



**HAL**  
open science

# Radix-2 algorithm for the fast computation of type-III 3-D discrete $W$ transform

Huazhong Shu, Jiasong Wu, Lotfi Senhadji, Limin M. Luo

► **To cite this version:**

Huazhong Shu, Jiasong Wu, Lotfi Senhadji, Limin M. Luo. Radix-2 algorithm for the fast computation of type-III 3-D discrete  $W$  transform. *Signal Processing*, 2008, 88 (1), pp.210-215. 10.1016/j.sigpro.2007.07.007 . inserm-00304096

**HAL Id: inserm-00304096**

**<https://inserm.hal.science/inserm-00304096>**

Submitted on 22 Jul 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Radix-2 algorithm for the fast computation of type-III 3-D discrete W transform

H.Z. Shu<sup>a, d</sup>, J.S. Wu<sup>a, d</sup>, L. Senhadji<sup>b, c, d</sup>, L.M. Luo<sup>a, d</sup>

<sup>a</sup>Laboratory of Image Science and Technology, School of Computer Science and Engineering, Southeast University, 210096, Nanjing, China;

<sup>b</sup>INSERM, U642, Rennes, F-35000, France;

<sup>c</sup>Université de Rennes 1, LTSI, Rennes, F-35000, France;

<sup>d</sup>Centre de Recherche en Information Biomédicale Sino-Français (CRIBs), China

Information about the corresponding author:

Huazhong Shu, Ph.D

Laboratory of Image Science and Technology

School of Computer Science and Engineering

Southeast University, 210096, Nanjing, China

Tel: 00-86-25-83 79 42 49

Fax: 00-86-25-83 79 26 98

Email: shu.list@seu.edu.cn

Abstract: In this letter, an efficient algorithm for three-dimensional (3-D) vector radix decimation in frequency is derived for computing the type-III 3-D discrete W transform (DWT). Comparison of the computational complexity with the familiar row-column algorithm shows that the proposed approach reduces significantly the number of arithmetic operations. Moreover, the proposed method has very regular structure, thus, it is easy to implement.

Keywords: 3-D DWT, fast algorithm, decimation in frequency

## 1. Introduction

The discrete Hartley transform (DHT) was first introduced by Bracewell [7] and then extended to the generalized DHT (GDHT) [12]. Wang and Hunt [18] presented four versions of the discrete W transform (DWT) which depend on the symmetry type chosen for the sequences in the temporal domain and in the frequency domain. It was shown [17] that the DWT and the type-I, -II, -III and -IV GDHT have a similar definition with difference in constant scaling factors.

The multi-dimensional DHT or type-I DWT has wide application in digital image processing such as image filtering, restoration, encoding and spectrum analysis [9, 14, 15]. The other types of DWT also find their application in various areas. In particular, the type-III DWT was used for harmonic analysis [18] and interpolation [19]. Hu et al. showed [12] that the type-III GDHT can be used to compute the DHT for any number of data points in the least number of operations. It is known that the circular and skew-circular convolutions (CC and SCC) are basic building blocks in the computation of FIR filters, convolutions, and fast Fourier transforms [10]. Wang [18] and Hu [12] showed respectively that the 1-D SCC can be efficiently computed using the type-III DWT. Hu and Lu [13] then extended their method to the computation of 2-D SCC.

In the past decade, a number of algorithms have been reported in the literature for fast computation of DWT or GDHT. For example, Bi [2] developed radix-type algorithms for efficiently computing the type-II, -III and -IV 2-D DWT. Boussakta et al. [5] and Alshibami and Boussakta [1] proposed respectively decimation in time and decimation in frequency methods for calculating the 3-D DHT. Using the polynomial transform, Zeng et al. [20] presented new approaches to speed up the calculation of multi-dimensional ( $m$ -D) DHT. Bouguezel et al. [3, 4] developed respectively split vector radix and vector radix algorithms for computing the 3-D and  $m$ -D DHT. However, to the authors' knowledge, little attention has been paid on the fast computation of type-II, III and -IV 3-D DWT. The 3-D DWT can be calculated using the traditional row-column method [11]. Cheng et al. [8] recently proposed an approach, based on the polynomial transform, to calculate the 3-D DWT. Their method can efficiently reduce the computational complexity compared to the row-column method, but at the expense of very complicated structure.

