Title

Enhanced use of contextual data for quantitative gamma-ray imaging in nuclear safeguards applications

Author

Knecht, Kalie

Publication Date

2024

Enhanced Use of Contextual Data for Quantitative Gamma-Ray Imaging in Nuclear Safeguards Applications

By

Kalie Knecht

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Nuclear Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Kai Vetter, Chair
Professor Lee Bernstein
Professor Karl van Bibber

Spring 2024

Enhanced Use of Contextual Data for Quantitative Gamma-Ray Imaging in Nuclear Safeguards Applications

Abstract

Enhanced Use of Contextual Data for Quantitative Gamma-Ray Imaging in Nuclear Safeguards Applications

by

Kalie Knecht

Doctor of Philosophy in Engineering - Nuclear Engineering

University of California, Berkeley

Professor Kai Vetter, Chair

Nuclear safeguards inspections aim to verify nuclear materials at facilities around the world, and are faced with constraints such as a limited amount of time to perform measurements, ideally with minimal disruption to facility operation. Gamma-ray imaging is a promising technology for performing these inspections because it presents an efficient and non-destructive method of quantifying nuclear material. Recent advances in gamma-ray imaging have included integrating mobile radiation imagers with real-time tracking and scene reconstruction algorithms, enabling a mobile mode of operation and 3D localization of gamma-ray sources. This technique, dubbed scene data fusion (SDF), holds promise for facilitating safeguards inspections. While developed for free-moving measurements, SDF also has applications in cases that require static measurements for increased counting statistics. In these applications, the scene information can be used to improve quantitative imaging results. This dissertation employs computer vision techniques with the scene information collected by SDF-enabled imagers to identify objects of radiological interest. Computational methods are developed to determine optimal measurement positions to reduce measurement time while still capturing enough data to quantify nuclear materials within desired statistical uncertainties. The computation techniques are experimentally validated with a series of example models and a safeguards-relevant measurement. The result is a software tool that could enable safeguards inspectors to rapidly formulate a measurement plan in the field and provide a quantitative measurement of materials present with low levels of uncertainty.

To Michael, Fermi, and Denna

Contents

List of Figures

List of Tables

Acknowledgments

While I worked very hard for this degree, it would be grossly inaccurate for me to say that my hard work alone has allowed me to finish this dissertation. I would run out of paper before I got a chance to acknowledge everyone who has enabled my success in this work, but I would like to acknowledge a few of the people who have helped me along the way.

First I would like to thank my qualifying exam committee: Dr. Lee Bernstein, Dr. Karl van Bibber, Dr. Fernando Pérez, and Dr. Kai Vetter. Your insights were very valuable in forming the basis for my dissertation work. I would like to additionally thank Karl, Lee, and Kai for serving on my dissertation committee and providing helpful comments during the writing of this dissertation. I would like to thank Kai in particular for continuing to provide excellent ideas and for advocating for his graduate students.

I would like to thank Dr. Dan Hellfeld for his mentorship during the first half of my PhD, where he equipped me with all of the tools I needed to succeed. I could not have asked for a better teacher as a new graduate student. I would also like to thank Dr. Mark Bandstra for mentoring me during the second half of my PhD. Mark's technical knowledge and support were invaluable to me in completing this dissertation.

I would like to express my gratitude to the Nuclear Science and Security Consortium for supporting me through graduate school, which enabled me to pursue this research and complete my dissertation. Their investment in my academic and professional development (including connecting me to great mentors like Dan and Mark) has been instrumental in the successful completion of this dissertation.

There have been a number of scientists affiliated with LBNL's Applied Nuclear Physics Program, past and present, who have been invaluable resources to me during my dissertation work. I would like to specifically thank Dr. Tenzing H.Y. Joshi and Dr. Micah Folsom for their ideas which inspired my work. I would like to thank Dr. Don Gunter for his mathematical expertise and enthusiasm for working with students. I want to thank Dr. Marco Salathe for the countless questions he has answered for me. Additionally I would like to thank Dr. Brian Quiter, Dr. Ren Cooper, Dr. Jayson Vavrek, Xin Chen, Kushant Patel, Joey Curtis, and Dr. Ryan Pavlovsky for their help over the years.

Thank you to the members of my research group, past and present, for their help and support, including Dr. Ali Hanks, Dr. Matt Marshall, Dr. Jake Hecla, Dr. Emily Frame, Dr. Yifan Zheng, Tyler Jordan, Jisu Park, Dr. Rebecca Krentz-Wee, Robin Peter, Jaewon Lee, Matthew Verlie, Chris Lamb, and Dr. Grey Batie. I would like to especially thank Grey for taking the time to help me decide which graduate school to go to over 5 years ago, always providing me with support and nuanced opinions, and letting me piggyback off of distributed source experiments.

I also greatly appreciate all of the friends I've made during my time at UC Berkeley. You all have been an excellent source of stress relief and laughs. I want to specifically thank Matthew and Yessie for providing me with literal physical support. I also want to thank Carla, Andrew, Chai, and Jaewon for being great hosts and occasionally pet-sitting my dog while I've travelled to conferences. I especially thank Vanessa, Jacob, Alex, and Borrello for continuing to advocate for graduate workers. Lastly, I'm thankful to Abigail Cooper, Rachel Williams, and Dresden Jones, who have continuously supported me throughout my academic journey.

I would like to recognize my mentors through the years whose insight helped make me into the scientist I am today: Dr. Scott Stewart, Dr. Steve Skutnik, Dr. Jamie Coble, Price Collins, Dr. Caleb Roecker, Dr. Maulik Patel, Dr. Caitlin (Taylor) Kohnert, and Dr. Acacia Brunett. I want to thank my high school physics teacher, Janet Richardson, for teaching us about basic nuclear reactions and encouraging me to pursue a bachelor's degree in nuclear engineering at UTK. I also want to thank Julie Ezold for advising me to pursue a PhD.

I want to thank my cat and dog, Denna and Fermi, for being beacons of emotional support who have comfortably and cluelessly napped next to me as I've written this dissertation.

Finally, I want to express my gratitude to Michael for being my unwavering companion throughout every stage of completing this degree. Your support has been a constant source of strength, guiding me through both the best and worst parts of this journey. I look forward to the next phase of our lives being enriched by the hard work we've put into our education over the past decade.

This material is based upon work supported by the Department of Energy National Nuclear Security Administration under Award Numbers DE-NA0003180 and NA0003996.

Chapter 1

Introduction

The goal of this dissertation is to develop a tool that could be used by nuclear safeguards inspectors to identify optimal measurement positions for quantitative gamma-ray imaging to assess nuclear materials. This chapter provides the context of the work completed in this dissertation. It also identifies the gaps in technology that were present at the beginning of this work and outlines the solutions to those gaps that were developed through this dissertation.

1.1 Motivation

The Treaty on the Non-proliferation of Nuclear Weapons (NPT) went into force 1970 with the objective to prevent the spread of nuclear weapons, promote peaceful use of nuclear technology, and achieve global nuclear disarmament. Under this treaty, nuclear weapons states promise to disarm and help countries develop peaceful civilian nuclear programs. Non-nuclear weapons states promise to never develop nuclear weapons, and in exchange receive technology and expertise in peaceful nuclear applications from the nuclear weapons states.

The International Atomic Energy Agency (IAEA) is tasked with ensuring member states are complying with this treaty through monitoring of their nuclear facilities. One method used by the IAEA to ensure treaty compliance is nuclear safeguards inspections. During these inspections, inspectors verify correctness and completeness of a state's declared nuclear materials and activities.

In recent years, the number of countries entering additional safeguards agreements and the amount of material accounted for in safeguards have both been increasing. However, the budget for the IAEA to carry out safeguards inspections is not increasing at the same rate [1]. Additionally these inspections are faced with constraints such as a limited amount of time to perform measurements, ideally with minimal disruption to facility operation.

Many nuclear materials covered by safeguards agreements emit gamma rays which can be measured through gamma-ray imaging. Gamma-ray imaging is a promising technology for performing these inspections because it presents a fast and efficient non-destructive method of quantifying nuclear material.

Recent advances in gamma-ray imaging have included integrating mobile radiation imagers with real-time tracking and scene reconstruction algorithms to enable a mobile mode of operation and 3D localization of gamma-ray sources [2]. This technique, dubbed scene data fusion (SDF), provides promising technology which could facilitate safeguards inspections. This combined with recent advances in robotics and data science presents an opportunity to use new technology to continue performing safeguards inspections in a more efficient manner.

1.2 Existing technology

At the onset of this dissertation scene data fusion was a relatively mature technology that could quickly and accurately localize sources in a variety of applications, including safeguards, emergency response, and nuclear contamination remediation. Work was also being performed to include quantification of source activity from point and distributed sources in addition to localization. While developed for free-moving measurements, SDF also has applications in cases that use static measurements to improve counting statistics. In these cases, the scene information can be used to improve quantitative imaging results.

Although SDF techniques were already quite advanced at the onset of this work, there were improvements that could be made which would be highly useful to a safeguards inspector. First, the contextual information already collected by SDF systems could be further leveraged to improve understanding of the scene which would allow modeling of gamma-ray interactions in materials, leading to more accurate reconstructed gamma-ray activity. Additionally, this contextual information could be used to construct a tool which could tell an operator where to place an imager for the best results during measurements, allowing efficient and precise data collection in a minimally invasive manner.

1.3 Proposed improvements

The goal of this work is to build a tool which further leverages contextual data and proposes optimal static measurement positions for safeguards inspectors to place a radiation imager to quantify nuclear material within desired levels of uncertainty.

To achieve this goal, computer vision techniques are employed with the scene information collected by SDF-enabled imagers to identify objects of radiological interest. Computational techniques are also developed to determine optimal measurement positions to reduce measurement time while still capturing enough data to quantify within desired statistical uncertainties. These methods are then experimentally validated with a safeguards-relevant measurement. The result is a software tool that could be used by safeguards inspectors to rapidly formulate a measurement plan in the field and provide a quantitative measurement of materials present with low levels of uncertainty.

1.4 Dissertation structure

This dissertation is organized as follows:

- Chapter 2 provides necessary introductory concepts including the basics of scene data fusion and information about contextual data and gamma-ray imaging.
- Chapter 3 discusses new ways contextual data has been leveraged in this dissertation to improve scene data fusion methods.
- Chapter 4 describes the computation techniques developed to identify optimal static measurement positions for a quantitative gamma-ray imaging measurement.
- Chapter 5 demonstrates the performance of the optimization techniques with a variety of example models.
- Chapter 6 applies the optimization techniques contextual data improvements to generate a 3D gamma-ray image with data collected in a distributed source measurement.
- Chapter 7 draws conclusions of the work and highlights relevant applications and future outlooks.

Chapter 2

Background

This chapter provides an overview of a 3D gamma-ray imaging technique called Scene Data Fusion. The concepts discussed in this chapter provide the necessary technical foundation for understanding the work completed in this dissertation in the following chapters.

2.1 Scene data fusion

Portable detection systems can be equipped with contextual sensors to allow for free-moving 3D gamma-ray imaging in a method called scene data fusion (SDF) [2], [3]. SDF presents a non-destructive means of analyzing nuclear material present in a variety of environments. SDF captures relevant scene data including a 3D map of an area, which can be used to inform operator decisions and enhance quantitative imaging results. The scene information can be leveraged to determine optimal measurement positions to reduce measurement time while still capturing enough data to quantify to desired statistical uncertainties.

SDF is useful for a variety of applications, such as contamination mapping, safeguards, and source search. Figure 2.1 shows 3D gamma-ray images generated with SDF for these applications. The image on the left shows a 3D Compton image of the 662 keV gamma rays in a measurement collected by walking around the bumper cars at the Pripyat amusement park in the Chernobyl Exclusion zone. The image shows Cs-137 contamination on the lower sides of the elevated platform due to the washout of contamination from rainwater [4]. The middle imageshows a measurement collected at the Paducah Gaseous Diffusion Plant in Kentucky, with three 30A cylinders, where one is 1/2 - 2/3 full of depleted UF_6 (DUF_6) [5]. The 3D Compton image of the 1001 keV gamma rays identifies the tank which contains DUF_6. The image on the right shows a Cs-137 point source inside a building which has been measured with an SDF system deployed on a unmanned aerial vehicle (UAV) where the source is localized through 3D Compton imaging on the 662 keV gamma rays.

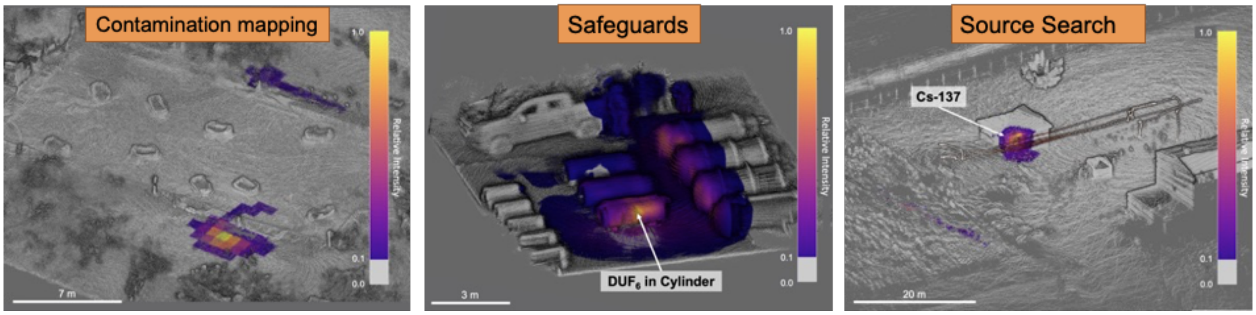


Figure 2.1: Gamma-ray imaging reconstructions for a variety of applications generated with scene data fusion

2.1.1 Current scene data fusion capabilities

SDF is currently deployed in hand carried, vehicle carried, and drone carried platforms. SDF has been extensively demonstrated in source search and non-quantitative mapping scenarios [2], [3], [6], [5], [7]. Recently methods have been developed to quantify these measurements in point and distributed source environments [8]. These tools present a way to evaluate the amount of material present, which is needed for safeguards applications. The scene information gathered by the system can be further leveraged to enhance imaging capabilities, especially in cases requiring assessment of weak radiological signatures.

2.1.2 H420-LAMP

One SDF-enabled imager that was used for the work in this dissertation is the H420-LAMP imaging system, shown in Figure 2.2. The H420-LAMP system combines nuclear radiation detection with other non-nuclear modalities to provide a spatially-contextual picture of the location of a radiation source in a real-world setting. The device consists of the commercially available H420 CdZnTe (CZT) detector, designed by H3D as a 2D static imager capable of coded-aperture and Compton imaging [9], labeled in blue in the figure. The imager has been retrofitted with the Localization and Mapping Platform (LAMP) developed at Lawrence Berkeley National Laboratory [10], which is labeled in red and includes a Intel RealSense D455 depth camera, Advanced Navigation Spatial inertial measurement unit (IMU), and a Velodyne 16-beam Puck LiDAR. The device also includes an Intel NUC single-board onboard computer to perform Simultaneous Localization and Mapping (SLAM) with Google Cartographer [11], enabling the system to estimate its position and orientation and generate a 3D map of its surroundings. For generating the 3D map, the system is hand-carried through the scene. For point-and-dwell static measurements, the imager is placed on a tripod. The entire system is powered with a single 98 W-hr Li-ion battery [12].

The H420 consists of a 2 × 2 array of 2.2 × 2.2 × 1 cm^3 CZT crystals separated by a 0.5 mm gap with a 0.5 mm guard ring at the edge of the detector, giving an active detector

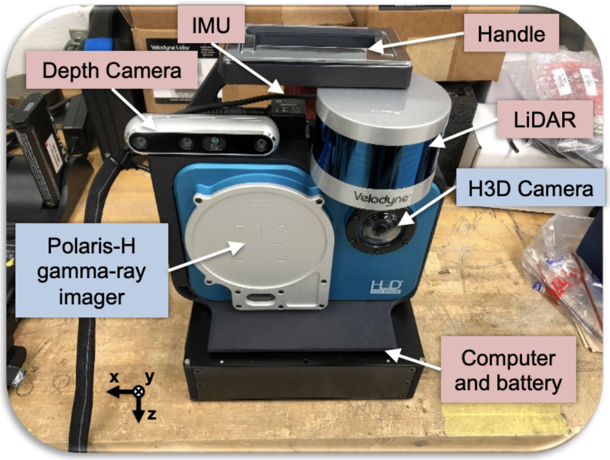


Figure 2.2: The H420-LAMP imaging system and its coordinate system

volume of greater than 19 cm^3. The H420 also has a 1.5 mm thick rank-19 tungsten modified uniformly redundant array (MURA) mask to enable coded-aperture imaging [13]. The subpixel and depth resolution of the detector is around 0.5 mm. The energy resolution is better than 1% FWHM at 662 keV [14]. The Compton response of the H420 was measured with a 4π scanning system with Ba-133 (356 keV), Cs-137 (662 keV), and Co-60 (1173 and 1332 keV). Figure 2.3 shows the Compton response at 662 keV, with 122 response measurements overlaid. The center of the image corresponds to the direction directly in front of the detector [12].

The IMU measures the orientation of the imaging system as it moves through the scene while the LiDAR collects laser beam scans. These data are fed into a SLAM algorithm to construct a 3D map of the environment and reconstruct the detector position within that map. The RealSense depth camera collects Red-Green-Blue (RGB) and depth image data (referred to as RGB-D data) at a rate of 30 Hz, providing information rich visual camera data.

2.1.2.1 Device operation

The H420-LAMP has a WiFi dongle which allows the user to remotely connect to the detector and access the web page user interface (UI) with a computer, phone, or tablet [15]. Figure 2.4 shows a screenshot of the UI as it collects data. The UI has a 'start' button, which turns on all of the sensors and begins data collection. After the sensors are turned on, there is a real time display of the depth camera view and a top-down view of the real-time SLAM map reconstruction. As counts are collected, a "heat-map" of reconstructed source activity is overlaid on the camera data.

While previous SDF systems are designed to be continuously moved through the mea-

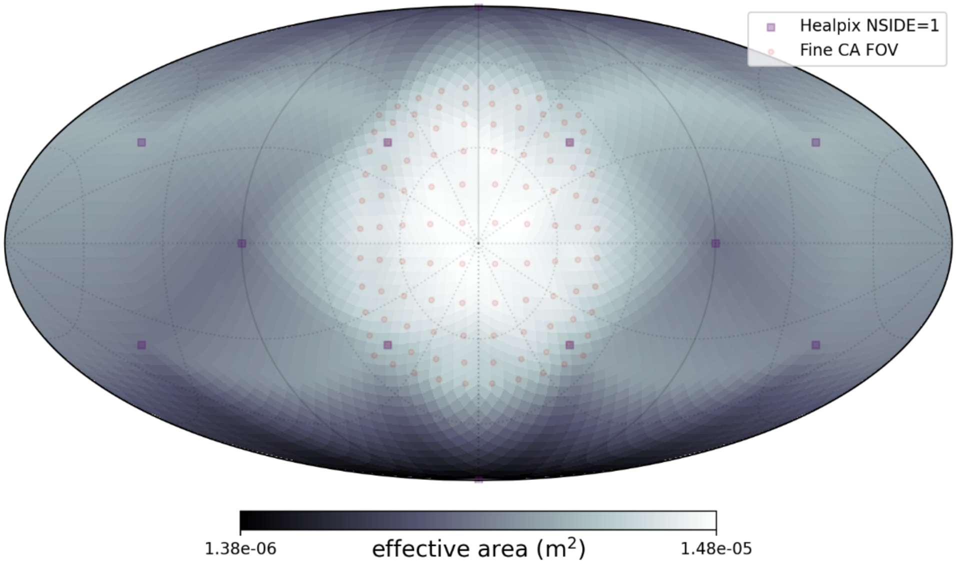


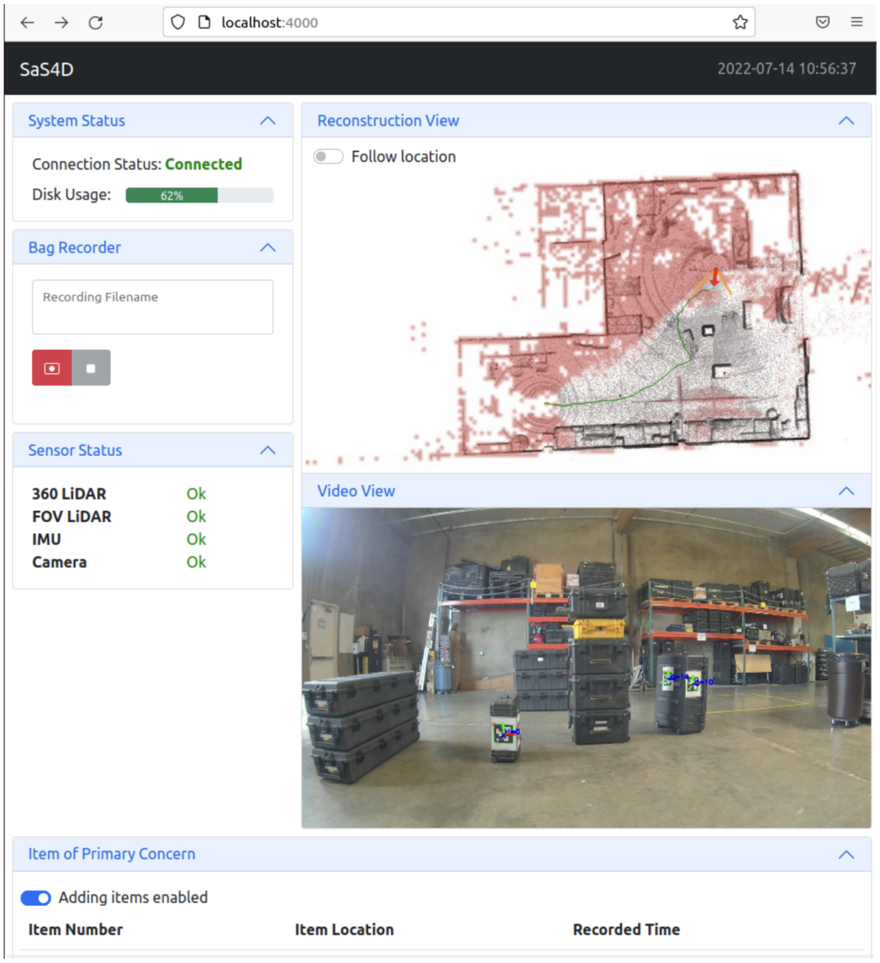
Figure 2.3: Mollweide projection of the 662 keV Compton response of the H420-LAMP interpolated onto a NSIDE=16 HEALPix grid with a 5° Gaussian smoothing applied [12]

surement, the H420-LAMP system is designed for safeguards applications which require quantification with low levels of uncertainty. In these scenarios, it is desirable to operate the imager in a static mode to collect sufficient counting statistics while keeping personnel dose low. This means that the system collects radiation data from several static measurement positions to construct a 3D image. However, a global map is still built by continuously moving through the scene with the device to construct a 3D map of the scene with the contextual sensors.

For point-and-dwell measurements, there is a separate data acquisition mode which turns all of the sensors on at first, but turns off the contextual sensors after 30 seconds of data collection, assuming the detector and its environment will be at rest during the radiation counts. The contextual data collected during the 30 second measurement can be used to rotate and translate the static measurement data into the frame of the global map which is collected during the initial scene survey. Contextual data are not collected for the entire duration of the static measurement because the information rich scene data requires tens of gigabytes of computer storage space for a 20 minute measurement, and during long measurement campaigns the onboard computer can run out of storage space.

2.1.3 Emesent Hovermap

While the H420-LAMP is a relatively compact hand-carried system, there are some confined spaces where the entire H420 system is too bulky to fit. Additionally, the resolution capabil-



Connection Status: **Connected**

Disk Usage: 62%

360 LiDAR	Ok
FOV LiDAR	Ok
IMU	Ok
Camera	Ok

Item Number	**Item Location**	**Recorded Time**

Figure 2.4: Screenshot of LAMP user interface [15]

ities of the Velodyne Puck LiDAR are not advanced enough to reconstruct scenes with a fine level of detail. To address these drawbacks, the Emesent Hovermap ST-X LiDAR system [16] is sometimes used to supplement the information collected by the H420-LAMP.

The Emesent Hovermap ST-X is a compact hand-carried system with a rotating puck-type 32-beam LiDAR. It is capable of generating LiDAR maps at a range from 0.5 to 300 m with a precision of ±10 mm in typical indoor and underground environments. The precision worsens to ±15 mm in general environments [16]. Due to its small size, it can easily fit into confined places.

2.1.4 Improvements to scene data fusion

The imaging techniques in scene data fusion are sophisticated, but they could be improved by leveraging data that are collected by the contextual sensors. In this work, the H420-LAMP is walked around the scene where the LAMP package captures and reconstructs scene data to create a 3D map of the scene which serves as the imaging space. The contextual 3D map of the scene is leveraged to improve gamma-ray imaging methods in two ways. First, computer vision techniques are employed to identify and label discrete objects in the scene. If computer vision techniques can identify types of material in a scene, then attenuation from those materials can be estimated to properly quantify nuclear materials present. Some work has already been done to estimate attenuation of intervening materials in SDF applications [17], [18], [19], but contextual data could improve those estimates. Second, a multi-objective optimization is performed on the 3D map to identify optimal measurement positions for quantifying activity from a gamma-ray source. Current SDF techniques rely on user knowledge and understanding of radiation signatures to perform an efficient quantitative measurement. This work seeks to develop a computational tool which can automatically perform efficient measurement planning.

2.2 Contextual data

The data collected by SDF-enabled imagers can be divided into two categories: radiological and contextual. Contextual data are data which are collected by the auxiliary sensors equipped to gamma-ray imagers, including the LiDAR, IMU, and depth camera. Radiological data are that which are collected by the radiation imager.

2.2.1 Simultaneous localization and mapping (SLAM)

The raw data collected by the contextual sensors must first be processed with SLAM to generate the 3D map of the environment and the position history of the imaging system as it moved through that 3D map. SLAM algorithms are widely used in robot navigation, robotic mapping, and virtual or augmented reality.

2.2.1.1 Google Cartographer

Google Cartographer is an open-source software developed by Google to perform SLAM in 2D and 3D across multiple platforms and sensor configurations [11]. Cartographer consists of two related subsystems: local SLAM and global SLAM. Local SLAM builds a succession of sub-maps as a sensor system moves through a scene. Global SLAM runs in the background to match sub-maps together and build a global map of a scene, while also solving for the direction of gravity. Figure 2.5 shows a high level overview of Google Cartographer. The output of Cartographer is a LiDAR point cloud and a time-stamped trajectory of the system within the coordinate frame of the point cloud.

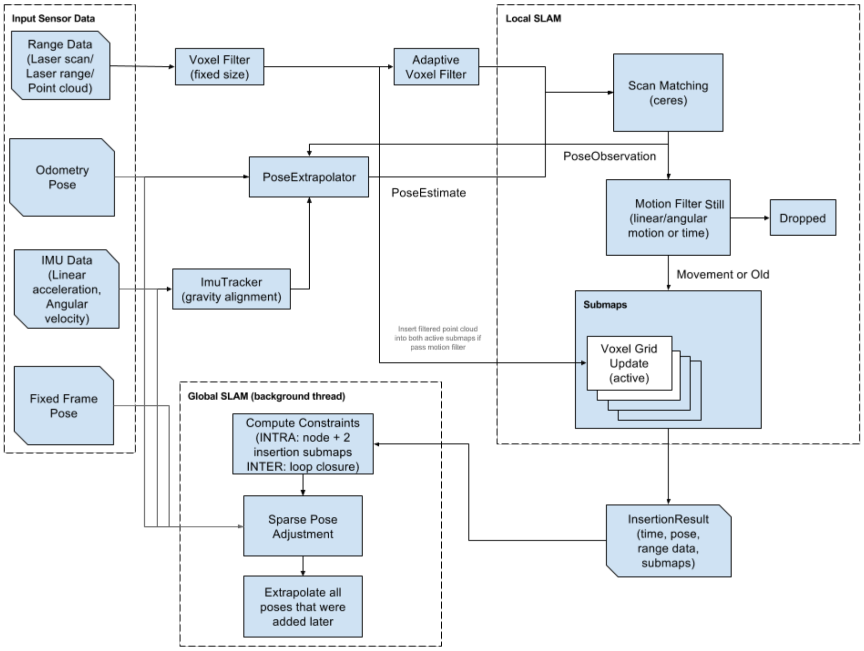


Figure 2.5: High level system overview of Google Cartographer from [11]

2.2.2 Point clouds

Point clouds are used to represent a 3D scene. A point cloud consists of many x-y-z points which describe the surfaces present in a scene. In addition to x-y-z coordinates, each point may also have additional data such as color, represented with RGB values. Figure 2.6 shows an example point cloud of a laboratory scene. The walking path of the contextual sensors that was used to collect data for building the map is shown in red. The zoomed inset shows some of the individual x-y-z points that comprise the column in the middle of the room. The LiDAR point cloud only renders surfaces of objects, so interiors of objects are assumed to be empty.

In this work, a point cloud is created by taking an initial mapping scan of a measurement scene. This mapping can be done with the sensors on the H420 LAMP or with a more advanced sensor, like the Emesent Hovermap. This point cloud serves as the global map of the scene and is then analyzed to gain information about the measurement scene prior to collecting radiation data.

2.2.2.1 Point cloud registration

In this work, a global map of a scene is constructed through an initial free-moving survey. Then, the imager is placed at static positions for longer dwell measurements, where 30 seconds of contextual data are collected. To reconstruct radiation data collected at these static

Figure 2.6: Point cloud of laboratory scene with zoomed inset showing individual point cloud points which comprise the column in the scene

positions into the reference frame of the global cloud, a transformation must be found between the contextual data collected at the static position and the global point cloud. Point cloud registration is a computation problem which aims to identify a spatial transformation (scaling, rotation and translation) which can rotate and translate one point cloud into the coordinate frame of another.

Iterative closest point (ICP) is one computational method that can be used to perform point cloud registration. This method keeps the reference point cloud static, while iterating over potential transformations for the second point cloud to find the best match between points of the two clouds. Open3D ICP tools are used to perform point cloud registration in this work [20].

2.2.3 Voxel grids

A voxel grid is a regular grid in 3D space, analogous to a pixel grid in a 2D space. To construct a voxel grid, a desired number of voxels or voxel pitch is specified to construct a uniform x-y-z grid. A voxel grid can be constructed with a point cloud as an input. Each voxel in a voxel grid will be considered to be occupied or unoccupied, based on the number of point cloud points which reside inside of it. The number of points inside a voxel for it to be considered occupied can be as low as 1, but it may be desirable to raise the threshold to a higher value in dense or noisy point clouds. To accelerate computation, voxel grids in this work are sparsely represented with only the occupied voxels.

2.2.4 Trajectory data

Trajectory data are represented by a series of time-stamped trajectory nodes. Each trajectory node has an x-y-z location in the coordinate frame of the 3D map as well as an orientation described by a rotation matrix, quaternion, or angles around the x-, y-, and z-axes (referred to as roll, pitch, and yaw respectively). The radiation data collected by an SDF system is also time-stamped, consistent with the trajectory, so individual radiation events can be tied to a trajectory node or an interpolation between two nodes.

2.3 Gamma-ray imaging

Radioactive elements have unique gamma-ray signatures which can be used to locate or characterize radioactive sources in our environment. When we use the radiation signature emitted by a radioactive source to determine the nature of the source and where it is located, we call it radiation imaging. This concept is leveraged in medical imaging techniques like positron emission tomography (PET), where a positron-emitting isotope is injected intravenously or applied to a patient to image internal organs. Gamma-ray imaging is an non-destructive analysis method which aims to quantify the types and amounts of nuclear material present in a scene.

2.3.1 Gamma-ray interaction mechanics

Gamma-ray imaging relies on information obtained by radiation detectors from gamma-ray interactions to reconstruct radiation events. A gamma ray interacts with matter through three key interactions of relevance for gamma-ray imaging: the photoelectric effect, Compton scattering, and pair production. In the photoelectric effect, the gamma ray is fully absorbed by an atom which ejects an energetic photoelectron. In Compton scattering, the gamma ray scatters off of an electron in the material, where the gamma ray is deflected at an angle and deposits some energy to the electron. Pair production occurs in the Coulomb field of a nucleus where the gamma-ray energy is converted into an electron-positron pair. This interaction occurs at high energies, requiring at least an energy equal to two times the rest mass of an electron, 1.022 MeV. The excess energy above the 1.022 MeV threshold is converted to kinetic energy shared by the electron-positron pair. The likelihood of each of these interactions to occur is a function of the energy of the incident gamma ray and the atomic number of the material interacted with. Figure 2.7 shows the dominant interaction for a material of a given atomic number (Z) and gamma ray of energy hν [21].

