Research in Astronomy and Astrophysics, 23:104004 (8pp), 2023 October
https://doi.org/10.1088/1674-4527/acd52b

Cleaning Radio Frequency Interference in Pulsar-Folded Data Based on the Conditional Random Fields with an Adaptive Prior

Xue Chen[1,2], J. L. Han[1,2,3], W. Q. Su[1,2], Z. L. Yang[1,2], and D. J. Zhou[1,2]
[1] National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China; hjl@nao.cas.cn
[2] School of Astronomy and Space Science, University of Chinese Academy of Sciences, Beijing 100049, China
[3] CAS Key Laboratory of FAST, NAOC, Chinese Academy of Sciences, Beijing 100101, China
Received 2022 December 29; revised 2023 April 10; accepted 2023 April 26; published 2023 September 13

Abstract

Radio astronomy observations are frequently impacted by radio frequency interference (RFI). We propose a novel method, named 2σCRF, for cleaning RFI in the folded data of pulsar observations, utilizing a Bayesian-based model called conditional random fields (CRFs). This algorithm minimizes the "energy" of every pixel given an initial label. The standard deviations (i.e., rms values) of the folded pulsar data are utilized as pixels for all subintegrations and channels. Non-RFI data without obvious interference is treated as "background noise," while RFI-affected data have different classes due to their exceptional rms values. This initial labeling can be automated and is adaptive to the actual data. The CRF algorithm optimizes the label category for each pixel of the image with the prior initial labels. We demonstrate the efficacy of the proposed method on pulsar folded data obtained from Five-hundred-meter Aperture Spherical radio Telescope observations. It can effectively recognize and tag various categories of RFIs, including broadband or narrowband, constant or instantaneous, and even weak RFIs that are unrecognizable in some pixels but picked out based on their neighborhoods. The results are comparable to those obtained via manual labeling but without the need for human intervention, saving time and effort.

Key words: methods: data analysis – techniques: image processing – (stars:) pulsars: general

1. Introduction

Sensitive radio telescopes receive not only radio signals from the universe but also terrestrial radio frequency interference (RFI), in addition to the thermal noise of their receiving systems. RFI significantly affects radio astronomy studies, including pulsar searching (Maan et al. 2021; Yuan et al. 2022) and radio imaging (e.g., Leshem et al. 2000; Fridman & Baan 2001; Offringa et al. 2010).

Cleaning RFI has always been a crucial step in data processing for radio astronomy studies (e.g., Leshem et al. 2000; Kocz et al. 2010; Maan et al. 2021; Yuan et al. 2022). Various methods have been proposed for RFI identification; however, the threshold method is the most commonly used approach. For instance, the SumThreshold (Offringa et al. 2010) is an RFI detection algorithm that fits the two-dimensional surface of frequency-time data and marks data points exceeding an iterative threshold along the X and Y axes for frequency and time, respectively. Later, the scale-invariant rank operator was developed and applied to data preceding the original SumThreshold detection method (Offringa et al. 2012), enabling automatic cleaning of weak RFI. Zeng et al. (2021) employed the asymmetrically reweighted penalized least squares smoothing (ArPLS) (originally proposed by Baek et al. 2015) to fit and remove the baseline of time-integrated spectra energy over all frequency channels and then identified RFI channels from the standard deviation curve using the SumThreshold method. This approach enabled one-dimensional RFI identification along both the time axis and the frequency axis on the frequency-time data through SumThreshold. Compared to SumThreshold alone, this strategy takes less execution time, and it can identify RFI efficiently and accurately. In Morello et al. (2019), it was assumed that RFI often occurs in a narrow frequency band or in a short time interval and rarely displays any dispersion. The authors utilized the standard deviation, peak-to-peak difference, and absolute Fourier transform value of the second bin to identify RFI, and their method is called "CLFD."

With the development of information technology, machine-learning techniques have been increasingly applied to RFI recognition in radio astronomy, such as principal component analysis (PCA) (see, e.g., Zhao et al. 2013) and independent component analysis (see, e.g., Dai et al. 2019). Recently, Yuan et al. (2022) added a classifier to the PCA method for RFI recognition in radio data. Additionally, there are techniques that use convolutional neural networks (e.g., Akeret et al. 2017a, 2017b; Burd et al. 2018; Czech et al. 2018; Yang et al. 2020); however, they require a significant amount of data to train them properly. The quality of the results from these techniques is highly dependent on the quality of the training data.

