



SAINT VINCENT COLLEGE

DEPARTMENT OF  
PHYSICS

JOHN PIERRE MENEGHINI

# COMPUTATIONAL MODELING OF X-RAY ATTENUATION AND SIMULATION OF MEDICAL IMAGING

BACHELOR OF SCIENCE

Saint Vincent College

*Draft: April 18, 2024*

# COMPUTATIONAL MODELING OF X-RAY ATTENUATION AND SIMULATION OF MEDICAL IMAGING

JOHN PIERRE MENEHINI

Thesis submitted in partial fulfillment of the requirements  
for the degree of *Bachelor of Science*

**Adviser:** Fr. Michael Antonacci O.S.B  
*Associate Professor*

BACHELOR OF SCIENCE

Saint Vincent College

*Draft: April 18, 2024*

# ABSTRACT

Regardless of the language in which the dissertation is written, usually there are at least two abstracts: one abstract in the same language as the main text, and another abstract in some other language.

The abstracts' order varies with the school. If your school has specific regulations concerning the abstracts' order, the NOVAthesis L<sup>A</sup>T<sub>E</sub>X (`novathesis`) (L<sup>A</sup>T<sub>E</sub>X) template will respect them. Otherwise, the default rule in the `novathesis` template is to have in first place the abstract in *the same language as main text*, and then the abstract in *the other language*. For example, if the dissertation is written in Portuguese, the abstracts' order will be first Portuguese and then English, followed by the main text in Portuguese. If the dissertation is written in English, the abstracts' order will be first English and then Portuguese, followed by the main text in English. However, this order can be customized by adding one of the following to the file `5_packages.tex`.

```
\ntsetup{abstractorder={<LANG_1>,...,<LANG_N>}}  
\ntsetup{abstractorder={<MAIN_LANG>={<LANG_1>,...,<LANG_N>}}}
```

For example, for a main document written in German with abstracts written in German, English and Italian (by this order) use:

```
\ntsetup{abstractorder={de={de,en,it}}}
```

Concerning its contents, the abstracts should not exceed one page and may answer the following questions (it is essential to adapt to the usual practices of your scientific area):

1. What is the problem?
2. Why is this problem interesting/challenging?
3. What is the proposed approach/solution/contribution?
4. What results (implications/consequences) from the solution?

**Keywords:** One keyword, Another keyword, Yet another keyword, One keyword more, The last keyword

# CONTENTS

<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>v</b>
<b>Acronyms</b>	<b>vii</b>
<b>1 X-ray Tracing</b>	<b>1</b>
1.0.1 The Brute Force Algorithm . . . . .	1
1.0.2 Ray-Triangle Intersections . . . . .	4
<b>2 Monte Carlo Approach</b>	<b>8</b>
2.1 Transport Theory . . . . .	8
2.1.1 Surface & $\delta$ -Tracking . . . . .	9
2.1.2 Photon-Matter Interactions . . . . .	9
2.2 Methods . . . . .	12
2.2.1 Geometry . . . . .	12
2.2.2 Materials & Data . . . . .	13
2.2.3 Source . . . . .	14
2.2.4 Tallies . . . . .	14
2.3 Validation . . . . .	15

## LIST OF FIGURES

1.1	A diagram of a parametric ray. . . . .	1
1.2	A basic ray-tracing setup with a camera, ray, and viewport. . . . .	2
1.3	A diagram showing the geometric representations of $\overrightarrow{HO}$ , $\overrightarrow{VE}$ , $\overrightarrow{FL}$ , and $\overrightarrow{LHC}$ with respect to the viewport. . . . .	3
1.4	A diagram showing the geometric representation of Equation 1.5. . . . .	4
2.1	Photon transport process in MIDSX. . . . .	15
2.2	Results for the (a) HVL and (b) QVL simulations as described by Case 1. The ratios of the primary HVL and QVL air kermas to the primary background air kermas is represented by $R_1$ and $R_2$ , respectively. The simulation was performed for the monoenergetic energies 30 keV and 100 keV, along with the polyenergetic spectrums of 30 kVp and 100 kVp provided by TG-195. A dashed-line is placed at $R_1 = 0.5$ and $R_2 = 0.25$ for comparison. . . . .	16
2.3	The energy deposited per initial photon (p.i.p.) (eV/photon) in the simulated tissue for the full-field simulation as described by Case 2. The simulation was performed at 56.4 keV and 120 kVp at both (a) $0^\circ$ and (b) $15^\circ$ , with the 120 kVp spectrum provided by TG-195. . . . .	17
2.4	The energy per initial photon (eV/photon) of photons incident upon each region of interest (ROI) for the $0^\circ$ , full-field, 56.4 keV simulation as described by Case 2. The incident energy was determined separately for photons that underwent (a) no real interactions, (b) a single incoherent scatter, (c) a single coherent scatter, (d) and multiple scatters. . . . .	17
2.5	The energy per initial photon (p.i.p.) (eV/photon) of photons incident upon each region of interest (ROI) for the $0^\circ$ , pencil beam, 56.4 keV simulation as described by Case 2. The incident energy was determined separately for photons that underwent (a) a single incoherent scatter and (b) a single coherent scatter. . . . .	18

2.6	The energy deposited per initial photon (eV/photon) in the volumes of interests (VOIs) for the full field simulation as described by Case 2. The simulation was performed at (a) $0^\circ$ /56.4 keV, (b) $0^\circ$ /120 kVp, (c) $15^\circ$ /56.4 keV, and (d) $15^\circ$ /120 kVp. . . . .	18
2.7	The energy deposited per initial photon (eV/photon) in the material IDs for the $180^\circ$ , 56.4 keV simulation as described by Case 5. . . . .	19
2.8	The energy deposited per initial photon (p.i.p.) (eV/photon) in the material IDs/organs composing a voxelized human phantom for the $0^\circ$ , 120 kVp simulation as described by Case 5. . . . .	19

## LIST OF TABLES

## LIST OF LISTINGS

1	C++ implementation of the Möller-Trumbore ray-triangle intersection algorithm. . . . .	7
---	--	---



# ACRONYMS

`novathesis` NOVAthesis L<sup>A</sup>T<sub>E</sub>X (*p. i*)

## X-RAY TRACING

Ray tracing is a computer graphics technique that involves tracing the path of rays as they pass through a virtual scene. Due to its ability to create highly-realistic images, it has been used extensively in animations, video games, scientific computing, and by designers who need a physically accurate design of a product. While not the only rendering technique or the fastest in computer graphics, its ability to consistently produce physically realistic renders has made it an indispensable tool in many industries. By accurately simulating the behavior of light, ray tracing can capture intricate details of how light interacts with objects, resulting in realistic shadows, reflections, refractions, and global illumination effects. This level of visual fidelity allows for the creation of visually stunning and immersive experiences that were previously challenging to achieve. As hardware capabilities continue to advance and real-time ray tracing becomes more accessible, its applications are expanding, and its impact on various fields is growing, pushing the boundaries of what is possible in computer graphics and visual simulation [Peddie].

### 1.0.1 The Brute Force Algorithm

The rays in a scene are described mathematically as the parametric representation of a line. In three dimensions, the point at the end of the line  $\vec{r}$  at parameter  $t$  is given by the following equation:

$$\vec{r}(t) = \vec{r}_0 + \hat{d}t, \quad (1.1)$$

where  $\vec{r}_0$  points from the origin to the start of the line and  $\hat{d}$  is a unit vector parallel to the direction of the line. The described ray is shown in Figure 1.1.

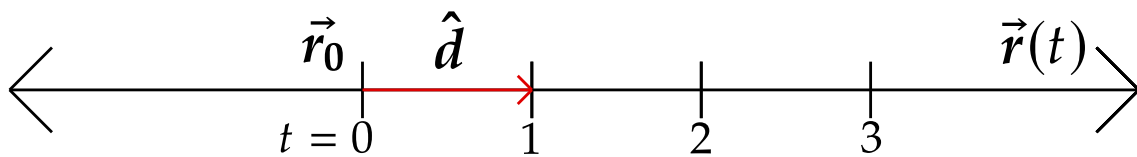


Figure 1.1: A diagram of a parametric ray.

Unlike in real life where light rays end at a camera, we instead trace light rays from the camera to locations in the scene. While one could trace light rays from a light source (forward ray tracing), many of these rays would end up missing the camera entirely, not affecting the image. More efficiently, the light rays are instead traced from the camera to the light source (backward ray tracing), drastically cutting back on the computational load while producing an identical image for most practical cases [Peddie]. So, in the case of backward ray tracing, the method used in this research, rays are emitted from the camera; therefore,  $\vec{r}_0$  represents the location of the camera in the scene.

