Muon Tomography for Detection of Shielded Nuclear Contraband Using GEM Detectors

by

Michael Justin Staib

Bachelor of Science in Physics
Florida Institute of Technology
2007

A thesis
submitted to the College of Science at
Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Physics

Melbourne, Florida
May, 2012

We the undersigned committee hereby recommend
that the attached document be accepted as fulfilling in
part the requirements for the degree of
Master of Science in Physics

"Muon Tomography for Detection of Shielded Nuclear Contraband Using GEM Detectors," a thesis by Michael Justin Staib

Marcus Hohlmann, Ph.D. Associate Professor Physics and Space Sciences	Hakeem Oluseyi, Ph.D. Assistant Professor Physics and Space Sciences
Debasis Mitra, Ph.D. Professor Computer Sciences	Terry D. Oswalt, Ph.D. Professor and Head Physics and Space Sciences

Abstract

Title: Muon Tomography for Detection of Shielded Nuclear Contraband Using GEM Detectors

Author: Michael Justin Staib

Advisor: Dr. Marcus Hohlmann

Muon tomography for homeland security applications aims at detecting well shielded nuclear contraband in cargo. By measuring the multiple Coulomb scattering of cosmic ray muons traversing some unknown material, we are able to produce tomographic images using a simple reconstruction based on the total scattering angle and the point of closest approach (POCA) of the incoming and outgoing particle trajectories. A prototype Muon Tomography Station (MTS) using Gas Electron Multiplier (GEM) detectors has been constructed and characterized. Data acquisition for the MTS is accomplished using the largest implementation of the RD51 Scalable Readout System to date. Software development for online monitoring and offline analysis is presented. Efforts to improve the sensor alignment using tracking information are presented, including an iterative method based on histograms of the residual distributions as well as efforts towards alignment based on a Kalman filter. Several POCA reconstructions of different materials are shown. It is shown that it is possible to discriminate between materials of different radiation lengths of similar volume using this method. A lower limit of 6 mm has been set for the spatial imaging resolution in the horizontal plane with ~100 k reconstructed tracks. It is also shown that it is possible to image shielded high-Z material inside a moderate amount of shielding. Recommendations for future work are also presented.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analog/Digital Converter
AMORE	Automatic MOnitoRing Environment
DATE	ALICE Data Acquisition and Test Environment
DOCA	Distance of Closest Approach
FEC	Front End Concentrator
FPGA	Field-Programmable Gate Array
GEM	Gas Electron Multiplier
LANL	Los Alamos National Lab
MPGD	Micro-Pattern Gaseous Detector
MTS	Muon Tomography Station
POCA	Point of Closest Approach
SNM	Special Nuclear Material
SRS	Scalable Readout System

Acknowledgements

There are a great number of people without which this thesis would not have been possible. I would first like to dedicate this thesis my parents and family, for their constant support of my endeavors, even if they still don't quite understand what a muon is. To my advisor, Dr. Hohlmann, I cannot thank you enough for the opportunity to perform such interesting research. Your useful advice and personal experience have helped me to develop greatly as a scientist. You certainly know how to get the most out of each and every one of your students.

I would very much like to thank all of the students in HEP lab A, both past and present. It was certainly a great team to work with, and I am sure that the group will continue to produce great things in the future. There are a few colleagues I would particularly like to thank. I would like to thank Amilkar Quitero for getting me up to speed and introducing me to the GEM detector. I would also like to thank Bryant Benson for being just as clueless as I was when this all started. It was very nice to have someone learn with and to test out new ideas. I would like to thank Vallary and Jessie for their help with data collection and processing of the data for this thesis. I would also like to thank Lenny Grasso for his work designing and building the support structure for the MTS. I would most like to thank Kondo Gnanvo for introducing me to data acquisition and analysis using advanced electronics. I certainly owe you many, many coffees by now.

None of this would have been possible without the help from the RD51 electronics group at CERN. I would like to thank Sorin Martiou and Hans Muller for helping me troubleshoot the SRS hardware, and for their continued efforts to improve the SRS for the entire user community. I owe the greatest debt to Filippo Costa, without whom I could never have fully implemented the DAQ system at Florida Tech.

I would lastly like to thank the Department of Homeland Security Domestic Nuclear Detection Office for their support of this work and the opportunity it has provided myself and many others.

Chapter 1
Introduction

1.1 Muon Tomography for Homeland Security

The difficulty of detecting well-shielded nuclear contraband is one of the major challenges facing comprehensive examination of cargo entering the country. Such materials could be used to create weapons targeting American cities. In the year 2011, more than 23 million cargo containers and 92 million privately-owned vehicles were processed at the by U.S. Customs and Border Protection [1]. As part of the 9/11 act signed into law on August 3, 2007, a mandate was set requiring 100% scanning of all US-bound containers using non-intrusive imaging equipment and radiation detection equipment by July 1, 2012 [2]. This initiative has helped support a large amount of research into effective cargo scanning methods in recent years. These methods seek to detect fissile materials such as U^{235} and Pu^{239}, designated as special nuclear material (SNM) by the Atomic Energy Act of 1954 [3], as well as radiation signatures of other radioactive materials that could be used in a radiological dispersion device also known as a "dirty bomb". Most of these methods rely on detecting the characteristic radiation emitted from nuclear decays within these materials by passive or active methods [3]. Unfortunately, these radiological signatures are easy to shield with innocuous materials such as iron or lead.

In situations such as these, SNM may be hidden inside well-shielded containers. In order to detect such a threat, it is necessary to probe the material with some highly penetrating method. One such technique, muon tomography, exploits the highly penetrating nature of cosmic-ray muons produced in the upper atmosphere to probe large volumes of dense material. A muon is a subatomic particle similar to an electron with a rest mass of 105.7 MeV/c^2 and a mean lifetime of 2.2 μs [4] that is

produced primarily by weak decays of pions formed by the nuclear interaction of cosmic rays with the upper atmosphere. At sea level the average cosmic-ray muon flux through a horizontal planar detector is roughly 1 cm^{-2} min^{-1} [5]. The use of muons to inspect large volumes of material can be traced back to 1955, when E.P. George used the method to determine the overburden of a mine deep inside a mountain by measuring the distribution of muon absorption [6]. This method was also used by Luis Alvarez to search for hidden chambers within the pyramids of Giza [7]. The concept of muon tomography for homeland security was first proposed and developed at Los Alamos National Laboratory (LANL). Instead of measuring the absorption of muons within some material, the LANL method exploits the multiple Coulomb scattering of muons in order to create a tomographic reconstruction of material through which the muons pass. The first results of imaging using this method were published in 2003 and can be found in references [8] and [9]. These results show the ability to discriminate high-Z from low-Z materials where Z is the charge number of the nucleus (atomic number).

Multiple Coulomb scattering is a non-deterministic process resulting from scattering of the electrically charged muon by the charged nuclei of atoms along its path. This produces a distribution of scattering angles where the central 98% can be well approximated by a Gaussian distribution with width [10]

$$\Theta_0 = \frac{13.6\ \text{MeV}}{\beta c p} \cdot z \cdot \sqrt{\frac{x}{X_0}} \cdot \left[1 + 0.038 \ln\left(\frac{x}{X_0}\right)\right],$$

where p is the momentum of the muon, βc is its velocity, z is the charge number of the incident particle (in this case 1), and x/X_0 is the amount of material traversed by the muon in radiation lengths. The functional dependence allowing the ability to discriminate between different materials comes from this final term. The radiation length X_0 of a material is well approximated by the formula [10]

$$X_0 = \frac{716.4 \text{ g cm}^{-2} \cdot \text{A}}{\text{Z}(\text{Z}+1)\ln(287/\sqrt{\text{Z}})}$$

where Z is the atomic number of the material being traversed and A is the atomic mass of the material. This approximately linear dependence on Z allows the discrimination of materials by measuring the muon momentum and the amount of total scattering taking place in the imaging region.

In order to measure the trajectory of the muons as they enter and exit the imaging region, particle detectors developed for high-energy physics experiments are used. These detectors exhibit high efficiency and excellent spatial resolution. At LANL, drift tubes are used for particle tracking and provide a spatial resolution of about 400 μm [11]. These detectors are a cost effective method to cover the large area needed to increase solid angle coverage when scanning large cargo containers. For the experimental results presented in this thesis, a compact cubic-foot Muon Tomography Station (MTS) has been assembled using a more recently developed detector technology, the Gas Electron Multiplier (GEM) detector. These detectors provide higher spatial resolution than drift tubes, on the order of 70 μm for perpendicular tracks [12], coupled with a more compact form factor. The detector itself is presented in further detail in the next chapter.

1.2 Scope of thesis

It is the intent of this thesis to demonstrate that a compact muon tomography station can be built using GEM detectors. In order to achieve this goal, certain design requirements must be met. The detector must exhibit high spatial resolution, on the order of a few hundred microns, in order to precisely measure the scattering of the muons. GEM detectors meet this goal, but in order to take full advantage of their excellent spatial resolution, it must be possible to align the detectors to a high precision. Electronics must be developed for acquiring data from ~12 k analog channels. Software tools must be developed in order to process and analyze these data.

These design requirements have all been met, and constitute the major work performed.

The topics covered in this thesis cover hardware design and commissioning, as well as software development for online and offline analysis. The Gas Electron Multiplier (GEM) is first introduced. Topics covered include the fabrication of triple-GEM detectors using infrared heating as well as results from commissioning detectors used for muon tracking. The data acquisition system is discussed next, including details of the hardware comprising the Scalable Readout System (SRS) as well as presenting the analysis packages that have been developed for online monitoring and offline analysis. The design of the Muon Tomography Station (MTS) used for the experimental reconstructions will then be introduced including a discussion of alignment algorithms used to improve the quality of the tomographic reconstructions. Experimental results of muon tomography with GEM detectors will then be presented. Results on material discrimination and the spatial imaging resolution of the prototype are discussed. Finally, conclusions and recommendations for future work are included.

Chapter 2

The Gas Electron Multiplier Detector

2.1 The Gas Electron Multiplier

The Gas Electron Multiplier (GEM) was introduced by Fabio Sauli in 1997 [13]. The GEM belongs to a larger class of detectors called Micro-Pattern Gaseous Detectors (MPGD) [14]. These detectors take advantage of advances in printed circuit technology to produce microscopic electron amplifying structures. In the case of the GEM, the base material is 50 μm Kapton foil that has been clad on both sides with 5 μm of copper. A pattern of holes is then etched into this base material using photolithography and chemical etching. The procedure for producing these foils has been developed by the CERN Surface Treatment Service. The holes are biconical in shape with outer diameters of 70 μm and inner diameters of 60 μm. They are equally spaced over the surface of the GEM foil with a pitch of 140 μm. An image of a GEM foil under an electron microscope can be found in Figure 2.1.

By placing a bias voltage across the two sides of the GEM foil, strong electric fields develop inside the holes. This field is high enough to allow amplification of a single electron by way of an electron avalanche in the gas. An image of the field lines in the area of the holes can be found in Figure 2.2. The primary electrons are from ionization of the gas in the drift region of the detector. An electric field is maintained throughout the detector volume to cause these primary electrons to drift into the holes of the GEM foil to be amplified. At a bias voltage of 400V in an Argon-CO_2 70:30 gas mix, each individual foil produces an effective gain of order 10^2 [14]. In the case of the GEM detectors used for this study, three GEM layers are cascaded to produce an effective gain on the order of 10^4 with very low discharge probability [15]. An illustration of this concept can be seen in Figure 2.3.

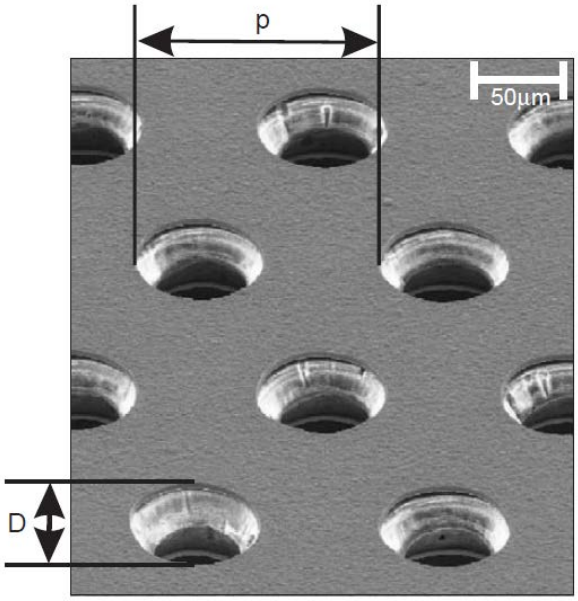


Figure 2.1 GEM foil under electron microscope. The outer diameter of the hole D = 70 μm, and the pitch between holes p = 140 μm [16].

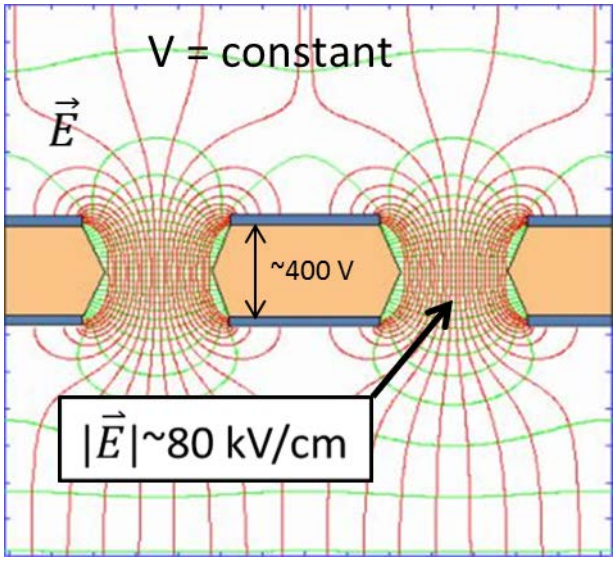


Figure 2.2 Cross-sectional view of a GEM foil showing the electric field and equipotential lines in the region inside the holes [16].

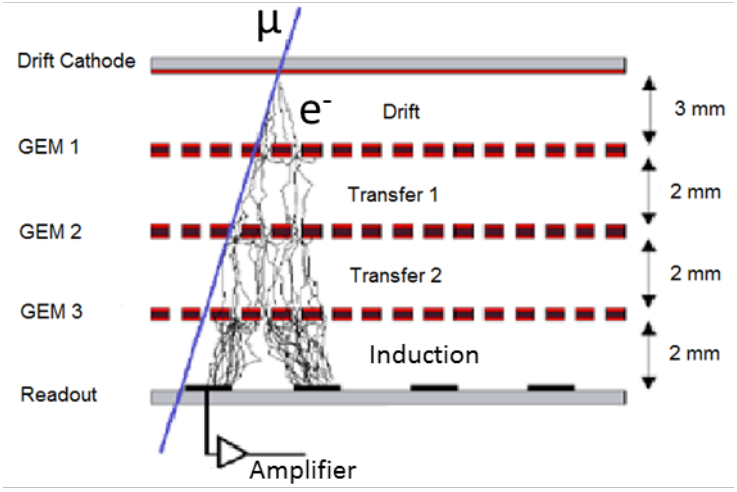


Figure 2.3 Illustration showing the cascaded layers of the triple-GEM detector and the amplification concept [15].

2.2 Design and Construction of GEM Detectors for Muon Tomography

The design used as a basis for the 30 cm x 30 cm triple-GEM detectors for muon tomography was developed for the COMPASS experiment at CERN [17]. These detectors were developed for particle tracking along the beam axis and were chosen for their high rate capability and spatial resolution. In a high intensity muon beam, spatial resolutions of ~70 μm and an average time resolution of 12 ns were measured [12]. An exploded view of the COMPASS design can be found in Figure 2.4 showing the general structure of the detector. Design parameters have been tuned to minimize the gas discharge probability and to maximize detector efficiency.

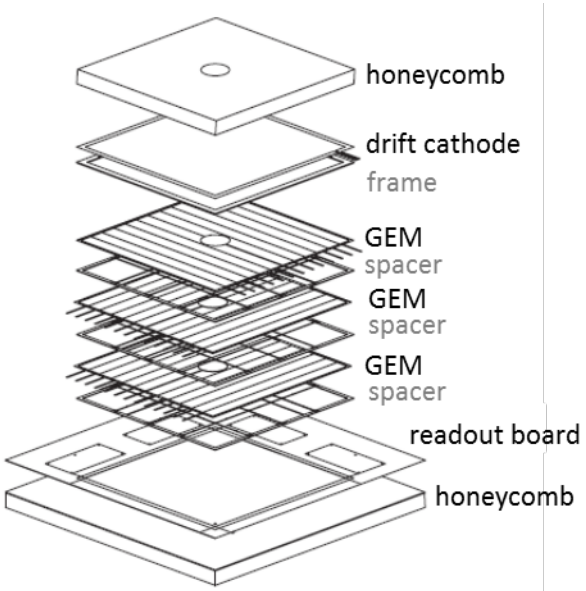


Figure 2.4 Exploded view of COMPASS triple-GEM detector [12].

Each GEM layer is separated by a fiberglass spacer frame with thin ribs to minimize any non-uniformity in the GEM spacing. This helps to maintain a consistent electric field between the different layers of the detector. For this design, a drift gap of 3 mm is used. The two remaining transfer gaps and the induction gap are kept at a spacing of 2 mm (see also Figure 2.3). A voltage divider is used to distribute a total bias voltage of 4.2 kV to the drift plane and each of the GEM layers. The individual GEM foil biases and the electric fields in the interstitial gas regions can be found in Table 2.1. The readout is provided by a Cartesian XY readout plane using copper strips at a pitch of 400 μm. This readout plane is built on a Kapton substrate, with an additional Kapton layer providing insulation between the two orthogonal sets of strips. The width of the strips has been optimized to provide close to equal charge sharing between these two layers. An image of the readout structure can be found in Figure 2.5. There are 768 strips in each projection connected to six 130-pin Panasonic AXK6SA3677YG connectors. The data acquisition system used to measure the induced charge on each of these strips will be discussed in the next chapter. An image of a fully assembled GEM detector for muon tomography can be found in Figure 2.6.

Table 2.1 Bias and field configurations for triple-GEM detector at a total bias voltage of 4.2 kV.

$\Delta V_{GEM\ 1}$	426 V
$\Delta V_{GEM\ 2}$	388 V
$\Delta V_{GEM\ 3}$	340 V
|E| Drift	2.54 kV/cm
|E| Transfer 1	3.81 kV/cm
|E| Transfer 2	3.81 kV/cm
|E| Induction	3.81 kV/cm

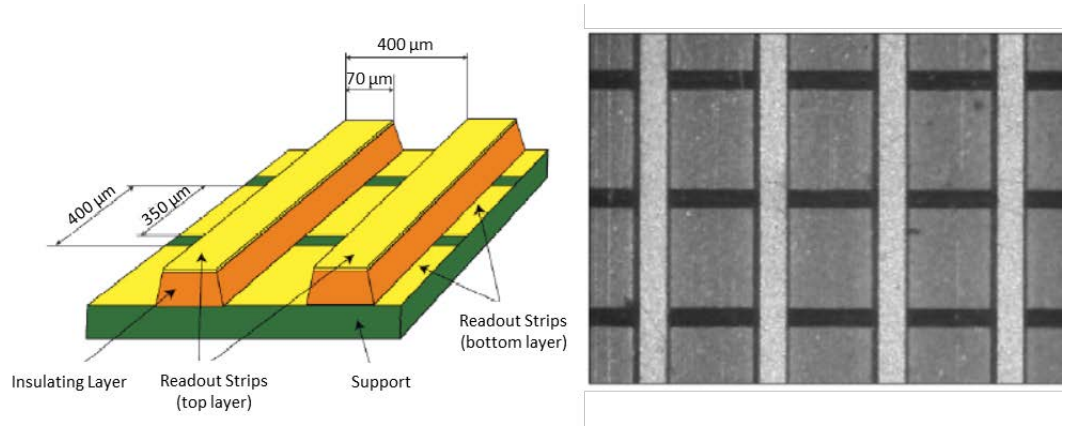


Figure 2.5 Depiction of the Cartesian readout structure (left) and a photograph taken under microscope (right) [16].

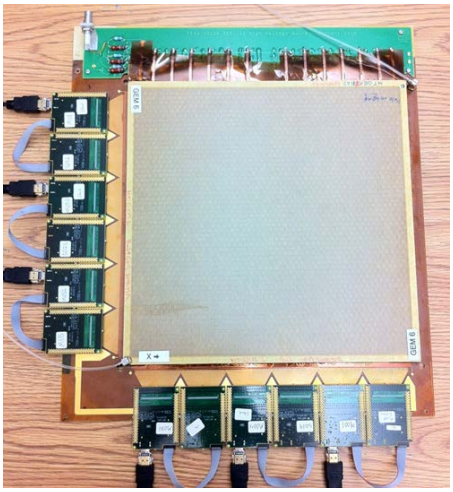


Figure 2.6 Fully assembled and instrumented 30 cm x 30 cm triple-GEM detector used in the Florida Tech muon tomography station.

Nine triple-GEM and one double-GEM detector were assembled for use in the muon tomography station [18]. As the GEM foils are produced on a flexible Kapton substrate, it is necessary to stretch the foils and glue them to the fiberglass spacer frames. Two of the triple-GEM detectors used in the MTS were assembled using a low-cost stretching technique using infrared heat lamps (Figure 2.7). This method has been described in detail in an earlier thesis [19], as well as an RD51 internal note [20]. The method derives from the existing method of stretching GEM foils mounted in a Plexiglas frame by heating in a clean-room oven. Large clean-room ovens can be quite costly. Since the required temperatures are quite low, infrared heat lamps may be used in place of the clean-room oven. The two detectors assembled using this method are functioning just as well as detectors with foil stretched by the traditional method [20].

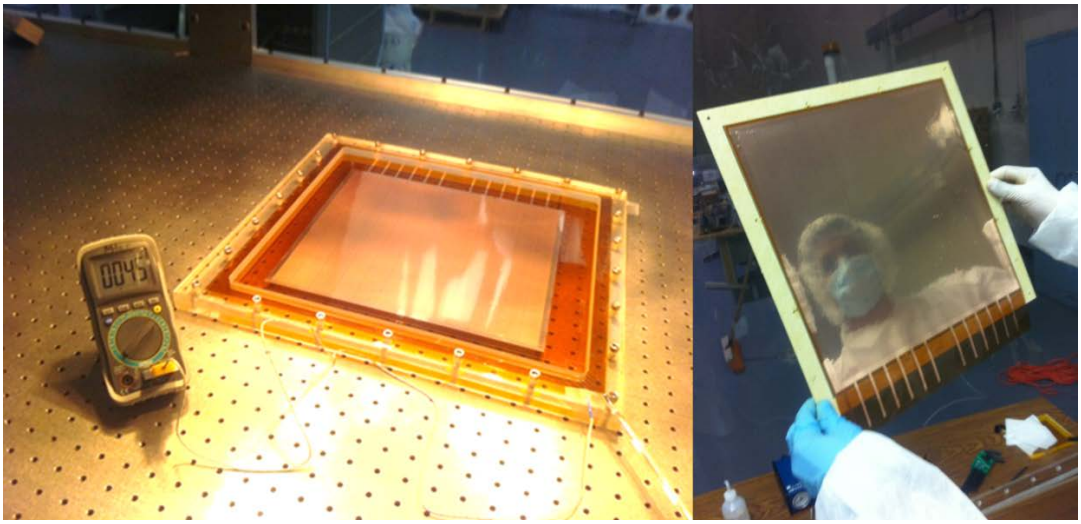


Figure 2.7 GEM foil stretched at 45° C under with infrared lamp heating (left) and the final result after being glued to the spacer frame (right).

Chapter 3

Data Acquisition with the Scalable Readout System

3.1 SRS Hardware

The SRS [21] has been developed as part of an effort by RD51 [22] at CERN to provide a flexible, scalable readout system to the user community specifically aimed at MPGDs. The initial system consists of four basic parts. A first front-end hybrid was developed utilizing the APV25 chip for analog signal acquisition from the GEM detector. This hybrid is then connected via HDMI cable to an ADC card. Two octal ADC chips handle the work of digitizing the analog information from 16 APV hybrids. This data is then sent to a Front End Concentrator (FEC) that collects all of the data from the front end cards and formats the data for transfer to the DAQ computer. The final formatting of data on the DAQ computer is handled by DATE, which was originally developed for use by the ALICE [23] collaboration at CERN, and has been expanded to support data from the SRS. A monitoring package was developed using ALICE's AMORE framework [24] which allows both online monitoring and offline analysis of raw data files saved by DATE. A conceptual depiction of a medium-size SRS can be found in Figure 3.1. The following sections will present a brief overview of the components of the DAQ system.

3.1.1 APV Hybrid

The APV25 [25] was originally designed for use in the CMS tracker at the LHC. The chip features 128 analog channels each with a 192-cell pipeline. It is fabricated using a 0.25 μm CMOS process and exhibits high radiation tolerance, and low noise and power requirements [25]. The APV25 is wirebonded to the APV hybrid, and a protective topglob layer is added to protect the chip during handling. The connection from the hybrid to the GEM detector is made using a 130-pin Panasonic

connector. 128 of these pins carry the analog signals from the detector to the APV, while the other 2 pins connect to ground. There is diode protection at the input to protect the APV chip from any discharges between the detector layers and the readout. It is possible to connect two hybrids together in a master/slave configuration to decrease the number of connections to the ADC. The master APV hybrid has an HDMI type C (mini) connector for data transfer [21]. Four different versions of the APV hybrid have currently been produced. Version 3 is used for the data presented in this thesis (Figure 3.2).

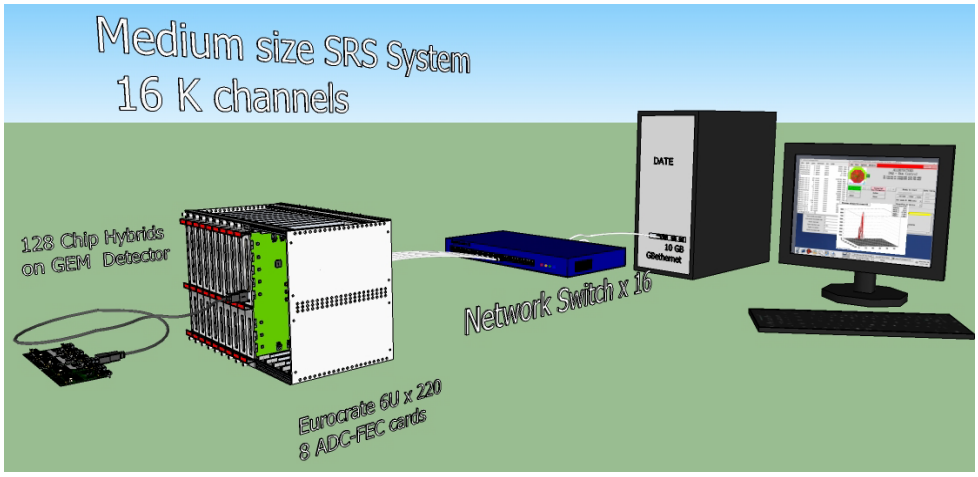


Figure 3.1 Conceptual depiction of a medium-size SRS system (courtesy H. Muller).

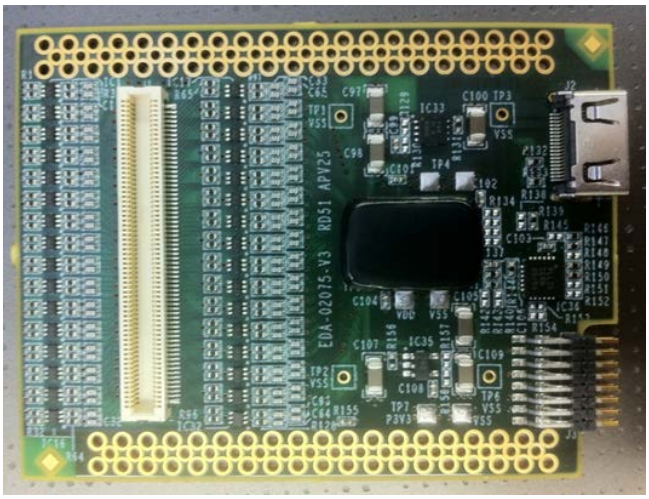


Figure 3.2 Version 3 APV hybrid with master configuration.

3.1.2 Analog-Digital Converter (ADC) card

After receiving a trigger from the FEC, the data is sent from the APV hybrid to the ADC card (Figure 3.3) via an HDMI A-C cable. The ADC card [26] features 8 HDMI type-A inputs which support a total of 16 APV hybrids in master/slave configuration. The digitization is performed by two 12-bit octal-channel ADCs. The ADCs each have programmable gain of 0 dB to 12 dB, low-frequency noise suppression, and 40 MHz sampling. Clock and trigger distribution from the FEC to the hybrids takes place on the ADC card. There is also on-board power and temperature monitoring [27].

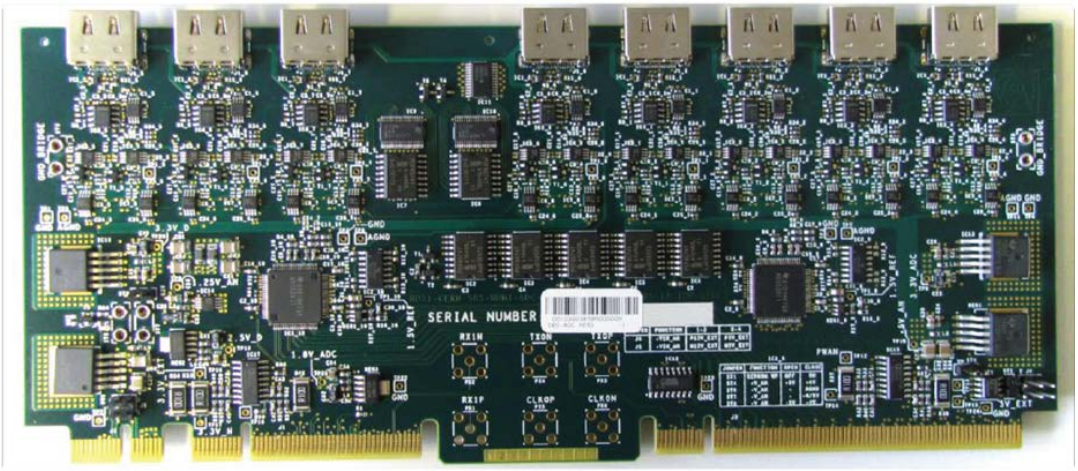


Figure 3.3 SRS Analog-Digital Converter card [26].

3.1.3 Front-End Concentrator (FEC) card

The final piece of specialized hardware in the DAQ chain is the Front-End Concentrator (FEC) card (Figure 3.4). The design was produced at the University of Valencia by Jose Toledo [26]. The FEC is designed to be common to all SRS applications while front-end electronics and digitizer cards may vary. The key component of the FEC is a Virtex LX50T FPGA. A firmware code with common functional blocks is implemented for the entire user community and controls all of the

I/O interfaces on the FEC. There is a 256-MB DDR2 buffer allowing 16-bit read/write at 400 MHz. On the back panel of the FEC, there are several I/O interfaces. NIM and LVDS general purpose inputs are available. For this application, the trigger signal is provided by a Quarknet discriminator/coincidence card [28] which is fanned out by a NIM module to the NIM input on the 6 different FECs. An SFP socket allows either copper or optical transmission of data to the DAQ computer. The FEC sends data using this link via the UDP protocol using jumbo frames of up to 9 kB. There are also two additional RJ45 ports that serve as a data, timing and control (DTC) link [26]. These are not currently used in this experiment, although their value will be discussed in the next section. Slow-control of all of the hardware components is handled via a C script run on the DAQ computer. A detailed description of the slow-control registers may be found in reference [29].

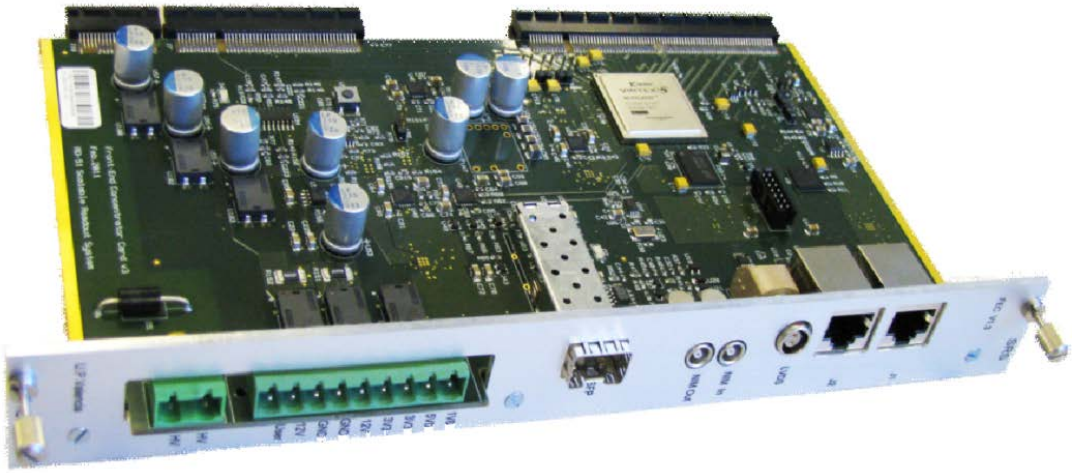


Figure 3.4 The SRS Front-End Concentrator card [26].

3.1.4 Troubleshooting SRS hardware

A few issues with the SRS hardware had to be addressed. One problem was discovered with the mapping of the HDMI pins from the APV Hybrid to the ADC. As a result of this problem, standard HDMI A-C cables do not properly work with the system. A workaround for this problem was developed. By inserting a small adapter piece (Multicomp 60S019P-301N-B) into a standard HDMI A-A cable, the mapping problem is addressed. The newest version of the APV Hybrid (Version 4) now resolves this mapping issue and uses smaller HDMI micro connectors.

There is also an outstanding issue with the trigger system of the SRS hardware. Without use of the DTC link, the clocks of each of the individual FECs are not synchronized. Occasionally, the trigger logic fails to detect a NIM trigger pulse in one of the cards. Since there is limited consistency between the clocks of the FECs, this may go unnoticed by the DAQ system, causing corruption of the data after the missed trigger. Several checks were implemented in order to mitigate this problem. An event counter and timestamp can be enabled in the slow control registers. The trigger count is then used by AMORE to cross check that it receives the same value from each FEC. An additional check of the number of clock cycles between events has been implemented in DATE. The time difference between the current trigger and the previous trigger is compared across all FECs. A threshold can be set in the DATE source code for the maximum value obtained in this comparison. DATE must then be recompiled for this change to take effect. If a particular comparison is above the threshold, the run is aborted. The threshold value has to be set quite high since the longer the time between triggers, the larger the amount the clocks on the different FECs will drift from one another. Currently this threshold is set at 150 clock cycles, or 3.75 μs. Ideally these clocks would be synchronized such that any difference between these values would indicate an error. There are plans to include synchronization such as this using the DTC links and a clock signal provided by either the Ethernet switch or a Clock/Trigger fan-out card currently being developed.

The last problematic issue with the SRS hardware still remains unaddressed. When using more than one FEC, an Ethernet switch may be used to send the data to the DAQ computer. It has been found that each Ethernet switch seems to have a limit to the number of FECs it can support simultaneously. If more than this number of FECs is used, there is a "packet mismatch" error in DATE where it reports receiving the data out of order. The reason for this limitation is unknown. There have been several hypotheses, but none of these have proven correct. It was first thought that perhaps the simultaneous arrival of the data from all of the FECs was overloading the switch. To test this theory there was a delay implemented in the firmware so that the data were not sent simultaneously. This did not fix the problem. Since the number of supported FECs varies from one model of network switch to another, it was then hypothesized that the problem could be that the buffer size of the switch was too low, or that different settings for a configurable network switch might enable the use of more FECs. A new configurable network switch with more buffer memory was purchased (Cisco SG300-10). Even without any change in the configuration, this switch supports simultaneous data from two FECs which is less than the first switch used (Netgear JGS516) which supports simultaneous data from 4 FECs. After attempting many changes to the configuration, there were no improvements. Interestingly, a switch can support more FECs in the case that some of the HDMI channels are disabled on the ADC. This may indicate that the data throughput is still too high for the switch to handle. However, simply decreasing the size of the data sent by a FEC with all channels enabled to a size that is equivalent to the disabling of HDMI channels does not have the same effect. All of these efforts still seem to indicate that it is some hardware limitation of the switch itself, but the exact limitation is not understood. Currently, there are two Netgear gigabit Ethernet switches in use for the 6 FECs used for the MTS. In total, these two network switches can support up to 8 FECs. Clearly, this could lead to problems when trying to scale up to a larger system.

3.2 DAQ software for the SRS

The original DAQ software for the implementation of the SRS consists of two software packages developed for the ALICE experiment at CERN [23]. DATE (Data Acquisition and Test Environment) [24] has been adapted to accept UDP packets from the SRS and is used for the collection of raw data from all of the FECs and event-building of data into a binary format that can be used for online monitoring or may be recorded to disk for offline analysis. The configuration of DATE is stored in a MySQL database. A GUI used for configuring DATE that can be accessed using the editDb command. Using this GUI, it is straightforward for the user to add and configure different equipment. It is also possible to configure start-of-run commands using this interface. Slow-control of the SRS hardware has been implemented using a bash script (see Appendix B) that is issued as a start-of-run command and is used to configure the hardware registers of the SRS.

While the size and complexity of the DATE package may be a bit excessive for a small DAQ system, there are the obvious advantages to using this preexisting package for the data collection. The scalability of the architecture is proven. The package is used to control data collection for the entire ALICE experiment, comprising several different detector types with millions of total readout channels spanning several DAQ subsystems [23]. As there is a large user and development community, there exist also packages for online monitoring and offline analysis of data for DATE. One such package, AMORE (Automatic MOnitoRing Environment) [24], has been selected for use with the SRS.

AMORE is used as a data quality monitoring framework for the ALICE experiment. It is based on the DATE monitoring library and ROOT [30] and is used to monitor online data and produce various monitoring objects such as histograms of important detector characteristics. It may also be used as an offline analysis framework using the DATE monitoring library to parse out raw data. For application with the SRS, AMORE serves both of these roles. The design philosophy of AMORE is based

on a publisher-subscriber paradigm. This design philosophy can be understood by the representation in Figure 3.5. To use AMORE for a new application, a developer needs to create a "module" that defines the behavior of the publishing and the subscribing methods for a particular detector type. An agent is then created that runs the publishing portion of the code. Multiple instantiations of the publishing portion of the module can be run on different agents, allowing multiple data streams to be processed simultaneously. Client methods can be written that subscribe to the monitor object that are published by the agents. These can then be displayed through a custom GUI. Published data, which can be any C++/ROOT object inheriting from "TObject", are sent to a MySQL database to be retrieved by the clients. There is no direct communication between the agents and the clients, but they may send notifications via DIM (Distributed Information Management).

Two different modules were written at Florida Tech for use with the SRS. The amoreMTS package was the first package developed for use during the commissioning of the SRS. It has the ability to publish a variety of detector characterization plots as well as provide some basic tracking for the MTS. There is also a customizable GUI for display of these monitor objects. The package will be discussed in further detail in the next section. More recent developments in the analysis tools are included as part of the second "amoreSRS" module. The groundwork for these modules including event decoding and creation of most of the base classes may be attributed to Kondo Gnanvo [31].

The user code for each AMORE module resides in a well-defined file structure that contains the folders publisher, subscriber, common and ui (Figure 3.6). The agents and clients run the user code by implementation as a finite state machine. This paradigm can be thought of as replacing the main() function of a standard C++ code by a discreet set of methods that are called in a fixed order. Details of the finite state machine can be found in Figure 3.7. The developer is then tasked with extending the AMORE class "PublisherModule" by overwriting these methods called by the finite

state machine. The subscriber portion of the code is implemented by extending the class "VisualModule" and overwriting the methods. The items published to the MySQL database may be stored for later viewing via the AMORE archiver, but this is not implemented at this time for use in this study. The software developments described in this thesis provide methods for tracking and alignment used for muon tomography with GEM detectors. There have also been other various improvements and bug fixes throughout the pre-existing code.

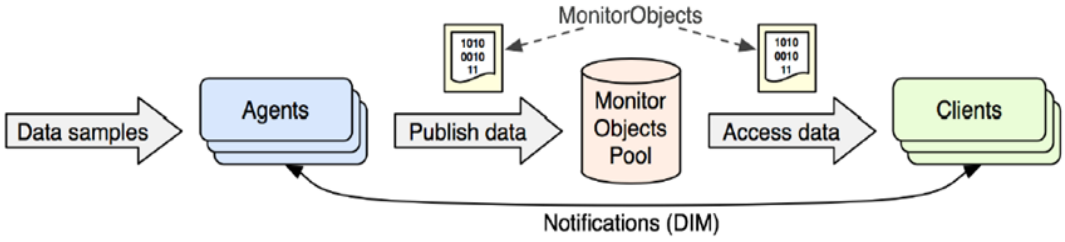


Figure 3.5 The publisher-subscriber model in AMORE [24].

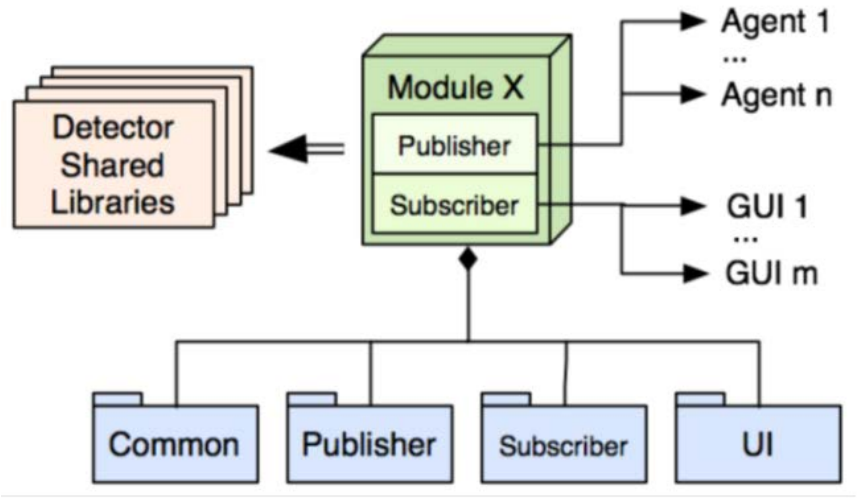


Figure 3.6 The structure of an AMORE module [24].

