

## 一种稀疏度自适应的稀疏傅里叶变换算法

刘 仲, 李立春, 李慧启

(信息工程大学 信息工程学院, 郑州 450001)

**摘 要:** 稀疏快速傅里叶变换需要信号以傅氏域的稀疏度为先验信息, 但稀疏度通常是未知的, 在一定程度上限制了算法的应用。为此, 提出一种新的稀疏傅里叶变换算法。在下采样域进行能量检测, 得到稀疏度的初始值, 通过增大下采样维度提高稀疏度估计的准确性, 从而近似估计稀疏度, 设定阈值剔除冗余信息从而得到较好效果。实验结果表明, 当信号长度大于  $2^{19}$  或稀疏度小于 900 时, 该算法性能优于西方快速傅里叶变换, 且具有较强的鲁棒性。

**关键词:** 快速傅里叶变换; 稀疏表示; 稀疏度自适应; 运行时间; 降维

**中文引用格式:** 刘 仲, 李立春, 李慧启. 一种稀疏度自适应的稀疏傅里叶变换算法 [J]. 计算机工程, 2018, 44(2): 141-146.

**英文引用格式:** LIU Zhong, LI Lichun, LI Huiqi. A Sparsity Adaptive Sparse Fourier Transform Algorithm [J]. Computer Engineering, 2018, 44(2): 141-146.

## A Sparsity Adaptive Sparse Fourier Transform Algorithm

LIU Zhong, LI Lichun, LI Huiqi

(Institute of Information System Engineering, Information Engineering University, Zhengzhou 450001, China)

**[Abstract]** Sparse Fast Fourier Transform (sFFT) requires the signal to have the sparsity of the Fourier domain as prior information, but the sparsity is usually unknown, which limits the application of the algorithm to a certain extent. therefore, a new sparse Fourier transform algorithm is proposed. The energy is detected in the downsampling domain to get the initial value of the sparsity. The accuracy of the sparsity estimation is increased by increasing the downsampling dimension so as to estimate approximatively the sparsity, and the threshold is set to eliminate the redundant information to obtain the better result. Experimental results show that the performance of the algorithm is superior to Faster Fourier Transform in the West (FFTW) when the signal sizes is greater than  $2^{19}$  or the sparsity is less than 900, and the algorithm has strong robustness.

**[Key words]** Fast Fourier Transform (FFT); sparse representation; sparsity adaptive; runtime; dimension reduction

**DOI:** 10.3969/j.issn.1000-3428.2018.02.025

### 0 概述

在信号处理领域, 离散傅里叶变换 (Discrete Fourier Transform, DFT) 是最基本的算法之一。而快速傅里叶变换 (Fast Fourier Transform, FFT) 是最快的 DFT 算法之一, 其算法复杂度只有  $O(n \log_a n)$ 。目前, 计算 DFT 的算法复杂度为  $\Theta(n)$ , 均与信号长度成正比。在图像处理<sup>[1]</sup>、宽带频谱感知<sup>[2-4]</sup>、机器学习理论<sup>[5]</sup>、多尺度分析<sup>[6]</sup>、卫星通信<sup>[7]</sup>等众多科学领域中, 许多实际信号都有一定程度的稀疏性或可压缩性。这种结构特殊、广泛存在的信号成为了信号处理领域热门的研究方向。针对离散傅里叶变换, 几种频域稀疏的亚线性算法<sup>[8-11]</sup>相继被提出, 但这些算法由于结构的限制只适用于非常稀疏

的信号。

近几年, 稀疏快速傅里叶变换 (sparse Fast Fourier Transform, sFFT) 算法成为研究稀疏信号的一大热点, 并相继提出了一些实用的 sFFT 算法。该类算法利用信号频域的稀疏性对频率进行“分桶”, 使 DFT 运算的点数成倍减少, 然后通过其他方法实现对原信号完整频谱的重构, 从而使算法的复杂度与信号长度成亚线性关系, 进而提高信号处理速度。近年来稀疏傅里叶变换的算法硕果累累, 研究人员提出了许多算法。这些算法降低了算法复杂度<sup>[12-13]</sup>, 拓宽了维度适用范围<sup>[14]</sup>, 并实现了算法的并行加速<sup>[15]</sup>, 并且推广到了其他变换域<sup>[16]</sup>。但稀疏傅里叶变换都需要信号以傅氏域的稀疏度为先验信息, 通常稀疏度  $K$  是未知的, 在一定程度

