Direct Computation of Length-N DHT from Three Adjacent Length-N/3 DHT Coefficients

Jiasong Wu, Lu Wang, Huazhong Shu
Laboratory of Image Science and Technology (LIST)
Southeast University
Nanjing, China
Centre de Recherche en Information Biomédical Sino-Français, Nanjing, China

Lotfi Senhadji
INSERM, U642, Rennes, France
LTSI, Université de Rennes 1,
Rennes, France
Centre de Recherche en Information Biomédical Sino-Français, Rennes, France

***Abstract*—A fast direct method for obtaining the length-N discrete Hartley transform (DHT) coefficients from three adjacent length-N/3 DHT coefficients is presented. The proposed method reduces significantly the number of arithmetic operations compared to the traditional approach. Furthermore, it is easy to implement.**

Keywords-compressed-domain processing; discrete Hartley transform; fast algorithm

I. INTRODUCTION

The discrete Hartley transform (DHT) introduced by Bracewell in 1983 [1] has become an important tool in signal and image processing [2-5]. Many fast algorithms for computing the DHT have already been reported in the literature. Sorensen, Jones, Burrus, and Heideman [6] proposed a set of fast algorithms for DHT, including radix-2 decimation-in-frequency (DIF), radix-4, split radix, prime factor [7], and Winograd transform algorithms. Zhao [8] derived a radix-3 fast Hartley transform (FHT) algorithm, which was then extended to 2-D DHT [9]. Bi [10] suggested a radix-3/9 FHT algorithm for further speeding up the computation of DHT. Bouguezel, Ahmad, and Swamy [11] proposed an optimized split-radix FHT algorithm, taking arithmetic complexity, the number of data transfers, address generations, and twiddle factor evaluations into consideration, for computing the DHT of sequences with length $N = q\times2^m$, where q is an odd integer.

Since the DHT can be used in signal or image compression techniques [2-4], a problem that is often encountered is how to directly manipulate a compressed data stored in the DHT domain. In particular, how to construct a long DHT sequence from several short DHT sequences? This kind of questions was first discussed in discrete cosine transform (DCT) domain by Kou and Fjallbrant [12], who proposed an efficient way for computing the length-N DCT coefficients when the two adjacent length-$N/2$ DCT coefficients are known. Their algorithm was then improved by Skodras [13], whose algorithm was further improved and extended to two dimensional DCT by Chuang and Wu [14] and multidimensional DCT by Dai, Chen, and Lin [15]. Jiang and Feng [16] discussed the spatial relationship between an image block of any size and all of its subblocks in the DCT domain. In [17], Pei and Kao presented an efficient method for obtaining the length-N DCT coefficients from three adjacent length-N/3 DCT coefficients. More recently, Shu, Wang, Senhadji, and Luo [18] realized a direct computation of length-N type-II generalized discrete Hartley transform (GDHT) coefficients from two adjacent length-N/2 GDHT coefficients. Shu's algorithm can be easily extended to the computation of DHT. In this paper, we address the problem of how to efficiently compute the length-N DHT coefficients from three consecutive length-N/3 DHT coefficients. Specifically, assume a length-N sequence x_n is created by the concatenation of three length-N/3 adjacent sequences, i.e. $a_n = x_n$, $b_n = x_{n+N/3}$, $c_n = x_{n+2N/3}$, for $n = 0, 1, \ldots, N/3-1$, and A_k, B_k, and C_k are their DHT coefficients, respectively. How can we efficiently compute the length-N DHT coefficients X_k when A_k, B_k, and C_k are known? We develop a new method, which is similar to the approach presented in [17], to solve the above mentioned question in the DHT domain.

II. TRADITIONAL APPROACH

The schematic representation of the traditional approach is shown in Fig. 1. According to the scheme, three length-N/3 inverse DHTs (IDHTs) and one length-N DHT are required. In this section, we apply two different schemes, based respectively on the algorithms presented in [8] and [11], to implement the length-N DHT according to the different values of N.

