Direct computation of length-N DHT from three adjacent length-N/3 DHT coefficients
Jiasong Wu, Lu Wang, Huazhong Shu, Lotfi Senhadji

To cite this version:

HAL Id: inserm-00535838
http://www.hal.inserm.fr/inserm-00535838
Submitted on 13 Nov 2010

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.
Direct Computation of Length-$N$ DHT from Three Adjacent Length-$N/3$ DHT Coefficients

Jiasong Wu, Lu Wang, Huazhong Shu
Laboratory of Image Science and Technology (LIST)
Southeast University
Nanjing, China
Centre de Recherche en Information Biomédical Sino-Français, Nanjing, China

Lotfi Senhadji
INSERM, U642, Rennes, France
LTSI, Université de Rennes 1,
Rennes, France
Centre de Recherche en Information Biomédical Sino-Français, Rennes, France

Abstract—A fast direct method for obtaining the length-$N$ discrete Hartley transform (DHT) coefficients from three adjacent length-$N/3$ DHT coefficients is presented. The proposed method reduces significantly the number of arithmetic operations compared to the traditional approach. Furthermore, it is easy to implement.

Keywords—compressed-domain processing; discrete Hartley transform; fast algorithm

I. INTRODUCTION

The discrete Hartley transform (DHT) introduced by Bracewell in 1983 [1] has become an important tool in signal and image processing [2-5]. Many fast algorithms for computing the DHT have already been reported in the literature. Sorensen, Jones, Burrus, and Heideman [6] proposed a set of fast algorithms for DHT, including radix-2 decimation-in-frequency (DIF), radix-4, split radix, prime factor [7], and Winograd transform algorithms. Zhao [8] derived a radix-3 fast Hartley transform (FHT) algorithm, which was then extended to 2-D DHT [9]. Bi [10] suggested a radix-3/9 FHT algorithm for further speeding up the computation of DHT. Bouguezzez, Ahmad, and Swamy [11] proposed an optimized split-radix FHT algorithm, taking arithmetic complexity, the number of data transfers, address generations, and twiddle factor evaluations into consideration, for computing the DHT of sequences with length $N = q \times 2^n$, where $q$ is an odd integer.

Since the DHT can be used in signal or image compression techniques [2-4], a problem that is often encountered is how to directly manipulate a compressed data stored in the DHT domain. In particular, how to construct a long DHT sequence from several short DHT sequences? This kind of questions was first discussed in discrete cosine transform (DCT) domain by Kou and Fjallbrandt [12], who proposed an efficient way for computing the length-$N$ DCT coefficients when the two adjacent length-$N/2$ DCT coefficients are known. Their algorithm was then improved by Skodras [13], whose algorithm was further improved and extended to two dimensional DCT by Chuang and Wu [14] and multidimensional DCT by Dai, Chen, and Lin [15]. Jiang and Feng [16] discussed the spatial relationship between an image block of any size and all of its subblocks in the DCT domain. In [17], Pei and Kao presented an efficient method for obtaining the length-$N$ DCT coefficients from three adjacent length-$N/3$ DCT coefficients. More recently, Shu, Wang, Senhadji, and Luo [18] realized a direct computation of length-$N$ type-II generalized discrete Hartley transform (GDHT) coefficients from two adjacent length-$N/2$ GDHT coefficients. Shu’s algorithm can be easily extended to the computation of DHT. In this paper, we address the problem of how to efficiently compute the length-$N$ DHT coefficients from three consecutive length-$N/3$ DHT coefficients. Specifically, assume a length-$N$ sequence $x_n$ is created by the concatenation of three length-$N/3$ adjacent sequences, i.e. $a_n = x_{3n}$, $b_n = x_{n+N/3}$, $c_n = x_{n+2N/3}$, for $n = 0, 1, \ldots, N/3-1$, and $A_k$, $B_k$, and $C_k$ are their DHT coefficients, respectively. How can we efficiently compute the length-$N$ DHT coefficients $X_k$ when $A_k$, $B_k$, and $C_k$ are known? We develop a new method, which is similar to the approach presented in [17], to solve the above mentioned question in the DHT domain.

II. TRADITIONAL APPROACH

The schematic representation of the traditional approach is shown in Fig. 1. According to the scheme, three length-$N/3$ inverse DHTs (IDHTs) and one length-$N$ DHT are required. In this section, we apply two different schemes, based respectively on the algorithms presented in [8] and [11], to implement the length-$N$ DHT according to the different values of $N$.

