IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, May 9-10, 1991

New Formulation of Fast Discrete Hartley Transform with the Minimum Number of Multiplications

Yuk-Hee CHAN and Wan-Chi SIU

Department of Electronic Engineering
Hong Kong Polytechnic
Hung Hom, Kowloon, Hong Kong

Abstract: The discrete Hartley transform(DHT) is a real-valued transform closely related to the discrete Fourier transform (DFT) of a real-valued sequence. It directly maps a real-valued sequence to a real-valued spectrum while preserving some useful properties of the Discrete Fourier Transform. In such case, the Discrete Hartley transform can act as an alternative form to the Fourier Transform for avoiding complex arithmetic, hence it becomes a valuable tool in digital signal processing. In this paper, a simple algorithm is proposed to realize one-dimensional DHT with sequence lengths equal to 2^m. *This algorithm achieves the same multiplicative complexity as Malvar's algorithm which requires the minimum number of multiplications reported in the literature. However, the present approach gives the advantage of requiring a smaller number of additions compared with the number that required in Malvar's algorithm.*

INTRODUCTION

The discrete Hartley transform (DHT) has received growing interest since it was introduced by Bracewell[1] in 1983. One of the main attractions of DHT is that it only involves real computations in contrast to complex computations in the discrete Fourier transform (DFT). Since then, most of works that had been done on the problem of computing the discrete Hartley transform[2-9] were based upon the idea of existing fast DFT algorithms. For instance, the so-called "split-radix" method was originally applied for computing 2^m-point DFT[10] and was subsequently applied to the computation of DHT[4,8]. Typically, among these proposed algorithms, Malvar[5,6]'s algorithm can achieve the minimum number of multiplications during the realization of DHT whilst the split-radix approach[4] can achieve the minimum number of additions.

In this paper, we propose a new algorithm to realize the DHT with the minimum number of multiplications. The algorithm also achieves the minimum number of additions compared with those algorithms[5,6,9] achieving the same minimum number of multiplications reported in literatures.

ALGORITHM DERIVATION

Recall that an N-length Discrete Hartley Transform[1] for a sequence $\{x(i):i=0,1...N-1\}$ is defined as

$$X(k) = \sum_{i=0}^{N-1} x(i)\ \mathrm{cas}\left(\frac{2\pi ik}{N}\right) \qquad \text{for } k=0,1...N-1 \quad (1)$$

If N is even, we have

$$X(2k) = \sum_{i=0}^{N/2-1} \{x(i)+x(N/2+i)\}\ \mathrm{cas}\left(\frac{4\pi ik}{N}\right) \qquad \text{for } k=0,1...N/2-1 \quad (2)$$

and

$$X(2k+1) = \sum_{i=0}^{N/2-1} \{x(i)-x(N/2+i)\}\ \mathrm{cas}\left(\frac{2\pi i(2k+1)}{N}\right) \qquad \text{for } k=0,1...N/2-1 \quad (3)$$

By defining the two sequences Y(k) and Z(k):

$$Y(k) = \sum_{i=0}^{N/2-1} y(i)\ \cos\left(\frac{2\pi i(2k+1)}{N}\right) \qquad \text{for } k=0,1...N/4-1 \quad (4)$$

$$Z(k) = \sum_{i=0}^{N/2-1} y(i)\ \sin\left(\frac{2\pi i(2k+1)}{N}\right) \qquad \text{for } k=0,1...N/4-1 \quad (5)$$

where $y(i) = x(i) - x(N/2+i)$

$$\text{for } i=0,1...N/2-1 \quad (6)$$

We have

$$X(2k+1) = Y(k) + Z(k)$$

and $$X(N-2k-1) = Y(k) - Z(k) \qquad \text{for } k=0,1...N/4-1 \quad (7)$$

To realize $\{Y(k):k=0,1..N/4-1\}$, we can define the following

$$Y(k) + Y(k-1) =$$

$$G(k) = \sum_{i=0}^{N/4-1} g(i) \cos\left(\frac{4\pi ik}{N}\right) \qquad \text{for } k=0,1...N/4-1 \quad (8)$$

$$\text{where } g(0) = 2\,y(0)$$

$$g(i) = 2\left\{y(i) - y(N/2-i)\right\} \cos(2\pi i/N) \qquad \text{for } i=1,2...N/4-1 \quad (9)$$

and $Y(-1)$ is defined as $Y(0)$.

