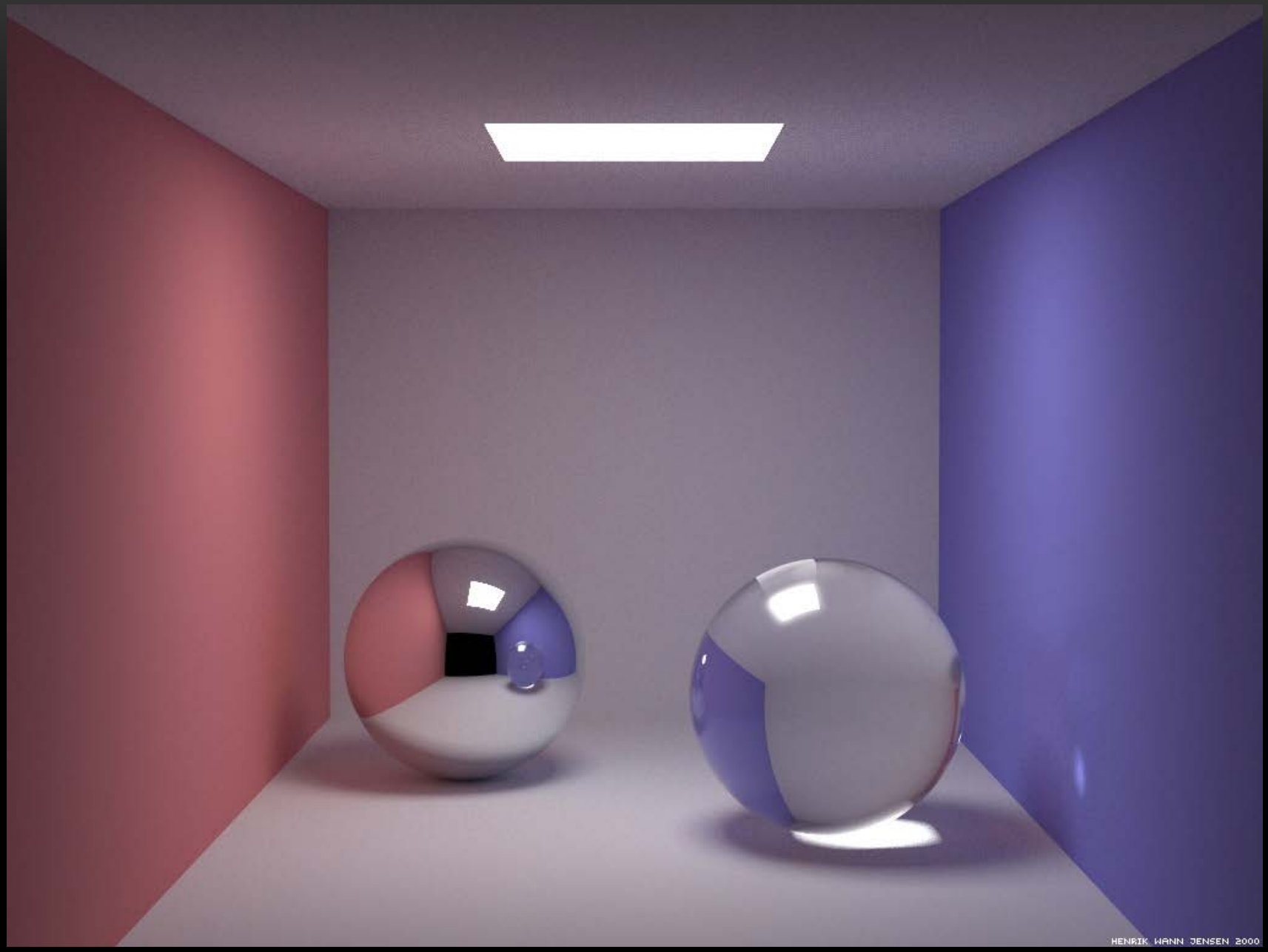
REAL-TIME VOXEL CONE-TRACING

Ian Lilley, Sean Lilley, Nop Jiarathanakul

realtimevoxels.blogspot.com

MOTIVATION

Why do we do this?



http://graphics.stanford.edu/~henrik/images/global.html

http://www.ea.com/


4
UNREAL
TECHNOLOGY

http://www.unrealengine.com/unreal_engine_4/

http://www.unrealengine.com/unreal_engine_4/

The Technology Behind the Elemental Demo. SIGGRAPH 2012

The Technology Behind the Elemental Demo. SIGGRAPH 2012

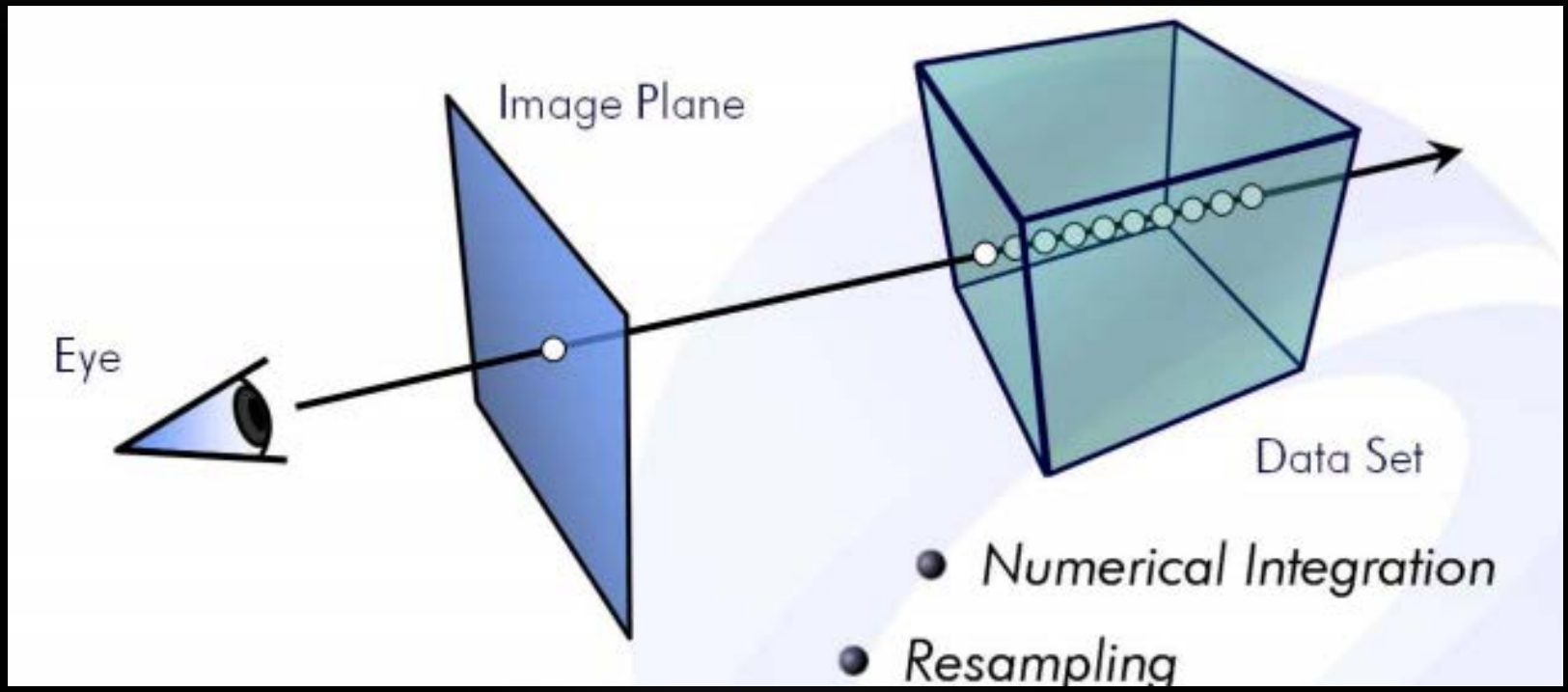
METHOD

How is it done?