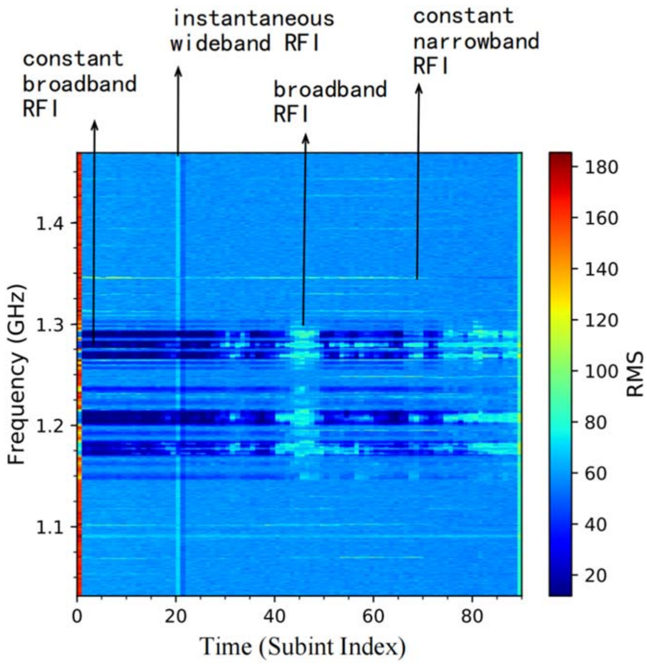


Figure 1. An example of folded pulsar data observed by FAST, specifically from FAST tracking data for PSR J1859+0430 with 90 subints of 10 s each over 1792 frequency channels covering the band of 1.03125–1.46875 GHz (after 128 channels on each side of the band been chipped off due to a low gain). The image values represent the root mean square values of folded pulsar profiles for all pixels in the two-dimensional array of subints and frequency channels. There are several types of RFI present in this data: (1) constant broadband RFI in the middle band between 1.16 and 1.30 GHz caused by satellites; (2) instantaneous wide-band RFI over the whole observation band due to some reasons, such as the first and last subints and subint No.20; (3) broadband RFI appearing for some minutes; (4) constant narrowband RFIs in some bands; (5) occasionally some instantaneous RFI in some channels (not marked in the figure).

Generally, the larger the receiving area of the radio telescope, the higher the observation sensitivity. Commissioned in September 2019, the Five-Hundred-Meter Aperture Spherical Telescope (FAST, Nan et al. 2011) is now the largest single-aperture telescope over the world with extremely high sensitivity, which has been used to search pulsars (e.g., Han et al. 2021) and observe the spectral lines (e.g., Hong et al. 2022; Hou et al. 2022), even scan for radio images (e.g., Gao et al. 2022). FAST has suffered the RFI problem during radio astronomy observations (Zhang et al. 2022). A more efficient and accurate RFI identification method is urgently needed for all kinds of FAST data processing.

During FAST pulsar observations, the raw data are recorded in the searching mode for a given observation time, e.g., 15 minutes, with a given number of frequency channels, e.g., 4096 or 2048 channels covering the observation band of 1.0–1.5 GHz. The data are folded according to the previously given period of a pulsar for a given length of integration time (e.g., 30 s, which is called a "subintegration" or "subint" for short) and also dedispersed for all frequency channels according to a previously given dispersion measure (DM) of the pulsar using the package DSPSR (van Straten & Bailes 2011). Sometimes the frequency channels can be reduced by combining a few channels into one. The pulsar-folded data are stored as a special FITS file (i.e., PSRFITS, Hotan et al. 2004) for all subints and frequency channels, in the form of a three-dimensional data cube. The three coordinate axes represent frequency (channel number), observation time (number of subints), and pulse profile values given for a number of bins for the period of a pulsar. All types of RFI may emerge in the data. For example, in Figure 1, broadband RFI often appears at the first few subints and subint No.20; many channels suffer from narrow-band RFI; some channels in the middle band between 1.16 and 1.3 GHz are affected by strong RFI from satellites; and occasionally some instantaneous RFI appears in some channels.

It would be beneficial to automatically classify RFI-affected folded pulsar data into different classes and label them automatically. Therefore, we have developed a new method that innovatively uses the algorithm of conditional random fields (CRFs, Lafferty et al. 2001) to optimize the label of each pixel according to the classification of connected pixels. The structure of this paper is as follows: In Section 2, we briefly introduce the RFI features of FAST observation data. The algorithm flow, method details, and data-processing methods are introduced in detail in Section 3. The results and discussion are presented in Section 3.3. Conclusions are given in Section 4.

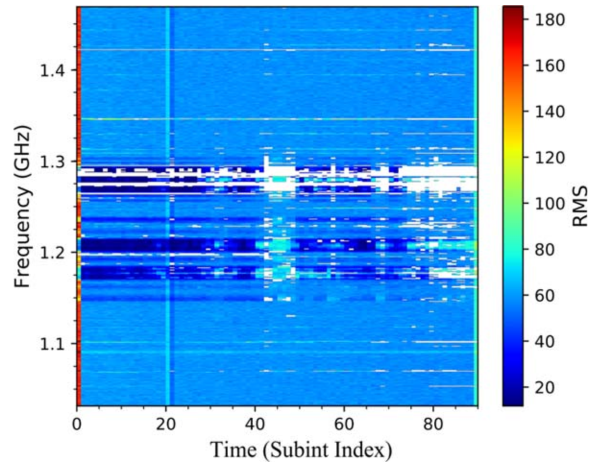
2. Experimental Data and RFI Features

FAST is an extremely sensitive radio telescope with 19-beam *L*-band receivers that cover a frequency band of 1.0–1.5 GHz. The signals within this 500 MHz band are recorded for 2048 channels, with a sampling time of 49.152 μs. The data used in this paper are all taken from the FAST Galactic Plane Pulsar Snapshot (GPPS) survey (Han et al. 2021). For each frequency channel, the original sampled data are folded according to a given pulsar period and a mean profile is obtained for each channel every 10 s for a subintegration, resulting in a total of 90 subints in this 15 minutes observation session. On each side of the FAST observation band, 128 of the 2048 frequency channels are removed due to low receiver gain, leaving only data for 1792 channels. Therefore, for each cross-pixel of 1792 channels $\times$ 90 subints, there are a series of data for a folded-pulse profile, although the signal-to-noise ratio of this pulse profile integrated for 10 s depends on the pulse strength in such a narrow frequency channel.

