Research Article

Generalized Orthogonal Discrete W Transform and Its Fast Algorithm

Jichao Sun¹,² and Zhengping Zhang ¹

¹College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China
²College of Computer and Information Engineering, Anhui Polytechnic University, Wuhu 241000, China

Correspondence should be addressed to Zhengping Zhang; zpzhang@gzu.edu.cn

Received 14 July 2020; Revised 19 November 2020; Accepted 26 November 2020; Published 11 January 2021

Academic Editor: Stefania Tomasiello

Copyright © 2021 Jichao Sun and Zhengping Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Based on the generalized discrete Fourier transform, the generalized orthogonal discrete W transform and its fast algorithm are proposed and derived in this paper. The orthogonal discrete W transform proposed by Zhongde Wang has only four types. However, the generalized orthogonal discrete W transform proposed by us has infinite types and subsumes a family of symmetric transforms. The generalized orthogonal discrete W transform is a real-valued orthogonal transform, and the real-valued orthogonal transform of a real sequence has the advantages of simple operation and facilitated transmission and storage. The generalized orthogonal discrete W transforms provide more basis functions with new frequencies and phases and hence lead to more powerful analysis and processing tools for communication, signal processing, and numerical computing.

1. Introduction

The orthogonal transform is a good mathematical tool and has been used in many applications of digital signal processing, such as harmonic analysis, numerical computation, image processing, data compression, and information hiding [1–6]. Therefore, the method of constructing orthogonal transforms has become an important research topic. The orthogonal transforms include many transforms such as discrete Fourier transform (DFT), discrete Hartley transform, discrete W transform, discrete cosine transform, discrete Walsh transform, and discrete Haar transform.

The DFT is the most widely used and influential transform. Since it was proposed by Fourier in 1822, the Fourier transform has been the most basic analysis method in the analysis of continuous signal and system [7]. In order to adapt to the spectrum analysis by computer, DFT was proposed, which is the approximation of spectrum analysis for continuous time signals. Frequency domain analysis was often superior to time-domain analysis. However, it was impractical to use DFT in the spectrum analysis due to high computational complexity before the advent of the fast algorithm of DFT (FFT). In 1965, Cooley and Tukey [8] published a famous paper on the FFT, which reduced the computation time of DFT by several orders of magnitude. As a result, FFT technology has been widely used in various scientific fields. It broadly promoted the effective combination of engineering and computer technologies. Also, it realized the rapid and effective analysis and processing of engineering problems by enabling use of computers. Fourier transform has a wide range of applications in discrete dynamics, acoustics, optics, and other physics, as well as number theory, combinatorial mathematics, probability, statistics, signal processing, cryptography, and many other fields [9–11].

In order to extend the application scope of DFT, the kernel function of DFT should be generalized to construct as many basis functions with new frequencies and phases as possible.

The kernel function of the DFT is \( \exp(i(2\pi/N))^{lx} \), where \( x = 0, 1, \ldots, N-1 \), \( l = 0, 1, \ldots, N-1 \). If \( \omega^N = 1 \) and \( \omega^r \neq 1 \), where \( n < N \) (\( n \) is a positive integer), \( \omega \) is called an \( N \)-th order primitive root of unity. It is easy to verify that \( \exp(i(2\pi/N)) \) is an \( N \)-th order primitive root of unity in the
field of complex numbers \((N \geq 2)\). In fact, the base \(\exp(i(2\pi/N))\) of the kernel function can be further extended to any \(N\)-th-order primitive root of unity in the field of complex numbers, and \((l, x)\) can be extended to \((l + a, x + \beta)\), where \(a\) and \(\beta\) can be any real numbers. In this paper, we prove that if \(\omega_N\) is an \(N\)-th-order primitive root of unity in the field of complex numbers \((N \geq 2)\), then the row vectors of the transform matrix associated with the kernel function \(\omega_N^{\alpha l x + \beta}\) are orthogonal to each other, where \(l\) and \(x\) denote, respectively, the row and column indices of the transform matrix, while the \(a\) and \(\beta\) parameters can be any real numbers. The generalized discrete Fourier transform is constructed using the normalized kernel function \(1/\sqrt{N} \omega_N^{(l \alpha + \beta)}\).

However, since the DFT is a complex-valued transform, a real sequence becomes a complex sequence after DFT. Complex sequences are not as easy to transmit and store as real sequence. Therefore, a real-valued transform was studied to replace the DFT. The discrete Hartley transform (DHT) advanced by Bracewell [12] is a real-valued orthogonal transform. When the DHT is applied to real sequences, it avoids complex operations and speeds up calculations [13]. A transformed real sequence is still a real sequence, which is easy to transmit and store. The DHT and the DFT are related by a simple conversion relationship: that is, the DHT kernel function \(\cos((2\pi/N)lx) + \sin((2\pi/N)lx)\), \(\\cos((2\pi/N)lx) - \sin((2\pi/N)lx)\), \(\\sin((2\pi/N)lx) + \cos((2\pi/N)lx)\), \(\\sin((2\pi/N)lx) - \cos((2\pi/N)lx)\) are orthogonal to each other, where \(l\) and \(x\) denote, respectively, the row and column indices of the DFT kernel function, \(\\cos((2\pi/N)lx)\), \(\sin((2\pi/N)lx)\), \(\sin((2\pi/N)lx)\) are the real and imaginary parts of the DFT kernel function, \(\exp(i(2\pi/N)lx)\) is the combination of several fractional frequency and phase functions. If only integer frequency and phase basis function are used for orthogonal decomposition, the number of basis functions will be increased. The proposed fractional multiple frequency and phase basis functions provide a basis for simplifying some problems. The choice of the basis function is actually a subtle matter. For example, if the function \(y = x^2\) is expanded by the Maclaurin series in a zero-centered finite interval, only one term arises in the expansion. However, there are an infinite number of terms if the same function is expanded by a trigonometric series. From this example, we can see the importance of the basis function choice for function approximation, spectral analysis, data compression, etc. A problem can be well solved only if it uses basis functions whose frequencies and phases are appropriate for or match the problem. Based on the generalized DFT and GODWT, we have at our disposal a large number of new basis functions with rich frequency and phase information.

In this paper, we use the primitive roots of unity to construct the generalized DFT. Based on the generalized DFT, we also propose and prove the GODWT. Thus, ODWT is expanded from the original four types to infinite types. Moreover, we propose the fast algorithms for computing the generalized DFT and the GODWT. At the end of the paper, the fast algorithm of GODWT and the application of new frequency and phase basis function in communication are illustrated and given as an example. In addition, the key space of GODWT used in digital hiding technology is analyzed. In a word, our generalization provides better mathematical tools for analysis and processing in engineering fields such as communication.
2. Construction of the Generalized DFT Using Primitive Roots of Unity

We show in three stages the construction of the generalized discrete Fourier transform using primitive roots of unity in the complex number field:

(a) If \( q \) is a positive integer, \( q \) and \( N \) are relatively prime; that is, \( (q, N) = 1 \), and we can know that \( \exp (i(2\pi/N)q) \) is the \( N \)th-order primitive root of unity in the complex number field \( (N \geq 2) \).

