A Metropolis Monte Carlo Simulation Scheme for Fast Scattering Calculation in Cone Beam CT

Yusi Chen, Jianhui Ma, Bin Li, Zhen Tian, Linghong Zhou, Xun Jia and Yuan Xu

***Abstract*—Low computational efficiency is the largest obstacle hindering practical applications of Monte Carlo particle transport simulation. We develop a new MC scheme for photon transport that employs Metropolis-Hasting sampling algorithm to conduct the simulations via a path-by-path scheme in this GPU-based Metropolis MC (gMMC) package. By using the Metropolis algorithm, gMMC is able to sample an entire path of a photon each time, extending from the x-ray source to the detector. The sampled paths over a long run will follow a distribution governed by the particle transport physics. gMMC was benchmarked against an in-house developed GPU-based Monte Carol simulation tool gMCDRR that performs simulations in the conventional particle-by-particle scheme. An example problem of x-ray scatter calculation was studied. For pencil beam case, speed-up factors of 200 to 300 times for first order scatter and 12 to 18 times for second order scatter can be achieved, while the average differences comparing to the conventional approach are within 1%.**

***Index Terms*—Mento Carlo simulation, Metropolis Hasting sampling, path-by-path scheme, scattering calculation**

I. INTRODUCTION

Monte Carlo (MC) simulation is an important numerical algorithm to handle problems in medical physics that require particle transport simulations in a high degree of accuracy. However, the low efficiency is the largest obstacle hindering practical applications of this method. One example in this category is MC-based x-ray scatter signal calculation in cone beam CT (CBCT) [1]. Scatter signal in CBCT is known to be the main factor degrading image quality. While MC method has the potential to accurately perform scatter calculation and therefore facilitate scatter removal, this method is rarely used in routine clinic due to the extremely long computation time.

Tremendous efforts have been devoted to accelerating the MC simulation process. On the algorithm side, variance reduction techniques have been introduced to improve the fast convergence. Efficiency enhancement methods have also been employed to improve computational speed at a cost of acceptable accuracy degradations. On the hardware side, novel computational platform has been employed. In addition to conventional CPU clusters, graphics processing unit (GPU) has also demonstrated its enormous capability to substantially boost the efficiency[2, 3]. The pursuit of continuous improvements of efficiency will probably never end. Higher and higher MC simulation speed will be desired by the ever increasing problem size and complexity in medical physics. Fast MC simulations will not only facilitate the translations of this novel method to clinical practice, it will also enable us to attack problems that were computationally prohibitive.

At present, majority of existing MC simulation packages perform calculations via a particle-by-particle scheme[4, 5]. This scheme spends a lot of time on the transport of simulation of particles that do not necessarily hit the detector, yielding a very low computational efficiency. The desire of improving computation efficiency motivates us to develop a new MC scheme that employs Metropolis-Hasting sampling algorithm. In this method, the simulations are conducted via a path-by-path scheme in this GPU-base Metropolis MC (gMMC) package. We sample the entire particle transport path at once and use the Metropolis algorithm to manage the relative weights between them. This algorithm allows us to focus computations only on sampling those paths that deposits signals to the detector with a high probability. It is this fact that enables the

This work was supported by the National Natural Science Foundation of China under Grant (81301940 and 81428019), National key research and development program under Grant (2016YFA0202003), Ministry of Science and Technology of China under Grant (2015BAI01B10), Guangdong Natural Science Foundation of China under Grant (2016A030310388) and Southern Medical University Startup Fund (LX2016003). The authors gratefully acknowledge their support.

Yusi Chen, Jianhui Ma, Bin Li, Linghong Zhou, Yuan Xu are with School of Biomedical Engineering, Southern Medical University, Guangzhou 510515, China.(e-mail: YX: yuanxu@smu.edu.cn; LHZ: smart@smu.edu.cn ; YSC: 772741048@qq.com;)

Xun Jia , Zhen Tian are with Department of Radiation Oncology, UT Southern Medical Center, Dallas, Texas, 75390 (e-mail:XJ: xun.jia@utsouthwestern.edu; ZT: zhen. tian@utsouthwestern.edu;)

DOI: 10.12059/Fully3D.2017-11-3202034

proposed method to achieve a much higher computational efficiency than the conventional approach.

