University of Michigan

NERS 492

Senior Design

Radiation Detection Robot Final Report

Authors:
Undarmaa Ganbaatar
Emeline Hanna
Katie Olivas
Isaac Reichow

RADical Robotics

Mentors:
Dr. David Goodman
Dr. Zhong He
Dr. Brian Kitchen
Jim Mason

Instructors:
Thomas Jayasankar
Dr. Carolyn Kuranz
Dr. Stephanie Sheffield
Mackenzie Warwick

Submitted: April 21st, 2023

1 Introduction

The use and implementation of robotics in the nuclear industry has become increasingly important over the past few decades. Catastrophic power plant accidents such as Chernobyl and Fukushima demonstrated the need for radiation detection robots that can explore places that are dangerous for humans. Nuclear power plants need to be able to identify radiation hotspots during regular leakage detection and locate sources during an emergency response. Additionally, radiation detection robots have many applications in defense. For example, border control personnel could use radiation robots as a safer way to identify nuclear materials crossing the border and other nuclear threats. In an extreme case, these robots could be used in preparation for a nuclear conflict, on the battlefield, and to navigate in an area with fallout.

2 Motivation

Given the human health risks due to high levels of radiation, robots are needed in as a method to detect radiation. Radiation detection robots could allow human operators to make decisions based on sensor information or visual information from cameras, which greatly reduces the physical demand of occupational workers. Currently, for occupational workers to perform radiation surveys, they have to be physically fit, and perform tiring,extensive work. Additionally, if occupational workers are performing these tasks, they will not be able to efficiently survey some hard to reach places. Utilizing radiation detection robots allows for occupational workers to lower their potential exposure from surveys and accident cleanup, in addition to efficiently measuring a wider range of areas.

The flowchart in Figure 1 describes how our current methods of radiation detection and clean up in high radiation areas are dangerous to humans, demonstrating the need for robots.

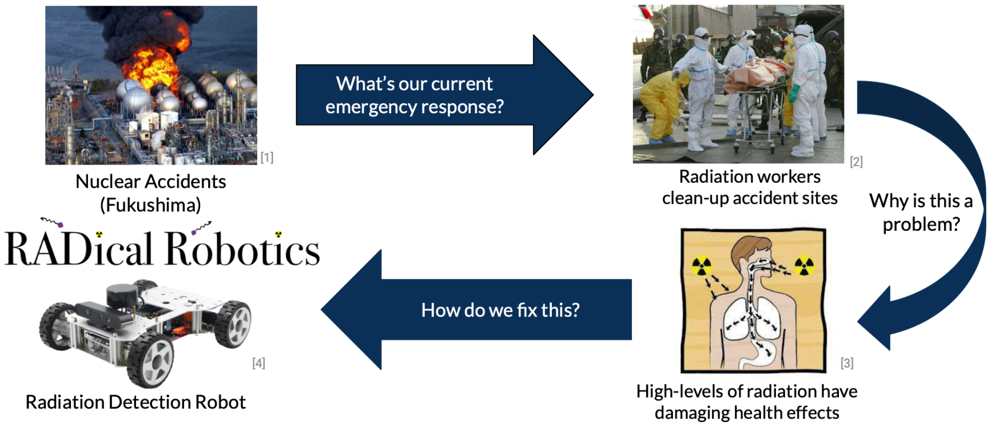


Figure 1: Radiation Detection Robot Project Motivation [1][2][3][4]

In response to this need, RADical Robotics is currently developing a radiation detection robot with the use of a Rosbot Robot Car [4] and a high resolution H3D M400 CZT gamma-ray spectrometer [5]. This has a Jetson TX1 computer and an on board depth camera and Light Detection and Ranging (LiDAR) sensor. We will communicate with the robot using the software of the Robot Operating System (ROS).

3 Background

There has been previous work in the area of using robots in radiation detection. The Lancaster University Neutron Laboratory used Gaussian process regression for radiation mapping of a nuclear reactor with a mobile robot. Using Simultaneous Localization and Mapping (SLAM), they generated a radiation map with a CeBr3 detector [6]. An example of the radiation map they created is shown in Figure 2.

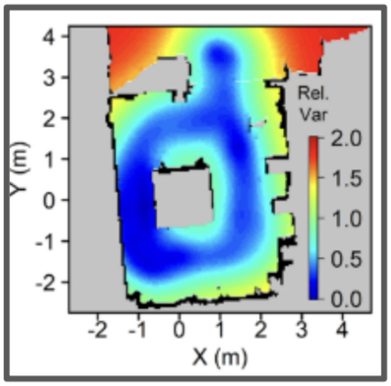


Figure 2: SLAM Generated Radiation Map using CeBr3 Detector [6]

This study was limited in the fact that a known ^{252}Cf source was used, so no isotope identificaiton was performed, and in the way that the CeBr3 detector was only responsive to gammas and not neutrons [6]. Furthermore, the State Key Lab of Fluid Power and Mechatronic Systems from Zhejiang University, China performed radioactive source recognition with moving Compton camera imaging robot using Geant4. The researchers were testing Compton reconstruction image quality based on robot angles and distances [7]. A figure describing how the angle of the robot changed in relation to the source is shown in Figure 3.

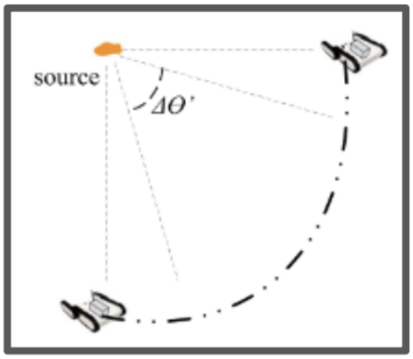


Figure 3: Compton Imaging Using a Robot [7]

The key takeaways were that image reconstruction precision was improved when the robot was closer to the source and when the robot moved around the source with a large angle distribution. Additionally, it was important to include positions with 90° angles [7].

4 Design Objectives

Using a provided robot and detector, our objective was to develop algorithms to successfully identify the location of and information about radioactive sources.

5 Methods

5.1 Experimental Methods Selection

The RADical Robotics team has selected and utilized several experimental methods to collect and analyze the data required to successfully achieve our objective. The methods were chosen based on several factors such as the availability of equipment and software, ease of use, reliability, and the extent of their capabilities. Python was chosen as the primary programming language for the project due to its free and open-source nature, which makes it easy to script algorithms and data structures. MATLAB was also selected for the project because of its easy data parsing and powerful plotting functions.

5.2 Compton Imaging

As RADical Robotics aims to localize the source, Compton imaging is being performed for one method of source localization.

5.2.1 Compton Imaging - Data Processing Workflow

The Compton imaging component is performed through post processing scripts, and outputs a radiation heat map, which can be displayed over the Rviz environment in a .png form.

As shown in Figure 4, the data processing begins with gathering information from the M400 CZT detector and the Rosbot. The detector provides source directional probabilities, from the point of view of the detector face, and the Rosbot provides the coordinates for which each source directional probability measurement was taken. The source directional probabilities and the Rosbot measurement coordinates are coupled together through a .bif file and fed into Python where a processing script is performed on the data to output the radiation heat map image.

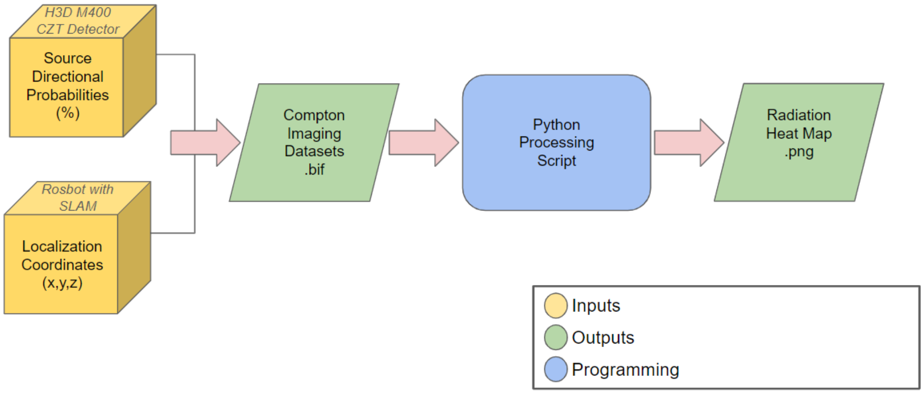


Figure 4: Compton Imaging Data Processing Workflow

5.2.2 Compton Imaging - Computational Work

It is important to note that the orientation of the robot needs to be known for proper Python script processing and image output. The left image below in Figure 5 displays the robots field of view, as the source directional probabilities are given in yaw degree bins. At a high level, the yaw angles essentially unfold from left to right, with 0° directly to the left of the detector face, 90° directly in front of the detector face, 180° directly to the right of the detector face, and 270° directly behind the detector face.

For preliminary testing, the detector source was placed in a known location, directly in front of the detector face. Based on the described orientation of the detector, this would indicate that the source was located at 90° for the experimental set up. As shown on the right image in Figure 5, there is a strong peak on the probability spectrum around 90°, indicating that the processing script correctly localized the direction of the known source location for this experiment.

