SANDIA REPORT
SAND2014-18497
Unlimited Release
Printed October 2014

Optimal Imaging for Treaty Verification FY2014 Technical Report

Nathan R. Hilton, Matthew A. Kupinski, Christopher J. MacGahan, William C. Johnson, Erik M. Brubaker

Prepared by
Sandia National Laboratories
Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

SAND2014-18497
Unlimited Release
Printed October 2014

Optimal Imaging for Treaty Verification FY2014 Technical Report

Nathan R. Hilton, William C. Johnson, and Erik M. Brubaker
Sandia National Laboratories
P.O. Box 969
Livermore, California 94550-MS9406

Matthew A. Kupinski and Christopher J. MacGahan
University of Arizona
1630 E. University Blvd.
Tucson, Arizona 85721

Abstract

FY2014 technical report of our project funded by DNN R&D that leverages advanced inference methods developed for medical and adaptive imaging to address arms-control applications. We seek a method to acquire and analyze imaging data of declared treaty-accountable items without creating an image of those objects or otherwise storing or revealing any classified information. Such a method would avoid the use of classified-information barriers. We present our progress on FY2014 tasks defined in our life-cycle plan. We also describe some future work that is part of the continuation of this project in FY2015 and beyond as part of a venture that joins ours with a related PNNL project.

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

1.1 Problem statement

This project leverages advanced inference methods developed for medical and adaptive imaging to address arms-control applications. We seek a method to acquire and analyze imaging data of declared treaty-accountable items (TAI) without creating an image of those objects or otherwise storing or revealing any classified information. Such a method would avoid the use of classified-information barriers (IB).

In a typical model of future arms-control monitoring regimes, one or more TAIs are presented for inspection using an agreed-on system of inspection tools. At a minimum, the inspection system must measure and report whether the inspected object is a TAI. The inspection system may also be required to measure and report the number and locations of TAIs present in its field of view. This system must be insensitive to any normal variability in the measurement configuration or environment, including background radiation.

Based on our experience with related radiation imaging systems for medical applications, we should achieve superior performance and avoid classified-information barriers by use of raw image data instead of reconstructed images. The medical tasks of detecting, locating, and classifying the radiation signatures from a medical patient as indicators of either threatening or benign conditions are adaptable to similar treaty-verification tasks. Aspects of task-based methods in nuclear medicine that are key to achieving our arms-control research objectives are: (1) data from individual radiation-detection events can be processed as they are measured, which removes the need to store those data for future analysis, (2) the resultant processed information cannot be analyzed to recreate those original event data, and (3) the resultant processed information cannot be analyzed to extract classified spectroscopic or geometric information about the object being measured. In a treaty scenario, the combination of these properties would enable the use of spectroscopic imaging hardware without an information barrier to prevent the collection or disclosure of classified information.

We are applying and creating methods having these properties. We are doing so in simulations and with existing measurement data using the most appropriate, available hardware. Initially, we are using the fast-neutron coded-aperture imager jointly developed by Oak Ridge and Sandia National Laboratories. Since members of our team contributed to its design, we have ready access to the necessary detector and data-acquisition system details. We will later apply our methods to other imaging systems, specifically those being used in the warhead measurement campaign (WMC).

Until now, the imaging systems used in the WMC as well as most others used in related applications share a major shortfall. Namely, using data collected with these systems to reconstruct and analyze images is not optimal for the task of detecting, locating, and classifying sources of radiation present in a cluttered background-radiation environment. Techniques based on image reconstruction are suboptimal primarily because they do not fully utilize the statistical information available in the raw detection data. Further, in most cases a human is used to analyze the images produced, which is costly and inefficient. Even in those cases where automated algorithms are employed, their performance is inherently hindered both by the information lost and ignored during image reconstruction and by the artifacts present when complex scenes are reconstructed from sparse data. Thus, even automated image-processing methods have limited ability to detect, locate, and classify weak radiation sources and yield high false-positive detection rates. Most importantly, in arms control, no images of the object can be generated for an automated analysis without the use of an IB, which is not a desired solution.

We expect our new, task-based imaging method will enable and enhance automated collection and analysis of raw detection data in a manner that avoids the need for a classified-information barrier. This method is automated, simple to calibrate and use, and is known to optimally utilize collected radiation-detection data (i.e., the raw data) to yield the best-available source detection, localization, and classification performance. In addition to maximizing the performance that can be attained from existing detection hardware, we will propose in future work the application of the task-based imaging method to guide the design of better hardware that is optimized for its task. Our proposed project will greatly benefit DNN R&D's mission to create a highly sought-after tool for use with arms-control treaties.

1.2 Tasks and observer models

In task-based imaging, we define the performance of an imaging system or image-analysis methodologies in terms of how well they can accomplish the relevant tasks. We define the *task* as the purpose for acquiring the image data. In the context of this work, the tasks are, generally, the verification of TAIs. Any person or computer algorithm that uses the image data to perform the task is referred to as an *observer* or *observer model*. Thus, our goal in this project is to develop observer models that can perform the relevant tasks without requiring an information barrier.

Using an imaging system for this type of analysis presents some key challenges because an "image" of a TAI would be considered classified information. Our proposed methods will, as much as possible, rely on list-mode processing to circumvent this issue. With list-mode processing, each detected event is processed by the observer model and then thrown away. This type of event-by-event processing does not generate an image of the TAI. However, the observer model itself must know some information about the task and the TAI in order to have reasonable performance. The primary advantage of our approach is that we can design and control the amount of sensitive information that is present in the observer model – the image data is sensitive by its very nature. The observer models will exhibit a tradeoff between overall performance on the task and the amount of sensitive or classified information required to compute the observer model. Observer models that have precise knowledge of the TAIs and their statistics can perform the task of with very high performance. In contrast, an observer model that has very limited information on the TAIs will have a lower performance, but this lower performance may be acceptable given the fact that an information barrier may not be required.

The relevant tasks for this work include

- Distinguishing between types of TAI
- Distinguishing between one TAI or multiple TAI
- Verify the absence of a TAI
- Distinguish between a TAI and a spoof TAI
- Estimate parameters of the TAI, e.g., position, orientation, etc.

We have thus far focused on similar tasks using the Idaho Inspection Objects described in later sections. A schematic for our techniques is shown in Figure 1.

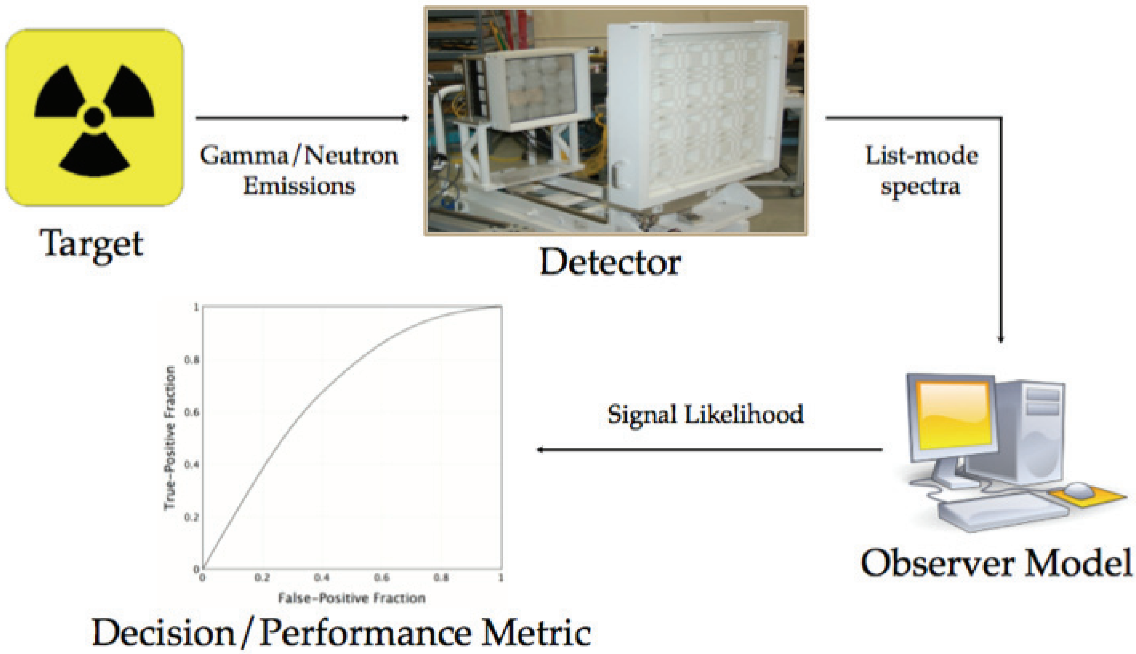

Figure 1 Schematic of our approach to list-mode or event-by-event processing for treaty verification.

1.3 ORNL/SNL fast-neutron coded-aperture imager

The imager used for the studies reported in this document was designed, built, and characterized from 2010–2013 by ORNL and SNL for NA-22, focusing on arms-control treaty-verification applications [Hausladen et al.]. It comprises a high-density polyethylene (HDPE) mask and an array of pixelated organic scintillator detectors that form a position-sensitive image plane for fast neutrons (Figure 2). These components are mounted on a cart with a linear stage that allows adjustment of the distance between them. The detectors are read out with electronics housed in a VME crate and controlled by a data-acquisition computer than can be situated remotely and connected to the imager via Ethernet cable.

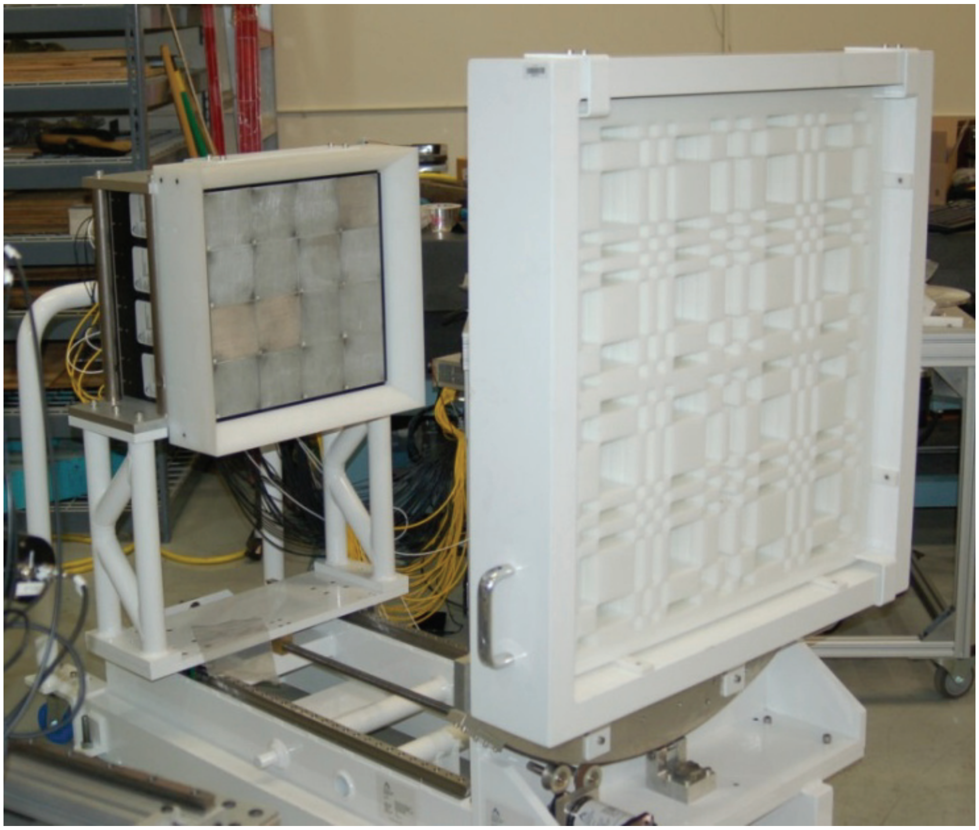

Figure 2 Photo of the ORNL/SNL fast-neutron coded-aperture imaging system. The component on the right is the HDPE coded mask; on the left is the 40 x 40 pixel organic scintillator image plane. These are mounted on a cart for ease of movement and relative positioning.

The image plane contains 16 block detectors, each of which have 10 x 10 scintillator pixels, 1 cm x 1 cm x 5 cm deep each, that are viewed by four PMTs. For an individual scattering event, the pixel in which the interaction occurred can be determined by the ratios of light detected in the four PMTs. An example of a block detector, removed from its housing, is shown in Figure 3. These detectors allow a large reduction in the number of PMTs and electronics channels relative to the number of pixels (4 PMTs for each 100 pixels), but introduce difficulties and uncertainties in data processing, image reconstruction, and modeling.

