

# (Rev) Multidimensional Sparse Fourier Transform Based on the Fourier Projection-Slice Theorem

Shaogang Wang, *Student Member, IEEE*, Vishal M. Patel, *Senior Member, IEEE*, and  
Athina Petropulu, *Fellow, IEEE*

**Abstract**—We propose FPS-SFT, a sparse Fourier transform for multidimensional, frequency-domain sparse signals, inspired by the idea of the Fourier projection-slice theorem. FPS-SFT identifies frequencies by operating on one-dimensional slices of the discrete-time domain data, taken along specially designed lines; those lines are parametrized by slopes that are randomly generated from a set at runtime. The DFTs of the data slices represent DFT projections onto the lines along which the slices were taken. When the lines’ lengths and slopes are properly designed so that they allow for orthogonal and uniform frequency projections, the multidimensional frequencies can be recovered from their projections with low sample and computational complexity. We show theoretically that the large number of degrees of freedom in frequency projections allows FPS-SFT to be applicable in recovery of less sparse, uniformly distributed frequencies; various numerical results suggest that FPS-SFT is also efficient for clustered frequencies. We also extend FPS-SFT into a robust version (RFPS-SFT), to address noisy signals that contain off-grid frequencies. The advantages of RFPS-SFT are demonstrated via simulations in the context of digital beamforming automotive radar signal processing, where the RFPS-SFT can be used to identify range, velocity and angular parameters of targets with low sample and computational complexity.

**Index Terms**—Multidimensional signal processing, sparse Fourier transform, Fourier projection-slice theorem, automotive radar signal processing.

## I. INTRODUCTION

Conventional signal processing methods in radar, sonar, and medical imaging systems usually involve multidimensional discrete Fourier transforms (DFT), which are typically implemented by the fast Fourier transform (FFT). The sample complexity of the FFT is  $O(N)$ , where  $N$  is the number of samples in the multidimensional sample space. For  $N$  equal to a power of 2, the computational complexity of the FFT is  $O(N \log N)$ . Recently, by leveraging the sparsity of signals in the frequency domain, the sparse Fourier transform (SFT) has been proposed [1]–[8]; this is a family of low-complexity DFT algorithms. The state-of-the-art SFT algorithms [5], [7] achieve sample complexity of  $O(K)$  and computational complexity of  $O(K \log K)$  for exactly  $K$ -sparse (in the frequency domain) signals. When  $K \ll N$ , those SFT algorithms achieve significant savings both in sample and computation compared to the FFT. SFT algorithms have been investigated for several applications including a fast Global Positioning

System (GPS) receiver, wide-band spectrum sensing, multidimensional radar signal processing [9]–[13].

In all SFT algorithms, the reduction of sample and computational complexity is achieved by reducing the input data samples. This is implemented via a well designed, randomized subsampling procedure, which leverages the resulting frequency domain aliasing. The significant frequencies contained in the original signal are then localized and the corresponding DFT coefficients are estimated with low-complexity operations. Such *subsampling-localization-estimation* procedure is carried out in an *iterative* manner in several SFT algorithms [1], [2], [5], [7], while in other SFT algorithms [3], [11]–[13], localization and estimation are implemented in *one-shot* after gathering sufficient copies of subsampled signals corresponding to different subsampling parameters, e.g., subsample rate, offset and number of samples. Generally, iterative based SFT algorithms exhibit lower complexity as compared to one-shot based SFT algorithms, since in the former, in each iteration, the contribution of the recovered frequencies are removed from the signal, which yields a sparser signal (an easier problem) in the next iteration.

Multidimensional signal processing requires multidimensional SFT algorithms. Most of the existing SFT algorithms, however, are designed for one-dimensional (1-D) signals and their extension to multidimensional signals is typically not straightforward. This is because the SFT algorithms are not separable in each dimension due to the fact that operations such as detection of significant frequencies in the subsampled signal within an SFT algorithm must be considered jointly for all the dimensions [11]. Multidimensional SFT algorithms are investigated in [5], [12], [13]. The sample-optimal SFT (SO-SFT) of [5] follows the subsampling-localization-estimation iteration, while the SFT algorithms of [12], [13] are one-shot approaches. SO-SFT achieves the sample and computational complexity lower bounds of all known SFT algorithms by reducing a 2-dimensional (2-D) DFT into 1-D DFTs along a few columns and rows of a data matrix; in the frequency domain, this results into projections of 2-D frequencies onto the corresponding columns and rows of the matrix. Under the assumption that the frequencies are sparse and uniformly distributed, the 2-D frequencies are not likely to be projected to the same bin (we will refer this as collision), and thus the 2-D frequencies can be effectively recovered from their 1-D projections (see Section III-A for details). The DFT along columns and rows provides two degrees of freedom; a frequency collision has low probability to occur both in the column and row direction. However, when frequencies are less

Shaogang Wang, Vishal M. Patel and Athina Petropulu are with the department of Electrical and Computer Engineering at Rutgers University, Piscataway, NJ USA. email: {shaogang.wang, vishal.m.patel, athinap}@rutgers.edu. The work was supported by ARO grant W911NF-16-1-0126, NSF Grant ECCS-1408437, China Scholarship Council and Shanghai Institute of Spaceflight Electronics Technology.

sparse, or when they are clustered, there is a high probability that a set of frequencies will collide both in row and column directions; this is referred to as ‘deadlock’ situation [5] and results in unrecoverable frequencies (see Fig. 1).

To reduce the probability of a deadlock, the SFT of [12], [13] introduces more degrees of freedom in projections by applying 1-D DFT to data samples extracted along some lines of predefined and deterministic slopes as well as along the axes of the data cube. However, the limited choice of line slopes in [12], [13] is still an obstacle in addressing less sparse signals. Moreover, the sample and computational complexity of [12], [13] are higher than that of SO-SFT, as the former applies the one-shot approach for frequency recovery, while the latter recovers the frequencies iteratively. In addition to the iterative approach, the low-cost frequency localization technique adopted in SO-SFT further contributes to the low-complexity of the algorithm. Specifically, SO-SFT applies the OFDM-trick (phase encoding) [2], [4], which effectively encodes the significant frequencies into the phase difference of a pair of DFTs applied on two datasets, obtained by subsampling the data with the same subsample rate but different offsets. In the exactly sparse case, the encoded frequencies can be decoded trivially with a low-complexity ( $O(1)$ ) operation (see Section III-A for details).

In this work we propose FPS-SFT, a multidimensional, Fourier projection-slice based SFT, which enjoys low complexity while avoiding the limitations of the aforementioned algorithms, i.e., it can handle less sparse data in the frequency domain and accommodate frequencies that are non-uniformly distributed. FPS-SFT uses the low-complexity frequency localization framework of SO-SFT and extends the multiple slopes idea of [12], [13] by using lines parameterized by slopes that are randomly generated from a set of sufficiently large support at runtime. This is not trivial since the line length and slope set should be carefully designed to enable an orthogonal and uniform frequency projection.

FPS-SFT can be viewed as a low-complexity, Fourier projection-slice approach for signals that are sparse in the frequency domain. In FPS-SFT, the DFT of an 1-D slice of the data is the projection of the  $D$ -D DFT of the data on that same line along which the time-domain slice was taken. The classical Fourier projection-slice based method either reconstructs the frequency domain signal via interpolation of frequency-domain slices or reconstructs the time-domain samples by solving a system of linear equations relating the DFT along projections and the time-domain samples. Different from the Fourier projection-slice based methods, the proposed FPS-SFT aims to reconstruct the signal directly based on frequency domain projections with low-complexity operations; this is achieved by leveraging the sparsity of the signal in the frequency domain.

Another body of related works, referred to as SFT based on rank-1 lattice sampling [6], [14], [15] also consider the problem of fast reconstruction of the underlying multidimensional signal based on samples along rank-1 lattices, i.e., lines. In [14], [15], the coefficients of the multidimensional DFT of the data can be efficiently calculated by applying DFT on samples along suitable lines, provided that the frequencies are

known. In particular, in [14], the exact evaluation of the DFT coefficients can be accomplished by calculating the DFT along a single line; such a line is called the reconstructing rank-1 lattices and can be found for any given sparse frequency set [14]. However, finding a reconstructing rank-1 lattice is non-trivial when the frequency set is unknown. That unknown frequency case is addressed in [6], at the expense of high complexity due to the complication of finding a reconstructing rank-1 lattice. Compared with the algorithms of [14], [15], the proposed FPS-SFT does not assume that the underlying frequency set is known. The frequency set as well as the corresponding DFT coefficients are estimated via DFT along lines progressively. Compared with the SFT of [6], FPS-SFT is based on multiple lines of randomized parameters and does not pursue to reconstruct the signal using a single line, which avoids the complication of locating a reconstructing rank-1 lattice and thus achieves a low complexity. In addition, the rank-1 lattice-based SFT algorithms assume that samples of the signal can be obtained at any arbitrary location, which is rather difficult to achieve in hardware [16]. In contrast, the FPS-SFT assumes that the samples are extracted from a predefined uniform sampling grid. Hence, FPS-SFT is less restrictive in sampling and more applicable to existing systems, which employ uniform sampling in each dimension.

While the FPS-SFT considers the ideal scenario, i.e., frequency-domain sparse data containing frequencies on the DFT grid, in realistic applications, the data is usually noisy and contains off-grid frequencies. One example of such data is the received signal of the digital beamforming (DBF) automotive radar, which usually employs a frequency modulation continuous waveform (FMCW). After demodulation of the returned signal, each radar target can be expressed as a  $D$ -D complex sinusoid [17], whose frequency in each dimension is related to target parameters, e.g., range, Doppler and direction of arrival. When the number of targets is much smaller than the number of samples, such signal is sparse in the  $D$ -D frequency domain. Due to the continuous nature of those parameters, the frequencies are also continuous and thus are typically off-grid. Meanwhile, the radar signal contains noise, which makes the signal approximately sparse. FPS-SFT suffers from the frequency leakage caused by the off-grid frequencies; also, the frequency localization procedure of FPS-SFT is prone to errors since the OFDM-trick is sensitive to noise [2]. We address these issues by extending FPS-SFT to a robust version, which we call RFPS-SFT. RFPS-SFT employs a windowing technique to reduce the frequency leakage caused by the off-grid frequencies and a voting based frequency decoding procedure to significantly reduce the localization error stemming from noise.

The off-grid frequencies are also addressed in [11], where a robust multidimensional SFT algorithm, termed RSFT, is proposed. In RSFT, the computational savings are achieved by folding the input  $D$ -D data cube into a much smaller data cube, on which a reduced sized  $D$ -D FFT is applied. Although the RSFT is more computationally efficient as compared to the FFT-based methods, its sample complexity is the same as the FFT-based algorithms. Essentially, the high sample complexity of RSFT is due to its two stages of windowing procedures,

which are applied to the entire data cube to suppress the frequency leakage. In the proposed RFPS-SFT, instead of applying the multidimensional window on the entire data, the window, while still designed for the full-sized data, is applied on samples along lines only, which does not cause overhead in sample complexity.

This paper makes the following contributions.

- We propose FPS-SFT, a multidimensional, low-complexity SFT algorithm that is based on the Fourier projection-slice theorem. Compared to the SFT algorithms of [5], [12], [13] that project multidimensional DFT of data onto deterministic lines, the frequency-domain projections in FPS-SFT are randomized. This enables good frequency recovery performance in less sparse data scenarios and even in scenarios in which the frequencies are clustered. Also, while the SFT of [5], [12], [13] requires the data to be equal-sized in each dimension, FPS-SFT applies to arbitrary-sized data, which is less restrictive.
- We extend FPS-SFT to a robust version, termed RFPS-SFT. RFPS-SFT is a practical extension of FPS-SFT to address noisy data containing off-grid frequencies arising from applications such as DBF automotive radar signal processing. The feasibility of applying RFPS-SFT in such application is demonstrated via simulations.

Preliminary version of FPS-SFT and RFPS-SFT appeared in [18] and [19], respectively. This paper extends [18], [19] by providing detailed analysis and extensive numerical results. [The MATLAB code of the proposed algorithms can be found at https://github.com/iamwsg/FPS-SFT.git](https://github.com/iamwsg/FPS-SFT.git).

**Notation:** We use lower-case (upper-case) bold letters to denote vectors (matrices).  $[\cdot]^T$  denotes transpose. The  $N$ -modulo operation is denoted by  $[\cdot]_N$ .  $[S]$  refers to the integer set of  $\{0, \dots, S-1\}$ . The cardinality of set  $\mathbb{S}$  is denoted as  $|\mathbb{S}|$ . The  $N$ -point DFT of signal  $x$ , normalized by  $N$ , is denoted by  $\hat{x}$ ; all the DFT is referred to as the normalized version.  $\|\mathbf{W}\|_1, \|\mathbf{W}\|_2$  are the  $l_1$  and  $l_2$  norm of matrix  $\mathbf{W}$ , respectively. We denote the least common multiple of  $N_0, N_1$  as  $\text{LCM}(N_0, N_1)$ .

The paper is organized as follows. The signal model is provided in Section II. The proposed FPS-SFT is described in Section III and its robust extension is in Section IV. Validation of theoretical results via simulations is provide in Section V and the application of RFPS-SFT in automotive DBF radar signal processing is presented in Section VI. Concluding remarks are made in Section VII.