II. METHODS

A. Metropolis-Hasting algorithm for gMMC

The Metropolis-Hasting algorithm[6, 7] is a method to draw samples from any probability distribution $P(x)$. Specifically, a sequence of random samples x_i is generated through the algorithm described in TABLE I. In line 3, the term mutation refers to generating a new sample based on the current sample. The design of the mutation strategies can be very creative and flexible. With the mutation probability $T(x \rightarrow y)$, the probability to accept this mutation in line 4 is expressed as:

$$p_{accept}(x \rightarrow y) = \min\left[1, \frac{P(y)T(y \rightarrow x)}{P(x)T(x \rightarrow y)}\right] \qquad (1)$$

Where $P(x)$ is the probability to be sampled from. Specifically, for the problem of scatter calculation in CBCT, the variable x_i in TABLE I refers to an entire photon path that starts from the x-ray source and ends on the detector.

TABLE I. Metropolis-Hasting Algorithm.

```
1   Initialize $x_0$;
2   For $i = 1$ to $N$
3       $y = \text{mutation}(x_{i-1})$;
4       Compute $p = p_{accept}(x_{i-1} \rightarrow y)$;
5       $x_i = y$ with probability $p$, and
        $x_i = x_{i-1}$ with probability $1 - p$;
6   End
```

B. gMMC for photon transport simulation

The basic idea of our MMC simulation is to sample an entire path of a particle extending from the x-ray source to the detector, as shown in Fig. 1. By using the Metropolis algorithm, the sampled paths over a long run follow a distribution governed by the particle transport physics. This approach is fundamentally different from sampling a particle at one scattering event each time and let it go freely following physics principles. The conventional approach has no control about where the particle ends, which wastes a lot of computations on tracking those particles that do not contribute to the signal of interest at the detector.

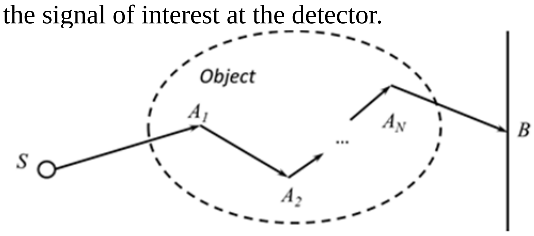

Fig. 1. Illustration of   photon transport simulation

C. Generate a new photon path

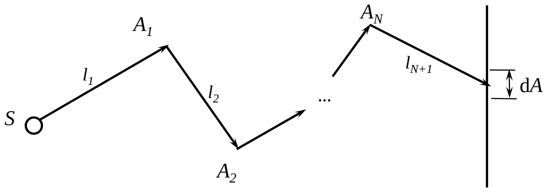

Fig. 2. Illustration of a photon path.

Let us assume a photon path starts from the x-ray source S, goes through a number of interaction points A_i, $i = 1,2,\dots,N$, and ends at a point B on the detector. Here N is the total number of interactions for this path. We sample the source energy E_0 uniformly in the possible energy range and sample number of interactions N uniformly in the range of $[0, N_{max}]$, where N_{max} is the largest number of scattering order considered. The locations of these N interaction points A_i, $i = 1,2,\dots,N$ are sampled within the object. After photon enter the object, we uniformly sample the first interaction point forward direction of photon within object. Then random-walk algorithm[8] is utilized to sample the second and high order interaction point. The interaction point is generated based on the previous scatter angle which is regard as expectation of a Gaussian distribution. The distance between the two scattering-interaction is also sampled uniformly. At each interaction point, we consider two types of photon interactions, namely Compton scatter and Rayleigh scatter. Photoelectric interaction is not considered here, as it cannot occur for those paths that extend to the detector. Finally, the detector point B is sampled uniformly on the detector.