To define the area of the scene visible from the point of view of the camera, one can create a rectangular surface in which all rays pass through, called a viewport. Increasing the size of the viewport increases the amount of the scene that is visible in the final image, and vice versa. This viewport is broken up into a grid, the resolution of which determines the resolution of the final image. Rays are then sent through each point in this grid and out into the scene.

So far, we have a camera, light rays, and a way to define what parts of the scene are visible; all three of which are visualized in Figure 1.2. It is important to note that, by convention, the z-axis in the scene points opposite the viewport.

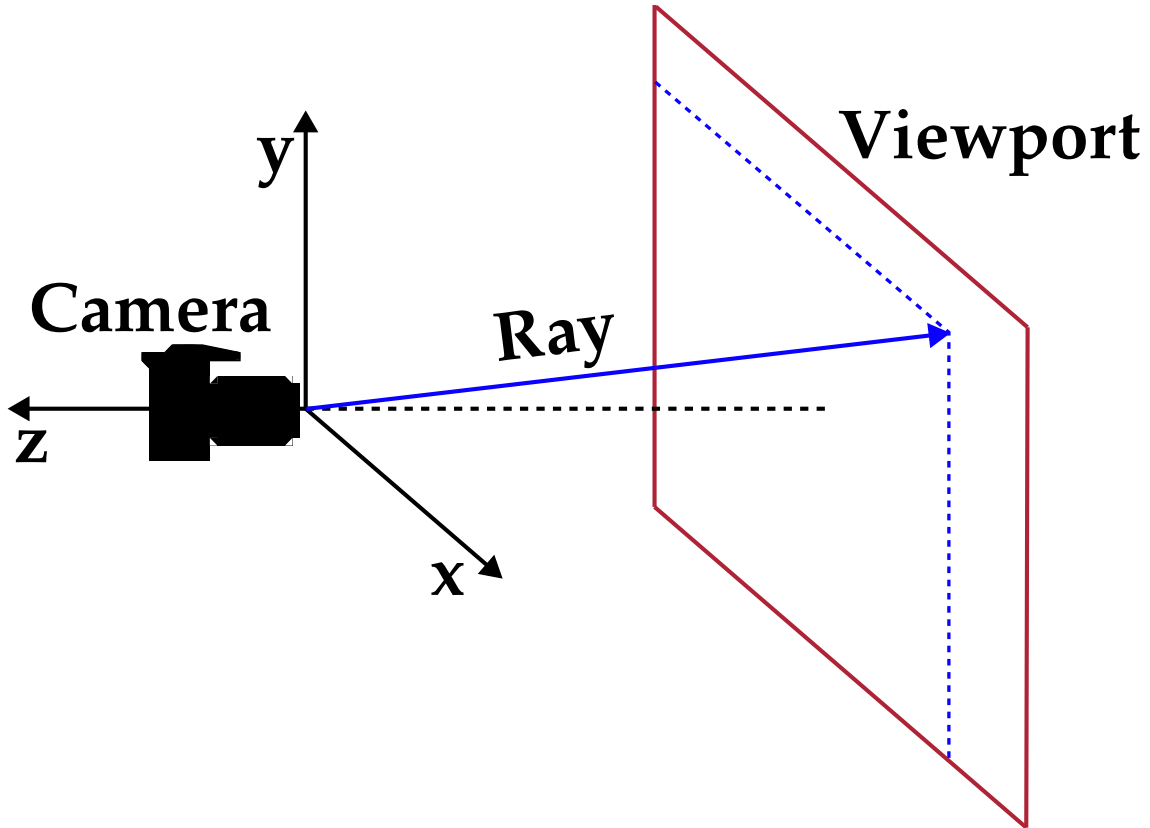


Figure 1.2: A basic ray-tracing setup with a camera, ray, and viewport.

To set the location of the viewport in the scene, we must first define three new vectors:

$\overrightarrow{HO}$ ,  $\overrightarrow{VE}$ , and  $\overrightarrow{FL}$ , which are vectors pointing from the left most to right most point ( $-x$  to  $x$  direction), the bottom most to top most point ( $-y$  to  $y$  direction), and from the center of the viewport to  $\vec{r}_0$ , respectively. A vector pointing from  $\vec{r}_0$ , to the bottom-left corner  $\overrightarrow{LHC}$  can be calculated using the following equation:

$$\overrightarrow{LHC} = \vec{r}_0 - \frac{1}{2}\overrightarrow{HO} - \frac{1}{2}\overrightarrow{VE} - \overrightarrow{FL}, \quad (1.2)$$

which was obtained geometrically from Figure 1.3 [Shirley].

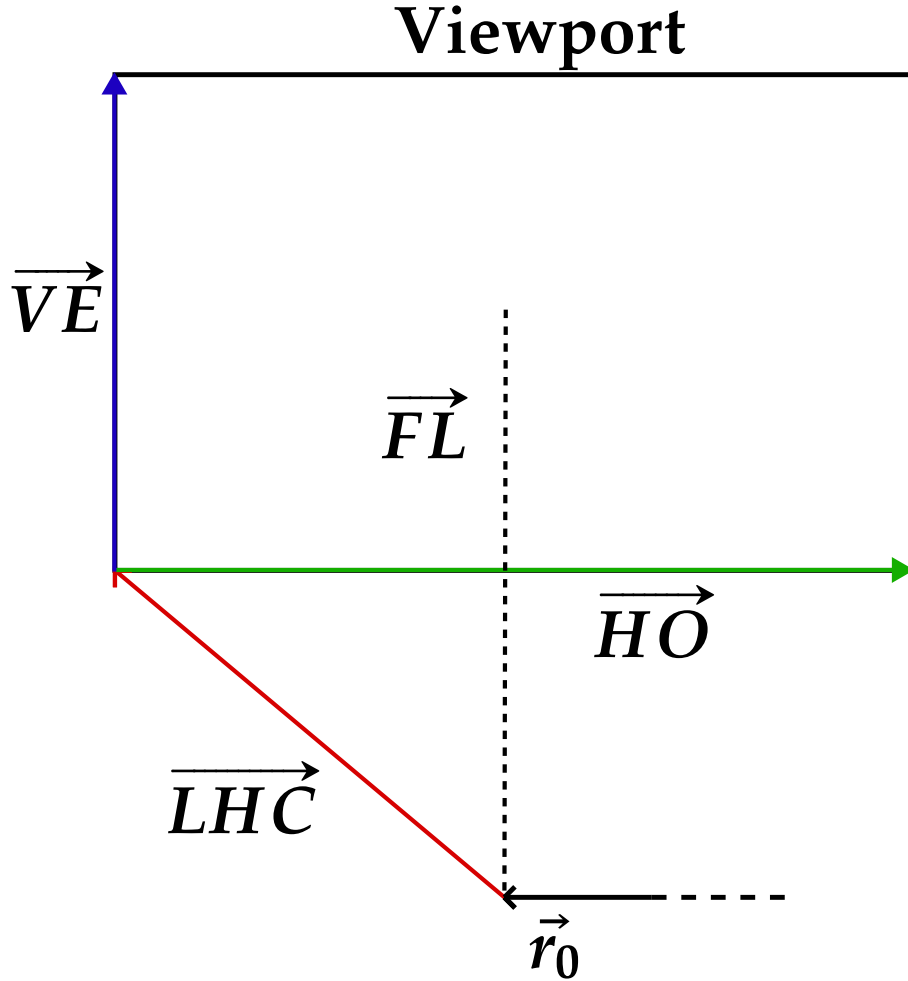


Figure 1.3: A diagram showing the geometric representations of  $\overrightarrow{HO}$ ,  $\overrightarrow{VE}$ ,  $\overrightarrow{FL}$ , and  $\overrightarrow{LHC}$  with respect to the viewport.

With  $\overrightarrow{LHC}$  pointing to the left-hand-corner of the viewport in the scene,  $\overrightarrow{HO}$  and  $\overrightarrow{VE}$  can be scaled to define a ray pointing from the camera to any point on the viewport. In particular, the viewport's horizontal and vertical coordinates,  $u_i$  and  $v_j$ , are defined as such,

$$u_i = i/(\text{image width} - 1); \quad (1.3)$$

$$v_j = j/(\text{image height} - 1), \quad (1.4)$$

where the image width and height are the resolution of the rendered image,  $i$  is an integer ranging from  $[0, \text{image width} - 1]$ , and  $j$  is an integer ranging from  $[0, \text{image height} - 1]$ . Note that both  $u_i$  and  $v_j$  range from  $[0, 1]$ .

