optimized (Fig. 8, column "Mask size") and 10187072 bits – for optimized (Fig. 9, column "Mask size").



The results of software complex operation coincided with the results of mathematics and computer-based modeling that proves of correctness and reliability of software complex operation.

Figure 7. Selected image of laser beam profile for test example

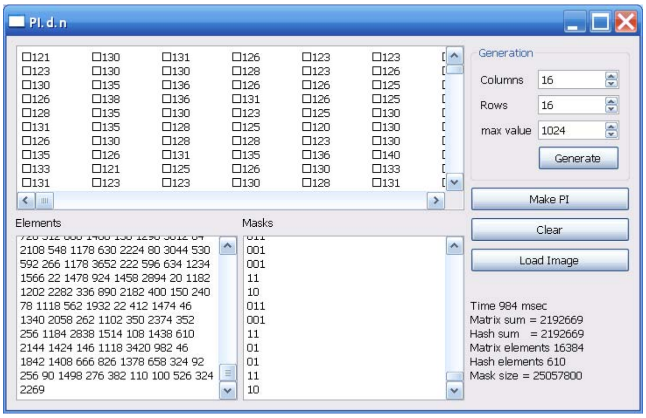


Figure 8. Direct PH-transformation without optimization of masks formation

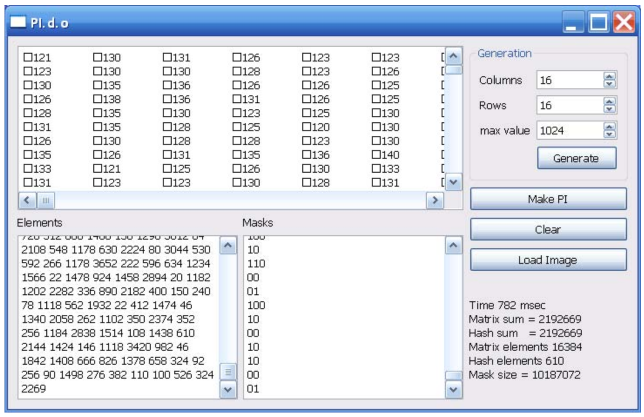


Figure 9. Direct PH-transformation with optimization of masks formation

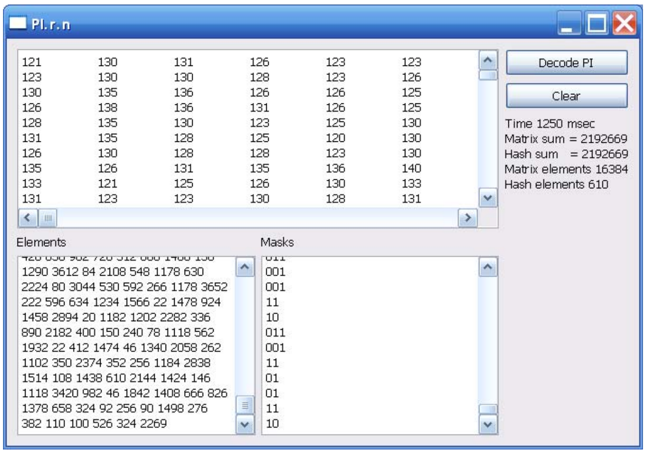


Figure 10. Reverse PH-transformation without masks optimization

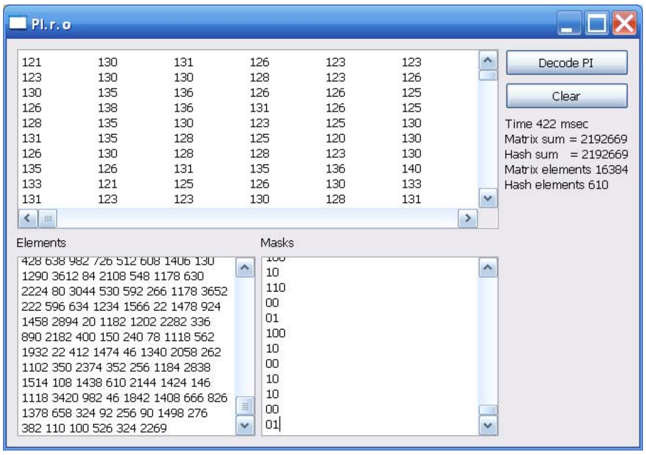


Figure 11. Reverse PH-transformation with masks optimization

VI. Conclusion

The paper considered basic provisions of organization of computational processes with pyramidal processing and PH-transformation of information, the analysis of peculiarities of structural-functional organization of networking model of parallel-hierarchical transformation of information environments, as well as corresponding methods of masks formation for realization of multilevel transformation has been carried out.

In the process of research main goal has been achieved. The efficiency of direct and reverse PH-transformation as a result of images transformation rate has been increased and optimization of the redundancy of masks representation has been achieved, dimensionality of which equaled dimensionality of data array being processed. The results obtained allow realizing real time preprocessing and recognition of images of large dimensionality with simultaneous compression. It increases the efficiency of recognition systems functioning, in particular, laser beam profiling systems [25].

The paper contains the preliminary results of simulation modeling and program emulation of the suggested method of masks presentation for optimization of reverse PH-transformation of images, the set of demonstration applications is created, they provide the possibility to evaluate the reliability of program product operation and show the efficiency of its usage in applied tasks of coding processing and comparison of images.

