Real-Time Compton Imaging for the GammaTracker Handheld CdZnTe Detector

Mitchell J. Myjak, *Member, IEEE*, and Carolyn E. Seifert, *Member, IEEE*

***Abstract*—We are currently developing a handheld radioisotope identifier containing 18 position-sensitive CdZnTe crystals. In addition to isotope identification, the device performs basic Compton imaging to determine the location of suspected sources. This paper presents two computationally efficient algorithms for this purpose. The first algorithm traces individual Compton cones onto the unit sphere, whereas the second algorithm computes the intersection of two Compton cones and the unit sphere. Simulations demonstrate that the algorithms are suitable for determining the directionality, even with features such as uncertainty calculations omitted. The one-cone algorithm works more efficiently at high count rates, but the two-cone algorithm generates fewer image artifacts.**

***Index Terms*—Cadmium zinc telluride, Compton camera, field programmable gate arrays, image reconstruction, nuclear imaging.**

I. INTRODUCTION

COMPTON imaging is a well-known method for deducing the origin of incoming gamma rays based on the observed interactions within the detector material. Suppose a gamma ray γ undergoes a sequence of interactions $i = 1, 2, \ldots, n$ in which it deposits energy E_i at location $\mathbf{d}_i = (d_{ix}, d_{iy}, d_{iz})$. Assuming that all events are captured inside the detector, the incident energy E_0 equals the sum of the deposited energies:

$$E_0 = \sum_{i=1}^{n} E_i. \tag{1}$$

The deflection angle θ for the first interaction can be computed using the Compton scatter formula:

$$\cos\theta = 1 - \frac{m_e c^2 E_1}{E_0(E_0 - E_1)}. \tag{2}$$

Here $m_e c^2 = 511.00$ keV is the rest mass energy of an electron. Although the incident direction of the gamma ray cannot be measured directly, it theoretically falls on a cone with opening angle θ and normalized axis direction

$$\mathbf{c} = \frac{\mathbf{d}_1 - \mathbf{d}_2}{\|\mathbf{d}_1 - \mathbf{d}_2\|}. \tag{3}$$

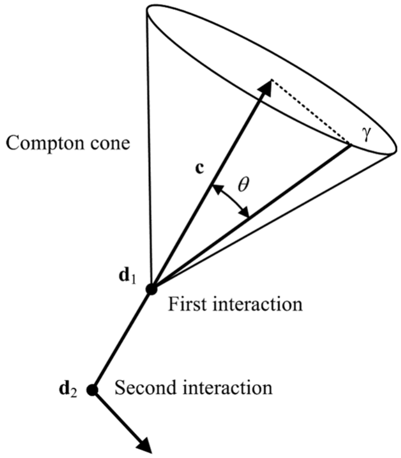


Fig. 1. Diagram of Compton imaging.

Fig. 1 depicts a Compton cone for a single gamma ray. Each detected gamma ray is expected to have a different axis $\mathbf{c}$ and scatter angle θ, resulting in different reconstructed Compton cones. Intersecting the cones for multiple gamma rays reveals the probable location of the radiation source.

In many applications, the size of the detector is negligible compared to the distance from the detector to the source. Hence, one common practice is to position all the Compton cones at the origin and project them onto a two-dimensional surface, such as a plane or a sphere. We refer to this approach as the far-field approximation [1]. The resulting image indicates the likely direction to the radiation source but contains no depth information. Computationally, using the far-field approximation is much simpler than intersecting arbitrary cones in three dimensions. However, algorithms for this case have also been developed [2].

We are currently developing a handheld radioisotope identifier for national security applications that combines gamma-ray spectroscopy with basic Compton imaging. This instrument, named GammaTracker, contains 18 CdZnTe crystals arranged in a two-level 3×3 array. Each crystal is $15 \times 15 \times 10\ \mathrm{mm}^3$ and features an 11×11 anode grid that provides position sensitivity in two dimensions. The third coordinate is determined by measuring the time difference between the anode and cathode pulses. The detector pitch is 22 mm laterally with a 20-mm spacing between layers to accommodate the readout electronics. This arrangement provides for 40.5 cm^3 of active detector volume. The outputs of each crystal are preprocessed by a specialized ASIC, converted to digital form, and loaded onto an embedded computing platform for further processing. These front-end electronics are based on a benchtop prototype developed by the University of Michigan [3]. The

Manuscript received April 20, 2007; revised October 18, 2007. This work was supported by the U.S. Department of Energy Office of Nonproliferation Research and Development under Contract DE-AC06-76RL01830.

The authors are with Pacific Northwest National Laboratory, Richland, WA 99352 USA (e-mail: mitchell.myjak@pnl.gov; carolyn.seifert@pnl.gov).

Digital Object Identifier 10.1109/TNS.2007.915008

embedded computing platform combines a microprocessor with a field-programmable gate array (FPGA). This arrangement permits the system to offload the most intensive signal processing tasks onto the FPGA.

The two-part processing platform implements two key functions: identifying key isotopes present in the vicinity, and indicating the origin of the detected gamma rays. The implementation of the latter function forms the subject of this paper. The general problem encompasses several tasks:

- energy correction and discrimination;
- sequence order reconstruction;
- Compton imaging;
- directionality calculations.

These tasks must run at or near real time to provide useful information to the operator. However, the instrument has limited computational capabilities and relies on battery power when operating in the field. Hence, algorithms that have an efficient FPGA implementation are essential for this project.

One algorithm proposed by Lackie *et al.* simplifies the imaging calculations by intersecting the bounding rectangles for each circular projection [4]. As the intersection of two rectangles is merely another rectangle, the algorithm can progressively refine the current estimate. However, the method currently assumes that only one source is present, and that the number of spurious cones is insignificant. These assumptions become invalid for weak radiation sources with high background rates.

In this paper, we present two algorithms for Compton imaging that are suitable for real-time implementation on embedded hardware. Both algorithms work with arbitrary numbers of sources and background rates. The first algorithm, described in Section III, projects individual Compton cones onto a sphere of unit radius. This method uses fairly standard imaging techniques, but performs them in an efficient manner. The second algorithm, described in Section IV, calculates the intersections of *two* Compton cones with the unit sphere. This method is optimized for calculating the directionality. We compare the two methods in Section V and illustrate some simulated images. Finally, Section VI gives some concluding remarks.

