Multicore World 2025
Sparsification

David Brebner

umajin

Context for Sparsification

- 3D data
- Time series data
- Simulation or AI extrapolated metadata expansion

- **Exponential data growth**

Tuned to find known quality defects

Sparsification

- Orin Nano Super
  - I showed the Orin Nano last year
  - This year same hardware is now “super”
  - Where is Moore's law for GPU’s?

- 3D time series data reduction
  - Reduce node count
  - 10,000x – 1,000,000x reduction over voxels
  - Pre computed connectivity
  - Allows for much faster simulation and analysis

Sparsification

Voxel

For a **16384 x 16384^3 voxel grid**, assuming 4 floats per voxel

Storage = **16384^3 × 4 floats**

≈ **17 x 10 ^ 12 floats**

(70 TB assuming a 32bit float)

Sparse Scaleable Point Cloud

Size depends on complexity, but assuming biological samples

1 million volumetric scaleable points

Storage = **1M × ~13 floats ≈ 13 million floats**

(52 MB assuming a 32bit float)

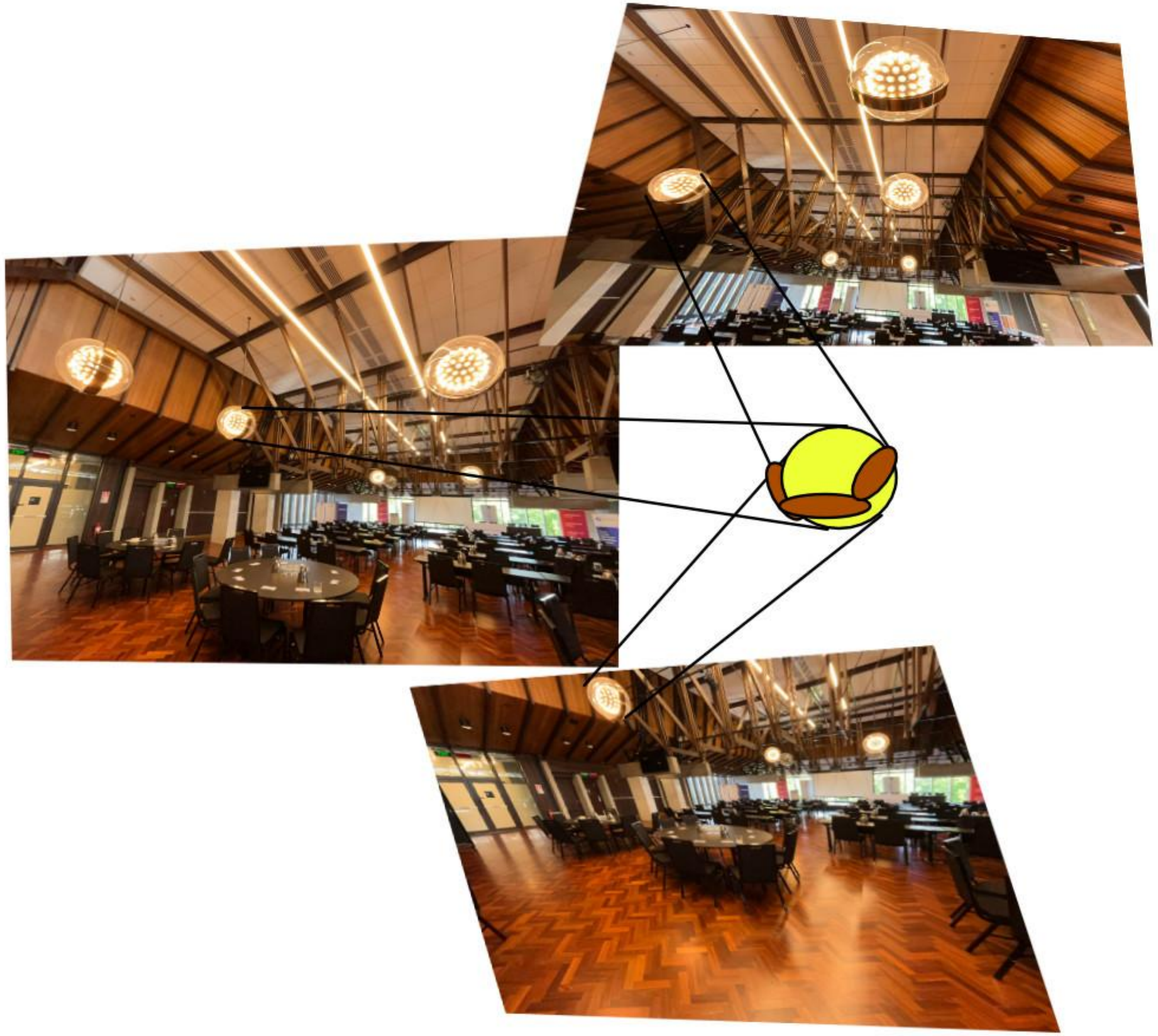
1 billion volumetric scaleable points

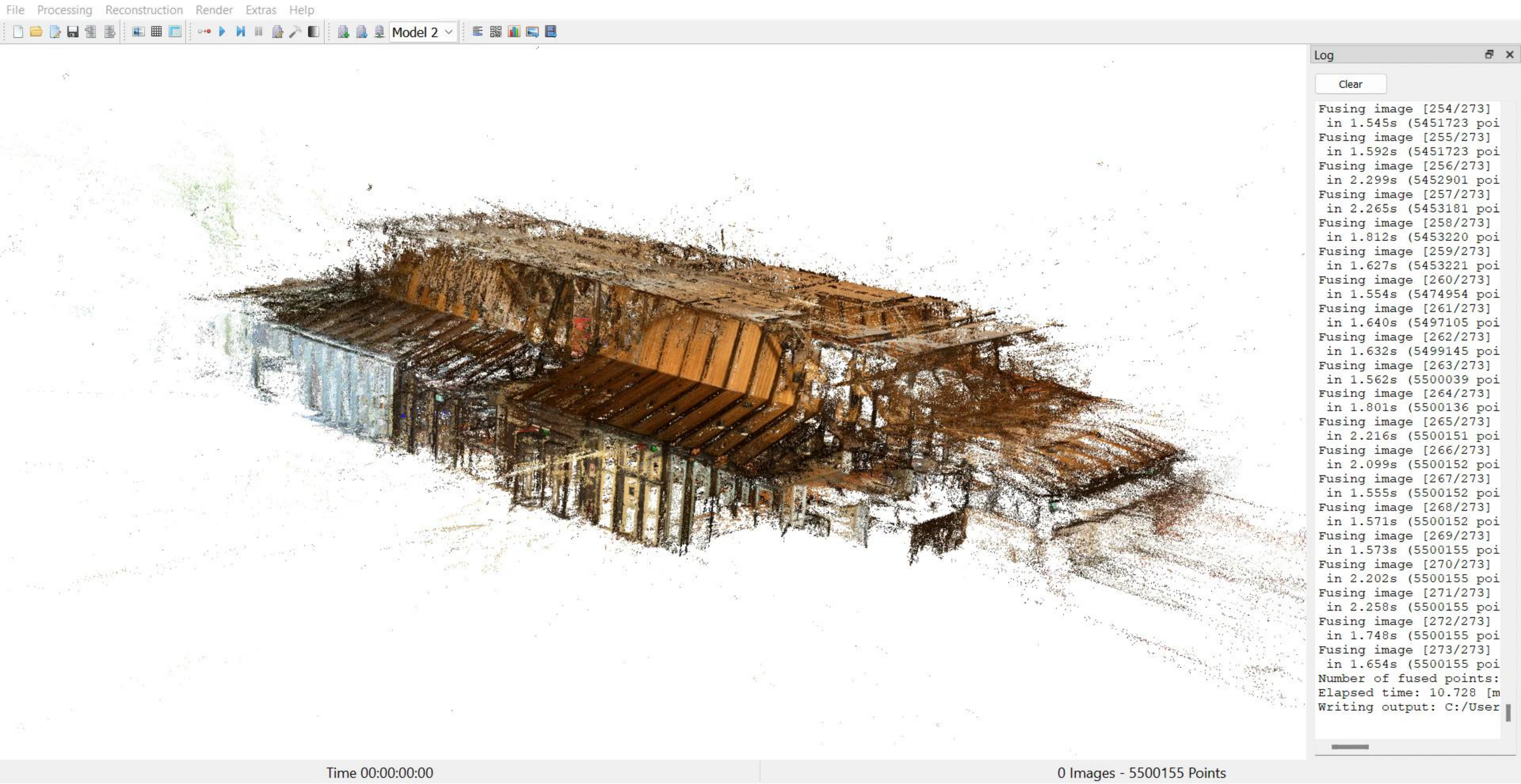
Storage = **1b × ~13 floats ≈ 13 billion floats**

