



# An RFI Mitigation Pipeline for CRAFTS Multi-beam Data Based on Signal Cross-Correlation Function and SumThreshold Algorithm

Zong-Hao Chen<sup>1</sup>, Shan-Ping You<sup>1</sup>, Xu-Hong Yu<sup>1</sup>, Pei Wang<sup>2,3</sup>, Di Li<sup>2,4,5</sup>, Xiao-Yao Xie<sup>1</sup>, Zhi-Jie Liu<sup>1</sup>, Chun-Qing Wang<sup>1</sup>,  
Peng Zeng<sup>1</sup>, and Bin Zhang<sup>1</sup>

<sup>1</sup> Key Laboratory of Information and Computing Science Guizhou Province, Guizhou Normal University, Guiyang 550001, China; [spy@gznu.edu.cn](mailto:spy@gznu.edu.cn),  
[xyx@gznu.edu.cn](mailto:xyx@gznu.edu.cn)

<sup>2</sup> CAS Key Laboratory of FAST, NAOC, Chinese Academy of Sciences, Beijing 100101, China; [wangpei@nao.cas.cn](mailto:wangpei@nao.cas.cn)

<sup>3</sup> Institute for Frontiers in Astronomy and Astrophysics, Beijing Normal University, Beijing 102206, China

<sup>4</sup> Research Center for Intelligent Computing Platforms, Zhejiang Laboratory, Hangzhou 311100, China

<sup>5</sup> NAOC-UKZN Computational Astrophysics Centre, University of KwaZulu-Natal, Durban 4000, South Africa  
*Received 2022 December 23; revised 2023 February 19; accepted 2023 March 8; published 2023 April 26*

## Abstract

The increasing radio frequency interference (RFI) is a well-recognized problem in radio astronomy research. Pulsars and Fast Radio Bursts (FRBs) are high-priority science targets of the ongoing Commercial Radio Astronomy FAST Survey (CRAFTS). To improve the quality of RFI removal in searches of pulsars and FRBs based on CRAFTS multi-beam data, we here propose an intuitive but powerful RFI mitigation pipeline (CCF-ST). The “CCF-ST” is a spatial filter constructed by signal cross-correlation function (CCF) and Sum-Threshold (ST) algorithm. The RFI marking result is saved in a “mask” file, a binary format for RFI masks in PRESTO. Three known pulsars, PSR B0525-21, PSR B0621-04, and PSR J0943 + 2252 from CRAFTS *L*-band 19 beams data are used for evaluation of the performance of CCF-ST in comparison with other methods, such as PRESTO’s “rfifind”, ArPLS-ST and ArPLS-SF. The result shows that CCF-ST can reduce effective data loss rate and improves the detected signal-to-noise ratio of the pulsations by  $\sim 26\%$  and  $\sim 18\%$  respectively compared with PRESTO’s “rfifind” and ArPLS-ST. The CCF-ST also has the advantage of low computational cost, e.g., reducing the time consumption by  $\sim 40\%$  and memory consumption by  $\sim 90\%$  compared with ArPLS-SF. We expect that the new RFI mitigation and analysis toolkit (CCF-ST) demonstrated in this paper can be applied to CRAFTS and other multi-beam telescope observations to improve the data quality and efficiency of pulsar and FRB searches.

*Key words:* methods: data analysis – (stars:) pulsars: general – stars: neutron

## 1. Introduction

Due to the high sensitivity of modern telescopes and human activities, radio frequency interference (RFI) is an increasing problem for searches in radio astronomy (Briggs et al. 2000; Baan et al. 2004). As for FAST, it is by far the largest and the most sensitive telescope in the low-frequency radio band (Nan et al. 2011), which is also vulnerable to RFI. The enormous data volume of Commercial Radio Astronomy FAST Survey (CRAFTS) roughly 10 PB per year (Li et al. 2018) poses an additional challenge. Therefore, it is critical to find an effective and efficient RFI mitigation method for CRAFTS data analysis.

Many RFI mitigation methods have been proposed (Fridman & Baan 2001; Kocz et al. 2010; Offringa et al. 2010; Akeret et al. 2017; Baan 2019). The threshold-based methods are the most widely used due to their simplicity and effectiveness (Akeret et al. 2017; Zeng et al. 2020). A typical threshold-based method in PRESTO is `rfifind` (Ransom et al. 2002), which we use in our pulsar search pipeline (You et al. 2021), and it sets thresholds based on time series and total spectral

power. Bhat et al. (2005) developed an algorithm by using a series of thresholding based on the median filter for simultaneous observations with Arecibo Observatory and the Green Bank Telescope (GBT). Zeng et al. (2020) designed a scheme for RFI mitigation based on the SumThreshold algorithm (Offringa et al. 2010) after removing the estimated baseline by using asymmetrically reweighted penalized least squares (ArPLS) method, and tested its effectiveness in FAST data. However, threshold-based methods do not work in weak and long-lasting RFI, and difficult to set a time and frequency resolution for the detection of varieties of RFI (Baan et al. 2004). The up-to-date methods are based on machine learning, such as Akeret et al. (2017) used U-Net, a special Convolutional Neural Network (CNN), to detect and mitigate RFI in 2D time-ordered data from a single-dish radio telescope. In recent years, many of these types of methods have been proposed (Burd et al. 2018; Kerrigan et al. 2019; Arzaga & Lynch 2021). However, these methods require training models from labeled data, which are time-consuming to acquire. They are novel and new trends but not widely used yet.

**Table 1**  
The Basic Information of a Single PSRFITS File from FAST 19 Beam Receiver

Format	Size	Observation Time	Subints	Spectra Per Subint	Channels	polns	Channel Width
8 bit	2 GB	12.8849 s	256	1024	4096	2	0.122 MHz
2 bit	140 MB	12.8849 s	128	1024	4096	2	0.122 MHz

It is important to consider the loss of the signal of interest (SOI) when assessing the results of RFI mitigation. (Fridman & Baan 2001). We found that the existing widely used RFI mitigation methods have varying degrees of misjudgment.

Kocz et al. (2010) developed spatial filtering techniques based on the eigen decomposition of the covariance matrix formed from multi-beam data. Wang et al. (2022) proposed an RFI mitigation pipeline based on the spatial filter (Kocz et al. 2010) with ArPLS (ArPLS-SF), which can filter out the common RFI from multi-beam data and reduce misjudgment. However, in our pulsar search pipeline, we have found that ArPLS-SF is time-consuming and takes up too much memory (more than 8GB) at high temporal resolution. We therefore considered the construction of a spatial filter with high accuracy and low computational cost for CRAFTS multi-beam data. Astronomical point sources appear in beams of the same direction, and RFI appears in beams of both the same and different directions. Therefore, RFI can be filtered out by the similarity of the signals from different beams. In signal processing, cross-correlation function (CCF) is a measure of the similarity of two signals (Smith 1997), has been widely used to construct correlation filters and applied in many areas (Henriques et al. 2014; Kiani Galoogahi et al. 2014; Wang et al. 2017) because of its simplicity and effectiveness, and it is suitable for acceleration by graphic processing unit (GPU) (Kapinchev et al. 2015). We proposed to use CCF to calculate the RFI quantized values from the original data with GPU. SumThreshold (Offringa et al. 2010) is a widely used method in RFI mitigation because it considers a time-frequency adjacency of RFI. We choose the SumThreshold algorithm to flag the RFI in RFI quantized values.

