一类特殊 DFT 的快速算法 *

孙世新 * * 郑文学
（电子科技大学计算机系 成都 610054）

【摘要】 一般的 DFT 算法都假定输入和输出序列长度相等，实际的情况并非总是如此。鉴于此，文中讨论了输入和输出序列长度不相等的这类 DFT 的快速计算方法，其结果比 Skinner 的剪枝法和 Sorensen，Burrus 的变换分解法更简洁高效。

关 键 词 离散傅里叶变换； 快速傅里叶变换； 广义傅里叶变换； 快速计算

中图分类号 TP399； O174

在数字信号处理领域，DFT（离散傅里叶变换）是一个广泛应用的数学工具，因此寻求各种高效的 FFT（快速傅里叶变换）算法具有非常重要的意义。一般的 DFT 算法都假定输入和输出序列长度相等，实际的情况并非总是如此。对于这类 DFT 已经有一些快速算法，如剪枝（Pruning method）以及变换分解（Transform decomposition）。这两种方法对于运算量的节省不很显著，且算法比较复杂。本文将把 GFFT（广义快速傅里叶变换）方法应用到这类输入输出序列不等的特殊 DFT 的求解上来，采用的方法是把一段长的 DFT 化为若干段较短的 GFT。我们将会看到采用这种方法会带来更高的速度，并且算法思想清晰、简洁，易于程序实现。在某些情况下比剪枝法减少约 40％的实数乘法运算量，10％的实数加法运算量；比变换分解减少约 30％的实数乘法运算量，10％的实数加法运算量。

关于剪枝及变换分解的方法可参见文献[1～7]。本文介绍了 GFT 的定义，推导出 SR GFFT（分裂基 GFFT）算法，并分别讨论输入序列仅有部分非零点和输出序列仅需部分点的快速计算，还与其他方法进行了对比。

1 GFT 和 SR GFFT

GFT 的定义是

$$X(k)=\sum_{n=0}^{N-1}x[n]\,W_N^{(k+\alpha)(n+b)}\qquad k=0,1,\cdots,N-1\quad W_N=e[-j(2\pi/N)]\tag{1}$$

其中 a 是频率参数；b 是时间参数。实际上，DFT 是在 $a=0$，$b=0$ 时 GFT 的一个特例。我们将 SR FFT（分裂基 FFT）的思想应用于 GFT 中，以获得高效的求解算法。对于 $b=0$ 时，从式(1)可以推导出 DIT SR GFFT（按时间抽选的 SR GFFT）。

$$X(k)=\sum_{n=0}^{N-1}x[n]\,W_N^{(k+a)n}=\sum_{r=0}^{N/2-1}x[2r]\,W_{N/2}^{(k+a)r}+W_N^{(k+a)}\sum_{r=0}^{N/4-1}x[4r+1]\,W_{N/4}^{(k+a)r}+$$
$$W_N^{3(k+a)}\sum_{r=0}^{N/4-1}x[4r+3]\,W_{N/4}^{(k+a)r}\qquad k=0,1,\cdots,N-1$$

1997 年 1 月 3 日收稿，1997 年 5 月 6 日修改定稿
* 电子部预研基金资助项目
* * 男 54 岁 大学 教授

设 $$G(k)=\sum_{r=0}^{N/2-1}x[2r]\ W_{N/2}^{(k+a)r} \qquad k=0,1,\cdots,\frac{N}{2}-1$$

$$H_1(k)=\sum_{r=0}^{N/4-1}x[4r+1]\ W_{N/4}^{(k+a)r},\ H_2(k)=\sum_{r=0}^{N/4-1}x[4r+3]\ W_{N/4}^{(k+a)r} \qquad k=0,1,\cdots,\frac{N}{4}-1$$

则

$$\begin{aligned}
X(k)&=G(k)+(W_N^{(k+a)}H_1(k)+W_N^{3(k+a)}H_2(k))\\
X(\frac{N}{4}+k)&=G(\frac{N}{4}+k)-j(W_N^{(k+a)}H_1(k)-W_N^{3(k+a)}H_2(k))\\
X(\frac{N}{2}+k)&=G(k)-(W_N^{(k+a)}H_1(k)+W_N^{3(k+a)}H_2(k)) \qquad 0\leqslant k\leqslant\frac{N}{4}-1\\
X(\frac{3N}{4}+k)&=G(\frac{N}{4}+k)+j(W_N^{(k+a)}H_1(k)-W_N^{3(k+a)}H_2(k))
\end{aligned}$$

