Internship Report

- Data extraction and basic image processing -

Aswin Manohar

Supervisor: Dr. Ralph Eatough

Max Planck Institute for Radio Astronomy

Contents

Chapter 1

Introduction

In this project, we extract the vital aspects present in the .pfd file of the the raw pulsar data folded by PRESTO software. The information from each pulsar candidate is synthesized in four diagnostic plots, with image data ranging up to thousand pixels. The 2D array plots are pre-processed using an pattern recognition technique called Hough transformation to detect lines in them, hinting towards the presence of a pulsar.

This data from these plots can then be used to recreate the plot and can further be fed into an AI. The AI mimics human experts and distinguishes pulsars from noise and interference by looking for patterns from candidate plots.

1.1 FEATURE EXTRACTION

The most important subplots in the .pfd files are : Time versus phase plot (2D array), Summed profile (1D array) , Dispersion Measure curve (1D array) and frequency versus phase plot (2D array). Although we extracted the data from all these plots, we managed to pre-process the data only from the time versus phase plot. The frequency versus phases plot can also be pre-processed using the same technique applied to the former. The other two subplots are one dimensional array , we did not subject them to pre-processing, however, we present the basic idea of extracting its key features. An example of extracting the array of the Time vs phase is presented below:

```
from DATAExtract.training import *
x=pfddata('GBT_Lband_PSR_4.62ms_Cand.pfd')
y=pfddata.time_vs_phase(x)
plt.imshow(y,interpolation='none')
plt.show()
```

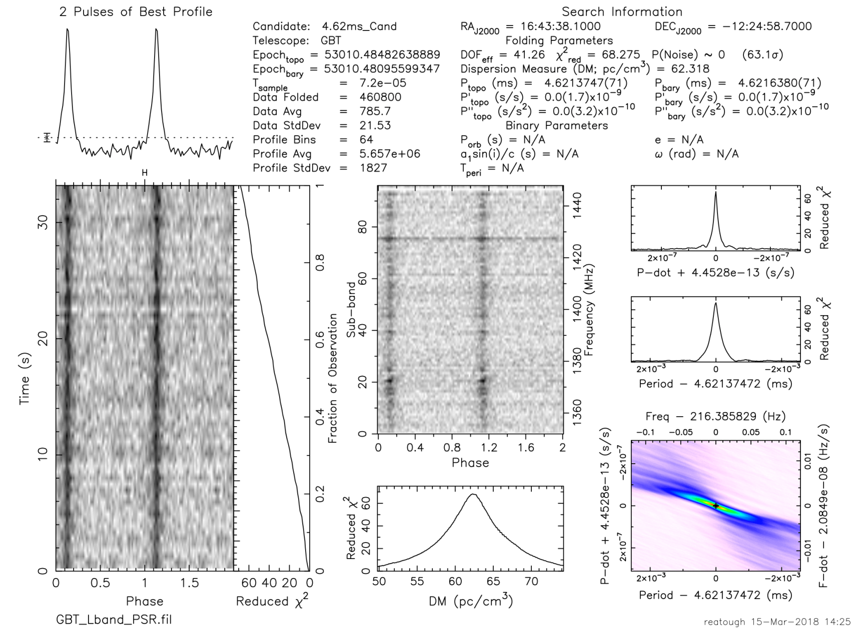


Figure 1.1: Candidate plot of GBT-Lband PSR-4.62 ms

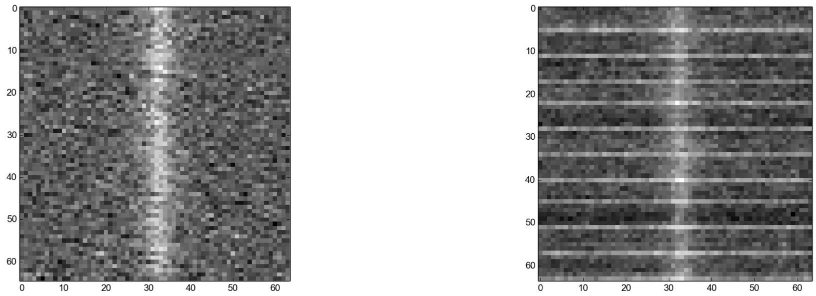
The output is shown in fig:1.2, (b)

Time versus Phase: This plot is obtained by summing the data over the different frequency channels. One or more vertical stripes in this plot indicates that a pulsed signal was observed for the duration of the scan.

