UNIVERSITY of MARYLAND
SCHOOL OF MEDICINE

Photon Optimization with GPU and Multi-Core CPU; What are the issues?

Arezoo Modiri, PhD

Outline

- Parallelization
  - CPUs/Clusters/Cloud/GPUs
  - Data management
- Computation-Intensive Applications in Photon Radiotherapy
  - Dose calculation
  - Image registration/reconstruction
  - Robustness analysis
  - Higher-dimensional inverse planning
- Through an Example (4D IMRT Inverse Planning)
  - Hardware configuration
  - Factors impacting process speed

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Disclosure

This work was supported in part by the National Cancer Institute (R01CA169102) and Varian Medical Systems.

Aaron Hagan
Department of Radiation Oncology
University of Maryland, Baltimore

Why is parallelization important?

- Radiotherapy applications

  use large data sets and/or complex numerical algorithms.

  are desired to be solved in a timely fashion.

  are sometimes desired to be solved in minutes or even in (near) real time, such as on-line adaptive radiation therapy (ART).

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Solutions for Speeding Up Processes

- Using devices with higher clock speed (we are hitting a technological limit)
- Using devices supporting parallel processing (multi-core CPUs and GPUs)

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

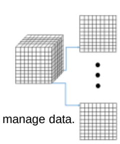
Solutions for Speeding Up Processes

- Using devices with higher clock speed (we are hitting a technological limit)
- Using devices supporting parallel processing (multi-core CPUs and GPUs)

- Managing data for parallel processing
  – The size of the data can be large. Yet, data are usually parallelization-friendly, in that the entire task can be naturally broken down to small operations at pixel/voxel/beamlet/aperture/etc. level.

  – Most works use single-precision float point data type.

  – Down sampling, reducing calculation volume and sparsification can be used to manage data.



Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Solutions for Speeding Up Processes

- Using devices with higher clock speed (we are hitting a technological limit)
- Using devices supporting parallel processing (multi-core CPUs and GPUs)

- Managing data for parallel processing
  - The size of the data can be large. Yet, data are usually parallelization-friendly, in that the entire task can be naturally broken down to small operations at pixel/voxel/beamlet/aperture/etc. level.
  - Most works use single-precision float point data type.
  - Down sampling, reducing calculation volume and sparsification can be used to manage data.

- Intermediate-size data: A GPU solution
- Large-size data: A CPU solution

4

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

GPU-based high-performance computing for radiation therapy, Jia et al., PMB 2014

GPUs versus CPUs

CPUs	# Cores	Clock Speed (GHz)	Maximum memory (GB)	Single- precision performance (TFLOPS)	Double-precision performance (TFLOPS)	Memory bandwidth
Intel Xeon E7 8893 v3	8 (Multi-threading)	3.2/3.5	1540	0.448	0.224	~200-400GB/s
AMD EPYC™ 7601	32	2.2/3.2	2000	0.409	0.204	

GPUs	# Cores	Clock Speed (GHz)	Maximum memory (GB)	Single- precision performance (TFLOPS)	Double-precision performance (TFLOPS)	
Radeon Instinct™ MI25	4096	1.5	16	12.3	6.15	
NVIDIA Tesla P100	3584	1.33-1.48	16	10.6	5.3	~700GB/s

Coprocessors	# Cores	Clock Speed (GHz)	Maximum memory (GB)	Single- precision performance (TFLOPS)	Double-precision performance (TFLOPS)	
Intel Xeon Phi 7290	72	1.5	16	6.92	3.46	~115GB/s

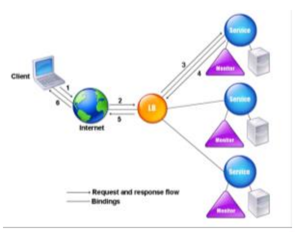
5

http://ark.intel.com/products/84688/
http://images.nvidia.com/content/tesla/pdf/nvidia-tesla-p100-datasheet.pdf
https://instinct.radeon.com/en-us/product/mi/radeon-instinct-mi25/

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Clusters

- Expensive (setup and maintenance)
- Performance dependency on number of users
- Citrix is an example



Citrix Load Balancing Process

https://www.citrix.com.br/glossary/load-balancing.html

6

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Cloud-based Clusters

Outsourcing computation resources to a 3rd party company (Amazon, google, etc.)

