第 26 卷第 6 期 电子与信息学报 Vol.26No.6
2004 年 6 月 Journal of Electronics & Information Technology June 2004

算术傅里叶变换的实际实现方法 [1]

张宪超 徐 云 陈国良

(国家高性能计算中心，合肥 230027)
(中国科技大学计算机科学与技术系 合肥 230027)

摘 要：算术傅里叶变换 (AFT) 结构简单，乘法量少，具有广阔的应用．但在 AFT 在具体实现中往往需要过采样来满足实际应用中的精度要求．过采样问题是 AFT 的一个重要缺陷且限制了它的应用范围．该文利用 AFT 的线性插值实现技术精度很高的特点，在线性插值实现技术和过采样技术的基础上提出了一个新的实现策略，可以达到接近过采样的精度．从而解决了 AFT 的过采样问题．

关键词：傅里叶分析，算术傅里叶变换，采样率，过采样
中图分类号：TN911.72 **文献标识码**：A **文章编号**：1009-5896(2004)06-0935-05

Practical Implementation of the Arithmetic Fourier Transform

Zhang Xian-chao Xu Yun Chen Guo-liang

(*National High Performance Computing Center at Hefei, Hefei* 230027, *China*)

(*Dept of Comp. and Sci., Univ. of Sci. and Tech. of China, Hefei* 230027, *China*)

Abstract The Arithmetic Fourier Transform (AFT) is widely used because of its simple computational structure and little multiplications. But over-sampling is often needed for the implementation of AFT to meet the accuracy requirements in real applications and this is one of the main drawbacks of AFT and limits its application. In this paper, with the fact that linear interpolation implementation can gain very high accuracy, a new implementation is presented based on linear interpolation and over-sampling. This implementation can get accuracy close to that by over-sampling, thus the over-sampling problem of AFT is overcome.

Key words Fourier analysis, Arithmetic Fourier Transform(AFT), Sampling rate, Over-sampling

1 引言

算术傅里叶变换 (AFT) 是一种利用数论中莫比乌斯变换计算函数的傅里叶系数的方法，它是由 H. Bruns 于 1903 年首次提出的 [1]，但没有引起重视． 1988 年， Dufts 和 Sadasiv 又独立地发现了类似的方法并把它命名为算术傅里叶变换 (AFT)[2]，由于 AFT 有许多优点，因此迅速引起关注，关于它的研究越来越多 [3−15]，并在许多领域里得到应用． AFT 已发展成为继快速傅里叶变换 (FFT) 之后的另一重要的傅里叶分析技术．

AFT 算法本身需要对信号函数进行大量不均匀的采样，而实际的系统往往只产生均匀采样．因此在实际应用中，需要利用某种插值方法来实现 AFT，但插值实现不可避免地会带来误差．研究表明，线性插值产生的误差非常小，具有很高的精度 [3−5,10] ．但线性插值需要大量的附加运算，其乘法运算量比 AFT 算法本身高出一个量级．因此 AFT 的线性插值实现不具有实用价值．

在实际应用中广泛采用的 AFT 实现方法是零次插值，但零次插值会带来较大的误差 [3,4,10]，为减少这种误差，需要对信号函数进行过采样，即采样频率高于奈奎斯特频率。文献 [5] 研究了

[1] 2003-01-07 收到， 2003-05-27 改回

这个问题并指出在许多应用中，AFT 的采样频率是奈奎斯特频率的若干倍. 过采样问题是 AFT 的主要缺点之一并在一定程度上限制了 AFT 的应用 [4,5] . 而在某些应用中，例如利用 AFT 计算离散傅里叶变换 (DFT)[7] 或离散余弦变换 (DCT)[8], 由于给定的输入是离散的值而不是连续函数，过采样方法是无法实现的.

本文利用线性插值精度很高的特点，提出了在线性插值的基础上进行过采样的方法. 这种方法可以达到接近过采样的精度. 它是通过简单计算来实现的，从而避免了线性插值的大量计算问题和零次插值的误差较大或需要过采样的问题.

本文的方法需要一定的附加计算，但同 AFT 算法本身的计算量相比较小，不影响 AFT 的优点.