同理，对于 $a=0$ 时从式(1)可以推导出 DIF SR GFFT(按频率抽选的 SR GFFT)。为节省篇幅在此略去。

从以上推导过程中可以看出，SR GFFT 与 SR FFT 不同之处仅在于使用不同的旋转因子(Twiddle factor)因此，SR GFFT 的运算量与 SR FFT 相同(不考虑旋转因子产生过程)。实数乘法、加法次数分别是

$$\frac{4}{3}N\log_2 N-\frac{8}{9}N+(-1)^{\log_2 N}\frac{8}{9}$$

$$\frac{8}{3}N\log_2 N-\frac{4}{9}N+(-1)^{\log_2 N}\frac{4}{9}$$

2 输入序列仅有部分非零点的计算

设输入序列 $x[n]$ 长 N 点，其中只有 $x[0]$，$x[1]$，…，$x[L-1]$ 这一段为非零点。这里限定 N 是 2 的幂(以下同样)。

设 $P=2^{\lceil\log_2 L\rceil}$，$Q=\frac{N}{P}$，DFT

$$X(k)=\sum_{n=0}^{N-1}x[n]\ W_N^{kn} \qquad k=0,1,\cdots,N-1 \tag{2}$$

设 $k=k_1Q+k_2$，$k_1=0,1,\cdots,P-1$，$k_2=0,1,\cdots,Q-1$，则

$$X(k_1Q+k_2)=\sum_{n=0}^{P-1}x[n]\ W_N^{(k_1Q+k_2)n}=\sum_{n=0}^{P-1}x[n]\ W_P^{(k_1+\frac{k_2}{Q})n}$$

这就把问题转换为求 Q 个 P 点的 GFT。根据第 1 节的结果，实数乘法次数和实数加法次数分别是

$$\frac{4}{3}N\log_2 P-\frac{8}{9}N+(-1)^{\log_2 P}\frac{8}{9}\frac{N}{P}$$

$$\frac{8}{3}N\log_2 P-\frac{4}{9}N+(-1)^{\log_2 P}\frac{4}{9}\frac{N}{P}$$

以上的讨论具有一个限制，即输入序列的非零点从 $x[0]$ 开始。下面再讨论更一般的情况。

设 $x[n]$ 的非零点段是 $x[s]$，$x[s+1]$，…，$x[s+L-1]$。那么式(2)就成为

$$X(k_1Q+k_2)=\sum_{n=s}^{s+P-1}x[n]\ W_N^{(k_1Q+k_2)n}=\sum_{n=0}^{P-1}x[s+n]\ W_N^{(k_1Q+k_2)(s+n)}=$$

$$W_P^{(k_1+\frac{k_2}{Q})s}\sum_{n=0}^{P-1}x[s+n]\ W_P^{(k_1+\frac{k_2}{Q})n}$$

这相当于求 Q 个 P 点的 GFT 以及 N 个复数乘法。那么采用 SR GFFT 来做,需要的实数乘法和实数加法次数分别是

$$\frac{4}{3}N\log_2 P+\frac{28}{9}N+(-1)^{\log_2 P}\ \frac{8}{9}\ \frac{N}{P}$$

$$\frac{8}{3}N\log_2 P+\frac{14}{9}N+(-1)^{\log_2 P}\ \frac{4}{9}\ \frac{N}{P}$$

3　输出序列仅有部分点的计算

设在输出序列当中,仅有 $X(0), X(1), \cdots, X(L-1)$ 是我们所需要的,同样设 $P=2^{\lceil \log_2 L\rceil}$, $Q=\frac{N}{P}$, 设 $n=n_1Q+n_2$, $n_1=0, 1, \cdots, P-1$, $n_2=0, 1, \cdots, Q-1$, 则

$$X(k)=\sum_{n_2=0}^{Q-1}\sum_{n_1=0}^{P-1}x[n_1Q+n_2]\ W_N^{k(n_1Q+n_2)}=$$
$$\sum_{n_2=0}^{Q-1}\sum_{n_1=0}^{P-1}x[n_1Q+n_2]\ W_P^{k(n_1+\frac{n_2}{Q})}\qquad k=0, 1, \cdots P-1 \tag{3}$$

设

$$X_{n_2}(k)=\sum_{n_1=0}^{P-1}x[n_1Q+n_2]\ W_P^{k(n_1+\frac{n_2}{Q})},\ k=0, 1, \cdots, P-1$$