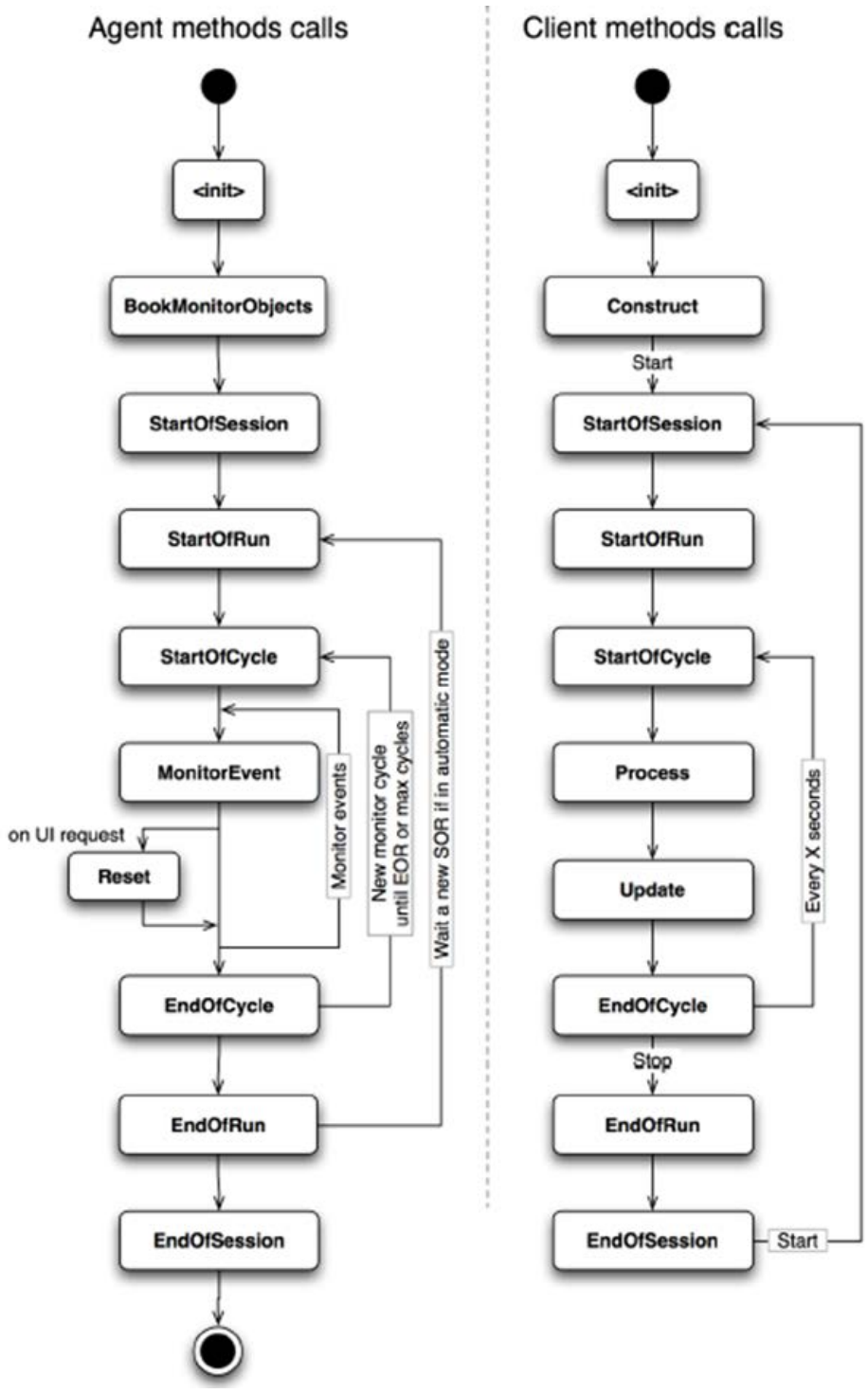


Figure 3.7 The AMORE finite state machine [24].

3.2.1 The amoreMTS module

The amoreMTS module was the first module written for commissioning of the SRS. It provides an extensible framework which handles the low level processing of the data as well as some higher level tracking and alignment. The "main" code for the publisher is contained in the class "MTSPublisher". Here the methods of "PublisherModule" are overwritten, and the finite state machine proceeds through the code as shown in Figure 3.7. There are five major run types that may be set via the run configuration file (see Appendix C). These are "RawPedestal", "Pedestal", "Cosmics", "POCAReco", and "Alignment".

3.2.1.1 Pedestal calculations

Since the data arrive from the FEC without zero suppression, the analysis code must do the work of suppressing any channels without signal. The RawPedestal run type calculates the common mode noise for all of the APVs used in the analysis. There is inherently some fluctuation in the average values between time frames due to external electronic noise. An example of raw data showing some major baseline fluctuations can be found in Figure 3.8. Each "slice" in this plot between the vertical lines contains the raw APV data for 128 channels during one 25 ns sample. The total number of such time samples sent from the APV per trigger can be set via the slow control. This has been decreased from 15 frames to 6 in order to decrease processing time. Six time frames have been found to be sufficient for tracking. Pedestal data is taken by setting the bias voltage on the detectors to 2000 V, which is below the level required for amplification. The RawPedestal run type uses this data to create a root file that is later used to correct for common mode fluctuations (Figure 3.8) in the final computation of the pedestal offsets and noise. This takes place during a run of type "Pedestal". The pedestal offset and RMS noise for the 128 channels for a typical APV can be found in Figure 3.9. This pedestal is then used for zero suppression of the raw data. The current trigger rate of the MTS is ~30 Hz, but zero suppression slows the analysis rate to ~3 Hz. There is development in progress by RD51 to support zero

suppression on the FPGA of the FEC. This will greatly decrease the processing time for the analysis, and should allow "real time" online analysis of the data.

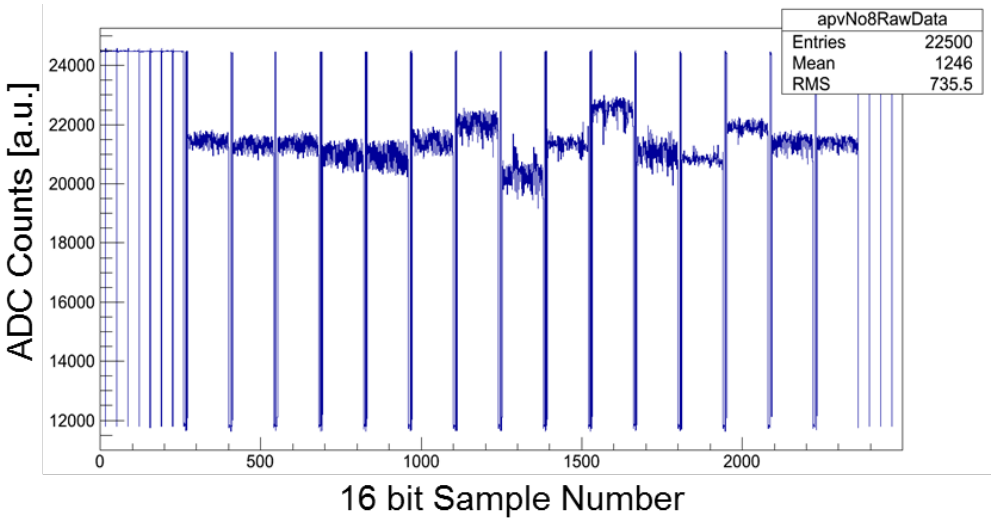


Figure 3.8 Raw APV data for a single event exhibiting large common mode fluctuations.

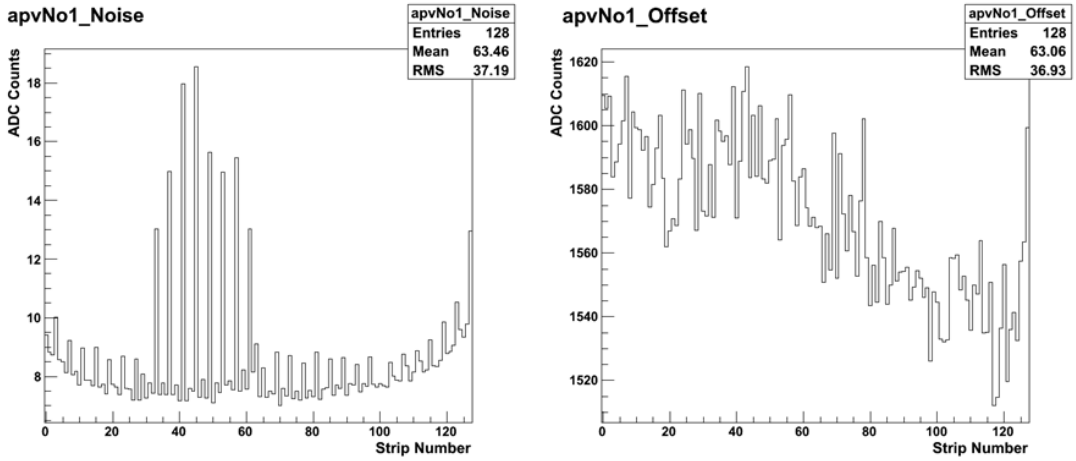


Figure 3.9 Example of computed pedestal offset and noise using the amoreMTS module for one APV.

3.2.1.2 Event building

The mapping of the APV hybrids to their locations on the different detectors is set via an additional configuration file (see Appendix C). Using this configuration file, the individual APVs are assembled into detector planes. The class "MTSDetectorPlaneEvent" takes all of the raw data from the APVs on a given plane and performs the zero suppression using the pedestal files generated by the Pedestal run type. The grouping of neighboring strips with non-zero data into strip clusters is done using this class. One can then retrieve the list of clusters for each plane and use these data to assemble tracks or it can be used to fill histograms requiring zero-suppressed data. The zero suppression process proceeds as shown in Figure 3.10. In the upper-left plot, the raw data for a single detector plane is shown. In the upper-right plot, the pedestal offsets have been subtracted. This is then followed by zero suppression by removing all strips with a signal less than five times the RMS pedestal noise on the strip. The time evolution of the zero-suppressed signal may be found in the lower-right plot. The Cosmics run type is generally used for detector characterization in amoreMTS. The monitor objects that are to be published to the MySQL database may be set via an additional configuration file (see Appendix C). In the case of amoreMTS, these monitor objects are 1D, 2D, and 3D ROOT histograms that show quantities of interest for detector characterization and monitoring of the data. These histograms are filled in the class "MTSHistoManager". Examples of these detector characterization plots that can be produced can be found in Figures 3.10-3.12.

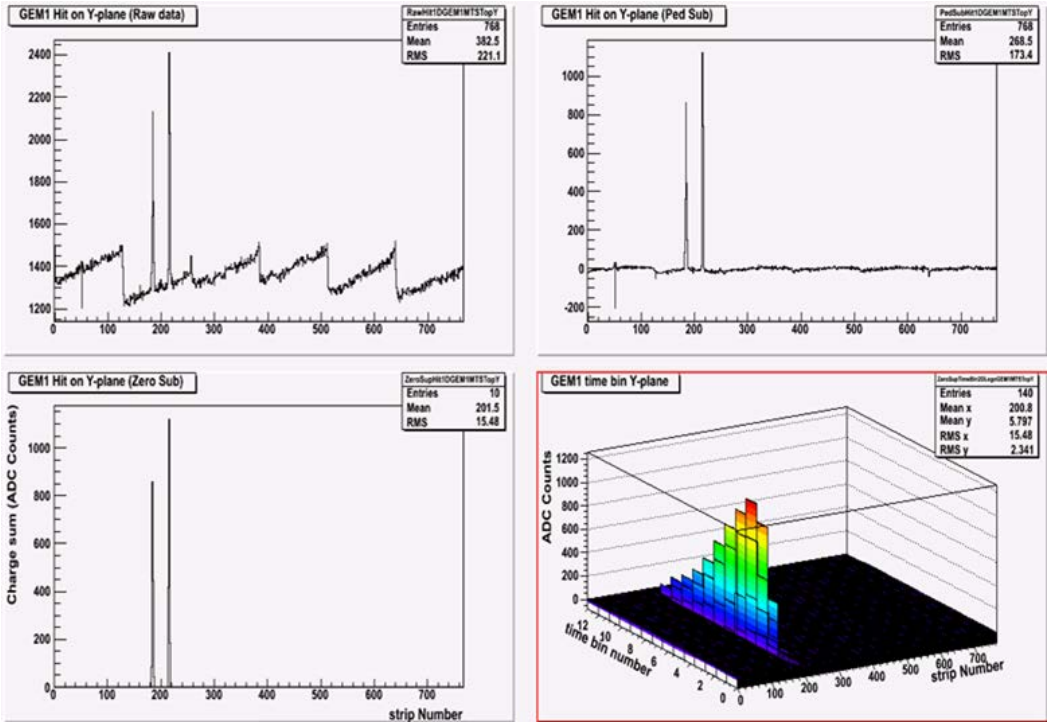


Figure 3.10 The zero suppression process in the amoreMTS and amoreSRS modules.

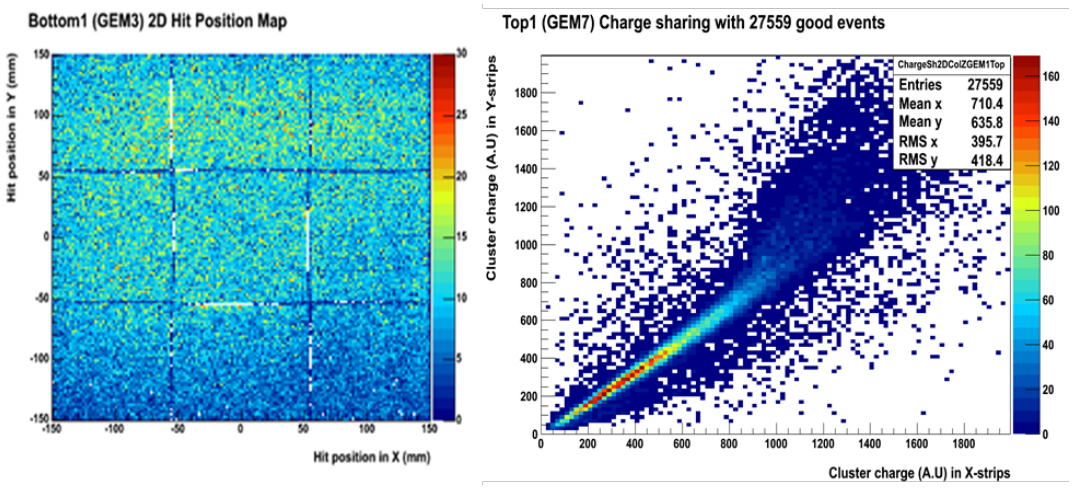


Figure 3.11 Hit map for a GEM detector in the MTS (left). The loss of efficiency near the spacer frames is clearly visible. The charge sharing between the X and Y strips (right) shows unequal charge sharing with more charge on the top strips (~0.8 bottom/top charge sharing ratio).

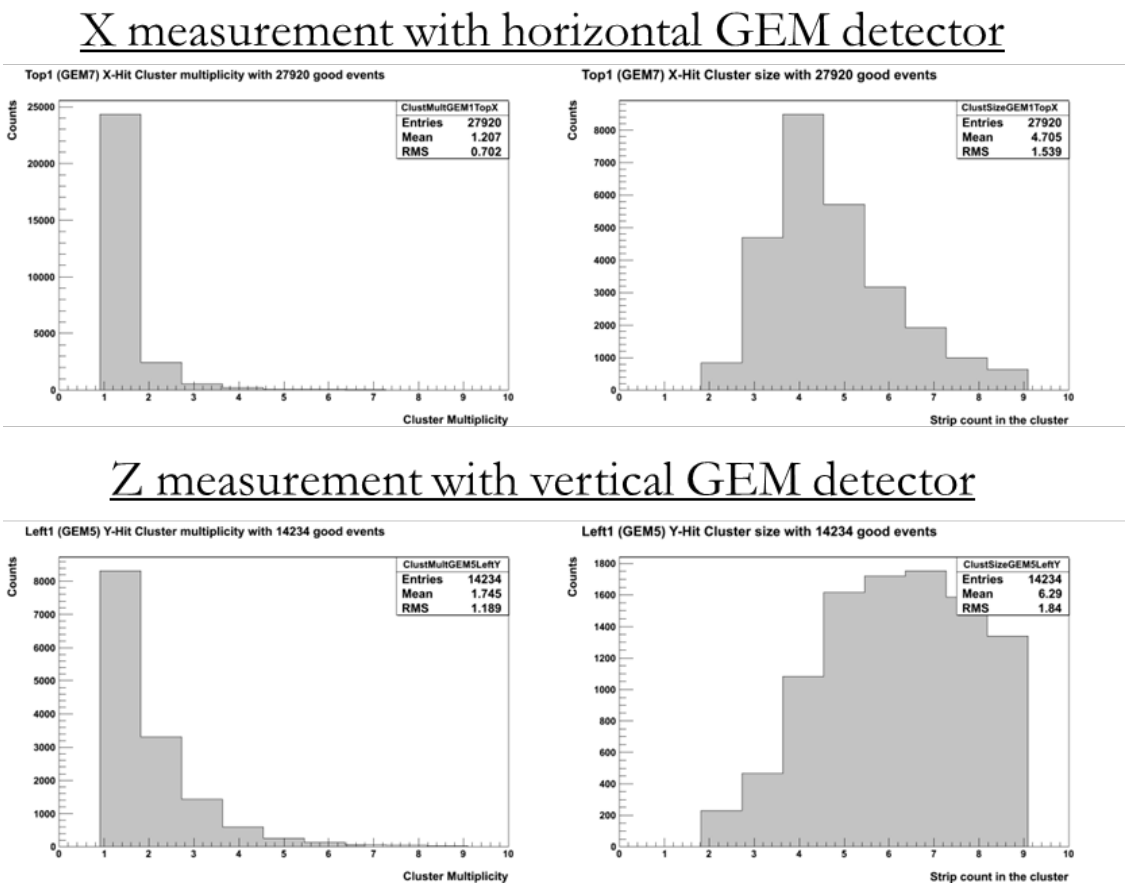


Figure 3.12 Strip cluster multiplicity (left) and cluster size (right) for a horizontal and a vertical detector plane measuring in global X and in global Z, respectively. The highly angular tracks in the vertical measurement cause an increase in the mean of both quantities.

For application in muon tomography, individual clusters must be assembled into tracks. These tracks can then be used for offline alignment, as well as for POCA reconstruction of the different scenarios. As can be seen in Figure 3.12, a typical detector plane has a mean strip cluster multiplicity higher than 1. The result is quite similar for all detectors measuring in a specified global coordinate. This total number of clusters per event is greater than one in ~15% of measurements in the horizontal X and Y axes, and ~40% for measurements in the vertical Z axis. This brings about the

problem of selecting what combination of clusters should be assigned to the muon track. The problem of track selection and pattern recognition is a very well-studied problem in high-energy physics experiments where the track multiplicity is typically high [32]. Methods are used to detect patterns in the data in order to assign the many position measurements from the different detectors in the experiments to tracks. In the case of the MTS, we assume that there is only one true track per event. This may not be true in the event of a cosmic-ray air shower, but the number of these events is negligible in comparison to the single muon events. While this simplification makes the task of track selection easier, the lack of redundancy in the measurements along a track limits the available methods for track selection and the rejection of "ghost hits" that do not belong to the track. Typically one would employ a method such as a Hough transform or track following methods to identify a track [32], however these methods require many measurements along the track to be made. In the case of the MTS, at the current time there are only two 2D measurements in each tracking module. This is enough to create a 3D track, but limits the available methods that can be used for rejection of ghost hits.

3.2.1.3 Tracking

In amoreMTS, the tracking framework is very simplistic. For each detector there are two detector planes, each measuring one projection of the 2D measurement, taken as the local X and local Y position measurement. These are then translated to the global 3D MTS coordinates. Often there is more than one cluster in a given projection, as seen in Figure 3.12. Initially, an algorithm was utilized that selected the cluster with maximum charge for all detectors contained in a given track. Unfortunately, this method often does not select the proper cluster. The selection of tracks for amoreMTS now relies on the cleanest of events: those with only one cluster in each plane of all detectors crossed by the track. With this strong cut we are limited to 4-6% of all recorded triggers being selected as "good events" that contain enough measurements that pass selection to be reconstructed. Monte Carlo simulation

estimates that based on a geometric acceptance, ~17% of recorded events triggered by coincidence in the plastic scintillator paddles could possibly be selected as good events with perfectly efficient detectors. Using a reasonable estimate of 96% efficiency per detector (typically 98% efficient detector with 2% loss due to spacer frames [12]) and requiring four detectors in a track, this estimate drops to ~14%. The development of a second module "amoreSRS" for tracking was motivated by this loss in acceptance caused by these stringent cuts on the data.

3.2.2 Improved tracking with the amoreSRS module

The amoreSRS module was originally developed as a lightweight version of the amoreMTS module for more general use with the SRS. Thus it served as an excellent starting point for new developments in the tracking code. Many of the pre-existing elements of amoreMTS are still included such as pedestal calculation and production of monitoring histograms. There have been several improvements to the code, including Gaussian fitting of the strip clusters to determine the position of a cluster, as well as enhancements in the configuration file (see Appendix C) allowing full rotation of the detector modules for tracking. In this code, the Alignment run type from amoreMTS has also been expanded to include histograms of the 3D residual measurements for straight tracks. The usefulness of this data is explained in the next chapter.

The process of track selection in amoreSRS is handled combinatorically. This method can be extremely resource intensive in the case of a large number of track samples. However, in the simple case of the MTS it works quite well. Several classes have been added to the code in order to improve the handling of data. In the new code, there are no stringent cuts on the number of clusters per detector plane. A SRSStationEvent object is used to contain the SRSDetectorEvents within a given two-detector tracking module (top, bottom, left, right). The SRSDetectorEvent contains all of the SRSDetectorHits on a given detector. The class SRSTrackBuilder then

assembles all of the detector hits within a tracking module into SRSTrackCandidates. All possible combinations of detector hits in all detectors within a tracking module are considered. SRSTrackBuilder then creates all combinations of track candidates between two different tracking modules. A simple illustration of this method may be found in Figure 3.13. The total number of track candidates considered per event is shown in Figure 3.14. Here it can be seen that the higher strip cluster multiplicity in the side detectors causes a higher number of track candidates on average.

The selection of the final track for this thesis is based on the selection of the pair of track candidates with the lowest distance of closest approach (DOCA) and requiring that DOCA < 1 cm for that track. A typical distribution of the minimum DOCA per event is shown in Figure 3.15. The distribution of polar angles for accepted tracks using the amoreSRS module can be found in Figure 3.16. A sample of five data sets processed using both the single cluster and combinatoric track selection show a 32% increase in the number of "good events" passing selection. This package has been used for some of the POCA reconstructions as indicated in Chapter 5.

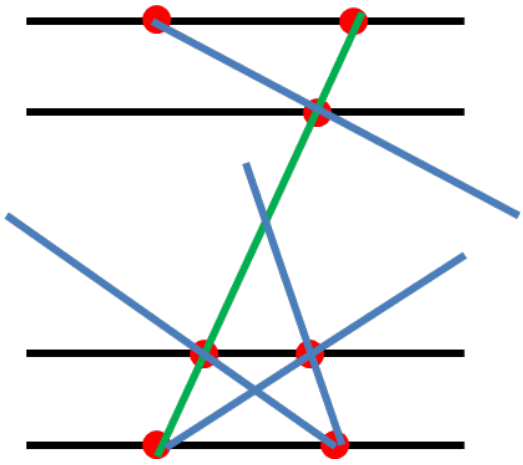


Figure 3.13 Simplified 2D illustration of combinatoric track selection. All track candidates are assembled within each tracking module, then the best track (green) is selected from the combination of these track candidates.

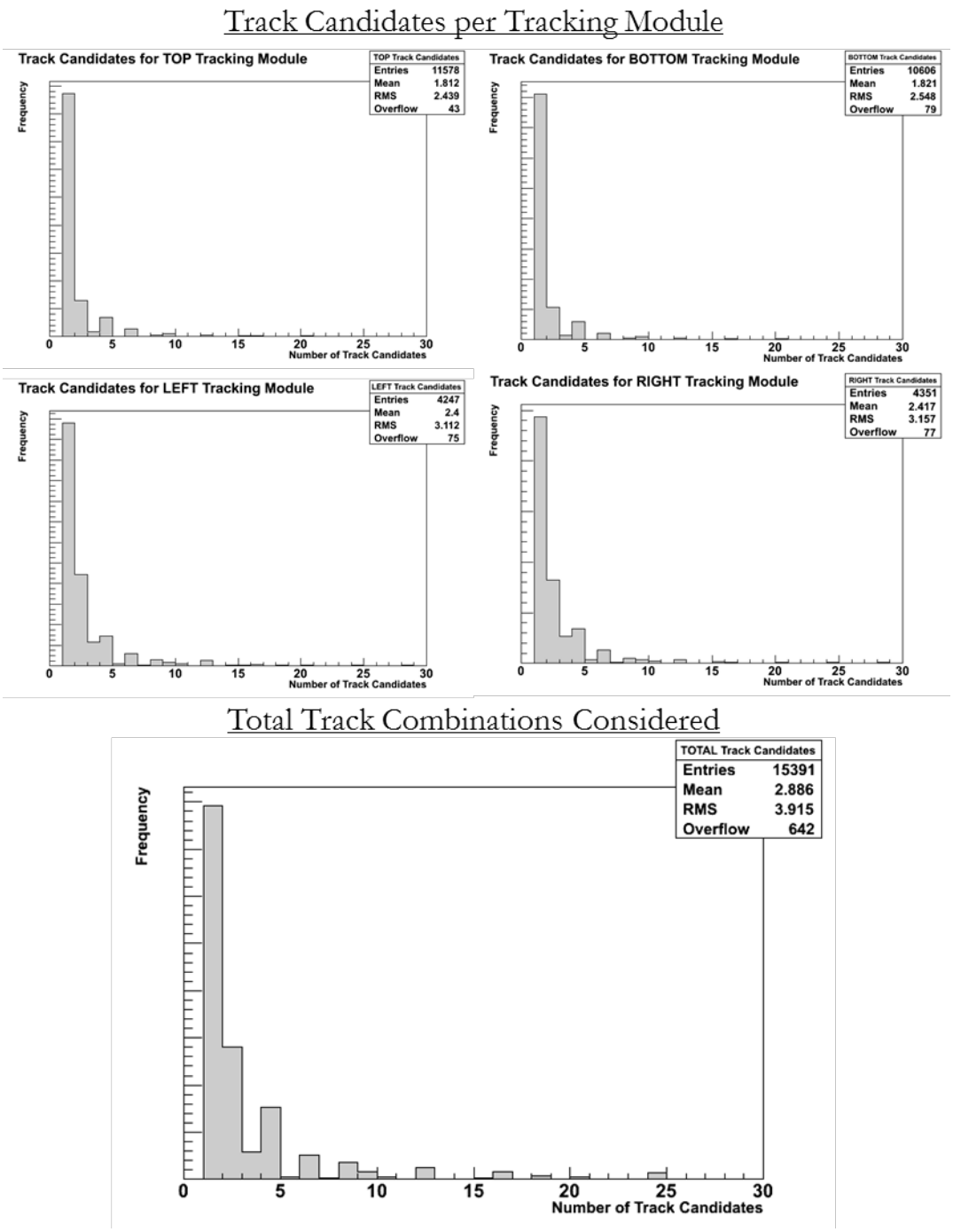


Figure 3.14 Total number of track candidate considered per tracking module per event (above) and the total number of tracks considered per event (below).

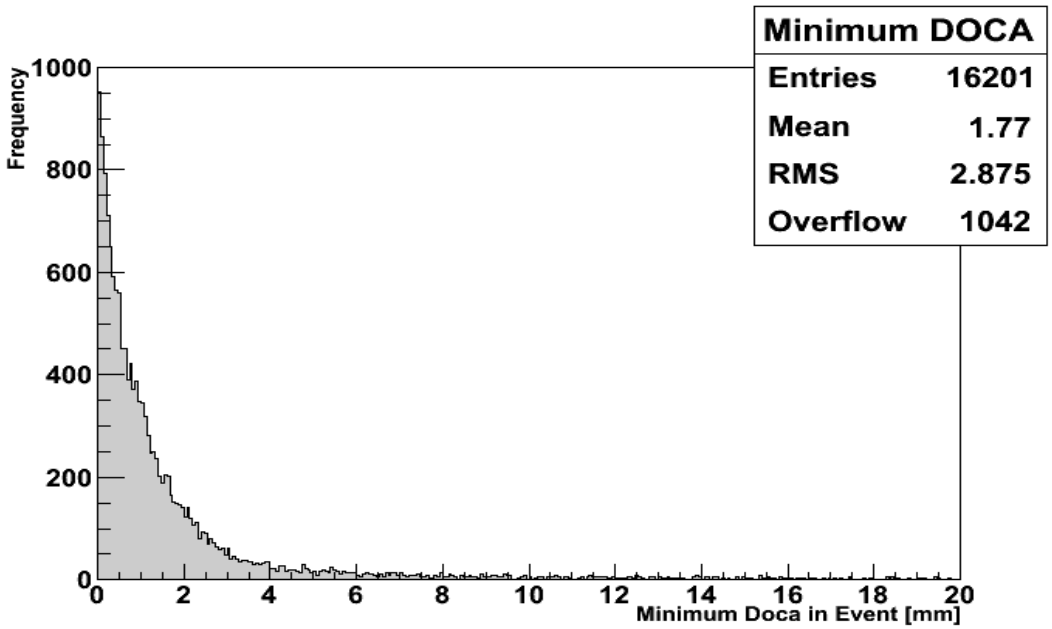


Figure 3.15 Minimum DOCA per event. A cut is made to reject tracks with DOCA > 10 mm.

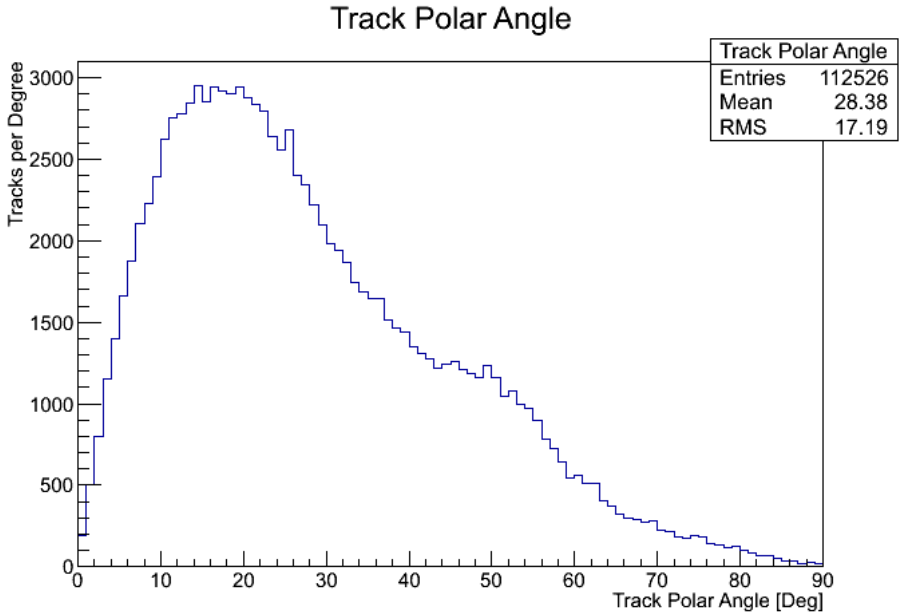


Figure 3.16 Distribution of polar angle for straight tracks through the MTS.

Chapter 4

A Cubic-Foot Muon Tomography Station

4.1 Muon Tomography Station design

There were several factors that were considered in the design of the muon tomography station. The station should allow for modular detector placement, allowing the user to select different detector orientations and spacing configurations. The support structure should be designed so as to minimize the material traversed by a measured muon to minimize multiple scattering within the station and detector materials. There should also be ample room for other aspects of the infrastructure such as gas lines and cables for the DAQ electronics. The design for the cubic-foot muon tomography station was developed mostly by Leonard Grasso, and details of the development are presented in his thesis [33]. An image of the cubic-foot MTS can be found in Figure 4.1.

The design of the station is basically an aluminum frame which supports PVC plates that hold the GEM detectors in place. The detectors are then grouped into four separate tracking modules on the four different sides of the station enclosing the imaging volume. The number of detectors within each module is variable. At minimum, two GEM detectors are required for measuring a 3D track within a module. However, it is preferable to have more than two detectors within each tracking module to aid in the selection of tracks and the rejection of noise or "ghost hits" in the data. For the experimental data in Chapter 5, only two GEM detectors were used in each of the tracking modules. Unfortunately, the four-channel CAEN N1470 power supply used to supply the bias voltage to the GEMs is limited to 8W per output. At an operating voltage of 4.2 kV each GEM detector draws roughly 3.2W of power. Thus

we are limited to powering eight total GEM detectors until an additional power supply is obtained.

There are a few design flaws that should be recognized to aid in the design of later stations. PVC was chosen as the material for the support plates so as to minimize the multiple scattering caused by muons passing through the station materials. The area of the PVC plate below the active area of the GEM detector has been cut out so

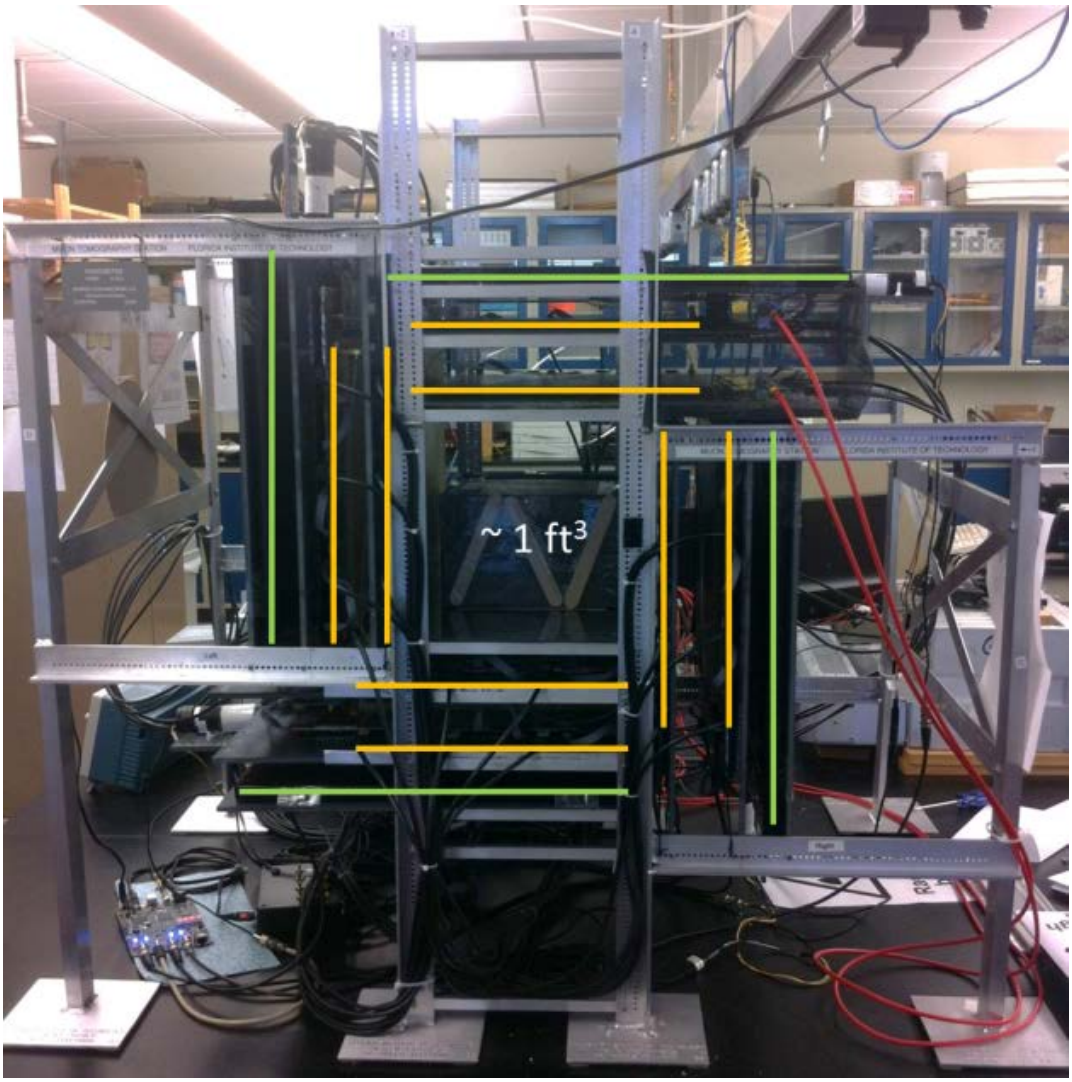


Figure 4.1 The cubic-foot MTS. Yellow lines indicate GEM detector locations; green lines indicate the position of plastic scintillator paddles for fast triggering.

as to minimize the material budget and multiple scattering in the tracking modules. In future iterations of MTS design, it is recommended to avoid any flexible support structure such as these. The horizontal PVC plates flex under the weight of the detector and are sensitive to fluctuations in the ambient temperature of the lab. In addition, while some slight misalignment is unavoidable in any tracking experiment, the flexibility of the material allows for an unacceptable uncertainty in the positioning of the detectors. Due to the bowing of some of these plates, simply measuring the location of the PVC plate is insufficient to ascertain the position of the GEM detector itself. This large uncertainty in the position of the detector modules has a definite negative impact on the imaging performance. The design of the aluminum support frame is also quite thin, and is susceptible to various types of misalignment from deformation of the frame. These deformations include bending of the feet of the station and torsions to the frame caused by repeated shipping to and from CERN. In the following sections of this chapter, different methods are discussed that were used to try to improve the quality of the reconstructed images using different software alignment methods.

4.2 Alignment of GEM detectors with tracks

The high intrinsic spatial resolution of the GEM detectors used in the MTS is degraded by uncertainties in the positions and orientation of the detectors. To take full advantage of intrinsic spatial resolutions of 50-200 μm, it is necessary that the uncertainty in alignment be negligible in comparison with the intrinsic resolution. This can be achieved with a combination of hardware and software alignment methods. Hardware alignment is performed first which includes survey measurements of detector positions and orientations. There are also advanced hardware alignment systems such as the muon alignment system of CMS [34]. Hardware alignment methods for CMS limit the uncertainty in the nominal detector positions to the level of 100-500 μm [35]. During the design of the cubic-foot MTS it was envisioned that simple measurements of the detector positions would be able to achieve an uncertainty of ~500 μm in the detector position. Unfortunately, due to the problems discussed in the previous section, making a measurement of the detector positions and orientations of this precision from simple survey becomes nearly impossible.

Software alignment methods are used to provide a more precise alignment of the detector modules. They rely on using tracking information to achieve the goal of negligible alignment uncertainty. There are several track-based alignment methods that have been developed for high-energy physics [36]. Most of these methods rely on minimizing the residual distance between the hit locations and the impact points of a fitted track crossing the detector. Two different methods have been investigated for use with the cubic-foot MTS, namely the Histogram method and the Kalman filter method. It is important to note that there are other promising methods that have not been investigated for use with the MTS in this thesis and may be an interesting topic for future work. For instance, the Hit and Impact Point (HIP) method [37] provides an alternative method that is perhaps easier to understand than the Kalman filter alignment presented in this thesis. Since the number of alignment parameters is quite

small, methods such as iterative alignment using gradient descending minimum search may also be a viable option [38].

4.2.1 The Histogram method

A naming convention has been established for the different detectors in the station that will be used frequently in the discussion of the alignment. The different GEM detectors are numbered from top to bottom and then from left to right. Descending in Z we have GEM1 and GEM2 in the top tracking module followed by GEM3 and GEM4 in the bottom tracking module. The naming then proceeds from left to right in the positive Y direction, through GEM5 and GEM6 in the left tracking module and GEM7 and GEM8 in the right tracking module. This naming scheme has been depicted in Figure 4.2. The "sensitive coordinate" is often discussed and refers to a global coordinate directly measured by a detector. For instance, the top and bottom detectors measure the global X and Y position of a track, while the side detectors measure the global X and Z position of a track.

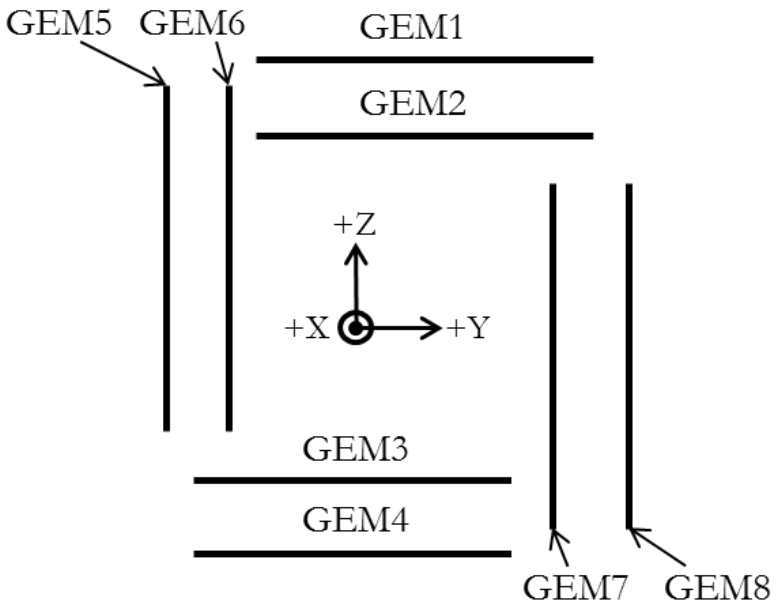


Figure 4.2 A schematic view depicting the detector naming convention. The 3D global MTS coordinate system is shown. The +X axis points out of the page.

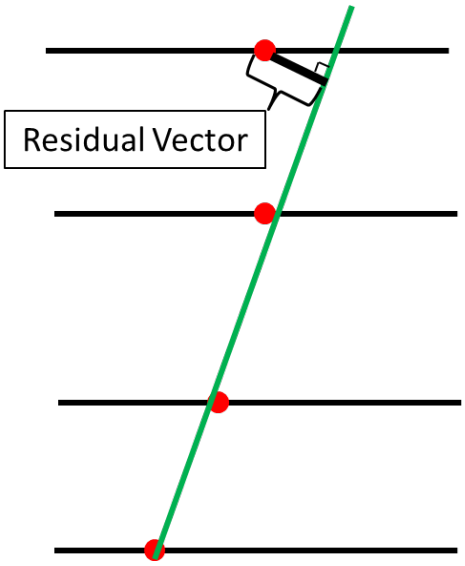


Figure 4.3 A simplified 2D illustration of the unbiased track fit method. The X, Y, and Z components of the residual vector are added to the residual distribution histograms.

As the number of tracks considered increases, a distribution begins to form that should be roughly Gaussian, reflecting the intrinsic spatial resolution of the detector convoluted with that of the other detectors in the track. Misalignments of the detectors can cause an increase in the width of the residual distribution or cause deviations from the expected symmetric Gaussian shape. With perfect alignment, these distributions would be nearly perfectly symmetric with zero mean. Without any alignment by tracks there is typically some offset in the mean of this distribution. Plots of the residual distributions before and after iterative alignment for all eight GEM detectors using the histogram method can be found in Figure 4.5. The final alignment parameters used for the analysis presented in this thesis can be found in Appendix C. The impact of this alignment on the POCA reconstruction is clearly shown in Figure 4.6. The POCA reconstruction method is discussed in more detail in Chapter 5.

