Journal of Mathematical Imaging and Vision 16: 269–279, 2002

Tree-Structured Haar Transforms

KAREN EGIAZARIAN & JAAKKO ASTOLA
Signal Processing Laboratory, Tampere University of Technology, P. O. Box 553, FIN-33101, Tampere, Finland
karen@cs.tut.fi
jta@cs.tut.fi

Abstract. The Haar transform is generalized to the case of an arbitrary time and scale splitting. To any binary tree we associate an orthogonal system of Haar-type functions - tree-structured Haar (TSH) functions. Unified fast algorithm for computation of the introduced tree-structured Haar transforms is presented. It requires $2(N-1)$ additions and $3N-2$ multiplications, where N is transform order or, equivalently, the number of leaves of the binary tree.

Keywords: Haar functions, Haar transform, fast algorithms, Walsh transform, binary tree, Fibonacci tree, wavelet packets

1. Introduction

The most common ways to decompose a signal into more elementary signals are the Fourier expansion and its many generalizations. The signal is represented as a linear combination of, usually orthogonal, basis functions of the linear space where the signal belongs to. The trigonometric functions are natural basis functions when the signal contains harmonic oscillations as is the case for many natural signals – sound signals and electromagnetic waves being prominent examples. For "nonharmonic" signals rectangular basis functions may be natural. Examples of such systems are the the well-known Haar, Rademacher and Walsh systems of functions, which have found a lot of applications in communication theory and signal processing since 1960s [2, 9, 13].

Historically, the first set of orthogonal rectangular functions is known nowadays as the Haar functions was described by the Hungarian Mathematician A. Haar in 1910 [9]. The Haar functions take essentially just two nonzero values, but still provide an expansion of a continuous function. The specific property of the Haar functions that could not be obtained by any other non-sinusoidal orthogonal functions at that time, is the property of uniform and rapid convergence to a given continuous function. Discretization of the Haar functions gives the set of orthogonal discrete Haar functions that forms the Haar transform matrix [3].

The classical Haar functions are defined by the dyadic splitting of time interval. In this paper we will extend the concept of Haar functions to functions having an arbitrary time splitting.

In Section 2 the concept of binary interval splitting tree is defined. Based on this tree structure, in Section 3, we define a system of orthogonal functions - tree-structured Haar (TSH) functions. The class of TSH functions contains as special cases the following systems of orthogonal functions: classical system of Haar functions when the underlying tree is a complete full binary tree, the system of canonical Haar functions [10], known also as the Fibonacci system by Agaian-Aizenberg-Alaverdian [1] when the underlying tree is the binary logarithmic tree, the system of generalized Fibonacci-Haar functions [6] if the underlying tree is the generalized Fibonacci tree, etc.

In Section 4 the discrete TSH functions and matrices, are defined. Some special examples of TSH matrices are considered. Fast algorithms to compute discrete TSH transforms are developed in Section 5. Various extensions of TSH transforms containing, as particular cases, the discrete Walsh-Hadamard transforms, as well as the Haar wavelet packet transforms are discussed in Section 6.

2. Interval Splitting Trees

A rooted tree is called *binary* if each node has outdegree at most two. The length of the path from the root to a node is called the *depth* of that node. A non-leaf (non-terminal) node of the binary tree is a *splitting node* if it has outdegree two. A binary tree whose all non-leaf nodes are splitting nodes is called *full*. If all the leaves have same depth the tree is called complete. If there is a path with origin a and end b, we say that a is a *predecessor* of b and that b is a *successor* of a. If $\text{depth}(b) = \text{depth}(a) + 1$ we say that b is an *immediate successor* or *child* of a and that a is an *immediate predecessor* or *parent* of b.

Starting from the root of the tree, we label each edge of the tree as follows.

1) If the node has two children, the left outedge will have label 0 and the right outedge will have label 1,

2) if the node has only one child, the outedge will have label 2.

Each node of the binary tree will be indexed by a ternary vector[1] $(\alpha_1(a), \ldots, \alpha_k(a))$ $(\alpha_j \in \{0, 1, 2\}, j = 1, \ldots, k)$ of length k, where k is the depth of this node and α_j are labels of the edges to that node starting from the root of the tree.

Figure 1 shows an example of a binary tree with indexed arcs and nodes.

The nodes that have depth equal to j form the jth *level* of the tree. The index vector of a node on level j has j components.

Before introducing the notion of binary interval splitting tree, we will label all the nodes of a binary tree in the following way: each node $\vec{\alpha} = (\alpha_1, \ldots, \alpha_k)$ is labeled by the number $\nu(\vec{\alpha})$ of leaves that are successors of this node.

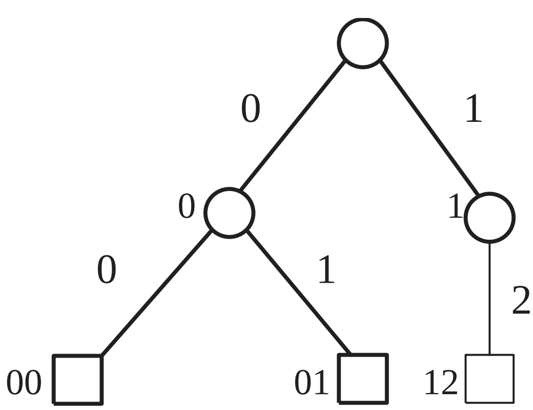

Figure 1. Binary tree with labeled edges and indexed nodes.

Definition 1. A *binary interval splitting tree (BIST)* of the depth n is a binary tree that has intervals assigned to each of non-leaf nodes as follows:

1. $I_{root} = (0, 1)$, $I_0 = I_{(root,0)}$, $I_1 = I_{(root,1)}$, where I_0 and I_1 are the left and the right sub-intervals of I_{root} corresponding to two children of the root.