那么式(3)可以写成

$$X(k)=\sum_{n_2=0}^{Q-1}X_{n_2}(k),\ k=0, 1, \cdots, P-1$$

这就把问题转换为求 Q 个 P 点的 GFT, 以及 $L(Q-1)$ 次数复数加法。

如果把 SR FFT 应用于式(3)的求解,那么,总的实数乘法次数和实数加法次数分别是

$$\frac{4}{3}N\log_2 P-\frac{8}{9}N+(-1)^{\log_2 P}\ \frac{8}{9}\ \frac{N}{P}$$

$$\frac{8}{3}N\log_2 P-\frac{4}{9}N+(-1)^{\log_2 P}\ \frac{4}{9}\ \frac{N}{P}+2L(\frac{N}{P}-1)$$

以上的讨论也附加了一个限制,那就是所需要的输出序列从 $X(0)$ 开始。下面再讨论更一般的情况。

设需要的输出序列是 $X(s), X(s+1), \cdots, X(s+L-1)$。那么式(2)就可以写成

$$X(k+s)=\sum_{n_2=0}^{Q-1}\sum_{n_1=0}^{P-1}x[n_1Q+n_2]\ W_N^{(k+s)(n_1Q+n_2)}=\sum_{n_2=0}^{Q-1}\sum_{n_1=0}^{P-1}x[n_1Q+n_2]\ W_P^{(k+s)(n_1+\frac{n_2}{Q})}=$$
$$\sum_{n_2=0}^{Q-1}\sum_{n_1=0}^{P-1}x[n_1Q+n_2]\ W_P^{s(n_1+\frac{n_2}{Q})}W_P^{k(n_1+\frac{n_2}{Q})},\ k=0, 1, \cdots, P-1 \tag{4}$$

设

$$X_{n_2}(k+s)=\sum_{n_1=0}^{P-1}\left(x[n_1Q+n_2]\ W_P^{s(n_1+\frac{n_2}{Q})}\right)W_P^{k(n_1+\frac{n_2}{Q})},\ k=0, 1, \cdots, P-1$$

那么式(4)可写成

$$X(k+s)=\sum_{n_2=0}^{Q-1}=\sum_{n_2=0}^{Q-1}X_{n_2}(k+s)\quad k=0,1,\cdots,P-1$$

这相当于求 N 个复数乘法和 Q 个 P 点的 GFT 以及 $L(Q-1)$次复数加法。那么采用 SR GFFT，需要的实数乘法和实数加法次数分别是

$$\frac{4}{3}N\log_2 P+\frac{28}{9}N+(-1)^{\log_2 P}\frac{8}{9}\frac{N}{P}$$

$$\frac{8}{3}N\log_2 P+\frac{14}{9}N+(-1)^{\log_2 P}\frac{4}{9}\frac{N}{P}+2L\left(\frac{N}{P}-1\right)$$

4 与其他方法的对比

本文讨论的方法都不考虑特殊旋转因子对运算量的节省。另外，一次复数乘法都通过 4 次实数乘法和 2 次实数加法来完成。至于考虑了特殊旋转因子对运算量的节省，以及一次复数乘法通过 3 次实数乘法和 3 次实数加法来完成的情况，读者可自行推导。

Skinner 的剪枝法[1,2] 需要的实数乘法次数是：$2N\log_2 L$，对应于输入和输出两种情况，实数加法次数分别是：$3N\log_2 L$ 和 $3N\log_2 L+2N-2L$。

变换分解法的实数乘法次数是：$\frac{4}{3}N\log_2 P-\frac{8}{9}N+(-1)^{\log_2 P}\frac{8}{9}\frac{N}{P}+4L\left(\frac{N}{P}-1\right)$，对应于输入和输出两种情况，实数加法次数分别是

$$\frac{8}{3}N\log_2 P-\frac{4}{9}N+(-1)^{\log_2 P}\frac{4}{9}\frac{N}{P}+2\left(L\frac{N}{P}-1\right)$$

$$\frac{8}{3}N\log_2 P-\frac{4}{9}N+(-1)^{\log_2 P}\frac{4}{9}\frac{N}{P}+4\left(L\frac{N}{P}-1\right)$$

图 1 示出了当 $N=512$ 点时，各种方法的运算量随非零输入点数 L 变化的图像。图 2 示出了当 $N=512$ 点时，各种方法的运算量随输出点数 L 变化的图像。图中上部曲线表示实数加法次数，下部表示实数乘法次数。

