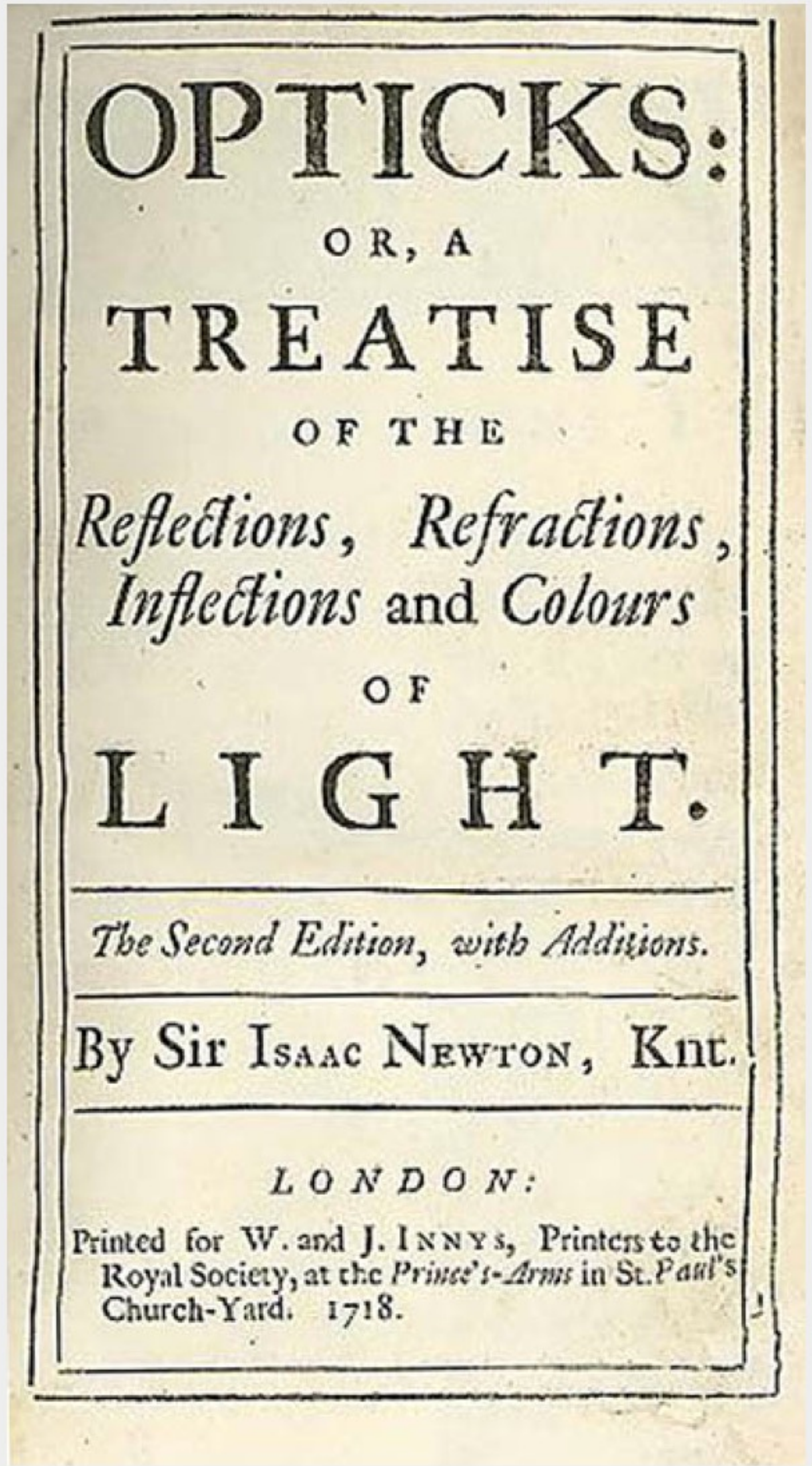
Opticks :
GPU ray trace accelerated optical photon simulation

Open source, https://bitbucket.org/simoncblyth/opticks

Simon C Blyth, IHEP, CAS — CHEP, Krakow, Poland — 21 October 2024

Outline

- Optical Photon Simulation : Context and Problem
  - p2: (JUNO) Optical Photon Simulation Problem...
  - p3: Optical photons limit many simulations => lots of interest in Opticks
  - p4: Optical Photon Simulation ≈ Ray Traced Image Rendering
  - p5: NVIDIA RTX Generations : **RT performance : ~2x every ~2 years**
  - p6: NVIDIA OptiX : Ray Tracing Engine
- Opticks : Solution to Optical Photon Simulation Problem
  - p7: Geant4 + Opticks + NVIDIA OptiX : Hybrid Workflow
  - p8: Geometry Model Translation : Geant4 => CSGFoundry => NVIDIA OptiX
  - p9: Full JUNO, Opticks, OptiX 7.5/8.0
  - p10: **Integrated Analytic + Triangulated Geometry (NEW)**
  - p11: **Interactive ray traced visualization via OpenGL/OptiX interop (NEW)**
  - p13: GuideTube : Torus Triangulated
  - p14: List-node : avoids deep CSG trees
  - p15: Pure Optical TorchGenstep scan : 1M to 100M photons
  - p17: **Optical simulation 4x faster 1st->3rd gen RTX**
  - p19: How much parallelized speedup actually useful to overall speedup?
- p20: Summary + Links
- p21: Acknowledgements
- p22: NEW Opticks User : Ilker Parmaksiz, NEXT-CRAB0 Prototype



(JUNO) Optical Photon Simulation Problem...

Huge CPU Memory+Time Expense

JUNO Muon Simulation Bottleneck
~99% CPU time, memory constraints

Ray-Geometry intersection Dominates
simulation is not alone in this problem...

Optical photons : naturally parallel, simple :

- produced by Cherenkov+Scintillation
- yield only Photomultiplier hits

Optical photons limit many simulations => lots of interest in Opticks

EXPT	Reactor neutrino
Daya Bay	neutrino oscillations
JUNO	mass heirarchy + oscillations => NVIDIA CN Contacts
	Long baseline neutrino beam
DUNE	FermiLab->Sanford, LAr TPC, => Assistance from Fermilab Geant4 Group
	Neutrinoless double beta decay, dark matter, other search
LZ	LUX-ZEPLIN dark matter experiment, Sandford => NVIDIA US Contacts
LEGEND	Large Enriched Germanium Experiment, Gran Sasso/SNOLAB
SABRE	dark matter direct-detection, Australia
AMoRE	Mo-based Rare process Experiment, S.Korea
nEXO	next Enriched Xenon Observatory, LLNL
NEXT-CRAB0	High Pressure Gaseous Xenon TPC with a Direct VUV Camera Based Readout
	Neutrino telescope
KM3Net	Cubic Kilometre Neutrino Telescope, Mediterranean
IceCube	IceCube Neutrino Observatory, South Pole
	Air shower : gamma-ray and cosmic-ray observatory
LHAASO	Large High Altitude Air Shower Observatory, Sichuan
	Accelerator
LHCb-RICH	LHCb ring imaging Cherenkov sub-detector, CERN => NVIDIA EU Contacts