Summed profile: We can sum all frequency channels and time intervals to create a summed intensity versus phase pulse profile. Pulse profiles of real pulsars are usually composed of one or several very narrow peaks.

Frequency versus phase plot: Summing the data cube over the different time intervals leaves the frequency versus phase plot. The presence of one or more persistent vertical lines in this subplot, as in the example, indicates a broadband signal during the pulsed emission, as expected for a pulsar candidate. However, scintillation caused by the interstellar medium may affect a pulsar's signal, degrading the signal in some frequency channels.

DM curve: The plotting program searches over a range of DMs around the best reported value. For each DM trial, it dedisperses the data cube accordingly and calculates the χ^2 of the dedispersed pulse profile against a horizontal line fit. The DM curve is a plot of the trial DMs against their corresponding $\chi 2$ values. A large χ 2 value indicates that the periodic signal deviates strongly from simple white noise. The DM curve of a real pulsar will likely peak at a nonzero value unless affected by strong RFI.

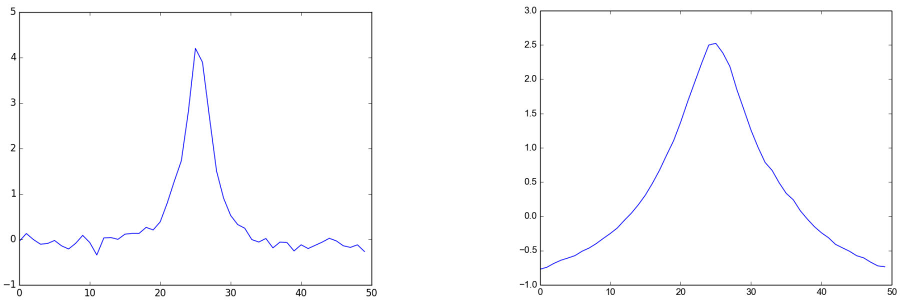


(a) Frequency vs Phase

(b) Time vs phase

Figure 1.2

We made a python script that can extract all the data from these subplots and store them as a binary file. The program is constructed by using dependencies from the PRESTO software and ubc-AI(PICS) [1]. Then we used the extracted data from the time versus phase subplot to find the hough transformation of its image. The task then was to see how efficient is hough transformation for detecting pulsars. So we used a sample of pulsar and non-pulsar candidates to see how many hough votes and how the theta and rho values (values equivalent to coordinates, x and y in hough space) change. The plots realized from the extracted array data from the pfd files are shown.



(a) pulse profile

(b) DM curve

Figure 1.3

1.2 HOUGH TRANSFORM

1.2.1 Introduction

The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing.The purpose of the technique is to find imperfect instances (lines,circles and ellipses) of objects by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is constructed by the algorithm for computing the **Hough transform**.

Since we are looking for straight lines in an image. We take a point (x,y) in the image, all lines which pass through that pixel have the form of the line equation. y=mx+b, For each pixel, all of the possible lines through it are represented by a single line in space. A single line which passes through various pixel points in the x_n,y_n space, can be realized as a intersection point of all the lines in the Hough space(m and b)

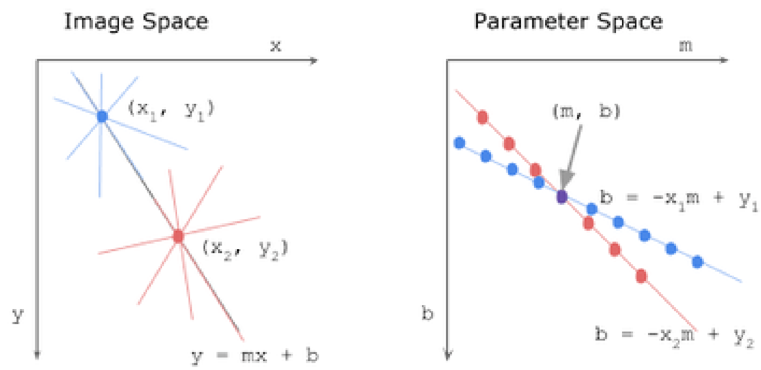


Figure 1.4: Interpretation of Hough transform from image to parameter space

As shown in the above figure, for each pixel, all possible lines through it are represented by a single line in the hough space or all pixels which lie on the same line in (x,y) space are represented by lines which all pass through a single point in (m,b) space. But the drawback is that, the line equation form breaks down for vertical lines as the value of m becomes undefined. For this reason we use an alternate form, the angle-distance parameter space.

