Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 4, No. 2, June 2016, pp. 120~125
ISSN: 2089-3272, DOI: 10.11591/ijeei.v4i2.220

Fast Algorithm for Computingthe Discrete Hartley Transform of Type-II

Mounir Taha Hamood
Department of Electrical Engineering
Collage of Engineering, Tikrit University, Tikrit P O Box 42, Iraq
e-mail: m.t.hamood@tu.edu.iq

Abstract

The generalized discrete Hartley transforms (GDHTs) have proved to be an efficient alternative to the generalized discrete Fourier transforms (GDFTs) for real-valued data applications. In this paper, the development of direct computation of radix-2 decimation-in-time (DIT) algorithm for the fast calculation of the GDHT of type-II (DHT-II) is presented. The mathematical analysis and the implementation of the developed algorithm are derived, showing that this algorithm possesses a regular structure and can be implemented in-place for efficient memory utilization.Theperformance of the proposed algorithm is analyzed and the computational complexity is calculated for different transform lengths. A comparison between this algorithm and existing DHT-II algorithms shows that it can be considered as a good compromise between the structural and computational complexities.

Keywords: *Generalized discrete Hartley transforms (GDHTs), discrete Hartley transform of type-II (DHT-II), fast transform algorithms*

1. Introduction

Over last years, the discrete Hartley transform (DHT) has gained popularity in the fields of digital signal and image processing, as it possesses many desirable properties, used in many applications [1-3]. In general, there are four types of this transform known as type-I, -II, -III and -IV DHTs respectively. In the literature, the type-I DHT is just the DHT and other types are called the generalized discrete Hartley transforms (GDHTs), because their definition contains shifts in either time, frequency index or both indices. The GDHTs are also known as the W transforms that were introduced by Z. Wang [4, 5], the only difference between them is that they use different scaling factors in their definitions. If these scaling factors are ignored, the same fast algorithms can be used for both GDHTs and W transforms. The (GDHT/W) transforms have proven to be important tools in the signal processing and related fields, such as for fast computation of different types of convolutions [6-9], filter banks, signal representations [10] and many other applications.

The direct computation of the GDHTs is intensive and requires large arithmetic operations of order N^2, where N is the transform length; therefore fast GDHTs (GFHTs) algorithms have been introduced to reduce the arithmetic complexity and implementation costs. Among them, Hu *et al.* [11] proposed several fast algorithms for computing the GDHTs, Bi and Chen [12] derived a split-radix algorithm for the computation of GDFTs and GDHTs. Chiper [13] introduced an algorithm for decomposing DHT-II of length N using two adjacent $N/2$ sets of coefficients. The aforementioned algorithms are focused on the sequences with length N being power of two. Other fast algorithm for calculating DHT-II with length N being power of three is introduced by Shu*et al.* [14] and for composite transform lengths is developed by Bi *et al.* [15]. Moreover, Shu*et al.* [16] developed a new fast algorithm for the direct computation of the (DHT-II) based on decomposition of DHT-II into two DHTs of type-II of length $N/2$ that can be used to solve such a problem.

While Shu's algorithm is based on decimation-in-frequency (DIF) approach; however, for any transform to stand as a good candidate for real time applications, its complete fast algorithms need to be developed, such as the decimation-in-time (DIT) approach. Therefore, it is the aim of this paper to develop such an algorithm for fast computation of the DHT-II.

The rest of the paper is organized as follows. Section 2 presents the derivation of the new fast DIT algorithm for computing the DHT-II. The analysis of the computational complexity

Received January 15, 2016; Revised April 3, 2016; Accepted April 20, 2016

for the developed algorithm is given and comparison with the Hu's algorithm is also provided in Section 3. Finally, Section 4 concludes the paper.

2. Algorithm Derivation

The DHT-II transform for a real-valued sequence $x(n)$ of length *N* is defined as [1]:

$$X(k) = \sum_{n=0}^{N-1} x(n) cas\left(\bar{} \, n(\tfrac{2k+1}{2})\right) \qquad k = 0,1,\ldots,N-1 \tag{1}$$

and the corresponding inverse transform (called type-III DHT) is given by:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) cas\left(\bar{\theta}k(\tfrac{2n+1}{2})\right) \qquad n = 0,1,\ldots,N-1 \tag{2}$$

where $\mathrm{cas}(\theta) = \cos(\theta) + \sin(\theta)$, $\theta = 2\pi/N$ and N is the transform length.

