A Compton Imaging Detector Integrated With A Robotic System as a Radiation Survey Tool

Report for:
NERS 492
Senior Design II

April 26, 2024

Students:
Giuliano F. Basso, Aaron Belman-Wells, Caleb M. Bush
University of Michigan - Ann Arbor

Mentors:
James Berry, Zhong He
University of Michigan - Ann Arbor

Instructors:
Kevin G. Field, Stephanie Sheffield, Thomas Jayasankar
University of Michigan - Ann Arbor

Team:
Spec-G

Abstract

Any level of radiation sufficient to pose a danger to human life is, by definition, sufficient to pose a threat to human radiation workers. Furthermore, there are regions, such as near the core of a nuclear reactor, where radiation surveys may be necessary, but rendering the space safe for workers can be prohibitively expensive. In order to keep radiation workers safe, a remotely controlled robot equipped with advanced sensors capable of detecting and characterizing radiation sources is highly desirable. With such a goal in mind, the foundational architecture for such a robotic system was developed by a previous team, relying on Compton imaging for measurements of incident radiation direction. Continuing work on this project has improved the system in two primary manners. First, the speed of data acquisition has been improved so that no post-processing of collected data is required. In measurements of a 63 uCi ^{137}Cs source taken at an average of 1 m, the completed system took no more than 45 seconds to produce the detector-source angle with less than 5° error. Second, the software used to determine the location of the radiation source has been upgraded to operate in three dimensions (3D), whereas previously, it only worked in two dimensions and could not give information about the height of the source. Tests of this software have been reliably successful in locating the same ^{137}Cs source to within 15 cm. These improvements have resulted in a system capable of constantly delivering source isotope and position data to a user as it is used, making it a powerful remote survey tool.

1 Introduction

Any situation involving an unknown location or quantity of radioactive material, such as a lost source or contamination event, presents a unique challenge: the more dangerous the source is, the harder it is for human operators to search for it without endangering their own health. This issue becomes significant in situations such as the lost source incident in Goiania, where a 1375 Ci ^{137}Cs source was lost in a junkyard [1]. Containing this source required human workers (shown in Figure 1.a) to search the area with survey meters, exposing them to the dangerous source. The appeal of a robotic survey system in radiation protection is simple: robots are not endangered by high radiation, and therefore, no consideration needs to be made to their accumulated dose as they work to find a potentially dangerous source. Many such systems have been designed, with many implementations [2] relying on only count or dose rate measurements transmitted wirelessly to the user. This information is much like a game of 'hot-and-cold': an operator must track which directions show higher radiation and continue on that path. This implementation (shown in Figure 1.b) provides data identical to a human walking around with a survey meter. It is, therefore, preferable to a human surveyor in situations with the potential to deliver a high dose to human operators.

(a) Radiation workers in full PPE search for a lost source in a junkyard

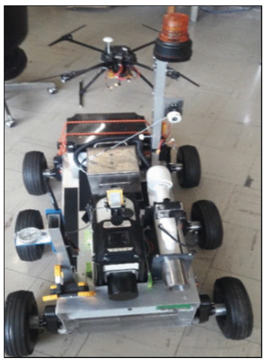

(b) A robotic system to take radiation flux measurements remotely [2]

Figure 1: Comparison of two methods of surveying radiation

1.1 Problem Definition and Project Goals

The ‘figure-of-merit’ for a robotic survey system is twofold: How precisely can the system determine the location and characteristics of a radioactive source, and how quickly is this information provided to a user? Determining a source’s location and characteristics accurately provides a radiological health benefit; a human who needs to retrieve a source from within a known area can do so much faster if that area is a single small bag or container rather than within a few meters. On the other hand, the measurement speed is more of a logistical issue, as a ‘faster’ system is helpful in more cases. For example, if an entire hospital needed to be surveyed for a lost source, having a system that takes 15 minutes to survey a single room accurately may not be reasonable. A system that takes only 5 minutes would be a far better option. Making the system provide information at speed is, therefore, also a radiological health benefit, as it can be used in more situations and, therefore, keep more human operators out of harm’s way. Therefore, this project aimed to create a robotic survey system capable of providing a user with information about the position of a source accurate to within 15 cm (about the size of a personal backpack or suitcase) in the shortest time possible.

1.2 Background and Systems Used

While the system shown in Figure 1.b is undoubtedly preferable to endangering an operator, it does not store any data, limiting the information delivered to the user. This means that a human operator will eventually need to hunt for the source with only a rough idea of its location. Suppose a system records the same information but stores and displays all information recorded during a survey. In that case, maps of the count or dose rate encountered can be made, allowing a user to understand the true source location better. For even higher fidelity, a detector known as a Compton camera, which can detect the incident direction of gamma radiation, can be used in place of, or in conjunction with, the survey-meter type detector. Using constant directional measurements allows for much faster and more accurate estimates of a source position and can determine source position in 3 dimensions, which is impossible for count-rate imaging. Figure 2 shows a simple demonstration of this principle; consider which image gives you more information in the same number of measurements.

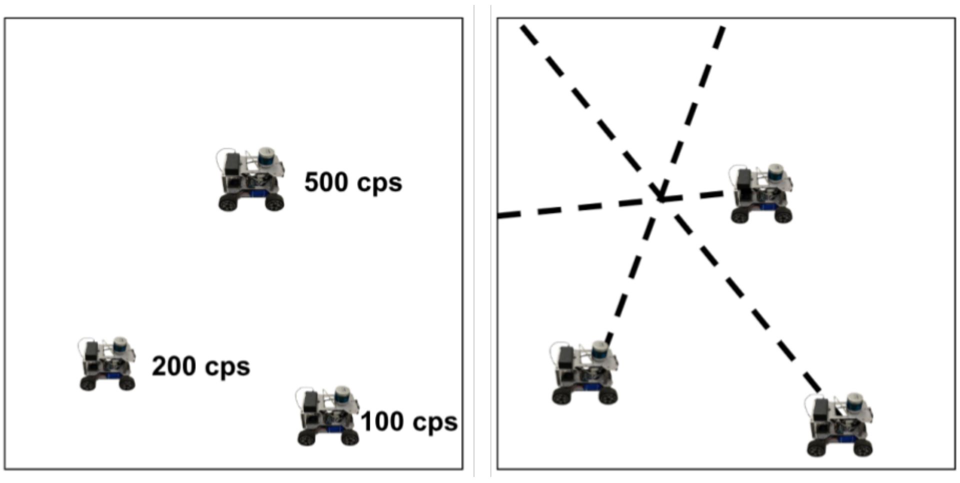


Figure 2: Demonstration of Count-Rate (Left) vs Angle-Based (Right) Position Finding

This project attempted to accomplish the precise localization and mapping of radioactive sources described above by building upon a pre-existing system (Figure 3), which relies on an M400 Compton camera to determine angles to a source [3] [4], as well as a commercially available robotic platform. In the first implementation of the project, which was completed by a previous Nuclear Engineering design team during the 2022-2023 academic year, the robotic system was driven around an area with a known source, returned to the operator, and the stored information was downloaded and processed into maps, as shown in Figure 3. This project implementation was highly accurate, successfully locating sources (in 2 dimensions only) to within 0.21 $\pm$ 0.05 m. However, post-processing of the data took several minutes and required the robot to return to the user [3]. This reduces the system's usefulness in most real-world situations, as the user cannot decide where to take more measurements based on the data they have collected, forcing them to spend several minutes taking measurements at any position where the source could exist. If the user could instead see the data as it was collected, they could quickly eliminate areas with only background radiation, only spending time imaging areas where a source is present.

