第6期 电 子 学 报 Vol.36 No.6
2009年6月 ACTA ELECTRONICA SINICA Jun. 2009

一种新的三维离散 Hartley 变换的分裂基快速算法

姜龙玉,舒华忠,伍家松,罗立民
(东南大学影像科学与技术实验室,江苏南京 210096)

摘 要: 离散 Hartley 变换(Discrete Hartley Transform,DHT)作为实值离散傅立叶变换的一种替代,在信号和图像处理领域已有广泛应用,针对现有三维 DHT 快速算法均仅能计算长度为2的整数次幂的 DHT,本文提出一种适用于更多不同长度三维 DHT 的分裂基-2/4快速算法,较之将已有最优算法补零计算的方法,该算法有效的降低了计算复杂度.

关键词: 三维 DHT; 分裂基; 快速算法
中图分类号: TP301 **文献标识码**: A **文章编号**: 0372-2112 (2009) 06-1252-04

A Novel Split-Radix Fast Algorithm for the 3-D Discrete Hartley Transform

JIANG Long-yu,SHU Hua-zhong,WU Jia-song,LUO Li-min
(*Laboratory of Image Science and Technology, Southeast University, Nanjing* 210096, *China*)

Abstract: The discrete Hartley transform has been proposed as an alternative tool suitable for DFT referring real data and has been used in many signal and image processing applications. Because the existed algorithms for three-dimension discrete Hartley transform(3-D DHT) only can compute sequences whose lengths are the m th power of 2(m is an integer), this paper proposes a novel split-radix-2/4 algorithm for the fast computation of 3-D DHT, which provides more flexibility in selecting the sequence length. Moreover, it achieves a large reduction in computational complexity compared to computing by zero padding.

Key words: 3-D DHT; split-radix; fast algorithm

1 引言

DHT 是一种从实数到实数的变换. 在实信号处理领域,DHT 替代 DFT 有广泛应用. 一维、二维 DHT 已有较多有效的快速算法. 近年来,由于三维信号处理应用增多,三维 DHT 的快速算法研究受到较大关注[1~4]. 其中,基-2[1,2]和分裂基-2/4[3,4]的三维 DHT 快速算法,是目前较有效的三维 DHT 快速算法,前者算法结构规则,易于实现,后者在规则的结构和较低的算术复杂度两方面做到了较好的平衡,但这些算法都仅适用于长度为 $N=q\times 2^m$ 的情形,当序列长度不是2的整数次幂时,需要用补零的方法使序列长度变为2的整数次幂,而补零的方法会增加不必要的计算. 文献[5,6]中提出了适用于 $N=q\times 2^m$ 的一维或二维 DHT 分裂基算法,由于多维 DHT 的不可分离性,这些算法不能被直接推广用于计算三维 DHT,因此本文提出一种适用于长度 $N=q\times 2^m$(q 为一个奇整数)的三维 DHT 的分裂基-2/4快速算法,并采用 Kronecker 乘积和矩阵表示,易于算法实现.

2 三维 DHT 的分裂基快速算法

三维离散信号 $x(n_1,n_2,n_3)$ 的 DHT 定义如下:

$$X(k_1,k_2,k_3)=\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}\sum_{n_3=0}^{N-1}x(n_1,n_2,n_3)\,\mathrm{cas}\left(\frac{2\pi}{N}\sum_{i=1}^{3}n_ik_i\right)$$
$$(0\le k_i\le N-1,\ i=1,2,3)\qquad(1)$$

其中 $\mathrm{cas}\,\theta=\cos\theta+\sin\theta$, $N=q\times 2^m$(q 为奇整数且 $m>0$).