The decimation-in-time algorithm derivation begins by dividing the input sequence $x(n)$into its even $X_e(k)$ and odd $X_o(k)$ parts. Therefore (1)can be decomposed as:

$$X(k) = X_e(k) + X_o(k) \tag{3}$$

Where

$$X_e(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n) cas\left(2\bar{} \, n(\tfrac{2k+1}{2})\right) = X_{2n}(k) \tag{4}$$

and

$$\begin{aligned} X_o(k) &= \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)\, cas\left(\bar{} \,(2n+1)(\tfrac{2k+1}{2})\right) \\ &= \sum_{n=0}^{\frac{N}{2}-1} x(2n+1)\, cas\left(2n\bar{} \,(\tfrac{2k+1}{2}) + \bar{} \,(\tfrac{2k+1}{2})\right) \end{aligned} \tag{5}$$

Using the following *cas*property

$$\mathrm{cas}(\sim + \quad) = \cos(\sim)\mathrm{cas}(\quad) + \sin(\sim)\mathrm{cas}(- \quad) \tag{6}$$

cas(.) term in (5) can be simplified to:

$$cas\left(2\theta n(\tfrac{2k+1}{2}) + \bar{} \,(\tfrac{2k+1}{2})\right) = \cos(\bar{} \,\tfrac{2k+1}{2}) cas\left(2\bar{} \, n(\tfrac{2k+1}{2})\right) + \sin(\bar{} \,\tfrac{2k+1}{2}) cas\left(\text{-}2\bar{} \, n(\tfrac{2k+1}{2})\right) \tag{7}$$

Therefore, $X_o(k)$ can be decomposed further to yield:

$$X_o(k) = \cos(\bar{} \,\tfrac{2k+1}{2})\sum_{n=0}^{\frac{N}{2}-1} x(2n+1)\, cas\left(2\bar{} \, n(\tfrac{2k+1}{2})\right) + \sin(\bar{} \,\tfrac{2k+1}{2})\sum_{n=0}^{\frac{N}{2}-1} x(2n+1)\, cas\left(\text{-}2\bar{} \, n(\tfrac{2k+1}{2})\right) \tag{8}$$

The second summation of (8) can be simplified further to

$$\begin{aligned}\sum_{n=0}^{N\text{-}1} x(2n+1)\,cas\left(-2^{-} n\frac{(2k+1)}{2}\right) &= \sum_{n=0}^{N\text{-}1} x(2n+1)\,cas\left(2^{-} n(-k-\tfrac{1}{2})\right)\\ &= \sum_{n=0}^{N\text{-}1} x(2n+1)\,cas\left(2^{-} n(N-k-\tfrac{1}{2})\right)\\ &= \sum_{n=0}^{N\text{-}1} x(2n+1)\,cas\left(2^{-} n\tfrac{2(N-k-1)+1}{2}\right)\end{aligned} \quad (9)$$

Substituting (9) into (8) we get:

$$X_o(k) = \cos\left(\theta\tfrac{2k+1}{2}\right)X_{2n+1}(k) + \sin\left(^{-}\tfrac{2k+1}{2}\right)X_{2n+1}(N\text{-}k\text{-}1) \quad (10)$$

where$X_{2n+1}(k)$ and $X_{2n+1}(N-k-1)$ can be identified as two *N*/2 point DHT-II for odd part $x(2n+1)$ of $x(n)$, given by:

$$\begin{aligned}X_{2n+1}(k) &= \sum_{n=0}^{\frac{N}{2}\text{-}1} x(2n+1)\,cas\left(2^{-} n(\tfrac{2k+1}{2})\right)\\ X_{2n+1}(N\text{-}k\text{-}1) &= \sum_{n=0}^{\frac{N}{2}\text{-}1} x(2n+1)\,cas\left(\text{-}2^{-} n(\tfrac{2k+1}{2})\right)\end{aligned} \quad (11)$$