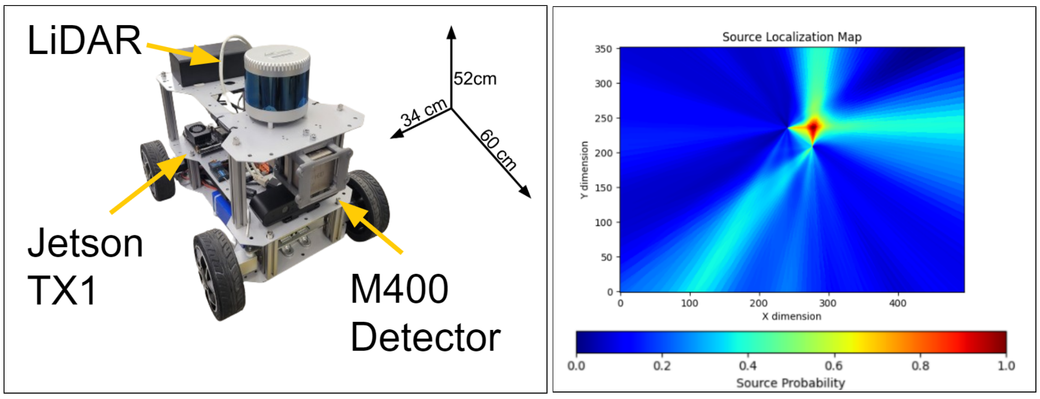


Figure 3: A robotic system with an integrated Compton camera (left) and an example map created by that system (right).

1.3 Approach

In order to achieve our goal of producing accurate information in a short time, work on this project focused on improving the existing system in two ways. The first improvement was to allow the system to collect 3D positional data regarding radiation sources. As discussed previously, providing users with detailed source position information minimizes human workers' time locating and containing hazardous sources. Second, a substantial overhaul of the software responsible for isotope identification and source location finding was performed to enable data to be simultaneously wirelessly transmitted to users and processed as the system runs.

2 Design Objectives and Project Requirements

This work focuses on taking a pre-existing system [3] and improving it to the point of being a usable radiation survey system. To this end, two key benchmarks were defined as the metric for a 'good' survey system: the accuracy of the system and the speed with which the system is capable of processing the data and sending information to a user. Specifically, the real-time software must deliver comparable accuracy to the prior system (sources located to within 15 cm).

2.1 Source Localization Accuracy

In order to be considered successful, the final system had to be capable of locating an unshielded 60 μCi point source with emissions in the 250 keV - 3 MeV range, the energy band in which Compton scattering is the most prominent photon interaction [4], to within 15 cm in at least 95% of laboratory tests.

The 60 μCi source was selected as being too small to be likely to cause harm to radiation workers searching for it while being sufficiently large as to make a single test take in the range of minutes. The 15 cm radius was chosen from an estimate of the maximum volume which could easily be collected and moved elsewhere if the source is not immediately apparent.

2.2 Source Identification Accuracy

The system was required to be able to identify a single photon-emitting radioisotope based off of the measured energy spectra from laboratory tests with at least 95% accuracy. It also had to be able to detect the presence of multiple radioisotopes in at least 95% of such tests. These two accuracy requirements were chosen to meet the IAEA American National Standard Performance Criteria for Handheld Instruments for the Detection and Identification of Radionuclides (N42.34) [5] as well as being reasonable given the M400 detector's energy resolution for the 662 keV ^{137}Cs peak of 0.64% [4].

2.3 Speed

The system had to be able to return all of the above analysis to the operator within 60 seconds of receiving all of the incoming data. This includes being capable of processing data faster than it comes in so that it will not exceed the 60 s time allotment for long measurements. The time was limited so as to ensure that the operator would be able to use information collected by the robot to give it new directions within a single survey and the specific time limit was set as approximately the same amount of time required to collect a new measurement for Compton imaging.

2.4 Major Technical Challenges

Several specific challenges resulted from these requirements. First, the minimum source activity was set significantly below any potentially dangerous source, and therefore measurements were subject to higher noise from background radiation. Second, lacking access to the inner workings of the H3D code already developed to perform Compton imaging meant that such a program had to be developed from scratch, which was time-intensive. Finally, a limited number of available isotope check sources decreased the amount of data available for tuning isotope detection parameters.

3 Project Plan and Schedule

Upon beginning the project, three main subsystems were noted as needing to be partially or wholly redone in order to meet the new design requirements. The first of these was Compton imaging, which, to operate in real-time, needed to draw on a less processed form of the data output by the M400. This meant a complete re-design of the Compton imaging algorithm. The second system that needed overhaul was the isotope identification, which had to be made to collect and act on real-time data and be more robust against spectroscopic aberrations caused by the physical system and background radiation. Finally, the parallax program (which compares all collected data to determine the most likely position of a radioactive source) needed to be redone to operate in 3D, which also required new functions to account for the fact that the rays were no longer guaranteed to intersect with each other.

3.1 Team Roles

Due to the three primary tasks not needing to be completed sequentially, though they did feed into each other, it was decided that each team member would work in parallel on one of the tasks, as determined by interest and expertise. Caleb M. Bush, the team member with the most experience in radiation detection, was chosen to be in charge of the Compton imaging program. This also involved learning to receive and understand the flatbuffers serialized data format the M400 uses to send live data. Giuliano

F. Basso lead development of the isotope identification program, working to make it fast enough to run in real time and sufficiently robust to filter out background noise and other real-world confounding factors without ignoring real sources. Aaron Belman-Wells redid the parallax program to operate in 3D quickly enough to run in its entirety between new Compton images being made, ensuring that no backlog would build up. In addition, he worked on tracking the robot's position, which is required for the parallax program to function.

3.2 Schedule

Work on the project took place over the course of two semesters, starting at the beginning of November 2023 and lasting until the middle of April 2024. The specific tasks being worked on by members at any given time are shown in Figure 14 in the appendix.

