离散傅里叶变换的算术傅里叶变换算法

张宪超[1],武继刚[1],蒋增荣[2],陈国良[1]
(1. 中国科技大学计算机科学与技术系,合肥 230027;2. 国防科技大学系统工程与数学系,长沙 410073)

摘 要: 离散傅里叶变换(DFT)在数字信号处理等许多领域中起着重要作用. 本文采用一种新的傅里叶分析技术—算术傅里叶变换(AFT)来计算 DFT. 这种算法的乘法计算量仅为 $O(N)$;算法的计算过程简单,公式一致,克服了任意长度 DFT 传统快速算法(FFT)程序复杂、子进程多等缺点;算法易于并行,尤其适合 VLSI 设计;对于含较大素因子,特别是素数长度的 DFT,其速度比传统的 FFT 方法快;算法为任意长度 DFT 的快速计算开辟了新的思路和途径.

关键词: 离散傅里叶变换 (DFT); 算术傅里叶变换 (AFT); 快速傅里叶变换 (FFT)
中图分类号: TN917 **文献标识码**: A **文章编号**: 0372-2112 (2000) 05-0105-03

An Algorithm for Computing DFT Using Arithmetic Fourier Transform

ZHANG Xian-chao[1], WU Ji-gang[1], JIANG Zeng-rong[2], CHEN Guo-liang[1]
(1. *Dept. of Computer Science & Technology, Univ. of Science & Technology of China, Hefei* 230027, *China*;
2. *Dept. of System Engineering & Mathematics, National Univ. of Defense Technology, Changsha* 410073, *China*)

Abstract: The Discrete Fourier Transform (DFT) plays an important role in digital signal processing and many other fields. In this paper, a new Fourier analysis technique called the arithmetic Fourier transform (AFT) is used to compute DFT. This algorithm needs only O(N) multiplications. The process of the algorithm is simple and it has a unified formula, which overcomes the disadvantage of the traditional fast method that has a complex program containing too many subroutines. The algorithm can be easily performed in parallel, especially suitable for VLSI designing. For a DFT at a length that contains big prime factors, especially for a DFT at a prime length, it is faster than the traditional FFT method. The algorithm opens up a new approach for the fast computation of DFT.

Key words: discrete Fourier transform (DFT); arithmetic Fourier transform (AFT); fast Fourier transform (FFT)

1 引言

离散傅里叶变换(DFT)在数字信号处理等许多领域中起着重要作用. 但 DFT 的计算量很大(N 点 DFT 需 $O(N^2)$ 乘法和加法). 因此,DFT 的快速计算问题非常重要. 1965 年,Cooley 和 Tukey 开创了快速傅里叶变换(FFT)方法,使 N 点 DFT 的计算量从 $O(N^2)$ 降到 $O(N\log N)$,开辟了 DFT 的快速计算时代. 但 FFT 的计算仍较复杂,且对不同长度的 DFT 其计算公式不一致,致使任意长 DFT 的 FFT 程序非常复杂,包含大量子进程.

1988 年,Tufts 和 Sadasiv[1]提出了一种用莫比乌斯反演公式(Möbius inversion formula)计算连续函数的傅立叶系数的方法并命名为算术傅立叶变换(AFT). AFT 有许多良好的性质:其乘法量仅为 $O(N)$;算法简单,并行性好,尤其适合 VLSI 设计. 因此很快得到广泛关注,并在数字图像处理等领域得到应用. AFT 已成为继 FFT 后一种新的重要的傅立叶分析技术[2~5].

根据 DFT 和连续函数的傅立叶系数的关系,可以用 AFT 计算 DFT. 这种方法保持了 AFT 的良好性质,且具有公式一致性. 大量实验表明,同直接计算相比,AFT 方法可以将 DFT 的计算时间减少 90%,对含较大素因子,特别是其长度本身为素数的 DFT,它的速度比传统的 FFT 快. 从而它为 DFT 快速计算开辟了新的途径.

2 算术傅立叶变换

本文采用文[3]中的算法. 设 $A(t)$ 为周期为 T 的函数,它的傅立叶级数只含有限项,即:

$$A(t) = a_0 + \sum_{n=1}^{N} a_n \cos 2\pi f_0 t + \sum_{n=1}^{N} b_n \sin 2\pi f_0 t \tag{1}$$

其中:$f_0 = 1/T$, $a_0 = \dfrac{1}{T}\displaystyle\int_0^T A(t)\,dt$.