In fact, when \( n = 1, 2, \ldots, N - 1 \), \( n \) is not divisible by \( N \), that is, \( N \nmid n \), and \( (q, N) = 1 \). Then, the product of \( q \) and \( n \) is not divisible by \( N \), that is, \( N \nmid qn \):

\[
\therefore \left[ \exp \left( \frac{2\pi i}{N} q \right) \right]^n = \exp \left( \frac{2\pi i}{N} qn \right) \neq 1,
\]

when \( n = N \),

\[
\left[ \exp \left( \frac{2\pi i}{N} q \right) \right]^n = \exp (i2\pi q) = 1.
\]

Hence, \( \exp (i(2\pi/N)q) \) is the \( N \)th-order primitive root of unity in the complex number field.

Obviously, \( \exp [i(2\pi/N)q] = [\exp (i(2\pi/N)q)]^{-1} \) is also an \( N \)th-order primitive root of unity in the complex number field.

(b) If \( \omega_N = r \cdot \exp (i\theta) \) is an \( N \)th-order primitive root of unity, then

\[
[r \cdot \exp (i\theta)]^N = 1 = \exp (i2\pi k),
\]

where \( k \) is any integer:

\[
\therefore r = 1,
\]

\[
\theta = \frac{2k\pi}{N}, \text{ then } \omega_N = \exp \left( \frac{2k\pi}{N} \right).
\]

When \( k = 0 \), \( \omega_N = 1 \), and \( \omega_N \) becomes a first-order primitive root of unity. This root has no practical significance and will not be considered.

When \( k \) is a positive integer, we get \( \omega_N = \exp (i(k2\pi/N)) \), where \( k \) and \( N \) are relatively prime. Otherwise, if \( k \) and \( N \) have a common divisor \( s \), \( s \neq 1 \), then \( k = sk_1 \), \( N = sN_1 \), and \( N_1 < N \); then

\[
(\omega_N)^{N_1} = \left[ \exp \left( \frac{k2\pi}{N} \right) \right]^{N_1} = \left[ \exp \left( \frac{k_12\pi}{N_1} \right) \right]^{N_1} = 1,
\]

Therefore, \( \omega_N \) is not an \( N \)th-order primitive root of unity. This contradicts the assumptions.

When \( k \) is a negative integer, we similarly get \( \omega_N = \exp (i(k2\pi/N)) \), where \( -k \) and \( N \) are relatively prime.

From (a) and (b), we realize that an \( N \)th-order primitive root of unity \( \omega_N \) in the complex number field can only be \( \exp (i(k2\pi/N)) \), where \( k \) is a positive or negative integer and \( |k| \) and \( N \) are relatively prime \( (N \geq 2) \).

(c) If \( \tau \) is a real number and the conjugate of \( \omega_N^\tau \) is denoted by \( \bar{\omega}_N^\tau \), then

\[
\bar{\omega}_N^\tau = \left[ \exp \left( \frac{k2\pi}{N} \right) \right]^\tau = \exp \left( \frac{-k2\pi}{N} \right) = \left[ \exp \left( \frac{k2\pi}{N} \right) \right]^{-\tau} = \omega_N^{-\tau}.
\]

Now, we prove the orthogonality of the rows of the transform matrix that is associated with the kernel function \( \omega_N^{(l_1+a)(x+b)} \), where \( l \) and \( x \) denote the row and column indices of the transform matrix, respectively. The parameters \( a \) and \( \beta \) can be any real numbers; \( \omega_N \) is \( N \)th-order primitive root of unity. For any two row vectors \( \omega_N^{(l_1+a)(x+b)} \) and \( \omega_N^{(l_2+a)(x+b)} \), where \( l_1, l_2 \in \{0, 1, \ldots, N - 1\}, x = 0, 1, \ldots, N - 1 \), denote their dot product by \( \langle \omega_N^{(l_1+a)(x+b)}, \omega_N^{(l_2+a)(x+b)} \rangle \):

\[
\langle \omega_N^{(l_1+a)(x+b)}, \omega_N^{(l_2+a)(x+b)} \rangle = \sum_{x=0}^{N-1} \omega_N^{(l_1+a)(x+b)} \bar{\omega}_N^{(l_2+a)(x+b)}
\]

\[
= \sum_{x=0}^{N-1} \omega_N^{(l_1-a)(x+b)} \bar{\omega}_N^{(l_2-a)(x+b)}
\]

\[
= \sum_{x=0}^{N-1} \omega_N^{(l_1-l_2)(x+b)}
\]

\[
= \omega_N^{(l_1-l_2)N} \sum_{x=0}^{N-1} \omega_N^{(l_1-l_2)x}.
\]
when \( l_1 \neq l_2 \), \( \omega_N^{(l_1-l_2)} \neq 1 \), and \( (\omega_N)^N = 1 \); therefore,

\[
\omega_N^{(l_1-l_2)} \sum_{x=0}^{N-1} \omega_N^{(l_1-l_2)x} = \omega_N^{(l_1-l_2)} - \omega_N^{(l_1-l_2)N} = 0. \tag{8}
\]

Thus, the row vectors of the transform matrix associated with the kernel function \( \omega_N^{(x+y)} \) are orthogonal to each other.

When \( l_1 = l_2 \),

\[
\omega_N^{(l_1-l_2)} \sum_{x=0}^{N-1} \omega_N^{(l_1-l_2)x} = N. \tag{9}
\]

The modulus of each row vector is \( \sqrt{N} \).

If \([U]\) is a complex matrix of order \( N \) and \([U][U] = [U]^* [U] = [E]\), \([U]\) is called unitary matrix, where \([U]^*\) is the transposed conjugate matrix of \([U]\), and \([E]\) is a unit matrix. Therefore, the \( N \times N \) matrix generated by the kernel function \( (1/\sqrt{N}) \omega_N^{(l+a)(x+y)} = (1/\sqrt{N}) \exp(\pm i(2\pi/N)q(l+a)(x+y)) \) is a unitary matrix, where \( l, x = 0, 1, \ldots, N-1, \) and each of the parameters \( a \) and \( \beta \) can be any real number. The kernel function \( (1/\sqrt{N}) \omega_N^{(l+a)(x+y)} \) lies at the \( l \)-th row and the \( x \)-th column of the unitary matrix.