Therefore, using these newly defined viewport coordinates, a ray taking the form of Equation 1.1, pointing from the camera to the pixel coordinates  $(i, j)$  is given by the following equation:

$$\vec{r}_{i,j}(t) = \vec{r}_0 + t \left( \overrightarrow{LHC} + u_i \overrightarrow{HO} + v_j \overrightarrow{VE} \right), \quad (1.5)$$

which is represented geometrically in Figure 1.4 [Shirley].

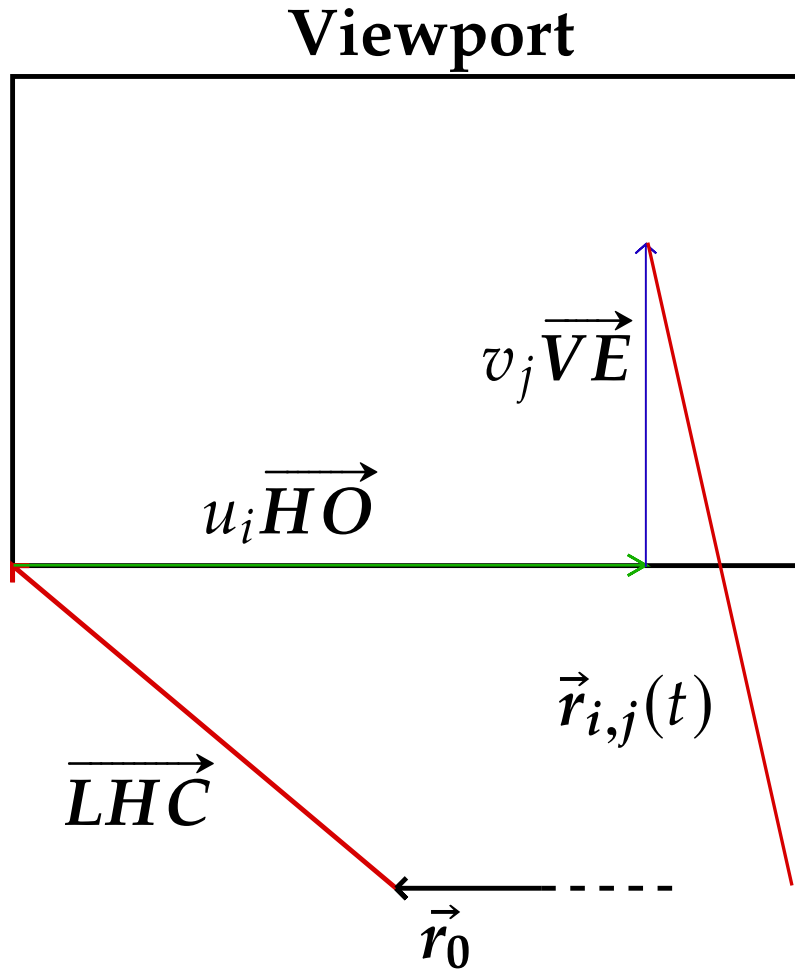


Figure 1.4: A diagram showing the geometric representation of Equation 1.5.

## 1.0.2 Ray-Triangle Intersections

With the end goal to be able ray-trace x-rays through an arbitrary user-defined mesh, we first need to choose which shape to compose said mesh with. Due to its efficient ray-intersection algorithm and its ease of parallelization on Graphics Processing Units (GPUs), these constituents, referred to as primitives, are typically chosen to be triangles. This following section will develop the ray-triangle intersection algorithm used by the

x-ray tracer. In particular, the algorithm used was invented by Tomas Möller and Ben Trumbor, and is subsequently referred to as the Möller-Trumbore ray-triangle intersection algorithm [moller2005fast].

### Barycentric Coordinates

To motivate the unique coordinate system used by the algorithm, let us first review the concept of center of mass for 3 massive point particles. If 3 objects each have their own location in space  $\vec{r}_i$ , each with mass  $m_i$ , the center of mass  $\vec{R}$  is given by:

$$\vec{R} = \frac{m_1\vec{r}_1 + m_2\vec{r}_2 + m_3\vec{r}_3}{m_1 + m_2 + m_3}. \quad (1.6)$$

Notice a few properties of  $\vec{R}$  that are intuitively clear:

- $\vec{R}$  will always lie in the plane containing the point particles.
- $\vec{R}$  will always lie in the triangle  $T$  with vertices  $\{\vec{r}_i\}$ .

This means that by changing  $\{m_i\}$ ,  $\vec{R}$  will span all set of points in  $T$ . This is the motivation behind barycentric coordinates, the coordinate system employed by Möller and Trumbore. In particular, to represent a vector  $\vec{v}$  pointing to the surface of a triangle  $T \in \mathbb{R}^3$  with vertices  $\vec{v}_0$ ,  $\vec{v}_1$ , and  $\vec{v}_2$ , barycentric coordinates can be utilized. These coordinates  $\{\alpha, \beta, \gamma\} \in \mathbb{R}$  act as the masses of the vertices of  $T$ . Thus, any point  $\vec{v}$  along the plane  $P$  of  $T$  can be represented by

$$\vec{v} = \alpha\vec{v}_0 + \beta\vec{v}_1 + \gamma\vec{v}_2. \quad (1.7)$$

For  $\vec{v}$  to be located on or within  $T$ , the following requirements on the barycentric coordinates must be satisfied:

$$\bullet \quad \alpha + \beta + \gamma = 1 \quad (1.8)$$

$$\bullet \quad \alpha, \beta, \gamma \geq 0 \quad (1.9)$$

With regard to the center of mass interpretation, Equation 1.8 removes the computation of the denominator in Equation 1.6, while Equation 1.9 eliminates the possibility of negative mass. For a rigorous proof of the requirements, see (find something).

### Möller-Trumbore Intersection Algorithm

The algorithm consists of 2 main steps:

1. Check if ray intersects with  $P$ ;
2. If so, check if intersection lies outside of  $T$ .

1. To check if the ray described by Equation 1.1 intersects with  $P$ , one can define two vectors that lie along 2 edges of  $T$ :

$$\vec{e}_1 = \vec{v}_1 - \vec{v}_0; \quad (1.10)$$

$$\vec{e}_2 = \vec{v}_2 - \vec{v}_0. \quad (1.11)$$

If  $\vec{p} = \hat{d} \times \vec{e}_2$ , then let  $C = \vec{e}_1 \cdot \vec{p}$ . The line  $\vec{r}$  does not intersect the plane  $P$  iff  $C = 0$ , which happens iff  $\hat{d} \parallel P$ . Thus, if  $C = 0$ , then  $\vec{r}$  does not intersect  $P$ ; otherwise,  $\vec{r}$  will intersect  $P$ .

2. Solving Equation 1.8 for  $\alpha$  and substituting into Equation 1.7 results in

$$\vec{v} = (1 - \beta - \gamma)\vec{v}_0 + \beta\vec{v}_1 + \gamma\vec{v}_2. \quad (1.12)$$

After distributing and rewriting Equation 1.12 in terms of  $\vec{e}_1$  and  $\vec{e}_2$ ,

$$\vec{v} = \vec{v}_0 + \beta\vec{e}_1 + \gamma\vec{e}_2. \quad (1.13)$$

Setting  $\vec{r} = \vec{v}$  to find the barycentric coordinates for the ray-triangle intersection and solving for  $\vec{r} - \vec{v}_0$  yields

$$\vec{r} - \vec{v}_0 = -\hat{d} + \beta\vec{e}_1 + \gamma\vec{e}_2. \quad (1.14)$$

One can then show that taking  $(\vec{r} - \vec{v}_0) \cdot \vec{p}$  gives the following equation for  $\beta$ :

$$\beta = \frac{(\vec{r} - \vec{v}_0) \cdot \vec{p}}{C}, \quad (1.15)$$

and similarly, taking  $\hat{d} \cdot [(\vec{r} - \vec{v}_0) \times \vec{e}_1]$  results in

$$\gamma = \frac{\hat{d} \cdot [(\vec{r} - \vec{v}_0) \times \vec{e}_1]}{C}. \quad (1.16)$$

