

GigaVoxels: Voxels Come Into Play

Cyril Crassin

▶ To cite this version:

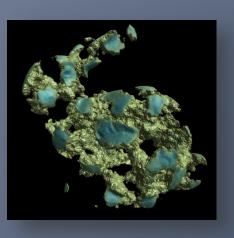
Cyril Crassin. Giga Voxels: Voxels Come Into Play. Crytek Conference, Nov
 2009, Frankfurt, Germany. inria-00526646

HAL Id: inria-00526646 https://inria.hal.science/inria-00526646

Submitted on 15 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

GIGAVOXELS: **VOXELS COME INTO PLAY**









Cyril Crassin, Fabrice Neyret, INRIA Rhône-Alpes & Grenoble Univ.

Sylvain Lefebvre,

Elmar Eisemann,

Miguel Sainz

INRIA Sophia-Antipolis Saarland Univ./MPI

NVIDIA Corporation





A (very) brief history of voxels

Rings a bell?





Comanche (Novalogic)

Voxel Engines in Special effects

- Natural representation
 - Fluid, smoke, scans, ...
- Volumetric phenomena
 - Semi-transparency
- Unified rendering representation
 - Particles, meshes, fluids...







Voxels in video games?

- Renewed interest
 - ID Software
 - John Carmack, Jon Olick (Siggraph 08)
 - Sparse Voxel Octree ray-casting
 - Crytek
 - Cevat Yerli
 - •
- Two goals :
 - Content generation
 - Rendering

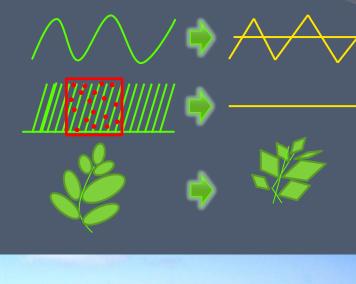




- Exploding number of triangles
 - Costly to transform & rasterize
 - Inefficient raster of small triangles on current generation GPUs
- Geometric LOD ill-defined
 - Eg. Progressive Meshes
 - Lot of manual intervention for the artist



- Filtering is an issue
 - Needs massive multisampling
 - Multi-sampling is expensive





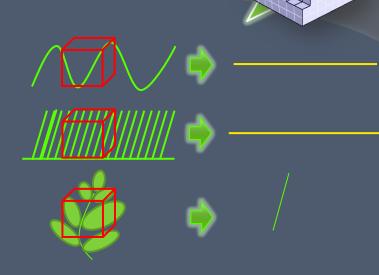
The Mummy 3, Digital Domain/Rhythm&Hues



- Unified Geometry + Texture representation
 - Avg space occupancy/density information
 - Avg color information

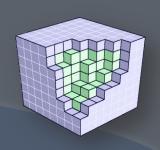


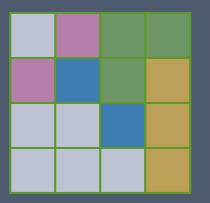
The Mummy 3, Digital Domain/Rhythm&Hues





- Filtering is well defined
 - LOD = Mip-Mapping
 - Similarly to 2D textures
- Unique multi-scale representation
 - No additional authoring
- Structured representation
 - Convenient to traverse & edit
 - Efficient to render
 - -> Ray-casting







How to exploit them?

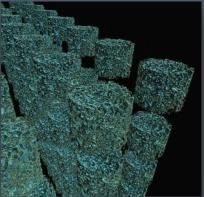
Main problems:

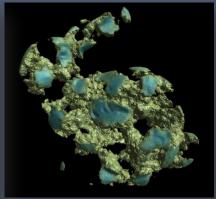
- How to render voxels quickly on the GPU?
 - How to exploit these properties ?
- Memory is a key issue!
 - E.g. 4096 ^ 3 x RGBA8 = 256 GB!!!
 - Transfer CPU ⇔ GPU expensive

GigaVoxels

- Goal: Real-time exploration of very large voxel scenes
- Full GPU rendering pipeline
 - Ray-tracing based approach
 - Fully scalable: Infinite resolution
- Publications:
 - I3D2009 paper [CNLE09]
 - Siggraph 2009 Talk
 - GPU Pro (ShaderX 8) Book Chapter







GigaVoxels: Ray-Guided Streaming for Efficient and Detailed Voxel Renderic

Cyril Crassin Fabrice Neyret LJK / INRIA / Grenoble Universities / CNRS

t Sylvain Lefeby

Elmar Eisemann MPI Informatik / Soarland Univ









Figure 1: Designs show volunte data that consist of billions of wards rendered with our dynamic sparse extree approach. Our algorithm achieves real-time to introceive rates volunts exceeding GPU memory capacity by far, tasks to efficient invasting and ray-caving unlations. Businelli, the volunts is only used at the resolution that in model to product the final internal relieful the matrix in moment and assent, and we mediatrical models in information of unlated.

bstract

'e propose a new approach to efficiently render large volumet ata sets. The system achieves interactive to real-time renderi erformance for several billion voxels.

solution is based on an adaptive data representation dependon the current view and occlusion information, coupled to an item two-casting rendering algorithm. One key element of our and is to guide data production and streaming directly based on mation extracted during readered during readered.

an data servicure exploits the fact that in CG scenes, details are no concentrated on the interface between the space and clusters density and shows that volunearies models might become a valuiiest alternative as a readering printive for read-time application, this spirit, we allow a quality-performance trade-off and exploit in particular and the properties of the properties of the properties of properties of the properties of the properties of the properties of the particular properties of the properties of the properties of the quality filtering. To further enrich the data set, we create addimated details through a variety of procedural methods.

e demenstrate our approach in several sconarios, like the exploion of a 3D scan (8192) resolution), of hypertextured meshes \$384 virtual resolution), or of a fractal (theoretically infinite reution). All examples are rendered on current generation handware 20-90 fips and respect the limited GPU memory budget.

his is the author's version of the paper. The ultimate versias been published in the I3D 2009 conference proceedings.

lume data has often been use

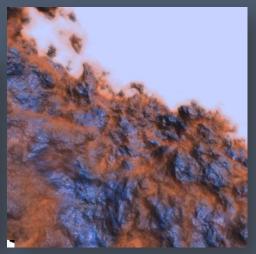
to sterage, the rendering of such data is also extremely costly, exfor previsualization.

