



ELSEVIER

Signal Processing 80 (2000) 2255–2260

**SIGNAL  
PROCESSING**

www.elsevier.nl/locate/sigpro

# Polynomial transform based algorithms for computing two-dimensional generalized DFT, generalized DHT, and skew circular convolution

Yuh-Ming Huang<sup>a</sup>, Ja-Ling Wu<sup>b,\*</sup>

<sup>a</sup>*Department of Information Engineering, National Chi-Nan University, University Road, Puli, Nantou 545, Taiwan*

<sup>b</sup>*Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan*

Received 22 December 1997; received in revised form 21 December 1999

---

## Abstract

Fast computation of two-dimensional (2-D) generalized discrete Fourier transforms (GDFTs) and generalized discrete Hartley transforms (GDHTs) are discussed in terms of a new method which is derived based on index permutation, linear congruences, and polynomial transforms. Further, the 2-D skew circular convolution computed by the proposed skew polynomial transform is also included. © 2000 Elsevier Science B.V. All rights reserved.

## Zusammenfassung

Die schnelle Berechnung zweidimensionaler (2-D) verallgemeinerter diskreter Fourier-Transformationen (GDFT's) und verallgemeinerter diskreter Hartley-Transformationen (GDHT's) wird anhand einer neuen Methode diskutiert, die auf Basis von Indexpermutationen, linearen Kongruenzen und Polynomialtransformationen abgeleitet wird. Die 2-D schiefe zyklische Faltung durch die vorgeschlagene schiefe Polynomialtransformation wird ebenfalls behandelt. © 2000 Elsevier Science B.V. All rights reserved.

## Résumé

Nous discutons le calcul rapide des transformations de Fourier discrètes généralisées bi-dimensionnelles et des transformations de Hartely discrètes généralisées, en termes d'une nouvelle méthode qui est dérivée d'une permutation d'indices, d'une congruence linéaire et de transformations polynomiales. De plus, Nous incluons aussi la convolution circulaire asymétrique calculée par la transformation polynomiale asymétrique. © 2000 Elsevier Science B.V. All rights reserved.

*Keywords:* Polynomial transform; Generalized discrete Fourier transform; Generalized discrete Hartley transform; Skew circular convolution; Index permutation; Linear congruence

---

\* Corresponding author.

E-mail addresses: ymhuang@csie.ncnu.edu.tw (Y.M. Huang), wjl@cmlab.csie.ntu.edu.tw (J.-L. Wu).

## 1. Introduction

The generalized discrete Fourier transforms (GDFT) and generalized discrete Hartley transforms (GDHT) [3] have many applications in digital signal processing such as filter banks, fast computation of convolution and signal representation [1]. Also, the skew circular convolution (SCC) is widely used for computing the discrete cosine transforms [5].

Polynomial transforms, defined in rings of polynomials, have been shown to provide efficient algorithms for the computation of two-dimensional (2-D) convolutions and to map multi-dimensional discrete Fourier transforms (DFTs) into one-dimensional (1-D) DFTs [7–9].

Gertner [2] has presented another scheme that can also be used to convert the 2-D DFTs into a number of 1-D DFTs. The main result of this approach is that the number of distinct  $N$ -point DFTs needed to calculate an  $(N \times N)$ -point 2-D DFT is equal to the number of linear congruences spanning the  $N \times N$  grids. Based on this idea and a local ring structure [6], the 2-D discrete Hartley transform (DHT) of length  $2^n \times 2^n$  can be transformed into  $(3/2)2^n$  distinct 1-D DHTs.

Recently, Hu and Leu [4] have shown that the fast calculation of 2-D GDHTs can be achieved by solving index mapping equations. This approach maps an  $(N \times N)$ -point GDHT into  $(N \times N)$ -point  $H_3$  transforms [4]. Thus, by recursively decomposing the  $(N \times N)$ -point DHT into a number of distinct 2-D GDHTs, an  $(N \times N)$ -point DHT requires  $(3N/2^i)$  1-D  $N/2^i$ -point  $H_3$  transforms,  $i = 1, \dots, \log_2 N - 1$ . This approach needs the same number of multiplications as that of [6] but needs fewer additions.

