Air Force Institute of Technology

# AFIT Scholar

Theses and Dissertations

Student Graduate Works

3-2-2004

# Special Nuclear Material Imaging Using a High Purity Germanium Double Sided Strip Detector

Clifford V. Sulham

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Nuclear Engineering Commons

**SPECIAL NUCLEAR MATERIAL IMAGING USING A HIGH PURITY GERMANIUM DOUBLE SIDED STRIP DETECTOR**

THESIS

Clifford V. Sulham, Captain, USAF

AFIT/GNE/ENP/04-07

**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**
# AIR FORCE INSTITUTE OF TECHNOLOGY

**Wright-Patterson Air Force Base, Ohio**

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GNE/ENP/04-07

**SPECIAL NUCLEAR MATERIAL IMAGING USING A HIGH PURITY GERMANIUM DOUBLE SIDED STRIP DETECTOR**

THESIS

Presented to the Faculty

Department of Engineering Physics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science (Nuclear Science)

Clifford V. Sulham, BS

Captain, USAF

March 2004

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

# SPECIAL NUCLEAR MATERIAL IMAGING USING A HIGH PURITY GERMANIUM DOUBLE SIDED STRIP DETECTOR

Clifford V. Sulham, BS
Captain, USAF

Approved:

<table>
<tr><td>_Larry W. Burggraf_ (signature)</td><td>2 Mar 04</td></tr>
<tr><td>Larry W. Burggraf (Chairman)</td><td>Date</td></tr>
<tr><td>(signature)</td><td>2 Mar 04</td></tr>
<tr><td>James C. Petrosky (Member)</td><td>Date</td></tr>
<tr><td>_George John_ (signature)</td><td>2 Mar. 04</td></tr>
<tr><td>George John (Member)</td><td>Date</td></tr>
<tr><td>(signature)</td><td>02 MAR 2004</td></tr>
<tr><td>Julie Skipper (Member)</td><td>Date</td></tr>
</table>
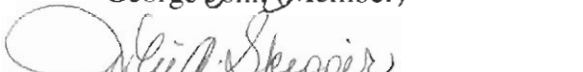
AFIT/GNE/ENP/04-07

## Abstract

The purpose of this research is to investigate the feasibility of using a High Purity Germanium Double Sided Strip Detector (HPGeDSSD) to image Special Nuclear Material (SNM) for use in the field of nuclear nonproliferation. SNM shape can be determined by using a gamma ray imaging system that is capable of imaging either Pu-239 or U-235. The detector used for this research is a liquid-nitrogen cooled germanium detector with five charge collection strips on both faces. The strips on the two faces are orthogonal to each other, creating a five by five array of data collection pixels. When the detector is coupled to a parallel hole collimator, an imaging system is achieved. The system is capable of imaging individual or multiple photon energies. The capability of the imaging system to produce images of simulated Pu-239 was demonstrated. The analyses of the images allowed distributed sources to be distinguished from point sources, which was the primary goal for developing this technology.

## Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. Burggraf, for his guidance and support throughout the course of this thesis effort. The insight and experience was certainly appreciated. I would, also, like to thank the Defense Threat Reduction Agency for sponsoring this effort. And a special thanks to Mr. Eric Taylor for all of his help in getting the equipment necessary to make this project a success. Finally thanks must go to my wife for all of her unending support as I put in long hours to complete this research, none of it could have been done without her support.

Clifford V. Sulham

**Table of Contents**

## List of Figures

## List of Tables

# SPECIAL NUCLEAR MATERIAL IMAGING USING A HIGH PURITY GERMANIUM DOUBLE SIDED STRIP DETECTOR

## I. Introduction

**Motivation**

The mission of the Defense Threat Reduction Agency (DTRA) includes reducing the threat of weapons of mass destruction. Detecting and monitoring special nuclear material (SNM) in various scenarios is important to DTRA objectives. For example, one monitoring mission involves performing on site weapons inspections of countries that are signatories of various arms control treaties. The purpose of these inspections is to verify and determine both the presence and quality of the SNM that is contained within the nuclear weapons, which are stored in a cask. The SNM is assumed to be plutonium for this work. Two other assumptions about the inspections are that the SNM cannot be removed from the cask, and active inspection techniques are not permitted. Because of treaty limitations, which only allow passive inspection techniques, only the radiation that penetrates the cask is available for measurement. Photon radiation above 200 keV is energetic enough to have a measurable flux that can be detected on the outside of the cask [6]. There are at least two types of information from gamma spectra that are useful in detecting spoof weapons, first the gamma spectrum of the weapons grade Pu, second the size and shape of the pit, the source of the gamma radiation.

Using a moderate resolution gamma detector, Pu-239, the fissile isotope of plutonium can be detected.  However, when using a high resolution gamma detector,  the grade of the plutonium can be determined, and weapons grade plutonium (WGPu) can be distinguished from reactor grade plutonium (RGPu).

Current detectors used during the inspections do not have the capability to distinguish whether the SNM inside a bomb is a distributed source or a point source.  The NaI(Tl) detectors typically used for field gamma spectroscopy do not have the energy resolution necessary to identify whether the SNM is WGPu or RGPu.  In fact, currently only high purity germanium (HPGe) detectors have both the resolution and efficiency needed to discriminate RGPu from WGPu.   In WGPu, the Pu-239 is the predominant isotope, having small amounts of Pu-240 (<7% of the Pu).  The amount of Pu-240 is minimized to reduce the rate of spontaneous fissions that occur, resulting in a lower probability of the weapon preinitiating.

**Background**

A device that shows promise as a detector that could be useful for treaty monitoring is a germanium strip detector.  These types of detectors have the energy resolution needed to differentiate RGPu from WGPu. In addition they may be able to determine the size and shape of the pit from which the Pu gamma radiation originates. The strip detector used for this research is smaller than conventional HPGe detectors, which has the disadvantage of lower detection efficiencies for gamma detection at high gamma energies. But, as a trade off, the strip detector provides the capability to image the

SNM contained within the pit. A discussion of the physical dimensions of the weapons pit is discussed in Chapter Two.

**Strip Detector Imaging**: To produce images a detector system must be position sensitive. One way to achieve this, without sacrificing energy resolution, is to use a segmented detector such as a germanium strip detector. Mating the segmented Ge detector to a collimator provides the capability for a germanium strip detector to produce an image. Collimators reduce the radiation that is incident on a portion of the detector to a known range of incident angles. While a directional collimator greatly reduces the number of incident photons on the detector, which reduces the absolute efficiency of the system, it also results in the ability of the system to produce an image. Figure 1 shows the coupling of the detector with the collimator to create an imaging system.
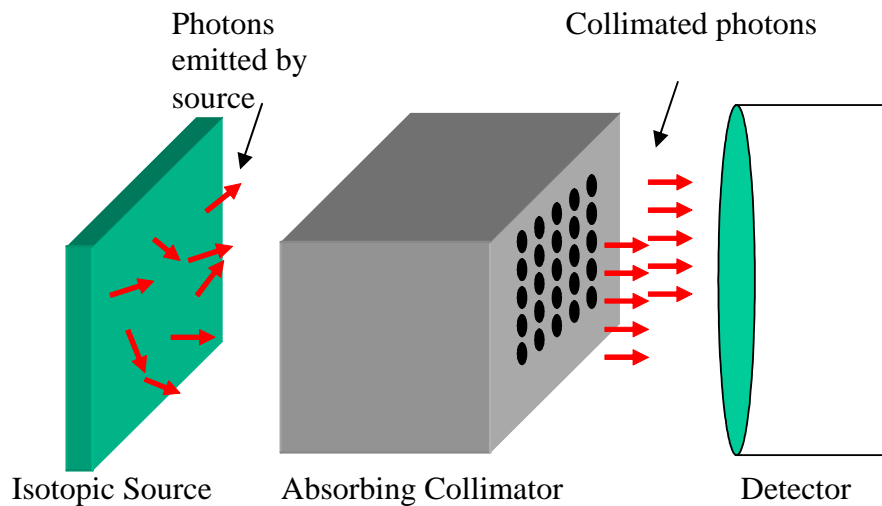
Figure 1. Collimated imaging system, demonstrating collimator use to select gammas directed toward the detector

**Imaging Device:** The device used for imaging employs a germanium strip detector that has 25 pixels in a five by five array. To make an imaging system, a parallel hole collimator with a square five by five array of cylindrical holes was placed in front of the detector. The alignment of the collimator with the detector ensures that each of the collimator holes is placed directly in front of a detector pixel. Each of the detector pixels is 5mm by 5mm square, and the collimator holes are circular with a diameter of 3.6mm. The collimator thickness is 50mm, and is made of lead. The positioning of the collimator in front of the detector is shown in Figure 2.



Lead Collimator          Detector pixels          Imaging Detector
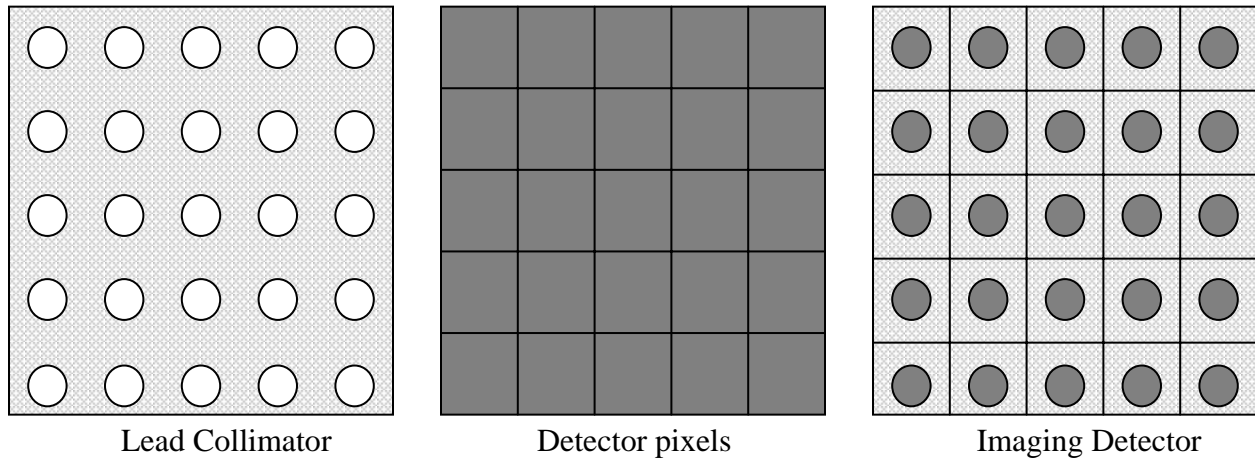
Figure 2. Alignment of the collimator in front of the detector pixels to create the imaging detector

**Objective**

The focus of this thesis is to use a high purity germanium double-sided strip detector (HPGeDSSD) to image simulated SNM by using gamma radiation emitted from

the source.  The detector has 25 pixels that are capable of differentiating a distributed

source from a point source by using a technique similar to a simple Anger camera [1-3].

If all of the pixels are used simultaneously to simulate a larger single detector, the

resolution and efficiency may be high enough to determine the grade of the plutonium.

This research determined if the energy resolution of a HPGeDSSD is sufficient to

determine the grade, while measuring the spatial information simultaneously with the

same detector.  The detection efficiency, particularly for gamma energies between 200

keV and 700 keV, was also determined.

 To image gamma radiation from the pit of a nuclear weapon, photons with enough

energy to escape from the weapons shielding/tamper and containment cask must be used.

However, the efficiency of germanium detectors decreases with increasing energies.

Therefore the photon energies used for imaging should be kept low enough to maximize

detection efficiency yet high enough to frequently escape from the containment cask.

 To image Pu-239 in SNM, gamma rays having good yields in the correct energy

range are required.  Pu-239 has a prominent 413.71 keV peak that should be high enough

in energy to escape from the system and still be low enough in energy to be efficiently

detected and imaged.  Pu-239 also has a 375.05 keV peak that may be used if it is high

enough in energy to escape from the system.  The 413 keV peak has a yield of 0.00147%

and the 375 keV peak has a yield of 0.00155 %.  In an operational system one or both of

these peaks are proposed to be used for imaging the pit.  However, for this work neutron-

activated Sn sources will be used to simulate the SNM.  The activated Sn has a 392 keV

peak as well as a 427 keV peak.

For determining whether the plutonium is RGPu or WGPu a ratio of counts in two gamma peaks from Pu-239 and Pu-240 can be used. When the areas under two gamma peaks from different isotopes are compared with each other, knowing the decay fraction of each gamma peak, the ratio can be used to approximate the relative amounts of the two isotopes if these two gamma rays are attenuated identically or if their different attenuations are known. For treaty verification applications the relative amounts of Pu-239 and Pu-240 in the SNM are desired. Pu-239 has a 645.94 keV peak with a yield of 0.00002%, and Pu-240 has a 642.35 keV peak with a yield of 0.000013%. In addition to these two peaks, there are also a 639.99 keV peak from Pu-239 and a 641.47 keV peak from Am-241. The Am-241 peak will build up over time from the decay of Pu-241. Because these gammas have similar energies their attenuations are similar, regardless of shielding around the SNM.

Unfortunately the efficiency of the detector around 645 keV is significantly worse than at 400 keV. There are also fewer of these photons that are emitted by the source. To maximize the efficiency of the detector for these high-energy photons, the information from all 25 pixels on the detector will be combined and used simultaneously to simulate a single larger detector.

**Scope**

This thesis work began by setting up all of the equipment and getting it to the point that data could be taken. Then the characteristics of the HPGeDSSD detector were obtained. This included determining the efficiency and resolution of each of the pixels,

as well as determining the energy calibration needed for each of the pixels. Then it was shown that a HPGeDSSD could be used to image radioactive materials in the range of 400 keV gammas. Next it was determined that the 25 pixels could be used as a single detector for determining the ratio of Pu-239 to Pu-240. This required a very accurate energy calibration of each of the pixels in the 650 keV range, since the separation of the two photons is close to the theoretical resolution achievable by HPGe detectors.

**Assumptions**

Several assumptions were made about the way the detector would be used in an operational environment. The first is an assumption that the SNM that is being imaged is symmetrical, since the image is only two dimensional. However, while this assumption must be made during inspections, this project used a source which was not symmetrical for demonstrating the ability of the HPGeDSSD to image. The second assumption is that the maximum amount of time allowed to image SNM during operations will be 30 minutes, however this work will not impose restrictions on the counting times. The third assumption is that the SNM located within the pit is plutonium. Finally it is assumed that during the inspection process, the SNM cannot be removed from the cask, and only passive inspections are allowed.

**General Approach**

To produce an image of a radioactive source, all of the detector elements must be calibrated to the same energy scale, and the collimator must be properly aligned in front of the detector, so that a single hole of the collimator allows photons to illuminate a single pixel. Then a code must sort through the raw data, which is collected by the data acquisition system discussed in Chapter III, to produce an image.

**Sequence of Presentation**

Chapter II gives the reader an introduction to the physics of germanium detectors, the pulse processing, an introduction to strip detectors, and finally how the strip detector is used to create an image. Chapter III describes the detector, processing electronics, controlling software, and radiation sources used during the experiment. Chapter IV details the experimental and computational procedures employed during the project. Chapter V presents the results obtained during the project, and Chapter VI provides conclusions and recommendations for further work.

<center>**II. Theory**</center>

**Chapter Overview**

        The principles of germanium detectors, pulse processing, and strip detectors are introduced in this chapter. These principles will lay the foundation for the experimental setup and data analysis, which are discussed in Chapters IV and VI respectively.

**Germanium Detectors  [5, 16]**

        Like all semiconductor detectors, germanium detectors operate on the collection of electron-hole pair charges produced by ionizing radiation. When an interaction occurs between a gamma-ray and the solid state detector, the number of electron-hole pairs produced is a function of the stopping power (dE/dX), and the energy required to produce an electron-hole pair. The number of electron-hole pairs that contribute to the output signal is determined by the charge collection efficiency, which is dependent on the carrier mobility, lifetime of the carriers, and the collection time. The collection time decreases with an increasing field gradient that is present in the detector. The bias voltage applied must produce full "depletion" of charge carriers to insure efficient charge collection from the entire active volume of the detector. Achieving full depletion reduces the detector capacitance and maximizes the signal-to-noise ratio. For detection at gamma ray energies from 200 to 700 keV, the depletion depth becomes very important. The greater

<center>9</center>

the depletion region the better the detector will be at stopping higher energy gammas.

The thickness of the depletion region is given by [16]

$$d = \left( \frac{2\varepsilon V}{eN} \right)^{\frac{1}{2}}$$ (1)

where:

V is the reverse bias voltage,
N is the net impurity concentration in the bulk semiconductor material,
e is the electronic charge, and
$\varepsilon$ is the dielectric constant.

Therefore at a fixed bias voltage, the only way to increase the depletion depth is to lower the impurity concentration. For the detector used in this research, this was achieved by using high purity germanium (HPGe), where the impurity concentrations are approximately $10^{10}$ atoms/cm$^3$. This allows for a depletion region of about 9mm at 1000V. An example of a planar HPGe detector, fabricated out of p-type germanium is shown in Figure 3 [16].
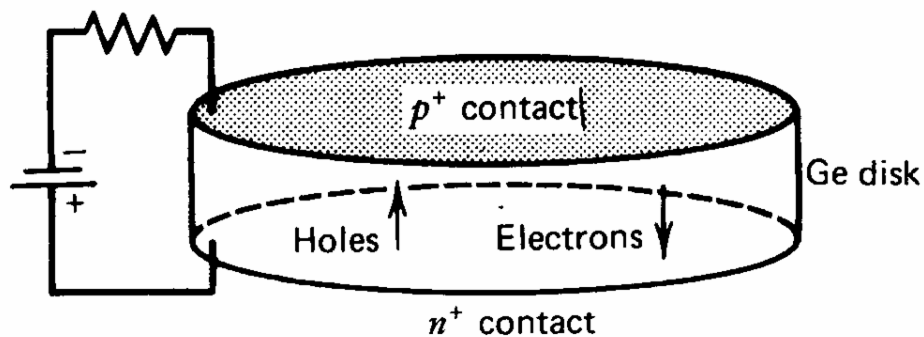


Figure 3. Planar germanium detector configuration

Planar HPGe detectors have electrical contacts on two flat surfaces. The n+ contact is formed by lithium evaporation, and the p+ contact is formed by ion implantation of boron atoms. When the junction is reverse biased, a depleted region of the detector develops. Typically, HPGe detectors are operated as fully depleted detectors. Reverse biasing of the detector requires that a positive voltage be applied to the n+ contact of the detector and the p+ contact be maintained at a relative ground. When planar germanium detectors are fully depleted and are operated with a large over voltage, the electric field can be assumed to be uniform from one contact to the other. So that the electrons and holes can be treated as drifting under the influence of a constant electric field through the entire volume of the detector.

The most important parameters for any gamma spectrometer are the efficiency and the energy resolution. The efficiency determines the number of counts that can be collected over a given time from a radiation source with a fixed activity, while the energy resolution is the ability to distinguish gamma-rays with similar energies. For germanium detectors the resolution is expressed as the full width at half maximum (FWHM) of a full energy peak, which is a function of three terms shown in Equation 2. [16]

$$W_t^{\,2} = W_D^{\,2} + W_x^{\,2} + W_E^{\,2} \tag{2}$$

where:

$W_t$ = FWHM,

$W_D$ = statistical deviation of charge production,

$W_x$ = charge collection variation, and

$W_E$ = electronic noise contribution.

The statistical deviation ($W_D$) can be expressed as: [16]

$$W_D{}^2 = (2.35)^2 \cdot F \cdot \varepsilon \cdot E \qquad (3)$$

where:

   F = fano factor,
   $\varepsilon$ = electron-hole pair energy (2.96 eV for Ge), and
   E = incident gamma-ray energy.

**Pulse shaping/processing** [24]

   Once the holes and electrons have been collected by the contacts on the detector, they are processed by the preamplifiers, which are built in to the detector cryostat housing. The magnitude of the pulses, when sent from the preamplifier to the pulse processing electronics is proportional to the amount of charge collected, and therefore on the energy that was deposited and collected. An example of the output from the F1 charge collection strip preamplifier is shown in Figure 4.
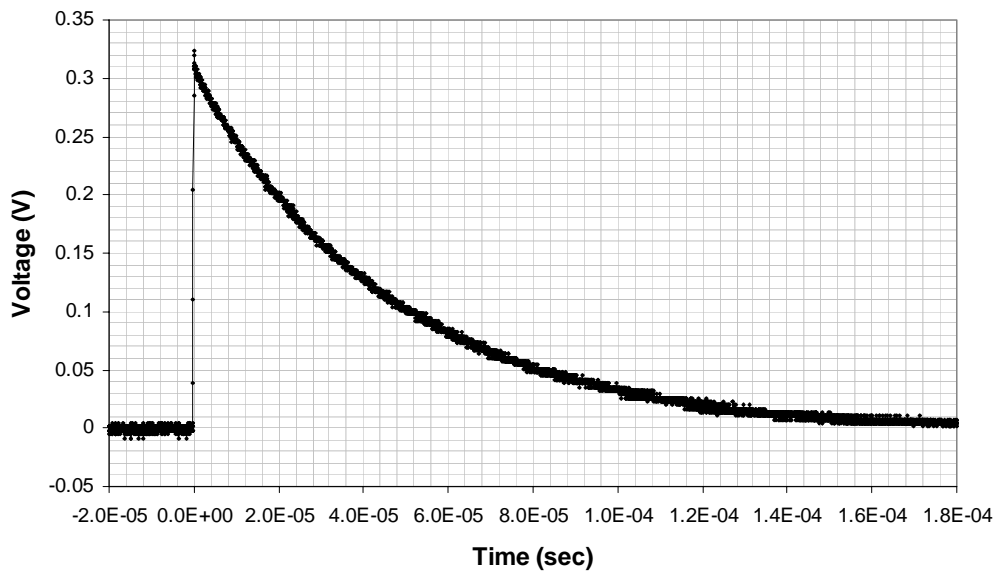


Figure 4. Preamplifier output from the F1 charge collection strip

12

When the pulse is sent from the preamplifier to the pulse processing electronics, the

pulse's waveform is digitized in an analog to digital converter (ADC). Next the signal is

passed to a real time processing unit (RTPU) where a digital filter is applied to shape and

amplify the pulse. The filter used is a trapezoidal filter; this filter typically has a flat top,

which covers the rise time of the pulse which makes the pulse height measurement less

sensitive to the variations of the signal shape. An example of a trapezoidal filter applied

to a preamplifier pulse is shown in Figure 5.
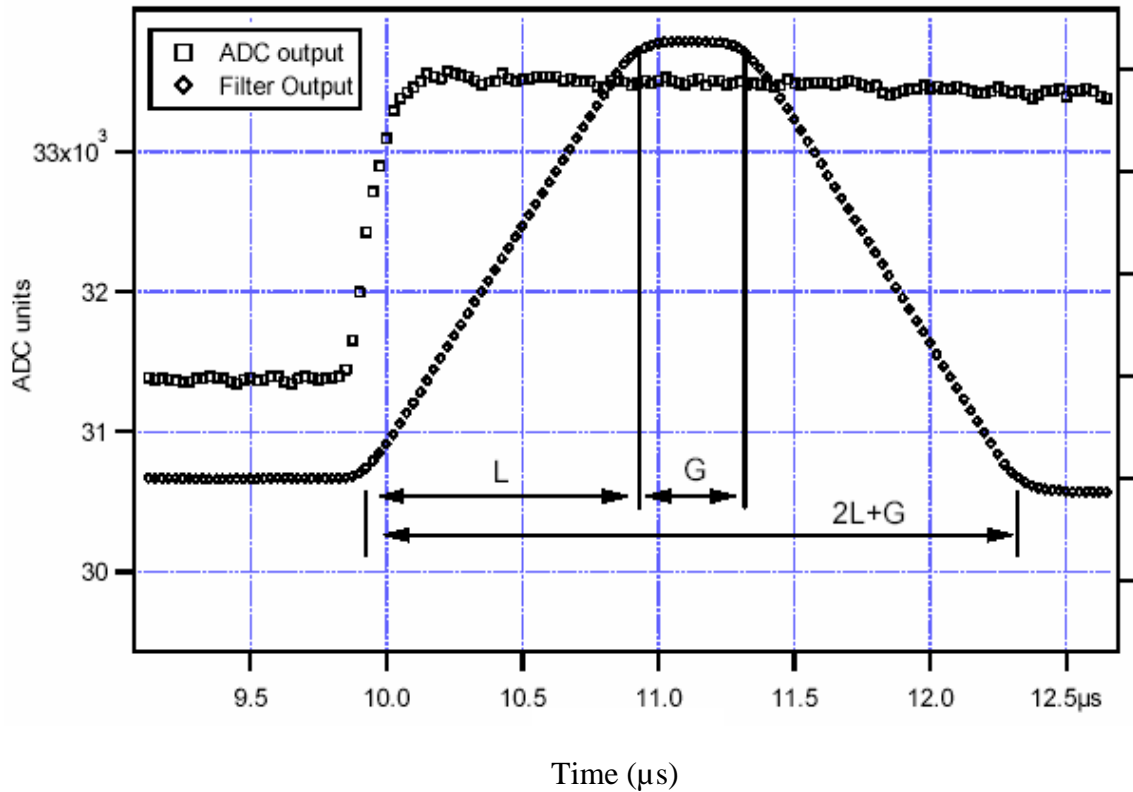


Figure 5. Trapezoidal filter applied to a preamplifier pulse [24]

The rise time L, and the flat top G of the filter can be controlled by the user through the

IGOR controlling software to maximize the energy resolution.

