A GPU based single-pulse search pipeline (GSP) with database and its application to the Commensal Radio Astronomy FAST Survey (CRAFTS)

Shan-Ping You (游善平)\(^1,2\), Pei Wang (王培)\(^3\), Xu-Hong Yu (于徐红)\(^2\), Xiao-Yao Xie (谢晓尧)\(^1,2\), Di Li (李菂)\(^3,4\), Zhi-Jie Liu (刘志杰)\(^2\), Zhi-Chen Pan (潘之辰)\(^3\), You-Ling Yue (岳友岭)\(^3\), Lei Qian (钱磊)\(^3\), Bin Zhang (张彬)\(^2\) and Zong-Hao Chen (陈宗浩)\(^2\)

\(^1\) School of Computer Science and Technology, Guizhou University, Guiyang 550025, China; xyx@gznu.edu.cn
\(^2\) Key Laboratory of Information and Computing Science Guizhou Province, Guizhou Normal University, Guiyang 550001, China; wangpei@nao.cas.cn
\(^3\) CAS Key Laboratory of FAST, NAOC, Chinese Academy of Sciences, Beijing 100101, China; wangpei@nao.cas.cn
\(^4\) NAOC-UKZN Computational Astrophysics Centre, University of KwaZulu-Natal, Durban 4000, South Africa

Received 2021 June 25; accepted 2021 September 23

Abstract We developed a GPU based single-pulse search pipeline (GSP) with a candidate-archiving database. Largely based upon the infrastructure of the open source PulsaR Exploration and Search Toolkit (PRESTO), GSP implements GPU acceleration of the de-dispersion and integrates a candidate-archiving database. We applied GSP to the data streams from the Commensal Radio Astronomy FAST Survey (CRAFTS), which resulted in quasi-real-time processing. The integrated candidate database facilitates synergistic usage of multiple machine-learning tools and thus improves efficient identification of radio pulsars such as rotating radio transients (RRATs) and fast radio bursts (FRBs). We first tested GSP on pilot CRAFTS observations with the FAST Ultra-Wide Band (UWB) receiver. GSP detected all pulsars known from the the Parkes multibeam pulsar survey in the corresponding sky area covered by the FAST-UWB. GSP also discovered 13 new pulsars. We measured the computational efficiency of GSP to be \(\sim 120\) times faster than the original PRESTO and \(\sim 60\) times faster than an MPI-parallelized version of PRESTO.

Key words: methods: data analysis — pulsars: general — surveys: astronomical databases

1 INTRODUCTION

Pulsars are rapidly rotating, highly magnetized neutron stars. Since the first pulsar was discovered in 1967, more than 2800 pulsars (Manchester et al. 2005) have been detected. Pulsar detection mainly relies on Fast Fourier Transform (FFT) to search in the time-frequency domain and Fast Folding Algorithm (FFA) (Staelin 1969) to search in the time domain. These two methods are very efficient for periodic pulsar searches. However, some pulsars have sporadic pulse trains, such as nulling pulsars and intermittent pulsars, and rotating radio transients (RRATs), which are better dealt with through single pulse detection. A fast radio burst (FRB) (Lorimer et al. 2007) is a radio pulse with extremely high dispersion measure (DM) and a duration of milliseconds, presumably from outside the Milky Way. FRBs can only be searched through single pulse search at present.

With faster sampling and more simultaneous beams, the data volume of pulsar search surveys has reached petabytes. Commensal Radio Astronomy FAST Survey (CRAFTS) (Li et al. 2018), for example, generates 6 GB s\(^{-1}\) pulsar search data streams. It is paramount to accelerate the data processing and prioritization of resulting pulse candidates. Due to its superior parallel processing capability and memory bandwidth, graphics processing units (GPUs) have been widely utilized in pulsar search. The AstroAccelerate software package\(^1\) has GPU implemented Fourier domain acceleration search, GPU accelerated harmonic sum of periodicity search and also a GPU implementation of single pulse detection algorithms for real-time FRB searches (Adámek & Armour 2020). Jameson and Barsdell developed the GPU accelerated transient detection pipeline Heimdall\(^2\). Yu et al. (2020)}

\(^1\) https://gitlab.com/ska-telescope/astro-accelerate
\(^2\) https://sourceforge.net/p/heimdall-astro/
developed a CPU-based parallel pipeline based on Pulsar Exploration and Search Toolkit (PRESTO, https://www.cv.nrao.edu/~sransom/presto/) to shorten the data processing time.