**作者简介:** 刘 仲 (1991—), 男, 硕士研究生, 主研方向为通信信号处理; 李立春, 副教授、博士; 李慧启, 硕士研究生。

**收稿日期:** 2017-03-02 **修回日期:** 2017-04-05 **E-mail:** liuzhongsky@163.com

上限制了上述算法的应用。为此,针对未知稀疏度的信号,本文提出一种稀疏度自适应的稀疏傅里叶变换算法(Sparsity Adaptive Algorithm for Sparse Fast Fourier Transform, SAsFFT)。

## 1 稀疏傅里叶变换算法

稀疏傅里叶变换算法通过时域降维对频域稀疏的信号进行处理,将信号的频域信息压缩到低维空间中,在快速傅里叶变换后,通过定位和估值2个环节重构得到频域信号。稀疏傅里叶变换算法的流程如图1所示。

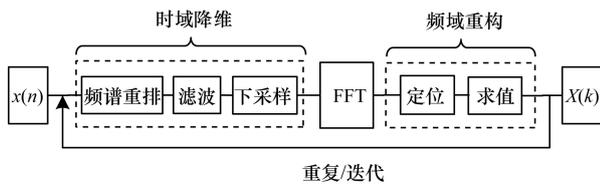


图1 稀疏傅里叶变换算法流程

### 1.1 信号的稀疏表示

对于离散输入信号  $x \in \mathbb{R}^{N \times 1}$ , 在频域上可以用正交基矢量  $\{f_i\}_{i=1}^N$  线性表示。令  $F = [f_1 f_2 \cdots f_N]$  为频域  $\mathbb{R}^{N \times 1}$  中的一组正交基, 那么信号  $x$  可表示为:

$$x = Fs \quad (1)$$

其中,  $s$  是信号  $x$  在正交基  $F$  上的投影系数向量。若  $s$  中只有  $K$  个系数不为 0, 其他  $N - K$  个系数全为 0, 那么信号  $x$  在频域上是  $K$ -稀疏的,  $K$  表示信号  $x$  的稀疏度。若  $s$  中  $K$  个系数值较大, 其他  $N - K$  个系数很小, 且几乎为 0, 则称信号  $x$  在频域上是近似  $K$ -稀疏的。若一个信号在频域上能够被稀疏表示, 则该信号可进行稀疏傅里叶变换, 也可以降维到低维空间中。

### 1.2 时域降维

如果信号在频域能够被稀疏表示, 当设置若干个桶且桶数比频率数更多时, 那么按照一定规则将各个有效频率分到这些桶中, 每个桶中只存在唯一的频率。从频域角度考虑, 将每个桶视为一个序列单元, 那么长序列可被映射为短序列。对短序列进行快速傅里叶变换, 则算法复杂度将大幅度降低。sFFT 的时域降维过程可以表示为:

$$y = \Psi_{B \times N} x \quad (2)$$

其中,  $\Psi_{B \times N}$  是时域信号  $x$  在时域降维过程中的矩阵表示,  $y$  为下采样域所对应频域的矢量,  $B \approx O(\sqrt{nk})$  为下采样域的维度或桶数。如图 1 所示, 时域降维可分为 3 个部分: 频谱重排, 滤波和下采样。

将每一桶作为一个序列单元进行处理, 其方法是取桶中的一个有效频点代表该桶的值, 该步骤称为下采样。下采样对  $x$  中的元素按索引进行等间隔

的累加, 其处理过程如式(3)所示。

$$y_i = \sum_{j=0}^{N/B-1} x_{i+B \cdot j}, i \in [0, B-1] \quad (3)$$

其中, 令  $W_N = e^{-2\pi i/N}$ , 信号  $x$  的频谱可由  $\tilde{x}$  表示。

为保证取到每一个有效频点, 利用频域卷积定理, 将时域信号与窗函数 2 个矢量相乘, 则在频域上各桶内唯一存在的有效频点展宽成多个频点, 这样在下采样时就能够取到有效频点。本文所使用的窗函数取自文献[11]的平滑窗函数。

如果多个有效频点落入同一个桶内, 频率会发生碰撞, 落入同一桶的全部频率就无法被识别出来, 这种现象称为频谱碰撞。为了避免频谱碰撞, 在对时域信号处理时, 通过频谱重排的方法使各频点在频域上均匀分布。根据文献[8, 11], 本文通过如下重排时域的方式实现傅氏域的重排。对于  $n$  维矢量  $x$ , 模  $n$  可逆的整数  $\sigma$  和整数  $\tau \in [0, n-1]$ , 如果  $(P_{\sigma, \tau} x)_i = x_{\sigma i + \tau}$ , 那么  $(\tilde{P}_{\sigma, \tau} x)_{\sigma i} = \tilde{x}_i \omega^{-\tau i}$ 。