4 Methods

In order to achieve live data processing, nearly all of the software running on the prior system was rewritten to handle data as it was produced. There are three main data producing modules: the M400, LiDAR, and 'dead-reckoning', which refers to the array of systems present on the robot which track its movement. The data from the M400 consists of a stream of events in which each event contains the Cartesian position of the interaction, the deposited energy at that position, and the time at which it occurred. For Compton imaging, only the instances in which two interactions appeared simultaneously are processed by the software, while the spectral identification algorithm processes all events, looking at the 'true' spectrum. A Python script that runs on the robot's computer consistently compares the dead-reckoning and LiDAR data and publishes estimates of the robot's position relative to some defined origin. The M400 constantly publishes its raw data, which is received wirelessly by a user's computer, where the isotope identification and Compton imaging algorithms run. All of this information is then processed by a fourth Python script, which provides estimates of source position. Figure 4 gives an overview of this process, demonstrating how each module references the others to produce a final output.

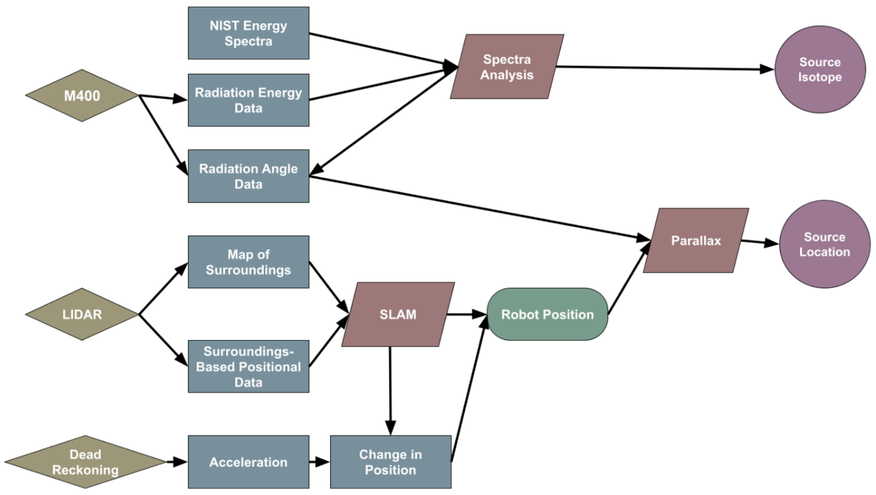


Figure 4: Data flow for integrated robotic and software system

4.1 Critical Component: M400

The M400 is a position sensitive Cadmium Zinc Telluride (CZT) gamma-ray imager. It has an active volume of 19 cm^3 and an energy resolution of 0.64% for ^{137}Cs. As discussed above, this detector constantly publishes data about the position and energy of radiation interactions. The system also comes with software that creates a live heatmap of incident radiation. Unfortunately, this is proprietary software which could not be directly accessed or changed by our design team. Therefore, in order to integrate the data produced by the Compton camera with other systems, data must either be pulled from the images produced by the pre-existing software, or the raw data can be received and processed. For this project, it was decided that processing the true data would provide both more accurate measurements and control over the processing time, and therefore a Compton imaging program needed to be written.

4.1.1 M400 Data

The H3D list-mode data stream is serialized using the open source FlatBuffers library [6]. By building the Flatbuffers compiler on a computer, a schema (specialized data format) which is provided by H3D [6], may be used to interpret the data published by a port on the M400's internal computer.

4.2 Compton Imaging

Once the data referring to the coordinates and energy of Compton scatters can be read on the user's device, it must be processed to find the incident angle. This is done according to the data flow described in Figure 5, where the data is filtered, mapped to a unit sphere, and the best point is found. Once an angular direction is determined, that information may be sent to the parallax program to determine the true position of the source.

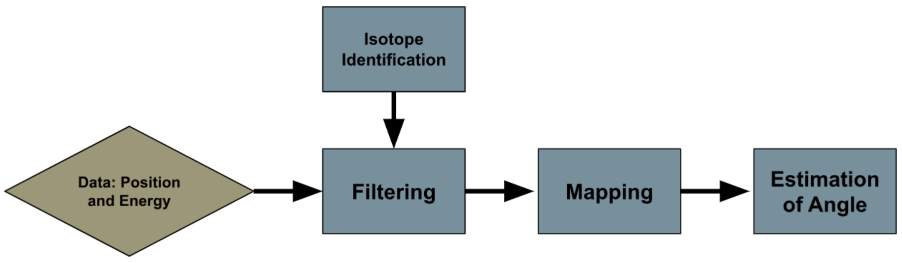


Figure 5: Compton imaging data flow

4.2.1 Data filtering

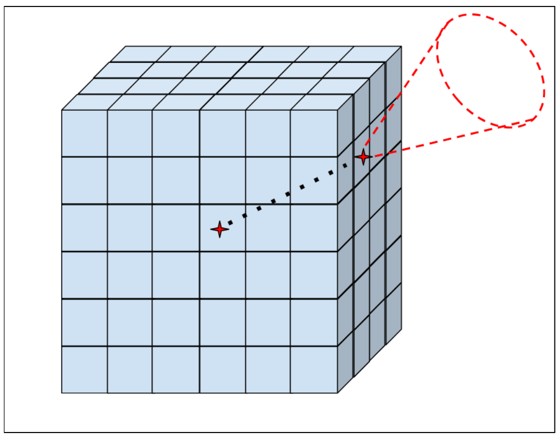
When trying to achieve fast processing, only the Compton scatters which provide useful data should be processed. Therefore the first module of the Compton imaging algorithm uses the defined photopeaks identified by the isotope identification program, and only processes interactions which deposit within 4% of the photopeak energy. The rationale for this comes from the Compton scattering formula (Formula 1) [7] which requires knowledge of the incident energy to find the angle a photon deflects along.

$$E_f = \frac{E_i}{1 + (\frac{E_i}{m_e c^2})(1 - cos(\theta_\gamma))} \tag{1}$$

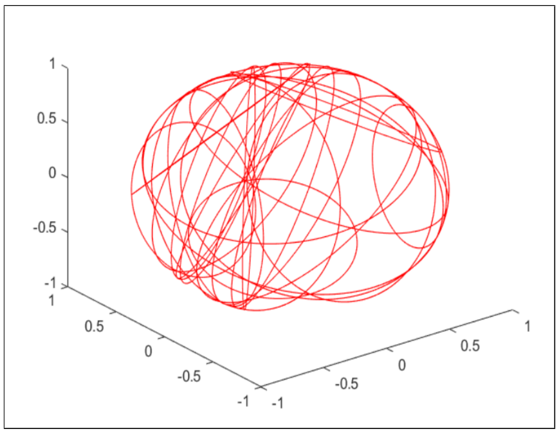
The second part of the data filtering module deals with the order of events. As a Compton scatter occurs in less than 10 pico-seconds (the time it takes light to move the full 2.7 cm across the M400), the events do not appear to the computer as one after the other, but at exactly the same time. In order to process the information, a guess of the correct order must be made. First, the program checks to see if either sequence of events is impossible, meaning the photon could not have deposited the specified energy in its first collision. If one direction is impossible, the correct sequence is known. This process usually applies to 30-40% of the Compton scatters. For other events, a guess is made based on the more likely sequence, which is that the higher energy deposition is more likely to be first for interactions with a total energy greater than 400 keV (and vice-versa) [7]. This module then outputs a directional vector and scatter angle. It should be noted that processing events with three or more interactions would require far more processing, so this system does not consider these events (this eliminates less than 10 % of ^{137}Cs Compton scatters.