The significant advantage of voxels is the richness of this represtation and the very regular structure which makes it easy to manig

This is one of the russons voxel data is often used to represposado-surfaces, which is an interface that resembles a surfat a certain distance, but appears complex (non-buildfield, in connected, or non-orque)s at close view. An example is the large of a row fut can be well approximated with volume data [RAMIO94], but this observation look for complex and in general. This volumetric redoring is also a graceful way

In this paper, we show that the current hardware generation is read to editive high-quality transiers volume readering at interactive to real-time rates. Hendits used as filtering, occlusion culling, and procedural data contains, as well as selved-of-deal interhanisms are minigrated in an efficient OPU votel engine. This crabbes us to obtain some of the visual quality but at was personally reserved for movie productions and enables the technique to be used to previsualize security different.

you've are born mapping black occurs indicag usualization freathering in massive volumes possible: overcome the memory limitations (as propose related update schemes) and the usually contry rendering black and the control of the schemes of the sc



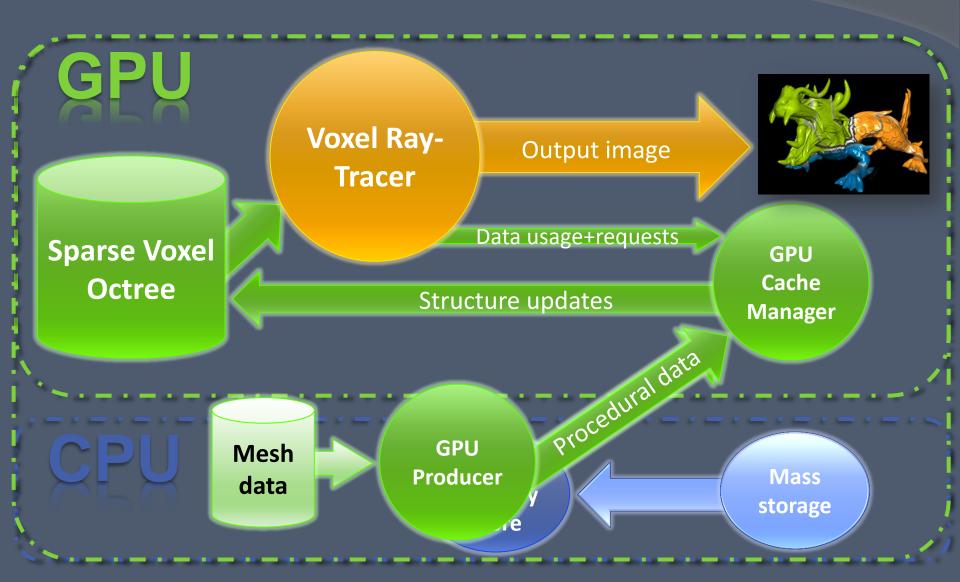
Key ideas

- Rendering only dependant on what is visible
 - Ray-tracing approach
- Load only needed data, at the needed resolution
 - Occlusion + LOD
 - Ray-guided streaming
- Reuse loaded data as much as possible
 - GPU cache mechanism





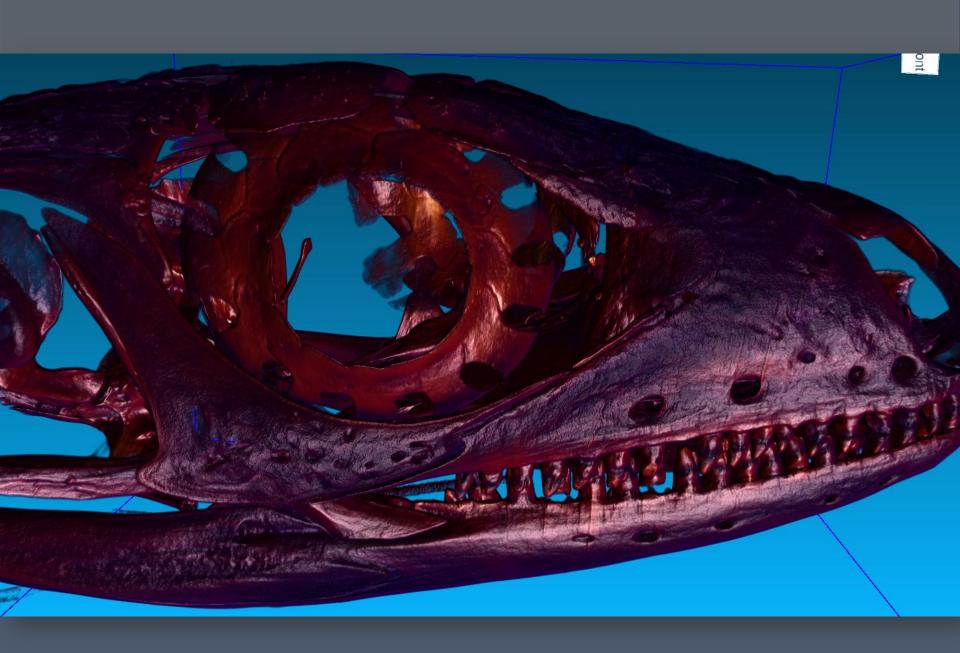
GigaVoxels CUDA pipeline





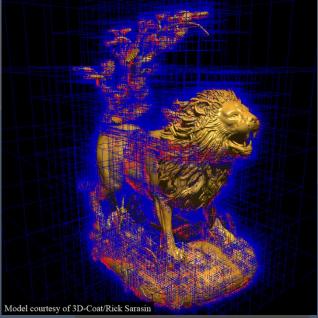






Voxel sculpting

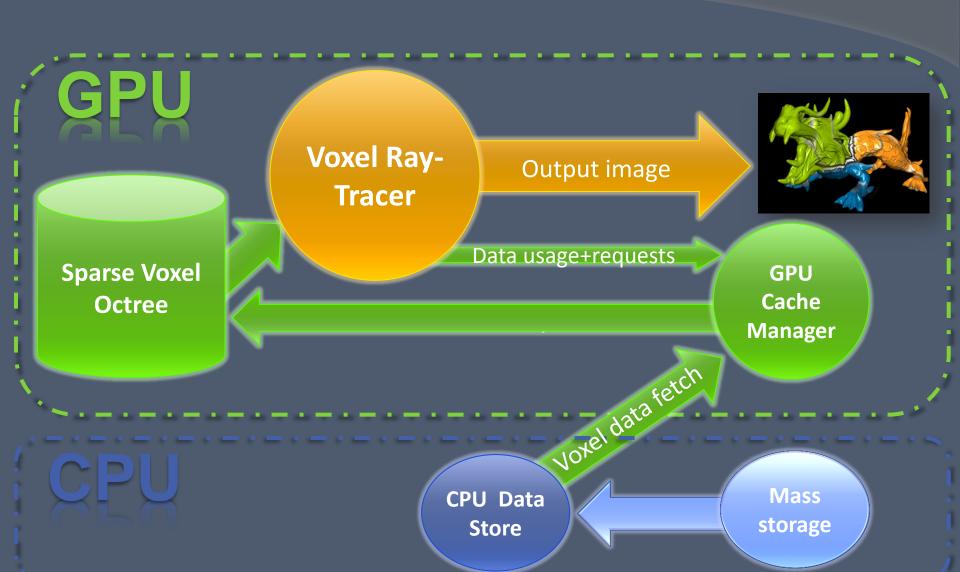
- Direct voxel scultping
 - 3D-Coat
 - Like ZBrush
- Generate a lot of details