## II. SIGNAL MODEL AND PROBLEM FORMULATION

For simplicity, in this section, we will present the ideas for 2-D signals. The generalization to higher dimensions, i.e.,  $D$ -D cases with  $D > 2$ , is straightforward.

Let us consider the following 2-D signal model, which is a superposition of  $K$  2-D complex sinusoids, i.e.,

$$x(\mathbf{n}) \triangleq \sum_{(a, \boldsymbol{\omega}) \in \mathbb{S}} a e^{j\mathbf{n}^T \boldsymbol{\omega}}, \quad \mathbf{n} \triangleq [n_0, n_1]^T \in \mathcal{X} \triangleq [N_0] \times [N_1], \quad (1)$$

where  $N_0, N_1$  denote the sample length of the two dimensions, respectively.  $(a, \boldsymbol{\omega})$  represents a 2-D frequency whose amplitude is  $a$  with  $a \in \mathbb{C}, a \neq 0$  and frequency is  $\boldsymbol{\omega} \triangleq [\omega_0, \omega_1]^T \triangleq [2\pi m_0/N_0, 2\pi m_1/N_1]^T$  with  $[m_0, m_1]^T \in \mathcal{X}$ . The set  $\mathbb{S}$  with  $|\mathbb{S}| = K$  includes all the 2-D frequencies. We assume that the signal is sparse in the frequency domain, i.e.,  $K \ll N \triangleq N_0 N_1$ .

In the above model, the frequencies lie on a grid. A more realistic signal model, addressing continuous-valued frequencies in  $[0, 2\pi)^2$  and also noise, is the following

$$r(\mathbf{n}) = y(\mathbf{n}) + n(\mathbf{n}) = \sum_{(a, \boldsymbol{\omega}) \in \mathbb{S}'} a e^{j\mathbf{n}^T \boldsymbol{\omega}} + n(\mathbf{n}), \quad \mathbf{n} \in \mathcal{X}, \quad (2)$$

where  $y(\mathbf{n}) \triangleq \sum_{(a, \boldsymbol{\omega}) \in \mathbb{S}'} a e^{j\mathbf{n}^T \boldsymbol{\omega}}$  is the superposition of  $K' = |\mathbb{S}'|$  continuous-frequency sinusoids;  $(a, \boldsymbol{\omega})$  denotes a significant 2-D frequency in  $\mathbb{S}'$ , whose complex amplitude and frequency are  $a, \boldsymbol{\omega} \triangleq [\omega_0, \omega_1]^T \in [0, 2\pi)^2$ , respectively, and it holds that  $0 < a_{\min} \leq |a| \leq a_{\max}$ . The noise,  $n(\mathbf{n})$ , is assumed to be i.i.d., circularly symmetric Gaussian, i.e.,  $\mathcal{CN}(0, \sigma_n)$ . The SNR of a sinusoid with amplitude  $a$  is defined as  $\text{SNR} \triangleq (|a|/\sigma_n)^2$ .

Conventionally,  $\mathbb{S}'$  can be estimated via a 2-D  $N_0 \times N_1$ -point DFT applied on signal (2), after windowing the signal with a 2-D window  $w(\mathbf{n})$ , used to suppress frequency leakage generated by off-grid frequencies. Assuming that the peak to side-lobe ratio (PSR) of the window is large enough, such that the side-lobes of each frequency in  $\mathbb{S}'$  can be neglected in the DFT domain, the signal contribution in the DFT domain can be viewed as a set of on-grid frequencies, i.e.,  $\mathbb{S} \triangleq \{(a, \boldsymbol{\omega}) : \boldsymbol{\omega} \triangleq [2\pi m_0/N_0, 2\pi m_1/N_1]^T, [m_0, m_1]^T \in \mathcal{X}\}$  with  $K' < K = |\mathbb{S}| \ll N$ . Hence, the sample domain signal component associated with the window  $w(\mathbf{n})$  and  $\mathbb{S}$  can be approximated by (1). Note that since the windowing degrades the frequency resolution, each continuous-valued frequency in  $\mathbb{S}'$  is related to a cluster of digital frequencies in  $\mathbb{S}$ ;  $\mathbb{S}$  can be estimated from the DFT of the signal, and then lead to the frequencies in  $\mathbb{S}'$  via quadratic interpolation [20].

## III. THE PROPOSED FPS-SFT ALGORITHM

The proposed FPS-SFT is a generalization of SO-SFT. In this chapter, we first introduce the basics of SO-SFT, then, the details of FPS-SFT is explained, followed by the analysis of its properties.

### A. SO-SFT

In SO-SFT [5], in order to recover  $\mathbb{S}$  of (1), 1-D DFTs are applied on a subset of columns and rows of the data matrix. The  $N_0$ -point DFT of the  $i$ -th,  $i \in [N_1]$  column of the data equals

$$\begin{aligned} \hat{c}_i(m) &\triangleq \frac{1}{N_0} \sum_{l \in [N_0]} x(l, i) e^{-j \frac{2\pi}{N_0} ml} \\ &= \frac{1}{N_0} \sum_{(a, \boldsymbol{\omega}) \in \mathbb{S}} \sum_{l \in [N_0]} a e^{j \frac{2\pi}{N_1} m_1 i} e^{j \frac{2\pi}{N_0} l(m_0 - m)} \\ &= \sum_{(a, \boldsymbol{\omega}) \in \mathbb{S}} a e^{-j \frac{2\pi}{N_0} m_1 i} \delta(m - m_0), \quad m \in [N_0], \\ &\boldsymbol{\omega} = [2\pi m_0/N_0, 2\pi m_1/N_1]^T, [m_0, m_1]^T \in \mathcal{X}, \end{aligned} \quad (3)$$

where  $\delta(\cdot)$  is the unit impulse function. Hence  $\hat{c}_i(m_0)$  can be viewed as the summation of modulated amplitudes of 2-D frequencies whose row frequency indices equal to  $m_0$ . Hence  $\hat{c}_i(m)$ ,  $m \in [N_1]$  is a projection of the 2-D spectrum,  $\hat{x}(m_0, m_1)$ ,  $[m_0, m_1]^T \in \mathcal{X}$ , onto the column. Similarly, the  $N_1$ -point DFT applied on a row of (1) is a projection of the 2-D spectrum onto the row.

Since the signal is sparse in the frequency domain, if  $|\hat{c}_i(m)| \neq 0$ , with high probability, there will be only one significant frequency projected to the frequency bin of  $m$ ; in other words, the frequency bin is ‘1-sparse’, and  $\hat{c}_i(m)$  is reduced to  $\hat{c}_i(m) = \hat{c}_i(m_0) = ae^{j\frac{2\pi}{N_1}m_1i}$ . The amplitude,  $a$ , can be determined by the  $m_0$ -th entry of the DFT of the 0-th column, i.e.,  $a = \hat{c}_0(m_0)$ , and the other frequency component,  $m_1$ , is ‘coded’ in the phased difference between the  $m_0$ -th entries of the DFTs of the 0-th and the 1-st columns, which can be decoded by  $m_1 = \phi(\hat{c}_1(m_0)/\hat{c}_0(m_0)) \frac{N_1}{2\pi}$ , where  $\phi(x)$  is the phase of  $x$ . Note that the 1-sparsity of the  $m$ -th bin can be effectively tested by comparing  $|\hat{c}_0(m)|$  and  $|\hat{c}_1(m)|$ ;  $\hat{c}_i(m)$  is 1-sparse almost for sure when  $|\hat{c}_0(m)| = |\hat{c}_1(m)|$ . Such frequency decoding technique is referred to as OFDM-trick [4]. The contribution of the recovered 2-D frequencies is removed from the signal, so that the following processing can be applied on a sparser signal, which is easier to solve in the subsequent processing.

A frequency bin that is not 1-sparse based on column processing might be 1-sparse based on row processing. Because the removal of frequencies in the column (row) processing may cause bins in the row (column) processing to be 1-sparse, SO-SFT runs iteratively and alternatively between columns and rows and the algorithm stops after a finite number of iterations. SO-SFT succeeds with high probability only when the frequencies are very sparse, and requires that either a row or a column of the DFT contains a 1-sparse bin. However, in many applications, the signal frequency exhibits a block sparsity pattern [21], i.e., the significant frequencies are clustered. In those cases, even when the signal is very sparse, 1-sparse bin may not exist; this is referred to as a ‘deadlock’ case [5].

## B. FPS-SFT

SO-SFT reduces a 2-D DFT into 1-D DFTs of the columns and rows of the input data matrix. The columns and the rows can be viewed as 1-D slices taken along discrete lines with slopes  $\infty$  and 0, respectively. In this section, by proposing FPS-SFT, we reduce the 2-D DFT into 1-D DFTs of the data slices taken along discrete lines with random slopes.

FPS-SFT is an iterative algorithm; each iteration returns a subset of recovered 2-D frequencies. After  $T$  iterations, the FPS-SFT returns a set,  $\hat{\mathbb{S}}$ , which is an estimate of  $\mathbb{S}$  of (1). The frequencies recovered in previous iterations are passed to the next iteration, and their contributions are removed from the signal in order to create a sparser signal.

Within each iteration of FPS-SFT, the signal of (1) is sampled along a line,  $\mathcal{E}(\alpha, \tau, l)$ ,  $l \in [L]$ , with  $\alpha \triangleq [\alpha_0, \alpha_1]^T$ ,  $\tau \triangleq [\tau_0, \tau_1]^T$ ,  $\alpha, \tau \in \mathcal{X}$ , satisfying the following equations

$$[\alpha_0 l + \tau_0]_{N_0} = n_0, [\alpha_1 l + \tau_1]_{N_1} = n_1, l \in [L]. \quad (4)$$

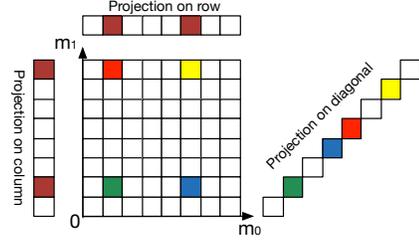


Fig. 1. Demonstration of projection of 2-D frequencies onto 1-D. The colored blocks mark significant frequencies. The projection onto the column or the row causes collisions, while the projection onto the diagonal creates 1-sparse bins.

where

$$n_1 = \left[ \frac{\alpha_1}{\alpha_0} (n_0 - \tau_0) + \tau_1 \right]_{N_1}. \quad (5)$$

Hence,  $\mathcal{E}(\alpha, \tau, l)$ ,  $l \in [L]$  is a discrete line segment whose slope is  $\alpha_1/\alpha_0$  and offset is  $\tau$ .

The sampled signal, representing a slice of the data along  $\mathcal{E}(\alpha, \tau, l)$ ,  $l \in [L]$ , can be expressed as

$$\begin{aligned} s(\alpha, \tau, l) &\triangleq x([\alpha_0 l + \tau_0]_{N_0}, [\alpha_1 l + \tau_1]_{N_1}) \\ &= \sum_{(a, \omega) \in \mathbb{S}} a e^{j2\pi \left( \frac{m_0[\alpha_0 l + \tau_0]_{N_0}}{N_0} + \frac{m_1[\alpha_1 l + \tau_1]_{N_1}}{N_1} \right)}, l \in [L]. \end{aligned} \quad (6)$$

Note that a slice can ‘wrap around’ within  $x(\mathbf{n})$ ,  $\mathbf{n} \in \mathcal{X}$  due to the modulo operation, and the sampling points along the line are always on the grid of  $\mathcal{X}$ , since  $\alpha, \tau$  are on grid.

Taking an  $L$ -point DFT of the data slice defined in (6), for  $m \in [L]$ , we get

$$\begin{aligned} \hat{s}(\alpha, \tau, m) &\triangleq \frac{1}{L} \sum_{l \in [L]} s(\alpha, \tau, l) e^{-j2\pi \frac{lm}{L}} \\ &= \frac{1}{L} \sum_{(a, \omega) \in \mathbb{S}} a e^{j2\pi \left( \frac{m_0 \tau_0}{N_0} + \frac{m_1 \tau_1}{N_1} \right)} \sum_{l \in [L]} e^{j2\pi l \left( \frac{m_0 \alpha_0}{N_0} + \frac{m_1 \alpha_1}{N_1} - \frac{m}{L} \right)}. \end{aligned} \quad (7)$$

Let us assume that for all  $m \in [L]$  and  $\alpha, [m_0, m_1]^T \in \mathcal{X}$ ,

$$\hat{f}(m) \triangleq \frac{1}{L} \sum_{l \in [L]} e^{j2\pi l \left( \frac{m_0 \alpha_0}{N_0} + \frac{m_1 \alpha_1}{N_1} - \frac{m}{L} \right)} \in \{0, 1\}. \quad (8)$$

This assumption holds when  $\frac{m_0 \alpha_0}{N_0} + \frac{m_1 \alpha_1}{N_1} - \frac{m}{L}$  is multiple of  $1/L$ , which can be expressed as

$$\left[ \frac{L}{N_0} m_0 \alpha_0 + \frac{L}{N_1} m_1 \alpha_1 \right]_L = m. \quad (9)$$

It is clear that  $L = \text{LCM}(N_0, N_1)$  satisfies (9), since  $L/N_0, L/N_1$  are integers. Moreover,  $\text{LCM}(N_0, N_1)$  is the minimum length of a line that satisfies (9) for arbitrary  $\alpha, [m_0, m_1]^T \in \mathcal{X}$ ; this can be proved using contradiction in the following.