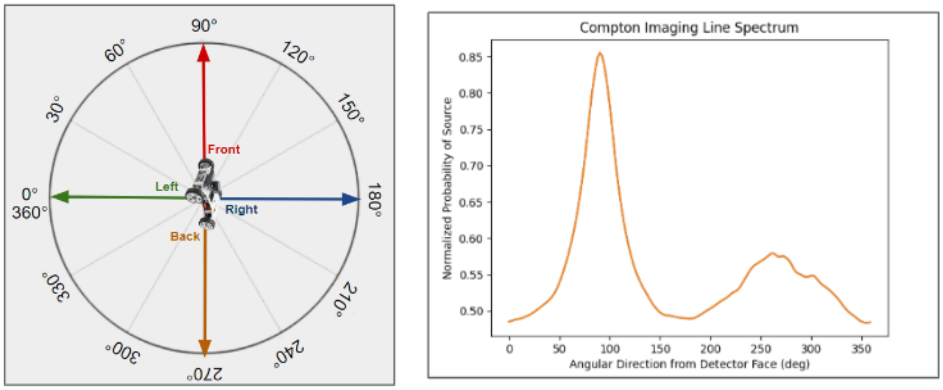
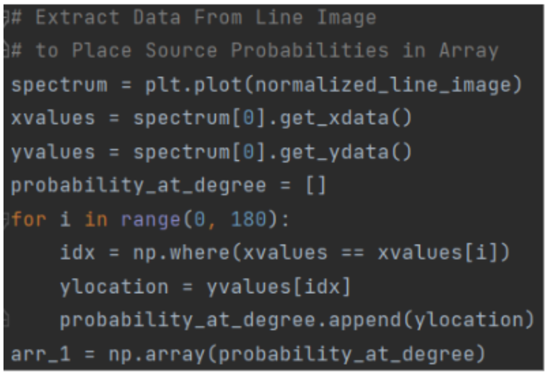


Figure 5: Compton Imaging Detector Orientation

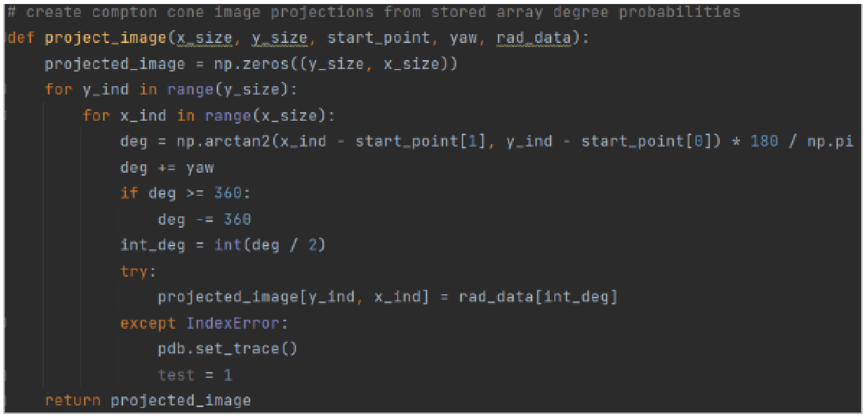
In order to correctly extract the source directional probabilities, an array had to be created to store the probabilities with a degree bin location. The Rosbot documents directional probabilities with 180 bins, meaning that each bin accounts for 2°, to cover the 360° detector range in 180 units. The process of developing and storing this array in Python is shown below in Figure 6.



```
# Extract Data From Line Image
# to Place Source Probabilities in Array
spectrum = plt.plot(normalized_line_image)
xvalues = spectrum[0].get_xdata()
yvalues = spectrum[0].get_ydata()
probability_at_degree = []
for i in range(0, 180):
    idx = np.where(xvalues == xvalues[i])
    ylocation = yvalues[idx]
    probability_at_degree.append(ylocation)
arr_1 = np.array(probability_at_degree)
```

Figure 6: Compton Imaging Code to Store Directional Probabilities

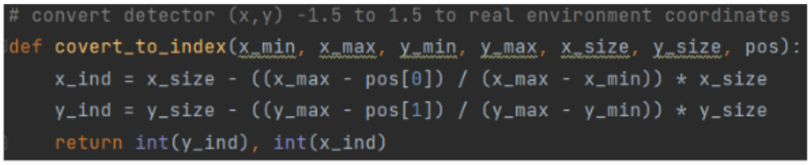
Once the source directional probabilities have been stored in an array, where each probability corresponds to a degree bin, this information can be used to develop a Compton imaging cone projection. The Python script shown below in Figure 7 outlines this process. The basic principle of developing the Compton cones involves considering the detector previously described, and projecting a cone by taking the tangent of the x and y indices in the stored array, and displaying them on a projected surface developed through a Python package plotting function with a jet map color scheme.



```
# create compton cone image projections from stored array degree probabilities
def project_image(x_size, y_size, start_point, yaw, rad_data):
    projected_image = np.zeros((y_size, x_size))
    for y_ind in range(y_size):
        for x_ind in range(x_size):
            deg = np.arctan2(x_ind - start_point[1], y_ind - start_point[0]) * 180 / np.pi
            deg += yaw
            if deg >= 360:
                deg -= 360
            int_deg = int(deg / 2)
            try:
                projected_image[y_ind, x_ind] = rad_data[int_deg]
            except IndexError:
                pdb.set_trace()
                test = 1
    return projected_image
```

Figure 7: Compton Imaging Python Script for Cone Projection

After the Compton cones have been developed, the map coordinates must be scaled, as the detector interface displays coordinate values from -1.5 units to +1.5 units for the x and y planes. For this experiment, the room was defined as a 200x200 unit space. The Python script shown below in Figure 8, displays the process of converting the detector coordinates to real environment coordinates. For further experimental testing, the coordinate system can be defined by any user input size.



```
# convert detector (x,y) -1.5 to 1.5 to real environment coordinates
def covert_to_index(x_min, x_max, y_min, y_max, x_size, y_size, pos):
    x_ind = x_size - ((x_max - pos[0]) / (x_max - x_min)) * x_size
    y_ind = y_size - ((y_max - pos[1]) / (y_max - y_min)) * y_size
    return int(y_ind), int(x_ind)
```

Figure 8: Compton Imaging Python Script for Real Environment Coordinate Development

When the Python script has processed the input bif file data, developed cone projections, and converted from detector coordinates to real environment coordinates, a projection image is output that displays the Compton imaging cones for the real environment. Thus far, an experiment has been performed with the source placed in a known location in the center of the defined environment, and measurements taken at 4 positions of 0°, 90°, 180°, and 270°. The Python script then overlays each of these Compton cones to display the defined source location as shown Figure 9.

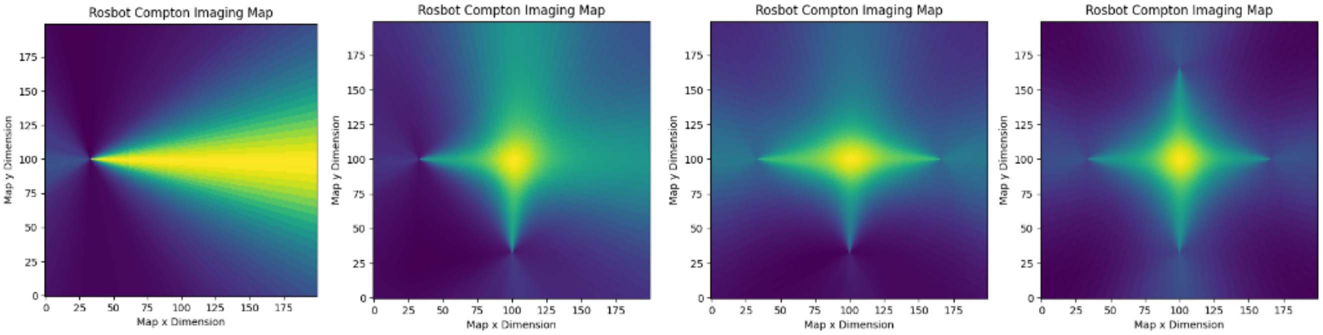


Figure 9: Compton Imaging Cone Projection for Real Environment Scale

Based on the experimental setup for preliminary trials, the Compton imaging processing script has shown to successfully localize the source. The source was placed in the center of the defined environment for this scenario, and the Compton imaging output successfully highlights the source in the correct location based on qualitative analysis.

5.3 Count Rate Imaging

As RADical Robotics aims to localize the source, count rate imaging is being performed for an additional method of source localization.

5.3.1 Count Rate Imaging - Data Processing Workflow

Count rate based imaging, also referred to as count rate based localization, utilizes inputs from the detector and the robot's internal simultaneous localization and mapping, or SLAM. The detector provides count data and SLAM provides positional information. This information is outputted from the robot's operating system into text files. These files are passed to a Matlab script which generates a count rate heat map and source location estimate through a least-squares algorithm. This workflow is visualized below in Figure 10.