II. EVENT AND SEQUENCE PROCESSING

In order to perform Compton imaging, some processing of the raw energy and position data from each gamma-ray event is required. First, the true energies and positions of every interaction in the detector array must be determined. Then, it is necessary to establish the correct sequence of events. These steps, while not unique to this work, are included in this paper for completeness.

A. Energy Correction and Discrimination

The front-end electronics reports four values for each interaction i within a crystal: the anode charge amplitude A_i, the anode pixel coordinates (d_{ix}, d_{iy}), and the time difference τ_i between the anode and cathode pulses. The depth of the event d_{iz} is roughly proportional to τ_i. To account for second-order effects, we assume that d_{iz} follows the piecewise linear approximation

$$d_{iz} = \alpha_z \tau_i + \beta_z,\ \tau_{low} \leq \tau_i \leq \tau_{high}, \tag{4}$$

where α_z and β_z are the gain and offset of the line segment. Likewise, we assume that the deposited energy E_i follows

$$E_i = \chi_E(\alpha_E A_i + \beta_E),\ A_{low} \leq A_i \leq A_{high}, \tag{5}$$

where α_E and β_E are as before, and χ_E is an adjustment for crosstalk between multiple events. These piecewise linear corrections encompass a wide range of effects, including material variations, geometric asymmetries, and so forth. Parameters α_z and β_z are functions of the pixel coordinates (d_{ix}, d_{iy}). Parameters α_E and β_E depend on the three-dimensional event location (d_{ix}, d_{iy}, d_{iz}) and the current temperature. Finally, parameter χ_E depends on the depth d_{iz} and the distance between multiple interactions. Each parameter must be determined individually via calibration with a known source.

To simplify the calibration process, the system divides the z-axis into 20 virtual layers. Thus, the crystal effectively contains an $11 \times 11 \times 20$ array of voxels (three-dimensional pixels). The system then stores α_z and β_z for each pixel and time interval, α_E and β_E for each voxel and amplitude interval, and χ_E for each depth and approximate distance. Since the temperature changes relatively slowly, the system can adjust α_E and β_E as necessary.

Once the algorithm determines the energy of each interaction, and hence the estimated incident energy E_0, the algorithm can apply windowing techniques to select gamma rays of certain energies. For example, the operator might want to mask a natural radiation source to search for other, hidden threats. This energy discrimination does not involve significant computational effort.

B. Sequence Order Reconstruction

The front-end electronics do not have sufficient timing resolution to capture individual events as they occur within the detector. Hence, the imaging algorithm must take the list of events reported by the electronics and arrange them in time order. Researchers have proposed several heuristics for this task. For two events, the most popular technique uses the energy information alone to determine the most likely sequence order [5]. For three events, one approach starts with an assumption about the final interaction and reconstructs the track backwards [6]. An alternative method uses Compton kinematics to reject non-physical sequences and calculates the probability of observing the remaining sequences [7]. A third approach applies Bayesian methods to compare a given sequence order against a multidimensional simulated dataset [8]. Finally, the algorithm can compare the actual scatter angles with the angle predicted from the Compton scatter formula [5], [9]. We use the same techniques as previous work [1] for reconstructing sequences of two or three interactions.

For $n = 2$, the algorithm determines which event comes first by comparing the deposited energies. If the incident energy E_0 is less than a certain threshold E_t, the events are ordered so that

$E_1 < E_2$. Otherwise, the algorithm takes $E_1 > E_2$. This empirical technique makes sense because gamma rays that deposit most of their energy in the initial scatter are more likely to be absorbed in the second interaction. One can derive the optimal value of E_t via a simulation package such as GEANT4 [10]. For example, $E_t \approx 400\ \mathrm{keV}$ for a single 2.25- cm^3 CdZnTe crystal [1].

For $n = 3$, the algorithm takes each permutation of the three events and examines the deflection angle ϕ of the second interaction. This value can be calculated in two different ways. Using the Compton scatter formula,

$$\cos\phi_E = 1 - \frac{m_e c^2 E_2}{(E_2 + E_3)E_3}. \tag{6}$$

Using the event locations within the detector,

$$\cos\phi_d = \frac{\mathbf{d}_3 - \mathbf{d}_2}{||\mathbf{d}_3 - \mathbf{d}_2||} \cdot \frac{\mathbf{d}_2 - \mathbf{d}_1}{||\mathbf{d}_2 - \mathbf{d}_1||}. \tag{7}$$

Permutations that produce invalid values for the cosines are discarded. The algorithm then selects the permutation that minimizes the absolute difference

$$\sigma = \left|\cos\phi_E - \cos\phi_d\right|. \tag{8}$$

For greater accuracy, one could compute the uncertainties in energy in position and refine the metric to reflect those values [9]. We decided against this approach in favor of simplicity.

For $n > 3$, the algorithm could use the same technique to check every permutation of three events and piece together the most likely sequence. However, this method involves significant computational effort, and the results become less reliable as n increases. GEANT4 simulations of the 18-crystal array with a 662-keV source indicate that only 9.4% of the sequences contain four or more events. Hence, we do not attempt to reconstruct these sequences.

III. ONE-CONE ALGORITHM

Unlike traditional two-plane Compton detectors such as [11]–[13], the CdZnTe crystal array used in the GammaTracker instrument has good angular resolution in all directions: the useful field of view is not limited to sources in front of the detector array. The imaging algorithms described in this paper project the Compton cones onto a sphere of unit radius. This approach has become practical with the expanded use of three-dimensional position-sensitive detectors [1]. The technique is sometimes referred to as " 4π Compton imaging" since a source at any angle (within all of 4π steradians) relative to the detector can be reconstructed. The following discussion summarizes the first algorithm, which traces individual cones onto the sphere. An estimated angular uncertainty determines the width of the trace.