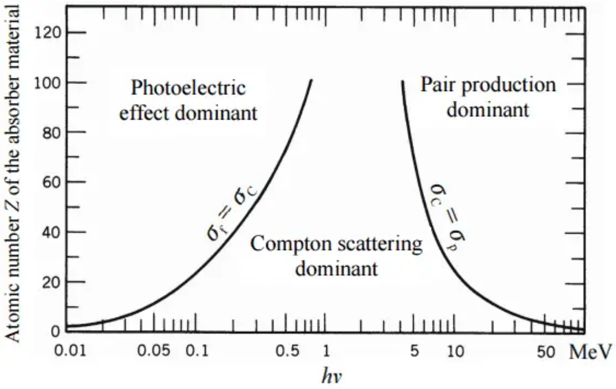


Figure 2.7: The dominant interaction processes for a gamma ray with energy hν in a material with atomic number Z [21]

2.3.2 Gamma-ray imaging modalities[1]

There are a variety of techniques to image gamma rays, which can be broadly divided into two categories: collimator-based and collimator-less. Collimator-based techniques rely on attenuation to constrain the relationship between detector pixel and imaging space (e.g., voxels). Collimator-less techniques typically rely on information about the energy and position of gamma-ray interactions to identify the subset of imaging space from which the gamma ray could have originated.

Basic collimator-based imaging techniques such as pinhole and parallel hole imaging allow for simple image reconstruction with good angular resolution. However, due to the high level of collimation, the efficiency and field-of-view (FOV) of these techniques are very limited, so they are typically only deployed in high count-rate scenarios. Safeguards measurements often concern low activity sources, so these inefficient collimator-based imaging methods would require a long measurement time to reach the low level of uncertainty required.

Coded-aperture and Compton imaging are both more advanced, higher efficiency imaging techniques which leverage information from the gamma-ray interactions previously described. Coded-aperture imaging is a collimator-based technique which retains the angular resolution of more basic techniques while increasing efficiency. Compton imaging is a collimator-less technique which relies on the kinematics of Compton scattering. The H420-LAMP system offers both coded-aperture and Compton imaging modalities. In combination, these imaging modalities provide a powerful ability to localize and map sources across a wide range of gamma-ray energies and activities.

The formulations for coded-aperture and Compton imaging which follow assume a far-field imaging approximation, which refers to the imager being operated in distances much greater than the largest dimension of the imager, which is around 5 cm for the H420. The

---

[1] Much of the text in this section is published in [5].

far-field approximation allows for the detector to be treated as a single point, allowing solid-angle effects of the finite detector size to be ignored. In this work, source-detector distance is always at least 1 m, ensuring the far-field approximation is accurate.

2.3.2.1 Coded-aperture imaging

Coded-aperture imaging is an imaging technique that leverages the photoelectric effect; therefore it is suited to relatively low energy gamma rays (<350 keV for the H420-LAMP). Coded-aperture imaging relies on the signal modulation imposed by distance and a coded mask created using an array of open and attenuating elements in front of a position-sensitive detector. This method offers high angular resolution, though at the cost of reconstruction complexity, limited FOV, sensitivity and proneness to artifacts. For the H420-LAMP, the FOV is limited to $86^\circ \times 86^\circ$.

Coded masks are typically square or hexagonal grids with open and filled elements ("binary masks"). Each open mask element acts as a pinhole, projecting an image of the scene onto the detector behind it. The detected image is therefore a number of overlapping projections of the scene. The primary difficulty posed by this is generating mask designs which result in the most clearly uncorrelated projections, and algorithms which can perform this reconstruction efficiently and without introducing artifacts. Though coded-aperture imaging approaches offer a compact, efficient means of imaging gamma rays, they do so at the cost of reconstruction complexity when compared to basic collimator-based techniques like pinhole and parallel hole imaging.

For a perfectly efficient detector with pixels (indexed k, l), 2D image formation can be represented by the following equation:

$$D_{kl} = \sum_{ij} S_{ij} A_{i+k,j+l} + B \tag{2.1}$$

where A represents the binary mask array, B contains all background terms (including electronic noise, gamma rays transmitted through the mask, etc.), D is the detected pattern on the detector face [22] and S represents the source distribution.

This can be reformulated into a 3D problem by assuming:

- a series of observations denoted by time-interval Δt,
- a detector with an orientation of $\vec{v}$ observing a source along the ray defined by $\hat{r}_s$,
- an angular detector (2D) response characterized across 4π by the function $\gamma_{kl}(\vec{v}, \hat{r}_s)$ (far-field assumption),
- detector and voxel centers defined by x and x_s,
- voxel activity Λ_s,
- background dependent on pixel index k, l and time, t.

This gives the following expression for rate in each pixel for each time bin:

$$D_{klt} = \sum_{s=1}^{N_{vox}} \frac{\gamma_{kl}(\vec{v}, \hat{r}_s)\Lambda_s \Delta t}{4\pi\|\vec{x}_s - \vec{x}\|^2} + B_{klt}\Delta t\,, \tag{2.2}$$

where attenuation has been neglected.

2.3.2.2 Compton imaging

Compton imaging offers a means of imaging sources with characteristic gamma-ray energies above a few hundred keV (>350 keV for the H420-LAMP). This imaging modality relies on multiple interaction events within the detector volume. It is possible to consider events comprised of multiple Compton scatters which terminate in a photoelectric absorption, but in this analysis we consider only double interaction events, assuming one Compton scatter and one photoelectric absorption.

Compton imaging requires the 3D positions of the first two interactions associated with one gamma-ray event, which form the symmetry axis of a Compton cone. It also requires the energy of the incident gamma ray and the energy deposited in the first interaction, which define the opening angle of the Compton cone. Assuming that the incident gamma ray scatters off of an unbound, stationary electron in the Compton scattering interaction, the opening angle can be calculated with the following equation:

$$\cos\theta = 1 + m_e c^2 \left(\frac{1}{E_\gamma} - \frac{1}{E'_\gamma}\right)\,, \tag{2.3}$$

in which:

- θ is the Compton cone opening angle
- m_e is rest mass of an electron
- c is the speed of light
- E_γ is incident gamma-ray energy
- E'_γ is outgoing gamma-ray energy

The incident gamma-ray energy is assumed to be the total energy of the interactions that comprise an event. Spectroscopic imaging uses the events whose incident energy is within a window around a feature of interest, most often a full-energy peak associated with a specific radionuclide. Each cone represents all possible incident directions for a specific gamma ray, and the accumulation of multiple cones can reconstruct the 2D projection of the gamma rays [6].

Compton imaging has lower efficiency than coded-aperture imaging due to its restrictive event selection. However, the angular resolution is often much higher due to the enhanced directional information provided by Compton kinematics. Compton imaging also offers an omnidirectional FOV.

The depth sensitivity of the H420 improves Compton imaging by reducing the uncertainty in the lever arm (distance between two interaction positions). Events are selected which have a lever arm above 2.1 mm to reduce uncertainty in the determination of the scatter axis and remove events with poor angular resolution [6]. While improving angular resolution, this lever arm cut also reduces Compton imaging efficiency.

In this analysis we consider only double interaction events. The sequence of the two interactions can be ambiguous at some energies. The interaction with the higher energy deposition is often sequenced first, except in physically impossible sequences (e.g., interaction with higher energy deposition is beyond Compton edge) [23] [24]. It is also possible to use the Klein-Nishina cross section to choose the sequence with the more likely probability [23] [25]. These methods can result in some cones being incorrectly sequenced, removing accurate cones which can be used to reconstruct source location. It is also possible to use both scattering sequences and to weight each cone based on the likelihood of the scattering sequence it is generated from [26]. One method is to give a higher weight to the cone generated from the sequence where the first interaction has a higher energy deposition. It is also possible to apply a weight based on the Klein-Nishina and photoelectric effect cross sections of the interactions. In this analysis both scattering sequences are considered and no weighting is applied to the cones.

3D Compton imaging can be achieved by performing static measurements, either by using the parallax provided by the extent of the detectors from a single position or by moving the detector to different locations in the scene. To perform these reconstructions, the position and orientation of the detector at each measurement must be known to properly rotate and translate Compton cones. It is also possible to accomplish 3D imaging while freely moving the system throughout the scene, but this requires some automatic and real time solution (i.e., SLAM) to know the position and orientation of the detector during each gamma-ray event [2], [6].

3D image reconstruction speed and noise reduction can be improved by constraining reconstructions to intersections of Compton cones with surfaces present in the scene. Each occupied voxel in the scene is treated as a spherical Gaussian volume with a standard deviation calculated as the product of the voxel size and $\sqrt{\pi/2}$. This formulation keeps the volume of the spherical Gaussian the same as the cubic voxel. To account for uncertainties in the Compton cone axis and opening angle caused by uncertainties in energy deposition and event position, the Compton cones are also given a width described with a Gaussian distribution. The Compton cone width can be calculated on a cone by cone basis, but in this analysis each cone is given a constant conservative value of 14°, which is an experimentally-determined value on the order of the angular resolution of the Compton imager used in this work. The cone-voxel intersection is performed by computing the overlap of the two Gaussian distributions in 3D[2].

---

[2] The computation is detailed in forthcoming LBNL Technical Report 2001559 by D. Gunter *et. al.*

2.3.3 Maximum-likelihood expectation-maximization

Maximum-likelihood expectation-maximization (MLEM) is an efficient means of calculating the maximum likelihood estimate of a function from incomplete data. In practice, this algorithm yields high quality reconstructions of 2D and 3D scenes, but at the cost of high computational complexity and propensity to create high-spatial-frequency artifacts [27].

This algorithm is used for image reconstruction for all imaging modalities on the H420-LAMP. The derivation of the MLEM algorithm can be found in [28, 29, 30], while the list-mode interpretation [31] is given by:

$$\lambda_m^{l+1} = \frac{\lambda_m^l}{\sigma_m} \sum_n \frac{t_{nm}}{\sum_k t_{nk} \lambda_k^l + b_n} , \tag{2.4}$$

in which:

- σ_m represents the sensitivity (chance that a given event, indexed by m, is detected at all)
- λ_m^l represents the image reconstruction for a given iteration, l
- t_{nm} is the 3D response function sampled on a per-event basis such that it represents the probability that an event emitted in image bin (pixel/voxel) m is detected as event n
- b_n is a background factor for each event

In the case of a detector with a relatively low count rate and a high scene update rate, the list-mode implementation is best used, as each measurement at each pose is starved for counts.

Chapter 3

Improvements to contextual data use

Previously the contextual data described in Section 2.2 has been used in Scene Data Fusion to unlock the ability to reconstruct 3D radiation images in free-moving measurements. In some cases, the data collected has been further leveraged to constrain image reconstruction to surfaces present in a scene and also estimate attenuation from air [18], [19]. While this is helpful in accurately reconstructing a scene, there is room for improvement in the use of the contextual data collected by SDF-enabled imagers. Recent advances in machine learning, particularly in computer vision, present advanced methods for analyzing contextual data to improve physics models in radiation image reconstruction. This chapter will discuss two computer vision techniques investigated to further improve SDF methodologies. It will also discuss a ray-tracing algorithm which was developed to inform attenuation estimates through materials.

3.1 3D object detection

3D object detection is a computer vision task which seeks to identify and locate objects in 3D environments based on their shape, location, and orientation. A point cloud represents a scene with a set of x-y-z points, but the relationship of these points to one another is not expressed and must be inferred by the viewer. 3D object detection presents a technique for identifying discrete objects in a point cloud based its x-y-z points. 3D object detection is a tool that can be used to quickly identify objects of safeguards relevance within a 3D point cloud that was previously collected with contextual sensors.

3.1.1 MMDetection3D

MMDetection3D [32] is an open-source object detection toolbox developed by OpenMMLab, a multi-institutional group of software developers that have developed a variety of computer vision tools. It is based on the PyTorch machine learning library [33] which performs tensor computation with GPU acceleration and builds deep neural networks. MMDetection3D is an extension of MMDetection [34], which is an object detection toolbox which uses 2D data (i.e. camera data). The MMDetection3D package has indoor and outdoor object detection architectures available, as well as LiDAR and camera based object detection. Each of these applications has several pre-trained machine learning models available, which means that they can be used without first having to train a model with training data. This tool was selected to perform 3D object detection because it offers a number of pre-trained machine learning models which can easily be switched between to analyze 3D point cloud data.

3.1.2 Datasets

Each of the pre-trained models in MMDetection3D has been trained on a open-source 3D contextual dataset. These datasets consist of contextual data (LiDAR scans, RGB and depth camera videos) which have been annotated with labels that define the bounds of discrete objects and information about each object, such as its orientation and what it is (e.g., desk, chair, etc.). These datasets are used to train and validate a variety of machine learning techniques, but for object detection they are used to identify discrete objects in a scene. The datasets used in this work consist only of indoor scenes.

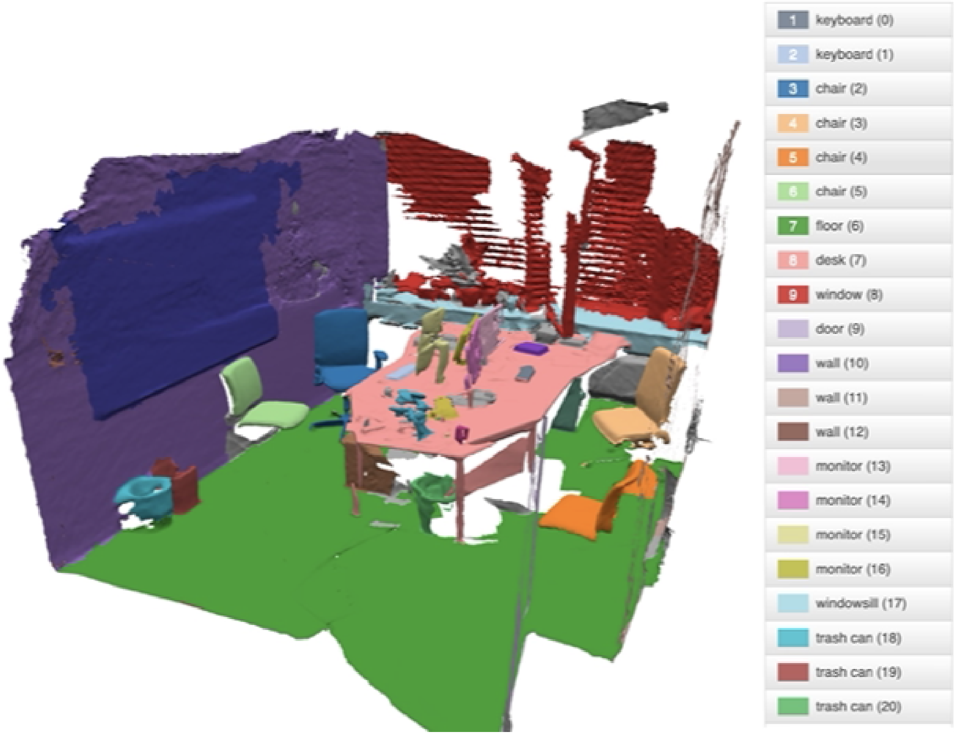


Figure 3.1: Manually annotated scene from ScanNet used for training computer vision models [35]

ScanNet [35] is one of the datasets available which is based on RGB-D video. It contains 2.5 million views in over 1500 scans, which have been annotated with 3D camera poses, surface reconstructions, and instance-level semantic segmentations. The scenes included in the dataset are indoor household or office scenes, and the semantic voxel labels include household objects such as wall, floor, chair, table, desk, and bed. Each scene consists of just a single room. Figure 3.1 shows one of the scenes in the ScanNet dataset with its crowd-sourced semantic annotations. Each instance of a given object is given a separate label; for example, although there are 5 chairs each one has its own label. One could select the label 'chair (5)' to identify only the points which are part of the fifth chair object. Table 3.1 shows all of the labels available in the ScanNet dataset.

Integer	Label
0	floor
1	wall
2	cabinet
3	bed
4	chair
5	sofa
6	table
7	door
8	window
9	bookshelf
10	picture
11	counter
12	desk
13	curtain
14	refrigerator
15	bathtub
16	shower curtain
17	toilet
18	sink
19	other

Table 3.1: ScanNet object labels

SunRGBD [36] is another RGB-D sensor based dataset. The dataset is captured by 4 different sensors and contains 10,000 RGB-D images that are densely annotated, including 146,617 2D polygons and 58,657 3D bounding boxes with accurate object orientations, as well as 3D room layout and category for scenes. This dataset is also only focused on household and office objects, with similar semantic labels to the ScanNet dataset.

The Stanford 3D Indoor Scene Dataset (S3DIS) [37] is the third and final dataset used to train models in MMDetection3D. This is a newer dataset which is composed of larger scenes (i.e. buildings with multiple rooms) but is still only focused on household or office scenes. The dataset consists of colorized point clouds, or point clouds where each x-y-z point also has an RGB value. Table 3.2 shows all of the object labels in the S3DIS dataset. The labels for objects in this dataset are more general than those in the ScanNet and SunRGBD datasets. In addition to labels for objects in a scene, this dataset also has labels for the type of room (e.g., office, hallway, auditorium, etc.) based on the objects inside of it.

Integer	Label
0	ceiling
1	floor
2	wall
3	beam
4	column
5	door
6	window
7	table
8	chair
9	sofa
10	bookcase
11	board
12	other

Table 3.2: S3DIS object labels

3.1.3 Pre-trained models

MMDetection3D has five architectures for indoor LiDAR-based 3D object detection, each of which has multiple pre-trained models available. While many of the data sets that the models are trained on contain colorized point clouds, these five architectures do not require the LiDAR point cloud to be colorized to make predictions. All of the pre-trained models available for all five of these architectures are investigated to detect discrete objects in the LiDAR point clouds in this work.

3.1.3.1 Deep Hough voting for 3D object detection in point clouds

Votenet [38] is an architecture which uses deep Hough voting for 3D object detection in point clouds. It was the earliest architecture avaialable for object detection in MMDetection3D. This architecture attempts to directly detect objects in point clouds rather than leveraging the existing 2D detector architectures, like many other architectures were attempting to do at the time of its publication. The architecture uses purely geometric information, without any color images, to identify objects. This architecture has pre-trained models available which were trained with both the ScanNet and SunRGBD datasets.

3.1.3.2 3D object detection using hybrid geometric primitives

3D Object Detection Using Hybrid Geometric Primitives (H3DNet) [39] is an architecture that uses a colorless 3D point cloud as input and outputs collections of oriented bounding boxes and their semantic labels. This architecture predicts a hybrid set of geometric primitives (bounding box centers, face centers, bounding box edge centers) and converts them into object proposals with a distance function between the object and those primitives. The distance function allows optimization of the object proposals.This model has a pre-trained model trained with the ScanNet dataset.

3.1.3.3 Group-free 3D object detection via transformers

The previous architectures described use a PointNet network backbone. Group-Free 3D [40] is an architecture that does not use a PointNet network, claiming that the grouping required to use PointNet hurts performance. Instead of grouping local points to each object candidate, this method computes features of an object from all the points in the point cloud with an attention mechanism in the transformers. This architecture has pre-trained models trained with the ScanNet dataset through four different methods.

3.1.3.4 Fully convolutional anchor-free 3D object detection

Fully convolutional anchor-free 3D object detection (FCAF3D) [41] is an architecture which uses voxel representation of a point cloud and processes voxels with sparse convolutions. Unlike the previously described models, it can handle large scale scenes with minimal run time. This architecture does not make prior assumptions of geometry of objects like other object detection methods. This architecture has pre-trained models trained with all three datasets described in Section 3.1.2.

3.1.3.5 Towards real time indoor 3D object detection

Towards real time indoor 3D object detection (TR3D) [42] is a fully convolutional 3D object detection model. It uses both point cloud and RGB inputs, and fuses 2D and 3D features in an attempt of multi-modal detection. This architecture has pre-trained models trained with all three datasets described in Section 3.1.2.

3.1.4 Object detection with pre-trained models

Each pre-trained model has a configuration and checkpoint file which can be downloaded from the MMDetection3D GitHub repository. MMDetection3D has a function called 'inference_detector' which performs 3D object detection on a point cloud with a trained model. Prior to using this function, the pre-trained model must be loaded by calling the 'init_model' function with the configuration and checkpoint files of the pre-trained model as parameters. The loaded model and a point cloud can then be provided to the 'inference_detector' function, which will return a 'pred_instances_3d' class that contains all of the 3D bounding boxes for all of the objects detected in the scene. Appendix A.1 shows a sample of how to perform 3D object detection with a pre-trained model and point cloud in MMDetection3D.

3.1.5 3D object detection performance

The performance of each of the pre-trained models is evaluated with the sample laboratory scene shown in Figure 2.6. The scene is first cropped to a relatively small portion of the room, containing two desks. Figure 3.2 shows the cropped region of the cloud that was used for this analysis. For this scene, the object of interest is the box placed on one of the desks, and the performance of the models is evaluated based on how well the bounding boxes they find describe the bounds of the box. Figure 3.3 shows an image of the box in the point cloud taken by the RealSense depth camera. Figure 3.4 shows the result of object

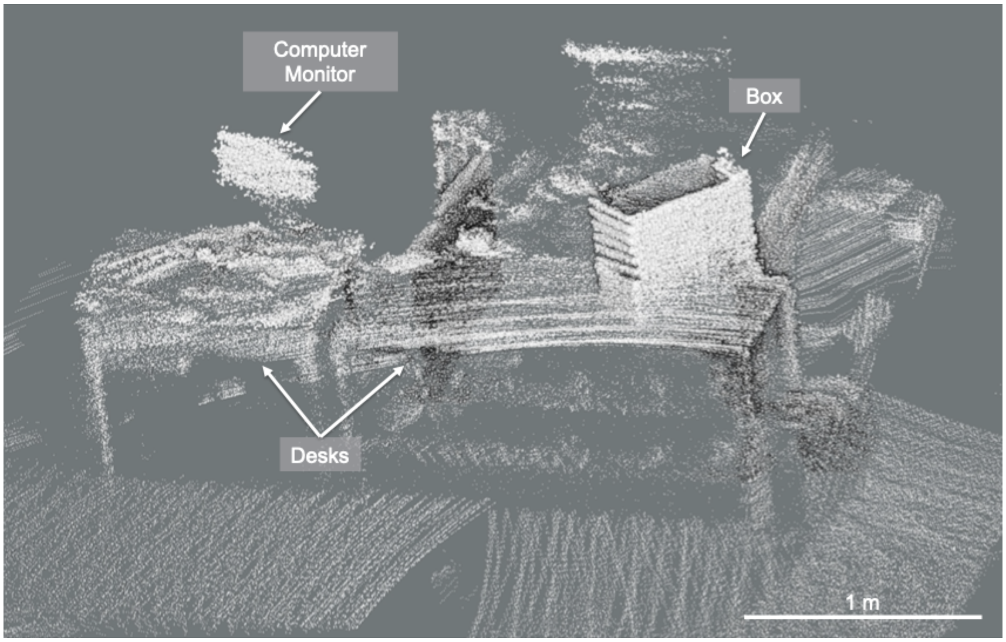


Figure 3.2: Cropped laboratory point cloud used to analyze 3D object detection tools

Figure 3.3: Camera image of laboratory point cloud scene

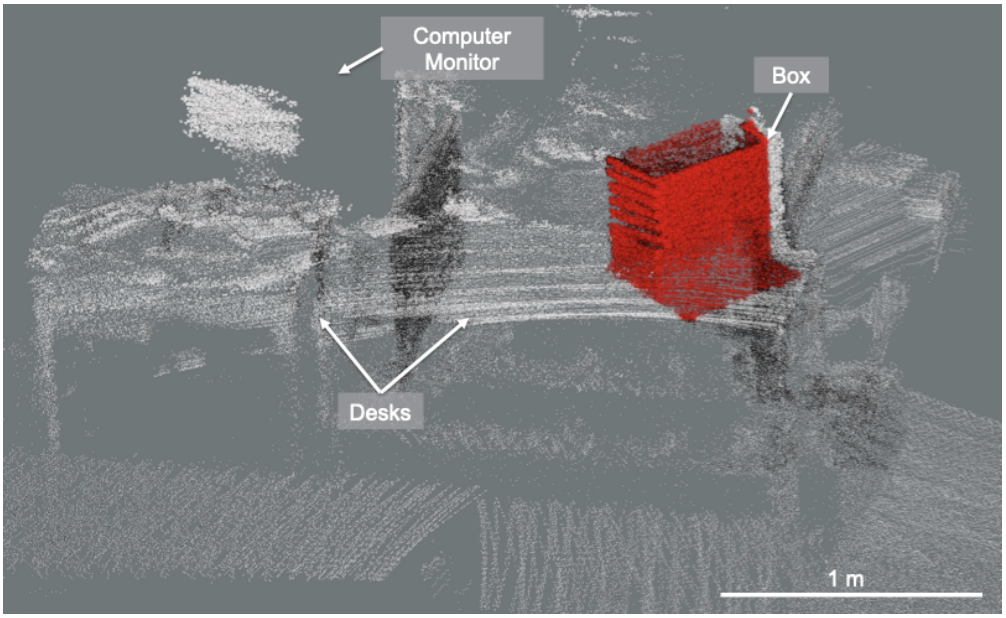


Figure 3.4: Object Detection Results

detection performed by the TR3D model pre-trained with SunRGBD data. The inference detector returned a series of bounding boxes which describe all of the discrete objects in the scene. The bounding box which contains the box on the desk is selected and the point cloud points which lie within this bounding box have been colored red. This model performed well in identifying the object of interest. A few of the point cloud points that comprise the object were not within the range of the bounding box, but it is a relatively small portion of the object. Appendix A.2 shows results for all of the other pre-trained models with this scene. The TR3D model pre-trained on the SunRGBD dataset performs the best in correctly identifying the bounds of the box and is used for performing object detection in this work.

3.2 Semantic segmentation

Semantic segmentation is a computer vision task which attempts to categorize each pixel in an image, or point in a point cloud, into a class or object. In this work, labels can be assigned to objects in our 3D scene to help identify the object we are interested in. These labels can also be used to identify the floor of a room, which has the engineering benefit of specifying the height at which the imager must be placed in reference to the floor. Additionally, these labels could also be used to provide initial guesses for attenuation coefficients for intervening material.

3.2.1 2D semantic segmentation

2D semantic segmentation refers to identifying semantic labels for each pixel in a 2D camera image. The RealSense depth camera on the H420-LAMP produces RGB-D images which can be labeled with 2D semantic segmentation techniques. Because these images also contain depth information, the labels could be projected onto a 3D scene.

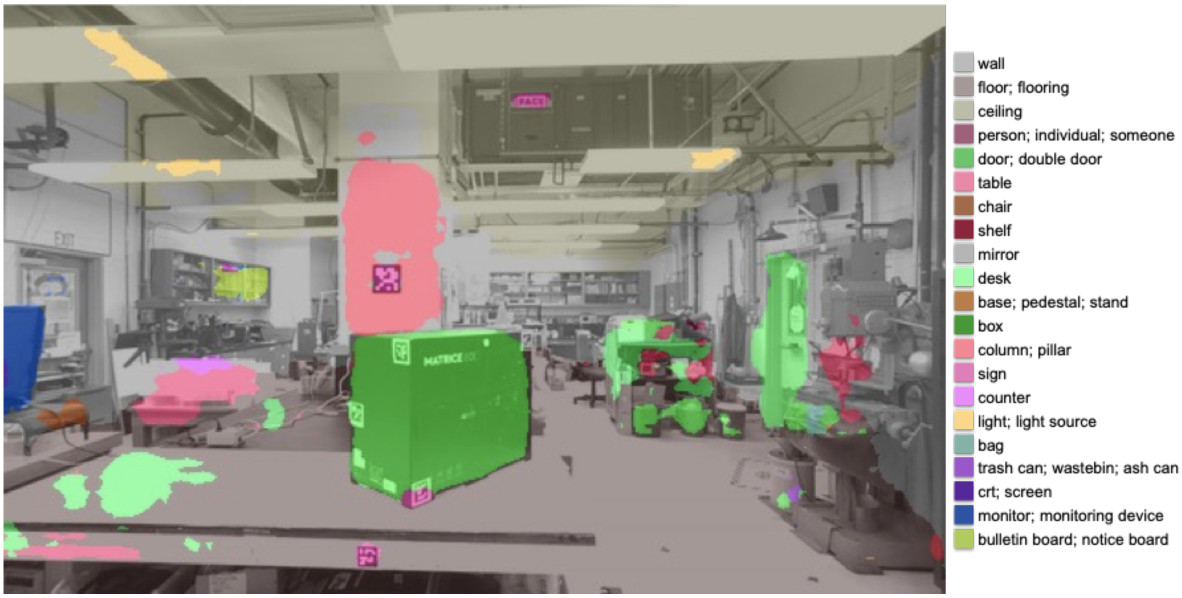


Figure 3.5: 2D semantic segmentation results

DeepLab-V3 is a state-of-the-art 2D semantic image segmentation model developed by Google [43]. Similar to 3D computer vision tasks, there are a variety of datasets with labeled camera images which are used to develop and validate 2D semantic segmentation techniques. ADE20K is one such dataset which contains over 27,000 labeled camera images of household and street scenes [44], [45].

Figure 3.5 shows the results of using a Google DeepLab-V3 model pre-trained on ADE20K to semantically segment the image in Figure 3.3. Many pixels in the scene are correctly labeled, including the box, the computer monitor and its stand, the column, as well as large regions of the ceiling, floor, and walls. Additionally, the ArUco markers in the image are correctly labeled as ‘sign’ and the door on the right side of the image is labeled as ‘door’. Some of the pixels which make up the desk in the scene are correctly labeled as ‘desk’ or ‘table’, but large portions of pixels which make up the desk are incorrectly identified as ‘floor’. Some of the pixels in three of the lights are labeled as ‘light’, but the rest of the light pixels are labeled as ceiling. Many pixels in the scene where objects are present are given the label ‘wall’, ‘floor’, or ‘ceiling’. This is likely because the labels in ADE20K do not adequately describe a laboratory scene. These results demonstrate that the pre-trained model can find gross features of the room and label some objects, but a more relevant dataset would be required to correctly label objects in a laboratory or industrial scene.

3.2.2 3D semantic segmentation

It is also possible to semantically segment a 3D scene directly from 3D data, such as a LiDAR or photogrammetry point cloud. MMDetection3D supports 3D semantic segmentation in addition to 3D object detection.

3.2.3 Pre-trained models

The MMDetection3D package has indoor and outdoor semantic segmentation architectures available for both LiDAR and camera based segmentation. There are two architectures available for indoor 3D semantic segmentation. Each of these architectures also has a number of pre-trained models available which use the ScanNet and S3DIS datasets.

3.2.3.1 PointNet++: Deep hierarchical feature learning on point sets in a metric space

PointNet++ [46] is a hierarchical neural network that applies PointNet recursively on a nested partitioning of the input point set. It learns local features with increasing contextual scales. This architecture uses LiDAR point clouds with and without color as an input. This architecture has models pre-trained with the ScanNet and S3DIS datasets.

3.2.3.2 Position adaptive convolution with dynamic kernel assembling on point clouds

Position adaptive convolution with dynamic kernel assembling on point clouds (PAConv) [47] is a generic convolution operation for 3D point cloud processing. It constructs the convolution kernel by dynamically assembling basic weight matrices stored in a weight bank, where coefficients of the weight matrices are self-adaptively learned from point positions through ScoreNet. This architecture has only two pre-trained models, each trained on the S3DIS dataset.

3.2.4 Semantic segmentation with pre-trained models

The approach to using MMDetection3D for 3D semantic segmentation is similar to the approach of using it for 3D object detection. In this case, a pre-trained model and point cloud is provided to a function called 'inference_segmentor', which will return an array called 'pts_semantic_mask' which contains an integer corresponding to the label assigned to each of the point cloud points. Tables 3.1 and 3.2 show the corresponding labels for the integers returned for ScanNet and S3DIS respectively. Appendix B.1 shows a sample of how to perform 3D object detection with a pre-trained model and point cloud.