Optical Photon Simulation ≈ Ray Traced Image Rendering

simulation
photon parameters at sensors (PMTs)

rendering
pixel values at image plane

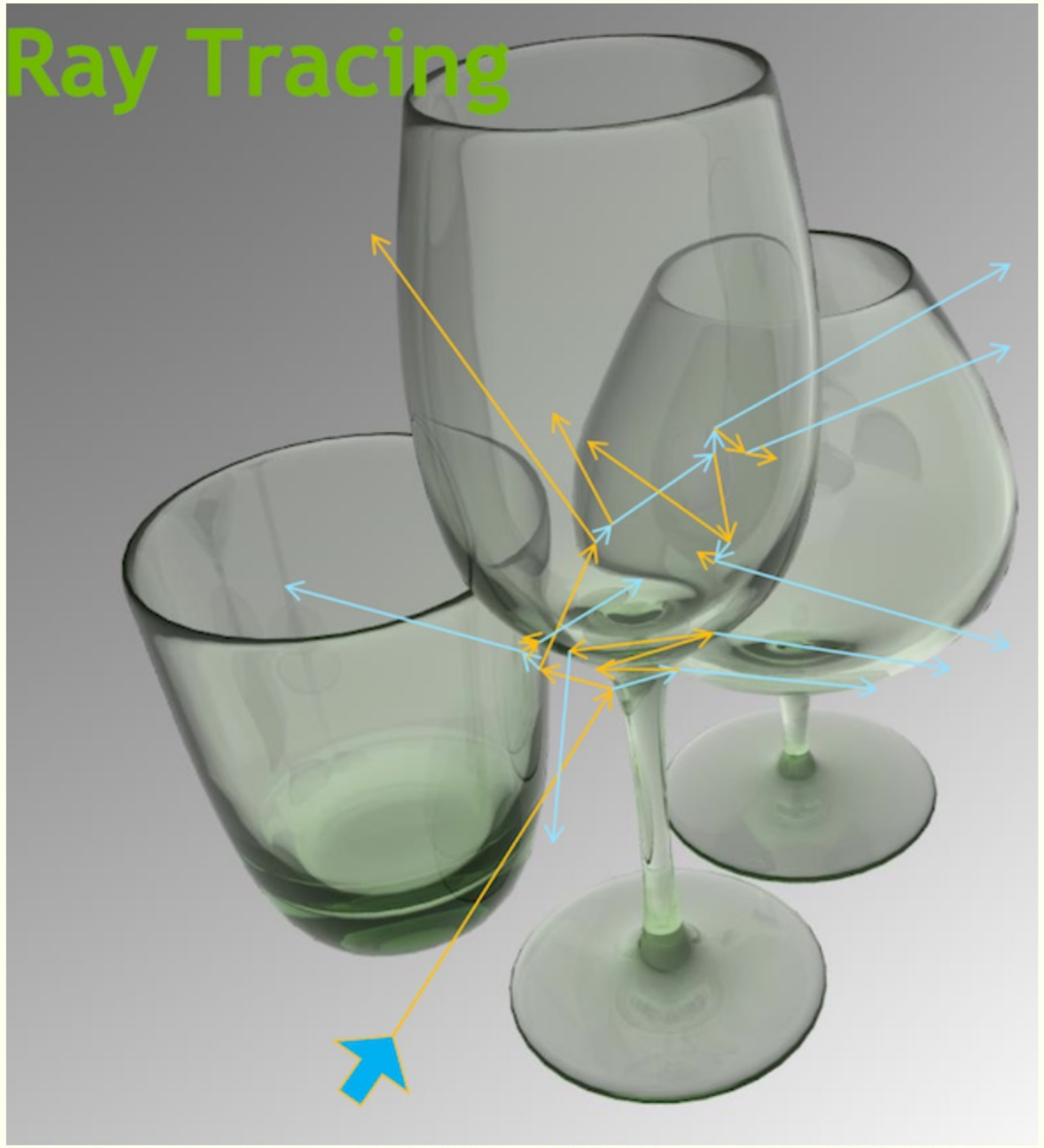
Much in common : geometry, light sources, optical physics

- **both limited by ray geometry intersection, aka ray tracing**

Many Applications of ray tracing :

- advertising, design, architecture, films, games,...
- -> huge efforts to improve hw+sw over 30 yrs