![Figure 1. Schematic representation of traditional approach](image)

A. First scheme: Decomposing length-$N$ DHT into three length-$N/3$ DHTs by using Zhao’s algorithm [8].

When $N$ is divisible by 3, i.e., $N = 3p$, we can apply Zhao’s DIF radix-3 FHT algorithm [8] to decompose length-$N$ DHT.
into three length-\(N/3\) DHTs. We will reformulate Zhao’s algorithm and analyze its computational complexity in the rest of this subsection.

The normalized DHT of the sequence \(x_n\) is defined by
\[
x_k = \text{DHT}_N \{x_n\} = \sum_{n=0}^{N-1} x_n \text{cas} \frac{2\pi nk}{N},
\]
and the corresponding IDHT is
\[
x_n = \text{IDHT}_N \{x_k\} = \frac{1}{N} \sum_{k=0}^{N-1} X_k \text{cas} \frac{2\pi nk}{N},
\]
with \(\text{cas}\theta = \cos \theta + \sin \theta\).

According to [8], we can realize three formulations
\[
\begin{align*}
D_k &= X_{3k} = \sum_{n=0}^{N/3-1} (a_n + b_n + c_n) \text{cas} \frac{2\pi nk}{N / 3}, \\
E_k &= X_{3k+1} + X_{3k-1} = \sum_{n=0}^{N/3-1} \left\{2a_n - (b_n + c_n)\right\}\cos \theta_n \text{cas} \frac{2\pi nk}{N / 3}, \\
F_k &= X_{3(N/3-k)+1} - X_{3(N/3-k)-1} = \sum_{n=0}^{N/3-1} \left\{2a_n - (b_n + c_n)\right\}\sin \theta_n \text{cas} \frac{2\pi nk}{N / 3},
\end{align*}
\]
where \(a_n, b_n,\) and \(c_n\) are the three adjacent sequences and \(\theta_n = 2\pi n / N\).

Once \(E_k\) and \(F_k\) are computed, \(X_{3k+1}\) and \(X_{3k-1}\) can be obtained by (7). The schematic representation of the proposed method is depicted in Fig. 2.

### III. PROPOSED METHOD

In this section, we propose a new method which is based on Zhao’s radix-3 FHT algorithm to calculate the length-\(N\) DHT coefficients from three consecutive length-\(N/3\) DHT coefficients.

#### A. Calculation of \(D_k = X_{3k}\), \(k = 0, 1, ..., N/3 - 1\).

From (3), we have
\[
D_k = X_{3k} = A_k + B_k + C_k
\]

#### B. Calculation of \(E_k = X_{3k+1} + X_{3k-1}, k = 0, 1, ..., N/3 - 1\).

From (4), we have
\[
E_k = \text{DHT}_{N/3} \left\{\text{IDHT}_{N/3} \{2A_k - (B_k + C_k)\}\cos \theta_n \right\} - \text{IDHT}_{N/3} \{B_k - C_k\} \sqrt{3} \sin \theta_n
\]

#### C. Calculation of \(F_k = X_{3(N/3-k)+1} - X_{3(N/3-k)-1}, k = 0, 1, ..., N/3 - 1\).

From (5), we have
\[
F_k = \text{DHT}_{N/3} \left\{\text{IDHT}_{N/3} \{2A_k - (B_k + C_k)\}\sin \theta_n \right\} + \text{IDHT}_{N/3} \{B_k - C_k\} \sqrt{3} \cos \theta_n
\]

Once \(E_k\) and \(F_k\) are computed, \(X_{3k+1}\) and \(X_{3k-1}\) can be obtained by (7). The schematic representation of the proposed method is depicted in Fig. 2.
IV. COMPUTATIONAL COMPLEXITY

We now analyze the computational complexity of the proposed method and the traditional approach. Since the same kernel is used for the DHT and the IDHT, they require the same computational complexity.

According to Fig. 2, the computational complexity of the proposed method is given by

\[
M_N^P = 4N / 3 - 3 + 4M_{N/3}^{DHT},
\]

\[
A_N^P = 8N / 3 - 3 + 4A_{N/3}^{DHT},
\]

(13)

The computational complexity of the traditional approach shown in Fig. 1 is as follows

\[
M_N^T = 3M_{N/3}^{DHT} + M_N^{DHT},
\]

\[
A_N^T = 3A_{N/3}^{DHT} + A_N^{DHT},
\]

(14)

A. Traditional approach using Zhao’s algorithm[8] (First Scheme)

When Zhao’s algorithm is used in the traditional approach, substitution of (8) into (14) yields

\[
M_N^T = 6M_{N/3}^{DHT} + 4N / 3 - 3,
\]

\[
A_N^T = 6A_{N/3}^{DHT} + 8N / 3 - 2.
\]

(15)

1) When length-N = \(3 \times 2^m\), \(m \geq 2\).