Since millions of candidates or diagnostic plots are generated by pulsar survey data processing, inspecting all of them by eye is time-consuming as well as a laborious task in which it is easy to make mistakes. Machine learning, particularly deep learning, has been shown to be effective in ranking the candidates (Zhu et al. 2014; Wang et al. 2019). There are many tools to realize single pulse search identification. Karako-Argaman presented a single pulse sifting algorithm named RRATtrap, which first groups the detected single pulse events of nearby DMs and times, then ranks and scores the grouped candidates according to some specified rules (Karako-Argaman et al. 2015). Devine presented a machine learning approach to identify and classify dispersed pulse groups (DPGs), which contain two-stages, first identifying DPGs in signal to noise ratio (S/N) versus DM domain by a peak identification algorithm, then using supervised machine learning for automatic DPG classification (Devine et al. 2016). Michilli developed a machine learning classifier, called Single-Pulse Searcher (SPS), to discriminate astrophysical signals from a strong RFI environment (Michilli & Hessels 2018). Pang et al. (2018) developed a two-stage single pulse search method, called Single Pulse Event Group IDentification (SPEGID), which first identifies single pulse candidates as single pulse event groups (SPEGs) within DMs versus time span, then automatically classifies unlabelled SPEGs as astrophysical (pulsars, RRATs or FRB) or nonastrophysical (RFI or noise) cases through supervised machine learning.

Pulsar search is one of the primary science objectives for FAST. We designed a GPU-based single-pulse search pipeline with candidate-archiving database (GSP) to realize a quasi-real-time processing of the CRAFTS data. GSP implements the PulsaR Exploration and Search Toolkit (PRESTO) de-dispersion algorithm ‘prepsubband’ on NVIDIA GPUs using CUDA, which performs 120 times faster than the original program and 60 times faster than MPI on an NVIDIA GTX 1080 GPU. GSP incorporates a pulsar-search database, through which astronomers can retrieve data analysis results, and achieve centralized management of the data processing.

We describe GSP in Section 2. In Section 3, we benchmark the GSP in test runs. In Section 4 we briefly present the processing that resulted in 13 new pulsars from CRAFTS. We provide our conclusions in Section 5.

### Table 1 Frequency Bands of FAST UWB Drift Scan Data in the Single Pulse Search Pipeline

<table>
<thead>
<tr>
<th>Band</th>
<th>Frequency (MHz)</th>
<th>Bandwidth (MHz)</th>
<th>Channels across the band</th>
<th>Central frequency (MHz)</th>
<th>Drifting time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Band1</td>
<td>271 to 398</td>
<td>128</td>
<td>512</td>
<td>334</td>
<td>52</td>
</tr>
<tr>
<td>Band2</td>
<td>291 to 802</td>
<td>512</td>
<td>2048</td>
<td>546</td>
<td>52</td>
</tr>
</tbody>
</table>

### Table 2 Subband De-dispersion Survey Plan for Band1

<table>
<thead>
<tr>
<th>No.</th>
<th>Low DM (pc cm$^{-3}$)</th>
<th>High DM (pc cm$^{-3}$)</th>
<th>dDM (pc cm$^{-3}$)</th>
<th>DownSamp</th>
<th>Number of DMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>11.34</td>
<td>0.01</td>
<td>1</td>
<td>1134</td>
</tr>
<tr>
<td>2</td>
<td>11.34</td>
<td>22.2</td>
<td>0.03</td>
<td>2</td>
<td>362</td>
</tr>
<tr>
<td>3</td>
<td>22.2</td>
<td>39.05</td>
<td>0.05</td>
<td>4</td>
<td>337</td>
</tr>
<tr>
<td>4</td>
<td>39.05</td>
<td>77.15</td>
<td>0.1</td>
<td>8</td>
<td>381</td>
</tr>
<tr>
<td>5</td>
<td>77.15</td>
<td>153.75</td>
<td>0.2</td>
<td>16</td>
<td>383</td>
</tr>
<tr>
<td>6</td>
<td>153.75</td>
<td>343.75</td>
<td>0.5</td>
<td>32</td>
<td>380</td>
</tr>
<tr>
<td>7</td>
<td>343.75</td>
<td>687.75</td>
<td>1</td>
<td>64</td>
<td>344</td>
</tr>
<tr>
<td>8</td>
<td>687.75</td>
<td>989.75</td>
<td>2</td>
<td>128</td>
<td>141</td>
</tr>
</tbody>
</table>

