<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Conjugate symmetric sequency-ordered complex Hadamard transform (Main article)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Aung, Aye; Ng, Boon Poh; Rahardja, Susanto</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>2009</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10220/7088">http://hdl.handle.net/10220/7088</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>© 2009 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The published version is available at: [DOI: <a href="http://dx.doi.org/10.1109/TSP.2009.2017572">http://dx.doi.org/10.1109/TSP.2009.2017572</a>].</td>
</tr>
</tbody>
</table>
Conjugate Symmetric Sequency-Ordered Complex Hadamard Transform

Aye Aung, Boon Poh Ng and Susanto Rahardja

Abstract—A new transform known as conjugate symmetric sequency-ordered complex Hadamard transform (CS-SCHT) is presented in this paper. The transform matrix of this transform possesses sequence ordering and the spectrum obtained by the CS-SCHT is conjugate symmetric. Some of its important properties are discussed and analyzed. Sequency defined in the CS-SCHT is interpreted as compared to frequency in the discrete Fourier transform. The exponential form of the CS-SCHT is derived, and the proof of the dyadic shift invariant property of the CS-SCHT is also given. The fast and efficient algorithm to compute the CS-SCHT is developed using the sparse matrix factorization method and its computational load is examined as compared to that of the SCHT. The applications of the CS-SCHT in spectrum estimation and image compression are discussed. The simulation results reveal that the CS-SCHT is promising to be employed in such applications.

Index Terms—Complex Hadamard transforms, discrete orthogonal transforms, fast algorithms, dyadic shift invariant, conjugate symmetric sequency-ordered complex Hadamard transform (CS-SCHT).

I. INTRODUCTION

The Hadamard transform has been considered widely as a practical tool to process signals, especially in the areas of digital signal and image processing, filtering, communications and digital logic design [1]–[9] due to its simple implementation with the use of fast algorithms. It is known from the literature that the Walsh functions can be arranged to form any of three main orderings which are in common use. They are the Walsh or sequency order, dyadic order and natural or Hadamard order [10], [11]. The choice of orderings depends on the particular applications [12]. Since, the row vectors of a Hadamard matrix are simply the sampled versions of the Walsh functions, the ordering of Hadamard transforms describes the sequence in which the Walsh functions are positioned in the transform matrix. For instance, the Walsh Hadamard transform (WHT) matrix simply expresses that a sampled Walsh function series is arranged in natural or Hadamard order. If the Walsh function values are arranged in ascending values of sequencies in the transform matrix, a sequency-ordered Walsh transform (SOWT) is obtained. In most applications of signal processing, sequency ordering is preferred due to its analogy to frequency in the discrete Fourier transform (DFT) [13], [14].

A set of complex orthogonal transforms known as unified complex Hadamard transforms (UCHTs) is introduced in [15]–[17] for such needs especially employed for multiple-valued logic design, communications and the signal processing applications dealing with complex-valued functions. The transform matrices are confined to four complex integer values \{±1, ±j\} where \( j = \sqrt{-1} \). But the UCHT matrices do not exhibit certain ordering which may correspond to the WHT or the SOWT. Thereafter, sequence-ordered complex Hadamard transform (SCHT) whose transform matrix shows the sequence ordering is introduced in [18] for some particular applications in communications and signal processing. It is a complex Hadamard transform whose row vectors are arranged in an increasing number of sequencies. The SCHT is shown to have certain significance in spectrum estimation as well as in image watermarking. It has also been applied in the direct sequence (DS) CDMA systems [19], in which each row vector of an SCHT matrix is used as a complex spreading sequence to be assigned to a particular user. But the SCHT coefficients are the complex numbers comprised of real and imaginary parts and they are not conjugate symmetric, hence, more memory are needed to store the coefficients for analysis and synthesis in transform implementation. For example, the SCHT may not be suitable to be employed in image compression as it contradicts the goal of reducing bit rate (which implies less transform coefficients to store) for image compression. On the other hand, only half of the spectral coefficients are required for the analysis in the UCHTs with the half spectrum property (HSP) and the DFT (whose spectrum is conjugate symmetric).

It is then the purpose of this paper to propose a new version of SCHT called conjugate symmetric sequency-ordered complex Hadamard transform (CS-SCHT for short) whose spectrum is conjugate symmetric, and which is expected to outperform previous introduced SCHT in spectrum estimation. As said, the CS-SCHT spectrum is conjugate symmetric so only half of spectral coefficients are required for synthesis and analysis. This in turn reduces the memory requirement in processing for the applications such as real-time image watermarking and spectrum estimation.

Now let us focus on sequency defined in the SCHT [18]. In fact, sequency of the SCHT is complex sequence which describes the amount of rotations (the number of zero crossings) of each row vector of an SCHT matrix in the unit circle of a complex plane over a normalized time base \( 0 \leq t \leq 1 \). This is analogous to frequency in the DFT if frequency is defined as the number of times that each row vector of a DFT matrix crosses the imaginary axis in the unit circle per unit time interval. In the DFT, however, the term frequency refers to periodic repetition rate of sinusoidal waves as well as the rotation of individual row vector in a DFT matrix in the unit circle. This is because the DFT transform matrix is conjugate symmetric and we can intuitively regroup the row vectors of a
DFT matrix to obtain the corresponding sinusoidal waves with respective frequencies (which is to be explained in Section III). The rotation of each row vector of a DFT matrix in the unit circle and the periodic repetition rate of the corresponding sinusoidal waves are observed to be the same, which defines frequency in the DFT.

In this paper, we introduce the conjugate symmetric version of the SCHT, whose sequency is directly related to frequency of the DFT. The rest of the paper is organized as follows. Section II describes the construction of the CS-SCHT transform matrix based on the conjugate symmetric natural-ordered complex Hadamard transform (CS-NCHT) whose construction is based on the WHT and the direct block matrix operation. The definition of the direct block matrix operator is also given. The interpretation of sequency in the CS-SCHT is presented in Section III. Some important properties of the CS-SCHT are derived and analyzed in Section IV, including the exponential SCHT power spectrum and its proof. Using the sparse matrix factorization approach, a fast algorithm to compute an N-point CS-SCHT is derived in Section V and its computational complexity is examined as compared with that of the SCHT. Some applications of the CS-SCHT are suggested in Section VI together with the supporting simulation results and discussions. Finally, this paper is concluded in Section VII.

II. BASIC DEFINITIONS OF THE CS-SCHT

In this section, we shall present the construction of the CS-SCHT matrices. Before going directly to the generation of the CS-SCHT, we first define the direct block matrix operator which is to be used in the subsequent construction.

