多维DFT的多维多项式变换与离散W变换算法

钟广军,成礼智,陈火旺
(国防科技大学,湖南长沙 410073)

摘 要: 本文首先通过引进一种序列的重排技术将 $m(m\geq 2)$ 维离散Fourier变换(m-D DFT)转化为一系列的一维广义离散Fourier变换(GDFT)的多重和.然后引入一维离散W变换(DWT)以及多维多项式变换(MD-PT)计算该多重和以减少冗余的算术运算,从而得到了高效的多维DFT算法,该算法与常用的行-列DFT算法相比,乘法仅约为行-列法的 $1/2m$,而加法仅约为行-列法的 $(2m+1)/4m$.对于2维DFT的计算,本文方法同单纯的多项式变换方法相比,乘法与加法分别减少50%与40%左右.另外,本文算法计算结构简单,易于编程实现,通过数值实验验证了本文算法的高效性.

关键词: 离散Fourier变换(DFT); 多维信号处理; 多项式变换(PT); 离散W变换; 快速算法
中图分类号: TN917 文献标识码: A 文章编号: 0372-2112(2001)08-1053-04

Multidimensional Polynomial Transform and Discrete W Transform Algorithms for Multidimensional DFT

ZHONG Guang-jun, CHENG Li-zhi, CHEN Huo-wang
(*National University of Defense Technology, Changsha, Hunan 410073, China*)

Abstract: This paper first presents a conversion of the m-dimensional discrete Fourier transform (MD-DFT) into a multiple sum involving a number of one-dimensional generalized discrete Fourier transforms (1D-GDFT's) by reordering the input data. An one-dimensional discrete W transform (DWT) and multidimensional polynomial transform (MD-PT) are then used to compute the m-D DFT so that the number of arithmetic operations needed is reduced. The number of multiplications and additions required by the proposed algorithm is only $1/2m$ and $(2m+1)/4m$ times that of the usually used row-column DFT algorithm. Besides, compared with the polynomial transform for 2-D DFT developed by Nussbaumer, the multiplications and additions needed by the proposed algorithm are reduced by 50% and 40%, respectively. The numerical experiments show that the proposed algorithm is not only simple in computational structure and but also highly efficient.

Key words: discrete Fourier transform (DFT); multidimensional signal processing; polynomial transform (PT); discrete W transform (DWT); fast algorithm

1 引言

离散Fourier变换(DFT)一直是数字信号处理与图像处理中的重要工具,正是快速Fourier变换(FFT)的出现极大地促进了上述学科的快速发展.对于在图像处理中有较广泛应用的多维DFT,例如二维与三维DFT[1,2],人们通常使用行-列法进行计算.由于多维离散变换的行-列算法的运算量是相当大的,因此,研究高维DFT的快速算法是必要的.对于2-维DFT,1981年Nussbaumer[3,4]曾提出用一维多项式变换(1D-PT)与一维DFT相结合来计算二维DFT,该方法与行-列DFT算法相比,运算量尤其是乘法有较大幅度的减少.但由于文[3]中算法是通过一维DFT,一维PT以及递归计算相结合来得到的,因此算法运算结构比较复杂,这给程序优化与编程实现带来一定难度.因而该算法很难推广到更高维DFT的计算.另一方面,在信号处理应用中我们有时候有必要计算更高维的DFT[2],基于此,本文首先提出对高维DFT的输入重新排序使之变为一系列的一维广义离散Fourier变换(1D-GDFT)的多重和,然后利用Z Wang[5]提出的第二类离散W变换(DWT-II)算法计算(GDFT),从而大大减少了算法所需的乘法次数.为了减少多重和的冗余加法,引进多维多项式变换(MD-PT)去计算该多重和,通过应用MD-PT,所需的加法次数也有较大幅度的减少.根据上述讨论,本文在第二节讨论了一类GDFT的离散W变换(DWT)算法;第三节研究了多维DFT到多个一维GDFT的转换;第四节则引入多维多项式变换(MD-PT)以减少加法次数;第五节为计算复杂性讨论以及与已有算法在理论与计算机上运算时间上的比较;最后给出了结论.

