IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

On Performance of Sparse Fast Fourier Transform Algorithms Using the Flat Window Filter

FIRST A. AUTHOR[1], (Fellow, IEEE), SECOND B. AUTHOR[2], AND THIRD C. AUTHOR, JR.[3], (Member, IEEE)

[1]National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@boulder.nist.gov)
[2]Department of Physics, Colorado State University, Fort Collins, CO 80523 USA (e-mail: author@lamar.colostate.edu)
[3]Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA

Corresponding author: First A. Author (e-mail: author@ boulder.nist.gov).

This paragraph of the first footnote will contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456."

ABSTRACT The problem of computing the Sparse Fast Fourier Transform(sFFT) of a K-sparse signal of size N has received significant attention for a long time. The first stage of sFFT is hashing the frequency coefficients of $\hat{x}$ into $B(\approx K)$ buckets named frequency bucketization. The process of frequency bucketization is achieved through the use of filters: Dirichlet kernel filter, aliasing filter, flat filter, etc. The frequency bucketization through these filters can decrease runtime and sampling complexity in low dimensions. It is a hot topic about the sFFT Algorithms using the flat filter because of its convenience and efficiency since its emerge and widely application. The next stage of sFFT is the spectrum reconstruction by identifying frequencies that are isolated in their buckets. Up to now, there are more than thirty different sFFT algorithms using the sFFT idea as mentioned above by their unique methods. An important question now is how to analyze and evaluate the performance of these algorithms in theory and practice. In this paper, it is mainly discussed about the sFFT algorithms using the flat filter. In the first part, the paper introduces the techniques in detail including two types of frameworks, five different methods to reconstruct spectrum and corresponding algorithms. We get the conclusion of the performance of these five algorithms including runtime complexity, sampling complexity and robustness in theory. In the second part, we make three categories of experiments for computing the signals of different SNR, different N, and different K by a standard testing platform and record the run time, percentage of signal sampled and L_0, L_1, L_2 error both in exactly sparse case and general sparse case. The result of experiments is consistent with the inferences obtained in theory. It can help us to optimize these algorithms and use them correctly in the right areas.

INDEX TERMS Sparse Fast Fourier Transform(sFFT), flat window filter, sub-linear algorithms

I. INTRODUCTION

The Discrete Fourier Transform(DFT) is one of the most important and widely used techniques in signal processing and mathematical computing. The most popular algorithm to compute the DFT is the fast Fourier transform(FFT) invented by Cooly and Tukey. The algorithm can compute the DFT of a signal of size N in $O(N\log N)$ time and use $O(N)$ samples. FFT greatly simplifies the operation process, however, with the emergence of big data problems, the FFT is no longer fast enough. Furthermore, sometimes it is hard to acquire a sufficient amount of data to compute the DFT. These two problems become the the major computational bottleneck in many applications. It motivates the need for the algorithms that can compute the Fourier transform in sublinear time and that use only a subset of the input data. People thought of many ideas in order to realize such an algorithm. Later, they focused on the study of the characteristics of the signal itself. The research found that a large number of signals are sparse in the frequency domain, only K frequencies are non-zeros or are significantly large. This feature is common and inherent in signals covers many fields (e.g. audio, video data, medical image, etc). In this case, when $K << N$,one can retrieve the information with high accuracy using only the coefficients of the K most significant frequencies. So the sFFT has been proposed and achieved good results. The research of sFFT has been a hot topic in signal processing research since its birth, it was named one of the 10 Breakthrough Technologies in MIT Technology Review in 2012.

The firsts stage of sFFT algorithm is bucketization such that the value of the bucket is the sum of the values of the frequency coefficients that hash into the bucket. The number of buckets is denoted by B and the size of one bucket is denoted by L. The process of bucketization is achieved through the use of filters. The effect of Dirichlet kernel filter is to make the signal convoluted a rectangular window in the time domain, it can be equivalent to the signal multiply a Dirichlet kernel window of size $L(L << N)$ in the frequency domain. The typical application using the Dirichlet kernel filter is AAFFT algorithm. The effect of aliasing filter is to make the signal multiply a comb window in the time domain, it can be equivalent to the signal convoluted a comb window of size $B(\approx K)$ in the frequency domain. The typical application using the aliasing filter is FFAST algorithm. The effect of flat filter is to make the signal multiply a mix window in the time domain, it can be equivalent to the signal convoluted a flat window of size $L(L << N)$ in the frequency domain. The typical application using the flat filter is sFFT1.0 algorithm. After bucketization, the algorithm then focuses on the non-empty buckets and computes the positions and values of the large frequency coefficients in those buckets in what we call the spectrum reconstruction or identifying frequencies. As we can see as follows, there are more than thirty algorithms using the sFFT idea and more than ten sFFT algorithms using the flat filter. A central question now is how to analyze and evaluate the performance of these algorithms for computing signals by the compare of themselves or other types of algorithms. It should be proved whether the runtime complexity, sampling complexity and robustness performance are consistent with the theory or not. Are there any better ways to improve these algorithms when using it in practice? The results of these performance analysis is the guide for us to optimize these algorithms and use them correctly in the different areas.

The first sFFT algorithm with sub-linear runtime and sub-sampling property was given in [1], which gave a randomized algorithm with runtime and sampling complexity $O(K^2 poly(\log N))$. This was later improved to $O(K poly(\log N))$ [2], [3] through the use of binary search technique for spectrum reconstruction and the use of unequally-spaced FFTs. The algorithm is so called Ann Arbor fast Fourier transform (AAFFT), the version of them are AAFFT0.5 and AAFFT0.9.

In [4], [5] an algorithm so called Fast Fourier Aliasing-based Sparse Transform(FFAST) which focuses on exactly K-sparse signals was given. Their approach is based on downsampling of the input signal using a constant number of co-prime downsampling factors guided by CRT. These aliasing patterns of different downsampled signals are formulated as parity-check constraints of good erasure-correcting sparse-graph codes. FFAST costs $O(K\log K)$ to compute the exactly signals and only use $O(K)$ samples. In [6], [7] the author adapt the FFAST framework to the case where the time-domain samples are corrupted by a white Gaussian noise. The author show that the extended noise robust algorithm R-FFAST computes DFT using $O(K\log^3 N)$ samples, in $O(K\log^4 N)$ runtime. These two algorithms perform well when N is a product of some smaller prime numbers.