When \( a = -(2l + \alpha) \),

\[
\frac{1}{\sqrt{N}} \exp \left[ \frac{2\pi}{N} q(l+a)(x+y) \right] = \frac{1}{\sqrt{N}} \exp \left[ -\frac{2\pi}{N} q(l+a')(x+y) \right]. \tag{10}
\]

Therefore, only the kernel function \( (1/\sqrt{N}) \exp(i(2\pi/N)q(l+a)(x+y)) \) constructed by the primitive root of unity \( \omega_N = \exp(i(2\pi/N)q) \) should be considered.

The transform with the kernel function \( 1/\sqrt{N} \exp(i(2\pi/N)q(l+a)(x+y)) \) is expressed as

\[
F(l) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \omega_N^{(l+a)(x+y)} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \exp \left[ \frac{2\pi}{N} q(l+a)(x+y) \right], \text{ where, } l = 0, 1, \ldots, N-1. \tag{11}
\]

### 3. Construction of the GODWT

The DWT with two parameters, \( a \) and \( \beta \), is defined as follows:

\[
F(l) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \sin \left[ \frac{\pi}{4} + \frac{2\pi}{N} (l+a)(x+y) \right] = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \cos \left[ \frac{2\pi}{N} (l+a)(x+y) \right]. \tag{13}
\]

DWT-I, DWT-II, DWT-III, and DWT-IV, respectively, and we call them ODWT.

The parameter \( q \) can be added to the expression in (13) to obtain

\[
F(l) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \cos \left[ \frac{2\pi}{N} q(l+a)(x+y) \right], \text{ where, } (x = 0, 1, \ldots, N-1), (l = 0, 1, \ldots, N-1). \tag{14}
\]

\[
\sum_{x=0}^{N-1} \sin \frac{2\pi}{N} pm(x+y) = 0. \tag{15}
\]

**Lemma 1.** If \( p \) is positive integer \( (p, N) = 1 \), \( \beta = (h/2p) \), and both \( h \) and \( m \) are integers, then

**Proof:** If \( N \mid m \), \( (p, N) = 1 \), \( \cdot \cdot \cdot N \mid pm \), \( \exp(i(2\pi/N)pm) \neq 1 \),
\[
\therefore 1 - \exp \left( i \frac{2\pi}{N} pm \right) \neq 0, \tag{16}
\]

\[
\left[ 1 - \exp \left( i \frac{2\pi}{N} pm \right) \right] \sum_{x=0}^{N-1} \exp \left( i \frac{2\pi}{N} pxm \right) = 1 - \exp (i2\pi pm) = 0.
\tag{17}
\]

\[
\therefore \sum_{x=0}^{N-1} \exp \left[ i \frac{2\pi}{N} p(x + \beta) \right] = \exp \left( i \frac{2\pi}{N} pm\beta \right) \cdot \sum_{x=0}^{N-1} \exp \left( i \frac{2\pi}{N} pxm \right) = 0, \therefore \sum_{x=0}^{N-1} \sin \frac{2\pi}{N} pm(x + \beta) = 0. \tag{18}
\]

If \( m \) is divisible by \( N \), that is, \( N|m, m = Nd \),

\[
\sum_{x=0}^{N-1} \sin \frac{2\pi}{N} pxm = \sum_{x=0}^{N-1} \sin \left( \frac{\pi}{N} \right) 2pN d \left( x + \frac{h}{2p} \right) = \sum_{x=0}^{N-1} \sin \pi d \left( 2px + h \right) = 0. \tag{19}
\]

Q.E.D.

\[\square\]

**Lemma 2.** A real or complex sequence \( f(x) \), where \( x = 0, 1, \ldots, N - 1 \), can be regarded as a function defined on the set \( J_N = \{0, 1, \ldots, N - 1\} \). A finite Abelian group can be formed through modulo-\( N \) addition on \( J_N \). This group is called a residual-class additive group and is denoted by \( Z_N \).

Next, we derive below the conditions for constructing a real-valued orthogonal system by adding the real and imaginary parts of a complex-valued orthogonal system together.

Let the complex-valued function set on \( Z_N \), that is,

\[
\{ \Phi_k(x) = R_k(x) + iN_k(x) | k = 0, 1, \ldots, N - 1 \}, \tag{20}
\]

be a normalized orthogonal function system, where \( R_k(x) \) and \( N_k(x) \) are the real and imaginary parts of \( \Phi_k(x) \), respectively. Then, the condition for the real-valued function set,

\[
R_{k_1}(x)N_{k_2}(x) = \sum_{n=0}^{N-1} R_{k_2}(x)N_{k_1}(x), \tag{25}
\]

\[
\sum_{x=0}^{N-1} \Phi_k(x)\Phi_k(x) = \sum_{x=0}^{N-1} R_k(x)\overline{R_k(x)} + \sum_{x=0}^{N-1} (R_k(x)N_k(x) + R_k(x)N_k(x)) + i \sum_{x=0}^{N-1} (R_k(x)N_k(x) - R_k(x)N_k(x)). \tag{26}
\]

From (25) and (26), we can get

\[
\sum_{x=0}^{N-1} \Psi_k(x)\Psi_k(x) = \sum_{x=0}^{N-1} \Phi_k(x)\overline{\Phi_k(x)} + 2 \sum_{x=0}^{N-1} R_k(x)N_k(x). \tag{27}
\]

From (16) and (17), we get \( \sum_{x=0}^{N-1} \exp (i2\pi pmx) = 0 \). And,

\[
\sum_{x=0}^{N-1} \sin \frac{2\pi}{N} pm(x + \beta) = 0. \tag{18}
\]

\[
\{ \Psi_k(x) = R_k(x) + N_k(x) | k = 0, 1, \ldots, N - 1 \}, \tag{21}
\]

to be a normalized orthogonal function system is

\[
\sum_{x=0}^{N-1} R_{k_1}(x)N_{k_2}(x) = 0, \tag{22}
\]

where \( k_1, k_2 \in \{0, 1, \ldots, N - 1\} \).

**Proof.** \( \{ \Phi_k(x) \} \) is a normalized orthogonal system. Hence, for any \( k_1, k_2 \in \{0, 1, \ldots, N - 1\} \),

\[
\text{Im} \left[ \sum_{x=0}^{N-1} \Phi_k(x)\overline{\Phi_k(x)} \right] = 0, \tag{23}
\]

which can be rewritten as

\[
\sum_{x=0}^{N-1} (-R_k(x)N_k(x) + R_k(x)N_k(x)) = 0. \tag{24}
\]

Hence, we find that

From (27), we find that, for any \( k_1, k_2 \in \{0, 1, \ldots, N - 1\} \), the condition \( \sum_{x=0}^{N-1} R_{k_1}(x)N_{k_2}(x) = 0 \) is the necessary and sufficient condition for \( \{ \Psi_k(x) = R_k(x) + N_k(x), k = 0, 1, \ldots, N - 1 \} \) to be a real normalized orthogonal function system. \( \square \)
3.1. The Generalized Orthogonal Discrete W Transform.