收稿日期: 2000-07-20; 修回日期: 2000-11-05

2 一维 GDFT 的 DWT 算法

为了讨论问题方便, 首先引入多维离散 Fourier 变换(MD-DFT), 一维广义离散 Fourier 变换(1-D GDFT) 以及一维离散 W 变换(1-D DWT) 的定义.

定义 1 序列 $x_{n_1,n_2,\cdots,n_m}$ 的 m-维离散 Fourier 变换(m-D DFT) 定义为[1,2]

$$X_{k_1,k_2,\cdots,k_m}=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}\cdots\sum_{n_m=0}^{N_m-1}x_{n_1,n_2,\cdots,n_m}\times\exp\left\{-\sum_{i=1}^{m}\frac{2n_ik_i\pi}{N_i}j\right\},$$

$$k_i=0,1,\cdots,N_i-1,\ i=1,2,\cdots,m,\ j=\sqrt{-1} \quad (1)$$

为简单计, 本文只考虑信号处理中的常见参数 $N_m=2^t$, $N_i=\frac{N_m}{2^{l_i}}$, $i=1,2,\cdots,m-1$, $m\ge 2$, $l_i\ge 0$ 的情形.

定义 2 序列 x_n 的一维广义离散 Fourier 变换(1-D GDFT) 定义为[1,10]

$$X_k(\alpha,\beta)=\sum_{n=0}^{N-1}x_n\exp\left\{-\frac{2(n+\alpha)(k+\beta)\pi j}{N}\right\},\quad k=0,1,\cdots,N-1 \quad (2)$$

其中, α 与 β 分别称之为时间与频率参数. 通常取 $\alpha,\beta=0$ 或 $\frac{1}{2}$[1,10]. 在本文中, 只考虑$(\alpha,\beta)=(\frac{1}{2},0)$的情况, 并设 $XG_k=X_k(\frac{1}{2},0)$.

定义 3 实序列 x_n 的一维离散 W 变换(1-D DWT) 定义为[5]

$$W_k=\sum_{n=0}^{N-1}x_n\operatorname{cas}\frac{(2n+1)k\pi}{N},\ k=0,1,\cdots,N-1 \quad (3)$$

其中 $\operatorname{cas}(z)=\sin(z)+\cos(z)$. 对于长 $N=2^t$ 的 1-D DWT, 如果我们采用 Z Wang[6,7] 所提出的快速算法, 则所需的乘法 M_W 与加法 A_W 次数分别满足

$$M_W=\frac{1}{2}Nt-\frac{1}{2}N,\ A_W=\frac{3}{2}Nt-\frac{3}{2}N,\quad N\ge 4 \quad (4)$$

下面讨论如何用一维 DWT 来计算一维 GDFT 的问题.

因为一个复序列的一维 GDFT(1-D GDFT) 很容易通过两个实序列的一维 GDFTs(1-D GDFTs) 与 $2N$ 次实加来得到, 因此为简单计我们只讨论 1-D GDFT 的输入序列 x_n 为实序列的情形.

对于 $k=0,1,\cdots,N-1$, 利用三角函数的性质, 从式(3) 得到

$$W_{N-k}=\sum_{n=0}^{N-1}x_n\operatorname{cas}\frac{(2n+1)(N-k)\pi}{N}=-\sum_{n=0}^{N-1}x_n\operatorname{cas}\frac{-(2n+1)k\pi}{N} \quad (5)$$

分别用式(3) 与式(5) 直接相加减, 不难得到

$$W_k-W_{N-k}=2\sum_{n=0}^{N-1}x_n\cos\frac{2(n+1/2)k\pi}{N},$$

$$W_k+W_{N-k}=2\sum_{n=0}^{N-1}x_n\sin\frac{2(n+1/2)k\pi}{N},$$

将上式与式(2) 比较, 可得到由式(2) 定义且$(\alpha,\beta)=(\frac{1}{2},0)$时的 1-D GDFT 实部与虚部分别满足

$$\mathrm{Re}XG_k=\frac{1}{2}(W_k-W_{N-k}),\ \mathrm{Im}XG_k=\frac{1}{2}(W_k+W_{N-k}) \quad (6)$$

利用式(3)之定义, 看到 W_k 为斜周期序列, 即 $W_N=-W_0$. 式(6)表明一个长 N 的一维实序列 GDFT(1-D GDFT) 可以通过一个同样长度的实序列 DWT(1-D DWT) 以及式(6) 中的 N 次实加得到(注意: 在式(6) 中可用 $\mathrm{Re}XG_{N-k}=-\mathrm{Re}XG_k$ 与 $\mathrm{Im}XG_{N-k}=\mathrm{Im}XG_k$ 来减少加法次数). 因而一个长 N 的复序列之 1-D GDFT 可以通过两个长度 N 的实序列 1-D DWT 以及 $2N$ 次实加得到.

