This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNS.2019.2948796, IEEE Transactions on Nuclear Science

Accelerated x-ray diffraction (tensor) tomography simulation using OptiX GPU ray-tracing engine

Joseph Ulseth, Zheyuan Zhu, Yangyang Sun, and Shuo 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). Current challenges presented with XDT include long image acquisition and simulation time. To accelerate the simulation speed our approach is to adopt NVIDIA's OptiX ray-tracing engine, a parallelized pipeline for graphics processing units (GPUs), to perform XDT simulations on objects by making use of the innovative transformation from conventional 3D physical space into a 2D quasi-reciprocal space. The advantage being that ray tracing in this domain requires only 3D mesh objects, 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 the ray-tracing time is 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 large-scale high-dimensional tensor tomography simulations.**

***Index Terms*—x-ray diffraction, GPU-based ray tracing, coherent scattering, x-ray diffraction tomography, OptiX**

I. Introduction

X-RAY diffraction (XRD) imaging makes use of the structural information from the small-angle scatter signal, which can uniquely identify the presence of specific materials. 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. 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, voxel models for CT 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 are able to 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.

In this work, we report 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 found here.

The paper is structured as follows: we begin with 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, we discuss 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 are discussed with comparisons to voxel-based simulation and experimental sinograms. Finally, we compare the execution

Joseph Ulseth, Zheyuan Zhu, Yangyang Sun, Shuo Pang* are with The College of Optics and Photonics, CREOL, University of Central Florida, Orlando, 32816, FL, US, e-mail: julseth@knights.ucf.edu, e-mail: pang@creol.ucf.edu

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNS.2019.2948796, IEEE Transactions on Nuclear Science

times of various configurations and demonstrate its feasibility for 3D XDT. The proposed simulation method has demonstrated accelerated execution times by taking the advantage of GPU parallelization that lends itself to high-dimensional tomography (projection) simulations with large sets of output data.

II. X-Ray Diffraction Tomography (XDT) Theory

The setup for the XDT simulation considers narrow-band x-ray photons generated from a copper anode with a strong peak at 8.03 keV. The dominant scattering mechanism is coherent scattering at low energies and small scattering angles [18] but increasing contributions from incoherent scattering arise with larger energies [19]. We consider the dominant process to be coherent scattering at 8.03 keV chosen for simulation by setting an upper limit on the scattering angles by placing the detector sufficiently far away from the objects in the scene. 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 and approximated as follows:

$$q = \frac{E}{hc} sin\left(\frac{\theta_s}{2}\right) \approx \frac{E}{Lhc}\frac{w}{2}(1 + \frac{y}{L}) \tag{1}$$

where q is the momentum transfer associated with a photon of incident energy E scattered at an angle θ_s, and h and c are Planck's constant and the speed of light, respectively. Bragg's law is approximated on the right-hand side of Equation (1) in the small angle regime by approximating $\sin(\theta_s/2)$ with $\tan(\theta_s/2)$ [10], [20] and applying the binomial approximation. The momentum transfer is thus expressed in terms of the radial distance, w, between the primary beam's location and the scattered beam'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. The factor $\frac{w}{2}(1 + \frac{y}{L})$ in Equation (1) represents the height of the ray seen in Figure 1(b) 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 n amorphous or powdered material and not a crystalline 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. 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 and are independent of object rotation ϕ. Hence, all object properties are subscripted with ϕ. Equation (1) 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).

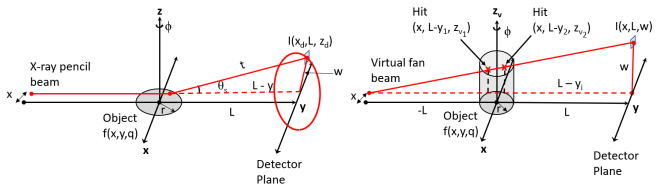

Fig. 1. (a) Experimental setup of an object of radius r placed a distance L from a detector array located at plane $y = L$. In the small angle regime, Bragg's law may be approximated as in Equation (1) by setting $t = L - y$. One incident x-ray is scattered a radial distance w from the primary beam's location to a position (x_d, L, z_d) on the detector. (b) The same 2D object 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. The incident x-ray pencil beam source translated along the x-direction is modeled as a translated fan beam, shown by the single virtual ray (solid red line). Intensity $I(x, L, w)$ is composed of all partial intensities associated with hits at locations (x, y_j, z_{v_j}) of this virtual ray with the virtual object that diffract to pixel (x, L, w) throughout a range of q.