Replacing (4) and (10) into (3), we obtain the following recursive formula:

$$X(k) = X_{2n}(k) + \left[\cos(^{-}\tfrac{2k+1}{2})X_{2n+1}(k) + \sin(^{-}\tfrac{2k+1}{2})X_{2n+1}(N\text{-}k\text{-}1)\right] \quad (12)$$

For radix-2 algorithm, another point $X\left(k+\frac{N}{2}\right)$needs to be computed. This point can be derived using trigonometric identities and the periodicity property of DHT-II, we get:

$$X(k+\tfrac{N}{2}) = X_{2n}(k) - \left[\cos(^{-}\tfrac{2k+1}{2})X_{2n+1}(k+\tfrac{N}{2}) + \sin(^{-}\tfrac{2k+1}{2})X_{2n-1}(N\text{-}k\text{-}1)\right] \quad (13)$$

Examine (12) and (13), we realize that the in-place property of these decompositions is not possible, due to the fact that the computation of $X(k)$ and $X\left(k+\frac{N}{2}\right)$ points require $X_{2n+1}(k)$ as well as $X_{2n+1}(N-k-1)$. However this difficulty can solve by additionally considering decompositions for $X(N-k-1)$ and $X\left(\frac{N}{2}-k-1\right)$ points, as follows:

$$X(N\text{-}k\text{-}1) = X_{2n}(\tfrac{N}{2}\text{-}k\text{-}1) + \left[\cos(^{-}\tfrac{2k+1}{2})X_{2n+1}(N\text{-}k\text{-}1) - \sin(^{-}\tfrac{2k+1}{2})X_{2n+1}(k+\tfrac{N}{2})\right] \quad (14)$$

$$X(\tfrac{N}{2}\text{-}k\text{-}1) = X_{2n}(\tfrac{N}{2}\text{-}k\text{-}1) - \left[\cos(^{-}\tfrac{2k+1}{2})X_{2n+1}(N\text{-}k\text{-}1) - \sin(^{-}\tfrac{2k+1}{2})X_{2n+1}(k+\tfrac{N}{2})\right] \quad (15)$$

Equations (12)-(15) can be implemented using an in-place butterfly structure shown in Figure 1. An example for calculating a 16-point DHT-II using the developed algorithm is shown in Figure 2.

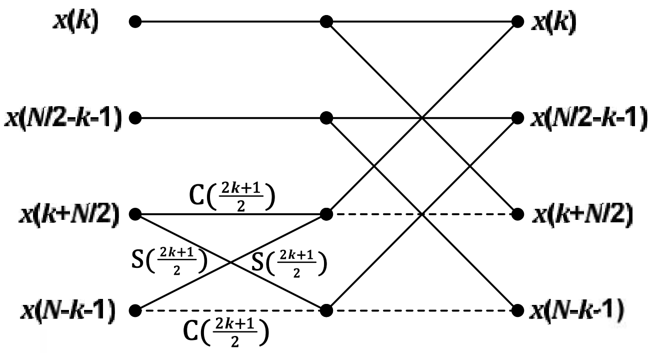

Figure 1. An in-place butterfly of radix-2 DHT-II DIT algorithm; where $C(i) = cos(2\pi i/N)$ and $S(i) = sin(2\pi i/N)$, solid and dotted lines stand for additions and subtraction respectively