### 1.3 频域重构

频域重构是指将下采样矢量恢复成原信号的频域矢量, 即由长度为  $B$  的  $\tilde{y}$  重构成长度为  $N$  的  $\tilde{x}$ 。目前 sFFT 的频域重构主要有两大类重构方法: 渐近法和统计法。

在正交频分复用(OFDM)技术的频偏估计方法中, 2 个采样点之间的相位差与频点坐标呈线性关系。渐近法从相位差角度出发, 运用该原理中不同频率所对应相位不同的特点, 通过对相位区间进行能量检测判断有效频率的存在与否。该方法可对相位二分查找, 或划分成多区段查找<sup>[17]</sup>逐步迭代得到频率。二分渐近法保证了算法的鲁棒性, 但复杂度相对较高; 相比之下多区段渐近法速度较快, 复杂度较低, 但鲁棒性较差。

统计法主要从重复统计角度出发, 利用哈希映射、孙子定理<sup>[11]</sup>等方法先确定下采样所对应频域中存在频点的桶, 反推出原始信号中可能包含的频率, 通过改变重排参数对哈希逆映射的多次统计, 或对不同参数下得到的线性同余方程求解, 得到原始信号中的有效频率。由于统计法重排随机进行, 因此会产生一定的频率冲突。在每次循环中存在的虚警也会产生极少量的重构信息误差<sup>[17]</sup>。但统计法均衡了算法复杂度和鲁棒性, 该方法性能较为稳定, 具有较好的研究前景。

## 2 稀疏度自适应的稀疏傅里叶变换算法

为解决未知稀疏度的问题, 基于 sFFT-1 算法的优势, 本文提出了 SAsFFT 算法。本节从算法总体流

程、稀疏度估计的影响因素和稀疏度的确定 3 个部分介绍 SAsFFT 算法。

### 2.1 算法总体流程

SAsFFT 利用了信号的频域稀疏性,在信号稀疏度未知的前提下,通过迭代逐步估计得到信号的稀疏度,从而使算法适用于未知稀疏度的信号。SAsFFT 与现有的 sFFT 算法相比,其改进在于增加了对稀疏度的估计和检测过程。该算法可以分为 3 个步骤,其总体流程如图 2 所示。

**步骤 1** 稀疏度估计。对信号稀疏度进行一定程度的过估计,即预测稀疏度  $\hat{k}$  稍大于信号的真实稀疏度  $k$ 。其中,预测稀疏度  $\hat{k}$  是对  $k$ -稀疏信号进行估计得到的稀疏度。

**步骤 2** 稀疏傅里叶变换。将预测稀疏度  $\hat{k}$  作为已知参数对信号进行稀疏傅里叶变换,得到信号中  $\hat{k}$  个傅氏域信息。

**步骤 3** 有效分量筛选。对稀疏傅里叶变换的结果进行能量检测,剔除无效的虚警分量,得到最终的  $k$  个傅氏域信息。

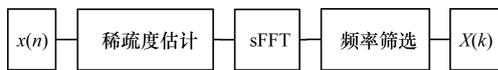


图 2 SAsFFT 算法流程

### 2.2 稀疏度估计的影响因素

利用第 2.1 节中信号的稀疏特性,以及稀疏傅里叶变换中时域降维的性质对频域稀疏的离散信号进行稀疏度的估计得到预测稀疏度。稀疏度估计的流程如下:

- 子步骤 1 初始化预测稀疏度;
- 子步骤 2 时域降维;
- 子步骤 3 快速傅里叶变换;

**子步骤 4** 能量检测,得到  $\hat{k}_m$ 。如果  $\hat{k}_m < \hat{k}_b$ ,则停止迭代;否则转跳至子步骤 2。

在稀疏度估计的算法步骤中,子步骤 1 初始化为一个较小的预测值;子步骤 2 和子步骤 3 与 sFFT-1 算法中的步骤相同;子步骤 4 通过能量检测的方法对下采样域矢量的有效分量进行计数,得出第  $m$  次迭代的预测稀疏度。如果  $\hat{k}_m < \hat{k}_b$ ,则迭代该步骤,符合要求则停止迭代。

