The Conceptualization and Parameterization of a Gaseous Detector Rasterizing Pinhole Gamma Camera

by

Terry J. Price

A thesis submitted in partial fulfillment

with the requirements for the degree of

Master of Applied Science in Nuclear Engineering

in

The Faculty of Energy Systems and Nuclear Science

University of Ontario Institute of Technology

August 2016

Abstract

This thesis details the conceptualization and parameterization of a gaseous detector rasterizing pinhole gamma camera. In this thesis, there is a literature review that describes the historical development of gamma imaging, a technical background that aims to give the reader the prerequisite background knowledge, a methodology, and, a result and discussion chapter. The thesis includes studies that determine if the concept of a gaseous detector rasterizing pinhole gamma camera is feasible, mathematical modeling that allowed for the exploration of the idea, software development that automated the mathematical modeling, parametric studies that explored the performance of various sets of design parameters, and, finally an iterative engineering design process that converged at a final set of design parameters. Ultimately, a set of design parameters, from which a prototype may be constructed, were developed.

Acknowledgements

The author would like to thank the following individuals:

- Alexander Miller who inspired this work.
- Dr. Rachid Machrafi for his patient pedagogy and patience.
- My Father Terry A. Price for pushing me to achieve and never give up.
- My Mother Susan Price for her unconditional love and support.

Contents

Table of Figures

Nomenclature

SYMBOLS

a	The elipse semi-major axis scaling factor
A_i, B_i, w	i^{th} coeffecient of the Fourier series
b	The ellipse semi-minor axis scaling factor
B-angle	The angle between a perpindicular to cylinders axis, and a ray intersecting the cylinder
c	Speed of Light, m s^{-1}
C(x,y)	Colour at Position, (x,y)
C_{max}	Maximum colour value
d	Distance through a material, cm or pinhole diameter, cm
d_i	Distnce through the i^{th} material, cm
$\mathbf{g}$	A point that intersects a plane
g	distance between shield and detector plane
h	Plank's Constant, m^2 kg s^{-1}
$\hat{\boldsymbol{\imath}},\hat{\boldsymbol{\jmath}},\hat{\boldsymbol{k}}$	The first, second and third unit vectors in 3-space. <1,0,0>, <0,1,0>, and <0,0,1>,respectively.
i	Index i
I	Photon intensity at a measured position, s^{-1}
I(x,y)	Photon Intensity at position (x,y) , s^{-1}
I_0	Initial photon intensity at source, s^{-1}
I_{max}	Maximum photon Intensity, s^{-1}

l	The mean of a poisson process; mean chord length, cm
M_e	Electron Mass, kg
$\mathbf{n}$	A plane normal
$\mathbf{p}$	A point in 3-space
$p(t)$	The value of the Poisson process at random value t
q	The random variable of a Poisson process
r	Distance from source to target, cm
R^2	Pearson's correlation coeffecient
S	Source strength at a position, s^{-1} cm^{-2}
s	Surface area, cm^2
S_0	Initial Source Strength, s^{-1}
$\left(\frac{S}{S_0}\right)_{x,y}$	The relative source strength at position (x,y)
$\mathbf{t}$	A position on a line; shield thickness, cm
$\mathbf{v}$	The direction vector of a line
v	Volume of a solid, cm^3
$\frac{V}{V_0}$	The geometric attenuation factor
x	Detector plane dimension
x,y	The position of the x or y coordinate in 2 dimensional space
$x_i, y_{i,}, z_i$	The i^{th} incident of the a set of coordinates in three space
Δ	Change in a quantity
$g(\zeta)$	The quantity of background noise that has not been scaled by an effieceny factor

ζ	A random real number between 0 and 1
θ	Scattering Angle
λ	The wavelength of light, m, or an arbitrary scalar
μ	Attenuation coefficient, cm^{-1}
π	Ratio of a circles circumference to its diameter -3.141592....
ρ	Material density, g cm^{-3}
σ	The standard deviation of a Poisson process $\sigma = \sqrt{l}$
ϵ	The detector effeciency
ϕ	Particle flux, $s^{-1}cm^{-2}$
μ_i	Attenuation coeffecient of the ith material, cm^{-1}
$\frac{\mu}{\rho}$	Density normalized attenuation constant, $cm^2\,g^{-1}$

Acronyms

PMT	Photomultiplier Tube
NaI	Thallium Doped Sodium Iodide – Sometimes denoted NaI:Tl
GM	Geiger-Muller
YAP	Cerium Doped Yttrium Aluminum Perovskite. Sometimes denoted YAP:Ce
CsI	Cesium Iodide

Introduction

The gamma camera is a device that can be used to visualize the spatial distribution of radioactive substances in an environment. The gamma camera produces images of radiation fields. It has many uses in various fields such as medical imaging, archaeology, and nuclear security. Its use is widespread. However, gamma cameras are expensive pieces of laboratory equipment that cost many tens of thousands of dollars [1].

Two elements of the gamma camera that contribute to its price are its Photomultiplier Tube (PMT) and its Sodium Iodide (NaI) scintillating (light producing) crystal. Newport, a scientific equipment supplier, lists the cost of its photomultiplier tubes more than 1,500 US Dollars [2]. Gamma Spectacular, a radiation spectroscopy equipment supplier, lists its 1.5" x 1.5" sodium iodide crystal for 900 US Dollars [3]. These are costly components that each gamma camera requires multiple units of.

There exists an alternative, simple and low-cost, method of detecting ionizing radiation: the use of a Geiger-Muller Tube. The tube is a gas filled detector that produces an electrical signal when radiation interacts with it. This thesis investigates the question as to if a Geiger-Muller tube could be used instead of a scintillating crystal. If it is possible to create a gamma camera that costs hundreds, rather than tens of thousands of dollars, this will represent an improvement in the field of gamma imaging technologies. Even if the quality of the image produced is significantly less than that of existing gamma cameras, the fact that it costs significantly less would enable new applications to be explored that were previously far too expensive. Furthermore, such a decrease in cost would allow scientists in fields that are not well funded to have access to the technology. In summary, the problem statement that prompted this thesis is:

"The technology deployed in current gamma cameras is inherently complex. Can an alternative system be developed that reduces the cost and complexity of imaging a radiation field?"

To this question, this research proposes the "Rasterizing Pinhole Gamma Camera" - a device that uses a moving Geiger-Muller detector in place of a scintillating crystal to image the radiation field. In this way, the operation can be simplified, and the cost can be considerably lowered.

It is the aim of this thesis to:

1. **Explore the idea of creating a rasterizing pinhole gamma camera.**
2. **Determine the feasibility of the rasterizing pinhole gamma camera.**
3. **Determine what the design parameters (e.g. pinhole diameter) of the rasterizing pinhole gamma camera should be.**

The thesis contains four parts outlined as follows:

Background: This chapter presents relevant information required to understand the rest of the thesis. Sections include a conceptual description that provides a brief description of the concept; a literature survey, in which a brief history of the gamma camera is detailed, and the development of the field is reviewed; and finally, a theoretical background is presented that describes some key mathematical and physical concepts that are drawn-upon in the development of the system.

Methodology: The methodology chapter presents relevant information as to how the conceptualization and parameterization of the system were achieved. The methodology chapter has four sections: preliminary work, mathematics, software development, and background measurement.

Results and Discussion: The results and discussion section presents the results produced during this research. The structure of this chapter mirrors that of chapter two: first, the results from the preliminary work are shown, followed by the mathematical results, then the software results, background measurements, and finally, the results of the engineering design process are presented.

Conclusion: The final section concludes by summarizing the work done in the thesis, followed by a commentary on the aims of the thesis and closed by commenting on future work.

Chapter 1: Background

1. Technical Background

This section provides some technical background pertinent to understanding the rest of the thesis.

1.1 Fundamentals of Radiation

Matter it comprised of atoms. Atoms, for the purpose of this overview, are comprised of three constituent parts: protons, neutrons, and electrons. The electrons orbit near the periphery of the atom, while the protons and neutrons remain in the dense inner core, known as the nucleus.

There are two forces at play inside the nucleus: the electric and the strong-nuclear force. Particles that have a *positive or negative* charge interact through the electric force: Particles that have the same charge repel each other, and attract in the converse. The strong-nuclear force obeys a more complicated relationship in which nucleons attract each other up to a certain distance, after which the force becomes repulsive.

Protons interact through both the strong nuclear force and the electric force, whereas neutrons interact only via the strong nuclear force. The interplay of these two forces with the energy of the protons and neutrons can create nuclear configurations that are not stable in their configuration. It is said that the nucleus, in an unstable configuration, has an excess of *energy*.

One way to visualize this is to imagine protons and neutrons as solid billiard balls. If looked at each individually, the motion of the protons and neutrons are chaotic and unpredictable. They are tugged to and from by the force-at-a-distance influence of the strong nuclear and electric forces. If, however, they are viewed from a distance, they seem to orbit a central point in space.

As the velocity of the protons and neutrons is increased, the diameter of the orbits are likewise increased, and the entire system now has an excess of energy. Nuclei that have an excess of energy are said to be in an excited state; conversely, nuclei that do not have an excess of energy are said to be in a ground-state. Nuclei can lose their excited energy and return to their ground state by the emanation of a particle.

There are three dominant modes in which these particles are emitted: alpha, beta, and gamma decay. There are some other modes of decay such as cluster decay, but these are rare, enough to be ignored for the purpose of this thesis. In alpha decay, a cluster of two protons and two neutrons is emitted from the nucleus. This cluster is known as an alpha particle. In Beta decay, a neutron turns into a proton, and the nucleus emits an electron. The emitted electron is known as a beta ray. In gamma decay, the nucleus returns to the ground state by the emission of a packet of light, known as a photon. Photons can be produced by other processes in nature, but the photons produced in the nuclear de-excitation process are particularly high in energy and referred to as gamma photons or gamma rays. This gamma radiation has an energy equal to the difference in energy of the nucleus when it is stable to when it is excited. The gamma ray has such a high energy, which it is capable of removing an electron, or ionizing, from another atom that it interacts with. Atoms that undergo such an emanation (radiation) are said to be radioactive.

1.2 The Electron Volt

An electron volt is a unit of energy equal to $1.6 * 10^{-19}$ Joules. It is the energy picked up by a single electron as it is accelerated, from rest, across a potential difference of 1 volt. Some radionuclides emit radiation with energies on the order of kilo electron volts (keV).

1.3. The Interaction of Gamma Radiation with Matter

Once emanated, gamma photons may interact with either, other nuclei or orbital electrons. The predominate methods by which they do this are called the photo-electric effect, the Compton Effect, and pair production. There are some other, more exotic, ways in which photons can interact; however, they are immaterial to the discussion at-hand.

The photoelectric effect is a phenomenon that was pioneered, in part, by A. Einstein in 1905 [4]. In the photoelectric effect, a photon is absorbed by an orbital electron and imparts energy to the electron. This imparted energy ejects the orbital electron. This phenomenon explains the experimental result in which an electric current is detected when a sample of metal is exposed to a source of ultra-violet light. Interestingly, there is a threshold in light energy before which this effect does not occur. This threshold is explained by the so-called *electron binding energy* in which the orbital electron is held in its orbit, and the energy of this *potential-well* must be exceeded before the electron can be ejected.

In the Compton effect, a photon once again interacts with an electron; however, this time, instead of being absorbed it bounces off (is deflected off) an electron that recoils and is ejected. With respect to the photon, one might say that the photon loses some of its energy to the electron upon its deflection. With respect to the electron, one might say that energy is imparted, by the photon, into the electron, which then recoils and is ejected from the atom's orbit. The relationship between energy lost by and recoil angle behaves in a well-established manner that can be described with an equation derived with classical mechanics:

$$\Delta\lambda = \frac{h}{m_e c}(1 - \cos(\theta)) \tag{1}$$

Where, '$\Delta\lambda$' is the change in photon wavelength in m; 'h' is Plank's constant in m^2 kg s^{-1}; 'm_e' is the mass of an electron in, kg; 'c' is the speed of light in m s^{-2}, and θ is the scattering angle.

Finally, in the phenomenon of pair production, two 511 keV photons are produced, traveling in opposite directions, by the annihilation of a positron and an electron. With respect to the decay process, when an atom undergoes the so-called *beta-plus* decay process and transmutes a proton into a neutron, charge, and energy in the universe are conserved by the emission of a positron (the anti-particle of an electron). This positron travels through the electron cloud of the matter in which it is in and loses its energy (velocity) in the process through Columbic interaction. When the positron finally comes to rest, it *annihilates* with an electron and releases energy equal to twice the rest-mass of an electron. This energy manifests itself in the form of two 511 keV photons that are traveling in opposite directions.

1.4. Beer's Law

Beer's law is a first-order ordinary differential equation of the form:

$$\frac{dI}{dx} = -\mu x, \tag{2}$$

and it describes the attenuation of a beam of photons (including gamma rays) of Intensity '*I*' in a material of thickness '*x*' [5]. The attenuation constant, μ, represents the rate at which the beam of photons is attenuated. The differential form of Beer's law is solved by the equation:

$$I = I_0 \exp(-\mu x), \tag{3}$$

where I is the intensity of the photon beam at a distance '*x*' through a material with an attenuation coefficient of 'μ'. The quantity '*I₀*' is the unattenuated beam intensity. Beer's Law is visualized in the diagram below.

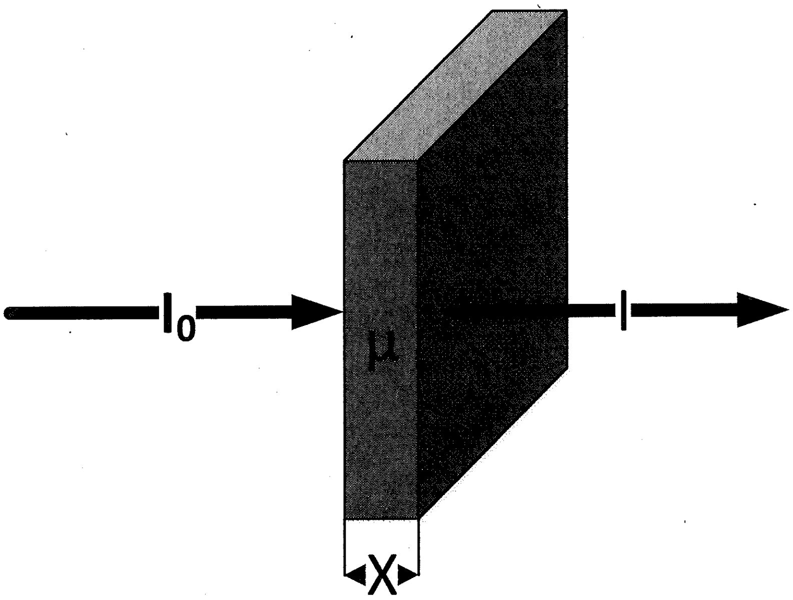


Figure 1: Beers Law Visualization

Occasionally, the attenuation coefficient in Beer's law in divided (normalized) by the density of the material. This is often expressed as the symbol $\frac{\mu}{\rho}$, and its unit is cm^{2} g^{-1}. This allows the calculation of the attenuation coefficient for materials with different densities.

1.5. Units for Radioactivity

Radioactive substances undergo a certain number of transmutations every second. This is referred to as the number of decays per second at which the substance transmutes. This is often represented in the unit of a Becquerel [Bq], which is equivalent to one decay per second. Alternatively, radioactivity can also be represented in Curries [Ci]. One Currie is equal to 3.7*10^{10} decays per second.

1.6. Radiation Detectors – Mechanisms of Detection

Radiation with energy that is sufficient to remove an orbital electron from an atom (ionize it) is capable of being detected with (ionizing) radiation detectors. There are two predominate methods of real-time radiation detection: ionization detectors and scintillation detectors.

In ionization detection, a radioactive particle interacts with a substance, such as a gas, and removes orbital electrons from it. These electrons are collected and amplified to create a signal that is then detected. The detector on which this work is based, the Geiger-Muller detector, uses this mechanism to detect radiation.

This is in contrast to a scintillation detector in which a radioactive particle creates light in the *scintillating material.* It is this light that is then collected, converted into an electrical signal, amplified, and detected. A common scintillation detector is the Sodium Iodide (NaI) detector, which features a crystal of near pure sodium iodide that is sealed to a light amplification device, known as a photomultiplier tube.

1.7. The Geiger-Muller Tube

The Geiger-Muller tube is a device that was invented by Hans Geiger and Walther Muller in 1908 [6]. It is a gaseous ionization-based detector that consists of a central anode (positively charged) wire, coaxially enclosed in a closed clerical cathode (negatively charged). A fill-gas is injected into the space between the cathode and anode.

An incoming gamma photon interacts with the cathode shell of the detector and ejects an electron via either the photo-electric effect or the Compton Effect. This electron is ejected into the fill gas. An electric field is established in the fill gas. The electron is accelerated towards the anode due to the influence of the electric field. The electron, during its transversal of the space between the anode and cathode, collides with other atoms in the gas and imparts energy to them. This energy is then released in the ejection of an orbital electron. The electrons created in this process go on to create their own path to the anode; during which, *secondary* electrons are created. This process by which many ejected electrons are created from an initial electron is called electron multiplication. If the electron multiplication process saturates the fill-gas volume and liberates all the electrons, it is called a *Townsend avalanche.* It is the pulse created by this avalanche that is detected by the amplifying and detection circuitry interfaced behind a Geiger-Muller tube.

1.8. Radiation Localization and Imaging Technology

There are many situations in which it is desirable to know the direction from which gamma photons are being emitted. For example, in the inspection of a package that is suspected to contain substances with excited nuclei, one might want to know more precisely where in the package the gamma photons are emanating from. Alternatively, a drug may be laced with a radioactive atom, and its uptake, in the body, monitored. Several devices can be used to do this, and are described in the subsequent sections.

1.8.1. The Rectilinear Scanner

The predecessor to the modern gamma camera is a device known as the *Rectilinear Scanner.* According to [7], the rectilinear scanner was developed by Ben Cassen in 1956. It, as its name implies, consists of a scintillating radiation detector placed on a mechanical linkage. This mechanical linkage transverses the area to be imaged.

When radiation is detected, a light-bulb is activated on the other end of the mechanical linkage. This light-bulb is stored in a light-proof box with a piece of photosensitive paper in it. In this way, whenever the light-bulb is activated, the reading is recorded on the photosensitive paper.

1.8.2. The Gamma Camera

A *gamma camera* is a device that produces an image of a radiation field. According to [8], gamma cameras have two major components: a collimator and a localization system. The collimator is an array of channels cut into an attenuator that ensures the gamma photons entering the camera are parallel. The localization system determines the position of any gamma photon detected and sends that position to processing electronics that reconstruct an image of the gamma field.

1.8.3. The Rasterizing Pinhole Gamma Camera

The *rasterizing pinhole gamma camera* has two major components: the shield and the detector plane. The shield is essentially a large lead slab with a pinhole in it that blocks out all radiation that is not passing through the pinhole. The detector plane consists of an XY table with a detector mounted on it. The XY table moves (rasterizes) the detector across a plane parallel to the plane of the shield. In this way, a reading can be made on any point on the detector plane. Computer software then performs post-processing on the recorded data to reconstruct the image.

1.9 Software Description

1.9.1. The C-Sharp Programing Language

The C-Sharp .Net (C#.Net) programing language is a modern object oriented programing language developed by Microsoft [9]. A feature that was particularly useful for the purposes of this project was the Language Integrated Query (LINQ). LINQ allows developers to query data structures using a standard set of operations.

1.9.2. POV-Ray

Visualization of the problems generated in the code was achieved by the use of the 3D rendering software, POV-Ray (Persistence of Vision Ray tracer). POV-Ray is a freeware software package that takes commands via a C-like language and produces three-dimensional images of the specified geometry. For more information, see [10].

1.9.3. MCNP

Monte Carlo N-Particle (MCNP) is a stochastic radiation transport code produced by Los Alamos National Labs. MCNP has existed for many decades and has become a de-facto standard for radiation transport calculations. MCNP was used in this thesis to calculate the proportion of radiation transported from a source to a target. More information can be found in [11].

1.9.4. ImageJ

ImageJ is an open-source image analysis package that has built-in tools that allow for the measurement of qualitative image properties. Of particular use to this project was the *magic-wand* tool that allowed an area of similarly coloured pixels to be selected in a region-of-interest. Furthermore, ImageJ has the functionality to provide region-of-interest properties such as area, or centroid location. Finally, ImageJ was also used to compile some animated movies that were used in the presentation of this thesis. More information can be found in [12].

2. LITERATURE REVIEW

2.1. Gamma Camera Construction

It is reported in [13] that the current gamma camera (not the one that this research is based on) consists of six major components: The collimator, the crystal array, the photomultiplier tube array, the position logic circuits, the pulse height analyzer, and the gating circuitry.

The collimator shapes the incoming radiation field so that when it impinges on the radiation detector inside the camera, it does so in a manner that is dependent on the position of the incident radiation source. The collimator, in effect, ensures a gamma ray coming in from one direction, does not strike the radiation detector in the same way that a gamma ray from another direction does. The collimator is placed in front of the camera.

The crystal array is an array of crystals that emit a pulse of light when radiation interacts with it. A type of crystal that is used in some gamma cameras is the Sodium Iodide (NaI) crystal. The crystal array is placed behind the collimator and in front of the photomultiplier tube array. An optical guide is sometimes placed between the crystal array and the photomultiplier tubes to guide light produced by radiation in the crystals into the photomultiplier tubes.

The position logic circuits take the signal generated by the photomultiplier tube array and convert it into an *X* and *Y-signal*. These two respective signals correspond to the horizontal and vertical position of the detected radiation-induced pulse in the photomultiplier tubes.

The pulse height analyzer analyzes the output from the photomultiplier tubes and produces the so-called *Z-Signal.* The later is an electrical signal that represents the magnitude of the energy of the incident gamma photon.

Finally, the gating circuitry is a circuit that either accepts or rejects an incoming signal depending on the magnitude of its Z-Signal (incident energy). A *region-of-interest* may be set with the gating circuitry that only accepts signals within a particular range of incident energies. This functionality allows the gamma camera to not only reject spurious signals and background noise but also examines only radionuclides of interest.

2.2. Collimation Methods

A collimator is a device that shapes the incoming radiation field for a gamma camera. The incoming radiation is shaped by the collimator so that the position at which it impinges on the radiation detection apparatus changes with respect to the direction in which the radiation originates. This process of radiation field shaping is known as Collimation.

As reported in [14], the most common type of collimator is the parallel hole collimator – a device that uses multiple parallel holes to collimate the radiation;

however, there are many different types. These types include the converging hole collimator, in which the holes point *outward* at a central focus; the diverging hole collimator in which the hole point *inward* towards a central focus; or, the pinhole type which this work is based upon. The type of collimation that a gamma camera uses depends on its intended application.

2.3. Scintillation Crystals

A scintillation crystal is a device that produces light (scintillation) when radiation interacts with it. The original gamma camera developed by Hal Anger used a Thallium doped Sodium Iodide (NaI) crystal to detect radiation. [15]. Sodium Iodide is a hygroscopic scintillation crystal with a density of 3.67 g cm^{-3}. Sodium Iodide must be enclosed in a hermetic seal to avoid degradation due to its hygroscopy. Since Anger's original camera, a number of other crystals have been used to perform radiation detection in gamma camera systems.

As reported in [16], a gamma camera with a 0.7mm resolution was developed using Cerium Doped Yttrium Aluminum Perovskite (YAP:Ce or $YAlO_3$:Ce). YAP:Ce yields 62-64% fewer photons per MeV of energy absorbed (photon yield) compared to NaI. The light response of YAP:Ce is about 9 times faster than that of Sodium Iodide (25 ns). This crystal, unlike sodium iodide, is not hygroscopic. Due to its non-hygroscopic property, YAP crystals do not need to be hermetically sealed; this allows arrays of YAP crystals to be constructed with a much finer pitch than that is attainable using NaI.

Cesium Iodide is another scintillation material that has been used in Gamma Cameras [17]. Cesium Iodide is a slightly hygroscopic scintillation material with a density of 4.51 g cm^{-3} [18]. Cesium has a higher atomic number (55) than sodium. This higher atomic number, combined with its higher density, increases the probability that a gamma ray passing through the crystal will interact with it. Cesium iodide scintillation crystals can either be sodium (CsI:Na) or Thallium doped (CsI:Tl). The thallium doped variety offers a photon yield that is 45% of the photon yield of NaI. The Sodium doped variety offers a photon yield that is 54% of the NaI photon yield.

2.4. Photomultiplier Tubes

A photomultiplier tube is a device that produces an electrical signal when a photon is incident upon its detection area – its photocathode. A brief overview of photomultipliers and their use in gamma camera is presented in this work. The reader is referred to [19] and [20] for a complete description of the photomultiplier tube.

A photomultiplier tube consists of a *photocathode*, focusing electrodes, a series of *dynodes*, and an anode. The photocathode is a sheet of alkali metal that is kept at a high negative voltage. This voltage creates a surplus of electrons on the surface of the photocathode. Since there is a surplus of electrons near the surface of the cathode, there is a higher probability that some of the electrons have an energy just below the potential well of the alkali metal. When a photon impinges upon the surface of the photocathode, it imparts energy into one of the electrons on the surface of the photocathode. This imparted energy is often high enough to eject the photon from the potential well of the photocathode, and launch the electron into the space behind the photocathode.

The focusing electrodes are a set of electrodes positioned behind the photocathode that focus electrons emitted from the dynode array. The dynode array is a series of metal plates, each kept at a successively higher voltage.

Since the dynode is kept at a higher voltage than the photocathode, the electron is accelerated along its path from the photocathode to the dynode. When the electron strikes the dynode, it does so with sufficient energy to release additional electrons from the dynode. These additional electrons are then accelerated by the next dynode towards it. When the additional electrons strike the next stage of the dynode array, they create further electrons. In this way, at each successive stage of the dynode array, the electrons are *multiplied*.

The entire apparatus is housed within an air-tight tube that is evacuated. This evacuation allows electrons to transverse the length of the photomultiplier tube unimpeded.

As reported in [21], the original Anger Gamma camera used an array of photomultiplier tubes arranged in a hexagonal close-packed structure. The photomultiplier tubes are coupled to the scintillation crystal via a fused-quartz optical guide. When an incident radiation particle induces a scintillation in the scintillator crystal, it can be picked up by multiple photomultiplier tubes in the array. By comparing the relative outputs of several adjacent photomultiplier tubes, the location of the scintillation can be found. The photomultiplier tube with the largest output will be the closest to the scintillation site.

This multiplication process continues for a number of stages until a sufficient multiplication of the initial electron is reached. The last stage of the dynode array serves as an *anode* kept at a positive voltage. The multiplied electrons are attracted to this surface and produce a voltage spike at the anode when they strike it.

Another type of photomultiplier tube used in gamma camera construction is the position sensitive photomultiplier tube (PSPMT) [22]. The position sensitive photomultiplier tube provides separate horizontal and vertical position signals that have magnitudes determined by the position on which photocathode on which the incident photon impinges. It accomplishes this by the use of multiple anodes and at multiple positions. The relative number of electrons collected at each anode determines the position of the location signal produced. A photon that impinges on the left side of the photocathode will produce more electrons on an anode placed on the left side than it would produce on an anode placed on the right side. Conversely, a photon impinging on the right side of the photocathode will preferentially produce electrons at the right side the tube.

2.5. Historical Problems

It is reported in [23] that before the modern gamma camera was invented; investigators had used Geiger-Muller detectors for localized activity measurements in the thyroids of patients. In these previous systems, detectors were physically moved over the patient to determine the location of the radioactivity; however, no image was produced. It is reported that Geiger-Muller detectors proved to have a serious limitation given their less than 1% detection efficiency. Furthermore, it reported that the motivation for switching to sodium iodide-based detectors from Geiger-Muller tubes is that sodium iodide is much more efficient at detecting gamma radiation.

