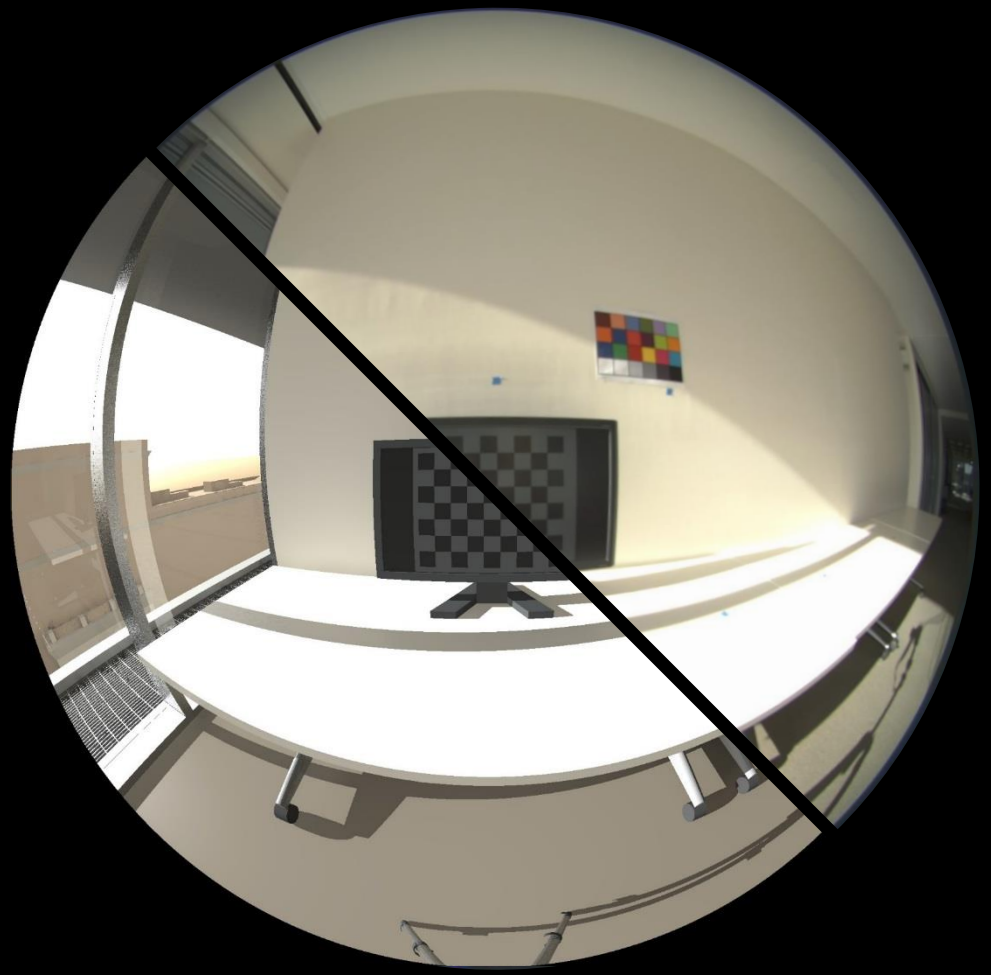


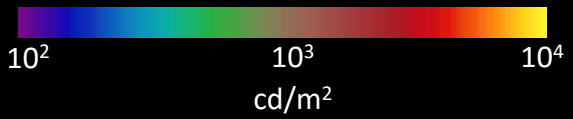
GPU-Enabled Lighting Simulation with Accelerad

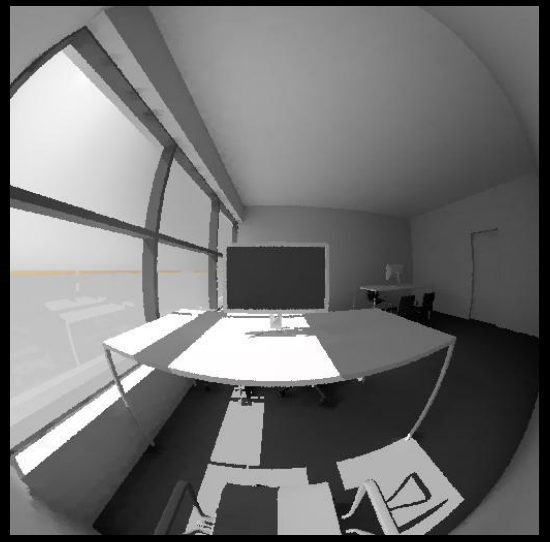
Nathaniel Jones and Christoph Reinhart

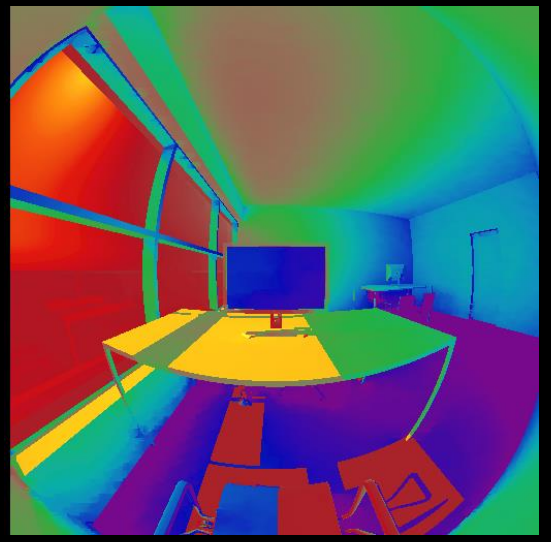
2015 International Radiance Workshop

Massachusetts Institute of Technology
Sustainable Design Lab

Speed vs Accuracy Trade-off

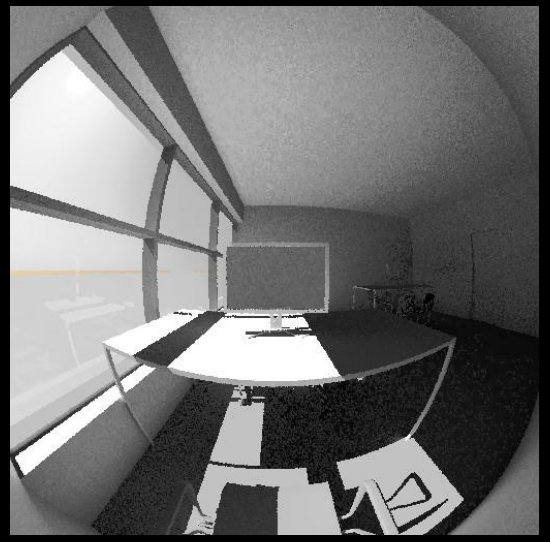


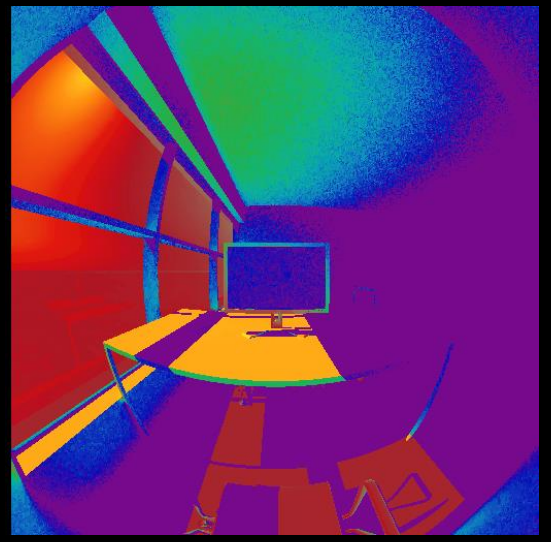




138,844,405 rays

49 minutes





41,010,721 rays

1.5 minutes

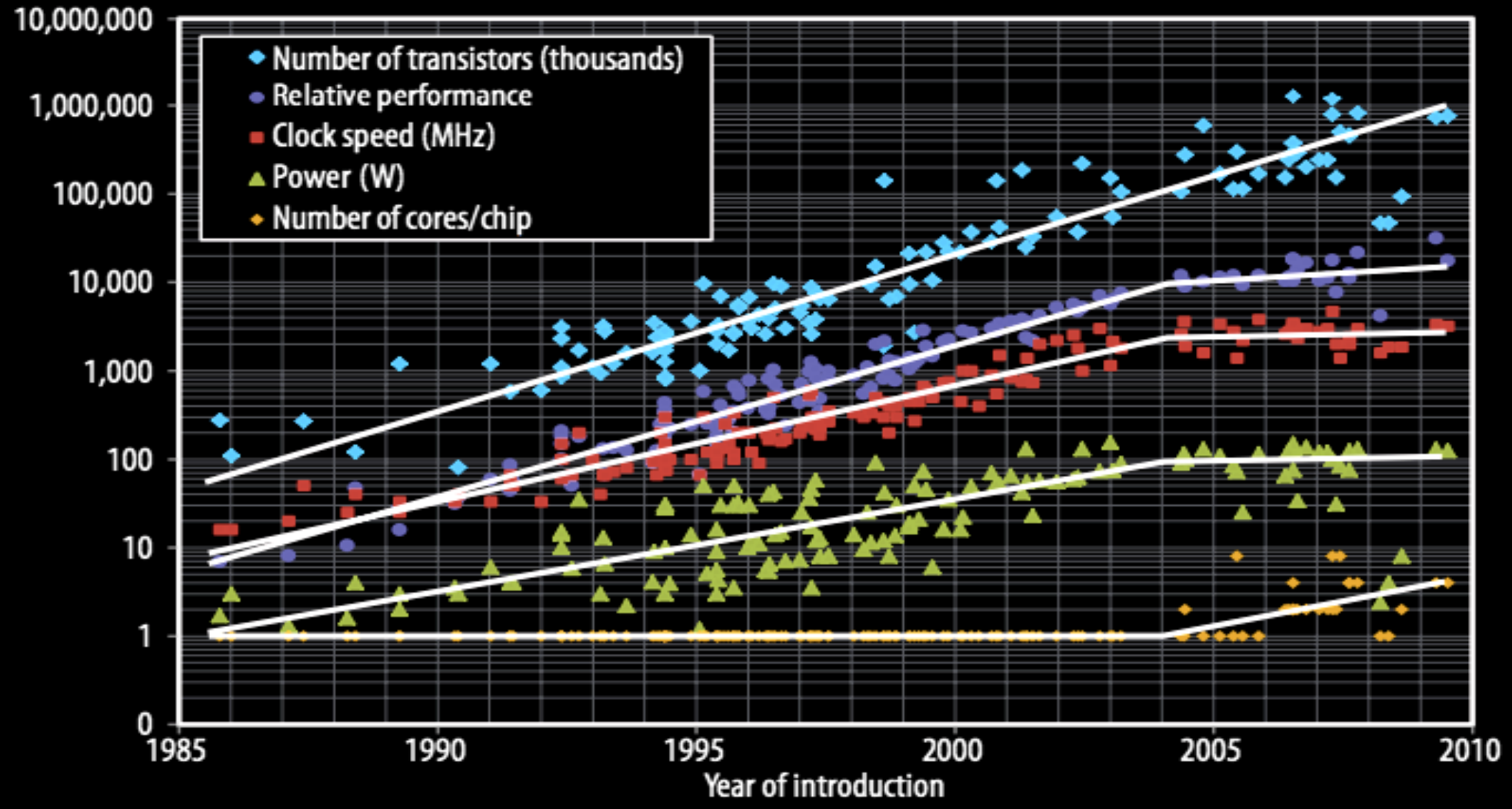
How Long Does It Take?

Point sensor	10^3 primary rays
Sensor grid	10^5
Glare prediction	10^6
Annual glare prediction	10^8
Adaptive glare prediction	10^{10}
Spatial glare mapping	10^{12}

seconds

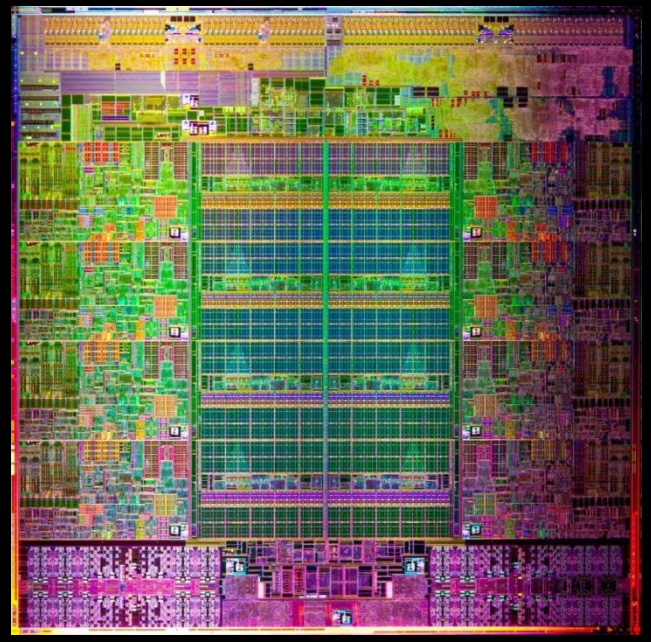
days

Moore's Law



Fuller, S.H. and Millett, L.I. Computing performance: Game over or next level? *Computer*, January 2011.

