New Orthogonal Transforms for Signal and Image Processing

Andrzej Dziech

Institute of Telecommunications, AGH University of Science and Technology, Mickiewicza 30, 30-059 Kraków, Poland; adzie@tlen.pl

Abstract: In the paper, orthogonal transforms based on proposed symmetric, orthogonal matrices are created. These transforms can be considered as generalized Walsh–Hadamard Transforms. The simplicity of calculating the forward and inverse transforms is one of the important features of the presented approach. The conditions for creating symmetric, orthogonal matrices are defined. It is shown that for the selection of the elements of an orthogonal matrix that meets the given conditions, it is necessary to select only a limited number of elements. The general form of the orthogonal, symmetric matrix having an exponential form is also presented. Orthogonal basis functions based on the created matrices can be used for orthogonal expansion leading to signal approximation. An exponential form of orthogonal, sparse matrices with variable parameters is also created. Various versions of orthogonal transforms related to the created full and sparse matrices are proposed. Fast computation of the presented transforms in comparison to fast algorithms of selected orthogonal transforms is discussed. Possible applications for signal approximation and examples of image spectrum in the considered transform domains are presented.

Keywords: orthogonal transform; signal processing; image processing

1. Introduction

The literature on orthogonal transforms for signal and image processing is extensive, including the often-cited books by Ahmed and Rao [1] and Wang [2]. In this paper, a new orthogonal transform is proposed that can be treated as a generalized Walsh–Hadamard transform [3–9]. A modification of the proposed transform is also investigated. This transform has a simple structure leading to a fast calculation of both forward and inverse transforms and hence very fast computational algorithms with a significant reduction in the number of calculations required. Indeed, the forward and inverse transforms have the same structure, only differing in the constant coefficient. The proposed transforms could be effectively applied in such areas of signal and image processing as watermarking technology [10–12], steganography, intelligent monitoring and others [13–16]. The general form of this transform enables the possibility of selecting optimal parameters for specific tasks, implying there are potentially other applications. Further investigations are required to determine in which applications it has maximum effectiveness.

2. Orthogonal Generalized Transform Matrix

Let us consider an elementary square matrix \( N \times N, N = 2 \) consisting of four elements. Assuming the first row of this matrix has different elements \( a_1, a_2 \) and the matrix is symmetric and orthogonal, we have:

\[
A(1) = \begin{bmatrix}
a_1 & a_2 \\
a_2 & -a_1
\end{bmatrix}
\]

where \( a_1, a_2 \in (-\infty, \infty) \), \( a_1, a_2 \)—any real numbers, \( a_1, a_2 \neq 0 \).

In the case of the matrix \( A(n) \), where \( n = \log_2 N, N = 4 \) the first row consists of four different elements \( a_1, a_2, a_3, a_4 \in (-\infty, \infty) \), \( a_1 \neq 0 \). Such a sequence is referred to as
a basis sequence of the matrix. Our aim is to create a symmetric and orthogonal matrix $A(2)$ for the given basis sequence $a_1, a_2, a_3, a_4$. This matrix consists of four elementary matrices $A(1)$, having the structure of (1), as shown below:

$$A(2) = \begin{bmatrix} A_1(1) & A_2(1) \\ A_2(1) & -A_1(1) \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & -a_1 & a_4 & -a_3 \\ a_3 & a_4 & -a_1 & -a_2 \\ a_4 & -a_3 & -a_2 & a_1 \end{bmatrix}$$ (2)

The matrix (2) is symmetric, i.e., the forward matrix is equal to its transpose: $A(2) = A^T(2)$. In order to obtain the condition for which the matrix (2) is orthogonal it is necessary to consider $(A_1^2) = 6$ dot products of any two rows: $v_i \cdot v_j, i \neq j$ (or columns) of the matrix (2). Then, each dot product should satisfy the equation: $v_i \cdot v_j = 0$. The following condition for the basis sequence is obtained to create an orthogonal matrix (2):

$$a_1a_4 = a_2a_3$$ (3)