In numerous ways, including choice of materials in the mask and the active volume, the detection system is optimized for passive fission-energy neutron imaging, which provides good sensitivity to the spatial distribution of plutonium in the measurement target. Additionally, the system can image gamma-ray emissions, but with poor spectral resolution. Measurements are typically performed at a standoff distance of 1-3 m. The field of view and intrinsic angular resolution are adjustable via changes in the mask aperture element size, mask-scintillator distance, and object-mask distance, with corresponding tradeoffs in detection rate.

After data cleanup and analysis cuts, as well as a flat-field correction, a map of the neutron (or gamma) rate in each detector pixel can be constructed for a given imaging run. Those pixel-by-pixel rates, along with knowledge of the coded mask pattern, can be used to reconstruct an image of the neutron- (or gamma-) emitting material in the field of view. The image resolution is driven by multiple factors and choices as described above, as well as the statistics of the data collected, but may be on the order of 1 cm for a high-resolution image of an object within 1-2 m. An example reconstructed image is shown in Figure 4.

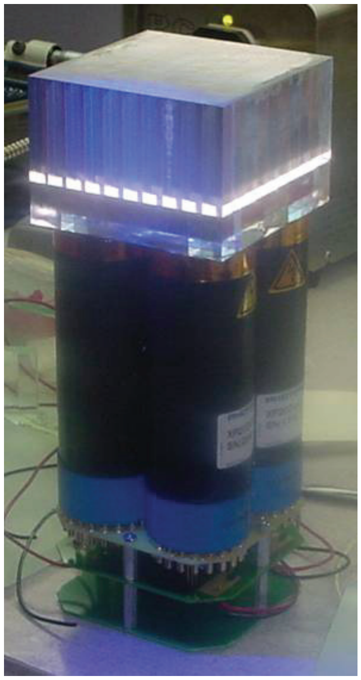

Figure 3 Pixelated block detector used in the coded-aperture imager, without Al housing. This example shows plastic-scintillator pixels; another version of this imager uses liquid scintillator with baffles inserted to pixelate the volume.

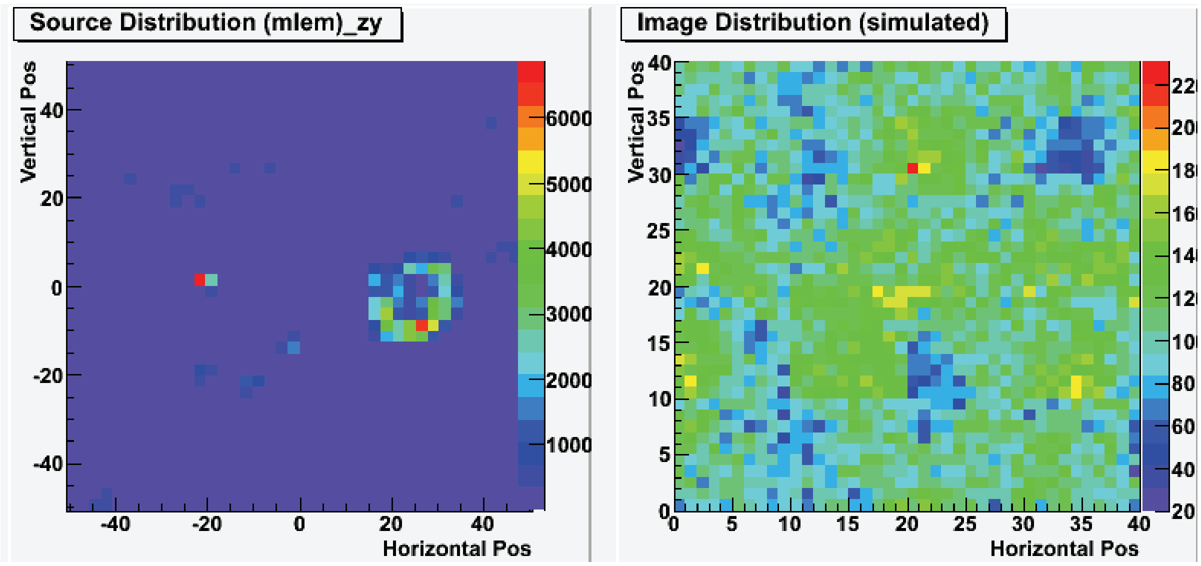

Figure 4 Example image reconstruction from the fast-neutron coded-aperture imager. MLEM was used to reconstruct the image on the left using the pixel-by-pixel neutron rates shown on the right. Those data were collected over a few hours and display a typical modulation pattern induced by the coded mask. The field of view contains both a point source (left side) and an emulated extended source (right side).

1.4 Description of list-mode data

In the calibration stage, data will be accumulated on an initial set of sources to develop the observer models. Depending on the observer model, we record count-rate, spectral, and various other data. In the testing stage, the goal is to use list-mode processing to perform classification and detection tasks on sources other than those used to develop the observer models. By list-mode processing we mean that detection-event data is processed one event at a time. After processing a detection event to update a test statistic, that event's data is forgotten by the processing algorithm. Sensitive data will not be accumulated to make a detection or classification decision. This approach is outlined in Figure 1 above.

The raw data from the imager itself is stored in a binary data format that includes low-level information such as trigger timing and the output of some rudimentary pulse processing performed by the high-speed digitizers. At a later analysis stage, the information from the four PMTs viewing a single block detector is combined to reconstruct each detector hit and apply calibrations. The reconstructed quantities include:

- X position estimate within the block detector
- Y position estimate within the block detector
- Summed pulse amplitude
- Summed pulse shape parameter

And the final calibrated output is:

- Pixel ID (or x, y)
- Energy deposited
- Neutron likelihood
- Gamma likelihood

Although currently implemented as a separate analysis stage, the hit reconstruction and applying calibrations could easily be performed as the data is acquired. For the purposes of our studies, this calibrated output is therefore treated as the list-mode data that is generated by the detector in real time. Our simulations are designed to reproduce as much of the statistical uncertainties and calibration uncertainties in these quantities as possible.

2. TASKS

Our life-cycle plan for the 2014 fiscal year entailed four tasks. Progress on each of these tasks is described in the corresponding named sections below.

2.1 Model imaging system and inspection objects

The end goal of this project is to create observer models that require as little sensitive information as possible yet effectively detect and classify TAIs. To develop and test our methods, we created accurate simulation models of unclassified inspection objects (IO) developed at Idaho National Laboratory (INL).

Gamma ray and neutron transport was accomplished with the GEANT4 toolkit [Agostinelli et al.; Allison et al.]. GEANT is an open source Monte Carlo physics simulation library and was chosen because it is highly flexible. Users have the ability to create their own emission and detection classes as well as interact with the simulation in any given step. For example, GEANT spends considerable time transporting thermal neutrons through plastic, so applying an energy cutoff for particles inside the plastic mask geometry has proven to provide a valuable 25% speed increase. GEANT has a built-in library developed by Lawrence Livermore National Laboratory to simulate fission-neutron emission (both energy spectra and number of neutrons emitted per fission event). Gamma emission is done with a Sandia-built library created by Will Johnson that offers custom aging, isotope mixtures, and quick computation time.

We coded into GEANT the unclassified IOs, as we will describe, and the ORNL/SNL fast-neutron imager already described in section 1.3. These models are illustrated in Figure 5.

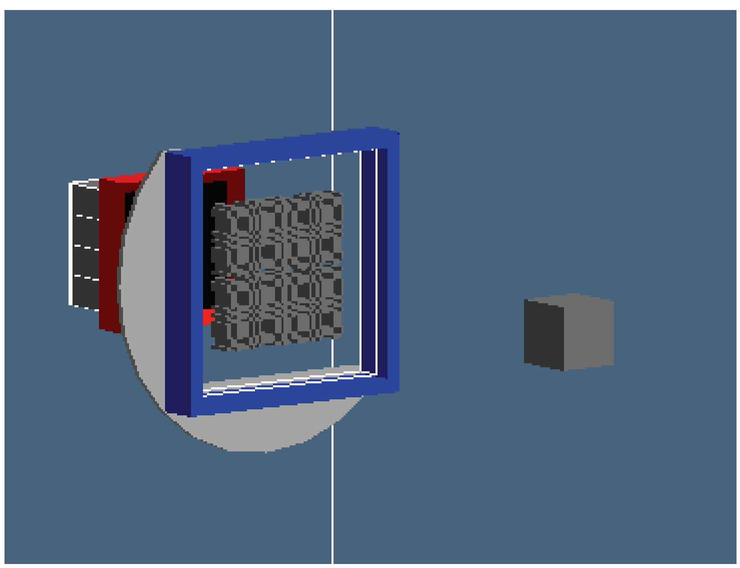

Figure 5 GEANT model of the imaging system and an inspection object (IO), which is enclosed in an aluminum box. The imager (on the left) is readily recognizable by the pattern of the HDPE coded-aperture mask. HDPE shielding located between the periphery of the mask and the blue-colored frame is shown as transparent for this illustration.

The number of detected particles required for source classification is task dependent as some tasks are easier than others. In order to determine the number of source particles to simulate, we instituted standard statistical checks from MCNP software [Shultis and Faw]. GEANT does not have any statistical checks already built into the code to identify when a sufficient number of particles have been simulated, so we coded in these checks ourselves. While these checks are all useful, we emphasized the relative error magnitude check, namely,

$$\frac{S_{\bar{x}}}{\bar{x}} \leq 0.05 \quad . \tag{1}$$

The relative standard deviation on the estimate of the mean weight binned into each spatial pixel must be less than or equal to 5%. For an unbiased simulation, in which all weights are 0 or 1, this corresponds to 400 or more counts on each pixel.

Fast-neutron detector

The geometry used for our simulation is a model of the fast-neutron imager (see Figure 2 and Figure 5). We obtained two GEANT models for this detector—one each from SNL and ORNL. We checked them for consistency and used them as the basis for the detector model we have used in this work. A newer, plastic-scintillator-based version of this imaging system has been developed (see Figure 3), but we do not yet have a GEANT model of this version. Nevertheless, differences between the detector geometries of the liquid- and plastic-scintillator designs are small.

The detector response as currently implemented is fairly simple – the energies collected in a given block are binned into a mean pixel location. Currently, the gamma and neutron PSD is unrealistically 100% accurate in our simulations, but we will soon implement realistic, observed PSD performance. We apply Gaussian energy blurring to each simulated particle's energy that is deposited in the detector array using the measured, parameterized detector response shown in Figure 6.

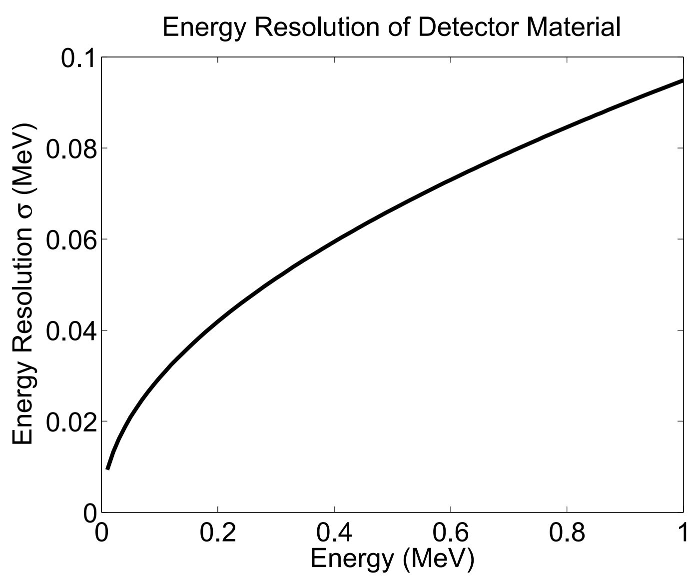

Figure 6 Detector energy resolution used in the simulations. In post-processing, we distribute the particle weight into nearby energy bins according to a Gaussian energy distribution of the width indicated and centered on the computed energy deposited in a detector pixel.

Inspection objects

INL designed and built a set of ten plutonium- and/or uranium-containing, unclassified inspection objects [Neibert et al.]. We modeled inspection objects 5 through 10 in GEANT. The naming convention for these inspection objects is IO5, IO6, etc. Figure 7 and Figure 8 give a visual representation of these objects and Figure 9 shows a portion of the GEANT model we created for IO8. These IOs generally consist of an assembly of radioactive source plates inside shielding materials and held in place by a surrounding framework and an enclosing box. We are the first group to model these IOs in GEANT and believe these models could prove useful to other researchers. We have found some and are actively seeking additional measurement data from these IOs and are in the process of validating our simulation results.

