

DIRECT VOLUME RENDERING

PART II: SPEED UP, HANDLING LARGE DATA

ALEXANDER BOCK
MOORE-SLOAN POSTDOCTORAL FELLOW
CENTER FOR DATA SCIENCE
NEW YORK UNIVERSITY

- Slides:
 - http://alexanderbock.eu/lectures/2018/ds_ga_3001_017_volumerendering_optimization.pdf
 - http://alexanderbock.eu/lectures/2018/ds_ga_3001_017_volumerendering_optimization/

WHY NOT BRUTE-FORCE



- 1. Video
- 4K HDR video (raw)
$$3840 \times 2160 \text{ pixels/frame} * 3 \text{ colors/pixel} * 10 \text{ bits/color} / 8 \text{ bits/bytes} * 60 \text{ frames/second} = 1.866 \text{ GB/s}$$
- 2. Time-varying volumetric rendering









WHY NOT BRUTE-FORCE

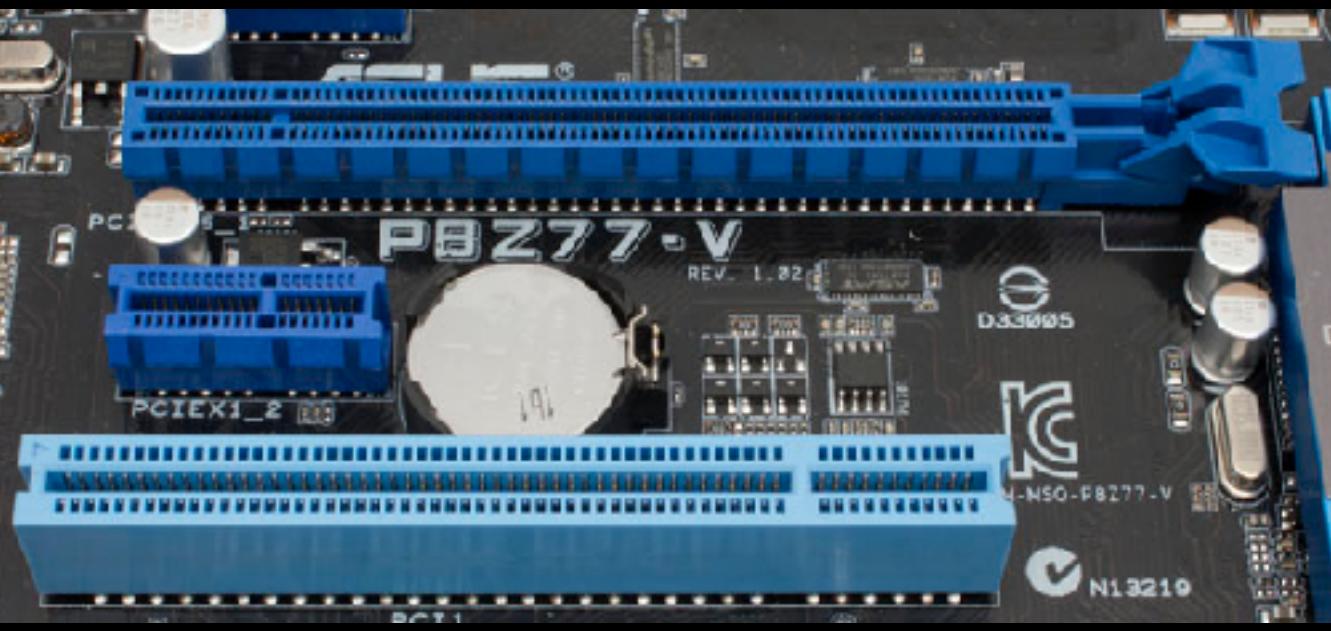
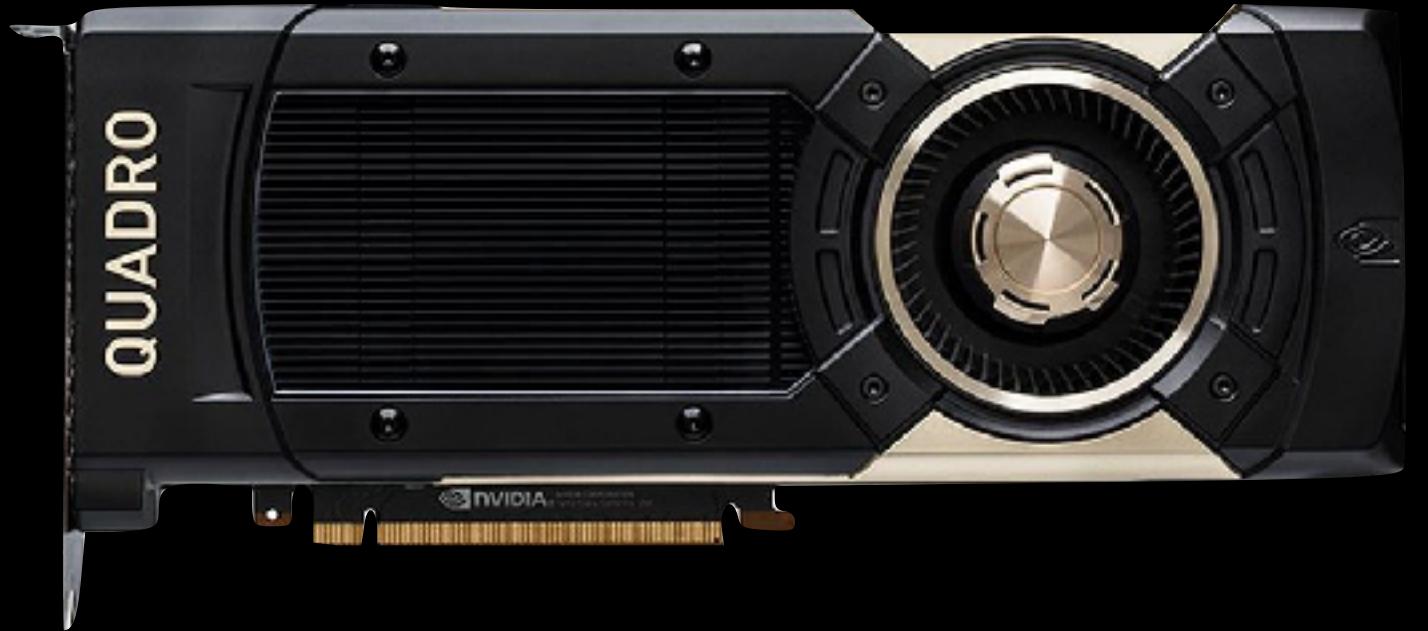


- 1. Video
 - 4K HDR video (raw)
 $3840 \times 2160 \text{ pixels/frame} * 3 \text{ colors/pixel} * 10 \text{ bits/color} / 8 \text{ bits/bytes} * 60 \text{ frames/second} = 1.866 \text{ GB/s}$
- 2. Time-varying volumetric rendering
 - Spatial resolution: $875 \times 556 \times 3052$ voxel
 - Temporal resolution: 66 ms
 - $875 \times 556 \times 3052 \text{ voxel/frame} * 4 \text{ byte/voxel} * 15 \text{ frames/second} = 89 \text{ GB/s}$



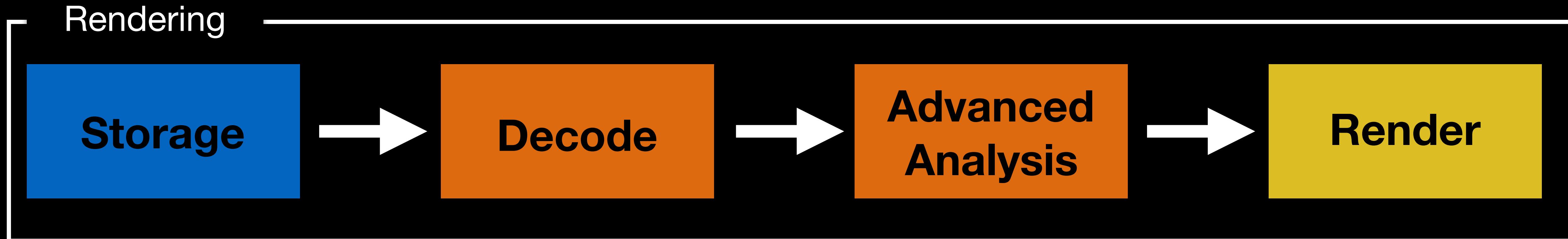
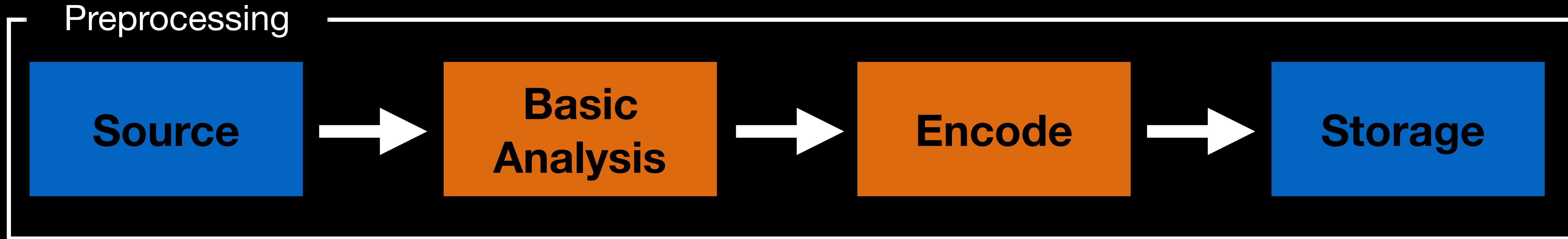
MEMORY SPEED

- NVIDIA Quadro GV100: 870 GB/s
- PCI-Express 4.0 @ x16: 32 GB/s
- DDR4-3200: 25.6 GB/s
- SSD: (Samsung 860 Pro) 550 MB/s
- -> Data reduction techniques are essential



