Draft version August 24, 2014
Preprint typeset using LaTeX style emulateapj v. 12/16/11

GENERALIZED EVENT FINDING IN THE FREQUENCY-TIME DOMAIN

J. B. Burt
Department of Astronomy and Center for Radiophysics and Space Research,
Cornell University, Ithaca, NY 14853, USA

Draft version August 24, 2014

ABSTRACT

We present a general event-finding algorithm for detecting arbitrarily shaped, high signal-to-noise regions in frequency-time domain data, and apply it to the detection of fast radio transients. This event-finding approach to transient detection is compared to de-dispersion techniques and other single-pulse search algorithms, both implemented in searches for periodic dispersed pulses from radio pulsars. We discuss multiple algorithms potentially suited for this purpose before justifying our implementation of a 'friends-of-friends' algorithm, in which individual data pixels exceeding a low intensity threshold are consolidated into contiguous 'blobs' of pixels which must exceed a higher cumulative intensity threshold; each blob exceeding the cumulative threshold becomes a candidate event which can be characterized and classified in a detection pipeline. We inform our choice of threshold values and quantify false-positive detection rates using simulations on Gaussian noise, before presenting the results of the algorithm's application to archival data.

Subject headings: event finding, radio pulses, dedispersion, transient

1. INTRODUCTION

Automated data-processing pipelines are required to thoroughly comb the vast data archives produced by astronomical surveys for the signatures of astrophysical processes. Pipelines analyzing frequency-time domain data from radio pulsar surveys are often equipped with algorithms optimized to detect periodic signals. These detection pipelines utilize trial dispersion measures to attempt to remove the frequency-dependent delay of electromagnetic radiation as it travels through the ionized plasma in the interstellar medium; the data are then reduced to one-dimensional time-series by summing along the frequency dimension. The time-series constructed using the correct trial dispersion measure (DM) will contain the entire broadband signal in the fewest number of time bins, equivalent to the signal's temporal extent, maximizing the signal-to-noise ratio (SNR) of the radio pulse in the time-series. Trial boxcar smoothings of varying temporal size can then be applied to the dedispersed time-series to reduce noise at different characteristic timescales. Fast Fourier Transforms (FFTs) or data-folding can then be employed to identify any underlying periodic pulse structure in the dedispersed time-series (Lorimer & Kramer 2004). While many known pulsars have been discovered using this approach, there are regimes in which periodic pulses are better detected by single-pulse search algorithms (Cordes & McLaughlin 2003).

Single-pulse searches also commonly operate on dedispersed time-domain data, utilizing matched filtering to locate strong individual pulses from intermittent sources (Deneva et al. 2009). This technique is not biased against singular, aperiodic events like the FFT or data-folding approaches, but it still assumes a dispersion relation with frequency dependence $\sim \nu^{-2}$. This detection scheme also discards spectral information in reducing the data to a one-dimensional time-series; it relies upon the entire signal being collected and localized to a few neighboring time bins in order to achieve a statistically significant detection above the background of noise. On the contrary, exploiting the full resolution of the two-dimensional frequency-time domain in an event-finding algorithm would allow us to treat signals of all frequency dependencies equally.

Single-pulse searches have recently been effective in discovering a handful of new classes of astrophysical objects, including rotating radio transients (RRATs) by McLaughlin et al. (2006), and fast radio bursts (FRBs) by Lorimer et al. (2007). These discoveries have illustrated the necessity for more robust singular event-detection algorithms; fast transient radio bursts originating from other highly energetic astrophysical phenomena may also someday be detectable if such algorithms are developed. Signatures from several novel and exotic sources, including merging neutron star binaries, black hole explosions, and magnetically-braked supramassive rotating neutron stars, are potentially observable extragalactic FRBs (Totani 2013; Rees 1977; Falcke & Rezzolla 2014).

To meet the need for a general event-finding algorithm, we have developed a pipeline that operates on full-resolution frequency-time domain data with minimal built-in assumptions about the nature of latent sources. The algorithm identifies regions of unusually high SNR, irrespective of any particular distribution or orientation within the frequency-time plane. This generalized approach is not partial to particular timescales, although we apply it to fast radio transients defined by characteristic timescales of less than a second. It is also not biased towards any specific frequency dependence, so long as the frequency-dependent delay across the observed band lies within our sub-second timescale.

We introduce and compare several potential algorithms for transient event detection in § 2. In § 3 we present a detailed description of the friends-of-friends algorithm, and apply it to simulated Gaussian noise to numerically place constraints on the algorithm's parameter-space.

We present results of the application of our pipeline to archival data in § 4.

2. COMPARISON OF ALGORITHMS

We seek an algorithm that identifies high SNR regions within two-dimensional arrays of chronologically sampled dynamic spectra, where intensity is recorded as a function of frequency and time; we assume N_ν frequency channels span a bandwidth B and are sampled every Δt_s seconds. For faint astrophysical signals smeared out across the frequency-time plane, we expect the mean intensity in a region containing a signal to be only marginally boosted. This shift may be imperceptible to unaided human eyes, but the hope is that a robust algorithm will reliably detect such deviations above the background, if and only when they are present. To this end, we considered multiple algorithms of varying levels of complexity, from a simple, single-pixel threshold to a bank of matched filtering templates.