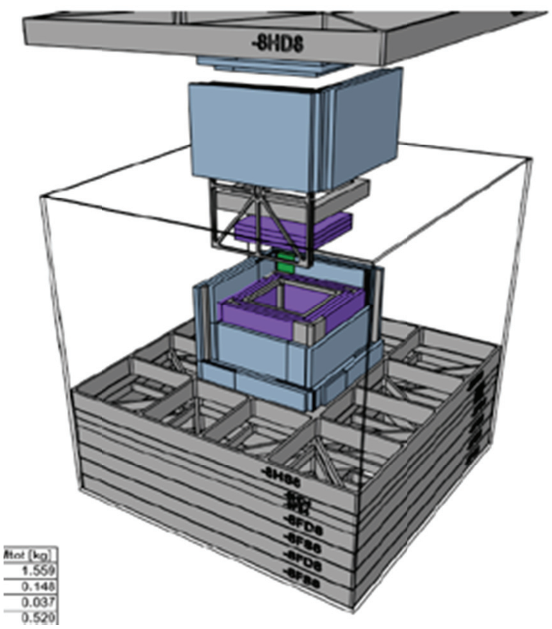

Figure 7 IO8 contains plutonium (purple) shielded by depleted uranium (light blue).

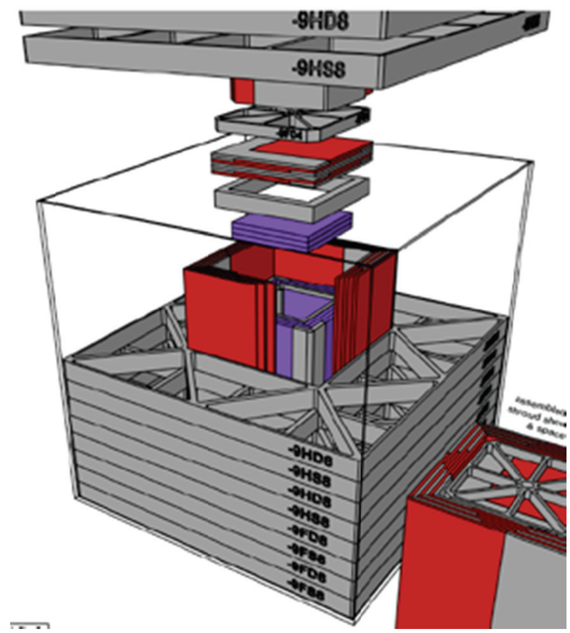

Figure 8 IO9 contains plutonium (purple) shielded by highly enriched uranium (red).

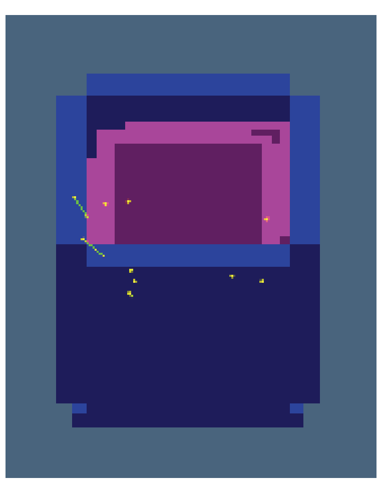

Figure 9 GEANT model of IO8 without aluminum box, supports, or top layer of depleted uranium**.**

Figure 10 and Figure 11 emphasize the dominant effect of uranium shielding on gamma-ray transport, and the lesser effect it plays in neutron transport. These results ignore the effects of self shielding, which further exaggerates the difference between the number of gamma-rays and neutrons required to be simulated.

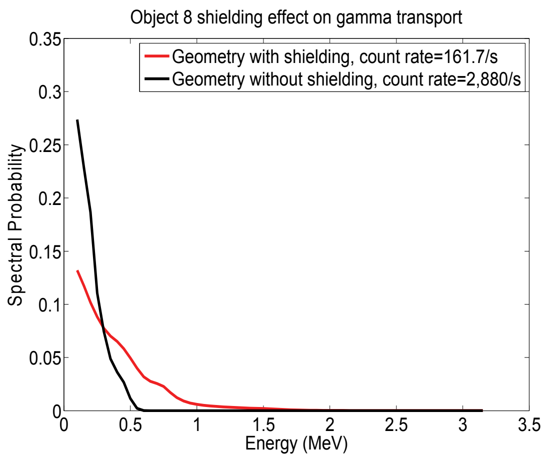

Figure 10 IO8 gamma spectral and count rate comparison with and without depleted uranium shielding.

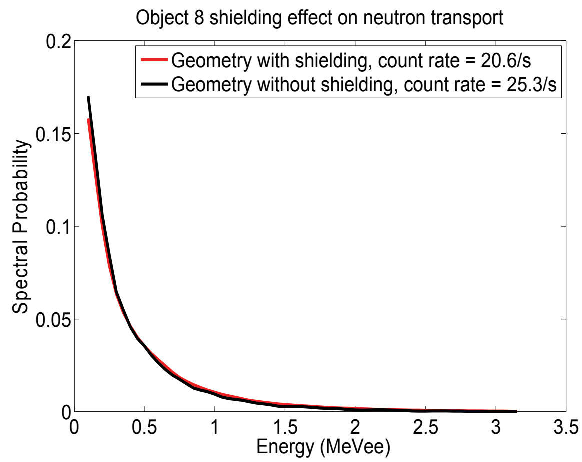

Figure 11 IO8 neutron spectral (after energy quenching) and count rate comparison with and without depleted uranium shielding.

Without variance reduction, one GEANT run imaging IO8 with 2 billion emissions yields 2 counts on the entire detector and takes 20 hours to simulate. To have a relative error on each pixel of 5% or less, which corresponds to 400 counts per pixel, implies 6.4M CPU hours. Further, we need to run numerous simulations over various nuisance parameters. Meanwhile, neutron simulations require ~300 to 1200 CPU hours depending on the source. Using a combination of techniques, we have been able to substantially reduce the computational burden of these simulations.

Variance reduction

We analyzed multiple variance reduction techniques to accelerate our simulations.

1. Primary-particle biasing – biasing in phase and energy space in order to preferentially sample from sections of the source distribution more likely to hit the detector.
2. Geometric importance sampling and weight windowing – forming a parallel geometry full of importance values where incoming particles are split at particle boundaries to make every particle count.
3. Energy cutoff – We are mostly interesting in observing gammas in the 200+keV range. A 100 keV gamma cutoff at the source and during the radiation transport proved very useful as materials containing plutonium have strong emission lines at 60 and 26 keV. Also, scattered low-energy gammas are numerous but not important.

Table 1 indicates the speedups achieved using variance-reduction techniques. The combination of energy cutoff and moderate linear energy biasing of primary gamma rays proved to be the most useful. With such biasing, simulated particles no longer have identical, unity weights. Therefore, "speedup" is the scale factor for the CPU time required to achieve an estimated relative error of 5% in the number of particles expected to interact with the scintillator material. In addition to estimating the relative error during our simulations, we also examine other statistical checks to confirm that a set of simulation results is reasonably accurate.

Table 1 Results of Monte Carlo variance reduction

VR method	CPU hours required	speedup
None	6.40E+06	1
100keV cutoff	17200	372
100keV cutoff + linear E bias	4800	1333

Even with these speedups, we were still faced with a daunting 4800 CPU hours to simulate a source. On two university-lab servers, each with eight cores, this would take about two weeks to simulate one position and orientation of the source and imager. The remaining improvements are structural in nature. They don't reduce the total CPU time required to simulate the object, but they push the real life time down to a manageable number.

Separating simulation parts

We observed that the large majority of computation time (80% for neutrons, 99% for gammas) was spent simulating emissions through the source geometry. We therefore separated the simulation into two parts, as illustrated in Figure 12 and Figure 13.

1. Transport particles to a spherical surface outside of the INL sources. Data for these successfully emitted particles is recorded in list-mode using a ROOT data structure.
2. Transport particles from that spherical surface to the detector. Read the list-mode data from the source file, transport to the detector (and any other structure in that model), and record list-mode detection data in another ROOT data file.

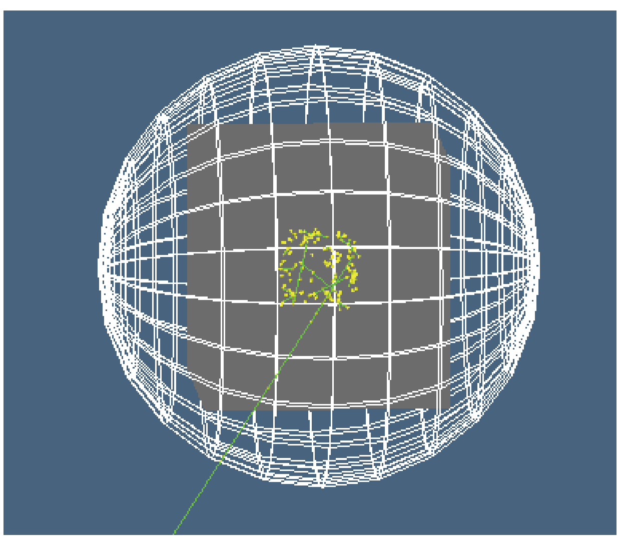

Figure 12 Part one of GEANT simulation. An IO (the gray cube) and any other sources are inside a sphere where incident particles' data are saved for use in subsequent simulations.

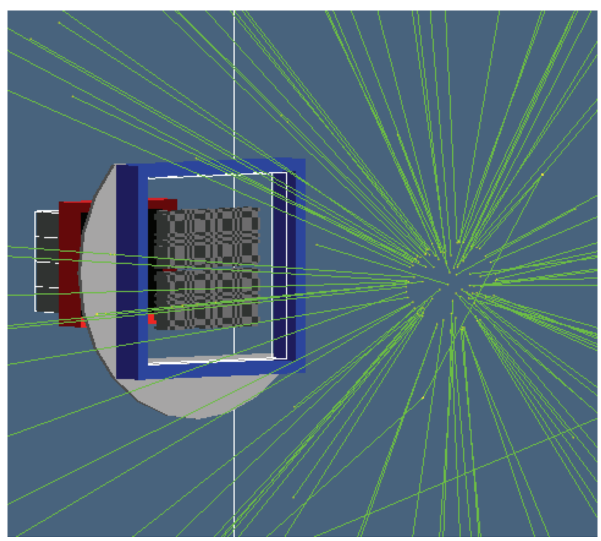

Figure 13 Part two of GEANT simulation. List-mode data from on spherical surface is inserted into geometry with detector, and particles are transported to the detector.

Separating the simulation into parts improves the computational situation dramatically. In part one, approximately 500 and 4780 CPU hours were required to obtain enough information on the sphere for neutrons and gammas, respectively. Approximately 72 and 28 CPU hours were necessary to transport the recorded neutrons and gammas to the imager in part two, which can then be more easily repeated for different imaging configurations. The downside is that significant data storage is required – 200-800 GB per source. However, this amount is available and manageable, and we use ROOT for relatively efficient data input/output.

Parallel processing

GEANT4.10, released in December 2013, offers the user a multithreaded option. This incorporates event-level parallelism as the emission class, detector class, and track information are stored separately in each thread. On a 16-core SNL CPU, we achieved the speedup factors show in Table 2 for simulating gamma-ray transport from an IO to the intermediate sphere (i.e., part one described above). Similar speedup was achieved for neutron source processing.

Table 2 Speed improvement in multithreaded GEANT4.

Time to transport 5 x10^7 IO gamma rays to the intermediate sphere		
threads	real time [s]	speedup
1	3543	1
4	934	3.8
8	514	6.9
12	390	9.1
16	345	10.3

We are still working on improving the speed of the second part of the transport simulations, which slows down due to user thread locking of the root input file during data read in. Thus, the second simulation currently takes about 28 and 20 hours for neutrons and gammas, respectively, to processes the same data sets discussed above.

Sandia HPC

We are now making use of Sandia's high-performance computing resources. At the moment, this project is running only on the Glory cluster, which has 272 nodes with 16 cores each. There is potential in the future to run on Red Sky (2846 nodes with 8 cores each) or Chama (1232 nodes with 16 cores each). Hypothetically, by utilizing all Glory nodes, source simulation can be accomplished in about 4 hours. As we discuss in section 3, this combination of computational accelerations should enable simulating a large number of source orientations and locations to study the effect of nuisance parameters when classifying these objects.