A. Compton Imaging

The objective of the one-cone algorithm is to trace individual Compton cones onto the unit sphere. The GammaTracker instrument is not intended to image close-up sources, so we use the far-field approximation described in the introduction. Hence, the projection of a Compton cone onto the unit sphere is simply a circle. The algorithm begins by computing the cone axis direction $\mathbf{c}$ and opening angle θ as described in the introduction. Referring to Fig. 2, let $\mathbf{s}$ be a direction vector from the origin to an arbitrary point on the sphere. Define α as the angle between $\mathbf{c}$ and $\mathbf{s}$. Since $\mathbf{c}$ and $\mathbf{s}$ are unit vectors,

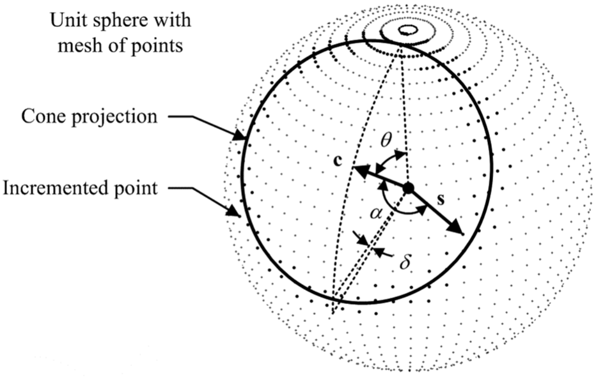


Fig. 2. Projection of Compton cone onto the unit sphere.

$$\cos\alpha = \mathbf{c} \cdot \mathbf{s}. \tag{9}$$

Then the circle can be described as the set of points on the sphere for which $\alpha = \theta$.

To create the image, the algorithm divides the sphere into a mesh of r points, and stores an integer for each point that indicates how many cones intersect it. The figure above illustrates this concept. The algorithm then traces the cone onto the image by finding all points on the mesh for which

$$\theta - \delta \le \alpha \le \theta + \delta. \tag{10}$$

The difference angle δ compensates for the fact that points on the mesh will not coincide with the projection in general. One could relate this value to the uncertainties in energy and position [1], [5], [14]. We simply set δ to a constant proportional to the angle between adjacent points on the mesh.

Implementing (10) directly is not computationally efficient on an FPGA since only the cosines of α and θ are known. However, we can assume without loss of generality that α and θ fall within the range $[0, \pi]$. We can also limit δ to the range $[0, \pi/2]$ without imposing undue restrictions on the imaging algorithm. The inequality then splits into three cases:

$$\left\{\begin{array}{ll} 0 \le \alpha \le \theta + \delta, & \theta - \delta \le 0 \\ \theta - \delta \le \alpha \le \pi, & \theta + \delta \ge \pi \\ \theta - \delta \le \alpha \le \theta + \delta, & \text{otherwise} \end{array}\right\}. \tag{11}$$

Simplifying,

$$\left\{\begin{array}{ll} \alpha \le \theta + \delta, & \theta \le \delta \\ \theta - \delta \le \alpha, & \theta \ge \pi - \delta \\ \theta - \delta \le \alpha \le \theta + \delta, & \text{otherwise} \end{array}\right\}. \tag{12}$$

Now the terms on both sides of every inequality fall within [0, π]. Since the cosine function monotonically decreases over this range, we can transform the inequalities to

$$\left\{\begin{array}{ll}\cos\alpha \geq \cos(\theta+\delta), & \cos\theta \geq \cos\delta \\ \cos(\theta-\delta) \geq \cos\alpha, & \cos\theta \leq -\cos\delta \\ \cos\alpha \geq (\theta+\delta)\,\mathrm{AND} & \\ \cos(\theta-\delta) \geq \cos\alpha, & \mathrm{otherwise}\end{array}\right\}. \tag{13}$$

Thus, to determine whether a point on the mesh is near the Compton cone, the algorithm can evaluate the expression

$$\begin{gathered}\{\cos\alpha \leq \cos(\theta-\delta)\ \mathrm{OR}\ \cos\theta \geq \cos\delta\}\mathrm{AND} \\ \{\cos\alpha \geq \cos(\theta+\delta)\ \mathrm{OR}\ \cos\theta \leq -\cos\delta\},\end{gathered} \tag{14}$$

which reduces to

$$\begin{gathered}\{\mathbf{c}\cdot\mathbf{s} \leq \cos\theta\cos\delta + \sin\theta\sin\delta\ \mathrm{OR}\ \cos\theta \geq \cos\delta\}\ \mathrm{AND} \\ \{\mathbf{c}\cdot\mathbf{s} \geq \cos\theta\cos\delta - \sin\theta\sin\delta\ \mathrm{OR}\ \cos\theta \leq -\cos\delta\}.\end{gathered} \tag{15}$$

The computational complexity of this expression is very low: two multiplications and three additions for $\mathbf{c}\cdot\mathbf{s}$, along with a few comparison and logic operations. The terms that depend on θ and δ need only be computed once for each cone. Notice that the algorithm does need to compute $\sin\theta$ using

$$\sin\theta = \sqrt{1-\cos^2\theta}. \tag{16}$$

To update the image, we simply increment the values corresponding to the points that satisfy the condition (15). One could also count the number of points k that meet the condition and add a factor of $1/k$ to the corresponding values. This normalization would ensure that all Compton cones contribute equally to the image in summation, but we do not use this technique in the one-cone algorithm.

One could also reduce the number of computations required by the one-cone algorithm by computing the latitude and longitude bounds of the circular projection. Rather than checking every point, the algorithm could iterate over all points in the spherical rectangle inside these bounds. Computing the necessary limits involves some trigonometry and becomes less efficient for projections that contain one of the poles. We decided not to implement this alternative on the FPGA to save area.

B. Directionality Calculations

The last step in the one-cone algorithm is to calculate the directionality to suspected sources. Rather than having the operator interpret the peaks in the raw image, the instrument will display a horizontal azimuth and possibly a vertical altitude. One can calculate these values in several ways. A simple method is to sum along lines of altitude to create a plot of intensity versus azimuth. Another approach is to identify peaks in the image and compute the centroid of each peak. Neither technique requires a large amount of computation.

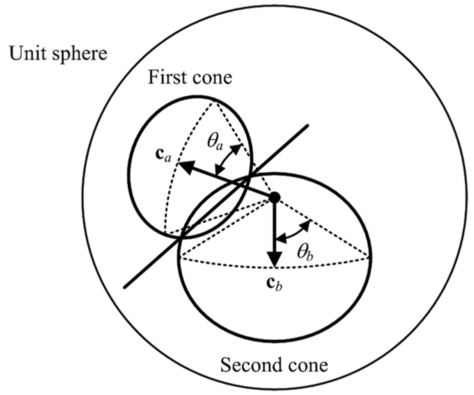


Fig. 3. Intersection of two Compton cones with the unit sphere.

