2011 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC): Valencia, Spain. 23-29 October 2011 (2011). IEEE, 3705-3709.
DOI: http://dx.doi.org/10.1109/NSSMIC.2011.6153699

Automatic Monte-Carlo Based Scatter Correction for X-ray Cone-Beam CT using General Purpose Graphic Processing Units (GP-GPU): A Feasibility Study

A. Sisniega, *Student Member, IEEE,* M. Abella, E. Lage, M. Desco and J.J. Vaquero, *Senior Member, IEEE*

***Abstract*– Scattered photons highly degrade the quality of X-ray images and their effect has become more important due to the increasing interest in cone-beam geometry for the acquisition of CT (CBCT) and micro-CT data. The random nature of scatter events and the great influence of the sample suggest that the most accurate methods for their estimation are Monte Carlo (MC) techniques, but their use is usually hampered by the large computation time required to obtain an acceptable estimation of the scattered radiation.**

We present an approach for scatter correction in CBCT by MC estimation, speeding up the computation by means of general purpose graphic processing units (GPGPU) and developing a framework for the automatic correction and reconstruction of projection data. The method consists of five stages: FDK reconstruction of the original data; histogram based automatic segmentation of the volume assigning a material and density to each voxel; fast MC estimation of the scatter signal; denoising of the independent scatter components and subtraction from original data; and FDK reconstruction of the corrected data. Every stage runs in a GPGPU using Nvidia CUDA.

The MC stage is based on the MC-GPU code. To simulate polychromatic X-ray beams, the Spektr model is used to generate the source spectrum. Photon scattering is forced in order to reduce the number of events needed to obtain an acceptable scatter image weighting the photon histories to assure the correctness of the result. Further reduction in the variance is obtained by split the photon in several virtual photons which are forced point to the detector and are transported with no further interaction to the detector's surface. Furthermore, the divergence of the execution path of GPGPU kernels has been minimized. These techniques achieve a reduction of the variance of the scatter signal of two orders of magnitude and the final efficiency is improved by a factor of ~30.

Results show the suitability of the proposed framework since good correction results are achieved in a reasonable time using MC only calculations.

This work has been partially funded by the Spain Ministry of Education, FPU program; Spain Ministry of Science and Innovation, projects TEC2008-06715 and TEC2007-64731; CDTI under the CENIT Programme (AMIT Project); Comunidad de Madrid (ARTEMIS S2009/DPI-1802); and EU-FP7, FMTXCT-201792

A. Sisniega, M. Abella, M. Desco and J.J. Vaquero are with Departamento de Bioingeniería e Ingeniería Aeroespacial, Universidad Carlos III de Madrid, Madrid, Spain.

E. Lage was with Unidad de Medicina y Cirugía Experimental, Hospital General Universitario Gregorio Marañón, Madrid, Spain and is now with the M+Vision Madrid-MIT consortium.

I. Introduction

Cone-beam CT (CBCT) is widely used in pre-clinical imaging and it is rapidly growing in the clinical area. One of the most important drawbacks of CBCT is the increase of scattered radiation received by the detector that worsens the image quality. Monte Carlo (MC) method is known to be the most accurate way to estimate the scatter signal, but its computational burden hampers its use for the correction of CBCT data. Recent advances in GP-GPU technology allow to speedup parallelizable tasks, such as MC calculations and open a way for the development of fast yet accurate methods for the correction of scatter effects. Here, we present a framework for the speedup and automation of MC based methods aimed at the correction of scatter effects in CBCT.

II. Method

The scatter correction framework presented consists of five stages that yield a scatter corrected tomographic volume as a final result requiring no interaction from the user. The whole process benefits of the use of GPGPU for those tasks suitable to be parallelized, using the Nvidia's CUDA architecture.

