Two-band fast Hartley transform

A.N. Skodras, M.F. Aburdene and A.K. Nandi

Efficient algorithms have been developed over the past 30 years for computing the forward and inverse discrete Hartley transforms (DHTs). These are similar to the fast Fourier transform (FFT) algorithms for computing the discrete Fourier transform (DFT). Most of these methods seek to minimise the complexity of computations and/or the number of operations. A new approach for the computation of the radix-2 fast Hartley transform (FHT) is presented. The proposed algorithm, based on a two-band decomposition of the input data, possesses a very regular structure, avoids the input or out data shuffling, requires slightly less multiplications than the existing approaches, but increases the number of additions.

Introduction: Hartley presented a new method, the continuous Hartley transform, for the analysis of transmission problems in 1942 [1]. Subsequently, Bracewell introduced the discrete Hartley transform (DHT) in 1983 [2] and the fast Hartley transform (FHT) in 1984 [3]. In the intervening years, many researchers have devised methods to improve the computation of the FHT and the highly similar inverse FHT [4–8], whereas others have tried to develop recursive [9] and/or parallel methods for computing the FHT [10]. The DHT is commonly used in signal processing, signal compression, image classification, image encryption and communication systems [6, 7, 9].

This Letter proposes a two-band method, an entirely new approach for computing the FHT, resulting in a highly regular structure with butterflies of constant geometry and a reduced multiplication operations count compared with existing algorithms, while increasing the additions operations count.

Discrete Hartley transform: The type II DHT of the N-point real-valued data x_n, $n = 0, 1, 2, \ldots, N-1$ is defined as

$$H_N(k) = \sum_{n=0}^{N-1} x(n)\,\mathrm{cas}\left(\frac{2\pi}{N}nk\right), \quad k = 0, 1, 2, \ldots, N-1 \tag{1}$$

where $\mathrm{cas}(\cdot) = \cos(\cdot) + \sin(\cdot)$. The transform is linear and its coefficients $H_N(k)$ are real numbers. The DHT is its own inverse (involuntary), up to an overall scale factor of $1/N$. Calculating the DHT directly requires $O(N^2)$ real operations, i.e. N^2 real multiplications and $N(N-1)$ real additions. Fast algorithms for different radices, similar to the fast Fourier transform, have been proposed over the past 30 years, in an attempt to reduce the complexity to $O(N \log N)$. The most common are the radix-2 decimation-in-time and decimation-in-frequency FHT [3, 4] with $(NP - 3N + 4)$ multiplications and $(3NP - 3N + 4)/2$ additions and the split-radix FHT with $(2NP/3) - (19N/9) + 3 + (-1)^P(1/9)$ multiplications and $(4NP/3) - (14N/9) + 3 + (-1)^P(5/9)$ additions, where $P = \log_2 N$ [4]. Other approaches that appeared have achieved an improvement on the above complexity figures at the expense of a more complicated computational structure [5, 8, 9]. Fast algorithms are usually in-place, resulting in a shuffling of the input data or the output coefficients.

Discrete Hartley transform: The proposed method is based on the decomposition of each pair of input data $x(2n)$, $x(2n+1)$ into low-band values $x_L(n)$ and high-band values $x_H(n)$. Specifically

$$x_L(n) = \frac{1}{2}[x(2n) + x(2n+1)] \tag{2a}$$

$$x_H(n) = \frac{1}{2}[x(2n) - x(2n+1)] \tag{2b}$$

or

$$x(2n) = x_L(n) + x_H(n) \tag{3a}$$

$$x(2n+1) = x_L(n) - x_H(n) \tag{3b}$$

Starting from the definition of DHT (1), decomposing into even-indexed and odd-indexed data and using (3a) and (3b), leads to