To then determine  $t$  at which the ray-triangle intersection occurs, one can take  $\vec{e}_2 \cdot [(\vec{r} - \vec{v}_0) \times \vec{e}_1]$ , yielding

$$t = \frac{\vec{e}_2 \cdot [(\vec{r} - \vec{v}_0) \times \vec{e}_1]}{C}. \quad (1.17)$$

```

1  bool triangle::hit(const ray &r, float t_min, float t_max, hit_record &rec) const {
2      float kEpsilon = 1e-9;
3
4      vec3 e1 = v1 - v0;
5      vec3 e2 = v2 - v0;
6      vec3 pvec = cross(r.direction(), e2);
7      float C = dot(e1, pvec);
8
9      // ray is parallel to triangle
10     if (std::abs(C) < kEpsilon) return false;
11
12     float inv_det = 1.0 / C;
13
14     vec3 tvec = r.origin() - v0;
15     float u = dot(tvec, pvec) * C;
16
17     // hit point is outside of triangle
18     if (u < 0.0 || u > 1.0) return false;
19
20     vec3 qvec = cross(tvec, e1);
21     float v = dot(r.direction(), qvec) * C;
22
23     // hit point is outside of triangle
24     if (v < 0.0 || u + v > 1.0) return false;
25
26     float t = dot(e2, qvec) * C;
27
28     // hit point is outside of ray
29     if (t < t_min || t > t_max) return false;
30
31     // record intersection point
32     rec.t.push_back(t);
33     rec.p.push_back(r.at(t));
34
35     return true;
36 }

```

Listing 1: C++ implementation of the Möller-Trumbore ray-triangle intersection algorithm.



## Monte Carlo Approach

### 2.1 Transport Theory

To represent the computational domain discretely, space is broken up into a grid of voxels (volume pixel), with each voxel being assigned a particular material depending on the geometries and compounds/elements in the domain. Within our discrete space, given a photon position  $\vec{r}$ , the corresponding voxel in which the photon resides can be calculated. Therefore, all possible  $\vec{r}$ 's can be assigned a particular material  $M$  in the domain.

A photon's position after taking the  $n$ -th step in the domain,  $\vec{r}_n$ , is represented by the parametric ray equation:

$$\vec{r}_n = \vec{r}_{n-1} + \hat{d}t_n, \quad (2.1)$$

where  $\vec{r}_{n-1}$  is the initial position before the  $n$ -th step,  $\hat{d}$  is a unit vector in the direction of the step, and  $t_n$  is the length of the  $n$ -th step.

To randomly sample  $t_n$  in a homogeneous domain, we utilize the following probability density function (PDF)  $p(t)$  of the distance traveled  $t$  by a photon of energy  $E$  through material  $M$  before interacting:

$$p(t) = n\sigma \exp[-t(n\sigma)], \quad (2.2)$$

where  $n$  is the number density of  $M$  and  $\sigma = \sigma(E, M)$  is the microscopic cross-section of  $M$  at  $E$ .

Using the inversion method for sampling a PDF on Equation 2.2, random values of the free path  $t$  can be generated by

$$t = -\frac{1}{n\sigma} \ln \gamma, \quad (2.3)$$

where  $\gamma$  is a uniformly distributed random number in the interval  $[0, 1)$ . This value of  $t$  is sampled for each step and is used as  $t_n$  in Equation 2.1 to determine the length of the  $n$ -th step [vassiliev\_monte\_2017].

### 2.1.1 Surface & $\delta$ -Tracking

If, after taking a step, the photon lands in a voxel with a different material (an inhomogeneous domain), then the corresponding free path for the new material must be accounted for. One method, called surface-tracking, requires photons to be stopped at voxel boundaries and intersections with surrounding voxels to be calculated, which can be computationally intensive for materials that have a large average free path.

Alternatively, the  $\delta$ -tracking algorithm offers a solution by sampling the maximum cross-section  $\sigma_{\max}$  in the computational domain. This, in turn, brings down the average free path to the minimum in the domain. To account for this decrease in free path, the algorithm introduces  $\delta$  interactions as an alternative to real interactions, resulting in no change to the energy or direction. The probability of a  $\delta$  interaction  $P_\delta$  is given by

$$P_\delta = \frac{\sigma_{\max}(E) - \sigma(E, M)}{\sigma_{\max}(E)}, \quad (2.4)$$

where  $E$  is the energy of the photon undergoing the step. When the photon lands in the material corresponding to the maximum cross-section,  $\sigma(E, M) = \sigma_{\max}$  and  $P_\delta = 0$ . On the contrary, if the photon landed in air and the domain's maximum cross-section corresponded to lead, then  $\sigma(E, M) \ll \sigma_{\max}$ , making  $P_\delta \approx 1$ .

Overall,  $\delta$ -tracking is significantly more computationally efficient for domains with similar cross-sections and can be shown to yield equivalent results to surface-tracking [vassiliev\_monte\_2017].

### 2.1.2 Photon-Matter Interactions

If a  $\delta$  interaction does not occur, then a real interaction is sampled. Therefore, the probability of a real interaction  $P_r$  is directly related to  $P_\delta$  by

$$P_r = 1 - P_\delta. \quad (2.5)$$

If a real interaction occurs in material  $M$ , then the probability of interaction  $i$  occurring is

$$P_i = \frac{\sigma_i(E, M)}{\sigma(E, M)} \quad (2.6)$$

where  $\sigma_i$  is the cross-section of interaction  $i$ . If there are  $N$  possible interactions for a particular  $E$  and  $M$ , then  $\sigma(E, M)$  is simply the sum of each interaction's cross-section.

For x-rays, there are three possible photon interactions: photoelectric effect, coherent scattering, and incoherent scattering.

#### 2.1.2.1 Photoelectric Effect

In the photoelectric effect model used in MIDSX, a rather simple approach is taken. When a photon interacts with an atom's electron, the photon is terminated and all energy is deposited at the location of interaction. In general purpose particle transport code systems,

when a photoelectric interaction occurs, a photon of energy  $E$  is absorbed by an electron in subshell  $i$ , causing the electron to leave the atom with energy  $E_e = E - U_i$ , where  $U_i$  is the binding energy of the  $i$ th subshell. In addition, photons are emitted due to atomic relaxations. For photon energies in the medical imaging range (30 - 120 keV), the energy of the released electrons does not allow for significant traversal through typically encountered materials, such as tissue, bone, and fat. This limited traversal results in a localized dose distribution, in turn, validating the model used by MIDSX.

### 2.1.2.2 Coherent Scattering

Thomson scattering is defined as an incoming photon of energy  $E$  elastically scattering with a free electron at rest, resulting in a scattered photon of same energy  $E$ . The atomic differential cross-section (DCS) per unit solid angle  $\Omega$  for the interaction can be derived with classical electrodynamics, and is given by

$$\frac{d\sigma_T}{d\Omega} = r_e^2 \frac{1 + \cos^2(\theta)}{2}, \quad (2.7)$$

where  $r_e^2$  is the classical electron radius.

In an atom, photons scatter off bound electrons rather than the free electrons described by Thomson scattering, resulting in what is known as coherent (Rayleigh) scattering. The DCS per unit solid angle  $\Omega$  of the interaction, ignoring absorption edge effects, is given by

$$\frac{d\sigma_{Co}}{d\Omega} = \frac{d\sigma_T}{d\Omega} F(x, Z), \quad (2.8)$$

where  $x$  is the momentum transfer between the photon and atom,  $Z$  is the atomic number of the atom, and  $F(x, Z)$  is the atomic form factor.  $x$  is related to the scattering angle  $\theta$  by

$$x = ak\sqrt{1 - \cos\theta}, \quad (2.9)$$

where

$$a = \frac{m_e c^2}{\sqrt{2}hc} \quad \text{and} \quad k = \frac{E}{m_e c^2}, \quad (2.10)$$

In these definitions,  $m_e$  is the mass of an electron,  $c$  is the speed of light, and  $h$  is Planck's constant.