3.2.5 Semantic segmentation performance

Each of the pre-trained models is used to infer semantic labels to the same point cloud used to evaluate the point cloud shown in Figure 3.2. Figure 3.6 shows the results of using the PointNet++ model pre-trained on the ScanNet dataset to semantically label the point cloud. The semantic labels are visually represented by coloring each point cloud point based on its label. The colors used for each of the labels are shown in the key at the bottom of the image. The points which comprise the desk in the scene are correctly given mixed labels of 'table' and 'desk'. However, most of the other points are incorrectly classified. The points within the box of interest are given the incorrect labels of 'window' and 'bookshelf'. The points which comprise the computer monitor are also given the label 'window'. There are a number of points which have false labels of 'bed' and 'chair'. The poor classification is due to the ScanNet dataset only containing labels for household objects, which are not relevant to a laboratory setting.

Appendix B.2 shows results for the other pre-trained models with the same scene. None of the models perform particularly well and have the same issues as the result shown in Figure 3.6. Some of these models use the S3DIS dataset, which has more general labels but is accustomed to larger, building-sized point clouds. Therefore, models trained with it do not perform well with the constrained clouds that are used in this work. While the labels are more general in the S3DIS dataset, they still do not adequately describe laboratory or industrial scenes.

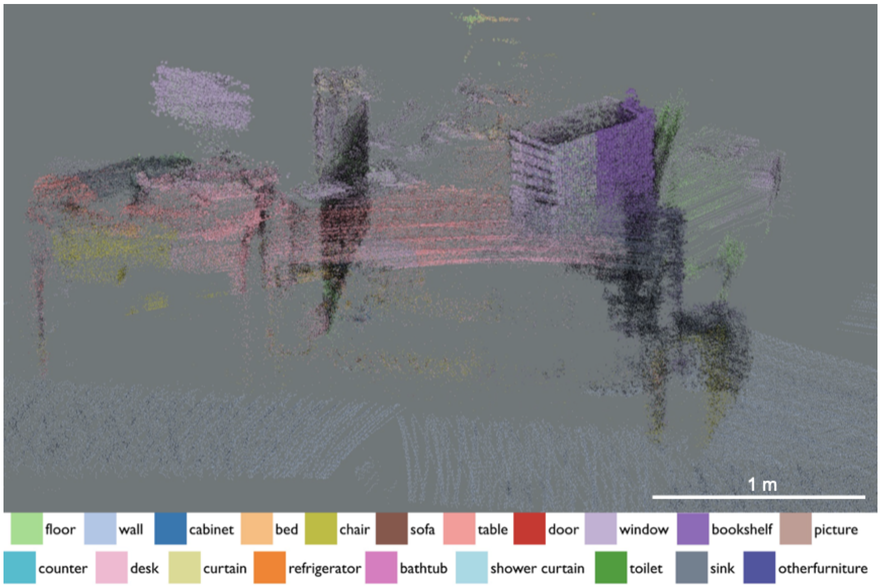


Figure 3.6: Semantic Segmentation Results

3.3 Attenuation estimation

To correctly reconstruct radiation emission rates, attenuation through intervening material must be accounted for. Previously, SDF has accounted for air attenuation in some reconstruction techniques by computing the distance between the detector pose and voxel and treating the entire path length as air, even if occupied material exists between the detector and voxel [18], [19]. In order to properly model attenuation, the path length traversed through various materials and their attenuation coefficients must be known. The 3D semantic segmentation techniques described in Section 3.2 can be used to inform attenuation estimates, or they can be provided by a user when the material is known.

3.3.1 Fast voxel traversal ray-tracing algorithm

Ray tracing is a technique for modeling light transport within a scene, enabling the calculation of path length traversed through various materials. The fast voxel traversal algorithm formulated by Amanatides and Woo [48] (with pseudo-code provided in [17]) is implemented in Python for ray tracing in this work.

The algorithm is initialized with a start and end point which both reside within the bounds of a voxel grid where each voxel has a label of occupied or unoccupied. Additionally, occupied voxels may have a label corresponding to the type of material which resides within it. In this work, the end point is typically the center of an occupied voxel which is treated as a gamma-ray source. The start point is any position where a radiation imager may be placed

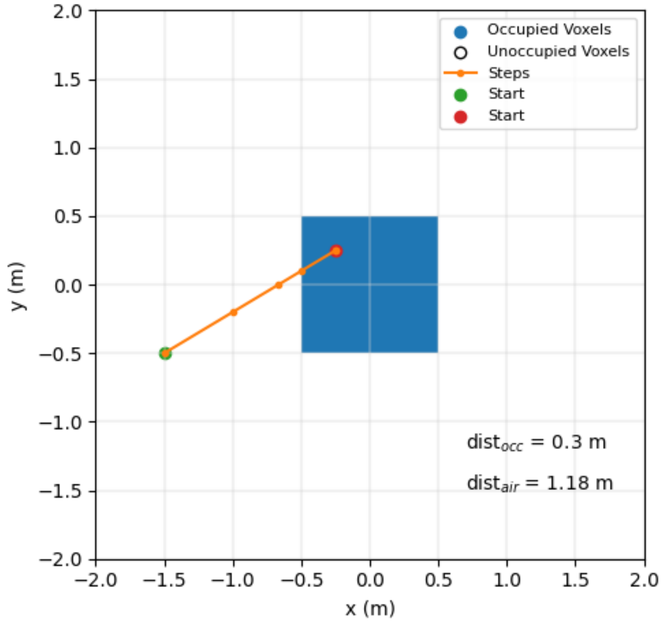


Figure 3.7: ray-tracing algorithm steps

that is within an unoccupied voxel. The voxel indices i, j, and k (for x, y, and z) are identified for both the start and end points. An incrementor array is then created based on the sign of each element of the direction vector between the start and end points, indicating whether each element of the start voxel indices must be increased or decreased to move towards the end voxel.

Ray tracing is performed in a loop through a series of ‘steps’. Here, a step refers to an x-y-z point lying on the line between the start and end points along which the ray is incrementally traced. The start point serves as the first step. The loop uses the incrementor array to find the next voxel in x, y, and z and identifies the voxel boundary closest to the current step in each dimension along the line between the start and end points. The ray is moved to the closest boundary, which will serve as the next step. During each iteration the algorithm stores the distance traveled and occupancy of the traversed voxel, including the occupied voxel label if provided. This process continues until the boundary of the final voxel is reached. Finally, the ray is traversed through the remaining portion of the terminating voxel to reach the end point.

Figure 3.7 shows a 2D representation of the ray-tracing algorithm. The start point is in green and the end point is in red. The orange line represents the entire length traversed in the algorithm, and each step of the loop is plotted with an orange marker.

3.3.2 Attenuation calculation

To compute the attenuation through materials, the path length through a material and its attenuation coefficient are needed. The ray-tracing algorithm described in Section 3.3.1 records the distance through occupied and unoccupied voxels as it traverses from 'start' to 'end', as well as occupied voxel labels if provided. Upon termination of the loop, the distances traversed through occupied and unoccupied voxels are summed. If labels were provided for occupied voxels, these distances are summed separately. In addition to showing the individual steps of the ray-tracing algorithm, Figure 3.7 shows the distances traversed through occupied and unoccupied voxels. The sum of distances computed are 30 cm for occupied voxels and 118 cm for unoccupied voxels, which is treated as air at standard temperature and pressure (STP) for attenuation estimates.

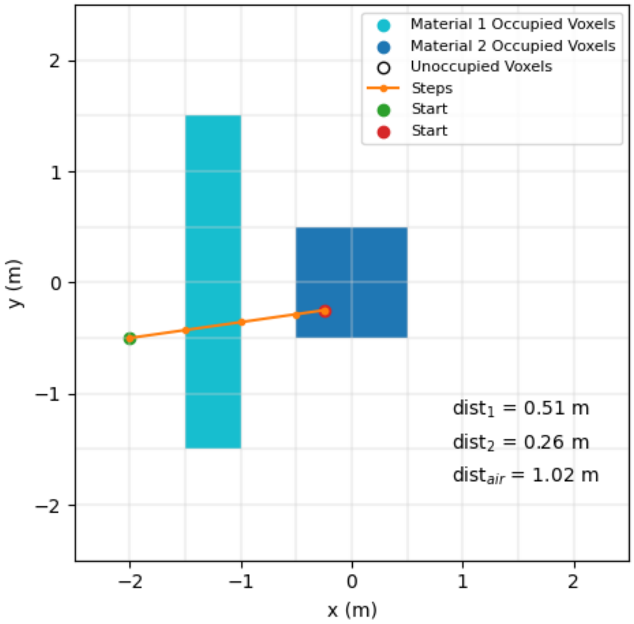


Figure 3.8: ray-tracing algorithm steps demonstrated for a scenario where two different material types are present

Scenes are typically composed of more than one type of material, so it is desirable to compute distances traversed through multiple individual materials. If a voxel grid is created where the occupied voxels are assigned labels based on what type of material they are (as determined through instance segmentation or object detection), the distances through each material can be computed separately. Figure 3.8 shows an example of the ray-tracing algorithm traversing through occupied voxels of two different types of materials. The distances traversed through occupied voxels are 51 cm and 26 cm for material 1 and 2 respectively.

Currently the ray-tracing algorithm is only implemented for cases where only two types of material in addition to air are present, but this could be expanded to handle a larger number of materials.

3D semantic segmentation techniques could be used to assign labels to different occupied voxels in the scene. In addition to assigning labels to voxels for calculating distances traversed through individual materials, these labels could be used to identify an attenuation coefficient. The labels could be used to inform an initial guess for attenuation coefficient which may be solved iteratively [17]. Attenuation coefficients could also be determined with a lookup table which contains materials associated with semantic labels for a range of gamma-ray energies. Given a path length and attenuation coefficient of a material, attenuation by that object can be calculated for any start point and end point.

3.3.3 Visibility cutoff

In addition to calculating path lengths through intervening material, the ray-tracing algorithm is also used to determine whether a voxel is visible from an x-y-z point. A voxel is considered visible if the only occupied material traversed from a start point to the voxel center is from the voxel itself. Figure 3.9 left shows a scenario where a voxel would be considered visible, because the only occupied material traversed is from the terminating voxel. The figure on the right shows an example from the same start point where the voxel is considered not visible because some of the occupied material traversed is from a different voxel.

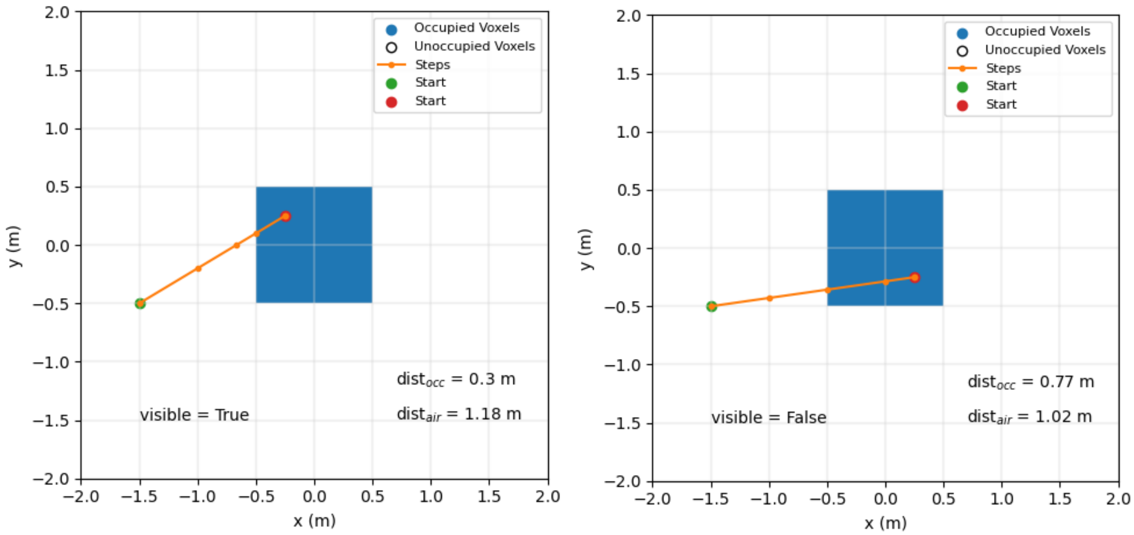


Figure 3.9: Example of voxels which are visible (left) and invisible (right) from a start position

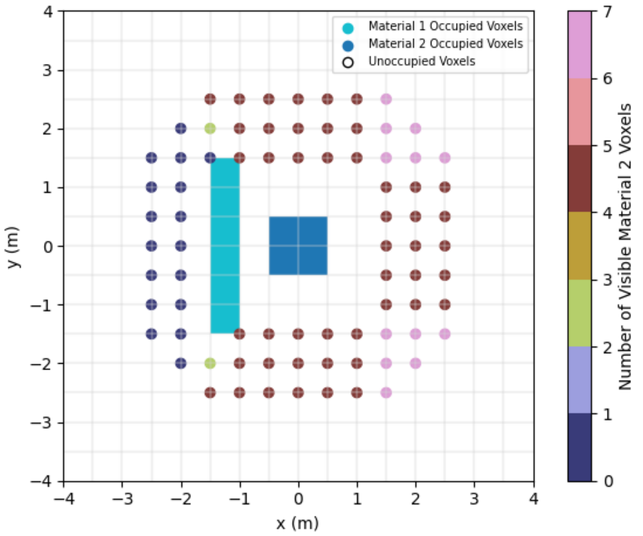


Figure 3.10: Number of voxels occupied with material 2 visible at several different start points

Figure 3.10 shows an example of how many voxels occupied with material 2 are visible at several different start points for scenario shown in Figure 3.8. This figure validates that the visibility calculation is performing as expected. Start points where more voxels are visible generally have less attenuation from intervening material than start points where less voxels are visible, so these positions may be considered desirable locations for gamma-ray imaging.

3.4 Discussion

One of the weaknesses of the 3D object detection and semantic segmentation methods used in this work is that all of the datasets used are generated with household and office scenes, whereas the setting for this work is in laboratory or industrial scenes. Object detection is less sensitive to the poor choice of data, as it is performing the relatively simple task of seeking point cloud points which comprise the same object. The H3D model is used for object detection in this work because it successfully identifies the bounds of very simple objects. However, in safeguards applications, there may be small details in a scene which must be identified as separate objects, such as pipes containing gaseous materials. None of the object detection methods investigated in this work are able to effectively detect small (on the order of two inches) objects because the data in the dataset was collected with contextual sensors with too low of a resolution to properly reconstruct fine levels of detail.

The household labels have more of an effect on the semantic segmentation tasks, because the only labels available are for household objects. Currently there is not a publicly available labeled dataset of 2D or 3D laboratory or industrial scenes which could be used to train the semantic segmentation architectures investigated in this work. Due to the lack of available data, this work does not use semantic segmentation techniques to inform attenuation estimates. Rather, the user must provide attenuation coefficients for known materials in a scene. With increasing use of computer vision in industrial and laboratory settings, a more relevant dataset may become available. The architectures explored in this work could be trained with such a dataset to produce a trained model which predicts labels for objects more accurately in laboratory and industrial scenes. These labels could then be used with a lookup table for attenuation coefficients to inform attenuation calculations in gamma-ray imaging reconstructions.

One weakness of the ray-tracing algorithm is that it assumes the entire voxel to be filled with material when it is occupied, when the voxel could only contain a few point cloud points. This means that the distances traversed through occupied voxels could be calculated to be higher than it is in reality. To prevent this overestimation, voxel sizes could be made smaller. However, the trade-off is that smaller voxel sizes mean that the ray-tracing algorithm will take longer to complete.

Chapter 4

Measurement position optimization[1]

Gamma-ray imaging is a useful tool to safeguards analysts because it allows for non-destructive characterization and quantification of nuclear materials. While gamma-ray imaging is a useful tool, measurements can be time consuming particularly if low levels of uncertainty are desired. Safeguards inspectors typically have a limited amount of time to perform measurements, ideally with minimal disruption to facility operation. Therefore, it is desirable to have a gamma-ray imaging tool which can identify optimal measurement positions to perform counting measurements. The contextual information collected in Scene Data Fusion can be leveraged to define a measurement space and identify optimal measurement positions within it.

This chapter will outline an optimization technique which uses contextual data and a gamma-ray imagers response to identify optimal measurement positions. The optimization consists of four steps:

1. Identify volume of interest
2. Select viable measurement positions
3. Calculate sensitivity to volume of interest at potential measurement positions
4. Find subset of positions which maximize sensitivity while maintaining uniform sensitivity to the volume of interest

4.1 Example model

First, a simple example model will be defined to demonstrate the computational methods described in this chapter. The example model consists of a 10 m x 10 m x 10 m voxel grid with a 0.5 m voxel pitch. The object of interest consists of 8 occupied voxels in the center

---

[1] Much of the text in this chapter is published in the 2023 INMM and ESARDA Joint Annual Meeting Proceedings [49].

of the voxel grid. These 8 occupied voxels are assumed to be uniformly distributed with a source which emits 662 keV gamma rays. All other voxels in the scene are assumed to be unoccupied. Figure 4.1 shows the extent of the voxel grid with the volume of interest in the center.

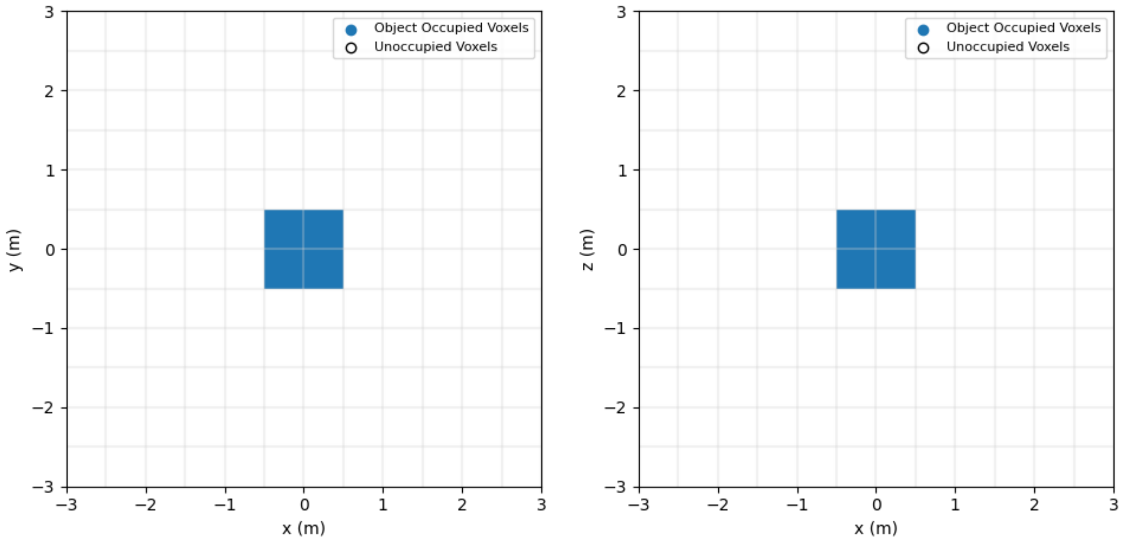


Figure 4.1: x-y (left) and x-z (right) planes of example model

4.2 Identification of volume of interest

The first step in the optimization is to select a volume of interest. This work assumes that the object being imaged occupies some volume within the voxel grid, which can consist of one or more occupied voxels. Therefore, the voxels which contain the imaged object must be identified. Selection in this context refers to identifying the x-y-z bounds in the point cloud which contains the volume of interest.

In safeguards applications, the safeguards inspector is tasked with verifying declared amounts of material present. Therefore, the object of interest is already known, and only needs to be identified within the 3D map of the scene by the operator.

4.2.1 Manual object detection

Object detection can be performed manually by inspecting the point cloud and identifying the x-y-z limits of the volume of interest. This region of the global voxel grid is cropped and identified as the object voxel grid.

4.2.2 Automatic object detection

Manual object identification would require too much time for this tool to be useful in the field, so an automatic object detection technique was developed to assist operators in identifying measurement objects. Section 3.1 describes how this task is performed with the contextual data collected by the detector system. Once all of the objects in the scene are identified, an operator will select the object they are interested in finding optimal positions for. The bounding box around the object will be used to construct a voxel grid where the number of voxels in the object or voxel pitch will be specified. The voxels within the object bounding box will be labeled as ‘object’, while all other occupied voxels in the scene will be labeled as ‘other’ or ‘non-object’.

4.2.3 Custom voxel grid

Once the object is selected within the point cloud, a voxel grid is constructed which contains the object at some desired voxelization (i.e desired number of voxels for the object or desired voxel pitch). The voxel grid is then extended around the object in all directions up to a user-provided distance. Typically this value will be a few meters or less due to the inversely proportional relationship between number of counts in the detector and distance from the source. The extended voxel grid has the same voxel size as the object of interest. Each voxel is considered occupied if there are a number of points within its bounds greater than a cutoff value. All other voxels are considered to be unoccupied. The object voxels are given an ‘object’ label and all other occupied voxels are labeled as ‘non-object’.

Care must be taken with the voxelization parameters to ensure the object and its surrounding space are accurately represented by the voxel grid. If there are small details in the scene which should be represented in the voxel grid, the voxel pitch must be sufficiently small to capture the level of detail desired. However, if the voxel size is decreased, the cutoff value for occupied voxels should also be decreased to account for the smaller voxel volume.

4.2.4 Example model object identification

In the case of the example model, the object is manually defined to be a 1 m x 1 m x 1 m box, which has been voxelized into 8 voxels with a 0.5 m pitch. These 8 voxels are all occupied and given the label ‘object’. These are the only occupied voxels in the scene, so no voxels are given the ‘non-object’ label. Figure 4.2 shows a similar example where an external object has been added. In this figure, the external object voxels and the central object voxels are each given unique labels.

4.3 Viable measurement position selection

Once an object of interest is selected, a grid of potential measurement locations is automatically constructed around the object voxel grid between a user-provided maximum

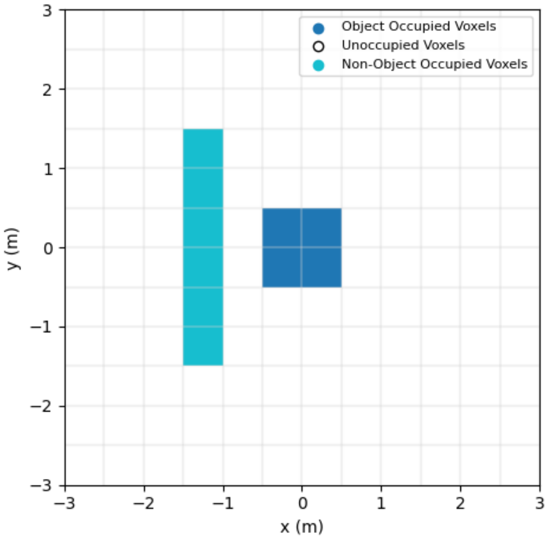


Figure 4.2: Example of measurement space with external shield

standoff distance and a minimum standoff distance such that all object voxels are in the coded-aperture imaging FOV of 86° × 86°. This constraint is included because the imager is capable of coded-aperture imaging, but the coded-aperture imaging modality is not used in this work. This constraint also ensures the imager remains in the far-field to ensure the imaging formulations described in Section 2.3 can be used. Maximum standoff distance is user provided, but it can also be inferred by the desired number of counts and the estimated count rate of the object from the initial scene survey.

Some measurement positions are nonphysical or undesirable for a counting measurement. To remove these points, the measurement positions which reside in occupied voxels are removed. The ray-tracing algorithm described in Section 3.3.1 is used to determine if at least one object voxel is visible at each potential measurement position, as some object voxels may be occluded by the object itself. Typically an object in a scene is divided into multiple voxels. For a candidate measurement position to be considered, it must be able to see at least one of those voxels. It is acceptable if they are not all visible - for example, a 3D object will have voxels on the other side of the object from the desired position. A 3D object might also have internal voxels in addition to surface voxels, which will not be visible at any measurement position. Positions where no object voxels are visible are removed from consideration.

Once the potential measurement x-y-z positions are identified, a discretization in orientation around the z-axis (gravity axis) of the imaging system may be defined. The coordinate system of the detector is shown in Figure 2.2. Orientations around the x- and y-axis are not included due to physical constraints of placing the imager on the tripod. Each potential orientation is considered for each of the x-y-z positions, but generally those which point toward the object of interest are desirable. If the orientation is not set, a uniform detector response is assumed (i.e. angular response is not considered).

Each combination of x-y-z positions and orientations is referred to as a potential 'pose' for the imager. Each of the potential poses are evaluated to serve as the the optimal imager placement for static imaging to quantify the volume of interest.

4.3.1 Example model position selection

The potential measurement positions are generated for the example model with an added constraint that all of the positions are on-axis with the object of interest in the z-dimension with a maximum standoff distance of 3 m from the center of the object, resulting in 84 x-y-z positions. The orientation is discretized in 45°increments around the z-axis, resulting in 8 potential orientations, making a total of 672 poses. Because the only occupied voxels in the example model are the source voxels, the ray tracing techniques do not mask out any of the measurement positions as infeasible. The 672 potential measurement poses generated are shown in the x-y and x-z plane in Figure 4.3.

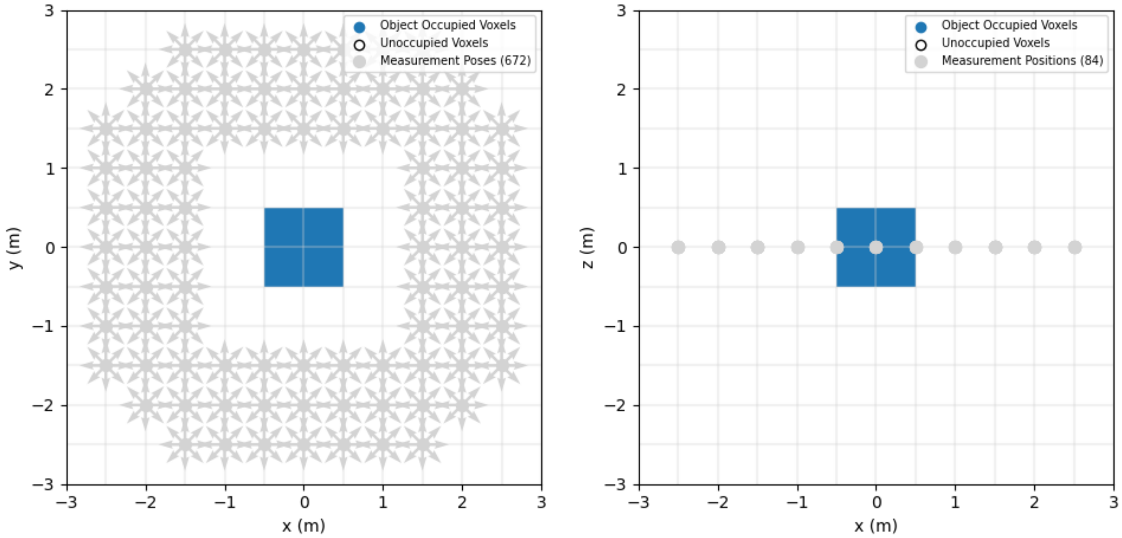


Figure 4.3: Potential measurement poses selected for example model shown in x-y plane (left) and x-z plane (right).

4.4 Sensitivity calculation

When trying to quantify the amount of nuclear material present in a container within desired levels of uncertainty, the primary objective is to maximize the number of radiation counts.

Sensitivity is a property we can optimize in order to achieve this, therefore prior to optimizing we must calculate the sensitivity from each potential measurement pose to each object voxel. The sensitivity to each object voxel at each potential measurement pose is computed as:

$$S = R \times \frac{1}{r^2} \times e^{-\mu_{unocc} r_{unocc}} \times e^{-\mu_{obj} r_{obj}} \times e^{-\mu_{occ} r_{occ}} \tag{4.1}$$

where:

- R is the expected detector response for double gamma-ray events at the desired energy at the solid angle between the given detector pose and an object voxel,
- r is the distance between the center of the object voxel and the detector position,
- μ_{unocc} is the linear attenuation coefficient of the provided gamma-ray energy in air at STP,
- r_{unocc} is the path length of air between the object voxel center and the measurement position,
- μ_{obj} is the linear attenuation coefficient of the provided gamma-ray energy in object occupied voxels, and
- r_{obj} is the path length of occupied object voxels between the object voxel center and the measurement position.
- μ_{occ} is the linear attenuation coefficient of the provided gamma-ray energy in non-object occupied voxels, and
- r_{occ} is the path length of non-object occupied voxels between the object voxel center and the measurement position. Note that $r_{obj} + r_{occ} + r_{air} = r$.

The detector response is generated through experimentally bench-marked Geant4 simulations [8]. The polar and azimuthal angles between the measurement pose and source voxel are computed and the response is interpolated from response data for the desired gamma-ray energy. The ray-tracing algorithm described in Section 3.3.1 is used to find the occupied and unoccupied path lengths. The occupied attenuation coefficient is manually provided for the object and non-object materials.

4.4.1 Example model sensitivity

For the example model, double interaction events which deposit a total of 662 keV in the detector are assumed for the detector response. The linear attenuation coefficient for water at 662 keV of 8.574 m^{-1} is used for the object of interest. There are no non-object materials in the scene, so the non-object attenuation term ($e^{-\mu_{occ} r_{occ}}$) of Equation 4.1 is not used. Figure 4.4 shows the mean sensitivity at each x-y-z position to each object voxel. Due to

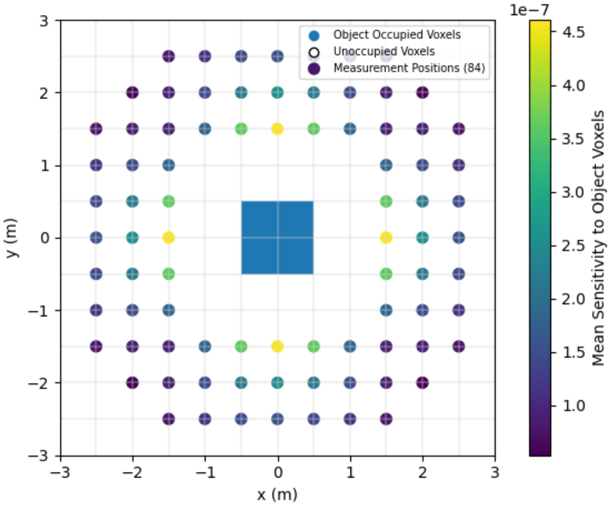


Figure 4.4: Mean sensitivity at each potential position to all object voxels of the example model

the simple, symmetrical nature of this model, the $\frac{1}{r^2}$ term is dominant. Figure 4.5 shows the mean sensitivity at each x-y-z position for each of its potential orientations. This figure demonstrates the directional dependence of the detector response.

4.5 Multi-objective optimization

In this work we are interested in optimizing sensitivity. A simple approach would be to find the position which maximizes the mean sensitivity to object voxels. However, this approach could lead to non-uniform exposure to all of the voxels that comprise the object of interest. To address this shortcoming, we introduce another optimization variable, the minimization of the mean of the variance in sensitivity to each of the object voxels at the measurement position. This means that we are performing a multi-objective optimization. Multi-objective optimization problems may not have a single best solution. Instead they have multiple solutions with trade-offs between the variables of interest which may conflict with one another.