(52 GB assuming a 32bit float)

Sparse Fuzzy Oriented and Scaled Blobs

- 3DGS -Three dimensional Gaussian Splatting
- Rendering is a type of ray-casting
- Generating blobs involves optimising to match the sample images from an estimated camera position

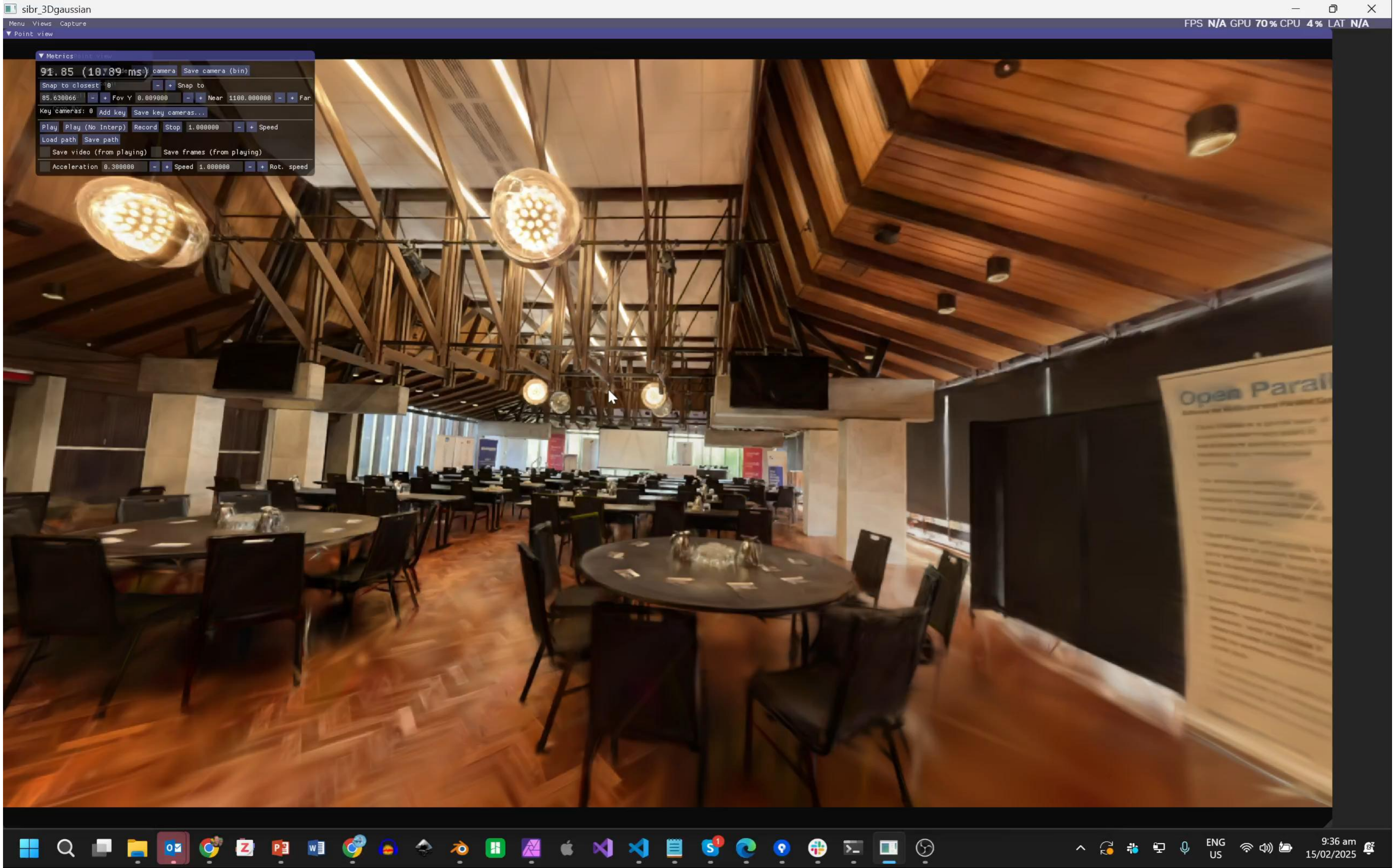




```
Fusing image [254/273]
 in 1.545s (5451723 poi
Fusing image [255/273]
 in 1.592s (5451723 poi
Fusing image [256/273]
 in 2.299s (5452901 poi
Fusing image [257/273]
 in 2.265s (5453181 poi
Fusing image [258/273]
 in 1.812s (5453220 poi
Fusing image [259/273]
 in 1.627s (5453221 poi
Fusing image [260/273]
 in 1.554s (5474954 poi
Fusing image [261/273]
 in 1.640s (5497105 poi
Fusing image [262/273]
 in 1.632s (5499145 poi
Fusing image [263/273]
 in 1.562s (5500039 poi
Fusing image [264/273]
 in 1.801s (5500136 poi
Fusing image [265/273]
 in 2.216s (5500151 poi
Fusing image [266/273]
 in 2.099s (5500152 poi
Fusing image [267/273]
 in 1.555s (5500152 poi
Fusing image [268/273]
 in 1.571s (5500152 poi
Fusing image [269/273]
 in 1.573s (5500155 poi
Fusing image [270/273]
 in 2.202s (5500155 poi
Fusing image [271/273]
 in 2.258s (5500155 poi
Fusing image [272/273]
 in 1.748s (5500155 poi
Fusing image [273/273]
 in 1.654s (5500155 poi
Number of fused points:
Elapsed time: 10.728 [m
Writing output: C:/User
```

