INDIAN INSTITUTE OF TECHNOLOGY ROPAR

"DEVELOPMENT OF MINE DETECTION TOOL USING GPR AND STUDY OF BEHAVIOR OF ELECTROMAGNETIC WAVES IN DIFFERENT CONDITIONS"

Submitted by-

Gursimer Singh [2021MEB1285]

Harshvardhan Singh [2021MEB1287]

Shubha Tanaya [2021MEB1325]

Under the guidance of-

Dr.Manish Agrawal

DEPARTMENT OF MECHANICAL ENGINEERING

IIT ROPAR

RUPNAGAR, PUNJAB

Acknowledgements

We are profoundly grateful to **Dr. Manish Agrawal** for his expert guidance and continuous encouragement throughout to see that this project meets its target since its commencement to its completion.

We would like to express our deepest appreciation towards **Dr. Harpreet Singh**, our Faculty Advisor for providing us with this opportunity.

At last we must express our sincere heartfelt gratitude to all the Staff members of the Mechanical Engineering Department who helped us directly or indirectly during this course of work.

- **Gursimer Singh** [2021MEB1285]
- **Harshvardhan Singh** [2021MEB1287]
- **Shubha Tanaya** [2021MEB1325]

Contents:

List of Figures:

1. Introduction:

Landmines are a major global problem that will be addressed by the project "Development of Mine Detection Tool using GPR and Study of Behavior of Electromagnetic Waves in Different Conditions." Long after hostilities have stopped, civilians are still seriously at risk from landmines, both anti-personnel and anti-tank ones. Every year, they cause thousands of deaths, with children being the most severely impacted demographic. The United Nations reports that landmines infest 78 nations and result in 15,000–20,000 fatalities annually. Civilians account for about 80% of landmine casualties, with children being the most commonly impacted age group.

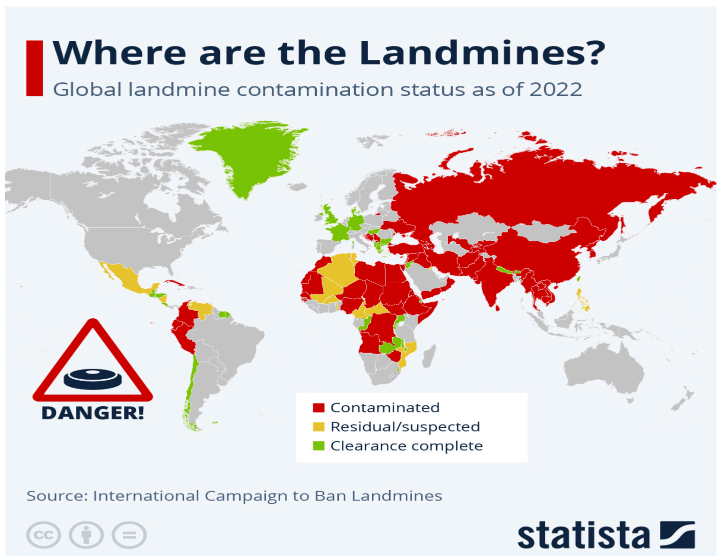


Figure 1.1 Landmine contamination status as of 2022

The principal aim of the project is to utilize Ground Penetrating Radar (GPR) technology to provide a dependable, effective, and secure approach for landmine detection. With a variety of soil types, GPR is a non-invasive subsurface sensing device that can identify metallic and non-metallic mines. The project's electromagnetic wave simulation tools, called gprMax, are based on the numerical Finite-Difference Time-Domain (FDTD) method and will be used to model GPR.

Land-penetrating radar, also known as ground-penetrating radar (GPR), is a non-invasive geophysical method that uses radar pulses to image the subsurface. This technology is particularly useful in detecting buried objects, including landmines, making it a critical tool. Land-penetrating radar operates by sending high-frequency radio waves into the ground from a transmitter. When these waves encounter different materials or objects underground, like landmines, they are reflected back to the surface and captured by a receiver.

2. Theory Regarding the Project:

2.1. Basic Principles:

Ground-penetrating radar (GPR) systems emit high-frequency electromagnetic pulses into the ground to detect subsurface structures. These pulses, consisting of radio waves oscillating at specific frequencies, are not continuous but sent in short bursts. As these waves travel through different underground materials, they encounter interface boundaries where the Earth's electromagnetic properties change abruptly. The behavior of the radio waves, including their reflection, transmission, and refraction, varies depending on the electromagnetic properties on either side of these interfaces and the angle at which the waves hit them. The primary aim of GPR is to identify these interfaces and gather detailed information about the subsurface structures they delineate.

2.2 Properties and Behavior of EM Waves:

1. **Wave Velocity:** Ground-penetrating radar (GPR) signals are high-frequency radio waves whose velocity depends on the medium's physical properties. Radio waves travel slower through materials with higher dielectric constants. For instance, the dielectric constant of air is 1, resulting in faster wave propagation, whereas water has a dielectric constant of about 80, significantly slowing the wave propagation. Water saturation decreases the velocity in sediments due to their high dielectric permittivities, whereas dry and igneous rocks, with lower dielectric values, facilitate higher propagation velocities.
2. **Skin Depth:** Skin depth (1/ attenuation const) defines the propagation distance at which the amplitude of an electromagnetic wave is reduced by a factor of 1/e i.e. reduced to 37% of its original amplitude.Skin depth, which measures how deeply electromagnetic waves can penetrate a material, varies based on several factors. Generally, higher frequencies result in shallower skin depths, but under the wave regime approximation, skin depth reaches a limit where it no longer depends on frequency. Additionally, materials with lower electrical conductivities and higher dielectric permittivities exhibit larger skin depths, allowing waves to penetrate deeper.
3. **Reflection and Transmission of Radio Waves:** When a radio wave reaches an interface, some of it is reflected and some of it is transmitted across the interface. This results in both a reflected and a transmitted wave. The reflection coefficient can be either positive or negative (sign of the reflection coefficient determines whether the reflected wave experiences a reverse in polarity) and has values between [-1,1].
4. **Refraction of Radio Waves:** Refraction is used to describe the change in propagation direction of a wave due to a change in the propagation medium.The angles at which the incident wave is reflected and refracted are illustrated-

$$sin\Theta 1/V1 = sin\Theta 2/V2$$

3. Tools used for Analysis:

3.1 gprMax Software:

GprMax is open-source software designed for simulating electromagnetic wave propagation in GPR systems. It utilizes Yee's algorithm to solve Maxwell's equations in 3D via the Finite-Difference Time-Domain (FDTD) method. This method is straightforward, explicit, and robust, allowing for the modeling of a wide range of frequencies in a single simulation, which is ideal for the ultra wide-band (UWB) nature of GPR. The primary limitation of FDTD is its need for extensive computational resources due to the requirement of discretizing the entire computational domain based on the highest frequency of interest. GprMax enables users to create customized subsurface models and simulate the necessary GPR data for those scenarios, using EM waves typically in the UHF/VHF range, between 30MHz and 3GHz.

3.2. A Scans and B scans Extraxtracted by gprMax Tool:

An A-scan in ground-penetrating radar (GPR) is a single trace showing the variation in field amplitude over time, recorded by a receiver. It helps identify subsurface structures by displaying reflections from different materials, indicating their location, size, and material properties. For instance, a metal cylinder buried in a dielectric medium would produce a strong reflection, with the time difference between this and other reflections indicating the depth of the object. A B-scan compiles multiple A-scans along a line to illustrate how field amplitude changes over both time and space. This can reveal the shape and orientation of underground objects, such as a hyperbolic reflection from a cylindrical object and a horizontal line from the soil-air interface, aiding in estimating the depth and dimensions of these targets.

```
A scan input: python -m gprMax user_models/cylinder_Ascan_2D.in

A scan output:python -m tools.plot_Ascan user_models/cylinder_Ascan_2D.out

B scan input:python -m gprMax user_models/cylinder_Bscan_2D.in -n 60

B scan combine:python -m tools.outputfiles_merge user_models/cylinder_Bscan_2D

B scan output:python -m tools.plot_Bscan user_models/cylinder_Bscan_2D_merged.out Ez
```

4. Review of Previous Work:

GPR: Ricker (Mexican Hat) Waveform: This offers a good balance between resolution and penetration depth. Suitable for detecting mines at varying depths and in different soil conditions. Provides a broadband signal that can effectively image subsurface features. In the B-scan (Ez of the field component), the initial part of the signal (~0.5-1.5 ns) represents the direct wave from transmitter to receiver. Then comes the reflected wave (~2-3 ns) from the metal cylinder which creates the hyperbolic shape.