The rapid growth of pulsar discoveries from CRAFTS promotes the international cooperation between CRAFTS and other telescopes, e.g., Parkes, Effelsberg, Green Bank radio telescopes, Arecibo telescope, and Fermi-LAT gamma-ray telescope, for pulsar follow-up and timing studies (Cameron et al. 2020; Wang et al. 2021a, 2021b; Cruces et al. 2021; Wen et al. 2022; Miao et al. 2023). More than 170 new pulsars have been discovered,<sup>6</sup> including 40 millisecond pulsars (Wang et al. 2021a; Miao et al. 2023), long-period pulsars (Zhang et al. 2019; Tedila et al. 2022). The CCF and SumThreshold algorithm (CCF-ST) will significantly improve the quality of

RFI removal on CRAFTS multi-beam data and help to make more significant discoveries.

The rest of this paper proceeds as follows. Section 2 introduces the data set we used in the experiment. Section 3 describes the pipeline of CCF-ST. Section 4 presents our experiment and results. Our method is compared with the rfind, ArPLS-ST and ArPLS-SF. Finally, we draw our conclusions in Section 5.

## 2. Experimental Data from FAST 19 beam Receiver and Their Characteristics

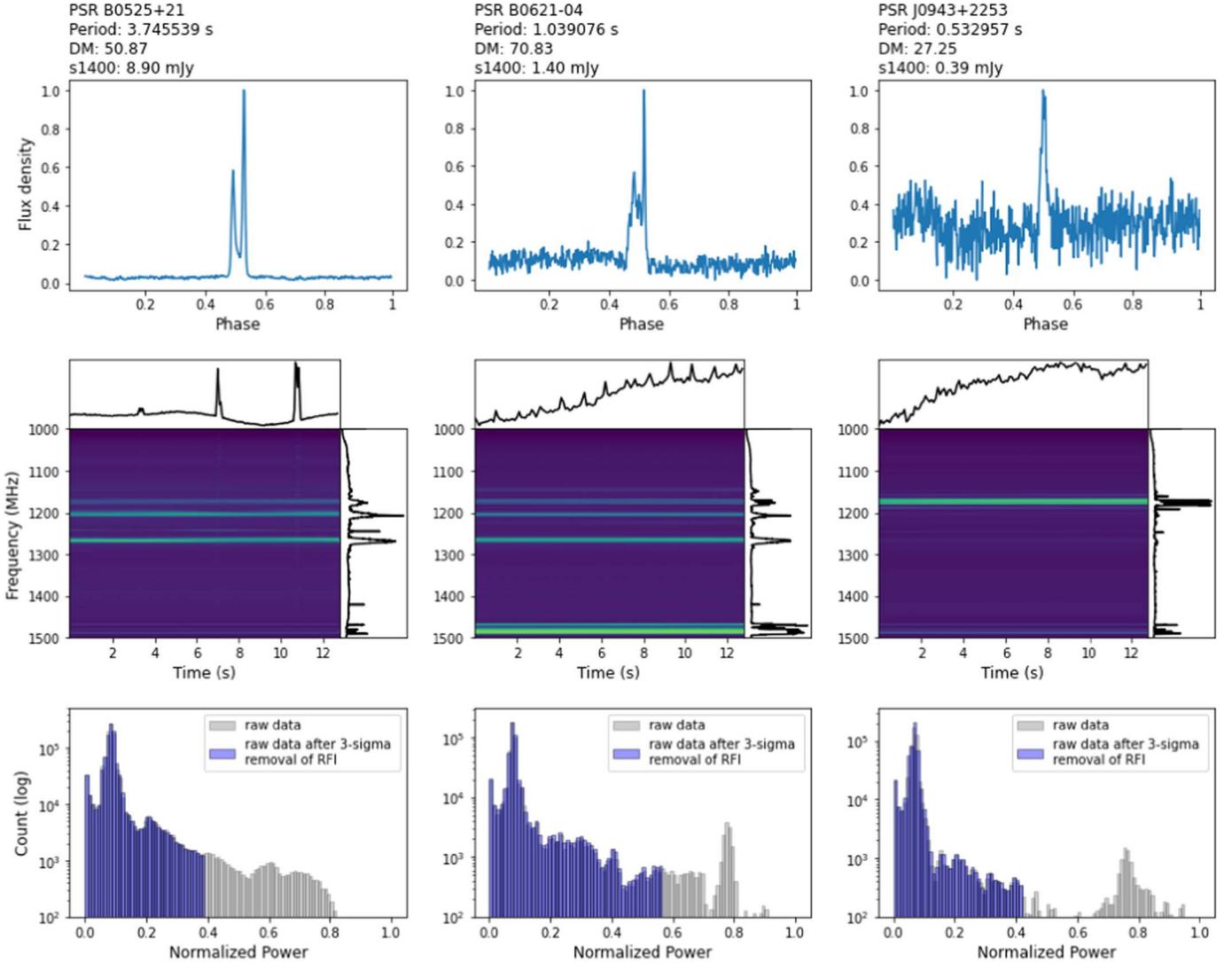
CCF-ST is designed for multi-beam data and we use the PSRFITS format (Hotan et al. 2004) data from CRAFTS 19 beam receiver. The basic information of CRAFTS 19 beam data is shown in Table 1. Taking 8 bit data as an example, there are 256 sub-integrations in each file, and 1024 spectra with 4096 channels (a time-frequency image with  $4096 \times 1024$  pixels) in each sub-integration.

To investigate the effectiveness of CCF-ST on pulsar observations with different flux density, rotational parameter and RFI environment, we selected three data sets of known pulsars, i.e., PSR B0525-21, PSR B0621-04, and PSR J0943 + 2253. As shown in Figure 1, the three pulsars have different properties, e.g., flux density, period and dispersion measure (DM). Flux density is the radiation energy received per unit time, at a central frequency 1.4 GHz. Period is the spin period time of pulsar. DM is the total column density of free electrons in the light of sight between the source and the observer. These parameters are from psrcat.<sup>7</sup>

RFI in FAST data mainly falls into three types (Jiang et al. 2020): The first one is narrow-band RFI, which comes from many sources, even from the instruments themselves. However, the narrow-band RFI has been much reduced after EM shielding was installed in 2019. The second is 1 MHz wide RFI, which is caused by a standing wave, and their distribution is not regular at time-frequency in the *L*-band. The last one is fixed frequency RFI, which is from satellite or civil aviation from the sky and usually has a fixed frequency in wider distribution. Besides, blob RFI with short durations and small-bandwidth from unknown sources also exists in FAST data (Zeng et al. 2020).

<sup>6</sup> [http://groups.bao.ac.cn/ism/CRAFTS/202203/t20220310\\_683697.html](http://groups.bao.ac.cn/ism/CRAFTS/202203/t20220310_683697.html)

<sup>7</sup> <https://www.atnf.csiro.au/research/pulsar/psrcat/>



**Figure 1.** A gallery of emission properties and noise background environment for data recording for three known pulsars. The pulse profiles under an unified integrated time of 12.8 s are in FAST drift-scan mode. Three pulsars have either different noise levels or different flux densities, respectively. The middle panel shows the dynamic spectra of the pulsars. The bottom panel indicates the histogram of data intensity, the gray and purple regions separately represent the distribution of the raw data and the effect of RFI removal by using  $3\sigma$  threshold, the remaining gray area represents the part dominated by RFIs. The RFI percentage of the three selected pulsar data sets is different.

### 3. The Proposed Pipeline

RFI as a near-field expansive source can be received on multiple pointing, while pulsars as a typical point source, their signals can be received only on specific pointing. Therefore, we can distinguish RFI and pulsar signals by the similarity of signals from FAST 19 beams. CCF is a measure of the similarity of two signals (Smith 1997). We proposed an RFI mitigation pipeline for CRAFTS multi-beam data based on CCF-ST. The workflow of CCF-ST is shown in Figure 2, including three main parts: “calculate correlation coefficient by

CCF between beams for each channel,” “calculate RFI quantized values by correlation coefficient” and “flag RFI by SumThreshold algorithm.” The RFI marking results are saved in “mask” files, a binary format for RFI masks in PRESTO. In addition, we designed a parallel algorithm based on GPU for CCF calculation between different beams.