Simulation results

Figure 14 and Figure 15 are two example count maps, one for gammas and one for neutrons for source IO8. These data sets were used in the observer studies we discuss in section 2.4. Figure 16 and Figure 17show the differences in IO8 and IO9 data as well as the change in spectra and count rate due to rotating a source.

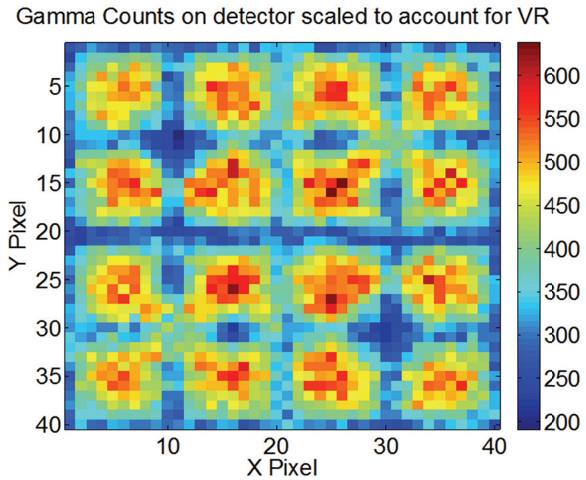

Figure 14 Gamma count map from imaged IO8 corresponding to 66 minutes of live time.

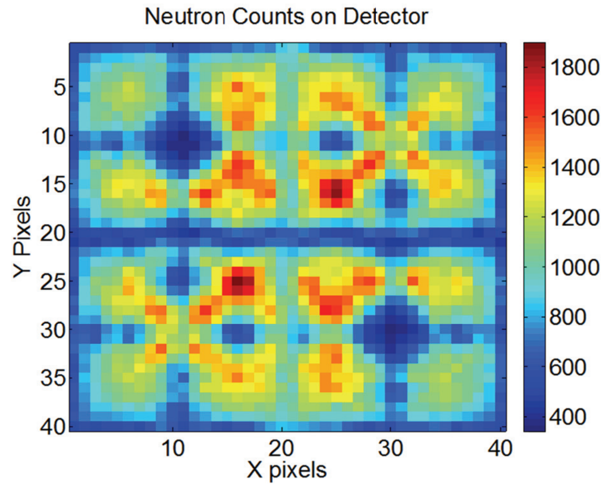

Figure 15 Neutron count map from imaged IO8 corresponding to 21 hours of live time.

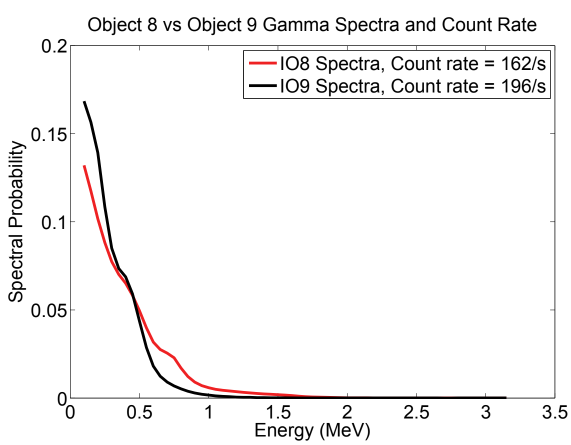

Figure 16 Comparison of IO8 and IO9 gamma spectra and count rate on the detector. The 186 keV line from U-235 plays a significant role in the IO9 spectrum.

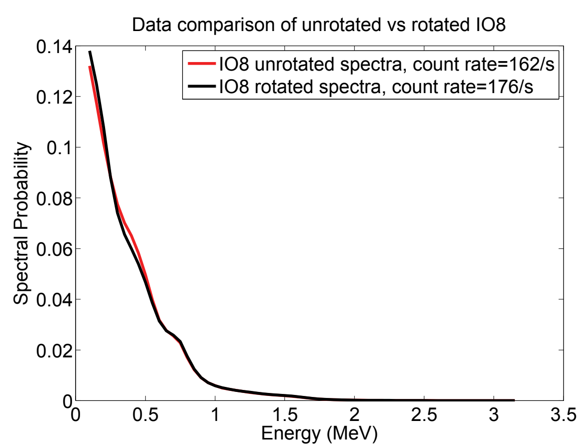

Figure 17 Comparison of IO8 rotated and unrotated spectra and count rate.

2.2 Develop and test methods with various levels of information barriers

One potential task in treaty verification could be differentiating between two or more sources, and our work has focused primarily on this task so far. Tasks that categorize objects into one of two types are known as binary categorization tasks. The observer that optimally performs this type of task is well known to be the Bayesian ideal observer, although it is often difficult to compute. We focus on this observer model because it has the best possible performance and requires a great deal of information. Thus, subsequent observer models can be compared to the ideal observer in terms of performance and information required.

We define an event for the fast-neutron imager as $A_n = \{d_n, E_n, p_n\}$ where d_n is the estimated position of the detected event, E_n is the estimated energy, and p_n is the estimated event type (i.e., gamma ray or neutron). We define one category or state as H_2. For example, this state could be IO8. In contrast, an opposing state H_1 represents a different category, such as IO9. The Bayesian ideal observer computes the likelihood ratio,

$$\Lambda\left(\{A_n\}\right) = \frac{pr(\{A_n\}, N \mid H_2)}{pr(\{A_n\}, N \mid H_1)} \tag{2}$$

where $pr(\{A_n\} \mid H_j)$ is the probability density of all of the list-mode data given that the object being imaged is of type H_j and N is the total number of detected events. We can expand the numerator and denominator of the likelihood ratio into,

$$\begin{aligned} & pr\left(\{A_n\}, N|H_j\right) \\ & = \int d\overline{N_j} \int d\overline{N_0}\, pr\left(\{A_n\}|N, \overline{N_j}, \overline{N_0}, H_j\right) \Pr\left(N|\overline{N_0}, \overline{N_j}, H_j\right) pr\left(\overline{N_0}\right) pr(\overline{N_j}) \end{aligned} \tag{3}$$

where N_0 is the number of background events, N_j is the number of primary events with the constraint that $N = N_0 + N_j$. The terms $\overline{N_0}$ and $\overline{N_j}$ represent the means of the background and primary events, respectively. Given knowledge of the number of events N, we can write the probability density of the list-mode data in N terms of a product over each individual event.

$$\begin{aligned} pr\left(\{A_n\}|N, \overline{N_j}, \overline{N_0}, H_j\right) &= \prod_{n=1}^{N} pr\left(A_n|\overline{N_j}, \overline{N_0}, H_j\right) \\ &= \prod_{n=1}^{N} [pr\left(A_n|h_0\right) \Pr\left(h_0|\overline{N_j}, \overline{N_0}\right) + pr\left(A_n|h_1\right) \Pr(h_1 \mid \overline{N_j}, \overline{N_0}, H_j)] \end{aligned} \tag{4}$$

where h_0 and h_1 denote that the event n is a background event or a primary event from the inspected object, respectively. Thus, the term $pr(A_n \mid h_0)$ represents the distribution of energy, detector position, and particle type for background events. The term $pr(A_n \mid h_1)$ represents the distribution of energy, position, and particle type for source events. This term represents the model of the source, and we currently use extensive Monte Carlo simulations (described in earlier sections) to determine the form of this distribution.

Referring back to (3), we can also write expressions for the subsequent terms in the integrand. Namely, we note that

$$\Pr\left(N|\overline{N_0},\overline{N_j},H_j\right)=\sum_{N_0=0}^{N}\Pr\left(N_0|\overline{N_0}\right)\Pr(N-N_0\,|\,\overline{N_j}) \tag{5}$$

and that each term in the sum is well modeled using Poisson distributions since they represent count-rate variabilities with known means. Finally, the last two terms in (3) represent the uncertainty in the rates from the inspected object and the background. For example, these terms account for variabilities due to the decay of the inspected object and the natural variability of the background strength caused by geographic location and other environmental changes.

The Bayesian ideal observer requires knowledge or accurate models of all sources of variability. In practice, we compute the likelihood ratio of the form,

$$\Lambda\left(\{A_n\}\right)=\int d\gamma\,\Lambda(\{A_n\}\,|\,\gamma)\,pr(\gamma\,|\,\{A_n\}) \tag{6}$$

where γ represents all possible nuisance parameters. A nuisance parameter is something that affects the image data but is of no interest in regards to the task. For example, the orientation of the TAI may not be important information to estimate but the image data may change substantially depending upon the orientation of the item. In the above formulation of the ideal observer, we have accounted for nuisance parameters relating to the count rate and background strength. Other nuisance parameters can be accounted for using the expression above. Computationally, nuisance parameters represent a difficulty in implementing observer models. Marginalizing over object position, orientation, etc. requires knowing the system response of the object to all of these changes. However, fundamentally, this can be treated as a methodology for determining which parameters require modeling and which do not. For example, we will determine whether the performance of the Bayesian ideal observer degrades substantially when the orientation of the inspected object is not accounted for. If the performance does not degrade, then the orientation nuisance parameter can be ignored. In addition, we have devised a number of alternative approaches to dealing with nuisance parameters that are described in the future-work section.

The Hotelling observer, named after the statistician Harold Hotelling, is a much simpler observer model than the Bayesian ideal observer. The Hotelling observer requires knowledge of the first- and second-order statistics of the data, which can include variability due to nuisance parameters, count-rate statistics, and detector cross talk. The Hotelling observer is akin to template-matching observers and can be written in terms of list-mode acquisition as,

$$t=\sum_{n=1}^{N}\mathbf{w}^t\mathbf{T}(A_n) \tag{7}$$

where $\mathbf{w}$ is the template vector and $\mathbf{T}(A_n)$ is a known, vector-valued function. The test statistic t (not to be confused with the transpose operator, which is shown in superscript) is then thresholded to determine whether to classify as H_1 or H_2. The Hotelling observer can be computed using templates that include all energy bins and detector bins, all detector bins summed over energy bins, all energy bin summed over detector bins, or any combination of detector and energy bins. Thus, with the Hotelling observer, we have fine control over the form of the template and what information is obscured by the observer. The performance of the Hotelling observer will be shown in later sections.

The Hotelling observer can also be generated for a set of values of the nuisance parameters. Then, a series of test statistics is generated for each possible value of the nuisance parameters. When thresholds are applied, the Hotelling observer generalizes to the scanning linear observer, which has been shown to approximate the Bayesian ideal observer but is much simpler to compute and requires less information. Implementation of this observer model will be completed in FY15.

Implementations of observer models

Signal-known-exactly ideal observer for a single orientation

As a reminder, the signal-known-exactly (SKE) ideal observer stores the count rates and spectra on each pixel in the calibration stage. In the testing stage, it only stores the counts received and will process the energies recorded in list mode. In simulation, the calibration and testing data sets are independent, as they would be in real life. The observer models are developed from the calibration data. Observed energies are processed in list mode. The steps to perform two-source classification in an SKE study with the ideal observer in MATLAB are as follows.

1. MATLAB reads in four data sets – calibration and testing data for each hypothesis. Currently, we insert background radiation events into each data set using a simplifying assumption that the background spectrum is the same on every detector pixel.
2. Calibration stage: Create observer model from H_1 and H_2 calibration data set that will act on sample data. Looking at (2) - (4) above, the variables $\overline{N_0}$, $\overline{N_1}$, $\overline{N_2}$, $pr\left(A_n|h_0\right)$, $pr\left(A_n|h_1\right)$, and $pr\left(A_n|h_2\right)$ are set in this step.
3. Testing stage: Loop over N_t trials. N_t sets of sample data will be taken from the H_1 and the H_2 testing sets. The observer then acts on these data, leading to N_t test-statistic values for H_2 and N_t values for H_1.
   a. For a given acquisition time, mean number of observed particles will be
   $$\overline{N_p} = DetectionRate * Acq\,time\,. \tag{8}$$
   b. Find number of particles N_p by taking a Poisson random sample from mean $\overline{N_p}$. Create two sets of sample data – one from data under the H_2 hypothesis and another for the H_1 hypothesis.
   c. Sample N_p particles from testing data set to create full sample data.
   d. Act on this sample data with observer model found in step 2 to compute the test-statistic value. Each likelihood will be the product of many small probabilities, which leads to a result too small to compute. Instead, we compute the sum of log likelihoods.
4. The end result is N_t test-statistic values for both the H_2 and H_1 images. We use receiver-operating-characteristic (ROC) analysis to characterize observer performance. The area under the ROC curve (AUC) is estimated using two-alternative forced-choice (2AFC) methods in which the test statistics for pairs of H_1 and H_2 images are compared. The fraction of correct decisions when the observer is forced to choose one image as class 2 and the other as class 1 is an unbiased estimate of the AUC [Barrett and Myers]. Section 2.4 has further discussion of these methods of task-performance analysis.

