

## Direct computation of type-II discrete Hartley transform

Huazhong Shu, Yuan Wang, Lotfi Senhadji, Limin Luo

► **To cite this version:**

Huazhong Shu, Yuan Wang, Lotfi Senhadji, Limin Luo. Direct computation of type-II discrete Hartley transform. IEEE Signal Processing Letters, Institute of Electrical and Electronics Engineers, 2007, 14 (5), pp.329 - 332. inserm-00149853

**HAL Id: inserm-00149853**

**<https://www.hal.inserm.fr/inserm-00149853>**

Submitted on 29 May 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Direct computation of type-II discrete Hartley transform

Huazhong Shu, *Member, IEEE*, Yuan Wang, Lotfi Senhadji, *Senior Member, IEEE* and Limin Luo, *Senior Member, IEEE*

**Abstract**—We present in this letter an efficient direct method for the computation of a length- $N$  type-II generalized discrete Hartley transform (GDHT) when given two adjacent length- $N/2$  GDHT coefficients. The computational complexity of the proposed method is lower than that of the traditional approach for length  $N \geq 8$ . The arithmetic operations can be saved from 16% to 24% for  $N$  varying from 16 to 64. Furthermore, the new approach can be easily implemented.

**Index Terms**—Compressed-domain processing, discrete Hartley transform, fast algorithm

## I. INTRODUCTION

THE generalized discrete Hartley transform (GDHT) has been used in many digital signal and image processing applications. One advantage of the GDHT over the generalized discrete Fourier transform (GDFT) is that it can be used to avoid complex operations when the input sequence is real [1], [2]. It was shown [3] that the type-I, -II, -III, and -IV GDHT and the discrete W transform (DWT) have a similar definition with difference in constant scaling factors. Many fast methods for efficient computation of GDHT were reported in the literature. Hu [4] proposed a fast algorithm for computing the GDHT, Bi and Chen [5] derived a split-radix algorithm for the computation of GDFT and GDHT. Both methods were focused on the sequences with length  $N$  being power of two. Recently, Bi [6] developed fast algorithms for calculating GDHT of composite sequence lengths.

Since the GDHT is widely used in signal or image compression techniques, the following question arises: How can we directly manipulate such a compressed data stored in the GDHT domain? In other words, how can we construct a long GDHT sequence from several short GDHT sequences? This problem may be encountered when the signals are stored in the

Manuscript received July 5, 2006. This work was supported in part by National Basic Research Program of China under grant No. 2003CB716102 and Program for New Century Excellent Talents in University under grant No. NCET-04-0477. It has been carried out in the frame of the CRIBs, a joint international laboratory associating Southeast University, the University of Rennes 1 and INSERM, with a grant provided by the French Consulate in Shanghai.

Huazhong Shu, Yuan Wang and Limin Luo are with the Laboratory of Image Science and Technology, Institute of Computer Science and Engineering, Southeast University, 210096 Nanjing, China, and they are also with Centre de Recherche en Information Biomédicale Sino-Français (CRIBs) (e-mail: shu.list@seu.edu.cn; wangyuan@seu.edu.cn; luo.list@seu.edu.cn).

Lotfi Senhadji is with the Laboratoire Traitement du Signal et de l'Image, Université de Rennes 1 – INSERM U642, 35042 France, he is also with Centre