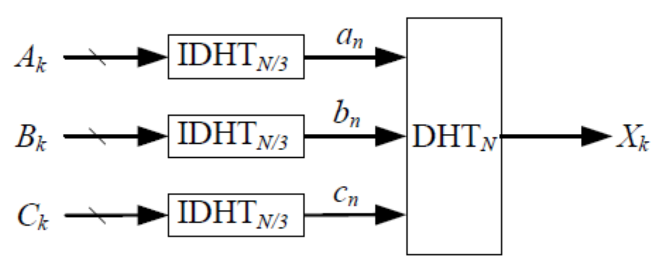

Figure 1. Schematic representation of traditional approach

A. First scheme: Decomposing length-N DHT into three length-N/3 DHTs by using Zhao's algorithm [8].

When N is divisible by 3, i.e., $N = 3p$, we can apply Zhao's DIF radix-3 FHT algorithm [8] to decompose length-N DHT.

into three length-$N/3$ DHTs. We will reformulate Zhao's algorithm and analyze its computational complexity in the rest of this subsection.

The normalized DHT of the sequence x_n is defined by [1]

$$X_k = \mathrm{DHT}_N\{x_n\} = \sum_{n=0}^{N-1} x_n \operatorname{cas}\frac{2\pi nk}{N}, \quad k = 0, 1, \ldots, N-1, \tag{1}$$

and the corresponding IDHT is

$$x_n = \mathrm{IDHT}_N\{X_k\} = \frac{1}{N}\sum_{k=0}^{N-1} X_k \operatorname{cas}\frac{2\pi nk}{N}, \quad n = 0, 1, \ldots, N-1, \tag{2}$$

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

According to [8], we can realize three formulations $D_k = X_{3k}$, $E_k = X_{3k+1}+X_{3k-1}$, and $F_{N/3-k} = X_{3k+1}-X_{3k-1}$, for $k = 0, 1, \ldots, N/3-1$, where $X_{-1}=X_{N-1}$, $X_0=X_N$ and $X_1=X_{N+1}$, to obtain the DHT coefficients X_k instead of computing (1) directly.

$$D_k = X_{3k} = \sum_{n=0}^{N/3-1}(a_n + b_n + c_n)\operatorname{cas}\frac{2\pi nk}{N/3}, \tag{3}$$

$$\begin{aligned} E_k &= X_{3k+1} + X_{3k-1} \\ &= \sum_{n=0}^{N/3-1}\Big\{[2a_n - (b_n + c_n)]\cos\theta_n \\ &\qquad - \sqrt{3}(b_n - c_n)\sin\theta_n\Big\}\operatorname{cas}\frac{2\pi nk}{N/3}, \end{aligned} \tag{4}$$

$$\begin{aligned} F_k &= X_{3(N/3-k)+1} - X_{3(N/3-k)-1} \\ &= \sum_{n=0}^{N/3-1}\Big\{[2a_n - (b_n + c_n)]\sin\theta_n \\ &\qquad + \sqrt{3}(b_n - c_n)\cos\theta_n\Big\}\operatorname{cas}\frac{2\pi nk}{N/3}, \end{aligned} \tag{5}$$

where a_n, b_n, and c_n are the three adjacent sequences and

$$\theta_n = 2\pi n / N. \tag{6}$$

X_{3k+1} and X_{3k-1} can be easily obtained as follows

$$\begin{aligned} X_{3k+1} &= (E_k + F_{N/3-k})/2, \\ X_{3k-1} &= (E_k - F_{N/3-k})/2. \end{aligned} \quad k = 0, 1, \ldots, N/3-1. \tag{7}$$

$F_{N/3-k}$ can be easily got by arranging F_k, where $F_0 = F_{N/3}$.