为了有效计算式(1)中的 $X(k_1,k_2,k_3)$,我们可以通过计算其八个分项:偶-偶-偶,偶-偶-奇,偶-奇-偶,奇-偶-偶,偶-奇-奇,奇-偶-奇,奇-奇-奇实现. 偶-偶-偶的输出项可由下式得到:

$$X(2k_1,2k_2,2k_3)=\sum_{n_1=0}^{N/2-1}\sum_{n_2=0}^{N/2-1}\sum_{n_3=0}^{N/2-1}y_{n_1,n_2,n_3}(0)\,\mathrm{cas}\left(\frac{2\pi}{N/2}\sum_{i=1}^{3}n_ik_i\right)\qquad(2)$$

对于其它项的计算,根据序列长度 N 的取值,分两种情况予以讨论.

2.1 N=2q

当 $m=1$,即 $N=2q$ 时,偶-偶-奇,偶-奇-偶,奇-

收稿日期:2008-04-03;修回日期:2009-01-22
基金项目:国家自然科学基金(No.60873048);教育部长江学者和创新团队发展计划;江苏省自然科学基金(No.BK2008279)

偶-偶,偶-奇-奇,奇-偶-奇,奇-奇-偶,奇-奇-奇的输出项可由下式计算得到:

$$X(2k_1+p_1q, 2k_2+p_2q, 2k_3+p_3q) = \sum_{n_1=0}^{q-1}\sum_{n_2=0}^{q-1}\sum_{n_3=0}^{q-1} y_{n_1,n_2,n_3}(p)(-1)^{\sum_{i=1}^{3} n_i r_i}\mathrm{cas}\left(\frac{2\pi}{q}\sum_{i=1}^{3}n_ik_i\right)$$

$(0 \le k_i \le q-1, i=1,2,3)$,此式中 $p \ne 0$ (3)

若无特殊说明,文中 $p=0,1,\cdots,7$, $p_1=\lfloor p/4 \rfloor$, $p_2=\lfloor p/2 \rfloor \bmod 2$, $p_3=p \bmod 2$ "$\lfloor\ \rfloor$"、"mod"分别表示向下取整、取余. 式(2)、(3)中 y_{n_1,n_2,n_3},可由初始输入序列从下式得到:

$$y_{n_1,n_2,n_3} = H_2^3 x_{n_1,n_2,n_3} (0 \le n_i \le q-1) \tag{4}$$

$$x_{n_1,n_2,n_3}(p) = x(n_1+p_1q, n_2+p_2q, n_3+p_3q)$$
$$y_{n_1,n_2,n_3}(p) = y_{p_1,p_2,p_3}(n_1,n_2,n_3) \tag{5}$$

$H_2^3 = H_2 \otimes H_2 \otimes H_2$,其中 $H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$,"$\otimes$"表示 Kronecker 乘积[8].

2.2 N ≥ 4q

当 $m \ge 2$,即 $N \ge 4q$ 时,偶-偶-奇,偶-奇-偶,奇-偶-偶,偶-奇-奇,奇-偶-奇,奇-奇-偶,奇-奇-奇的输出项可由式(6)计算得到:

$$X[(N+4k_1 \pm r_1q) \bmod N, (N+4k_2 \pm r_2q) \bmod N, (N+4k_3 \pm r_3q) \bmod N]$$
$$= \sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}\sum_{n_3=0}^{N-1} x(n_1,n_2,n_3)[\cos\alpha\,\mathrm{cas}\beta \pm \sin\alpha\,\mathrm{cas}(-\beta)]$$
$$= \sum_{n_1=0}^{N/4-1}\sum_{n_2=0}^{N/4-1}\sum_{n_3=0}^{N/4-1} f_{r_1,r_2,r_3}\mathrm{cas}(\beta) \pm \sum_{n_1=0}^{N/4-1}\sum_{n_2=0}^{N/4-1}\sum_{n_3=0}^{N/4-1} g_{r_1,r_2,r_3}\mathrm{cas}(-\beta) \tag{6}$$

其中 $\alpha = \frac{2\pi}{N/q}\sum_{i=1}^{3} n_ir_i$, $\beta = \frac{2\pi}{N/4}\sum_{i=1}^{3} n_ir_i$

$(0 \le k_1, k_2, k_3 \le \frac{N}{4} - 1)$

f_{r_1,r_2,r_3}、g_{r_1,r_2,r_3}由以下(a)~(g)的分解过程计算得到,r_1, r_2, r_3 的值为分解过程中分别设定的值.