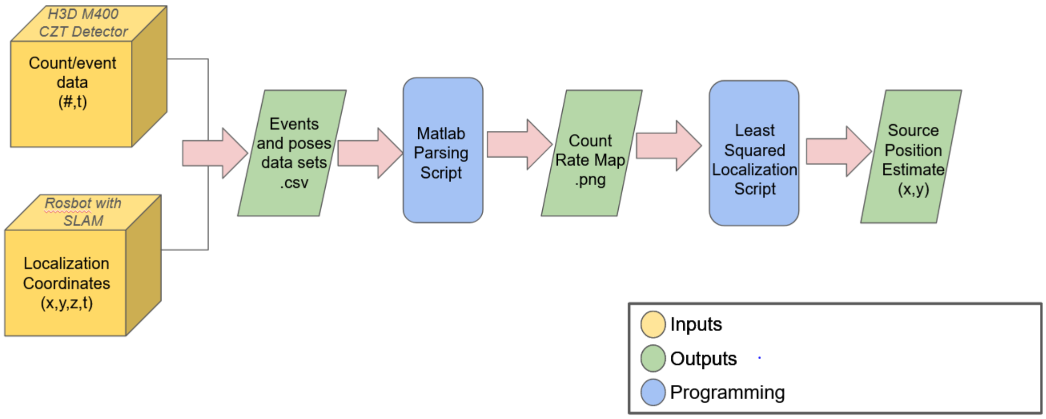

H3D M400 CZT Detector
Count/event data (#,t)
Rosbot with SLAM
Localization Coordinates (x,y,z,t)
Events and poses data sets .csv
Matlab Parsing Script
Count Rate Map .png
Least Squared Localization Script
Source Position Estimate (x,y)
Inputs
Outputs
Programming

Figure 10: Count Rate Imaging Data Processing Workflow

5.3.2 Count Rate Imaging - Computational Work: Least-Squares Iteration Scheme

The goal of count rate imaging is combine robot positional data with detector count data to create a map of source location likelihoods, and from that map find a global minimum which indicates a most probable x,y location for the radioactive source. Localization in this manner requires a least-squares regression algorithm. This method is described here.

For every x,y position around the robot's path for a range of source activities I, the following equation was applied:

$$S_{x,y,I} = \sum_{i=1}^{n} \frac{1}{e^2}(R_{rec} - \frac{\epsilon_{int}\epsilon_{emis} I A_{det}}{4\pi r^2})^2 \tag{1}$$

Where the summation is along every measurement point, e is measurement error, R_{rec} is the actual recorded count rate, ϵ_{int} is detector internal efficiency at a detected energy, ϵ_{emis} is the emission probability at a detected energy, I is source activity, A_{det} is detector area, and r is the distance between detector and the current x,y coordinate being assessed. This equation is measuring the difference between recorded count rate and a theoretical count rate. A point source approximation was used. Local minimum values of this equation, referred to as S-values, were recorded at every x,y coordinate assessed, and a single global minimum was recorded. The x,y location of this global minimum S-value was determined to be the location of the radioactive source.

This equation is physically accurate when its focused around a single photopeak energy. This is do to the nature of how full energy deposition works in a detector. At a single photopeak energy ϵ_{int}, ϵ_{emis}, and detector noise are all fixed values. Without prior knowledge of the identity of the source in question, all counts as opposed to just photopeak counts would need to be analyzed making the equation technically non-physical. Therefore, for count rate imaging, the only energies analyzed were in a 5 keV range around 662 keV, Cs-137 photopeak energy.

Heat maps were generated by plotting every local minimum S-value. Examples of what these maps look like are provided below in Figure 11. The left image is from a sample data set where only a single Cs-137 source was present, and the right image is from a sample data set where one Cs-137 source and two other arbitrary sources were present.

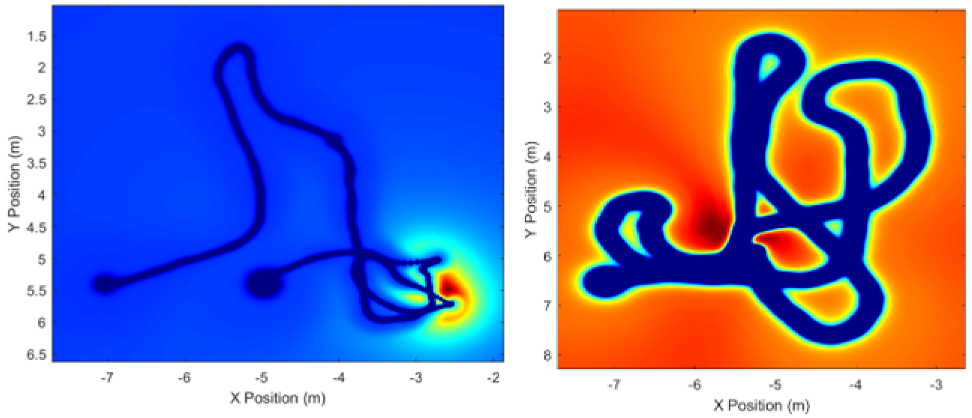


Figure 11: Least-Squares Sample Maps

5.4 Isotope Identification

As one aspect of characterizing the localized source or sources, RADical Robotics aims to determine the isotopes present based off of gamma energy spectra.

5.4.1 Isotope Identification - Data Processing Workflow

The isotope identification process has several steps. Firstly, a data file from the detector containing counts per energy channel information is used as an input to a Python processing script, which will output a plot of the energy spectrum and identified peaks and the identified isotopes. This workflow is described visually in Figure 12.

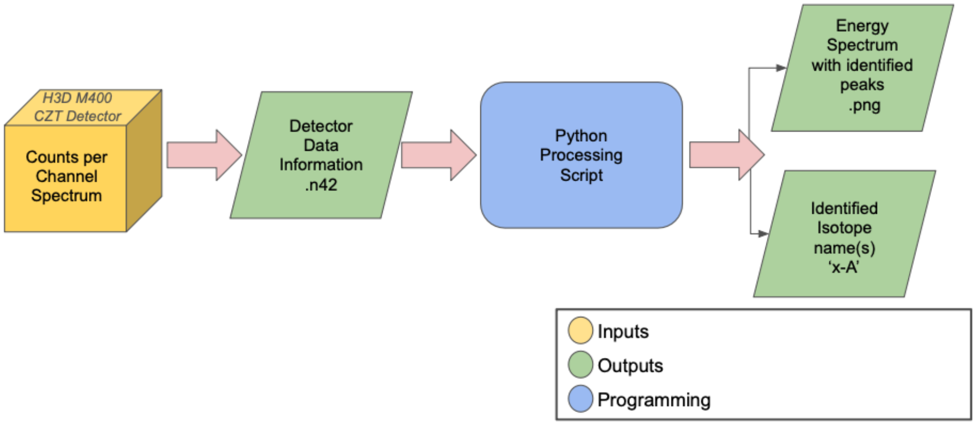

H3D M400 CZT Detector
Counts per Channel Spectrum
Detector Data Information .n42
Python Processing Script
Energy Spectrum with identified peaks .png
Identified Isotope name(s) 'x-A'
Inputs
Outputs
Programming

Figure 12: Workflow of Isotope Identification

5.4.2 Isotope Identification - Computational Work

Breaking down this process further, the Python processing script first plots the energy spectra using a Python package called Matplotlib. To identify isotopes in a gamma spectra, the energies where peaks occur must be found and matched to the known energies of gamma emissions for each isotope.

5.4.3 Peak Finding

To find the peaks on a spectra, the Python package SciPy is used. The peak finding algorithm requires many inputs that had to be tuned for this application. Three important input parameters for peak finding are width, prominence, and height. The minimum width of a peak depends on the electronic noise in the detector and depends linearly on the energy of the peak since charge production in the detector follows a Poisson distribution. The prominence of the peak describes how much higher the peak is compared to the continuum of the spectra. The minimum prominence for a peak to be recognized was set to be a fraction of the maximum number of counts in a channel, since the range of maximum number of counts in a channel varies widely between spectra. Thirdly, the minimum height of the spectra is the minimum counts of the spectra at each energy. This is set to be the smoothed spectra with an added uncertainty based on neighboring counts. As can be seen in Figure 13, height threshold suppresses noise (which is especially present in the right plot due to the low number of counts) and allows for peaks to be identified.

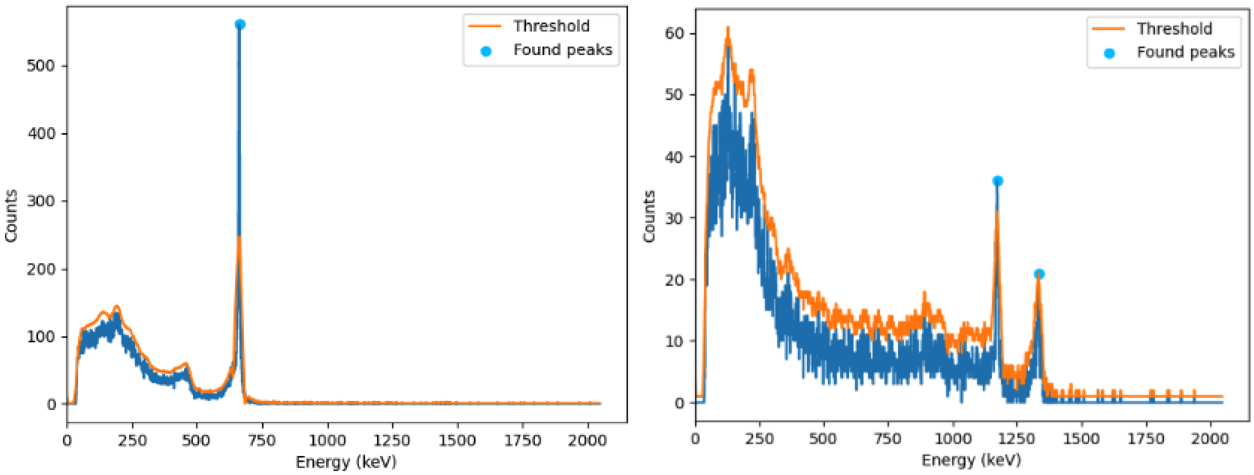


Figure 13: Two Spectra with a Height Threshold

As an important note, all of these parameters are calculated based off of the input spectra, and there is no correction needed between widely different spectra. In the above figure, note the difference in scale of the y-axis, the difference in energy of the peaks, and the height of the peaks compared to the maximum number of counts for any channel. Despite many differences between the spectra, the peaks were correctly identified in both cases.

