SIGGRAPH 2012

The **39th** International **Conference** and **Exhibition**
on **Computer Graphics** and **Interactive Techniques**

Beyond Programmable Shading Course
ACM SIGGRAPH 2012

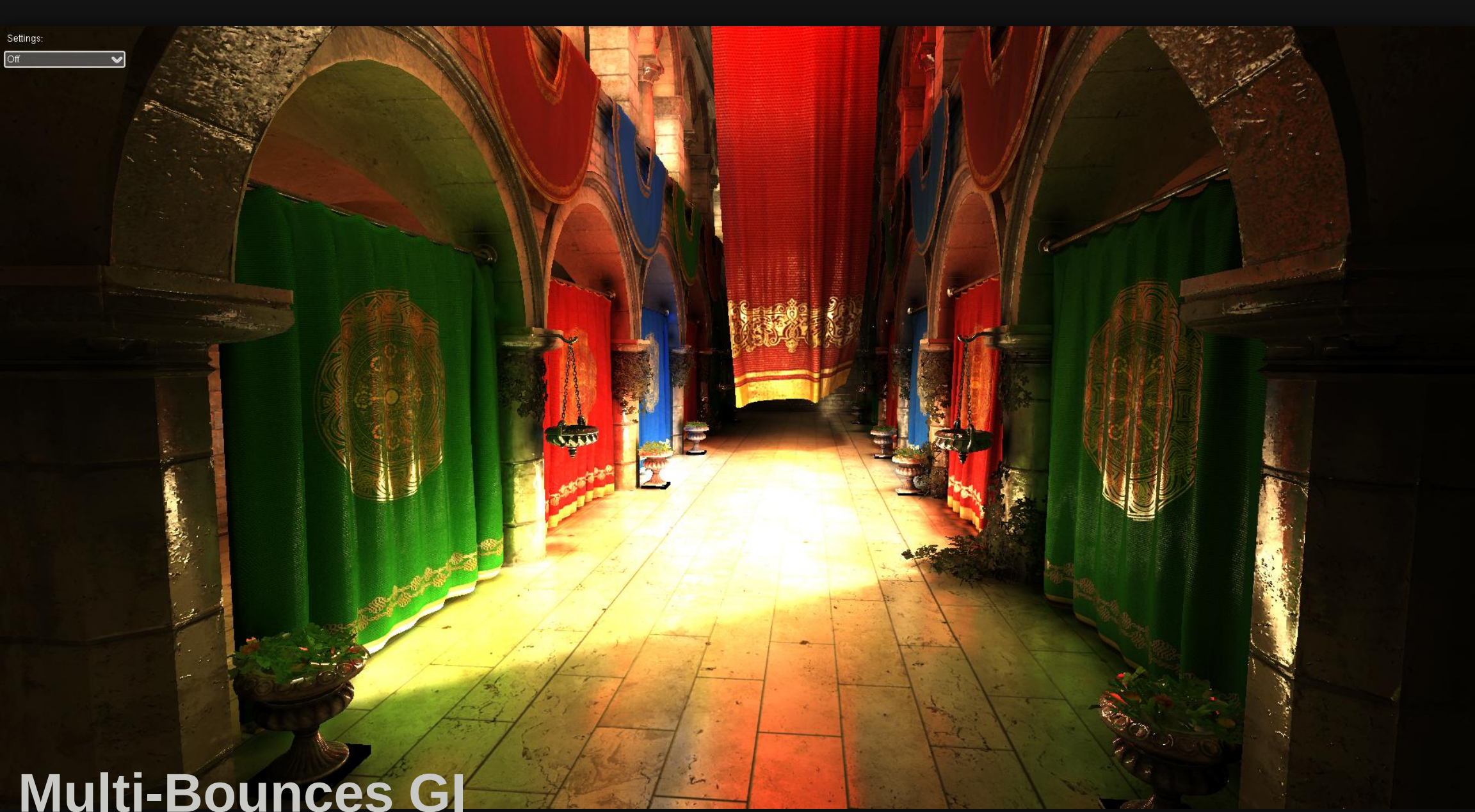
Dynamic Sparse Voxel Octrees For Next-Gen Real-Time Rendering

Cyril Crassin,
NVIDIA Research

Massive complex scenes

Procedural content

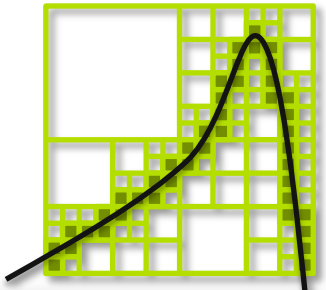
[Matt Swoboda]

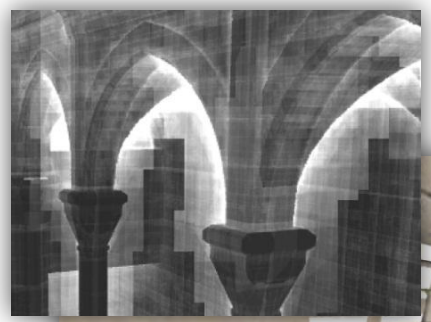

Settings:
Off
Multi-Bounces GI

Sparse Voxel Octree

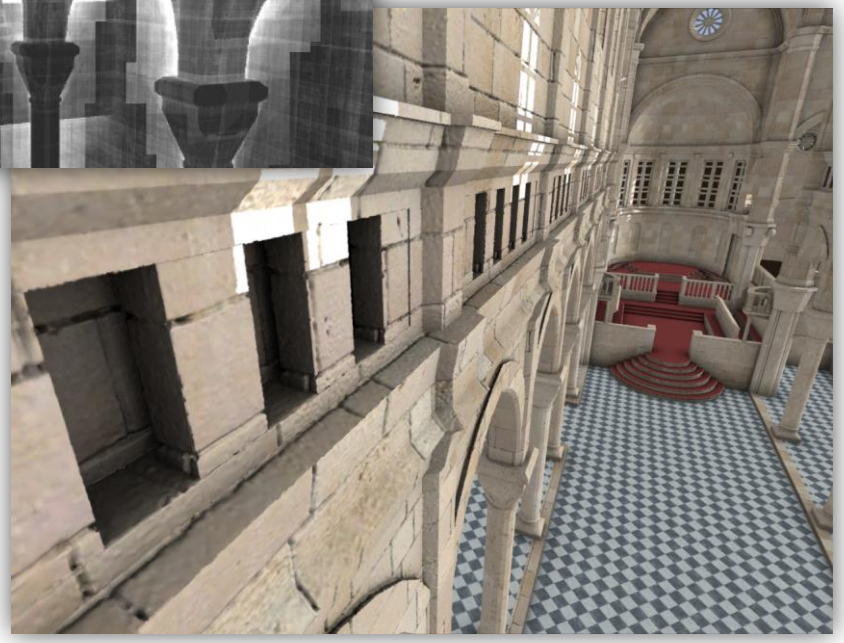
- Compact multi-resolution voxel representation

Olick. 2008

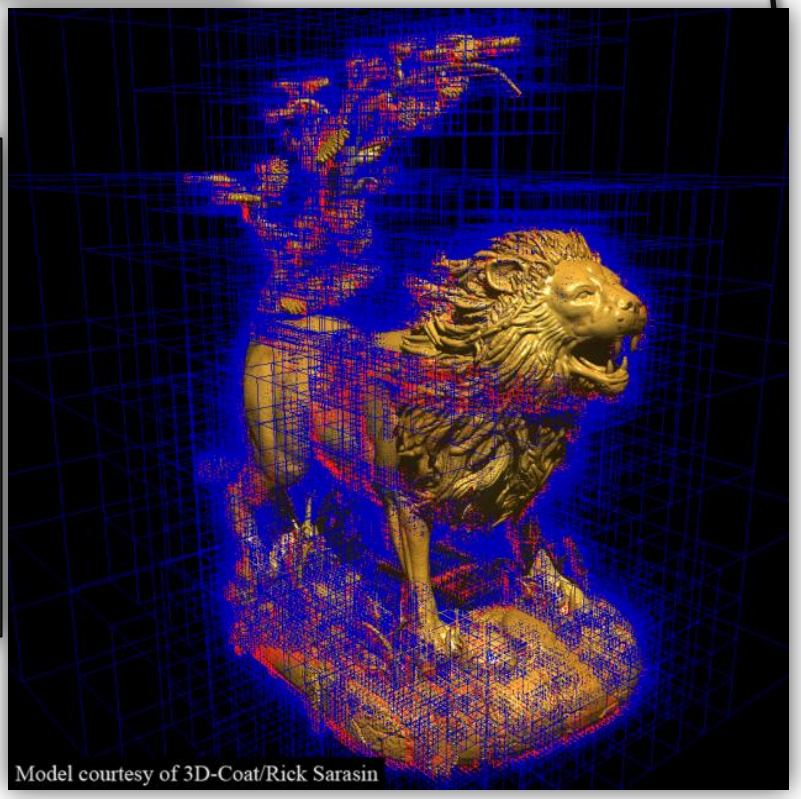


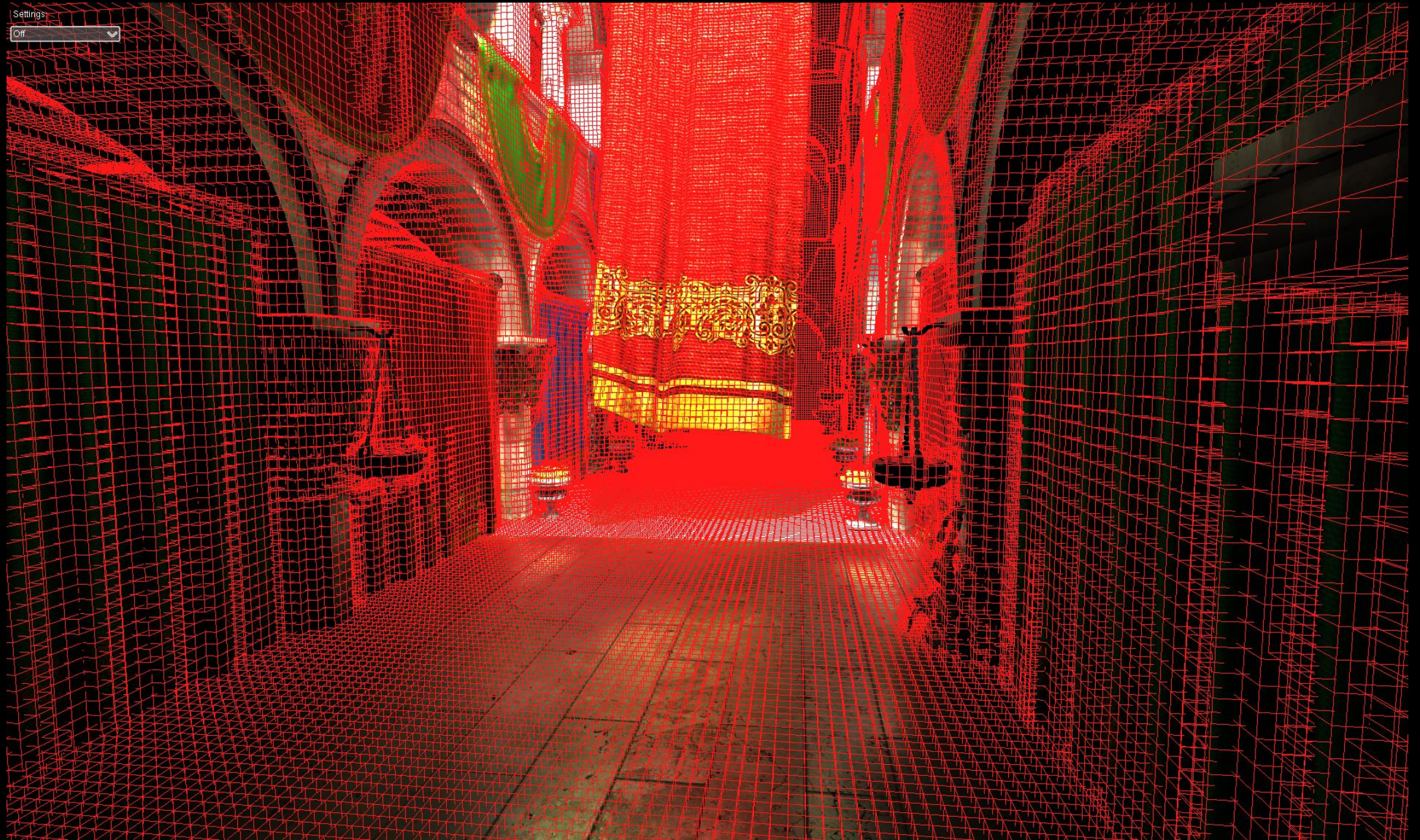


Laine and Karras (NVIDIA) 2010

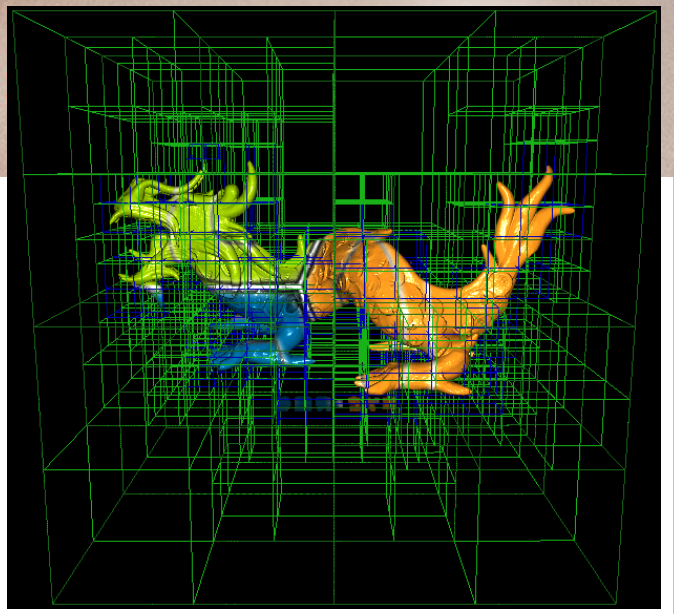


Crassin et al. 2009 (GigaVoxels)