对预测稀疏度准确性的影响因素主要有 2 个:频谱碰撞和窗函数卷积。本节主要从以下的 2 个方面进行分析。

#### 2.2.1 频谱碰撞的影响

稀疏度决定了一个信号变换到下采样域后的矢

量维度,进而影响频谱碰撞概率。首先可推导在未知稀疏度的条件下,设定稀疏度与碰撞概率的关系。文献[11]给出了定理 1,该定理限定了非零系数落入同一桶中的概率。

**定理 1** 若  $j \neq 0, n$  是 2 的幂次方,  $\sigma \in [1, n]$  是一个随机奇数,则  $\Pr[\sigma j \in [-C, C]_{\text{mod } n}] \leq 4C/n$ 。

**推论 1** 如果令  $S$  是  $\tilde{x}$  的支撑集,  $S_i$  为  $\tilde{x}$  中第  $i$  个桶区间的支撑集,那么对于任意  $i \in S$ ,有:

$$\Pr_{\sigma, \tau}[S_i \cap S_j \neq \emptyset] \leq \frac{4k}{\sqrt{\hat{nk}}}$$

证明:令  $b_i$  为下采样域的第  $i$  个元素。考虑任意  $i, j \in S$ ,由定理 1 有:

$$\Pr_{\sigma, \tau}[b_i = b_j] \leq \Pr_{\sigma, \tau}[|(\sigma i + \tau)_{\text{mod } n} - (\sigma j + \tau)_{\text{mod } n}| < n/B] = \Pr_{\sigma, \tau}[|\sigma(i - j)_{\text{mod } n}| < n/B] \leq 4/B$$

那么,从原时域的角度考虑,对于信号矢量  $x$  有:

$$\Pr_{\sigma, \tau}[S_i \cap S_j \neq \emptyset] \leq k \cdot 4/B = \frac{4k}{\sqrt{\hat{nk}}}$$

得证。

在下采样时,桶数偏小会增大碰撞概率,使预测稀疏度小于真实稀疏度。当桶数增大到一定程度时,频谱碰撞概率将会很小,几乎不会产生频谱碰撞。刚开始对稀疏度初始化时,预测稀疏度可能远小于真实稀疏度,使频谱碰撞概率  $\Pr_{\sigma, \tau}$  较大。经过对信号稀疏度的估计,由于频点碰撞的个数较少,且  $\Pr_{\sigma, \tau} < 1$ ,预测后稀疏度仍会大于预测前稀疏度。 $\hat{k}_m$  的增大使下采样维度数随之增大,从而进一步降低频谱碰撞概率。由于已知稀疏度的稀疏傅里叶算法<sup>[11]</sup>具有很强的鲁棒性,其频谱碰撞概率很小,频谱碰撞的影响可以忽略不计。如果不考虑窗函数卷积虚警的影响,那么经过几次迭代后,随着  $\hat{k}_m$  的不断增大,预测稀疏度  $\hat{k}_m$  将逐步接近真实稀疏度  $k$ 。

#### 2.2.2 窗函数卷积的影响

本节阐明窗函数的卷积会对信号中有效频率产生虚警,使得估计的频率数增加,导致预测稀疏度大于真实稀疏度。

根据文献[11],定义一个窗函数参数为  $(\varepsilon, \varepsilon', \delta, O(\frac{1}{\varepsilon - \varepsilon'} \log_a \frac{n}{\delta}))$  的平滑窗  $G \in \mathbb{R}^n$ ,其中,  $\varepsilon' < \varepsilon$ ,  $\text{supp}(G) \subseteq [-\omega/2, \omega/2]$ 。对于  $i \in [-\varepsilon'n, \varepsilon'n]$ ,通带  $\hat{G}_i \in [1 - \delta, 1 + \delta]$ ;对于  $i \notin [-\varepsilon n, \varepsilon n]$ ,阻带

$|\hat{G}_i| < \delta$ 。因为  $\hat{G}_{-o_{\sigma,\tau}(i)}$  是平滑窗函数,在频域中无法呈理想矩形窗,存在过渡带,所以对于  $(-\varepsilon n, -\varepsilon' n) \cup (\varepsilon' n, \varepsilon n)$  有过渡带  $\hat{G}_i \in (\delta, 1 - \delta)$ 。