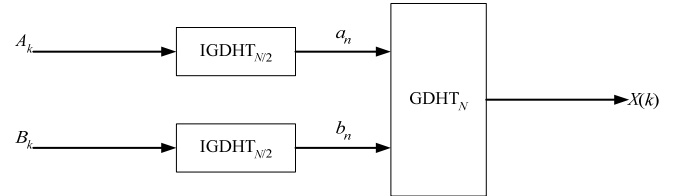


Fig. 1 Flow graph of traditional approach

GDHT domain. In such a case, the signals need to be processed before being displayed, transmitted, etc. Some of the frequently used processing operations are scaling, filtering, translation, and rotation. When the scaling operation is concerned, it is necessary to construct a long GDHT sequence from several short GDHT sequences or to decompose a long GDHT sequence into some short GDHT sequences in the compressed domain. We address this problem in the present letter. In particular, we are interested in the problem of the direct computation of the length- $N$  type-II GDHT coefficients, when the two adjacent  $N/2$  sets of type-II GDHT coefficients are given. The traditional way of implementing this calculation is shown in Fig. 1. According to this scheme, two inverse GDHTs of length- $N/2$  are required, followed by a length- $N$  type-II GDHT. Such a process may not be optimal in terms of the number of arithmetic operations. Inspired by a method proposed by Skodras [7], which was derived to the direct computation of  $N$ -point DCT when the two adjacent  $N/2$ -point DCT coefficients are known, we propose in this letter a new approach that is directly based on the GDHT domain for solving the above mentioned problem.

The rest of paper is organized as follows. Section II presents two new radix-2 algorithms for computing the  $N$ -point type-II GDHT coefficients. Section III describes the way of obtaining the  $N$ -point GDHT coefficients when given the two adjacent  $N/2$ -point GDHTs. The computational complexity of the proposed approach is analyzed in Section IV.

## II. THE TYPE-II GDHT ALGORITHMS FOR $N$ BEING EVEN NUMBER

The type-II GDHT of an input data sequence  $x(n)$ ,  $n = 0, 1, \dots, N-1$ , is defined as [4]

$$X(k) = \text{GDHT}_N^{\text{II}} \{x(n)\} = \sum_{n=0}^{N-1} x(n) \text{cas} \frac{\pi(2n+1)k}{N}, \quad k=0, 1, \dots, N-1, \quad (1)$$

de Recherche en Information Biomédicale Sino-Français (CRIBs) (e-mail: lotfi.senhadji@univ-rennes1.fr).

and the corresponding inverse GDHT (IGDHT) is given by [4]

$$x(n) = \text{IGDHT}_N^{\text{II}} \{X(k)\} = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \text{cas} \frac{\pi(2n+1)k}{N},$$

$$n=0, 1, \dots, N-1, \quad (2)$$

with  $\text{cas} \theta = \cos \theta + \sin \theta$ .

When the sequence length  $N$  is even, for the even index  $k$ , (1) can be decomposed into

$$\begin{aligned} X(2k) &= \sum_{n=0}^{N-1} x(n) \text{cas} \frac{2\pi(2n+1)k}{N} \\ &= \sum_{n=0}^{N/2-1} x(n) \text{cas} \frac{2\pi(2n+1)k}{N} + \sum_{n=N/2}^{N-1} x(n) \text{cas} \frac{2\pi(2n+1)k}{N} \\ &= \sum_{n=0}^{N/2-1} [x(n) + x(N/2+n)] \text{cas} \frac{\pi(2n+1)k}{N/2}, \end{aligned}$$

$$k=0, 1, \dots, N/2-1. \quad (3)$$

The above equation shows that  $X(2k)$  is a length- $N/2$  type-II GDHT. For the odd index, we have

$$X(2k+1) = \sum_{n=0}^{N-1} x(n) \text{cas} \frac{\pi(2n+1)(2k+1)}{N}. \quad (4)$$

By using the property

$$\text{cas}(\alpha + \beta) = \cos(\beta) \text{cas}(\alpha) + \sin(\beta) \text{cas}(-\alpha), \quad (5)$$

(4) can be expressed as

$$\begin{aligned} X(2k+1) &= \sum_{n=0}^{N-1} x(n) \cos \frac{(2n+1)\pi}{N} \text{cas} \frac{2\pi(2n+1)k}{N} \\ &\quad + \sum_{n=0}^{N-1} x(n) \sin \frac{(2n+1)\pi}{N} \text{cas} \frac{-2\pi(2n+1)k}{N} \end{aligned}$$

$$k=0, 1, \dots, N/2-1. \quad (6)$$

Making the change of variable  $n' = N-1-n$  in the second term of the right-hand side of (6), we obtain

$$\begin{aligned} X(2k+1) &= \sum_{n=0}^{N-1} \left[ x(n) \cos \frac{(2n+1)\pi}{N} - x(N-1-n) \sin \frac{(2n+1)\pi}{N} \right] \\ &\quad \times \text{cas} \frac{2\pi(2n+1)k}{N}. \end{aligned} \quad (7)$$