Our purpose is to obtain a general rule for the creation of a symmetric and orthogonal matrix $A(n)$ for any $n = \log_2 N$. Let us consider now the basis sequence consisting of $N$ elements represented by any real numbers $a_1, a_2, a_3, \ldots, a_N, a_i \in (-\infty, \infty), a_i \neq 0, N \geq 4, N = 2^n, n = 2, 3, 4, 5, 6, \ldots$ in order to obtain a symmetric and orthogonal matrix $A(n)$. A general form of such a matrix (having the structure of (1) and (2)) is given by:

$$A(n) = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 & A_5 & \cdots & A_{N/2} \\ A_2 & -A_1 & A_4 & -A_3 & A_6 & \cdots & A_{N/2} \\ A_3 & A_4 & -A_1 & -A_2 & A_7 & \cdots & \vdots \\ A_4 & -A_3 & -A_2 & A_1 & A_8 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{N/2} & \vdots & \vdots & \vdots & \vdots & \ddots & A_{N/2} \end{bmatrix}$$ (4)

Each of the matrices $A_1, A_2, \ldots, A_{N/2}$ has the size $N = 2$ and the structure of matrix (1).

To ensure the orthogonality of matrix $A(n)$ for each group of four successive elements of the basis sequence $a_1, a_2, a_3, a_4, a_5, \ldots, a_n$, the following condition holds true:

$$a_i \cdot a_{i+3} = a_{i+1} \cdot a_{i+2}$$ (5)

for

$$i = 1, 2, 3, \ldots, N-2, N = 2^n.$$

Thus, for the basis sequence consisting of $N$ elements ($N \geq 4$), the $(N-3)$ equations defined by (5) should be fulfilled. In this case, the matrix $A(n)$ is symmetric and orthogonal, i.e.,

$$A(n) = A^T(n), A(n) \cdot A^T(n) = C \cdot I, \frac{1}{C} \cdot A^T(n) = A^{-1}(n)$$ (6)

where

$A^T(n)$—transpose matrix $I$—unit matrix.

$A^{-1}$—inverse matrix.

C—coefficients $C = \sum_{i=1}^{N} a_i^2$.

It is seen that for the selection of basis sequences that fulfill (5), it is necessary to select only the first three elements of the sequence and then calculate the successive elements as follows:

$$a_{i+3} = \frac{a_{i+1} \cdot a_{i+2}}{a_i} \quad \text{for} \quad i = 1, 2, 3, \ldots, N-3$$ (7)
where also fulfilled for the more general case when the basis sequence is in the form: Equation (4), the basis sequence for this case is:

3. Exponential Form of the Transform Matrix

The matrix has the form:

obtained by multiplying the previous group by \( \beta \). Another sequence matching Equation (5) is the sequence 3, 9, 2, 6, . . . ; \( a_8, a_4, a_2, a_1 \). Finally, we obtain the basis sequence for which the matrix \( A \) Equation (5) to be not fulfilled. For instance, by changing the elements 1, 2, 3, 4 and the matrix \( A \) 4 and the matrix \( A \) 3 is:

The above matrix is symmetric and orthogonal and Equation (5) holds true. It should be emphasized that changing the order of basis sequence elements causes at least one of Equation (5) to be not fulfilled. For instance, by changing the elements 1, 2, 3, 4 and the matrix \( A \) the matrix is not orthogonal. An example of another sequence matching Equation (5) is the sequence 3, 9, 2, 6, 4, 8, 8, or -2, 1, 4, -2, -8, 4, 16, -8. For each subsequent group of four elements of the basis sequence:

there is a relationship:

The coefficient \( \beta \) may be chosen arbitrarily as any positive number or by Equation (5). Using this equation we find:

This means that each subsequent group of four elements of the basis sequence is obtained by multiplying the previous group by \( \beta \) under condition (3).