4.1 Manipulation of Parameters and Quantification of Scans:

As we know that we can use gpr for the detection of mines and we do changing of different physical parameters of the cylinder (buried object) and waveform for plotting B Scans. So, we have done the manipulation in the source waveform parameters and changed the material properties of soil and generated the B scans.

1. **Altering Source Waveform parameters:**

   Initial: #waveform: ricker 1 1.5e9 my_ricker
   Case1: #waveform: ricker 1 5e8 my_ricker (decreasing frequency)
   Case2: #waveform: ricker 5 1.5e9 my_ricker (increasing maximum amplitude)
   Case3: #waveform: gaussian 1 1.5e9 my_gaussian (changing waveform)

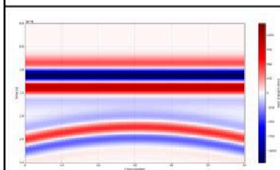
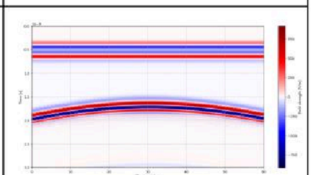
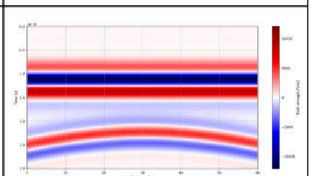
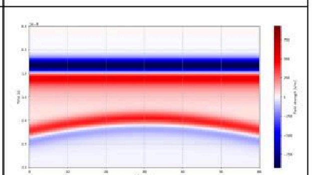
Initial B scan	Altered	Altered B scan
	Case1: #waveform: ricker 1 5e8 my_ricker (decreasing frequency)	
	Case2: #waveform: ricker 10 1.5e9 my_ricker (increasing maximum amplitude)	
	Case3: #waveform: gaussian 1 1.5e9 my_gaussian (changing waveform)	

Figure 4.1 Altering Source Waveform Parameters B Scans

2. Altering Medium Material properties:

a. Initial: #material: 6 0 1 0 half_space
➔ Altering: Permittivity, Conductivity, Magnetic Loss, Permeability 3636
b.Initial: #soil_peplinski: 0.5 0.5 2.0 2.66 0.001 0.25 my_soil
➔ Altering: ◆ Clay Fraction ◆ Sand Fraction ◆ Bulk density of soil ◆Water content range ◆ Density of sand

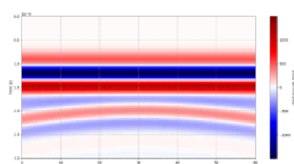
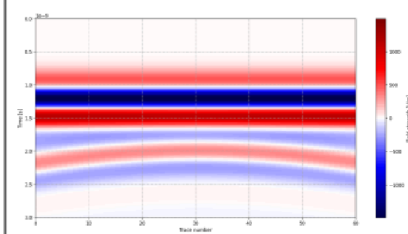
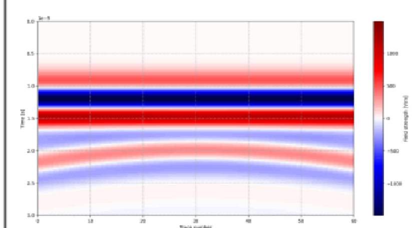
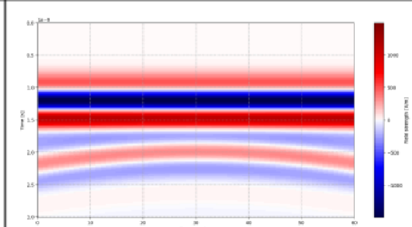
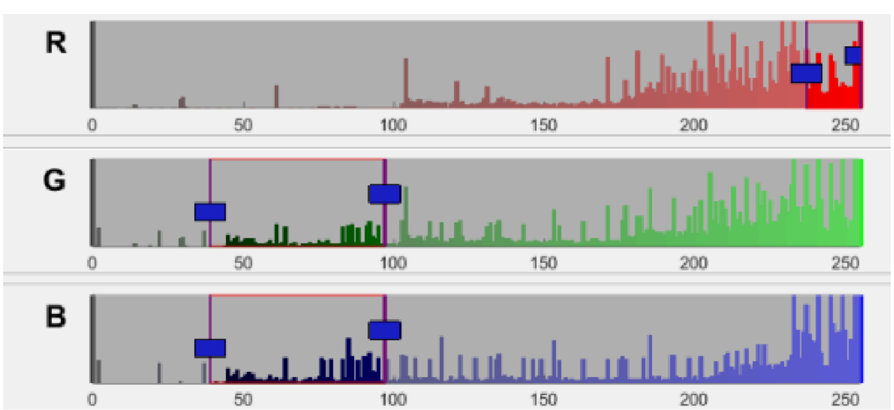
Initial B scan	Altered	Altered B scan
 #material: 3 0.01 1 0 my_sand #soil_peplinski: 0.5 0.5 2.0 2.66 0.001 0.25 my_sand #waveform: ricker 1 1.3e9 my_ricker	Greater sand fraction: 0.8	
	Greater clay fraction: 0.8	
	Increasing water content to 0.5 to 0.7	

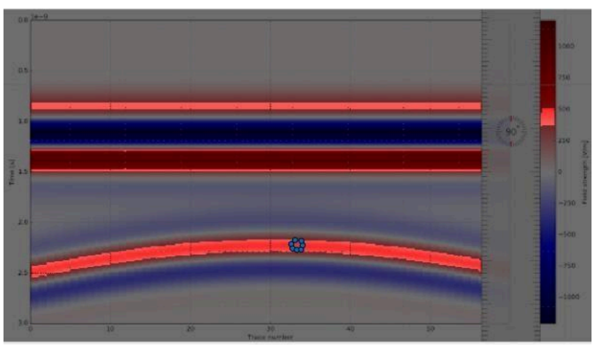
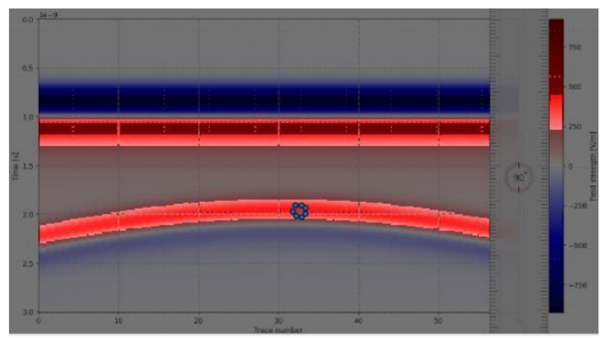
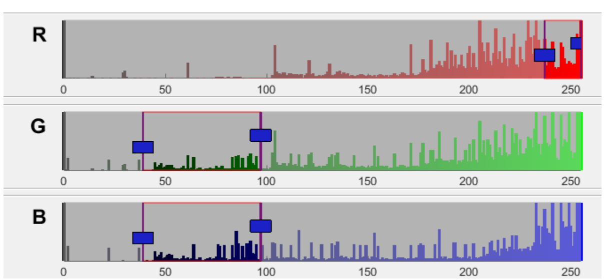
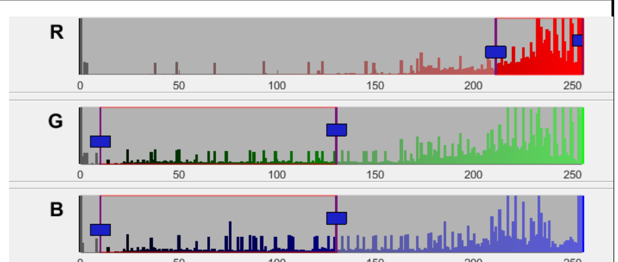
Figure 4.2 Altering Medium Material Properties

We have also done the quantification of B Scan Heat Maps in the B scan image generated by GprMax using Python; the scale of field strength in (V/m) corresponding to different color shades is given. By adding a ruler from powerpoint we further increase its precision. Next, we



