Accelerated Monte Carlo Simulation

of the Xray Radiography

Geant 4 revisited for GPU

work in progress

Alain Bonissent
CPPM Marseille

X ray source

detector

Typical radiography. Can be humans, mechanical parts …

CT scanner : many images at various angles => 3D volume

Optimize instruments design

Debug image processing software and procedures

Evaluate radiation damage to subject

Generation of Events and Tracks

Most popular tool for Simulations of medical imaging

Developed at CERN, for particle physics, worldwide contributions
> 10^6 lines of code
All interactions of particles with matter, tuned on experimental data

C++, many many classes
Efficient internal representation of geometric structures
and propagation in non-uniform medium where am I and how far

Alternative approach for electromagnetic processes :
EGS (SLAC) Fortran, still works and well maintained

Detectors have more and more pixels (million is current);
Tomography requires many images (typical 1000)
Subject moves (respiration and heart beat)

Detailed models of animals, high granularity.

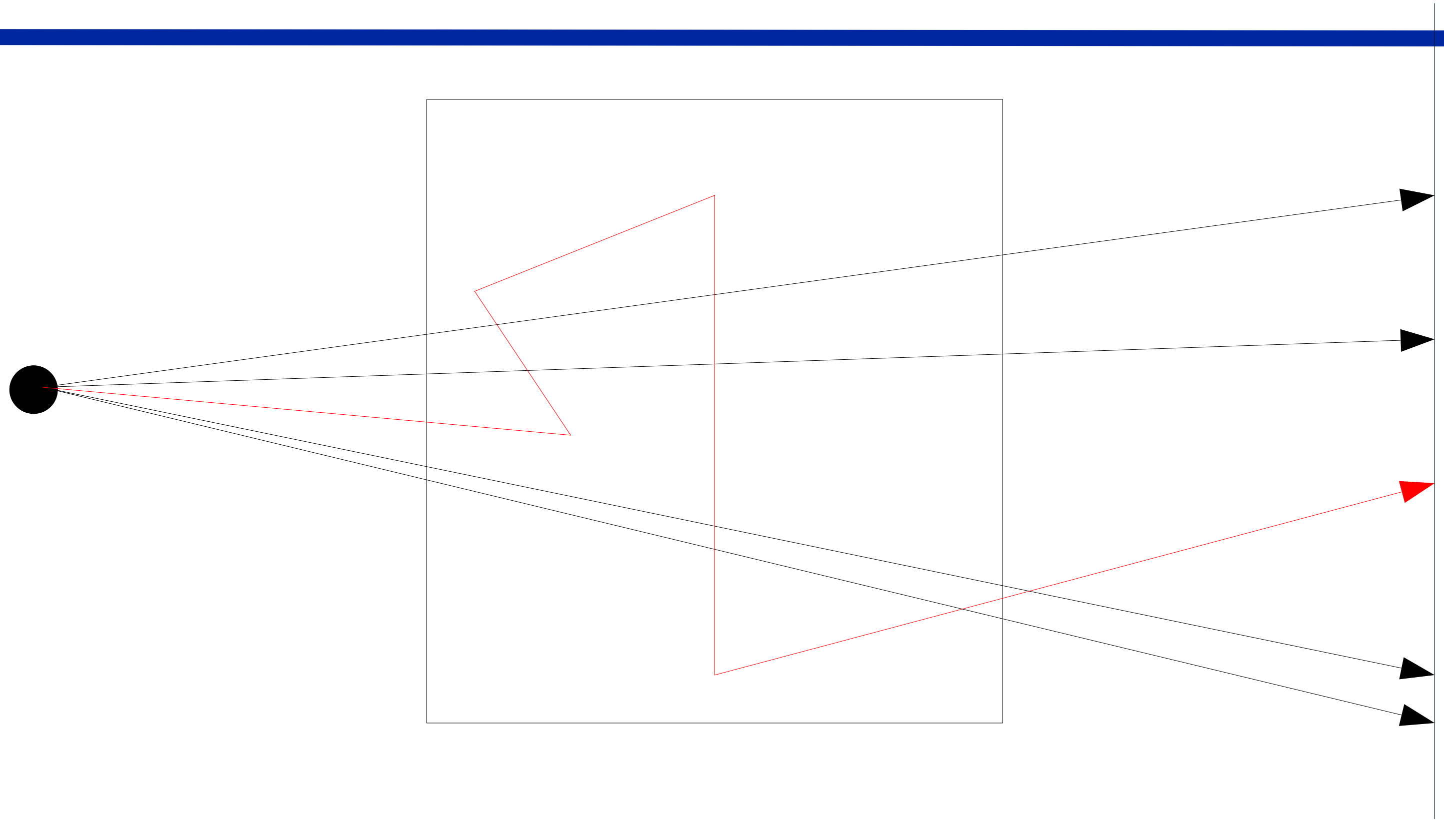
Currently 100 cores for 100 hours needed for single study (tomography of a mouse with respiration movements)
Use the computing grid, not always convenient, not always available