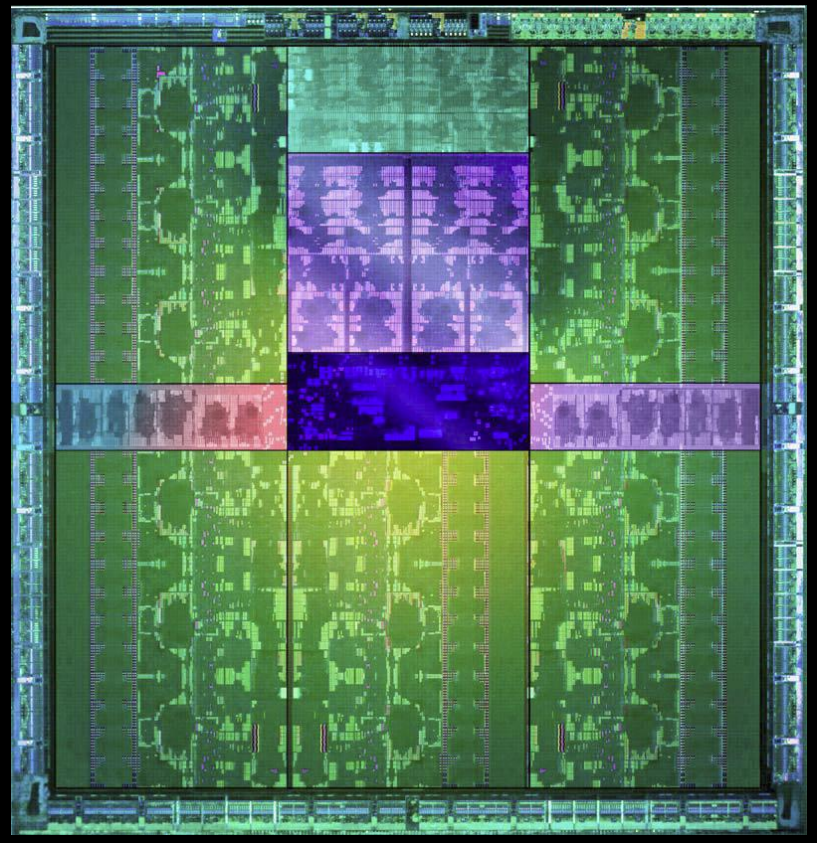
Multicore Processors



http://www.maximumpc.com/article/features/sandy_bridge-e_benchmarked_intel_retains_performance_crown

