Published in final edited form as:

Proc SPIE Int Soc Opt Eng. 2014 August 17; 9214: . doi:10.1117/12.2066181.

A SPECT system simulator built on the SolidWorks™ 3D-Design package

Xin Li and Lars R. Furenliid

Center for Gamma-Ray Imaging and College of Optical Sciences, University of Arizona, Tucson, AZ 85724, USA

Abstract

We have developed a GPU-accelerated SPECT system simulator that integrates into instrument-design workflow [1]. This simulator includes a gamma-ray tracing module that can rapidly propagate gamma-ray photons through arbitrary apertures modeled by SolidWorks™-created stereolithography (.STL) representations with a full complement of physics cross sections [2, 3]. This software also contains a scintillation detector simulation module that can model a scintillation detector with arbitrary scintillation crystal shape and light-sensor arrangement. The gamma-ray tracing module enables us to efficiently model aperture and detector crystals in SolidWorks™ and save them as STL file format, then load the STL-format model into this module to generate list-mode results of interacted gamma-ray photon information (interaction positions and energies) inside the detector crystals. The Monte-Carlo scintillation detector simulation module enables us to simulate how scintillation photons get reflected, refracted and absorbed inside a scintillation detector, which contributes to more accurate simulation of a SPECT system.

Keywords

SPECT; gamma-ray tracing; STL; GPU; Monte-Carlo; scintillation detector; list-mode

1. INTRODUCTION

Single photon emission computed tomography (SPECT) is a modern molecular imaging technique that is being extensively used in a variety of clinical and research applications. Many novel SPECT system designs have been proposed. In order to assist and accelerate the design process of new systems, efforts have been invested in simulation tools such as EGS4, MCNP and GATE [4–6]. With a goal of providing a simple and convenient method of simulation of novel-SPECT systems, we have developed a SPECT simulator with a GPU-aided fast ray-tracing module capable of loading arbitrary aperture STL-format models with different cross sections and tracing gamma-ray photons from the source through the aperture to interaction positions inside the detector crystal. A scintillation detector Monte-Carlo simulation module was also developed to provide accurate simulation of visible scintillation photon transport process inside a scintillation detector, producing list-mode output of the

number of photoelectrons generated in PMTs or other light sensors [7]. This can be used to provide an accurate estimation of the scintillation detector's mean detector response function (MDRF).

Combining these two modules, the complete forward process of SPECT system simulation can be performed easily and quickly. In addition to saving time and effort of the designer, this simulator allows one to accurately simulate arbitrary shapes of apertures as long as they can be modeled in SolidWorksTM, which helps in the development and assessment of aperture parameters such as keel-edges, thicknesses, choice of materials, and overall shape [8].

2. GAMMA-RAY TRACING MODULE

The ray-tracing module loads the STL file format model of an arbitrary aperture and detector crystal, then generates gamma-ray photons from a source distribution such as a digital phantom, tests whether these gamma-ray photons penetrate the aperture, are attenuated by the aperture, or generate secondary scattered or fluorescence photons. The module will also test if and where a gamma-ray photon interacts within the detector crystal. At last, it records all of the interaction positions and deposited energies. If a gamma-ray photon is scattered instead of being absorbed, its scattered photon will also be recorded, including its propagation direction. The scattered photon is treated as a new gamma-ray photon and will be traced again by the ray-tracing module. Both the photoelectric absorption and Compton scattering can trigger fluorescence photons; a fluorescence photon's random propagation direction is uniformly distributed across the whole 4π solid angle, and its energy spectrum is fixed. A fluorescence photon is also treated as a new photon and will be traced again in this module.

2.1 Generation of STL model of aperture

The STL model can be simply generated by clicking the "save as" button of SolidWorksTM, after creating the aperture model with the 3D editing tools. The STL format is a finite-element representation of the SolidWorksTM model's surfaces, which consists of a series of small triangular facets. Fig. 1 shows an example of an aperture and its corresponding STL format. Fig. 2 shows a zoom-in view of the small pinhole in the center of the aperture in fig. 1.

2.2 Ray-tracing approach

The ray-tracing approach is aimed at testing whether a gamma-ray photon can penetrate through the aperture material. The penetration probability of a gamma-ray photon can be calculated from tabulated cross sections and compared with a random number sampled from a uniform distribution of [0, 1] to tell whether a particular gamma-ray photon can penetrate or is attenuated. If it cannot penetrate the aperture (attenuation), another random number will be generated to decide whether the gamma-ray photon is absorbed or scattered. The scattered gamma-ray photon will be treated as a new source photon, and be traced again.