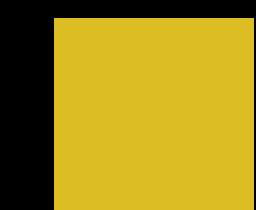
HANDLING LARGE DATA BRICKING

VOLUME PIPELINE



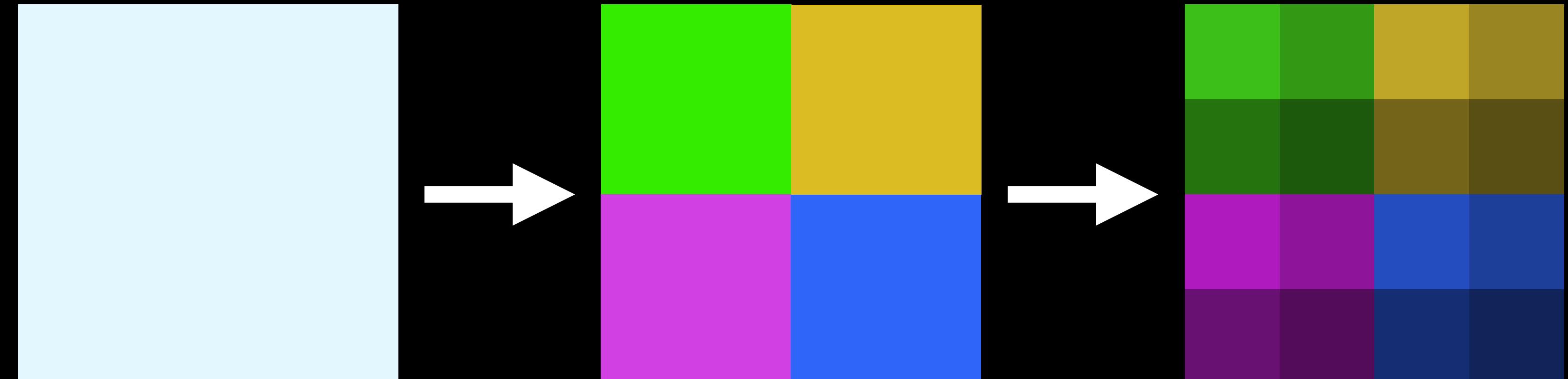
 Data

 Process

 System

QUADTREE

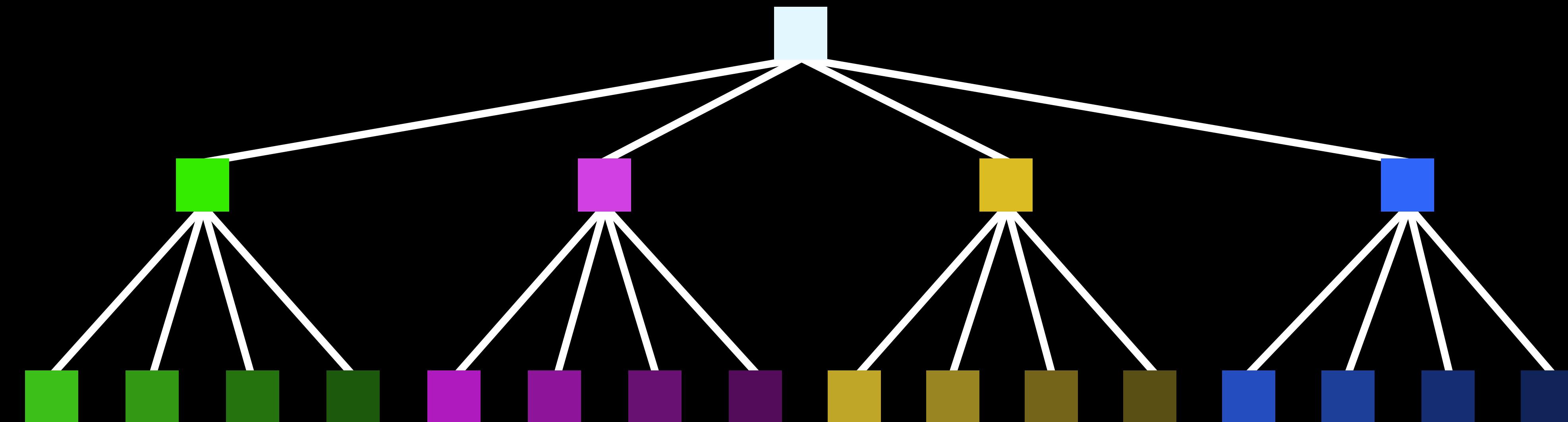
- Data is organized in levels of varying resolution
- Higher level
-> lower resolution



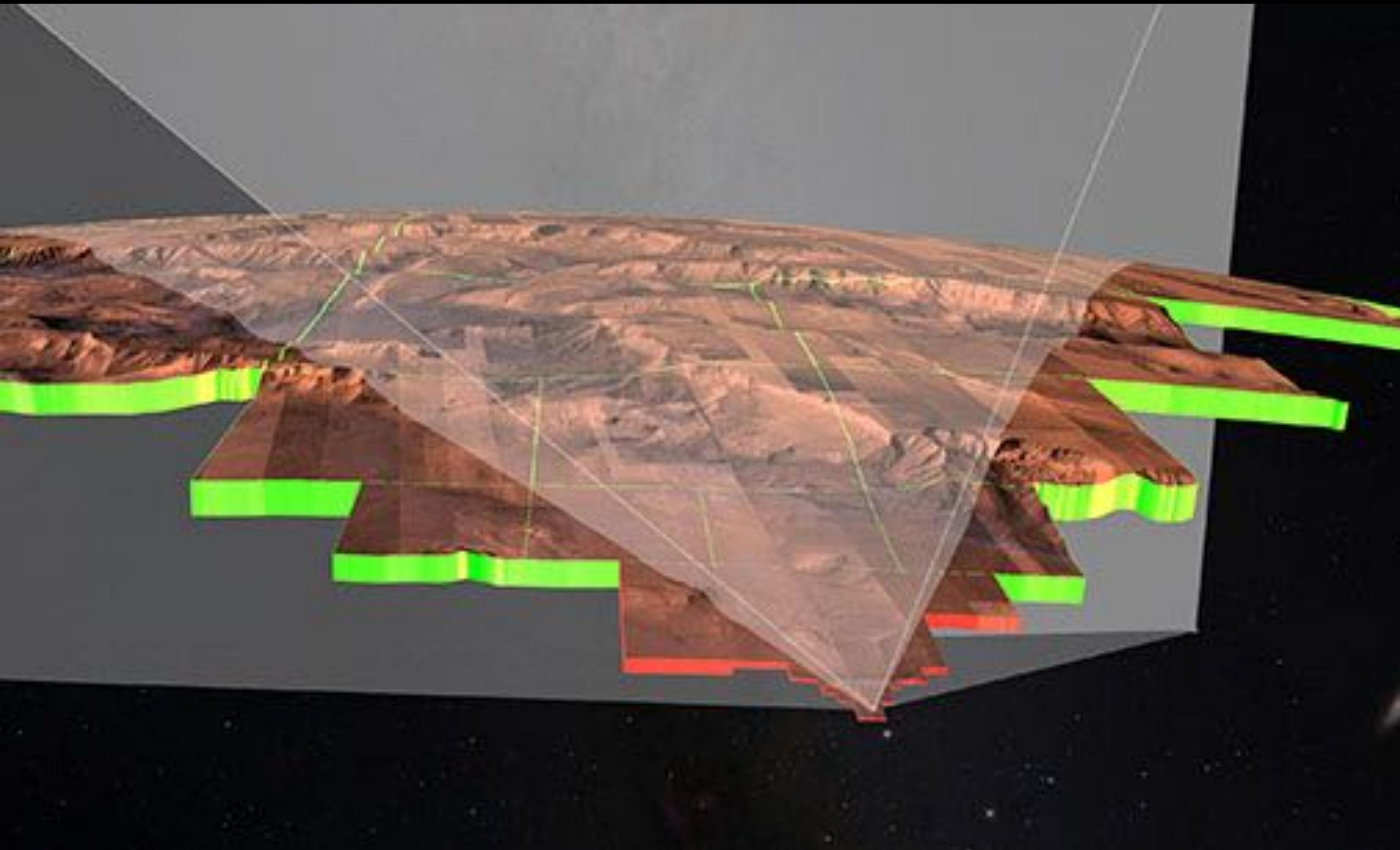
Level 0

Level 1

Level 2

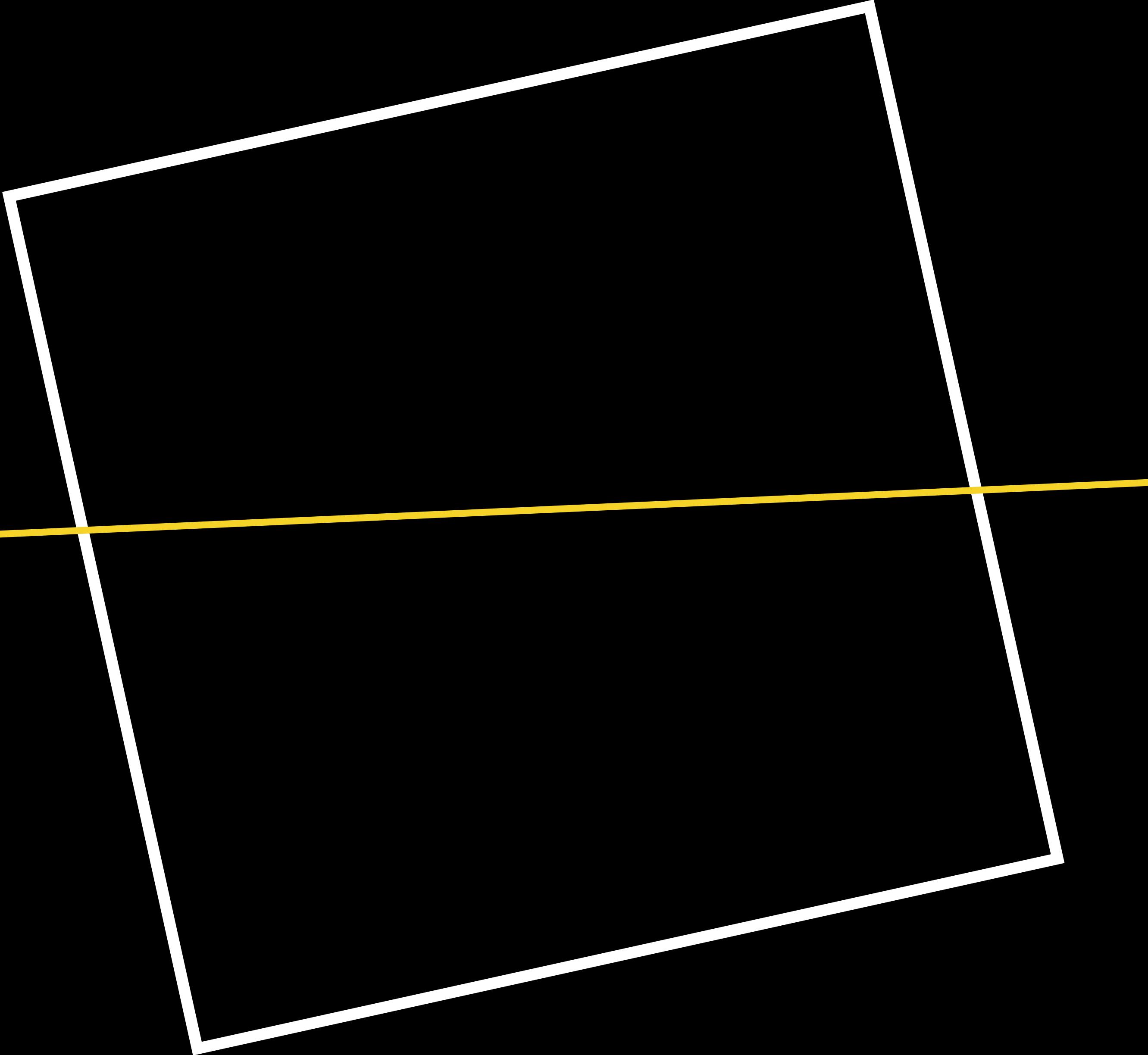
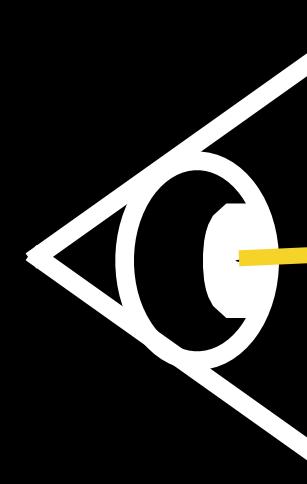