Let \( B \) be a block matrix such that

\[
B = \begin{bmatrix}
  B_{11} & B_{12} & \cdots & B_{1N} \\
  B_{21} & B_{22} & \cdots & B_{2N} \\
  \vdots & \vdots & \ddots & \vdots \\
  B_{M1} & B_{M2} & \cdots & B_{MN}
\end{bmatrix},
\]

\( R \) be a row block matrix, that is,

\[
R = [ \begin{bmatrix} R_1 & R_2 & \cdots & R_N \end{bmatrix} ]
\]

and \( C \) be a column block matrix which is

\[
C = [ \begin{bmatrix} C_1 & C_2 & \cdots & C_M \end{bmatrix} ]^T.
\]

Then the direct block matrix operator \( \circ \) is defined as

\[
B \circ R = \begin{bmatrix}
  B_{11}R_1 & B_{12}R_2 & \cdots & B_{1N}R_N \\
  B_{21}R_1 & B_{22}R_2 & \cdots & B_{2N}R_N \\
  \vdots & \vdots & \ddots & \vdots \\
  B_{M1}R_1 & B_{M2}R_2 & \cdots & B_{MN}R_N
\end{bmatrix}.
\]

and

\[
B \circ C = \begin{bmatrix}
  B_{11}C_1 & B_{12}C_1 & \cdots & B_{1N}C_1 \\
  B_{21}C_1 & B_{22}C_1 & \cdots & B_{2N}C_1 \\
  \vdots & \vdots & \ddots & \vdots \\
  B_{M1}C_M & B_{M2}C_M & \cdots & B_{MN}C_M
\end{bmatrix}.
\]

where the submatrices \( B_{mn}, R_k \) and \( C_l \) are the square matrices which have the same dimensions, and \( m, n, k \) and \( l \) are positive integers such that \( 1 \leq m, l \leq M \) and \( 1 \leq n, k \leq N \). If the dimensions of the submatrices are \( 1 \times 1 \), the operation of the direct block matrix operator is equivalent to that of the matrix operator defined in [15].

Now we first define the CS-NCHT based on the WHT and the direct block matrix operator as the CS-SCHT is a bit-reversed version of the CS-NCHT. Hence, the generation of the CS-SCHT is different from that of the SCHT where the SCHT is constructed based on the complex Rademacher matrices [18]. Let \( H_N \) be any CS-NCHT matrix of dimension \( N \times N \) where \( N = 2^n \). Then, it is a square matrix defined as

\[
H_N = \begin{bmatrix}
  H_{N/2} & H_{N/2}^T \\
  H_{N/2}^T & -H_{N/2}^T
\end{bmatrix}
\]

where

\[
S_{2^n-1} = \begin{bmatrix}
  I_{2^n-2} & 0 \\
  0 & jI_{2^n-2}
\end{bmatrix}
\]

and \( H_{N/2}' \) is a real Hadamard matrix whose rows are arranged in a certain manner and it is defined recursively as

\[
H_{N/2}' = \begin{bmatrix}
  H_{N/4}' & H_{N/4}' \cdot V_{N/4} \\
  H_{N/4}' \cdot V_{N/4} & -H_{N/4}'
\end{bmatrix}
\]

where \((N/4) = 2^{n-2}\),

\[
I_{2^n-2} = \begin{bmatrix}
  I_{2^n-3} & 0 \\
  0 & jI_{2^n-3}
\end{bmatrix}
\]

is the identity matrix of size \( 2^{n-2} \times 2^{n-2} \) where the bottom half of the elements are multiplied by \((-1)\). Consequently, \( H_{N/2} \) and \( H_{N/2}' \) can be further reduced to the dimension of \( 2 \times 2 \), which is defined as

\[
H_2 = \begin{bmatrix}
  1 & 1 \\
  1 & -1
\end{bmatrix}.
\]

In this way, an \( N \times N \) CS-NCHT matrix can be expressed using the CS-NCHT matrices of order \((N/2) \times (N/2)\) and the smallest dimension of the CS-NCHT matrix will be the size of \( 4 \times 4 \) as shown below.

\[
H_4 = \begin{bmatrix}
  H_2 & H_2' \\
  H_2'^T & -H_2'^T
\end{bmatrix} = \begin{bmatrix}
  1 & 1 & 1 & 1 \\
  1 & -1 & 1 & -1 \\
  1 & j & -1 & -j \\
  1 & -j & 1 & j
\end{bmatrix}.
\]

As a result, any CS-NCHT matrix of dimension \( N \times N \) can be defined using the WHT matrix and the direct block matrix operator as follows:

\[
H_N = W_N \circ A_{n-1,n-1}' \circ \cdots \circ A_{2,2}' \circ A_{1,1}
\]

where \( N = 2^n \), \( W_N \) is the \( N \times N \) WHT matrix,

\[
A_{1,1} = [I_{2^n-1}, S_{2^n-1}]^T
\]

\[
A_{2,2}' = [I_{2^n-2}, S_{2^n-2}, I_{2^n-2}, V_{2^n-2}]^T
\]

\[
A_{n-1,n-1}' = [I_2, S_2, I_2, I_2', \cdots, I_2, I_2]^T
\]
where \((\cdot)^T\) represents the transpose, and \(\circ\) denotes the direct block matrix operator which was previously defined.

Let us, for example, consider \(N = 8\). Then, \(n = \log_2 8 = 3\) and (8) becomes

\[
\mathcal{H}_8 = W_8 \circ A_{2,2} \circ A_{1,1}.
\]  

Substituting the corresponding values defined in (9) into (10), we have

\[
\mathcal{H}_8 = \begin{bmatrix} W_2 & W_2 & W_2 \\ W_2 & -W_2 & -W_2 \\ W_2 & -W_2 & -W_2 \end{bmatrix} \circ \begin{bmatrix} I_2 \\ S_2 \\ I_2 \end{bmatrix} \circ \begin{bmatrix} I_2 \\ S_4 \end{bmatrix}
\]  

(11)

where \(W_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\) is the 2×2 WHT matrix. Therefore,

\[
\mathcal{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & j & -j & 1 & j & -j & 1 & -j \\ 1 & j & -j & 1 & j & -j & 1 & -j \\ 1 & -1 & j & -j & 1 & j & -j & 1 \\ 1 & -1 & j & -j & 1 & j & -j & 1 \\ 1 & -1 & j & -j & 1 & j & -j & 1 \end{bmatrix}.
\]  

(12)

As such any CS-NCHT matrix of dimension \(N \times N\) where \(N = 2^n\) can be achieved. Having defined the CS-NCHT, a CS-SCHT can be obtained from a CS-NCHT matrix through a bit reversal conversion and vice versa. Let \(H_N\) be any CS-SCHT matrix of size \(N \times N\). Then it is defined as

\[
H_N(p, k) = H_N(b(p), k)
\]  