In this paper, a polynomial transform based algorithm for computing the GDHT or GDFT is proposed. Although the resultant complexity is the same as that of [4], however, the proposed approach exhibits not only a new succinct and efficient algorithm but also points out how to change the 2-D transform into a number of 1-D transforms by the index permutation and linear congruence methods. In other words, it may provide one new direction to derive fast algorithms for computing the other 2-D discrete transforms.

Finally, based on the derived forward and inverse skew polynomial transforms, a fast algorithm for computing 2-D skew circular convolution will be presented.

## 2. Fast algorithms for the GDFTs, GDHTs, and 2-D skew circular convolution

For a given input data sequence  $f_{m,n}$ ,  $0 \leq m \leq N - 1$ ,  $0 \leq n \leq N - 1$ , the 2-D GDFT and GDHT of various half-sample generalized DFTs and DHTs are given by [4]

$$F_{k,h} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_{m,n} \ker \left[ \frac{2\pi((k+k_0)m + (h+h_0)n)}{N} \right],$$

$$0 \leq k \leq N - 1, 0 \leq h \leq N - 1, \quad (1)$$

where  $N$  is a power of two,  $k_0, h_0 = 0$  or  $0.5$  but not all equal to zero, and  $\ker(x)$  is defined to be  $e^{-ix}$  and  $\cos x + \sin x$ , respectively.

Case A:  $k_0 = 0$  and  $h_0 = 0.5$ . In this case, (1) can be rewritten as

$$F_{k,h} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_{m,n} \ker \left[ \frac{2\pi(km + (h + 0.5)n)}{N} \right],$$

$$0 \leq k \leq N - 1, 0 \leq h \leq N - 1. \quad (2)$$

Since  $2h + 1$  and  $N$  are coprime to each other, i.e.  $\gcd(2h + 1, N) = 1$ , the permutation  $(2h + 1)k$  modulo  $N$  maps all values of  $k$ . Let  $q_{hk}$  be the quotient of  $(2h + 1)k$  divided by  $N$ . Thus, the kernel of the 2-D transform in (2) can be represented as 1-D GDFT (or GDHT) by replacing  $k$  with  $(2h + 1)k - Nq_{hk}$ , that is,

$$F_{\langle (2h+1)k \rangle_N, h}$$

$$= \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_{m,n} \ker \left[ \frac{\pi(2h + 1)(2km + n)}{N} \right],$$

$$0 \leq k \leq N - 1, 0 \leq h \leq N - 1, \quad (3)$$

where  $\langle x \rangle_N$  denotes  $x$  modulo  $N$ .

For each given  $k$  and  $q, 0 \leq q \leq N - 1$ , it follows that the linear congruence equation  $2km + n = q$  modulo  $N$  has  $N$  solutions  $(m, n) = (s, \langle q - 2ks \rangle_N), 0 \leq s \leq N - 1$ , in the quadratic  $N \times N$  grid domain. Furthermore, these  $N$  linear congruences  $2km + n = q$  modulo  $N, 0 \leq q \leq N - 1$ , do span the quadratic  $N \times N$  grid. Let  $(2km + n - q)/N = t$  and since  $\ker[\pi(2h + 1)(t + q/N)] = (-1)^t \ker[\pi(2h + 1)q/N]$ , we get

$$F_{\langle(2h+1)k\rangle_N, h} = \sum_{q=0}^{N-1} f_k(q) \ker \left[ \frac{\pi(2h+1)q}{N} \right], \quad (4a)$$

where

$$f_k(q) = \sum_{s=0}^{N-1} f_{s, \langle q - 2ks \rangle_N} (-1)^{[2ks + \langle q - 2ks \rangle_N - q/N]}. \quad (4b)$$

Next, let us define

$$F_k(Z) = \sum_{q=0}^{N-1} f_k(q) Z^q. \quad (5)$$

It is easy to verify that the calculation of  $f_k(q)$  is equivalent to computing one forward polynomial transform which is defined as [7]

$$F_k(Z) = \sum_{m=0}^{N-1} f_m(Z) Z^{2mk} \bmod Z^N + 1, \quad (6a)$$

where

$$f_m(Z) = \sum_{n=0}^{N-1} f_{m,n} Z^n. \quad (6b)$$

Therefore, the computation of 2-D GDFT or GDHT (defined in Eq. (2)) can be accomplished through one polynomial transform followed by  $N$  1-D GDFTs or GDHTs of length  $N$ , as shown in Eq. (4).