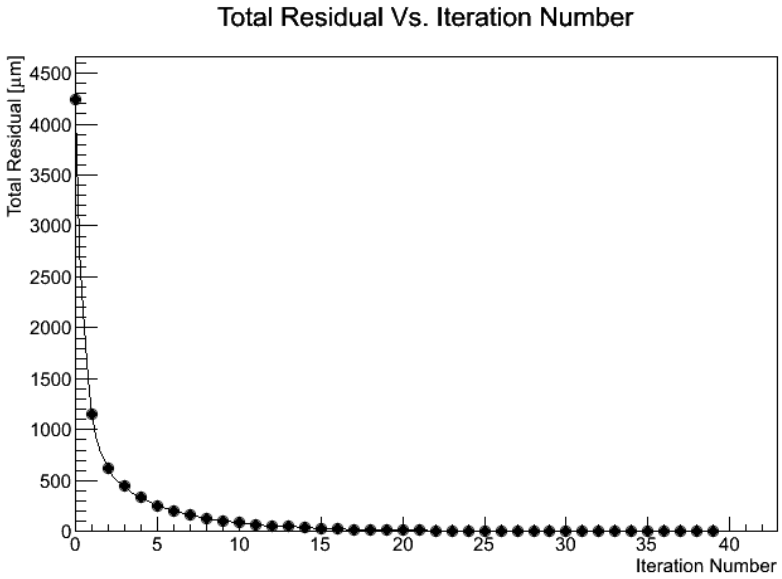


Figure 4.4 Total residual for all detectors versus iteration number.

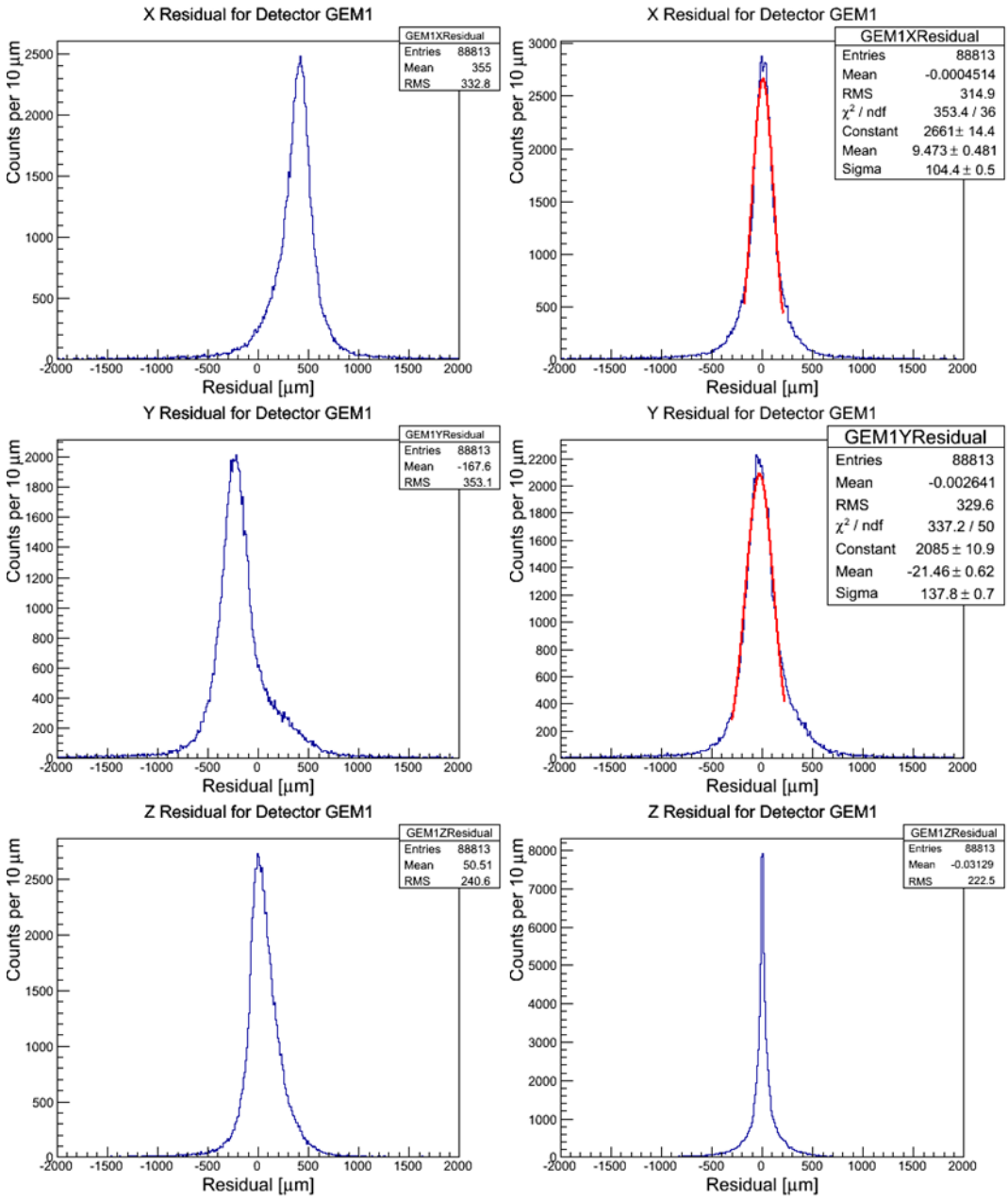


Figure 4.5 Unbiased global X, Y and Z residual distributions for GEM1 before (left) and after iterative alignment (right). The central regions of the distributions for the sensitive coordinates after alignment are fit with a Gaussian.

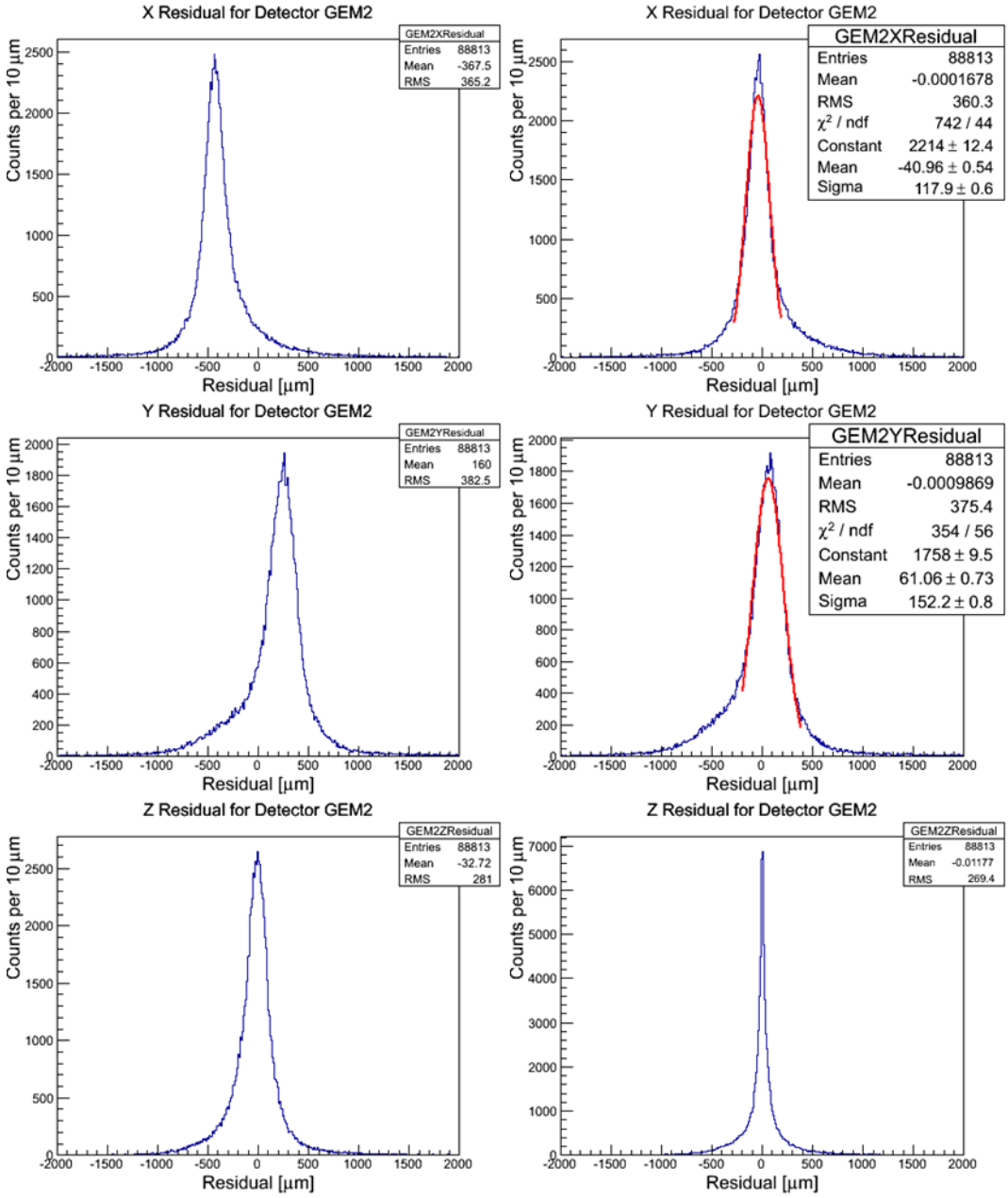


Figure 4.5 (cont'd) Unbiased global X, Y and Z residual distributions for GEM2 before (left) and after iterative alignment (right). The central regions of the distributions for the sensitive coordinates after alignment are fit with a Gaussian.

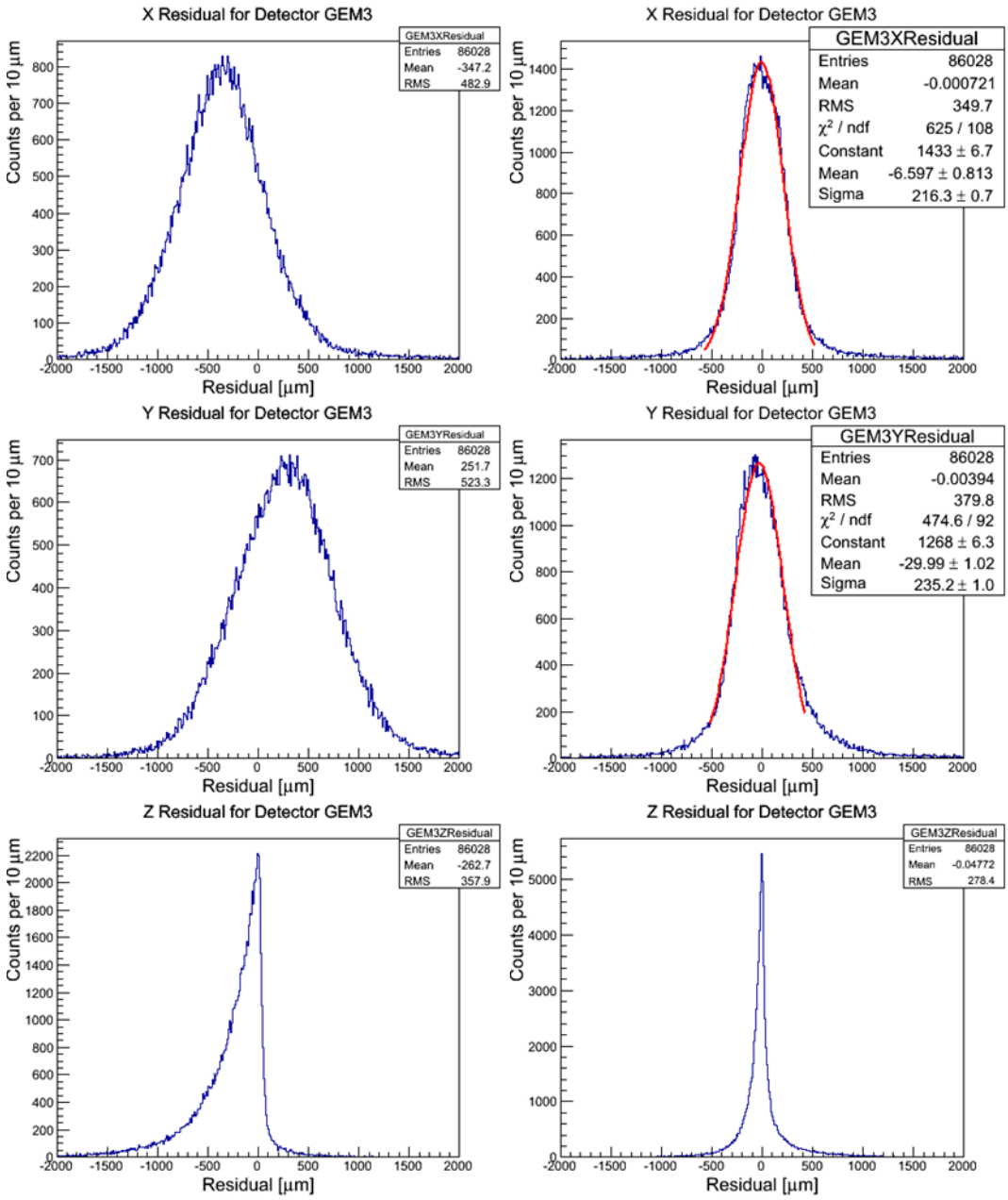


Figure 4.5 (cont'd) Unbiased global X, Y and Z residual distributions for GEM3 before (left) and after iterative alignment (right). The central regions of the distributions for the sensitive coordinates after alignment are fit with a Gaussian.

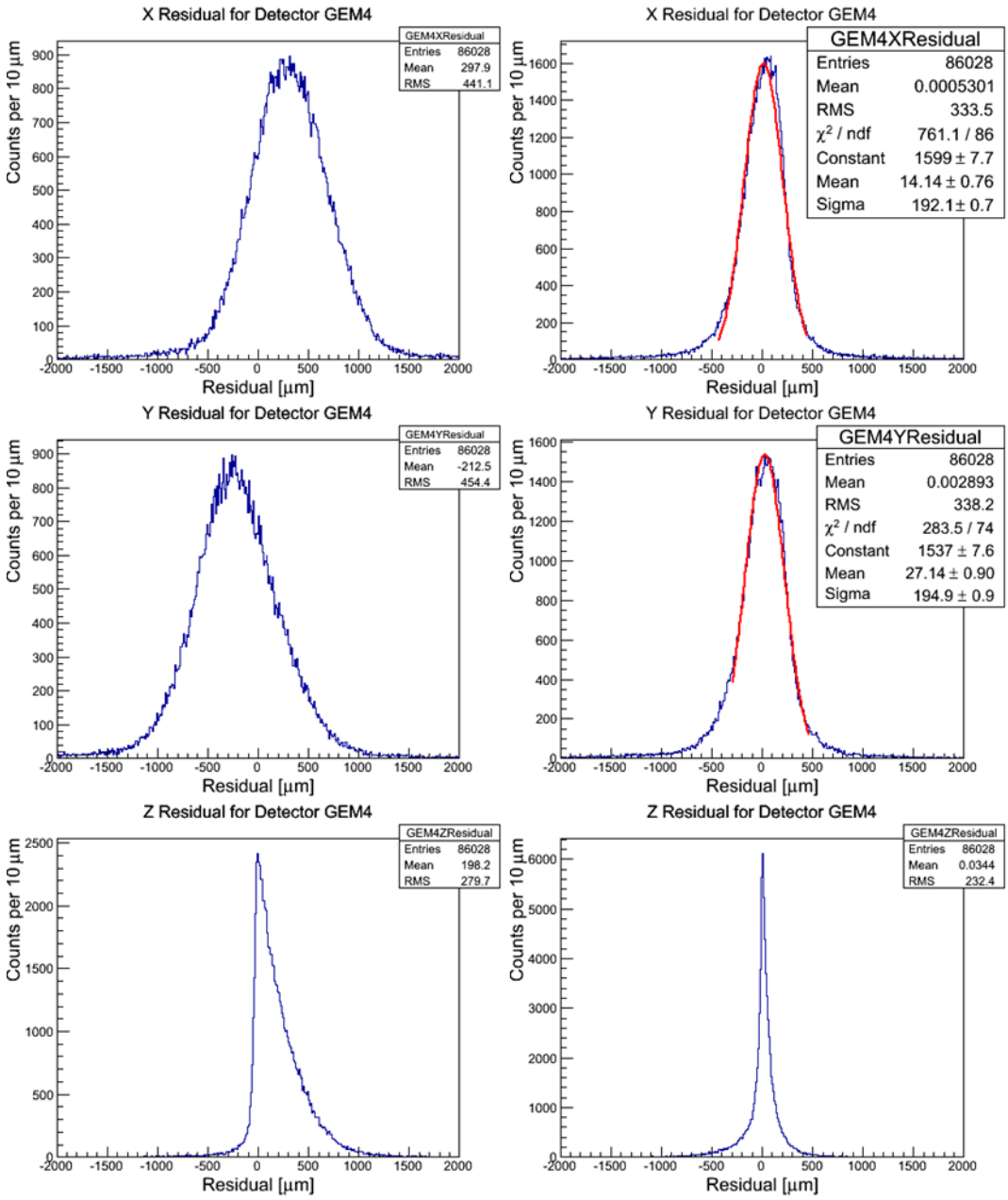


Figure 4.5 (cont'd) Unbiased global X, Y and Z residual distributions for GEM4 before (left) and after iterative alignment (right). The central regions of the distributions for the sensitive coordinates after alignment are fit with a Gaussian.

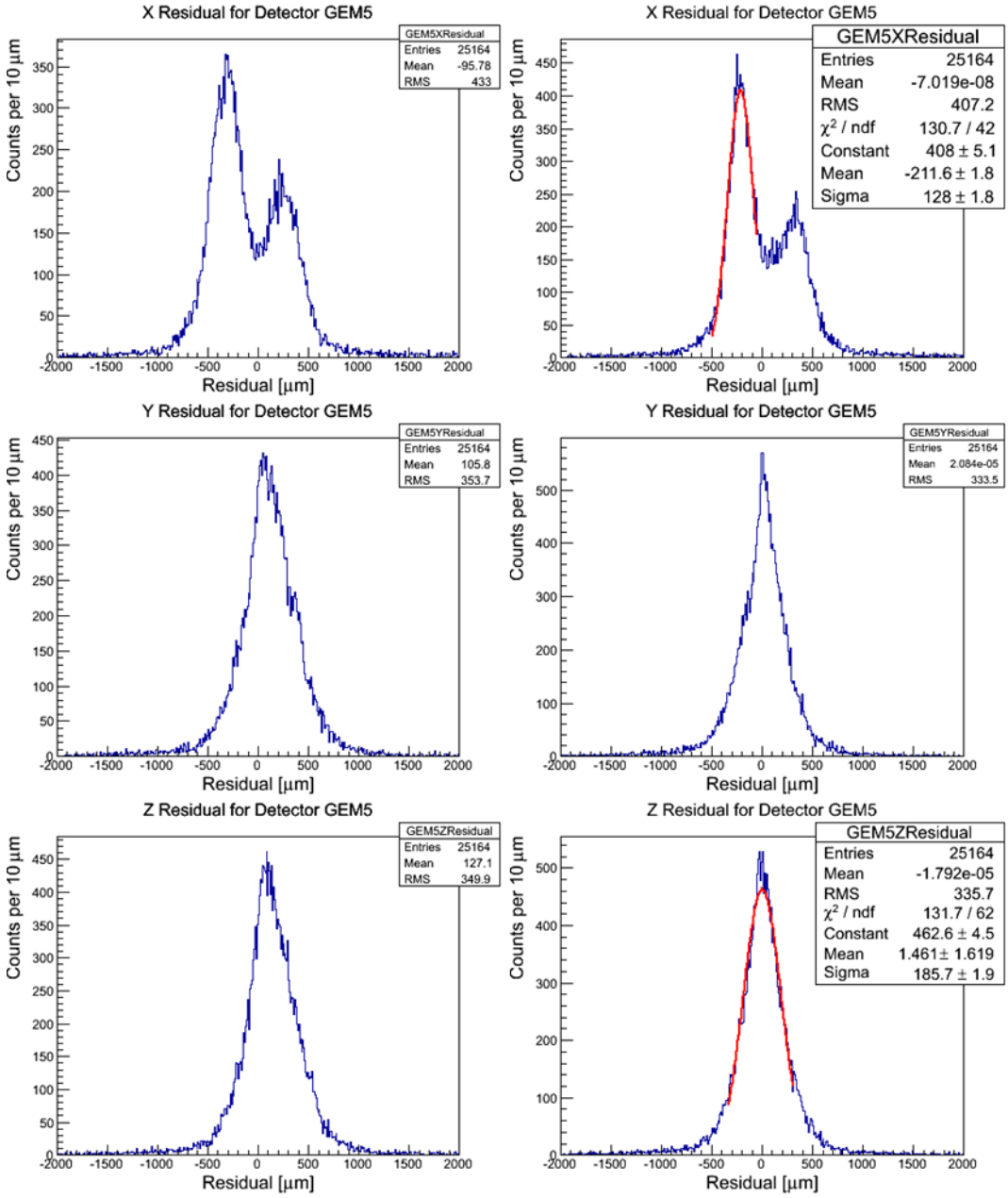


Figure 4.5 (cont'd) Unbiased global X, Y and Z residual distributions for GEM5 before (left) and after iterative alignment (right). The central regions of the distributions for the sensitive coordinates after alignment are fit with a Gaussian.

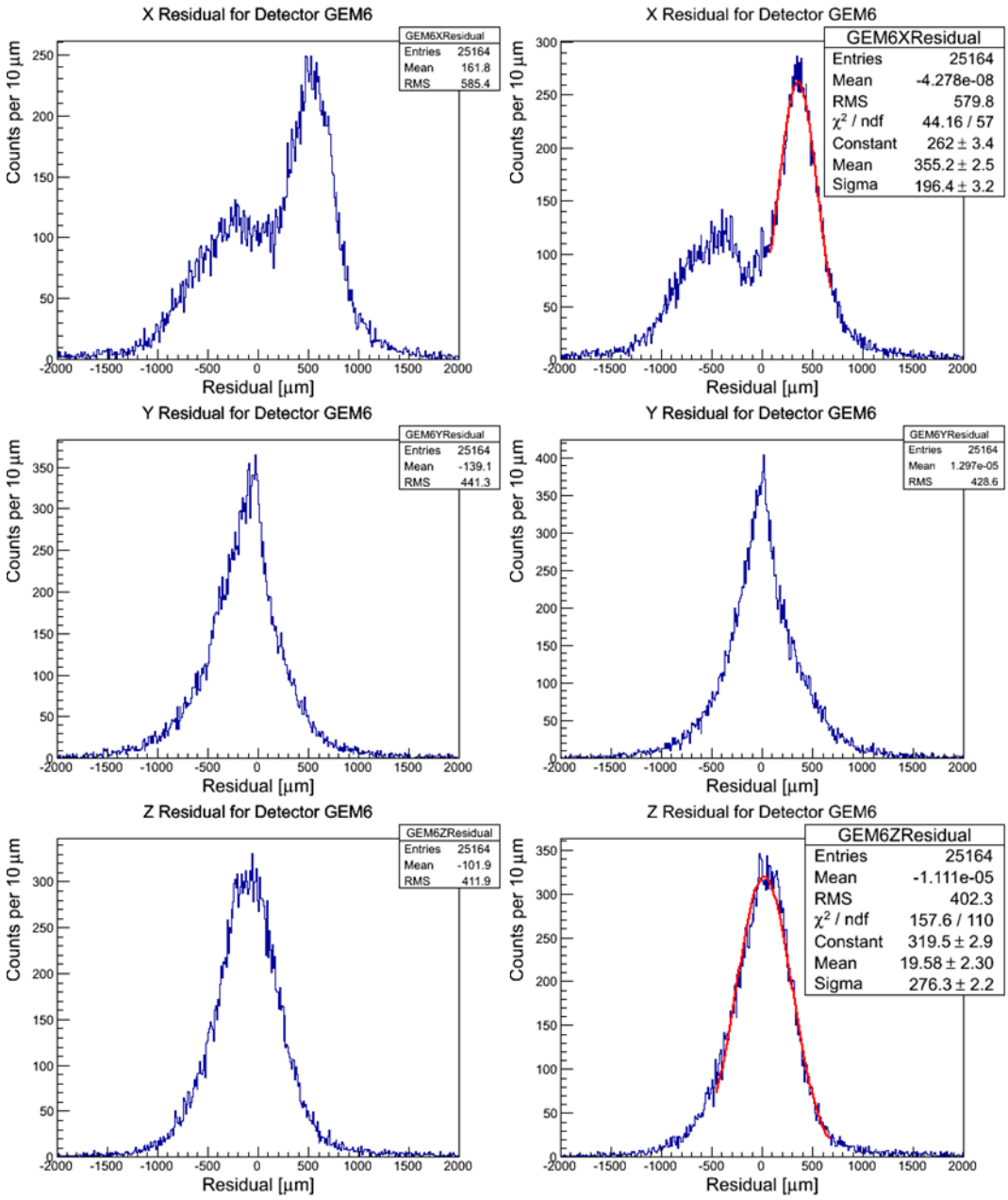


Figure 4.5 (cont'd) Unbiased global X, Y and Z residual distributions for GEM6 before (left) and after iterative alignment (right). The central regions of the distributions for the sensitive coordinates after alignment are fit with a Gaussian.

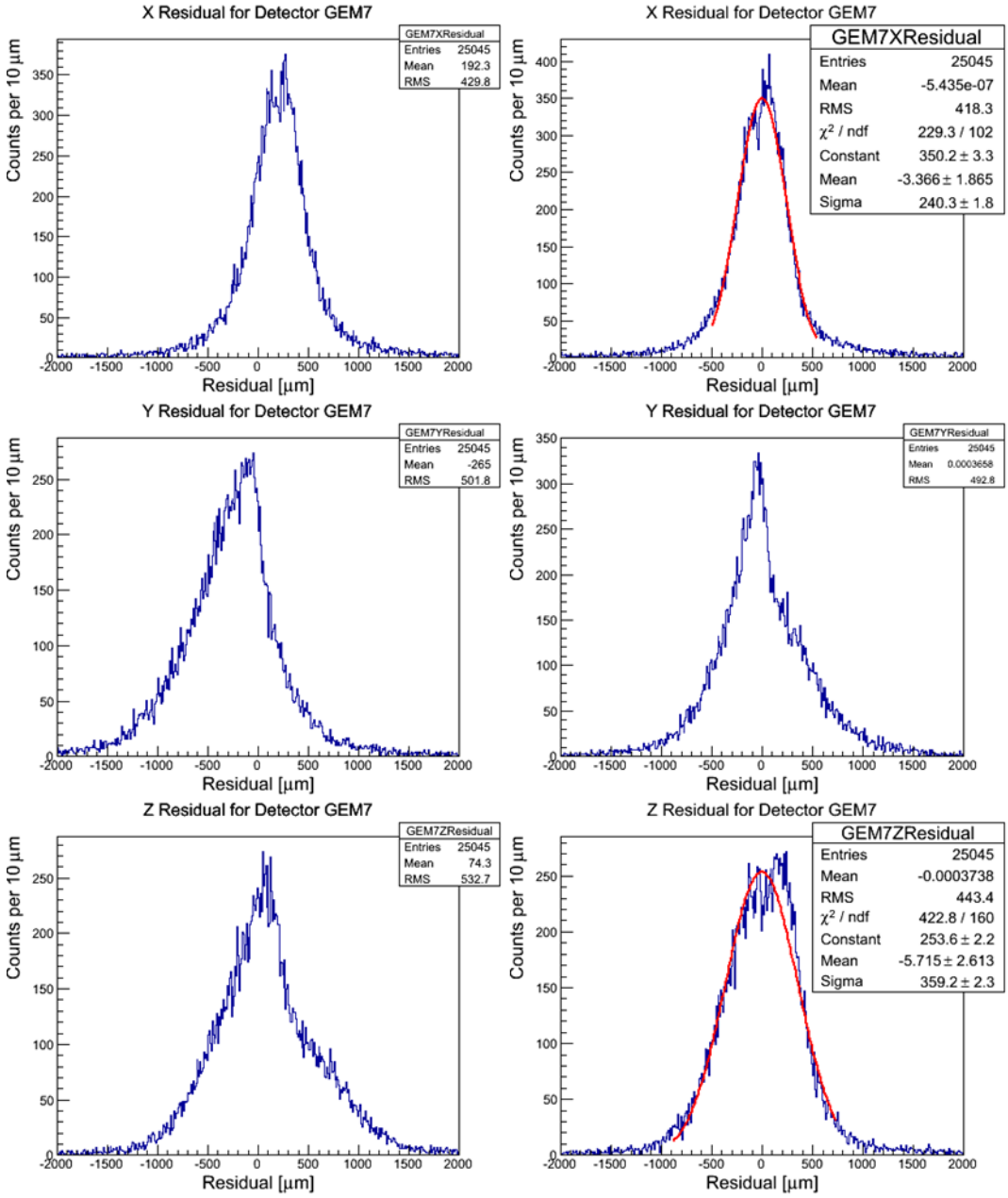


Figure 4.5 (cont'd) Unbiased global X, Y and Z residual distributions for GEM7 before (left) and after iterative alignment (right). The central regions of the distributions for the sensitive coordinates after alignment are fit with a Gaussian.

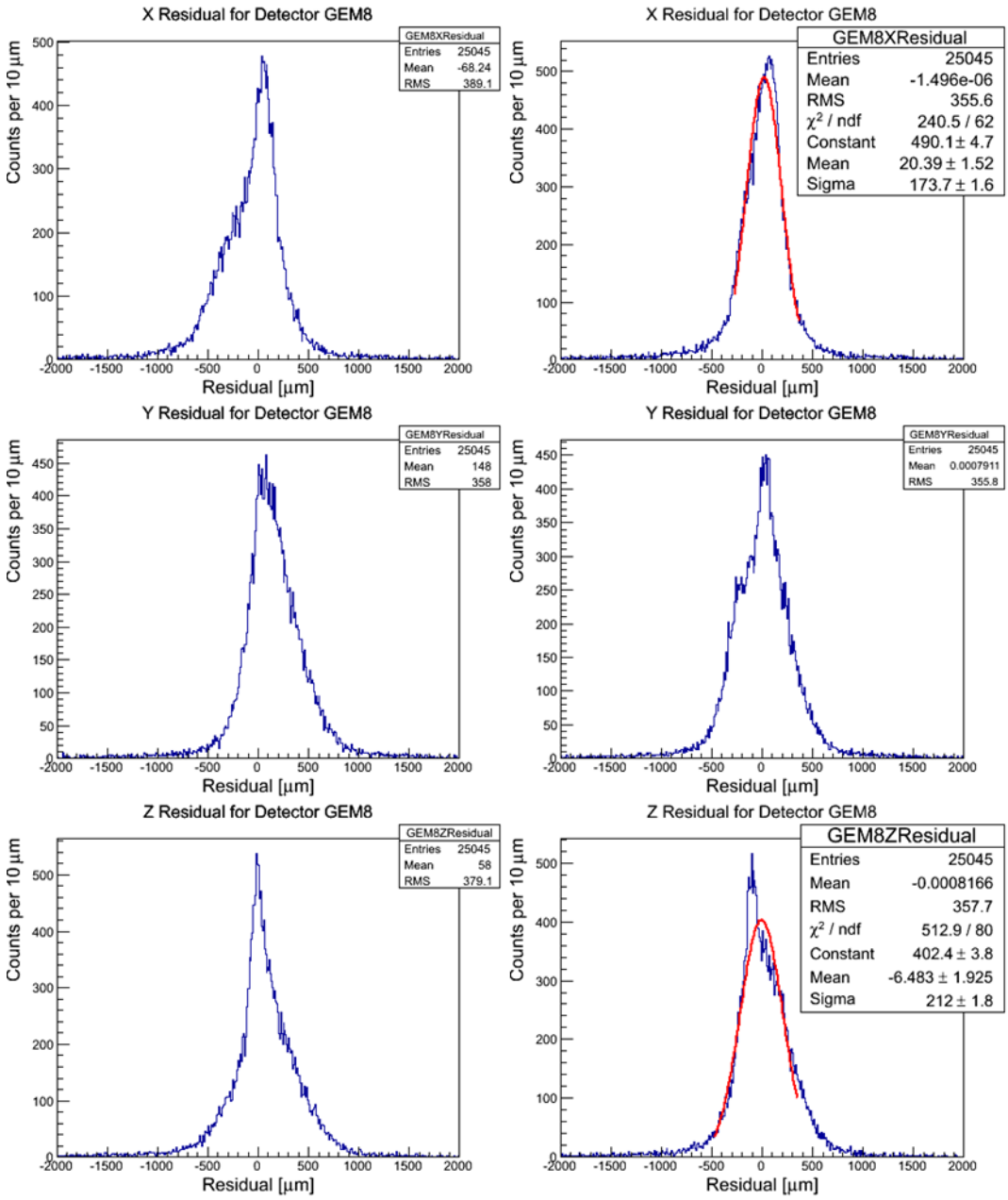


Figure 4.5 (cont'd) Unbiased global X, Y and Z residual distributions for GEM8 before (left) and after iterative alignment (right). The central regions of the distributions for the sensitive coordinates after alignment are fit with a Gaussian.

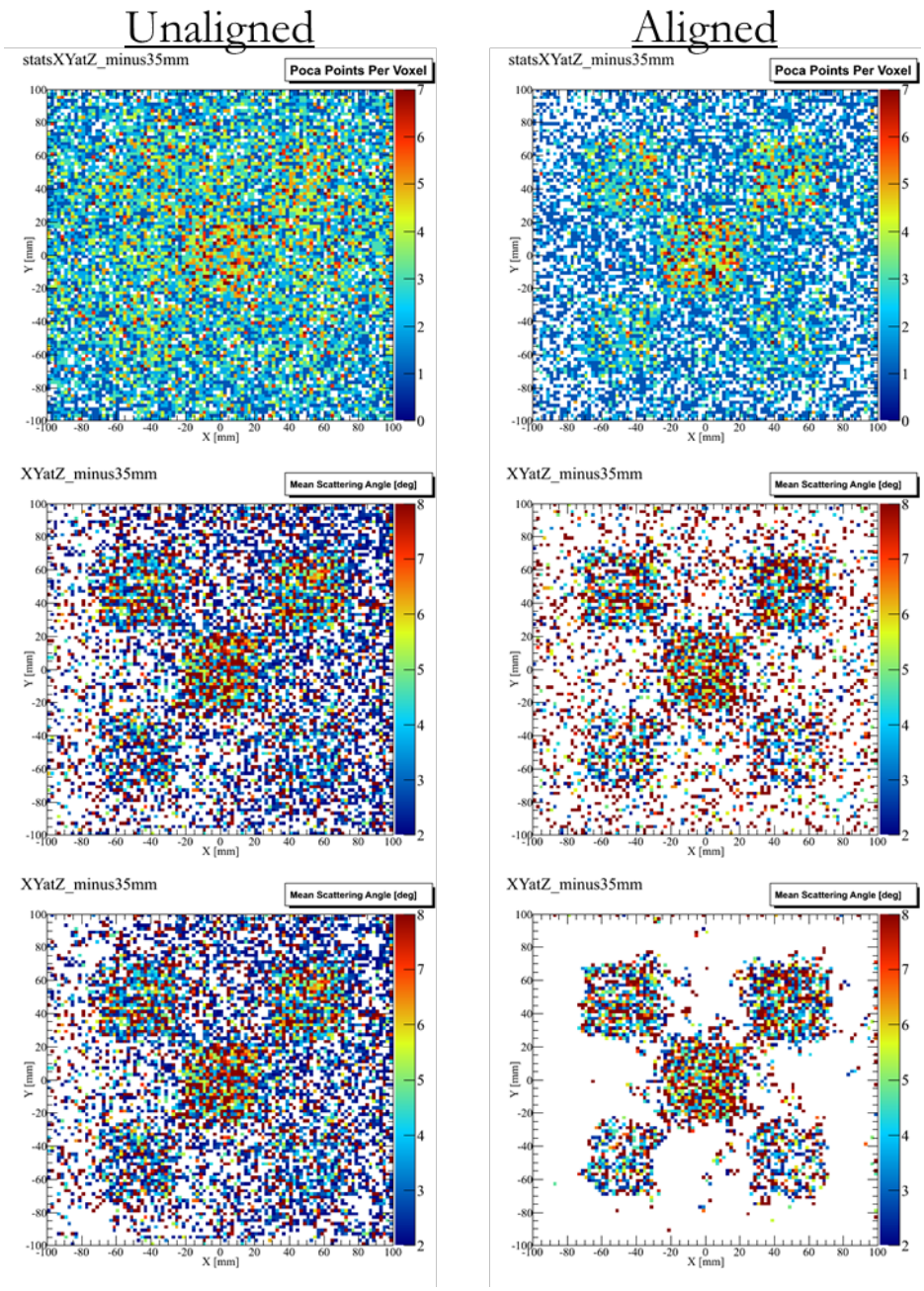


Figure 4.6 Effect of alignment on the result of POCA reconstruction with 155,104 tracks. The number of POCA points per voxel is shown in the top plots. The mean scattering angle per voxel is plotted without (middle) and with (bottom) post-processing to remove noise. Voxel size is 2 mm x 2 mm x 40 mm.

Using the histogram method, it is quite simple to align parameters directly accessible from the residual histograms. We can iteratively correct the alignment based on the means of the residuals with some success as shown in the plots of Figure 4.5. This is reflected in the results of POCA reconstruction where there is a clear improvement thanks to this alignment. However, nonlinear corrections such as adjustments of the rotations of the detectors are not as easily accounted for using this method [36]. If the local coordinate on the detector is plotted against the residual value from the fit, there is a clear correlation that would indicate a misalignment in some of these nonlinear degrees of freedom such as rotations about the axes. Examples of such correlation plots for GEM4 can be found in Figure 4.7. These correlation plots for all of the detectors in the MTS can be found in Appendix A. Due to the large uncertainties in the position and orientation of the detectors, several of the detectors exhibit either a wide or asymmetric residual distribution, or in the case of the side detectors, a double-peaked structure. This double-peaked structure arises from different contributions to the residuals from tracks through different combinations of the detector modules as shown in Figure 4.8. Certain modes of misalignment are also very difficult to detect using the histogramming method including torsions of the detector stand. The method is also prone to settling into local minima of the solution space. Clearly a more robust alignment algorithm is needed to correct each of the three translational and three rotational alignment parameters per detector simultaneously. An alignment method using a Kalman filter has been investigated to solve this problem, and the progress of development of the method is described in the next section.

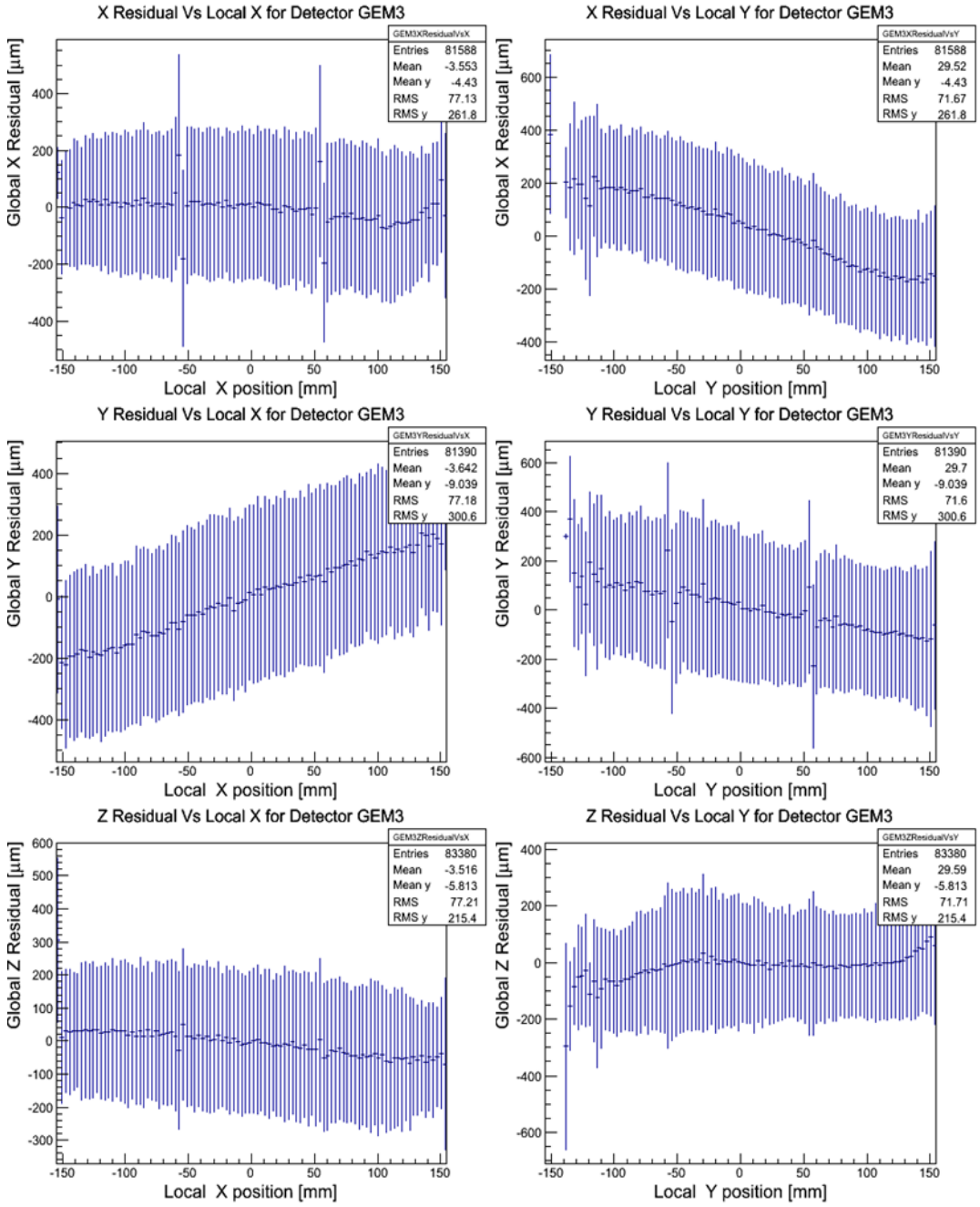


Figure 4.7 Residual correlation plots showing clear correlations between the local measured coordinate on the detector and the global residual value for GEM1. The error bars in these plots correspond to the RMS of the bin entries.

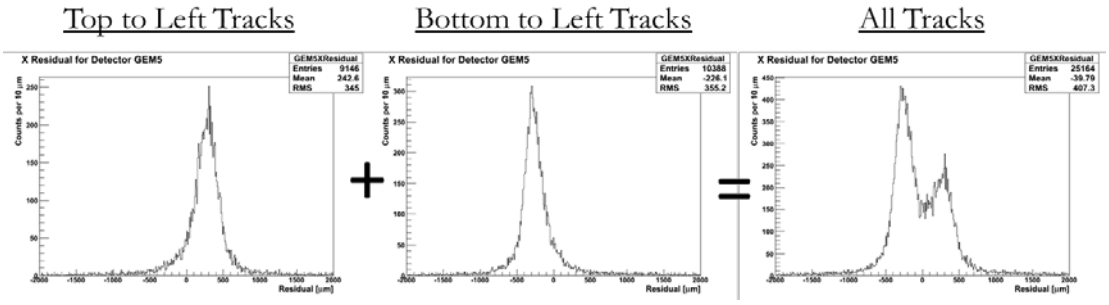


Figure 4.8 The double-peaked structure in the global residual distributions on the side detectors (here GEM5) comes from different contributions from the top to side and bottom to side tracks.

