FAST COMPUTATION OF MULTIDIMENSIONAL DISCRETE HARTLEY TRANSFORM

P. K. Meher, J. K. Satapathy and G. Panda

Indexing terms: Signal processing, Transforms, Algorithms

Efficient algorithms for the fast computation of 2-D and 3-D discrete Hartley transforms have been proposed. It is shown that the proposed algorithms offer a significant saving in computation over the existing methods for various array sizes.

Introduction: Fast computation of the multidimensional discrete Hartley transform (DHT) is a subject of current interest [1–3] due to its application in image processing and spectral analysis. Bracewell *et al.* [1, 2] have proposed a computation of the multidimensional DHT by adding a certain number of temporary arrays; the temporary arrays are computed by a one-dimensional (1-D) fast Hartley transform algorithm. Also, Boussakta and Holt [3] have reported a fast multidimensional DHT algorithm using Fermat number transforms (FNTs).

We propose relatively simple algorithms for 2-D as well as 3-D DHTs which may further be extended to higher dimensions depending on the requirement. According to the proposed scheme the multidimensional DHT may be computed using any of the fast DFT algorithms and fast DHT algorithms in one dimension. We have calculated the operational requirement of the proposed algorithms using the Winograd DFT algorithm [4]/prime factor FFT of real-valued series [5] and corresponding Hartley transform algorithms. It is found that the proposed method offers significant computational saving over the others.

Fast algorithm for 2-D DHT: The 2-D DHT of an $M \times N$ array $[x(m, n)]$ may be defined as

$$X(k, l) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x(m, n) \times \left[\cos 2\pi\left(\frac{km}{M} + \frac{ln}{N}\right) + \sin 2\pi\left(\frac{km}{M} + \frac{ln}{N}\right)\right] \quad (1)$$

By splitting the arguments of sine and cosine functions of eqn. 1 we obtain

$$X(k, l) = \sum_{n=0}^{N-1} u(k, n)\left(\cos\frac{2\pi ln}{N} + \sin\frac{2\pi ln}{N}\right) + \sum_{n=0}^{N-1} v(k, n)\left(\cos\frac{2\pi ln}{N} - \sin\frac{2\pi ln}{N}\right) \quad (2)$$

where

$$u(k, n) = \sum_{m=0}^{M-1} x(m, n)\cos\frac{2\pi km}{M} \quad (3)$$

and

$$v(k, n) = \sum_{m=0}^{M-1} x(m, n)\sin\frac{2\pi km}{M} \quad (4)$$

It may be noted here that $u(k, n)$ and $v(k, n)$ represent the real part and negative imaginary part of an M-point DFT of the nth column of $[x(m, n)]$.

Substituting $n = (N - n)$ into the second sum of eqn. 2, we obtain

$$X(k, l) = \sum_{n=0}^{N-1} w(k, n)\left[\cos\frac{2\pi ln}{N} + \sin\frac{2\pi ln}{N}\right] \quad (5)$$

where

$$w(k, n) = u(k, n) + v(k, N - n) \quad \text{and} \quad v(k, N) = v(k, 0) \quad (6)$$

Eqns. 3–6 indicate that a 2-D DHT of an array of size $M \times n$ may be computed in the following sequence:

(i) the M-point DFT of each column of $[x(m, n)]$ is computed

(ii) the real part of the DFT of the nth column is added to the negative imaginary part of the DFT of the $(N - n)$th column for $1 \le n \le N - 1$; the real part of the DFT of the zeroth column, however, is added to the negative imaginary part of the DFT of the same column: the results should be stored in the corresponding positions of $[x(m, n)]$ under its new variable name $[w(k, n)]$

(iii) the n-point DHT of each row of $[w(k, n)]$ is computed to obtain the desired 2-D DHT.