In [8] an algorithm is proposed and so called sFFT by downsampling in the time domain(sFFT-DT). The idea behind sFFT-DT is to downsample the original input signal first and then all subsequent operations are conducted on the downsampled signals. To overcome aliasing problem, the author consider the locations and values of K non-zero entries as variables and the aliasing problem is found to be equivalent to MPP problem, which can be solved via orthogonal polynomials or syndrome decoding with a CS(compressive sensing) based solver.

In [9] a deterministic algorithm so called Gopher Fast Fourier Transform(GFFT) which based on CRT was given. The GFFT is a aliasing-based search algorithm. The approximation error bounds in [9] are further improved in [10]. Later, an algorithm so called Christlieb Lawlor Wang Sparse Fourier Transform(CLW-SFT), which used phase encoding method was given in [11]. The noiseless version of this algorithm is an adaptive algorithm [12] which has runtime $O(K\log K)$. In [11] the author developed this algorithm by using the multiscale error-correcting method to cope with high-level noise with runtime $O(K^2\log K)$. In [13] the author evaluate the performance of DMSFT (generated from GFFT) and CLW-DSFT (generated from CLW-SFT) and compare their runtime and robustness characteristics with other algorithms. These four algorithms all have a hypothesis that the algorithms can sample anywhere they want.

In [14] an algorithm is proposed and so called Deterministic Sparse FFT(DSFFT). In the algorithm K needs not to be known in advance but will be determined during the algorithm. The method is based on the divide-and-conquer approach and may require the solution of a Vandermonde system of size at most $K \times K$ at each iteration step j if $K^2 < 2^j$.

The sFFT algorithms using the flat window filter so called sFFT1.0-sFFT4.0 that can compute the exactly K-sparse signals in time $O(K\log N)$ and general K-sparse signals in time $O(K\log N\log(N/K))$ were given in [15], [16]. These algorithms leverage characteristic of flat window filter. The sFFT1.0 and sFFT2.0 algorithms can identify and estimate the K largest coefficients in one shot. The sFFT3.0 algorithm can estimate the position by using only two samples of the filtered signal inspired by the frequency offset estimation in the exactly sparse case. Later, a new robust algorithm so called Matrix Pencil FFT(MPFFT) was proposed in [17] on the basic of sFFT3.0 algorithm. The major new ingredient is a mode collision detector based on the matrix pencil method. This method enables the algorithm to use fewer samples of the input signal.

In [18] the paper proposes an overview of sFFT and summarize a three-step approach in the stage of spectrum reconstruction and provides a standard testing platform that can be used to evaluate different sFFT algorithms. There are also some researches try to conquer the sFFT problem from a lot of aspects: complexity [19], [20], performance [21], [22], software [23], [24], higher dimensions [25], [26], implementation [27], hardware [28] and special setting [29], [30] perspectives.

The identification of different sFFT algorithms can be know through a brief analysis as above. The Dirichlet kernel filter is not efficient because it only bins some frequency coefficients into one bucket one time. As to the aliasing filter it is difficult to solve the worst case because there may be many frequency coefficients in the same bucket accidentally if B only can be supposed as a power of two by the reason of the scaling operation is of no use. In comparison to them, using the flat filter is very convenience and efficiency.

This paper is structured as follows. Section II and Section III provides a brief overview of the sFFT technique. Section IV introduce and analyze two frameworks and five spectrum reconstruction methods of algorithms. In the one shot framework, sFFT1.0 and sFFT2.0 algorithm use voting method with the help of the stochastic characteristics. In the framework of iteration the sFFT3.0 and sFFT4.0 algorithm use phase encoding method with the help of the time shift characteristics. The MPSFT algorithm use matrix pencil method with the help of prony characteristics. In section V, we do three categories of comparison experiments. The first kind of experiment is to compare them with each other. The second is to compare them with other sFFT algorithms. The third is to compare them of optimization with them without optimization. The analyze of the experiment results satisfy theoretical inference.

II. NOTATION

In this section, we initially present some notation and basic definitions about sFFT. We use $\omega_N = e^{-2\pi \mathbf{i}/N}$ as the N-th root of unify. Let $\mathbf{F}_N \in \mathbb{C}^{N\times N}$ be the DFT matrix of size N defined as follows:

$$\mathbf{F}_N[j,k] = \frac{1}{N}\omega_N^{jk} \tag{1}$$

The DFT of a vector $x \in \mathbb{C}^N$ (consider a signal of size N, where N is a power of two) is a vector $\hat{x} \in \mathbb{C}^N$ defined as follows:

$$\hat{x} = \mathbf{F}_N x \tag{2}$$

$$\hat{x}_i = \frac{1}{N}\sum_{j=0}^{N-1} x_j \omega_N^{ij} \tag{3}$$

It is necessary to consider the inverse of the DFT matrix above. $\mathbf{F}_N^{-1} \in \mathbb{C}^{N\times N}$ defined as follows:

$$\mathbf{F}_N^{-1}[j,k] = \omega_N^{-jk} \tag{4}$$

The inverse DFT of is a vector x defined as follows:

$$x = \mathbf{F}_N^{-1}\hat{x} = \mathbf{F}_N^{-1}(\mathbf{F}_N x) \tag{5}$$

$$x_i = \sum_{j=0}^{N-1} \hat{x}_j \omega_N^{-ij} \tag{6}$$

For $x_{-i} = x_{N-i}$, we may define convolution as follows:

$$(x * y)_i = \sum_{j=0}^{N-1} x_j y_{i-j} \tag{7}$$

For coordinate-wise product $(xy)_i = x_i y_i$ and the DFT of xy is performed as described in Equation 8:

$$\widehat{xy} = \hat{x} * \hat{y} \tag{8}$$

For exactly signals, $\hat{x}$ is exactly K-sparse if it has exactly K non-zero frequency coefficients while the remaining $N-K$ coefficients are zero. For general signals, $\hat{x}$ is general K-sparse if the largest K frequency coefficients $>>$ remaining $N-K$ coefficients. The goal of the sFFT is to recover a K-sparse approximation $\hat{x}$ by finding frequency positions f and estimating values $\hat{x}_f$ of the K largest coefficients.

III. TECHNIQUES

In this section, we start with an overview of the techniques that we will use in the sFFT.