The entire photon trajectory consists of $N+1$ segments. The probability density of this trajectory is hence the product of those for all segments, as well as those for all interaction points. Let us start with the first segment that connects the source S with the first scatter point A_1 at x_1. Suppose the x-ray source emits photons with a directional probability density function $F(x)$, i.e. the probability density of emitting a photon to a unit solid angle towards a point x. The probability density of having a photon with a certain interaction at point x_1 is $\mu(x_1)\exp[-\int_{l_1} \mu(s)\mathrm{d}s]$. Furthermore, the probability for the interaction type T_1 to occur at this point is $\mu_{T_1}(x_1)/\mu(x_1)$. Multiplying these factors these together, the probability density of having a photon moving from the source S to a unit solid angle towards x_1, experiencing an interaction T_1 at this point is

$$F(x_1)\mu_{T_1}(x_1)\exp[-\int_{l_1} \mu(s)\mathrm{d}s].$$

Similarly, we can derive the probability for the second segment. The only difference is that the probability density for the photon moving from the scatter point A_1 to the point A_2 is governed by the scattered photon angular distribution $\rho_{T_1}(x_1)$ after an event T_1. This probability density is simply the normalized differential cross section function. Note that the argument of this probability density function is actually the scatter angle defined by the event $S \to x_1 \to x_2$. However, we use x_1 here to simplify notation. With this term, the probability density for a photon to move from the point A_1 to a unit solid angle towards the point A_2, having an interaction T_2 at this point is $\rho_{T_1}(x_1)\mu_{T_2}(x)\exp[-\int_{l_2} \mu(s)\mathrm{d}s]$.

Generally speaking, the probability associated with the path l_i for $i = 2,3, \dots, N$ is

$$\rho_{T_{i-1}}(x_{i-1})\mu_{T_i}(x_i)\exp[-\int_{l_i} \mu(s)\mathrm{d}s].$$

For the last segment that extends from the last scatter point A_N to the detector, we consider a small area $\mathrm{d}A$ on the detector plane. Denote the angle between the photon travel direction and the normal direction of the detector by α. Then the probability of having the photon at the point A_N scattered towards the area $\mathrm{d}A$ on the detector is $\rho_{T_N}(x_N)\cos\alpha\,\mathrm{d}A/l_{N+1}^2$. Note that the factor $\mathrm{d}A\cos\alpha$ is the effective area that is perpendicular to the photon travel direction and $\cos\alpha\,\mathrm{d}A/l_{N+1}^2$ is the solid angle of $\mathrm{d}A$ with respect to A_N. Furthermore, the probability for the photon reaching the detector is $\exp[-\int_{l_{N+1}} \mu(s)\mathrm{d}s]$. Here an ideal detector that captures all the photons is assumed. Thus, the probability of the last segment is

$$\rho_{T_N}(x_N)\exp[-\int_{l_{N+1}} \mu(s)\mathrm{d}s]\cos\alpha\,/l_{N+1}^2\mathrm{d}A.$$

Dividing this expression by the area measure $\mathrm{d}A$ yields the probability density for this last segment.

With all the probability density defined, we can multiply all of them together to form the probability density of the entire path, yielding the expression used in our Metropolis algorithm

$$p(x) = F(x_1)\prod_{i=1}^{N}\mu_{T_i}(x_i)\rho_{T_i}(x_i)\prod_{i=1}^{N+1}\exp[-\int_{l_i} \mu(s)\mathrm{d}s]\cos\alpha\,/\,l_{N+1}^2 \quad (2)$$

We finally remark that this expression is also valid for a primary photon path that directly extents from the source to the detector. In the case of $N = 0$, which means no scattering event , one gets the probability density of $F(x_1)\frac{\exp[-\int_{l_1}\mu(s)\mathrm{d}s]}{l_1^2}\cos\alpha$.