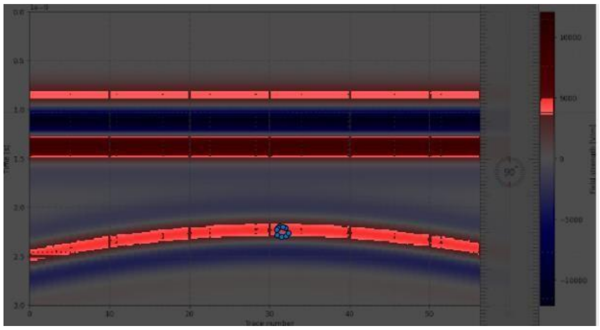
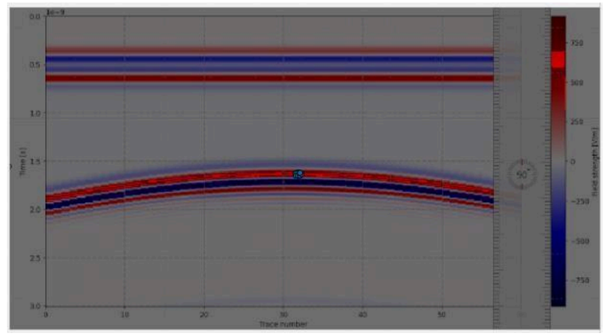
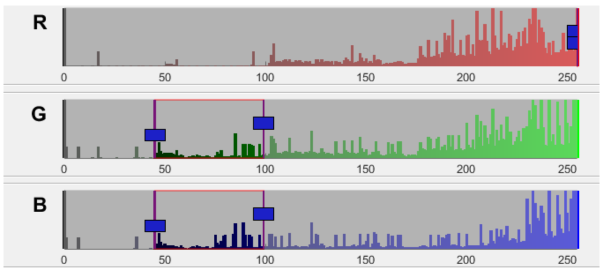
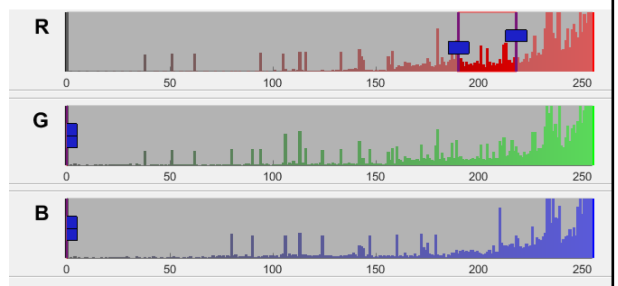
import the image to Matlab’s Image Processing Toolbox- Color Thresholder. We select the RGB color space and select the required region of the plot in order to quantify it.
The corresponding color is highlighted on the scale itself, we note down this reading in (V/m).For the specific region selected for the initial scan, the Field Strength is about 446 V/m. For each scan image we consider 3 regions to finalize the Field strength by taking the average. We do so with every B scan plotted after changing one of the input parameters and then find the mean squared error of individually altered Field Strength with the initial plot’s field strength. The same process is followed by changing 3 parameters and the results are formulated in the table below:

Initial Case:

Initial case	Altered case 1
#waveform: ricker 1 1.5e9 my_ricker	#waveform: gaussian 1 1.5e9 my_gaussian (changing waveform)
	
	
Field Strength (V/m): 446	Field Strength (V/m): 341

Altered Case:

Altered case 2	Altered case 3
#waveform: ricker 10 1.5e9 my_ricker (increasing maximum amplitude)	#waveform: ricker 1 5e8 my_ricker (decreasing frequency)

	
	
Field Strength (V/m): 4118	Field Strength (V/m): 660

Perfect Electric Conductor (PEC) cylinder buried in different Half Spaces:

We also made a simulation using GprMax software and the animation was made using ParaView.

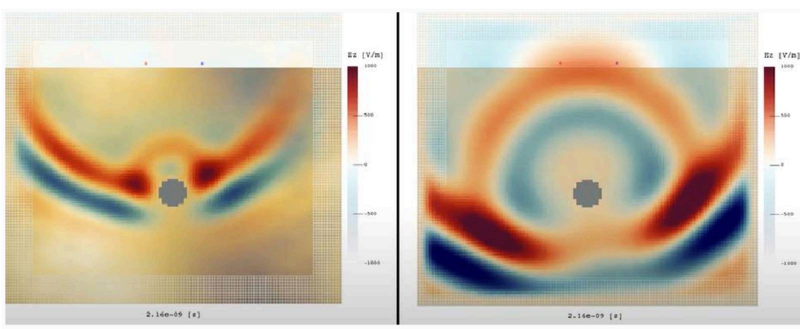


Figure 4.3 Electromagnetic wave propagation in an environment

Animation screenshot of electromagnetic wave propagation in an environment with a perfect electric conductor (PEC) cylinder buried in:

● CASE 1: A lossy, dispersive, heterogeneous environment with static relative permittivities in the range ~10 (light brown color) to ~40 (dark brown color).

● CASE 2: A lossless, homogeneous half-space with a relative permittivity of 6.

The red cell is a Hertzian dipole source (line source in 2D) fed with a Ricker waveform with center frequency 1.5GHz, and the blue cell is a receiver point. Model uses a Peplinski mixing model and fractal distribution of materials.

5. Enhancing Signal Extraction from B-Scans Using Image Processing:

In the process of extracting hyperbolas and removing background noise from B-scan images using image processing, the standard deviation in the RGB values of pixels plays a crucial role. This method aims to ensure efficiency and consistency in the analysis.

Principles and Math for Hyperbola Extraction & Noise Reduction in B-Scan Images:

1. B-scan images, which are often used in fields like medical imaging (ultrasound), geophysics (seismic reflection), and non-destructive testing (ultrasonic testing), display the reflection intensity of a sound wave as it travels through different materials or tissues. Features of interest in these images, like hyperbolas, represent interfaces where the acoustic impedance changes significantly, such as between different subsurface layers or between healthy and diseased tissue.
2. The contrast between these features and the surrounding medium is typically more significant than the background "noise" of the image, which might be due to granular material or other less reflective interfaces. The standard deviation is a measure of how much variation or "dispersion" there is from the average (mean) value. A higher standard deviation indicates more variation and therefore potentially significant features.

Equation used for Calculation:

The standard deviation (σ) for a set of values is calculated using the formula:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\mu)^2}{n}}$$

Where:

- x_i represents each value in the dataset
- μ is the mean of all values
- n is the number of values

For an image, we compute this separately for each row of pixel values (after converting to grayscale to simplify calculations). In a grayscale image, each pixel value represents intensity and is a single number (as opposed to three in a color image), typically ranging from 0 (black) to 255 (white).

5.1 Applying Hyperbola Extraction and Noise Reduction Techniques to B-Scan Images:

1. **Converting the Image to Grayscale:** By reducing the three RGB channels into a single intensity channel, we reduce the computational complexity and focus on intensity variations regardless of color.

2. **Calculated the Standard Deviation:** For each row in the grayscale image, we calculate the standard deviation of the pixel intensities. This gives a measure of contrast variation along the row.

3. **Thresholding:** we now apply a threshold to the standard deviation values to differentiate rows with significant variations (potential features) from those with low variations (likely noise).

4. **Masking:** Created a binary mask where rows above the threshold are marked to be kept, and those below are marked to be altered or removed.

5. **Applying the Mask:** we now apply the binary mask to the original image, replacing the low-variance sections with a uniform background, enhancing the visibility of the features.

6. **Post-Processing:** Additional image processing techniques, such as morphological operations or smoothing filters, can be applied to refine the results.

5.2. Source Code used for above purpose:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
# Load the image
image_path = '/Users/CP301/Desktop/Sample/Screenshot.png' image =
cv2.imread(image_path)
# Convert the image to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# Calculate the standard deviation for each row
std_dev = np.std(gray, axis=1)
# Thresholding: choose a threshold value that might need to be
adjusted manually for the best results
threshold = np.mean(std_dev) # We can change this value based on
experimentation
mask = std_dev > threshold
```

