2020

Parallelized X-Ray Tracing with GPU Ray-Tracing Engine

Joseph Ulseth
University of Central Florida

Part of the Electromagnetics and Photonics Commons, and the Optics Commons

Find similar works at: https://stars.library.ucf.edu/etd2020
University of Central Florida Libraries http://library.ucf.edu

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2020- by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation
https://stars.library.ucf.edu/etd2020/144
PARALLELIZED X-RAY TRACING WITH GPU RAY-TRACING ENGINE

by

JOSEPH ULSETH
B.S. Astronomy, University of Florida, 2015
B.A. Physics, University of Florida, 2015
B.A. Mathematics, University of Florida, 2015

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Sciences
in the College of Optics and Photonics
at the University of Central Florida
Orlando, Florida

Spring Term
2020

Major Professor: Sean Pang
ABSTRACT

X-ray diffraction tomography (XDT) is used to probe material composition of objects, providing improved contrast between materials compared to conventional transmission based computed tomography (CT). In this work, a small angle approximation to Bragg’s Equation of diffraction is coupled with parallelized computing using Graphics Processing Units (GPUs) to accelerate XDT simulations. The approximation gives rise to a simple yet useful proportionality between momentum transfer, radial distance of diffracted signal with respect to incoming beam’s location, and depth of material, so that ray tracing may be parallelized. NVIDIA’s OptiX ray-tracing engine, a parallelized pipeline for GPUs, is employed to perform XDT by tracing rays in a virtual space, \((x,y,z_v)\), where \(z_v\) is a virtual distance proportional to momentum transfer. The advantage gained in this approach is that ray tracing in this domain requires only 3D surface meshes, yielding calculations without the need of voxels. The simulated XDT projections demonstrate high consistency with voxel models, with a normalized mean square difference less than 0.66\%, and ray-tracing times two orders of magnitude less than previously reported voxel-based GPU ray tracing results. Due to an accelerated simulation time, XDT projections of objects with three spatial dimensions (4D tensor) have also been reported, demonstrating the feasibility for largescale high-dimensional tensor tomography simulations.
ACKNOWLEDGMENTS

I would like to acknowledge a friend and colleague whose previous work made this thesis a possibility. Thank you Zheyuan Zhu for your guidance and patience.
# TABLE OF CONTENTS

LIST OF FIGURES .................................................................................................................. v

LIST OF TABLES ...................................................................................................................... vi

LIST OF ACRONYMS ............................................................................................................ vii

INTRODUCTION AND BACKGROUND ............................................................................. 1

RELEVANT XDT THEORY .................................................................................................... 4

SIMULATION SETUP, METHODS, AND CONFIGURATION ........................................ 10

RESULTS ................................................................................................................................. 15

  Accuracy Comparison to Voxel Model ........................................................................... 15

  Execution Time ............................................................................................................... 18

  Simulation of Objects with Three Spatial Dimensions ................................................. 20

DISCUSSION ........................................................................................................................... 22

CONCLUSION ......................................................................................................................... 24

APPENDIX: COPYRIGHT PERMISSION ........................................................................... 25

REFERENCES ......................................................................................................................... 27
LIST OF FIGURES

Figure 1: (a) Experimental setup of circular object with radius $r$ positioned a distance $L$ from a detector array located at plane $y = L$. (b) The same 2D object shown in (a) but represented as a virtual 3D object by extruding the object’s physical 2D cross-section into a virtual $z$-direction that is proportional to $q$ for each specified $w$. © 2019 IEEE................................................................. 5

Figure 2: Validity of the linear approximation of $q$ for small objects, and low radial distances $w$ on detector. © 2019 IEEE................................................................. 7

Figure 3: Virtual scene phantom composed of 4 materials. (a) Coherent scatter profiles of the materials used. (b) 2D slice of scene at $\phi = 0^\circ$. (c) Mesh view of the 4 materials, displaying all triangle primitives. (d) 3D virtual view of all surfaces. © 2019 IEEE................................................................. 11

Figure 4: Flow chart of basic ray-tracing technique implemented with OptiX......................... 14

Figure 5: Sinograms comparing OptiX to voxel model and experimental data at various detector heights $z_v = (a)$ 6 mm, (b) 22 mm, (c) 38 mm on detector. © 2019 IEEE.............................. 17

Figure 6: Line slices through the virtual projection at $\phi = 0^\circ$ for OptiX (solid black), Voxel (dashed), and experimental (blue) results at translations (a) (b) (c) (d). © 2019 IEEE............... 17

