
Optical Photon Simulation on GPUs

Simon C. Blyth

Dec 16, 2016

Contents

1	Overview	1
2	Plans for the Year Ahead	2
2.1	Opticks integration with JUNO Offline	2
2.2	Visualization Rollout	2
2.3	Nurturing Opticks Usage	3
2.4	Attracting Users and Developers	3
2.5	Improve analytic GPU geometry handling	4
2.6	Computing Resources for Opticks	4
2.7	Integration of Opticks with Geant4	4
3	Highlights of the past year	5
3.1	Matched Opticks and Geant4 Optical Photon Generation and Propagation	5
3.2	Marketing Activities	6
4	Conclusion	7

1 Overview

Over the next year my primary goal is to integrate the GPU optical photon simulation toolkit I have developed, Opticks, with the JUNO Offline software in order to enable both JUNO monte carlo production and all JUNO physicists to benefit from unprecedented optical photon simulation performance. Opticks works together with Geant4, replacing just the optical photon processing with an equivalent GPU implementation that is accelerated with the NVIDIA OptiX ray tracing engine.

Massively parallel GPU processing results in performance improvements expected to exceed 1000x the existing CPU photon simulations on workstation GPU machines. Performance factors are obtained by linear extrapolation according to CUDA core counts from development machine measurements. Such performance means that optical photon simulation time will become effectively zero compared to the rest of the simulation and optical photon memory consumption constraints are also removed as only photons that hit PMTs need to consume CPU memory.

Time and memory efficient optical photon simulation is vitally important for photomultiplier based neutrino detectors such as Daya Bay, JUNO, IceCube, Baikal GVD and MiniBooNE. Fast optical photon simulation is also crucial for some dark matter and rare decay search experiments such as LZ and EXO and also has industrial uses in the design of medical imaging systems. Physicists associated with all the above mentioned experiments as well as a physicist working in a company on the design of time-of-flight positron emission tomography (TOF-PET) scanners and several of the Geant4 development team have expressed an interest in using Opticks and several of them are taking the first steps in evaluating Opticks. This interest in Opticks was mostly generated due to the high

ranking of <http://simoncblyth.bitbucket.org> in web searches for related terms. This high ranking has been achieved by regular updating of the Opticks web presence with all presentations, demonstration videos and code updates being publically discoverable.

My secondary goal for the year ahead is to nurture the interest from the community by working with some of the users in order to improve the user experience and also to implement some features within Opticks that their usage requires. Currently the principal difficulty of Opticks adoption with new detector geometries is the need for manual implementation of analytic geometry intersection code for optically critical volumes such as photomultiplier tubes. Ideas for improving this situation are detailed in the Plans section below. In addition to nurturing existing leads I also intend to continue to generate more interest from the community through marketing activities such as conference presentations, demonstration videos and technical papers describing Opticks.

My primary goal over the past year was reaching equivalence between the GPU and CPU implementations of all relevant optical physics processes including scintillator reemission. I have achieved this goal as detailed in the Highlights section below.

2 Plans for the Year Ahead

2.1 Opticks integration with JUNO Offline

Although monte carlo simulations are critical throughout the lifetime of an experiment they are particularly important during the current JUNO detector design and optimization phase. Many questions such as the number, characteristics and arrangement of small PMTs and the design of light concentrators need answers that only simulation can provide. Opticks performance promises to be transformative for JUNO simulation studies. Because of this my primary goal over the coming months is to integrate Opticks with the JUNO Offline software.

Opticks requires generation steps to be collected from Geant4 scintillation and Cerenkov processes and photon detector hits are subsequently returned en masse to the Geant4 hit collections allowing subsequent electronics simulation to proceed unmodified.

Cosmic muon events with maximum path length through the JUNO central detector sphere of 35m diameter of liquid scintillator are expected to yield of order 60 millions of optical photons. Automated GPU launch splitting depending on available GPU memory will probably be required to generate and propagate such numbers of photons. Such splitting can be straightforwardly implemented as every scintillation or Cerenkov generation step that is transferred to the GPU carries the number of photons to be generated for the step.