Not a Photo, a Calculation

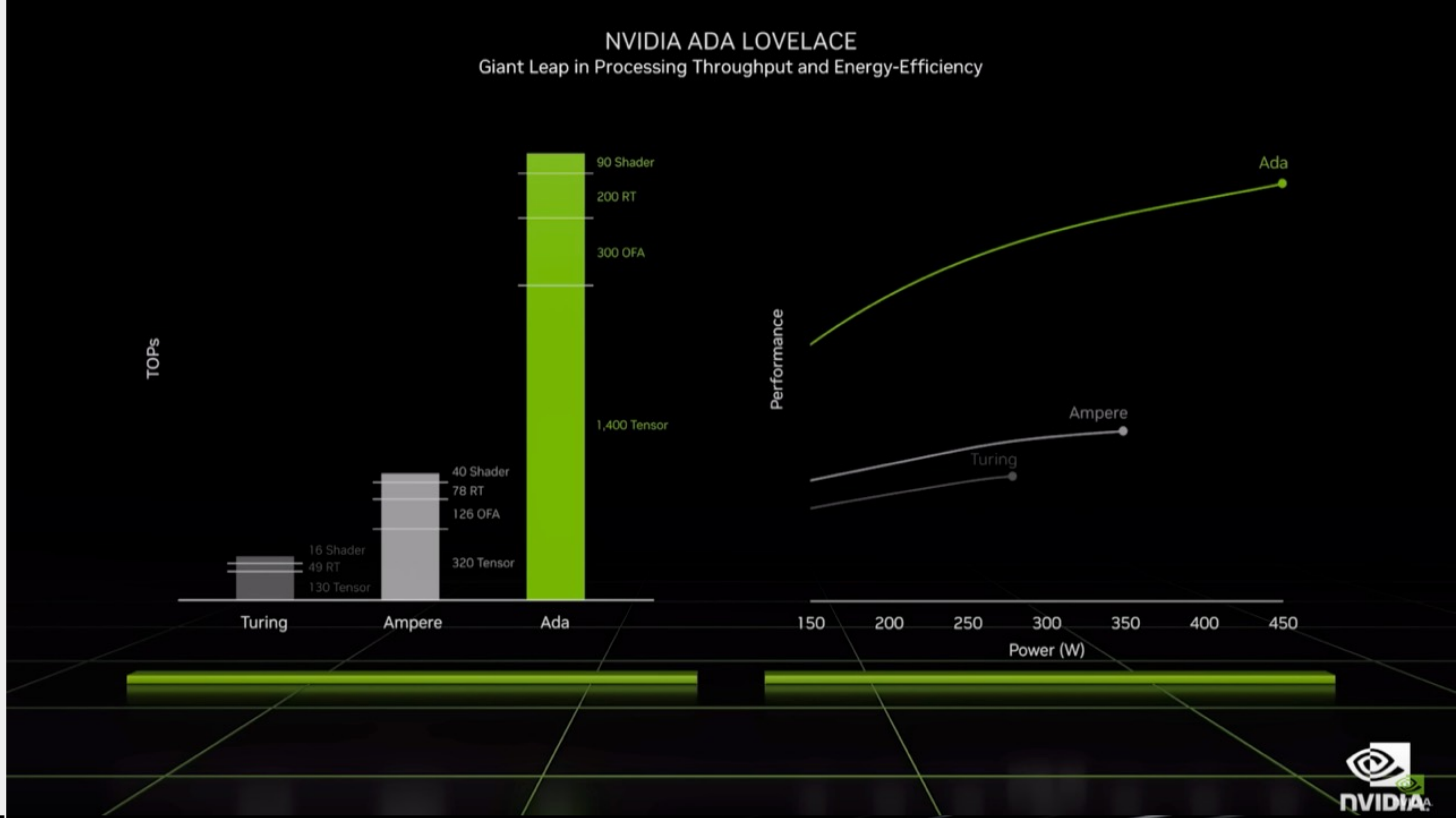


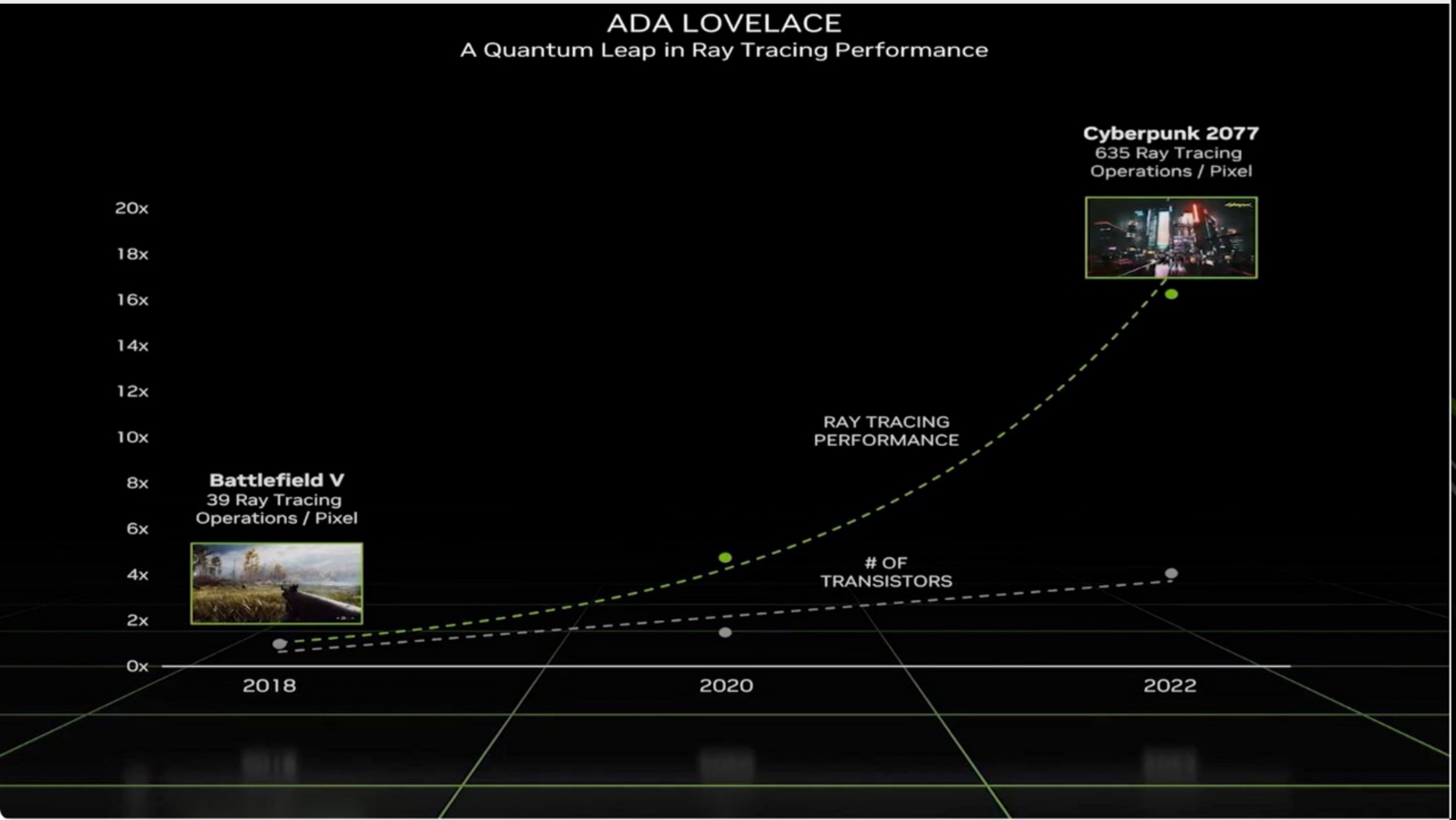
http://on-demand.gputechconf.com/siggraph/2013/presentation/SG3106-Building-Ray-Tracing-Applications-OptiX.pdf

NVIDIA RTX Generations

- **RT Core** : ray trace dedicated GPU hardware
- **Each gen : large ray tracing improvements:**
  - **Blackwell** (2024?5) **Expect:** ~2x ray trace over **Ada**
  - **Ada** (2022) ~2x ray trace over **Ampere**
  - **Ampere** (2020) ~2x ray trace over **Turing** (2018)
- **NVIDIA Blackwell 4th Gen RTX : expected Q1 2025**

ray trace performance : ~2x every ~2 years





NVIDIA® OptiX™ Ray Tracing Engine -- Accessible GPU Ray Tracing

OptiX makes GPU ray tracing accessible

- **Programmable GPU-accelerated Ray-Tracing Pipeline**
- Single-ray shader programming model using CUDA
- ray tracing acceleration using RT Cores (RTX GPUs)
- "...free to use within any application..."

OptiX features

- acceleration structure creation + traversal (eg BVH)
- instanced sharing of geometry + acceleration structures
- compiler optimized for GPU ray tracing

User provides (Green):

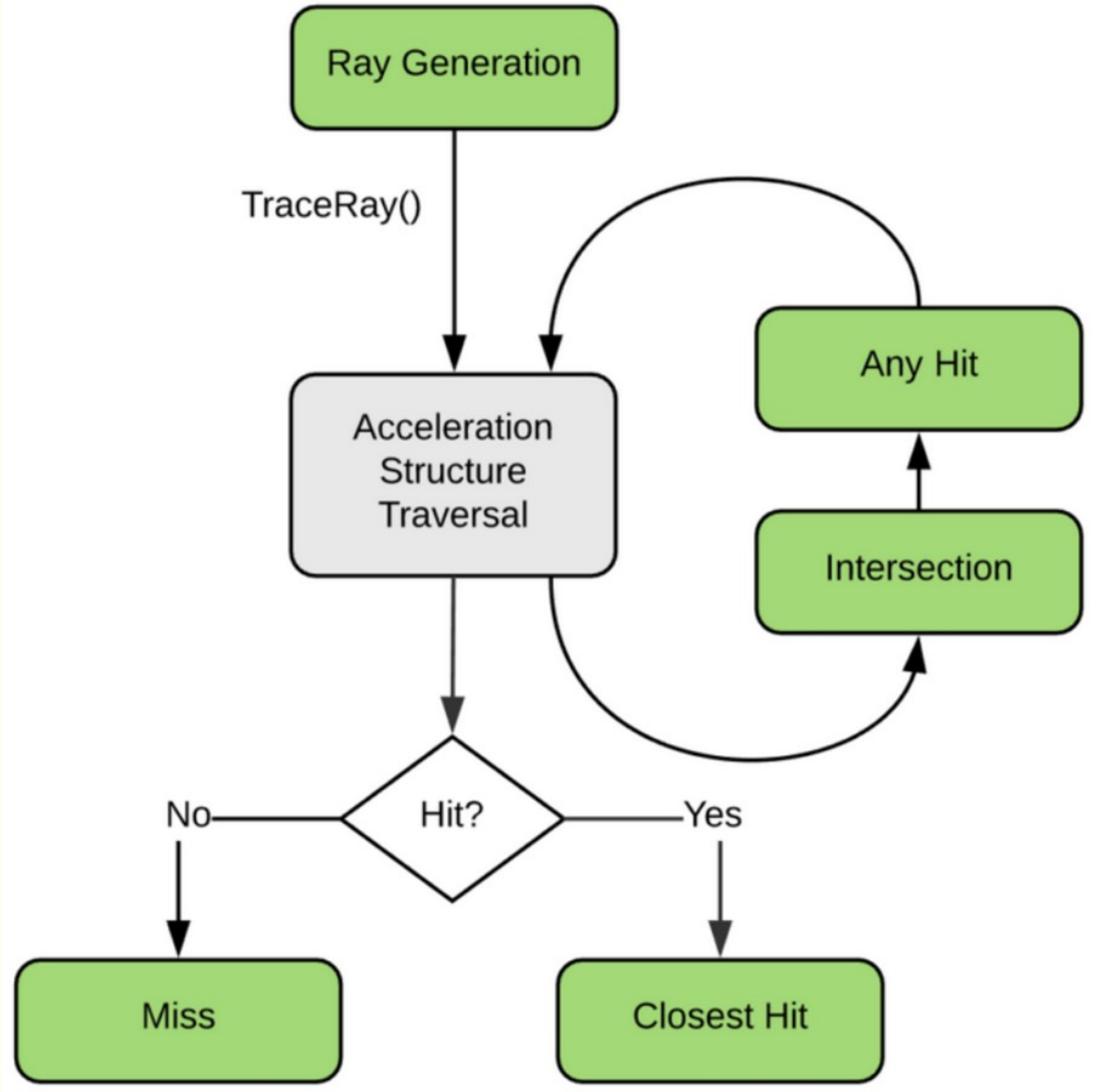
- ray generation
- geometry bounding boxes
- intersect functions
- instance transforms

Latest Release : **NVIDIA® OptiX™ 8.0.0 (Aug 2023)** NEW:

- **Shader Execution Reordering (SER) (Ada: up to 2x)**
- SER: reduced execution+data divergence (on-the-fly)

Flexible Ray Tracing Pipeline

Green: User Programs, **Grey: Fixed function/HW**



Analogous to OpenGL rasterization pipeline

Geant4 + Opticks + NVIDIA OptiX : Hybrid Workflow

https://bitbucket.org/simoncblyth/opticks

Opticks API : split according to dependency -- Optical photons are GPU "resident", only hits need to be copied to CPU memory

