

# A GPU tool for efficient, accurate, and realistic simulation of cone beam CT projections

Xun Jia,<sup>a)</sup> Hao Yan, and Laura Cerviño

*Center for Advanced Radiotherapy Technologies and Department of Radiation Medicine and Applied Sciences, University of California, San Diego, La Jolla, California 92037*

Michael Folkerts

*Center for Advanced Radiotherapy Technologies and Department of Radiation Medicine and Applied Sciences, University of California, San Diego, La Jolla, California 92037 and Department of Physics, University of California, San Diego, La Jolla, California 92093*

Steve B. Jiang<sup>a)</sup>

*Center for Advanced Radiotherapy Technologies and Department of Radiation Medicine and Applied Sciences, University of California, San Diego, La Jolla, California 92037*

(Received 28 April 2012; revised 23 October 2012; accepted for publication 23 October 2012; published 27 November 2012)

**Purpose:** Simulation of x-ray projection images plays an important role in cone beam CT (CBCT) related research projects, such as the design of reconstruction algorithms or scanners. A projection image contains primary signal, scatter signal, and noise. It is computationally demanding to perform accurate and realistic computations for all of these components. In this work, the authors develop a package on graphics processing unit (GPU), called gDRR, for the accurate and efficient computations of x-ray projection images in CBCT under clinically realistic conditions.

**Methods:** The primary signal is computed by a trilinear ray-tracing algorithm. A Monte Carlo (MC) simulation is then performed, yielding the primary signal and the scatter signal, both with noise. A denoising process specifically designed for Poisson noise removal is applied to obtain a smooth scatter signal. The noise component is then obtained by combining the difference between the MC primary and the ray-tracing primary signals, and the difference between the MC simulated scatter and the denoised scatter signals. Finally, a calibration step converts the calculated noise signal into a realistic one by scaling its amplitude according to a specified mAs level. The computations of gDRR include a number of realistic features, e.g., a bowtie filter, a polyenergetic spectrum, and detector response. The implementation is fine-tuned for a GPU platform to yield high computational efficiency.

**Results:** For a typical CBCT projection with a polyenergetic spectrum, the calculation time for the primary signal using the ray-tracing algorithms is 1.2–2.3 s, while the MC simulations take 28.1–95.3 s, depending on the voxel size. Computation time for all other steps is negligible. The ray-tracing primary signal matches well with the primary part of the MC simulation result. The MC simulated scatter signal using gDRR is in agreement with EGSnc results with a relative difference of 3.8%. A noise calibration process is conducted to calibrate gDRR against a real CBCT scanner. The calculated projections are accurate and realistic, such that beam-hardening artifacts and scatter artifacts can be reproduced using the simulated projections. The noise amplitudes in the CBCT images reconstructed from the simulated projections also agree with those in the measured images at corresponding mAs levels.

**Conclusions:** A GPU computational tool, gDRR, has been developed for the accurate and efficient simulations of x-ray projections of CBCT with realistic configurations. © 2012 American Association of Physicists in Medicine. [<http://dx.doi.org/10.1118/1.4766436>]

Key words: CBCT projection, digitally reconstructed radiograph, GPU, Monte Carlo

## I. INTRODUCTION

Cone beam computed tomography (CBCT) (Refs. 1 and 2) has become an important tool in medical imaging for direct visualization of patient anatomy. In many CBCT-related research topics, for instance the design of CBCT scanners and the development of reconstruction algorithms, it is highly desirable to perform accurate and realistic simulations to obtain x-ray projection images. Not only is this a cost-effective way of acquiring data without performing real experiments,

it also offers the opportunities and freedoms to disentangle all the physical effects in CBCT, such as various types of scatter signals, so that researchers can specifically focus their studies.

Generally speaking, there are three components that one needs to consider in a projection image, namely, primary signal, scatter signal, and noise signal, all of which are of interest to certain research projects and applications. The computations of these components are very demanding, especially if one would like to achieve a high level of accuracy and realism.

Over the years, there have been a number of research efforts dedicated to the computations of these components.

The primary signal characterizes x-ray attenuation while traveling from an x-ray source to a detector pixel. This signal forms the fundamentals for the CT technology. Therefore, computation of the primary signal is widely employed in studies regarding the design and validation of reconstruction algorithms. Although it is conceptually straightforward to compute this signal by ray-tracing methods,<sup>3-5</sup> it becomes very computationally intensive to obtain accurate results in a realistic context. For instance, repeated ray-tracing calculations are needed in those cases with a polyenergetic x-ray spectrum, each corresponding to an energy channel.

Scatter signal is the second component of interest. As it is the primary contamination in CBCT imaging, calculating this signal serves as the basis for understanding, modeling, and removing scatter.<sup>6-8</sup> Monte Carlo (MC) methods have been widely used for scatter calculations<sup>9-12</sup> due to its faithful descriptions of the underlying physical process and the accurate considerations of the problem geometry. Nonetheless, the extremely prolonged computation time required to achieve an acceptable precision level has seriously impeded its applications. To speed up the computations, variance reduction techniques have been utilized.<sup>10,11</sup> Moreover, because of the smoothness of a scatter signal, it has also been proposed to compute it on a detector grid with a low resolution and a large pixel size to improve signal-to-noise ratio, and hence effectively reduce computational time.<sup>12,13</sup> Yet, the computation time, especially when computing a large number of projections, is still not satisfactory.

The third component in a projection image is noise. Since it is usually desirable to acquire CBCT projections at low mAs levels for the consideration of imaging dose reduction, studying properties of the amplified noise is necessary to facilitate the development of noise removal techniques. In the past, a number of noise models have been proposed,<sup>14-16</sup> where variance of the noise at each pixel is usually assumed to be a function of the primary signal and the parameters in these models are obtained by fitting against measurement data. However, these models generate noise signals only in a phenomenological manner, and the physical process of noise formation, namely the quantum fluctuation of photons arrival at a detector pixel, is neglected.

To our knowledge, there is no single package that computes all of these components together to an adequate degree in terms of combined accuracy, realism, and efficiency. This fact motivates us to develop a new package, gDRR, aiming at meeting all of these requirements. Generally, satisfactory accuracy and realism usually lead to compromised efficiency. gDRR overcomes this problem by employing the high-performance platform of graphics processing unit (GPU), as well as simulation algorithms and schemes suitable for GPU. Recently, GPUs have been increasingly utilized in medical physics to speed up computationally intensive tasks.<sup>17-26</sup> In particular, it has been demonstrated that GPUs can greatly enhance the MC simulation efficiency of particle transport,<sup>27-32</sup> the most computationally demanding task in x-ray projection simulations. Among them, MC-GPU

(Refs. 30 and 31) has been developed for x-ray radiograph simulations, and up to  $\sim 30$  times speedup has been observed compared to CPU simulations. Yet, the photon transport functions in MC-GPU are essentially a straightforward translation of the PENELOPE (Ref. 33) subroutines and the well-tuned implementations in PENELOPE for CPU may not be optimal on the GPU platform. Moreover, a number of realistic features required in simulations are missing in MC-GPU, such as detector response.

In this work, we will present our recent progress toward a high performance x-ray imaging simulation package, gDRR. This package computes primary signals using ray-tracing algorithms, while a MC simulation optimized for the GPU platform is employed to obtain the primary and the scatter signals with noise. A denoising procedure designed for Poisson noise removal is utilized to yield the scatter signal that is smooth across the detector. Finally, gDRR computes the difference between the simulation results with and without noise, yielding the computed noise components, whose amplitude is then properly scaled according to a specified mAs level. Realistic CBCT geometry and detailed physical aspects are considered in gDRR. The entire computation is performed on GPU, which leads to a high efficiency.