Assume that  $L < \text{LCM}(N_0, N_1)$ , then at least either  $L/N_0$  or  $L/N_1$  is not an integer. Without loss of generality, we assume that  $\frac{L}{N_0} \notin \mathbb{Z}$ , then the right side of (9) equals  $[L/N_0]_L \notin [L]$  for  $m_0 = 1, \alpha_0 = 1, m_1 = 0$ ,

which is contradictory to the premise that (9) holds for any  $[m_0, m_1]^T, [\alpha_0, \alpha_1]^T \in \mathcal{X}$ .

When  $\hat{f}(m) = 1$ , i.e.,

$$\left[ \frac{m_0 \alpha_0}{N_0} + \frac{m_1 \alpha_1}{N_1} - \frac{m}{L} \right]_1 = 0, [m_0, m_1]^T \in \mathcal{X}, \quad (10)$$

(7) can be simplified as

$$\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m) = \sum_{(a, \omega) \in \mathbb{S}} a e^{j2\pi \left( \frac{m_0 \tau_0}{N_0} + \frac{m_1 \tau_1}{N_1} \right)}, \quad m \in [L], \quad (11)$$

which can be viewed as the 1-D projection of the 2-D frequencies satisfying (10). The solutions of (10) with respect to  $m$  are equally spaced points lie on line

$$\mathcal{E}([\alpha_1 L/N_1, -\alpha_0 L/N_0]^T, [m'_0, m'_1]^T, l), l \in [L'], \quad (12)$$

where  $[m'_0, m'_1]^T \in \mathcal{X}$  is one of the solutions of (10). The line of (12) is orthogonal to the line of (4). The orthogonality is necessary for the projected 2-D frequencies to be exactly recoverable. Moreover, for certain choices of  $\boldsymbol{\alpha}$ , such projection is uniform, and  $L' = N/L$ . The uniformity of the projection means that the DFT coefficients of  $N$  grid locations of the  $N_0 \times N_1$ -point DFT are uniformly projected to the  $L$  entries of the  $L$ -point DFT along a line. Compared with a non-uniform projection, the uniform projection creates more 1-sparse bins, which allows for fewer iterations of FPS-SFT to exactly reconstruct the signal. The condition for orthogonal and uniform projection is stated in the following lemma.

**Lemma 1. (Condition for orthogonal and uniform projection):** Consider the slice of the signal of (1), as defined in (6), with  $L = \text{LCM}(N_0, N_1)$ ,  $\boldsymbol{\alpha} \in \mathcal{A} \subset \mathcal{X}$ ,  $\boldsymbol{\tau} \in \mathcal{X}$  where  $\mathcal{A} \triangleq \{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \mathcal{X}; (\alpha_0, \alpha_1), (\alpha_0, L/N_1), (\alpha_1, L/N_0) \text{ are co-prime pairs}\}$ . Then each entry of (11) is the projection of samples of  $\hat{x}(\mathbf{m})$ ,  $\mathbf{m} \in \mathcal{P}_m \subset \mathcal{X}$ , where  $\mathcal{P}_m, m \in [L]$  contain sample locations satisfying (10). Moreover,  $|\mathcal{P}_m| = N/L$ ,  $\mathcal{P}_m \cap \mathcal{P}_{m'} = \emptyset$  for  $m \neq m', m, m' \in [L]$ . Thus,  $\hat{x}(\mathbf{m})$ ,  $\mathbf{m} \in \mathcal{X}$  is uniformly projected to (11).

*Proof.* Please see the proof in Appendix A-A. ■

A slice satisfying Lemma 1 is the longest slice that does not contain any duplicated samples. Thus, the  $L$ -point DFT along such slice captures the maximum information in the frequency domain with the least number of samples. The set  $\mathcal{A}$  defined in Lemma 1 contains a large number of elements, providing sufficient randomness for frequency projection. For example, when  $N_0 = N_1 = 4$ ,  $\mathcal{A} = \{[1, 1]^T, [1, 2]^T, [1, 3]^T, [2, 1]^T, [2, 3]^T, [3, 1]^T, [3, 2]^T\}$ . This means that  $\frac{|\mathcal{A}|}{|\mathcal{X}|} \approx 44\%$  of all the possible values of  $\boldsymbol{\alpha}$  yields a uniform projection. When  $N_0 = N_1 = 256$ ,  $|\mathcal{A}| \approx 39636$  and  $\frac{|\mathcal{A}|}{|\mathcal{X}|} \approx 60\%$ .

Fig. 2 shows an example of a time domain line designed to allow an orthogonal and uniform projection. The corresponding frequency domain line satisfying (10) for  $m = 1$  is also shown; the two lines are orthogonal to each other and intercept at  $[11, 1]^T$ . The length of the time domain and the frequency domain lines are 16, 8, respectively. Each line is composed of several line segments due to the modulo operation.

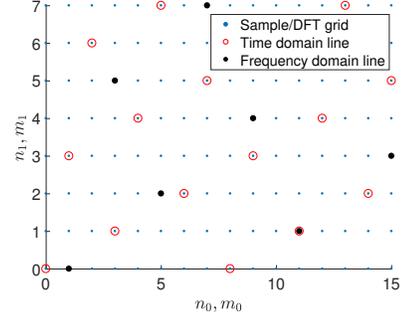


Fig. 2. An orthogonal pair of time and frequency domain lines.  $N_0 = 16, N_1 = 8, L = 16, \boldsymbol{\alpha} = [1, 3]^T, \boldsymbol{\tau} = [0, 0]^T$ .

**Remark 1.** In the  $L$ -point DFT of samples along a time-domain line with slope  $\alpha_1/\alpha_0$ , each entry represents a projection of the 2-D DFT along the line with slope  $-\alpha_0 N_1/(\alpha_1 N_0)$  in the  $N_0 \times N_1$ -point DFT domain, which is orthogonal to the time-domain line. This is closely related to the Fourier projection-slice theorem, which states that the Fourier transform of a projection is a slice of the Fourier transform of the projected object. While the classical projection is in the time domain and the corresponding slice is in the frequency domain, in the FPS-SFT case, the projection is in the DFT domain and the corresponding slice is in the sample (discrete-time) domain. The important difference between the Fourier projection-slice theorem and FPS-SFT is that while the former reconstructs the frequency domain of the signal via interpolation of frequency-domain slices or reconstructs the time-domain samples by solving a system of linear equations relating the DFT along projections and the time-domain samples, the latter efficiently recovers the significant frequencies of the signal directly based on the DFT of time-domain 1-D slices, i.e., samples along lines; the latter involves lower complexity.

The efficiency of FPS-SFT is achieved by exploring the sparsity nature of the signal in the frequency domain, which is explained in the following.

We assume that signal is sparse in the frequency domain, i.e.,  $|\mathbb{S}| = O(L)$ . Then, with high probability,  $|\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)| = |\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}_0, m)| = |\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}_1, m)| \neq 0$ , where  $\boldsymbol{\tau}_0 \triangleq [[\tau_0 + 1]_{N_0}, \tau_1]^T$ ,  $\boldsymbol{\tau}_1 \triangleq [\tau_0, [\tau_1 + 1]_{N_1}]^T$ . Thus, the  $m$ -th bin is 1-sparse, and it holds that

$$\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m) = a e^{j2\pi \left( \frac{m_0 \tau_0}{N_0} + \frac{m_1 \tau_1}{N_1} \right)}, (a, \omega) \in \mathbb{S}. \quad (13)$$

In such case, the 2-D frequency,  $(a, \omega)$ , can be ‘decoded’ as

$$\begin{aligned} m_0 &= \left[ \frac{N_0}{2\pi} \phi \left( \frac{\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}_0, m)}{\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)} \right) \right]_{N_0}, \\ m_1 &= \left[ \frac{N_1}{2\pi} \phi \left( \frac{\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}_1, m)}{\hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m)} \right) \right]_{N_1}, \\ a &= \hat{s}(\boldsymbol{\alpha}, \boldsymbol{\tau}, m) e^{-j2\pi (m_0 \tau_0 / N_0 + m_1 \tau_1 / N_1)}. \end{aligned} \quad (14)$$

This is the OFDM-trick adapted to FPS-SFT; such design requires sampling along three lines of the same slope but different offsets, allowing the frequency components to be decoded independently in each dimension.

In order to recover all the frequencies in  $\mathbb{S}$  efficiently, each iteration of FPS-SFT adopts a random choice of line slope from the set of  $\mathcal{A}$  defined in Lemma 1. Furthermore, the contribution of the recovered frequencies in the previous iterations is removed via a *construction-subtraction* approach so that the signal becomes sparser in future iterations. Specifically, assuming that for the current iteration the line slope and offset parameters are  $\alpha, \tau$ , respectively, the recovered 2-D frequencies are projected into  $L$  frequency bins to construct the DFT of the slice taken along the line of  $\mathcal{E}(\alpha, \tau, l), l \in [L]$ , i.e.,  $\hat{s}_r(\alpha, \tau, m) \triangleq \sum_{(a, \omega) \in \mathcal{I}_m} a e^{j2\pi(\frac{m_0 \tau_0}{N_0} + \frac{m_1 \tau_1}{N_1})}$ ,  $m \in [L]$ , where  $\mathcal{I}_m, m \in [L]$  represent the subsets of the recovered frequencies, i.e.,  $\mathcal{I}_m \triangleq \{(a, \omega) : \omega \text{ satisfies (10)}\}, m \in [L]$ . Next, the  $L$ -point inverse DFT (IDFT), multiplied by  $L$ , is applied on  $\hat{s}_r(\alpha, \tau, m), m \in [L]$ , from which the slice,  $s_r(\alpha, \tau, l), l \in [L]$ , due to the previously recovered frequencies is constructed. Subsequently, the constructed slice is subtracted from the slice of the current iteration. **The pseudo-code of the FPS-SFT algorithm can be found in Appendix B.**

### C. Convergence of FPS-SFT

In this section, we investigate the convergence of FPS-SFT. First, let us look at a special case, where  $N_0, N_1$  are co-prime.

**Theorem 1. (One-projection theorem of FPS-SFT):** Consider the signal model of (1), where  $N_0, N_1$  are co-prime and  $0 \leq K \leq N$ . The exact reconstruction of  $\mathbb{S}$  via FPS-SFT only takes one iteration.

*Proof.* Please see the proof in Appendix A-B. ■

In the Fourier projection-slice theorem, a band-limited signal of finite size of  $N_0 \times N_0$  can be exactly reconstructed by a single projection in the time domain, which is equivalent to a single slice in the frequency domain, provided that the slope parameters,  $\alpha_0, \alpha_1$ , of the line, from which the slice is evaluated, are co-prime and the equality  $\alpha_0 m_0 + \alpha_1 m_1 = \alpha_0 m'_0 + \alpha_1 m'_1$  holds only for  $m_0 = m'_0, m_1 = m'_1$  when  $m_0, m'_0, m_1, m'_1 \in [N_0]$ ; this is referred to as the one-projection theorem [22]. Theorem 1 is the one-projection theorem of FPS-SFT and provides the conditions for exact recovery of the signal with arbitrary sparsity level using only one projection in the frequency domain. Hence, the exact recovery of the signal requires only one iteration of FPS-SFT.

**Remark 2.** The two one-projection theorems establish an unambiguous one-to-one mapping from a 2-D sequence to a 1-D sequence, respectively. Specifically, the classic one-projection theorem establishes the mapping from the 2-D time-domain samples to 1-D time-domain samples of the length of  $N_0^2$ ; each entry of the DFT of the projection can be represented by a weighted summation of the  $N_0^2$  time-domain samples. Hence, the exact recovery of the time domain samples requires inverting a linear equation system containing at least  $N_0^2$  equations. On the other hand, the one-projection theorem of FPS-SFT establishes the one-to-one mapping from the coefficients of the  $N_0 \times N_1$ -point DFT of the 2-D data to the coefficients of the  $N$ -point DFT along a slice of the 2-D data;

such slice can be viewed as a rearrangement of the 2-D data into a 1-D sequence of the same number of samples. The exact recovery of the  $N_0 \times N_1$ -point DFT of the data is achieved by the low-complexity OFDM-trick under the framework of FPS-SFT. For  $N_0, N_1$  are co-prime, the 2-D  $N_0 \times N_1$ -point DFT can also be implemented via an  $N$ -point DFT based on the Good-Thomas mapping [23], where the unambiguous mapping is achieved via the Chinese Remainder Theorem-based indexing.

**Theorem 2. (Convergence of FPS-SFT):** Consider the application of FPS-SFT on the signal model of (1), where the frequencies in  $\mathbb{S}$  are assumed to be distributed uniformly at random. Then,  $T$ , the expected number of iterations needed to recover  $\mathbb{S}$  in average cases can be found by evaluating the following inequality

$$\sum_{i \in [T]} M_i \geq K, \quad (15)$$

where  $M_i = Q_i K_i$  is the number of the recovered frequencies in the  $i$ -th iteration;  $K_i = K \prod_{k \in [i]} (1 - Q_k)$  with  $K_0 = K$  is the number of remaining frequencies that have not been recovered in the  $i$ -th iteration;  $Q_i = (1 - K_i/N)^{N/L-1}$  is the probability of a remaining significant frequency being projected into a 1-sparse bin, and thus being recovered in the  $i$ -th iteration.