根据文献[17]中的定理 3.3,以及对本文算法中信号和窗函数的假设,对降维过程结果可得如下的推论。

**推论 2** 将信号矢量  $\mathbf{x}$  按图 1 的流程进行时域降维,其窗函数滤波采用的窗函数为  $G$ ,那么第  $j$  桶的输出  $\hat{u}_j = \sum_{h_{\sigma,\tau}(i)=j} \hat{x}_i \hat{G}_{-o_{\sigma,\tau}(i)} \omega^{a\sigma i} \pm \delta \|\hat{x}\|_1$ ,其中  $o_{\sigma,\tau}(i)$  是频谱重排后第  $i$  个实际频率距离窗内中心频率的偏移量。

当降维过程的输出  $\hat{u}_j$  中  $o_{\sigma,\tau}(i)$  恰在过渡带中时,下采样域矢量元素的大小难以确定。如果某个元素  $\hat{u}_j$  较小,则被视为虚警频点。为了避免有效频点的丢失,设定一定的阈值  $\alpha$ ,当  $\hat{u}_j > \alpha$  时则判定存在频率。而设定阈值  $\alpha$  后,虽然避免了有效频率的丢失,但同时会引入少量的虚警频率。如果不考虑频谱碰撞,估计得到的预测稀疏度将比真实稀疏度更大。

### 2.3 信号稀疏度的确定

在实际信号中,真实稀疏度  $k$  是一个定值。通过多次迭代对稀疏度的估计,随着频谱碰撞概率逐步减小,预测稀疏度  $\hat{k}_m$  逐步增大。而窗函数卷积的虚警使预测稀疏度  $\hat{k}_m$  稍大于真实稀疏度  $k$ 。

在设计窗函数时,令其稀疏度参数为  $\hat{k}_b$ ,并使  $\hat{k}_b$  稍大于  $\hat{k}_m$ ,使得在下采样域上能够进行更精确的检测。经过迭代,在当前桶数下使得稀疏度  $\hat{k}_m < \hat{k}_b$ ,此时的碰撞概率可以忽略不计,又存在一定虚警,于是有  $\hat{k}_m > k$ 。这时停止估计稀疏度,可视  $\hat{k}_m$  为最终的预测稀疏度。而  $\hat{k}_m > k$ ,不会丢失信号有效频点的信息,对频谱重构的准确性不会产生影

响。经过算法流程步骤 2 的稀疏傅里叶变换后,输出结果中包含了  $\hat{k}_m$  个傅氏域信息。但由 2.2 节知  $\hat{k}_m > k$ 。设定阈值  $\beta$ ,遍历输出结果中的傅氏域信息,若  $\tilde{x}_i < \beta$  则视该频点为虚警频点,将该频点的值设为零,即剔除该虚警频点。

### 2.4 算法复杂度分析

根据式(3)可以计算得到下采样域  $y$  的算法复杂度为  $O(w)$ ,进行标准 FFT 的算法复杂度为

$O(B \log_a B)$ ,对下采样域进行能量检测的算法复杂度为  $O(B)$ ,因此可计算单次迭代稀疏度估计的总算法复杂度为  $O(w + B \log_a B + B) = O\left(B \log_a \frac{n}{\delta}\right)$ 。若

带入  $B = O\left(\sqrt{\frac{nk_m}{\varepsilon \log_a(n/\delta)}}\right)$ ,则第  $m$  次迭代算法复杂度为  $O\left(\sqrt{\frac{nk_m \log_a(n/\delta)}{\varepsilon}}\right)$ 。在迭代停止前,有  $\hat{k}_1,$

$\hat{k}_2, \dots, \hat{k}_{m-1} < k, \hat{k}_m$  相对于  $k$  的过估计偏差也很小,而次数  $m$  仅有数次(在后续仿真实验中可予以验证),因此该步骤的算法复杂度为  $O\left(mB \log_a \frac{n}{\delta}\right)$ 。

sFFT-1 的算法复杂度为  $O\left(LB \log_a \frac{n}{\delta} + Ldk \frac{n}{B}\right)$ ,步骤 3 的算法复杂度仅有  $O(B)$ ,因此 SAsFFT 的算法复杂度为:

$$O\left(mB \log_a \frac{n}{\delta} + LB \log_a \frac{n}{\delta} + Ldk \frac{n}{B} + B\right) \quad (4)$$

代入参数  $B, d$  后为  $O\left(\sqrt{\frac{nk \log_a(n/\delta)}{\varepsilon}} \log_a n\right)$ ,

这与原算法复杂度相同。从上述分析中可以看出,如式(4) SAsFFT 的算法复杂度中算法增加的算法复杂度为  $O\left(mB \log_a \frac{n}{\delta} + B\right)$ 。在化简中可与中间两项合并,因此,对稀疏傅里叶变换算法的整体影响较小,只增加了少量运算时间。