## II. METHODS AND MATERIALS

### II.A. System setup

Let us consider the geometry for a CBCT system as illustrated in Fig. 1. An x-ray source is at one side of a patient, which is able to rotate inside the  $xOy$  plane about the  $z$  axis. The location of the x-ray source  $S$  is parameterized by the source-to-axis distance (SAD) and the rotation angle  $\varphi$  between  $\vec{SO}$  and the positive  $x$  direction. An x-ray image detector is perpendicular to the source rotational plane  $xOy$  as well as the direction  $\vec{OP}$ . The imager location is defined by the axis-to-imager distance (AID) and another angle  $\theta$  between  $\vec{SO}$  and  $\vec{OP}$ . This configuration allows for an easy placement of the imager not necessarily in the opposite side of the source required in studies such as Compton scatter tomography.<sup>34</sup> A coordinate system  $(u, v)$  is defined on the detector plane with its origin at the point  $P$  and the  $v$  axis is parallel to the  $z$  axis.

gDRR computes the projections of a voxelized phantom represented by a 3D array indicated by the cube shown in Fig. 1. At each voxel, a material type index  $i(\mathbf{x})$  and a

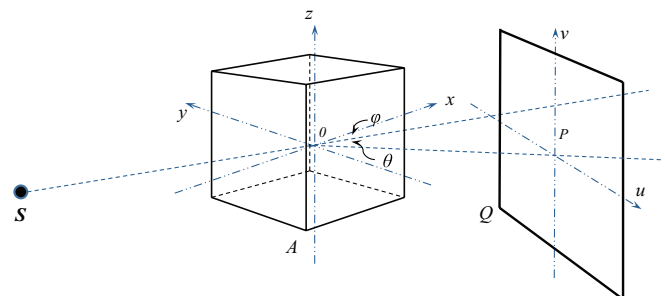


FIG. 1. An illustration of simulation geometry in gDRR.

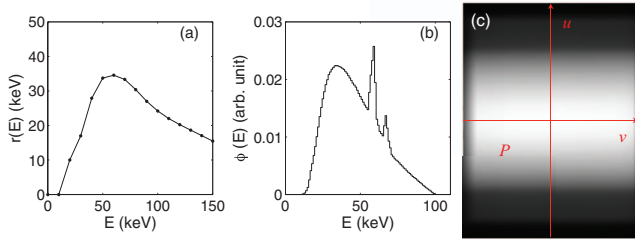


FIG. 2. From (a) to (c): a typical detector response curve, a typical 100 kVp source energy spectrum, and a photon fluence map after a full-fan bowtie filter.

density value  $\rho(x)$  are specified. X-ray mass attenuation coefficient  $\hat{\mu}_k(i, E)$  is also available corresponding to each type of material  $i$ , where the subscript  $k = 1, 2, 3$  labels the three interaction types relevant in the kilo-voltage energy regime, namely Rayleigh scattering, Compton scattering, and photoelectric effect.

As for the image detector, it is modeled to be a 2D pixel array. Each detector pixel acquires photon signals in energy integration mode, where the total photon energy deposited to the pixel is recorded. Detector response is considered in gDRR through a user supplied response curve  $r(E)$ , which specifies the amount of energy deposited by an incoming photon of energy  $E$ . An example of the detector response curve of XVI flat-panel is shown in Fig. 2(a). In principle, the detector response could also be pixel dependent. In our calculation, we ignore this pixel dependence for simplicity.

gDRR does not transport photons inside an x-ray source, e.g., the x-ray target and the bowtie filter. Therefore, all quantities used to characterize the source properties are defined after the bowtie filter. Specifically, the x-ray source is defined by its energy spectrum and its fluence map. The energy spectrum  $\phi(E)$  describes the probability density of a source photon as a function of its energy  $E$ . A typical 100 kVp energy spectrum with a tungsten target and 2 mm Al filtration is depicted in Fig. 2(b). User can specify such an energy spectrum by using the method developed by Boone and Seibert.<sup>35</sup> As for the photon fluence map,  $w(\mathbf{u})$ , it is used to specify the probability density of a photon traveling toward the detector coordinate  $\mathbf{u} = (u, v)$  after coming from the source and can be obtained by acquiring a CBCT air scan image. A typical example of the fluence map after a full-fan bowtie filter for a Varian TrueBeam On-Board-Imaging system (Varian Medical System Inc. Palo Alto, CA) is illustrated in Fig. 2(c). In

gDRR, both the spectrum and the fluence map are normalized, such that  $\int dE \phi(E) = 1$  and  $\int d\mathbf{u} w(\mathbf{u}) = 1$ . Note that this simple approach is only an effective source model. It cannot capture some realistic features in a real x-ray source, which include, but not limited to spectra variation due to the beam filtration of the bowtie filter, the bowtie filter scatter, and the detector spatial spread function. It is our future goal to further improve the degree of reality of this package and its feasibility to support more geometry and source models.

## II.B. Overall computational structure

Figure 3 illustrates the overall workflow and data flow in gDRR. Those key steps are labeled with numbers, while the shaded boxes are the data sets generated during calculations. Let us denote the primary, scatter, and noise signals by  $P(\mathbf{u})$ ,  $S(\mathbf{u})$ , and  $N(\mathbf{u})$ . In gDRR, Step 1 utilizes a ray-tracing algorithm to compute the primary x-ray attenuation signal at the detector  $P(\mathbf{u})$ . Step 2 calculates the primary signal  $\hat{P}(\mathbf{u})$  and the scatter signal  $\hat{S}(\mathbf{u})$  via MC simulations and the results contains noise due to the stochastic nature of the MC method. The difference between the noisy primary signal from the MC simulation and the noise-free one from the ray-tracing method yields the noise signal in the primary signal calculation  $\hat{N}_P(\mathbf{u}) = \hat{P}(\mathbf{u}) - P(\mathbf{u})$ . A denoising technique (Step 3) is then applied to the noisy scatter signals  $\hat{S}(\mathbf{u})$ , leading to the smoothed scatter signal  $S(\mathbf{u})$ , as well as the noise on the scatter part  $\hat{N}_S(\mathbf{u}) = \hat{S}(\mathbf{u}) - S(\mathbf{u})$ . The noise from the primary and that from the scatter add up to the total simulated noise in the projection  $\hat{N}(\mathbf{u}) = \hat{N}_P(\mathbf{u}) + \hat{N}_S(\mathbf{u})$ . Finally, a noise scaling step in Step 4 is invoked to scale the noise amplitude according to an mAs level specified in the simulation, resulting in the final noise signal  $N(\mathbf{u}) = \alpha \hat{N}(\mathbf{u})$ . After launching gDRR, three components in a CBCT projection are computed, namely the primary signal  $P(\mathbf{u})$ , the scatter signal  $S(\mathbf{u})$ , and the noise signal  $N(\mathbf{u})$ . Subsections II.C–II.F will be devoted to the description of the detailed computational strategies in those key steps.

## II.C. Ray-tracing for primary signal calculation

The primary signal at a detector pixel corresponds to the x-ray attenuation process while photons travel from the source to the detector pixel. This process can be accurately modeled

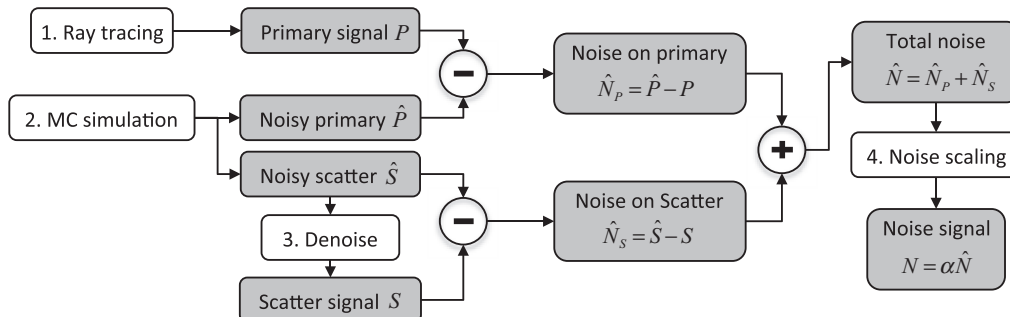


FIG. 3. Task and data flow of gDRR. Boxes with numbers are those key steps in gDRR, while shaded boxes indicate key data sets generated during calculations.