CPU

8 cores



http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf

GPU

2880 cores

Introducing Accelerad

Accelerad vs Radiance

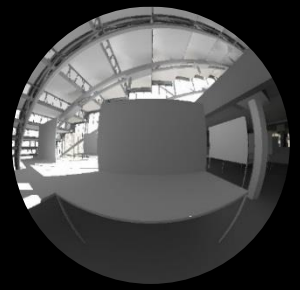
Setup and Installation

Tips and Tricks

New in Version 0.4 beta

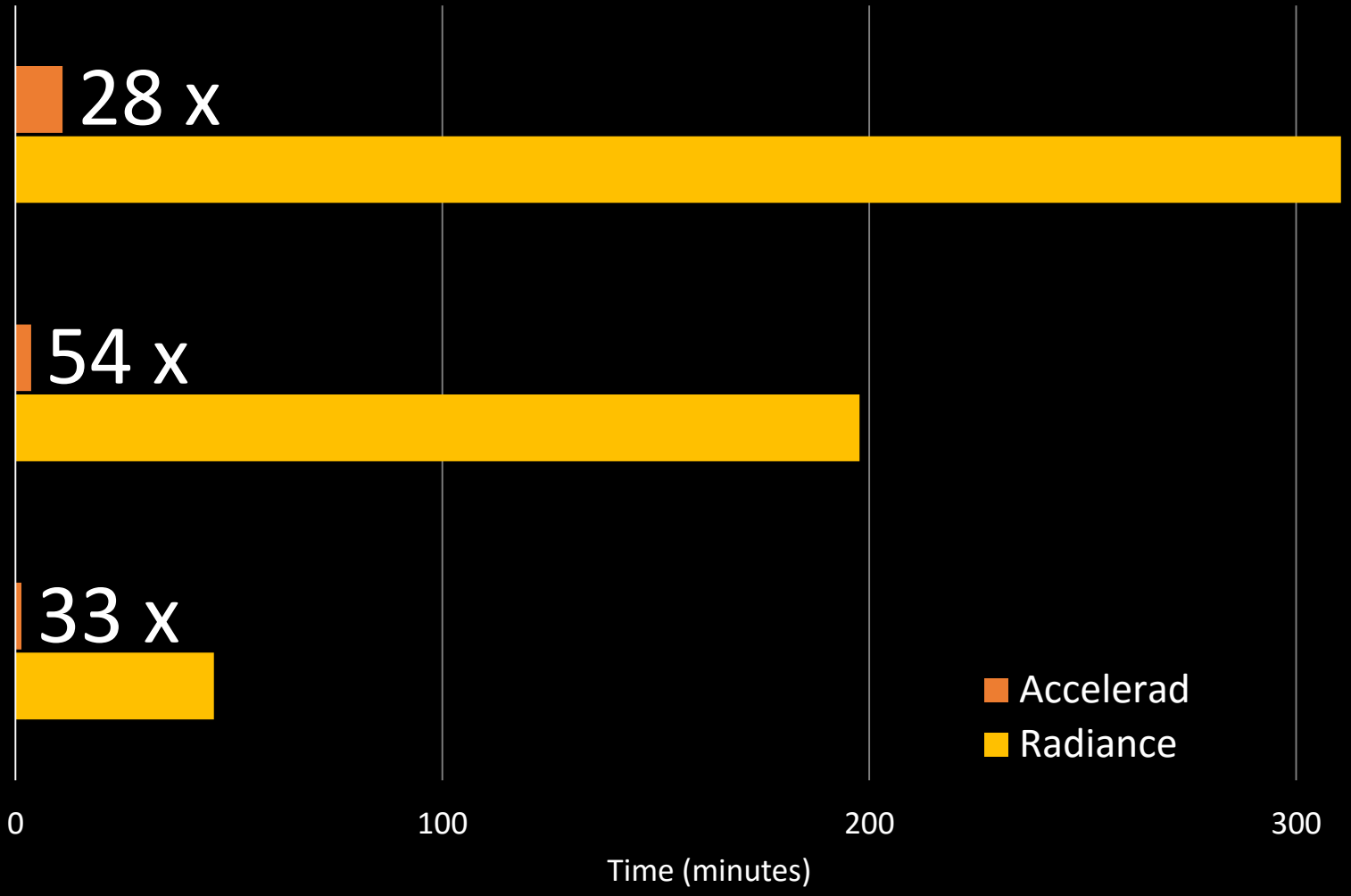
Speedup

Media Lab

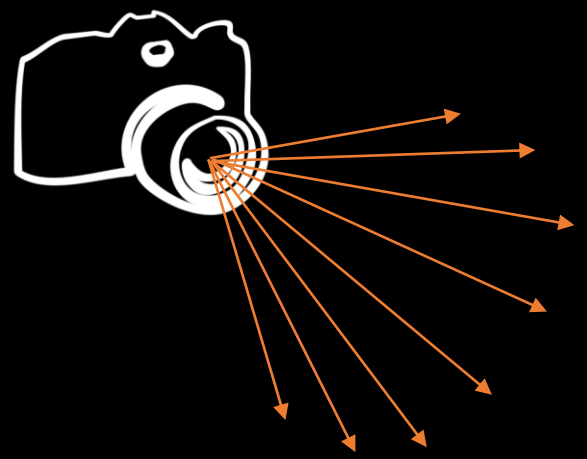


Gund Hall

Small Office

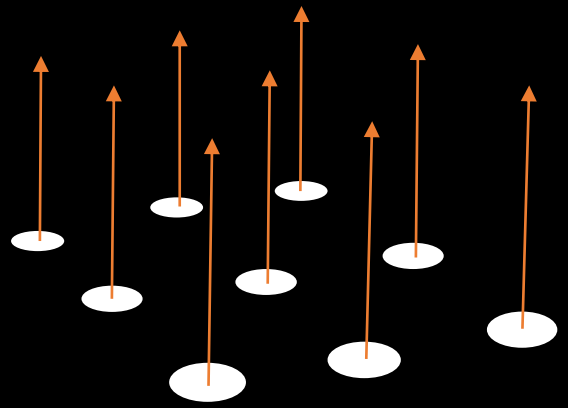


Parallelism



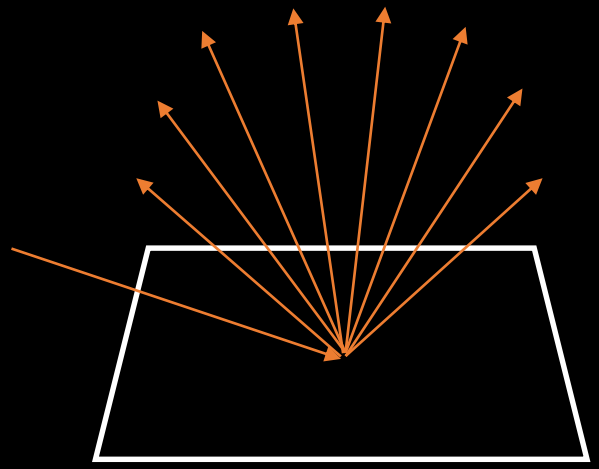
Camera

rpict



Sensors

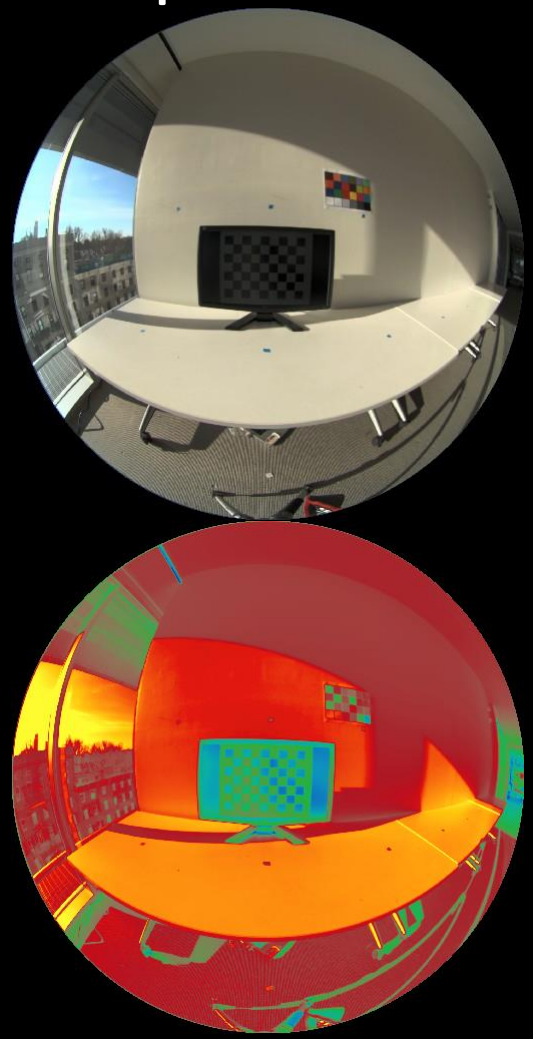
rtrace



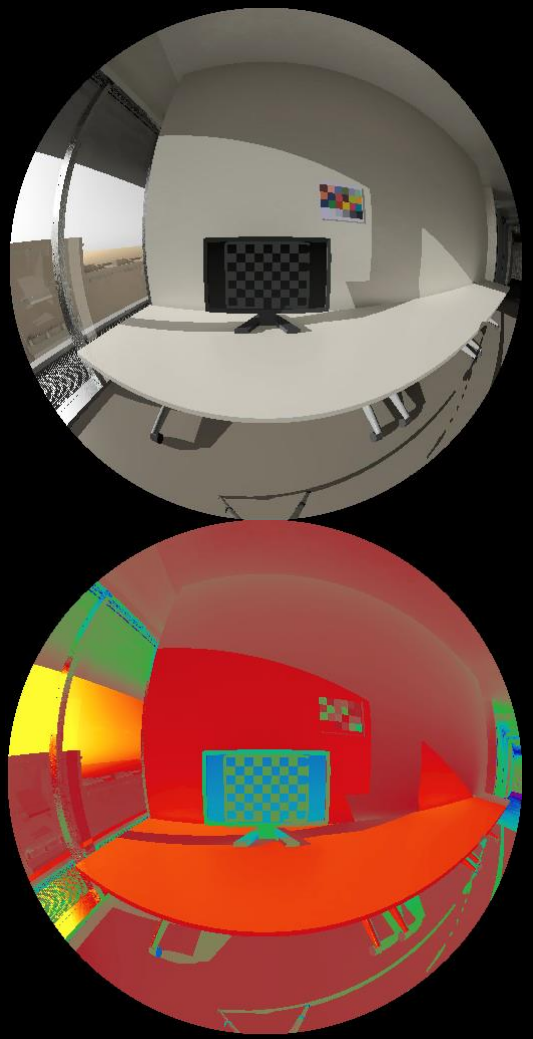
Ambient

preprocess

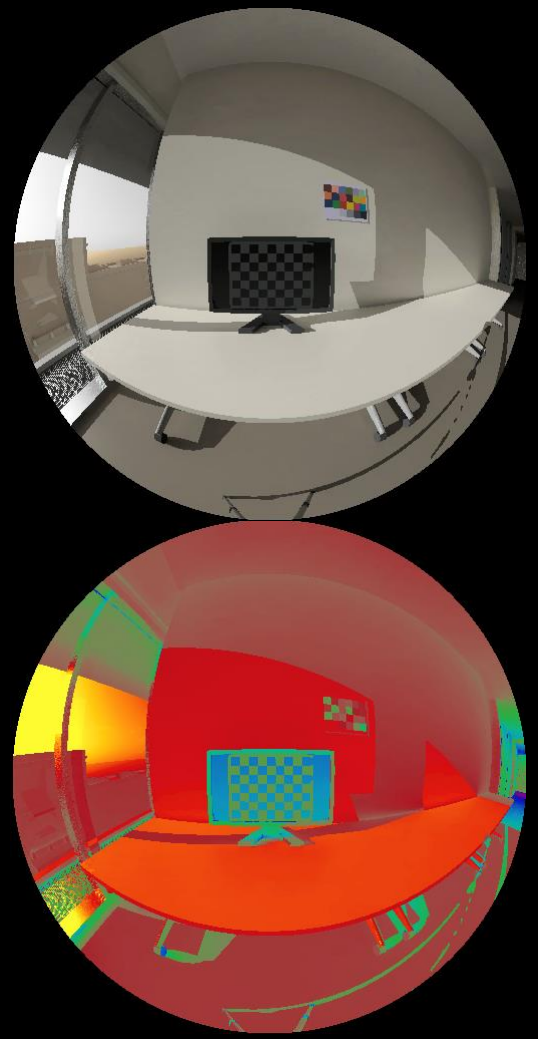
Comparison to HDR



HDR Photograph



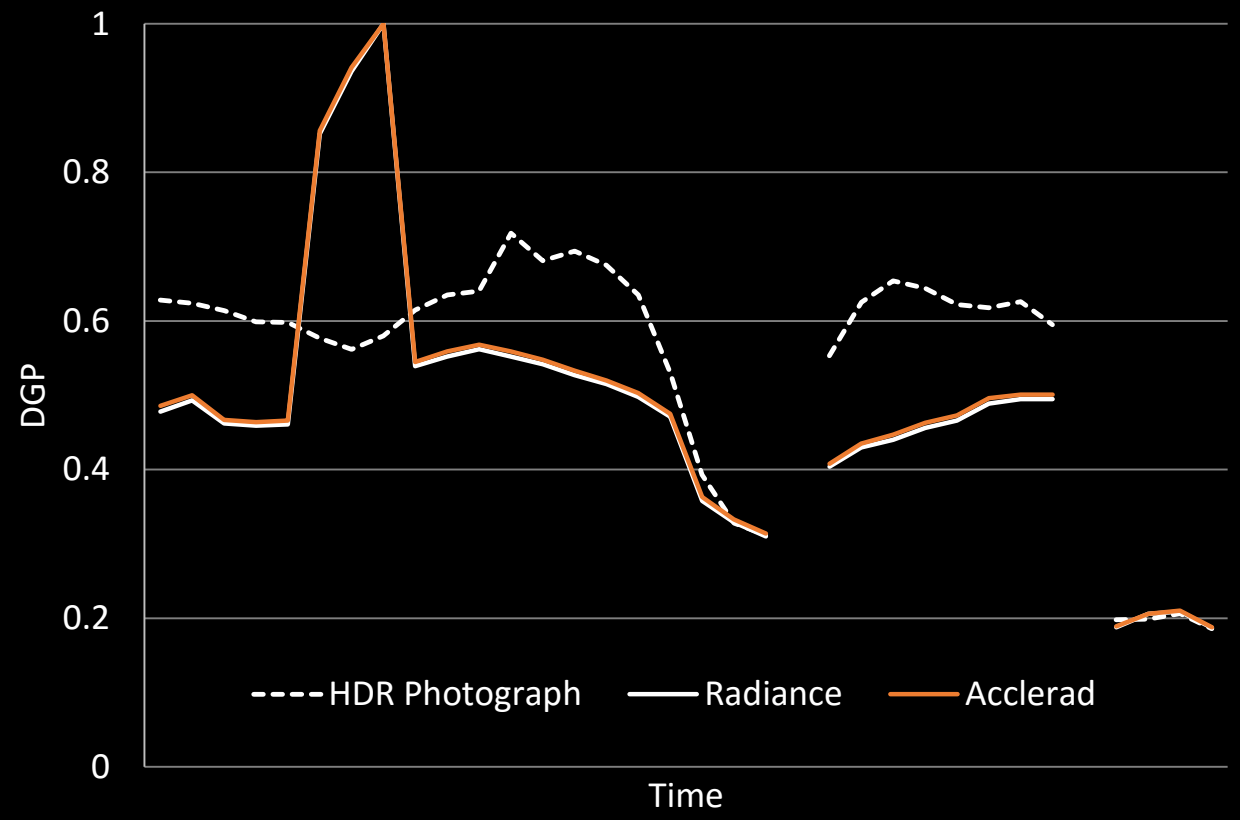
Accelerad
11 Minutes



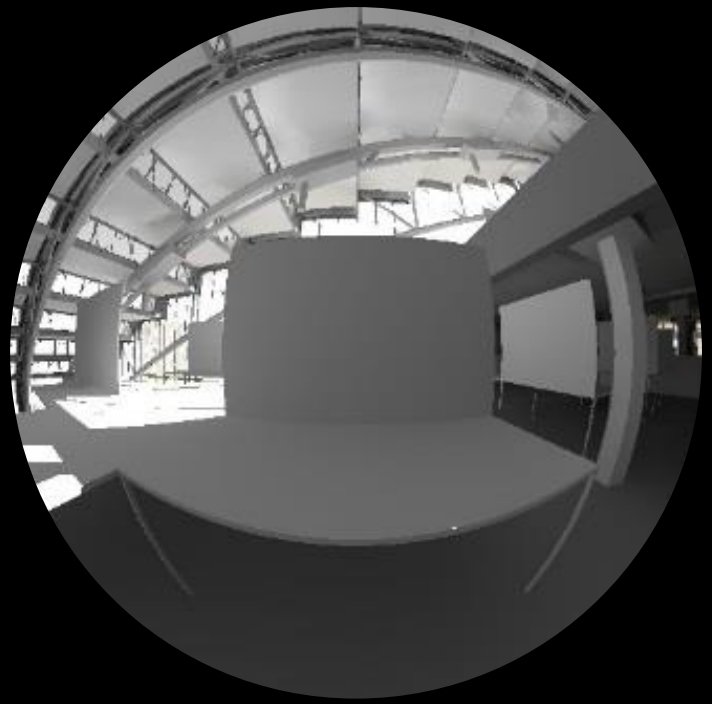
Radiance
303 Minutes

Visual Comfort Metrics

Daylight Glare Probability (DGP)



Jones and Reinhart, 2015. Validation of GPU lighting simulation in naturally and artificially lit spaces.
14th International Conference of the International Building Performance Simulation Association, accepted manuscript.



Introducing Accelerad

Accelerad vs Radiance

Setup and Installation