Inspired by a research work reported in [1], we present in this letter a vector radix decimation in frequency algorithm for the fast computation of the type-III 3-D DWT. Rather than decomposing the 3-D DWT into lower dimensional computation, the proposed approach decomposes the whole computation task into a number of smaller ones. The computational complexity of the method is analyzed and compared to the existing algorithms. We also show briefly how to compute the 3-D SCC via the type-III 3-D DWT.

## 2. Method

The forward type-III 3-D DWT of a real 3-D input data  $x(n_1, n_2, n_3)$  is defined by

$$\begin{aligned}
 & X(k_1, k_2, k_3) \\
 &= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \sum_{n_3=0}^{N_3-1} x(n_1, n_2, n_3) \text{cas} \left[ \frac{\pi}{N_1} (2k_1 + 1)n_1 + \frac{\pi}{N_2} (2k_2 + 1)n_2 + \frac{\pi}{N_3} (2k_3 + 1)n_3 \right] \\
 & \quad 0 \leq n_i \leq N_i ; 0 \leq k_i \leq N_i - 1, i = 1, 2, 3, \quad (1)
 \end{aligned}$$

where  $\text{cas}(\theta) = \cos(\theta) + \sin(\theta)$ . Without loss of generality, we assume that  $N_i = N$ , for  $i = 1, 2, 3$ , where  $N$  is a power of two.

Depending on the parity of  $k_1, k_2$  and  $k_3$ , eight cases need to be considered:  $X(2k_1, 2k_2, 2k_3)$ ,  $X(2k_1, 2k_2, 2k_3+1)$ ,  $X(2k_1, 2k_2+1, 2k_3)$ ,  $X(2k_1+1, 2k_2, 2k_3)$ ,  $X(2k_1, 2k_2+1, 2k_3+1)$ ,  $X(2k_1+1, 2k_2, 2k_3+1)$ ,  $X(2k_1+1, 2k_2+1, 2k_3)$ , and  $X(2k_1+1, 2k_2+1, 2k_3+1)$  for  $0 \leq k_i \leq N/2 - 1$ ,  $i = 1, 2, 3$ . Instead of computing these coefficients directly, we propose in this section an algorithm for 3-D vector radix decimation in frequency suitable for fast computation.

**2.1.** Computation of  $A_1(k_1, k_2, k_3) = \frac{1}{2} [X(2k_1, 2k_2, 2k_3) + X(2k_1+1, 2k_2+1, 2k_3+1)]$ .

$$\begin{aligned} & A_1(k_1, k_2, k_3) \\ &= \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \sum_{n_3=0}^{N-1} x(n_1, n_2, n_3) \cos \frac{\pi(n_1 + n_2 + n_3)}{N} \text{cas} \frac{\pi}{N/2} [(2k_1+1)n_1 + (2k_2+1)n_2 + (2k_3+1)n_3] \quad (2) \\ &= \sum_{n_1=0}^{N/2-1} \sum_{n_2=0}^{N/2-1} \sum_{n_3=0}^{N/2-1} a_1(n_1, n_2, n_3) \text{cas} \phi, \end{aligned}$$

where

$$a_1(n_1, n_2, n_3) = u_{111}(n_1, n_2, n_3) \cos \frac{\pi(n_1 + n_2 + n_3)}{N} + v_{001}(n_1, n_2, n_3) \sin \frac{\pi(n_1 + n_2 + n_3)}{N}, \quad (3)$$

$$\begin{aligned} u_{ijk}(n_1, n_2, n_3) &= x(n_1, n_2, n_3) + (-1)^i x(n_1, n_2 + \frac{N}{2}, n_3 + \frac{N}{2}) + (-1)^j x(n_1 + \frac{N}{2}, n_2, n_3 + \frac{N}{2}) \\ &\quad + (-1)^k x(n_1 + \frac{N}{2}, n_2 + \frac{N}{2}, n_3), \quad i, j, k = 0 \text{ or } 1. \end{aligned} \quad (4)$$