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 detector. This critical observation enables the use of OptiX as discussed in the next section, allowing for straight virtual rays in the z_v-y domain.

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

$$dI = I_0 dV n_{0,\phi}(x,y)\frac{d\sigma}{d\Omega} \tag{2}$$

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 ϕ about the z-axis and

$\frac{d\sigma}{d\Omega}$ is the differential cross-section of the elastic x-ray scattering in cm^2,

$$\frac{d\sigma}{d\Omega} = \frac{r_e^2}{2}\left(1 + cos^2\left(\frac{\theta_s}{2}\right)\right)F_\phi^2(x,y,q) \tag{3}$$

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

At small angles and with small objects compared to the object-detector size, L, we may approximate dI as:

$$dI \approx \frac{I_0 A \Delta^2 r_e^2}{L^2} F_\phi^2(x,y,q) n_{0,\phi}(x,y) dy \tag{4}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNS.2019.2948796, IEEE Transactions on Nuclear Science

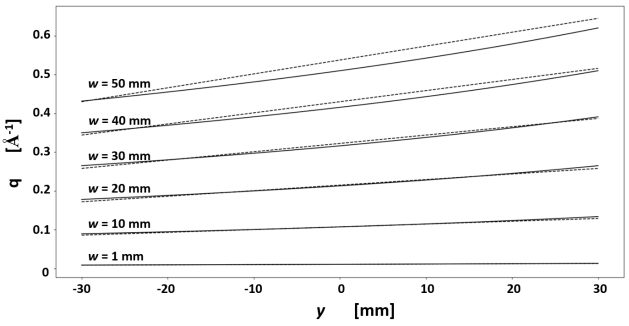

Fig. 2. Momentum transfer using the binomial approximated value (dashed lines) compared to the exact form of Bragg's law in Equation (1) (solid lines). Here E = 20 keV, L = 150.0 mm, y is a typical range for an object in the scene, and w is the distance from the primary beam's location. These curves demonstrate how well q is approximated at any distance y from the object's center, and radial distance w on detector. These virtual rays in the q-y domain may be approximated as straight through virtual 3D objects in an acceptable range of object sizes. Larger values of w lead to a range of slope through the material, and thus may not be suitable for the linear virtual ray approximation.

where $dV = Ady$, and A is the area of the beam. Here we now define:

$$f_\phi(x, y, q) = r_e^2 n_{0,\phi}(x, y) F_\phi^2(x, y, q) \tag{5}$$

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

$$f_{\phi,x}(y, q) = \sum_{n=1}^{N_{\phi,x}} \frac{1}{\Delta y_n} rect\left(\frac{y - y_n}{\Delta y_n}\right) f_{\phi,n}(q) \tag{6}$$

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, Δy_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 ϕ respectively, and $f_{\phi,n}(q) = r_e^2 n_{0,(\phi,n)} F_{\phi,n}^2(q)$ is the n^{th} object's coherent scatter profile. Combining Equation (4) through Equation (6) we obtain the expected total number of coherently scattered photons from the pencil beam source:

$$I_{\phi,x,tot}(w) \approx \frac{I_0 A \Delta^2}{L^2} \int f_{\phi,x}(y, q(y, w)) dy \tag{7}$$

Equation (7) 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 q-y space by way of transforming hits from the z_v -y space. This motivates the use of OptiX and its parallelized architecture for fast ray tracing compared to a voxel-based method.

III. Simulation Setup, Methods, and Configuration

We have tailored OptiX 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 post-processing, sinograms and reconstructions of the scene through a range of momentum transfer may be obtained.

OptiX enables us to trace rays through 3D objects and use the intersection data such as the length of the ray from 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.