5.4.4 Isotope Database

After the peaks are identified, they must be compared with a database. The database has been created by RADical Robotics to include radioactive isotopes with their atomic number, element symbol, mass number, gamma peaks, emission probability, and x-ray peaks. Currently, the database includes 72 isotopes, although this could be expanded as necessary.

The identified peaks are compared with the true peaks of each isotope in the database. If an identified peak matches to a true peak, the isotope with that peak gets an increase in score. The score is weighted based on how prominent the peak is in the gamma spectra compared to the emission probability. After iterating through all isotopes, the isotope(s) with a high enough score are given as the identified isotope(s).

An example spectra is shown in Figure 14 below. The blue dots show where the peak finding algorithm found a peak. Since the isotope was known, the vertical lines are the true peaks of the isotope present, which is Ba-133.

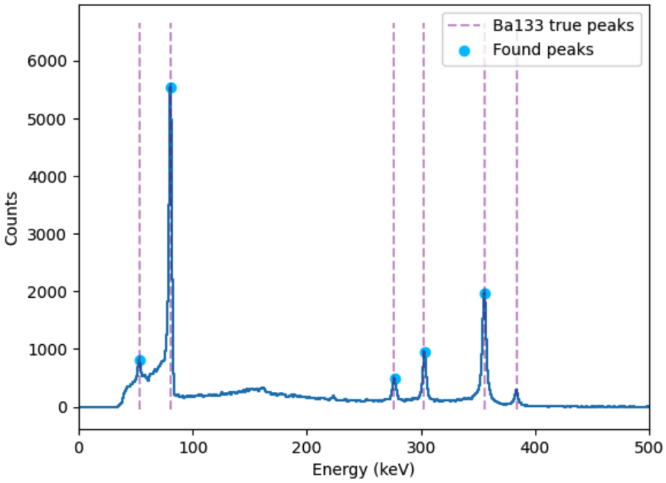


Figure 14: Example Ba-133 Spectra showing Identified Peaks

Note that even though all peaks were not identified, the most prominent ones were, and the spectra was identified as Ba-133.

5.5 Activity Analysis

RADical Robotics team aims to develop a methodology for determining the activity of a source based on data collected from multiple subsections of the project. The team has created a data processing workflow that integrates the output from the isotope identification, compton imaging, and count rate-based imaging subsections as shown in Figure 15.

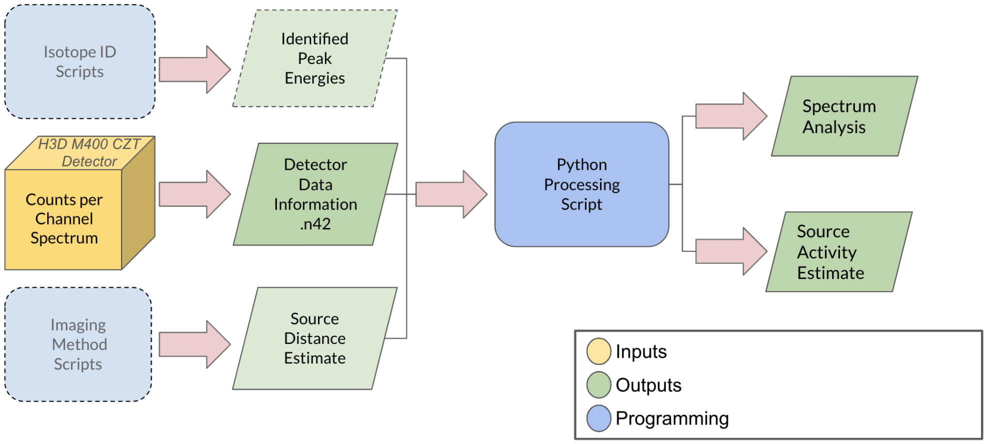


Figure 15: Workflow of Activity Analysis

The isotope identification method provides an output that determines the isotope ID and peak energies seen, while the two imaging methods estimate the source position. These outputs serve as inputs for the Python processing scripts that perform spectrum analysis and estimate the activity of the source.

The spectrum data structure maintains variables relating to the spectrum information, such as FWHM, peak energies, and background count. The team has also included member functions that can invoke the Isotope ID methodology and provide spectrum information to other sections of the project.

The activity analysis methodology depends on knowing the detector intrinsic peak efficiency. To calculate the detector peak efficiency, the team has worked with separate data sets than the ones used for activity analysis, with the ability to update the efficiency lookup by using more data as it becomes available. The efficiency calculation depends on the spectrum data structure and its member functions, as it invokes these methods during data processing. The activity of the source (given to the team) and the distance from the detector are used to calculate the intrinsic peak efficiency. The efficiency class maintains variables relating to the calculated efficiency at energy, which can be used as a lookup function or interpolated as more data becomes available. The formula used for determining the detector intrinsic peak efficiency is shown below:

$$\epsilon_{\text{ip}} = \frac{N_{\text{detected}}}{N_{\text{emitted}}} \cdot \frac{4\pi}{\Omega}$$

where:

$$\begin{aligned} N_{\text{detected}} &= \text{Radiation quanta detected from source} \\ N_{\text{emitted}} &= \text{Radiation quanta emitted from source} \\ \epsilon_{\text{ip}} &= \text{Intrinsic peak efficiency of the detector} \\ \Omega &= \text{Solid angle subtended by the detector} \end{aligned}$$

RADical Robotics has developed scripts to aid in the calculation of the detector efficiency and source activity. The total count under full energy peaks were determined by calculating the region of interest using the calculated FWHM times the appropriate multiplier and subtracting the background count. To demonstrate the peak area determination, the team has plotted a sample detector spectrum with a background subtracted spectrum, as shown in Figure 16. The background count was estimated by performing a min-max normalization on the original spectrum and subtracting it from the total count to obtain the peak count. The formula for background estimate (y_{val}) is shown below:

$$x_{\text{mixing}} = \frac{x_{\text{val}} - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

$$y_{\text{val}} = (1 - x_{\text{mixing}})y_{\text{min}} + x_{\text{mixing}}y_{\text{max}}$$

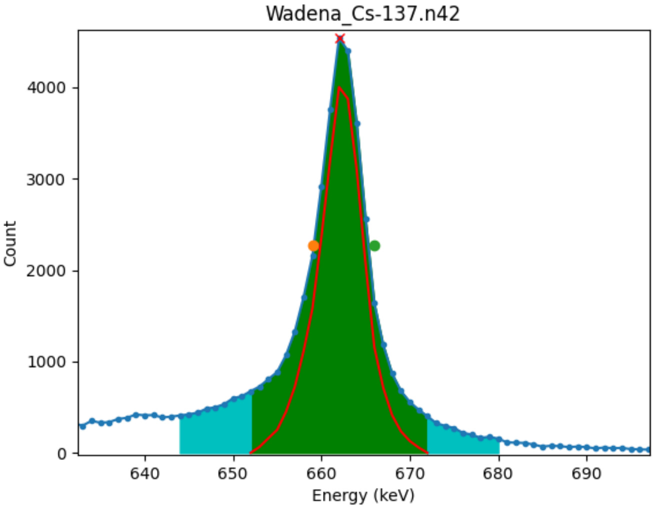


Figure 16: An example spectrum for Cs-137 is shown, with the peak and FWHM identified. Superimposed spectrum shows background subtracted peak

Once the detector peak efficiency and source distance has been estimated, the activity of the source can be calculated using the source count emitted formula:

$$\mathrm{S} = \mathrm{N}\frac{4\pi}{\epsilon_{\mathrm{ip}}\Omega}$$

where:

$$\begin{aligned} \mathrm{S} &= \text{Radiation quanta emitted from source} \\ \mathrm{N} &= \text{recorded events under full energy peak} \\ \epsilon_{\mathrm{ip}} &= \text{Intrinsic peak efficiency of the detector} \\ \Omega &= \text{Solid angle subtended by the detector} \end{aligned}$$

As previously noted, to calculate the solid angle, we can use two imaging methods to estimate the position of the source and its distance from the detector. However, since the detector efficiency depends on the direction in which it is facing, additional measurements and analysis are needed to better account for this. The Compton imaging method will help to determine the direction of the source, which is critical for accurately determining the activity estimate. Comparing the two imaging methods can determine which one provides more accurate results.

Currently, the detector peak efficiency has been calculated using data obtained from the H3D S100 detector. The plotted efficiency vs energy graph can be seen from Figure 17. The calculated efficiency had a mean relative difference of 6.5% compared to the reference efficiency provided by our mentors.