## 3 仿真结果与分析

为了验证 SAsFFT 的可行性和鲁棒性,本节对 SAsFFT、FTW 和 sFFT-1 3 种算法进行了仿真实验。仿真实验分别是在不同信号长度、不同稀疏度下的算法对比,以及 SAsFFT 算法的鲁棒性仿真。本文算法中所采用测试信号的产生方式参考了文献[9]。时域降维过程使用的窗函数为 2.2 节中所定义的窗函数。

信号参数设置如下:信号  $\mathbf{x}$  的矢量长度为  $n$ ,在信号的傅氏域上从  $[1, n]$  中随机选择  $k$  个分量,各分量的幅度为 1,相位随机,其他分量设为 0。在仿真结果中,每个点代表不同参数的信号,每个点是经过 100 次独立的重复实验的平均结果。在实验中,与 SAsFFT 算法进行对比的是 FTW(the Faster Fourier Transform in the West)和 sFFT-1<sup>[11]</sup>。

实验平台采用 Ubuntu Linux 16.04 版本,处理器为 Intel(R) Core(TM) i5-4210M CPU @ 2.60 GHz,内存为 8 GB。

### 3.1 信号长度对不同算法的影响比较

对于本文实验,固定信号稀疏度参数  $k = 50$ , 选择了 8 个不同的信号长度分别为  $n = 2^{17}, 2^{18}, \dots, 2^{24}$ 。在不同的信号长度下,对同一信号分别运行 SAsFFT、FFTW 和 sFFT-1,得到 3 种算法的平均运行时间对比如图 3 所示。

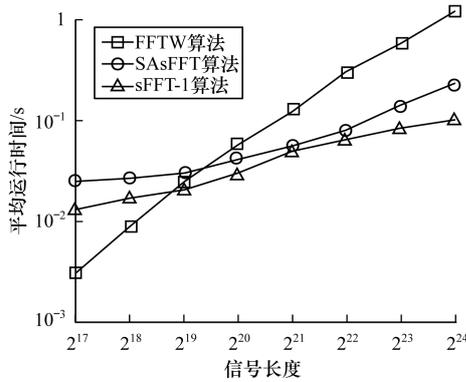


图 3 不同信号长度的算法平均运行时间对比

从图 3 中可以发现,在横坐标为对数分度时, SAsFFT、FFTW 和 sFFT-1 的运行时间均呈线性关系, SAsFFT 和 sFFT-1 的斜率基本相同,且比 FFTW 的斜率小。SAsFFT 与 sFFT-1 相比,在不同信号长度下 SAsFFT 的平均运算时间均相对较长,但平均运算时间差比较稳定。同时,由图 3 可知,若信号稀疏度  $k = 50$ ,当信号长度  $n > 2^{19}$ 时,则 SAsFFT 运算速度比 FFTW 快。信号长度越长, SAsFFT 的平均运算时间越短,其运算速度优势体现就越明显。

### 3.2 信号稀疏度对不同算法的影响比较

对于本文实验,固定信号长度参数  $n = 2^{22}$ , 选择了 11 个不同的信号稀疏度  $k = 50, 100, 200, \dots, 1000$ 。在不同的信号稀疏度下,对同一信号分别运行 SAsFFT、FFTW 和 sFFT-1,得到 3 种算法的平均运行时间对比如图 4 所示。

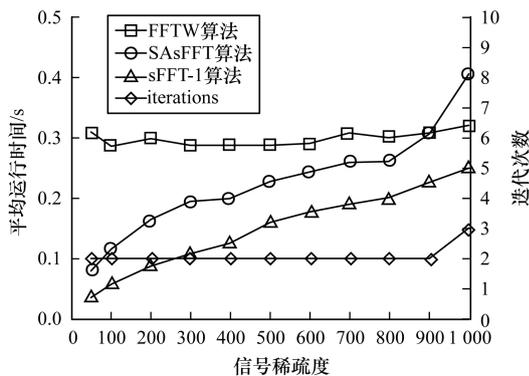


图 4 不同信号稀疏度的算法平均运行时间对比

从图 4 可以观察到,FFTW 在不同信号稀疏度下运算时间基本保持不变,而 SAsFFT 和 sFFT-1 的

平均运算时间均呈线性关系。在不同信号稀疏度下 SAsFFT 的平均运算时间比 sFFT-1 长,但两者的平均运算时间差也比较稳定。当信号长度固定不变时,信号稀疏度为 900, SAsFFT 与 FFTW 的平均运算时间基本相等。因此,综合来看,对于小于 900 的信号稀疏度, SAsFFT 的运算时间更快。