by the Beer–Lambert law. In the context with a polyenergetic source spectrum, fluence map, and detector response, the primary x-ray signal at a pixel  $\mathbf{u}$  is expressed as

$$P(\mathbf{u}) = \int dE \phi(E) w(\mathbf{u}) r(E) \exp \left[ - \int_L dl \mu(x, E) \right]. \quad (1)$$

Inside the exponential term, a line integral represents the radiological length for an energy  $E$ , where the path  $L$  is a straight line connecting the source to the detector pixel at  $\mathbf{u}$  and  $\mu(\mathbf{x}, E) = \rho(\mathbf{x}) \sum_k \hat{\mu}_k(i(\mathbf{x}), E)$  is the total x-ray linear attenuation coefficient at a spatial location  $\mathbf{x}$  for the energy  $E$ . The user should be aware that Eq. (1) assumes that an infinitely narrow beam hits on a detector pixel. In reality, due to the finite size of a pixel, multiple rays should be traced to the pixel to get a more accurate result. This strategy will inevitably lead to an increased computation time proportional to the number of rays used. Regarding its accuracy, it has been discovered that the single ray approach leads to  $\sim 0.2\%$  relative error compared to the result obtained with  $1024^2$  rays per pixel in a typical CBCT setup.<sup>36</sup>

In gDRR, the first integration over energy is approximated by a discrete summation over all energy channels considered. Within each energy channel, the evaluation of the line integral for the corresponding radiological length is needed. This line integral is usually evaluated using Siddon's ray-tracing algorithm.<sup>3</sup> However, it is known that the Siddon's algorithm leads to square-block like artifacts in the projection image, especially when the voxel size is large. In gDRR, we utilize a trilinear interpolation algorithm to generate more realistic projections.<sup>37</sup> Specifically, we divide the x-ray path into a set of intervals of equal length  $\Delta l$  labeled by  $j$  and compute the linear attenuation  $\mu(\mathbf{x}_j, E)$  at the midpoint of each interval using a trilinear interpolation scheme. The sum over all intervals  $\sum_j \Delta l \mu(\mathbf{x}_j, E)$  is considered as the radiological length. Mathematically, it can be proven that the numerical result converges to the line integral  $\int_L dl \mu(\mathbf{x}, E)$  in the limit of zero voxel size and  $\Delta l \rightarrow 0$ . To avoid over smoothing caused by a large step size  $\Delta l$ , gDRR sets  $\Delta l$  to be half of the voxel size, which have been shown to be sufficiently small in our calculations. Yet, the users should be cautious about this approach, as this is only a practical way of removing the square-block artifacts in a projection image. The ultimate solution should be using a voxel size smaller than the detector pixel size.

In terms of computation, it is straightforward to implement the algorithm on a GPU platform. By simply having each GPU thread compute the projection value at one pixel location  $\mathbf{u}$ , considerable speedup factors can be obtained due to the vastly available GPU threads. In our implementation, GPU texture memory is used to store the voxel linear attenuation coefficients to enable fast memory access. Hardware-supported linear interpolation is also used in the trilinear interpolation algorithm.

#### II.D. Monte Carlo simulation

A GPU-based MC simulation is also developed for photon transport in the energy range from 1 keV to 150 keV.

Specifically, multiple GPU threads are launched to transport a group of photons simultaneously, with one thread tracking a photon history. Within each thread, a source photon is first generated at the x-ray source according to the user specified energy spectrum and fluence map. The photon transport process is then handled by using the Woodcock tracking method,<sup>38</sup> which significantly increases the simulation efficiency of voxel boundary crossing process. Three possible physical interactions are considered for photons in this energy range, namely Compton, Rayleigh, and photoelectric absorption. In an event of photoelectric effect, the photon transport process is terminated. After Compton or Rayleigh scattering events, the scattering angles are sampled according to corresponding differential cross section formula using the techniques developed in gCTD,<sup>32</sup> a package for fast patient-specific CT/CBCT dose calculations using MC method. The photon is tracked until its energy is below 1 keV or it escapes from the phantom. This process is repeatedly performed till a preset number of photon histories are simulated. More details regarding the MC simulations of the photon transport can be found in our previous work.<sup>32</sup>

Two counters are designed to store the primary signal  $\hat{P}(\mathbf{u})$  and the scatter signal  $\hat{S}(\mathbf{u})$  at the image detector, respectively. Meanwhile, an indicator is carried by each photon that records if any scattering events have taken place during the transport. In the case when a photon exits from the phantom, a simple geometrical calculation determines if it hits the detector. If so, an amount of energy  $r(E)$  is recorded at a corresponding detector pixel in either the primary counter or the scatter counter depending on whether some scatter events have occurred, where  $E$  is the photon energy before hitting the detector and  $r(E)$  is the response curve. The user also has the option to tally the signal of a specific type, such as the first order Compton.

One issue in the MC simulation is that the way of density interpolation may impact on the projection image quality. In a MC simulation, a voxelized phantom image is defined by the user. Conventionally, it is assumed that each voxel is homogeneous, as no further information is given regarding the variations of material properties at a subvoxel length scale. Yet, akin to the aforementioned Siddon's ray-tracing algorithm for primary signal calculation, such a configuration leads to an apparent artifact in the simulated primary image due to the finite voxel size. This artifact makes the primary signal obtained from the MC simulation not compatible with the one from the trilinear interpolation algorithm, when it comes to the noise calibration step to be discussed later. To overcome this problem, we employed a trilinear interpolation strategy on the density grid used in the MC simulation. As such, whenever a density value is requested by a photon, it calculates the value at the photon's instant location using the trilinear interpolation scheme. Note that this is a practical method employed in gDRR to reduce artifacts. It does not represent the reality. For instance, this interpolation will apparently blur the boundary between organs, where the density is essentially discontinuous. As for the scatter signals, the impacts of this density interpolation scheme seems to be minimal, as the scattered photons toward various directions smear out this effect.

## II.E. Noise removal in scatter signals

Due to the randomness inherent to the MC method, noise exists in both the scatter and the primary signals. For the scatter component, it is expected that it varies smoothly along the spatial dimension. This assumption allows us to perform some noise removal techniques to retrieve the scatter signal from the noise contaminated one obtained from the MC simulations. We note that the a powerful noise removal algorithm is only utilized to estimate the scatter signals based on MC simulations with a much reduced number of photons, hence improving computational efficiency. The scatter signal obtained as such cannot be fully regarded as physically accurate. In gDRR, we develop an effective method for this purpose by solving an optimization problem. We assume the noise signal  $\hat{S}(\mathbf{u})$  at a pixel in MC simulations follows a Poisson distribution with an underlying true signal  $S(\mathbf{u})$ , which is determined by solving such an optimization problem

$$\begin{aligned} S(\mathbf{u}) &= \operatorname{argmin}_{\mathcal{S}} E[S] \\ &= \operatorname{argmin}_{\mathcal{S}} \int d\mathbf{u} (S - \hat{S} \log S) + \frac{\beta}{2} \int d\mathbf{u} |\nabla S|^2. \end{aligned} \quad (2)$$

There are two terms in the energy function  $E[S]$ . The first one is a data-fidelity term that is customized for Poisson noise,<sup>39</sup> while the second one is a penalty term that ensures the smoothness of the recovered solution  $S(\mathbf{u})$ .  $\beta$  is a constant to adjust the relative weights between the two terms. Such an energy function is convex, and hence it is sufficient to consider the optimality condition to be satisfied by the solution, namely

$$0 = \frac{\delta E}{\delta S} = \left(1 - \frac{\hat{S}}{S}\right) - \beta \nabla^2 S. \quad (3)$$