Generally, the computation of $\left\{[2a_n - (b_n + c_n)]\cos\theta_n - \sqrt{3}(b_n - c_n)\sin\theta_n\right\}$ in (4) and $\left\{[2a_n - (b_n + c_n)]\sin\theta_n + \sqrt{3}(b_n - c_n)\cos\theta_n\right\}$ in (5) requires 4 multiplications and 5 additions for each n. Moreover, when $n = 0$, we have $\cos\theta_n = 1$ and $\sin\theta_n = 0$. In this case, 3 multiplications and 2 additions can be saved. Thus, the computational complexity of Zhao's algorithm is given by [8]

$$\begin{aligned} M_N^{\mathrm{DHT}} &= 3M_{N/3}^{\mathrm{DHT}} + 4N/3 - 3, \\ A_N^{\mathrm{DHT}} &= 3A_{N/3}^{\mathrm{DHT}} + 8N/3 - 2. \end{aligned} \tag{8}$$

B. Second scheme: Direct computation of length-N DHT by using Bouguezel's algorithm [11]

When the sequence length $N = q\times 2^m$, where q is an odd integer, we can use Bouguezel's split-radix FHT algorithm [11] to compute the length-N DHT directly. In the following, we assume that a butterfly computation is implemented by using 3 multiplications and 3 additions. The computational complexity of Bouguezel's algorithm for length-$N = q\times 2^m$ is given by

$$\begin{aligned} M_N^{\mathrm{DHT}} &= M_{N/2}^{\mathrm{DHT}} + 4M_{N/8}^{\mathrm{DHT}} + N - 8q, \\ A_N^{\mathrm{DHT}} &= A_{N/2}^{\mathrm{DHT}} + 4A_{N/8}^{\mathrm{DHT}} + 3N - 8q. \end{aligned} \quad N>8q. \tag{9}$$

Detailed computational complexity for the DHT with length-$N = q\times 2^m$, $q = 1, 3, 9, 15\ldots$, can be found in [11]. Note that $M_2^{\mathrm{DHT}} = 0$, $A_2^{\mathrm{DHT}} = 2$, $M_4^{\mathrm{DHT}} = 0$, and $A_4^{\mathrm{DHT}} = 8$ ..

III. Proposed Method

In this section, we propose a new method which is based on Zhao's radix-3 FHT algorithm to calculate the length-N DHT coefficients from three consecutive length-$N/3$ DHT coefficients.

A. Calculation of $D_k = X_{3k}$, $k = 0, 1, \ldots, N/3-1$.

From (3), we have

$$D_k = X_{3k} = A_k + B_k + C_k \tag{10}$$

B. Calculation of $E_k = X_{3k+1}+X_{3k-1}$, $k = 0, 1, \ldots, N/3-1$.

From (4), we have

$$\begin{aligned} E_k = \mathrm{DHT}_{N/3}\Big\{&\mathrm{IDHT}_{N/3}\{2A_k - (B_k + C_k)\}\cos\theta_n \\ &- \mathrm{IDHT}_{N/3}\{B_k - C_k\}\sqrt{3}\sin\theta_n\Big\} \end{aligned} \tag{11}$$

C. Calculation of $F_{N/3-k} = X_{3k+1}-X_{3k-1}$, $k = 0, 1, \ldots, N/3-1$.

From (5), we have

$$\begin{aligned} F_k = \mathrm{DHT}_{N/3}\Big\{&\mathrm{IDHT}_{N/3}\{2A_k - (B_k + C_k)\}\sin\theta_n \\ &+ \mathrm{IDHT}_{N/3}\{B_k - C_k\}\sqrt{3}\cos\theta_n\Big\} \end{aligned} \tag{12}$$

Once E_k and F_k are computed, X_{3k+1} and X_{3k-1} can be obtained by (7). The schematic representation of the proposed method is depicted in Fig. 2.

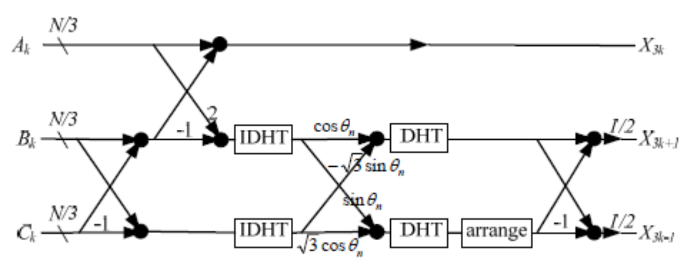

Figure 2. Schematic representation of the proposed method ("●" denotes the adder).