### Table 3 Subband De-dispersion Survey Plan for Band2

<table>
<thead>
<tr>
<th>No.</th>
<th>Low DM (pc cm$^{-3}$)</th>
<th>High DM (pc cm$^{-3}$)</th>
<th>dDM (pc cm$^{-3}$)</th>
<th>DownSamp</th>
<th>Number of DMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>48.77</td>
<td>0.01</td>
<td>1</td>
<td>4877</td>
</tr>
<tr>
<td>2</td>
<td>48.77</td>
<td>93.14</td>
<td>0.03</td>
<td>2</td>
<td>1479</td>
</tr>
<tr>
<td>3</td>
<td>93.14</td>
<td>164.74</td>
<td>0.05</td>
<td>4</td>
<td>1432</td>
</tr>
<tr>
<td>4</td>
<td>164.74</td>
<td>325.04</td>
<td>0.1</td>
<td>8</td>
<td>1603</td>
</tr>
<tr>
<td>5</td>
<td>325.04</td>
<td>485.04</td>
<td>0.2</td>
<td>16</td>
<td>800</td>
</tr>
<tr>
<td>6</td>
<td>485.04</td>
<td>582.84</td>
<td>0.2</td>
<td>16</td>
<td>539</td>
</tr>
</tbody>
</table>

### 2 SINGLE PULSE SEARCH PIPELINE

#### 2.1 Overview of the Search Pipeline

We first tested GSP on pilot CRAFTS observations with the FAST-Ultra-Wide-Band (UWB) receiver (270 – 1620 MHz) from August 2017 to May 2018. A series of original fits files with a fixed size separately recorded two frequency bands of 0–1 GHz and 1–2 GHz in PSRFITS format (Hotan et al. 2004) with two polarization channels of XX and YY with 8 bit sampling, including ~1800 hours of observation time (about 500 TB) in 2017, and ~960 hours of observation time (about 260 TB) in 2018, stored in the cluster disk array and tape library of the FAST early scientific data center.

The GSP implements GPU acceleration of the de-dispersion and integrates a candidate-archiving database. We applied GSP to the data streams from the CRAFTS survey, and the data processing is performed in the FAST early scientific data center. For the single-pulse pulsar search, there are five major steps involved in processing the data: data preparing, pulsar search (RFI mitigation, de-dispersion and single-pulse search) and results evaluation. See Figure 1 for a schematic of data processing.
In data preparation, we first merged every two adjacent fits files for each actual pointing, and the data stream overlapped half of the time for the Nyquist sampling. For pulsar search, as illustrated in Algorithm 1, under the premise that the channels and samples are determined, the number of DMs determines the calculation load of the dedispersion. In GSP, we mainly use PRESTO’s ‘DDplan.py’ to generate the optimized DM trails. For two different frequency bands (Table 1), the parameters of de-dispersion trails are displayed in Tables 2 and 3.

For the single pulse search of CRAFTS, there are two challenges to be solved in data processing. First, the de-dispersion accounts for a large proportion of the data processing, and secondly, it needs to identify a large number of single pulse candidates. In order to improve the search efficiency, we implemented a parallel acceleration policy to the search process, in which the most time-consuming de-dispersion part is accelerated by a GPU (see in the next section), and the single-pulse search, single pulse identification and data folding steps were executed in parallel on CPUs, which saved a lot of data processing time. The entire processing flow starts with data processing. We manage with the help of a database to facilitate the later single pulse identification and diagnosis. For the single pulse candidates that failed to find periods, we used the SPS software package as an identification tool to generate single pulse diagnostic plots to determine whether it is an astronomical signal from the universe.

### 2.2 Single Pulse Candidate-archiving Database

In order to facilitate the storage, management and query of single-pulse candidates, we implemented a database based on the framework of MariaDB⁴. As displayed in Figure 2, the database consists of six tables, mainly recording the single pulse candidates identified by single pulse sifting algorithm. The database stores the psrfits file to be processed in the T_FITS_INFO table. For the single pulse results identified by RRATtrap, we save the overall information of each file in T_SINGLEPULSE_GROUPS, and store the information about each single pulse candidate, including the maximum DM, minimum DM, number of events of single pulse candidates, maximum sigma, pulse appearance time, pulse duration and other information, in T_SINGLEPULSE_CANDIDATES. We saved the pulsars in the latest ATNF Pulsar Catalogue in T_PULSAR, and compared them with DM according to the direction of the observation file, saved the results of the comparison in T_FILE_PULSAR and the path of pictures generated by the pipeline was saved in T_SINGLEPULSE_PIC.