Figure 7: Scene reconstructions of OptiX compared to Voxel model, at evenly spaced momentum transfers. © 2019 IEEE................................................................. 18

Figure 8: 3D scene composed of Teflon bolts and an acrylic nut. © 2019 IEEE......................... 23
LIST OF TABLES

Table 1: Execution Timing for Various Configurations of Simulating 180 XDT Projections..... 19
# LIST OF ACRONYMS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT</td>
<td>Computed Tomography</td>
</tr>
<tr>
<td>FDK</td>
<td>Feldkamp-Davis-Kress (reconstruction)</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphics Processing Unit</td>
</tr>
<tr>
<td>XDT</td>
<td>X-ray Diffraction Tomography</td>
</tr>
</tbody>
</table>
INTRODUCTION AND BACKGROUND

X-ray diffraction (XRD) is used in a wide variety of applications to determine properties of essentially any material, such as structure, crystal orientation, stress, and phase analysis. Making use of the structural information from the small-angle scatter signal, the presence of specific materials may be uniquely identified. Applications of XRD imaging include classification of biological tissues [1] [2], airport checkpoint baggage screening [3], and compound structural analysis in pharmaceutical science [4]. Similar to computed tomography (CT) where the attenuation cross-section of a volumetric object is reconstructed from a series of projections, x-ray diffraction tomography (XDT) reconstructs the XRD signatures at each voxel [5]. The XRD signatures, in general, lie in the 3D reciprocal space, \((q_x, q_y, q_z)\). To match with the dimension of the object (4-6 dimensional tensor), the projection acquisition needs to introduce extra degrees of freedom, making the acquisition and tensor reconstruction processes extremely time-consuming [6], [7], [8].

Recently, the speed of XDT image acquisition has been greatly increased by combining novel system with signal multiplexing [9] with region of interest (ROI) scanning [10]. However, limited by the availability of XDT projection measurement and simulation tools, little research has been conducted for reconstruction techniques for high-dimensional tensor tomography. For x-ray attenuation modality, specimen database and fast CT projection simulation tools for both central processing unit (CPU) and graphical processing unit (GPU) architectures are readily available. At small scales, CT voxel models show faster execution times on CPU compared to equivalent GPU algorithms [11], [12] due to the necessary time required to transfer data to and from the GPU from the host CPU. However, as scene complexity and size increase, so does the
CPU execution time. GPUs can parallelize the ray tracing process to drastically reduce total execution time for realistic complex scenes by orders of magnitude compared to a CPU. Efficient deterministic simulation tools for XRD would greatly accelerate the reconstruction algorithm development for XDT and high-dimensional tensor tomography in general.

This work reports a fast simulation platform for high-dimensional XRD imaging by leveraging the open source API OptiX, a 3D ray-tracing engine designed for NVIDIA GPUs and other highly parallel architectures. Features such as ray generation, hits, misses, shading, and masking are customizable so that a scene may be properly set for the implementation of various ray tracing based simulations [11]. OptiX uses a mesh representation of the surfaces of CAD models instead of a 3D voxel representation. By transitioning from a voxel-based model with objects composed of $10^6$-$10^8$ voxel primitives to a surface model with $10^5$-$10^6$ triangle primitives, ray tracing complexity reduces significantly [13]. For simulations involving scattering processes, Monte Carlo based simulations are commonly used, though computationally expensive [14], [15]. Faster methods using hybrid models with deterministic scattering centers and mesh representation achieves speed improvement down to a few to tens of minutes per fan beam projection [12]. In contrast, the mesh-only algorithms discussed here produce fan beam projections on the order of milliseconds to a few seconds. Differences in computational efficiency from hybrid models to this mesh-only model stem from the evenly spaced scattering centers requiring calculations for every scattering center [16], [17], rather than a single computation for each traced ray that uses the scatter profile defined by two hits used in the mesh-only approximation.
This document is structured as follows: first, the theory of XRD and the impetus for transforming the 3D ray tracing lab coordinate system into a 2D quasi-reciprocal space virtual coordinate system. Then, a discussion on the methods and flow of the OptiX simulation to define the expected intensity distribution of virtual projections corresponding to the detected diffracted signal. Next, the validity of the new method’s results is discussed with comparisons to voxel-based simulation and experimental sinograms. Finally, a comparison between the execution times of various configurations and demonstrate its feasibility for 3D XDT.
RELEVANT XDT THEORY