**Strip Detectors**

A germanium strip detector combines the excellent energy resolution of a high

purity germanium detector with the ability to produce an image.  Historically germanium

strip detectors have been used in astrophysics applications, to exploit the high energy

resolution and imaging capabilities, for space applications [18].  Germanium strip

detectors are similar to planar HPGe detectors, except that single contacts to collect the

charges on each side are replaced with discrete strips of contacts that are orthogonal to

each other.  Figure 6 is a diagram showing the charge collection strips of a five by five

strip detector [18].
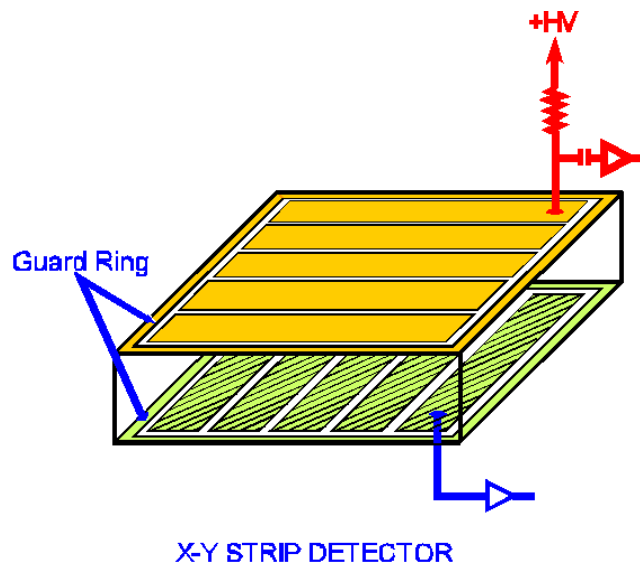


Figure 6. Electrode strip configuration of a strip detector [18]

14

With a standard HPGe detector the charge typically is only read out from one of the contacts, usually the n+ contact which collects the electrons. However, with a germanium strip detector contacts from each strip are read out to individual preamplifiers. Since each interaction within the germanium must produce a hole and an electron, at least one of the strips on each side must collect the charges. The crossed electrodes provide a way to obtain two-dimensional position localization from an ionizing interaction. It is important to note that if the charge recorded from a photoelectric event is recorded by two strips on a single side instead of a single strip; neither of the two strips on that side will record a full energy event. They will only record the fraction of the full energy event that they received. For example if a 400 keV full-energy event is shared equally between two strips, each of them will report a 200 keV event.

**Imaging Using a Strip Detector**

When a gamma photon deposits energy in the detector, the resulting energetic secondary electrons create electron-hole pairs as its energy is dissipated [16]. When an electric field is applied to the system, both the electrons and the holes will undergo a net migration [16]. At the high bias voltage values used (+1000V), the saturated drift velocity is on the order of $10^7$ cm/s, resulting in a collection time of approximately 9 ns for the charge carriers [16]. It is this creation of the electron-hole pair that allows the detector to be used as an imaging device. In order for an event to be used in the formation of the image, one of the front strips and one of the back strips must both record

an event that occurred at about the same energy and at the same time. To be considered the same energy, two pulses must be within two FWHM of a full-energy event peak. And to be considered the same time the two pulses must fall within some coincidence window of time. For this project the time coincidence window was set to 100 ns; this time was determined experimentally and is twice the time necessary to collect the holes.

Therefore, if one of the front strips and one of the back strips have a coincident event in energy and time, the location of the interaction can be inferred as the place where the two strips intersect. As an example, if a charge collection strip on the front, F2, and a charge collection strip on the back, R3, both record an event at the same time and energy, then the intersection of the two strips is the location of the event, this intersection is shown in Figure 7.
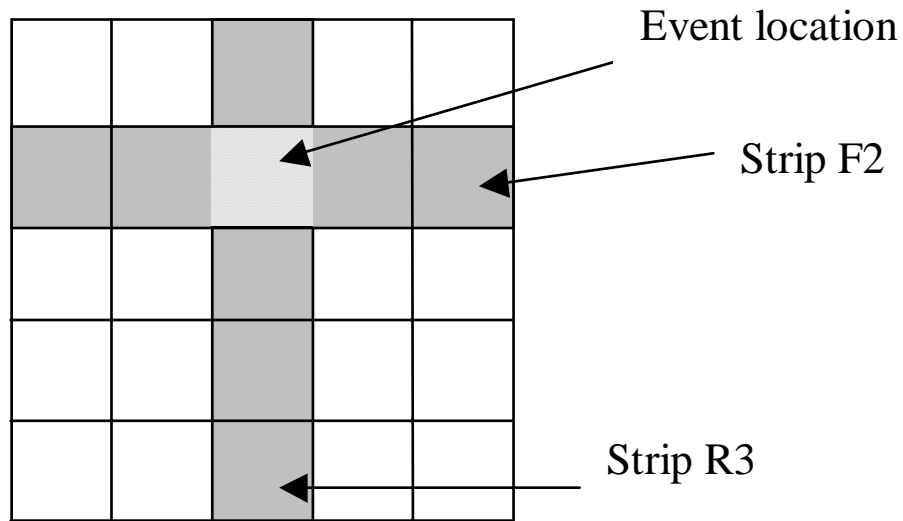
Figure 7. Position localization using two strips of a strip detector

If the source that is illuminating the detector is well collimated, then once the location of the interaction within the detector is known, the location from which the photon emanated will also be known.

**Weapon Assumptions**

The material and physical dimensions of a nuclear weapon are key components that will affect the ability of the HPGeDSSD to produce an image of the SNM. The SNM considered for this project is assumed to be Pu-239. The high-energy portion of a Pu-239 spectrum collected with a germanium detector by the Idaho National Engineering & Environmental Laboratory is shown in Figure 8. The Pu-239 peaks of interest for this project are the 375 keV peak, which is not shown on the spectrum, and the 413 keV peak both of which will be used for imaging, also the 639 keV, and 645 keV peaks, which will be used for grade determination [19]. The 451 keV peak is not used for imaging because its count rate is an order of magnitude less than the 413 keV peak.
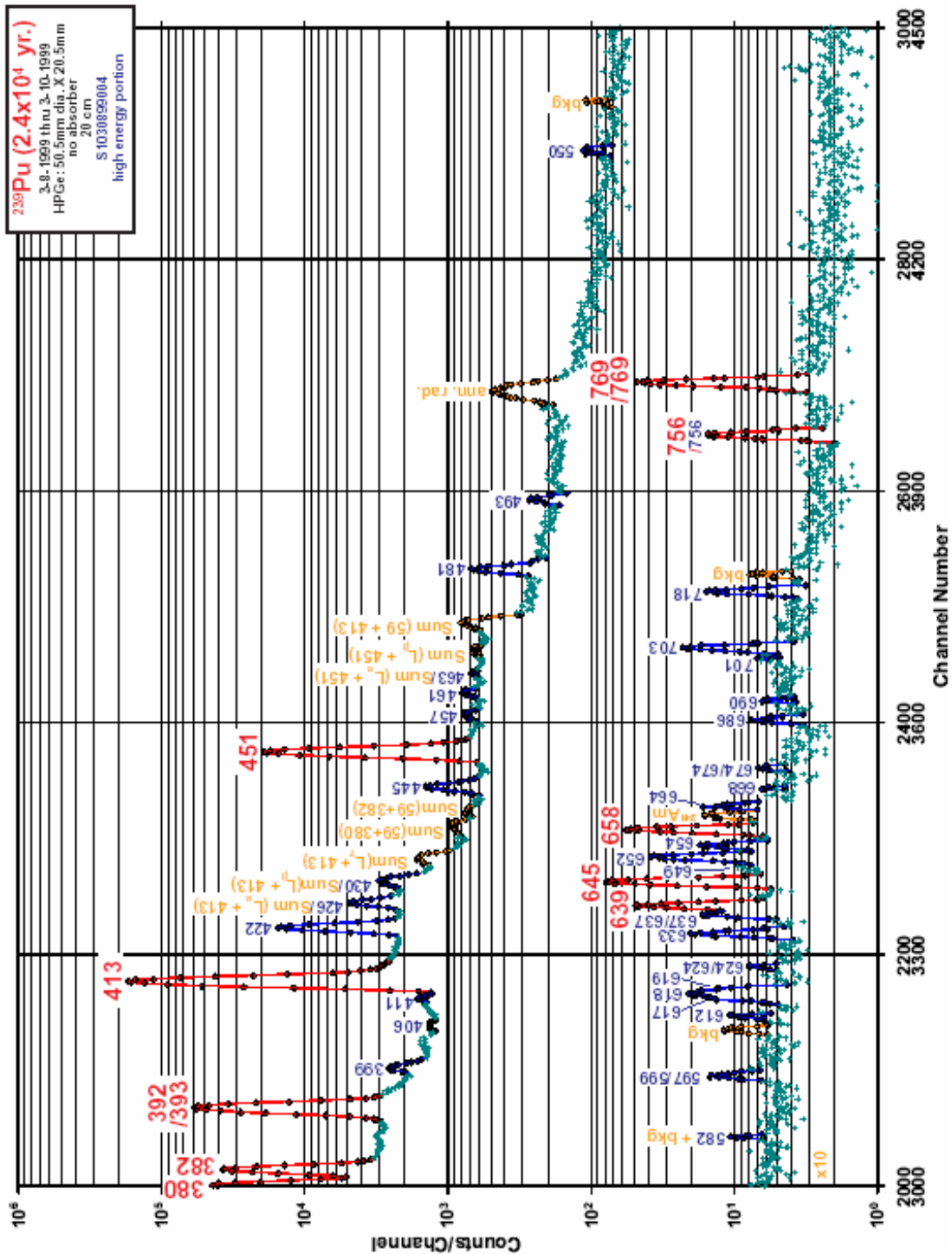
Figure 8. Pu-239 gamma spectrum from a 50.5mm by 20.5mm HPGe detector [12]

18

The time required to produce an image depends primarily on the flux of radiation, at the detector, that is to be imaged. In an imaging application of nuclear weapons, the fluxes are low because of the attenuating materials surrounding the SNM, as well as the low activity of plutonium due to its long half life. An understanding of the physical dimensions of the pit and the shielding around a weapon can be obtained by studying a model published by Steve Fetter in "Detecting Nuclear Materials". In this publication, Fetter presents what he refers to as the general characteristics of an implosion-type fission weapon. Figure 9 is a description of a hypothetical weapon presented in reference [10].
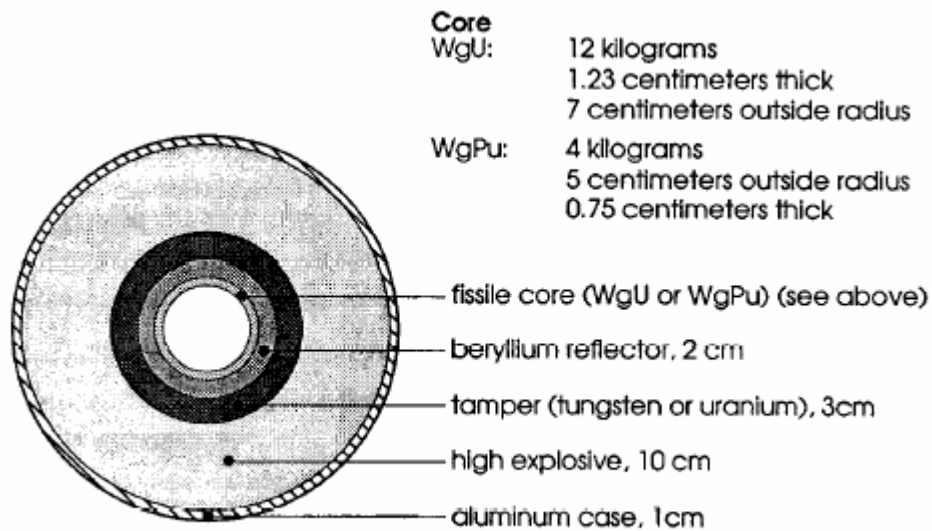


Figure 9. Fetter model of a hypothetical implosion weapon [10]

Without any further information a very broad estimate of the flux on the outside of a cask could be made. An analysis of the estimated flux is performed in Appendix 2. Another paper by Fetter shows a spectrum collected from a Soviet warhead mated to an

19

SS-N-12 sea launched cruise missile [9].  The paper reports the count rate recorded by the

detector for the most prominent Pu-239 peaks, as well as the U-235 peaks.  Because of

the assumption that the SNM is plutonium, only those count rates are shown in Table 1.

The spectrum from the Fetter et. al. paper gives the count rate recorded for 14 different

energy peaks from Pu-239.  It also lists the branching ratios for each of these peaks.

Table 1.  Weapon spectrum data for Pu-239 collected by a HPGe detector [9]

| Energy (keV) | Count rate (counts/sec) | Branching ratio (per decay) |
|---|---|---|
| 332.81 | 0.137 | 5.05E-04 |
| 344.94 | 0.191 | 5.70E-04 |
| 375.02 | 0.862 | 1.58E-03 |
| 380.17 | 0.131 | 3.07E-04 |
| 382.68 | 0.16 | 2.60E-04 |
| 392.99 | 0.373 | 5.60E-04 |
| 413.69 | 1.582 | 1.51E-03 |
| 422.57 | 0.139 | 1.19E-04 |
| 451.44 | 0.318 | 1.92E-04 |
| 640.15 | 0.083 | 7.90E-06 |
| 645.98 | 0.113 | 1.45E-05 |
| 652.18 | 0.075 | 6.40E-06 |
| 756.42 | 0.051 | 3.40E-06 |
| 769.37 | 0.158 | 1.10E-05 |

This rarely published information is important, because knowing the

characteristics of the detector which was used to collect this information will allow a

comparison with another detector, as long as the two detectors are similar types.

Appendix 2 also compares the Fetter model predictions to the reported values from the

Soviet weapon.  The Fetter model did not match the measured values, but if the

thicknesses of the material in the model are modified then the model will produce fluxes

similar to the measured values.

During the collection of the data from the Soviet weapon, Fetter used a HPGe detector which was larger than the HPGe detector used for this project, but the counting rates can be scaled down, which will be done in Chapter IV.  This scaling of the count rates will provide an estimate of count rates that can be expected for the HPGeDSSD used for this research.

## III.  Equipment

**Chapter Overview**

  The primary equipment that was used to assemble the imaging system, was the germanium strip detector, the collimator, the digital gamma finder data acquisition cards (DGF), the power distribution module (PDM), the computer automated measurement and control (CAMAC) crate, the crate controller, and the software to control the data acquisition cards.  A schematic of the entire system is shown in Figure 10.
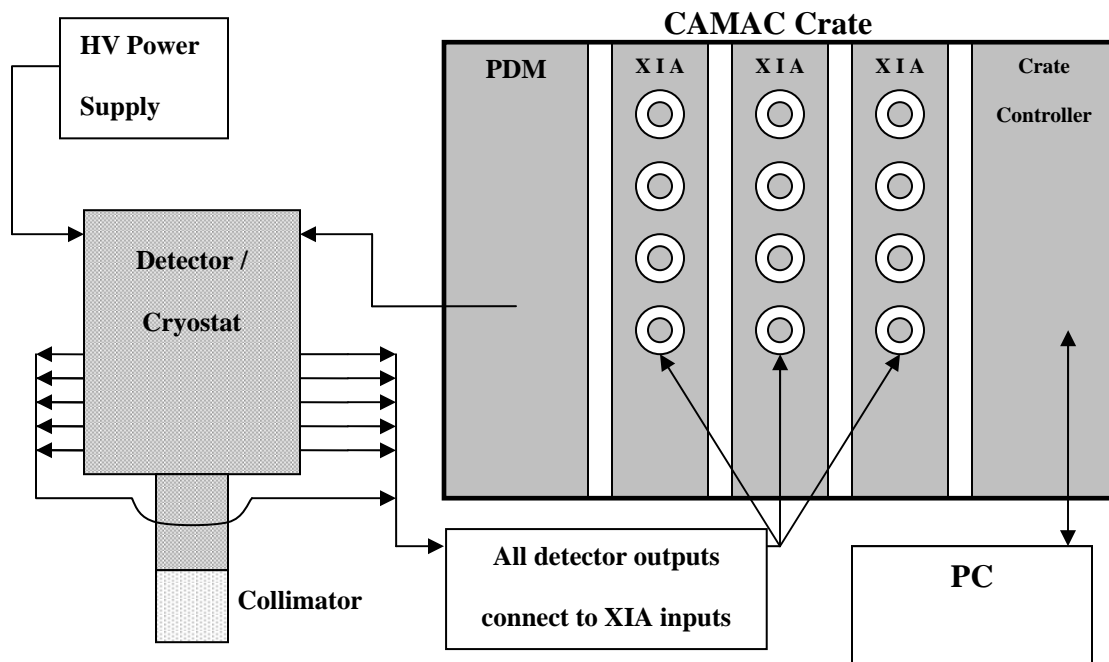
Figure 10. HPGe strip detector imaging system

**Detector**

  The detector used for this project was an ORTEC Model HPGeDSSD, Serial number 42-WPAFB-01.  The detector is a HPGeDSSD that operates at liquid

nitrogen temperature.  A photo of the detector, dewer and collimator is shown in Figure

11.



**Dewar**

**Preamplifier Housing**

**Detector / Cryostat**
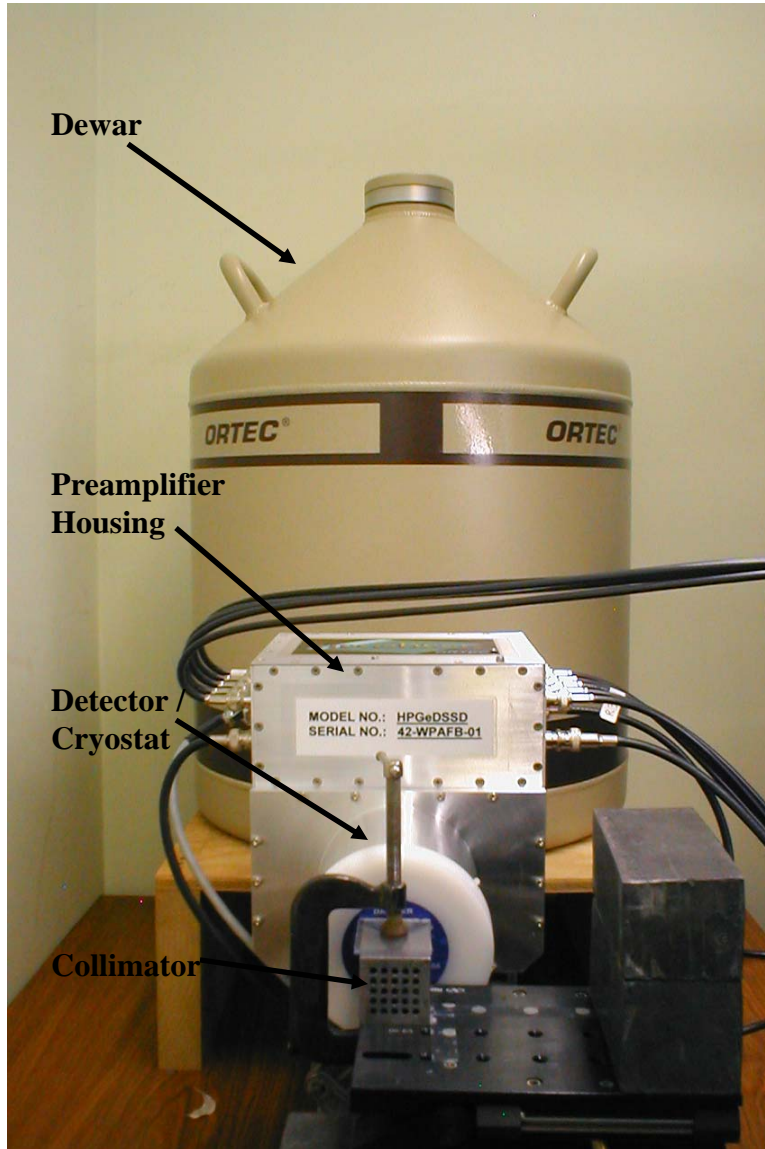
**Collimator**

Figure 11. Photograph of the detector, dewer, and collimator

The detector has 5 charge collection strips on each side of the detector, it was

designed and procured by Mr. Matt Lange, a previous AFIT masters student.  The strips

on opposing sides are orthogonal to each other, which allows for the determination of

location when a gamma interaction occurs within the germanium crystal.  This effectively creates 25 pixels on the detector.  Each of the charge collection strips is 25mm long and 5mm wide, so that the total active surface area of the detector is 25mm by 25mm. The detector has an active depth of 9mm.  The dimensions of the detector and the charge collection strips are shown in Figure 12.
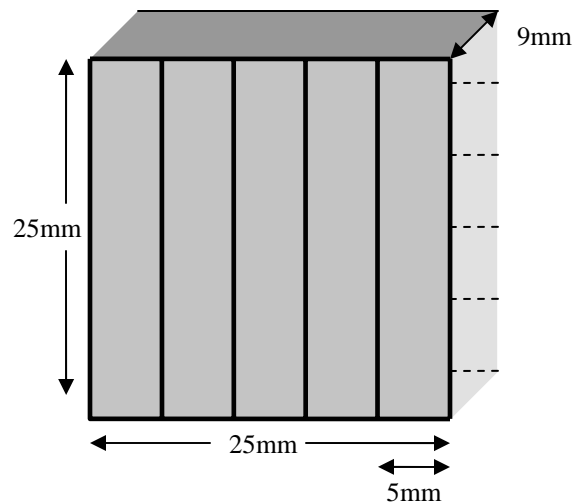


Figure 12. ORTEC germanium strip detector dimensions

There is a 0.5 mm beryllium entrance window, as well as 0.3 µm of inactive germanium.  The detector is operated at +1000 volts reverse bias.  The detector is oriented so that the 25mm by 25mm face is perpendicular to the incident radiation. The charge collecting strips, on the back face of the detector, are lithium and are at +1000V bias.  The strips on the front face, are boron and are at about +1V.  Each of the strips are electrically separated from one another.  Since there is a 1000V potential between the strips, the two sides can be treated as parallel plates, which force any charges or holes to

the charge collection strips. There are 10 preamplifiers built into the detector housing. Each of the charge collecting strips has its own preamplifier.

**DGF Hardware [24]**

For this project, three X-Ray Instrumentation Associates (XIA) DGF model 4C CAMAC cards were used. Their serial numbers were 1240, 1244, and 1230. The DGF-4C is a four-channel all-digital waveform acquisition and spectrometer card. The incoming signals from the detector are digitized by analog-to-digital converters (ADC). Digital waveforms are analyzed by the digital signal processor (DSP) for pulse shape analysis. Then the waveforms, timestamps, and the pulse shape analysis results are read by the host system. The modules can process up to 200,000 counts per second for all four channels combined. The individual modules support coincidence spectroscopy, but there is no way to perform coincidence measurements between modules with the software provided by XIA. The individual modules can be synchronized to the same clock, and trigger signals can be distributed between modules across a backplane.

The DGF-4C modules have four building blocks; the analog signal conditioning, the real-time processing units (RTPU), the digital signal processors (DSP), and the CAMAC interface.

**Analog Signal Conditioning**: Each of the analog inputs has its own signal conditioning unit, which adapts the input signals to the input voltage range of the ADC. There is also a software-controllable gain setting at this stage. The ADC is a waveform digitizer.

**RTPU**:  There is one RTPU for each channel, and it consists of a field-programmable-gate-array (FPGA) and a first in first out (FIFO) memory.  When data arrives from the ADC, the RTPU applies a digital trapezoidal filter which acts as a shaping amplifier.  In addition to the pulse shaping, the RTPU contains a pileup inspector.  If the logic detects a second pulse too close to the one being processed, so that the pulse height measurement of the first pulse would be corrupted, both of the pulses are rejected.  If the pulse is not rejected as a pile up, a trigger is issued to notify the digital signal processor (DSP) that there is raw data ready for processing.  When the trigger is issued to the DSP, the RTPU is latched until it has been serviced by the DSP.

**DSP**:  The DSP controls the operation of the pulse processing electronics, reads the raw data from the RTPU, determines the pulse heights, time stamps each pulse, and prepares the data for output to the controlling computer.

**CAMAC Interface**:  The CAMAC interface is the means of communication with the DGF-4C.

**Power Distribution Module**

The PDM used for this effort is an XIA CAMAC module Revision 3 and serial number 108.  The PDM provides + 12 V and + 24 V preamplifier supply voltages.  It is capable of supplying power for up to 20 preamplifiers.

**Crate and Crate Controller**

The CAMAC crate used in this research is a Wiener type UEC 01 VH / Typ 342, part number 0305.0102, and serial number 3698023.  It is a 25-slot CAMAC crate.  The crate controller is a Jorway model 73A-2, serial number 662.  It is the interface between the CAMAC crate and the host computer.  The interface is via a SCSI connection.