2.6. The Rectilinear Scanner

The predecessor to the modern gamma camera was a device known as the "Rectilinear Scanner", which was developed by Ben Cassen in 1950 [7]. The device itself was constructed from a single scintillating crystal that was encased in shielding in such a way that only beams parallel to the axis of the detector could enter it. This detector was moved manually over a grid that had been drawn on a transparent piece of Lucite (clear plastic). The image of the radiation field was then manually constructed by use of a hand-held drawing implement (such as a pen or marker).

This device was soon improved upon by the development of the rectilinear *photoscanner*. The photo-scanner featured the same shielded detector arrangement, but this time, the arrangement was on a motorized mechanical linkage. To the other end of this linkage was mounted a light bulb, that was enclosed in a light-proof box. When the crystal of the rectilinear photoscanner detected radiation, the light bulb was momentarily turned-on. Below the bulb, inside the light-proof box, was placed a sheet of photosensitive paper that darkened when exposed to light. In this way, whenever radiation was detected, its presence was recorded on the photo-paper. The position on the photo-paper was representative of the position of the scanning head when it detected the radiation. The result was that an image of the radiation field was constructed. A complete history of the rectilinear scanner and the source of much of the information in this section can be found in [7].

2.7. The Anger Camera

In 1958, the predecessor to what would become the modern gamma camera was created by Hal Anger. The device was known as the *Anger Camera*, and it featured a radio-scintillating sodium iodide crystal positioned behind a pinhole. A number of light-pipes were attached to this crystal that fed into photomultiplier tubes. These photomultiplier tubes detected the scintillations produced by radiation at various positions in the crystal. Analogue electronics then processed the outputs of the photomultiplier tubes and produced a visual representation of the field on an oscilloscope display.

Although there is a considerable difference in the image produced by a modern gamma camera, and the Anger Camera of 1958, the methodology of using multiple photomultiplier tubes and processing the signal to produce an image remains the same. The reader is referred to [24] for an excellent history of the invention of the Anger Camera.

Another type of photomultiplier deployed in the design of Gamma Cameras is the Silicon Drift Detector (SDD) [25]. As reported in [26], Silicon Drift Detectors are similar to P-Type, Intrinsic, and N-Type (PIN) Photodiodes in that they are PN junctions that produce a current when illuminated. However, unlike photodiodes, they employ multiple electrodes that allow the charge carriers produced by the impinging photons to be collected at different positions along the device. By comparing the relative quantities of charge collected at the different anode locations, one may determine the position that the photon was incident.

2.8. The Autofluoroscope and Multi-Detector Cameras

It is reported in [27] that before the development of the large area scintillation crystals used in modern gamma-cameras, a device known as the Autofluoroscope was developed by scientists Bender and Blau in 1963. This device made use of multiple banks of sodium iodide crystals to detect an image of the radiation field.

This development allowed for gamma cameras with much larger detection areas than that which was made possible by single crystals. This was also one of the first devices to make use of digital techniques for data processing; in fact, [28] reports that "magnetic core memories" and "magnetic tape" were used in the *Digital Autofluoroscope.*

2.9. Complete Digitization

It appears that by the mid-1970's, the gamma camera was nearly entirely digital. It is reported in [29] that a multi-detector system was developed, and it had a computer system featuring a 128x128 pixel 16-bit colour display. This system used software written in Assembler and could store its output on floppy disks.

2.10. SPECT Camera

One key avenue in the development of the field of radiation imaging was the advent of Single Photon Emission Computed Tomography (SPECT). In this field, a three-dimensional model of the volume being imaged is reconstructed using two-dimensional *slices* of data from radiation images. It is reported in [30] that the first SPECT system was described in 1963. The SPECT camera represents a departure from the development of the *two-dimensional* gamma camera that this research is focused on.

2.11. High Purity Germanium Camera

As reported in [31], another type of gamma camera is the High Purity Germanium (HPGe) Camera. In this type of camera, strips of P-type semiconductor sandwich a block of high-purity N-type germanium – the top and bottom are P-type semiconductor whereas the middle is an N-type semiconductor.

A typical model of how the detection process works in an HPGe detectors was described in many sources such as [19]. The layers of the semiconductor are joined, a negative voltage is applied to the P-type regions, and a positive voltage is applied to the positive region – a reverse bias in diode terminology. This reverse bias creates a *space-charge* region between the two different types of semiconductors; that is, a region that is sparse in its availability of free charge carriers (holes and electrons). When A gamma-ray interacts with the semiconductor in the vicinity of this space-charge region is creates free charge carriers that experience a force produced by the applied voltage. These charge carriers produce a current in the contacts that connect the semiconductor to a copper wire.

The P-type layers are grooved in regularly spaced intervals such that each grooved segment is separate from the next. The direction of the grooving is orthogonal, and each segment of grooved material is connected to the successive one via a resistor – that is, each grooved segment has a known resistance, and one can determine the number of grooves between two points by measuring the resistance between them. The along each plane of grooved tracks can be determined by measuring the charge exiting from the left connecting node and the right connecting node of the grove and comparing one of those measurements to the total charge exiting both nodes. Each plane (top and bottom) provides one dimension of positional determination; by measuring the position in both planes, one is able to center in on the position of the radiation interaction.

2.12. GEM Cameras

A recent trend in gamma camera development is the use of Gas Electron Multiplier (GEM) foils in gamma cameras. The paper, written in 2011, details the development of a gamma camera system using four GEM plates and a pinhole collimator to produce an image of a gamma field [32]. The signal processing is performed using Application Specific Integrated Circuits (ASIC) for signal digitization, and Field Programmable Gate Arrays (FPGA) for digital signal processing. Interestingly, [32] makes note of the expense of existing Sodium Iodide gamma cameras.

3. Conceptual Description

The rasterizing pinhole gamma camera itself consists of two major components: the detector plane, in which a pinhole is drilled, and a shield. The system is visualized below.

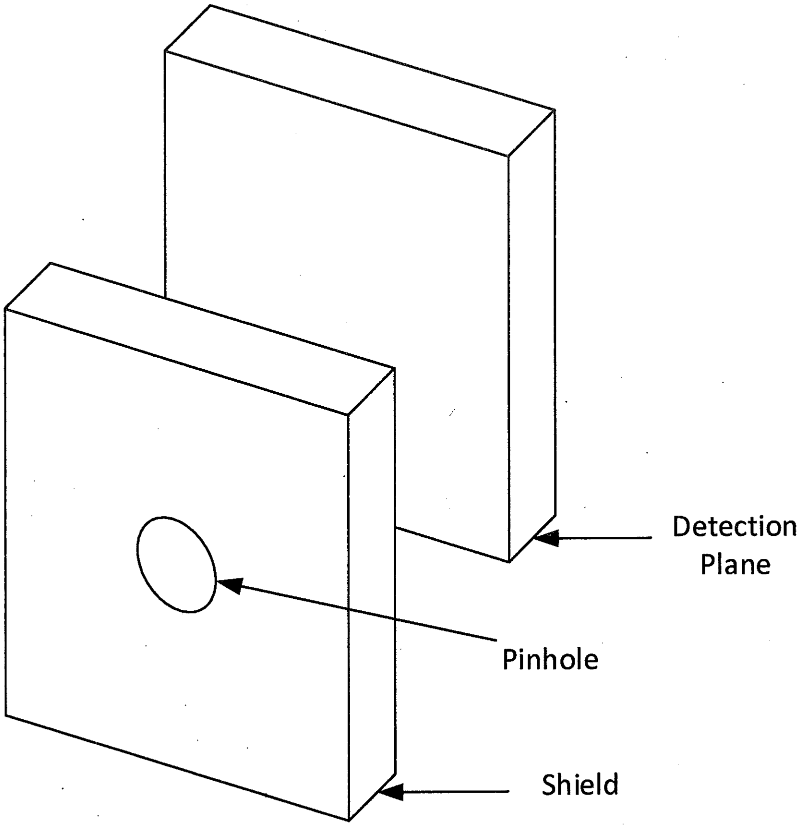


Figure 2 Diagram of Rasterizing Pinhole Gamma Camera System.

The working principle, in brief, is that the system is designed such that the shield differentially attenuates incoming gamma-rays from a radiation source. The rays that pass through the pinhole are projected onto the detection plane where they are detected by a Geiger-Muller detector.

The shield is a slab of material that attenuates the radiation and has a pinhole cut into it. The pinhole is cut all the way through the shield and allows radiation to pass from one side of the shield to the other. The detection plane is an abstract construct that represents all the possible positions in space of the Geiger-Muller detector. The

Geiger-Muller detector is moved by actuators in such a way that it may be positioned at any point in the detection plane. Conceptually, the image produced is the accumulation of all the data collected at all the positions data collected by the detector on the detection plane.

To construct an image, measurements are taken by the Geiger-Muller detector at various points on the detector plane. These points are then processed by a computer software package that performs post-processing and then reconstructs the data into an image.

The image reconstruction is done by first converting each of the data collected at a position (x, y) on the detector plane to a colour. The (x,y) pair of numbers indicates the horizontal and vertical position on the detection plane. This colour is determined by dividing the reading at the position (x,y) by the maximum possible reading, and scaling the brightest colour representable by that factor:

$$C(x,y) = \frac{I(x,y)}{I_{max}} * C_{max}, \tag{4}$$

where C(x,y) is the color at position (x,y); I(x,y) is the reading in [s^{-1}] at position (x,y); I_{max} is the maximum reading at all position; and, C_{max} is the maximum colour representable.

Each reading is then mapped to a bitmap image and a position that corresponds to the position at which the reading was made. This process is illustrated below.

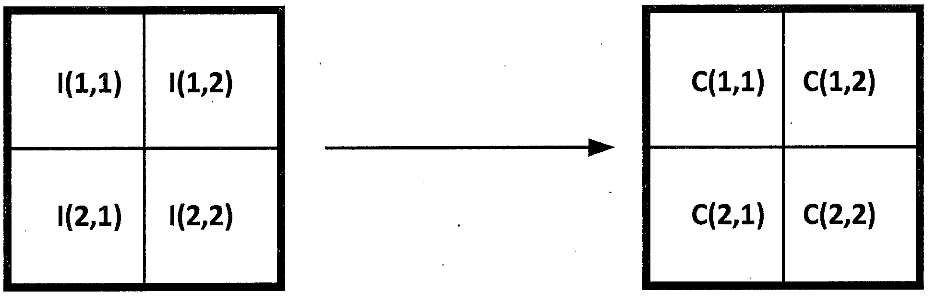


Figure 3: Diagram of Mapping Process

This approach is different from the traditional approach to gamma cameras in that the way in which the image data is collected is a serial rather than a parallel process. This offers several advantages and disadvantages to the traditional gamma camera as summarized in the table below.

Table 1: Comparison with Existing Camera Technology

	Traditional Gamma Camera	**Rasterizing Pinhole Gamma Camera**
Cost	More than $100,00 USD [33]	Expected to be much lower
Parts required to detect radiation	Multiple photomultiplier tubes and scintillation crystals	A single gaseous radiation detector
Radiation Field Measurement Method	Measures multiple points in parallel	Measures a single point in series– takes longer to collect same amount of data
Spectroscopic Capability	Capable of measuring the energy of incoming radiation	Not capable of measuring the energy of incoming radiation

Chapter 2: Methodology

1. Methodology Overview

This section provides an overview of the methodology. Details as to the specifics of the methodology are found in the succeeding sections. The overall approach in this research is as follows: First, preliminary calculations were performed to determine if ideas were feasible or not. If the calculations had given results that were impractical for reification, then the project would have ended there. If, for example, the system would weigh several tons, then for all intent and purpose, the system would be unrealistic. After the initial feasibility calculations had been completed, a first simulation was performed using the readily available MCNP code. The simulation was run on a consumer-grade desktop computer, and it took more than a month to complete. Since the only computing resources available were several lab computers, it was inferred from this that large-scale parametric studies would not be possible within the two-year time frame of this project given the resources available. The precise timing of how long the simulation took and the precise specifications of the computer used are not available. However, the overall time was more than a month, and the computer used was a mid-2010's consumer grade computer.

The second part of the project consisted of figuring out how to perform the calculations for the parameterization in a time-frame that would fit within two years. This consisted of an extensive study of the field of geometry, and linear algebra followed by performing derivations for the relevant equations. It was decided that software would be written to solve the equations and perform the parametric studies.

The software, in addition the equations that needed to be solved, required another input: the background noise quantity. This quantity would be used to simulate the output of the system. The background count-rate was measured experimentally with a detector that is analogous to the proposed system. The background count-rate measurement was performed by placing the Geiger-Muller detector on a desk and measuring the count rate without any radiation sources present. More details as to how the background radiation measurements were performed are detailed in the section titled "Background Measurements" .

The fourth part of the methodology involves writing software to solve the derived equations. This involved writing consecutively more full-featured implementations of the software until a version of the software was written that could perform the required calculations. This software was then validated in a number of test cases.

Finally, after the software was written, and validated, the camera could finally be parameterized. This was done by taking a best-guess – that is, a guess that is best-given experience with what a gamma camera should perform - as to what the design parameters should be, then simulating hundreds of system variations with design parameters centered about the initial best guess. Each variation had its performance measured with respect to analysis criteria, and each measurement was compared to the performance of the initial best-guess. From there, the system that scored highest in its analysis criteria was chosen and the process repeated until the gain by each addition iteration was marginal. In this way, a final parameterization of the system was *converged* upon.

2. Preliminary Work

Feasibility calculations were performed to provide assurance that the concept would perform if further work went into its development. These calculations included a signal attenuation calculation, a scan-time calculation, a shielding weight calculation, and an initial simulation using available software.

The first of the initial feasibility calculations were done by considering a hypothetical test-case, and performing analytic calculations that determined if the idea was feasible.

First, a measure of the signal attenuation was needed. This was done by solving Beer's law at a distance '*r*' away from a radiation source:

$$\frac{S}{S_0} = \frac{\exp(-\frac{\mu}{\rho}\rho d)}{4\pi r^2}, \tag{5}$$

where '*S*' is the source strength at the distance '*r*' away; '*S₀*' is the initial source strength; $'\frac{\mu}{\rho}'$ is the density-normalized linear attenuation constnat, $'\rho'$ is the density of the attenuating material; and 'd' is the distance traveled in the attenuating material. This calculation is performed for a source 100 cm away from a 500 keV gamma source in dry air.

The next feasibility calculation performed was a determination if the scan could be completed in a reasonable period of time. This was done by supposing a theoretical 100 measurement experiment wherein each measurement was performed for ten seconds. Feasibility was determined by finding the maximum number of counts that would be received in that ten second period of time.

Finally, an estimate as to how much shielding would be required and how much the shielding would weigh was performed. This was done by assuming that the shielding would attenuate half the incoming radiation of a Cobalt-60 radiation source. Cobalt-60 is a radioactive substance that emits a 1.17 and 1.33 MeV gamma ray [34].

A simulation was developed in MCNP that simulated a right-circular cylindrical detector transversing a detection plane in a rectilinear manner. A pinhole was placed in the center of a slab of lead material, and a detector transversal plane was placed further behind the slab. A photon source was placed on the lower right-hand side of the detector.

Software was written to vary the input of MCNP so that the position of the detector was changed in each run by a predetermined step size. After the position was varied, and the input file written, the simulation was run, and the results were stored. After all the simulations at the various positions, the results were processed into an image. For this particular simulation, the detector varied over an area of 64x64 positions (0.04 Megapixels in modern consumer camera terminology).

In total, these efforts represent an attempt to show that the concept was feasible and determine if future work ought to be performed.

3. Mathematics

Before developing the software, a number of mathematical equations has been formulated to include the following:

1. Beer's law for multiple materials
2. The intersection of a ray with a plane [35]
3. The intersection of a ray with a cylinder
4. The calculation of the minimum detection plane size

For the method of ray-cylinder intersection, the cylinder was taken as an equation of a line with a certain radius, and a ray as a point with a direction. The method used in this work is described in [35], and it was verified with visualization using the MuPad Package of Mathworks Matlab.

4. Background Noise Measurement

An experiment was performed in order to find a reasonable estimate of the background noise. In this experiment, a Lind-Inc. (20) 712 "End-Window Alpha, Beta, Gamma" (the product name) Geiger-Muller tube was placed horizontally on a desk. The tube was operated at 550V, and was connected to a signal processing circuit – a GCK-01 from Imagesco (21). Ten sixty-second readings were taken, and the number of counts during each period was recorded.

The average background count rate was determined from these measurements, along with the standard deviation of the measurements. These two parameters provide sufficient information to fit a Poisson distribution to the data.

5. Software Development

5.1. Overview

The code itself calculates the solution to Beers law:

$$\frac{I}{I_0} = \exp(-\mu d), \tag{6}$$

Where *'I'* is the photon flux at a distance*'d'* through an attenuator; *'Io'* is the unattenuated flux; and 'μ' is the linear attenuation coeffecient.

The software then accounts for geometric attenuation by factoring in the inverse square law:

$$\frac{S}{S_0} = \frac{1}{4\pi r^2}\frac{I}{I_0} = \frac{\exp(-\mu d)}{4\pi r^2}. \tag{7}$$

The *'r'* term is the total distance between the source and target. The ratio '$\frac{S}{S_0}$' is the fraction of the original source strength that is measured at the detector position. This resultant equation thereby is giving the photon intensity per unit area (photon flux) over an arbitrary number of cells bounded by planes and cylinders with arbitrarily defined materials.

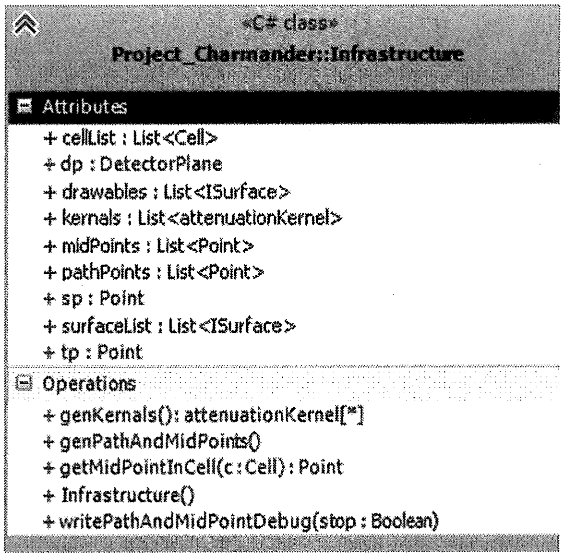
An overview of the code's algorithm is mentioned in the next sections, followed by an in-depth discussion of each step.

5.2. Algorithm

1. Define the surfaces involved in the problem.
2. Define the cells involved in the problem and store them in a data structure called, the Cell List.
3. Define the set of source and target points.
4. For each combination of source and target points, calculate the distance the ray travels through the cell, and find out the total amount of ray attenuation.
5. Compile the results into a data structure.
6. Compile the results into an image.

5.3. Code Core Structure

The methodology of the code relies on the Infrastructure class as shown below.



```
«C# class»
Project_Charmander::Infrastructure
Attributes
    + cellList : List<Cell>
    + dp : DetectorPlane
    + drawables : List<ISurface>
    + kernals : List<attenuationKernel>
    + midPoints : List<Point>
    + pathPoints : List<Point>
    + sp : Point
    + surfaceList : List<ISurface>
    + tp : Point
Operations
    + genKernals(): attenuationKernel[*]
    + genPathAndMidPoints()
    + getMidPointInCell(c : Cell) : Point
    + Infrastructure()
    + writePathAndMidPointDebug(stop : Boolean)
```

Figure 4: Infrastructure Class List

An explanation of each of the attributes and operations is as follows:

- The *cellList* structure contains a listing of all the cells used in the problem.
- The *dp* is of type DectorPlane - it contains all the detectors used in the solution to the problem.
- The *drawables* list contains all the surfaces that are to be drawn during visualization.
- The *kernels* list-object contains all the terms passed into the calculation of Beer's law during the solution to the problem.
- The *midPoints* structure contains a listing of all the midpoints used in the calculation.
- The *pathPoints* structure contains a listing of all the path points (described later) used in the colocation.
- The *sp* and *tp* fields contains information about the source and target points, respectively.
- The *surfaceList* collection contains information about all the surfaces used in the problem.

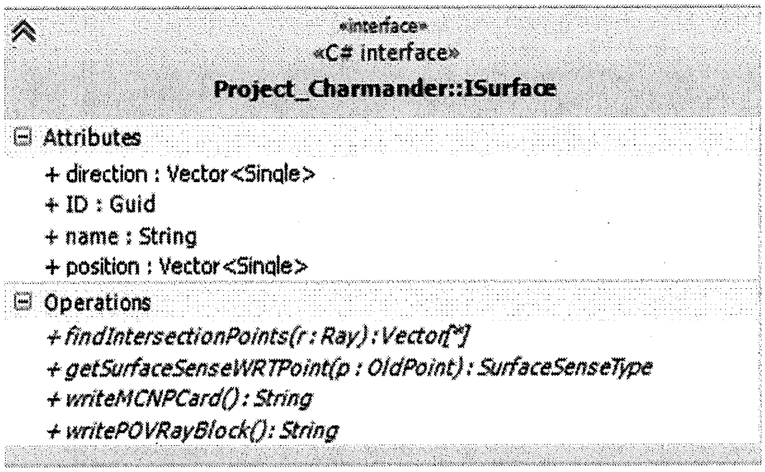
5.4. Step One – Surface Definition

All geometry in the code, like MCNP, is modeled as a parametric surface that partitions space into two sides – a positive and a negative side. Currently, only planar and cylindrical parametric surfaces are implemented.

In regards to surface senses: first, consider a plane that contains points **g** with position vector **t** and normal vector **n,** and a point in space **p**. The vector equation $g = (\boldsymbol{p} + \boldsymbol{t}) \cdot (\boldsymbol{n} + \boldsymbol{t})$ is solved. If the statement g = 0 is true, then point **p** is said to be coincident with the plane. If the statement g > 0 is true, the point is said to be positive with respect to the plane. Finally, if g < 0 is true, then the point is said to be negative with respect to the plane. As a corollary to this, the direction in which the plane is normal is important – that is to say, any given plane may be described identically by a given surface normal **n** or its negative **–n**; however, when given a point on either side of the plane, the surface sense of the point with respect to the plane will change signs depending on if the normal is defined as **n** or **–n.**

For a cylinder, to avoid unnecessary mathematics, it will suffice to say, for our purposes that if a point is inside the cylinder's radius, it is evaluated to be negative, and positive if it is outside of the cylinder's radius.

Each surface implements the interface "*ISurface*". This interface guarantees the class will implement the attributes and operations provided in Figure 5.



«interface»
«C# interface»
Project_Charmander::ISurface

Attributes

+ direction : Vector<Single>
+ ID : Guid
+ name : String
+ position : Vector<Single>

Operations

+ findIntersectionPoints(r : Ray) : Vector[*]
+ getSurfaceSenseWRTPoint(p : OldPoint) : SurfaceSenseType
+ writeMCNPCard() : String
+ writePOVRayBlock(): String

Figure 5: ISurface Attributes and Operations.

The *direction* and *position* attributes are vectors in three-dimensional space. They describe the spatial position and orientation of the surface. The ID attribute is a 128-bit integer that uniquely identifies the surface for later computational operations. The name is a user-friendly (human readable) name that is set by the user.

The *findIntersectionPoints()* method takes in a ray (line) and outputs an array of R3 vectors that correspond to the intersection points of the ray with the surface. The number of points in the array of intersection can vary from none to many – a plane can have either zero or one intersection points, whereas a cylinder can have up to two intersection points. If no intersection points are found, the array is set to null. It should be noted that this is distinct from the array being filled with null.

The *getSurfaceSenseWRTPoint()* method takes in a point in R3 and return the surface sense. Since each surface divides space into a positive and negative half, an arbitrary point in space will either be positive, negative, or coincident with each surface.

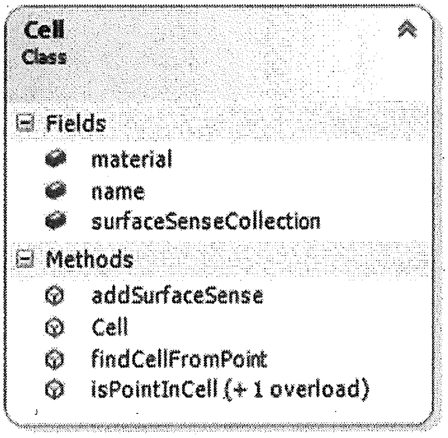
The *writeMCNPCard()* is a method – as yet unimplemented – that writes the geometry of the surface to a string that represents its MCNP equivalent.

The *writePOVRayBlock()* outputs a string for the surface to be visualized in the software package POV-Ray.

5.5. Step Two – Cell Definition

Conceptually, a cell is a region of space, bound by surfaces, that has some sort of attenuation coefficient associated with it. For example, if one considered a golf ball floating in the air, this could be modeled with two cells: one outside the golf ball and one inside the golf ball. Each of these cells would have an associated material and thereby attenuation coefficient.

In software, the cell concept is implemented as a class as shown below (Figure 6):



Cell
Class

Fields
- material
- name
- surfaceSenseCollection

Methods
- addSurfaceSense
- Cell
- findCellFromPoint
- isPointInCell (+ 1 overload)

Figure 6 Cell Class.

The material field is of type, Material. The material type has an associated enumerated (bound set of options) name and attenuation coefficient. The attenuation coefficient is implemented as a monoenergetic value, so only photons of a single energy can be modeled at one time; however, it would be fairly straightforward to implement poly-energetic attenuation coefficients by making the attenuation coefficient a function of energy.

The name, once again, is a user-friendly name that is used to describe the cell. Such an example could be "inside the ball" or "inside frying pan handle". It's implemented as a string type so there is no practical limitation on name length.

The *surfaceSenseCollection* field is a data structure that contains a number of two-tuples (dyadic object types) with a reference to a surface and its corresponding surface sense. This data structure stores the surface sense of each cell to each surface. It is used later in determining if a point is in a cell or not. Elements can be added to the collection with the *addSurfaceSense* method.

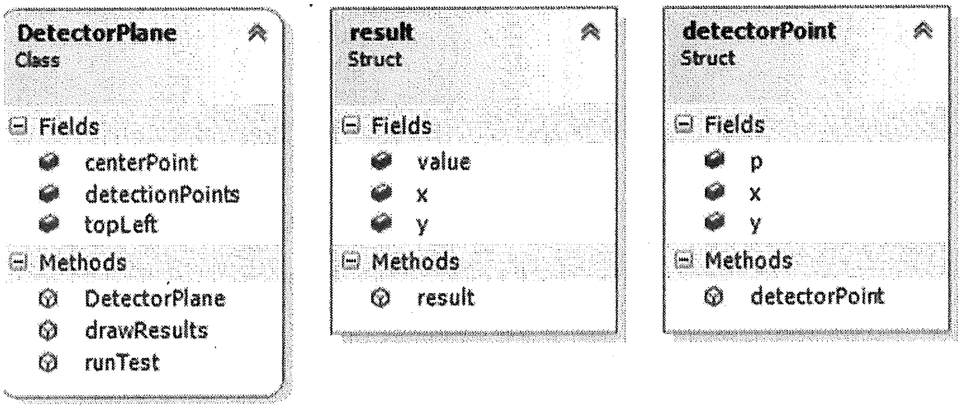
findCellFromPoint is a static method (stored in the cell class for naming convenience) that takes in a list of cells and a point, and returns a reference to the cell that the point is contained in. If this method cannot find a cell that contains the point passed in, it returns null.

Finally, the *isPointInCell* method takes in a point and returns a Boolean true or false value depending on if the point is in the instantiated cell.

5.6. Step Three – Source and Target Point Definition

The source and target points are where the ray tracing calculations start and end, respectively. They are simply points in R3 and members of the Point class, but they have special flags (from the PointType enumeration) to distinguish them from ordinary points.