The setup for the XDT simulation considers narrow-band x-ray photons generated from a copper anode with a strong $K\alpha$ peak at 8.03 keV, corresponding to a vacuum wavelength of approximately 1.54 Å. Source intensity at other energies is significantly lower ($<30x$) than the main peak [8], and so is treated as negligible in this simulation. At higher energies, the source intensity is significantly lower, which is vital to possess due to an incrementally larger proportion of scattering being inelastic incoherent Compton scattering [18]. Thus, the dominant scattering mechanism under consideration is coherent scattering at low energies and small scattering angles [19]. By setting an upper limit on scattering angles by placing the detector sufficiently far away from the objects in the scene, the dominant process is further guaranteed to be coherent scattering at 8.03 keV. Because coherent scattering does not change the energy of the photons traversing through the material, the photon energy is treated as a constant. However, the direction of the photon changes with respect to the incident direction due to the scattering process, and is defined by Bragg’s law:

$$q = \frac{E}{\hbar c} \sin \left( \frac{\theta_s}{2} \right)$$

where $q$ is the momentum transfer associated with a photon of incident energy $E$ scattered at an angle $\theta_s$, and $\hbar$ and $c$ are Planck’s constant and the speed of light, respectively. The momentum transfer here is the magnitude of the 3D reciprocal-space vector, typically denoted by $\mathbf{G} = (a_1, a_2, a_3)$. Using crystalline samples, the reciprocal-space lattice points are sparse due to the periodicity and highly structured physical lattice, giving rise to the observation of discrete Laue
Figure 1: (a) Experimental setup of circular object with radius \( r \) positioned a distance \( L \) from a detector array located at plane \( y = L \). (b) The same 2D object shown in (a) but represented as a virtual 3D object by extruding the object’s physical 2D cross-section into a virtual \( z \)-direction that is proportional to \( q \) for each specified \( w \). © 2019 IEEE.

spots that is the crystal’s Fourier Transform pair. However, using amorphous samples as considered in this paper; diffraction occurs uniformly due to the randomized orientation of atoms/grains in the material lacking any global order for a beam traversing through the sample. Smooth rings of intensity are observed in a radial direction away from the central beam’s location on detector, hence the magnitude of the momentum transfer is equal for all points on detector of equal radius from the central beam. Thus, the magnitude of the momentum transfer in Equation 1 can be approximated in the small-angle regime by approximating \( \sin \left( \frac{\theta_s}{2} \right) \) with \( \tan \left( \frac{\theta_s}{2} \right) \) [10], [20] and applying the binomial approximation:

\[
q \approx \frac{E}{\hbar c} \frac{w}{2(L - y)} \approx \frac{E}{L \hbar c} \frac{w}{2} \left( 1 + \frac{y}{L} \right) \tag{2}
\]

The momentum transfer is thus expressed in terms of the radial distance, \( w \), between the primary beam’s location and the scattered ray’s location on detector, and the distance, \( L - y \), from where diffraction takes place inside the material to the detector plane, where \( y \) is the distance from the \( x \)-axis centered on the material to the instance of diffraction, and \( L \) is the distance from object
center to detector, as seen in Figure 1(a). The factor \( \frac{w}{z} \left( 1 + \frac{y}{L} \right) \) in Equation 2 represents the height of the ray, seen in Figure 1(b), at the location of a hit point on the virtual object and is denoted \( z_v \), for the virtual \( z \)-coordinate. The ring in the detector plane, seen in Figure 1(a), portrays the ideally radially isotropic intensity observed when illuminating an amorphous or powdered material. In such applications, the intensity \( I(x_d, L, z_d) \) is composed of all light scattered towards pixel \((x_d, L, z_d)\) from locations \( y \) within the material in the \( z = 0 \) plane, where the \( d \) subscript indicates real detector coordinates. It is assumed that \( I(x_d, L, z_d) = I(x, L, w) \) for all \( x_d \) and \( z_d \) satisfying \( |x_d - x|^2 + z_d^2 = w^2 \). Experimentally, the intensity at each radius is obtained by azimuthally binning data on detector to calculate an average value. In simulation, this average value, \( I(x, L, w) \), is directly calculated and used for comparison with experimentally binned results.

The \((x,y,z)\) and \((x,y,z_v)\) coordinate systems are fixed to the center of the object, the origin of the coordinate systems, and are independent of object rotation, \( \phi \). Hence, all object properties are subscripted with \( \phi \). Equation 2 shows that \( q \) is approximately linearly proportional to the depth of the scatter location, \( y \), and thus straight virtual rays may be used. Hits on the virtual object always correspond to different virtual \( z_v \) coordinates due to the non-zero slope of rays that are distinct from the primary beam. Pairs of hits on a virtual object thus define lower and upper bounds of momentum transfer that contribute to the overall intensity measured at \((x, L, w)\).