(a) 偶-偶-奇输出项的分解过程

$(r_3=1, r_1=0$ 或 $2, r_2=0$ 或 $2)$

$$\begin{bmatrix} f_{r_1,r_2,1} \\ g_{r_1,r_2,1} \end{bmatrix} = \begin{bmatrix} C_{001} & -S_{001} \\ S_{001} & C_{001} \end{bmatrix}\begin{bmatrix} I_4 & 0 \\ 0 & D \end{bmatrix}\begin{bmatrix} B_{001} & R_1B_{001} \\ -R_2B'_{001} & -B'_{001} \end{bmatrix} Y_{001} \tag{7}$$

其中

$$f_{r_1,r_2,1}(r) = f_{r_1,r_2,1}(n_1,n_2,n_3), \quad g_{r_1,r_2,1}(r) = g_{r_1,r_2,1}(n_1,n_2,n_3)$$
$$C_{001}(r,r) = \cos\alpha, S_{001}(r,r) = \sin\alpha, r = r_1 + r_2 \times \frac{1}{2} \tag{8}$$

在以下(b)~(g)情形中均有类似的表示,为节省篇幅,仅写出 r_1, r_2, r_3 与 r 之间关系等式

I_L 为 L 阶单位矩阵, $D = (-1)^{(q-1)/2} I_4$,

I_L^{QP} 为 I_L 的第 P、Q 行互换后的矩阵, $H_2^2 = H_2 \otimes H_2$

$$B_{001} = I_4^{32} H_2^2(I_2 \otimes J_1), B'_{001} = I_4^{32} H_2^2(I_2 \otimes J_2) \tag{9}$$

$$J_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, J_2 = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix} \tag{10}$$

$$R_1 = \mathrm{diag}(1,1,-1,-1), R_2 = \mathrm{diag}(-1,-1,1,1) \tag{11}$$

$$y_{001}(p) = y_{001}(n_1+p_1N/4, n_2+p_2N/4, n_3+p_3N/4) \tag{12}$$

(b) 偶-奇-偶输出项的分解过程

$(r_2=1, r_1=0$ 或 $2, r_3=0$ 或 $2)$

$$\begin{bmatrix} f_{r_1,1,r_3} \\ g_{r_1,1,r_3} \end{bmatrix} = \begin{bmatrix} C_{010} & -S_{010} \\ S_{010} & C_{010} \end{bmatrix}\begin{bmatrix} I_4 & 0 \\ 0 & D \end{bmatrix}\begin{bmatrix} B_{010} & R_1B_{010} \\ R_2B'_{010} & -B'_{010} \end{bmatrix} Y_{010} \tag{13}$$

其中 $r = r_1 + r_3 \times (1/2)$ (14)

$$B_{010} = H_2^2(J_1 \otimes I_2), \quad B'_{010} = H_2^2(J_2 \otimes I_2) \tag{15}$$

$$y_{010}(p) = y_{010}(n_1+p_1N/4, n_2+p_2N/4, n_3+p_3N/4) \tag{16}$$

(c) 奇-偶-偶输出项的分解过程$(r_1=1, r_2=0$ 或 $2, r_3=0$ 或 $2)$

$$\begin{bmatrix} f_{1,r_2,r_3} \\ g_{1,r_2,r_3} \end{bmatrix} = \begin{bmatrix} C_{100} & -S_{100} \\ S_{100} & C_{100} \end{bmatrix}\begin{bmatrix} I_4 & 0 \\ 0 & D \end{bmatrix}\begin{bmatrix} B_{100} & 0 \\ 0 & B_{100} \end{bmatrix} Y_{100} \tag{17}$$