The results of the research can find application in experimental study of functioning not only PH intelligent computation systems but for further research in the sphere of development of highly productive hierarch-hierarchical networks based on optoelectronic element base.

References

[1] S. Engelberg, "Digital Signal Processing. An Experimental Approach", Springer, 2008.
[2] S.S. Bhattacharyya, E.F. Deprettere, R. Leupers, J. Takala, "Handbook of Signal Processing Systems", Springer, 2010.
[3] U. Prett, "Digital image processing", Mir, Moscow, 1982.
[4] U. Grenader , "Lections of theory of patterns. Patterns analysis" – Mir, Moscow, 1981.
[5] B. Sklyar, "Digital communication. Theory and practical applications", Publishing house "Wilyams", Moscow, 2003.
[6] A. Sergienko, "Digital signal processing", Publishing house "Piter", St. Petersburg, 2006.
[7] V. Kozhemyako, Yu. Kutaev, S. Svechnikov, L. Timchenko, A. Yarovyy, "Parallel-hierarchical model of systemic transformation as optoelectronic means of artificial intelligence", Univ-Vinnytsia, 2003.
[8] A.A. Yarovyy, "Applied Realization of Neural Network and Neurolike Parallel-Hierarchical System Based on GPGPU", Proceedings of the 10th International Conference on DAS-2010, Universitatea Stefan cel Mare Suceava, Romania, pp. 351-356, 2010.
[9] V. Kozhemyako, L. Timchenko, A. Yarovyy, "Methodological Principles of Pyramidal and Parallel-Hierarchical Image Processing on the Base of Neural-Like Network Systems", Advances in Electrical and Computer Engineering, Vol. 8 (15), no 2 (30), pp. 54-60, 2008.
[10] V.P. Kozhemyako, L.I. Timchenko, A.A. Yarovyy, "Software Support of Accurately Measurement and Prediction of Laser Beam Profile Characteristics", Proceedings of the X International Conference „Swiatiowody i ich zastosowania", Vol. 2, Wydawnictwo-Drukarnia Liber Duo s.c., Poland, pp. 675-684, 2006.
[11] L. Timchenko, Yu. Kutaev, V. Kozhemyako, A. Gertsiy, A. Yarovyy, N. Kokryatskaya, "Method for Processing of Extended Laser Paths Images", Advances in Electrical and Computer Engineering, Vol. 3 (10), no 2 (20), pp. 66-78, 2003.
[12] D. Knut, "Art of programming for computer", Mir, Moscow, 1978.
[13] B. Chazelle "Computing on a Free Tree via Complexity-Preserving Mappings", Algorithmica 2, Springer-Verlag, pp. 337-361, 1987.
[14] L. Timchenko, Yu. Kutaev, V. Kozhemyako, A. Yarovyy, A. Gertsiy, A. Terenchuk, N. Shweiki, "Method for Training of a Parallel-Hierarchical Network, Based on Population Coding for Processing of Extended Laser Paths Images", Proceedings of SPIE, Vol. 4790, pp. 465-479, 2002.
[15] Yarovyy A. "Method of optimization of masks formation procedure while information coding by parallel-hierarchical transformation", Sc. journal "Visnyk of Vinnytsia Polytechnical Institute", no 6 (99). – pp. 216-223, 2011.
[16] Bryukhovich E., "Automatic control and performance of computer systems", Sc. journal "Control systems and machines", no 4, 1979.
[17] E. Metlitskiy, V. Kaverznev, V. Timokhin, "The systems of parallel memory: theory, development, application", St. Petersburg University Press, 1989.
[18] V. Kozhemyako, L. Timchenko, A. Yarovyy, "The models of parallel-hierarchical information processing based of logic-time function apparatus", Abstracts of International Sc. Conf. "Information and computer technologies, modeling, control", Georgian Tech. Univ. Press, pp. 231-232, 2010.
[19] N. Ahmed, K.R. Rao, "Orthogonal Transforms for Digital Signal Processing", Springer-Verlag, Berlin, 1975.
[20] R.S. Stankovic, M.S. Stankovic, K. Egiazarian, L.P. Yaroslavsky, "Remarks on history of FFT and related algorithms", In: Astola, J. et al. (eds). Proceedings of the 2004 Int. TICSP Workshop on Spectral Methods and Multirate Signal Processing, Austria, pp. 281-288, 2004.
[21] K.J. Horadam, "A Generalised Hadamard Transform", Proc. 2005 IEEE ISIT, Australia, pp. 1006–1008, 2005.
[22] A.M. Grigoryan, "Brief notes in advanced DSP", CRC Press Taylor and Francis Group, 2009.
[23] A. Yarovyy, I. Sugak, Author's certificate of registration of copyright on software product № 39490. "Computer program of direct PH transformation with optimization of masks formation", 2011.
[24] A. Yarovyy, I. Sugak, Author's certificate of registration of copyright on software product № 39489. "Computer program of reverse PH transformation, based on optimized mask method", 2011.
[25] A. Yarovyy, L. Timchenko, N. Kokriatskaia, "Theoretical Aspects of Parallel-Hierarchical Multi-Level Transformation of Digital Signals", Proceedings of the 11th International Conference on DAS-2012, Universitatea Stefan cel Mare Suceava, Romania, pp. 1-9, 2012.