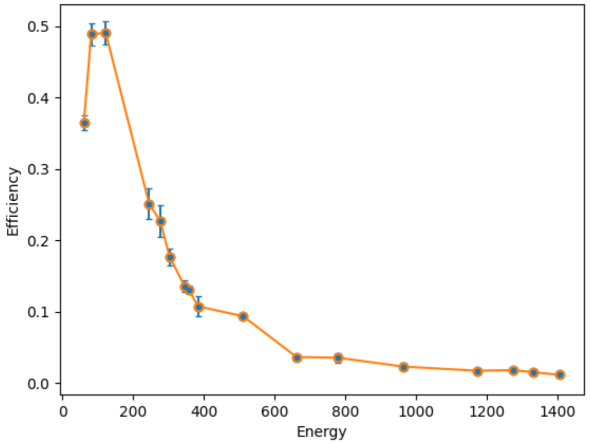


Figure 17: The detector efficiency graph for S100 detector

6 Results

6.1 Compton Imaging Results

6.1.1 Real World Dimensions

In order for the source to be localized within the real world environment,the map pixel values were converted into meters. This enabled the projected image to display the source environment on a map that expressed its size in meters, which enhances a potential users understanding of the map. As shown in Figure 18 below, the initial map iteration displays itself in a 496 x 353 pixelated format.

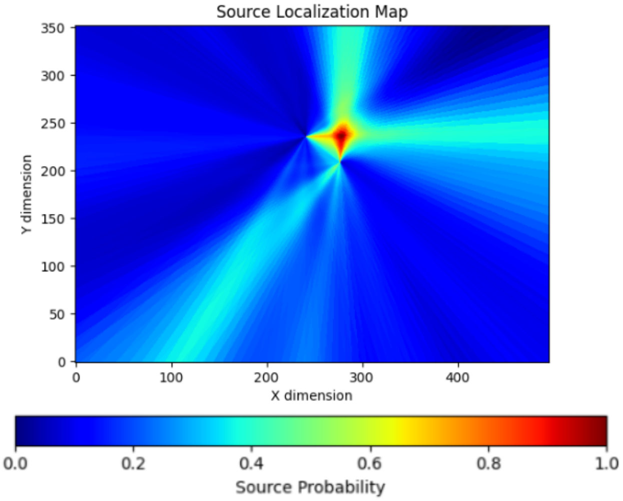


Figure 18: Source Localization Map in Pixelated Dimensions

In order to display the image in a way that made sense to a potential user, the map dimensions were converted to meters utilizing the fact that each pixel represents 0.05 meters of the true mapped environment. As shown in the figure below, applying the correction factor of $0.05\frac{meters}{pixel}$ generates a map that is representative of a true unit of measurement that users would be familiar with. As shown in Figure 19 below, the new iteration of the map displaying dimensions of 24.8 m x 17.65 m.

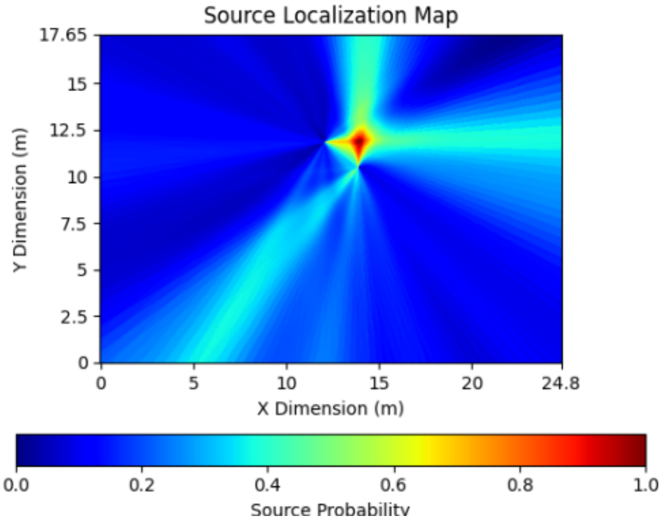


Figure 19: Source Localization Map in Real World Dimensions

Once the map dimensions have been converted into real world dimensions, a weighted average approach was applied in order to properly localize the source. As shown in Figures 18 and 19, the map displayed is essentially a probability density grid, with the hot spot which indicates the source location holding the highest probability value.

6.1.2 Real World Source Localization

Once the real world map dimensions were established, an analysis of proper source localization was performed. The figure below displays a gray scale source localization map for a 17.65 m x 24.8 m real world environment. The gray scale map was utilized to determine a weighted average location for the map. When compared to Figure 19, it is clear that the white spot in Figure 20 relatively aligns with the red, high probability regions in Figure 19. In order to eliminate any potential biases from back scatter projections, as shown in the lighter blue regions in Figure 19, a normalized and threshold gray scale map was developed. As shown in Figure 20 below, the probability density grid was normalized to the highest probability (meaning the highest probability was assigned a grey scale value of 255), and any matrix location that did not contain a probability within 75% of the maximum grey scale value of 255 was minimized to a grey scale value of zero.

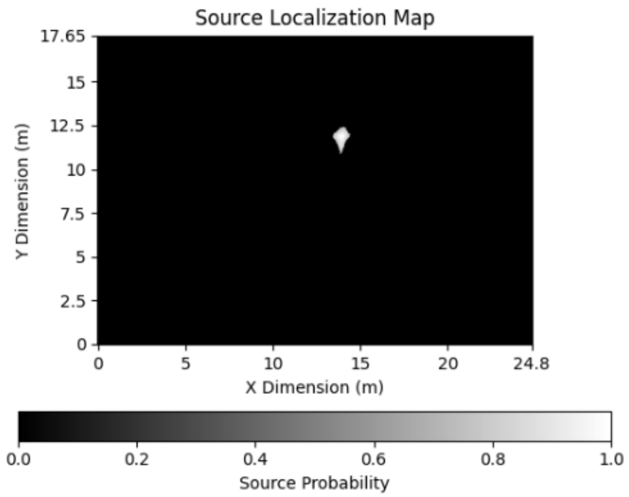


Figure 20: Weighted Average Source Localization

Once a normalized and pixel value discriminated map has been output, a weighted average approach is utilized to determine the location of the source. As the map has been discriminated to contain only values of 255 (white hot spot) and zero (black surrounding areas), the weighted average of the map displays the center of the white region, providing a pixel location of the source. In this example, the source was located at pixel 278.27 in the x direction and pixel 233.82 in the y direction, which corresponds to 13.91 meters in the x direction and 11.69 meters in the y direction using the conversion of 0.05 meters per pixel.

In order to confirm the accuracy of this projection, the source localization map was overlaid onto the cartographer environment map developed by the Rosbot's built in cartographer platform. This is depicted in Figure 21 below.

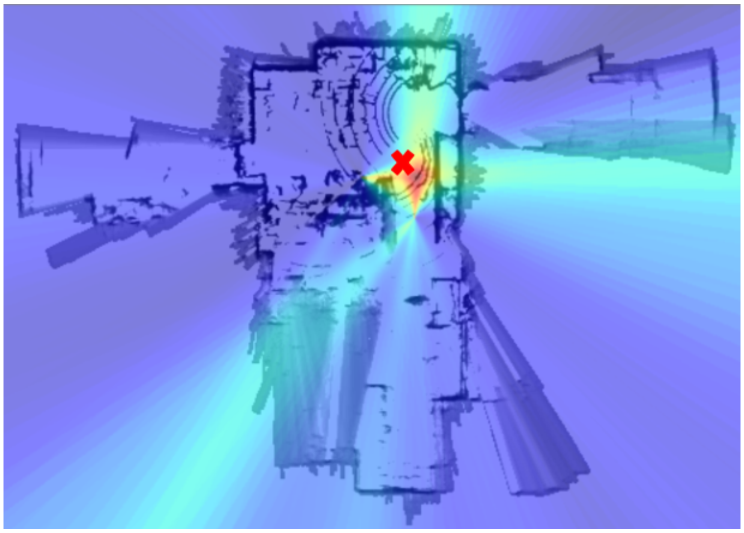


Figure 21: Source Localization in Orion Lab (True Source Location Marked with Red X)

From a quick look at the figure above, the Compton imaging approach was able to localize the source relatively close to the true source location. This is evident as the red X (true source location) is very close to the estimated source location (red circular hot spot). Upon further analysis, it was determined that the estimated source location was 0.21 m ± 0.05 m from the true source location. The uncertainty of this measurement is limited by the cartographer pixel resolution, which is currently programmed at 0.05 m per pixel. This result supported the hypothesis that Compton imaging is an effective method for source localization.

6.2 Count Rate Imaging Results

6.2.1 Real-World Test of Count Rate Imaging

In the Orion Measurements Laboratory, a single Cs-137 button source was placed in the middle of the floor, and a continuous measurement of incident radiation events and robot position was taken. A least-squares map was generated then overlayed on top of the robot's generated LiDAR map. This is shown below in Figure 22.