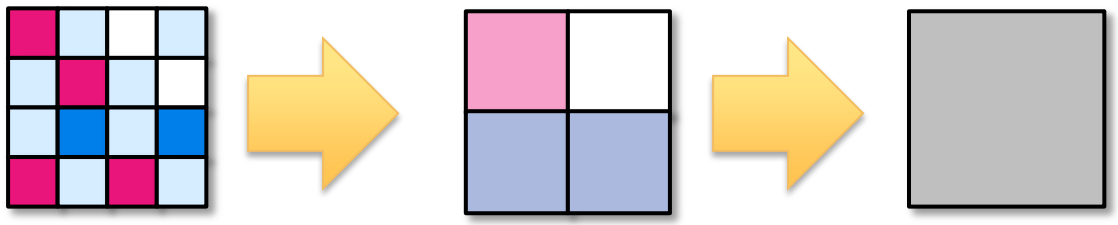


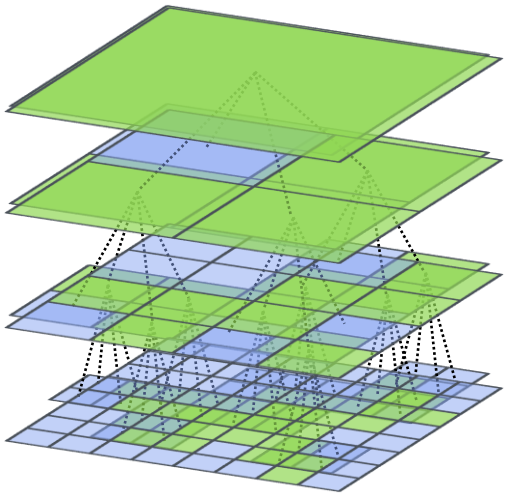

Settings
Off

Voxel Octree



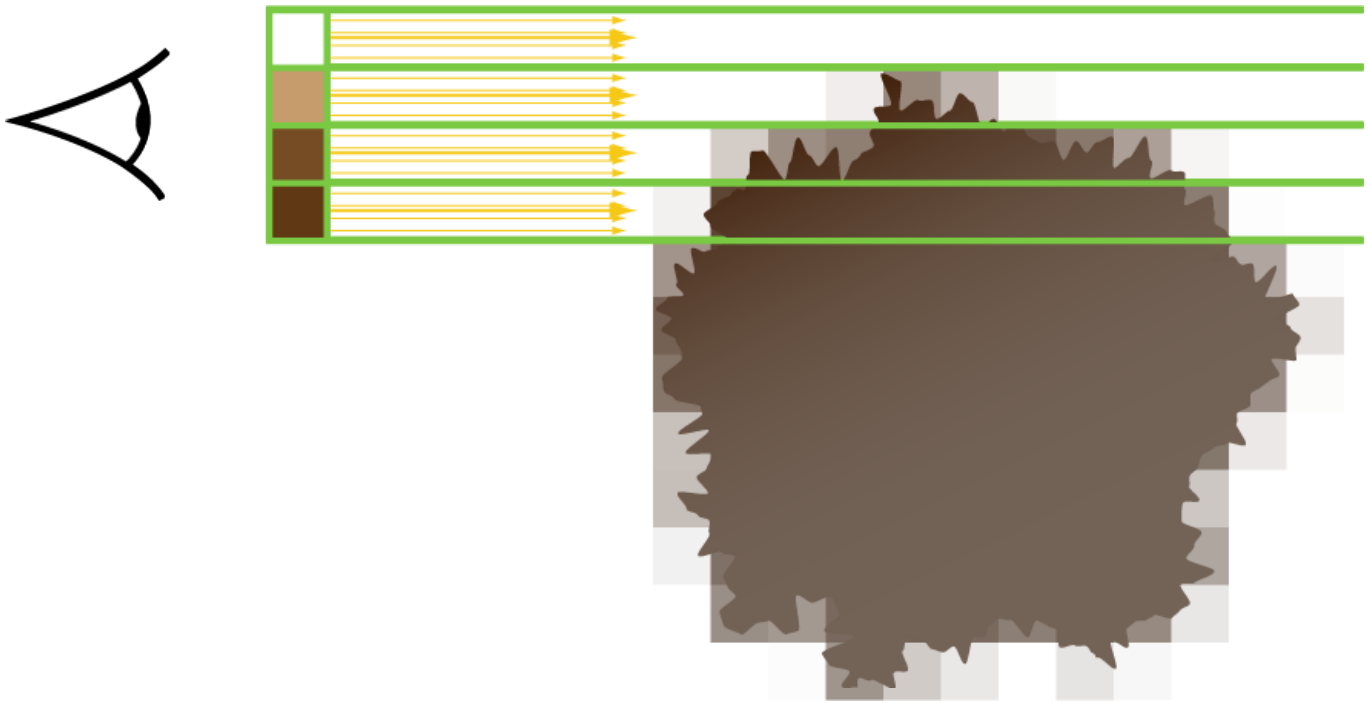
- **Hierarchical volumes**
  - Multi-resolution geometry
  - Filtered values



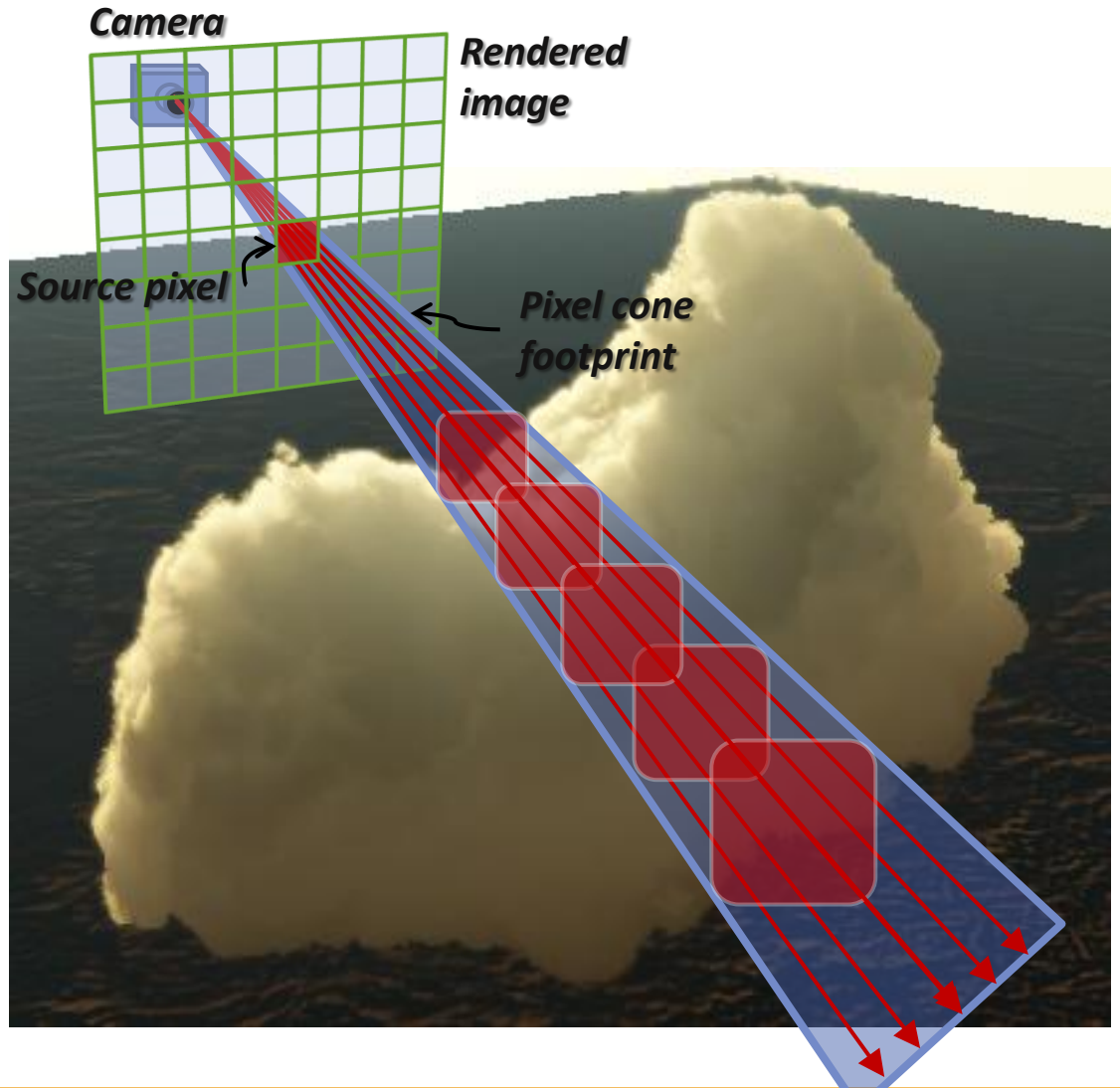


Geometry pre-filtering

- **Pre-filtered** geometric representation
  - Adapt **geometry**-resolution to **sampling resolution**

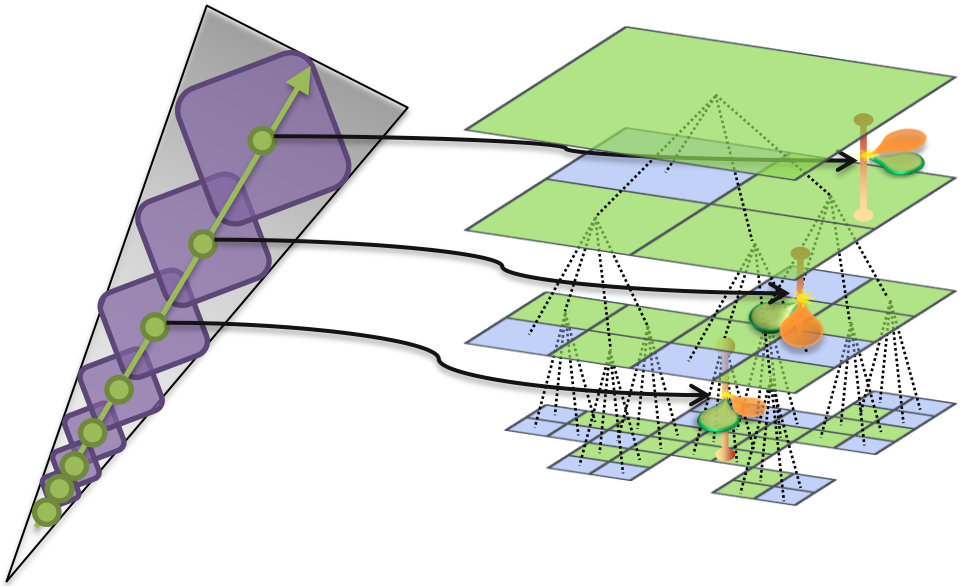


Voxel cone-tracing model



Voxel cone-tracing model

- **Filtered geometry is integrated as a participating media**
  - Volume *ray-casting*



Challenges

Representation

Global scene access

Filtering

#1 : Cinematic image quality

- Aliasing, Depth Of Field, Geometry Complexity
- Transparency

#2 : Illumination

- Visibility and light transport

#4 : Production costs

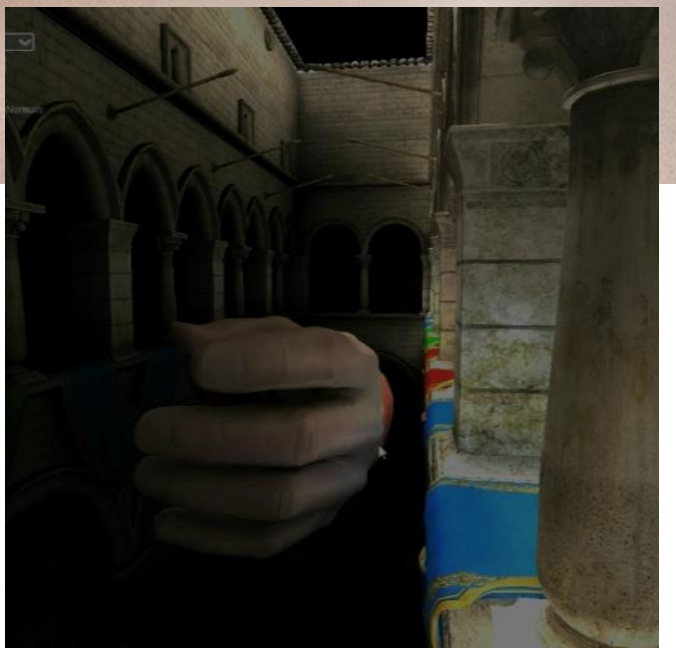
- Content creation, Procedural amplification /Generation, LODs