Case B:  $k_0 = 0.5$  and  $l_0 = 0.5$ . In this case, Eq. (1) can be rewritten as

$$F_{k,h} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_{m,n} \ker \left[ \frac{2\pi((k+0.5)m + (h+0.5)n)}{N} \right],$$

$$0 \leq k \leq N - 1, 0 \leq h \leq N - 1. \quad (7)$$

Since  $\langle(2h + 1)k + h\rangle_N$  is just a permutation of index  $k$ , the kernel of 2-D transform in (7) can also be represented by 1-D GDFT (or GDHT) as follows:

$$F_{\langle(2h+1)k+h\rangle_N, h} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_{m,n} \ker \left[ \frac{\pi(2h+1)((2k+1)m+n)}{N} \right],$$

$$0 \leq k \leq N - 1, 0 \leq h \leq N - 1. \quad (8)$$

For each given  $k$  and  $q, 0 \leq q \leq N - 1$ , the linear congruence equation  $(2k + 1)m + n = q$  modulo  $N$  has  $N$  solutions  $(m, n) = (s, \langle q - (2k + 1)s \rangle_N), 0 \leq s \leq N - 1$ . Furthermore, these  $N$  linear congruences  $(2k + 1)m + n = q$  modulo  $N, 0 \leq q \leq N - 1$ , also do span the quadratic  $N \times N$  grid. Hence, by the same technique as that of Case A, Eq. (8) can be rewritten as

$$F_{\langle(2h+1)k+h\rangle_N, h} = \sum_{q=0}^{N-1} f_k(q) \ker \left[ \frac{\pi(2h+1)q}{N} \right], \quad (9a)$$

where

$$f_k(q) = \sum_{s=0}^{N-1} f_{s, \langle q - (2k+1)s \rangle_N} (-1)^{[(2k+1)s + \langle q - (2k+1)s \rangle_N - q/N]}. \quad (9b)$$

Now, if  $F_k(Z)$  and  $f_m(Z)$  were defined as shown in Eqs. (5) and (6b), respectively, then

$$F_k(Z) = \sum_{m=0}^{N-1} f_m(Z) Z^{(2k+1)m} \bmod Z^N + 1. \quad (10)$$

Therefore, the fast computation of another kind of 2-D GDFT or GDHT, defined in Eq. (7), can also be achieved by computing one skew polynomial transform and  $N$  1-D GDFTs or GDHTs of length  $N$ , as shown in Eq. (9).

By decomposing the 2-D DHT of size  $N \times N$  into one  $(N/2) \times (N/2)$ -point DHT and three distinct  $(N/2) \times (N/2)$ -point GDHTs [4], one fast algorithm for computing 2-D DHTs can be obtained by combining the polynomial transforms and 1-D GDHTs as shown in Fig. 1.