3 m-维 DFT 到一维 GDFT 的转换

现在讨论如何将一个 m-维 DFT 转换成一系列一维 GDFTs 的问题. 作转换的目的是为了减少多维变换所需的冗余运算, 其原理非常简单. 事实上, 在定义 1 的 m-维 DFT 表达式中, 当 n_i, $k_i=0,1,\cdots,N_i-1$; $i=1,2,\cdots,m$ 时, $\sum_{i=1}^{m}\frac{2n_ik_i\pi}{N_i}$所能取得的不同值个数远超过 N_m, 而另一方面, 式(1) 定义的 m-D DFT 变换核 $K=\exp\{-\sum_{i=1}^{m}\frac{2n_ik_i\pi}{N_i}j\}=\exp\{-\frac{\sum_{i=1}^{m}2^{l_i+1}k_in_i\pi}{N_m}j\}$只能取到 N_m 个不同值, 这一事实表明: 当用常见的行-列法计算多维 DFT 时, 会出现同一个变换核的值在不同的求和层次反复与不同的变换系数相乘的情形, 作转换的目的即相当于将含有同一变换核值的系数合并到一起以减少冗余乘法, 即作最简单的变换: $bc_1+bc_2+\cdots+bc_N=b(c_1+c_2\cdots+c_N)$以减少冗余乘法. 本文所需要的序列转换基于下面的命题:

命题 1 设 $r(q_i)\equiv(2n_m+1)q_i \bmod N_i$, $i=1,2,\cdots,m-1$, 集合 $A=\{(n_1,n_2,\cdots,n_m):0\le n_i\le N_i-1;1\le i\le m\}$ 与 $B=\{(r(q_1),r(q_2),\cdots,r(q_{m-1}),n_m):0\le q_i\le N_i-1;1\le i\le m-1;0\le n_m\le N_m-1\}$, 则 $A=B$.

证明: 很显然, $B\subset A$, 因此要证 $A=B$, 只需证明对于不同的序列 $q_1,q_2,\cdots,q_{m-1}$, 集合 B 中的元素不同即可. 或者反过来, 设$(r(q_1),r(q_2),\cdots,r(q_{m-1}),n_m)=(r(p_1),r(p_2),\cdots,r(p_{m-1}),n'_m)$, 则

$$r(q_i)=r(p_i),\ i=1,2,\cdots,m-1;\ n_m=n'_m$$

根据 $r(q_i)$的定义, 我们得到$(2n_m+1)q_i\equiv(2n_m+1)p_i \bmod N_i$, 因此$(2n_m+1)(q_i-p_i)\equiv 0 \bmod N_i$, $i=1,2,\cdots,m-1$. 由于 $2n_m+1$ 与 N_i 为互素的, 我们得到 $p_i\equiv q_i \bmod N_i$, 即有 $q_i=p_i$. 上述讨论表明 $A=B$.

由命题 1, 按式(1)定义的 m-维 DFT 可以等价地表示为

$$X_{k_1,k_2,\cdots,k_m}=\sum_{q_1=0}^{N_1-1}\sum_{q_2=0}^{N_2-1}\cdots\sum_{q_{m-1}=0}^{N_{m-1}-1}\sum_{n_m=0}^{N_m-1}\cdot x_{r(q_1),r(q_2),\cdots,r(q_{m-1}),n_m}\times\exp\left\{-\left(\sum_{i=1}^{m-1}\frac{2r(q_i)k_i}{N_i}+\frac{2n_mk_m}{N_m}\right)\pi j\right\} \quad (7)$$