2 算术傅里叶变换

目前已有几种算术傅里叶变换算法，其中文献 [4] 给出的简化 Bruns 是最常用的. 本文讨论此算法.

定义 1 (莫比乌斯函数) 设 n 为自然数，它的莫比乌斯函数 $\mu(n)$ 定义为

$$\mu(n)=\begin{cases}1, & n=1\\ (-1)^r, & n=p_1\cdots p_r(p_i\text{为相异素数})\\ 0, & \exists\text{素数}p\text{使}p^2|n\end{cases} \tag{1}$$

定义 2 (算术傅里叶变换) 设 $A(t)$ 是周期为 T 的函数，它的傅里叶级数只含有限项，即

$$A(t)=a_0+\sum_{n=1}^{N}a_n\cos 2\pi nf_0t+\sum_{n=1}^{N}b_n\sin 2\pi nf_0t \tag{2}$$

其中 $f_0=1/T$, $a_0=\int_0^T A(t)\mathrm{d}t$.

定义如下交替平均值:

$$B(2n,\alpha)=\frac{1}{2n}\sum_{m=0}^{2n-1}(-1)^m A(mT/(2n)+\alpha T),\qquad -1<\alpha<1 \tag{3}$$

则式 (2) 中的傅里叶系数 a_n 和 b_n 可由下列公式计算:

$$a_n=\sum_{l=1,3,5,\cdots}^{\lceil N/n\rceil}\mu(l)B(2nl,0) \tag{4}$$

$$b_n=\sum_{l=1,3,5,\cdots}^{\lceil N/n\rceil}\mu(l)(-1)^{(l-1)/2}B(2nl,1/(4nl)) \tag{5}$$

其中 $n=1,\cdots,N$. 称这种计算傅里叶系数的方法为算术傅里叶变换 (AFT)[4] .

AFT 需要 $O(N)$ 实数乘法和 $O(N^2)$ 实数加法. 它的结构简单，非常适合 VLSI 设计 [2,4], 具有广泛的应用.

3 AFT 的线性插值实现和零次插值实现

从 AFT 的计算公式可以看出，它需要 $O(N^2)$ 的不均匀的样本点，而实际系统一般只产生均匀的样本点. 因此在实际应用中需要用某种插值来实现 AFT .

3.1 线性插值实现

AFT 的线性插值实现是按奈奎斯特频率对信号函数采样，然后用采样点的线性插值函数代替原函数进行计算 (图 1) .

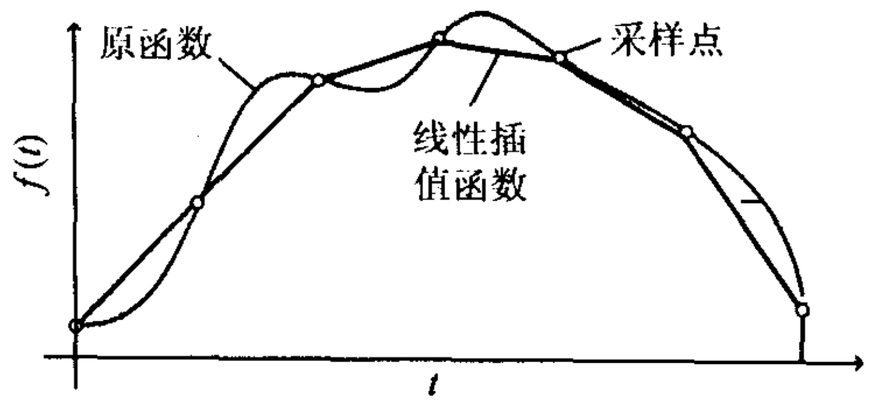

图 1　AFT 的线性插值实现

理论和实验研究都表明，线性插值实现的误差很小，精度很高 [3−5,10]．为便于与其他方法比较，我们在表 1 中给出一组实验数据．它计算了一个有限长度傅里叶级数的前 1024 个傅里叶系数 (以下实验采用相同的输入)．我们分别用 E_a 和 E_b 表示 a_n, b_n 的最大误差．