### 2.3 GPU Accelerated De-dispersion

For performing the blind search, we do not know the DM of pulsars, so we must make an optimized dedispersion plan and search over a wide range of trial DMs. Computing the dedispersion transform is a computationally intensive task. There are three algorithms: brute force dedispersion, tree dedispersion algorithm (Taylor 1974), and sub-band dedispersion algorithm (Barsdell et al. 2012). We analyzed acceleration of the parallel dedispersion algorithm on GPU, and developed a C language GPU dedispersion library. We implemented the sub-band dedispersion algorithm based on PRESTO’s ‘prepsubband’. This article examines the acceleration and designs an accelerated program that supports multiple GPUs.

In PRESTO prepsubband, the input of dedispersion was stored in two preprocessed channelized time-series matrices with numpts × numchan, where numpts is the number of time samples per read and numchan is the frequency channels. The delay of each channel was pre-calculated and stored in a numdm × numchan matrix, where numdm is the number of DMs to disperse. The results after dedispersion were stored in a matrix with numdm × numpts. The algorithm is illustrated in Algorithm 1.

#### Algorithm 1 Pseudocode of the dedispersion algorithm of PRESTO prepsubband

**Require:** data array with numpts × numchan, delays array with numdm × numchan, numpts, numchan  
**Ensure:** result array with numdm  

```plaintext
for ii = 0 → numpts do  
    result[ii] = 0  
end for

for ii = 0 → numchan do  
    jj = ii + delays[ii] × numchan  
    for kk = 0 → numpts – delays[ii] do  
        result[kk] += lastdata[jj]  
    end for
end for
```

### Table 4 Basic Parameters of CPU and GPU

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>AMD Ryzen 7 2700X</td>
<td>NVIDIA GeForce GTX 1080</td>
</tr>
<tr>
<td><strong>Core number</strong></td>
<td>8 cores, 16 threads</td>
<td>2560 cores</td>
</tr>
<tr>
<td><strong>Clock rate</strong></td>
<td>3.7 GHz</td>
<td>1.6/1.7 GHz</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>64 GB</td>
<td>8 GB</td>
</tr>
<tr>
<td><strong>Compute ability</strong></td>
<td>——</td>
<td>6.1</td>
</tr>
</tbody>
</table>

---

⁴ https://github.com/danielemichilli/SpS  
⁵ https://mariadb.org/
Since PRESTO was originally developed based on CPU serial programs, the code was gradually optimized to support parallel programs with OPENMP and MPICH, but the overall program was based on CPU implementation. Therefore, to achieve GPU accelerated prepsubband calculation, it is necessary to redesign the de-dispersion acceleration algorithm based on the GPU programming model. From Algorithm 1, the loop of time sampling is divided into two parts by the delay matrix, the part with delay less than numpts minus delays gets data.
Fig. 3 When the number of channels is 256, 512, 1024, and 2048 the gpuprepsubband is compared with the acceleration effect of prepsubband running 1 process. It can be seen from the figure that as the number of tested DMs increases, the acceleration effect becomes better and better; when the number of DMs is relatively large, the acceleration effect of short files is better than that of long files.

from lastdata matrix, and the rest gets data from the data matrix. This type of design will result in divergence on the GPU, thereby reducing the performance of dedispersion. To eliminate the divergence, we merge the two matrices and produce a new matrix with \(2 \times \text{numpts} \times \text{numchan}\). As displayed in Algorithm 2, we revised the dedispersion algorithm with \(\text{numpts} \times \text{numdm}\) threads. Each thread computes the sum of a single time sample over every channel sequentially. Unlike Algorithm 1, for each thread, we add the sum of samples over channels to a variable which is stored in the GPU registers, thus reducing the number of accesses to global memory.

For input data preparation, in order to be compatible with the original PRESTO, the original code has not been changed, which can ensure that the parameters of the dedispersion input are consistent with the original parameters, thereby ensuring that the result of the GPU computation is the same as that of the original CPU.

3 PIPELINE TESTS

In this section, we present the test of GSP to the data streams from the CRAFTS survey.