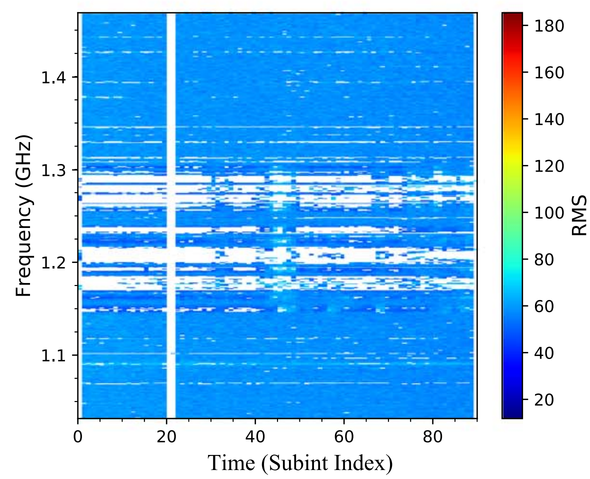
If a channel is affected by RFI in a given subint, the mean or standard deviation (rms) of the folded profile in this subint would be unusually different from those of unaffected pixels.

An image of the mean or rms could show bright spots or deep holes for these RFI-affected pixels. As shown in Figure 1 for a 15-minute observation session of PSR J1859+0430, there are various RFI features, some shown as wide-band interference, some only affecting a narrow frequency band, some appearing for a short time but with high intensity in a few channels, and some lasting for a long time and also present in many bands. Therefore, the RFI features could be recognized on a two-dimensional image of rms values (or mean values).

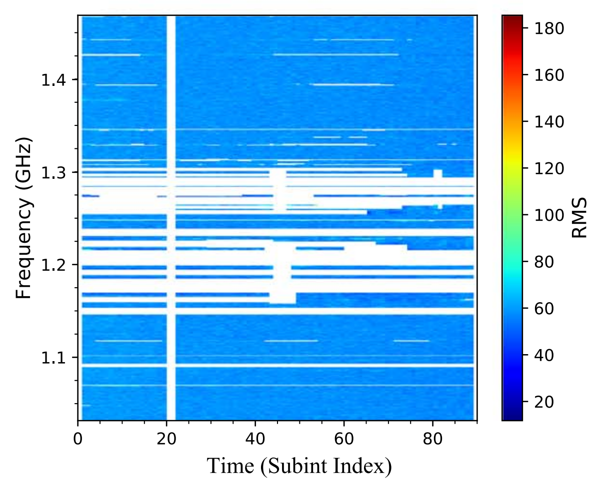
The PSRCHIVE package (Hotan et al. 2004; van Straten et al. 2012) is widely used for analyzing folded pulsar data. The first step is typically to use the “paz -r” command to automatically recognize and clean RFI in the data. This process involves applying a median filter algorithm to identify frequency channels affected by RFI in the two-dimensional plane of frequency channels and pulse phases. However, this method may not be effective for weak RFI or broadband RFI that persists for an entire pulsar period or several periods. If better RFI cleaning is desired, the data must be manually processed using the interactive “psrzap” command, which displays a two-dimensional image with the subint index on the horizontal axis and frequency channels on the vertical axis (see Figure 1). For pixels affected by RFI that cannot be automatically removed, particularly short-duration broadband interference or satellite interference, various interactive commands and parameters can be applied within the “psrzap” module to remove residual RFI shown in the image (see Figure 2). However, note that manually marking RFI pixels is a very time-consuming process.



(a) automatically via “paz -r”



(b) automatically via “CLFD”



(c) Manually with “psrzap”

Figure 2. The cleaned data obtained by “paz -r” in panel (a), the result obtained by the “CLFD” in panel (b), and by the manually interactive tool “parzap” in panel (c). Some RFIs cannot be removed through “paz -r” or “CLFD” automatically, and have to be cleaned manually.

3. Procedures for Auto RFI Cleaning

We have developed an auto-clean RFI package for folded pulsar data using the CRF algorithm in two steps: (1) simple classification for initial labels; (2) CRF optimization of all pixel labels. For the two-dimensional rms image of folded pulsar data with subint (x-axis) and frequency (y-axis) axes, we first obtain the distribution of rms values for all pixels (see Figure 3). The initial RFI labels are assigned to pixels with exceptional values in the distribution. The CRF-based algorithm (Lafferty et al. 2001) is then iteratively applied to optimize all pixel labels for marking RFI. The procedures are outlined in Figure 4.

3.1. Initial Labels

The CRF algorithm (Lafferty et al. 2001) requires a prior as input to the “class network” for optimal data segmenting and labeling. It is easy to identify obvious RFI-affected pixels from normal pixels based on quantitative statistical characteristics that are desired in the CRF algorithm. The rms of the folded profile for an RFI-affected pixel would be significantly different from that of normal pixels; therefore, the rms values