(13)

where \(p\) and \(k\) are the row and column indices of a matrix such that \(0 \leq p, k \leq N - 1\) and \(b(p)\) is the decimal number obtained by the bit-reversed operation of the decimal \(p\). As an example, \(H_8\) is obtained as

\[
H_8 = \begin{bmatrix} H_8(0, k) \\ H_8(1, k) \\ H_8(2, k) \\ H_8(3, k) \\ H_8(4, k) \\ H_8(5, k) \\ H_8(6, k) \\ H_8(7, k) \end{bmatrix} = \begin{bmatrix} \mathcal{H}_8(0, k) \\ \mathcal{H}_8(4, k) \\ \mathcal{H}_8(2, k) \\ \mathcal{H}_8(6, k) \\ \mathcal{H}_8(1, k) \\ \mathcal{H}_8(5, k) \\ \mathcal{H}_8(3, k) \\ \mathcal{H}_8(7, k) \end{bmatrix}
\]  

(14)

in which the row vectors of the matrix are arranged in increasing order of zero crossings in the unit circle of a complex plane. Besides, the \(j\)th row of the matrix is the conjugate of the \((N - j)\)th row vector where \(p = 1, 2, \ldots, (N/2) - 1\) and \(N = 2^n\), hence, the spectrum obtained by using this matrix is shown to be conjugate-symmetric. The \(j\)th and \((N/2)\)th row vectors correspond to the DC and Nyquist frequency components in the DFT matrix, respectively. As such any CS-SCHT matrix of dimension \(N \times N\) can be generated. This completes the construction of the CS-SCHT. Having developed the CS-SCHT matrix, the CS-SCHT of an \(N\)-point complex signal vector \(x_N = [x(0), x(1), \ldots, x(N - 1)]^T\) is defined as

\[
x_N = \frac{1}{N} H_N^* x_N
\]  

(15)

where \(X_N = [X(0), X(1), \ldots, X(N - 1)]^T\) is the transformed complex column vector, \((\cdot)^*\) denotes the complex conjugate and \(H_N\) is the CS-SCHT matrix as defined in (13). The data sequence can be uniquely recovered from the inverse transform, that is,

\[
x_N = H_N^* x_N
\]  

(16)

since \(H_N^* H_N = N I_N\) (unitary property) where \((\cdot)^*\) represents the complex conjugate transpose.

### III. Interpretation of Sequency

The complex Hadamard matrix generated in (13) exhibits sequency ordering, that is, the row vectors of the CS-SCHT matrix are positioned in an ascending order of sequencies. Sequency is defined as one half of the average number of zero crossings per unit time in the unit circle of a complex plane [11], [13]. As indicated earlier, sequency in the CS-SCHT is analogous to frequency in the DFT. Since the CS-SCHT matrix of dimension \(N \times N\) where \(N = 2^n\) is conjugate symmetric, we can intuitively separate the real and imaginary terms and regroup them as in the DFT to obtain the corresponding waveforms (which is an approximation of the sinusoidal waves). This process is illustrated below. The new real matrix obtained from the \(N \times N\) CS-SCHT matrix is denoted as \(H_{R,N}\) and

\[
H_{R,N} = \begin{bmatrix} H_N(0, k) \\ \frac{1}{2}(3 H_N(1, k) - 3 H_N(N - 1, k)) \\ \frac{1}{2}(3 H_N(1, k)) + 3 H_N(N - 1, k) \\ \frac{1}{2}(3 H_N(2, k) - 3 H_N(N - 2, k)) \\ \vdots \\ \frac{1}{2}(3 H_N(N - 1, k) - 3 H_N(N + 1, k)) \\ \frac{1}{2}(3 H_N(N - 1, k)) + 3 H_N(N + 1, k) \\ H_N(N, k) \end{bmatrix}
\]  

(17)

Considering \(N = 8\) as an example, \(H_{R,8}\) is obtained as

\[
H_{R,8} = \begin{bmatrix} H_8(0, k) \\ \frac{1}{2}(3 H_8(1, k) - 3 H_8(7, k)) \\ \frac{1}{2}(3 H_8(1, k)) + 3 H_8(7, k) \\ \frac{1}{2}(3 H_8(2, k) - 3 H_8(6, k)) \\ \frac{1}{2}(3 H_8(2, k)) + 3 H_8(6, k) \\ \frac{1}{2}(3 H_8(3, k) - 3 H_8(5, k)) \\ \frac{1}{2}(3 H_8(3, k)) + 3 H_8(5, k) \\ H_8(4, k) \end{bmatrix}
\]  

(18)

where \(H_8\) is the CS-SCHT matrix of size \(8 \times 8\). \(\Re\{\cdot\}\) and \(\Im\{\cdot\}\) represent real and imaginary parts of a complex number, respectively. The waveforms which are relevant to
each row vector of (18) are shown in Fig. 1 together with the corresponding sinusoidal waveforms from the DFT. It can be seen from the figure that the CS-SCHT is an approximation of sine and cosine waveforms in the DFT by using the pertinent staircase waveforms. Sequency can be expressed as periodic repetition rate of the approximated waveforms as in the case of the DFT where frequency indicates the periodic repetition rate of the approximated waveforms as in the case of the DFT.

**IV. PROPERTIES OF THE CS-SCHT**

In this section, we shall present the properties of the CS-SCHT.

**Property 1: Exponential Property of the CS-SCHT.** Let \( h(p, k) \) be the element at the \( p \)th row and \( k \)th column of any CS-SCHT matrix of dimension \( N \times N \) where \( N = 2^n \) and \( 0 \leq p, k \leq N - 1 \). Then,

\[
h(p, k) = (-1)^{\sum_{r=0}^{n-1} g_r k_r} \sum_{r=0}^{n-1} f_r k_r
\]

where \( n = \log_2 N \) and \( k_r \) is the \( r \)th bit of the binary representation of the decimal integer \( k \), i.e., \( (k)_{10} = (k_{n-1}, k_{n-2}, \ldots, k_0)_{2} \). Besides, \( g_r \) is defined to be the \( r \)th bit of the binary Gray code [11] of bit reversal representation of the decimal \( p \), and we also define \( f_r \) to be the \( r \)th bit of the binary bits of the highest power of 2 in \( b(p)/2 \) where \( b(p) \) is the decimal number obtained through a bit-reversed conversion of the decimal \( p \). The example for \( N = 8 \) is illustrated in Table I to achieve the binary values for the said notations. The complex conjugate of the CS-SCHT matrix can be generated easily by just changing \(-j \) to \( j \) in (19).

**TABLE I: BINARY REPRESENTATIONS FOR \( g_r \) AND \( f_r \).**

<table>
<thead>
<tr>
<th>( p )</th>
<th>Binary</th>
<th>Bit reversal</th>
<th>( g_r )</th>
<th>( b(p) )</th>
<th>( b(p)/2 )</th>
<th>Highest power of 2</th>
<th>( f_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000</td>
<td>000</td>
<td>000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>000</td>
</tr>
<tr>
<td>1</td>
<td>001</td>
<td>100</td>
<td>110</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>010</td>
</tr>
<tr>
<td>2</td>
<td>010</td>
<td>010</td>
<td>011</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>001</td>
</tr>
<tr>
<td>3</td>
<td>011</td>
<td>110</td>
<td>101</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>010</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>001</td>
<td>001</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>000</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>101</td>
<td>111</td>
<td>5</td>
<td>2.5</td>
<td>2</td>
<td>010</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>011</td>
<td>010</td>
<td>3</td>
<td>1.5</td>
<td>1</td>
<td>001</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>111</td>
<td>100</td>
<td>7</td>
<td>3.5</td>
<td>2</td>
<td>010</td>
</tr>
</tbody>
</table>

**Property 2: Unitarity.** The CS-SCHT is an orthogonal transform whose row vectors are orthogonal in the complex domain as

\[
| \text{det} H_N | = N^{N/2}
\]

and

\[
H_N H_N^* = H_N^* H_N = N I_N
\]

where \( H_N \) symbolizes any CS-SCHT matrix of dimension \( N \times N \).