The unitary matrix with the kernel function \((1/\sqrt{N})\exp[i(2\pi/N)q(l + \alpha)(x + \beta)]\) is constructed as discussed above. The \(N\) row vectors of an \(N \times N\) unitary matrix form a normalized orthogonal system. The sum of the real and imaginary parts of the kernel function \((1/\sqrt{N})\exp[i(2\pi/N)q(l + \alpha)(x + \beta)]\) is given as

\[
\frac{1}{\sqrt{N}} \left\{ \cos \left[ \frac{2\pi}{N} q(l + \alpha)(x + \beta) \right] + \sin \left[ \frac{2\pi}{N} q(l + \alpha)(x + \beta) \right] \right\} = \frac{1}{\sqrt{N}} \cos \left[ \frac{2\pi}{N} q(l + \alpha)(x + \beta) \right],
\]

and this sum is used as the kernel function of a new real matrix. According to Lemma 2, we know that the necessary and sufficient conditions for the orthonormality of the \(N\) row vectors of the real matrix can be formulated as

\[
\frac{1}{N} \sum_{x=0}^{N-1} \cos \left[ \frac{2\pi}{N} q(l_1 + \alpha)(x + \beta) \right] \sin \left[ \frac{2\pi}{N} q(l_2 + \alpha)(x + \beta) \right] = 0, \quad \text{for any} \ l_1, l_2 \in \{0, 1, \ldots, N - 1\},
\]

Let

\[
\sigma(\alpha, \beta) = \frac{1}{N} \sum_{x=0}^{N-1} \cos \left[ \frac{2\pi}{N} q(l_1 + \alpha)(x + \beta) \right] \sin \left[ \frac{2\pi}{N} q(l_2 + \alpha)(x + \beta) \right].
\]

Using the product-to-sum formula (i.e., the Prosthaphaeresis formula), from (30), we get

\[
\sigma(\alpha, \beta) = \frac{1}{2N} \sum_{x=0}^{N-1} \left\{ \sin \left[ \frac{2\pi}{N} q(l_1 + l_2 + 2\alpha)(x + \beta) \right] - \sin \left[ \frac{2\pi}{N} q(l_1 - l_2)(x + \beta) \right] \right\}.
\]

If \(q = q_1q_2\), both \(q_1\) and \(q_2\) are positive integers, \(\vdash (q, N) = 1, \vdash (q_2, N) = 1\); let \(\alpha = (g/2q_1)\) and \(\beta = (h/2q_2)\), where \(g\) and \(h\) can be any integers; by substitution in (31), we get

\[
\sigma \left( \frac{g}{2q_1}, \frac{h}{2q_2} \right) = \frac{1}{2N} \sum_{x=0}^{N-1} \left\{ \sin \left[ \frac{2\pi}{N} q_2q_1 (l_1 + l_2 + 2g/2q_1) \right] \left( x + \frac{h}{2q_2} \right) \right\} - \sin \left[ \frac{2\pi}{N} q_2q_1 (l_1 - l_2) \left( x + \frac{h}{2q_2} \right) \right],
\]

\(\vdash q_1(l_1 + l_2 + 2\alpha) = q_1(l_1 + l_2 + 2(g/2q_1)) = q_1(l_1 + l_2) + g\) is an integer and \(q_1(l_1 - l_2)\) is also an integer.

According to Lemma 1, we get

\[
\sigma \left( \frac{g}{2q_1}, \frac{h}{2q_2} \right) = \frac{1}{2N} \sum_{x=0}^{N-1} \left\{ \sin \left[ \frac{2\pi}{N} q_2q_1 (l_1 + l_2 + 2g/2q_1) \right] \left( x + \frac{h}{2q_2} \right) \right\} - \sin \left[ \frac{2\pi}{N} q_2q_1 (l_1 - l_2) \left( x + \frac{h}{2q_2} \right) \right] = 0,
\]
Then, when \( a = (g/2q_1) \) and \( b = (h/2q_2) \), equation (29) holds, and the \( N \) functions,

\[
\frac{1}{\sqrt{N}} \text{cas} \left[ \frac{2\pi}{N} q(l + \alpha)(x + \beta) \right] = \frac{1}{\sqrt{N}} \text{cas} \left[ \frac{2\pi}{N} q \left( l + \frac{g}{2q_1} \right) \left( x + \frac{h}{2q_2} \right) \right], \quad \text{where } (l = 0, 1, \ldots, N - 1),
\]

form a normalized orthogonal basis, where the independent variable \( x = 0, 1, \ldots, N - 1 \).

The real-valued orthogonal matrix is obtained with the kernel functions in (34), where \( l \) and \( x \) are the row and column indices of the matrix. The transform

\[
F(l) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \text{cas} \left[ \frac{2\pi}{N} q \left( l + \frac{g}{2q_1} \right) \left( x + \frac{h}{2q_2} \right) \right], \quad \text{where } (l = 0, 1, \ldots, N - 1),
\]

is GODWT, and its inverse transform is

\[
f(x) = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} F(l) \text{cas} \left[ \frac{2\pi}{N} q \left( l + \frac{g}{2q_1} \right) \left( x + \frac{h}{2q_2} \right) \right], \quad \text{where } (l = 0, 1, \ldots, N - 1).
\]

If \( s_1 \) is the factor of \( 2q_1 \) and \( s_2 \) is the factor of \( 2q_2 \), we get \( 2q_1 = t_1s_1 \) and \( 2q_2 = t_2s_2 \). So, parameter \( \alpha \) of (28) and (29) becomes \( \alpha = (g/2q_1) = (g/t_1s_1) \). Similarly, parameter \( \beta \) of (28) and (29) becomes \( \beta = (h/2q_2) = (h/t_2s_2) \). Then, we set \( g = t_1g_1 \) and \( h = t_2h_2 \), where \( g_1 \) and \( h_2 \) can be any integers. Finally, we obtain \( \alpha = (g_1/s_1) \) and \( \beta = (h_2/s_2) \). Thus, the \( N \) row vectors of the real matrix with the kernel functions \( (1/\sqrt{N}) \text{cas} \left[ \frac{2\pi}{N} q \left( l + (g_1/s_1) \right) \left( x + (h_2/s_2) \right) \right] \) form a normalized orthogonal basis. If you change \( g_1 \) and \( h_2 \) into \( g \) and \( h \), or any other two letters, the new letters express the same settings as long as they are any integers.