The CB projection data is first reconstructed by means of the well-known FDK algorithm which implements a modified weighted filtered backprojection for cone-beam geometry. The reconstructed data is then segmented automatically into three different materials, namely soft tissue, bone and air, yielding the voxelized volume that is used for the MC simulation. In the third stage, the scatter distribution in the CB projection data is estimated using a fast MC calculation for a subset of the acquired angular positions. The three scatter components - i.e. Compton scatter, Rayleigh scatter and multi-scattered photons - are stored in three different buffers and they are smoothed to minimize the impact of the MC limited number of photons, taking advantage of the smooth nature of the scatter signal. The smoothing kernel is different for the Rayleigh scatter component, due to its highly forward directed distribution, which makes the smoothness assumption weaker in its case.

The smoothed scatter signals are then divided by the analytical gain image, calculated using the geometrical parameters of the system, the X-ray source spectrum, the detector response and the number of photons used for the simulation. The scatter components are added to form the projection images which are interpolated in the angular domain to obtain the whole projection dataset to be subtracted from the original data. The corrected data are then reconstructed.

To assess the validity of the correction, a new segmentation is performed. If the segmented volume differs from the one used for the MC simulation, the whole process is repeated again.

The method is outlined in figure 1, and the following paragraphs describe in more detail its main steps.

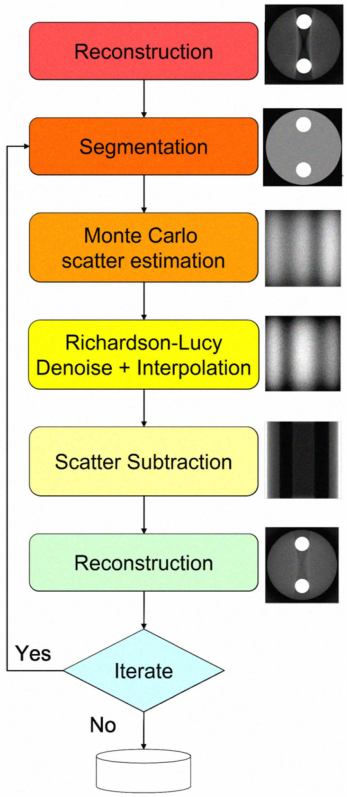

Fig. 1. Outline of the scatter correction method proposed. For datasets highly degraded by scatter, the segmentation step could assign a wrong material to some part of the volume, then a new iteration of the method must be performed if the segmented volume is different from the one obtained in the previous iteration.

A. *FDK reconstruction*

For the reconstruction of the CBCT data, an FDK reconstruction kernel has been implemented in the GPGPU. The reconstruction code makes use of the CUFFT library to perform the filtering step, providing several filtering kernels. The backprojection operation uses a parallelized voxel-driven approach which uses the GPGPU texture memory to speedup the memory access and the interpolation of the voxel's projection positions.

B. *Segmentation*

The reconstructed data is segmented using the method proposed by Otsu [1]. This algorithm performs an unsupervised segmentation, given the number of classes present in the volume. We use three different classes for the description of biological tissues: soft tissue, bone, and air. The histogram of the volume is computed and the two levels for the segmentation are found by maximizing the between-class variance, yielding the statistically best separation assuming there are three different tissues in the sample.

In the present work, best results were obtained applying a two-class version of the Otsu's algorithm to separate air from tissue, followed by a second iteration to find the best threshold for the segmentation of soft-tissue and bone.

In the described implementation, each class is defined by means of a material description plus a fixed intensity. It is straightforward to assign different densities inside a material according to original voxel values.

C. *Monte Carlo scatter estimation*

The MC engine for scatter estimation is based on the MC-GPU code [2] to which several modifications have been made to improve the performance of the scatter estimation and to allow the simulation of polychromatic X-ray beams and the accurate response of the detector. The original code is a parallelized version of some of the routines of the Penelope package [3], implemented using the CUDA programming model for GPGPUs.

To simulate polychromatic X-ray beams the TASMIP-based [4] Spektr [5] generator has been incorporated into the MC engine. During the initialization phase, the spectrum of the given source is computed and filtered applying the Beer´s law and it is saved to a text file to be read for the MC simulation. Alternatively, the user could provide a text file containing the X-ray source spectrum obtained by any other mean.