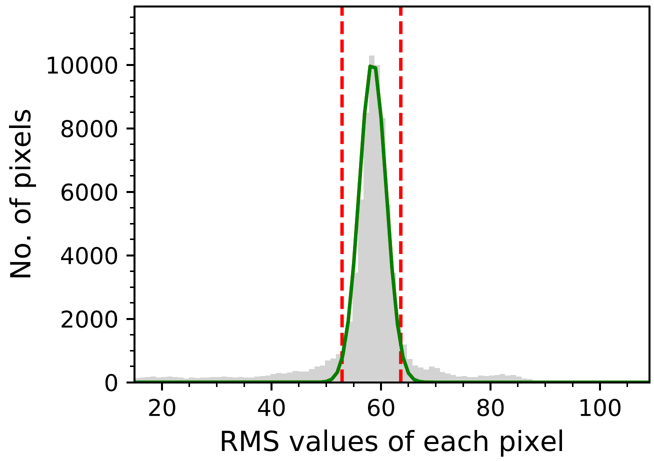


Figure 3. The distribution of rms values of the folded pulse profiles for all pixels in the two-dimensional image for subints and channels. The green curve represents the fitted Gaussian function to the distribution, and dotted lines indicate the Gaussian width of $\pm 2\sigma$ from the distribution peak, which is used as the threshold for initial RFI labels. It is clear that pixels with extreme values are influenced by RFI, but some data outside $\pm 2\sigma$ may not be RFI-affected and some data within $\pm 2\sigma$ may be affected.

can be used as input for the CRFs and are adaptive to observation data. However, the non-smooth bandpass may cause rms variation. If this occurs, the bandpass can be fitted using asymmetrically reweighted penalized least squares smoothing (ArPLS) (see Baek et al. 2015).

To correct for the gain curve of the band and "baseline" of rms values, we normalize the image of rms values of folded pulse profiles (see Figure 3). The rms distribution of "normal data" follows a Gaussian distribution, which is classified as "background noise." We fit the distribution with a Gaussian function and categorize pixels within $\pm 2\sigma$ of the distribution peak as "normal" pixels in the initial labels, where σ is the width of the fitted Gaussian. Pixels with very high rms values ($>3\sigma$ from the peak) are assigned to the RFI-affected category. However, pixels outside $\pm 2\sigma$ of the peak may not be RFI-affected, while some pixels within $\pm 2\sigma$ may still be RFI-affected. Figure 5(a) shows the "segmentation"-cleaned result according to the histogram threshold of $\pm 2\sigma$ from the peak.

A simple threshold for "segmentation" cannot differentiate between RFI-affected pixels and normal ones. Therefore, a subsequent step is necessary to optimize the labeling result. If a larger threshold (e.g., $\pm 3\sigma$) is used, some RFI-affected pixels will be missed in the initial labels, but if a smaller threshold is used, more normal pixels will be misjudged as RFI-affected.

3.2. RFI-labeling Based on the CRFs

We use the two-dimensional rms data as an image and optimize RFI labeling using the CRF algorithm (Lafferty et al. 2001). This algorithm is a specific type of graphical model that

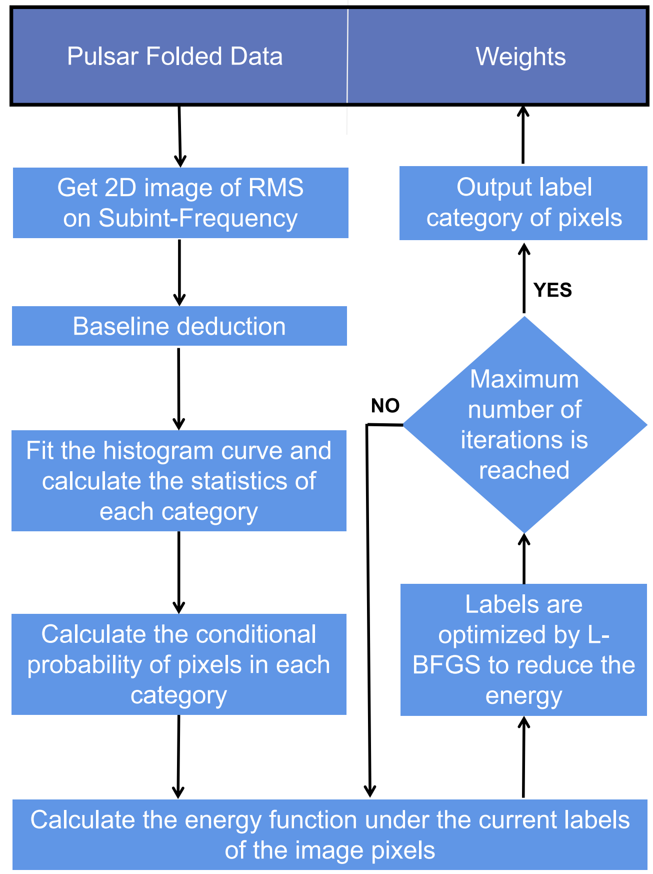
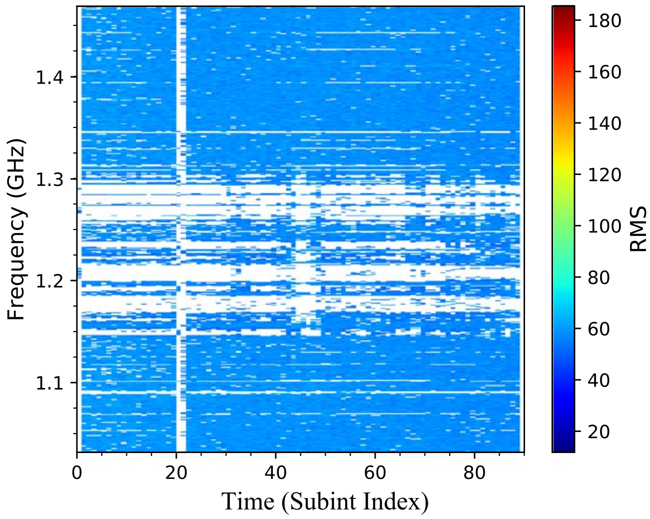


Figure 4. Flowchart for automatic RFI cleaning. The input data set is a two-dimensional rms image of folded pulsar data with subint (x-axis) and frequency (y-axis) axes. The output is the data with proper labels identifying RFI-affected pixels in the weight.

minimizes the defined energy functions, making image segmentation and pixel classification more accurate by predicting results from the model obtained from prior distribution estimation.