I plan to implement as much as possible of the integration beneath the level of the experiment, at the Geant4 level, allowing this work to be carried over to other experiments straightforwardly. To what extent this is possible will depend on how tightly JUNO Offline couples with Geant4. Fortunately JUNO Offline is in active development so it is possible that changes can be made to ease the integration.

2.2 Visualization Rollout

Migrating detector geometry and photon data to the GPU not only enables unprecedented optical photon simulation performance it also enables unprecedented detector and event visualization performance. Opticks provides detector and photon propagation visualizations on a wide range of OpenGL 4 capable systems, NVIDIA GPUs are not required. This capability is due to the layered and ordered dependency structure of the almost 20 C++ packages that make up Opticks. This structure also maximises code reuse and promotes clear organization due to my policy of implementing code within the lowest level package that meets its dependency requirements.

Opticks visualization features such as interactive propagation time scrubbing and interactive photon history selection were motivated by the requirements of debugging the simulation. So far only a very small number of users have experienced these interactive visualization features, integration with JUNO and DayaBay Offline software and the layered dependency structure can provide performant visualizations on a wide range of systems, enabling many more users to benefit from these capabilities. Hopefully such features will encourage more users to install the software on their systems and attract some people into development.

The performance makes it possible to create screen capture movies illustrating simulated event propagations within the detector geometry, that have proved useful for outreach to a wide public audience.

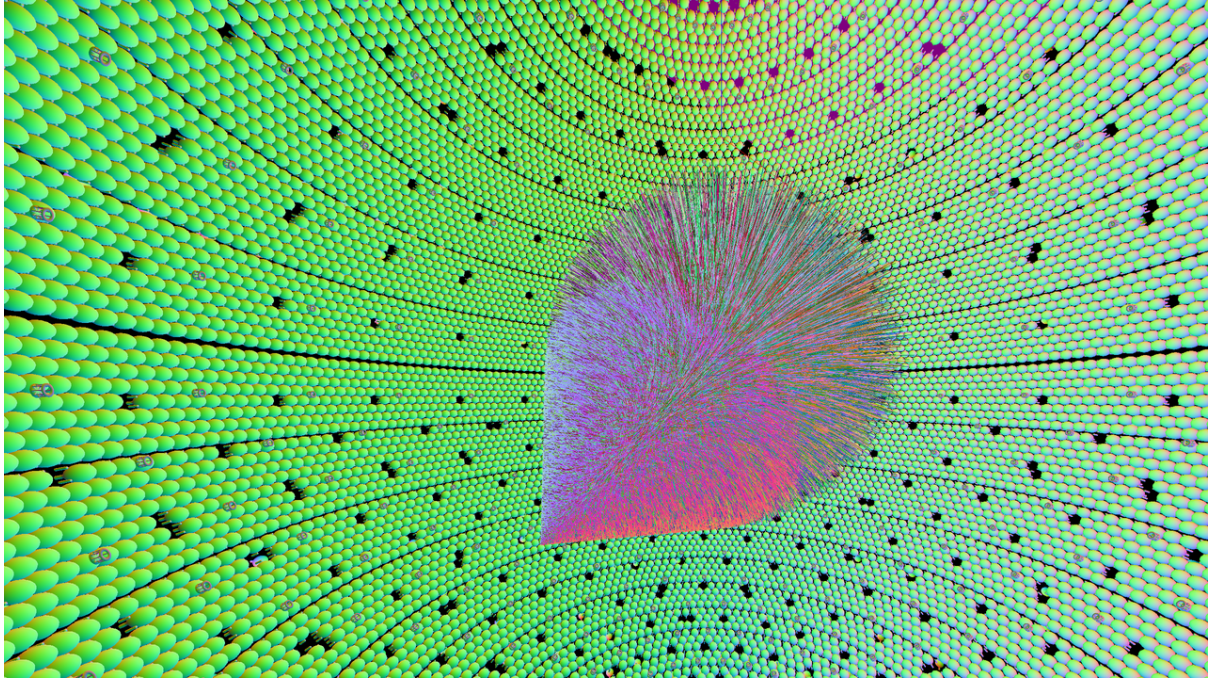


Figure 1: Simulated Cerenkov and scintillations photons from a 100 GeV muon travelling across the JUNO antineutrino detector viewed from inside the spherical scintillator. Primary particles are simulated by Geant4, generation “steps” of the primaries are transferred to the GPU and photons are generated, propagated and visualized all on the GPU. Photons that hit PMTs are returned back to Geant4 to be included into standard hit collections. Photon colors indicate the polarization direction.