其中: $B_{100} = H_2^2, r = r_2 + r_3 \times (1/2)$ (18)

$$y_{100}(p) = y_{100}(n_1+p_1N/4, n_2+p_2N/4, n_3+p_3N/4) \tag{19}$$

(d) 偶-奇-奇输出项的分解过程

$(r_3=1, r_1=0$ 或 $2, r_2=1$ 或 $-1)$

$$\begin{bmatrix} f_{r_1,r_2,1} \\ g_{r_1,r_2,1} \end{bmatrix} = \begin{bmatrix} C_{011} & -S_{011} \\ S_{011} & C_{011} \end{bmatrix}\begin{bmatrix} I_4 & 0 \\ 0 & D \end{bmatrix}\begin{bmatrix} B_{011} & R_1B_{011} \\ I_4^{43}B'_{011} & R_2I_4^{21}B'_{011} \end{bmatrix} Y_{011} \tag{20}$$

其中

$$B_{011} = I_4^{21} H_2^4\begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix}, B'_{011} = H_2^4\begin{bmatrix} J'_1 & 0 \\ 0 & J'_2 \end{bmatrix} \tag{21}$$

$$J'_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, J'_2 = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} \tag{22}$$

$$r = r_1 + (r_3 - r_2) \times (1/2) \tag{23}$$

$$y_{011}(p) = y_{011}(n_1+p_1N/4, n_2+p_2N/4, n_3+p_3N/4) \tag{24}$$

(e) 奇-偶-奇输出项的分解过程

$(r_3=1, r_2=0$ 或 $2, r_1=1$ 或 $-1)$

$$\begin{bmatrix} f_{r_1,r_2,1} \\ g_{r_1,r_2,1} \end{bmatrix} = \begin{bmatrix} C_{101} & -S_{101} \\ S_{101} & C_{101} \end{bmatrix} \begin{bmatrix} I_4 & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} B_{101} & -B'_{101} \\ R_3 B'_{101} & R_3 B_{101} \end{bmatrix} Y_{101} \tag{25}$$

$$B_{101} = H_2^{4}(I_2 \otimes J_1),\ B'_{101} = H_2^{4}(I_2 \otimes J_2) \tag{26}$$

$$R_3 = \mathrm{diag}(-1, 1, -1, 1) \tag{27}$$

$$r = r_2 + (r_3 - r_1) \times (1/2) \tag{28}$$

$$y_{101}(p) = y_{101}(n_1 + p_1 N/4, n_2 + p_2 N/4, n_3 + p_3 N/4) \tag{29}$$

(*f*) 奇-奇-偶输出项的分解过程

($r_2 = 1$, $r_1 = 1$ 或 -1, $r_3 = 0$ 或 2)

$$\begin{bmatrix} f_{r_1,1,r_3} \\ g_{r_1,1,r_3} \end{bmatrix} = \begin{bmatrix} C_{110} & -S_{110} \\ S_{110} & C_{110} \end{bmatrix} \begin{bmatrix} I_4 & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} B_{110} & -B'_{110} \\ R_3 B'_{110} & R_3 B_{110} \end{bmatrix} Y_{110} \tag{30}$$

$$B_{110} = I_4^{32} H_2^{2}(J_1 \otimes I_2),\ B'_{110} = I_4^{32} H_2^{2}(J_2 \otimes I_2) \tag{31}$$

$$r = r_3 + (r_2 - r_1) \times (1/2) \tag{32}$$

$$y_{110}(p) = y_{110}(n_1 + p_1 N/4, n_2 + p_2 N/4, n_3 + p_3 N/4) \tag{33}$$

(*g*) 奇-奇-奇输出项的分解过程

($r_3 = 1$, $r_1 = 1$ 或 -1, $r_2 = 1$ 或 -1)

$$\begin{bmatrix} f_{r_1,r_2,1} \\ g_{r_1,r_2,1} \end{bmatrix} = \begin{bmatrix} C_{111} & -S_{111} \\ S_{111} & C_{111} \end{bmatrix} \begin{bmatrix} I_4 & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} I_4^{21} B_{111} & R_1 I_4^{21} B'_{111} \\ I_4^{43} B'_{111} & R_2 I_4^{43} B_{111} \end{bmatrix} Y_{111} \tag{34}$$

$$r = (r_3 - r_1) + (r_3 - r_2) \times (1/2) \tag{35}$$

$$B_{111} = H_2^{2} \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix},\ B'_{111} = H_2^{2} \begin{bmatrix} J'_1 & 0 \\ 0 & J_2 \end{bmatrix} \tag{36}$$

$$y_{111}(p) = y_{111}(n_1 + p_1 N/4, n_2 + p_2 N/4, n_3 + p_3 N/4) \tag{37}$$

3　计算复杂度分析与讨论

设一次蝶形运算需要四次乘法和两次加法.