4.5.1 Best single position

In multi-objective optimization, the optimization of one metric may be at the expense of another. To find the candidate optimal solutions we use the Pareto front, a concept borrowed from economic and game theory which seeks to find solutions that ‘help’ one metric without

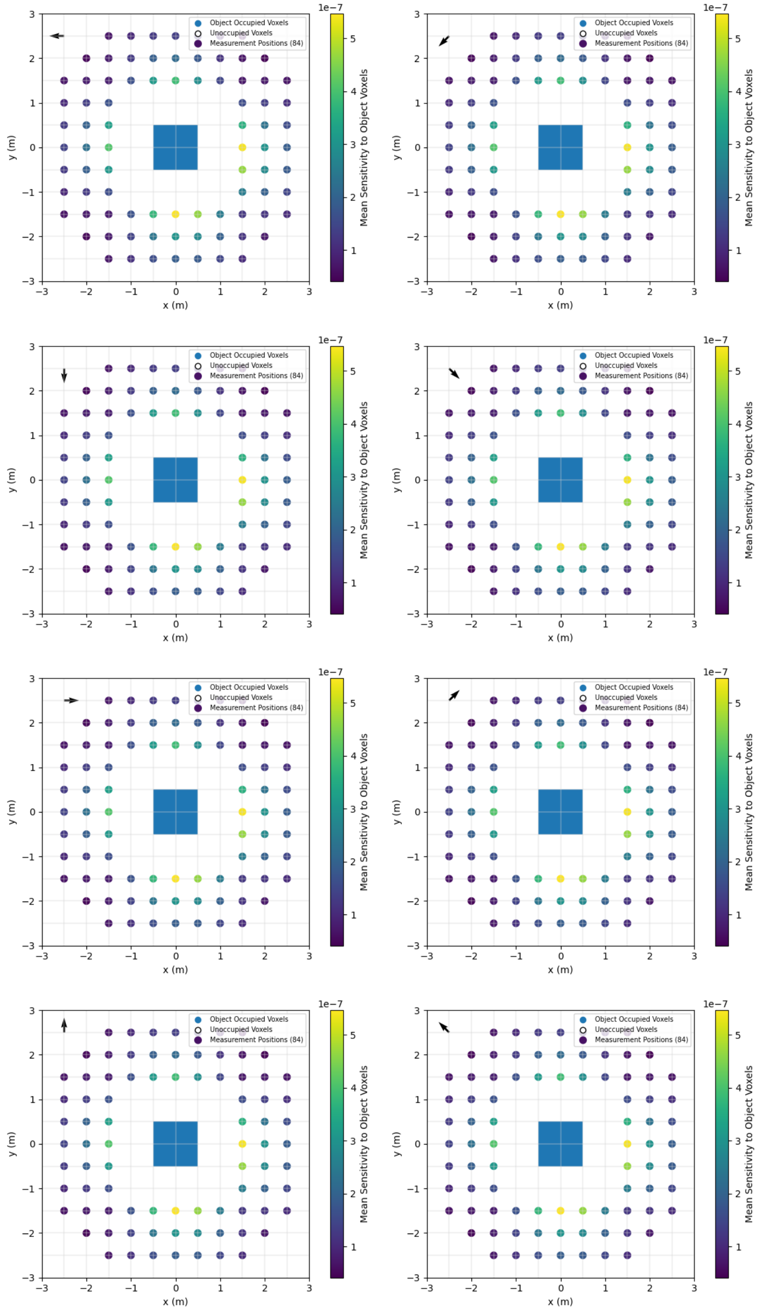


Figure 4.5: Mean sensitivity for each potential pose of the example model. The arrow on each plot represents the orientation of the poses.

‘hurting’ the other [50]. The Pareto front comprises all of the optimal solutions to a multi objective optimization problem.

Before finding the Pareto front, the optimization variables are computed. The mean and variance in sensitivity to each object voxel for each of the potential measurement poses is computed to serve as the two optimization variables for each of the potential poses. Figure 4.6 shows the sensitivity mean and variance for each of the potential poses in the example model in gray. Each list of values is min-max normalized, and the normalized values are used to construct the Pareto front.

The Pareto front is found by sorting the list of optimization variables in increasing order of sensitivity variance, followed by decreasing order of sensitivity mean. The first value in the sorted list is selected as the first point on the Pareto front. The list is then looped through in order, adding values which have mean values greater than the previous. At the conclusion of the loop, the measurement positions on this list are considered to be optimal. Appendix C.1 contains the Python code that was developed to compute the Pareto front in this work. Figure 4.6 shows the Pareto front in cyan. Any of the metrics which lie on the cyan line are considered optimal solutions to the problem.

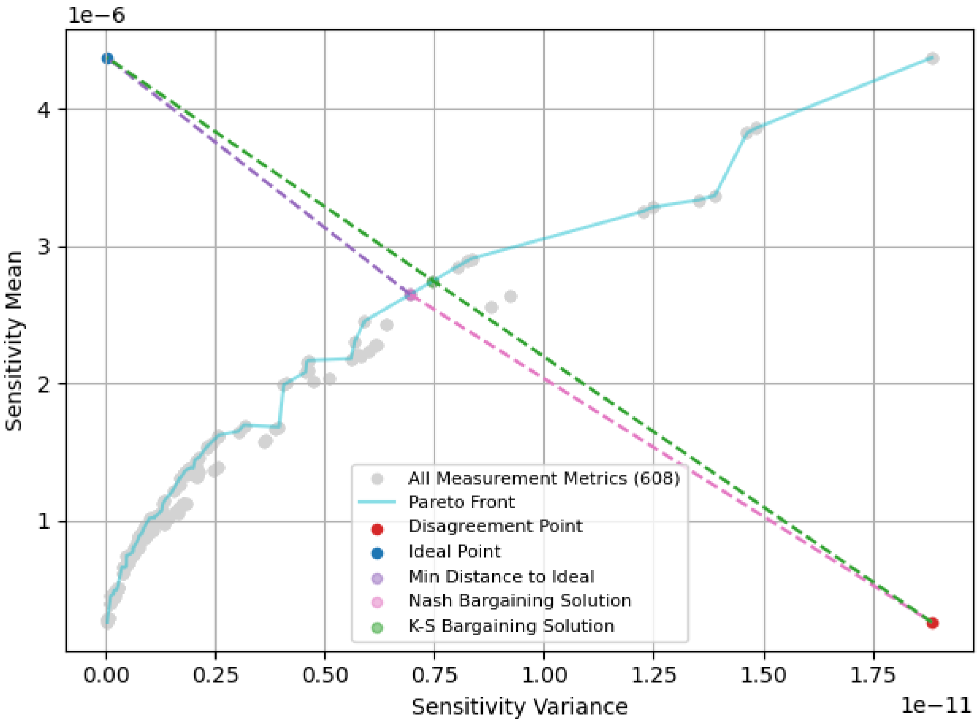


Figure 4.6: Pareto front and how different solution methods are derived

A method for selecting a single position from the set of Pareto optimal solutions is required to create a measurement plan. There are a variety of methods to choose the desired candidate from the Pareto front, including the Nash bargaining solution [51] and the Kalai-Smorodinsky bargaining solution [52]. To use these solutions, we first calculate two hypothetical points, the ideal and disagreement points. The ideal point is the maximum sensitivity mean value and the minimum sensitivity variance value, representing the best value for each of the parameters which does not physically exist. The disagreement point is the minimum sensitivity mean value and maximum sensitivity value, representing the worst values in each parameter. Figure 4.6 shows the ideal point in blue and the disagreement point in red.

One solution method is the Nash bargaining method, which seeks the Pareto front value which has the furthest Euclidean distance from the disagreement point. The minimum distance to ideal method borrows from the Nash bargaining solution, but rather than maximizing distance from the disagreement point, it finds the closest solution to the ideal point. The final solution method examined is the Kalai-Smorodinsky (K-S) method, which finds the Pareto front value which lies closes to the line connecting the ideal and disagreement points. Figure 4.6 shows the solution identified by each of these methods. Note that the Nash bargaining solution and the minimum distance to ideal methods identify the same solution.

4.5.2 Best combination of positions

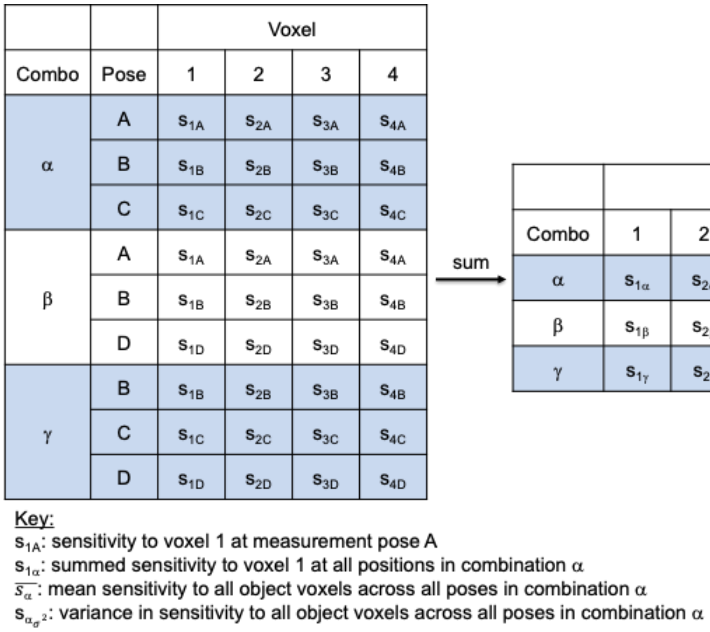
The methods described so far find the best single position to perform a radiation imaging measurement. However, a single position will likely result in non-uniform exposure to the object of interest, meaning that our sensitivity variance values will be poor. It is also desirable to have multiple view points to construct a 3D gamma-ray image. Therefore, it is preferred to consider solutions with multiple measurement positions.

First a permutation is performed to find all possible combinations of measurement positions. Typically a constraint is imposed to allow this to be computationally tractable by setting a desired number of measurement positions. Note for many potential positions or for a high desired number of positions, the permutation is computationally intensive and infeasible for real time operation.

Once all of the combinations of measurement positions are found, the optimization variables are computed. Figure 4.7 illustrates the steps taken to obtain the optimization variables. First the sum of sensitivity to each of the object voxels across all poses in each combination is computed. Then, similar to the single best position methodology, the mean and variance in summed sensitivity to each object voxel is computed to serve as the optimization variables. The rest of the optimization is performed in the same manner as the single position methodology - the Pareto front is assembled and a solution is identified through the Nash, K-S, or minimum distance to ideal method.

4.5.3 Example model optimization

A combination of 3 measurement poses from the 672 candidate poses results in 50,803,424 potential combinations to perform the optimization over. The sensitivity to each of the 8 object voxels is computed for all 672 poses, and the metrics of interest are computed for



		Voxel			
Combo	Pose	1	2	3	4
α	A	s_{1A}	s_{2A}	s_{3A}	s_{4A}
	B	s_{1B}	s_{2B}	s_{3B}	s_{4B}
	C	s_{1C}	s_{2C}	s_{3C}	s_{4C}
β	A	s_{1A}	s_{2A}	s_{3A}	s_{4A}
	B	s_{1B}	s_{2B}	s_{3B}	s_{4B}
	D	s_{1D}	s_{2D}	s_{3D}	s_{4D}
γ	B	s_{1B}	s_{2B}	s_{3B}	s_{4B}
	C	s_{1C}	s_{2C}	s_{3C}	s_{4C}
	D	s_{1D}	s_{2D}	s_{3D}	s_{4D}

	Voxel			
Combo	1	2	3	4
α	$s_{1\alpha}$	$s_{2\alpha}$	$s_{3\alpha}$	$s_{4\alpha}$
β	$s_{1\beta}$	$s_{2\beta}$	$s_{3\beta}$	$s_{4\beta}$
γ	$s_{1\gamma}$	$s_{2\gamma}$	$s_{3\gamma}$	$s_{4\gamma}$

Combo	
α	$\overline{s_\alpha}$
β	$\overline{s_\beta}$
γ	$\overline{s_\gamma}$

Combo	
α	$s_{\alpha_\sigma^2}$
β	$s_{\beta_\sigma^2}$
γ	$s_{\gamma_\sigma^2}$

Key:
s_{1A}: sensitivity to voxel 1 at measurement pose A
$s_{1\alpha}$: summed sensitivity to voxel 1 at all positions in combination α
$\overline{s_\alpha}$: mean sensitivity to all object voxels across all poses in combination α
$s_{\alpha_\sigma^2}$: variance in sensitivity to all object voxels across all poses in combination α

Figure 4.7: Calculation of optimization variables for multiple position optimization

each potential combination for the brute-force optimization. Figure 4.8 shows the sensitivity mean and variance metrics for a down-sampled fraction of all possible combinations of poses in gray. The cyan curve shows the Pareto front for these metrics, representing the set of optimal combinations for the example model. Figure 4.9 left shows where the poses that comprise these optimal combinations lie in the x-y plane. The solution identified by the Nash bargaining solution is shown in purple in Figure 4.8, lying close to the 'ideal point' of maximum sensitivity and minimum variance. The poses that comprise this combination are shown in Figure 4.9 right. Due to the symmetrical nature of this example model, there are degenerate combinations with the same metrics in 90° rotations around the object and in mirrored configurations.

4.6 Application of genetic algorithm to accelerate optimization

In the case of many potential measurement poses, the brute-force approach results in a large computational burden (e.g., for 1000 potential measurement poses and 3 optimal poses, there are over 1.67×10^8 possible combinations of poses which must be computed). In these cases, it is desired to use an optimization method such as a genetic algorithm (GA) to accelerate these calculations to facilitate real-time operation.

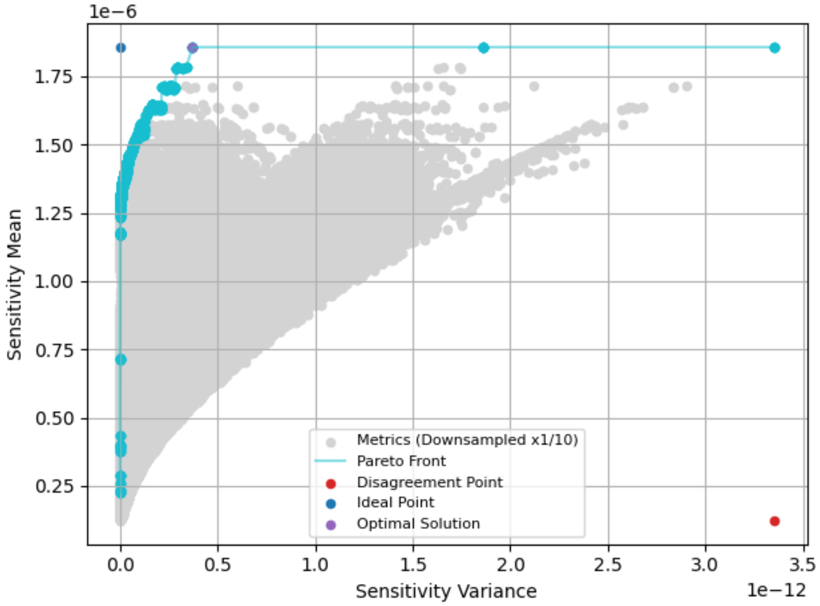


Figure 4.8: Optimization metrics for a 1/10 sample of potential combinations of measurement poses with Pareto front and optimal solution identified through the Nash bargaining solution

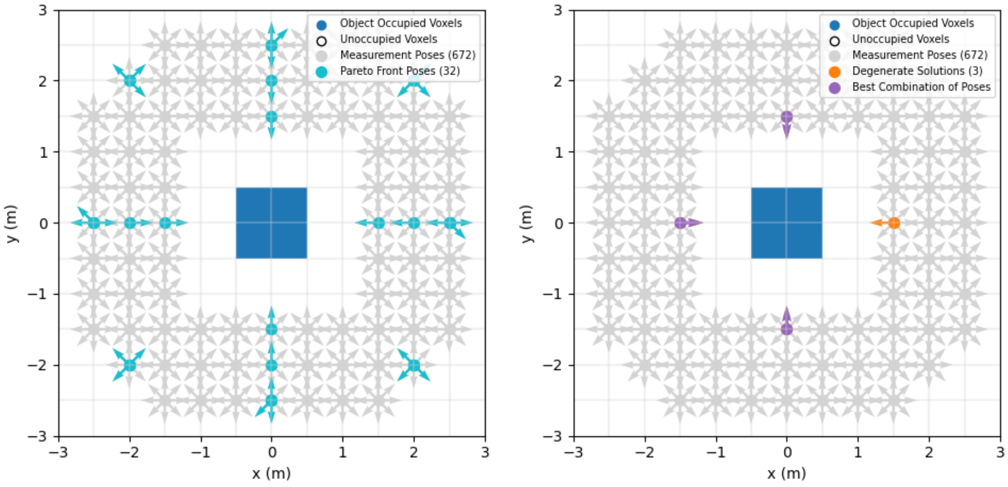
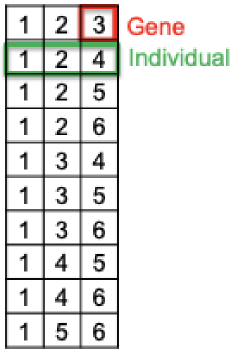


Figure 4.9: Pareto front poses in x-y plane (left) with optimal position identified by the Nash bargaining solution (right)

4.6.1 Genetic algorithms

Genetic algorithms are an optimization technique which mimics biological evolution to improve a population of potential solutions iteratively. Within the context of a GA, a 'population' consists of a number of 'individuals', each of which represent a potential solution to the optimization problem. Each individual is composed of 'genes', which is an encoded representation of the variables being evaluated for the optimization problem. Figure 4.10 demonstrates the concept of genes and individuals in an example where the global population consists of permutations of 3 for the integers 1 through 6. An individual is a single permutation. Each individual has three genes, which correspond to one of the integers.



1	2	3
1	2	4
1	2	5
1	2	6
1	3	4
1	3	5
1	3	6
1	4	5
1	4	6
1	5	6

Figure 4.10: Sample global population for GA

Genetic algorithms 'evolve' a population of individuals through six steps:

1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation
6. Termination

Figure 4.11 shows a diagram of how the genetic algorithm would evolve a problem initialized from the global population in Figure 4.10. In the first step, an initial population is randomly selected from a global population. In the figure, the GA's initial sample consists of 3 individuals from the global population. Note that none of the individuals sampled contain the integer 6, which is present in the global population. In the second step, each of the individuals in the sampled population are evaluated by a fitness function. The fitness function is the function which the optimization is seeking to minimize or maximize. The figure shows the solution to the fitness function for each individual in the sampled population. The third step of the algorithm, selection, selects the fittest individuals from the population. In this example, we are seeking maximization of the fitness function, so the first two rows in the sample population are selected. The crossover step of the algorithm allows the selected individuals to share their genes to create 'offspring'. Selected individuals and their offspring are kept in the sampled population. The example figure shows the selected individuals and an offspring in red. The fifth step of the algorithm is mutation, where genes may be randomly mutated into individuals. In the example figure, the integer 6 is randomly mutated into the individual in the second row. The mutation step marks the end of a 'generation' in the GA. The result of these steps is a new population of individuals. The new population is then evolved again with steps 2-5. The process is repeated until the desired number of evolutions is met, resulting in a final population that is more fit than the initial population.

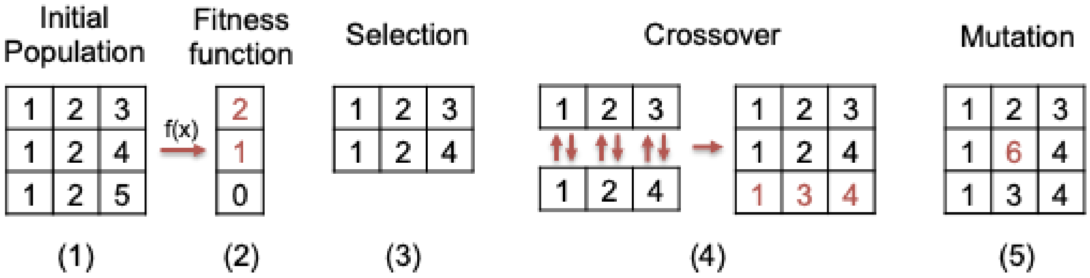
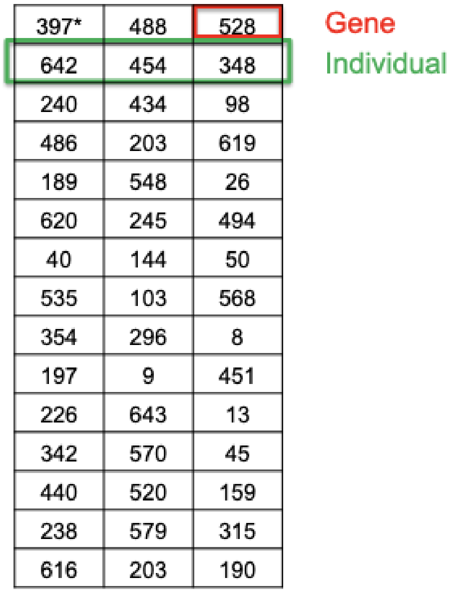


Figure 4.11: Diagram of genetic algorithm steps

4.6.2 Application of a genetic algorithm to this problem

To formulate the optimization problem in this work as a GA problem, measurement poses are treated as 'genes' and a combination of measurement poses is treated as an 'individual'. The global 'population' of individuals consists of all possible combinations of measurement poses. Only a small fraction of the global population is present in an initial population randomly sampled by the GA. Figure 4.12 illustrates an example random initial population of the example model optimization for use with a genetic algorithm. Each gene is an integer which corresponds to an index of one of the 672 measurement poses. Fifteen individuals are present in this figure, representing fifteen combinations of measurement poses out of the 50,803,424 possible combinations.

In this formulation, a random sample of combinations of measurement positions is taken for the initial population. This accelerates the calculation by skipping the full permutation of all possible measurement positions. The fitness functions are the optimization variables as used in the brute-force method - the mean and variance of the summed sensitivity across all of



397*	488	528
642	454	348
240	434	98
486	203	619
189	548	26
620	245	494
40	144	50
535	103	568
354	296	8
197	9	451
226	643	13
342	570	45
440	520	159
238	579	315
616	203	190

*Index of measurement pose

Figure 4.12: Example initial population of optimization problem for use with a genetic algorithm

the poses that make up the individual combinations represented in the population. The individuals selected are those combinations of poses with relatively high sensitivity mean and/or relatively low sensitivity variance. Crossover is performed on the selected combinations by exchanging measurement poses between the different selected combinations. Mutation is performed by swapping in random measurement poses from the global population, which may or may not have been present in the initial population. This prevents convergence to a local minima or maxima. This process is repeated until the desired number of iterations is achieved.

4.6.3 Genetic algorithm parameters

The main parameters for the GA are population size, crossover and mutation. Typically the initial population must be sufficiently large, or the mutation parameter sufficiently high to ensure all of the potential positions are explored by the algorithm. If the total number of positions represented in the initial population (the number of combinations multiplied by the number of positions in each combination) is around the same as the number of measurement positions available, the global population should be sufficiently represented. If there are more positions in each combination, more individual poses will be represented in the initial population. If the number of positions per combination is increased, the population size can be decreased. Conversely, the population should be increased if the number of positions per combination is decreased.

If the population is not sufficiently large, the mutation parameter can be tweaked to increase chances of seeing unrepresented poses without increasing computation time. Figure 4.13 shows the effect of changing the mutation parameter on how many evolutions it takes for the population to converge. Too high of a mutation parameter means that the population does not converge within 50 evolutions, as random new poses are being mutated in at a very high rate. Higher mutation rates correspond to a higher number of unique poses and combinations in the final population.

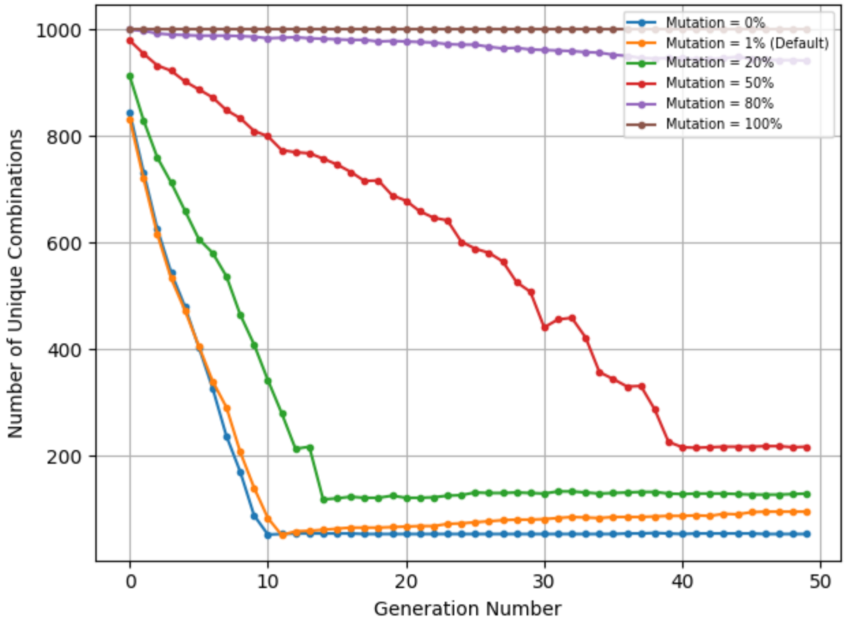


Figure 4.13: Number of unique individuals in a population for various mutation parameters

The crossover parameter can also be changed. Figure 4.14 shows the effects of changing the crossover parameter. The effect of changing the crossover parameter has less of an effect than changing the mutation parameter. The crossover rate is not strongly related to the uniqueness of the final population over 50 generations. However, an increased crossover rate does correlate to a slower convergence, but the effect is not significant.

For this work, the crossover parameter was determined to be sufficient at 95%. The mutation parameter was increased from the default value of 1% to 20% to increase computation speed while ensuring the entire population is sufficiently represented.

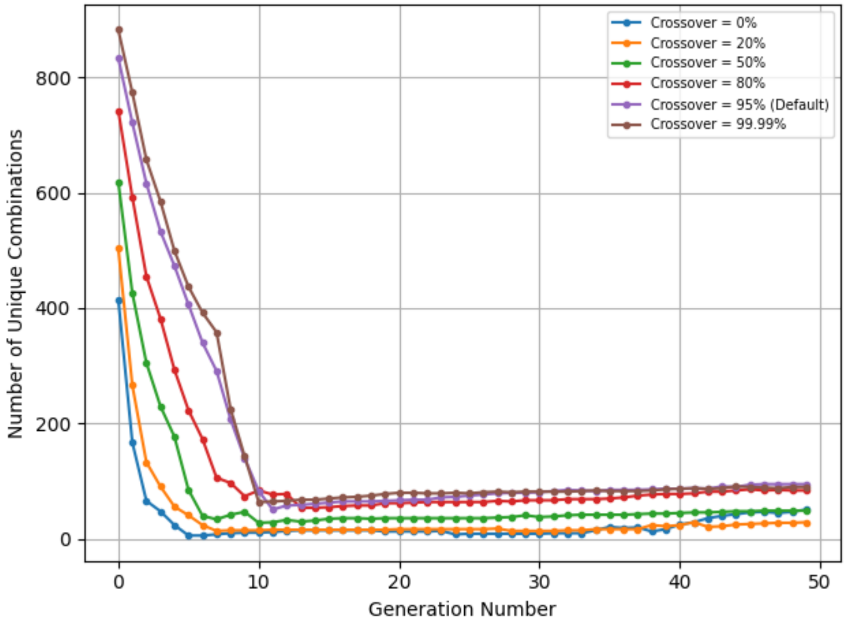


Figure 4.14: Number of unique individuals in a population for various crossover parameters

4.6.4 PyGMO optimization package

PyGMO [53] is an open-source Python library for massively parallel optimization with a number of optimization techniques available. It provides a uniform interface for optimization algorithms and problems, making it easy to explore different algorithms. The PyGMO package requires optimizations to be constructed into a 'Problem' class. The 'Problem' class requires class functions to characterize the optimization problem, including the fitness function and the bounds of potential function inputs. The custom 'Problem' class developed for this optimization problem is in Appendix C.2.

PyGMO includes a 'Population' class, which can be used to randomly sample a global population at a desired population size. The bounds of the population are defined in the 'Problem' class. To initialize a 'Population', the 'Problem' is loaded, and a 'Population' is loaded with the 'Problem' class and population size as input parameters, which samples a population based on the problem constraints. For this optimization problem, the population is sampled from a list of integers which index the potential measurement poses. The population is sized such that most of the potential poses are represented in the sample at least once. A random seed can also be used to ensure replicability in generating the initial population.

The Non-dominated Sorting Genetic Algorithm (NSGA-II) [54] is selected to perform the optimization because it is a genetic algorithm in PyGMO which allows for multi-dimensional optimization problems. The algorithm is initiated with a mutation parameter set at 20% to increase the likelihood that those poses not initially represented in the population are mutated into the population during one of the evolutions. The crossover parameter of the algorithm is left at the default value of 95%. The genetic algorithm is used to evolve the population 50 times. For each generation of the GA, the optimization metrics are computed on-the-fly for each of the combinations of poses in that generation. This means that the number of computations required to find the optimal position is the population size multiplied by the number of generations. For a population size of 1000 and 50 generations, this means only 50,000 computations are required, which is substantially fewer than the 1.67×10^8 computations required to calculate all potential combinations in the example at the beginning of this section.

4.6.5 Genetic algorithm applied to example model

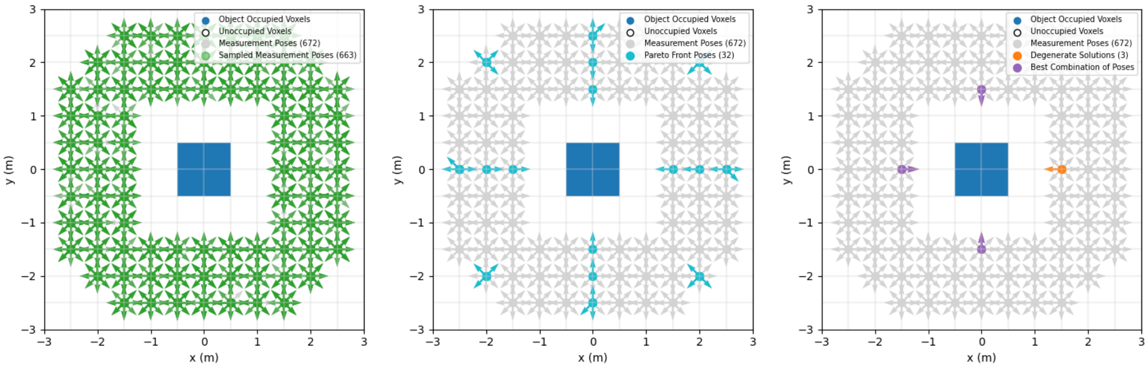


Figure 4.15: Initial population randomly sampled by the GA (left) and poses in Pareto optimal combinations after 50 evolutions (middle) with the combination of poses identified by the Nash bargaining solution (right)

The GA-accelerated optimization is used to find the optimal positions of the example model, and bench-marked against the brute-force method to demonstrate the feasibility of using this approach. Figure 4.15, left shows the initial population of 1000 combinations used in the optimization, where 663/672 potential poses are represented. The initial population is evolved for 50 generations, with the default crossover parameter and 20% mutation parameter to allow for mutation of the 9 unrepresented positions into the population. Figure 4.15 middle shows the final population on the x-y plane, where all of the poses lie on the Pareto front of the brute-force method as shown in Figure 4.9 left. This indicates that the final population is overall more fit, and comprised of the optimal poses within the global scope of the problem. Figure 4.15 right shows the optimal combination of positions selected by the Nash bargaining solution, which matches the optimal combination of positions selected by the brute-force method (including the same degenerate solutions). The GA approach only required 50,000 computations (population of 1000 evaluated over 50 generations), while the brute-force approach required 43,895,680 computations, demonstrating that the same result can be reached with significantly fewer computations when using a genetic algorithm. This validates that this problem is a suitable use of the GA, facilitating the use of this optimization framework in the field.

4.7 Discussion

In this chapter a framework for identifying optimal measurement positions for a radiation imaging measurement was developed and demonstrated with an example model. In this framework, a volume of interest is identified, either manually or through automatic object detection techniques. Next, all viable measurement poses are identified, with some constraints to ensure physical feasibility. Then, the sensitivity to each of the volume of interest voxels from each of the potential measurement poses is computed. Finally, the mean and variance of the sensitivity are computed to serve as the optimization variables in a multi-objective optimization. The Pareto front is used to identify the set of optimal solutions, and the Nash bargaining solution is used to identify a single optimal solution from the set. While this approach is straightforward for a simple example model, real-world measurement scenarios are much more complicated and demand fast and simple computation to enable on-board real-time optimization. Therefore, the genetic algorithm is used to accelerate the optimization to facilitate its use in the field.

One drawback of using a GA is that there is not a guarantee to arrive at the same solution as the brute-force method. However, due to the nature of multi-objective optimization having multiple optimal solutions, it is very likely that the solution determined by a GA will still be a Pareto optimal solution. While the solution might not always be identical to the brute-force method, the time savings of using algorithmic approaches enables the use of these methods in real-time.

Another drawback of using a GA to solve this problem is that the parameters of the GA might need to be altered in different scenarios, which would require operator knowledge of how the algorithm works. In the example model example, 602 out of 608 measurement poses were represented in the initial sample. If significantly fewer measurement poses were represented, either the population size or the mutation parameter should be increased to more effectively represent the global population. However, if the population size is increased too much the computation time might become infeasible, and if the mutation parameter is set too high, the population may not converge. Algorithmic approaches other than the GA, such as gradient-free algorithms, could be investigated as additional means to solve this problem, which might require less parameter tweaking.

Chapter 5

Simple models

Chapter 4 laid out the framework for optimizing static measurement positions for a quantitative gamma-ray imaging measurement. Section 4.1 laid out an example model which was used to explain the various steps of the optimization. This model was used to develop the optimization code and verify that the various software features were performing as expected. However, the simple, symmetrical nature of the model can not accurately simulate complicated real world measurements. In addition to the simple model laid out in the previous chapter, other simple models were created and investigated to develop more complex scenarios to validate various features of the optimization.