令:$$B(2n,\alpha) = \frac{1}{2n}\sum_{m=0}^{2n-1}(-1)^m A\left(\frac{m}{2n}T + \alpha T\right),\ -1 < \alpha < 1 \tag{2}$$

则傅立叶系数 a_n 和 b_n 可以由下列公式计算:

$$a_n = \sum_{l=1,3,5,\cdots}^{[N/n]} u(l)\,B(2nl, 0)$$

$$b_n = \sum_{l=1,3,5,\cdots}^{[N/n]} u(l)\,(-1)^{(l-1)/2} B\left(2n, \frac{1}{4nl}\right),\ n = 1, \cdots, N \tag{3}$$

收稿日期:1999-01-28;修订日期:1999-05-10
基金项目:国防“九五”预研项目资助课题;国家 863 计划项目资助课题

其中：
$$\mu(l)=\begin{cases}1, & l=1\\ (-1)^r, & l=p_1p_2\cdots p_r\\ 0, & \exists\, p \text{ 使 } p^2 \mid l\end{cases}$$
为莫比乌斯(Möbious)函数.

这就是 AFT,其计算量为:加法:$N^2+[N/2]+[N/3]+\cdots+1-2N$; 乘法:$2N$.

AFT 需要函数大量的不均匀样本点,而在实际应用中,若计算函数前 N 个傅立叶系数,根据奈奎斯特(Nyquist)抽样定律,只需在函数的一个周期内均匀抽取 $2N$ 个样本点. 这时可以用零次插值解决样本不一致问题. 文献[2、3]已作了详细的分析,本文不再重复.

3 DFT 的 AFT 算法

3.1 DFT 的定义及性质

定义 1 设 X_k 为一长度为 N 的序列,它的 DFT 定义为:

$$Y_n=\sum_{k=0}^{N-1}X_k w^{nk},\ n=0,1,\cdots,N-1;\ w=e^{-i2\pi/N} \tag{4}$$

性质 1 用记号 $X_k \Longleftrightarrow Y_n$ 表示序列 Y_n 为序列 X_k 的 DFT, $G_k \Longleftrightarrow H_n$,则:

$$pX_k+qG_k \Longleftrightarrow pY_n+qH_n \tag{5}$$

因此,一个复序列的 DFT 可以用两个实序列的 DFT 计算. 故本文只讨论实序列 DFT 的计算问题.

性质 2 设 X_k 为一实序列, $X_k \Longleftrightarrow Y_n$,则:

$\mathrm{Re}\,Y_n=\mathrm{Re}\,Y_{N-n}$, $\mathrm{Im}\,Y_n=-\mathrm{Im}\,Y_{N-n}$ ($\mathrm{Re}\,Y_n$ 和 $\mathrm{Im}\,Y_n$ 分别代表 Y_n 的实部和虚部) (6)

因此,对 N 点实序列 DFT,只需计算: $\mathrm{Re}\,Y_n$ 和 $\mathrm{Im}\,Y_n$ ($n=0,\cdots,\lceil N/2\rceil$).

3.2 DFT 的 AFT 算法

离散序列的 DFT 和连续函数的傅立叶系数有着密切的联系. 事实上,若序列 X_k 是一段区间 $[0,T]$ 上的函数 $A(t)$ 经过离散化后得到的,再设 $A(t)$ 的傅立叶级数只含前 $N/2$ 项,即:

$$A(t)=a_0+\sum_{n=1}^{\lceil N/2\rceil-1}a_n\cos 2\pi f_0 t+\sum_{n=1}^{\lceil N/2\rceil-1}b_n\sin 2\pi f_0 t \tag{7}$$

则 DFT Y_n 和傅立叶系数的关系为:

$$\begin{cases}\mathrm{Re}\,Y_n=\lceil N/2\rceil a_n/2\\ \mathrm{Im}\,Y_n=\lceil N/2\rceil b_n/2\end{cases},\ n=0,\cdots,\lceil N/2\rceil \tag{8}$$

式(7)中函数代表的是一种截频信号. 对一般函数,式(8)中的“=”要改为“≈”[7]. 因此,序列 X_k 的 DFT 可以通过函数 $A(t)$ 的傅里叶系数计算.