The shape of the spectrum is used as the probability density function (pdf) for the generation of the energy of the photons simulated. To minimize the number of memory accesses and achieve the better performance possible, the function is sampled using the alias method, particularly efficient in terms of memory access.

While modifying the code two main goals were considered; to achieve a lower variance of the scatter signal for the same number of primary photons simulated, and to reduce the kernel divergence – i.e. different execution paths for kernels running in parallel – which worsens to a great extent the performance of the GPGPU code.

To get the most out of the parallelization capabilities of GPGPUs when variance reduction techniques are included in the code, the photon tracking process was modified. The tracking algorithm employed by conventional MC engines and, in particular, by MC-GPU sequentially tracks a photon from the source until it is absorbed or leaves the volume. In MC-GPU, each thread sequentially tracks a group of photons. The parallelization is achieved by launching a bunch of threads simultaneously. Splitting and detection forcing techniques offer a poor performance improvement if they are directly inserted into the defined tracking scheme with no further modification, since all the split photons must be transported sequentially. To overcome this problem we propose here a new tracking scheme which split the tracking

of the photons into two stages, namely, the tracking of a photon to an interaction point and the interaction execution. The two stages are chained until the photons leave the volume.

By using the proposed tracking algorithm, the split of a photon into several *virtual* interactions forced to be detected, can be accomplished efficiently, and the track of those *virtual* photons can be performed in a fully parallel fashion. The proposed tracking is depicted in figure 2.

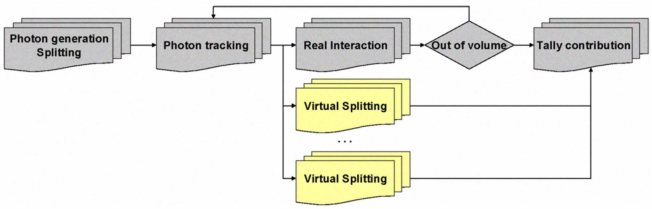

Fig. 2. Tracking algorithm proposed. The tracking of the photon is stopped at every interaction point, and the CPU takes the control of the process. Several interactions are launched in parallel in the GPU and the photons are transported directly to the detector, weighted by the splitting factor and by the accumulated attenuation probability though their path to the detector. One of the interaction results is assumed to be the real interaction and the photon parameters are stored to continue the tracking. The process finishes when the photon leaves the volume.

The variance reduction techniques developed could have a non-desired negative impact in the performance of the simulation process, for particular situations or features of the system to simulate. For that reason, it is possible to turn off each one of them, achieving the best performance possible. Below, the variance reduction techniques implemented are detailed.

First, to maximize the probability of interaction of the photons generated at the source, the sampling of their direction is biased, to prioritize directions pointing to high attenuating paths across the volume. The distribution is biased by using the projection of the volume for the effective energy of the beam as a 2-dimensional pdf which is sampled using a rejection process. Each photon receives a weight to compensate for the non-uniform sampling of the cone-beam space. Those photons directed towards low attenuating areas are split to avoid very large weight values.

Once a photon has been generated, it can be forced to interact inside the volume, by modifying the transport function using the known exponential transform [6] and assign the appropriate weight to the photon.

To reduce the computational burden of the photon transportation, the original MC-GPU implementation performed the Woodcock tracking method [7], avoiding the exact tracking of the photon and reducing the number of accesses to global memory which has a great latency for GP-GPUs. The Woodcock tracking has been preserved in the current implementation.

Once the photon has been positioned at its interaction point, a type of interaction is selected, its attributes – i.e. position, direction and energy – are stored and the thread is terminated. To avoid the time wasted tracking a photon that is absorbed at the interaction point, photons are forced to perform a scatter interaction and they are weighted to account for the absorption probability.