5-20 FPS



Data Structure



Sparse Voxel MipMap Pyramid

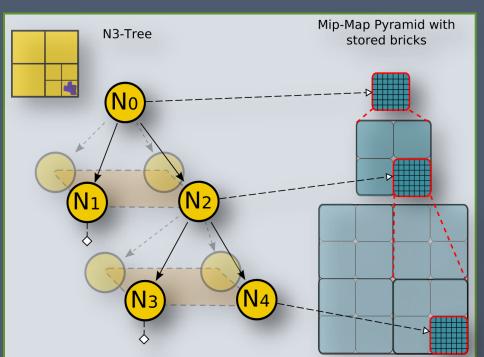
Data structure

Generalized Octree

Empty space compaction

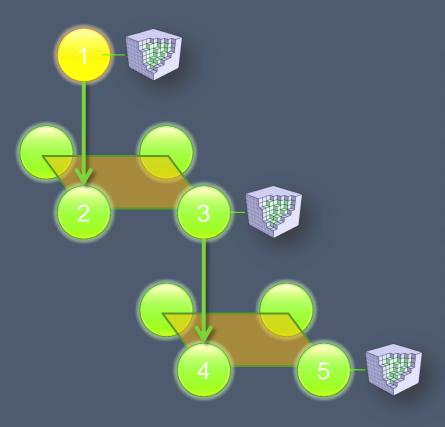
Bricks of voxels

- Linked by octree nodes
- Store opacity, color, normal,...

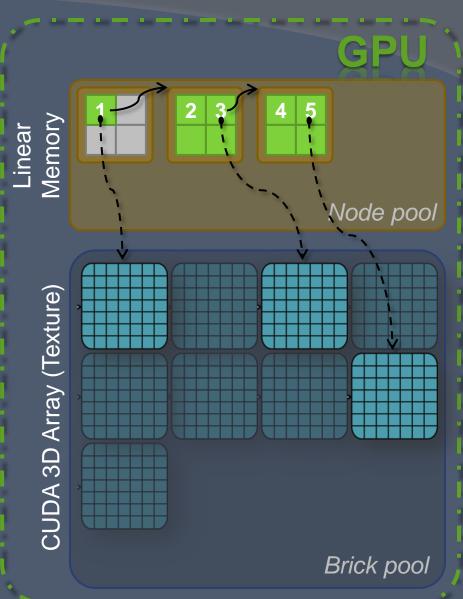




Octree of Voxel Bricks



- One child pointer
 - Compact structure
 - Cache efficient



Rendering

GPU

Sparse Voxel
Octree

Voxel Ray-Tracer

Output image

Data usage+requests

Structure updates

GPU Cache Manager

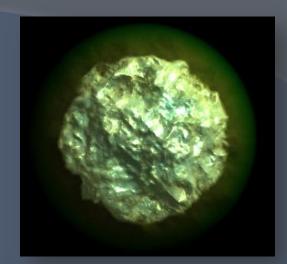
Joxel data fetch

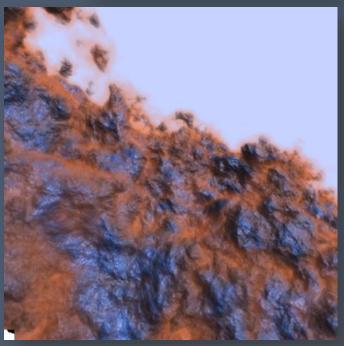
CPU

CPU Data Store demand
Data
Producer

Hierarchical Volume Ray-Casting

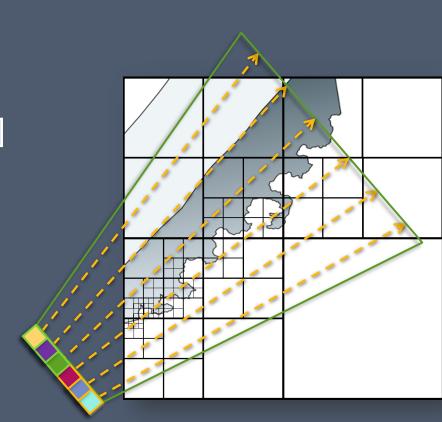
- Render semi-transparent materials
 - Participating medias
- Emission/Absorption model for each ray
 - Accumulate Color intensity + Alpha
 - Front-to-back
 - Stop when opaque