Equation 2.8 can be integrated over  $\phi$  to obtain the DCS per unit polar angle  $\theta$ :

$$\frac{d\sigma_{Co}}{d\theta} = \pi r_e^2 \sin\theta (1 + \cos^2\theta) F(x, Z)^2. \quad (2.11)$$

The PDF of the polar angle  $\theta$  is then given by

$$p(\theta)d\theta = \frac{d\sigma_{Co}}{d\theta} \frac{1}{\sigma_{Co}} d\theta = \frac{\pi r_e^2}{\sigma_{Co}} \sin\theta (1 + \cos^2\theta) F(x, Z)^2 d\theta. \quad (2.12)$$

The PDF of  $\theta$  can be transformed into a PDF of  $\mu = \cos \theta$ , resulting in

$$p(\mu) = \frac{\pi r_e^2}{\sigma_{Co}} (1 + \mu^2) F(x, Z)^2. \quad (2.13)$$

To then sample  $\mu$  for a particular scattering event, the inversion method is used. In particular, a look up table for the CDF of  $P(\mu)$  is generated for each material in the domain for a grid of  $\mu$  values. The details of this algorithm are discussed in Appendix ...

### 2.1.2.3 Incoherent Scattering

Incoherent (Compton) scattering is defined as an incoming photon of energy  $E$  interacting with an atom's electron, resulting in a scattered photon of energy  $E'$  and a released electron with energy  $E_e = E - E' - U_i$ , where  $U_i$  is the binding energy of the interacting subshell. While coherent scattering effectively interacts with the entire electron cloud, incoherent scattering interacts with an individual electron. The DCS per unit solid angle  $\Omega$  of the interaction was derived by Klein and Nishina in 1929, making it one of the first results of quantum electrodynamics. The Klein-Nishina formula is given by

$$\frac{d\sigma_{KN}}{d\Omega} = \frac{r_e^2}{2} \left( \frac{E'}{E} \right)^2 \left( \frac{E'}{E} + \frac{E}{E'} - \sin^2 \theta \right), \quad (2.14)$$

Note that when  $E' = E$ , the Equation 2.14 simplifies to Equation 2.8, showing that incoherent scattering is a generalization of coherent scattering for inelastic interactions.

Applying conservation of energy and momentum to a free electron at rest, the following equation can be derived relating the scattered photon energy  $E'$  to the scattering angle  $\theta$  and the incident photon energy  $E$ :

$$E' = \frac{E}{1 + k(1 - \cos \theta)}. \quad (2.15)$$

Similar to the Thomson DCS, the Klein-Nishina DCS assumes a free electron at rest. In an atom, the electron is bound, resulting in a modified DCS. In the case of incoherent scattering, the Klein-Nishina DCS is modified by the Coherent Scattering Function, making the DCS per unit solid angle  $\Omega$  of the interaction

$$\frac{d\sigma_{In}}{d\Omega} = \frac{d\sigma_{KN}}{d\Omega} S(x, Z). \quad (2.16)$$

Instead of directly sampling the PDF of the DCS,  $\mu$  is first sampled using the acceptance-rejection method developed by Ozmutlu, then  $S(x, Z)$  is sampled once again with the acceptance-rejection method. The details of this algorithm are discussed in Appendix ...

## 2.2 Methods

In order to simulate the transport of x-rays through a domain, the following is required:

1. The geometry of the domain
2. The materials in the domain
3. The associated cross-sections, form factors, and scattering functions of the materials
4. The source of the x-rays (energy spectrum, position, and direction)
5. The number of photons to simulate

and to retrieve information about the performed simulation, the following is required:

1. Geometries to check for intersection
2. Quantities to tally
3. Derived quantities to calculate from tallied quantities

### 2.2.1 Geometry

In MIDSX, the geometry of the domain is represented by a 3D array of voxels, with each voxel being assigned a material ID. The domain is assigned a particular size, indicated by its spatial extent along the  $x$ ,  $y$ , and  $z$  dimensions. In addition, the domain is assigned a background material ID. Geometries inside the domain are specified by NIFTI files [nifti2004]. These NIFTI files are assigned material IDs, spatial size, voxel size, and an origin, which is the location of the voxel in the domain that corresponds to the origin of the NIFTI file. After constructing the NIFTI files, the domain is defined by supplying the background material ID, domain size, and a list of NIFTI files into a .JSON file, which is then read by the MIDSX executable.

In the code, both a `VoxelGrid` and `ComputationalDomain` object are created, with the `ComputationalDomain` consisting of the specified dimensions, background material ID, and a vector of `VoxelGrid` objects which are created via the provided NIFTI files. To determine the current material of a photon, it first checks if the photon is inside the `ComputationalDomain`. If so, it then checks if the photon is inside any of the `VoxelGrid` objects. If so, it is then determined which `Voxel` object within the `VoxelGrid` the photon is located inside, then the corresponding material ID is returned. If the photon is not inside the `ComputationalDomain`, then the photon is terminated.

### 2.2.2 Materials & Data

In MIDSX, materials are all defined in an SQLite database [**sqlite2020hipp**], which is read by the MIDSX executable. The database contains the following information for each element:

1. Symbol
2. Atomic Number
3. Mass
4. Mass Density
5. Number Density
6. Mass Number

All of the above data was obtained from the periodictable [**periodictable2022**] and mendeleeve [**mendeleeve2021**] python packages. In addition, the database contains the following data from the Electron Photon Interaction Cross Sections (EPDL2017) library [**cullen\_survey\_nodate**] for each element:

1. Total Microscopic Cross-Section
2. Photoelectric Microscopic Cross-Section
3. Coherent Scattering Microscopic Cross-Section
4. Incoherent Scattering Microscopic Cross-Section
5. Atomic Form Factor
6. Scattering Function

In MIDSX, all the above data is initialized upon creation of the `InteractionData` object with a vector of strings of materials names. These material names correspond to entries in the SQLite database, which contains a table of material compositions and mass densities which were obtained from NIST's mass attenuation coefficient database [**hubbell\_x-ray\_2004**]. The `InteractionData` object contains a map of `Material` objects with their names, along with additional computed data, such as the maximum cross-section which is used for delta-tracking. The `Material` objects construct the above data for the specified material by performing an additivity approximation of the data for each element in the material as described by [**HUSSEIN2007153**]. The data is separated into two further objects: `MaterialData` and `MaterialProperties`. `MaterialData` contains the microscopic cross-sections, form factors, and scattering functions, while `MaterialProperties` contains the mass density, number density, and mass.

In addition to retrieving and storing the above data, the `MaterialData` object constructs interpolators for all its data. The interpolators for each type of data vary depending on its shape. For example, the photoelectric and total cross-sections are interpolated with a log-log linear interpolator, while the incoherent and coherent cross-sections are interpolated with a log-log cubic spline interpolator.

### 2.2.3 Source

To generate the initial position, direction, and energy of a photon, a `PhotonSource` object is initialized with a `SourceGeometry`, `Directionality`, and an `EnergySpectrum` object. The three initializing object are virtual classes, allowing the user to specify the attributes of the source. The inheritance structure is shown in Fig ().

### 2.2.4 Tallies

To measure simulation data, one must first decide when to trigger the measurement. In particular, MIDSX supports both surface and volume geometries that trigger when a photon passes through or enters the geometry, respectfully. For surfaces, users can choose discs and rectangles, while for volumes, there are cuboids. In the code, these geometries are defined by `Tally` objects.

At the start of the simulations, users can specify quantities that they want measured upon the trigger of a `Tally`. For surfaces, these include incident photon energy, entrance cosine, and number of photons. For volumes, in addition to their own implementation of incident energy and number of photons, energy deposition and number of interactions are available for measurement. Furthermore, each quantity has the ability to group together measurements that were triggered by photons that underwent a single coherent scatter, a single incoherent scatter, multiple scatters, and no scatters. While this does noticeably increase the computation time, it was necessary to validate the interaction models discussed in Section 2.1.

At the end of simulation, one might want to calculate additional derived quantities from the tallied quantities from the simulation. With this mind, MIDSX contains a `DerivedQuantity` object that can calculate planar fluence and air kerma, which were implemented for use in half value layer simulations.

Using the described theory and methodology, the object `Transporter` transports a photon through the computational domain until termination. This process is summarized in Figure 2.1.