Tips and Tricks

New in Version 0.4 beta

Accelerad *is* Radiance

Same programs*

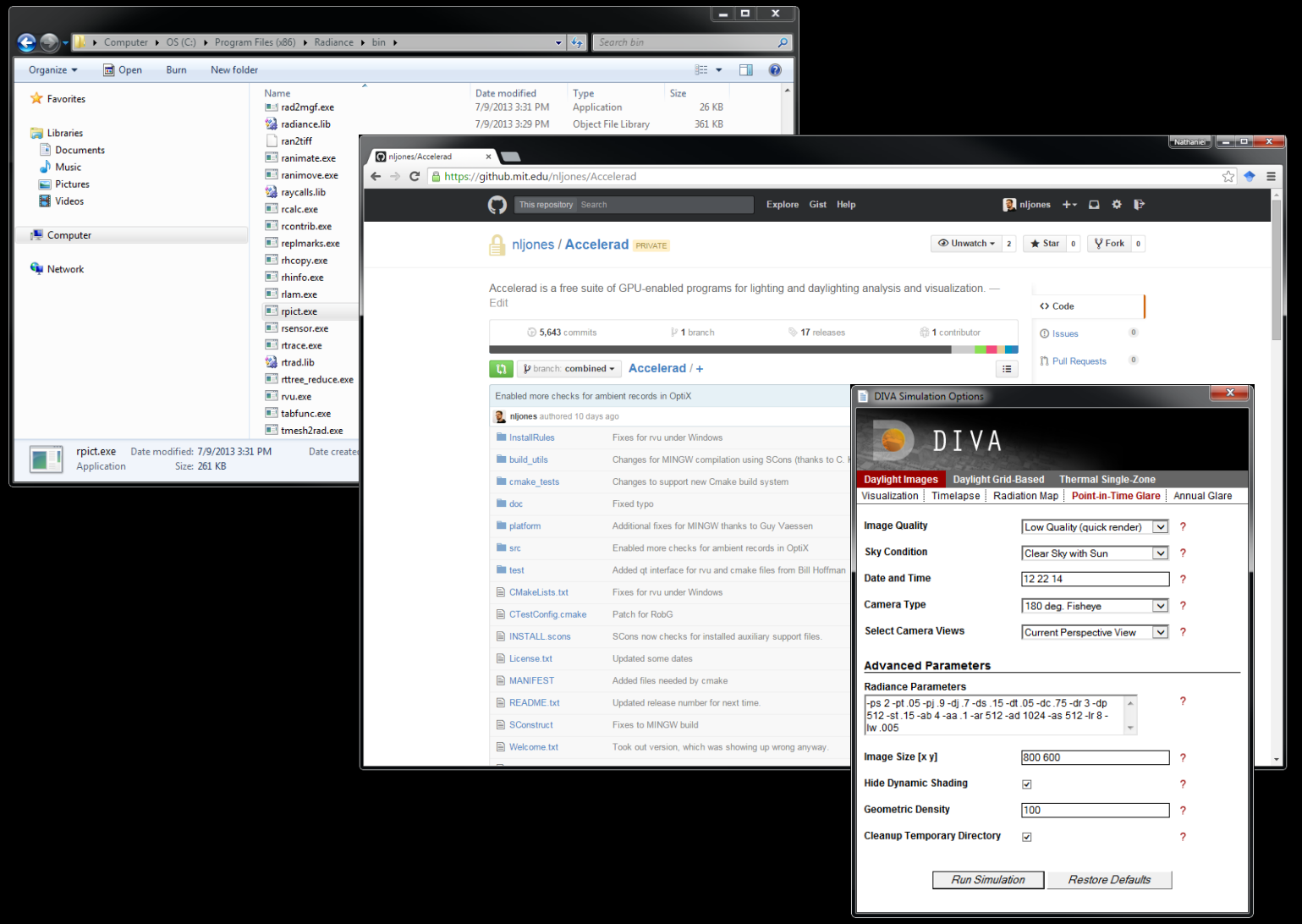
- Images with *rpict*
- Sensors with *rtrace*

Same source code*

- Fork of NREL's GitHub mirror repository
- Use *-g* 0 for Radiance behavior

Same interfaces

- Command line
- GUIs



* Modified

OptiX™

Parallel CUDA-based ray tracing library from NVIDIA®

Built-in ray traversal using BVH or *k*-d trees

User-defined shader programs

- Ray generation
- Intersection testing
- Closest hit
- Any hit
- Miss

Translated from Radiance source

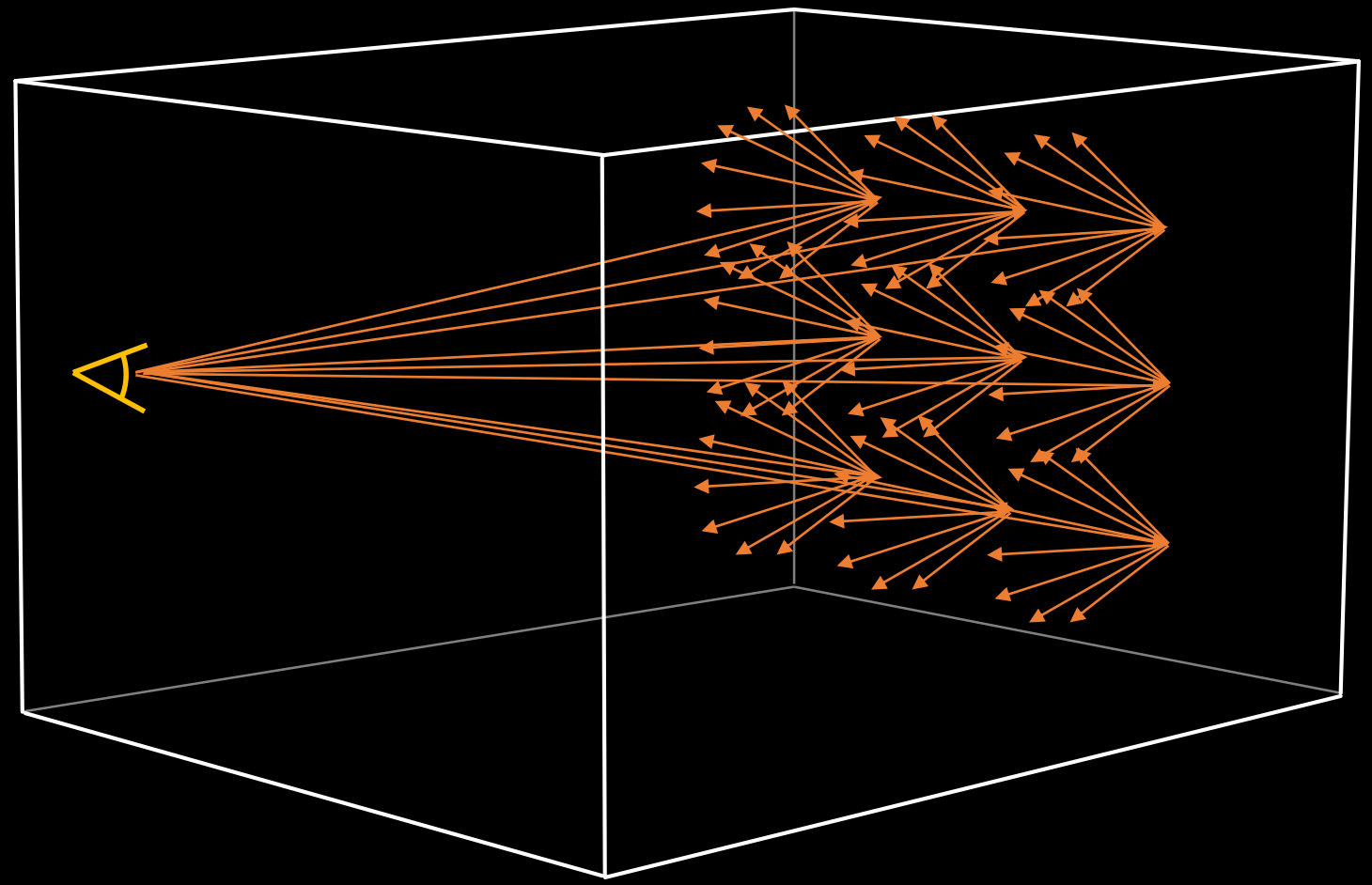
Radiance (C/C++)

```
if (rayorigin(&p, REFLECTED, r, refl) == 0)
{
    VSUM(p.rdir, r->rdir, pnorm, 2.*pdot);
    checknorm(p.rdir);
    rayvalue(&p);
    multcolor(p.rcol, p.rcoef);
    addcolor(r->rcol, p.rcol);
}
```

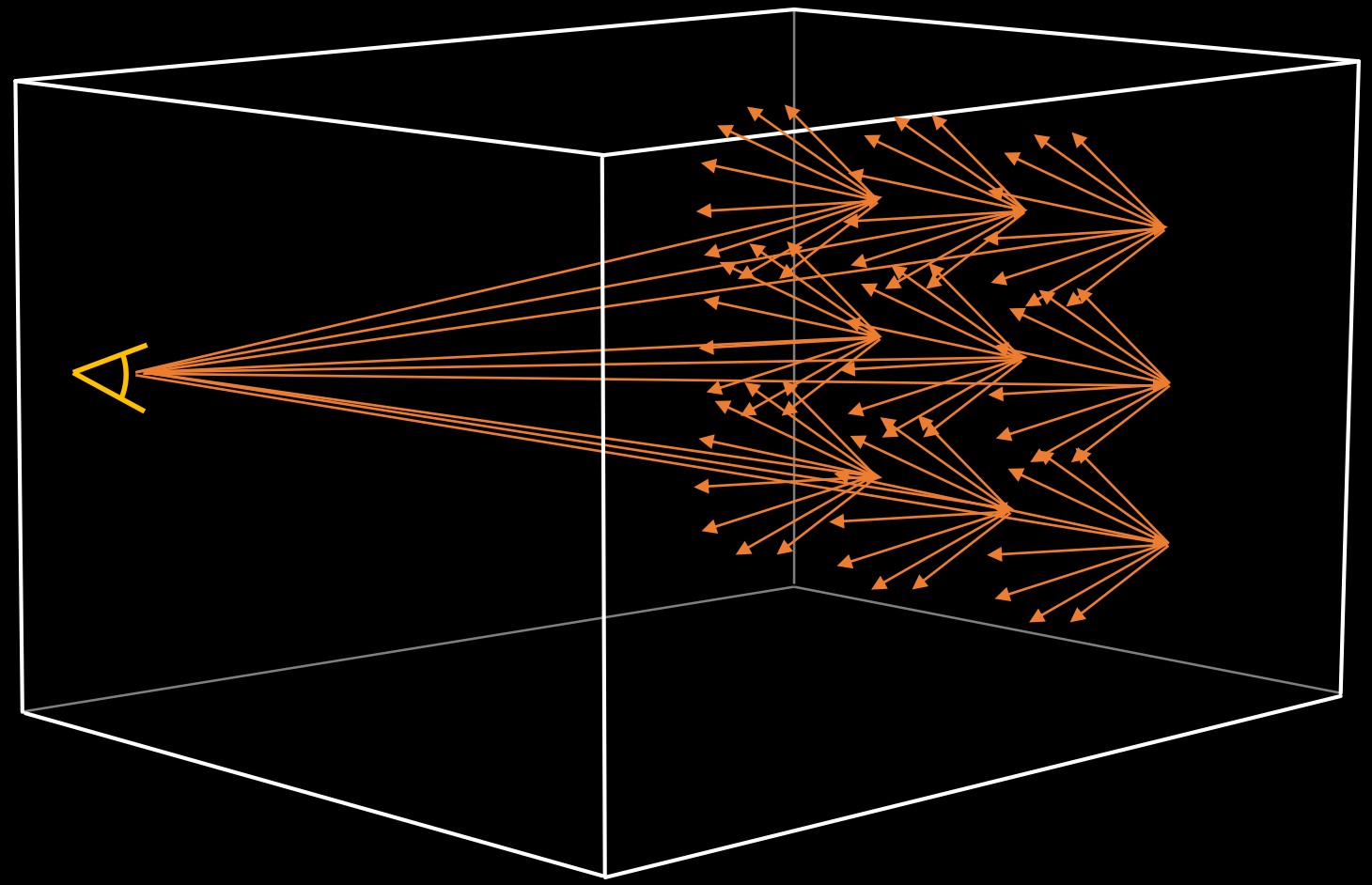
Accelerad (CUDA/OptiX)

```
if (prd.weight >= minweight &&
        prd.depth <= abs(maxdepth))
{
    float3 rdir = reflect(ray.dir, pnorm);
    Ray ray = make_Ray(hit_point, rdir,
         ray_type, RAY_START, RAY_END);
    rtTrace(top_object, ray, prd);
    result += prd.result * rcoef;
}
```