$$\rho = xcos\theta + ysin\theta \tag{1.1}$$

ρ is the distance from origin (0,0) to the line and θ is the angle from origin to the line.

1.2.2 Method:

We first binarize the image so that only the brightest pixels which is visualized as a line in the image space are visible by filling the brightest with 1s and the rest of the image array with zeros.

```
y=pfddata.time_vs_phase('GBT_Lband_PSR_4.62ms_Cand.pfd')
img=np.zeros_like(y)
img[np.arange(len(y)),y.argmax(1)]=1
```

The binarized image will have 0s indicating non-edges and 1s indicating edges. This is our input image.

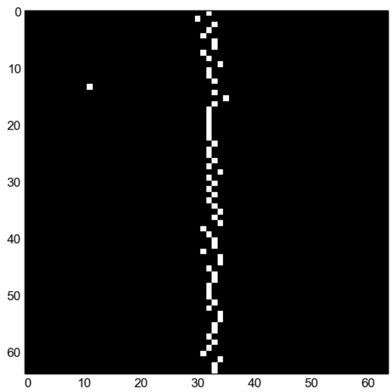


Figure 1.5: Binarized image

It is then fed into the hough transform algorithm and transformed to hough space by calculating the corresponding ρ values with a point at each angle ranging from -90° to 90°.

- First, we create ranges for θ and ρ as mentioned earlier.
- The hough accumulator is defined as a 2D array with number of rows equal to the ρ values and the columns as θ values.
- Now, for each edge point and for each θ value, it finds the nearest ρ value and increments that index in the accumulator.
- Each element in the accumulator array tells how many points/pixels contributed "votes" for potential line candidates with parameters (ρ,θ).
- Finally, we can find the peak, that is,the point with the most line intersections. Local maxima in the accumulator indicates the parameters of the most prominent lines in the input image.

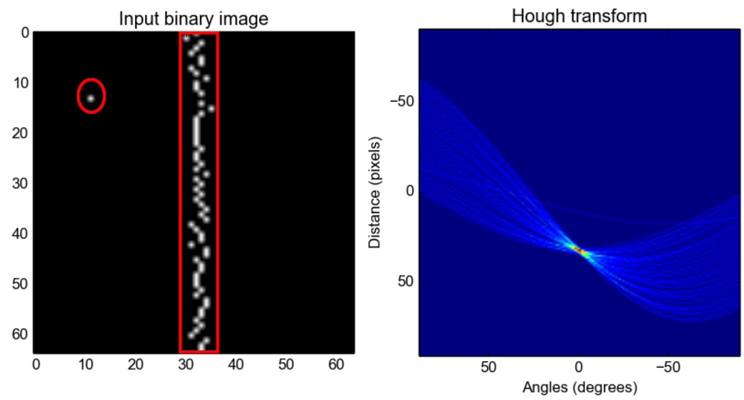


Figure 1.6: Hough transformed image

The more curves intersect at a point, the more "votes" a line in image space will receive.This feature is quite useful to detect the lines in the time vs phase subplot. The pixel points in a line is represented as a intersection in the peak values of ρ and θ.

Another example with pulse period modified is shown in figure 1.4. The change in the values of θ and ρ are visible. Hough transformed image of a pulsar and a non-pulsar is easily distinguishable. We present the images of a non-pulsars and how they look in the hough space. See fig 1.6.

The more curves intersect at a point, the more “votes” a line in image space will receive. We’ll see this in the next implementation section.