4.2.2 Mapping Data to a Unit Sphere

The second module considers the world as a unit sphere around the robot. For each Compton scatter, this module receives the photon's directional vector within the M400 and the angle by which the photon must have changed to begin traveling in that direction. This creates a cone of possible incident angles and the points at which this cone intersects with the unit sphere are recorded. Figure 6 shows how two points within the M400 are projected into a cone, as well as how those cones overlap on the unit sphere. The mathematical process behind this projection is detailed in the appendix.



(a) Direction and angular data from module 1 turned into a cone with a central vector



(b) Compton cones mapped to an arbitrary unit sphere surrounding the M400

Figure 6: Visualization of Compton scatter data processed into a map of overlapping cones

4.2.3 Estimation of Angle to Source

With enough Compton scatters ($\sim$ 200), there will be enough overlap to determine the angle to the source with reasonable accuracy. This could be done by simply taking the point of greatest overlap. However, a more accurate method is to take an average of some of the highest points. H3D reports the M400 as having a Full-Width at Half-Maximum (FWHM) of approximately 30° for 4π imaging in real-time. Therefore, the program finds the 30° piece of the unit sphere with the most intersections and uses a weighted average of only that area to determine a final angle to output.

4.3 Isotope Identification

In addition to understanding the position of a source, classifying the source isotope allows an operator to make informed decisions. For example, searching a lab with multiple sources for a lost one requires disregarding the others. Furthermore, the Compton scattering algorithm can run far more efficiently if it is told what source it is identifying; therefore, creating a live isotope identification program is beneficial. The newly written Python isotope identification program wirelessly receives n42 data and con-

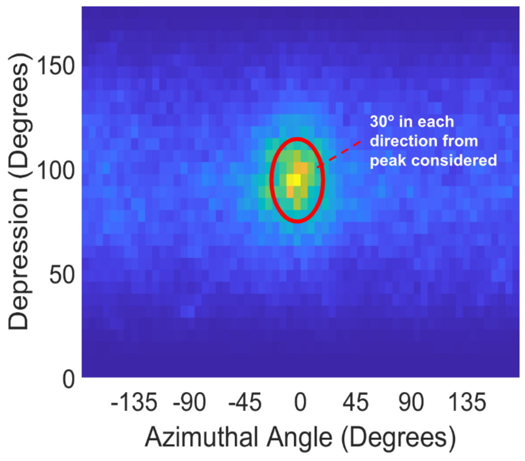


Figure 7: Number of overlaps in 5° sections of the unit sphere plotted as a heatmap

tinuously stitches the data into a single energy spectrum that can be accessed at any time, as shown in Figure 8. This can either be done on a user's computer wirelessly or run entirely on the robot's onboard computer. The most recent spectra are displayed for the operator at all times and used to determine the photopeak energies the Compton imaging program should attempt to look for.

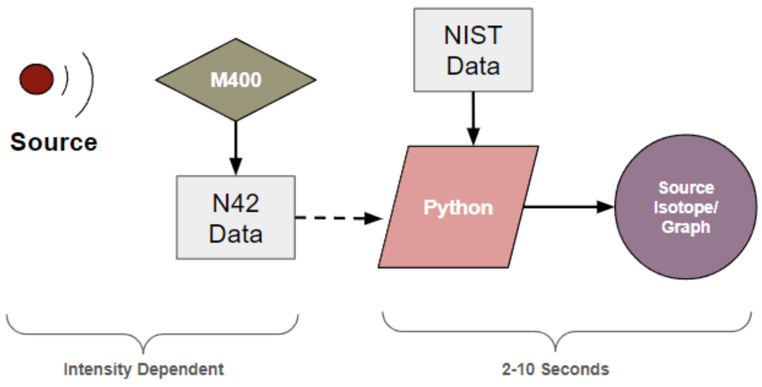


Figure 8: Workflow of Isotope Identification

In order to make full use of the excellent energy resolution of the M400 and produce high-accuracy isotope identification, the energy spectrum is smoothed using a weighted moving average method known as the Savgol_filter. This filter allows for the suppression of local peaks and ensures that only statistically meaningful local maxima are processed by the find peaks algorithm. This is especially meaningful at low gamma energies, which blend with many X-ray emissions, and when low count rates were taken where statistical uncertainty can show up as small 'peaks.'

Critical peaks are identified using the Python library SciPy find_Peaks [8], a tuneable function that takes a 1-D array and finds all local maxima by simply comparing neighboring values. It has several parameters that can be altered: height, threshold, distance, prominence, width, and plateau, which must be tuned for specific situations. Prominence, which determines the height relative to the rest of the spectra that define a photopeak, must be set higher for low-count spectra to avoid noise. However, it can be decreased as more counts are collected, providing better accuracy. Therefore, the prominence for each section of the spectrum is constantly updated based on the number of counts collected in that energy range. A minimum height threshold was set based on the smoothed spectra and the uncertainty of neighboring counts, ensuring that noise is suppressed for low counts, low energy, and high background conditions. Lastly, the minimum width of a peak depends on electronic noise in the detector; this relation varies linearly with the energy of the peak due to the Poisson distribution of charge production [7]. Thus, we know the peaks will broaden with increasing energy, and when two or more photopeaks are seen, one with a small and one with a large FWHM tuning parameter is vital. We then use a second pass of the find peaks with a broader width setting for broader spectra to ensure all peaks are identified.

A database of the known gamma spectrum for 90 key isotopes in the gamma range we can detect (30 keV to MeVs) has been compiled from data published by the National Institute of Standards and Technology (NIST)[9]. This data set contains each element's name, isotope, gamma emission, and probabilities. After identifying critical peaks from the M400 data, we compare them to the database with the true peaks. If a peak matches, a weighted score is given based on the emission probabilities, and the program iterates through all isotopes until the highest correlation for any given number of 'found' sources. This operation is performed every 25 seconds, and the data is displayed to the user.

4.4 Robot Position Finding

For the Compton imaging program to operate, the location and orientation of the detector must be known for each measurement. Since the robot is expected to be able to operate without the presence of humans, this information must be gathered by the robot itself. This is done by the comparison of two position-finding methods: dead-reckoning and LiDAR.