**Collimator**

The incoming photons must be collimated to ensure that they are perpendicular to the detector face.  The collimator used for this research is a parallel hole lead collimator fabricated by Nuclear Fields Inc, based on requirements driven by the pixel array size and attenuation required.  The collimator is 30mm x 30mm x 50mm, and has 25 round holes 3.6 mm in diameter.  The amount of attenuation that is provided by the collimator is a function of the amount of lead that a photon must pass through and the energy of the photon.  If a point source is placed in front of one pixel, the contrast is the difference between the number of counts recorded in that pixel minus the number of counts recorded in an adjacent pixel [7].  At 392 keV, the contrast between the two pixels was 3.4%.  This means that whatever the count rate is at the pixel that is being intentionally irradiated, all adjacent pixels will record a count rate of 3.4% its value.
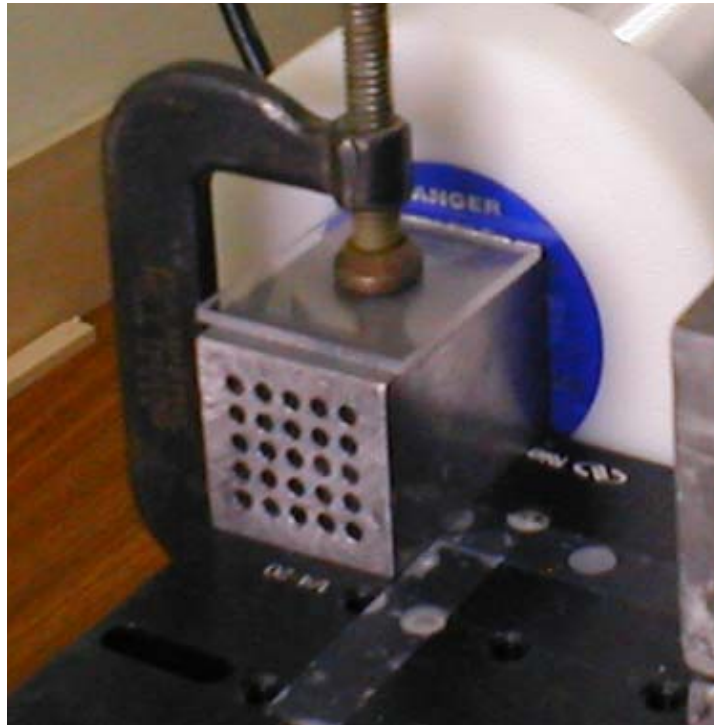
Figure 13. Parallel hole collimator positioned in front of the GSD

## Image Processing Code

To process the raw data that is collected by the DGFs, a processing code had to be written. The processing code was written in the FORTRAN 95 computer language. The code contains seven main algorithms for processing the raw data. The seven steps are listed below, and the full code is presented in appendix A.

1. Read in the raw data. The IGOR software saves the raw data as a text delimited file. This data is read in by the code and stored into an array whose size is 1 x N, where N is the file length.

2. Convert the raw data to an array format that can be manipulated easily. The new array is size four by total number of events recorded. The four entries for each row

are; module that recorded the event, channel that recorded the event, trigger time, and energy of the event.

3. Pick out the events that correspond to the energy peak of choice. Here the events that occur within the energy range of choice are separated out and stored into an array that is again four wide.

4. Sort the chosen events by time. This step sorts the events that are left by time, with the earliest time events at the beginning of the array

5. Pick out coincident events (energy and time). After the events have been sorted by time, the events that are coincident in both energy and time are picked out of the data and stored into a new array.

6. Determine the pixels where the coincident events occurred. Once all coincident events have been found the pixel location where the event took place must be recorded.

7. Print out a matrix of total counts for all pixels. Finally print out the matrix that lists the total number of coincident events that occurred for each pixel.

**Radiation Sources**

**Activated Tin Source**. For this research a distributed radiation source was needed that had photons in the 400 keV range and 600 keV range. These energy ranges were necessary so that the source could simulate the Pu-239 that is assumed to be the SNM. The easiest source to get that satisfied the necessary requirements was neutron-activated tin. Mr. Russ Hastings in the AFIT model fabrication shop made five strips of

Sn. The Sn strips were 5mm wide and 25mm long and approximately 1.7mm thick.

These dimensions were chosen so that the combination of all five strips would cover the

collimator, or individual strips could be imaged. The five Sn strips were activated at the

Ohio State University research reactor on 28 August 2003. The strips had masses

ranging from 1.493g to 1.512g. All five strips were placed in the OSU reactor central

irradiation facility (CIF) for two and a half hours, while the reactor was run at 90%

power, resulting in a flux of $10^{13}$ neutrons per $cm^2$-second. After the irradiation of the

sources was completed, the reactor was shut down and the sources remained in the CIF

overnight to allow the extremely short-lived isotopes to decay. Then the sources were

removed from the CIF and were placed into a lead pig, where they were allowed to sit for

three weeks to let the short-lived activation products decay away. After this cooling off

period, only long-lived activation products remained. The long-lived activation products

resulting from neutron-activating Sn are given in Table 2.

Table 2. Activated tin gamma emissions

| Element | Half life | Photons(keV) | Branching ratios |
|---------|-----------|--------------|------------------|
| Sn 117m | 14 d | 158.91 | .86 |
| Sn 113 | 115 d | 255.30 | .018 |
| | | 391.73 | .6497 |
| Sb 124 | 60 d | 602.7 | .9826 |
| Sb 125 | 2.76 y | 176.57 | .0682 |
| | | 427.89 | .30 |
| | | 463.34 | .105 |
| | | 600.56 | .179 |
| | | 606.34 | .0503 |
| | | 635.88 | .1131 |

Although Sn-117m is not a particularly long lived isotope (14 days), there is so much of it present from activation that it is present in the spectrum for greater than seven half lives. A spectrum of the neutron activated Sn samples recorded by a single strip from the HPGeDSSD is shown in Figure 14. The collection time for the spectrum was 13 hours. The spectrum was collected on 4 December 2003 (99 days after activation).
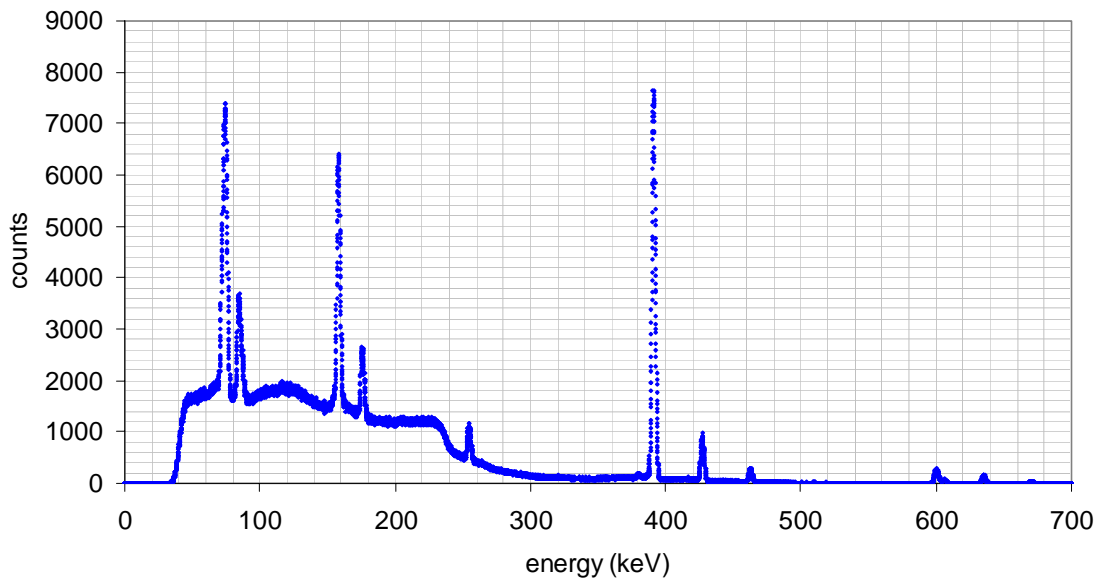


Figure 14. Neutron activated tin spectrum collected with the GSD

**Standard Sources:** The following standard sources were used for determining the efficiency and resolution of the detector. The sources were also used for aligning the collimator with the detector.

Table 3. Radiation sources used for efficiency measurements

| Source | Source Number | Activity (μCi) | Date |
|---|---|---|---|
| Multi Nuclide | T-105 | N/A | 1 Nov 02 |
| Na-22 | T-106 | 0.8883 | 15 Feb 03 |
| Co-57 | T-086 | 85.8 | 15 Aug 98 |
| Cs-137 | T-083 | 8.829 | 15 Jul 98 |

**Fan-In/Fan-Out System**

In order to get each of the three DGF modules to start and stop simultaneously,

the busy out outputs of the three modules had to be passed through a fan-in / fan-out

system and then connected to the three synch in inputs on the three modules.  In an

attempt to create a fan-in / fan-out system, a system of diodes were used to force the

signals to only flow in the correct direction.  For this system, 1N277 diodes were used.

The black band sides of the diodes were connected to the inputs (DGF Busy out) and the

other ends to the outputs (DGF Synch In).  Figure 15 shows the layout of the fan-in / fan-

out system.
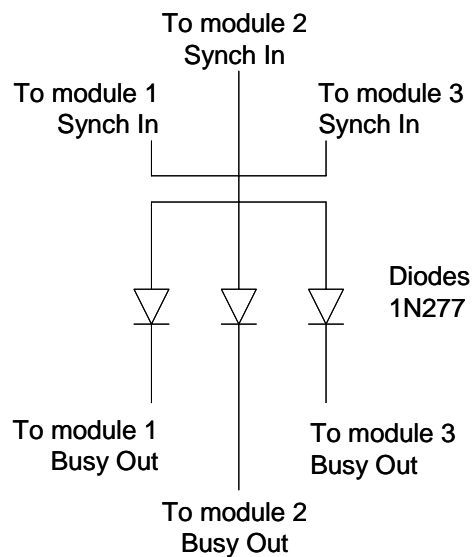


Figure 15. Fan-In / Fan-Out system

This system was found to cause a voltage drop across the diodes which reduced

the logic signal to a value too low to be recognized by the DGF module.  This voltage

drop was not discovered until a new version of the XIA software was received, because

the older version of the software did not produce the correct pulses out of the busy-out

connection.  To correct this problem an ORTEC CO4020 quad four input logic module

was ordered.  Unfortunately the module was not received in time to use for this work.

**Controlling Software / SCSI Interface**

The controlling software used for the DGF-4C modules was the XIA DGF-4C

viewer, release number 3.02.  This software runs inside of the IGOR pro software which

was supplied by XIA with the purchase of the DGF-4C modules.  The SCSI interface

card used in the PC is an Adaptec 2930 SCSI card that fits into a PCI slot inside the

computer.

## IV.  Procedure

**Chapter Overview**

      To create an image, it was first necessary to set up all of the equipment and adjust

the hardware and software settings, so that spectra could be collected for each of the

detector outputs.  Then the detectors had to be characterized and the energies of each of

the channels had to be calibrated.  This ensured that each of the channels had the same

energy calibration.  Finally a code had to be written to process the raw data and produce

an image.

**Equipment Setup**

      The first step in setting up the equipment ensured that all of the internal switch

settings and jumper settings for the Jorway crate controller were in the correct position.

There are four internal switches that were all placed in the OFF position, a jumper was

placed spanning X2 and X3, and it was verified that the controller was internally

terminated.  Then the controller was placed into the two right-most slots of the crate, and

the PDM was placed into the two left-most slots.

      Initially, only a single DGF-4C module was used to verify that the

communication with the host computer was functioning properly.  All internal jumper

settings were verified as being positioned correctly for a single module.  Once the ver

3.02 XIA software and an Adaptec SCSI card were installed on the host computer and the

host computer was connected to the controller with a SCSI cable, the module was

successfully booted.  During the start-up process the operator inputs the SCSI bus ID

number and crate number.  For the current configuration the bus ID number is 2 and the

crate number is set to 3.

Once the module was booted up, a signal pulse generator was used as a signal

input to become familiar with the controlling software.  The setup shown in Figure 16

was used during this familiarization.
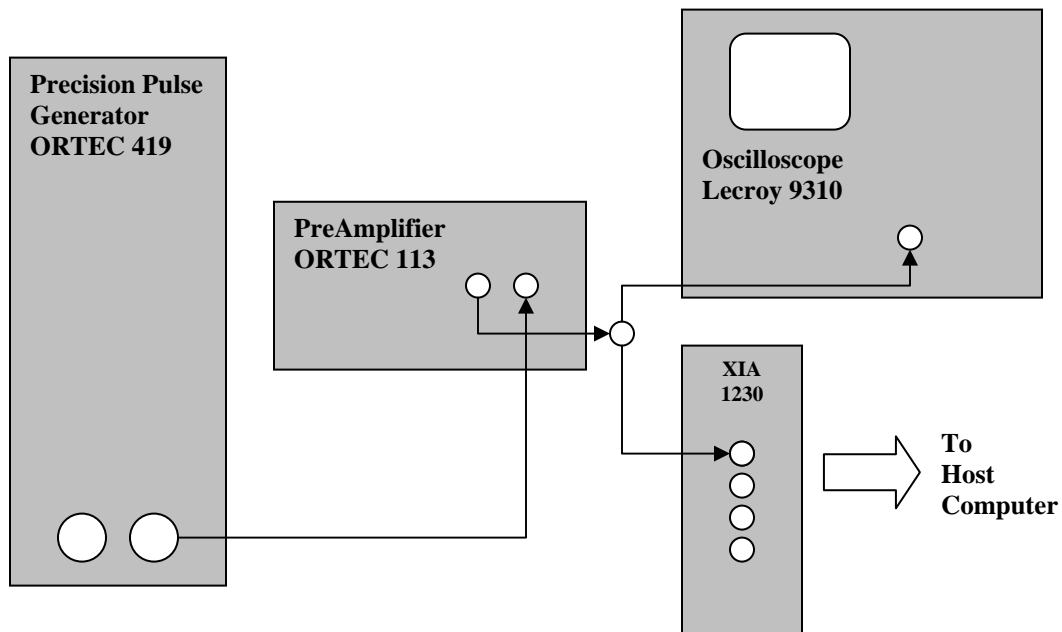
Figure 16. Experiment test setup

After becoming familiar with the software, all three of the DGF-4C modules were

placed into the crate and connected across their backplane to distribute the master clock,

and the internal jumpers were all placed in the XIA-recommended positions.  The

following notation is used throughout this project, module 1 is the right most XIA

module, module 2 is in the center and module 3 is the left most XIA module.  Table 4

gives the jumper settings used when all three modules were placed in the crate together.

Table 4.  DGF-4C module jumper settings

| | Module 1 - 1230 | Module 2 - 1240 | Module 3 - 1244 |
|---|---|---|---|
| Jumper 1 | Installed | Removed | Removed |
| Jumper 2 | Installed | Removed | Removed |
| Jumper 3 | Installed | Installed | Installed |
| Jumper 4 | Installed | Installed | Installed |
| Jumper 5 | Board Clock | External | External |

After the boards were placed in the crate, ribbon cables provided by XIA were used to connect the modules together via the 8-pin connectors on the back of the modules.  Figure 17 illustrates the connections made, as viewed from behind the modules.
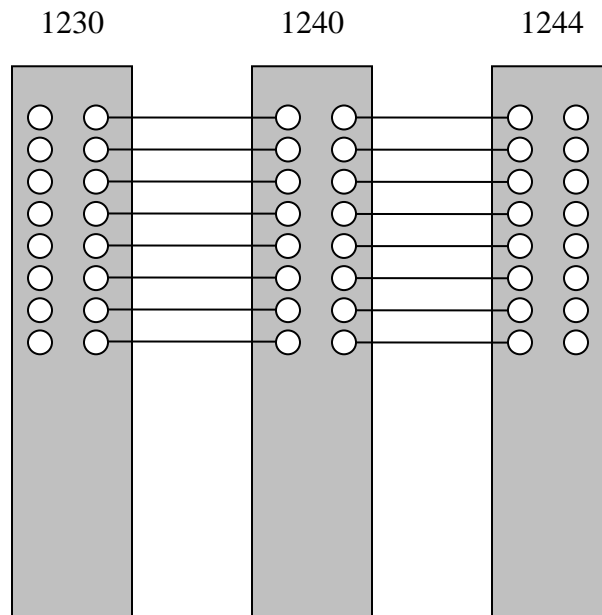
Figure 17. DGF-4C module backplane connection

After the modules had been hooked together across the backplane, the fan-in / fan-out system was connected to the front of the modules.  The three busy out connectors

were hooked to the input side of the fan-in / fan-out system, and the three synch in connectors were hooked to the output side.  Once all connections had been made, the three modules were booted up and seven pulse generators were connected to the inputs of the DGF-4C's.  All seven of the signals were recorded by the MCA software, showing that the system was functioning properly and ready for the detector outputs to be connected to the DGF-4C modules.

The dewer for the HPGeDSSD was filled, and the detector was allowed to cool down for 24 hours.  After the cool down period, the detector was hooked to a ORTEC 659 high voltage power supply.  The power supply internal board was placed in the positive polarity position, and in a position to allow automatic shut off.  The high voltage shut down (HVSD) connector, on the detector, was hooked to the back of the power supply at its automatic shut down connection.  This is a safety feature which will force a shutdown of the high voltage power supply if the detector temperature is allowed to rise above that of LN2.

The high voltage supply line was hooked to the HV-In connector on the detector, and the PDM was connected to the preamplifier supply connection on the detector.  Then the high voltage power was turned on and slowly increased to +1000 volts in 100 volt increments.  At each increment of 100 volts, the detector outputs were observed on an oscilloscope to ensure that as power was increased the noise on each channel was decreasing.   The detector outputs were connected to the DGF-4C modules once the detector had been powered up.  Table 5 lists which connections were made between detector outputs and DGF-4C module inputs.

Table 5.  Detector output to DGF-4C module connections

| Detector Output | Module Number | Channel Number |
|:---:|:---:|:---:|
| F1 | 3 | 0 |
| F2 | 3 | 1 |
| F3 | 3 | 2 |
| F4 | 3 | 3 |
| F5 | 2 | 0 |
| R1 | 2 | 1 |
| R2 | 2 | 2 |
| R3 | 2 | 3 |
| R4 | 1 | 1 |
| R5 | 1 | 3 |

The XIA notation for connectors is: the top connector on the DGF-4C modules is channel 0, and the bottom connector is channel 3.

**Charge-Sharing Between Strips**

To ensure that the charges were being forced to the charge collection strips by the potential; all ten of the detector outputs were viewed simultaneously on oscilloscopes. Five two channel oscilloscopes were used.  All ten of the channels were triggered by the channel displaying the detector output F3.

**Software Settings**

**Gain Settings**: To get the same energy calibration for each of the channels, the gain setting for each channel had to be adjusted so that it would place the centroid of a full energy peak (FEP) at a specified location.  The location that was specified was

38

chosen by finding a channel in the middle of the range and then forcing the other channel

to that point. The FEP selected for the energy calibration was the 392 keV peak from Sn-

117m. The location for this peak was chosen to be channel 20200. Table 6 gives the

gain settings that were used to align the channels.

Table 6. XIA software gain settings

| Output | Gain | Channel |
|--------|--------|---------|
| F1 | 1.2948 | 20204 |
| F2 | 1.3084 | 20198 |
| F3 | 1.2051 | 20201 |
| F4 | 1.2328 | 20203 |
| F5 | 1.1995 | 20201 |
| R1 | 1.2704 | 20199 |
| R2 | 1.2495 | 20198 |
| R3 | 1.2118 | 20200 |
| R4 | 1.2398 | 20205 |
| R5 | 1.2516 | 20204 |

**Filter Settings**: To minimize the FWHM of the peaks, the energy filter rise time

was varied, to determine the effect that it had on the resolution. This was done at the

suggestion of the XIA users guide for maximizing performance. Eleven data points were

taken at different energy filter rise times, and resolution was measured. All

measurements were done using a Cs-137 source and a 900 second collection time.

**XIA Software Settings**: Each of the DGF-4C modules and channels has settings

that had to be set within the software. Many of the settings are the same for all of the

channels. The settings that are constant for each active channel are listed in Table 7.

Table 7. XIA software common settings

| | |
|---|---|
| Trigger filter rise time (μs) | 0.3 |
| Trigger filter flat top (μs) | 0.2 |
| Energy filter rise time (μs) | 3.6 |
| Energy filter flat top (μs) | 1.2 |
| Threshold | 100 |
| Module CSRA | 2400 |
| Channel CSRA | 02B4 |
| Cutoff | 5000 |
| Binning factor | 1 |

There are also several settings that must be adjusted for each of the channels that are active. An active channel is one that had an input. The individual software settings are listed in Table 8.

Table 8. XIA software individual settings

| Module | Channel | Gain | Offset | Tau | Coin. Pattern |
|---|---|---|---|---|---|
| 1 | 1 | 1.2398 | 0.0409 | 45.7 | 0504 |
| 1 | 3 | 1.2516 | 0.0660 | 50.8 | 0504 |
| 2 | 0 | 1.1995 | -0.1042 | 43.2 | FFFE |
| 2 | 1 | 1.2704 | 0.0598 | 43.1 | FFFE |
| 2 | 2 | 1.2495 | 0.0830 | 45.0 | FFFE |
| 2 | 3 | 1.2118 | 0.0443 | 47.6 | FFFE |
| 3 | 0 | 1.2948 | -0.0825 | 47.4 | FFFE |
| 3 | 1 | 1.3085 | -0.0944 | 43.8 | FFFE |
| 3 | 2 | 1.2051 | -0.0968 | 41.8 | FFFE |
| 3 | 3 | 1.2329 | -0.0981 | 52.1 | FFFE |

**Efficiency Calculation**

To determine the efficiency of the entire detector, three radiation sources were used, for a total of four FEPs. The sources, one at a time, were placed 11cm from the detector, and located along the axis of the detector. The counting time for each source was 600 seconds. For the calculation of the solid angle, I assumed a right circular

40

detector with a radius that had a surface area equal to the surface area of the ORTEC

detector.  Equation three was used to calculate the solid angle [16]

$$\Omega = 2\pi\left(1 - \frac{d}{\sqrt{d^2 + a^2}}\right)$$ (3)

where:

      $\Omega$= Solid angle
      $d$ = Distance between the source and detector
      $a$ = Detector radius

And the intrinsic efficiency was found using equation 4 [16]

$$\varepsilon_{int} = \frac{counts\ recorded}{A \cdot t \cdot b \cdot \dfrac{\Omega}{4\pi}}$$ (4)

where:

      $\varepsilon_{int}$= Intrinsic efficiency
      $A$ = Activity of the source (Bq)
      $t$ = Counting time (sec)
      $b$=Branching ratio
      $\Omega$ = Solid angle

**Resolution Determination**

      The resolution was measured at 635.88 keV with emissions from the Sb-125

isotope, in the activated Sn source, with a counting time of 46,800 seconds.  The

resolution of each of the charge-collection strips was measured using a normal

distribution fit that is built in to the IGOR software.  The fit was used because it

automatically subtracts out the background.  The fit gives both the absolute FWHM, and

the percentage FWHM.  For purposes of reporting the resolution, the absolute FWHM value was used.

**Imaging**

To create an image, the raw data must first be collected by the detector, and then that data has to be processed by the in-house AFIT processing code to produce the image.

**Data Collection**:  For the collection of data, that is to be used for an image, the XIA software must be set to take list-mode data.  In the run tab of the XIA software, the run type must be set to energy and time only.  In this list mode, only the energies and the time stamps are stored for each event that occurs.  During operation the memory in the level-1 buffer can only store up to 744 events, and then it must dump the data to a larger memory.  Each of the collections of 744 events is referred to as a spill.  Within the XIA software a maximum time for each spill can be specified, assuming that the buffer is not filled before the time runs out.  You also must specify the number of spills to be stored during data collection.  Once the data collection is complete, the DGF-4C module downloads all of the data to the PC as a binary file.  Within the XIA software, the data can be converted from its binary form and then saved as a text delimited file.  It is this file that is read in by the processing code.

**Code**:  The key algorithms found in the code are discussed in chapter III, but there are several steps that must be followed to execute the code.  When it is run, there is a default energy calibration that can be changed if it does not match what is seen on the

collected spectrum. The user inputs either yes or no when prompted about whether the energy calibration should be changed. After the default calibration has been accepted or a new calibration entered, the user must enter the energy of interest for imaging. Finally, the user must enter the file name where the raw data delimited text file was saved. While the code is executing it will inform the user when each of the algorithms is complete, and when the image matrix has been compiled the code will open the array visualizer and graphically display the image.

**Collimator Alignment**

The alignment of the collimator with the detector to create an imaging system was done experimentally. Initially, it was assumed that the detector was located in the center of the cylindrical cryostat housing. The collimator was place in the approximate center, and an image of a point source located at the center collimator hole was taken, and the collimator moved if the image of the source was not reported in the correct pixel. This was repeated until the image of the point source was located in the center pixel. The collimator was placed on a lab jack for vertical alignment and was simply repositioned on the jack for the horizontal alignment. Once the collimator was in the correct position, it was clamped to the lab jack and lead weights were placed on the jack to keep it from moving in the horizontal direction.

## Statistics of Imaging

In order to measure the standard deviation and the mean of the number of pixel counts recorded a Komolgorov-Smirnov test was performed. This was done to determine if the standard deviation was equal to the square root of the mean,

$$\sigma = \sqrt{\overline{x}} \qquad (5)$$

where:

$\sigma$ = standard deviation
$\overline{x}$ = mean

a simple two by two array was imaged using the distributed source. The energy of interest for the image was 392 keV and the image was taken with the pixels uniformly illuminated. In creating the images the energy discrimination window was set at $15200 \pm 125$ channels. The energy coincidence window was set to increment from 0 to 250 in steps of 10, and the time coincidence window was set to increment from 0 to 4 in steps of 1. Thirty images were taken, without allowing the source or collimator to move. The data collection was set at a maximum of 5 seconds per spill and 180 spills. But during data collection the time for each spill was approximately 3 seconds, resulting in about a 540 second run time. Once all of the data had been collected, JMP software was used to analyze the data, and determine the estimated mean and the standard deviation.


## Expected Count Rates

To determine what counting rates are reasonable for this detector when used to measure an operational warhead, the data from Fetter et al. were used. Fetter reported the

counting rates recorded by their germanium detector. Knowing the size ratio of their

detector and the one used for this project, An estimate of the counting rates that would be

received if the ORTEC HPGeDSSD were used to collect data from the same warhead can

be made.

# V. Results

## Chapter Overview

This chapter presents the results of the detector characterization, the statistics involved with the imaging, and the actual imaging done with the detector.