Voxel Cone Tracing

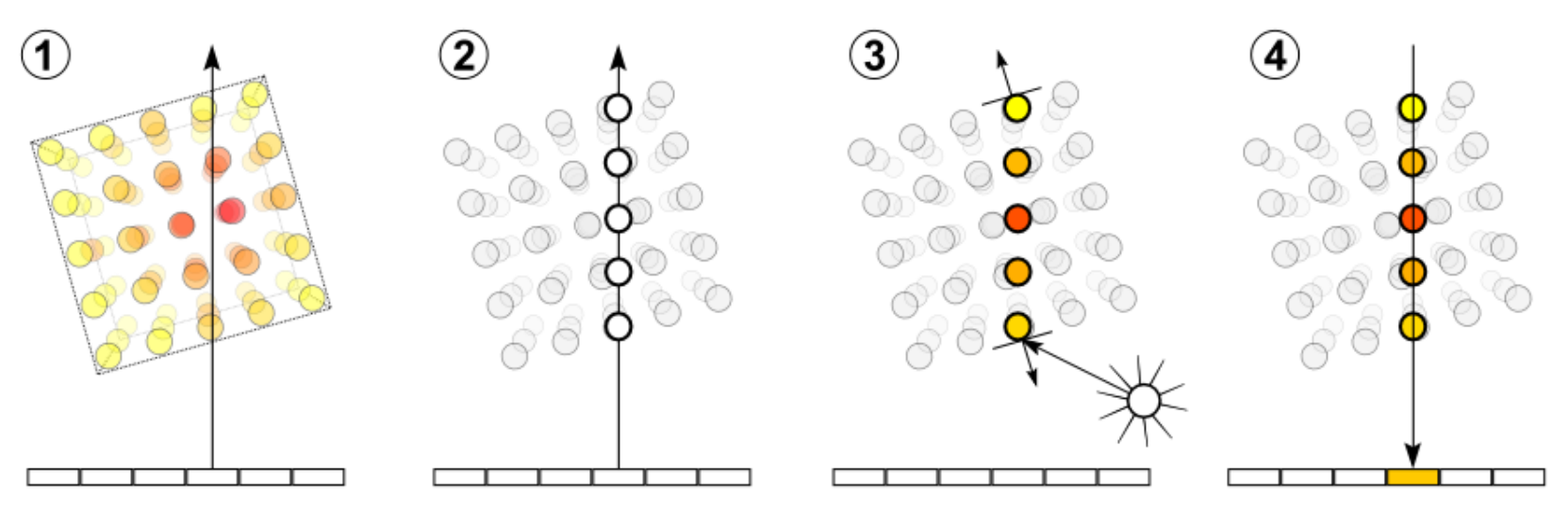
http://www.icare3d.org/research-cat/publications/interactive-indirect-illumination-using-voxel-cone-tracing.html