2.1. *Simple Threshold*

The simplest conceivable algorithm to separate signals from noise has only a single parameter, implemented as a threshold on intensity. This algorithm returns pixels above this intensity threshold, which can then be explored for evidence of an astrophysical signal. This parameter is effectively implemented in units of the root-mean-square (RMS) of a data set, so that the threshold value and the SNR are equivalently representative of a detection's significance. That is, for an array of data spanning T seconds, with N_ν frequency channels and $N_t = T/\Delta t_s$ time samples, we want the set of pixels for which

$$\frac{I_{j,k}}{\sigma_p} > n, \tag{1}$$

where $I_{j,k}$ is the intensity of the pixel in the j-th frequency channel and k-th time bin, σ_p is the single pixel RMS of the data, and n is the adjustable discrimination threshold.

For our purposes, the shortcomings of this algorithm outweigh the benefit of its simplicity. Data quantization is problematic because each unique pixel intensity value corresponds to a unique SNR value, effectively quantizing the parameter n and imposing a restriction on where the distinction between high- and low-signal pixels can be drawn. When the range of pixel values is very small relative to the total number of data pixels, this restriction is especially salient and undesirable. Digitization also has the effect of placing an upper limit on the SNR of any single pixel – e.g., the SNR of the maximum allowable intensity value. If that upper limit is on the order of a few multiples of the RMS, it will be difficult to distinguish strong signal pixels from noise; random noise will enable individual pixels to saturate in intensity, rendering the discrimination threshold ineffective. These effects can be mediated by smoothing the data, because smoothing by N samples increases the number of possible pixel intensity values by a factor of N as well, but as we will show, far more robust algorithms exist which require very little additional complexity.

This algorithm also fails to acknowledge extended, contiguous regions of high intensity and instead considers only individual pixels. An additional step must be introduced to combine these pixels into distinct clusters within the frequency-time plane to realize the true significance of extended signals; this is increasingly important as data resolution is improved and signals are contained within a larger number of pixels.

2.2. *Flood-fill*

The flood-fill or seed-fill algorithm, well-known for its implementation in paint programs as a fill tool for coloring contiguous monochromatic regions, augments the simple threshold algorithm by introducing one additional threshold and one procedural instruction. The first intensity threshold n_{seed} is identical to the parameter n introduced in § 2.1, except that now it functions as a filter for high-significance seeds or nodes. The second intensity threshold n_{fill} is necessarily lower than n_{seed} and is used to fill or expand a signal into the region surrounding the node; beginning with an 'island' comprised of a single node, adjacent pixels exceeding the fill threshold are iteratively annexed to the island until all possible neighboring pixels have been explored and the process has been exhausted. This procedure is performed for each node, in some cases joining multiple nodes and the aggregate of their appended pixels into a single encompassing island. Each island then represents some region in the frequency-time plane of notably high SNR, possibly indicating the presence of a signal, surrounded on all sides by pixels failing to exceed either threshold.

Unfortunately, this method is plagued by many of the same defects found in the simple threshold algorithm. The nodes will be too numerous and diagnostically ambiguous if the maximum allowable intensity value is not sufficiently high to permit identification of the few, most significant nodes. Data quantization will restrict the choices of n_{fill} and n_{seed}, imaginably preventing a user from selecting appropriate thresholds high enough to omit spurious noise while remaining low enough to detect legitimate signals. As before, these drawbacks are somewhat remediable by introducing a degree of data smoothing. For reasons discussed in § 2.5, we chose not to implement the flood-fill algorithm in our pipeline.

2.3. *Friends-of-friends*

The friends-of-friends algorithm bears resemblance to the flood-fill algorithm, but it is methodologically inverted: instead of beginning at a high-intensity node and trickling down to adjacent low-intensity pixels, it builds up contiguous low-significance pixels into islands of high accumulated significance. The first of two thresholds, n_{pix}, functions as a low single-pixel intensity threshold. These pixels are then grouped into distinct collections, such that each collection of pixels forms a contiguous, isolated island in the frequency-time plane, each demarcated by a perimeter of pixels not exceeding n_{pix}. For each of these islands, or 'blobs' as they will henceforth be referred to, we compute the SNR of the accumulated signal of its constituent pixels. For a blob comprised of

N pixels, we compute

$$\bar{I} = \frac{1}{N}\sum_{j=1}^{N} I_j, \tag{2}$$

$$\sigma_N = \frac{\sigma_p}{\sqrt{N}}, \tag{3}$$

$$s \equiv \frac{\bar{I}}{\sigma_N} = \frac{\bar{I} \times \sqrt{N}}{\sigma_p} \tag{4}$$

where I_j is the intensity of the j-th pixel in the blob, σ_p is the single-pixel RMS in the frequency-time plane, and s is the quantity we associate with the SNR of a sample of N aggregate pixels. In a sample of N independent pixels, the effective RMS scales like $N^{-1/2}$, motivating our use of σ_N as the denominator in equation 4. This introduces a $\sqrt{N}$-dependency in the signal-to-noise ratio s, an effect we will discuss in § 3.1.

The second parameter, $n_{\rm blob}$, is implemented as a threshold on s. This blob thresholding procedure selectively picks out blobs with high accumulated significance. Blobs with $s > n_{\rm blob}$ are identified and reported by the algorithm as candidate signal detections.