There is also the special DetectorPlane class:



DetectorPlane
Class

Fields
- centerPoint
- detectionPoints
- topLeft

Methods
- DetectorPlane
- drawResults
- runTest

result
Struct

Fields
- value
- x
- y

Methods
- result

detectorPoint
Struct

Fields
- p
- x
- y

Methods
- detectorPoint

Figure 7: DetectorPlane, Result, and Detector Point Classes

The detector plane is centered about the R3 Vector named *centerPoint.* It is bounded by a width and height that are passed into its constructor and given an *up* and *left* vector that describe the relative directions of *up* and *left* for the plane. Finally, when instantiated, the constructor space-fills the detector plane with target points, at a resolution passed through a constructor parameter. Each *target point* is encapsulated in a *detectorPoint* structure and then appended to the detection points data structure.

The *runTest* method takes in a list structure containing a series of detector points and outputs an array of *result* structures, the value member of which contains the relative photon flux for each corresponding detector point. In this way, it can be said the *runTest* method maps a set of detector points into a result space.

Finally, the *drawResults* method takes in an array of result structures and outputs them as a bitmap image that has each pixel's intensity scaled, such that it is proportional to the value-member of its corresponding result-object. It should be noted that the *drawResults* method has an optionally normalized parameter – set to false by default – that when set to true will normalize each pixel in the output image to the highest measurement in the result structure; this allows for very low measurements to be easily visualized.

5.7. Step Four – Attenuation

Now that the source and target points are defined, the attenuation calculation is performed. This consists of the following steps:

1. Path point generation
2. Mid-point generation
3. Cell finding
4. Kernel generation
5. Final calculation of beam attenuation

The whole process can be interpreted, from a functional perspective, as a transformation of path-points to attenuation kernels with the method for finding the final attenuation of the radiation beam using the transformed data set.

The first step in this algorithm is to generate a set of points known as the path points which are defined to be all the points where there is a change in the path of the beam. This includes the source point, all the intersection points, and the target point. To achieve this, a data structure is created. It contains all the path points. Then, the source and target points are added to the data structure. Next, the array of intersection points is calculated for each surface in the problem, and the non-null results are added to the list of path points. Finally, since there is no guarantee that the methods that generated the intersection points did so in any particular order, the path points are sorted and ordered with respect to their distance from the source point. Next, a test is performed on each element in the path point data structure to see if it is outside the bounds set by the distance between the source and target point; no point should be before the source point, nor beyond the target point. The result of this process is that a list is generated. Each point of this list indicates the position of every major change in the ray's path from the source point to the target point.

Now that the path points are generated, the midpoint generation step can be performed. Thus, each consecutive pair of path points is taken and the following equation is applied to their position:

$$P_{mid} = \left(\frac{x_1 + x_2}{2}\hat{\imath} + \frac{y_1 + y_2}{2}\hat{\jmath} + \frac{z_1 + z_2}{2}\widehat{\boldsymbol{k}}\right) \tag{6}$$

The generated mid-point, by definition, should have the distance to both of its parent points equal. This result is stored in the *distanceFromParents* field, and is used elsewhere in the code in some error checking routines. This process is performed for each consecutive pair of path points, and resulting midpoints are then stored in the midpoints data structure.

Now that the midpoints have been obtained, the cell finding can begin. Thus, each element in the midpoint list is compared against each element in the cell list using the *isPointInCell* static method from the cell class. This is done by checking the surface sense of each element in the surface list against the position of the point. If there is a match between the surface senses, as described in the cell's surface sense data structure, with the surface senses found by the check, the midpoint is said to have found its cell, and the relationship is recorded. When all the midpoints have found their cell, the cell finding stage is over. If there is any point where a mid-point cannot find its cell, the ray is considered lost, and an error (exception) is thrown.

Once all the midpoints have found their cells, the attenuation kernel stage begins. In this stage, two quantities must be found: the attenuation coefficient, and the distance the beam transverses in the host cell of the kernel. The attenuation coefficient is found by simply referencing the attenuation coefficient of the cell that the midpoint was found in. The distance is found by multiplying the distance to parent path points (which are either source, target, or intersection points), by a factor of two (to compensate for the fact that the midpoint is found in the middle of the parent path points.) This process is performed for each mid-point, and the result stored in the kernels infrastructure component.

Next, the equation:

$$\frac{I}{I_0} = \exp(-\textstyle\sum_{i=0}^{n} \mu_i d_i), \tag{8}$$

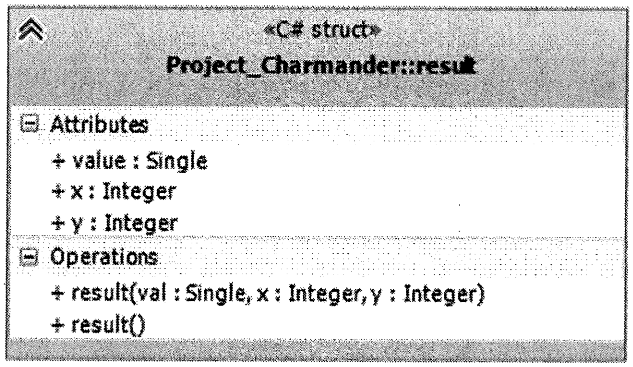
is solved for the 'n' midpoints, each with attenuation coefficient, 'μ_i' and distance, 'd_i'. The solution to this equation gives the total attenuation factor for a beam that has transversed from the source point to the target point. Finally, the geometric attenuation factor,

$$\frac{V}{V_0} = \frac{1}{4\pi(\sum_i r_i)^2}, \tag{9}$$

is applied to the previously calculated value, and the result recorded. In the previous equation, '$\frac{V}{V_0}$' is a dimensionless geometric attenuation factor, and 'r_i' is the distance between the source and the target. If an exposure time is given, the whole result is multiplied by the exposure time (in seconds). This entire process is repeated for all the points in a detector plane structure.

5.8. Step Five – Result Processing

After all the attenuation kernels have been processed for the detector plane, the results are stored in a result data structure shown Figure 8.



«C# struct»
Project_Charmander::result

Attributes
+ value : Single
+ x : Integer
+ y : Integer

Operations
+ result(val : Single, x : Integer, y : Integer)
+ result()

Figure 8: Result Data Structure

The field *'value'* is where the result is stored. The field '*x*' is the relative location on the x-axis of a detector plane where a result was obtained. Likewise, the relative location on the y-axis of a detector plane is stored in the '*y*' field. This structure is used in the next step – image processing.

5.9 Step Six – Image Processing

The final step is to save the results as an image. This is done with the *drawResults* static method of the *DetectorPlane* class. In this method, a result array is passed in, and the method finds the maximum x and y values of the array – these represent the width and height of the image. The method then uses the bitmap object of the *System.Drawing* namespace to create a new bitmap. Each result is mapped to a pixel, such that the intensity of the pixel is set to:

$$Intensity = (2^8 - 1) * \left(\frac{S}{S_0}\right)_{x,y} \tag{10}$$

Where the ratio $'\left(\frac{S}{S_0}\right)_{x,y}\ ' is$ the fraction of the original source strength at position (x,y) on the detector. If the optional *normalized* parameter is set to true, the channel intensity is set to:

$$Intensity = (2^8 - 1) * \frac{\left(\frac{S}{S_0}\right)_{x,y}}{Max\left(\left(\frac{S}{S_0}\right)_{x,y}\right)}. \tag{11}$$

The equation (10) is normalized to the maximum transmission factor. In this way, very small differences between background and signal can be distinguished. The normalization parameter, by default, is set to false.

5.10 Background Noise Considerations

The software also implements stochastic background-radiation generation. One can model the effect of background radiation as a Poisson process as given in [36]. It is described by the probability density function:

$$p(q;l) = \frac{e^{-l} l^q}{q!}, \tag{12}$$

and a standard deviation is given by

$$\sigma = \sqrt{l}, \tag{13}$$

where *'l'* is the mean, and *'q'* is the number of events – in this case, both of which are counts per minute.

The Justification for using a Poisson process to model the background radiation can be made as follows: If the Poisson process appropriately represents a given process in reality, it is conjectured that the Poisson standard deviation and the measured standard deviation will have similar values. This can be measured by the following equation:

$$percent\ difference = \frac{\left|\sigma_{measured} - \sigma_{poisson}\right|}{\frac{1}{2}\left(\sigma_{measured} + \sigma_{poisson}\right)}, \tag{14}$$

The background noise is then factored into the incident radiation flux by the equation:

$$S_{final} = S_{prime} + S_{background} \tag{15}$$

Where S_{prime} is given by:

$$S_{prime} = \frac{\epsilon I * \exp(-\sum \mu_i x_i)}{4\pi r^2\, I_0}. \tag{16}$$

the background factor is scaled by:

$$S_{background} = \epsilon\, g(\zeta). \tag{17}$$

In the above equations, 'S' is the source intensity at the measured point,

$g(\zeta)$ is a function that takes a randomly generated floating-point number, ζ, which ranges in value between 0 and 1. It is defined as follows:

$$g(\zeta)\ returns\ the\ value\ of\ x\ such\ that\ \int_0^x p(t)dt = \zeta. \tag{18}$$

The equation was solved by iteratively solving for the integral and accumulating the solution on a monotonically increasing interval. Thereby, in effect, the g function will provide a randomly generated quantity of the background noise based on Poisson-process. Thus, the total radiation influence on the detector at each location is determined.

5.11 Bloom Algorithm

To determine the *bloom-criteria* tolerance factor, an automated algorithm needed to be developed. To illustrate how the algorithm works, consider the following image (Figure 9).

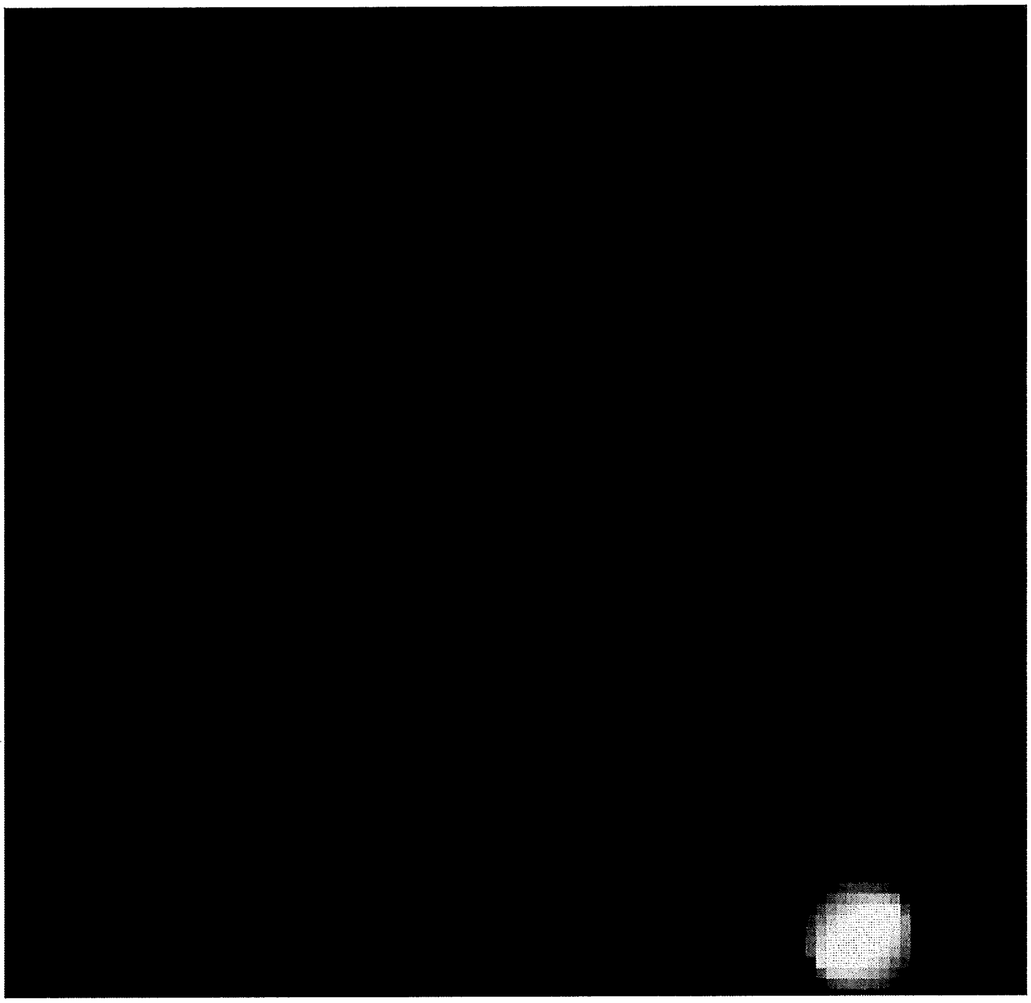


Figure 9: Example Image Bloom

The *diffuse spot* in the lower-right hand corner is caused by direct exposure from the radiation source. The algorithm automatically measures the area of these spots by the algorithm illustrated in the flow diagram, shown in Figure 10.

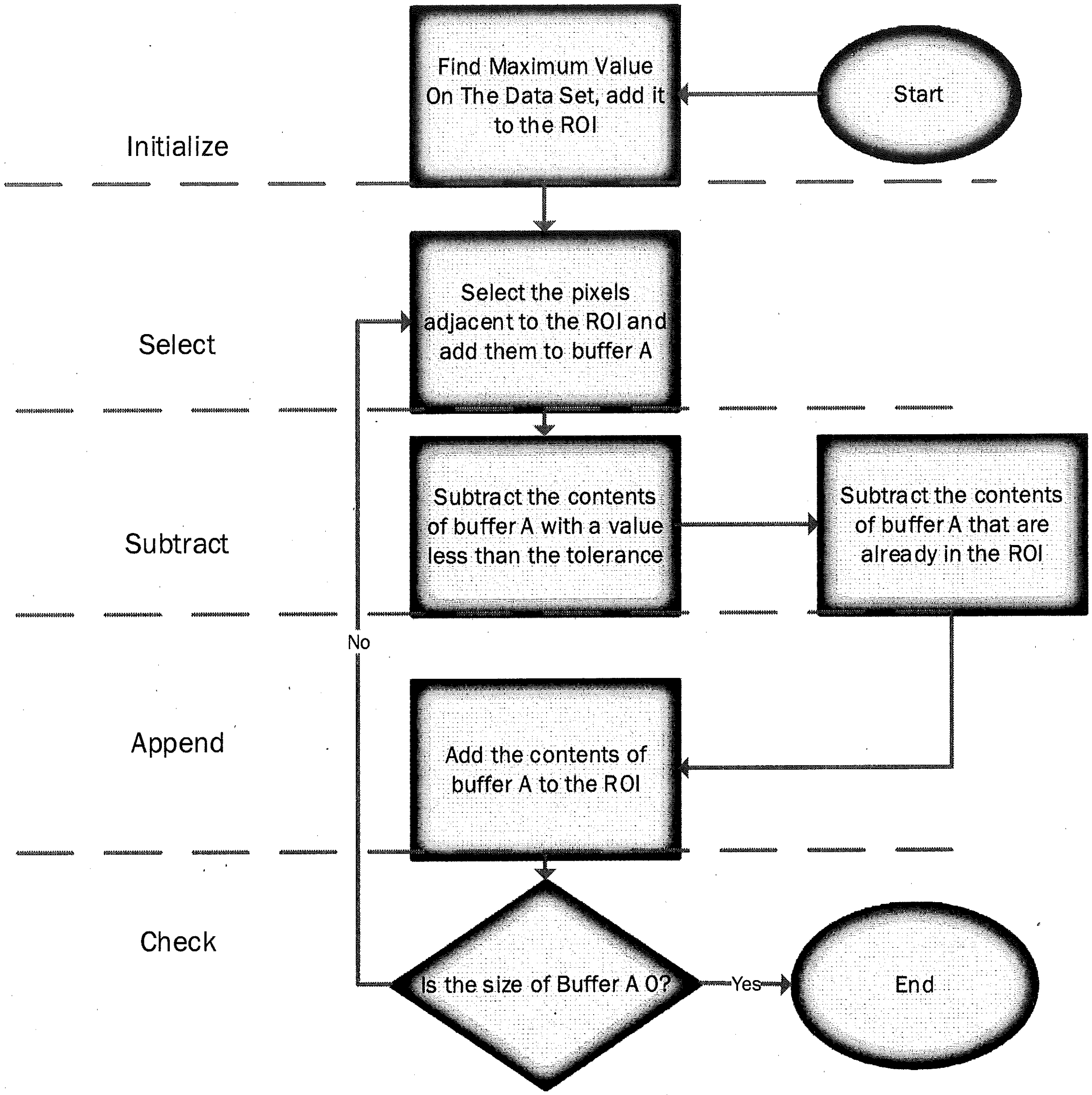

Initialize
Select
Subtract
Append
Check
Start
Find Maximum Value On The Data Set, add it to the ROI
Select the pixels adjacent to the ROI and add them to buffer A
Subtract the contents of buffer A with a value less than the tolerance
Subtract the contents of buffer A that are already in the ROI
Add the contents of buffer A to the ROI
Is the size of Buffer A 0?
No
Yes
End

Figure 10: Bloom Algorithm Structure

The algorithm has five stages: initialize, select, subtract, append, and check. There are two buffers that are used in this algorithm: the Region-Of-Interest (ROI) buffer and the A-buffer. In the initialize stage, the maximum value of the data-set is found and added to the ROI-buffer. Next, is the select stage in which all the pixels adjacent to that which are already in the ROI-buffer are selected and added to the A-buffer. In the subtracting stage, all the pixels that are either already in the ROI-buffer or have a reading less than the product of the tolerance factor and the maximum value are removed. In The appending stage, the contents of the A-buffer are added to the ROI-buffer. Finally, in the check stage, the size of the A-buffer is checked. If the size is non-zero, the program-pointer returns to the select stage to get another *layer* of pixels. If the size is zero, this means that no adjacent pixels survived the subtracting stage, and the ROI-buffer cannot expand any further with the tolerance criteria pre-defined. Finally, once the program pointer reaches the end, the size of the ROI-buffer is returned, thereby giving the number of pixels inside the region of interest that are greater than the tolerance value.

To detrmine the tolerance factor, a simulation was performed with the parameters listed in Table 2.

Table 2: Simulation Parameters

Pinhole Diameter	1.2 cm
Detector-Plane Distance to Source	1 m
Detector-Plane Width and Height	11 cm
Shield Thickness	1 cm
Exposure Time	600 s
X-Resolution	32
Y-Resolution	32

The size of the centrally exposed area was determined by inspection measured with the software program, ImageJ. The algorithm was *"parameter swept"* tested with a range of tolerance factors and the area measured by the algorithm was recorded.

The tolerance-factor that caused the algorithm to measure the same area that matches the hand measurement was selected as a final value.

5.12 Pinhole Gamma Camera Simulation Setup

For the simulations, a geometrical model has been built for the pinhole gamma camera as shown in Figure 11.

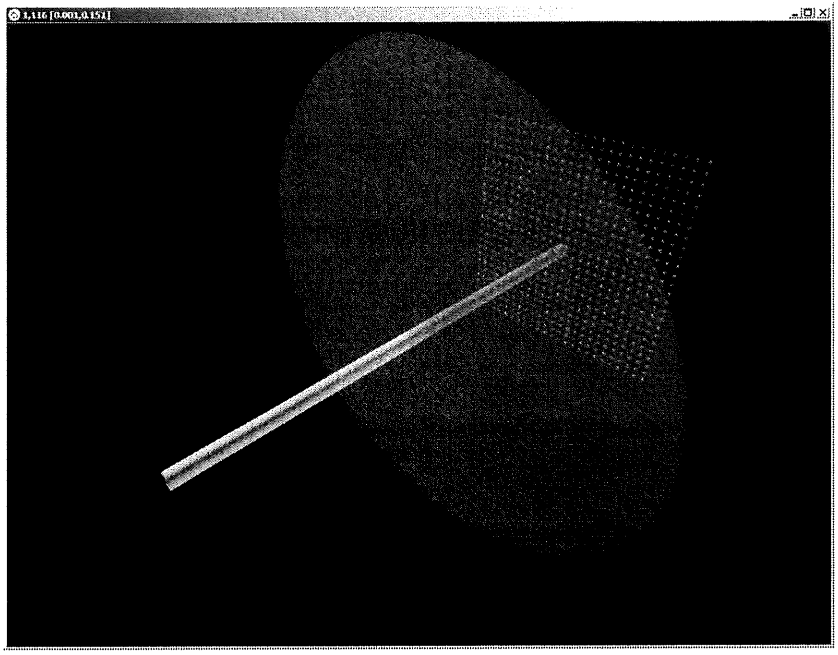


Figure 11: Pinhole Camera Geometry.

The steel shield is defined by the volume between the two disks, and outside of the cylinder. Everything else is in a vacuum. The detector plane is shown by the dots in Figure 11. Each dot represents one sample (detector) point. The source-point is not visible in this image. The dimensions of this geometry can be programmatically set.

6. Engineering Design Process

6.1. Overview

The fundamental problem encountered in the design of the gamma camera is that the entire system was highly coupled, and any change in a single variable could lead to many secondary and tertiary effects. Therefore, a top-down design process (the so-called waterfall method) was impractical. Instead, an iterative design-process was used to develop the gamma camera.

The variable design parameters used in this design process were: pinhole diameter, distance from the detector plane to the shield, and the shield thickness.

The process is illustrated in Figure 12 and described below.

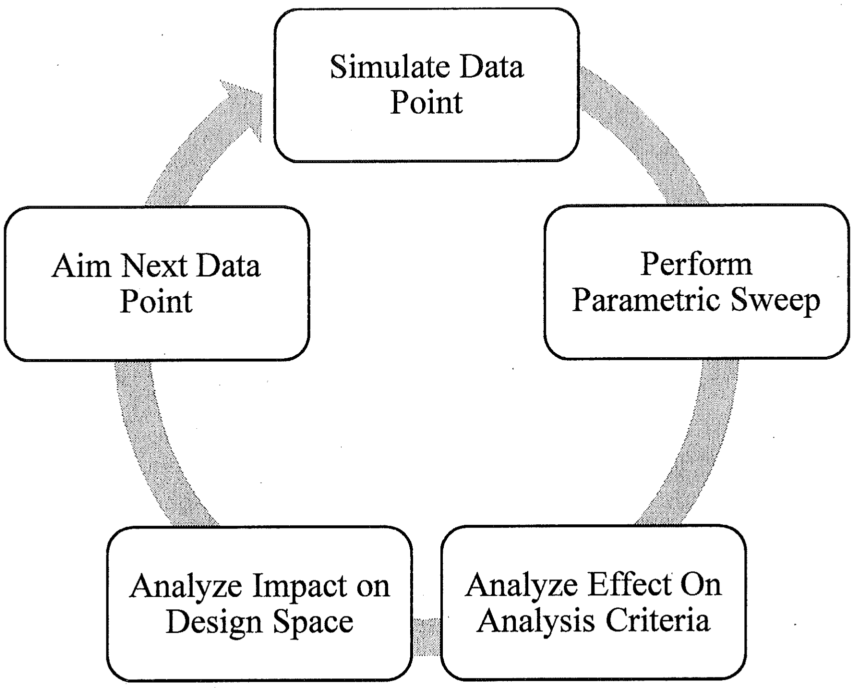

Simulate Data Point
Perform Parametric Sweep
Analyze Effect On Analysis Criteria
Analyze Impact on Design Space
Aim Next Data Point

Figure 12: Design Process.

The first step in the design process is the simulation of a data-point. A set of camera design parameters is chosen, based on a best-guess. The camera is simulated, and its performance analysis criteria are measured and recorded. The recorded data provides the first data-point. Next, the parametric sweep is performed. In this step, the design parameters are varied around the previously simulated data point. These data are compiled into charts. Further analysis of the data is performed to improve the camera performance.

In this step, the critical points are determined, and a decision is made to select the designer parameters.

6.2. Design Parameters

The illustration below shows the investigated design parameters:

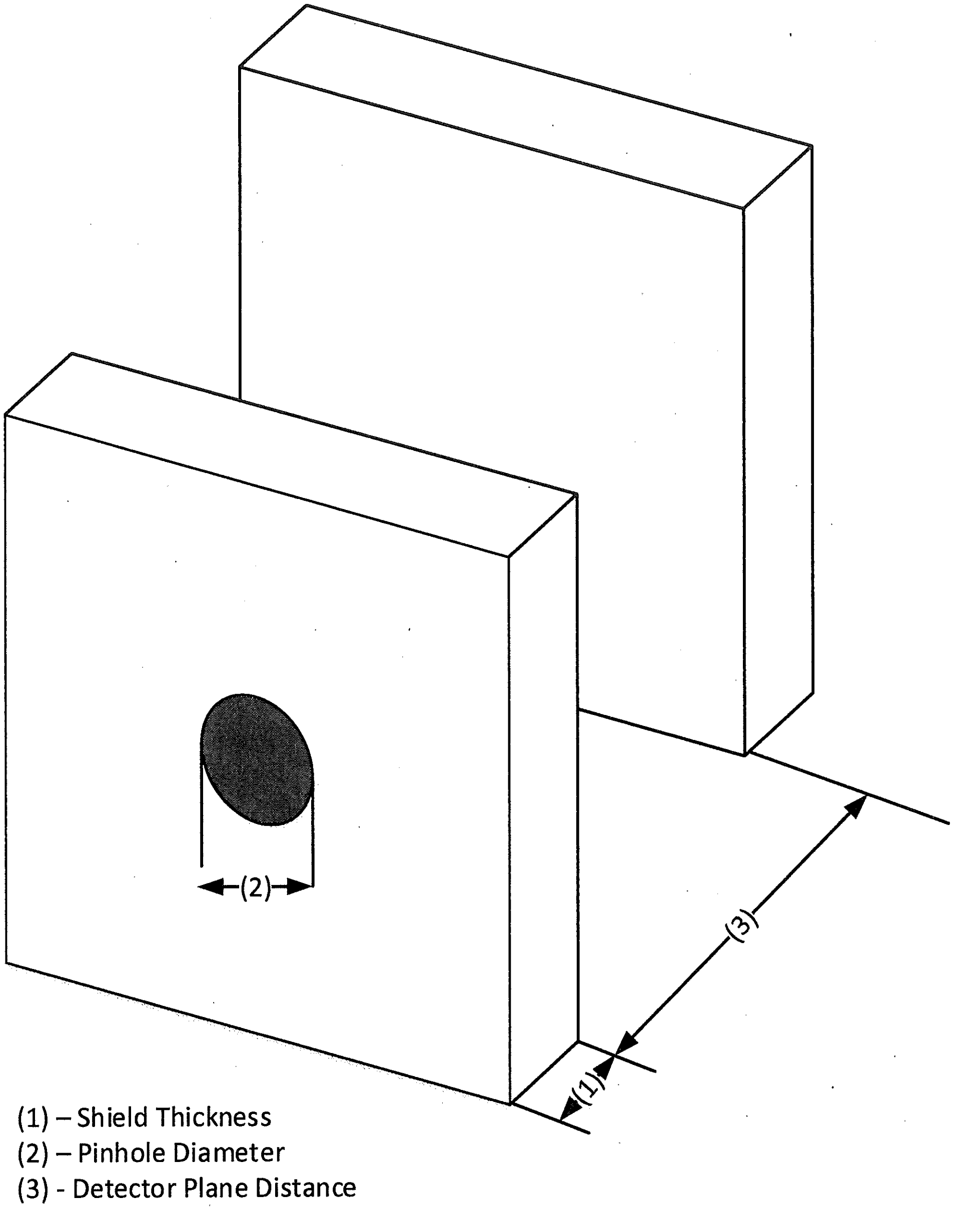


Figure 13: Design Parameters

1. **The Shield Thickness** - how thick the attenuating shield in front of the camera system is.
2. **The Pinhole Diameter** - the diameter of the pinhole that allows radiation into the system.