*Proof.* Please see the proof in Appendix A-C. ■

Fig. 3 shows the relationship between  $T$  and  $K/L$ . When  $K/L$  is a small number such as  $K/L = 3$ ,  $T$  is small; this results in a low “big-Oh” overhead [5] of the algorithm. However,  $T$  grows super-linearly against  $K/L$ ; such growth rate depends on  $K/N$ , i.e., the greater the  $K/N$ , the larger the growth rate. In a non-sparse scenario, i.e.,  $K/N$  approaches to 1,  $T$  is too large to be applicable. Also, FPS-SFT can fail in a non-sparse scenario (except that  $N_0, N_1$  are co-prime), in which none of the projections creates a 1-sparse bin.

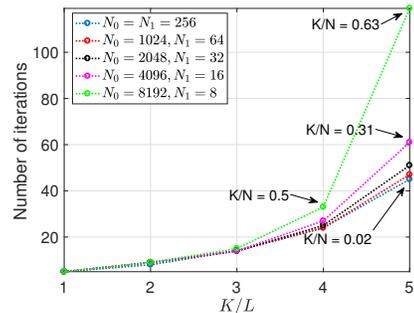


Fig. 3. Number of iterations of FPS-SFT versus  $K/L$ .

Note that although in order to prove Theorem 2 we assume that the frequency distribution in the signal model of (1) is uniform, the numerical results show no significant difference in the convergence of FPS-SFT when the frequencies are clustered (see Section V-D for details). This is because the multidimensional clustered frequencies are uniformly projected to one dimension due to the randomly generated line slopes of the FPS-SFT.

#### D. Complexity analysis of FPS-SFT

FPS-SFT executes  $T$  iterations in average cases; in the 2-D case, each iteration uses  $3L$  samples, since 3  $L$ -length slices, with  $L = \text{LCM}(N_0, N_1)$  are extracted in order to decode the two frequency components of a 2-D frequency (see (14)). Hence, the sample complexity of FPS-SFT is  $O(3TL) = O(TL)$ . The core processing of FPS-SFT is the  $L$ -point 1-D DFT, which can be implemented by the FFT with the computational complexity of  $O(L \log L)$ . The  $L$ -point IDFT in the construction-subtraction procedure can also be implemented by the FFT. In addition to the FFT, each iteration needs to evaluate up to  $L$  frequencies. Hence the computational complexity of FPS-SFT is  $O(T(L \log L + L)) = O(TL \log L)$ . If we let  $T$  equal to  $T_{max} \in \mathbb{N}$ , which is a sufficiently large constant to allow the convergence of FPS-SFT in average cases for a given signal size and a range of  $K$ , then, the sample and computational complexity of FPS-SFT become  $O(L)$  and  $O(L \log L)$ , respectively. For  $K = O(L)$ , FPS-SFT achieves the lowest sample and computational complexity, i.e.,  $O(K)$  and  $O(K \log K)$ , respectively, of all known SFT algorithms [5], [7].

In general, in the  $D$ -D case, according to the multidimensional extension [18], it is easy to see that the sample and computational complexity of FPS-SFT are  $O(DK)$  and  $O(DK \log(DK))$ , respectively when  $K = O(L)$ .

#### IV. RFPS-SFT: A ROBUST EXTENSION OF FPS-SFT

FPS-SFT is developed for the signal model of (1), where the data is exactly sparse in the frequency domain and the frequencies are assumed to be on the DFT grid. However, in real-world applications, the data usually contains noise and thus is only approximately sparse, i.e., dominated by a few significant frequencies. Also, the significant frequencies are typically off-grid. In what follows, we propose RFPS-SFT for signals that follow the signal model of (2). RFPS-SFT works within the framework of FPS-SFT and employs a windowing technique to reduce frequency leakage due to the off-grid frequencies and a voting-based frequency localization to reduce the frequency decoding error due to noise. The pseudo-code of the RFPS-SFT algorithm can be found in Appendix B.

##### A. Windowing

To address the issue of off-grid frequencies, we apply a window  $w(\mathbf{n})$ ,  $\mathbf{n} \in \mathcal{X}$  on the signal of (2). The PSR of the window,  $\rho_w$ , is designed such that the side-lobes of the strongest frequency are below the noise level, hence the leakage of the significant frequencies can be neglected and the sparsity of the signal in the frequency domain can be preserved to some extent. Lemma 2 reflects the relationship between  $\rho_w$  and the maximum SNR of the signal.

**Lemma 2. (Window design for RFPS-SFT):** Consider  $\hat{r}(\mathbf{m})$ , which is the  $N_0 \times N_1$ -point DFT of the windowed signal of (2). Let  $\mathbf{W} \in \mathbb{R}^{N_0 \times N_1}$  be the matrix generated by the window

function of  $w(\mathbf{n})$ ,  $\mathbf{n} \in \mathcal{X}$ . The PSR of the window,  $\rho_w$ , should be designed such that

$$\rho_w > \frac{2\|\mathbf{W}\|_1}{\sqrt{\pi}\|\mathbf{W}\|_2} \sqrt{\text{SNR}_{max}}, \quad (16)$$

Where  $\text{SNR}_{max} \triangleq a_{max}^2/\sigma_n^2$ .

*Proof.* Please see the proof in Appendix A-D.  $\blacksquare$

Note that while the window is designed for the entire data cube, the windowing is applied only to the sampled locations, which does not increase the sample complexity of RFPS-SFT.

##### B. Voting-based frequency decoding

When the signal is approximately sparse, the frequencies decoded by (14) are not integers. Since we aim to recover the gridded frequencies, i.e.,  $\mathbb{S}$  of (1), the recovered frequency indices are rounded to the nearest integers. When the SNR is low, the frequency decoding could result in false frequencies; those false frequencies enter the future iterations and generate more false frequencies. To suppress the effect of false frequencies, motivated by the classical  $m$ -out-of- $n$  radar signal detector [24], RFPS-SFT adopts an  $n_d$ -out-of- $n_s$  voting procedure in each iteration. Specifically, within each iteration of RFPS-SFT,  $n_s$  sub-iterations are applied; each sub-iteration adopts randomly generated line slope and offset parameters and recovers a subset of frequencies,  $\mathbb{S}_i, i \in [n_s]$ . Within those frequency sets, a given frequency could be recovered by  $n$  out of  $n_s$  sub-iterations. For a true significant frequency,  $n$  is typically larger than that of a false frequency, thus only those frequencies with  $n \geq n_d$  are retained as the recovered frequencies of the current iteration. When  $(n_s, n_d)$  is properly designed, the false frequencies can be reduced significantly.

##### C. Lower bound of the probability of correct localization and convergence of RFPS-SFT

The probability of decoding error is related to the SNR, signal sparsity and also the parameters  $(n_s, n_d)$  of the RFPS-SFT. In the following, we provide a lower bound for the probability of correct localization of the significant frequencies for each iteration of RFPS-SFT, from which one can study the convergence of RFPS-SFT, i.e., the number of iterations needed in order to recover all the significant frequencies with sufficient SNR.

From Section II, a 2-D continuous-valued frequency  $(a, \omega) \in \mathbb{S}'$  of (2) is associated with a cluster of 2-D on-grid frequencies  $\mathbb{S}_0 \subseteq \mathbb{S}$  of (1). Let us assume that the frequency  $(a_d, 2\pi[m_0/N_0, m_1/N_1]^T) \in \mathbb{S}_0$  with  $[m_0, m_1]^T \in \mathcal{X}$  has the largest absolute amplitude among the frequencies in  $\mathbb{S}_0$ . In addition, let us assume that the SNR of  $(a, \omega)$  is sufficiently high. Then the probability of correctly localizing  $(a_d, 2\pi[m_0/N_0, m_1/N_1]^T)$  in each iteration is lower bounded by

$$P_d \triangleq \sum_{n'_d=n_d}^{n_s} \binom{n_s}{n'_d} (P_1 P_w)^{n'_d} (1 - P_1 P_w)^{n_s - n'_d}, \quad (17)$$

where  $P_1 \triangleq (1 - |\mathbb{S}''|/N)^{N/L-1}$  with  $L = \text{LCM}(N_0, N_1)$  is the probability of a frequency in  $\mathbb{S}''$  being projected to a

1-sparse bin, and  $\mathbb{S}''$  with  $\mathbb{S}'' \subseteq \mathbb{S}$  represents the remaining frequencies to be recovered in future iterations of RFPS-SFT;  $P_w \triangleq (1 - P_u)(1 - P_v)$  is the lower bound of the probability of correct localization for a 2-D frequency that is projected into an 1-sparse bin in one sub-iteration of RFPS-SFT;  $P_u, P_v$  are the upper bounds of the probability of localization error for the two frequency components,  $m_0, m_1$ , respectively, which are defined as  $P_u \triangleq (\sigma_p(1 - f_{|a_n|}(\delta_u)))^2$ ,  $P_v \triangleq (\sigma_p(1 - f_{|a_n|}(\delta_v)))^2$ , where  $\delta_u \triangleq a\pi\|\mathbf{W}\|_1/(2NN_0)$ ,  $\delta_v \triangleq a\pi\|\mathbf{W}\|_1/(2NN_1)$ , with  $\mathbf{W} \in \mathbb{R}^{N_0 \times N_1}$  the window that is applied on the data;  $\sigma_p$  with  $\frac{1}{2} \leq \sigma_p \leq \frac{1}{2\pi}$  is the parameter determined by the phases of the error vectors contained in the 1-sparse bin;  $f_{|a_n|}(x)$  is the cumulative distribution function of the Rayleigh distribution, which is defined as  $f_{|a_n|}(x) \triangleq 1 - e^{-x^2/(2\sigma_{a_n}^2)}$ ,  $x > 0$ , where  $\sigma_{a_n}^2 \triangleq \sigma_n^2\|\mathbf{W}\|_2^2/(2NL)$ .

The proof of (17) can be found in Appendix A-E. Essentially, (17) represents the complementary cumulative binomial probability resulted from the  $n_d$ -out-of- $n_s$  voting procedure, where the success probability of each experiment, i.e., localizing  $(a_d, 2\pi[m_0/N_0, m_1/N_1]^T)$  in each sub-iteration of RFPS-SFT is  $P_1P_w$ . When  $K = |\mathbb{S}|$  is known, (17) can be applied to estimate the largest number of iterations (the upper bound) of RFPS-SFT in order to recover all the frequencies in  $\mathbb{S}$  since the least number of recovered frequencies in each iteration can be estimated by  $|\mathbb{S}''|P_d$ .

## V. NUMERICAL RESULTS

In this section, we provide some numerical results to verify the theoretical findings related to the proposed FPS-SFT and RFPS-SFT algorithms. Unless stated otherwise, the size of the test data is set equal to  $N_0 = N_1 = 256$ . We simulate cases when frequencies are uniformly distributed and when they are clustered; for clustered cases, we consider clusters of 9 and 25 frequencies. The experimental results are averaged over 100 iterations of Monte Carlo simulation.

### A. Comparison between FPS-SFT and SO-SFT

We compare the performance of SO-SFT and the proposed FPS-SFT for the 2-D case. When  $N_0 = N_1$ , the line length,  $L$ , of FPS-SFT equals  $N_0$ , and each iteration of FPS-SFT uses  $3N_0$  samples. We limit the maximum iterations to  $T_{max} = N/(3L) \approx 85$ ; which corresponds to roughly 100% samples of the input data. Fig. 4 (a) shows the probability of exact recovery versus the level of sparsity for FPS-SFT and SO-SFT. When the signal is very sparse, i.e.,  $K < N_0/2$ , SO-SFT has a high probability of exact recovery, while it fails when the sparsity is moderately large, i.e.,  $K > 2N_0$ . Moreover, SO-SFT only works for the scenario in which frequencies are distributed uniformly, while it fails when there exists even a single frequency cluster. On the contrary, FPS-SFT applies to signals with a wide range of sparsity levels. For instance, the success rate of FPS-SFT is approximate 97% when  $K = 5N_0$ . In all cases, the success rates drop to 0 when  $K = 6N_0$ , since in such case, the exact recovery needs roughly 100 iterations, which exceeds  $T_{max}$ . Fig. 4 (b) shows the ratio of samples used by the FPS-SFT and SO-SFT for exact recovery

to the total number of data sample  $N$  versus different sparsity level. The figure shows that the sparser the signal, the fewer samples are required by the FPS-SFT. For example, when  $K = N_0$ , only 5.9% of the signal samples are required for FPS-SFT. SO-SFT only needs 1.6% of the signal samples in very sparse scenarios, while it fails in less sparse or non-uniformly distributed frequency cases. The performance of FPS-SFT is similar for both uniformly-distributed and clustered frequency cases at the same sparsity level; this is due to the randomized projections that can effectively isolate the 2-D frequencies into 1-sparse bins, even when the signal is less sparse ( $K$  is large) and the frequencies are clustered. **Note that the supper linearity of the growth of the ratio of samples against  $K$  is due to that the growth of the number of iterations of FPS-SFT against  $K$  is super-linear.**

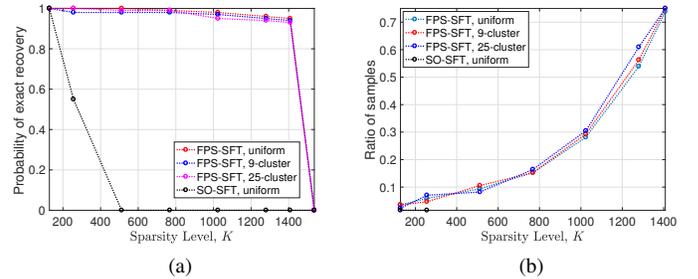


Fig. 4. Comparison between FPS-SFT and SO-SFT. (a) Probability of exact recovery versus sparsity level,  $K$ . (b) Ratio of samples (the averaged number of samples used by FPS-SFT over  $N$ ) needed for exact recovery versus  $K$ .