By changing \( w \), the slope of the virtual ray also changes and the corresponding momentum transfers are also modified. Figure 2 demonstrates this concept and the validity of the linear approximation and the regime for which it is applicable, namely at or below a height of 50 mm on the detector for a setup with the 8.03 keV source and the object placed 150 mm from the
Figure 2: Validity of the linear approximation of $q$ for small objects, and low radial distances $w$ on detector. © 2019 IEEE.

detector. The straight dotted lines are the approximated $q(y, w)$ while the solid lines are from Bragg’s Law. This critical observation enables the use of OptiX as discussed in the next section, allowing for straight virtual rays in the $y$-$z_v$ domain.

In previously considered voxel-based models [10] the number of photons coherently scattered, $dl$, from a voxel $dV$ into a solid angle $d\Omega$ at an angle is $\theta_s$ is given by:

$$dl = I_0 dV \, n_{0,\phi}(x, y) \frac{d\sigma}{d\Omega}$$

(3)

where $I_0$ is the incident number of photons per cm$^2$, $n_{0,\phi}$ is the number of scatterers per cm$^3$ for an object rotated an angle $\phi$ about the $z$-axis and $d\sigma/d\Omega$ is the differential cross-section of the elastic x-ray scattering in cm$^2$. 
\[
\frac{d\sigma}{d\Omega} = \frac{r_e^2}{2} \left( 1 + \cos^2 \left( \frac{\theta_s}{2} \right) \right) F^2_\phi(x, y, q) \tag{4}
\]

where \( r_e \) is the classical electron radius, \( F^2_\phi(x, y, q) \) is the unitless spatially varying form factor \( F^2(q) \) profile that depends on the material within the scene, and \( d\Omega = \Delta^2/(w^2+(L-y)^2) \) with \( \Delta \) as the detector pixel size.

At small angles and with small objects compared to the object-detector size, \( L \), an approximation to \( dl \) may be written as:

\[
dl = \frac{I_0 A \Delta^2 r_e^2}{L^2} F^2_\phi(x, y, q) n_{0,\phi}(x, y) \tag{5}
\]

where \( dV = Ady \), and \( A \) is the area of the beam. One may now define:

\[
f_\phi(x, y, q) = r_e^2 n_{0,\phi}(x, y) F^2_\phi(x, y, q) \tag{6}
\]

in the units of cm\(^{-1}\) to be the coherent scatter profile [20]. To demonstrate the spatial dependence of this profile, one may write \( f_\phi(x, y, q) \) for any arbitrary source at translation \( x \) and projection angle \( \phi \) as:

\[
f_{\phi,x}(y, q) = \sum_{n=1}^{N_{\phi,x}} \frac{1}{\Delta y_n} rect \left( \frac{y - y_n}{\Delta y_n} \right) f_{\phi,n}(q) \tag{7}
\]

where \( n \) is the index corresponding to the object number within the scene of \( N_{\phi,x} \) total hit objects ordered by increasing distance from source to detector, \( \Delta y_n \) and \( y_n \) are the \( y \) spatial extent and central offset of the \( n^{th} \) object in the scene at the translation \( x \) and rotation angle \( \phi \) respectively, and \( f_{\phi,n}(q) = r_e^2 n_{0,\phi,n} F^2_{\phi,n}(q) \) is the \( n^{th} \) object’s coherent scatter profile. Combining Equation 5 through Equation 7 one obtains the expected total number of coherently scattered photons from the pencil beam source:
Equation 8 is the physical basis on which OptiX computes XDT projections. However, compared to conventional methods of simulating scattering effects through objects that integrate through voxels in space \((y)\) [12], [13] this method performs calculations exclusively with integrations in the \(q\)-domain. In computing the integral, the scattering x-rays must first be re-formulated as traversing through \(y-q\) space by way of transforming hits from the \(y-z_v\) space. This motivates the use of OptiX and its parallelized architecture for fast ray tracing compared to a voxel-based method.
SIMULATION SETUP, METHODS, AND CONFIGURATION

OptiX is tailored to function as an XDT simulator by utilizing ray and hit payload data to determine the output intensity recorded on a detector array through each object (of specified material via a coherent scatter profile assigned to the object) in the scene. As with conventional CT scans, the scene is rotated with OptiX and projections are measured. With postprocessing, sinograms and reconstructions of the scene through a range of momentum transfer may be obtained.