5.1 Model 2: box with internal voxels

The problem explored in Chapter 4 consisted of a simple 1 m x 1 m x 1 m box with a 0.5 m voxel pitch. This coarse voxelization allowed for all of the object voxels to be visible at some measurement position in the scene. In real world measurements, objects may be very large, requiring voxelization to be set such that some voxels are internal to the object and not visible at any measurement positions. In this section, a simple model which contains internal voxels is explored.

5.1.1 Model definition

This model consists of the same box as in Chapter 4, except with a finer voxel pitch of 0.25 m. This will help us explore the effects of self-shielding on the algorithm. The model now consists of 64 occupied voxels, and all other voxels in the scene are again assumed to be unoccupied. The 64 occupied voxels are assumed to be uniformly distributed with a source which emits 662 keV gamma rays. 8 of the occupied source voxels are internal to the object and not visible at any positions outside of the object. Figure 5.1 shows the voxelization of the measurement space in the x-y and y-z plane.

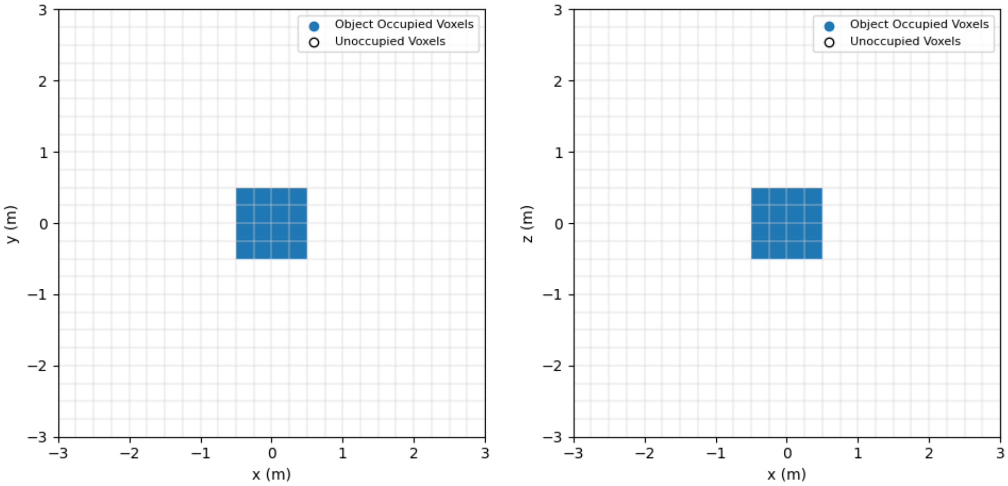


Figure 5.1: x-y (left) and x-z (right) plane of model 2

5.1.2 Optimization results

The GA-accelerated optimization described in the previous chapter is performed on this example model, with the 64 occupied voxels considered to be the volume of interest. The measurement position selection is again given the constraint that all positions are on-axis with the center of the object with a maximum standoff distance of 3 m. The source voxels are assumed to be composed of water, and the optimization assumes a gamma-ray energy of 662 keV. Figure 5.2 left shows the 672 potential measurement poses (84 positions × 8 orientations) identified by the optimization. The positions identified are identical to the positions identified for the model in Chapter 4 (see Figure 4.3). This is expected because the measurement scenario is physically the same, but the voxel sizes are smaller. Figure 5.2 right shows how many voxels are visible to each of the potential measurement poses. While 64 voxels are present, only 56 voxels are present on the surface of the object and possible to be seen at any of the potential poses. This demonstrates the importance of the visibility cut discussed in Section 4.3. Typically, a cutoff value of 1 is used, meaning that at least 1 object voxel must be visible for a pose to be considered. The value can be increased, but if it is set too high, (e.g., 64 voxels) all potential poses will be removed from consideration.

Figure 5.3 shows the mean sensitivity at each x-y-z position to each object voxel. When compared to the sensitivity calculated for the example model in Chapter 4 (see Figure 4.4), we can see that the mean sensitivity values are higher for each position. Due to self-shielding and increased distance, the sensitivity to internal voxels is lower than sensitivity to external voxels. Figure 5.4 shows the contribution to sensitivity for the 8 internal voxels and 56 external voxels scaled by the number of voxels in each group. We can see that the sensitivity to the internal voxels is an order of magnitude smaller than the sensitivity to external voxels.

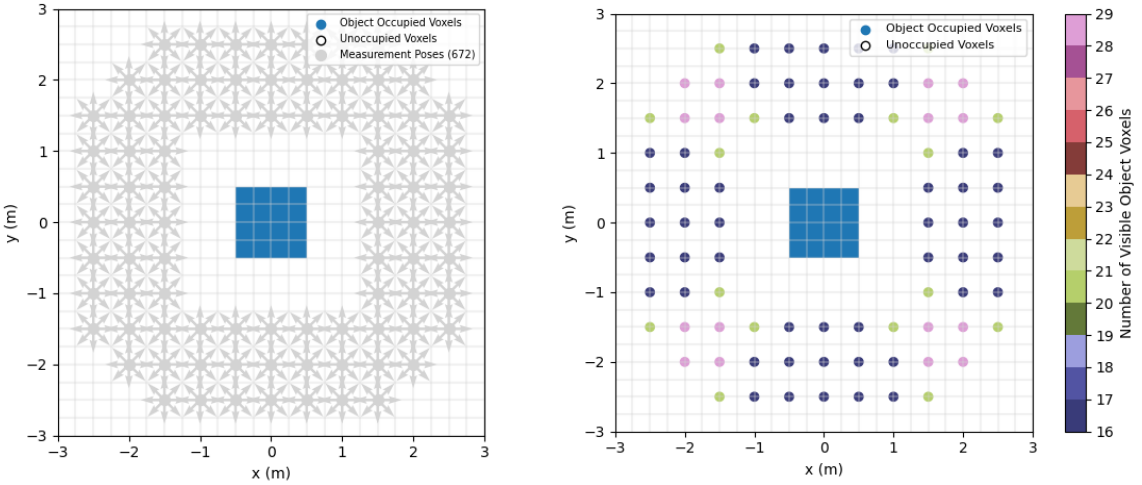


Figure 5.2: Potential measurement poses identified (left) and number of object voxels visible at each position (right) for example model 2

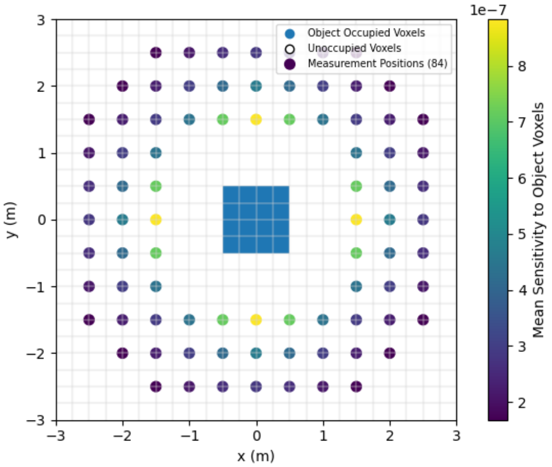


Figure 5.3: Mean sensitivity to object voxels for model 2

This is because the mean free path of a 662 keV photon through water is around 10 cm, which is slightly less than half of a voxel size for this example model.

To determine the best combination of measurement positions, the genetic algorithm is used with the default crossover parameter and the mutation parameter set to 20%. The initial population consists of 1000 combinations of poses, and the population is evolved for 50 generations. Figure 5.5 shows the optimization metrics for all potential combinations in

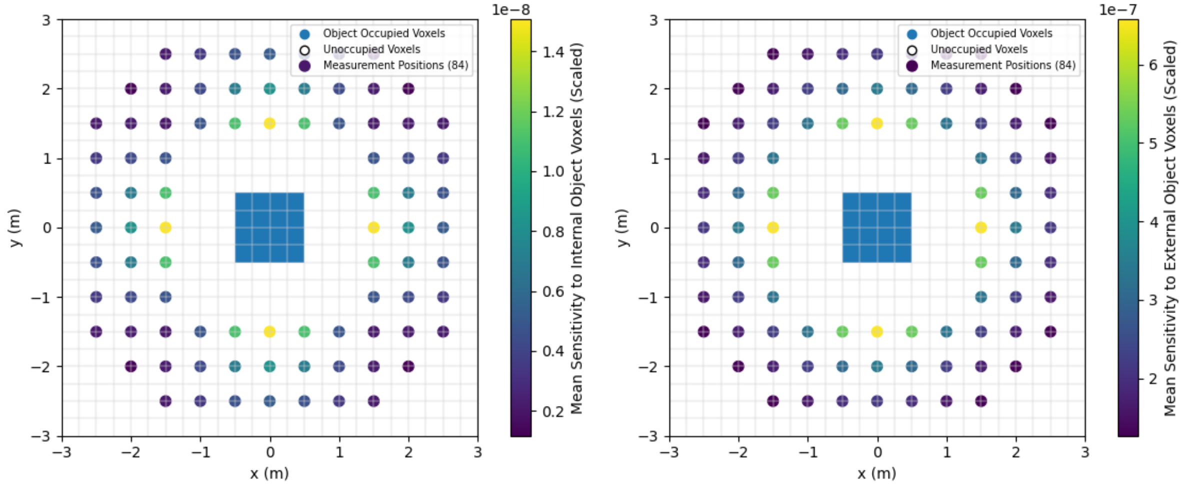


Figure 5.4: Mean sensitivity to object voxels for internal voxels (left) and external voxels (right) for example model 2 scaled by the number of voxels per group

gray and the metrics that lie on the Pareto front of the final population in cyan. The Nash bargaining solution is used to identify the best combination of positions, which is shown in purple on the Pareto front plot. Figure 5.6 shows the optimal poses in the x-y plane of the scene. This solution is identical to the Chapter 4 solution.

5.1.3 Effects of increased attenuation

When the object of interest is assumed to be composed of water and the gamma-ray energy is assumed to be 662 keV, the self-shielding effect is appreciable. If the attenuation value is increased, the effect of attenuation becomes more dominant. The optimization of this problem is performed again with the same parameters previously described, except the object is assumed to be composed of UF_6 (density $= 5.09\ g/cm^3$) with a linear attenuation coefficient of 56.7 m^{-1}, which is an order of magnitude higher than the linear attenuation coefficient of water.

The potential poses are the same as for water, shown in Figure 5.2 left. Figure 5.7 shows the mean sensitivity to all object voxels at each potential measurement pose. The sensitivity is two orders of magnitude lower than what is shown in Figure 5.3 for water. Figure 5.8 shows the contribution to sensitivity for the 8 internal voxels and 56 external voxels scaled by the number of voxels in each group. The sensitivity to internal voxels is seven orders of magnitude lower than the sensitivity to external voxels, which is a much greater difference than Figure 5.4 where the occupied voxels are filled with water. The mean free path of a 662 keV gamma ray in UF_6 is around 1.8 cm, so gamma rays emitted from internal voxels must travel through at least 14 mean free paths to exit the object of interest.

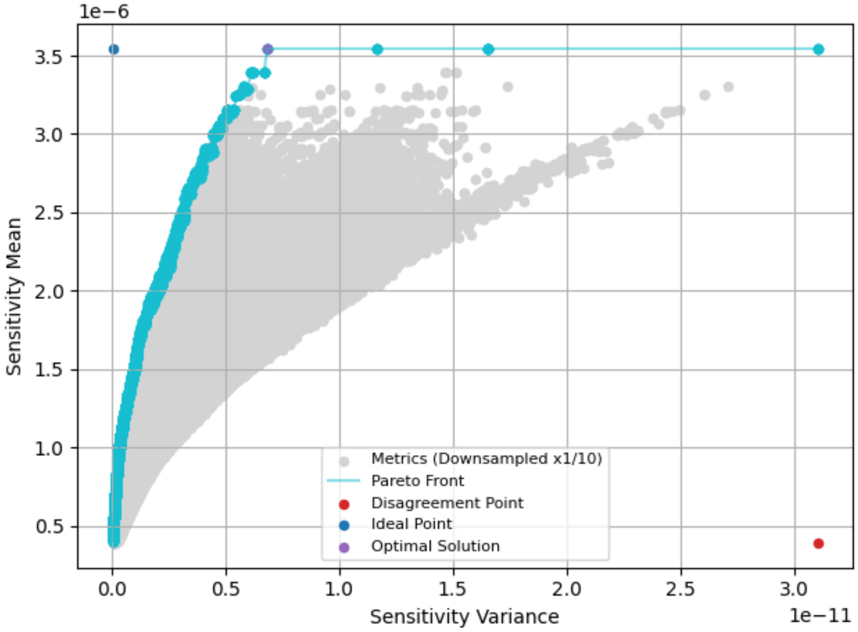


Figure 5.5: Pareto front for example model 2

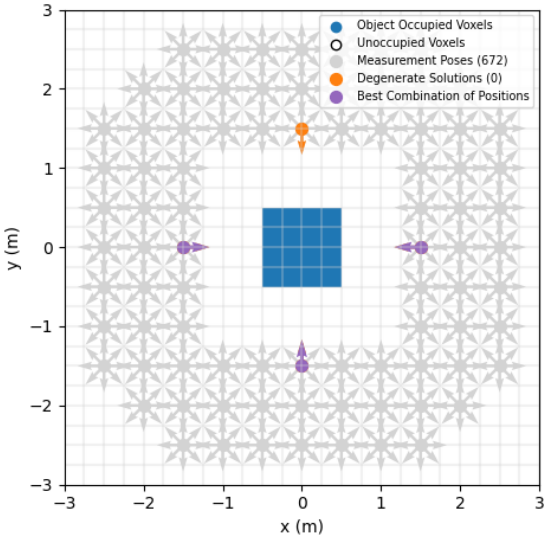


Figure 5.6: Optimal poses selected for example model 2

The GA parameters used for the water scenario are also used to optimize this example. Figure 5.9 shows the optimization metrics in gray with the metrics of combinations which lie

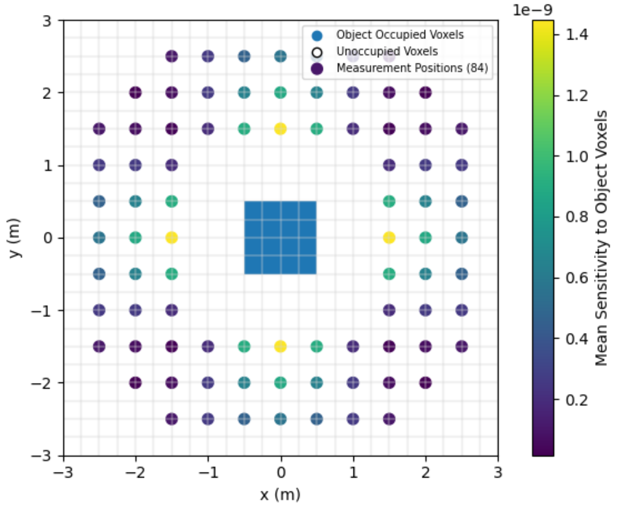


Figure 5.7: Mean sensitivity to object voxels for model 2 with increased attenuation

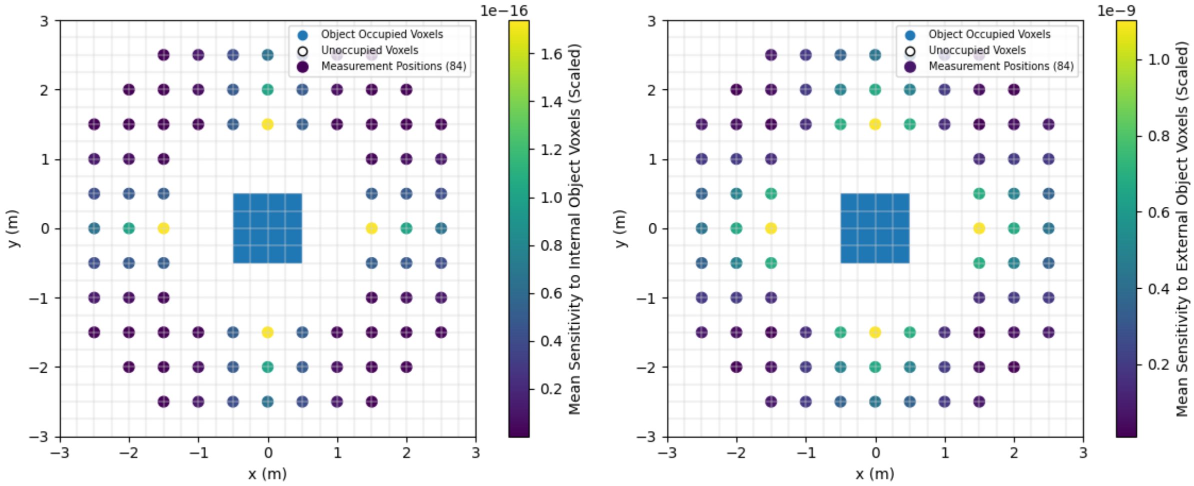


Figure 5.8: Mean sensitivity to object voxels for internal voxels (left) and external voxels (right) for example model 2 with increased attenuation scaled by the number of voxels per group

on the Pareto front in cyan. The metrics of the combination identified by the Nash bargaining solution are shown in purple. Compared to the optimization with water, the sensitivity mean and variance is lower for all of the metrics. Figure 5.10 shows the combination identified by the Nash Bargaining solution on the x-y plane. The poses are the same, but due to the

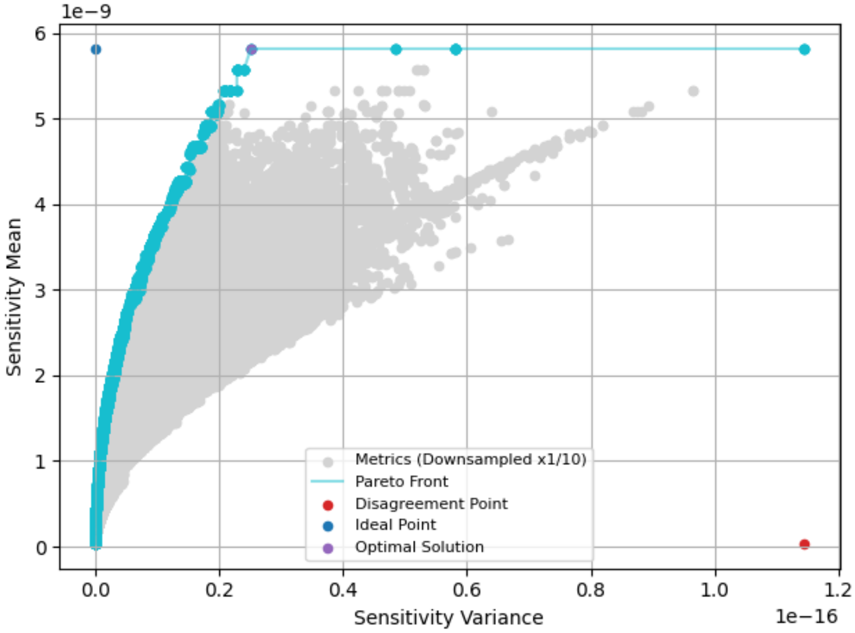


Figure 5.9: Pareto front for example model 2 with increased attenuation

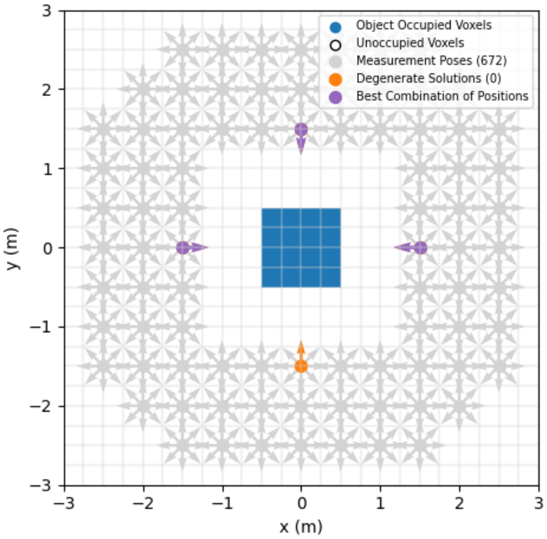


Figure 5.10: Optimal poses selected for example model 2 with increased attenuation

decreased sensitivity from self-shielding, the counting measurement should be performed for a longer period of time.

5.2 Model 3: external shield

The models explored so far have consisted of scenes where the only occupied material is the volume of interest. In real measurements, there is often a number of intervening materials which constrain the potential measurement positions and potentially attenuate the radiation signatures. Therefore, it is important to investigate the effect of external shielding on the optimization procedure.

5.2.1 Model definition

We will now consider the same box as Chapter 4, except with an external shield 0.5 m to the left of the left face of the object. The shield is 0.5 m wide with a length of 3 m and a height of 1 m on axis with the box. The object is assumed to be water, and the shield is assumed to be concrete, a highly attenuating material with a linear attenuation coefficient of 15861 m^{-1}. Figure 5.11 shows the voxelization of the example model.

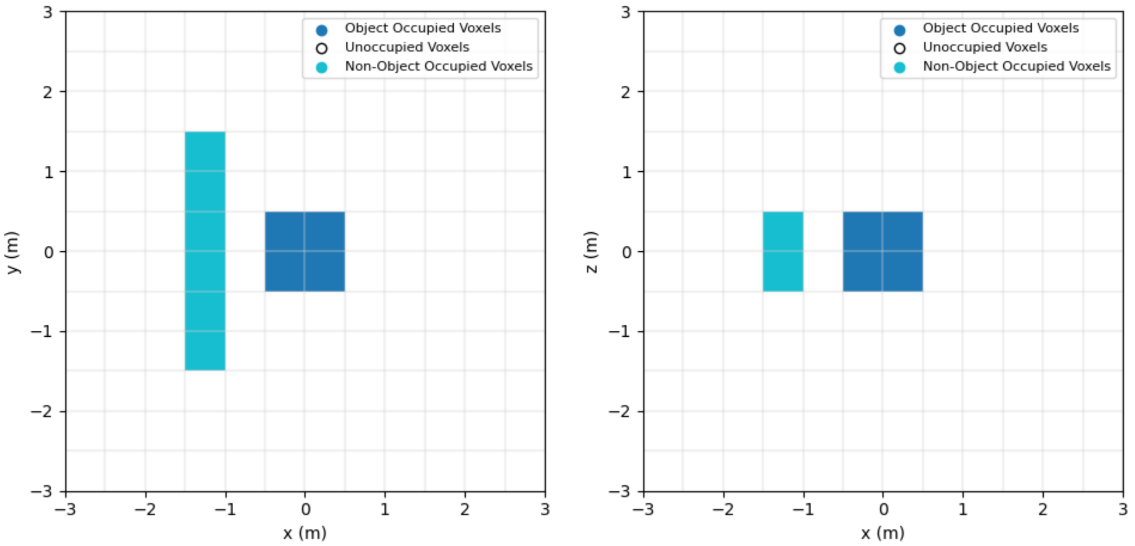


Figure 5.11: x-y (left) and x-z (right) plane of example model 3

5.2.2 Optimization results

The GA-accelerated optimization described in the previous chapter is performed on this example model, with the 8 central voxels considered to be the object of interest. The measurement position selection is given the constraint that all positions are on-axis with the center of the object with a maximum standoff distance of 3 m. Figure 5.12 left shows the potential poses which were identified by the optimization. The initial candidate pool is identical to the previously discussed models, but the positions to the left of the external shield which cannot see any of the object voxels were removed from consideration as discussed in Section 4.3. This results in 61 potential measurement positions with 8 potential orientations, for a total of 488 potential poses. Some of the poses can have line of sight to at least one of the object voxels, but experience some attenuation from the external shield.

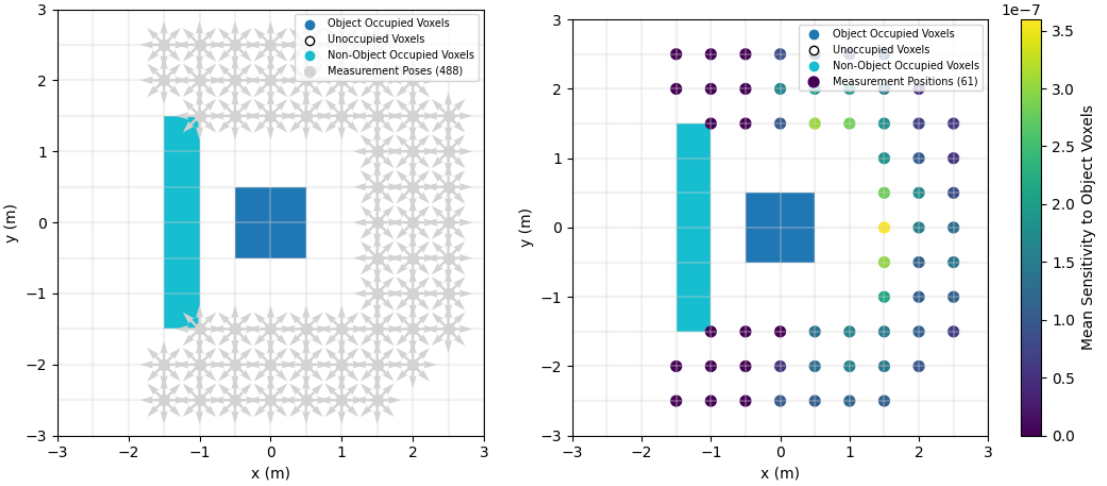


Figure 5.12: Potential measurement poses identified (left) and mean sensitivity to object voxels (right) of example model 3

When calculating sensitivity, the non-object occupied voxel term of Equation 4.1 is used due to the presence of an external shield. Figure 5.12 right shows the mean sensitivity at each x-y-z position to the object voxels. Most of the positions have the same sensitivity as what was calculated in Chapter 4 (see Figure 4.4); however, the positions which have attenuation from the shield wall have lower sensitivity than their symmetric counterparts. This is due to the high attenuation coefficient ascribed to the external shield.

To determine the best combination of measurement positions, the genetic algorithm is used with the default crossover parameter and the mutation parameter set to 20%. The initial population consists of 1000 combinations of poses, and the population is evolved for 50 generations. Figure 5.13 shows the optimization metrics for all potential combinations in gray and the metrics that lie on the Pareto front of the final population in cyan. The measurement positions which are attenuated by the concrete wall are/are not present in any of the Pareto front combinations due to the large decrease in sensitivity from the concrete shielding. The Nash bargaining solution is used to identify the best combination of poses, which is shown in purple on the Pareto front. Figure 5.14 shows the poses identified by the Nash bargaining solution on the x-y plane. The solution is one of the degenerate solutions of the example model in Chapter 4 (shown in Figure 4.9). The previous model had a number

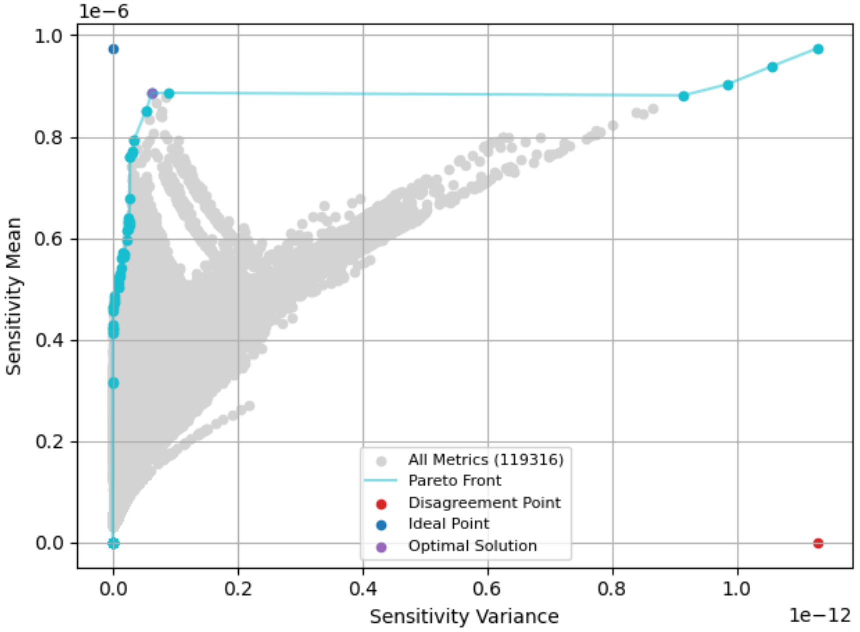


Figure 5.13: Pareto front for example model 3

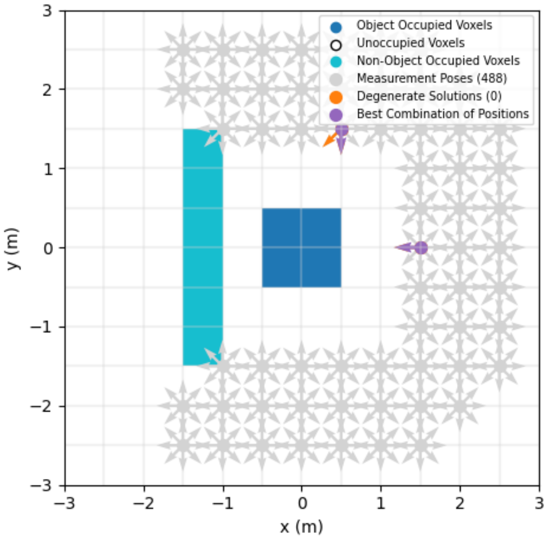


Figure 5.14: Optimal poses selected for example model 3

of degenerate solutions due to the symmetric nature of the problem. Because this problem is asymmetrical with a shield on the left side, there is only one solution identified.

5.2.3 Effect of removing visibility restriction

The visibility constraint described in Section 4.3 serves to remove positions which are physically infeasible or undesirable. This restriction can be removed, allowing measurement positions on the other side of the concrete wall to be considered. We still remove positions from inside the wall as they are considered physically infeasible, rather than just undesirable. Figure 5.15 left shows the potential poses for this example model when the visibility constraint is removed. Some of the poses which were present in the Chapter 4 example are removed from consideration because they reside in the voxels occupied by concrete. This constraint serves to remove positions which are inside of objects and therefore infeasible for placing the radiation imager. Figure 5.15 right shows the mean sensitivity for each x-y-z position to all of the object voxels. The positions to the left of the wall which have been added to consideration have a mean sensitivity of zero, unlike their mirrored counterparts due to the presence of the shielding wall. Figure 5.16 shows the mean distance traversed through the concrete for a voxel emitted by one of the 8 source voxels to each measurement position. The mean sensitivity in the measurement positions to the left of the wall are zero because the mean free path of a 662 keV gamma ray in concrete is around 64 μm, which is four orders of magnitude smaller than the voxel pitch in this example model. The figure shows that gamma rays would have to traverse approximately 50 cm of concrete to reach positions to the left of the concrete wall.