To maximize the outcome of every interaction, a block of N threads is launched for every interacting photon, each one of them performing a virtual interaction. The angular deflection of the virtual scattered photon is selected randomly, but constrained to point into the detector's area. To get unbiased results, the virtual photon receives a weight proportional to the probability of deflection in the given direction, as provided by the physical model implemented in Penelope, multiplied by $1/(N+1)$ to take into account the splitting of the contribution between the different photons. The virtual photon is transported to the detector without any further interaction and its contribution is tallied, weighted by the attenuation suffered through the path from the interaction point.

All the virtual photons within a block perform the same type of interaction, and are transported in parallel, minimizing the in-block divergence between threads.

After the virtual photons are tallied, the tracking of the original photons continues from the interaction point, launching a GPU thread per photon to track.

Photons arriving at the detector surface are tallied and their contribution is weighted by the detector spectral response, obtained from the cascaded model of the flat-panel detector, provided as a text file.

D. *Scatter signal denoising*

To minimize the simulation time, it is convenient to allow a certain amount of noise in the estimated scatter signal. Thus, a denoising step is necessary to obtain the desired noise-free scatter signal to subtract. In the present work, we have followed the approach proposed in [8].

The scatter signal is smoothed by means of several iterations of the Richardson-Lucy algorithm, stopping at the denoising step of the algorithm.

We propose to use different Gaussian smoothing kernels to take profit of the nature of the different types of scatter components. A wide kernel is used for the denoising of the noisier and more evenly distributed Compton and multi-scattered signals, while the forward direction of the Rayleigh signal is preserved by a narrower smoothing kernel.

The denoised projections are interpolated in case the number of projections simulated is lower than the number of projections present in the original dataset.

III. EVALUATION

We have assessed the quality of the preliminary results obtained with the proposed method, in terms of computational burden and quality of the corrected images.

The quality of the corrected data was evaluated quantitatively using profiles of the voxel values across a line on a slice of the reconstructed value, inspecting the reduction of the cupping artifact. The reduction of streaks was observed visually and is shown in a slice of the reconstructed volume.

Execution time was measured as a figure of computational performance of the method. Special attention was paid to the impact of the variance reduction techniques included into the code, by assessing the variance reduction achieved by the new simulation engine taking into account the increase in time needed for the simulation of the same number of photons. To

get a representative number we used the well known MC performance equation, given by

$$\varepsilon = \frac{1}{\sigma^2 T}$$

Where T is the time needed to obtain the simulated signal and σ^2 is the variance of the estimated signal through several runs, using different seeds for the initialization of the random number generator.

To quantify the improvement in the performance of the simulation process compared to the original one, the same volume and simulation parameters were used in the original MC-GPU code and the ratio of the performance achieved by MC-GPU to the one achieved by the modified engine was used as a figure of merit for the improvement.

We explored the effect of the different variance reduction techniques implemented, by turning them on or off, generating several combinations.

The method was tested in a simulated phantom and in real data. The phantom consisted of a 70 mm diameter cylinder of soft tissue with two bone inserts of 15 mm diameter each. The simulated dataset consisted of 360 projections (1° spaced) obtained with the MC-GPU package and was corrected using the proposed method with 90 projections. The X-ray spectrum simulated had 45 kVp energy and 1 mm Al and 0.2 mm Cu added filtration.

The real dataset was the upper part of the body and the head of a rat; acquired with an Argus-PET/CT system (Sedecal, Madrid, Spain) [9] using 45 kVp energy and 1mm Al added filtration. The dataset consisted of 360 projections (1° spaced). The simulated scatter signal consisted of 90 projections.

IV. Results

Results for the simulated dataset are shown in figure 3. Profile data showed that the cupping artifact was corrected, apart from the residual part caused by beam-hardening effects, which are not addressed in this work. Also, a great reduction in the level of streaks is observed, again not considering the residual error caused by beam hardening.