**Property 3: Linearity.** The CS-SCHT is a linear transformation like the DFT.

**Property 4: Parseval’s Theorem.** The energy of the signal in the time and sequency domains is expressed as

\[
\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |X(k)|^2
\]

where \( X(k) \) represents the CS-SCHT coefficients in the sequency domain and \( x(n) \) is the signal in the time domain.

**Property 5: Conjugate Symmetry.** The spectrum transformed by the CS-SCHT is conjugate-symmetric, i.e., the CS-SCHT coefficients \( X(k) \) and \( X(N-k) \) are conjugate-symmetric pairs where \( k = 1, 2, \ldots, (N/2) - 1 \).

**Property 6: Dyadic Shift Invariant Power Spectrum.** Let \( h(p, k) \) be any element of a CS-SCHT transformation matrix. Then the transformation is said to be dyadic shift invariant (DSI) as [20]

\[
h(p, k+\Delta) = \begin{cases} h(p, k) \cdot h(p, m), & \text{if both elements} \\ h(p, k) \cdot h(p, m), & \text{are imaginary} \\ h(p, k) \cdot h(p, m), & \text{otherwise} \end{cases}
\]
Therefore, it has only one bit of 1 and the rest are 0. As indicated in Property 1, the exponential form of the CS-SCHT is

$$ h(p, k) = (-1)^{\sum_{r=0}^{n-1} g_r k_r} \cdot f_r k_r, $$

$$ h(p, k + m) = (-1)^{\sum_{r=0}^{n-1} g_r (k_r + m_r)} \cdot f_r (k_r + m_r). $$

From Table II, the Boolean function $g_r (k_r + m_r)$ is equivalent to the function $g_r k_r + g_r m_r$. Therefore, we have

$$ h(p, k + m) = (-1)^{\sum_{r=0}^{n-1} g_r k_r + g_r m_r} \cdot (-1)^{\sum_{r=0}^{n-1} f_r (k_r + m_r)} = (-1)^{\sum_{r=0}^{n-1} g_r k_r + g_r m_r} \cdot (-1)^{\sum_{r=0}^{n-1} f_r (k_r + m_r)}. $$

(26)

**TABLE II: THE TRUTH TABLE FOR $g_r (k_r + m_r)$ AND $g_r k_r + g_r m_r$.**

<table>
<thead>
<tr>
<th>$g_r$</th>
<th>$k_r$</th>
<th>$m_r$</th>
<th>$k_r + m_r$</th>
<th>$g_r k_r$</th>
<th>$g_r m_r$</th>
<th>$g_r k_r + g_r m_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Now let us consider each term contained in (26). Then, from Table III, it can be found that

$$ (-1)^{\sum_{r=0}^{n-1} g_r k_r + g_r m_r} = (-1)^{\sum_{r=0}^{n-1} g_r k_r + g_r m_r} = (-1)^{\sum_{r=0}^{n-1} g_r k_r + \sum_{r=0}^{n-1} g_r m_r} = (-1)^{\sum_{r=0}^{n-1} g_r k_r} \cdot (-1)^{\sum_{r=0}^{n-1} g_r m_r}. $$

(27)

On the other hand, Table IV shows that the functions $(-j)^{\sum_{r=0}^{n-1} f_r k_r + f_r m_r}$ and $(-j)^{\sum_{r=0}^{n-1} f_r k_r + f_r m_r}$ are not equivalent whenever $f_r = k_r = m_r = 1$. Therefore, it is expressed as

$$ (-j)^{\sum_{r=0}^{n-1} f_r k_r + f_r m_r} = (-j)^{f_{k_{00}} + f_{m_{00}}} \cdot (-j)^{f_{k_{01}} + f_{m_{01}}} \cdot \ldots \cdot (-j)^{f_{k_{n-1,n-1}} + f_{m_{n-1,n-1}}}. $$

(28)

In order to convert $(b_{1,r} \oplus b_{2,r})$ to $(b_{1,r} + b_{2,r})$, consideration is taken into account only if the two binary operands (i.e., $b_{1,r}$ and $b_{2,r}$) are both equal to one. As indicated in Property 1, $f_r$ represents the binary code of highest power of 2 in $b(p)/2$ where $b(p)$ is the bit-reversed converted decimal number. Therefore, it has only one bit of 1 in its binary expression and the rest are 0 (see Table I for an example). $f_r k_r$ in (28) where $r = 0, \ldots, n - 1$ has only one bit of 1 in its binary notation. Hence, (28) can be rewritten as

$$ h(p, k + m) = (-1)^{\sum_{r=0}^{n-1} f_r k_r + f_r m_r} = (-j)^{f_{k_{00}} + f_{m_{00}}} \cdot (-j)^{f_{k_{01}} + f_{m_{01}}} \cdot \ldots \cdot (-j)^{f_{k_{n-1,n-1}} + f_{m_{n-1,n-1}}}. $$

(29)

The term $(-1)^{\sum_{r=0}^{n-1} f_r k_r + f_r m_r}$ is included in the above expression (its value can be $\pm 1$) considering for the case where $f_r = k_r = m_r = 1$ as shown in Table IV. It is also observed from this expression that whenever $f_r = k_r = m_r = 1$, the corresponding elements $h(p, k)$ and $h(p, m)$ are imaginary and one of them is multiplied with $-1$, which in fact is similar to the conjugation of that imaginary element. Therefore, this will prove the requirement defined in (23). Finally, substituting (27) and (29) into (26), we obtain

$$ h(p, k + m) = (-1)^{\sum_{r=0}^{n-1} g_r k_r} \cdot (-1)^{\sum_{r=0}^{n-1} g_r m_r} \cdot (-j)^{\sum_{r=0}^{n-1} f_r k_r + f_r m_r} = \begin{cases} h(p, k) \cdot h(p, m), & \text{if both elements are imaginary} \\ h(p, k) \cdot h(p, m), & \text{otherwise} \end{cases} $$

(30)

and also

$$ ||h(p, k)||^2 = h(p, k) \cdot \overline{h(p, k)} = 1. $$

(31)

This completes the proof of (23) and (24). The proof is completed here. It is noted that the power spectrum of a real-valued input signal obtained through the DSI transformation is dyadic shift invariant [20]. Hence, the CS-SCHT power spectrum is also dyadic shift invariant.
V. FAST ALGORITHM OF THE CS-SCHT