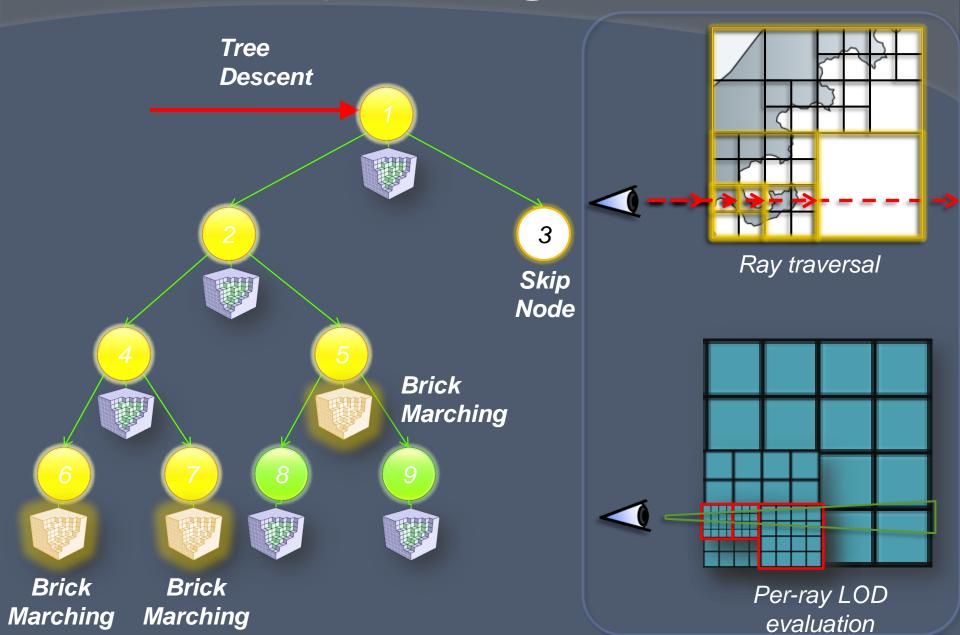


Hierarchical Volume Ray-Casting

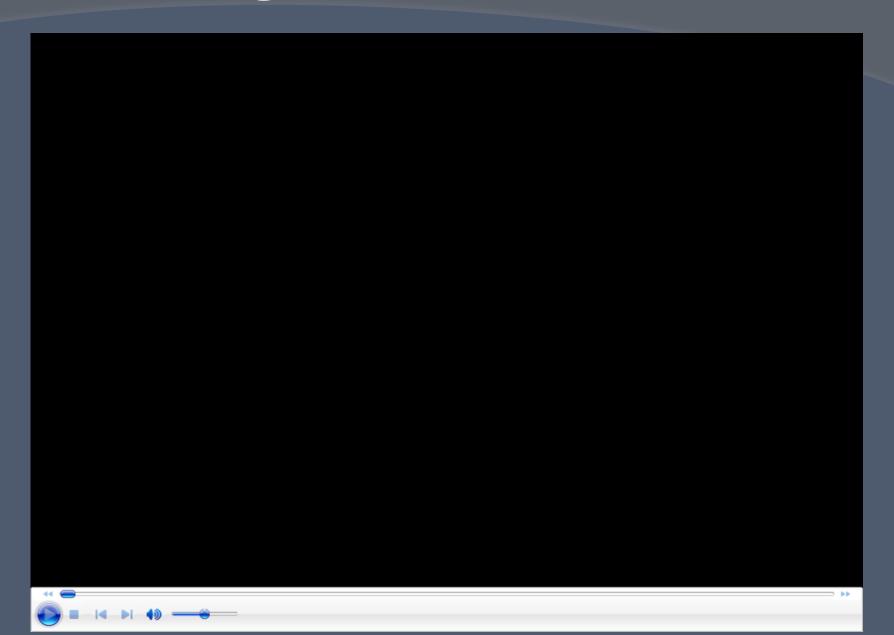
- Volume ray-casting
 [Sch05, CB04, LHN05a, Olick08, GMAG08, CNLE09]
- One big CUDA kernel
 - One thread per ray
- Octree traversal
 - KD-restart algorithm [FS05]
 - Ray-driven LOD
- Bricks marching
 - Regular sampling into the 3D texture



Volume Ray-Casting



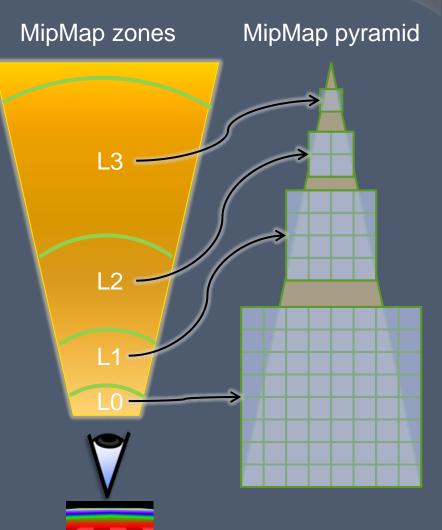
Rendering costs



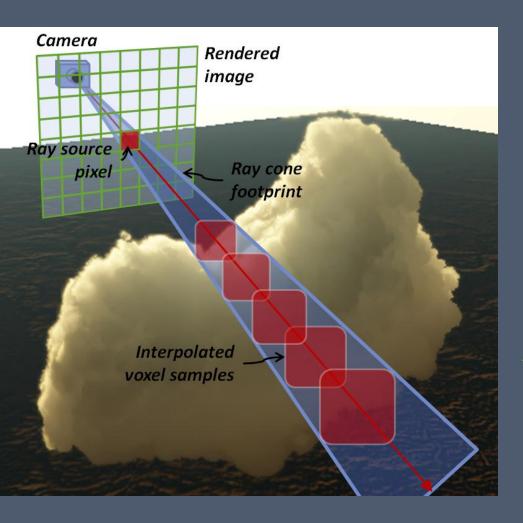
Volume MipMapping mechanism

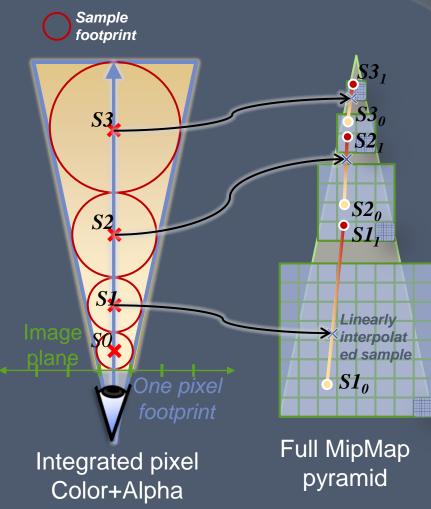
Problem: LOD uses discrete downsampled levels

- Popping + Aliasing
- Same as bilinear only for 2D textures
- Geometry is texture ©
 - Uses pre-integrated LOD!
 - No need of multi-sampling (eg. MSAA)