Signal-known-exactly ideal observer for multiple orientations

In the case of the generalized ideal observer models averaging over different source orientations, multiple calibration data sets are read in. Their count rates and spectra are all stored in the calibration stage, and their likelihoods are averaged. The likelihood expression for the source 1 hypothesis is:

$$pr\left(\{\mathrm{A_n}\}|H_1\right)=\frac{1}{2}\left(pr\left(\{\mathrm{A_n}\}|H_1\ ,\theta=0^o\right)+pr\left(\{\mathrm{A_n}\}|H_1\ ,\theta=45^o\right)\right). \tag{9}$$

Thus, the ideal observer that classifies multiple orientations is

$$\Lambda\left(\{A_n\}\right)=\frac{pr\left(\{A_n\}|H_2\ ,\theta=0^o\right)+pr\left(\{A_n\}|H_2\ ,\theta=45^o\right)}{pr\left(\{A_n\}|H_1\ ,\theta=0^o\right)+pr\left(\{A_n\}|H_1\ ,\theta=45^o\right)}. \tag{10}$$

This method is relatively straightforward. Each likelihood goes to zero because it is the product of many small numbers, so the logs of these expressions are tallied instead. The four terms above must be calculated independently. Since likelihoods are less than one, and therefore, their logs are negative, the largest log is subtracted out and the data is re-exponentiated to find the observer value.

Ideal observer incorporating count-rate variability

The next observer model attempts to account for a lack of knowledge about the age of the source material of the objects to be classified. While a difference in age will lead to changes in both spectrum and count rate, only count rate was considered here. The MATLAB implementation is more complicated than the previous models. The calibration data could have a different count rate than the source to be classified. To account for this, we create a user guess on the count-rate variability. We assume the distribution is Gaussian with a mean centered at the calibration-data mean and a user-defined standard deviation. The algorithm outline is as follows. Since the observer model changes with the sample data, generating sample data occurs before creating an observer model. Similar to the SKE observer, this observer requires the number of counts to be stored in the sample data, but it will process the particle energy event by event from the list-mode data.

1. Generate calibration data
    a. Calibration data (spectra and count rate data on each pixel) is generated in GEANT, similar to the SKE ideal observer.
2. Generating Sample Data
    a. For a given acquisition time, the mean number of observed particles will be
    $$\overline{\overline{N_p}}=DetectionRate*Acq\,time\ . \tag{11}$$
    b. For a single trial, a Gaussian random variable $\overline{N_p}$ is chosen from a distribution with mean $\overline{\overline{N_p}}$ and variance $\sigma_p{}^2$.
    c. Sample number of counts N_p taken from Poisson random variable with mean $\overline{N_p}$.
    d. Sample N_p counts from pixel/spectral distribution.

3. Create observer model

The adaptation of (3) to something usable in MATLAB is not trivial. Starting with the ideal observer

$$\Lambda\left(\{A_n\},N\right)=\frac{pr\left(\{A_n\},N\mid H_2\right)}{pr\left(\{A_n\},N\mid H_1\right)}\ , \tag{12}$$

both the numerator and dominator include integrals over the background distribution and a signal distribution. If we expand this expression to incorporate variability in the sources and background, we obtain

$$\Lambda\left(\{A_n\},N\right)=\frac{\iint pr\left(\{A_n\},N|\overline{N_b},\overline{N_2},H_2\right)pr\left(\overline{N_b}\right)pr\left(\overline{N_2}\right)d\overline{N_b}d\,\overline{N_2}}{\iint pr\left(\{A_n\},N|\overline{N_b},\overline{N_1},H_1\right)pr\left(\overline{N_b}\right)pr\left(\overline{N_1}\right)d\overline{N_b}d\,\overline{N_1}}\ , \tag{13}$$

which is difficult to compute because the numerator and denominator must be evaluated separately. Ideally, we would evaluate the numerator with Monte Carlo methods that randomly sample $\overline{N_b}$ and $\overline{N_2}$ from their distributions, evaluate $pr\left(\{A_n\},N|\overline{N_b},\overline{N_2}\right)$, and average the value over the number of samples. Unfortunately, the numerator and denominator will each go to zero numerically. Each likelihood contains a product of at least 1600 values (for the Poisson component) that are all less than one, and spectral terms further reduce the overall product. Unlike the SKE case, we cannot take the log of these values, as they need to be summed to find the integral value. Working with these numbers without taking the log is computationally infeasible.

Instead of using brute force to compute (13), we create the observer model by adapting a previously studied method that uses Markhov-chain Monte Carlo techniques [Kupinski et al.]. The first step is to replace the numerator in (12) with (3). Thus,

$$\Lambda\left(\{A_n\},N\right)=\frac{\iint pr\left(\{A_n\},N|\overline{N_b},\overline{N_2}\right)pr\left(\overline{N_b}\right)pr\left(\overline{N_2}\right)d\overline{N_b}d\,\overline{N_2}}{pr\left(\{A_n\},N\mid H_1\right)} \tag{14}$$

where $\overline{N_2}$ is a Gaussian random variable with a known mean (from calibration data) and standard deviation (user guess) over the source-2 counts, and $\overline{N_b}$ is the same over the background counts.

Next, we marginalize over $\overline{N_1}$ in the numerator, which will help with future math. As there are no $\overline{N_1}$ variables in the numerator, integrating over the $\overline{N_1}$ probability density does not change the result.

$$\Lambda\left(\{A_n\},N\right)=\frac{1}{pr\left(\{A_n\},N\mid H_1\right)}*$$
$$\iiint pr\left(\{A_n\},N|\overline{N_b},\overline{N_2}\right)pr\left(\overline{N_b}\right)pr\left(\overline{N_1}\right)pr\left(\overline{N_2}\right)d\overline{N_b}d\,\overline{N_1}d\overline{N_2} \tag{15}$$

Then we multiply numerator and denominator inside the integral by $pr\left(\{A_n\},N\mid\overline{N_b},\overline{N_1}\right)$, which is the SKE likelihood used in the observer models discussed previously, to obtain

$$\Lambda\left(\{A_n\},N\right)=\frac{1}{pr\left(\{A_n\},N\mid H_1\right)}*$$

$$\iiint\frac{pr\left(\{A_n\},N|\overline{N_b},\overline{N_2}\right)pr\left(\{A_n\},N|\overline{N_b},\overline{N_1}\right)pr\left(\overline{N_b}\right)pr\left(\overline{N_1}\right)pr\left(\overline{N_2}\right)}{pr\left(\{A_n\},N|\overline{N_b},\overline{N_1}\right)}d\overline{N_b}d\overline{N_1}d\overline{N_2}. \tag{16}$$

Noting the leftmost terms in numerator and dominator form the SKE ideal observer $\Lambda(\{A_n\},N\mid\overline{N_b},\overline{N_1},\overline{N_2})$, this expression becomes

$$\Lambda\left(\{A_n\},N\right)=\frac{1}{pr\left(\{A_n\},N\mid H_1\right)}*$$

$$\iiint\Lambda\left(\{A_n\},N|\overline{N_b},\overline{N_1},\overline{N_2}\right)pr\left(\{A_n\},N|\overline{N_b},\overline{N_1}\right)pr\left(\overline{N_b}\right)pr\left(\overline{N_1}\right)pr\left(\overline{N_2}\right)d\overline{N_b}d\overline{N_1}d\overline{N_2}. \tag{17}$$

We reduce this expression further using Bayes' rule:

$$\frac{pr\left(\{A_n\},N|\overline{N_b},\overline{N_1}\right)pr\left(\overline{N_b}\right)pr\left(\overline{N_1}\right)}{pr(\{A_n\},N\mid H_1)}=pr(\overline{N_b},\overline{N_1}\mid\{A_n\},N). \tag{18}$$

$pr\left(\overline{N_b},\overline{N_1}|\{A_n\},N\right)$ is a posterior probability density. In the MATLAB model, this is found by creating a matrix where the row indices are given by potential counts in the $\overline{N_b}$ distribution and column indices by the $\overline{N_1}$ distribution, multiplying the three terms in the numerator together, and normalizing the result. The spectral terms cancel out of the numerator and denominator, leaving only count-rate terms. The end result for the ideal observer is

$$\Lambda\left(\{A_n\},N\right)=\iiint\Lambda\left(\{A_n\},N|\overline{N_b},\overline{N_1},\overline{N_2}\right)pr\left(\overline{N_b},\overline{N_1}\mid\{A_n\},N\right)pr\left(\overline{N_2}\right)d\overline{N_b}d\overline{N_1}d\overline{N_2}. \tag{19}$$

We calculate this integral using Monte Carlo methods by sampling values for $\overline{N_b}$, $\overline{N_1}$, and $\overline{N_2}$ many times; calculating the SKE ideal observer for each (which requires full knowledge of the sample data to compute); and finding the average SKE value. Unlike (13), (19) is calculable because it uses the SKE ideal observer, which takes the ratio of the two likelihoods, which is computationally feasible. To calculate this given a set of sample data:

a. Sample $\overline{N_2}$ from its probability distribution, which has known Gaussian mean and a user guess at standard deviation.
b. Find $pr\left(\overline{N_b},\overline{N_1}\mid\{A_n\},N\right)$ through the method described above, and sample $\overline{N_b}$ and $\overline{N_1}$ from this distribution.
c. Find $\Lambda\left(\{A_n\},N|\overline{N_b},\overline{N_1},\overline{N_2}\right)$.
d. Repeat steps a through c and average.

Hotelling observer models

Before discussing the implementation of the Hotelling observer, it is convenient to recast (7) with a notation change. We can represent the operator $\mathbf{T}$ acting on the list-mode data $\{A_n\}$ by the vector $\mathbf{g}$. Bolded quantities are vectors. Depending on the operator, $\mathbf{g}$ can have different meaning (and values). It could contain the number of counts in a given pixel-spectral bin, similar to the ideal observer, or it could be some other combination of pixel-spectral data.

The Hotelling observer (7) as discussed in [Barrett and Myers] is derived to be the linear observer that maximizes the signal-to-noise ratio of the observer's test statistics. The test statistic for the Hotelling observer is given by,

$$t = \overline{\Delta\mathbf{g}}^t K_{\mathbf{g}}^{-1}\mathbf{g} \tag{20}$$

where the product of the first two terms in (20) is a vector of length *M* containing the weights $\mathbf{w}$ in (7). Thus,

$$\begin{aligned} \mathbf{g} &= \sum_n \mathbf{T}(A_n) \\ \mathbf{w} &= K_{\mathbf{g}}^{-1}\overline{\Delta\mathbf{g}} \\ t &= \mathbf{w}^t \sum_n \mathbf{T}(A_n) \end{aligned} \tag{21}$$

In (20), $\overline{\Delta\mathbf{g}}$ is the difference in each calibration data bin between the H_1 and H_2 hypothesis,

$$\overline{\Delta\mathbf{g}} = \overline{\mathbf{g}_2} - \overline{\mathbf{g}_1} \ , \tag{22}$$

and $K_{\mathbf{g}}$ is the averaged covariance matrix of the calibration image data from the two hypotheses,

$$K_{\mathbf{g}} = \frac{1}{2}\left(K_1 + K_2\right) \ . \tag{23}$$

The Hotelling observer lends itself readily to list-mode processing. The execution of each observer is carried out in 3 steps. First, we define the data vector $\mathbf{g}$ to be used, which is determined by the choice of operator $\mathbf{T}(A_n)$. Second, we assign $\mathbf{w}$ based on the acquired calibration data using (21). Specifically, we set $\overline{\mathbf{g}_1}$ and $\overline{\mathbf{g}_2}$ equal to the calibrated data for source 1 and source 2 and find the corresponding covariance matrixes K_1 and K_2 from the calibrated data. Third, during list-mode processing in the testing stage, the list-mode entry for event n is processed using (7), which corresponds to multiplying one value from the weight vector with a value determined from the attribute A_n and adding this result to the summed test statistic *t*. After this process, that event's data can then be forgotten. Besides the test statistic, we make no other aggregation of detection-event data (e.g., spectra, count rates). This process is illustrated below in two versions of the Hotelling observer we created in this project.