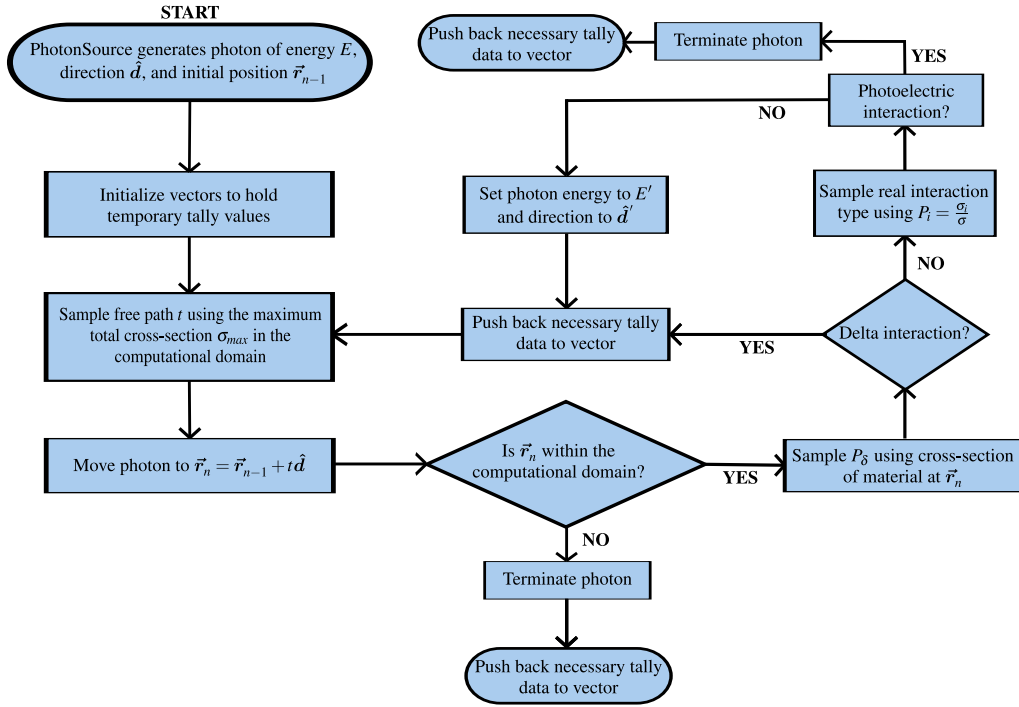


Figure 2.1: Photon transport process in MIDSX.

## 2.3 Validation

To validate the accuracy of MIDSX, validation simulations were performed and compared to reference data (PENELOPE, EGSrc, Geant4, and MCNP) obtained by the American Association of Physicists in Medicine Task Group Report 195 (TG-195) [sechopoulos\_monte\_2015]. The simulations performed from TG-195 were Case 1: "Half Value Layer," Case 2: "Radiography and Body Tomosynthesis," and Case 5: "CT with a Voxelized Solid." For Case 1, the primary air kerma was measured on a far away, circular region of interest (ROI) with a cone beam point source collimated such that all primary particles would be incident upon the ROI. The primary air kerma was measured with the domain filled only with air and then compared to the measured air kerma with an aluminum filter of thickness  $t$  placed between the source and ROI. The ratios of the half value layer (HVL) and quarter value layer (QVL) primary air kerma to the primary background air kerma is represented by  $R_1$  and  $R_2$ , respectively. By setting  $t$  to correspond to the HVL and QVL for a particular spectrum, one can validate the material attenuation properties of an MC code system by comparing the simulated  $R_1$  and  $R_2$  to their theoretical values of 0.5 and 0.25, respectively. The simulation was performed for the monoenergetic energies of 30 keV and 100 keV, along with the polyenergetic spectrums of 30 kVp and 100 kVp, which were provided by TG-195. The term kVp (kilovolt peak) refers to the maximum voltage applied to an x-ray tube, which determines the highest energy of the x-rays produced. MIDSX's results for Case 1 agree to within 0.39% of the mean results published by TG-195 (not shown).

For Case 2, a full-field and pencil beam x-ray source were directed towards a cuboid



tissue phantom at  $0^\circ$  and  $15^\circ$  from an axis drawn perpendicularly from the center of the cuboid. Directly behind and inside the phantom, a grid of square ROIs and cube volumes of interests (VOI) were placed, respectively. The simulation was performed for the TG-195-provided polyenergetic spectrum of 120 kVp and its mean energy of 56.4 keV. For the  $0^\circ$ , full-field ROI measurements (not shown), a  $< 3.9\%$  mean percent error (MPE) is seen for MIDSX's results to each ROI simulation. Furthermore, for the  $0^\circ$  pencil-beam ROI measurements shown in Figure 2.5, a  $< 2.1\%$  MPE is observed for each ROI simulation except for the case of a single incoherent scatter. In this particular case, MIDSX's results for ROI 4 and 5 are significantly lower, with the MPE reaching 13.1% for ROI 5. The full-field VOI energy deposition measurements depicted in Figure 2.3 show a minimal MPE of less than 0.1% for the  $0^\circ$  source. Conversely, for the MIDSX results at  $15^\circ$ , the MPE reaches an unexpectedly larger value of approximately 0.6%.

For Case 5, a fan beam was collimated to the center of a voxelized human torso phantom provided by TG-195. To replicate a CT image, the simulation was repeated for several angles along a circle surrounding the phantom. The simulation was performed with Case 2's 120 kVp energy spectrum, and energy deposition was measured in the different materials/organs composing the phantom. Almost all of MIDSX's results for the  $0^\circ$  source presented in Figure 2.8 are systematically lower than the mean of the reference code systems, with MPE's ranging from 1.1% to 6.3%. This pattern is disrupted by the thyroid, which is larger than the mean by 2.9%.

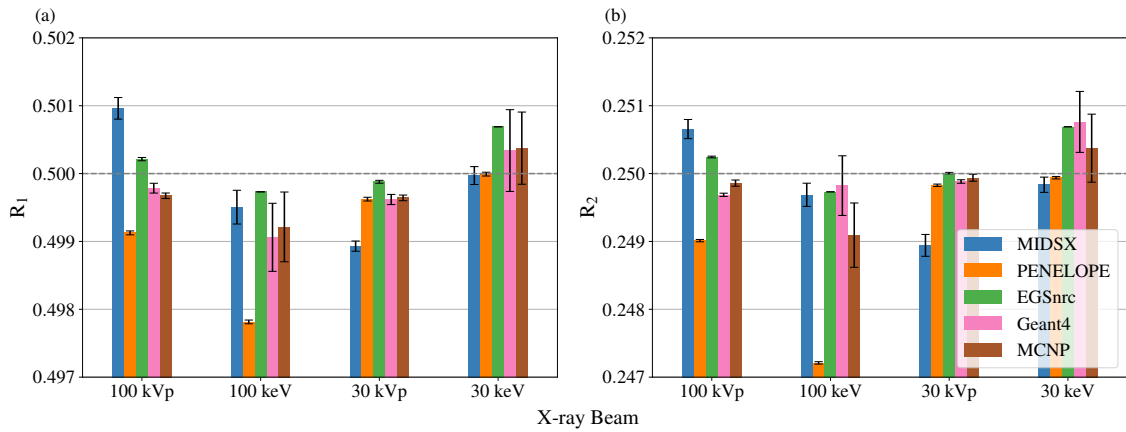


Figure 2.2: Results for the (a) HVL and (b) QVL simulations as described by Case 1. The ratios of the primary HVL and QVL air kermas to the primary background air kermas is represented by  $R_1$  and  $R_2$ , respectively. The simulation was performed for the monoenergetic energies 30 keV and 100 keV, along with the polyenergetic spectrums of 30 kVp and 100 kVp provided by TG-195. A dashed-line is placed at  $R_1 = 0.5$  and  $R_2 = 0.25$  for comparison.

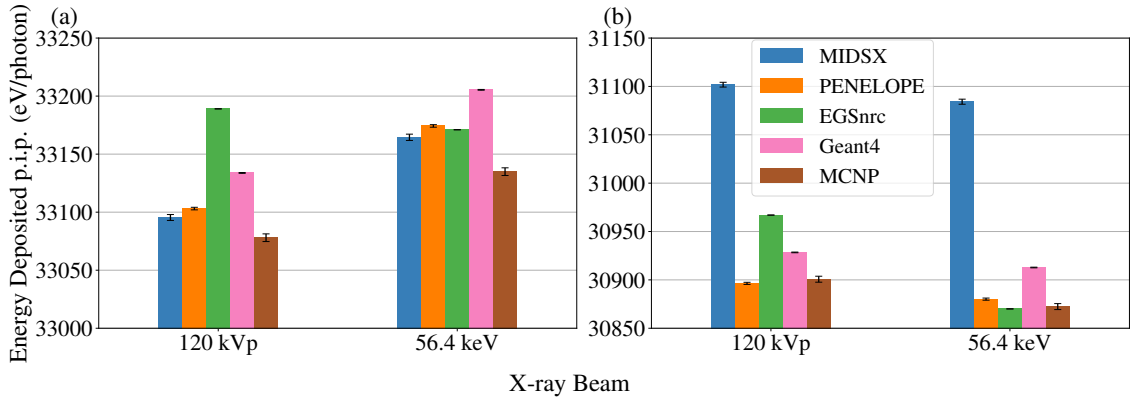


Figure 2.3: The energy deposited per initial photon (p.i.p.) (eV/photon) in the simulated tissue for the full-field simulation as described by Case 2. The simulation was performed at 56.4 keV and 120 kVp at both (a) 0° and (b) 15°, with the 120 kVp spectrum provided by TG-195.

