



HAL
open science

Fast radix-3 algorithm for the generalized discrete Hartley transform of type II

Huazhong Shu, Jiasong Wu, Chufeng Yang, Lotfi Senhadji

► **To cite this version:**

Huazhong Shu, Jiasong Wu, Chufeng Yang, Lotfi Senhadji. Fast radix-3 algorithm for the generalized discrete Hartley transform of type II. *IEEE Signal Processing Letters*, 2012, 19 (6), pp.348 - 351. 10.1109/LSP.2012.2194782 . inserm-00692219

HAL Id: inserm-00692219

<https://inserm.hal.science/inserm-00692219>

Submitted on 29 Apr 2012

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.

Fast Radix-3 Algorithm for the Generalized Discrete Hartley Transform of Type II

Huazhong Shu, *Senior Member, IEEE*, Jiasong Wu, *Member, IEEE*, Chunfeng Yang, and Lotfi Senhadji, *Senior Member, IEEE*

Abstract—We present a new fast radix-3 algorithm for the computation of the length- N generalized discrete Hartley transform of type-II (GDHT-II), where $N = 3^m$, $m \geq 2$. Then we apply this algorithm to the direct computation of length- N GDHT-II coefficients when given three adjacent length- $N/3$ GDHT-II coefficients. The computational complexity of the proposed method is lower than that of the traditional approach for length $N \geq 9$. The arithmetic operations can be saved from 19% to 29% for $N = 3^m$ varying from 9 to 243 and from 17% to 29% for $N = 3 \times 2^m$ varying from 12 to 384. Furthermore, the new approach can be easily implemented.

Index Terms—Generalized discrete Hartley transform of type II, fast radix-3 algorithm, low computational complexity

I. INTRODUCTION

THE generalized discrete Hartley transform (GDHT) has been used in many digital signal and image processing tasks. The GDHT can be used as an alternative of the corresponding discrete Fourier transform 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 similar definitions with different scaling factors. Many fast algorithms for efficient computation of GDHT have been reported in the literature [4-7]. Among them, Hu [4] proposed a fast algorithm for computing the GDHT, which is devoted to

This work was supported by the National Basic Research Program of China under Grant 2011CB707904, by the National Natural Science Foundation of China under Grants 60873048, 60911130370 and 61073138, by the Ministry of Education of China under grant 20110092110023, the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, and by a Program of Jiangsu Province under Grant SBK200910055.

H. Shu is with the Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and also with the Centre de Recherche en Information Biomédicale Sino-Français (CRIBs), Nanjing 210096, China (e-mail: shu.list@seu.edu.cn).

J. Wu is with the Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, the Centre de Recherche en Information Biomédicale Sino-Français (CRIBs), Nanjing 210096, China, INSERM U 1099, 35000 Rennes, France, and the Laboratoire Traitement du Signal et de l'Image (LTSI), Université de Rennes 1, 35000 Rennes, France (e-mail: jswu@seu.edu.cn).

C. Yang and L. Senhadji are with INSERM U1099, 35000 Rennes, France, with the Laboratoire Traitement du Signal et de l'Image (LTSI), Université de Rennes 1, 35000 Rennes, France, and with the Centre de Recherche en Information Biomédicale Sino-Français (CRIBs), 35000 Rennes, France (e-mail: cfyang1981@gmail.com, lotfi.senhadji@univ-rennes1.fr).

sequences with length N being power of two. Zhao [6] derived a fast radix-3 decimation in frequency algorithm for computing the type-I GDHT. Bi [7] developed fast algorithms for calculating GDHT of composite sequence lengths.

In signal or image compression technique using the GDHT, 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 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.

In a previous work [8], we have addressed this problem by considering the direct computation of the length- N type-II GDHT (GDHT-II) coefficients, when the two adjacent $N/2$ sets of GDHT-II coefficients are given. Recently, Chiper [9] has made an improvement on the Shu's algorithm.

Since both algorithms reported in [8] and [9] deal with the case of two sequences, we now focus on the case of three adjacent sequences in this letter. It is worth noting that the same problem has been already treated by Pei and Kao in DCT domain [10]. By combining the algorithm proposed in this paper with Chiper's method, many GDHT domain operations, such as fractional scaling by the factor of $1/2^m 3^n$, can be realized in a simple and efficient way.