OptiX traces rays through 3D objects and uses the intersection data such as the length of the ray from the ray origin to the hit location, the normal to the hit surface, the \((x,y,z_v)\) coordinate of the hit, and the ID of a hit triangle (primitive) \([11] [21]\), all of which are used to determine the hit material and momentum transfer associated with each scattering event.

In this section the object undergoing x-ray probing is a 2D phantom, considered to be a single slice of a 3D object, shown in Figure 3(b) through some plane at an offset \(z\). The spatial extent of the slice can theoretically be any 2D shape with any complex inner structure, as is oftentimes the case with real samples.

To create the virtual 3D \(q\)-space objects for use with OptiX, the 2D physical cross-section of the scene is extruded upwards in the virtual \(z\)-direction, \(z_v\) (a physical dimension with units of length), where \(z_v\) is proportional to the momentum transfer, \(q\). In general, this effectively creates an irregular right prism for each material. The height of each virtual object, \(z_v\), of a specified material is only limited by the available data of the form factor (i.e. an object that is taller than the corresponding largest momentum transfer data point of the form factor will yield no more additional added intensity for XDT projections because the \(z_v\)-\(q\) mapping is undefined). To
correctly set the height of the \( q \)-space object, one must look at the lab setup to determine the largest diffraction angle from the scene that the detector is able to detect. For example, if the center of a 2D object of radius \( r \) is placed a distance \( L \) from a detector array of height \( z = z_v = d \), it will detect a maximum diffraction angle set by these parameters. In virtual space we know that this maximum diffraction angle corresponds to a required maximum virtual object height, \( z_v = h \), that is proportional to the detector height, \( d \): \( h = \frac{L + r}{2L} d \).

In the small angle regime this reduces to half the detector height. Once the virtual object dimensions are properly set, ray tracing in the \((x, y, z_v)\) space may be performed.

An entire scene of varying geometries, number of primitives, and materials may be created by loading multiple models into an OptiX instance vector with proper model IDs. After loading the models into the scene, material properties are assigned to each model via the model ID. After rays are launched through the scene, the material of the hit will be identified via the hit data structure that includes model ID and the corresponding form factor that will be used for coherent scatter calculations.
Experimental setups of pencil beams are analogous to virtual vertical fan beams in OptiX and are easily created with ray generation programs on GPU(s) or CPU. Each ray in virtual space is directed to a unique detector pixel. Translations of the experimental pencil beam correspond to translations of the virtual vertical fan beam. Because the virtual rays have a non-zero $z_v$ direction, they are traced upwards through a range of momentum transfer. Thus, a single virtual ray represents a set of diffracted rays in experiment, and hence the intensity recorded from the single virtual ray at each pixel of height $w$ and translation $x$ is identical to the intensity accumulated from real diffracted rays that are directed towards pixels at a radius of $w$ from the primary beam’s location. Smaller pixel pitches with the same detector footprint will of course yield tighter probing of the virtual object due to a larger number of rays and thus higher resolution in $q$-space at the cost of extra computational time.

The data of one pixel at a height $w$ obtained from one virtual projection at an angle $\phi$ is given by Equation 8. Combining the results of Equation 1 with Equation 7 to change the variable of integration from $y$ to $q$, one finds the same total intensity, $I_{tot}$, received by the pixel to be comprised of the summation of $N_{\phi,x}$ partial intensities from $N_{\phi,x}$ objects in the trajectory of the single ray through the scene due to crossing $2N_{\phi,x}$ surfaces when traversing towards the detector:

$$I_{\phi,x,tot}(w) \propto \sum_{n=1}^{N_{\phi,x}} \int_{q_{j-1}}^{q_j} f_n(q) \, dq$$

Equation 9 is the OptiX equivalent to Equation 8. Here $w$ is the height of a pixel on the virtual detector array. The summation is indexed by the object number, $n$, ordered by increasing source to object distance. The surface number in the scene is $j = 2n$. The $n^{th}$ object’s coherent scatter
profile is \( f_n(q) \), dependent on the rotation angle \( \phi \) and spatial location \((x, y)\), determined by the model ID via hit data.