In the general case, the basis sequence for the creation of a symmetric and orthogonal matrix has the form:

For instance, to create the basis sequence for the orthogonal matrix \( A(3) \) we select the first three elements: \( a_1, a_2, a_3 \), e.g., 1, \( \frac{1}{2} \), 2. Then, the next elements are obtained using (7):

\[ a_4 = \frac{a_2 a_4}{a_2} = 1, \quad a_5 = \frac{a_2 a_5}{a_5} = 4, \text{ etc.} \]

Finally, we obtain the basis sequence: 1, \( \frac{1}{2} \), 2, 1, 4, 2, 8, 4 and the matrix \( A(3) \) is:

\[
A(3) = \begin{bmatrix}
1 & \frac{1}{2} & 2 & 1 & 4 & 2 & 8 & 4 \\
\frac{1}{2} & -1 & 1 & -2 & 2 & -4 & 4 & -8 \\
2 & 1 & -1 & -\frac{1}{2} & 8 & 4 & -4 & -2 \\
1 & -2 & -\frac{1}{2} & 1 & 4 & -8 & -2 & 4 \\
-\frac{1}{2} & 2 & 8 & -\frac{1}{2} & 4 & -1 & -\frac{1}{2} & -2 & -1 \\
2 & -4 & 4 & -8 & 1 & 2 & -1 & 2 \\
8 & 4 & -4 & -2 & 2 & -1 & 1 & \frac{1}{2} \\
4 & -8 & -2 & 4 & 1 & -1 & 2 & \frac{1}{2} & -1
\end{bmatrix}
\] (8)

3. Exponential Form of the Transform Matrix

If we select the powers of the real number \( a \) as elements of matrix \( A(n) \) defined by Equation (4), the basis sequence for this case is:

\[ a^1, a^2, a^3, a^4, \ldots, a^n, \quad N = 2^n, \quad n = 2, 3, 4, 5, \ldots \]

The above sequence meets the condition expressed by Equation (5). This condition is also fulfilled for the more general case when the basis sequence is in the form:

\[ a^k, a^{k+1}, a^{k+2}, a^{k+3}, \ldots, a^{k+(N-1)} \]

where

- \( a \)—any real number, \( a \neq 0 \).
- \( k \)—integer, \( k \in (-\infty, +\infty) \).
For this case, Equation (5) is:

\[ a^i \cdot a^{i+3} = a^{i+1} \cdot a^{i+2} \quad \text{for } i = k, k + 1, k + 2, \ldots \]  

(14)

We now define the general form of the matrix \( A(n) \) having an exponential form of order \( N, N = 2^n \), for which the following recursive relationship holds:

\[
A(n) = \begin{bmatrix}
A(n-1) & A(n-1) \cdot a^{2(n-1)} \\
A(n-1) \cdot a^{2(n-1)} & -A(n-1)
\end{bmatrix}
\]

(15)

where \( n = 1, 2, 3, \ldots \) and \( A(0) = a^k, k \) — integer.

The matrix \( A(n) \) defined by Equation (15) is symmetric and orthogonal.

The coefficient \( C \) is equal to the energy of the row of matrix \( A(n) \) and is defined as follows:

\[ C = \sum_{i=1}^{N} a^{2i} = \text{const} \]

(16)

Using Equation (15) for \( n = 1 \) and assuming \( k = 1 \), the matrix \( A(1) \) is:

\[
A(1) = \begin{bmatrix}
a & a^2 \\
a^2 & -a
\end{bmatrix}
\]

(17a)

For \( n = 2 \) we have:

\[
A(2) = \begin{bmatrix}
A(1) & A(1) \cdot a^2 \\
A(1) \cdot a^2 & -A(1)
\end{bmatrix}
= \begin{bmatrix}
a & a^2 & a^3 & a^4 \\
a^2 & -a & a^4 & -a^3 \\
a^3 & a^4 & -a & -a^2 \\
a^4 & -a^3 & a & a^2
\end{bmatrix}
\]

(17b)

For \( n = 3 \) we have:

\[
A(3) = \begin{bmatrix}
A(2) & A(2) \cdot a^4 \\
A(2) \cdot a^4 & -A(2)
\end{bmatrix}
\]

(17c)

The choice of element \( a \) depends on the specific case of the analysis. It should be noted that for the particular case when \( a = 1 \) matrix \( A(n) \) becomes the Hadamard matrix. For the basis sequence represented by Equations (12) and (13) we have the following relationship:

\[
a^{k+(N-1)/2} \cdot \sum_{i=k}^{k+(N-1)} a^{i} = a^{i-k+3(N-1)/2}
\]