The rest of the letter is organized as follows. Section II presents a new fast radix-3 algorithm for computing the GDHT-II. In Section III, we propose the way to obtain the N -point GDHT-II coefficients when given the three adjacent $N/3$ -point GDHTs' coefficients. The analysis of the computational complexity is given and comparison with the traditional scheme is also provided in Section IV. Finally, Section V concludes the paper.

II. RADIX-3 ALGORITHM FOR FAST COMPUTATION OF GDHT-II

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

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

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

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

If we form (1) for all integers $k+jN$ with $k = 0, 1, \dots, N-1$, and $j \in \mathbb{Z}$, then $X(k+jN) = (-1)^j X(k)$. Thus, we see that $X(-1) = -X(N-1)$ and $X(N) = -X(0)$. When the sequence length N is a multiple of three, the GDHT-II coefficients $X(k)$ defined by (1) can be computed as follows.

A. *Computation of $R(k) = X(3k)$, $k = 0, 1, \dots, N/3 - 1$.*

We have

$$\begin{aligned} R(k) &= \sum_{n=0}^{N-1} x(n) \text{cas} \frac{\pi(2n+1)k}{N/3} \\ &= \sum_{n=0}^{N/3-1} \left[x(n) + \left(x\left(\frac{N}{3} + n\right) + x\left(\frac{2N}{3} + n\right) \right) \right] \\ &\quad \times \text{cas} \frac{\pi(2n+1)k}{N/3}. \end{aligned} \quad (3)$$

B. *Computation of $S(k) = X(3k+1) + X(3k-1)$, $k = 0, 1, \dots, N/3 - 1$.*

Using $\text{cas}(x+y) + \text{cas}(x-y) = 2 \cos(y) \text{cas}(x)$, we obtain

$$\begin{aligned} S(k) &= \sum_{n=0}^{N-1} x(n) \left[\text{cas} \frac{\pi(2n+1)(3k+1)}{N} \right. \\ &\quad \left. + \text{cas} \frac{\pi(2n+1)(3k-1)}{N} \right] \\ &= 2 \sum_{n=0}^{N-1} x(n) \cos \theta_n \text{cas} \frac{\pi(2n+1)k}{N/3} \\ &= \sum_{n=0}^{N/3-1} \left[u(n) \cos \theta_n - \sqrt{3}v(n) \sin \theta_n \right] \\ &\quad \times \text{cas} \frac{\pi(2n+1)k}{N/3}, \end{aligned} \quad (4)$$

where

$$u(n) = 2x(n) - \left(x\left(\frac{N}{3} + n\right) + x\left(\frac{2N}{3} + n\right) \right), \quad (5)$$

$$v(n) = x\left(\frac{N}{3} + n\right) - x\left(\frac{2N}{3} + n\right), \quad (6)$$

$$\theta_n = \frac{\pi(2n+1)}{N}. \quad (7)$$

C. *Computation of $T(k) = X(3k+1) - X(3k-1)$, $k = 0, 1, \dots, N/3 - 1$.*

Applying

$\text{cas}(x+y) - \text{cas}(x-y) = 2 \sin(y) \text{cas}(-x)$, we obtain

$$\begin{aligned} T(k) &= \sum_{n=0}^{N-1} x(n) \left[\text{cas} \frac{\pi(2n+1)(3k+1)}{N} \right. \\ &\quad \left. - \text{cas} \frac{\pi(2n+1)(3k-1)}{N} \right] \\ &= 2 \sum_{n=0}^{N-1} x(n) \sin \theta_n \text{cas} \frac{-\pi(2n+1)k}{N/3} \\ &= \sum_{n=0}^{N-1} \left[u(n) \sin \theta_n + \sqrt{3}v(n) \cos \theta_n \right] \\ &\quad \times \text{cas} \frac{-\pi(2n+1)k}{N/3}. \end{aligned} \quad (8)$$

From $\text{cas}(x-\pi) = -\text{cas}(x)$, it follows that

$$\begin{aligned} T(N/3 - k) &= T_1(k) = \\ &= \sum_{n=0}^{N/3-1} \left[-u(n) \sin \theta_n - \sqrt{3}v(n) \cos \theta_n \right] \text{cas} \frac{\pi(2n+1)k}{N/3}. \end{aligned} \quad (9)$$

Hence we obtain $T(k) = T_1(N/3 - k)$, $k = 0, 1, \dots, N/3 - 1$, where $T(0) = T_1(N/3) = -T_1(0)$ and $T(N/3) = T_1(0) = -T(0)$. Thus it follows from equations (3), (4) and (9) that the coefficients $R(k)$, $S(k)$ and $T_1(k)$ can be computed by length- $N/3$ GDHT-II. Note that $S(0) = -S(N/3)$ and $T(0) = -T(N/3)$.