- Internet browser enabling a user to view DICOM-RT file.
- Performs computing tasks (registration, segmentation, treatment planning, dose calculation)
- Data base

Need to pay per hour

7

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Courtesy – Lei Xing – Stanford University

Reviews of GPU-based Computation in Radiotherapy

GPU Computing in Medical Physics, Lei Xing et al., Med. Phys. 2011

GPU-based high-performance computing for radiation therapy, Xun Jia et al., PMB 2014

8

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Computationally Intense Radiotherapy Applications

- Dose calculation
- Image registration/reconstruction
- Plan optimization
- Robustness analysis

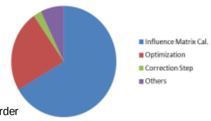
9

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Computationally Intense Radiotherapy Applications

An Example

ECHO (Expedited Constrained Hierarchical Optimization)
Computational time 1 to 4 hours
Express the clinical criteria as hard constraints
Prioritize the clinical objectives and optimize them in order



Depending on data size, registration may take less or more time compared to plan optimization (from our group's study).

Courtesy - Masoud Zarepisheh – Memorial Sloan Kettering

10

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Computationally Intense Radiotherapy Applications

- Dose calculation techniques
  - Pencil-beam
  - Superposition/convolution
  - Monte Carlo
  - Treatment planning systems

11

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Monte Carlo Dose Calculation

- GPU-based MC project at UT Southwestern

2009	2011	2012	2014	2015	2016
gDPM	gCTD gMCDRR	gPMC	goMC gBMC	goCMC (Carbon-ion therapy)	goMicroMC (brachytherapy)

g: GPU
go: GPU OpenCL

- Particle types: photon, electron, proton, carbon ion, free radical…
- Clinical applications: external beam therapy, brachytherapy
- Energy ranges: eV → keV → MeV → GeV
- Spatial scales: nm (DNA level) → m (human level)

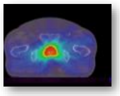
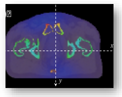
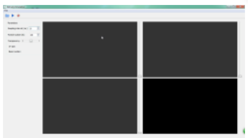
Courtesy – Xun Jia – UT Southwestern

12

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Clinical Application of MC Dose Calculation

- Dose calculation
- Including imaging dose in optimization
- Treatment monitoring/verification

Courtesy – Xun Jia – UT Southwestern

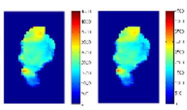
Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Dose Calculation
Efficient MC Implementation

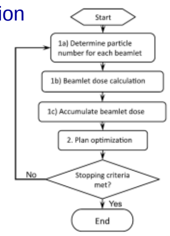
- Instead of calculating dose deposition matrices for all beamlets using MC prior to optimization, dose calculation is performed inside optimization loop but number of particles for MC is optimized.

Offline vs Online

CT was resampled to 128 × 128 × 86 voxels.



Optimized fluence map with 1×10^6 / beamlet and 1×10^4 / beamlet/iteration (4x acceleration)

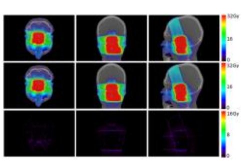


- The computation time including both MC dose calculations and plan optimizations was reduced by a factor of 4.4, from 494 to 113 s, using only one GPU card.

A new Monte Carlo-based treatment plan optimization approach for intensity modulated radiation therapy, Li et al., PMB 2015

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Dose Calculation
Hardware-Independent Implementation



Dose calculated by OpenCL and CUDA versions of the code (first and second rows) and their comparison (last row).

- In terms of efficiency, goMC was ~4–16% slower than gDPM when running on the same NVidia TITAN card for all the cases tested, due to both the different electron transport models and the different development environments.
- AMD GPU cards are faster for OpenCL applications.