One important consideration for the target application is how long to keep the projected Compton cones on the image. The algorithm could clear all the data every 30 or 60 seconds and start afresh, but then the directionality display would update infrequently. A better alternative is a sliding-window buffer, in which individual cones are discarded after a set time. The instrument could also use an accelerometer to detect when the operator is moving, and discard sequences more rapidly in that case.

IV. TWO-CONE ALGORITHM

Handheld radiation identifiers can use the one-cone algorithm described above to produce a conventional image of the environment surrounding the operator. This image provides a comprehensive probability density function about nearby sources. In many cases, however, imaging may not be necessary—directionality may be sufficient. The Gamma Tracker instrument in particular merely needs to tell the user which way to walk next. If directionality is the objective, algorithms other than the one-cone approach may be viable and more efficient.

With this in mind, we have developed a second algorithm that provides directionality information more directly. As described below, the algorithm works by calculating the intersection points between two Compton cones and the unit sphere. This approach eliminates the need to trace individual cones onto the sphere. The energy correction and sequence reconstruction steps are identical to those in the one-cone algorithm.

A. Compton Imaging

Rather than iterating over points on the unit sphere, the two-cone algorithm iterates over the last m sequences captured by the detector. Let $\mathbf{c}_a$ and θ_a be the parameters of the Compton cone for one of the previous sequences, and let $\mathbf{c}_b$ and θ_b be the corresponding parameters for the current sequence. Projecting these two cones onto the unit sphere might produce the result in Fig. 3. Assuming the two cones do not coincide, they mutually intersect the sphere at no more than two points.

Now the intersection of the unit sphere with one Compton cone is equivalent to the intersection of the unit sphere with a certain plane. This plane is normal to the cone axis $\mathbf{c}$, and located a distance $\cos\theta$ away from the origin. For two Compton cones, the corresponding planes will intersect in a line, if they intersect at all. The intersection of this line with the unit sphere equals the mutual intersection of the two Compton cones with the unit sphere.

The two-cone algorithm begins with the equations of the two planes in point-normal form:

$$\begin{aligned}\mathbf{c}_a \cdot \mathbf{u} &= \cos\theta_a \\ \mathbf{c}_b \cdot \mathbf{u} &= \cos\theta_b.\end{aligned} \tag{17}$$

Here $\mathbf{u}$ is an arbitrary position vector. All vectors are represented in Cartesian coordinates with x-, y-, and z-components, so one could rewrite the equations above as

$$\begin{aligned}c_{ax}u_x + c_{ay}u_y + c_{az}u_z &= \cos\theta_a \\ c_{bx}u_x + c_{by}u_y + c_{bz}u_z &= \cos\theta_b.\end{aligned} \tag{18}$$

To determine the intersection line, the algorithm needs to find a point $\mathbf{p}$ that lies on both planes, and hence on the line. The simplest method is to solve the two equations in (17) with the z-coordinate set to zero. Using Cramer's rule,

$$\begin{aligned}p_x &= \frac{c_{by}\cos\theta_a - c_{ay}\cos\theta_b}{c_{ax}c_{by} - c_{ay}c_{bx}} \\ p_y &= \frac{c_{ax}\cos\theta_b - c_{bx}\cos\theta_a}{c_{ax}c_{by} - c_{ay}c_{bx}} \\ p_z &= 0.\end{aligned} \tag{19}$$

The notation c_{ax} refers to the x-component of vector $\mathbf{c}_a$. This method fails in the case when either cone points along the z-axis, since its x- and y-components are both zero. In a single CdZnTe crystal, this situation is invalid anyway, since the detector cannot resolve two events that occur within the same pixel. In the two-level crystal array, it is possible for two events to occur, one in each layer, at the same lateral position. For this case, one can solve the equations in (17) with the x-coordinate or the y-coordinate set to zero. GEANT4 simulations indicate that less than 0.1% of 2-event or 3-event sequences would require this special treatment, given a 662-keV source at a 30° angle relative to the z-axis. Thus, we discard these events altogether with negligible loss of system efficiency.

Given $\mathbf{p}$, the algorithm can determine the direction $\mathbf{L}$ of the intersection line by taking the cross product $\mathbf{c}_a \times \mathbf{c}_b$:

$$\begin{aligned}L_x &= c_{ay}c_{bz} - c_{az}c_{by} \\ L_y &= c_{az}c_{bx} - c_{ax}c_{bz} \\ L_z &= c_{ax}c_{by} - c_{ay}c_{bx}.\end{aligned} \tag{20}$$

Then the equation of the line is

$$\mathbf{u} = \lambda\mathbf{L} + \mathbf{p}, \tag{21}$$

where λ represents an arbitrary real number. Since the unit sphere is described by

$$\mathbf{u} \cdot \mathbf{u} = 1, \tag{22}$$

intersecting the line with the sphere produces

$$(\lambda\mathbf{L} + \mathbf{p}) \cdot (\lambda\mathbf{L} + \mathbf{p}) = 1, \tag{23}$$

which expands to the quadratic equation

$$(\mathbf{L} \cdot \mathbf{L})\lambda^2 + 2(\mathbf{L} \cdot \mathbf{p})\lambda + (\mathbf{p} \cdot \mathbf{p} - 1) = 0. \tag{24}$$

Solving this equation for λ using the quadratic formula and substituting back in (21) yields the desired intersection points:

$$\mathbf{u} = \frac{-(\mathbf{L} \cdot \mathbf{p}) \pm \sqrt{(\mathbf{L} \cdot \mathbf{p})^2 - (\mathbf{L} \cdot \mathbf{L})(\mathbf{p} \cdot \mathbf{p} - 1)}}{(\mathbf{L} \cdot \mathbf{L})}\mathbf{L} + \mathbf{p}. \tag{25}$$

The algorithm records these points in a buffer and then repeats the process for another previous sequence, again determining its intersection points with the last-measured sequence.