2. Let $(\alpha_1, \ldots, \alpha_j)$ be a splitting node and $I_{\alpha_1,\ldots,\alpha_j} = (a, b)$, $0 \leq a < b < 1$, then for $j = 1, \ldots, n-2$,

$$I_{(\alpha_1,\ldots,\alpha_j,0)} = \left(a, a + \frac{\nu_{(\alpha_1,\ldots,\alpha_j,0)}}{\nu_{(\alpha_1,\ldots,\alpha_j)}}(b-a)\right)$$

and

$$I_{(\alpha_1,\ldots,\alpha_j,1)} = \left(a + \frac{\nu_{(\alpha_1,\ldots,\alpha_j,0)}}{\nu_{(\alpha_1,\ldots,\alpha_j)}}(b-a), b\right)$$

3. Let $(\alpha_1, \ldots, \alpha_j)$ be a non-splitting node. Then

$$I_{(\alpha_1,\ldots,\alpha_j,2)} = I_{(\alpha_1,\ldots,\alpha_j)}.$$

Denote by $|I_{\alpha_1,\ldots,\alpha_j}| = \nu_{(\alpha_1,\ldots,\alpha_j)}$ the cardinality of the interval $I_{(\alpha_1,\ldots,\alpha_j)} \subseteq I_{root}$, assigned to the node $\vec{\alpha} = (\alpha_1, \ldots, \alpha_j)$, $j = 1, \ldots, n-1$. Notice that

$$I_{\alpha_1,\ldots,\alpha_j,0} \bigcup I_{\alpha_1,\ldots,\alpha_j,1} = I_{(\alpha_1,\ldots,\alpha_j)},\ I_{\alpha_1,\ldots,\alpha_j,0} \bigcap I_{\alpha_1,\ldots,\alpha_j,1} = \emptyset,$$
$$\frac{|I_{\alpha_1,\ldots,\alpha_j}|}{|I_{\alpha_1,\ldots,\alpha_j,0}|} = \frac{\nu_{(\alpha_1,\ldots,\alpha_j)}}{\nu_{(\alpha_1,\ldots,\alpha_j,0)}},\quad \frac{|I_{\alpha_1,\ldots,\alpha_j}|}{|I_{\alpha_1,\ldots,\alpha_j,1}|} = \frac{\nu_{(\alpha_1,\ldots,\alpha_j)}}{\nu_{(\alpha_1,\ldots,\alpha_j,1)}},$$
$$\nu_{(\alpha_1,\ldots,\alpha_j)} = \nu_{(\alpha_1,\ldots,\alpha_j,0)} + \nu_{(\alpha_1,\ldots,\alpha_j,1)}.$$

In other words, $I_{\alpha_1,\ldots,\alpha_j,0}$ and $I_{\alpha_1,\ldots,\alpha_j,1}$ split the interval $I_{\alpha_1,\ldots,\alpha_j}$ into two non-intersecting sub-intervals in the proportion of the numbers of leaves that are successors of these nodes.

In Figure 2, the binary tree from Figure 1 with labeled nodes and corresponding intervals are shown.

3. Tree-structured Haar Functions

Let T be a binary interval splitting tree with N leaves and of depth n. An example of such a tree is in Figure 2 where $N = 3$ and $n = 2$.

Let us define the set of basis functions corresponding to the tree T by the following procedure: to the root of the tree we associate two basis functions $H_{root,0}$ and $H_{root,1}$ (playing a similar role as the scaling

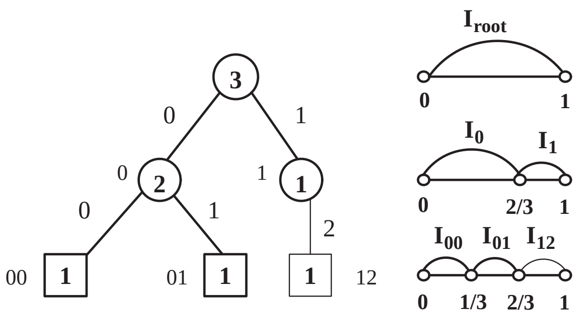

Figure 2. Binary tree of Figure 1 with the corresponding splitting of the interval.

function and the wavelet function in wavelet theory [12]):

$$H_{root,0}(t) = \frac{1}{\sqrt{N}},\ t \in (0,1). \tag{1}$$

$$H_{root,1}(t) = \begin{cases} \sqrt{\frac{\nu_{(1)}}{N\nu_{(0)}}}, & \text{if } t \in I_0, \\ -\sqrt{\frac{\nu_{(0)}}{N\nu_{(1)}}}, & \text{if } t \in I_1, \end{cases} \tag{2}$$

where $\nu_{(0)}$ and $\nu_{(1)}$ are the labels of the left and the right children of the root of the tree, respectively;

and to each non-root **splitting** node of T with index $(\alpha_1, \dots, \alpha_k)$ we associate the basis function $H_{(\alpha_1,\dots,\alpha_k)}(t), t \in (0,1)$, $k = 1, \dots, n-1$, defined by:

$$H_{(\alpha_1,\dots,\alpha_k)}(t) = \begin{cases} \sqrt{\frac{\nu_{(\alpha_1,\dots,\alpha_k,1)}}{\nu_{(\alpha_1,\dots,\alpha_k)}\nu_{(\alpha_1,\dots,\alpha_k,0)}}}, & \text{if } t \in I_{\alpha_1,\dots,\alpha_k,0}, \\ -\sqrt{\frac{\nu_{(\alpha_1,\dots,\alpha_k,0)}}{\nu_{(\alpha_1,\dots,\alpha_k)}\nu_{(\alpha_1,\dots,\alpha_k,1)}}}, & \text{if } t \in I_{\alpha_1,\dots,\alpha_k,1}, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Definition 2. The set of functions $H_{root,0}(t)$, $H_{root,1}(t)$ and $H_{(\alpha_1,\dots,\alpha_k)}(t)$, defined by (1) - (3), for all splitting nodes of a binary interval splitting tree T, is called the set of *tree-structured Haar (TSH) functions* of T.