4.4.1 Robot Position Finding: Dead Reckoning

The first method of finding where the robot is dead reckoning, which uses an initially known location and a measured change in position to determine the new position. Since no information is needed about the location of the source with respect to anything outside the room it is in, the initial position of the robot can taken to be the origin and facing directly towards the positive X-axis.

The robot is equipped with a total of six inertial measurement units (IMUs) which use changes in capacitance to determine acceleration along three translational and three rotational axes. These values can then be integrated twice with respect to time to determine the change in position of the robot. Additionally, an onboard odometer measures the number of times each of the four wheels of the robot turn and at what angle the front wheels are set relative to the rest of the robot. Since the size of each wheel and its position relative to the others is known, this can then be used to determine the distance traveled and the radius of any turn being made, which can be combined to again find change in position.

Both of these methods can offer extremely precise data, but have a tendency to drift over time. This is especially true for the IMUs, which require extremely precise initial calibration and can be knocked out of calibration by sharp jolts.

4.4.2 Robot Position Finding: LiDAR

Light Detection And Ranging (LiDAR) determines the distance between a sensor and the nearest surface in a particular direction by bouncing a laser pulse off of that surface and back to a detector on the sensor. When many of these sensors are put together, they can create a map of a surface. The array of sensors mounted on the robot covers a 270° arc in front of the robot and parallel to the ground. This allows for Simultaneous Localization And Mapping (SLAM) to create a map of objects at the height of the LiDAR unit on the robot and where on that map the robot is. This has the benefit that, so long as the robots surroundings are stationary, there is no drift, but does not offer precision to better than a few centimeters, depending on the robots surroundings.

4.4.3 Comparing Dead Reckoning and LiDAR

Both dead reckoning and LiDAR can be used to determine the position and facing of the robot, but have their own drawbacks: dead reckoning starts off with high precision which decays as the robot moves and LiDAR offers a roughly constant but never excellent level of precision. To minimize the faults in the system as a whole, then, Robot Operating System (ROS) was used, which is able to determine when the error in the dead reckoning systems exceeds the error in the LiDAR system and move to LiDAR, thereby offering the best possible accuracy at any point, as shown in Figure 9.

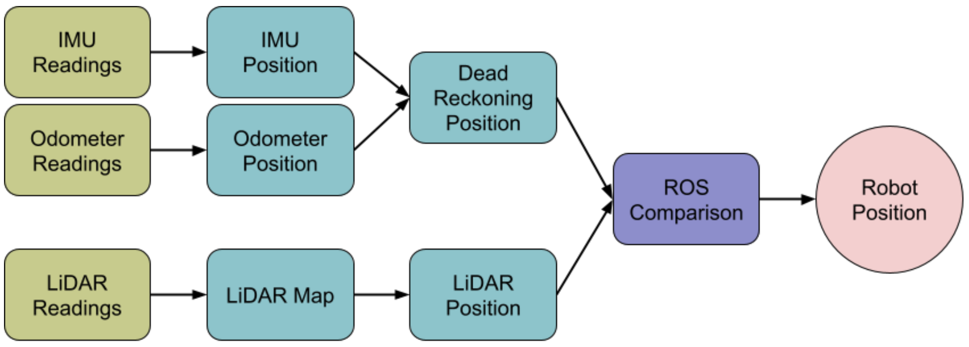


Figure 9: Workflow of Robot Position Finding

4.5 Parallax

The data gathered from Compton imaging guess is combined with the time at which it was made, and each guess is appended to a CSV. Additionally, the time at which each position datum for the robot was recorded is included with the robots coordinates and direction of facing, allowing for each Compton imaging estimation to be matched with the position that it was at when that guess was made, assuming that the robot had not moved significantly during the data collection period. The stored map data of the volume the robot is surveying is then used to create a 3D matrix, with each element in the matrix representing a 125 cm^3 cube of real space, and the matrix is iterated through to determine which element comes closest to the rays, each of which is constructed from the position of the detector combined with the absolute angle between the detector and the source, as estimated from Compton imaging. This process is shown in Figure 10.

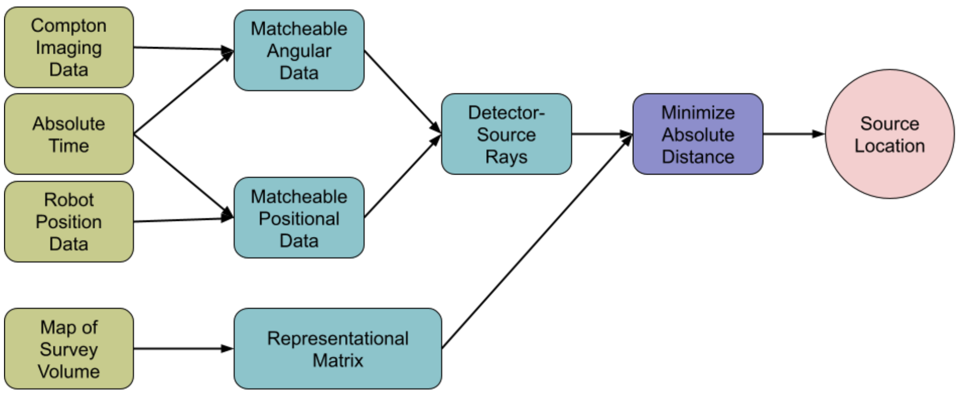


Figure 10: Workflow of Parallax Analysis

4.5.1 Parallax: Mean Absolute Distance

Since real world conditions mean that the rays calculated are unlikely to all precisely intersect, a method had to be chosen to determine which point was most likely to have caused the Compton imaging results previously obtained. The method selected was to find the point with the smallest mean absolute distance between it and each of the rays. For a line in Cartesian space that includes point $\overrightarrow{a}$ and direction $\hat{b}$, the minimum distance between it and point $\overrightarrow{c}$ is $d = |\hat{b} \times (\overrightarrow{c} - \overrightarrow{a})|$ [10].

For any point $\overrightarrow{c}$ that is in front of $\overrightarrow{a}$, that is $(\overrightarrow{c} - \overrightarrow{a}) \cdot \hat{b} \geq 0$, that equation is functional. In the cases where $\overrightarrow{c}$ is behind $\overrightarrow{a}$, however, the shortest distance between the point and the ray will be that from the point to the base of the array, $d = |\overrightarrow{c} - \overrightarrow{a}|$. The code used to determine the distance between a selected point and a given ray is shown in Figure 15 in the appendix.

4.5.2 Parallax: Volumes of Interest

One method that could be used to determine the estimated source location would be to iterate through every element in the volume matrix, select the element with the smallest mean absolute distance, and then perform a search within the space defined by that element to find the best fit for whatever level of granularity is desired. This, however, does not scale well; the time complexity of the calculations would be of the order of the size of the room, so while small rooms would be fast to iterate through, large rooms could easily take orders of magnitude longer.