This section will derive the fast algorithm to compute an $N$-point CS-SCHT using the matrix factorization method. Since the CS-SCHT is the bit-reversed counterpart of the CS-NCHT, the method applied to the latter is also applicable to the former and vice versa. In this section, we shall present the factorization method of the CS-NCHT matrix, which leads to the fast algorithms for the CS-NCHT as well as the CS-SCHT for which the output is arranged in bit-reversed order. The forward transformation matrix defined in (15) is the conjugate of the CS-SCHT matrix but we start with the factoring of the CS-SCHT matrix to derive the fast algorithm. Let us, for example, consider $N = 8$ to conceptualize the method of factorization. Then, the expression in (3) becomes

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_4 \mathbf{I}_4 \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} \mathbf{H}_4 \mathbf{S}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_4 & \mathbf{H}_4 \mathbf{I}_4 \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} \mathbf{H}_4 \mathbf{S}_4 \end{bmatrix}. \quad (32)$$

where

$$\mathbf{S}_4 = \begin{bmatrix} \mathbf{I}_2 & 0 \\ 0 & j \mathbf{I}_2 \end{bmatrix} \text{ and } \mathbf{H}_4' = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \mathbf{I}_2' \\ \mathbf{H}_2 \mathbf{I}_2' & -\mathbf{H}_2 \mathbf{I}_2' \end{bmatrix}. \quad (33)$$

Equation (32) can be expressed as

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H} & 0 \\ 0 & \mathbf{H}_4' \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & 0 \\ 0 & \mathbf{S}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \mathbf{I}_4' \\ \mathbf{I}_4 \mathbf{I}_4' & -\mathbf{I}_4 \mathbf{I}_4' \end{bmatrix}. \quad (33)$$

Subsequently, $\mathbf{H}_4$ and $\mathbf{H}_4'$ can be further factorized and $\mathbf{H}_8$ becomes

$$\mathbf{H}_8 = \begin{bmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \mathbf{H}_2 & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{S}_2 \\ \mathbf{S}_2 & \mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & \mathbf{I}_4 \end{bmatrix} \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4' \\ \mathbf{I}_4 \mathbf{I}_4' & -\mathbf{I}_4 \mathbf{I}_4' \end{bmatrix}. \quad (34)$$

where the blank spaces represent the zero elements,

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathbf{S}_2 = \begin{bmatrix} 1 & 0 \\ 0 & j \end{bmatrix} \text{ and } \mathbf{I}_2' = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (34)$$

The factorization of $\mathbf{H}_8$ can be obtained by just changing $j$ to $-j$ in the $\mathbf{S}$ matrices which are included in (34). This factorization will result in fast forward CS-NCHT algorithm. As mentioned earlier, rearranging the output in a bit-reversed order will provide the CS-SCHT coefficients, which gives rise to the fast forward CS-SCHT algorithm. The decomposition of fast forward CS-SCHT for $N = 8$ is illustrated by the signal flow graph in Fig. 2.

For simplicity, the multiplication by the scaling factor $(1/8)$ is omitted in the figure, which in fact can be computed by the binary shifting operations without needing any arithmetic operations. As shown in Fig. 2, the signal flow graph contains three stages and each stage requires eight addition/subtractions. But a total of $2^0 + 2^1 = 3$ complex multiplications by factor $-j$ is needed for three stages. Therefore, it can be concluded that for an $N$-point CS-SCHT, the fast algorithm has $n = \log_2 N$ stages and each stage needs $N$ addition/subtractions. In general, the total computational requirements for complex input data are $N \log_2 N$ complex addition/subtractions and

$$2^0 + 2^1 + \cdots + 2^{n-2} = \frac{1 - 2^{n-1}}{1 - 2} = (2^{n-1} - 1) \quad (35)$$

complex multiplications for the trivial twiddle factors $-j$. Compared to fast SCHT algorithm reported in [18], they have the same number of complex addition/subtractions but this presented fast CS-SCHT algorithm requires less trivial complex multiplications than fast SCHT algorithm, that is, $(2^{n-1} - 1)$ for the CS-SCHT and $(N/4) \log_2 (N/2) = 2^{n-2} (n - 1)$ for the SCHT. The numerical numbers are presented in Table V for the comparison purpose. It can be seen from the table that when $n$ is increased, the difference becomes larger and significant.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Number of complex multiplications by the elements $-j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>31</td>
</tr>
<tr>
<td>7</td>
<td>63</td>
</tr>
<tr>
<td>8</td>
<td>127</td>
</tr>
<tr>
<td>9</td>
<td>255</td>
</tr>
<tr>
<td>10</td>
<td>511</td>
</tr>
</tbody>
</table>

This fast CS-SCHT algorithm is well suited to adopt the pipelined hardware structure, which was proposed for the SCHT in [21]. The signal flow graph in Fig. 2 is similar to that of the decimation-in-sequency (DIS) fast SCHT algorithm (which was used to develop the pipelined hardware structure for the SCHT) except that the former needs less complex multiplications by the trivial twiddle factors. As pointed out in [21], the operations of swapping and subtraction can be jointly performed with the aid of the $2 : 1$ complex multiplexers to accomplish the multiplications with $-j$. Therefore, it should be noted that less number of operations are needed to implement the CS-SCHT as compared to the SCHT.
VI. APPLICATIONS OF THE CS-SCHT

In this section, we shall present some potential applications of the CS-SCHT including the supporting simulation results.

A. Spectrum Estimation

The CS-SCHT may have certain significance in signal analysis and synthesis due to its sequency ordering and its simple implementation which only requires additions and subtractions for transformation. Fig. 3 illustrates the magnitude response for each row vector of the CS-SCHT matrix for \( N = 8 \) and Fig. 4 shows the magnitude response for the DFT matrix. It can be seen from the figures that sequency in the CS-SCHT is closely related to frequency in the DFT and the locations of their main lobes are the same even though there are uneven main lobes in the magnitude response of the CS-SCHT as compared to that of the DFT.

![Fig. 3: Magnitude response of the CS-SCHT matrix \((N = 8)\).](image)

![Fig. 4: Magnitude response of the 8 × 8 DFT matrix.](image)

As an example, the CS-SCHT is applied in spectrum estimation of the sinusoidal waves. Fig. 5 shows the magnitude spectrum of 512-point DFT, CS-SCHT and SCHT of the sinusoidal wave of 1000 Hz using the sampling frequency of 500000 Hz (in order to show the complete spectrum). The corresponding highlighted portion in Fig. 5, which is zoomed in to provide the clear view, is shown in Fig. 6. It can be seen from Fig. 5 that both CS-SCHT and SCHT spectra are closely matched with that of the DFT. The desired main peaks occur in the same locations for all spectra as shown in Fig. 6. But it is observed that the SCHT spectrum has more unwanted side lobes than that of the CS-SCHT as illustrated in the figures. Therefore, it can be concluded that sequency of the CS-SCHT is more closely related to frequency of the DFT than that of the SCHT. Another obvious advantage is that the spectrum obtained by the CS-SCHT are conjugate-symmetric (like the DFT spectrum) as shown in Fig. 5, hence, only half of the spectrum coefficients are required for analysis as compared with the SCHT spectrum. In a nutshell, the CS-SCHT can be considered as a good substitute to replace the DFT in...
spectrum estimation while providing simple implementation and less computational time.