#### 3.1. Pretreatment

To make the program run in the production environment, we need to consider the limitation of memory and the complexity of

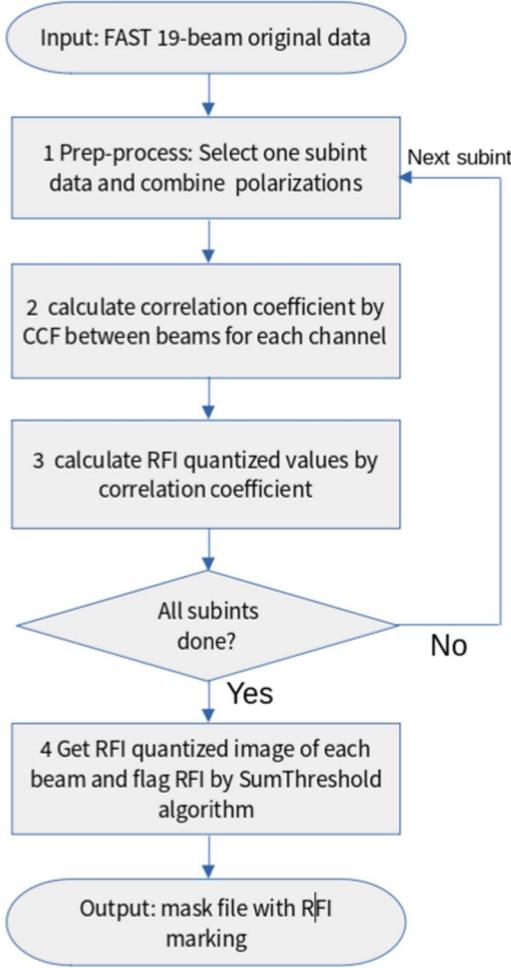


Figure 2. Workflow of the CCF-ST pipeline.

computing. In the pre-processing, the data of two polarizations are averaged. For each beam, each channel, a sub-integration data is read for post-processing. Taking 8 bit data as an example, the size of the raw data is  $19 \times 256 \times 1024 \times 2 \times 4096$ . After pre-processing, only data with a size of  $19 \times 1024 \times 4096$  needs to be processed in each loop. In addition, there is no zero-padding operation during the calculation, resulting in higher execution efficiency.

### 3.2. Cross-correlation Operations

Before RFI detection, we need to convert the raw data into RFI quantized image. We call this series of operations cross-correlation operations.

#### 3.2.1. Calculate Correlation Coefficient

We use CCF (Bracewell & Bracewell 1986) to calculate the correlation coefficient and express the similarity of signals. The

signal cross-correlation function is defined as

$$C_{ij}(f, \tau) = \sum_t M_i(f, t) * M_j(f, t + \tau) \quad (1)$$

In our situation,  $i$  and  $j$  represented two different beams,  $f$  represented the observation frequency,  $t$  is the observation time, and  $\tau$  is the time delay. We can get the cross-correlation sequence (vector) of two beams by using Equation (1) to calculate the sliding point product:

$$\vec{S}_{ij}: (C_{ij}(f, \tau_1), C_{ij}(f, \tau_2), \dots, C_{ij}(f, \tau_n)) \quad (2)$$

then calculate the 2-norm of vector  $\mathbf{S}$  and it is regarded as the correlation coefficient between beam  $i$  and beam  $j$ :

$$Q_{ij} = \|\vec{S}_{ij}\|_2 \quad (3)$$

#### 3.2.2. Calculate RFI Quantized Values

For each channel of each sub-integration, we can get the result of the cross-correlation coefficient ( $Q$ ) as follows:

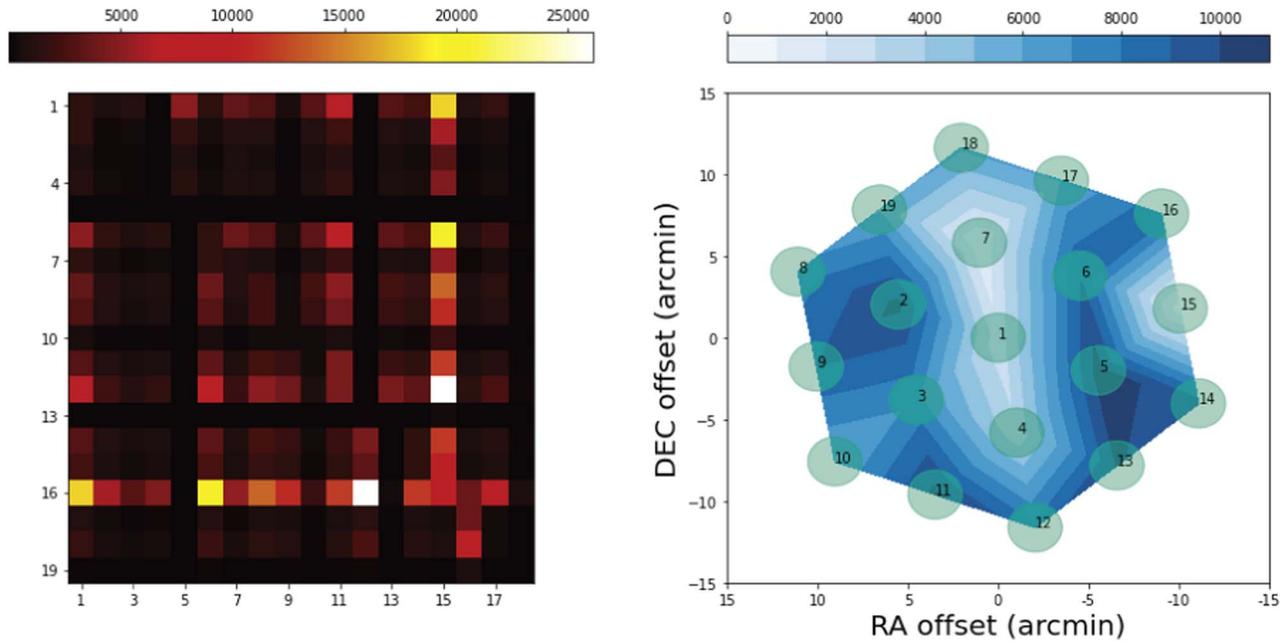
$$\begin{pmatrix} Q_{1,2} & Q_{1,3} & Q_{1,4} & \dots & Q_{1,n-1} & Q_{1,n} \\ Q_{2,1} & Q_{2,3} & Q_{2,4} & \dots & Q_{2,n-1} & Q_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Q_{n-1,1} & Q_{n-1,2} & Q_{n-1,3} & \dots & Q_{n-1,n-2} & Q_{n-1,n} \\ Q_{n,1} & Q_{n,2} & Q_{n,3} & \dots & Q_{n,n-2} & Q_{n,n-1} \end{pmatrix} \quad (4)$$

Taking beam  $i$  as an example, we have two strategies for obtaining the RFI quantized value. First,  $Q_{i,k}$  ( $k = 1, 2, \dots, n, k \neq i$ ) that are lower than the average value of  $Q$  are added to RFI quantized value. For those  $Q_{ik}$  that are higher than the average, we compare  $Q_{ij}$ ,  $Q_{ik}$  and  $Q_{jk}$ , if the three numbers are close ( $|Q_{ij} - Q_{ik}| \leq cQ_{jk}$ ,  $c$  is a constant and we defined it as 3), it is considered that this is caused by the RFI present in beam  $i$ , beam  $j$ , and beam  $k$ , and  $\text{Count}_{i,k}$  is defined to count this situation if  $\text{Count}_{i,k}$  is greater than threshold  $C$  (we use 14),  $Q_{i,k}$  can be added to RFI quantized value. The RFI quantized value of beam  $i$  can express as follows:

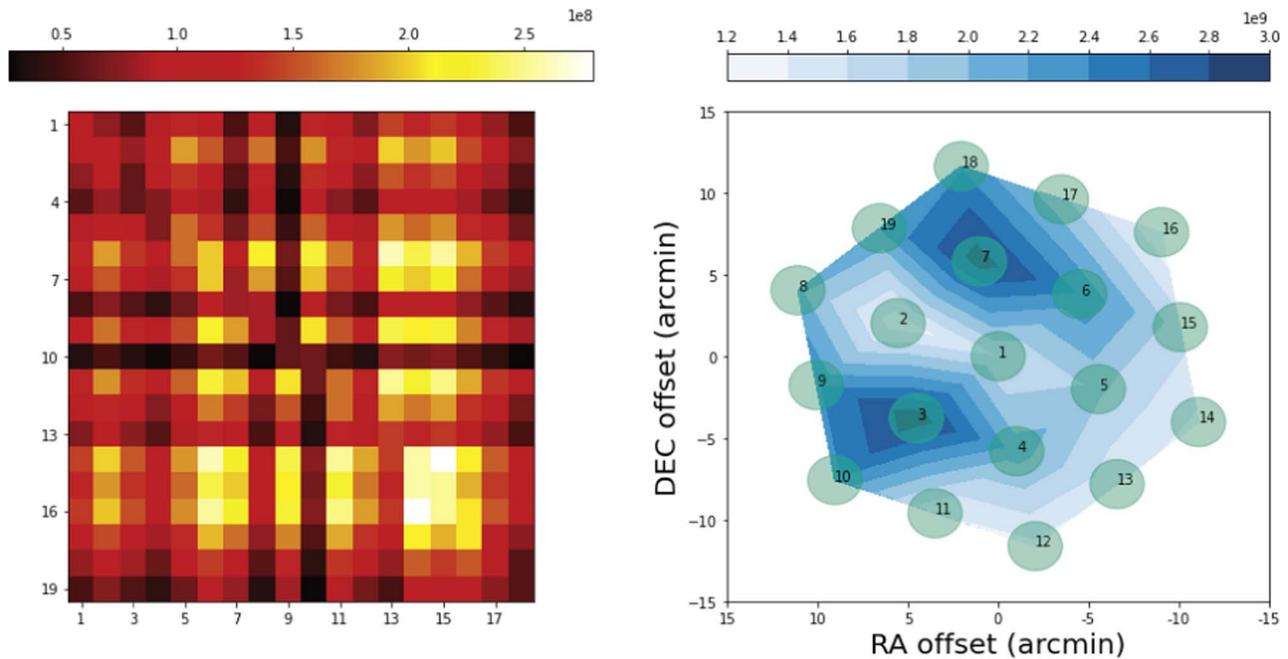
$$I_i = \sum_{k,k \neq i}^n Q_{ik}, Q_{ik} \leq \frac{\sum_i \sum_{j,j \neq i}^n Q_{ij}}{n \times (n-1)} \|\text{Count}_{i,k} > C \quad (5)$$

$$\text{Count}_{i,k} = \sum_{j,j \neq i,k}^n 1, |Q_{ij} - Q_{ik}| \leq cQ_{jk} \quad (6)$$

Figure 3 demonstrates the features of spatial signal cross-correlation in the FAST-CRAFTS mode. Figure 3(a) shows that when there is only pulsar signal in the frequency channel, the cross-correlation coefficient between 19 beams is relatively low, resulting in a small RFI quantized value. On the contrary, in Figure 3(b), when there is RFI in the channel,



(a) channel 50 (pulsar)



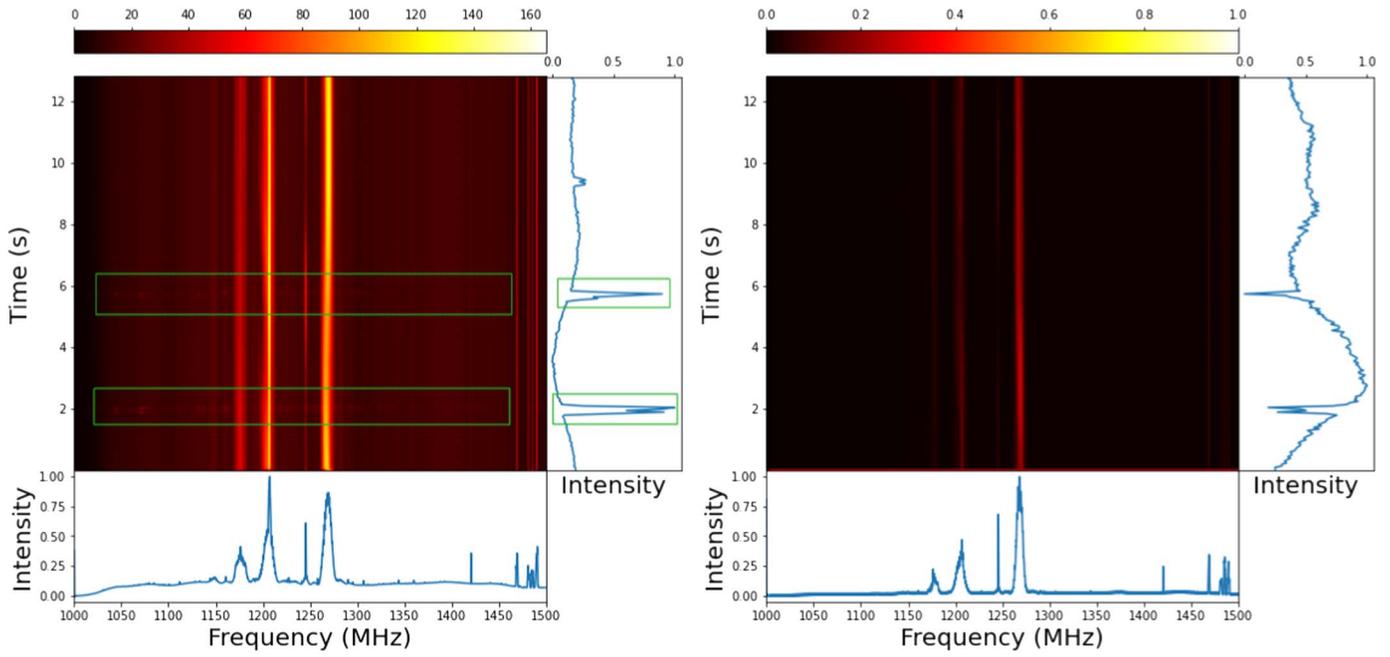
(b) channel 2183 (RFI)

**Figure 3.** Comparison of cross-correlation coefficient (left) and RFI quantized value (right) of pulsar and RFI in different frequency channels at the same time on the data set containing PSR B0525-21.

the cross-correlation coefficient is higher than the result in Figure 3(a), thus obtaining a larger RFI quantized value.