#5 : Scaling up

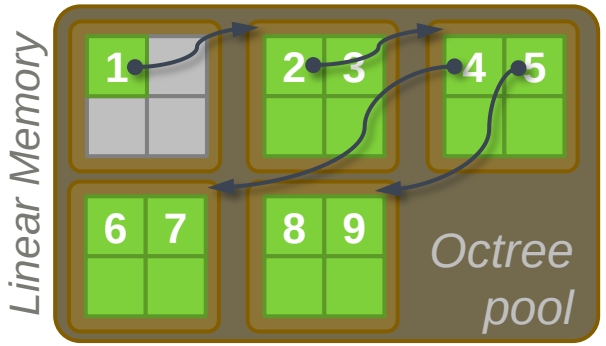
- Massive detailed scenes

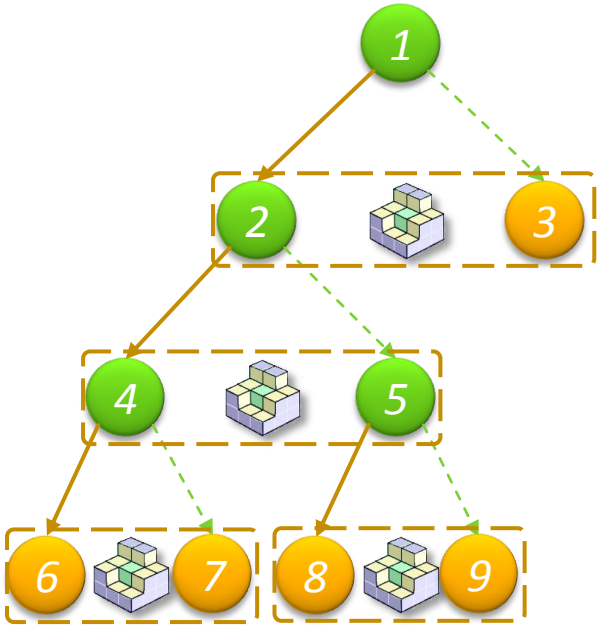
SVO STORAGE AND TRAVERSAL

SVO Structure

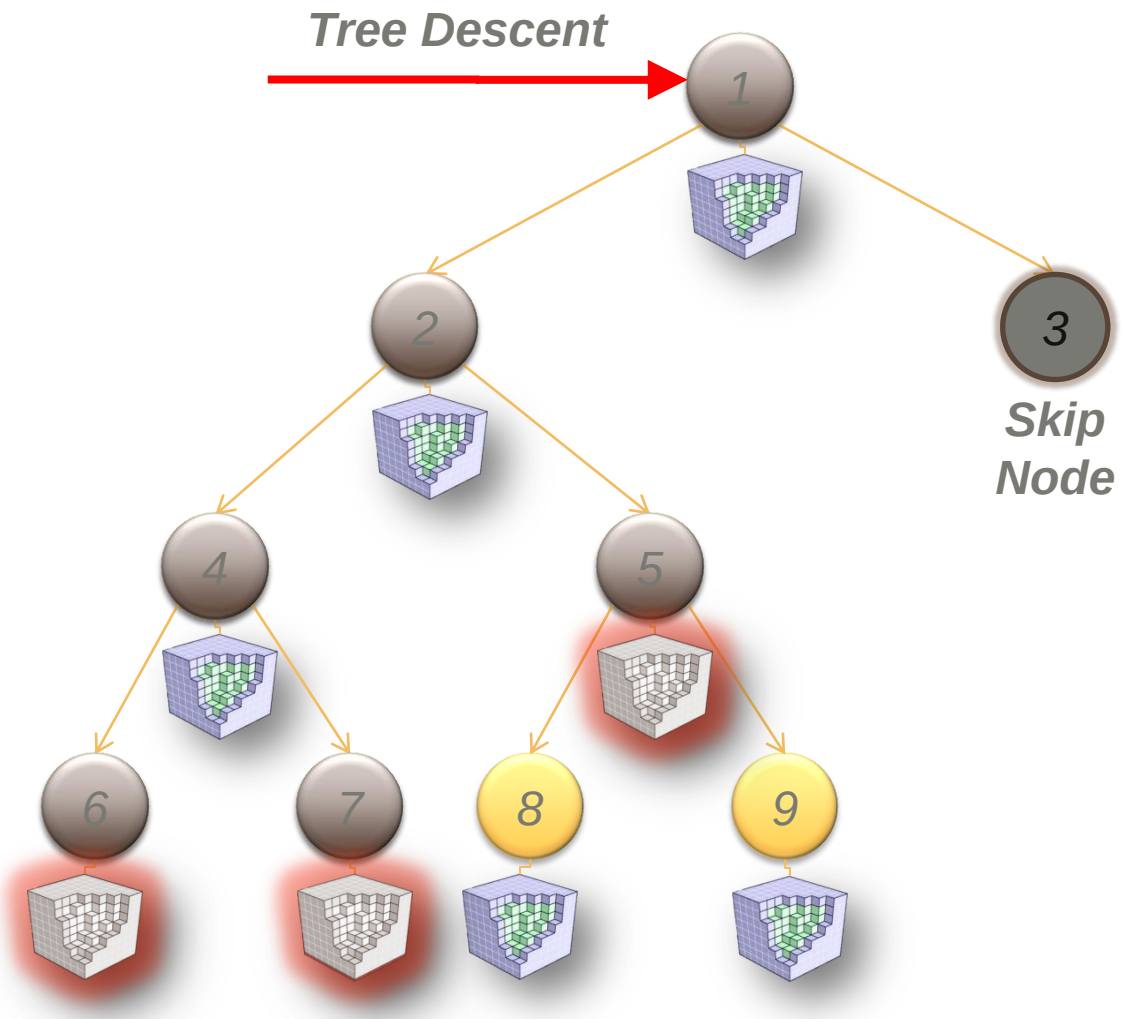


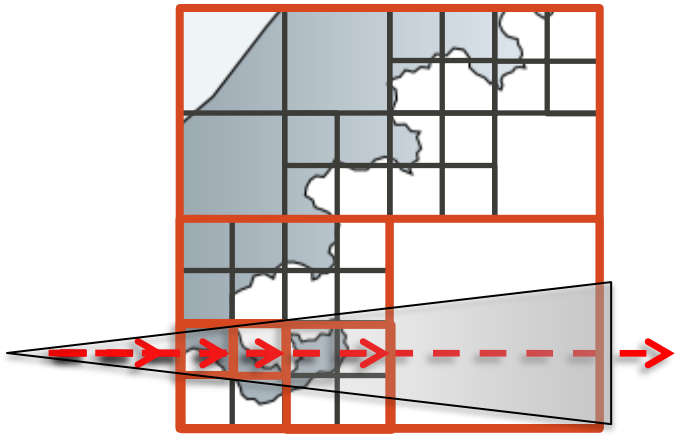
- **Nodes in linear video memory**
  - 2x2x2 nodes tiles
  - 1 pointer per node to a node-tile
- **Voxels stored into a 3D texture**
  - Allows hardware tri-linear interpolation





Octree traversal





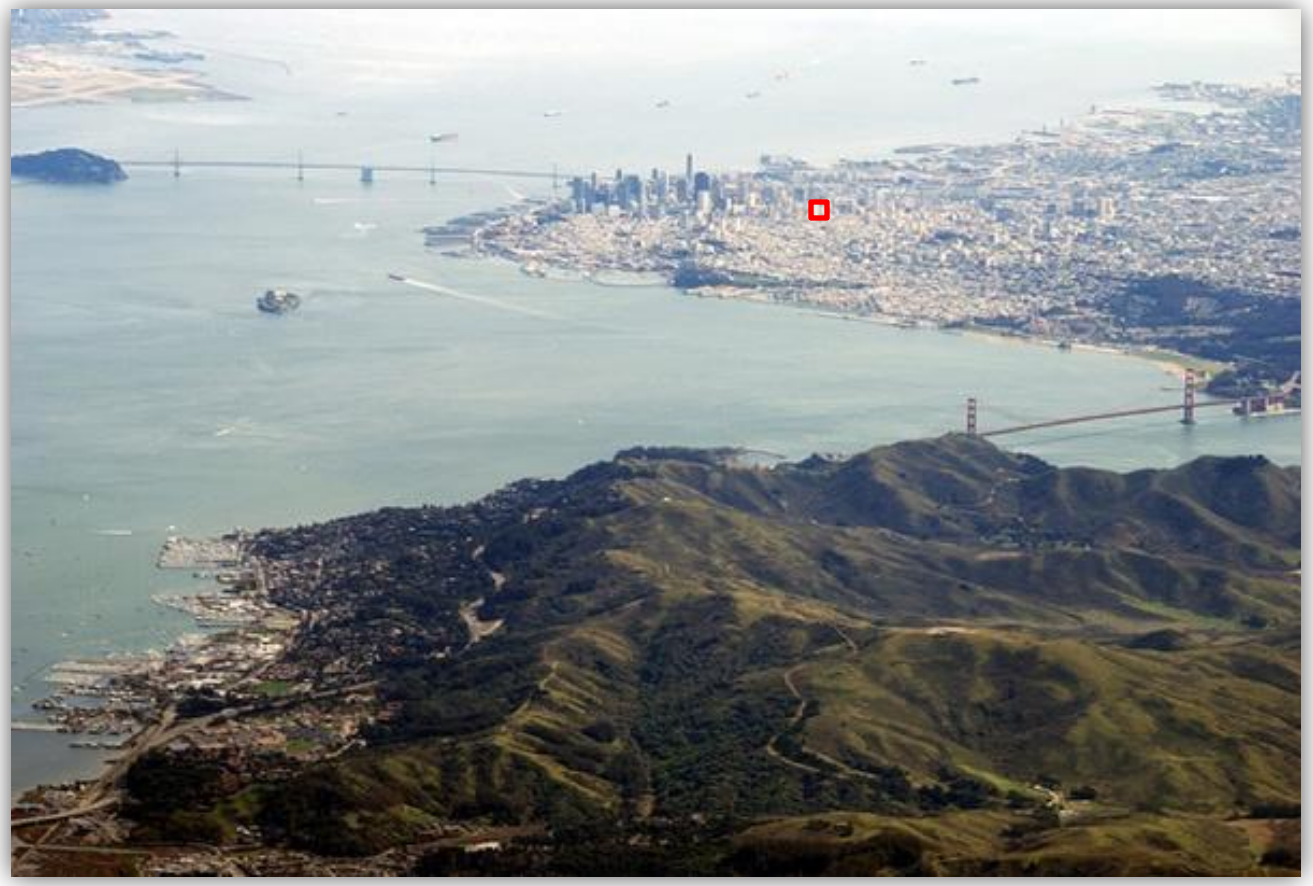
One thread/cone

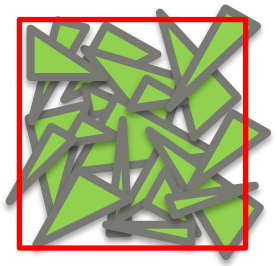
- **KD-restart**
  - Dependent accesses
- **Stack-based**
  - *Local thread storage*

Challenge #1

CINEMATIC IMAGE QUALITY

Massive detailed scenes

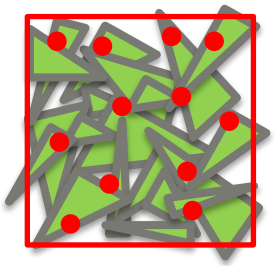




Pixel

Per-pixel integration (AA)

- **Supersampling / MSAA not scalable**
  - Will always be beaten by geometry



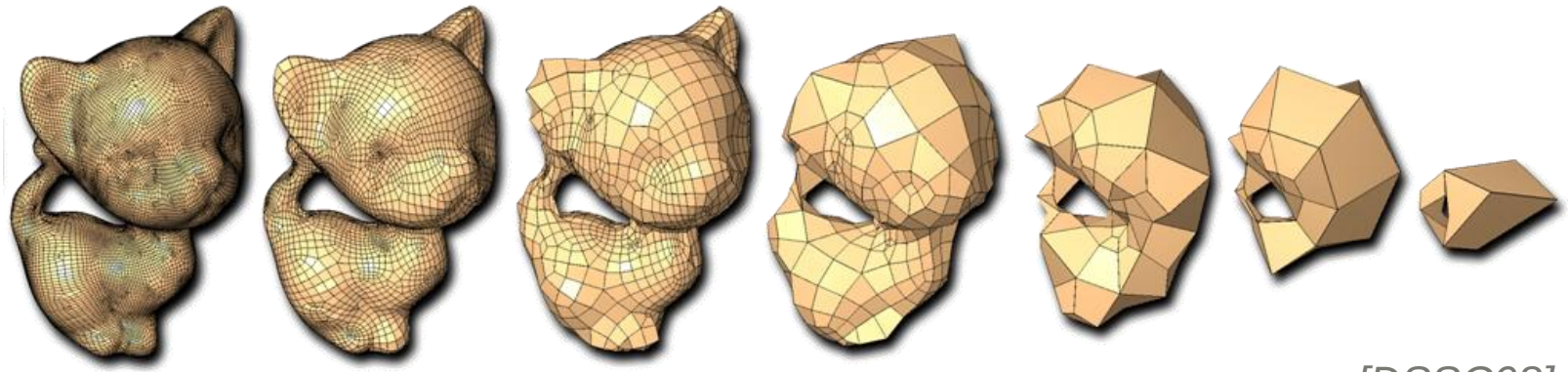
Pixel

- **Per-primitive transform + raster**
  - For mostly averaged geometries

Pre-filtering

- **Pre-filtering geometry is the only-way to go !**
  - Adapt geometry-complexity to sampling resolution

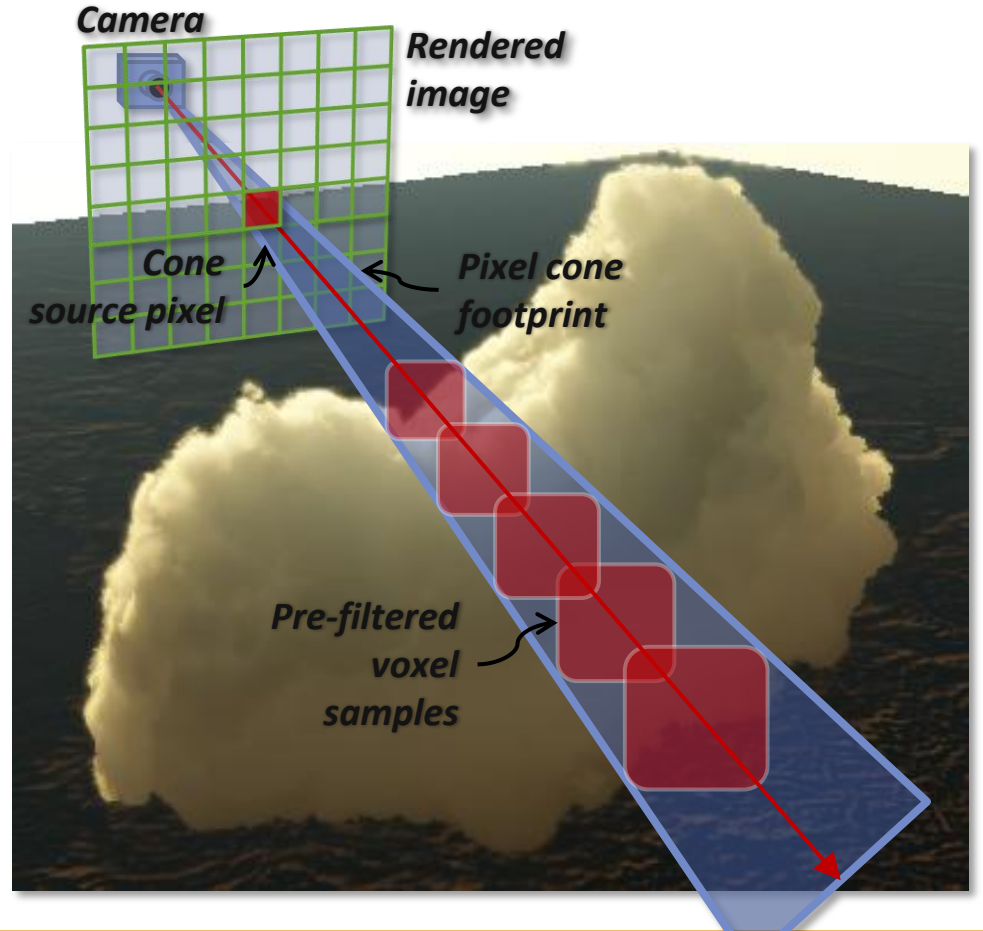
- **But B-reps don’t filter correctly ☹**