The advantage to this approach is also its flaw: it reliably and indiscriminately identifies arbitrarily-shaped regions of collective accumulated signal. The difficulty lies in discerning these regions' origins. To detect very weak signals, we choose $n_{\rm pix}$ to be only a fraction of σ_p, with the understanding that many, if not the majority, of pixels exceeding $n_{\rm pix}$ are simply due to noise. The goal is to select an appropriate choice of $n_{\rm blob}$ so as to minimize the false-positive detection rate while retaining as many of the true signals as possible, under the assumption that s is a good choice of test statistic for discriminating true signals from blobs arising solely from noise fluctuations. The specific details of the threshold selection are discussed in § 3.3 when we introduce our simulation results.

2.4. *Template Bank*

Fundamentally different from the other algorithms presented here, the template bank is a set of predefined kernels which are correlated with an array of data to produce matched filters. For a given template, the SNR will be maximized for a frequency-time signature of the same shape as the template; when the correct template is convolved with a true signal, white Gaussian noise is optimally suppressed and the SNR is accordingly boosted.

The advantage of a template bank is the freedom to choose templates with any particular shape or distribution in frequency and time, generating an optimal linear filter for detecting signals with a given signature. The disadvantage is that the infinite expanse of imaginable templates can never be realized by finite computational capabilities. The template bank is therefore impractical for a blind event-finding algorithm, although it may be appropriate for searches with a well-defined class of target sources.

2.5. *Selecting an Algorithm*

The simple threshold algorithm satisfies the need for generalized event-finding, but its restricted consideration of individual pixels without any consolidation procedure is unequipped for the detection of extended, multi-pixel features. A template bank is not generalized by construction, and better suited for a targeted search for specific classes of astrophysical sources in which constraints on the anticipated signals narrow the spectrum of appropriate templates. The two most promising candidates, friends-of-friends and flood-fill, are both characterized by high- and low-intensity thresholds and a contiguity constraint, resulting in comparable operative behavior.

To quantify the performance of these algorithms, we constructed ROC curves by applying friends-of-friends and flood-fill to simulated data containing modeled dispersed pulses superimposed on a background of white Gaussian noise. ROC curves are utilized in signal processing to illustrate the behavior of a binary classification procedure as a function of the algorithm parameters. Each point on an ROC curve corresponds to a true positive rate (TPR) and a false positive rate (FPR) for a particular chosen combination of thresholds; because each algorithm has two parameters which can be independently varied, we constructed the curves by dialing one threshold while holding the other constant. The explicit procedure is discussed in more detail following the description of our simulated data sets.

Each simulated data realization contained a single modeled dispersed pulse. Pulses were characterized by three parameters: their DM, assuming a ν^{-2} frequency dependence; their full width at half max (FWHM), the temporal width between the points at which the signal intensity falls to half its peak value, using a one-dimensional Gaussian function to model the pulse intensity as a function of time; and the peak SNR of the pulse, occurring in each frequency channel at the dispersed pulse's temporal center.

We generated one hundred one-second realizations of data, each with 512 frequency channels and 15270 time samples, consistent with the resolution of the Mock spectrometers in operation at the Arecibo telescope. The modeled pulses were constructed channel-by-channel using a one-dimensional Gaussian function of pixel intensities, with the appropriate standard deviation and amplitude to achieve the desired FWHM and SNR. The peak of the pulse in each frequency channel was shifted in time according to the cold plasma dispersion relation, which goes like ν^{-2} and acts to increasingly delay signals arriving at successively lower frequencies.

For each realization, pulse parameters were randomly chosen within a range of allowed values; the DM was randomly chosen between 10 and 750 pc cm^{-3}, and the FWHM was randomly chosen between 1 and 5 ms (equivalent to a span of 15 to 76 time bins for our time resolution of 65μs). Each pulse had a peak amplitude per channel of $1\sigma_p$ (the RMS of the noise background), and spanned the entire band. The simulated pulse was then superimposed on a background of white Gaussian noise. To construct our ROC curves, we averaged the results from the hundred trials at each point along the curve for each uniquely chosen combination of algorithm thresholds.

We pragmatically designed both algorithms to return binary masks illustrating the locations of blob pixels in the frequency-time plane which had met the algorithms' respective thresholding criteria. That is, the output was an array of the same shape as the input data, with ones

at the locations of pixels belonging to blobs and zeroes everywhere else. With this implementation, we utilized a pixel-oriented normalization scheme for our TPR and FPR computations. Letting $N_{\rm FWHM}$ be the number of time bins corresponding to the pulse FWHM (and rounded to the nearest odd integer), we define

$$N_{1/2} \equiv \frac{N_{\rm FWHM} - 1}{2}. \tag{5}$$

Then if t_i^p is the time bin in the i-th channel where the pulse peak occurs, the TPR is computed as

$$\mathrm{TPR} = \frac{\sum_{i=1}^{N_\nu} \sum_{j=t_i^p - N_{1/2}}^{t_i^p + N_{1/2}} M_{i,j}}{N_\nu \times N_{\rm FWHM}}\ , \tag{6}$$

where $M_{i,j}$ is the value at the i-th channel and j-th time bin of the returned binary blob mask.