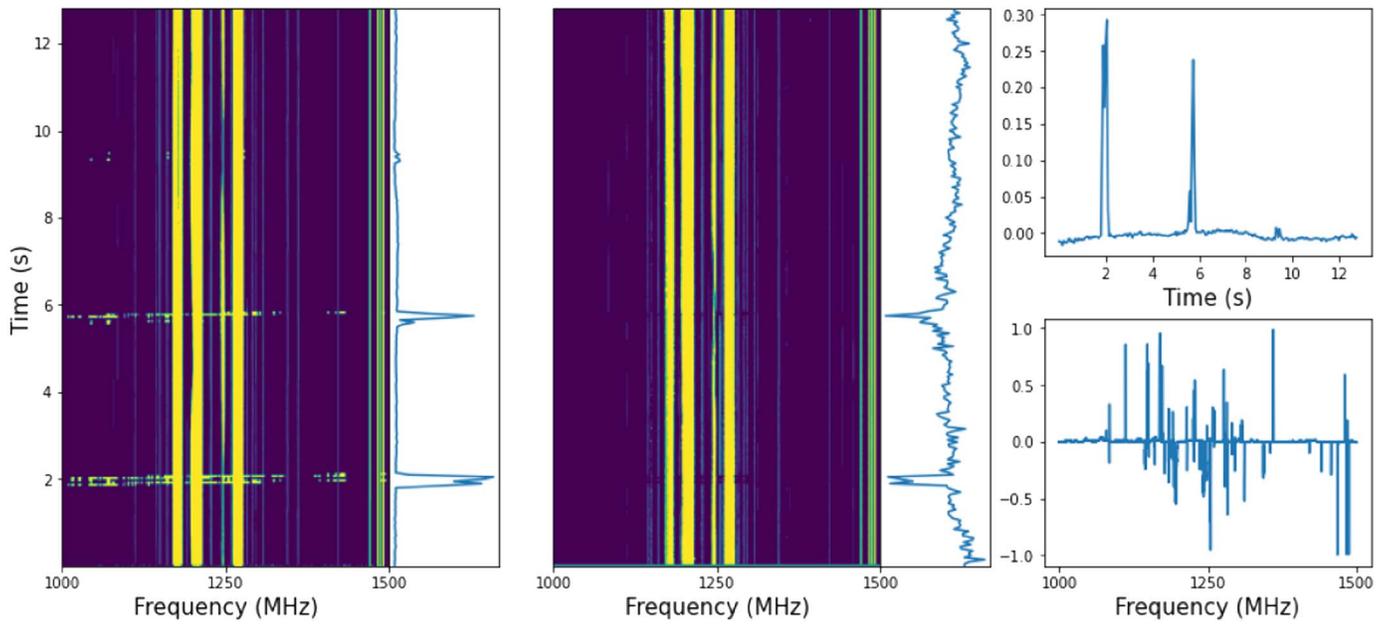
Figure 4(a) shows the time-frequency image from raw PSRFITS data containing PSR B0525-21, the pulsar's signals

are obvious in the green area, and Figure 4(b) shows the corresponding RFI quantized image, which filters pulsar signals while retaining RFI information, and it is advantageous to SumThreshold for RFI detection.



(a) time-frequency image of original data

(b) RFI quantized image



(c) RFI flagged by SumThreshold

**Figure 4.** Comparison of the time-frequency image from original PSRFITS data and RFI quantized image (a and b). The green area shows strong pulsar signals from PSR B0525-21. Comparison of the RFI flagging results of SumThreshold on the original time-frequency image and RFI quantized image respectively, and their difference in the time domain and frequency domain integrations (c).

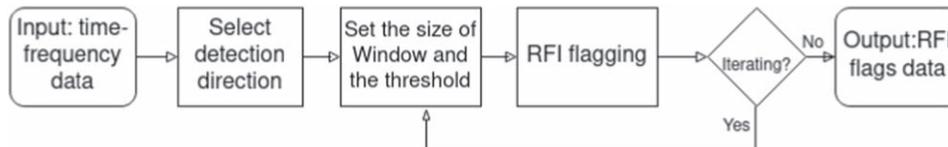


Figure 5. The flowchart of the SumThreshold algorithm.

### 3.2.3. Parallelization

The calculation of cross-correlation is similar to convolution but no need to do a time-flip (Smith 1997), and it is a time-consuming calculation. Therefore, parallelization is necessary, and there are two ways to realize parallelization, one is based on the CPU, and the other is based on the heterogeneous structure of the CPU and GPU. The key in solving the parallel problem is to decompose the computations into a set of independent tasks (Kumar 2002).

For the CPU, the computations can decompose as task parallelism based on the independence of beam and channel. We used the OPENMP library with C-language to realize parallelism, and used OpenBLAS,<sup>8</sup> an optimized Basic Linear Algebra Subprograms (BLAS) library, to accelerate the calculation of CCF between different beams.

For the heterogeneous structure of CPU and GPU, we used Compute Unified Device Architecture<sup>9</sup> (CUDA), a platform for heterogeneous computing. Based on CUDA, thread hierarchy is decomposed into blocks of threads and grids of blocks. Organizing threads is the most important part of CUDA programming, which is based on the independence of data. In our situation, except for each beam and channel, each element in the sequence also has no data dependencies. There are 4096 channels and 19 beams and each beam shall be cross-correlated with the other 18 beams. Therefore, we chose a 3D grid (4096, 19, 18) to organize threads to unrolling loops in CPU-based version.

Table 2 shows the results of execution time in different environments. The GPU-based version has significantly improved and performs  $\sim 3.5$  times faster than the OpenBLAS-Parallelized version, which performs  $\sim 25$  times faster than python-sequentialized version in the same environment.

### 3.3. RFI Flagging Based on SumThreshold

The SumThreshold method is a variant of the VarThreshold method (Offringa et al. 2010), a kind of combinatorial thresholding method. The advantage of this method over the traditional threshold method is that it makes use of the neighborhood of pixels. In order to detect RFI more accurately, the SumThreshold method uses a bidirectional detection strategy based on time and frequency and detects a series of samples with higher than expected values (Offringa et al. 2010). The flowchart of the SumThreshold algorithm is shown in Figure 5.

Table 2

Average Execution Time to Generate One Mask File

Environment	Method	Execution Time
AMD R4600U (6 cores), python, numpy	CPU sequential	$\sim 25$ minutes
AMD R4600U, C, OpenBLAS	CPU sequential	$\sim 7$ minutes
AMD R4600U, C, OpenBLAS	CPU parallel	$\sim 1$ minutes
AMD 2700X (8 cores), C, OpenBLAS	CPU parallel	$\sim 40$ s
AMD 2700X, NVIDIA GTX 1080 GPU (2560 CUDA cores), CUDA	CPU + GPU parallel	$\sim 14$ s

Figure 4(c) shows that the SumThreshold method works on original data and RFI quantized image respectively, and there is an obvious misjudgment for strong pulsar signals on the original data, while there is no obvious misjudgment on RFI quantized image.

## 4. Experiment and Results

Three known pulsars, PSR B0525-21, PSR B0621-04, and PSR J0943 + 2253 from CRAFTS *L*-band 19 beam data are used to evaluate the performance of CCF-ST and other methods, e.g., rfind,<sup>10</sup> ArPLS-ST<sup>11</sup> and ArPLS-SF.<sup>12</sup> We use PRESTO’s “prepfold” to test the effect of four RFI mitigation methods on pulsar periodic searches. Table 3 shows the parameters we use in prepfold. The four applications are run on the same server with an AMD 2700X CPU and an NVIDIA GTX 1080 GPU. The following describes the experimental steps and presents the experimental results.