A. RANDOM SPECTRUM PERMUTATION

The random permutation include two operations, one is shift operation, another is scaling operation. Let $\tau \in \mathbb{R}$ be the offset parameter. Let matrix $\mathbf{S}_\tau \in \mathbb{R}^{N\times N}$ representing the shift operation, is defined as follows:

$$\mathbf{S}_\tau[j,k] = \begin{cases} 1, & j-\tau \equiv k (\mathrm{mod} N) \\ 0, & \text{o.w.} \end{cases} \tag{9}$$

Let $\sigma \in \mathbb{R}$ be the scaling parameter. Let matrix $\mathbf{P}_\sigma \in \mathbb{R}^{N\times N}$ representing the scaling operation, is defined as follows:

$$\mathbf{P}_\sigma[j,k] = \begin{cases} 1, & \sigma j \equiv k (\mathrm{mod} N) \\ 0, & \text{o.w.} \end{cases} \tag{10}$$

Suppose $\sigma^{-1} \in \mathbb{R}$ exists mod N, σ^{-1} satisfies $\sigma^{-1}\sigma \equiv 1(\mathrm{mod} N)$. If a vector $x' \in \mathbb{C}^N, x' = \mathbf{S}_\tau \mathbf{P}_\sigma x$, such that:

$$\begin{aligned} x'_i &= x_{\sigma(i-\tau)} \\ x'_{\sigma^{-1}i+\tau} &= x_i \end{aligned} \tag{11}$$

The random permutation isolates spectral components from each other, and it is performed as follows: if $x' = \mathbf{S}_\tau \mathbf{P}_\sigma x$, such that:

$$\begin{aligned} \hat{x}'_{\sigma i} &= \hat{x}_i \omega^{\sigma\tau i} \\ \hat{x}'_i &= \hat{x}_{\sigma^{-1}i} \omega^{\tau i} \end{aligned} \tag{12}$$

B. WINDOW FUNCTION

The window funciton is a mathematical tool and can be seen as a matrix multiply the original signal. We introduce three filters used in the sFFT algorithm mentioned in this paper.

The first filter is the frequency aliasing filter. Through the filter, the signal in the time domain is subsampled such that the corresponding signal in the frequency domain is aliased.

Let $L \in \mathbb{Z}^+$ be the subsampling factor. Let $B \in \mathbb{Z}^+$ be the subsampling number. Let matrix $\mathbf{D}_L \in \mathbb{R}^{B\times N}$, representing the subsampling operator, is defined as follows:

$$\mathbf{D}_L[j,k] = \begin{cases} 1, & k = jL \\ 0, & \text{o.w.} \end{cases} \tag{13}$$

Let vector $y_{L,\tau}, \hat{y}_{L,\tau} \in \mathbb{C}^B$, be the filtered signal obtained by shift operation and aliasing filter. If $y_{L,\tau} = \mathbf{D}_L\mathbf{S}_\tau x,\ \hat{y}_{L,\tau} = \mathbf{F}_B\mathbf{D}_L\mathbf{S}_\tau x$, we get formula (14) and if τ=0 we get formula (15).

$$\begin{aligned}\hat{y}_{L,\tau}[i] =& \hat{x}[i]\omega^{-\tau i} + \hat{x}[i+B]\omega^{-\tau(i+B)} + ... \\ & \hat{x}[i+(L-1)B]\omega^{-\tau(i+(L-1)B)}\end{aligned} \tag{14}$$

$$\hat{y}_{L,0}[i] = \hat{x}[i] + \hat{x}[i+B] + ... + \hat{x}[i+(L-1)B] \tag{15}$$

The second filter is the frequency flat filter. We use a filter vector G that is concentrated both in time and frequency domain, G is zero except at a small number of time coordinates with $\text{supp}(G) \subseteq [-w/2, w/2]$ and its Fourier transform $\hat{G}$ is negligible except at a small fraction L ($\approx \varepsilon N$) of the frequency coordinates (the pass region). The paper [15] claim there exists a standard window function $G(\varepsilon, \varepsilon', \delta, w)$ satisfies the formula (16). The filter can be obtained by convoluted a Gassian function with a box car window function and $\text{supp}(G) = w = O(1/\varepsilon \log(1/\delta))$. One can potentially use a Dolph-Chebyshev window function with minimal big-Oh constant. In this paper, we use filter $G \in \mathbb{C}^N$ be a $(L/N, L/2N, \delta, w)$ flat window. The width of the filter in the time domain is denoted by w, the width of the passband region in the frequency domain is denoted by L, the number of buckets is denoted by B and $B = N/L$.

$$\begin{aligned}\hat{G}_i &\in [1-\delta, 1+\delta] \ for\ i \in [-\epsilon' N, \epsilon' N] \\ \hat{G}_i &\in [-\delta, \delta] \ for\ i \notin [-\epsilon N, \epsilon N] \\ \hat{G}_i &\in [0,1] \ for\ all\ i\end{aligned} \tag{16}$$

Let matrix $\mathbf{Q}_L \in \mathbb{C}^{N\times N}$ be a diagonal matrix whose diagonal entries represent filter coefficients in the time domain, is defined as follows:

$$\mathbf{Q}_L[j,k] = \begin{cases} G_j, & j = k \\ 0, & \text{o.w.} \end{cases} \tag{17}$$

The third filter is the frequency subsampled filter. Through the filter, the signal in the time domain is aliased such that the corresponding signal in the frequency domain is subsampled. Let matrix $\mathbf{U}_L \in \mathbb{R}^{B\times N}$ represents the aliasing operator as follows:

$$\mathbf{U}_L[j,k] = \begin{cases} 1, & j-k \equiv 0 (\text{mod} B) \\ 0, & \text{o.w.} \end{cases} \tag{18}$$

Let vector $y_L,\ \hat{y}_L \in \mathbb{C}^B$, be the filtered signal obtained by subsampled filter. If $y_L = \mathbf{U}_L x,\ \hat{y}_L = \mathbf{F}_B\mathbf{U}_L x$ we get formula(19).