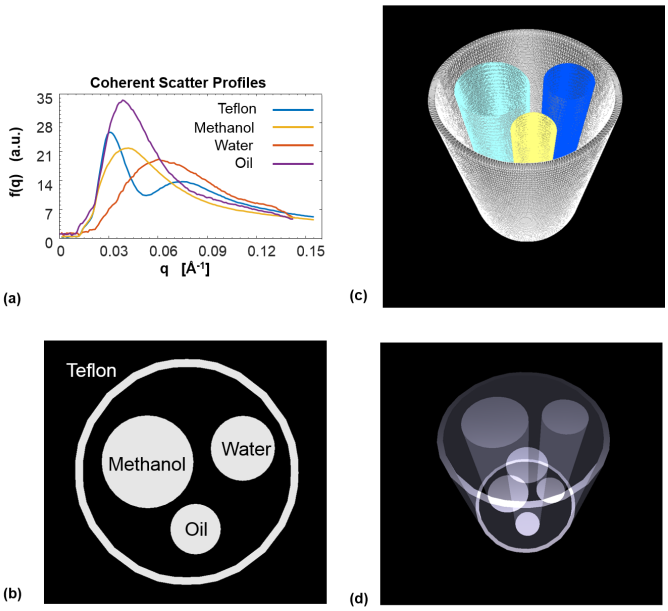

Fig. 3. Virtual scene phantom composed of 4 materials. (a) Coherent scatter profiles of the materials used. (b) Physical 2D slice of scene at 0°. (c) Mesh view of the four materials, displaying all triangle primitives. (d) 3D virtual view displaying all surfaces. Every virtual object is defined as an irregular right prism with respect to the virtual z-axis because the cross-sections of the virtual objects at different momentum transfers must agree on the real 2D shape of the object.

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 we extrude the 2D physical cross-section of the scene upwards in the virtual z-direction (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, we 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 of 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 ϕ is given by Equation (7). 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:

$$I_{\phi,x,tot}(w) \propto \sum_{n=1}^{N_{\phi,x}} \int_{q_{j-1}}^{q_j} f_n(q)dq \tag{8}$$

Equation (8) is the OptiX equivalent to Equation (7). 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 ϕ 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, $p_{x,w}$, on the detector in virtual space, we first launch rays into the scene and determine the corresponding momentum transfer associated with each hit's z_v-coordinate at the first surface crossing, 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 ϕ 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), and not from an unknown location within the material (different translation).

IV. Results

To validate our approach to perform XDT ray-tracing using OptiX, we include sinogram and reconstruction comparisons to voxel models and experimental measurements. The execution times for each sub-routine are recorded and compared under various simulation conditions.

A. 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. We place the scene 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,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNS.2019.2948796, IEEE Transactions on Nuclear Science