In order to properly obtain the correct amount of intensity for each pixel, \( p_{x,w} \), the flowchart seen in Figure 4 is followed on the detector in virtual space. First, rays are launched into the scene and the corresponding momentum transfer associated with each hit’s \( z_v \)-coordinate at the first surface crossing is determined, along with triangle IDs and model IDs associated with all objects of the scene. Once the momentum transfer has been recorded, the primitives (triangles) that have been hit are masked so that a second set of identical rays launched into the scene may hit second surfaces within the scene. Again, because the virtual rays are defined with a non-zero angle with the \( x-y \) plane, each ray’s subsequent hit \( z_v \)-coordinate at surfaces deeper into the scene must be larger, and hence a larger momentum transfer is calculated. That is to say, for each surface \( j \), \( q_{j-1} < q_j \). Once the momentum transfers of the second surfaces are calculated, the first pair of surfaces defines the first object(s) hit in the ray tracing process and is used to calculate the partial intensity associated with traversing through the object(s) from surface one to surface two towards pixel \( p_{x,w} \) on the detector. The range of momentum transfer for each object is used to define the limits of integration of the form factor profile, to yield an expected transmitted intensity. This model uses the assumption that the distance between an odd numbered hit and an even numbered hit is composed of only one material, and that an even numbered hit to an odd numbered hit is composed of only air so no scattering takes place. It is further verified that if a hit and a subsequent hit have different model IDs then there must not be any added intensity of the ray because the ray must have left an object and traversed in air to a new, second object.
The process of masking surfaces, launching identical rays, pairing surfaces, and integrating over material’s coherent scattering profile is repeated until there are no longer any hits from any of the rays, and the projection at the angle $\phi$ is finished. It is important to note that experimental pencil beams are used so that during a measurement all rays diffracted onto a pixel must originate from somewhere within the material along the primary beam’s path (same source translation, $x$), and not from an unknown location within the material (different translation).
RESULTS

The performance of this GPU accelerated scheme is evaluated according to the comparison between sinograms and reconstruction results generated by voxel models and experimental measurements. Execution times for each sub-routine are recorded and compared under various simulation conditions such as number of triangle primitives and rays launched through the scene.

Accuracy Comparison to Voxel Model

The experimental XDT setup for comparison uses a quasi-monochromatic filtered copper-anode x-ray tube (XRT60, Proto Manufacturing) source operating at 45 kV and 40mA, emitting an 8.03 keV peak. Lead pinholes are employed to collimate the beam to a diameter of 2 mm. The scene is placed 120 mm in front of the flat panel detector (1215CF-MP, Rayence) on a rotational stage (RV1200P, Newport) and linear translational stage (UTM120CC, Newport). 33 translations of the sample with a 1 mm step and 46 projections with a 4° step were used to gather the experimental pencil beam data taken over ~19 hours [22]. The Teflon ring has an outer radius of 14.5 mm, and an inner radius of 13.5 mm, with methanol, oil, and water situated inside with radii of 6 mm, 3.3 mm, and 4.1 mm respectively. The central transmitted direct beam is blocked with a 10 mm x 10 mm lead beam stop, and results are binned with concentric rings of the 2D XRD images to obtain a 1D intensity profile along the radial direction. Stitched together with the 33 translations, this allows for comparison to the OptiX virtual projections of the wedge-shaped parallel vertical fan beams. The beam stop effectively creates the artificial lower bound of \( q \approx 0.02 \text{ Å}^{-1} \) seen in Figure 3(a) and Figure 8(b).
Using the same host machine running Ubuntu 16.04.3 LTS with 4 hyper-threaded Intel(R) i7-7700K cores at 4.20 GHz and 64 GB RAM, both voxel and OptiX simulations are run. For GPU accelerated simulations an NVIDIA GeForce GTX 1080 Ti GPU with 11 GB memory is used. Collecting the projection data for the 46 projection angles $\phi$ from 0° to 180° with a 4° step size, sinograms for all detector height, $z_v$, are obtained.

OptiX and voxel sinograms agree well with each other and with experimental results as shown in Figure 5. The average NMSE between the voxel results and OptiX results is approximately 0.66%. NMSE at 6 mm, 22 mm, and 38 mm is 1.15%, 0.70%, and 0.74% respectively. This suggests the assumptions made at large diffraction angles are valid.

Figure 6 shows line slices through the stack of sinograms at $\phi = 0^\circ$ at four different translations and are effectively 1D projections of the evenly spaced virtual vertical fan beams. The solid black curves show OptiX performance, while voxel-model is dashed, and experimental is blue. The simulated peak signals match with experimental binned results at all translations. Offsets in the signal seen at $x = 24$ mm is due to displacements in the scene from experiment to simulation, but overall matches well.

