Developments in Monte Carlo Methods for Medical Imaging

Introduction to Monte Carlo simulations in medical imaging and state-of-the-art computational acceleration methods

Andreu Badal, PhD

DIDSR, OSEL, CDRH
U.S. Food and Drug Administration

AAPM 2014 innovation
56th Annual Meeting & Exhibition • July 20–24, 2014 • Austin, TX

Background

- X-ray imaging is an essential medical tool but is not exempt of ionizing radiation risks
  - Total number of imaging exams increases every year and radiation from different exposures adds up
  - A fraction of patients will develop radiation induced diseases (risk/benefit analysis required)
  - Important to keep the imaging radiation dose "As Low As Reasonably Achievable" (dose/image-quality tradeoff, different situation than radiotherapy)

2

- Imaging system design and optimization is an active field of research and innovation
- Ethical and public health reasons strongly limit the non-clinical exposure of humans:
  - Bench testing with objects reproducing anatomical structures (phantoms) is essential
  - Computer simulations are a valuable tool to obtain more information on system performance
    - Advantages: inexpensive, "fast", repeatable, limited error sources, flexibility in system configuration, measurement of any quantity of interest, ideal detectors
    - Disadvantages: model approximations and omissions limit realism, not warrantied to reproduce reality without thorough validation, programming and user error

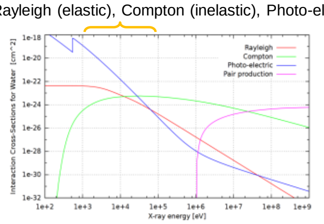
3

Simulating x-ray imaging systems

- Variable Components to model:
  - X-ray source:
    - Energy spectra, filtration
    - Focal spot size, shape
    - Heel effect, off-focus radiation
  - Object geometry (anatomy)
    - Quadrics, voxels, triangle-meshes, NURBS (*more in following talk!*)
  - X-ray detector:
    - Image formation model
    - Pixel correlation (MTF)
    - Electronic noise

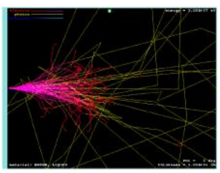
4

- Known component to model:
  - X-ray/atom interaction physics
    - Interaction cross-sections (probabilities) can be obtained from analytical equations or experimental tables
    - Any material of known atomic composition can be modeled with Independent Atom Approximation
  - Interactions of interest in 1-100 keV range:
    - Rayleigh (elastic), Compton (inelastic), Photo-electric



5

- Not having electric charge, x-ray photons propagate in straight lines and interact sparsely with matter
- Secondary electrons and fluorescence photons emitted after Compton and photo-electric interactions
- Electron transport has no impact on projection images and minimal effect on dosimetry at keV energies:
  - Electron range in water at 100 keV =143 μm, at 10 keV = 2.5 μm



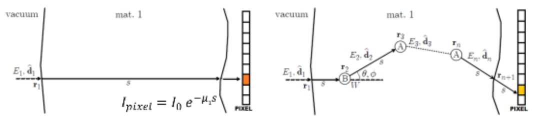
- High-energy particle tracks: 10 MeV electrons (pink) inside 10 cm water slab (Bremsstrahlung photons in yellow)

6

- Main algorithms used to simulate medical images:
  - Analytic ray-tracing
    - Image estimated by computing the probability that x-rays traveling in a straight line from the focal spot to each pixel will not interact with the object (primary image, no scatter)
    - Interaction probability described by exponential attenuation equation with known material attenuation coefficients (μ_i)

$$Prob. \propto e^{-(\mu_1 s_1 + \mu_2 s_2 + \ldots)}$$

    - A geometric algorithm (ray-casting) estimates the path length travelled across each material (s_i) in the flight to each pixel
    - Simulation time directly proportional to the number of pixels in the image (and, optionally, to the number of energy levels in the input energy spectrum)



- Monte Carlo method:
  - Image estimated by averaging the contribution to each pixel from a large number of individual x-ray tracks
  - X-ray tracks simulated as random walks: random numbers used to sample the initial position, direction and the distance to and kind of the following interaction
  - Differential interaction cross sections used to sample the energy loss and angular deflection following each interaction
  - Simulation time proportional to the number of x rays, which depends on field size and exposure (indirectly to pixel size)
  - Simulated images might display realistic quantum noise statistics if the simulation sampled as many x rays as real acquisitions

- Multiple Monte Carlo codes freely available: **EGS**, **GEANT4**, **MCNP**, **PENELOPE**…
  - Codes implement different physics and geometry models, have different strengths and weaknesses
  - Consistency among the results from different codes increases reliability of results (*see last talk!*)
- Monte Carlo simulations may require long computing times and large computing resources:
  - Extremely large number of x-ray tracks to simulate
  - Quantities of interest estimated in small volumes (e.g., dose in voxels, energy fluence in pixels)
  - High resolution phantoms may require lots of memory

- Example Monte Carlo algorithm: **PENELOPE**

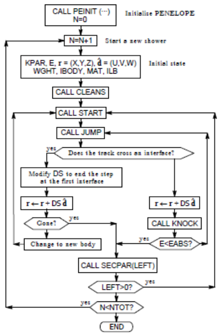
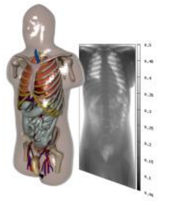


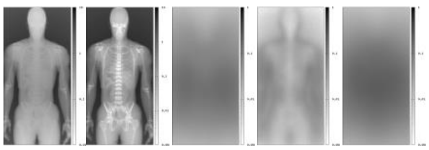
Figure 6.1: Flow diagram of the main program for simulating electron-photon showers with PENELOPE.