$$\hat{y}_L[i] = \hat{x}[iL] \tag{19}$$

C. FREQUENCY BUCKETIZATION

The process of bucketization in this paper is achieved through the use of flat filter, subsampled filter and shift operation, scaling operation. It can be equivalent to the signal multiply $\mathbf{F}_B\mathbf{U}_L\mathbf{Q}_L\mathbf{S}_\tau\mathbf{P}_\sigma$. The filtered signal is performed as follows: If $y_{L,\tau,\sigma} = \mathbf{U}_L\mathbf{Q}_L\mathbf{S}_\tau\mathbf{P}_\sigma x,\ \hat{y}_{L,\tau,\sigma} = \mathbf{F}_B\mathbf{U}_L\mathbf{Q}_L\mathbf{S}_\tau\mathbf{P}_\sigma x$, such that:

$$\begin{aligned}\hat{y}_{L,\tau,\sigma}[0] \approx& \hat{G}_{\frac{L}{2}}\hat{x}_{\sigma^{-1}(-\frac{L}{2})}\omega_N^{\tau(-\frac{L}{2})} + \dots \\ & \hat{G}_{-\frac{L}{2}+1}\hat{x}_{\sigma^{-1}(\frac{L}{2}-1)}\omega_N^{\tau(\frac{L}{2}-1)} \\ \hat{y}_{L,\tau,\sigma}[1] \approx& \hat{G}_{\frac{L}{2}}\hat{x}_{\sigma^{-1}(\frac{L}{2})}\omega_N^{\tau(\frac{L}{2})} + \dots \\ & \hat{G}_{-\frac{L}{2}+1}\hat{x}_{\sigma^{-1}(\frac{3L}{2}-1)}\omega_N^{\tau(\frac{3L}{2}-1)} \\ \hat{y}_{L,\tau,\sigma}[i] \approx& \hat{G}_{\frac{L}{2}}\hat{x}_{\sigma^{-1}(\frac{(2i-1)L)}{2})}\omega_N^{\tau(\frac{(2i-1)L}{2})} + \dots \\ & \hat{G}_{-\frac{L}{2}+1}\hat{x}_{\sigma^{-1}(\frac{(2i+1)L}{2}-1)}\omega_N^{\tau(\frac{(2i+1)L}{2}-1)}\end{aligned} \tag{20}$$

If the set I is a set of coordinates position, the position $f = (\sigma^{-1}u)\text{mod}N \in I$, suppose there is no hash collision in the bucket i, i = round(u/L), round() means to make decimals rounded. Through formula(20), we can get the formula(21)

$$\begin{aligned}\hat{y}_{L,\tau,\sigma}[i] &\approx \hat{G}_{iL-u}\hat{x}_{\sigma^{-1}u}\omega_N^{\tau u} \text{ for } u \in [\frac{(2i-1)L}{2}, \frac{(2i+1)L}{2} - 1] \\ \hat{x}_f &\approx \hat{y}_{L,\tau,\sigma}[i]\omega_N^{-\tau u}/\hat{G}_{iL-u} \text{ for } u = \sigma f \text{mod} N, i = \text{round}(u/L)\end{aligned} \tag{21}$$

As we see above, frequency bucketization includes 3 steps: random spectrum permutation($x' = \mathbf{S}_\tau\mathbf{P}_\sigma x$, it cost 0 runtime and unknow samples), flat window filter($x'' = \mathbf{Q}_L x'$, it cost w runtime and w samples), fourier transform of the aliasing signal($\hat{y}_{L,\tau,\sigma} = \mathbf{F}_B\mathbf{U}_L x''$, it cost $B\log B$ runtime and 0 samples). So totally frequency bucketization one round cost $w + B\log B$ runtime and w samples

IV. ALGORITHMS ANALYSIS

As mentioned above the goal of frequency bucketization is to decrease runtime and sampling complexity in advantage of low dimensions, after bucketizationafter the filtered signal $\hat{y}_{L,\tau,\sigma}$ can be obtained by original signal x. In this section, we introduce two frameworks, five methods and corresponding algorithms to recover the spectrum $\hat{x}$ of the filtered signal $\hat{y}_{L,\tau,\sigma}$ by their own way.

A. THE SFFT1.0 ALGORITHM BY ONE-SHOT FRAMEWORK

The first framework can directly reconstruct the spectrum by one-shot, does not need iteration. The process to reconstruct the spectrum of sFFT1.0 algorithm includes two kinds of rounds, the first are location rounds and another are estimation rounds. Every location round one time generate a list of candidate coordinates I_r. Candidate coordinates $i \in I_r$ have a certain probability of being indices of one of the K significant coefficients in spectrum. By running multiple rounds, this probability can be increased so it is certain to vote the candidate coordinates with a high probability after $R(\approx \log N)$ times' rounds. The next step is to do estimation rounds used to exactly determine the value of identified frequency $\hat{x}_f$ isolated in the bucket in the reason of the value of the bucket is approximate the frequency that identified in the bucket if there is no hash collision. The block diagram of the sFFT algorithms system of one-shot framework is shown in Figure 1. We explain the details as below.

Stage1 Bucketization: Run R times' round for set $\tau = \{\tau_1, \tau_2, ...\tau_R\}$ and set $\sigma = \{\sigma_1, \sigma_2, ...\sigma_R\}$, Calculate $\hat{y}_{L,\tau,\sigma} = \mathbf{F}_B \mathbf{U}_L \mathbf{Q}_L \mathbf{S}_\tau \mathbf{P}_\sigma x$ representing filtered spectrum.

Stage2-Step1 Location rounds: After R times' round, return R sets of coordinates $I_1, , I_R$(set I_r representing union of $2K$ sets J from B sets J in the No.r' round). Then do the vote, count the number s_i of occurrences of each found coordinate i, that is: $s_i = \| \{r | i \in I_r\} \|_0$ ($\| \|_0$ representing ℓ_0-norm). Only keep the coordinates occurred in at least fifty percentage proportion($I = \{i \in I_1 \cup ... \cup I_R | s_i > R/2\}$).

Stage2-Step2 Estimation rounds: After location rounds the set I can be obtained then estimate R sets of frequency coefficients $\hat{x}^1, ..., \hat{x}^R$. The method is if position $f \in I$, we can get the value of position f through formula(21). For identified position f, R different $\hat{x}_f^r$ can be obtained in R times' round, finally use the median value of the sets as the final estimator.

Finally we analyze the performance of sFFT1.0 algorithm. In stage1 it cost $R(w + BlogB)$ runtime, in stage2-step1 it cost $2RK(N/B)$ runtime, in stage2-step 2 it cost $2RK$ runtime, totally it cost $O(R(w+BlogB+KN/B))$ runtime. And in fact the runtime satisfy Lemma 4.1.