Volume Ray Marching



Real-time Volume Graphics. SIGGRAPH 2004

Volume Ray Marching

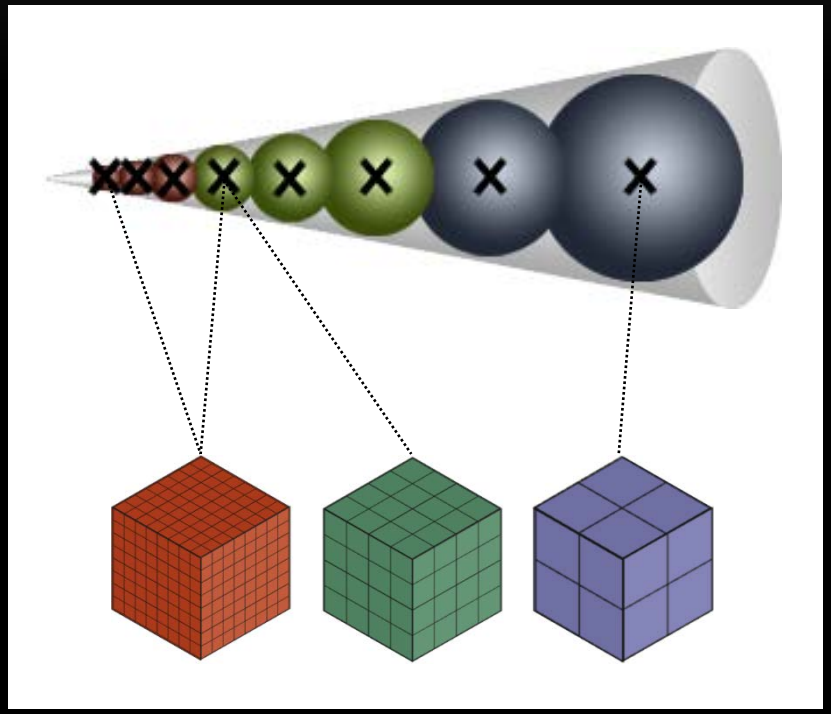


http://en.wikipedia.org/wiki/Volume_ray_casting

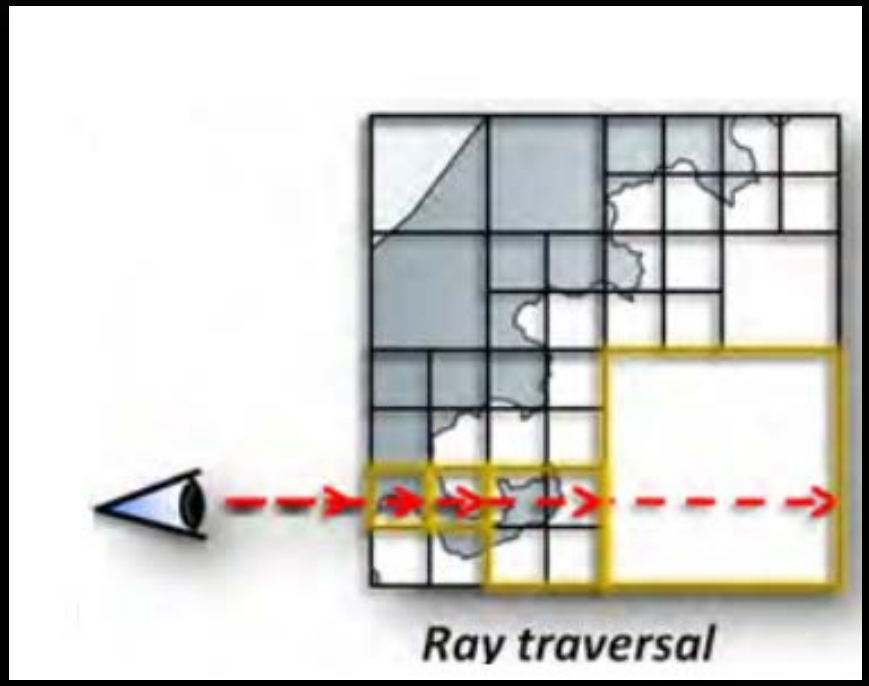
Voxel Cone Tracing

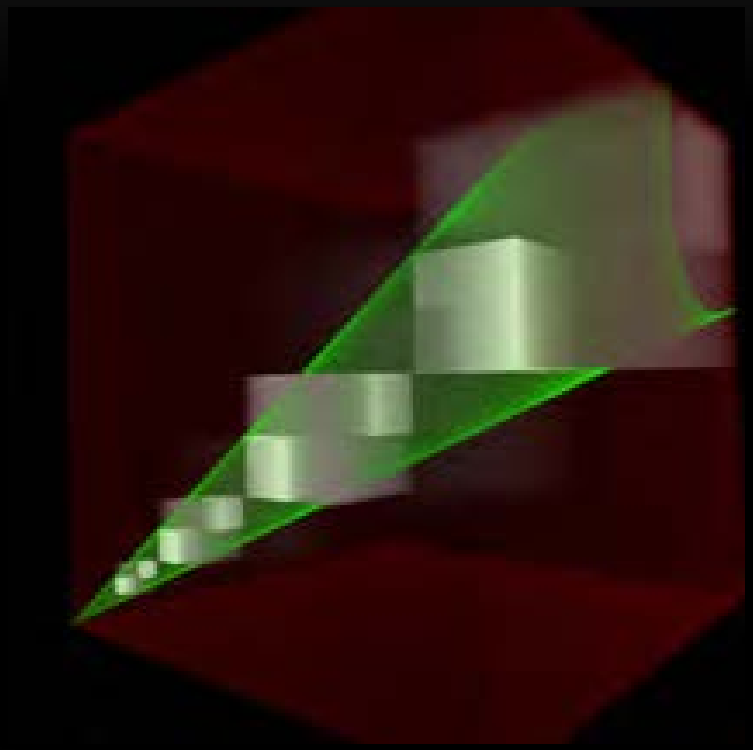
- Pre-filtered Data
- Cone Angle / FOV
- Sample from mips

- Anti-Aliased
- Levels of Detail
- Adaptive Step Size
- Empty Space Skipping



Voxel Cone Tracing





Cyril Crassin, GigaVoxels: A Voxel-Based Rendering Pipeline For Efficient Exploration Of Large And Detailed Scenes

OUR APPROACH

How do we do it?