Time 00:00:00:00

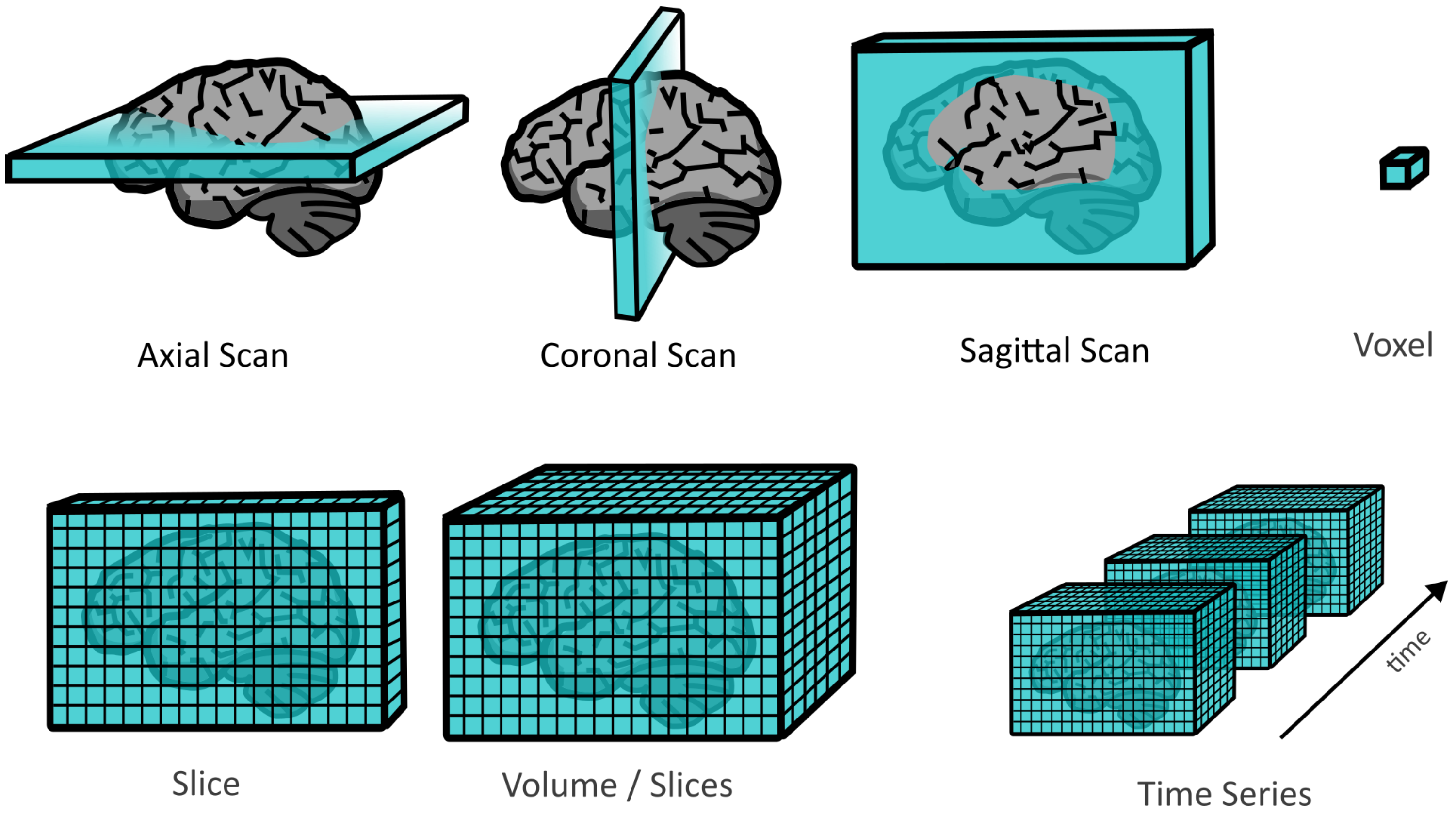
0 Images - 5500155 Points



```
91.85 (10.89 ms)   camera   Save camera (bin)
Snap to closest  0   -  +  Snap to
85.630066  -  +  Fov Y  0.009000  -  +  Near  1100.000000  -  +  Far
Key cameras: 0  Add key  Save key cameras...
Play  Play (No Interp)  Record  Stop  1.000000  -  +  Speed
Load path  Save path
□ Save video (from playing)  □ Save frames (from playing)
□ Acceleration  0.300000  -  +  Speed  1.000000  -  +  Rot. speed
```