1. The first Hotelling observer we created uses the number of counts in each pixel-spectral bin.
   a. Choose operator $\mathbf{T}$ such that $\mathbf{g}$ is a vector of counts N_m observed in each spatio-spectral bin *m*. Using notation that is useful when performing list-mode data processing,
$$g_m = N_m = \sum_{n=1}^{N_m} 1 \ . \tag{24}$$
   b. Record first- and second-order statistics of calibration data. All Hotelling observers require the difference of means and the average covariance of $\mathbf{g}$, but these values depend on the meaning of linear operator used in $\mathbf{g}$. In the calibration stage for this version of the Hotelling observer, the numbers of counts in each energy bin of each pixel define the mean-value vectors $\overline{\mathbf{N}_2}$ (under source 2 hypothesis) and $\overline{\mathbf{N}_1}$ (under source 1 hypothesis). As the number of counts in a bin is Poisson distributed, the variance is equal to the mean. Because there are no correlations between pixel-energy bins, the covariance matrix is diagonal. Thus,
$$\begin{aligned} \overline{\Delta\mathbf{g}} &= \overline{\mathbf{N}_2} - \overline{\mathbf{N}_1} \\ K_1 &= Diag\left(\overline{\mathbf{N}_1}\right) \\ K_2 &= Diag\left(\overline{\mathbf{N}_2}\right) \end{aligned} \tag{25}$$
   where the $Diag$ function turns a vector into a diagonal matrix. The product $\overline{\Delta\mathbf{g}}^t K_{\mathbf{g}}^{-1}$ yields a series of weights $\mathbf{w}^t$ corresponding to the bins of $\mathbf{g}$.
   c. Acquire test statistic for sample data. In the testing stage, we acquire and process each individual particle's pixel and energy value. The observer sums the weights corresponding to each observed particle's pixel-spectral bin and makes its decision based on the test-statistic value,
$$t = \sum_{m=1}^{M} w_m g_m = \sum_{m=1}^{M} w_m \sum_{n=1}^{N_m} 1 = \sum_{m=1}^{M} \sum_{n=1}^{N_m} w_m \quad . \tag{26}$$

2. The second Hotelling observer we created uses the sum of the energy values in a given pixel.
    a. Choose operator $\mathbf{T}$ such that $\mathbf{g}$ contains the sum of the detected energies in each pixel,

$$g_m = \sum_{n=1}^{N_m} E_{n,m} \tag{27}$$

    where $E_{n,m}$ is a subset of the list-mode data and is the energy deposited in pixel m by the n^{th} particle, and N_m is the number of particles that hit a given pixel. Thus, $\mathbf{g}$ consists of 1600 values, one for each pixel.
    b. Record first- and second-order statistics of calibration data. As shown below, to find the appropriate expressions for $\overline{\Delta\mathbf{g}}$ and $K_{\mathbf{g}}$, calibration data will require keeping track of the number of counts in a given pixel, the mean energy in that pixel, and the mean value of the square of the energy in that pixel. The Hotelling-observer weights are found by utilizing the following expressions with (20) and (23).

$$\begin{aligned}
&\overline{\Delta\mathbf{g}} = \left(\overline{\mathbf{N}_2} \odot \overline{\mathbf{E}_2}\right) - \left(\overline{\mathbf{N}_1} \odot \overline{\mathbf{E}_1}\right) \\
&K_1 = Diag\left(\overline{\mathbf{N}_1} \odot \boldsymbol{\sigma}^2_{E1}\right) \\
&K_2 = Diag\left(\overline{\mathbf{N}_2} \odot \boldsymbol{\sigma}^2_{E2}\right) \\
&\boldsymbol{\sigma}^2_{E1} = \overline{\mathbf{E}_1^{\,2}} - \overline{\mathbf{E}_1} \odot \overline{\mathbf{E}_1} \\
&\boldsymbol{\sigma}^2_{E2} = \overline{\mathbf{E}_2^{\,2}} - \overline{\mathbf{E}_2} \odot \overline{\mathbf{E}_2}
\end{aligned} \tag{28}$$

    The $\odot$ symbol denotes the Hadamard product, which is element-by-element multiplication of the elements of two equal-dimension arrays. $\mathbf{E}_1$ (not to be confused with the italicized E above) is a vector of 1600 random variables on the energy deposited in each pixel given source 1 is observed. Each random variable in $\mathbf{E}_1$ has a probability distribution governed by the spectra on that pixel. $\overline{\mathbf{E}_1}$ is a vector containing the mean energy deposited in each pixel when observing source 1, $\boldsymbol{\sigma}^2_{E1}$ is the variance on the energy distribution of each pixel, and $\overline{\mathbf{E}_1^{\,2}}$ is the expectation value of the square of the energy in each pixel. $\mathbf{E}_2$, $\overline{\mathbf{E}_2}$, and $\boldsymbol{\sigma}^2_{E2}$ are comparable vectors for source 2.
    c. Acquire test statistic for sample data. In the testing stage, we multiply each detected energy by the weight corresponding to its pixel number and add the resulting value to the test statistic. The event data can then be forgotten. The classification decision is based on the value of the test statistic,

$$t = \sum_{m=1}^{M} w_m \sum_{n=1}^{N_m} E_{n,m} = \sum_{m=1}^{M} \sum_{n=1}^{N_m} w_m E_{n,m} \quad . \tag{29}$$

2.3 Acquire calibration data required by observer models

Data collection and cataloguing

The most relevant data from the coded aperture imager at this stage of the project is measurements of the INL inspection objects, as well as other unclassified sources and test objects that can be used to develop and characterize the task-based methods. An example of a reconstructed image from such a data run is shown in Figure 18.

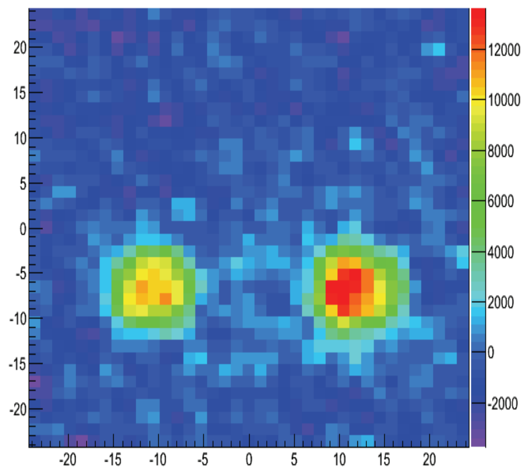

Figure 18 Reconstructed image from data acquired by the fast-neutron coded-aperture imager of a scene containing both IO8 (left) and IO9 (right).

Table 3 is a list of currently identified datasets in our possession that are available for this effort. We are seeking to both identify other data we already possess and data of these inspection objects that we can request from the data owner(s).

Table 3 Catalog (in progress) of fast-neutron imager measurements of inspection objects.

Date	Category	# datasets	Measurement location	Total time [s]
Sep-13	INL test objects	16	INL ZPPR	36120
Sep-13	INL test object scenes	14	INL ZPPR	37920
Sep-13	INL IO pairs	20	INL ZPPR	24000
Sep-13	ZPPR background	1	INL ZPPR	7200
Nov-11	SNL test objects	34	SNL/CA	~1e6

2.4 Compare task performance of approaches

We use receiver-operating-characteristic (ROC) analysis to characterize observer performance. An ROC curve is a plot of the true-positive rate (e.g., the probability of correctly categorizing an object as a TAI) vs the false-positive rate (e.g., the probability of incorrectly categorizing an object as a TAI) as the detection threshold is varied. It is desirable to have an ROC curve with a consistently high true-positive rate with a consistently low false-positive rate.

The area under the ROC curve (AUC) is a summary figure of merit that characterizes the overall performance of an observer. The worst possible observer, which is equivalent to a guess, has an AUC of 0.5. The best observer, which never makes incorrect decisions, has an AUC of 1. We produce an unbiased estimate of the AUC using a technique known as a two-alternative forced-choice (2AFC) experiment. In this experiment, the observer is repeatedly shown pairs of image data, one from each of the two categories, and must classify one as type 1 and the other as type 2. The fraction correct is an unbiased estimate of the AUC that is produced without ever having to compute the entire ROC curve [Barrett and Myers].

The figures in the following sections show the AUC as a function of number of events processed. The AUC is a metric for task performance, which is expected to improve as the information from more and more events is accumulated. Different observers or measurement scenarios can be compared by looking at the number of signal counts required to reach a given value of AUC. In some plots, the AUC vs counts curve asymptotes to a value less than 1.0. In those cases, systematic uncertainties in the observer model, such as those due to nuisance parameters, limit the task performance even for very high-statistics datasets.

Background using GADRAS

We simulate the background radiation signals for our studies using GADRAS (GAmma Detector Response and Analysis Software). The user can create a geometry in GADRAS and come up with template spectra due to certain concentrations of uranium, thorium, potassium, and cosmic muons in the environment. We combine the resulting template spectra for known concentrations for a location to create a local background spectrum. The spectrum below is the gamma background for Buffalo, New York and was used in our simulations.

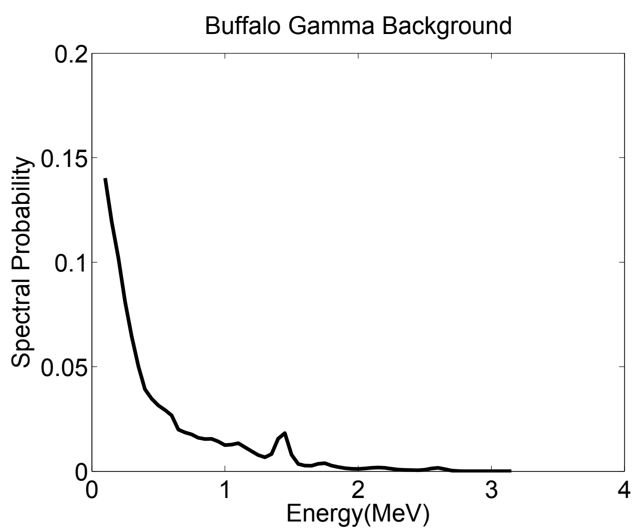

Figure 19 Background spectra used in simulations. This particular background is that seen in the Buffalo, NY area.

Background was added to each pixel of the detector with a strength chosen by the user, and no correlations across pixels were assumed. The detector response for the background template is handled by a custom detector response in GADRAS, while the detector response for the object being simulated is handled in the GEANT simulation. In the large majority of simulations, the background strength was chosen to be weak. We present here the effect of a strong background on observer performance in only one study.

Ideal observer performance

Signal-known-exactly ideal observer with gamma data for one or more orientations

Prior to showing observer performance results, it is helpful to look again at the simulated data. Figure 16 and Figure 17show the spectra and count rate for IO8 vs IO9 as well as the unrotated IO8 vs rotated IO8. It is quite evident that while there is a count rate change in both cases, the spectral difference when comparing the two sources is far greater than the spectral difference in the unrotated vs rotated case. This aspect will be reflected in the following results. All results in this section use gamma events only in the classification task.

Figure 20 is an SKE study that shows the effect a strong background has on observer capability. The SKE model is analogous to (4) – the observer knows the average observed count rate and spectra, and variability in source activity is ignored.

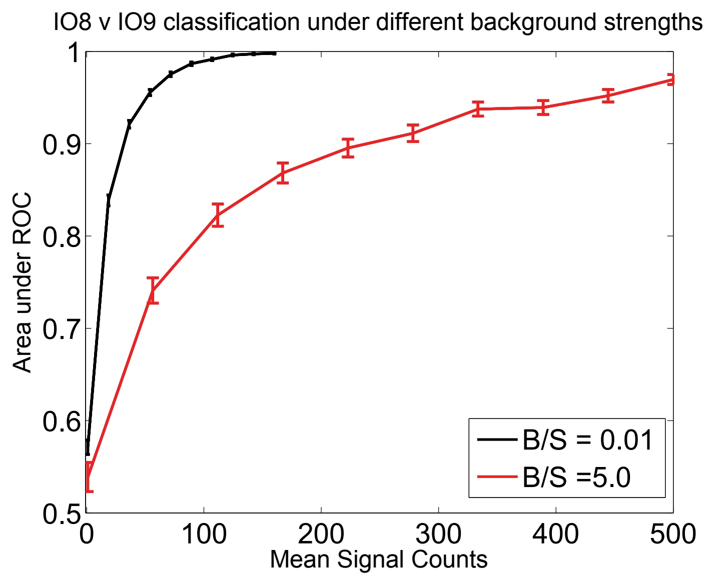

Figure 20 Observer performance under different background strengths. Performed with both calibration and testing data taken from simulation with source face perpendicular to propagation axis.