IV. Computational Complexity

We now analyze the computational complexity of the proposed method and the traditional approach. Since the same kernel is used for the DHT and the IDHT, they require the same computational complexity.

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

$$\begin{aligned} M_N^P &= 4N/3 - 3 + 4M_{N/3}^{\mathrm{DHT}}, \\ A_N^P &= 8N/3 - 2 + 4A_{N/3}^{\mathrm{DHT}}. \end{aligned} \tag{13}$$

The computational complexity of the traditional approach shown in Fig. 1 is as follows

$$\begin{aligned} M_N^T &= 3M_{N/3}^{\mathrm{DHT}} + M_N^{\mathrm{DHT}}, \\ A_N^T &= 3A_{N/3}^{\mathrm{DHT}} + A_N^{\mathrm{DHT}}. \end{aligned} \tag{14}$$

A. Traditional approach using Zhao's algorithm[8] (First Scheme)

When Zhao's algorithm is used in the traditional approach, substitution of (8) into (14) yields

$$\begin{aligned} M_N^{T_1} &= 6M_{N/3}^{\mathrm{DHT}} + 4N/3 - 3, \\ A_N^{T_1} &= 6A_{N/3}^{\mathrm{DHT}} + 8N/3 - 2. \end{aligned} \tag{15}$$

1) When length-$N = 3\times 2^m$, $m \geq 2$.

In this case, the split radix FHT algorithm [11] can be used to efficiently compute length-$N/3 = 2^m$ DHTs where the computational complexity is shown in (9). The comparison results of the proposed algorithm and this scheme are shown in table I.

2) When length-N is other values

In this case, the prime factor DHT algorithm [6][7] can be used to efficiently compute length-$N/3$ DHTs. The computational complexity of this algorithm can be found in [7]. Note that the number of additions involved in computing the DHT via prime-factor FFT algorithm for real-valued series [7] is calculated by adding (N-2) to those involved in computing the DFT. The number of multiplications required for computing the DHT is the same as those of the DFT computation. The comparison results of the proposed algorithm and this scheme are shown in table II.

B. Traditional approach using Bouguezel's algorithm[11] (Second Scheme)

When Bouguezel's algorithm is used in the traditional approach, setting q = 3 in (9), and then substituting the results into (14), we have

$$\left.\begin{aligned} M_N^{T_2} &= 3M_{N/3}^{\mathrm{DHT}} + M_{N/2}^{\mathrm{DHT}} + 4M_{N/8}^{\mathrm{DHT}} + N - 24, \\ A_N^{T_2} &= 3A_{N/3}^{\mathrm{DHT}} + A_{N/2}^{\mathrm{DHT}} + 4A_{N/8}^{\mathrm{DHT}} + 3N - 24. \end{aligned}\right. \quad N > 24. \tag{16}$$

We summarize the computational complexity of the proposed method and the second scheme of traditional approach in table III.

TABLE I. Computational complexity of the proposed method and the first scheme of traditional approach (N<500).

N	**Traditional approach ([8],[11])**			**Proposed method**			**Saved (%)**
	$M_N^{T_1}$	$A_N^{T_1}$	$M_N^{T_1}+A_N^{T_1}$	M_N^P	A_N^P	$M_N^P+A_N^P$	
12	13	78	91	13	62	75	18
24	41	218	259	37	166	203	22
48	121	570	691	101	422	523	24
96	329	1418	1747	261	1030	1291	26
192	841	3402	4243	645	2438	3083	27
384	2045	7946	9991	1533	5638	7171	28

TABLE II. Computational complexity of the proposed method and the first scheme of traditional approach (N<500).