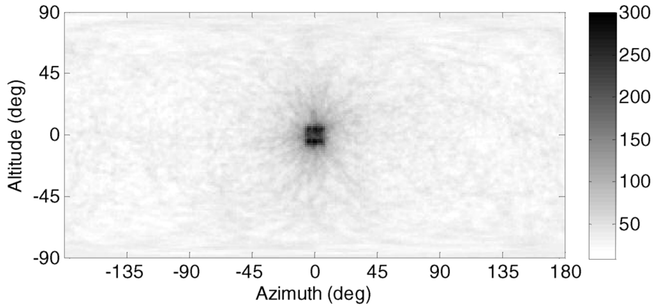
B. Energy Discrimination

A direct implementation of the two-cone algorithm would compute the intersection points for all possible pairs of Compton cones. If the detector stored m valid sequences in memory, the two-cone algorithm would intersect $m(m+1)/2$ pairs of cones in the worst case. Each pair would produce zero or two intersection points (disregarding the cases where both cones coincide or intersect the sphere at the same point). However, sequences that have different incident energies E_0 are less likely to come from the same source than sequences with similar incident energies. The intersection points between cones from different sources would only add unnecessary clutter to the result. Thus, our implementation of the two-cone algorithm only intersects cones whose incident energies differ at most by a fixed percentage ε.

Not only does this approach reduce the total computational effort, but the inherent energy discrimination improves the ability to locate weak sources in the presence of background. Clearly, the best results occur when the energy spectrum has definite full-energy peaks. One could extend this approach to intersect cones with different energies that correspond to the same isotope or decay chain.

This energy discrimination method is similar to—but much simpler than—the "data cube" image structure originally proposed in [15]. Rather than binning Compton cones into different images based on the deposited energy (thus requiring a large 3D energy-image data structure), the algorithm can intersect only those cones with similar deposited energies, and integrate the resulting intersection points in a single image. Thus, only a 2D image data structure is required to achieve a significant degree of energy discrimination, conserving much-needed memory on instruments such as GammaTracker.

(a) One-cone algorithm



(b) Two-cone algorithm

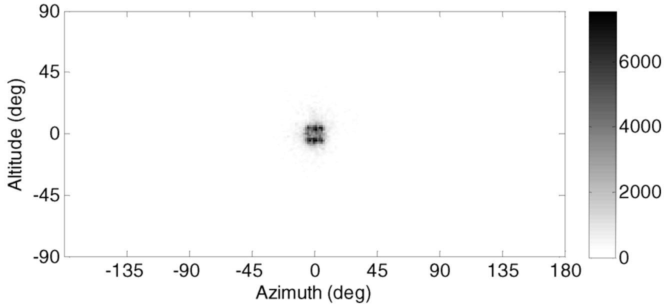
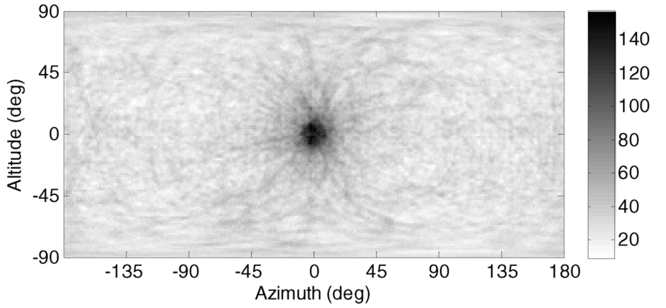


Fig. 4. Simulation of 2000 two-event and three-event sequences from a 662-keV point source with perfect position and energy resolution.

(a) One-cone algorithm



(b) Two-cone algorithm

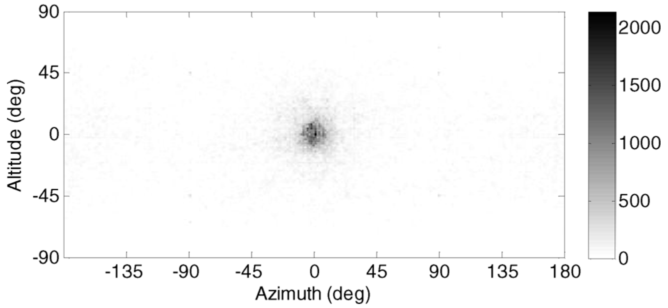


Fig. 5. Repeat of previous simulation with finite position and energy resolution.

C. Directionality Calculations

The two-cone algorithm offers a number of ways to compute the directionality. At least, one could plot each intersection point on a spherical surface and use the same peak-finding techniques as the one-cone algorithm. One could also skip the imaging step completely, grouping the intersection points into clusters and finding the centroid of each cluster. Perhaps the simplest approach, and one that is appropriate for the handheld unit, is to compute the horizontal bearing of each intersection point. The system would then display the directions that corresponded to the most points. While simple, this method does neglect the vertical information.

V. ANALYSIS

In this section, we analyze the performance and efficiency of the two Compton imaging algorithms. First, we illustrate the images generated for a variety of simulated sources. We then present the estimated directional error as a function of number of events. Finally, we estimate the resources required to execute the algorithms on embedded hardware.

A. Ideal Simulations

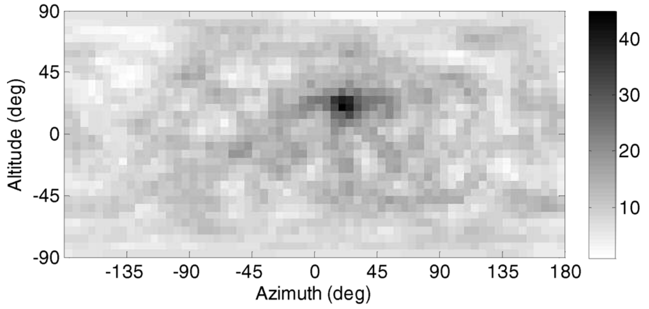
We wrote MATLAB [16] code to implement the new Compton imaging algorithms and verify their functionality. We then used GEANT4, version 4.6.1, to model the CdZnTe array and generate a data file of positions and energies. The built-in low-energy physics processes, including Doppler broadening, were used. The MATLAB scripts adjust the positions and energies to account for the finite resolution of the detector, apply the imaging algorithm, and plot the results onto a spherical surface. This last step is not necessary for the two-cone algorithm, but enables visual comparison of the results. The figures in this section represent the image as a flat surface with the azimuth as the horizontal axis and the altitude as the vertical axis. The center point of the image corresponds to a location directly in front of the operator.

Fig. 4 compares the images generated by the two algorithms for an ideal scenario. The GEANT model featured a 662-keV source directly in front of the operator, at a distance of 25 cm. The simulation captured 2000 sequences with two or three interactions. To illustrate the relative performance of the algorithms without detector-specific blurring, we temporarily disabled the adjustments for finite position and energy resolution. However, we did not exclude partial-energy sequences from the image reconstruction. The one-cone algorithm traced the Compton cones onto a 256×128 image using an angular uncertainty of $\delta = 1.4^\circ$. The two-cone algorithm plotted the intersection points onto an image of the same dimensions.