In order to calculate the probability of penetration, we need to know the gamma-ray photon's specifications such as current position, direction and energy (an example is shown

in table I), the type of material of the aperture, and the path length of the gamma-ray photon inside the aperture.

If the STL model of the aperture is given, the path length of the gamma-ray photon inside the aperture material can be calculated from the positions of the points where the path intersects the small triangular facets of the model. In order to know the positions of intersections, we need to determine which triangular facets lie in the photon's path. Fig. 3 shows the method of deciding whether a photon hits a small triangular facet or not.

The following three conditions ensure that a gamma-ray photon's path intersects a triangular facet, the vector $\vec{v}$ represents the direction vector of the gamma-ray photon, as shown by v_x, v_y, v_z in table I,

$$\vec{v} \cdot (\vec{OC} \times \vec{OB}) \geq 0;$$

$$\vec{v} \cdot (\vec{OA} \times \vec{OC}) \geq 0;$$

$$\vec{v} \cdot (\vec{OB} \times \vec{OA}) \geq 0.$$

Once the testing procedure is finished, we can proceed to calculate the path length a gamma-ray photon could traverse inside an aperture, as fig. 4 shows.

In fig. 4, the distance between the origin of the photon and its intersected points with the two facets OP, OP' can be calculated by,

$$\vec{OA} \cdot \hat{n} = \vec{OP} \cdot \hat{n} = OP \left(\frac{\vec{OP}}{OP} \cdot \hat{n} \right) = OP(\vec{V} \cdot \hat{n}). \quad (1)$$

$$OP = \frac{\vec{OA} \cdot \hat{n}}{\vec{V} \cdot \hat{n}}$$

$$OP' = \frac{\vec{OA}' \cdot \hat{n}'}{\vec{V} \cdot \hat{n}'}$$

Next, the path length of the gamma-ray photon between these two facets $\vec{PP'}$ can be calculated,

$$PP' = OP' - OP. \quad (2)$$

Finally the probability of penetration can be accessed. If μ_t is the total attenuation coefficient, which depends on both the gamma-ray photon energy and aperture material, then

$$P_{\text{penetration}} = \exp^{-\mu_t |\vec{P}\vec{P}'|}. \quad (3)$$

A random number is generated uniformly from [0, 1] to be compared with $P_{\text{penetration}}$. If it is less or equal than $P_{\text{penetration}}$, the gamma-ray photon will penetrate through the aperture, otherwise it will be attenuated.

If a gamma-ray photon penetrates the aperture and interacts within the detector crystal, then its interaction position and energy will be recorded, and saved in list mode. If the gamma-ray photon is attenuated by the aperture, it is possible that a secondary photon will be emitted (scattered photon or fluorescence photon). The probability of a scattered photon can be found by comparing the photoelectric cross section with the Compton-scatter cross section of the photon at its current energy; the scattering angle can be sampled according to the Klein-Nishina formula [9]. The deposited energy can be decided by the Compton-scattering formula [10].

The process of fluorescence is more complicated, since both the photoelectric effect and the Compton-scattering effect can yield fluorescence photons. Our simulator is capable of simulating both the Compton scattering and fluorescence in the aperture. However, for simplicity, the rest of this paper will not consider these relatively rare effects.

Programming with CUDA (GPGPU) greatly accelerates the testing process, in that batches of gamma-ray photons can be tested simultaneously. In this code, a batch typically contains 2048 gamma-ray photons. The speed is increased up to 20 times compared to a CPU approach.

2.3 Examples of some results

As examples, a point source and a digital brain gray-matter phantom were used as sources of gamma-ray photons, and their shadow images on a perfect detector are shown in Figs. 6, 8 and 9. The energy of simulated gamma-ray photon was 140 keV, and the apertures made of tungsten of 3 mm thickness were placed 200 mm from the point source or the center of the FOV (field of view) of the digital brain gray-matter phantom. The detector plane is placed 200 mm behind the aperture.

2.3.1 Point source—Fig. 5 shows apertures with openings shaped like the letters CGRI and outline of Mickey MouseTM. Fig. 6 shows their corresponding shadow images, produced by a point source. Fig. 7 shows cross-section images of pinhole apertures with different types of pinhole chamfers (keel-edge, knife-edge and circular chamfering). Fig. 8 shows the shadow images of the pinholes in Fig. 7, illustrating the different amounts of edge leakage.