除此之外,图 4 给出了不同稀疏度下 SAsFFT 算法的迭代次数。当稀疏度  $k \leq 900$  时,迭代次数维持在 2 次,可见 SAsFFT 的稀疏度估计过程比较稳定,即 SAsFFT 算法的运算时间比较稳定,不会出现较大波动。

### 3.3 信号稀疏度对算法鲁棒性的影响

对于本文实验,固定信号长度参数  $n = 2^{22}$ , 选择了 11 个不同的信号稀疏度  $k = 50, 100, 200, \dots, 1000$ 。对于每次运算,这里误差测度采用 SAsFFT 的输出  $\tilde{x}'$  与  $k$ -稀疏矢量  $x$  的每符号平均  $l_1$  误差,如式(5)所示。

$$Error = \frac{1}{k} \sum_{0 < i \leq n} |\tilde{x}'_i - \bar{x}_i| \quad (5)$$

在不同的信号稀疏度下,对同一信号分别运行 SAsFFT,得到该算法的每符号平均  $l_1$  误差对比,如图 5 所示。

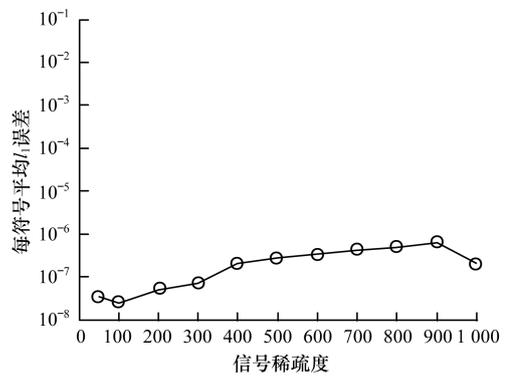


图 5 不同信号稀疏度的鲁棒性对比

由图 5 可以得到,对于固定长度的信号,算法不同信号稀疏度的每符号平均  $l_1$  误差都小于  $10^{-6}$ ,算法保持了较高的稳定性,完全可以满足绝大部分信号的实际应用需求。

## 4 结束语

本文提出一种基于稀疏度自适应的稀疏傅里叶变换算法。该算法根据频谱碰撞和窗函数卷积对频率检测的影响估计,得到一个稍大于真实稀疏度的预测稀疏度,通过已有的稀疏傅里叶变换后,在算法的输出中剔除冗余数据,从而得到较好的结果。根据数学理论分析和算法运行的实验结果表明,该算法实现

了未知稀疏度条件下的稀疏傅里叶变换,扩展了稀疏傅里叶变换算法的应用范围;算法运算量小,复杂度低,在运算时间上快于传统 FFT 算法,且完成傅里叶变换后的效果理想,验证了算法的可靠性。