B. Image Compression

Transform coding technique is one of the most popular methods to compress the digital images. In transform coding, a linear and reversible transform is used to map the image into a set of transform coefficients which are then quantized and coded. The fundamental goal of image compression is to reduce the bit rate which implies less transform coefficients to reconstruct the image while maintaining an acceptable fidelity or image quality. Therefore, the transforms having good energy compaction property are advisable to be employed in image compression. It is a well-known fact that most orthogonal transforms tend to pack a large fraction of the average energy of the images into a relatively few components of the transform coefficients (energy compaction property) [22]. The most well-known orthogonal transform is the discrete cosine transform (DCT) [11], [22], [23] for its highly information packing capability as well as the availability of various efficient fast computational algorithms [24]. The current standard for compression of still images, JPEG, uses the DCT to compress the images. The WHT [2] is the simplest non-sinusoidal orthogonal transform and also provides comparable performance in image compression as compared with the DCT. It is perhaps a better choice for real-time implementation in some applications.

In this subsection, the CS-SCHT is proposed to be applied in image compression. Since an image can be represented as a matrix consisting of real numbers, real-valued transforms are preferred to be used in order to obtain the real transform coefficients. For this purpose, the \(8 \times 8\) transform matrix which is derived from the CS-SCHT (shown in (18)) could be used for image compression with very low computational cost, which merely consists of addition/subtractions. Since the CS-SCHT is a complex orthogonal transform, its real-valued transform is also an orthogonal transform. After normalizing the elements of each row vector of (18) by its 2-norm, the \(8 \times 8\) transform matrix is obtained as

\[
C_8 = D_8 \cdot H_{R,8}
\]

where \(D_8 = \text{diag}\left\{\sqrt{\frac{1}{2}},\sqrt{\frac{1}{2}},\sqrt{\frac{1}{2}},\sqrt{\frac{1}{2}},\sqrt{\frac{1}{2}},\sqrt{\frac{1}{2}},\sqrt{\frac{1}{2}},\sqrt{\frac{1}{2}}\right\}\) is the diagonal matrix which consists of the normalization factors. As mentioned in Section III, the row vectors of this matrix are the approximations of the sinusoidal waves in the DFT. Therefore, the input signal can be represented as a linear combination of its row vectors expecting that the signal energy could be concentrated on a few transform coefficients for good compression. The corresponding transform is defined as follows:

\[
X_8 = C_8 x_8
\]

where \(X_8 = [X(0), X(1), \ldots, X(7)]^T\) is the transformed column vector and \(x_8 = [x(0), x(1), \ldots, x(7)]^T\) represents the input data vector. The data sequence can be recovered uniquely from the inverse transform, that is,

\[
x_8 = C_8^T X_8
\]

where \(C_8^T C_8 = I_8\) (orthogonal property). The two-dimensional (2-D) transform for an \(8 \times 8\) real image matrix can be obtained by applying 1-D transform on the rows followed by the columns. The resultant transform coefficients are the real numbers, which are suitable for image compression.

The transformation by \(C_8\) can be done efficiently with minimum computational cost by using the fast algorithm. The fast algorithm to compute this transform is derived for the purpose of implementation for image compression. The readers may refer to Appendix for details. The signal flow graph for fast forward 8-point transform is illustrated in Fig. 7. For simplicity, the normalization factors are omitted in the figure, which in fact can be computed after or before the transformation. It can be seen from the figure that only the first stage needs eight addition/subtractions whereas the second and third stages merely requires four and six addition/subtractions, respectively. Therefore, a total of \((8 + 4 + 6) = 18\) addition/subtractions is needed to compute an 8-point 1-D transform. On the other hand, 24 addition/subtractions are required to compute an 8-point traditional fast WHT using the fast algorithm [11]. The same number of multiplications (which is eight) is required for both transforms when the signal is scaled by the normalization factors, which is \(D_8\) mentioned in (36). Hence, an additional savings of \((24 - 18) = 6\) addition/subtractions (which is \(\frac{6}{24} \times 100\% = 25\%\)) can be gained for one transformation by using the proposed transform as compared to the WHT.

The computer simulations are also conducted to evaluate the mean square error (MSE) performance of the proposed transform for energy compaction property as well as the bit rate (BR) vs peak signal-to-noise ratio (PSNR) performance for image compression. The image block size considered in the simulation is the standard size of \(8 \times 8\). The original images are divided into non-overlapping \(8 \times 8\) blocks and the corresponding transforms are performed on these blocks. During the reconstruction process, the coefficients with maximum magnitudes on each transformed \(8 \times 8\) block are selected based on the compression ratio. Fig. 8 shows the MSE comparison of various transforms based on the \(8 \times 8\) image blocks with respect to the number of coefficients chosen for reconstructing the Lena image. We have also tested for the Lena image based
on the $16 \times 16$ image blocks and the result is shown in Fig. 9.
It can be seen from the figures that the MSE performances of
the proposed transform and the WHT are comparable to each
other for most of the compression ratios (i.e., the number of
coefficients selected). Hence, it is observed that the proposed
transform also possesses good energy compaction property
like the WHT and it is suitable for application in image
compression as well.

For image compression, the default settings for the JPEG
are used [25] to evaluate the bit rate vs PSNR performance.
The test images used in this simulation are the standard Lena
and Barbara images. We have tested on other standard images
as well and the results are consistent. Table VI lists the
comparisons of the PSNR and BR values among the DCT,
the proposed transform and the WHT at different values of
$M$ (which is the number of transform coefficients selected to
reconstruct the original image) for the Lena image. The orig-

![Fig. 8: MSE comparison among the DCT, the proposed transform, and the WHT with respect to the number of coefficients selected during reconstruction for the Lena image.](image)

![Fig. 9: MSE comparison based on the $16 \times 16$ image blocks.](image)