After discretizing the Laplacian operator  $\nabla^2$  using a standard numerical scheme, we arrive at

$$0 = \left[1 - \frac{\hat{S}(i, j)}{S(i, j)}\right] - \beta [\Sigma S(i, j) - 4S(i, j)], \quad (4)$$

where  $i$  and  $j$  are pixel location indices on the detector array and  $\Sigma S(i, j)$  is a short notation for  $S(i+1, j) + S(i, j+1) + S(i-1, j) + S(i, j-1)$ . We can now rearrange this equation and design an iterative scheme as

$$S^{(k+1)}(i, j) = \frac{1}{4} \left[ \Sigma S^{(k)}(i, j) - \frac{1}{\beta} \left(1 - \frac{\hat{S}(i, j)}{S^{(k)}(i, j)}\right) \right], \quad (5)$$

where the superscript  $k$  is an index for the iteration step. In practice, an successive over-relaxation algorithm<sup>40</sup> is employed to speed up the convergence, which leads to the scheme

$$\begin{aligned} S^{(k+1)}(i, j) &= (1 - \omega) S^{(k)}(i, j) \\ &+ \frac{\omega}{4} \left[ \Sigma S^{(k)}(i, j) - \frac{1}{\beta} \left(1 - \frac{\hat{S}(i, j)}{S^{(k)}(i, j)}\right) \right]. \end{aligned} \quad (6)$$

An empirical value of  $\omega = 0.8$  is used in our implementation. Although the solution is expected to be independent of the

initial guess  $S^{(0)}$  due to the convex nature of this problem, it is found that the choice of  $S^{(0)} = \hat{S}$  leads to faster convergence than other initialization values we considered. Such a denoising technique is particularly suitable for GPU-based parallel computation, as the component-wise multiplication and division in Eq. (6) can be easily carried out by GPU.

## II.F. Noise calibration

Noise signal is a random component in a projection. It varies at different irradiation levels and even the same experiment is performed repeatedly, the noise realization will be very different each time. Here, we take a simple approach that assumes that this noise component can be obtained by removing the primary and the scatter parts from the MC simulated signals. Note that the noise signal computed in this way will be different each time the simulation is performed. The noise amplitude in a real CBCT scan depends on the actual number of photons emitted by the x-ray source  $n_{\text{act}}$ , while the noise amplitude obtained in a MC simulation is governed by the number of source photons in the simulation  $n_{\text{sim}}$ , which is typically much less than  $n_{\text{act}}$ . Therefore, it is necessary to scale the noise amplitude to yield a correct level of noise. We would like to note that, this approach only considers the noise generation due to the photon number fluctuation at the detector. There are other components in a real noise signal, such as electronic noise, that are not included in this model. The overall validity of our simulation requires further investigations, which will be the future work.

The noise component on a projection image can be estimated by combining that from the primary signal and that from the scatter signal, namely

$$\hat{N} = \hat{N}_p + \hat{N}_s = (\hat{P} - P) + (\hat{S} - S). \quad (7)$$

In our simulation, the primary signal given in Eq. (1) is expressed in terms of per particle and all the MC simulation results are normalized by the number of source photons. For the noise signal obtained as such, it can be expected that the noise amplitude at a detector pixel is approximately proportional to  $1/\sqrt{n}$ , where  $n$  is the number of photons hitting the detector pixel. While the exact value of  $n$  is unknown for both the simulation study and real experiments, it is reasonable to expect that it is proportional to  $n_{\text{act}}$  in an experiment and to  $n_{\text{sim}}$  in the MC simulation. Furthermore,  $n_{\text{act}}$  is linearly related to the mAs level,  $I$ , used in an experiment. In consideration all of these factors, we propose to scale the calculated noise  $\hat{N}$  as

$$N = \alpha \hat{N} = \hat{N} \sqrt{\frac{n_{\text{sim}}}{\zeta I}}, \quad (8)$$

where  $I$  is the mAs level and  $\zeta$  is an unknown factor that can be interpreted as the effective number of source photons at unitary mAs from an x-ray source in an experiment. The exact value of  $\zeta$  apparently depends on the specific CBCT machine used and can be determined by a calibration process as described in the following.

In principle, the calibration process can be accomplished by equating the noise amplitude of the calculated projection data with that of the measurements. Let us first denote the

noise amplitude at a pixel  $\mathbf{u}$  on an x-ray projection of a calibration phantom as  $\sigma_X(\mathbf{u}, I)$  in an experiment with an mAs level of  $I$ . Meanwhile, we can obtain the calculated noise amplitude in gDRR denoted by  $\sigma_{\hat{N}}(\mathbf{u})$ . If our assumption regarding the noise model holds, it follows that

$$\sigma_X(\mathbf{u}, I) = \sqrt{\frac{n_{\text{sim}}}{\zeta I}} \sigma_{\hat{N}}(\mathbf{u}), \quad (9)$$

i.e., the function  $\sigma_X(\mathbf{u}, I)/\sigma_{\hat{N}}(\mathbf{u})$  should be a constant value that is independent of the coordinate  $\mathbf{u}$  and only depends on the mAs level  $I$  for a given number of photons in the simulation. The level of this constant linearly decreases as  $1/\sqrt{I}$ , as the mAs level increases, and the slope of this linear relationship indicate the level of  $\zeta$ .

As such, let us take the calibration of gDRR against a kV on board imaging (OBI) system integrated in a TrueBeam medical linear accelerator (Varian Medical System, Palo Alto, CA) as an example. We have acquired CBCT scans of a Catphan<sup>®</sup>600 phantom (The Phantom Laboratory, Inc., Salem, NY) under various mAs levels. We specifically focus on the regions on the projection image corresponding to the homogeneous phantom layer. For a fixed coordinate  $\mathbf{u}$  inside this region, the standard deviation  $\sigma_X(\mathbf{u}, I)$  can be estimated by using the pixel values at this coordinate in the projections at different angles. The underlying assumption is that the projection geometry and the phantom in this layer is approximately rotationally symmetric, and pixel values at a fixed coordinate in different projections can be interpreted as results from different experimental realizations. Meanwhile, the Catphan phantom is digitized based on its CT image and the projection images are calculated using gDRR. The standard deviation  $\sigma_{\hat{N}}(\mathbf{u})$  can be determined in the same manner. For each mAs level of the experimentally acquired data, we compute  $\sigma_X(\mathbf{u}, I)/\sigma_{\hat{N}}(\mathbf{u})$  and plotted this value as a function of  $\mathbf{u}$ . The resulting constant, independent of  $\mathbf{u}$ , serves as a test regarding the validity of the noise model in Eq. (8). Finally, we plot the spatial average value of  $\sigma_X(\mathbf{u}, I)/\sigma_{\hat{N}}(\mathbf{u})$  as a function of  $1/\sqrt{I}$  and the data are found to be on a straight line. A linear regression yields the slope  $k$  of this line and hence the level of  $\zeta$  can be derived as  $\zeta = n_{\text{sim}}/k^2$ .

## II.G. Validation studies

We have performed calculations on a Catphan phantom and a head-and-neck (HN) cancer patient to demonstrate the feasibility of using our gDRR package for computing realistic CBCT projection images. Meanwhile, the computational efficiency is assessed by recording the computation time at each key step. For the hardware used in this section, the GPU is an NVIDIA GTX580 card equipped with 512 processors and 1.5 GB GDDR5 memory. A desktop computer with a 2.27 GHz Intel Xeon processor and 4 GB memory is also used, on which EGSnrc code is executed for the purpose of validating our MC simulations.

Our computations are conducted under a geometry resembling that of the kV OBI system on a TrueBeam linear accelerator. The x-ray source-to-axis distance is 100 cm and the source-to-detector distance is 150 cm. The x-ray imager reso-

lution is  $512 \times 384$  with a pixel size of  $0.776 \times 0.776$  mm<sup>2</sup>. The detector is positioned in the opposite direction to the source, namely  $\theta = 0^\circ$  in Fig. 1, and the point  $P$  is at the center of the detector for a full-fan scan setup. For the purpose of demonstrating principles, the x-ray source energy spectrum, the detector response, and the source photon fluence map are chosen as those shown in Fig. 2. A Catphan phantom is used for calibration with a size of  $256 \times 256 \times 70$  voxels, and that for the HN patient is  $512 \times 512 \times 100$  voxels. The resolutions in a transverse slice are  $1.0 \times 1.0$  mm<sup>2</sup> and  $0.976 \times 0.976$  mm<sup>2</sup>, respectively. Both phantoms have the slice thickness of 2.5 mm.  $5 \times 10^8$  source photons are simulated for each projection image in MC simulations, unless stated otherwise.