F. Salvat, J. M. Fernandez-Varea and J. Sempau, PENELOPE-2006: A Code System for Monte Carlo Simulation of Electron and Photon Transport, OECD/NEA Workshop Proceeding, Barcelona 4-7 July 2006, ISBN 92-64-02301-1 (http://www.oecd-nea.org/science/pubs/2006/nea6222-penelope.pdf)

10

- Example simulated radiography:
  - 10^{10} primary x rays, 90 kVp
  - Anthropomorphic male phantom from the Virtual Family
    - » 305 x 155 x 930 voxels, 2 mm voxels, 8 materials
  - Ideal detector [eV/cm^2]:
    - » 750x1500, 1 mm pixels



» PenEasy_Imaging code based on PENELOPE available at: http://code.google.com/p/peneasy-imaging/

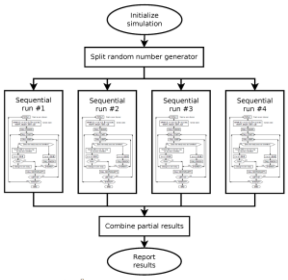
11

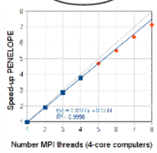
Computational methods to speed up the simulations

- Common approaches used to speed up simulations:
  - **Simplify** the implemented physical models to improve the performance at the expense of accuracy or general applicability (e.g., not modeling electrons, Klein-Nishina)
  - Implement **variance reduction** techniques to optimize the information obtained from each x-ray track without affecting the estimated mean values (*more in next talk!*)
  - Use high-performance computing techniques to execute the same algorithm in less effective time:
    - **Optimizing compilation** parameters, memory access patterns and instruction scheduling (vectorization) for a specific processor
    - Using **parallel execution** in multiple processors or accelerators

12

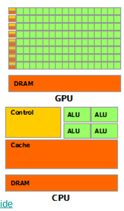
- Parallel execution in multiple CPUs (Central Processing Units):
  - Use multiple CPU cores in one or many computers (openMP, MPI - Message Passing Interface…)
  - Monte Carlo simulations are *embarrassingly parallel*: each x-ray track is independent from other
  - Nearly linear speed-up with increasing number of processors (minimal communication overhead)

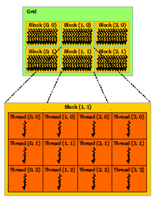




- Parallel execution in GPUs:
  - A GPU (Graphics Processing Unit) is a massively multi-core co-processor designed to process thousands of concurrent threads
  - Memory handling, cache and execution control simplified to maximize computational power
  - NVIDIA's GPU architecture (CUDA):

- Example state-of-the-art GPU:
  NVIDIA GeForce GTX 780
  12 microprocessors with 192 computing cores: **2304 cores**
  Clock speed: **0.9 GHz**
  Video memory**: 3.0 GByte**





http://docs.nvidia.com/cuda/cuda-c-programming-guide

- Multiple GPU-accelerated Monte Carlo codes for diagnostic imaging simulation available: **MC-GPU**, **GPUMCD**, **gCTD**, **GMC**…
  - Most codes are based on traditional, reliable codes
    - More information: G Pratx and L Xing, GPU computing in medical physics: A review, Med Phys 38, p. 2685 (2011)

- Other computational accelerators in development operate similarly to GPUs (massively parallel):
  - Intel Xeon Phi
  - FPGA

- Example x ray Monte Carlo trans. algorithm for GPU:

MC-GPU

- 10^{11} x-rays = 65000 blocks of 64 threads, 25000 tracks per thread
- No electrons and no secondary stack
- Virtual interactions to optimize voxel tracing

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)
http://code.google.com/p/mcgpu/

16

- Example MC-GPU simulation cone beam CT:
  - 1 mm voxels; 540 proj; $2{\cdot}10^{10}$ x-rays/proj; 65 keV
  - Total simulation time: 4.2 days, 90 min/proj/GPU

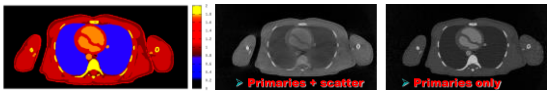


Figure 1. Voxelized phantom (left), and reconstructed CT images with (center) and without scatter (right).

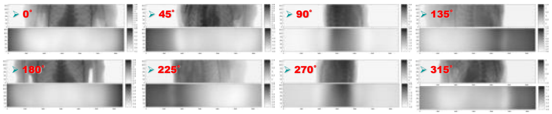


Figure 2. Eight of 540 GPU-simulated radiographic projections around the phantom (above, primary+scatter; below, scatter only). The gray scales have units of [log10] eV/cm^2 per history.