对于一般给定序列 X_k,注意到在任意一个区间上,经过离散后能得到序列 X_k 的函数有无穷多个. 对所有这些插值函数,公式(8)都近似地满足(仅式(7)中的函数精确地满足式(8))[7]. AFT 的零次插值实现实质上就是用这些插值函数中的零次插值函数代替原来的函数进行计算的. 而从 AFT 的零次插值实现方法可知,用 AFT 计算傅里叶系数,实际上参与计算的只是函数经离散化后得到的序列,而不必知道函数本身. 因此,我们可以任取一个区间,在这个区间上,把序列 X_k 作为插值样本点序列,再利用 AFT 的零次插值实现方法,计算(8)中的“傅里叶系数”,再通过式(8),就可以计算出序列的 DFT.

算法描述如下(采用[0,1]区间):

```
for n = 1 to ⌈N/2⌉
  for m = 0 to 2n - 1
    B(2n,0) := B(2n,0) + (-1)^m X[Nm/2n + 0.5]
    B(2n,1/4n) :=
    B(2n,1/4n) + (-1)^m X[Nm/2n + N/4n + 0.5]
  endfor
  B(2n,0) := B(2n,0)/2n
  B(2n,1/4n) := B(2n,1/4n)/2n
endfor
for j = 0 to N - 1
  a0 := a0 + X(j)/N
for k = 1 to ⌈N/2⌉
  for n = 1 to[⌈N/2⌉/n] by 2
    an := an + μ(k) B(2nk,0)
    bn := bn + μ(k) (-1)^((K-1)/2) B(2nk,1/4nk)
  endfor
  Re Yn := ⌈N/2⌉ an/2
  Re Y(N-n) := Re Yn
  Im Yn := ⌈N/2⌉ an/2
  Im Y(N-n) := - Im Yn
endfor
endfor
```

图 1 DFT 的 AFT 算法程序

AFT 方法的误差主要是由零次插值引起的,大量实验表明,同 FFT 相比,其误差是可以接受的(部分实验结果见附录).

4 算法的性能

4.1 算法的程序

DFT 的 AFT 算法具有公式一致性,且公式简单,因此算法的程序也很简单(图 1).

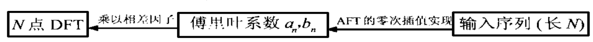


图 2 DFT 的 AFT 算法进程示意

为便于比较,不妨看一下 FFT 的流程.

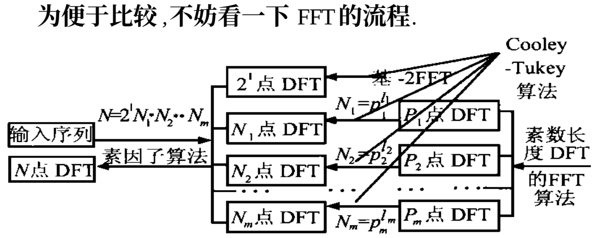


图 3 FFT 算法进程示意

可以看出,FFT 的程序中包含大量子进程,且这些子程序都较复杂. 其中素数长度 DFT 的 FFT 算法程序尤其复杂. 因此,任意长 DFT 的 FFT 算法其程序是非常复杂的.

4.2 算法的计算效率

AFT 方法把 DFT 的乘法计算量从 $O(N^2)$ 降到 $O(N)$,它也是 DFT 的一种快速算法. 大量实验表明,AFT 方法把 DFT 的

计算时间减少 90 %. 当 DFT 的长度 N 为 2 的幂时 ,FFT 比 AFT 方法快. 对一般长度的 DFT,当 N 含较大素因子时 ,AFT 方法比 FFT 快;当 N 的因子都较小时 ,AFT 方法不如 FFT 快. 当 DFT 长度 N 本身为一较大素数时 ,AFT 方法比 FFT 快. 附录中给出部分实验结果以便比较.

特别指出,对素数长度 DFT,FFT 的计算过程非常复杂,很难在实际中应用. 而 AFT 方法算法简单,提供了较好的素数长度 DFT 快速算法. 表 1 是两种算法计算效率较详细的比较.