A GPU OpenCL based cross-platform Monte Carlo dose calculation engine (goMC), Tian et al., PMB 2015

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Dose Calculation
Hardware-Independent Implementation

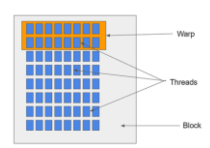
It is quite straightforward to port an existing CPU algorithm onto GPU and achieve acceleration to a certain degree. It is, nonetheless, quite challenging to write a high-efficiency code that fully exploit the potential of a GPU.

16

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

MC Imaging Photon-Electron Simulation

- GPU implementation of the photon transport mechanism of EGSnrc
- Speedups of 20 to 40 times for 64^3 to 256^3 voxels were observed



Thread divergence control

17

A GPU implementation of EGSnrc's Monte Carlo photon transport for imaging applications, Lippuner et al., PMB 2011
GPUMCD: a new GPU-oriented Monte Carlo dose calculation platform, Hissoini et al., Med. Phys. 2011

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

MC Particle Transport Simulation

- Using parametrized geometry, the computational time ranged in 1.75–2.03 times of the voxelized geometry for coupled photon/electron transport depending on the voxel dimension of the auxiliary index array, and in 0.69–1.23 times for photon only transport.

Table 2. Comparison of transport simulation time without using the auxiliary index array.

Case	Parameterized geometry (μ s/history) Global memory	Texture memory	Shared memory	Voxelized geometry (s/billion)	α_1
Brachytherapy photon transport	5.546	3.792	2.121	0.761	2.79
Coupled electron–photon transport	0.292	0.234	0.198	0.060	3.29

Table 3. Comparison of transport simulation time with the auxiliary index array and shared memory.

Case	Parameterized geometry (μ s/history)	Voxelized geometry (μ s/history)	α_2
Brachytherapy photon transport	0.614	0.761	0.81
Coupled electron–photon transport	0.105	0.060	1.75

Algorithmic solutions

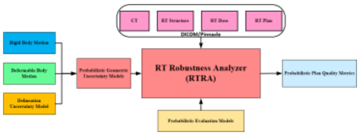
18

Modeling parametrized geometry in GPU-based Monte Carlo particle transport simulation for radiotherapy, Chi et al., PMB 2016

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Robustness Analysis

- Determining the geometric uncertainties effects on the quality of the RT plans is computationally expensive and demands high performance computation capabilities.
- An in-house radiation therapy robustness analyzer (RTRA)
  - Simulates uncertainties due to:
    - Daily patient setup error
    - Deformable body motion
    - Delineation uncertainties



19

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Courtesy – Hamid Nourzadeh – University of Virginia
Clinical adequacy assessment of autocontours for prostate IMRT with meaningful endpoints, Nourzadeh et al., Med. Phys. 2017

Higher-Dimensional Inverse Planning

- IMRT *Ziegenhein et al., PMB 2013; Men et al., PMB 2009*
- VMAT *Tian et al., Med. Phys. 2015; Chin et al., Med. Phys. 2013*
- 4D *Nohadani et al., PMB 2010; Suh et al., PMB 2009*
- 4π *Chiu et al., Med. Phys. 2016; Dong et al., RED 2012*
  TORUS *Locke et al., Med. Phys. 2017*

Increased degrees of freedom in planning

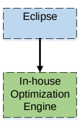
Figure Courtesy – Karl Bush – Stanford University

20

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

An Example: 4D IMRT Inverse Planning

The pipeline for our work consisted of
(i) creating treatment plans for each phase in Eclipse 13.6 TPS,
(ii) exporting dose-deposition matrices for all (tens of thousands) apertures,
(iii) optimizing aperture MU weights using GPU-based in-house optimization.



For 4D dose summation, we used a GPU-enabled deformable image package (Elastix).