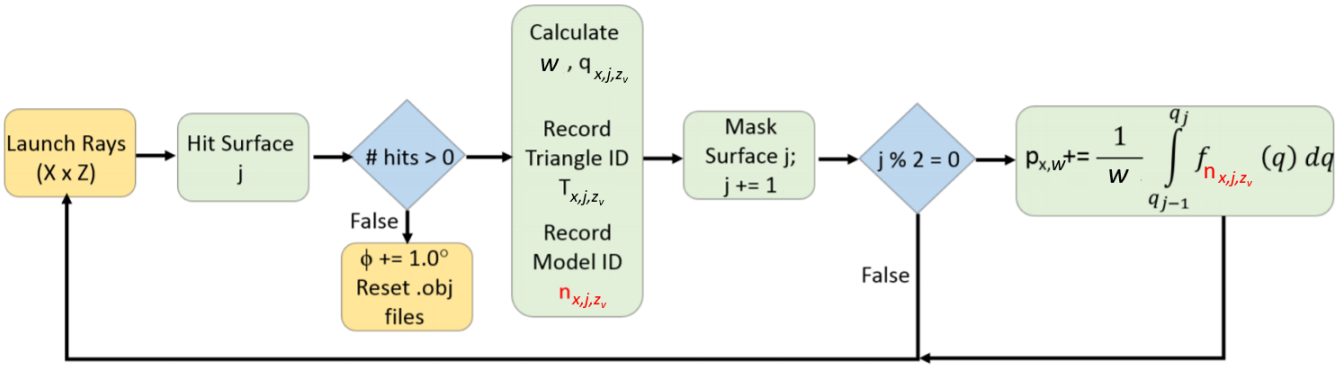

Fig. 4. Flow diagram of the basic ray-tracing technique implemented with OptiX. Rays in q-y space are directly sent to pixels $p_{x,w}$ and hit objects along the way at increasing momentum transfer values as they traverse closer to the detector. Each hit contains vital data such as the ID of the primitive that is hit, along with the model (object/material) ID. Because primitive IDs are degenerate (a simple enumerated list per object), the model ID is necessary to uniquely identify the primitive within the entire scene. After each hit primitive is identified, it is masked so that a new set of identical rays may penetrate deeper into the scene. Hit data from hits $h_{x,j-1,z_{v_1}}$ and $h_{x,j,z_{v_2}}$ is paired to calculate the partial added intensity from object n traversed by the ray $r_{x,w}$. Partial intensities are calculated until ray $r_{x,w}$ no longer hits any object within the scene.

oil, and water situated inside with radii of 6 mm, 3.3 mm, and 4.1 mm respectively. We block the central transmitted direct beam with a 10 mm x 10 mm lead beam stop, and bin results 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 $q \approx 0.02$ 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, we run both voxel and OptiX simulations. For GPU accelerated simulations we use an NVIDIA GeForce GTX 1080 Ti GPU with 11 GB memory. Collecting the projection data for the 46 projection angles ϕ from 0° to 180° with a 4° step size, we obtain sinograms for all detector height, z_v.

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 ϕ = 0° at four different translations and are effectively 1D projections of the evenly spaced virtual vertical fan beams. 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 FDK reconstruction algorithm [23] to suit our 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.

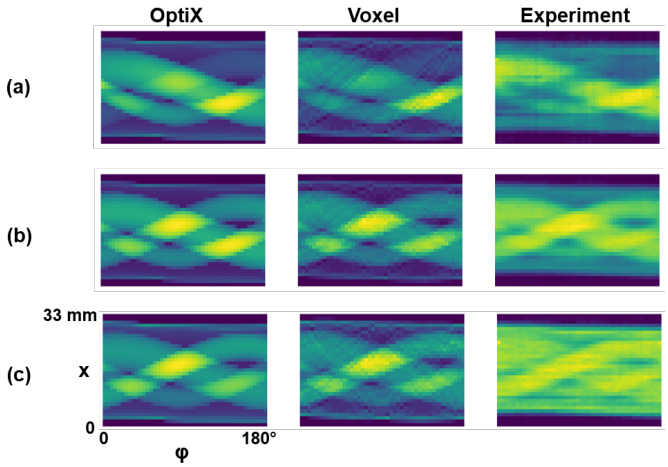

Fig. 5. Sinograms comparing OptiX to voxel model and experimental data at various detector heights of z_v = (a) 6 mm, (b) 22 mm, and (c) 38 mm on detector.

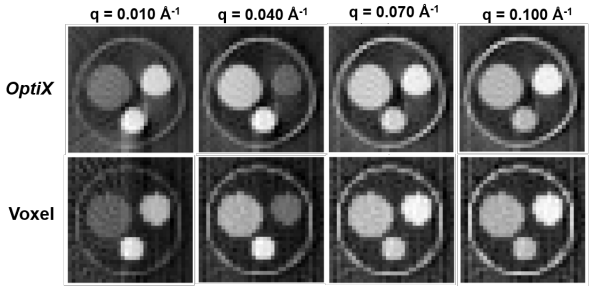

Fig. 6. Line slices through the virtual projection at ϕ = 0° for OptiX (solid black curve), voxel model (dashed), and experimental (blue) at translations (a) x = 3 mm (edge of the Teflon ring) (b) x = 10 mm (25% through scene, Teflon and methanol) (c) x =17 mm (50% through scene, Teflon and oil) and (d) x = 24 mm (75% through scene, Teflon and water).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNS.2019.2948796, IEEE Transactions on Nuclear Science

B. Execution Time

The execution timing analysis concerns the scene common to both the OptiX and voxel simulations shown in Figure 3. 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 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.