Algorithm 2 Pseudocode of the parallel dedispersion algorithm on GPU

Require: data array with \(2 \times \text{numpts} \times \text{numchan}\), delays array with \(\text{numdm} \times \text{numchan}\), \text{numpts}, \text{numchan}

Ensure: result array with \(\text{numpts}\)

\[
\text{tid} = \text{blockIdx.x} \times \text{blockDim.x} + \text{threadIdx.x}
\]

if \(\text{tid} < \text{numpts} \times \text{numdm}\) then

\[
\text{resultvar} = 0
\]

for \(\text{ii} = 0 \rightarrow \text{numchan}\) do

\[
\text{jj}=\text{ii} \times 2 \times \text{numpts} + (\text{tid} \mod \text{numpts}) + \text{delays} \left[ \left\lfloor \frac{\text{tid}}{\text{numpts}} \right\rfloor \times \text{numchan} + \text{ii} \right]
\]

\[
\text{resultvar} += \text{data}[\text{jj}]
\]

end for

result[\text{tid}] = resultvar

end if

3.1 Experiments of GPU Accelerated De-dispersion

The FAST UWB receiver achieves a better than 60 K receiver temperature covering 270 MHz–1.62 GHz. The beam-passing time of a drift scan equals approximately 52 s at frequency of Bands in Table 1. With proper weighting, the equivalent integration time per beam will be around 65 s. The recorded FAST data stream for pulsar observations is a time series of total power per frequency channel, stored in PSRFITS format (Hotan et al. 2004)
When the number of channels is 256, 512, 1024 and 2048, the gpuprepsubband is compared with the acceleration effect of mpiprepsubband running 15 processes. It can be seen from the figure that as the number of tested DMs increases, the acceleration effect becomes better and better; when the number of DMs is relatively large, the acceleration effect of short files is better than that of long files.

Table 5: Comparison of Eight Pulsars both Detected by FAST and Parkes

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>P0</th>
<th>DM</th>
<th>Number of pulses detected by FAST</th>
<th>Number of pulses detected by Parkes</th>
<th>Max S/N of pulses detected by FAST</th>
<th>Max S/N of pulses detected by Parkes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B1822-09</td>
<td>0.769006</td>
<td>19.38</td>
<td>83714</td>
<td>76403</td>
<td>35.35</td>
<td>20.26</td>
</tr>
<tr>
<td>2</td>
<td>B1839+09</td>
<td>0.381319</td>
<td>49.16</td>
<td>101928</td>
<td>296</td>
<td>53.86</td>
<td>9.28</td>
</tr>
<tr>
<td>3</td>
<td>B1845-01</td>
<td>0.659432</td>
<td>159.53</td>
<td>13431</td>
<td>168190</td>
<td>12.42</td>
<td>22.17</td>
</tr>
<tr>
<td>4</td>
<td>B1846-06</td>
<td>1.451319</td>
<td>148.17</td>
<td>345565</td>
<td>11054</td>
<td>79.3</td>
<td>30.38</td>
</tr>
<tr>
<td>5</td>
<td>J1823-0154</td>
<td>0.759777</td>
<td>135.87</td>
<td>13431</td>
<td>70</td>
<td>17.66</td>
<td>8.97</td>
</tr>
<tr>
<td>6</td>
<td>J1824-0127</td>
<td>2.492468</td>
<td>88.80</td>
<td>14235</td>
<td>109</td>
<td>78.22</td>
<td>11.36</td>
</tr>
<tr>
<td>7</td>
<td>J1852-0635</td>
<td>0.524151</td>
<td>171</td>
<td>24782</td>
<td>3711</td>
<td>58.6</td>
<td>17.33</td>
</tr>
<tr>
<td>8</td>
<td>J2005-0020</td>
<td>2.279661</td>
<td>35.93</td>
<td>5077</td>
<td>27</td>
<td>11.8</td>
<td>9.06</td>
</tr>
</tbody>
</table>