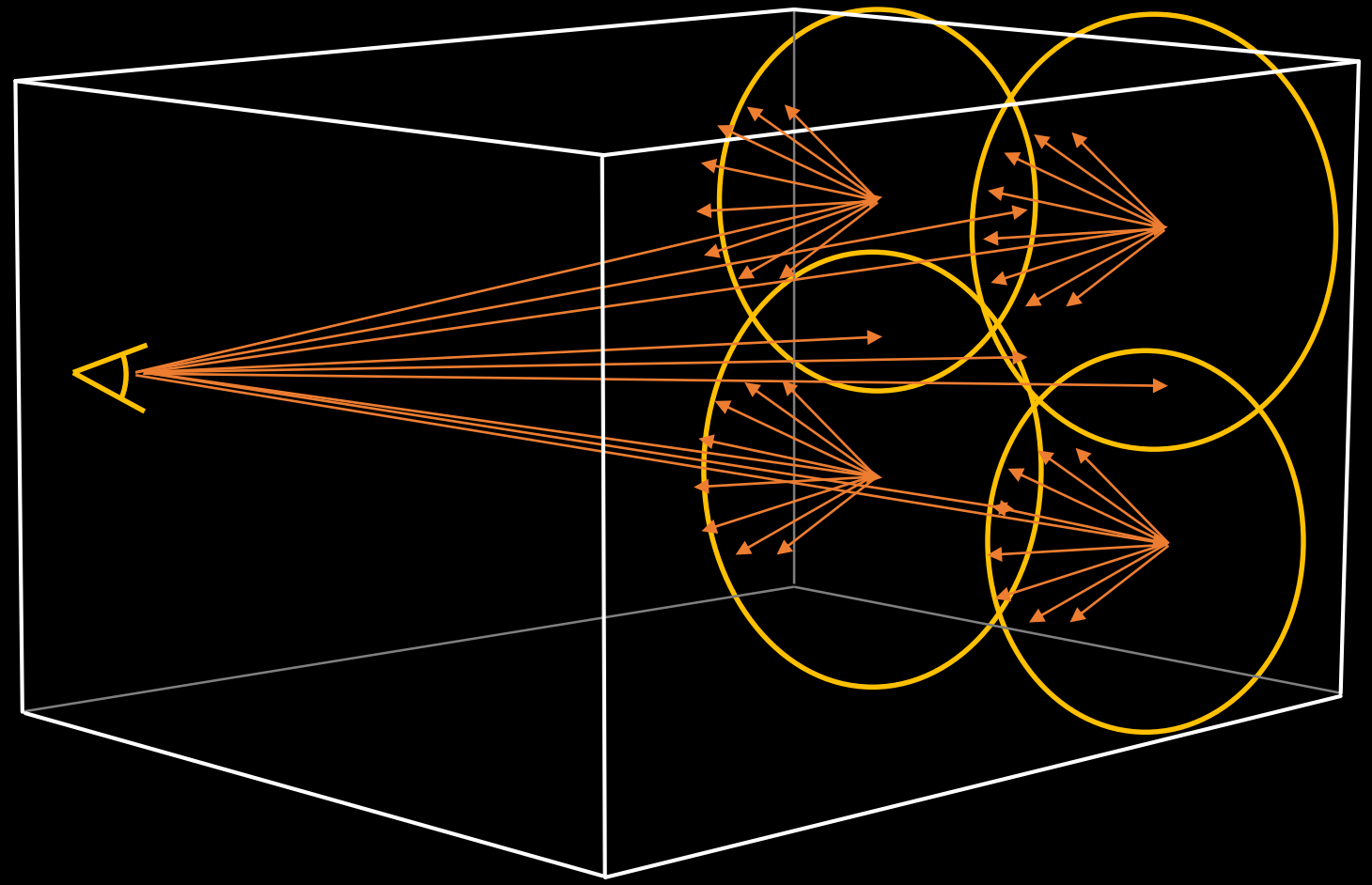
Whitted-Style Ray Tracing: CPU



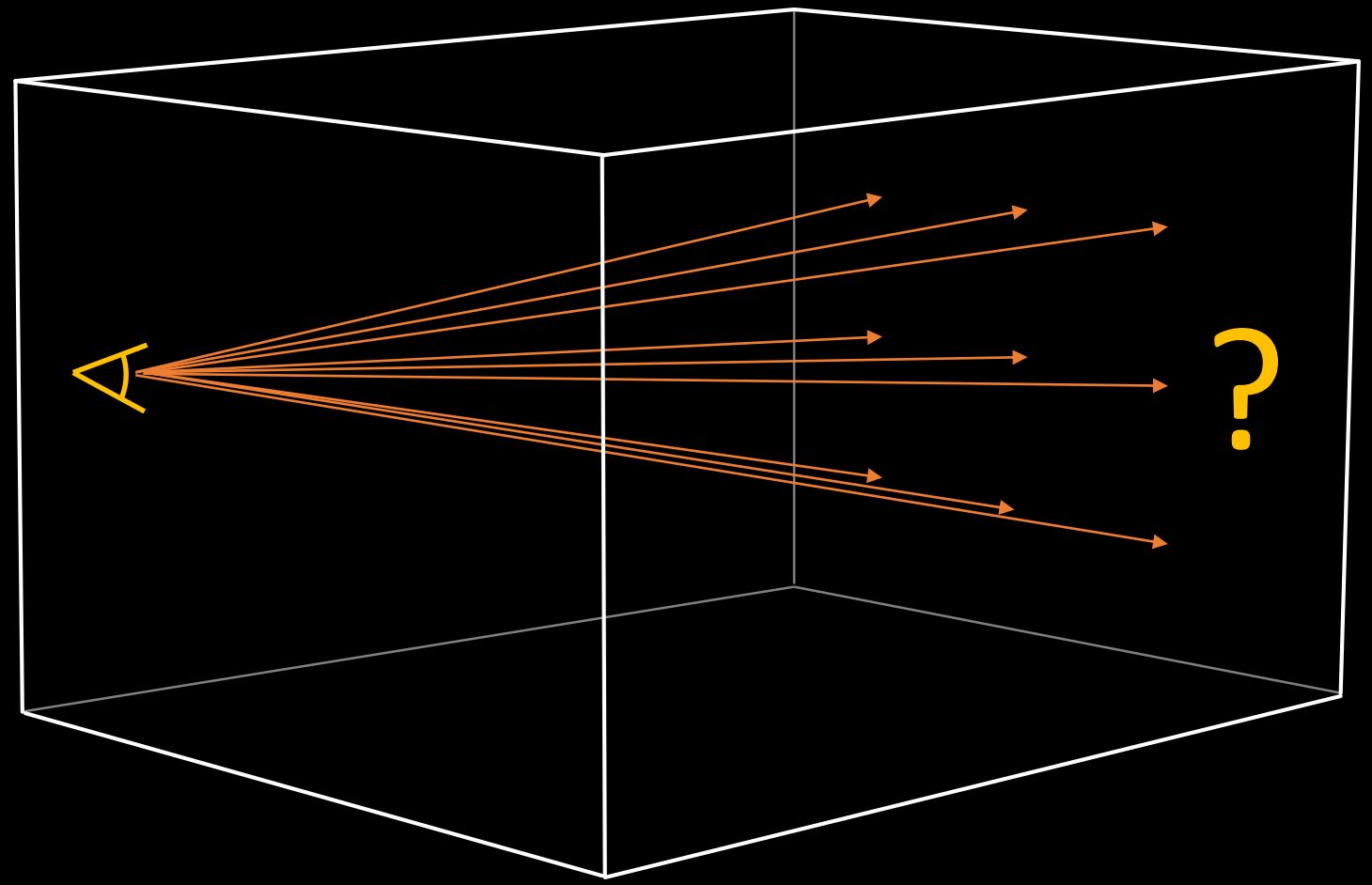
Whitted-Style Ray Tracing: GPU



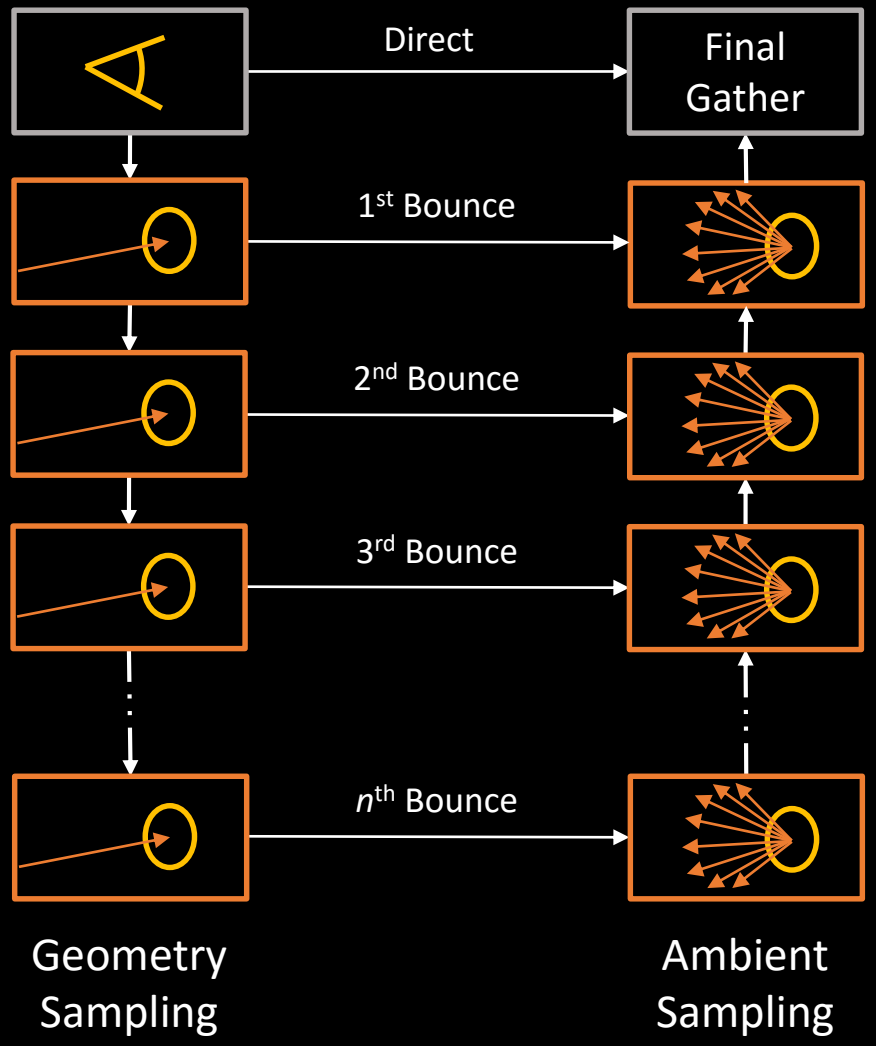
Irradiance Caching: CPU



Irradiance Caching: GPU

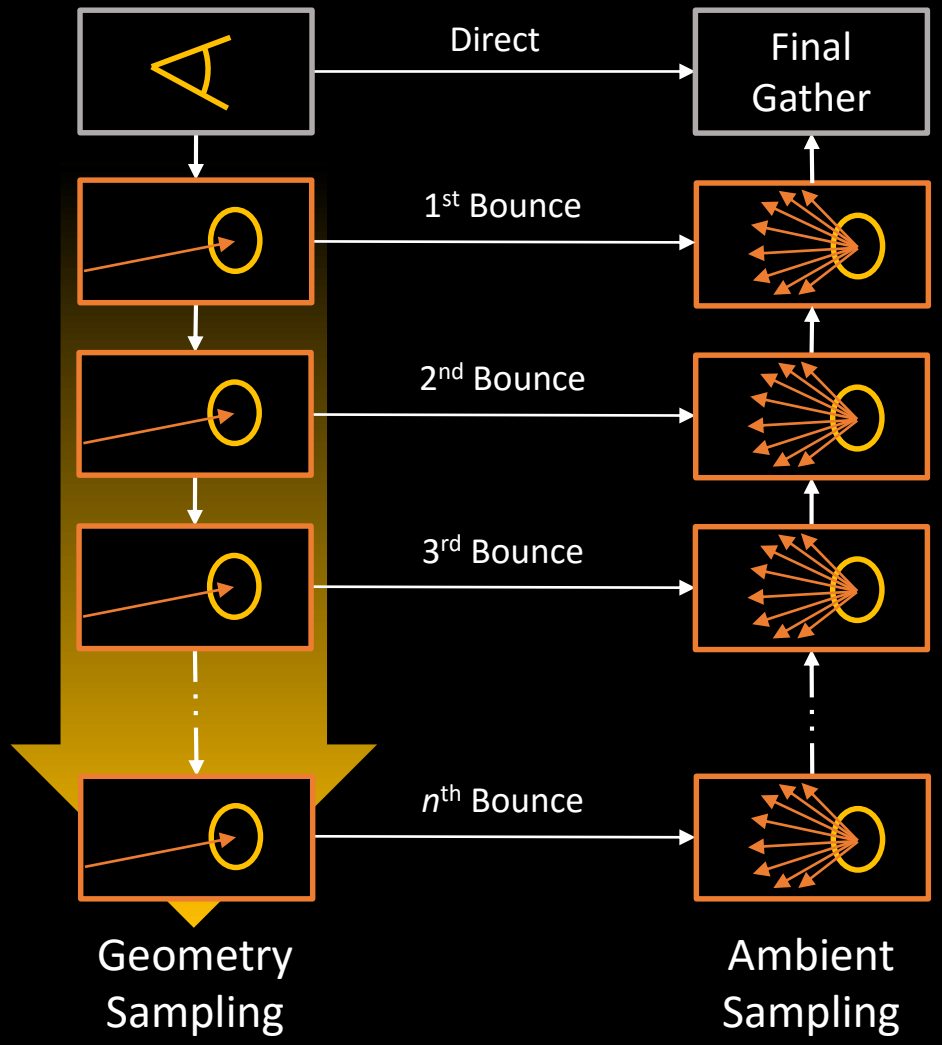


Parallel Multiple-Bounce Irradiance Cache



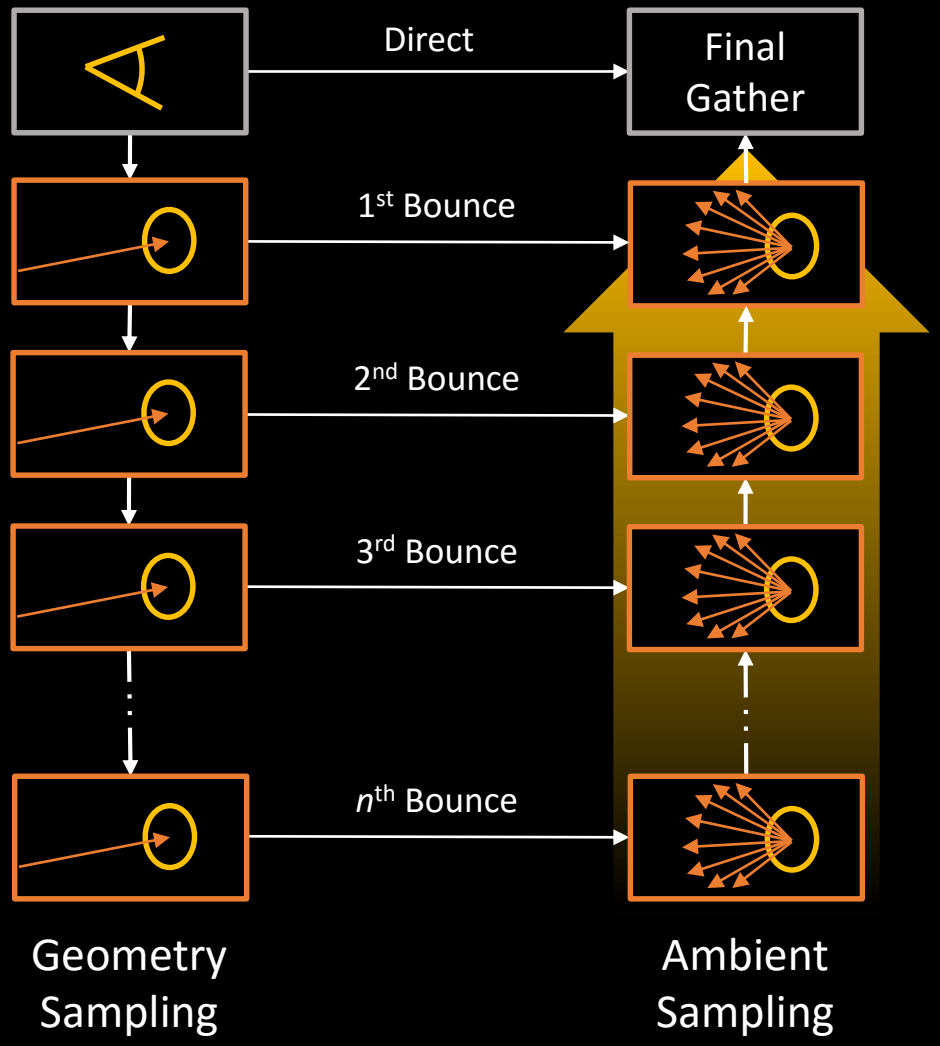

```
OptiX 3080 found 2 GPU devices:
Device 0: Tesla K40c with 15 multiprocessors, 1024 threads per
block, 745000 kHz, 3489464320 bytes global memory, 128 hardware
textures, compute capability 3.5, timeout disabled, Tesla compute
cluster driver enabled, cuda device 0.

Geometry build time: 1014 milliseconds.
OptiX compile time: 765 milliseconds.
OptiX kernel time: 2215 milliseconds (2 seconds).
Adaptive sampling: 16 milliseconds.
Retrieved 262144 of 262144 potential seeds at level 0.
K-means performed 6 loop iterations in 655 milliseconds.
K-means produced 4090 of 4096 clusters at level 0.

OptiX kernel time: 1076 milliseconds (1 seconds).
Retrieved 2012131 of 2166784 potential seeds at level 1.
K-means performed 6 loop iterations in 4773 milliseconds.
K-means produced 4074 of 4096 clusters at level 1.

OptiX kernel time: 515 milliseconds (0 seconds).
Retrieved 1014954 of 1048576 potential seeds at level 2.
K-means performed 6 loop iterations in 2449 milliseconds.
K-means produced 4075 of 4096 clusters at level 2.

Using 3944 of 3944 ambient records
OptiX kernel time: 780 milliseconds (1 seconds).
OptiX kernel time: 1295 milliseconds (1 seconds).
Retrieved 3950 ambient records from 4096 queries at level 2.

Using 3950 of 7894 ambient records
OptiX kernel time: 1513 milliseconds (1 seconds).
OptiX kernel time: 1420 milliseconds (2 seconds).
Retrieved 3943 ambient records from 4096 queries at level 1.

Using 3943 of 11837 ambient records
OptiX kernel time: 2855 milliseconds (3 seconds).
OptiX kernel time: 2730 milliseconds (2 seconds).
Retrieved 3831 ambient records from 4096 queries at level 0.

Using 3831 of 15668 ambient records
OptiX kernel time: 18018 milliseconds (18 seconds).
rpict: ray tracing time: 46769 milliseconds (47 seconds).
```

Parallel Multiple-Bounce Irradiance Cache



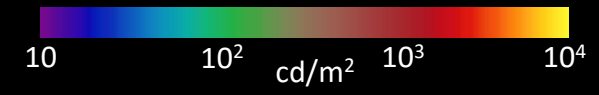

```
OptiX 3080 found 2 GPU devices:
Device 0: Tesla K40c with 15 multiprocessors, 1024 threads per
block, 745000 kHz, 3489464320 bytes global memory, 128 hardware
textures, compute capability 3.5, timeout disabled, Tesla compute
cluster driver enabled, cuda device 0.

Geometry build time: 1014 milliseconds.
OptiX compile time: 765 milliseconds.
OptiX kernel time: 2215 milliseconds (2 seconds).
Adaptive sampling: 16 milliseconds.
Retrieved 262144 of 262144 potential seeds at level 0.
K-means performed 6 loop iterations in 655 milliseconds.
K-means produced 4090 of 4096 clusters at level 0.

OptiX kernel time: 1076 milliseconds (1 seconds).
Retrieved 2012131 of 2166784 potential seeds at level 1.
K-means performed 6 loop iterations in 4773 milliseconds.
K-means produced 4074 of 4096 clusters at level 1.

OptiX kernel time: 515 milliseconds (0 seconds).
Retrieved 1014954 of 1048576 potential seeds at level 2.
K-means performed 6 loop iterations in 2449 milliseconds.
K-means produced 4075 of 4096 clusters at level 2.

Using 3944 of 3944 ambient records
OptiX kernel time: 780 milliseconds (1 seconds).
OptiX kernel time: 1295 milliseconds (1 seconds).
Retrieved 3950 ambient records from 4096 queries at level 2.

Using 3950 of 7894 ambient records
OptiX kernel time: 1513 milliseconds (1 seconds).
OptiX kernel time: 1420 milliseconds (2 seconds).
Retrieved 3943 ambient records from 4096 queries at level 1.

Using 3943 of 11837 ambient records
OptiX kernel time: 2855 milliseconds (3 seconds).
OptiX kernel time: 2730 milliseconds (2 seconds).
Retrieved 3831 ambient records from 4096 queries at level 0.

Using 3831 of 15668 ambient records
OptiX kernel time: 18018 milliseconds (18 seconds).
rpict: ray tracing time: 46769 milliseconds (47 seconds).
```

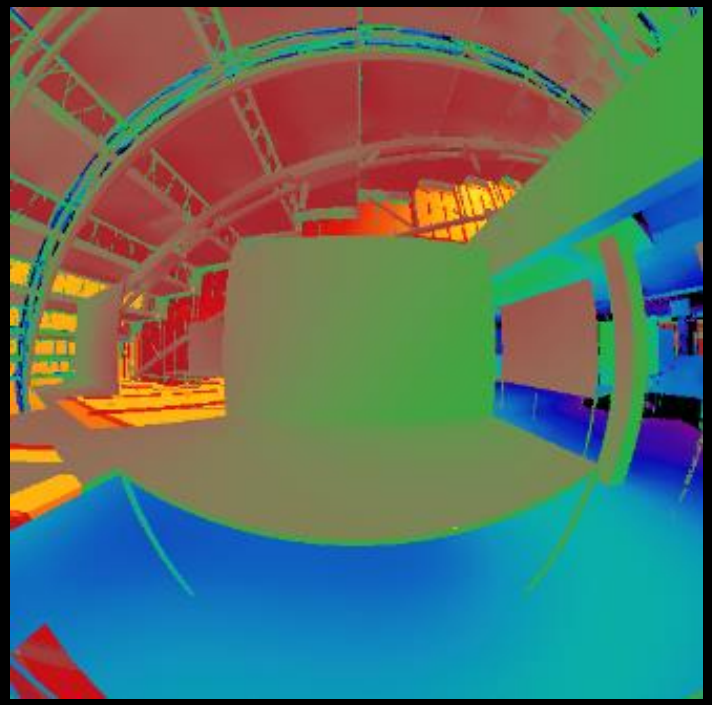
Parallel Multiple-Bounce Irradiance Cache



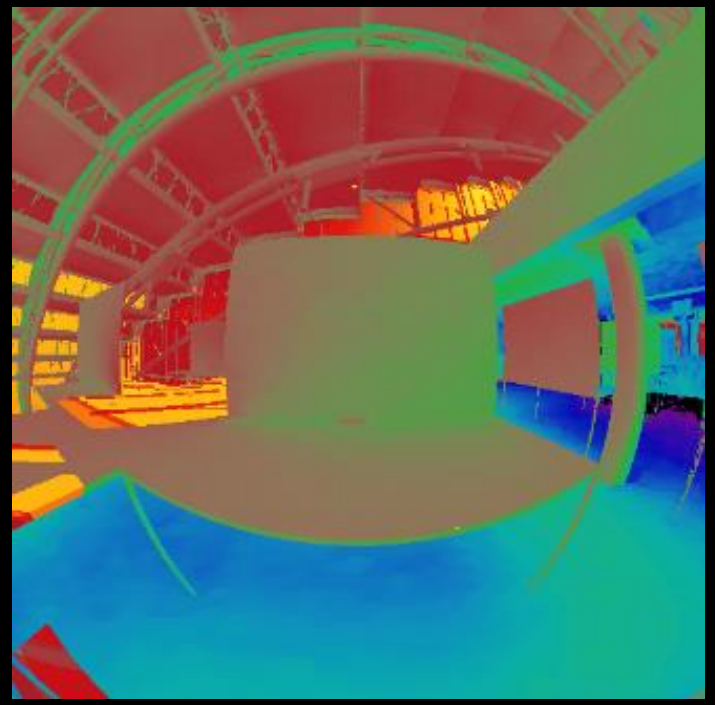

```
OptiX 3080 found 2 GPU devices:
Device 0: Tesla K40c with 15 multiprocessors, 1024 threads per
block, 745000 kHz, 3489464320 bytes global memory, 128 hardware
textures, compute capability 3.5, timeout disabled, Tesla compute
cluster driver enabled, cuda device 0.

Geometry build time: 1014 milliseconds.
OptiX compile time: 765 milliseconds.
OptiX kernel time: 2215 milliseconds (2 seconds).
Adaptive sampling: 16 milliseconds.
Retrieved 262144 of 262144 potential seeds at level 0.
K-means performed 6 loop iterations in 655 milliseconds.
K-means produced 4090 of 4096 clusters at level 0.

OptiX kernel time: 1076 milliseconds (1 seconds).
Retrieved 2012131 of 2166784 potential seeds at level 1.
K-means performed 6 loop iterations in 4773 milliseconds.
K-means produced 4074 of 4096 clusters at level 1.

OptiX kernel time: 515 milliseconds (0 seconds).
Retrieved 1014954 of 1048576 potential seeds at level 2.
K-means performed 6 loop iterations in 2449 milliseconds.
K-means produced 4075 of 4096 clusters at level 2.

Using 3944 of 3944 ambient records
OptiX kernel time: 780 milliseconds (1 seconds).
OptiX kernel time: 1295 milliseconds (1 seconds).
Retrieved 3950 ambient records from 4096 queries at level 2.

Using 3950 of 7894 ambient records
OptiX kernel time: 1513 milliseconds (1 seconds).
OptiX kernel time: 1420 milliseconds (2 seconds).
Retrieved 3943 ambient records from 4096 queries at level 1.

Using 3943 of 11837 ambient records
OptiX kernel time: 2855 milliseconds (3 seconds).
OptiX kernel time: 2730 milliseconds (2 seconds).
Retrieved 3831 ambient records from 4096 queries at level 0.

Using 3831 of 15668 ambient records
OptiX kernel time: 18018 milliseconds (18 seconds).
rpict: ray tracing time: 46769 milliseconds (47 seconds).
```

Irradiance Caching Results





Accelerad
10 minutes



Radiance
198 minutes

Jones and Reinhart, 2014. Irradiance caching for global illumination calculation on graphics hardware.
2014 ASHRAE/IBPSA-USA Building Simulation Conference, 111-120.