The set of TSH functions of the binary interval splitting tree of Figure 2 is shown in Figure 3.

Theorem 1. *The set of tree-structured Haar functions defined by (1) - (3), form a set of orthogonal functions.*

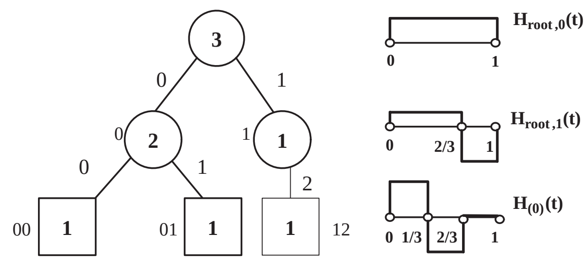

Figure 3. The tree-structured Haar functions corresponding the tree from the Figure 2.

Proof: First, one can easily check that eq. (2) and (3) will be of the form

$$H_{root,1} = \begin{cases} \sqrt{\frac{1}{N}}, & \text{if } t \in I_0, \\ -\sqrt{\frac{1}{N}}, & \text{if } t \in I_1, \end{cases} \tag{4}$$

$$H_{(\alpha_1,\dots,\alpha_k)}(t) = \begin{cases} \sqrt{2^{-(n-k)}}, & \text{if } t \in I_{\alpha_1,\dots,\alpha_k,0}, \\ -\sqrt{2^{-(n-k)}}, & \text{if } t \in I_{\alpha_1,\dots,\alpha_k,1}, \\ & k = 1, \dots, n-1, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

if the tree is a complete and full one, since $N = 2^n$ and $\nu_{(\alpha_1,\dots,\alpha_k)} = 2\nu_{(\alpha_1,\dots,\alpha_k,0)} = 2\nu_{(\alpha_1,\dots,\alpha_k,1)} = 2^{n-k}$.

Thus, the classical system of the Haar functions is a particular case of the general tree-structured Haar functions.

Now let us show that the general functions are orthogonal. Thus, it is enough to show that

$$\begin{aligned} S &= \int_0^1 H_{(\alpha_1,\dots,\alpha_j)}(t) H_{(\beta_1,\dots,\beta_k)}(t)dt \\ &= \begin{cases} \frac{|I_{\alpha_1,\dots,\alpha_k}|}{\nu_{(\alpha_1,\dots,\alpha_k)}} \neq 0 & \text{j = k}, \alpha_i \equiv \beta_i \text{ for all } i \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{6}$$

for j=1,...,n-1.

We can assume that $j \leq k$ and let us consider separately the following three cases:

1. Let $j = k$ and $\alpha_i = \beta_i$ for $i = 1, \dots, j$. Then

$$S = |I_{\alpha_1,\dots,\alpha_k,0}| \frac{\nu_{(\alpha_1,\dots,\alpha_k,1)}}{\nu_{(\alpha_1,\dots,\alpha_k)}\nu_{(\alpha_1,\dots,\alpha_k,0)}} + |I_{\alpha_1,\dots,\alpha_k,1}| \frac{\nu_{(\alpha_1,\dots,\alpha_k,0)}}{\nu_{(\alpha_1,\dots,\alpha_k)}\nu_{(\alpha_1,\dots,\alpha_k,1)}}$$

Since from (1) it follows that

$$\frac{|I_{\alpha_1,\ldots,\alpha_k,0}|}{|I_{\alpha_1,\ldots,\alpha_k,1}|} = \frac{\nu_{(\alpha_1,\ldots,\alpha_k,0)}}{\nu_{(\alpha_1,\ldots,\alpha_k,1)}},$$

and

$$|I_{\alpha_1,\ldots,\alpha_k,0}| + |I_{\alpha_1,\ldots,\alpha_k,1}| = |I_{\alpha_1,\ldots,\alpha_k}|,$$

we have

$$S = \frac{|I_{\alpha_1,\ldots,\alpha_k,1}|}{\nu_{(\alpha_1,\ldots,\alpha_k)}} + \frac{|I_{\alpha_1,\ldots,\alpha_k,0}|}{\nu_{(\alpha_1,\ldots,\alpha_k)}} = \frac{|I_{\alpha_1,\ldots,\alpha_k}|}{\nu_{(\alpha_1,\ldots,\alpha_k)}}.$$

2. Let $j = k$ and let there be $i, i \in \{1, \ldots, j\}$ such that $\alpha_i \neq \beta_i$. Then, by the construction of the interval splitting tree (Definition 1.), $I_{(\alpha_1,\ldots,\alpha_j)} \cap I_{(\beta_1,\ldots,\beta_k)} = \emptyset$, therefore $S = 0$.