Let \( \alpha = (g/s_1) \) and \( \beta = (h/s_2) \), where \( g \) and \( h \) can be any integers; if \( q = q_1q_2 = 1 \), the factors \( s_1 \) of \( 2q_1 \) and \( s_2 \) of \( 2q_2 \) can only be taken as \( 1 \) or \( 2 \). When \( (\alpha, \beta) \in \{(0, 0), ((1/2), 0), (0(1/2)), ((1/2)(1/2))\} \), the four transforms are given by the kernel function \( (1/\sqrt{N}) \text{cas} \left[ \frac{2\pi}{N} q \left( l + (g/s_1) \right) \left( x + (h/s_2) \right) \right] \), and it is obvious that they are DW1-1, DW1-2, DW1-3, and DW1-4, respectively, that is, ODWT.

For the GODWT, when \( q = 1 \), it reduces to the ODWT, and when \( q \neq 1 \), a new family of orthogonal transforms arises. This family greatly expands the ODWT and also subserves a family of symmetric transforms.

If \( q \) contains a factor \( s^2 \), we can construct a transform matrix such that \( \alpha = (f/2s) \) and \( \beta = (f/2s) \), where \( f \) can be any integer. This transform matrix includes two special cases, namely, \( (\alpha = (f/2), (\beta = (f/2) \) and \( (\alpha = (f/s), (\beta = (f/s) \). In these cases, the real-valued orthogonal matrix is a symmetric matrix, and the forward and inverse transforms have the same form.

### 4. Fast Algorithms for the Generalized Discrete Fourier and GODWT

#### 4.1. The Fast Algorithm for the Generalized Discrete Fourier Transform

Let \( f(x) \) be a sequence, where \( x = 0, 1, \ldots, N - 1 \), and let \( \omega_N \) be an \( N^{th} \) order primitive root of unity in the complex number field. Then, the transform of \( f(x) \) by the kernel function \( (1/\sqrt{N}) \omega_N^{(l+x)(x+b)} \) (the generalized DFT) is expressed as

\[
F(l) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \omega_N^{(l+x)(x+b)} = \left\{ \begin{array}{ll}
\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} [f(x)\omega_N^{ax}]\omega_N^{bx} & \text{if } (l = 0, 1, \ldots, N - 1),
\end{array} \right.
\]

From (37), we can identify two steps for the transform of \( f(x) \) with the kernel function \( (1/\sqrt{N}) \omega_N^{(l+x)(x+b)} \):

(1) When \( N \) is a composite number (nonprime, \( N \geq 2 \),

\[ f(x)\omega_N^{ax} \]

is transformed by the kernel function
(1/√N)ω_N^{lα}. This is adopted for the generalized fast Fourier transform (FFT) [35].

(2) The result from the first step is multiplied by ω_N^{l(α+β)}.

So, when N is a composite number, there is a fast algorithm for computing the transform of f(x) with the kernel function (1/√N)ω_N^{l(α+β)}.

\[
f(x) = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} F(l) \left[ \omega_N^{-1} \right]^{l(α+β)} = \frac{1}{\sqrt{N}} \sum_{l=0}^{N-1} F(l) \omega_N^{l(α+β)}, \quad \text{where } (l = 0, 1, \ldots, N-1).
\]  

4.2. The Fast Algorithm for the GODWT. Let [Φ] be a unitary matrix: [Φ] = [R] + i[N], where [R] and [N] are real matrices and [Ψ] = [R] + [N] forms an orthogonal matrix. Then, the inverse transform matrix of [Φ] is its conjugate transposed matrix [Φ]^T = [R]^T i[N]^T. The inverse transform matrix of [Ψ] is its transposed matrix [Ψ]^T = [R]^T + [N]^T.

If [L] is a real-valued column vector, then the actions of the matrices on this vector are as follows:

\[
[Φ][L] = ([R] + i[N])[L] = [R][L] + i[N][L], \quad (39)
\]

\[
[Ψ][L] = ([R] + [N])[L] = [R][L] + [N][L], \quad (40)
\]

\[
[Φ]^T[L] = ([R]^T i[N])[L] = [R]^T[L] - i[N]^T[L], \quad (41)
\]

\[
[Ψ]^T[L] = ([R]^T + [N]^T)[L] = [R]^T[L] + [N]^T[L], \quad (42)
\]

\[
F(l) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \cos \left[ \frac{2\pi}{N} q \left( l + \frac{g}{2q_1} \right) \left( x + \frac{h}{2q_2} \right) \right] = \frac{\sqrt{2}}{4} \sum_{x=0}^{7} f(x) \cos \left[ \frac{2\pi}{8} \left( l + \frac{1}{6} \right) \left( x + \frac{1}{6} \right) \right], \quad \text{where } f(x) \text{ is the real sequence.}
\]  

From (39) and (40), we see that [Ψ][L] is the sum of the real and imaginary parts of [Φ][L]. Moreover, from (41) and (42), we see that [Ψ]^T[L] is the real part minus the imaginary part of [Φ]^T[L].

For forward and inverse transforms of the GODWT, the transformed sequences are all real-valued sequences. Using the above relations of [Ψ][L] and [Φ][L] as well as [Ψ]^T[L] and [Φ]^T[L], we can obtain fast algorithms for computing the GODWT and its inverse transform. However, the advantages of real-valued operations can be brought into play only by developing direct fast algorithms of real-valued orthogonal transforms on real sequences. This is one of the problems in the anticipated future work on the GODWT.

Example 1. Let N = 8, q = 9, q_1 = 3, q_2 = 3, g = 1, and h = 1. Therefore, a = (g/2q_1) = (1/6), and β = (h/2q_2) = (1/6). The transform is

\[
F''(l) = \frac{\sqrt{2}}{4} \sum_{x=0}^{7} f(x) \omega^{l(1/6)}(x+1/6) = \left\{ \sum_{x=0}^{7} f(x) \omega^{l(1/6)x} \right\} \omega^{l(1/6)1/6}. \quad (44)
\]

In order to get F''(l), first, we calculate f'(x), f'(x) = f(x)ω^{l(1/6)x}, where x = 0, 1, ..., 7. Second, f''(x) is transformed with the kernel function ω^{l(1/6)x} into F''(l); then we calculate F''(l), F''(l) = (√2/4)F'(l)ω^{l(1/6)1/6}, where l = 0, 1, ..., 7. We get F(l) after the real and imaginary parts of F''(l) are added.