<table>
<thead>
<tr>
<th>$M$</th>
<th>PSNR</th>
<th>BR</th>
<th>PSNR</th>
<th>BR</th>
<th>PSNR</th>
<th>BR</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>27.2634</td>
<td>0.5563</td>
<td>24.7848</td>
<td>0.5587</td>
<td>25.6427</td>
<td>0.5562</td>
</tr>
<tr>
<td>12</td>
<td>28.1484</td>
<td>0.6408</td>
<td>25.6359</td>
<td>0.6776</td>
<td>26.5533</td>
<td>0.6756</td>
</tr>
<tr>
<td>16</td>
<td>29.6632</td>
<td>0.7408</td>
<td>27.3659</td>
<td>0.8103</td>
<td>27.6964</td>
<td>0.7690</td>
</tr>
<tr>
<td>20</td>
<td>30.9772</td>
<td>0.8322</td>
<td>28.1081</td>
<td>0.8982</td>
<td>28.3369</td>
<td>0.8526</td>
</tr>
<tr>
<td>24</td>
<td>31.1807</td>
<td>0.8864</td>
<td>28.4672</td>
<td>0.9577</td>
<td>28.6208</td>
<td>0.9132</td>
</tr>
<tr>
<td>28</td>
<td>31.9008</td>
<td>0.9321</td>
<td>29.2968</td>
<td>1.0211</td>
<td>29.5068</td>
<td>0.9957</td>
</tr>
<tr>
<td>32</td>
<td>32.3919</td>
<td>0.9786</td>
<td>30.0928</td>
<td>1.0972</td>
<td>30.3276</td>
<td>1.0468</td>
</tr>
</tbody>
</table>

$M$ - The number of coefficients taken for reconstruction. 
PSNR - Peak signal-to-noise ratio (dB). 
BR - Bit rate (bit/pixel).

TABLE VII: COMPARISONS AMONG THE TRANSFORMS FOR THE BARBARA IMAGE.

<table>
<thead>
<tr>
<th>$M$</th>
<th>PSNR</th>
<th>BR</th>
<th>PSNR</th>
<th>BR</th>
<th>PSNR</th>
<th>BR</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>26.9354</td>
<td>0.5805</td>
<td>24.7945</td>
<td>0.5812</td>
<td>25.3570</td>
<td>0.5764</td>
</tr>
<tr>
<td>12</td>
<td>27.9808</td>
<td>0.6988</td>
<td>25.8175</td>
<td>0.7330</td>
<td>26.5008</td>
<td>0.7287</td>
</tr>
<tr>
<td>16</td>
<td>29.0273</td>
<td>0.7976</td>
<td>27.1819</td>
<td>0.8649</td>
<td>27.3502</td>
<td>0.8300</td>
</tr>
<tr>
<td>20</td>
<td>29.9303</td>
<td>0.8863</td>
<td>27.7014</td>
<td>0.9602</td>
<td>27.8418</td>
<td>0.9120</td>
</tr>
<tr>
<td>24</td>
<td>30.1210</td>
<td>0.9260</td>
<td>28.0872</td>
<td>1.0203</td>
<td>28.1861</td>
<td>0.9623</td>
</tr>
<tr>
<td>28</td>
<td>30.6784</td>
<td>0.9805</td>
<td>28.6956</td>
<td>1.0860</td>
<td>28.8896</td>
<td>1.0510</td>
</tr>
<tr>
<td>32</td>
<td>31.0625</td>
<td>1.0248</td>
<td>29.2284</td>
<td>1.1500</td>
<td>29.4820</td>
<td>1.1007</td>
</tr>
</tbody>
</table>

During reconstruction for the Lena image, and the WHT with respect to the number of coefficients selected
to reconstruct the original image) for the Lena image. The orig-
inal Lena image and the respective decompressed images are
shown in Fig. 10 with 20 coefficients chosen for reconstruction
using different transforms. It is observed from Table VI that
when the values of $M$ become larger, the corresponding PSNR
values and bit rates are increased for various transforms.
That is, more binary bits are required to represent the pixel
values of the images with increased values of $M$, hence, less
distortion is occurred to the reconstructed images. Obviously,
at any particular value of $M$, the DCT outperforms the other
transforms in terms of PSNR and bit rates at the expense of
highly computational cost. On the other hand, it is found from
the table that the PSNRs of the reconstructed images and the
corresponding bit rates for the proposed transform and the
WHT are close to each other at any individual value of $M$.
Besides, their reconstructed image qualities are also equally
good enough to compare to that of the DCT as shown in Fig.
10. No significant visual degradation occurs in both images.

But the advantage of the proposed transform over the WHT
is the reduced computational cost. As mentioned earlier, an
extra savings of 6 addition/subtractions (25%) can be acquired
for one transformation by using the proposed transform as
compared to the WHT. If an image of $512 \times 512$ pixels is
considered for image compression, the total number of blocks
will be $(512 \times 512) \div (8 \times 8) = 4096$ and each block needs 16
transformations using (37). As one transformation performs
24 addition/subtractions, a total of $24 \times 16 \times 4096 = 1572864$
addition/subtractions is needed for the whole computation
if the WHT is used. However, only a total of $1572864 \times
0.75 = 1179648$ addition/subtractions is required for the
proposed transform, hence, we can obtain an additional saving.
of $1572864 \times 0.25 = 393216$ addition/subtractions, which is significant in the hardware implementation especially for real-time applications. Therefore, if the hardware savings and simplicity are the first priority to be considered rather than the image quality in the applications, the proposed transform and the WHT are shown to be better choices than the DCT, but among them, the proposed transform is better option to be considered.

Table VII summarizes the comparisons for the Barbara image which contains high frequency components. We have observed from the table that the results are consistent with previous findings concluded for the Lena image.

VII. CONCLUSION

The conjugate symmetric sequency-ordered complex Hadamard transform is discussed in this paper. It has been shown that sequence defined in the CS-SCHT is more closely related to frequency in the DFT as compared to that of the SCHT. It can be seen that the CS-SCHT is a close approximation to the sinusoidal waves in Fourier analysis using the staircase waveforms while the SOWT provides the approximation of the sine and cosine waves in the DFT using the square waves [26]. As the name follows, the CS-SCHT spectrum of a real-valued data sequence is conjugate-symmetric as well as the transform itself is sequency-ordered and suitable for complex-valued functions, which makes it unique among the transforms. Due to the conjugate-symmetric property, only half of the spectrum are needed for analysis and synthesis, which leads to achieve memory savings in the processing of the transform in the applications as compared to the SCHT.

The construction of the CS-SCHT is based on the CS-NCHT. It has been shown that they can be obtained from the WHT and the direct block matrix operation. Some important properties of the CS-SCHT are discussed and analyzed. The exponential property of the CS-SCHT is mentioned and the definition of dyadic shift invariant property for complex Hadamard transform is given. Subsequently, it has been proved that the power spectrum acquired by the CS-SCHT is dyadic shift invariant. The fast CS-SCHT algorithm is derived using the sparse matrix factorization approach and its computational complexity is examined. It has been investigated that less number of complex multiplications by the trivial twiddle factors, i.e., $-j$ are required to compute an $N$-point CS-SCHT as compared to that of the SCHT using the fast algorithm. Therefore, less number of swapping and subtraction operations are required when implementing the transform in the pipelined hardware structure [21]. The applications of the CS-SCHT in spectrum estimation and image compression are also mentioned in this paper. The CS-SCHT has been shown to be a better choice to replace the DFT (at the cost of some loss in accuracy of estimation) in signal analysis and synthesis as compared to the SCHT with simple implementation and less computational complexity. Moreover, the new transform which is derived from the CS-SCHT is proposed in image compression. The fast algorithm to compute the 8-point proposed transform is also derived especially for use in image compression. It has been found that the PSNR values of the reconstructed images with respect to the original image using the proposed transform are comparable to that of the WHT at any chosen value of $M$, the number of coefficients selected. Their corresponding bit rates are also close to each other. But the proposed transform requires less number of addition/subtractions with the use of the presented fast algorithm as compared to that of the WHT. Therefore, the proposed transform have been shown to be better alternative to substitute the DCT in image compression if simplicity and computational cost are the important factors to be considered in the applications.