QUADTREE - EXAMPLE



OCTREE

- Quadtree: Spatial subdivision in 2D
- Octree: Spatial subdivision in 3D



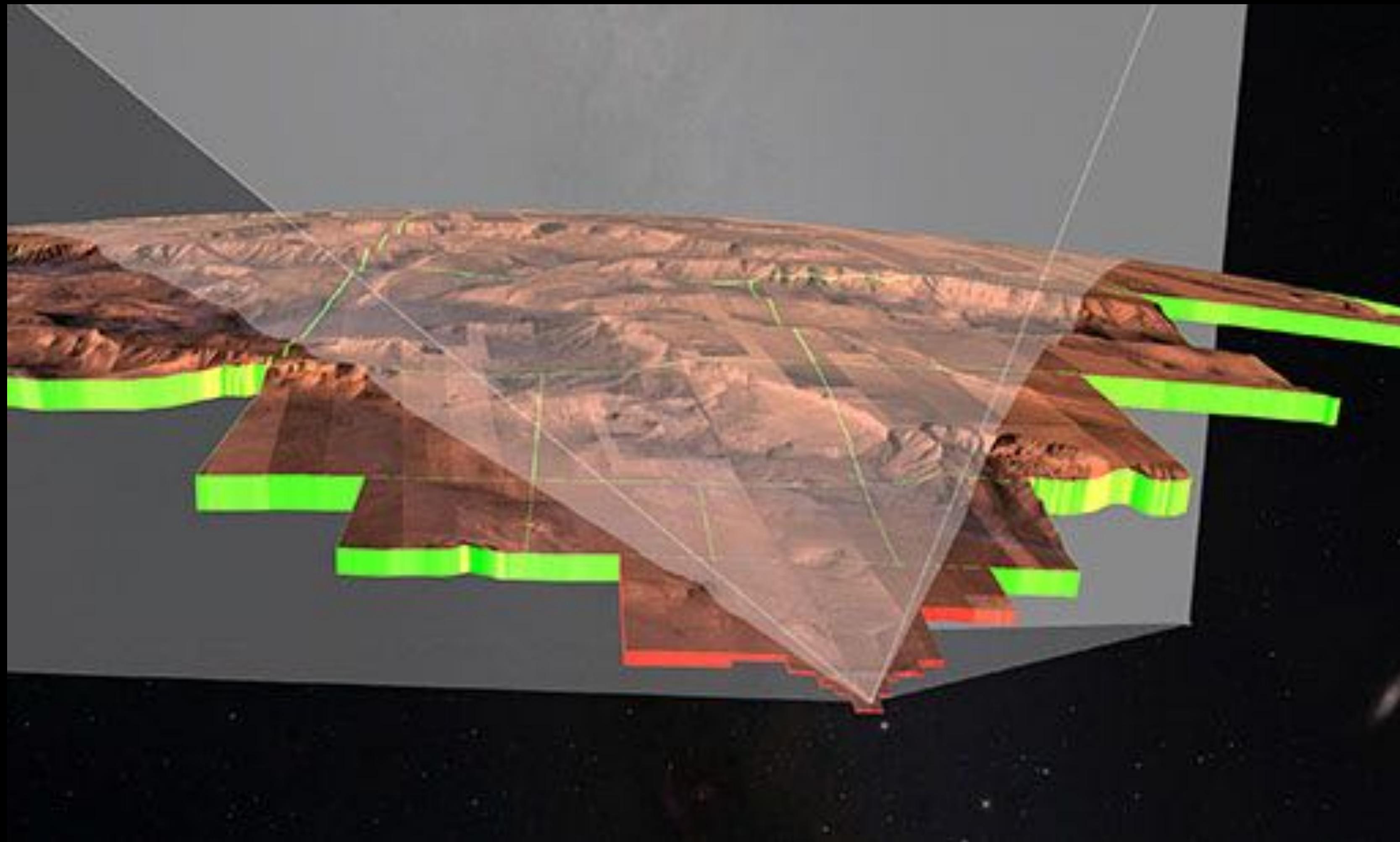
OCTREE-BASED VOLUME RENDERING

- During sampling:
 1. Get position of sample in volume coordinates
 2. Determine the desired octree level
 3. Load the correct block from the octree
 4. Convert global sample position to local coordinate system
 5. Sample in block



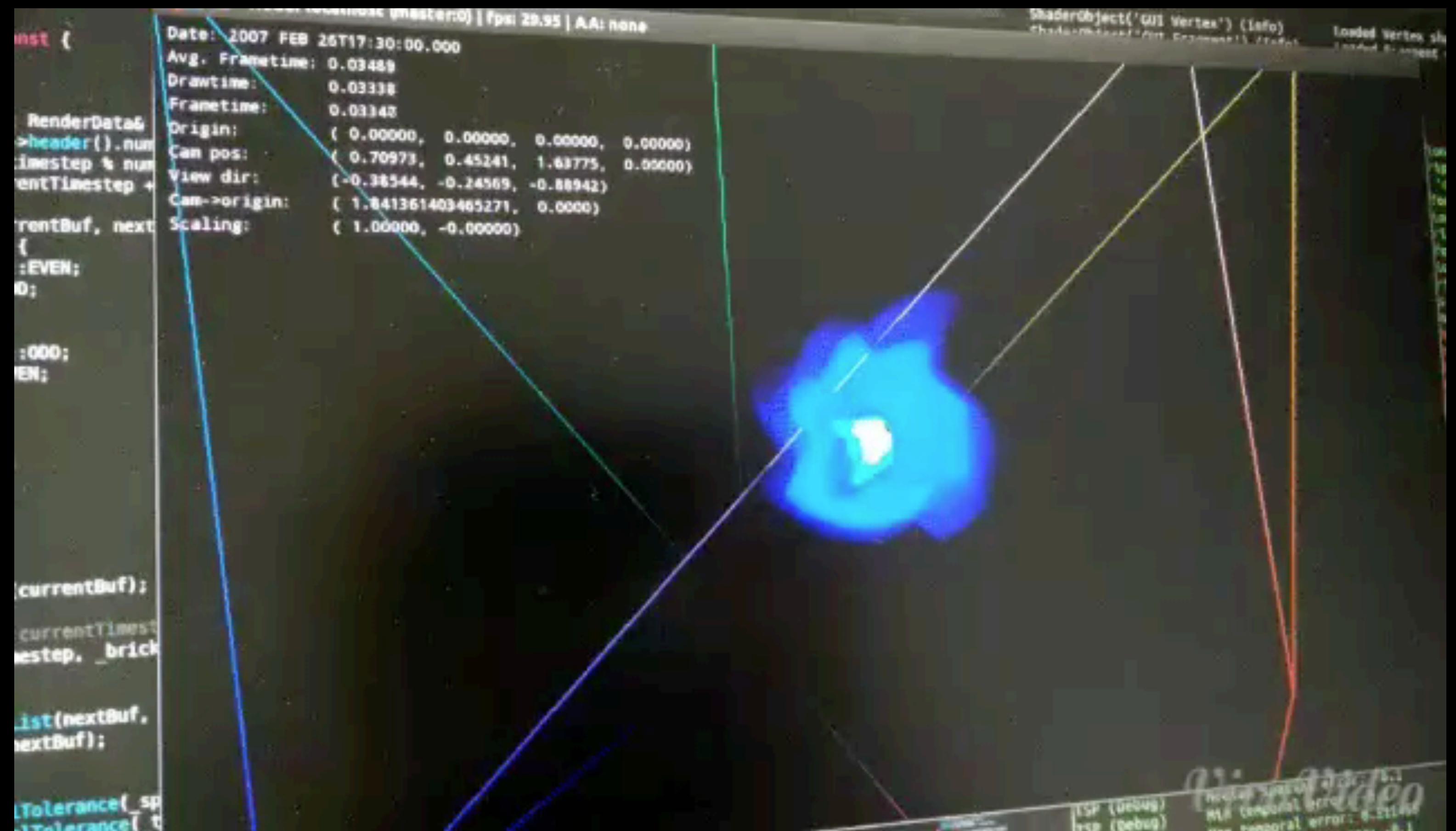
OCTREE LEVEL TRAVERSAL

- Good first heuristic:
The closer a block is to
the camera, the higher
level to use



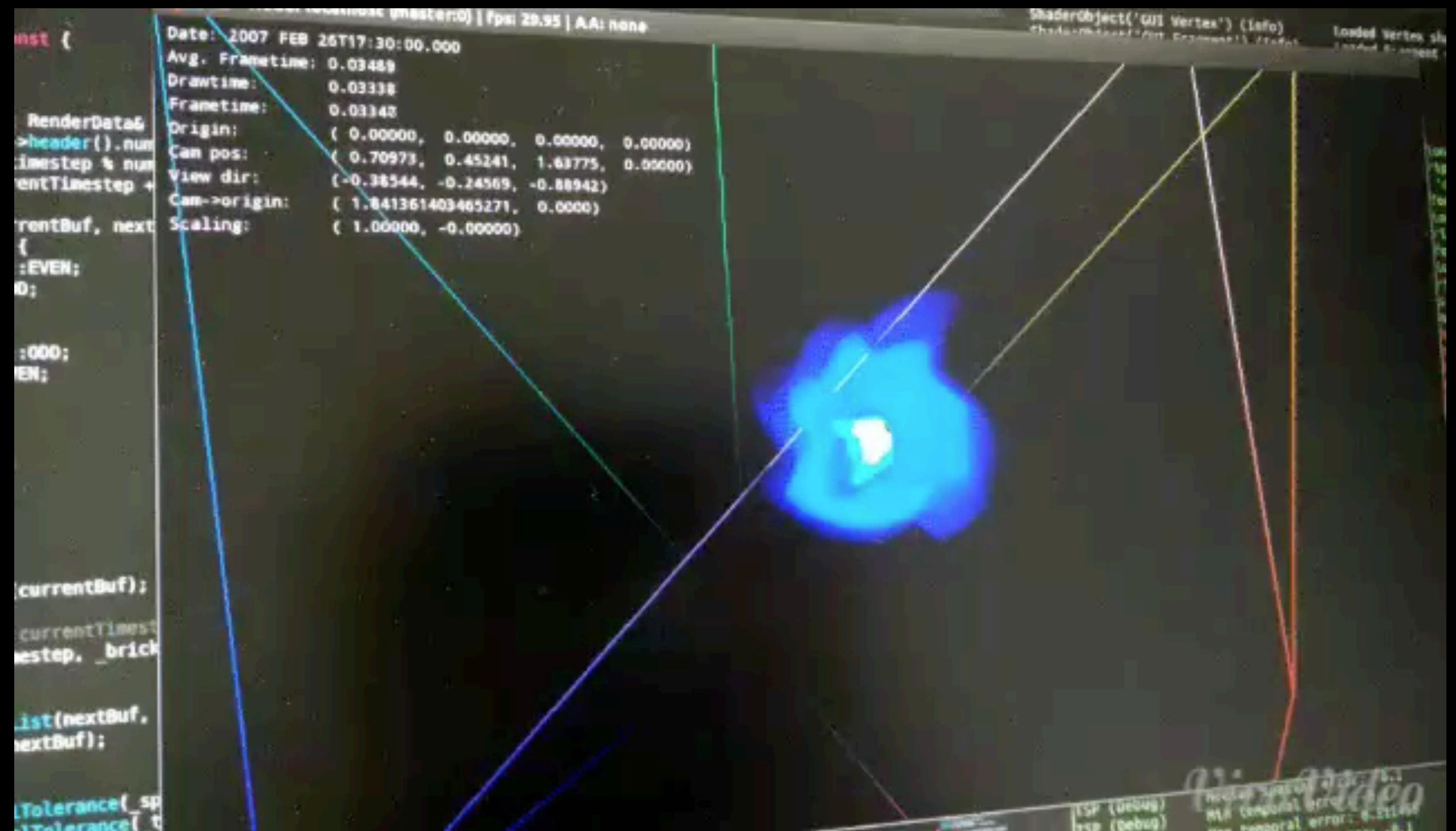
OCTREE LEVEL TRAVERSAL

- Good first heuristic:
The closer a block is to
the camera, the higher
level to use
 - Also works for volumetric
data