3. **The Detector-Plane Distance** – what distance between the shield and the detector plane is.

These three parameters are varied, and the results on the analysis criteria were measured.

6.3. Analysis Criteria

The performance of the system was calculated using analysis criteria:

- A – Maximum:
  - Maximize the reading on an image.
- B - Bloom
  - Minimize the area of a spot produced by a point source.
- C – Dynamic Range
  - Maximize the ratio of the maximum to minimum reading.

6.4. Simulation Parameters

A number of parameters are kept constant throughout all the simulations:

- **Detector-Plane-Distance.**
- **Total Exposure Time**: The total amount of time used to form an image.

$$(Time\ Per\ Measurement) = \frac{(Total\ Exposure\ Time)}{(X - Resolution) * (Y - Resolution)} \qquad (19)$$

- **X-Resolution**: The number of measurements taken per horizontal scan.
- **Y-Resolution**: The number of measurements taken per vertical scan.
- **Source Strength**: The activity of the radiation source used.
- **Detector Efficiency**: The number of photons registered in the image (detected) per photon emitted
- **Gamma Energy**: The energy of each photon emitted by the gamma source
- **Shield Material**: The material from which the shield is constructed.

6.5. Design Process Inputs

A total of two iterations of the design process were performed.

Simulation Parameters

The table below shows the values of the design parameters used as inputs into the design process.

Table 3: Simulation Parameters

Variable	**Value**
Detector-Plane Distance to Source	1 m
Total Exposure Time	3600 s (1 hour)
X-Resolution	32 pixels
Y-Resolution	32 pixels
Source Strength	1 mCi
Detector Efficiency	0.01
Gamma Energy	662 keV
Shield Material	Steel

Input one

The table below details the first set of design parameters

Table 4: Design Parameters Set One

Variable	**Value**
D - Pinhole Diameter	1.2cm
P - Detector Plane Distance	10cm
T - Shield Thickness	1cm

Input Two

The table below details the second set of design parameters.

Table 5: Design Parameters Set Two

Variable	**Value**
D - Pinhole Diameter	0.5 cm
P - Detector Plane Distance	9.2 cm
T - Shield Thickness	3 cm

Chapter 3: Results and Discussion

1. Preliminary Work

1.1. Signal Attenuation

Let us consider our hypothetical gamma camera with a 500 keV source placed at a distance of 1 m from the camera. The radiation attenuation can be calculated by

$$\frac{S}{S_0} = \frac{\exp\left(-\frac{\mu}{\rho}\rho x\right)}{4\pi r^2}, \tag{20}$$

Where '$\frac{S}{S_0}$' is the fraction of the original source strength being evaluated at a distance 'r' away from the source; '$\frac{\mu}{\rho}$' is the density-normalized attenuation coeffecient of dry air ($8.712 * 10^{-2}\left[\frac{cm^2}{g}\right]$ according to [38]); and 'ρ' is the density of dry air ($1.275 * 10^{-3}\left[\frac{g}{cm^3}\right]$ according to [39]). Substituting in these values in equation (20):

$$= \frac{\exp(-8.712 * 10^{-2}\left[\frac{cm^2}{g}\right] * 1.275 * 10^{-3}\left[\frac{g}{cm^3}\right] * 100\ [cm])}{4\pi * (100\ [cm])^2} \tag{21}$$

$$= \sim 8 * 10^{-6}\frac{photons}{(source\ photon) * cm^2}.$$

Furthermore, if a Geiger-Muller detector is used, its intrinsic efficiency according to [40] is about 1%, so a Geiger-Muller tube at this distance would receive a signal strength relative to the source of about 8E-8.

Thus, if the source strength was 1mCi (3.7E7 Bq), then a Geiger-Muller tube would register about 3CPS.

1.2. Measurement Time

A 10 x 10 grid was established for calculation. Each point was calculated for ten seconds. It was found that a complete scan could be taken in about 17 minutes. Given that the previous calculation predicted a reading of three counts per second, a ten second measurement could potentially produce 30 counts.

1.3. Pinhole Shielding

One consideration in the design was the thickness of the pinhole shielding. The first step in the shielding consideration would be to decide on a shielding material. The shielding material must be of low-cost and provide reasonable attenuation of incoming gamma rays. The most common choice for radiation shielding material is lead; however, due to the safety concerns of working with such a material, it was deemed that the use of lead was outside of the design space of the project.

The next obvious choice for a shielding material is steel which has a relatively high atomic number, low in cost, and safe to work with. According to [41], given a source of Cobalt-60 (a relatively high energy gamma emitter), steel has a half-value layer thickness of 21.6mm (~2cm). Given that steel is readily available in ¼" thick plates (6.35mm), this would mean that only 4 plates of steel would be needed to attenuate the incoming beam by more than half. Furthermore, since steel has a density of about 7.6 g cm^{-3}, a steel plate measuring 2x50x50 cm^{3} would weigh 38kg – a mass that can be lifted by a single person.

1.4. MCNP Simulation

The results of the MCNP simulation developed are shown in the image of Figure 14.

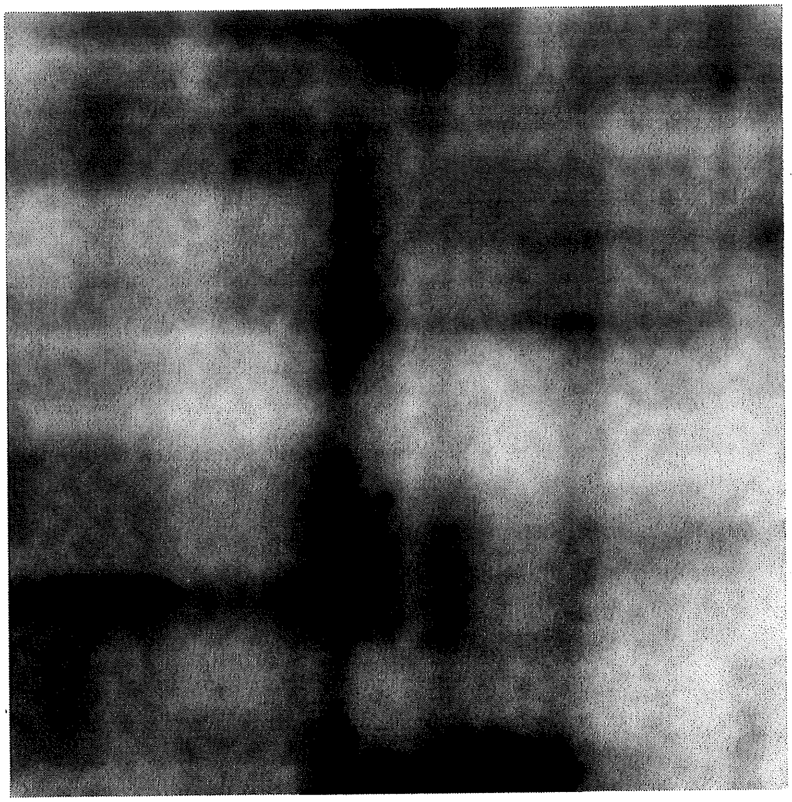


Figure 14: First MCNP Results.

As can be seen from the image, very little detail can be seen. Furthermore, the error for each pixel was higher than 50%. Nevertheless, there does appear to be two distinct regions of increased activity. These regions are shown in Figure 15.

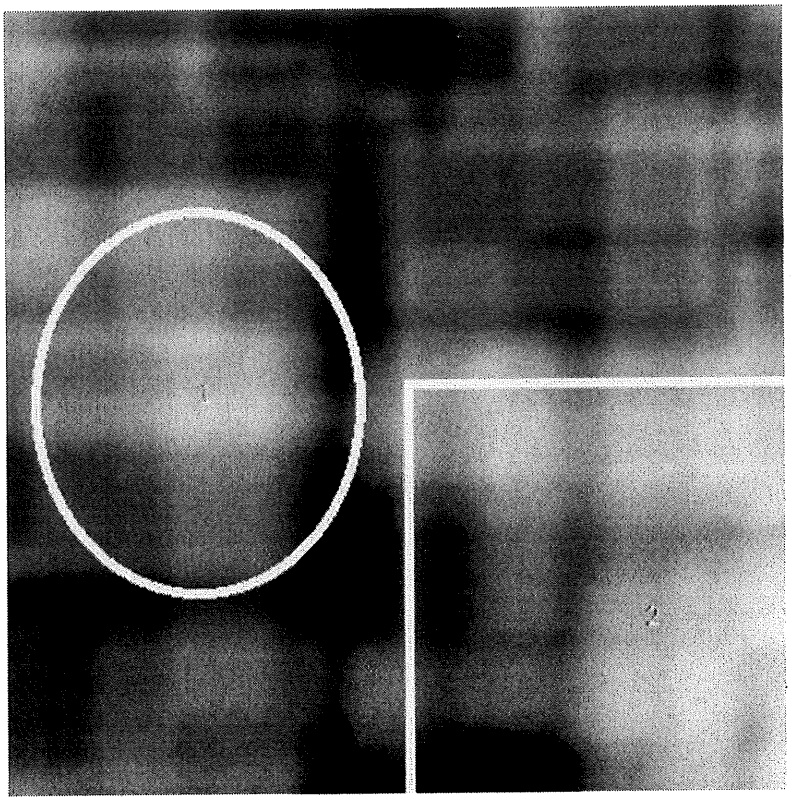


Figure 15: Highlighted MCNP Results.

Region 1 is probably created by the gamma photons directly impinging on the detection plan while region 2 is probably created by radiation that is first attenuated by the shield. The larger area of region 2 is a result of the scattering effect of the gamma photons traveling through the shield. Conversely, it is also possible regions 1 and 2 are nothing more than noise created by error, and the image above does not show any results.

Nevertheless, the entire simulation took more than a month to run. Hence, an alternative technique had to be developed.

2. Mathematics

2.1. Beer's Law for Multiple Materials

Beer's law describes how a beam of light is attenuated as it passes through a medium. It is a first order linear differential equation:

$$\frac{dI}{dx} = -\mu I, \tag{22}$$

where 'I' is the beam's intensity, 'x' is the distance the beam traverses through the material, and 'μ' is the attenuation coefficient.

In summary, this equation states that the rate that the intensity of a beam of light decreases as it transverses a medium is inversely proportional to the intensity of the beam. The solution to this equation is:

$$I = I_0 \exp(-\mu x). \tag{23}$$

If the beam passes through multiple materials, then each material will attenuate the beam by its respective amount, and the expression becomes:

$$I = I_0 \exp(-\textstyle\sum \mu_i x_i). \tag{24}$$

The beam intensity relative to its initial intensity is called the transmission factor. It is given by the ratio '$\frac{I}{I_0}$'.

Furthermore, if the beam originates from a point-source, it will be attenuated geometrically by the inverse square law,

$$\frac{V}{V_0} = \frac{1}{4\pi r^2}. \tag{25}$$

Where $\frac{V}{V_0}$ is the geometric attenuation factor.

The final flux at a point at distance *'r'* away from the source is given by the product of the geometric and material attenuation

$$\frac{S}{S_0} = \frac{I}{I_0} * \frac{V}{V_0}, \qquad (26)$$

where $\frac{S}{S_0}$ is the total attenuation factor.

Thus, given a point source emitting *'S_0'* emanations per second, and a point in space at a distance, *'r'* away will experience a particle flux of

$$S = \frac{S_0 \exp(-\sum \mu_i x_i)}{4\pi r^2}, \qquad (27)$$

where μ_i is the attenuation coefficient of the ith material that the beam travels a distance x_i through.

2.2. The Intersection of a Ray with a Plane

Given a point in space **p,** normal **n**, and intersecting point **g**, the equation of a plane can be written as

$$(\boldsymbol{p} - \boldsymbol{g}) \cdot \boldsymbol{n} = \mathbf{0}. \qquad (28)$$

The equation of a ray in R3 is:

$$\boldsymbol{l}(\lambda) = \boldsymbol{t} + \lambda \boldsymbol{v}, \qquad (29)$$

where **l** is an arbitrary point on the ray, **t** is a point that the line passes through, λ is a scalar, and **v** is a normalized direction vector.

To find where the line intersects the plane, we substitute in the equation of the line for the arbitrary point on the plane

$$(\boldsymbol{t} + \lambda \boldsymbol{v} - \boldsymbol{g}) \cdot \boldsymbol{n} = \mathbf{0}. \tag{30}$$

Rearranging this for λ, we find that the plane crosses the line at the scalar

$$\lambda = \frac{(\boldsymbol{g}-\boldsymbol{t})\cdot\boldsymbol{n}}{\boldsymbol{v}\cdot\boldsymbol{n}}. \tag{31}$$

The point in R3 where the line intersects the plane is found by substituting the result from the above equation back into the equation of a ray

$$\boldsymbol{x} = \boldsymbol{t} + \left(\frac{(\boldsymbol{g} - \boldsymbol{t}) \cdot \boldsymbol{n}}{\boldsymbol{v} \cdot \boldsymbol{n}}\right)\boldsymbol{v}\,. \tag{32}$$

2.3. The Intersection of a Ray with a Cylinder

The cylinder is described by a point **p**, an axis parallel to the unit direction vector **v**, and has a radius 'r'. The Normal of such a cylinder is a vector that is drawn from the axis to the surface of the cylinder. This is visualized in Figure 16.

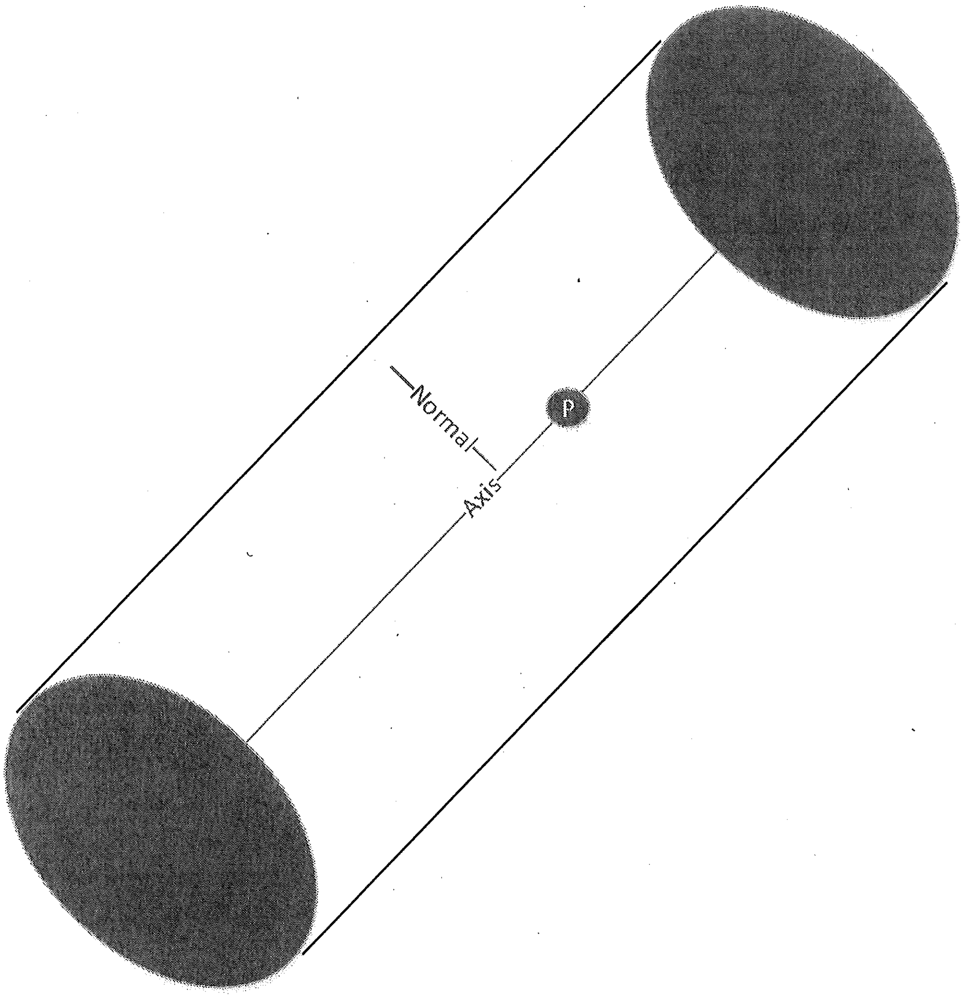


Figure 16: Cylinder under analysis.

Let us consider a ray, **l** intersecting such a volume, as in Figure 17.

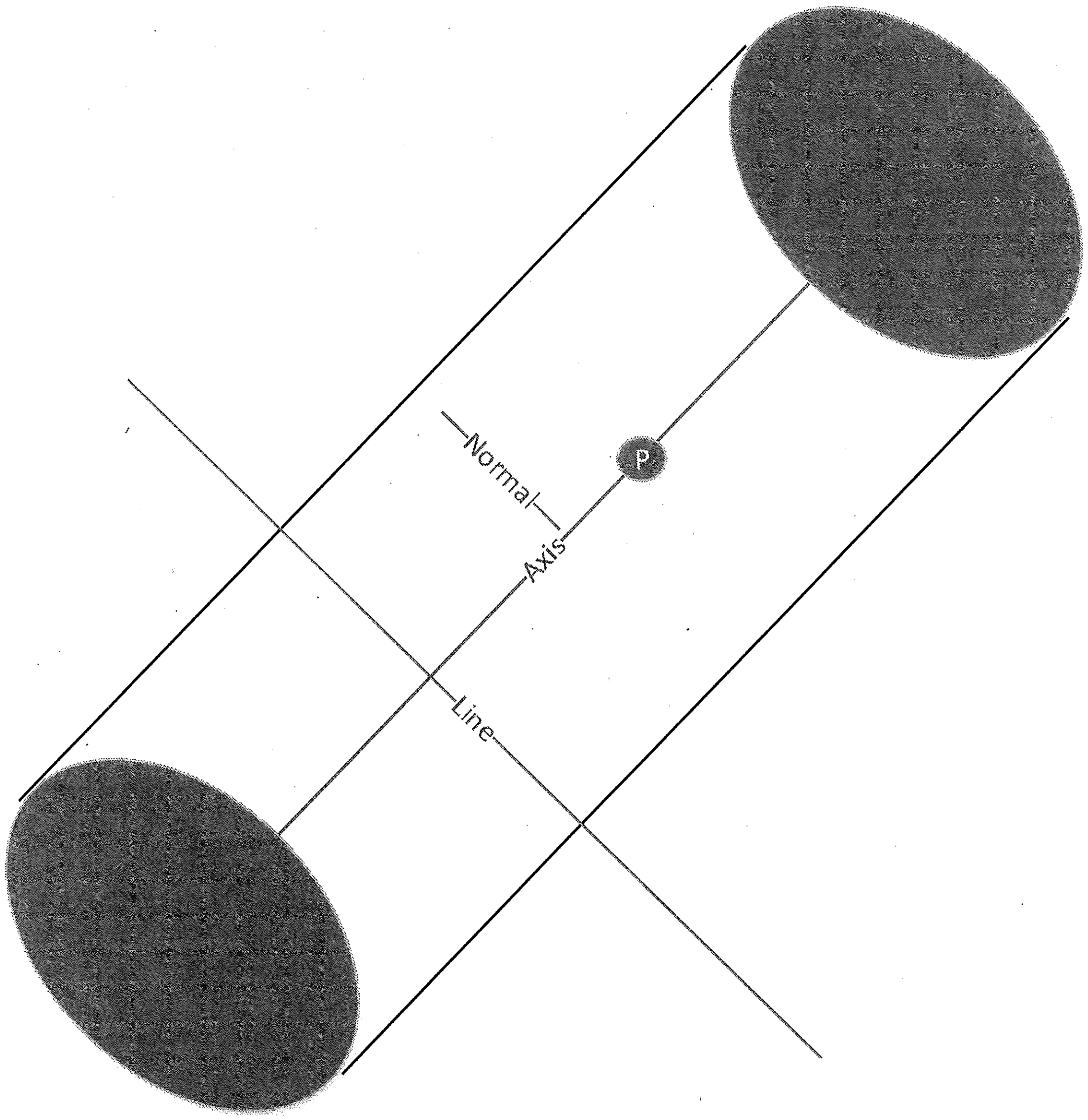


Figure 17 Ray Intersecting Cylinder.

The first step in finding the intersection points is to project the ray such that it passes through the axis of the cylinder (Figure 18). This line will be called **K.**

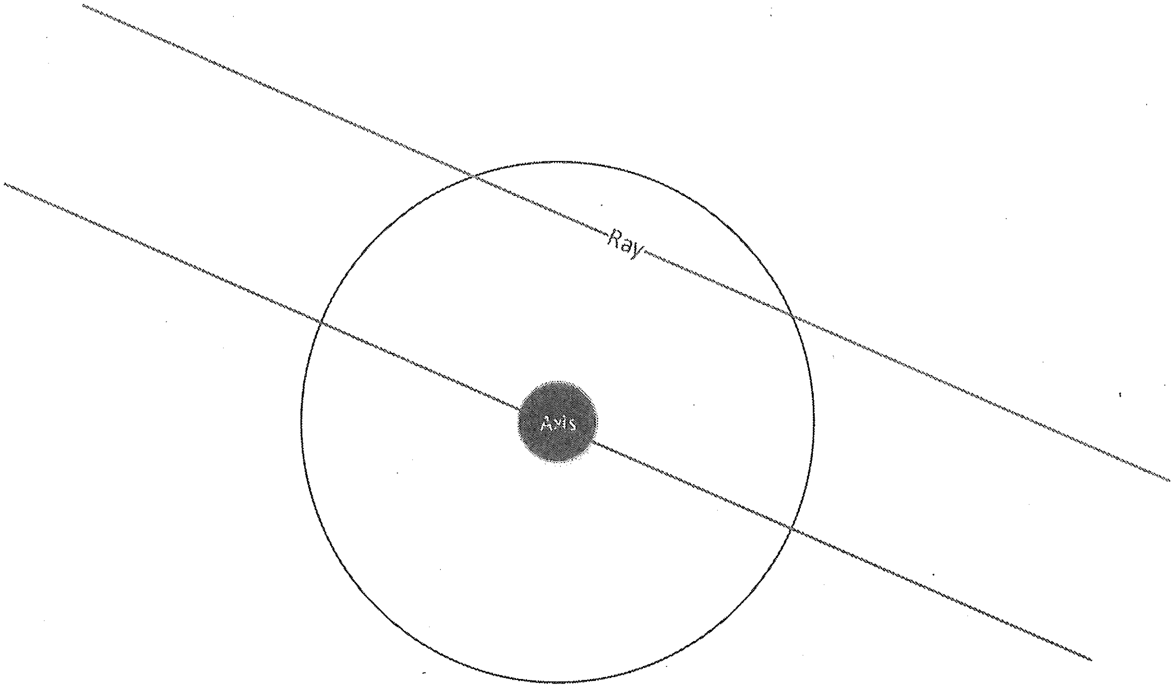


Figure 18: Top View of the Cylinder under Analysis

Once this is done, a new vector, **J** must be drawn such that it is perpendicular to both the axis and the projected line **K** (Figure 19). The angle between the **K** line and the axis referred to as the **B Angle** for reasons that will become clear later.

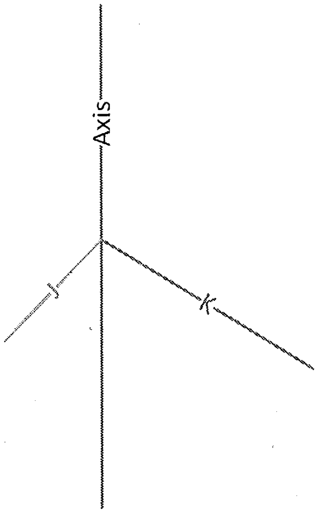


Figure 19: Basis for Vectors J and K.

Next, a plane (the cut-plane) is drawn using the vectors **J** and **K** and passing through the point at which they cross the axis. The intersection of this plane and the cylinder is an ellipse, as shown in Figure 20.

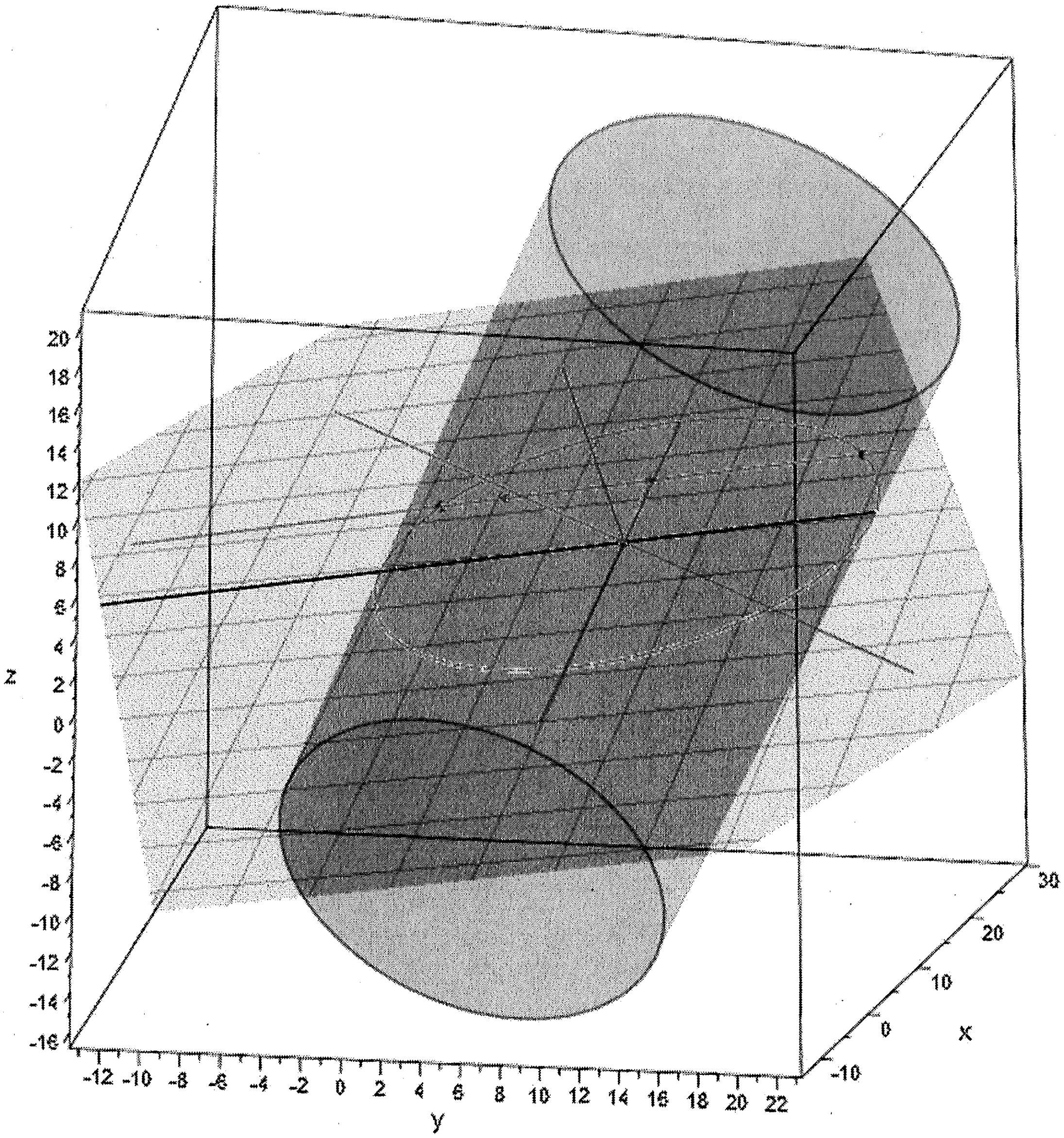


Figure 20: Three Dimensional Visualization

The equation of an ellipse with its center at (0, 0) is given by:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \tag{33}$$

where 'a' is the radius on the x-axis and 'b' is the radius on the y-axis. Since the ellipse has been generated such that its y-axis is perpendicular to the axis of the cylinder, the 'a' term is simply the radius of the cylinder. The 'b' term, on the other hand, has been stretched by a factor that is dependent on the angle formed by the intersection of the plane with the cylinder. The ellipse is illustrated in the image shown in Figure 21.