PIPELINE

Diffuse Shading

- Traditional rasterization-based technique

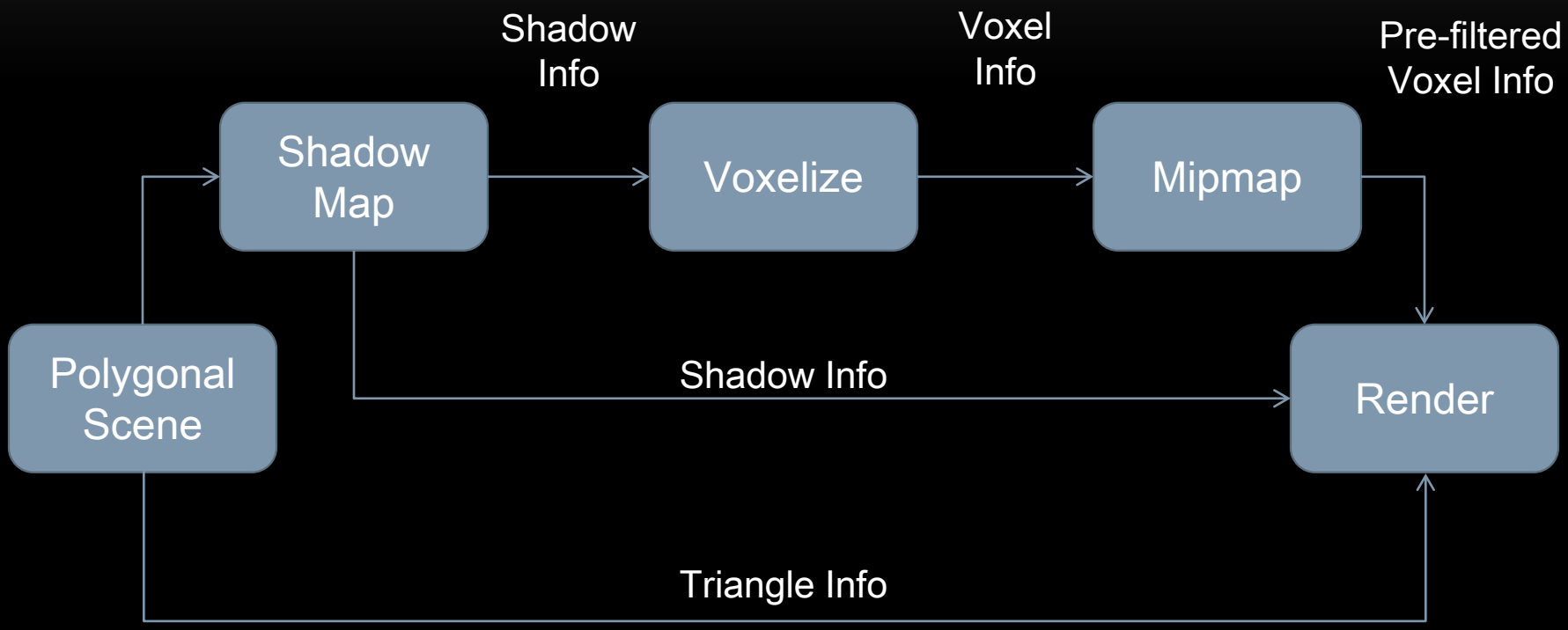
Beyond Diffuse Shading

- Voxel cone-tracing

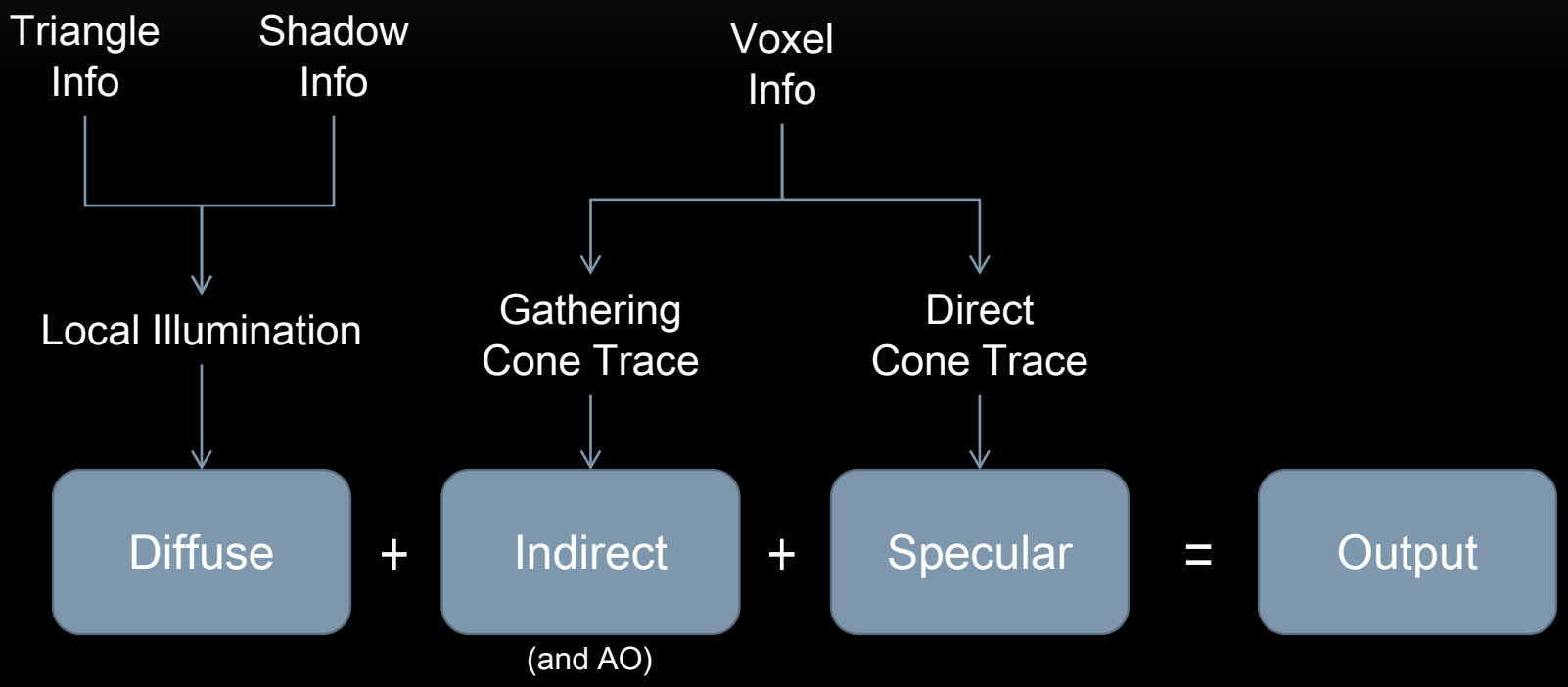
2 Separate Scene Representations

- Polygonal Mesh
- Voxel 3D Texture

PIPELINE


Shadow Info
Voxel Info
Pre-filtered Voxel Info
Shadow Map
Voxelize
Mipmap
Polygonal Scene
Shadow Info
Render
Triangle Info

PIPELINE – RENDERING



By the way…

Everything is on the GPU

Variance Shadow Map

- Soft shadows
- Apply a Gaussian blur to the shadow map
- Use probability function to determine shadow strength

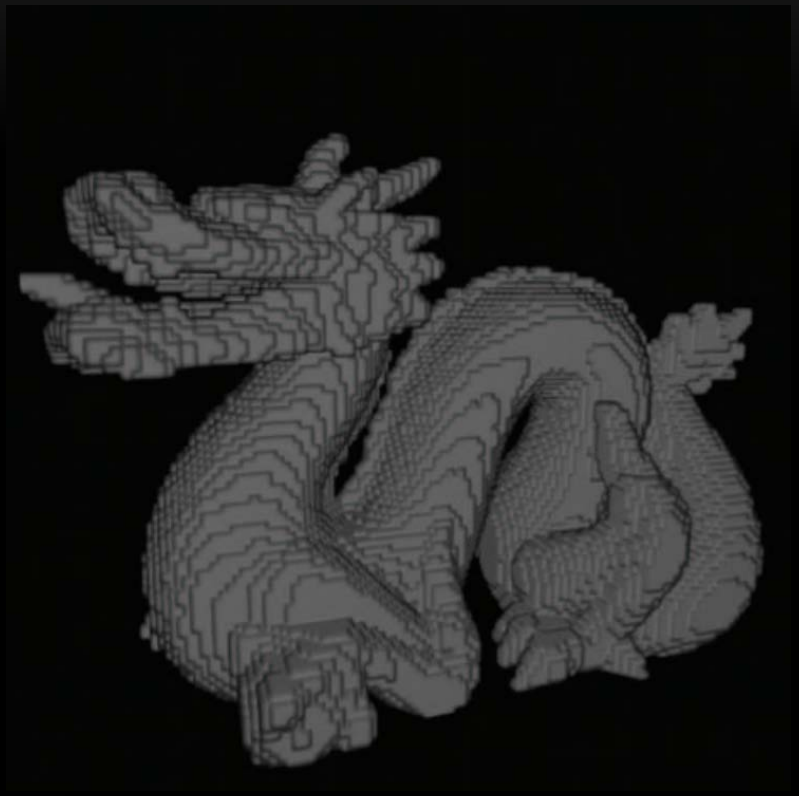
http://pixelstoomany.wordpress.com/category/shadows/variance-shadow-maps/

Voxelize Triangles

- Render from orthographic camera
- 3 passes for each axis
- Viewport = voxel grid dimensions

- Output: radiance color and direction

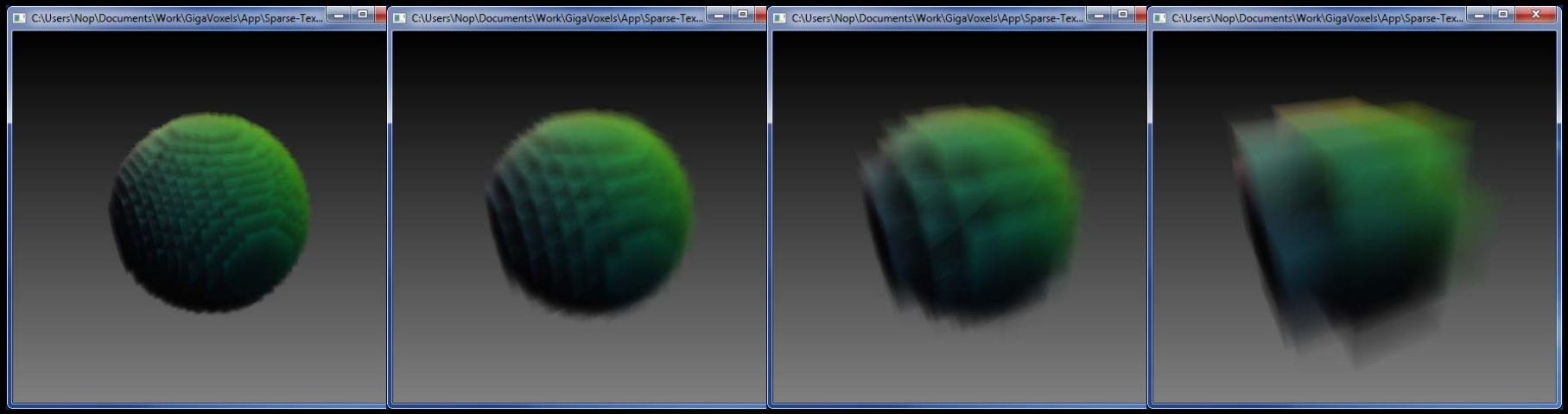
- On fragment shader,
  Image load/store (OpenGL 4.2)



Cyril Crassin. OpenGL Insights Sparse Voxelization.