To solve this issue, an algorithm was developed to find volumes where the source was likely to be and check those, discarding regions the source was unlikely to be in and significantly speeding up the calculations. This was done by finding which elements of the volume matrix each of the rays passed through and finding the mean absolute distance at their centers. The ten elements from the previous set with the smallest mean absolute distance then had each of their neighbors checked, shown in Figure 16 in the appendix, and this process was repeated until the list of ten best elements stopped changing. Out of this final list, the best element was selected and the 15 cm cube centered around it was broken up into 0.5 cm cubes, each of which was checked. Finally, the best of these was taken to be the location of the source, which could then be reported to the user.

5 Results

Each subsystem was created in isolation from the other systems, and tested and improved until it was determined that it functioned within our needs. Therefore, detailed data about the accuracy and error of each subsystem was compiled before the systems were integrated. These results are explained in detail in the following sections.

5.1 Compton Imaging

The accuracy of the Compton imaging program is determined by the number of Compton scatters processed, becoming sufficiently accurate for our purposes once roughly 200 Compton scatters have been analyzed. Processing further Compton scatters provides diminishing returns, as demonstrated by Figure 11. When tested on a 63 μCi check source of ^{137}Cs at 1 m, the M400 took between 30-45 seconds to acquire the necessary number of photopeak Compton scatters. In lab tests, the program was more than 95% successful at producing position estimates to within 5° of the true source position when given more than 200 Compton scatters, and processing this amount of data took between 10-15 seconds. H3D quotes the time to find a check source using their algorithm as 90s for a 10 μCi check source of ^{137}Cs at 1 m, and quotes the angular accuracy as within 1° [6]. Our program runs in comparable time, although the angular accuracy cannot reach 1° reliably without taking significantly longer measurements.

A 5° angular uncertainty was determined to be low enough for the parallax program to create accurate measurements. Therefore, this program meets both of our stated goals: it is accurate enough to allow further systems to identify the source position within 15 cm, and it runs in less time than the data collection, therefore taking essentially zero time for data processing. This program also provides comparable accuracy and timing to the quoted figures of the M400, implying our accuracy is constrained by the physical limitations of our equipment, and not by our approach or implementation.

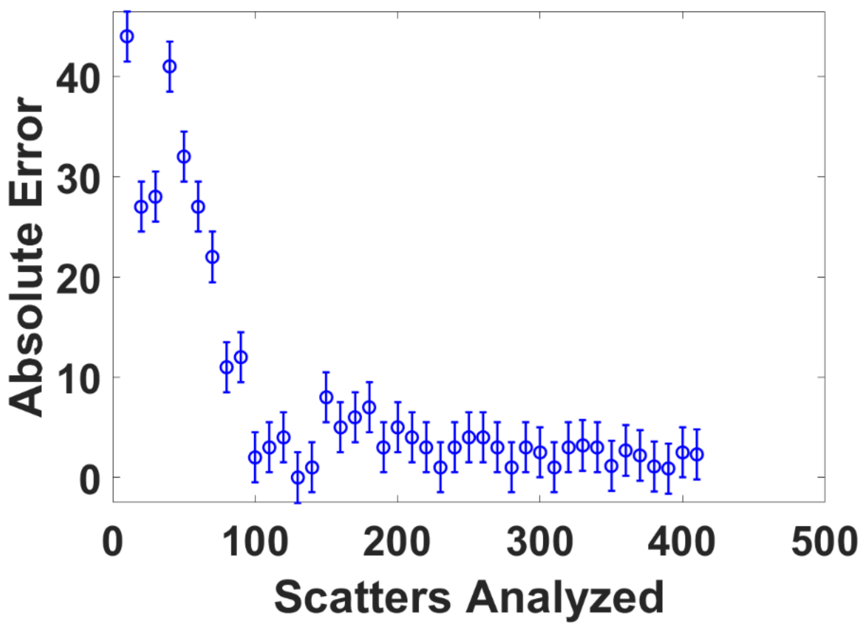


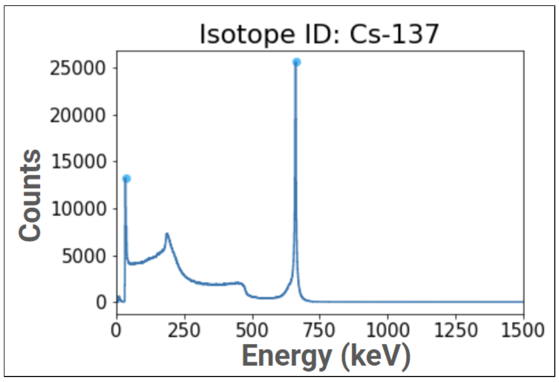
Figure 11: Error in angular estimation by imaging program as a function of scatters analyzed

5.2 Isotope Identification

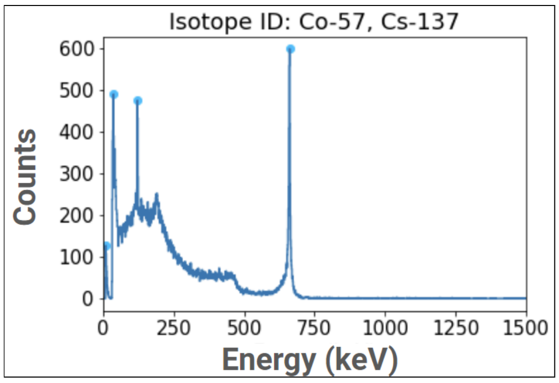
For thorough testing, the isotope identification program was provided two data sets: idealized data (hour-long with minimal background) and data taken in test source finding surveys. The script was able to identify the correct isotopes in 160 out of 160 H3D-provided spectra. For our 'realworld' test, several in-lab tests using ^{137}Cs, ^{60}Co, and ^{57}Co sources were taken over several different measurement lengths ranging from ten seconds to 11 minutes. All measurements were made 50 cm away from the detector, with no shielding. These measurements gave us good insight into the system's maximum capabilities. When the program was given real-time data, the combined data collection and identification process took 20 seconds, while when given longer measurements to post-process, the isotope identification algorithm was able to provide an output within 2 seconds, and the alternate double pass find_peaks script took 10 seconds. The data retrieved by our script can be seen in Figure 12 with the found peaks highlighted and the isotope guesses above. Once again, the program takes longer to collect data than process it, suggesting the physical equipment is the limiting factor, and our accuracy was determined to be sufficient for integration with other systems.

5.3 Robot Position Finding

The capabilities of the robot position finding software were measured by driving the robot around a track to a new known position and measuring the distance between the reported and actual final positions. This was performed on dry floor tiling, reducing the chances of the robot slipping, and in a room with no large empty sections of floor, giving the LiDAR system more points of reference.