[DSSC08]

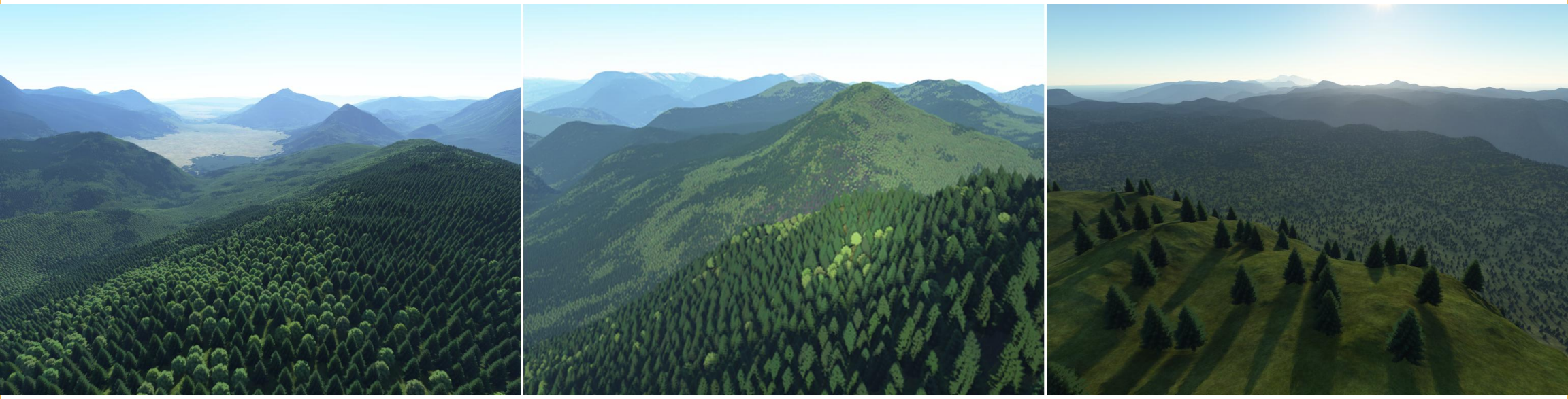
Voxel LOD

- **Scalable representation**
  - Unique rep + multi-scale
  - Continuous LOD
- **Very high voxel resolution ☹**
  - Animation difficult

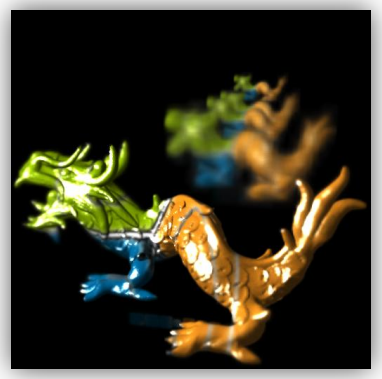
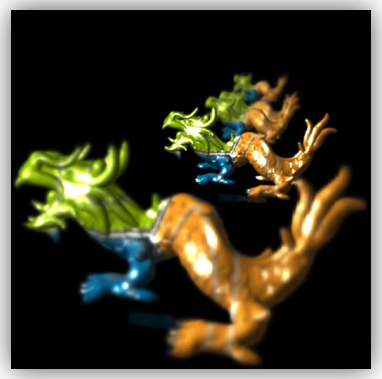
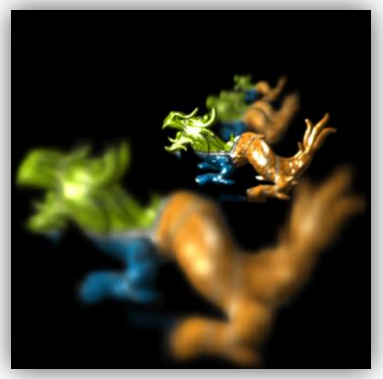
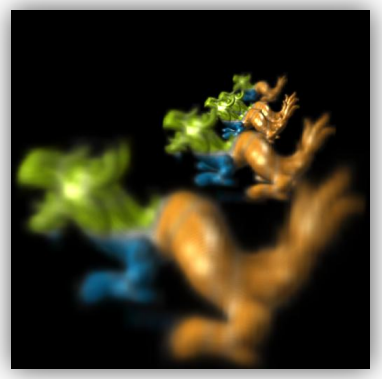


Forest rendering

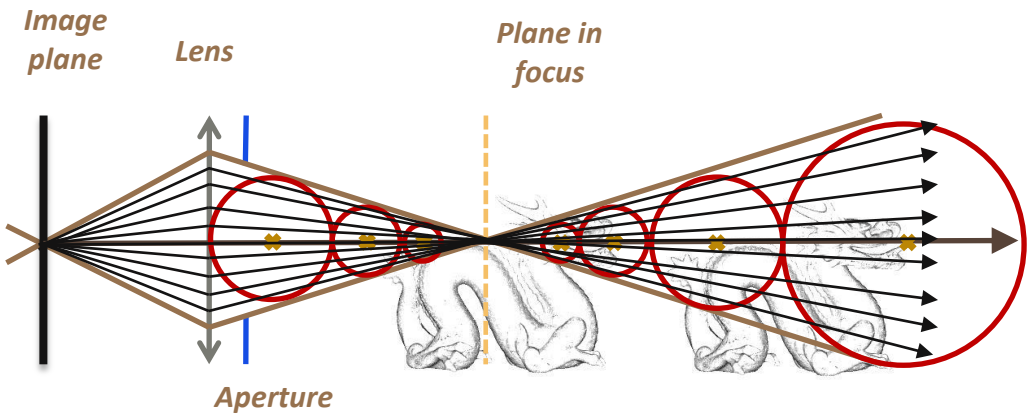
- One 3D texture/tree [Bruneton et al. 2012]
  - See PROLAND : *http://proland.inrialpes.fr*
- 51FPS @1024x768, **180,000 trees**. GTX 580
- Trees only : **10.6ms**



Voxel depth-of-field

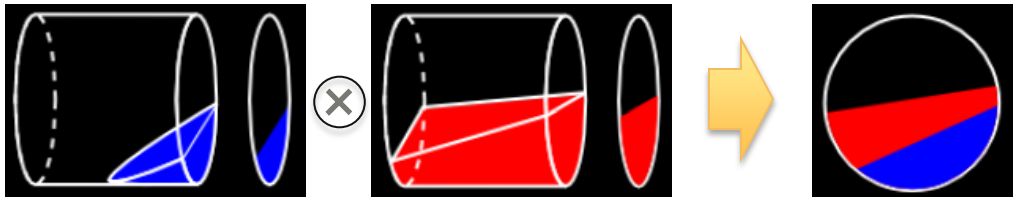
The blurrier, the faster !



Correlation problem

- ***Representing Appearance And Pre-Filtering Subpixel Data In SVOs***

  [Heitz and Neyret 2012]



- **3-10Mcones/s**
  - 100-300ms / frame @720p (1280x720)

Limitations

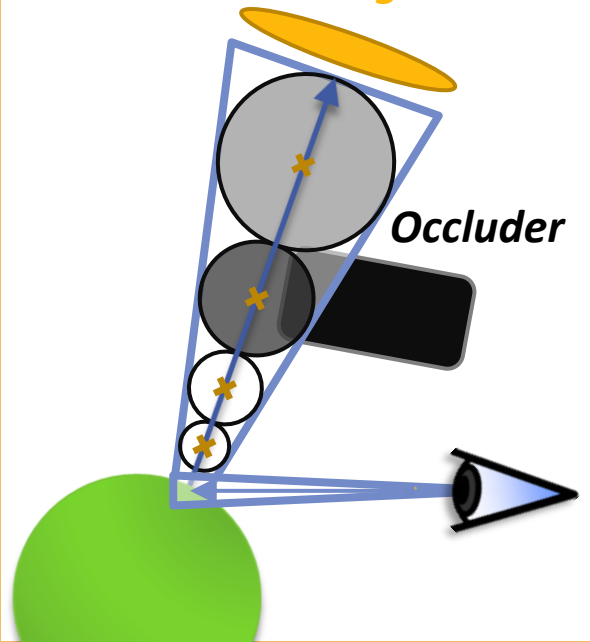
- **Primary rays require massive voxel resolutions**
  - Require out-of-core rendering
- **Animation is difficult**
  - Highly tessellated object in one voxel

Voxel soft shadows

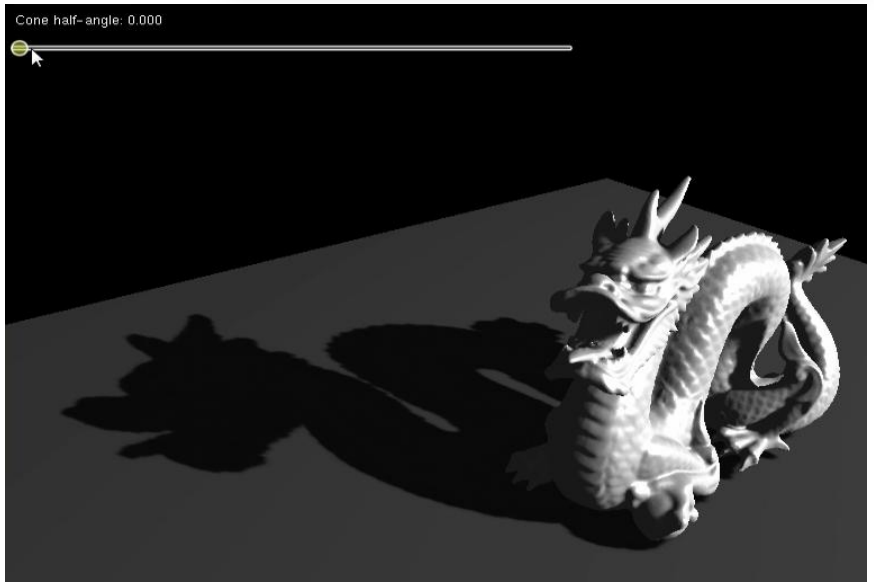
- **One cone per pixel**

  The smoother, the faster to compute !

Area light source

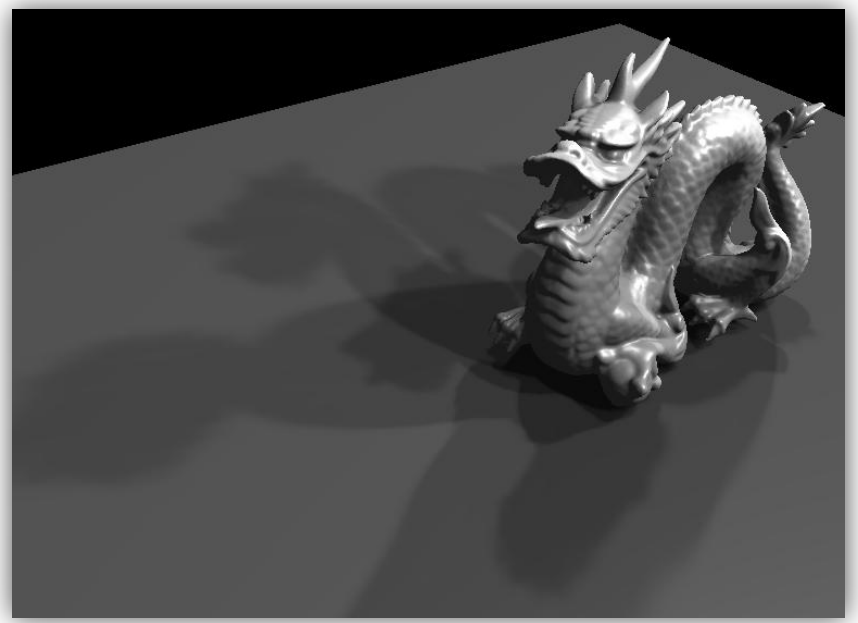


3-9ms @ 1280x720



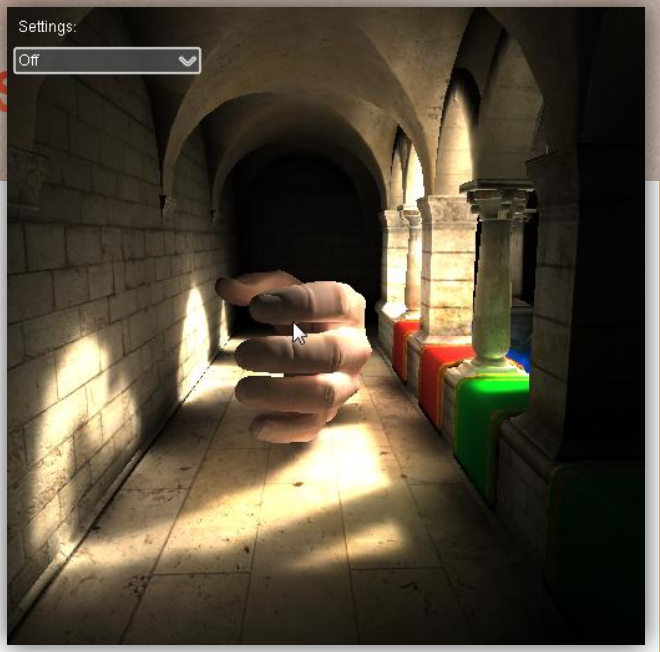
Voxel soft shadows

- **Multiple shadowing lights sources**
  - Only one geometry pass
  - Scales much better than shadow maps !