According to Lafferty et al. (2001), the CRFs are defined on the random variables of pixel values $x = \{x_1, x_2, \ldots, x_n\}$ and their corresponding class variables $\omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$. A crucial term in CRFs is the "clique c," which is defined as a subset of the pixel set. A clique c can represent either a unary clique composed of a single pixel or a multivariate clique composed of some connected pixels. The CRF algorithm calculates a conditional probability distribution $P(\omega|x)$. However, it is challenging to directly solve the probability of CRFs. According to the Hammersley–Clifford theory, the random fields and the Gibbs distribution are equivalent. It is more convenient to calculate the Gibbs distribution for the CRFs using the following equation:

$$P(\omega|x) = \frac{1}{Z}\exp(-U(\omega|x)) = \frac{1}{Z}\exp\left(-\sum_{c\in C} V_c(\omega|x)\right), \quad (1)$$



(a) Segmentation result of $\pm 2\sigma$ in the distribution

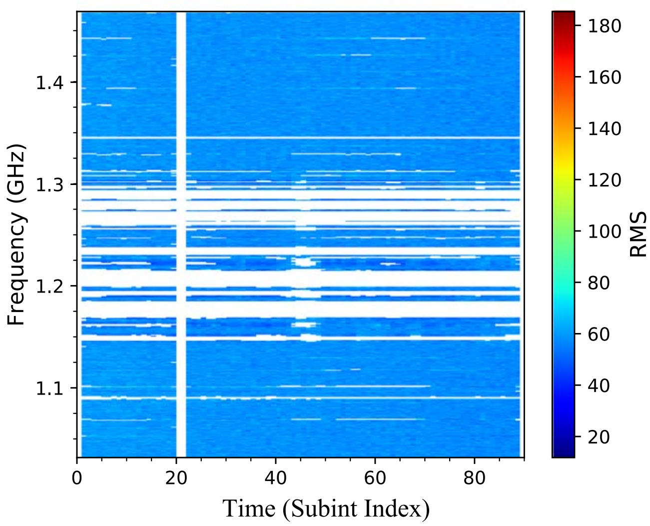


Figure 5. Comparison of RFI labeling results using the $\pm 2\sigma$ threshold based on the distribution of rms values and the final optimized pixel category labels using the CRFs.

where Z is a normalized constant, $V_c(\omega|x)$ is the potential defined on the clique c, and c is one type clique in the clique set C. Meanwhile, $U(\omega|x)$ is the energy function of CRFs, which is a summation of the potentials of all types of cliques in the clique set C. For ease of notation, we will omit the conditioning x throughout the rest of this paper and use $V_c(\omega)$ to denote $V_c(\omega|x)$ and $U(\omega)$ to denote $U(\omega|x)$.

To obtain fine-grained results for the pulsar data image, we adopt fully connected CRFs (Krähenbühl & Koltun 2011) and its implementation, denseCRF,[4] to optimize the pixel labels based on the data characteristics. In this case, the clique set C is defined as a complete graph G. However, in practice, only the unary and pairwise terms in the clique set C are considered for computing the potentials. The energy function can be expressed as:

$$U(\omega) = \sum_{c \in C} V_c(\omega) = \sum_{i} V_1(\omega_i) + \sum_{(i,\, i')} V_2(\omega_i,\, \omega_{i'}). \tag{2}$$

The unary term in our model represents the potential energy for each pixel when it is labeled ω_i, given a specific conditional observation. To calculate the unary potential energy for pixel i with label ω_i, we first obtain the mean and standard deviation of the image for each category of pixels, using the initial labels obtained from Section 3.1 as a guide. We then apply the Bayesian conditional probability formula to calculate the probability that each pixel belongs to each category, and take the negative logarithm of this probability as the unary potential energy value (i.e., the real value less than 1.0) in Equation (2).

The second term in Equation (2) is the pairwise potential function. This function takes into account the correlations between connected pixel points on the graph GG, which we define as follows:

$$V_2(\omega_i,\, \omega_{i'}) = \mu(\omega_i,\, \omega_{i'})(\lambda_a k_a(\boldsymbol{f}_i, \boldsymbol{f}_{i'}) + \lambda_s k_s(\boldsymbol{f}_i, \boldsymbol{f}_{i'})), \tag{3}$$

where $\mu(\omega_i,\, \omega_{i'}) = 1_{[\omega_i \neq \omega_{i'}]}$ to introduce a penalty between pixels that belongs to the different labels. λ_a and λ_s are weights for the appearance kernel k_a and smoothness kernel k_s. These two kernels measure the rms similarity and the positional similarity of pairwise pixels. The kernels are defined as

$$k_a(\boldsymbol{f}_i, \boldsymbol{f}_{i'}) = \exp\left(-\frac{|p_i - p_j|^2}{2\xi_1^2} - \frac{|I_i - I_j|^2}{2\xi_2^2}\right), \tag{4}$$

and

$$k_s(\boldsymbol{f}_i, \boldsymbol{f}_{i'}) = \exp\left(-\frac{|p_i - p_j|^2}{2\xi_3^2}\right). \tag{5}$$

This appearance kernel k_a is used to penalize the case where pixels of different classes have a similar appearance. On the other hand, the smooth kernel k_s is designed to penalize the case where the pixels of different classes have a close location.