OCTREE LEVEL TRAVERSAL

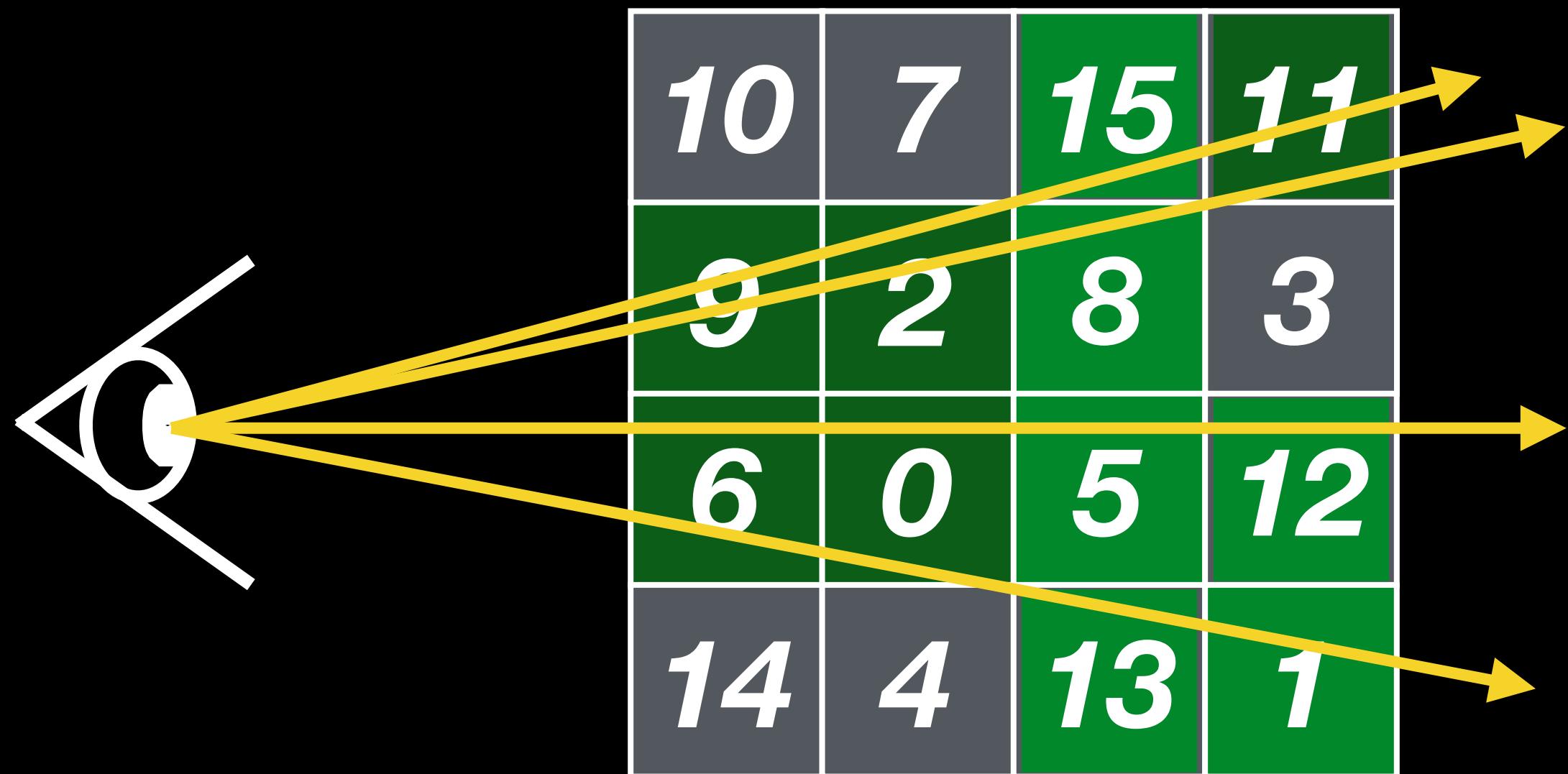
- Good first heuristic:
The closer a block is to
the camera, the higher
level to use
 - Also works for volumetric
data



OCTREE TRAVERSAL - PROBE RAYS

Count	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

index 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15



Count	2	1	2	0	0	1	2	0	1	2	0	2	1	1	0	1
-------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

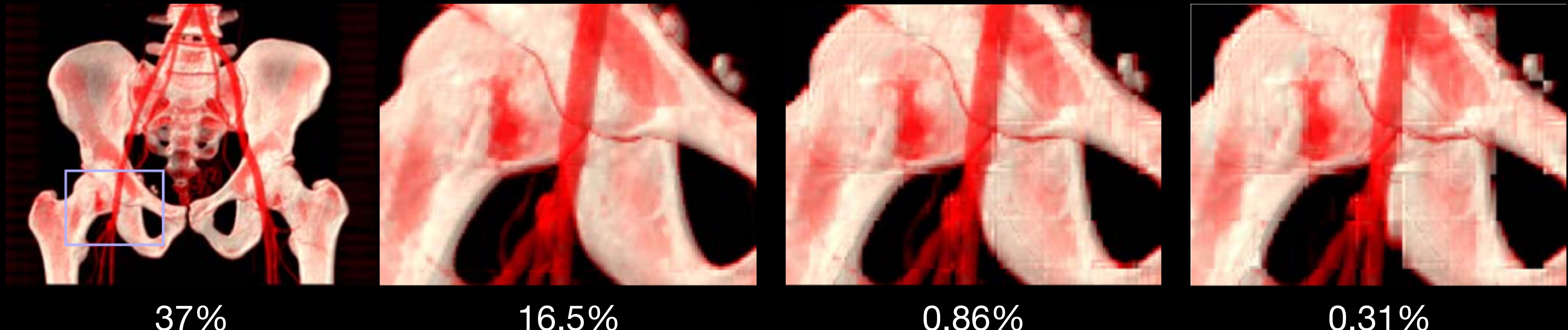
index 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

KNOWLEDGE-ENCODING

Ljung, P., Lundström, C., Ynnerman, A., & Museth, K. Transfer function based adaptive decompression for volume rendering of large medical data sets. In IEEE Symposium on Volume Visualization and Graphics, 2004

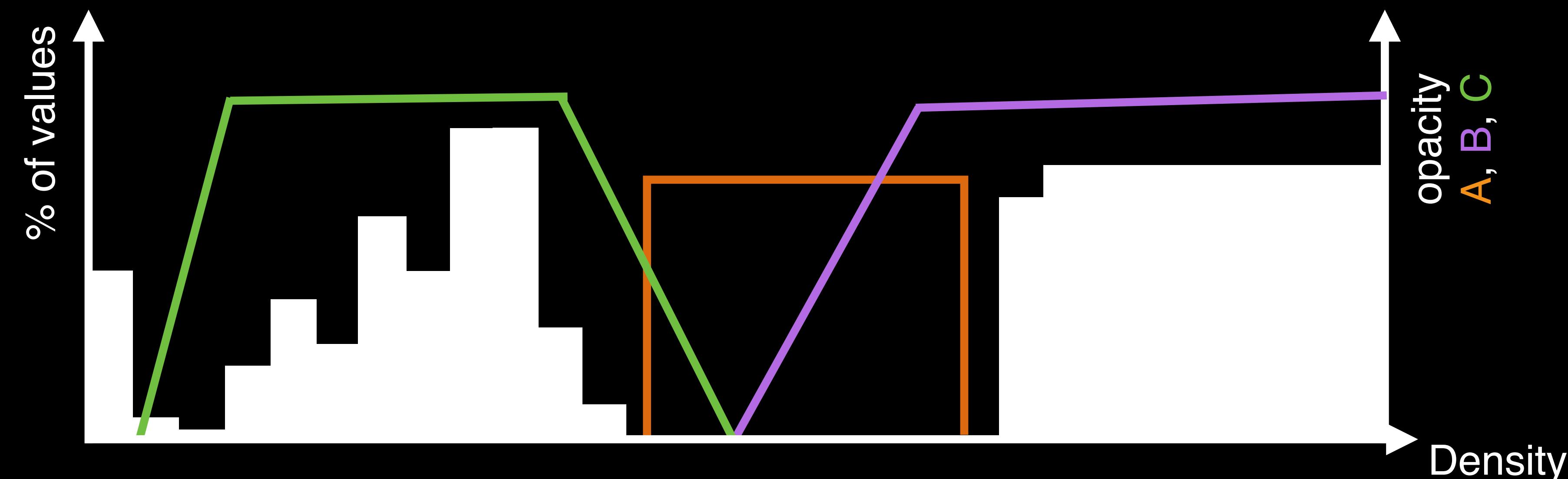
KNOWLEDGE ENCODING

- Idea: The user provides information about what is important through the transfer function
- Only load high resolution blocks in which the transfer function mapped change
- Example: Percentage of remaining data after filtering



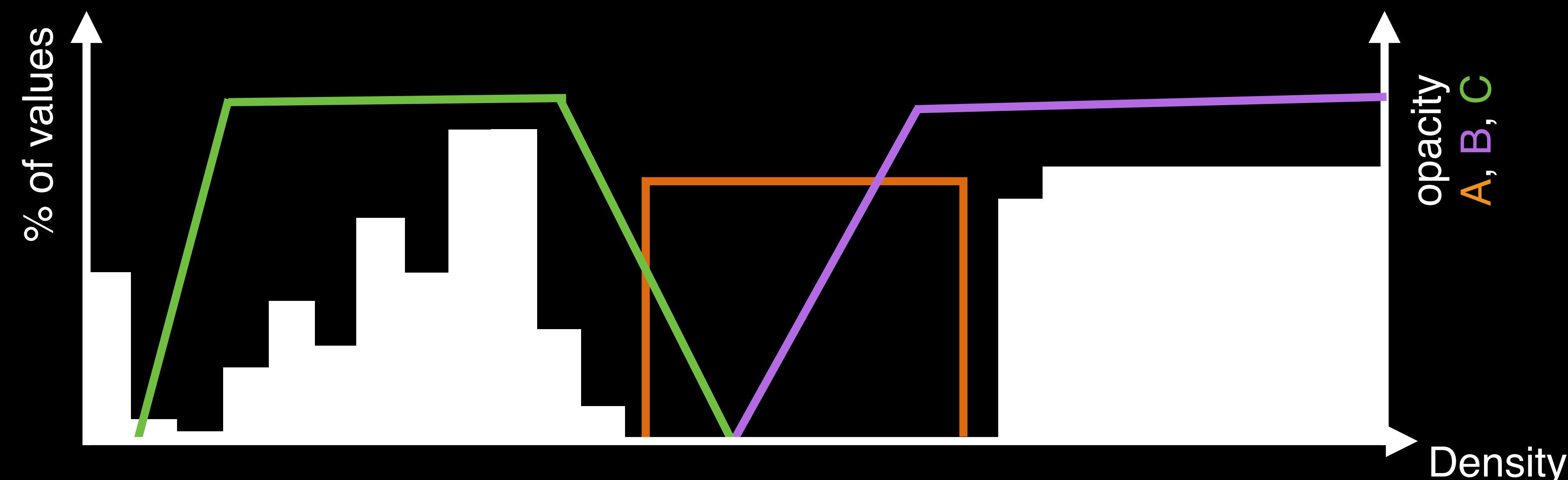
KNOWLEDGE ENCODING

- Idea: The user provides information about what is important through the transfer function
- Only load high resolution blocks in which the transfer function mapped change
- Example: Histogram of **single block, multiple potential global transfer functions, A, B, C**