APPENDIX

FAST ALGORITHM FOR A REAL TRANSFORM DERIVED FROM THE CS-SCHT

From (37), the transform is given by

$$X_8 = C_8 x_8$$

(39)

where $C_8$ is the transform matrix as defined in (36), $x_8$ and $X_8$ are the input data and transformed column vectors, respectively. Firstly, the row vectors of $C_8$ are rearranged through a conversion of the bit reversal and converted Gray code [11]. The example for such conversion for $N = 8$ is tabulated in Table VIII. Therefore, (39) becomes

$$PX_8 = PC_8 x_8$$

(40)

where $P$ is a permutation matrix which provides the arrangement of the row vectors of $C_8$ as mentioned in Table VIII.
Therefore,

\[
C_P = PC_8 = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

(41)

For simplicity, the scaling factor diagonal matrix \(D_8\) is omitted in the above expression and the subsequent derivation. Now let us focus on the factorization of the matrix in (41) to derive the fast algorithm. \(C_P\) can be rewritten and factorized as

\[
C_P = \begin{bmatrix}
C_4 & C_4 \\
C_4' & -C_4'
\end{bmatrix} = \begin{bmatrix}
I_4 & I_4 \\
I_4 & -I_4
\end{bmatrix}
\]

(42)

where

\[
C_4 = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 \\
0 & 1 & 0 & -1 \\
1 & 0 & -1 & 0
\end{bmatrix}, \quad C_4' = \begin{bmatrix}
0 & 0 & 1 & 1 \\
0 & 0 & -1 & 1 \\
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0
\end{bmatrix}
\]

Equation (42) can be further factorized as

\[
C_P = \begin{bmatrix}
C_2 & 0 \\
I_2 & -I_2
\end{bmatrix} \begin{bmatrix}
I_4 & I_4 \\
I_4 & -I_4
\end{bmatrix} \begin{bmatrix}
C_2' & 0 \\
0 & C_2
\end{bmatrix}
\]

(43)

where

\[
C_2 = \begin{bmatrix}
1 & 1 \\
1 & -1
\end{bmatrix}, \quad C_2' = \begin{bmatrix}
1 & 1 \\
-1 & -1
\end{bmatrix} \quad \text{and} \quad I_2 = \begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix}
\]

Finally, we have

\[
C_P = \begin{bmatrix}
C_2 & I_2 \\
I_2 & C_2'
\end{bmatrix} \begin{bmatrix}
I_2 & I_2 \\
I_2 & -I_2
\end{bmatrix} \begin{bmatrix}
0 & I_2 \\
I_2 & 0
\end{bmatrix} \begin{bmatrix}
I_4 & I_4 \\
I_4 & -I_4
\end{bmatrix}
\]

(44)

which results in fast forward algorithm after substituting it into (40). The output sequence is needed to be rearranged according to the permutation matrix \(P\). The signal flow graph for the fast forward algorithm is illustrated in Fig. 7.
Aye Aung received the B.Eng. degree in electrical and electronic engineering from the Nanyang Technological University (NTU), Singapore, in 2005. Since 2006, he has been working towards the Ph.D. degree at NTU, Singapore.

His research interests include digital signal processing for multimedia applications and communications, signal analysis and synthesis, fast algorithms for various transforms, image signal processing in general, multimedia digital watermarking, and wireless communications.

Boon Poh Ng received the B.Eng. degree in electrical engineering from the Nanyang Technological Institute, Singapore, in 1987, the D.I.C. and M.Sc. degrees in communications and signal processing from the Imperial College, University of London, London, U.K., in 1991, and the Ph.D. degree from the Nanyang Technological University (NTU), Singapore, in 1995.

He was a Lecturer with the Department of Electronics and Communication Engineering, Singapore Polytechnic, from 1987 to 1996. From 1996 to 1999, he was a Senior Research Fellow at the Centre for Signal Processing, NTU. He is currently an Associate Professor at the School of Electrical and Electronic Engineering, NTU. His research interests include array synthesis, adaptive array processing, spectral estimation, and digital signal processing in general.

Dr. Susanto Rahardja received his B.Eng. degree from the National University of Singapore, and M.Eng. and Ph.D. degrees from the Nanyang Technological University (NTU, Singapore), all in Electrical & Electronic Engineering. He is currently the Director of Personal 3D Entertainment System program and Head of Signal Processing department at the Institute for Infocomm Research (IR). He is also a program director of Infuse initiative at Science and Engineering Research Council (SERC) of A*STAR. His research interests are in Audio/Video Signal Processing, Image Processing, Spread Spectrum and Multi-user Detection Techniques for CDMA Applications, Digital Signal Processing Algorithms and Implementations, of which he has more than 200 publications in internationally refereed journals and conferences.

Dr Rahardja is an Associate Professor at the School of Electrical and Electronic Engineering in NTU and a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE). He was the recipient of the IEE Hartree Premium Award in 2002. In 2003, Dr Rahardja received the prestigious Tan Kah Kee Young Inventors’ Gold award in the Open Category, for his contributions on scalable to lossless audio compression technology.

Since 2002, Dr Rahardja actively participated in the international ISO/IEC JTC1/SC29/WG11 (Moving Picture Expert Group, or MPEG) where he contributed to MPEG-4 Scalable to Lossless System (SLS) and the technology is incorporated and published in ISO/IEC 14496-3:2005/Amd.3:2006. He also contributed technology to the MPEG-4 Audio Lossless System (ALS) where it is now incorporated in ISO/IEC 14496-3:2005/Amd.2:2006. In recognition for his contributions to the national standardization program, Dr Rahardja was awarded the Standards Council Merit Award by SPRING Singapore in 2006.

For his leadership and technical contribution to advancement of digital audio signal processing and its adoption to the MPEG, he was awarded the National Technology Award in 2007.

Dr. Rahardja has served in several boards, advisory and technical committees in various IEEE and SPIE as well as ACM SIGGRAPH related professional activities in the areas of multimedia. He is elected members of the Technical Committee of the Visual Signal Processing and Communications, Circuits and Systems for Communications and Multimedia Systems and Applications of the IEEE Circuits & Systems Society. He is currently serving as Associate Editors for IEEE Transactions on Audio, Speech and Language Processing and IEEE Transactions on Multimedia as well as the Journal of Visual Communication and Image Representation. He received the A*STAR Most Inspiring Mentor Award in March 2008.