4.2.2 Alignment using a Kalman Filter

In the histogram method and both the amoreMTS and amoreSRS modules, the track fit parameters are determined using a least-squares linear fit of the data using the global Z coordinate as the abscissa. In reality, this method may not be the best estimator of these values. In an ideal scenario, where the detector alignment is known to a precision better than the intrinsic detector resolution, the Z value is not necessarily the most precise value, specifically in the case of the side detectors. These detectors measure the X and Z coordinates of the track and are nominally located in the XZ plane. Therefore the measurement of the global Z coordinate should have a higher uncertainty than that of the Y position. In this case it is inappropriate to use the Z value as the "best known" abscissa value. The best estimator of the track parameters in this case would come from a more general least-squares estimator, where the uncertainty in each value could be properly accounted for. The naive description of the track with Z as the abscissa also derives from the minimum least-squares method, but does not provide a way to propagate uncertainties in the update of the track parameters. The application of a more general least-squares method for track fitting in local detector coordinates for the MTS is not addressed in this thesis but could be developed. A recursive track fit based on a Kalman filter is presented.

The following description of the Kalman filter for track fitting follows reference [32]. A more detailed derivation of the method may be found in the original paper [39]. This filter is a recursive estimator for a discreet linear dynamic system. The filter provides a method of combining a predicted measurement and a real measurement in order to provide optimal estimation of the track parameters. In the case where the track model which governs propagation from one detector layer to the next can be well approximated by a linear expansion about the impact point on the detector, the track can be treated as a discreet linear dynamic system. The track of a particle between any two surfaces can be modeled by the equation [32]

$$\boldsymbol{p}_k = \boldsymbol{f}_k(\boldsymbol{p}_{k-1}) + \boldsymbol{P}_k\boldsymbol{\delta}_k$$

where $\boldsymbol{f}_k(\boldsymbol{p})$ is the track model that maps the track parameters $\boldsymbol{p}$ from the surface $k-1$ to k. The term $\boldsymbol{P}_k\boldsymbol{\delta}_k$ is meant to describe the errors in the track model due to stochastic noise such as multiple scattering in a known thickness of scattering material. The expectation value of $\boldsymbol{\delta}_k$ is taken to be zero and the covariance matrix of this parameter, $\boldsymbol{Q}_k$ is assumed to be known. Any general measurement (location in space, trajectory, drift time, etc.) may be represented by the measurement equation [32]

$$\boldsymbol{m}_k = \boldsymbol{h}_k(\boldsymbol{p}_k) + \boldsymbol{\varepsilon}_k$$

where $\boldsymbol{h}_k(\boldsymbol{p}_k)$ is simply an evaluation of the track parameters $\boldsymbol{p}_k$ at measurement k, and ε_k are the measurement errors. The covariance matrix of these errors $\boldsymbol{V}_k$ is assumed to be known. In our case this is taken to be a diagonal matrix with entries reflecting the intrinsic spatial resolution of the measurements. This track model may be nonlinear in many cases such as for particle trajectories in a magnetic field. The Kalman filter is based on the assumption that the track model is linear, and to satisfy this requirement, the nonlinear track model can be approximated by a Taylor

expansion about the measurement point. In the case of the MTS, both $\boldsymbol{f}_k$ and $\boldsymbol{h}_k$are assumed to be linear so the problem becomes somewhat simpler. In the case that these two functions are nonlinear, we can make the expansion [32]

$$\boldsymbol{f}_k(\boldsymbol{p}_{k-1}) = \boldsymbol{F}_k\boldsymbol{p}_{k-1} + \boldsymbol{c}_k$$

and

$$\boldsymbol{h}_k(\boldsymbol{p}_k) = \boldsymbol{H}_k\boldsymbol{p}_k + \boldsymbol{d}_k$$

where $\boldsymbol{F}_k$ and $\boldsymbol{H}_k$ are the first-order derivative Jacobian matrices of the function $\boldsymbol{f}_k$ and $\boldsymbol{h}_k$ respectively. The constant terms $\boldsymbol{c}_k$ and $\boldsymbol{d}_k$ are ignored for simplicity.

The Kalman filter provides a method of estimating the track parameters from the set of observations. Unlike a global least-squares method, which uses all measurements simultaneously to solve for the track parameters, the Kalman filter is recursive and the track parameters are updated as the track parameters are propagated through the measurements by the track equation. The Kalman filter can be thought of as prediction of the state of the track at the next measurement, followed by filtering of the predicted state at the measurement by including the measurement itself. The current estimate of the track parameters $\boldsymbol{p}_k$ based on measurements $\boldsymbol{m}_1$ to $\boldsymbol{m}_j$ is denoted by $\boldsymbol{p}_{k|j}$. There is first a prediction step of the track parameters from measurement $k-1$ to k [32]

$$\boldsymbol{p}_{k|k-1} = \boldsymbol{F}_k\boldsymbol{p}_{k-1|k-1}$$

along with a prediction of the covariance matrix $\boldsymbol{C}_k$ of the track parameters.

$$\boldsymbol{C}_{k|k-1} = \boldsymbol{F}_k\boldsymbol{C}_{k-1|k-1}\boldsymbol{F}_k^T + \boldsymbol{P}_k\boldsymbol{Q}_k\boldsymbol{P}_k^T$$

The measurement is then filtered using the measurement $\boldsymbol{m}_k$

$$\boldsymbol{p}_{k|k} = \boldsymbol{p}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{m}_k - \boldsymbol{H}_k\boldsymbol{p}_{k|k-1}\right)$$
$$\boldsymbol{K}_k = \boldsymbol{C}_{k|k-1}\boldsymbol{H}_k^T(\boldsymbol{V}_k + \boldsymbol{H}_k\boldsymbol{C}_{k|k-1}\boldsymbol{H}_k^T)^{-1}$$
$$\boldsymbol{C}_{k|k} = (\mathbf{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{C}_{k|k-1}$$

Where $\boldsymbol{K}_k$ is the so-called Kalman gain matrix, and $\mathbf{I}$ is the identity matrix. After the fitter has been run in the forward direction, it can then be run backwards in a in order to smooth the track parameters using knowledge of all of the measurements. In practice a seed track is created for the initial set of parameters with an arbitrary high covariance. This seed track then goes through the prediction and filtering steps outlined above. Thus the covariance matrix is not the true global covariance matrix of the tracks, but rather a recursive approximation to the global matrix. As a note, the true global matrix may be required for other alignment methods [40].

This method has been implemented for the MTS through the adaptation of a pre-existing package [41] using C++, ROOT, and the CLHEP libraries. It is unclear whether the code is functioning properly and requires some additional debugging. When a track is fit through only the top and bottom detectors, the results from the Kalman filter and the least-least squares method are nearly identical. However, in the case that a track includes measurements from the side detectors, the results for the track parameters typically differ by ~0.1% (see Figure 4.9). Since the Kalman filter correctly takes the measurement errors into account, it is possible that the estimates from this method are a more accurate estimator of the appropriate track parameters. However, it is also possible that this is caused by a bug in the code that remains to be addressed.

In addition to track fitting, an extended Kalman filter can also be used to provide an optimal solution for a set of alignment parameters [42]. This method is more general than the method presented in the preceding section. The formalism is derived in an analogous manner to the Kalman filter for tracking.

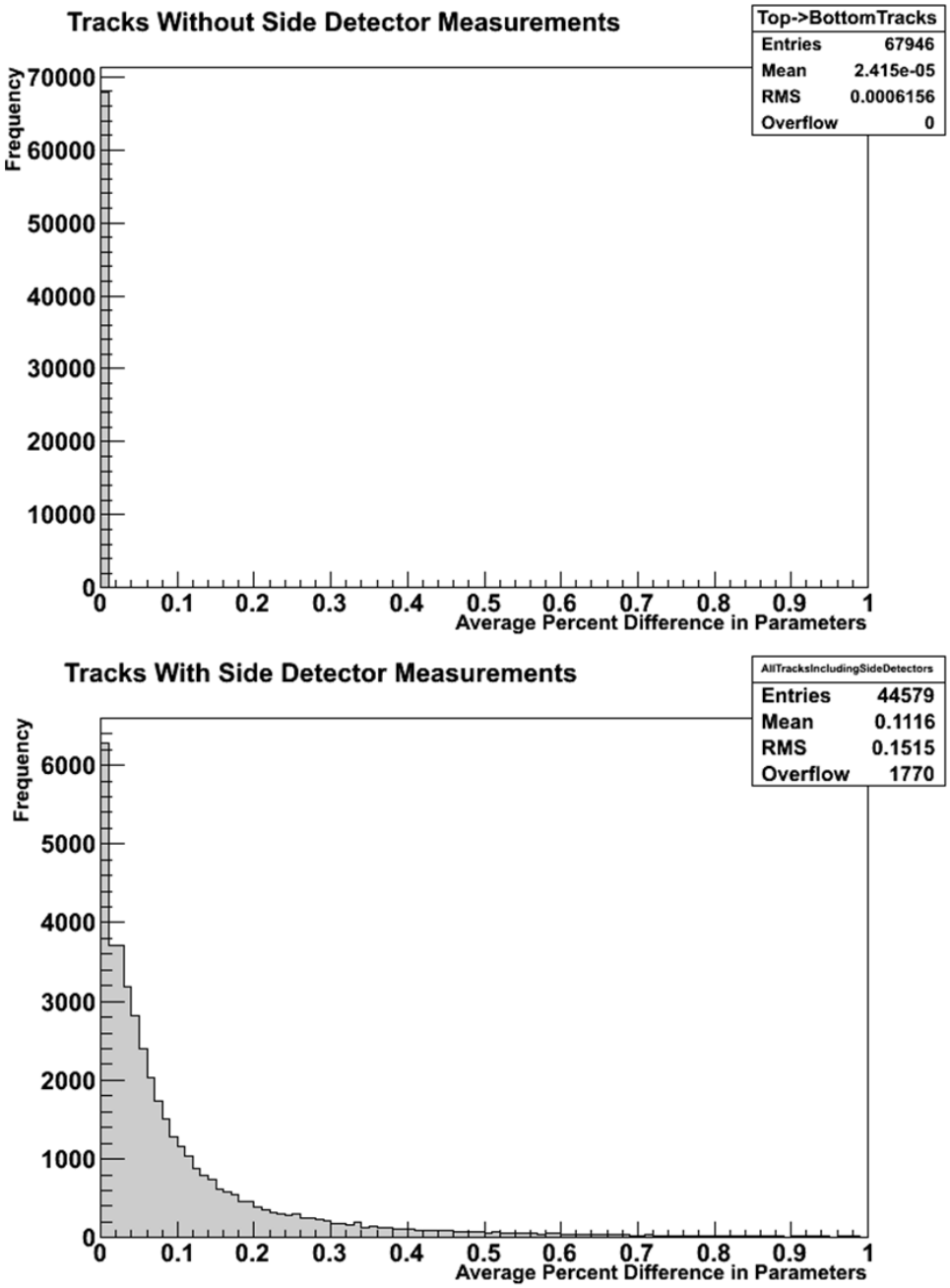


Figure 4.9 Average percent difference between track parameters obtained using a linear least-squares fit with Z as the abscissa and using a Kalman filter. Track parameters are nearly identical for tracks through only the top and bottom detectors (top), but vary by an average of 0.1 % for tracks including the side detectors (bottom).

In the case of some set of alignment parameters $\boldsymbol{a}$ the measurement equation becomes

$$\boldsymbol{m}_k = \boldsymbol{h}_k(\boldsymbol{p}_k, \boldsymbol{a}_k) + \boldsymbol{\varepsilon}_k$$

As in the case of track fitting, the function $\boldsymbol{h}_k(\boldsymbol{p}_k, \boldsymbol{a}_k)$ may be nonlinear in either parameter, thus we once again expand the function about the measurement location

$$\boldsymbol{m}_k = \boldsymbol{H}_k\boldsymbol{p}_k + \boldsymbol{D}_k\boldsymbol{a}_k + \boldsymbol{d}_k + \boldsymbol{\varepsilon}_k$$

where once again $\boldsymbol{H}_k$ and $\boldsymbol{D}_k$ are the Jacobian matrices evaluated at the expansion points. The constant term $\boldsymbol{d}_k$ is neglected. The predicted state of $\boldsymbol{p}$ and $\boldsymbol{a}$ and their associated covariance matrices $\boldsymbol{C}$ and $\boldsymbol{E}$ are calculated in an analogous manner to the track fitting Kalman filter. The filtering step using the newest measurement $\boldsymbol{m}_k$ then becomes [42]

$$\boldsymbol{a}_{k|k} = \boldsymbol{a}_{k|k-1} + \boldsymbol{E}_{k|k-1}\boldsymbol{D}_k^T\boldsymbol{W}_k(\boldsymbol{m}_k - \boldsymbol{h}_k(\boldsymbol{p}_k, \boldsymbol{a}_k))$$

$$\boldsymbol{E}_{k|k} = \boldsymbol{E}_{k|k-1} - \boldsymbol{E}_{k|k-1}\boldsymbol{D}_k^T\boldsymbol{W}_k\boldsymbol{D}_k\boldsymbol{E}_{k|k-1}$$

$$\boldsymbol{p}_{k|k} = \boldsymbol{p}_{k|k-1} + \boldsymbol{C}_{k|k-1}\boldsymbol{H}_k^T\boldsymbol{W}_k(\boldsymbol{m}_k - \boldsymbol{h}_k(\boldsymbol{p}_k, \boldsymbol{a}_k))$$

$$\boldsymbol{C}_{k|k} = \boldsymbol{C}_{k|k-1} - \boldsymbol{C}_{k|k-1}\boldsymbol{H}_k^T\boldsymbol{W}_k\boldsymbol{H}_k\boldsymbol{C}_{k|k-1}$$

where

$$\boldsymbol{W}_k = (\boldsymbol{V}_k + \boldsymbol{H}_k\boldsymbol{C}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{D}_k\boldsymbol{E}_{k|k-1}\boldsymbol{D}_k^T)^{-1}$$

While this result looks slightly different, it is equivalent to the Kalman filter used for track fitting presented earlier. This algorithm has been implemented for alignment using the same pre-existing package used for the Kalman track fit [41]. At the current time, results from this method do not converge. The exact cause of this is unknown. It is possible that there is some bug in the code that is still to be found. However, it has

been noted [42] that the convergence of the alignment parameters using this method depends on the precision of the starting value. The Kalman alignment method also assumes a Gaussian distribution of errors. In the case of the MTS the uncertainty in the nominal detector positions is not very well represented by a Gaussian as seen in Figure 4.5. There also may be misalignments in the rotations of the detectors that are too large for convergence of the Kalman alignment method. It is possible that this method may not converge until these issues are addressed. The local detector residual distributions after biased fitting using a Kalman filter are shown in Figure 4.10. There is still some work to be done to fully understand these results.

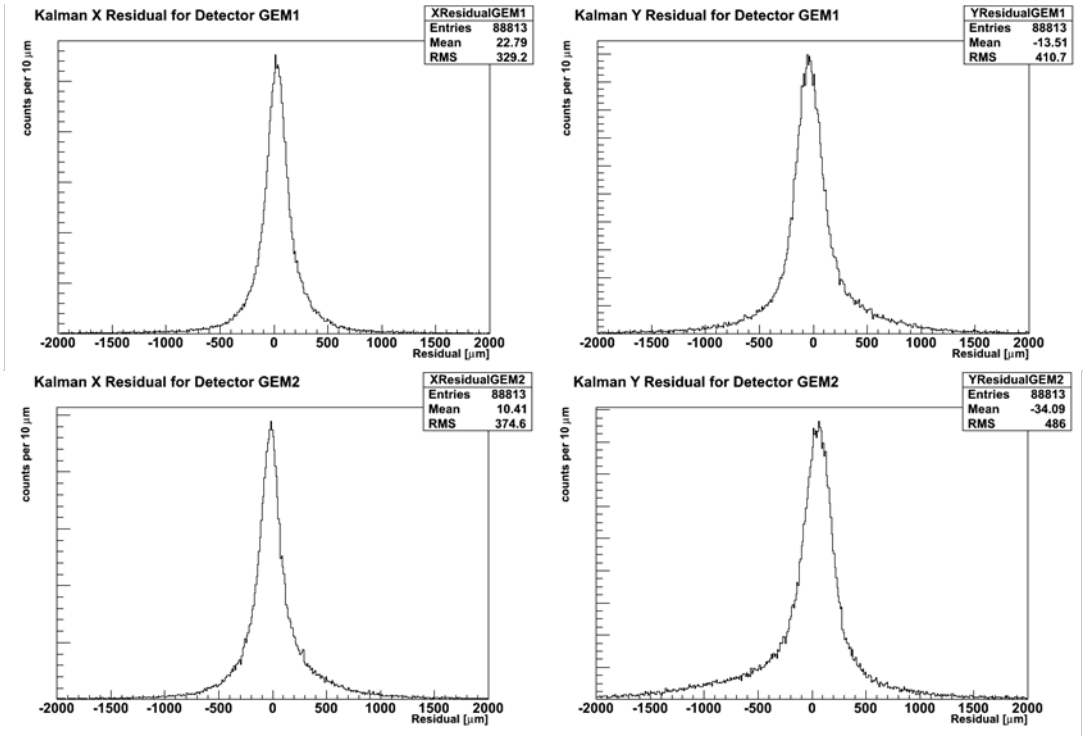


Figure 4.10 Biased local X (left) and Y (right) residuals for GEM1 and GEM2 using Kalman tracking. The alignment constants obtained using the histogram method are used.

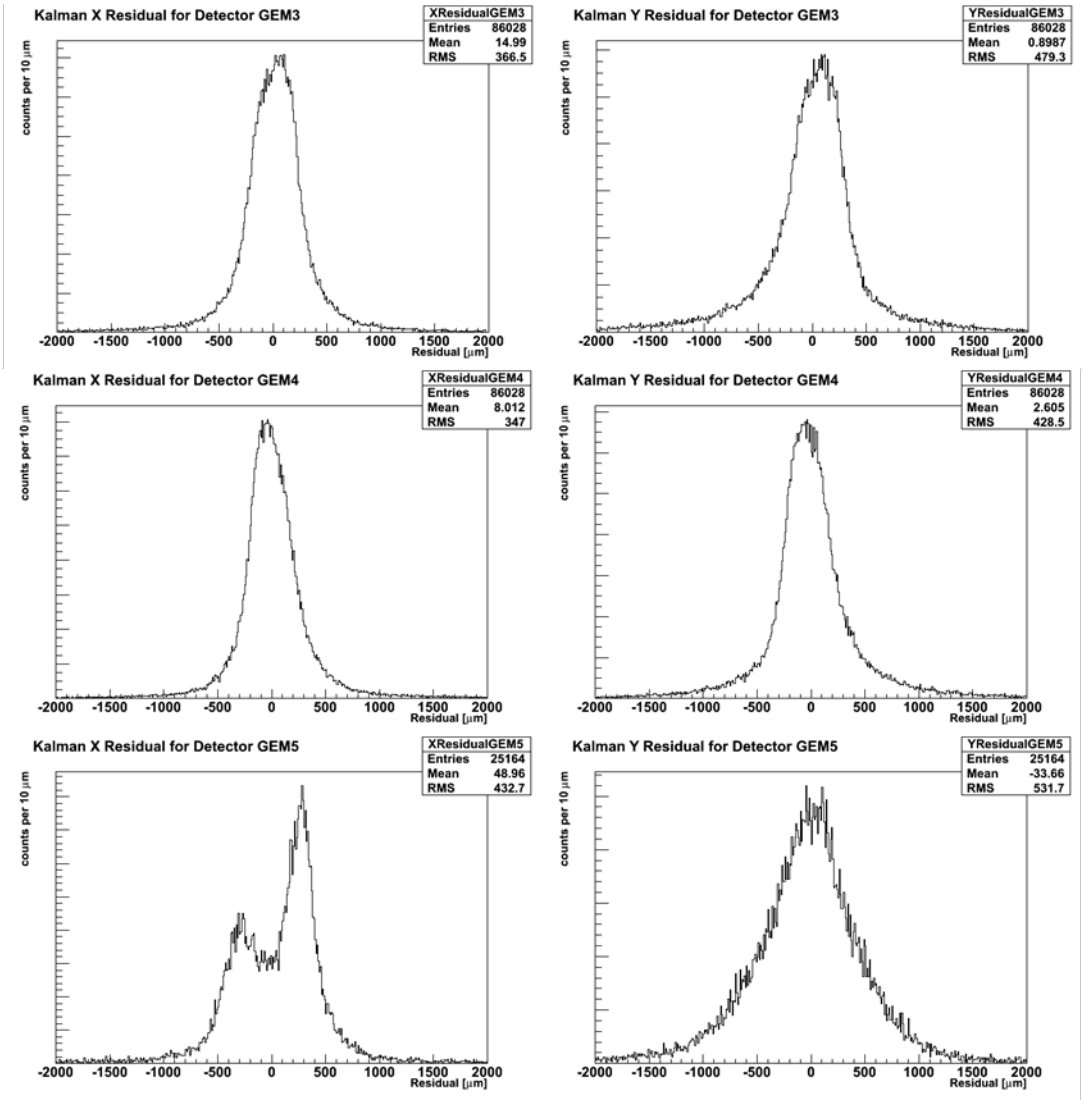


Figure 4.10 (cont'd) Biased local X (left) and Y (right) residuals for GEM3, GEM4 and GEM5 using Kalman tracking. The alignment constants obtained using the histogram method are used.

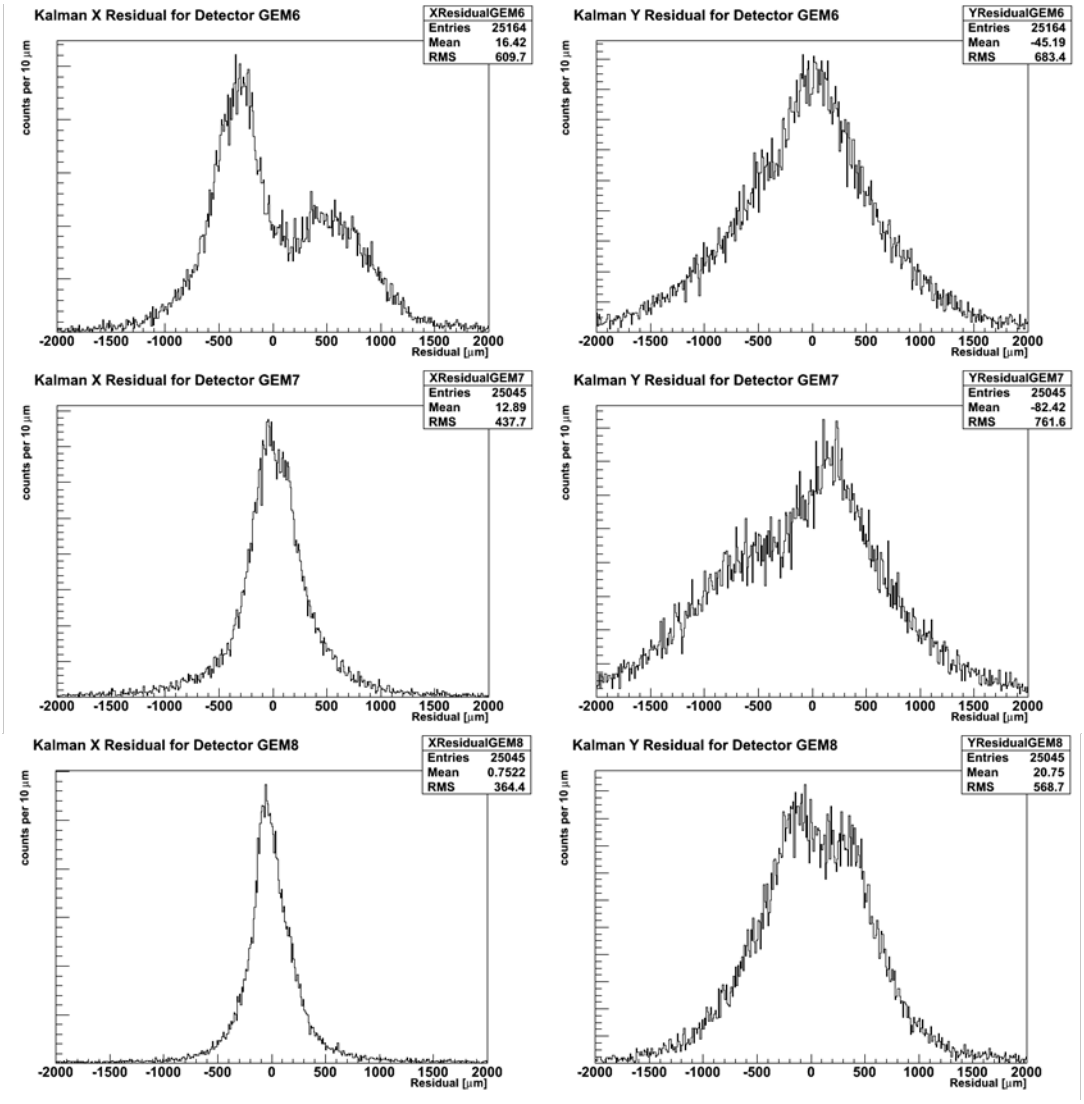


Figure 4.10 (cont'd) Biased local X (left) and Y (right) residuals for GEM6, GEM7 and GEM8 using Kalman tracking. The alignment constants obtained using the histogram method are used.

Chapter 5

Tomographic Reconstruction of Experimental Data

5.1 POCA reconstruction of target scenarios

Experimental data has been reconstructed for tomographic imaging using a simple algorithm based on determining the point of closest approach (POCA) of the incoming and outgoing tracks.This method was also used in early prototype work at LANL [43]. The scattering angle is simply found by the formula

$$\theta = \cos^{-1}\left(\frac{\vec{\boldsymbol{A}} \cdot \vec{\boldsymbol{B}}}{|\vec{\boldsymbol{A}}||\vec{\boldsymbol{B}}|}\right)$$

where $\vec{\boldsymbol{A}}$ and $\vec{\boldsymbol{B}}$ are two vectors which point along the incoming and outgoing track respectively. The point of closest approach can be found by a variety of methods. The method used for the analysis presented here is computationally light and has been outlined in a previous thesis [44] [45]. The algorithm proceeds as follows. If we consider two lines L_1 and L_2 represented by the parametric equations $\boldsymbol{L_1}: A(p) = A_0 + p(A_1 - A_0)$ and $\boldsymbol{L_2}: B(q) = B_0 + q(B_1 - B_0)$ where A_0, A_1, B_0 and B_1 are any two points on the given line and p and q are any real number. A vector formed between any two points on the line $\boldsymbol{v}(p, q) = A(p) - B(q)$ is of minimum length when p and q are chosen such that the vector is perpendicular to both lines. If we denote the "closest" vector as $\boldsymbol{w_c} = \boldsymbol{w}(p_c, q_c)$ this is equivalent to stating the two conditions $(A_1 - A_0) \cdot \boldsymbol{w_c} = 0$ and $(B_1 - B_0) \cdot \boldsymbol{w_c} = 0$. This system of equations can be solved to yield the equations $(\boldsymbol{u} \cdot \boldsymbol{u})p_c - (\boldsymbol{u} \cdot \boldsymbol{w})q_c = -\boldsymbol{u} \cdot \boldsymbol{w}_0$ and $(\boldsymbol{v} \cdot \boldsymbol{u})p_c - (\boldsymbol{v} \cdot \boldsymbol{v})q_c = -\boldsymbol{v} \cdot \boldsymbol{w}_0$ where $\boldsymbol{u} = (A_1 - A_0)$, $\boldsymbol{v} = (B_1 - B_0)$, and $\boldsymbol{w}_0 = A_0 - B_0$. These equations can be solved algebraically for p_c and q_c. With

$a = \boldsymbol{u} \cdot \boldsymbol{u}$, $b = \boldsymbol{u} \cdot \boldsymbol{v}$, $c = \boldsymbol{v} \cdot \boldsymbol{v}$, $d = \boldsymbol{u} \cdot \boldsymbol{w}_0$ and $e = \boldsymbol{v} \cdot \boldsymbol{w}_0$ the solution for p_c and q_c becomes:

$$p_c = \frac{be - cd}{ac - b^2}$$

$$q_c = \frac{ae - bd}{ac - b^2}$$

These results can then be used in the original parametric equations to retrieve the two closest points on the lines. It is then possible to create a line segment between these two closest points. The length of this line segment is called the Distance of Closest Approach (DOCA), and the midpoint of this line segment the Point of Closest Approach (POCA).

The simplicity of the POCA reconstruction method relies on the assumption that the multiple scattering that does occur in a small amount of material is well approximated by scattering at a single point. This is only well suited for reconstruction in situations where a muon traverses a small amount of material (in terms of radiation lengths). In the case of larger amounts of scattering, statistical methods must be employed since this assumption is no longer valid. There has been a great deal of work done on the subject of statistical methods for tomographic reconstruction. A Maximum Likelihood / Expectation Maximization (ML/EM) method has been developed and tested which seeks to calculate the density profiles of the probed region [44] [46]. Improvements to this method are also an active area of research. Some results of applications of a Gaussian scale mixture model and Bayesian estimation techniques for muon tomography reconstruction are presented in [47] and [48]. Application of more advanced statistical methods such as these will be critical in future large versions of the MTS capable of imaging more complex material scenarios.

In the next few sections, the experimental results of POCA reconstruction for several different target scenarios are presented. The particular scenarios were chosen to investigate different performance characteristics of the cubic-foot MTS including

coverage, material discrimination, spatial resolution, and the detection capability for shielded high-Z materials. The results are presented as 2D histograms where the color of each 2 mm x 2 mm bin corresponds to the average scattering angle for all POCA points within a 2 mm x 2 mm x 40 mm 3D voxel. All voxels with a mean scattering angle of less than 2 degrees are removed from the reconstructed images. In addition a "number of neighboring POCA" (NNP) cut is made to the data to improve the quality of the images. The NNP cut is performed by adding up the total number of POCA points in the eight surrounding voxels in the viewing plane (indicated in blue in Figure 5.1) and then removing the contents of the middle voxel if it is less than the NNP cut.

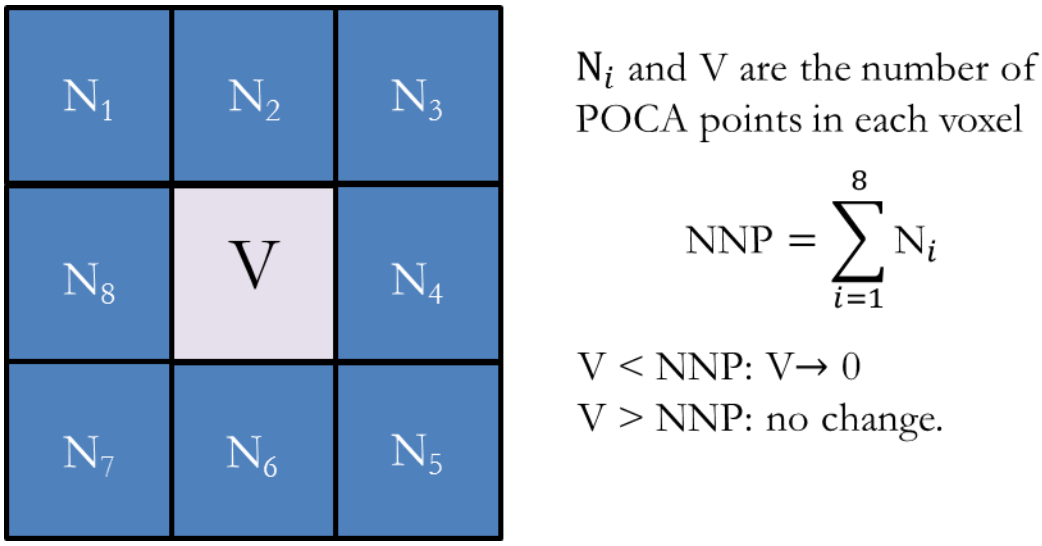


Figure 5.1 Illustration of the "number of neighboring POCA" (NNP) cut.

5.2 Five-Target scenarios

One of the primary goals of this study is to investigate the material discrimination performance of the cubic-foot MTS. It is also important to better understand the coverage throughout the imaging region. A scenario was chosen to be able to examine these two issues. For this scenario, five different blocks of material are placed in the imaging region as shown in Figure 5.2. These materials are cubes with an

identical volume of 75 cm^3. The depleted uranium cube has an additional 6 mm of aluminum shielding on all sides. The properties of all materials used in each of the experimental scenarios in this thesis may be found in Table 5.1. These materials were imaged in three different vertical positions within the detection volume of the MTS with the targets centered at z = -100 mm, -35 mm and 75 mm. Data were also taken for a scenario using stacks of the five different materials. The total number of tracks used for each reconstruction is shown in Table 5.2. Track selection was restricted in these scenarios to single-cluster events. Statistics could be improved by reanalyzing the raw data using all combinatoric tracks with selection on DOCA. The final alignment using the histogram method has been used for all reconstructions presented.

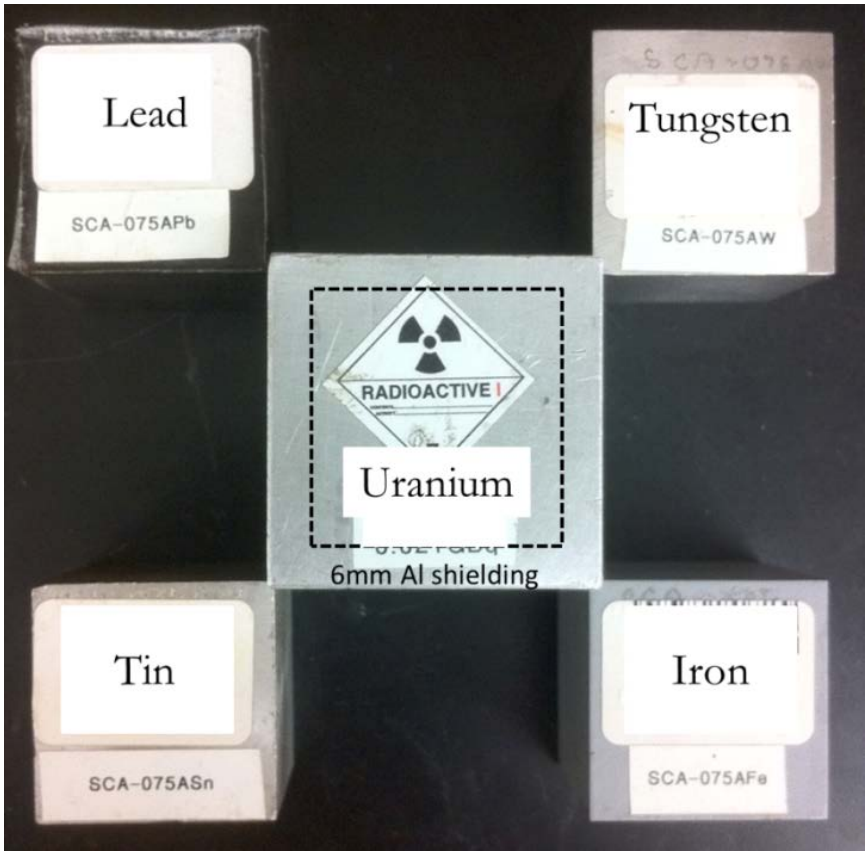


Figure 5.2 Photo of the Five-Target scenario.

Table 5.1 Properties of materials used for the experimental scenarios [49].

Material	Atomic Number Z	Density (g/cm^3)	Radiation Length (cm)	Classification
Iron	26	7.87	1.757	Low Z
Tin	50	7.31	1.206	Medium Z
Bronze (83% Cu, 7% Sn, 7% Pb, 3% Zn)	Alloy	8.83	1.285	Medium Z
Tungsten	74	19.30	0.350	High Z
Lead	82	11.35	0.561	High Z
Uranium	92	18.95	0.317	High Z

Table 5.2 Total number of tracks and POCA points used for reconstruction of the Five-Target scenarios.

Scenario	Total Reconstructed Tracks
Top Five-Target (z = 75 mm)	163,754
Middle Five-Target (z = -35 mm	155,104
Bottom Five-Target (z = -100 mm)	146,272
Stacked Five-Target	175,022

The results of POCA reconstruction in Figures 5.3-5.5 show clear tomographic images of the five targets. This result is similar for all three of the vertical positions used. An NNP cut of 5 is used for the three reconstructions. It is possible to conclude from these data that there is enough coverage to allow imaging throughout the entire vertical region of the detection volume. There do appear to be some problems with the reconstruction for the Top Five-Target scenario. There is a large ammount of high-angle noise evident in all of the reconstructions that may be caused by corrupted data being included in the reconstruction. This may be fixed by reprocessing the data with combinatoric track selection and then removing any data

set with an abnormally low number of "good tracks" as this may indicate the presence of corruption in the data.

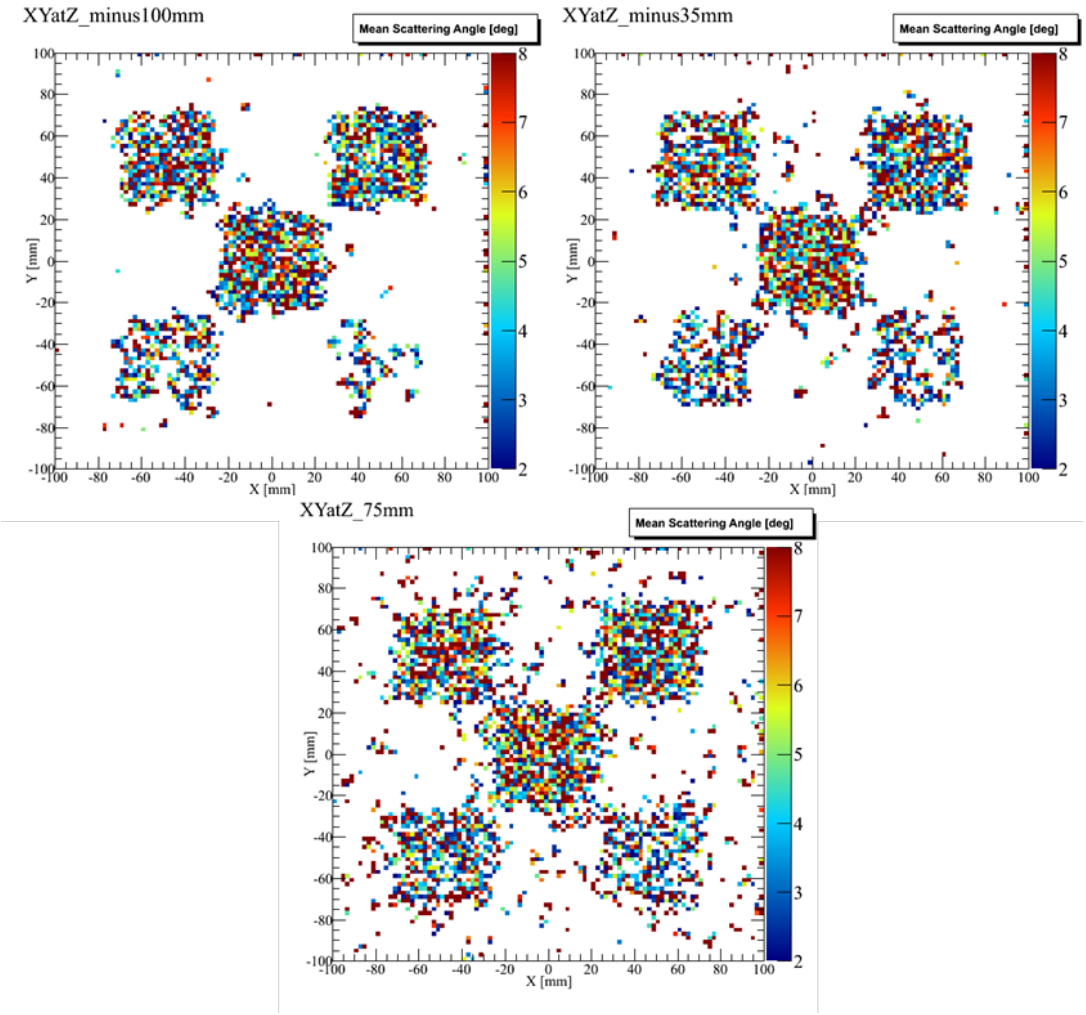


Figure 5.3 POCA reconstruction of the Five-Target scenarios using 40 mm slices in the XY plane centered at the three different vertical locations; z = -100 mm (top-left), z = -35 mm (top-right), and z = 75 mm (bottom). Single-cluster track selection is used with an NNP cut of 5.

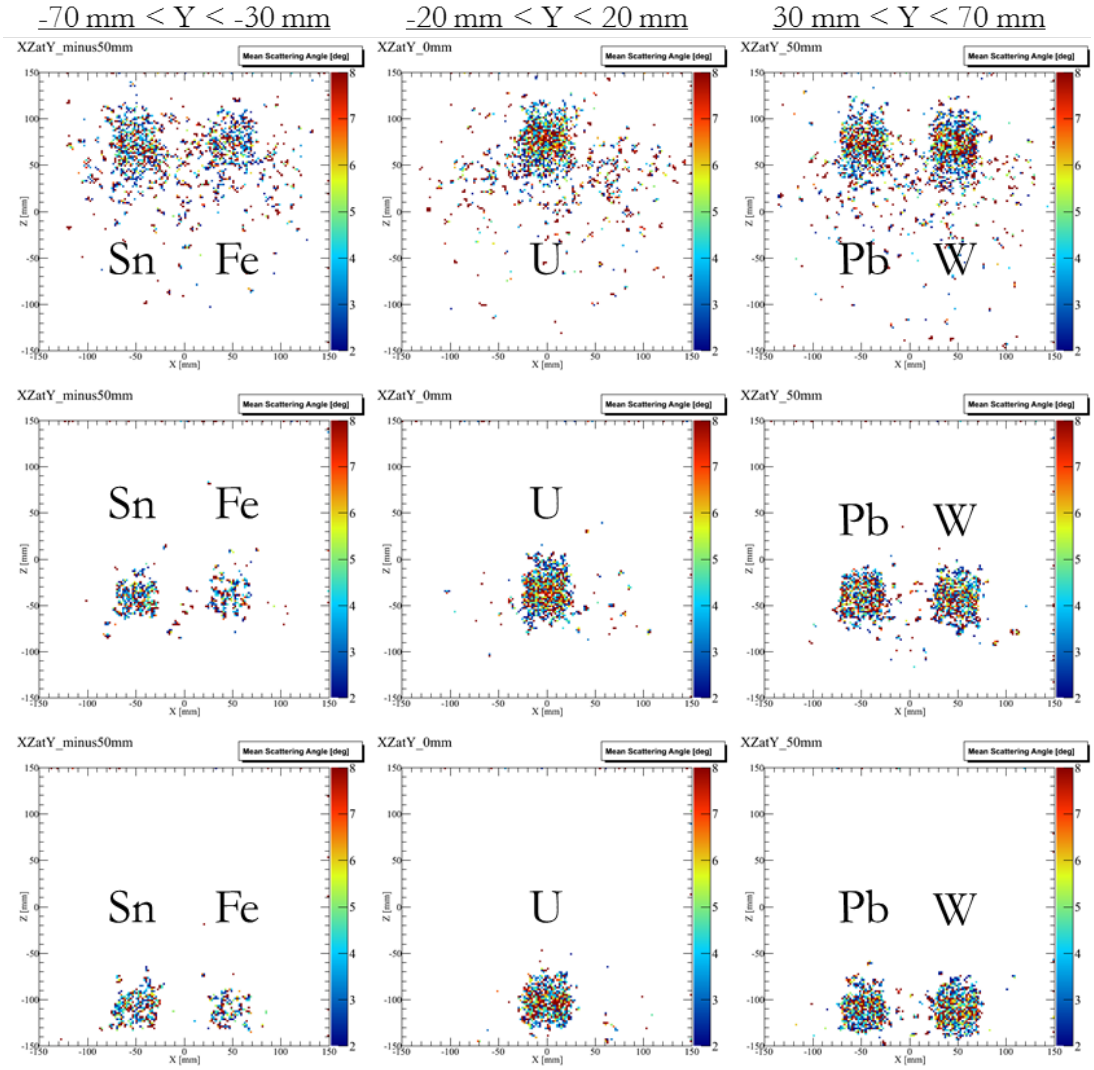


Figure 5.4 POCA reconstruction of the Five-Target scenario using 40 mm slices in the XZ plane for the three vertical locations a) top, b) middle, and c) bottom. Single-cluster track selection is used with an NNP cut of 5.