2.3 Nurturing Opticks Usage

Several potential Opticks users from experiments based on liquid Xenon scintillation rely on the NEST (Noble Element Simulation Technique) models for photon generation on the CPU. I plan to enable the use of CPU generated photons with Opticks, despite the overheads of this approach, in order to ease adoption for users that have this requirement.

Opticks normally treats scintillation and Cerenkov photons as resident to the GPU, the photons are generated on the GPU and only the small fraction that hit detectors need to be copied to the CPU. This avoids duplicated photon memory allocation and copying from CPU to GPU. The generation step parameter data that still needs to be copied to the GPU for each event is typically more than a factor of 100 times smaller than the photon data.

A further feature requested by users that I plan to provide is transparent fallback to Geant4 optical photon simulation on systems that do not have the NVIDIA GPU required for accelerated simulation.

2.4 Attracting Users and Developers

Opticks was conceived and has so far been developed almost entirely by myself. Although there have been several expressions of interest to help with Opticks by JUNO colleagues the steep learning requirements have so far prevented any substantial contributions.

In order to ensure the survival of Opticks it is vital to attract users and developers. In preparation for a course on Opticks that I gave at the 2016 Weihai summer school, organized by Shandong University, I prepared installation scripts and documentation and tutorial materials. Following experience from the school I plan to split the materials into sections separately targeting inexperienced, intermediate and advanced users.

I have started compiling a mailing list of people who have expressed an interest in Opticks. Once validations within a production setting are achieved it will be appropriate to formalize the list and start using it to keep people informed of Opticks milestones as well as adopting a more active approach to growing the list by making contacts with groups that use simulations of large numbers of photons.

2.5 Improve analytic GPU geometry handling

Geometry information is provided to the NVIDIA OptiX ray tracing engine via bounding box and intersection CUDA programs. The intersection program computes whether a ray intersects with a primitive and reports the parametric t-value of the intersection if it does.

Tessellated detector geometries are exported from Geant4 using the G4DAE geometry exporter that I developed. The tessellation approximation was however found to be unacceptable for optically critical surfaces such as those of the photomultiplier tubes. Geant4 geometry can be considered to be comprised of a tree of boolean operations such as unions and intersections acting upon primitives such as spheres and cylinders. I developed a partitioning approach that exploits the rotational symmetry of the photomultiplier tube allowing the tree representation to be decomposed into a list of partial primitives by splitting at the geometrical intersections of the basis volumes. This results in an analytic GPU geometry that yields intersection positions that precisely match Geant4, however development of this analytic description is currently a manual process that needs to be repeated for each new geometry.

Currently the principal difficulty of Opticks adoption with new detector geometries is this need for manual implementation of analytic geometry intersection code for optically critical volumes. I plan to initially apply a similar manual approach again for the JUNO photomultiplier tube shapes and other optically critical surfaces. While a fully automated solution to generate analytic OptiX intersection code for any Geant4 geometry is not feasible without additional manpower I expect the implementation time can be greatly reduced by improved documentation and modularization to enable code reuse.

2.6 Computing Resources for Opticks

Opticks development has so far been performed on my personal MacBook Pro (2013) laptop, with NVIDIA GeForce GT 750M 2048 MB, which I purchased for this purpose. Occasional ports of Opticks from Mac OSX to Linux running on GPU workstations owned by Shandong University and the IHEP Computing Center have been done and a partial non-GPU port of Opticks to a borrowed Windows desktop at NTU has demonstrated the feasibility of a full Windows port.