3. Let, finally, $j \neq k$ (without loss in generality we suppose that $j < k$) and $\alpha_i = \beta_i$ for $i = 1, \ldots, j$. By the construction of the intervals and the functions H

$$S = \sqrt{\frac{\nu_{(\alpha_1,\ldots,\alpha_j,1)}}{\nu_{(\alpha_1,\ldots,\alpha_j)}\nu_{(\alpha_1,\ldots,\alpha_j,0)}}} \Bigg(|I_{\beta_1,\ldots,\beta_k,0}| \sqrt{\frac{\nu_{(\beta_1,\ldots,\beta_k,1)}}{\nu_{(\beta_1,\ldots,\beta_k,0)}\nu_{(\beta_1,\ldots,\beta_k)}}} - |I_{\beta_1,\ldots,\beta_k,1}| \sqrt{\frac{\nu_{(\beta_1,\ldots,\beta_k,0)}}{\nu_{(\beta_1,\ldots,\beta_k,1)}\nu_{(\beta_1,\ldots,\beta_k)}}} \Bigg)$$

and utilizing (1), we get

$$S = \sqrt{\frac{\nu_{(\alpha_1,\ldots,\alpha_j,1)}}{\nu_{(\alpha_1,\ldots,\alpha_j)}\nu_{(\alpha_1,\ldots,\alpha_j,0)}\nu_{(\beta_1,\ldots,\beta_k)}}} \Bigg(|I_{\beta_1,\ldots,\beta_k,0}| \sqrt{\frac{I_{\beta_1,\ldots,\beta_k,1}}{I_{\beta_1,\ldots,\beta_k,0}}} - |I_{\beta_1,\ldots,\beta_k,1}| \sqrt{\frac{I_{\beta_1,\ldots,\beta_k,0}}{I_{\beta_1,\ldots,\beta_k,1}}} \Bigg) = 0.$$

Thus, Theorem 1. is proved.

Remark 1. Tree-structured Haar functions coincide with the classical Haar functions in the case of a complete full binary tree.

4. Discrete tree-structured Haar functions and matrices: case studies

Definition 3. Let T be a binary tree with N leaves. The *Discrete tree-structured Haar functions* of T are defined by sampling the tree-structured Haar functions of T at points j/N, $j = 0, 1, \ldots, N-1$. The $N \times N$ orthogonal matrix whose rows are these functions is a *tree-structured Haar* (*TSH*) *matrix* (of tree T).

In the following we discuss some properties of TSH matrices that follow directly from their construction.

Property 1. Let T be a binary tree of depth n and with N leaves. Then

1. The number of rows of a TSH matrix H based on the tree T is equal to N, which is the number of all splitting nodes of the tree T plus 1, since we put into correspondence by one basis function (i.e. rows of the matrix H) to each splitting node of the tree T plus one (first) constant basis function corresponding to the root of the tree.

2. The rows of a TSH matrix H can be divided into the following $n+1$ subgroups: the first subgroup ($j = 0$) contains a single row which is a constant $\frac{1}{\sqrt{N}}$, the number of rows in the j-th subgroup ($j = 1, .., n$) is equal to the number of splitting nodes in the j-th level of the tree, and the number of nonzero elements in each row in the j-th level of the tree (corresponding to the splitting node in that level) is equal to the label of the corresponding splitting node, $j = 1, \ldots, n$.

Let us consider some constructions of the TSH matrices, based on the different structures of the underlying tree.

4.1. *Classical Haar matrices, or TSH matrices based on complete full binary tree*

Let T be complete full binary tree of the depth 3 having $N = 8$ leaves (see Fig. 4).

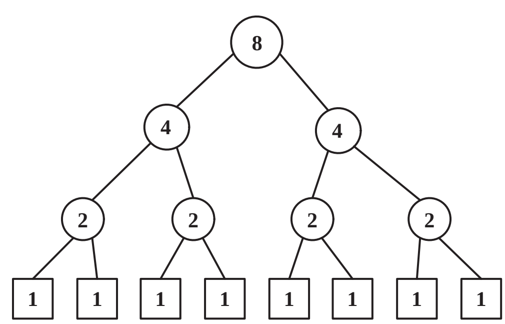

Figure 4. Complete full tree of the depth 3.

From (1) - (3) and Theorem 1, after discretization of TSH functions, we will obtain the classical Haar matrix of order 8:

$$H=\begin{pmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} & -\frac{1}{2\sqrt{2}} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

4.2. *Canonical Haar matrices, or TSH matrices based on full logarithmic binary tree*

Let T be a full logarithmic binary tree of the depth 4 having $N = 5$ leaves (see Fig. 5a). Then the corresponding TSH matrix will have a form

$$H=\begin{pmatrix} \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & -\frac{1}{2\sqrt{5}} & -\frac{1}{2\sqrt{5}} & -\frac{1}{2\sqrt{5}} & -\frac{1}{2\sqrt{5}} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{1}{2\sqrt{3}} & -\frac{1}{2\sqrt{3}} & -\frac{1}{2\sqrt{3}} \\ 0 & 0 & \frac{\sqrt{2}}{\sqrt{3}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}. \quad (7)$$

This matrix is known as the canonical Haar matrix[2] by Resnikoff and Wells [10].

Let T be another logarithmic binary tree of the depth 4 having $N = 5$ leaves (see Fig. 5b).

Then the corresponding TSH matrix will have a form

$$H=\begin{pmatrix} \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ \frac{1}{2\sqrt{5}} & \frac{1}{2\sqrt{5}} & \frac{1}{2\sqrt{5}} & \frac{1}{2\sqrt{5}} & -\frac{2}{\sqrt{5}} \\ \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{\sqrt{2}}{\sqrt{3}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 \end{pmatrix}. \quad (8)$$

This matrix is known as the Fibonacci matrix by Agaian-Aizenberg-Alaverdian [1]. As it is not difficult to see, the canonical Haar matrix can be obtained by permutations of the columns of the Fibonacci matrix.

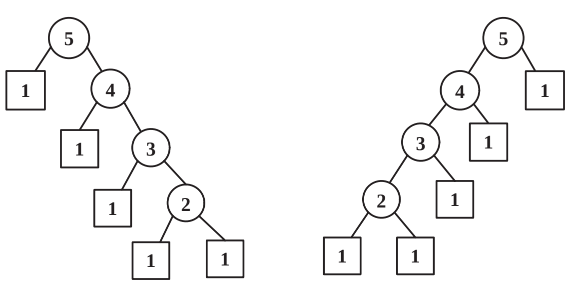
a) b)

Figure 5. Logarithmic full binary trees.