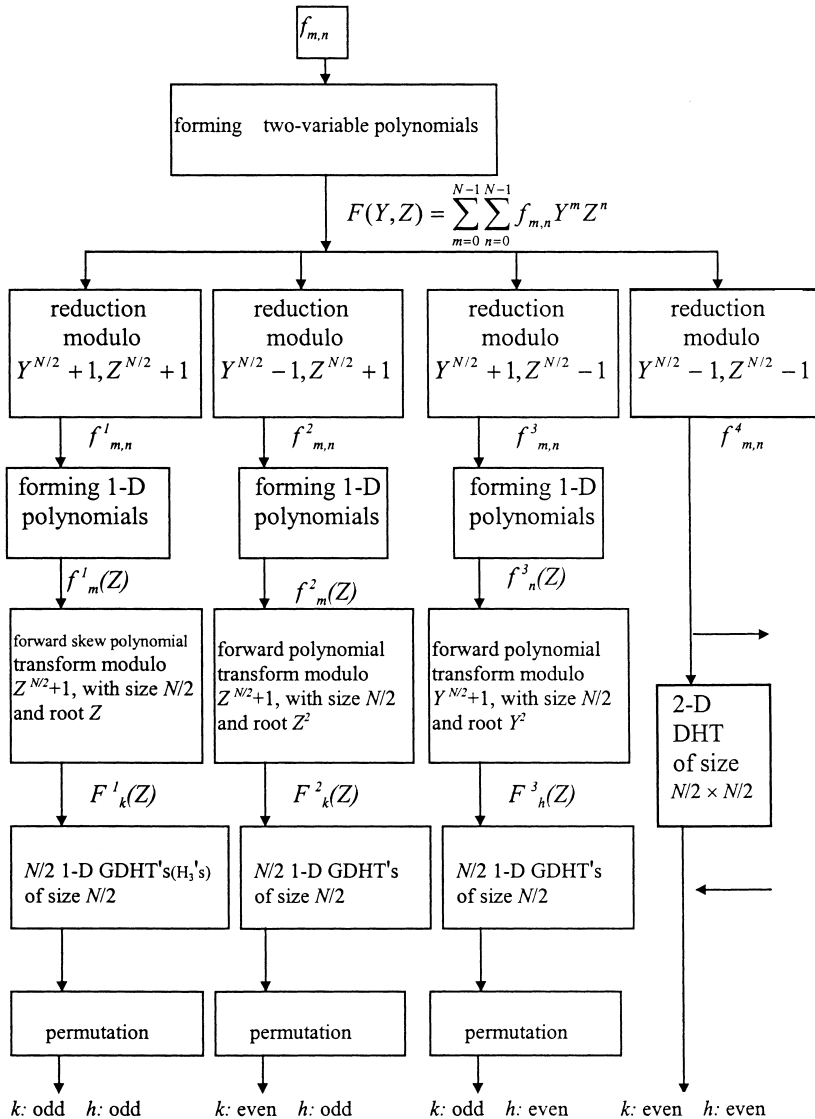


Fig. 1. Computation of a 2-D DHT of size  $N \times N$  by polynomial transforms, where  $N = 2^p$ .

Next, consider the 2-D skew circular convolution (2-D SCC) of size  $N \times N$ ,

$$y(p, q) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \mu(p-m)\mu(q-n)x_{m,n}h_{\langle p-m \rangle_N, \langle q-n \rangle_N},$$

$$0 \leq p \leq N-1, \quad 0 \leq q \leq N-1 \tag{11}$$

where

$$\mu(i) = \begin{cases} 1, & i \geq 0. \\ -1, & i < 0. \end{cases}$$

By rewriting Eq. (11) in the form of 1-D skew polynomial convolution, we obtain

$$y_p(Z)$$

$$= \sum_{m=0}^{N-1} \mu(p-m)x_m(Z)h_{\langle p-m \rangle_N}(Z) \bmod (Z)^N + 1,$$

$$0 \leq p \leq N - 1. \tag{12}$$

Then, it is easy to prove that we can apply the following skew polynomial transform pair to compute the 2-D skew circular convolution:

$$\bar{X}_k(Z) = \sum_{m=0}^{N-1} x_m(Z)Z^{m(2k+1)} \bmod Z^N + 1, \tag{13a}$$

$$x_m(Z) = \frac{1}{N} \sum_{k=0}^{N-1} \bar{X}_k(Z)Z^{-m(2k+1)} \bmod Z^N + 1, \tag{13b}$$

An example of 2-D  $4 \times 4$  skew circular convolution computed by the proposed approach is shown in the appendix.

### 3. Discussions and conclusions

Polynomial-transform-based algorithms for efficiently computing various 2-D generalized GDFTs and GDHTs are investigated in this section. Although, in the previous work [7], the approach used to compute the multidimensional DFTs had been described by the forward and the inverse polynomial-transforms. However, the polynomial-transform-based algorithm presented in [7] is not the same as, although similar to, the proposed skew polynomial transform based algorithm. The proposed approach combined the skills of index permutation and linear congruence to derive the skew polynomial transform based algorithms. Whereas Nussbaumer’s approach [7] is primarily based on the property that  $e^{-j2\pi/N}$  is a root of unity of order  $N$ . In other words, our approach exhibited not only one succinct and efficient algorithm, but also pointed out how to change the 2-D transform into a number of 1-D transforms, more clearly. It is our belief that the proposed approach may provide a new direction to derive fast algorithms for computing 2-D discrete unitary transforms.