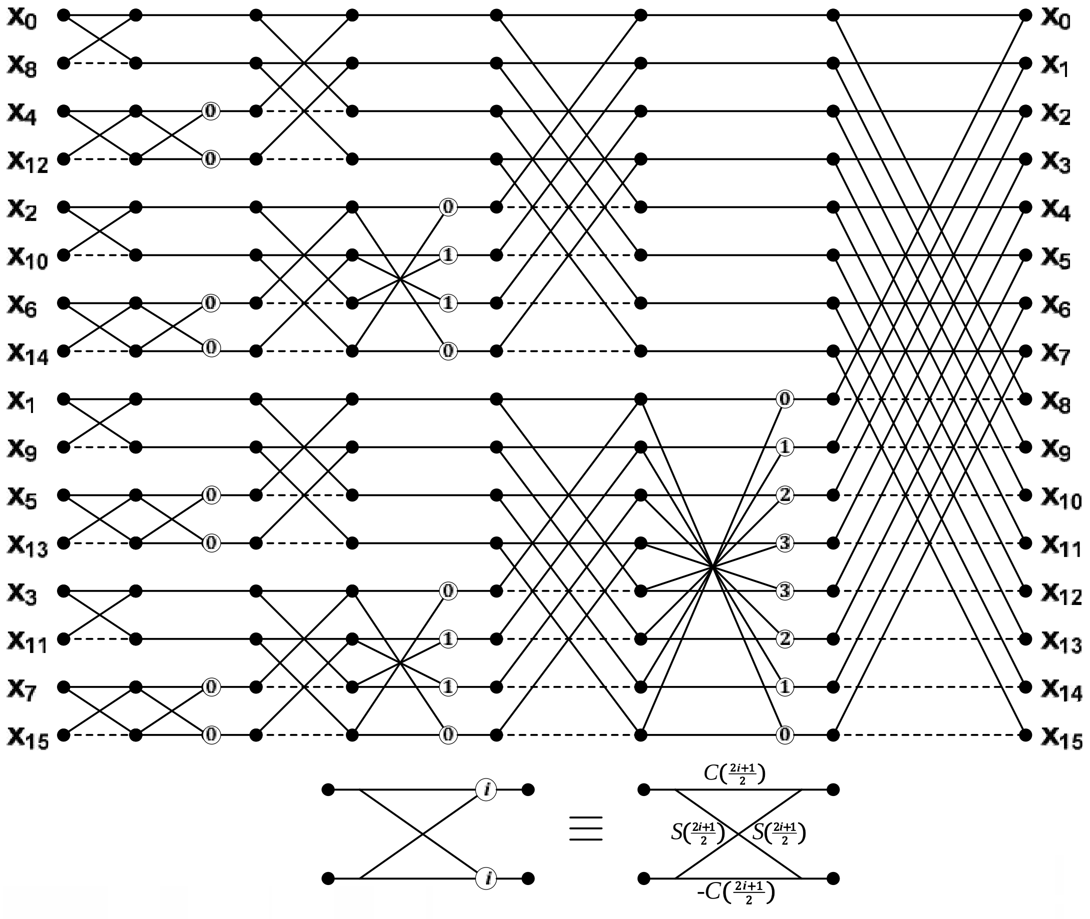

Figure 2. The calculation of 16-point DHT-II using radix-2 DIT approach; where $C(i) = cos(2\pi i/N)$and $S(i) = sin(2\pi i/N)$, solid and dotted lines stand for additions and subtraction respectively

3. Computational Complexity

The radix-2 DHT-II DIT algorithm combines four points together to formulate the in-place butterflyshown in Figure 1. Each butterfly calculates four points and requires four multiplications and six additions. In general, this algorithm requires $\log_2 N$ stages of butterfly computations in which each stage uses N multiplications and $3N/2$ additions. Therefore, the calculation of the whole transform is satisfies:

$$\begin{aligned} M\,(N) &= 2M\,(\tfrac{N}{2}) + N \\ A(N) &= 2A(\tfrac{N}{2}) + \tfrac{3N}{2} \end{aligned} \qquad (16)$$

where$M(N)$ and $A(N)$ stand for the number of multiplications and additions respectively.

It should be noted that (16) is for general calculation of the arithmetic complexity of radix-2 DIT algorithm using single butterfly; thus ±1 are considered as a twiddle factor that is counted as a multiplication. The total number of multiplications and additions could be reduced further using more than one butterfly.

The arithmetic complexities given by (16) are recursive. To obtain complexity in a closed form, the initial values of these complexities are required. In this case, the initial values can be the number of operations that are needed by length-4 DHTs-II, which are equal to $M(4)=2$ for multiplications and $A(4)=6$ for additions. Solving (16) by repeated substitution of the initial values and using multiple butterflies, we get:

$$\begin{aligned} M\,(N) &= N\log_2 N - \tfrac{3N}{2} \\ A(N) &= \tfrac{3N}{2}\log_2 N - \tfrac{3N}{2} \end{aligned} \qquad (17)$$

A comparison has been made between this algorithm and Hu’s radix-2 DIT algorithm [11] in terms of number of multiplications and additions, as shown in Table 1. The result of this comparison reveals that the total number of multiplications and additions of the proposed algorithm is better than Hu’s algorithm. Furthermore, comparison in terms of structural complexity for the signal flow graphs of the latter [Figure 1 of 11] with the former shown in Figure 2, we can easily deduce a high regularity and structural simplicity of the developed algorithm in contrast to Hu’s algorithm.