Thus, the original three-dimensional problem has been reduced to a two-dimensional problem – a ray ellipse intersection.

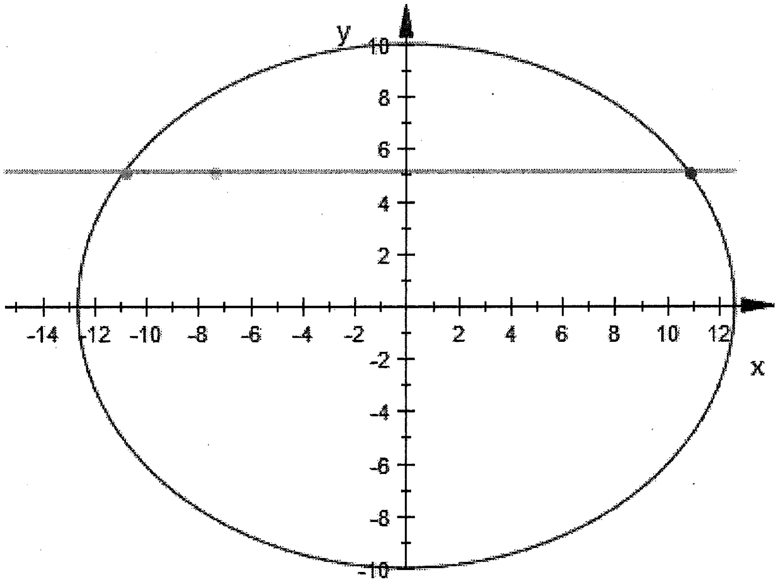


Figure 21: Two-Dimensional Cut Illustration

If looked at from the side, the cylinder-plane intersection will appear as shown in Figure 22.

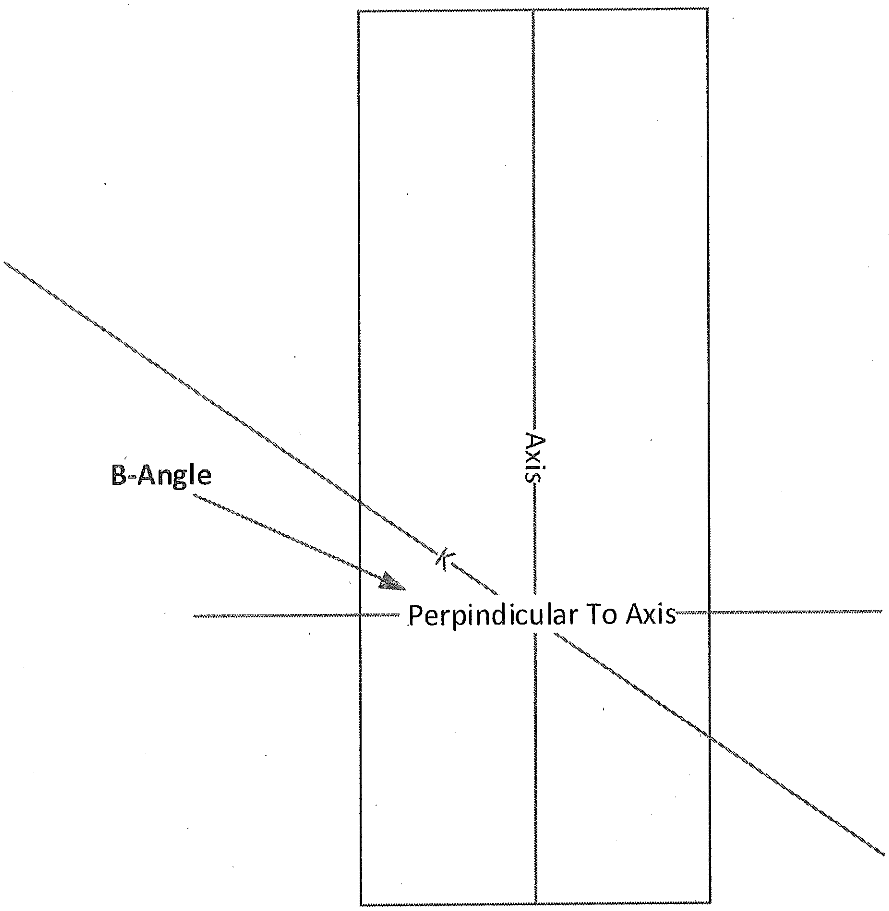


Figure 22: Side View of Cylinder-Plane Intersection.

As we can see, the B-Angle is the angle between the K line and the perpendicular to the axis. Thus, the ellipse will be elongated by a factor, $b = r * \cos(B\ angle)$.

Finally, the line in R3 is projected onto a cut-plane, and the intersection points found by solving the resulting equation by the method mentioned in [35].

2.4. Calculation of Minimum Detection Plane Size

We seek to find the minimum detection plane size. Thus, consider a horizontal slice of the rasterizing pinhole camera represented by the geometry provided in Figure 23

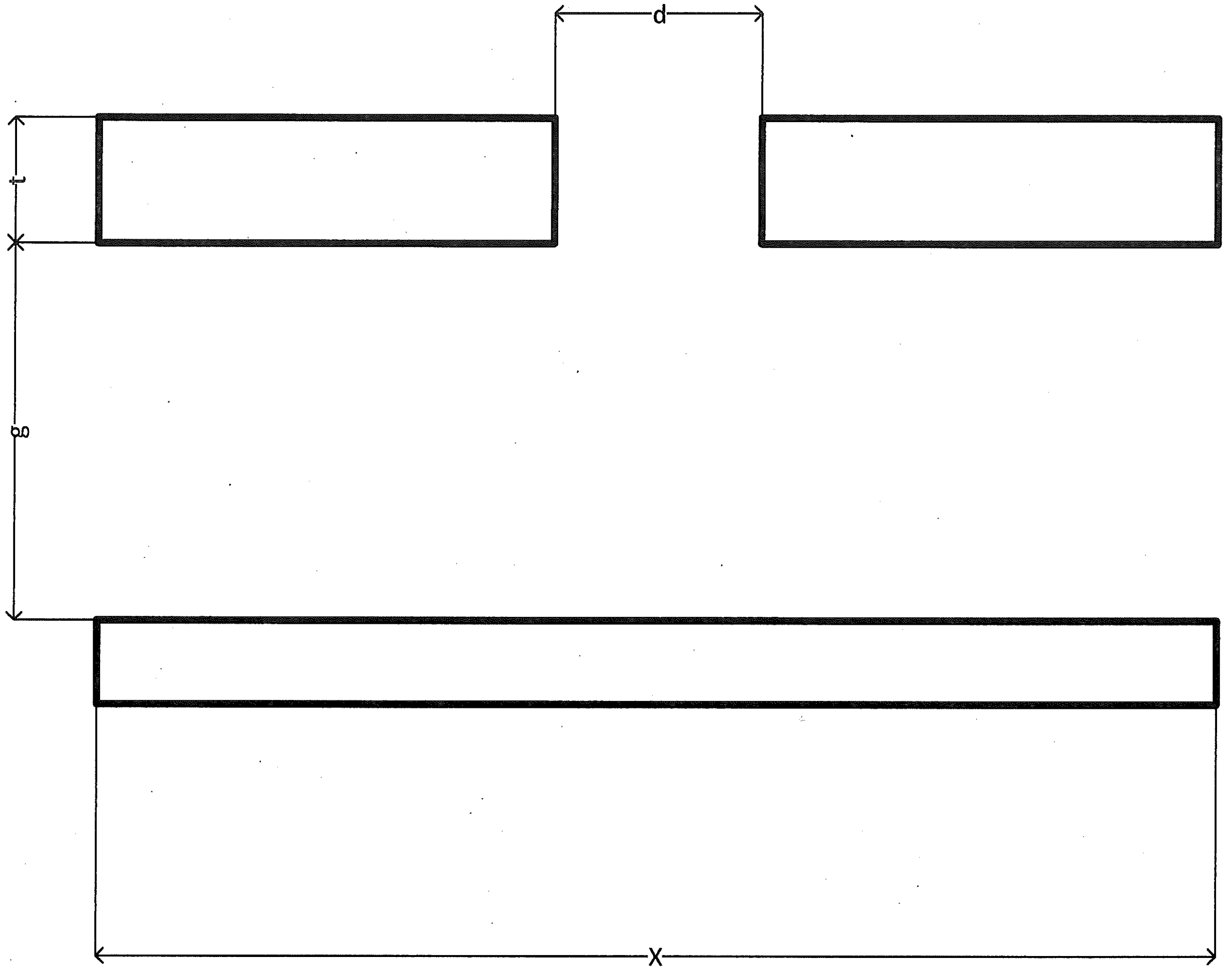


Figure 23: Top View Slice of System

Please note, this slice has been made exactly about the mid-plane, so that there is no material present in the pinhole. Here, 'd' represents the pinhole diameter, 't' represents the shield thickness, 'g' represents the distance between the back of the shield to the front of the detector plane, and x represents the length of the detection plane.

The maximum distance an unattenuated ray may travel on one side, by the methods of geometric optics, is illustrated in Figure 24.

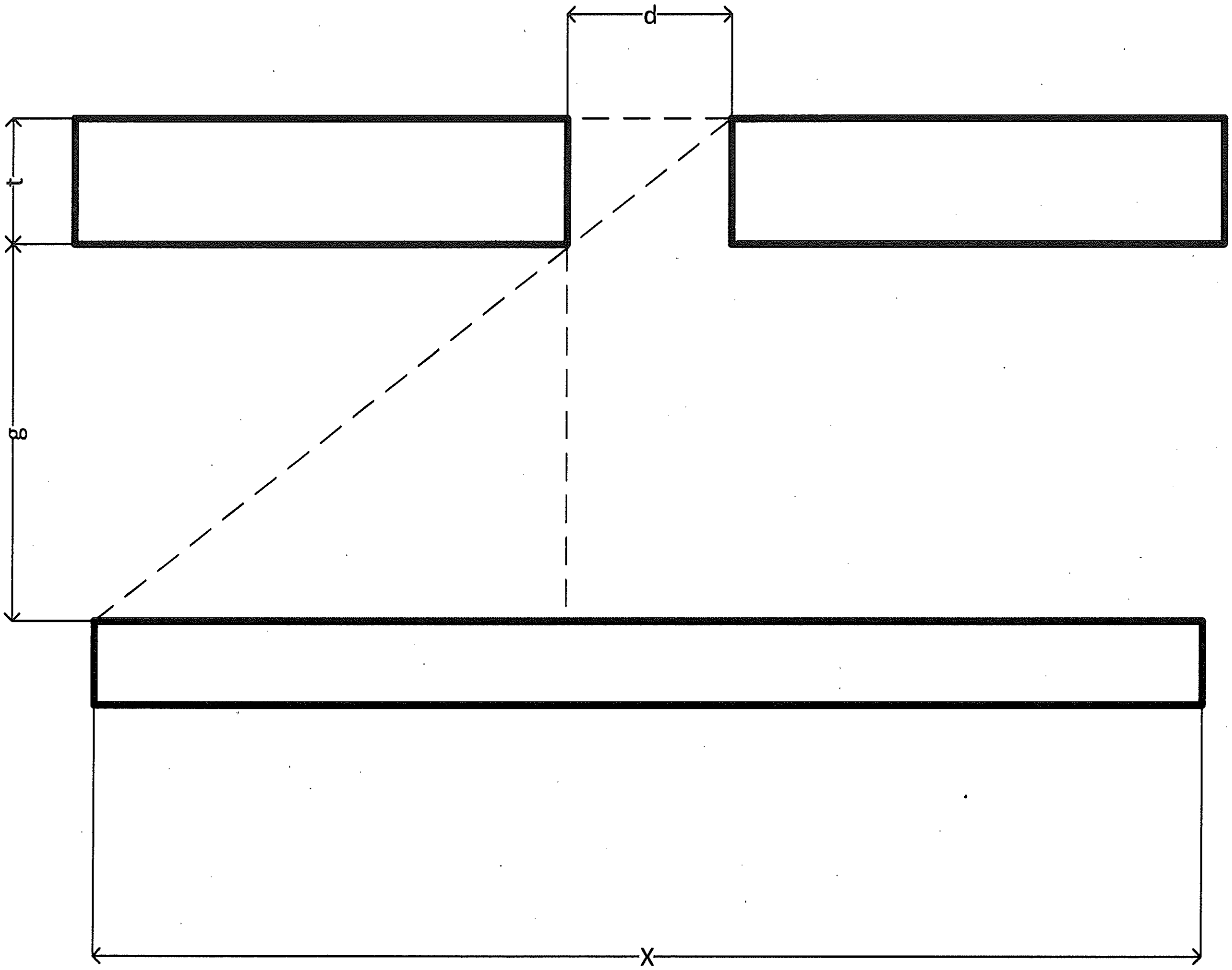


Figure 24: Top View Slice of System with Geometry Drawn In

If we isolate the triangles from the setup, we have the following set of similar triangles, illustrated in Figure 25.

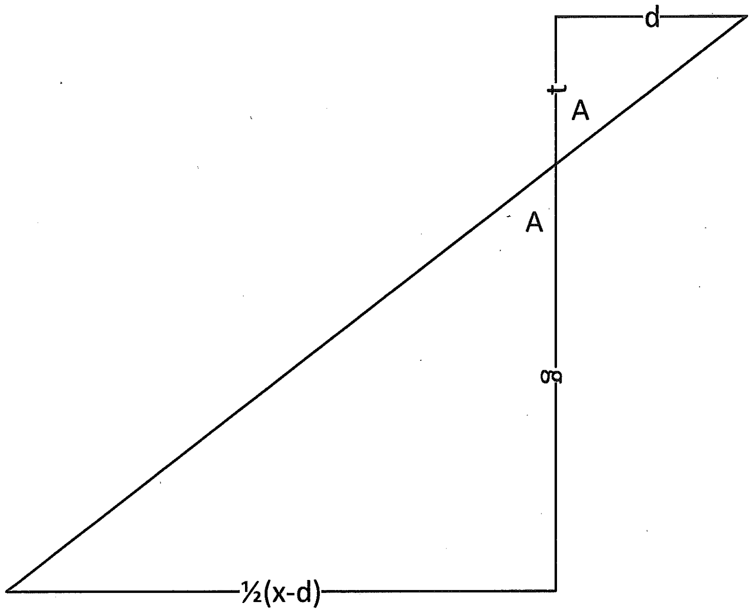


Figure 25 Geometry Extracted.

From which, the following derivation may be made:

$$\tan(A) = \frac{d}{t}, \tag{34}$$

$$\tan(A) = \frac{\left(\frac{1}{2}\right)(x-d)}{g}. \tag{35}$$

These two equations are set equal to one another through:

$$\frac{d}{t} = \frac{\left(\frac{1}{2}\right)(x-d)}{g}, \tag{36}$$

and solving this equation for x, generates:

$$x = \frac{2dg}{t} + d. \tag{37}$$

This equation will give us the minimum detector plane dimension that captures all the primary incoming rays.

3. Background Noise Measurement

It was found that the mean number of recorded counts in 60 seconds was 11.8 counts with a standard deviation of 3.6. This works out to approximately 0.2 counts per second. If the efficiency of the detector is 1% as suggested in [40], then the true rate at which particles pass through the detector is closer to 20 particles per second.

As mentioned in the 712 data sheet [42], the window has a diameter of 9.1 mm. This, in turn, corresponds to a window area of 0.65 cm^2. Contributions to the count-rate can come from either, particles passing through the detector window, particles passing through the detector casing, or electronic noise. If we assume that the particles are passing through the detector casing, and the electronic noise bears an insignificant contribution to the count rate, then the flux can be found by:

$$\phi = \frac{count\ rate}{window\ area} = \frac{20\ cps}{0.65\ cm} = 30\frac{count\ producing\ particles}{cm^2 s}. \tag{38}$$

In order to ensure conservative calculations are made, a safety factor of three will be introduced. Thus,

$$\phi_{background} = 90\ cm^{-2}s^{-1}. \tag{39}$$

The standard deviation given by the Poisson process model was found to be 3.6. The measured average from the experiment was found to be 11.8 – this value has a square root of 3.4. Thus, the percent difference between these two values is 6. Since the expected and measured standard deviations are so close together, this justifies the use of the Poisson process to model background radiation.

4. SOFTWARE DEVELOPMENT

4.1. Software Validation

4.1. Mean Chord Length

The first validation test was designed to ensure that the code is calculating the correct locations for the intersection points and the exact distances between cells. The test calculates the mean chord length of one million random chords that intersect a convex solid, and compares that result to the expected mean chord length calculated by the analytic expression, Cauchy's Formula.

The code was set up to find the mean chord length of a right circular cylinder with a height of 4.5 cm and a radius of 0.8 cm.

The random chords were created by randomly selecting two points in space that are outside of the bounding volume, and drawing a chord between them. Once the chord is found, and if and only if it intersects with the cylinder, the distance between the intersection points is calculated and recorded. The average of the recorded results is then calculated.

Cauchy's formula [43] is given by:

$$l = \frac{4v}{s}, \tag{40}$$

where l is the length of a random chord transversing the solid, v is the volume, and s is the surface area. It is an analytical expression that gives the mean length of a randomly selected chord that transverses a solid convex shape.

When evaluated for the solid under consideration, we find a mean chord length, as given by Cauchy's formula, of 1.36 cm.

The simulation calculated the length of 1 million random chords and found a mean chord length of 1.34cm. A plot of the findings is shown in Figure 26. The result is within 1.4% of the expected value.

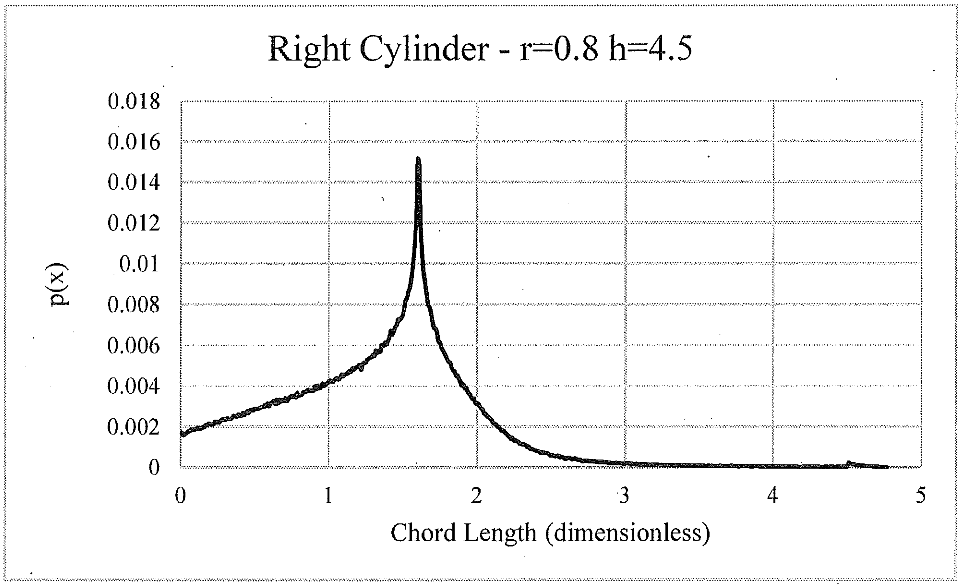


Figure 26: Chord Length for a Cylinder with Radius of 0.8 and Height of 4.5.

4.2. Slab Attenuation

The next test is a slab attenuation test. In this test, the ability of the code to perform an attenuation calculation was tested. Attenuation calculations can only reasonably be done by hand in a very limited number of situations. One typical situation is the so-called slab geometry attenuation. In this test, a beam is attenuated by a slab of finite depth and infinite width and height.

In the attenuation setup, a lead slab with a thickness of 1cm was constructed. The detector was placed 11cm behind the slab and the source was placed 9cm in front of the slab. A linear attenuation for a 500 keV photon passing through lead of 1.72 cm^{-1} from [44, p. 53] was used. The equation to calculate the transmission was:

$$\frac{S}{S_0} = \frac{\exp(-\mu x)}{4\pi r^2} \qquad (41)$$

where μ is the linear attenuation coefficient, 'x' is the slab distance, and 'r' is the distance between source and target.

The code calculated a result that was in complete agreement with the analytical calculation. Using both methods, the code calculated a value of 3.56E-5.

4.2. Sample Simulation

The results of this method are shown in Figure 27.

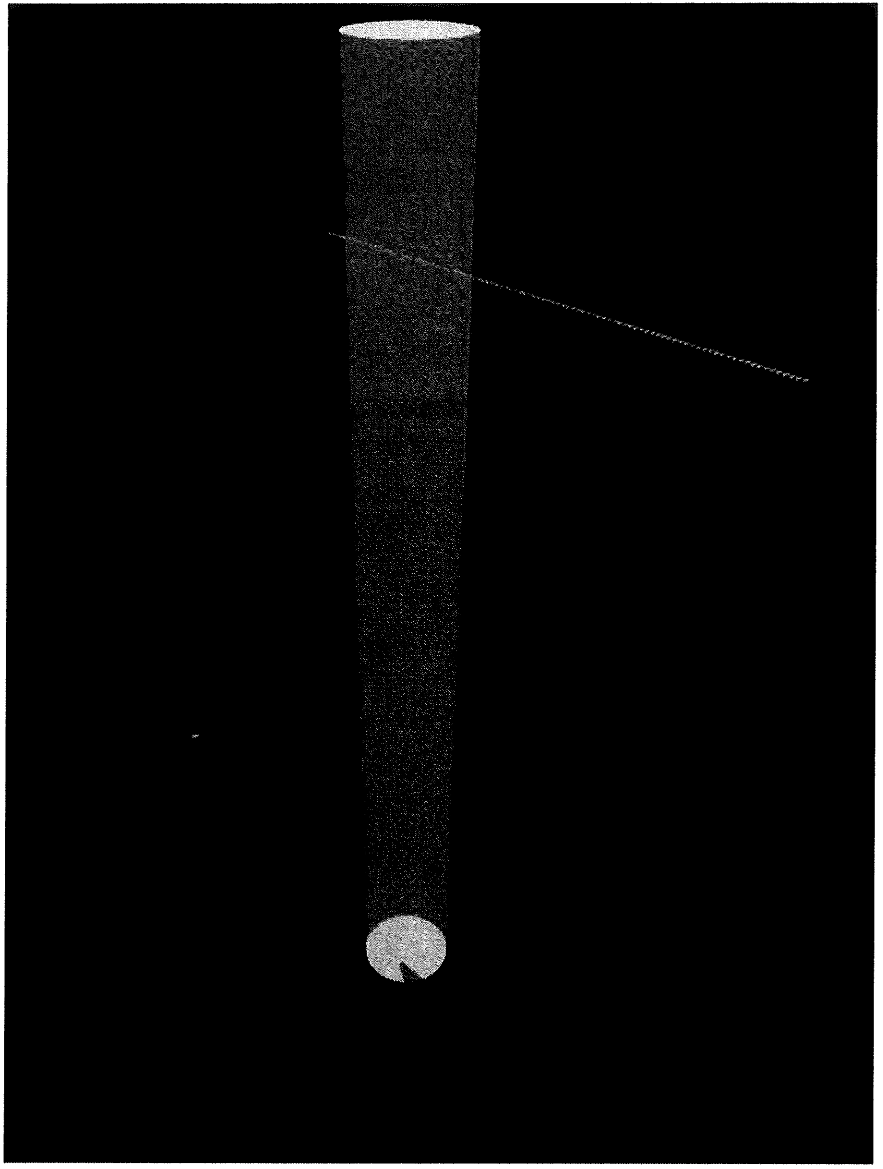


Figure 27: Visualization of Geometry for Sample Simulation

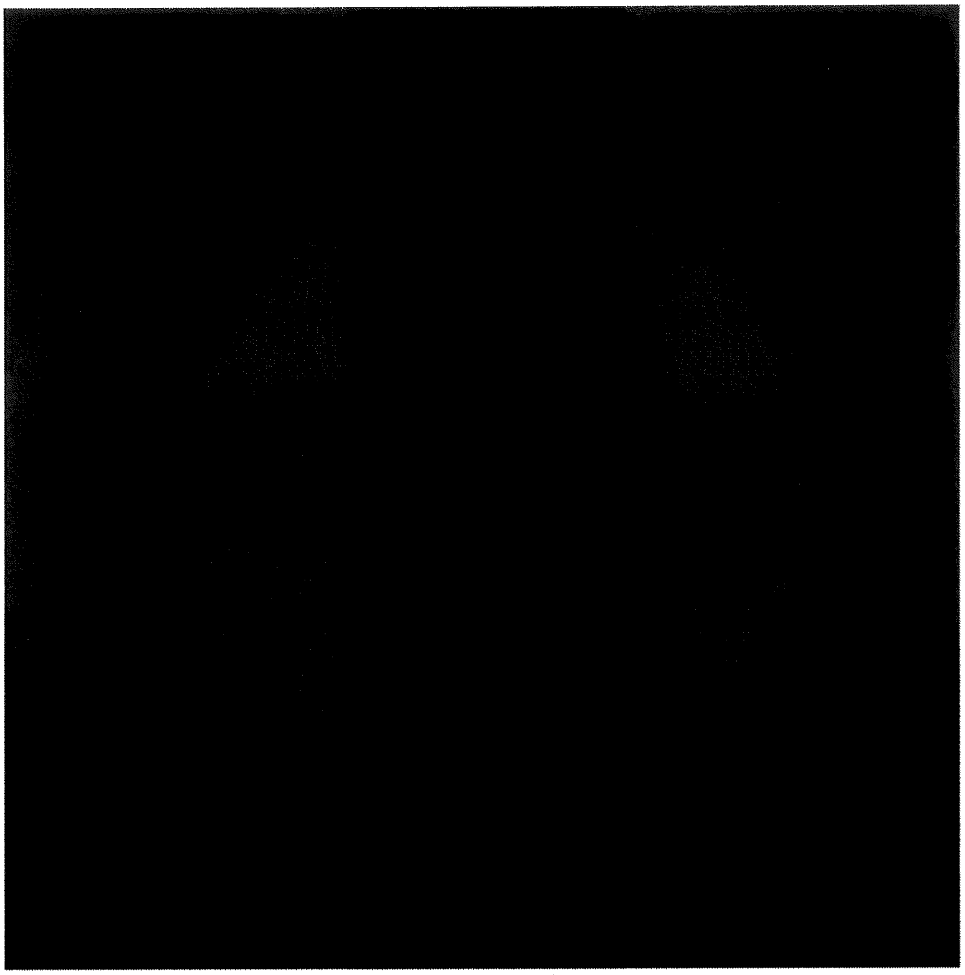


Figure 28: Initial Image with Black Rectangular Centre Section Representing the Blocked Source by the Lead Cylinder

In Figure, there is a black rectangular section in the center of the image where the source has been blocked by the lead cylinder. The intensity of the field is greatest in the center of the image – which corresponds to the center of the detector plane – and gradually fades as the edge is neared – which is furthest away from the source.

5. Engineering Design Process

5.1. Iteration One

The image, displayed in Figure 29, was produced by the first iteration gamma camera under consideration. This image was obtained by normalizing each pixel to the brightest pixel in the image. It is displayed at a magnification factor of 16X and without any background noise.