4.3. *Generalized Fibonacci-Haar p-matrices, or TSH matrices based on generalized Fibonacci p-tree*

Fibonacci matrix [1] is the special case of generalized Fibonacci-Haar matrices [7], since the logarithmic tree is the particular case of the Fibonacci p-tree when $p \to \infty$.

Let T be a generalized Fibonacci p-tree ($p = 2$) of the depth 4 having $N = 6$ leaves (see Fig. 6).

Then the corresponding TSH matrix will have a form

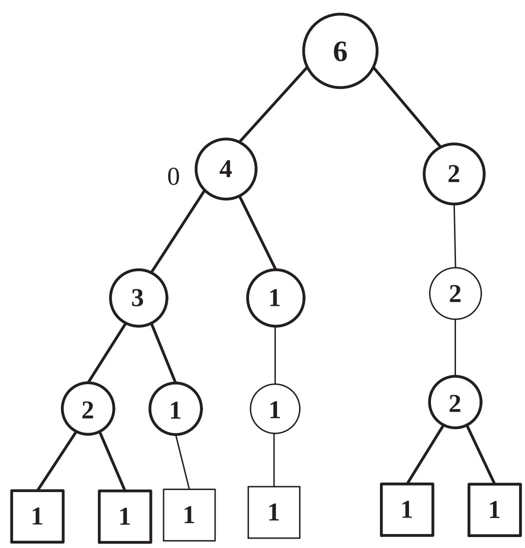

Figure 6. Generalized Fibonacci 2-tree.

$$H = \begin{pmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & \frac{1}{2\sqrt{3}} & -\frac{\sqrt{3}}{2} & 0 & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{\sqrt{2}}{\sqrt{3}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}. \quad (9)$$

4.4. *Other examples of TSH matrices based on binary interval splitting trees*

Let T be an arbitrary binary tree. As an example we will consider T without any specific predefined form. In Figure 7 three such binary trees are depicted. First of them is a full one, and the two others are isomorphic[3] to the first one.

Similarly to the previous constructions, the TSH matrix for the first and the second trees will have the same form:

$$H_1 = \begin{pmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{\sqrt{3}}{2\sqrt{10}} & \frac{\sqrt{3}}{2\sqrt{10}} & \frac{\sqrt{3}}{2\sqrt{10}} & \frac{\sqrt{3}}{2\sqrt{10}} & \frac{\sqrt{3}}{2\sqrt{10}} & -\frac{\sqrt{5}}{2\sqrt{6}} & -\frac{\sqrt{5}}{2\sqrt{6}} & -\frac{\sqrt{5}}{2\sqrt{6}} \\ \frac{1}{2\sqrt{5}} & \frac{1}{2\sqrt{5}} & \frac{1}{2\sqrt{5}} & \frac{1}{2\sqrt{5}} & -\frac{2}{\sqrt{5}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \end{pmatrix} \quad (10)$$

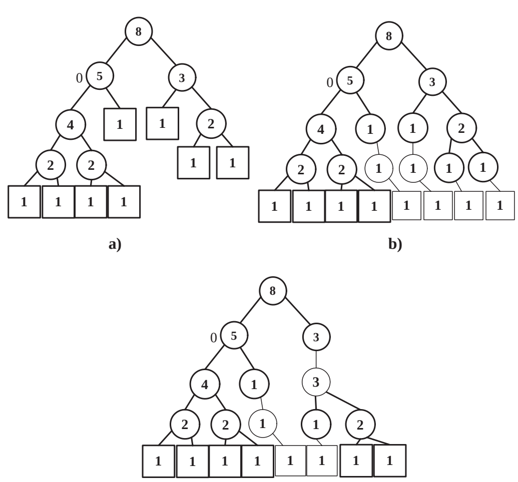

Figure 7. Three isomorphic binary interval splitting trees.

and the TSH matrix for the third tree is the following:

$$H_2 = \begin{pmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} \\ \frac{\sqrt{3}}{2\sqrt{10}} & \frac{\sqrt{3}}{2\sqrt{10}} & \frac{\sqrt{3}}{2\sqrt{10}} & \frac{\sqrt{3}}{2\sqrt{10}} & \frac{\sqrt{3}}{2\sqrt{10}} & -\frac{\sqrt{5}}{2\sqrt{6}} & -\frac{\sqrt{5}}{2\sqrt{6}} & -\frac{\sqrt{5}}{2\sqrt{6}} \\ \frac{1}{2\sqrt{5}} & \frac{1}{2\sqrt{5}} & \frac{1}{2\sqrt{5}} & \frac{1}{2\sqrt{5}} & -\frac{2}{\sqrt{5}} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \quad (11)$$

As we can see, one of these matrices can be obtained from another one just by permutation of its rows. This property holds for all the matrices corresponding to isomorphic trees.

5. Fast tree-structured Haar transforms

In the following we will discuss the properties of the transform defined by a TSH-matrix.

Definition 4. Let T be a binary interval splitting tree with N leaves and H its TSH-matrix. The TSH-transform corresponding to T of a column vector f of lenth N is $F = Hf$.

It is well known that the classical Haar transform (corresponding to the case of complete full binary tree, see section 4.1) can be computed very efficiently with an algorithm of linear complexity. This algorithm is based on decomposing the Haar matrix into a product of sparse matrices. This reduces the complexity from the $(N-1)^2$ addition and N^2 multiplication operations of the direct implementation to $2(N-1)$ addition and N multiplication operations [3].

We will show that also the matrix of a tree-structured Haar transform can be decomposed into a product of sparse matrices and derive an straightforward algorithm to find the factorization. This factorization leads to an algorithm of linear complexity to compute a TSH-transform.