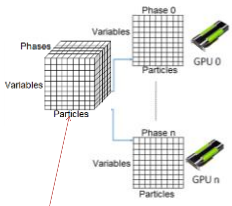
21

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Hagan et al., University of Maryland

An Example: 4D IMRT Inverse Planning

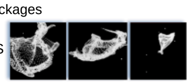
Parallelized over phases



Search agents in particle swarm optimization

Deformable image registration:

- Open source packages
- Commercial TPS
- In-house tools



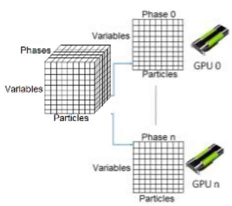
22

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

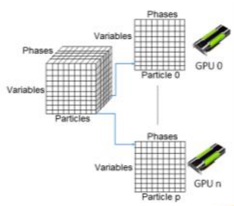
Hagan et al., University of Maryland

An Example: 4D IMRT Inverse Planning

Parallelized over phases



Parallelized over particles

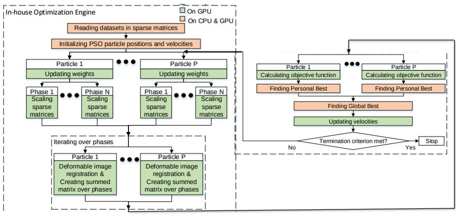


22

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Hagan et al., University of Maryland

An Example: 4D IMRT Inverse Planning



23

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Hagan et al., University of Maryland

An Example: 4D IMRT Inverse Planning

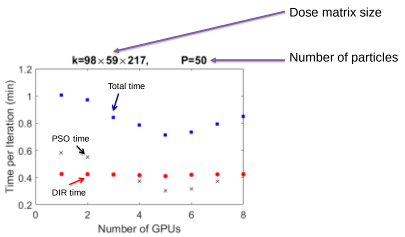
Our implementation was hardware dependent.

12GB of memory available per GPU card
Non-uniform memory access (NUMA) structure
8-core CPUs , 256GB RAM

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Hagan et al., University of Maryland

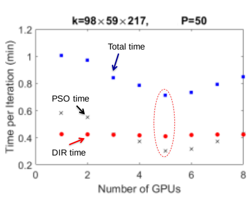
An Example: 4D IMRT Inverse Planning



Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Hagan et al., University of Maryland

An Example: 4D IMRT Inverse Planning



We generally use 25-30 iterations.

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Hagan et al., University of Maryland

An Example: 4D IMRT Inverse Planning

- More details on this study will be presented at

Thursday, Session# TH-CD-205-4

GPU-accelerated Higher-Dimensional Inverse Planning, ***Hagan et al., University of Maryland***

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Other Applications

- Real-time Monte Carlo based Treatment Dose Reconstruction and Monitoring
- DVH-guided IMRT and VMAT auto- and adaptive-planning
  - Algorithms for micro- (small operations in parallel) and macro- (large operations) parallelization being designed
- Biological endpoint calculation using Monte Carlo
- Radiomics and artificial intelligence
  - Problematic lymph node identification
  - Organ-at-risk labeling given contours

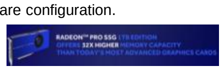
Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Courtesy – Troy Long – UT Southwestern

Conclusion

GPU implementation has enabled various radiotherapy applications being processed in minutes or even seconds.

Data size is an important factor in choosing hardware configuration.



Optimal number of GPUs is not necessarily equal to maximum number of GPUs available.

The implementation technique and process time are hardware dependent.

The choice and design of algorithms are important in parallelization and avoidance of thread divergence.

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

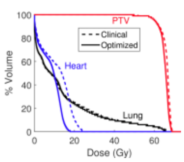
Thank you.

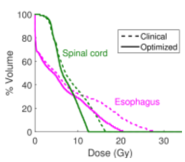
Questions?

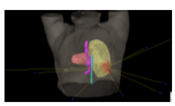
Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Backup Slides

An Example: 4D IMRT Inverse Planning







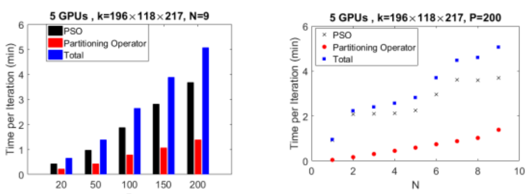
PTV : 144 cc
Target motion: 1.5 cm

Hagan et al., University of Maryland, NIH (R01CA169102) and Varian Medical Systems.