表 1　AFT 线性插值实现的误差

采样数		2048
误差	E_a	5.171 E-06
	E_b	2.481 E-06

3.2 零次插值实现

AFT 的零次插值实现的思想是用最近的采样点的函数值代替未被采样的、但 AFT 公式中需要的点的函数值．由于零次插值的误差较大，在实际应用中往往需要通过过采样的方法 (即采样频率高于奈奎斯特频率) 来减少误差，提高精度．如果某种过采样的频率恰好是奈奎斯特频率的 n 倍，我们称这种采样是 n 倍过采样．图 2 演示了零次插值和过采样技术．

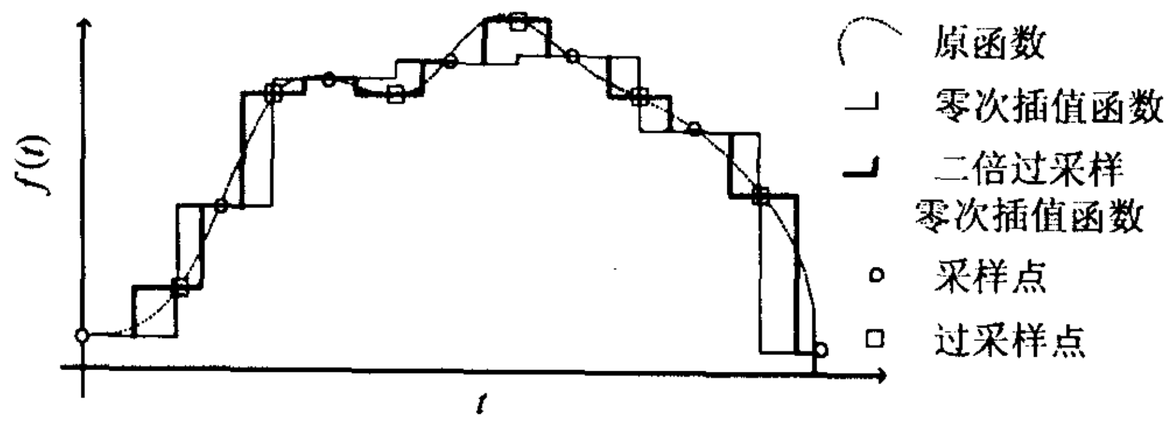

图 2　AFT 零次插值实现和二倍过采样零次插值实现

从图上可以看出，过采样的零次插值函数更加逼近原函数．通过过采样可以明显地提高 AFT 实现的精度 (表 2)．

表 2　AFT 零次插值实现的误差

采样数		2048(1 倍)	4096(2 倍)	8192(4 倍)
误差	E_a	2.740 E-03	6.290 E-04	3.573 E-04
	E_b	6.234 E-03	2.513 E-04	1.452E-04

4 对线性插值函数进行过采样

我们希望避免对函数进行过采样但提高 AFT 的实现精度．注意到线性插值实现的精度是非常高的，即用 AFT 公式计算出的线性插值函数的傅里叶系数和原函数的傅里叶系数十分接近．因此线性插值函数的傅里叶系数的好的近似也是原函数的傅里叶系数的好的近似．

现假设我们希望计算一个函数 $f(t)$ 的傅里叶系数，记它的线性插值函数为 $L(t)$，记线性插值实现对原函数的误差为 E_{lf}，$L(t)$ 上的一个零次插值实现的误差为 E_{0l}，这个零次插值实现对

原函数 $f(t)$ 的误差为 E_{0f}，则 $|E_{0f}| \le |E_{lf}| + |E_{0l}|$。在线性插值函数上使用过采样可以减少 E_{0l}。因此可以通过在一个信号函数的线性插值函数上进行过采样来提高 AFT 的实现精度。

注意到线性函数的一个简单性质：

性质 1 设 $f(x)$ 是一条线段，它的起点是 $(a, f(a))$，终点是 $(b, f(b))$，则此线段的中点是 $((a+b)/2, (f(a)+f(b))/2)$。

因此对一个线性插值函数，可以简单算出它的每个线段中点的函数值，从而得到 2 倍过采样 (图 3)。重复这样的过程，可以得到 4 倍，8 倍，等等的线性插值函数的过采样。表 3 给出了通过过采样技术得到的线性插值函数的傅里叶系数的误差。