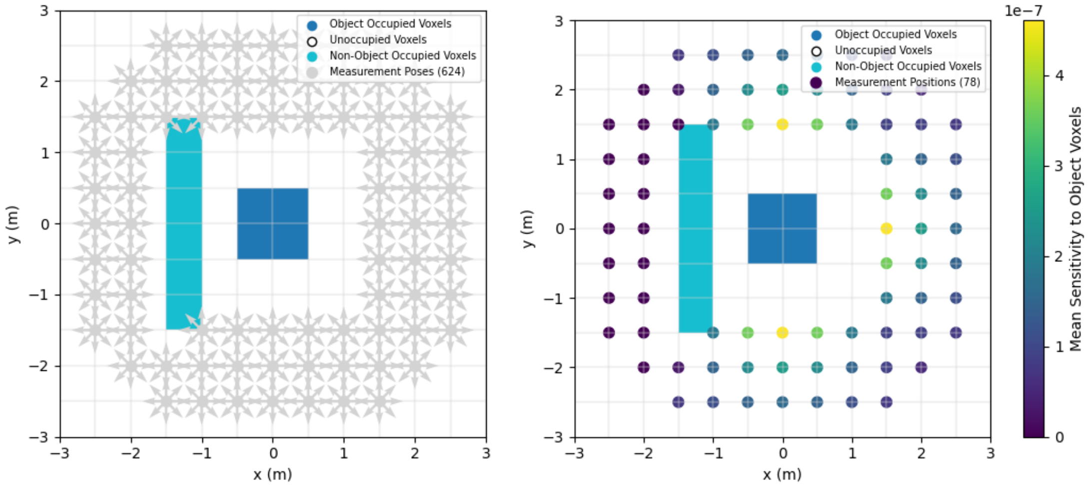


Figure 5.15: Potential poses identified (left) and sensitivity (right) for example model 3 when the visibility constraint is removed

The GA parameters used for the constrained example are used to optimize this unconstrained example. Figure 5.17 shows the optimization metrics of all potential combinations in gray and the metrics which lie on the Pareto in cyan. The metrics of the combination

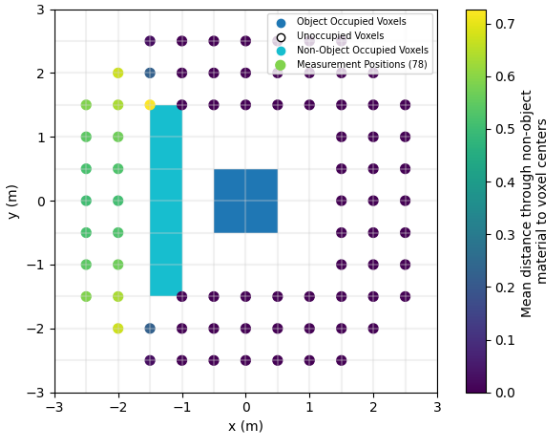


Figure 5.16: Sum of distance traversed through non-object material for example model 3 with the visibility constraint removed

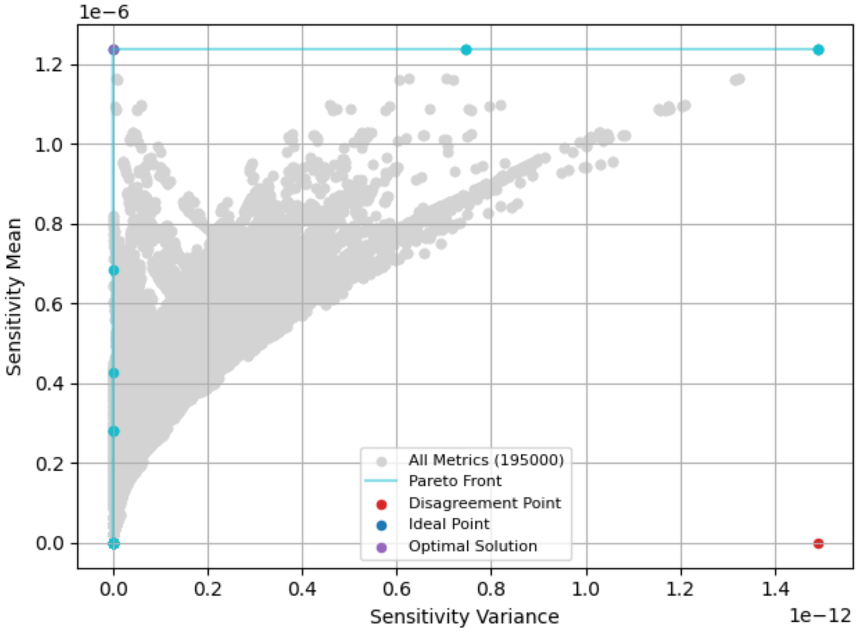


Figure 5.17: Pareto front for example model 3 with unconstrained positions

identified by the Nash bargaining solution is shown in purple. Figure 5.18 shows the poses identified by the Nash bargaining solution in the x-y plane. This solution is the same as the constrained solution in Figure 5.14. This means that it is feasible to use the visibility constraint because it removes poses with a low sensitivity which are not desirable for a counting measurement. Adding this constraint simplifies the computation because there are less poses to consider, and less potential combinations to optimize over.

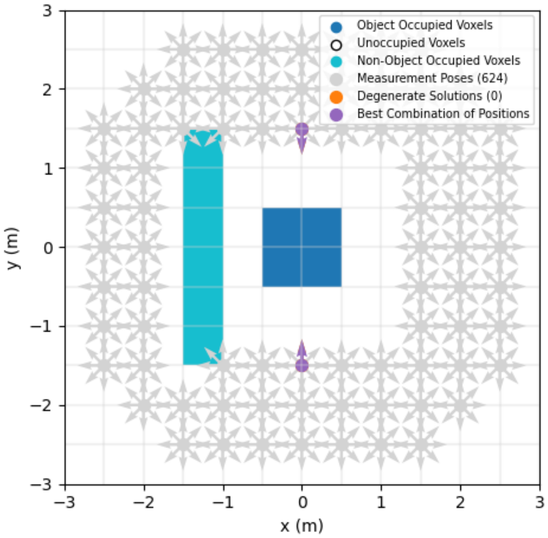


Figure 5.18: Optimal poses selected for example model 3 with unconstrained positions

5.3 Model 4: two separated boxes

The example models presented so far consist of scenes where is only one object of interest present. In some cases, it may be preferred to optimize for two or more objects. An example would be a holdup scenario with two containers right next to each other, where one has nuclear material and the other is declared to be empty. Rather than optimizing and measuring both separately, it would save time to measure both simultaneously.

5.3.1 Model definition

We will now consider a model where there are two 1 m^3 cubic boxes separated by 1 m. The scene is extended to 10 m × 10 m × 10 m, with a voxel pitch of 0.5 m. Both objects are assumed to be composed of water. The model consists of 16 occupied voxels total, with all other voxels in the scene assumed to be unoccupied. The 16 occupied voxels are assumed to be uniformly distributed with a source which emits 662 keV gamma rays. Figure 5.19 shows the voxelization of the measurement space.

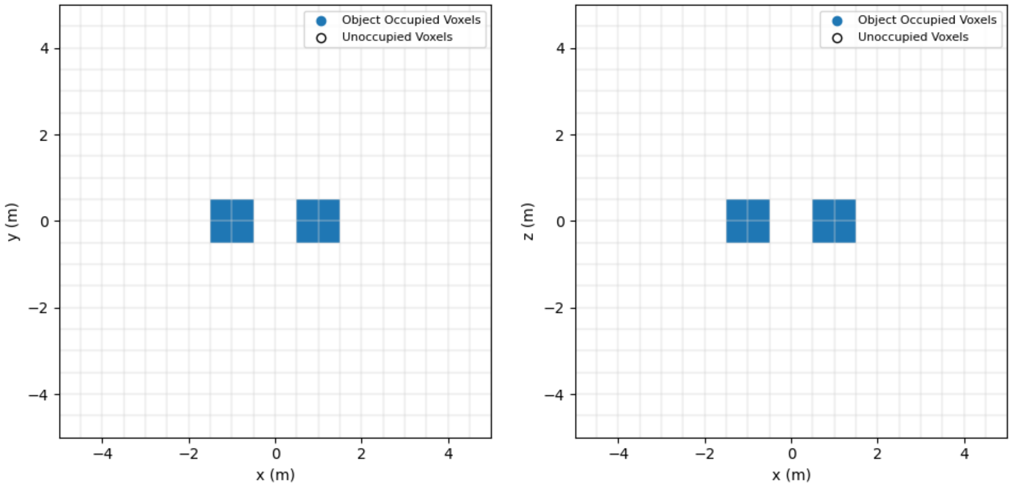


Figure 5.19: x-y (left) and x-z (right) plane of example model 4

5.3.2 Optimization results

The GA-accelerated optimization described in the previous chapter was performed on this example model, with the two separated 8 voxel boxes considered to be the combined object of interest. The measurement position selection is given the constraint that all positions are on-axis with the center of the objects in the z-axis with a maximum standoff distance of 3 m. In this case, standoff distance is defined as the maximum standoff from the closer of the two boxes to a given position. Figure 5.20 left shows the 576 poses (72 positions × 8 orientations) identified by the optimization.

Figure 5.20 right shows the mean sensitivity at each x-y-z position to the object voxels. The positions to the left and right of the objects experience more attenuation due to the presence of the second object acting as intervening material. Because of this the positions on the top and bottom of the two volumes in the y-axis have a relatively high sensitivity.

To determine the best combination of measurement positions, the genetic algorithm is used with the default crossover parameter and a mutation parameter of 20%. The initial population consists of 1000 combinations and the population is evolved for 50 generations. Figure 5.21 shows the optimization metrics for all potential combinations in gray and the metrics which lie on the Pareto front in cyan. The Nash bargaining solution is used to identify the best combination of positions, which is shown in purple on the Pareto front plot. Figure 5.22 shows the poses in the combination identified by the Nash bargaining solution in purple with degenerate solutions in orange. The positions selected are on the top and bottom of the box in the y-axis, as expected due to their relatively high sensitivity.

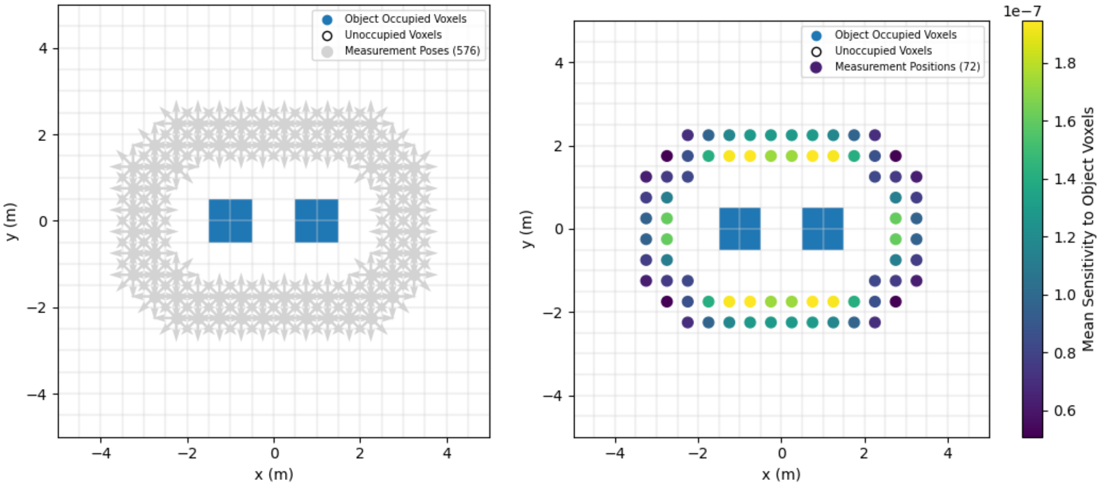


Figure 5.20: Potential measurement poses identified (left) and mean sensitivity to object voxels (right) of example model 4

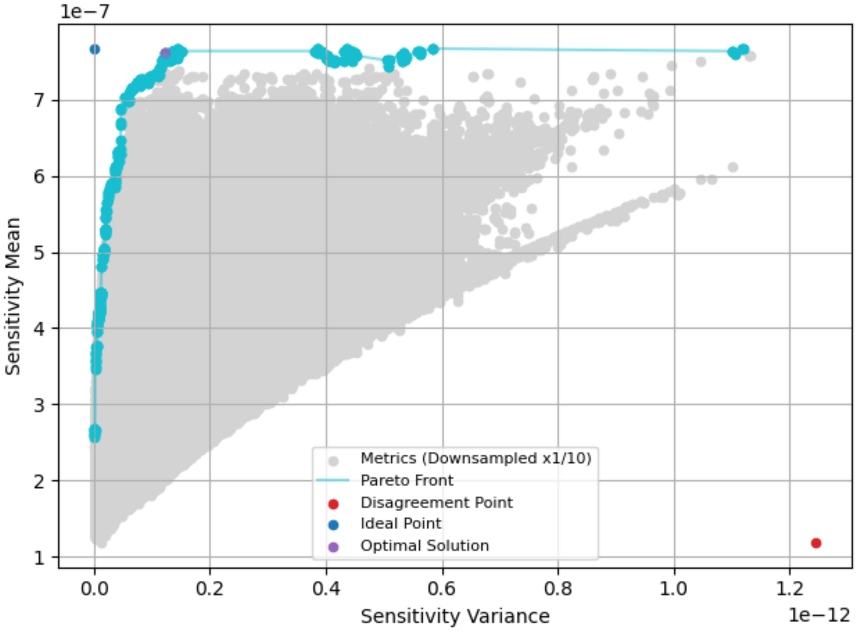


Figure 5.21: Pareto front for example model 4

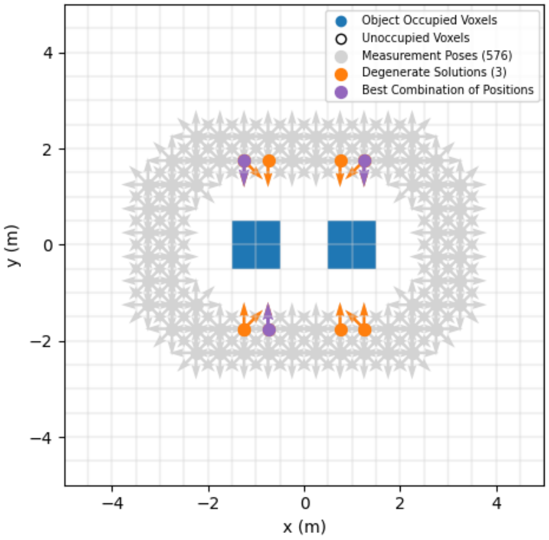


Figure 5.22: Optimal poses selected for example model 4

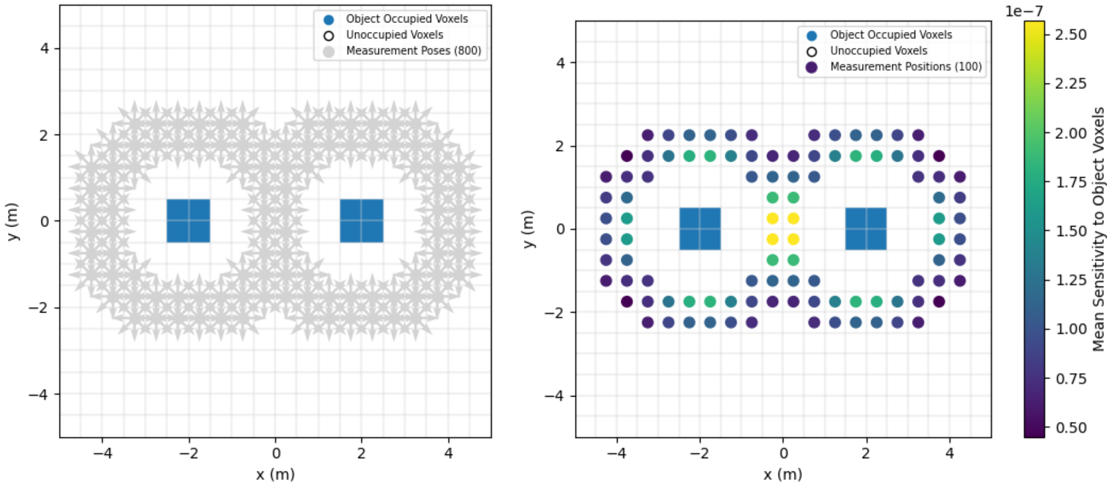


Figure 5.23: Potential measurement poses (left) and sensitivity (right) identified for example model 4 with increased object separation

5.3.3 Effect of increasing distance between object volumes

The objects in this example model were placed 1 m from one another, and with a minimum standoff distance of 1 m imposed by the optimization, no poses between the two objects were considered. To explore the result of increasing the separation distance of the two objects, the spacing between the two objects is increased to 3 m and the optimization is computed again. The size and composition of the two boxes remain the same.

Figure 5.23 left shows the new measurement scene with the objects in blue and potential poses in gray. With the increased separation distance, there are potential poses between the two objects. Figure 5.23 right shows the mean sensitivity of all of the x-y-z positions to the object voxels for this scenario. As expected, the sensitivity of the positions between the two positions is much higher than the sensitivities on any other side of either of the objects.

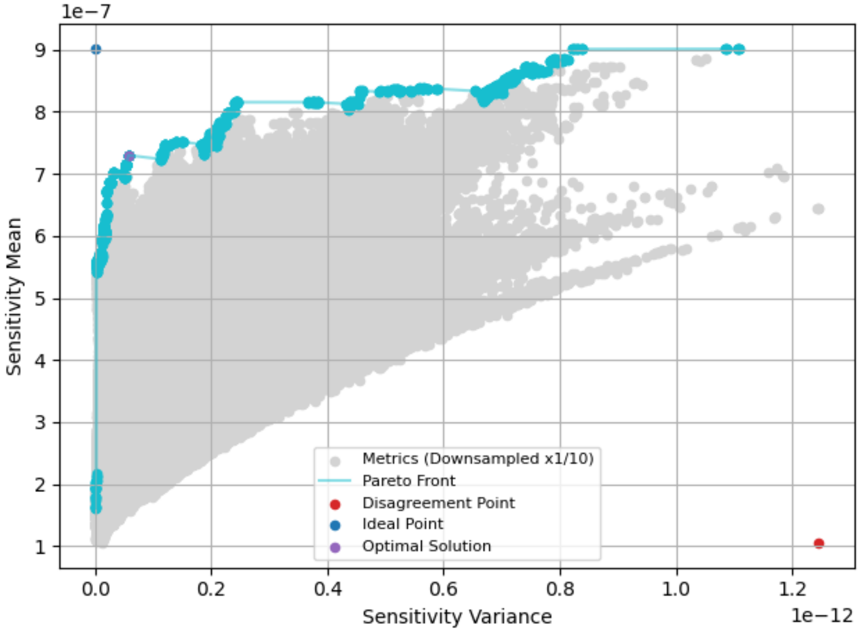


Figure 5.24: Pareto front of combinations of three poses for example model 4 with increased object separation

The GA parameters used to optimize the model with a 1 m separation distance are also used for this optimization. Figure 5.24 shows the optimization metrics for all potential combinations of measurement poses in gray and the metrics of combinations which lie on the Pareto front in cyan. The combination identified by the Nash bargaining solution is shown in purple. Figure 5.25 left shows the poses in the optimal combination in the x-y plane.

The optimization was also performed for all combinations of 2 positions to compare to this solution. Figure 5.26 shows the the optimization metrics for all potential combinations in gray and the metrics for combinations which lie on the Pareto front in cyan. While the sensitivity mean is not substantially worse for this case, the sensitivity variance is significantly degraded. This is because for a case with two separated volumes, it is desirable to have at least two view angles for each of the two volumes, one of which can be shared between the two. Because there are two volumes in this example model, there should be at least three

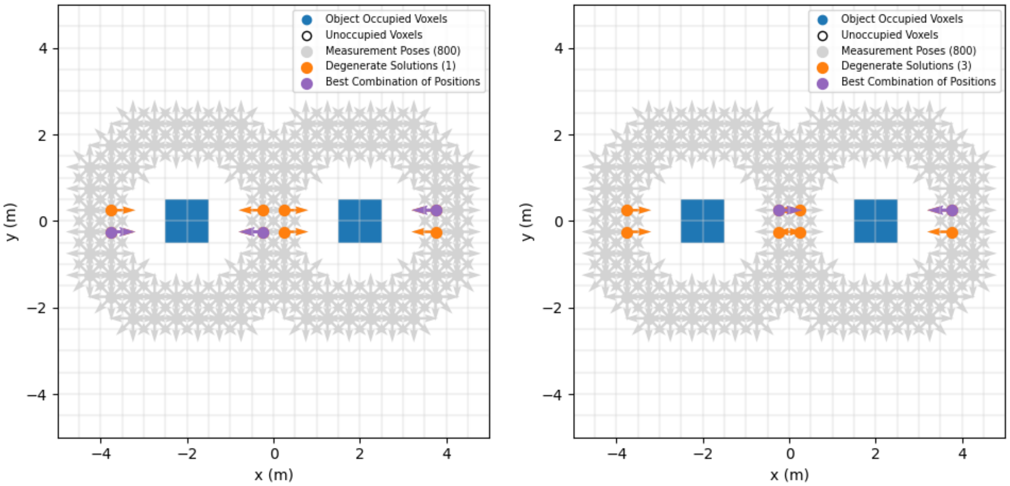


Figure 5.25: Three (left) and two (right) optimal poses selected for example model 4 with increased object separation

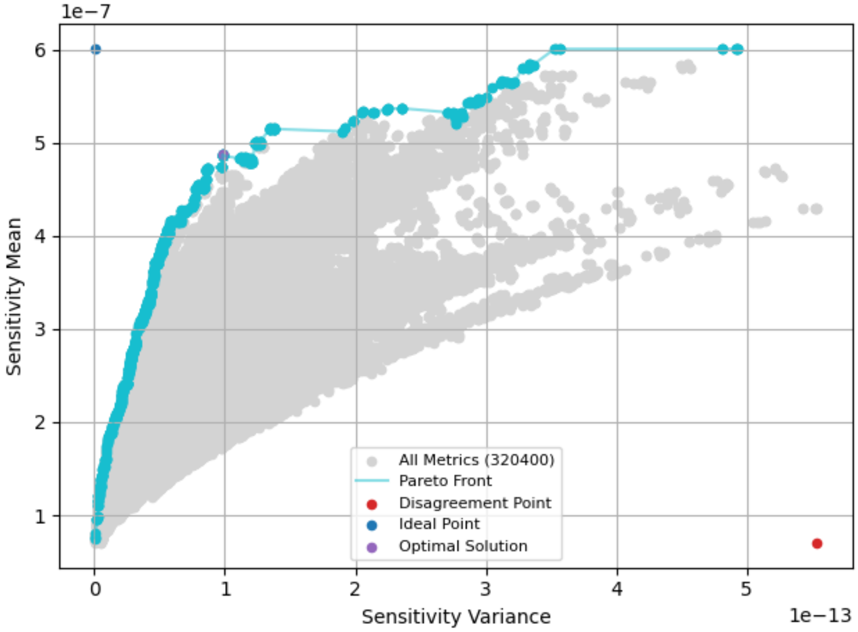


Figure 5.26: Pareto front of combinations of two poses for example model 4 with increased object separation

poses in the optimal combination to reduce the variance in sensitivity and ensure uniform exposure to both of the objects. Figure 5.25 right shows the poses identified by the Nash bargaining solution.

5.4 Model 5: asymmetrical object

All of the models explored so far consist of one or more uniform volumes. However, in the real world, objects may not be symmetrical. Therefore it is important to study the effects that an asymmetrical object has on the optimization.

5.4.1 Model definition

This model is similar to the example model in Chapter 4, except two of the object voxels are removed such that the object is asymmetrical. The extent of the object is 1 m x 1 m x 1 m, except for the removed voxels. The voxel pitch is again 0.5 m. This results in a total of 6 occupied voxels which are assumed to be composed of water, and all other voxel in the scene are assumed to be unoccupied. The 6 occupied voxels are assumed to be uniformly distributed with a source which emits 662 keV gamma rays. Figure 5.27 shows the voxelization of the measurement space. The transparency of the object voxels in the x-z plane image has been increased for this figure to display the location of the removed voxel.

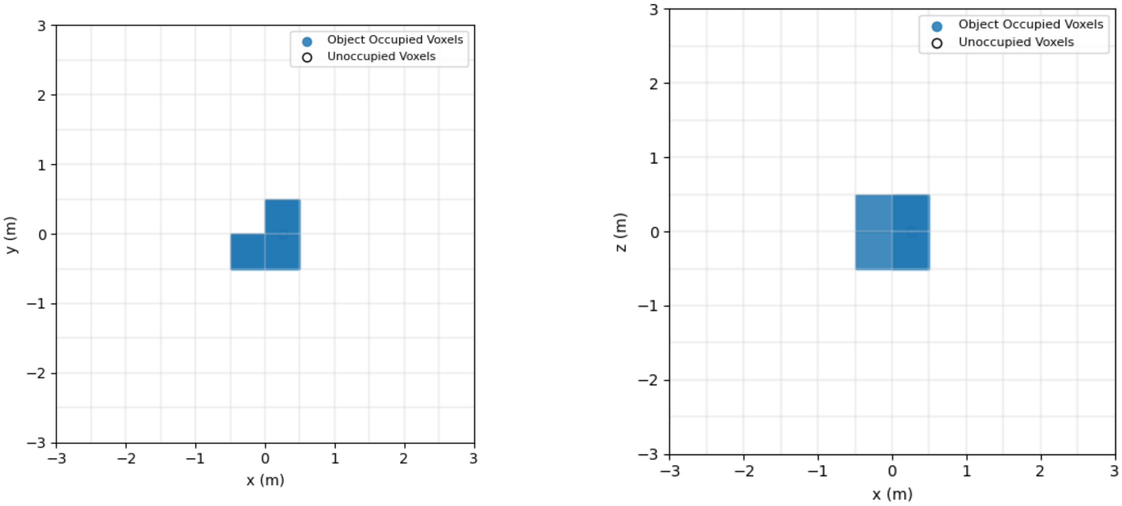


Figure 5.27: x-y (left) and x-z (right) plane of example model 5

5.4.2 Optimization results

The GA-accelerated optimization discussed in the last chapter is performed on this example model, with the 6 occupied source voxels as the object of interest. The measurement position selection is given the constraint that all positions are on-axis with the center of the object in the z-axis with a maximum standoff distance of 3 m. Figure 5.28 left shows the poses identified by the optimization. There are 81 potential x-y-z positions and 8 potential orientations, resulting in 648 potential poses. There are fewer positions than the example in Chapter 4 because the top left voxels in the x-y plane are gone, so the maximum standoff distance filters out positions in the second quadrant. The minimum standoff distance brings some of the potential poses closer than the previous examples.

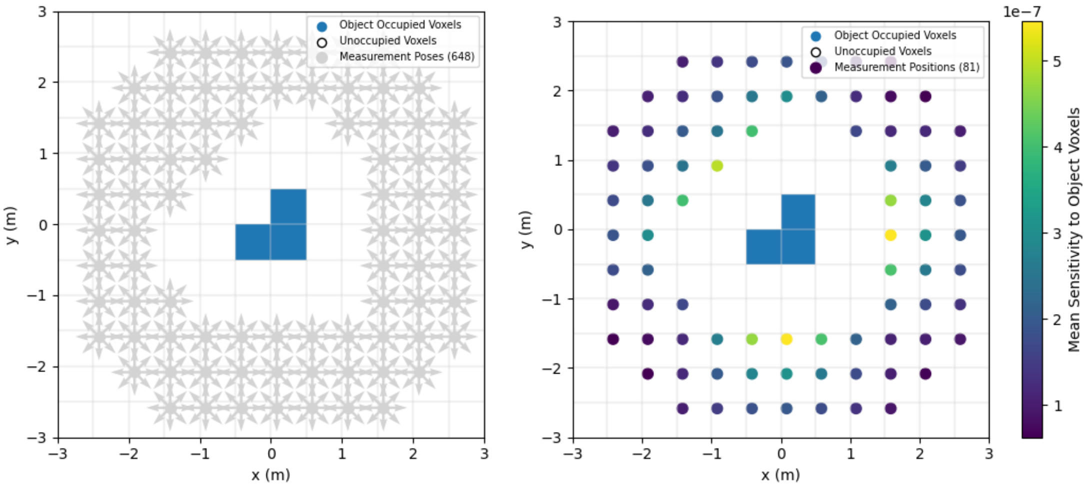


Figure 5.28: Potential measurement poses(left) and mean sensitivity to object voxels (right) plane of example model 5

Figure 5.28 right shows the mean sensitivity at each x-y-z position to the object voxels. The positions closest to the removed voxels are overall further from the voxels that are present so their sensitivities are reduced. However, they have direct line of sight to more voxels and less attenuation in intervening material, so the sensitivity is not dramatically reduced. The maximum mean sensitivity of the positions is approximately the same as what was calculated in Chapter 4 (see Figure 4.4), but the summed sensitivity is lower due to two of the voxels being removed.

To determine the best combination of measurement positions, the genetic algorithm is used with the default crossover parameter and a mutation parameter of 20%. The initial population consists of 1000 combinations and the population is evolved for 50 generations. Figure 5.29 shows the optimization metrics for all potential combinations in gray and the metrics of the combinations which lie on the Pareto front in cyan. The Nash bargaining solution is used to identify the best combination of positions, which is shown in purple on the Pareto front plot. Figure 5.30 shows the poses in the optimal combination in the x-y plane. Unlike the solution of the example model in Chapter 4, there are no degenerate solutions because the problem is asymmetrical.

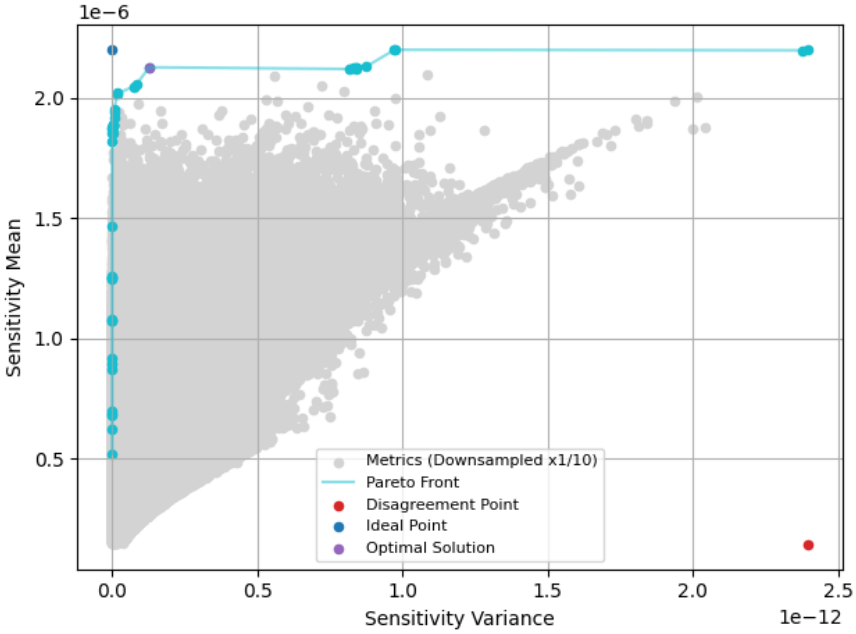


Figure 5.29: Pareto front for example model 5

5.5 Discussion

In this chapter we investigated a number of simplified example models to examine various aspects of the optimization code to see how the optimization performs in scenarios that more closely resemble real world examples. These examples were useful for developing the code, building out features, and ensuring it performs properly before demonstrating it with a real measurement.

We see that attenuation is being properly handled, for both self-shielding and external shields in the measurement scene. Measurement positions which are more shielded are less likely to be selected as the optimal measurement position. For objects with high attenuation coefficients, it may be desirable to place the imager closer to the object. It may also be required to count longer at these positions to get the desired number of counts. We also saw that the feasibility constraint which removes potential positions that cannot see at least one object voxel is sensible because it masks out positions which would have a low sensitivity

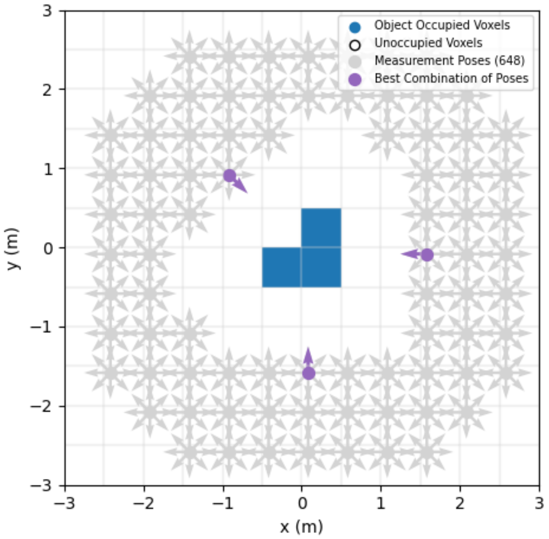


Figure 5.30: Optimal poses selected for example model 5

to the object and not get selected in the optimization. This results in less computations, even in the GA-accelerated code because the sensitivity does not need to be computed for positions which are not considered and the ray tracing does not need to be performed.

We also saw that it is possible to optimize for two separated volumes. It is worth noting that in cases where there are N volumes, it is desired to have at least N+1 optimal positions to properly image the result. This requirement is not included in the code, rather is required user knowledge to correctly formulate the optimization problem.

Overall these models showed the optimization identified measurement poses which make sense to place an imager for a radiation imaging measurement, even in more complicated scenarios than what was explored in Chapter 4.

Chapter 6

Distributed Source Measurement

The toy models previously discussed show the possibility of using this optimization approach, but a field measurement must be taken to study the feasibility of the use of the methods in the field. Many of the nuclear materials accounted for under safeguards agreements are not point sources, but rather large distributed volumes of source, such as DUF_6 cylinders and tanks of bulk materials during fuel recycling. Even sources that can be considered items, such as fuel assemblies, occupy a volume. The methods developed for this project assume that the source occupies some volume that can be described by multiple voxels. Therefore, it is desired to take a field measurement with a distributed source. The next generation mock material holdup loop (NGL) described in Chapter 4 of [55] is used as a distributed source to validate this work.