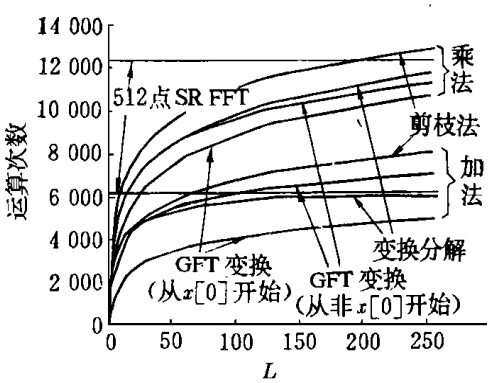

图 1 L 个非零输入点时的运算量($N=512$)

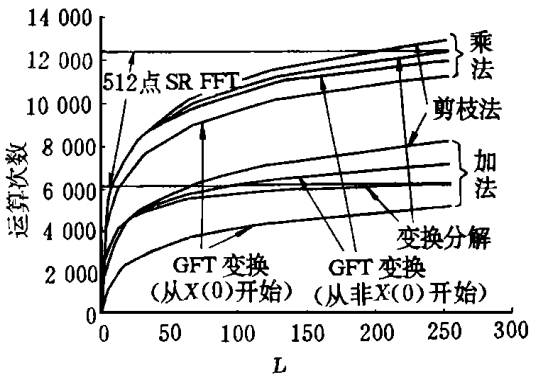

图 2 L 个输出点时的运算量($N=512$)

从图 1 和图 2 可以看出，这种 GFT 变换法(暂且如此命名)具有较少的运算量，特别是在输入或输出序列从零点开始的时候是上述几种方法当中最优的方法。

5 结 论

本文给出的 GFT 变换方法把一段长的 DFT 化为若干段较短的GFT，使得运算得以减少。算

法思想清晰,能充分利用已有的 FFT 成果,并且其运算量能达到较少或最少,是计算部分输入或输出点的这类特殊 DFT 的比较好的快速算法。同其他两种方法一样,当 $L \leqslant N/2$ 时用这种方法较好,而当 $L > N/2$ 时直接采用 SR FFT 的方法会更好。另外,这些快速算法在 N, L 比较小的情况下更出色。如果实际的计算机乘法速度比加法速度慢得多,那么可以把上面介绍的方法当中,1 次复数乘法用 4 次实数乘法和 2 次实数加法改为 3 次实数乘法和 3 次实数加法。

另外需要注意的是,这种 GFT 变换法需要更多的运算量来产生旋转因子,使得实际运算量变大。不过,对于大批量的运算可以先将所有的旋转因子计算好。这样一来就很好地解决了这个问题。

参 考 文 献

1　Markel J D. FFT pruning. IEEE Trans on Audio Electroacoust, 1971, AU-19(4): 305 ~ 311

2　Skinner D P. Pruning the decimation in-time FFT algorithm. IEEE Trans on ASSP, 1976, ASSP-24(2): 193 ~ 194

3　Sreenivas T V, Rao P V S. High resolution narrow-band spectra by FFT pruning. IEEE Trans on ASSP, 1980, ASSP-28(2): 254 ~ 257

4　Nagai K. Pruning the decimation-in-time FFT algorithm with frequency shift. IEEE Trans on ASSP, 1986, ASSP-34(4): 1 008 ~ 1 010

5　Wild R de. Method for partial spectrum computation. Proc Inst Elec Eng, 1987, 134(7): 659 ~ 666

6　Sorensen H V, Burrus C S. A new efficient algorithm for computing a few DFT points. In Proc IEEE 1988 Int Symp Circuits Syst, 1988: 1 915 ~ 1 918

7　Sorensen H V, Burrus C S. Efficient computation of the DFT with only a subset of input or output points. IEEE Trans on Signal Porcessing, 1993, 41(3): 1 184 ~ 1 200

Fast Algorithm of DFT with Only A Subset of Input or Output Points Using GFFT

Sun Shixin　　Zheng Wenxue

(Dept. of Computer Science, UEST of China　Chengdu　610054)

Abstract　Most of FFT algorithms are designed for the general situation in which the length of input and output sequences are equal. In special situations, it should be studied carefully for practical applications. This paper provides a new algorithm using GFFT which is more efficient than pruning method, transform decomposition and so on.

Key words　discrete fourier transform;　fast fourier transform;　generalized fourier transform;　fast computation

编辑　徐安玉