===> the hgate project : run on GPU or hybrid : CPU-GPU

Recoding for GPU is not an easy job :
Many man-years of development → complex code
Million lines of code, it is hard to extract a minimum subset;
Object oriented, not ideal for GPU
Many classes with lots of interdependence, performance, maintainability and user friendliness achieved at the cost of complex code

Typical execution path :

Where am I → current medium → cross sections
Random selection of length to next interaction
Compute next change of medium
If closer, move to next boundary
Else choose process (random, use X sections)
Random choice of direction and energy of secondary particle(s)



Photons end at random positions on detector : Scatter operation

Majority go straight and fast,
a few have one or more interactions : non-uniform processing
the slowest determines total execution time
voxelized volume makes it even worse

An orgy of random numbers

input X sections : the probability of process to occur : P(E, ϑ)

need to generate events with prob = P : invert P, integration

not always easy or possible

Composition-rejection method requires several random numbers

how many not known a priori

Need to generate in-flight, more branchings

Explored the following :

Identify small fractions of the code which consume most of the exec. Time
There is no such thing !

Run one particle per thread. High parallelisms but :
Size of code and data, does not fit in local memory, use global with loss of performance
Divergence : not all particles follow same path
Particles which do not reach the detector : useless, waste of resources

Moreover : for medical imaging, material changes quickly, in contrast with particle detectors which are optimized on probability of interaction : prediction of the next boundary in small volumes will affect performances.

This will be hard work, not worth the effort for a small acceleration

Conclusion : need to revisit the algorithm

For medical or animal imaging, there are simplifying factors :

Xray radiography is low energy photons (10-100 keV << me=500keV)
no pair production
Only 3 processes to consider :
- Rayleigh scattering : direction modified, same energy;
- Photoelectric : photon stops
- Compton scattering : direction and energy modified

Secondary particles (electrons) are very low energy and stop,
no need to propagate secondaries

Biology material rather uniform, close to water
no huge variations in cross sections along photon trajectory

The question :
How many photons will reach each pixel, and what will be their energy spectrum

$$\int P(\text{photon from source to pixel})\, d\Omega$$

Geant algorithm to compute the integral :

Generate photons from the source, propagate with realistic probabilities see
Where they end and accumulate in pixels.

Can be expressed differently ...