A modified Feldkamp, Davis, and Kress (FDK) reconstruction algorithm [23] to suit the parallel vertical fan beam source is performed on projection results for both OptiX and voxel models. Comparisons between the reconstructions are shown in Figure 7 at various momentum transfers and show obvious agreement for the entire range.
Figure 5: Sinograms comparing OptiX to voxel model and experimental data at various detector heights $z_v = (a) 6$ mm, (b) 22 mm, (c) 38 mm on detector. © 2019 IEEE

Figure 6: Line slices through the virtual projection at $\phi = 0^\circ$ for OptiX (solid black), Voxel (dashed), and experimental (blue) results at translations (a) (b) (c) (d). © 2019 IEEE
Execution Time

The execution timing analysis concerns the scene common to both the OptiX and voxel simulations shown in Figure 3(b). With OptiX, the scene of object files is used at two mesh resolutions, a high resolution (1,002,152 total primitives), and a low resolution (144,384 total primitives). Results from one resolution to the other show no significant visual or statistical differences but result in different execution times.

Six sets of simulations were run and averaged to demonstrate the typical computational time required to compute XDT results with OptiX on GPU, and is displayed in Table I. Similar voxel-based simulations were performed with 77x77x144, 154x154x288, and 308x308x576 voxels, with detector grid sizes of 77x144, 154x288, and 308x576 respectively, so that an equivalent number of rays are traced through the voxels and meshes. The average CPU execution times of the three voxel-based configurations are 0.65s, 8.83s, and 127.80s per projection, respectively. A direct comparison between the two simulation methods is not meaningful, considering the
Table 1: Execution Timing for Various Configurations of Simulating 180 XDT Projections

<table>
<thead>
<tr>
<th>Resolution (# Primitives)</th>
<th>Detector Size (# rays)</th>
<th>Avg (s)</th>
<th>Avg. Computational Time (ms)</th>
<th>Avg. GPU Ray Execution (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>142,712</td>
<td>77x144</td>
<td>0.215025</td>
<td>1.653</td>
<td>0.199</td>
</tr>
<tr>
<td>142,712</td>
<td>154x288</td>
<td>0.241171</td>
<td>2.919</td>
<td>0.280</td>
</tr>
<tr>
<td>142,712</td>
<td>308x576</td>
<td>0.332482</td>
<td>7.734</td>
<td>0.743</td>
</tr>
<tr>
<td>1,002,152</td>
<td>77x144</td>
<td>1.506853</td>
<td>4.458</td>
<td>0.260</td>
</tr>
<tr>
<td>1,002,152</td>
<td>154x288</td>
<td>1.549855</td>
<td>5.780</td>
<td>0.376</td>
</tr>
<tr>
<td>1,002,152</td>
<td>308x576</td>
<td>1.669055</td>
<td>10.740</td>
<td>1.067</td>
</tr>
</tbody>
</table>

Times are per projection, where each projection is from one phantom scene, calculated using 1 NVIDIA GPU. All times include the time to save the data. © 2019 IEEE.

Architectural difference between the CPU and GPU. However, execution times for previous reported voxel-based simulations on GPU show 1.2 s execution time for 256x256x70 voxels, and 2.3 s for 512x512x100 voxels on a 512x384 detector grid [15].

Column 3 of Table I shows the execution time of a single projection by taking the average of 10 repeated simulations of 180 projections each to determine an accurate time expected for such a detector size and object resolution. The average computational time for each projection is on the order of milliseconds. It can be observed that object resolution influences execution time more so than the number of rays shot through the scene. For tomography projections, the majority of the time is spent between projection angles reloading objects (~80-95% of total time). Other necessary processes such as rotating to a new projection angle (~0.005% of total time), resetting masks (~2-4% of total time), and other preparatory code before the ray tracing begins (~3-16%) also contribute to the computational time overhead. Using adaptive mesh generated objects may reduce the number of required primitives to effectively sample fine features within the scene, and thus significantly reduce execution times.
The average execution time in Column 4 is composed of the time it takes to calculate the diffracted signal intensity with processes such as calculating the momentum transfer of each hit, pairing consecutive hits for the integration bounds, computing the correct object(s) coherent profile integration(s) for each pixel, and the idle CPU time that is necessary to synchronize GPU with CPU to avoid race conditions and segmentation faults. Column 5 is the total GPU time taken to execute all launches of rays for each projection angle, where the number of launches is determined by the number of hit surfaces. The total time taken by GPU is roughly 10 times less than the computational time taken by intensity calculations, and approximately 0.1% of the total execution time of the simulation. Thus, it is almost negligible compared to all other factors, and varies for each configuration because the number of hits is always different, either due to the object resolution differing, and/or the number of rays launched into the scene.