Similarly, by defining $Z(-1) = -Z(0)$, we can define the following to take care of the sequence $\{Z(k):k=0,1...N/4-1\}$,

$$Z(k) - Z(k-1) =$$

$$F(k) = \sum_{i=0}^{N/4-1} f(i) \cos\left(\frac{4\pi ik}{N}\right) + (-1)^k 2y(N/4) \qquad \text{for } k=0,1...N/4-1 \quad (10)$$

$$\text{where } f(0) = 0$$

$$f(i) = 2\left\{y(i) + y(N/2-i)\right\} \sin(2\pi i/N) \qquad \text{for } i=1,2...N/4-1 \quad (11)$$

Note that $G(k)$ and $F(k)$, defined by eqns. 8 and 10, are respectively the symmetric cosine structure(SCS) and the alternative form of the symmetric cosine structure(SCS_0^*) defined by Chan and Siu[9]. Hence, an N-length DHT can be decomposed into an N/2-length DHT, an N/4-length SCS and an N/4-length SCS_0^* with a cost of N/2-2 multiplications and 5N/2-4 additions. In particular (See Appendix), we have

$$M(N\text{-SCS}) = M(N\text{-SCS}_0^*) = M(N/2\text{-SCS}) + M(N/2\text{-SCS}^*) + N/2 - 1$$

$$A(N\text{-SCS}) = A(N\text{-SCS}_0^*) = A(N/2\text{-SCS}) + A(N/2\text{-SCS}^*) + 3N/2 - 3 \qquad \text{for } N > 4$$

$$M(4\text{-SCS}) = M(4\text{-SCS}_0^*) = 1$$

$$A(4\text{-SCS}) = A(4\text{-SCS}_0^*) = 7 \qquad (12)$$

where $M(N\text{-}Y)$ and $A(N\text{-}Y)$ are the numbers of multiplications and additions respectively for a length-N structure denoted by Y and SCS^* is another alternative form of the symmetric cosine structure, which is defined the same as SCS_0^* in eqn 10 except that the input data $f(0)$ is not necessary to be zero, whose mathematical complexity is given by

$$M(N\text{-SCS}^*) = M(N/2\text{-SCS}) + M(N/2\text{-SCS}^*) + N/2 - 1$$

$$A(N\text{-SCS}^*) = A(N/2\text{-SCS}) + A(N/2\text{-SCS}^*) + 3N/2 - 1 \qquad \text{for } N > 4$$

$$M(4\text{-SCS}^*) = 1$$

$$A(4\text{-SCS}^*) = 9 \qquad (13)$$

On the other hand, it is well-known that a 4-length DHT can be realized with 8 additions only and an 8-length DHT can be realized with 2 multiplications and 22 additions[4]. Therefore, the mathematical complexity of the new algorithm is given by the following equations:

$$M(4\text{-DHT}) = 0$$

$$A(4\text{-DHT}) = 8$$

$$M(8\text{-DHT}) = 2$$

$$A(8\text{-DHT}) = 22$$

$$M(N\text{-DHT}) = M(N/2\text{-DHT}) + M(N/4\text{-SCS}) + M(N/4\text{-SCS}_0^*) + N/2 - 2$$

$$A(N\text{-DHT}) = A(N/2\text{-DHT}) + A(N/4\text{-SCS}) + A(N/4\text{-SCS}_0^*) + 5N/2 - 4 \qquad \text{where } N=2^m,\ m>3 \quad (14)$$

Table 1 shows the computational complexity of the realization of DHT of different lengths using the present approach and other efficient techniques[4-6,9]. The computational complexity of other techniques is quoted directly from table 1 of reference[4] and table 1 of reference[6] respectively. The split-radix algorithm[4] always gives the minimum number of additions, whereas Malvar[5,6]'s and Chan and Siu[9]'s algorithms always give the minimum number of multiplications for all lengths shown. The present approach requires the same minimum number of multiplications for all lengths, however, it requires a smaller number of additions than that required by those two approaches. Figures 1 and 2 also illustrate clearly the comparison of the present approach with other major approaches for the realization of the Discrete Hartley Transform.