Geometry Model Translation : Geant4 => CSGFoundry => NVIDIA OptiX 7/8

Geant4 Geometry Model (JUNO: 400k PV, deep hierarchy)

PV	*G4VPhysicalVolume*	placed, refs LV
LV	*G4LogicalVolume*	unplaced, refs SO
SO	*G4VSolid,G4BooleanSolid*	binary tree of SO "nodes"

CSGFoundry Model

- **array-based -> simple serialization + upload**
- entire geometry in 4 GPU allocations
- factorized using subtree digests

Opticks CSGFoundry Geometry Model (index references)

struct	Notes	Geant4 Equivalent
CSGFoundry	vectors of the below, easily serialized + uploaded + **used on GPU**	None
qat4	4x4 transform refs *CSGSolid* using "spare" 4th column (**becomes IAS**)	Transforms ref from PV
CSGSolid	refs sequence of *CSGPrim*	Grouped Vols + Remainder
CSGPrim	bbox, refs sequence of *CSGNode*, root of CSG Tree of nodes	root *G4VSolid*
CSGNode	CSG node parameters (JUNO: ~23k *CSGNode*)	node *G4VSolid*

NVIDIA OptiX 7/8 Geometry Acceleration Structures (JUNO: 1 IAS + 10 GAS, 2-level hierarchy)

IAS	Instance Acceleration Structures	JUNO: 1 IAS created from vector of ~50k *qat4* (JUNO)
GAS	Geometry Acceleration Structures	JUNO: 10 GAS created from 10 *CSGSolid* (which refs *CSGPrim,CSGNode*)

JUNO : Geant4 ~400k volumes "factorized" into 1 OptiX IAS referencing ~10 GAS

Full JUNO, Opticks, OptiX 7.5/8.0

- mostly analytic CSG
- few complex solids (eg tori) : triangulated

raytrace 2M pixels	
TITAN RTX (1st)	0.0118s (85 fps)
Ada 5000 RTX (3rd)	0.0031s (323 fps)