(18)

If two matrices \( A_1(n) |_k \) and \( A_2(n) |_l \), based on Equation (15) have the basis sequences \( a^k, a^{k+1}, a^{k+2}, \ldots, a^{k+(N-1)} \) and \( a^l, a^{l+1}, a^{l+2}, \ldots, a^{l+(N-1)} \), respectively, for \( k \neq l \) then:

\[
A_1(n) |_k \cdot A_2(n) |_l = C_0 \cdot I
\]

(19)

where

\[
C_0 = \sum_{i=1}^{N} a_{i_1} \cdot a_{i_2}
\]

\( a_{i_1}, a_{i_2} \)—the elements of basis sequences for matrices \( A_1(n) \) and \( A_2(n) \), respectively.

4. Signal Approximation—Orthogonal Expansion

The set of rows of matrix \( A(n) \) defined by Equation (15) or Equation (4) represents the set of orthogonal discrete basis functions. We assume that these functions are defined within the interval \([0, T]\). In relation to the rows of matrix \( A(n) \) we have:

\[
\sum_{i,j} \varphi_i \cdot \varphi_j = \begin{cases} 
0 & \text{for } i \neq j \\
C & \text{for } i = j 
\end{cases} \\
C = \sum_{i=1}^{N} (a^i)^2
\]

(20)
An example of \( N = 4 \) orthogonal basis functions based on matrix \( A(2) \) for parameter \( a = \frac{1}{2} \) is illustrated in Figure 1.

![Basis functions](image)

Figure 1. Illustration of 4 orthogonal basis functions.

A continuous signal with finite energy \( x(t) \) can be approximated by linear combinations of \( N \) orthogonal basis functions represented by rows of \( A(n) \) [17]. Such an orthogonal expansion is expressed as follows:

\[
x_A(t) = \sum_{i=1}^{N} d_i \cdot \varphi_i(t)
\]  

Multiplying both sides of the Equation (21) by \( \varphi_j(t) \), then integrating both sides and accounting for the orthogonality of the basis functions we obtain:

\[
d_i = \frac{1}{C} T \int_{0}^{T} x_A(t) \cdot \varphi_i(t) dt
\]  

where \( C \) is defined by Equation (20).

Figure 2 shows the approximated signals for a sinewave for the given \( N \) and selected value of \( a \).
As can be seen from the results in Figure 1 (as expected), increasing the number of basis functions $N$ decreases the value of the mean square error. For instance, for $N = 4, 8, 32, 64, 128$ the $mse$ equals: $2.245 \times 10^{-1}, 3.09 \times 10^{-2}, 1.6 \times 10^{-3}, 4.0096 \times 10^{-4}, 9.8877 \times 10^{-5}$, respectively.

5. Proposed Orthogonal Transform and Its Modifications

5.1. One-Dimensional (1D) Transform

Let us consider a discrete one-dimensional signal represented by an $N$-element vector $X(n) = [x_1 \ x_2 \ \ldots \ \ x_N], N = 2^n$.

A 1D orthogonal Transform based on the matrix $A(n)$ defined earlier has the form:

Forward Transform: $\quad S(n) = A(n) \cdot X(n)$

Inverse Transform: $\quad X(n) = \frac{1}{C} \cdot A(n) \cdot S(n)$ \hspace{1cm} (23)

where

- $S(n)$—vector of spectral components.
- $X(n)$—vector of 1D signal.
- $A(n)$—transform matrix defined by Equation (15).
- $C$—constant defined by Equation (16).

As was shown, the transform matrix $A(n)$ is symmetric and orthogonal and $A^T = A^{-1}$. Therefore, the forward transform (FT) differs from the inverse transform (IT) only by the constant $C$. Another more detailed form of the FT can be written as follows:
where \( A(n) = a^n \), \( k = 1, 2, 3, \ldots \)

We generally assume that \( k = 1 \). Generally, we also assume that the parameter \( a \neq 0 \) is any real number within the interval \( (-\infty, +\infty) \). For a particular case \( a = 1 \) the transform (24) becomes the 1D Walsh–Hadamard transform.