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

- To conserve GPU memory, we used the Compressed Column Row sparsification (10:1 compression ratio). An open-source deformable image registration (Elastix), was employed for dose summation. To avoid deforming tens of thousands of dose matrices, we applied deformation vector fields, calculated prior to optimization, to summed dose matrices inside iteration loop. For evaluation, several 4D-IMRT planning tests were performed on patient data, considering 10 phases, 9 beams, 166 apertures (14940 variables).

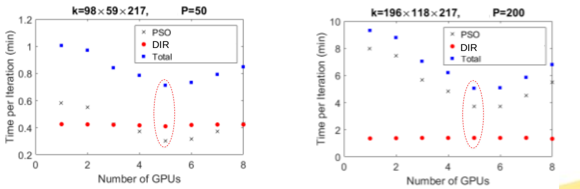
- A typical 10 phase, 200 particle study would equate to 2000 DIR operations in parallel. For a typical patient with each dose matrix being 18.7 MB in size, this would equate to 37.4 GB of dedicated GPU memory that would need to be allocated by elastix.



Process time increases both with number of particles and number of respiratory phases.

Aaron Hagan
Department of Radiation Oncology
University of Maryland, Baltimore

An Example: 4D IMRT Inverse Planning



The optimal number of GPUs (five, in this study) is directly related to the hardware specifications of employed GPU cards.

Modiri et al., University of Maryland, NIH (R01CA169102) and Varian Medical Systems.

Aaron Hagan
Department of Radiation Oncology
University of Maryland, Baltimore

METHODS

Our implementation is distinct from existing 4D planning applications in commercial TPSs because

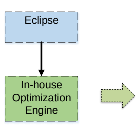
(i) it is not based on internal target volume generation,
(ii) it optimizes across phases and not for each phase, individually,
(iii) particle swarm optimization is used to solve an inverse plan optimization consisted of dose-volume-based objective function, and
(iv) aperture MU weights are optimized not fluence.

Aaron Hagan
Department of Radiation Oncology
University of Maryland, Baltimore

METHODS

The pipeline for our work consisted of

(i) creating treatment plans for each phase in Eclipse 13.6 TPS,
(ii) exporting dose-deposition matrices for all (tens of thousands) apertures,
(iii) optimizing aperture MU weights using GPU-based PSO, implemented in-house.

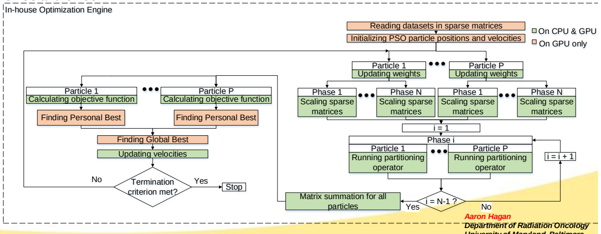


tens of thousands of variables (e.g., in our case study, we had 9 beams × 166 apertures per beam × 10 sampled respiratory phases = 14940 variables).

Aaron Hagan
Department of Radiation Oncology
University of Maryland, Baltimore

METHODS

For 4D dose summation, we used a deformable image package (Elastix).
Due to GPU memory limitations, we needed to use the data in chunks and spread the tasks between GPUs and CPUs.



Aaron Hagan
Department of Radiation Oncology
University of Maryland, Baltimore

GPU versus CPU

SPECIFICATIONS

GPU Architecture	**NVIDIA Pascal**
NVIDIA CUDA® Cores	**3584**
Double-Precision Performance	**5.3 TeraFLOPS**
Single-Precision Performance	**10.6 TeraFLOPS**
Half-Precision Performance	**21.2 TeraFLOPS**
GPU Memory	**16 GB CoWoS HBM2**
Memory Bandwidth	**732 GB/s**
Interconnect	**NVIDIA NVLink**
Max Power Consumption	**300 W**
ECC	**Native support with no capacity or performance overhead**
Thermal Solution	**Passive**
Form Factor	**SXM2**
Compute APIs	**NVIDIA CUDA, DirectCompute, OpenCL™, OpenACC**

TeraFLOPS measurements with NVIDIA GPU Boost™ technology

- High computational power, small size, low maintenance cost
- Single instruction multiple data

Table 2. Comparing high-end products of CPU and GPU.