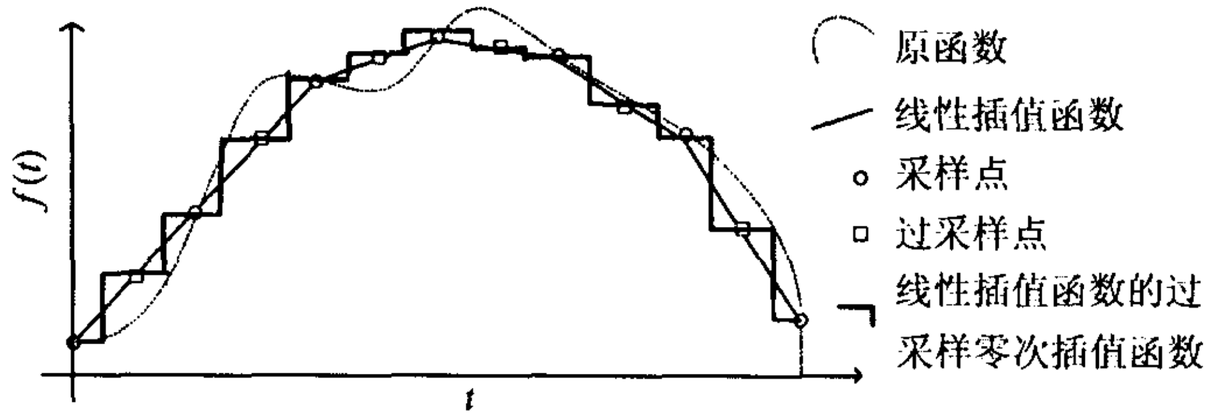

图 3 在线性插值函数上过采样

表 3 过采样技术对线性插值函数的 AFT 实现精度的提高

采样数		2048(1 倍)	4096(2 倍)	8192(4 倍)
误差	E_a	2.740 E-03	6.917 E-04	3.569 E-04
	E_b	6.234 E-03	2.529 E-04	1.454E-04

如果采用这样计算出的近似线性插值函数的傅里叶系数作为原信号函数的傅里叶系数，则误差见表 4。

表 4 线性插值 + 过采样实现 AFT 的误差

采样数		2048(1 倍)	4096(2 倍)	8192(4 倍)
误差	E_a	2.740 E-03	6.931 E-04	3.518 E-04
	E_b	6.234 E-03	2.518 E-04	1.474E-04

比较表 2 和表 4 中的数据，可以看出在原函数上进行过采样和在线性插值函数上进行过采样的效果是比较接近的。

现在分析这种方法的计算量。设在一个有 N 个节点的线性插值函数上进行 $m = 2^k$ 倍的过采样。则需 $(2^{k-1} + 2^{k-2} + \cdots + 1)N = (2^k - 1)N = (m-1)N$ 次乘法 (除 2) 和加法计算。由于在实际应用中需要的过采样倍数远远小于采样点数 [4,5]，即 $m << N$。因此与 AFT 本身的 $O(N)$ 次乘法和 $O(N^2)$ 次加法相比，这个附加计算量是可以接受的。

5 用于计算离散傅里叶变换 (DFT)

傅里叶系数和 DFT 有着紧密的联系，可以通过傅里叶系数计算 DFT。由于 DFT 的输入是离散的，而 AFT 需要大量的函数样本点。因此若用 AFT 计算 DFT，需要把 DFT 的离散输入序列构造成一个函数。文献 [7] 给出了一种构造方法，即在 $[0,1]$ 区间上构造这个离散序列的零次插值函数，用这个函数实现 AFT。由于在零次插值函数进行过采样显然是没有任何意义的，因此无法提高文献 [7] 中用 AFT 计算 DFT 的精度。

通过前面的讨论，用文献 [7] 的方法，我们可以这样计算 DFT，即在 $[0,1]$ 区间上构造离散输入序列的线性插值函数 (实际上并不构造，只是计算线段中点的函数值)，在这个函数的基础上进行过采样来实现 AFT，从而计算 DFT。

表 5 给出了用本文方法 (8 倍过采样) 和文献 [7] 方法计算 DFT 的误差比较。其中 E_r 和 E_i 分别表示实部和虚部的最大相对误差。

表 5　本文方法与文献 [7] 方法计算 DFT 的误差比较

长度	实虚部	本文方法误差	文献 [7] 方法误差
1024	实部误差 (E_r)	4.3076 E-03	2.1939 E-02
	虚部误差 (E_i)	2.8046 E-03	2.1938 E-02
2048	实部误差 (E_r)	2.1094 E-03	4.2212 E-03
	虚部误差 (E_i)	1.4483 E-03	6.1257 E-03
4096	实部误差 (E_r)	1.1707 E-03	2.3697 E-03
	虚部误差 (E_i)	1.1765 E-03	2.0422 E-03

可以用类似的方法改进 DCT 的算术傅里叶变换算法 [8] 的计算精度.