### 5.2. Two-Dimensional (2D) Transform

This case applies to two-dimensional signals represented by the square matrix \( X(i, j) \) of \( N \) dimensions, \( N = 2^n \), \( i = 1, 2, 3, \ldots, N \), \( j = 1, 2, 3, \ldots, N \), which is mainly associated with image processing. In the general case a 2D transform has the form:

\[
\begin{align*}
\text{FT:} & \quad S(i, j) = A(n) \cdot X(i, j) \cdot A(n) \\
\text{IT:} & \quad X(i, j) = \frac{1}{C^2} \cdot A(n) \cdot S(i, j) \cdot A(n) 
\end{align*}
\]

(25)

where

- \( S(i, j) \) — matrix of spectral components.
- \( X(i, j) \) — matrix of 2D signal.
- \( A(n) \) — transform matrix described by Equation (15).
- \( C \) — constant described by Equation (16).

It is seen that the forward and inverse transforms have the same structure—they differ only in the constant \( C \). The 2D transform can be calculated directly using Equation (24) or by grouping the channels as shown in Figure 3e,f. The transform matrix \( A(n) \) consists of the power of parameter \( a \). For \( a = 1 \) we obtain the 2D Walsh–Hadamard transform. It is obvious that if the transform (25) is used for image processing we are dealing with large values of \( N \). For this reason, large powers of parameter \( a \neq 1 \) may not be calculated by a computer. Thus, it seems purposeful to propose a sparse matrix \( A_m(n) \) with the following form:

\[
A_m(n) = \begin{bmatrix}
A(m) & 0 & 0 & \cdots & 0 \\
0 & A(m) & 0 & \cdots & 0 \\
0 & 0 & A(m) & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & A(m)
\end{bmatrix}
\]

(26a)

where \( A_m(n) \) — matrix with dimension \( N = 2^n \), \( n = 2, 3, 4, \ldots \)

- \( A(m) \) — matrix (15) or (4) with dimension, \( M = 2^m \), \( m = 1, 2, 3, \ldots, m < n \).

For example, for the parameters \( n = 3, m = 2 \) the matrix \( A_2(3) \) is:

\[
A_2(3) = \begin{bmatrix}
A(2) & \cdots \\
\cdots & A(2)
\end{bmatrix}
\]

(26b)

where \( A(2) \) is defined either by (2) or (17b).
It is seen that like \( A(n) \) matrix \( A_m(n) \) is symmetric and orthogonal, so we have:

\[
A_m(n) = A_m^T(n), \quad A_m(n) \cdot A_m^{-1}(n) = C \cdot I
\]  
(27)

\[
A_m^{-1}(n) = A_m^T(n), \quad I — \text{unit matrix}
\]

where \( C = \sum_{i=1}^{2^m} a^{2i} \)  
(28)

The matrix \( A_m(n) \) consists of many zeros, defined by:

\[
L_0 = N \cdot (N - M)  
\]  
(29)

The percentage ratio of the number of zeros to all elements of the matrix \( A_m(n) \) is as follows:

\[
\frac{L_0}{L_{\text{total}}} = \frac{N \cdot (N - M)}{N^2} = \left(1 - \frac{M}{N}\right) \cdot 100\%  
\]  
(30)

For instance, if \( M = 4, N = 512, \) more than 99% of the transform matrix \( A_2(9) \) elements are zeros.

Matrix (26a) can therefore be used as a transform matrix for the following transform, which is a modified version of transforms (23) and (25) using submatrices \( A(m) \).

1D Transform:

\[
\text{FT: } S(n) = A_m(n) \cdot X(n) 
\]  
(31)

\[
\text{IT: } X(n) = \frac{1}{C} \cdot A_m(n) \cdot S(n) 
\]

2D Transform:

\[
\text{FT: } S(i,j) = A_m(n) \cdot X(i,j) \cdot A_m(n) 
\]  
(32)

\[
\text{IT: } X(i,j) = \frac{1}{C^2} \cdot A_m(n) \cdot S(i,j) \cdot A_m(n) 
\]

\( C \) is defined by Equation (28).