	# Cores	Clock speed (GHz)	Memory size (max) (GB)	Efficiency (GFLOPS W^{-1})	Memory bandwidth (GB s^{-1})	Single precision performance (GFLOPS)	Double precision performance (GFLOPS)
Intel Xeon E5-2687 W	8	3.1/3.8	750	1.4	51.2	486	243
NVIDIA GPU K20	2688	0.732	6	16.80	250	3520	1170

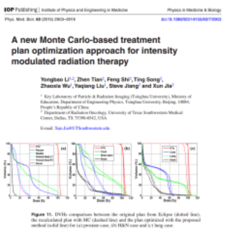
http://images.nvidia.com/content/tesla/pdf/nvidia-tesla-p100-datasheet.pdf

Outline

- Why is parallelization important?
  - Dose calculation
  - Inverse plan optimization
  - Offline processes versus online/real-time/on-the-fly processes
  - Fluence optimization versus aperture weight optimization
  - Dealing with large number of variables in IMRT, ARC treatment planning or in 4D and non-uniform-fractionation treatment planning
- What is the impact of data size?
  - GPU memory
  - Downsampling versus keeping original data size
  - Sparcification
  - Computationally expensive processes; e.g., deformable image registration
  - Staying Compatible with existing treatment planning systems
- Solver and algorithm matter.
  - Dealing with non-convexity: DVH-based goals, BED-based goals
  - Using global versus Local optimization

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Dose Calculation

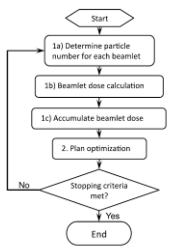


Intensity-modulated radiation treatment (IMRT) plan optimization needs beamlet dose distributions. Pencil-beam or superposition/convolution type algorithms are typically used because of their high computational speed. However, inaccurate beamlet dose distributions may mislead the optimization process and hinder the resulting plan quality. To solve this problem, the Monte Carlo (MC) simulation method has been used to compute all beamlet doses prior to the optimization step. The conventional approach samples the same number of particles from each beamlet. Yet this is not the optimal use of MC in this problem. In fact, there are beamlets that have very small intensities after solving the plan optimization problem. For those beamlets, it may be possible to use fewer particles in dose calculations to increase efficiency. Based on this idea, we have developed a new MC-based IMRT plan optimization framework that iteratively performs MC dose calculation and plan optimization. At each dose calculation step, the particle numbers for beamlets were adjusted based on the beamlet intensities obtained through solving the plan optimization problem in the last iteration step. We modified a GPU-based MC dose engine to allow simultaneous computations of a large number of beamlet doses. To test the accuracy of our modified dose engine, we compared the dose from a broad beam and the summed beamlet doses in this beam in an inhomogeneous phantom. Agreement within 1% for the maximum difference and 0.55% for the average difference was observed. We then validated the proposed MC-based optimization schemes in one lung IMRT case. It was found that conventional scheme required 106 particles from each beamlet to achieve an optimization result that was 3% difference in fluence map and 1% difference in dose from the ground truth. In contrast, the proposed scheme achieved the same level of accuracy with on average 1.2 × 105 particles per beamlet. Correspondingly, the computation time including both MC dose calculations and plan optimizations was reduced by a factor of 4.4, from 494 to 113 s, using only one GPU card.