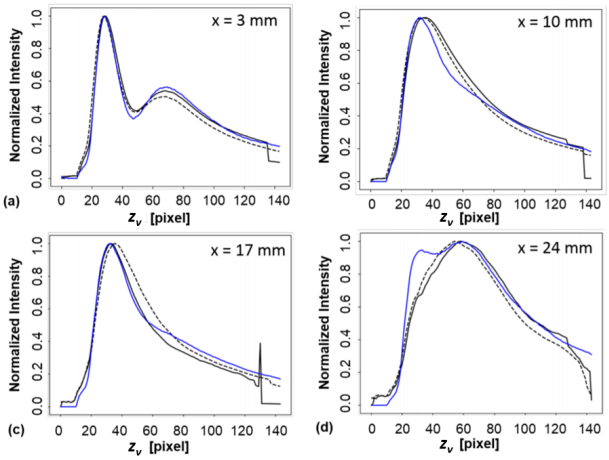

Fig. 7. Scene reconstructions of OptiX compared to Voxel model, at evenly spaced momentum transfers.

C. 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.

We apply the technique 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, we obtain sinograms for each of the 288 virtual detector rows from 0 to 57.6 mm. Reconstructions at a single momentum transfer, q = 0.043 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%. Visualizations of the 3D reconstructions at each momentum transfer are included in the Supplementary Materials.

V. 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.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNS.2019.2948796, IEEE Transactions on Nuclear Science

TABLE I
EXECUTION TIMING FOR VARIOUS CONFIGURATIONS OF SIMULATING AND SAVING 180 XDT PROJECTIONS FOR ONE PHANTOM SCENE, USING 1 NVIDIA GPU.

Resolution (# Primitives)	Detector Size (number of rays)	Average Execution Time per Projection (s)	Average Computational Time per Projection (ms)	Average GPU Ray Execution per Projection (ms)
142,712	77x144	0.215025	1.653	0.199
142,712	154x288	0.241171	2.919	0.280
142,712	308x576	0.332482	7.734	0.743
1,002,152	77x144	1.506853	4.458	0.260
1,002,152	154x288	1.549855	5.780	0.376
1,002,152	308x576	1.669055	10.740	1.067