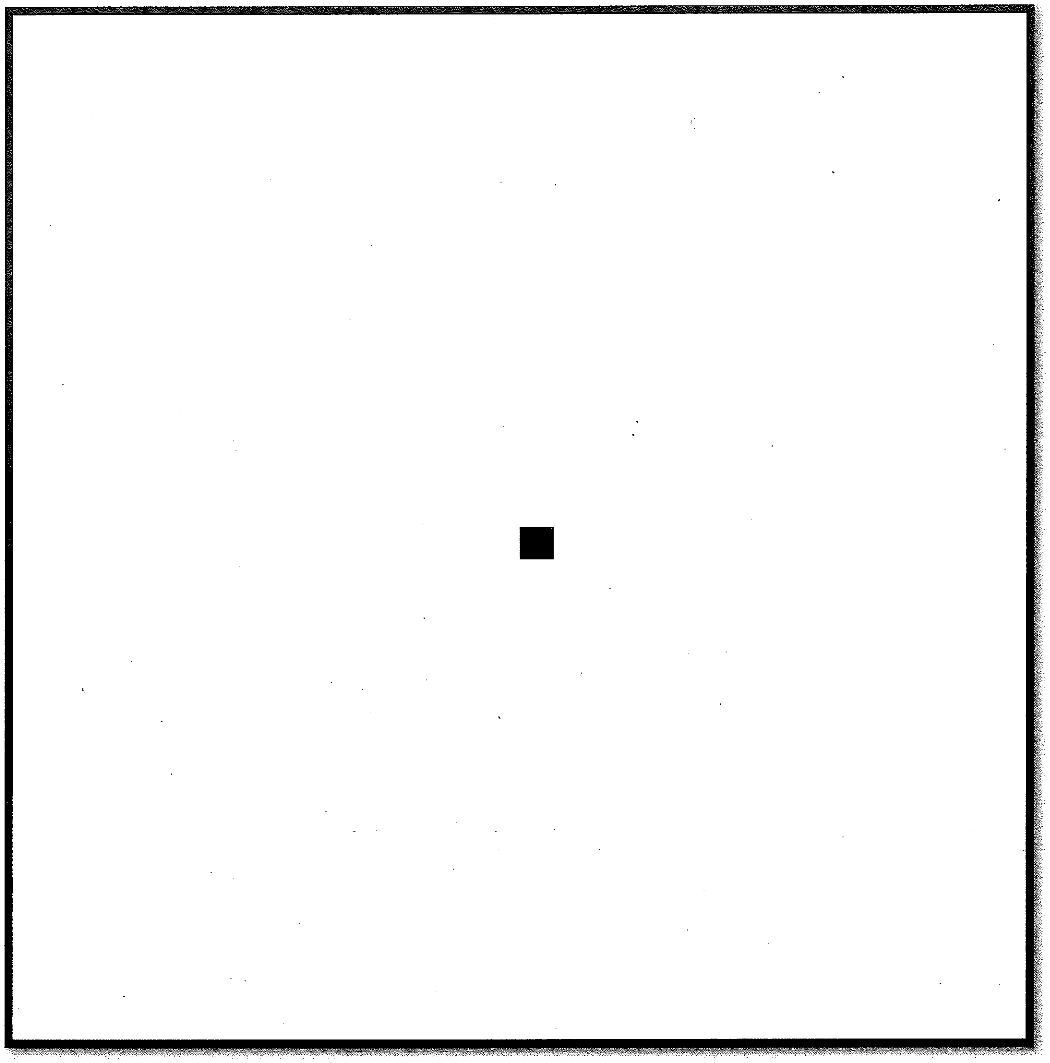


Figure 29: Image Produced by the First Iteration Gamma Camera

The following table summarizes the analysis criteria of the above image.

Table 6: Analysis Criteria Score of First Camera

Criteria	Value
A(32,32,600) - Maximum	8.55
B(32,32,100,7) - Bloom	45
C(32,32) - Dynamic Range	5.63

5.2. Background Noise Studies

Figure 30 shows the effect of background noise on the first iteration camera wherein the background noise ranges from a mean flux of 0 to 270 cm^{-2} s^{-1} in steps of 30.

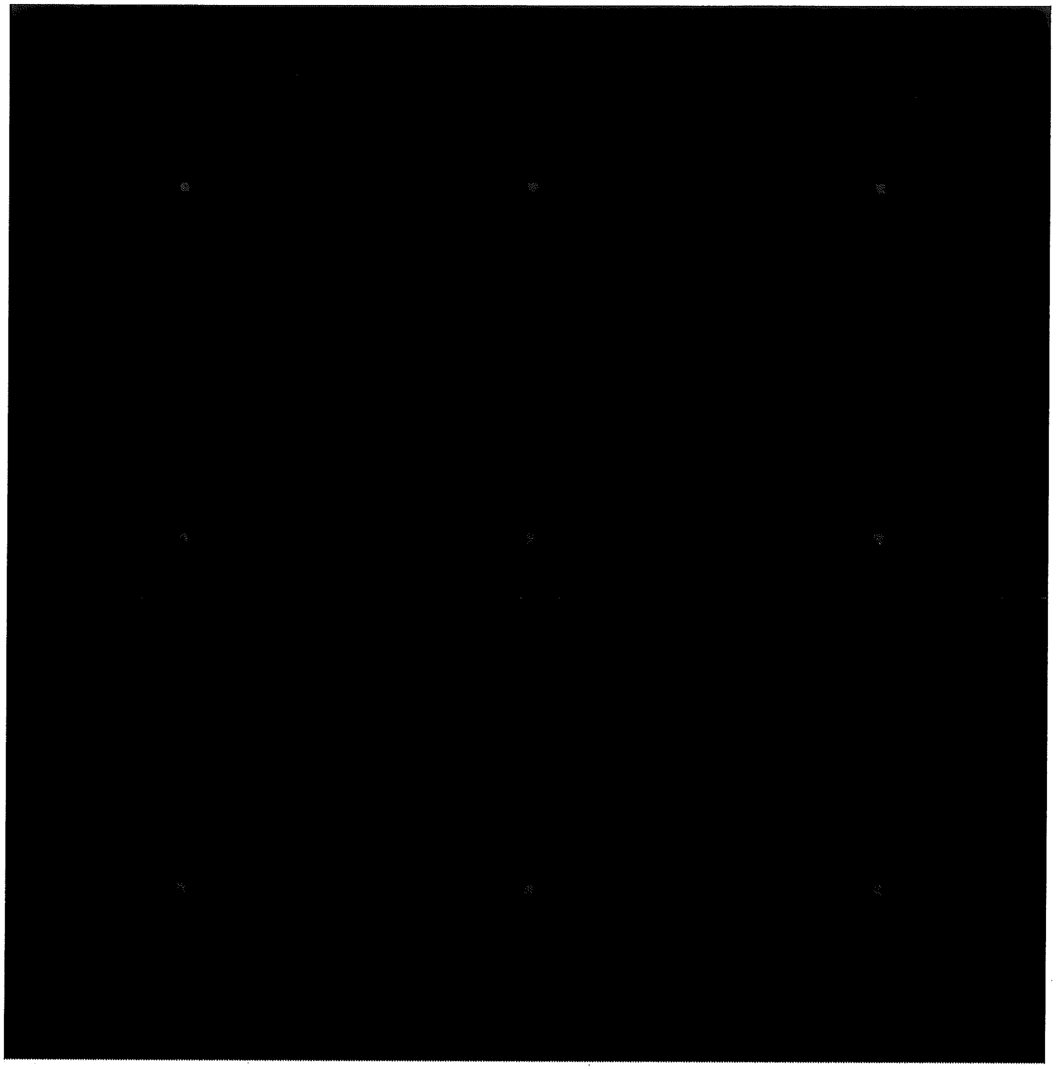


Figure 30: Background Noise Studies

From Top Left to Bottom Right the flux of particles is: 0, 30, 60, 90, 120, 150, 180, 210, and 240, cm^{-2} s^{-1}

Figure 31 shows the effect of the background noise on the image with the mean value of the background noise ranging from 100 to 900 cm^{-2} s^{-1}, in steps of 100 cm^{-2} s^{-1}.

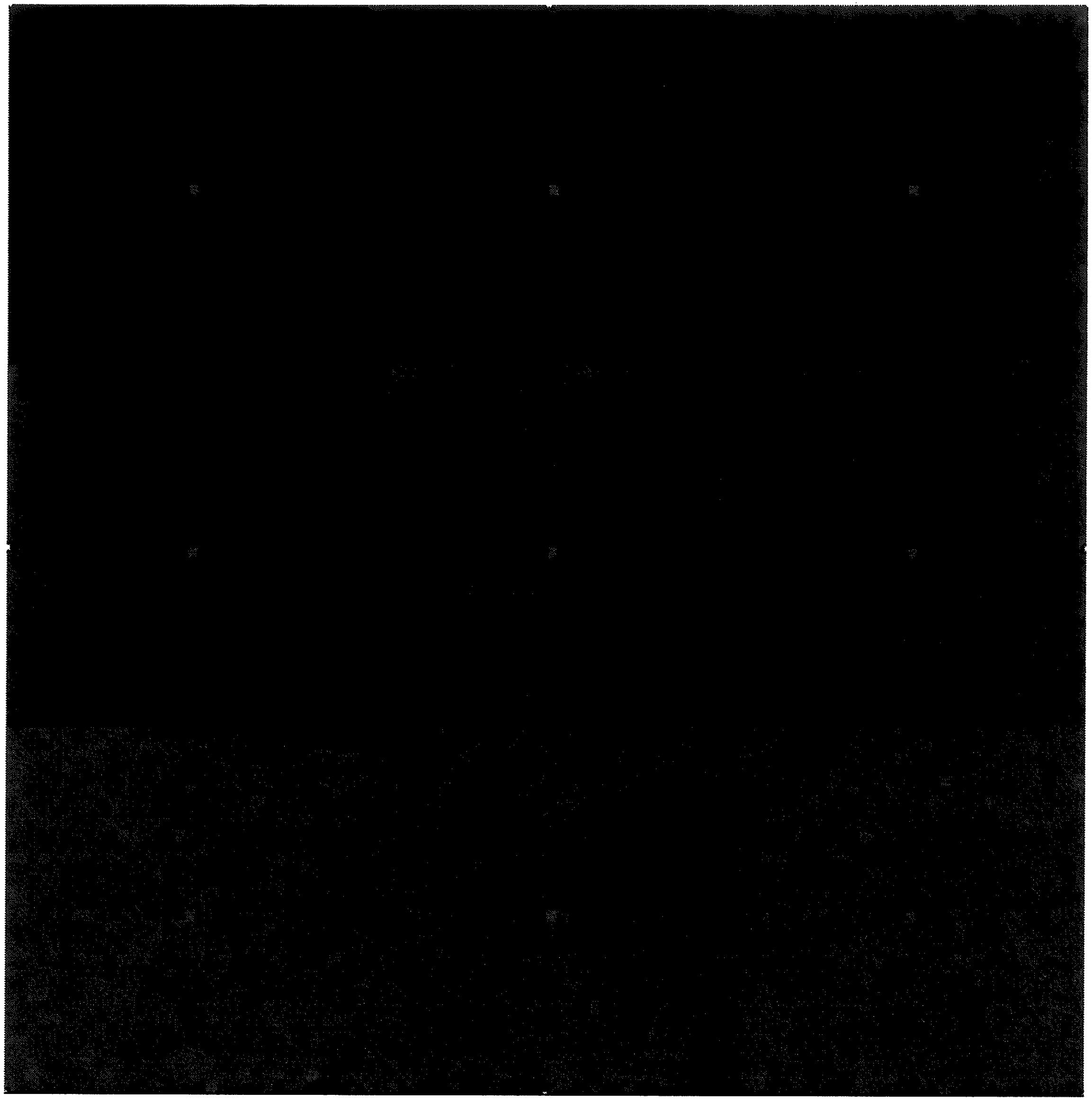


Figure 31: Background Noise Studies. From Top Left to Bottom Right: 100, 200, 300, 400, 500, 600, 700, 800, and 900 cm^{-2} s^{-1}.

It can be seen that by the end of the series, 900 cm^{-2} s^{-1}, the image appears to be quite washed-out.

This investigation is concluded by examining the effects of extremely high levels of background noise. Figure 32 shows background noise in the range of 1000, 2000, and 3000 cm^{-2} s^{-1} on the image produced by the first iteration camera.

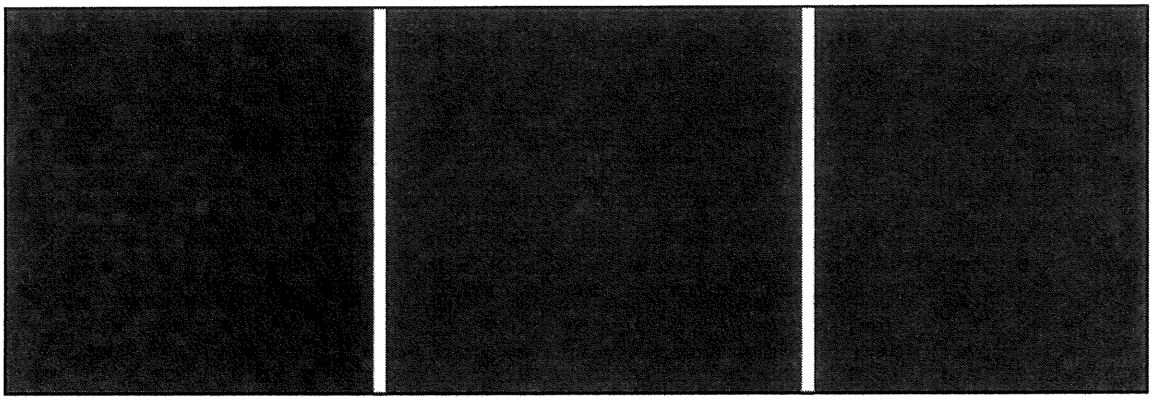


Figure 32: High Background Noise Studies: 1000, 2000, and 3000 cm^{-2} s^{-1}

5.3. Bloom Algorithm

Figure 33 shows the results of the benchmark simulation with the central area to be manually selected.

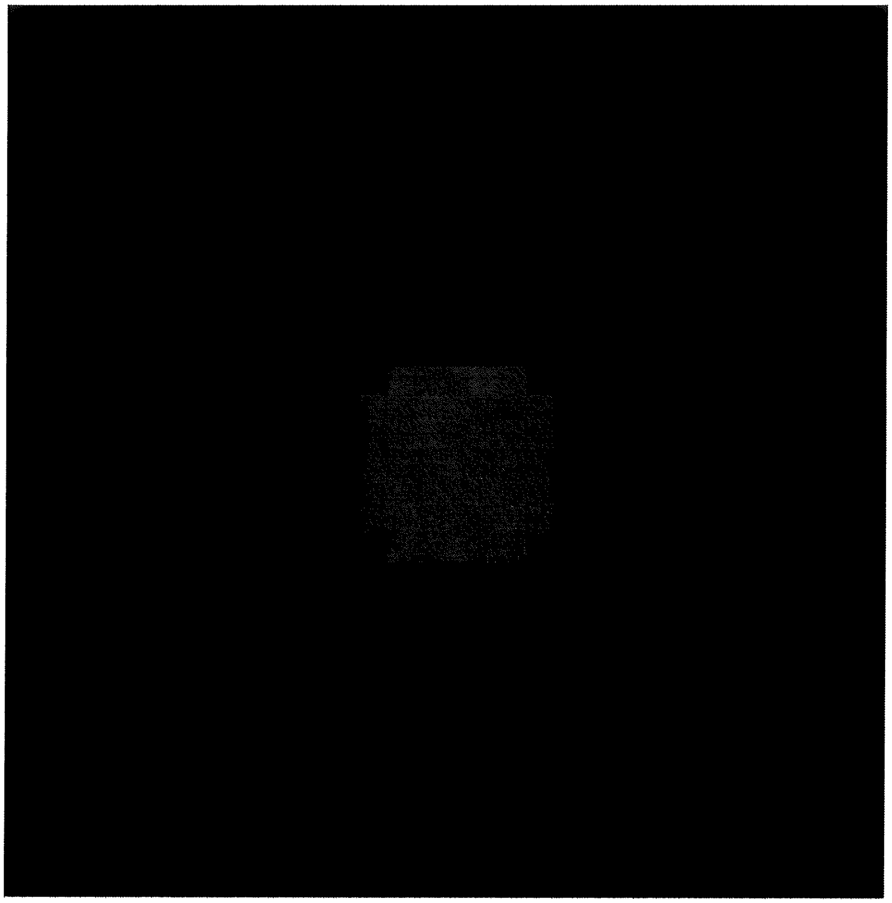


Figure 33: Hand Selected Bloom Area

The area at the center of Figure 33 was found to be 45 pixels. The tolerance factor was parameter-swept from 0.1 to 3.0, in steps of 0.1; the results of this study are illustrated in Figure 34.

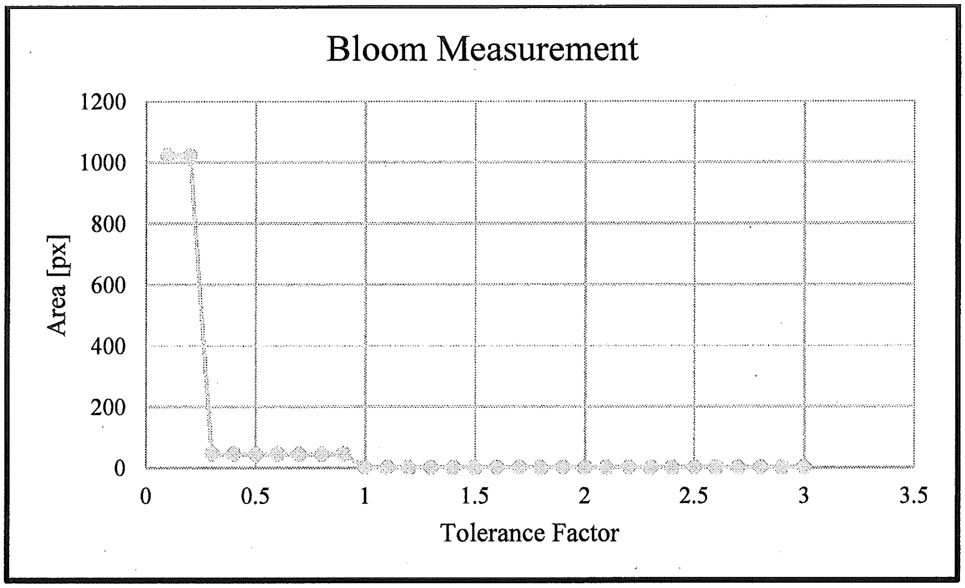


Figure 34: Tolerance Factor Parameter Sweep.

Closer inspection of the data from a parameter value of 0.3 to 1.0 shows the following:

Table 7: Tolerance Factor Scores

Tolerance Factor	Area
0.3	46
0.4	45
0.5	45
0.6	45
0.7	45
0.8	45
0.9	45
1	1

The values corresponding to a tolerance factor between 0.4 and 0.9, which give the same result when the procedure is performed by a human. Taking the mid-point of these tolerance factors, we arrive at a final value of 0.65 for the tolerance factor.

It is possible, however, that given a particular gamma camera configuration, the tolerance factor might change, and it is assumed that the images produced will be similar enough that the same tolerance factor can be used.

5.4. Parametric Studies of Criteria – Presentation and Discussion

In order to provide an *aiming-point* for the second iteration of the gamma camera, the design parameters need to be varied, and the resultant effect on the analysis criteria measured. We performed parametric studies for the following design parameters: pinhole diameter, shield thickness, and the distance between the detector and the shield (detector-shield distance). These studies were performed with background noise enabled, and, as such, there is a variation in their results from point-to-point. The results of these studies are presented below.

5.4.1. Pinhole Diameter

Criteria A - Maximum

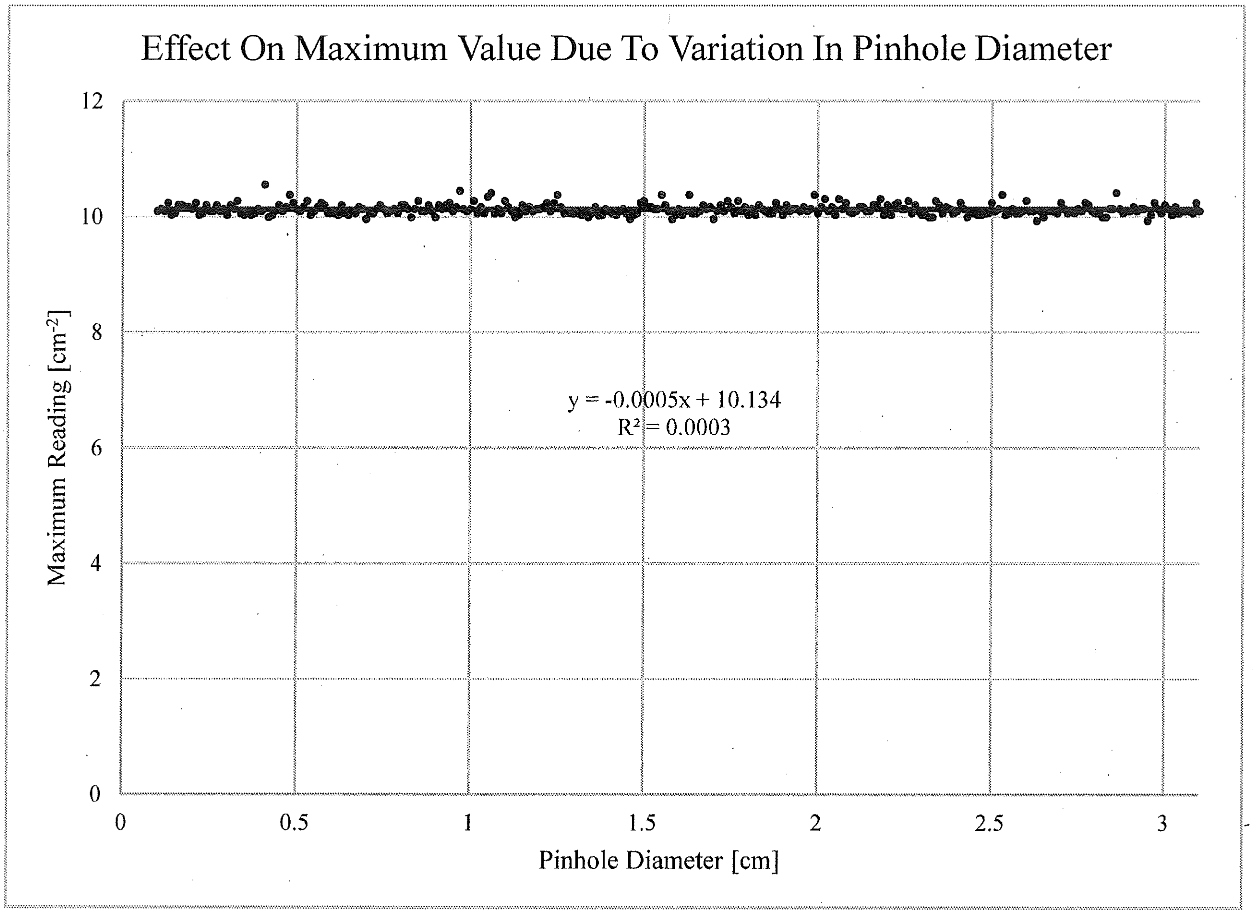


Figure 35: Effect on Maximum Value due to Variation in Pinhole Diameter

Figure 35 shows the effect that variation in pinhole diameter caused to the maximum reading. The range of the variation study is from 0.1 to 3.0 cm, in steps of 0.1 cm. The relationship between maximum reading and pinhole diameter appears to be roughly constant; that is, the maximum reading is unaffected by the variation in pinhole diameter. It is proposed that this is because the maximum reading represents a fully unattenuated beam of radiation, incident upon the detection surface. Varying the pinhole diameter does not change the intensity of the uncollided flux since the pinhole adds no material between the detection plane and the radiation source.

Criteria B - Bloom

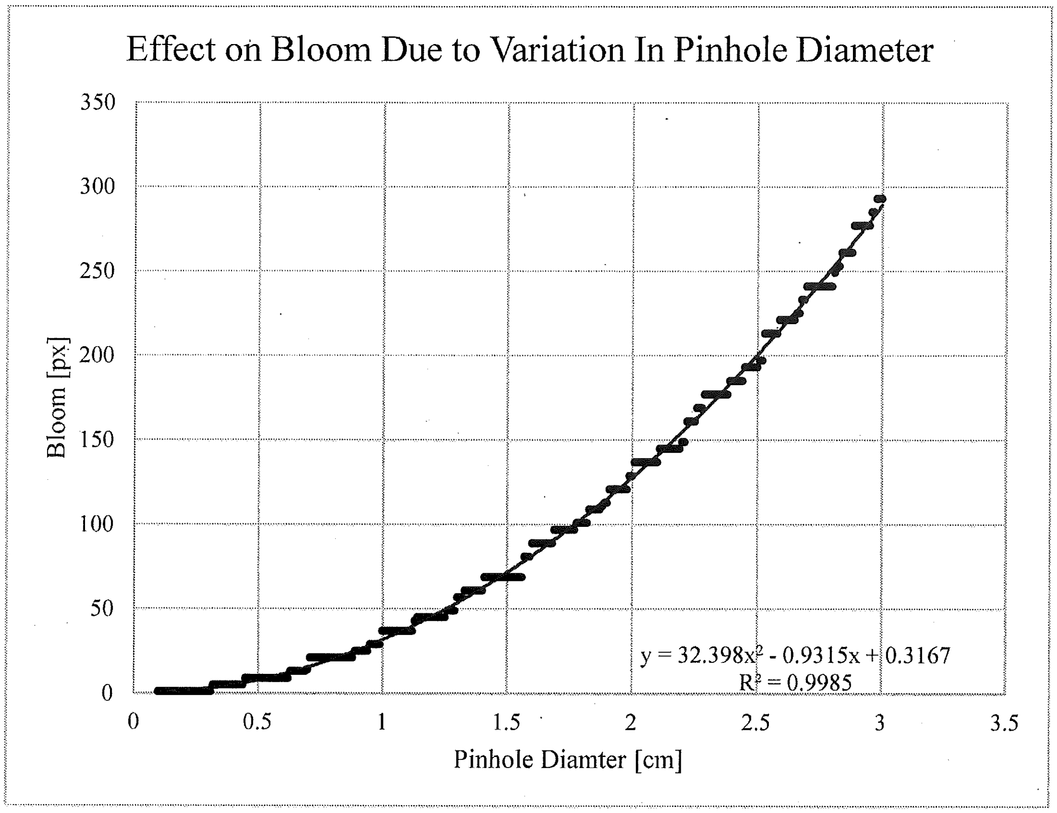


Figure 36: Effect on Bloom Due to Variation in Pinhole Diameter

Figure 36 represents the effect the variation in pinhole diameter has on the bloom analysis criteria. The data shows a quadratic and stepwise-like characteristic to the data. There is a positive relationship between Bloom and pinhole diameter. It is proposed that the stepwise-like appearance of the relationship is caused by the discrete steps in the position of the detector on the detection plane. One can see that there is a gradual, but quick transition from one exposure to the next.

It is proposed that if the resolution of the system is increased, the width of the individual steps would decrease. Furthermore, it is noted that, in general, there is a relationship between square laws and projections of phenomena on planes. In this particular case, for every unit increase in pinhole diameter, the number of pixels affected by the uncollided flux increases in proportion to the square of the pinhole diameter. It is, in effect, describing how an area defined by a characteristic dimension has its area related to it by the square of the said dimension.

Criteria C – Dynamic Range

Figure 37 represents the relationship between pinhole diameter and dynamic range. The study was performed over a range from 0.1 to 3.0 cm, in steps of 0.1 cm.