Lemma 4.1: Suppose $R = O(\log N)$, $w = B\log(N/\delta)$, $\delta = 1/(N^c)$, it cost $O(\log N\sqrt{NK\log N})$ runtime in sFFT1.0 algorithm

Proof 4.1:

$$\begin{aligned}
&O(R(w + B\log B + KN/B)) \\
=&O\left(\log N \log(N/\delta)B + \log NKNB^{-1}\right) \\
\geq&O(\log N\sqrt{NK\log N})(\text{ for } B = \sqrt{NK\log^{-1}(N/\delta)})
\end{aligned}$$

In stage 1 it needs w samples one time. In the first round, the signal not chosen is in probability of $(N - w)/N$, suppose the probability does not change, in average the samples chosen after R times' round is in the number of $N\left(1 - \left(\frac{N-w}{N}\right)^R\right) = N\left(1 - \left(\frac{N-\sqrt{KN\log(N/\delta)}}{N}\right)^{(\log N)}\right)$

B. THE SFFT2.0 ALGORITHM BY ONE-SHOT FRAMEWORK

As is shown in the Figure 1, sFFT2.0 is very similar to sFFT1.0. Additional another bucketization and location round are used with the frequency aliasing filter to restrict the locations of the large coefficient. Let M be the size of the aliasing filter and M divides N, it does a pre-processing stage as below. Firstly Obtain $\hat{y}_{M,0} = \mathbf{F}_M \mathbf{D}_M \mathbf{S}_0 x$, then union of $2K$ sets from M sets $\{0, M, ..., (L-1)M\}...\{M-1, 2M-1, ..., N-1\}$ by selecting the $2K$ largest coefficients of magnitude of $\hat{y}_{M,0}[i]$ for $i \in [0, M-1]$, the union of $2K$ sets is set I' , assuming that all large coefficients j have j mod M in I'. That is, we restrict out sets I_r talked above to contain only coordinates i with i mod $M \in I'$,we expect that $|I_r| \approx 2K/M(2KN/B)$ rather than the previous $|I_r| \approx 2KN/B$.

Finally we analyze the performance of sFFT2.0 algorithm. In stage1 it is the same as the sFFT1.0, in stage2-step1 it cost $R(2K/M \cdot 2K(N/B) + M) + M\log M$ runtime, in stage2-step 2 it is the same, totally it cost $O(R(w+BlogB+K^2N/(BM) + M) + M\log M)$ runtime. And in fact the runtime satisfy Lemma 4.2.

Lemma 4.2: Suppose $R = O(\log N)$, $w = B\log(N/\delta)$, $\delta = 1/(N^c)$, it cost $O\left(\log N\left(K^2N\log(N)\right)^{1/3}\right)$ runtime in sFFT1.0 algorithm

Proof 4.2:

$$\begin{aligned}
&O(R(w + BlogB + K^2N/(BM) + M) + M\log M) \\
=&O\left(\log N\log(N/\delta)B + \log NK^2NM^{-1}B^{-1} + M\log M\right) \\
\geq&O(\log NK\sqrt{\log(N/\delta)NM^{-1}} + M\log M) \\
\approx&O(\log NK\sqrt{\log(N/\delta)NM^{-1}} + M\log N) \\
\geq&O\left(\log N\left(K^2N\log(N)\right)^{1/3}\right)
\end{aligned}$$

Compared to sFFT1.0, the runtime is factor $(N\log N)^{1/6}$ smaller. In average the sample of sFFT2.0 chosen is in the number of $N\left(1 - \left(\frac{N-w}{N}\right)^R\right) + M$, compared to the sFFT1.0, B decreases so w decreases so that the samples decreases as well.

C. THE SFFT3.0 ALGORITHM BY ITERATION FRAMEWORK

Compared to the one shot framework the iteration framework has two improvements. The first advantage of iteration framework is that once a frequency coefficient of the signal was found and estimated, it can be subtracted from the signal. This fact can be used to reduce the amount of work to be done in subsequent steps. It is not necessary to update the whole input signal. Instead, it is sufficient to update the B-dimensional buckets. This way the removal of the effects of already found coefficients can be done in $O(B)$ time. The second important addition in iteration framework is an improved scheme for finding the signal's significant frequency coordinates using other methods rather than voting method by some numbers of rounds. In one-shot framework, R times' rounds were run and their results combined in order to get correct locations at a high probability. In the iteration algorithms, two or $\log_2 L$ rounds is enough by their own ways.

V. CONCLUSION

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate

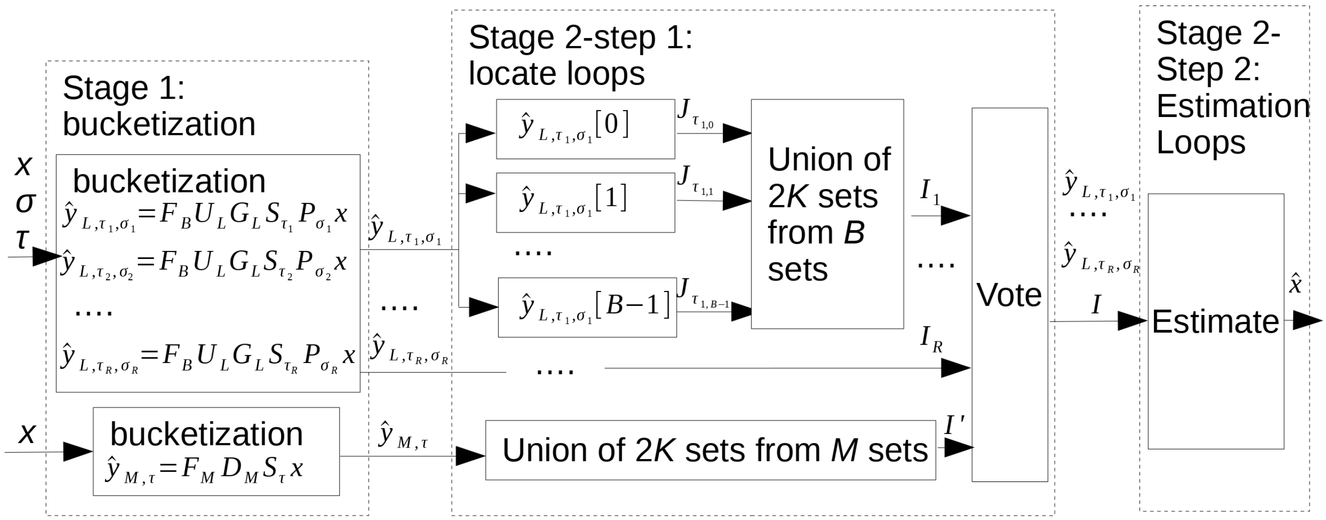

FIGURE 1. A system block diagram of one-shot framework.

the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

.

Appendixes, if needed, appear before the acknowledgment.