The observer classifies the two sources very easily if there is no background present, but for a background 5 times stronger than the signal, the observer takes around 10 times more counts to achieve an AUC of 0.9. Figure 21 and Figure 22 show observer performance when both the IO8 and IO9 sample data match the model vs when neither match the model, incorporating both count rate and spectra in the study.

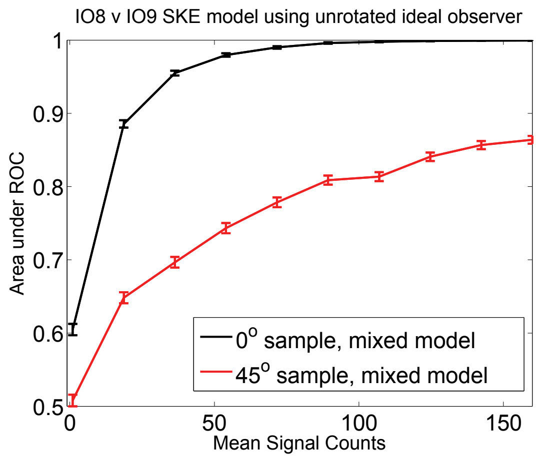

Figure 21 Observer performance under model match/mismatch. This study does not normalize the count rate of the two sources.

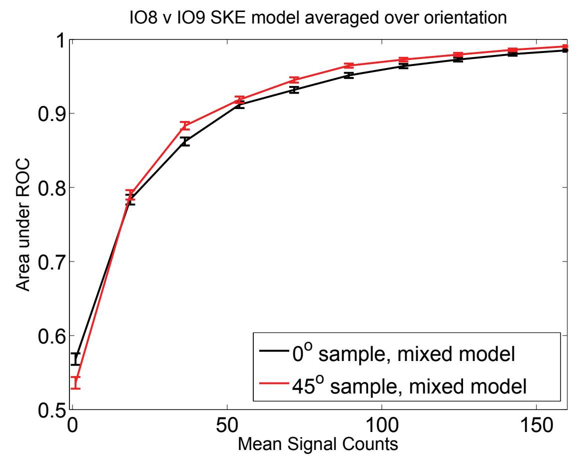

Figure 22 Observer performance when the orientation nuisance parameter is properly accounted for in the observer model. This study does not normalize the count rate of the two sources.

The count rate makes a significant difference here, as IO9's count rate is higher. When the 45 degree sample is classified, the observed count rate in the IO8 sample data rises, and it becomes difficult to differentiate IO8 vs IO9. Figure 22 shows the effect for an observer model averaged over orientation, for which the general expression is (6) and the implementation is (10). The result is the observer classifies the rotated orientation much better. The count rate is normalized for Figure 23 and Figure 24, meaning that the detector hit rate for IO9 is set to the same as IO8. The motivation is to prove that there is more to these models than a simple count-rate study.

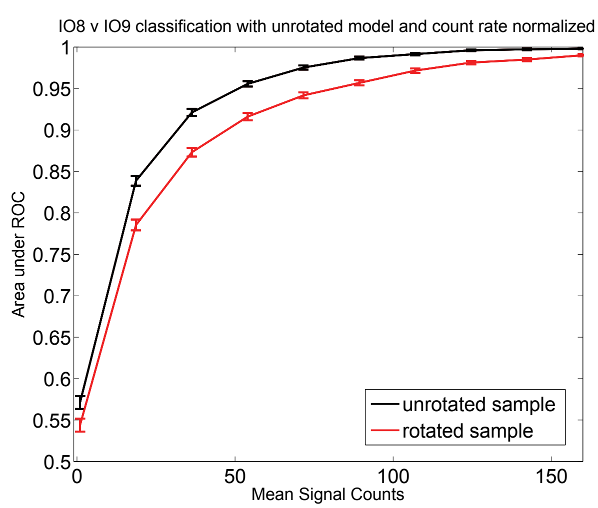

Figure 23 Observer performance using unrotated model. All count rates were set equal.

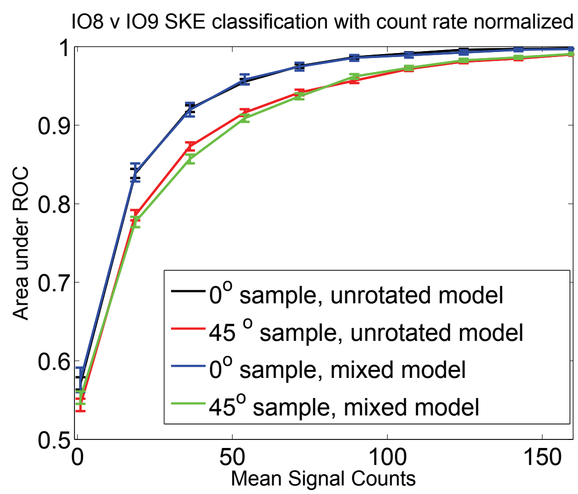

Figure 24 Observer performance for the 4 combinations of model match and mismatch. All count rates were set equal.

It is apparent from Figure 23 that observer performance is strong even when an unrotated model is used to classify data from rotated sources, which is expected given that inspection objects 8 and 9 have notably different spectra. Figure 24 shows what happens to the curves when we move to a mixed model. The unrotated data set is classified equally as well by the mixed model, while the 45-degree sample experiences degraded performance, as the 45-degree model (not shown) actually classified the 45-degree data worse than the 0-degree model. This initially surprising result is due to the rotated and unrotated spectra for a given IO being so similar while spectra for IO8 and IO9 are so different, regardless of orientation. Using only spectral terms in the observer model does not yield a significant change in observer performance when moving to the mixed model, and statistical variation in the testing data can cause minor perturbations in the AUC curves. Comparing observer performance for two sources more similar than IO8 and IO9 would make the count-rate normalization comparison in Figure 23 and Figure 24 look more like those in Figure 21 and Figure 22, as the effect of rotation on the spectra would carry more weight in task performance.

Ideal observer incorporating count rate variability with gamma data

We also analyzed the impact that variable source activity could have on the observer models. The basic concept here is that the sources could have a range of ages, but we only have one measurement (from calibration data) and a guess at the count rate variability to incorporate in the observer models. In Figure 25, the black line shows performance with no variability in count rate (i.e., SKE model), and the red line shows performance with a 20% variability in count rate.

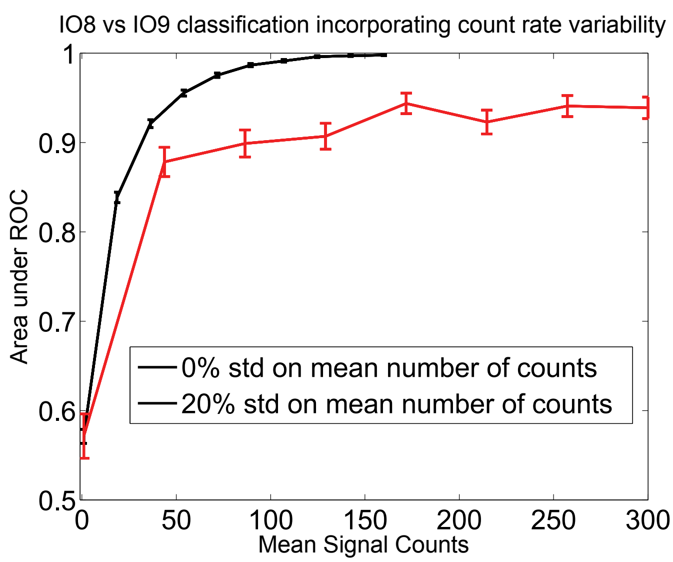

Figure 25 Observer performance incorporating count-rate variability. Performance degrades when a larger spread on activity rate is assumed.

Signal-known-exactly ideal observer with neutron data for one orientation

All prior results have been for gammas only. We also have the capability of analyzing the observer models' performance with neutron data, which turn out to be far less interesting for these inspection objects. Their neutron spectra and count rates are shown in Figure 26, and the observer results are shown in Figure 27.

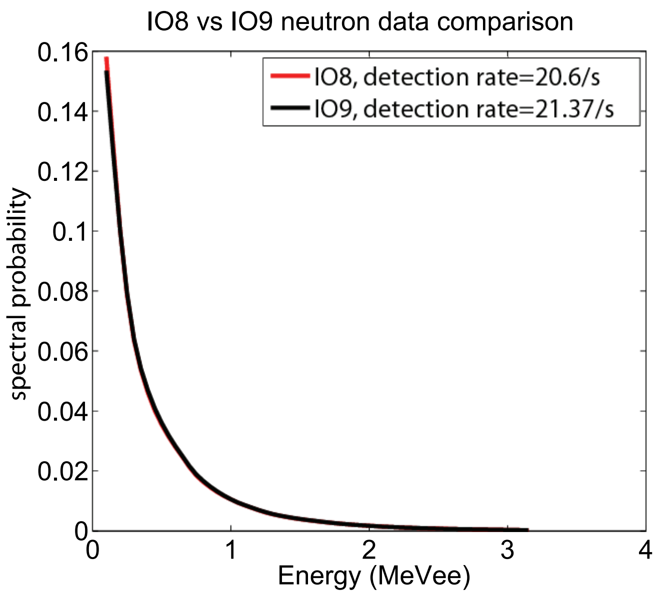

Figure 26 Comparison of IO8 and IO9 neutron spectra and count rate. A 4% difference exists in detected count rate, but negligible difference exists in the two spectra.

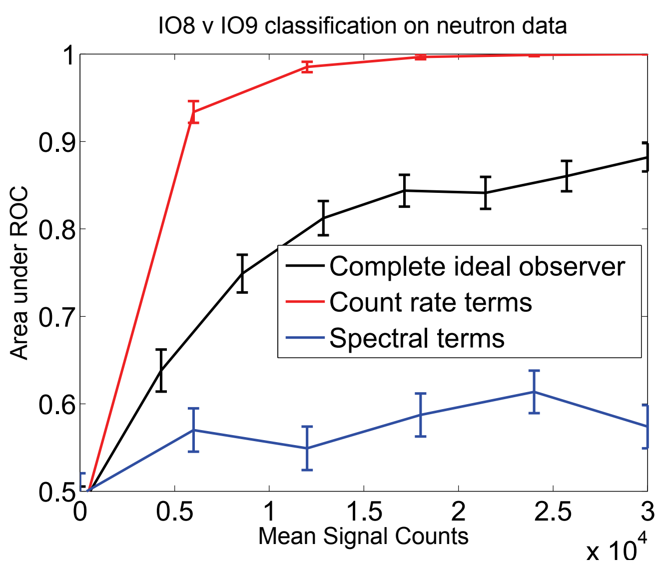

Figure 27 Observer performance on neutron data alone ignoring count-rate variability and observing one orientation. Spectral components are very similar (see Figure 26), and incorporating the spectral component in the ideal observer is seen to degrade performance. This initially surprising result is due to limited statistics in the calibration datasets, which leads to "over-training" of the model. We are updating the observer model implementations to account for calibration statistics.

These observer results deserve a more thorough explanation, as one would naturally assume that incorporating more information should only improve performance. This discrepancy is due to time constraints in the calibration stage. Figure 26 compares the IO8 and IO9 spectra on the entire detector, but the variability in the spectra in a single pixel is far greater. With more calibration data (we are using 21 hours of neutron data, which is a physically realistic number), it would likely be the case that the spectra on each pixel that enter the observer model converge when comparing IO8 and IO9. The end result is that the values returned by the ideal observer's spectral component are unnaturally large and sum to numbers greater than the Poisson component up to around 10,000 signal counts, at which point the Poisson component starts dominating. With more calibration data, the difference between the two spectra on each pixel would be less. Thus, the spectral ratio for a given observed energy would be smaller, and the Poisson component would dominate immediately.

The variability in the GEANT data that we are accepting as "truth data" and have used as calibration data has been considered. We have developed a model to incorporate the expected variability due to time limitations in calibration data acquisition, but we have not yet implemented or tested it.

Signal-known-exactly ideal observer with gamma data and neutron data

While the developed MATLAB models have the capability of including neutron data, the task of classifying IO8 vs IO9 is too easy for any possible benefits to arise. Figure 27 shows that roughly 500 neutron counts are required for an AUC of 0.9 in the count rate observer model. This corresponds to a time of roughly 25 seconds. In comparison, figure 20 shows that only 20 gamma counts are needed for an AUC of 0.9, which corresponds to an acquisition time of about 0.1 seconds. A study of gamma and neutron data would therefore yield the same results as discussed in the ideal observers that use gamma data. We expect utilizing both gamma and neutron data to be very valuable in future tasks, just not this one.