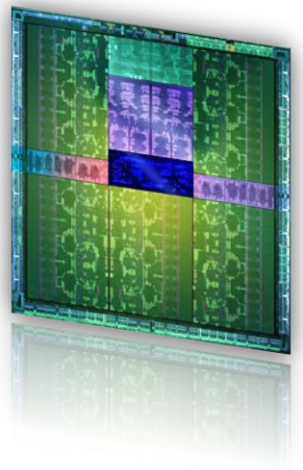



Settings:
Off
Voxel-Based GI
8ms @ 512x512 – 27ms @ 720p - 62ms @ FullHD

Rendering pipeline



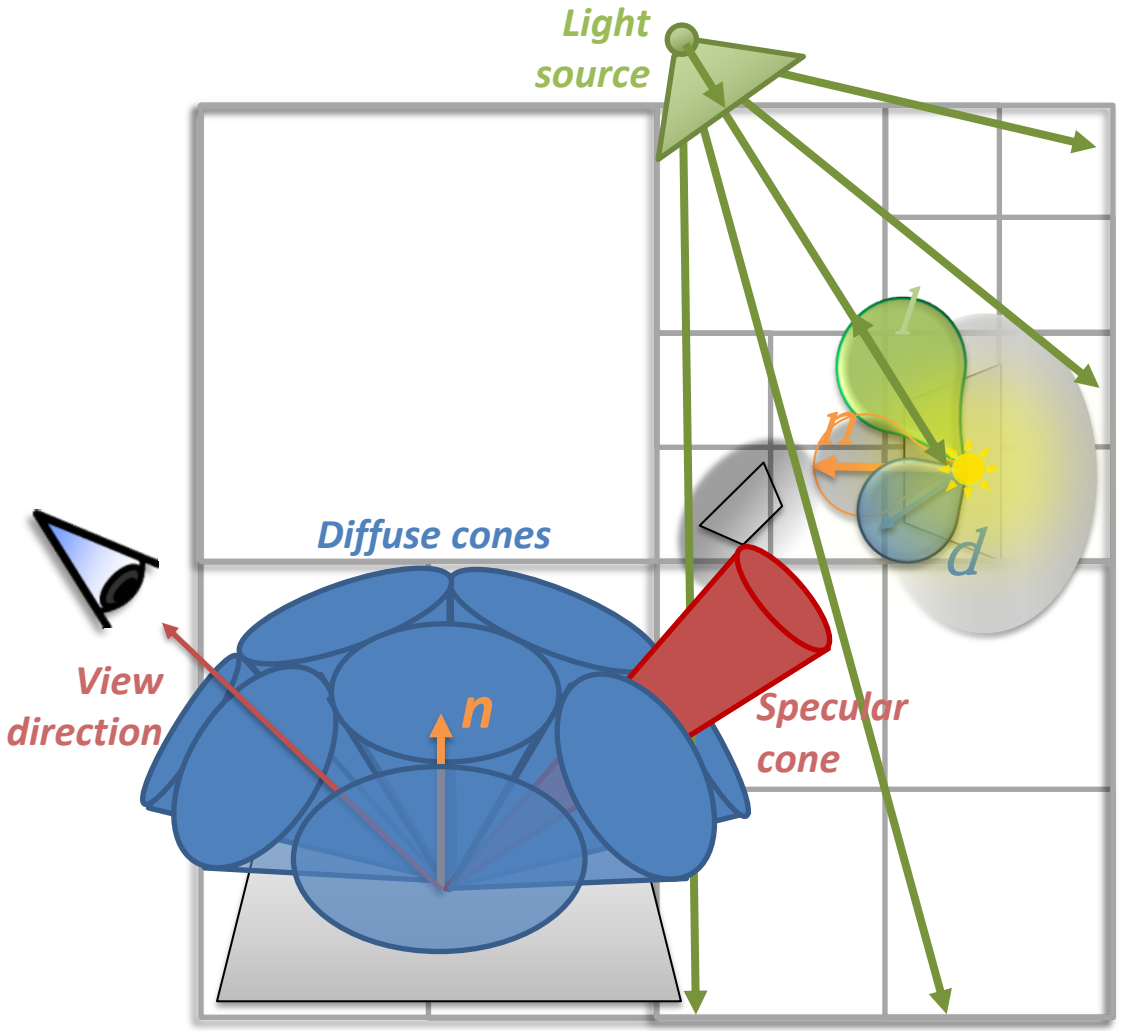
- **Hybrid** rendering pipeline
  - **Rasterized** primary rays
    - GPU pipeline optimized for direct visibility
  - **Cone-traced** secondary rays
    - Flexibility and scalability



- Forward or deferred rendering

Interactive Indirect Illumination Using Voxel Cone Tracing

Rendering algorithm

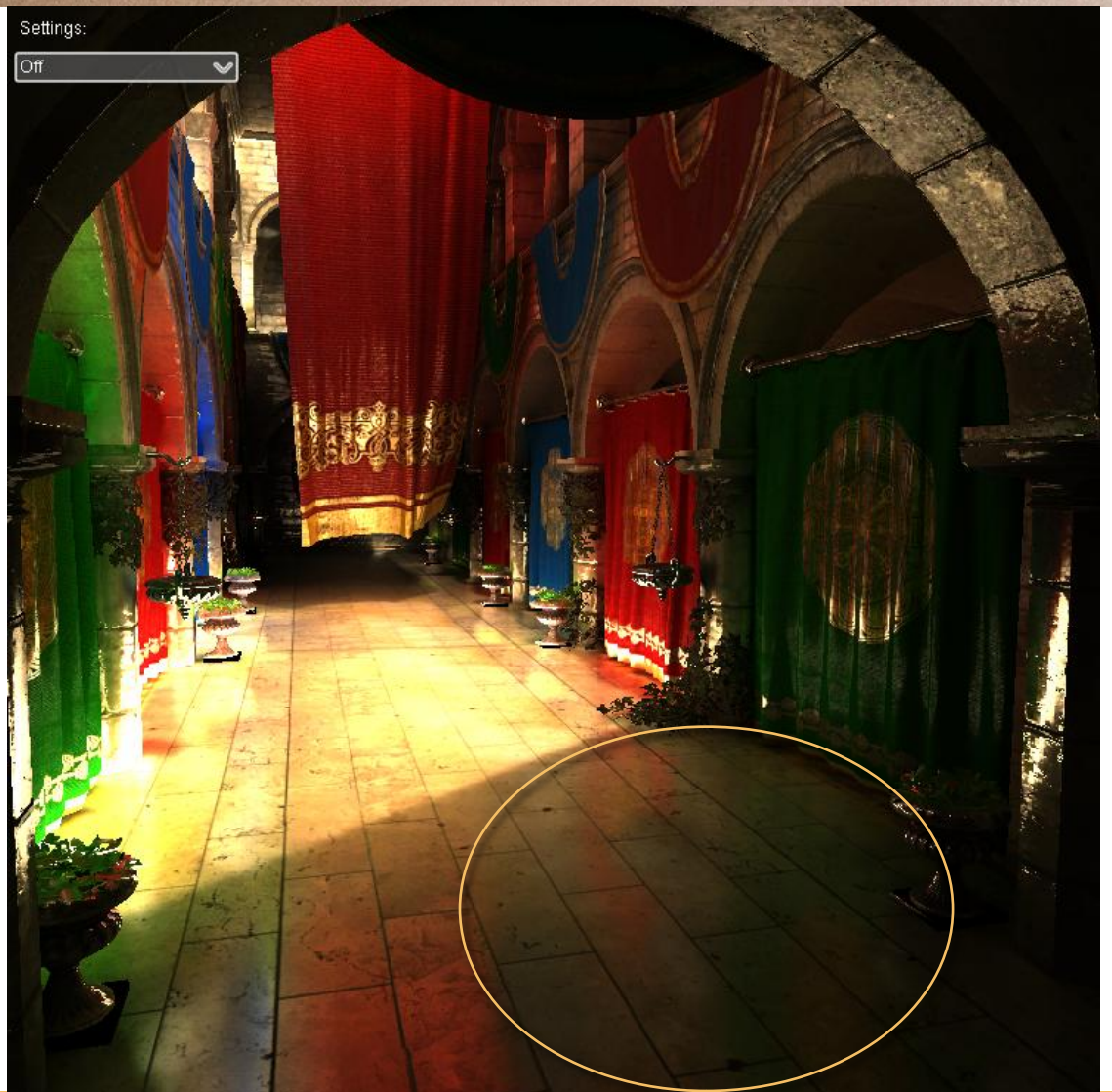


1. **Light injection**
   - *Rasterization (RSM)*
2. **Light filtering**
   - *Compute*
3. **Camera pass**
   (final gathering)
   - *Forward FS or deferred compute shader*

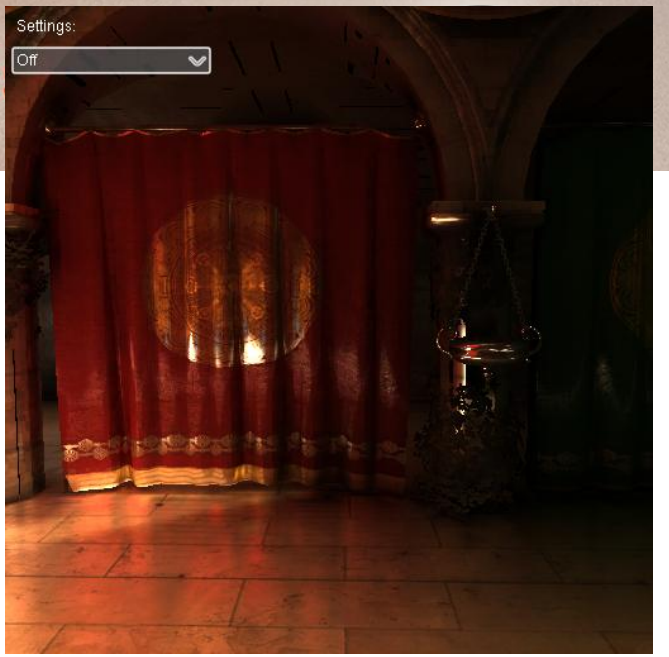
Importance sampling of the BRDF

Glossy reflections

Multiple-bounces



Performances and scalability

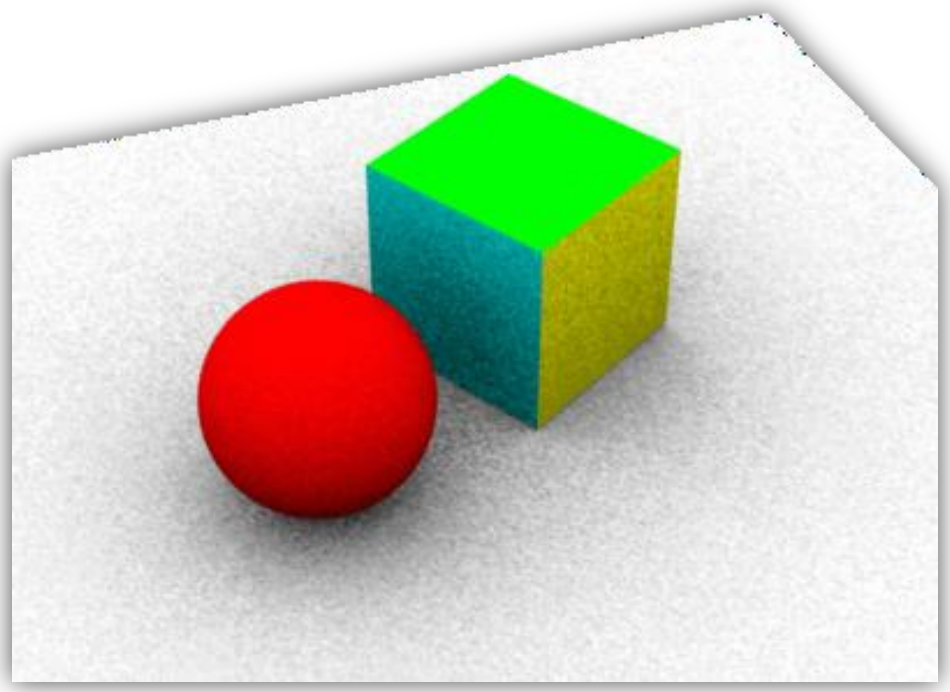


- **Scalable** lighting rep. !
  - **Independent** of geometric complexity
- **Control** over rendering time
  - Maximum voxel resolution
    (*Number of octree levels)*
  - Number of cones per pixel / Aperture of the cones: The wider, the faster !

➢ Graceful performance degradation

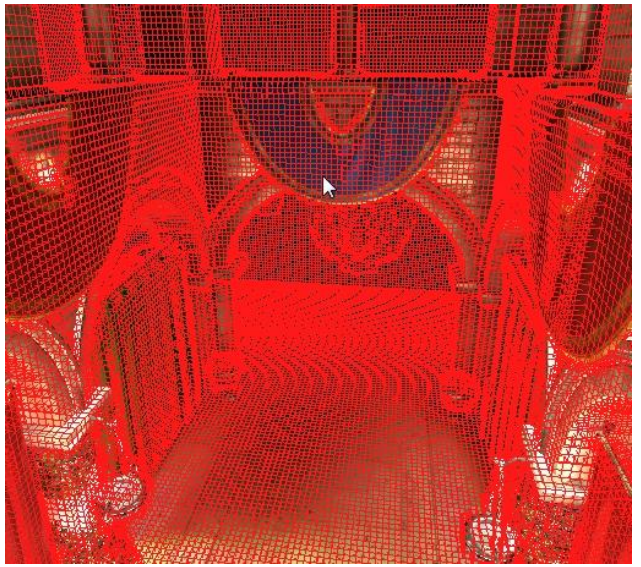
Discussion

- **Large cones**
  - Precision / Light leaking
- **But never noisy !!**



What is the cost of an SVO ?