Once the energy function is defined, the original problem can be transformed into an optimization problem. As shown in Equation (1), maximizing the conditional probability is equivalent to minimizing the energy function U. The energy function can be optimized using the limited-memory BFGS algorithm (i.e., the L-BFGS method; see Liu & Nocedal 1989). In our implementation, we terminate the optimization process after five iterations, which generally leads to an ideal segmentation output. The final output is a set of labels that indicate which pixels are affected by RFI.

Finally, after the classification labels for all pixels are obtained, a post-cleaning procedure is performed to identify frequency channels that have only a small fraction (i.e., less

[4] https://github.com/HiLab-git/SimpleCRF

than 10%) of remaining pixels. These channels are marked out, as the vast majority of pixels in such channels are likely affected by RFI. Even pixels with a normal rms value may not be completely safe from RFI in such cases. In previous manual pulsar-folded data processing, these pixels were often identified and removed.

As shown in Figure 5(b), the final segmentation results obtained through pixel-optimization labeling by the CRFs exhibit more accurate RFI recognition. Some of the non-RFI pixels that were incorrectly labeled as RFI are restored, and real RFI-affected pixels with low rms values are now accurately labeled. Compared to the results obtained from automatic PSRCHIVE cleaning, as shown in Figure 1(b), our method significantly improves RFI recognition and cleaning accuracy.

3.3. Discussion and More RFI-cleaning Experiments

To avoid the impact of gain variation over time in a radio telescope, we perform RFI recognition on the two-dimensional rms values rather than the mean values of folded pulsar data. We fit the rms distribution using a Gaussian function around the peak, which allows for the adaptive selection of an appropriate initial segmentation threshold. We name this new algorithm "2σCRF." During the CRF label optimization phase, we consider the connected pixel values and labels rather than just independent pixel values. This approach enables us to effectively identify all types of RFI, including broadband interference and weak RFI, by taking into account the relationship between pixels.

We believe that this new algorithm is applicable to all folded pulsar data obtained with any radio telescope. For ultra-wide-band observations that exhibit very different rms values in the lower and upper parts of the observation band, the algorithm can be applied separately to different parts of the band, allowing for accurate labeling of all RFI-affected pixels.

In the 2σCRF algorithm, some parameters must be set properly. During the initial labeling phase, the threshold can be set to $\pm 3\sigma$ or even $\pm 1\sigma$, but we have found that a threshold of $\pm 2\sigma$ from the rms distribution peak provides the best results. In the CRF label optimization phase, the maximum number of iterations (MaxIterations) must be defined. We use five iterations as the default, but have found that using 10 iterations does not significantly improve performance and is more time-consuming.

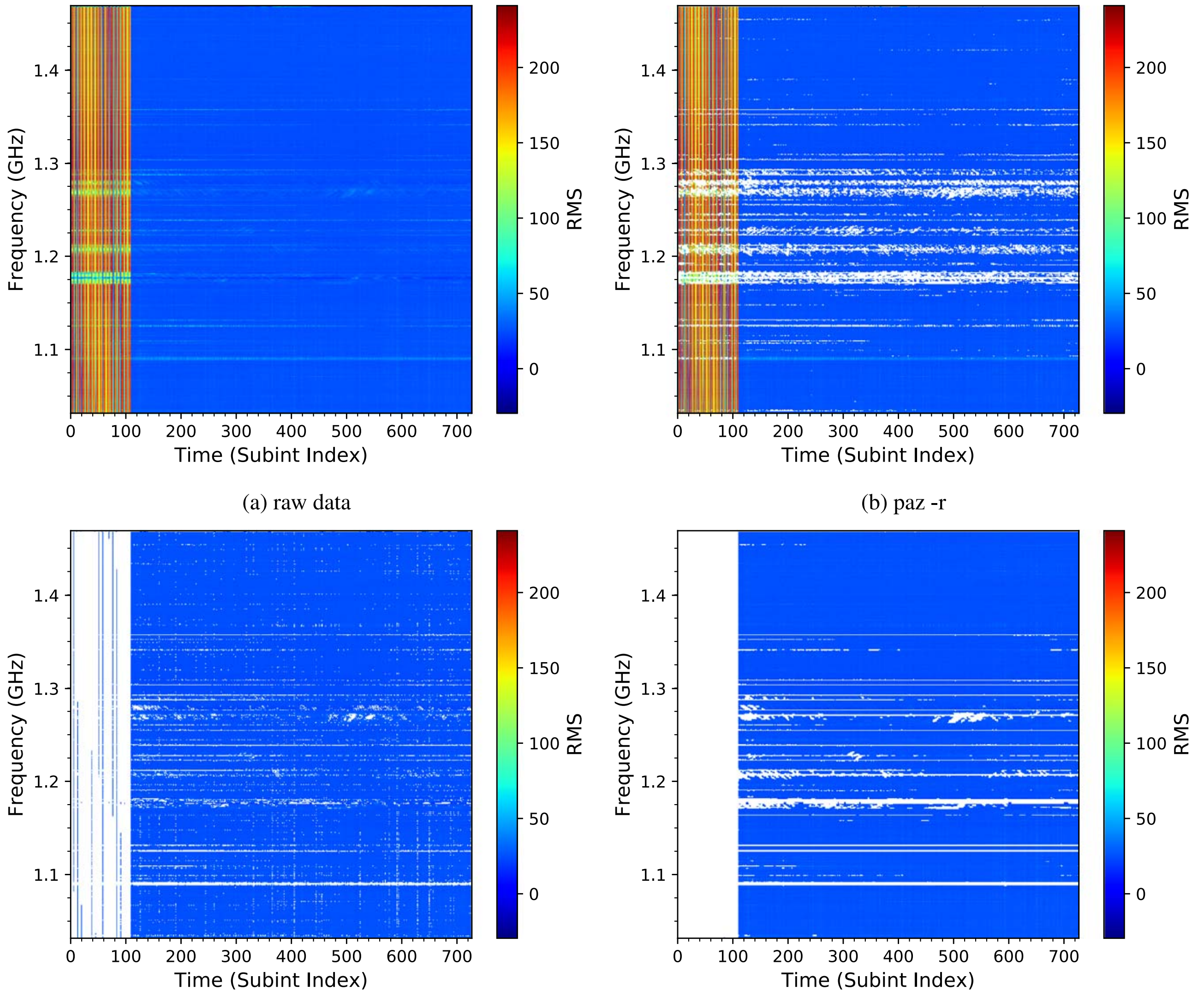
The appearance kernel weight λ_a, smoothness kernel weight λ_s, and standard deviations ξ_1, ξ_2, ξ_3 must also be set properly. These parameters control the similarity between related pixels. A higher appearance kernel weight λ_a penalizes pixels with similar appearances in different classes, resulting in smoother segmentation results. Similarly, a higher smoothness kernel weight λ_s penalizes pixels with similar locations in different classes. A larger standard deviation increases the neighborhood of the pixel affected by nearby pixel labels. In our implementation, we set $\lambda_a = 3$, $\lambda_s = 20$ for different kernels, with $\xi_1 = 1$, $\xi_2 = 10$, $\xi_3 = 1$ as the standard deviations. These values were found to work best for FAST pulsar observation data. However, new parameters may need to be tested and selected for data obtained from other telescopes. To more accurately identify RFI-affected data for different observations, we offer command line options for input parameters that can be fine-tuned.