from a ROACH-2\(^5\) based backend, which produces 8-bit sampled data over 256 – 4k frequency channels at 98 µs cadence. For the GPU de-dispersion test in this work, we carried out an experiment by adding simulated pulses into real data. The injected pulses were generated assuming various DMs and then sampled in time and frequency in exactly the same fashion as those of the FAST data. Since the DM of a pulsar is not known a priori, we created de-dispersed time series for each pseudo-pointing over a range of DMs, from 0 to \(\sim 1000\) pc cm\(^{-3}\), which is a factor of three to four larger than the maximum DMs predicted by the NE2001 model (Cordes & Lazio 2002) and YMW16 model (Yao et al. 2017) in the low Galactic latitude regions of the survey. The step size between subsequent trial DMs \(\Delta DM\) was chosen such that over the entire band \(t_{\Delta DM} = \frac{t_{\text{chan}}}{\Delta DM}\). This ensures that the maximum extra smearing caused by any trial DM deviating from the source DM by \(\Delta DM\) is less than the intra-channel smearing. The set of trial values is chosen using the ‘DDplan.py’ tool from PRESTO and is shown in Tables 2 and 3. We tested a total of 48 test files, and the number of DMs start from 30 steps and test to 900 steps. Each test process tests CPU single-

\(^5\) https://casper.ssl.berkeley.edu/wiki/ROACH-2_Revision_2
B1822-09 detected by FAST

B1822-09 detected by Parkes

B1839+09 detected by FAST

B1839+09 detected by Parkes

B1845-01 detected by FAST

B1845-01 detected by Parkes

B1846-06 detected by FAST

B1846-06 detected by Parkes
Fig. 5 Comparisons of DM vs. S/N plots of the eight pulsars that were detected by both FAST and Parkes.
Fig. 6 Single pulse search of 13 FAST pulsars discovered by GSP. The subplots are scatterplots of the DM versus time for each single-pulse event which is proportional to its S/N. The horizontal red line signifies the DM value corresponding to the maximum S/N. Subplots are scatterplots of S/N versus DM. The vertical red line signifies the DM value corresponding to the maximum S/N.
### Table 6 13 New Pulsars Discovered by GSP

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>P0 (s)</th>
<th>DM (cm$^{-3}$ pc)</th>
<th>RA(2000) (h:m:s)</th>
<th>Dec(2000) (±d:m:s)</th>
<th>GL (°)</th>
<th>GB (°)</th>
<th>$D_{NE2001}$ (kpc)</th>
<th>$D_{YMW16}$ (kpc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>J1931-0144</td>
<td>0.593661359782 (12)</td>
<td>38.3 (13)</td>
<td>19:31:32.025 (9)</td>
<td>–01:44:22.5 (4)</td>
<td>35.974</td>
<td>–9.711</td>
<td>1.7</td>
<td>1.4</td>
</tr>
<tr>
<td>2</td>
<td>J1926-0652</td>
<td>1.608816302697 (18)</td>
<td>85.3 (7)</td>
<td>19:26:37.041 (16)</td>
<td>–06:52:42.7 (4)</td>
<td>30.751</td>
<td>–10.936</td>
<td>2.9</td>
<td>5.3</td>
</tr>
<tr>
<td>4</td>
<td>J0402+4825</td>
<td>0.512194448728 (3)</td>
<td>85.7 (3)</td>
<td>04:02:40.633 (9)</td>
<td>+48:25:57.51 (7)</td>
<td>152.4276</td>
<td>–3.1772</td>
<td>2.3</td>
<td>1.8</td>
</tr>
<tr>
<td>5</td>
<td>J2129+4119</td>
<td>1.68741829528 (1)</td>
<td>32 (1)</td>
<td>21:29:21.46 (4)</td>
<td>+41:19:55 (1)</td>
<td>87.1948</td>
<td>–7.1093</td>
<td>2.3</td>
<td>1.9</td>
</tr>
<tr>
<td>6</td>
<td>J0021-0909</td>
<td>2.31413082912 (16)</td>
<td>87.3 (4)</td>
<td>00:21:51.47 (3)</td>
<td>–09:09:58.7 (11)</td>
<td>210.065</td>
<td>–21.582</td>
<td>&gt;25</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>J0803-0942</td>
<td>0.571256559156 (14)</td>
<td>21.1 (3)</td>
<td>08:03:26.848 (13)</td>
<td>–09:42:50.81 (14)</td>
<td>230.070</td>
<td>11.197</td>
<td>1.3</td>
<td>0.8</td>
</tr>
<tr>
<td>8</td>
<td>J1951+4724</td>
<td>0.18192759882 (1)</td>
<td>104.35 (8)</td>
<td>19:51:07.45 (3)</td>
<td>+47:24:35.1 (2)</td>
<td>81.1828</td>
<td>10.3473</td>
<td>6.0</td>
<td>9.0</td>
</tr>
<tr>
<td>9</td>
<td>J1822+2617</td>
<td>0.591417585722 (1)</td>
<td>64.7 (1)</td>
<td>18:22:44.819 (2)</td>
<td>+26:17:26.83 (4)</td>
<td>54.0268</td>
<td>17.5535</td>
<td>3.6</td>
<td>7.8</td>
</tr>
</tbody>
</table>