## III. RESULTS

### III.A. Primary signal

The first result we present is the primary signal of the projection image for a HN patient with the CBCT source on the left side. Figure 4(a) shows the ray-tracing result with the trilinear interpolation algorithm. We have also presented the primary signals computed by the MC simulation with density interpolation in Fig. 4(b). These two figures are visually close to each other, although a certain amount of noise presents in the MC results. Artifacts caused by the finite voxel size are still observable, especially in the zoom-in view. Figure 5 plots the profiles along the coordinate axes  $u$  and  $v$  shown in Fig. 2(c) of the primary signal  $P$  computed from the trilinear ray-tracing algorithm and  $\hat{P}$  from the MC simulations with density interpolation switched on. These two signals agree well. Taking the difference between  $P$  and  $\hat{P}$  yields the noise  $\hat{N}_P$ . Note that the amplitude of  $\hat{N}_P$  is governed by the number of photons simulated in the MC simulation, but do not represent the real noise level in an experiment.

### III.B. Scatter signal

For the scatter signal calculations, we first show in Figs. 6(a) and 6(b) the signals calculated for a HN patient along the left-right projection direction with  $5 \times 10^8$  and  $5 \times 10^{10}$  source photons, respectively. There is a visible amount of noise in Fig. 6(a). The noise level is diminishing when the source photon number becomes large in

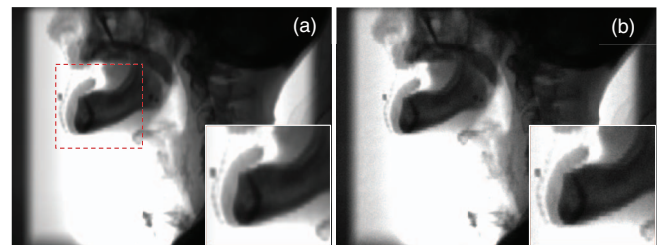


FIG. 4. Primary signal simulated in gDRR of a HN patient by ray-tracing method using the trilinear interpolation algorithm (a) and MC simulations with density interpolation (b). Insets show a zoomed-in view of the area indicated by the square in (a).

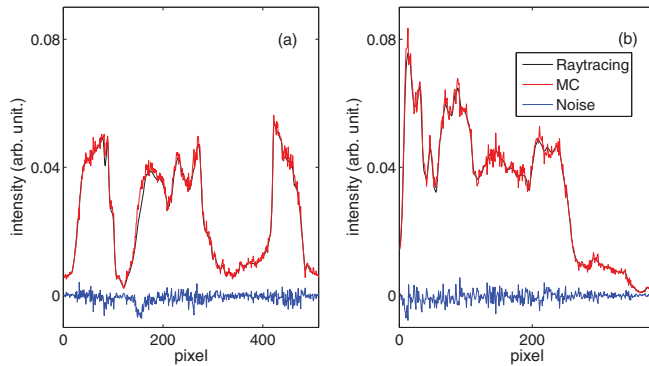


FIG. 5. Intensity profiles along the  $u$  and the  $v$  axes of the primary signal  $P$  computed from trilinear ray-tracing and  $\hat{P}$  from MC simulations with density interpolation switched on. The difference between them is the unscaled noise  $\hat{N}_P$ .

Fig. 6(b), as expected. In Fig. 6(c), the scatter image after removing the noise component from Fig. 6(a) is presented. Visually, Figs. 6(b) and 6(c) are very close to each other. We have also quantitatively computed the relative difference between these two images, e.g.,  $\|\hat{S}_{5 \times 10^{10}} - S_{5 \times 10^8}\|_2 / \|\hat{S}_{5 \times 10^{10}}\|_2$ , where  $\hat{S}_{5 \times 10^{10}}$  is the MC simulated image in (b) and  $S_{5 \times 10^8}$  is the denoised image in (c).  $\|\cdot\|_2$  stands for the L-2 norm, namely  $\|S\|_2 = [\sum_i S_i^2]^{1/2}$ , where the summation is over all the entries of  $S$ . This relative difference is about 2%, indicating the effectiveness of this denoising algorithm. The small, but finite, relative difference can be ascribed into the residual noise component in Fig. 6(b), though it is hardly visible in the image.

We further plot the scatter image profiles along the  $u$  and the  $v$  axes in Figs. 6(d) and 6(e), as well as the difference between the MC result  $\hat{S}$  and the denoised result  $S$ , namely  $\hat{N}_S = \hat{S} - S$ . Finally, in Fig. 6(f) we depict the profiles of

the denoised image based on simulations with a wide range of number of photons. The image profiles from  $5 \times 10^7$  to  $5 \times 10^{10}$  photons almost coincide on a single curve, and only in the case with  $5 \times 10^6$  photons do we observe a slight difference.

To test the accuracy of the simulated scatter components, we have validated our simulations against EGSnrc,<sup>9,10</sup> a commonly used MC package for photon transport. For this purpose, the scatter image under the identical configuration as in the above case is performed with EGSnrc. One billion source photons are used in the simulation. Due to the different signal intensities from EGSnrc and from gDRR, we rescale the EGSnrc result, so that its mean value equals to that of the gDRR results. The resulting scatter signal and the corresponding denoised image are shown in Figs. 7(a) and 7(b), respectively. Visually, these two images are indistinguishable from the corresponding results in gDRR, namely Figs. 6(a) and 6(c). We have also plotted the denoised image profiles along the  $u$  axis for the two simulation packages. A good agreement is observed in Fig. 7(c).

Finally, we calculated the relative difference of the scattering signals  $\|S_{\text{gDRR}} - S_{\text{EGSnrc}}\|_2 / \|S_{\text{EGSnrc}}\|_2 = 3.8\%$ . This value quantitatively indicates the accuracy of the scatter simulations in gDRR.

### III.C. Noise signal

The noise component calibration as described in Sec. II.F is conducted. For a Catphan phantom at the homogeneous layer as indicated in Fig. 8(a), we first calculated the quantity  $\sigma_X(u, I) / \sigma_{\hat{N}}(u)$  at a given mAs level  $I$  and plotted the results as a function of  $u$  in Fig. 8(b). Apart from the noise, this quantity is almost a constant independent of the pixel location  $u$ , which indicates the validity of our noise model.

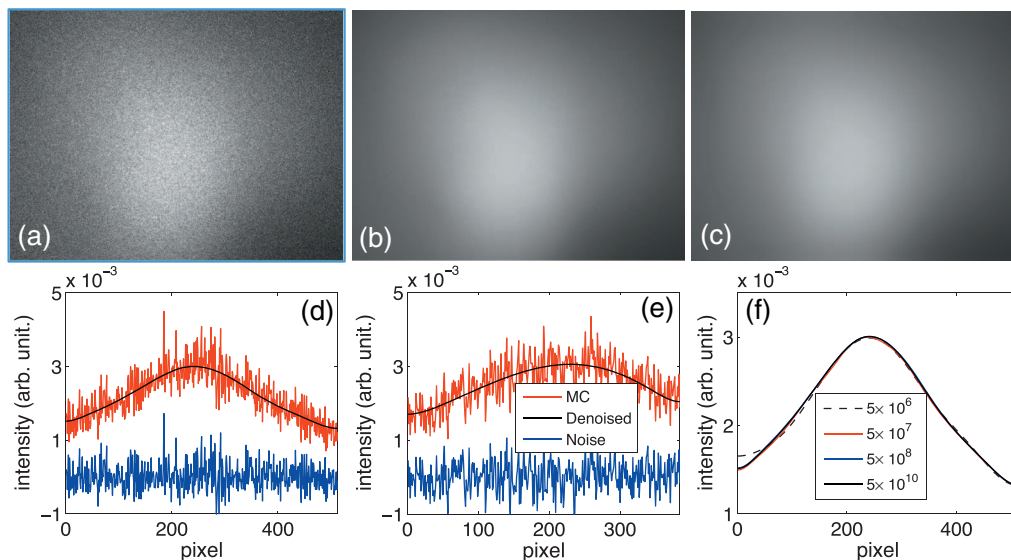


FIG. 6. (a) and (b) are scatter signals of a HN patient simulated using gDRR with  $5 \times 10^8$  and  $5 \times 10^{10}$  source photons. (c) shows the denoised scatter image from simulations in (a). (d) and (e) show the signal profiles along the  $u$  and the  $v$  axes, respectively. (f) is the comparisons of the denoised scatter image profiles along the  $u$  axis with various number of photons.