D. *Computation of $X(3k+1)$ and $X(3k-1)$, $k = 0, 1, \dots, N/3 - 1$.*

These coefficients can be computed as follows

$$\begin{aligned} X(3k+1) &= (S(k) + T(k))/2, \\ X(3k-1) &= (S(k) - T(k))/2, \\ &\quad k = 0, 1, 2, \dots, N/3 - 1. \end{aligned} \quad (10)$$

The decomposition method described above is categorized as ‘decimation in frequency’. Its arithmetic cost for $N = 3^n$, $n \geq 2$ is given by

$$\begin{aligned} M_{\text{GDHT}}^{\text{II}}(N) &= 3M_{\text{GDHT}}^{\text{II}}(N/3) + 4N/3 - 3 \\ &= (4 \log_3 N - 9/2)N/3 + 3/2, \\ A_{\text{GDHT}}^{\text{II}}(N) &= 3A_{\text{GDHT}}^{\text{II}}(N/3) + 8N/3 - 2 \\ &= (8 \log_3 N - 5)N/3 + 1, \end{aligned} \quad (11)$$

where $M_{\text{GDHT}}^{\text{II}}(N)$ and $A_{\text{GDHT}}^{\text{II}}(N)$ denote the number of real multiplications and real additions required for computation of length- N GDHT-II, respectively. The initial values are $M_{\text{GDHT}}^{\text{II}}(3) = 1$ and $A_{\text{GDHT}}^{\text{II}}(3) = 4$. Note that multiplications with powers of 2 and permutations are not counted in (11).

Note also that when $n = (N/3 - 1)/2$, we have $\cos \theta_n = 1/2$ and $\sin \theta_n = \sqrt{3}/2$. The inputs of (4) and (9) become

$$\left[x(n) - x(n + 2N/3) \right] - 2 \left[x(n + N/3) - x(n + 2N/3) \right]$$

and $-\sqrt{3}[x(n) - x(n + 2N/3)]$, respectively. In this case, 3 multiplications and 2 additions can be saved.

Therefore, the proposed algorithm has the same arithmetic cost as Bi's 'decimation in time' GDHT-II algorithm [7].

III. DIRECT COMPUTATION OF LENGTH- N GDHT-II BY LENGTH- $N/3$ TRANSFORMS

Let N be divisible by 3. The problem we will consider in this section is described as follows: Assume an N -point sequence $x(n)$ be created by the concatenation of three adjacent sequences of length $N/3$, that is, $a_n = x(n)$, $b_n = x(N/3 + n)$, and $c_n = x(2N/3 + n)$, and A_k , B_k and C_k are their length- $N/3$ GDHT-II coefficients, respectively. How can we efficiently compute the N -point GDHT-II coefficients $X(k)$ when A_k , B_k , and C_k are known? The schematic representation of the traditional approach is shown in Fig. 1. We propose a new approach based on the algorithm derived in the previous section.

A. Computation of $X(3k)$, $k = 0, 1, \dots, N/3 - 1$.

From (3), we have

$$X(3k) = A_k + (B_k + C_k), \quad k = 0, 1, \dots, N/3 - 1. \quad (12)$$

B. Computation of $S(k)$, $k = 0, 1, \dots, N/3$.

From equations (4) to (6), we have

$$\begin{aligned} S(k) &= \text{GDHT}_{N/3}^{\text{II}} \{u(n) \cos \theta_n - v(n) \sin \theta_n\} \\ &= \text{GDHT}_{N/3}^{\text{II}} \{[\text{GDHT}_{N/3}^{\text{II}} [2A_k - D_k] \cos \theta_n \\ &\quad - \sqrt{3} \text{GDHT}_{N/3}^{\text{II}} [E_k] \sin \theta_n]\} \end{aligned} \quad (13)$$

where

$$D_k = B_k + C_k, \quad E_k = B_k - C_k. \quad (14)$$

C. Computation of $T(k)$, $k = 0, 1, \dots, N/3$.

Similar as for $S(k)$, we can deduce from (9) that

$$\begin{aligned} T(N/3 - k) &= -\text{GDHT}_{N/3}^{\text{II}} \{u(n) \sin \theta_n + \sqrt{3}v(n) \cos \theta_n\} \\ &= -\text{GDHT}_{N/3}^{\text{II}} \{[\text{GDHT}_{N/3}^{\text{II}} [2A_k - D_k] \sin \theta_n \\ &\quad + \sqrt{3} \text{GDHT}_{N/3}^{\text{II}} [E_k] \cos \theta_n]\} \end{aligned} \quad (15)$$