Courtesy – Yongbao Li et al. – UT Southwestern

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

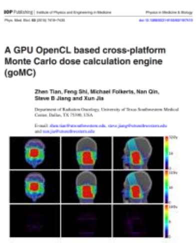
Dose Calculation



Courtesy – Yongbao Li et al. – UT Southwestern

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

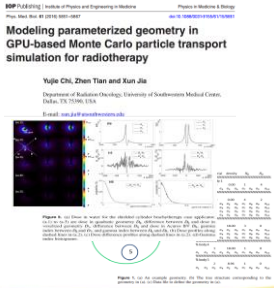
Dose Calculation



Monte Carlo (MC) simulation has been recognized as the most accurate dose calculation method for radiotherapy. However, the extremely long computation time impedes its clinical application. Recently, a lot of effort has been made to realize fast MC dose calculation on graphic processing units (GPUs). However, most of the GPU-based MC dose engines have been developed under NVidia's CUDA environment. This limits the code portability to other platforms, hindering the introduction of GPU-based MC simulations to clinical practice. The objective of this paper is to develop a GPU OpenCL based cross-platform MC dose engine named goMC with coupled photon–electron simulation for external photon and electron radiotherapy in the MeV energy range. Compared to our previously developed GPU-based MC code named gDPM (Jia et al 2012 Phys. Med. Biol. 57 7783–97), goMC has two major differences. First, it was developed under the OpenCL environment for high code portability and hence could be run not only on different GPU cards but also on CPU platforms. Second, we adopted the electron transport model used in EGSnrc MC package and PENELOPE's random hinge method in our new dose engine, instead of the dose planning method employed in gDPM. Dose distributions were calculated for a 15 MeV electron beam and a 6 MV photon beam in a homogenous water phantom, a water-bone-lung-water slab phantom and a half-slab phantom. Satisfactory agreement between the two MC dose engines goMC and gDPM was observed in all cases. The average dose differences in the regions that received a dose higher than 10% of the maximum dose were 0.48–0.53% for the electron beam cases and 0.15–0.17% for the photon beam cases. In terms of efficiency, goMC was ~4–16% slower than gDPM when running on the same NVidia TITAN card for all the cases we tested, due to both the different electron transport models and the different development environments. The code portability of our new dose engine goMC was validated by successfully running it on a variety of different computing devices including an NVidia GPU card, two AMD GPU cards and an Intel CPU processor. Computational efficiency among these platforms was compared.

A GPU OpenCL based cross-platform Monte Carlo dose calculation engine (goMC), Zhen Tian et al., 2015

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore


IOP Publishing | Institute of Physics and Engineering in Medicine   Physics in Medicine & Biology

Modeling parameterized geometry in GPU-based Monte Carlo particle transport simulation for radiotherapy

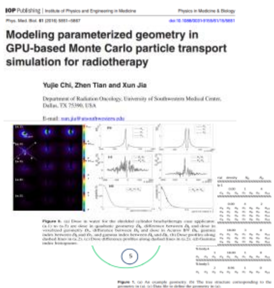
Yujie Chi, Zhen Tian and Xun Jia

Department of Radiation Oncology, University of Southwestern Medical Center, Dallas, TX 75390, USA

Monte Carlo (MC) particle transport simulation on a graphics-processing unit (GPU) platform has been extensively studied recently due to the efficiency advantage achieved via massive parallelization. Almost all of the existing GPU-based MC packages were developed for voxelized geometry. This limited application scope of these packages. The purpose of this paper is to develop a module to model parametric geometry and integrate it in GPUbased MC simulations. In our module, each continuous region was defined by its bounding surfaces that were parameterized by quadratic functions. Particle navigation functions in this geometry were developed. The module was incorporated to two previously developed GPU-based MC packages and was tested in two example problems: (1) low energy photon transport simulation in a brachytherapy case with a shielded cylinder applicator and (2) MeV coupled photon/electron transport simulation in a phantom containing several inserts of different shapes. In both cases, the calculated dose distributions agreed well with those calculated in the corresponding voxelized geometry. The averaged dose differences were 1.03% and 0.29%, respectively. We also used the developed package to perform simulations of a Varian VS 2000 brachytherapy source and generated a phase-space file. The computation time under the parameterized geometry depended on the memory location storing the geometry data. When the data was stored in GPU's shared memory, the highest computational speed was achieved. Incorporation of parameterized geometry yielded a computation time that was ~3 times of that in the corresponding voxelized geometry. We also developed a strategy to use an auxiliary index array to reduce frequency of geometry calculations and hence improve efficiency. With this strategy, the computational time ranged in 1.75–2.03 times of the voxelized geometry for coupled photon/electron transport depending on the voxel dimension of the auxiliary index array, and in 0.69–1.23 times for photon only transport.