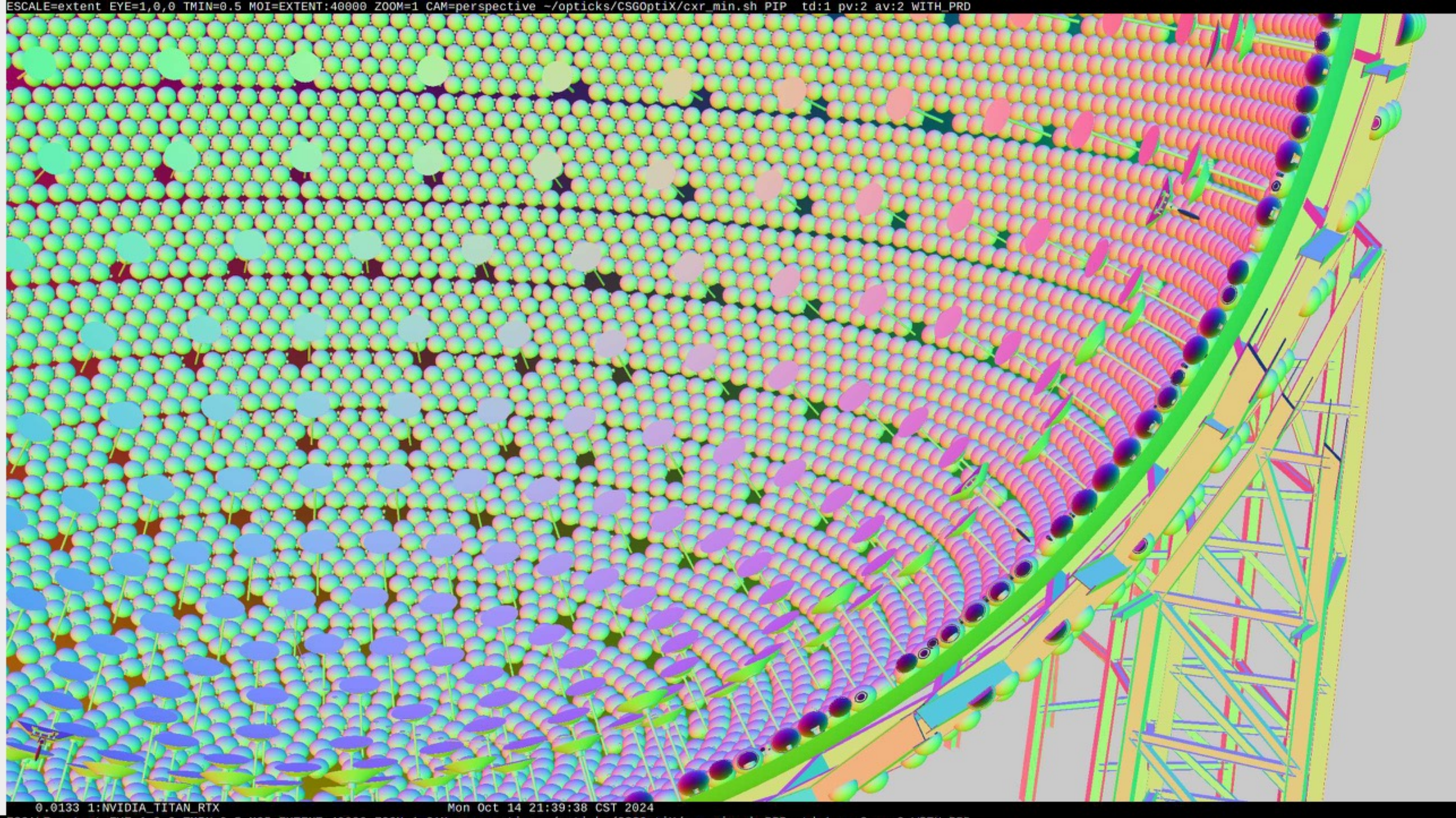
- **1st -> 3rd gen RTX : ~4x**

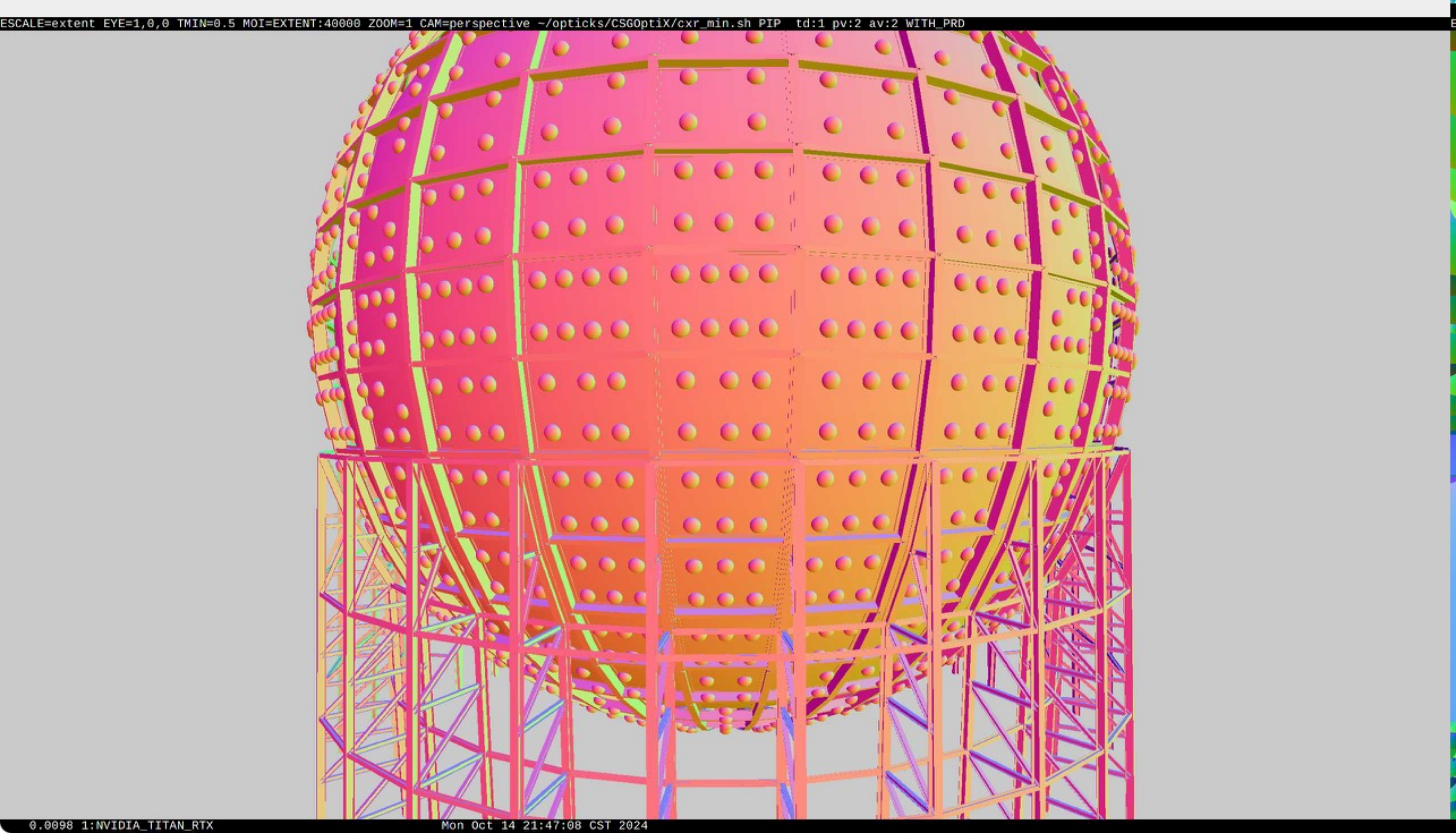
Analytic + triangulated geometry

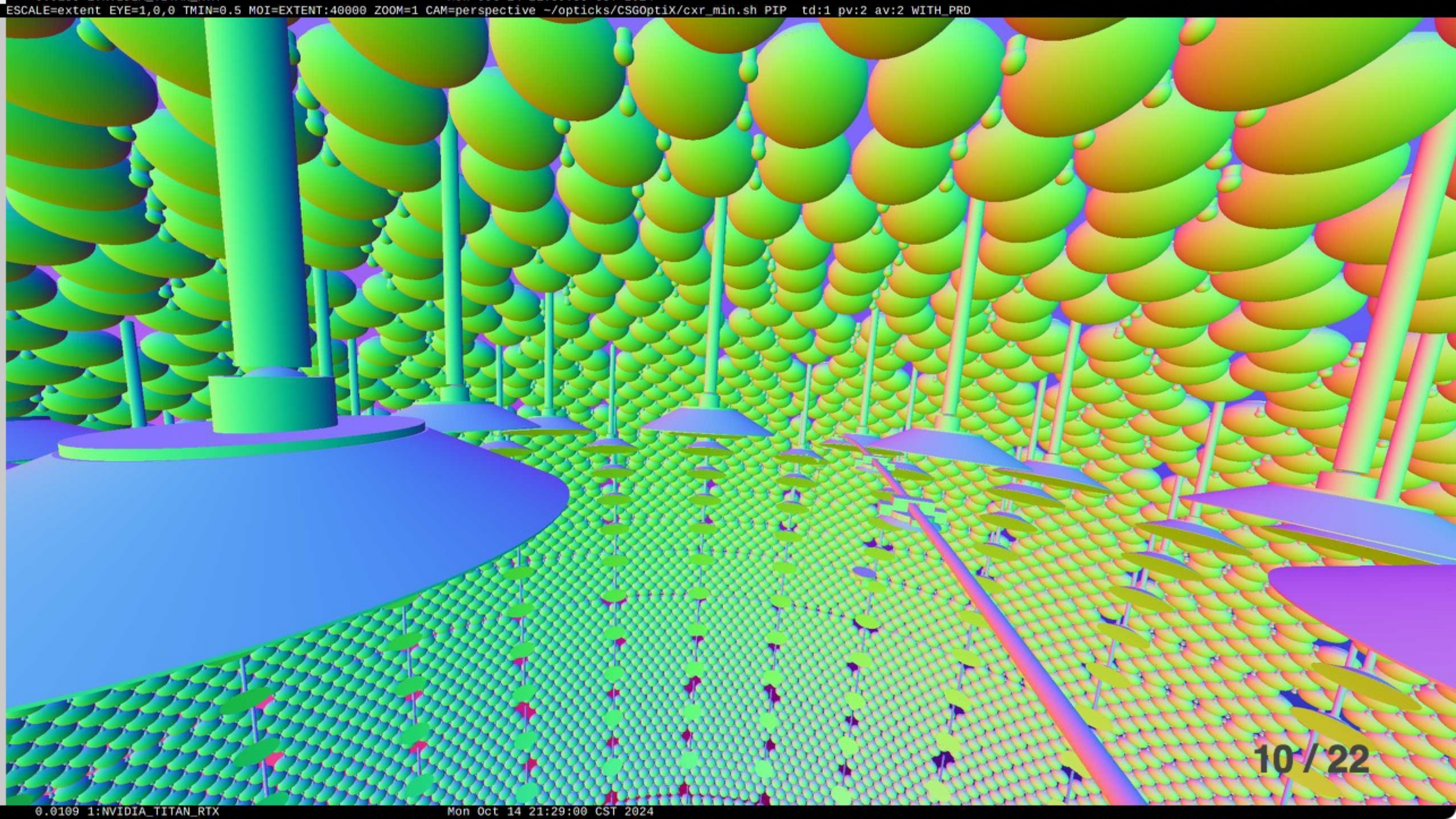
- default : analytic CSG solids
- user can name solids for triangulation
  - avoids issue with toruses + complex solids
  - BUT : approximate geometry
  - triangulation from G4Polyhedron
  - config per-solid NumberOfRotationSteps by envvars
  - uses OptiX "built-in" triangle intersection

NEW FEATURE

Integration of analytic + triangulated geometry







Interactive ray traced visualization via OpenGL/OptiX interop

initial viewpoint, geometry exclusions via envvars

WASDQE+mouse 3D navigation


ESCALE=extent EYE=1,0,0 TMIN=0.5 MOI=sSurftube_0V1_0:0:-1 ZOOM=1 CAM=perspective ~/opticks/CSGOptiX/cxr_min.sh PIP td:1 pv:2 av:2 WITH_PRD
Render on NVIDIA RTX 5000 Ada Generation in 0.0060 s (not 0.0200 s)
0.0060 0:NVIDIA_RTX_5000_Ada_Generation
Thu Oct 17 02:34:04 PM CST 2024

GuideTube : Torus Triangulated

GuideTube (39*2*2 = 156 G4Torus)
split in phi segments, radius breaks

Intersect with torus expensive on GPU

- requires double precision to solve quartic
- even with double precision analytic solution imprecise
- **numerical approach favored** => **triangulation**