$$\begin{aligned} v_{lmn}(n_1, n_2, n_3) &= x(n_1, n_2, n_3 + \frac{N}{2}) + (-1)^l x(n_1, n_2 + \frac{N}{2}, n_3) + (-1)^m x(n_1 + \frac{N}{2}, n_2, n_3) \\ &\quad + (-1)^n x(n_1 + \frac{N}{2}, n_2 + \frac{N}{2}, n_3 + \frac{N}{2}), \quad l, m, n = 0 \text{ or } 1. \end{aligned} \quad (5)$$

$$\phi = \frac{\pi}{N/2} [(2k_1+1)n_1 + (2k_2+1)n_2 + (2k_3+1)n_3]. \quad (6)$$

Eq. (2) shows that  $A_1(k_1, k_2, k_3)$  is the  $N/2 \times N/2 \times N/2$  type-III DWT.

**2.2.** Computation of  $B_1(k_1, k_2, k_3) = \frac{1}{2} [X(2k_1+1, 2k_2+1, 2k_3+1) - X(2k_1, 2k_2, 2k_3)]$ .

We have

$$B_1(k_1, k_2, k_3) = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \sum_{n_3=0}^{N-1} x(n_1, n_2, n_3) \sin \frac{\pi(n_1 + n_2 + n_3)}{N} \text{cas}(-\phi). \quad (7)$$

Eq. (7) can be rewritten as

$$\begin{aligned} & B_1(N/2-1-k_1, N/2-1-k_2, N/2-1-k_3) \\ &= \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \sum_{n_3=0}^{N-1} x(n_1, n_2, n_3) \sin \frac{\pi(n_1 + n_2 + n_3)}{N} \text{cas} \phi \\ &= \sum_{n_1=0}^{N/2-1} \sum_{n_2=0}^{N/2-1} \sum_{n_3=0}^{N/2-1} b_1(n_1, n_2, n_3) \text{cas} \phi, \end{aligned} \quad (8)$$

which is also the  $N/2 \times N/2 \times N/2$  type-III DWT, and the input sequence  $b_1(n_1, n_2, n_3)$  is given by

$$b_1(n_1, n_2, n_3) = u_{111}(n_1, n_2, n_3) \sin \frac{\pi(n_1 + n_2 + n_3)}{N} - v_{001}(n_1, n_2, n_3) \cos \frac{\pi(n_1 + n_2 + n_3)}{N}. \quad (9)$$

Let

$$A_2(k_1, k_2, k_3) = \frac{1}{2} [X(2k_1 + 1, 2k_2 + 1, 2k_3) + X(2k_1, 2k_2, 2k_3 + 1)], \quad (10)$$

$$B_2(k_1, k_2, k_3) = \frac{1}{2} [X(2k_1 + 1, 2k_2 + 1, 2k_3) - X(2k_1, 2k_2, 2k_3 + 1)], \quad (11)$$

$$A_3(k_1, k_2, k_3) = \frac{1}{2} [X(2k_1 + 1, 2k_2, 2k_3 + 1) + X(2k_1, 2k_2 + 1, 2k_3)], \quad (12)$$

$$B_3(k_1, k_2, k_3) = \frac{1}{2} [X(2k_1 + 1, 2k_2, 2k_3 + 1) - X(2k_1, 2k_2 + 1, 2k_3)], \quad (13)$$

$$A_4(k_1, k_2, k_3) = \frac{1}{2} [X(2k_1 + 1, 2k_2, 2k_3) + X(2k_1, 2k_2 + 1, 2k_3 + 1)], \quad (14)$$

$$B_4(k_1, k_2, k_3) = \frac{1}{2} [X(2k_1 + 1, 2k_2, 2k_3) - X(2k_1, 2k_2 + 1, 2k_3 + 1)], \quad (15)$$

for  $0 \leq k_i \leq N/2 - 1$ ,  $i = 1, 2, 3$ .