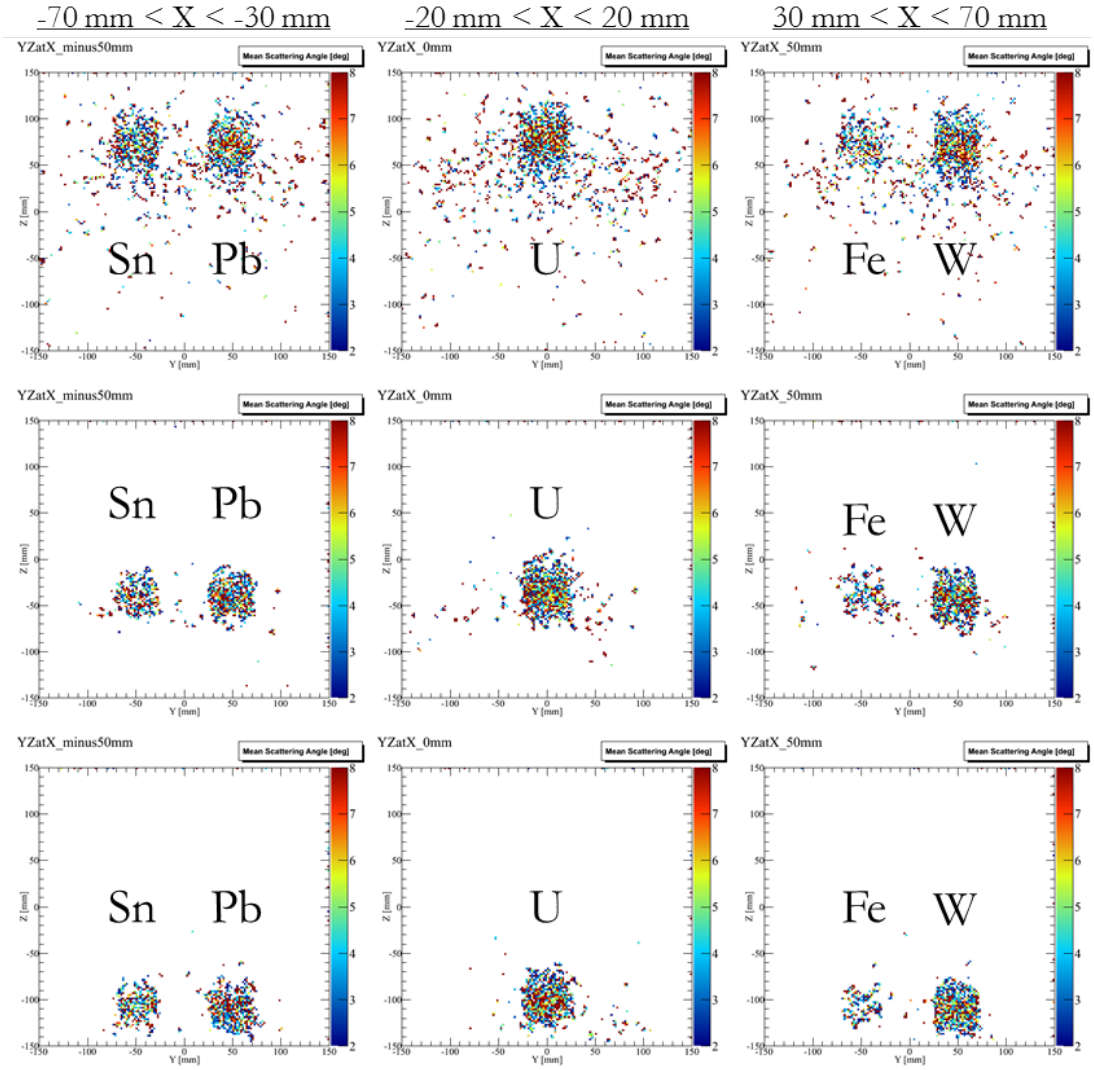


Figure 5.5 POCA reconstruction of the Five-Target scenario using 40 mm slices in the YZ plane for the three vertical locations a) top, b) middle, and c) bottom. Single-cluster track selection is used with an NNP cut of 5.

To investigate the ability of the MTS to discriminate between the different materials, we calculate a metric that we will call the "simple scattering density". This quantity is defined as the sum of all scattering angles within a given volume, divided by the size of that volume. This is then normalized to 100 k tracks. The simple scattering density within the actual physical volume occupied by each of the five targets in each of the three vertical locations can be found in Table 5.3. We estimate the error in this quantity by considering the convolution of the errors in the measurement of the polar angle for the incoming and outgoing tracks. This error in the polar angle is calculated using the measurement uncertainty taken from the gaussian fits in the residual distributions after alignment using the histogram method (Figure 4.5). This yields an average uncertainty in each reconstructed angle of 0.4 degrees. For the bottom and middle location, the relationship between the simple scattering density values show a trend that matches predictions from theory. For the bottom location, it is not possible to discriminate between the uranium and tungsten targets. Due to possible corruptions of the data, the result for the top location is inconclusive and will be excluded from the combined analysis. Overall, the MTS is clearly able to discriminate between low/medium-Z and high-Z targets of similar volume, but discrimination between two very dense high-Z objects is more difficult.

Table 5.3 Simple scattering density in a 40 x 40 x 40 mm^3 volume centered within each of the materials for the three different vertical locations of the Five-Target scenario normalized to 100 k tracks.

	Simple Scattering Density (Degrees/cm^3)				
Location	**Uranium**	**Tungsten**	**Lead**	**Tin**	**Iron**
Top	77.5 ± 4.7	78.1 ± 4.5	61.2 ± 3.4	63.9 ± 3.6	49.3 ± 3.2
Middle	110.1 ± 6.5	78.2 ± 5.1	68.6 ± 4.3	49.4 ± 3.5	43.6 ± 3.1
Bottom	91.8 ± 5.7	85.5 ± 5.6	71.8 ± 4.7	44.7 ± 3.3	55.3 ± 2.7

A stacked Five-Target scenario was also investigated. In this scenario, stacks of each material were arranged in an analogous manner to the initial Five-Target measurements. For the bottom of each stack, 150 cm^3 cubes were used, and towers were constructed using cubes of the same material of decreasing volume. In the case of the uranium stack, 150, 100 and 75 cm^3 blocks were used with a total height of 17.8 cm. This scenario is designed to test whether it is possible to discriminate materials in the case of larger amounts of material. Photos of the stacked Five-Target scenario can be found in Figure 5.6. Results of POCA reconstruction may be found in Figures 5.7-5.8. The simple scattering density has been calculated within each of the 150 cm^3 cubes. Results found in Table 5.4 indicate the highest scattering density within the depleted uranium. It is once again quite easy to discriminate between the high and low/medium-Z materials, but discrimination among different materials with similar radiation lengths such as tungsten and uranium is more difficult, but not impossible.

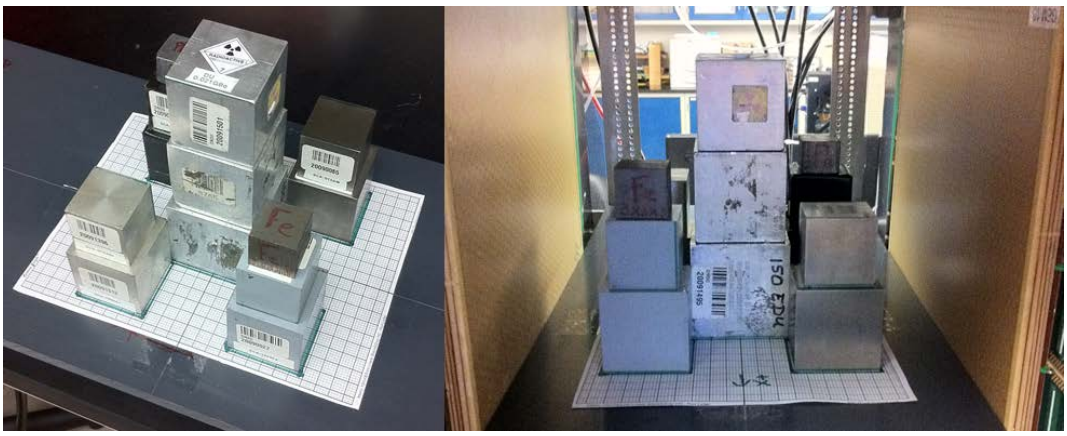


Figure 5.6 Photos of the stacked Five-Target scenario on bench (left) and inside MTS (right).

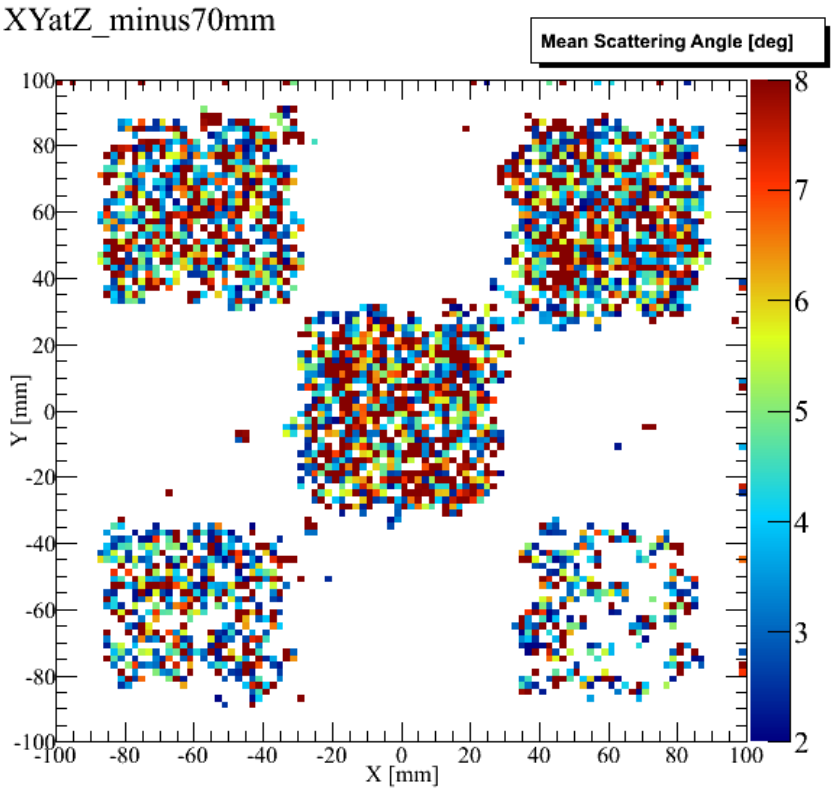


Figure 5.7 POCA reconstruction of the stacked Five-Target scenario for a 40 mm slice in the XY plane centered at Z = -70 mm, which is centered in the 150 cm^3 targets on the bottom of the stack. Single-cluster track selection is used with an NNP cut of 6.

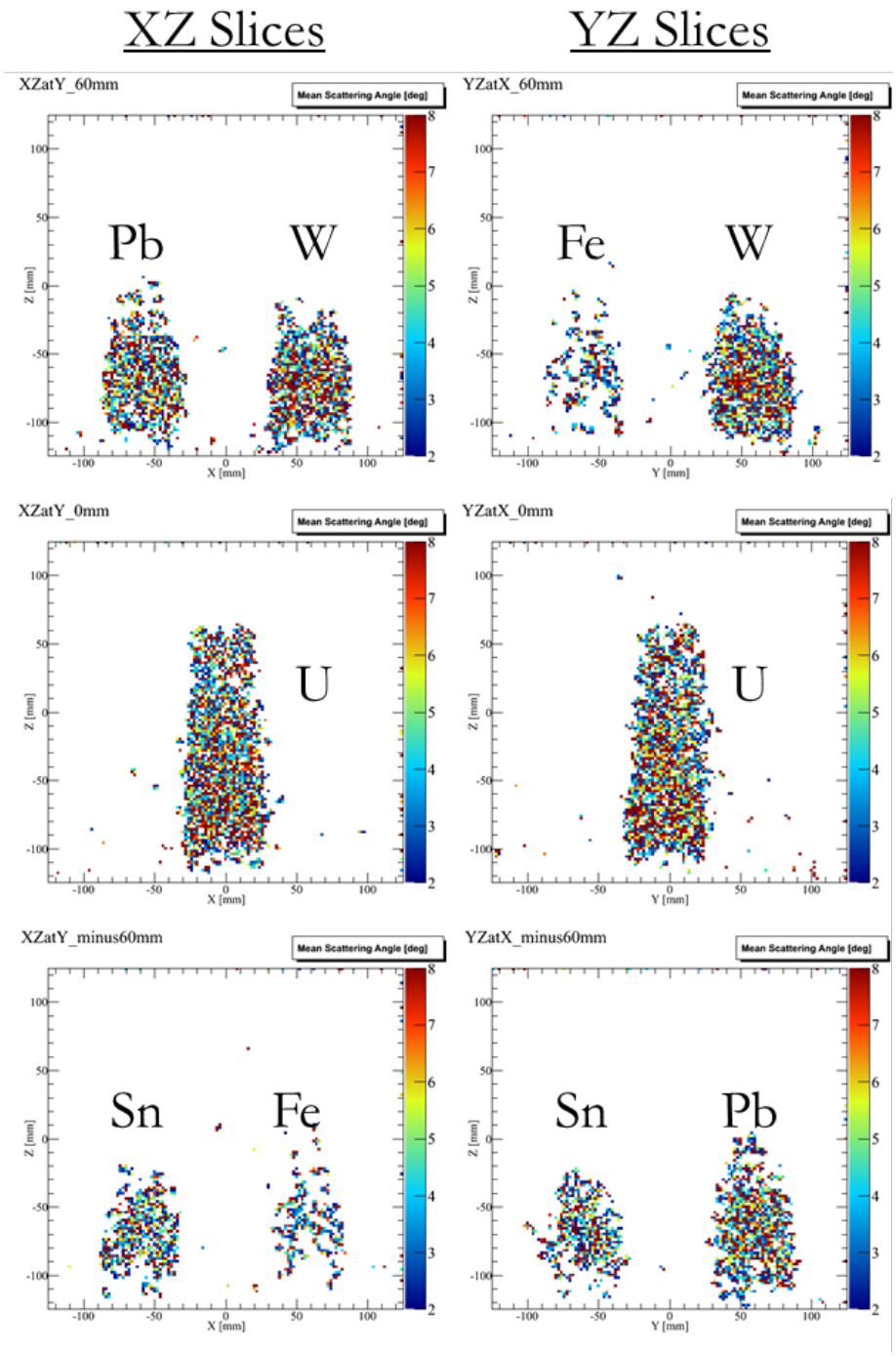


Figure 5.8 POCA reconstruction of the stacked Five-Target scenario for a 40 mm slices in the XZ and YZ planes centered at the target locations. Single-cluster track selection is used with an NNP cut of 6.

Table 5.4 Simple scattering density in a 48 x 48 x 48 mm^3 volume centered within each of the 150 cm^3 targets at the bottom of the stacked Five-Target scenario normalized to 100 k tracks.

Simple Scattering Density (Degrees/cm^3)				
Uranium	**Tungsten**	**Lead**	**Tin**	**Iron**
88.1 ± 5.0	83.2 ± 4.8	62.2 ± 4.2	42.6 ± 3.4	36.8 ± 2.8

A combined analysis using the top, middle, and stacked Five-Target scenarios is performed by averaging each of the simple scattering densities which is then plotted against the radiation length of each material. The results match quite well with the expected $X_0{}^{-1/2}$ dependence from theory, as shown in Figure 5.9.

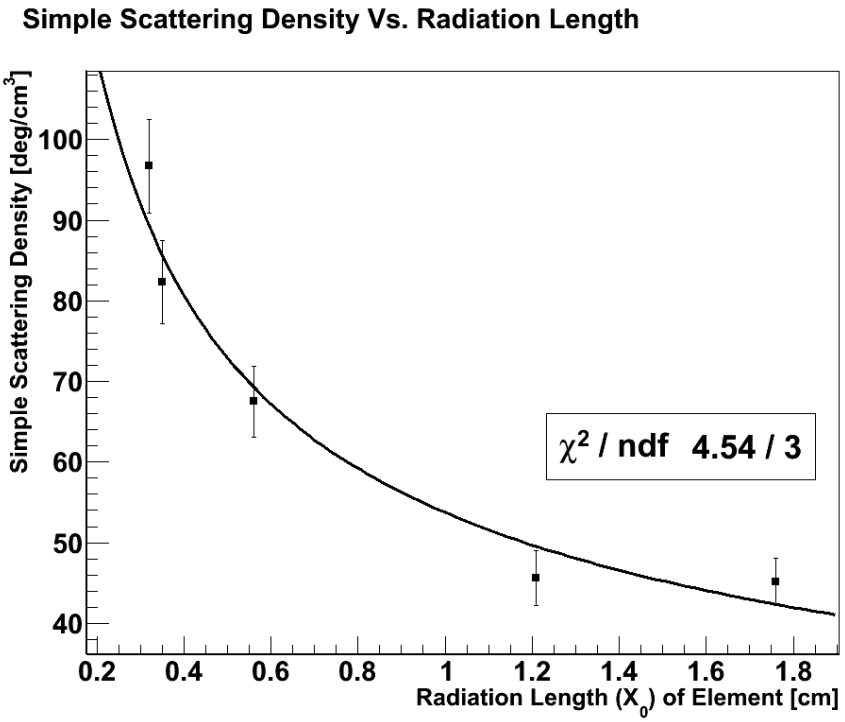


Figure 5.9 Combined results for simple scattering density versus radiation length of material for the Five-Target scenarios fitted to a $X_0{}^{-1/2}$ function.

5.3 Spatial imaging resolution of the cubic-foot MTS

In order to estimate the spatial imaging resolution of the MTS, a simple scenario using 75 cm^3 blocks of tungsten and lead was placed inside the imaging volume. The cubes were initially touching, and were then separated in steps of 2 mm. The total number of tracks used for the reconstruction of each spacing is shown in Table 5.5. Corresponding POCA reconstructions are found in Figure 5.10. The scattering density as a function of Y is plotted for each of the target spacings in Figure 5.11. The entries in these plots correspond to the simple scattering in a volume 2 mm x 40 mm x 40 mm within the targets. A statistically significant signal for the separation is evident with 6 mm of separation. Thus we can conclude that the spatial imaging resolution has a lower limit of 6 mm in the XY plane after ~100k events.

Table 5.5 Total number of reconstructed tracks used for the POCA reconstructions shown in Figure 5.10.

Target Spacing	Total Reconstructed Tracks
0 mm	115,834
2 mm	94,719
4 mm	111,036
6 mm	107,506
8 mm	121,634

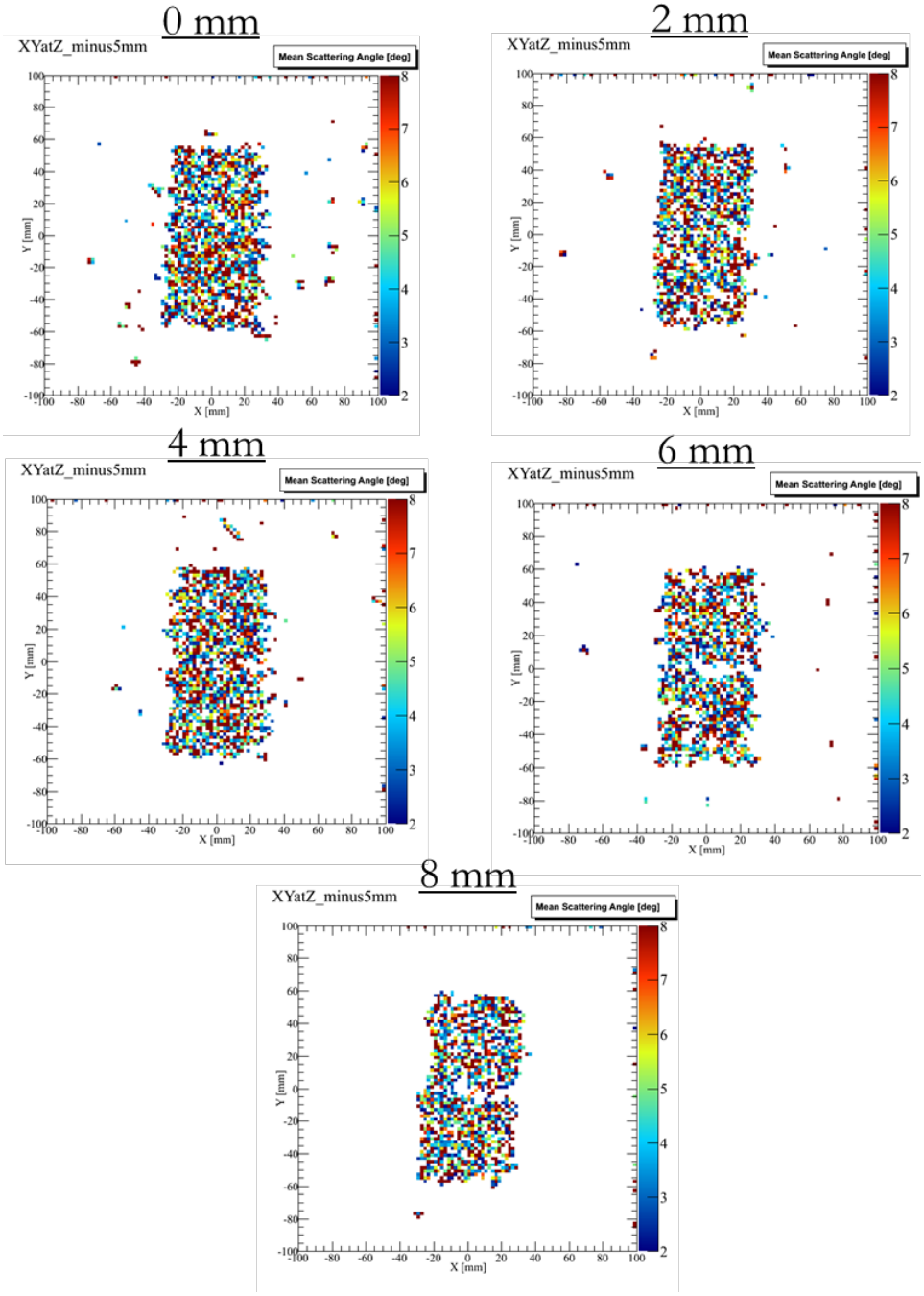


Figure 5.10 POCA reconstructions for a 40 mm slice in the XY plane centered on the tungsten and lead targets. The actual physical spacing between the two cubes is indicated above each plot. Mixed track selection is used with an NNP cut of 4.

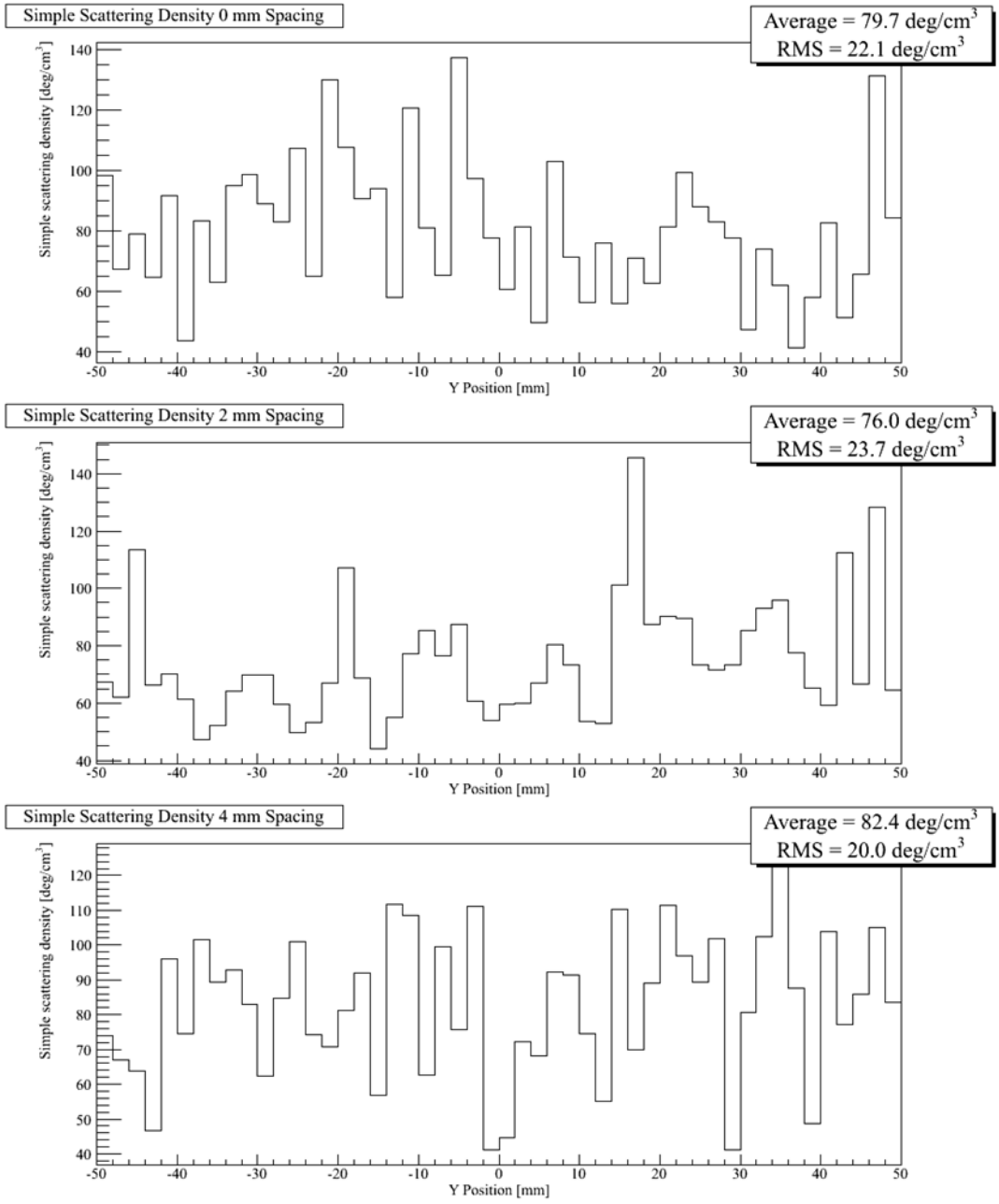


Figure 5.11 Simple scattering density as a function of Y position for the region where the tungsten and lead cubes are located. A significant signal indicating separation of the targets develops with 6 mm of spacing.

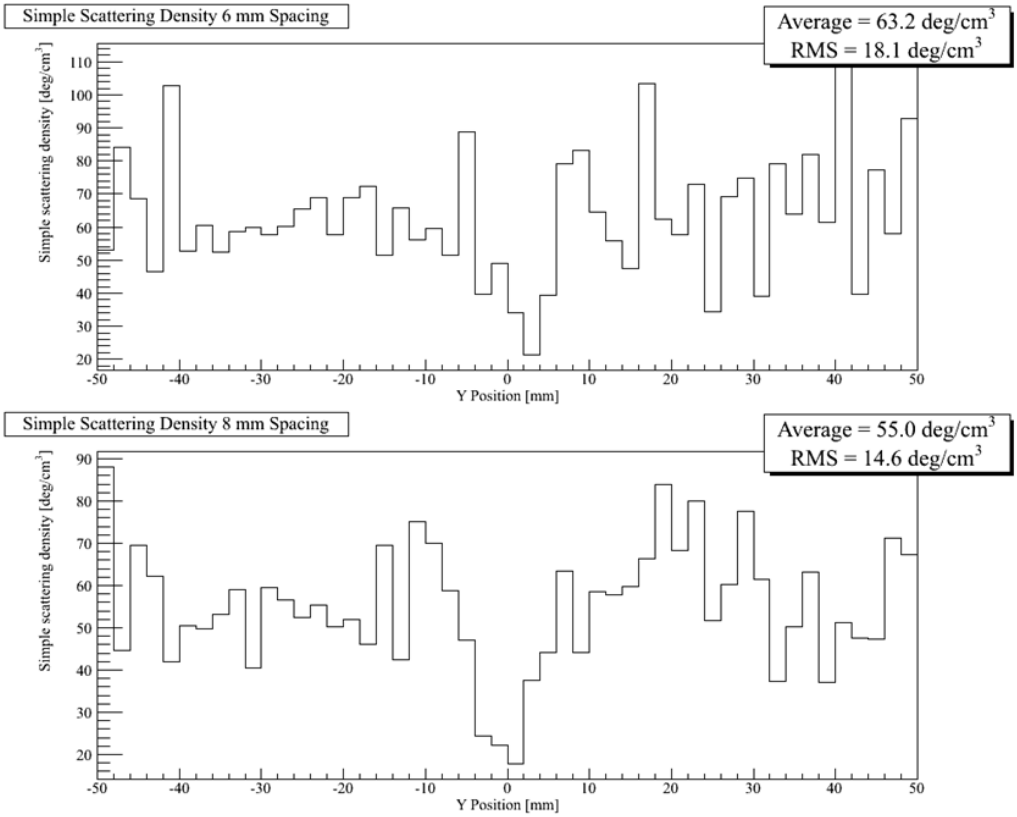


Figure 5.11 (cont'd) Simple scattering density as a function of Y position for the region where the tungsten and lead cubes are located. A significant signal indicating separation of the targets develops with 6 mm of spacing.

5.4 Depleted uranium in an iron stack

From the Five-Target scenario reconstructions, it is clear that we are able to discriminate between low-Z and high-Z materials quite well using the cubic-foot MTS. Since one of the primary goals of muon tomography is to detect well-shielded SNM, it is important to investigate such scenarios. For a first test, a "vertical clutter" stack was built with 8.3 cm of iron above and 4.2 cm of iron below the 75 cm^3 uranium block. A photo of the scenario and the results of POCA reconstruction may be found in Figure 5.12. The results show a clear ability to see the increased scattering in the high-Z uranium compared with the low-Z iron that makes up the rest of the stack.

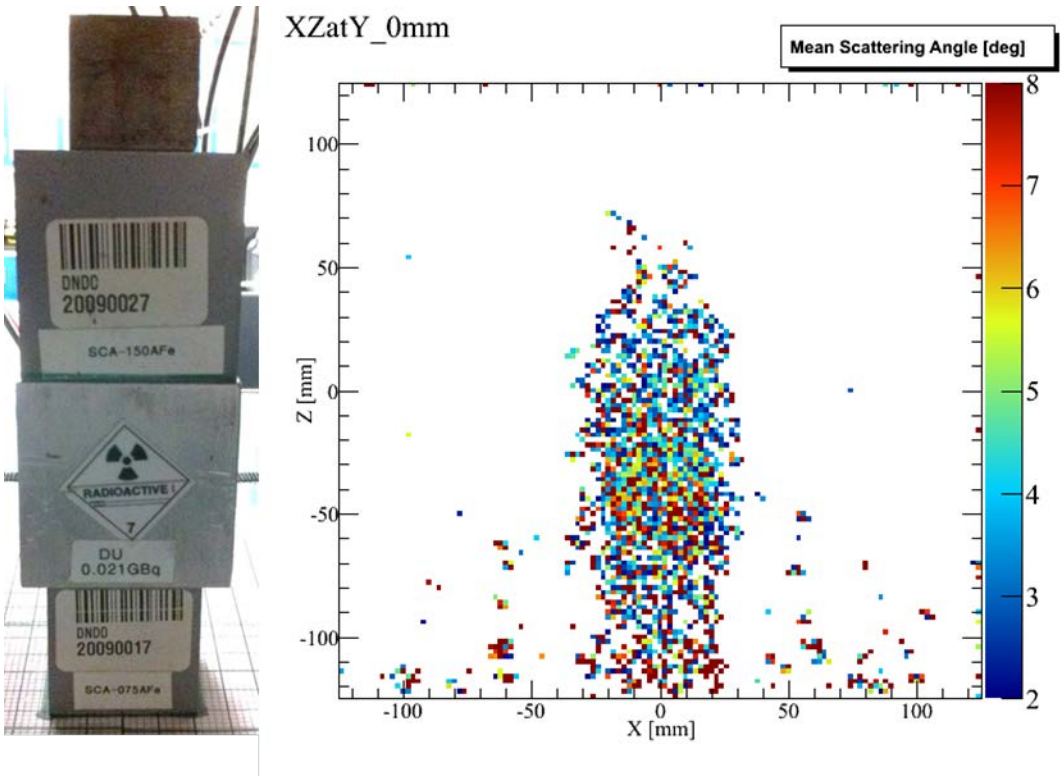


Figure 5.12 Photo of the iron and uranium stack (left) and the result of POCA reconstruction for a 40 mm XZ slice centered on the targets (right). Combinatoric track selection yielding 306,835 tracks with an NNP cut of 5 is used.

5.5 Depleted uranium shielded with bronze

In the realistic case, it is necessary to test whether the method of muon tomography is able to image threat material that has been shielded on all sides. For this study we have used an enclosed cylinder of tin-bronze (83% Cu, 7% Sn, 7% Pb, 3% Zn) [50]. The cylinder walls are 1.7 cm thick surrounding an inner cavity with a radius of 4 cm. The cylinder is capped at both ends with 1.5 cm of bronze forming the bottom cap and 3.3 cm of brass forming the top cap. The 75 cm^3 depleted uranium target is placed inside this sealed region. The prominent gamma emission from the decay chain of U^{238} is of energy 1 MeV [51]. Uranium is not a very strong neutron emitter, making it especially easy to shield from radiological detection. In this

composition, 1.7 cm of bronze will block ~60% of these 1 MeV photons [52]. The bronze has a shorter radiation length than iron (Table 5.1). This makes for an interesting test of the ability of the POCA reconstruction algorithm to discriminate between the high-Z uranium and a modest amount of shielding material. Photos of this scenario may be found in Figure 5.13. Using single-cluster track selection, 187,731 tracks were reconstructed for this scenario. The result of POCA reconstructions are found in Figure 5.14. From these plots, the uranium block can clearly be discriminated from the shielding material in both the XY and XZ projections by eye. Even the space between the uranium and the shielding material can be seen. This demonstrates that even using a simple reconstruction method we are able to image high-Z material with a moderate amount of medium-Z shielding.

Figure 5.13 Images of the bronze-shielded scenario with depleted uranium. The left image shows the placement of the uranium within the hollow cavity. The right image shows the cylinder after being sealed.

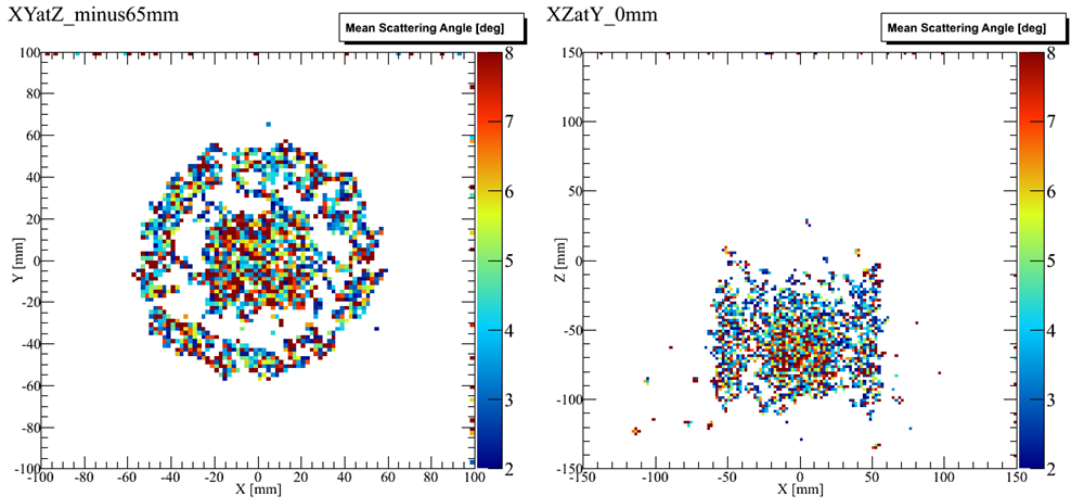


Figure 5.14 POCA reconstruction of the bronze-shielded scenario for a 40 mm slice in the XY and XZ planes centered on the depleted uranium. The strong scattering by the depleted uranium is clearly visible through the bronze shielding. Single-cluster track selection is used with an NNP cut of 10 (left) and 5 (right).

5.6 Depleted uranium with weak lead shielding

In a real scenario, it is very likely that lead may be the chosen material to shield the radiological signatures of SNM as it is cheap and effective. In order to test the ability of the MTS to detect materials shielded with lead, six different materials were imaged inside a 3.5 mm thick lead box. The five 75 cm^3 targets used earlier are used with the addition of a small tantalum cylinder (Z = 73, X_0 = 0.409 cm [49]). Photos of the scenario are found in Figure 5.15. The targets have been placed inside a Styrofoam housing to provide spacing between the targets and the lead walls of the box. This amount of lead will only block ~25% of 1 MeV gammas, and constitutes roughly half of the material thickness of the bronze shielding in terms of radiation lengths [52]. Results of POCA reconstruction with 292,255 tracks are shown in Figure 5.16. These results show that the MTS is able to image the shielded materials quite well using the POCA algorithm. All six targets are evident in the reconstruction.

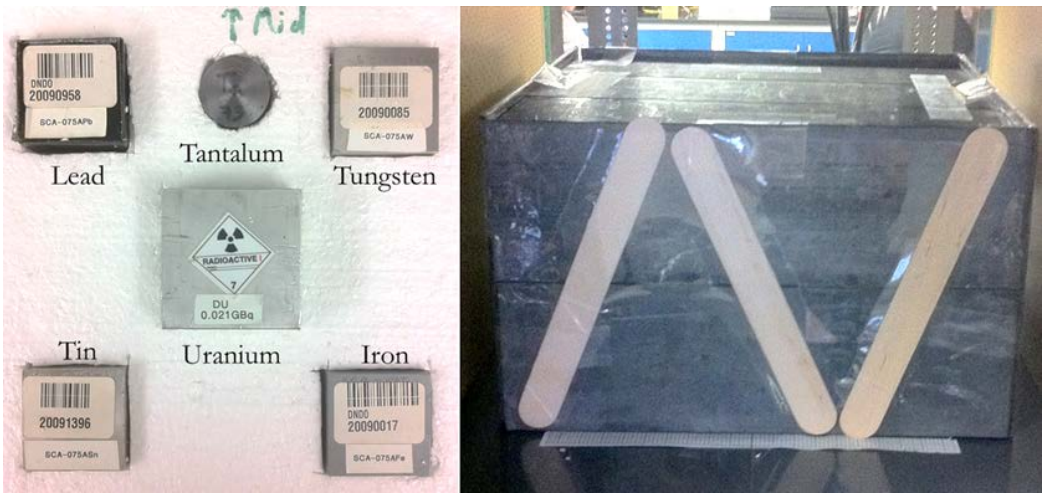


Figure 5.15 Photos of the placement of the six targets used (left) inside of a 3.5 mm thick lead box (right).

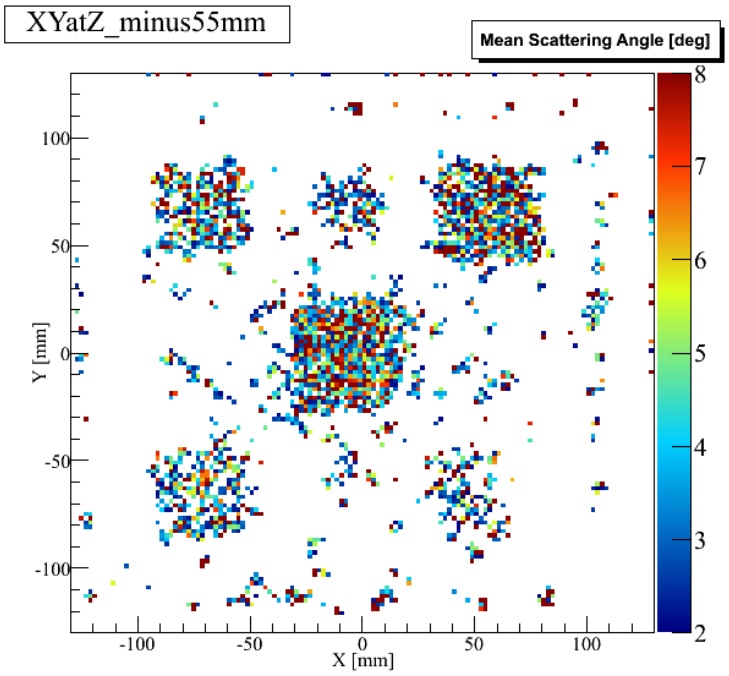


Figure 5.16 POCA reconstruction of the lead-shielded scenario for a 40 mm slice in the XY plane centered on the six targets. All of the targets are visible as well as parts of the lead shielding. Combinatoric track selection is used with an NNP cut of 6.

5.7 Depleted uranium with strong lead shielding

In order to test the limits of the MTS's ability to image well-shielded threat material, a scenario was constructed using strong lead shielding. The 75 cm^3 block of depleted uranium rests on top of a lead brick of 4.6 cm thickness. It is then covered by an additional 3.1 cm of lead, and the sides are enclosed with a lead thickness of 1.4 cm. The 75 cm^3 and 150 cm^3 lead blocks were added to fully surround the uranium block. Photos of the lead-shielded scenario can be found in Figure 5.17. At the thinnest point, this thickness of lead shields ~70% of 1 MeV photons and constitutes twice the material thickness of the bronze shielding in terms of radiation lengths. At the thickest point this value increases to ~98% shielding of 1 MeV photons and is nearly seven times the effective material thickness of the bronze shielded scenario [52]. Since this scenario clearly should be more challenging than the others, considerably more statistics were accrued. In total, 318,015 tracks were reconstructed using single cluster selection. Using combinatoric track selection, this number could increase substantially. Even with the high statistics recorded, it is very difficult to distinguish the uranium from the high-Z shielding material, as is evident in the POCA reconstructions presented in Figure 5.18. It is likely that this inability to discriminate these two high-Z materials may be attributed to the limitations of the POCA algorithm since this large amount of scattering can no longer be well approximated by the method. While the POCA reconstruction may not be able to localize the threat material, it can be concluded from the reconstructions that there is something of interest in the imaging volume that may require further screening.

Figure 5.17 Photos of the scenario with strong lead shielding. The 75 cm^3 depleted uranium block is placed in the central region and is heavily shielded on all sides.