当 $m = 1$,即 $N = 2q$ 时,由式(2)、(3)计算,算法所需的乘法和加法数分别为:

$$M(2q \times 2q \times 2q) = 8M(q \times q \times q) \tag{38}$$

$$A(2q \times 2q \times 2q) = 8A(q \times q \times q) + 24q^3 \tag{39}$$

当 $m = 2$,即 $N = 4q$ 时,$\cos\alpha$ 或 $\sin\alpha$ 的值为 0 或 ± 1,不需做乘法运算,此时算法需要的乘法和加法数分别为:

$$M(4q \times 4q \times 4q) = M(2q \times 2q \times 2q) + 56M(q \times q \times q) \tag{40}$$

$$A(4q \times 4q \times 4q) = A(2q \times 2q \times 2q) + 56A(q \times q \times q + 360q^3 \tag{41}$$

当 $m > 2$,即 $N > 4q$ 时

(1) 式(7)、(13)、(17)、(20)、(25)、(30)、(34)中矩阵运算需做的乘法和加法数分别为:$112 \times (N/4)^3 - M_t$; $168 \times (N/4)^3 - A_t$,其中:$M_t = 10.5 \times qN^2$, $A_t = 3.5 \times qN^2$ 是当 $\cos\alpha$ 或 $\sin\alpha$ 为特殊值 0, ± 1, $\sqrt{2}/2$ 可以节省的乘法和加法数;

(2) 计算式(2)中 $y_{000}(n_1, n_2, n_3)$ 及式(12)、(16)、(19)、(24)、(29)、(33)、(37)需做的加法数为 $3N^3$;

(3) 式(6)需做加法 $56 \times (N/4)^3$;

综上,当 $N > 4q$ 时,算法所需的乘法和加法数分别为:

$$M(N \times N \times N) = 1.75N^3 - 10.5qN^2 + M(N/2 \times N/2 \times N/2) + 56M(N/4 \times N/4 \times N/4) \tag{42}$$

$$A(N \times N \times N) = 6.5N^3 - 3.5qN^2 + A(N/2 \times N/2 \times N/2) + 56A(N/4 \times N/4 \times N/4) \tag{43}$$

当 $q = 1$ 时,算法的初始值为:

$$M(4 \times 4 \times 4) = M(2 \times 2 \times 2) = 0 \tag{44}$$

$$A(4 \times 4 \times 4) = 384\quad A(2 \times 2 \times 2) = 24 \tag{45}$$

当 $q = 3$ 时,用行列方法计算 $3 \times 3 \times 3$ DHT,所需的乘法和加法数分别为:

$$M(3 \times 3 \times 3) = 90\quad A(3 \times 3 \times 3) = 239 \tag{46}$$

当 $q = 1$ 时,文中算法与目前最优的文献[4,5]中的算法具有相同的算术复杂度;当 $q = 3$ 时,用文中的方法以及将文献[5]中算法补零的方法,计算不同长度序列,每点所需的乘法和加法数如表1所示,文中的算法比较将文献[5]中算法补零的方法,每点约能节省40%～50%的乘法计算和50%～60%的加法计算.本文提出的算法不仅可以直接计算 $N = q \times 2^m$ 的序列,而且即便同样用补零的方法计算某些特定长度的序列值时,也比文献[5]中方法作加法和乘法的次数少.如:计算一个$(254 \times 3) \times (254 \times 3) \times (254 \times 3)$点的 DHT,通过补零,用文中的方法只需计算一个$(256 \times 3) \times (256 \times 3) \times (256 \times 3)$点的 DHT,用文献[5]中的方法需计算一个 $1024 \times 1024 \times 1024$ 点的 DHT 前者需作的乘法和加法数分别为 3,324,138,593 和 16,025,153,805,后者需作的乘法和加法数分别为 6,989,415,030 和 35,929,441,541,前者比后者节省约52%的乘法计算和约55%的加法计算.