Introducing Accelerad

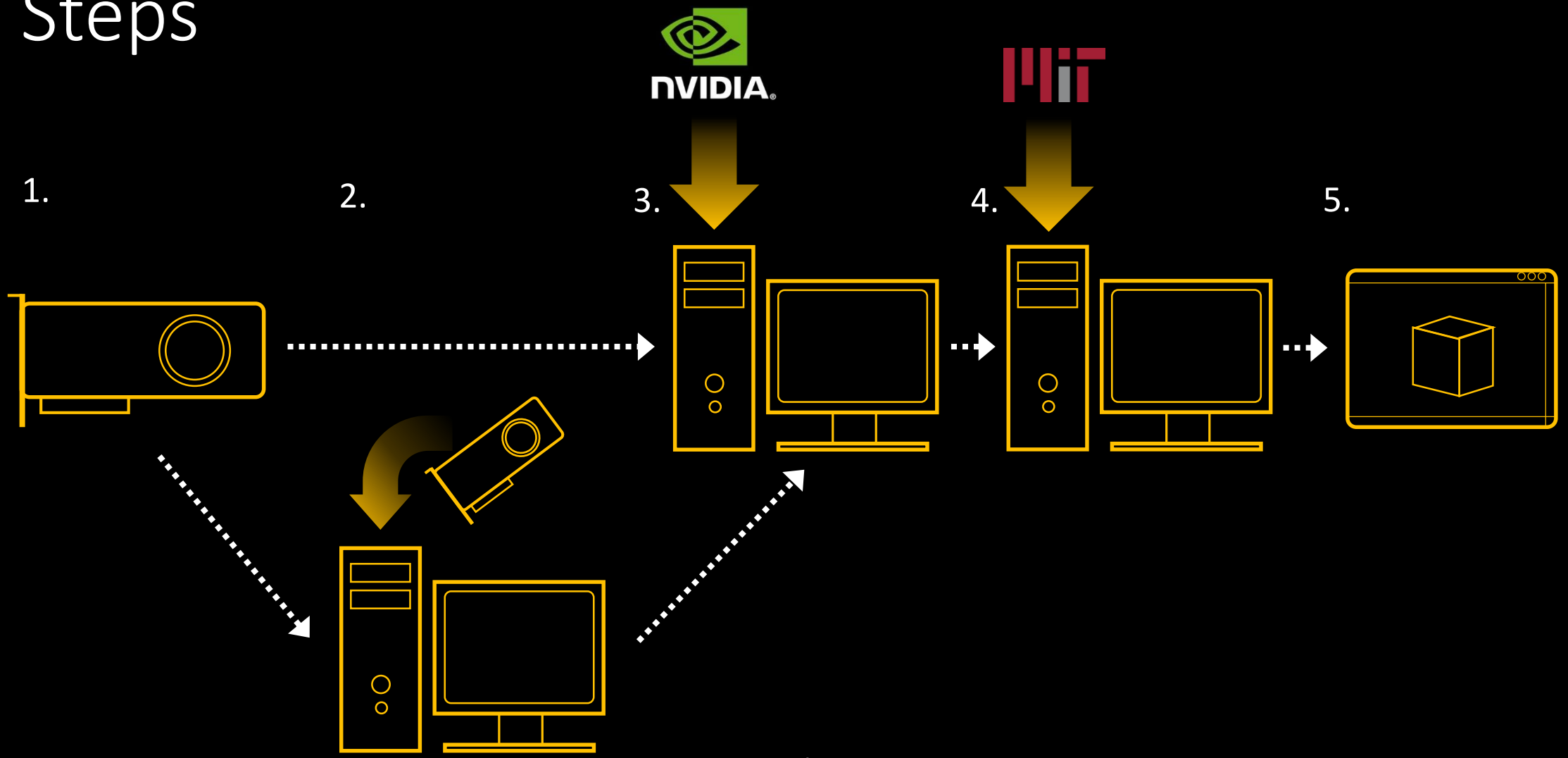
Accelerad vs Radiance

Setup and Installation

Tips and Tricks

New in Version 0.4 beta

Steps


NVIDIA
1.
2.
3.
4.
5.
Choose a GPU
Install hardware
Update graphics driver
Install Accelerad
Run as usual

Compute Capability

Fermi

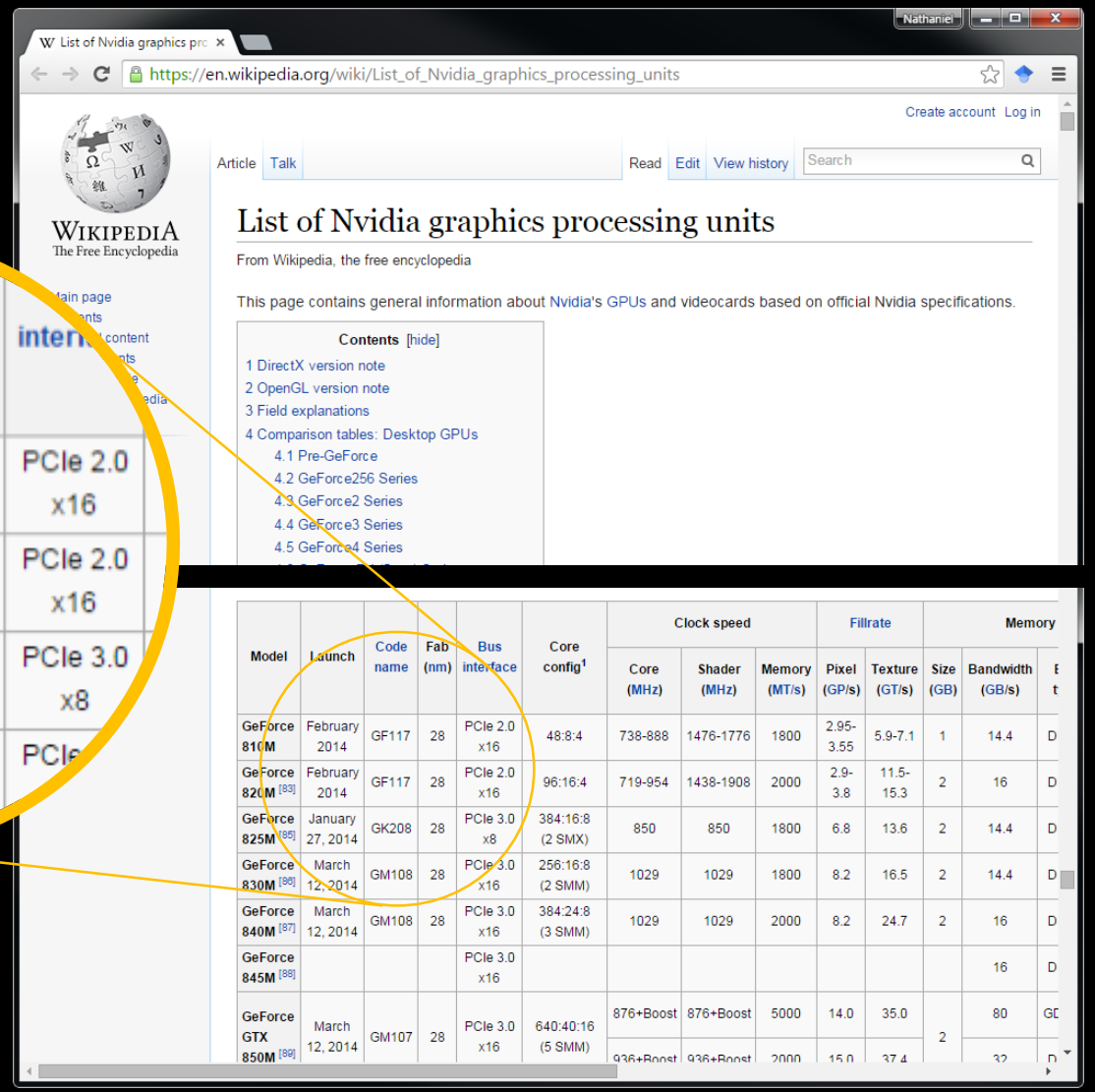
- introduced 2009
- compute 2.X
- code GF

Kepler

- introduced 2012
- compute 3.X
- code GK

Maxwell

- introduced 2014
- compute 5.X
- code GM



Product Family

TCC: *Tesla Compute Cluster*

Gaming | Graphics | Compute

GeForce® | Quadro® | Tesla®

WDDM: *Windows Display Driver Model*

Cost

http://www.nvidia.com/

Other Factors

Check compute benchmark tests

Don’t worry too much about specs

- CUDA cores/shader processors
- Clock speed
- Memory
- Memory bandwidth

Use multiple GPUs

- For additional cores
- For separate display

DIY installation

- Physical dimensions
- Power supply

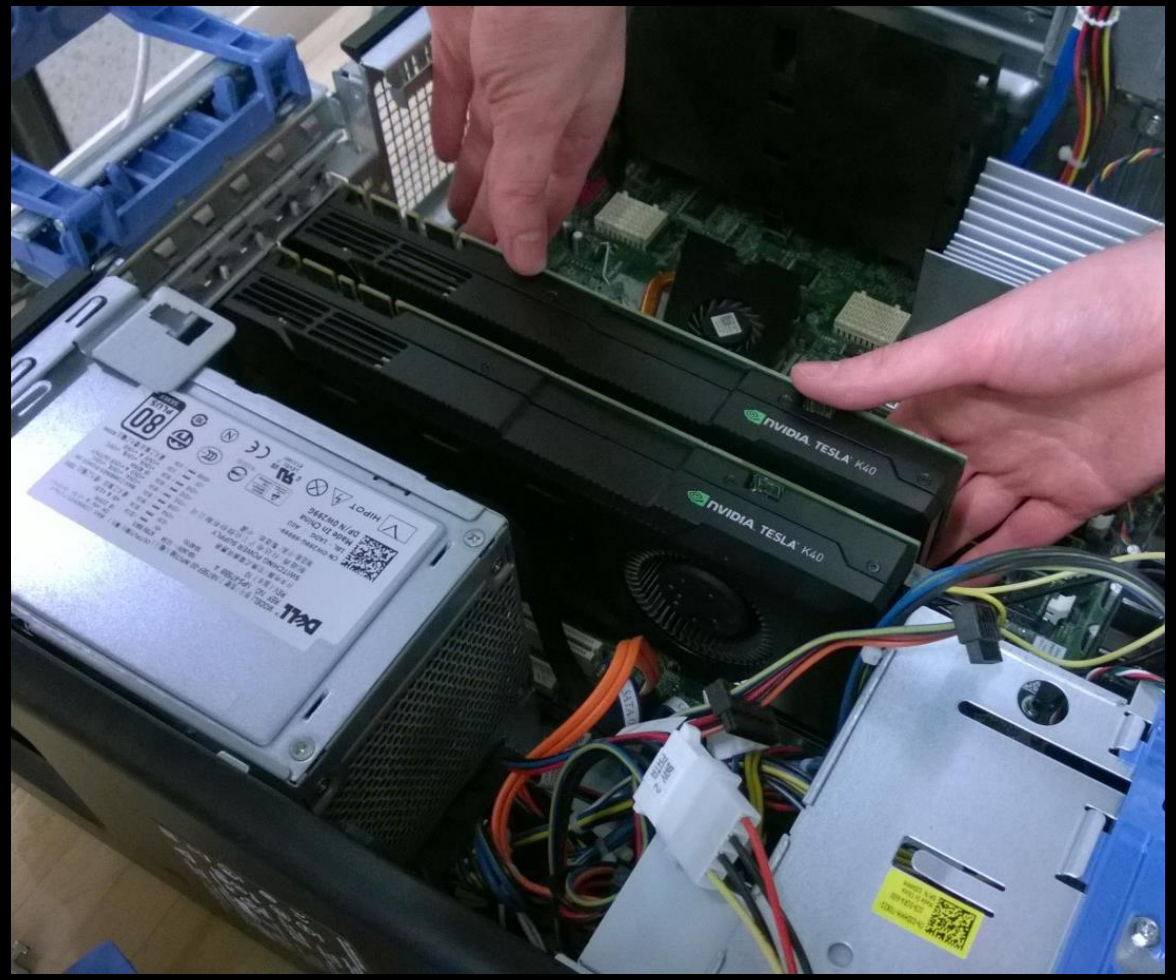
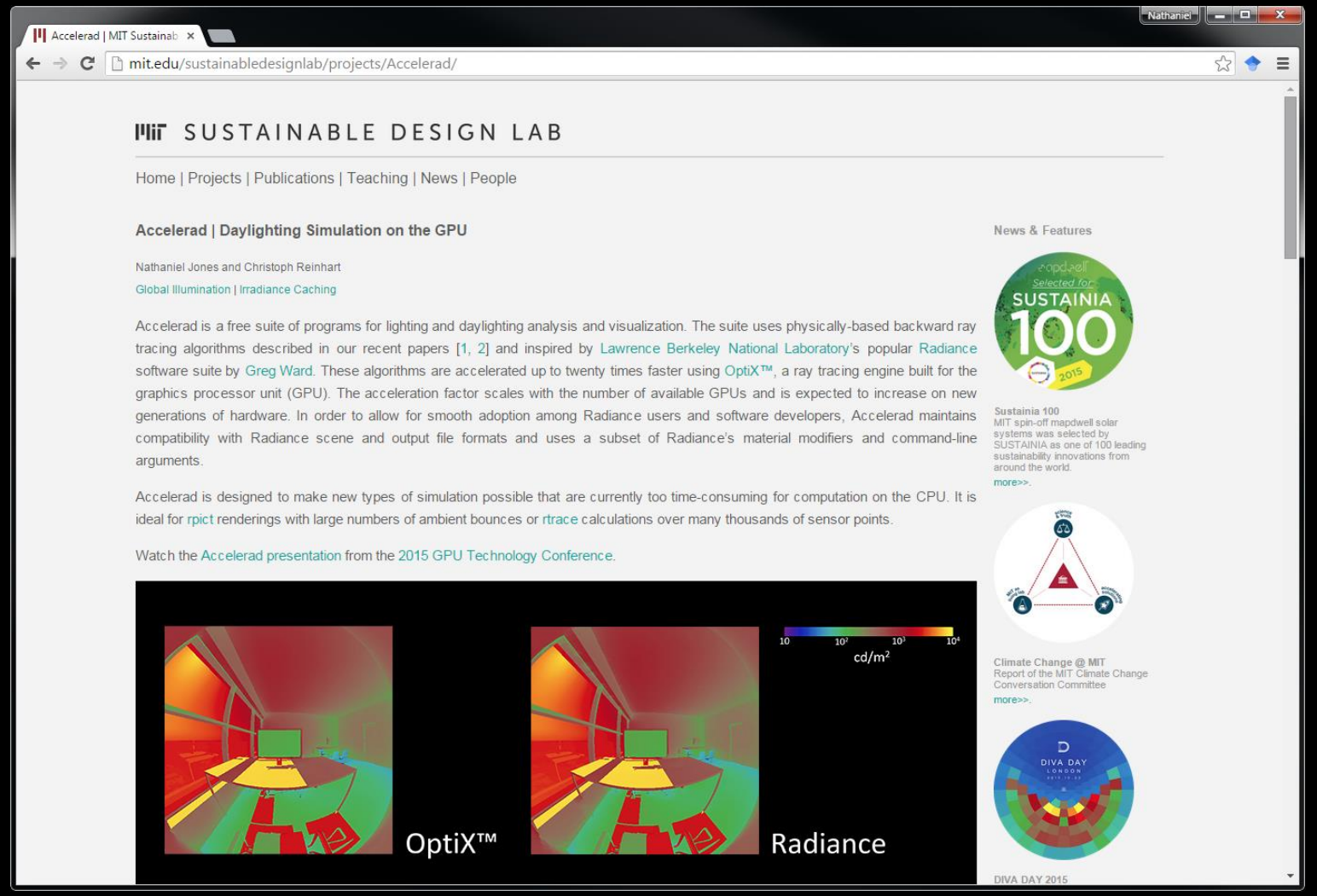


Image Credit: J. Alstan Jakubiec

Download and Install



http://mit.edu/sustainabledesignlab/projects/Accelerad/

Download and Install (Advanced)