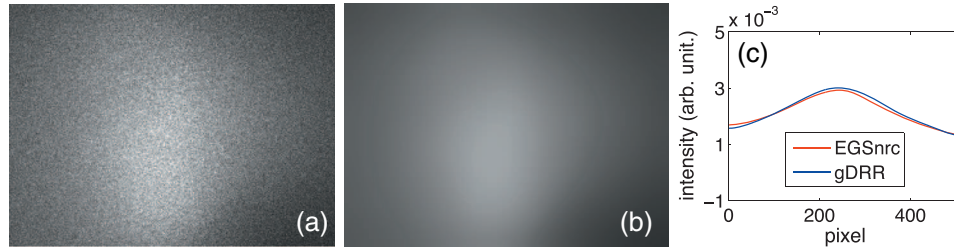


FIG. 7. (a) Scatter signal of the HN case using EGSnrc. (b) Denoised result of (a). (c) A comparison of denoised scatter image profiles along the  $u$  axis between the EGSnrc result and the gDRR result.

Moreover, we average  $\sigma_X(u, I)/\sigma_{\hat{N}}(u)$  for a range of pixel locations  $u$  for a given level of mAs and then plot the averaged  $\langle \sigma_X(u, I)/\sigma_{\hat{N}}(u) \rangle$  as a function of  $1/\sqrt{I}$  in Fig. 8(c). The resulting data are found to be along a straight line. A linear regression yields the slope  $k = 0.3074$  and hence the level of  $\zeta$  can be derived as  $\zeta = \frac{n_{\text{sim}}}{k^2} = 5.2913 \times 10^9 \text{mAs}^{-1}$ . This value enables us to convert the simulated noise signal into the realistic noise signal according to the given number of photons and the desired mAs level based on the Eq. (8). Note that detector spatial spread function is not included in our simulation. Neglecting this realistic feature could in principle lead to over estimation of noise levels. Yet, the calibration procedure converts the simulated noise level to the measured noise level, which may partially account for the detector spatial spread issue regarding the noise signal. In practice, the detector spatial spread could be included by a convolution process in the simulation of projection images, which will be available in our future release of this package.

### III.D. Computation time

To assess the computational efficiency, we have recorded the computation time of each key step in both the Catphan phantom case and the HN patient case. The results are summarized in Table I. All computation time are expressed as per projection.  $5 \times 10^8$  photons are used in MC simulations. A

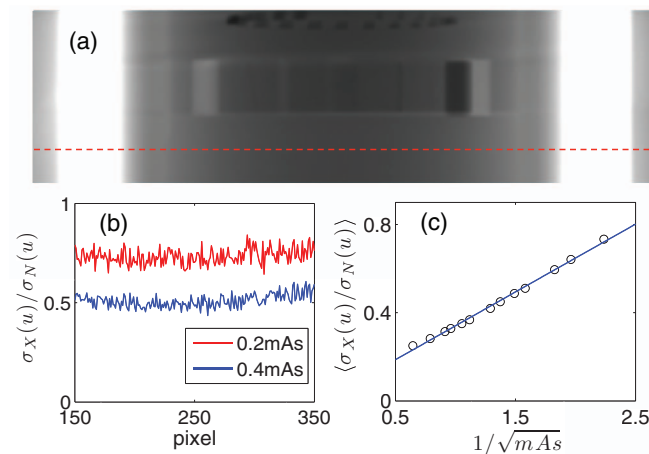


FIG. 8. (a) One projection of the Catphan phantom. A dashed line indicates the location used for noise calibration. (b) The values of  $\sigma_X(u, I)/\sigma_{\hat{N}}(u)$  as a function of  $u$  at two different mAs levels. (c)  $\langle \sigma_X(u, I)/\sigma_{\hat{N}}(u) \rangle$  as a function of  $1/\sqrt{\text{mAs}}$  and the straight line is the linear fit.

polyenergetic spectrum with 92 energy channels is used in these cases. Hence the ray-tracing calculation time is longer than what has been previously reported in other similar research works.<sup>41,42</sup> The ray-tracing time of the patient case is about two times longer than that of the Catphan phantom due to the doubling of voxels in each axis inside a transverse plane. As for the MC simulation time, the time is almost tripled. In addition to the more voxel numbers, it can also be ascribed to the fact that the maximum photon attenuation coefficient in the patient case is larger than that of the Catphan phantom. Due to the application of Woodcock transport in MC, this fact creates more fictitious photon interactions, reducing the computational efficiency.<sup>32</sup> The computation time for the denoising and noise calibration part is independent of the phantoms, as both tasks operate in the projection image domain. Among all of the steps in gDRR, MC simulation is the most time-consuming. Comparing with the ray-tracing calculations and the MC simulations, the time spent on denoising and noise calibration tasks is negligible. Overall, this recorded time clearly indicate the achieved high efficiency in gDRR. For instance, the computation time for EGSnrc is a few CPU hours. Yet, we would like to point out that the comparison of the computation time with EGSnrc is not fair, as the latter utilizes much more detailed simulation schemes to handle photon transport.

### III.E. CBCT artifacts

It is well known that various artifacts can be observed in realistic CBCT images due to various physical processes involved in data acquisition. To demonstrate the feasibility of using gDRR to compute realistic x-ray projection images, we have simulated 360 projection images of the HN patient in an angular range of  $2\pi$  at 0.6 mAs/projection and have reconstructed the CBCT image using an FDK algorithm. Various artifacts are observed in the CBCT images reconstructed as

TABLE I. Computation time of each key step in gDRR. MC simulation time for EGSnrc is also included.

	Ray-tracing (s)	MC simulation		Denoising and noise calibration (s)
		gDRR (s)	EGSnrc (CPU-s)	
Catphan	1.2	28.1	$7.2 \times 10^3$	0.041
HN patient	2.3	95.3	$2.7 \times 10^4$	0.042



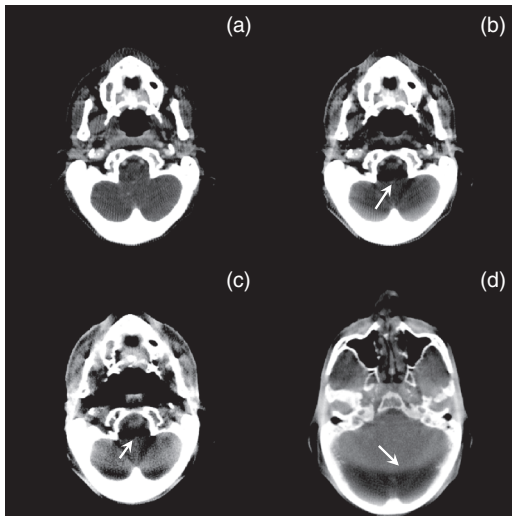


FIG. 9. CBCT images reconstructed from the simulated projections. (a) From only primary signal with a monoenergetic source. (b) From primary signal with a polyenergetic source. (c) and (d) From all three components. The display window is  $[-100, 340]$  HU for (a),  $[20, 460]$  HU for (b) and (c), and  $[-400, 890]$  HU for (d). Arrows indicate various CBCT artifacts.

such. First, Figs. 9(a) and 9(b) show one slice of the CBCT with a monoenergetic 60 keV source and a polyenergetic 100 kVp source, respectively. Only primary signals are used in these two reconstructions. Comparing these two images, artifacts caused by beam-hardening effect are clearly observed in (b), as indicated by the arrow. Figure 9(c) is the same CBCT slice but reconstructed with all of the primary, the scatter, and the noise signals. A polyenergetic 100 kVp source is used in this case. Apart from the obvious level of noise due to the inclusion of noise signal in the projections, scatter-caused artifacts are also observed, which reduce the overall image contrast, and strengthen the artifacts evidenced by the arrow. Finally, in Fig. 9(d) we show another slice of the CBCT reconstructed in the same simulation setup as in (c), but at a different display window level. An obvious ring shadow artifact presents due to the interplay between the scatter signal and the bow-tie filter. This artifact can also be observed in Fig. 9(c).