Triangulation using **G4Polyhedron**

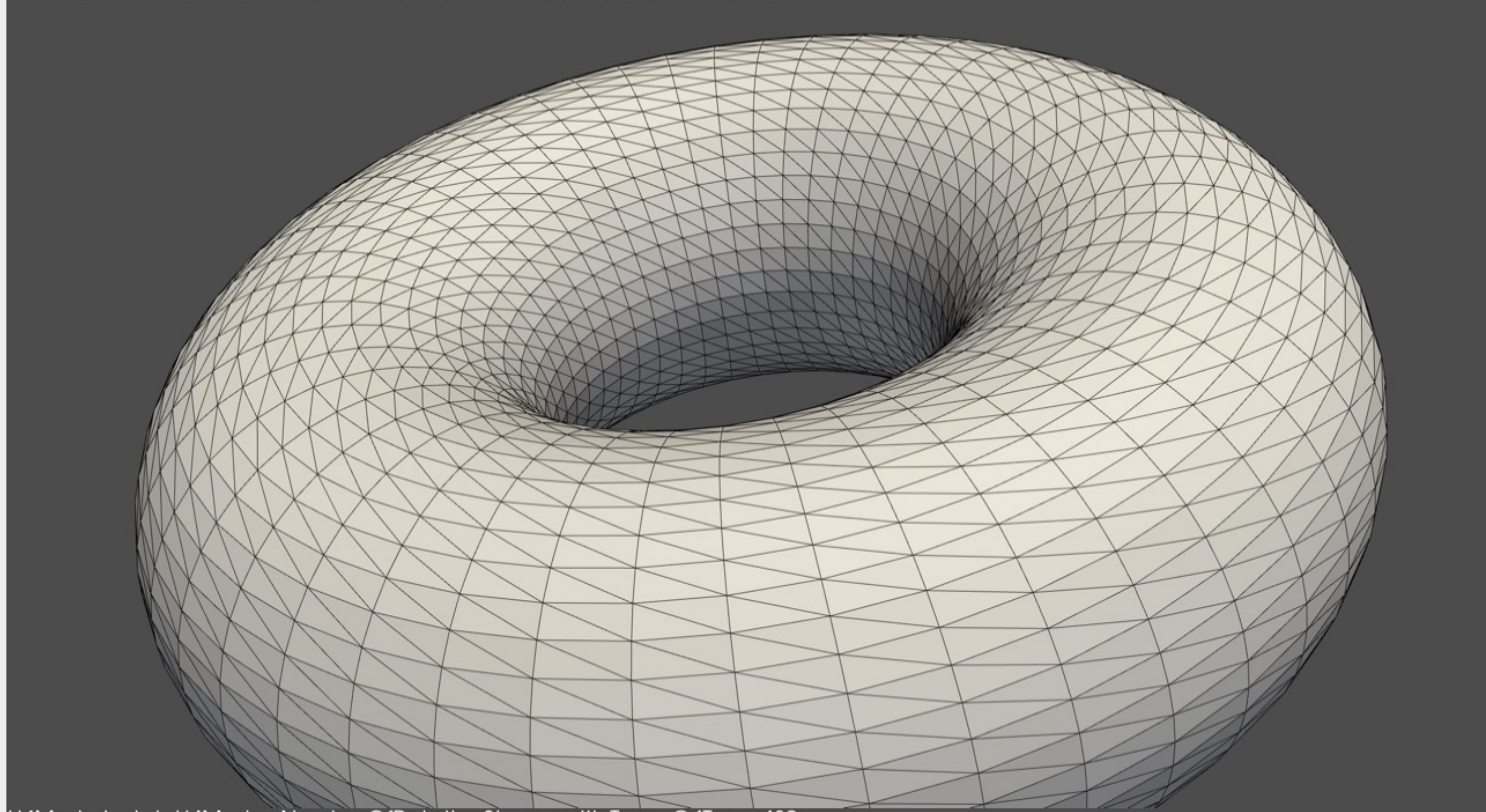
```
G4Poly..::SetNumberOfRotationSteps
```

	NumberOfRotationSteps
HepPolyhedron Default	24
Top Right	48
Bottom Right	480

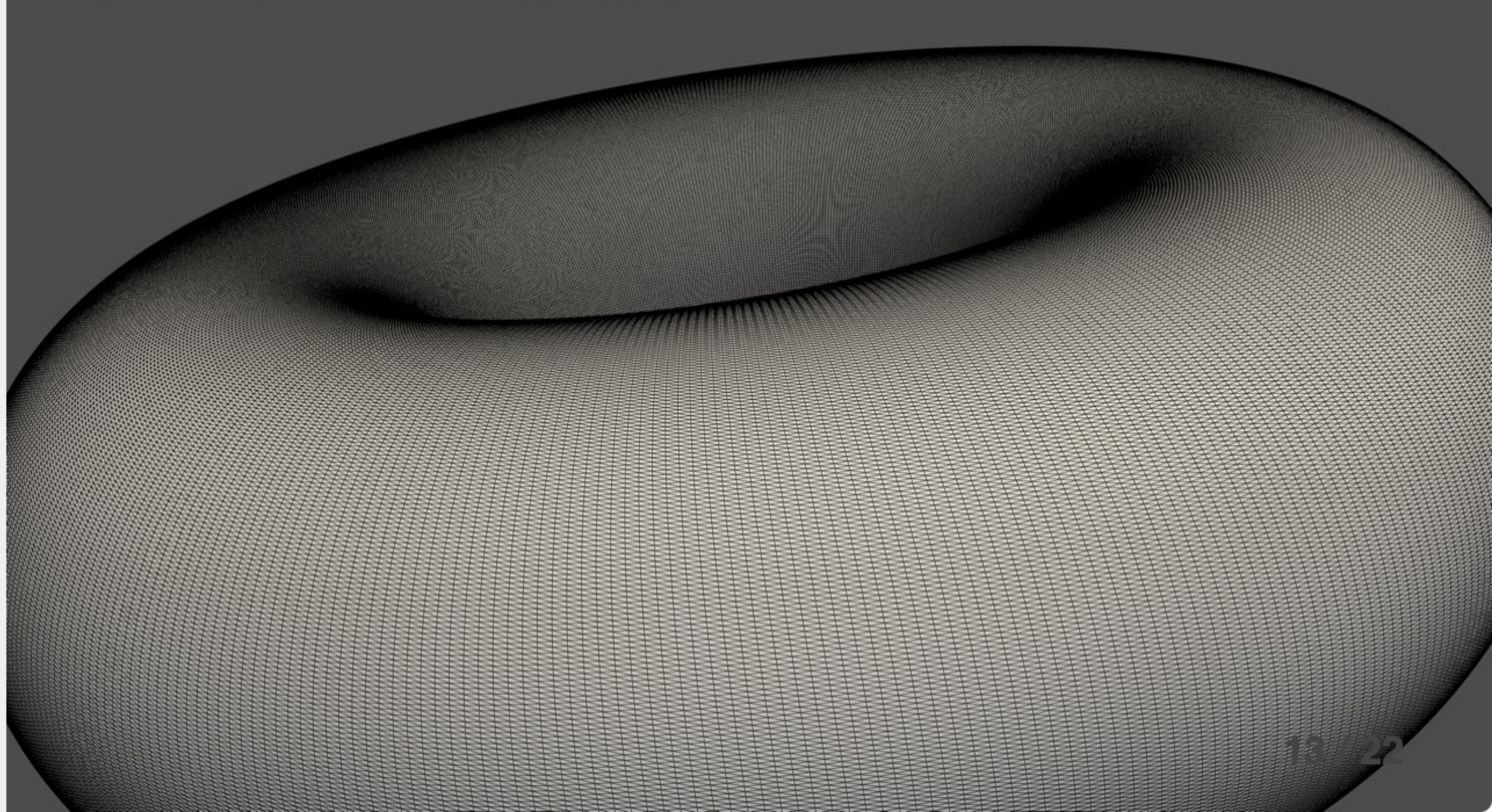
Adjustable: precision of intersect, number of triangles