2.3.2 Phantom—Fig. 9 shows a projection of a digital brain gray-matter phantom through a keel-edge pinhole aperture of 3 mm thickness, 2 mm pinhole diameter, 30° keel angle and

0.5 mm keel thickness. The aperture was placed 200 mm from the center of the digital brain gray-matter phantom, and the detector plane was 200 mm behind the aperture, such that the magnifying ratio is approximately 1:1.

3. SCINTILLATION-DETECTOR MONTE-CARLO SIMULATION MODULE

The major purpose of developing this scintillation-detector Monte-Carlo simulation module is to provide accurate estimation of the $MDRF(s|\vec{r})$ of the scintillation detector to produce realistic list-mode data that incorporates detector response. In order to compute the mean counts on light sensors as a function of gamma-ray interaction locations, 100000 scintillation photons were generated at each node of a grid inside the scintillation crystal, with a spacing of 0.5 mm in each dimension. The propagation of all photons are tracked to get the mean counts on all light sensors for each node in the grid.

3.1 Structure of a sample scintillation detector

The structure of a sample scintillation detector is shown in Fig. 10.

This simulated scintillation detector included a hexagonal layer of NaI(Tl) crystal (scintillation detector crystal) of 12 mm thickness with a refractive index of 1.85, and a hexagonal light guide of 12 mm thickness with a refractive index of 1.406, 7 PMTs with glass window's thickness 1 mm and refractive index 1.5. The side length of the hexagon was 178.4 mm, the radius of each end-on PMT was 56 mm.

The front surface of the scintillation crystal is treated with a Lambertian reflection layer, so photons colliding on this surface will be reflected according to the Lambertian law.

At the back surface of the light guide and where the PMTs are not attached (beyond the PMT circular area), a specular reflection condition is applied, any visible photon colliding on this surface will be reflected according to Snell's law.

The side surfaces of the scintillation crystal and the light guide were modeled as black. Any photon colliding on them was considered absorbed.

3.2 Process of photon propagation

Once a gamma-ray photon interacts inside the scintillation crystal, low-energy scintillation photons will be released. The number of released photons depends on the gamma-ray photon energy and the type of the scintillation crystal. For NaI(Tl) the light yield was taken as 38000 photons/Mev[11]. For a 140 keV gamma-ray photon, 5320 visible photons on average were generated and propagated. The number of emitted scintillation photons was assumed to be Poisson-distributed.

3.2.1 Reflection—If a visible photon arrives at a surface boundary, it will be either reflected, refracted, or absorbed. There are two kinds of reflections modeled in the simulator – Lambertian reflection and specular reflection.

For Lambertian reflection, the probability of reflected direction follows Lambertian cosine law.

For specular reflection, the reflection will be perfect-mirror reflection. The reflections at the interfaces between the scintillation crystal and the light guide, and the interface between the light guide and the surface where PMTs are attached on are all modeled as specular reflection with transmission and reflection coefficient calculated (as shown below). And total internal reflection (TIR) between two media is also accounted for.

3.2.2 Refraction—If the refractive index between two boundaries are not the same, there is possibility of reflection or refraction. According to the Fresnel equations, the probabilities of refraction for s and p type polarized photons are [12],

$$P_s = 1 - \left| \frac{n_1 \cos \theta_i - n_2 \sqrt{1 - \left(\frac{n_1}{n_2} \sin \theta_i\right)^2}}{n_1 \cos \theta_i + n_2 \sqrt{1 - \left(\frac{n_1}{n_2} \sin \theta_i\right)^2}} \right|^2 \quad (4)$$

$$P_p = 1 - \left| \frac{n_2 \cos \theta_i - n_1 \sqrt{1 - \left(\frac{n_1}{n_2} \sin \theta_i\right)^2}}{n_2 \cos \theta_i + n_1 \sqrt{1 - \left(\frac{n_1}{n_2} \sin \theta_i\right)^2}} \right|^2$$

P_s is the refraction probability for s-polarized visible photons, P_p is the refraction probability for p-polarized visible photons. Since the polarization states of scintillation photons are random, the averaged refraction probability is used instead,

$$P_{refraction} = \frac{P_s + P_p}{2} \quad (5)$$

A uniformly distributed random number at [0, 1] is generated to determine whether the scintillation photon is refracted or not. If the random number is smaller or equal to $P_{refraction}$, then the visible photon will be refracted (Snell's law is applied), otherwise the visible photon is reflected (specular reflection is applied)