Proceeding in a similar way as for  $A_1(k_1, k_2, k_3)$  and  $B_1(k_1, k_2, k_3)$ , we obtain

$$A_2(k_1, k_2, k_3) = \sum_{n_1=0}^{N/2-1} \sum_{n_2=0}^{N/2-1} \sum_{n_3=0}^{N/2-1} a_2(n_1, n_2, n_3) \text{cas} \phi, \quad (16)$$

$$B_2(N/2 - 1 - k_1, N/2 - 1 - k_2, N/2 - 1 - k_3) = \sum_{n_1=0}^{N/2-1} \sum_{n_2=0}^{N/2-1} \sum_{n_3=0}^{N/2-1} b_2(n_1, n_2, n_3) \text{cas} \phi, \quad (17)$$

$$A_3(k_1, k_2, k_3) = \sum_{n_1=0}^{N/2-1} \sum_{n_2=0}^{N/2-1} \sum_{n_3=0}^{N/2-1} a_3(n_1, n_2, n_3) \text{cas} \phi, \quad (18)$$

$$B_3(N/2 - 1 - k_1, N/2 - 1 - k_2, N/2 - 1 - k_3) = \sum_{n_1=0}^{N/2-1} \sum_{n_2=0}^{N/2-1} \sum_{n_3=0}^{N/2-1} b_3(n_1, n_2, n_3) \text{cas} \phi, \quad (19)$$

$$A_4(k_1, k_2, k_3) = \sum_{n_1=0}^{N/2-1} \sum_{n_2=0}^{N/2-1} \sum_{n_3=0}^{N/2-1} a_4(n_1, n_2, n_3) \text{cas} \phi, \quad (20)$$

$$B_4(N/2 - 1 - k_1, N/2 - 1 - k_2, N/2 - 1 - k_3) = \sum_{n_1=0}^{N/2-1} \sum_{n_2=0}^{N/2-1} \sum_{n_3=0}^{N/2-1} b_4(n_1, n_2, n_3) \text{cas} \phi, \quad (21)$$

where

$$a_2(n_1, n_2, n_3) = u_{001}(n_1, n_2, n_3) \cos \frac{\pi(n_1 + n_2 - n_3)}{N} - v_{111}(n_1, n_2, n_3) \sin \frac{\pi(n_1 + n_2 - n_3)}{N}, \quad (22)$$

$$b_2(n_1, n_2, n_3) = u_{001}(n_1, n_2, n_3) \sin \frac{\pi(n_1 + n_2 - n_3)}{N} + v_{111}(n_1, n_2, n_3) \cos \frac{\pi(n_1 + n_2 - n_3)}{N}, \quad (23)$$

$$a_3(n_1, n_2, n_3) = u_{010}(n_1, n_2, n_3) \cos \frac{\pi(n_1 - n_2 + n_3)}{N} + v_{100}(n_1, n_2, n_3) \sin \frac{\pi(n_1 - n_2 + n_3)}{N}, \quad (24)$$

$$b_3(n_1, n_2, n_3) = u_{010}(n_1, n_2, n_3) \sin \frac{\pi(n_1 - n_2 + n_3)}{N} - v_{100}(n_1, n_2, n_3) \cos \frac{\pi(n_1 - n_2 + n_3)}{N}, \quad (25)$$

$$a_4(n_1, n_2, n_3) = u_{100}(n_1, n_2, n_3) \cos \frac{\pi(n_1 - n_2 - n_3)}{N} - v_{010}(n_1, n_2, n_3) \sin \frac{\pi(n_1 - n_2 - n_3)}{N}, \quad (26)$$

$$b_4(n_1, n_2, n_3) = u_{100}(n_1, n_2, n_3) \sin \frac{\pi(n_1 - n_2 - n_3)}{N} + v_{010}(n_1, n_2, n_3) \cos \frac{\pi(n_1 - n_2 - n_3)}{N}, \quad (27)$$

where  $u_{ijk}(n_1, n_2, n_3)$  and  $v_{lmn}(n_1, n_2, n_3)$  are defined in Eqs. (4) and (5).