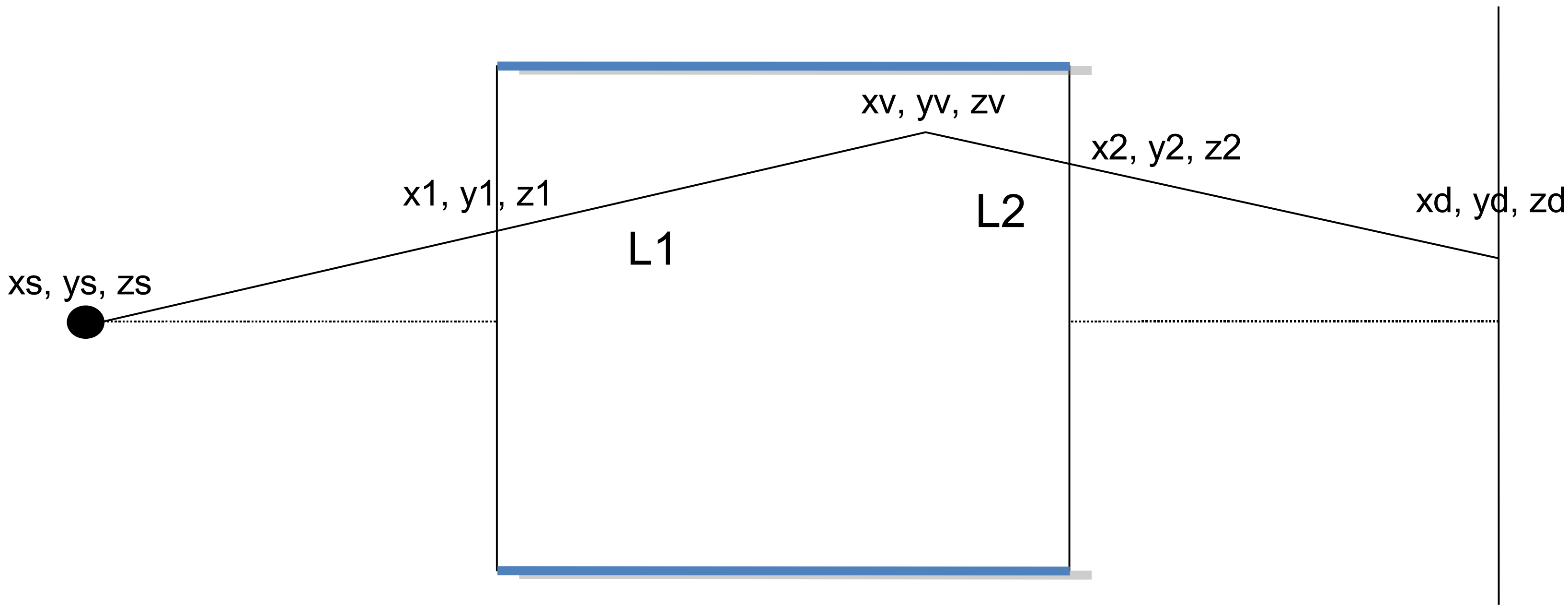
All photons are issued from same point source

For a given pixel, can integrate separately for 0, 1, 2, … interactions

For a given number of interactions (vertices), the only free variables are the vertex positions.

Cross sections are relatively uniform, we can distribute vertices uniformly, then compute probability along the path and reweight the events.

Should give the same result.



$X1 = Xs + (Xv - Xs)*(z1 - zs)/(zv - zs)$

X2 = same with Xd

$L1 = |Xv - X1|$

L2 same with X2

$V1 = Xv - X1$

V2 same with X2

$Cos\ \vartheta\vartheta = V2.V1/norm(V2)/norm(V1)$

$Prob(evt) = exp(-\ \mu(L1 + L2))\ .P(E, \vartheta)$

Non-uniform medium :

compute vertex probability fron Xsection at the precise location;
integrate attenuation over particle path instead of – L μ

loop over all detector pixels, then :

For each event (photon)

random choice of vertex number and position(s) (x, y, z)
choose interaction (Compton or Rayleigh) for each vertex, equal proabilities
Compute and factorize vertex probabilities (evaluate cross section)
integrate attenuation along linear sections (account for local materialproperties)
correct for solid angle of pixel seen from last vertex
correct for probability of source to emit photon in direction of 1^{st} vertex

All photons with same number and type of vertices can be processed in parallel.

Events with many vertices less probable, need to simulate smaller quantity for the same statistical error.

3 random numbers per vertex : dedicated Mersennetwister kernel at init

Simulate the detector response.

No need to follow the particle through active medium, use a parametrized psf, derived from detailed simulations or experimental date.

Add noise to the final image

Shot noise, readout, leakage current ...