As shown, both images contain the source in the correct location. The internal structure of the spot results from the far-field approximation. Notice that the two-cone image has fewer artifacts outside the main spot, partly because the maximum amplitude is much higher. However, the one-cone algorithm requires far less time: 2.4 seconds as opposed to 76.6 seconds on a typical 3.2-GHz desktop computer. The vast difference in performance resulted from the two-cone algorithm having to iterate over many pairs of cones. In practice, the detector will not be exposed to such high count rates during typical operation.

Incorporating finite position and energy resolution into the reconstruction produces the results in Fig. 5. We rounded the x and y positions to the nearest 0.1 cm to account for the pixelated anode, and rounded the z position to the nearest 0.05 cm to account for the finite depth resolution. Similarly, we added Gaussian noise with a 6-keV FWHM to the energy values. In this case, the source is still located in the same position, but no

(a) One-cone algorithm



(b) Two-cone algorithm

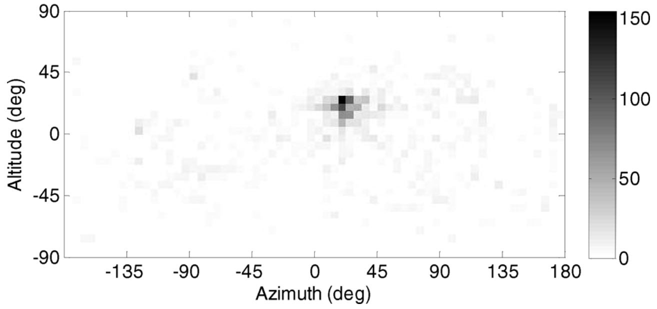
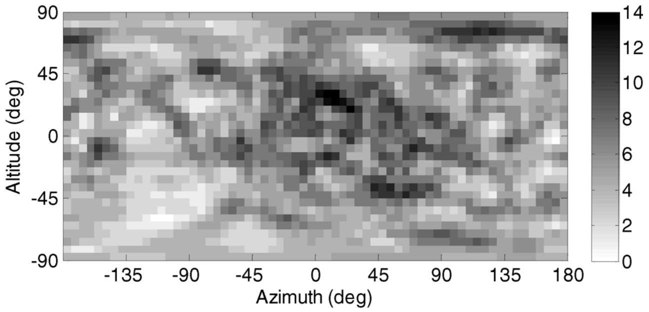


Fig. 6. Simulation of an off-center 1-μCi 662-keV point source plus a soil background for a run time of 10 s. The distance to the source is 25 cm.

(a) One-cone algorithm



(b) Two-cone algorithm

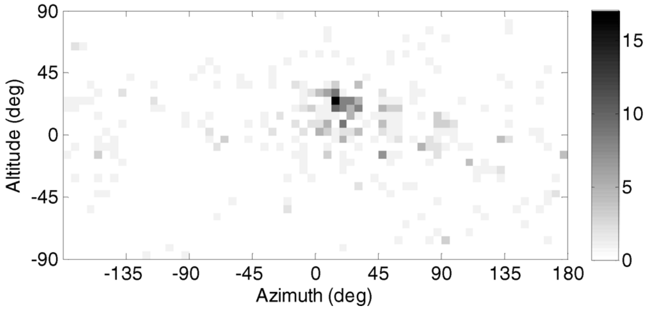


Fig. 7. Simulation of an off-center 1-μCi 2614-keV point source plus a soil background for a run time of 10 s. The distance to the source is 25 cm.

internal structure is visible. The background noise level also increases somewhat. The amplitude of the main spot decreases since approximately 19% of sequences are rejected as invalid after the quantization process.

B. Realistic Simulations

To evaluate the performance of the algorithms in a more realistic scenario, we constructed a GEANT4 model consisting of a 1-μCi 662-keV point source superimposed on a circle of soil with a 5-m radius. The point source was positioned off-axis at (21.8°, 20.2°), and 25 cm from the detector. The soil model was based on MCNP [17] calculations that assume a density of 1.82 $\mathrm{g/cm^3}$ and primary constituents of O, Si, Ca, Al, Fe, and Mg. The radioisotope constituents are as follows: 2.44 mg ^{40}K, 9.6 mg $^{232}\mathrm{Th}$, 19.4 μg ^{235}U, 2.68 mg ^{238}U, and 200 pCi ^{137}Cs (fallout) per kilogram of soil with 10^9 years of grow-in. The soil was positioned behind the source to give the worst-case situation.

GEANT4 simulations indicate that the instrument will observe 43.4 sequences per second from the source, and 8.8 sequences per second from the soil. Of the sequences from the source, 63% of the total do not deposit the full energy in the detector, an additional 15% of the total do not consist of either two or three interactions, and an additional 4% of the total are not correctly ordered using the methods in Section II.B. Some events are also lost due to dead time in the detector readout (60 μs to 1 ms per sequence), but these are negligible for such a low count rate. The remaining 18% of the sequences—around 7.9 per second—will contribute to the correct source location. Note that the same count rate would be produced by an unshielded 100-μCi source at 2.5 m.

We then evaluated the two algorithms against a dataset that contained 10 seconds of simulated events from the scenario above. The one-cone algorithm plotted the results on a 64 × 32 image with an angular uncertainty $\delta = 5.6°$. The two-cone algorithm used the same image size and only intersected cones with $\varepsilon < 0.05$. Results from this simulation appear in Fig. 6.

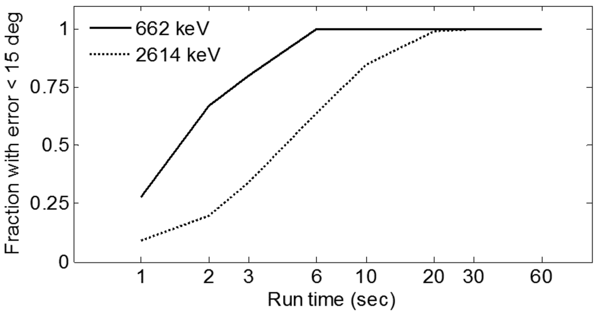
The two-cone image contains less visible clutter, as before, although both images indicate the source near the proper location. The system could easily find the maximum pixels in either image to estimate the directionality. These occur at (19.7°, 19.7°) for the one-cone image, and (19.7°, 25.3°) for the two-cone image. Note that each pixel represents about a 6° × 6° square, so the discrepancies from the actual source location are on the same order as the resolution.