Table 1. Comparison of computational complexity for decimation-in-time DHT-II algorithms

Transform length *N*	Proposed Algorithm			Hu’s Algorithm [11]		
	Mults.	Adds.	Total	Mults.	Adds.	Total
8	10	22	32	8	28	36
16	34	66	110	28	84	112
32	98	178	276	84	220	304
64	258	450	708	228	540	768
128	642	1090	1732	580	1276	1856
256	1538	2562	4100	1412	2940	4352
512	3586	5890	9476	3332	6652	9984
1024	8194	13314	21508	7684	14844	22528

.

4. Conclusion

This paper has been focused on a new fast algorithm for direct computation of the DHT-II transform. The presented radix-2 decimation-in-time GFHT algorithm has a regular signal flow graph that provides flexibility for different transform lengths, substantially reducing the arithmetic complexity as compared with the indirect algorithms. The developed algorithm has been implemented through the DIT approach, and its computational complexity is analyzed and compared with existing algorithms, showing its significantly reduce the structural complexity with a better indexing scheme and ease implementation.

References

[1] G Bi and Y Zeng. *Transforms and Fast Algorithms for Signal Analysis and Representations*. Birkhauser Boston. 2004.

[2] V Britanak and KR Rao. “The fast generalized discrete Fourier transforms: A unified approach to the discrete sinusoidal transforms computation”. *Signal Process.* 1999; 79: 135-150.

[3] G Bongiovanni, P Corsini and G Frosini., "One-dimensional and two-dimensional generalized discrete Fourier transforms”. *IEEE Trans. Acoust., Speech, Signal Process.* 1976; 24: 97-99.

[4] Z Wang. “Fast algorithms for the discrete W transform and for the discrete Fourier transform”. *IEEE Trans. Acoust., Speech, Signal Process.* 1984; 32: 803-816.

[5] Z Wang and BR Hunt. “The discrete W transform”. *Appl. Math. Comput*. 1985; 16: 19-48.

[6] OK Ersoy. *Fourier-related transforms, fast algorithms, and applications*. Prentice Hall PTR. 1997.

[7] Z Wang, GA Jullien and WC Miller. “The generalized discrete W transform and its application to interpolation”. *Signal Process.* 1994; 36: 99-109.

[8] O Ersoy. “Semisystolic array implementation of circular, skew circular, and linear convolutions”. *IEEE Trans. Comput.* 1985; 34: 190-196.

[9] SA Martucci. “Symmetric convolution and the discrete sine and cosine transforms”. *IEEE Trans. Signal Process.* 1994; 42: 1038-1051.

[10] AK Jain. “A Sinusoidal Family of Unitary Transforms”. *IEEE Trans. Pattern Anal. Mach. Intell.* 1979; 1: 356-365.

[11] H Neng-Chung, IC Hong and OK Ersoy. “Generalized discrete Hartley transforms”. *IEEE Trans. Signal Process.* 1992; 40: 2931-2940.

[12] G Bi and Y Chen. “Fast generalized DFT and DHT algorithms”. *Signal Process.* 1998; 65: 383-390.

[13] DF Chiper. “Fast radix-2 algorithm for the discrete Hartley transform of type II”. *IEEE Signal Process. Lett.* 2011; 18(11): 687-689.

[14] HZ Shu, J Wu, C Yang and L Senhadji. “Fast radix-3 algorithm for the generalized discrete Hartley transform of type II”. *IEEE Signal Process. Lett.* 2012; 19(6): 348-351.

[15] G Bi, Y Chen and Y Zeng. “Fast algorithms for generalized discrete Hartley transform of composite sequence lengths”. *IEEE Trans. Circuits Syst. II, Exp. Brief*s. 2000; 47(9): 893-901.

[16] HZ Shu, Y Wang, L Senhadji and LM Luo. “Direct computation of type-II discrete Hartley transform”. *IEEE Signal Process. Lett.* 2007; 14(5): 329-332.