The FPR was analogously computed by analyzing the binary masks returned after applying the algorithms to pure noise, and determining the fraction of all pixels falsely identified as blob members:

$$\mathrm{FPR} = \frac{\sum_{i=1}^{N_\nu} \sum_{j=1}^{N_t} M_{i,j}}{N_\nu \times N_t}. \tag{7}$$

Note that the FPR denominator is a factor of $N_t/N_{\rm FWHM}$ larger than the TPR denominator, affecting the scalings of the ROC curve axes but not the relevant qualitative behavior.

The ROC curves for friends-of-friends were parametrized by $n_{\rm blob}$, while holding $n_{\rm pix}$ constant; similarly, the curves for flood-fill were parametrized by $n_{\rm seed}$ while holding $n_{\rm fill}$ constant. In each case, it was the high threshold that we varied while choosing different constant values of the low threshold. This procedure is justified despite the apparent methodological inversion of the two algorithms; an easy way to see this is to consider a functionally identical implementation of flood-fill, where contiguous blobs are identified using pixels exceeding the fill threshold, exactly like it is done for friends-of-friends, but requiring a single pixel within each blob to exceed the node threshold instead of constraining the accumulated signal of the entire blob.

Using these parametrizations, we were able to probe the algorithms' behavior as we dialed up the discriminatory parametrization threshold. The values of the parametrization thresholds were deliberately chosen through preceding trial-and-error in an attempt to construct an informative curve containing data points extending from an FPR of 0 up through the highest TPR allowable by the choice of parameters (e.g., when the two thresholds are equivalent and the algorithms effectively operate like the simple threshold). However, because this procedure was computationally intense and intended to be used only as a diagnostic tool, the curves do not appear smooth and differentiable.

The results are displayed in Figure 1: the top subplot displays the friends-of-friends ROC curve, the center subplot illustrates the flood-fill curve, and the bottom subplot provides an overlay of the two. Curves using low threshold values of 0.25, 0.5, 0.75, and 1.0 were constructed, with corresponding high threshold values growing as large as necessary to fully parametrize the curve through the desired range. Notice that increasingly larger values of the low threshold necessarily result in a smaller maximum-possible TPR, simply because fewer signal pixels will exceed the threshold and be considered for blob membership to begin with. A dashed line of unit slope, representing the expected outcome of random chance, is included for reference. We conclude from these tests that the performance of friends-of-friends exceeds that of flood-fill for each low threshold value we attempted, its ROC curves extending further into the optimal upper-left-hand corner of the plot where TPR is high and FPR is low, indicating a high true-positive detection rate and low false-positive detection rate. We therefore selected friends-of-friends as our generalized event-finding algorithm for implementation in our pipeline.

More sophisticated and potentially useful ROC curve normalization schemes are possible, but necessitate the introduction of additional arbitrary parameters. For example, one might consider an event-oriented normalization to replace our pixel-oriented normalization, where multiple signals are injected into each data realization and the TPR is the fraction of those events that are found; however, because the algorithms identify blobs and return blob masks, one must decide what constitutes an event detection. Is it a detection if the algorithm finds one blob containing 50% of the pixels in a simulated signal? Is it a detection if the algorithm identifies several smaller blobs, which together include 75% of the pixels in a simulated signal? If an event-oriented normalization scheme like this is employed, answers to questions like these will need to first be established, and may ultimately be a matter of personal inclination and pragmatism.

3. DEVELOPMENT OF THE FRIENDS-OF-FRIENDS ALGORITHM

The difficulty in practical applications of friends-of-friends lies in distinguishing true astrophysical signals from spurious, probabilistic noise-induced blobs. To improve our algorithm's robustness, we considered different N dependencies for $n_{\rm blob}$ and we introduced additional supplementary parameters to complement the two existing intensity thresholds. We ran simulations on Gaussian noise to inform our choice of thresholds by characterizing the algorithm's behavior in the absence of signals. Together, these modifications and simulations enabled us to fine-tune the algorithm to minimize the frequency of false-positive detections, and narrow our focus to particular regions of the algorithm's parameter space.

3.1. *Selecting an N-dependence*

As we mentioned in § 2.3, the blob test statistic s, which we defined to be the blob's aggregate SNR, has a $\sqrt{N}$ dependence; the algorithm therefore preferentially selects larger blobs, whose higher test statistics will tend to push them above the threshold even when their average mean intensity is low. To compensate for this effect, very high values of $n_{\rm blob}$ are necessary when using low $n_{\rm pix}$ values, because the size of noise-induced blobs increases as $n_{\rm pix}$ decreases. Consequently, successfully rejecting spurious blobs of large N requires rejecting all blobs below some characteristic size, because the digi-