KNOWLEDGE ENCODING

- **A:** For the selected transfer function all voxels in the block are invisible
→ The block will not contribute to the rendered image
- **B:** The voxel histogram and transfer function is constant
→ Nothing interesting is happening and a lower resolution block will suffice
- **C:** The voxel histogram changes dramatically, so we need high resolution here



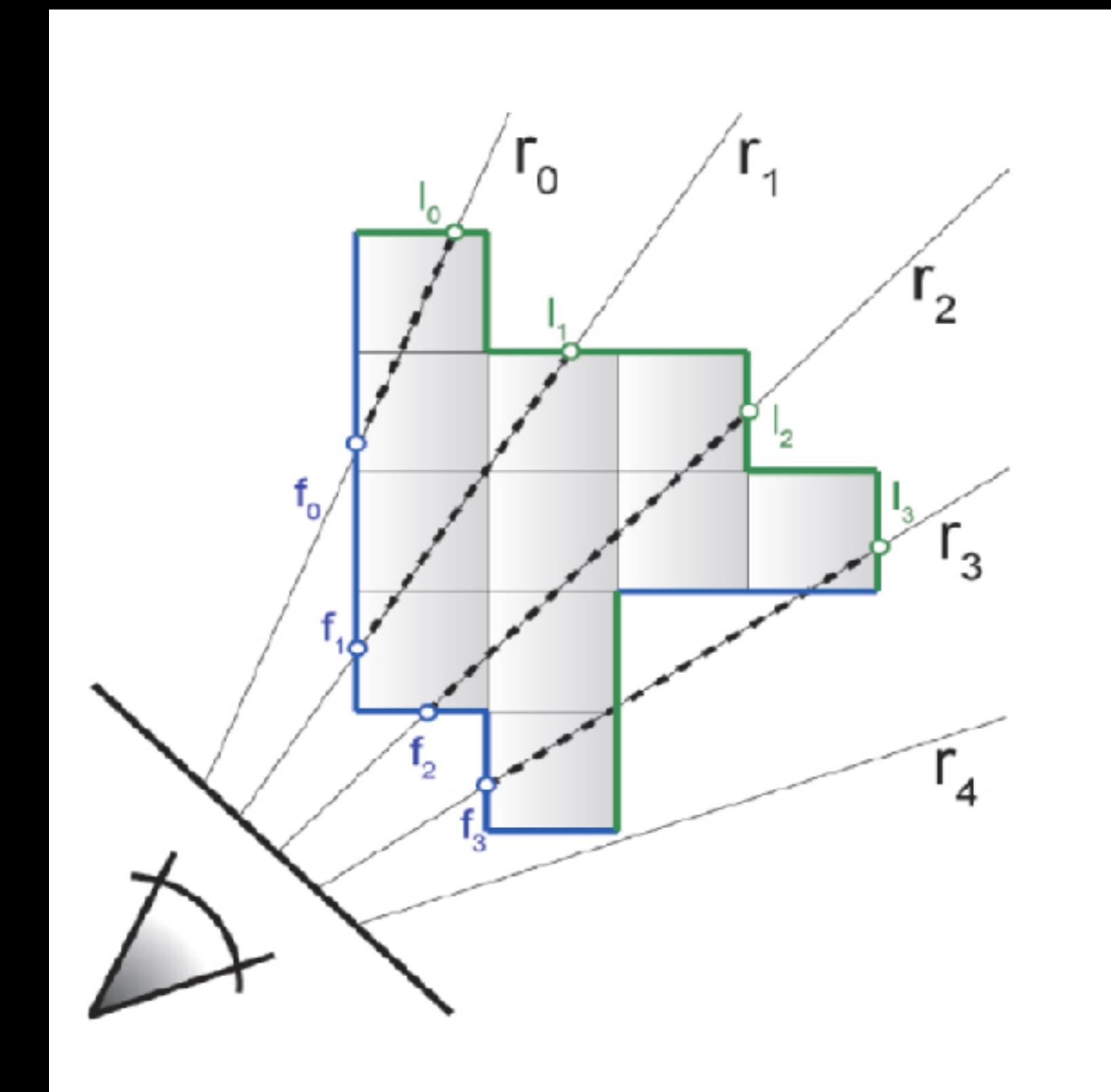
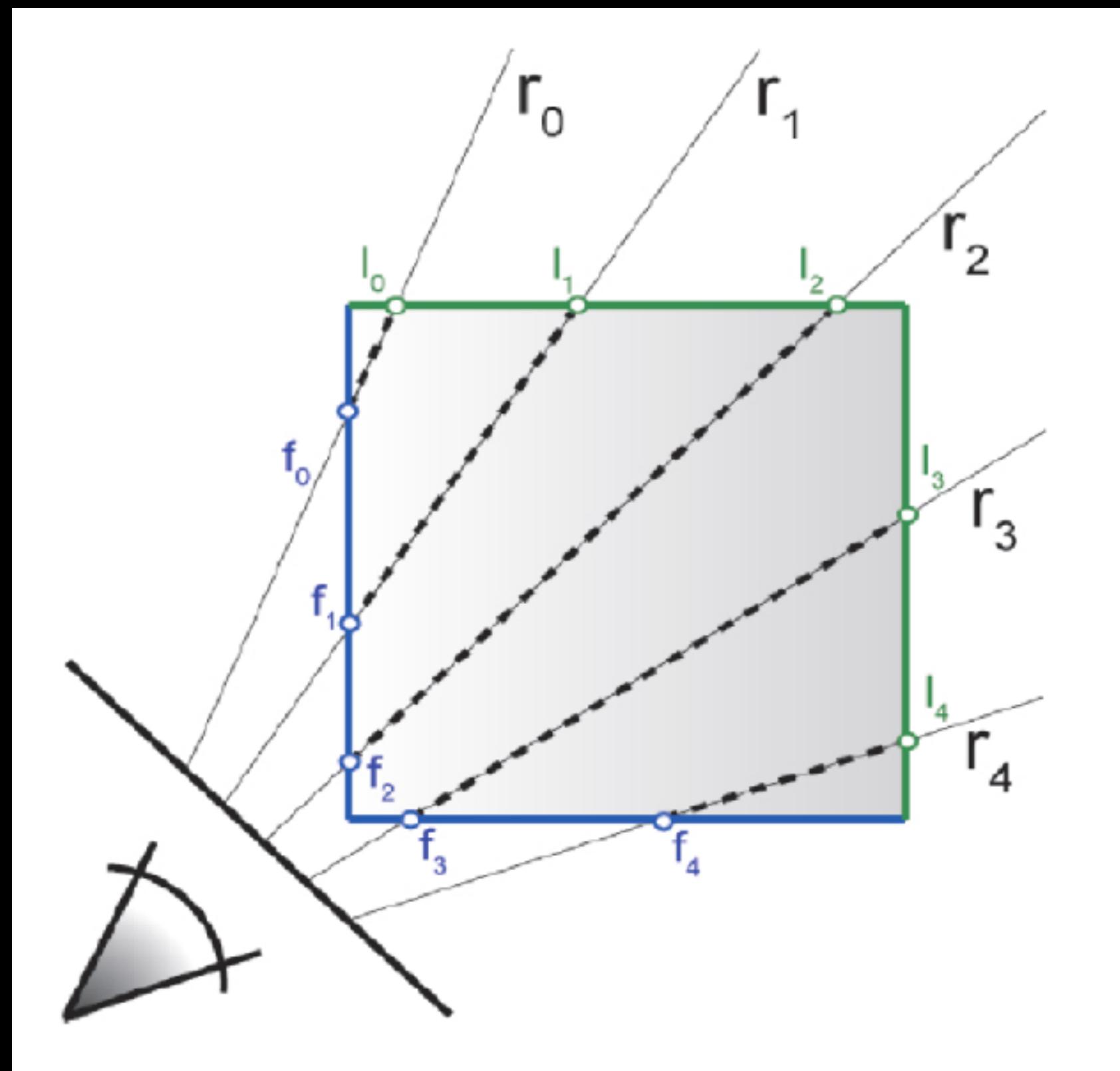
EMPTY SPACE SKIPPING

EMPTY SPACE SKIPPING

- Object-order: Modify bounding box to approximate the volume better
- Image-order: Provide additional information to improve the sampling

OBJECT-ORDER EMPTY SPACE SKIPPING

- Construct a coarse grid over the volume (for example 16^3 voxels)
- Store minimum/maximum voxel values per resampled grid and test against transfer function