### 4.1. Experimental Steps

First, we use the four methods to flag the RFI in data sets and generate mask files respectively. For rfind, we use a time resolution of 1 second, which is recommended in the PRESTO manual. As for the other three methods, they are all based on a time resolution of 0.05 s (one sub-integration). And then, we use PRESTO’s “prepfold” to fold data with the mask files generated by the four methods as parameters. The specific

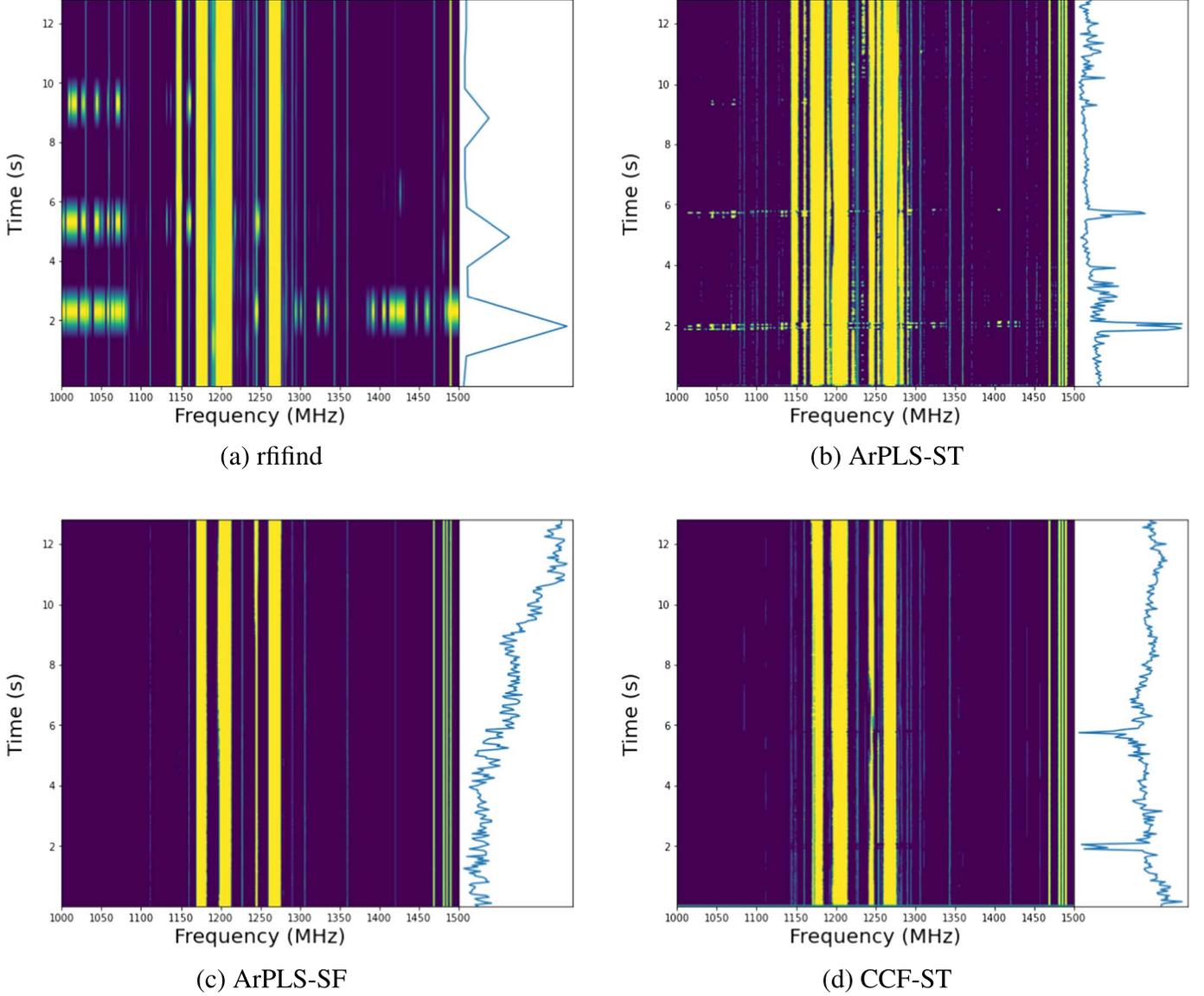
<sup>8</sup> <https://github.com/xianyi/OpenBLAS>

<sup>9</sup> <https://developer.nvidia.com/cuda-toolkit>

<sup>10</sup> <https://github.com/scottransom/presto>

<sup>11</sup> <http://zmtt.bao.ac.cn/GPPS/RFI>

<sup>12</sup> <https://github.com/wangy-nao/ArPLS-SF>



**Figure 6.** The RFI flagging results for the four methods on the data set containing PSR B0525-21.

parameters we use are shown in Table 5. The prepfold uses reduced chi-squared ( $\chi^2_{\text{reduced}}$ ) to determine pulsation significance. This technique is often known as “epoch folding” (Leahy et al. 1983):

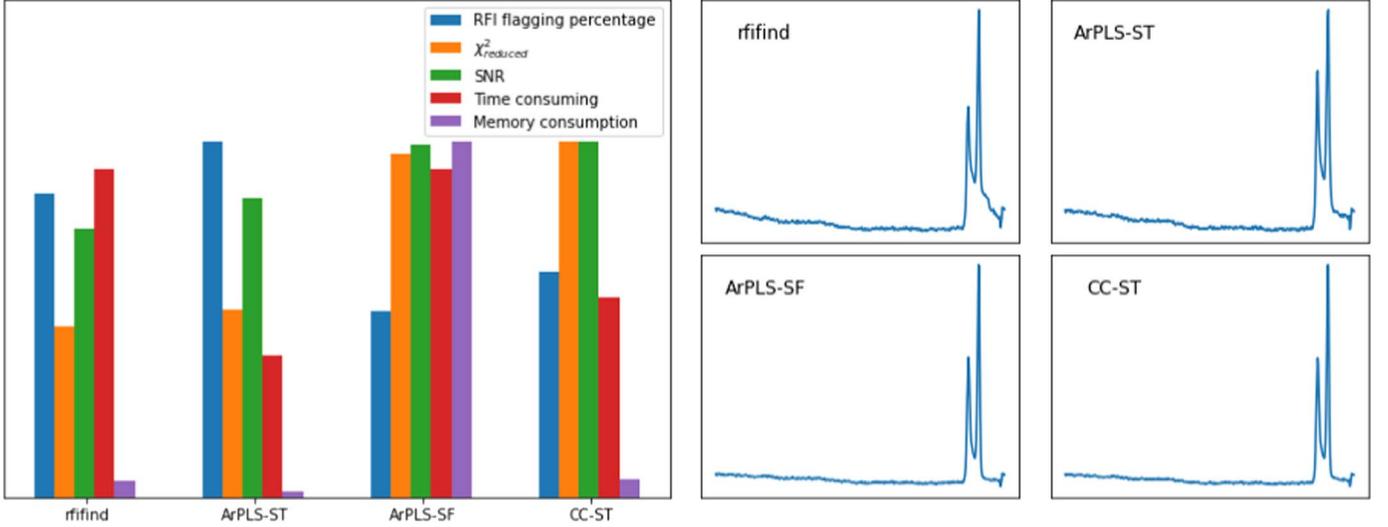
$$\chi^2_{\text{reduced}} = \sum_{j=1}^n \frac{(R_j - R)^2}{\sigma_j^2}, \quad (7)$$

where  $R = N_y/T'$ ,  $\sigma_j^2 = R/T_j$ ,  $N_y$  is the total number of photons, and  $T_j$  is the total integration time for the  $j$ th pulse phase bin, and  $T'$  is the sum of  $T'_j$ . The quantity  $R_j$  is the counting rate in the  $j$ th pulse phase bin.

## 4.2. Results

We chose four metrics to measure the performance:  $\chi^2_{\text{reduced}}$  and SNR from prepfold, as well as program execution time and memory consumption.

Figure 6 compares the RFI flagging results of four methods on the data set containing PSR B0525-21. The pulsar signals appear between 2 to 4 s, 6 to 8 s, and 10 to 12 s. The rfind method mainly identifies strong narrow-band and/or short-duration broad-band RFI by the threshold measurement of time series and total FFT spectrum power from a single data stream (Ransom 2001). ArPLS-ST first corrects the baseline by ArPLS and then uses the combined thresholds to detect band



**Figure 7.** A Comprehensive comparison of four methods on the data set containing PSR B0525-21.

**Table 3**  
Parameters We Use in Prepfold

Parameter	Explaining
$p$	The folding period (s)
dm	Te central DM of search
mask	File containing masking information
$n$	The number of bins in the profile (512)
nosearch	Show but do not search the p/pdot and/or DM phase spaces

**Note.** Pulsars in the data set are known sources. We need to input their period and DM, and use the “nosearch” parameter to reduce the running time. The “mask” parameter comes from the mask files generated by four methods.