#### 参考文献

- [1] CHANDRAKASAN A, GUTNIK V, XANTHOPOULOS T. Data Driven Signal Processing: An Approach for Energy Efficient Computing [C]//Proceedings of International Symposium on Low Power Electronics and Design. Washington D. C., USA; IEEE Press, 1996:347-352.
- [2] DONOHO D L. Compressed Sensing [J]. IEEE Transactions on Information Theory, 2006, 52(4):1289-1306.
- [3] CANDES E J, WAKIN M B. An Introduction to Compressive Sampling [J]. IEEE Signal Processing Magazine, 2008, 25(2):21-30.
- [4] 卢迅,李立春,李鸥,等.非重构压缩样值跳频信号时频联合估计算法[J].信息工程大学学报,2016, 17(4):425-430.
- [5] LINIAL N, MANSOUR Y, NISAN N. Constant Depth Circuits, Fourier Transform, and Learnability[J]. Journal of the ACM, 1989, 40(3):607-620.
- [6] DAUBECHIES I, RUNBORG O, ZOU A J. A Sparse Spectral Method for Homogenization Multiscale Problems[J]. Siam Journal on Multiscale Modeling & Simulation, 2007, 6(3):711-740.
- [7] 范星宇.低轨卫星星载通信信号处理关键技术研究[D].北京:北京理工大学,2014.
- [8] GILBERT A C, GUHA S, INDYK P, et al. Near-optimal Sparse Fourier Representations via Sampling[C]//Proceedings of ACM Symposium on Theory of Computing. New York, USA; ACM Press, 2002:152-161.
- [9] GILBERT A C, MUTHUKRISHNAN S, STRAUSS M. Improved Time Bounds for Near-optimal Sparse Fourier Representations [C]//Proceedings of SPIE '05. Washington D. C., USA; IEEE Press, 2005:398-412.
- [10] IWEN M A, GILBERT A, STRAUSS M. Empirical Evaluation of a Sub-linear Time Sparse DFT Algorithm[J]. Communications in Mathematical Sciences, 2007, 5(4):981-998.
- [11] IWEN M A. Combinatorial Sublinear-time Fourier Algorithms[J]. Foundations of Computational Mathematics, 2010, 10(3):303-338.
- [12] HASSANIEH H, INDYK P, KATABI D, et al. Simple and Practical Algorithm for Sparse Fourier Transform [C]//Proceedings of ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics. New York, USA; ACM Press, 2012:1183-1194.
- [13] 孟祥玉,李静,彭华.一种基于迭代更新的稀疏傅里叶变换改进算法[J].信息工程大学学报,2016, 17(6):669-674.
- [14] INDYK P, KAPRALOV M. Sample-optimal Fourier Sampling in Any Constant Dimension[C]//Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science. Philadelphia, USA; IEEE Press, 2014:514-523.
- [15] WANG C, MAURICIO A P, CHANDRASEKARAN S, et al. Parallel Sparse FFT [C]//Proceedings of Workshop on Irregular Applications: Architectures and Algorithms. New York, USA; ACM Press, 2013:10-23.
- [16] 周广福,文成林,高敬礼.基于小波变换与稀疏傅里叶变换相结合的光场重构方法[J].电子学报,2017, 45(4):782-790.
- [17] HAITHAM H, INDYK P, KATABI D, et al. Nearly Optimal Sparse Fourier Transform[C]//Proceedings of the 44th ACM Symposium on Theory of Computing. New York, USA; ACM Press, 2012:563-578.

编辑 索书志

(上接第140页)

#### 参考文献

- [1] IEEE. IEEE Standard for Signal and Test Definition: 1641<sup>TM</sup>-2010[S]. New York, USA; Institute of Electrical and Electronics Engineers, Inc., 2010.
- [2] IEEE. IEEE Trial Use Standard for Automatic Test Markup Language (ATML) for Exchanging Automatic Test Equipment and Test Information via XML: Exchanging Instrument Descriptions; 1671. 5<sup>TM</sup>-2008[S]. New York, USA; Institute of Electrical and Electronics Engineers, Inc., 2008.
- [3] 刘金宁,孟晨,赵锦成.面向信号的自动测试系统软件模型研究[J].测控技术,2010,29(3):63-66.
- [4] 梁旭,李行善,高占宝.基于测试引擎的自动测试系统软件设计[J].电子测量与仪器学报,2006,20(5):11-16.
- [5] 钟天云.面向信号的ATS软件平台研究一系统建模工具与运行时服务设计[D].成都:电子科技大学,2013.
- [6] 齐少华.TPS流程式开发环境与仪器管理模块的研究与实现[D].成都:电子科技大学,2013.
- [7] 赵瑞贤,孟晓风,王国华.通用ATE开关资源测试路径模型及应用[J].北京航空航天大学学报,2006,32(2):181-185.
- [8] 王国华,杨中亮,陈妮亚.ATE开关矩阵动态路径搜索算法[J].北京航空航天大学学报,2010,36(1):39-42.
- [9] 王怡苹,李文海,文天柱.面向信号测试的路径搜索算法研究[J].仪器仪表学报,2013,34(7):1650-1658.
- [10] 王国华,杨中亮,陈妮亚.ATE开关矩阵动态路径搜索算法[J].北京航空航天大学学报,2010,36(1):39-42.
- [11] MCMURCHIE L, EBELING C. PathFinder: A Negotiation-based Performance-driven Router for FPGAs [C]//Proceedings of International ACM Symposium on Field-programmable Gate Arrays. New York, USA; ACM Press, 1995:111-117.
- [12] CORMEN Y H. 算法导论[M].3版.徐建平,译.北京:机械工业出版社,2013.
- [13] 宋常城.基于最小费用最大流算法的若干研究与分析[D].南京:南京邮电大学,2012.
- [14] 赵礼峰,宋常城,白睿.基于最小费用最大流问题的“排序”算法[J].计算机技术与发展,2011,21(12):82-85.
- [15] 段凡丁.关于最短路径的SPFA快速算法[J].西南交通大学学报,1994,29(2):207-212.

编辑 索书志