N	**Traditional approach ([7],[8])**			**Proposed method**			**Saved (%)**
	$M_N^{T_1}$	$A_N^{T_1}$	$M_N^{T_1}+A_N^{T_1}$	M_N^P	A_N^P	$M_N^P+A_N^P$	
6	5	26	31	5	22	27	13
9	21	52	73	17	42	59	19
12	13	78	91	13	62	75	18
15	47	134	181	37	102	139	23
21	73	264	337	57	194	251	26
24	41	218	259	37	166	203	22
27	93	316	409	73	234	307	25
33	161	584	745	121	418	539	28
36	93	382	475	77	286	363	24
39	169	672	841	129	482	611	27
45	207	598	805	157	438	595	26
48	121	570	691	101	422	523	24
51	275	1070	1345	205	758	963	28
57	301	1260	1561	225	890	1115	29
60	197	806	1003	157	590	747	26
63	309	1060	1369	233	762	995	27
72	225	946	1171	181	694	875	25
84	301	1422	1723	237	1022	1259	27
90	417	1390	1807	317	1006	1323	27
105	587	2066	2653	437	1470	1907	28
108	381	1726	2107	301	1246	1547	27
120	457	1914	2371	357	1382	1739	27
144	561	2290	2851	437	1654	2091	27
168	689	3266	3955	533	2326	2859	28
180	837	3142	3979	637	2254	2891	27
216	873	3994	4867	677	2854	3531	27
240	1097	4490	5587	837	3206	4043	28
252	1245	4750	5995	941	3390	4331	28
312	1529	7466	8995	1157	5254	6411	29
315	2187	7474	9661	1597	5262	6859	29
336	1633	7458	9091	1237	5270	6507	28
360	1857	7186	9043	1397	5110	6507	28
420	2357	9974	12331	1757	7022	8779	29
432	2073	9178	11251	1573	6502	8075	28

TABLE III. Computational complexity of the proposed method and the second scheme of traditional approach (N<500).

N	**Traditional approach ([11])**			**Proposed method**			**Saved (%)**
	$M_N^{T_2}$	$A_N^{T_2}$	$M_N^{T_2}+A_N^{T_2}$	M_N^P	A_N^P	$M_N^P+A_N^P$	
12	4	72	76	13	62	75	1
24	18	204	222	37	166	203	9
48	74	540	614	101	422	523	15
96	234	1356	1590	261	1030	1291	19
192	642	3276	3918	645	2438	3083	21
384	1652	7692	9344	1533	5638	7171	23

As stated in [19], for modem computers, the run time for a multiplication is almost the same as that for an addition. They are exactly the same for some advanced signal processors, such as the series of TMS320. Therefore, we give the savings of the proposed algorithm in terms of the total number of multiplications and additions. It can be concluded from the tables that the proposed method reduces significantly the number of arithmetic operations compared to the first scheme of traditional approach. Although the reduction in terms of the arithmetic operations of the proposed method is less remarkable compared to the second scheme, it needs only to compute two length-$N/3$ DHTs and IDHTs, which can be implemented with one structure for sequences with length-2^m [11]. On the other hand, for the second scheme of traditional approach, the computation of three length-$N/3$ IDHTs needs one structure with length-2^m, but the computation of length-N DHT requires another structure with length-3×2^m. Furthermore, the proposed algorithm supports a more wide range of choices of sequence length when compared to the second scheme whose length is limited to length-3×2^m.

V. CONCLUSIONS

We have proposed in this paper an efficient method for the direct computation of length-N DHT from three adjacent length-$N/3$ DHT coefficients. Results show that the proposed method allows reducing the number of arithmetic operations in comparison with the traditional approaches. Another important feature of the proposed algorithm is its savings in memory and data transfers. Since the method proposed by Shu, Wang, Senhadji, and Luo [18] for the direct computation of type-II GDHT is also applicable to the DHT, its combination with the proposed method could find applications in signal processing tasks. In fact, scaling a DHT-based signal by the factor of $1/(2^p3^q)$ can be easily obtained by cascading p for Shu's structures and q for the authors' structures.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 60873048, by the National Basic Research Program of China under Grant 2010CB732503, by a Program of Jiangsu Province under Grant SBK200910055 and the Natural Science Foundation of Jiangsu Province under Grant BK2008279. This work was also supported by the "Eiffel Doctorate" excellence grant of the French Ministry of Foreign and European Affairs.