Executable files

- accelerad-rpict.exe
- accelerad-rtrace.exe

Move and rename as needed

Typically found in PATH

Libraries

- optix.1.dll
- cudart64_XX.dll

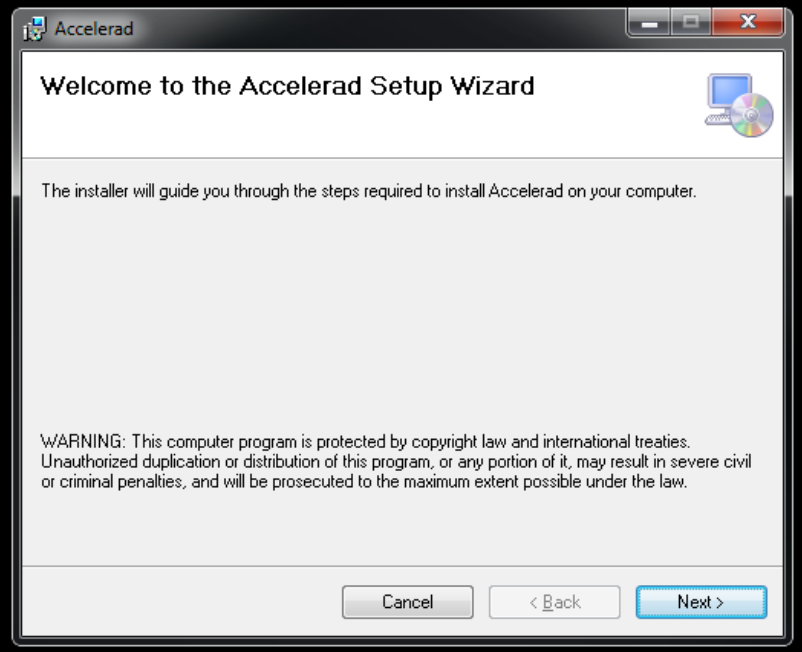
Found in PATH (or LD_LIBRARY_PATH)

Parallel Thread Execution (PTX) files

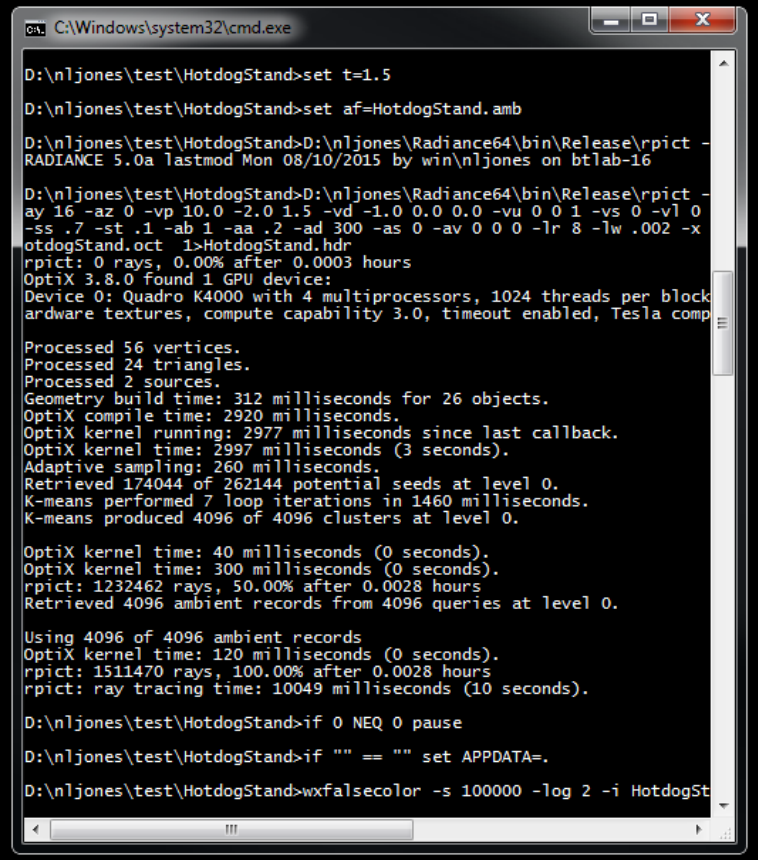
- *.ptx

Move as needed

Found in RAYPATH



Use in Normal Workflow



OR

Caveats

-dj	-ds	-dv	-dt	-dc	-dr	-dp
-ss	-st	-x	-y	-lr	-lw	-ld
-av	-aw	-ab	-ar	-aa	-ad	-as
-af	-ap	-am	-ae	-aE	-ai	-al
-bv	-I	-i	-me	-ma	-mg	-ms
-vt	-vp	-vd	-vu	-vh	-vv	-vo
-va	-vs	-vl	-vf	-e	-w	-u
-pa	-pj	-pm	-pd	-ps	-pt	-S
-o	-t	-z	-h	-r	-ro	-n

source	tube	spotlight	trans	plasfunc	BSDF	brightdata
sphere	ring	mirror	plastic2	metfunc	antimatter	colorpict
bubble	mesh	prism1	metal2	transfunc	texfunc	colortext
polygon	instance	prism2	trans2	BRTDfunc	texdata	brighttext
cone	light	mist	dielectric	plasdata	colorfunc	mixfunc
cup	illum	plastic	interface	metdata	brightfunc	mixdata
cylinder	glow	metal	glass	transdata	colordata	mixpict

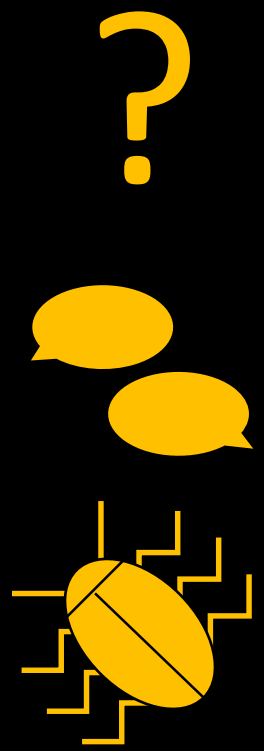
Implemented

Partial Implementation

Not yet Implemented

Never to be Implemented

Feedback



Questions

Discussions

Bug Reports

Google Group:
Accelerad Users

Introducing Accelerad

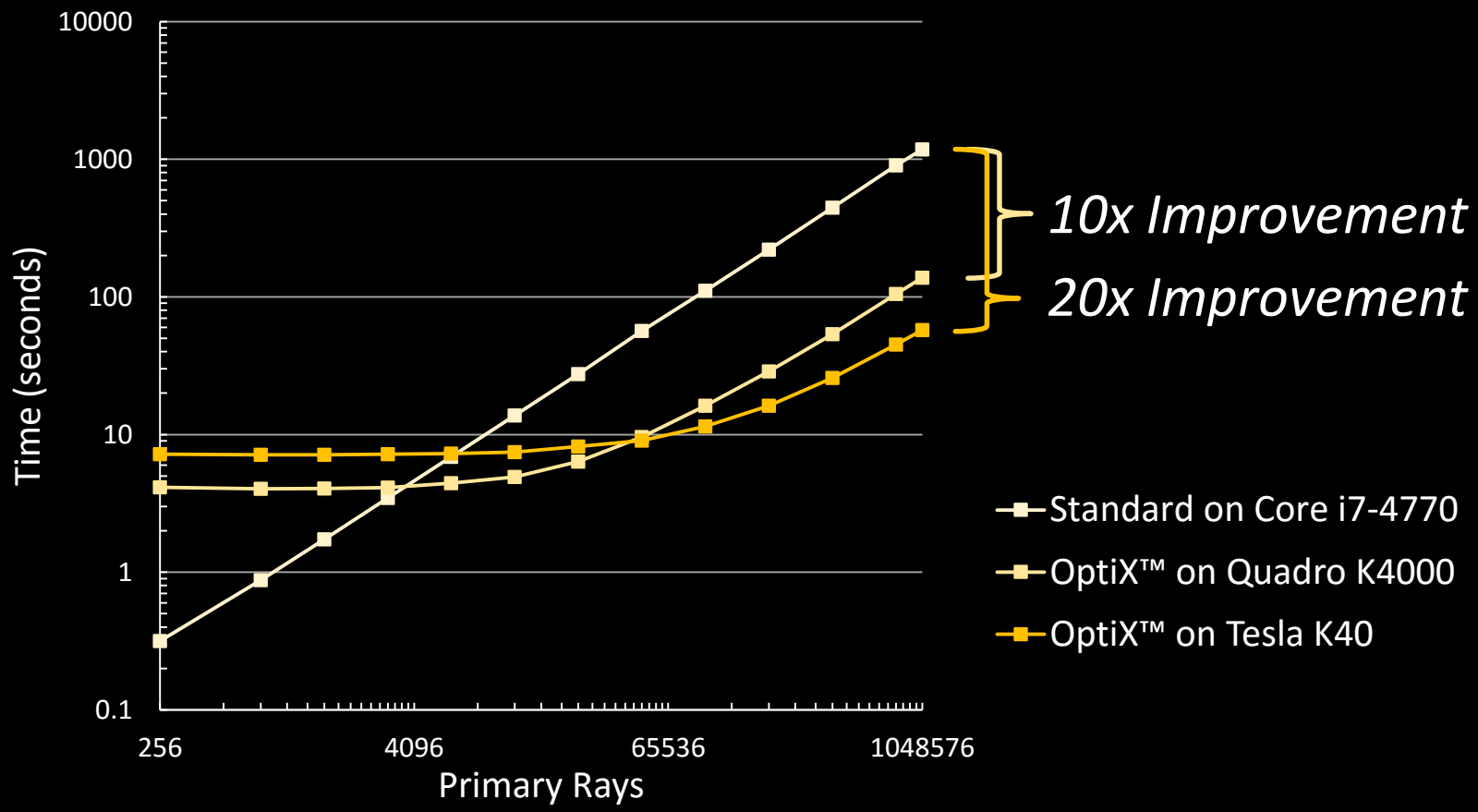
Accelerad vs Radiance

Setup and Installation

Tips and Tricks

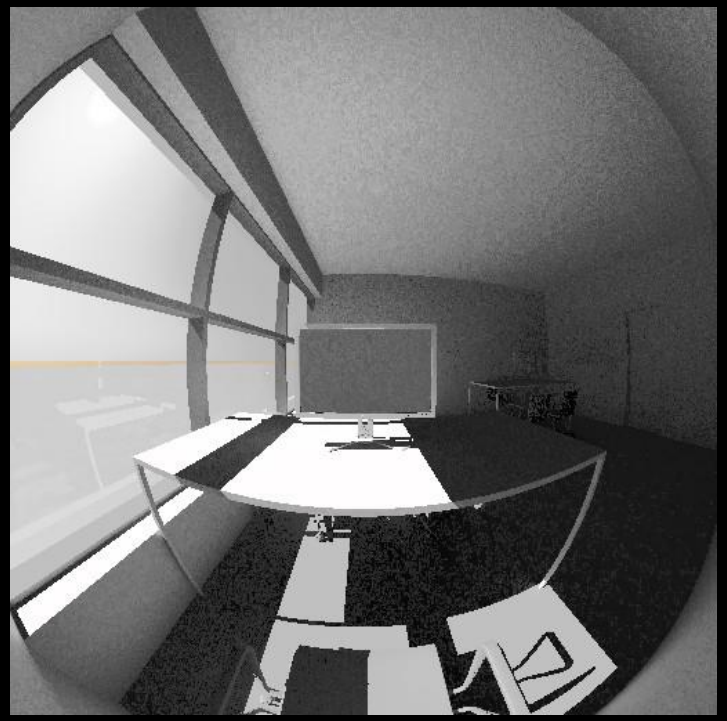
New in Version 0.4 beta

Ray Count (-*x*, -*y*)



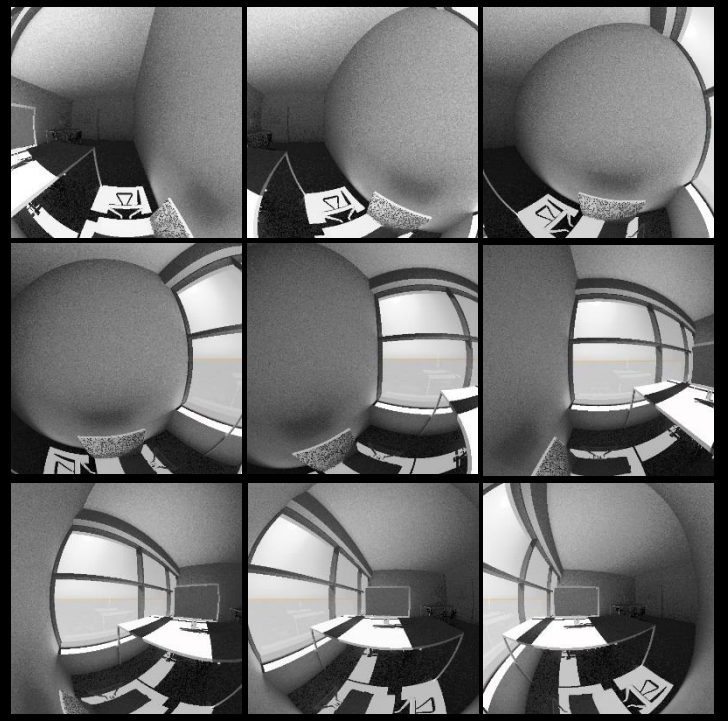
Jones and Reinhart, 2014. Physically based global illumination calculation using graphics hardware.
Proceedings of eSim 2014: The Canadian Conference on Building Simulation, 474-487.