A good improvement is also observed for the small-animal dataset, shown in figure 4. Slight poorer quality was obtained for real data. We attributed this loss of quality to deviations of the system model compared to the real one arising from the animal holder and the presence of a PMMA tube inside the beam to protect the system from liquids which was not taken into account in the present model.

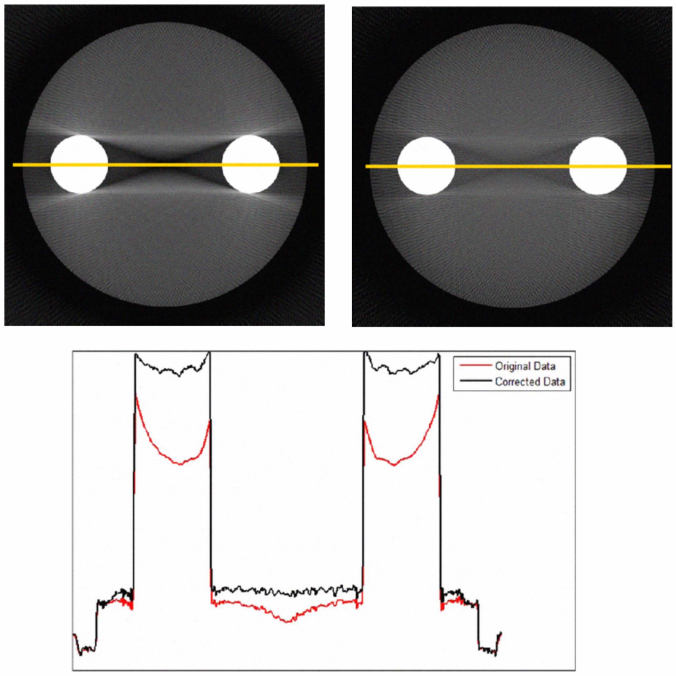

Fig. 3. 70mm Soft-Tissue cylinder with two bone inserts. Simulated using 45 kVp 1mm Al and 0.2 mm Cu. Scatter correction calculated using $5x10^6$ photons (~14 s/prj) and 90 projections and 3 Richardson-Lucy iterations with a Gaussian kernel (10px for Compton and Multi, 3px for Rayleigh). Note the reduction of the cupping artifact and of the streaks.

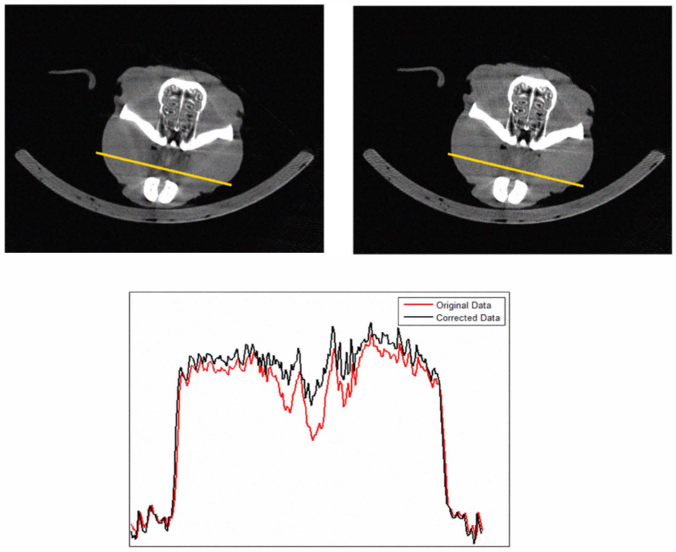

Fig. 4. Images of a rat head acquired using 45 kVp and 1mm Al. Scatter correction was calculated using $5x10^6$ photons (~10 s/prj), 90 projections and 3 Richardson-Lucy iterations with a Gaussian kernel (10px for Compton and Multi, 3px for Rayleigh). Note the reduction of the cupping artifact and of the streaks.

Regarding the improvement in the performance of the MC simulation, the new engine achieves an improvement factor of ~30 compared to the original code. The specific factor is shown in figure 5, for different combinations of the variance reduction techniques.