Table 1
The Execution Time of Different Automatic Methods for RFI-cleaning. Our Own Method 2σCRF is marked in bold

Method	Steps	Execution Time (s)
"paz -r" (in PSRCHIVE)	all	4.15 ± 0.66
CLFD	all	4.88 ± 0.47
2σCRF	**all**	$\mathbf{6.55 \pm 0.62}$
2σCRF-1	Initial labels	4.05 ± 0.10
2σCRF-2	CRF optimization	0.37 ± 0.00
2σCRF-3	read/write file	1.98 ± 0.63

In general, the effectiveness of an RFI-cleaning algorithm should be evaluated based on its execution speed and recognition accuracy. We evaluated the execution time of the 2σCRF package, including the time required for RFI-pixel marking and reading/writing a new FITS data file. The execution time was obtained as the average value from 10 runs using the same data. As shown in Table 1, the CRF label optimization process is very fast and takes only a few seconds, which is not significantly longer than the execution times of "paz -r" and "CLFD." Additionally, the 2σCRF package does not require any further manual interactive processing, which can take a much longer time. Overall, the execution speed and accuracy of the 2σCRF algorithm make it a highly effective tool for RFI cleaning.

To evaluate the accuracy of RFI recognition, we used the results of cleaned pulsar data from the manually interactive "psrzap" as the ground truth template. We tested the algorithms using five data sets of FAST pulsar observations (see an additional example in Figure 6), and the results of quantitative evaluations are listed in Table 2. The table contains the accuracy of proper classification and the rate of mis-identifications. The overall accuracy of an algorithm is defined as the percentage of properly classified pixels (both "noise" and "RFI-affected") over the number of all pixels. The fraction of normal pixels misclassified as RFI pixels is expressed as $f_{(N->R)}$, and the fraction of RFI pixels misclassified as normal pixels is expressed as $f_{(R->N)}$. Our results show that the 2σCRF algorithm consistently provides more accurate labeling than both "paz -r" in PSRCHIVE and "CLFD." The last column of the table is the signal-to-noise ratio for the final averaged pulse profile, all improved due to RFI removal. For pulsars with



(a) raw data (b) paz -r

(c) CLFD (d) 2σCRF

Figure 6. RFI recognition results for a 15-minute observation of PSR J1913+3732 obtained using the FAST are shown. The data, which is "cal-affected," consist of all types of RFIs in the full frequency channel in subint No. 0–110. The original data is presented in panel (a), the result obtained using "paz -r" is shown in panel (b), the result obtained using "CLFD" is shown in panel (c), and the result obtained using the 2σCRF algorithm is shown in panel (d). The 2σCRF algorithm proves to be the most effective at RFI recognition.

strong RFIs, such as the data set for PSR J1913+3732, the SNR could be significantly improved. However, for the data of a weak pulsar with a low signal-to-noise ratio, the improvement is limited, e.g., PSR J1859+0430, depending on the properties of RFIs.