Fast algorithm for 3-D DHT: The 3-D DHT of an array $[x(l, m, n)]$ of size $L \times M \times N$ may be defined as

$$X(i, j, k) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(l, m, n) \times \left[\cos 2\pi\left(\frac{il}{L} + \frac{jm}{M} + \frac{kn}{N}\right) + \sin 2\pi\left(\frac{il}{L} + \frac{jm}{M} + \frac{kn}{N}\right)\right] \quad (7)$$

Splitting the arguments of the sine and the cosine functions of eqn. 7 by a procedure the same as that of the 2-D case, we have

$$X(i, j, k) = \sum_{n=0}^{N-1} w_2(i, j, n)\left[\cos\frac{2\pi kn}{N} + \sin\frac{2\pi kn}{N}\right] \quad (8)$$

where

$$w_2(i, j, n) = u_2(i, j, n) + v_2(i, j, N - n) \quad (9)$$

$$u_2(i, j, n) = \sum_{m=0}^{M-1} w_1(i, m, n)\cos\frac{2\pi jm}{M} \quad (10)$$

$$v_2(i, j, n) = \sum_{m=0}^{M-1} w_1(i, m, n)\sin\frac{2\pi jm}{M} \quad (11)$$

$$w_1(i, m, n) = u_1(i, m, n) + v_1(i, M - m, N - n) \quad (12)$$

$$u_1(i, m, n) = \sum_{l=0}^{L-1} x(l, m, n)\cos\frac{2\pi il}{L} \quad (13)$$

$$v_1(i, m, n) = \sum_{l=0}^{L-1} x(l, m, n)\sin\frac{2\pi il}{L} \quad (14)$$

by assuming

$$v_1(i, M, n) = v_1(i, 0, n) \quad (15a)$$

$$v_1(i, m, N) = v_1(i, m, 0) \quad (15b)$$

$$v_1(i, M, N) = v_1(i, 0, 0) \quad (15c)$$

and

$$v_2(i, j, N) = v_2(i, j, 0) \quad (15d)$$

for $0 \le i \le L - 1$, $0 \le j \le M - 1$ and $0 \le k \le N - 1$.

A 3-D DHT may be computed according to eqns. 8–15 by a procedure similar to that of a 2-D DHT.

Computational complexity: To obtain the 2-D DHT of an array of size $M \times N$, N M-point DFTs and M N-point DHTs have to be computed. Besides, to add the real parts with the appropriate negative imaginary parts of the DFT in the first stage of computation $(M - 1)N$ or $(M - 2)N$, additions will be required for M odd or even, respectively. For the 3-D DHT of an array of size $L \times M \times N$, MN L-point DFTs followed by LN M-points DFTs and LM N-point DHTs have to be computed. The number of interstage additions for 3-D amounts to $[(L-p)MN+(M-q)LN]$; where, $p=1$ if L is odd and $p=2$ if L is even. Similarly, $q=1$ if M is odd and $q=2$ if M is even.

The operational requirements of the proposed 2-D and 3-D DHT algorithms using the Winograd DFT algorithm [4]/prime factor FFT of real-valued series [5] and corresponding DHT algorithms are calculated and compared in Table 1 with those of the existing algorithms [1–3]. It is observed that the proposed algorithms offer significant saving of multiplications as well as additions over the method of Bracewell *et al.* for various array sizes. For every output point, the proposed method, on average, offers a saving of more than 12·5 additions at the cost of nearly 2·4 extra multiplications over the FNT method. Furthermore, the time required to perform multiplications is greatly reduced with the improvement in hardware technology of present day computers, such that the computation time of a multiplication has become comparable to that of an addition. The proposed algorithm, therefore, will be more efficient compared with the FNT method as well.