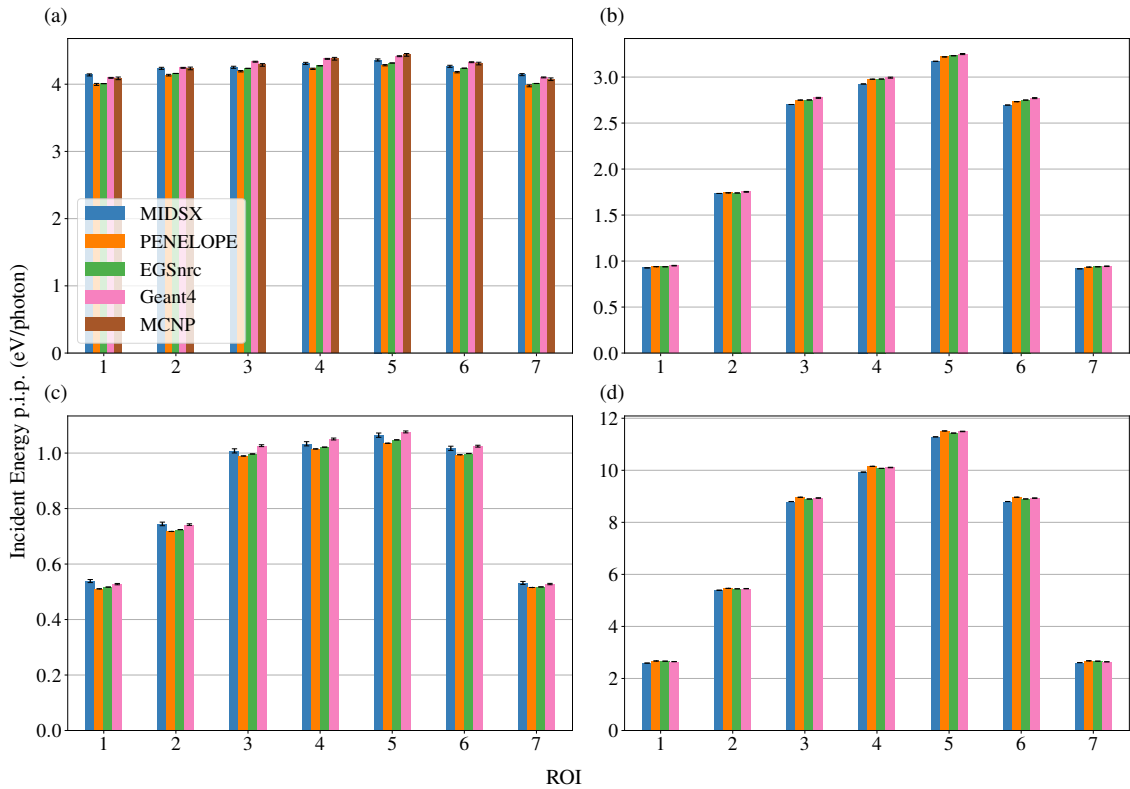


Figure 2.4: The energy per initial photon (eV/photon) of photons incident upon each region of interest (ROI) for the 0°, full-field, 56.4 keV simulation as described by Case 2. The incident energy was determined separately for photons that underwent (a) no real interactions, (b) a single incoherent scatter, (c) a single coherent scatter, (d) and multiple scatters.

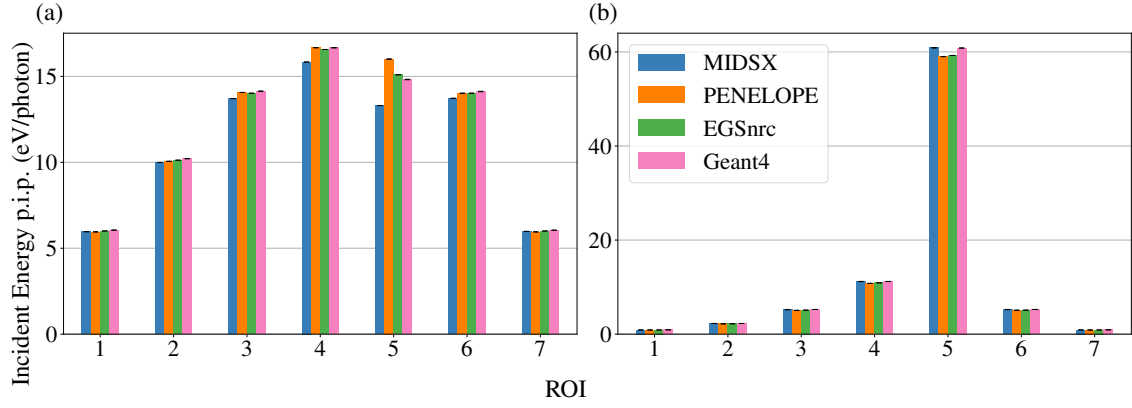


Figure 2.5: The energy per initial photon (p.i.p.) (eV/photon) of photons incident upon each region of interest (ROI) for the 0°, pencil beam, 56.4 keV simulation as described by Case 2. The incident energy was determined separately for photons that underwent (a) a single incoherent scatter and (b) a single coherent scatter.

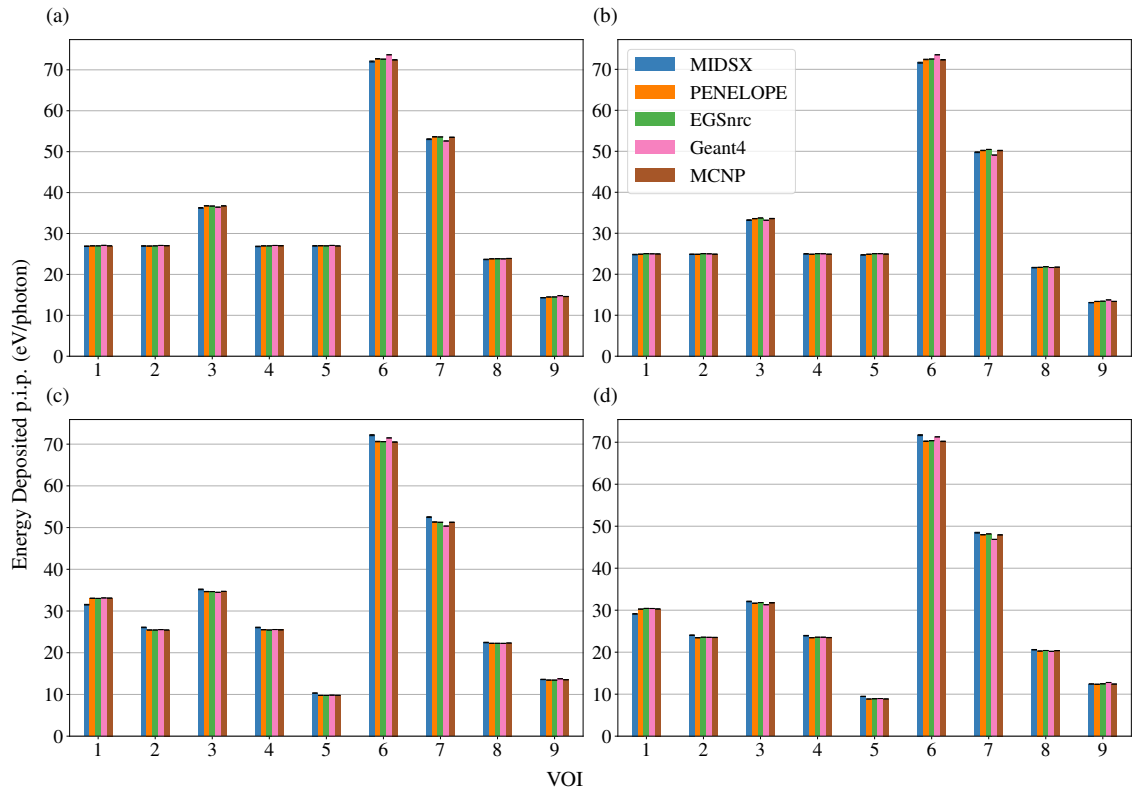


Figure 2.6: The energy deposited per initial photon (eV/photon) in the volumes of interests (VOIs) for the full field simulation as described by Case 2. The simulation was performed at (a) 0° / 56.4 keV, (b) 0° / 120 kVp, (c) 15° / 56.4 keV, and (d) 15° / 120 kVp.