Consider a binary tree $\tilde{T}$ of depth n and its tree-structured Haar matrix H constructed according to Definition 3. As noted in Section 3 we can assume that $\tilde{T}$ is a full tree and, thus, all its internal nodes are splitting nodes. Now, without changing the matrix H we can replace $\tilde{T}$ with T that is obtained from $\tilde{T}$ by adding non-splitting nodes to make each path from root to leaf have length n.

Now, T has $n+1$ levels, $0, 1, \ldots, n$ and we denote by μ_j the number of nodes on level j and by σ_j the number of splitting nodes on level j. Let $c(j, i)$, $i = 1, \ldots, \mu_j$ be the nodes of level j from left to right and, as before, we label each node $c(j, i)$ by the number $\nu(c(j, i))$ of leaves in the subtree rooted at $c(j, i)$.

Let us construct the matrices A_j, $j = 1, \ldots, n$ as follows.

$$A_j = \begin{pmatrix} P_j & 0 \\ 0 & I_{N-\mu_j} \end{pmatrix}, \; j = 1, ..., n-1, \tag{12}$$

where I_k is the identity matrix of order k, and P_j is the $(\mu_j \times \mu_j)$ matrix of the form

$$P_j = \begin{pmatrix} U_j \\ V_j \end{pmatrix}, \tag{13}$$

$A_n = P_n$.

U_j is the $(\mu_{j-1} \times \mu_j)$ block diagonal matrix constructed as follows.

Initialize the $(\mu_j \times \mu_j)$ matrix $\tilde{U}_j = 0$
For $i = 1, \ldots, \mu_j$
if $c(j, i) =$ left child, then $\tilde{U}_j(i, i) = \tilde{U}_j(i, i+1) = 1$
if $c(j, i) =$ right child, then $\tilde{U}_j(i, i) = 0$
if $c(j, i) =$ only child, then $\tilde{U}_j(i, i) = 1$
End For
U_j is now obtained from $\tilde{U}_j$ by deleting all zero rows.

$V_j$is the $((\mu_j - \mu_{j-1}) \times \mu_j)$ block diagonal matrix constructed as follows.

Initialize $(\mu_j \times \mu_j)$ matrix $\tilde{V}_j = 0$
For $i = 1, \ldots, \mu_j$
if $c(j, i) =$ left child, then $\tilde{V}_j(i, i) = \nu(c(j, i+1))$, $\tilde{V}_j(i, i+1) = -\nu(c(j, i))$

if $c(j, i) =$ right child, then $\tilde{V}_j(i, i) = 0$
if $c(j, i) =$ only child, then $\tilde{V}_j(i, i) = 0$
End For
V_j is now obtained from $\tilde{V}_j$ by deleting all zero rows.

Let D be the diagonal $N \times N$ matrix that normalizes $A_1, A_2, \ldots, A_n$. We note that D can be read directly from the tree T in the following way. Let $c_1, \ldots, c_{N-1}$ be all the splitting nodes scanned level by level from left to right and $c_{1,0}$ and $c_{1,1}$ their left and right children respectively. Then

$$D = \operatorname{diag}\left(\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{(\nu(c_{1,0})\nu(c_{1,1}))\nu(c_1)}}, \cdots, \frac{1}{\sqrt{(\nu(c_{N-1,0})\nu(c_{N-1,1}))\nu(c_{N-1})}} \right), \tag{14}$$

Now, we are ready to state

Theorem 2. *Let H, $A_1, A_2, \ldots, A_n$ and D as above. Then*

$$H = DA_1A_2 \cdots A_n. \tag{15}$$

Corollary 1. *There exists a fast algorithm computing a tree-structured Haar transform of length N with $2(N-1)$ additions and $3N-2$ multiplications.*

Proof: According to Property 1, the rows of TSH matrix H consist of the following submatrices:

$$H = D \begin{pmatrix} H_{root} \\ H_1 \\ H_2 \\ \cdots \\ H_{n-1} \end{pmatrix} \tag{16}$$

First, we show that, for $j = 0, 1, ..., n-1$:

$$B_j = A_{j+1} \cdots A_n = \begin{pmatrix} M_j \\ H_j \\ \vdots \\ H_{n-1} \end{pmatrix} \tag{17}$$

where $H_j, \cdots, H_{n-1}$ are submatrices from (16), and

$$M_j = \operatorname{diag}(e_{\nu(c(j,1))}, e_{\nu(c(j,1))}, \cdots, e_{\nu(c(j,\mu_j))}), \tag{18}$$

where e_k is the row vector of ones of the length k.

One can easily check that the last $\mu_n - \mu_{n-1}$ rows of the matrix P_n form the submatrix H_{n-1} from (16), and the first μ_{n-1} rows form a matrix M_{n-1} from (18). Thus, the formula (17) holds for $j = n-1$. Let us prove (17) by induction. Let this equation hold for $j = k$. We will show that it holds also for $j = k-1$, i.e.

$$B_k = \begin{pmatrix} M_k \\ H_k \\ \vdots \\ H_{n-1} \end{pmatrix} = A_{k+1} B_{k+1} = A_{k+1} \begin{pmatrix} M_{k+1} \\ H_{k+1} \\ \vdots \\ H_{n-1} \end{pmatrix}.$$

This is equivalent to show that

$$P_k M_{k+1} = \begin{pmatrix} M_k \\ H_k \end{pmatrix},$$

or, equivalently,

$$U_k M_{k+1} = M_k, \text{ and } V_k M_{k+1} = H_k.$$

The first equation is true since the label of the parent is equal to the sum of the labels of its children, and the second equation is true by the construction of the basis functions.

The complexity of the fast tree-structured Haar transform based on the decomposition (16) consists of the following operations: N multiplications from product by the diagonal matrix D, $2(N-1)$ additions and multiplications from the products by the matrices P_j. Overall these result in $2(N-1)$ additions and $3N-2$ multiplications. The theorem and its corollary are proved.