Proceeding in a similar way as for  $X(2k)$ , we have

$$\begin{aligned} X(2k+1) &= \sum_{n=0}^{N/2-1} \left\{ \left[ x(n) - x\left(\frac{N}{2}+n\right) \right] \cos \theta_n + \left[ x\left(\frac{N}{2}-1-n\right) - x(N-1-n) \right] \sin \theta_n \right\} \\ &\quad \times \text{cas} \frac{\pi(2n+1)k}{N/2} \\ &= \sum_{n=0}^{N/2-1} [a(n) \cos \theta_n + a(N/2-1-n) \sin \theta_n] \text{cas} \frac{\pi(2n+1)k}{N/2}, \end{aligned} \quad (8)$$

with  $a(n) = x(n) - x(N/2+n)$  and  $\theta_n = (2n+1)\pi/N$ .

(8) shows that  $X(2k+1)$  is also a length- $N/2$  type-II GDHT. Moreover, the computation of  $X(2k+1)$ , for each given  $k$ ,  $k=0, 1, \dots, N/2-1$ , needs  $N$  additions and  $N$  multiplications.

The above decomposition method is categorized as “decimation in frequency.” Its computational complexity is given by

$$M_{\text{GDHT}}^{\text{II}}(N) = 2M_{\text{GDHT}}^{\text{II}}(N/2) + N, \quad (9)$$

$$A_{\text{GDHT}}^{\text{II}}(N) = 2A_{\text{GDHT}}^{\text{II}}(N/2) + 3N/2.$$

Hu [4] proposed a “decimation in time” algorithm for computing the type-II GDHT coefficients. The computational complexity of their method is

$$M_{\text{GDHT}}^{\text{II}}(N) = 2M_{\text{GDHT}}^{\text{I}}(N/2) + 3N/2 - 4, \quad (10)$$

$$A_{\text{GDHT}}^{\text{II}}(N) = 2A_{\text{GDHT}}^{\text{I}}(N/2) + 5N/2 - 8.$$

where  $M_{\text{GDHT}}^{\text{I}}(N/2)$  and  $A_{\text{GDHT}}^{\text{I}}(N/2)$  are the number of multiplications and additions for the computation of type-I GDHT of length  $N/2$ .

It can be seen from (9) and (10) that the efficiency of the proposed method is comparable to that derived by Hu. Note that in [6], Bi derived a radix-2 algorithm for computing the type-II GDHT coefficients where  $X(2k)$  is calculated with equation (3), and  $X(2k+1)$  defined by (4) is transformed into a length- $N/2$  type-IV GDHT, and the latter can be further decomposed into two sequences of type-II GDHT with length  $N/4$ . The computational complexity of Bi's method is [6]

$$\begin{aligned} M_{\text{GDHT}}^{\text{II}}(N) &= M_{\text{GDHT}}^{\text{II}}(N/2) + M_{\text{GDHT}}^{\text{IV}}(N/2) \\ &= M_{\text{GDHT}}^{\text{II}}(N/2) + 2M_{\text{GDHT}}^{\text{II}}(N/4) + N, \end{aligned} \quad (11)$$

$$\begin{aligned} A_{\text{GDHT}}^{\text{II}}(N) &= A_{\text{GDHT}}^{\text{II}}(N/2) + A_{\text{GDHT}}^{\text{IV}}(N/2) + N \\ &= M_{\text{GDHT}}^{\text{II}}(N/2) + 2M_{\text{GDHT}}^{\text{II}}(N/4) + 2N. \end{aligned} \quad (12)$$

It was shown [6] that Bi's algorithm requires about the same computational complexity in terms of the total number of arithmetic operations as that needed in Hu's algorithm.