Notes: The parameter are obtained from the timing solutions utilizing the 64-m Parkes radio telescope (Cameron et al. 2020) and the 100-m Effelsberg radio telescope (Cruces 2021 MNRAS accepted, Wang Shen 2021 RAA accepted).

3.2 Confirmation of Eight Known Pulsars with Cross-matching in GSP & Parkes Database

We used the servers of the FAST early scientific data center to perform a single pulse search on the data of Band1 and Band2. After single pulse identification with RRATtrap.py in PRESTO, 2,244,298 single pulse candidates were classified and recorded into the database. Of the classified candidates, 1,469,415 were classified as being RFI or noise, 100,791 as excellent astrophysical candidates (with group rank 6), and 674,092 candidates with group rank in 3, 4 or 5. In these candidates, we compared the records of FAST UWB single pulse database with Parkes database and found eight known pulsars both detected by FAST and Parkes (Zhang et al. 2020), as affirmed in Table 5 and Figure 5. For pulsar B1845-01, since the FAST drifting scan is far from the pulsar's position, the S/N detected by FAST was weaker than that of Parkes.

4 DISCOVERY OF 13 PULSARS FROM THE CRAFTS SURVEY

Using the GSP pipeline, we discovered 13 new pulsars as listed in Table 6, and applied FFA (Staelin 1969) for periodical search. For the single pulses adjacent to the DM, the maximum range of the FFA search is 1.5 times the time interval of the two single pulses adjacent to the DM; the follow-up timing observation has been completed by Parkes and Effelsberg Radio Telescope. These pulsars were reported upon (Cameron et al. 2020; Qian et al. 2019; Cruces et al. 2021; Wang et al. 2021a; Zhang et al. 2019; Wang et al. 2021b) and the discovery plots of GSP for each of the 13 pulsars are displayed in Figure 6.

---

https://crafts.bao.ac.cn/pulsar/
5 CONCLUSIONS

We described a new, more efficient single pulse search pipeline, namely GSP, to search for pulsars, RRATs and FRBs. The pipeline relies on GPUs to accelerate dedispersion, and utilizes CPU parallel threading to implement single pulse search, single pulse identification and folding processes. Our main results are as follows:

- GSP integrates data preparation, dedispersion, single pulse search, candidate ranking and candidate archiving.
- GSP implements the PRESTO dedispersion package ‘prepsubband’ with CUDA, which was shown to be \( \sim 120 \) times faster than the original ‘prepsubband’ and \( \sim 60 \) times faster than the MPI version for processing CRAFTS data.
- We detected all known pulsars in the overlapping sky of the FAST-UWB observation and Parkes Multibeam surveys.
- GSP discovered 13 new FAST pulsars in the pilot CRAFTS survey with the FAST-UWB.

Acknowledgements This work is supported by the National Natural Science Foundation of China (NSFC; Grant Nos. 11988101, 11725313, 11690024, 12041303, U1731238, U2031117, U1831131 and U1831207), and supported by the Science and Technology Foundation of Guizhou Province (No. LKS[2010]38). PW 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


Hotan, A. W., van Straten, W., & Manchester, R. N. 2004, PASA, 21, 302
Li, D., Wang, P., Qian, L., et al. 2018
Qian, L., Pan, Z., Li, D., et al. 2019, Science China Physics, Mechanics, and Astronomy, 62, 959508
Wang, H., Zhu, W., Guo, P., et al. 2019, Science China Physics, Mechanics, and Astronomy, 62, 959507
Wang, P., Li, D., Clark, C. J., et al. 2021a, Science China Physics, Mechanics, and Astronomy, 64, 129562
Wang, S., Zhu, W.-W., Li, D., Pan, Z.-C., et al. 2021b, RAA (Research in Astronomy and Astrophysics), 21, 251
Yu, Q.-Y., Pan, Z.-C., Qian, L., et al. 2020, RAA (Research in Astronomy and Astrophysics), 20, 091