

# Progress on Opticks

<sup>1</sup>Yunlong Li, <sup>1</sup>Keith Evans, <sup>1</sup>Adam Davis, <sup>2</sup>Sajan Easo, <sup>2</sup>Raja Nandakumar, <sup>1</sup>Evelina Gersabeck, <sup>1</sup>Marco Gersabeck, <sup>2,3</sup>Lucas George Girardey

<sup>1</sup>University of Manchester,  
<sup>2</sup>Rutherford Appleton Laboratory  
<sup>3</sup>Southampton University  
yunlong.li@cern.ch

November 3, 2021

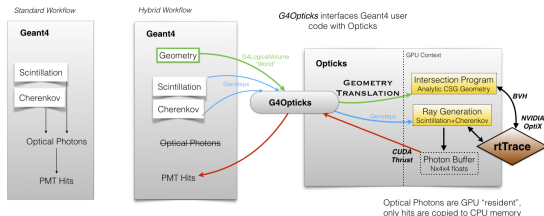


# Introduction: Opticks

## Geant4 + Opticks Hybrid Workflow : External Optical Photon Simulation

<https://bitbucket.org/simonblyth/opticks>

- An open source project developed by Simon Blyth
- Integrates the NVIDIA OptiX GPU ray tracing techniques with GEANT4 simulations



- At initialization, the GEANT4 geometry is translated by Opticks into OptiX GPU context
- The parameters of photon generations, named as "genstep", are collected, including the number of photons, line segment and other parameters needed to generate
- The gensteps and CUDA ports of Cherenkov and scintillation generation are passed to GPU to generate and propagate the photons within the ray generation program provided to OptiX
- Only the hits, which are intersections of rays representing photons and the geometry, are passed back to CPU

# Introduction: Opticks

- Shows ability to accelerate optical photon simulation  $\sim 1000$  times speedup compared to `GEANT4` simulation
- `CaTS`: Ongoing project by Fermilab, formerly `G4OpticksTest`, a calorimeter and tracker simulation framework using Opticks with `GEANT4` for the creation and propagation of optical photons
- `CerenkovMinimal`: Modifications made by Raja and Sajan, showing minimal use of embedded Opticks to act as an starting point for users familiar with Geant4 examples

# Environment settings

- Opticks depends on a lot of external packages and libraries
- All kinds of problems may appear when installing Opticks on a GPU machine
- Using the LHC Computing Grid (LCG) environment makes the installation much easier

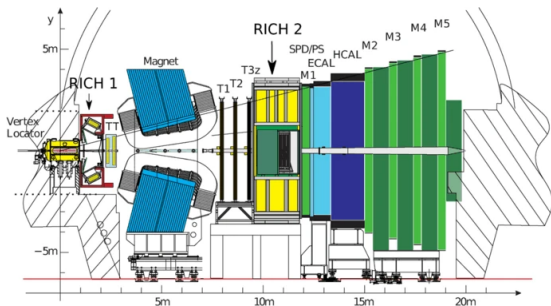
//Add these lines to the opticks config file

```
export LCG_ENV=/cvmfs/sft.cern.ch/lcg
source $LCG_ENV/views/LCG_100/x86_64-centos7-gcc10-opt/setup.sh
opticks--prepend--prefix $LCG_ENV/releases/LCG_100/clhep/2.4.4.0/x86_64-centos7-gcc9-opt/
opticks--prepend--prefix $LCG_ENV/releases/LCG_100/XercesC/3.2.3/x86_64-centos7-gcc9-opt/
opticks--prepend--prefix $LCG_ENV/releases/LCG_100/Boost/1.75.0/x86_64-centos7-gcc9-opt/
```

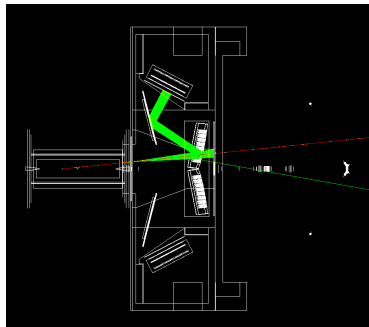
- Build Opticks with Geant4.10.07.p01

# RICH1 with Opticks

- Test Opticks with The two Ring Imaging Chevenkov (RICH) detectors in the LHCb detector
- RICH1 is located before magnet



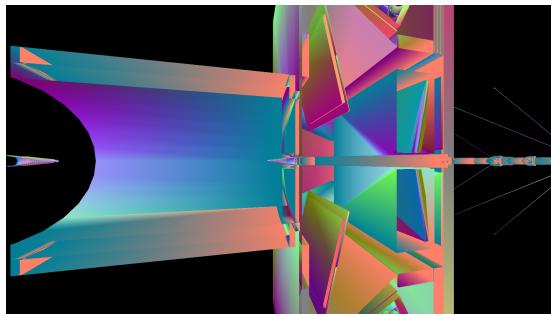
Schematic side-view of the LHCb detector.



GEANT4 simulation results of a 2 GeV electron injected with a small angle between beam pipe.