4. Conclusion

We present a novel approach for mitigating RFI in pulsar observations. Our algorithm utilizes the CRF method for the first time, making it adaptive for pulsar folding data. We use a threshold of $\pm 2\sigma$ from the peak of the rms distribution for initial labeling, and then apply the CRF model to optimize labeling. By utilizing this Bayesian-based technique, our new algorithm can effectively eliminate all types of RFI in the data automatically, including both narrow-band and wide-band RFI, constant or instantaneous. Experimental results presented in Section 3.3 demonstrate that the new algorithm provides results that are very close to the best results obtained manually, while saving significant time and manpower.

While our new RFI-cleaning algorithm can achieve ideal RFI-labeling results for current FAST pulsar observation data, there is still room for improvement in two aspects. First, it may

Table 2
The Comparison of Accuracy and Mis-classification Rate of RFI-labeling. Our Own Method 2σCRF is marked in bold

Data Set for	Method	Accuracy (%)	$f_{(N->R)}$ (%)	$f_{(R->N)}$ (%)	S/R
J1854+0319	raw data				245.2
	"paz -r"	90.93	6.21	29.92	257.0
	CLFD	90.38	4.66	45.75	247.1
	2σCRF	**93.45**	**5.18**	**16.53**	**255.2**
	manual				253.2
J1859+0430	raw data				28.2
(Figures 1, 2, 5)	"paz -r"	73.61	2.94	83.04	22.3
	CLFD	87.69	1.68	38.01	29.6
	2σCRF	**93.42**	**1.69**	**18.40**	**32.6**
	manual				32.4
J1913+3732	raw data				1273.1
(Figure 6)	"paz -r"	79.36	4.21	74.21	1164.7
	CLFD	91.52	2.24	28.84	3330.5
	2σCRF	**95.32**	**1.71**	**14.36**	**4348.9**
	manual				2455.2

be necessary to investigate if the segmentation results of RFI-affected pixels still depend on the selection of thresholds and parameters. Second, there is an issue with variable rms in very ultra-wide band broadband observations. To address this issue, the algorithm should be carried out independently and automatically for different frequency ranges.

Acknowledgments

This work makes use of the data from the GPPS survey project, as one of five key projects of FAST, a Chinese national mega-science facility, operated by the National Astronomical Observatories, Chinese Academy of Sciences. The authors are supported by the National Natural Science Foundation of China (NSFC, Nos. 11988101 and 11833009) and the Key Research Program of the Chinese Academy of Sciences (grant No. QYZDJ-SSW-SLH021).

Data and Package Availability

A Python software package of this work and also the sample data are available at http://zmtt.bao.ac.cn/GPPS/RFI.

ORCID iDs

J. L. Han https://orcid.org/0000-0002-9274-3092
D. J. Zhou https://orcid.org/0000-0002-6423-6106

References

Akeret, J., Chang, C., Lucchi, A., & Refregier, A. 2017a, A&C, 18, 35
Akeret, J., Seehars, S., Chang, C., et al. 2017b, A&C, 18, 8
Baek, S.-J., Park, A., Ahn, Y.-J., & Choo, J. 2015, Ana, 140, 250
Burd, P. R., Mannheim, K., März, T., et al. 2018, AN, 339, 358
Czech, D., Mishra, A., & Inggs, M. 2018, A&C, 25, 52
Dai, W., Shang, Z., Xu, Y., et al. 2019, AR&T, 16, 268
Fridman, P. A., & Baan, W. A. 2001, A&A, 378, 327
Gao, X., Reich, W., Sun, X., et al. 2022, SCPMA, 65, 129705
Han, J. L., Wang, C., Wang, P. F., et al. 2021, RAA, 21, 107
Hong, T., Han, J., Hou, L., et al. 2022, SCPMA, 65, 129702
Hotan, A. W., van Straten, W., & Manchester, R. N. 2004, PASA, 21, 302
Hou, L., Han, J., Hong, T., Gao, X., & Wang, C. 2022, SCPMA, 65, 129703
Kocz, J., Briggs, F. H., & Reynolds, J. 2010, AJ, 140, 2086
Krähenbühl, P., & Koltun, V. 2011, in Advances in Neural Information Processing Systems, NeurIPS '11, ed. J. Shawe-Taylor et al., 1
Lafferty, J. D., McCallum, A., & Pereira, F. C. N. 2001, in Proc. Eighteenth Int. Conf. on Machine Learning, ICML '01 (San Francisco, CA: Morgan Kaufmann), 282
Leshem, A., van der Veen, A.-J., & Boonstra, A.-J. 2000, ApJS, 131, 355
Liu, D. C., & Nocedal, J. 1989, MatPr, 45, 503
Maan, Y., van Leeuwen, J., & Vohl, D. 2021, A&A, 650, A80
Morello, V., Barr, E. D., Cooper, S., et al. 2019, MNRAS, 483, 3673
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., van de Gronde, J. J., & Roerdink, J. B. T. M. 2012, A&A, 539, A95
van Straten, W., & Bailes, M. 2011, PASA, 28, 1
van Straten, W., Demorest, P., & Oslowski, S. 2012, AR&T, 9, 237
Yang, Z., Yu, C., Xiao, J., & Zhang, B. 2020, MNRAS, 492, 1421
Yuan, M., Zhu, W., Zhang, H., et al. 2022, MNRAS, 513, 4787
Zeng, Q., Chen, X., Li, X., et al. 2021, MNRAS, 500, 2969
Zhang, C.-P., Xu, J.-L., Wang, J., et al. 2022, RAA, 22, 025015
Zhao, J., Zou, X., & Weng, F. 2013, ITGRS, 51, 4830