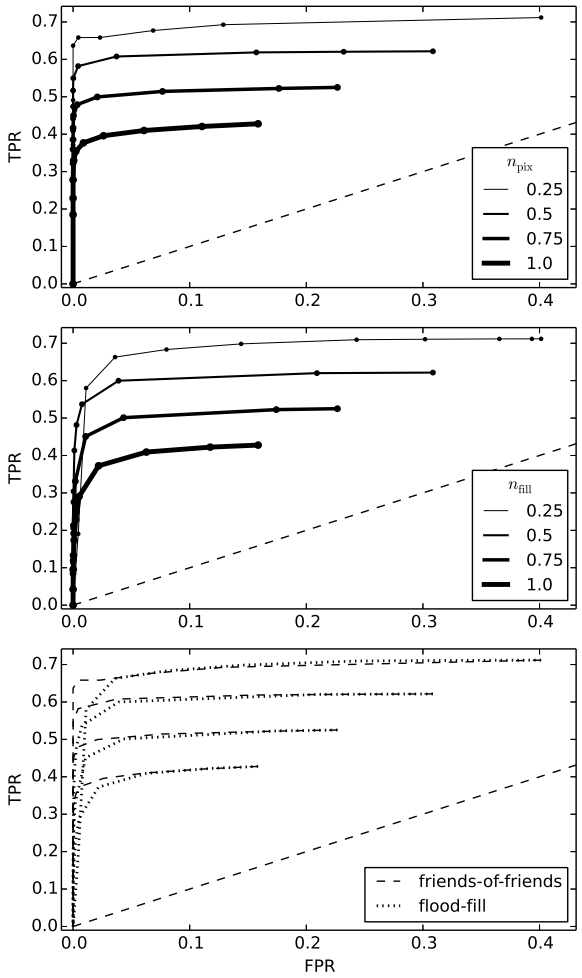

FIG. 1.— *Top*: ROC curves for friends-of-friends; *Middle*: ROC curves for flood-fill; *Bottom*: Overlay of the two preceding plots. The friends-of-friends curve is parametrized by $n_{\rm blob}$, and the flood-fill curve by $n_{\rm seed}$. Curves for low threshold values of 0.25, 0.5, 0.75, and 1.0 were constructed for each algorithm. Dots indicate locations of the 11 data points used to construct each curve. See text for more details.

tized intensities have a finite upper bound and for blobs of sufficiently low N, the mean signal intensity will never be great enough to exceed the imposed threshold. For example, if spurious blobs arising for a given choice of $n_{\rm pix}$ applied to white Gaussian noise of size N_ν by N_t contain up to 10,000 pixels, and on average exhibit a mean signal intensity of $1\sigma_p$, then the threshold necessary to exclude such a blob ($n_{\rm blob} = 100$) would also exclude a 400-pixel blob with mean intensity of $5\sigma_p$! Because we were attempting to construct a generalized algorithm with as few inherent biases as possible, we considered modifying the N dependence to bolster our resistance to false-positives without sacrificing a significant number of true signal detections.

An obvious option is to divide s by $\sqrt{N}$, to get a completely N-independent test statistic; however, referring to equation 4, this amounts to effectively thresholding the mean pixel intensity. Because we know that random fluctuations will give rise to one- or several-pixel blobs of very high mean intensity, especially in large data sets with many pixels drawn from a Gaussian distribution, having no N-dependence at all is a poor choice when false-positive rejection is considered. After all, a large blob of moderate mean intensity is more interesting than a very small blob of high intensity, within the realm of Gaussian statistics.

Alternatively, a compromising approach between no N-dependence and $\sqrt{N}$-dependence is to divide s by $\sqrt[4]{N}$ to get a $\sqrt[4]{N}$-dependent blob test statistic. Following a similar procedure to the one used in constructing the top subplot in Figure 1, we constructed N-dependent ROC curves for friends-of-friends to compare $\sqrt[4]{N}$ and $\sqrt{N}$ dependencies. As illustrated in Figure 2, both N-dependencies exhibit very similar behavior.

For our pipeline, we ultimately selected a $\sqrt{N}$-dependence, keeping the aggregate blob SNR as our blob test statistic s. This was primarily due to two observed effects. First, because many smaller blobs are identified using a $\sqrt[4]{N}$ dependency, the computational cost of blob characterization is consequently much greater. Additionally, when applied to diagnostic data sets whose signals had been identified beforehand using other methods, the algorithm's lack of bias towards large N actually became detrimental, as far too many blobs were being identified by chance association or due to radio frequency interference (RFI), obscuring any real astrophysical signals. Considering the nearly indistinguishable ROC curves, the $\sqrt{N}$-dependence seemed sufficiently meritorious to pursue trial applications to real data sets in our pipeline.

3.2. *Auxiliary Parameters*

The first conceivable choice for an additional parameter is one indicating the number of adjacent samples to smooth prior to applying the intensity thresholds. For user flexibility we allowed independent smoothing sample sizes along each dimension, resulting in the addition of two parameters we designate $N_{s,\nu}$ and $N_{s,t}$, referring to the number of samples smoothed along the frequency and time axes, respectively. Smoothing by a factor of N ($\equiv N_{s,\nu} \times N_{s,t}$) samples decreases the RMS by a factor of $\sqrt{N}$. The smoothing procedure is most efficacious when applied to data containing a signal of temporal extent $N_{s,t}$ time bins and spectral extent $N_{s,\nu}$ frequency channels – the reduced RMS, together with the non-zero mean signal, preferentially boosts the SNR within the signal region.