Table 1. Computational complexity of the realization of DHT for different lengths using the present approach and other efficient techniques[4,6,9].

	Present Approach		Chan & Siu[9]		radix-2[4]		radix-4 [4]		split-radix[4]		Malvar[6]	
N	M	A	M	A	M	A	M	A	M	A	M	A
8	2	22	2	29	4	26			2	22	2	26
16	10	72	10	83	20	74	14	70	12	64	10	74
32	34	198	34	217	68	194			42	166	34	206
64	98	500	98	535	196	482	142	450	124	416	98	522
128	258	1202	258	1269	516	1154			330	998	258	1278
256	642	2800	642	2931	1284	2690	942	2498	828	2336	642	3018
512	1538	6382	1538	6641	3076	6146			1994	5350	1538	6974
1024	3586	14316	3586	14831	7172	13826	5294	12802	4668	12064	3586	15818
2048	8194	31722	8194	32749	16388	30722			10698	26854	8194	35390

The basic idea of the proposed algorithm implies a recursive decomposition technique. By using the recursion, a 2^m-length DHT ($m>3$) is eventually decomposed into an 8-length DHT and a number of 4-length SCS, SCS^* and SCS_0^*. Hence, the realization modules of 4-length SCS etc. are actually basic elements of the implementation of the algorithm. Consider that the 4-length SCS and the 4-length SCS_0^* are only variants of the

4-length SCS*, all of them can be realized with a typical 4-length SCS* module without extra cost. This property is practical and desirable especially when the algorithm is used for the hardware realization of the DHT. In such case, only simple 4-length SCS* hardware modules are required, which makes the proposed algorithm very suitable for VLSI implementation.

Some multiplications are required for the generation of sine or cosine coefficients. In general, they can be precomputed and stored in a memory table as they are known values. The table lookup technique can then be applied to save computation effort during the realization. Note that this algorithm requires only a small table for storing values of $\{\sin(2\pi i/N):i=1,2...N/4-1\}$ and $\{\cos(2\pi i/N):i=1,2...N/4-1\}$, the size of which is practical for most applications.

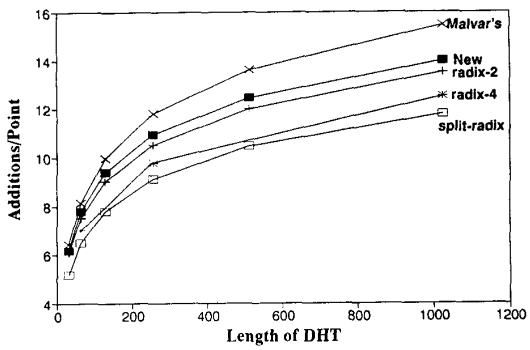

Fig.1 Comparison of additions per point among different approaches

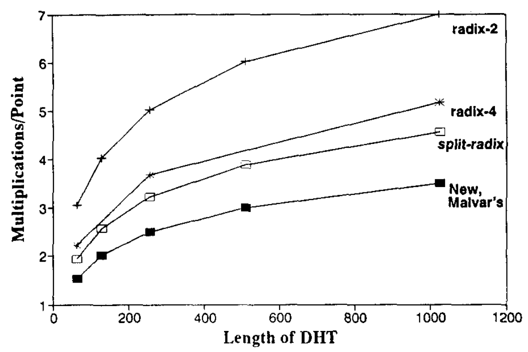

Fig.2 Comparison of multiplications per point among different approaches

EXAMPLE

Let us use a 16-length DHT on $\{x(i):i=0,1...15\}$ to clarify our approach:

Firstly, we compute the even sequence of the output, $\{X(2k):k=0,1..7\}$, via an 8-length DHT on the sequence $\{x(i)+x(8+i):i=0,1..7\}$ as shown in eqn 2.