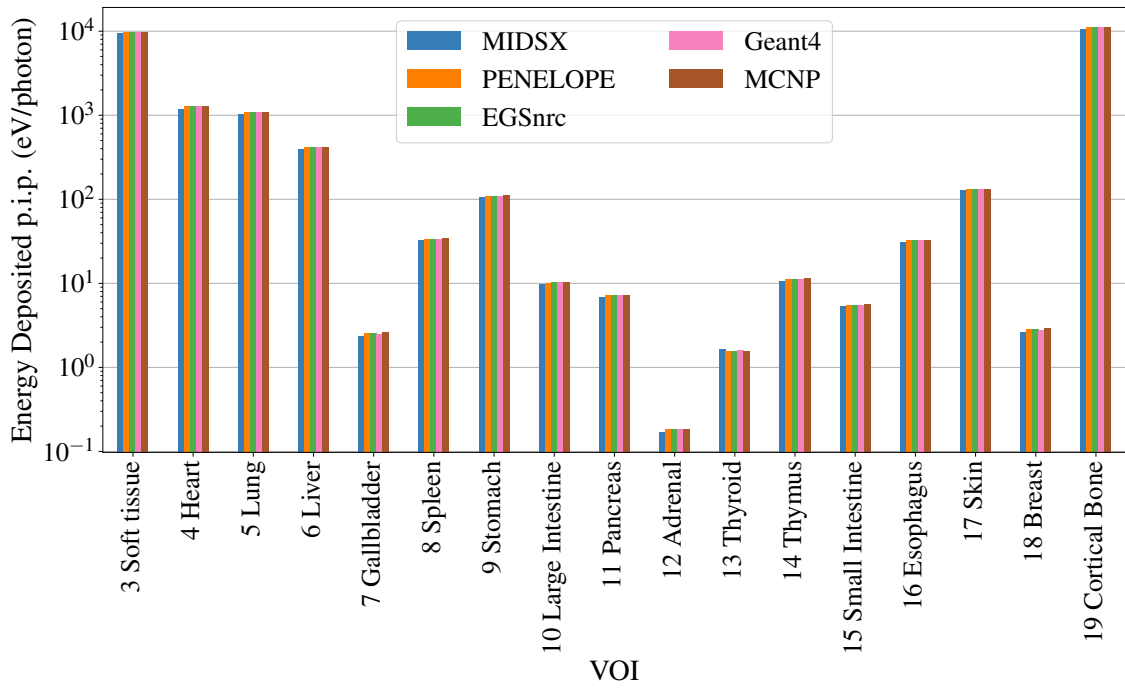


Figure 2.7: The energy deposited per initial photon (eV/photon) in the material IDs for the  $180^\circ$ , 56.4 keV simulation as described by Case 5.

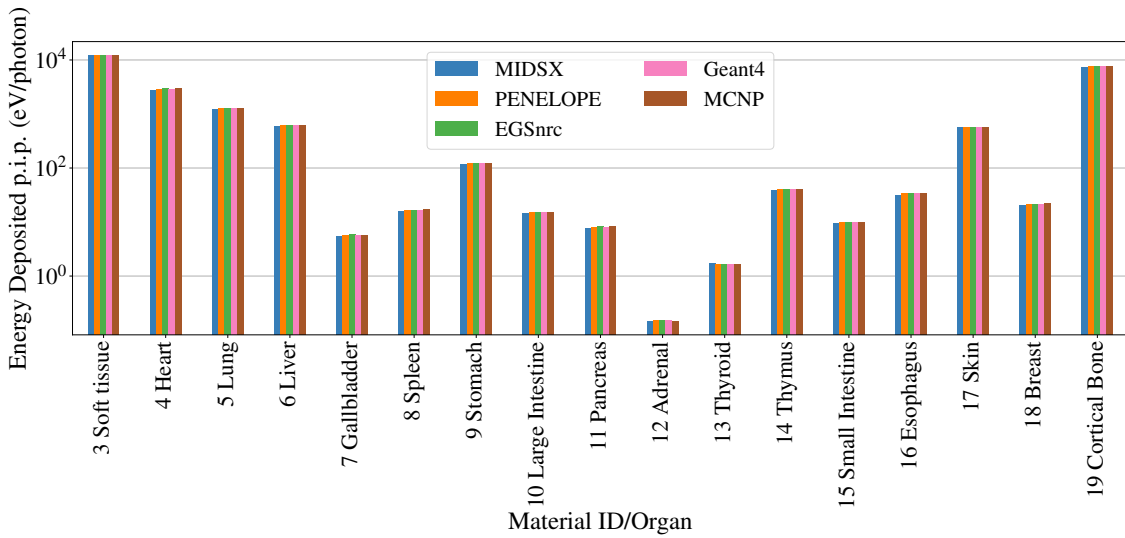


Figure 2.8: The energy deposited per initial photon (p.i.p.) (eV/photon) in the material IDs/organs composing a voxelized human phantom for the  $0^\circ$ , 120 kVp simulation as described by Case 5.

## 2.4 Discussion

Overall, MIDSX shows varied but reasonable agreement with the reference code systems of TG-195. For the HVL layer measurements described by Case 1, shown in Figure 2.2, statistical agreement is seen with all code systems except for PENELOPE in the HVL and QVL 100 keV simulation. In addition, for the 30 keV simulation, statistical agreement is seen with PENELOPE, Geant4, and MCNP for the HVL, along with PENELOPE and MCNP for the QVL. Note that while no statistical agreement is observed for the other energies/thicknesses, all MIDSX results have a mean percent error (MPE) of  $< 0.32\%$ .

For the ROIs of Case 2, agreement is seen almost universally, except for the single incoherent scatter deposition energy in ROI 4 and 5 for the pencil beam source shown in Figure 2.5. In particular, the results reached a max MPE of 13.1% for ROI 5, indicating a potential error in the incoherent scattering energy/angular distribution sampling algorithm. This discrepancy is likely a result of an error in the rejection sampling algorithm employed by MIDSX. While this algorithm shows agreement for the full-field ROI 5, the geometry of the ROI, combined with the pencil beam, results in only narrow-angle scatters hitting the ROI. Since the scattering angle distribution of incoherent scattering at the medical imaging energy range becomes extremely steep at the scattering angle  $\theta = 0^\circ$ , there is likely numerical instability presenting itself in the algorithm that needs to be analyzed. In addition, the  $15^\circ$  full-field tissue energy deposition measurements, shown in Figure 2.3, were larger than the reference code systems', with an MPE of 0.6%. With the MPE increasing significantly from the  $0^\circ$  to  $15^\circ$  measurements, this hints at a possible geometric error with the source and/or body. However, the source's position and angular distribution, along with the domain's and tissue's dimensions, were carefully verified, making the scene geometry unlikely to be the source of discrepancy.

For Case 5, almost all organ energy deposition results were lower than the reference code results, with the MPE reaching 6.3%, except for the thyroid, with an MPE of 2.9% larger. One common error reported by TG-195 that could result in the observed discrepancies is the incorrect orientation of the voxelized phantom in the computational domain. The orientation was verified by taking the root mean square percent error (RMSPE) of the MIDSX data with respect to the results of each simulated angle reported by TG-195. As expected, the RMSPE with respect to  $0^\circ$  was the minimum, verifying that the phantom's orientation during the CT simulation was correct.

Despite verifying the orientation, the deviation of MIDSX's energy deposition results for both Cases 2 and 5 suggest that there may be other underlying errors in the MIDSX system that need further investigation. Potential factors could include the software's handling of scattering events, cross-section data initialization, and interpolation. Future work will study these aspects to pinpoint and rectify the source of the systematic errors observed in the MIDSX results.