D. Compute acceptance probability

Once a path is sampled, we first sequentially go through the interaction points to determine the energy after each interaction E_i, $i = 1,2, \dots, N$, which is needed to calculate the probability of the path. For Compton scatter, the photon energy is:

$$E_i = E_{i-1}/\left[1 + \frac{E_{i-1}}{m_e c^2}(1 - cos\varphi_i)\right] \quad (3)$$

where the item $m_e c^2$ is electron mass energy and φ_i is the scatter angle. For Rayleigh scatter, it is the elastic scatter event, which means $E_i = E_{i-1}$.

With all the probability density defined, we can multiply all of them together to form the probability density of the entire path. As for the transition probability $T(x \to y)$, since the way of generating a photon path y in the previous subsection is independent of the previous path x, $T(x \to y)$ is only a function of the path y as $T(y)$.

E. Data acquisitions

We first demonstrate the performance of our method in an Aluminum phantom, which is shown in Fig. 3 a), with a pencil beam of 60kVp normally hitting on it. The size of the phantom is 10 × 10 × 2.8 cm^3. The x-ray source to the front surface of the phantom and the detector is 14.19 cm and 65 cm, respectively. In the second case we studied, one inhomogeneous phantom illustrated in Fig.3 b) containing tissue and bone is also utilized. The size of this phantom is 12.8 × 12.8 × 12.8 cm^3.

Finally, in Fig. 3 c), a real head-and-neck cancer patient case is used. CT data is used to form the phantom. The size of phantom is 37.9 × 37.9 × 12.5 cm^3. In all the cases, the results are compared with gMCDRR with pencil beam of 60kVp photon particles, where gMCDRR is a full MC simulation package that transport particles via the conventional particle-by-particle scheme.

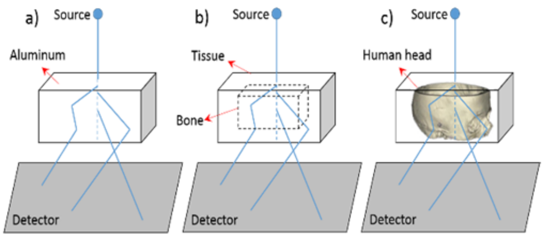

Fig. 3. Simulated phantoms in gMMC.

III. RESULTS

A. homogeneous phantom case

In the homogeneous phantom case, the computation time to transport 4.5×10^{11}

photons is 45 hrs. for achieving the high quality of the simulated result with the average stand deviation below 0.05% in gMCDRR. For gMMC, the single Compton, single Rayleigh, and multiple scattering were recorded separately to reach the same uncertainty of gMCDRR result for comparison. In gMMC, the computation time of the first order scatter signal and second order scatter signal is 13 mins. for 5×10^9 paths and 3.78 hrs. for 5×10^{10}paths, respectively. So we can reach speed up factors of around 208 times and 12 times for first order and second order scatter signal. As shown in Fig4, the signals between gMMC and gMCDRR are matched well. Note that due to the capability of specifically sampling photons hitting the detector, the needed number of paths in gMMC to reach the same level of uncertainty as in gMCDRR, and therefore number of particles is much lower. The substantially reduced number of particles yield dramatically improved computational efficiency.