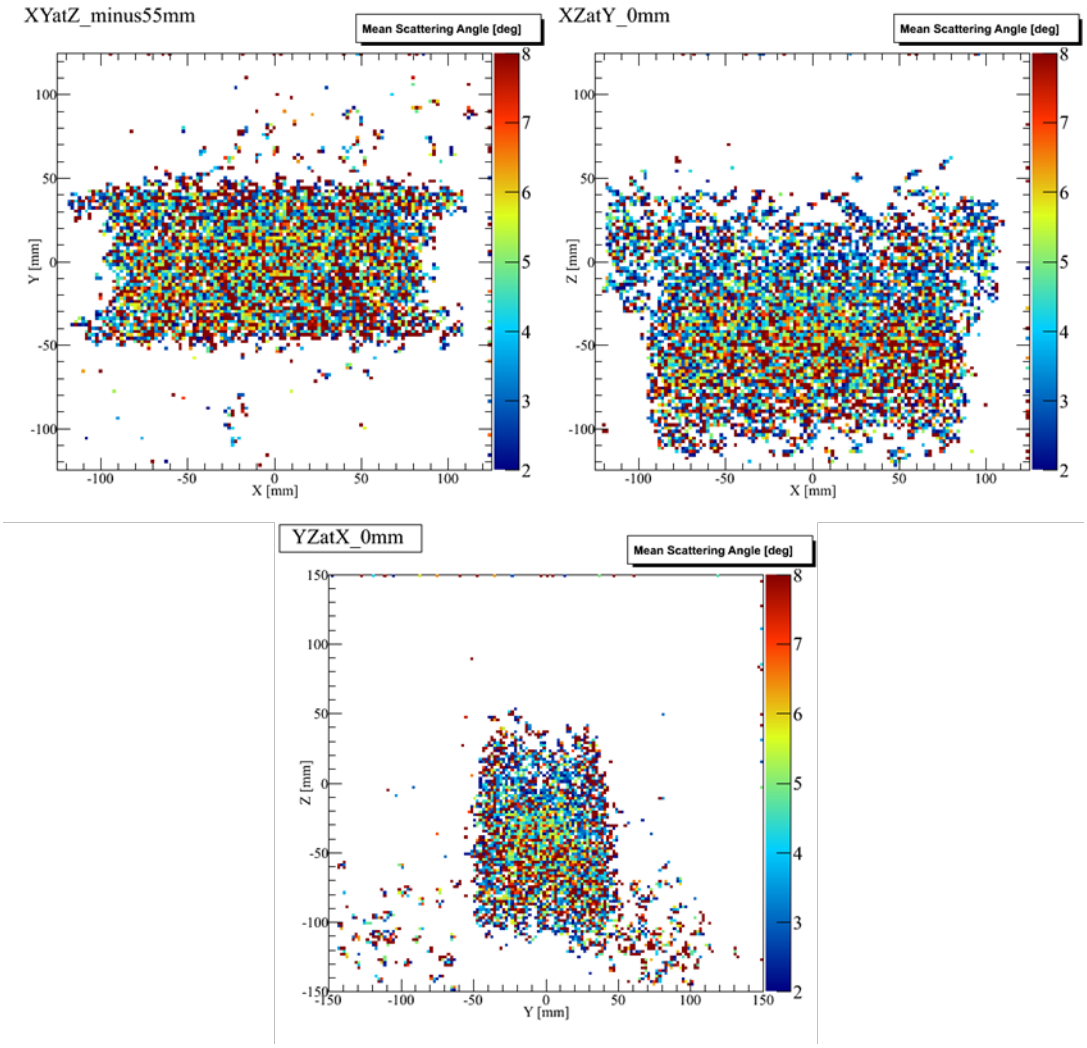


Figure 5.18 POCA reconstruction of the strongly lead-shielded scenario for a 40 mm slice in the XY (top-left), XZ (top-right), and YZ (bottom) planes centered on the depleted uranium. The scattering by the depleted uranium is not discernible from the scattering in the shielding material. Single-cluster track selection is used with an NNP cut of 6.

Chapter 6

Conclusions and Summary

6.1 Summary

A cubic-foot muon tomography station has been assembled, commissioned, and characterized using 30 cm x 30 cm triple-GEM detectors with the largest implementation of the Scalable Readout System to date. These triple-GEM detectors exhibit the highest spatial resolution among detectors that have been used for muon tomography for homeland security. It has been shown that even with a simple POCA reconstruction algorithm, we are able to discriminate between materials of similar volume but with different radiation lengths as well as image high-Z material that has been moderately shielded with medium-Z material. Imaging resolutions of ~6 mm in the XY plane are found with ~100 k muon tracks. The limitations of the POCA algorithm and the histogram method for sensor alignment have been explored. The first steps forward have been presented, and will provide ample opportunity for future improvement of the system. From the culmination of these efforts, we can conclude that muon tomography certainly provides a very useful and powerful tool for imaging different materials without an external radiation source. This is one of the few available options for detecting a well-shielded threat, and may one day be a critical component of cargo screening practices.

6.2 Future work

As mentioned throughout this thesis, there is still a great deal of work left to do in order to fully realize the potential of the muon tomography station. One of the improvements that will be made is to increase the number of detectors in the top and bottom tracking module. This will help to improve the track selection, and will also improve the quality of the fitted track. In order to improve the acceptance of the MTS

and to minimize the uncertainty in nominal detector alignment, the support structure holding the detectors should be redesigned. A sturdy frame made from aluminum cross bar could easily be designed and implemented.

Statistical methods for reconstruction will have to be developed for use in situations with a large amount of scattering. This will also be critical for determining the threat discrimination performance of the prototype. A measurement of the muon momentum will have to be included in order to take full advantage of these statistical methods. The Kalman alignment package is still a work in progress. Once hardware zero suppression has been implemented for the SRS, this alignment could be implemented in the AMORE package. Kalman tracking can also be added to AMORE but requires more redundancy in the tracking. While some tools for online and offline analysis of the data have been created, a more robust and user-friendly analysis framework for muon tomography would certainly be required for a deployable system.

The final goal for future work should be to scale-up the prototype to image a larger volume. By combining all of this development, it is hoped that muon tomography plays an active role in the future efforts to ensure the domestic safety of our country.

References

[1] "CBP Snapshot," [Online]. Available: http://cbp.gov/xp/cgov/about/accomplish/. [Accessed 12 March 2011].

[2] "Public Law 110-53: Implementing Recommendation of the 9/11 Commission Act of 2007," 3 August 2007. [Online]. Available: http://intelligence.senate.gov/laws/pl11053.pdf. [Accessed 10 March 2012].

[3] J. Medalia, "Detection of Nuclear Weapons and Materials: Science, Technologies, Observations," Congressional Research Service, 2010.

[4] K. Nakamura, et al., "Summary Tables of Particle Properties," Particle Data Group, *Journal of Physics G: Nuclear and Particle Physics,* vol. 37, no. 7A, pp. 25-88, 2010.

[5] T. K. Gaisser and T. Stanev, "Cosmic Rays," Particle Data Group, *Journal of Physics G: Nuclear and Particle Physics,* vol. 37, no. 7A, pp. 269-276, 2010.

[6] E. P. George, *Commonwealth Engineer,* p. 455, July 1955.

[7] L. W. Alvarez, et al., "Search for Hidden Chambers in the Pyramids," *Science,* vol. 167, no. 3919, pp. 832-839, 1970.

[8] K. N. Borozdin, et al., "Radiographic imaging with cosmic-ray muons," *Nature,* vol. 422, p. 277, 2003.

[9] W. C. Priedhorsky, et al., "Detection of high-Z objects using multiple scattering of cosmic ray muons," *Review of Scientific Instruments,* vol. 74, no. 10, 2003.

[10] H. Bichsel, D. E. Groom and S. R. Klein, "Passage of Particles Through Matter," *Journal of Physics G: Nuclear and Particle Physics,* vol. 37, no. 7A, pp. 285-299, 2010.

[11] C.L. Morris, et al., "Tomographic Imaging with Cosmic Ray Muons," *Science and Global Security,* no. 16, pp. 37-53, 2008.

[12] B. Ketzer, Q. Weitzel, S. Paul, F. Sauli and L. Ropelewski, "Performance of triple GEM tracking detectors in the COMPASS experiment," *Nuclear Instruments and Methods in Physics Research A,* vol. 535, pp. 314-318, 2004.

[13] F. Sauli, "GEM: A new concept for electron amplification in gas detectors," *Nuclear Instruments and Methods in Physics Research A,* vol. 386, pp. 531-534, 1997.

[14] F. Sauli and A. Sharma, "Micropattern Gaseous Detectors," *Annu. Rev. Nuck. Part. Sci.,* vol. 49, pp. 341-388, 1999.

[15] F. Murtas, "Development of a gaseous detector based on GEM Technology," 28 November 2002. [Online]. Available: http://www.lnf.infn.it/seminars/talks/murtas_28_11_02.pdf. [Accessed 10 April 2012].

[16] M. Altunbas, et al., "Aging measurements with the Gas Electron Multiplier (GEM)," *Nuclear Instruments and Methods in Physics Research A,* no. 515, pp. 249-254, 2003.

[17] The COMPASS Collaboration, "COMPASS: a proposal for a COmmon Muon and Proton Apparatus for Structure and Spectroscopy," *CERN/SPSLC,* Vols. 96-14, 1996.

[18] A. Q. Segovia, "Construction and Performance of Gas Electron Multiplier Detectors for Nuclear Contraband Detection using Muon Tomography," Master's Thesis, Florida Institute of Technology, 2010.

[19] B. Benson, "Thermal Stretching of GEM Foils and Characterization of Triple-GEM Detectors for Uses in Astronomy," Master's Thesis, Florida Institute of Technology, 2011.

[20] M. Staib, et al., "Thermal Stretching of Large-Area GEM Foils Using an Infrared Heating Method," RD51-Note-2011-004, 2011.

[21] S. Martoiu, H. Muller and J. Toledo, "Front-end electronics for the Scalable Readout System of RD51," in *IEEE Nuclear Science Symposium Conference Record*, Valencia, 2011.

[22] "RD51 Proposal," [Online]. Available: http://rd51-public.web.cern.ch/rd51-public/Documents/RD51Proposal_21082008.pdf. [Accessed 4 April 2012].

[23] The ALICE Collaboration, "The ALICE experiment at the CERN LHC," *Journal of Instrumentation,* vol. 3, no. S08002, 2008.

[24] "ALICE DAQ and ECS Manual," December 2010. [Online]. Available: https://edms.cern.ch/document/1056364/. [Accessed 20 March 2012].

[25] M. J. French, et al., "Design and results from the APV25, a deep sub-micron CMOS front-end chip for the CMS tracker," *Nuclear Instruments and Methods in Physics Research A,* no. 466, pp. 359-365, 2001.

[26] J. Toledo, et al., "The Front-End Concentrator card for the RD51 Scalable Readout System," in *Topical Workshop on Electronics for Particle Physics*, Vienna, 2011.

[27] S. Martoiu, "Digitizer Card in SRS 'C-format'," March 2010. [Online]. Available: https://espace.cern.ch/rd51-wg5/srs/Shared%20Documents/Sorin-Digitizer_Card_in_SRS.pdf. [Accessed 4 April 2012].

[28] J. Rylander, et al., "Quarknet Cosmic Ray Muon Detector User's Manual Series "6000" DAQ," January 2010. [Online]. Available: http://www18.i2u2.org/cosmic/library/upload/b/ba/6000CRMDUserManual.pdf. [Accessed 16 April 2012].

[29] S. Martoiu, "SRS Slow Control Manual," 21 March 2012. [Online]. Available: https://espace.cern.ch/rd51-wg5/srs/Documentation/SRS_Slow_Control_Manual.pdf. [Accessed 4 April 2012].

[30] R. Brun and F. Rademakers, "ROOT - An object oriented data analysis framework," *Nuclear Instruments and Methods in Physics Research A,* vol. 389, pp. 81-86, 1997.

[31] K. Gnanvo, "Tutorial For Running AMORE Online & Offline in HEP Lab A, FIT," 2 October 2011. [Online]. Available: http://research.fit.edu/hep_labA/documents/amoreTutorialGuide.pdf. [Accessed 16 April 2012].

[32] R. Frühwirth, M. Regler, R. K. Bock, H. Grote and D. Notz, "Data Analysis Techniques for High-Energy Physics," Cambridge: Cambridge University Press, 2000.

[33] L. Grasso, "Designs for Muon Tomography Station Prototypes Using GEM Detectors," Master's Thesis, Florida Institute of Technology, 2012.

[34] M. Hohlmann, et al., "Design and Performance of the Alignment System for the CMS Muon Endcaps," in *IEEE Nuclear Science Symposium Conference Record*, San Diego, 2006.

[35] T. Lampen, "General alignment concept of the CMS," *Nuclear Instruments and Methods in Physics Research A,* vol. 566, pp. 100-103, 2006.

[36] V. Blobel, "Software alignment for tracking detectors," *Nuclear Instruments and Methods in Physics Research A,* vol. 566, pp. 5-13, 2006.

[37] V. Karimäki, et al., "Sensor Alignment by Tracks," in *International Conference on Computing in High Energy and Nuclear Physics*, La Jolla, 2003.

[38] G. Zech and T. Zeuner, "A Simple Iterative Alignment Method Using Gradient Descending Minimum Search," in *PHYSTAT2003*, Stanford, 2003.

[39] R. Frühwirth, "Application of Kalman Filtering to Track and Vertex Fitting," *Nuclear Instruments and Methods in Physics Research A,* vol. 262, pp. 444-450, 1987.

[40] W. D. Hulsbergen, "The global covariance matrix of tracks fitted with a Kalman filter and an application in detector alignment," *Nuclear Instruments and Methods in Physics Research A,* vol. 600, pp. 471-477, 2009.

[41] M. Weber and D. Sprenger, "Kalman Alignment," 2010. [Online]. Available: http://kalmanalignment.hepforge.org/. [Accessed 5 April 2012].

[42] R. Frühwirth, T. Todorov and M. Winkler, "Estimation of detector alignment parameters using the Kalman filter with annealing," *Journal of Physics G: Nuclear and Particle Physics,* vol. 29, pp. 561-574, 2003.

[43] L. J. Schultz, et al., "Image reconstruction and material Z discrimination via cosmic ray muon radiography," *Nuclear Instruments and Methods in Physics Research A,* no. 519, pp. 687-694, 2004.

[44] R. Hoch, "Advances in Cosmic Ray Muon Tomography Reconstruction Algorithms," Master's Thesis, Florida Institute of Technology, 2009.

[45] R. Hoch, et al., "Muon Tomography Algorithms for Nuclear Threat Detection," in *Proc. of the 22nd International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems - IEA/AIE*, 2009.

[46] L. J. Schultz, et al., "Statistical Reconstruction for Cosmic Ray Muon Tomography," *IEEE Transactions on Image Processing,* vol. 16, no. 8, pp. 1985-1993, 2007.

[47] G. Wang, L. J. Schultz and J. Qi, "Statistical Image Reconstruction for Muon Tomography Using a Gaussian Scale Mixture Model," *IEEE Transactions on Nuclear Science,* vol. 56, no. 4, pp. 2480-2486, 2009.

[48] G. Wang, L. J. Schultz and J. Qi, "Bayesian Image Reconstruction for Improving Detection Performance of Muon Tomography," *IEEE Transactions on Image Processing,* vol. 18, no. 5, pp. 1080-1089, 2009.

[49] "Atomic and Nuclear Properties of Materials," Particle Data Group, February 2011. [Online]. Available: http://pdg.lbl.gov/2011/AtomicNuclearProperties/. [Accessed 2 April 2012].

[50] J. R. Dwyer, et al., "Measurements of x-ray emission from rocket-triggered lightning," *Geophysical Research Letters,* vol. 31, 2004.

[51] T. E. Sampson, "Precision Measurement of Gamma Ray Energies from 238U Daughters," *Nuclear Instruments and Methods,* vol. 98, pp. 37-40, 1972.

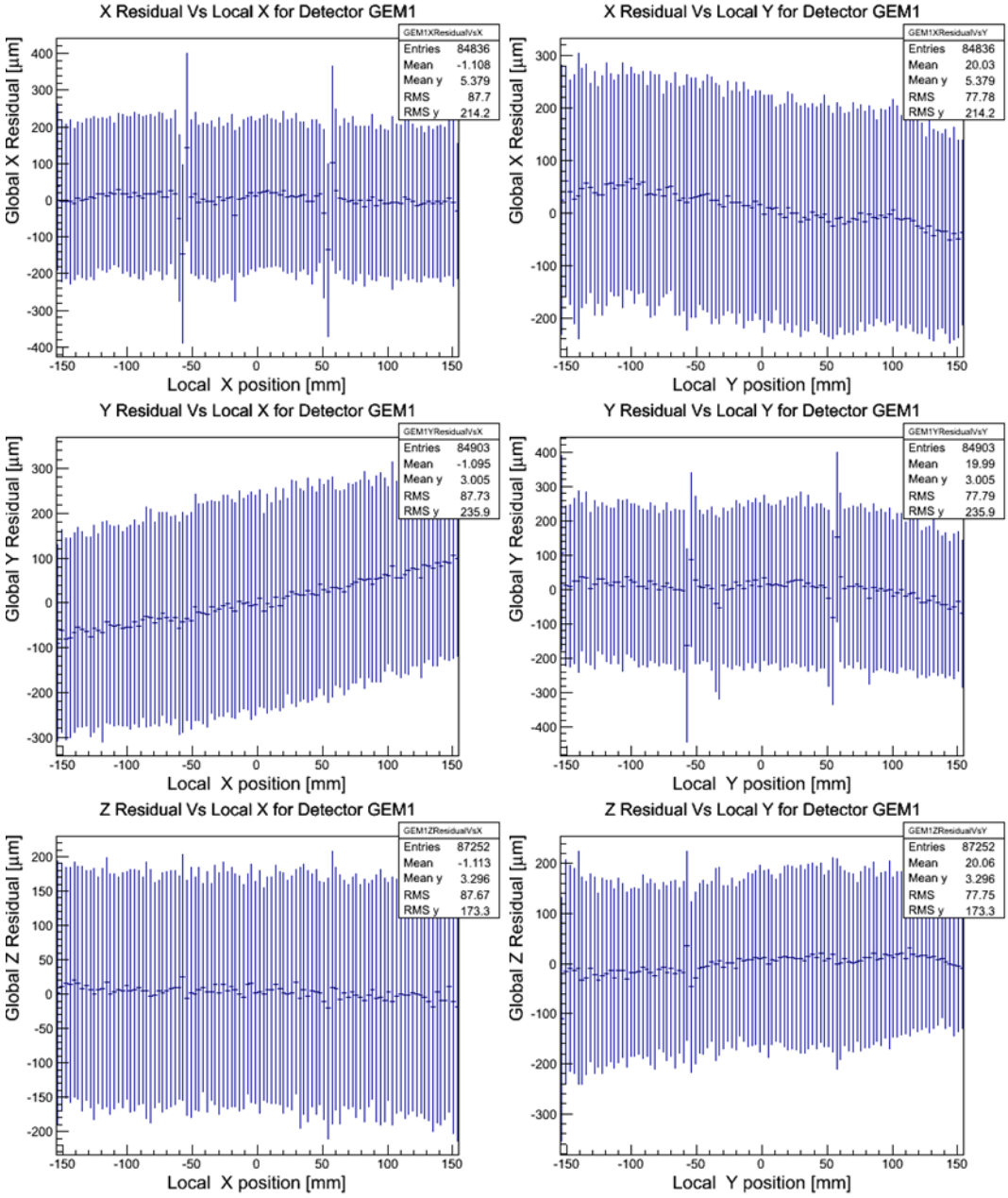
[52] S. M. Seltzer and J. H. Hubbel, "Tables of X-Ray Mass Attenuation Coefficients and Mass Energy-Absorption Coefficients," 2004. [Online]. Available: http://physics.nist.gov/xaamdi. [Accessed 2 April 2012].

Appendix A

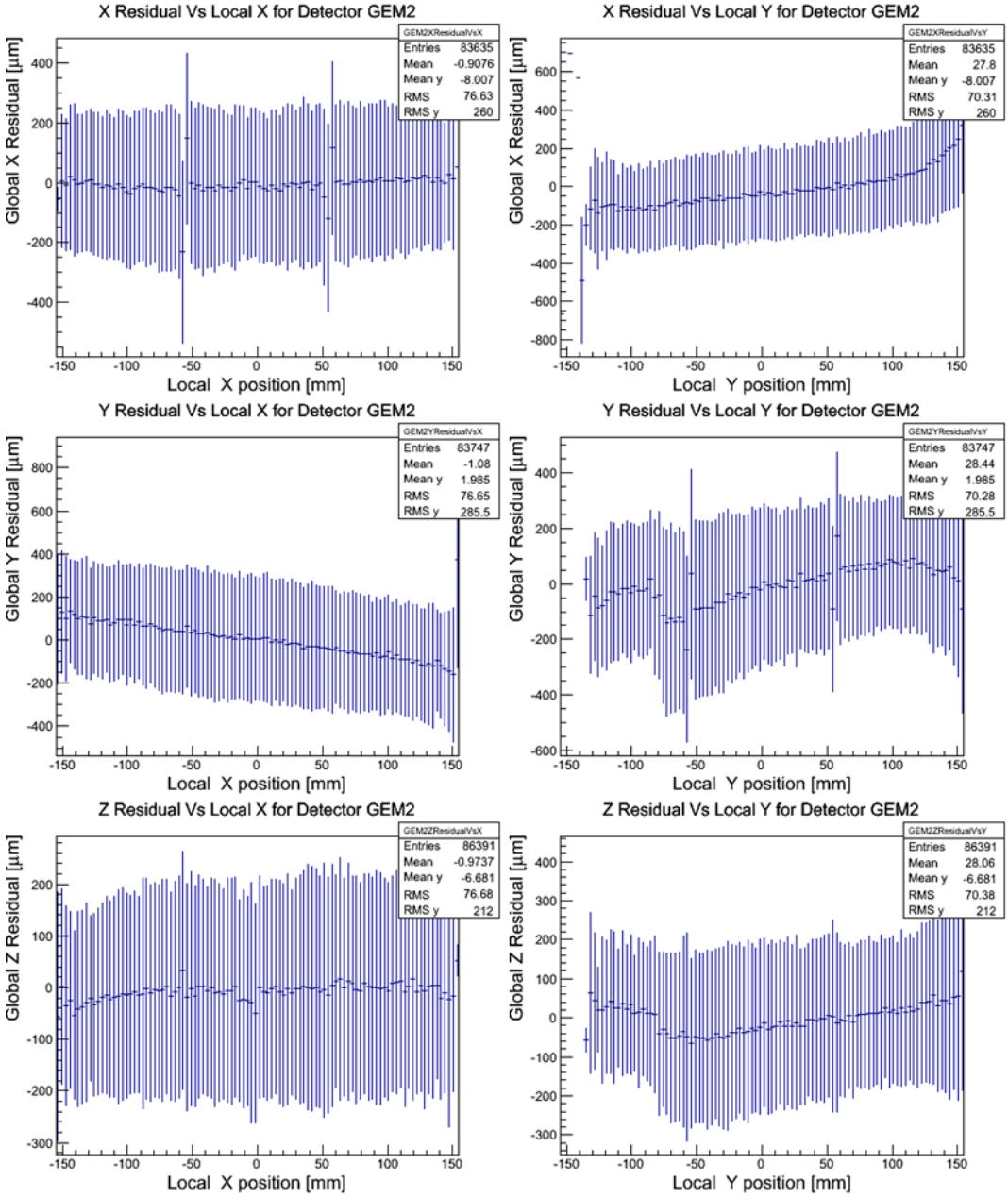
Residual Correlation Plots with Final Alignment

The residual correlation plots for all eight of the GEM detectors in the MTS appear on the following pages. The local coordinate system refers to the 2D coordinate system of the GEM detector itself, while the global coordinate system refers to the 3D MTS coordinate system. These correlation plots exhibit the effect of nonlinear misalignments on the quality of the detector alignment by the histogram method. The top row of histograms plot the global X residual against the local X and Y coordinate on the GEM detector. The middle row of histograms plot the global Y residual against the local X and Y coordinate on the GEM detector. The bottom row of histograms plot the global Z residual against the local X and Y coordinate on the GEM detector. The error bars indicate the RMS of the entries within the bin.

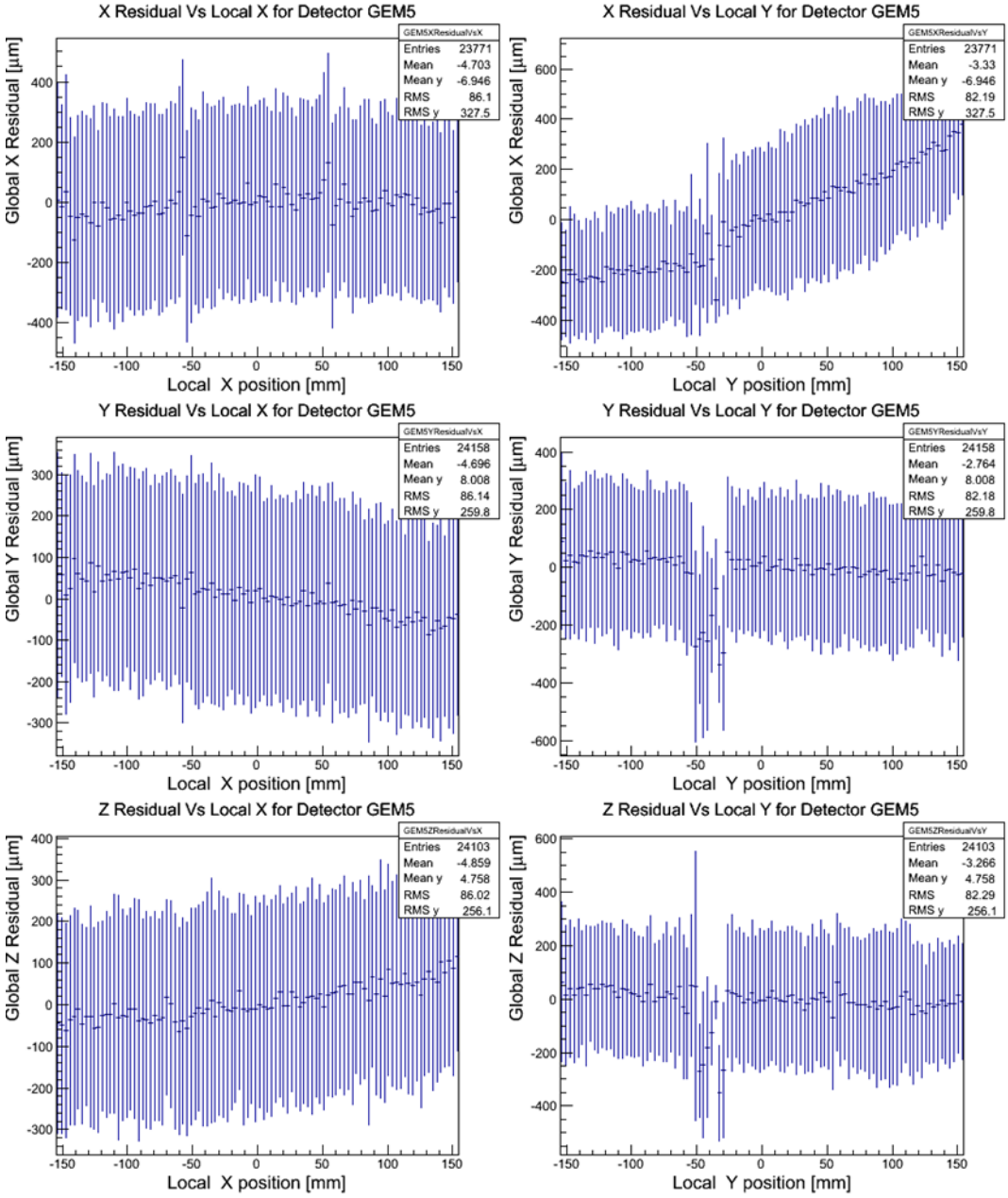
GEM1 Residual Correlation Plots



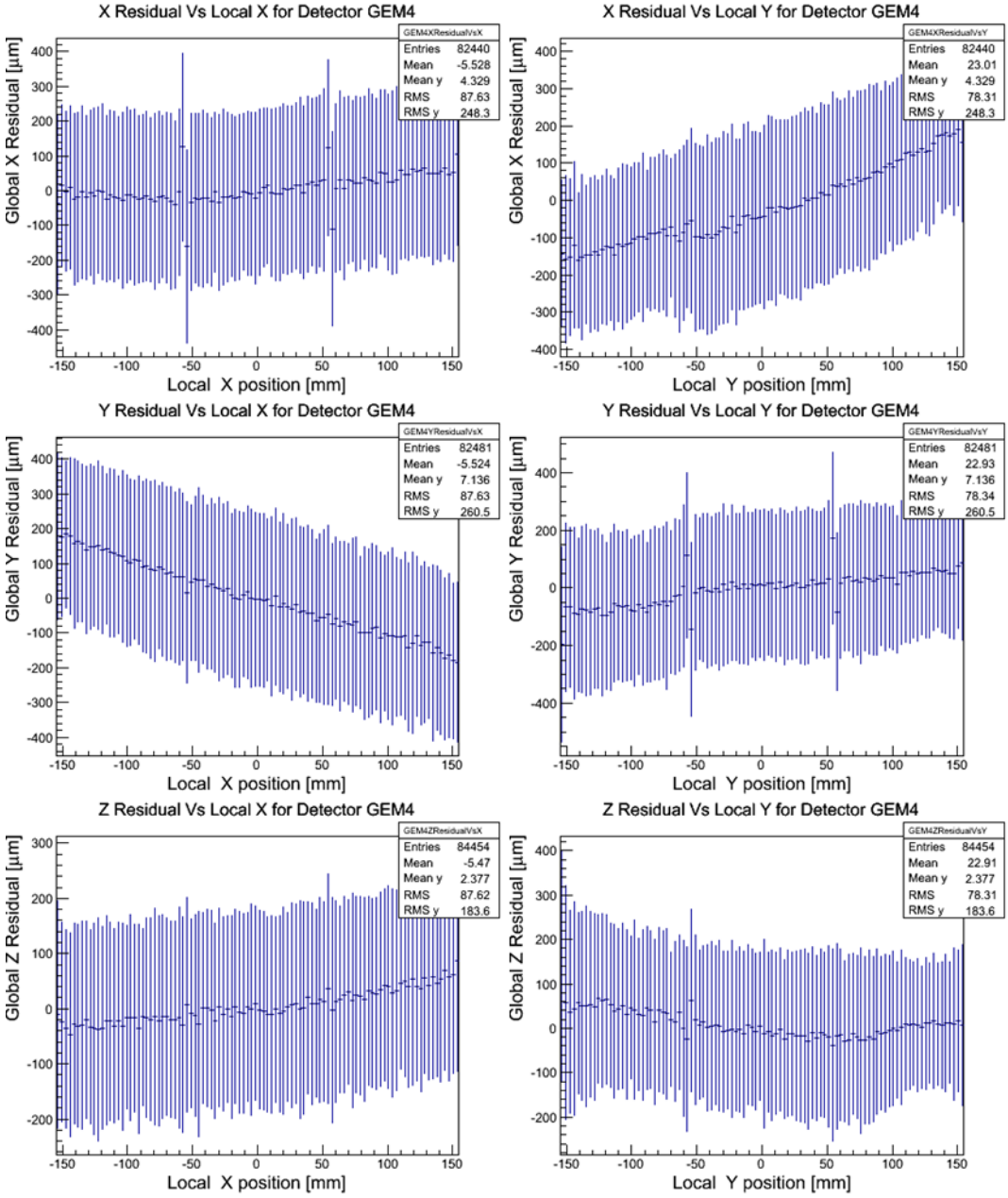
GEM2 Residual Correlation Plots



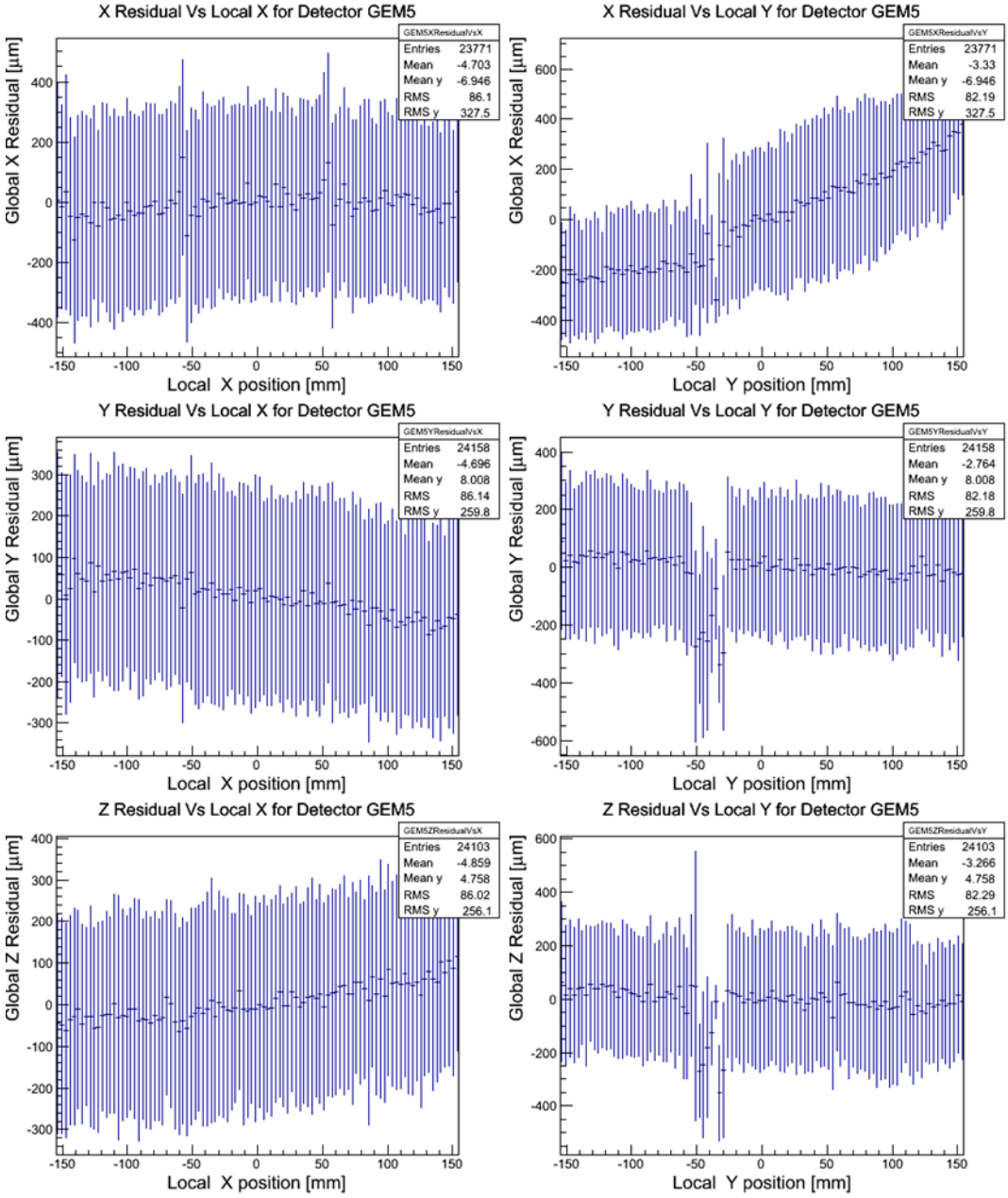
GEM3 Residual Correlation Plots



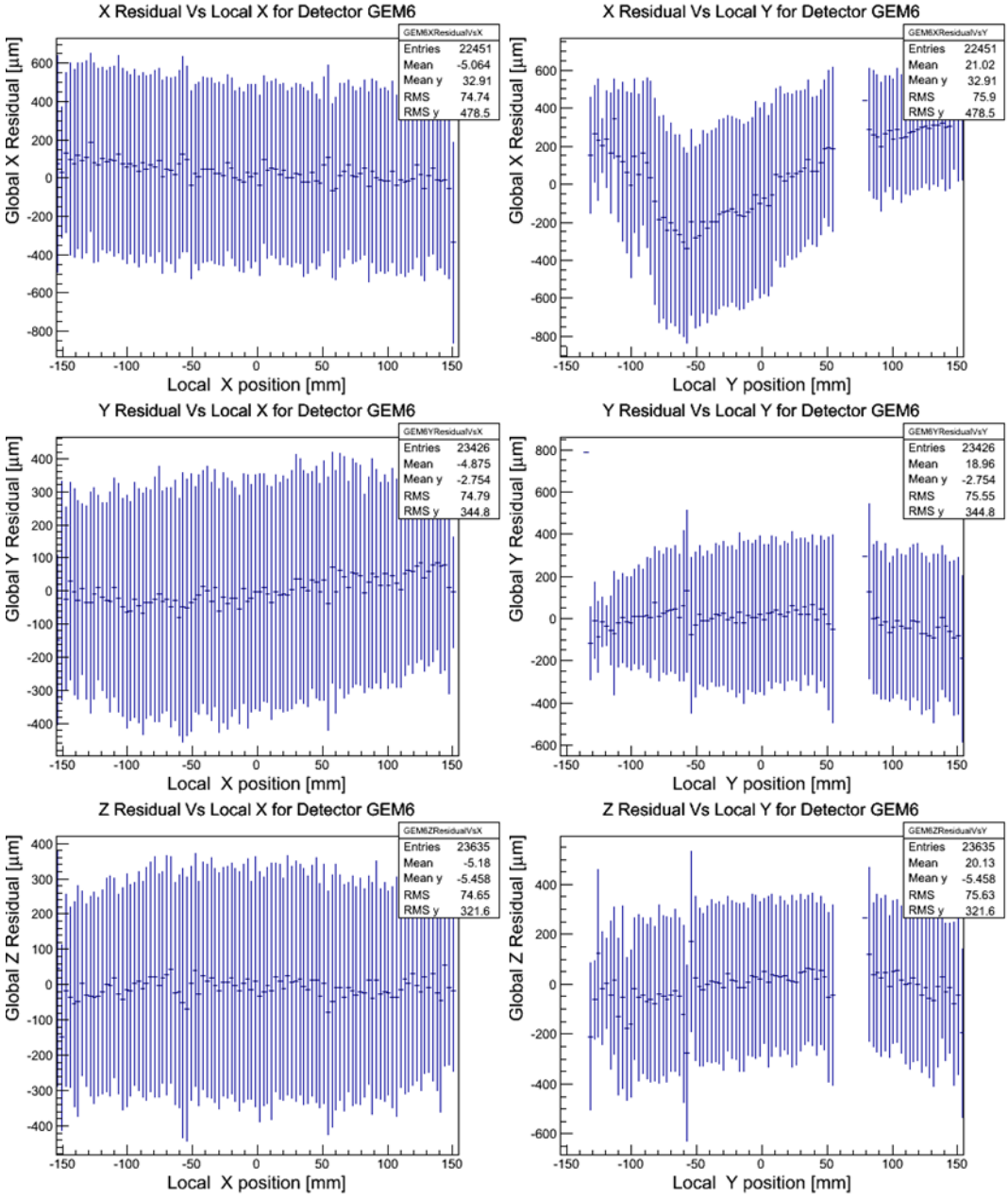
GEM4 Residual Correlation Plots



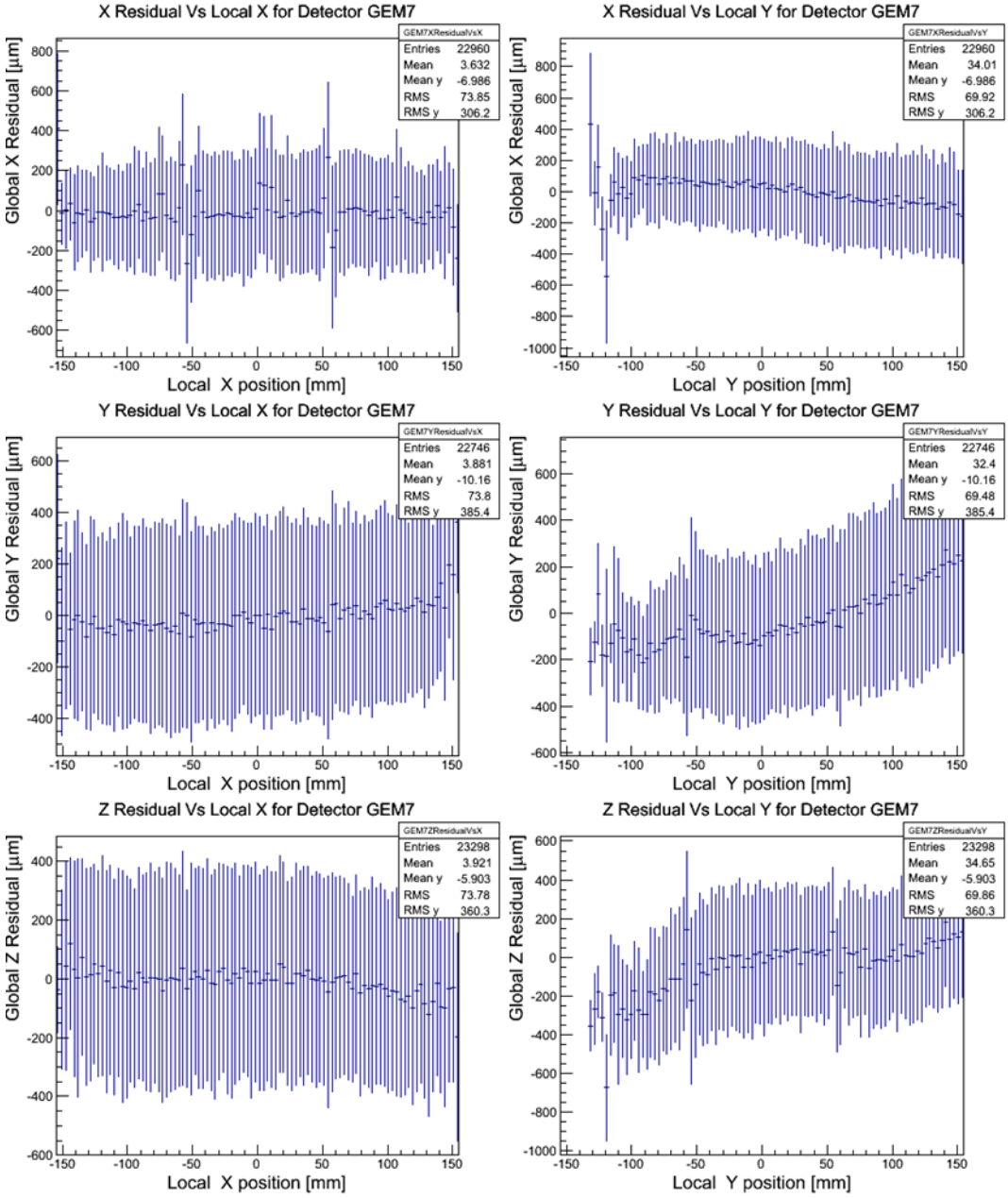
GEM5 Residual Correlation Plots



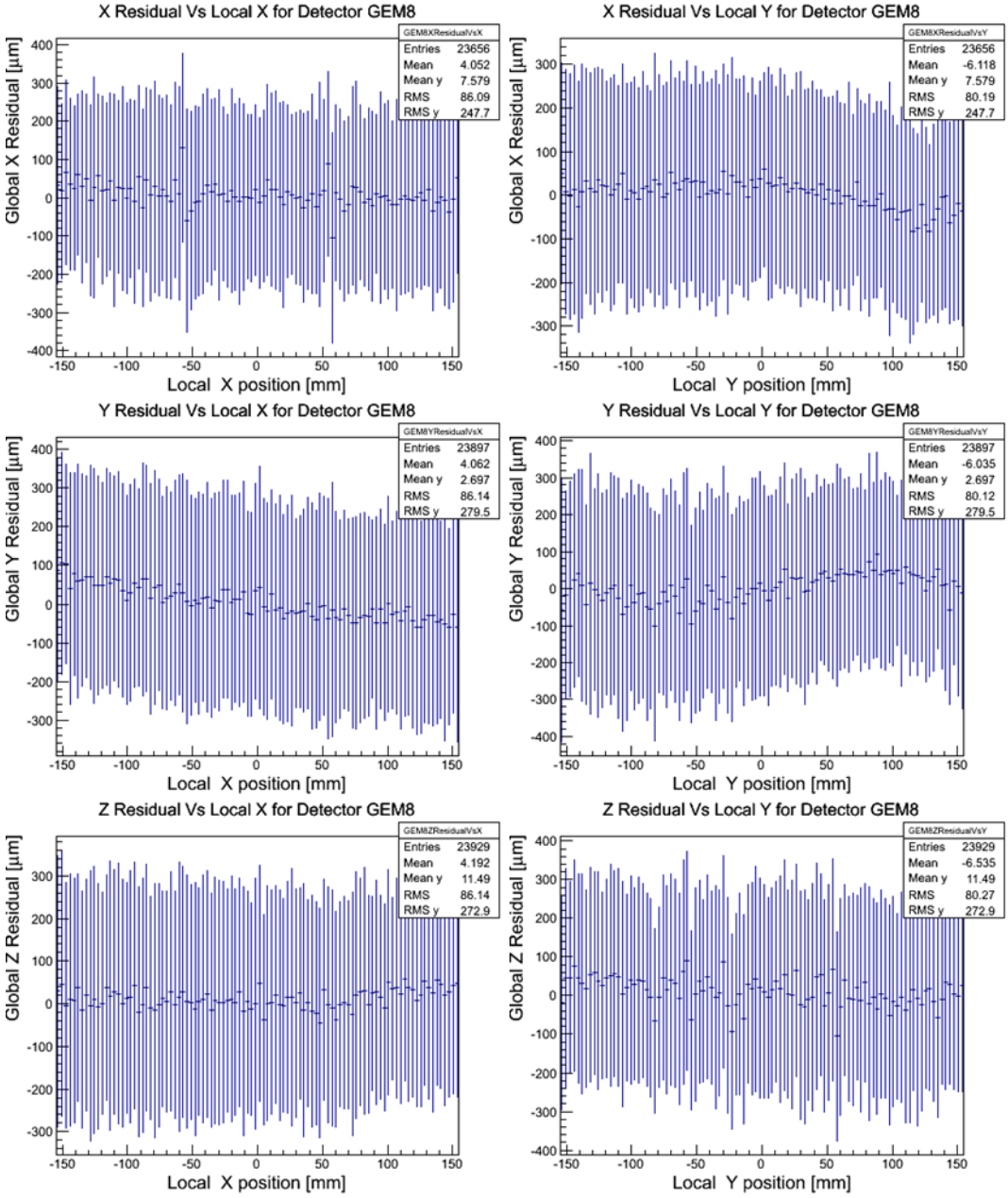
GEM6 Residual Correlation Plots



GEM7 Residual Correlation Plots



GEM8 Residual Correlation Plots



Appendix B

Bash Scripts used for the MTS

Two bash scripts are presented below. The first is a script that sends the slow-control commands to the SRS hardware via Ethernet. The second is an automation script for the processing of multiple raw data files using AMORE. The slow-control script is initiated during the start of run commands in DATE. The values assigned to each of the memory registers are contained in the text files used as an argument for the command /home/MTSUser/slow_control/slow_control in the script below. The C script used to send the commands to the FEC via Ethernet was developed by Sorin Martiou [29].