```
# Masking: Created a new image that only contains the rows with high
standard deviation
masked_image = gray * mask[:, np.newaxis]
# Apply the mask to the original image to keep the color information
masked_color_image = image * mask[:, np.newaxis, np.newaxis]
# Post-Processing: Define additional processing steps if necessary
# For example, we can apply a Gaussian blur to smooth the image
# Here, we skip this step because we want to preserve the hyperbolas
as much as possible
# Show the standard deviation plot plt.figure(figsize=(10, 5))
plt.plot(std_dev)
plt.title('Standard Deviation of Each Row') plt.xlabel('Row Number')
plt.ylabel('Standard Deviation') plt.show()
# Show the original, grayscale, and masked images
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1) plt.imshow(cv2.cvtColor(image,
cv2.COLOR_BGR2RGB)) plt.title('Original Image')
plt.axis('off') plt.subplot(1, 3, 2) plt.imshow(gray, cmap='gray')
plt.title('Grayscale Image') plt.axis('off') plt.subplot(1, 3, 3)
plt.imshow(cv2.cvtColor(masked_color_image, cv2.COLOR_BGR2RGB))
plt.title('Masked Image')
plt.axis('off') plt.tight_layout() plt.show()
# Save the masked image to a file
output_path = '/Users/CP301/Desktop/OUTPUTS/Figure_1.png'
cv2.imwrite(output_path, masked_color_image)
output_path
```

5.3 Result obtained from above procedure:

We used the image given below as the sample for the above procedure -

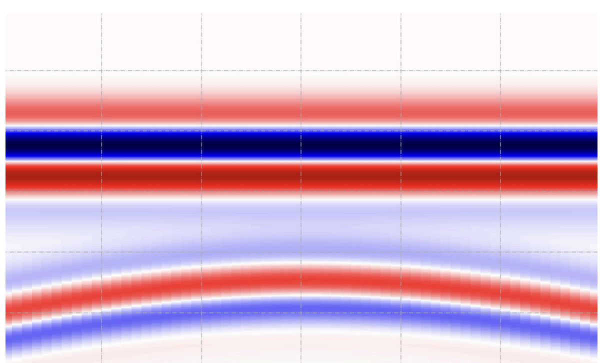


Figure 5.1 Sample B Scan Image

We firstly obtained the graph for the standard deviation of every row (1000+ slices of image is made) and plotted a graph given below -

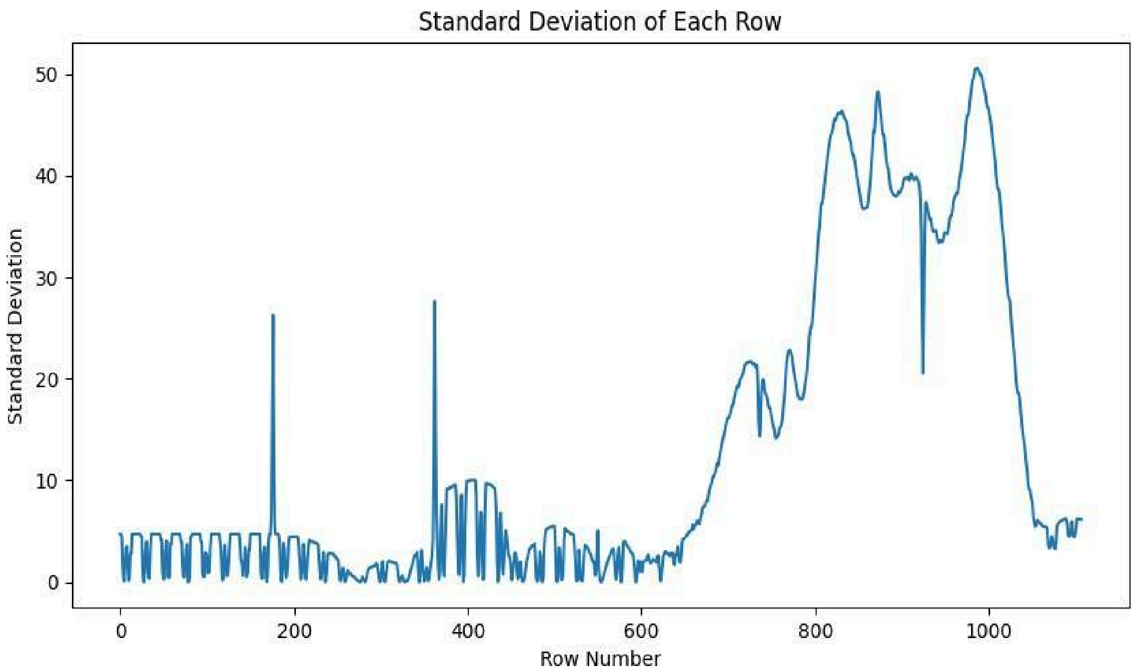
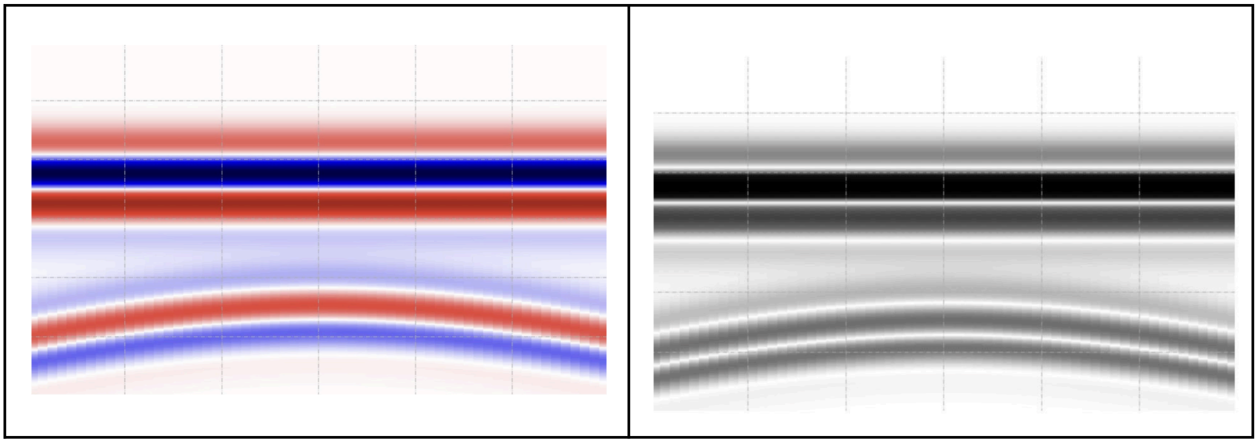


Figure 5.2 Graph of Standard Deviation of B Scan

Converting an image to grayscale involves reducing the three RGB (Red, Green, Blue) color channels into a single intensity channel. This process simplifies the image representation, focusing solely on intensity variations irrespective of color.In RGB images, each pixel is represented by three color channels, where the intensity of each channel contributes to the overall color perception. However, in grayscale images, only one channel is used to represent the brightness or intensity of each pixel. By discarding color information, grayscale images are simpler to process and analyze, which can significantly reduce computational complexity, particularly in tasks where color is not relevant.Converting to grayscale is often beneficial in image processing tasks where color is not essential for analysis, such as edge detection, feature extraction, and pattern recognition.

Original B Scan	Modified Greyscale B Scan



Processed Masked Image Produced:

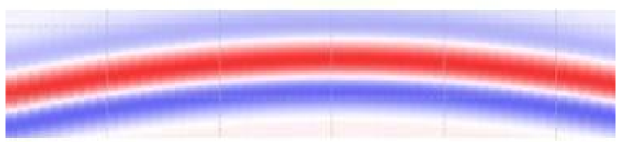


Figure 5.3 Processed Masked Image

In the above image we can see the hyperbola clearly and by using this technique of noise reduction we can increase the efficiency of the ground penetrating radar.

6. Automation of Image Processing procedure:

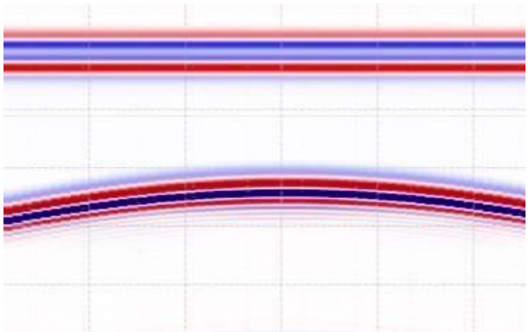
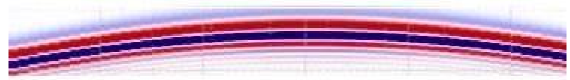
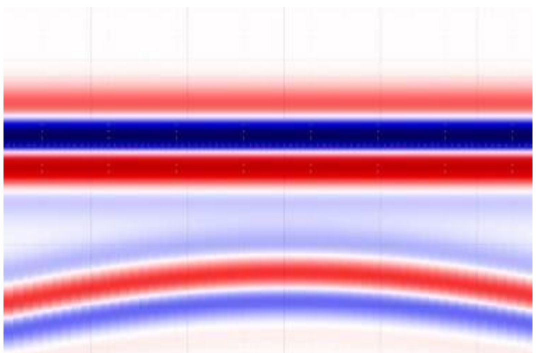
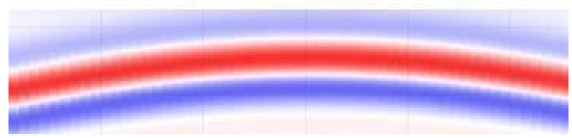
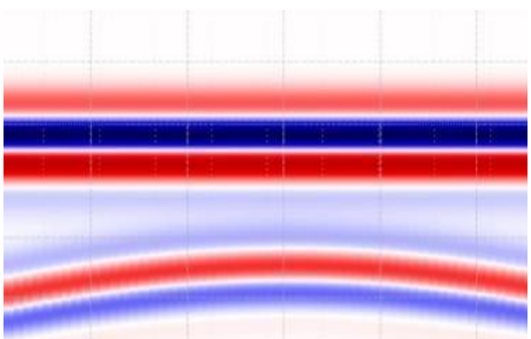
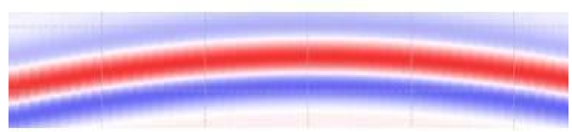
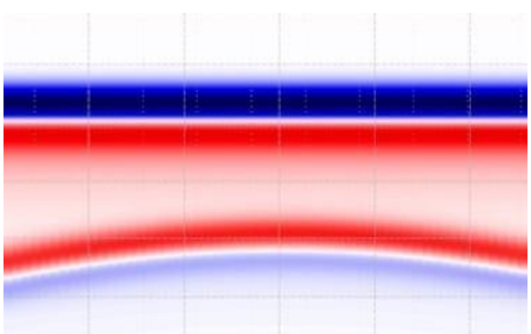
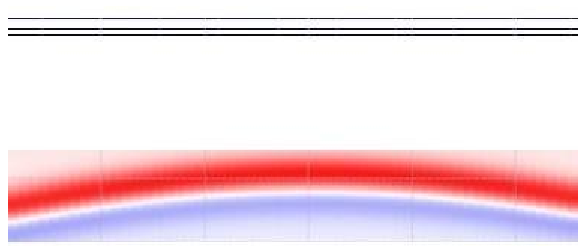
As we know that we will be provided with many B Scans so we need to do the automation of this process because we cannot do the procedure on every image individually. Automating makes everything faster, more reliable, and easier. Instead of doing the same tasks over and over again, computers can do it quickly and accurately. This helps ensure that the analysis is consistent and doesn't miss any important details. Automated analysis also lets us handle huge amounts of data easily, and it can even help us spot problems in real-time, like issues with structures or changes in the ground. Overall, automation is like having a super-efficient assistant that helps us get the most out of GPR data without the hassle.

For automating the process of extracting hyperbolas and removing background noise from a large number of B scan images we can approach the method of Batch Processing with scripting. In this approach first we make a directory in our machine in which all the B scan images will be present in it and then we will supply these images to our code which will apply the standard deviation method to each image and save the output processed images in another directory, and this process can be done using OpenCV, Numpy and Matplotlib.

6.1 Source Code for this procedure:

```
import cv2
import numpy as np
import os
from tqdm import tqdm
# Define the directories
input_directory = '/Users/CP301/Desktop/INPUT' output_directory =
'/Users/CP301/Desktop/OUTPUTS' if not
os.path.exists(output_directory):
os.makedirs(output_directory)
# Process each image in the directory
for filename in tqdm(os.listdir(input_directory)):
if filename.lower().endswith(('.png', '.jpg', '.jpeg')): image_path =
os.path.join(input_directory, filename)
image = cv2.imread(image_path)
# Convert the image to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# Calculate the standard deviation for each row
std_dev = np.std(gray, axis=1)
threshold = np.mean(std_dev)
mask = std_dev > threshold
# Invert the mask for a white background inverted_mask = ~mask
# Apply the inverted mask to a white canvas
white_background = np.ones_like(image) * 255
white_masked_image = white_background * inverted_mask[:, np.newaxis,
np.newaxis]
# Where the original mask is true, keep the original image
white_masked_image[mask] = image[mask]
# Save the processed image with a white mask
output_path = os.path.join(output_directory,
filename)cv2.imwrite(output_path, white_masked_image)
```

Result Obtained:

INPUT DIRECTORY IMAGES	OUTPUT DIRECTORY MODIFIED
	
	
	
	

7. Further Processing of GPR B Scans:

Data display is of uttermost importance as it is the key to data interpretation. In GPR systems, the display of information can be done in three different ways: A-scans and B-scans.

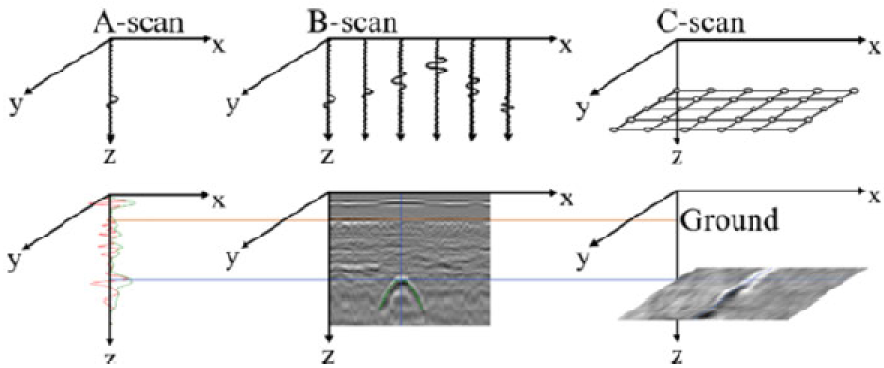


Figure: Comparison between A-scan, B-scan and C-scan

During GPR surveys, multiple sets of traces are taken. When the survey is a simple line, the traces can be aggregated in a position-time graph known as a B-scan. On such graphs,
The signals are highlighted based on their amplitude, using colors or brightness modulation.

Characteristics of B Scans:

- If the reflected object has a symmetric shape and its dimensions are bounded, then a hyperbolic shape will appear on the graph, whose apex corresponds to the real depth of the object.
- Furthermore, the aperture of said hyperbola depends on the object's dimensions, e.g. a spherical object with diameter D1 shows a bigger hyperbola aperture than a spherical object with diameter D2 if D1 > D2.
- Bright spots: Represent areas where electromagnetic waves have encountered high-contrast interfaces or changes in material properties. This could indicate the presence of subsurface objects, such as buried utilities, voids, or geological features.
- The B-scans of perpendicularly scanned cylindrical objects resulted in hyperbolic shapes but oblique survey lines resulted in wider, flatter and asymmetrical hyperbolas.

The existence of an automatic systems capable of discriminating hyperbola patterns from the background samples when real time applications are envisaged would ease the data treatment and interpretation tasks of geologists.

The used techniques can coarsely be grouped into two categories: fitting methods and classification-based methods.
The fitting methods, like the Hough transform, are highly computationally demanding approaches due to the high resolution of B-scan images. In the presence of noise and multiple disturbance sources, they produce random results. The classification category of methods adopts the idea of target discrimination, through a classification stage, before applying a fitting technique to build signature skeletons from edge points.

7.1 Fitting Method-Image Processing Algorithm Approach:

The general purpose is to segment the hyperbolas by eliminating background noise and ground plane.The proposed system includes **preprocessing, segmentation, and object detection stages**.

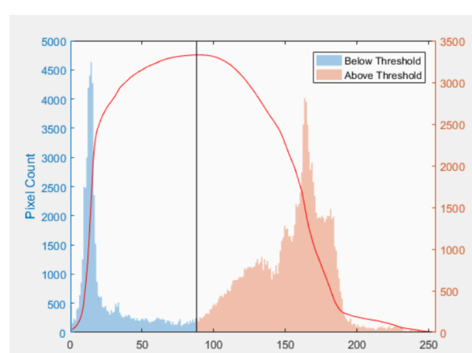
- The preprocessing process for GPR images requires separating different RGB color spaces to segment the ground plane and the location of the buried objects. For this reason, the red channel is selected in RGB space and soil and objects are segmented by using the **Otsu threshold method**.
- Soil planes are extracted by **Hough transformation**. Using Hough Transform **Line detection** is done. The soil planes in the form of thick lines and also the color map legend at the right side of the initial image is detected.
- We now color the detected region white and in this way we are able to **Isolate the Hyperbolas** of the B Scan.
- Estimated buried object region is what we are left with now.

OTSU THRESHOLD:
It performs automatic image thresholding, the algorithm returns a single intensity threshold that separates pixels into two classes, foreground and background. This threshold is determined by maximizing inter-class variance.

The pixels are divided into two classes as C0 and C1 at gray level t. Gray-level probability distributions are calculated for C0 and C1.

$$\omega_0(t) = \sum_{i=0}^{t-1} p(i)$$

$$\omega_1(t) = \sum_{i=t}^{L-1} p(i)$$



The means of C0 and C1 classes are computed as:

$$\mu_0(t) = \frac{\sum_{i=0}^{t-1} ip(i)}{\omega_0(t)}$$

$$\mu_1(t) = \frac{\sum_{i=t}^{L-1} ip(i)}{\omega_1(t)}$$

The total mean of C0 and C1 classes is:

$$\mu_T = \omega_0\mu_0 + \omega_1\mu_1$$

The between-class variance is formulated as:

$$\sigma_B^2 = (\omega_0)(\mu_0 - \mu_T)^2 + (\omega_1)(\mu_1 - \mu_T)^2$$

$$= \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2$$

7.2 Hough Transform:

Hough Transform is characterized by the robustness against noise and partial occlusion of features. It has many formulations, the simplest being Hough Lines (line detection).
To tackle the fact that there is a infinite number of different lines, defined by slope (m) and y-intercept (b), that pass through a single point, Hough Lines maps the coordinates (x, y) of every pixel of a binary image from the Cartesian plane to a new plane (ρ, θ)

$$\boldsymbol{\rho = x \cos\theta + y \sin\theta}$$

In the new plane, lines are represented by their distance to the origin, ρi, and by the slope of their normal, theta θi. Each point of the Cartesian plane is represented by a sinusoidal curve in the (ρ, θ) plane.

7.3 Matlab implementation of the proposed method:

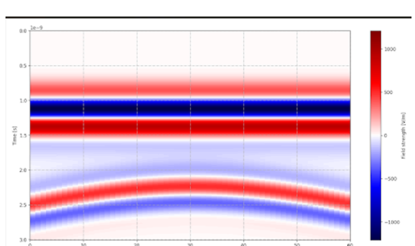


Figure Initial Image

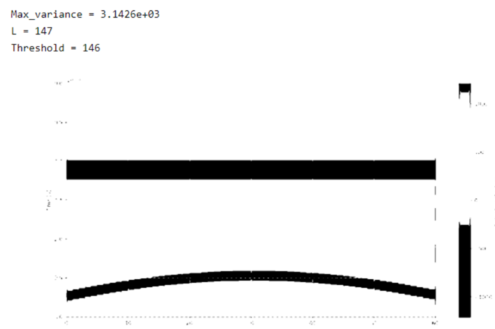


Figure Otsu Thresholding of Red Channel

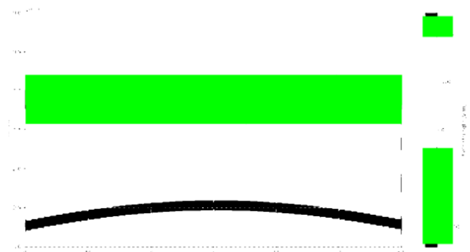


Figure Line detection Using Hough Transform

Figure Required Image with only Hyperbola

‘

Intensities corresponding to the above pixel in original B Scan:

Pixel_Row	Pixel_Column
66	70
67	70
68	70
69	70
70	70
71	70
72	70
73	70
74	70
129	70
130	70
131	70
132	70
133	70
129	71