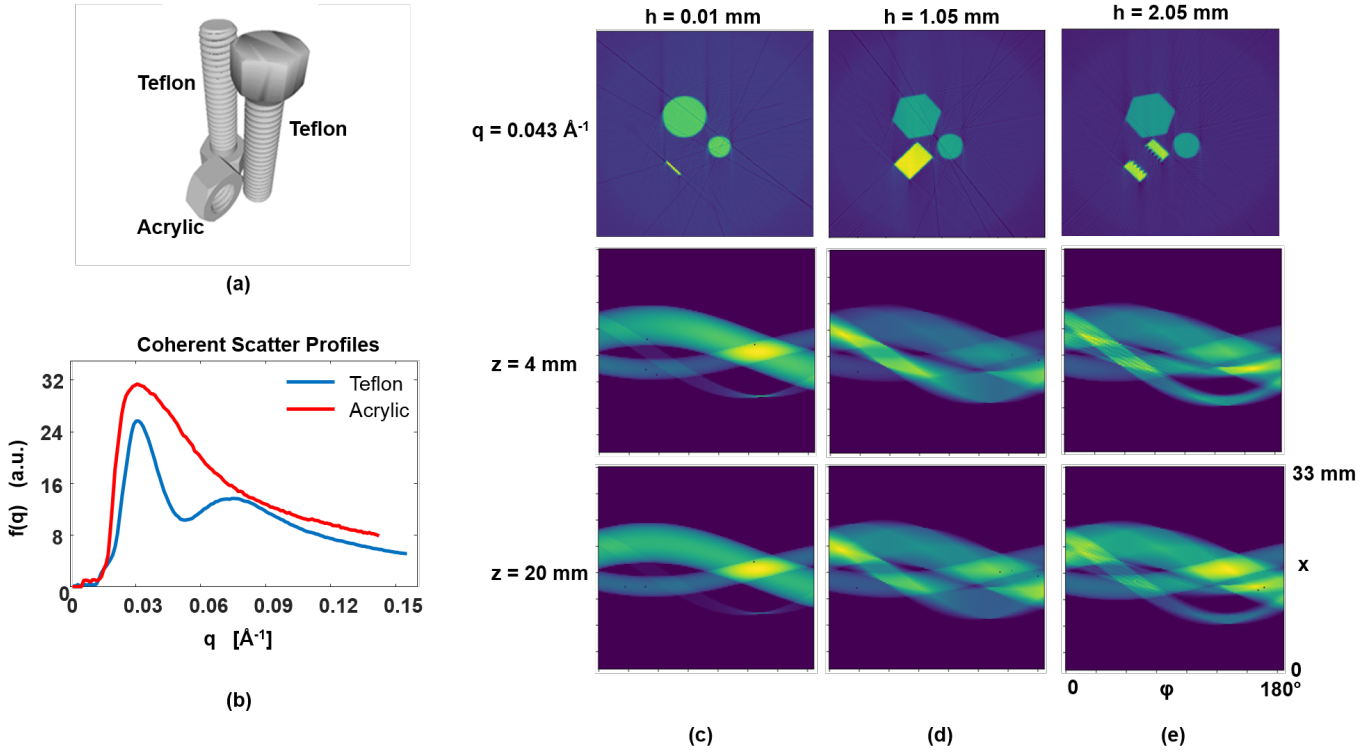

Fig. 8. (a) 3D scene composed of Teflon bolts and an acrylic nut. (b) Coherent scatter profiles of materials within scene. (c) Reconstruction of a horizontal slice at h = 0.01 mm through the 3D scene at q = 0.043 Å^{-1} and sinograms of the slice from detector data from a height z_v = 4 mm and z_v = 20 mm. (d), (e) same as part (c) but for horizontal slices through the scene at h = 1.05 mm and h = 2.05 mm, respectively.

Factors that contribute to the differences in results may be attributed 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 rays that show these artifacts in some projections of the 3D scene, we set them to 0 added intensity and see streaks through the reconstructions.

VI. CONCLUSION

In conclusion, we report a mesh-based ray-tracing approach for high-dimensional XDT simulations implemented with the OptiX API that 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 our 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.

ACKNOWLEDGMENT

The authors would like to thank Mr. Timothy Harvey of EMF Industries for helpful discussions.

REFERENCES

[1] S. Sidhu, K. K. Siu, G. Falzon, S. A. Hart, J. G. Foxe, and R. A. Lewis, "Mapping structural changes in breast tissue disease using x-ray scattering," Med. Phys, vol. 36, pp. 3211-3217, 2009.

[2] C. Cui, S. M. Jorgensen, D. R. Eaker, and E. L. Ritman, "Direct three-dimensional coherently scattered x-ray microtomography," Med. Phys, vol. 37, pp. 6317-6322, 2010.

[3] R. W. Madden, J. Mahdavieh, R. C. Smith, and R. Subramanian, "An explosives detection system for airline security using coherent x-ray scattering technology," in Proc. SPIE 7079, Hard X-Ray, Gamma-Ray, and Neutron Detector Physics X, vol. 7079, 2008, pp. 707911-707915.

[4] Y. D. Dong and B. J. Boyd, "Applications of X-ray scattering in pharmaceutical science," Int. J. Pharm, vol. 417, pp. 101-111, 2011

[5] G. Harding, "Energy-dispersive x-ray diffraction tomography," Phys. Med. Biol, vol. 35, 1990.

[6] Y. Sharma, M. Wieczorek, F. Schaff, S. Seyyedi, F. Prade, F. Pfeiffer, and T. Lasser, "Six dimensional X-ray Tensor Tomography with a compact laboratory setup," Appl. Phys. Lett, vol. 109, p. 134102, 2016.

[7] A. Castoldi, C. Ozkan, C. Guazzoni, A. Bjeoumikhov, and R. Hartmann, "Experimental qualification of a novel X-ray diffraction imaging setup based on polycapillary X-ray optics," IEEE Trans. Nucl. Sci, vol. 57, pp. 2564-2570, 2010.