因为 $2r(q_i)\equiv 2(2n_m+1)q_i \bmod 2N_i$ 以及函数 exp 的周期为 $2\pi j$, 因此 $\exp\{-\sum_{i=1}^{m-1}\frac{2r(q_i)k_i}{N_i}\pi j\}=\exp\{-\sum_{i=1}^{m-1}$

$\frac{2(2n_m+1)q_ik_i}{N_i}\pi j\} = \exp\{-\sum_{i=1}^{m-1}\frac{2(2n_m+1)q_ik_i2^{l_i}}{N_m}\pi j\}$. 将该等式代入式(7)之右端有

$$X_{k_1,k_2,\dots,k_m} = \sum_{q_1=0}^{N_1-1}\sum_{q_2=0}^{N_2-1}\cdots\sum_{q_{m-1}=0}^{N_{m-1}-1} \cdot [\sum_{n_m=0}^{N_m-1}x_{r(q_1),r(q_2),\dots,r(q_{m-1}),n_m} \times \exp\{-\frac{\sum_{i=1}^{m-1}2(2n_m+1)q_ik_i2^{l_i}+2n_mk_m}{N_m}\pi j\}] \quad (8)$$

定义 $N_1N_2\cdots N_{m-1}$个长 N_m 的一维广义 DFT(1-D GDFT) 如下:

$$Y_{q_1,q_2,\dots,q_{m-1}}(p) = \sum_{n_m=0}^{N_m-1}x_{r(q_1),r(q_2),\dots,r(q_{m-1}),n_m} \times \exp\{-\frac{(2n_m+1)p}{N_m}\pi j\},$$

其中, $p=0,1,\dots,N_m-1$. 由于 $0\le q_i\le N_i-1$, $i=1,\dots,m-1$, 根据第二节的讨论, 看到 1-D GDFT $Y_{q_1,q_2,\dots,q_{m-1}}(p)$可以通过 $N_1N_2\cdots N_{m-1}$个序列 $x_{r(q_1),\dots,r(q_{m-1}),n_m}$的 1-D DWT 来计算. 另外, 由于式(8)中和式含括号部分满足

$$\begin{aligned}&\sum_{n_m=0}^{N_m-1}x_{r(q_1),r(q_2),\dots,r(q_{m-1}),n_m} \cdot \exp\{-\frac{\sum_{i=1}^{m-1}2(2n_m+1)q_ik_i2^{l_i}+2n_mk_m}{N_m}\pi j\}\\ &= \exp\{\frac{k_m\pi j}{N_m}\}\sum_{n_m=0}^{N_m-1}x_{r(q_1),\dots,r(q_{m-1}),n_m} \cdot \exp\{-\frac{(2n_m+1)(\sum_{i=1}^{m-1}2^{l_i+1}k_iq_i+k_m)}{N_m}\pi j\}\\ &= \exp\{\frac{k_m\pi j}{N_m}\}Y_{q_1,q_2,\dots,q_{m-1}}(\sum_{i=1}^{m-1}2^{l_i+1}k_iq_i+k_m). \end{aligned} \quad (9)$$

将式(9)代入(8)右端又有

$$X_{k_1,k_2,\dots,k_m} = \exp\{\frac{k_m\pi j}{N_m}\}\sum_{q_1=0}^{N_1-1}\cdots\sum_{q_{m-1}=0}^{N_{m-1}-1}Y_{q_1,\dots,q_{m-1}}(\sum_{i=1}^{m-1}2^{l_i+1}k_iq_i+k_m). \quad (10)$$

注意到 $Y_{q_1,q_2,\dots,q_{m-1}}(p+kN_m) = (-1)^kY_{q_1,\dots,q_{m-1}}(p)$, 只要 $N_1N_2\cdots N_{m-1}$个一维 GDFT's $Y_{q_1,\dots,q_{m-1}}(p)$, $p=0,1\dots,N_m-1$ 被求出, 则希望得到的 m-维 DFT 输出 MD-DFT $X_{k_1,k_2,\dots,k_m}$可以通过式(10)的加法得到. 如果算法到此结束, 后面的计算复杂性分析表明, 与常用的基-2 行-列 DFT 算法相比, 乘法次数已经大幅度减少. 但同时也要看到, 式(10)中的多重求和其加法运算量是相当大的, 这一点对于较大的 N_i 尤其如此. 下一节引入多维多项式变换(MD-PT)的目的正是为了降低式(10)中所需的加法次数.