$$\begin{aligned} H_N(k) &= \sum_{n=0}^{N/2-1} x(2n)\,\mathrm{cas}\left(\frac{2\pi k}{N}2n\right) + \sum_{n=0}^{N/2-1} x(2n+1)\,\mathrm{cas}\left(\frac{2\pi k}{N}(2n+1)\right) \\ &= \sum_{n=0}^{N/2-1} [x_L(n) + x_H(n)]\,\mathrm{cas}\left(\frac{2\pi k}{N}2n\right) \\ &\quad + \sum_{n=0}^{N/2-1} [x_L(n) - x_H(n)]\,\mathrm{cas}\left(\frac{2\pi k}{N}(2n+1)\right) \end{aligned} \tag{4}$$

By taking into account that

$$\mathrm{cas}(A+B) = \mathrm{cas}A\,\cos B + \mathrm{cas}(-A)\sin B \tag{5}$$

(4) becomes

$$H_N(k) = H_L(k) + [\cos\vartheta\, H_H(k) + \sin\vartheta\, H_H(-k)] \tag{6}$$

where

$$\vartheta = 2\pi k/N \tag{7}$$

$$H_L(k) = H^L_{N/2}(k) + H^H_{N/2}(k) \tag{8}$$

$$H_H(k) = H^L_{N/2}(k) - H^H_{N/2}(k) \tag{9}$$

$$H_H(-k) = H^L_{N/2}(-k) - H^H_{N/2}(-k) \tag{10}$$

and

$$H^L_{N/2}(k) = \sum_{n=0}^{N/2-1} x_L(n)\,\mathrm{cas}\left(\frac{2\pi k}{N/2}n\right) \tag{11}$$

$$H^H_{N/2}(k) = \sum_{n=0}^{N/2-1} x_H(n)\,\mathrm{cas}\left(\frac{2\pi k}{N/2}n\right) \tag{12}$$

$H^L_{N/2}(k)$ and $H^H_{N/2}(k)$ are the $N/2$-point DHTs of $x_L(n)$ and $x_H(n)$, respectively. On the basis of the properties of the $\cos(\cdot)$, $\sin(\cdot)$ and $\mathrm{cas}(\cdot)$ functions, it is easily derived that the DHT coefficients of (6) that are $N/2$ positions apart from k become

$$H_N\left(k + \frac{N}{2}\right) = H_L(k) - [\cos\vartheta\, H_H(k) + \sin\vartheta\, H_H(-k)] \tag{13}$$

Equations (6) and (13) constitute the DHT pair for its fast computation. By noting that

$$H_H(-k) = H_H\left(\tfrac{N}{2} - k\right) \tag{14}$$

the DHT pair of (6) and (13) eventually becomes

$$H_N(k) = H_L(k) + \left[\cos\vartheta\, H_H(k) + \sin\vartheta\, H_H\left(\tfrac{N}{2} - k\right)\right] \tag{15}$$

$$H_N\left(k + \frac{N}{2}\right) = H_L(k) - \left[\cos\vartheta\, H_H(k) + \sin\vartheta\, H_H\left(\tfrac{N}{2} - k\right)\right] \tag{16}$$

From (15) and (16), we can realise that the computation of an N-point DHT has been decomposed into two DHTs of length $N/2$ each, combined with $(N-1)$ multiplications by twiddle factors, as depicted in the flow graph of Fig. 1. By eliminating the trivial multiplications occurring for k equal to 0, $N/4$ and $(N/2-k)$ the flow graph is further simplified.

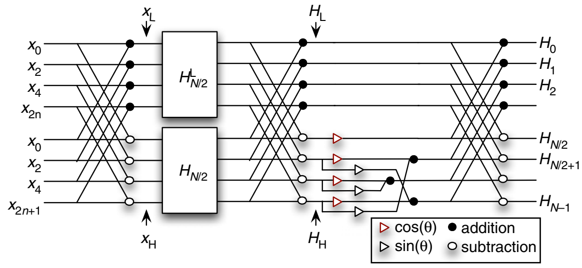

Fig. 1 *Flow graph for computation of N-point two-band fast DHT by means of (15) and (16)*

Note: for clarity purposes input data $x(n)$ and output coefficients $H(k)$ are denoted as x_n and H_k, respectively.