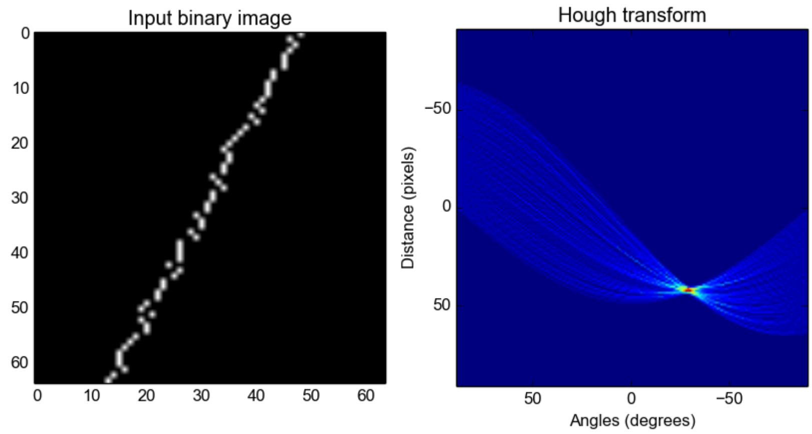


Figure 1.7: Hough transformed image of a pulsar

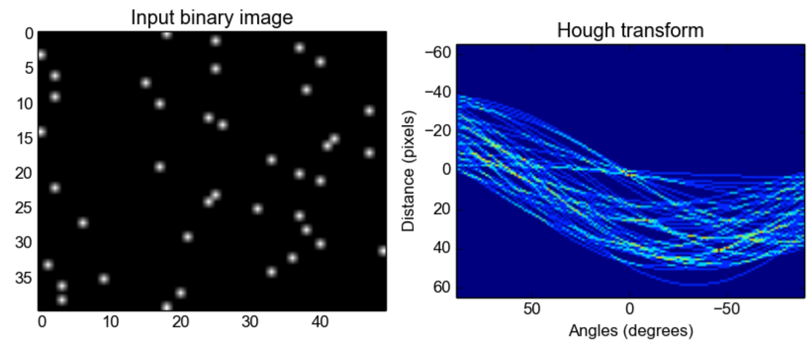


Figure 1.8: Hough transformed image of a non-pulsar

Chapter 2

Conclusion

We found the values of θ and ρ by finding the peak using maximum votes. The maximum value of the accumulator for both pulsars and non-pulsars were found and visualized as a histogram. The results are as follows:

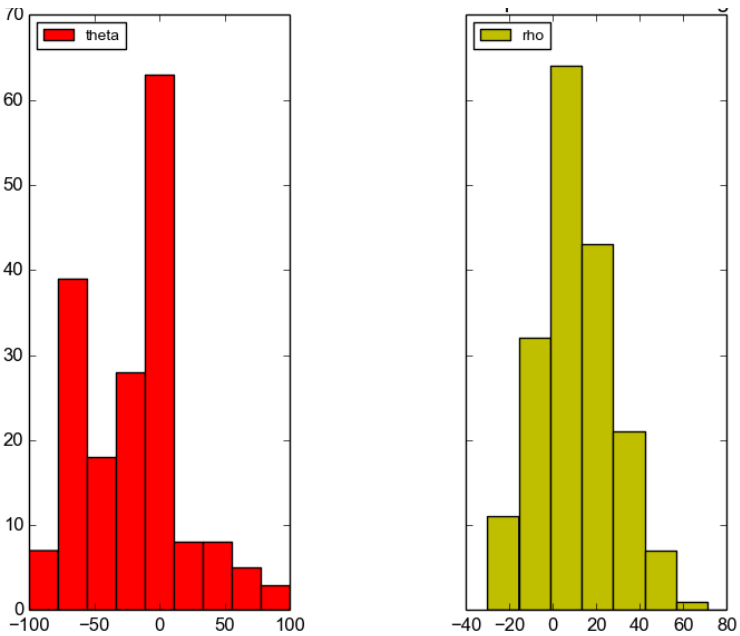


Figure 2.1: Distribution of hough space values for Non-pulsars

The maximum values of the accumulator tells us about the peak.