Sparse Fuzzy Oriented and Scaled Blobs

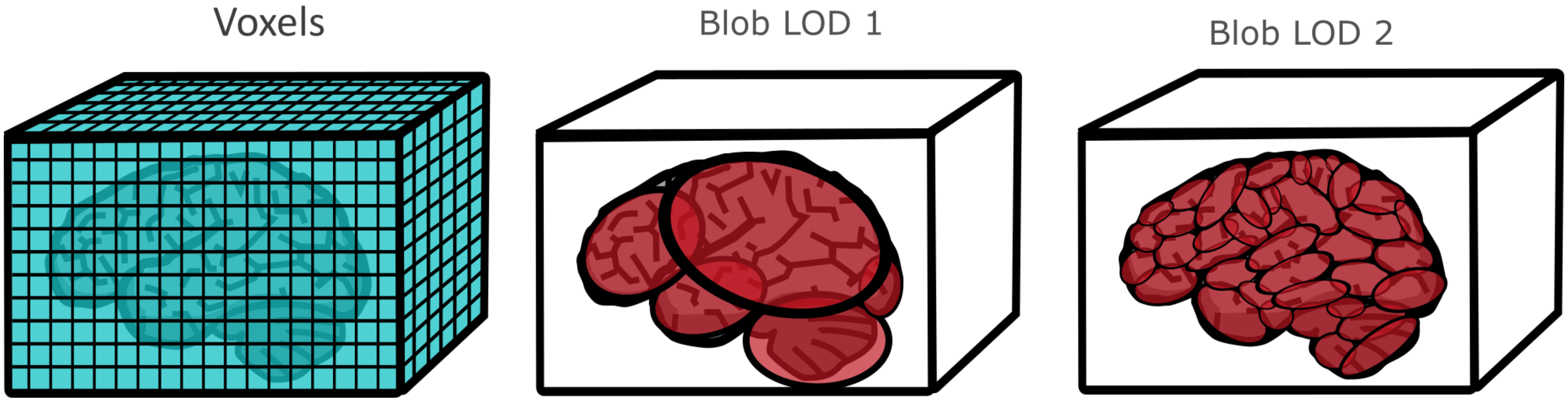
- Known 'camera position'
- Volumes not surfaces
- Voxel Homogeneity
- Voxel Variation Frequency



Conceptual MRI Data

Sparse Fuzzy Oriented and Scaled Blobs

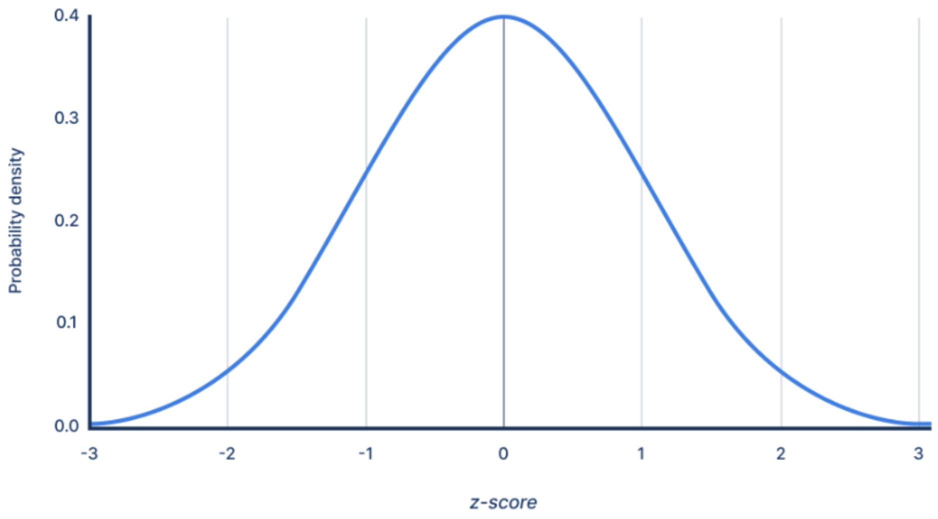
- Voxels converted into sparse, fuzzy, oriented and scaled blobs that represent the voxel values in 3D shape not just surface
- Different levels of detail of fidelity are able to be precomputed