```
255
255
251
195
138
 67
  0
  0
  0
203
193
190
185
182
203
192
190
185
```

7.5 MATLAB CODE for the above output:

OSTU Thresholding

Red channel of GPR images is the most appropriate platform for the image segmentation.

```
% Read the Bscan image
originalImage = imread('image.png');
resizedImage = imresize(originalImage, [200, 300]);

% Extract only the red channel as we need to segment the ground plane and the location of the buried objects
redChannel = resizedImage(:, :, 1);
%mage is represented as a 3-dimensional array row-column-color channels (Red, Green, and Blue).
% By indexing the array with (:, :, 1), we are extracting only the first channel,
% which corresponds to the red channel.

% Convert the red channel to grayscale
Image = im2gray(redChannel);

H=imhist(Image);
Max_variance=0;
Sigma=zeros(1,256);
TP=sum(H);

P=H/TP;

%run loop
for k=1:250
    %W0 and W1 are the class probabilities
    W0=sum(P(1:k));
    W1=sum(P(k+1:256));
    %U0 and U1 are class means
    U0=dot(0:k-1,P(1:k))/W0;
    U1=dot(k:255,P(k+1:256))/W1;
    Sigma(k)=W0*W1*((U0-U1)^2);
end
```

```
Max_variance=max(Sigma)
L=find(Sigma==Max_variance)
Threshold= L-1 %we have got the required threshold value

% Apply thresholding and save the thresholded image
thresholdedImage = Image > Threshold;
imwrite(thresholdedImage, 'thresholded_image.png');

% Display the thresholded image
imshow(thresholdedImage);
```

HOUGH Transform for line detection

```
% Read the already thresholded B-scan image
I = imread('thresholded_image.png');

% Apply edge detection (you may need to adjust the parameters based on your image characteristics)
BW = edge(I, 'Canny', [0.1 0.3], 2);

% Perform Hough Transform
[H, theta, rho] = hough(BW);

% Find Peaks in Hough Space
P = houghpeaks(H, 10, 'threshold', ceil(0.3 * max(H(:))));

% Extract Lines
lines = houghlines(BW, theta, rho, P, 'FillGap', 5, 'MinLength', 2);
```

```
% Visualize Detected Lines on the original image
figure, imshow(I), hold on
for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:, 1), xy(:, 2), 'LineWidth', 15, 'Color', 'white');
    %setting high linewidth as we need to detect thick surface planes

end

% Save the image with detected lines as a separate file
saveas(gcf, 'image_with_detected_lines.png');
```

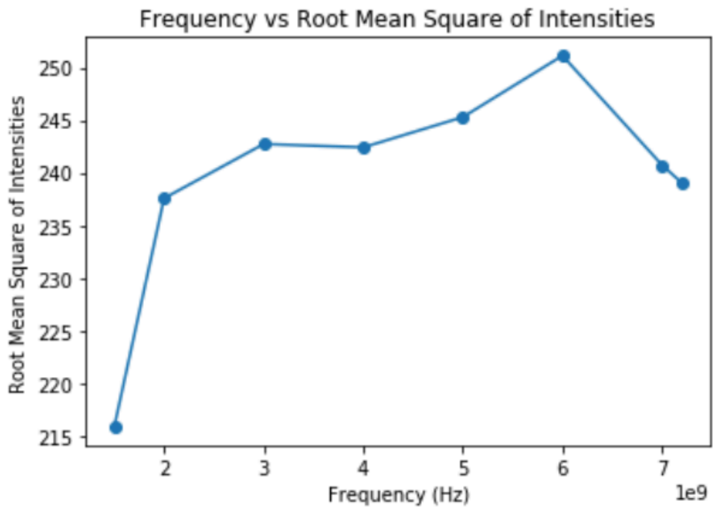
```
% after having the final image
% we now want to find the pixels with black color, or intensity 0
% Load the binary image
image_ = imread('image_with_detected_lines.png');
image = imresize(image_, [200, 300]);
gray_image = rgb2gray(image);
bw_image = imbinarize(gray_image);

% Find black pixels
[rows, cols] = find(bw_image == 0);

% Create a table to tabulate the coordinates
pixel_table = table(rows, cols, 'VariableNames', {'Pixel_Row', 'Pixel_Column'});

% Display the table
disp(pixel_table);
%4650 pixels containing hyperbola
```

```
% Get intensity values for each of the above pixel but for initial image
intensities = zeros(size(pixel_table, 1), 1);
for i = 1:size(pixel_table, 1)
    row = pixel_table.Pixel_Row(i);
    col = pixel_table.Pixel_Column(i);
    intensity = resizedImage(row, col);
    intensities(i) = intensity;
end