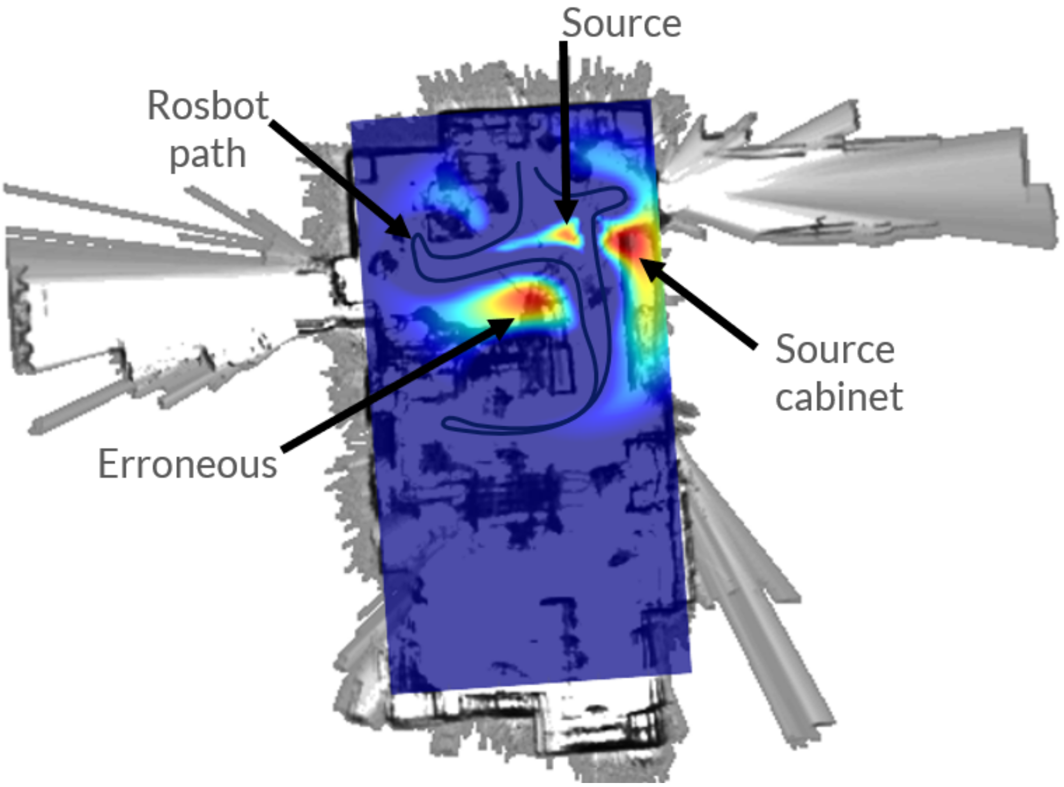


Figure 22: Source Localization in Orion Lab Using Count Rate Imaging

Three distinct hot spots can be seen here and are labeled. The hot spot labeled "source" was the actual position of the Cs-137 source, however the algorithm seemed to prefer the location labeled "source cabinet". The source cabinet is the exact location in the lab where all radioactive sources are stored. Although the sources were behind lead shielding, this is the best guess as to why the algorithm prefers this location. The third hot spot is labeled "Erroneous". It is not exactly known why this location is shown as being a likely source location. Most likely it is due to a lack of data points, however there may have been an unknown radioactive source at that location

Overall, the least-squares regression identified the source cabinet as the approximate source location. Therefore, the estimated position of the source was $1.8m \pm 0.9m$ away from the true source location. Due to this test as well as information gathered from sample data, it was determined that this method of count rate imaging is ineffective in localizing sources when multiple sources are present. Additionally, the method requires photopeak energies be known before the test begins, making its application very limited unless paired with an isotope identification script. This method of count rate imaging can, however, provide insight into the location of generally radioactive areas as opposed to an accurate x,y location.

6.3 Isotope Identification Results

6.3.1 Background Dataset

To determine the false positive rate, the isotope identification algorithm was tested on background spectra. To make a large dataset, 120 second background spectra were bootstrapped from an approximately 14 hour background measurement. Bootstrapping is a method of random sampling the long measurement with the replacement, creating spectra with random statistical noise.

The bootstrapping method lead to 3002 background spectra. Two background spectra are shown in Figure 23.

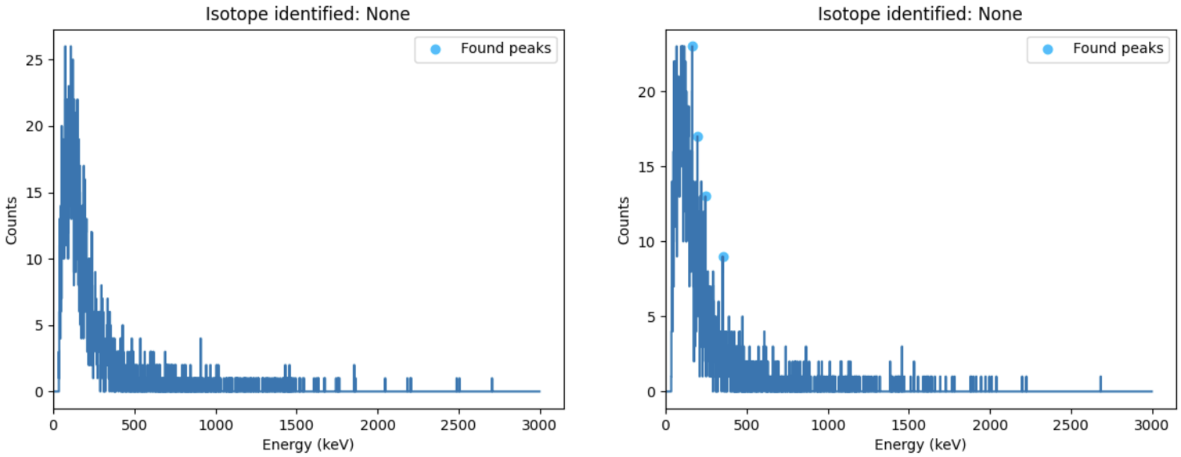


Figure 23: Background Spectra created using Bootstrapping

Note that in both background spectra, no isotopes are identified, as expected. Initially, the isotope algorithm would identify isotopes in background spectra at a higher rate than desirable, but the algorithm was weighted more heavily to value the prominence of peaks. In background spectra, the noise peaks are less prominent than peaks from isotope emissions.

The isotope identification algorithm correctly identified 2990 of these spectra. This means that 99.6% of the spectra accurately identified only background isotopes or no isotopes and 0.4% of the spectra had a false positive result. This meets the IAEA American National Standard Performance Criteria for Handheld Instruments for the Detection and Identification of Radionuclides (N42.34) background false alarm rate requirements [11].

6.3.2 Isotope Dataset

To determine the false negative rate and true positive rate, the isotope identification algorithm was tested on isotope spectra. The testing spectra included 13 unique isotopes and 3 combinations, resulting in 16 types of spectra. Each spectra type has 10 measurements, resulting in a total of 160 testing spectra.

Initially, the isotope algorithm would struggle on identifying isotopes that produced a lot of x-ray peaks. This flaw was improved by including x-ray peaks into the isotope dataset, as mentioned above. Two spectra including isotopes with tricky x-ray peaks are shown in Figure 24.

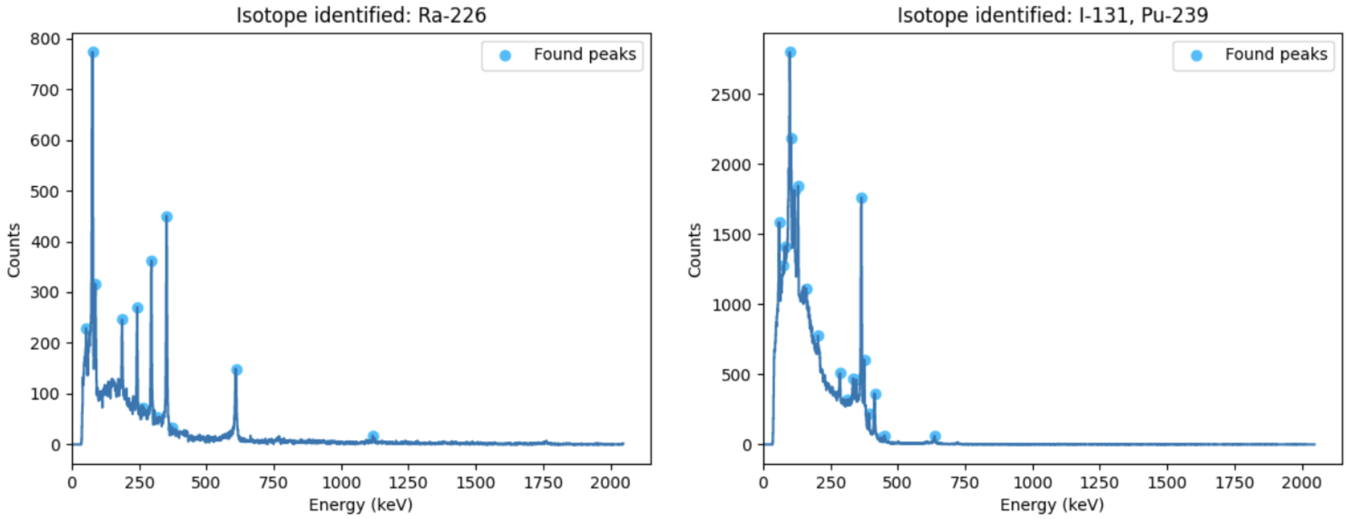


Figure 24: Isotope Spectra with Prominent X-ray Peaks

Including x-ray peaks allowed for more accurate detection of isotopes by decreasing the importance of prominent x-ray peaks. Importantly, this allows for isotopes to be detected when being somewhat attenuated, since this would stop low energy x-rays and significantly reduce the amount of x-rays detected. Therefore, x-rays should not be relied on to identify an isotope, and this isotope identification algorithm takes this into account.