A. Badal, I. Kyprianou, D. Sharma and A. Badano, Fast cardiac CT simulation using a GPU-accelerated Monte Carlo code, SPIE Medical Imaging 2010, 762231-9 (2010)

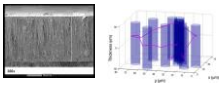
17

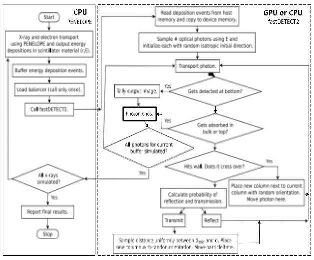
- Hybrid computing: parallel execution in CPU+GPU
  - Heterogeneous computers are becoming prevalent
  - Different parts of an algorithm may scale better in different computational architectures
  - Example hybrid Monte Carlo algorithm for the simulation of indirect scintillator detectors: **hybridMANTIS***
    - X-rays and electrons simulated by PENELOPE
    - Optical photons inside the scintillator modeled by fastDETECT2
    - On-the-fly geometry to reproduce the CsI columnar structure
    - Simulation outputs: Point Spread Function, Modulation Transfer Function, Pulse Height Spectrum, Swank factor.
    - Parallel CPU/GPU execution allows efficient, concurrent x-ray/optical transport.

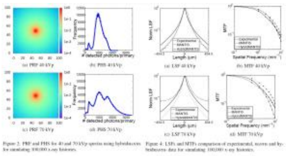
* D. Sharma, A. Badal and A. Badano, hybridMANTIS: a CPU–GPU Monte Carlo method for modeling indirect x-ray detectors with columnar scintillators, Phys Med Biol 57, p. 2357 (2012)

18

- **hybridMANTIS** algorithm to simulate indirect scintillator detectors. Optical photon transport in GPU runs simultaneously with x-ray, electron transport in CPU.







	X-ray transport time PENELOPE	Optical transport time fastDETECT2	Total time	Speed x-ray per second
CPU only	00:20:56	14:00:14	14:21:10	18
Serial CPU+GPU	00:20:34	00:24:06	00:44:40	375
HybridMANTIS	–	–	00:26:43	627

Question 1

One of the main limitations of ray-tracing algorithms used in medical imaging is that they…

39%	1.	Require too much RAM memory and other expensive computational resources
6%	2.	Do not intrinsically model scattered radiation
22%	3.	Can be used only to model simple geometric objects
22%	4.	Simulate lower resolution images than clinical systems
11%	5.	Require execution times that are too large for clinical use

Answer question 1

- **Answer: 2** – Ray tracing algorithms only model the trajectories of x-rays that do not interact with the objects, scattering events are neglected.
- Computational resources, speed, geometric complexity and resolution are typically not a major limitation for ray-tracing algorithms.
  - https://en.wikipedia.org/wiki/Ray_tracing_(physics)

Question 2

GPUs are able to significantly speed up Monte Carlo simulation codes because they…

15% 1. Have much faster access to the main video memory than the CPU to the RAM memory
30% 2. Are programmer-friendly and much easier to program than CPUs
20% 3. Use the Many Integrated Core Architecture with 61 computing cores that work in parallel and communicate at fast speed
35% 4. Have thousands of computing cores that can work in parallel, although some groups of cores have to execute the same instruction each time interval.

22

Answer question 2

- **Answer: 4** – GPUs have multiple SIMD (Single Instruction Multiple Data) microprocessors that contain hundreds of separate computing cores.
  - Example: NVIDIA GeForce GTX 780: 2304 cores
    - https://developer.nvidia.com/cuda-zone
- Option c describes another useful co-processors accelerator: the **Intel Xeon Phi**
    - http://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-coprocessor-overview.html

23

Question 3

Monte Carlo simulation algorithms can reproduce many physical effects that are observed in real CT scans, but one of the following effects is not directly modeled by Monte Carlo transport:

36% 1. Motion blurring
18% 2. Beam hardening
14% 3. Quantum noise
18% 4. Multiple scattering
14% 5. Metal artifacts

24

Answer question 3

- **Answer: 1** – Motion blurring caused by heart beat and other time-dependent events does not affect the individual x-ray movement and therefore is not directly modeled by Monte Carlo algorithms. However, multiple simulations with slightly different geometries could be combined to reproduce blurring.
- Metal artifacts such as streaks emerge after the reconstruction of simulated projections because Monte Carlo correctly models photon starvation

A. Badal, I. Kyprianou, D. Sharma and A. Badano, Fast cardiac CT simulation using a GPU-accelerated Monte Carlo code, SPIE Medical Imaging 2010, 762231-9 (2010)

25

Developments in Monte Carlo Methods for Medical Imaging

Introduction to Monte Carlo simulations in medical imaging and state-of-the-art computational acceleration methods

Andreu Badal, PhD

DIDSR, OSEL, CDRH
U.S. Food and Drug Administration

AAPM 2014 innovation

innovation 56th Annual Meeting & Exhibition • July 20–24, 2014 • Austin, TX