6 结论

过采样问题是 AFT 的主要缺点之一，本文给出了通过简单计算实现达到接近过采样技术的精度的 AFT 实现方法。由于这种方法是通过简单计算来实现的，因此保持了 AFT 的优点。对解决实际应用问题来说，这种方法的附加计算量和 AFT 本身相比可以接受.

参　考　文　献

[1] Bruns H. Grundlinien des Wissenschaftlichnen Rechnens[M]. Leipzig, Personal Publication, 1903.

[2] Tufts D W, Sadasiv G. The arithmetic Fourier transform[J]. *IEEE ASSP Mag*, 1988, 5(1): 13–17.

[3] Reed I S, Tufts D W, Xiao Yu, *et al*.. Fourier analysis and signal processing by use of Mobius inversion formular[J]. *IEEE Trans. on Acoust, Speech, Signal Processing*, 1990, 38(3): 458–470.

[4] Reed I S, Shih M T, Troung T K, *et al*.. A VLSI architecture for simplified arithmetic Fourier transform algorithms[J]. *IEEE Trans. on Acoust, Speech, Signal Processing*, 1993, 40(5): 1122–1132.

[5] Lovine F P, Tantaratanas S. Some alternate realizations of the arithmetic Fourier transform. [C]. Proceedings of the Twenty-Seventh Annual Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California, 1993: 310–314.

[6] Ge Xi-Jin, Chen Nan-Xian, Chen Zhao-Dou. Efficient algorithm for 2-D arithmetic Fourier transform[J]. *IEEE Trans. on Signal Processing*, 1997, 45(8): 2136–2140.

[7] 张宪超，武继刚，蒋增荣，陈国良. 离散傅里叶变换的算术傅里叶变换算法 [J]. 电子学报，2000, 28(5): 105–107.

[8] 张宪超，李宁，陈国良. 离散余弦变换的改进的算术傅里叶变换算法 [J]. 电子学报，2000, 28(9): 88–90.

[9] 张宪超，陈国良，李宁. 改进的算术傅里叶变换算法 [J]. 电子学报，2001, 29(3): 329–331.

[10] Wigley N Jullien. A sampling reduction for the arithmetic Fourier transform[C]. Proc, 32nd Midwest Symposium on Circuits and Systems, Champaign, IL, 1990: 841–844.

[11] Knckaert L. A generalized Mobius transform, arithmetic Fourier transform, and primitive roots[J]. *IEEE Trans. on Signal Processing*, 1996, 44(5): 1307–1310.

[12] Schiff J, Walker W. The arithmetic Fourier transform. Analysis, geometry and groups: A Riemann legacy volume, Hadronic Press Collect. Orig. Artic., Palm Harbor, FL, Hadronic Press, 1993: 613–625.

[13] Walker W. The arithmetic Fourier transform and real neural networks: summability by primes[J]. J. Math. Anal. Appl. 1995, 190: 211–219.

[14] Walker W. A summability method for the arithmetic Fourier transform[J]. BIT, 1994, 34(2): 304–309.

[15] Tufts D W, Chen H. Iterative realization of the arithmetic Fouier transform. *IEEE Trans. Signal Processing*, 1993, 41(1): 152–161.

张宪超：　男，1971 年生，博士，主要研究方向为数字信号处理的快速、并行算法、并行分布式计算、网络优化等.

徐　云：　男，1960 年生，副教授，博士，主要研究方向为算法分析与设计、并行算法等.

陈国良：　男，1938 年生，教授，博士生导师，国家高性能中心 (合肥) 主任，主要研究方向为算法分析与设计、并行分布式计算、网络并行计算等.