When all these coefficients are computed, the coefficients $X(3k+1)$ and $X((3k-1) \bmod N)$ for $0 \leq k \leq N/3 - 1$, can be obtained recursively by using (10). The schematic representation of the proposed method is depicted in Fig. 2.

IV. COMPLEXITY ANALYSIS AND COMPARISON RESULTS

For a sequence of length- N , the computational complexity of IGDHT-II is that of GDHT-II plus N multiplications if the scaling factor $1/N$ is not the reciprocal of power of 2. A fast radix-3 algorithm for IGDHT-II can be found in [7]. For simplicity, we just choose two types of sequence length for the comparison of arithmetic cost.

1) Case $N = 3^m$, $m \geq 2$.

According to Fig. 2, the arithmetic cost of the proposed algorithm is given by

$$\begin{aligned} M_N^P &= 2M_{\text{IGDHT}}^{\text{II}}(N/3) + 2M_{\text{GDHT}}^{\text{II}}(N/3) + 4N/3 - 3 \\ &= 4M_{\text{GDHT}}^{\text{II}}(N/3) + 2N - 3, \\ A_N^P &= 2A_{\text{IGDHT}}^{\text{II}}(N/3) + 2A_{\text{GDHT}}^{\text{II}}(N/3) + 8N/3 - 2 \\ &= 4A_{\text{GDHT}}^{\text{II}}(N/3) + 8N/3 - 2. \end{aligned} \quad (16)$$

According to Fig. 1, the computational complexity of the traditional approach is as follows

$$\begin{aligned} M_N^T &= 3M_{\text{IGDHT}}^{\text{II}}(N/3) + M_{\text{GDHT}}^{\text{II}}(N) \\ &= 6M_{\text{GDHT}}^{\text{II}}(N/3) + 7N/3 - 3, \\ A_N^T &= 3A_{\text{IGDHT}}^{\text{II}}(N/3) + A_{\text{GDHT}}^{\text{II}}(N) \\ &= 6A_{\text{GDHT}}^{\text{II}}(N/3) + 8N/3 - 2. \end{aligned} \quad (17)$$

The same initial values are used as those of (11). Because the scaling factor $1/(N/3)$ in IGDHT-II is not the reciprocal of power of 2, so, its computation is taken into account. The comparison results of the proposed algorithm and traditional approach are shown in Table I. It can be concluded from the table that the proposed method reduces the total number of arithmetic operations from 19% to 29% for N varying from 9 to 243 when compared to the traditional approach. Note that the computational complexity shown in Table I is referring to the direct method but not to the radix-3 algorithm.

2) Case $N = 3 \times 2^m$, $m \geq 2$.

According to Fig. 2, the computational complexity of the proposed algorithm is given by

$$\begin{aligned} M_N^P &= 4M_{\text{GDHT}}^{\text{II}}(N/3) + 4N/3, \\ A_N^P &= 4A_{\text{GDHT}}^{\text{II}}(N/3) + 8N/3. \end{aligned} \quad (18)$$

Note that when $N = 3 \times 2^m$, the special angle $\theta_n = \pi/3$ can not be reached.

According to Fig. 1, the computational complexity of the traditional approach is as follows

$$\begin{aligned} M_N^T &= 6M_{\text{GDHT}}^{\text{II}}(N/3) + 4N/3, \\ A_N^T &= 6A_{\text{GDHT}}^{\text{II}}(N/3) + 8N/3. \end{aligned} \quad (19)$$

We then use Bi's algorithm to compute length- $N/3=2^m$ GDHT-II for $m \geq 2$, whose computational complexity can be found in [7]. Note that in this case the scaling factor $1/(N/3)$ in IGDHT-II is the reciprocal of power of 2, so, it does not need multiplication. The comparison results of the proposed algorithm and traditional approach are shown in Table II. It can be concluded from the table that the proposed method reduces the total number of arithmetic operations from 17% to 29% for N varying from 12 to 384 when compared to the traditional approach.