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Modeling parametrized geometry in GPU-based Monte Carlo particle transport simulation for radiotherapy, Yujie Chi et al., 2016


IOP Publishing | Institute of Physics and Engineering in Medicine   Physics in Medicine & Biology

Modeling parameterized geometry in GPU-based Monte Carlo particle transport simulation for radiotherapy

Yujie Chi, Zhen Tian and Xun Jia

Department of Radiation Oncology, University of Southwestern Medical Center, Dallas, TX 75390, USA

Monte Carlo (MC) particle transport simulation on a graphics-processing unit (GPU) platform has been extensively studied recently due to the efficiency advantage achieved via massive parallelization. Almost all of the existing GPU-based MC packages were developed for voxelized geometry. This limited application scope of these packages. The purpose of this paper is to develop a module to model parametric geometry and integrate it in GPUbased MC simulations. In our module, each continuous region was defined by its bounding surfaces that were parameterized by quadratic functions. Particle navigation functions in this geometry were developed. The module was incorporated to two previously developed GPU-based MC packages and was tested in two example problems: (1) low energy photon transport simulation in a brachytherapy case with a shielded cylinder applicator and (2) MeV coupled photon/electron transport simulation in a phantom containing several inserts of different shapes. In both cases, the calculated dose distributions agreed well with those calculated in the corresponding voxelized geometry. The averaged dose differences were 1.03% and 0.29%, respectively. We also used the developed package to perform simulations of a Varian VS 2000 brachytherapy source and generated a phase-space file. The computation time under the parameterized geometry depended on the memory location storing the geometry data. When the data was stored in GPU's shared memory, the highest computational speed was achieved. Incorporation of parameterized geometry yielded a computation time that was ~3 times of that in the corresponding voxelized geometry. We also developed a strategy to use an auxiliary index array to reduce frequency of geometry calculations and hence improve efficiency. With this strategy, the computational time ranged in 1.75–2.03 times of the voxelized geometry for coupled photon/electron transport depending on the voxel dimension of the auxiliary index array, and in 0.69–1.23 times for photon only transport.

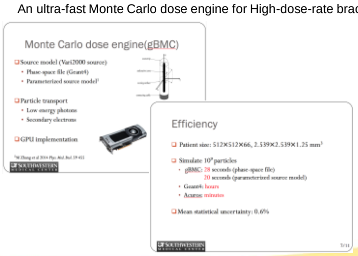
Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Modeling parametrized geometry in GPU-based Monte Carlo particle transport simulation for radiotherapy, Yujie Chi et al., 2016

Dose Calculation

Medical Physics
The International Journal of Medical Physics Research and Practice

An ultra-fast Monte Carlo dose engine for High-dose-rate brachytherapy

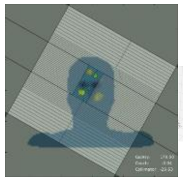


A phase space file was generated for the Varian VS2000 Ir-192 source. In a water phantom, the calculated radial dose function was within 0.6% of the TG43 calculations for radial distances from 1 cm to 20 cm. The anisotropy functions were within 1% for radial distances from 1 cm to 20 cm except for polar angles larger than 173°. Local point-dose differences were within 2%. In a Mammosite breast cancer case with 22 dwell locations, gBMC and Geant4 isodose lines compared well. The computation time was about 28 seconds using the phase-space file source and 20 seconds using the parameterized source to simulate 1 billion particles, yielding less than 1% statistical uncertainty.

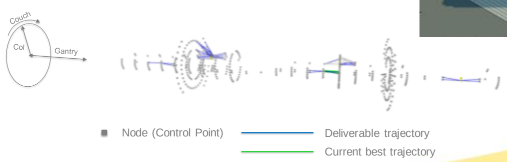
Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore

Courtesy – Zhen Tian et al. – UT Southwestern

MLC Trajectory Optimization



- Graph optimization to generate efficient dynamic trajectories for delivery while maximizing the angular flux through all PTV voxels.
- 3D dose optimization is performed for trajectories using a commercial TPS progressive resolution optimizer.



Courtesy – Karl Bush – Stanford

Arezoo Modiri
amodiri@som.umaryland.edu
Department of Radiation Oncology
University of Maryland, Baltimore