The flow graphs for the computation of the 8-point and 16-point two-band fast DHTs are depicted in Figs. 2 and 3, respectively. As in all fast DHT algorithms, only two additions or subtractions are needed for the 2-point FHT, i.e. $M_2 = 0$ and $A_2 = 2$. For the 4-point DHT only eight additions/subtractions are required, i.e. $M_4 = 0$ and $A_4 = 8$.

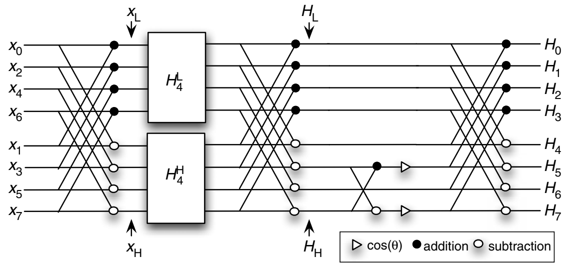

Fig. 2 *Flow graph for computation of 8-point two-band fast DHT*

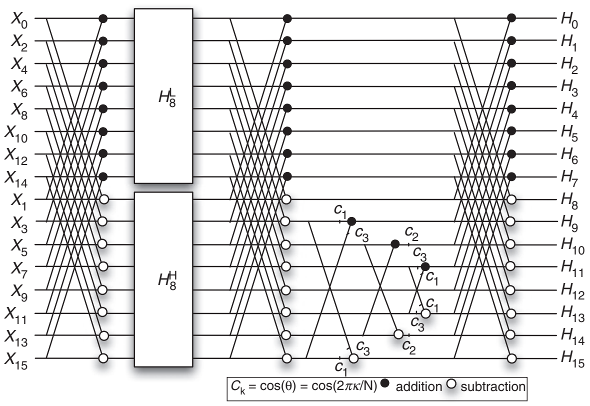

Fig. 3 *Flow graph for computation of 16-point two-band fast DHT*

The structure is simple, regular, modular and scalable, which facilitates software or hardware implementations. The structure resembles 'constant geometry' of fast transforms. Constant geometry algorithms avoid the area and delay overhead of multiplexing different registers, something that is desirable in high-throughput designs.

Table 1: Operation counts of two-band FHT

N	Multiplications	Additions
4	0	8
8	2	42
16	14	138
32	54	386
64	166	994
128	454	2434
256	1158	5762
512	2822	13 314
1024	6662	30 210
2048	15 366	67 586
4096	34 822	149 506

Computational complexity: On the basis of (15) and (16) and the corresponding flow graphs, we can easily derive the number of operations needed for the computation of the two-band N-point fast DHT. This is calculated by means of the formulae

$$A_N = 2A_{N/2} + 3N + N/2 - 2 \quad (17)$$

or

$$A_N = (7N/2)\log_2 N - (11/2)N + 2 \quad (18)$$

$$M_N = 2M_{N/2} + N - 4 \quad (19)$$

where M_N is the number of multiplications and A_N is the number of the total additions and subtractions. Another two multiplications could be saved for $k = N/8$ and $k = (N/2 - N/8)$. In the first case ($k = N/8$), we have $\theta = \pi/4$ and thus $\cos(\theta) = \sin(\theta)$, i.e. only one multiplication is needed; in the second case [$k = (N/2 - N/8)$], $\theta = 3\pi/4$ and thus $\cos(\theta) = -\sin(\theta)$, i.e. only one multiplication is performed. Taking into account this additional saving, the multiplications count reduces to

$$M_N = 2M_{N/2} + N - 6 \quad (20)$$

or

$$M_N = N \log_2 N - (7/2)N + 6, \ \text{for } N \geq 8 \quad (21)$$

The counts of multiplications and additions/subtractions for different N, where N is a power of 2, are summarised in Table 1.