Cone tracing



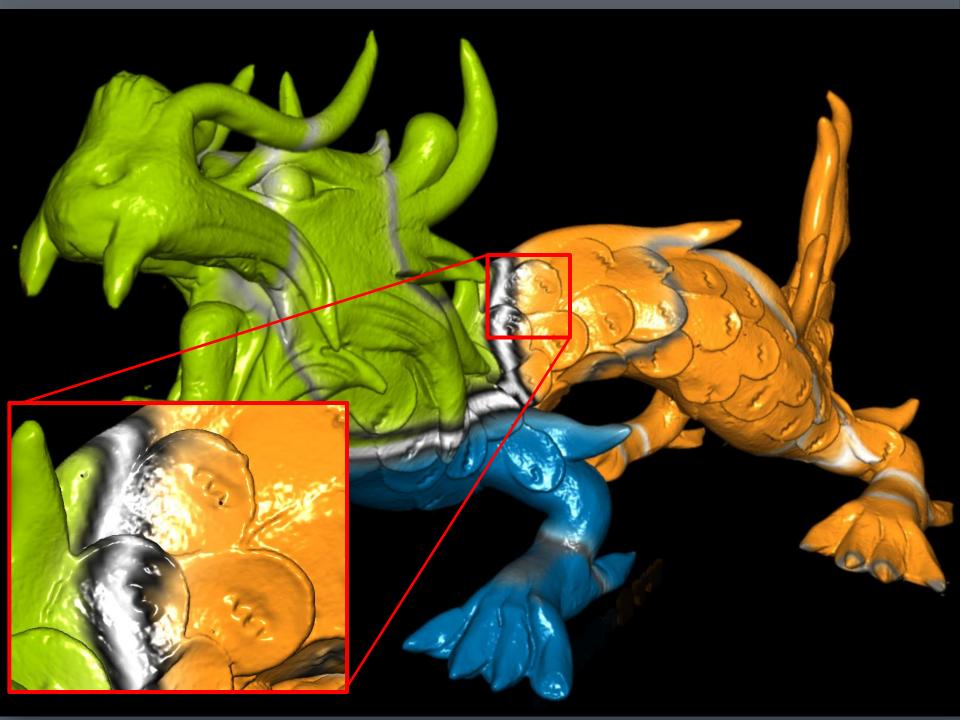


Shading computation

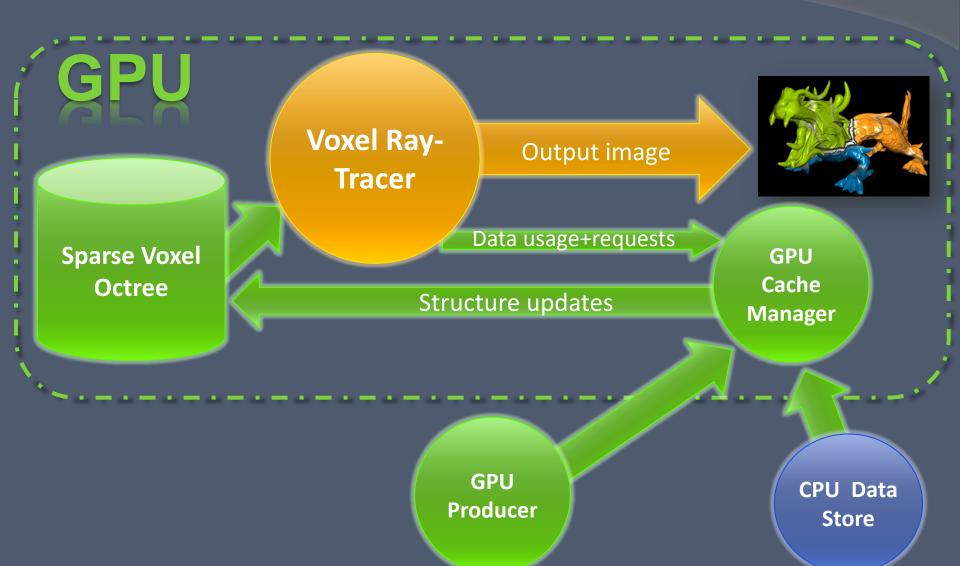
- Standard Blinn-Phong illumination
 - Per sample
- Normal information
 - On-the-fly gradient with finite differences
 - Stored normal information

Deferred for opaque objects



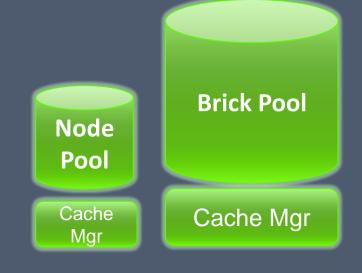


Data Management



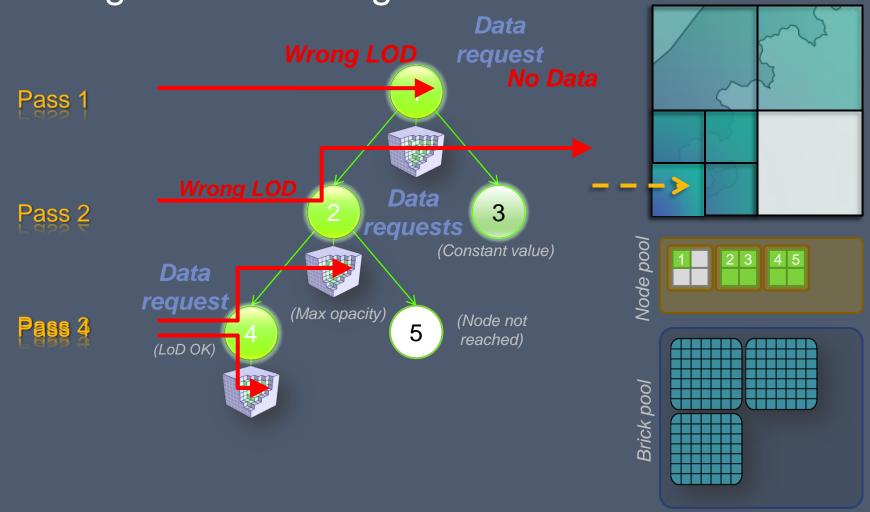
GPU Caches

- Data management made through a cache mechanism
 - Used for both the node pool and brick pool
 - Allows full scalability
- Rely on the octree to address elements
 - The node pool is addressing itself!
 - No page table
- Data requests generated by the ray-tracing
 - Node subdivision
 - Brick loading



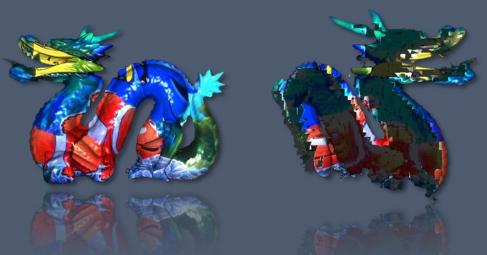
Incremental octree update

Progressive loading



Ray-based visibility & requests

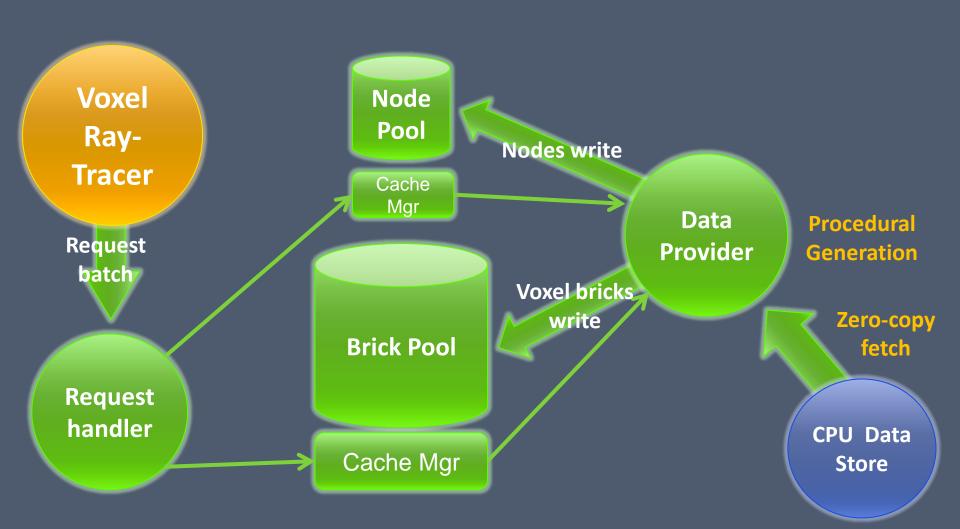
Minimum amount of data is loaded



- Progressive refinement
 - Always ensure interactivity
- Fully compatible with secondary rays and exotic rays paths
 - Reflections, refractions, shadows, curved rays, ...