4　结论

本文推导出一种适用于长度 $N = q \times 2^m$ (q 为奇整数)的三维 DHT 的分裂基快速算法,较之已有算法,文中算法可适用于更多不同长度的序列,在 $q = 1$ 时,本文算法与目前最优的算法具有相同的算术复杂度,在 $q = 3$ 时,本文算法较文献[5]中算法补零的方法有效降低了计算复杂度.且文中算法实现结构规则,可进行同址计算,采用 Kronecker 乘积和矩阵表示,更易于算法实现.

表1　$q=3$ 时算术复杂度比较

	用文献[5]中算法补零计算			本文提出的算法		
Transform size ($q=3$) ($N\times N\times N$)	Muts/ p	Adds/ p	(M + A) / p	Muts/ p	Adds/ p	(M + A) / p
$(2^5\times3)\times(2^5\times3)\times(2^5\times3)$	8.8343	54.8265	63.6608	5.0885	25.0892	30.1776
$(2^6\times3)\times(2^6\times3)\times(2^6\times3)$	11.0339	62.874	73.9079	5.4617	28.4489	33.9106
$(2^7\times3)\times(2^7\times3)\times(2^7\times3)$	13.2088	71.2236	84.4324	6.8031	31.9818	38.7849
$(2^8\times3)\times(2^8\times3)\times(2^8\times3)$	15.4296	79.317	94.7466	7.3383	35.3768	42.7152
$(2^9\times3)\times(2^9\times3)\times(2^9\times3)$	17.6224	87.6387	105.261	8.5995	38.8993	47.4988
$(2^{10}\times3)\times(2^{10}\times3)\times(2^{10}\times3)$	19.8457	95.7626	115.608	9.2357	42.3137	51.5494

参考文献:

[1] Boussakta S, Alshibami O H, Aziz M Y. Radix-2 ×2 ×2 algorithm for the 3-D discrete Hartley transform[J]. IEEE Trans Signal Process, 2001, 49(12): 3145 - 3156.

[2] Alshibami O, Boussakta S. Fast 3-D decimation-in-frequency algorithm for 3-D Hartley transform[J]. Signal Process, 2002, 82(1): 121 - 126.

[3] Bouguezel S, Ahmad M O, Swamy M N S. An efficient three-dimensional decimation-in-time FHT algorithm based on the radix-2/4 approach[J]. In Proc IEEE Int Symp Signal Process Inf Technol, 2004. 52 - 55.

[4] Bouguezel S, Ahmad M O, Swamy M N S. A split vector-radix algorithm for the 3-D discrete Hartley transform[J]. IEEE Trans Circuits and Systems-I: regular papers, 2006, 53(9): 1966 - 1976.

[5] Bi G. Split-radix algorithm for 2-D discrete Hartley transform[J]. Signal Process, 1997, 63(1): 45 - 53.

[6] Bi G, Chen Y Q. Fast DHT algorithms for length-N = q ×2^m, IEEE Trans Signal Process[J]. 1999, 47(3): 900 - 903.

[7] Granata J, Conner M, Tolimieri R. The tensor product: A mathematical programming language for FFT's and other fast DSP operations[J]. IEEE Signal Process Mag, 1992, 9(1): 40 - 48.

作者简介:

姜龙玉　女, 1982 年生, 甘肃定西人. 硕士研究生, 目前主要的研究方向为 DSP 快速算法.

E-mail: jianglongyu01412 @yahoo.com.cn

舒华忠　男, 1965 年生, 江西省玉山县人. 教授, 博士生导师, 主要研究领域为医学图像处理、模式识别和 DSP 快速算法等.