4　式(10)的 MD-PT 算法

现在研究式(10)多重和的$(m-1)$维多项式变换算法. 为此, 设 $X^{(1)}_{k_1,\dots,k_m} = \sum_{q_1=0}^{N_1-1}\cdots\sum_{q_{m-1}=0}^{N_{m-1}-1}Y_{q_1,\dots,q_{m-1}}(\sum_{i=1}^{m-1}2^{l_i+1}k_iq_i+k_m)$, 由式(10)有

$$X_{k_1,\dots,k_m} = \exp\left\{\frac{k_m\pi j}{N_m}\right\}X^{(1)}_{k_1,\dots,k_m} \quad (11)$$

并设 $X^{(1)}_{k_1,\dots,k_{m-1}}(z) = \sum_{k_m=0}^{N_m-1}X^{(1)}_{k_1,\dots,k_m}z^{k_m}$, 因为 $Y_{q_1,\dots,q_{m-1}}(p+k_m)z^{k_m} \bmod (z^{N_m}+1)$对于变量 k_m 周期为N_m, 并利用式(10)与(11), 可以得到

$$X^{(1)}_{k_1,k_2,\dots,k_{m-1}}(z) \equiv \sum_{q_1=0}^{N_1-1}\sum_{q_2=0}^{N_2-1}\cdots\sum_{q_{m-1}=0}^{N_{m-1}-1}Q_{q_1,\dots,q_{m-1}}(z) \times z^{2q_1k_1}z_2^{q_2k_2}\cdots z_{m-1}^{q_{m-1}k_{m-1}} \bmod (z^{N_m}+1), \quad (12)$$

其中 $z_i \equiv z^{-2^{l_i+1}} \bmod (z^{N_m}+1)$以及 $Q_{q_1,\dots,q_{m-1}}(z) = \sum_{k_m=0}^{N_m-1}Y_{q_1,q_2,\dots,q_{m-1}}(k_m)z^{k_m}$.

利用文[10]中建立起来的多维多项式变换理论, 可以直接验证式(12)构成一个$(m-1)$维的多项式变换, 因而可以用多项式序列 $Q_{q_1,q_2,\dots,q_{m-1}}(z)$的$(m-1)$维-PT 来计算[10]. 从而得到多项式序列 $X^{(1)}_{k_1,k_2,\dots,k_{m-1}}(z)$, 另一方面, 该多项式序列的系数为 $X^{(1)}_{k_1,\dots,k_m}$. 因此当多项式序列 $X^{(1)}_{k_1,k_2,\dots,k_{m-1}}(z)$通过多维多项式变换求出后, 由式(11)作 N_m 次复数乘法即可得到所求的 MD-DFT $X_{k_1,k_2,\dots,k_m}$.

5　计算复杂性讨论

从第四节的算法推导可以看出, 算法主要由一系列 1-D GDFT 以及一个$(m-1)$-维多项式变换(MD-PT)组成. 为了估计总的运算次数, 取在数字与图像处理中经常使用的长度, $N_1=N_2=N_3=\dots=N_m$ 且$N=N_m^m$. 当用本文提出的方法计算 1-D GDFT 时, 由于 $N_1N_2\cdots N_{m-1}$个长为 N_m 的复序列的 1D-GDFT's 需要 $N\log_2N_m-N$ 次实乘与 $3N\log_2N_m-N$ 次实加(见式(4)), 而一个$(m-1)$-维多项式变换(MD-PT)仅需 $2N\log_2N_m^{m-1}$次实加[10]. 另外, 如果将一次复乘视为 4 个实乘, 2 个实加的话, 则本文算法所需的实乘 RM_1 与实加 RA_1 分别为