As a validation of the noise model, we have attempted to reconstruct the CBCT image of the Catphan phantom using the simulated projections at various mAs levels. A square region of interest (ROI) is selected in the center of the transverse CBCT slice inside the homogeneous phantom layer and noise amplitude in the reconstructed CBCT images  $\sigma_{sim}$  are measured as the standard deviation inside the ROI. Meanwhile, such a phantom is scanned under the CBCT system with the same mAs levels and the CBCT images are reconstructed. Noise amplitude  $\sigma_{exp}$  is also measured in the same ROI. The same FDK algorithm is used to reconstruct CBCT images in all cases for a fair comparison between the simulation and the experimental studies. Figure 10 plots the noise amplitudes  $\sigma_{sim}$  and  $\sigma_{exp}$  as functions of  $1/\sqrt{\text{mAs}}$ . The two function curves are in good agreement, indicating the capability of gDRR in terms of reproducing noise signals in

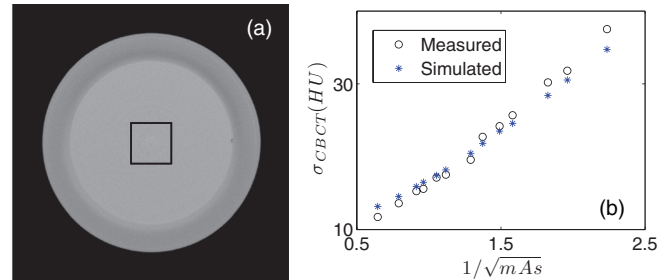


FIG. 10. (a) The homogeneous layer of the Catphan phantom. The square indicates the ROIs selected for noise comparison. (b) Comparison of the noise amplitudes from the simulated results and from the experimental results.

the projection images and hence in the reconstructed CBCT images.

#### IV. CONCLUSION AND DISCUSSIONS

In this paper, we have presented our recent progress toward the development of a GPU-based package gDRR for the simulations of x-ray projections in CBCT with a number of realistic features included, e.g., a bowtie filter, a polyenergetic spectrum, and detector response. The input of gDRR includes a voxelized phantom data that defines material type and density at each voxel, x-ray projection geometry, as well as source and detector properties. gDRR then computes three components, namely primary, scatter, and noise at the detector. The primary signal is computed by a trilinear ray-tracing algorithm. A MC simulation is then performed, yielding the primary component and the scatter component, both with noise. A denoising process specifically designed for Poisson noise removal is applied to generate the smooth scatter signal. The noise component is then obtained by taking the sum of the difference between the MC primary and the ray-tracing primary, and the difference between the MC simulated scatter and the denoised scatter. Finally, a calibration stage converts the calculated noise to a realistic noise by scaling its amplitude according to the desired mAs levels. The calculated projections are found to be realistic, such that various artifacts in real CBCT images can be reproduced by the simulated projections, including beam hardening, scattering, and noise levels. gDRR is developed on the GPU platform with a finely tuned structure and implementations to achieve a high computational efficiency.

Although gDRR is initially developed and calibrated for the OBI system on a TrueBeam machine, with simple modifications, it can also be applied to the simulations of x-ray projection images in other geometry, such as C-arm CBCT. Also, each component of the package, namely the calculations of the primary and the scatter signals, can be singled out for different research purposes. The scatter signals simulations can also be configured to tally scatter photons of different types and orders. These features greatly enable the wide applicability of gDRR and facilitate CBCT-related research projects in a variety of contexts. The entire package will be in public domain for research use, and is currently available upon request.

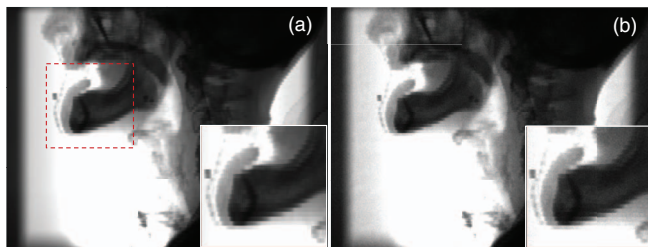


FIG. 11. Primary signal simulated in gDRR of a HN patient by ray-tracing method using the Siddon's algorithm (a) and MC simulations without density interpolation (b). Insets show a zoomed-in view of the area indicated by the square in (a).

Siddon's ray-tracing algorithm is also available in gDRR for primary signal calculation, although it is not the default algorithm for this purpose. A user can select this option, if the block artifacts are not a concern and a higher computational efficiency is more desired. Figure 11(a) shows the primary projection signal for the same patient case as in Fig. 4. When comparing Fig. 11(a) and Fig. 4(a), especially the insets, it is found that the projection computed by the Siddon's algorithm has more apparent artifacts caused by the finite voxel size. Using a smaller voxel size in the simulations can reduce this artifact. On the other hand, because of the absence of trilinear interpolation and hence the reduced memory access, the Siddon's algorithm attains a higher computational efficiency, such that the computation time for the result in Fig. 11(a) is 1.9 s, a 17% improvement compared to the time of 2.3 s reported in the Table I. Similarly, when the density interpolation is switched off in the MC simulations, the finite voxel size causes more obvious artifacts in the primary projection image, as illustrated in Fig. 11(b). The computation time is, however, shortened from 95.3 s to 80.7 s. The impacts of density interpolation on the scatter signal calculation are not observed.

Another important question is how to determine the number of photons used in a MC simulation, so that the denoising algorithm can produce a good scatter estimation. Generally speaking, the photon number should be large to provide enough photons at detector pixels, which ensures a high signal-to-noise ratio to allow the utility of the denoise algorithm. Whether this is satisfied depends on many factors, such as source photon fluence map, phantom scatter process, and detector pixel resolution, etc. For the HN case studied in this paper, it seems that  $5 \times 10^7$  source photons are sufficient. For a general case with not too much different in phantom composition and scanner setup, a photon number of the same order of magnitude should also give a good scatter signal. If the pixel resolution is different, the number should be scaled accordingly. Of course this is a simple rule of thumb estimation. One can start with this number of photons in a simulation, and increase it if necessary.

Regarding the noise calculation, another possible and simpler approach is to add random Poisson noise on the combined primary and scatter image. There are two reasons that we develop the noise model here. First, the noise obtained in this model is based on the physical process of photon count-

ing at a detector pixel. Although in general one considers that the noise in a real projection follows a Poisson distribution, the true distribution may be complicated, if many details are considered, such as the polyenergetic spectrum, the energy integration of a detector, and its response. In contrast, the direct simulation in our model includes these details naturally. Second, there is no much additional computational burden in our model. As long as the MC and the primary ray-tracing are conducted, the noise signal is available with little computations, as the time for denoise and noise calibration is very minimal, see Table I.

Note that a general assumption of gDRR is that the input volumetric data are of a high quality, e.g., noise-free and fully corrected from artifacts. In the validations of this paper, the volumetric data are taken from high quality CT scans and are assumed to be free of noise and artifacts. Yet, in the case with imperfection of the input data, the users should be aware that the simulated results will be affected and may not truly reflect the reality any more, although they are still accurate with respect to the input data.

## ACKNOWLEDGMENTS

This work is supported in part by National Institutes of Health (NIH) Grant No. 1R01CA154747-01, the University of California Lab Fees Research Program, the Master Research Agreement from Varian Medical Systems, Inc., and the Early Career Award from the Thrasher Research Fund.