Hough transform proves to be a reliable image processing technique to identify bright pulsar sources. Nevertheless it fails with faint sources. Due to its feature of voting procedure, it often identifies non-pulsars as pulsars. From the below histogram plotted on the basis of maximum votes, that is, maximum intersection of

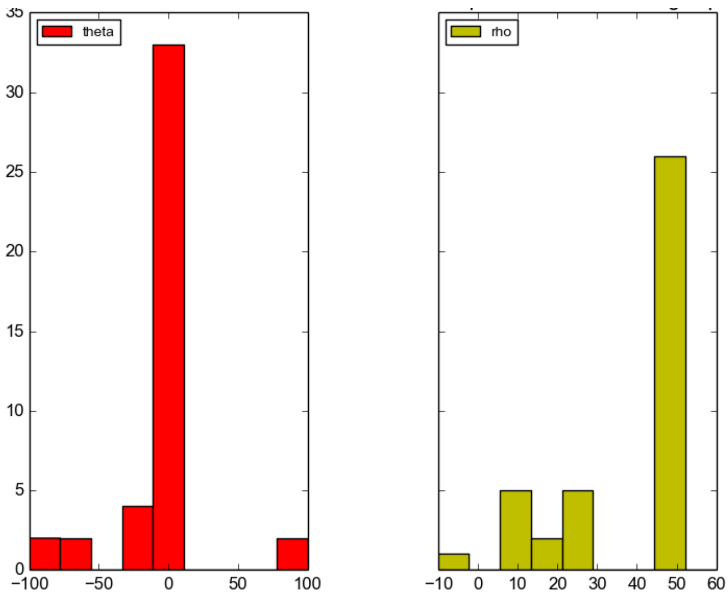


Figure 2.2: Distribution of hough space values for pulsars

lines at a point: It is evident that most of the non-pulsars have less votes compared to that of pulsars.Although, there are few sources which are pulsars, but still with less votes. This is a problem that we need to tackle. Pulsars are known to have votes ranging from 25-30 and non-pulsars below 10-15. This pattern is broken for faint sources and for sources which are not pulsars by a thin line.

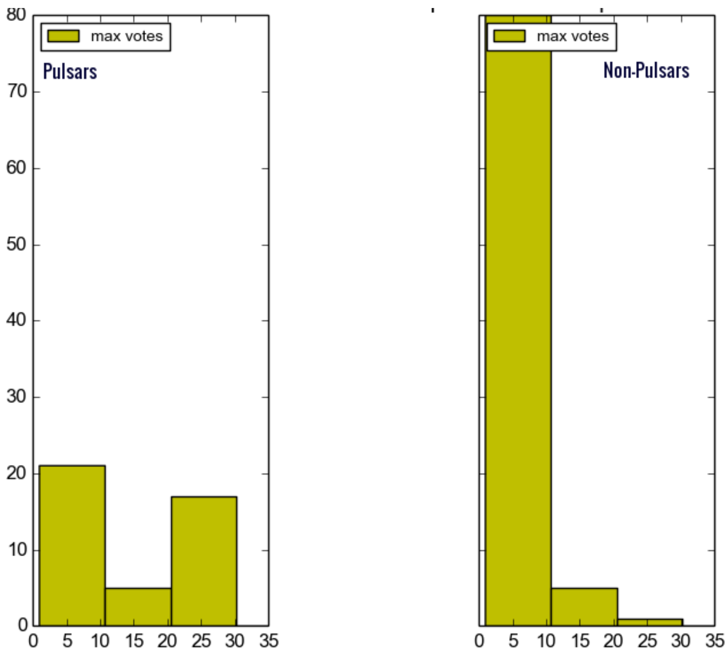


Figure 2.3: Distribution of pulsars and non-pulsars based on maximum votes

Bibliography

[1] [**W.W. Zhu et al**] *Searching for pulsars using image pattern recognition*, ApJ 781 (2014) 117 Searching for pulsars using image pattern recognition

[2] [**Hough transform**] Alyssa Quek: Understanding hough transform

[3] Lecture on Hough transform,
`https://webmail.mpifr-bonn.mpg.de/roundcube`

[4] [1]Pulsar Image-based Classification System

[5] Chritopher Bishop:
`Pattern Recognition and Machine Learning`