Fig. 7 depicts a similar scenario, replacing the 662-keV source with a 2614-keV source of the same strength. The observed count rate from the source is now 30.7 sequences per second, with 2.4 per second contributing to the correct source location. The one-cone image has three adjacent maximum pixels around (12.2°, 28.5°). The other dark spots are artifacts from the background and sequences that did not deposit the full energy. In contrast, the two-cone has one maximum pixel at (14.1°, 25.3°). Increasing the number of source events or performing energy windowing to specifically image only the 2614 keV line would improve both images.

C. Estimated Angular Error

Rerunning the simulations in Figs. 6 and 7 with a different set of sequences would produce artifacts in different locations. As a result, the maximum pixel might be near the true source in one instance, and far away in another instance. To quantify these effects, we ran the previous simulations for run times ranging from

(a) One-cone algorithm



(b) Two-cone algorithm

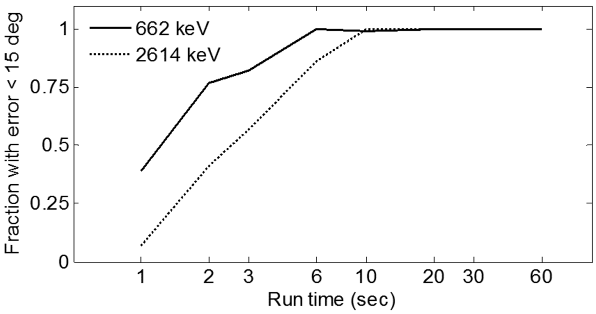


Fig. 8. Estimated angular error for off-center 662-keV and 2614-keV sources superimposed on a soil background.

1 s to 100 s. We performed 100 trials with randomly-chosen sets of sequences for each run time.

Fig. 8 plots the fraction of images whose maximum pixels were all less than 15° from the true location at (21.8°, 20.2°). Both algorithms produce better results for the 662-keV source than the 2614-keV source. However, the two-cone algorithm achieved somewhat better performance. Note that the images will still contain a significant spot near the location of the source, even if the maximum pixel is located elsewhere. More sophisticated directionality algorithms would identify the source with greater reliably.

D. Energy Efficiency

We have selected a field-programmable gate array (FPGA) to implement the imaging and directionality calculations on the GammaTracker radioisotope identifier. This integrated circuit contains a reconfigurable array of computational cells and interconnection structures. Advanced devices such as the Xilinx Virtex-4 contain a number of features to accelerate digital signal processing, including dedicated multipliers and fast adder logic. In general, FPGA platforms provide better performance and power efficiency than microprocessors for highly parallel, data intensive tasks. The imaging algorithms proposed here are ideal candidates for this approach.

To determine which algorithm would be better to implement on the FPGA, we do not compare the execution time or power consumption directly, because these metrics depend on the manner in which the algorithm is mapped onto the device. For example, the FPGA could update the image one point at a time, or update several points in parallel using multiple functional units. Rather, we estimate the energy consumption (power multiplied by time) of each algorithm. Higher energy consumption directly translates into lower battery life.

TABLE I
COMPUTATIONAL REQUIREMENTS OF IMAGING ALGORITHMS

Operation	Energy (pJ)	Operations per sequence[a]	
		One-cone	Two-cone
$+, -, <$	2.648	$2r + 24t + 12$	$(8s + 3)(m - 1) + 24t + 5$
$\times$	30.000	$3r + 42t + 12$	$27s(m - 1) + 24t + 9$
$\div$	234.081	$6t + 1$	$2s(m - 1) + 6t + 1$
$\surd$	290.433	$3t + 1$	$s(m - 1) + 3t$
memory	25.391	$2r + 5$	$(3s + 1)(m - 1) + 11$

[a] Recall that r represents the image size for the one-cone algorithm, m the number of cones for the two-cone algorithm, s the fraction of sequences with similar energies, and t the fraction of three-event sequences.

We chose the Xilinx Virtex-4 FX20 as the target platform for this analysis. We also assume that the algorithms work with 16-bit fixed-point data. This assumption does not create a significant loss of precision, since the position and energy resolution are no more than 12 bits.

To begin, we synthesized a number of mathematical operations using the Xilinx ISE 8.2 software, including the four arithmetic functions plus square root. For multiplication, the Virtex-4 contains embedded 18-bit multipliers that can also add a third term to the result. For division and square root, we synthesized a fixed-point CORDIC stage using the Xilinx CORE generator. We set the pipelining level to maximum for high performance. Next, we used the Xilinx power estimator tool [18] to determine the energy consumption of each mathematical operation. This spreadsheet allows designers to quickly estimate the power dissipation of synthesized designs, knowing the resources used. We also used the spreadsheet to find the energy consumption of memory operations. The resulting values appear in Table I.

We then counted the number of mathematical operations required for each algorithm to process a sequence of two or three interactions. For the one-cone algorithm, the system must perform the energy correction, arrange the events in sequence, calculate the parameters of the Compton cone, and evaluate the expression (15) for each point on the image. Recall that the image contains r points in all. For the two-cone algorithm, the system performs the first three steps and then calculates the intersection points with the last m cones that have similar energies. Table I lists the resulting operation counts. The variable s represents the fraction of sequences with similar energies, and the variable t represents the fraction of sequences with three interactions.

Finally, Fig. 9 illustrates the estimated energy consumption of the two algorithms as a function of number of pixels r and number of sequences m. We set s to 44% and t to 37% based on GEANT4 simulated data with a 662-keV source. As shown, the two-cone algorithm is more efficient for large m than the MATLAB script would suggest. Although division and square root require more energy than addition and multiplication, the fixed-point CORDIC implementation achieves much higher performance than floating-point calculations. The value of s will also decrease for situations with multiple sources, further improving the results. The one-cone algorithm does work well for smaller image sizes.