6.1 Material holdup loop

Figure 6.1 left shows a schematic of the NGL from [55]. Figure 6.1 right shows a camera image of the loop. The loop contains two 5 gallon PVC tanks (referred to as tank 1 and 2) and a large 30 gallon PVC inventory tank. The tanks are connected with 3/4 inch PVC piping with 3/4 inch PVC gate valves to control the flow between tanks. The pipes and tanks are held up by stainless steel support beams. There is also an Iwaki magnetic pump which can be used to circulate water through the pipes and into the different tanks. Liquid source can be injected into the inventory tank and diluted in water for use in experiments.

A series of measurements were taken with this loop to validate the optimization methods developed in this work. For the measurement, the inventory tank was injected with 5.25 mCi of F-18 diluted in 15 gallons of water and thoroughly mixed with the mixing loop. F-18 decays to O-18 through positron emission, where the positron annihilates upon collision with an electron and emits two 511 keV annihilation photons which can be imaged with Compton imaging.

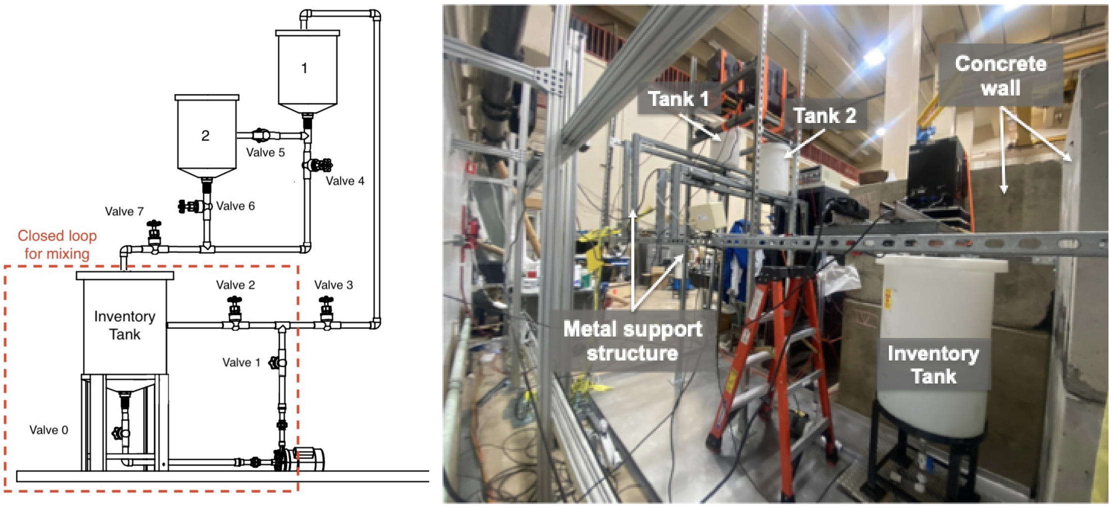


Figure 6.1: Schematic (left) and camera image of loop from [55]

6.2 Optimization procedure

Due to the relatively short half-life of F-18 ($t_{1/2} = 109.77$ minutes), the optimal measurement positions where identified and planned in advance of receipt of the isotope to reduce time between measurements and personnel dose. The point cloud generated by the H420-LAMP contextual sensors is shown in Figure 6.2. While the tanks of the loop are visible, the Velodyne LiDAR on the H420-LAMP does not have a high enough precision to effectively generate a 3D map of the loop which contains the 3/4 inch PVC pipes connecting the tanks.

The Emesent Hovermap has a much better precision of 10 mm, so it was used to generate a 3D map of the loop and surrounding environment several days before the measurement, shown in Figure 6.3. The high level of detail captured by this sensor includes the relatively small pipes between the tanks, as well as clearly defined tank boundaries.

The MMDetection pipeline described in Section 3.2 is used to semantically segment the objects in the 3D point cloud in Figure 6.3. Figure 6.4 shows the results of using the PointNet++ model pre-trained on the ScanNet dataset to semantically segment the point cloud. The floor is correctly identified, as well as the concrete walls. However, many points are given false labels such as 'chair', 'desk', and 'door'. Many of the points are given the label 'otherfurniture'. This is a fitting label, as many of the objects in this scene do not fit any of the labels; however, it is not a useful label for classifying objects. This figure is only included for informational purposes, and is not used to perform the measurement pose optimization.

The measurement scenario investigated consisted of source only in the inventory tank and mixing loop following injection of the source and mixing. The MMDetection pipeline

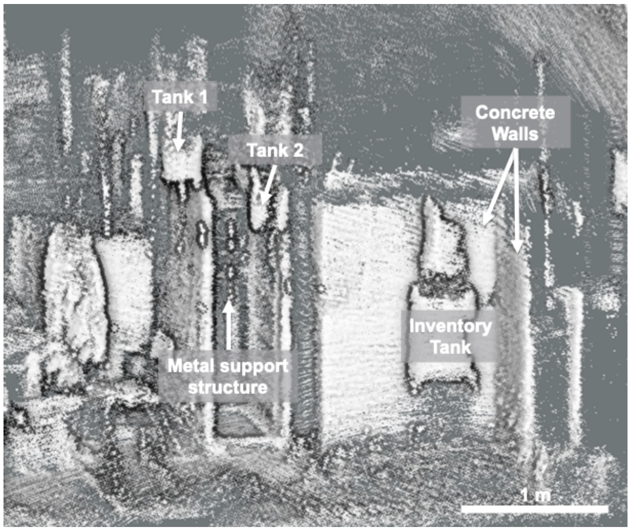


Figure 6.2: 3D point cloud map of mock holdup loop generated with the H420-LAMP

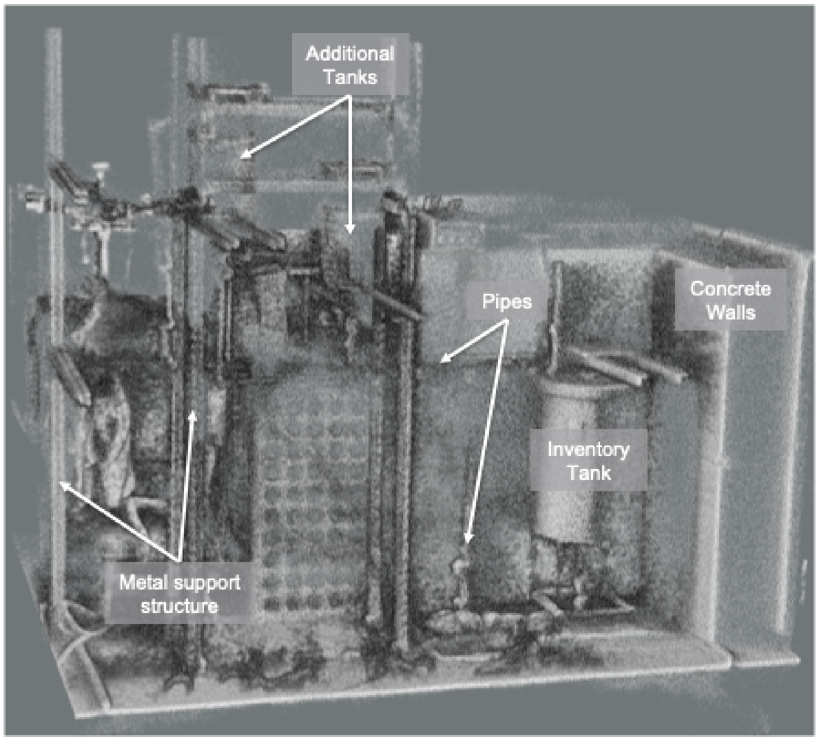


Figure 6.3: 3D point cloud map of mock holdup loop generated with the Emesent Hovermap

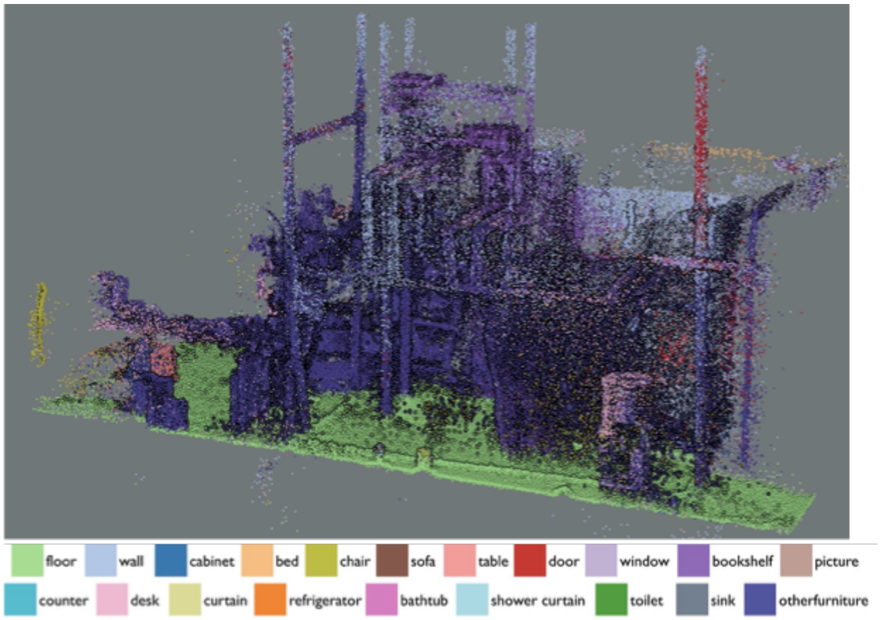


Figure 6.4: Semantic segmentation results of mock material holdup loop

described in Section 3.1 is used to identify the inventory in the 3D point cloud shown in Figure 6.3. The volume selected by the TR3D model pre-trained on SunRGBD data is shown in Figure 6.5. The bounding box returned by the pipeline fully contains the inventory tank. The voxel grid was constructed with 20 cm voxel pitch, resulting in 557,928 voxels, where there are 43 voxels which comprise the inventory tank and 3,768 are non-object occupied voxels. Similar to the example model in Section 5.1, the inventory tank has internal voxels which are not visible at any position in the scene. The inventory tank voxels are given the attenuation coefficient of water, while all other occupied voxels are assumed to be steel, because most of the material in the scene is the stainless steel support structures.

The GA-accelerated optimization is used to identify static measurement poses to quantitatively image the inventory tank. Figure 6.6 shows all of the potential measurement positions identified as feasible in gray. The measurement positions were constrained in the z-axis by the limits of the tripod that the imager is placed on, and by a maximum standoff distance of 3 m. Each of these positions has line of sight to at least one of the inventory tank voxels as determined by the ray-tracing algorithm, and none of them reside in an occupied voxel. There are 8 potential orientations in 45° increments around the z-axis. The complex nature of this scene results in a fairly restricted number of potential measurement poses due to the voxel grid occupancy and ray tracing constraints.

The GA used an initial population of 1000 combinations with the default crossover parameter and a mutation parameter of 20%. Figure 6.6 shows the optimal poses identified by the Nash bargaining solution in orange, with a degenerate solution shown in blue.

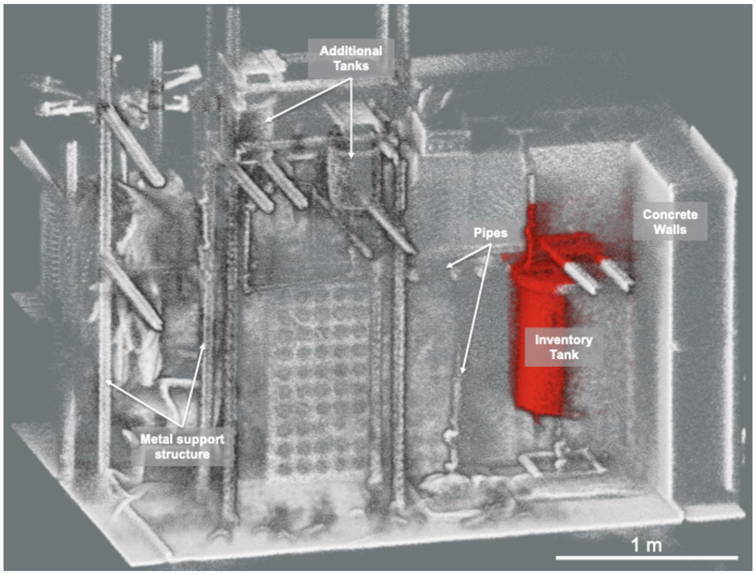


Figure 6.5: Object detection of inventory tank

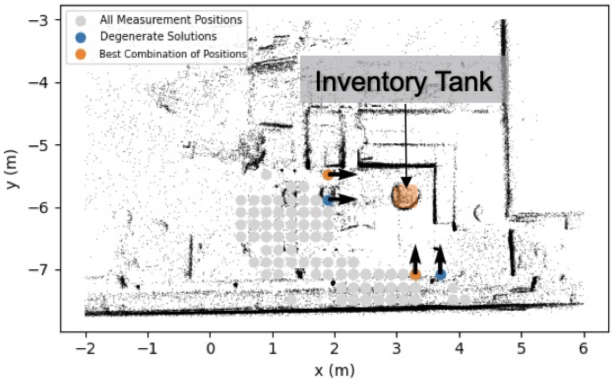


Figure 6.6: Potential measurement positions identified for the inventory tank with the optimal combination of poses identified by the Nash bargaining solution and its degenerate solution identified

The optimal poses are mapped out prior to the measurement by placing tape in the desired tripod foot locations. The measurement poses are verified to be the optimal positions by taking a sample of contextual data at the positions and using ICP (described in Section 2.2.2.1) to plot the actual imager location in the context of the global cloud collected by the Emesent which the optimizations were performed on. If the positions were not within 10 cm or the orientation is not within 10°of the optimal pose, the imager location is adjusted and checked again with ICP.

6.3 Compton imaging results

Data were collected at each of the optimal poses for 10 minutes and 3D Compton imaging is performed. The imaging algorithm includes double-interaction events which deposited a total energy of 511 ± 11 keV in the detector. A lever arm cut of 2.1 mm is imposed, which removes events where interactions are less than 2.1 mm apart. The attenuation through occupied materials which was calculated during the measurement pose optimization is used in the reconstruction algorithm.

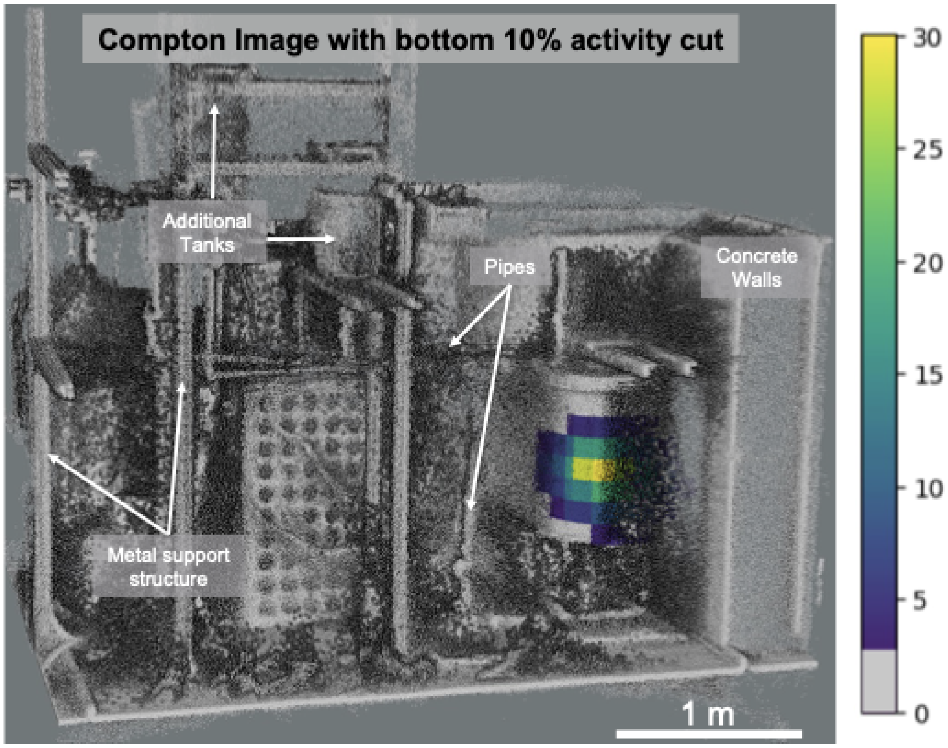


Figure 6.7: Quantitative Compton image of inventory tank with bottom 10% of activity suppressed

The Compton image of the inventory tank and its estimated activity are shown in Fig. 6.7. The bottom 10% of activity is suppressed to improve the visibility of the scene. The source is localized to the position of the water within the inventory tank. The sum of activity in the inventory tank voxels is 756 μCi of F-18. At the time the measurements were taken, the activity in the inventory tank was between 806 μCi and 673 μCi over 20 minutes.

Therefore, the Compton imaging techniques accurately provided a quantification of source activity in the measurement.

6.4 Discussion of measurement results

This measurement showed the feasibility of using the optimization technique to identify static measurement poses for a radiation imaging measurement. The 3D object detection tools described in Section 3.1 were effective in identifying the point cloud points which described the inventory tank in the measurement scene. This information allowed the construction of a custom voxel grid, where object occupied voxels could be given attenuation coefficients of water, and all other occupied voxels could be given attenuation coefficients of steel.

The potential measurement poses identified by the optimization technique were all physically feasible. Due to the amount of support structures and relatively busy nature of the scene, the potential poses are fairly restricted. This saves time in the computation as there are less poses and combinations to be considered.

The 3D Compton image reconstruction correctly localized the emission of the 511 keV annihilation photons to the location of the water in the inventory tank. Additionally the reconstructed source activity is within the range of ground truth during the 20 minute measurement. This result means that the optimization identified measurement poses which were able to accurately quantify the activity of F-18 within a relatively short amount of time.

F-18 is a medical isotope and not considered a safeguard-relevant material. It is used in this analysis because it is relatively cheap and easy to obtain due to its use in medical imaging. It was selected because its 511 keV emission is well within the Compton imaging regime for the H420-LAMP. Nuclear material accounted for under safeguards agreements typically has a much longer half-life than F-18. This means that to quantify safeguards relevant materials within desired levels of uncertainty, longer measurements would need to be performed or the imager would need to be placed closer to the object of interest.

In safeguards applications, the gamma-ray emissions are typically lower than the 511 keV photons in F-18. Some of these gamma rays will still be in the Compton imaging regime of the H420-LAMP (down to around 350 keV), so the optimization techniques are expected to perform similarly for those gamma rays. However, for gamma-ray energies below 350 keV, it would be preferred to use coded-aperture reconstruction techniques. The optimization is built with coded-aperture constraints in mind, but a coded-aperture response function would need to be used to calculate sensitivity for optimization. Measurements with lower energy gamma rays should be performed to ensure the optimization performs as expected in real world measurements for coded-aperture imaging applications, but it is expected that the optimization would perform similarly well as it does for the Compton imaging modality.

Chapter 7

Conclusions

The methods developed in this work demonstrate the improved use of scene data to determine optimal measurement poses for quantitative gamma-ray imaging of volumetric sources. The following sections summarize the methods developed and the measurements performed in the course of this dissertation and discuss opportunities for future work.

7.1 Summary

The identification of optimal measurement poses for quantitative gamma-ray imaging is useful in a variety of applications, including safeguards, where an analyst has limited time to make decisions about how to measure relevant materials. Placing the imager at poses with optimal sensitivity to the volume of interest permits shorter measurement times while still maintaining desired counting statistics, which results in reduced disruption to facility operation. Additionally, the point-and-dwell measurement mode employed in this work allows for reduced dose to personnel while measuring radiation data. This tool presents a way for nuclear safeguards inspections to be carried out more efficiently, allowing safeguards inspectors to continue verifying NPT compliance with the growing number of countries and amount of materials under safeguards agreements.

This work demonstrated the feasibility of using contextual data to identify static measurement poses for quantitative gamma-ray imaging. Computer vision techniques are developed which can identify radiological objects of interest in a 3D point cloud generated in an initial scene survey, even in complex laboratory scenes. A brute-force optimization method is developed and tested on a series of example models to consider all potential measurement poses in a scene, resulting in sensible measurement poses but requiring many computations. A genetic algorithm is used to accelerate this optimization, and bench-marked against the brute-force method, showing that it is a feasible acceleration to the problem that identifies the same optimal poses. These GA-accelerated optimization is then applied to a mock holdup loop with a tank containing F-18 diluted in water. The optimal poses identified by the algorithm are used to measure the tank and generate a quantitative 3D Compton image from a 20 minute measurement. This technique presents a promising method to inform operators on optimal device operation in safeguards field measurements.

7.2 Future Work

This work could be improved by leveraging semantic segmentation techniques to inform attenuation estimates for objects in scenes. The household datasets investigated in this work are insufficient for semantically labeling the laboratory and industrial scenes which would be relevant to safeguards measurements. Computer vision techniques are rapidly advancing, so this is a plausible avenue to pursue in future work. A dataset could be developed with 3D data from laboratory and industrial scenes and used to train the models used in this work, or train more advanced models as 3D computer vision techniques become more sophisticated.

As demonstrated in Chapter 6, high resolution point clouds are necessary for successful use of computer vision techniques. The LiDAR on the H420-LAMP does not have sufficient resolution to identify pipes and small structures in 3D scenes, which will be necessary in safeguards-relevant measurements where bulk materials may be located inside pipes. A higher resolution LiDAR, such as what is available on the Emesent Hovermap, is necessary to correctly identify the object of interest with object detection techniques and to model attenuation of the scene. Contextual sensors continue to improve, which suggest that contextual data collected with state-of-the-art sensors will better capture relevant scene details in the near future. It is also anticipated that next generation datasets will be collected with advanced contextual sensors and analyzed with improved computer vision techniques.

Techniques could also be developed to adaptively determine upcoming measurement poses as information is collected during measurements, rather than computing optimal poses from an initial scene survey alone. This approach could be useful in facilities where generation of a 3D map of the scene is prohibited or a very limited amount of time is available to take measurements.

This work focused on the Compton imaging implementation. Many sources of safeguards relevance are below the 350 keV threshold where Compton scattering is no longer dominant and the number of double interaction events is too low for efficient Compton imaging. In these energy ranges, it would be preferred to use the coded-aperture imaging modality. The optimization techniques developed in this work allow for optimization of coded-aperture imaging poses as well, but these techniques would need to be experimentally validated.

Bibliography

[1] T. Suzuki. "Future challenges in safety, security, and safeguards". In: *Advanced Security and Safeguarding in the Nuclear Power Industry*. Elsevier, 2020, pp. 229–241. ISBN: 978-0-12-818256-7. DOI: `10.1016/B978-0-12-818256-7.00009-X`.

[2] R. Barnowski et al. "Scene Data Fusion - Real-time Standoff Volumetric Gamma-ray Imaging". In: *Nucl. Instrum. Methods A* 800 (2015), pp. 65–69. DOI: `https://doi.org/10.1016/j.nima.2015.08.016`.

[3] K. Vetter et al. "Advances in Nuclear Radiation Sensing: Enabling 3-D Gamma-Ray Vision". In: *Sensors* 19.11 (June 2019), p. 2541. ISSN: 1424-8220. DOI: `10.3390/s19112541`.

[4] K. Knecht et al. "3D Compton Imaging of Distributed Sources around the Chernobyl Nuclear Power Plant". In: *2021 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*. 2021, pp. 1–4. DOI: `10.1109/NSS/MIC44867.2021.9875432`.

[5] J. Hecla, K. Knecht, et al. "Polaris-LAMP: Multi-Modal 3-D Image Reconstruction With a Commercial Gamma-Ray Imager". In: *IEEE TNS* 68.10 (2021), pp. 2539–2549. DOI: `10.1109/TNS.2021.3110162`.

[6] A. Haefner et al. "Handheld Real-time Volumetric 3-D Gamma-ray Imaging". In: *Nucl. Instrum. Methods A* 857 (2017), pp. 42–49. DOI: `https://doi.org/10.1016/j.nima.2016.11.046`.

[7] B.J. Quiter et al. "Ongoing advancement of free-moving radiation imaging and mapping". In: *Hard X-Ray, Gamma-Ray, and Neutron Detector Physics XXIV*. Ed. by Nerine J. Cherepy, Michael Fiederle, and Ralph B. James. Vol. 12241. International Society for Optics and Photonics. SPIE, 2022, 122410H. DOI: `10.1117/12.2635752`.

[8] D. Hellfeld et al. "Free-moving Quantitative Gamma-ray Imaging". In: *Scientific Reports* 11.1 (Oct. 15, 2021), p. 20515. ISSN: 2045-2322. DOI: `10.1038/s41598-021-99588-z`.

[9] C. Wahl et al. "Coded-Aperture Imaging with High-Resolution Large-Volume CZT". In: *2018 IEEE NSS/MIC*. 2018, pp. 1–5. DOI: `10.1109/NSSMIC.2018.8824490`.

[10] R. Pavlovsky et al. "3-D Radiation Mapping in Real-Time with the Localization and Mapping Platform LAMP from Unmanned Aerial Systems and Man-Portable Configurations". In: (2018). arXiv: `1901.05038 [physics.app-ph]`.

[11] W. Hess et al. "Real-Time Loop Closure in 2D LIDAR SLAM". In: *Proc. 2016 IEEE Int. Robot. Automat. Conf.* 2016, pp. 1271–1278. DOI: `10.1109/ICRA.2016.7487258`.

[12] D. Hellfeld et al. "Quantitative Compton Imaging in 3D". In: 63rd INMM Annual Meeting. May 2022. URL: `https://resources.inmm.org/annual-meeting-proceedings/quantitative-compton-imaging-3d`.

[13] H3D Inc. *H420 Gamma-Ray Imaging Spectrometer Fact Sheet.* URL: `https://h3dgamma.com/h100.php`.

[14] K.P. Ziock et al. "Efficiency Calibration of the H3D H420 Gamma-Ray Imager". In: INMM & ESARDA Joint Annual Meeting. May 2023. URL: `https://resources.inmm.org/annual-meeting-proceedings/efficiency-calibration-h3d-h420-gamma-ray-imager`.

[15] M. Salathe et al. "A multi-modal scanning system to digitize CBRNE emergency response scenes". In: *2022 IEEE SSRR.* 2022, pp. 74–79. DOI: `10.1109/SSRR56537.2022.10018826`.

[16] Emesent. *Hovermap ST-X Fact Sheet.* URL: `https://emesent.com/hovermap-series/`.

[17] M. S. Bandstra et al. "Improved Gamma-Ray Point Source Quantification in Three Dimensions by Modeling Attenuation in the Scene". en. In: *IEEE TNS* 68.11 (Nov. 2021), pp. 2637–2646. ISSN: 0018-9499, 1558-1578. DOI: `10.1109/TNS.2021.3113588`.

[18] J. R. Vavrek et al. "Surrogate Distributed Radiological Sources—Part I: Point-Source Array Design Methods". In: *IEEE Transactions on Nuclear Science* 71.2 (2024), pp. 213–223. DOI: `10.1109/TNS.2024.3351597`.

[19] J. R. Vavrek et al. "Surrogate Distributed Radiological Sources—Part II: Aerial Measurement Campaign". In: *IEEE Transactions on Nuclear Science* 71.2 (2024), pp. 224–233. DOI: `10.1109/TNS.2024.3351611`.

[20] Q.-Y. Zhou, J. Park, and V. Koltun. "Open3D: A Modern Library for 3D Data Processing". In: (2018). arXiv: `1801.09847 [cs.CV]`.

[21] G. F. Knoll. *Radiation detection and measurement.* en. 4th ed. Hoboken, N.J: John Wiley, 2010. ISBN: 978-0-470-13148-0.

[22] E. E. Fenimore and T. M. Cannon. "Coded Aperture Imaging with Uniformly Redundant Arrays". In: *Appl. Opt.* 17.3 (1978), pp. 337–347. DOI: `10.1364/AO.17.000337`.

[23] D. Xu et al. "4-pi Compton Imaging with Single 3D Position-Sensitive CdZnTe Detector". en. In: *Proc. SPIE 5540, Hard X-Ray and Gamma-Ray Detector Physics VI.* Denver, CO, Oct. 2004, p. 144. DOI: `10.1117/12.563905`.

[24] C.E. Lehner et al. "4-pi Compton Imaging Using a 3-D Position-Sensitive CdZnTe Detector Via Weighted List-Mode Maximum Likelihood". en. In: *IEEE Trans. on Nucl. Sci.* 51.4 (Aug. 2004), pp. 1618–1624. ISSN: 0018-9499. DOI: `10.1109/TNS.2004.832573`.

[25] M. Galloway et al. “Simulation and Detector Response for the High Efficiency Multimode Imager”. en. In: *Nucl. Instrum. Methods Phys. Res., Sect. A* 652.1 (Oct. 2011), pp. 641–645. ISSN: 01689002. DOI: `10.1016/j.nima.2010.08.101`.

[26] A. C. Zoglauer. “First Light for the Next Generation of Compton and Pair Telescopes”. de. PhD thesis. Max-Planck-Institut für extraterrestrische Physik, 2005, p. 174. URL: `https://mediatum.ub.tum.de/doc/603105/document.pdf`.

[27] A. Martineau et al. “Coded Aperture Optimization Using Monte Carlo Simulations”. In: *Nucl. Instrum. Methods Phys. Res., Sect. A* 616.1 (2010), pp. 75–80. DOI: `10.1016/j.nima.2010.02.261`.

[28] S. Borman. “The Expectation Maximization Algorithm: a Short Tutorial”. In: (2004). URL: `http://www.seanborman.com/publications`.

[29] R. Sundberg. “An Iterative Method for Solution of the Likelihood Equations for Incomplete Data from Exponential Families”. In: *Communication in Statistics-Simulation and Computation* 5.1 (1976), pp. 55–64. DOI: `10.1080/03610917608812007`.

[30] L. A. Shepp and Y. Vardi. “Maximum Likelihood Reconstruction for Emission Tomography”. In: *IEEE Trans. Med. Imag.* 1.2 (1982), pp. 113–122. ISSN: 0278-0062. DOI: `10.1109/TMI.1982.4307558`.

[31] L. Parra and H. H. Barrett. “List-Mode Likelihood: EM Algorithm and Image Quality Estimation Demonstrated on 2-D PET”. In: *IEEE Trans. Med. Imag.* 17.2 (1998), pp. 228–235. DOI: `10.1109/42.700734`.

[32] MMDetection3D Contributors. *MMDetection3D: OpenMMLab next-generation platform for general 3D object detection.* `https://github.com/open-mmlab/mmdetection3d`. 2020.

[33] J. Ansel et al. “PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation”. In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS ’24)*. ACM, Apr. 2024. DOI: `10.1145/3620665.3640366`.

[34] K. Chen et al. *MMDetection: Open MMLab Detection Toolbox and Benchmark.* 2019. arXiv: `1906.07155 [cs.CV]`.

[35] A. Dai et al. “ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes”. In: *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*. 2017. arXiv: `1702.04405 [cs.CV]`.

[36] S. Song, S. P. Lichtenberg, and J. Xiao. “SUN RGB-D: A RGB-D scene understanding benchmark suite”. In: *2015 IEEE CVPR*. Boston, MA, USA: IEEE, June 2015, pp. 567–576. ISBN: 978-1-4673-6964-0. DOI: `10.1109/CVPR.2015.7298655`.

[37] I. Armeni et al. “Joint 2D-3D-Semantic Data for Indoor Scene Understanding”. In: *ArXiv e-prints* (Feb. 2017). arXiv: `1702.01105 [cs.CV]`.

[38] C. R. Qi et al. "Deep Hough Voting for 3D Object Detection in Point Clouds". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019. arXiv: `1904.09664 [cs.CV]`.

[39] Z. Zhang et al. "H3DNet: 3D Object Detection Using Hybrid Geometric Primitives". In: *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII*. Glasgow, United Kingdom: Springer-Verlag, 2020, pp. 311–329. ISBN: 978-3-030-58609-6. DOI: `10.1007/978-3-030-58610-2_19`.

[40] Z. Liu et al. "Group-Free 3D Object Detection via Transformers". In: *Proceedings of the IEEE/CVF ICCV* (2021). DOI: `10.1109/ICCV48922.2021.00294`.

[41] D. Rukhovich, A. Vorontsova, and A. Konushin. "FCAF3D: Fully Convolutional Anchor-Free 3D Object Detection". In: *European conference on computer vision*. 2022. arXiv: `2112.00322 [cs.CV]`.