GPUs evolved for triangles => fast even with many

U4Mesh_test.sh U4Mesh__NumberOfRotationSteps_entityType_G4Torus 48



U4Mesh_test.sh U4Mesh__NumberOfRotationSteps_entityType_G4Torus 480



List-node avoids deep CSG trees

Problematic deep CSG tree without list-node

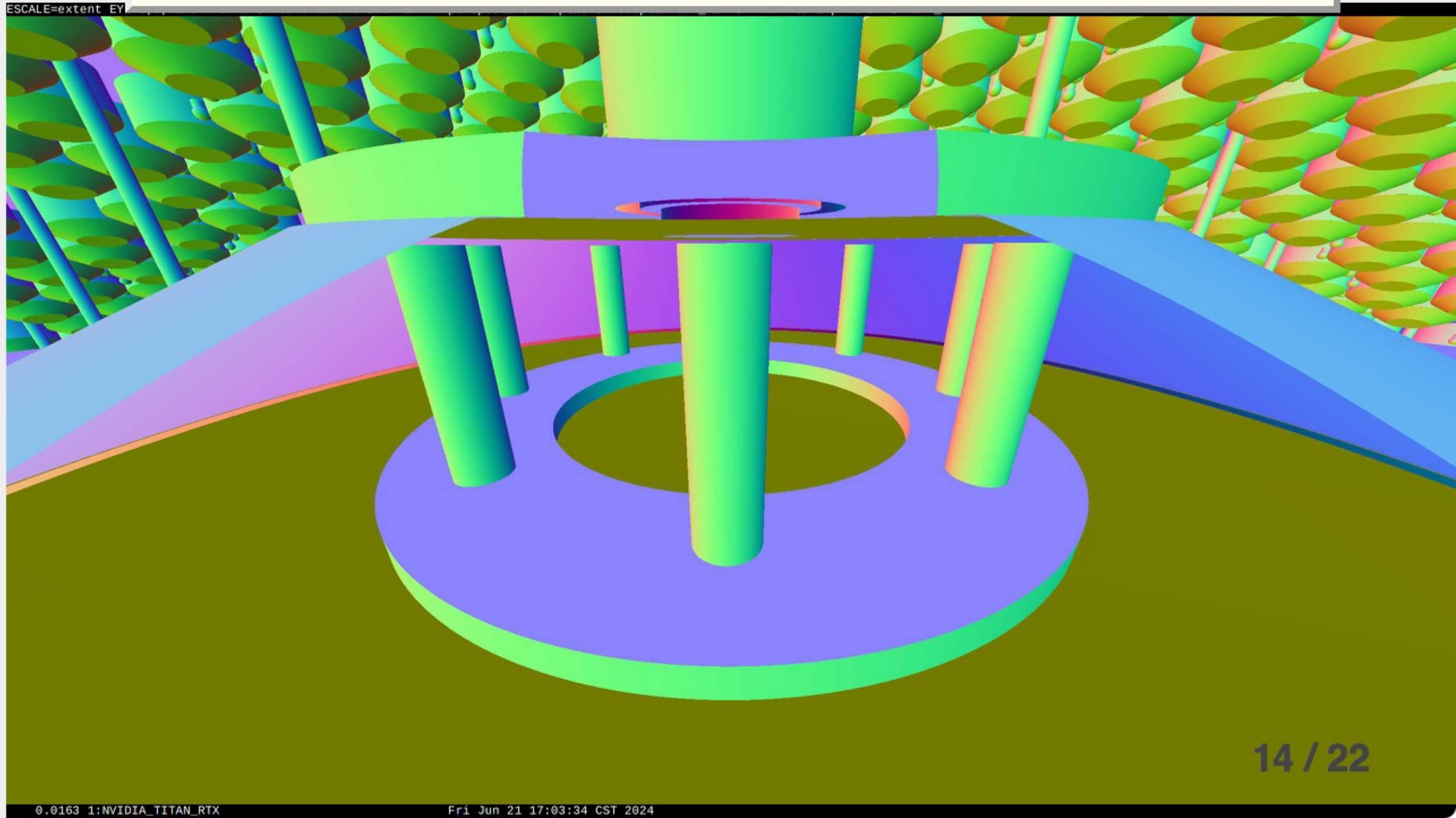
```
+------------------------------------------+
|                                          |
|                                          |
|                     U                    |
|                    / \                   |
|                   /   \                  |
|                  /     S                 |
|                 U     / \                |
|                / \   I   J               |
|               U   H                      |
|              / \                         |
|             U   G                        |
|            / \                           |
|           U   F                          |
|          / \                             |
|         U   E                            |
|        / \                               |
|       U   D                              |
|      / \                                 |
|     U   C                                |
|    / \                                   |
|   A   B                                  |
|                                          |
+------------------------------------------+
```

```
U : Union
S : Subtraction
A-J : Tubs (cylinder) primitive
```

Simple G4MultiUnion is translated to Opticks list-node

With list-node : shrink CSG tree

```
+------------------------------+
|         U                    |
|        / \                   |
|       /   \                  |
|      S     U[A,B,C,D,E,F,G,H]|
|     / \                      |
|    I   J                     |
+------------------------------+
```



Pure Optical TorchGenstep scan : 1M to 100M photons

```
TEST=medium_scan ~/opticks/cxs_min.sh
```

Generate optical only events with 1M->100M photons starting from CD center, gather and save only Hits.

```
OPTICKS_RUNNING_MODE=SRM_TORCH  ## "Torch" running enables num_photon scan
OPTICKS_NUM_PHOTON=M1,10,20,30,40,50,60,70,80,90,100
OPTICKS_NUM_EVENT=11
OPTICKS_EVENT_MODE=Hit
```

- uses CSGOptiXSMTest executable (no Geant4 dependency, avoids ~150s of initialization time)
- load and upload geometry in ~2s

Compare simulation scans on two Dell Precision Workstations:

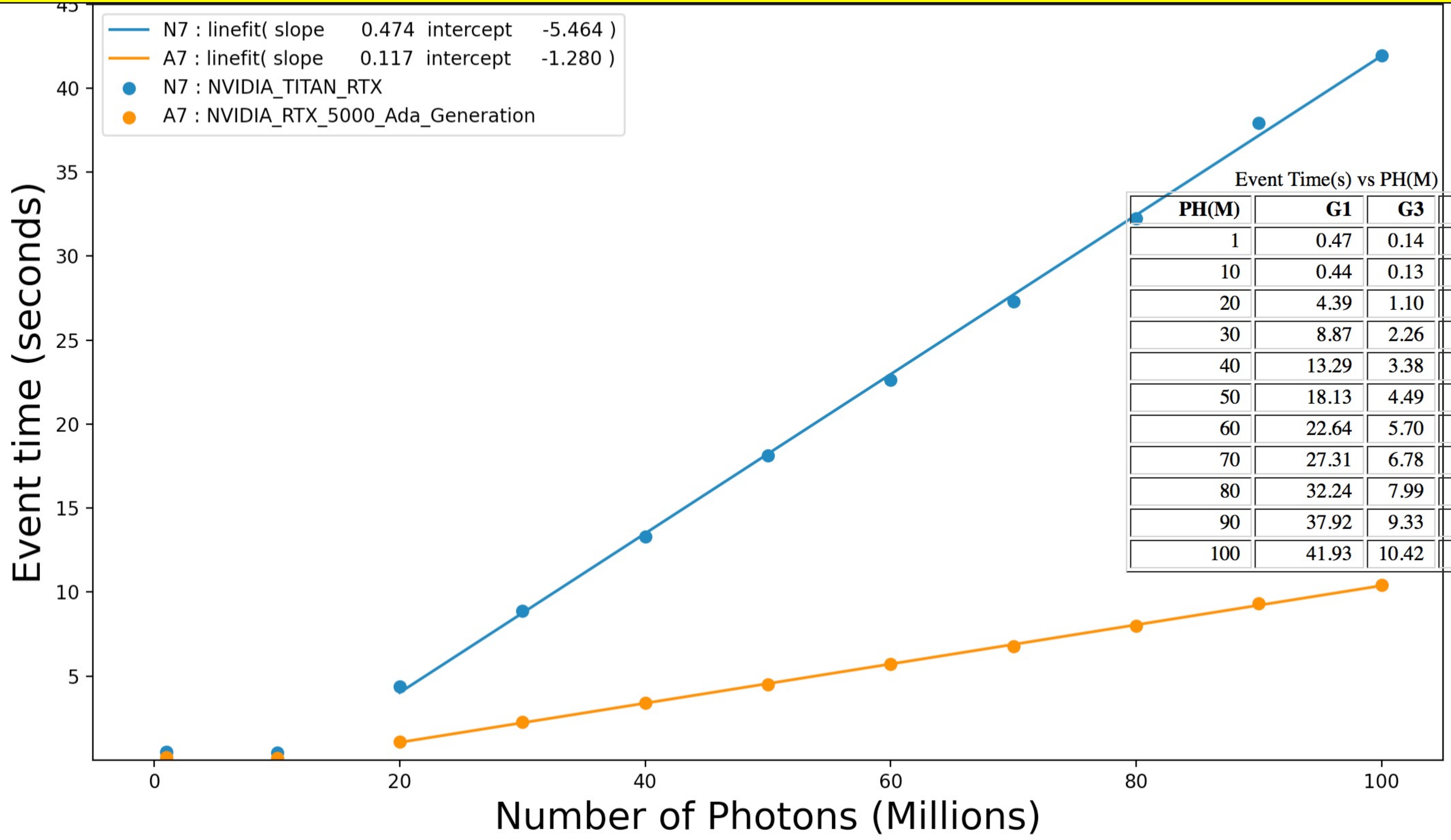
GPU (VRAM)	Arch	GPU Release	CUDA(RT) Cores	RTX Gen	Driver	CUDA	OptiX
NVIDIA TITAN RTX(24G)	Turing	Dec 2018	4,608(72)	1st	515.43	11.7	7.5
NVIDIA RTX 5000(32G)	Ada	Aug 2023	12,800(100)	3rd	550.76	12.4	8.0