**Table 4**  
Comparison of Average Execution Time to Generate a Mask File and the Memory Consumption for Four Methods

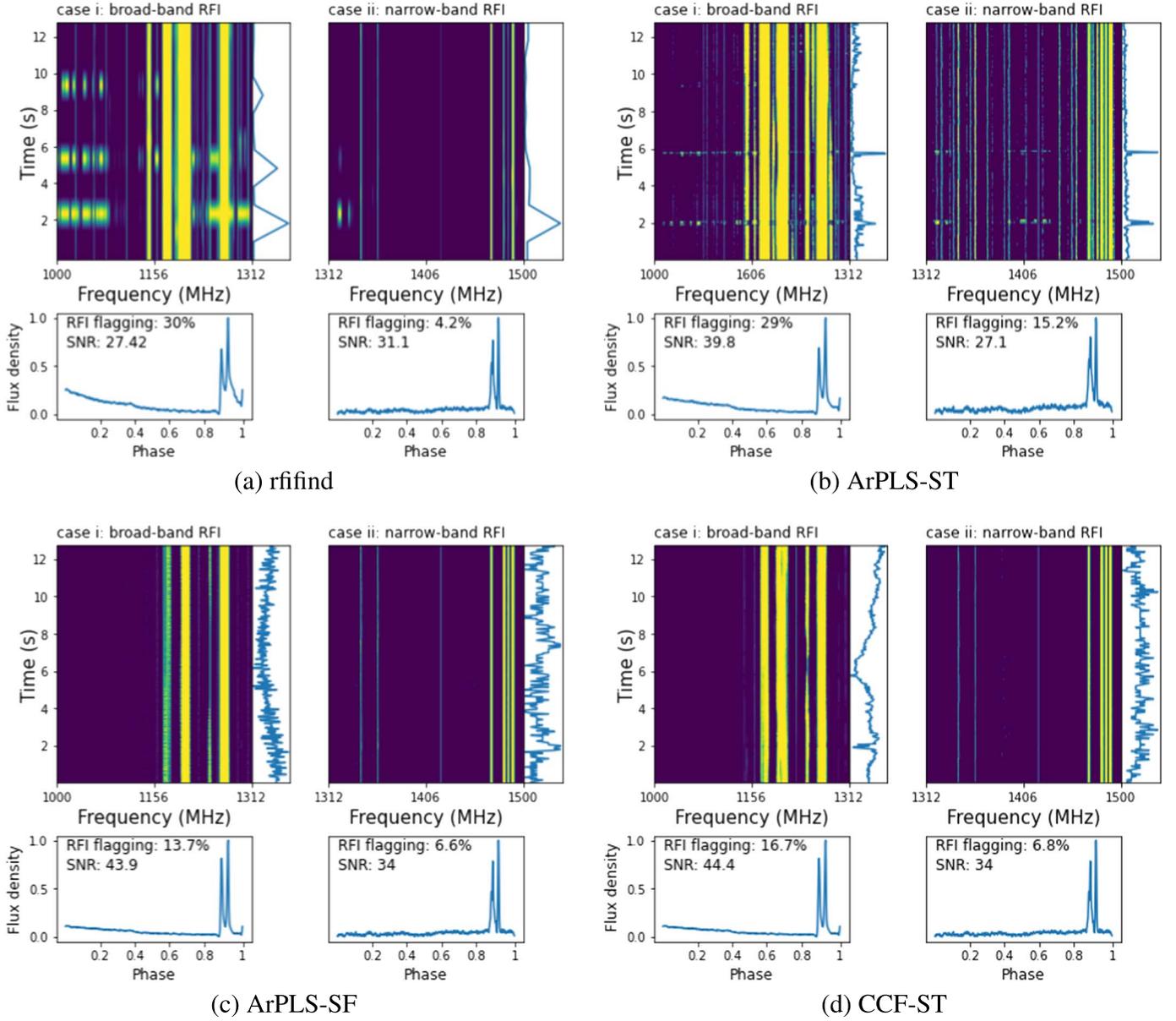
Method	Execution Time (s)	Memory Consumption (MB)
rfifind	~25	~360
ArPLS-ST	~10	~120
ArPLS-SF	~23	>8000
CCF-ST	~14	~420

and blob RFI in time and frequency dimensions (Zeng et al. 2020). Both of the ArPLS-ST and the rfifind are threshold-based methods and are applied to a single-beam data stream. The corresponding data will be masked as RFI, if the strength of the value is greater than the threshold, resulting in the strong astronomical signal being misclassified (Figures 6(a) and (b)).

ArPLS-SF first corrects the baseline by ArPLS, then constructs the RFI projection matrix using the covariance matrix, and eigenvectors solved by SVD. Finally, RFI is flagged on residuals by k-sigma (Wang et al. 2022). CCF-ST uses CCF to calculate the correlation between 19 beam data and then constructs the RFI quantized images, and finally uses SumThreshold to mark RFI on the images (Section 3). ArPLS-SF uses matrix diagonalization to solve the eigenvalue decomposition, which ignored the contribution from the non-diagonal components. While it reduces operational complexity, it also reduces the sensitivity to RFI inspection. Compared to SVD in ArPLS-SF, the CCF calculation between 19 beams in CCF-ST is full-rank and can reveal more faint RFI details (Figures 6(c), (d) and 7).

The above shows the visual differences between the four methods of RFI flagging. Figure 7 shows the comprehensive comparison. Due to misjudgment, rfifind and ArPLS-ST get a lower SNR and  $\chi^2_{\text{reduced}}$  than ArPLS-SF and CCF-ST. However, CCF-ST has faster execution efficiency and lower memory consumption than ArPLS-SF (Table 4).

The two typical types of RFI that affect pulsar searches in CRAFTS are fixed frequency (broad-band) RFI and narrow-band RFI. In the data set containing PSR B0525-21, the broad-band RFI is mainly located below 1300 MHz while narrow-band RFI is mainly located above 1300 MHz. In order to investigate the effectiveness of four methods on two types of RFI, we split the raw data into two parts bounded by 1300 MHz. Then we use the four algorithms to perform RFI mitigation in these two parts of the data. Finally, we fold data to get the pulse profile and SNR respectively by using PRESTO’s “prepfold.” Figure 8 shows the results. The experimental result shows that rfifind and ArPLS-ST have



**Figure 8.** Comparison of broad-band and narrow-band RFI removing effects on four algorithms.

**Table 5**

Comparison of SNR and  $\chi^2_{\text{reduced}}$  in Prepfold Results for Four Methods (the Bold Values Indicate the Maximum Value in the Column)

Method	B0525-21		B0621-04		J0943 + 2253	
	$\chi^2_{\text{reduced}}$	SNR	$\chi^2_{\text{reduced}}$	SNR	$\chi^2_{\text{reduced}}$	SNR
rfind	1015.6	26.5	101.2	65.5	6.7	56.5
ArPLS-ST	1120.1	29.5	94.7	65.8	6.2	59.6
ArPLS-SF	2078.5	34.7	98.7	<b>66.1</b>	<b>6.9</b>	59.8
CCF-ST	<b>2124.1</b>	<b>35.1</b>	<b>106.4</b>	65.6	6.8	<b>61.4</b>

misjudged pulsar signal as RFI in both cases. For broad-band case, we proposed that the CCF-ST algorithm gives a more sophisticated and accurate RFI inspection and thus get a higher SNR of pulse profile than using ArPLS-SF.