% Display intensities
disp(intensities);

imshow(resizedImage)
```

Using the above code to find the intensity variation by changing the GPR input parameters:

1. Frequency vs Root Mean Square of Intensities:

Frequency (Hz)	Root Mean Square of Intensities
1.5e9	215.8778
2e9	237.5721
3e9	242.7299
4e9	242.4095
5e9	245.2886
6e9	251.1254
7e9	240.7305
7.2e9	239.0225
8e9, 10e9, 15e9 not supported on GPRmax Software	

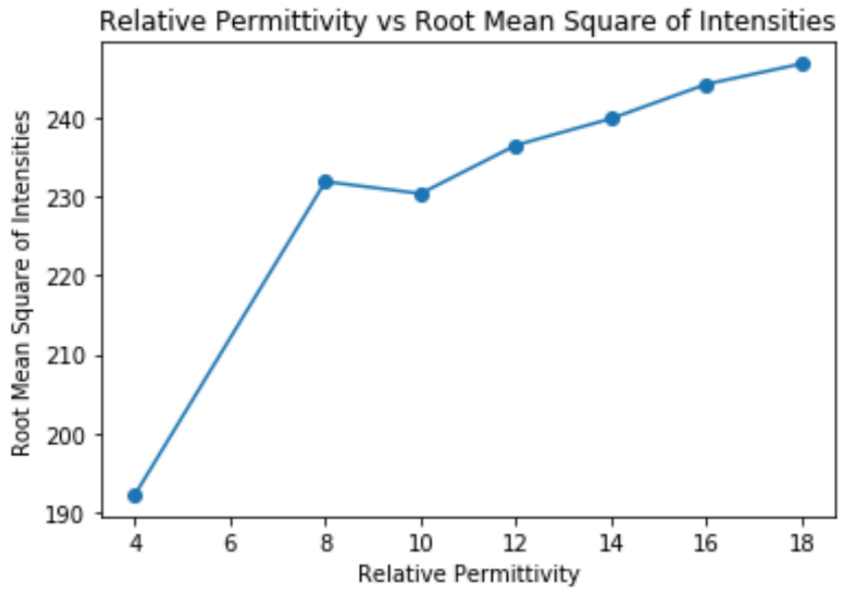


Physical Interpretation:

- Higher frequencies generally provide better resolution, allowing smaller objects or features to be detected.
- Consequently, the red band shows higher intensity in B-scans produced by higher-frequency GPR systems.
- Higher frequencies provide better resolution, allowing smaller features to be detected. This can result in more detailed images in the red channel.
- However, higher frequencies may result in reduced penetration depth as the electromagnetic waves are absorbed more rapidly by the subsurface materials.

2. Relative Permittivity vs Root Mean Square of Intensities:

Relative Permittivity	Root Mean Square of Intensities
4	192.2619
8	231.8939
10	230.3688
12	236.4967
14	239.8665
16	244.2191
18	246.8073



Physical Interpretation:

- Materials with higher relative permittivity slow down the propagation of electromagnetic waves.
- Materials with higher relative permittivity cause greater signal attenuation. Electromagnetic waves lose more energy as they pass through materials with higher relative permittivity, resulting in decreased signal strength at greater depths.
- Materials with higher relative permittivity values tend to reflect more electromagnetic energy. Consequently, the intensity of the red channel may be higher.
- Greater differences in relative permittivity between layers or materials can lead to more pronounced reflections, resulting in higher intensity in the red channel.

8. Modeling of Soil using Stochastic Dist. of Properties:

To model realistic soils in GprMax we need to use stochastic distribution of dielectric properties that involve simulating soil conditions that reflect the variability found in natural environments. This is achieved by incorporating stochastic, or random, distribution of dielectric properties within the GPRMAX simulation framework. Stochastic distribution refers to the statistical randomness or variability in the values of dielectric properties such as permittivity and conductivity across the simulated soil domain.By incorporating stochastic distribution of dielectric properties, GPRMAX simulations can better capture the complex interactions between electromagnetic waves and heterogeneous soils, leading to more realistic modeling results.

Principles of Modeling Soils in GprMax Using Stochastic Dist. of Dielectric Properties:

Use the `#soil_peplinski` command to define a series of dispersive materials to represent the soil. We can specify the sand fraction, clay fraction, bulk density, sand particle density, and a volumetric water fraction range. Create a Stochastic Distribution: The `#fractal_box` command is used to distribute these materials stochastically over a volume. You can define the volume, a fractal dimension, a number of materials, and a mixing model to use. The fractal dimension controls how the materials are stochastically distributed.

Wave Propagation in Heterogeneous Soils

For GPR applications, the soil can be modeled as a mixture of different materials each with distinct ϵ_r, μ_r, and σ. The propagation of electromagnetic waves is then influenced by these properties:

- **Wave Speed (v)**

$$v = \frac{c}{\sqrt{\epsilon_r \mu_r}}$$

  Where c is the speed of light in vacuum ($\approx 3 \times 10^8$ m/s).

- **Intrinsic Impedance (η)**

$$\eta = \sqrt{\frac{\mu_r \mu_0}{\epsilon_r \epsilon_0}}$$

- **Attenuation Constant (α)** due to conductivity:

$$\alpha = \frac{\sigma\sqrt{\mu_r \mu_0}}{2}\sqrt{\sqrt{1+\left(\frac{\epsilon_0 \epsilon_r \omega}{\sigma}\right)^2} - 1}$$

  Where ω is the angular frequency of the radar pulse.

CODE FOR GENERATING B-SCANS FROM HETEROGENOUS SOIL MODEL:

```
#title: Heterogeneous soil using a stochastic distribution of
dielectric properties given by a mixing model from Peplinski
#domain: 0.15 0.15 0.1
#dx_dy_dz: 0.001 0.001 0.001
#time_window: 6e-9
#waveform: ricker 1 1.5e9 my_ricker
#hertzian_dipole: y 0.015 0.075 0.085 my_ricker
#rx: 0.045 0.075 0.085
#src_steps: 0.001 0 0
#rx_steps: 0.001 0 0
#soil_peplinski: 0.5 0.5 2.0 2.66 0.001 0.25 my_soil
#fractal_box: 0 0 0 0.15 0.15 0.070 1.5 1 1 1 50 my_soil my_soil_box
#add_surface_roughness: 0 0 0.070 0.15 0.15 0.070 1.5 1 1 0.065
0.080 my_soil_box
#geometry_view: 0 0 0 0.15 0.15 0.1 0.001 0.001 0.001
heterogeneous_soil n
#cylinder: 0.075 0 0.035 0.075 0.150 0.035 0.010 pec
```

8.1 PARAVIEW MODEL OF THE SOIL:

We have made a VTK file which can be viewed in the Para view software.

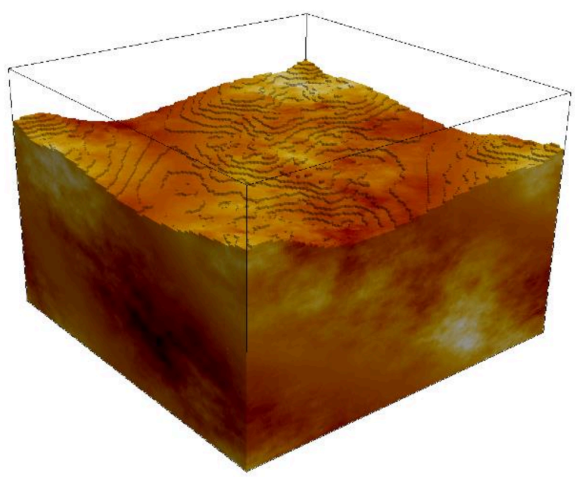


Figure 7.1 VTK file Paraview software.

B-SCANS FOR HETEROGENEOUS SOIL WITH CYLINDRICAL OBJECT BURIED:

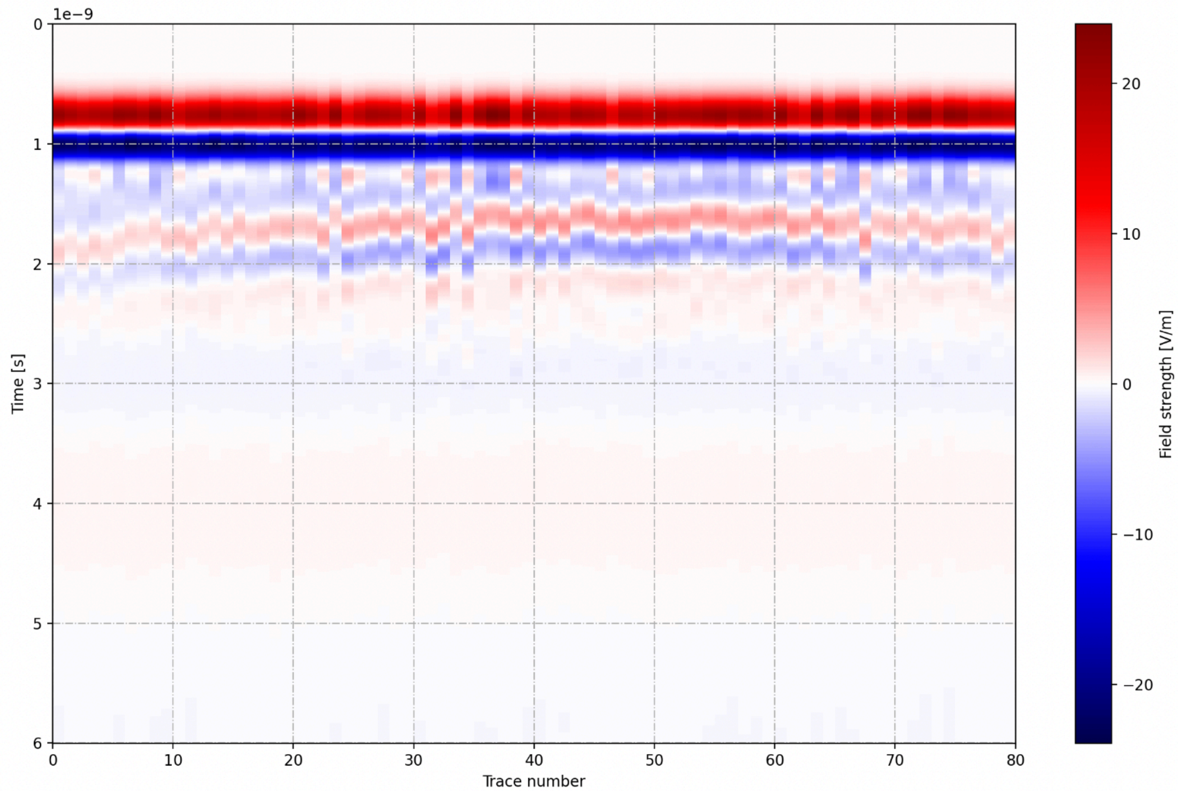


Figure 7.2 B-SCANS FOR HETEROGENEOUS SOIL

8.2 Modeling the Soil with Mines Present and Generating Respective B Scans:

8.2.1 Code for Procedure:

```
#title: heterogenous soil modeling with stochastic distribution of
dilelectric properties with PMN mines buried beneath
#domain: 1.75 0.4375 0.7875
#dx_xy_dz: 0.001 0.001 0.001
#time_window: 6e-9

#python:
from user_libs.antennas.GSSI import antenna_like_GSSI_1500
antenna_like_GSSI_1500(0.1 + current_model_run * 0.006, 0.126,
0.24,0.001)
```

```
#end_python:

#soil_peplenski: 0.5 0.5 2.0 2.66 0.001 0.25 my_soil
#fractal_box: 0 0 0 1.0 0.25 0.2 1.5 1 1 1 50 my_soil my_soil_box
#add_surface_roughness: 0 0 0.2 1.0 0.25 0.2 0.203 my_soil_box

#material: 3.2 0.02 1.0 0 bakelite
#material: 6.0 0.0 1.0 0 rubber
#material: 2.9 0.00050 1.0 9.75 TNT
#material: 2.5 0 1 0 plastic

     PMN
#cylinder: 0.3 0.126 0.15 0.3 0.126 0.147 0.056 rubber
#cylinder: 0.3 0.126 0.147 0.3 0.126 0.094 0.056 bakelite
#cyclinder: 0.3 0.126 0.147 0.3 0.126 0.097 0.053 TNT
#cyclinder: 0.3 0.126 0.147 0.3 0.126 0.097 0.002 pec

     PMN-1
#box: 0.65 0.09 0.12 0.79 0.16 0.15 plastic
#box: 0.67 0.10 0.13 0.77 0.15 0.14 TNT
#cyclinder: 0.67 0.12 0.135 0.77 0.12 0.135 0.002 pec
```

Output Model for Above Code:

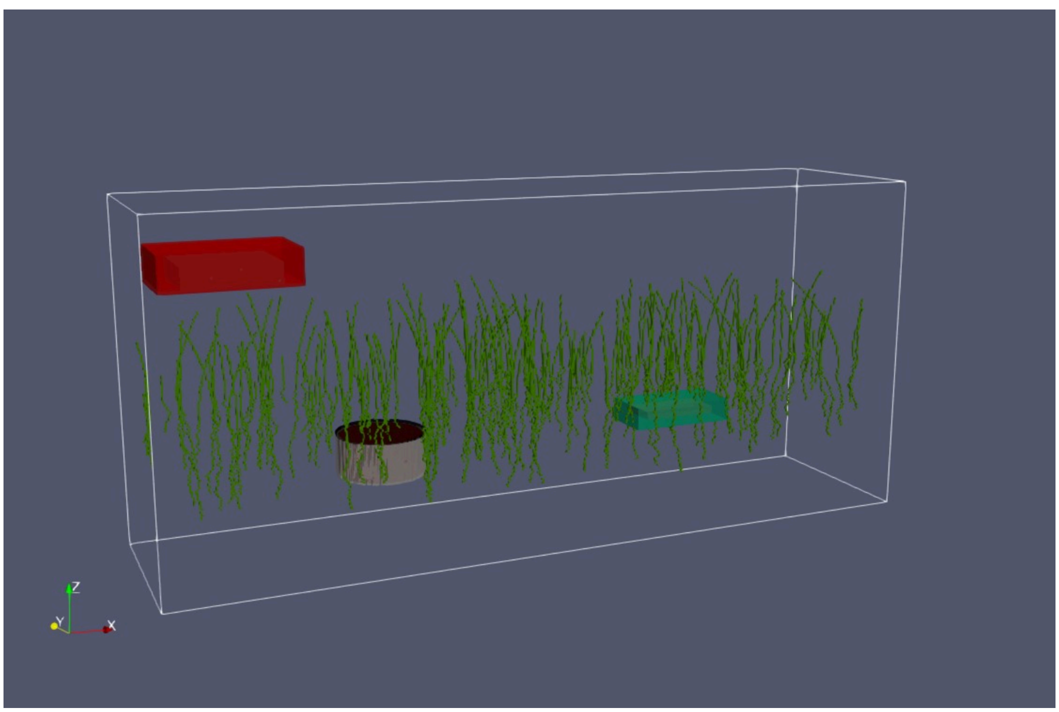


Figure: PMN and PMA-1 mines

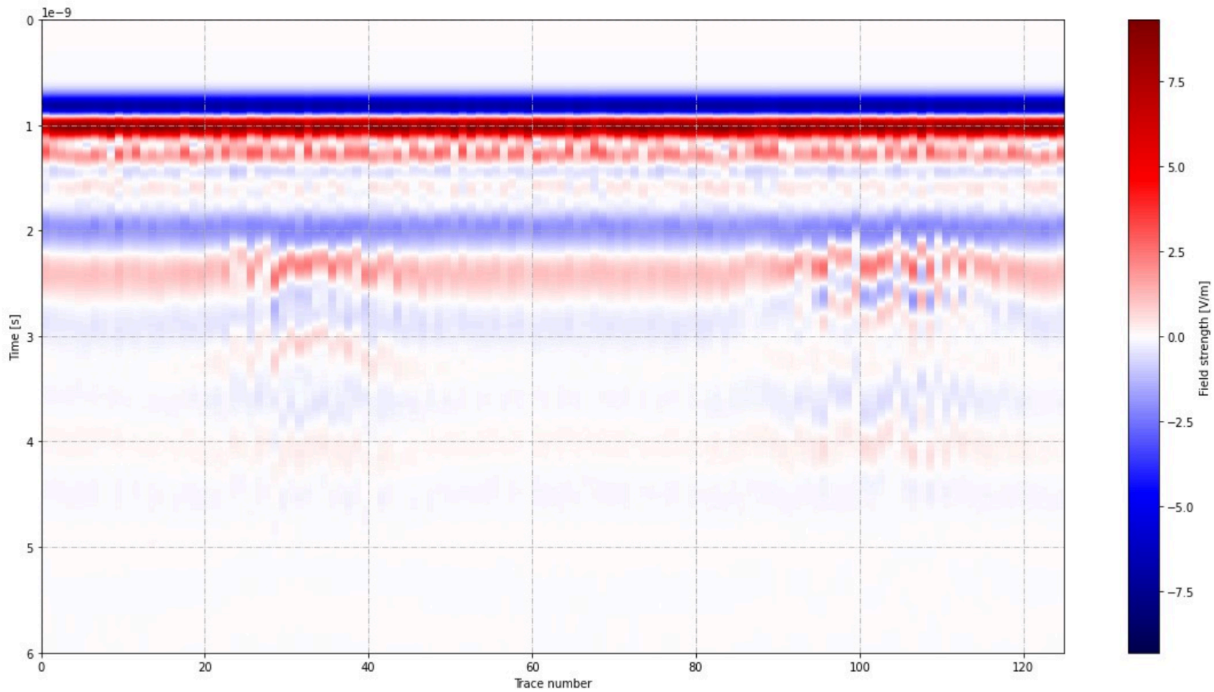


Figure: B-scan (using Δx = 1 mm)