ACKNOWLEDGMENT
The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Avoid expressions such as “One of us (S.B.A.) would like to thank . . . .” Instead, write “F. A. Author thanks . . . .” In most cases, sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page, not here.

REFERENCES AND FOOTNOTES

A. REFERENCES

References need not be cited in text. When they are, they appear on the line, in square brackets, inside the punctuation. Multiple references are each numbered with separate brackets. When citing a section in a book, please give the relevant page numbers. In text, refer simply to the reference number. Do not use “Ref.” or “reference” except at the beginning of a sentence: “Reference [?] shows . . . .” Please do not use automatic endnotes in Word, rather, type the reference list at the end of the paper using the “References” style.

Reference numbers are set flush left and form a column of their own, hanging out beyond the body of the reference. The reference numbers are on the line, enclosed in square brackets. In all references, the given name of the author or editor is abbreviated to the initial only and precedes the last name. Use them all; use et al. only if names are not given. Use commas around Jr., Sr., and III in names. Abbreviate conference titles. When citing IEEE transactions, provide the issue number, page range, volume number, year, and/or month if available. When referencing a patent, provide the day and the month of issue, or application. References may not include all information; please obtain and include relevant information. Do not combine references. There must be only one reference with each number. If there is a URL included with the print reference, it can be included at the end of the reference.

Other than books, capitalize only the first word in a paper title, except for proper nouns and element symbols. For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation See the end of this document for formats and examples of common references. For a complete discussion of references and their formats, see the IEEE style manual at http://www.ieee.org/authortools.

B. FOOTNOTES

Number footnotes separately in superscript numbers.[1] Place the actual footnote at the bottom of the column in which it is cited; do not put footnotes in the reference list (endnotes). Use letters for table footnotes (see Table **??**).

APPENDIX A SUBMITTING YOUR PAPER FOR REVIEW

A. FINAL STAGE

When you submit your final version (after your paper has been accepted), print it in two-column format, including figures and tables. You must also send your final manuscript on a disk, via e-mail, or through a Web manuscript submission system as directed by the society contact. You may use Zip for large files, or compress files using Compress, Pkzip, Stuffit, or Gzip.

[1] It is recommended that footnotes be avoided (except for the unnumbered footnote with the receipt date on the first page). Instead, try to integrate the footnote information into the text.

Also, send a sheet of paper or PDF with complete contact information for all authors. Include full mailing addresses, telephone numbers, fax numbers, and e-mail addresses. This information will be used to send each author a complimentary copy of the journal in which the paper appears. In addition, designate one author as the "corresponding author." This is the author to whom proofs of the paper will be sent. Proofs are sent to the corresponding author only.

B. REVIEW STAGE USING SCHOLARONE® MANUSCRIPTS

Contributions to the Transactions, Journals, and Letters may be submitted electronically on IEEE's on-line manuscript submission and peer-review system, ScholarOne® Manuscripts. You can get a listing of the publications that participate in ScholarOne at http://www.ieee.org/publications_standards/publications/authors/authors_submission.html. First check if you have an existing account. If there is none, please create a new account. After logging in, go to your Author Center and click "Submit First Draft of a New Manuscript."

Along with other information, you will be asked to select the subject from a pull-down list. Depending on the journal, there are various steps to the submission process; you must complete all steps for a complete submission. At the end of each step you must click "Save and Continue"; just uploading the paper is not sufficient. After the last step, you should see a confirmation that the submission is complete. You should also receive an e-mail confirmation. For inquiries regarding the submission of your paper on ScholarOne Manuscripts, please contact oprs-support@ieee.org or call +1 732 465 5861.

ScholarOne Manuscripts will accept files for review in various formats. Please check the guidelines of the specific journal for which you plan to submit.

You will be asked to file an electronic copyright form immediately upon completing the submission process (authors are responsible for obtaining any security clearances). Failure to submit the electronic copyright could result in publishing delays later. You will also have the opportunity to designate your article as "open access" if you agree to pay the IEEE open access fee.

C. FINAL STAGE USING SCHOLARONE MANUSCRIPTS

Upon acceptance, you will receive an email with specific instructions regarding the submission of your final files. To avoid any delays in publication, please be sure to follow these instructions. Most journals require that final submissions be uploaded through ScholarOne Manuscripts, although some may still accept final submissions via email. Final submissions should include source files of your accepted manuscript, high quality graphic files, and a formatted pdf file. If you have any questions regarding the final submission process, please contact the administrative contact for the journal.

In addition to this, upload a file with complete contact information for all authors. Include full mailing addresses, telephone numbers, fax numbers, and e-mail addresses. Designate the author who submitted the manuscript on ScholarOne Manuscripts as the "corresponding author." This is the only author to whom proofs of the paper will be sent.

D. COPYRIGHT FORM

Authors must submit an electronic IEEE Copyright Form (eCF) upon submitting their final manuscript files. You can access the eCF system through your manuscript submission system or through the Author Gateway. You are responsible for obtaining any necessary approvals and/or security clearances. For additional information on intellectual property rights, visit the IEEE Intellectual Property Rights department web page at http://www.ieee.org/publications_standards/publications/rights/index.html.

APPENDIX B IEEE PUBLISHING POLICY

The general IEEE policy requires that authors should only submit original work that has neither appeared elsewhere for publication, nor is under review for another refereed publication. The submitting author must disclose all prior publication(s) and current submissions when submitting a manuscript. Do not publish "preliminary" data or results. The submitting author is responsible for obtaining agreement of all coauthors and any consent required from employers or sponsors before submitting an article. The IEEE Access Department strongly discourages courtesy authorship; it is the obligation of the authors to cite only relevant prior work.

The IEEE Access Department does not publish conference records or proceedings, but can publish articles related to conferences that have undergone rigorous peer review. Minimally, two reviews are required for every article submitted for peer review.

APPENDIX C PUBLICATION PRINCIPLES

The two types of contents of that are published are; 1) peer-reviewed and 2) archival. The Access Department publishes scholarly articles of archival value as well as tutorial expositions and critical reviews of classical subjects and topics of current interest.

Authors should consider the following points:

1) Technical papers submitted for publication must advance the state of knowledge and must cite relevant prior work.
2) The length of a submitted paper should be commensurate with the importance, or appropriate to the complexity, of the work. For example, an obvious extension of previously published work might not be appropriate for publication or might be adequately treated in just a few pages.
3) Authors must convince both peer reviewers and the editors of the scientific and technical merit of a paper;

the standards of proof are higher when extraordinary or unexpected results are reported.