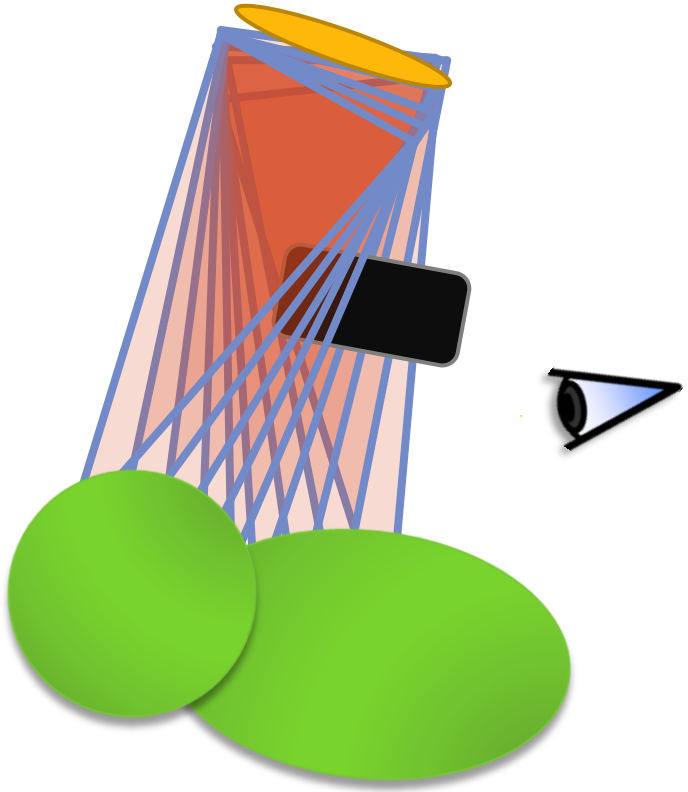
GI Sponza Demo



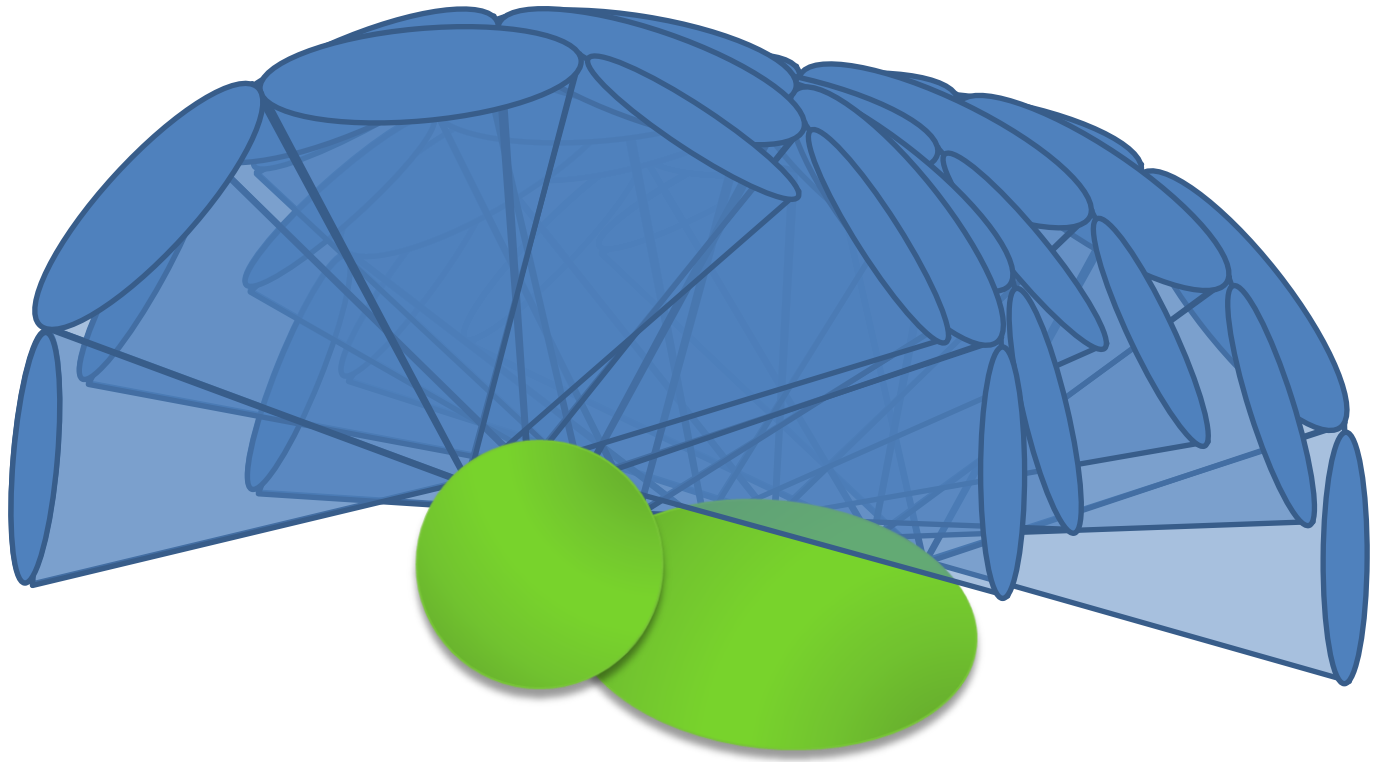
- **Memory consumption:**
  - 9 levels SVO: ~200MB-1GB
    - + Temporary buffers for building
- **Construction + Update Time *(GK104)***
  - Construction : ~70ms at initialization time
  - Update: ~4-5ms / frame

Traversal coherency

- **Coherency**: Execution + Data access

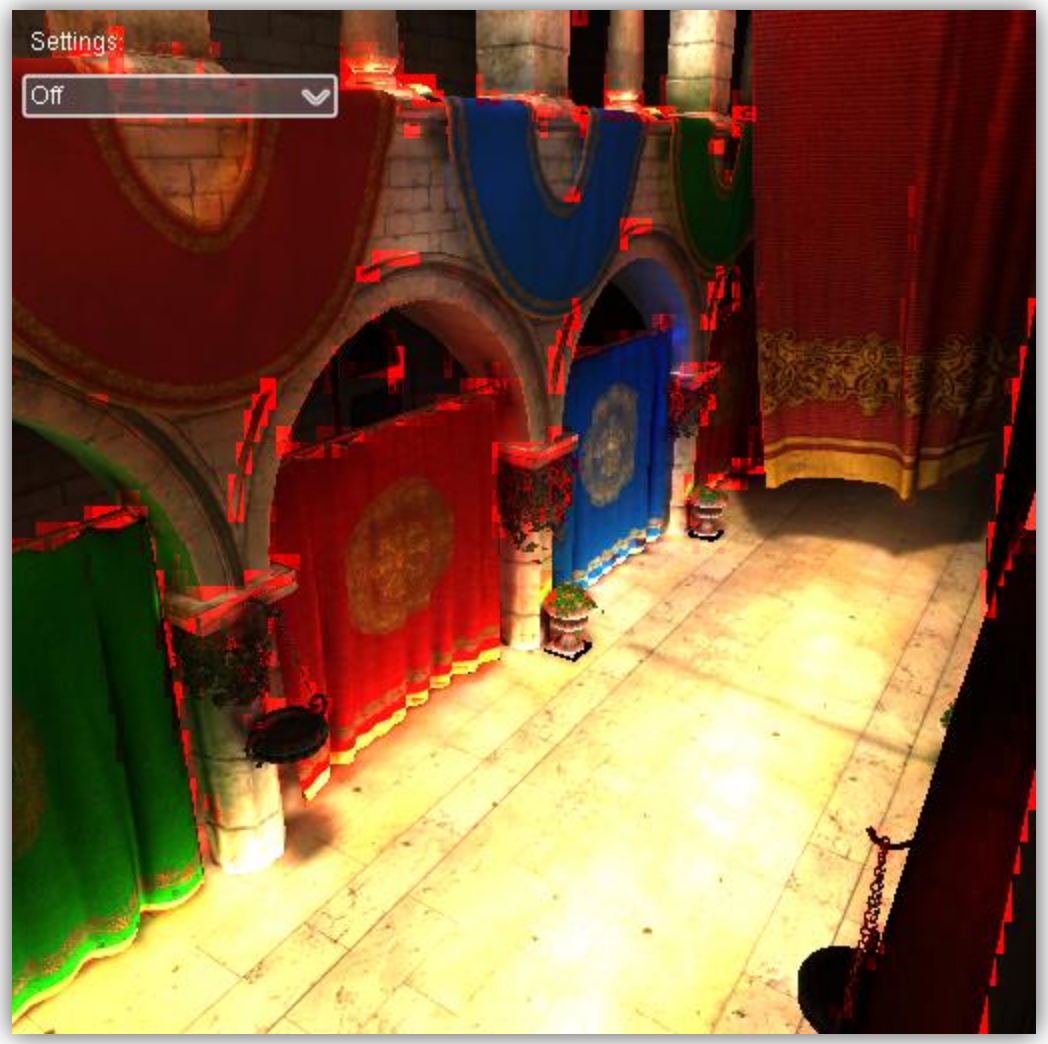


Soft Shadow



Diffuse Tracing

Traversal coherency



Diffuse Only

Diffuse + Specular

Voxel-based GI

- This can run in a game !
  - SVOgi

The Technology Behind the "Unreal Engine 4 Elemental Demo"

Martin Mittring

- ***Advances in Real-Time Rendering in Games: Part II***

Wednesday, 8 August 2:05 pm - 3:05 pm

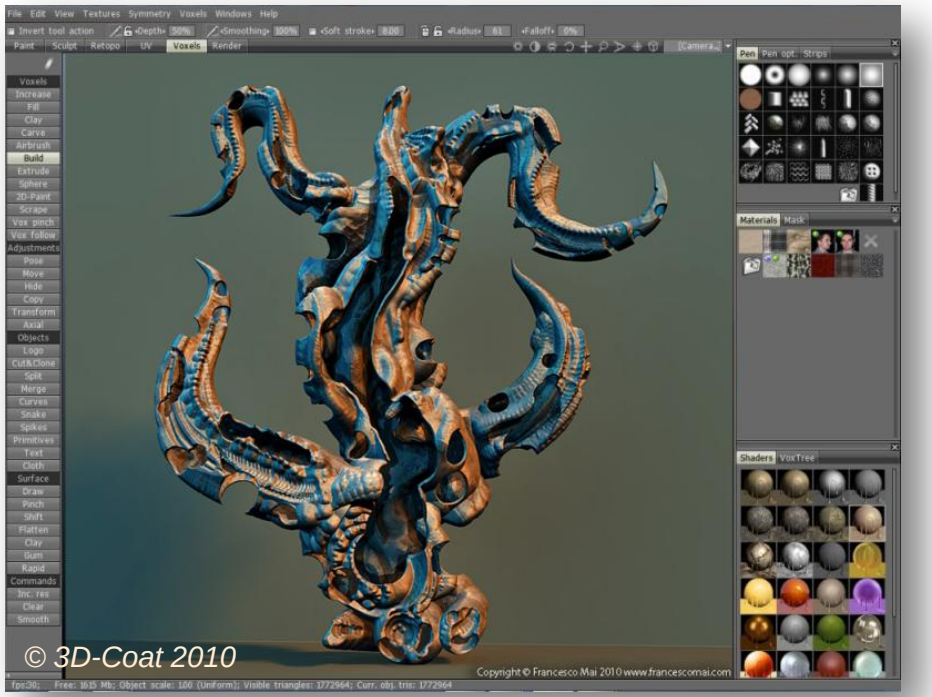
CONTENT CREATION

Procedural content generation

- Signed **Distance Field**
  - Procedural generation and Amplification

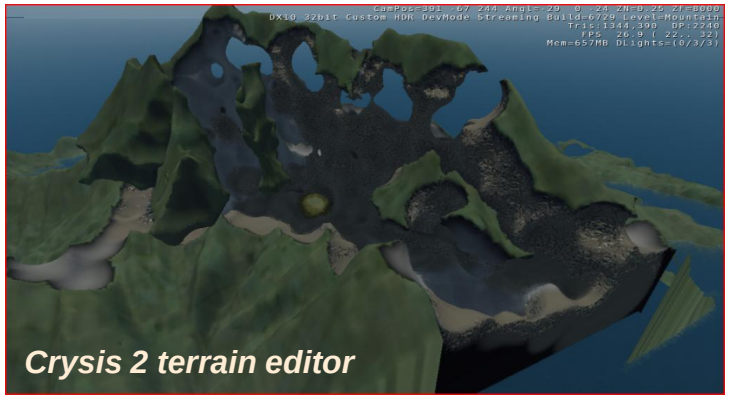
Voxel sculpting

- **CSG operations**
  - 3D-Coat



Procedural content generation

- **LODs**
  - Eg. Terrain generation in Crysis
  - No problem of topology



OTHER COOL USAGES

World-space light baking

- **Brick maps**

  [Christensen and Batali 2004]

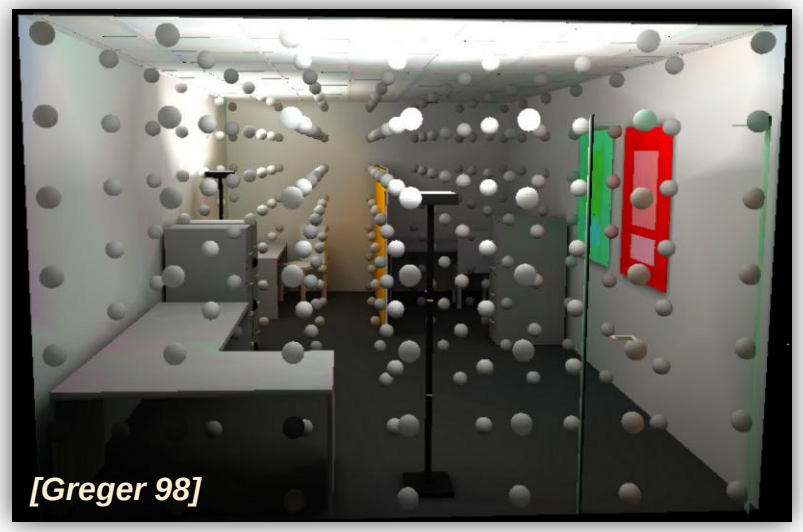
  - Bake static lighting

- **Irradiance volumes**

  [Greger et al 1998]

  - Volume of diffuse lighting samples
  - Pre-computed Radiance Transfer

    [Sloan et al. 2002]



Participating medias

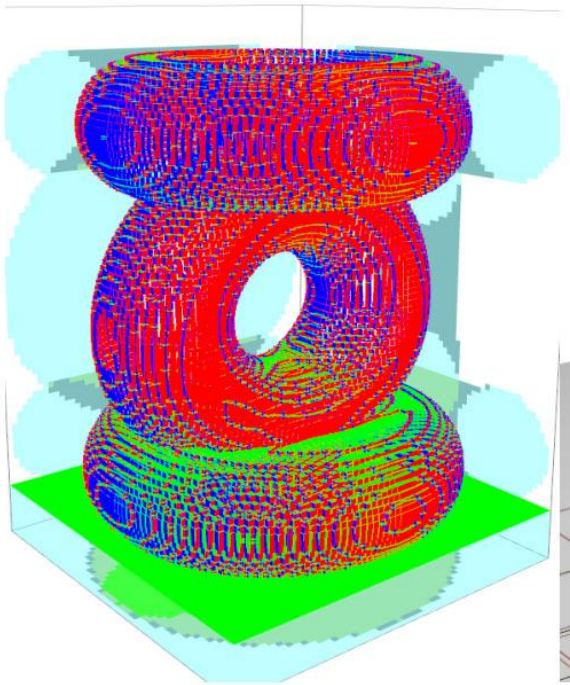
- **Rendering volumetric effects**
  - Smoke, clouds…