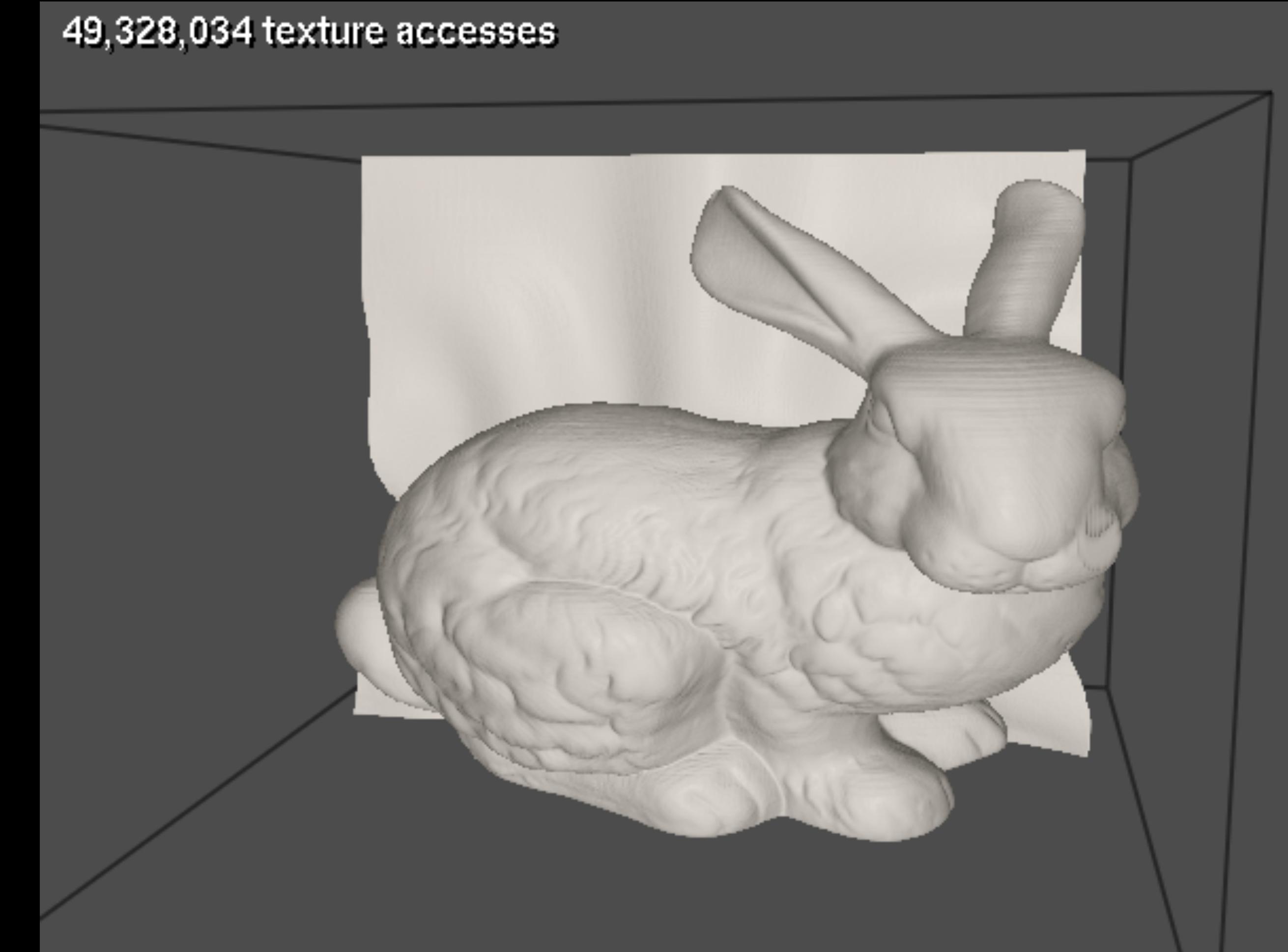


Images from
Stefan Bruckner

OBJECT-ORDER EMPTY SPACE SKIPPING

- Volume rendering

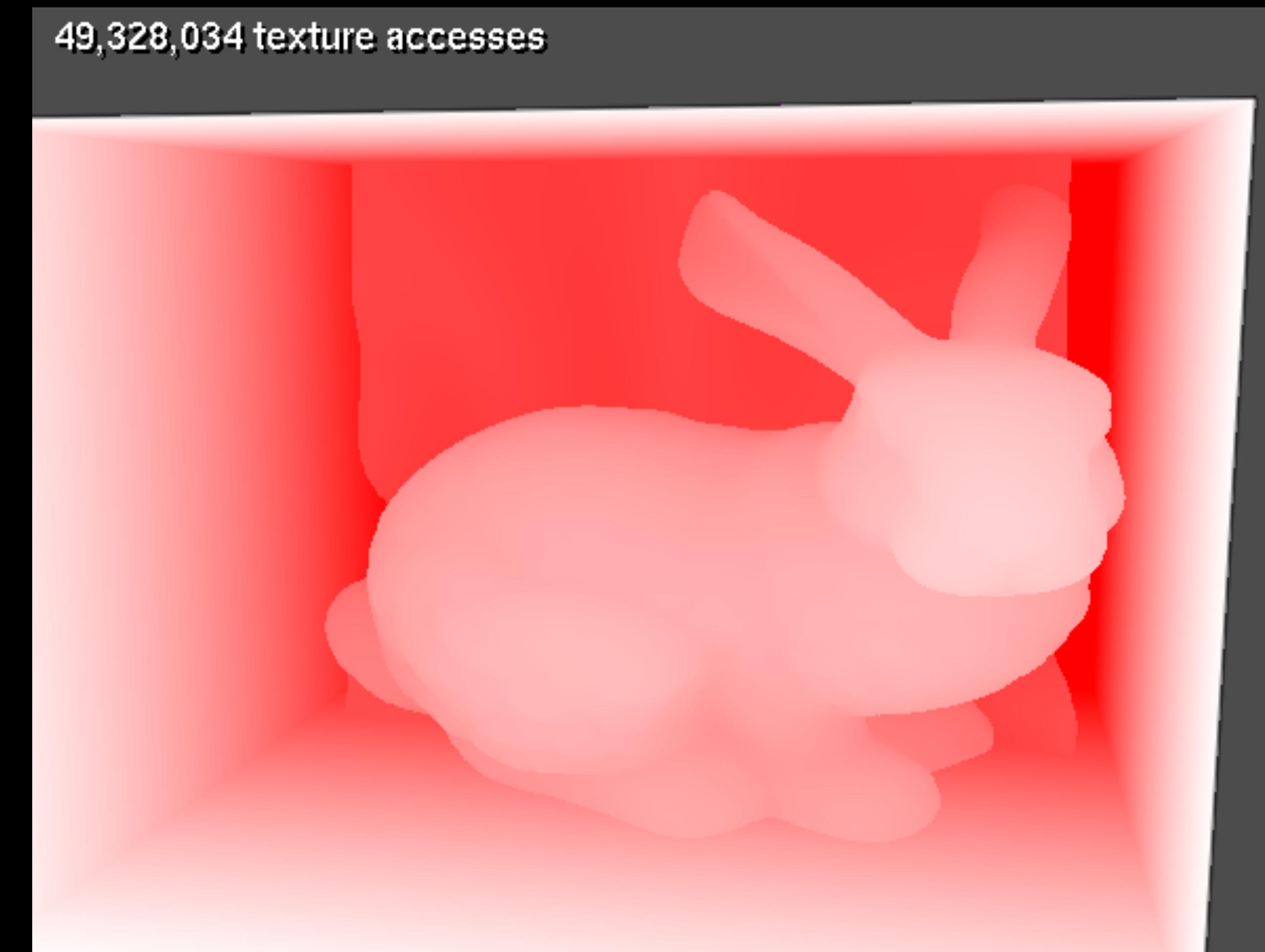
49,328,034 texture accesses



Images from <http://haferburg.github.io>

OBJECT-ORDER EMPTY SPACE SKIPPING

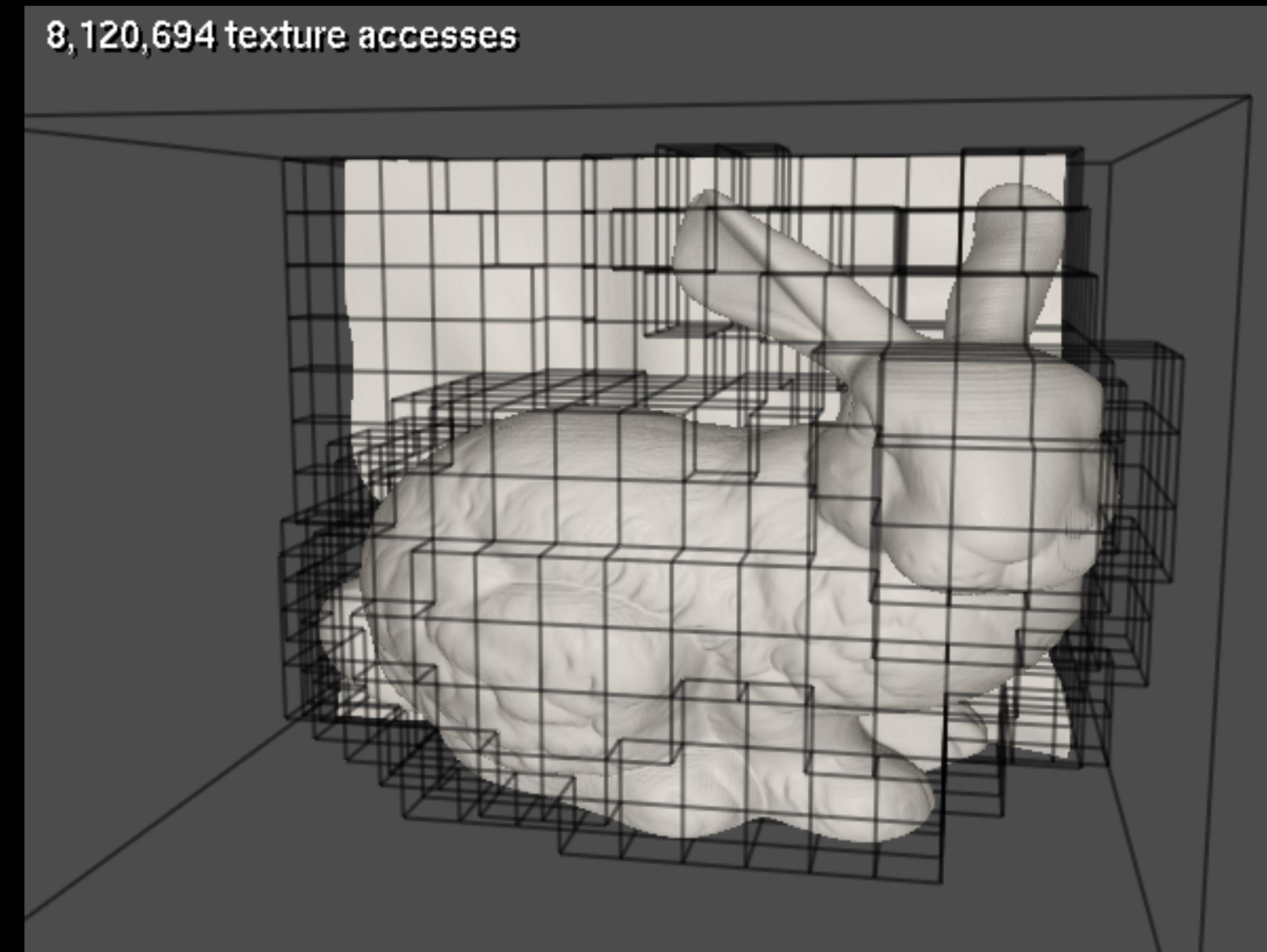
- Volume rendering
- Number of texture accesses per pixel
darker → more accesses



Images from <http://haferburg.github.io>

OBJECT-ORDER EMPTY SPACE SKIPPING

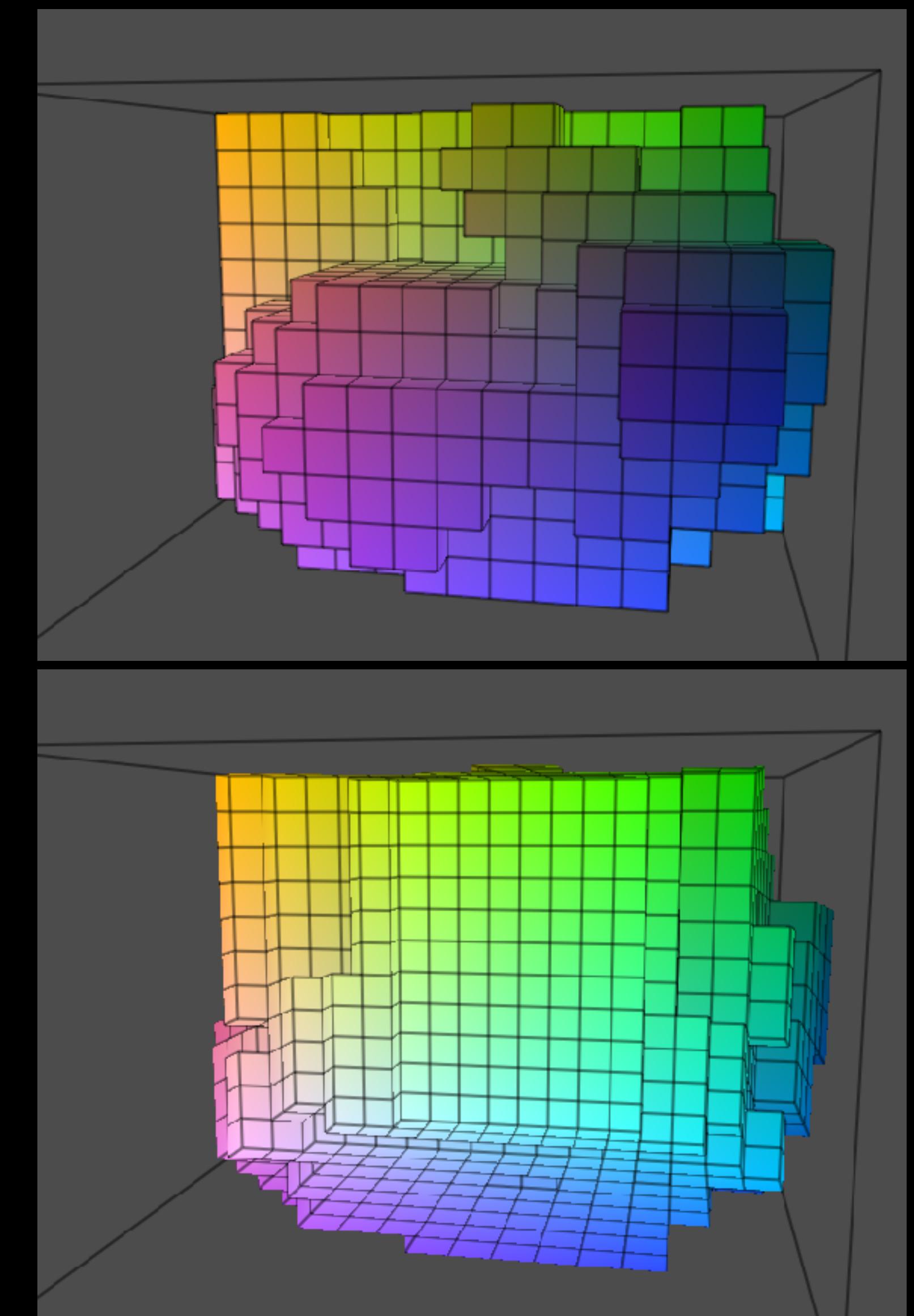
- Volume rendering
- Number of texture accesses per pixel
darker → more accesses
- Create the grid of filled voxels