Mipmaps

- Alpha weighted average
- On fragment shader. Image load/store (OpenGL 4.2)



Mipmaps

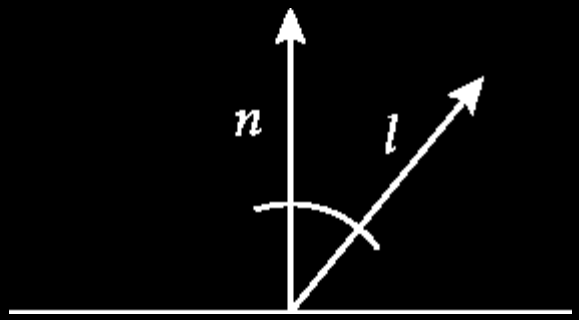
Cyril Crassin, GigaVoxels: A Voxel-Based Rendering Pipeline For Efficient Exploration Of Large And Detailed Scenes

LIGHTING

Making things look pretty

Diffuse Lighting

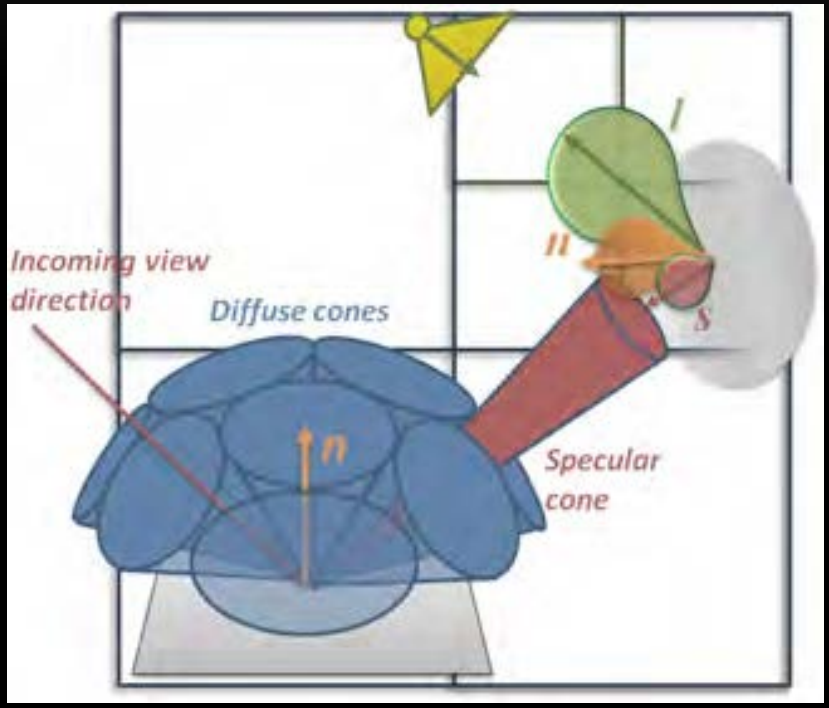
$$I = k_a + I_L k_d (\hat{l} \cdot \hat{n}) + I_L k_s (\hat{h} \cdot \hat{n})^n$$



Indirect Lighting

- Approximate a hemisphere by tracing cones outwards
- Treat each gather point as point light.

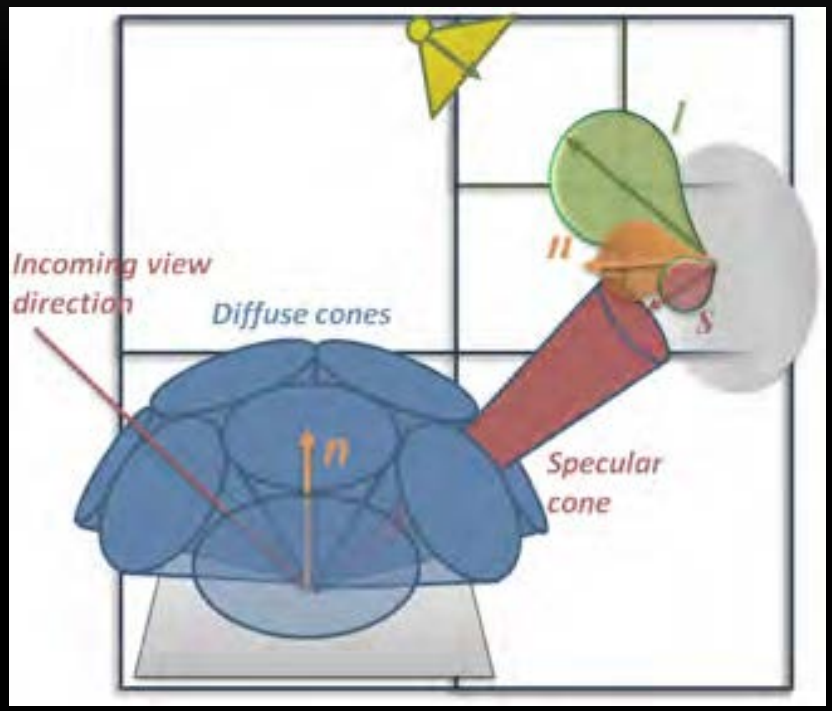
- Ambient Occlusion can be done in a similar way



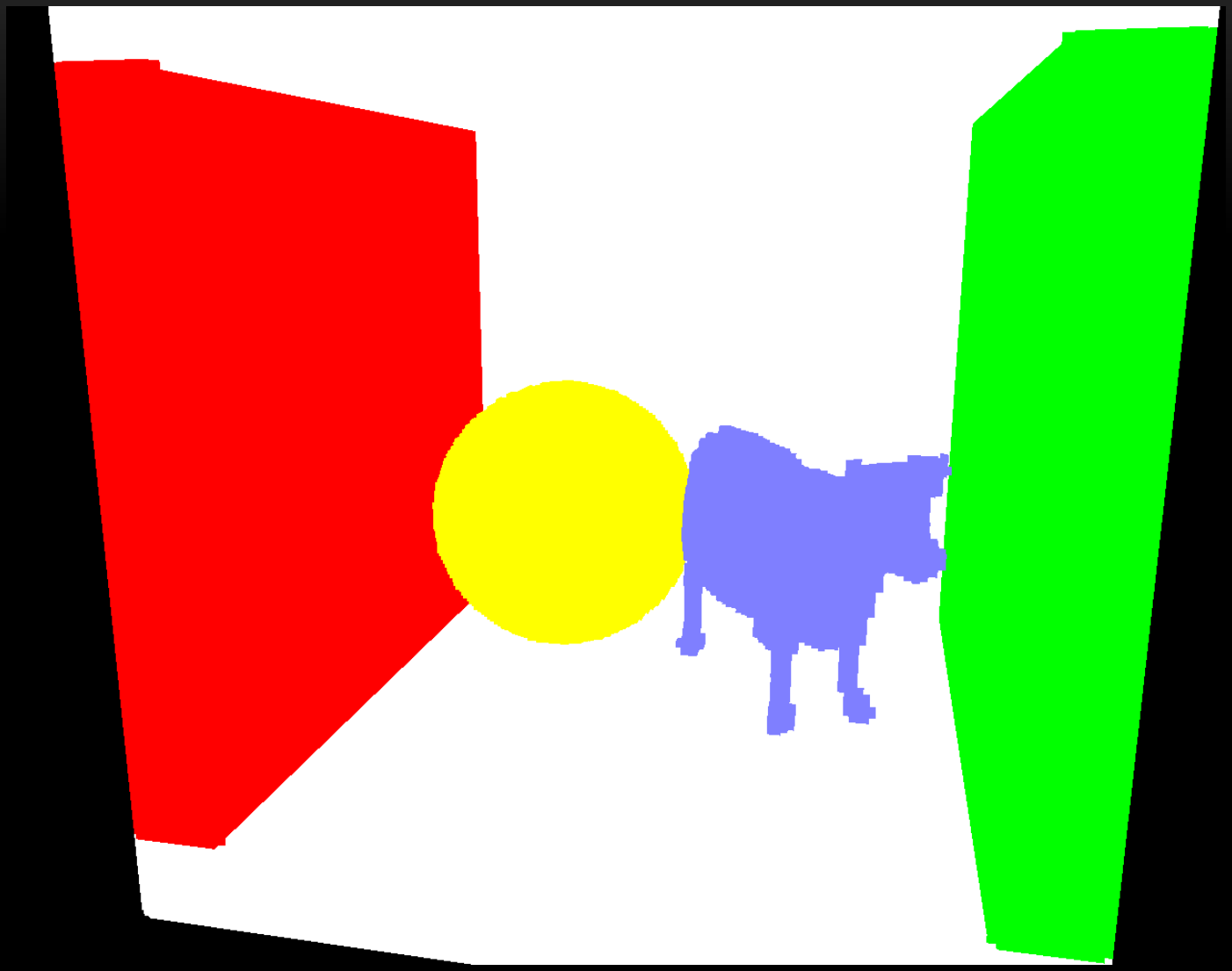
Cyril Crassin, GigaVoxels: A Voxel-Based Rendering Pipeline For Efficient Exploration Of Large And Detailed Scenes