## Noise

When measuring the noise that was present on each of the channels with oscilloscopes, it was found that the charge collection strip R5 pre-amplifier output had a considerably larger noise component than any of the other charge collection strip outputs. One possible explanation to the increase in noise on this channel is that the connector for the pre-amplifier output is the closest one to the high voltage input connector. Luckily however, the extra noise on this charge collection strip readout does not degrade the energy resolution considerably. Figure 18 shows a comparison between the charge collection strip F1 noise level and the R5 strip noise level. The other eight charge collection strip channels have noises that are comparable to the F1 strip, which is shown in black.
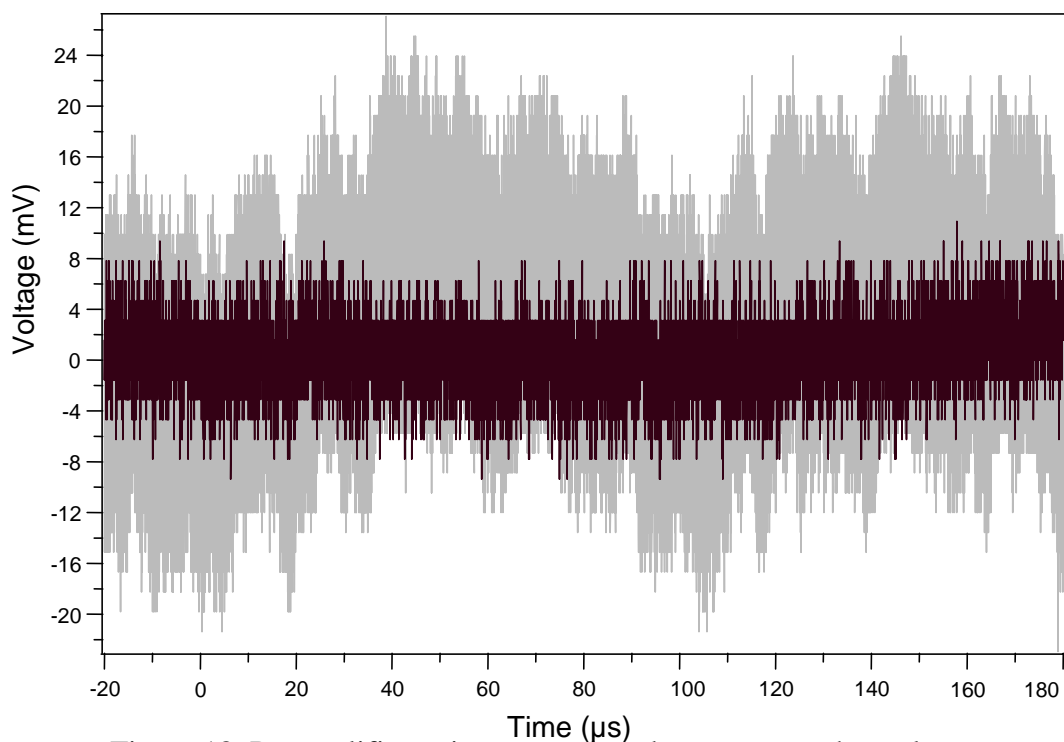
Figure 18. Preamplifier noise comparison between two channels

**Energy Calibration**

After all ten of the charge collection strips had been calibrated to the same

channel, the energy calibration had to be known in order to image within a given energy

range.  The energy of an event in raw data reported by the list mode data collection is a

channel number.  There is no setting in the software to calibrate the energy scale within

the XIA environment.  So the raw data is read into the AFIT in-house processing code as

channel numbers, and then a subroutine allows the user to manually enter the energy

calibration.  The energy calibration used for the default in the code used eight full energy

peaks, whose values of channel and energy were plotted in Excel.  Once the data was

plotted, Excel performed a least squares fit on the data and determined an equation of the

line. The plot of the data is shown in Figure 19.
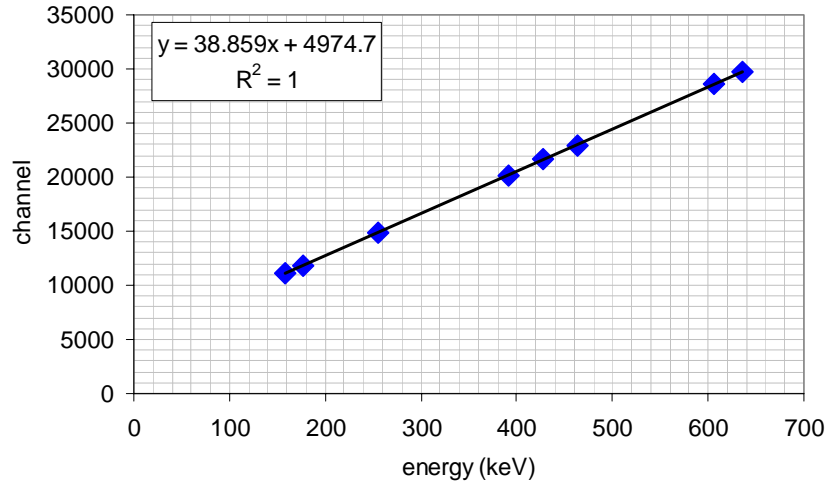


Figure 19. Energy calibration of the HPGeDSSD

The value of the least squares fit which was calculated is

$$\text{Channel} = 38.859 * (\text{Energy(keV)}) + 4974.7 \tag{6}$$

Since the data was collected in MCA mode with an energy cutoff of 5000, the data was

shifted to the right by 5000 channels. This does not occur when operating in list mode

data collection, so the energy fit used within the code is

$$\text{Channel} = 38.859 * (\text{Energy (keV)}) \tag{7}$$

**Preamplifier Decay Constant (TAU)**

One of the user inputs within the XIA software is the preamplifier decay constant.

This is used by the software to correct for pulses that occur before the previous pulse has

returned to the baseline. There is a feature in the XIA software to measure the decay

constant, but the user's manual suggests taking repeated measurements and averaging

48

them.  Each of the detectors had seven measurements taken of the decay constant, and the

values obtained as well as the averages are listed in Table 9.

Table 9. Preamplifer decay constants

| Strip | Tau 1 | Tau2 | Tau 3 | Tau 4 | Tau 5 | Tau 6 | Tau 7 | Tau Avg. |
|-------|-------|------|-------|-------|-------|-------|-------|----------|
| R4 | 29.3 | 48.1 | 44.9 | 50.1 | 47.1 | 73.8 | 26.7 | 45.7 |
| R5 | 39.9 | 49.6 | 60.5 | 55.6 | 43.3 | 46.6 | 60.3 | 50.8 |
| F5 | 30.9 | 33.1 | 62.1 | 56.1 | 38.8 | 39.1 | 42.0 | 43.2 |
| R1 | 37.1 | 44.9 | 35.4 | 42.6 | 46.9 | 46.6 | 48.2 | 43.1 |
| R2 | 33.9 | 30.1 | 59.2 | 56.6 | 36.5 | 45.5 | 53.0 | 45.0 |
| R3 | 49.0 | 40.8 | 55.6 | 47.0 | 36.8 | 50.2 | 53.6 | 47.6 |
| F1 | 40.2 | 41.8 | 57.5 | 43.2 | 59.1 | 37.8 | 52.0 | 47.4 |
| F2 | 42.2 | 45.7 | 50.3 | 55.8 | 35.5 | 36.3 | 40.7 | 43.8 |
| F3 | 45.3 | 45.0 | 35.9 | 39.5 | 38.5 | 41.3 | 28.7 | 41.8 |
| F4 | 46.4 | 54.6 | 60.3 | 44.0 | 52.2 | 61.8 | 45.6 | 52.1 |

**Charge Collection**

In order to ensure that the calculated values of the preamplifier decay constant

were correct, the preamplifier outputs were viewed on an oscilloscope.  While measuring

the decay constants it was obvious that individual interactions could be viewed if all ten

channels were viewed simultaneously.  To do this, 5 oscilloscopes with 2 channels each

were used, so that all of the preamp outputs could be viewed at the same time.  All 10

oscilloscope channels were set to trigger off of detector channel F3.  Figure 20 and 21

show coincident events between front and back strips.  Figure 20 is a preamplifier output

trace which shows a coincident event where the full energy is deposited on only two

strips, one in the front of the detector and the other in the back.  Figure 21 shows a split

between the charges for the two back strips.  However, the full energy was still collected;

as the summation of the two back strips adds to the value of the front strip.
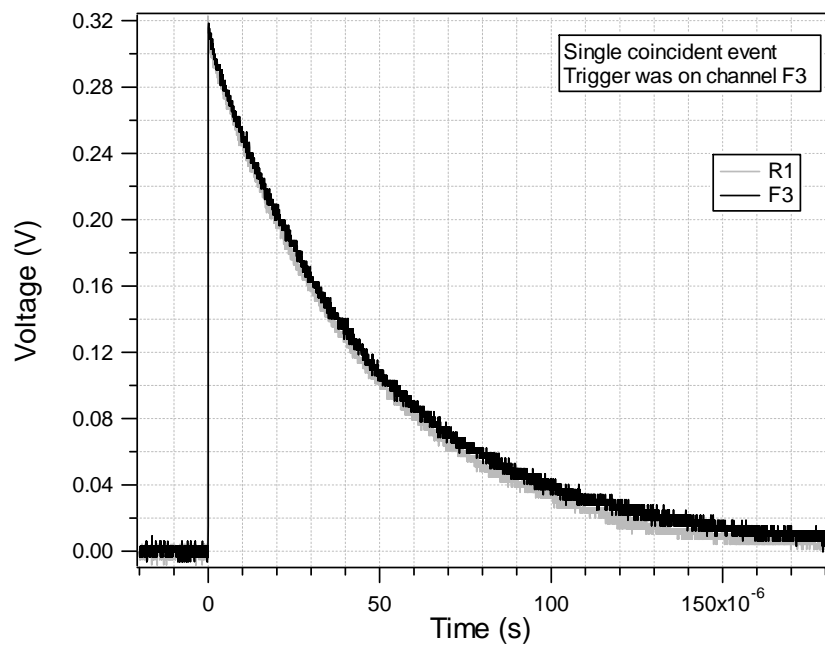
Figure 20. Preamplifier charge collection by two strips
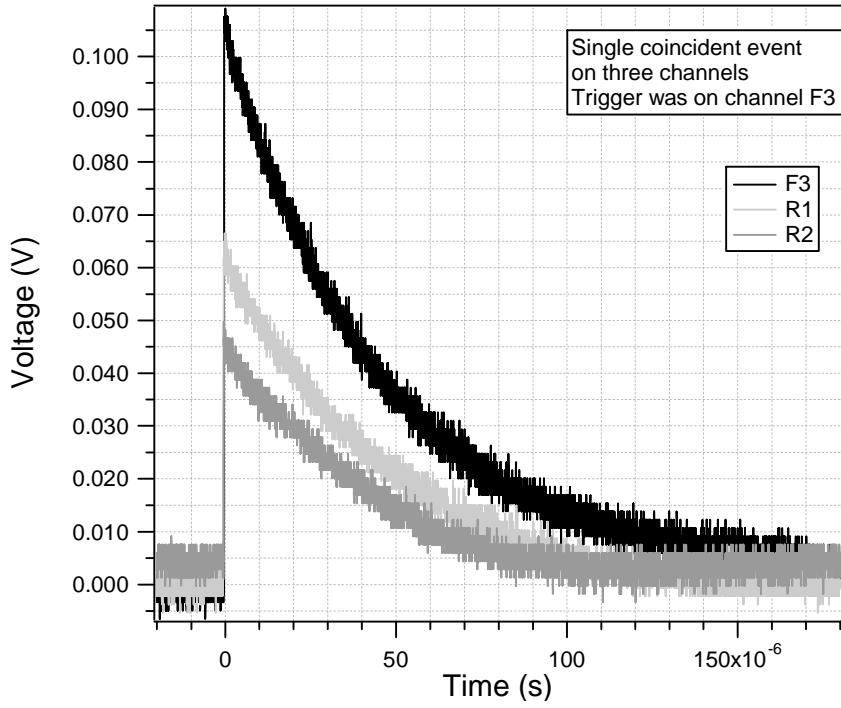


Figure 21. Preamplifier charge splitting by three strips

50

Since neither the R1 nor R2 strips in Figure 21 recorded the same energy as F1, the event that was collected on the front strip which was at the FEP, will not have a coincident event in both energy and time. Both the rear strips are coincident in time with the front strip but not coincident in energy.

**Energy Resolution of the Detector**

In order to characterize a detector, the energy resolution must be known. Chapter IV discusses the process of measuring the energy resolution. Since the HPGeDSSD has ten charge readouts, each of these must be characterized. Figures 22 and 23, are two examples of the spectra collected to measure the resolution. In fact, all ten channels were measured and the remaining eight spectra are located in Appendix C. The spectra were collected with the DGF-4C modules in the list mode used for collecting data for imaging. Even though only the energy and time information is passed back to the host computer, the DSP will still record an MCA spectrum of the data as it is being processed. The time for the data collection was 46,800 seconds. Several of the images have very poor 392 keV peaks, and a representative example of these is shown in Figure 23, and a spectra with good energy resolution is shown in Figure 22. The resolution of each of the detector strips was viewed using the Canberra Genie software which did not produce the same poor peaks obtained using the DGF modules. Since the DGF modules did not always produce these poor spectrums, it was believed that the problems were caused by the XIA software and its settings.
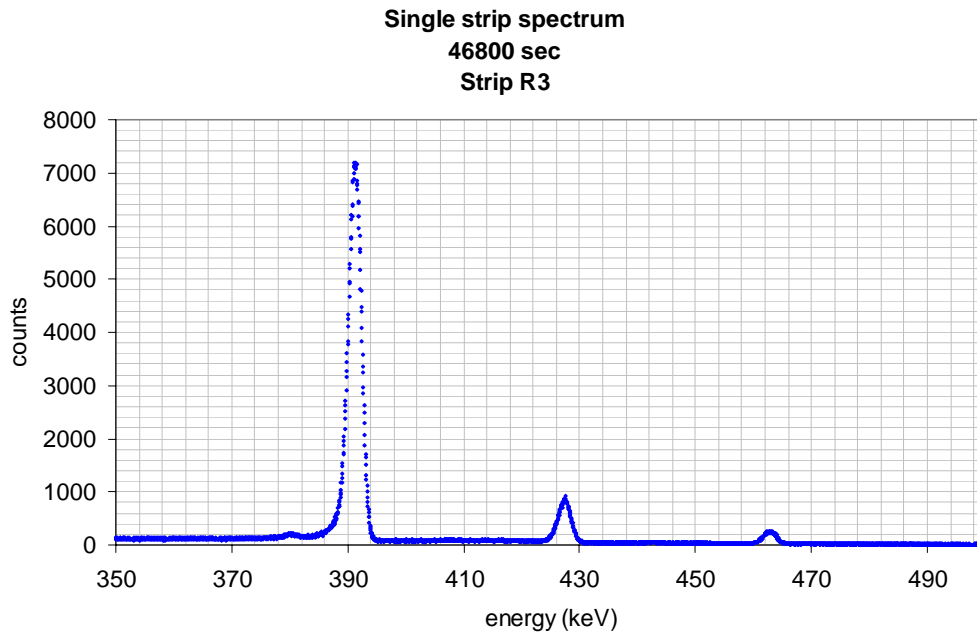
Figure 22. Strip R3 activated Sn spectrum
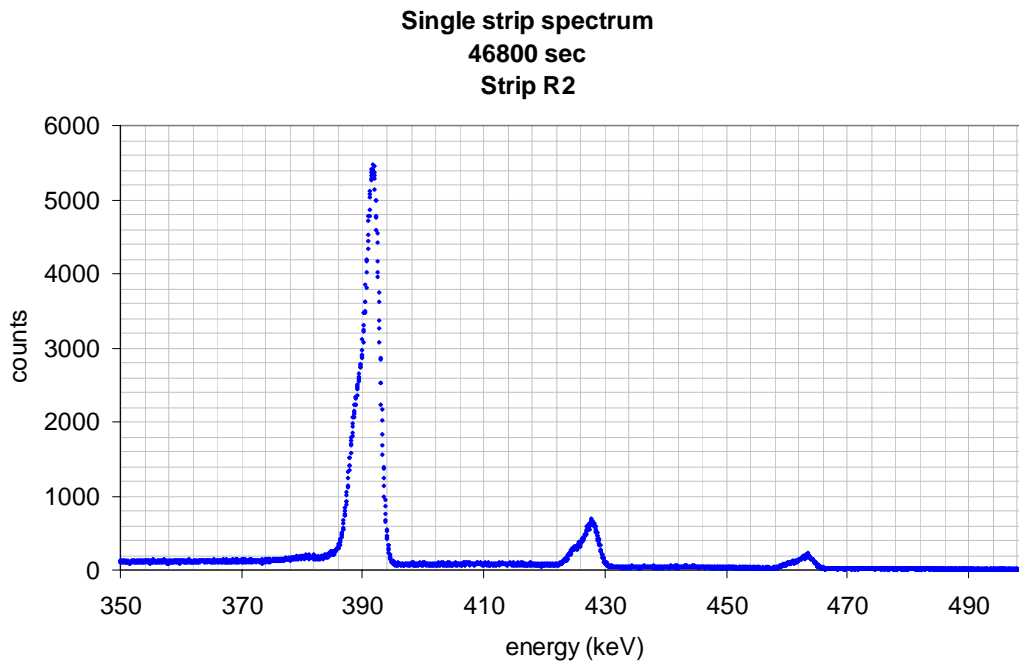


Figure 23. Strip R2 activated Sn apectrum

From each of the collected spectra, the XIA software was used to find the absolute

FWHM of each of the full energy peaks. The Pb x-ray peaks were not included in the

resolution measurements.  Figure 24 shows the FWHM measurements for all of the

detector elements.
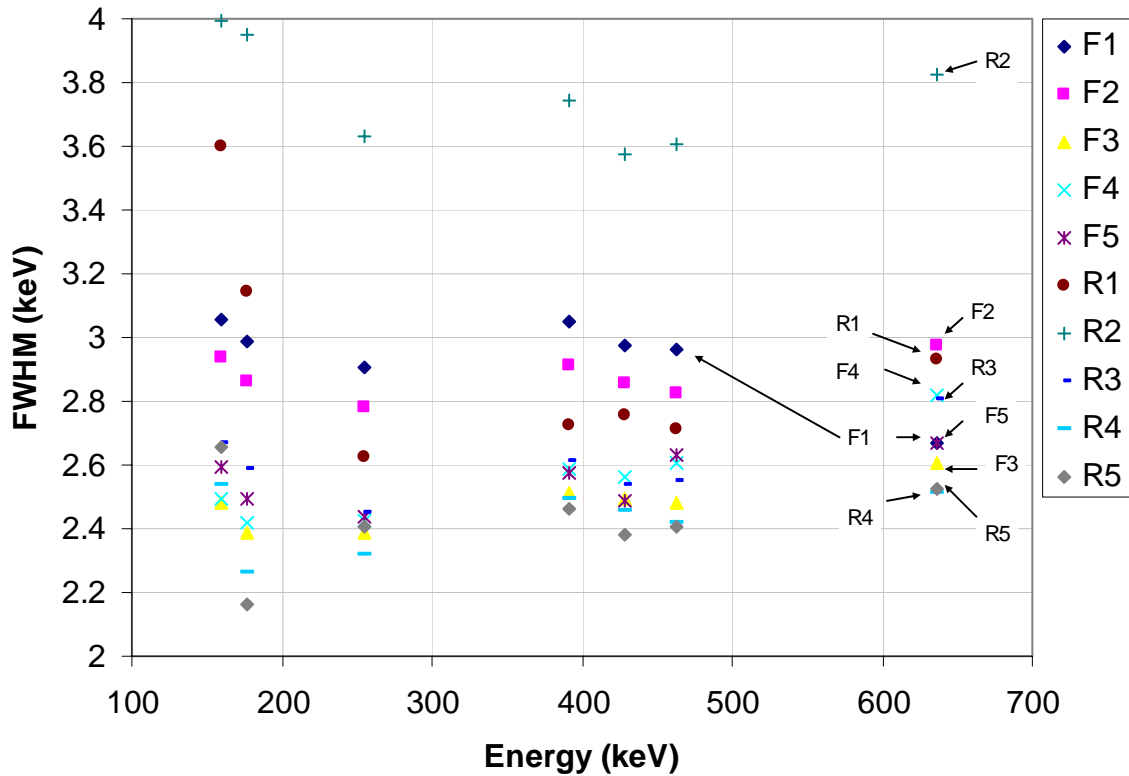


Figure 24. Plot of the FWHM vs. energy for individual charge collection strips

**Efficiency of the Detector**

The efficiency of the detector must be known if the time required to produce an

image is to be calculated.  As discussed in the procedure section, four values of the

intrinsic efficiency were measured.  Table 10 gives the values of efficiency that were

measured with the three radiation sources, and Figure 25 shows the data plotted on a log-log scale.

Table 10. Detector efficiency measurements

| Energy (keV) | Intrisic efficiency |
|---|---|
| 122 | 0.323 |
| 136 | 0.265 |
| 511 | 0.0134 |
| 662 | 0.0067 |
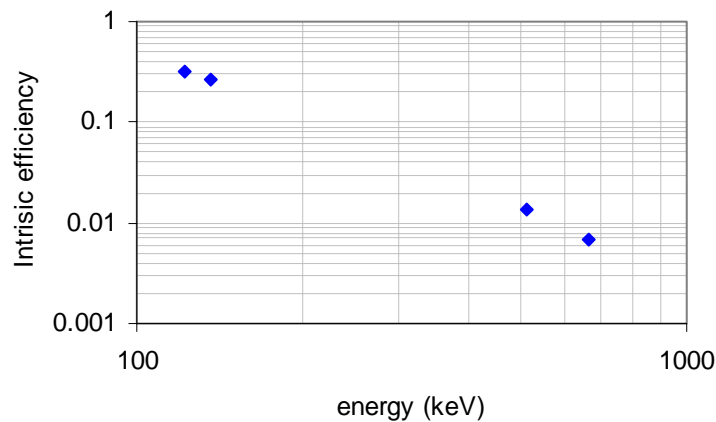


Figure 25. Detector efficiency measurements

**Variance of Individual Pixels**

To determine if the standard deviation of the number of counts recorded and processed in producing an image follows standard counting statistics, the standard deviation for a two by two image was calculated using the distributed Sn sources. With the JMP statistics package, the 30 measured values for counts recorded could be fit to a normal distribution, which estimated the mean and the standard deviation for the data. The JMP program also displays a histogram of the data set with the graph of the fit

normal distribution. The results from the JMP program for the estimates of the mean,

standard deviation, and also the standard error of the mean are shown in Table 11.