### B. Comparison between (R)FPS-SFT and the SFT of [12], [13]

We compare (R)FPS-SFT and the SFT of [12], [13] in 2-D cases. The main difference between the SFT of [12] and the SFT of [13] is that the former takes the slices only from the borders and the diagonals from the input data matrix, while the latter, in addition to taking slices from the borders, also takes slices along many lines with predefined slopes; this increases the degrees of freedom of projecting 2-D frequencies onto 1-D lines.

Fig. 5 (a) shows the frequency localization performance of SFT of [12], [13] with respect to  $K$  in noiseless cases. Compared to FPS-SFT, SFT of [12], [13] only successes in very sparse scenarios. For instance, when  $K = 50$ , the best success rate that the SFT of [13] can achieve is approximately 67%, and it fails completely when  $K > 150$ , while the successful rate of FPS-SFT is approximately 97% when  $K = 1280$  (see Fig. 4 (a)). One way to increase the success rate of SFT of [13] is to use a larger  $T$  at the expense of increasing complexity. However, the increasing of the success rate saturates when  $T$  is greater than a certain value. For instance, the success rates when  $T = 20$  and  $T = 30$  are similar.

Despite that the SFT of [12], [13] suffers from less sparse signals, it is more robust to noise as compared to RFPS-SFT. For instance, in Fig. 5 (b), one can see that when  $K < 30$  the success rate of localization of the SFT of [13] for signal with SNR equal to  $-5dB$  is similar to that of RSFT applied

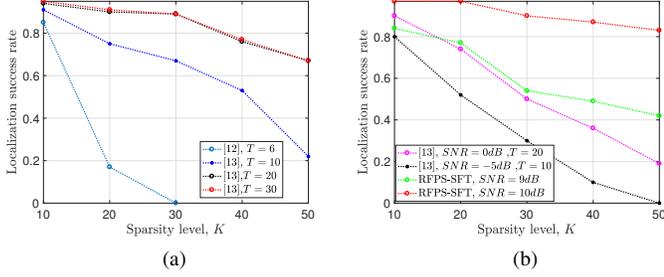


Fig. 5. Comparison between (R)FPS-SFT and SFT of [12], [13]. (a) Localization success rate versus  $K$  in noiseless cases. (b) Localization success rate versus  $K$  in noisy cases. Frequencies are on the grid.

on signals whose SNR is 9dB. When the SNR is greater than 11dB, the success rate of RFPS-SFT approaches to 100%.

The computation of the SFT of [12], [13] is significantly slower as compared to that of (R)FPS-SFT, as the computation complexity of the former is  $O((N + K^3) \log N)$  [13] in the 2-D case, which is even greater than that of the FFT.

### C. Line slope of FPS-SFT

From Lemma 1, when the line slope parameters  $\alpha$  are randomly selected from the set  $\mathcal{A}$  for each iteration of FPS-SFT, the expected number of iterations for exact recovery can be reduced as compared to choosing  $\alpha$  to be arbitrary from the set of  $\mathcal{X}$ ; this is because as compared to the latter case, in the former case, more 1-sparse bins are likely to be created in each iteration due to the uniformity of projections. In Fig. 6 (a), we compare the number of iterations of FPS-SFT when  $\alpha$  is chosen from  $\mathcal{A}$  and  $\mathcal{X}$ , respectively. The former uses fewer iterations to achieve an exact recovery for all the sparsity range than that of the latter.

The high probability of exact recovery of FPS-SFT in less sparse cases is due to the abundance of degrees of freedom in frequency projection, which requires a sufficiently large  $|\mathcal{A}|$ . When  $N_0 = N_1 = 256$ , it is easy to verify that  $|\mathcal{A}| = 39639$ . Fig. 6 (b) shows the probability of exact recovery versus sparsity when we use subsets of  $\mathcal{A}$  of different support sizes. The slope parameter set,  $\mathcal{A}'$ , in each experiment, is created by randomly picking a subset of  $\mathcal{A}$  with a specific size of support. The figure shows that the less sparse the signal (the larger  $K$ ), the larger size of  $\mathcal{A}'$  is needed to achieve a high probability of exact recovery. Essentially,  $|\mathcal{A}'|$  should be large enough so that for each iteration of FPS-SFT, a distinct slope can be obtained from  $\mathcal{A}'$  with high probability. Compared to the uniformly distributed frequency cases, the clustered frequency cases require a larger  $|\mathcal{A}'|$ , since the latter requires larger degrees of freedom than the former in order to isolate the clustered frequencies by randomly projecting those frequencies to distinct 1-sparse bins of the DFT along lines.

### D. Convergence of FPS-SFT

We verify the expected number of iterations of FPS-SFT in order to exactly recover the signal (see Theorem 2). The relationship between the number of iterations,  $T$ , and sparsity

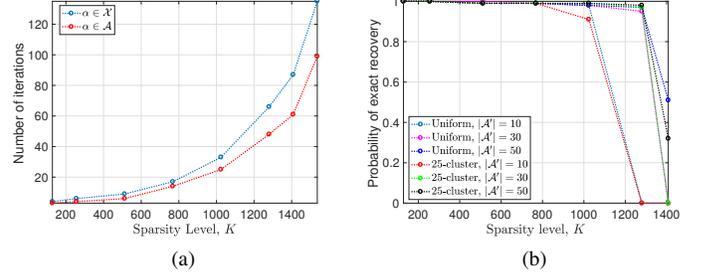


Fig. 6. Effect of the line slope to the FPS-SFT. (a) Number of iterations of exact recovery versus sparsity. (b) Effect of the size of the slope parameter set ( $|\mathcal{A}'|$ ) to the exact recovery probability.

level,  $K$ , for different data sizes of the same number of samples (i.e., different  $N_0, N_1$  but the same  $N$ ) are shown in Fig. 7 (a). As expected, for all cases,  $T$  increases as  $K$  increases; the increasing rate in the cases of  $N_0 = N_1 = 256$  is greater than the cases when  $N_0 = 512, N_1 = 128$  and  $N_0 = 1024, N_1 = 64$ . Also, the former case requires a large number of iterations than the latter two cases. This is because the line length,  $L$ , equals to 256, 512 and 1024 for the three cases, respectively. A larger  $L$  leads to a higher probability of creating more 1-sparse bins in each iteration of FPS-SFT, which results in a faster convergence of the algorithm. The clustered frequencies do not require larger  $T$  as compared to the uniformly distributed frequencies, which shows that FPS-SFT is efficient in solving non-uniformly distributed frequencies. The number of samples used by FPS-SFT depends both on  $T$  and  $L$ . Fig. 7 (b) shows that when the signal is very sparse, i.e.,  $K < 640$ , the equal-length case ( $N_0 = N_1$ ) uses the least number of samples, while for less sparse cases, the number of samples required by FPS-SFT is less in the cases when  $N_0 = 512, N_1 = 256$  and  $N_0 = 1024, N_1 = 64$  than the case when  $N_0 = 256, N_1 = 256$ .

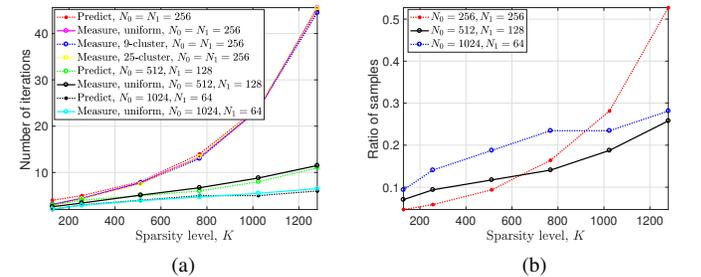


Fig. 7. The convergence of FPS-SFT. (a) Number of iterations versus sparsity. The predicted values are calculated via Theorem 2. (b) Ratio of samples versus sparsity. The data under test has the same number of samples but different size.

According to Theorem 1, the set  $\mathcal{S}$  of the signal model (1) can be reconstructed exactly based on only one iteration of FPS-SFT when  $N_0, N_1$  are co-prime. Fig. 8 provides the visualization of the exact recovery using only one iteration of FPS-SFT when  $N_0 = 32, N_1 = 31, K = 640$ . Note that this scenario is not sparse.

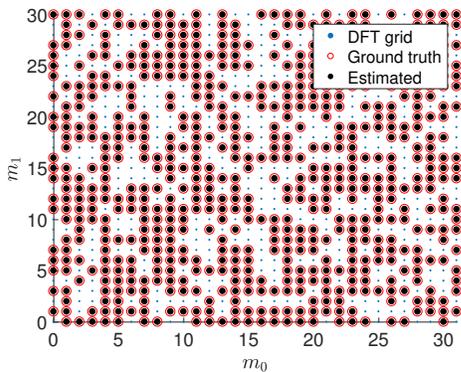


Fig. 8. Exact recovery of the signal by one iteration of FPS-SFT.  $N_0 = 32, N_1 = 31, K = 20N_0$ .

### E. Effect of windowing on frequency localization of RFPS-SFT

For the data that contains off-grid frequencies, the PSR of the window,  $\rho_w$ , should be sufficiently large in order to reduce the side-lobes of the significant frequencies (see Lemma 2). However, the larger the  $\rho_w$ , the wider the main-lobe of the window, which results in larger frequency clusters in the DFT domain and thus larger  $|\mathbb{S}|$  of the signal model of (1), i.e., a less sparse signal. Moreover, the larger the  $\rho_w$ , the smaller the SNR of the windowed signal, which leads to larger frequency localization error. Fig. 9 (a) shows the numerical evaluation of windows with various  $\rho_w$  for signals of various  $SNR_{max}$  and sparsity level,  $K' = |\mathbb{S}'|$  (see (2)). According to (16), for signals with  $SNR_{max}$  equal to 20dB and 30dB, the  $\rho_w$  of the window should be larger than 56dB and 60dB, respectively. In those two cases, the frequency localization success rate, i.e., the ratio of number of correctly localized frequencies to the number of remaining significant frequencies in each iteration of RFPS-SFT appears to be the highest when  $\rho_w$  equal to 60dB and 70dB, respectively. Fig. 9 (a) shows the success rate of the first iteration of RFPS-SFT, which is the lowest success rate of all the iterations. Fig. 9 (b) shows the percentage of the recovered energy, defined as the ratio of the recovered signal energy over the energy of the windowed original signal for each iteration for a typical execution of RFPS-SFT. For the on-grid cases, we apply the rectangular window and all the signal energy can be recovered; for the off-grid cases, since the Dolph-Chebyshev window of large PSR smears the peaks of significant frequencies, a significant portion of signal energy spreads across the frequency spectrum. As a result, RFPS-SFT only recovers signal energies that concentrates in the main lobes of significant frequencies.

Fig. 10 demonstrates localization of off-grid 2-D frequencies of RFPS-SFT using Dolph-Chebyshev window for various values of  $\rho_w$ . A window with insufficient  $\rho_w$  leads to miss detections and false alarms (see Fig. 10 (a)), while a window with sufficiently large  $\rho_w$  yields good performance in frequency localization, with a trade-off of causing larger frequency cluster sizes (see Fig. 10 (b)).

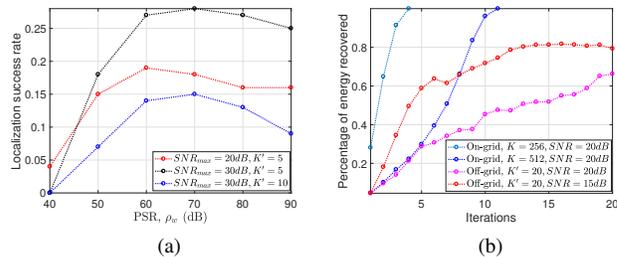


Fig. 9. Effect of windowing. (a) Frequency localization success rate of the first iteration of RFPS-SFT versus window PSR. The Dolph-Chebyshev windows with various PSR is applied.  $(n_s, n_d) = (3, 2)$ . (b) Percentage of energy recovered versus iteration.