Table 1 COMPARISON OF OPERATION COUNTS PER POINT FOR COMPUTATION OF MULTIDIMENSIONAL DHT BY DIFFERENT ALGORITHMS

Array size	Proposed method			Method of Bracewell *et al.*			FNT method		
	Mult.	Add.	Total	Mult.	Add.	Total	Mult.	SA*	Total
16 × 16	1·25	8·75	10·0	4·25	14·25	18·5			
17 × 17							1·11	17·7	18·81
252 × 252	4·51	23·62	28·13						
256 × 256				12·02	30·02	42·04			
257 × 257							1·01	34·75	35·76
16 × 16 × 16	1·87	13·37	15·25	6·38	20·38	26·76			
17 × 17 × 17							1·18	29·18	30·36
16 × 16 × 252	3·50	21·06	24·56						
16 × 16 × 256				10·26	28·26	38·52			
17 × 17 × 257							1·18	36·82	38·00
16 × 252 × 252	5·13	28·24	33·38						
16 × 256 × 256				14·14	36·14	50·28			
17 × 257 × 257							1·07	44·55	45·62
252 × 252 × 252	6·76	35·43	42·19						
256 × 256 × 256				18·02	46·00	64·02			
257 × 257 × 257							1·01	51·95	52·96

* SA denotes shift-adds operation per output point
Mult: multiplications, Add: additions

Conclusion: An efficient scheme for the computation of 2-D and 3-D DHTs is presented. It is shown that the proposed method offers significant computational saving over the other reported methods for various array sizes. The proposed algorithms, although using a DFT algorithm, do not involve complex arithmetic because the real and imaginary parts of the 1-D DFT of real-valued data are used separately instead of complete DFT components.

9th April 1992

P. K. Meher (*Dept. of Physics, Govt. College, Rourkela 769004, India*)

J. K. Satapathy (*Dept. of Electrical Engineering, RE College, Rourkela 769008, India*)

G. Panda (*Dept. of AEIE, RE College, Rourkela 769008, India*)

References

1 BRACEWELL, R. N., BUNEMAN, O., HAO, H., and VILLASENOR, J.: 'Fast two-dimensional Hartley transform', *Proc. IEEE*, 1986, **74**, (9), pp. 1282–1283

2 HAO, H., and BRACEWELL, R. N.: 'A three-dimensional DFT algorithm using fast Hartley transform', *Proc. IEEE*, 1987, **75**, (2), pp. 264–266

3 BOUSSAKTA, S., and HOLT, A. G. J.: 'Fast multidimensional discrete Hartley transform using Fermat number transform', *IEE Proc. G*, 1988, **135**, (6), pp. 253–257

4 SILVERMAN, H. F.: 'An introduction to programming the Winograd Fourier transform algorithm (WFTA)', *IEEE Trans.*, 1977, **ASSP-25**, pp. 152–165

5 HEIDEMAN, M. T., BURRUS, C. S., and JOHNSON, H. W.: 'Prime factor FFT algorithms for real-valued series'. Proc. ICASSP, 1984, pp. 28.A.7.1–28.A.7.4

1·55 μm MULTIQUANTUM-WELL LASERS WITH RECORD PERFORMANCE OBTAINED BY ATMOSPHERIC PRESSURE MOVPE USING ORGANOMETALLIC PHOSPHORUS PRECURSOR

A. Ougazzaden, A. Mircea, R. Mellet, G. Primot and C. Kazmierski

Indexing terms: Lasers, Semiconductor lasers

The recently available precursor biphosphinoethane (BPE) was used, alongside with phosphine and with tertbutylphosphine (TBP), to grow advanced multiquantum-well (MQW) laser wafers with five quaternary, compressive strained wells. The lowest threshold current densities and the lowest optical losses were obtained with BPE. In particular, the lowest threshold current density, 328 A/cm^2, is a record among published values for lasers with five wells. In this comparison, the wafer grown with phosphine came a close second and that grown with TBP was third.

Introduction: Among several possible phosphorus organometallic compounds which could replace the hazardous phosphine in MOVPE, up to now, tertbutylphosphine (TBP) has received the most attention. We studied the use of TBP in growing laser devices [1]. Recently, the compound BPE, the synthesis and properties of which were already reported 31 years ago by Meier [2], was proposed as a commercial product. The advantage of BPE with respect to other proposed compounds is the high share of phosphorus in the molecular weight and the high phosphorus to carbon ratio