In this case, the split radix FHT algorithm [11] can be used to efficiently compute length-N/3 = \(2^m\) DHTs where the computational complexity is shown in (9). The comparison results of the proposed algorithm and this scheme are shown in table I.

2) When length-N is other values

In this case, the prime factor DHT algorithm [6][7] can be used to efficiently compute length-N/3 DHTs. The computational complexity of this algorithm can be found in [7]. Note that the number of additions involved in computing the DHT via prime-factor FFT algorithm for real-valued series [7] is calculated by adding \((N-2)\) to those involved in computing the DFT. The number of multiplications required for computing the DHT is the same as those of the DFT computation. The comparison results of the proposed algorithm and this scheme are shown in table II.

B. Traditional approach using Bouguezel’s algorithm[11] (Second Scheme)

When Bouguezel’s algorithm is used in the traditional approach, setting \(q = 3\) in (9), and then substituting the results into (14), we have

\[
M_N^T = 3M_{N/3}^{DHT} + M_{N/2}^{DHT} + 4M_{N/8}^{DHT} + N - 24, \quad N > 24.
\]

\[
A_N^T = 3A_{N/3}^{DHT} + A_{N/2}^{DHT} + 4A_{N/8}^{DHT} + 3N - 24.
\]

(16)

We summarize the computational complexity of the proposed method and the second scheme of traditional approach in table III.

![Table I: Computational Complexity of the Proposed Method and the First Scheme of Traditional Approach (N=500).](image1)

<table>
<thead>
<tr>
<th>N</th>
<th>Traditional approach [8][11]</th>
<th>Proposed method</th>
<th>Saved (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>13</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>24</td>
<td>41</td>
<td>56</td>
<td>31</td>
</tr>
<tr>
<td>48</td>
<td>121</td>
<td>170</td>
<td>39</td>
</tr>
</tbody>
</table>

![Table II: Computational Complexity of the Proposed Method and the First Scheme of Traditional Approach (N=500).](image2)

<table>
<thead>
<tr>
<th>N</th>
<th>Traditional approach [7][8]</th>
<th>Proposed method</th>
<th>Saved (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>5</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>42</td>
<td>50</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>26</td>
<td>50</td>
</tr>
</tbody>
</table>

![Table III: Computational Complexity of the Proposed Method and the Second Scheme of Traditional Approach (N=500).](image3)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>4</td>
<td>6</td>
<td>33</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>12</td>
<td>33</td>
</tr>
</tbody>
</table>

As stated in [19], for modern computers, the run time for a multiplication is almost the same as that for an addition. They are exactly the same for some advanced signal processors, such as the series of TMS320. Therefore, we give the savings of the proposed algorithm in terms of the total number of multiplications and additions. It can be concluded from the tables that the proposed method reduces significantly the
number of arithmetic operations compared to the first scheme of traditional approach. Although the reduction in terms of the arithmetic operations of the proposed method is less remarkable compared to the second scheme, it needs only to compute two length-$N/3$ DHTs and IDHTs, which can be implemented with one structure for sequences with length-$2^n$ [11]. On the other hand, for the second scheme of traditional approach, the computation of three length-$N/3$ IDHTs needs one structure with length-$2^n$, but the computation of length-$N$ DHT requires another structure with length-$3\times2^n$. Furthermore, the proposed algorithm supports a more wide range of choices of sequence length when compared to the second scheme whose length is limited to length-$3\times2^n$.

V. CONCLUSIONS

We have proposed in this paper an efficient method for the direct computation of length-$N$ DHT from three adjacent length-$N/3$ DHT coefficients. Results show that the proposed method allows reducing the number of arithmetic operations in comparison with the traditional approaches. Another important feature of the proposed algorithm is its savings in memory and data transfers. Since the method proposed by Shu, Wang, Senhadji, and Luo [18] for the direct computation of type-II GDHT is also applicable to the DHT, its combination with the proposed method could find applications in signal processing tasks. In fact, scaling a DHT-based signal by the factor of $1/(2^n3^q)$ can be easily obtained by cascading $p$ for Shu’s structures and $q$ for the authors’ structures.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China under Grant 60873048, by the National Basic Research Program of China under Grant 2010CB732503, by a Program of Jiangsu Province under Grant SBK200910055 and the Natural Science Foundation of Jiangsu Province under Grant BK2008279. This work was also supported by the “Eiffel Doctorate” excellence grant of the French Ministry of Foreign and European Affairs.

REFERENCES