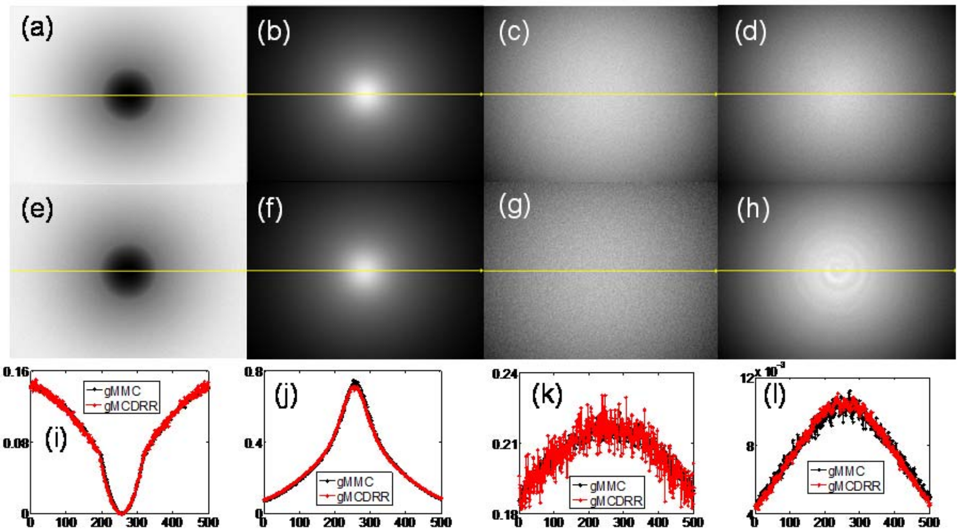

Fig. 4. (a)-(d) are the single Compton, single Rayleigh, second Compton and second Rayleigh scattering image from gMMC and (e)-(h) are that from gMCDRR. (i)~(l) are profiles on the yellow line in (a)~(h).

B. Inhomogeneous phantom case

For the inhomogeneous phantom case, the computation time is 30 hrs. after 4.5×10^{11} photons simulated in gMCDRR. This is in sharp contrast to gMMC, in which the computation time of the first order scatter signal and second order scatter signal is 5.3 mins. for 5×10^{10} paths and 1.67 hrs. for 5×10^{10} paths, respectively. As shown in Fig5, the single Compton, single Rayleigh, second Compton and second Rayleigh signals between gMMC and gMCDRR are matched well. Thus we can reach speed up around 340 times and 18 times for first order and second order scatter signal.

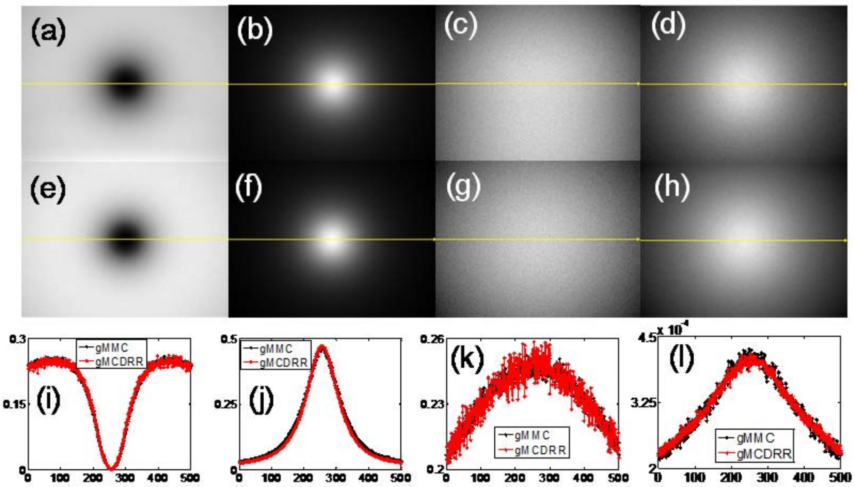

Fig. 5. (a)-(d) are the single Compton, single Rayleigh, second Compton and second Rayleigh scattering image from gMMC and (e)-(h) are that from gMCDRR. (i)~(l) are profiles on the yellow line in (a)~(h).

C. clinical case

In the real head and neck cancer patient case, the computation time is 30 hrs. after 4.5×10^{11} photons in gMCDRR. In comparison, the computation time of the first order scatter signal and second order scatter signal is 7 mins. for 5×10^{9} paths and 2.2 hrs. for 5×10^{10} paths, respectively. Therefore the speed up is around 257 times and 14 times for first order and second order scatter signal.

In this case as shown in Fig 6, discrepancy was observed between gMMC and gMCDRR for the second order scatter signal. This is ascribed to the different sampling techniques of air component in the two programs. However, since the scatter signal contributed by the second order scatter is much smaller than the first order, this discrepancy does not lead to significant difference in the overall scatter signal calculation.