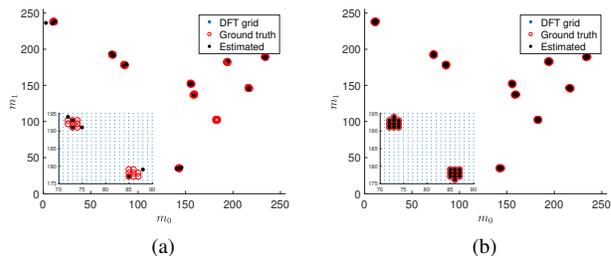


Fig. 10. 2-D frequency recovery with different window.  $K' = 10, \sigma_n = 1, a_{min} = a_{max}, SNR_{max} = 30dB, (n_s, n_d) = (3, 2), T = 30$ . Dolph-Chebyshev windows with various PSR are adopted. The ground truth represents  $\mathbb{S}$  of (1), which relates to the window. (a) The PSR of the window  $\rho_w = 45dB$ . (b)  $\rho_w = 70dB$ .

### F. Effect of voting on frequency localization of RFPS-SFT

The  $n_d$ -out-of- $n_s$  voting in frequency decoding procedure of RFPS-SFT can significantly reduce the false alarm rate. For a fixed  $n_s$ , larger  $n_d/n_s$  results in smaller false alarm rate. However, the smaller the false alarm rate, the larger the number of the iterations required to recover all the significant frequencies. Figs. 11 and Fig. 10 (b) show the examples of 2-D frequency recovery using different values of  $(n_s, n_d)$ . In Fig. 11 (a), we set  $(n_s, n_d) = (1, 1)$ , which reduces the frequency localization of RFPS-SFT to that of FPS-SFT, i.e., without voting. In this case, one can see that many false frequencies are generated. Figs. 11 (b), (c) show the frequency localization result with  $(n_s, n_d)$  equal to  $(3, 1)$ ,  $(3, 3)$ , respectively; while the former generates large amount of false frequencies, the latter exhibits miss detection, which implies the insufficiency of number of iterations of RFPS-SFT used in this case. Fig. 10 (b) shows the ideal performance when  $(n_s, n_d)$  is designed as  $(3, 2)$ .

### G. Effect of the SNR and the sparsity level on the convergence of RFPS-SFT

The expected number of iterations of RFPS-SFT to recover all the significant frequencies is affected by the SNR and the sparsity level of the signal. A low SNR and less sparse signal require a large number of iterations. As discussed in Section IV-C, we are able to estimate the expected largest number of iterations that recover all the significant frequencies of sufficient SNR. Fig. 12 shows the predicted and measured

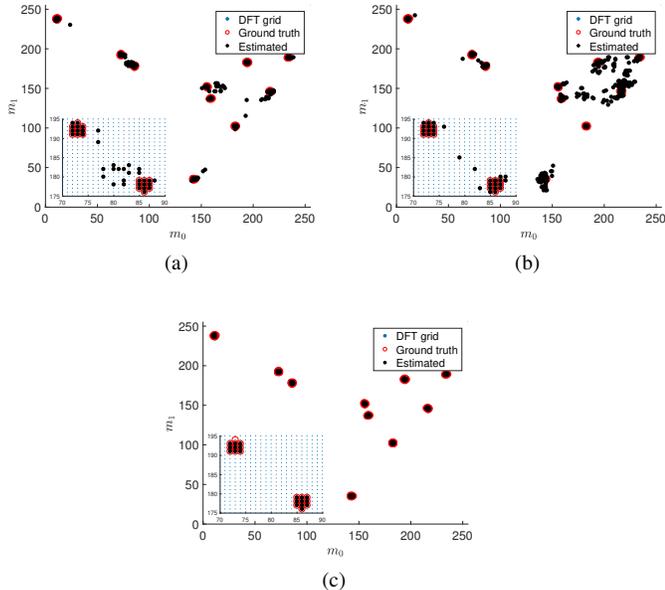


Fig. 11. Effect of voting on 2-D frequency recovery.  $K' = 10, \sigma_n = 1, a_{min} = a_{max}, SNR_{max} = 30dB, T = 30$ . Dolph-Chebyshev windows with  $\rho_w = 70dB$  is applied. (a)  $(n_d, n_s) = (1, 1)$ . (b)  $(n_d, n_s) = (3, 1)$ . (c)  $(n_d, n_s) = (3, 3)$ .

number of iterations of RFPS-SFT for signals with various SNR and sparsity level. The figure shows that the number of iterations upper bounds are consistent with the measurements.

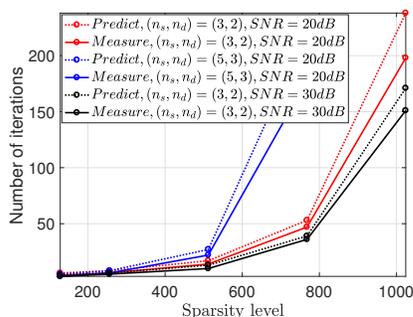


Fig. 12. Effect of SNR and sparsity level on the convergence of RFPS-SFT.

## VI. APPLICATION OF RFPS-SFT IN DBF AUTOMOTIVE RADAR SIGNAL PROCESSING

With the rapid developments in the advanced driver-assistance systems and self-driving vehicles, the automotive radar plays an increasingly important role in providing multidimensional information on the dynamic environment to the control unit of the vehicle. Traditional automotive radars measure range and range rate (Doppler) of the targets including cars, pedestrians and obstacles using FMCW. A DBF automotive radar [25] can provide angular information both in azimuth and elevation [26] of the targets, which is desirable in self-driving applications.

A typical DBF automotive radar uses uniform linear array as the receive array (see Fig. 13). Let us assume that the array

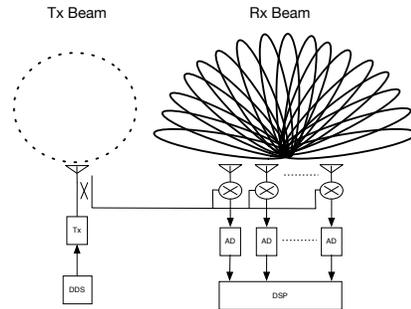


Fig. 13. DBF automotive radar system structure. A broad beam is formed by a transmit antenna, while multiple narrow beams are formed by the receive array. The demodulated and digitized received signal is processed by a digital signal processor (DSP), while the transmit waveform is generated by a direct digital synthesizer (DDS).

has  $N_2$  half-wavelength-spaced elements. The radar transmits FMCW waveform with a repetition interval of  $T_p$ . We also assume that there exist  $K'$  targets in the radar coverage. After de-chirping, sampling ( $N_0$  samples within a repetition interval) and analog-to-digital conversion for both I and Q channels, the received signal within  $N_1$  repetition interval can be expressed as (2) [17], where the vectors are 3-dimensional (3-D). The 3-D frequency  $\omega = [\omega_0, \omega_1, \omega_2]^T$  relates to the target parameters as

$$\begin{aligned} \omega_0 &= 2\pi(2\rho r/c + f_d)/f_s, \\ \omega_1 &= 2\pi f_d T_p, \\ \omega_2 &= \begin{cases} \pi \sin \theta, & \theta \in [0, 90^\circ) \\ 2\pi + \pi \sin \theta, & \theta \in [-90^\circ, 0). \end{cases} \end{aligned} \quad (18)$$

where  $\rho, c, f_s, f_d$  are the chirp rate, the speed of wave propagation, sampling frequency, and the Doppler frequency, respectively; the chirp rate is defined as the ratio of the signal bandwidth and the repetition interval. Thus, the target parameters are embedded in  $\omega$ , which can be estimated via RFPS-SFT when  $K' \ll N$ . The conventional processing requires multidimensional FFT, which is still challenging to real-time processing as the increasing of the data size due to the increasing of array size as well as the increasing of dimensionality (e.g., the DBF along both azimuth and elevation).

We simulate the target reconstruction for a DBF automotive radar via RFPS-SFT and compare with the FFT and RSFT based methods. The main radar parameters are listed in Table I; such radar configuration represents a typical long-range DBF radar [17] except that we set the number of antenna elements to be moderately large to provide a better angular resolution performance. Fig. 14 shows the target reconstruction of 3 radar targets via 3-D FFT, RSFT and RFPS-SFT. All the three algorithms are able to reconstruct all the targets. Compared to the FFT and RSFT, RFPS-SFT only requires approximately 3% of data samples, which exhibits low sample complexity. Also, the computation via RFPS-SFT is more efficient than the RSFT and FFT based algorithms. However, we note that RFPS-SFT requires larger SNR than the FFT

TABLE I  
RADAR PARAMETERS

Parameter	Symbol	Value
Center frequency	$f_c$	76GHz
Pulse bandwidth	$b_w$	200MHz
Pulse repetition time	$T_p$	89us
Number of range bins	$N_0$	512
Number of PRI	$N_1$	256
Number of antenna elements	$N_2$	16
Maximum range	$R_{max}$	300m

and the RSFT based methods. In near range radar applications, such as automotive radar, high SNR is relatively easy to obtain.

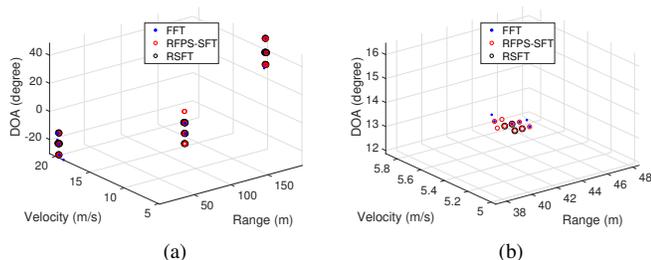


Fig. 14. Radar target reconstruction via FFT, FPS-SFT and RSFT. (a) Reconstruction of three targets. (b) Details of the frequency locations that are reconstructed for one of the three targets.

## VII. CONCLUSION

We have proposed FPS-SFT, a multidimensional sparse Fourier transform that is inspired by the Fourier projection-slice theorem. We have shown that FPS-SFT can handle less sparse data in the frequency domain of non-uniformly distributed frequencies while enjoys low sample and computational complexity. The relationship between FPS-SFT and the Fourier projection-slice theorem has been discussed. Especially, the connections between the one-projection theorems under the context of FPS-SFT and the Fourier projection-slice theorem has been exploited. We have also extended FPS-SFT to a robust version, i.e., RFPS-SFT which addresses noisy data containing off-grid frequencies; such data arises from real-life applications such as DFB automotive radar signals. We have shown via simulation that RFPS-SFT based algorithm could significantly reduce the sample and computational complexity of DBF automotive radar signal processing. For instance, as compared to the FFT and RSFT based methods, the proposed RFPS-SFT based approach only requires 3% of data samples to reconstruct the targets when a scenario only contains 3 targets.

## APPENDIX A

### COLLECTIONS OF PROOFS

#### A. Proof of Lemma 1

*Proof.* This proof is organized as follows. First, by exploring the Bézout's lemma [27], we prove that with the specified line parameters, i.e.,  $L = \text{LCM}(N_0, N_1), [\alpha_0, \alpha_1]^T \in \mathcal{A}, [\tau_0, \tau_1]^T \in \mathcal{X}$ , each entry of (11) contains at least the

projection of the DFT coefficient from one frequency location  $(m'_0, m'_1)$  in  $\mathcal{X}$ , i.e.,  $|\mathcal{P}_m| > 0, m \in [L]$ . Next, we prove that  $|\mathcal{P}_m| \geq N/L$ , followed by the proof of  $\mathcal{P}_m \cap \mathcal{P}_{m'} = \emptyset$  for  $m \neq m', m, m' \in [L]$ , and finally, we conclude that  $|\mathcal{P}_m| = N/L$ .

Let  $\alpha'_0 = \alpha_0 L/N_0, \alpha'_1 = \alpha_1 L/N_1$ . Since  $(\alpha_0, \alpha_1), (\alpha_0, L/N_1), (\alpha_1, L/N_0)$ , and  $(L/N_0, L/N_1)$  are co-prime pairs due to  $L = \text{LCM}(N_0, N_1)$ , it is obvious that  $\alpha'_0, \alpha'_1$  are also co-prime. According to the Bézout's lemma, there exist  $m_0, m_1 \in \mathbb{Z}$ , such that

$$\alpha'_0 m_0 + \alpha'_1 m_1 = 1. \quad (19)$$

By multiplying by  $m \in [L]$  the two sides of (19), we get  $\alpha'_0 m m_0 + \alpha'_1 m m_1 = m$ , which, using the Euclidean division, can be written as

$$\alpha'_0 (m'_0 + k_0 N_0) + \alpha'_1 (m'_1 + k_1 N_1) = m, \quad (20)$$

where  $m'_0 = [m m_0]_{N_0}, m'_1 = [m m_1]_{N_1}$  and  $k_0, k_1 \in \mathbb{Z}$ .

Since that

$$[\alpha'_0 k_0 N_0 + \alpha'_1 k_1 N_1]_L = [L(\alpha_0 k_0 + \alpha_1 k_1)]_L = 0, \quad (21)$$

on taking modulo- $L$  of the two sides of Eq. (20), we have

$$[\alpha'_0 m'_0 + \alpha'_1 m'_1]_L = m, \quad (22)$$

which is equivalent to (9). This means that there exists a frequency location  $[m'_0, m'_1]^T \in \mathcal{X}$ , whose DFT coefficient projects to  $\hat{s}(\alpha, \tau, m)$ , i.e.,  $|\mathcal{P}_m| > 0, m \in [L]$ .