Specular Reflection

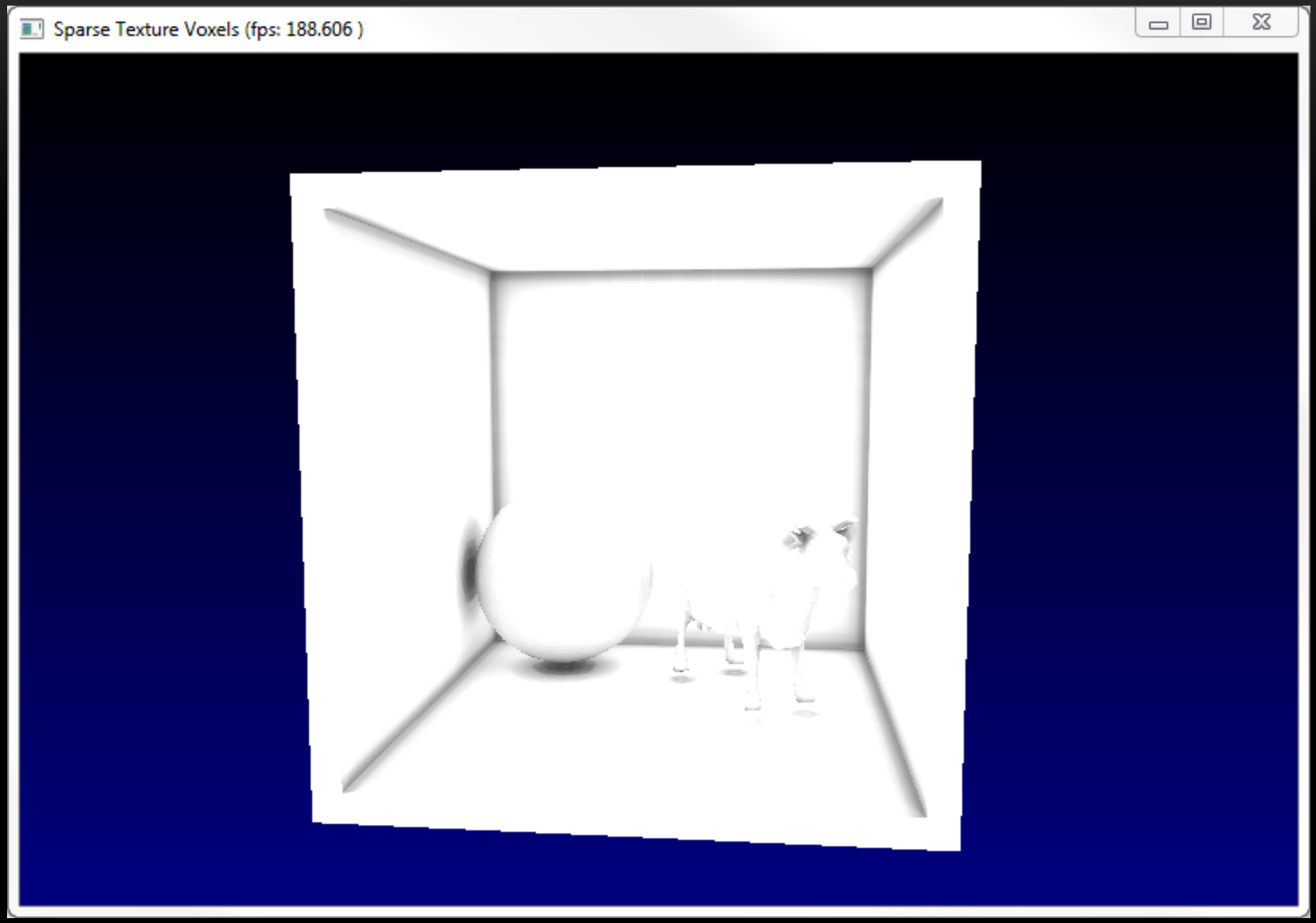
- A sharper cone in the direction of the reflected view direction.
- Direct visualization of voxel data



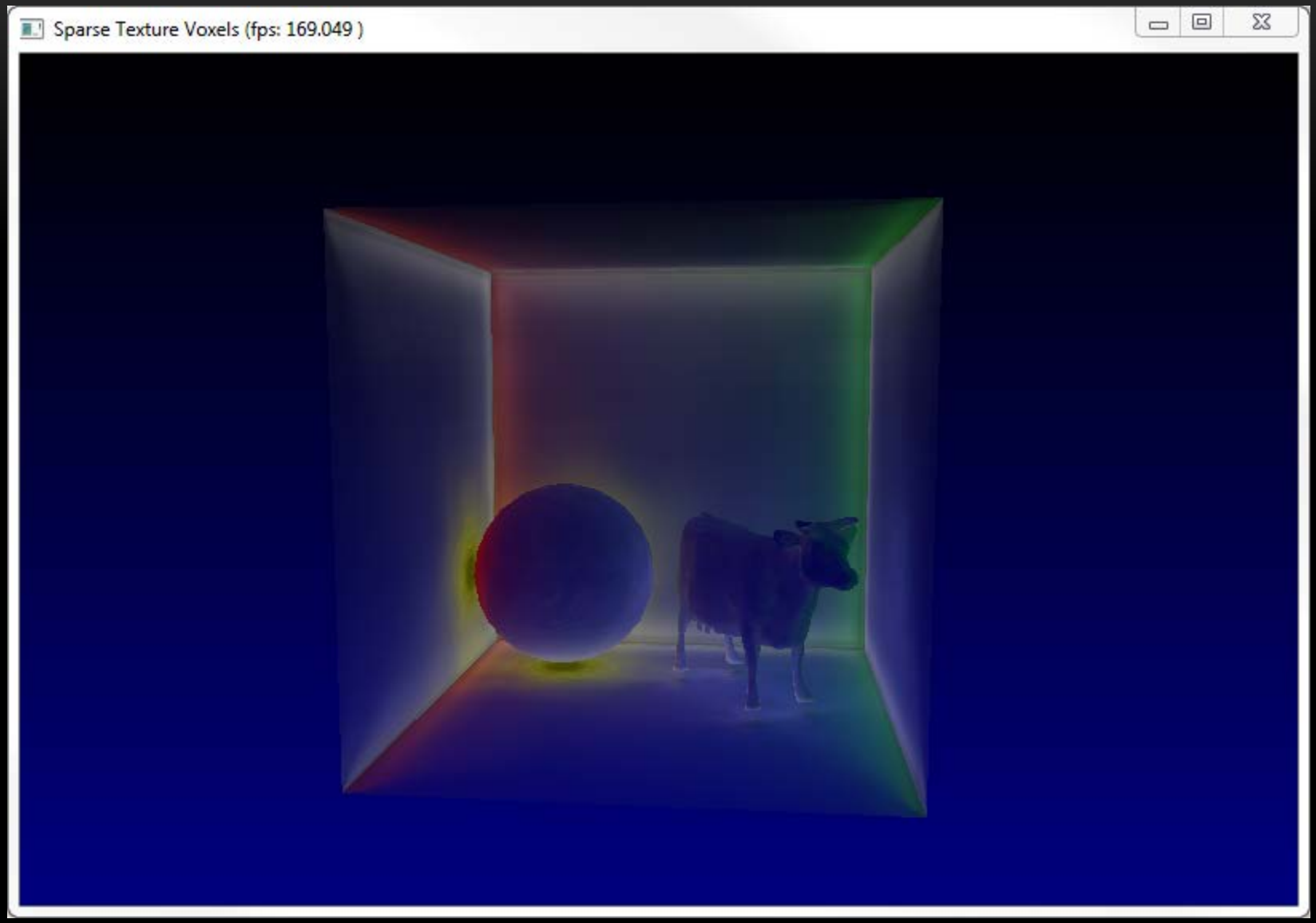
Cyril Crassin, GigaVoxels: A Voxel-Based Rendering Pipeline For Efficient Exploration Of Large And Detailed Scenes