It is noted that Eqs. (16)–(21) are all the  $N/2 \times N/2 \times N/2$  type-III DWT. When  $A_1(k_1, k_2, k_3)$ ,  $B_1(k_1, k_2, k_3)$ ,  $A_2(k_1, k_2, k_3)$ ,  $B_2(k_1, k_2, k_3)$ ,  $A_3(k_1, k_2, k_3)$ ,  $B_3(k_1, k_2, k_3)$ ,  $A_4(k_1, k_2, k_3)$  and  $B_4(k_1, k_2, k_3)$  are computed, the final output sequences can be obtained with  $N^3$  additions only.

### 3. Computational complexity and comparison analysis

Fig. 1 shows the flowgraph of the realization of the proposed algorithm. The whole process for computing  $N \times N \times N$  type-III DWT using the new approach (including trivial multiplications and additions) requires  $2N^3 \log_2 N$  real multiplications and  $4N^3 \log_2 N$  real additions. On the other hand, if the traditional row-column approach is used to compute the 3-D type-III DWT, it requires  $3N^3 \log_2 N$  real multiplications and  $(9/2)N^3 \log_2 N + 3N^3$  additions. If the algorithm presented in [8] is applied, it needs approximately  $(1/2)N^3 \log_2 N$  real multiplications and  $(7/2)N^3 \log_2 N$  real additions. Fig. 2 shows the number of multiplications plus additions per point, involved in the computation of the  $N \times N \times N$  type-III DWT, using the proposed method, the row-column approach and the algorithm based on the polynomial transform.

It can be seen from Fig. 2 that the polynomial transform algorithm has the best performance in terms of the number of arithmetic operations. This is because it belongs to the non-radix category, which usually requires smaller computational complexity than radix-type algorithm [8]. However, the algorithm based on the polynomial transform needs a process to reorder the input sequence for the conversion process and to reorder the coefficients of the  $(m-1)$ -dimensional polynomial transform to obtain the final transform, which is very difficult to implement in-place and needs extra data storage spaces to store temporary work arrays and extra computation [2, 8, 20]. As stated in [6], the main consideration in choosing a fast algorithm is computational and structural complexities. As the technology of computers and DSPs evolves, the execution time of arithmetic operation has become very short, and regular computational structure becomes the most important factor [6]. Although the proposed algorithm does not achieve the lowest number of arithmetic operations, it has a regular computational structure to achieve desirable features such as regular data indexing and in-place computation [2].

### 4. Fast computation of 3-D SCC

The SCC is usually computed via the separable type-III DWT. By transforming the 2-D separable DWT into the corresponding 2-D DWT, Hu and Lu [13] presented an efficient algorithm for computing the 2-D SCC. We extend here their algorithm to the computation of 3-D SCC. The type-III 3-D separable DWT is defined as [16]

$$\begin{aligned} X_s(k_1, k_2, k_3) \\ = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \sum_{n_3=0}^{N-1} x(n_1, n_2, n_3) \text{cas} \left[ \frac{\pi}{N} (2k_1 + 1)n_1 \right] \text{cas} \left[ \frac{\pi}{N} (2k_2 + 1)n_2 \right] \text{cas} \left[ \frac{\pi}{N} (2k_3 + 1)n_3 \right]. \end{aligned} \quad (28)$$

By using the identity

$$\text{cas}\alpha\text{cas}\beta\text{cas}\gamma = \frac{1}{2}[\text{cas}(\alpha + \beta - \gamma) + \text{cas}(\alpha - \beta + \gamma) + \text{cas}(-\alpha + \beta + \gamma) - \text{cas}(-\alpha - \beta - \gamma)], \quad (29)$$