REFERENCES

[1] R. N. Bracewell, "Discrete Hartley transform," *J. Opt. Soc. Amer.*, vol. 73, pp. 1832-1835, 1983.

[2] I. Duleba, "Hartley transform in compression of medical ultrasonic images," in *Proc. IEEE ICIAP*, Sept. 1999, pp. 722-727.

[3] P.K. Meher, T. Srikanthan, J. Gupta, and H.K. Agarwal, "Near lossless image compression using lossless Hartley like transform," in *Proc. IEEE ICICS-PCM*, Dec. 2003, pp. 213-217.

[4] R.S. Sunder, C. Eswaran, and N. Sriraam, "Medical image compression using 3-D Hartley transform," *Computers in Biology and Medicine*, vol. 36, pp. 958-973, Sept. 2006.

[5] L.Y. Jiang, H.Z. Shu, J.S. Wu, L. Wang, L. Senhadji, "A novel split-radix fast algorithm for 2-D discrete Hartley transform," *IEEE Trans. Circuits Syst.-I: Regular papers*, vol. 57, pp. 911-924, Apr. 2010.

[6] H.V. Sorensen, D. L. Jones, C.S. Burrus, M.T. Heideman, "On computing the discrete Hartley transform," *IEEE Trans. Acoust. Speech Signal Process.*,vol. 33, pp. 1231-1238, Oct. 1985.

[7] M.T. Heideman, C. S. Burrus, and H.W. Johnson, "Prime-factor FFT algorithms for real-valued series," *Proc. ICASSP*, pp. 28A.7.1-28A.7.4, 1984.

[8] Z.J. Zhao, "In-place radix-3 fast Hartley transform algorithm," *Electron. Lett.,* vol. 28, pp. 319-321, Jan. 1992.

[9] J.S.Wu, H.Z. Shu, L. Senhadji, L.M. Luo, "Radix-3×3 algorithm for the 2-D discrete Hartley transform," *IEEE Trans. Circuits Syst.-II: Express Briefs*, vol. 55, pp. 566-570, Jun. 2008.

[10] G. Bi, "Radix-3/9 FHT algorithm," *Electron. Lett.*, vol. 31, pp. 166-168, Feb. 1995.

[11] S. Bouguezel, M.O. Ahmad, M.N.S. Swamy, "A new split-radix FHT algorithm for length-$q*2^m$ DHTs," *IEEE Trans. Circuits Syst.-I: Regular papers*, vol. 51, pp. 2031-2043, Oct. 2004.

[12] W. Kou and T. Fjallbrant, "A direct computation of DCT coefficients for a signal block taken from two adjacent blocks," *IEEE Trans. Signal Process.,* vol. 39, pp. 1692-1695, July 1991.

[13] A. N. Skodras, "Direct transform to transform computation," *IEEE Signal Process. Lett.*, vol. 6, pp. 202-204, Aug. 1999.

[14] Y.J. Chuang and J.L. Wu, "Direct splitting and merging of 2-D DCT in the DCT domain," *Digital Signal Process.*, vol. 14, pp. 614-624, Nov. 2004.

[15] Q. Dai, X. Chen, C. Lin, "Fast algorithms for multidimensional DCT-to-DCT computation between a block and its associated subblocks," *IEEE Trans. Signal Process.,* vol. 53, pp. 3219-3225, Aug. 2005.

[16] J. J. Jiang and G. Feng, "The spatial relationship of DCT coefficients between a block and its subblocks," *IEEE Trans. Signal Process.*, vol. 50, pp. 1160–1169, May 2002

[17] 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, pp. 89-92, Feb. 2005.

[18] H.Z. Shu, Y. Wang, L. Senhadji, and L. Luo, "Direct computation of type-II discrete Hartley transform," *IEEE Signal Process. Lett.*, vol. 14, pp. 329-332, May 2007.

[19] Z. Wang, "A prime factor fast W transform algorithm," *IEEE Trans. Signal Process.,* vol. 40, pp. 2361-2368, July 1991.