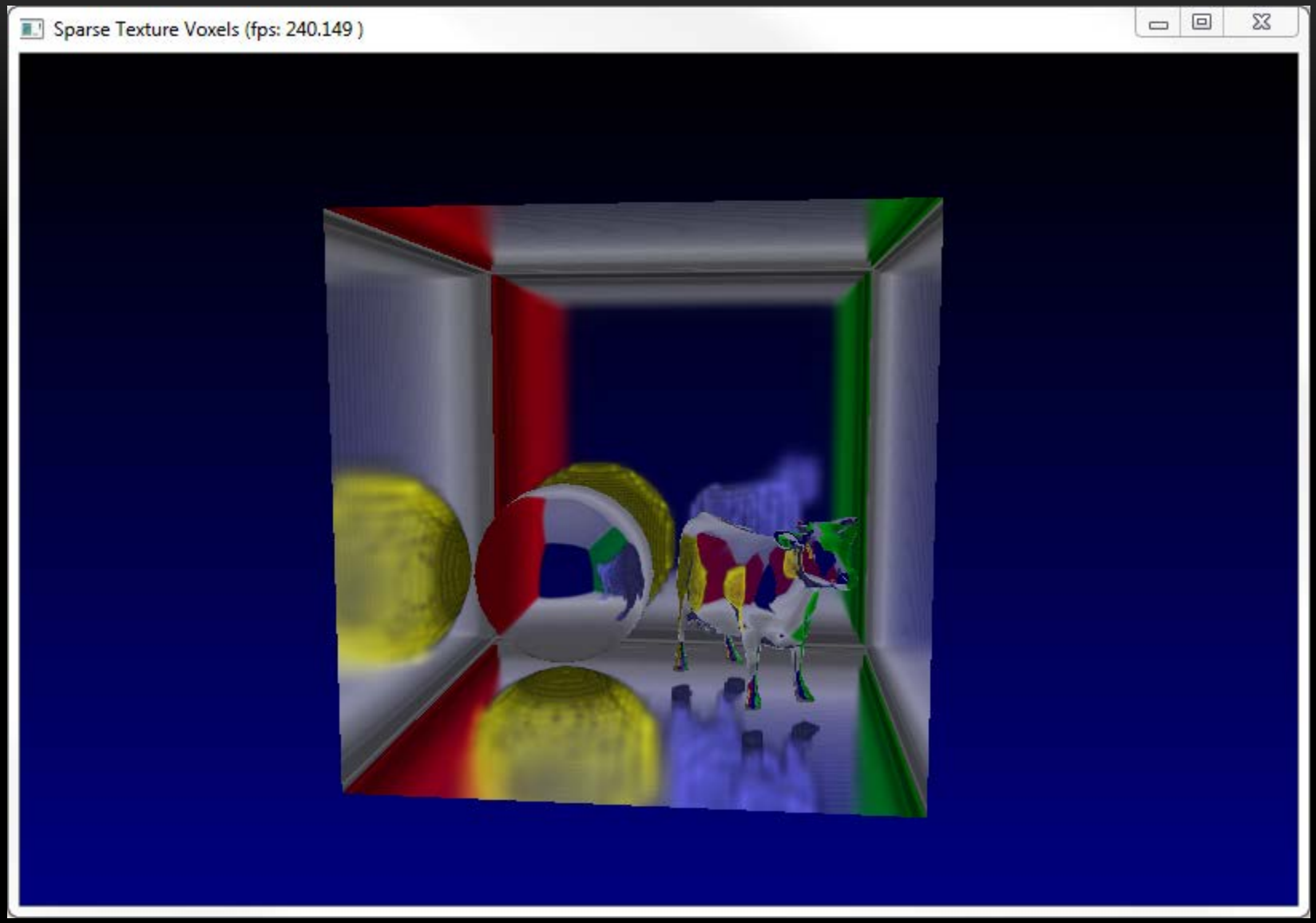
Diffused Color (no lighting)