Table 11. Standard deviation of the pixel counts recorded

| Pixel | Mean | Std Deviation | Std error of Mean |
|---|---|---|---|
| 1 | 169.17 | 18.17 | 3.32 |
| 2 | 175.3 | 15.26 | 2.79 |
| 3 | 161.07 | 15.29 | 2.80 |
| 4 | 158.2 | 16.55 | 3.02 |

The histogram produced by JMP for one of the pixels is shown in Figure 26.
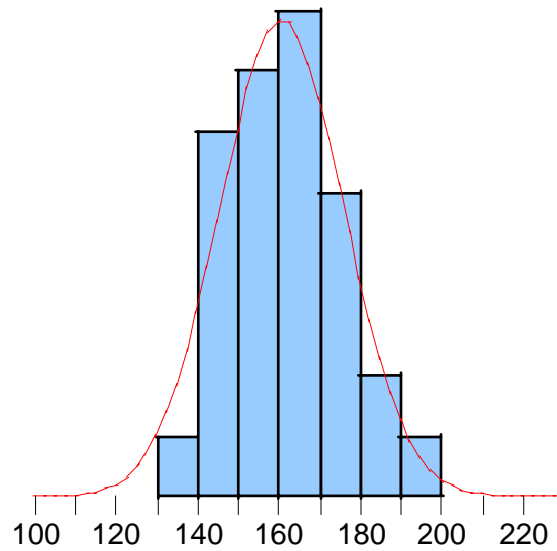


Figure 26. Histogram of pixel counts recorded and normal distribution fit

**Image Formation**

Three representative images were collected to demonstrate the performance of the

detector in distinguishing a point source from a distributed source. These three images

are shown below, and for each of the images, the results are displayed as a standard black and white image and also a three-dimensional surface plot. The first image is of a Cs137 point source that was imaged for approximately 5 minutes. The second is a uniformly distributed activated Sn source, imaged for approximately 1.5 hours. The third is an image of two activated Sn strips. The two strips in the last image were located along the left edge and right edge of the detector. Imaging time for the third image was 30 minutes. For each of the standard black and white images a solid white pixel indicates zero counts, and a solid black pixel indicates the pixel with the highest number of counts. For the second image, a photo showing the placement of the radiation sources on the collimator is included (Figure 31).
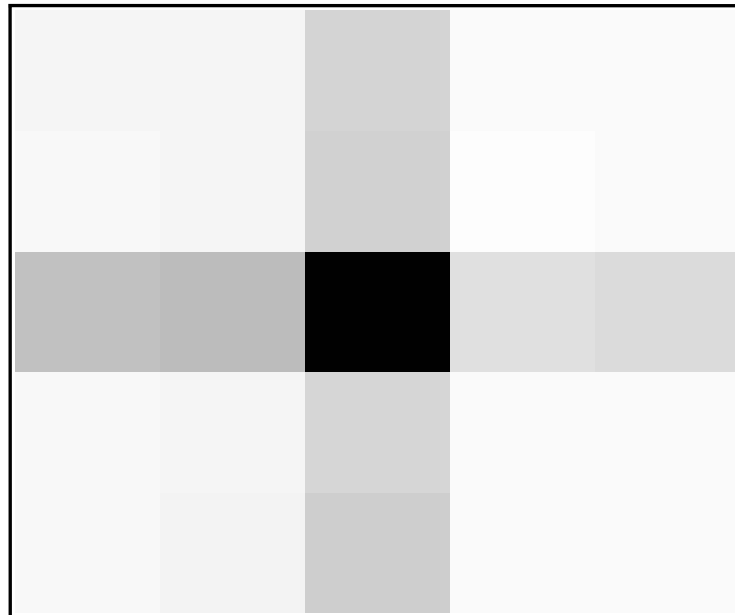


Figure 27. Cs-137 point source image plot
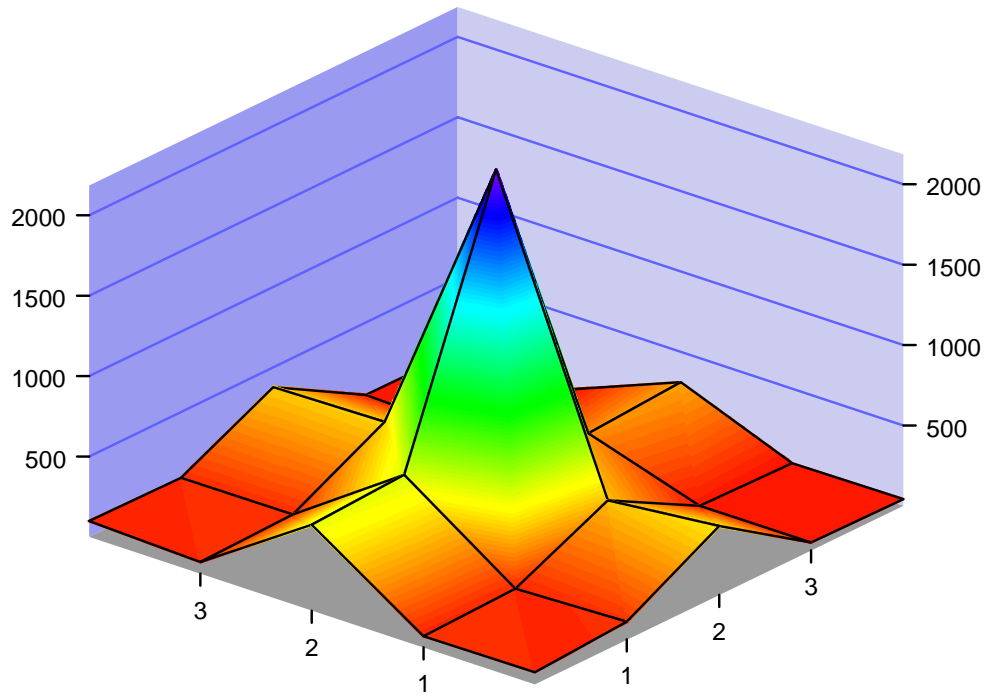
Figure 28. Cs-137 point source surface plot
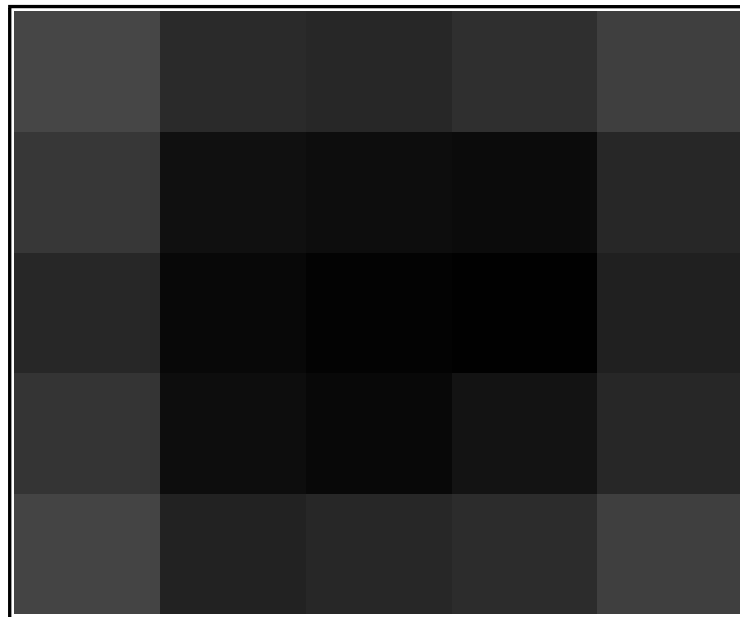


Figure 29. Distributed source image plot
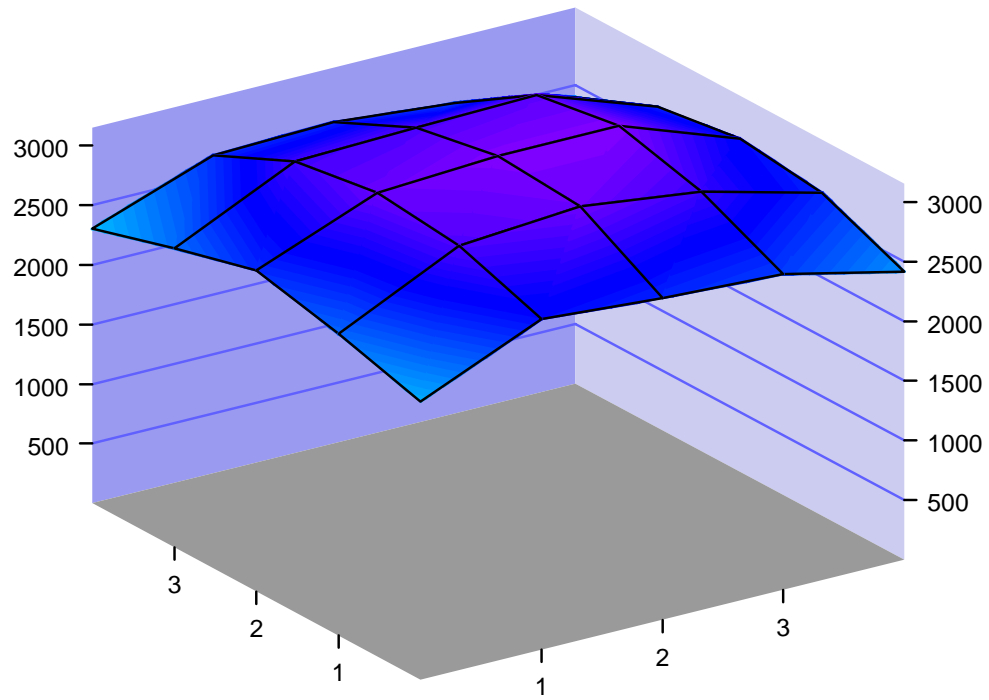
Figure 30. Distributed source surface plot
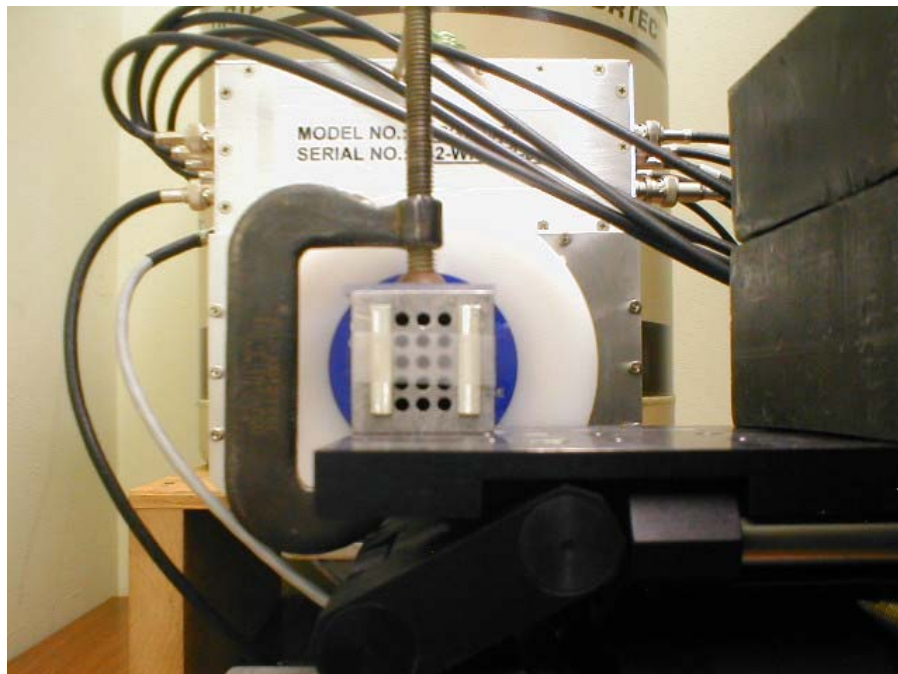

Figure 31. Source positioning for activated Sn strips

Figure 32. Two strips image plot
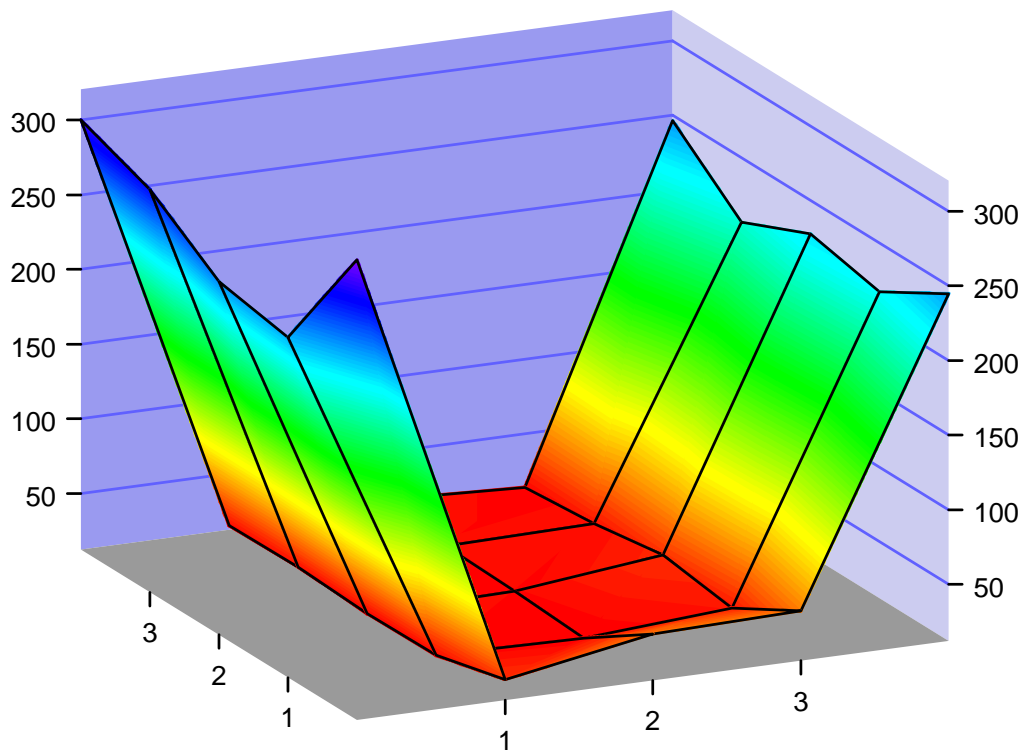


Figure 33. Two strips surface plot

**Grade Determination**

The goal for the determination of the grade of the plutonium was to use the same spectrum collected during the imaging process and then combine the spectrum from all of the strips to increase the effective size of the detector. If the energy resolution of the combined strips is less than the 3.59 keV separating the plutonium peaks for Pu-239 (645.94 keV) and Pu-240 (642.35 keV), then the grade should be obtainable [19].

Combined Spectra From All 10 Strips
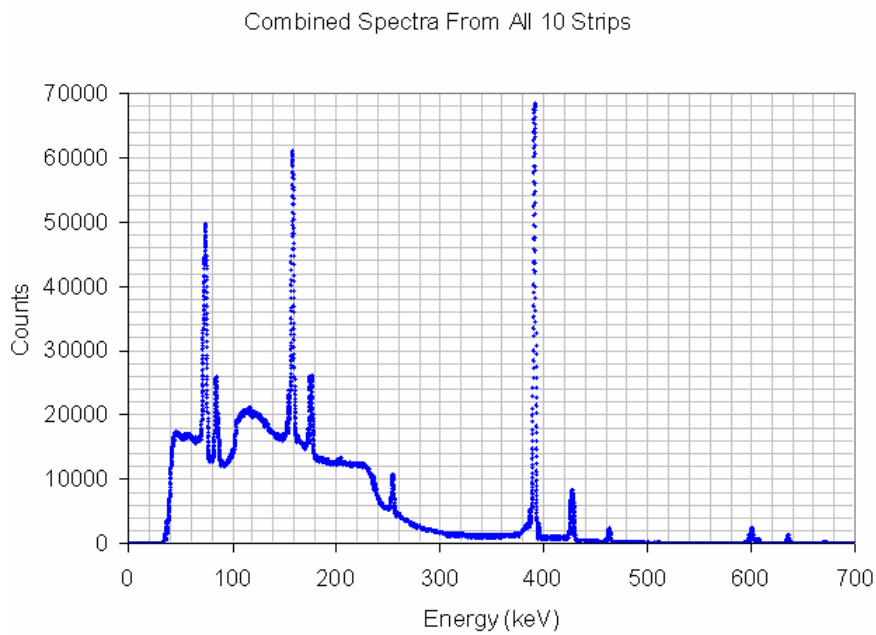


Figure 34. Sn spectrum from a combination of all 10 strips

The enlargement and analysis of the Figure 34, shown in Figure 35, demonstrates a FWHM of 2.94 keV at 602 keV when all of the charge collection strips are combined into a single spectrum. The 602.7 Sb-124 peak was no longer present when this spectrum was collected.
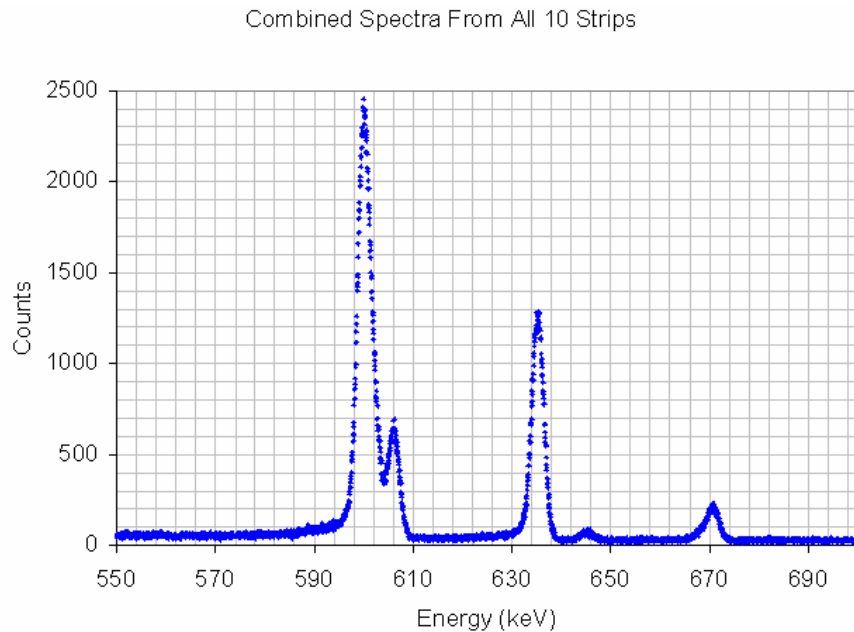
Figure 35. Combined spectrum showing energy resolution at 602 keV

## Rise Time Optimization

The resolution of the detector is dependent on both the rise time and the flat top time of the energy trapezoidal filter. The XIA documentation suggests that the resolution is relatively independent of the flat top time, and highly dependent on the rise time. It also suggests that the values of the rise time should be varied in order to determine the optimized value for the rise time for a given detector. For this project it was assumed that all of the detectors would respond similarly to changes in the rise time, so only a single strip was optimized. For the optimization of the rise time, eleven values of the energy filter rise time were used, and the FWHM of the full energy peak of Cs-137 was measured. The eleven values were plotted against the FWHM, and the minimum value was chosen as the optimum setting. Figure 36 is a chart of eleven values of the energy filter rise time plotted against the resolution at that time.

61
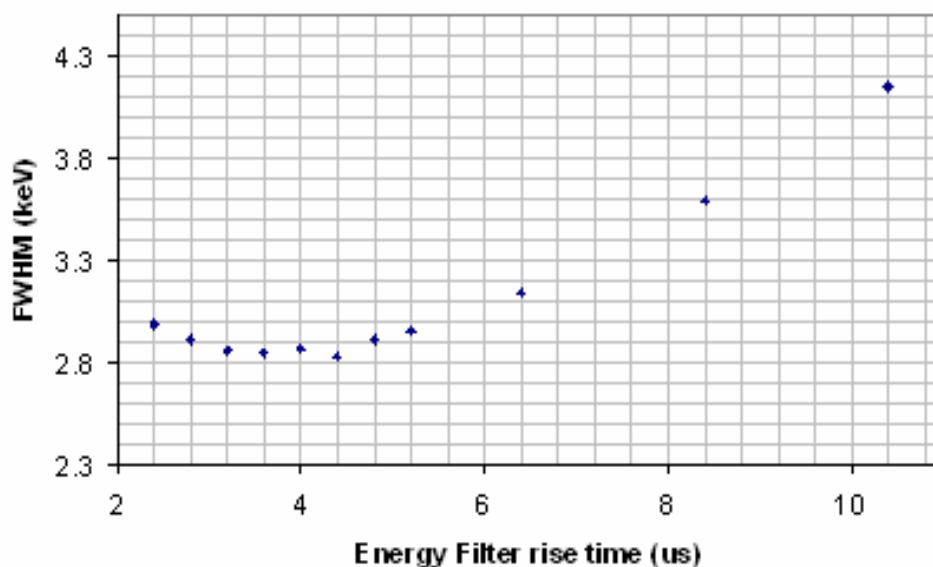
Figure 36. Filter rise time optimization

The optimum setting for the energy filter rise time was chosen to be 3.6 µs. After setting all of the channels to this value the resolutions were measured to ensure that they did not vary drastically. Since they were all in relatively close agreement, within 1 keV, the other detectors were not optimized.

**Time Required to Produce an Image at Expected Count Rates**

The detector Fetter used for the data in Table 2-1 was a HPGe detector with $27cm^2$ area, compared with the $6.25cm^2$ area for my detector. So the detector used for this research would have a count rate of about a factor of four less than the one recorded in the Fetter data. Additionally, for producing images, the count rate must again be reduced due to the multiple strips on each side. So it is reduced by another factor of 5 for

the individual strips. Combining the 375.02 keV (.862 counts/sec) and 413.69 keV

(1.582 counts/sec) count rates and correcting for the differences in detectors.

$$.862 + 1.582 = 2.444 \qquad (8)$$

$$\frac{2.444}{4} = .611 \qquad (9)$$

$$\frac{.611}{5} \approx .1222 \qquad (10)$$

So, using this detector to image in these two energy peaks, about .1222 counts per second

will be recorded on each of the strips. Assuming a 30-minute counting time, 220 counts

will be recorded.
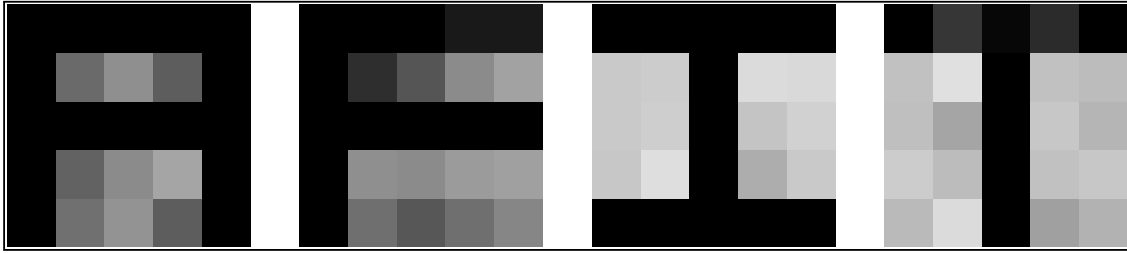
Figure 37. Combination of four images to spell AFIT

Figure 37 was created by combining four images that were produced by placing the radiation sources in a configuration that formed the letter A, F, I, and T.  An example of the placement of the radiation sources for the T is shown in Figure 38.
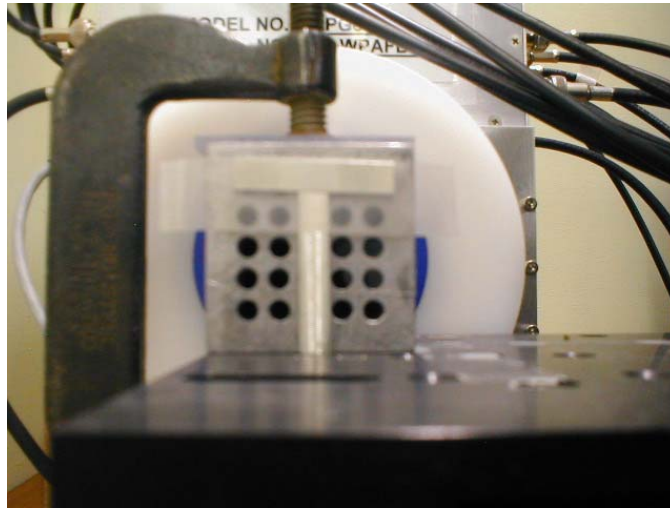

Figure 38. Source positioning for activated Sn strips to produce the AFIT "T"

**Chapter Overview**

The principle objective of this project was to setup and characterize the high purity germanium strip detectors performance, and demonstrate and characterize an imaging capability.  Additionally, the ability of the system to determine the grade of the plutonium was desired.  The XIA software version 3.02 that was used for the majority of

this project was inadequate. It did not produce the correct signals in the busy-out connector, so that the modules could not be started and stopped simultaneously. The XIA Company provided a test copy of a new version, 3.04, which was tremendously improved, but was received late in the project. The diode fan-in / fan-out system did not function as desired once the new software was installed, and the modules were sending the correct logic pulses, so the modules could still not be operated simultaneously.

**Detector Performance**

The performance of the detector was essentially what was expected for a high purity germanium detector. The resolution of the charge collecting strips was slightly poorer (larger) than a typical HPGe detector, but the efficiency measurements were in agreement with expectations for a detector of this size.

**Resolution**: The resolution measurements made using the XIA modules were sometimes in line with expectations, and were sometime very far off, this was probably due to the software settings used to optimize the resolution and efficiency. In order to verify that the problem lied within the XIA hardware/software settings, all of the resolutions were measured using Canberra MCAs and the Genie processing software that was available. Table 12 is a table of the resolutions reported by the manufacturer; the measured resolutions using the XIA hardware when the settings were optimized and when they were not, as well as the resolutions measured using the Canberra MCAs. It was assumed that all of the settings for optimization were identical for all ten strips.

However, strip R2 had a resolution about 60% worse than the other strips. This could be due to a bad strip or due to non-optimized settings for this strip. Further investigation is needed to determine if all of the strips are optimized with the same settings.

Table 12. FWHM of detector

| Strip | Manufacturer Reported FWHM ( 122 keV) | XIA optimized settings (635 keV) | XIA Not optimized (635 keV) | Canberra MCA FWHM (662 keV) |
|-------|------|------|------|------|
| F1 | 1.64 | 3.13 | 5.82 | 3.5 |
| F2 | 1.62 | 2.98 | 11.1 | 4.0 |
| F3 | 1.23 | 2.61 | 2.86 | 2.9 |
| F4 | 1.31 | 2.83 | 2.43 | 3.0 |
| F5 | 1.28 | 2.68 | 2.42 | 2.8 |
| R1 | 1.48 | 2.94 | 2.38 | 2.9 |
| R2 | 1.50 | 3.82 | 7.34 | 3.6 |
| R3 | 1.41 | 2.80 | 2.92 | 2.8 |
| R4 | 1.40 | 2.53 | 2.51 | 3.0 |
| R5 | 1.41 | 2.55 | 2.17 | 2.9 |

**Efficiency**: The NRL has a 45mm x 45mm x 10mm germanium strip detector, with an efficiency of 2% at 662 keV. Comparing this with the detector used in this project. The volume difference between detectors is

$$\frac{AFIT \text{detector volume}}{NRL \text{detector volume}} = \frac{5625 mm^3}{20250 mm^3} = 0.278 \tag{11}$$

Therefore, a quick estimate of the expected efficiency for this system would be 0.278 times the one reported by the NRL.

$$0.278 \cdot 0.02 = .0056 \tag{12}$$

This is in very close agreement with the measured value of 0.00667 discussed in chapter four. Also the plot of efficiency as a function of energy is what was expected.

**Imaging Artifacts**

There are three artifacts of interest in the images shown in Figures 27, 29, and 32. Figure 27 is an image of a point source of radiation, and an artifact of an increased number of counts in all pixels located on the two strips which measured the point source is seen. Figure 29 is an image of a uniformly distributed point source, and an artifact of a decreased number of counts around the edges is observed. Finally Figure 32 is an image of two strips and has an artifact of an increased number of counts on the bottom row of the image.

The artifact in image 29 appears to be a result of the imperfect collimation. The strips on the edges are only irradiated by the source in front of the strip and the cross talk from a single adjacent strip. While all the center strips are irradiated by the source in front of the strip and the cross talk from two adjacent strips. The artifact is symmetric and a collimator with a larger amount of septa between the holes would reduce this artifact. However, if the septa were increase for a parallel hole collimator the number of photons that could reach the detector would decrease. This can be solved by using a diverging collimator. This allows the same size hole diameter to be used, and increases the amount of septa between the holes at the side of the collimator closest to the source.

The artifact in image 27 appears to be a result of the electronics. Since the number of counts in the artifact pixels do not decrease with the distance from the source it does not appear to be due to collimation. However, further investigation is needed to determine if the source were placed in a corner pixel, if the counts in the artifact pixels would still remain constant of if they would decrease with increasing distance from the source.