slowControlConfig.txt

```
#! /bin/bash
for (( i=$1;$i<$2;i++ ))
do
echo "======== CONFIGURATION OF SRS CARD $i"
echo ""
echo "set 10.0.$i.2 -> 10.0.$i.3"
/home/MTSUser/slow_control/slow_control /home/MTSUser/slow_control/set_ip$
i.txt
usleep 100000
echo "ADC_$i config"
/home/MTSUser/slow_control/slow_control /home/MTSUser/slow_control/adc_car
d$i.txt
usleep 100000
echo "FEC_$i config"
/home/MTSUser/slow_control/slow_control /home/MTSUser/slow_control/fec${i}
CosmicRun.txt
usleep 100000
echo "APV_$i reset"
/home/MTSUser/slow_control/slow_control /home/MTSUser/slow_control/fec${i}
apvreset.txt
usleep 100000
echo "APV_$i config"
/home/MTSUser/slow_control/slow_control /home/MTSUser/slow_control/apv$i.t
xt
usleep 100000
echo "PLL_$i config"
/home/MTSUser/slow_control/slow_control /home/MTSUser/slow_control/pll$i.t
xt
usleep 100000
done
```

set_ip0.txt

```
10.0.0.2
6007
80000000
00000000
aaaaffff
00000000
0000000a
0a000003
```

adc_card0.txt

```
10.0.0.2
6519
80000000
00000000
aaaaffff
00000000
# reset all apvs
00000000
00000000
# adc card registers
00000001
00000000
00000002
00000000
00000003
00000000
00000004
00000000
00000005
00000000
00000006
000000ff
# de-reset all apvs
00000000
000000ff
```

fec0CosmicRun.txt

```
10.0.0.2
6039
80000000
00000000
aaaaffff
00000000
00000000
00000004
00000001
00000001
00000002
00000980
00000003
0000007a
```

```
00000004
00000080
00000005
000000e1
00000008
0000ffff
00000009
000003e8
0000000a
00000002
0000000b
00000000
0000000c
00000200
ffffffff
00000001
```

apv0.txt

```
10.0.0.2
6263
80000000
0000ff03
aaaaffff
00000000
# mode
00000001
00000019
# latency
00000002
00000080
# muxgain
00000003
00000004
# ipre
00000010
00000062
# ipcasc
00000011
00000034
# ipsf
00000012
00000022
# isha
00000013
00000022
# issf
00000014
00000022
# ipsp
00000015
00000037
# imuxin
00000016
00000010
# ical
00000018
```

```
00000064
# vpsp
00000019
00000028
# vfs
0000001a
0000003c
# vfp
0000001b
0000001e
# cdrv
0000001c
000000ef
# csel
0000001d
000000f7
```

pll0.txt

```
10.0.0.2
6263
80000000
ffffff00
aaaaffff
00000000
00000001
00000000
```

ProcessDataSRS.sh

```
#!/bin/bash
# Bash script to process a given day of data from the MTS
# Author: Michael Staib
# This script will generate the pedestal files and run POCA analysis on
all files
# Change the lines below to match the log book for a given day

#########################################################################
# These Lines are the most important

listOfFiles="leadbox13apr1 leadbox13apr2"
pedestalFile="leadbox13aprPed"
agentToUse="SRS02"

#########################################################################

numEvents="1000"
numCycles="100"
numPedEvents="100"
numPedCycles="50"
configFile="${HOME}/configFileDirSRSNew/amoreSRSconfig.cfg"
#mapping="~/configFileDirSRSNew/mappingMTS8Detectors_5target.cfg"
mapping="~/configFileDirSRSNew/mappingUsedForThesis.cfg"
histos="~/configFileDirSRSNew/histosMTS8Detectors.cfg"
```

```
# You shouldn't have to change the lines below

rm $configFile
exec 3>$configFile
echo "HISPATH   ~/configFileDirSRSNew" >&3
echo "RAWPEDFILE
~/results/pedestalDir/rawPedestalSRS${pedestalFile}.root" >&3
echo "PEDFILE   ~/results/pedestalDir/pedestalSRS${pedestalFile}.root" >&3
echo "MAPFILE   ${mapping}" >&3
echo "HISTCFG   ${histos}" >&3
echo "CYCLEWAIT      0" >&3
echo "NBOFSIGMA      5" >&3
echo "RUNTYPE      RAWPEDESTAL" >&3
echo "RUNNAME        ${pedestalFile}" >&3

amoreAgent -a ${agentToUse} -s /mnt/raid/${pedestalFile}.raw -e
${numPedEvents} -c ${numPedCycles}

rm $configFile
exec 3>$configFile
echo "HISPATH   ~/configFileDirSRSNew" >&3
echo "RAWPEDFILE
~/results/pedestalDir/rawPedestalSRS${pedestalFile}.root" >&3
echo "PEDFILE   ~/results/pedestalDir/pedestalSRS${pedestalFile}.root" >&3
echo "MAPFILE   ${mapping}" >&3
echo "HISTCFG   ${histos}" >&3
echo "CYCLEWAIT      0" >&3
echo "NBOFSIGMA      5" >&3
echo "RUNTYPE      PEDESTAL" >&3
echo "RUNNAME        ${pedestalFile}" >&3

amoreAgent -a ${agentToUse} -s /mnt/raid/${pedestalFile}.raw -e
${numPedEvents} -c ${numPedCycles}

for datafile in $listOfFiles
do
rm $configFile
exec 3>$configFile
echo "HISPATH   ~/configFileDirSRSNew" >&3
echo "PEDFILE   ~/results/pedestalDir/pedestalSRS${pedestalFile}.root" >&3
echo "MAPFILE   ${mapping}" >&3
echo "HISTCFG   ${histos}" >&3
echo "CYCLEWAIT      0" >&3
echo "POCATXTFILE   ~/results/pocaResult/SRS${datafile}.txt" >&3
echo "POCATXTOUTPUT  YES" >&3
echo "TXTHITOUTPUT    YES" >&3
echo "HITTXTFILE    ~/results/HitTextFiles/SRS${datafile}.txt" >&3
echo "NBOFSIGMA      5" >&3
echo "TXTOUTFORMAT    NEW" >&3
echo "RUNTYPE      TRACKING" >&3
echo "RUNNAME        ${datafile}" >&3
echo "MAXCLUSTSIZE   10" >&3
echo "ALIGNTXTFILE   ~/results/AlignTextFiles/align${datafile}.txt" >&3

amoreAgent -a ${agentToUse} -s /mnt/raid/${datafile}.raw -e ${numEvents} -
c ${numCycles}
done
```

Appendix C

Configuration Files used for AMORE Modules

There are four major configuration files that are used with the AMORE modules that have been developed. The first is a general configuration file that must be present with the file name ~/configFileDir/amoreMTSTest.cfg for the amoreMTS module, or file name ~/configFileDirSRSNew/amoreSRSconfig.cfg for the amoreSRS module. These exact names can be changed in the source code. In this file it is possible to set the important parameters including selection of the run type, the names of the other configuration files to be used, as well as other run options. The mapping configuration file contains the mapping from APV hybrid to FEC and then to the detector plane. The positions and rotation of the GEM detectors and their assignment to tracking modules are set in this file as well as any alignment "shifts" that have been added to the data. An example of each of the files is contained below including the mapping configuration file used for this thesis. The alignment parameters used can be found in this mapping file. The histogram configuration file allows the user to decide what particular monitoring histograms will be published to the monitoring pool. The amoreDisplay.cfg file configures the settings for the custom GUI. The generic GUI may also be used and does not require any configuration file. Additional information may be found in reference [31].

amoreSRSconfig.cfg

```
#########################################################################
#  Setup Config Files and Pedestal Files                                #
#########################################################################


HISPATH      ~/configFileDirSRS
MAPFILE      ~/configFileDirSRS/mappingUsedForThesis.cfg
HISTCFG      ~/configFileDirSRS/histosMTS8Detectors.cfg
PEDFILE      ~/results/pedestalDir/pedestalTest.root
RAWPEDFILE   ~/results/pedestalDir/rawPedestalTest.root
```

```
##########################################################################
#  General Run Parameters                                                #
##########################################################################

RUNNAME        test
NBOFSIGMA      5
CYCLEWAIT      0
MAXCLUSTSIZE   10

##########################################################################
#  Setup Global Hit Text File output                                     #
##########################################################################

TXTHITOUTPUT    NO
HITTXTFILE     ~/results/HitTextFiles/test.txt

TXTOUTFORMAT    NEW
#TXTOUTFORMAT   OLD

##########################################################################
#  Setup POCA Text File output                                           #
##########################################################################

POCATXTOUTPUT  YES
POCATXTFILE   ~/results/pocaResult/test.txt

##########################################################################
#  Select Run Type                                                       #
##########################################################################

#RUNTYPE      RAWPEDESTAL
#RUNTYPE      PEDESTAL
RUNTYPE      TRACKING
#RUNTYPE      PHYSICS
#RUNTYPE      ALIGNMENT
```

mappingUsedForThesis.cfg

```
##############################################################################
#     DetName    PlaneName    DetNo    Plane   planeSize   nbConnectors
##############################################################################


DET,   GEM1,      GEM1TopX,      0,       0,      307.2,          6
DET,   GEM1,      GEM1TopY,      0,       1,      307.2,          6
DET,   GEM2,      GEM2TopX,      1,       0,      307.2,          6
DET,   GEM2,      GEM2TopY,      1,       1,      307.2,          6
DET,   GEM3,      GEM3BottomX,   2,       0,      307.2,          6
DET,   GEM3,      GEM3BottomY,   2,       1,      307.2,          6
DET,   GEM4,      GEM4BottomX,   3,       0,      307.2,          6
DET,   GEM4,      GEM4BottomY,   3,       1,      307.2,          6
DET,   GEM5,      GEM5LeftX,     4,       0,      307.2,          6
DET,   GEM5,      GEM5LeftY,     4,       1,      307.2,          6
```

```
DET,   GEM6,      GEM6LeftX,     5,       0,      307.2,          6
DET,   GEM6,      GEM6LeftY,     5,       1,      307.2,          6
DET,   GEM7,      GEM7RightX,    6,       0,      307.2,          6
DET,   GEM7,      GEM7RightY,    6,       1,      307.2,          6
DET,   GEM8,      GEM8RightX,    7,       0,      307.2,          6
DET,   GEM8,      GEM8RightY,    7,       1,      307.2,          6

##############################################################################
#  Local to Global Detector Position Mapping
#  Detector Station   X    Y     Z    RotationX   RotationY   RotationZ
##############################################################################

MTS, GEM1,    TOP,    0.7,   45.4505,  228.75,   0,          0,         180
MTS, GEM2,    TOP,    0.7,   45.4505,  153.25,   0,          0,         180
MTS, GEM3, BOTTOM,   -0.7,  -45.4505, -153.25,   0,          0,         0
MTS, GEM4, BOTTOM,   -0.7,  -45.4505, -228.75,   0,          0,         0
MTS, GEM5,   LEFT,    1.5,  -218.1,    35.2,     270,        0,         0
MTS, GEM6,   LEFT,    1.5,  -143.1,    35.2,     270,        0,         0
MTS, GEM7,  RIGHT,   -1.5,   143.1,   -54.4,     90,         0,         0
MTS, GEM8,  RIGHT,   -1.5,   218.1,   -54.4,     90,         0,         0

##############################################################################
#  Detector Shifts for Alignment
#     Detector      X             Y              Z        XRot  YRot  ZRot
##############################################################################

SHIFT, GEM1,  0.4150778671, -0.1969695701,  0.2409290728, 0.0,  0.0,  0.0
SHIFT, GEM2, -0.3941820800,  0.1749540232, -0.1587483130, 0.0,  0.0,  0.0
SHIFT, GEM3, -0.4089827542,  0.1999259089, -1.5132563900, 0.0,  0.0,  0.0
SHIFT, GEM4,  0.3286518522, -0.1811051783,  1.1038946484, 0.0,  0.0,  0.0
SHIFT, GEM5, -0.1763614425,  0.1824171576,  0.1725545277, 0.0,  0.0,  0.0
SHIFT, GEM6,  0.1697535627, -0.3010611160, -0.1670153554, 0.0,  0.0,  0.0
SHIFT, GEM7,  0.2272211259, -0.5862345338,  0.3199160774, 0.0,  0.0,  0.0
SHIFT, GEM8, -0.1086733997,  0.3925590304,  0.0177165291, 0.0,  0.0,  0.0

##############################################################################
#     FEC Name    APV Name        FEC Id   FEC chNo   APV No   APV Hdr
##############################################################################

FEC,    FEC1,   GEM1TopXApv1,       1,        0,         0,     1300
FEC,    FEC1,   GEM1TopXApv2,       1,        1,         1,     1300
FEC,    FEC1,   GEM1TopXApv3,       1,        2,         2,     1300
FEC,    FEC1,   GEM1TopXApv4,       1,        3,         3,     1300
FEC,    FEC1,   GEM1TopXApv5,       1,        4,         4,     1300
FEC,    FEC1,   GEM1TopXApv6,       1,        5,         5,     1300
FEC,    FEC1,   GEM1TopYApv1,       1,        6,         6,     1300
FEC,    FEC1,   GEM1TopYApv2,       1,        7,         7,     1300
FEC,    FEC1,   GEM1TopYApv3,       1,        8,         8,     1300
FEC,    FEC1,   GEM1TopYApv4,       1,        9,         9,     1300
FEC,    FEC1,   GEM1TopYApv5,       1,        10,       10,     1300
FEC,    FEC1,   GEM1TopYApv6,       1,        11,       11,     1300
FEC,    FEC1,   GEM2TopXApv1,       1,        12,       12,     1300
FEC,    FEC1,   GEM2TopXApv2,       1,        13,       13,     1300
FEC,    FEC1,   GEM2TopXApv3,       1,        14,       14,     1300
FEC,    FEC1,   GEM2TopXApv4,       1,        15,       15,     1300

FEC,    FEC2,   GEM2TopXApv5,       2,        0,        16,     1300
FEC,    FEC2,   GEM2TopXApv6,       2,        1,        17,     1300
```

```
FEC,    FEC2,   GEM2TopYApv1,       2,      2,      18,     1300
FEC,    FEC2,   GEM2TopYApv2,       2,      3,      19,     1300
FEC,    FEC2,   GEM2TopYApv3,       2,      4,      20,     1300
FEC,    FEC2,   GEM2TopYApv4,       2,      5,      21,     1300
FEC,    FEC2,   GEM2TopYApv5,       2,      6,      22,     1300
FEC,    FEC2,   GEM2TopYApv6,       2,      7,      23,     1300
FEC,    FEC2,   GEM3BottomXApv1,    2,      8,      24,     1300
FEC,    FEC2,   GEM3BottomXApv2,    2,      9,      25,     1300
FEC,    FEC2,   GEM3BottomXApv3,    2,      10,     26,     1300
FEC,    FEC2,   GEM3BottomXApv4,    2,      11,     27,     1300
FEC,    FEC2,   GEM3BottomXApv5,    2,      12,     28,     1300
FEC,    FEC2,   GEM3BottomXApv6,    2,      13,     29,     1300
FEC,    FEC2,   GEM3BottomYApv1,    2,      14,     30,     1300
FEC,    FEC2,   GEM3BottomYApv2,    2,      15,     31,     1300

FEC,    FEC3,   GEM3BottomYApv3,    3,      0,      32,     1300
FEC,    FEC3,   GEM3BottomYApv4,    3,      1,      33,     1300
FEC,    FEC3,   GEM3BottomYApv5,    3,      2,      34,     1300
FEC,    FEC3,   GEM3BottomYApv6,    3,      3,      35,     1300
FEC,    FEC3,   GEM4BottomXApv1,    3,      4,      36,     1300
FEC,    FEC3,   GEM4BottomXApv2,    3,      5,      37,     1300
FEC,    FEC3,   GEM4BottomXApv3,    3,      6,      38,     1300
FEC,    FEC3,   GEM4BottomXApv4,    3,      7,      39,     1300
FEC,    FEC3,   GEM4BottomXApv5,    3,      8,      40,     1300
FEC,    FEC3,   GEM4BottomXApv6,    3,      9,      41,     1300
FEC,    FEC3,   GEM4BottomYApv1,    3,      10,     42,     1300
FEC,    FEC3,   GEM4BottomYApv2,    3,      11,     43,     1300
FEC,    FEC3,   GEM4BottomYApv3,    3,      12,     44,     1300
FEC,    FEC3,   GEM4BottomYApv4,    3,      13,     45,     1300
FEC,    FEC3,   GEM4BottomYApv5,    3,      14,     46,     1300
FEC,    FEC3,   GEM4BottomYApv6,    3,      15,     47,     1300

FEC,    FEC4,   GEM5LeftXApv1,      4,      0,      48,     1300
FEC,    FEC4,   GEM5LeftXApv2,      4,      1,      49,     1300
FEC,    FEC4,   GEM5LeftXApv3,      4,      2,      50,     1300
FEC,    FEC4,   GEM5LeftXApv4,      4,      3,      51,     1300
FEC,    FEC4,   GEM5LeftXApv5,      4,      4,      52,     1300
FEC,    FEC4,   GEM5LeftXApv6,      4,      5,      53,     1300
FEC,    FEC4,   GEM5LeftYApv1,      4,      6,      54,     1300
FEC,    FEC4,   GEM5LeftYApv2,      4,      7,      55,     1300
FEC,    FEC4,   GEM5LeftYApv3,      4,      8,      56,     1300
FEC,    FEC4,   GEM5LeftYApv4,      4,      9,      57,     1300
FEC,    FEC4,   GEM5LeftYApv5,      4,      10,     58,     1300
FEC,    FEC4,   GEM5LeftYApv6,      4,      11,     59,     1300
FEC,    FEC4,   GEM6LeftXApv1,      4,      12,     60,     1300
FEC,    FEC4,   GEM6LeftXApv2,      4,      13,     61,     1300
FEC,    FEC4,   GEM6LeftXApv3,      4,      14,     62,     1300
FEC,    FEC4,   GEM6LeftXApv4,      4,      15,     63,     1300

FEC,    FEC5,   GEM6LeftXApv5,      5,      0,      64,     1300
FEC,    FEC5,   GEM6LeftXApv6,      5,      1,      65,     1300
FEC,    FEC5,   GEM6LeftYApv1,      5,      2,      66,     1300
FEC,    FEC5,   GEM6LeftYApv2,      5,      3,      67,     1300
FEC,    FEC5,   GEM6LeftYApv3,      5,      4,      68,     1300
FEC,    FEC5,   GEM6LeftYApv4,      5,      5,      69,     1300
FEC,    FEC5,   GEM6LeftYApv5,      5,      6,      70,     1300
FEC,    FEC5,   GEM6LeftYApv6,      5,      7,      71,     1300
FEC,    FEC5,   GEM7RightXApv1,     5,      8,      72,     1300
```

```
FEC,    FEC5,   GEM7RightXApv2,     5,        9,        73,     1300
FEC,    FEC5,   GEM7RightXApv3,     5,        10,       74,     1300
FEC,    FEC5,   GEM7RightXApv4,     5,        11,       75,     1300
FEC,    FEC5,   GEM7RightXApv5,     5,        12,       76,     1300
FEC,    FEC5,   GEM7RightXApv6,     5,        13,       77,     1300
FEC,    FEC5,   GEM7RightYApv1,     5,        14,       78,     1300
FEC,    FEC5,   GEM7RightYApv2,     5,        15,       79,     1300

FEC,    FEC6,   GEM7RightYApv3,     6,        0,        80,     1300
FEC,    FEC6,   GEM7RightYApv4,     6,        1,        81,     1300
FEC,    FEC6,   GEM7RightYApv5,     6,        2,        82,     1300
FEC,    FEC6,   GEM7RightYApv6,     6,        3,        83,     1300
FEC,    FEC6,   GEM8RightXApv1,     6,        4,        84,     1300
FEC,    FEC6,   GEM8RightXApv2,     6,        5,        85,     1300
FEC,    FEC6,   GEM8RightXApv3,     6,        6,        86,     1300
FEC,    FEC6,   GEM8RightXApv4,     6,        7,        87,     1300
FEC,    FEC6,   GEM8RightXApv5,     6,        8,        88,     1300
FEC,    FEC6,   GEM8RightXApv6,     6,        9,        89,     1300
FEC,    FEC6,   GEM8RightYApv1,     6,        10,       90,     1300
FEC,    FEC6,   GEM8RightYApv2,     6,        11,       91,     1300
FEC,    FEC6,   GEM8RightYApv3,     6,        12,       92,     1300
FEC,    FEC6,   GEM8RightYApv4,     6,        13,       93,     1300
FEC,    FEC6,   GEM8RightYApv5,     6,        14,       94,     1300
FEC,    FEC6,   GEM8RightYApv6,     6,        15,       95,     1300
```

histosMTS8Detectors.cfg

```
# Histos configuration. Separate columns with commas. Don't use comma in
histo title
# Comments start with (only at the beginning of line)
#
# ==== General Format:
# TYPE, NAME, TITLE, GEM PLANE1 (GEM PLANE2)
# Please make sure histo name is UNIQUE, use -1, -1 if the histo is not to
be drawn anywhere
#
# Type can be one of the following:
#   - SPECTRUM1D : 1D Histo=> Total (x+y)-plane pulse height spectrum for
the current run
#   - HITZEROSUP      : 1D Histo=> Position (x) and cluster charge (y)
display of the hit for the current event
#   - HIT2D      : 2D Histo=> Position (x,y) and total charge (z) display
of the hit for the current event
#   - HITDIST    : 1D Histo=> Spatial (x or y) charge distribution (w =
acumulated charges) per plane for the current run
#   - HITMAP     : 2D Histo=> Spatial (x vs y) charge distribution (w =
acumulated charges) per plane for the current run
#   - CHARGES_SH : 1D Histo=> Charge sharing between x and y planes for
the current run
#   - CLUSTSIZE    : 1D Histo=> Cluster size distribution per detector
plane for the current run
#   - CLUSTCOR  : 1D Histo=> Cluster size correlation between x and y
planes for the current run
```

```
# Other types may be found by browsing SRSHistoManager.cxx in the source
code

###################### BINNING ##############################
#Type,   Histo Name,     Histo Title,         GEM Plane1
#############################################################
BINNING,   HITZEROSUP,   768, -0.5, 767.5
BINNING,   SPECTRUM,     100, -0.5, 1995.5
BINNING,   CLUSTSIZE,    11,   0,   10
BINNING,   CLUSTMULT,    11,   0,   10
BINNING,   HITMAP,       31,   -15.5,   14.5
BINNING,   ENERGYMAP,    31,   -15.5,   14.5
BINNING2D, PLANETIMEBIN, 768, -0.5, 767.5, 14, 0, 13
BINNING2D, CLUSTCOR,     11, 0, 10, 32, -0.5, 4095.5
BINNING,   SPECTRUM,     100, -0.5, 1995.5

############################ HIT ############################
#Type,   Histo Name,     Histo Title,         GEM Plane1
#############################################################
#HITZEROSUP,    ZeroSupHit1DGEM1TopX,    Top1 (GEM2) Hit on X,
GEM1TopX
#HITZEROSUP,    ZeroSupHit1DGEM1TopY,    Top1 (GEM2) Hit on Y,
GEM1TopY
#HITZEROSUP,    ZeroSupHit1DGEM2TopX,    Top2 (GEM7) Hit on X,
GEM2TopX
#HITZEROSUP,    ZeroSupHit1DGEM2TopY,    Top2 (GEM7) Hit on Y,
GEM2TopY
#HITZEROSUP,    ZeroSupHit1DGEM3BottomX, Bottom1 (GEM3) Hit on X,
GEM3BottomX
#HITZEROSUP,    ZeroSupHit1DGEM3BottomY, Bottom1 (GEM3) Hit on Y,
GEM3BottomY
#HITZEROSUP,    ZeroSupHit1DGEM4BottomX, Bottom2 (GEM3) Hit on X,
GEM4BottomX
#HITZEROSUP,    ZeroSupHit1DGEM4BottomY, Bottom2 (GEM3) Hit on Y,
GEM4BottomY
#HITZEROSUP,    ZeroSupHit1DGEM5LeftX,   Bottom2 (GEM10) Hit on X,
GEM5LeftX
#HITZEROSUP,    ZeroSupHit1DGEM5LeftY,   Bottom2 (GEM10) Hit on Y,
GEM5LeftY
#HITZEROSUP,    ZeroSupHit1DGEM6LeftX,   Bottom2 (GEM10) Hit on X,
GEM6LeftX
#HITZEROSUP,    ZeroSupHit1DGEM6LeftY,   Bottom2 (GEM10) Hit on Y,
GEM6LeftY
#HITZEROSUP,    ZeroSupHit1DGEM7RightX,  Bottom3 (GEM10) Hit on X,
GEM7RightX
#HITZEROSUP,    ZeroSupHit1DGEM7RightY,  Bottom3 (GEM10) Hit on Y,
GEM7RightY
#HITZEROSUP,    ZeroSupHit1DGEM8RightX,  Bottom3 (GEM10) Hit on X,
GEM8RightX
#HITZEROSUP,    ZeroSupHit1DGEM8RightY,  Bottom3 (GEM10) Hit on Y,
GEM8RightY



##############################  HIT & ENERGY MAP ######################
#Type,    Histo Name,        Histo Title,        GEM Plane1   GEM Plane2
#########################################################################
```

```
#HITMAP,      Hit2DColZGEM1Top,     Top1 (GEM2) 2D Hit Position Map,
GEM1TopX,       GEM1TopY
#HITMAP,      Hit2DColZGEM2Top,     Top2 (GEM7) 2D Hit Position Map,
GEM2TopX,       GEM2TopY
#HITMAP,      Hit2DColZGEM3Bottom,  Bottom1 (GEM3) 2D Hit Position Map,
GEM3BottomX,       GEM3BottomY
#HITMAP,      Hit2DColZGEM4Bottom,  Bottom2 (GEM10) 2D Hit Position Map,
GEM4BottomX,       GEM4BottomY

#ENERGYMAP,   Energy2DColZGEM1Top,     Top1 (GEM2) 2D Hit Charges Map,
GEM1TopX,       GEM1TopY
#ENERGYMAP,   Energy2DColZGEM2Top,     Top2 (GEM7) 2D Hit Charges Map,
GEM2TopX,       GEM2TopY
#ENERGYMAP,   Energy2DColZGEM3Bottom,  Bottom1 (GEM3) 2D Hit Charges Map,
GEM3BottomX,       GEM3BottomY
#ENERGYMAP,   Energy2DColZGEM4Bottom,  Bottom2 (GEM10) 2D Hit Charges Map,
GEM4BottomX,       GEM4BottomY

############################### HIT DISTRIBUTION ####################
#Type,     Histo Name,        Histo Title,              GEM Plane1
#########################################################################

#HITDIST,   HitDistGEM1TopX,      Top1 (GEM2) X-Hit Distribution,
GEM1TopX
#HITDIST,   HitDistGEM1TopY,      Top1 (GEM2) Y-Hit Distribution,
GEM1TopY
#HITDIST,   HitDistGEM2TopX,      Top2 (GEM7) X-Hit Distribution,
GEM2TopX
#HITDIST,   HitDistGEM2TopY,      Top2 (GEM7) Y-Hit Distribution,
GEM2TopY
#HITDIST,   HitDistGEM3BottomX,   Bottom1 (GEM3) X-Hit Distribution,
GEM3BottomX
#HITDIST,   HitDistGEM3BottomY,   Bottom1 (GEM3) Y-Hit Distribution,
GEM3BottomY
#HITDIST,   HitDistGEM4BottomX,   Bottom2 (GEM10) X-Hit Distribution,
GEM4BottomX
#HITDIST,   HitDistGEM4BottomY,   Bottom2 (GEM10) Y-Hit Distribution,
GEM4BottomY

######################### CHARGES SHARING CORRELATION ###################
#Type,         Histo Name,     Histo Title,         GEM Plane1   GEM Plane2
#########################################################################

#CHARGES_SH,   ChargeSh2DColZGEM1Top,      Top1 (GEM2) Charge sharing,
GEM1TopX,       GEM1TopY
#CHARGES_SH,   ChargeSh2DColZGEM2Top,      Top2 (GEM7) Charge sharing,
GEM2TopX,       GEM2TopY
#CHARGES_SH,   ChargeSh2DColZGEM3Bottom,   Bottom1 (GEM3) Charge sharing,
GEM3BottomX,    GEM3BottomY
#CHARGES_SH,   ChargeSh2DColZGEM4Bottom,   Bottom2 (GEM10)  Charge
sharing,  GEM4BottomX,    GEM4BottomY
```

```
############ CHARGES SHARING CORRELATION ##############################
#Type,         Histo Name,          Histo Title,                GEM Plane1
#######################################################################

#HITCROSSTALK,   HitXtalk2DGEM2TopX,  GEM2 X-strip crosstalk,     GEM2TopX
#HITCROSSTALK,   HitXtalk2DGEM2TopY,  GEM2 Y-strip crosstalk,     GEM2TopY

############ CHARGES SHARING CORRELATION ##############################
#Type,         Histo Name,          Histo Title,                GEM Plane1
#######################################################################

#CHARGECROSSTALK,   ChargeXtalk2DGEM2TopX,  GEM2 X-charge crosstalk,
GEM2TopX
#CHARGECROSSTALK,   ChargeXtalk2DGEM2TopY,  GEM2 Y-charge crosstalk,
GEM2TopY

##############  CLUSTSIZE SIZE CORRELATION ############################
#Type,        Histo Name,         Histo Title,       GEM Plane
#######################################################################

#CLUSTCOR,  ClustCor2DColZGEM2TopX,      GEM2 X-Cluster charge sum vs.
strip count,       GEM2TopX
#CLUSTCOR,  ClustCor2DColZGEM2TopY,      GEM2 Y-Cluster charge sum vs.
strip count,       GEM2TopY

############  CLUSTER SIZE DISTRIBUTION ##################################
#Type,        Histo Name,          Histo Title,                  GEM Plane
#######################################################################################

#CLUSTSIZE,   ClustSizeGEM1TopX,      Top1 (GEM2) X-Hit Cluster size,
GEM1TopX
#CLUSTSIZE,   ClustSizeGEM1TopY,      Top1 (GEM2) Y-Hit Cluster size,
GEM1TopY
#CLUSTSIZE,   ClustSizeGEM2TopX,      Top2 (GEM7) X-Hit Cluster size,
GEM2TopX
#CLUSTSIZE,   ClustSizeGEM2TopY,      Top2 (GEM7) Y-Hit Cluster size,
GEM2TopY
#CLUSTSIZE,   ClustSizeGEM3BottomX,      Bottom1 (GEM3) X-Hit Cluster
size,       GEM3BottomX
#CLUSTSIZE,   ClustSizeGEM3BottomY,      Bottom1 (GEM3) Y-Hit Cluster
size,       GEM3BottomY
#CLUSTSIZE,   ClustSizeGEM4BottomX,      Bottom2 (GEM10) X-Hit Cluster
size,       GEM4BottomX
#CLUSTSIZE,   ClustSizeGEM4BottomY,      Bottom2 (GEM10) Y-Hit Cluster
size,       GEM4BottomY


########### CLUSTER MULTIPLICITY ######################################
#Type,        Histo Name,          Histo Title,                  GEM Plane
#######################################################################################

#CLUSTMULT,   ClustMultGEM1TopX,      Top1 (GEM2) X-Hit Cluster
multiplicity,       GEM1TopX
#CLUSTMULT,   ClustMultGEM1TopY,      Top1 (GEM2) Y-Hit Cluster
multiplicity,       GEM1TopY
#CLUSTMULT,   ClustMultGEM2TopX,      Top2 (GEM7) X-Hit Cluster
multiplicity,       GEM2TopX
```

```
#CLUSTMULT,   ClustMultGEM2TopY,      Top2 (GEM7) Y-Hit Cluster
multiplicity,       GEM2TopY
#CLUSTMULT,   ClustMultGEM3BottomX,      Bottom1 (GEM3)  X-Hit Cluster
multiplicity,       GEM3BottomX
#CLUSTMULT,   ClustMultGEM3BottomY,      Bottom1 (GEM3)  Y-Hit Cluster
multiplicity,       GEM3BottomY
#CLUSTMULT,   ClustMultGEM4BottomX,      Bottom2 (GEM10) X-Hit Cluster
multiplicity,       GEM4BottomX
#CLUSTMULT,   ClustMultGEM4BottomY,      Bottom2 (GEM10) Y-Hit Cluster
multiplicity,       GEM4BottomY

####################### SPECTRUM #######################################
#Type,        Histo Name,        Histo Title,               GEM Plane1
########################################################################

#SPECTRUM,   SpectrumGEM1TopX,         Top1 (GEM2) X-Hit Pulse height
distribution,       GEM1TopX
#SPECTRUM,   SpectrumGEM1TopY,         Top1 (GEM2) Y-Hit Pulse height
distribution,       GEM1TopY
#SPECTRUM,   SpectrumGEM2TopX,         Top2 (GEM7) X-Hit Pulse height
distribution,       GEM2TopX
#SPECTRUM,   SpectrumGEM2TopY,         Top2 (GEM7) Y-Hit Pulse height
distribution,       GEM2TopY
#SPECTRUM,   SpectrumGEM3BottomX,      Bottom1 (GEM3) X-Hit Pulse height
distribution,    GEM3BottomX
#SPECTRUM,   SpectrumGEM3BottomY,      Bottom1 (GEM3) Y-Hit Pulse height
distribution,    GEM3BottomY
#SPECTRUM,   SpectrumGEM4BottomX,      Bottom2 (GEM10) X-Hit Pulse height
distribution,   GEM4BottomX
#SPECTRUM,   SpectrumGEM4BottomY,      Bottom2 (GEM10) Y-Hit Pulse height
distribution,   GEM4BottomY

#################### RAW DATA ##########################################
#Type,        Histo Name,    Histo Title, FEC number, Channel number
########################################################################

RAWDATA,     apvNo0RawData,     Fec1 ch0,    1,         0
RAWDATA,     apvNo1RawData,     Fec1 ch1,    1,         1
RAWDATA,     apvNo2RawData,     Fec1 ch2,    1,         2
RAWDATA,     apvNo3RawData,     Fec1 ch3,    1,         3
RAWDATA,     apvNo4RawData,     Fec1 ch4,    1,         4
RAWDATA,     apvNo5RawData,     Fec1 ch5,    1,         5
RAWDATA,     apvNo6RawData,     Fec1 ch6,    1,         6
RAWDATA,     apvNo7RawData,     Fec1 ch7,    1,         7

####################### PEDESTALS  #####################################
#Type,       Histo Name,       Histo Title,   FEC number, Channel number
########################################################################

#PEDESTALS,     apvNo0PedestalMean,     Fec1 ch0 APVn0,    1,         0
#PEDESTALS,     apvNo1PedestalMean,     Fec1 ch1 APVn1,    1,         1
#PEDESTALS,     apvNo2PedestalMean,     Fec1 ch2 APVn2,    1,         2
#PEDESTALS,     apvNo3PedestalMean,     Fec1 ch3 APVn3,    1,         3
#PEDESTALS,     apvNo4PedestalMean,     Fec1 ch4 APVn4,    1,         4
#PEDESTALS,     apvNo5PedestalMean,     Fec1 ch5 APVn5,    1,         5
#PEDESTALS,     apvNo6PedestalMean,     Fec1 ch6 APVn6,    1,         6
#PEDESTALS,     apvNo7PedestalMean,     Fec1 ch7 APVn7,    1,         7
```

```
#PEDESTALS,     apvNo0PedestalRMS,     Fec1 ch0 APVn0,    1,         0
#PEDESTALS,     apvNo1PedestalRMS,     Fec1 ch1 APVn1,    1,         1
#PEDESTALS,     apvNo2PedestalRMS,     Fec1 ch2 APVn2,    1,         2
#PEDESTALS,     apvNo3PedestalRMS,     Fec1 ch3 APVn3,    1,         3
#PEDESTALS,     apvNo4PedestalRMS,     Fec1 ch4 APVn4,    1,         4
#PEDESTALS,     apvNo5PedestalRMS,     Fec1 ch5 APVn5,    1,         5
#PEDESTALS,     apvNo6PedestalRMS,     Fec1 ch6 APVn6,    1,         6
#PEDESTALS,     apvNo7PedestalRMS,     Fec1 ch7 APVn7,    1,         7
```

amoreDisplay.cfg

```
############################################################
# CANVAS,    NAME,        NTABX,   NTABY,  SIZEX,   SIZEY
############################################################

#CANVAS,     RAWDATA,          2,       3,      1000,   800
#CANVAS,     PEDESTAL,         2,       2,      1000,   800
#CANVAS,     PEDESTAL2,        3,       4,      1000,   800
#CANVAS,      HIT,             2,       2,      1000,   800
#CANVAS,      HITPED,          2,       2,      1000,   800
#CANVAS,      HITRAWDATA,      2,       2,      1000,   800
#CANVAS,      HITMAP,          1,       2,      1000,   800
CANVAS,      HITDIST,          2,       2,      1000,   800
#CANVAS,      CLUST,           1,       1,      1000,   800
#CANVAS,      CLUSTMULT,       3,       2,      1000,   800
#CANVAS,      SPECTRUM,        1,       1,      1000,   800
#CANVAS,      OFFSET,          2,       2,      1000,   800

#RAWDATA,     apvNo0RawData,     Fec1 APVn0 raw data
#RAWDATA,     apvNo1RawData,     Fec1 APVn1 raw data
#RAWDATA,     ZeroSupHit1DGEM1TopX,   GEM1 Hit on Y-plane
#RAWDATA,     apvNo2RawData,     Fec1 APVn2 raw data
#RAWDATA,     apvNo3RawData,     Fec1 APVn3 raw data
#RAWDATA,     ZeroSupHit1DGEM1TopY,   GEM1 Hit on Y-plane
#RAWDATA,     apvNo4RawData,     Fec2 APVn4 raw data
#RAWDATA,     apvNo5RawData,     Fec2 APVn5 raw data
#RAWDATA,     apvNo6RawData,     Fec2 APVn6 raw data
#RAWDATA,     apvNo7RawData,     Fec2 APVn7 raw data
#RAWDATA,     apvNo8RawData,     Fec2 APVn8 raw data
#RAWDATA,     apvNo9RawData,     Fec2 APVn9 raw data

#PEDESTAL,     apvNo0PedestalMean,     Fec1 APVn0 raw data
#PEDESTAL,     apvNo8PedestalMean,     Fec1 APVn1 raw data
#PEDESTAL,     apvNo0PedestalRMS,     Fec1 APVn2 raw data
#PEDESTAL,     apvNo8PedestalRMS,     Fec1 APVn3 raw data
#PEDESTAL,     apvNo0PedestalMean,     Fec2 APVn4 raw data
#PEDESTAL,     apvNo1PedestalMean,     Fec2 APVn5 raw data
#PEDESTAL,     apvNo6PedestalMean,     Fec2 APVn6 raw data
#PEDESTAL,     apvNo8PedestalMean,     Fec2 APVn7 raw data

#OFFSET, OffsetXGEM12, OffsetXGEM
#OFFSET, OffsetYGEM12, OffsetYGEM
#OFFSET, OffsetXGEM13, OffsetXGEM
#OFFSET, OffsetXGEM23, OffsetYGEM

#HIT,  RawDataHit1DGEM1TopX,       GEM
#HIT,  PedSubHit1DGEM1TopX,        GEM1 Hit on X-plane
```

```
#HIT,  ZeroSupHit1DGEM1TopX,       GEM1 Hit on Y-plane
#HIT,  timeLego2DHit1DGEM1TopX,    GEM1 Hit on X-plane

#HITPED,  PedSubHit1DGEM1TopX, GEM1 Hit on X-plane
#HITPED,  PedSubHit1DGEM2TopY, GEM1 Hit on Y-plane
#HITPED,  PedSubHit1DGEM1TopX, GEM2 Hit on X-plane
#HITPED,  PedSubHit1DGEM1TopY, GEM2 Hit on Y-plane
#HITPED,  PedSubHit1DGEM3TopX, GEM3 Hit on X-plane

#HITRAWDATA,  RawDataHit1DGEM1TopX, GEM1 Hit on X-plane
#HITRAWDATA,  RawDataHit1DGEM2TopY, GEM1 Hit on Y-plane
#HITRAWDATA,  RawDataHit1DGEM1TopX, GEM2 Hit on X-plane
#HITRAWDATA,  RawDataHit1DGEM1TopY, GEM2 Hit on Y-plane

HITDIST,  Hit2DColZGEM1Top,      GEM1 2D Hit
HITDIST,  HitDistGEM1TopX,      GEM1 X-Hit Distribution
#HITDIST,  HitDistGEM1TopY,      GEM1 Y-Hit Distribution
HITDIST,  SpectrumGEM1TopX,        GEM1 X Pulse Height X
HITDIST,  SpectrumGEM1TopY,        GEM1 Y Pulse Height X

#HITDIST,  HitDistGEM2TopX,      GEM2 X-Hit Distribution
#HITDIST,  HitDistGEM2TopY,      GEM2 Y-Hit Distribution
#HITDIST,  HitDistGEM3StripNbTopX,      GEM2 Y-Hit Distribution
#HITDIST,  HitDistGEM4TopX,      GEM2 Y-Hit Distribution

#HITDIST,  HitDistGEM1TopX,       GEM1 X-Hit Distribution
#HITDIST,  HitDistGEM1TopY,       GEM1 Y-Hit Distribution
#HITDIST,  ClustSizeGEM1TopX,     GEM1 X-Cluster strip count
#HITDIST,  ClustSizeGEM1TopY,     GEM1 X-Cluster strip count
#HITDIST,  SpectrumGEM1TopX,      GEM1 X Pulse Height X
#HITDIST,  SpectrumGEM1TopY,      GEM1 X Pulse Height X
#HITDIST,  Hit2DColZGEM1Top,      Triple-GEM1 2D Hit Position Map
#HITDIST,  ChargeSh2DColZGEM1Top, Triple-GEM1 2D Hit Charges Map

#HITMAP,  Hit2DColZGEM1Top,      GEM1 2D Hit
#HITMAP,  Hit2DColZGEM2Top,      GEM2 2D Hit
#HITMAP,  Energy2DColZGEM1Top,   GEM1 2D Hit
#HITMAP,  Energy2DColZGEM2Top,   GEM2 2D Hit
#HITMAP,  ChargeSh2DColZGEM1Top, GEM1 Charge sharing
#HITMAP,  ChargeSh2DColZGEM2Top, GEM2 Charge sharing

#CLUST,   ClustSizeGEM1TopX,      GEM1 X-Cluster strip count
#CLUST,   ClustSizeGEM1TopY,      GEM1 Y-Cluster strip count
#CLUST,   ClustSizeGEM2TopX,      GEM2 X-Cluster strip count
#CLUST,   ClustSizeGEM2TopY,      GEM2 Y-Cluster strip count
#CLUST,   ClustSizeGEM3TopX,      GEM2 Y-Cluster strip count
#CLUST,   ClustSizeGEM4TopX,      GEM2 Y-Cluster strip count

#CLUSTSTAT,   ClustStatGEM1TopX,      GEM1 X-Cluster strip count
#CLUSTSTAT,   ClustStatGEM1TopY,      GEM1 Y-Cluster strip count
#CLUSTSTAT,   ClustStatGEM2TopX,      GEM2 X-Cluster strip count
#CLUSTSTAT,   ClustStatGEM2TopY,      GEM2 Y-Cluster strip count
#CLUSTSTAT,   ClustStatGEM3TopX,      GEM2 Y-Cluster strip count
#CLUSTSTAT,   ClustStatGEM4TopX,      GEM2 Y-Cluster strip count

#CLUSTMULT,   ClustMultGEM1TopX,      GEM1 X-Cluster strip count
#CLUSTMULT,   ClustMultGEM1TopY,      GEM1 Y-Cluster strip count
#CLUSTMULT,   ClustMultGEM2TopX,      GEM2 X-Cluster strip count
```

```
#CLUSTMULT,   ClustMultGEM2TopY,      GEM2 Y-Cluster strip count
#CLUSTMULT,   ClustMultGEM3TopX,      GEM2 Y-Cluster strip count
#CLUSTMULT,   ClustMultGEM4TopX,      GEM2 Y-Cluster strip count

#SPECTRUM,   SpectrumGEM1TopX,        GEM1 X Pulse Height X
#SPECTRUM,   SpectrumGEM1TopY,        GEM1 Y Pulse Height X
#SPECTRUM,   SpectrumGEM2TopX,        GEM2 X Pulse Height X
#SPECTRUM,   SpectrumGEM2TopY,        GEM2 Y Pulse Height X
#SPECTRUM,   SpectrumGEM3TopX,        GEM2 Y Pulse Height X
#SPECTRUM,   SpectrumGEM4TopX,        GEM2 Y Pulse Height X
```