Transforms (31) and (32) with many zeros can be treated as a specific case of compression in the spatial domain. Calculating modified orthogonal transforms using sparse matrices expressed by (31) and (32) is not only simple but can also provide very fast calculations. However, this is the subject considered in the next section.

It is obvious that instead of using matrices (15) in all of the proposed orthogonal transforms (23), (25), (31) and (32) we can also use the symmetric and orthogonal matrix (4) as the transform matrix. In this case, we can consider different versions of the orthogonal transforms, in particular the transforms described by Equations (31) and (32) for which the submatrices \( A(m) \) are the same or different. It was shown that each submatrix \( A(m) \) of the matrix \( A_m(n) \) is defined by the first three values of the basis sequence. For different submatrices we deal with a matrix of coefficients instead of coefficient \( C \) for the inverse transform.

Figure 3 shows: \( a \)—the input image for \( N = 256 \) and its spectra obtained for various transform matrices; \( b \)—full exponential matrix \( (N = 256, a = 0.3) \); \( c \)—full exponential matrix \( (N = 256, a = 0.7) \); \( d \)—full non-exponential matrix \( (N = 256, \text{basis sequence: 1, 2, 3}) \); \( e \)—full exponential matrix after channel grouping \( (N = 8, a = 0.3; 8 \times 8 \text{ channels}) \); \( f \)—full exponential matrix after channel grouping \( (N = 8, a = 0.7; 8 \times 8 \text{ channels}) \); \( g \)—sparse exponential matrix \( (N = 256, M = 4, a = 0.7) \); \( h \)—sparse exponential matrix \( (N = 256, M = 16, a = 0.7) \); \( i \)—sparse non-exponential matrix with the same submatrices \( (N = 256, M = 8, \text{basis sequence: 1, 2, 3}) \); \( j \)—sparse non-exponential matrix with the same submatrices \( (N = 256, M = 16, \text{basis sequence: 1, 2, 3}) \); \( k \)—full non-exponential matrix with arbitrary \( \beta \) parameter \( (N = 256, \beta = 0.75 \text{ basis sequence: 1, 2, 3}) \); and \( l \)—full non-exponential matrix with arbitrary \( \beta \) parameter \( (N = 256, \beta = 1.25 \text{ basis sequence: 1, 2, 3}) \).
6. Fast Algorithms

Fast algorithms are used to reduce the number of computations required to determine the transform coefficients in comparison to direct computation of the transform. The main idea of efficient or fast computational algorithms is the ability to subdivide the total computational load into a series of computational steps in such a way that partial results obtained from initial steps can be repeatedly utilized in the subsequent steps.

Fast computation of this matrix can be performed by the well-known techniques of sparse matrix factoring or matrix partitioning. These techniques result in fast algorithms that reduce the computation requirements from $N^2$ additions using direct computation to $N \log_2 N$ additions using fast algorithms [2].

In Figure 4, the flow graph of a fast algorithm for the computation of a full exponential orthogonal matrix for $N = 8$ is presented.
The flow graph has a butterfly structure and is similar to the graph of the fast Walsh–Hadamard transform (WHT) algorithm [2].

The plot in Figure 5 shows the number of additions required for the proposed transform with respect to its dimensions. This plot also presents the comparison of the proposed transform (PT) and proposed sparse transforms (PST) for $M = 4$ and $M = 8$. with other well-known transforms, like periodic Haar piecewise-linear (PHL) [18], fast fourier (FFT), Walsh–Hadamard (WHT), Haar (HT), Slant (ST) and discrete cosine (DCT) transforms.

**Figure 4.** Construction of fast algorithm for the computation of full exponential orthogonal matrix for $N = 8$.

**Figure 5.** Number of additions in relation to transform matrix size.
7. Conclusions

The method of creating symmetric, orthogonal matrices and orthogonal transforms presented in this paper is a generalization of Hadamard matrices and the Walsh–Hadamard transform. The experiments performed show that the proposed transforms can be effectively used for signal and image processing. The advantage of these transforms is simplicity of implementation and a relatively small number of operations necessary for calculations. Moreover, an important feature of the considered transforms is the possibility of forming spectral components of 1D and 2D signals by selecting the transform matrix parameters, in particular the parameter $a$. This feature, combined with the variety of structures of the considered orthogonal transforms, also offers promising possibilities for further applications.