Although the above proposed method is efficient for computing the type-II GDHT coefficients, but for the purpose of this paper, we will derive another approach for computing  $X(2k+1)$  defined by (4). In fact, (4) can be rewritten as

$$X(2k+1) = X_1(k) + X_2(k), \quad (13)$$

where

$$\begin{aligned} X_1(k) &= \sum_{n=0}^{N-1} x(n) \cos \frac{(2n+1)\pi}{N} \text{cas} \frac{2\pi(2n+1)k}{N}, \\ &\quad k=0, 1, \dots, N/2-1, \end{aligned} \quad (14)$$

$$\begin{aligned} X_2(k) &= \sum_{n=0}^{N-1} x(n) \sin \frac{(2n+1)\pi}{N} \text{cas} \frac{-2\pi(2n+1)k}{N}, \\ &\quad k=0, 1, \dots, N/2-1. \end{aligned} \quad (15)$$

As for  $X(2k)$ ,  $X_1(k)$  can be decomposed into

$$X_1(k) = \sum_{n=0}^{N/2-1} [x(n) - x(N/2+n)] \cos \theta_n \text{cas} \frac{\pi(2n+1)k}{N/2}, \quad (16)$$

which is a length- $N/2$  type-II GDHT.

For  $X_2(k)$ ,  $k=0, 1, \dots, N/2-1$ , using the property  $\text{cas}(-\pi + \theta) = -\text{cas}(\theta)$ , we have

$$\begin{aligned}
X_2(N/2-k) &= \sum_{n=0}^{N-1} x(n) \sin \frac{(2n+1)\pi}{N} \operatorname{cas} \frac{-2\pi(2n+1)(N/2-k)}{N} \\
&= -\sum_{n=0}^{N-1} x(n) \sin \frac{(2n+1)\pi}{N} \operatorname{cas} \frac{2\pi(2n+1)k}{N}, \\
& \quad k = 0, 1, \dots, N/2-1, \quad (17)
\end{aligned}$$

thus

$$X_2(N/2-k) = -\sum_{n=0}^{N/2-1} [x(n) - x(N/2+n)] \sin \theta_n \operatorname{cas} \frac{\pi(2n+1)k}{N/2}, \quad (18)$$

which is also a length- $N/2$  type-II GDHT. Note that  $X_2(0) = -X_2(N/2)$ .

Since  $X_1(k)$  and  $X_2(k)$  are both  $N/2$ -point type-II GDHTs, we can obtain the sequence  $X(2k+1)$ ,  $k = 0, 1, \dots, N/2-1$ , from  $X_1(k)$  and  $X_2(k)$  with  $N/2$  additions. The computational complexity of the above method is given by

$$\begin{aligned}
M_{\text{GDHT}}^{\text{II}}(N) &= 3M_{\text{GDHT}}^{\text{II}}(N/2) + N, \\
A_{\text{GDHT}}^{\text{II}}(N) &= 3A_{\text{GDHT}}^{\text{II}}(N/2) + 3N/2.
\end{aligned} \quad (19)$$

(19) shows that the computation of  $X(2k+1)$  based on (16) and (18) requires much more arithmetic operations than that based on (8), but the advantage of the former algorithm is that it can be used to efficiently calculate the  $N$ -point GDHT from two adjacent  $N/2$ -point sequences of GDHT when the latter coefficients are given. This will be explained in the next section.

### III. PROPOSED APPROACH FOR COMPUTING THE LENGTH- $N$ TYPE-II GDHT COEFFICIENTS

Let us readdress the problem as follows. Assume an  $N$ -point sequence  $x(n)$  be created by the concatenation of two adjacent sequences of length  $N/2$ , i.e.,  $a_n = x(n)$ , and  $b_n = x(N/2+n)$ , and  $A_k$  and  $B_k$  are their length- $N/2$  type-II GDHT coefficients, respectively. How can we efficiently compute the  $N$ -point type-II GDHT coefficients  $X(k)$  when  $A_k$  and  $B_k$  are known? To answer this question, we propose a new approach based on the algorithm derived in the previous section.