Eq. (28) can be converted into the 3-D type-III DWT as follows

$$\begin{aligned} X_s(k_1, k_2, k_3) &= \frac{1}{2} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \sum_{n_3=0}^{N-1} x(n_1, n_2, n_3) \left\{ \text{cas} \left[ \frac{\pi}{N} (2k_1 + 1)n_1 + \frac{\pi}{N} (2k_2 + 1)n_2 - \frac{\pi}{N} (2k_3 + 1)n_3 \right] \right. \\ &\quad + \text{cas} \left[ \frac{\pi}{N} (2k_1 + 1)n_1 - \frac{\pi}{N} (2k_2 + 1)n_2 + \frac{\pi}{N} (2k_3 + 1)n_3 \right] \\ &\quad + \text{cas} \left[ -\frac{\pi}{N} (2k_1 + 1)n_1 + \frac{\pi}{N} (2k_2 + 1)n_2 + \frac{\pi}{N} (2k_3 + 1)n_3 \right] \\ &\quad \left. - \text{cas} \left[ -\frac{\pi}{N} (2k_1 + 1)n_1 - \frac{\pi}{N} (2k_2 + 1)n_2 - \frac{\pi}{N} (2k_3 + 1)n_3 \right] \right\} \end{aligned} \quad (30)$$

$$\begin{aligned} &= \frac{1}{2} \{ [X(k_1, k_2, N-1-k_3) + X(k_1, N-1-k_2, k_3)] \\ &\quad + [X(N-1-k_1, k_2, k_3) - X(N-1-k_1, N-1-k_2, N-1-k_3)] \}. \end{aligned}$$

It can be easily seen from Eq. (30) that

$$\begin{aligned} X_s(k_1, N-1-k_2, N-1-k_3) &= \frac{1}{2} \{ [X(k_1, k_2, N-1-k_3) + X(k_1, N-1-k_2, k_3)] \\ &\quad - [X(N-1-k_1, k_2, k_3) - X(N-1-k_1, N-1-k_2, N-1-k_3)] \}. \end{aligned} \quad (31)$$

Based on Eqs. (30) and (31),  $X_s(k_1, k_2, k_3)$  can be computed from  $X(k_1, k_2, k_3)$  with additional  $2N^3$  real additions. Therefore, the computation of the separable 3-D type-III DWT requires  $2N^3 \log_2 N$  real multiplications and  $4N^3 \log_2 N + 2N^3$  real additions.

## 5. Conclusions

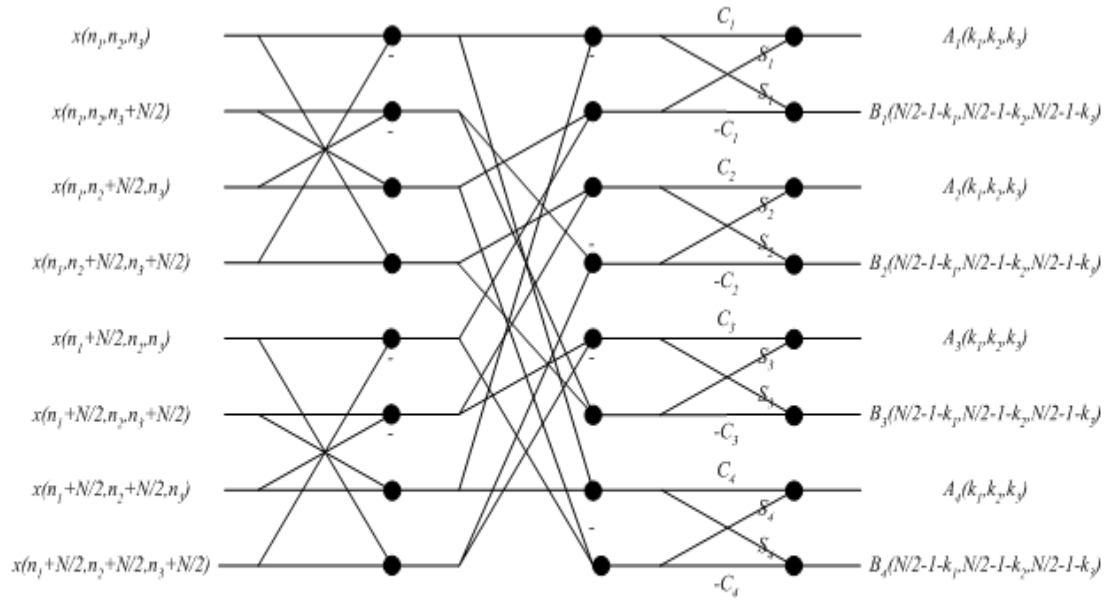
In this letter, we have presented a new 3-D vector radix decimation in frequency algorithm for efficient computation of type-III 3-D DWT. We have also discussed the application of type-III 3-D DWT in the fast computation of 3-D SCC. The computational complexity of the proposed approach has been analyzed and compared with the existing algorithms. The comparison shows that the new algorithm is more efficient than the traditional row-column method, and it needs more arithmetic operations than that required by the polynomial transform algorithm presented in [8]. However, the proposed algorithm has significantly reduced structural complexities compared to the polynomial transform algorithm reported in [8]. The simplicity and regularity of the computational structure achieved by the proposed approach permit minimizing the overall computational costs for high processing throughput.