In summary, for the values of s and t assumed above, the two algorithms consume about the same energy per sequence for an

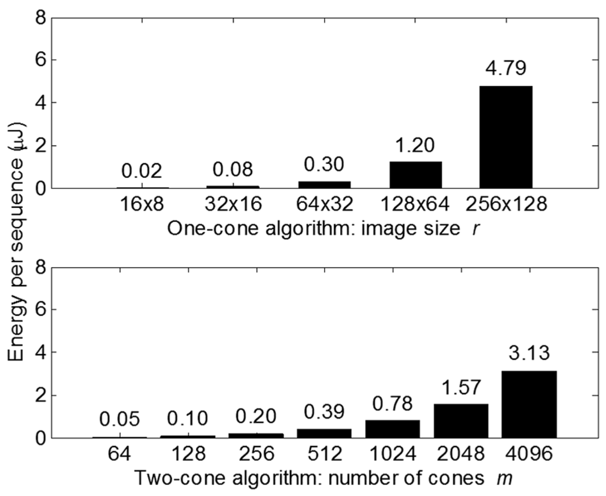


Fig. 9. Estimated energy consumption per sequence for a realistic mixture of two-event and three-event sequences.

image size of $r = 128 \times 64$ and a buffer size of $m = 1568$. For fewer events per image, the two-cone algorithm developed for the GammaTracker project will consume less energy than the one-cone algorithm, resulting in lower power consumption, faster processing speed, or both, depending on the FPGA implementation.

VI. CONCLUSION

The two Compton imaging algorithms presented in this paper are well suited for handheld radiation identifiers that have limited processing capability and battery life. The one-cone algorithm is more efficient for higher event rates, since the complexity of the two-cone algorithm grows with the number of events in the buffer. However, the two-cone algorithm is more efficient for higher-resolution images, since the complexity of the one-cone algorithm grows with the image size. The two-cone algorithm is also attractive for its inherent energy discrimination, which naturally lowers the computational overhead. This energy windowing results in fewer spurious cones when reconstructing events from high-energy sources. The two-cone algorithm also identifies the source more reliably.

We have designed the proposed imaging algorithms for efficient implementation on an FPGA. These devices offer better performance than microprocessors for signal processing applications. To reduce the energy consumption further, the algorithms rely on basic functions such as addition and multiplication as much as possible. The target system could also implement both algorithms on the FPGA and switch between them as dictated by application needs.

REFERENCES

[1] C. E. Lehner, Z. He, and F. Zhang, "4π Compton imaging using a 3-D position-sensitive CdZnTe detector via weighted list-mode maximum likelihood," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 4, pp. 1618–1624, Aug. 2004.

[2] S. Wilderman, W. L. Rogers, G. F. Knoll, and J. C. Engdahl, "Fast algorithm for list mode back-projection of Compton scatter camera data," *IEEE Trans. Nucl. Sci.*, vol. 45, no. 3, pp. 957–962, Jun. 1998.

[3] F. Zhang, Z. He, D. Xu, G. F. Knoll, D. K. Wehe, and J. E. Berry, "Improved resolution for 3-D position-sensitive CdZnTe spectrometers," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 5, pp. 2427–2431, Oct. 2004.

[4] A. W. Lackie, K. L. Matthews, II, B. M. Smith, W. Hill, W.-H. Wang, and M. L. Cherry, "A directional algorithm for an electronically-collimated gamma-ray detector," in *Proc. IEEE Nucl. Sci. Symp.Conf. Rec.*, San Diego, CA, Oct. 2006, pp. 264–269.

[5] S. E. Boggs and P. Jean, "Event reconstruction in high resolution Compton telescopes," *Astron. Astrophys. Suppl. Ser.*, vol. 145, pp. 311–231, 2000.

[6] J. van der Marel and B. Cederwall, "Backtracking as a way to reconstruct Compton scattered γ-rays," *Nucl. Instrum. Methods Phys. Res. A*, vol. A437, pp. 538–551, 1999.

[7] R. A. Kroeger, W. N. Johnson, J. D. Kurfess, B. F. Phlips, and E. A. Wulf, "Three-Compton telescope: Theory, simulations, and performance," *IEEE Trans. Nucl. Sci.*, vol. 49, no. 4, pp. 1887–1892, Aug. 2002.

[8] A. C. Zoglauer, "First Light for the Next Generation of Compton and Pair Telescope," Ph.D. dissertation, Technical Univ. Munich, Munich, Germany, 2005.

[9] U. G. Oberlack, E. Aprile, A. Curioni, V. Egorov, and K. L. Giboni, "Compton scattering sequence reconstruction algorithm for the liquid xenon gamma-ray imaging telescope (LXeGRIT)," in *Proc. SPIE*, 2000, vol. 4141, pp. 168–177.

[10] S. Agostinelli *et al.*, "G4—A simulation toolkit," *Nucl. Instrum. Methods Phys. Res. A*, vol. A506, no. 3, pp. 250–303, Jul. 2003.

[11] V. Shoenfelder, A. Hirner, and K. Schneider, "A telescope for soft gamma ray astronomy," *Nucl. Instrum. Methods*, vol. 107, pp. 385–394, 1973.

[12] B. F. Phlips, S. E. Inderhees, R. A. Kroeger, W. N. Johnson, R. L. Kinzer, J. D. Kurfess, B. L. Graham, and N. Gehrels, "Performance of a Compton telescope using position-sensitive germanium detectors," *IEEE Trans. Nucl. Sci.*, vol. 43, no. 3, pp. 1472–1475, Jun. 1996.

[13] K. Vetter, M. Burks, and L. Mihailescu, "Gamma-ray imaging with position-sensitive HPGe detectors," *Nucl. Instrum. Methods Phys. Res. A*, vol. A525, no. 1–2, pp. 322–327, Jun. 2004.

[14] C. E. Ordonez, W. Chang, and A. Bolozdynya, "Angular uncertainties due to geometry and spatial resolution in Compton cameras," *IEEE Trans. Nucl. Sci.*, vol. 46, no. 4, pp. 1142–1147, Aug. 1999.

[15] M. Woodring *et al.*, "Advanced multi-dimensional imaging of gamma-ray radiation," *Nucl. Instrum. Methods Phys. Res. A*, vol. A505, no. 1–2, pp. 415–419, Jun. 2003.

[16] [Online]. Available: http://www.mathworks.com

[17] F. B. Brown *et al.*, "MCNP version 5," *Trans. Amer. Nucl. Soc.*, vol. 87, pp. 273–276, Nov. 2002.

[18] [Online]. Available: http://www.xilinx.com/products/design_resources/power_central