Table 5 shows the folding results for three pulsars using four methods. It can be seen that multi-beam-oriented methods (ArPLS-SF and CCF-ST) have obvious advantages on strong signals (B0525-21), and the four methods have close results on the other two data sets. In general, CCF-ST has better results in

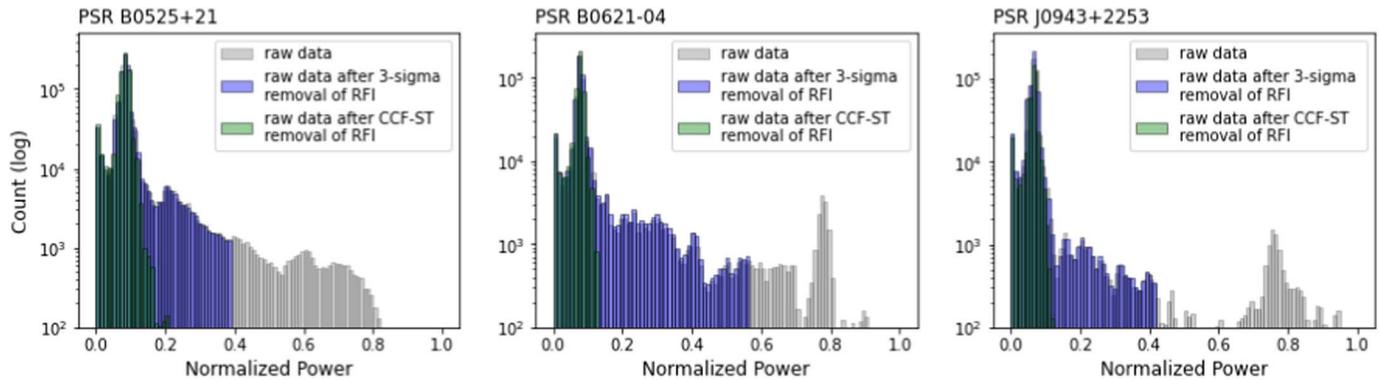


Figure 9. Data statistics.

our experiment. Figure 9 shows the data statistics by using CCF-ST. Compared with Figure 1, the distribution of the data is approximately Gaussian distribution after the RFI is removed by CCF-ST.

## 5. Conclusions

RFI mitigation is an important problem in CRAFTS. We here proposed a new RFI mitigation pipeline for CRAFTS multi-beam receiver, constructed by the signal CCF and SumThreshold algorithm, namely CCF-ST. The conclusions are summarized as follows:

1. CCF-ST can reduce effective data loss rate and misjudgment. We used three known pulsars, PSR B0525-21, PSR B0621-04, and PSR J0943 + 2253 to measure the pulsar search results. Compared with rfind and ArPLST-ST, CCF-ST increases the SNR of pulsation detection by  $\sim 26\%$  and  $\sim 18\%$  respectively, and the  $\chi^2_{\text{reduced}}$  both increases by  $\sim 50\%$ .
2. CCF-ST is suitable for GPU acceleration with the advantage of high efficiency and low memory cost, which reduces the time consumption by  $\sim 40\%$  and memory consumption by  $\sim 90\%$  compared with ArPLS-SF.
3. CCF-ST provides a general interface to generate mask files, a binary format RFI masks in PRESTO, thus we can use the mask file conveniently in the PRESTO pipeline. In the future, we plan to use CCF-ST in CRAFTS to help make more significant discoveries.

## Acknowledgments

This work is supported by National Natural Science Foundation of China (NSFC) under Nos. 11988101, U183110134, 11703047, 11773041, and U1831131. P.W. acknowledges support by the Youth Innovation Promotion Association CAS (id. 2021055) and cultivation project for FAST scientific payoff and research

achievement of CAMS-CAS. We thank the anonymous referee for useful comments on the manuscript.

This work made use of data from the Five-hundred-meter Aperture Spherical radio Telescope (FAST). FAST is a Chinese national mega-science facility, built and operated by the National Astronomical Observatories, Chinese Academy of Sciences.

## References

- Akeret, J., Chang, C., Lucchi, A., & Refregier, A. 2017, *A&C*, **18**, 35
- Arzaga, E. R., & Lynch, R. 2021, *BAAS*, **53**, 353.06
- Baan, W. A. 2019, *JAI*, **8**, 1940010
- Baan, W. A., Fridman, P. A., & Millenaar, R. P. 2004, *AJ*, **128**, 933
- Bhat, N. D. R., Cordes, J. M., Chatterjee, S., & Lazio, T. J. W. 2005, *RaSc*, **40**, 1
- Bracewell, R. N., & Bracewell, R. N. 1986, *The Fourier Transform and its Applications* (New York: McGraw-Hill)
- Briggs, F. H., Bell, J. F., & Kesteven, M. J. 2000, *AJ*, **120**, 3351
- Burd, P. R., Mannheim, K., Marz, T., et al. 2018, *AN*, **339**, 358
- Cameron, A. D., Li, D., Hobbs, G., et al. 2020, *MNRAS*, **495**, 3515
- Cruces, M., Champion, D. J., Li, D., et al. 2021, *MNRAS*, **508**, 300
- Fridman, P. A., & Baan, W. A. 2001, *A&A*, **378**, 327
- Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. 2014, arXiv:1404.7584
- Hotan, A. W., van Straten, W., & Manchester, R. N. 2004, *PASA*, **21**, 302
- Jiang, P., Tang, N.-Y., Hou, L.-G., et al. 2020, *RAA*, **20**, 64
- Kapinchev, K., Bradu, A., Barnes, F., & Podoleanu, A. 2015, in 2015 9th Int. Conf. on Signal Processing and Communication Systems (ICSPCS), 1
- Kerrigan, J., Plante, P. L., Kohn, S., et al. 2019, *MNRAS*, **488**, 2605
- Kiani Galoogahi, H., Sim, T., & Lucey, S. 2014, arXiv:1403.7876
- Kocz, J., Briggs, F. H., & Reynolds, J. 2010, *AJ*, **140**, 2086
- Kumar, V. 2002, *Introduction to Parallel Computing*, vol. 92 (2nd ed.; New York: Addison-Wesley), 147
- Leahy, D. A., Darbro, W., Elsner, R. F., et al. 1983, *ApJ*, **266**, 160
- Li, D., Wang, P., Qian, L., et al. 2018, *IMMag*, **19**, 112
- Miao, C. C., Zhu, W. W., Li, D., et al. 2023, *MNRAS*, **518**, 1672
- Nan, R., Li, D., Jin, C., et al. 2011, *IJMPD*, **20**, 989
- Offringa, A. R., de Bruyn, A. G., Biehl, M., et al. 2010, *MNRAS*, **405**, 155
- Offringa, A. R., de Bruyn, A. G., Zaroubi, S., & Biehl, M. 2010, arXiv:1007.2089
- Ransom, S. M. 2001, *New Search Techniques for Binary Pulsars* (Harvard Univ.)
- Ransom, S. M., Eikenberry, S. S., & Middleditch, J. 2002, *AJ*, **124**, 1788
- Smith, S. W. 1997, *The Scientist and Engineer's Guide to Digital Signal Processing*
- Tedila, H. M., Yuen, R., Wang, N., et al. 2022, *ApJ*, **929**, 171

Wang, C., Zhang, L., Xie, L., & Yuan, J. 2017, arXiv:1709.05936  
Wang, P., Li, D., Clark, C. J., et al. 2021a, *SCPMA*, **64**, 129562  
Wang, S., Zhu, W.-W., Li, D., et al. 2021b, *RAA*, **21**, 251  
Wang, Y., Zhang, Z., Zhang, H., et al. 2022, *A&C*, **39**, 100568

Wen, Z. G., Yuan, J. P., Wang, N., et al. 2022, *ApJ*, **929**, 71  
You, S.-P., Wang, P., Yu, X.-H., et al. 2021, *RAA*, **21**, 314  
Zeng, Q., Chen, X., Li, X., et al. 2020, *MNRAS*, **500**, 2969  
Zhang, L., Li, D., Hobbs, G., et al. 2019, *ApJ*, **877**, 55