**Acknowledgements :** This work was supported by National Basic Research Program of China under grant no. 2003CB716102 and Program for New Century Excellent Talents in University under grant no. NCET-04-0477. It has been granted by all partners of the Joint Research Centre, CRIBs (INSERM and the Universities of Southeast and Rennes 1) and French Foreign Ministry.

## References

- [1] O. Alshibami, S. Boussakta, Fast 3-D decimation in frequency algorithm for 3-D Hartley transform, *Signal Processing* 82 (2002) 121-126.
- [2] G.A. Bi, Fast algorithms for the 2-D discrete W transform, *Signal Processing* 74 (1999) 297-308.
- [3] S. Bouguezel, M.O. Ahmed, M.N.S. Swamy, A split vector radix algorithm for the 3-D discrete Hartley transform, *IEEE Trans. Circuits Syst.-I: Regular papers* 53 (2006) 1966-1976.
- [4] S. Bouguezel, M.N.S. Swamy, M.O. Ahmed, Multidimensional vector radix FHT algorithms, *IEEE Trans. Circuits Syst.-I: Regular papers* 53 (2006) 905-917.
- [5] S. Boussakta, O. Alshibami, M. Aziz, Radix-2 $\times$ 2 $\times$ 2 algorithm for the 3-D discrete Hartley transform, *IEEE Trans. Signal Process.* 49 (2001) 3145-3156.
- [6] S. Boussakta, O. Alshibami, Fast algorithm for the 3-D DCT-II, *IEEE Trans. Signal Process.* 52 (2004) 992-1001.
- [7] R. N. Bracewell, Discrete Hartley transform. *J. Opt. Soc. Amer.* 73 (1983) 1832-1835.
- [8] L.Z. Cheng, Z. Wang, Z.H. Zhang, Fast unified computation of the multi-dimensional discrete sinusoidal transforms, *Appl. Math. Comput.* 132 (2002) 455-487.
- [9] A. Erdi, E. Yorke, M. Loew, Y. Erdi, M. Sarfaraz, B. Wessels, Use of the fast Hartley transform for three-dimensional dose calculation in radionuclide therapy, *Med. Phys.* 25 (1998) 2226-2233.
- [10] O.K. Ersoy. Semisystolic array implementation of circular, skew circular, and linear convolutions, *IEEE Trans. Computers* c-34 (1985) 190-196.
- [11] H. Hao, R.N. Bracewell, A three-dimensional DFT algorithm using the fast Hartley transform, *Proc. IEEE* 75 (1987) 264-266.
- [12] N.C. Hu, H.I. Chang, O.K. Ersoy, Generalized discrete Hartley transforms. *IEEE Trans. Signal Process.* 40 (1992) 2931-2940.
- [13] N.C. Hu, F.F. Lu, Fast computation of the two-dimensional generalised Hartley transforms. *IEE Proc. Vis. Image Signal Process.* 142 (1995) 35-39.
- [14] S.A. Mahmoud, Motion analysis of multiple moving objects using Hartley transform, *IEEE Trans. Syst. Man Cybernet.* 21 (1991) 280-287.
- [15] C.H. Paik, M.D. Fox, Fast Hartley transforms for image processing, *IEEE Trans. Med. Imaging* 7 (1988) 149-153.
- [16] M.G. Perkins, A separable Hartley-like transform in two or more dimensions. *Proc. IEEE*, 75 (1987) 1127-1129.
- [17] Z. Wang, Comments on "Generalized discrete Hartley transforms", *IEEE Trans. Signal Process.* 43 (1995) 1711-1712.
- [18] Z. Wang, B. R. Hunt, The discrete W transform, *Appl. Math. Comput.* 16 (1985) 19-48.
- [19] Z. Wang, G.A. Jullien, W.C. Miller, The generalized discrete W transform and its application to interpolation. *Signal Processing* 36 (1994) 99-109.
- [20] Y.H. Zeng, G.A. Bi, A.R. Leyman, New algorithms for multidimensional discrete Hartley transform, *Signal Processing* 82 (2002) 1086-1095.