Additionally, including x-ray peaks and taking into account the prominence of peaks allowed for more accurate identification of multiple isotopes that could have overlapping peaks. Two spectra including multiple isotopes are shown in Figure 25. These spectra include isotopes of uranium which can be tricky since they have strong x-ray peaks but the main peaks have lower prominences.

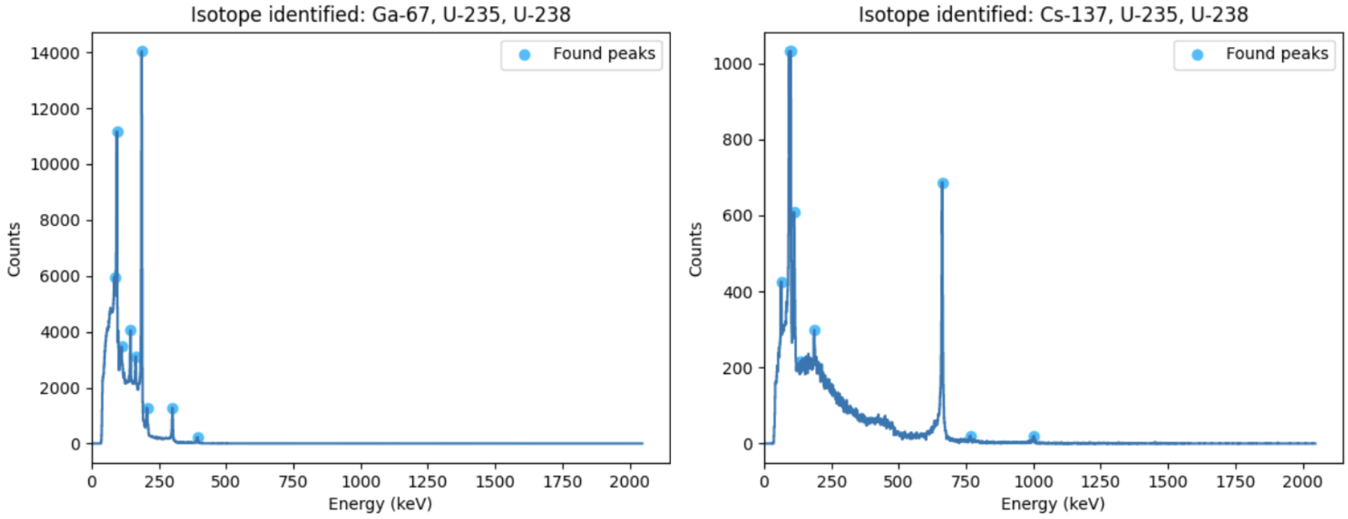


Figure 25: Isotope Spectra with Multiple Isotopes

The isotope identification algorithm correctly identified 160 of these spectra. This means that 100% of the spectra accurately identified the isotopes present and no others. Therefore, there were no false positives or false negatives in the isotope dataset, only true negatives and true positives.

6.3.3 Limits of Detectability

To determine the detection probability of the isotope identification algorithm it is important to look at the detection probability and false alarm probability. The algorithm performance can be characterized by Receiver Operating Characteristic (ROC) curves. A ROC curve is shown in Figure 26.

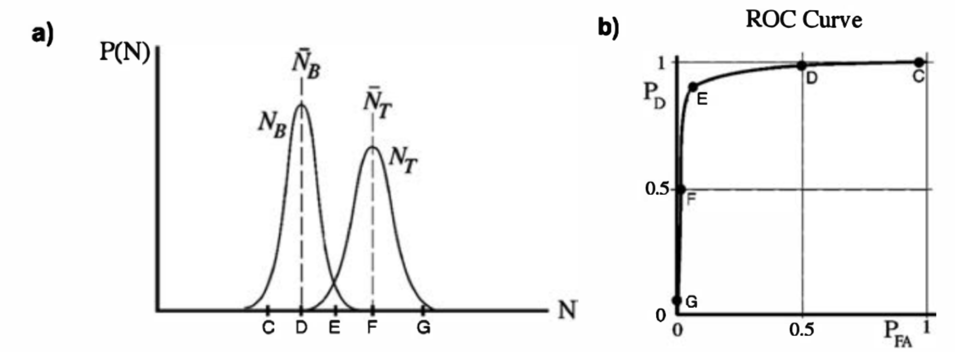


Figure 26: (a) Probability Distributions for Background Counts and Counts when Source is Present (b) ROC curve for Distributions in (a) Adapted from [12]

The horizontal axis for Figure 26b is the probability of a false positive and the vertical scale is the probability of a true positive when detecting an isotope. The ROC curve depends on the distributions shown in Figure 26a. Based on the test cases performed, the probability of a false positive is 0.006 and the probability of a true positive when detecting an isotope is 1. Although the test cases may not be representative of all possible measurements, this information likely means that the isotope algorithm falls between points E and D on the ROC curve in Figure 26b. This shows a high algorithm performance for measuring both background and isotopes.

6.4 Activity Analysis Results

6.4.1 Spectrum Analysis and Isotope Identification

The Spectrum class has been fully integrated with the isotope identification method, enabling the Python script to read .n42 files, identify peaks, and extract relevant information from these peaks, such as total counts, background estimates, background-subtracted estimates, and full-width half-maximum (FWHM) values. The ability to process .n42 files and accurately identify isotopes in the spectrum significantly streamlines the analysis process and enables a more comprehensive understanding of the data. Figure 27 displays a spectrum plot where the relevant photopeak is identified and both the background and net count sections are filled in. The current spectrum analysis can only account for one source at a time, as the algorithm struggles to identify counts under peaks other than the photopeak. The background count is estimated by interpolating both sides of the peak, which the algorithm has difficulty with for any peaks other than the photopeak.

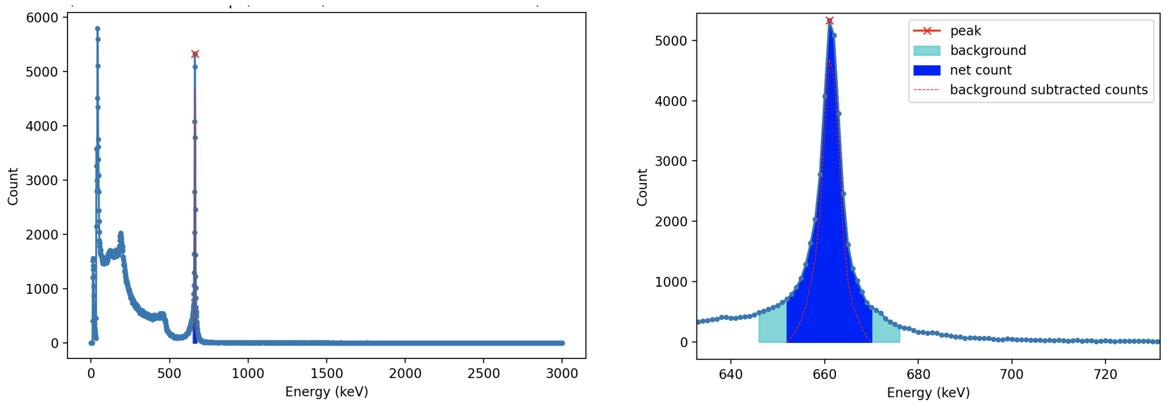


Figure 27: Cs-137 Spectrum with Peak Finding Algorithm Results on Left and Section Views on Right

6.4.2 Detector Efficiency Calibration

The detector efficiency calibration was performed using four sources at a 45-degree angle, with measurements taken for approximately 30 minutes. The source was set to be located 1m away from the detector crystal. The efficiency of the detector was calculated for the full energy photopeaks of each source at each angle, resulting in a lookup table that can be used for further calculation. Moreover, the Python script has been developed to interpolate the efficiency of the detector, given the approximate angle of the face of the detector with respect to the source. This allows for a more accurate estimation of the detector's efficiency under various experimental conditions. Figure 28 displays partial data on the detector efficiency, showing variations as the source direction is varied with respect to the detector face.

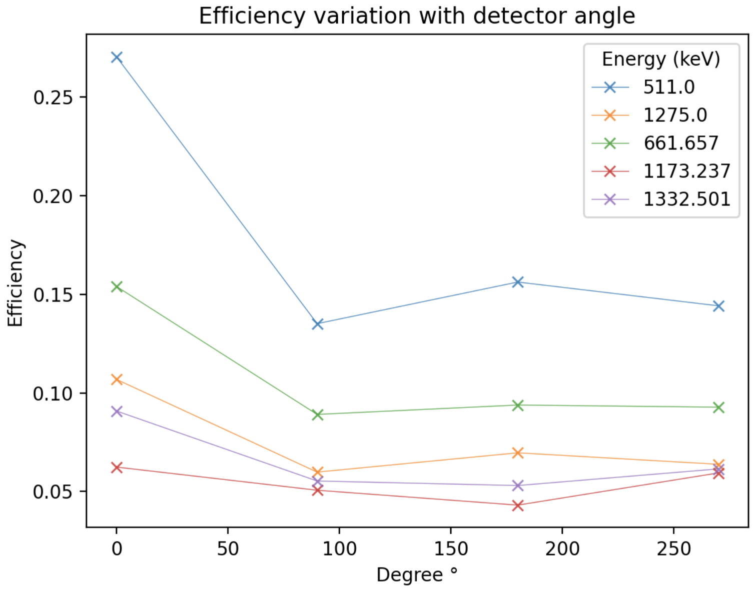


Figure 28: Detector efficiency with variation of source direction with respect to detector face

6.4.3 Activity Calculation

The activity calculation scripts have been developed to estimate the activity of the source given the approximate direction and distance of the source from the detector. The scripts to determine the peak energy and the counts under the peak remain the same. The source position estimate and direction estimate has been replaced with hard coded values for testing purposes.