4) Because replication is required for scientific progress, papers submitted for publication must provide sufficient information to allow readers to perform similar experiments or calculations and use the reported results. Although not everything need be disclosed, a paper must contain new, useable, and fully described information. For example, a specimen's chemical composition need not be reported if the main purpose of a paper is to introduce a new measurement technique. Authors should expect to be challenged by reviewers if the results are not supported by adequate data and critical details.
5) Papers that describe ongoing work or announce the latest technical achievement, which are suitable for presentation at a professional conference, may not be appropriate for publication.

APPENDIX D REFERENCE EXAMPLES

- Basic format for books:
J. K. Author, "Title of chapter in the book," in Title of His Published Book, xth ed. City of Publisher, (only U.S. State), Country: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx–xxx.
See [**?**], [**?**].
- Basic format for periodicals:
J. K. Author, "Name of paper," Abbrev. Title of Periodical, vol. x, no. x,pp. xxx–xxx, Abbrev. Month, year, DOI. 10.1109.XXX.123456.
See [**?**]– [**?**].
- Basic format for reports:
J. K. Author, "Title of report," Abbrev. Name of Co., City of Co., Abbrev. State, Country, Rep. xxx, year.
See [**?**], [**?**].
- Basic format for handbooks:
Name of Manual/Handbook, x ed., Abbrev. Name of Co., City of Co., Abbrev. State, Country, year, pp. xxx–xxx.
See [**?**], [**?**].
- Basic format for books (when available online):
J. K. Author, "Title of chapter in the book," in Title of Published Book, xth ed. City of Publisher, State, Country: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx–xxx. [Online]. Available: http://www.web.com
See [**?**]– [**?**].
- Basic format for journals (when available online):
J. K. Author, "Name of paper," Abbrev. Title of Periodical, vol. x, no. x, pp. xxx–xxx, Abbrev. Month, year. Accessed on: Month, Day, year, DOI: 10.1109.XXX.123456, [Online].
See [**?**]– [**?**].
- Basic format for papers presented at conferences (when available online):
J.K. Author. (year, month). Title. presented at abbrev. conference title. [Type of Medium]. Available: site/path/file
See [**?**].
- Basic format for reports and handbooks (when available online):
J. K. Author. "Title of report," Company. City, State, Country. Rep. no., (optional: vol./issue), Date. [Online] Available: site/path/file
See [**?**], [**?**].
- Basic format for computer programs and electronic documents (when available online):
Legislative body. Number of Congress, Session. (year, month day). Number of bill or resolution, Title. [Type of medium]. Available: site/path/file
***NOTE:* ISO recommends that capitalization follow the accepted practice for the language or script in which the information is given.**
See [**?**].
- Basic format for patents (when available online):
Name of the invention, by inventor's name. (year, month day). Patent Number [Type of medium]. Available: site/path/file
See [**?**].
- Basic formatfor conference proceedings (published):
J. K. Author, "Title of paper," in Abbreviated Name of Conf., City of Conf., Abbrev. State (if given), Country, year, pp. xxxxxx.
See [**?**].
- Example for papers presented at conferences (unpublished):
See [**?**].
- Basic format for patents:
J. K. Author, "Title of patent," U.S. Patent x xxx xxx, Abbrev. Month, day, year.
See [**?**].
- Basic format for theses (M.S.) and dissertations (Ph.D.):
  1) J. K. Author, "Title of thesis," M.S. thesis, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.
  2) J. K. Author, "Title of dissertation," Ph.D. dissertation, Abbrev. Dept., Abbrev. Univ., City of Univ., Abbrev. State, year.

  See [**?**], [**?**].
- Basic format for the most common types of unpublished references:
  1) J. K. Author, private communication, Abbrev. Month, year.
  2) J. K. Author, "Title of paper," unpublished.
  3) J. K. Author, "Title of paper," to be published.

  See [**?**]– [**?**].
- Basic formats for standards:
  1) Title of Standard, Standard number, date.
  2) Title of Standard, Standard number, Corporate author, location, date.

  See [**?**], [**?**].

- Article number in reference examples:
  See [?], [?].
- Example when using et al.:
  See [?].

REFERENCES

[1] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss, “Near-optimal sparse fourier representations via sampling,” in Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, MontrÃl’al, QuÃl’bec, Canada, 2002.

[2] M. A. Iwen, A. Gilbert, Strauss, and M, “Empirical evaluation of a sub-linear time sparse dft algorithm,” Communications in Mathematical Sciences, vol. 5, no. 4, pp. 981–998, 2007.

[3] A. C. Gilbert, M. J. Strauss, and J. A. Tropp, “A tutorial on fast fourier sampling,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 57–66, 2008.

[4] S. Pawar and K. Ramchandran, “Computing a k-sparse n-length Discrete Fourier Transform using at most 4k samples and O(k log k) complexity,” IEEE International Symposium on Information Theory - Proceedings, pp. 464–468, 2013.

[5] ——, “FFAST: An algorithm for computing an exactly k-Sparse DFT in O(k log k) time,” IEEE Transactions on Information Theory, vol. 64, no. 1, pp. 429–450, jan 2018.

[6] ——, “A robust sub-linear time R-FFAST algorithm for computing a sparse DFT,” pp. 1–35, 2015. [Online]. Available: http://arxiv.org/abs/1501.00320

[7] F. Ong, R. Heckel, and K. Ramchandran, “A Fast and Robust Paradigm for Fourier Compressed Sensing Based on Coded Sampling,” ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, vol. 2019-May, pp. 5117–5121, 2019.

[8] S. H. Hsieh, C. S. Lu, and S. C. Pei, “Sparse Fast Fourier Transform by downsampling,” ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, pp. 5637–5641, 2013.

[9] M. A. Iwen, “Combinatorial sublinear-time Fourier algorithms,” Foundations of Computational Mathematics, vol. 10, no. 3, pp. 303–338, 2010.

[10] ——, “Improved approximation guarantees for sublinear-time Fourier algorithms,” Applied and Computational Harmonic Analysis, vol. 34, no. 1, pp. 57–82, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.acha.2012.03.007

[11] A. Christlieb, D. Lawlor, and Y. Wang, “A multiscale sub-linear time Fourier algorithm for noisy data,” Applied and Computational Harmonic Analysis, vol. 40, no. 3, pp. 553–574, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.acha.2015.04.002

[12] D. LAWLOR, Y. WANG, and A. CHRISTLIEB, “ADAPTIVE SUB-LINEAR TIME FOURIER ALGORITHMS,” Advances in Adaptive Data Analysis, 2013.