$$C_1 = \cos\left[\frac{\pi}{N}(n_1 + n_2 + n_3)\right], C_2 = \cos\left[\frac{\pi}{N}(n_1 + n_2 - n_3)\right], C_3 = \cos\left[\frac{\pi}{N}(n_1 - n_2 + n_3)\right], C_4 = \cos\left[\frac{\pi}{N}(n_1 - n_2 - n_3)\right]$$

$$S_1 = \sin\left[\frac{\pi}{N}(n_1 + n_2 + n_3)\right], S_2 = \sin\left[\frac{\pi}{N}(n_1 + n_2 - n_3)\right], S_3 = \sin\left[\frac{\pi}{N}(n_1 - n_2 + n_3)\right], S_4 = \sin\left[\frac{\pi}{N}(n_1 - n_2 - n_3)\right]$$

Fig.1. 3-D vector-radix algorithm for 3-D type-III DWT

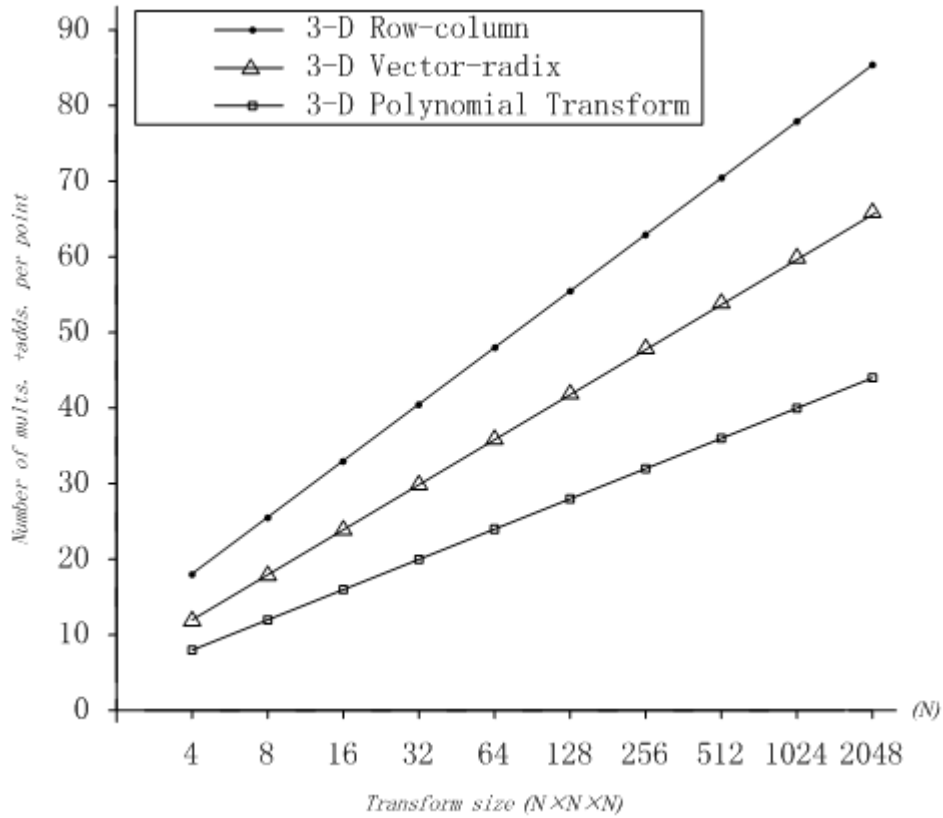


Fig.2. Arithmetic complexity comparison