Illustrations of how this algorithm works for the trees presented in Figure 7 are given in Figure 8 and Figure 9.

The action of the proposed algorithm for fast tree-structured Haar transform can be represented by the logarithmic tree decomposition structure, presented in Figure 10 a). Starting from the root of the tree where we have an input signal - vector x of the length N to be transformed by the TSH matrix H, we apply to x 2 operators: the "low-pass" operator U_n and the "high-pass" operator V_n, defined in Theorem 2. This will divide our signal into 2 parts: signal $x^{(1)} = U_n x$ of the length μ_{n-1}, and the signal $V_n x$ of the length $\mu_{n-1} - \mu_{n-2}$. This procedure is continued with the vector $x^{(1)}$, applying to it operators U_{n-1} and V_{n-1}, resulting again in 2 signals: $x^{(2)} = U_{n-1} x^{(1)}$ of the length μ_{n-2}, and the signal $V_{n-1} x^{(1)}$ of the length $\mu_{n-2} - \mu_{n-3}$. Continuing this process by the same way with $x^{(2)}$, etc., until $x^{(n)} = U_1 x^{(n-1)}$, we will obtain the resulting transform (reading from left to right the leaves in Figure 10 a), without normalization by the diagonal matrix D.

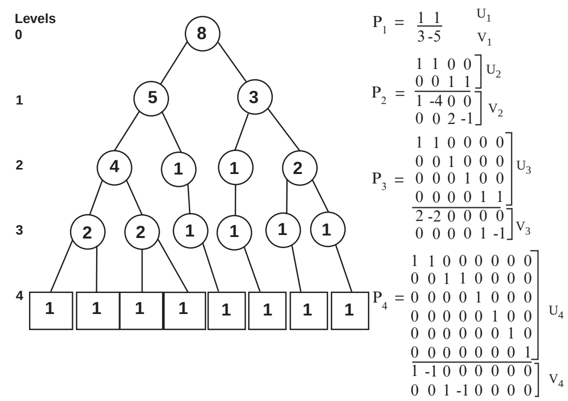

Figure 8. Decomposition matrices for the TSH matrix for the tree from the Figure 7a).

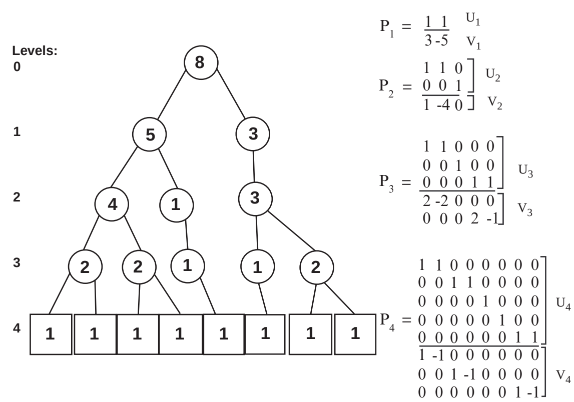

Figure 9. Decomposition matrices for the TSH matrix for the tree from the Figure 7c).

In the Figure 10 b) and c) this decomposition tree is presented for the fast TSH transforms of Figures 8 and 9, respectively.

For computation of the inverse TSH transform we can use a similar logarithmic tree as in Figure 10, starting now from the leaves of the tree and combining results by adding them from both "low-pass" and "high-pass" parts. Since the basic computational elements of the fast TSH algorithm are so-called "butterfly" operations – transforms by the matrices of the form $Q = \begin{pmatrix} 1 & 1 \\ a & -b \end{pmatrix}$ (see the structure of the matrices P_j in the Theorem 2), in order to invert the operation we need to apply the inverse "butterfly" $Q^{-1} = \begin{pmatrix} b & 1 \\ a & -1 \end{pmatrix} / (a+b)$.

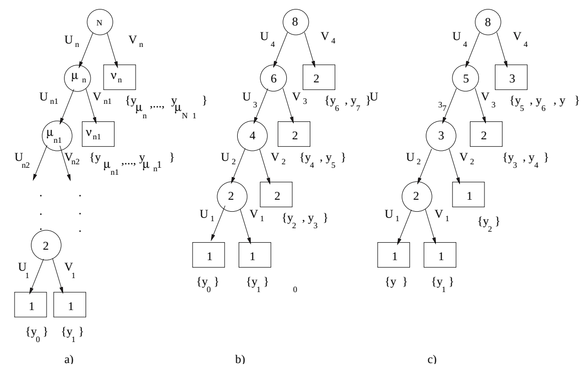

Figure 10. Logarithmic tree decomposition schemes for the TSH transforms: a) general scheme, b) a scheme for the TSH transform from Figure 8, c) a scheme for the TSH transform from the Figure 9.

Another possibility for efficient implementation of TSH transforms (both direct and inverse) is to use a time-varying switching filter banks, as it is done for the generalized Fibonacci - Haar transforms in [5].

6. From (time) interval splitting trees to scale splitting trees: the case of general time-scale tiling

In the previous sections we have defined an orthogonal tree-structured Haar (TSH) transform based on the concept of binary interval splitting trees, and developed a fast computational scheme for TSH transforms. This fast computational scheme gives a logarithmic tree decomposition structure for any TSH transform. But this is just one possible case of decomposition structure. Changing this logarithmic decomposition tree to any binary decomposition tree structure, we come to the concept of the *tree-structured Haar* (*TSH*) *packets*[4]. In another extreme case, when the decomposition tree is complete and full, we will obtain an extension of the Walsh transform [3].