Finally, the artifact in Figure 32 seems to show that the number of counts in a pixel is dependent on which module it is in.  The center three columns in Figure 32 should be symmetric.  However, the bottom row shows an increased number of counts. The top four rows are all on module three, and the bottom row is on module two.  This artifact is probably due to uncertainties in module start and stop times.  When the fan-in / fan-out system is used properly, this artifact should be eliminated.


**Imaging Performance**

With the parallel hole collimator used for the system demonstration, it was only possible to image a 25mm by 25mm area.  In an operational system a diverging collimator should be used to increase the size of the area that is imaged.  The parameters needed for the design of this collimator will have to come from personnel familiar with weapons inspections.  An estimate of the distance from source to collimator must be known, as well as the expected dimensions of the source.  Using the data in the Fetter model of an implosion weapon, the area that would be covered by each pixel using this detector is shown below.  Each pixel would have to cover a 2cm x 2cm area.
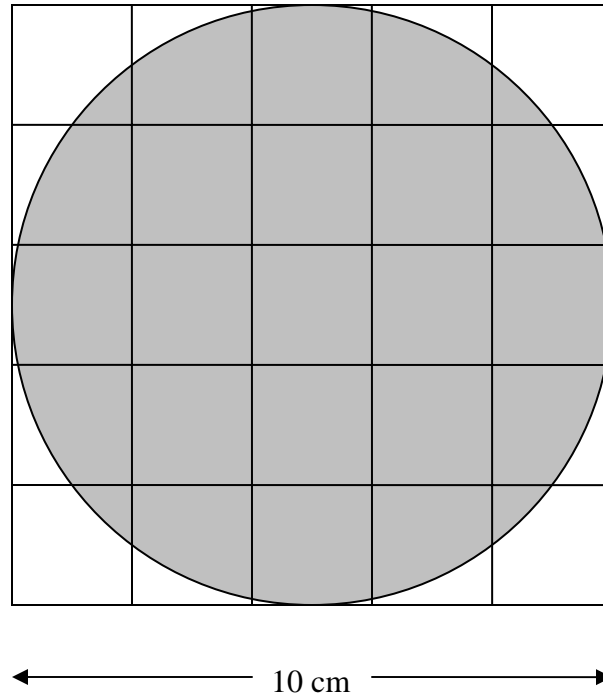
Figure 39.  Individual pixel coverage for a Pu source

It was believed that the number of counts recorded within an individual pixel would follow a Gaussian distribution, which is what should occur for counting statistics. However, with the measurements of the distribution and the fit within the JMP program, this was not the case.  One possible explanation is that this was due to the faulty XIA software, or it could have been due to the fact that a single interaction is not guaranteed to get recorded as a full energy event at both sides of the detector.  This test should be performed again once the module are started and stopped simultaneously.

I was not able to find any information on the times that are allowed for spectrum collection during a weapons inspection.  However, assuming that the times used by Fetter when collecting the spectrum from the Soviet warhead are close to the time allowed during inspections, it can be concluded that a 30 minute data collection will be all that is

allowed.  This is enough time to produce an image of the pit since, as equations 8-10 showed in Chapter V, 220 counts will be recorded during this time.


**Grade Determination**

To determine the weapons grade of the plutonium in a weapon that has been imaged the ratio of the Pu-239 645.94 keV peak and the Pu-240 642.35 keV peak must be used along with the branching ratios to find the relative amounts of the two isotopes [19]. To minimize the time required for an inspection, the spectrum collected during imaging should be used if possible.  This appears to be an achievable goal.  When all ten detector readouts are combined into a single spectrum, the energy resolution in the 600 keV range is 2.92 keV.  Theoretically, this will easily allow for the ratio of the two peaks that are 3.59 keV apart.  It was shown that two peaks, 6 keV apart are distinctly separated. However, the two peaks in the 600 keV range may not be available during an inspection; in the spectrum collected by Fetter, the Pu-240 peak of 642.35 keV was not visible.  So either two other peaks must be used or the absence of the Pu-240 peak can be taken as an indicator that the material is WGPu.  It is important to note that even if the spectra collected during imaging is not used for grade determination, a separate measurement can be taken using different processing electronics and the grade could certainly be determined.  Another advantage to this approach, is that the collimator can be removed, which will increase the number of counts recorded for a given time.

**Recommendations for Future Work**

The standard interface between the XIA software and the DGF-4C modules does not allow the data to be downloaded to the host computer until the entire data set has been collected. It would have been extremely helpful if the data could have been transferred to the host computer at near real time. This would have allowed an image to be processed and built up over time and the operator could determine when enough data had been collected and stop data collection. Currently, as estimate of the time required must be made and the data collected, followed by image processing to determine if there was indeed enough time.

More important than the convenience of processing the data in near real time is the ability to get sub pixel resolution for the charge collecting strips. While having discussions with the NRL, they said that it was possible to determine which portion of the charge collecting strip actually collected the charge by viewing the induced noise of the two strips adjacent to the one that collected the charge. This would allow an image array of 10 x 10, if the strips were only divided in half, instead of the current 5 x 5. The advantage to processing the data in this way to increase the resolution as opposed to simply creating a strip detector with 10 strips on each side is that the increase in hardware is minimal with this method, and if the system is to be used for treaty monitoring, portability is a concern and hardware must be minimized.

Also, the OR gate from ORTEC needs to be installed in the system to complete the busy-synch loop. This will allow all the module to be stated and stopped simultaneously, which will possibly eliminate some of the artifacts seen in the images.

**Summary**

The ability to use a HPGeDSSD as an imaging device to detect some spoof weapons has been demonstrated. Due to software problems the data collection parameters were not ideal, yet the imaging system was clearly able to distinguish point sources from distributed sources, while collecting spectral information at the same time. In addition to demonstrating the imaging capability, it was also shown that the two peaks most likely to be used for determining the grade of the plutonium are far enough apart in energy to be resolved by the combined spectrum of all ten detector channels with a good energy calibration.

## Appendix A.  Processing Code

The AFIT in house processing code was written using Compaq Visual Fortran version 6.6, in FORTRAN 95.  The program uses a modular structure which should make it easy for an operator to use or modify.  This appendix contains all of the processing code, beginning with the main program code followed by the modules which contain the subroutines.  Inputs to the code are; the location of the raw data file from data collection, the equation of the line for the energy fit of the detectors when the data was taken, and the desired energy for imaging.  The code outputs an array which displays the counts obtained for each pixel in the energy range of interest, and also opens the array visualizer and graphically displays a grayscale image of the counts recorded.


**Program ThesisCode**

```
! This program reads in data that was created using the XIA/IGOR software, and creates
! a 5 x 5 array of pixel counts for a image of a desired energy .
! It is intended to be used for data that was collected by a HPGeDSSD that has
! 5 strips on each side.
! The only inputs required for code execution are the equation of the line for
! an energy fit, the desired energy for imaging and the data location of the file that
! was created by the XIA/IGOR software of the collection data.

Use Coincidence
Use Sorter
Use Inversion
Use Readdata
Use Fit
Use display

Implicit None

Integer :: Counter
Integer :: spills
Integer, Dimension(5,5) :: matrix
Integer, allocatable, Dimension (:) :: rawdata
```

Integer, allocatable, Dimension (:,:) :: sorteddata
Character(len=1) :: answer

! This subroutine calculates the energy desired from a energy fit.
! the energy fit can also be hardcoded here
Call energyfit(energy)

! This subroutine takes a filename as an input from the user and reads the
! file into the rawdata array.  The subroutine also calculates the number
! of spills that were used during the data collection by the XIA modules
Call dataread(rawdata,spills)

! This subroutine organizes the raw data into a (4,n) array,
! the four columns contain the following information for each event
!              module        channel              energy        time
!              ...            ...                  ...                     ...
!              module        channel              energy        time
Call Inverter(counter,rawdata,spills)


! This subroutine will eliminate any events that don't fall within the
! specified energy range and will then sort
! out the events by time with the lowest time in the first row and the
! largest time in the last row
Call Sort(counter,iter,sorteddata,energy,spills)

! This subroutine sorts through the organized data to find events that were
! coincindent in both time and energy (these coincidence windows can be changed
! within the subroutine.  Then the subroutine displays an array of pixel counts
! for the specified energy for imaging and also writes the array to a file
Call Coin (iter,sorteddata,matrix)

! this subroutine passes in the matrix of pixel values and uses the array
! visualizer to display them in an image.
Call visualizer (matrix)

! pauses code execution so that the image stays displayed
Print *, 'Press any key to continue'
Read (*,*)

End Program ThesisCode

**Module Fit**

Implicit None

```
Contains
Subroutine EnergyFit (energy)
Integer, Intent(Out) :: energy
Integer :: keV
Integer :: m     ! this is the slope of the energy fit line
Integer :: b     ! this is the y intercept of the energy fit line
Character(len=1) :: answer
! Default slope and y intercept
m = 15
b = 0
Print *, 'The current equation of the line is '
Print *, 'channel =',m,'*energy(keV) +',b
Print *, ' '
Do
        Print *, 'Would you like to update the fit? '
        Print *, 'Y, y, N, n'
        Read (*,*) answer
        If (answer=="Y".or.answer=="y".or.answer=="N".or.answer=="n") Then
        Exit
        End If
End Do
If (answer=="y".or.answer=="Y") Then
Print *, ' '
Print *, 'Enter new slope of the line'
Read (*,*) m
Print *, ' '
Print *, 'Enter new y intercept'
Read (*,*) b
End If
Print *, ' '
Print *, 'Enter desired energy for image in keV'
Read (*,*) kev
Print *, ' '
energy = m*keV + b   !actually channel =m*keV + b
energy = 10116
Print *, 'channel of interest = ',energy
End subroutine energyfit
End Module fit
```

**Module Readdata**

```
Implicit None

Integer :: number
Integer :: status
```

```fortran
Character(len=20) :: filename
real :: value
Integer :: nvals

Contains

Subroutine dataread(rawdata,spills)
Integer, allocatable, Dimension (:), Intent(out) :: rawdata
Integer,Intent(out) :: spills
! Open the file of raw data read it in and close the file
! Stephen J. Chapman's Fortran 90/95 pg 194
Write (*,*) 'Please enter input file name: '
read (*,*) filename
Print *, ' '
Open (Unit=30,File=filename,Action='Read',Iostat=status)
openif: If (status == 0) then
readloop: Do
        Read(30,*,Iostat=status) value
        if (status /= 0) exit
        nvals = nvals + 1
        1010 FORMAT (' ','Line ', I8, ': Value = ',F10.4)
End Do readloop
readif: If (status > 0) Then
Write (*,1020) nvals +1
1020 format('0','an error occurred reading line ', I8)
Else
Write (*,1030) nvals + 1
1030 Format ('0','End of file reached.  there were ', I8,' values in the file.')
Print *, ' '
End If readif
Else openif
write(*,1040)status
1040 format(' ','error opening file: iostat = ',I8)
End if openif
Close (30)
Open (30, file=filename)
! allocate the size of the raw data array
! the raw data is a text delimited file that was created in the IGOR program
allocate(rawdata(nvals))
! read in the raw data
Read (30,*) rawdata
Close (30)

! calculate the number of spills that were used for the data collection
!        a spill is a data collection of up to 744 events
```

```
spills = 0
number = 1
Do
        number=rawdata(number)+number
        spills = spills + 1
        If (number == nvals+1) exit
End Do
spills = spills / 3
Print *, 'Data read successful'
Print *, ' '
End subroutine dataread
End Module readdata
```

## Module Inversion

```
Implicit None
Integer :: i,j,iter,k,p,Q
Integer :: largest
Integer :: spillcount
Integer, allocatable, Dimension (:) :: length
Integer, allocatable, Dimension (:) :: length2
Integer, allocatable, Dimension (:) :: modulenumber
Integer, allocatable, Dimension (:,:) :: modu
Integer, allocatable, Dimension (:) :: spillevents
Contains
Subroutine Inverter(counter,rawdata,spills)
Integer, Intent(in) :: spills
Integer, Intent(out) :: counter
Integer, Dimension(:), Intent(in) :: rawdata
! Open the file that will store the results of this subroutine
Open (40, file='formatteddata.csv')
Open (77, file='Spill data.csv')
! reformat the data for modules 1,2,and 3
! lengths
!allocate size for the length of the array that will hold the data lengths
Allocate(length(spills*3))
Allocate(length2(spills*3))
length2(1)=0
! length of each data set
Allocate(modulenumber(spills*3))
Allocate(spillevents(spills*3))
! determine all of the lengths of the data sets and also assign the number of the
! module that the data came from into the module number array.  This must be done
! three times for each of the number of spills, since each spill contains data from
! three modules.
```

```fortran
length(1)=rawdata(1)
modulenumber(1)=rawdata(2)
counter = 0
Do i = 2,spills*3
j=i
length(i) = 1
Do while (j.ne.1)
length(i) = length(j-1) + length(i)
j=j-1
End Do
modulenumber(i) = rawdata(length(i)+1)
length2(i) = length(i)
length(i) = rawdata(length(i))
End Do
Print *, 'lengths detemined'
! this is just set at an arbitrarily high number if a array bounds exceeded error
! occurs during code execution.  Suspect that this is too low
largest = 6000
allocate(modu(3*spills,largest))
modu=0
! the modu array contains all of the raw data in the following format
! col 1              col 2              col 3              col 4                      .......
! module1 spill1  module2 spill1  module3 spill1     module1 spill2          .......
!       Hit pattern
!       not used
!       not used
!       time
!       energy
!       Hit pattern
!                 .
!                 .
!                 .
Do j = 1,3*spills
Do i = 1,length(j)-6
If (j==1)Then
modu(j,i)=rawdata(i+length2(j)+6)
Else
modu(j,i)= rawdata(i+length2(j)+5)  !length(j-1)
End If
End Do
End Do
Deallocate (length)
Deallocate (length2)
Print *, 'modu done'
j=0
```

```
Do k = 1,spills
Do p = 1,3
j=j+1
i=1
spillcount=0
Do
If (modu(j,i)==0)exit
If (modu(j,i)==1) Then
!                          mod number   channel                time   energy
Write (40,*) modulenumber(j),',',"0",",",modu(j,i+3),",",modu(j,i+4),',',k
i=i+5
counter=counter+1
spillcount=spillcount+1
Else If (modu(j,i)==2) Then
Write (40,*) modulenumber(j),',',"1",",",modu(j,i+3),",",modu(j,i+4),',',k
i=i+5
counter=counter+1
spillcount=spillcount+1
Else If (modu(j,i)==3) Then
Write (40,*) modulenumber(j),',',"0",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"1",",",modu(j,i+5),",",modu(j,i+6),',',k
i=i+7
counter=counter+2
spillcount=spillcount+1
Else If (modu(j,i)==4) Then
Write (40,*) modulenumber(j),',',"2",",",modu(j,i+3),",",modu(j,i+4),',',k
i=i+5
counter=counter+1
spillcount=spillcount+1
Else If (modu(j,i)==5) Then
Write (40,*) modulenumber(j),',',"0",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"2",",",modu(j,i+5),",",modu(j,i+6),',',k
i=i+7
counter=counter+2
spillcount=spillcount+1
Else If (modu(j,i)==6) Then
Write (40,*) modulenumber(j),',',"1",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"2",",",modu(j,i+5),",",modu(j,i+6),',',k
i=i+7
counter=counter+2
spillcount=spillcount+1
Else If (modu(j,i)==7) Then
Write (40,*) modulenumber(j),',',"0",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"1",",",modu(j,i+5),",",modu(j,i+6),',',k
Write (40,*) modulenumber(j),',',"2",",",modu(j,i+7),",",modu(j,i+8),',',k
```

i=i+9
counter=counter+3
spillcount=spillcount+1
Else If (modu(j,i)==8) Then
Write (40,*) modulenumber(j),',',"3",",",modu(j,i+3),",",modu(j,i+4),',',k
i=i+5
counter=counter+1
spillcount=spillcount+1
Else If (modu(j,i)==9) Then
Write (40,*) modulenumber(j),',',"0",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"3",",",modu(j,i+5),",",modu(j,i+6),',',k
i=i+7
counter=counter+2
spillcount=spillcount+1
Else If (modu(j,i)==10) Then
Write (40,*) modulenumber(j),',',"1",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"3",",",modu(j,i+5),",",modu(j,i+6),',',k
i=i+7
counter=counter+2
spillcount=spillcount+1
Else If (modu(j,i)==11) Then
Write (40,*) modulenumber(j),',',"0",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"1",",",modu(j,i+5),",",modu(j,i+6),',',k
Write (40,*) modulenumber(j),',',"3",",",modu(j,i+7),",",modu(j,i+8),',',k
i=i+9
counter=counter+3
spillcount=spillcount+1
Else If (modu(j,i)==12) Then
Write (40,*) modulenumber(j),',',"2",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"3",",",modu(j,i+5),",",modu(j,i+6),',',k
i=i+7
counter=counter+2
spillcount=spillcount+1
Else If (modu(j,i)==13) Then
Write (40,*) modulenumber(j),',',"0",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"2",",",modu(j,i+5),",",modu(j,i+6),',',k
Write (40,*) modulenumber(j),',',"3",",",modu(j,i+7),",",modu(j,i+8),',',k
i=i+9
counter=counter+3
spillcount=spillcount+1
Else If (modu(j,i)==14) Then
Write (40,*) modulenumber(j),',',"1",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"2",",",modu(j,i+5),",",modu(j,i+6),',',k
Write (40,*) modulenumber(j),',',"3",",",modu(j,i+7),",",modu(j,i+8),',',k
i=i+9

```
counter=counter+3
spillcount=spillcount+1
Else If (modu(j,i)==15) Then
Write (40,*) modulenumber(j),',',"0",",",modu(j,i+3),",",modu(j,i+4),',',k
Write (40,*) modulenumber(j),',',"1",",",modu(j,i+5),",",modu(j,i+6),',',k
Write (40,*) modulenumber(j),',',"2",",",modu(j,i+7),",",modu(j,i+8),',',k
Write (40,*) modulenumber(j),',',"3",",",modu(j,i+9),",",modu(j,i+10),',',k
i=i+11
counter=counter+4
spillcount=spillcount+1
Else
Print *, 'Found an invalid hit pattern'
End If
End Do
write (77,*) spillcount
End Do
End Do
Close (40)
Close (77)
Deallocate (modulenumber)
Deallocate (modu)
Open (77, file='Spill data.csv')
Read (77,*) spillevents
Open (79, file='spills.csv')
Do i = 1,3*spills,3
Write (79,*) spillevents(i),',',spillevents(i+1),',',spillevents(i+2)
End Do
Close (79)
Deallocate (spillevents)
Print *, 'Inversion Complete'
Print *, ' '
End Subroutine Inverter
End Module Inversion
```

**Module Sorter**

```
Implicit None
Integer :: k,i,q
Integer :: M
Integer, dimension(1) :: Min
Integer, allocatable, dimension(:,:) :: array
Integer, allocatable, dimension(:,:) :: array2
Integer, allocatable, dimension(:) :: times
Contains
Subroutine Sort(counter,iter,sorteddata,energy,spills)
```

```fortran
Integer,Intent(In) :: Counter
Integer,Intent(In) :: spills
Integer, Intent(In) :: Energy
Integer,Intent(Out) :: iter
Integer, allocatable, dimension(:,:),Intent(Out) :: sorteddata
Integer, allocatable, dimension(:,:) :: spillevents
Open (15,file='formatteddata.csv')
Open (75,file='temporary.csv')
Open (17,file='fep.csv')
Open (84, file='spills.csv')
k=Counter
allocate(spillevents(3,spills))
allocate(array(5,Counter))
allocate(sorteddata(4,Counter))
array=0
Read (15,*) array
Read (84,*) spillevents
Close (15)
iter = 0
! pick out the events in the desired energy range
Do i = 1,Counter ! all added
If (array(4,i)>=energy-75.and.array(4,i)<=energy+75) Then
Write (17,*) array(1,i),',',array(2,i),',',array(3,i),',',array(4,i)
iter=iter+1
End If
End Do
Close (17)
deallocate (spillevents)
deallocate (array)
allocate(array2(4,iter))
allocate(times(iter))
open (17,file='fep.csv')
! this reads the events that were in the full energy peak into array2
read (17,*) array2
! this stores all of the time information into a single array
Do i = 1,iter
times(i) = array2(3,i)
End Do
! this finds that maximum time
M=MaxVal(times)
Do i=1,iter
! finds the location of the minimum time within the array
Min=MinLoc(times)
! stores the value of the minimum location into Q
Q=Sum(Min)
```

```
! store the array2 row for the minimum time into temp
sorteddata(:,i)=array2(:,q)
! increase the value of the time to larger than the greatest
Times(q)=M+1
End Do
Deallocate (times)
Do i = 1,100
Write (75,*) sorteddata(1,i),',',sorteddata(2,i),',',sorteddata(3,i),',',sorteddata(4,i)
End Do
Print *, 'Sorter Complete'
Print *, ' '
Deallocate (array2)
End Subroutine Sort
End Module sorter
```

**Module Coincidence**

```
Implicit None
Integer :: i
Integer :: j
Integer :: k
Integer :: time
Integer :: energy
Integer :: check
Integer,allocatable, Dimension (:,:) :: formatteddata
Integer,allocatable, Dimension (:,:) :: events
Contains
Subroutine Coin (counter,sorteddata,matrix)
Integer, Intent(In) :: counter
Integer, Dimension (:,:),Intent(In) :: sorteddata
Integer, Dimension(5,5), Intent (Out)  :: matrix
allocate(formatteddata(5,counter))
formatteddata=0
formatteddata(1,:)=sorteddata(1,:)
formatteddata(2,:)=sorteddata(2,:)
formatteddata(3,:)=sorteddata(3,:)
formatteddata(4,:)=sorteddata(4,:)
!This file will store coincidence events
Open (60, file='events.csv')
Open (100, file='eventsused.csv')
j=0
Do i = 1,counter-25
time = 1
!Do time = 0,4
Do Energy = 0,150,25
```

Do k = i+1,i+25
check = 0
If
(formatteddata(1,i)==3.and.formatteddata(2,i)==0.and.formatteddata(1,k)==2.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==0.and.formatteddata(1,k)==2.and.formatteddata(2,k)==2)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==0.and.formatteddata(1,k)==2.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==0.and.formatteddata(1,k)==1.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==0.and.formatteddata(1,k)==1.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==1.and.formatteddata(1,k)==2.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==1.and.formatteddata(1,k)==2.and.formatteddata(2,k)==2)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==1.and.formatteddata(1,k)==2.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==1.and.formatteddata(1,k)==1.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==1.and.formatteddata(1,k)==1.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==2.and.formatteddata(1,k)==2.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==2.and.formatteddata(1,k)==2.and.formatteddata(2,k)==2)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==2.and.formatteddata(1,k)==2.and.formatteddata(2,k)==3)
Then
check =1

Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==2.and.formatteddata(1,k)==1.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==2.and.formatteddata(1,k)==1.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==3.and.formatteddata(1,k)==2.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==3.and.formatteddata(1,k)==2.and.formatteddata(2,k)==2)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==3.and.formatteddata(1,k)==2.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==3.and.formatteddata(1,k)==1.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==3.and.formatteddata(2,i)==3.and.formatteddata(1,k)==1.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==0.and.formatteddata(1,k)==2.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==0.and.formatteddata(1,k)==2.and.formatteddata(2,k)==2)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==0.and.formatteddata(1,k)==2.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==0.and.formatteddata(1,k)==1.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==0.and.formatteddata(1,k)==1.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==1.and.formatteddata(1,k)==3.and.formatteddata(2,k)==0)
Then
check =1

Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==2.and.formatteddata(1,k)==3.and.formatteddata(2,k)==0)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==3.and.formatteddata(1,k)==3.and.formatteddata(2,k)==0)
Then
check =1
Else If
(formatteddata(1,i)==1.and.formatteddata(2,i)==1.and.formatteddata(1,k)==3.and.formatteddata(2,k)==0)
Then
check =1
Else If
(formatteddata(1,i)==1.and.formatteddata(2,i)==3.and.formatteddata(1,k)==3.and.formatteddata(2,k)==0)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==1.and.formatteddata(1,k)==3.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==2.and.formatteddata(1,k)==3.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==3.and.formatteddata(1,k)==3.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==1.and.formatteddata(2,i)==1.and.formatteddata(1,k)==3.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==1.and.formatteddata(2,i)==3.and.formatteddata(1,k)==3.and.formatteddata(2,k)==1)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==1.and.formatteddata(1,k)==3.and.formatteddata(2,k)==2)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==2.and.formatteddata(1,k)==3.and.formatteddata(2,k)==2)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==3.and.formatteddata(1,k)==3.and.formatteddata(2,k)==2)
Then
check =1
Else If
(formatteddata(1,i)==1.and.formatteddata(2,i)==1.and.formatteddata(1,k)==3.and.formatteddata(2,k)==2)
Then
check =1

```
Else If
(formatteddata(1,i)==1.and.formatteddata(2,i)==3.and.formatteddata(1,k)==3.and.formatteddata(2,k)==2)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==1.and.formatteddata(1,k)==3.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==2.and.formatteddata(1,k)==3.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==3.and.formatteddata(1,k)==3.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==1.and.formatteddata(2,i)==1.and.formatteddata(1,k)==3.and.formatteddata(2,k)==3)
Then
check =1
Else If
(formatteddata(1,i)==1.and.formatteddata(2,i)==3.and.formatteddata(1,k)==3.and.formatteddata(2,k)==3)
Then
check =1

Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==1.and.formatteddata(1,k)==2.and.formatteddata(2,k)==0)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==2.and.formatteddata(1,k)==2.and.formatteddata(2,k)==0)
Then
check =1
Else If
(formatteddata(1,i)==2.and.formatteddata(2,i)==3.and.formatteddata(1,k)==2.and.formatteddata(2,k)==0)
Then
check =1
Else If
(formatteddata(1,i)==1.and.formatteddata(2,i)==1.and.formatteddata(1,k)==2.and.formatteddata(2,k)==0)
Then
check =1
Else If
(formatteddata(1,i)==1.and.formatteddata(2,i)==3.and.formatteddata(1,k)==2.and.formatteddata(2,k)==0)
Then
check =1
End IF
If (check /=1) Cycle
If (Abs(formatteddata(3,i)-formatteddata(3,k))<time .and. Abs(formatteddata(4,i)-formatteddata(4,k)) <
Energy&
& .and. formatteddata(5,i)/=1 .and. formatteddata(5,k)/=1) Then
Write (60,*)
formatteddata(1,i),",",formatteddata(2,i),",",formatteddata(3,i),",",formatteddata(4,i)
```

```
Write (60,*)
formatteddata(1,k),",",formatteddata(2,k),",",formatteddata(3,k),",",formatteddata(4,k)
formatteddata(5,i)=1
formatteddata(5,k)=1
j = j+1
End If
!End Do
End Do
End Do
End Do
close (60)
Allocate(events(4,j*2))
Print *, '1'
Open(70, file='events.csv')
Read (70,*) events
Close (70)
Print *, '2'
Open(80, file='EventMatrix.csv')
matrix = 0
! Check to see if the coincident event happenend at a pixel and if so update the matrix
count
Do i = 1,j*2,2
If (events(1,i)==3.and.events(2,i)==0.and.events(1,i+1)==2.and.events(2,i+1)==1) Then
matrix(1,1) = matrix(1,1) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==0.and.events(1,i+1)==2.and.events(2,i+1)==2)
Then
matrix(2,1) = matrix(2,1) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==0.and.events(1,i+1)==2.and.events(2,i+1)==3)
Then
matrix(3,1) = matrix(3,1) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==0.and.events(1,i+1)==1.and.events(2,i+1)==1)
Then
matrix(4,1) = matrix(4,1) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==0.and.events(1,i+1)==1.and.events(2,i+1)==3)
Then
matrix(5,1) = matrix(5,1) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
```

```
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==1.and.events(1,i+1)==2.and.events(2,i+1)==1)
Then
matrix(1,2) = matrix(1,2) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==1.and.events(1,i+1)==2.and.events(2,i+1)==2)
Then
matrix(2,2) = matrix(2,2) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==1.and.events(1,i+1)==2.and.events(2,i+1)==3)
Then
matrix(3,2) = matrix(3,2) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==1.and.events(1,i+1)==1.and.events(2,i+1)==1)
Then
matrix(4,2) = matrix(4,2) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==1.and.events(1,i+1)==1.and.events(2,i+1)==3)
Then
matrix(5,2) = matrix(5,2) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==2.and.events(1,i+1)==2.and.events(2,i+1)==1)
Then
matrix(1,3) = matrix(1,3) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==2.and.events(1,i+1)==2.and.events(2,i+1)==2)
Then
matrix(2,3) = matrix(2,3) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==2.and.events(1,i+1)==2.and.events(2,i+1)==3)
Then
matrix(3,3) = matrix(3,3) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==2.and.events(1,i+1)==1.and.events(2,i+1)==1)
Then
matrix(4,3) = matrix(4,3) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
```

write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==2.and.events(1,i+1)==1.and.events(2,i+1)==3)
Then
matrix(5,3) = matrix(5,3) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==3.and.events(1,i+1)==2.and.events(2,i+1)==1)
Then
matrix(1,4) = matrix(1,4) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==3.and.events(1,i+1)==2.and.events(2,i+1)==2)
Then
matrix(2,4) = matrix(2,4) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==3.and.events(1,i+1)==2.and.events(2,i+1)==3)
Then
matrix(3,4) = matrix(3,4) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==3.and.events(1,i+1)==1.and.events(2,i+1)==1)
Then
matrix(4,4) = matrix(4,4) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==3.and.events(2,i)==3.and.events(1,i+1)==1.and.events(2,i+1)==3)
Then
matrix(5,4) = matrix(5,4) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==2.and.events(2,i)==0.and.events(1,i+1)==2.and.events(2,i+1)==1)
Then
matrix(1,5) = matrix(1,5) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==2.and.events(2,i)==0.and.events(1,i+1)==2.and.events(2,i+1)==2)
Then
matrix(2,5) = matrix(2,5) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==2.and.events(2,i)==0.and.events(1,i+1)==2.and.events(2,i+1)==3)
Then
matrix(3,5) = matrix(3,5) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)

write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==2.and.events(2,i)==0.and.events(1,i+1)==1.and.events(2,i+1)==1)
Then
matrix(4,5) = matrix(4,5) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i)==2.and.events(2,i)==0.and.events(1,i+1)==1.and.events(2,i+1)==3)
Then
matrix(5,5) = matrix(5,5) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==0.and.events(1,i)==2.and.events(2,i)==1)
Then
matrix(1,1) = matrix(1,1) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==0.and.events(1,i)==2.and.events(2,i)==2)
Then
matrix(2,1) = matrix(2,1) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==0.and.events(1,i)==2.and.events(2,i)==3)
Then
matrix(3,1) = matrix(3,1) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==0.and.events(1,i)==1.and.events(2,i)==1)
Then
matrix(4,1) = matrix(4,1) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==0.and.events(1,i)==1.and.events(2,i)==3)
Then
matrix(5,1) = matrix(5,1) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==1.and.events(1,i)==2.and.events(2,i)==1)
Then
matrix(1,2) = matrix(1,2) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==1.and.events(1,i)==2.and.events(2,i)==2)
Then
matrix(2,2) = matrix(2,2) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)

```
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==1.and.events(1,i)==2.and.events(2,i)==3)
Then
matrix(3,2) = matrix(3,2) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==1.and.events(1,i)==1.and.events(2,i)==1)
Then
matrix(4,2) = matrix(4,2) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==1.and.events(1,i)==1.and.events(2,i)==3)
Then
matrix(5,2) = matrix(5,2) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==2.and.events(1,i)==2.and.events(2,i)==1)
Then
matrix(1,3) = matrix(1,3) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==2.and.events(1,i)==2.and.events(2,i)==2)
Then
matrix(2,3) = matrix(2,3) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==2.and.events(1,i)==2.and.events(2,i)==3)
Then
matrix(3,3) = matrix(3,3) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==2.and.events(1,i)==1.and.events(2,i)==1)
Then
matrix(4,3) = matrix(4,3) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==2.and.events(1,i)==1.and.events(2,i)==3)
Then
matrix(5,3) = matrix(5,3) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==3.and.events(1,i)==2.and.events(2,i)==1)
Then
matrix(1,4) = matrix(1,4) + 1
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
```

```
write (100,*) events(1,i+1),'','',events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==3.and.events(1,i)==2.and.events(2,i)==2)
Then
matrix(2,4) = matrix(2,4) + 1
write (100,*) events(1,i),'','',events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),'','',events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==3.and.events(1,i)==2.and.events(2,i)==3)
Then
matrix(3,4) = matrix(3,4) + 1
write (100,*) events(1,i),'','',events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),'','',events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==3.and.events(1,i)==1.and.events(2,i)==1)
Then
matrix(4,4) = matrix(4,4) + 1
write (100,*) events(1,i),'','',events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),'','',events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==3.and.events(2,i+1)==3.and.events(1,i)==1.and.events(2,i)==3)
Then
matrix(5,4) = matrix(5,4) + 1
write (100,*) events(1,i),'','',events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),'','',events(2,i+1),',',events(3,i+1),',',events(4,i+1)


Else If (events(1,i+1)==2.and.events(2,i+1)==0.and.events(1,i)==2.and.events(2,i)==1)
Then
matrix(1,5) = matrix(1,5) + 1
write (100,*) events(1,i),'','',events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),'','',events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==2.and.events(2,i+1)==0.and.events(1,i)==2.and.events(2,i)==2)
Then
matrix(2,5) = matrix(2,5) + 1
write (100,*) events(1,i),'','',events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),'','',events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==2.and.events(2,i+1)==0.and.events(1,i)==2.and.events(2,i)==3)
Then
matrix(3,5) = matrix(3,5) + 1
write (100,*) events(1,i),'','',events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),'','',events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==2.and.events(2,i+1)==0.and.events(1,i)==1.and.events(2,i)==1)
Then
matrix(4,5) = matrix(4,5) + 1
write (100,*) events(1,i),'','',events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),'','',events(2,i+1),',',events(3,i+1),',',events(4,i+1)
Else If (events(1,i+1)==2.and.events(2,i+1)==0.and.events(1,i)==1.and.events(2,i)==3)
Then
matrix(5,5) = matrix(5,5) + 1
```

```
write (100,*) events(1,i),",",events(2,i),',',events(3,i),',',events(4,i)
write (100,*) events(1,i+1),",",events(2,i+1),',',events(3,i+1),',',events(4,i+1)
End If
End Do
close (100)
! write the matrix to a file and to the screen
Do i = 5,1,-1
Write (80,*) matrix(1,i),",",matrix(2,i),",",matrix(3,i),",",matrix(4,i),",",matrix(5,i)
Write (*,*) matrix(1,i),",",matrix(2,i),",",matrix(3,i),",",matrix(4,i),",",matrix(5,i)
End Do
Close (80)
Print *, ' '
Print *, 'There were ',Sum(Matrix),' coincident events.'
Print *, ' '
End Subroutine Coin
End Module Coincidence
```

**Module display**

```
use AVDef
use DFLib
USE AVVIEWER
implicit none
contains
subroutine visualizer(matrix)
Real*8 :: minvalue,maxvalue
Integer, dimension(1) :: Max
Integer*4 :: hv,image,onoroff,paletteid,status,yflip
Integer, Dimension(5,5),Intent(InOut) :: matrix
! transose used here just to make the array visualizer
! display the data in the same orientation as the detector
matrix=transpose(matrix)
! Use the array viewer to display the matrix of values
call faglStartWatch(Matrix,status)
call favStartViewer (hv, status)
! favSetArray tells the viewer to observe our array.
call favSetArray(hv, matrix, status)
! Array Viewer comes up hidden, use favShowWindow(AV_TRUE) to make it visible
! on the screen.
call favShowWindow(hv, AV_TRUE, status)
! sets the type of graph to an IMAGE
call favSetGraphType (hv, Imagemap, status)
! turns off the palette auto adjust
onoroff=0
call favSetPaletteAutoAdjust (hv, onoroff, status)
```

! sets the value of the palette to range from 0 to the max value in the array matrix
minvalue = 0.
max = maxval(matrix)
maxvalue = sum(max)
call favSetPaletteRange (hv, minvalue, maxvalue, status)
! sets the color scaling on the image
!grayscale inverted
paletteid=3
!rainbow inverted
!paletteid=6
call favSetPaletteId (hv, paletteId, status)
! this is used to orient the image correctly.  It flips the image along the yaxis
yflip=2
call favSetImageOrientation (hv, yflip, status)
! tells the array visualizer to stop watching how the array changes
call faglEndWatch(matrix, status)
End subroutine visualizer
End module display

## Appendix B.  Fetter model analysis

The Fetter model provides a hypothetical design of an implosion type nuclear weapon.  In collaboration with Captain Chris Williams, an attenuation analysis was conducted using the mass and thicknesses of all material layers presented in the Fetter model.  The analysis concentrated on 640 and 413 keV gammas from the Pu-239 in order to compare the results of this analysis to spectral data results of the Soviet SS-N-12 SLCM presented by Fetter et al [9].

Prior to conducting the analysis on the Fetter model, an assumption about the grade of the plutonium was necessary.  The plutonium was assumed to be weapons grade with a Pu-239 concentration of 93%.

The mass attenuation coefficients and densities for each material needed for the attenuation analysis were obtain by Capt Williams using XMuDat and are as follows.

| Material | Density g/cm$^3$ | 413 keV Mass Attenuation Coefficient in cm$^2$/g | 640 keV Mass Attenuation Coefficient in cm$^2$/g |
|---|---|---|---|
| Beryllium | 1.848 | 0.08351 | 0.06945 |
| Uranium | 18.95 | 0.2771 | 0.1355 |
| High Explosive | 0.221 | 0.1197 | 0.09947 |
| Aluminum | 2.699 | 0.09137 | 0.07566 |

An estimate of the Pu-239 activity was calculated using the mass and dimensions of the plutonium in the Fetter model and energy and branching ratios for the Pu-239 gammas of interest.  The following is the activity calculation.

$Energy_{413} := 413.713 \quad Energy_{640} := 639.99 \qquad Ratio_{413} := \dfrac{.00151}{100} \quad Ratio_{640} := \dfrac{.0000079}{100}$

Activity of Pu based on mass of Pu core in Fetter Model

$PuDensity := 19.84 \qquad \dfrac{g}{cm^3} \qquad\qquad N_a := 6.0221367 \cdot 10^{23}$

$OutsideSphereRadius := 5.0 \quad cm \qquad\qquad InsideSphereRadius := 4.25 \quad cm$

$$SphereVolume := \left(\dfrac{4}{3} \cdot \pi \cdot OutsideSphereRadius^3\right) - \left(\dfrac{4}{3} \cdot \pi \cdot InsideSphereRadius^3\right) \ cm^3$$

Assuming 93% $^{239}Pu$ in the Pu core,

$Mass_{239} := PuDensity \cdot SphereVolume \cdot 0.93$

$Mass_{239} = 3.728 \times 10^3 \ g$

$Moles_{239} := \dfrac{Mass_{239}}{239.05}$

$Moles_{239} = 15.595$

The total number of atoms in the Pu core of the Fetter model is

$Atoms_{239} := Moles_{239} \cdot N_a \quad$ where $N_a$ is Avagadro's Number

$Atoms_{239} = 9.391 \times 10^{24}$

The half-life of Pu in seconds

$PuHalfLife_{239} := 24110 \cdot 365 \cdot 24 \cdot 60 \cdot 60 \ sec$

Therefore, the decay constant, $\lambda$, is

$LambdaPu_{239} := \dfrac{\ln(2)}{PuHalfLife_{239}}$

$LambdaPu_{239} = 9.116 \times 10^{-13} \ \dfrac{1}{sec}$

The activity of the Pu is simply the number of atoms decayed in 1 second. Using equation (10), with a time of 1 second

$Activity_{239} := Atoms_{239} - Atoms_{239} \cdot e^{\left(-LambdaPu_{239} \cdot 1\right)}$

$Activity_{239} = 8.561 \times 10^{12} \qquad Bq$

Therefore, the activity for each gamma energy is:

$\text{Activity}_{413} := \text{Activity}_{239} \cdot \text{Ratio}_{413}$ $\qquad\qquad$ $\text{Activity}_{640} := \text{Activity}_{239} \cdot \text{Ratio}_{640}$

$\text{Activity}_{413} = 1.293 \times 10^{8} \quad \text{Bq}$ $\qquad\qquad$ $\text{Activity}_{640} = 6.763 \times 10^{5} \quad \text{Bq}$

The relative attenuation for each layer of 413 and 640-keV gammas are as follows. Self-attenuation by the Pu was neglected.

Attenuation through 2 cm Be

$\text{Bedensity} := 1.848 \cdot \dfrac{g}{cm^3}$

$\text{BeMassAttenCoeff}_{413} := 0.08351 \cdot \dfrac{cm^2}{g}$ $\qquad$ $\text{BeMassAttenCoeff}_{640} := 0.06945 \cdot \dfrac{cm^2}{g}$

$\text{BeLinearAttenCoeff}_{413} := \text{BeMassAttenCoeff}_{413} \cdot \text{Bedensity}$

$\text{BeLinearAttenCoeff}_{640} := \text{BeMassAttenCoeff}_{640} \cdot \text{Bedensity}$

$\text{BeLinearAttenCoeff}_{413} = 15.433 \ m^{-1}$ $\qquad$ $\text{BeLinearAttenCoeff}_{640} = 12.834 \ m^{-1}$

$\text{Beattenuation}_{413} := e^{-\text{BeLinearAttenCoeff}_{413} \cdot 2 \cdot cm}$ $\qquad$ $\text{Beattenuation}_{413} = 0.734$

$\text{Beattenuation}_{640} := e^{-\text{BeLinearAttenCoeff}_{640} \cdot 2 \cdot cm}$ $\qquad$ $\text{Beattenuation}_{640} = 0.774$

Attenuation through 3 cm $^{238}U$

$\text{Udensity} := 18.95 \cdot \dfrac{g}{cm^3}$

$\text{UMassAttenCoeff}_{413} := 0.2771 \cdot \dfrac{cm^2}{g}$ $\qquad$ $\text{UMassAttenCoeff}_{640} := 0.1355 \cdot \dfrac{cm^2}{g}$

$\text{ULinearAttenCoeff}_{413} := \text{UMassAttenCoeff}_{413} \cdot \text{Udensity}$

$\text{ULinearAttenCoeff}_{640} := \text{UMassAttenCoeff}_{640} \cdot \text{Udensity}$

$\text{ULinearAttenCoeff}_{413} = 525.105 \ m^{-1}$ $\qquad$ $\text{ULinearAttenCoeff}_{640} = 256.772 \ m^{-1}$

$\text{Uattenuation}_{413} := e^{-\text{ULinearAttenCoeff}_{413} \cdot 3 \cdot cm}$ $\qquad$ $\text{Uattenuation}_{413} = 1.44 \times 10^{-7}$

$\text{Uattenuation}_{640} := e^{-\text{ULinearAttenCoeff}_{640} \cdot 3 \cdot cm}$ $\qquad$ $\text{Uattenuation}_{640} = 4.514 \times 10^{-4}$

Attenuation through 10 cm PETN .

$HEdensity := 0.221 \cdot \dfrac{g}{cm^3}$

$HEMassAttenCoeff_{413} := 0.1197 \cdot \dfrac{cm^2}{g}$ $\qquad$ $HEMassAttenCoeff_{640} := 0.09947 \cdot \dfrac{cm^2}{g}$

$HELinearAttenCoeff_{413} := HEMassAttenCoeff_{413} \cdot HEdensity$

$HELinearAttenCoeff_{640} := HEMassAttenCoeff_{640} \cdot HEdensity$

$HELinearAttenCoeff_{413} = 2.645 \ m^{-1}$

$HELinearAttenCoeff_{640} = 2.198 \ m^{-1}$

$HEattenuation_{413} := e^{- HELinearAttenCoeff_{413} \cdot 10 \cdot cm}$ $\qquad$ $HEattenuation_{413} = 0.768$

$HEattenuation_{640} := e^{- HELinearAttenCoeff_{640} \cdot 10 \cdot cm}$ $\qquad$ $HEattenuation_{640} = 0.803$

Attenuation through 1 cm Al

$Aldensity := 2.699 \cdot \dfrac{g}{cm^3}$

$AlMassAttenCoeff_{413} := 0.09137 \cdot \dfrac{cm^2}{g}$ $\qquad$ $AlMassAttenCoeff_{640} := 0.07566 \cdot \dfrac{cm^2}{g}$

$AlLinearAttenCoeff_{413} := AlMassAttenCoeff_{413} \cdot Aldensity$

$AlLinearAttenCoeff_{640} := AlMassAttenCoeff_{640} \cdot Aldensity$

$AlLinearAttenCoeff_{413} = 24.661 \ m^{-1}$ $\qquad$ $AlLinearAttenCoeff_{640} = 20.421 \ m^{-1}$

$Alattenuation_{413} := e^{- AlLinearAttenCoeff_{413} \cdot 1 \cdot cm}$ $\qquad$ $Alattenuation_{413} = 0.781$

$Alattenuation_{640} := e^{- AlLinearAttenCoeff_{640} \cdot 1 \cdot cm}$ $\qquad$ $Alattenuation_{640} = 0.815$

Therefore, the maximum total photons transmitted by the Pu-239 through the entire

hypothetical model are as follows.

$$\text{Photonstransmitted}_{413} := \text{Activity}_{413} \cdot e^{-\text{BeLinearAttenCoeff}_{413} \cdot 2 \cdot cm} \cdot e^{-\text{ULinearAttenCoeff}_{413} \cdot 3 \cdot cm}$$

$$e^{-\text{HELinearAttenCoeff}_{413} \cdot 10 \, cm} \cdot e^{-\text{AlLinearAttenCoeff}_{413} \cdot 1 \cdot cm}$$

$$\text{Photonstransmitted}_{640} := \text{Activity}_{640} \cdot e^{-\text{BeLinearAttenCoeff}_{640} \cdot 2 \cdot cm} \cdot e^{-\text{ULinearAttenCoeff}_{640} \cdot 3 \cdot cm}$$

$$e^{-\text{HELinearAttenCoeff}_{640} \cdot 10 \, cm} \cdot e^{-\text{AlLinearAttenCoeff}_{640} \cdot 1 \cdot cm}$$

$$\text{Photonstransmitted}_{413} = 8.203 \quad \frac{1}{s} \qquad\qquad \text{Photonstransmitted}_{640} = 154.55 \quad \frac{1}{s}$$

Assuming a solid angle of 0.00052 steradians, a detector efficiency of 30% at 413-keV and a detector efficiency of 20% [9], the detected count rate for each gamma of interest was determined and is shown below.  The solid angle and efficiency from the Soviet weapon was used to replicate the geometry dependence allowing for an accurate comparison.

$$\text{Omega} := .00052$$

$$\text{Photonsrecorded}_{413} := \text{Photonstransmitted}_{413} \cdot \text{Omega} \cdot 0.30$$

$$\text{Photonsrecorded}_{640} := \text{Photonstransmitted}_{640} \cdot \text{Omega} \cdot 0.20$$

$$\text{Photonsrecorded}_{413} = 1.28 \times 10^{-3} \quad \frac{\text{counts}}{\text{sec}} \qquad\qquad \text{Photonsrecorded}_{640} = 0.016 \quad \frac{\text{counts}}{\text{sec}}$$

The resulting detected count rate at 640-keV is an order of magnitude larger than 413-keV.  This means either detection time can be reduced by an order of magnitude or the 640-keV spectrum will have an order of magnitude larger number of information carriers.

The Soviet nuclear SS-N-12 SLCM analyzed by Fetter, had observed count rates of 1.582±0.064 and 0.083±0.025 counts per second for 413-keV and 640-keV, respectively, using a 151 cm$^3$ HPGe detector [9].  This doesn't compare well with the above analysis of the Fetter model.  In fact, the Fetter model analysis suggests the 640-

keV count rate is larger than the 413-keV count rate and the actual weapon analysis suggests the opposite. One possible explanation for this phenomenon is the thickness of the uranium. The relative attenuation of the uranium in the Fetter model was approximately 3 orders of magnitude greater for 413-keV photons versus the 640-keV. Therefore, a reduction in the thickness of uranium would decrease this large attenuation difference. Changing the uranium thickness to 1.67 cm and re-accomplishing the Fetter model analysis, resulted in observed count rates of 1.381 and 0.489 counts per second for 413-keV and 640-keV gammas, respectively. These results are much closer to the weapon analysis with a simple change in uranium thickness.

In conclusion, the Fetter model, did not compare well with data from the analysis of an actual Soviet weapon. The model did compare well, however, when the uranium tamper thickness was reduced from 3 cm to 1.67 cm in the Fetter model. The analyzed Russian weapon's design is unknown. Therefore, it is difficult to discount the Fetter model, since a country with a large nuclear arsenal would most probably have numerous designs. As a result, the Fetter model, will be assumed accurate for the purposes of this study.