Cache requests handling

Entirely handled on the GPU



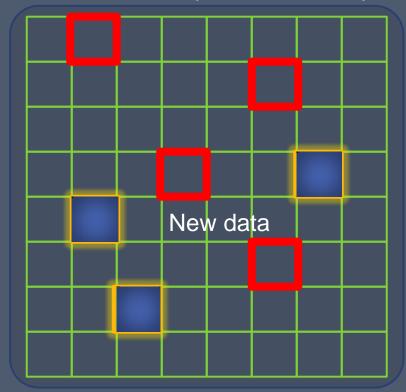
Cache strategy

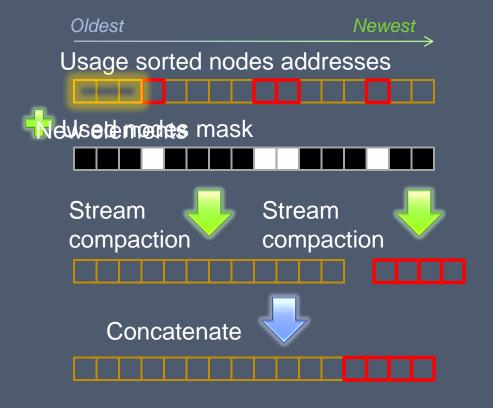
- Least Recently Used (LRU) strategy
 - Older elements replaced first
- Sorted usage list maintained for each cache on the GPU.
- Usage info provided by the ray-tracer
- Maintained as a data-parallel process
- Used when new elements have to be inserted

SVMP caches

- LRU (Least Recently Used)
 - Track elements usage
 - Maintain list with least used in front

Cache Elements (Node Tile/Brick)

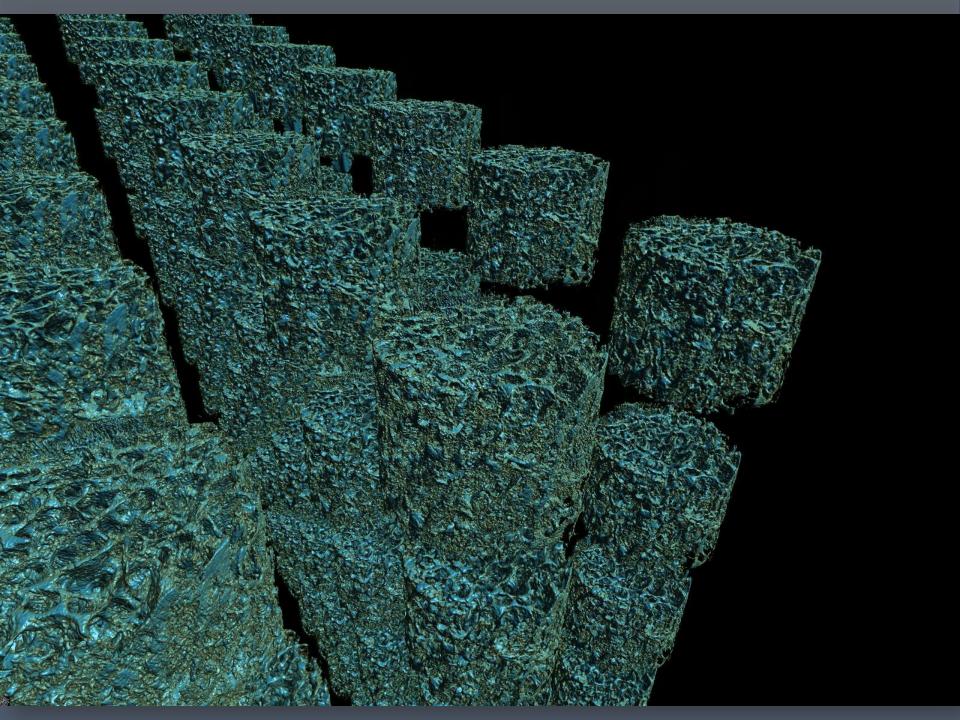




Octree/Bricks Pool

Global cache characteristics

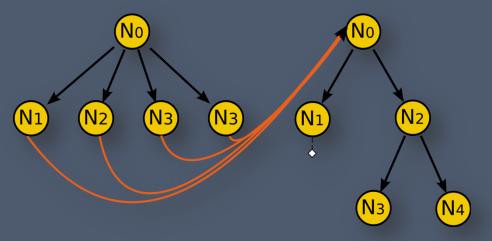
- Driven by ray-tracing
- Fully managed on the GPU
 - Zero CPU intervention apart kernel launches.
 - Leads to fully on-chip structure management and building
- More efficient when large amount of updates