[42] D. Rukhovich, A. Vorontsova, and A. Konushin. "TR3D: Towards Real-Time Indoor 3D Object Detection". In: *2023 IEEE ICIP*. 2023, pp. 281–285. DOI: `10.1109/ICIP49359.2023.10222644`.

[43] L.-C. Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Vol. 11211. Cham: Springer International Publishing, 2018, pp. 833–851. DOI: `10.1007/978-3-030-01234-2_49`.

[44] B. Zhou et al. "Scene Parsing through ADE20K Dataset". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5122–5130. DOI: `10.1109/CVPR.2017.544`.

[45] B. Zhou et al. *Semantic Understanding of Scenes through the ADE20K Dataset*. 2018. arXiv: `1608.05442 [cs.CV]`.

[46] C. R. Qi et al. "PointNet++ deep hierarchical feature learning on point sets in a metric space". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, pp. 5105–5114. arXiv: `1706.02413 [cs.CV]`.

[47] M. Xu et al. "PAConv: Position Adaptive Convolution with Dynamic Kernel Assembling on Point Clouds". In: *2021 IEEE/CVF CVPR*. 2021, pp. 3172–3181. DOI: `10.1109/CVPR46437.2021.00319`.

[48] J. Amanatides and A. Woo. "A Fast Voxel Traversal Algorithm for Ray Tracing". In: *Proceedings of EuroGraphics* 87 (Aug. 1987).

[49] K. Knecht et al. "Scene-informed optimal measurement positions for quantitative safeguards measurements". In: INMM & ESARDA Joint Annual Meeting. May 2023. URL: `https://resources.inmm.org/annual-meeting-proceedings/scene-informed-optimal-measurement-positions-quantitative-safeguards`.

[50] P. Vanderschraaf. "Learning Bargaining Conventions". In: *Social Philosophy and Policy* 35.1 (2018), pp. 237–263. DOI: `10.1017/S0265052518000110`.

[51] J. Nash. “The Bargaining Problem”. In: *The Econometric Society* 18 (Apr. 1950), pp. 155–162. DOI: `10.2307/1907266`.

[52] E. Kalai and M. Smorodinsky. “Other Solutions to Nash’s Bargaining Problem”. In: *The Econometric Society* 43 (May 1975), pp. 513–518. DOI: `10.2307/1914280`.

[53] F. Biscani and D. Izzo. “A parallel global multiobjective framework for optimization: pagmo”. In: *Journal of Open Source Software* 5.53 (2020), p. 2338. DOI: `10.21105/joss.02338`.

[54] K. Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197. DOI: `10.1109/4235.996017`.

[55] G. Batie. “Methods for Process Monitoring to Accurately Detect and Quantify Material Holdup in Advanced Recycle Facilities”. PhD thesis. UC Berkeley, 2022. 113 pp. URL: `https://escholarship.org/uc/item/3j91s8kn`.

Appendix A

Object Detection

A.1 Object Detection Code

```
from mmdet3d.apis import init_model, inference_detector
import numpy as np

checkpoint_file = 'checkpoint.pth'
config_file = 'config.py'
model = init_model(config_file, checkpoint_file)
result, data = inference_detector(model, 'point_cloud.bin')
boxes_np = result.get(
    'pred_instances_3d').get('bboxes_3d').cpu().numpy()
np.save('file_name',boxes_np)
```

Listing A.1: MMDetection Object Detection Pipeline

A.2 Object Detection Results

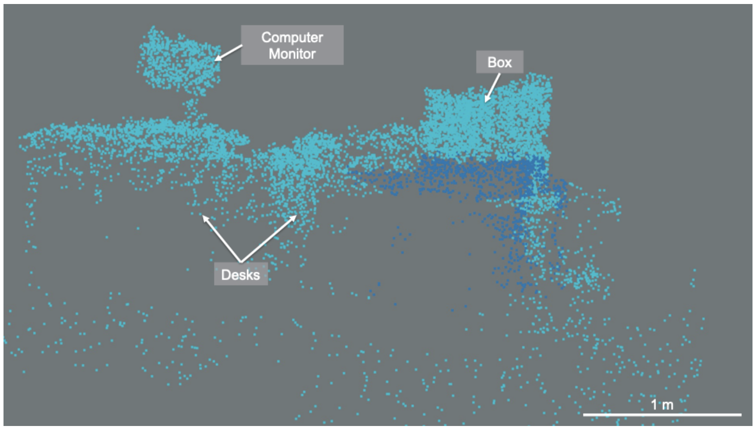


Figure A.1: Object detection by VoteNet model pre-trained on SunRGBD data

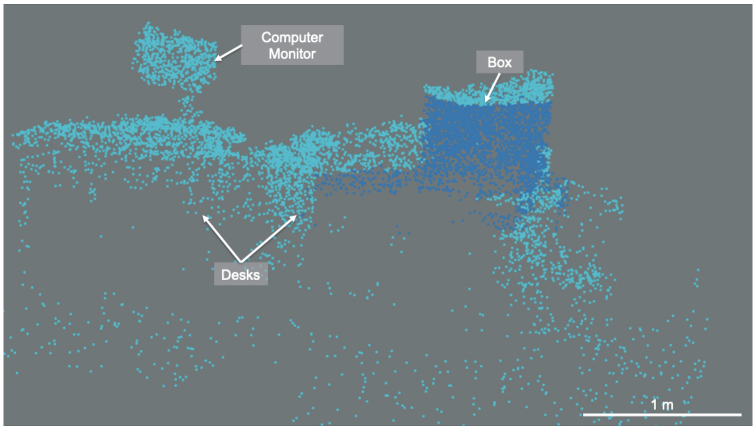


Figure A.2: Object detection by VoteNet model pre-trained on ScanNet data

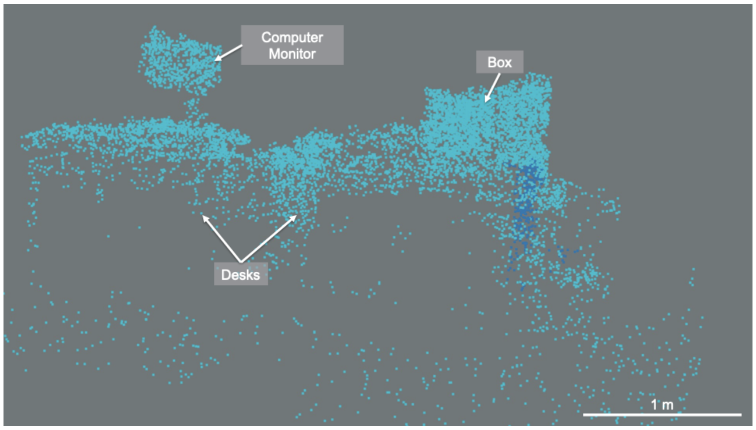


Figure A.3: Object detection by H3D model pre-trained on ScanNet data

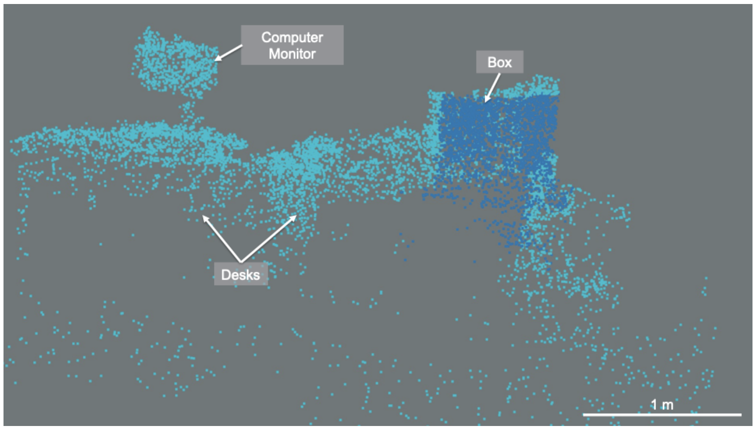


Figure A.4: Object detection by GF3D model pre-trained on ScanNet data with 6 layers and 256 proposals

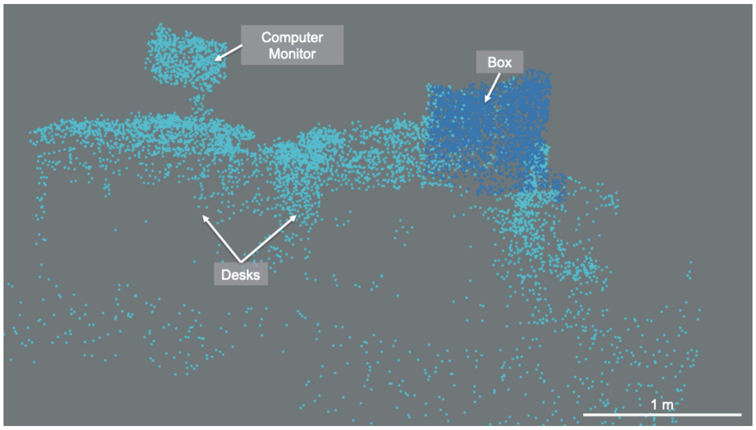


Figure A.5: Object detection by GF3D model pre-trained on ScanNet data with 12 layers and 256 proposals

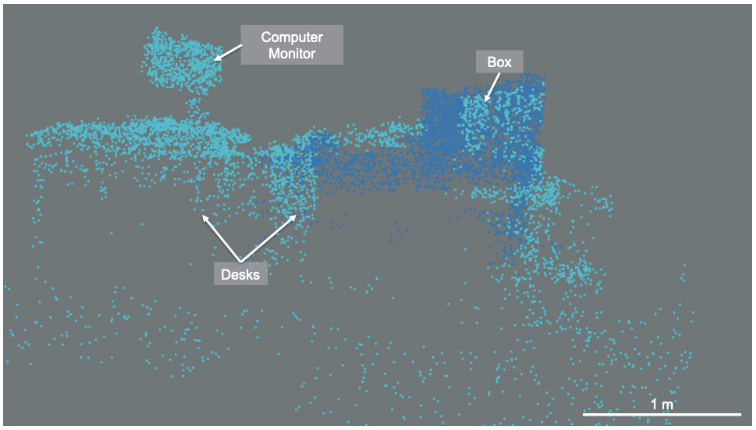


Figure A.6: Object detection by GF3D model pre-trained on ScanNet data with 12 layers and 256 proposals with twice model backbone weig h

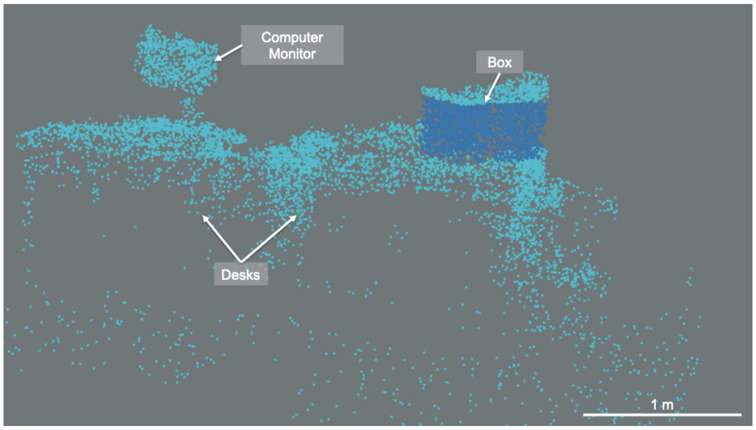


Figure A.7: Object detection by GF3D model pre-trained on ScanNet data with 12 layers and 512 proposals with twice model backbone weig h

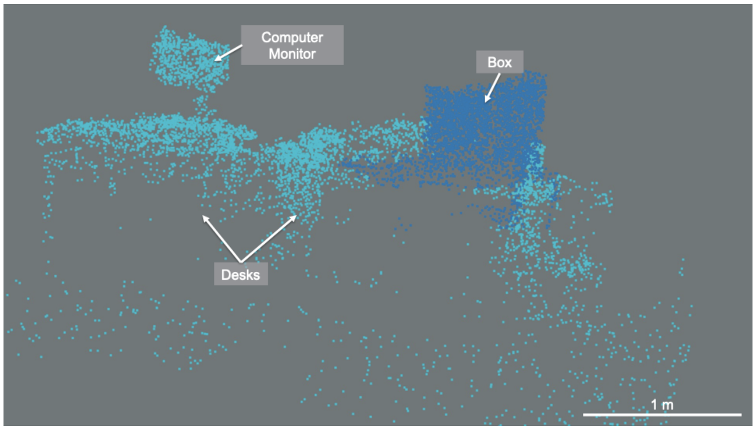


Figure A.8: Object detection by FCAF3D model pre-trained on S3DIS data

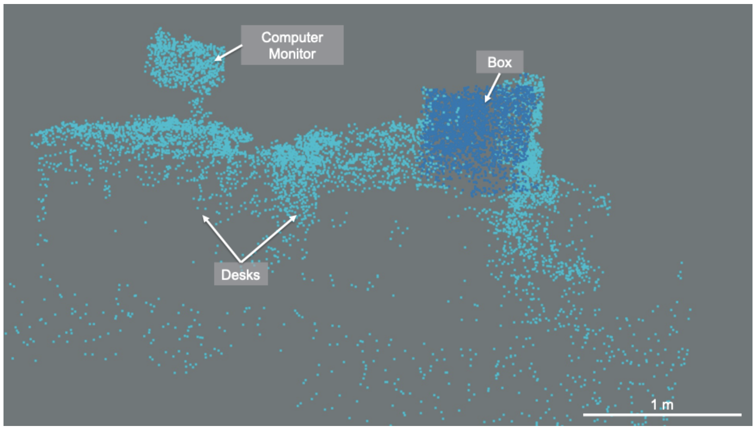


Figure A.9: Object detection by TR3D model pre-trained on SunRGBD data

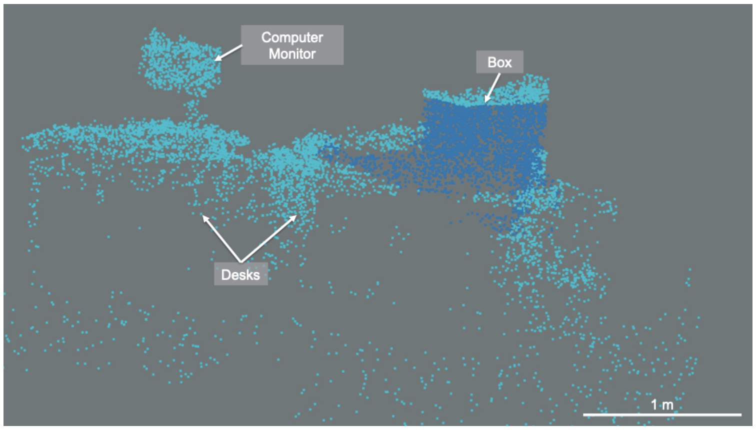


Figure A.10: Object detection by TR3D model pre-trained on ScanNet data

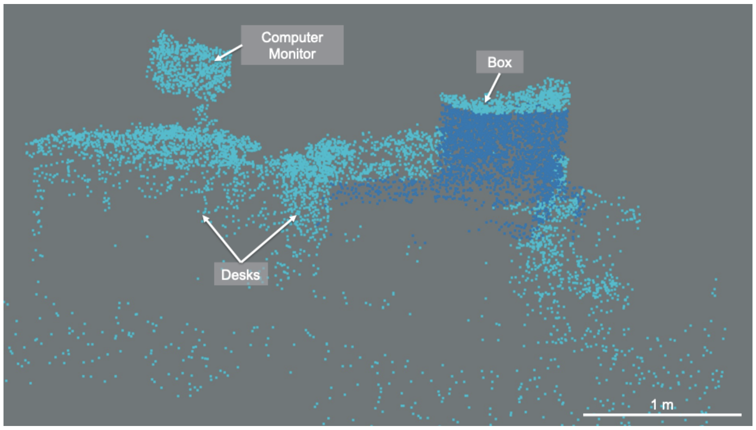


Figure A.11: Object detection by TR3D model pre-trained on S3DIS data

Appendix B

Semantic Segmentation

B.1 Semantic Segmentation Code

```python
from mmdet3d.apis import init_model, inference_segmentor
import numpy as np

checkpoint_file = 'path/to/checkpoint.pth'
config_file = 'path/to/config.py'
model = init_model(config_file, checkpoint_file)
res, data = inference_segmentor(model, 'point_cloud.bin')
mask = res.get('pred_pts_seg').cpu().numpy()['pts_semantic_mask']
logits = res.get('pts_seg_logits').cpu().numpy()['pts_seg_logits']
np.save('mask_file_name',mask)
np.save('logits_file_name',logits)
```

Listing B.1: MMDetection Semantic Segmentation Pipeline

B.2 Semantic Segmentation Results

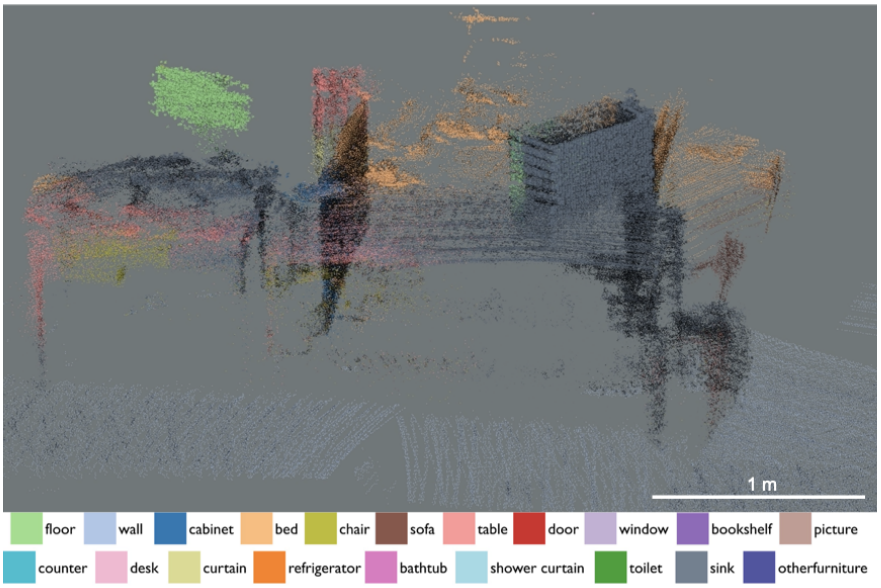


Figure B.1: Semantic segmentation by PointNet++ model pre-trained on ScanNet data with single-scale grouping and x-y-z and color input

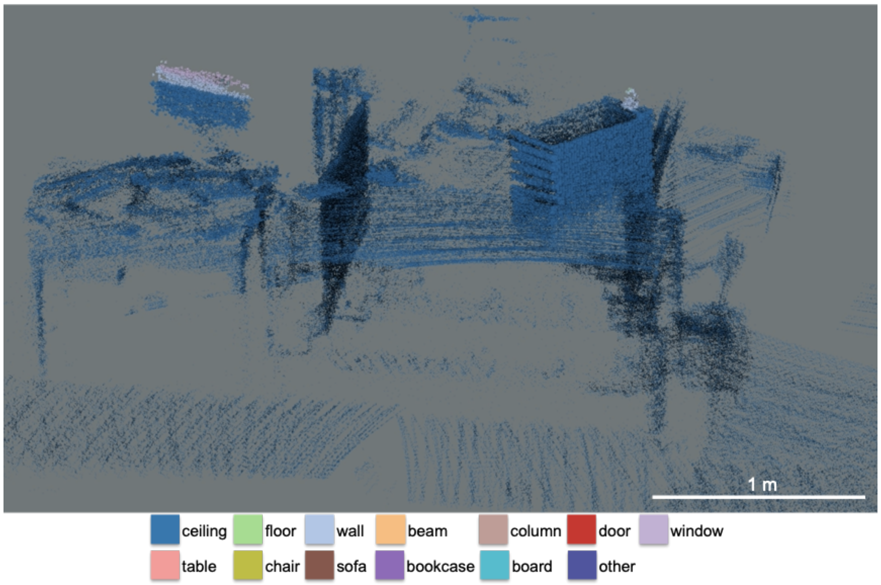


Figure B.2: Semantic segmentation by PointNet++ model pre-trained on S3DIS data with single-scale grouping and x-y-z and color input

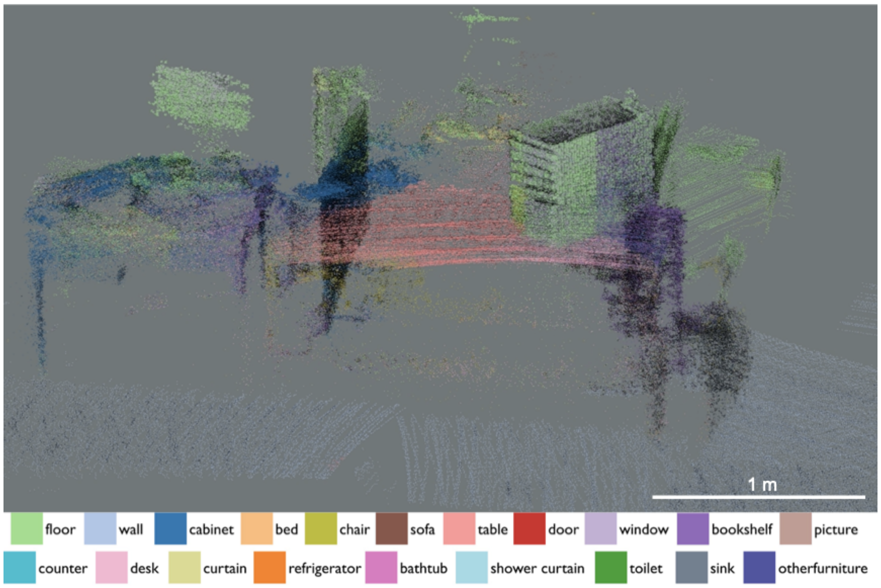


Figure B.3: Semantic segmentation by PointNet++ model pre-trained on ScanNet data with multi-scale grouping and x-y-z input

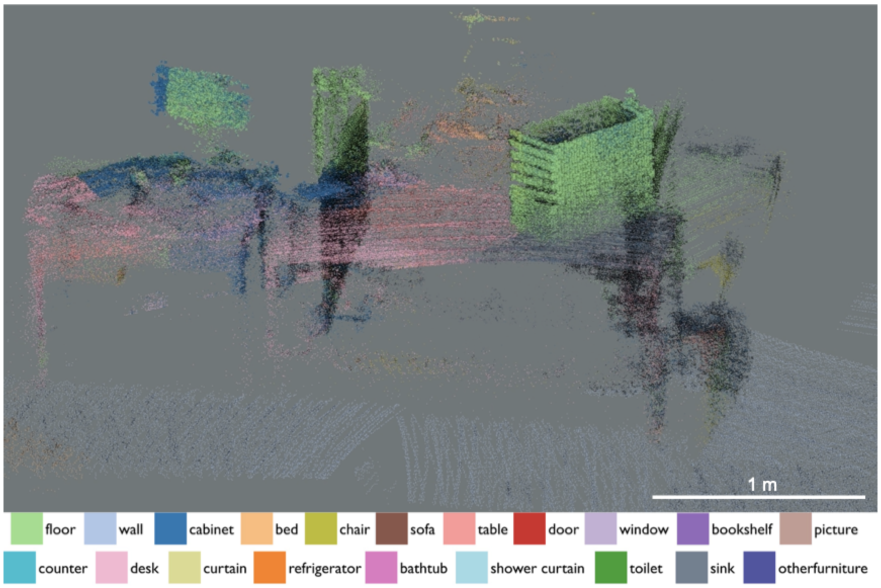


Figure B.4: Semantic segmentation by PointNet++ model pre-trained on ScanNet data with multi-scale grouping and x-y-z and color input

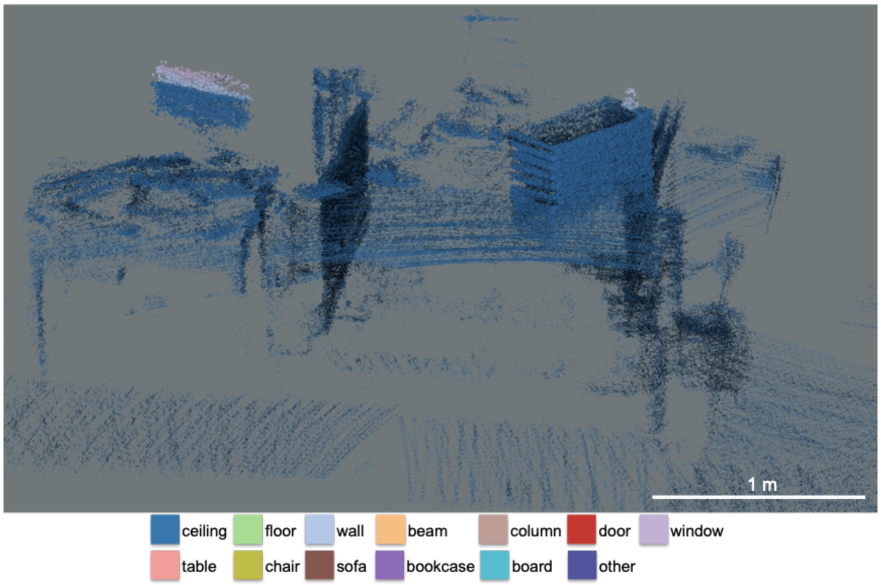


Figure B.5: Semantic segmentation by PointNet++ model pre-trained on S3DIS data with multi-scale grouping and x-y-z and color input

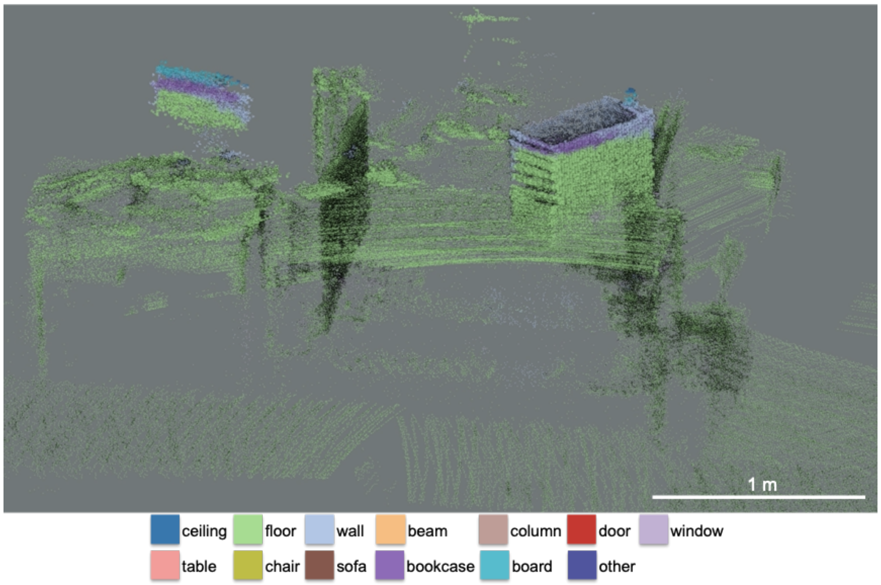


Figure B.6: Semantic segmentation by PAConv model pre-trained on S3DIS data with x-y-z and color input

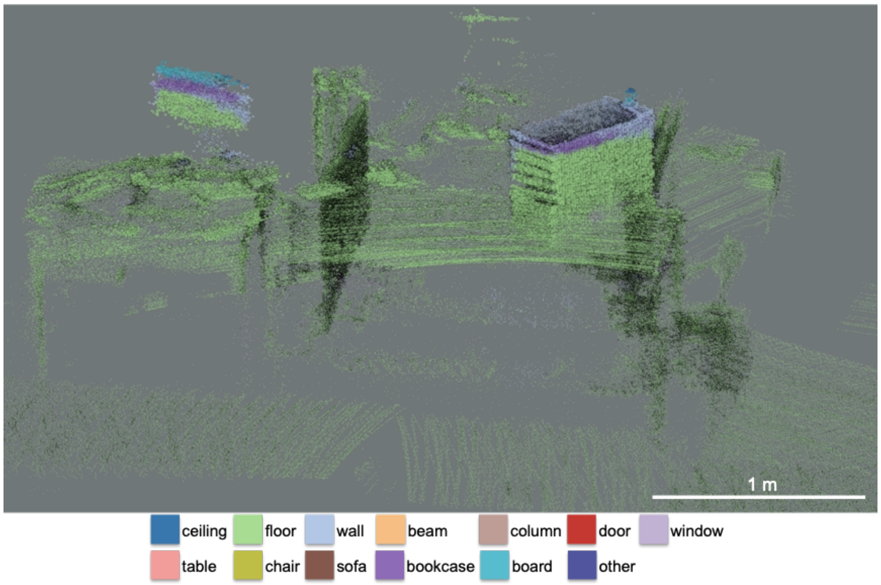


Figure B.7: Semantic segmentation by CUDA implementation of PAConv model pre-trained on S3DIS data with x-y-z and color input

Appendix C

Optimization code

C.1 Pareto front Python code

```python
import numpy as np

def pareto_front(x, y, maximize_x, maximize_y):
    '''
    Find Pareto Frontier of a 2 dimensional optimization

    Parameters
    ----------
    x: np.array
        1st list of optimization variables
    y: np.array
        2nd list of optimization variables
    maximize_x: boolean
        whether to maximize or minimize x
    maximize_y: boolean
        whether to maximize or minimize y

    Returns
    -------
    p_front: np.array
        Pareto front of the two metrics
    '''
    # sort the list in either ascending or descending order of x
    z = np.c_[x, y][np.argsort(x)]
    z = z[:,::-1] if maximize_x else z
```

```
    # start Pareto front with the first value in the sorted list
    p_front = [z[0]]

    # loop through the sorted list
    for xy in z[1:]:
        if maximize_y:
            # find higher values of y and add to Pareto front
            if xy[1] >= p_front[-1][1]:
                p_front.append(xy)
        else:
            # find lower values of y and add to the Pareto front
            if xy[1] <= p_front[-1][1]:
                p_front.append(xy)

    return np.array(p_front)
```

Listing C.1: Pareto front code

C.2 PyGMO Problem Formulation

```
from tools import calc_sensitivity

class meas_pos_optimization:
    def __init__(self,
        N,
        object_xyz,
        object_dxyz,
        meas_pos,
        response,
        E,
        dist_unocc,
        dist_1,
        dist_2):
        '''
        Parameters
        ----------
        N: int
            number of optimal measurement positions
        object_xyz: np.array
            array of object voxel center x-y-z coordinates
        object_dxyz: np.array
```

```
            array of voxel pitches in x, y, and z
        meas_pos: np.array
            array of potential measurement poses
            [x, y, z, roll, pitch, yaw]
        response: np.array
            detector response
        E: num
            gamma ray energy in keV
        dist_unocc: np.array
            unoccupied distance in m - air attenuation
        dist_1: np.array
            occupied distance of object material in m
        dist_2: np.array
            occupied distance of non-object material in m

        Returns
        -------
        meas_pos_optimization: pygmo.Problem
            pygmo Problem object
        '''
        # save parameters to class for use in computations
        self.object_xyz = object_xyz
        self.object_dxyz = object_dxyz
        self.meas_pos = meas_pos
        self.N = N
        self.E = E
        self.response = response
        self.dist_1 = dist_1
        self.dist_2 = dist_2
        self.dist_unocc = dist_unocc

    def fitness(self,x):
        # x is index of measurment position
        x = x.astype('int')
        meas_pos = self.meas_pos[x]
        d0 = self.dist_unocc[x]
        d1 = self.dist_1[x]
        d2 = self.dist_2[x]

        # calculate sensitivity at each pose to each voxel
        sensitivity = calc_sensitivity(meas_pos,
                            self.object_xyz,
```

```
                              d0,
                              d1,
                              d2,
                              self.object_dxyz,
                              response=self.response,
                              E=self.E)

        # minimize sensitivity variance
        if self.N == 1:
            f1 = sensitivity.var()
        else:
            f1 = sensitivity.sum(axis=0).var()

        # maximize sensitivity mean
        if self.N == 1:
            f2 = -sensitivity.mean()
        else:
            f2 = -sensitivity.sum(axis=0).mean()

        return [f1,f2]

    def get_nobj(self):
        # return number of objectives
        return 2

    def get_nix(self):
        # number of interger dimensions
        return self.N

    def get_bounds(self):
        # pre computed measurement positions for indexing meas_pos
        min_bounds = int(0)
        max_bounds = int(len(self.meas_pos)-1)

        return ([min_bounds] * self.N, [max_bounds] * self.N)
```

Listing C.2: PyGMO Problem Formulation