Next, let us explore the solution structure of (22). It is easy to see that the frequency locations,  $[m'_0 + k\alpha'_1, m'_1 - k\alpha'_0]^T, k \in \mathbb{Z}$ , satisfies (22), i.e.,  $[\alpha'_0 (m'_0 + k\alpha'_1) + \alpha'_1 (m'_1 - k\alpha'_0)]_L = m$ , which can be written as  $[\alpha'_0 ([m'_0 + k\alpha'_1]_{N_0} + k_0 N_0) + \alpha'_1 ([m'_1 - k\alpha'_0]_{N_1} + k_1 N_1)]_L = m$ , where  $k_0, k_1 \in \mathbb{Z}$ . Again, by substituting (21), we have  $[\alpha'_0 [m'_0 + k\alpha'_1]_{N_0} + \alpha'_1 [m'_1 - k\alpha'_0]_{N_1}]_L = m$ . Hence, the DFT coefficients at frequency locations  $[[m'_0 + k\alpha'_1]_{N_0}, [m'_1 - k\alpha'_0]_{N_1}]^T \in \mathcal{P}_m \subseteq \mathcal{X}$ , also projects to  $\hat{s}(\alpha, \tau, m)$ ; those frequencies locate along the line with slope  $-\alpha_0 N_1 / (\alpha_1 N_0)$  and offset  $[m'_0, m'_1]^T$ ; such frequency-domain line is orthogonal to the time-domain line defined in (4).

Next, we prove that  $|\mathcal{P}_m| \geq N/L$ . Assume that for  $k \neq k'$ , there exists two duplicated frequency locations, i.e.,  $[[m'_0 + k\alpha'_1]_{N_0}, [m'_1 - k\alpha'_0]_{N_1}]^T = [[m'_0 + k'\alpha'_1]_{N_0}, [m'_1 - k'\alpha'_0]_{N_1}]^T$ . It follows that  $[k\alpha'_1]_{N_0} = [k'\alpha'_1]_{N_0}, [k\alpha'_0]_{N_1} = [k'\alpha'_0]_{N_1}$ , which can be rewritten as  $k\alpha'_1 = k'\alpha'_1 + k_0 N_0, k\alpha'_0 = k'\alpha'_0 + k_1 N_1$ , where  $k_0, k_1 \in \mathbb{Z}$ . It is easy to conclude that  $k_1/k_0 = \alpha_0/\alpha_1$ . Hence we have  $k\alpha'_1 = k'\alpha'_1 + i\alpha_1 N_0, k\alpha'_0 = k'\alpha'_0 + i\alpha_0 N_1$ , where  $i \in \mathbb{Z}, i \neq 0$ . Hence  $k - k' = iN/L$ , which means that the frequency location,  $[[m'_0 + k\alpha'_1]_{N_0}, [m'_1 - k\alpha'_0]_{N_1}]^T$ , repeats every  $N/L$  points. Hence, there exist at least  $N/L$  frequency locations whose DFT values projecting to  $\hat{s}(\alpha, \tau, m)$ , i.e.,  $|\mathcal{P}_m| \geq N/L$ .

Next, we prove that  $\mathcal{P}_m \cap \mathcal{P}_{m'} = \emptyset$  for  $m \neq m', m, m' \in [L]$ . Assume that  $[m_0, m_1]^T \in \mathcal{P}_m \cap \mathcal{P}_{m'}$ , it can be seen that  $[\alpha'_0 m_0 + \alpha'_1 m_1]_L = m = m'$ , which is contradict with  $m \neq m'$ . Hence  $\mathcal{P}_m \cap \mathcal{P}_{m'} = \emptyset$ .

Finally, by combing  $\mathcal{P}_m \cap \mathcal{P}_{m'} = \emptyset$ ,  $m \in [L]$ ,  $|\mathcal{P}_m| \geq N/L$  and  $|\mathcal{X}| = N$ , we can conclude that  $|\mathcal{P}_m| = N/L$ . This completes the proof. ■

### B. Proof of Theorem 1

*Proof.* Since  $N_0, N_1$  are co-prime,  $L = \text{LCM}(N_0, N_1) = N_0 N_1 = N$ . According to Lemma 1, each entry of the  $L$ -point DFT contains exactly one distinct sample of  $\hat{x}(\mathbf{m})$ ,  $\mathbf{m} \in \mathcal{X}$ . Hence,  $\hat{x}(\mathbf{m})$  can be recovered by only one iteration of FPS-SFT. ■

### C. Proof of Theorem 2

*Proof.* For the  $i$ -th,  $0 \leq i \leq T$  iteration of FPS-SFT, the probability of any entry of  $\hat{\mathbf{x}}_r(\mathbf{m})$ ,  $\mathbf{m} \in \mathcal{X}$  containing a significant frequency is  $K_i/N$ , where  $\hat{\mathbf{x}}_r(\mathbf{m})$ ,  $\mathbf{m} \in \mathcal{X}$  is the  $N_0 \times N_1$ -point DFT of (1) after removal of the contribution of the recovered frequencies in the previous  $i$  iterations;  $K_i$  is the expected number of frequencies in  $\hat{\mathbf{x}}_r(\mathbf{m})$ ,  $\mathbf{m} \in \mathcal{X}$ . According to Lemma 1, each entry of the  $L$ -point DFT along the line, i.e.,  $\hat{s}(\alpha, \tau, m)$ ,  $m \in [L]$  contains projection of  $N/L$  distinct entries from  $\hat{\mathbf{x}}_r(\mathbf{m})$ ,  $\mathbf{m} \in \mathcal{X}$ , where  $L = \text{LCM}(N_0, N_1)$ . Since the significant frequencies are assumed to be randomly distributed, if the  $m$ -th entry of  $\hat{s}(\alpha, \tau, m)$ ,  $m \in [L]$  is significant, i.e.,  $|\hat{s}(\alpha, \tau, m)| > 0$ , then the probability of such entry being 1-sparse is  $Q_i = (1 - K_i/N)^M$ , with  $M = N/L - 1$ . Hence, the expected number of the significant frequencies being projected into 1-sparse bins and hence been recovered in the  $i$ -th iteration is  $M_i = K_i Q_i$  with  $K_0 = K$ .

$K_i$  is the expected number of the significant frequencies that 'survived' in the previous  $i$  iterations. The probability of non-recovery (surviving rate) for the  $i$ -th iteration is  $(1 - Q_i)$ . Hence  $K_i = K \prod_{k \in [i]} (1 - Q_k)$ .

The algorithm stops at the  $T$ -th iteration when all the  $K$  significant frequencies are recovered, i.e.,  $\sum_{i \in [T]} M_i \geq K$ . This completes the proof. ■

### D. Proof of Lemma 2

*Proof.* After windowing, the maximum absolute amplitude of the strongest frequency in the  $N_0 \times N_1$ -DFT domain becomes

$$|\hat{a}_w| = \frac{a_{max}}{N} \sum_{\mathbf{n} \in \mathcal{X}} w(\mathbf{n}) = \frac{\|\mathbf{W}\|_1}{N} a_{max}. \quad (23)$$

The noise in the DFT domain becomes

$$\hat{n}_w(\mathbf{m}) = \frac{1}{N} \sum_{\mathbf{n} \in \mathcal{X}} w(\mathbf{n}) n(\mathbf{n}), \quad \mathbf{m} \triangleq [m_0, m_1]^T \in \mathcal{X}. \quad (24)$$

Note that since  $\hat{n}_w(\mathbf{m})$  is a weighted summation of i.i.d. Gaussian noises,  $\hat{n}_w(\mathbf{m})$  is also i.i.d. Gaussian, i.e.,

$$\hat{n}_w(\mathbf{m}) \sim \mathcal{CN}(0, \sigma \|\mathbf{W}\|_2/N). \quad (25)$$

The noise absolute amplitude, i.e.,  $|\hat{n}_w(\mathbf{m})|$  is i.i.d. Rayleigh distributed with mean equal to  $\sigma \hat{n}_w \sqrt{\pi}/2$ , where  $\sigma \hat{n}_w$  is the standard deviation of the real or the imaginary component of  $\hat{n}_w(\mathbf{m})$ , and  $\sigma \hat{n}_w = \sigma \|\mathbf{W}\|_2/(\sqrt{2}N)$ .

Since we need that the side-lobe level of the strongest frequency being below the noise level, i.e.,

$$\frac{\|\mathbf{W}\|_1}{N \rho_w} a_{max} < \frac{\sqrt{\pi} \sigma \|\mathbf{W}\|_2}{2N}, \quad (26)$$

we can conclude that

$$\rho_w > \frac{2\|\mathbf{W}\|_1}{\sqrt{\pi}\|\mathbf{W}\|_2} \sqrt{SNR_{max}}. \quad (27)$$

### E. Proof of (17)

*Proof.* We consider to decode the frequency location component  $m_0$  from a 1-sparse bin. The decoding of  $m_1$  is similar.

With noise, a 1-sparse bin contains the projection of one frequency  $(a_d, \omega) \in \mathbb{S}$  and noise component  $a_n$ , hence (13) becomes

$$\hat{s}(\tau_0, \tau_1) = a_d e^{j2\pi(m_0 \tau_0/N_0 + m_1 \tau_1/N_1)} + a_n(\tau_0, \tau_1), \quad (28)$$

where we have ignored the line slope parameters  $[\alpha_0, \alpha_1]^T$  and the bin number  $m$  for conciseness as they are irrelevant to the decoding process.

The noise component  $a_n$  is due to the noise frequencies that are projected to  $\hat{s}(\tau_0, \tau_1)$ . According to Lemma 1,  $a_n(\tau_0, \tau_1)$  is the summation of  $N/L$  samples of the  $N_0 \times N_1$ -point DFT of the i.i.d noise samples, which can be expressed as

$$a_n(\tau_0, \tau_1) = \sum_{\mathbf{m} \in \mathcal{P}_m} \hat{n}_w(\mathbf{m}) e^{j2\pi\left(\frac{m_0 \tau_0}{N_0} + \frac{m_1 \tau_1}{N_1}\right)}, \quad (29)$$

where  $\mathbf{m} \triangleq [m_0, m_1]^T$ ;  $\mathcal{P}_m \triangleq \{[m_0, m_1]^T : m_0, m_1 \text{ satisfy (10)}\}$  represents the frequency set that projects to the  $m$ -th bin of the DFT of the slice defined in (6) (see Lemma 1).

The same entry (the  $m$ -th bin) of the DFT along the other line with delay  $[(\tau_0 + 1)_{N_0}, \tau_1]^T$  can be decomposed as

$$\hat{s}(\tau_0 + 1, \tau_1) = a_d e^{j2\pi(m_0(\tau_0+1)/N_0 + m_1 \tau_1/N_1)} + a_n(\tau_0 + 1, \tau_1). \quad (30)$$

The frequency location  $m_0$  is decoded as in (14). A graphical representation of the components of (28) and (30) is shown in Fig. 15, from where one can see that the angular error  $\phi_e$  changes with the rotation of the noise components  $a_n(\tau_0, \tau_1), a_n(\tau_0 + 1, \tau_1)$ . The angular error due to decoding, i.e.,  $\Delta\phi = \left| \phi\left(\frac{\hat{s}(\tau_0+1, \tau_1)}{\hat{s}(\tau_0, \tau_1)}\right) - \phi(e^{j2\pi m_0/N_0}) \right|$  reaches maximum when  $\hat{s}(\tau_0, \tau_1), \hat{s}(\tau_0+1, \tau_1)$  are out of phase and perpendicular to  $a_n(\tau_0, \tau_1), a_n(\tau_0+1, \tau_1)$ , respectively, as shown in Fig. 15. In such case, assuming that  $|a_d| \gg |a_n|$ , we can have the following approximation

$$\Delta\phi = 2|\phi_e| = 2\text{asin}(|a_n|/|a_d|) \approx 2|a_n|/|a_d|. \quad (31)$$

Since the localization error  $\Delta u$  due to  $\Delta\phi$  has to be less than  $1/2$ , i.e.,  $\Delta u = \frac{N_0}{2\pi} \Delta\phi < \frac{1}{2}$ , we need that

$$\frac{|a_d|}{|a_n|} > \frac{2N_0}{\pi}. \quad (32)$$

In the following, we derive the distribution of  $|a_n|$  in order to derive the decoding error probability.

From (25) and (29),  $a_n$  also follows a circularly symmetric Gaussian distribution, i.e.,  $a_n \sim \mathcal{CN}(0, \sigma_{a_n})$ , where  $\sigma_{a_n} =$

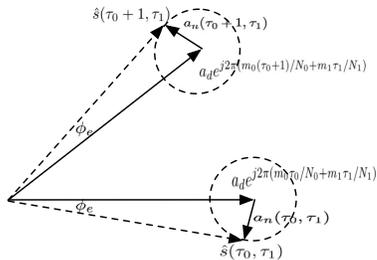


Fig. 15. OFDM-trick for the 1-sparse bin with noise.

$\sigma\|\mathbf{W}\|_2/\sqrt{NL}$ . Hence  $|a_n|$  follows a Rayleigh distribution whose cumulative distribution function can be expressed as  $f_{|a_n|}(x) = 1 - e^{-x^2/(2\sigma_{a_n}^2)}$ ,  $x > 0$ , where  $\sigma_{a_n}^2 = \sigma_{a_n'}^2/2$ .