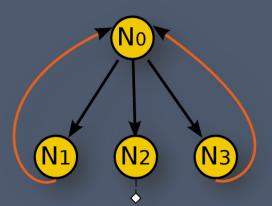
APPLICATIONS

Voxel data synthesis

Instantiation

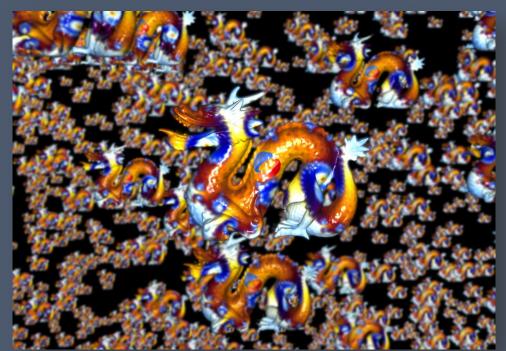


- Recursivity
 - Infinite details



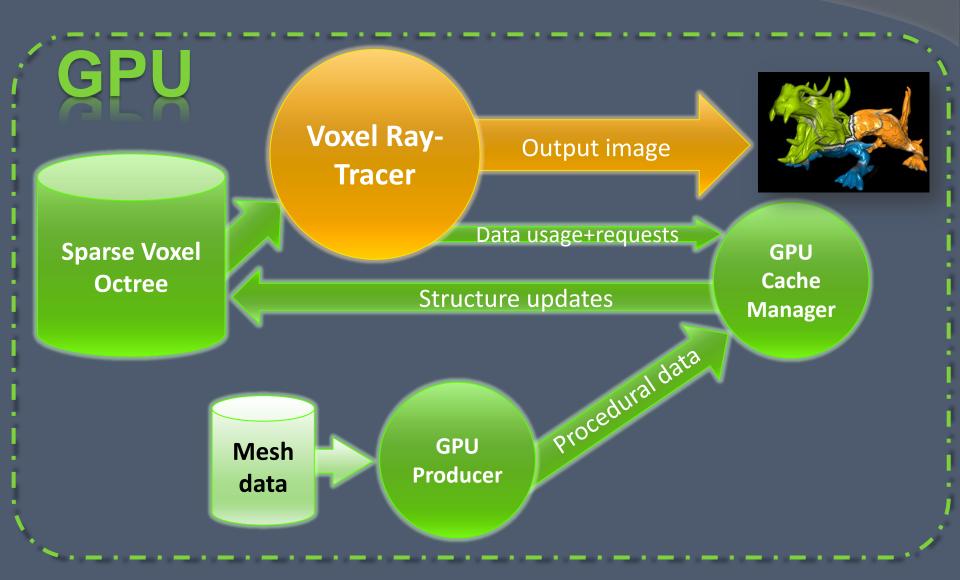
Free voxel objects instancing

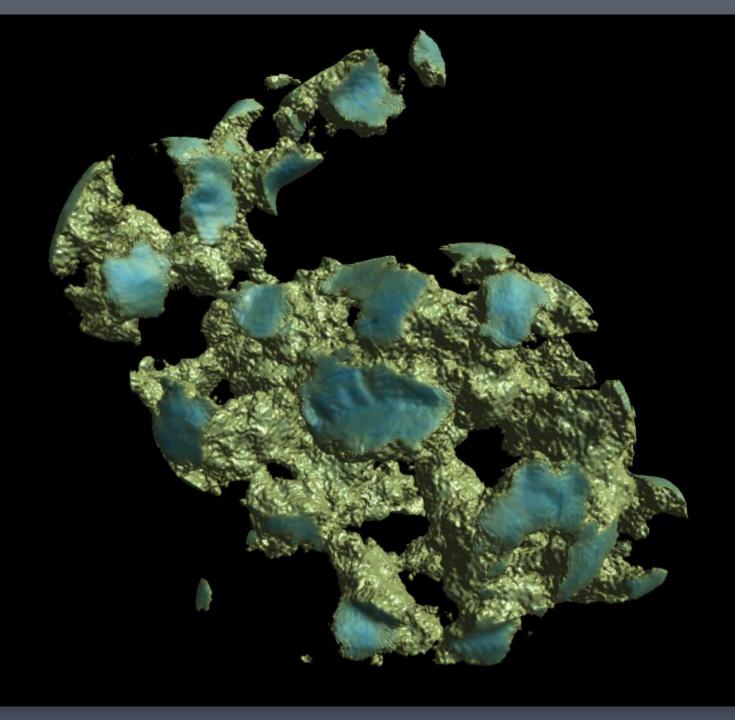
- BVH structure ray-casting
 - Cooperative ray packet traversal [GPSS07]
 - Shared stack
- WA-Buffer
 - Deferred compositing





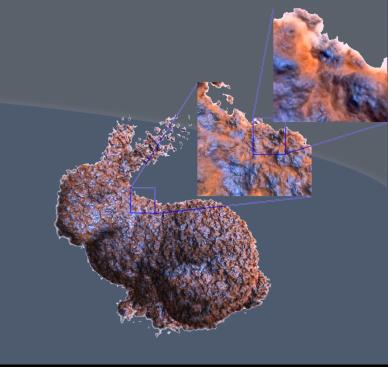
Voxels generation



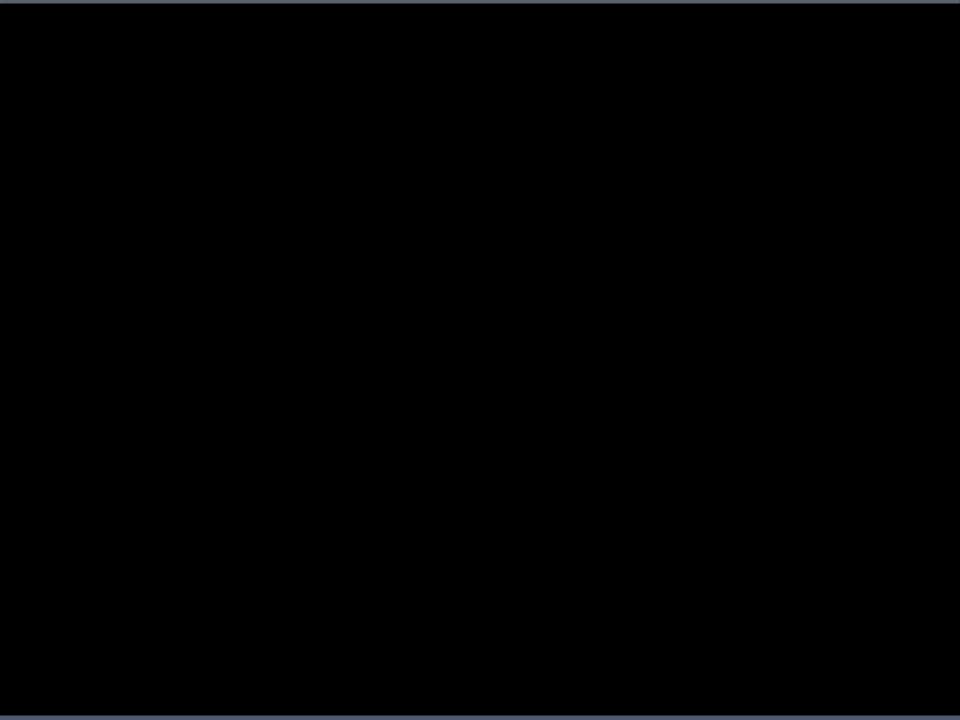


Procedural noise

- On-the-fly mesh voxelization
 - Distance field
- Procedural noise







Cool Blurry Effects

- Going further with 3D MipMapping
 - Full pre-integrated versions of objects

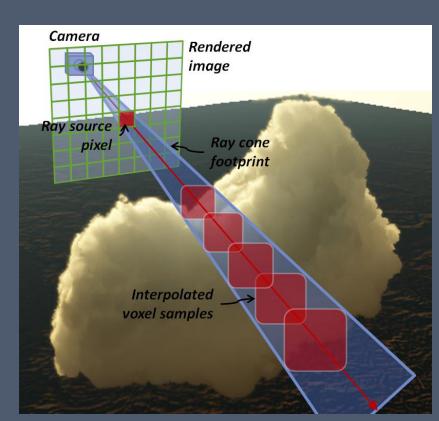
• Idea: Implements blurry effects very

efficiently

Without multi-sampling

Tuning the mipmap level

- Soft shadows
- Depth of field
- Glossy reflections...