Animations (*-S*, *-az*)



1 image

7x speedup

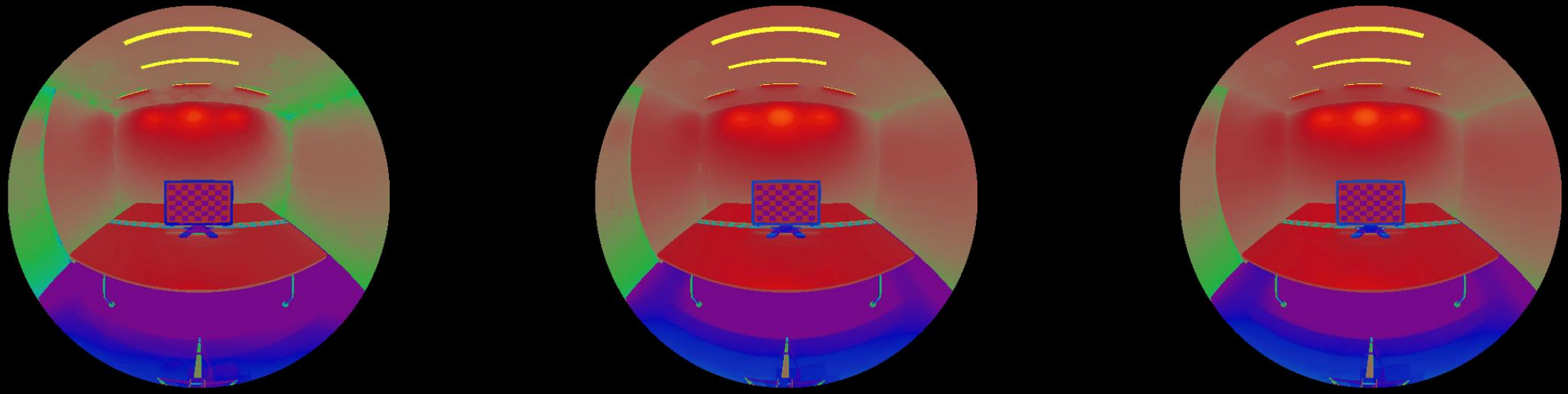


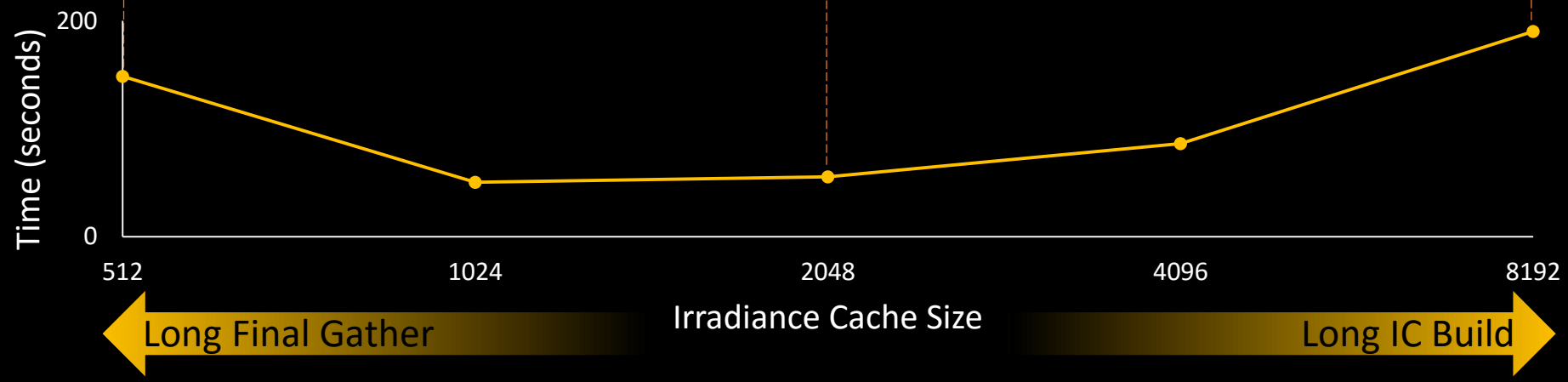
120 images

17x speedup

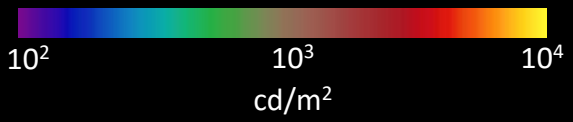
Jones and Reinhart, 2014. Physically based global illumination calculation using graphics hardware. *Proceedings of eSim 2014: The Canadian Conference on Building Simulation*, 474-487.

Irradiance Cache Size (*-ac*)





Ambient Accuracy (*-aa*, *-ar*)



Radiance

Accelerad

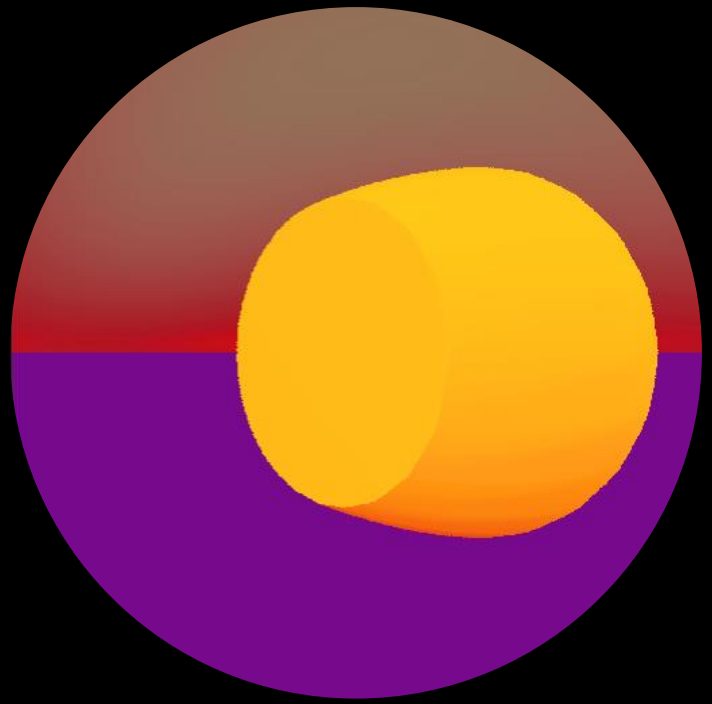
Smoother Shading

Better Coverage

-aa 0.05

-aa 0.1

-aa 0.2



Introducing Accelerad

Accelerad vs Radiance

Setup and Installation

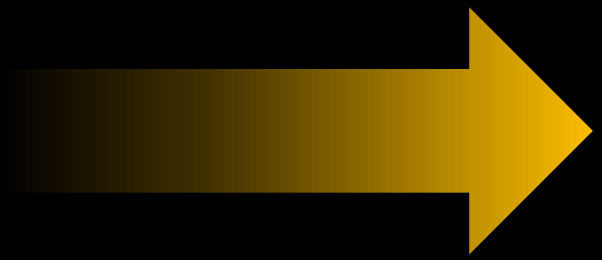
Tips and Tricks

New in Version 0.4 beta

New Base Libraries

Accelerad 0.3 beta

- Radiance 5.0.a.3
- OptiX 3.8.0 beta
- CUDA 6.5



Accelerad 0.4 beta

- Radiance 5.0
- OptiX 3.8.0
- CUDA 7.0

New Geometric Primitives

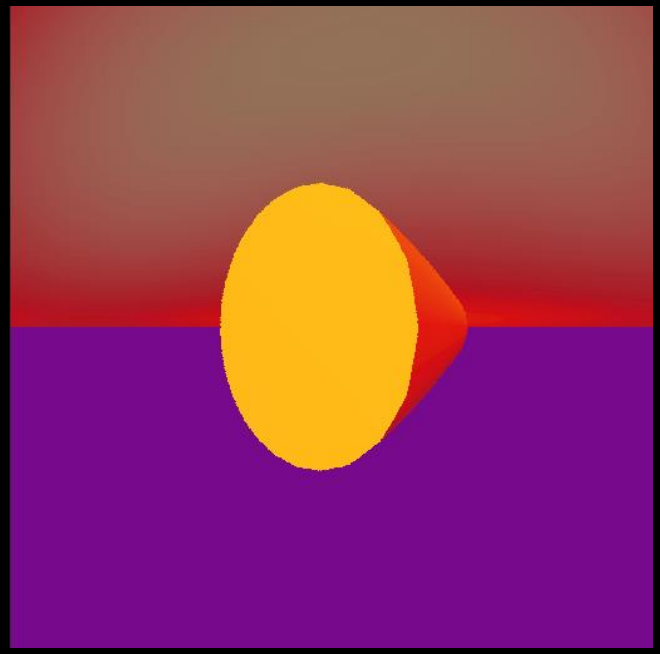
Cone

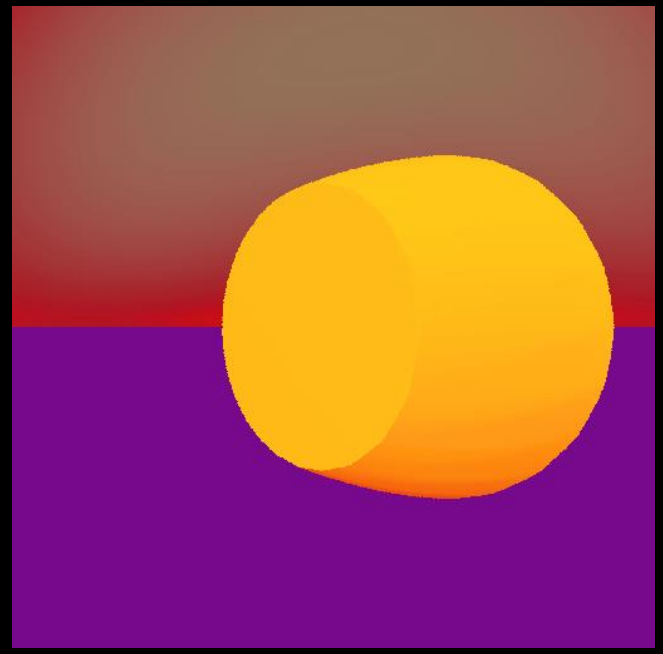
Cup

Cylinder

Tube

Ring





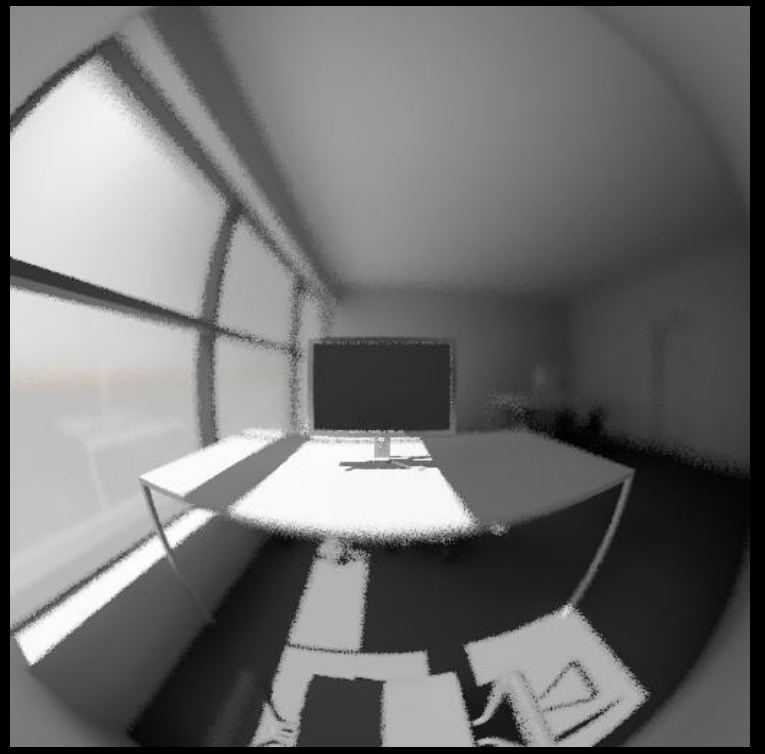
New Arguments

Ambient super-sampling (*-as*)

Random seeding (*-u*)

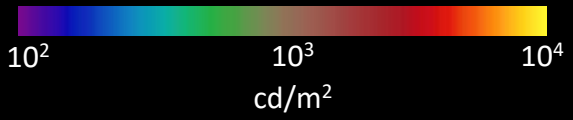
Motion blur (*-pm*)

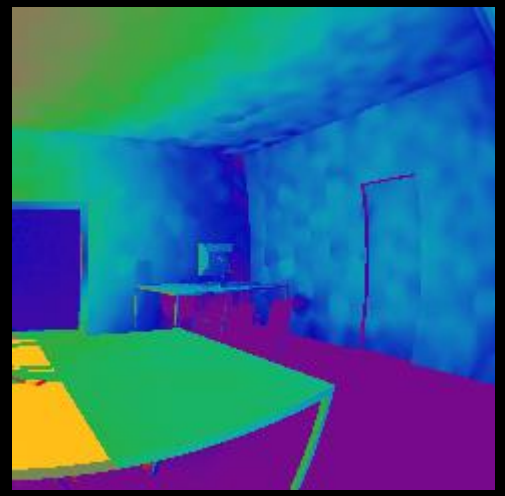
Depth-of-field blur (*-pd*)



Combined depth-of-field blur (*-pd*) *and random seeding* (*-u*)

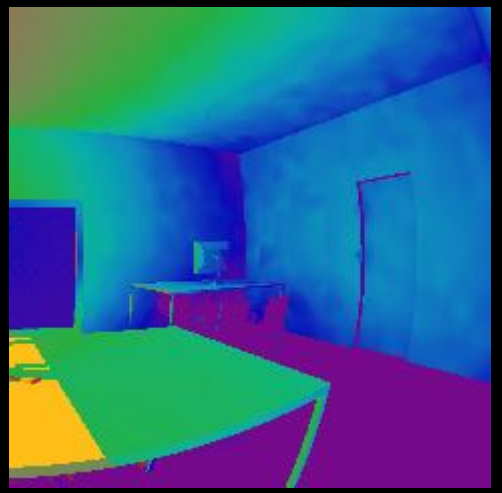
New Speedup (*-ag*)





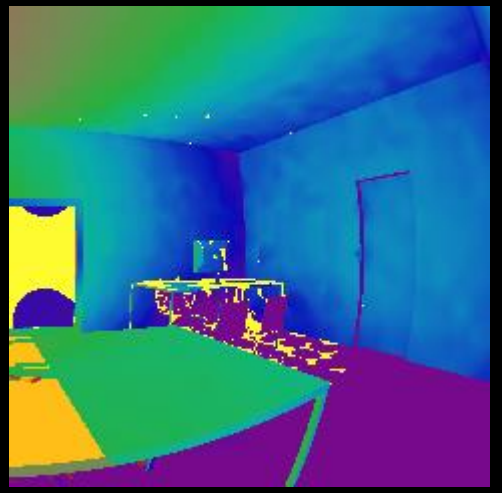
-ad 256

22 seconds



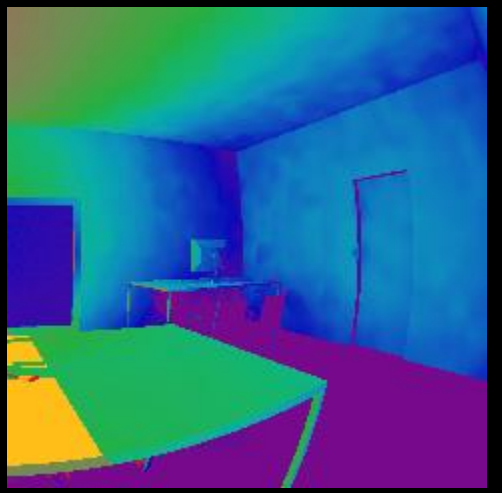
-ad 1024

54 seconds



-ad 1024
No Infill

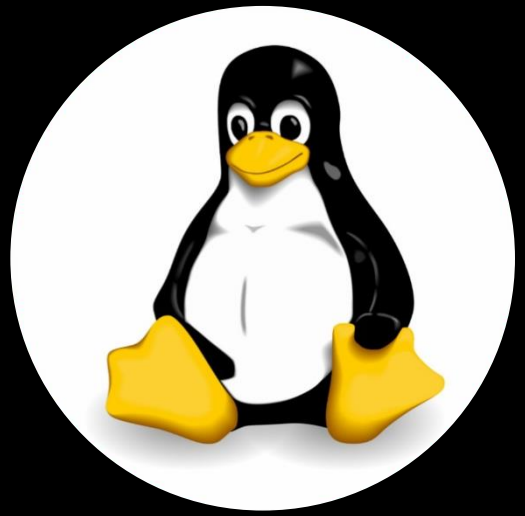
30 seconds



-ad 1024
-ag 256

36 seconds

New Platform



LINUX

Thanks

مؤسسة الكويت للتقدم العلمي
KUWAIT FOUNDATION
For the Advancement Of Sciences

Massachusetts Institute of Technology

SDLAB

NVIDIA

Download

http://mit.edu/sustainabledesignlab/projects/Accelerad/

Questions?

Nathaniel Jones <nljones@mit.edu>