# Appendix C.  Resolution measurement spectra

**Single strip spectrum**
**46800 sec**
**Strip F1**
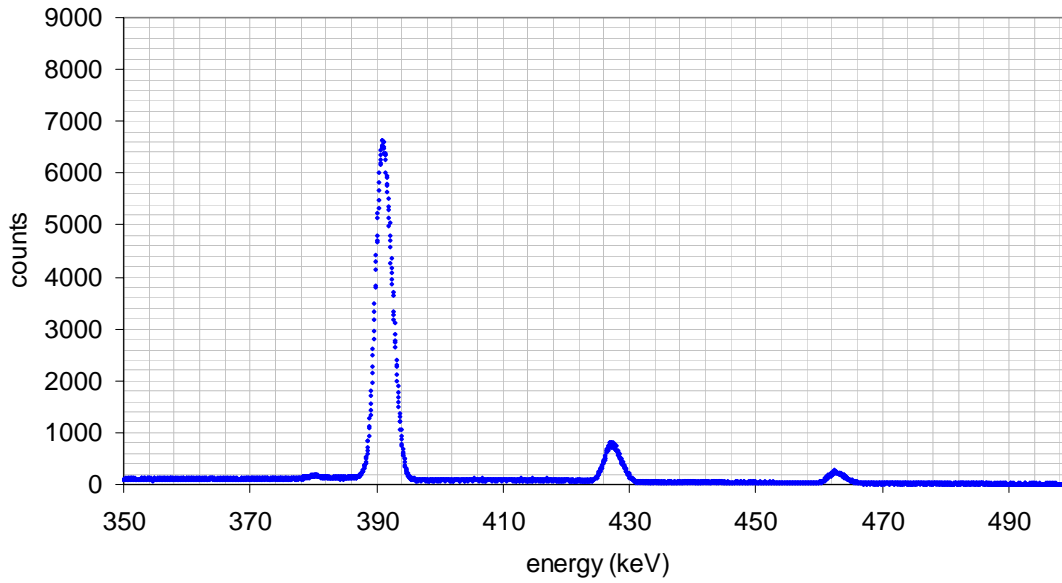


**Single strip spectrum**
**46800 sec**
**Strip F2**
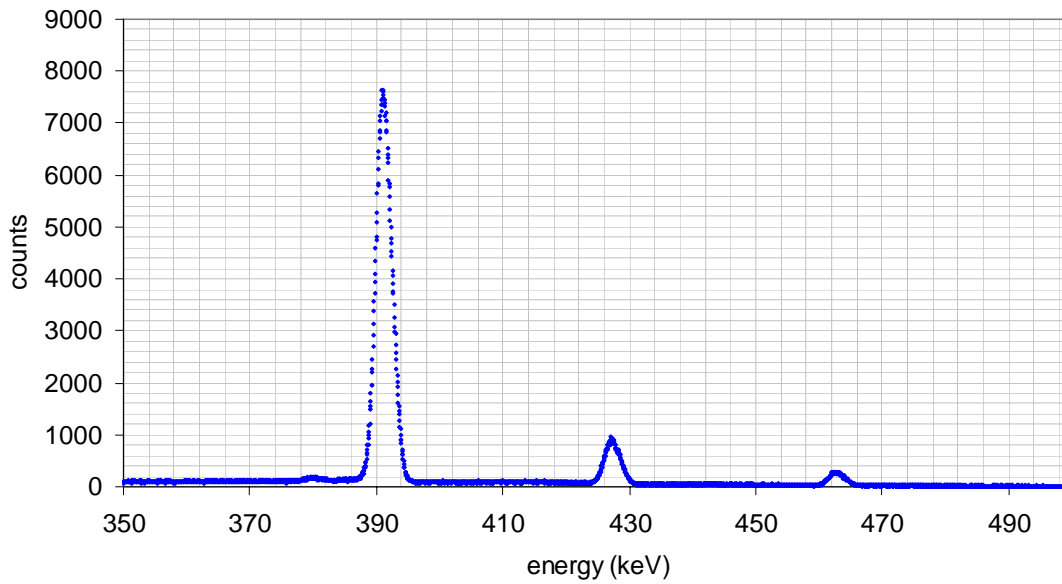
**Single strip spectrum**
**46800 sec**
**Strip F3**
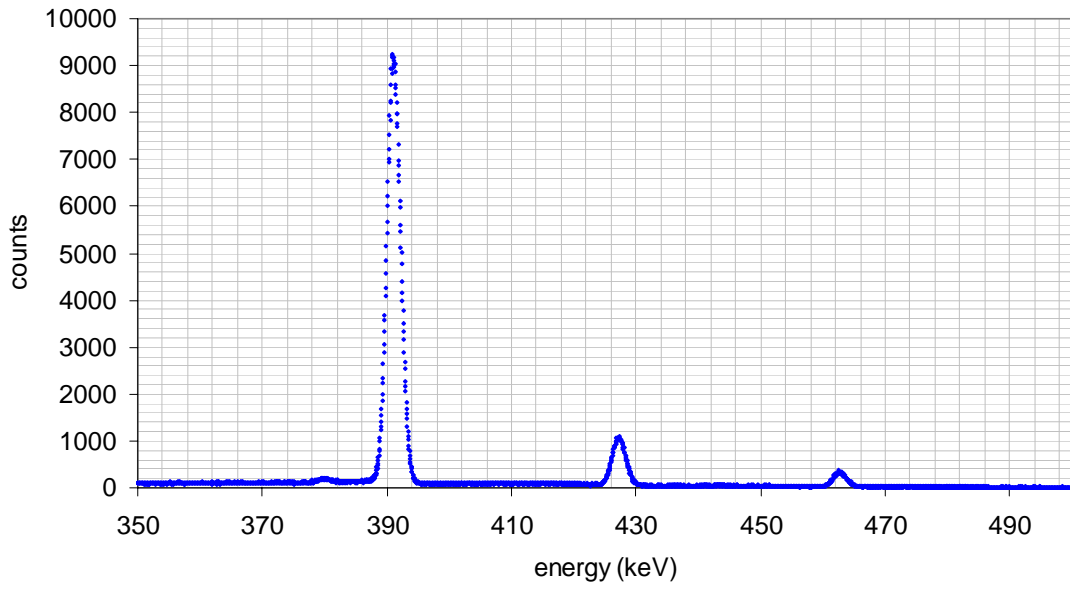


**Single strip spectrum**
**46800 sec**
**Strip F4**

103
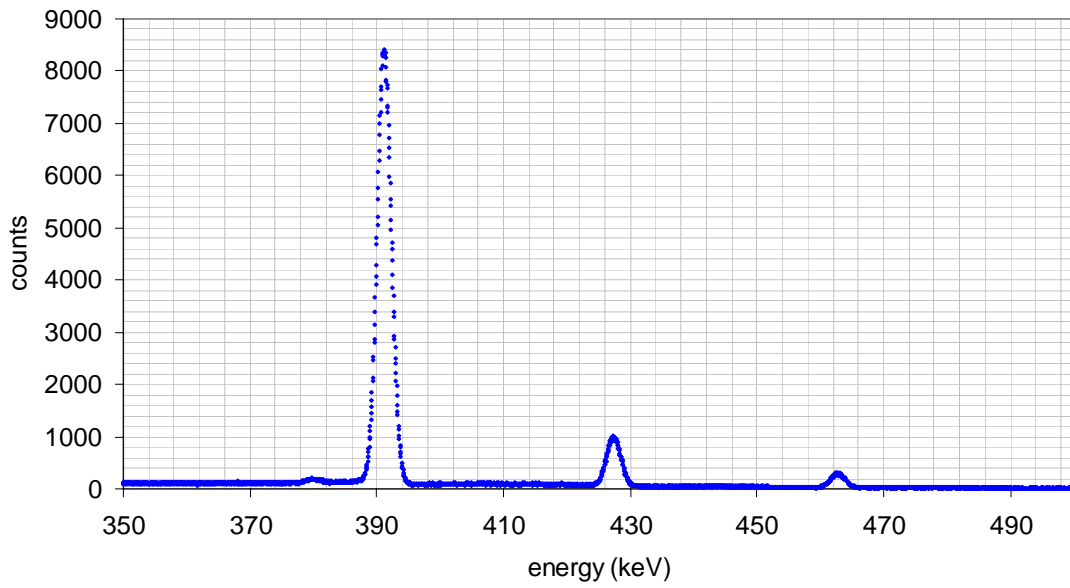
**Single strip spectrum**
**46800 sec**
**Strip F5**
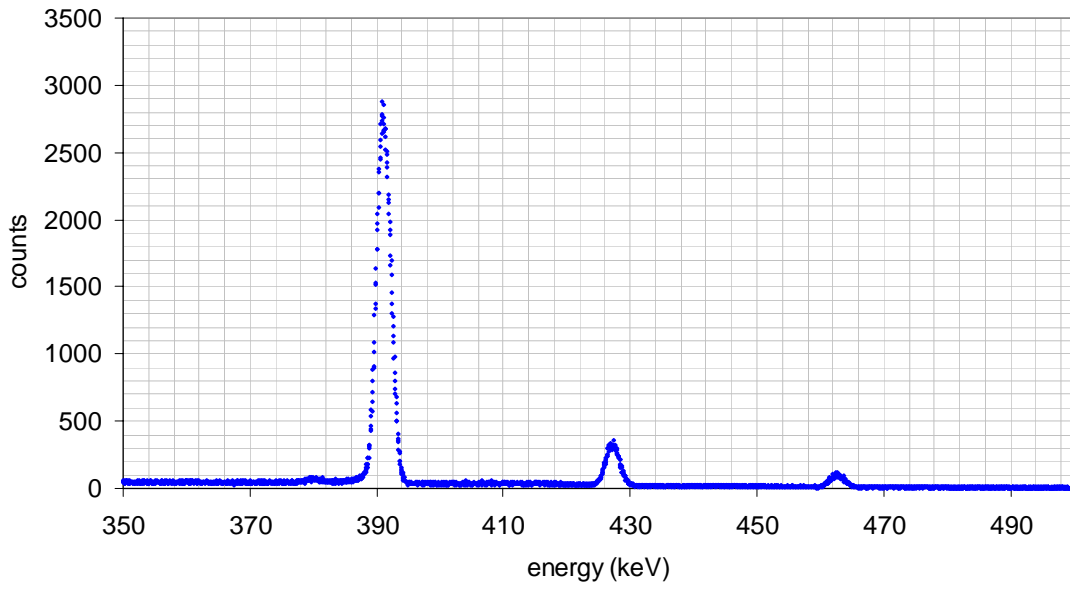


**Single strip spectrum**
**46800 sec**
**Strip R1**
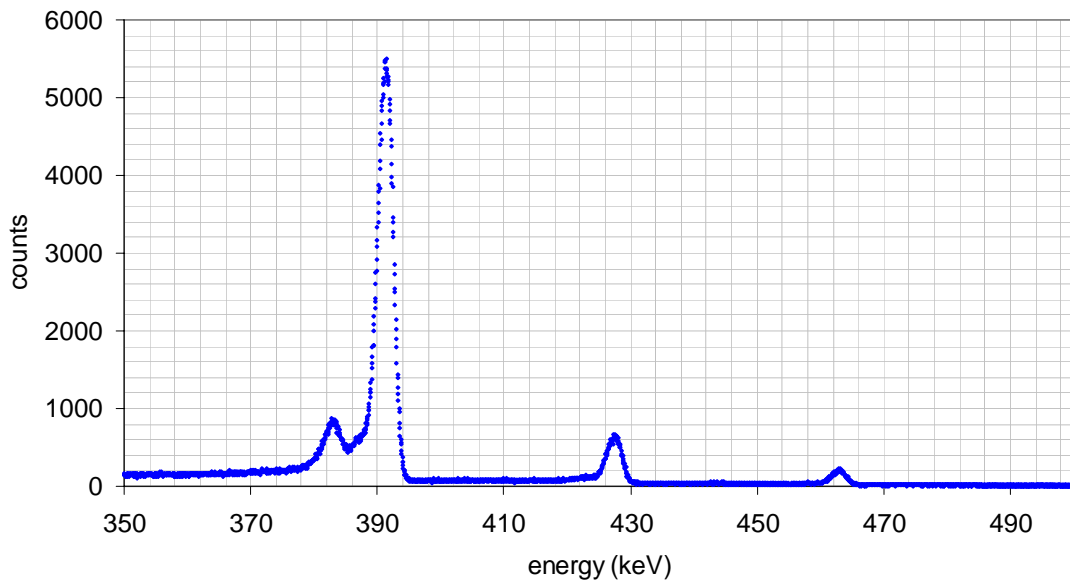
**Single strip spectrum
46800 sec
Strip R2**
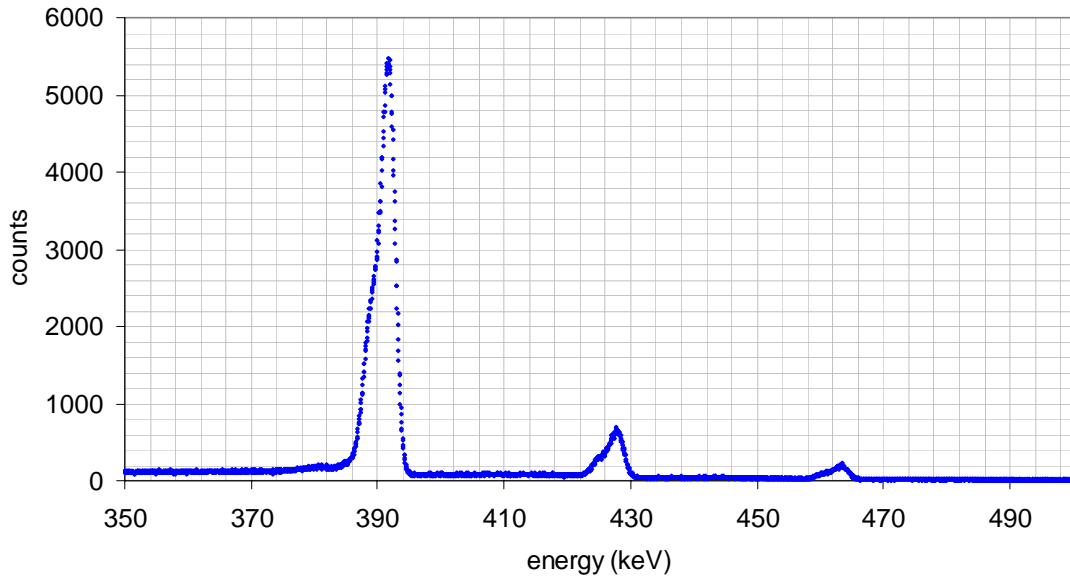
**Single strip spectrum
46800 sec
Strip R3**

105
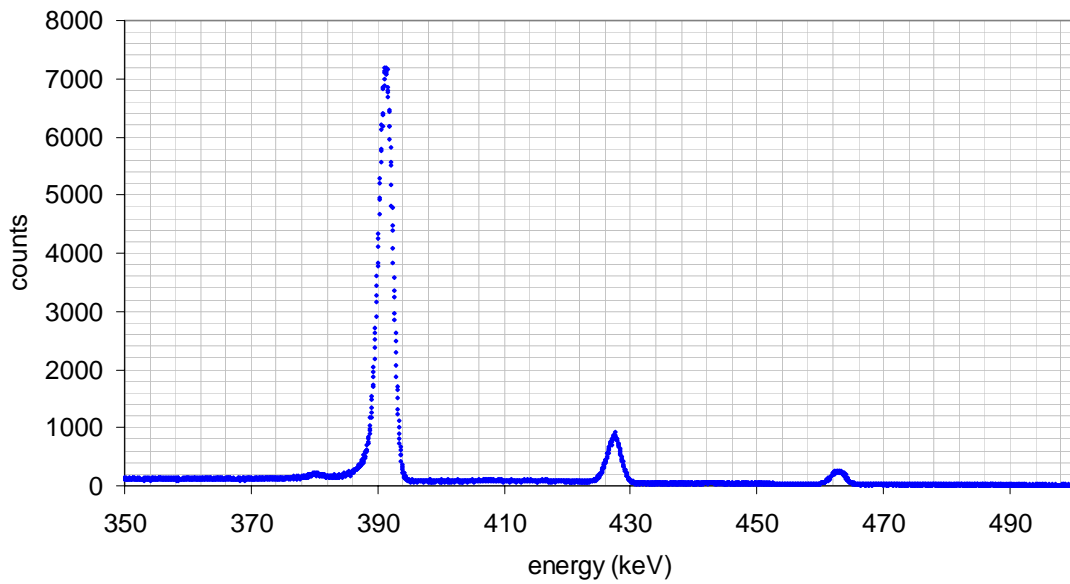
**Single strip spectrum**
**46800 sec**
**Strip R4**
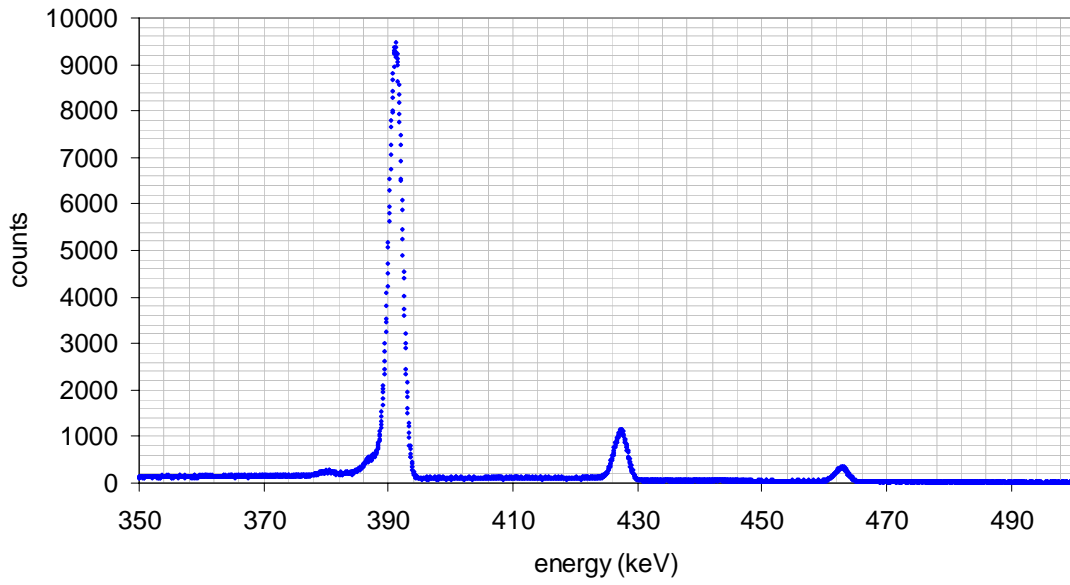


**Single strip spectrum**
**46800 sec**
**Strip R5**
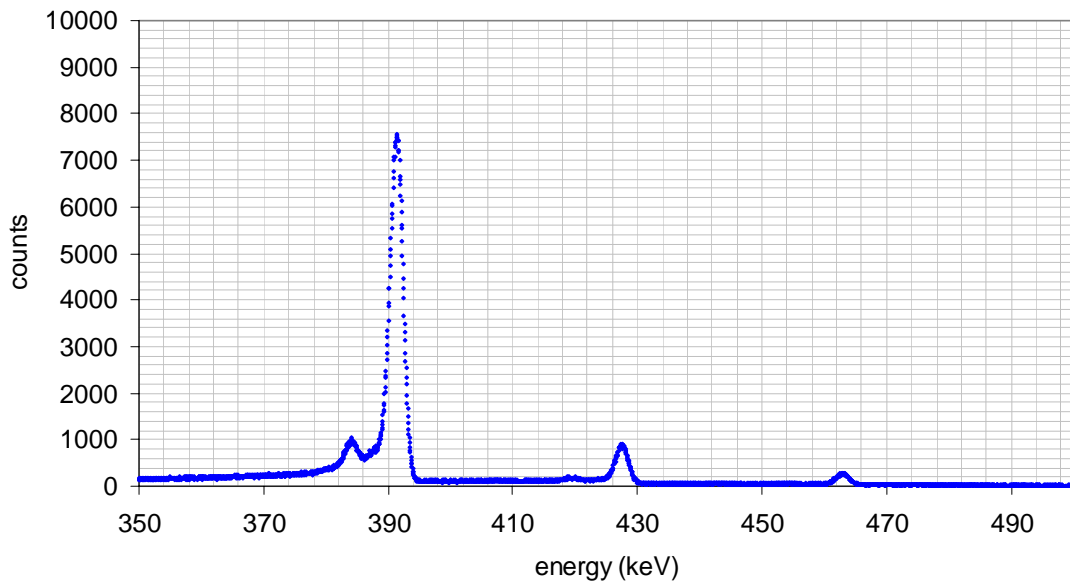
## Bibliography

[1] Amersham Health, "Anger Camera Description http://www.amershamhealth.com/medcyclopaedia/Volume%20I/Anger%20camera.asp,"

[2] H.O. Anger, "Scintillation Camera," *Rev. Sci. Instrum,* vol. 29, pp. 27-33, 1958.

[3] H.O. Anger, "Scintillation Camera with Multichannel Collimators," *J. Nucl. Med.,* vol. 5, pp. 515-531, 1964.

[4] V.G. Baryshevsky, A.A. Khrutchinsky, V.I. Moroz and M.D. Dezhurko, "Gamma-Ray Spectroscopic System for Remote Detection and Monitoring of Fissile Materials," *IEEE Transactions on Nuclear Science,* vol. 41, August 1994.

[5] L.W. Burggraf, "NENG 650 laboratory write-ups," 2003.

[6] T.E. Cartledge, "Comparison of Spectral Analysis Software Programs (RobWin and RSEMCA)," *Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH,* March 2001.

[7] Z.H. Cho, J.P. Jones, M. Singh, *Foundations of Medical Imaging,* New York: John Wiley & Sons Inc., 1993.

[8] B.L. Evans, "A Portable Compton Gamma-Ray Camera Design," *Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH,* November 1995.

[9] S. Fetter, T.B. Cochran, L. Grodzins, H.L. Lynch and M.S. Zucker, "Gamma-Ray Measurements of a Soviet Cruise-Missile Warhead," *Science,* vol. 248, pp. 828-834, 18 May 1990.

[10] S. Fetter, V.A. Frolov, M. Miller, R. Mozley, O.F. Prilutsky, S.N. Rodionov and R.Z. Sagdeev, "Detecting Nuclear Weapons," *Science and Global Security,* vol. 1, pp. 225-302, 1990.

[11] M. Gombia, A.B. Brill, D. Bollini and A. Del Guerra, "A Simulation and Modeling Study Comparing the Performance of a Germanium Orthogonal Strip Detector and an Anger Camera,"

[12] Idaho National Engineering & Environmental Laboratory, "Ge(Li)-Si(Li) Gamma Spectrum Catalogue http://wastenot.inel.gov/gamma/data1.html,"

[13] S.E. Inderhees, B.F. Phlips, R.A. Kroeger, W.N. Johnson, R.L. Kinzer, Kurfess J D., B. Graham and N. Gehrels, "Spectroscopy, Imaging and Compton-scatter Polarimetry with a Germanium Strip Detector," *IEEE Transactions on Nuclear Science,* vol. 43, pp. 1467-1471, June. 1996.

[14] Jorway Corporation, *Instuction Maunual, Model 73A, SCSI Bus CAMAC Crate Controller,*

[15] A. Kantzas, K. Hamilton, T. Zarabi, A. Bhargava, I. Wright, G. Brook and J. Chen, "Application of Gamma Camera Imaging and SPECT Systems in Chemical Processes," *1st World Congress on Industrial Process Tomography,* pp. 264-271, April 14-17. 1999.

[16] G.F. Knoll, *Radiation Detection and Measurement,* New York: John Wiley & Sons, Inc., 2000.

[17] J.D. Kurfess, W.N. Johnson, R.A. Kroeger, B.F. Phlips and E.A. Wulf, "Development and applications of position-sensitive solid-state gamma ray detectors," *Nuclear Instruments and Methods in Physics Research A,* vol. 505, pp. 256-264, 2003.

[18] Naval Research Laboratory, "Gamma and Cosmic Ray Astrophysics Branch http://hese.nrl.navy.mil/gamma/detector/detector.htm,"

[19] M.B. Nelson, "Detection of Special Nuclear Material with High Purity Germanium (HPGe) and Mercuric Iodide (HgI2) Gamma Detectors," *Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH,* March 2003.

[20] ORTEC, *High-purity Germanium Double-Sided Strip Detector -HPGeDSSD-Operating Manual,*

[21] J.G. Rogers and P. Gumplinger, "A Pixelated 3D Anger Camera with Light-Loss Compensation," IEEE NM1 Joint NSS/MIC Session, 1999.

[22] J.E. Turner, *Atoms, Radiation, and Radiation Protection*, New York: John Wiley & Sons Inc., 1995.

[23] W-ie-Ne-R, NIM / CAMAC User's and Service-Manual,

[24] XIA, DGF-4C Users Manual

**Vita**


Captain Clifford V. Sulham was born at Barksdale Air Force Base in Shreveport, Louisiana. He graduated from Henry Clay High School in Lexington, Kentucky, and entered undergraduate studies at the University of Kentucky in Lexington, Kentucky, and graduated with a Bachelor of Science degree in Physics. He received his Air Force commission, and his first assignment was to the 32nd Air Refueling Squadron at McGuire Air Force Base, New Jersey where he was the squadron assistant mobility officer. He transferred to his second assignment, at the Air Force Research Laboratory, Space Vehicles directorate at Kirtland Air Force Base, New Mexico where he worked as a research physicist. In July 2002, he entered the School of Engineering Physics, Air Force Institute of Technology.

| **1. REPORT DATE** *(DD-MM-YYYY)* March 2004 | **2. REPORT TYPE** Master's Thesis | **3. DATES COVERED** *(From – To)* August 2002 – March 2004 |
|---|---|---|

| **4. TITLE AND SUBTITLE** SPECIAL NUCLEAR MATERIAL IMAGING USING A HIGH PURITY GERMANIUM DOUBLE SIDED STRIP DETECTOR | **5a. CONTRACT NUMBER** |
|---|---|
| | **5b. GRANT NUMBER** |
| | **5c. PROGRAM ELEMENT NUMBER** |
| **6. AUTHOR(S)** Sulham, Clifford V., Captain, USAF | **5d. PROJECT NUMBER** |
| | **5e. TASK NUMBER** |
| | **5f. WORK UNIT NUMBER** |

| **7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)** Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865 | **8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GNE/ENP/04-07 |
|---|---|

| **9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)** DTRA/TDND Attn: LTC Thomas Cartledge 8725 John J. Kingman Rd    DSN: 221-9670 FT Belvoir, VA 22060-6201   email: thomas.cartledge@dtra.mil | **10. SPONSOR/MONITOR'S ACRONYM(S)** |
|---|---|
| | **11. SPONSOR/MONITOR'S REPORT NUMBER(S)** |

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

   The purpose of this research is to investigate the feasibility of using a High Purity Germanium Double Sided Strip Detector (HPGeDSSD) to image Special Nuclear Material (SNM) for use in the field of nuclear nonproliferation. SNM shape can be determined by using a gamma ray imaging system that is capable of imaging either Pu-239 or U-235. The detector used for this research is a liquid-nitrogen cooled germanium detector with five charge collection strips on both faces. The strips on the two faces are orthogonal to each other, creating a five by five array of data collection pixels. When the detector is coupled to a parallel hole collimator, an imaging system is achieved. The system is capable of imaging individual or multiple photon energies. The capability of the imaging system to produce images of simulated Pu-239 was demonstrated. The analyses of the images allowed distributed sources to be distinguished from point sources, which was the primary goal for developing this technology.

**15. SUBJECT TERMS**
   Special Nuclear Material, Spoof Identification, High Purity Germanium, HPGe, Strip Detector

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** Larry W. Burggraf, PhD, (ENP) |
|---|---|---|---|---|---|
| **a. REPORT** U | **b. ABSTRACT** U | **c. THIS PAGE** U | UU | 121 | **19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-3636, ext 4507 (larry.burggraf@afit.edu) |