To compute the odd sequence, we have to compute sequences $\{Y(k):k=0,1..3\}$ and $\{Z(k):k=0,1..3\}$. By using eqn 9, we have

$$\begin{bmatrix} g(0) \\ g(1) \\ g(2) \\ g(3) \end{bmatrix} = \begin{bmatrix} 2\,y(0) \\ 2\,[y(1)-y(7)]\cos(\pi/8) \\ 2\,[y(2)-y(6)]\cos(2\pi/8) \\ 2\,[y(3)-y(5)]\cos(3\pi/8) \end{bmatrix}$$

where $y(i) = x(i) - x(8+i)$ for $i=0,1..7$

Then, after we compute the 4-length SCS on $\{g(i):i=0,1..3\}$ with eqn 8 to obtain $\{G(k):k=0,1..3\}$, we have

$$\begin{bmatrix} Y(0) \\ Y(1) \\ Y(2) \\ Y(3) \end{bmatrix} = \begin{bmatrix} G(0)/2 \\ G(1)-Y(0) \\ G(2)-Y(1) \\ G(3)-Y(2) \end{bmatrix}$$

Similarly, by using eqn 11, we have

$$\begin{bmatrix} f(0) \\ f(1) \\ f(2) \\ f(3) \end{bmatrix} = \begin{bmatrix} 0 \\ 2\,[y(1)+y(7)]\sin(\pi/8) \\ 2\,[y(2)+y(6)]\sin(2\pi/8) \\ 2\,[y(3)+y(5)]\sin(3\pi/8) \end{bmatrix}$$

and, after computing the 4-length SCS_0^* on $\{f(i):i=0,1..3\}$ with eqn 10 to obtain $\{F(k):k=0,1..3\}$, we have

$$\begin{bmatrix} Z(0) \\ Z(1) \\ Z(2) \\ Z(3) \end{bmatrix} = \begin{bmatrix} F(0)/2 \\ F(1)+Z(0) \\ F(2)+Z(1) \\ F(3)+Z(2) \end{bmatrix}$$

Then, we can determine the odd sequence of the final output by using eqn 7:

$$\begin{cases} X(1) = Y(0)+Z(0) \\ X(3) = Y(1)+Z(1) \\ X(5) = Y(2)+Z(2) \\ X(7) = Y(3)+Z(3) \end{cases} \text{and} \quad \begin{cases} X(15) = Y(0)-Z(0) \\ X(13) = Y(1)-Z(1) \\ X(11) = Y(2)-Z(2) \\ X(9) = Y(3)-Z(3) \end{cases}$$

Totally, 10 multiplications and 72 additions are required.

CONCLUSIONS

A new algorithm is proposed in this paper to realize a 2^m-length Discrete Hartley Transform. This algorithm requires the same minimum number of multiplications as other algorithms reported in the literature[2-9] and requires a smaller number of additions compared with other algorithms[5,6,9]. It is significant to point out that the present algorithm is most suitable for VLSI realization. The resultant structure of the approach is stable and regular.

REFERENCES

[1] R.N.Bracewell, "The Discrete Hartley Transform," J. Opt. Soc. Amer., Vol.73, pp.1832-1835, 1983

[2] R.N. Bracewell, "The fast Hartley transform," IEEE Proc., Vol.72, pp.1010-1018, 1984

[3] R.N.Bracewell, *The Hartley Transforms*, England: Oxford University Press, 1985.

[4] H.V.Sorensen, D.L.Jones, C.S.Burrus, and M.T.Heideman, "On computing the Discrete Hartley Transform", IEEE Trans. on ASSP., Vol.ASSP-33, No.4, pp.1231-1238, Oct 1985.