As an example, the Fibonacci - Walsh p-matrix defined by the tree structure given in Figure 6 ($p = 2$), as well as its decomposition into the product of sparse matrices is given below:

$$H = D\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & -4 & -4 \\ 1 & 1 & 1 & -3 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & -1 \\ 1 & 1 & -2 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 \end{pmatrix}$$

$$= D\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & -4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\times \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix},$$

where

$$D = \mathrm{diag}\left(\frac{1}{\sqrt{6}}, \frac{1}{4\sqrt{3}}, \frac{1}{2\sqrt{3}}, \frac{1}{2}, \frac{1}{\sqrt{6}}, \frac{1}{2}\right).$$

7. Conclusion

Based on the concept of binary interval splitting trees, a class of Haar-like orthogonal systems and transforms based on them is defined. For any binary tree we associate a system of Haar-like orthogonal functions – tree-structured Haar (TSH) functions. Particular cases of these systems are the classical Haar system (the case of complete full binary tree), the canonical Haar system (the case of logarithmic tree), the Fibonacci Haar system (the case of Fibonacci tree), etc. A fast algorithm of linear complexity for computing TSH transform is developed. An extension of the TSH transforms toward TSH packets is described.

Notes

1. In the case of a full binary tree each node *a* will be indexed by a binary vector $(\alpha_1(a), ..., \alpha_k(a))$ $(\alpha_j \in \{0, 1\}, j = 1, ..., k)$
2. changing sign of all rows except the first one
3. here we call trees isomorphic if after deleting all non-splitting non-leaf nodes from these trees we will obtain exactly the same trees
4. name is similar to the Haar wavelet packets (which is a particular case of TSH packets [14]) based on the similar idea to expand a Haar transform

References

1. S.S. Agaian, "Advances and Problems of Fast Orthogonal Transforms for Signal-Images Processing Applications (part 1)," in the Yearbook: *Pattern Recognition, Classification, Forecasting*, Nauka, Moscow, No. 3, pp. 146–215, 1992 (in Russian).
2. N. Ahmed, K.R. Rao, "Orthogonal Transforms for Digital Signal Processing," New York: Springer 1975.
3. K.G. Beauchamp, *Walsh Functions and Their Applications*, Academic Press, 1975.
4. K. Egiazarian and J. Astola, "Discrete orthogonal transforms based on Fibonacci-type recursion," *Proc. IEEE Digital Signal Processing Workshop (DSPWS-96)*, Norway, 1996.
5. K. Egiazarian, D. Gevorkian, and J. Astola, "Time-varying filter banks and multiresolution transforms based on generalized Fibonacci topology," *Proc. 5th IEEE Int. Workshop on Intelligent Signal Proc. and Communication Systems*, Kuala Lumpur, Malaysia, 11–13 Nov. 1997, S16.5.1–S16.5.4.
6. K. Egiazarian, J. Astola, and S. Agaian, "Orthogonal transforms based on generalized Fibonacci recursions", *Proc. 2nd Int. Workshop on Spectral Techniques and Filter Banks*, Brandenburg, Germany, March 5–7, 1999.
7. K. Egiazarian and J. Astola, "On generalized Fibonacci cubes and unitary transforms," *Applicable Algebra in Engineering, Communication and Computing*, Vol. AAECC 8, 1997, pp. 371–377.
8. D.F. Elliott and K.R. Rao, *Fast Transforms: Algorithms, Analyses, Applications*, Academic Press, 1982.
9. A. Haar, "Zur theorie der orthogonalen functionsysteme," *Math. Annal.*, Vol. 69, pp. 331–371, 1910.
10. H.L. Resnikoff and R.O. Jr. Wells, *Wavelet Analysis: The Scalable Structure of Information*, Springer, 1998.
11. A.P. Stakhov, *Algorithmic Measurement Theory*, Moscow: Znanie 1979 (in Russian).
12. M. Vetterli and J. Kovaĉeviĉ, *Wavelets and Subband Coding*, Prentice Hall, 1995.
13. J. L. Walsh, "A closed set of orthogonal functions," *Amer. J. Math.*, Vol. 55, pp. 5–24, 1923.
14. M.V. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, IEEE Press, A.K.Peters, Wellesley, Massachusetts, 1994.

Karen Egiazarian was born in Yerevan, Armenia, in 1959. He received the M.Sc. degree in mathematics from Yerevan State University in 1981, and the Ph.D. Degree in physics and mathematics from Moscow M.V. Lomonosov State University in 1986. In 1994 he was awarded the degree of Doctor of Technology by Tampere University of Technology, Finland. He has been a Senior Researcher at the Department of Digital Signal Processing of the Institute of Information Problems and Automation, National Academy of Sciences of Armenia. He is currently a Full Professor in the Institute of Signal Processing at Tampere University of Technology. His research interests are in the areas of applied mathematics, digital logic, signal and image processing. He has published more than 200 articles in these areas, and is coauthor (with S. Agaian and J. Astola) of the book "Binary Polynomial Transforms and Nonlinear Digital Filters", published by Marcel Dekker, Inc. in 1995, and three book chapters.

Jaakko Astola was born in Helsinki, Finland, in 1949. He received the B.Sc., M.Sc., Licenciate, and Ph.D. degrees in Mathematics from Turku University, Turku, Finland, in 1972, 1973, 1975, and 1978, respectively.

From 1976 to 1977, he was a Research Assistant at the Research Institute for Mathematical Sciences of Kyoto University, Kyoto, Japan. Between 1979 and 1987, he was with the Department of Information Technology, Lappeenraanta University of Technology, Lappeenraanta, Finland, holding various teaching positions in mathematics, applied mathematics, and computer science. From 1988 to 1993, he was an Associate Professor in applied mathematics at Tampere University, Tampere, Finland. Currently, he is a Professor of digital signal processing at Tampere University of Technology, the head of Signal Processing Laboratory and the Director of the Tampere International Center in Signal Processing (TICSP). His research interests include signal processing, coding theory, and statistics.