The multiplications counts are less than those of the corresponding well-known radix-2 algorithm of Bracewell [3]. The additions are approximately twice as large. It should be noted that a final multiplication of each coefficient by 1/2 is needed in order for the result to be correct, as dictated by (2).

Conclusions: A new two-band radix-2 algorithm has been proposed for the computation of the fast DHT. The algorithm proceeds by applying the DHT core on the summations and the differences of adjacent samples, i.e. on the low-band and high-band values of adjacent samples. This is equivalent to applying a two-band filter bank followed by a down-sampling by 2. The computational structure is simplified and symmetric, at the expense of increased numbers of additions and subtractions. Multiplications are restricted only to the highs and their count is slightly decreased. The derived structure of the algorithm facilitates its fast implementation in high-level or low-level applications. The computation is in-place and compares favourably with the well-known fast DHT algorithms.

5 September 2014
doi: 10.1049/el.2014.3170

A.N. Skodras (*Electrical and Computer Engineering, University of Patras, Patras, Achaia, Greece*)

E-mail: skodras@upatras.gr

M.F. Aburdene (*Electrical Engineering, Bucknell University, Lewisburg, PA, USA*)

A.K. Nandi (*Department of Electronic and Computer Engineering, Brunel University, Uxbridge, Middlesex, UK*)

References

1 Hartley, R.V.L.: 'A more symmetrical Fourier analysis applied to transmission problems', *Proc. IRE*, 1942, **30**, (3), pp. 144–150

2 Bracewell, R.N.: 'Discrete Hartley transform', *J. Opt. Soc. Am.*, 1983, **73**, (12), pp. 1832–1835

3 Bracewell, R.N.: 'The fast Hartley transform', *Proc. IEEE*, 1984, **72**, (8), pp. 1010–1018

4 Sorensen, H.V., Jones, D.L., Burrus, C.S., and Heideman, M.: 'On computing the discrete Hartley Transform', *IEEE Trans. Acoust. Speech Signal Process.*, 1985, **33**, (5), pp. 1231–1238

5 Bouguezel, S., Ahmad, M.O., and Swamy, M.N.S.: 'A new split-radix FHT algorithm for length-$q \times 2m$ DHTs', *IEEE Trans. Circuits Syst. I*, 2004, **51**, (10), pp. 2031–2043

6 Hamood, M.T., and Boussakta, S.: 'New radix-based FHT algorithm for computing the discrete Hartley transform'. IEEE Int. Conf. on Acoustics, Speech and Signal Proc. (ICASSP), 22–27 May 2011, pp. 1581–1584

7 Chiper, D.F.: 'Fast radix-2 algorithm for the discrete Hartley transform of type II', *IEEE Signal Proc Lett.*, 2011, **18**, (11), pp. 687–689

8 de Oliveira, R.C., Campello de Souza, R.M., and de Oliveira, H.M.: 'Matrix expansions for computing the discrete Hartley transform for blocklength $N \equiv 0 \pmod 4$)'. IEEE Latin-America Conf. on Communications (LATINCOM), 7–9 November 2012, pp. 1–6

9 Yu, P., and Hua, H.: 'A new fast recursive algorithm for computing discrete Hartley transform and its implementation', *Signal Process.*, 2001, **81**, (10), pp. 2235–2241

10 Chiper, D.F.: 'A novel VLSI DHT algorithm for a highly modular and parallel architecture', *IEEE Trans. Circuits Syst. II, Express Briefs*, 2013, **60**, (5), pp. 282–286

11 Shah, G.A., and Rathore, T.S.: 'A new fast radix-2 DIF algorithm and architecture for computing the DHT', *Int. J. Simul. Syst. Sci. Technol.*, 2011, **12**, (1), pp. 1–9