表 1

长度	521	911	971	1483	2417
FFT 效率	67.30 %	68.03 %	72.50 %	71.23 %	76.22 %
AFT 方法效率	91.39 %	91.78 %	91.63 %	91.81 %	91.83 %

4.3　算法的并行性

AFT 具有良好的并行性,尤其适合 VLSI 设计,已有许多 VLSI 设计方案被提出,并在数字图像处理等领域得到应用. DFT 的 AFT 算法继承了 AFT 优点,同样具有良好的并行性.

5　结论和展望

本文采用算术傅里叶变换 (AFT) 计算 DFT. 这种方法把 AFT 的各种优点引入 DFT 的计算中来,开辟了 DFT 快速计算的新途径. 把 AFT 方法同 FFT 结合起来,还可以进一步提高 DFT 的计算速度.

参考文献

[1] D. W. Tufts and G. Sadasiv. The arithmetic Fourier transform. IEEE ASSP Mag ,Jan. 1988 :13～17

[2] I. S. Reed ,D. W. Tufts ,Xiao Yu ,T. K. Troung ,M. T. Shih and X. Yin. Fourier analysis and signal processing by use of Mobius inversion formular. IEEE Trans. Acoust. Speech Speech Processing ,Mar ,1990 ,38 (3) :458～470

[3] I. S. Reed ,Ming Tang Shih ,T. K. Truong ,R. Hendon and D. W. Tufts. A VLSI architecture for simplified arithmetic fourier transform algorithm. IEEE Trans. Signal Processing ,May ,1993 ,40(5) :1122～1132

[4] H. Park and V. K. Prasanna. Modular VLSI architectures for computing the arithmetic fourier transform. IEEE. Signal Processing ,June ,1993 , 41(6) :2236～2246

[5] Lovine. F. P ,Tantaratanas. Some alternate realizations of the arithmetic Fourier transform. Conference on Signal ,system and Computers ,1993 , (Cat ,93 ,CH3312-6) :310～314

[6] 蒋增荣 ,曾泳泓 ,余品能. 快速算法. 长沙:国防科技大学出版社 ,1993

[7] E. O. 布赖姆. 快速傅立叶变换. 上海:上海科学技术出版社 ,1976

附录:较详细的实验结果(机型:586 微机,主频:166MHz 单位:秒)

2 的幂长度

长度	AFT 方法	基-2 FFT	直接算法
256	0.00516	0.00240	0.11
512	0.01860	0.00440	0.44
1024	0.07580	0.01100	1.81
2048	0.29830	0.02450	7.20

素数长度

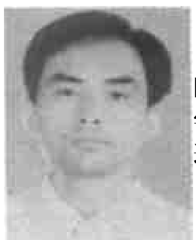
长度	AFT 方法	FFT	直接算法
521	0.0379	0.1439	0.44
971	0.1340	0.4400	1.60
1483	0.3103	1.0904	3.79
2417	0.8206	2.3899	10.75

任意长度

长度	因子分解	AFT 方法	FFT	直接算法
1346	2 *637	0.27	0.44	3.14
2986	2 *1483	1.26	2.14	14.82
3579	3 *1193	1.81	1.92	22.16
4637	4637	3.08	21.42	37.47
5574	2 *3 *929	4.45	2.47	52.29
6436	4 *1609	5.94	3.57	72.62
7893	3 *3 *877	8.96	1.92	105.49

最大相对误差

长　度		AFT 方法	FFT
1024	实部	2.1939×10^{-2}	2.3328×10^{-2}
	虚部	2.1938×10^{-2}	9.9342×10^{-2}
2048	实部	4.2212×10^{-3}	1.1967×10^{-2}
	虚部	6.1257×10^{-3}	4.9385×10^{-2}
4096	实部	2.3697×10^{-3}	6.0592×10^{-3}
	虚部	2.0422×10^{-3}	2.4615×10^{-3}



张宪超　1971 年生. 1994 年、1998 年分别获国防科技大学学士、硕士学位. 现在中国科技大学攻读博士学位. 主要研究方向为信号处理的快速、并行计算等.

武继刚　1963 年生. 烟台大学副教授,现在中国科技大学攻读博士学位. 主要研究方向为算法设计和分析等.