The second set of supplementary parameters incorporated into the algorithm specify allowable pixel gaps, defined independently along each axis and labelled $N_{g,\nu}$ and $N_{g,t}$; these parameters enable pixels to be considered members of a single contiguous blob in spite of small 'gaps' of pixels failing to exceed $n_{\rm pix}$ due to noise fluctuations. These parameters are pragmatic in origin and intended to improve overall algorithm robustness and resistance to the vagaries of real astronomical data. However, because we have chosen $n_{\rm pix}$ to be less than or on the order of the RMS, allowed gaps must be sufficiently small to avoid returning a single blob blanketing the entire frequency-time plane; this will occur when the mean inter-pixel separation between pixels exceeding $n_{\rm pix}$ is on the order of the allowed gap sizes. A detailed numerical analysis should be performed to identify the regimes in

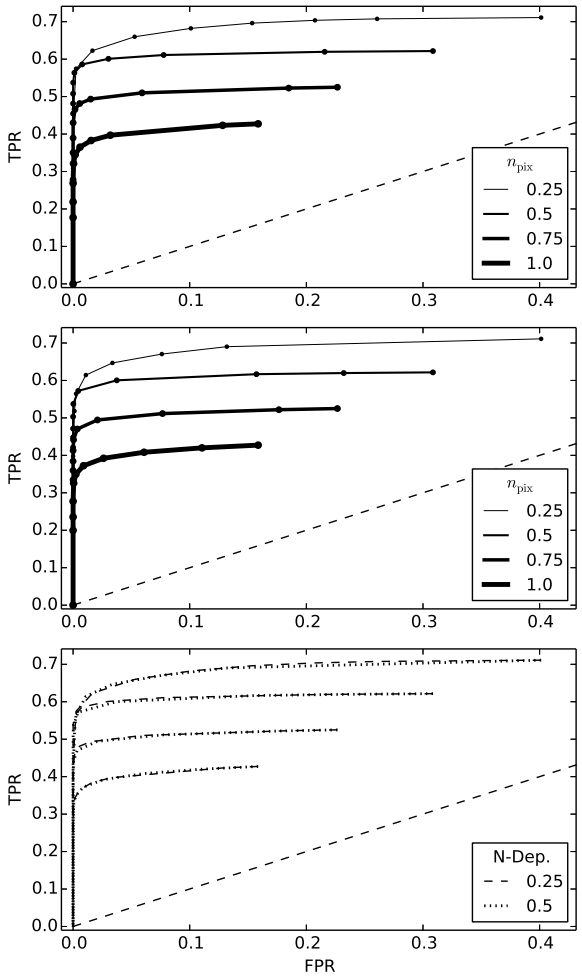

FIG. 2.— *Top*: ROC curve using a $\sqrt[4]{\rm N}$-dependent test statistic; *Middle*: ROC curve using a $\sqrt{\rm N}$-dependent test statistic; *Bottom*: Overlay of the two preceding plots. Both curves are parametrized by $n_{\rm blob}$, using whatever values were necessary in each case to fully parametrize the curve. Curves for low threshold values of 0.25, 0.5, 0.75, and 1.0 were constructed for each algorithm. Dots indicate locations of the 11 data points used to construct each curve.

which this limiting behavior occurs.

3.3. *Simulations on Gaussian Noise*

Applying friends-of-friends to white Gaussian noise provides a better understanding of the algorithm's behavior in the absence of any signals. We used such simulations to explore two critical questions: how frequently do noise-originating blobs of varying size occur as a function of $n_{\rm pix}$, and what combinations of thresholds provide appropriate false-positive detection rates?

3.3.1. *Histograms of Blob Sizes Prior to $n_{\rm blob}$ Thresholding*

Using the same set of $n_{\rm pix}$ thresholds that were used in constructing the ROC curves, we performed 100 trials for each threshold on 512-N_ν by 15270-N_t arrays of randomly generated intensities following a Gaussian distribution with zero mean and unit RMS. In each trial, we did not apply $n_{\rm blob}$. We recorded the size of each blob identified by the algorithm, and then constructed histograms of blob sizes as a function of $n_{\rm pix}$. The results are contained in Figure 3. The histogram for each $n_{\rm pix}$ value was constructed identically, with 10 bins initially chosen linearly spaced between blob sizes of zero and the maximum blob size occurring in all trials for that threshold. The final n bins in which less than one average count occurred were consolidated to a single bin, and the preceding $10 - n$ bins were reconstructed to create nine linearly spaced bins. Notice the significant $n_{\rm blobs}$ drop-off for successive size bins for the lowest $n_{\rm pix}$ value of 0.25; this drop in counts for increasingly larger blobs becomes far more gradual and consistent across bins as $n_{\rm pix}$ rises to 1.0. This makes intuitive sense, as very low thresholds will necessarily pick out more pixels, which then have a greater chance of lying adjacent to other such pixels, resulting in the accumulation of fewer, larger blobs.