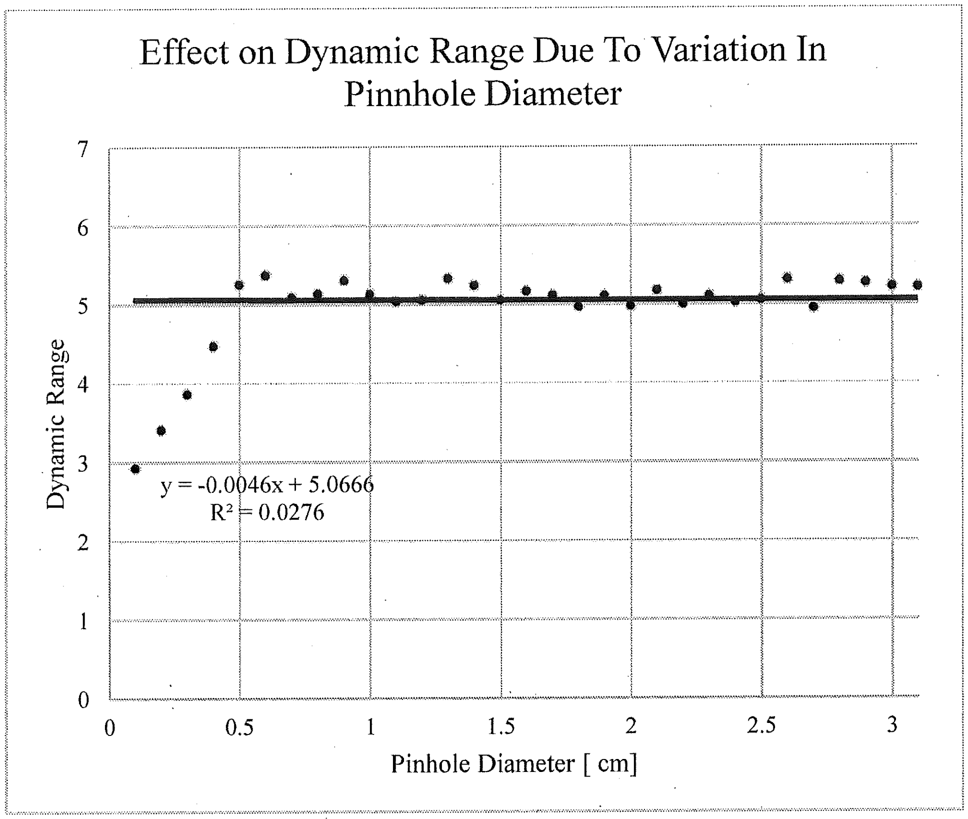


Figure 37: Effect on Dynamic Range Due to Variation in Pinhole Diameter

In the data, up until 0.5 cm, the dynamic range increases with the pinhole diameter. After 0.5 cm in diameter, the relationship between pinhole diameter and dynamic range can be said to be constant.

The initial increase is proposed to be caused by a decrease in the minimum reading given in the image. As shown previously, the maximum reading remains constant with respect to the pinhole diameter. Since the dynamic range is defined to be the ratio of the maximum to minimum readings, and the maximum reading is constant, it follows that the minimum reading is the quantity that is decreasing. Or, more succinctly: the increase in dynamic range is proposed to be caused by a decrease in minimum reading.

5.4.2. Detector Shield Distance

Criteria A – Maximum

Figure 38 details the effect on maximum reading due to variation of the distance between the detector-plane and the shield. The study was conducted over a range of 0.1 to 29.5 cm, in steps of 0.1 cm.

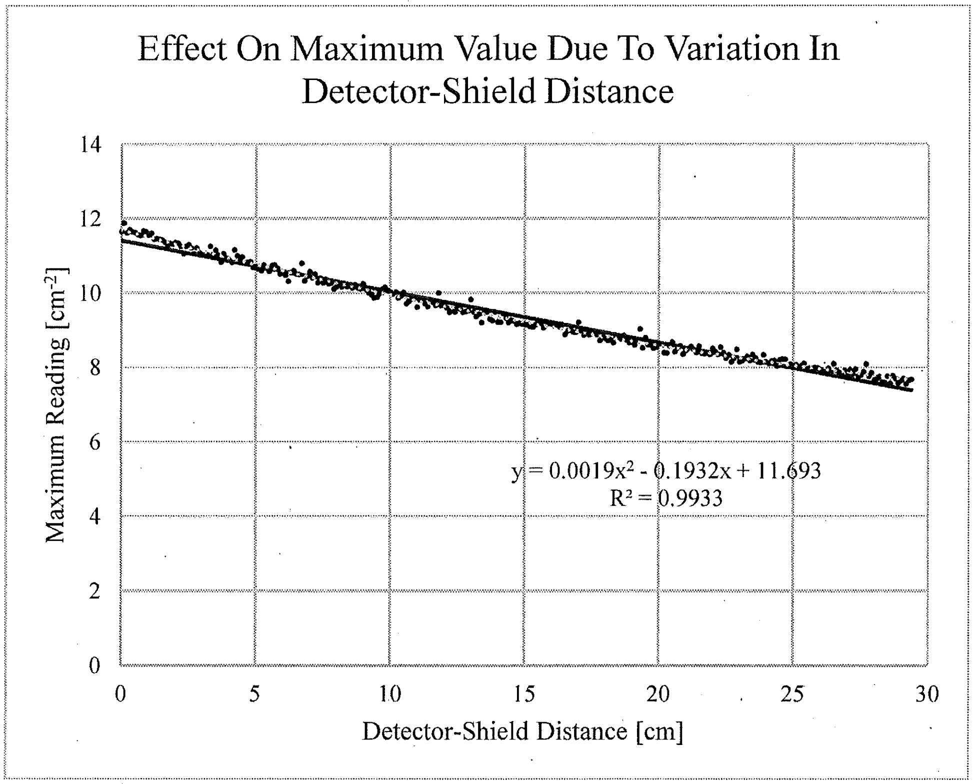


Figure 38: Effect on Maximum Value Due to Variation in Detector-Shield Distance

There appears to be an inverse relationship between the detector shield distance and the maximum reading; that is, as detector-shield distance increases, the maximum reading decreases.

One possible explanation for this could be the decreased maximum reading is due to the increased distance between the detector and the source vis-à-vis the inverse square law. A priori, one would expect an inverse-quadratic nature to the data produced, rather than a line as appears. This, however, is contradictory for two reasons: first, the hand-calculation check showed the same result as that produced by the code; second, the code itself was tested to ensure the inverse square was being calculated.

A final explanation could be that in fact, the data does show an inverse-quadratic relationship; however, the region of data explored is well approximated by a straight line.

Criteria B – Bloom

Figure 39 shows the variation in bloom caused by variation in the distance between the detector and shield. The study was conducted across a range of 0.1 to 30.0 cm, in steps of 0.1 cm. The data appears to have a step-wise and quadratic nature.

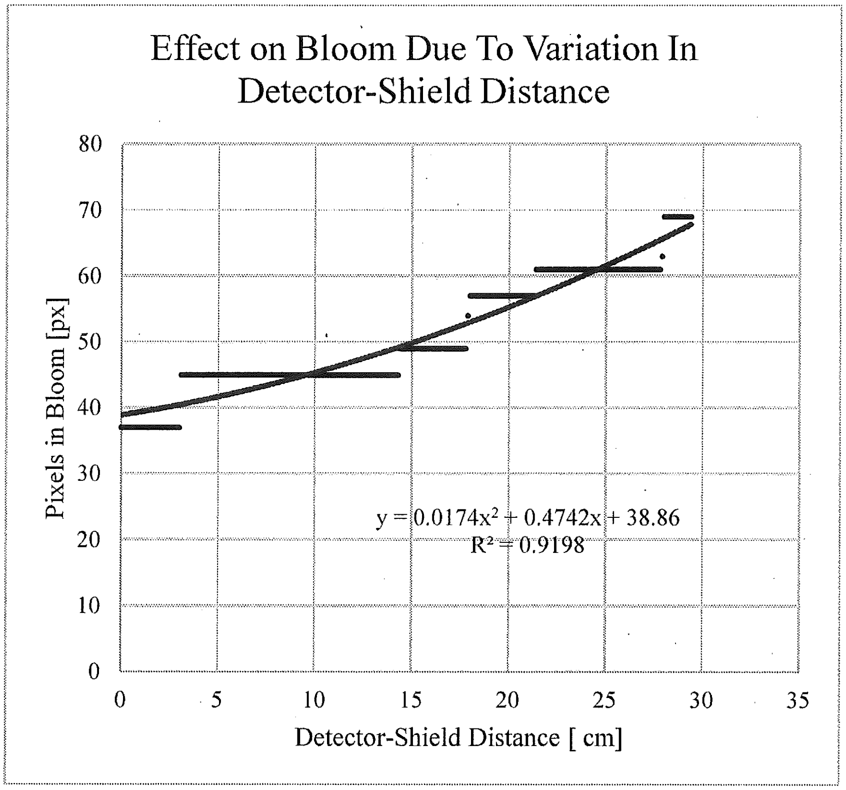


Figure 359: Effect on Bloom Due to Variation in Detector-Shield Distance

There appears to be two levels of bloom that could arise within the design envelope of the proposed system: The first level spans from 0.1 cm to 3.6 cm and has a bloom magnitude of 37; and the second level that spans from 3.7 to 14.8 cm and increases the bloom by a factor of 1.2. It is this second level that is consistent with the minimum detection plane size. That being said, it is foreseeable that a zooming system may wish to decrease the detector-plane distance in order to "zoom in" and reduce the bloom of a source.

It is proposed that the quadratic nature of this curve is caused by the following line of reasoning: bloom is a real quantity that measures the number of pixels in an area. The detector-shield distance on the other hand is a lineal quantity, that measures the distance between the shield and detector. The relationship between the characteristic distance and area is typified by a square relation. Thus, since bloom is a real quantity, and detector-shield distance is a lineal quantity, it follows that any relationship between them should be a square relationship.

It is further proposed that the step-wise nature of the data is explained as follows: the measurements were taken at discrete positions. As the distance between the shield and detector increases, the projected area of the spot also increases; however, this increase is not registered by the system because the increased area still resides in the area covered by the same pixels that register the smaller sized spot. As the spot size continues to increase, there finally reaches a point where the spot size *bleeds* over into adjacent pixels. This causes a discrete jump in bloom size, and thereby the stepwise nature of the data.

It is interesting to note that the step-wise nature of the data appears unperturbed by variation due to background noise. This lends credence to the robustness of the bloom-finding algorithm in that it is able to distinguish the spot regardless of influence from the background noise. If this were not the case, one would expect local variation in each of the steps.

Criteria C – Dynamic Range

Figure 40 shows the results of a variational study between detector-plane distance and dynamic range. The study was conducted over a range of 0.1 to 29.5 cm. Although the line-of-best-fit has a slightly negative slope, for all intent and purpose, the relationship between the dynamic range and detector-plane distance is roughly constant, at slightly above five.

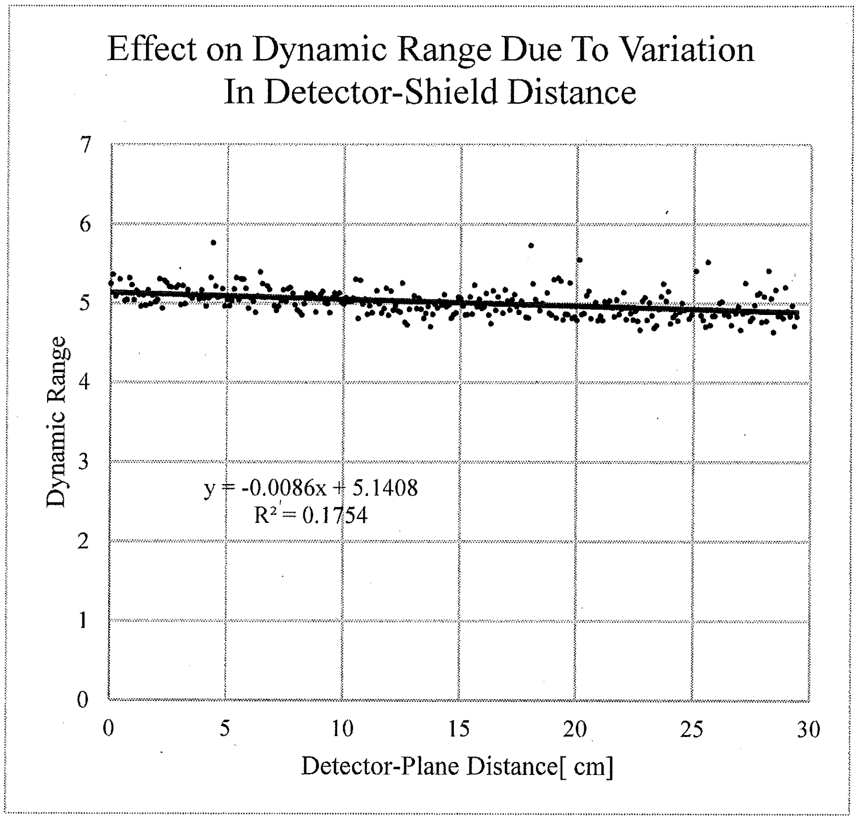


Figure 40: Effect on Dynamic Range Due to Variation in Detector-Shield Distance

The dynamic range, is again, defined to be the ratio of the maximum to minimum reading. The maximum, from the first part of these variational studies on detector-plane distance, decreases with increasing detector-plane distance, yet, the ratio remains roughly constant. This should imply that the minimum is increasing with increasing distance to compensate for this. Interestingly enough, this phenomenon parallels that which was found in the dynamic-range result from the pinhole-diameter study.

5.4.3. Shield Thickness

Criteria A – Maximum

Figure 41 shows the relationship between maximum reading and shielding thickness. The study was performed over a shielding thickness range of 0.1 to 10.0 cm, at steps of 0.01 cm. The data shows the maximum reading is unaffected by the shielding thickness in that the data is fit by a straight flat line.

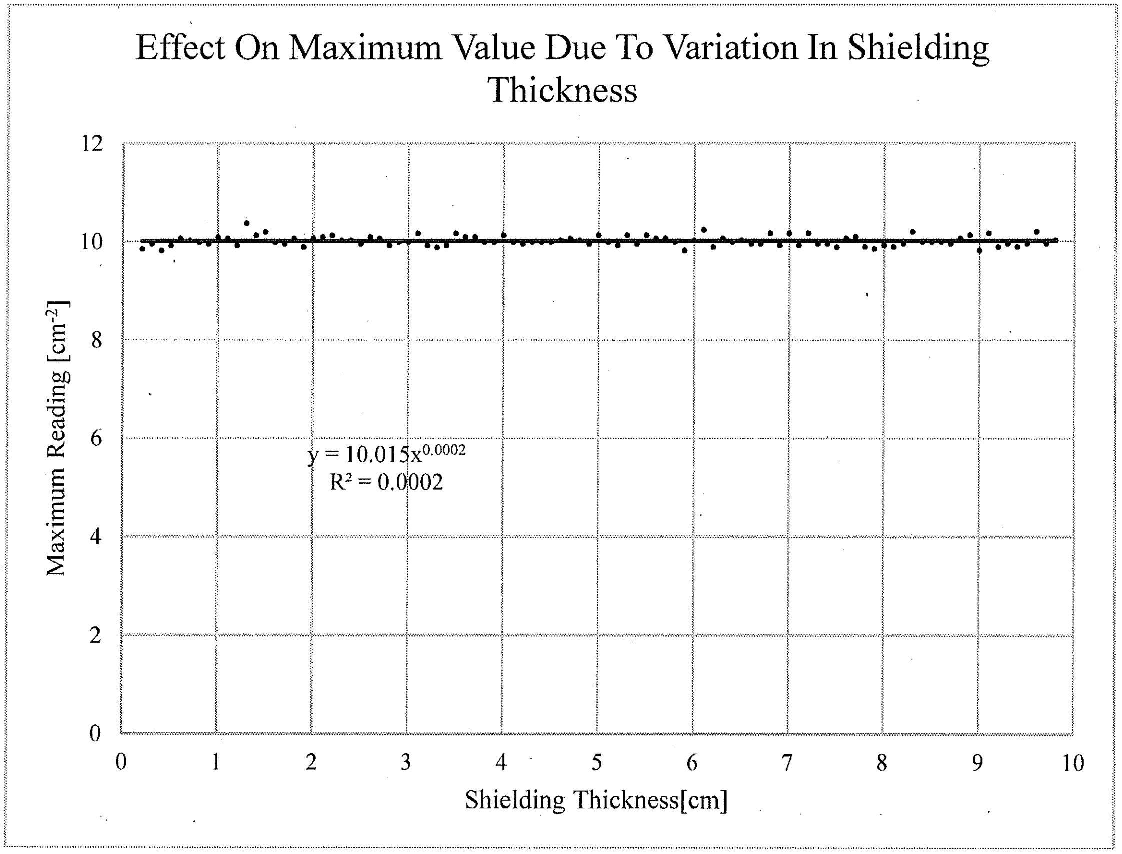


Figure 41: Effect on Maximum Value Due to Variation in Shielding Thickness

It is proposed this relationship is due to the fact that the maximum reading on the detector plane corresponds to a region of unattenuated flux i.e. the flux that passes through the central pinhole and is unaffected by the shielding.

Criteria B – Bloom

Figure 42 shows the relationship between bloom and dynamic shielding thickness. The study was performed in a range of 0.1 to 10.0 cm, in steps of 0.01 cm. The relationship between bloom and shielding thickness appears to be an *anti-threshold* function; that is, above 0.28 cm, the bloom is significantly reduced.

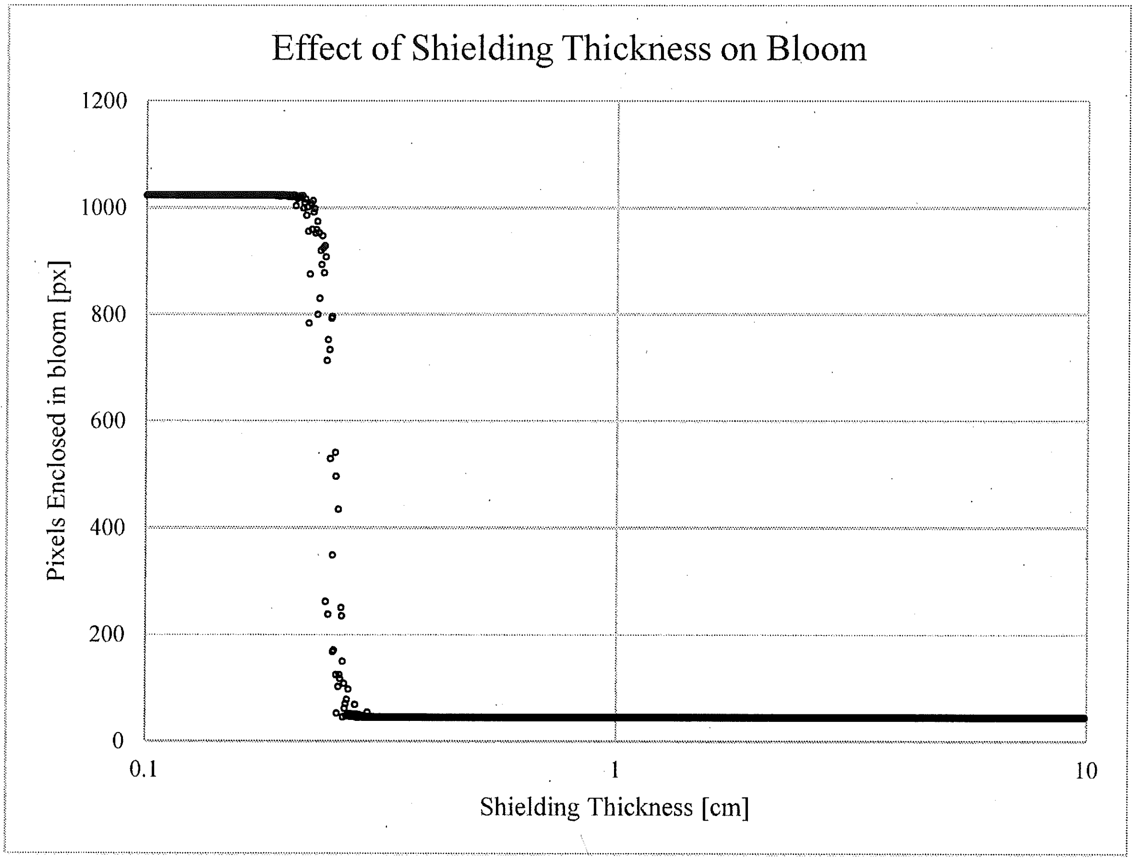


Figure 42 Effect of Shielding Thickness on Bloom in Log Scale

It is hypothesized that this is due to the shield becoming thick enough to attenuate a significant enough fraction of the incoming radiation. At a 'critical thickness', a clearly delineated region of exposure and non-exposure becomes visible. Before the critical thickness, the bloom-finding algorithm is unable to distinguish between the spot produced and the background. After the critical thickness has been reahed, there is a clear distinction.

Finally, the data also show the bloom-finding algorithm is robust across many different simulations, and the tolerance factor is selected correctly, for if it were not, discontinuous jumps would appear in the data indicating the bloom-finding algorithm had chosen two significantly different sets of data in each case.

Criteria C – Dynamic Range

Figure 43 shows the effect of shielding thickness on dynamic range. The study is conducted for a shielding thickness between 0.1 and 10.0 cm, in steps of 0.001 cm.

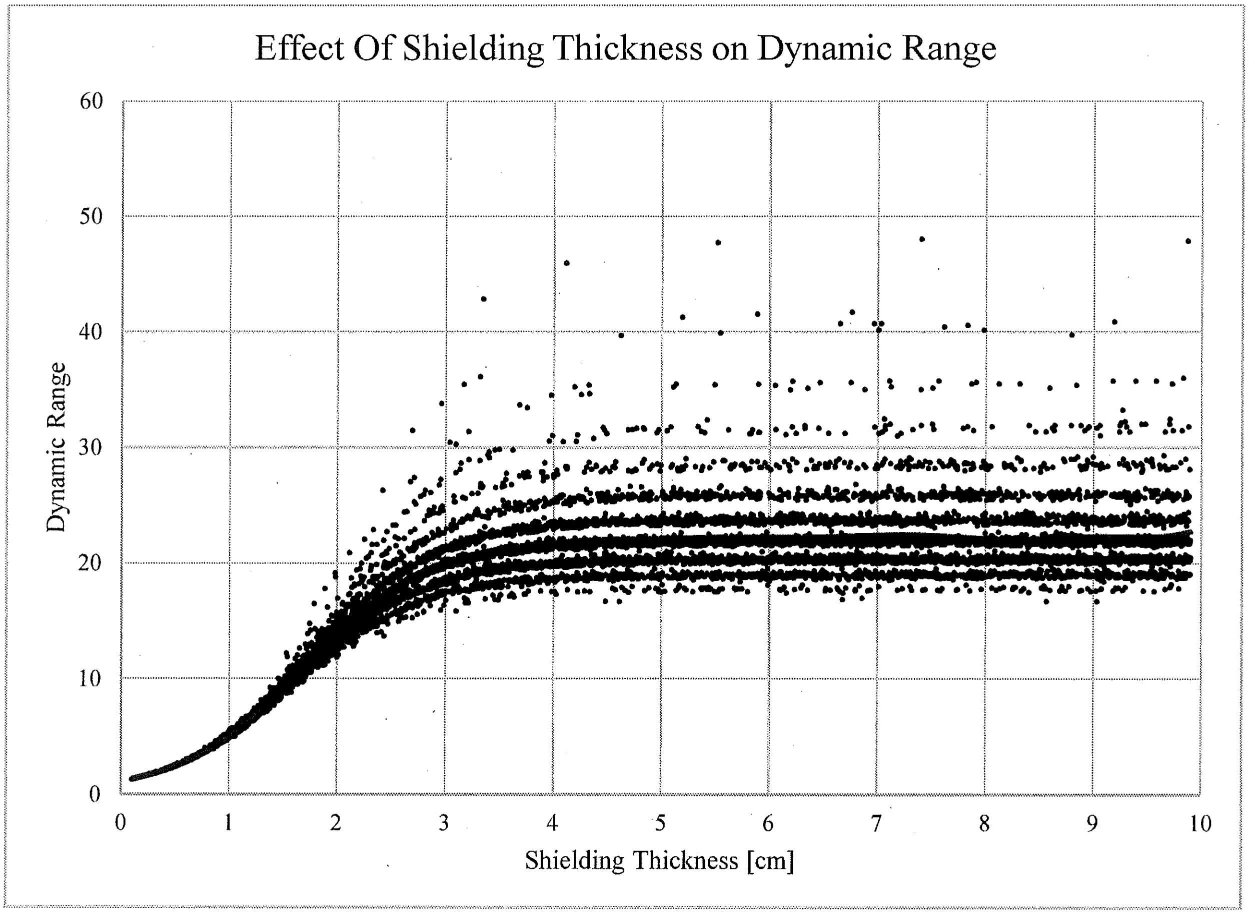


Figure 43: Effect on Dynamic Range Due to Variation in Shielding Thickness

There appears to be a saturation relationship between shielding thickness and dynamic range. The data can be fit by a Fourier series:

$$f(x) = A_0 + \sum_{i=1}^{8} (A_i \sin(i * x * w) + B_i \cos(i * x * w) \tag{42}$$

where the values of A and B and w are provided in Table 8.

Table 8: Fourier Series Values

Index	A	B
0	15.29	-
1	-11.57	2.018
2	-5.584	3.523
3	0.3195	2.64
4	2.492	-0.2081
5	1.039	-1.649
6	-0.4139	-0.9018
7	-0.4676	0.1113
8	0.00066	0.1887
w	0.4325	0.4325

In the fitting function, the dynamic range grows by 3.7 in the first centimeter of shielding thickness, 8.8 in the second, 6.2 in the third, 1.7 in the fourth, and 0.35 in the fifth. For design considerations, the function reaches 90% of its maximum tested value within 3 cm of shielding thickness (effective saturation).

There is a notable banding structure to the noise. It is proposed this is due to the interplay of the Poisson noise occurring at the relatively low camera resolution being tested (32x32 pixels or 1024 total pixels). The minimum pixel value, vis-à-vis the Poisson background noise generated, is proposed to be highly variable, and as such, the dynamic range is highly variable.

It is foreseeable that increasing the resolution – total number of pixels in an image – should even out some of the noise by the action of the law of large numbers. That is, if the sample size is increased, the probability that a consistent minimum value will appear in the data-set will be likewise increased.

It is further proposed that the distinct banding structure is due to the discrete nature of the Poisson noise. It is argued that a *noiseless* sample-set is immaterial to the further design of the camera system. So as long as the general relationship between variables and criteria are known, a second design target can be targeted.

It is interesting to note that this behavior represents a sensitive dependence of dynamic range and background noise.

Nevertheless, the direction of the relationship is clear regardless of the banding structure and noise, and it is the opinion of the author that further refinement of the exact relationship between dynamic range, shielding thickness, and noise is inconsequential to the further development of the gamma camera.

5.5. Summary of Findings from Parametric Studies

Table 9 summarizes the findings of the parametric studies:

Table 9: Design Parameter Variations

Varied Quantity	Measured Criteria	Governing Relationship
Pinhole Diameter		
	A – Maximum	Constant
	B – Bloom	Increases with diameter
	C - Dynamic Range	Increases up to a diameter of 0.5 cm, then is constant
Detector-Plane Distance		
	A – Maximum	Decreases with increasing distance
	B – Bloom	Stepwise and increasing. Between 0.1-3.6 cm is best, followed by 3.7-14.8 cm.
	C - Dynamic Range	Slightly decreasing with distance, but relatively constant
Shield Thickness		
	A – Maximum	Constant
	B – Bloom	Anti-threshold – critical value of 0.28 cm
	C - Dynamic Range	Saturation. 90% at 3 cm

5.6. Aiming the Design Target for the Second Iteration

Let us examine how the results of the parametric studies affect the design-space with some visualizations. In the following visualizations, a green dashed line represents a criterion whose measure is unchanged by the variation, a red dotted arrow represents a parameter that is undesirably changing in the direction of the parameter, and a blue solid arrow represents a parameter that is desirably changing in the direction of the arrow. As a reminder to the reader, Criteria A is the maximum, Criteria B is the bloom, and Criteria C is the dynamic range. Let us begin with the pinhole diameter.