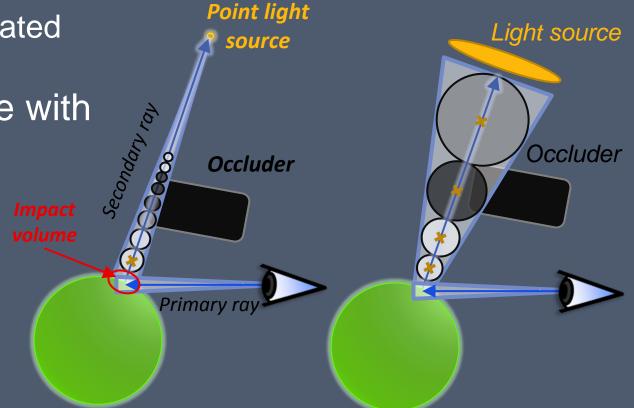


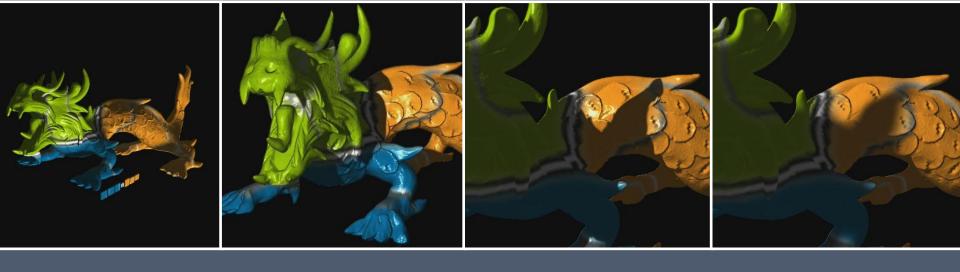
Soft shadows

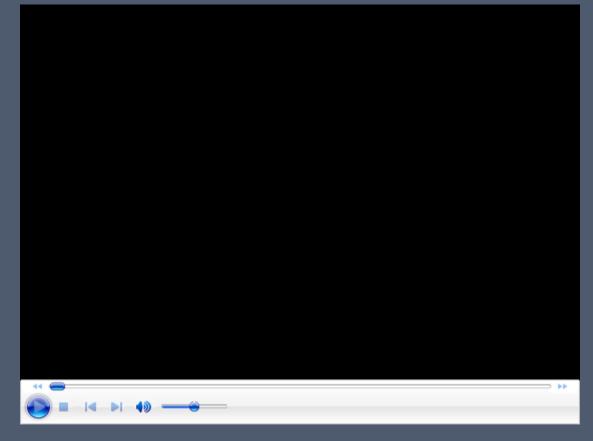
- Secondary rays
 - When ray hit object surface
- MipMap level chosen to approximate light source cone

Resulting integrated opacity

Fully compatible with the cache







Depth-Of-Field

- Similarly for depthof-field...
 - MipMap leveld based on circle-ofconfusion size

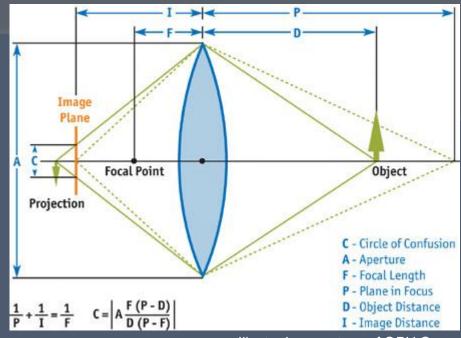
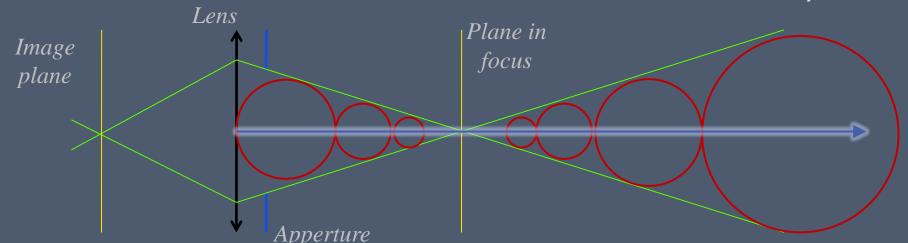
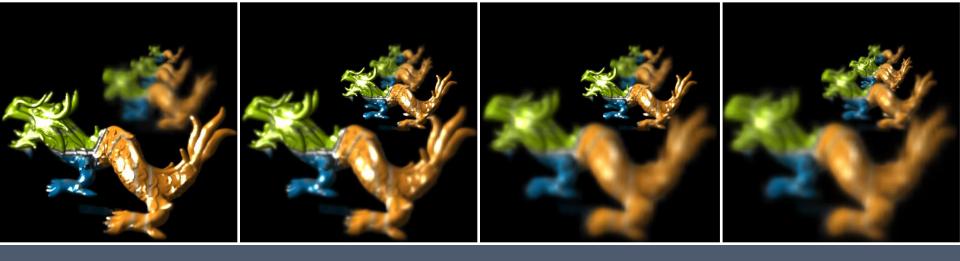
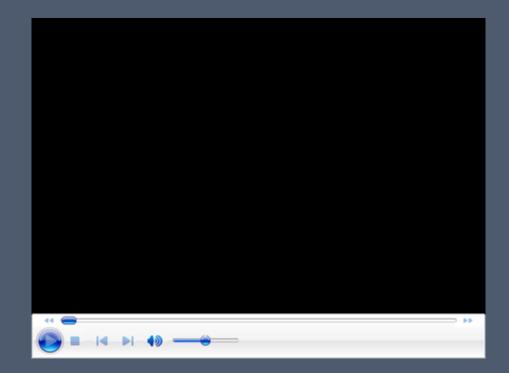


Illustration courtesy of GPU Gems







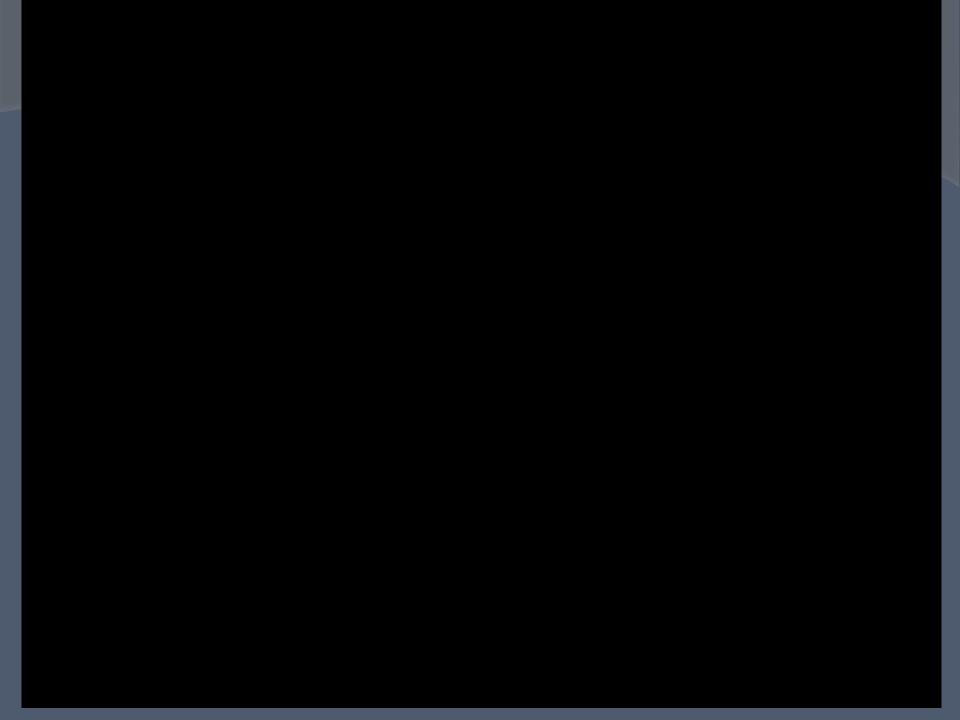
Ambient occlusion

- Uses one filtered sample
 - Covers the surrounding region





Without AO With AO



Future work direction

- Animation
 - Yes, this can be efficiently animated!
 - Volume deformation (skinning)
- Improved visibility integration
- Filtering
 - Shading/Normals
 - Isotropic pre-integration
 - Two walls problems

Many thanks go to ...

- Digisens Corporation
- Rhone-Alpes Explora'doc program
- Cluster of Excellence on Multimodal Computing and Interaction (M2CI)
- Output
 3D-Coat and Rick Sarasin
- Erklaerbar