#### A. Computation of $X(2k)$ , $k = 0, 1, \dots, N/2-1$

From (3), we have

$$X(2k) = A_k + B_k, \quad k = 0, 1, \dots, N/2-1. \quad (20)$$

#### B. Computation of $X_1(k)$ , $k = 0, 1, \dots, N/2-1$

From (16), we have

$$\begin{aligned}
X_1(k) &= \text{GDHT}_{N/2}^{\text{II}} \{ (a_n - b_n) \cos \theta_n \} \\
&= \text{GDHT}_{N/2}^{\text{II}} \{ \text{IGDHT}_{N/2}^{\text{II}} (A_k - B_k) \cos \theta_n \}.
\end{aligned} \quad (21)$$

#### C. Computation of $X_2(N/2-k)$ , $k = 0, 1, \dots, N/2-1$

From (18), we have

$$\begin{aligned}
X_2(N/2-k) &= -\text{GDHT}_{N/2}^{\text{II}} \{ (a_n - b_n) \sin \theta_n \} \\
&= -\text{GDHT}_{N/2}^{\text{II}} \{ \text{IGDHT}_{N/2}^{\text{II}} (A_k - B_k) \sin \theta_n \}.
\end{aligned} \quad (22)$$

According to the derivation above, we can construct the whole structure in Fig. 2.

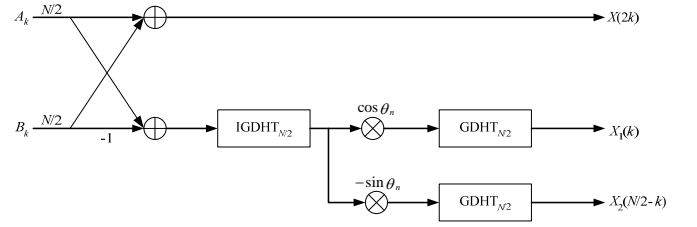


Fig. 2 Flow graph of the proposed method

#### D. Computation of $X(2k+1)$ , $k = 0, 1, \dots, N/2-1$

Once the values  $X_1(k)$  and  $X_2(N/2-k)$ ,  $k = 0, 1, \dots, N/2-1$ , are obtained, we can compute  $X(2k+1)$  by simply using equation (13). Note that for  $k = 0$ , we have  $X_2(0) = -X_2(N/2)$ .

### IV. COMPUTATIONAL COMPLEXITY

Fig. 2 shows that the computation of  $X(2k)$ ,  $X_1(k)$ , and  $X_2(N/2-k)$  requires two  $N/2$ -point vector multiplications and additions, respectively, and three  $N/2$ -point type-II GDHTs, assuming that both of GDHT and IGDHT are of the same complexity. On the other hand, the computation of  $X(2k+1)$  using (13) needs  $N/2$  additions. Therefore, the computational complexity of the proposed method is

$$\begin{aligned}
M_{\text{GDHT}}^{\text{P}}(N) &= 3M_{\text{GDHT}}^{\text{II}}(N/2) + N, \\
A_{\text{GDHT}}^{\text{P}}(N) &= 3A_{\text{GDHT}}^{\text{II}}(N/2) + 3N/2.
\end{aligned} \quad (23)$$

For comparison purpose, we choose  $N = 2^m$ , where  $m$  is a positive integer, so that the fast computation of  $N/2$ -point GDHT can be applied. According to the algorithm proposed by Hu [4], the number of operations for computing the type-II GDHT of size  $N = 2^m$  is given by

$$M_{\text{GDHT}}^{\text{P}}(N) = \frac{N}{4}(3\log_2 N - 2), \quad (24)$$

$$A_{\text{GDHT}}^{\text{P}}(N) = 3N \left( \frac{3}{4} \log_2 N - 1 \right) + 12.$$