### Appendix A

Consider the 2-D SCC of two matrices  $X$  and  $H$  of size  $4 \times 4$ , and let  $X$  and  $H$  be given by

$$X = H = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix}.$$

According to Eq. (13a), we have

$$x_0(Z) = h_0(Z) = 1 + Z + Z^2 + Z^3,$$

$$\bar{X}_0(Z) = \bar{H}_0(Z) = Z^2 + 4Z^3,$$

$$x_1(Z) = h_1(Z) = Z^2 + Z^3,$$

$$\bar{X}_1(Z) = \bar{H}_1(Z) = Z - Z^2 + Z^3,$$

$$x_2(Z) = h_2(Z) = 1 + Z,$$

$$\bar{X}_2(Z) = \bar{H}_2(Z) = 2 + 2Z + 3Z^2,$$

$$x_3(Z) = h_3(Z) = 1 + Z^2 + Z^3,$$

$$\bar{X}_3(Z) = \bar{H}_3(Z) = 2 + Z + Z^2 - Z^3.$$

Suppose the calculation of 1-D skew circular convolutions

$$\bar{X}_k(Z)\bar{H}_k(Z) \bmod Z^4 + 1, 0 \leq k \leq 3,$$

are achieved by 1-D  $H_3$  transform and defined as  $\bar{Y}_k(Z)$ , then we obtain

$$\bar{Y}_0(Z) = -1 - 8Z - 16Z^2,$$

$$\bar{Y}_1(Z) = -3 + 2Z - 2Z^3,$$

$$\bar{Y}_2(Z) = -5 + 8Z + 16Z^2 + 12Z^3,$$

$$\bar{Y}_3(Z) = 5 + 6Z + 4Z^2 - 2Z^3.$$

Finally, the inverse skew polynomial transform of  $\bar{Y}_k(Z)$  can be obtained by Eq. (13b):

$$y_0(Z) = -1 + 2Z + Z^2 + 2Z^3,$$

$$y_1(Z) = -4 - 6Z - 2Z^2,$$

$$y_2(Z) = -1 + 4Z + 2Z^2 + 2Z^3,$$

$$y_3(Z) = -4 - 2Z + 4Z^2 + 10Z^3.$$

That is the skew circular convolution of  $X$  and  $H$  is equal to

$$\begin{bmatrix} -1 & 2 & 1 & 2 \\ -4 & -6 & -2 & 0 \\ -1 & 4 & 2 & 2 \\ -4 & -2 & 4 & 10 \end{bmatrix}.$$

## References

- [1] G. Bi, Y. Chen, Fast generalized DFT and DHT algorithms, *Signal Process.* 65 (1998) 383–390.
- [2] I. Gertner, A new efficient algorithm to compute the two-dimensional discrete fourier transform, *IEEE Trans. Acoustics, Speech, Signal Process.* 36 (1988) 1036–1050.
- [3] N.C. Hou, H.I. Chang, O. Ersoy, Generalized discrete Hartley transform, *IEEE Trans. Signal Process.* 40 (1992) 2931–2940.
- [4] N.C. Hu, F.F. Leu, Fast computation of the two-dimensional generalized Hartley transforms, *IEE Proc. Visual. Image. Signal Process.* 142 (1995) 35–39.
- [5] W. Li, A new algorithm to compute the DCT and its inverse, *IEEE Trans. Signal Process.* 39 (1991) 1305–1313.
- [6] W. Ma, Number of multiplications necessary to compute length- $2^n$  two-dimensional discrete Hartley transform DHT ( $2^n; 2$ ), *Electron. Lett.* 28 (5) (1992) 480–482.
- [7] H.J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*, 2nd Edition, Springer, Berlin, 1982.
- [8] J.-L. Wu, Y.M. Huang, Modularized fast polynomial transform algorithms for two-dimensional digital signal processing, *Proc. IEE, part. F* 137 (1990) 253–261.
- [9] J.-L. Wu, Y.M. Huang, Two-variable modularized fast polynomial transform algorithm for 2-D discrete fourier transforms, *IEEE Trans. Circuits Systems Video Technol.* 2 (1992) 84–87.