As Linux GPU workstations and desktops are the target platforms for production usage of Opticks it would be highly beneficial to purchase such a system to become the principal development machine for Opticks. Dual boot Linux/Windows systems are preferred as GPU debugging tools are more mature on Windows. The IHEP Computing Center and Shandong University also plan to purchase further GPU workstations that can be used to develop and test Opticks+Geant4 running in a production setting.

2.7 Integration of Opticks with Geant4

At the 19th Geant4 Collaboration Meeting in September 2014, the Geant4 Collaboration accepted my proposal to contribute the G4DAE exporter to the 2015 Geant4 release. However during 2015 my ongoing Opticks validation effort revealed several bugs in the exported Daya Bay geometry, including the cleaved meshes issue and some missing optical surfaces. In retrospect it is clear that the only way to validate an exported geometry is to be able to validate simulations within that geometry by comparison with pure Geant4 simulations. Thus I have deferred the integration of the G4DAE geometry exporter with Geant4 until full geometry simulation validation has been achieved.

I have maintained contact with some Geant4 developers, who remain interested to incorporate the G4DAE geometry exporter within the official Geant4 release. If full geometry simulation validations with JUNO are achieved promptly in 2017 it may be possible to do the incorporation work in time for the 2017 Geant4 release. I expect incorporation of the G4DAE geometry exporter with Geant4 to be relatively straightforward, as it has the same dependencies and usage pattern as the standard Geant4 GDML exporter and it has clear benefits for any Geant4 user. Whether and how the Opticks GPU photon propagation might be integrated with Geant4 in a general manner is much less clear. Experience with Opticks use in production and a general solution for the problem of analytic GPU geometry is needed before deciding what approach to pursue on this.

3 Highlights of the past year

3.1 Matched Opticks and Geant4 Optical Photon Generation and Propagation

All of the Geant4 optical photon propagation processes relevant to Daya Bay and JUNO are implemented on the GPU within OptiX CUDA programs. The processes include absorption, Rayleigh scattering, Fresnel reflection and refraction, diffuse reflection and scintillator reemission. Similarly optical photon generation from scintillation and Cerenkov processes using buffers of generation step parameters collected from Geant4 are implemented on the GPU within OptiX CUDA programs.

Validation comparisons use a single executable that performs both the Geant4 and Opticks simulations and writes two events using an Opticks event format that includes highly compressed information for the first 16 photon propagation points. These events are compared by forming chi-squared distances for:

- photon history counts : within the 100 most frequent categories
- photon step-by-step distributions : 8 quantities, position, time, polarization and wavelength