Images from <http://haferburg.github.io>

OBJECT-ORDER EMPTY SPACE SKIPPING

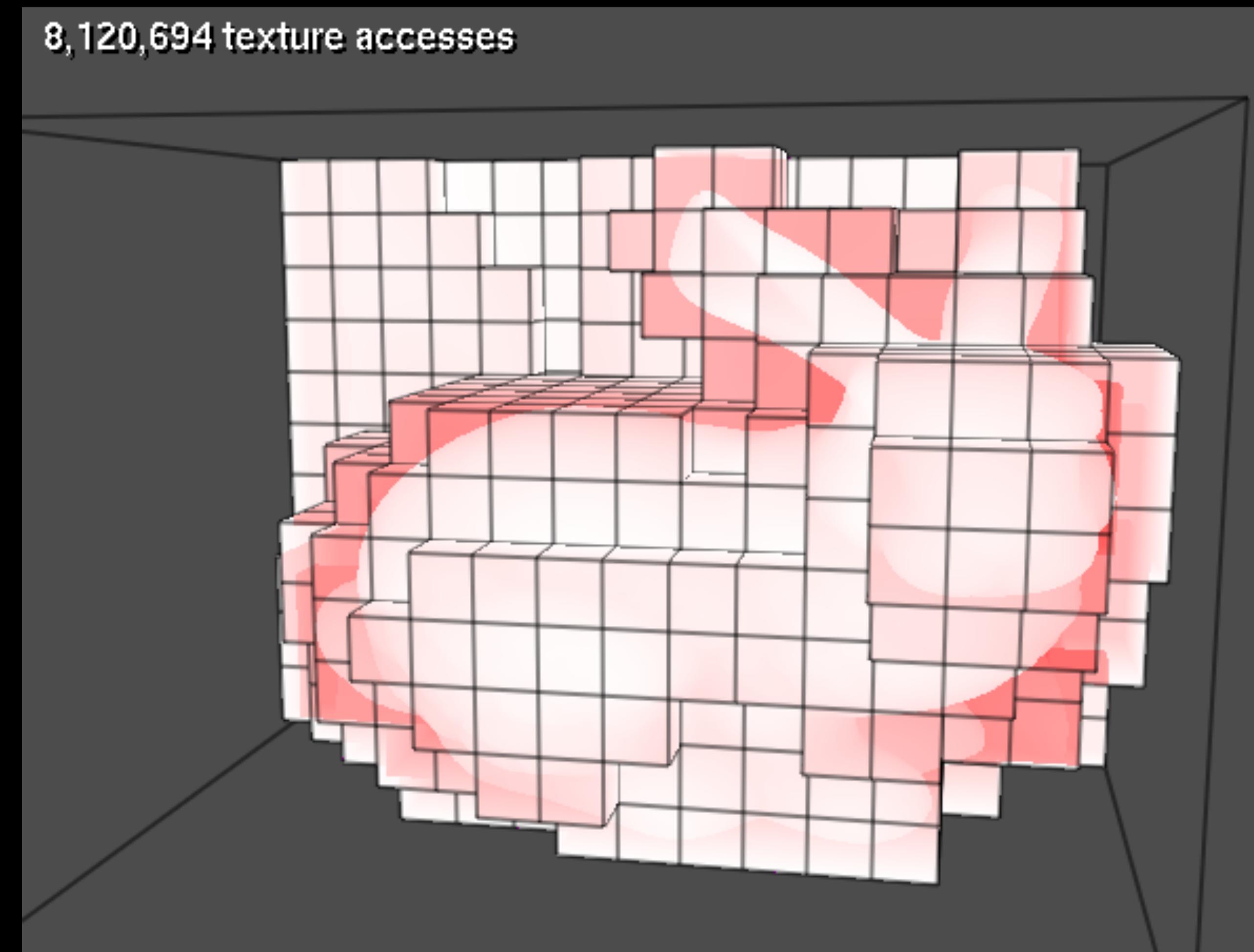
- Volume rendering
- Number of texture accesses per pixel
darker → more accesses
- Create the grid of filled voxels
- Render entry and exit point images



Images from <http://haferburg.github.io>

OBJECT-ORDER EMPTY SPACE SKIPPING

- Volume rendering
- Number of texture accesses per pixel
darker → more accesses
- Create the grid of filled voxels
- Render entry and exit point images
- Initiate volume rendering
- Saving, in this case, about 82% of texture accesses



Images from <http://haferburg.github.io>

IMAGE-ORDER EMPTY SPACE SKIPPING / ADAPTIVE SAMPLING

- Create a second helper volume that provides neighborhood information
 - Each voxel of the helper volume encodes the minimum distance to the nearest inhabited voxel
 - Use, for example, transfer function to encode which voxels participate in the neighborhood
- During ray marching, sample the helper volume to modify the step length

IMAGE-ORDER EMPTY SPACE SKIPPING / ADAPTIVE SAMPLING

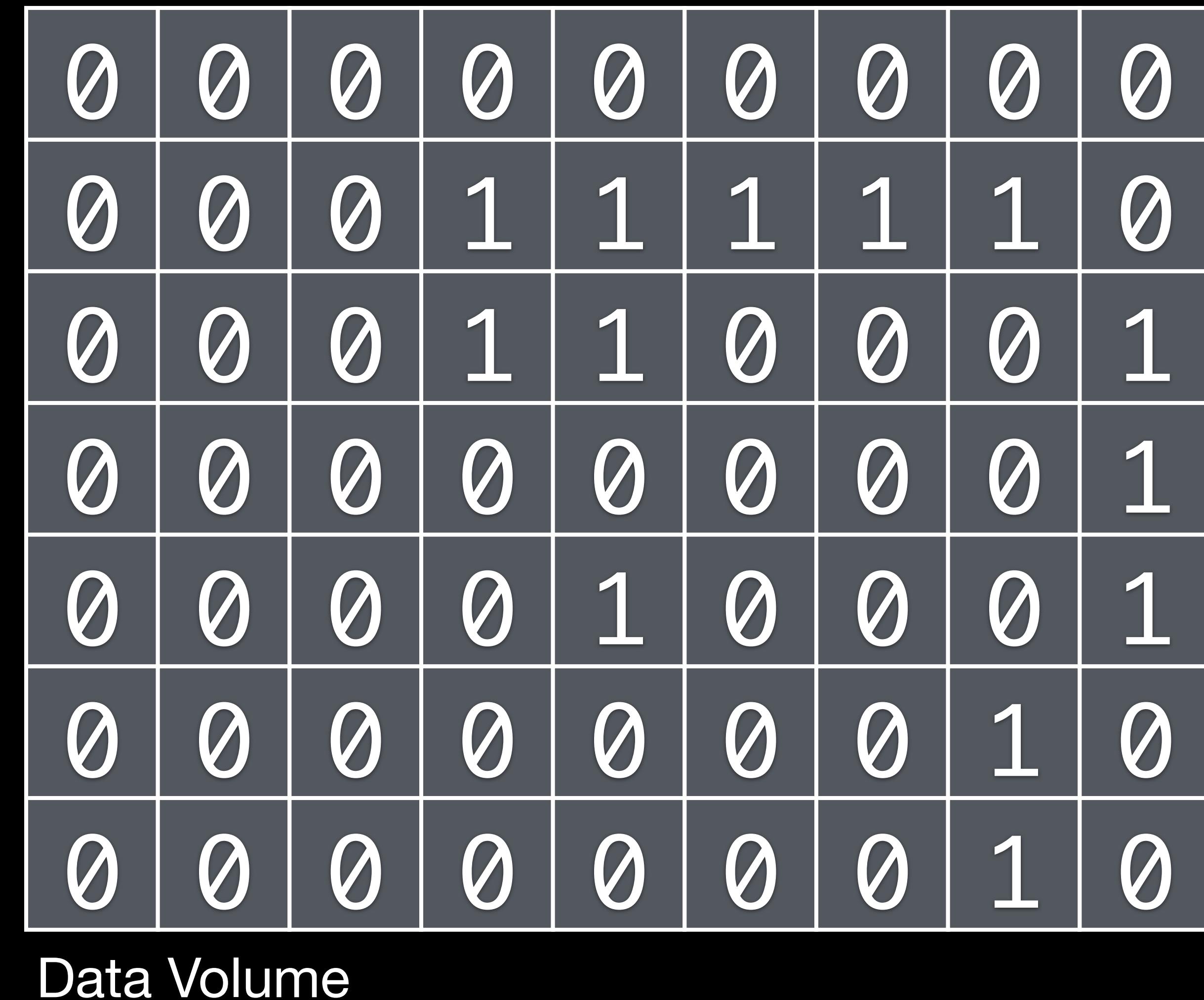
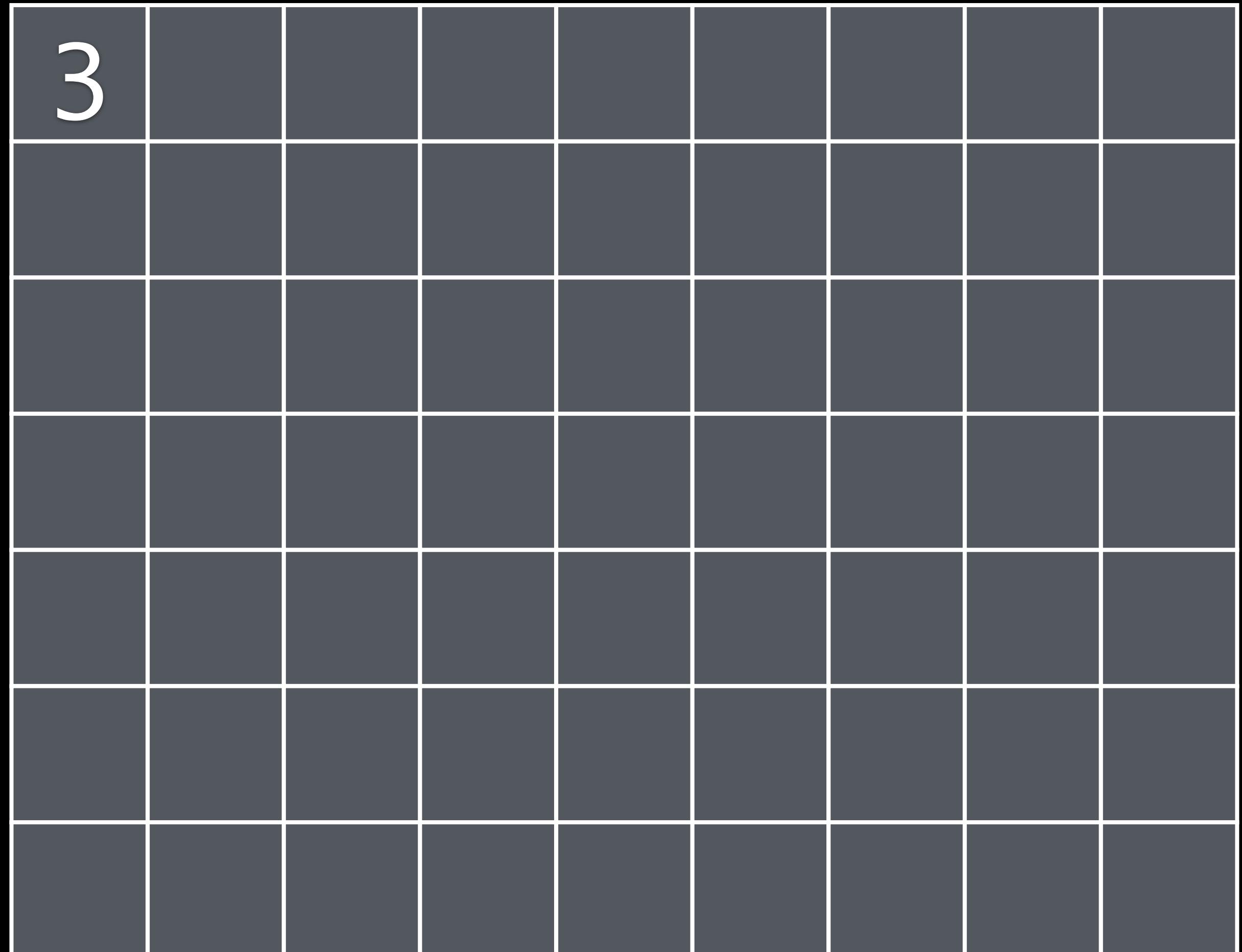