[5] Malvar, "Fast Computation of the Discrete Cosine Transform and the Discrete Hartley Transform", IEEE Trans. on ASSP., Vol.ASSP-35, No.10, PP.1484-1485, Oct, 1987.

[6] Malvar, "Corrections to 'Fast Computation of the Discrete Cosine Transform and the Discrete Hartley Transform'", IEEE Trans.on ASSP., Vol.36, No.4,pp.610, Apr 1988.

[7] H.S.Hou, "The fast Hartley transform algorithm," IEEE Trans. Comput., Vol.C-36, pp.147-156, 1987.

[8] S.C.Pei and J.L.Wu, "Split -radix fast Hartley transform," Electron. Lett., Vol.22, pp.26-27, 1986.

[9]Y.H.Chan, and W.C.Siu, "Modular Structures for Fast Discrete Hartley Transform and Real-valued Discrete Fourier Transform," Proceedings, International Symposium on Signal Processing and Its Applications (ISSPA'90), 27-31 August, 1990, Gold Coast, Australia, pp.221-224.

[10]P.Duhamel and H.Hollmann, "'Split radix' FFT algorithm," Electronics Letters, Vol.20, No.1, pp.14-16, 5th Jan 1984.

APPENDIX

An N-length SCS* on sequence $\{x'(i):i=0,1...N-1\}$ is defined as follows:

$$T(k) = \sum_{i=0}^{N-1} x'(i) \cos\left(\frac{\pi ik}{N}\right) + (-1)^k x_0$$

$k=0,1..N-1$ (A1)

where x_0 is a real number. This structure forms the basic module of the proposed algorithm. Actually, the N-length SCS and SCS_0^* are only special cases of this basic structure. In particular, if x_0 is zero, eqn A1 becomes the definition of an N-length SCS and, when $x'(0)$ is zero, it becomes the definition of an N-length SCS_0^*.

If N is even, we can split eqn A1 into two sequences. For the even sequence, we have

$$T(2k) = \sum_{i=0}^{N/2-1} g'(i) \cos\left(\frac{2\pi ik}{N}\right) + (-1)^k x'(N/2)$$

for $k=0,1...N/2-1$ (A2)

where $g'(0) = x'(0) + x_0$

$$g'(i) = x'(i) + x'(N-i)$$

for $i=1,2..N/2-1$ (A3)

and, if we define $T(-1)$ as $T(1)$, we have

$$T(2k+1)+T(2k-1) = F(k) = \sum_{i=0}^{N/2-1} e'(i) \cos\left(\frac{2\pi ik}{N}\right)$$

for $k=0,1...N/2-1$ (A4)

where $e'(0) = 2\{x'(0) - x_0\}$

$$e'(i) = 2\left\{x'(i) - x'(N-i)\right\}\cos(\pi i/N)$$

for $i=1,2..N/2-1$ (A5)

Hence, an N-length SCS* can be realized through an N/2-length SCS and an N/2-length SCS* with a cost of N/2-1 multiplications and 3N/2-1 additions. Eqn. 13 gives the mathematical complexity of this basic structure.

In particular, if N = 4, it can be realized with 9 additions and 1 multiplication as follows:

$temp0 = \{x'(1)-x'(3)\}\cos(\pi/4)$

$temp1 = x'(0) + x_0$

$temp2 = x'(0) - x_0$

$T(0) = temp1 + x'(1) + x'(2) + x'(3)$

$T(3) = temp2 - temp0$

$T(1) = temp2 + temp0$

$T(2) = temp1 - x'(2)$

The substitution of $x'(0)=0$ and $x_0=0$ into eqns A1 to A5 gives the mathematical complexity of an N-length SCS_0^* and SCS respectively. Eqn. 12 describes the mathematical complexity of both structures precisely.

Acknowledgment

The authors wish to acknowledge the support of the Research Sub-Committee of the Hong Kong Polytechnic and the Video Technology Computers Ltd.