The matrix [Φ] is formed with the kernel function ω^{l(1/6)x}, x = 0, 1, ..., 7, l = 0, 1, ..., 7. It can be decomposed into the
product of a sparse matrix according to the method in [35], so it has a fast algorithm [\[\Phi\]] = [\[F_1\]][\[F_2\]][\[F_3\]],

\[
[\Phi] = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \omega & \omega^2 & \omega^3 & \omega^4 & \omega^5 & \omega^6 & \omega^7 \\
1 & \omega^2 & \omega^4 & \omega^6 & 1 & \omega^2 & \omega^4 & \omega^6 \\
1 & \omega^3 & \omega^6 & \omega^4 & \omega^7 & \omega^2 & \omega^5 & 1 \\
1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 & 1 & \omega^4 \\
1 & \omega^5 & \omega^2 & \omega^7 & \omega^4 & \omega^6 & \omega^3 & 1 \\
1 & \omega^6 & \omega^4 & \omega^2 & \omega^4 & \omega^6 & \omega^3 & 1 \\
1 & \omega^7 & \omega^6 & \omega^5 & \omega^4 & \omega^3 & \omega^2 & \omega
\end{bmatrix}
\] (45)

where \(\omega^4 = -1\).

\[
[\Phi] = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & \omega^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^3 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & \omega^6 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^6
\end{bmatrix}
\] (46)

\[
[\Phi] = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & \omega^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^3 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & \omega^6 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & \omega^6
\end{bmatrix}
\]

By using the sparse matrix, it is easy to draw a signal flow diagram that a column vector is transformed by the matrix [\[\Phi\]]. The FFT flow diagram is depicted in Figure 1.

We calculate the inverse transform of the transform associated with the kernel \((\sqrt{2}/4)\omega^{(1+(1/6))}(x+1/6)\) as follows:

\[
f_2(x) = \frac{\sqrt{2}}{4} \sum_{l=0}^{7} F(l)\omega^{-(1+(1/6))(x+1/6)} = \left\{ \sum_{l=0}^{7} [F(l)\omega^{(1/6)}] \omega^x \right\} \frac{\sqrt{2}}{4} \omega^{(x+1/6)(1/6)},
\] (47)

where \(\overline{\omega}\) is the conjugate of \(\omega\).

In order to get \(f_2(x)\), first, \(F_1(l)\) is computed as \(F_1(l) = F(l)\overline{\omega}^{(1/6)}\). Then, the FFT of \(F_1(l)\) is computed using a kernel function \(\overline{\omega}^{1/6}\). The transform matrix \([\Phi]\) formed by kernel function \(\overline{\omega}^{1/6}\) is the conjugate transpose matrix \([\Phi]^\ast\) of \([\Phi]\). \([\Phi]\) is the symmetric matrix. So \([\Phi] = [\Phi]^\ast\). In the product of sparse matrix of \([\Phi]\), that is, \([F_1][F_2][F_3]\) and its corresponding signal flow diagram, \(\omega\) is replaced with \(\overline{\omega}\), the product of the sparse matrix decomposed by \([\Phi]\) and its corresponding signal flow diagram can be obtained. The input and output of the signal flow diagram are \(F_1(l)\) and \(f_1(x)\), respectively. Then, given \(f_2(x) = (\sqrt{2}/4)f_1(x)\omega^{(x+1/6)(1/6)}\), \(f(x)\) is obtained using the real part minus the imaginary part of \(f_2(x)\).

Another method of computing \(\sum_{l=0}^{7} [F(l)\omega^{(1/6)}] \omega^x\) is as follows:

\[
\sum_{l=0}^{7} [F(l)\omega^{(1/6)}] \omega^x = \sum_{l=0}^{7} [F(l)\omega^{(1/6)}] \omega^x,
\]

in order to calculate the FFT of \(F(l)\omega^{(1/6)}\) with kernel function \(\omega^x\), we can firstly calculate the FFT of \(F(l)\omega^{(1/6)}\) with kernel function \(\omega^x\) and then take the conjugate. The product of sparse matrix and signal flow graph of \(\Phi\) can be directly used for FFT with kernel function \(\omega^x\).

In this example, the transform \(F(l) = (\sqrt{2}/4)\sum_{x=0}^{7} f(x)\omega^{[(2\pi \times 9/8)(l + (1/6))(x + (1/6))]\}}\) is symmetric transform; thus, the inverse transform is the same as the positive transform. The inverse transform is

\[
f(x) = \frac{\sqrt{2}}{4} \sum_{l=0}^{7} F(l)\omega^{\left[\frac{2\pi \times 9}{8} \left( l + \frac{1}{6} \right) \left( x + \frac{1}{6} \right) \right]}.
\] (48)

Therefore, we can use the same fast algorithm as the positive transform.
4.3. Examples of the GODWT Application

(1) In communication field, the communication between different users using different frequency bands is frequency division multiplexing, and the communication between different users relying on different address codes is code division multiplexing. The code division multiplexing and frequency division multiplexing can be combined by using GODWT. We still take the primitive unit root $\omega = \exp[i(2\pi \times 9/8)]$, and three kernel functions of GODWT, that is, $\{\sqrt{2}/4\}a(1)\cos[(2\pi \times 9/8)y]$, $\{\sqrt{2}/4\}a(7)\cos[(2\pi \times 9/8)(x + 1/3)y]$, and $\{\sqrt{2}/4\}a(x)\cos[(2\pi \times 9/8)(x + 2/3)y]$ can be obtained. For the three transmitted code sequences, that is, $a(x)$, $b(x)$, and $c(x)$, $x = 0, 1, \ldots, 7$, the transforms of them are calculated, respectively, to get $A(l)$, $B(l)$, and $C(l)$, where $l = 0, 1, \ldots, 7$.

$$A(l) = \frac{\sqrt{2}}{4} \sum_{x=0}^{7} a(x)\cos\left[\frac{2\pi \times 9}{8}xl\right],$$

$$B(l) = \frac{\sqrt{2}}{4} \sum_{x=0}^{7} b(x)\cos\left[\frac{2\pi \times 9}{8}(x + 1/3)l\right],$$

$$C(l) = \frac{\sqrt{2}}{4} \sum_{x=0}^{7} c(x)\cos\left[\frac{2\pi \times 9}{8}(x + 2/3)l\right],$$

$$A(l) = \{\sqrt{2}/4\} \sum_{x=0}^{7} a(x)\cos[(2\pi \times 9)/8]xl; \quad \text{the equation is written in matrix forms as}$$

$$[A(0), A(1), \ldots, A(7)] = \frac{\sqrt{2}}{4} [a(0), a(1), \ldots, a(7)]$$

Let the carrier of code division multiplexing be $Z_{10}, Z_{11}, \ldots, Z_{17}$:

$$Z_{10} = \begin{bmatrix} \cos\left[\frac{9\pi}{4}0\times0\right] & \cos\left[\frac{9\pi}{4}0\times1\right] & \ldots & \cos\left[\frac{9\pi}{4}0\times7\right] \end{bmatrix}, \quad \text{the angular frequency of } Z_{10} \text{ is } \frac{9\pi}{4} \times 0,$$

$$Z_{11} = \begin{bmatrix} \cos\left[\frac{9\pi}{4}1\times0\right] & \cos\left[\frac{9\pi}{4}1\times1\right] & \ldots & \cos\left[\frac{9\pi}{4}1\times7\right] \end{bmatrix}, \quad \text{the angular frequency of } Z_{11} \text{ is } \frac{9\pi}{4} \times 1,$$

$$Z_{17} = \begin{bmatrix} \cos\left[\frac{9\pi}{4}7\times0\right] & \cos\left[\frac{9\pi}{4}7\times1\right] & \ldots & \cos\left[\frac{9\pi}{4}7\times7\right] \end{bmatrix}, \quad \text{the angular frequency of } Z_{17} \text{ is } \frac{9\pi}{4} \times 7$$