- max launch size : 24/32/48G VRAM ~200/266/400M photons


/data/blyth/opticks/GEOM/J23_1_0_rc3_ok0/CSGOptiXSMTest/ALL1/A000 : found hit
4.5M hits from 20M photon TorchGenstep, 4.4(1.1) seconds
with: NVIDIA TITAN RTX(NVIDIA RTX 5000 Ada) 1st(3rd) gen RTX

```
A=N7 B=A7 PLOT=AB_Substamp_ALL_Etime_vs_Photon ~/o/sreport_ab.sh ## cxs_min.sh
RUNNING_MODE:SRM_TORCH EVENT_MODE:Hit MAX_BOUNCE:31 MAX_PHOTON:M100
```

Optical simulation 4x faster 1st->3rd gen RTX, (3rd gen, Ada : 100M photons simulated in 10 seconds) [TMM PMT model]



Event Time(s) vs PH(M)

PH(M)	G1	G3	G1/G3
1	0.47	0.14	3.28
10	0.44	0.13	3.48
20	4.39	1.10	3.99
30	8.87	2.26	3.93
40	13.29	3.38	3.93
50	18.13	4.49	4.03
60	22.64	5.70	3.97
70	27.31	6.78	4.03
80	32.24	7.99	4.03
90	37.92	9.33	4.06
100	41.93	10.42	4.03

How much parallelized speedup actually useful to overall speedup?

optical photon simulation, P ~ 99% of CPU time

- => limit on overall speedup S(n) is 100x
- even with parallel speedup factor >> 1000x

Traditional simulation use:

- **speedup beyond 1000x not needed**

Amdahls "Law" : Expected Speedup

$$S(n) = \frac{1}{(1-P) + P/n}$$

Overall speed limited by serial portion

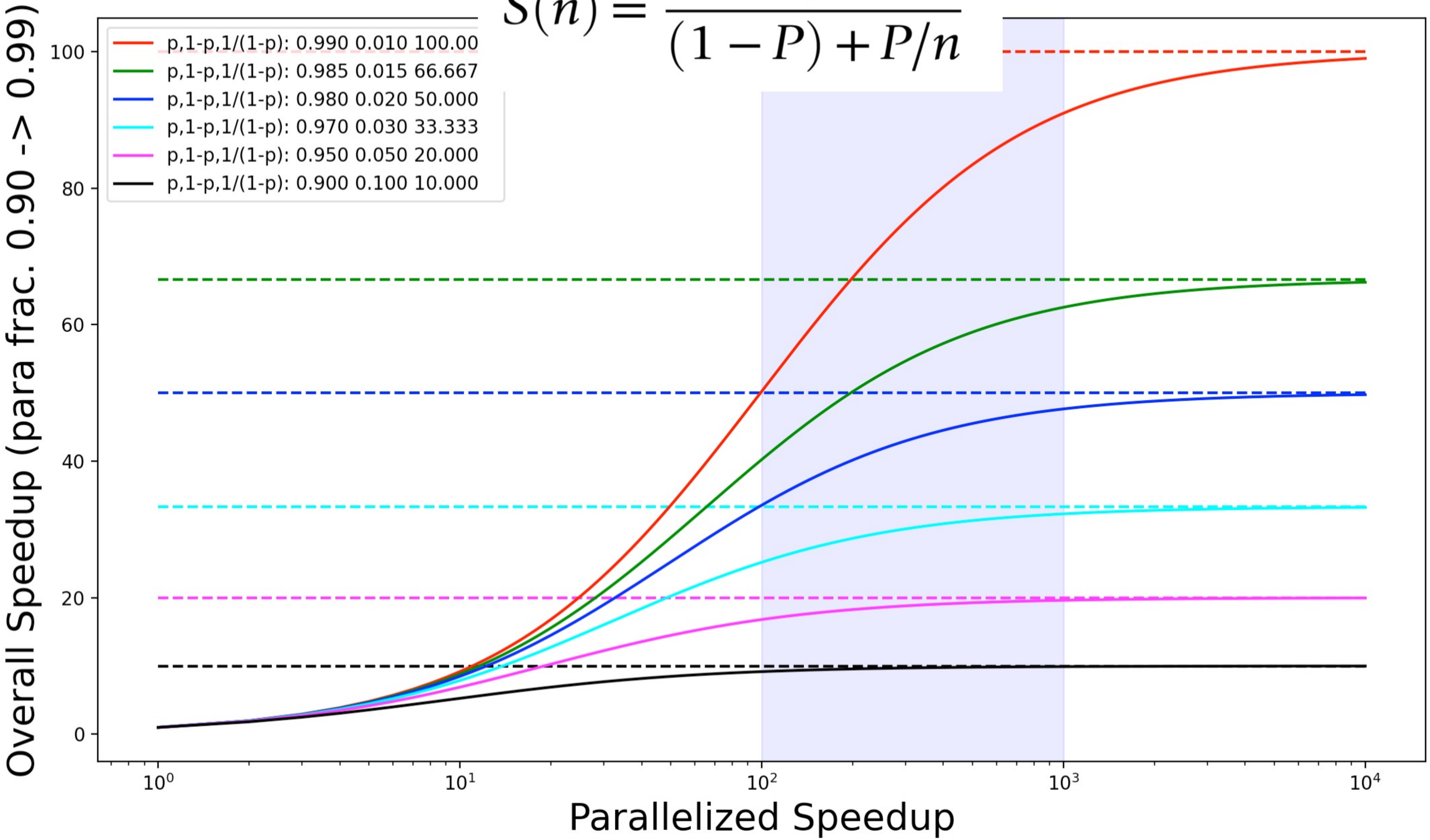
P
: parallelizable proportion

1-P
: non-parallelizable portion

n
: parallel speedup factor

~/opticks/ana/amdahl.sh : Overall Speedup for Parallel Fractions, p

$$S(n) = \frac{1}{(1-P) + P/n}$$



Summary and Links

Opticks : state-of-the-art GPU ray traced optical simulation integrated with *Geant4*, with automated geometry translation into GPU optimized form.

Extra Benefits of Adopting Opticks

- **high performance novel visualization**
- detailed photon instrumentation, validation
- comparisons find issues with both simulations:
  - complex geometry, overlaps, bugs...

=> using Opticks improves CPU simulation too !!

- NVIDIA Ray Trace Performance continues rapid progress (2x each gen., every ~2 yrs)
- **any simulation limited by optical photons can benefit from Opticks**
- more photon limited -> more overall speedup (99% -> ~90x)

https://bitbucket.org/simoncblyth/opticks	day-to-day code repository
https://simoncblyth.bitbucket.io	presentations and videos
https://groups.io/g/opticks	forum/mailing list archive
email: `opticks+subscribe@groups.io`	subscribe to mailing list
`simon.c.blyth@gmail.com`	any questions

Acknowledgements

- Opticks users
  - ~38 members of forum : https://groups.io/g/opticks ❏
  - **many thanks to active bug reporting users**
    - (especially from JUNO, LZ, LHAASO, LHCb-RICH, DUNE, NEXT-CRAB0)
- JUNO Collaboration
  - Tao Lin, Yuxiang Hu, ... (+ many more : changing geometry and physics models)
  - **forced Opticks to continuously improve**
- Geant4 collaboration
  - **especially Hans Wentzel, Fermilab Geant4 group, early adopter of Opticks**
  - guest invites to Okinawa, Wollongong meetings
- Dark Matter Search Community (XENON,LZ,DARWIN,..) : DANCE invite 2019
- Many NVIDIA Engineers:
  - NVIDIA GPU Technology Conferences (San Jose, Suzhou)
  - **seven dedicated meetings in 2021 : migrating to OptiX 7 API**
  - UK GPU Hackathon 2022

Ilker Parmaksiz, NEXT-CRAB0 Prototype

New active bug reporting Opticks user : Ilker Parmaksiz

- careful comparison : Data, Geant4, Opticks
- **Opticks 181x over Geant4**