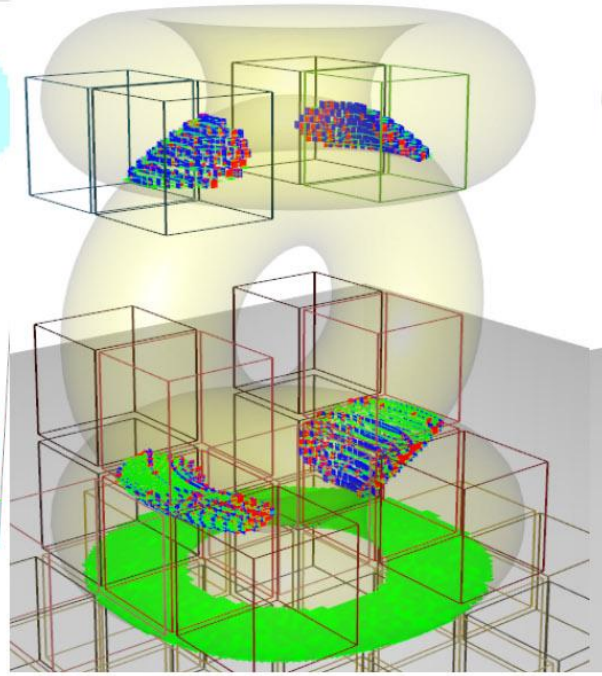
[Greger 98]

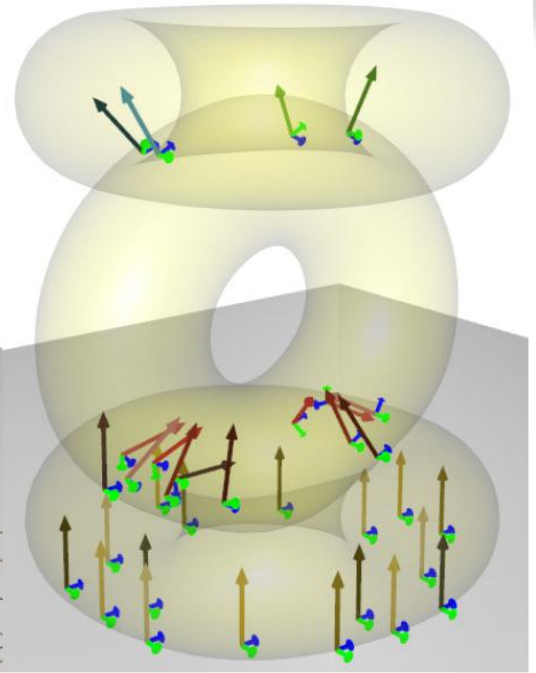
Other applications

- Physics: Collision detection

Allard et al. Siggraph 2010

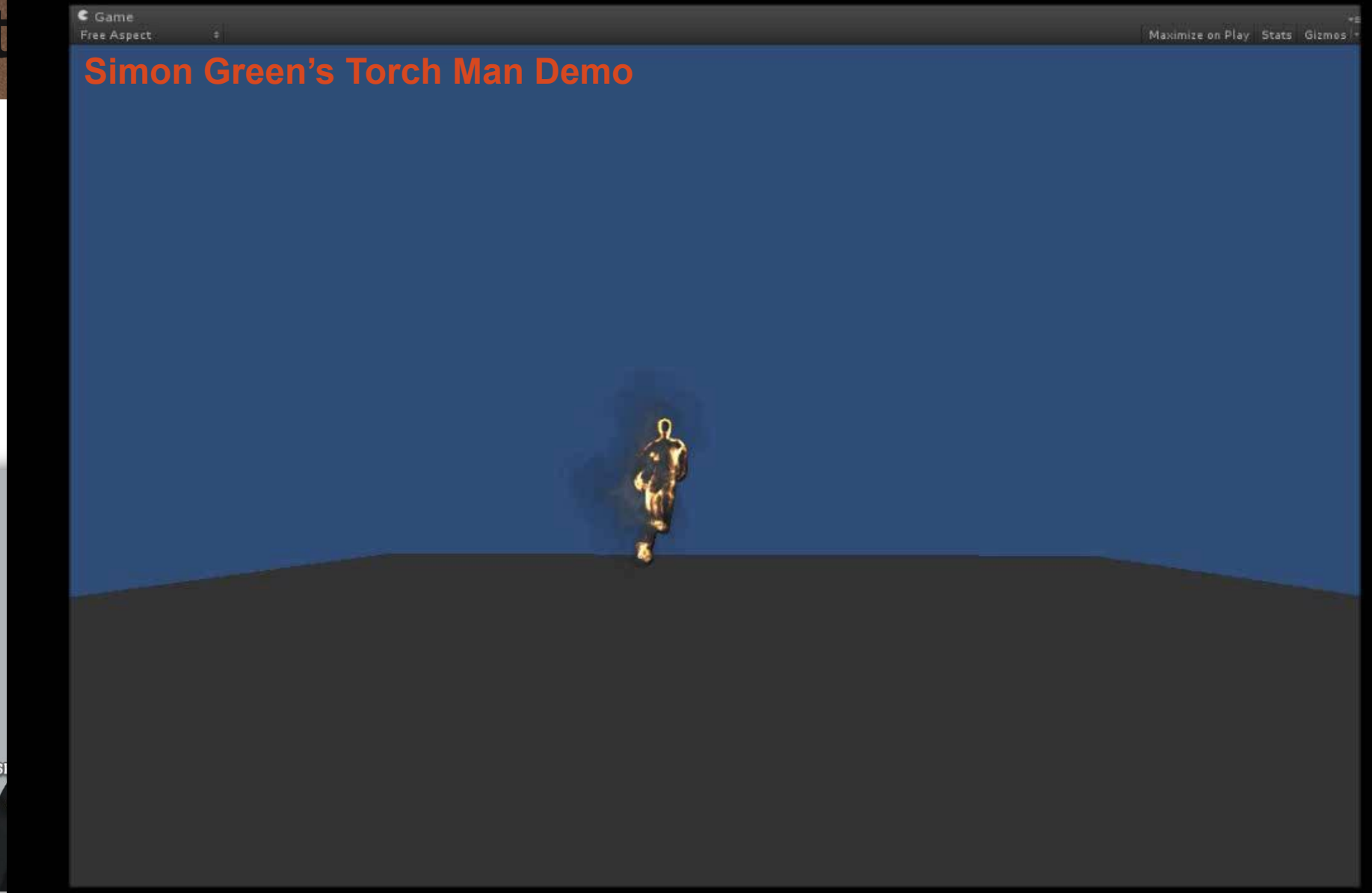


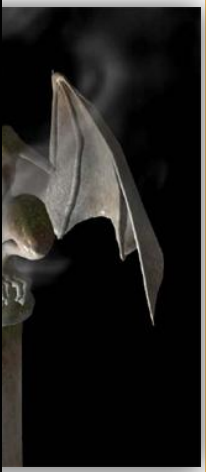


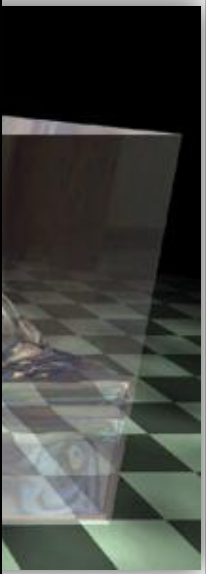


Ot

-
-


Game
Free Aspect
Maximize on Play
Stats
Gizmos
Simon Green's Torch Man Demo





DISCUSSIONS AND DIRECTIONS

Limitations

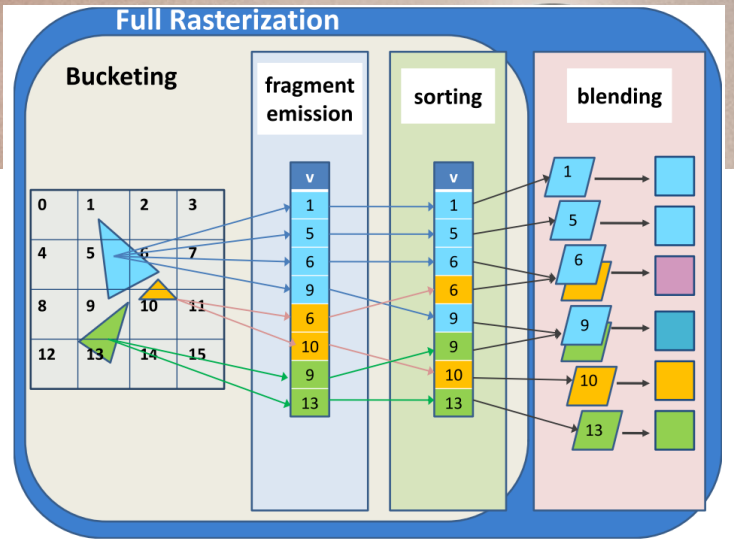
- **Animation and dynamic content**
  - If you use the geometry enough time to amortize, it is fine !

Voxelization

- ***Compute*-based**

  [Schwarz and Seidel 10, Pantaleoni 11]

  – Not using hw rasterizer

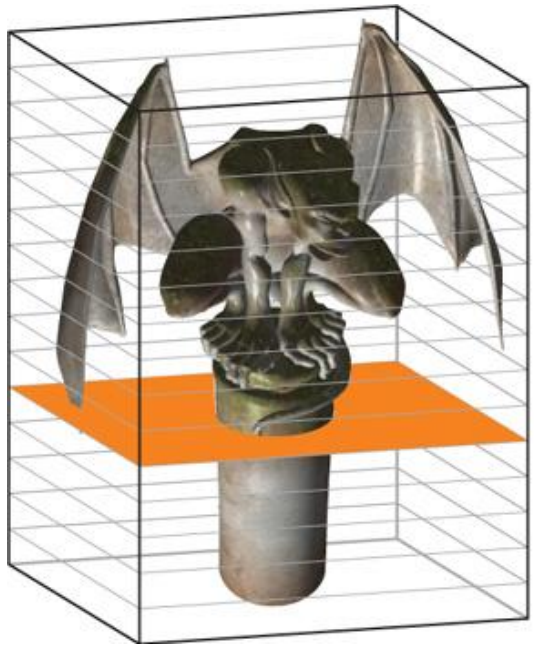


VoxelPipe [Pantaleoni 11]

- **Multi-pass *graphics*-based**

  – Slice-by-slice / Multiple-slices MRT

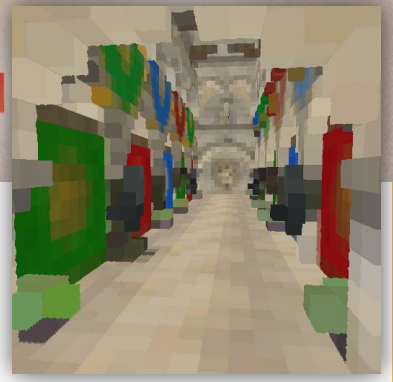
  [Fang et al. 00, Crane et al. 07, Li et al. 05,
  Dong et al. 04, Zhang et al. 07, Eisemann and Decoret 08]