Simulation of Objects with Three Spatial Dimensions
Ray tracing with arbitrarily complex objects is not a challenge for OptiX because of the ability to directly load object files created with CAD software like SolidWorks, MeshLab, or Blender, whereas the voxel methods require large data cubes with necessarily lower resolution for the sake of memory and computational time. Because of the speed at which OptiX can perform ray-tracing, reconstructions of entire 3D objects are made possible for any momentum transfer in the range specified by the objects within the scene, creating an enormous amount of data with 4D results.

The technique is applied to 64 horizontal cross-sections from 0.0 mm to 6.25 mm with a 0.097 mm step through the scene composed of two Teflon bolts and an acrylic nut shown in Figure 8(a). Tracing 154x288 rays for 180 virtual projections with a 1° step through each of the
individual 64 extruded cross-sections, sinograms for each of the 288 virtual detector rows from 0 to 57.6 mm are obtained. Reconstructions at a single momentum transfer, $q = 0.043 \, \text{Å}^{-1}$, for 3 cross-sections are shown in the top row of Figure 8. For each cross-section, 2 sinograms at virtual detector heights $z_v = 4$ mm and $z_v = 20$ mm are shown.

Each extruded slice contains roughly 375,000 primitives but varies from slice to slice. Projections take approximately 0.79s to calculate and hence fall between the low and high resolution scenes in Table I. The total simulation time is roughly 4.5 hours. Ideally the total time is ~2.5 hours for the entire 3D scene, but with a large number of slices and imperfect memory management the process slowed down by up to an additional 80%.
DISCUSSION

The results of the simulation are in good agreement with voxel-based simulations and experimental results for all attempted objects with various sizes, resolutions, and materials. Factors that contribute to the differences in results may be attributed to displacements between simulations and experimental scenes, coherent scatter profiles (due to an imperfect source with a spectrum of radiation), as well as object primitive hit confusion.

Sub-voxel displacement has been noted and attempted to be reproduced in both voxel and OptiX, which can be observed in the Teflon edges of the sinograms in Figure 5. The displacement leads to differences in the path length of rays traversing the scene, and ultimately differing recorded intensities between OptiX and the voxel results.

Discretizing the coherent scatter profile and performing a trapezoidal integration with interpolated nearest momentum transfer values may also lead to variations in the intensity results, which becomes more obvious when the molecular form factor has sharp peaks. In other words, one simulation may capture the peak while the other does not, simply due to a different sampling of $q$. Thus, mesh and voxel primitives are ideally as large as possible to have small data sets but small enough to adequately capture each material’s scatter signature.

Object hit confusion may be the result of sampling sharp edges in the geometry of objects in a scene (such as the threads of the nut in the 3D scene), and causing errors in the intensity calculations for those rays and ultimately leading to artifacts in reconstructions. In addition, rays may hit an odd number of surfaces due to the edges, which leads to an incorrect pairing of surfaces and ultimately nonsense results for the calculated diffracted signal. For the ~0.0135% of
Figure 8: 3D scene composed of Teflon bolts and an acrylic nut. © 2019 IEEE

rays that show these artifacts in some projections of the 3D scene, they are set to 0 added intensity, yielding streaks through the reconstructions.
CONCLUSION

In conclusion, the mesh-based ray-tracing approach for high-dimensional XDT simulations implemented with the OptiX API shows an acceleration between 3 to 300 times faster than similar voxel-based methods, which can be attributed to acceleration from GPU resources coupled with mesh objects. The simulation output is the measurement of diffraction signal $I(x,z,\phi)$ from a single cross-section of a scene. Three dimensional volumetric reconstructions have also been demonstrated for each momentum transfer. The advantage of this simulation is its ability to compute diffracted signals from significantly larger objects, on the order of 10s of millimeters as opposed to ~1 mm with voxel-based methods, with the assumption that each object within a scene is a homogeneous material with no variations of density or constituents within its volume.

By using adaptive meshes to reduce the number of triangle primitives of the 3D virtual objects but still effectively capture the fine details, one could expect shorter execution times with reduced artifacts. Simultaneous ray-tracing models including both attenuation and the scatter signal with multiple energy channels could also be implemented, which would allow more accurate simulation for large objects with length scale of ~ 1 meter, where attenuations for both incident and scatter beam need to be considered.
Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within the thesis), users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © [year].
2) In the case of illustrative or tabular material, we require that the copyright line © [year of original publication] appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication].
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis online.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any [university/educational entity's] name, products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/index.html to learn how to obtain a license from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK  

CLOSE
REFERENCES


In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University of Central Florida's products or services.