6.4.4 Integration with Compton Imaging and Count Rate Imaging Methods

While significant progress has been made in developing the activity analysis components, the current implementation does not yet integrate the Compton imaging and count rate imaging methods for source distance and direction estimate. Future work will focus on merging these components to provide a complete and comprehensive analysis of the experimental data.

The imaging methods for source distance and direction estimate will result in uncertainties in source activity. Moreover, the accuracy of source position direction is limited to approximately 2 degrees. Thus, further analysis is required to determine the optimal measurement time and distance from the source to accurately determine its activity. This includes improving the accuracy of the source position estimate by determining the tolerance for position estimate uncertainty necessary to achieve an activity estimate within 30%. Additionally, the measurement time must be optimized to minimize uncertainty in the count rate measurement. These efforts require further experimentation and data analysis with the fully integrated algorithm along with the Rosbot data output.

7 Conclusions and Future Work

7.1 Key Takeaways

Based on the project results and analysis, it was determined that radiation detection robots can be utilized to effectively localize a source within 0.21 m ± 0.05 m. The ability of the Rosbot to localize a radioactive source within 0.21 m makes it practical for radiation detection use. This is beneficial because robots can be deployed in place of occupational workers, which has potential to lower occupational dose, and adds capabilities to survey hard to reach areas. Additionally, the results displayed that radiation detection robots are able visualize radiation hot spots. Compton imaging uses projected Compton cones to effectively determine the precise source location, while count rate imaging can identify general radioactive areas. Finally, radiation detection robots can be coupled with methods to characterize sources. By utilizing gamma spectroscopy, the specific isotopes present in an environment can be identified and further analyzed for activity.

7.2 Future Work

While the Rosbot and current methods described throughout the report are a solid basis for radioactive source localization, further iterations and improvements would increase the effectiveness and efficiency of this process. First and foremost, work should be dedicated towards autonomous development. This would involve developing further iterations of the processing scripts to instantly process measurement data. Furthermore, additional real world testing of environments with multiple sources should be explored. This would test both the imaging and characterization methods, as the imaging methods would have to correctly account for multiple sources of radiation, and the characterization methods would have to process and analyze data sets from multiple isotopes simultaneously. Finally, it is of interest to study the timing capabilities of the CZT M400 detector. By determining the minimum measurement time needed to provide sufficient data for both localization and characterization, the efficiency of the entire process can be increased. Overall, the Rosbot is currently a sufficient method for source localization and characterization, and further iterations of this development will improve the potential efficiency and applicability of the product.

8 CredIT Statements

- **Undarmaa Ganbaatar:** Software, Data Processing, Methodology, Visualization, Formal Analysis
- **Emeline Hanna:** Methodology, Software, Validation, Formal Analysis, Visualization, Writing
- **Katie Olivas:** Software, Data Processing, Image Development, Experimental Procedures, Organization
- **Isaac Reichow:** Software, Data Processing, Image Development, Experimental Procedures
- **David Goodman:** Conceptualization, Methodology, Software, Resources, Supervision, Project Administration, Funding Acquisition, Data Curation
- **Zhong He:** Conceptualization

- **Brian Kitchen:** Conceptualization, Methodology, Software, Resources, Supervision, Project Administration, Funding Acquisition, Data Curation

9 References

[1] "FUKUSHIMA DAIICHI DISASTER", Olivia's Blog, 31-Oct-2014, https://sites.suffolk.edu//fukushima-daiichi-disaster/ (accessed Nov. 20, 2022).

[2] M. Yamaguchi, "An ex-Fukushima worker is the first confirmed to have cancer from radiation," Business Insider. https://www.businessinsider.com/an-ex-fukushima-worker-is-the-first-confirmed-to-have-cancer-from-radiation-2015-10 (accessed Nov. 20, 2022).

[3] O. US EPA, "RadTown Classroom Materials: Radiation Exposure," Jan. 31, 2019. https://www.epa.gov/radtown/radtown-classroom-materials-radiation-exposure (accessed Nov. 20, 2022).

[4] "Rosbot mini," RoboWorks. https://ozrobotics.com/vendors/roboworks/ (accessed Nov. 20, 2022).

[5] "M400 Custom Integrable Detector Module." H3D, Ann Arbor, 2017.

[6] West, A., Tsitsimpelis, I., Licata, M. et al. Use of Gaussian process regression for radiation mapping of a nuclear reactor with a mobile robot. Sci Rep 11, 13975 (2021). https://doi.org/10.1038/s41598-021-93474-4

[7] Feiyun Cong, Yusuke Tamura, Kenji Shimazoe, Hiroyuki Takahashi, Jun Ota, Shuiguang Tong, Radioactive source recognition with moving Compton camera imaging robot using Geant4, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 953, 2020, 163108, ISSN 0168-9002, https://doi.org/10.1016/j.nima.2019.163108.

[8] "How Does Gamma-Ray Imaging Work?" H3D, Ann Arbor.

[9] "Cesium-137," nuclear-news, Dec. 03, 2011. https://nuclear-news.net/2011/12/03/high-radioactive-cesium-levels-in-abukumagawa-river-japan/cesium-137/

[10] "What is the Difference Between rviz and Gazebo? – Automatic Addison," Jun. 24, 2020. https://automaticaddison.com/what-is-the-difference-between-rviz-and-gazebo/ (accessed Jan. 30, 2023).

[11] "American National Standard Performance Criteria for Handheld Instruments for the Detection and Identification of Radionuclides - Amendment 1," in ANSI N42.34a-2020 (Amendment to ANSI N42.34-2015) , vol., no., pp.1-15, 10 Nov. 2020, doi: 10.1109/IEEESTD.2020.9257087.

[12] Knoll, Glenn F. *Radiation Detection And Measurement,* Fourth Edition, 2010

10 Appendix

10.1 Gantt Chart

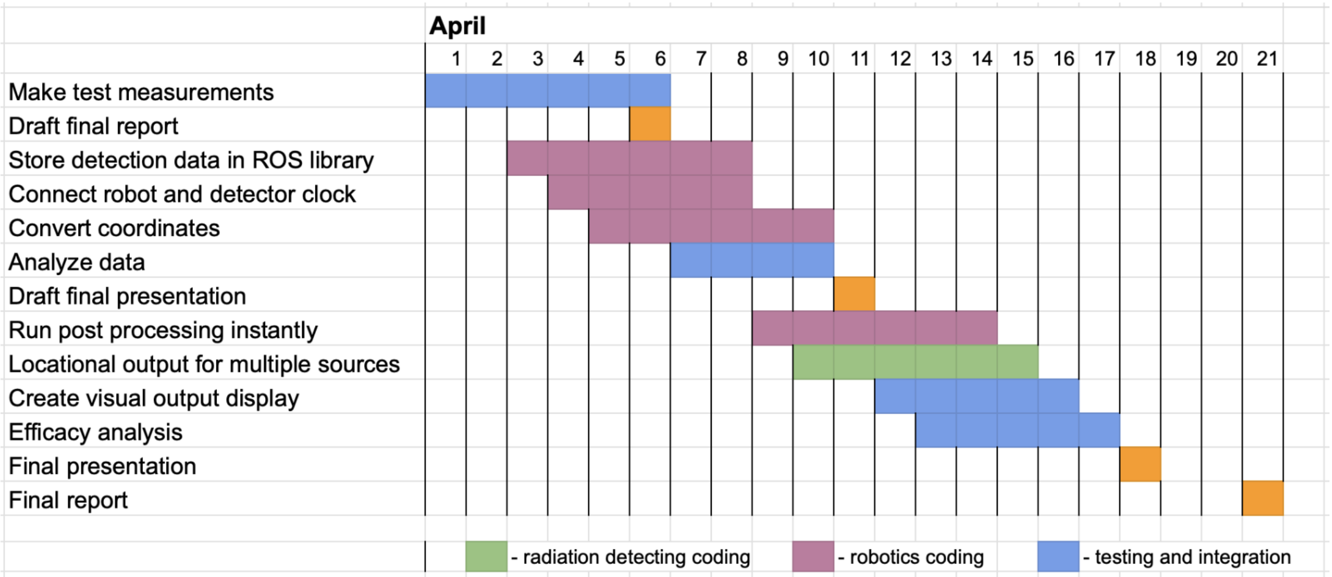


Figure 29: Final Gantt Chart

10.2 Codes

All relevant codes can be found at the GitHub page https://github.com/EmRobotics/RadRobo.