# RICH1 with Opticks

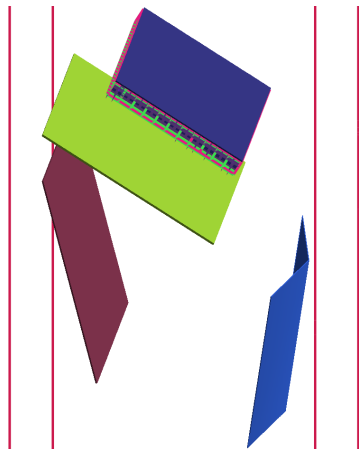
- Exported the RICH1 gdml file from LHCb detector geometry
- Ran into some problems with the geometry translation
  
- Translation of materials: naming conflict in the abbreviation of material names
- Translation of optical surfaces: polished and ground optical surfaces
- Translation of solids: polycone conversion
- Translation of boundary libraries
- Solved with the help of Simon (See the notes [here](#))



OpenGL rendering of the RICH geometry by Opticks.

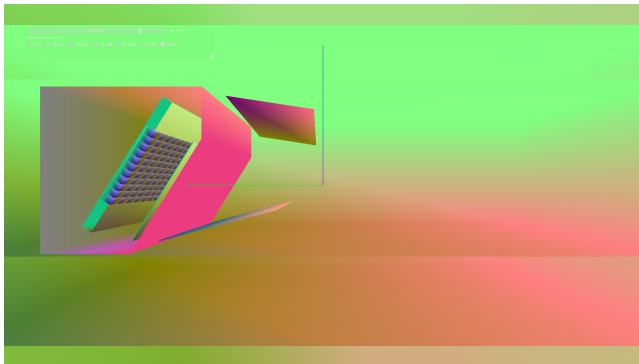
# Simplified RICH

- To better investigate the simulation results between GEANT4 and Opticks, a **simplified RICH** geometry is introduced
- A sphere mirror and a flat mirror reflecting the optical photons into the detector
- A quartz window before the detector
- The detector consists of  $6 \times 11$  modules, and each module have  $4 \times 4$  PMTs



# Simplified RICH

- This geometry can be translated by Opticks with no errors

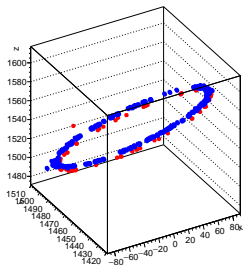


OpenGL rendering of the simplified RICH geometry by Opticks.

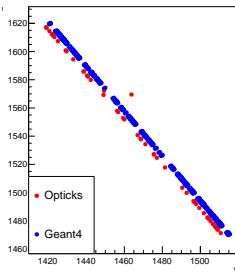


# Simplified RICH

Hits collected by Opticks and Geant4



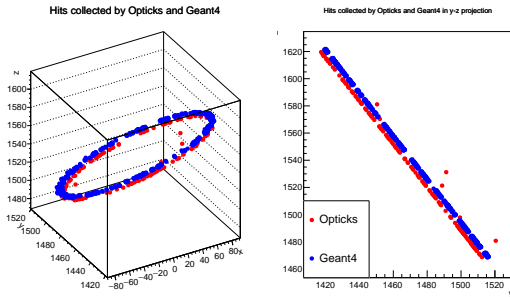
Hits collected by Opticks and Geant4 in y-z projection



- The hits collected by GEANT4 simulation and Opticks simulation give the same ring
- GEANT4 simulation produces  $\sim 500$  photon hits, but Opticks only gives  $\sim 50$  photon hits
- The ways GEANT4 and Opticks record hits are different:
  - GEANT4 records the hits on the sensitive detector, which should be attached to a G4LogicalVolume object. In our case, it's the quartz window before the photon-cathode in each PMT
  - Opticks records the hits on the optical surface with non-zero efficiency, which should be between two G4PhysicalVolume objects. In our case, it's the surface of physical volume for each PMT

# Simplified RICH

- The version of Opticks we are using is back to May (See this [commit](#)). We tested the project with version v0.1.1 (released 28/08/2021) of Opticks and found more hits are given by Opticks (about 150 hits)



- Opticks is undergoing a transition from compatibility with OptiX 6.5 to new OptiX 7 API and all the GPU code is being re-architected by Simon
- What causes such different results needs to be checked

# Summary and outlook

- Reported some problems about geometry translation to Simon and helped solve the problems
- Build a project using simplified RICH geometry to test and compare GEANT4 and Opticks simulation results
- Plans:
  - Work with Simon to package our changes into a general form and merge them into Opticks
  - Further investigate why GEANT4 and Opticks give different number of photon hits
  - Continue the fast forward process to reach the most recent version v0.1.6 (released 06/10/2021) before the move to OptiX 7
  - Push this project into an example embedded in Opticks
  - Perform the speed test in a real-world example of min bias events for LHCb
  - Compare the results of G4 physics process with those from Opticks.
  - Eventually this will lead to integrating Opticks based software into LHCb framework