- <sup>a)</sup>Authors to whom correspondence should be addressed. Electronic addresses: sbjiang@ucsd.edu and xunjia@ucsd.edu
- <sup>1</sup>D. A. Jaffray and J. H. Siewerdsen, "Cone-beam computed tomography with a flat-panel imager: Initial performance characterization," *Med. Phys.* **27**, 1311–1323 (2000).
- <sup>2</sup>D. A. Jaffray *et al.*, "Flat-panel cone-beam computed tomography for image-guided radiation therapy," *Int. J. Radiat. Oncol., Biol., Phys.* **53**, 1337–1349 (2002).
- <sup>3</sup>R. L. Siddon, "Fast calculation of the exact radiological path for a 3-dimensional CT array," *Med. Phys.* **12**, 252–255 (1985).
- <sup>4</sup>F. Jacobs *et al.*, "A fast algorithm to calculate the exact radiological path through a pixel or voxel space," *J. Comput. Inf. Technol.* **6**, 89–94 (1998).
- <sup>5</sup>S. J. Tang *et al.*, "X-ray projection simulation based on physical imaging model," *J. X-Ray Sci. Technol.* **14**, 177–189 (2006).
- <sup>6</sup>J. Star-Lack *et al.*, "Efficient scatter correction using asymmetric kernels," *Proc. SPIE* **7258**, 72581Z–72581Z-12 (2009).
- <sup>7</sup>M. Sun and J. M. Star-Lack, "Improved scatter correction using adaptive scatter kernel superposition," *Phys. Med. Biol.* **55**, 6695–6720 (2010).
- <sup>8</sup>L. Zhu *et al.*, "Scatter correction for cone-beam CT in radiation therapy," *Med. Phys.* **36**, 2258–2268 (2009).
- <sup>9</sup>I. Kawrakow, "Accurate condensed history Monte Carlo simulation of electron transport. I. EGSnrc, the new EGS4 version," *Med. Phys.* **27**, 485–498 (2000).
- <sup>10</sup>E. Mainegra-Hing and I. Kawrakow, "Fast Monte Carlo calculation of scatter corrections for CBCT images," *J. Phys.: Conf. Ser.* **102**, 012016 (2008).
- <sup>11</sup>E. Mainegra-Hing and I. Kawrakow, "Variance reduction techniques for fast Monte Carlo CBCT scatter correction calculations," *Phys. Med. Biol.* **55**, 4495–4507 (2010).
- <sup>12</sup>G. Poludniowski *et al.*, "An efficient Monte Carlo-based algorithm for scatter correction in keV cone-beam CT," *Phys. Med. Biol.* **54**, 3847–3864 (2009).

- <sup>13</sup>H. Yan et al., "Projection correlation based view interpolation for cone beam CT: Primary fluence restoration in scatter measurement with a moving beam stop array," *Phys. Med. Biol.* **55**, 6353–6375 (2010).
- <sup>14</sup>H. Lu et al., "Noise properties of low-dose CT projections and noise treatment by scale transformations," *2001 IEEE Nuclear Science Symposium Conference Record (Cat. No. 01CH37310)* (2002), Vol. 3, pp. 1662–1666.
- <sup>15</sup>J. Wang et al., "Noise properties of low-dose X-ray CT sinogram data in Radon space," *Proc. SPIE* **6913**, 69131M-1-69131M-10 (2008).
- <sup>16</sup>J. Wang et al., "An experimental study on the noise properties of x-ray CT sinogram data in Radon space," *Phys. Med. Biol.* **53**, 3327–3341 (2008).
- <sup>17</sup>S. S. Samant et al., "High performance computing for deformable image registration: Towards a new paradigm in adaptive radiotherapy," *Med. Phys.* **35**, 3546–3553 (2008).
- <sup>18</sup>R. Jacques et al., "Towards real-time radiation therapy: GPU accelerated superposition/convolution," *Comput. Methods Programs Biomed.* **98**, 285–292 (2010).
- <sup>19</sup>S. Hissoiny, B. Ozell, and P. Després, "Fast convolution-superposition dose calculation on graphics hardware," *Med. Phys.* **36**, 1998–2005 (2009).
- <sup>20</sup>C. Men et al., "GPU-based ultra fast IMRT plan optimization," *Phys. Med. Biol.* **54**, 6565–6573 (2009).
- <sup>21</sup>X. Gu et al., "GPU-based ultra fast dose calculation using a finite size pencil beam model," *Phys. Med. Biol.* **54**, 6287–6297 (2009).
- <sup>22</sup>X. Jia et al., "GPU-based fast cone beam CT reconstruction from under-sampled and noisy projection data via total variation," *Med. Phys.* **37**, 1757–1760 (2010).
- <sup>23</sup>X. Gu et al., "Implementation and evaluation of various demons deformable image registration algorithms on a GPU," *Phys. Med. Biol.* **55**, 207–219 (2010).
- <sup>24</sup>C. H. Men et al., "Ultrafast treatment plan optimization for volumetric modulated arc therapy (VMAT)," *Med. Phys.* **37**, 5787–5791 (2010).
- <sup>25</sup>X. Jia et al., "GPU-based iterative cone beam CT reconstruction using tight frame regularization," *Phys. Med. Biol.* **56**, 3787 (2011).
- <sup>26</sup>X. J. Gu et al., "A GPU-based finite-size pencil beam algorithm with 3D-density correction for radiotherapy dose calculation," *Phys. Med. Biol.* **56**, 3337–3350 (2011).
- <sup>27</sup>X. Jia et al., "Development of a GPU-based Monte Carlo dose calculation code for coupled electron-photon transport," *Phys. Med. Biol.* **55**, 3077–3086 (2010).
- <sup>28</sup>S. Hissoiny et al., "GPUMCD: A new GPU-oriented Monte Carlo dose calculation platform," *Med. Phys.* **38**, 754–764 (2011).
- <sup>29</sup>X. Jia et al., "GPU-based fast Monte Carlo simulation for radiotherapy dose calculation," *Phys. Med. Biol.* **56**, 7017–1031 (2011).
- <sup>30</sup>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.* **36**, 4878–4880 (2009).
- <sup>31</sup>A. Badal and A. Badano, "Monte Carlo simulation of X-ray imaging using a graphics processing unit," *2009 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC 2009)* (2009), pp. 4081–4084.
- <sup>32</sup>X. Jia et al., "Fast Monte Carlo simulation for patient-specific CT/CBCT imaging dose calculation," *Phys. Med. Biol.* **57**, 577–590 (2012).
- <sup>33</sup>J. Baro et al., "PENELOPE: An algorithm for Monte-Carlo simulation of the penetration and energy-loss of electrons and positrons in matter," *Nucl. Instrum. Methods Phys. Res. B* **100**, 31–46 (1995).
- <sup>34</sup>S. J. Norton, "Compton-scattering tomography," *J. Appl. Phys.* **76**, 2007–2015 (1994).
- <sup>35</sup>J. M. Boone and J. A. Seibert, "Accurate method for computer-generating tungsten anode x-ray spectra from 30 to 140 kV," *Med. Phys.* **24**, 1661–1670 (1997).
- <sup>36</sup>M. Folkerts, X. Jia, and S. B. Jiang, "A fast GPU-optimized DRR calculation algorithm for iterative CBCT reconstruction" (unpublished).
- <sup>37</sup>A. Watt and M. Watt, *Advanced Animation and Rendering Techniques: Theory and Practice* (Addison-Wesley, Reading, MA, 1992).
- <sup>38</sup>E. Woodcock et al., Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. Applications of Computing Methods to Reactor Problems: Argonne National Laboratories Report (1965), p. ANL-7050.
- <sup>39</sup>T. Le, R. Chartrand, and T. J. Asaki, "A variational approach to reconstructing images corrupted by Poisson noise," *J. Math. Imaging Vision* **27**, 257–263 (2007).
- <sup>40</sup>G. H. Golub and C. F. van Loan, *Matrix Computation* (Johns Hopkins University Press, 1996).
- <sup>41</sup>M. Folkerts et al., "Implementation and evaluation of various DRR algorithms on GPU," *Med. Phys.* **37**, 3367 (2010).
- <sup>42</sup>C.-Y. Chou et al., "A fast forward projection using multithreads for multi-rays on GPUs in medical image reconstruction," *Med. Phys.* **38**, 4052–4065 (2011).