The carriers $Z_{10}, Z_{11}, \ldots, Z_{17}$ are periodic and repetitive. The amounts of information carried by each carrier wave over a period are constant, which are $\{\sqrt{2}/4\}a(0)$, $\{\sqrt{2}/4\}a(1)$, $\{\sqrt{2}/4\}a(7)$, respectively.

$$[A(0), A(1), \ldots, A(7)] = \frac{\sqrt{2}}{4} [a(0)Z_{10} + a(1)Z_{11} + \cdots + a(7)Z_{17}].$$
In order to transmit \( A(l) = (\sqrt{2}/4) \sum_{x=0}^{7} a(x) \text{cas} \left[ \frac{2\pi x \times 9}{8} \times l \right], l = 0, 1, \ldots, 7 \), the angular frequency of carriers is from \( (9\pi/4) \times 0 \) to \( (9\pi/4) \times 7 \). It is called code division multiplexing where we have transmitting \( A(l), l = 0, 1, \ldots, 7 \) instead of transmitting \( a(x), x = 0, 1, \ldots, 7 \). Both the input and output are eight real numbers. In contrast, if we use the DFT, the output is eight, \( \frac{\sqrt{2}}{4} \sum_{x=0}^{7} (a(x) \text{cas} \left[ \frac{2\pi x \times 9}{8} \right] + \frac{\sqrt{2}}{4} \sum_{x=0}^{7} b(x) \text{cas} \left[ \frac{2\pi x \times 9}{8} (x \times \frac{1}{3}) \right] + \frac{\sqrt{2}}{4} \sum_{x=0}^{7} c(x) \text{cas} \left[ \frac{2\pi x \times 9}{8} (x \times \frac{2}{3}) \right] \right). \]

These three components are separated with a filter, and then \( a(x)b(x)\text{cas}(x) \) were restored using the inverse GODWT.

Time-division multiplexing (TDM) is also a method to transmit multiway signals. Time-division multiplexing is used primarily for digital signals but may be applied in analog multiplexing in which two or more signals or bit streams are transferred appearing simultaneously as subchannels in one communication channel but are physically taking turns on the channel. Each signal appears on the line only a fraction of time in an alternating pattern. The time domain is divided into several recurrent time slots of fixed length.

Transmitting \( A(l), B(l), \) and \( C(l) \) with different time slots, they can be separated at the receiving end, and then \( a(x), b(x), \) and \( c(x) \) were restored using the inverse GODWT. This realizes the combination of code-division multiplexing and time-division multiplexing. It shows that the proposed GODWT provides more means for the transmission in communication, compared with DHT and ODWT.

In the digital hiding technology, the password can be embedded into transform coefficients.

The secret key space of ODWT is \( (N, a, \beta) \), where \( (a, \beta) = \{(0, 0), (1/2, 0), (0, 1/2), (1/2, 1/2)\}, N \) is a positive integer. Actually, \( N \) cannot be big; if \( N \) is bigger, it is not easy to calculate. The secret key space of GODWT is \( (N, q, a, \beta) \). It includes the secret key space of ODWT. Because the value of \( q \) is not limited in principle and the parameter value pairs \( (a, \beta) \) changed from 4 pairs of ODWT to infinity pairs of GODWT: \( (a, \beta) = \{(g/2q_1), (h/2q_2)\} \) where \( q_1, q_2 = g, h \) and \( c \) can be any integers, the secret key space of GODWT has been greatly generalized.

5. Conclusions

The kernel function of the generalized DFT is \( (1/\sqrt{N}) \exp[i(2\pi/N)q(l + a)(x + \beta)] \), where \( q, N = 1 \), and parameters \( a \) and \( \beta \) can be any real numbers. When \( l = 0, 1, \ldots, N-1 \), these \( N \) functions form a complex-valued normalized orthogonal system.

For the kernel function \( (1/\sqrt{N}) \exp[i(2\pi/N)q(l + a)(x + \beta)] \) of the generalized DFT, the real and imaginary parts are added together leading to the functions: \( (1/\sqrt{N}) \exp[i(2\pi/N)q(l + a)(x + \beta)] \), where \( q, N = 1 \), and \( l = 0, 1, \ldots, N-1 \). If \( q = q_1, q_2 \), both \( q_1 \) and \( q_2 \) are positive integers; let \( a = (g/2q_1), \beta = (h/2q_2) \), where \( g \) and \( h \) are any integers. These \( N \) functions form a real normalized orthogonal system, where the independent variable \( x = 0, 1, \ldots, N-1 \). The transform with the kernel function \( (1/\sqrt{N}) \exp[i(2\pi/N)q(l + a)(x + \beta)] \) is GODWT. For setting the values of \( a \) and \( \beta \), if the factor \( \mu_1 \) of \( 2q_1 \) replaces \( 2q_2 \), and the factor \( \mu_2 \) of \( 2q_2 \) replaces \( 2q_2 \), the conclusion is still valid.

When \( N \) is a composite number, based on the fast algorithms of the generalized DFT and its inverse transform, we get the fast algorithm of the GODWT and its inverse transform.

GODWT can replace the generalized DFT in wide applications. The transformed signal of the real sequence is still of real values, which makes the transmission and storage convenient. Also, GODWT provides a large number of basis functions with a new frequency, phase, and a large number of new transforms and subsumes a family of symmetric transforms; their forward and reverse transformations can be implemented by the same computer programs or hardware.

It can be seen that GODWT can provide more transmission means for communication, better security for digital hiding technology, and many new methods for data
compression. Hence, it can be used as a more powerful analysis and processing tool for communication, signal processing, and numerical computing.

**Abbreviations**

\[ N, n, q, q_1, q_2, p, s, t: \] Positive integers

\[ d, f, g, g_1, h, h_2, k, m: \] Integers

\[ \alpha, \beta, \tau: \] Real numbers

\[ \omega: \] Complex numbers

\[ [\Phi], [\Psi], [R], [N], [F_1], [F_2], [F_3]: \] The matrices

\[ [L]: \] Real-valued column vector

\[ \omega_1: \] Nth-order primitive root of unity in the complex number field

\[ \bar{\omega}: \] Conjugate of \( \omega \)

\[ \overline{\Phi}: \] Conjugate matrix of the matrix \( \Phi \)

\[ \Phi^T: \] Transposed matrix of the matrix \( \Phi \)

\[ \overline{\Phi}^T: \] Conjugate transposed matrix of the matrix \( \Phi \).