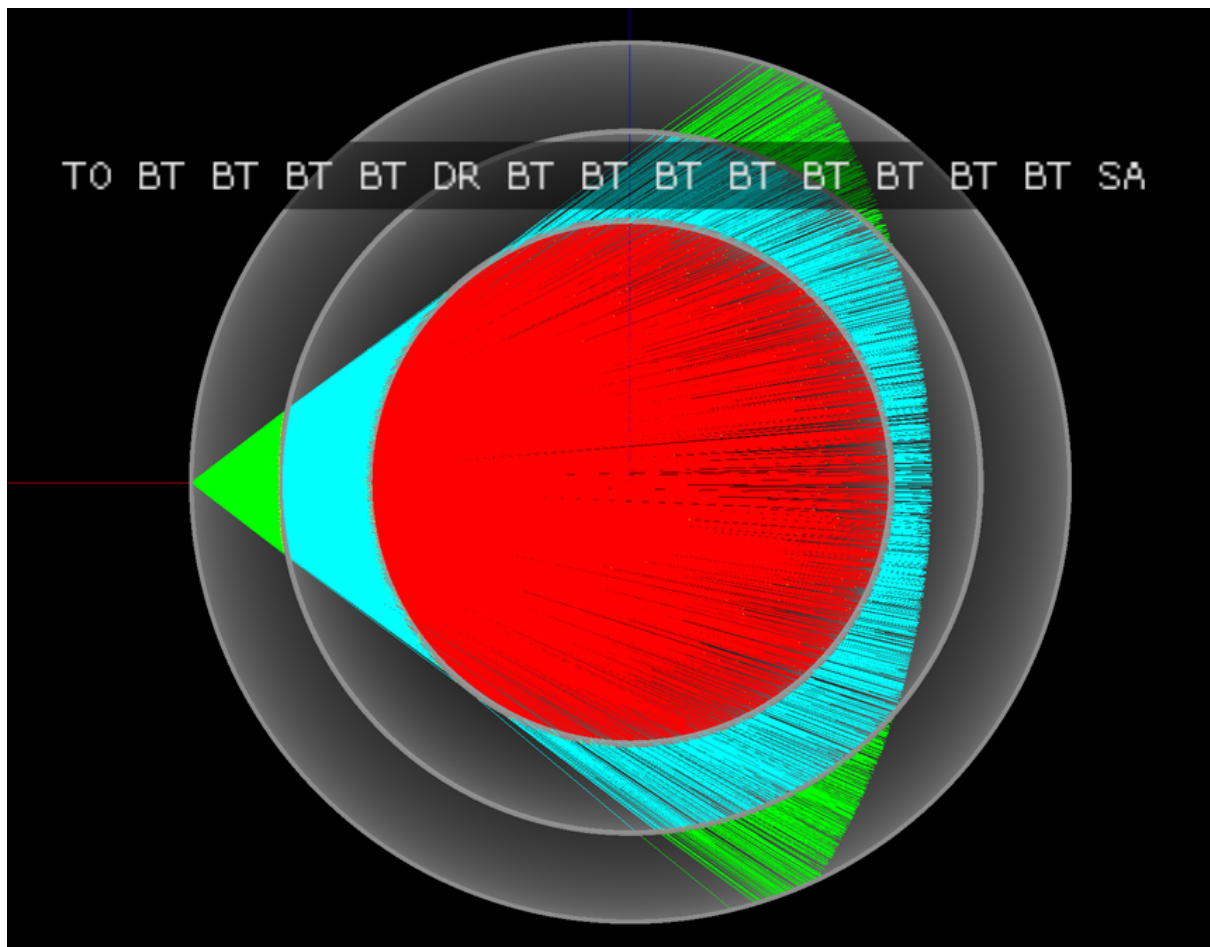


Figure 2: Simplified three liquid detector geometry arranged in concentric spheres separated by acrylic. Photons from history category “TO BT BT BT BT DR BT BT BT BT BT BT BT BT BT SA” are shown, where the abbreviations are, TO:Torch, BT:Boundary Transmit, DR:Diffuse Reflect, SA:Surface Absorb and the initial photons all travel along the X axis indicated by the red line. The line colors represent the material the photon is traversing, red:Gadolinium doped liquid scintillator, cyan: liquid scintillator and green: mineral oil. The simplicity of this test geometry was adopted in order to debug an issue of Geant4 using the group velocity from the wrong material after refraction.

.	1000000	1000000	373.13/356 =	1.05	(pval 0.256 prob 0.744)
	Opticks	Geant4			
0000	669843	670001	0.02	TO BT BT BT BT SA	

0001	83950	84149	0.24	TO AB
0002	45490	44770	5.74	TO SC BT BT BT BT SA
0003	28955	28718	0.97	TO BT BT BT BT AB
0004	23187	23170	0.01	TO BT BT AB
0005	20238	20140	0.24	TO RE BT BT BT BT SA
0006	10214	10357	0.99	TO BT BT SC BT BT SA
0007	10176	10318	0.98	TO BT BT BT BT SC SA
0008	7540	7710	1.90	TO BT BT BT BT DR SA
0009	5976	5934	0.15	TO RE RE BT BT BT SA
0010	5779	5766	0.01	TO RE AB
0011	5339	5269	0.46	TO BT BT BT BT DR BT BT BT BT BT BT SA
0012	5111	4940	2.91	TO BT BT RE BT BT SA
0013	4797	4886	0.82	TO SC AB
0014	4494	4469	0.07	TO BT BT BT BT DR BT BT BT BT SA
0015	3317	3302	0.03	TO BT BT SC BT BT BT BT BT SA
0016	2670	2675	0.00	TO SC SC BT BT BT BT SA
0017	2432	2383	0.50	TO BT BT BT BT DR AB
0018	2043	1991	0.67	TO SC BT BT BT BT AB
0019	1755	1826	1.41	TO SC BT BT AB