3.2.3 Absorption—The destiny of a scintillation photon is to be absorbed, by either the black boundary or by the PMTs' photocathode. A photon might bounce multiple times between the interfaces before being absorbed. If a scintillation photon arrives at the black boundary, it will be absorbed, and the next scintillation photon will be traced. If a scintillation photon arrives at the PMT's photocathode, a uniformly distributed random number at [0, 1] is generated to be compared with the photocathode's quantum efficiency (QE), if the random number is smaller or equal to the QE of the photocathode, the visible photon will be absorbed by the photocathode and produce a photoelectron, thus incrementing the count of primary photoelectrons at this PMT by one. If the random number is larger than the QE of the photocathode, the scintillation photon will be absorbed and produce nothing, and the next scintillation photon will be traced.

3.3 Simulation result of mean detector response function (MDRF)

100 000 scintillation photons were launched at each grid node of 0.5 mm spacing throughout a 2D area $672 \times 768 \text{ mm}^2$, seven 2D MDRF surfaces were formed based on the percentage of scintillation photons that lead to photoelectrons in each of the seven PMTs in fig. 10. For simplicity, in this paper, only the MDRF on the middle plane of the scintillation crystal was modeled (6 mm depth inside the crystal). The images are smoothed by a $3.5 \times 3.5 \text{ mm}^2$ moving average filter. Since there are many branches in the flow of the program, CPU-cluster parallel computing is more suitable to this simulation module than CUDA.

4. MAXIMUM-LIKELIHOOD ESTIMATION WITH GPU ACCELERATED CONTRACTING-GRID SEARCH ALGORITHM

4.1 Maximum-likelihood estimation in 2D plane

The mathematical interpretation of this algorithm is,

$$(\hat{x}, \hat{y}) = \underset{(x,y)}{\operatorname{argmax}} P(\vec{s} | x, y) = \underset{(x,y)}{\operatorname{argmax}} \prod_{i=1}^n P_i(s_i | x, y) = \underset{(x,y)}{\operatorname{argmax}} \sum_{i=1}^n \ln(P_i(s_i | x, y)). \quad (6)$$

Where n is the number of PMTs in the scintillation detector, in our simulation, $n=7$. $P_i(s_i | x, y)$ is the probability of the event at position (x, y) given the i th PMT photoelectron count s_i . Assuming s_i follows Poisson distribution, probability of observing s_i at location (x, y) is given by,

$$P_i(s_i | x, y) = \frac{\exp(-\bar{s}(x, y)) \bar{s}^{s_i}}{s_i!}; \quad (7)$$

$$\ln(P_i(s_i | x, y)) = (-\bar{s}(x, y)) + s_i \ln(\bar{s}(x, y)) - \ln(s_i!).$$

Substituting (7) into (6),

$$\begin{aligned} (\hat{x}, \hat{y}) &= \underset{(x,y)}{\operatorname{argmax}} \sum_{i=1}^n \ln(P_i(s_i | x, y)) \\ &= \underset{(x,y)}{\operatorname{argmax}} \sum_{i=1}^n [(-\bar{s}(x, y)) + s_i \ln(\bar{s}(x, y)) - \ln(s_i!)] \quad (8) \\ &= \underset{(x,y)}{\operatorname{argmax}} \sum_{i=1}^n [(-\bar{s}(x, y)) + s_i \ln(\bar{s}(x, y))]. \end{aligned}$$

The last step of (8) is possible due to the fact that observed data s_i does not change with (x, y) .

An exhaust search for implementing the argmax is exceptionally computationally consuming when the detector pixel or voxel number is large. The contracting-grid search

algorithm [13] has been introduced to estimate event positions in an efficient manner. This algorithm's main idea is illustrated in fig. 12.

The grid is continuously scaled downward until the grid spacing reaches a predefined step size. Then the process is stopped, and the winner node in the last grid search is chosen as the estimated position for this event.

4.2 GPU accelerated contracting-grid search algorithm

To speed up the contracting-grid search, a GPU is used for calculation of the likelihood and finding the maximum-likelihood node. This code can process 6.2×10^5 events per second on a personal desktop with an Intel i7 4770 CPU and a Geforce GTX 760 GPU. Fig. 13 shows a flow chart of this code.

In fig. 13, there are three major modules: the trial-position-generation module, the likelihood-calculation module, and the likelihood-comparison module.