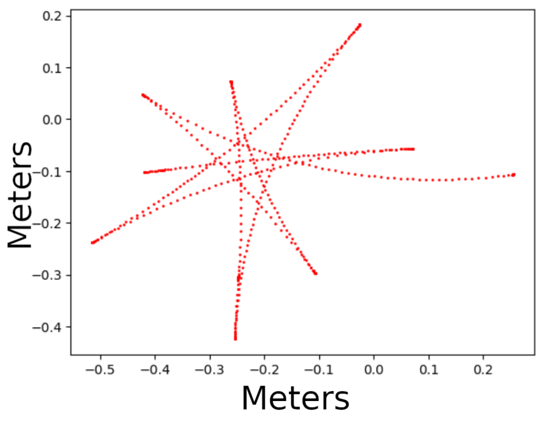
(a)Isotope Identification Output for ^{137}Cs taken in 5 minutes



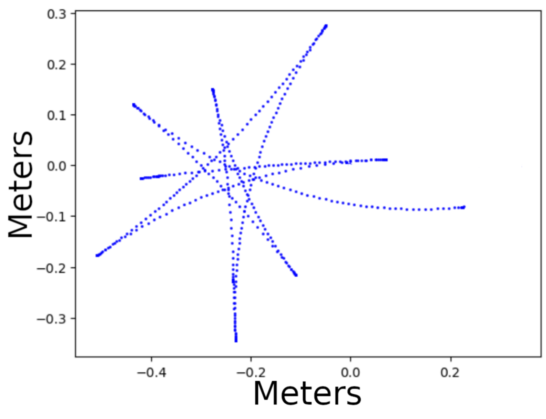
(b) Isotope Identification Output for ^{137}Cs and ^{57}Co taken in 90 seconds

Figure 12: Spectra Output with Isotopes Found

When used alone, the odometer was not able to provide accurate results, showing a leftward bias of several degrees per meter traveled compared to the actual robot path, which quickly compounded, rendering that method alone unusable. When the IMUs were added to the data set, however, the system became more accurate, being able to identify a 4 m straight forward path to within a half centimeter. The most accurate results, however, were produced when the LiDAR was also used, the results being shown in Figure 13.



(a) Dead Reckoning Only



(b) Dead Reckoning and LiDAR

Figure 13: Results of Position Finding

When it starts to move, the robot's estimated position drifts by approximately one millimeter per meter, traveling straight forward. This drift is increased by turning or significant acceleration. However, once the drift reaches approximately 10 cm, it ceases to grow as the LiDAR takes over. A 10 cm positional error is small enough for the parallax program to reliably produce less than a 15 cm error in the location of a source, so this program was also deemed successful and integrated.

5.4 System as a Whole

Due to time limitations, a limited number of full tests were able to be performed. For that reason, theoretical data was generated using artificially induced, worst-case situations of 5° angular and 10 cm translational error, making the assumption that the two errors were unrelated. One billion test cases with said data were run, with four artificial measurements being generated at one of several positions before being fed into the parallax program. In these cases, 98.3% of the test runs found the source to be within 15 cm, at an average of 8.32 cm error among successful tests, and failed cases still remained largely in the correct region, with an average 23.2 cm error.

5.4.1 Full System Test

The system as a whole was physically tested by placing a 63 μCi ^{137}Cs source on a raised pedestal and, the robot was moved around it, taking four measurements at an average of 1 m away over five minutes. Following the processing of the data, which took under one minute, the source was found to have an estimated position that was 3.9 cm away from its actual position during the test. The spectroscopy program also succeeded in identifying the source in under 30 seconds. In addition a similar trial conducted to test system integration and an older form of the robot position-finding algorithm localized the source with an error of 8 cm.

Both test cases returned results that were within the design goals of this project, and more accurate than the simulated results predicted. This is a very promising result, as it demonstrates our simulated tests with worst-case assumptions were likely an accurate representation of the worst performance that can be expected of this system. Nonetheless, both real-world and the theoretical tests exceeded the design goals of the project.

6 Conclusions and Future Work

Based on the results of both laboratory experiments run and more than a billion simulations run based on real world experimental data, the radiation detection robot developed by this team can be confidently characterized as being able to localize a radiation source to within 15 cm in more than 95% of cases and identify the radioisotopes present from an energy spectrum in more than 95% of cases. The robot can also create a map of the volume surveyed, with the likely location of the radiation source indicated thereon. Furthermore, these functions can be performed remotely and in real-time.

The final product of this work is therefore, as our goal stated, a robotic survey system which can deliver accurate information about the position of a radioactive source in three-dimensions to a user in little time. The developed software can take full advantage of the very high fidelity supplied hardware, enabling incredibly accurate source localization in real time. Such a system could be easily implemented in a variety of hazardous conditions; keeping radiation workers out of dangerous areas.

6.1 Future Work

Three primary areas of interest for future work have been identified. The first of these is increasing the number of sources the system can localize. Since the algorithm for creating Compton cones already only considers sets of events with the total energy deposited approximately equal to the emission energy of the detected isotope, simultaneous localization of multiple isotopes would not be a complex process. Localizing multiple sources of the same isotope, or even a continuous source, would be a significantly more difficult task but would greatly increase the applications of this robot. Secondly, further testing in real-world conditions, such as with a source significantly shielded in one direction or on a particularly hot or cold day, would increase confidence in the robot's capability to be used in the field. The final area of interest is in further documentation of the robot's capabilities, creating protocols for maximizing its efficacy under certain conditions, and developing a more advanced user interface to decrease the amount of training required to operate the robot.

7 Credit Statement

- Caleb Bush: Generating the Compton imaging algorithm, developing methods to offload data from the system wirelessly for both Compton and isotope identification programs, integrating systems, partial creation of all written and verbal reports.
- Aaron Belman-Wells: Development of robot position finding systems, creation of source localization algorithm, and partial creation of all written and verbal reports.
- Giuliano Basso: Research in source-spectra identification, live N42 communication, logo design, partial creation of all written and verbal reports.
- James Berry: Leading weekly team meetings, making physical changes to the robotic system, assistance with review of existing systems and literature.
- Dr. He: Advising on project goals and reasonable time constraints and access to equipment.

7.1 Acknowledgements

Thank you to the staff of the ORION Lab for their help with programs and access to check sources, and to the NERS 492 instructional team for help presenting results.

8 Generative AI Use Statement

The UM-GPT generative AI was used during this project for the creation and testing of certain code. It was not used during the creation of this report, or any other presentation.

References

[1] IAEA in Austria. “THE RADIOLOGICAL ACCIDENT IN GOIANIA”. In: (1988).

[2] R. M. Vázquez-Cervantes and F. J. Ramirez-Jiménez. “6-Wheel Terrestrial Robot for Radiation Detection”. In: *IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)* (2017).

[3] U. Ganbaatar et al. *Radiation Detection Robot Final Report*. 2022.

[4] Inc. H3D. *M400 Spec Sheet*. 2024.

[5] *American National Standard Performance Criteria for Handheld Instruments for the Detection and Identification of Radionuclides - Amendment 1*. 10 Nov. 2020.