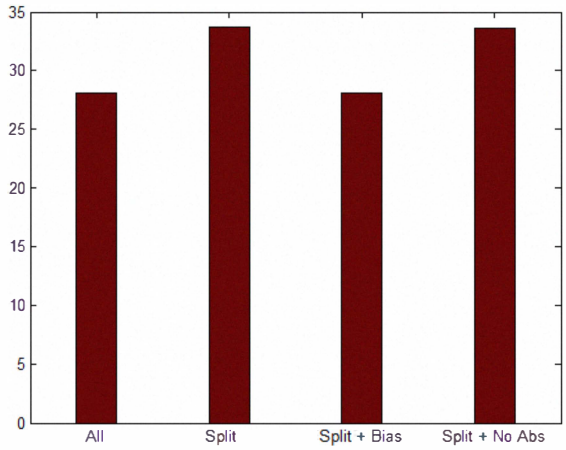

Fig. 5. Ratio of ε achieved by the presented MC engine and the one achieved by the original MC-GPU code. Data consisted of 7 runs of a projection image with 10^8 photons per run.

The splitting and forced detection yielded most of the improvement factor, while the forced interaction degraded the performance of the process when the volume contained air regions. The biasing of the source showed good results for small samples inside large volumes, but otherwise degraded the performance.

V. Discussion and Conclusions

The proposed Monte Carlo approach is able to alleviate the image quality degradation caused by the presence of scattered radiation. The parallel implementation in GP-GPU together with the variance reduction achieved and the separate denoising, allows the estimation of an accurate scatter signal in an acceptable time, opening a door for the implementation of more complex correction processes, specially when high frequency components are present in the scatter signal, such as in compact geometries or when grids are placed to reduce the scattered radiation reaching the detector.

Further work is carried out to improve the performance of the correction process and reduce the variance of the estimated signal, allowing a reduction in the computation time.

Acknowledgment

The authors want to thank Dr Zbijewski and Dr Siewerdsen from Johns Hopkins University for their advice during the development of the MC simulation engine.

References

[1] N. Otsu, "Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on Systems Man and Cybernetics* 9, pp. 62-66 (1979).

[2] A. Badal and A. Badano, "Accelerating Monte Carlo simulations of photon transport in a voxelized geometry using a massively parallel Graphics Processing Unit", *Medical Physics* 36, pp. 4878–4880 (2009).

[3] F.Salvat, *et.al.* "PENELOPE, A Code System for Monte Carlo Simulation of Electron and Photon Transport", *Proceedings of a Workshop/Training Course, OECD/NEA 5-7 November 2001*, (2001).

[4] J. M. Boone, *et al.*, "Molybdenum, rhodium, and tungsten anode spectral models using interpolating polynomials with application to mammography", *Medical Physics* 24, pp. 1863-1874, (1997).

[5] J. H. Siewerdsen, et al., "Spektr: A computational tool for X-ray spectral analysis and imaging system optimization", Medical Physics 31, pp. 3057-3067, (2004).

[6] E. Mainegra-Hing and I. Kawrakow, "Variance reduction techniques for fast Monte Carlo CBCT scatter correction calculations", *Physics in Medicine and Biology* 55, pp. 4495-4507, (2010).

[7] E. Woodcock, *et al.*, "Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry", *Proc. Conf. on Apll. Of Computing Methods to Reactor Problems, Argonne National Laboratories Report ANL-7050*, (1965).

[8] W. Zbijewski and F. J. Beekman, "Efiicient Monte Carlo Based Scatter Artifact Reduction in Cone-Beam Micro-CT", IEEE Transactions on Medical Imaging 25, pp. 817-827, (2006).

[9] J.J. Vaquero, *et al.*, "Assessment of a New High-Performance Small-Animal X-Ray Tomograph", IEEE Transactions on Nuclear Science 55, pp. 898-905, (2008).