4.2.1 Trial-position-generation module—The trial position generation module is in charge of generating the positions of the nodes in the grid (such as the nodes in fig. 12) centered around the maximum-likelihood node's position from the previous iteration. Events are processed in batches, in our simulator, one batch contains 2048 events, so in the trial position module, 2048 threads are launched to assign the 16-node positions to each of the events. The results are recorded in a matrix (circled by the dotted line in fig. 13) whose row index represents each event, and whose column index represents each node position. Each element (yellow block) contains three parameters: (x, y) and $\ln(P(s|\vec{x}, y))$ (the latter will be calculated in the likelihood calculation module below).

4.2.2 Likelihood-calculation module—The likelihood-calculation module is responsible of calculating $\ln(P(s|\vec{x}, y))$ in each yellow element in fig. 13. In total, 2048×16 threads shall be launched, and each thread is responsible of calculating $\ln(P(s|\vec{x}, y))$ for each yellow element. Since the MDRF will be referenced to retrieve s at each grid node (x, y) , and interpolation of the MDRF will be applied for most of the cases, the texture memory of the GPU is a good choice for efficient calculation.

4.2.3 Likelihood-comparison module—This module's role is to find the maximum-likelihood node among all trial nodes for each event for each iteration. Binary decimation is a good choice. For example, if an event has 16 trial nodes in one iteration, we can compare the $\ln(P(s|\vec{x}, y))$ of node 1 with that of node 8 to find a winner in this duel, compare 2 with 9, 3 with 10, etc. After the first round of elimination, there are only 8 nodes left to be compared, and the same procedure can be repeated to get the top 4 nodes, and then 2 nodes, and at last the maximum-likelihood node (the champion) is determined. This process can be realized easily with CUDA.

4.3 Results of Maximum-Likelihood estimation in the scintillation detector

In order to simulate the process of the estimation of event positions, we start with list-mode data. From the gamma-ray tracing module, the exact event interaction 3D position are known in the scintillation detector crystal. Sample PMT counts can be drawn from the 3D

MDRF under the assumption of Poisson statistics. However in this paper, we only consider the 2D case: all the events are mapped onto the middle plane of the detector crystal. We then refer to the MDRF to look up the sensitivities of all PMTs at the 2D position of the event. For each of the PMTs, this probability is multiplied with the mean scintillation-photon number generated by the incident gamma-ray photon to arrive at mean-count number of that PMT. The final event data (s shown above) are achieved by Poisson sampling with each PMT's mean count.

4.3.1 Example of PSF—A PSF example at center of FOV is shown in fig. 14. The pinhole chosen was the first type in fig. 7, The aperture is 3 mm thick and located 200 mm from the center of FOV. The detector plane is located 200 mm from the aperture.

4.3.2 Example of digital brain phantom—A digital brain gray-matter phantom's exact projection (perfect detector) image and estimated projection image are compared in fig. 15. For simplicity, only the central slice was simulated. The aperture and detector configuration are the same as that of fig. 14.

5. CONCLUSION

In this paper, a SPECT system simulator that works within the system design workflow is presented. It is capable of loading the STL format model of any instrument-design shape from SolidWorksTM, which will help save time and effort in SPECT system modeling and optimization. A detector simulator module allows list-mode light sensor count output to be generated for arbitrary scintillation detector geometries.

This simulator is capable of simulating the entire process from gamma-ray photon emission of source to light sensor photoelectron count output. Also, with the help of GPUs, this simulator is fast, making it a convenient tool for SPECT system validation on realistic clinical tasks.

Acknowledgment

The authors would like to thank Cecile Chaix for her suggestion to ray-trace with SolidWorksTM models and Luca Caucci for advice on GPU programming. This work was partially supported by NIH/NIBIB grants P41-EB002035 The Center for Gamma-Ray Imaging and 5R01EB000803.

REFERENCES

1. Nvidia CUDA. Compute unified device architecture programming guide. 2007
2. SolidWorksTM is a product of Dassault Systèmes SolidWorks Corp, a subsidiary of Dassault Systèmes. S. A. Vélizy, France:
3. StereoLithography Interface Specification, 3D Systems, Inc. 1989 Oct.
4. Bielajew, A.; Hirayama, H.; Nelson, WR.; Rogers, DWO. History, Overview and Recent Improvement of EGS4. United States: Department of Energy. Office of Science; 2004.
5. Du Y, Frey EC, Wang WT, Tocharoenchai C, Baird WH, Tsui BMW. Combination of MCNP and SimSET for Monte Carlo simulation of SPECT with medium-and high-energy photons. Nuclear Science, IEEE Trans. Nucl. Sci. 2002; 49(3):668–674.
6. Jan S, Santin G, Strul D, Staelens Steven, Assie K, Autret D, Avner S, et al. GATE: a simulation toolkit for PET and SPECT. Phys. Med. Biol. 2004; 49(19):4543–4561. [PubMed: 15552416]