$$RM_1 = N\log_2N_m+3N,\ RA_1 = N[(2m+1)\log_2N_m+1] \quad (13)$$

而利用行-列 DFT 方法的实乘 RM_2 与实加 RA_2 分别满足

$$RM_2 = 2mN\log_2N_m-2mN,\ RA_2 = 4mN\log_2N_m-N \quad (14)$$

由式(11)与(12)不难得到 $RM_1 \approx \frac{1}{2m}RM_2$ 以及 $RA_1 \approx \frac{2m+1}{4m}RA_2$. 由于在引言中曾经指出, Nussbaumer 在 1981 年曾提出用一维多项式变换(1D-PT)与一维 DFT 相结合计算二维 DFT, 因此还对本文算法与文[3, 4]中算法的运算量进行了比较, 此时取 $m=2$, $N_1=N_2=\sqrt{N}$, 则文[3, 4]中算法所需实乘 RM_3 与实加 RA_3 分别满足

$$RM_3 = 2N\log_2N_1-2N+N_1,\ RM_3 = 8N\log_2N_1-N+2N_1 \quad (15)$$

式(15)的结论也可参见文[10]. 比较式(13)与(15), 不难得到本文算法与文[3, 4]的方法相比其乘法减少 50% 而加法减少 40% 左右.

表 1 中列出了用 Visual C++ 6.0 for Windows 98 软件平台在 P-Ⅱ 450 微机上对于本文算法, 行-列法以及文[3]中多项式变换方法计算二维 DFT 的时间比较. 一般说来, 本文方法所需时间约为行-列法的$\frac{1}{4}$, 约为文[3]中方法的$\frac{1}{2}$, 这说明本文方法的理论结果与计算机上运行时间一致性.

表 1 实序列 2 维-DFT 运算时间比较(单位:秒)

长度 $N_1 \times N_2$	本文算法时间	行一列法时间	文[3]中算法时间
$2^7 \times 2^8$	0.09093	0.26789	0.18315
$2^8 \times 2^9$	0.18045	0.59940	0.30001
$2^9 \times 2^9$	0.78813	2.60083	1.41250
$2^9 \times 2^{10}$	1.60461	5.61614	2.31252
$2^{10} \times 2^{11}$	7.31871	27.07923	13.41771
$2^{12} \times 2^{12}$	66.89200	260.87881	128.94125

6 结论

本文提出了用多维多项式变换(MD-PT)计算多维 DFT 的快速算法.与常用的行-列 DFT 算法相比,实乘次数约为行-列法的 $1/2m$,而实加次数约为行-列法的$(2m+1)/4m$.特别地,对于二维 DFT 的计算,本文算法所需的实乘与实加次数与已有的 PT 算法相比分别减少 50% 与 40% 左右.另外,本文算法结构简单,易于程序实现.

参考文献:

[1] V Britanak, K R Rao. The fast generalized discrete Fourier transforms: A unified approach to the discrete sinusoidal transform computation [J]. Signal Processing, 1999, 79: 135-150.

[2] R W Cox, R Q Tong. Two- and three-dimensional image rotation using the FFT [J]. IEEE Trans. Image Processing, 1999, 8(9): 1023-1031.

[3] H J Nussbaumer. New polynomial transform algorithms for multidimensional DFT's and convolutions [J]. IEEE Trans. 1981, ASSP-29(1): 105-115.

[4] H J Nussbaumer. Fast Fourier Transform and Convolutions [M]. 2nd ed., Springer-Verlag, Berlin, 1982.

[5] Z Wang. Fast algorithm for the discrete W transform and for the discrete Fourier transform [J]. IEEE Trans. 1984, ASSP-32(4): 803-816.

[6] 王中德.快速 W 变换——算法与程序[J].中国科学(A 辑),1988,5:549-560.

[7] Z Wang. The discrete W transform [J]. Appl. Math. Comput., 1985, 16: 19-48.

[8] 曾泳泓,蒋增荣.二维离散 W 变换的多项式变换算法[J].电子学报,1997,25(8):63-66.

[9] Y Zeng, X Li. Multidimensional polynomial transform algorithm for multidimensional discrete W transform [J]. IEEE Trans. 1999, SP-47(7): 2050-2052.

[10] 蒋增荣,等.快速算法[M].长沙:国防科技大学出版社,1994.

作者简介:

钟广军 男,1974 年生于湖南省澧县.现在国防科技大学计算机学院攻读博士学位.目前感兴趣的研究领域有图像处理,视频压缩,形式化软件开发方法.