This is still work in progress, so far stop at detector entry

Uniform cube of water, with opaque faces except entry and exit

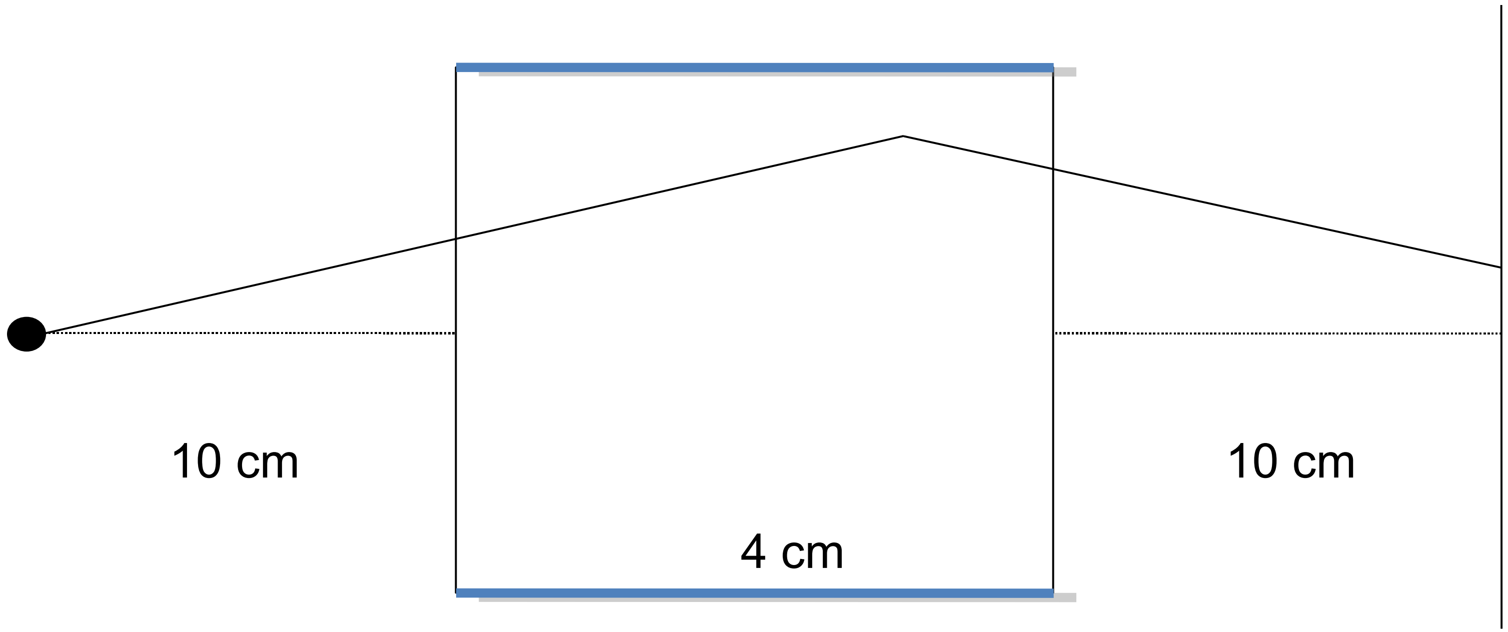
Only 1 vertex per photon

Compton scattering only but full attenuation

Detector not simulated, record coordinates of photon at entry

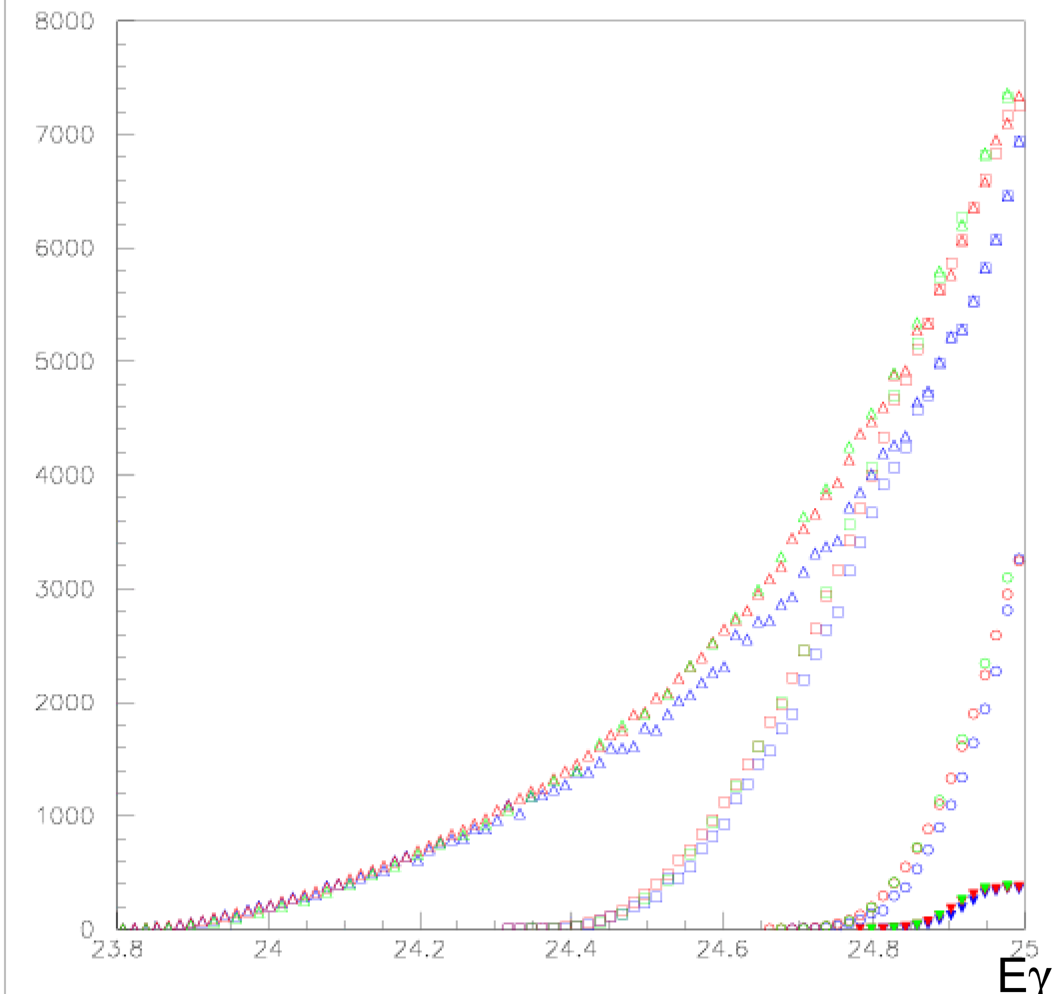
Cross sections and attenuation measured experimentally from EGS simulations and fitted with polynoms. Not perfect but adequate for first test.

Histogram photon energy at detector plane



Max photon energy : monochromatic source energy, photons which go straight
Min photon energy determined by max scattering angle.
Spectrum depends on Xsections and geometry.

Compare results



EGS : full simulation cpu
Fast sim, cpu
GPU, new approach

Only 1 Compton interaction

Différents symbols : Rmax det

1 cm, 3 cm, 10 cm, 50 cm

Features well reproduced, small Differences due to parametrization of X sections by polynoms

Monochromatic source 25 keV

Comparison with Geant or EGS would be unfair at this stage
the new approach does far less than the standard code

Compare Java version with GPU implementation :
openCL on AMD/ATI Firepro V7800

Speedup was of order x60, no optimization performed on any side

GPU version suffers from low computational intensity
And transfer of final results to CPU. Calculations are simple

Conclusion

A new approach has been explored for the simulation of Xray radiography

Specific problem significantly simpler than general case

It is also easier to code for GPU.

First results are encouraging

Things to do :

Multiple interactions, multiple processes, multiple materials.
No deep changes in the algorithm

Push the parallelization of standard Geant4, explore the bottlenecks

Optimize the GPU code