[6] Inc. H3D. *H3D Detector APIs Version 1.2*. 2022.

[7] Glenn Knoll. *Radiation Detectibn and Measurement 3rd Edition*. John Wiley Sons, Inc., 2000.

[8] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[9] “Table of Nuclides - Nuclear structure and decay data”. In: (2019).

[10] Gilbert Strang and Edwin Herman. *Calculus Volume I*. Rice University, 2020.

9 Appendix

9.1 Gantt Chart

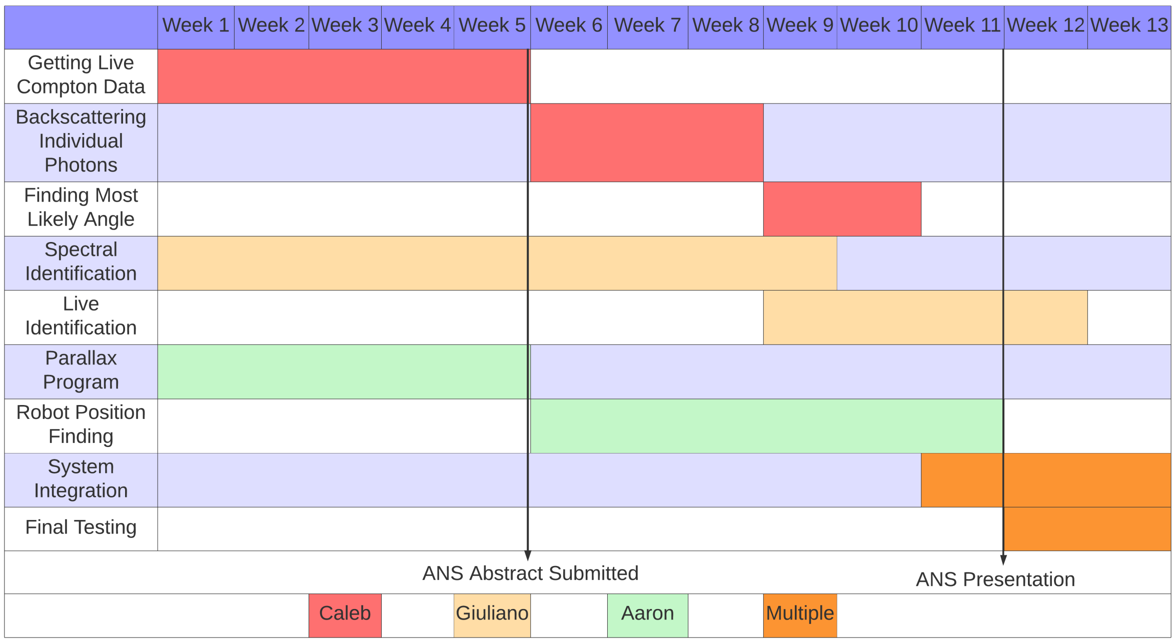
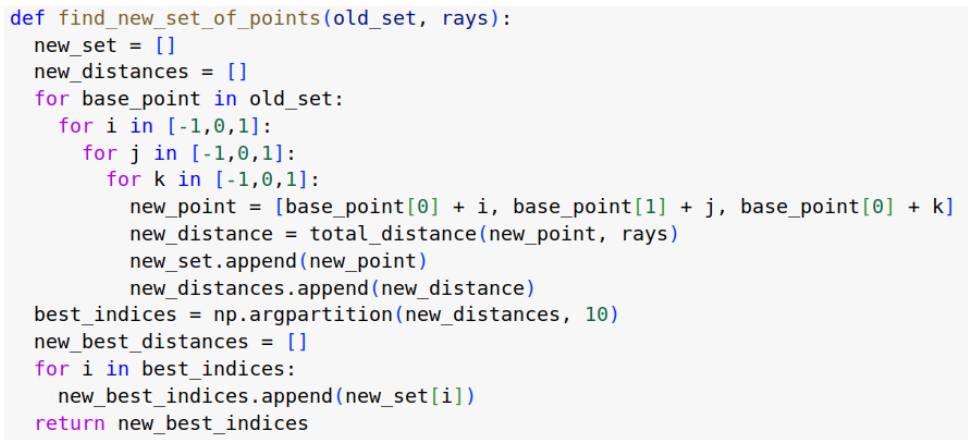


Figure 14: Schedule of Work Done

9.2 Code

```python
def single_line_distance(point, ray):
  if np.dot(point[3:6], ray[3:6]) > 0:
    base_to_point = (ray[0]-point[0], ray[1]-point[1], ray[2]-point[2])
    unit_vector_direction = (ray[3], ray[4], ray[5])
    cross_product = np.cross(base_to_point, unit_vector_direction)
    distance = math.sqrt(sum(pow(element, 2) for element in cross_product))
    return distance
  else:
    distance = math.sqrt(sum(pow(point[i] - ray[i]) for i in range(0, 3)))
    return distance
```

Figure 15: Minimum Distance Between a Point and a Ray



```python
def find_new_set_of_points(old_set, rays):
  new_set = []
  new_distances = []
  for base_point in old_set:
    for i in [-1,0,1]:
      for j in [-1,0,1]:
        for k in [-1,0,1]:
          new_point = [base_point[0] + i, base_point[1] + j, base_point[0] + k]
          new_distance = total_distance(new_point, rays)
          new_set.append(new_point)
          new_distances.append(new_distance)
  best_indices = np.argpartition(new_distances, 10)
  new_best_distances = []
  for i in best_indices:
    new_best_indices.append(new_set[i])
  return new_best_indices
```

Figure 16: Method for Finding Best Elements in Neighborhoods

All code used in the project can be found here: https://drive.google.com/drive/folders/ 1xsWQEc-NDJkDLQREcpcrIX2RolPC2qC4O?usp=sharing.

9.3 Cone-Sphere Intersection formula

If $\overline{v}$ = A normalized direction vector with an origin at the center of a sphere and θ = the angle of a cone around that vector, the cone intersects the unit sphere in Cartesian coordinates at:

$$\gamma = \arccos(\overline{v}_z)$$

$$\arctan(\frac{\overline{v}_y}{\overline{v}_x})$$

$$x = (\sin(\theta) * \cos(\beta) * cos(\gamma)) * \cos(t) + (\sin(\theta) * \sin(\gamma)) * \sin(t) - (\cos(\theta) * \sin(\beta) * \cos(\gamma))$$

$$y = -(\sin(\theta) * \cos() * \sin(\gamma)) * \cos(t) + (\sin(\theta) * \cos(\gamma)) * \sin(t) + (\cos(\theta) * \sin(\beta) * \sin(\gamma))$$

$$z = (\sin(\theta) * \sin(\beta)) * \cos(t) + \cos(\theta) * \cos(\beta)$$

Where t represents all values between 0 and 2 π