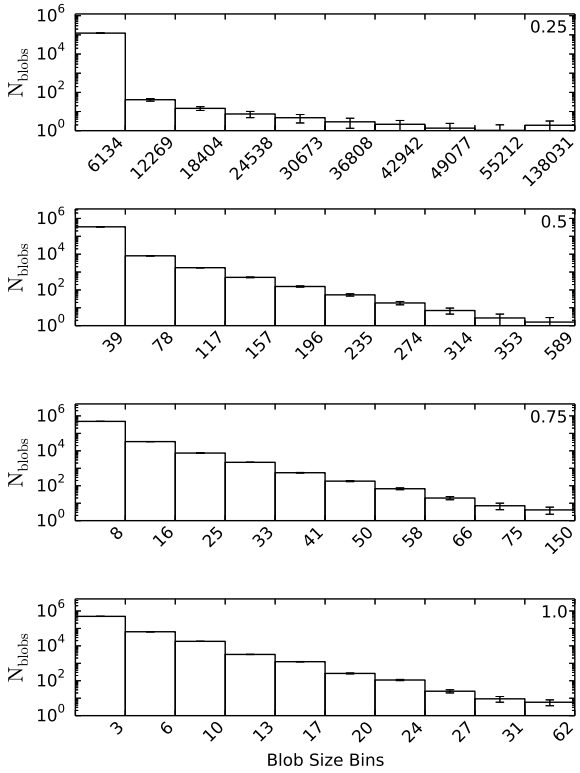

FIG. 3.— Histograms of blob sizes for $n_{\rm pix}$ values of 0.25, 0.5, 0.75, and 1.0 (shown in the upper-right-hand corner of each subplot). The thresholds were applied to 100 realizations of simulated intensities following a zero-mean Gaussian distribution, each realization containing 512 frequency channels and 15270 time samples.

3.3.2. *Rates of False-positive Detections*

If the threshold choices for $n_{\rm pix}$ and $n_{\rm blob}$ are not carefully chosen, the algorithm's discriminatory power will be diminished. For each unique combination of thresholds, there will be some corresponding characteristic response of the algorithm to Gaussian noise. Spurious blobs will appear with varying frequency and size as a function of the chosen thresholds.

To quantify our false-positive detection rates we again applied friends-of-friends to simulated Gaussian noise. The algorithm was applied to 100 realizations, each containing 512 frequency channels and 15270 time samples. To establish a reference point for the case when $n_{\rm blob}$ was

omitted, we applied $n_{\rm pix}$ to pick out individual pixels just like in § 3.3.1, and then analyzed the size and frequency of the identified blobs. The results are contained in Table 1, where the mean and RMS for the sizes and frequencies of blobs are presented as a function of $n_{\rm pix}$.

We then constructed a reference table for varied $n_{\rm blob}$ choices, as a function of $n_{\rm pix}$ and smoothing sample sizes, for a predetermined false-positive detection rate. Blobs identified within 100 realizations of Gaussian noise were recorded, and we subsequently determined the $n_{\rm blob}$ value necessary to keep only the four blobs with the highest test statistic, yielding a false-positive detection rate of 0.04. We chose this rate because PALFA data sets are typically on the order of 270 seconds, and we intended to allow approximately 10 false-positive detections per data set. For those four remaining blobs, we recorded the mean and maximum number of pixels. The results of this procedure are contained in Table 2.

4. APPLICATION TO REAL DATA

Before applying the algorithm to real data sets, we had to embed it into a more robust data-processing pipeline. The pipeline enables a user to provide a raw data set as input and be returned a detailed list of identified blobs and their properties, accompanied by images representing blob locations and orientations within the data set. To accomplish this, blobs had to be characterized, compared, combined, and filtered.

Once the locations of blobs within the frequency-time plane have been identified by the friends-of-friends algorithm, we compute a variety of quantities associated with each blob. These quantities include the SNR, centroid location in frequency and time, characteristic widths along each dimension, best-fit slope and curvature, roundness (computed by identifying the orientation axis and comparing the ratio of the second moments about that axis), blob size, and bandpass occupation fraction, among others. Each blob's attributes can then be compared to those of other nearby blobs: a sufficient degree of similarity among multiple attributes is suggestive of a larger, overarching blob which was erroneously broken up into smaller sub-blobs in the friends-of-friends procedure. That is, it is likely that the algorithm mistakenly identified multiple blobs that all belonged to one extended signal, presumably due to noise disruption. By establishing a threshold for blob similarity contingent on the relative quantitative similitude of several key blob attributes, we further reduce the number of blobs by consolidating those exhibiting a sufficient degree of likeness.

After exhausting this "friends" paradigm, first by grouping individual pixels with the friends-of-friends algorithm and then by consolidating apparent sub-blobs using the blob attribute similarity-thresholding procedure, we have completed our search for latent astrophysical signatures. We selectively filter out blobs of particularly high or low slope (using the computed best-fit slope) in the frequency-time plane, suggestive of broadband and narrowband radio frequency interference (RFI), respectively.

The following subsections describe interesting blobs potentially astrophysical in origin, identified utilizing this pipeline for the analysis of real archival PALFA data.