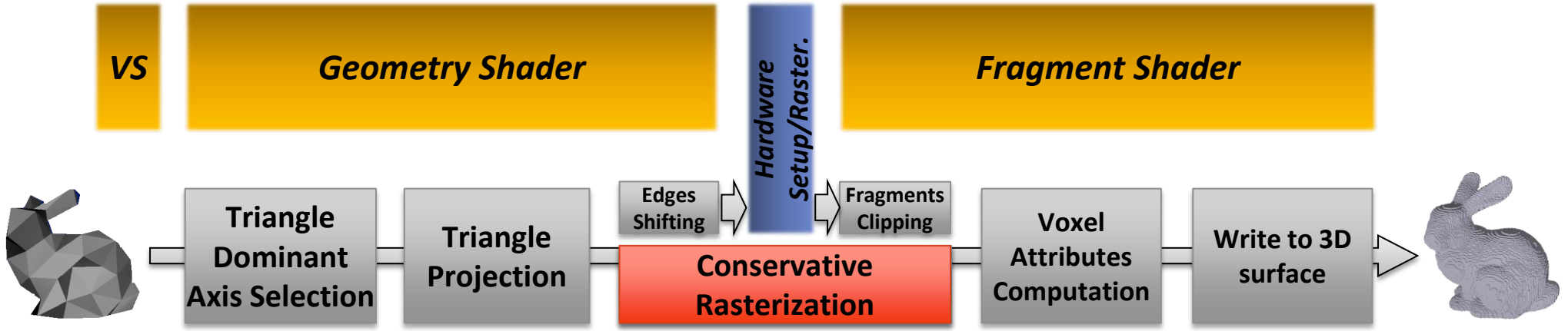
Crane et al. 2007

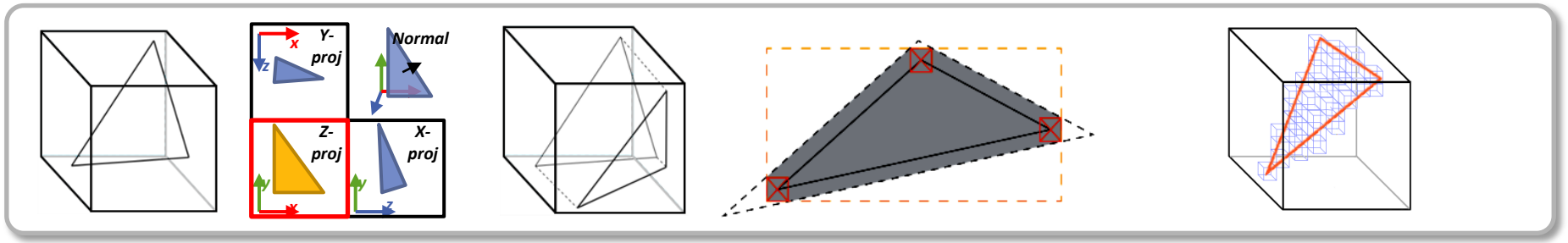
Single Pass Dense Voxelization

SIGGRAPH



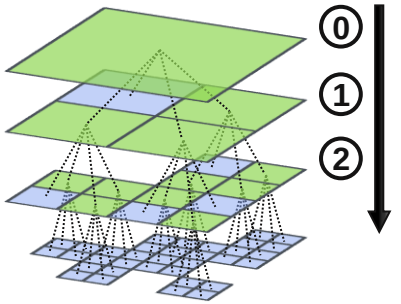
- Thin surface / Classical conservative

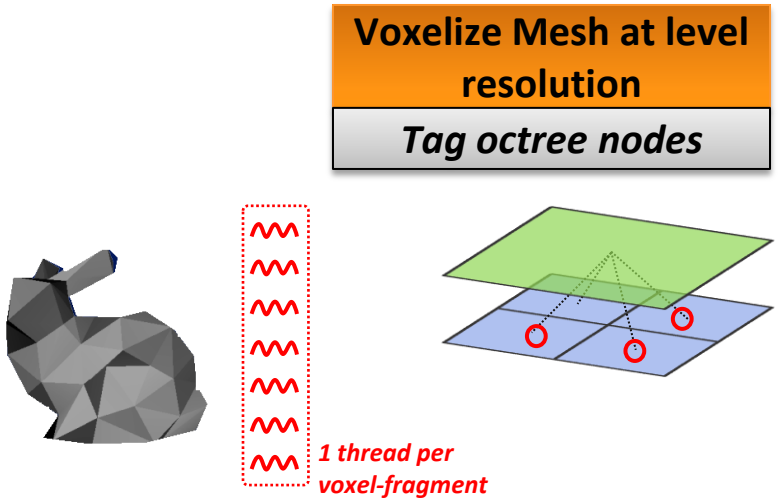


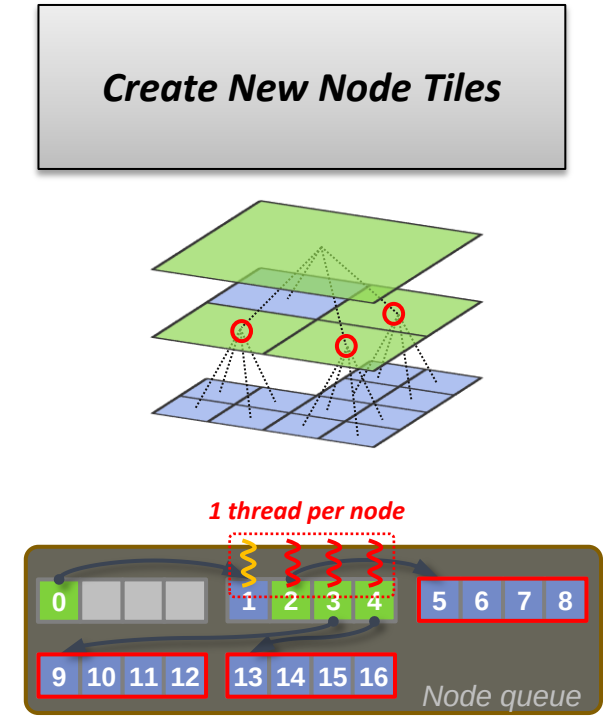


Octree construction

- **Top-down** octree construction
  - Compute + Graphics







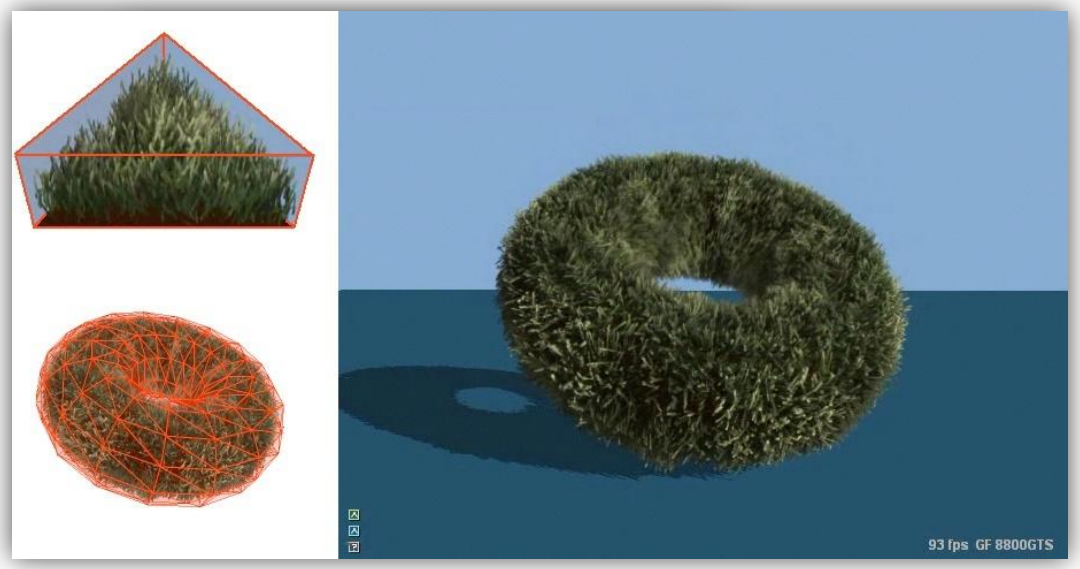
Performances

- **9 levels octree (512^3)**
  - RGBA32F
- **Kepler GK104 performance**
  - **30% - 58%** faster than Fermi GF100

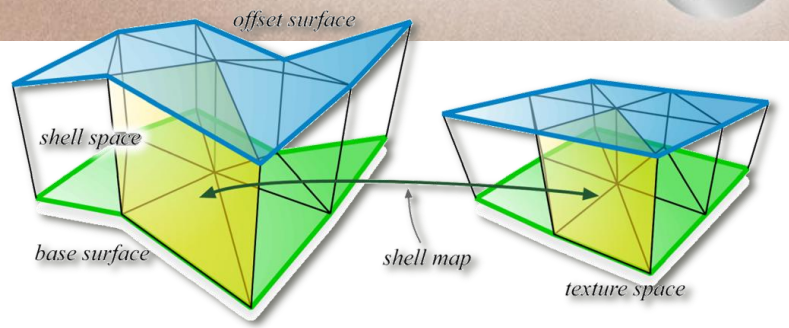
Times in ms	Frag list	Octree construction				Write	MIP map	Total
Scene		Flag	Create	Init	Total			
Sponza	2.07	5.65	0.37	1.32	7.34	3.94	2.09	15.44

Voxel deformation

- How to animate large scale of small details ?
  - FFD deformation



[Decaudin and Neyret 2009]



Shell Maps [Porumbescu et al. 05, Jeschke et al. 07]

Limitations

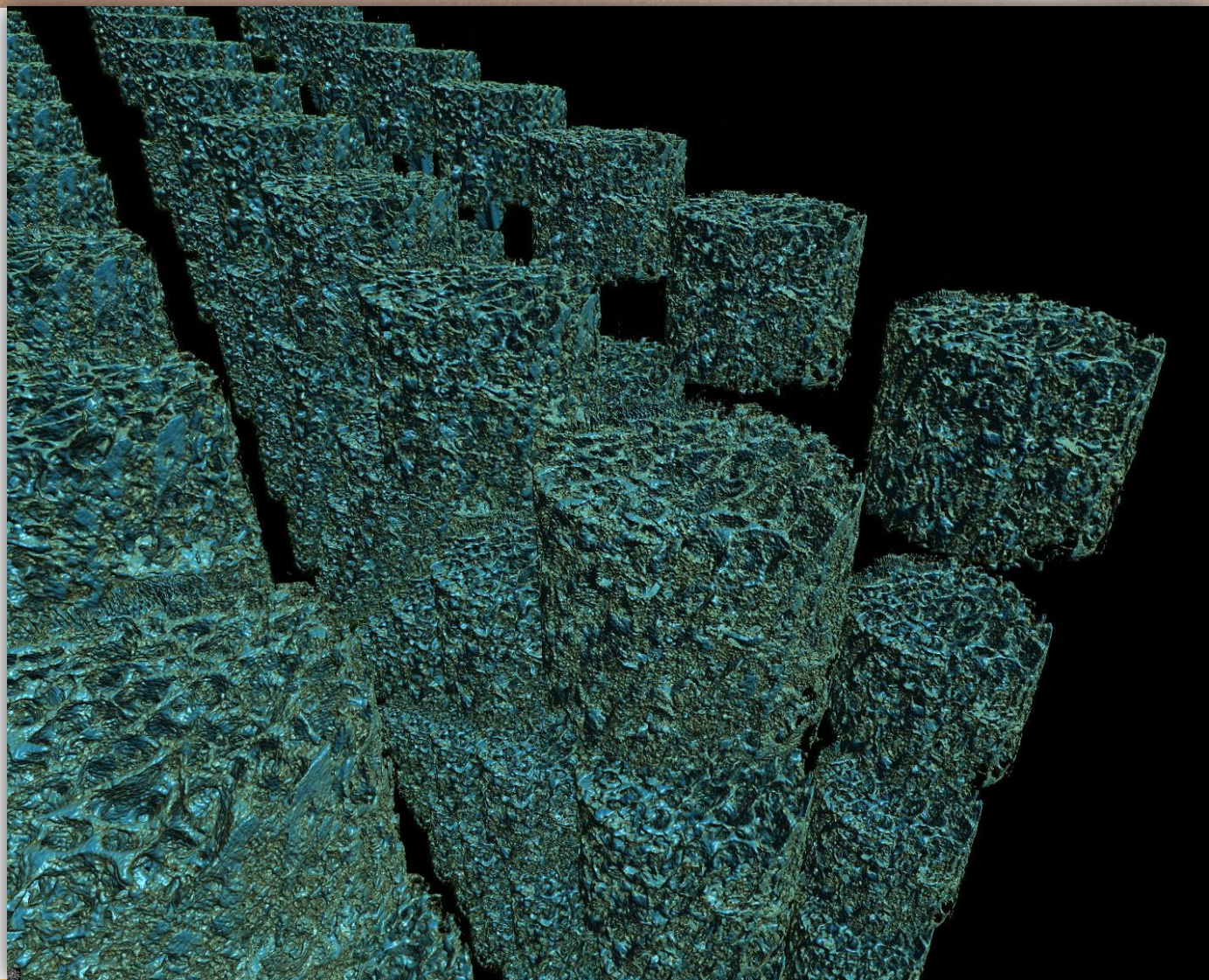
- **Memory cost of large scenes**
  - Needs streaming or dynamic re-voxelization
    - Well fitted to streaming

Challenges

- **Problem: Primary rays require massive amounts of voxels !**
  - Large scenes + details (screen resolution @ all necessary scales and everywhere)
- **Dynamic streaming can be affordable**
  - Ideal case: 2/3 voxels per pixel @1080p
    1920x1080 x3 x32B/voxel ~= 200MB

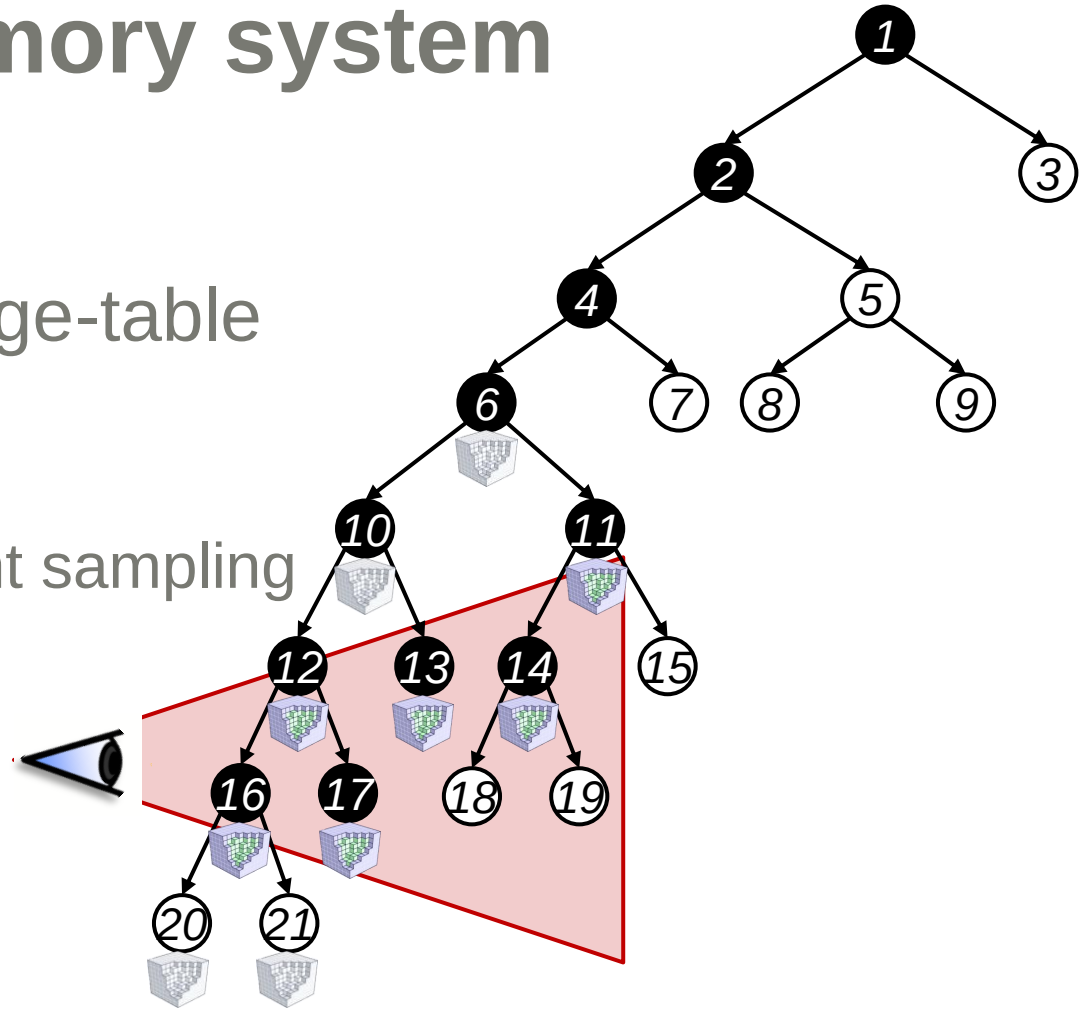
Massive scenes

- **8K^3** virtual resolution
- **50ms** @512x512
  8800 GT *(2007)*
  ~5Mcones/s



Out-of-core

- **Virtualized virtual memory system**
  - Dynamic paging
  - Octree as a hierarchical page-table
    - Virtually unlimited resolution
    - At the cost of a log time point sampling



Acknowledgements

- Johan Andersson
- Elmar Eisemann
- Aaron Lefhon
- David Luebke
- Yury Uralsky
- Ignacio Llamas

For helpful suggestions and discussions

Thank you !

- **Contact details**
  - *Email:* ccrassin@nvidia.com
  - *Website:* http://www.icare3d.org
  - *Twitter:* @Icare3D