V. CONCLUSION

We have proposed in this letter an efficient method for the direct computation of length- N GDHT-II from three adjacent

length- $N/3$ GDHT-II coefficients. Results show that the proposed method allows reducing the number of arithmetic operations compared to the traditional approach. Since the method proposed by Chiper [9] for the direct computation of GDHT-II is also available, its combination with the proposed method could find applications in signal processing tasks. In fact, scaling a GDHT-II-based signal by the factor of $1/(2^m 3^n)$ can be easily obtained by cascading Chiper's structure and that presented in this letter.

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] Z.J. Zhao, "In-place radix-3 fast Hartley transform algorithm," *Electron. Lett.*, vol. 28, no.3, pp. 319-321, 1992.
- [7] 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.
- [8] H.Z. Shu, Y. Wang, L. Senhadji, and L.M. Luo, "Direct computation of type-II discrete Hartley transform," *IEEE Signal Process. Lett.*, vol. 14, no. 5, pp. 329-332, 2007.
- [9] D.F. Chiper, "Fast radix-2 algorithm for the discrete Hartley transform of type II," *IEEE Signal Process. Lett.*, vol. 18, no. 11, pp. 687-689, 2011.
- [10] S.C. Pei and M.P. Kao, "Direct N -point DCT computation from three adjacent $N/3$ -point DCT coefficients," *IEEE Signal Process. Lett.*, vol. 12, no. 2, pp. 89-92, 2005.

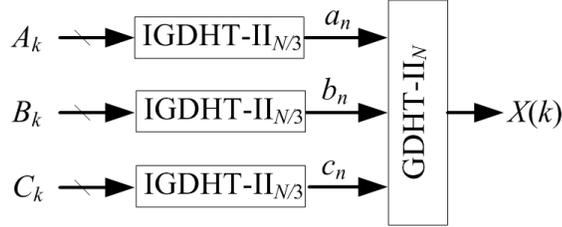


Fig. 1. Schematic representation of traditional approach

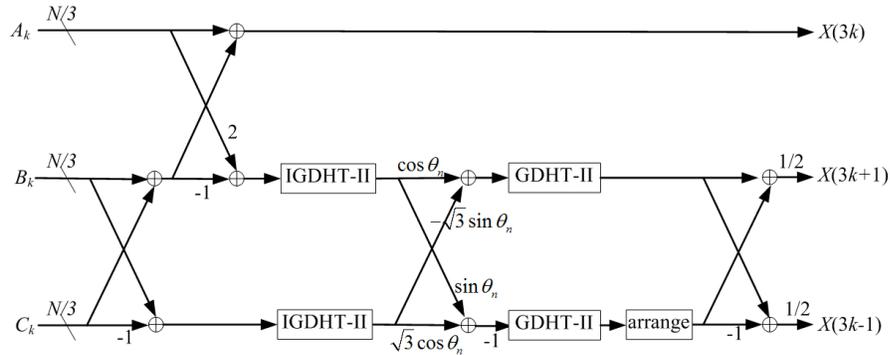


Fig. 2. Schematic representation of the proposed method ("arrange" denotes arranging $T_1(k)$ to get $T(k)$ in (9))

Table I Computational complexity of the proposed algorithm and the traditional approach for $N = 3^m$, $m \geq 2$. ($N < 500$)

N	Traditional approach ([7])			Proposed algorithm			Saved (%)
	M_N^T	A_N^T	$M_N^T + A_N^T$	M_N^P	A_N^P	$M_N^P + A_N^P$	
9	24	46	70	19	38	57	19
27	132	274	406	99	206	305	25
81	600	1246	1846	435	902	1337	28
243	2436	5026	7462	1731	3566	5297	29

Table II Computational complexity of the proposed algorithm and the traditional approach for $N = 3 \times 2^m$, $m \geq 2$. ($N < 500$)

N	Traditional approach ([7])			Proposed algorithm			Saved (%)
	M_N^T	A_N^T	$M_N^T + A_N^T$	M_N^P	A_N^P	$M_N^P + A_N^P$	
12	28	68	96	24	56	80	17
24	68	220	288	56	168	224	22
48	220	548	768	168	408	576	25
96	548	1372	1920	408	1000	1408	27
192	1372	3236	4608	1000	2328	3328	28
384	3236	7516	10752	2328	5352	7680	29