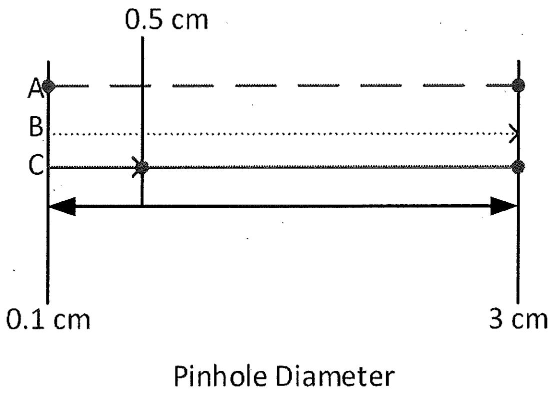


Figure 44: Pinhole Diameter Design Space

In Figure 44, one can see that the maximum is unaffected by change in the pinhole diameter. The bloom increases with increasing pinhole diameter, and the dynamic range increases up to a maximum at 0.5 cm. Therefore, the second design iteration will have a pinhole diameter of 0.5 cm.

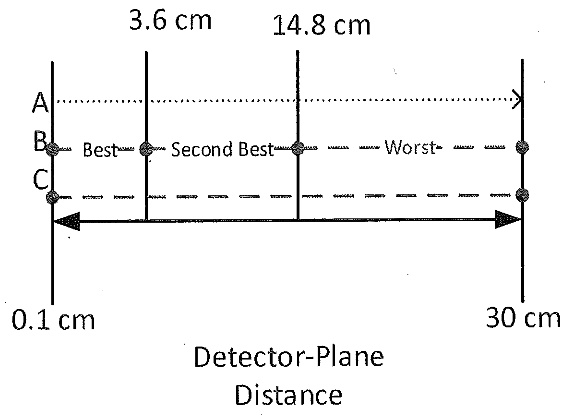


Figure 45: Detector-Plane Distance Design Space

Figure 45 shows that the maximum decreases and the bloom increases with detector-plane distance. The bloom is split into three regions: 0.1 - 3.6 cm; 3.6 - 14.8 cm; and theregion beyond 14.8 cm. Finally, the dynamic range is relatively unaffected by bloom. There is a third, unmentioned engineering factor at play: the detector is of finite size. The detector subtends a definite solid angle, so the notion that the detector-plane distance can be reduced *ad-infinitum* is invalidated – that is, below a certain detector-plane distance, the detection points begin to overlap. To compromise between the parametric results and the engineering constraints, the second iteration shall have a detector-plane distance mid-way between 3.6 cm and 14.8 cm, or 9.2cm.

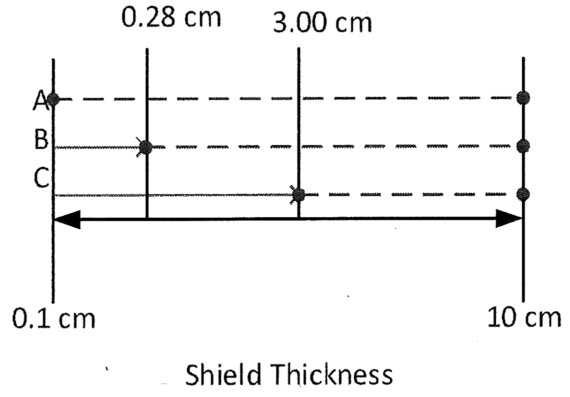


Figure 46: Shield Thickness Design Space

The final parameter under study is shielding thickness. The maximum is unaffected by the shielding thickness. The bloom decreases until 0.28 cm of shielding thickness, and the dynamic range increases up until 3 cm of shielding point, at which point it saturates and its increase due to increased shielding thickness is insignificant. As such, the shielding thickness for the second design iteration shall be set to 3.0 cm.

Finally, since the detector-plane distance has changed to a distance of 9.2 cm, the detector-plane width and height must be recalculated using equation 34 using the following parameters:

Table 10: Detection Plane Equation Parameters

Variable	Description	Value [cm]
D	Pinhole diameter	0.5
G	Detector to shield distance	9.2
T	Shield thickness	3
X	Detector-plane dimension	To be solved For

We arrive at a detector-plane dimension of 3.57 cm.

5.7. The Second Iteration

Based on the previous parametric studies, a simulation of the second iteration of the camera was run with the following parameters:

Table 11: Camera Iteration Two

Variable	Value
Pinhole Diameter	0.5 cm
Detector-Plane Distance to Source	1 m
Detector Plane Distance	9.2cm
Detector-Plane Width and Height	3.57 cm
Shield Thickness	3 cm
Exposure Time	600 s
X-Resolution	32
Y-Resolution	32
Source Strength	1 mCi
Detector Efficiency	0.01

The following results were found:

Table 12 Iteration Two Results

Criteria	Value	% Improvement |original- new| / original
A(32,32,600) - Maximum	8.68	15%
B(32,32,100,7) - Bloom	9	18%
C(32,32) – Dynamic Range	5.63	0%

5.8. Beyond the Second Iteration

To improve the camera beyond the second iteration of the design, we could once again, sweep the parameters of the camera, and find a new design target. However, it is the opinion of the author that such an action would be superfluous and not add significant value to this thesis. The gains made with the second iteration of the design were marginal, and no significant improvements in the design are expected. The purpose is to show that the camera works, not optimize the design.

Conclusion

This thesis aimed to show the conceptualization and parameterization of a rasterizing pinhole gamma camera. As mentioned in the introduction, the objectives of this thesis were to:

1. Explore the idea of creating a rasterizing pinhole gamma camera
2. Determine the feasibility of the rasterizing pinhole gamma camera
3. Determine what the design parameters (e.g. pinhole diameter) of the rasterizing pinhole gamma camera should be.

This thesis met the first objective by first exploring the history and background of gamma cameras. This exploration set a context for the development of the rasterizing pinhole gamma camera. Furthermore, multiple sets of design parameters were explored that provided a view as to the possible performance and output of the system.

The second objective, determination of system feasibility, was met in two ways: feasibility calculations and simulation. The initial feasibility calculations provided assurance that a system could be built using realistic materials and technologies. The simulations also provided assurance that the system would work by the following chain of reasoning: It is believed that the written software provides a realistic analogous representation of how radiation is distributed in space. Further, it is believed that the system modeled in the written software is a realistic analogue of a prototype to be built; therefore, it follows that a prototype developed with design parameters matching that of the simulated system should perform analogously to the simulated system.

Finally, after the simulation software was written, the design parameters of the system were determined. This was achieved by an iterative engineering design process in which an initial best-guess of the design parameters was made, then the performance of the system, quantified by analysis criteria, was measured with respect to local variance in the different design parameters. Maxima in performance of the design parameters were found, and the process was repeated, but with the maxima design parameters from the first iteration as input. It was found that this second iteration did not produce a notable substantial increase in system performance, and as such, a third iteration was determined to be superfluous. The final design parameters found are shown in the table below:

Table 13: Camera Parameters

Parameter	Value
Pinhole Diameter	0.5 cm
Detector Plane Distance	9.2cm
Detector-Plane Width and Height	3.57 cm
Shield Thickness	3 cm
Exposure Time	600 s
X-Resolution	32
Y-Resolution	32

In total, the efforts of this thesis has aimed to develop the concept of the rasterizing pinhole gamma camera, and determine feasible design parameters.

With that being said, there is plenty of room for future work on this subject. The next immediate step in this project would be to build a working prototype of the camera itself. This would require substantial engineering effort, and would be an excellent cross-field endeavor between mechatronics, electrical, and nuclear engineering. This prototype would provide a feedback pathway for the simulative development of the camera in that measurements could be taken using the prototype, then compared to the results produced by the written software for the prototype development. The software could then be modified to represent more accurately the prototype developed, and an additional design iteration could be added. The modified software would be used to perform an additional variational study, where after design parameter maxima could be found, that a second prototype could be built from. This entire process could then be repeated until the finished device was converged to.

References

[1] A. Perkins, M. Frier, "Nuclear Medicine in Pharmaceutical Research", Taylor and Francis Books, 1999.

[2] Newport, "Photomultiplier Tubes," 11 8 2016. [Online]. Available: https://www.newport.com/f/photomultiplier-tubes.

[3] Gamma Spectacular, "GS-1515," [Online]. Available: http://www.gammaspectacular.com/index.php (Accessed 11 8 2016).

[4] A. Einstien, "The Photoelectric Effect," Annals of Physics, no. 18, p. 132, 1905.

[5] J. Lamarsh, "Introduction to Nuclear Engineering", edit. 3, Publisher Pearson, 2001.

[6] A. Ghoshi, "Introduction to Measurements and Instrumentation", Publisher Phi Learning, 4th edition, 2012.

[7] W. Blahd, "Ben cassen And The Development of The Rectilinear Scanner," Seminars in Nuclear Medicine, vol. 26, no. 3, pp. 165-170, 1996.

[8] B. Brown, R.H Smallwood, D. Barber, P. Lawford, D Hose, "Medical Physics and Biomedical Engineering", CRC Press, 1998.

[9] Microsoft, "Introduction to The C# Lanaguge and the .NET Framework," [Online]. Available: https://msdn.microsoft.com/en-CA/library/z1zx9t92.aspx (Accessed 11/ 8/2016).

[10] Persistence of Vision Raytracer Pty. Ltd "POV-Ray," [Online]. Available: http://www.povray.org/ (Accessed 13 8 2016).

[11] Los Alamos, "MCNP," [Online]. Available: https://mcnp.lanl.gov/ (Accessed 8/13 /2016).

[12] National Institute for Health, "ImageJ,". [Online]. Available: https://imagej.nih.gov/ij/ (Accessed 13/ 8/ 2016)

[13] H. Biersack, L. Freeman, "Clinical Nuclear Medicine", Springer, 2008.

[14] A. Movahed, G. Gnanasegaran, J. Buscombe, M. Hall, "Integrating Cardiology for Nuclear Medicine Physicians: A Guide to Nuclear Medicine Physicians", Springer, 2008.

[15] J. Cremer, "Advances in Imaging and Electron Physics", Academic Press, 2012.

[16] K. Blazek, F. Notaristefani, P. Maly, R. Pani, R. Pellegrini, A. Pergola, F. Scopinaro, A. Soluri, "YAP Multi-Crystal Gamma Camera Prototype," IEEE Transactions on Nuclear Science, vol. 42, no. 5, p. 1474, 1995.

[17] P. Busca, C. Fiorini, A. Marone, M. Occhipinti, R. Peloso, N. Blasi, F. Camera, B. Million, O. Wieland "Study and Experimentation of a High Resolution Gamma Camera Based on Thick CsI(TI) Crystals," Proceeding of the Nuclear Science Symposium and Medical Imaging Conference, Anaheim, California, USA, October 29- November 3, 2012.

[18] "CsI(Tl),CsI(Na), Cesium Iodide Scintillation Material Datasheet", Saint-Gobain Co. available at: www.crystals.saint-Gobain.com.

[19] G. Knoll, Radiation Detection and Measurement, Wiley and Sons, 4th edition 2010.

[20] Radio Corperation of America, “Photomultiplier Tube Manual”, RCA Corporation Publication, 1970.

[21] T. Peterson, L. Furenlid, "SPECT detectors: The Anger Camera and Beyond," Physics in Medicine and Biology, Vol. 56, no. 17, 2011.

[22] S. Guru, Z. He, J. Ferreria, D. Wehe, G. Knoll, "A High Energy Gamma Camera Using a Multiple Hole Collimator and PSPMT," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 353, no. 1-3, pp. 328-333, 1994.

[23] G. Hine, "The Inception of Photoelectric Scintillation Detection Commemorated After Three Decades." The Journal Of Nuclear Medicine Vol. 18, no. 9, 1977.

[24] W. Myers, "The Anger Scintillation Camera Becomes Of Age," Journal of Nuclear Medecine, Vol. 20, no.6 pp. 565-567, 1979.

[25] C. Fiorini, A. Gola, R. Peloso, A. Longoni, P. Lechner, H. Soltau, L. Strüder, L. Ottobrini, C. Martelli, R. Lui, L. Madaschi, and S. Belloli, "The DRAGO gamma camera," Review of Scientific Instruments, Vol. 81, no. 10, 2010.

[26] L. Michael, “Handbook of Radioactivity Analysis”, Academic Press, third edition 2012.

[27] S. Webb, "The Physics of Medical Imaging," Medical Science Series, CRC press, Jan.1, 1988.

[28] R. Rocco, P. Grenier and J. Di, "Performance Characteristics of The Autofluoroscope" IEEE Transactions on Nuclear Science, vol. 15, no. 1, pp.

366-375, 1968.

[29] E. Sveinsdottir, B. Larsen, P. Rommer,and N. Lassen, "A Multidetector Scintillation Camera with 254 Channels" Journal of Nuclear Medecine, vol. 18, no2, pp. 168-174, 1977.

[30] B. Hutton, "The Origins of SPECT and SPECT/CT," European Journal of Nuclear Medecine, vol. 41, no. 1, 2014.

[31] M. Short, "Gamma Camera Systems," Nuclear Instruments and Methods in Physics Research A, Vol. 221, issue 1, pp. 142-149, 1984.

[32] T. Koike, S. Uno, T. Uchida M. Sekimoto T. Murakami, M. Shoji, F. Nagashima, K. Y. Nakanoe, "A New Gamma Camera With a Gas Electron Multiplier," Journal of Instrumentation, Vol.7, 2011.

[33] W. Stason, “Cardiac Radionuclide Imaging and Cost Effectiveness”, Report of the Congress of The United States Office of Technology Assesment, 1982.

[34] A. Cerry, Practical Radiotherapy: Physics and Equipment, Cambridge University Press, second edition, 1998.

[35] J. Vince, “Vector Analysis for Computer Graphics”, Springer, first edition, 2007.

[36] National Institute of Science and Technology, "Poisson Process - Engineering Statistics Handbook," [Online]. Available: http://www.itl.nist.gov/div898/handbook/eda/section3/eda366j.htm. (Accessed 16/ 8/2016).

[37] National Institute of Science and Technology, "Photon Attenuation Coeffecients for Air," [Online]. Available: http://physics.nist.gov/PhysRefData/XrayMassCoef/ComTab/air.html(Accessed 16/8/2016).

[38] National Institute of Science and Technology, "X-Ray Mass attenuation Coeffecients," [Online]. Available: http://physics.nist.gov/PhysRefData/XrayMassCoef/ComTab/air.html (Accessed 11/8/2016].

[39] Wolfram Alpha, "Query for "Density of Dry Air"," [Online]. Available: http://www.wolframalpha.com/input (Accessed 11/8/2016).

[40] S. Cherry, J. Sorenson, M. Phelps, “Physics in Nuclear Medicine”, Elsevier,fourth edition, 2012.

[41] Non-Destructive Testing Resource Center, "Half-Value Layers," [Online]. Available: https://www.nde-ed.org/EducationResources/CommunityCollege/Radiography/Physics/HalfValueLayer.htm (Accessed 16/7/2016].

[42] LIND Incorporated, "712 Datasheet," [Online]. Available: http://www.lndinc.com/products/pdf/711/ (Accessed 16/ 7/2016).

[43] A. Mazzolo, "Cauchy's formulas for random walks in bounded domains," Journal of Math and Physics, vol. 55, no. 8, 2014.

[44] L. Mordfin, Handbook of Reference Data for Nondestructive Testing, ASTM, second edition, 2002.

[45] Imagesco, "Geiger Counters," [Online]. Available: http://www.imagesco.com/geiger/geiger-counter-kits.html (Accessed 18/08/ 2016).

[46] "Math Dot Net Numerics," [Online]. Available: http://numerics.mathdotnet.com/. (Accessed 18/10/2015).

[47] W. Blahd, "Ben Cassen and the Development of the Rectilinear Scanner," Seminars in Nuclear Medecine, vol. 26, no. 3, 1996.

[48] Canadian Nuclear Safety Comission, "Natural Background Radiation Fact Sheet," [Online at: www.nuclearsafety.gc.ca], (Accessed 25/1/2016).

[49] J. Bushberg, The Essential Physics of Medical Imaging, Wiley, third edition, 2002.

[50] Wolfram Alpha, "Query on "U238 Half Life"," [Online]. Available: http://www.wolframalpha.com/input/ (Accessed 14/8/2016).

Appendix D: Software main functions

PROGRAM.CS

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MathNet.Numerics.LinearAlgebra;
using Excel;
using System.Data;
using ClosedXML.Excel;


namespace Project_Charmander
{
    class Program
    {
        static void Main(string[] args)
        {

            //ITTERATATION 1
            /*float pinholeDiameter = 1.2f;
            float detectorPlaneDistanceToSource = 100.0f;
            float detectorPlaneDimension = 11.0F;
            float shieldThickness = 1.0f;
            float detectorPlaneDistance = 10.0f;
            int xRes = 32;
            int yRes = 32;
    float exposureTime = 3600.0f/(float)(xRes*yRes); //in seconds.*/



    //detector parameters.
     //ITTERATION 2
    float pinholeDiameter = 0.5f;
    float detectorPlaneDistanceToSource = 100.0f;
    //float detectorPlaneDimension = 3.57f;
    float detectorPlaneDimension = 11.0f;
    float shieldThickness = 1.0f;
    float detectorPlaneDistance = 9.2f;
    int xRes = 32;
    int yRes = 32;
    float exposureTime = 3600.0f/(float)(xRes*yRes); //in seconds.*/

    string studyName = "Shield Thickness";
    float paraMin = 30.0f;
    float paraMax = 270.0f;
    float paraStep = 30.0f;

    //float para = 0.03f;

    List<Tuple<float,criteriaSet>> paraCriteriaList = new
List<Tuple<float,criteriaSet>>();
    List<Tuple<float, result[]>> paraResultList = new List<Tuple<float,
result[]>>();
    int ittr = 0;
    //for(float para =paraMin; para <= paraMax; para += paraStep)
    //{
      //Console.WriteLine("{0} Para @ {1:0.000}",ittr, para);
```

```
        Vector<float> spPos = Vector<float>.Build.DenseOfArray(new float[]
{ detectorPlaneDistanceToSource, 0.0f, 0.0f });
        Infrastructure infra = Setups.pinhole1(shieldThickness,
detectorPlaneDistance, detectorPlaneDimension, detectorPlaneDimension,
pinholeDiameter, spPos, xRes,yRes);



        infra.drawables.AddRange(infra.surfaceList);
        infra.drawables.Add(infra.sp);
        infra.drawables.AddRange(infra.dp.detectionPoints.Select(u => u.p));

        infra.dp.detectEffeciency = 0.01f;
        infra.dp.time = exposureTime;
        //1 mCi
        infra.dp.sourceStrenght = 3.7f * (float)Math.Pow(10.0f,7); //bq
        //TODO: implement this as parameter.

        //TEST RUN HERE!
        result[] resultsA = DetectorPlane.runTest(infra, infra.dp,false);
        //paraResultList.Add(new Tuple<float, result[]>(para, resultsA));

        PovRayWritter.WritePOVRayFile("cam1.txt", infra.drawables.ToArray());
        //why isn't this string formater working?
        //  string fileName = String.Format("\\images\\{0} image
{1:0.000}.bmp",ittr,para);
        string fileName = "\\Image Out.bmp";
        DetectorPlane.drawResults(resultsA.ToArray(),
Environment.CurrentDirectory + fileName, true);

        //find max, then find average.
        //the delta between max and average will be the tolerance.c
        float resultMax = resultsA.Max(u => u.value);
        float resultAvg = resultsA.Average(u => u.value);
        float ratio = resultMax / resultAvg;
  //I'll just parasweep the multiplier of B to find out what the multiplier should be.

        float A = criteriaA(resultsA, 1);
        float B = criteriaB(resultsA, 0.65f*resultMax);
        float C = criteriaC(resultsA);

        Console.WriteLine("A:{0}\nB:{1}\nC:{2}", A, B, C);

        //criteriaSet cs = new criteriaSet{critA = A, critB = B, critC = C};
        //Tuple<float, criteriaSet> data = new Tuple<float, criteriaSet>(para, cs);
        //paraCriteriaList.Add(data);
        ittr++;
      //}

      //writeParametricStudyToFile(paraCriteriaList, studyName);
      //writeParametricResultSetToFile(paraResultList, studyName);
        Console.WriteLine("Done....Press any key to exit.");
        Console.ReadKey();
    }

    static void writeParametricResultSetToFile(List<Tuple<float,result[]>> data,
string description)
    {
      var wb = new XLWorkbook();
      //make the first sheet the description.
      var wsCover = wb.AddWorksheet("Cover Page");
      wsCover.Cell(1, 1).Value = description;
```

```
        foreach(var x in data)
        {
          var ws = wb.AddWorksheet(String.Format("{0:0.00}", x.Item1));
          for(int i = 0; i <= x.Item2.Max(u=>u.x); i++)
          {
            for(int j = 0; j <= x.Item2.Max(u=>u.y); j++)
            {
              ws.Cell(i + 1, j + 1).Value = x.Item2.Single(u=> u.x==i && u.y == j).value;
            }
            //maybe implement accessors on the result class so they can be accesed by [] and .getLength(i)
          }
        }

        wb.SaveAs("paraResults.xlsx");


    }

    static void writeParametricStudyToFile(List<Tuple<float,criteriaSet>> data, string description)
    {
        var wb = new XLWorkbook();
        var ws = wb.Worksheets.Add("Data Sheet");
        //ws.Cell(1, 1).Value = "Hello Worldz!";

        ws.Cell(1, 1).Value = description;
        ws.Cell(2, 1).Value = "Parameter Value";
        ws.Cell(2, 2).Value = "Criteria A";
        ws.Cell(2, 3).Value = "Criteria B";
        ws.Cell(2, 4).Value = "Criteria C";
```

```
      for (int i = 0; i < data.Count; i++ )
      {
        ws.Cell(i + 3,1).Value = data[i].Item1;
        ws.Cell(i + 3,2).Value = data[i].Item2.critA;
        ws.Cell(i + 3,3).Value = data[i].Item2.critB;
        ws.Cell(i + 3,4).Value = data[i].Item2.critC;
      }

        wb.SaveAs("paraCrit.xlsx");
    }



    static bool resultSetsSame(result[] a, result[] b)
    {
      bool setsSame = true;
      result[] blobSorted = a.OrderByDescending(u => u.x).ThenBy(u =>
u.y).ToArray();
      result[] oldBloblSorted = b.OrderByDescending(u => u.x).ThenBy(u =>
u.y).ToArray();

      if(blobSorted.Count() != oldBloblSorted.Count())
      {
        setsSame = false;
      }
      else
      {
      //count starts at one.
        for (int i = 0; i < blobSorted.Count(); i++ )
        {
```

```
        if(blobSorted[i] != oldBloblSorted[i])
        {
          setsSame = false;
        }
    }
  }

  return setsSame;
}


static result[] removeBelowTolerance(result[] r, float tolerance)
{
  return r.Where(u => u.value > tolerance).ToArray();


}

static result[] getTouching(result[] oldBlob, result[] data)
{
  result[] dataWithoutBlob = data.Except(oldBlob).ToArray();

  //we now have two arrays. the old blob and results without the old blob.
  //cycle through each combination of these, and see if they touch.  If they do,
put this in a list of candidate adds.
  List<result> touching = new List<result>();
  //a is the
  foreach (var a in oldBlob)
  {
    foreach (var b in dataWithoutBlob)
    {
      if (areTouching(a, b))
```

```
          {
            touching.Add(b);
          }
        }
      }

      return touching.ToArray();
    }


    static void testCriteriaB2()
    {
      #region loaddata
      System.IO.FileStream fs = System.IO.File.Open(Environment.CurrentDirectory
+ @"\someData2.xlsx",System.IO.FileMode.Open);
      //IExcelDataReader xlReader = ExcelReaderFactory.CreateBinaryReader(fs);
      IExcelDataReader xlReader = ExcelReaderFactory.CreateOpenXmlReader(fs);
      DataSet result = xlReader.AsDataSet();

      while(xlReader.Read())
      {
      }

      int dataWidth = result.Tables[0].Columns.Count;
      int dataHeight = result.Tables[0].Rows.Count;
      //subtact one to make it zero indexed.

      //double[,] data = new double[dataWidth + 1, dataHeight + 1];
      result[] data = new result[(dataWidth) * (dataHeight)];

      int cntr = 0;
      for (int i = 0; i < dataWidth; i++ )
      {
        for(int j=0; j < dataHeight; j++)
        {
          float value = (float)((double)result.Tables[0].Rows[j][i]);

          data[cntr] = new result(value,i,j);
          cntr++;
        }
      }

      for(int j=0; j<= data.Max(u=>u.y); j++)
      {
        for (int i = 0; i <= data.Max(u=>u.x); i++ )
        {
          Console.Write("\t{0}", data.Single(u=>u.x ==i && u.y == j).value);

        }
        Console.WriteLine();
      }
        fs.Close();
#endregion

      //counting not working. should be returning 2.
        //float count = criteriaB(data, 0.07f);
      //giving me 5 when I have an unrelated dot.
        float count = criteriaB(data, 0.07f);
    }

    static bool isLessThanCutoff(result a, result b, float tolerance)
    {
```

```
  float dif = Math.Abs(a.value - b.value);
  if (dif< tolerance)
  {
    return true;
  }
  else
  {
    return false;
  }
}

static float criteriaA(result[] r, float time)
{
  return r.Max(u => u.value);
}
static float criteriaB(result[] r, float tolerance)
{
  //get the max. seed the blob with the max.
  result max = r.OrderByDescending(u => u.value).ToArray()[0];

  result[] oldBlob = new result[]{max}; //seed the old blob with the max.
  result[] blob = oldBlob;

  do
  {
    oldBlob = blob;

  //get the pixels that touch the blob.
  result[] touching = getTouching(oldBlob, r);
    //create a new data set where the elements are the r set without the old
blob.
```

```
    result[] withoutOldBlob = touching.Except(oldBlob).ToArray();
    //subtract those where the difference is too great.
      //need to pass in r without hte old blob, not just r
      //NB! need to remove those not touching!
    result[] toAdd = removeBelowTolerance(withoutOldBlob,tolerance);

    //if the number to change stays the same, break the loop.
    blob = oldBlob.Concat(toAdd).ToArray();

      //does the oldBlob and the blob have all the same elements
      //asssume the two sets are the same.
    }while(!resultSetsSame(blob, oldBlob));
    return blob.Count();
  } //think in terms of object transformations.
  static float criteriaC(result[] r)
  {
    float max = r.Max(u=>u.value);
    float min = r.Min(u=>u.value);

    return max / min;
  }
static void logResults(string fileName, result[] results)
{
  List<string> entries = new List<string>();
  using (System.IO.StreamWriter file = new System.IO.StreamWriter(fileName))
  {
    foreach (var r in results)
    {
      entries.Add(String.Format("({0},{1})\t{2}", r.x, r.y, r.value));
    }
    foreach (var s in entries)
```

```
            {
                file.WriteLine(s);
            }
        }
    }
}
    class resultPositionComparator : IEqualityComparer<result>
    {
        public bool Equals(result a, result b)
        {
            return (a.x == b.x) && (a.y == b.y);
        }

        public int GetHashCode(result obj)
        {
            return obj.x ^ obj.y;
        }
    }
    class criteriaSet
    {
        public float critA;
        public float critB;
        public float critC;
    }
}
```