IMAGE-ORDER EMPTY SPACE SKIPPING / ADAPTIVE SAMPLING



Skipping Volume



Data Volume

IMAGE-ORDER EMPTY SPACE SKIPPING / ADAPTIVE SAMPLING

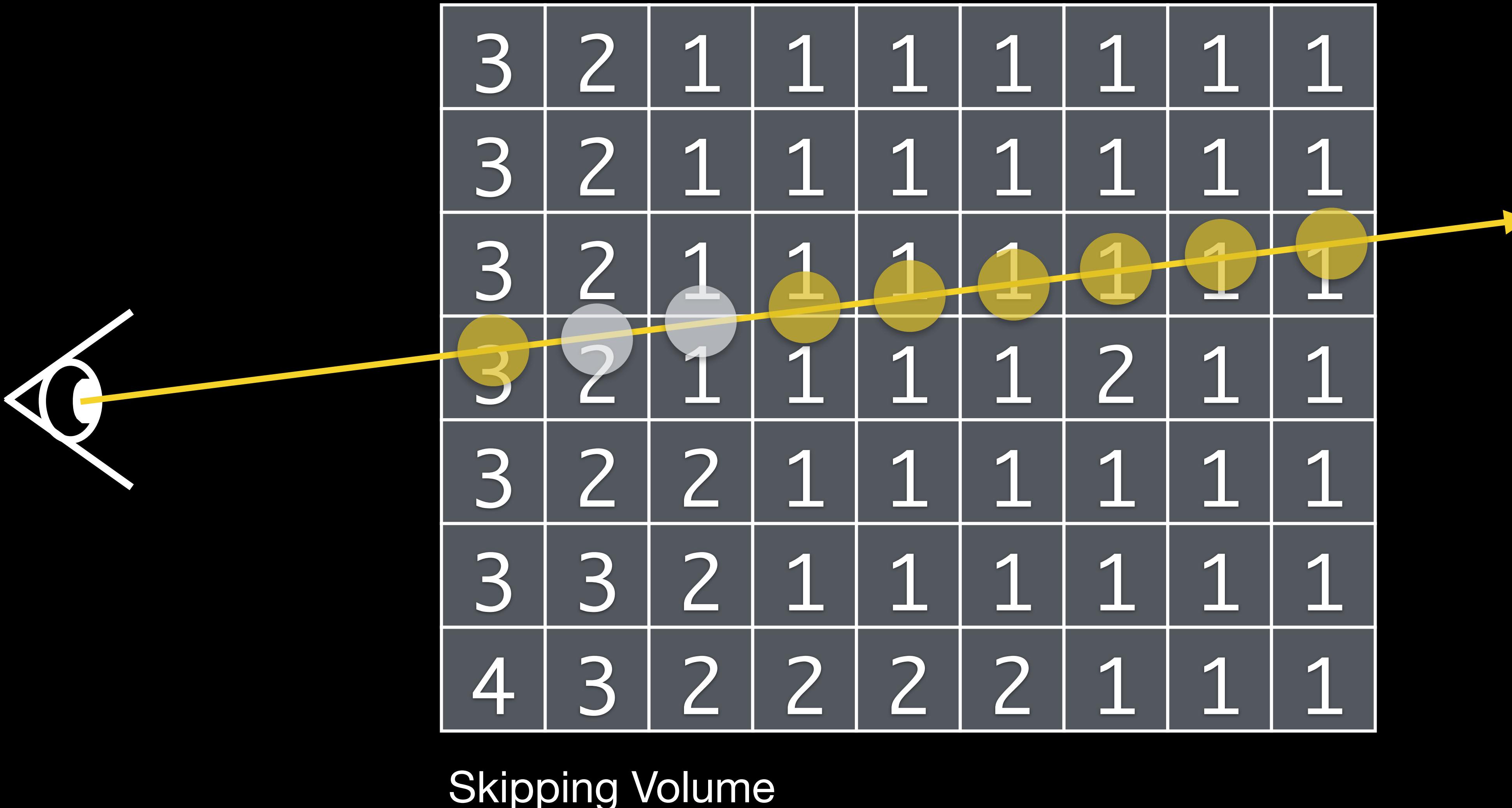
3	2	1	1	1	1	1	1	1	1
3	2	1	1	1	1	1	1	1	1
3	2	1	1	1	1	1	1	1	1
3	2	1	1	1	1	2	1	1	1
3	2	2	1	1	1	1	1	1	1
3	3	2	1	1	1	1	1	1	1
4	3	2	2	2	2	1	1	1	1

Skipping Volume

0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0

Data Volume

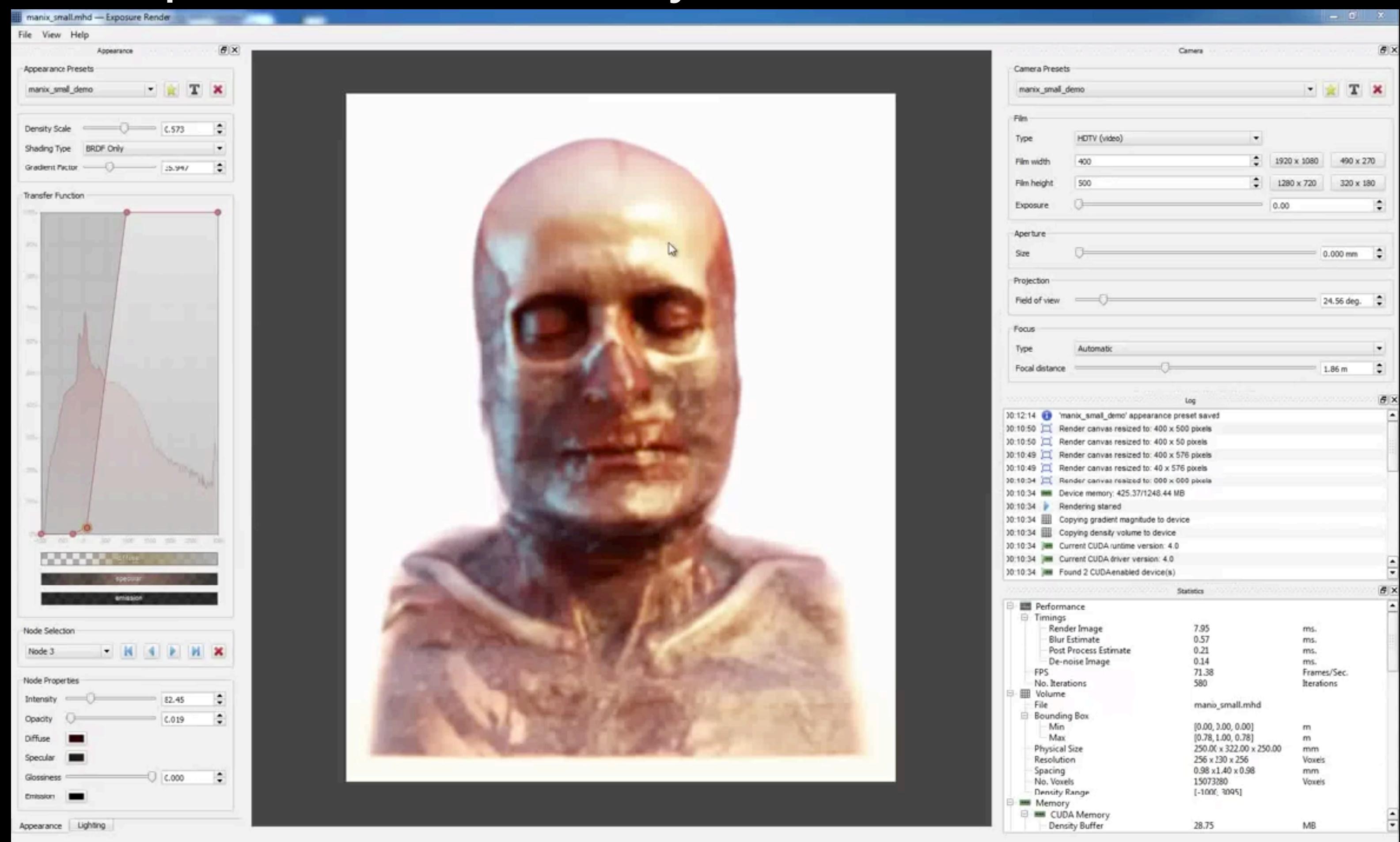
IMAGE-ORDER EMPTY SPACE SKIPPING / ADAPTIVE SAMPLING



ITERATIVE REFINEMENT

STOCHASTIC SAMPLING

- Instead of creating a ray for each pixel, stochastically determine a subset of pixels to render per frame



STOCHASTIC SAMPLING

- Instead of creating a ray for each pixel, stochastically determine a subset of pixels to render per frame