7. Peterson, Todd E.; Furenliid, Lars R. SPECT detectors: the Anger Camera and beyond. *Phys. Med. Biol.* 2011; 56(17):R145–R182. [PubMed: 21828904]
8. Beekman, Freek; van der Have, Frans. The pinhole: gateway to ultra-high-resolution three-dimensional radionuclide imaging. *Eur. J. Nucl. Med. Mol. Imag.* 2007; 34(2):151–161.
9. Klein, O.; Nishina, Y. On the Scattering of Radiation by Free Electrons According to Dirac's New Relativistic Quantum Dynamics. In: Ekspong, Gsta, editor. *The Oskar Klein Memorial Lectures, Vol. 2: Lectures by Hans A. Bethe and Alan H. Guth with Translated Reprints by Oskar Klein.* Singapore: World Scientific Publishing Co. Pte. Ltd.; 1994. p. 113-139.
10. Compton, Arthur H. A Quantum Theory of the Scattering of X-Rays by Light Elements. *Phys. Rev.* 1923; 21(5):483–502.
11. Knoll, Glenn F. *Radiation Detection and Measurement* (3rd, edition). 1999:231.
12. Griffiths DJ. Pearson Education, Dorling Kindersley. *Introduction to Electrodynamics* (3rd, edition). 2007:386–392.
13. Hesterman, Jacob Y.; Caucci, Luca; Kupinski, Matthew A.; Barrett, Harrison H.; Furenliid, Lars R. Maximum-Likelihood Estimation With a Contracting-Grid Search Algorithm. *IEEE Trans. Nucl. Sci.* 2010; 57(3):1077–1084. [PubMed: 20824155]

Figure 1.
An aperture modeled in SolidWorksTM (left) and its corresponding stereolithography (.STL) format representation (right).

Figure 2.
The zoom-in view of the pinhole centered at the aperture in fig. 1 and its corresponding stereolithography (.STL) format representation.

Figure 3.
Geometry of deciding whether a gamma-ray photon intersects a triangular facet.

Figure 4. Illustration of how to calculate the gamma-ray photon's path length between 2 intersected facets. $\hat{n}$ and $\hat{n}'$ are the normal vectors of the left and right facets; P and P' are the intersected points of the left and right facets respectively. $\vec{v}$ is the direction vector of the photon.

Figure 5.
Images of apertures with openings of letters "CGRI" and a Mickey Mouse.

Figure 6. Shadow projections on a perfect-detector plane (detector crystal middle plane, which is 6 mm deep), produced by a point source along with the apertures in fig. 5.

Figure 7. 3 mm thick aperture with a pinhole of diameter 1 mm, keel angle 30° with keel-edge, knife-edge and circular-chamfering from left to right, respectively.

Figure 8. Projections onto a perfect detector plane of keel-edge, knife-edge and circular-chamfering from left to right respectively.

Figure 9.
Shadow projection of a digital brain phantom on the detector plane.

Figure 10. Structure of a hexagonal scintillation detector. Typically, it includes the scintillation crystal, light guide, and PMT tubes with glass window.

Figure 11.
MDRF surfaces for the seven PMTs in fig. 10, blue hexagons represent the boundaries.

Figure 12.

A 4×4 grid (red) is shown in the left graph, these 16 grid nodes are the original trial positions. After finding the node whose likelihood value is the largest, another smaller scaled grid (blue) is formed and centered at this node, as shown in the right graph. This process repeats until a desired level of precision is achieved in the estimate.

Figure 13.

GPU based contracting-grid search algorithm architecture flow chart.

Figure 14.

Comparison of PSFs at the center of FOV. Left graph shows an ideal detector's PSF, right graph shows the ML-estimated PSF. Simulated scintillation detector presents realistic spatial resolution.

Figure 15.
Comparison between ideal projection (left) and estimated projection (right) of a slice of digital brain grey-matter phantom.

Table 1

An example of photon specifications. p_x , p_y , p_z represent the photon's position in mm; v_x , v_y , v_z represent the photon's direction in terms of unit vectors in the x, y and z direction; and photon energy is in keV.

p_x	p_y	p_z	v_x	v_y	v_z	energy
100.1	57.2	-30.0	0.6313	0.3185	0.7071	140