Using (24), (23) becomes

$$M_{\text{GDHT}}^{\text{P}}(N) = \frac{N}{4}(3\log_2 N - 2), \quad (25)$$

$$A_{\text{GDHT}}^{\text{P}}(N) = 3N \left( \frac{3}{4} \log_2 N - 1 \right) + 12.$$

If the conventional approach shown in Fig. 1 is applied, two  $N/2$ -point IGDHTs and one  $N$ -point GDHT are required. Using the algorithm proposed by Hu to compute the  $N$ -point GDHT, the total numbers of additions and multiplications for the traditional approach are given by

$$\begin{aligned}
M_{\text{GDHT}}^{\text{T}}(N) &= 2M_{\text{GDHT}}^{\text{II}}(N/2) + M_{\text{GDHT}}^{\text{II}}(N) = N \left( \log_2 N - \frac{3}{2} \right) \\
A_{\text{GDHT}}^{\text{T}}(N) &= 2A_{\text{GDHT}}^{\text{II}}(N/2) + A_{\text{GDHT}}^{\text{II}}(N) = \frac{3N}{2}(2\log_2 N - 3) + 12.
\end{aligned} \quad (26)$$

Table 1 shows the number of arithmetic operations of the two methods for some specific values of  $N$ . It can be seen that about 7-24% arithmetic operations can be saved except for  $N = 4$ .

	Traditional approach ((4))	Proposed method	Save
--	----------------------------	-----------------	------

$N$	$M_N^T$	$A_N^T$	$M_N^r + A_N^r$	$M_N^p$	$A_N^p$	$M_N^r + A_N^r$	d
4	2	18	20	4	18	22	0%
8	12	48	60	14	42	56	7%
16	40	132	172	40	108	148	16%
32	112	348	460	104	276	380	21%
64	288	876	1164	256	684	940	24%

Table 1. Computational complexity of traditional and proposed methods

## V. CONCLUSIONS

We propose in this letter an efficient approach to compute a length- $N$  type-II GDHT given two adjacent length- $N/2$  type-II GDHT coefficients. The arithmetic operations can be saved from 16% to 24% for  $N$  varying from 16 to 64. Generally speaking, greater the value of  $N$ , more operations we can save. Therefore, the proposed method could find its application in signal processing tasks.

## ACKNOWLEDGMENT

We thank the anonymous referees for their careful review and valuable comments to improve the quality of the paper.

## REFERENCES

- [1] Z. Wang, "Harmonic analysis with a real frequency function, I Aperiodic case, II Periodic and bounded cases, and III Data sequence," *Appl. Math. Comput.*, vol. 9, pp. 53-73; 153-156; 245-255, 1981.
- [2] J. Xi and J. F. Chicharo, "Computing running discrete Hartley transform and running discrete W transforms based on the adaptive LMS algorithm," *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 257-260, 1997.
- [3] Z. Wang, "Comments on 'Generalized discrete Hartley transform'," *IEEE Trans. Signal Process.*, vol. 43, pp. 1711-1712, 1995.
- [4] N. C. Hu, H. I. Chang, and O. K. Ersoy, "Generalized discrete Hartley transforms," *IEEE Trans. Signal Process.*, vol. 40, no. 12, pp. 2931-2940, 1992.
- [5] G. Bi and Y. Chen, "Fast generalized DFT and DHT algorithms," *Signal Processing*, vol. 65, no. 3, pp. 383-390, 1998.
- [6] G. Bi, Y. Chen, and Y. Zeng, "Fast algorithms for generalized discrete Hartley transform of composite sequence lengths," *IEEE Trans. Circuits Syst. II*, vol. 47, no. 9, pp. 893-901, 2000.
- [7] A. N. Skodras, "Direct transform to transform computation," *IEEE Signal Process. Lett.*, vol. 6, no. 8, pp. 202-204, 1999.