After windowing, in the DFT domain the highest peak of a frequency with the amplitude of  $a$  becomes  $|a_d| = \|\mathbf{W}\|_1 a/N$ . By substituting into (32), the correct decoding needs that  $|a_n| < \delta_u \triangleq a\pi\|\mathbf{W}\|_1/(2NN_0)$ . Hence, the decoding error for  $m_0$  is upper bounded by

$$P_u = (\sigma_p(1 - f_{|a_n|}(\delta_u)))^2, \quad (33)$$

where  $\sigma_p$  with  $1/2 \leq \sigma_p \leq 1/(2\pi)$  represents the probability of  $a_n(\tau_0, \tau_1), a_n(\tau_0 + 1, \tau_1)$  being out-of-phase; the lower bound of  $\sigma_p$  represents to the probability of  $a_n(\tau_0, \tau_1), a_n(\tau_0 + 1, \tau_1)$  pointing to the opposite direction and the upper bound of  $\sigma_p$  represents  $a_n(\tau_0, \tau_1), a_n(\tau_0 + 1, \tau_1)$  pointing to the opposite direction and perpendicular to  $\hat{s}(\tau_0, \tau_1), \hat{s}(\tau_0 + 1, \tau_1)$ , respectively, as shown in Fig. 15.

Similarly, the decoding error for  $m_1$  is upper bounded as  $P_v = (\sigma_p(1 - f_{|a_n|}(\delta_v)))^2$ , where  $\delta_v \triangleq a\pi\|\mathbf{W}\|_1/(2NN_1)$ . Hence the lower bound of the probability of correct decoding of  $[m_0, m_1]^T$  that projects to a 1-sparse bin is

$$P_w = (1 - P_u)(1 - P_v). \quad (34)$$

Next, the probability of a frequency being project to a 1-sparse bin is determined by

$$P_1 \triangleq (1 - |\mathbb{S}''|/N)^{N/L-1}, \quad (35)$$

where  $L = \text{LCM}(N_0, N_1)$ ;  $\mathbb{S}''$  is the set of remaining frequencies in the signal in each iteration of RFPS-SFT.

The success probability of the  $n_d$ -out-of- $n_s$  voting decoding procedure can be expressed as the complementary cumulative binomial probability of (17), whose success probability of each experiment, i.e., localizing  $(a_d, \omega)$  in each sub-iteration is  $P_1P_w$ . This completes the proof. ■

## APPENDIX B PSEUDO-CODE OF (R)FPS-SFT

The pseudo-code of RFPS-SFT is shown in Algorithm 1. The input and output of the algorithm are following.

**Input:** Input signal function  $r(\mathbf{n}), \mathbf{n} \in \mathcal{X}$ ; window function  $w(\mathbf{n})$ ; data length for the the two dimensions  $N_0, N_1$ ; number of iterations  $T$ ; the threshold of detecting significant frequencies in a slice  $\epsilon$ ; the threshold for 1-sparsity detection  $\gamma$ ;

parameters of  $n_d$ -out-of- $n_s$  detection.

**Output:** the set  $\mathbb{S}$  containing all the significant frequencies.

Note that the line length  $L$  and the set of slope parameters,  $\mathcal{A}$ , can be precomputed for efficiency. FPS-SFT can be viewed as a special case of RFPS-SFT, where the input signal is (1); the window is the rectangular window; and  $\epsilon = 0, \gamma = 0, n_d = n_s = 1$ .

**Algorithm 1** (R)FPS-SFT algorithm

---

```

1: procedure RFPS-SFT( $r, w, N_0, N_1, T, \epsilon, \gamma, n_d, n_s$ )
2:    $L \leftarrow \text{LCM}(N_0, N_1)$ 
3:   Compute the set of  $\mathcal{A}$  defined in Lemma 1
4:    $\mathbb{S} \leftarrow \emptyset$ 
5:   for  $t \leftarrow 1$  to  $T$  do
6:      $\mathbb{S}' \leftarrow \text{SFT-INNER}(r, w, N_0, N_1, \mathbb{S}, L, \epsilon, \gamma, n_d, n_s)$ 
7:      $\mathbb{S} \leftarrow \mathbb{S} \cup \mathbb{S}'$ 
8:   end for
9:   return  $\mathbb{S}$ 
10: end procedure

1: procedure SFT-INNER( $r, w, N_0, N_1, \mathcal{I}_m, L, \epsilon, \gamma, n_d, n_s$ )
2:    $\mathbb{S}, \mathbb{S}' \leftarrow \emptyset$ 
3:    $O(\omega) \leftarrow 0, \omega \in \mathcal{X} \triangleright$  Number of detection for a same
   location
4:   for  $i \leftarrow 1$  to  $n_s$  do
5:     Choose  $\alpha$  uniformly random from  $\mathcal{A}$ 
6:     Choose  $\tau$  uniformly random from  $\mathcal{X}$ 
7:      $\mathbb{S}_0 \leftarrow \text{SLICING}(y, w, N_0, N_1, \mathcal{I}_m, L, \epsilon, \alpha, \tau)$ 
8:      $\mathbb{S}_1 \leftarrow \text{SLICING}(y, w, N_0, N_1, \mathcal{I}_m, L, \epsilon, \alpha, \tau_0)$ 
9:      $\mathbb{S}_2 \leftarrow \text{SLICING}(y, w, N_0, N_1, \mathcal{I}_m, L, \epsilon, \alpha, \tau_1)$ 
10:    for  $(m, \hat{s}_0) \in \mathbb{S}_0, (m, \hat{s}_1) \in \mathbb{S}_1, (m, \hat{s}_2) \in \mathbb{S}_2$  do
11:      if  $\text{Var}(|\hat{s}_0|, |\hat{s}_1|, |\hat{s}_2|) < \gamma$  then  $\triangleright$  1-sparse
12:        Decoding  $(a, \omega)$  according to (14)
13:         $\mathbb{S}' \leftarrow \mathbb{S}' \cup (a, \omega), O(\omega) \leftarrow O(\omega) + 1$ 
14:      end if
15:    end for
16:  end for
17:  for  $(a, \omega) \in \mathbb{S}'$  do
18:    if  $O(\omega) \geq n_d$  then
19:       $\mathbb{S} \leftarrow \mathbb{S} \cup (a, \omega)$ 
20:    end if
21:  end for
22:  return  $\mathbb{S}$ 
23: end procedure

1: procedure SLICING( $y, w, N_0, N_1, \mathcal{I}_m, L, \epsilon, \alpha, \tau$ )
2:    $\mathbb{S} \leftarrow \emptyset$ 
3:    $s(l) \leftarrow y([\alpha_0 l + \tau_0]_{N_0}, [\alpha_1 l + \tau_1]_{N_1}), l \in [L]$ 
4:    $w(l) \leftarrow w([\alpha_0 l + \tau_0]_{N_0}, [\alpha_1 l + \tau_1]_{N_1}), l \in [L]$ 
5:    $s_w(l) \leftarrow s(l) * w(l), l \in [L]$ 
6:    $s_r(l) \leftarrow \text{CONSTRUCTION}(\mathcal{I}_m, N_0, N_1, \alpha, \tau, L)$ 
7:    $d(l) \leftarrow s_w(l) - s_r(l)$ 
8:    $\hat{d}(l) \leftarrow \frac{1}{L} \text{DFT}(d(l))$ 
9:    $\mathbb{S} \leftarrow \{(l, \hat{d}(l)) : |\hat{d}(l)| > \epsilon\}$ 
10:  return  $\mathbb{S}$ 
11: end procedure

1: procedure CONSTRUCTION( $\mathcal{I}_m, N_0, N_1, \alpha, \tau, L$ )
2:    $\hat{s}_r(m) \leftarrow 0, m \in [L]$ 
3:   for  $(a, \omega) \in \mathcal{I}_m$  do
4:      $a' \leftarrow ae^{j2\pi(u\tau_0/N_0 + v\tau_1/N_1)}$ 
5:      $m = [\alpha_0 uL/N_0 + \alpha_1 vL/N_1]_L$ 
6:      $\hat{s}_r(m) = \hat{s}_r(m) + a'$ 
7:   end for
8:    $s_r(l) \leftarrow L \text{IDFT}(\hat{s}_r(m)) \triangleright$  Inverse DFT
9:   return  $s_r(l), l \in [L]$ 
10: end procedure

```

---

## REFERENCES

- [1] 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.
- [2] A. C. Gilbert, P. Indyk, M. Iwen, and L. Schmidt, "Recent developments in the sparse Fourier transform: A compressed Fourier transform for big data," *Signal Processing Magazine, IEEE*, vol. 31, no. 5, pp. 91–100, 2014.
- [3] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Simple and practical algorithm for sparse Fourier transform," in *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '12*, pp. 1183–1194, SIAM, 2012.
- [4] H. Hassanieh, P. Indyk, D. Katabi, and E. Price, "Nearly optimal sparse Fourier transform," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 563–578, ACM, 2012.
- [5] B. Ghazi, H. Hassanieh, P. Indyk, D. Katabi, E. Price, and L. Shi, "Sample-optimal average-case sparse Fourier transform in two dimensions," in *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pp. 1258–1265, IEEE, 2013.
- [6] D. Potts and T. Volkmer, "Sparse high-dimensional FFT based on rank-1 lattice sampling," *Applied and Computational Harmonic Analysis*, 2015.
- [7] S. Pawar and K. Ramchandran, "FFAST: An algorithm for computing an exactly k-sparse DFT in  $O(k \log k)$  time," *IEEE Transactions on Information Theory*, vol. PP, no. 99, pp. 1–1, 2017.
- [8] F. Ong, S. Pawar, and K. Ramchandran, "Fast sparse 2-d dft computation using sparse-graph alias codes," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 4059–4063, IEEE, 2016.
- [9] H. Hassanieh, F. Adib, D. Katabi, and P. Indyk, "Faster GPS via the sparse Fourier transform," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, pp. 353–364, ACM, 2012.
- [10] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi, "GHz-wide sensing and decoding using the sparse Fourier transform," in *INFOCOM, 2014 Proceedings IEEE*, pp. 2256–2264, IEEE, 2014.
- [11] S. Wang, V. M. Patel, and A. Petropulu, "The robust sparse Fourier transform (RSFT) and its application in radar signal processing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 6, pp. 2735–2755, 2017.
- [12] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand, "Light field reconstruction using sparsity in the continuous Fourier domain," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 1, p. 12, 2014.
- [13] H. Hassanieh, M. Mayzel, L. Shi, D. Katabi, and V. Y. Orehkov, "Fast multi-dimensional NMR acquisition and processing using the sparse FFT," *Journal of Biomolecular NMR*, pp. 1–11, 2015.
- [14] L. Kämmerer, "Reconstructing multivariate trigonometric polynomials from samples along rank-1 lattices," in *Approximation Theory XIV: San Antonio 2013*, pp. 255–271, Springer, 2014.
- [15] L. Kämmerer, "Multiple rank-1 lattices as sampling schemes for multivariate trigonometric polynomials," *Journal of Fourier Analysis and Applications*, pp. 1–28, 2016.
- [16] D. Hand and M. S.-W. Chen, "A non-uniform sampling ADC architecture with embedded alias-free asynchronous filter," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 3707–3712, IEEE, 2012.
- [17] F. Engels, P. Heidenreich, A. M. Zoubir, F. K. Jondral, and M. Wintermantel, "Advances in automotive radar: A framework on computationally efficient high-resolution frequency estimation," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 36–46, 2017.
- [18] S. Wang, V. M. Patel, and A. Petropulu, "FPS-SFT: a multi-dimensional sparse Fourier transform based on the Fourier projection-slice theorem," *arXiv preprint arXiv:1711.11407*, 2017.
- [19] S. Wang, V. M. Patel, and A. Petropulu, "Robust sparse fourier transform based on the fourier projection-slice theorem," in *Radar Conference (RadarConf18), 2018 IEEE*, pp. 1427–1432, IEEE, 2018.
- [20] J. O. Smith and X. Serra, *PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation*. CCRMA, Department of Music, Stanford University, 1987.
- [21] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3042–3054, 2010.
- [22] R. M. Mersereau and A. V. Oppenheim, "Digital reconstruction of multidimensional signals from their projections," *Proceedings of the IEEE*, vol. 62, no. 10, pp. 1319–1338, 1974.

- [23] P. Duhamel and M. Vetterli, "Fast fourier transforms: a tutorial review and a state of the art," *Signal processing*, vol. 19, no. 4, pp. 259–299, 1990.
- [24] M. I. Skolnik, *Radar handbook*. McGraw-Hill Education, 3rd ed., 2008.
- [25] M. Schneider, "Automotive radar-status and trends," in *German microwave conference*, pp. 144–147, 2005.
- [26] K. Shirakawa, S. Kobashi, Y. Kurono, M. Shono, and O. Isaji, "3d-scan millimeter-wave radar for automotive application," *Fujitsu Ten Tech. J.*, vol. 38, pp. 3–7, 2013.
- [27] K. H. Rosen, *Elementary number theory and its applications*. Addison-Wesley, 1993.