**Data Availability**

The data used to support the findings of this study are included within the article.

**Conflicts of Interest**

The authors declare that there are no conflicts of interest regarding the publication of this paper.

**Acknowledgments**

This work was supported by the projects of the Chinese Natural Science Foundation (Grant no. 61865002), Anhui Province Higher Education Promotion Plan (Grant no. TSK2017B30), the Natural Science Fund of Anhui Province (Grant no. 108085ME128), the Natural Science Foundation of Education Department of Anhui Province (Grant no. KJ2018A0123), and Shanghai Key Laboratory of Materials Laser Processing and Modification (Grant no. MLPM2017-3).

**References**

[1] H. Andrews, "Orthogonal transforms for digital signal processing," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 24, no. 5, 438 pages, 1976.

[2] D. Puchala, "Involuntary parametric orthogonal transforms of cosine-walsh type with application to data encryption," Advances in Intelligent Systems and Computing, vol. 689, 2017.

[3] Y. Önal and Ü. Ç. Turhal, "The orthogonal Hilbert-Huang transform application in voltage flicker analysis," in Proceedings of the 4th International Conference on Power Engineering, Energy and Electrical Drives, pp. 700–704, Istanbul, Turkey, May 2013.

[4] A. Zemliachenko, V. Lukin, and M. Alhihi, "Analysis of approaches to hyperspectral data compression based on orthogonal transforms," in Proceedings of the 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), pp. 909–914, Slavskoe, Ukraine, February 2018.

[5] R. Ezhilarasi and K. Venkatalakshmi, "Low complexity orthogonal transforms for low cost image/video codec design," Journal of Computational and Theoretical Nanoscience, vol. 15, no. 3, pp. 895–869, 2018.

[6] L. Cheng, Z. Jiang, and Z. Zhang, "The generalized unified computation of multidimensional discrete orthogonal transforms," Science in China Series: Information Sciences, vol. 44, pp. 401–411, 2001.

[7] A. G. Kjelaas, "The fourier transform: properties and applications," Surveillance of Environmental Pollution and Resources by Electromagnetic Waves, vol. 45, pp. 329–351, 1978.

[8] J. W. Cooley and J. W. Tukey, "An algorithm for the machine computation of complex fourier series," Mathematics of Computation, vol. 19, pp. 297–301, 1965.

[9] R. Brenner, "Numerical computation of the response of piezoelectric composites using fourier transform," Physical Review B Condensed Matter, vol. 79, no. 18, Article ID 184106, 2009.

[10] R. Bracewell, "The Fourier transform and its applications," American Journal of Physics, vol. 34, 2002.

[11] W. Zhou, X. Luo, B. Chen, Y. Zhang, and X. Cai, "Estimation of particle depth from two defocused images using the Fourier transform," Pattercogy, vol. 49, pp. 48–54, 2020.

[12] R. N. Bracewell, "Discrete Hartley transform," Journal of the Optical Society of America, vol. 73, no. 12, pp. 1832–1835, 1983.

[13] H. S. Hou, "The fast Hartley transform algorithm," IEEE Transactions on Computers, vol. 36, no. 2, pp. 147–156, 1987.

[14] X. Ouyang, J. Jin, G. Jin, and Z. Wang, "Low complexity discrete Hartley transform preconded OFDM for peak power reduction," Electronics Letters, vol. 48, no. 2, pp. 90–91, 2012.

[15] Z. Wang and B. R. Hunt, "The discreteW transform," Applied Mathematics and Computation, vol. 16, no. 1, pp. 19–48, 1985.

[16] N. C. Hu, H. I. Chang, and O. K. Ersoy, "Generalized discrete Hartley transforms," IEEE Transactions on Signal Processing, vol. 40, no. 12, pp. 2921–2940, 1992.

[17] A. Bortoletti and C. D. Fiore, "On a set of matrix algebras related to discrete Hartley-type transforms," Linear Algebra and Its Applications, vol. 366, pp. 65–85, 2003.

[18] S. Pei, C. Tseng, M. Yeh et al., "Discrete fractional Hartley and fourier transforms," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no. 6, pp. 665–675, 1998.

[19] C. Moraga, "Analysis of Mosaics by means of the Chrestenson and Zhang-Hartley transforms," in Proceedings The Nineteenth International Symposium on Multiple-Valued Logic, pp. 421–427, Guangzhou, China, May 1989.
[24] N. X. Thao and H. T. Van Anh, "On the Hartley-fourier sine generalized convolution," *Mathematical Methods in the Applied Sciences*, vol. 37, no. 15, pp. 2308–2319, 2014.

[25] M. Al-Gharabally and A. F. Almutairi, "Frequency-domain subcarrier diversity receiver for discrete Hartley transform OFDM systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, 2019.

[26] M. R. Abuturab, "An asymmetric single-channel color image encryption based on Hartley transform and gyrator transform," *Optics and Lasers in Engineering*, vol. 69, pp. 49–57, 2015.

[27] Z. Liu, Y. Zhang, W. Liu, F. Meng, Q. Wu, and S. Liu, "Optical color image hiding scheme based on chaotic mapping and Hartley transform," *Optics and Lasers in Engineering*, vol. 51, no. 8, pp. 967–972, 2013.

[28] W. A. Martins and P. S. R. Diniz, "Memoryless block transceivers with minimum redundancy based on Hartley transforms," *Signal Processing*, vol. 91, no. 2, pp. 240–251, 2011.

[29] X. Ouyang, J. Jin, G. Jin, and Z. Wang, "Discrete Hartley transform based SFBC-OFDM transceiver design with low complexity," in *Proceedings of the 2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2744–2749, Shanghai, China, April 2013.

[30] P. Rajavel, "Directional Hartley transform and content based image retrieval," *Signal Processing*, vol. 90, no. 4, pp. 1267–1278, 2010.

[31] I. Prots’ko, "Algorithm of efficient computation of generalised discrete Hartley transform based on cyclic convolutions," *IET Signal Processing*, vol. 8, pp. 301–308, 2013.

[32] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy, "New parametric discrete fourier and Hartley transforms and algorithms for fast computation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, pp. 562–575, 2010.

[33] G. Bi, "On computation of the discrete W transform," *IEEE Transactions on Signal Processing*, vol. 47, pp. 1450–1453, 1999.

[34] V. Kober, "Fast algorithms for the computation of sliding discrete Hartley transforms," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2937–2944, 2007.

[35] J. A. Glassman, "A generalization of the fast fourier transform," *IEEE Transactions on Computers*, vol. 19, no. 2, pp. 105–116, 1970.