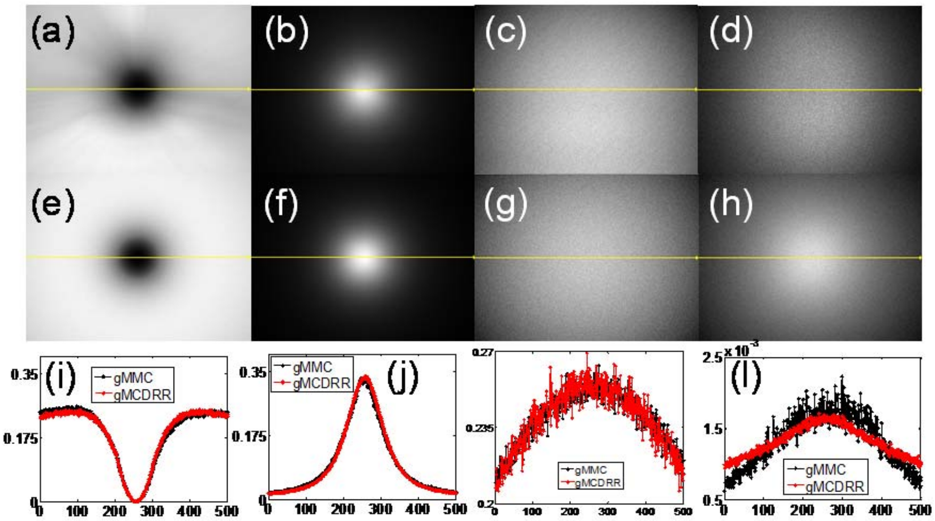

Fig. 6. (a)-(d) are the single Compton, single Rayleigh, second Compton and second Rayleigh scattering image from gMMC and (e)-(h) are that from gMCDRR. (i)~(l) are profiles on the yellow line in (a)~(h).

IV. CONCLUSION

In this study, we proposed a Metropolis-Hasting sampling algorithm to conduct the simulations via a path-by-path scheme in the MC package. Different from conventional scheme that performs the simulation in a particle-by-particle fashion, the proposed method is able to control the sampled path such that they can hit on the detector with a high probability. This fact allows for the substantial reduction of number of particles needed to reach a certain level of uncertainty. Hence, a greatly enhanced computational efficiency can be reached. In our testing pencil-beam cases, speed-up factors of 200 to 300 times for single scatter and 12 to 18 times for second order scatter can be achieved, while the average differences comparing to the conventional approach are within 1%.

REFERENCES

[1] Y. Xu, T. Bai, H. Yan, L. Ouyang, A. Pompos, J. Wang, L. Zhou, S. B. Jiang, and X. Jia, "A practical cone-beam CT scatter correction method with optimized Monte Carlo simulations for image-guided radiation therapy," Physics in Medicine & Biology, vol. 60, pp. 3567-87, 2015.

[2] G. Pratx and L. Xing, "GPU computing in medical physics: a review," Medical Physics, vol. 38, pp. 2685-97, 2011.

[3] X. Jia, P. Ziegenhein, and S. B. Jiang, "GPU-based high-performance computing for radiation therapy," Physics in Medicine and Biology, vol. 59, p. R151, 2014.

[4] X. Jia, H. Yan, L. Cervino, M. Folkerts, and S. B. Jiang, "A GPU tool for efficient, accurate, and realistic simulation of cone beam CT projections," Med Phys, vol. 39, pp. 7368-78, Dec 2012.

[5] X. Jia, H. Yan, X. Gu, and S. B. Jiang, "Fast Monte Carlo simulation for patient-specific CT/CBCT imaging dose calculation," Phys Med Biol, vol. 57, pp. 577-90, Feb 07 2012.

[6] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," Journal of Biochemical & Biophysical Methods, vol. 21, pp. 1087-1092, 1952.

[7] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," Biometrika, vol. 57, pp. 97-109, 1970.

[8] C. P. Robert and G. Casella, The Metropolis—Hastings Algorithm: Springer New York, 2004.