[8] Z. Zhu and S. Pang, "Three-dimensional reciprocal space x-ray coherent scattering tomography of two-dimensional object," Med. Phys, vol. 45, pp. 1654-1661, 2018.

[9] Z. Zhu, R. A. Ellis, and S. Pang, "Coded cone-beam x-ray diffraction tomography with a low-brilliance tabletop source," Optica, vol. 5, p. 733, 2018.

[10] Z. Zhu, A. Katsevich, and S. Pang, "Interior x-ray diffraction tomography with low-resolution exterior information," Phys. Med. Biol, vol. 64, no. 2, 2019.

[11] F. Mauch, M. Gronle, W. Lyda, and W. Osten, "Open-source graphics processing unit-accelerated ray tracer for optical simulation," Opt. Eng, vol. 52, no. 5, p. 053004, 2013.

[12] Q. Gong, J. A. Greenberg, R.-I. Stoian, D. Coccarelli, E. Vera, and M. E. Gehm, "Rapid simulation of X-ray scatter measurements for threat detection via GPU-based ray-tracing," Nucl. Instruments Methods Phys. Res. Sect. B Beam Interact. with Mater. Atoms, vol. 449, no. March, pp. 86-93, 2019.

[13] Q. Gong, R. I. Stoian, D. S. Coccarelli, J. A. Greenberg, E. Vera, and M. E. Gehm, "Rapid simulation of X-ray transmission imaging for baggage inspection via GPU-based ray-tracing," Nucl. Instruments Methods Phys. Res. Sect. B Beam Interact. with Mater. Atoms, vol. 415, pp. 100-109, 2018.

[14] A. Badal and A. Badano, "Accelerating monte carlo simulations of photon transport in a voxelized geometry using a massively parallel graphics processing unit," Med. Phys. , vol. 36, no. 11, pp. 4878-4880, 2009.

[15] X. Jia, S. B. Jiang et al., "A GPU tool for efficient, accurate, and realistic simulation of cone beam CT projections," American Association of Physicists in Medicine, vol. 39, no. 12, pp. 7368-7378, 2012. [Online]. Available: 10.1118/1.4766436

[16] N. Freud., P. Duvauchelle, S. Pistrui-Maximean, J. M. Letang, D. Babot, and Nucl. Instrum. Methods Phys. Res. Sect. B, "Deterministic simulation of first-order scattering in virtual x-ray imaging," vol. 222, no. 1, pp. 285-300, 2004.

[17] P. Duvauchelle, N. Freud, V. Kaftandjian, D. Babot, and Nucl. Instr. Meth. Phys. Res. Sect. B, "A computer code to simulate x- ray imaging techniques," vol. 170, no. 1, pp. 245-258, 2000.

[18] P. Johns and M. J. Yaffe, "Coherent scatter in diagnostic radiology," Med. Phys., vol. 10, no. 1, pp. 40-50, 1983. [Online]. Available: 10.1118/1.595443

[19] P. Villanueva-Perez, S. Bajt, and H. N. Chapman, "Dose efficient Compton X-ray microscopy," Optica, vol. 5, no. 4, pp. 450-457, 2018.

[20] J. Kosanetzky, B. Knoerr, G. Harding, and U. Neitzel, "X-ray diffraction measurements of some plastic materials and body tissues," Am. Assoc. Phys. Med., vol. 14, no. 4, pp. 526-532, 1987.

[21] S. G. Parker, "OptiX," ACM Trans. Graph, vol. 29, no. 4, pp. 1-13, 2010.

[22] Z. Zhu, A. Katsevich, A. J. Kapadia, J. A. Greenberg, and S. Pang, "Xray diffraction tomography with limited projection information," Sci. Rep, vol. 8, no. 1, pp. 4-9, 2018.

[23] L. A. Feldkamp, L. C. Davis, and J. W. Kress, "Practical Cone-Beam Algorithm," J. Opt. Soc. Am. A-Optics Image Sci. Vis, vol. 1, no. 6, pp. 612-619, 1984.