REFERENCES

Cordes, J. M., & McLaughlin, M. A. 2003, ApJ, 596, 1142
Deneva, J. S., Cordes, J. M., McLaughlin, M. A., et al. 2009, ApJ, 703, 2259
Falcke, H., & Rezzolla, L. 2014, A&A, 562, A137
Lorimer, D. R., Bailes, M., McLaughlin, M. A., Narkevic, D. J., & Crawford, F. 2007, Science, 318, 777
Lorimer, D. R., & Kramer, M. 2004, Handbook of Pulsar Astronomy
McLaughlin, M. A., Lyne, A. G., Lorimer, D. R., et al. 2006, Nature, 439, 817
Rees, M. J. 1977, Nature, 266, 333
Totani, T. 2013, PASJ, 65, L12

TABLE 1
GAUSSIAN SIMULATION RESULTS PRIOR TO $n_{\rm blob}$ THRESHOLDING

$n_{\rm pix}$	$N_{\rm pixels}$	$N_{\rm blobs}$	Mean Blob Size	Max Blob Size
0.25	3137500 ± 1484.7	123480 ± 508.19	25.41 ± 0.11338	77779 ± 24548
0.5	2412300 ± 1328.1	349170 ± 661.68	6.9087 ± 0.015818	402.17 ± 55.998
0.75	1771900 ± 1204.9	528800 ± 551.03	3.3509 ± 0.0048613	90.82 ± 9.5324
1.0	1240400 ± 1049.4	578520 ± 470.86	2.1441 ± 0.0022706	38.03 ± 3.7026

NOTE. — The friends-of-friends algorithm was applied to 100 realizations of simulated intensities following a Gaussian distribution, each containing 512 frequency channels by 15270 time samples, and the results were averaged. Included in the results are 1σ uncertainties. $n_{\rm pix}$ is the single-pixel threshold; $N_{\rm pixels}$ is the number of pixels contained within blobs (e.g. exceeding $n_{\rm pix}$), expected to be consistent with a Gaussian survival function; and $N_{\rm blobs}$ is the number of blobs identified. Blob size is found to decrease and blob frequency to increase as $n_{\rm pix}$ is dialed up, consistent with the histogram results in figure 3.

TABLE 2
False-Positive Detection Rate Reference

$n_{\rm pix}$	Smoothing Samples	$n_{\rm blob}$	Mean Size	Max Size
0.25	1, 1	317.1	115222.0	123736.0
0.25	1, 2	30.9	1964.25	2523.0
0.25	1, 4	14.6	630.25	735.0
0.25	1, 8	9.6	382.75	414.0
0.25	1, 16	7.6	300.0	362.0
0.25	2, 2	15.0	646.75	712.0
0.25	2, 4	9.4	376.5	409.0
0.25	2, 8	7.4	289.25	301.0
0.25	2, 16	5.9	246.25	297.0
0.25	4, 4	7.6	301.5	332.0
0.25	4, 8	5.8	229.5	286.0
0.25	4, 16	4.8	170.75	224.0
0.25	8, 8	4.8	210.0	239.0
0.25	8, 16	3.8	138.5	152.0
0.25	16, 16	2.4	78.75	82.0
0.5	1, 1	26.2	537.5	586.0
0.5	1, 2	12.0	150.5	156.0
0.5	1, 4	8.2	105.25	109.0
0.5	1, 8	6.4	74.0	88.0
0.5	1, 16	5.4	53.25	60.0
0.5	2, 2	8.3	105.25	111.0
0.5	2, 4	6.6	78.5	91.0
0.5	2, 8	5.6	54.75	71.0
0.5	2, 16	4.5	47.75	51.0
0.5	4, 4	5.2	65.5	73.0
0.5	4, 8	4.2	46.0	60.0
0.5	4, 16	2.9	29.0	34.0
0.5	8, 8	2.6	24.75	29.0
0.75	1, 1	14.0	107.5	110.0
0.75	1, 2	8.6	56.25	61.0
0.75	1, 4	6.7	34.25	43.0
0.75	1, 8	5.5	31.25	37.0
0.75	1, 16	4.7	25.25	28.0
0.75	2, 2	6.3	33.75	37.0
0.75	2, 4	5.0	25.25	28.0
0.75	2, 8	4.3	26.0	31.0
0.75	2, 16	2.9	13.25	15.0
0.75	4, 4	4.1	18.0	20.0
0.75	4, 8	2.8	12.0	13.0
1.0	1, 1	10.7	45.25	48.0
1.0	1, 2	7.4	21.75	23.0
1.0	1, 4	5.8	15.75	19.0
1.0	1, 8	5.1	15.75	19.0
1.0	1, 16	4.0	12.5	14.0
1.0	2, 2	5.6	14.75	15.0
1.0	2, 4	4.5	11.0	13.0
1.0	2, 8	3.4	9.0	10.0
1.0	2, 16	1.0	2.25	3.0
1.0	4, 4	3.2	7.25	8.0

Note. — Numerically determined $n_{\rm blob}$ values, as a function of $n_{\rm pix}$ and smoothing sample size, for which four false-positively identified blobs were allowed in 100 realizations of one second of simulated Gaussian noise, including the blobs' mean and maximum sizes. The friends-of-friends algorithm was applied to 100 realizations of simulated Gaussian noise, each containing 512 frequency channels and 15270 time samples. The simulated pixel intensities followed a Gaussian distribution, with zero mean and unit RMS. $n_{\rm pix}$ is the single pixel threshold; Smoothing samples are the numbers of pixels smoothed independently along each dimension; $n_{\rm blob}$ is the blob test statistic threshold; and Mean Size and Max Size are the mean and maximum sizes of the four blobs exceeding $n_{\rm blob}$ in the 100 trials. This table is intended to be used to select threshold combinations with a known false-positive detection rate.