Funding: This research was funded by the European Union’s Horizon 2020 Research and Innovation Programme, under Grant Agreement No. 830943, the ECHO project.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The author would like to thank Piotr Bogacki from AGH University of Science and Technology, Kraków, for fruitful discussion and carrying out experiments and calculations, and Mariusz Ziółko from AGH University of Science and Technology for valuable discussions and comments.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Ahmed, N.; Rao, K.R. Orthogonal Transforms for Digital Signal Processing; Springer: Berlin/Heidelberg, Germany, 1975.
2. Wang, R. Introduction to Orthogonal Transforms: With Applications in Data Processing and Analysis; Cambridge University Press: Cambridge, UK, 2012.
3. Johnson, J.; Puschel, M. In search of the optimal Walsh–Hadamard transform. In Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, 5–9 June 2000; Volume 6, pp. 3347–3350.
4. Gonzalez, R.; Woods, R. Digital Image Processing, 4th ed.; Pearson: London, UK, 2018.
5. Madisetti, V. The Digital Signal Processing Handbook; CRC Press: Boca Raton, FL, USA, 2010.
6. Ashrafi, A. Chapter One—Walsh–Hadamard Transforms: A Review; Advances in Imaging and Electron Physics; Elsevier: Amsterdam, The Netherlands, 2017.
7. Sayood, K. Chapter 13—Transform Coding. Introduction to Data Compression, 5th ed.; The Morgan Kaufmann Series in Multimedia Information and Systems; Morgan Kaufman: Burlington, MA, USA, 2018.
8. Hamood, M.; Boussakta, S. Fast Walsh–Hadamard–Fourier Transform Algorithm. IEEE Trans. Signal Process. 2011, 59, 5627–5631. [CrossRef]
9. Thompson, A. The Cascading Haar Wavelet Algorithm for Computing the Walsh–Hadamard Transform. IEEE Signal Process. Lett. 2017, 24, 1020–1023. [CrossRef]
10. Korus, P.; Dziech, A. Efficient Method for Content Reconstruction With Self-Embedding. IEEE Trans. Image Process. 2013, 22, 1134–1147. [CrossRef] [PubMed]
11. Korus, P.; Dziech, A. Adaptive Self-Embedding Scheme With Controlled Reconstruction Performance. IEEE Trans. Inf. Forensics Secur. 2014, 9, 169–181. [CrossRef]
12. Kalarikkal Pullayikodi, S.; Tarhuni, N.; Ahmed, A.; Shiginah, F.B. Computationally Efficient Robust Color Image Watermarking Using Fast Walsh Hadamard Transform. J. Imaging 2017, 3, 46. [CrossRef]
13. Pan, H.; Dabawi, D.; Cetin, A. Fast Walsh–Hadamard Transform and Smooth-Thresholding Based Binary Layers in Deep Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Virtual Conference, 19–25 June 2021.
14. Subathra, M.S.P.; Mohammed, M.A.; Maashi, M.S.; Garcia-Zapirain, B.; Sairamya, N.J.; George, S.T. Detection of Focal and Non-Focal Electroencephalogram Signals Using Fast Walsh–Hadamard Transform and Artificial Neural Network. Sensors 2020, 20, 4952.
15. Wang, X.; Liang, X.; Zheng, J.; Zhou, H. Fast detection and segmentation of partial image blur based on discrete Walsh–Hadamard transform. Signal Process. Image Commun. 2019, 70, 47–56. [CrossRef]
16. Andrushia, A.D.; Thangarjan, R. Saliency-Based Image Compression Using Walsh–Hadamard Transform (WHT). In Biologically Rationalized Computing Techniques for Image Processing Applications; Springer: Cham, Switzerland, 2018.
17. Aristidi, E. Representation of Signals as Series of Orthogonal Functions. *EAS Publ. Ser.* **2016**, *78–79*, 99–126. [CrossRef]

18. Dziech, A.; Ślusarczyk, P.; Tibken, B. Methods of Image Compression by PHL Transform. *J. Intell. Robot. Syst.* **2004**, *39*, 447–458. [CrossRef]