Appendix D

ROOT Script used for POCA Plots and Analysis

This root script has been written to create the 2D sliced histograms that display the POCA data in a voxelized form. Contributions to this code have been made by K. Gnanvo, J. B. Locke, and the author. Post-processing for reduction of noise has recently been added, as well as the code to calculate the simple scattering density inside a volume. Various bug fixes have also been made. Two histograms are produced per slice, one containing the average scattering angle within each voxel, and a second containing the number of POCA points per voxel. Both are produced with a black and a white background. A configuration file with the name Poca2DConfig.txt must reside in the same directory where the Poca2DPlots.C script is run. The configuration file is explained below. In order to run the code, navigate to the directory, open a session of ROOT with the command "root –l –b", and enter ".x Poca2DPlots.C" at the prompt.

poca2DConfig.txt

```
##################################################
# Input file is first declared, do not include the trailing ".txt"
##################################################
inputFile:        leadshieldPOCA

##################################################
# Select the slice plane and the size of the volume
##################################################
slicePlane:       YZ
sizeX:             40
sizeY:             250
sizeZ:             250

####################################
# Set the depth of the slices
####################################
slice:            40

##################################################
# Add some offset to the starting central location
##################################################
offset:           0
```

```
####################################################
# Set bin size. Voxel size is binSize x binSize x slice
####################################################
binSize:          2

####################################################
# Set min and max value for the color scale on the Average scattering
angle plots
####################################################
minValue:         2
maxValue:         8

####################################################
# Set values for cuts that occur after the histograms are filled
####################################################
muonsCut:              2
minAngleCut:           2
maxAngleCut:          90
neighborMuonsCut: 4
```

Poca2DPlots.C

```
void Poca2DPlots(TString configurationFile = "Poca2DConfig.txt") {

  getStyle() ;
  getStyle() ;

  //To do: These variables could be made global to allow easier
extensibility
  TString inputFile, slicePlane;
  Int_t sizeX, sizeY, sizeZ, binSize, slice, offset, muonsCut,
neighborMuonsCut;
  Float_t minValue, maxValue, minAngleCut, maxAngleCut ;

  TString configFile = gSystem->UnixPathName(gInterpreter-
>GetCurrentMacroName());
  configFile.ReplaceAll("Poca2DPlots.C","");
  configFile.ReplaceAll("/./","/");
  configFile.Append(configurationFile) ;

  TEnv env(configFile);

  inputFile        = env.GetValue("inputFile", "foo");
  slicePlane       = env.GetValue("slicePlane", "foo");
  sizeX            = env.GetValue("sizeX", 300);
  sizeY            = env.GetValue("sizeY", 300);
  sizeZ            = env.GetValue("sizeZ", 300);
  slice            = env.GetValue("slice", 30);
  offset           = env.GetValue("offset", 0);
  binSize          = env.GetValue("binSize", 2);
  minValue         = env.GetValue("minValue", 2.0);
  maxValue         = env.GetValue("maxValue", 8.0);
  muonsCut         = env.GetValue("muonsCut", 0);
```

```
  minAngleCut      = env.GetValue("minAngleCut", 0.0);
  maxAngleCut      = env.GetValue("maxAngleCut", 90.0);
  neighborMuonsCut = env.GetValue("neighborMuonsCut", 0);

  printf("inputFile        %s\n",inputFile.Data());
  printf("slicePlane       %s\n",slicePlane.Data());
  printf("sizeX            %i\n",sizeX);
  printf("sizeY            %i\n",sizeY);
  printf("sizeZ            %i\n",sizeZ);
  printf("slice            %i\n",slice);
  printf("offset           %i\n",offset);
  printf("binSize          %i\n",binSize);
  printf("minValue         %f\n",minValue);
  printf("maxValue         %f\n",maxValue);
  printf("muonsCut         %i\n",muonsCut);
  printf("minAngleCut      %f\n",minAngleCut);
  printf("maxAngleCut      %f\n",maxAngleCut);
  printf("neighborMuonsCut %d\n",neighborMuonsCut);

  // Results are also stored in a ROOT file
  TFile *f = new TFile(Form("%s.root",inputFile.Data()),"RECREATE");

  Int_t nbinx = sizeX/binSize;
  Int_t nbiny = sizeY/binSize;
  Int_t nbinz = sizeZ/slice;

  // This for loop may be used to automate the process of producing
multiple plots
  // To Do: Enable this functionality via the configuration file
  //for (int i = 0; i < 60; i ++){
  //offset = (i * 5) - 150;
  //for (int i = 0; i < 46; i ++){
  //neighborMuonsCut = i;

get2DSlicedHists(inputFile,slicePlane,sizeX,sizeY,sizeZ,slice,offset,binSi
ze,minValue,maxValue,muonsCut,minAngleCut, maxAngleCut, neighborMuonsCut)
;
  //}

  f->Write();
}

void getStyle() {

  gStyle->SetOptStat(0);
  gROOT->SetStyle("Plain");
/*
  gStyle->SetLabelFont(62,"xyz");
  gStyle->SetLabelSize(0.04,"xyz");
  gStyle->SetTitleBorderSize(0) ;
  gStyle->SetTitleSize(0.05,"xyz");
  gStyle->SetTitleOffset(4.5,"xyz");
  gStyle->SetPadTickX(1) ;
  gStyle->SetPadTickY(1) ;
  gStyle->SetMarkerStyle(21) ;
  gStyle->SetMarkerSize(.75) ;
*/
  // Style.
```

```
  const int timesNewRoman = 132;
  gStyle->SetTitleFont(timesNewRoman, "");
  gStyle->SetTitleFont(timesNewRoman, "XYZ");
  gStyle->SetLabelFont(timesNewRoman, "XYZ");
  gStyle->SetStatFont(timesNewRoman);
  gStyle->SetTextFont(timesNewRoman);

  const Int_t NRGBs = 5;
  const Int_t NCont = 255;
  Double_t stops[NRGBs] = { 0.00, 0.34, 0.61, 0.84, 1.00 };
  Double_t red[NRGBs]   = { 0.00, 0.00, 0.87, 1.00, 0.51 };
  Double_t green[NRGBs] = { 0.00, 0.81, 1.00, 0.20, 0.00 };
  Double_t blue[NRGBs]  = { 0.51, 1.00, 0.12, 0.00, 0.00 };
  TColor::CreateGradientColorTable(NRGBs, stops, red, green, blue, NCont);
  gStyle->SetNumberContours(NCont);
}

void get2DSlicedHists(TString sim1, TString slicePlane, Int_t sizeX, Int_t
sizeY, Int_t sizeZ, Float_t slice, Int_t offset, Float_t binSize, Float_t
minValue, Float_t maxValue,  Int_t nbMuonCut, Float_t cut, Float_t
maxAngleCut, Int_t neighborMuonsCut) {

  TString dir1 = gSystem->UnixPathName(gInterpreter-
>GetCurrentMacroName());
  dir1.ReplaceAll("Poca2DPlots.C","");
  dir1.ReplaceAll("/./","/");
  dir1.Append(sim1) ;

  ifstream in1 ;
  in1.open(Form("%s.txt",dir1.Data()));
  if (!in1) {
    printf("Problem opening file name %s.txt\n", dir1.Data());
    return;
  }

  Int_t size1 = sizeX ;
  Int_t size2 = sizeY ;
  Int_t size3 = sizeZ ;

  Int_t nbin1 = sizeX/binSize ;
  Int_t nbin2 = sizeY/binSize ;
  Int_t nbin3 = sizeZ/slice ;

  TString histTitle1 = "h1xy2d" ;

  if(slicePlane == "XZ") {
    nbin1 = sizeX/binSize ;
    nbin2 = sizeZ/binSize ;
    nbin3 = sizeY/slice ;

    size1 = sizeX ;
    size2 = sizeZ ;
    size3 = sizeY ;

    histTitle1 = "h1xz2d" ;
  }

  if(slicePlane == "YZ") {
```

```
    nbin1 = sizeY/binSize ;
    nbin2 = sizeZ/binSize ;
    nbin3 = sizeX/slice ;

    size1 = sizeY ;
    size2 = sizeZ ;
    size3 = sizeX ;

    histTitle1 = "h1yz2d" ;
  }

  //printf("nbin1      %d\n", nbin1 );
  //printf("nbin2      %d\n", nbin2 );
  //printf("nbin3      %d\n", nbin3 );

  Float_t x, y, z, meanAngle;
  Int_t nlines = 0;

  TH2F *h2d1[200];
  TH2F *h2d1N[200];

  for(int k=0; k<200; k++) {
    TString title2d1Hist  = Form("%s%s",histTitle1.Data(),sim1.Data()) ;
    h2d1[k] = new
TH2F(Form("%s%d",title2d1Hist.Data(),k),Form("%s%d",title2d1Hist.Data(),
k), nbin1,-size1/2,size1/2,nbin2,-size2/2,size2/2);
    h2d1N[k] = new TH2F(Form("1%d",k),Form("1%d",k), nbin1,-
size1/2,size1/2,nbin2,-size2/2,size2/2);
  }

  while (1) {
    in1 >> x >> y >> z >> meanAngle ;
    if (!in1.good()) break;
    if(slicePlane == "XY") {
      if((x>=-sizeX/2)&&(x<=sizeX/2)&&(y>=-
sizeY/2)&&(y<=sizeY/2)&&(z>=(offset-sizeZ/2))&&(z<=(offset+sizeZ/2))) {
       Int_t j = (Int_t) ((size3/2 + offset - z)/slice) + 1  ;
      Int_t k = nbin3 - j ;
      if(k<0) k=0 ;
      h2d1[k]->Fill(x,y,meanAngle) ;
      h2d1N[k]->Fill(x,y) ;
      }
    }

    if(slicePlane == "XZ") {
      if((x>=-sizeX/2)&&(x<=sizeX/2)&&(y>=(offset-
sizeY/2))&&(y<=(offset+sizeY/2))&&(z>=-sizeZ/2)&&(z<=sizeZ/2)) {
      Int_t j = (Int_t) ((size3/2 + offset - y)/slice) + 1 ;
      Int_t k = nbin3 - j ;
      if(k<0) k=0 ;
      h2d1[k]->Fill(x,z,meanAngle) ;
      h2d1N[k]->Fill(x,z) ;
      }
    }

    if(slicePlane == "YZ") {
      if((x>=(offset-sizeX/2))&&(x<=(offset+sizeX/2))&&(y>=-
sizeY/2)&&(y<=sizeY/2)&&(z>=-sizeZ/2)&&(z<=sizeZ/2)) {
```

```
      Int_t j = (Int_t) ((size3/2 + offset - x)/slice) + 1 ;
      Int_t k = nbin3 - j ;
      if(k<0) k=0 ;
      h2d1[k]->Fill(y,z,meanAngle) ;
      h2d1N[k]->Fill(y,z) ;
     }
    }
    nlines++;
  }

  for(int k=0; k<nbin3; k++) {
    Int_t voxel = -size3/2 + offset + (k+0.5)*slice ;
    Float_t maxVal = 1.0 ;
    maxVal = maxValue ;
    h2d1[k]->Divide(h2d1N[k]) ;

    for(int i=0; i<=nbin1; i++) {
      for(int j=0; j<=nbin2; j++) {
       Int_t nbMuons1 = h2d1N[k]->GetBinContent(i,j) ;
       Float_t param1 = h2d1[k]->GetBinContent(i,j) ;
       if((nbMuons1 < nbMuonCut) || (param1 < cut) || (param1 >
maxAngleCut) ) {
         h2d1[k]->SetBinContent(i,j,0) ;
         h2d1N[k]->SetBinContent(i,j,0) ;
       }
      }
    }

//Post-processing to remove noise

   for(int i=1; i<=(nbin1-1); i++) {
      for(int j=1; j<=(nbin2-1); j++) {
        Float_t CenterBin = h2d1[k]->GetBinContent(i,j) ;
        Float_t TL = h2d1[k]->GetBinContent(i-1,j+1) ;
        Float_t T = h2d1[k]->GetBinContent(i,j+1) ;
        Float_t TR = h2d1[k]->GetBinContent(i+1,j+1) ;
        Float_t R = h2d1[k]->GetBinContent(i+1,j) ;
        Float_t BR = h2d1[k]->GetBinContent(i+1,j-1) ;
       Float_t B = h2d1[k]->GetBinContent(i,j-1) ;
        Float_t BL = h2d1[k]->GetBinContent(i-1,j-1) ;
        Float_t L = h2d1[k]->GetBinContent(i-1,j) ;

        Float_t nCenterBin = h2d1N[k]->GetBinContent(i,j) ;
        Float_t nTL = h2d1N[k]->GetBinContent(i-1,j+1) ;
        Float_t nT = h2d1N[k]->GetBinContent(i,j+1) ;
        Float_t nTR = h2d1N[k]->GetBinContent(i+1,j+1) ;
        Float_t nR = h2d1N[k]->GetBinContent(i+1,j) ;
        Float_t nBR = h2d1N[k]->GetBinContent(i+1,j-1) ;
       Float_t nB = h2d1N[k]->GetBinContent(i,j-1) ;
        Float_t nBL = h2d1N[k]->GetBinContent(i-1,j-1) ;
        Float_t nL = h2d1N[k]->GetBinContent(i-1,j) ;

//Remove all bins that are not surrounded by a total mean scattering angle
of less than TotalAngleCut

/*
        Float_t TotalAngleCut = 1E-4;
        if((T+B+L+R+TL+TR+BL+BR) < TotalAngleCut) {
```

```
         h2d1[k]->SetBinContent(i,j,0) ;
         h2d1N[k]->SetBinContent(i,j,0) ;
       }
*/

//Remove all voxels with less than neighborMuonsCut muons in surrounding
voxels
        if (neighborMuonsCut != 0){
          if(((Int_t)(nT+nB+nL+nR+nTL+nTR+nBL+nBR)) <= neighborMuonsCut )
{
            h2d1[k]->SetBinContent(i,j,0) ;
           h2d1N[k]->SetBinContent(i,j,0) ;
         }
        }

      }
    }

//End Filtering

////////////////////////////////////////////////////////////////////////
////
// The following sections have been hard-coded!
// the bin ranges analyzed must be set below
// To Do: Should be generalized to be set ranges by coordinates in config
file
////////////////////////////////////////////////////////////////////////
////

// Begin Spatial resolution Analysis for two block scenario
/*
   TH1F * averageScattering = new TH1F("Average Scattering Angle",
"Average Scattering Angle", 50, -50, 50);
   TH1F * scatteringDensityDistribution = new TH1F("Scattering Density
Distribution", "Scattering Density", 200, 0, 160);
   TH1F * scatteringDensity = new TH1F("Scattering Density", "Simple
Scattering Density 8 mm Spacing", 50, -50, 50);
   TH1F * totalPOCA = new TH1F("Total POCA Points", "Total POCA Points",
50, -50, 50);
   TH1F * totalPOCADistribution = new TH1F("Total POCA", "Total POCA",
200, 0, 70);
   TH1F * averageVoxel = new TH1F("Avg. Voxel Value", "Avg. Voxel Value",
50, -50, 50);
   TH1F * averageVoxelDistribution = new TH1F("Avg. Voxel", "Avg. Voxel",
100, 0, 25);
   Float_t totalAngle = 0, totalMuons =0, count = 0, totalVoxelAngle = 0;
   Int_t bin = 1;
    for(int yy=(int)(nbin2/2-25); yy<=(int)(nbin2/2 +25); yy++) {
      totalAngle = 0; totalMuons =0; count = 0; totalVoxelAngle = 0;
      for(int xx=(int)(nbin1/2-9); xx<=(int)(nbin1/2+11); xx++) {
      totalAngle += (h2d1N[k]->GetBinContent(xx,yy))*(h2d1[k]-
>GetBinContent(xx,yy));
      totalMuons += (h2d1N[k]->GetBinContent(xx,yy));
      totalVoxelAngle += h2d1[k]->GetBinContent(xx,yy);
      count++;
      h2d1[k]->SetBinContent(xx,yy, ((Float_t) bin)/6);
      }
      averageScattering->SetBinContent(bin, totalAngle/totalMuons);
```

```
      totalPOCA->SetBinContent(bin, totalMuons);
      scatteringDensity->SetBinContent(bin, (totalAngle* 1000
*(100000/(Float_t) nlines) / (count * binSize * binSize * slice)));
      averageVoxel->SetBinContent(bin, totalVoxelAngle/count);
      scatteringDensityDistribution->Fill(totalAngle*1000
*(100000/(Float_t) nlines)/ (count * binSize * binSize * slice));
      totalPOCADistribution->Fill(totalMuons);
      averageVoxelDistribution->Fill(totalVoxelAngle/count);
      bin++;
    }
    TCanvas * ccc = new TCanvas("ga", "ga", 800, 800);
    TCanvas * ccd = new TCanvas("gg", "gg", 1000, 425);

    ccc->cd();
    gStyle->SetOptStat(1111);
    averageScattering->Draw();
    gStyle->SetOptStat(1111);

    ccc->SaveAs("averageScattering"+sim1+".png");
    ccc->SaveAs("averageScattering"+sim1+".root");

    ccd->cd();
    totalPOCA->Draw();
    TPaveText *t2 = new TPaveText(0.65, 0.85, 0.99 , 0.99, "brNDC");
    t2->AddText(Form("Average = %.1f", totalPOCADistribution->GetMean()));
    t2->AddText(Form("RMS = %.1f", totalPOCADistribution->GetRMS()));
    t2->Draw();
    ccd->SaveAs("TotalPOCA"+sim1+".png");
    ccd->SaveAs("TotalPOCA"+sim1+".root");
    ccd->Clear();


    scatteringDensity->SetYTitle("Simple scattering density
[deg/cm^{3}]");
    scatteringDensity->SetXTitle("Y Position [mm]");
    scatteringDensity->GetXaxis()->SetTitleColor(1);
    scatteringDensity->GetYaxis()->SetTitleColor(1);
    scatteringDensity->GetXaxis()->CenterTitle();
    scatteringDensity->SetTitleOffset(0.80, "Y");
    scatteringDensity->Draw();
    TPaveText *t1 = new TPaveText(0.75, 0.85, 0.99 , 0.99, "brNDC");
    t1->AddText(Form("Average = %.1f deg/cm^{3}",
scatteringDensityDistribution->GetMean()));
    t1->AddText(Form("RMS = %.1f deg/cm^{3}",
scatteringDensityDistribution->GetRMS()));
    t1->SetFillColor(0);
    t1->Draw();
    ccd->SaveAs("ScatteringDensity"+sim1+".png");
    ccd->SaveAs("ScatteringDensity"+sim1+".root");
    ccd->Clear();

    averageVoxel->Draw();
    TPaveText *t3 = new TPaveText(0.65, 0.85, 0.99 , 0.99, "brNDC");
    t3->AddText(Form("Average = %.1f", averageVoxelDistribution-
>GetMean()));
    t3->AddText(Form("RMS = %.1f", averageVoxelDistribution->GetRMS()));
    t3->Draw();
    ccd->SaveAs("AvgVoxel"+sim1+".png");
```

```
    ccd->SaveAs("AvgVoxel"+sim1+".root");
    delete ccc;
    delete ccd;
    gStyle->SetOptStat(0);

*/
//End Spatial Resolution Analysis

//Begin Scattering Density calculation for 5-target scenario
/*

   Float_t UDen=0, FeDen=0, WDen=0, SnDen=0, PbDen=0, UAvg =0, FeAvg=0,
WAvg =0, PbAvg = 0, SnAvg =0, UTotAvg =0, FeTotAvg=0, WTotAvg =0, PbTotAvg
= 0, SnTotAvg =0;

   Float_t totalAngle = 0, totalMuons =0, totalAverageAngle = 0;
   Int_t count = 0;
//    for(int yy=(int)(nbin2/2-35); yy<=(int)(nbin2/2 -15); yy++) {
//      for(int xx=(int)(nbin1/2-35); xx<=(int)(nbin1/2-15); xx++) {
    for(int yy=(int)(nbin2/2-42); yy<=(int)(nbin2/2 -18); yy++) {
      for(int xx=(int)(nbin1/2-42); xx<=(int)(nbin1/2-18); xx++) {
      totalAngle += (h2d1N[k]->GetBinContent(xx,yy))*(h2d1[k]-
>GetBinContent(xx,yy));
      totalMuons += (h2d1N[k]->GetBinContent(xx,yy));
      totalAverageAngle += h2d1[k]->GetBinContent(xx,yy);
      h2d1[k]->SetBinContent(xx,yy, 8);
      count++;
      }
    }
    SnDen = totalAngle / (count * binSize * binSize * slice);
    SnAvg = totalAverageAngle / count;
    SnTotAvg = totalAngle / totalMuons;

    count = 0; totalAngle = 0; totalMuons = 0; totalAverageAngle = 0;
//     for(int yy=(int)(nbin2/2+15); yy<=(int)(nbin2/2 + 35); yy++) {
//      for(int xx=(int)(nbin1/2-35); xx<=(int)(nbin1/2-15); xx++) {
    for(int yy=(int)(nbin2/2+18); yy<=(int)(nbin2/2 + 42); yy++) {
      for(int xx=(int)(nbin1/2-42); xx<=(int)(nbin1/2-18); xx++) {
      totalAngle += (h2d1N[k]->GetBinContent(xx,yy))*(h2d1[k]-
>GetBinContent(xx,yy));
      totalMuons += (h2d1N[k]->GetBinContent(xx,yy));
      totalAverageAngle += h2d1[k]->GetBinContent(xx,yy);
      h2d1[k]->SetBinContent(xx,yy, 7);
      count++;
      }
    }
    PbDen = totalAngle / (count * binSize * binSize * slice);
    PbAvg = totalAverageAngle / count;
    PbTotAvg = totalAngle / totalMuons;

    count = 0; totalAngle = 0; totalMuons = 0;totalAverageAngle = 0;
//     for(int yy=(int)(nbin2/2+15); yy<=(int)(nbin2/2 +35); yy++) {
//      for(int xx=(int)(nbin1/2+15); xx<=(int)(nbin1/2+35); xx++) {
     for(int yy=(int)(nbin2/2+18); yy<=(int)(nbin2/2 +42); yy++) {
      for(int xx=(int)(nbin1/2+18); xx<=(int)(nbin1/2+42); xx++) {
      totalAngle += (h2d1N[k]->GetBinContent(xx,yy))*(h2d1[k]-
>GetBinContent(xx,yy));
      totalMuons += (h2d1N[k]->GetBinContent(xx,yy));
```

```
      totalAverageAngle += h2d1[k]->GetBinContent(xx,yy);
      h2d1[k]->SetBinContent(xx,yy, 6);
      count++;
      }
    }
    WDen = totalAngle / (count * binSize * binSize * slice);
    WAvg = totalAverageAngle / count;
    WTotAvg = totalAngle / totalMuons;

    count = 0; totalAngle = 0; totalMuons = 0;totalAverageAngle = 0;
//     for(int yy=(int)(nbin2/2-35); yy<=(int)(nbin2/2 -15); yy++) {
//      for(int xx=(int)(nbin1/2+15); xx<=(int)(nbin1/2+35); xx++) {
     for(int yy=(int)(nbin2/2-42); yy<=(int)(nbin2/2 -18); yy++) {
      for(int xx=(int)(nbin1/2+18); xx<=(int)(nbin1/2+42); xx++) {
      totalAngle += (h2d1N[k]->GetBinContent(xx,yy))*(h2d1[k]-
>GetBinContent(xx,yy));
      totalMuons += (h2d1N[k]->GetBinContent(xx,yy));
      totalAverageAngle += h2d1[k]->GetBinContent(xx,yy);
      h2d1[k]->SetBinContent(xx,yy, 5);
      count++;
      }
    }
    FeDen = totalAngle / (count * binSize * binSize * slice);
    FeAvg = totalAverageAngle / count;
    FeTotAvg = totalAngle / totalMuons;

    count = 0; totalAngle = 0; totalMuons = 0;totalAverageAngle = 0;
//     for(int yy=(int)(nbin2/2-10); yy<=(int)(nbin2/2 +10); yy++) {
//      for(int xx=(int)(nbin1/2-10); xx<=(int)(nbin1/2+10); xx++) {
     for(int yy=(int)(nbin2/2-12); yy<=(int)(nbin2/2 +12); yy++) {
      for(int xx=(int)(nbin1/2-12); xx<=(int)(nbin1/2+12); xx++) {
      totalAngle += (h2d1N[k]->GetBinContent(xx,yy))*(h2d1[k]-
>GetBinContent(xx,yy));
      totalMuons += (h2d1N[k]->GetBinContent(xx,yy));
      totalAverageAngle += h2d1[k]->GetBinContent(xx,yy);
      h2d1[k]->SetBinContent(xx,yy, 4);
      count++;
      }
    }
    UDen = totalAngle / (count * binSize * binSize * slice);
    UAvg = totalAverageAngle / count;
    UTotAvg = totalAngle / totalMuons;

printf("The Scattering density is defined as the sum of all scattering\n
angles for all poca points in the volume considered divided by the volume
considered\n\n");
printf("The scattering Density for Fe is %.1f +/- %.1f deg/cm^3\n",
FeDen*1000*(100000/(Float_t) nlines),
(.4258/FeTotAvg)*FeDen*1000*(100000/(Float_t) nlines));
printf("The scattering Density for Sn is %.1f +/- %.1f deg/cm^3\n",
SnDen*1000*(100000/(Float_t) nlines),
(.4258/SnTotAvg)*SnDen*1000*(100000/(Float_t) nlines));
printf("The scattering Density for Pb is %.1f +/- %.1f deg/cm^3\n",
PbDen*1000*(100000/(Float_t) nlines),
(.4258/PbTotAvg)*PbDen*1000*(100000/(Float_t) nlines));
printf("The scattering Density for W is %.1f +/- %.1f deg/cm^3\n",
WDen*1000*(100000/(Float_t) nlines),
(.4258/WTotAvg)*WDen*1000*(100000/(Float_t) nlines));
```

```
printf("The scattering Density for U is %.1f +/- %.1f deg/cm^3\n\n",
UDen*1000*(100000/(Float_t) nlines),
(.4258/UTotAvg)*UDen*1000*(100000/(Float_t) nlines));
printf("The Average Scattering is defined as the average voxel value\n in
the volume considered (including zeros)\n\n");
printf("The Average Scattering per voxel for Fe is %f deg\n", FeAvg);
printf("The Average Scattering per voxel for Sn is %f deg\n", SnAvg);
printf("The Average Scattering per voxel for Pb is %f deg\n", PbAvg);
printf("The Average Scattering per voxel for W is %f deg\n", WAvg);
printf("The Average Scattering per voxel for U is %f deg\n\n", UAvg);
printf("The Total average scattering is defined as the average\n
scattering angle of all poca points in the volume considered\n\n");
printf("The Total Average Scattering for Fe is %f deg\n", FeTotAvg);
printf("The Total Average Scattering for Sn is %f deg\n", SnTotAvg);
printf("The Total Average Scattering for Pb is %f deg\n", PbTotAvg);
printf("The Total Average Scattering for W is %f deg\n", WTotAvg);
printf("The Total Average Scattering for U is %f deg\n", UTotAvg);

*/
//End Scattering Density Calculation

    save2dSlicedHistsStats(h2d1N[k],sim1,slicePlane,voxel,k,7,0, slice,
nbMuonCut, neighborMuonsCut) ;
    save2dSlicedHists(h2d1[k],sim1,slicePlane,voxel,k,maxVal,minValue,
slice, nbMuonCut, neighborMuonsCut) ;

  }
  in1.close() ;
  TH2F *h2d1[200];
  TH2F *h2d1N[200];

  for(int k=0; k<200; k++){
    delete h2d1[k]; delete h2d1N[k];
  }

}

void save2dSlicedHistsStats(TH2F *h2d, TString input, TString slicePlane,
Int_t sliceCut, Int_t number, Float_t maxVal, Float_t minVal, Float_t
slice, Int_t nbMuonCut, Int_t neighborMuonsCut) {

  TString title       =
image2DFileName(input,slicePlane,sliceCut,number,"", slice, nbMuonCut, 0)
;
  TString pictureName = image2DFileName(input,slicePlane,sliceCut,number,
".png", slice, nbMuonCut, neighborMuonsCut)  ;
  Int_t canvasSize1 = 600 ;
  Int_t canvasSize2 = 600 ;

  if(slicePlane == "XY") {
    h2d->GetXaxis()->SetTitle("X [mm]");
    h2d->GetYaxis()->SetTitle("Y [mm]");
  }
  if(slicePlane == "XZ") {
    h2d->GetXaxis()->SetTitle("X [mm]");
    h2d->GetYaxis()->SetTitle("Z [mm]");
  }
  if(slicePlane == "YZ") {
```

```
    h2d->GetXaxis()->SetTitle("Y [mm]");
    h2d->GetYaxis()->SetTitle("Z [mm]");
  }

  TCanvas * cWhite = new TCanvas ("cWhite","2DWhite",0,0,canvasSize1,
canvasSize2);
  cWhite->cd();
  cWhite->SetFillColor(0);
  cWhite->SetBorderMode(0);

  h2d->Draw("colz");
  h2d->UseCurrentStyle();

  h2d->GetXaxis()->SetTitleOffset(0.85);
  h2d->GetXaxis()->SetTitleSize(0.03);
  h2d->GetXaxis()->SetLabelOffset(0.001);
  h2d->GetXaxis()->SetLabelSize(0.03);
  h2d->GetXaxis()->CenterTitle();

  h2d->GetYaxis()->SetTitleOffset(0.85);
  h2d->GetYaxis()->SetTitleSize(0.03);
  h2d->GetYaxis()->SetLabelOffset(0.001);
  h2d->GetYaxis()->SetLabelSize(0.03);
  h2d->GetYaxis()->CenterTitle();

  h2d->SetMinimum(minVal);
  h2d->SetMaximum(maxVal);
  h2d->SetTitle("stats"+title);

  TPaveText *t = new TPaveText(0.65, 0.92, 0.99 , 0.975, "brNDC");
  t->AddText("Poca Points Per Voxel");
  t->Draw();
  t->SetTextFont(62);
  t->SetTextColor(1);
  t->SetFillColor(0);
  t->SetLineColor(1);

  gStyle->SetTitleFillColor(0);
  gStyle->SetTitleTextColor(1);
  gPad->Update();

  TPaletteAxis *palette = (TPaletteAxis*)h2d->GetListOfFunctions()-
>FindObject("palette");
  palette->SetLabelColor(1);

  cWhite->Modified();
  cWhite->Update();

  cWhite->SaveAs("Plots/statsWhite"+pictureName);
  cWhite->Close();

  TCanvas * cBlack = new TCanvas("cBlack","2DBlack",0,0,
canvasSize1,canvasSize2);
  cBlack->cd();

  cBlack->SetFillColor(1);
  h2d->Draw("colz");
  h2d->UseCurrentStyle();
```

```
  h2d->GetXaxis()->SetTitleOffset(0.85);
  h2d->GetXaxis()->SetTitleSize(0.03);
  h2d->GetXaxis()->SetLabelOffset(0.001);
  h2d->GetXaxis()->SetLabelSize(0.03);
  h2d->GetXaxis()->CenterTitle();
  h2d->GetXaxis()->SetLabelColor(0);
  h2d->GetXaxis()->SetAxisColor(0);
  h2d->GetXaxis()->SetTitleColor(0);

  h2d->GetYaxis()->SetTitleOffset(0.85);
  h2d->GetYaxis()->SetTitleSize(0.03);
  h2d->GetYaxis()->SetLabelOffset(0.001);
  h2d->GetYaxis()->SetLabelSize(0.03);
  h2d->GetYaxis()->CenterTitle();
  h2d->GetYaxis()->SetLabelColor(0);
  h2d->GetYaxis()->SetAxisColor(0);
  h2d->GetYaxis()->SetTitleColor(0);

  h2d->SetMinimum(minVal);
  h2d->SetMaximum(maxVal);
  h2d->SetTitle("stats"+title);

  t->Draw();
  t->SetTextColor(0);
  t->SetFillColor(1);
  t->SetLineColor(0);

  gStyle->SetTitleFillColor(1);
  gStyle->SetTitleTextColor(0);

  palette->SetLabelColor(0);

  cBlack->Modified();
  cBlack->Update();

  cBlack->SaveAs("Plots/statsBlack"+pictureName);
  cBlack->Close();
  delete cWhite;
  delete cBlack;
}


void save2dSlicedHists(TH2F *h2d, TString input, TString slicePlane, Int_t
sliceCut, Int_t number, Float_t maxVal, Float_t minVal, Float_t slice,
Int_t nbMuonCut, Int_t neighborMuonsCut) {

  TString title       = image2DFileName(input,slicePlane,sliceCut,number,
"", slice, nbMuonCut, 0)  ;
  TString pictureName = image2DFileName(input,slicePlane,sliceCut,number,
".png", slice, nbMuonCut, neighborMuonsCut)  ;
  Int_t canvasSize1 = 600 ;
  Int_t canvasSize2 = 600 ;

  if(slicePlane == "XY") {
    h2d->GetXaxis()->SetTitle("X [mm]");
    h2d->GetYaxis()->SetTitle("Y [mm]");
  }
```

```
  if(slicePlane == "XZ") {
    h2d->GetXaxis()->SetTitle("X [mm]");
    h2d->GetYaxis()->SetTitle("Z [mm]");
  }
  if(slicePlane == "YZ") {
    h2d->GetXaxis()->SetTitle("Y [mm]");
    h2d->GetYaxis()->SetTitle("Z [mm]");
  }

  TCanvas * cWhite = new TCanvas("cWhite","2DWhite",0,0,
canvasSize1,canvasSize2);
  cWhite->cd();
  cWhite->SetFillColor(0);
  cWhite->SetBorderMode(0);

  h2d->Draw("colz");
  h2d->UseCurrentStyle();

  h2d->GetXaxis()->SetTitleOffset(0.85);
  h2d->GetXaxis()->SetTitleSize(0.03);
  h2d->GetXaxis()->SetLabelOffset(0.001);
  h2d->GetXaxis()->SetLabelSize(0.03);
  h2d->GetXaxis()->CenterTitle();

  h2d->GetYaxis()->SetTitleOffset(0.85);
  h2d->GetYaxis()->SetTitleSize(0.03);
  h2d->GetYaxis()->SetLabelOffset(0.001);
  h2d->GetYaxis()->SetLabelSize(0.03);
  h2d->GetYaxis()->CenterTitle();

  h2d->SetMinimum(minVal);
  h2d->SetMaximum(maxVal);
  h2d->SetTitle(title);

  TPaveText *t = new TPaveText(0.65, 0.92, 0.99 , 0.975, "brNDC");
  t->AddText("Mean Scattering Angle [deg]");
  t->Draw();
  t->SetTextFont(62);
  t->SetTextColor(1);
  t->SetFillColor(0);
  t->SetLineColor(1);

  gStyle->SetTitleFillColor(0);
  gStyle->SetTitleTextColor(1);
  gPad->Update();

  TPaletteAxis *palette = (TPaletteAxis*)h2d->GetListOfFunctions()-
>FindObject("palette");
  palette->SetLabelColor(1);

  cWhite->Modified();
  cWhite->Update();

  cWhite->SaveAs("Plots/White"+pictureName);
  cWhite->Close();

  TCanvas * cBlack = new TCanvas("cBlack","2DBlack",0,0,canvasSize1,
canvasSize2);
```

```
  cBlack->cd();

  cBlack->SetFillColor(1);
  h2d->Draw("colz");
  h2d->UseCurrentStyle();

  h2d->GetXaxis()->SetTitleOffset(0.85);
  h2d->GetXaxis()->SetTitleSize(0.03);
  h2d->GetXaxis()->SetLabelOffset(0.001);
  h2d->GetXaxis()->SetLabelSize(0.03);
  h2d->GetXaxis()->CenterTitle();
  h2d->GetXaxis()->SetLabelColor(0);
  h2d->GetXaxis()->SetAxisColor(0);
  h2d->GetXaxis()->SetTitleColor(0);

  h2d->GetYaxis()->SetTitleOffset(0.85);
  h2d->GetYaxis()->SetTitleSize(0.03);
  h2d->GetYaxis()->SetLabelOffset(0.001);
  h2d->GetYaxis()->SetLabelSize(0.03);
  h2d->GetYaxis()->CenterTitle();
  h2d->GetYaxis()->SetLabelColor(0);
  h2d->GetYaxis()->SetAxisColor(0);
  h2d->GetYaxis()->SetTitleColor(0);

  h2d->SetMinimum(minVal);
  h2d->SetMaximum(maxVal);
  h2d->SetTitle(title);

  t->Draw();
  t->SetTextColor(0);
  t->SetFillColor(1);
  t->SetLineColor(0);

  gStyle->SetTitleFillColor(1);
  gStyle->SetTitleTextColor(0);

  palette->SetLabelColor(0);

  cBlack->Modified();
  cBlack->Update();

  cBlack->SaveAs("Plots/Black"+pictureName);
  cBlack->Close();
  delete cWhite;
  delete cBlack;
}


TString image2DFileName(TString input, TString slicePlane, Int_t sliceCut,
Int_t number, TString extension, Float_t slice, Int_t nbMuonCut, Int_t
neighborMuonsCut) {

  TString sliceAxis ;
  if(slicePlane == "XY") sliceAxis ="Z" ;
  if(slicePlane == "XZ") sliceAxis ="Y" ;
  if(slicePlane == "YZ") sliceAxis ="X" ;
```

```
  TString output = "";
  if(extension == ".png") output = Form("%s",input.Data()) ;
  TString slicePlane= Form("%sat%s",slicePlane.Data(),sliceAxis.Data())  ;
  output.Append(slicePlane) ;
  output.Append("_") ;

  TString image ;
  image = Form("%d",sliceCut) ;
  image.Append("mm") ;
  image.ReplaceAll("-","minus") ;

  TString cuts;

cuts=Form("_Slice%immMinMuon%iNMC%i",(Int_t)slice,nbMuonCut,neighborMuonsC
ut);
  if(extension == ".png") image.Append(cuts);
  image.Append(extension) ;

  output.Append(image) ;
  return output ;
}
```