The above text table shows the photon counts from Opticks and Geant4 for the top 20 photon history categories obtained from a simulation of 1 million photons within the tconcentric test geometry together with the chi-square distance of each category and the overall chi-squared distance. The abbreviations are TO:torch, BT:boundary transmit, SA:surface absorb, AB:bulk absorb, RE:reemission, SC:scatter, DR:diffuse reflect. The overall chi-squared per degree of freedom and a similar one obtained for the distributions at each propagation point is used to check the consistency of the simulations.

The top 100 photon history categories correspond to ~900 photon propagation points with 8 quantities per point this corresponds to 7200 histogram pairs that are compared. After numerous bug fixes directed by the next largest chi-square contributor statistically consistent GPU and CPU simulations for the photon counts and distributions have been achieved.

3.2 Marketing Activities

Slides and some videos of the presentations below are accessible from <http://simoncblyth.bitbucket.org>

- December 2016, JUNO Workshop, LLR, Ecole Polytechnique, Paris.
Opticks : Optical Photon Simulation for Particle Physics with NVIDIA OptiX
Invited workshop talk.
- October 2016, 22nd International Conference on Computing in High Energy and Nuclear Physics (CHEP).
Hosted by SLAC and LBNL, San Francisco.
Opticks : Optical Photon Simulation for Particle Physics with NVIDIA OptiX
Contributed conference talk.
- July 2016, Particle Physics Summer School, Weihai, Organized by Shandong University.
Opticks : Optical Photon Simulation for Particle Physics with NVIDIA OptiX
Invited course on Opticks, including 90 minute lecture and two 90 minute tutorial sessions
- May 2016, LeCosPA Seminar, National Taiwan University, Taipei.
Opticks : Optical Photon Simulation for Particle Physics with NVIDIA OptiX
Invited seminar.
- April 2016, NVIDIA's GTC (GPU Technology Conference), San Jose, California.
Opticks : Optical Photon Simulation for Particle Physics with NVIDIA OptiX
Invited conference talk on Opticks to a diverse audience,
<http://on-demand.gputechconf.com/gtc/2016/video/s6320-simon-blyth-opticks-nvidia-opt>
- April 2016, Shared Opticks simulation animation to YouTube and YouKu video sharing sites.
<https://www.youtube.com/watch?v=QzH6y0pKXk4>

- March 2016, Daya Bay Collaboration Meeting, IHEP, Beijing.
Opticks : GPU Optical Photon Simulation
Update to Daya Bay Collaboration, including single PMT validation
- Jan 2016, PSROC Meeting, SYSU, Kaoshiung, Taiwan.
Opticks : GPU Optical Photon Simulation
Same ground as Xiamen talk, but aiming at more diverse audience
- Jan 2016, JUNO Meeting, Xiamen University.
Opticks : GPU Optical Photon Simulation
Progress report on handling JUNO geometry and beginning validations.

Important Earlier Activities

- September 2014, 19th Geant4 Collaboration Meeting, Okinawa.
G4DAE : Export Geant4 Geometry to COLLADA/DAE XML files
Invited guest talk to the Geant4 Collaboration introducing geometry exporter.

4 Conclusion

Opticks promises to provide unprecedented optical photon simulation performance with effectively zero simulation time compared to other processing and that only requires CPU memory allocation for photons that hit detectors. My primary objective for the year ahead is to realize this promise within JUNO and Daya Bay and to lay the ground work to fulfil the promise within other experiments.

Opticks performance arises directly from the massive GPU parallelism that the NVIDIA OptiX ray tracing engine makes accessible. GPU implementations of all relevant optical physics processes have been validated. Implementing analytic GPU geometry for optically critical surfaces remains a stumbling block for Opticks adoption.