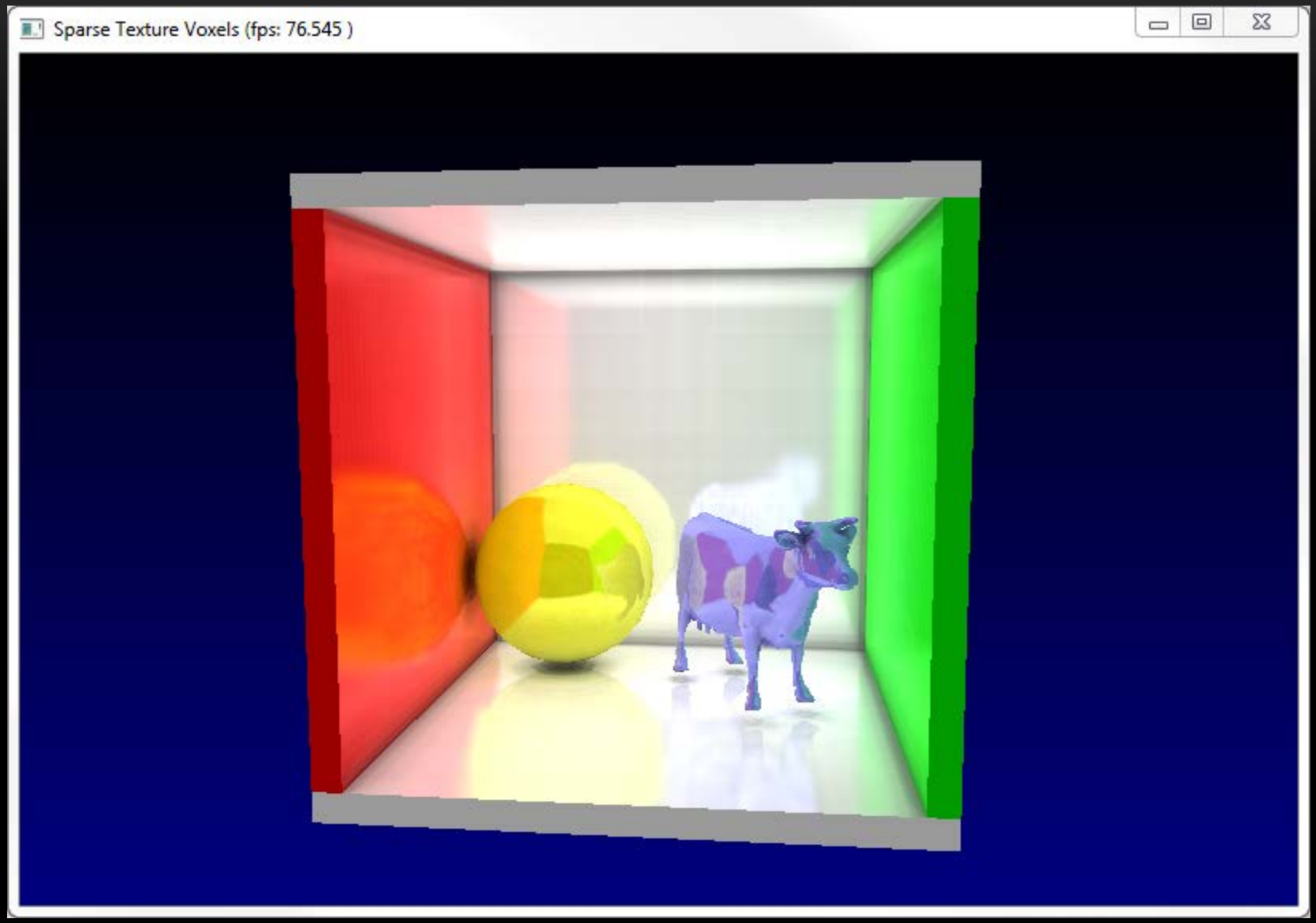
Ambient Occlusion



Indirect Diffuse



Specular Reflection



Composited Output

DEMO

The real deal

FUTURE WORK

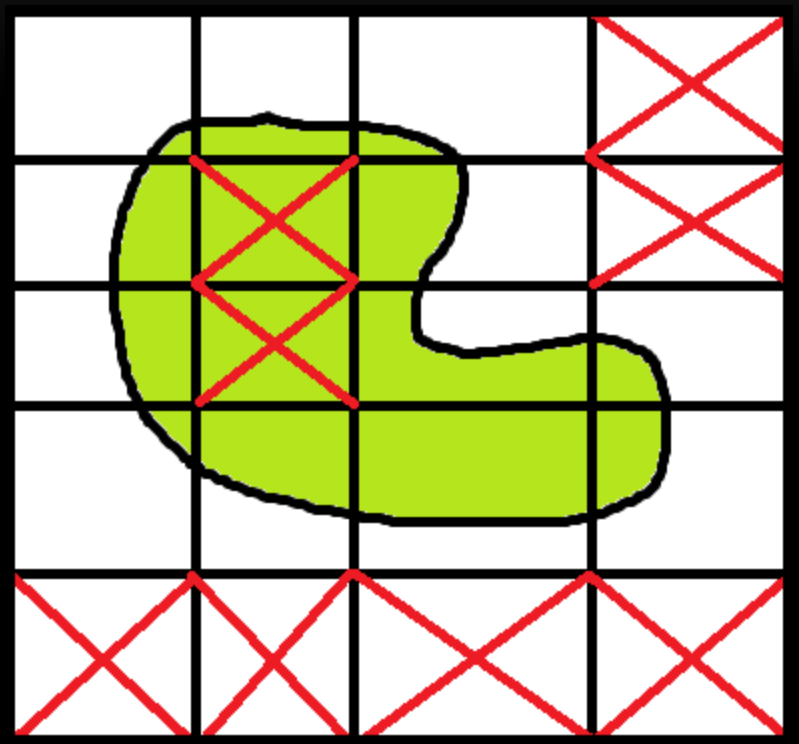
We still got lots to do.

Sparse Texture

- Different from Cyril's method of using Oct-tree

- Large OpenGL 3D Texture
- Choose regions to be in memory

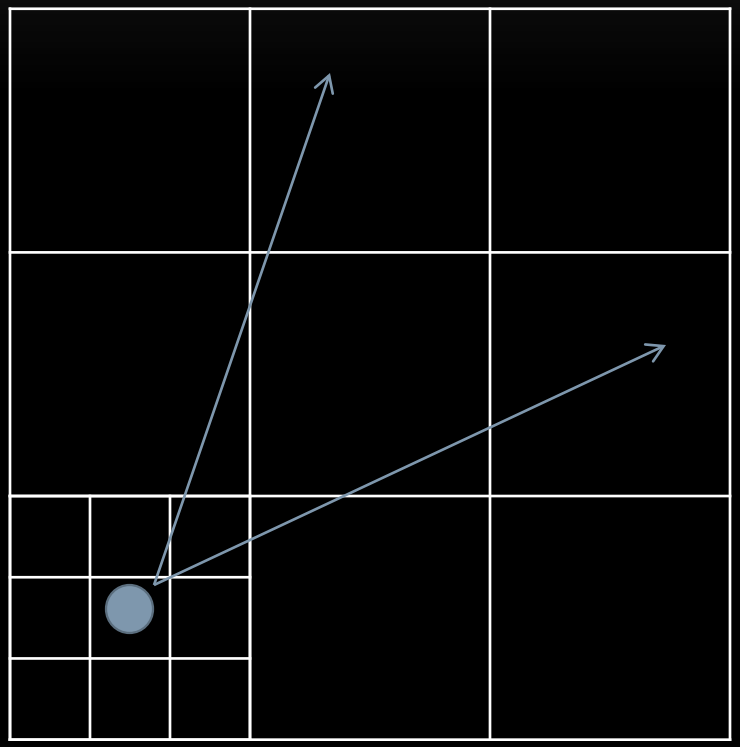
- Spatially coherent
- Hardware interpolation (trilinear)
- Mipmap interpolation (quadrilinear)



View-Dependent Voxelization

- Voxelize scene only around the eye, on demand
- The renderer can handle larger scenes

- Inspired by cascaded shadow maps



Sources

GigaVoxels: A Voxel-Based Rendering Pipeline For Efficient Exploration Of Large And Detailed Scenes

http://www.icare3d.org/research-cat/publications/gigavoxels-a-voxel-based-rendering-pipeline-for-efficient-exploration-of-large-and-detailed-scenes.html

Octree-Based Sparse Voxelization Using The GPU Hardware Rasterizer

http://www.icare3d.org/research-cat/publications/octree-based-sparse-voxelization-using-the-gpu-hardware-rasterizer.html

The Technology Behind the Elemental Demo. SIGGRAPH 2012.

http://www.unrealengine.com/files/misc/The_Technology_Behind_the_Elemental_Demo_16x9_(2).pdf