Hotelling observer performance

Hotelling observer with gamma data for one orientation

A Hotelling-observer analysis of gamma rays for the two-source classification study is shown in Figure 28. Once again, two Hotelling observers were analyzed – an observer that stores the number of counts in each energy bin of each pixel, and an observer that only stores the sum of all of the energies in a given pixel.

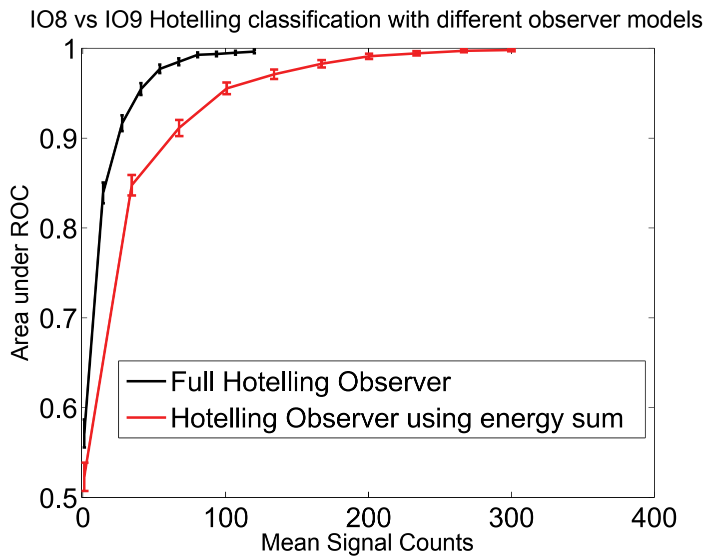

Figure 28 Full Hotelling observer shows better performance than the Hotelling observer using only an energy sum. 100 gamma counts correspond to about 15 ms of real time. Result is from gamma-ray data.

As expected, the full Hotelling observer performs much better than the observer only looking at the sum of energies. However, a sum-over-energy-count observer would inherently require far less information than the full Hotelling observer and, therefore, be more likely to avoid an information barrier. We will continue to explore the space of possible observer models, and in particular investigate models that require even less information.

It should be noted that the models themselves aren't perfect. There are correlations introduced by the energy smearing in the detector-response stage that cause crossover among neighboring energy bins. These correlations are ignored for convenience in the observer studies, but they exist. Incorporating these correlations would improve performance, though it is expected the difference would be small.

3. FUTURE WORK

3.1 Validation of simulation models

We obtained previously acquired high-purity germanium (HPGe) and neutron-multiplicity data for IO9 and IO10. We are also currently inquiring about data taken of other inspection objects from the same measurement campaign, as well as more data from those objects. The HPGe data was taken with a detector of known response (i.e., Detective-EX100) and experimental setup, which will allow us to compare simulation results to the acquired data as a validation of the simulations. Since the gamma spectrum contains information on not just the isotopes present in the object, but also the effects of attenuation, this allows us to help validate geometric construction of our models, as well as the physics of the simulation. The gamma data we have for IO10 was taken "face on" of the object from two different distances, as well as from a corner, and both with and without a moderator present. The data for IO9 is face on, and only from the HPGe detector (and its onboard He-3 detector), but both with and without a moderator present. Since the simulation of the IO objects is split into two parts, the simulation of particles exiting the object, and then the propagation to, and in the detector, we can re-use the results of the first stage of simulation and add a simulation of the propagation to, and response of, the HPGe detector with minimal amount of work and reduced CPU time.

The quantities we expect to validate will include the overall gross count rates of both neutrons and gammas; the relative attenuation of gamma lines; the presence of all expected gamma lines; the shape and amplitude of the gamma continuum; the effect of the moderator on the detected signal; as well as the relative differences between the measured locations. We expect to perform this study in the immediate future.

3.2 Nuisance parameters

As discussed in the above section, the orientation study has already begun. It is possible that some nuisance parameters will only prove to be a minor hindrance when classifying sources, meaning they can be ignored, but others will undoubtedly need to be accounted for. Effects from some of the sources may be handled by simple scaling of distributions or quantities according to statistical distributions, however, other parameters will require performing simulations to calculate the effects at the various values if the parameters. Now that the simulations are running on the Sandia clusters, we are currently performing a location study to determine how large of an effect the location and orientation plays on the classification.

Given the number of nuisance parameters that may need to be simulated in order to calculate their effect, as well as the need to potentially vary multiple nuisance parameters at a time, the number of points which would need to be simulated can easily grow to intractable levels since each nuisance parameter may multiplicatively increase the necessary simulation scenarios. To mitigate this issue we will investigate *template morphing* [Read] as a means to not just interpolate between simulated values of a nuisance parameter, but also to interpolate the effects from multiple nuisance parameters onto the template, bringing the number of points necessary to be simulated from a multiplicative scenario, to an additive one. Figures 29 and 30 describe the mechanics if the morphing techniques we plan to investigate to reduce the number of simulations we will have to perform.

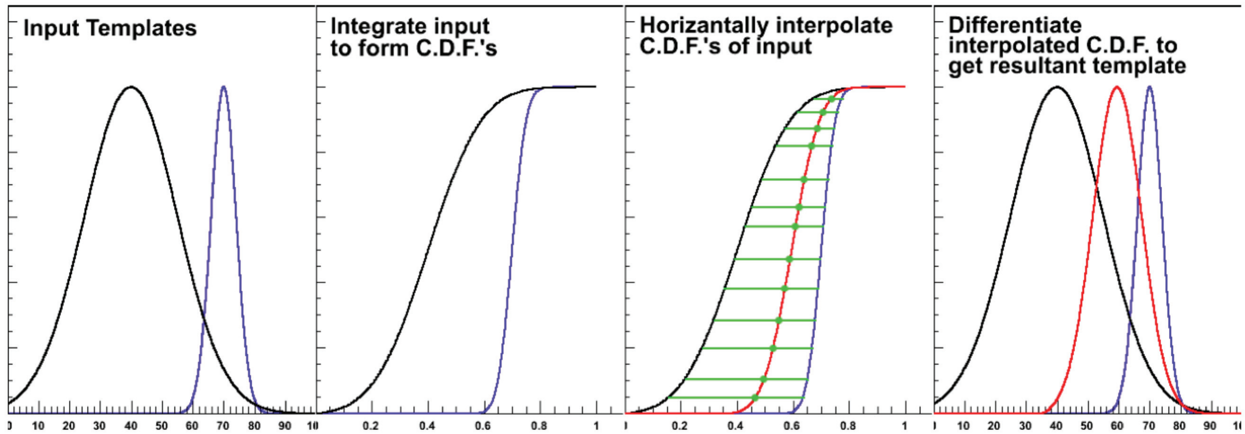

Figure 29 The basic mechanics of horizontal template morphing. Given two input templates (e.g., results of simulations at two values of a nuisance parameter), the cumulative distribution functions are formed for each distribution, and then those are interpolated between with the result differentiated to form the final interpolated distribution. This method can be extended to append the resulting "delta" of a morph onto a separate (typically nominal) distribution to allow accounting for multiple nuisance parameters; this is referred to as compound morphing.

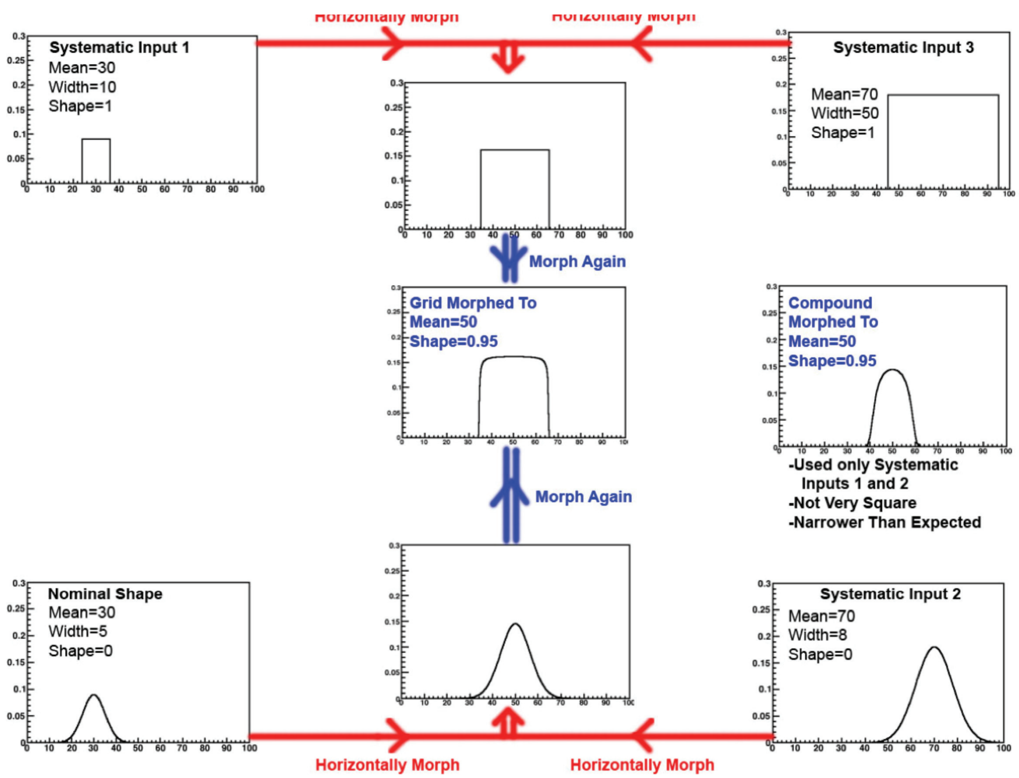

Figure 30 An illustration of grid morphing, where two nuisance parameters (here the location and shape of the distribution) are independently varied and simulated, the results of which are combined through horizontal morphing. This method provides superior results to compound morphing (see Figure 29) for correlated variable or correlated differences.

3.3 Pulse-shape discrimination

Pulse-shape discrimination (PSD) is currently assumed to be perfect, but this is not physically realistic. An accurate model needs to be obtained and incorporated into the GEANT simulation in post processing. The observer models will then need to use what knowledge we have of the detectors PSD capabilities to best classify the sources given the errors in particle classification.

3.4 Observer-model development

Continue to develop observer models that act on testing data in list mode that use less sensitive information. We hope to develop an observer that is an analogue to the chi-squared hypothesis testing observer and to analyze the channelized Hotelling observer.

4. REFERENCES

S. Agostinelli et al. (the GEANT4 collaboration), GEANT4 – a simulation toolkit, *Nucl. Instr. Meth. Phys. Res. A*, **506** (3),250-303, July 2003.

J. Allison et al. (the GEANT4 collaboration), GEANT4 developments and applications, *IEEE Trans. Nucl. Sci.*, **53** (1), 270-278, February 2006.

H. H. Barrett and K. J. Myers, *Foundations of Image Science*. John Wiley & Sons, Hoboken, New Jersey, 2004.

P. A. Hausladen, M. A. Blackston, E. Brubaker, D. L. Chichester, P. Marleau, and R. J. Newby, Fast-neutron coded-aperture imaging of special nuclear material configurations, *Proc. of 53rd INMM Annual Meeting*, Orlando, Florida, 2012.

M. A. Kupinski, J. W. Hoppin, E. Clarkson, and H. H. Barrett, Ideal-observer computation in medical imaging with use of Markhov-chain Monte Carlo techniques*, J. Opt. Soc. Am. A. Opt. Image. Sci. Vis.*, **20** (3), 430-438, March 2003.

R. Neibert, J. Zabriskie, C. Knight, and J. L. Jones, *Passive and Active Radiation Measurements Capability at the INL Zero Power Physics Reactor (ZPPR) Facility*, Idaho National Laboratory report INL/EXT-11-20876, Idaho Falls, Idaho, December 2010.

A. L. Read, Linear interpolation of histograms, *Nucl. Instr. Meth. Phys. Res. A*, **425** (1-2), 357-360, April 1999.

J. K. Shultis and R. E. Faw, *An MCNP Primer*. Kansas State University, Manhattan, Kansas, 2013.

DISTRIBUTION

#	MailStop	Name	Org. Number
1	MS0899	Technical Library	9536 (electronic copy)

Sandia National Laboratories