[13] S. Merhi, R. Zhang, M. A. Iwen, and A. Christlieb, “A New Class of Fully Discrete Sparse Fourier Transforms: Faster Stable Implementations with Guarantees,” Journal of Fourier Analysis and Applications, vol. 25, no. 3, pp. 751–784, 2019.

[14] G. Plonka, K. Wannenwetsch, A. Cuyt, and W. shin Lee, “Deterministic sparse FFT for M-sparse vectors,” Numerical Algorithms, 2018.

[15] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, “Simple and practical algorithm for sparse fourier transform,” Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1183–1194, 2012.

[16] ——, “Nearly optimal sparse fourier transform,” Proceedings of the Annual ACM Symposium on Theory of Computing, pp. 563–577, 2012.

[17] J. Chiu, “Matrix probing Fourier Transform,” Encyclopedia of Thermal Stresses, pp. 1742–1769, 2014.

[18] A. C. Gilbert, P. Indyk, M. Iwen, and L. Schmidt, “Recent Developments in the Sparse Fourier Transform,” IEEE Signal Processing Magazine, pp. 1–21, 2014.

[19] M. Kapralov, “Sample efficient estimation and recovery in sparse FFT via isolation on average,” Annual Symposium on Foundations of Computer Science - Proceedings, vol. 2017-Octob, no. 1, pp. 651–662, 2017.

[20] P. Indyk, M. Kapralov, and E. Price, “(Nearly) sample-optimal sparse fourier transform,” Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 480–499, 2014.

[21] A. López-Parrado and J. Velasco Medina, “Efficient Software Implementation of the Nearly Optimal Sparse Fast Fourier Transform for the Noisy Case,” Ingeniería y Ciencia, vol. 11, no. 22, pp. 73–94, 2015.

[22] G. L. Chen, S. H. Tsai, and K. J. Yang, “On Performance of Sparse Fast Fourier Transform and Enhancement Algorithm,” IEEE Transactions on Signal Processing, vol. 65, no. 21, pp. 5716–5729, 2017.

[23] J. Schumacher and M. Püschel, “High-performance sparse fast Fourier transforms,” IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation, 2014.

[24] C. Wang, “CusFFT: A High-Performance Sparse Fast Fourier Transform Algorithm on GPUs,” Proceedings - 2016 IEEE 30th International Parallel and Distributed Processing Symposium, IPDPS 2016, pp. 963–972, 2016.

[25] S. Wang, V. M. Patel, and A. Petropulu, “Multidimensional Sparse Fourier Transform Based on the Fourier Projection-Slice Theorem,” IEEE Transactions on Signal Processing, vol. 67, no. 1, pp. 54–69, 2019.

[26] M. Kapralov, A. Velingker, and A. Zandieh, “Dimension-independent sparse fourier transform,” in Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, 2019.

[27] C. Pang, S. Liu, and Y. Han, “High-speed target detection algorithm based on sparse fourier transform,” IEEE Access, vol. 6, pp. 37 828–37 836, jul 2018.

[28] O. Abari, E. Hamed, H. Hassanieh, A. Agarwal, D. Katabi, A. P. Chandrakasan, and V. Stojanovic, “A 0.75-million-point fourier-transform chip for frequency-sparse signals,” in Digest of Technical Papers - IEEE International Solid-State Circuits Conference, 2014.

[29] G. Plonka and K. Wannenwetsch, “A deterministic sparse FFT algorithm for vectors with small support,” Numerical Algorithms, vol. 71, no. 4, pp. 889–905, 2016.

[30] ——, “A sparse fast Fourier algorithm for real non-negative vectors,” Journal of Computational and Applied Mathematics, vol. 321, pp. 532–539, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.cam.2017.03.019

FIRST A. AUTHOR (M’76–SM’81–F’87) and all authors may include biographies. Biographies are often not included in conference-related papers. This author became a Member (M) of IEEE in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author’s educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state, and country, and year the degree was earned. The author’s major field of study should be lower-cased.

The second paragraph uses the pronoun of the person (he or she) and not the author’s last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (publisher name, year) similar to a reference. Current and previous research interests end the paragraph. The third paragraph begins with the author’s title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies other than the IEEE. Finally, list any awards and work for IEEE committees and publications. If a photograph is provided, it should be of good quality, and professional-looking. Following are two examples of an author’s biography.

SECOND B. AUTHOR was born in Greenwich Village, New York, NY, USA in 1977. He received the B.S. and M.S. degrees in aerospace engineering from the University of Virginia, Charlottesville, in 2001 and the Ph.D. degree in mechanical engineering from Drexel University, Philadelphia, PA, in 2008.

From 2001 to 2004, he was a Research Assistant with the Princeton Plasma Physics Laboratory. Since 2009, he has been an Assistant Professor with the Mechanical Engineering Department, Texas A&M University, College Station. He is the author of three books, more than 150 articles, and more than 70 inventions. His research interests include high-pressure and high-density nonthermal plasma discharge processes and applications, microscale plasma discharges, discharges in liquids, spectroscopic diagnostics, plasma propulsion, and innovation plasma applications. He is an Associate Editor of the journal Earth, Moon, Planets, and holds two patents.

Dr. Author was a recipient of the International Association of Geomagnetism and Aeronomy Young Scientist Award for Excellence in 2008, and the IEEE Electromagnetic Compatibility Society Best Symposium Paper Award in 2011.

THIRD C. AUTHOR, JR. (M'87) received the B.S. degree in mechanical engineering from National Chung Cheng University, Chiayi, Taiwan, in 2004 and the M.S. degree in mechanical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2006. He is currently pursuing the Ph.D. degree in mechanical engineering at Texas A&M University, College Station, TX, USA.

From 2008 to 2009, he was a Research Assistant with the Institute of Physics, Academia Sinica, Tapei, Taiwan. His research interest includes the development of surface processing and biological/medical treatment techniques using nonthermal atmospheric pressure plasmas, fundamental study of plasma sources, and fabrication of micro- or nanostructured surfaces.

Mr. Author's awards and honors include the Frew Fellowship (Australian Academy of Science), the I. I. Rabi Prize (APS), the European Frequency and Time Forum Award, the Carl Zeiss Research Award, the William F. Meggers Award and the Adolph Lomb Medal (OSA).