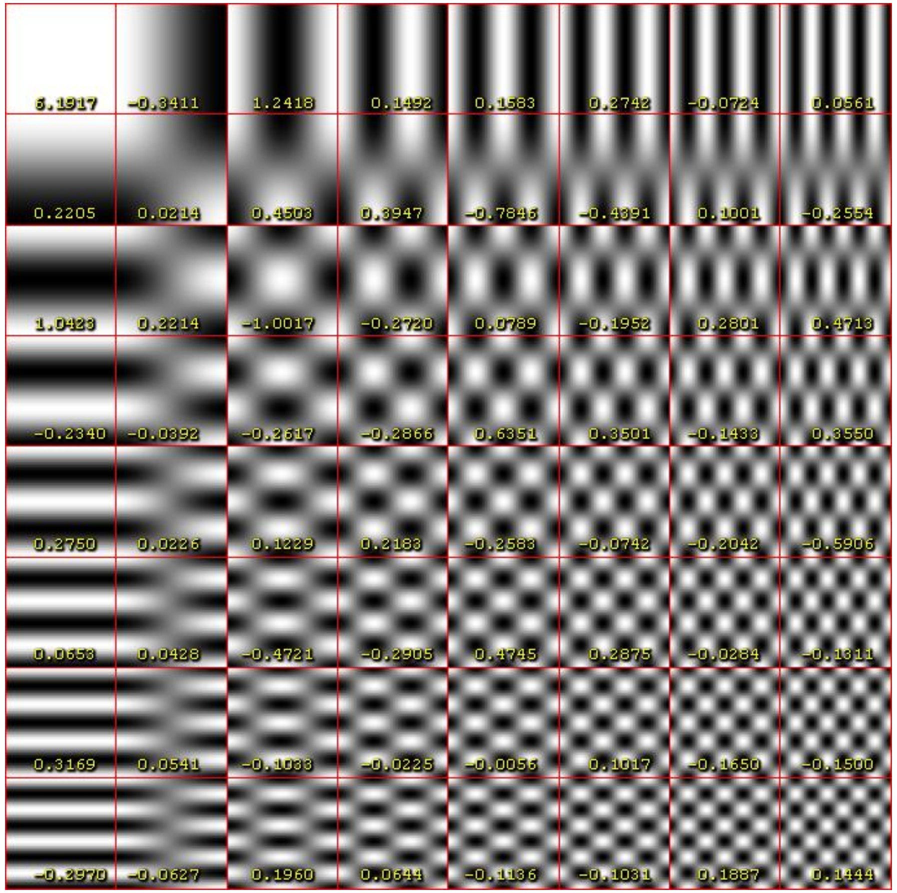
Adding more complexity to blobs

- DCT is the basis for JPG and VarDCT for JpegXL – we have added DCT rather than just a ‘normal’ Gaussian

Standard normal distribution

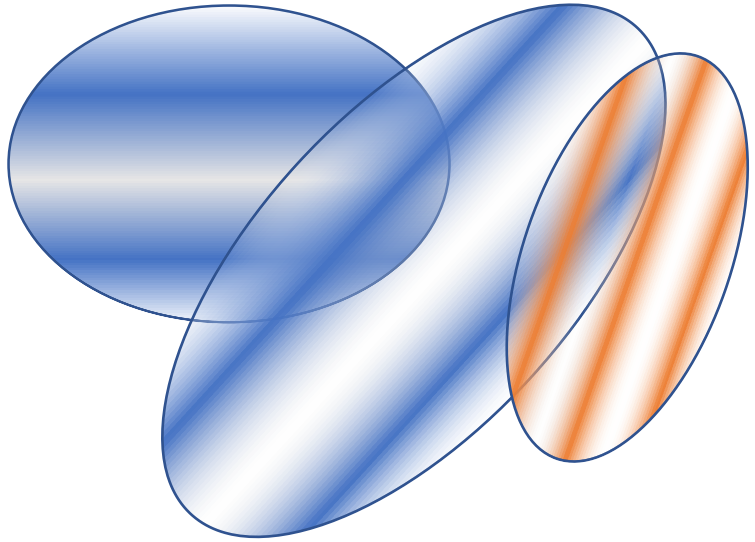


Discrete Cosine Transform



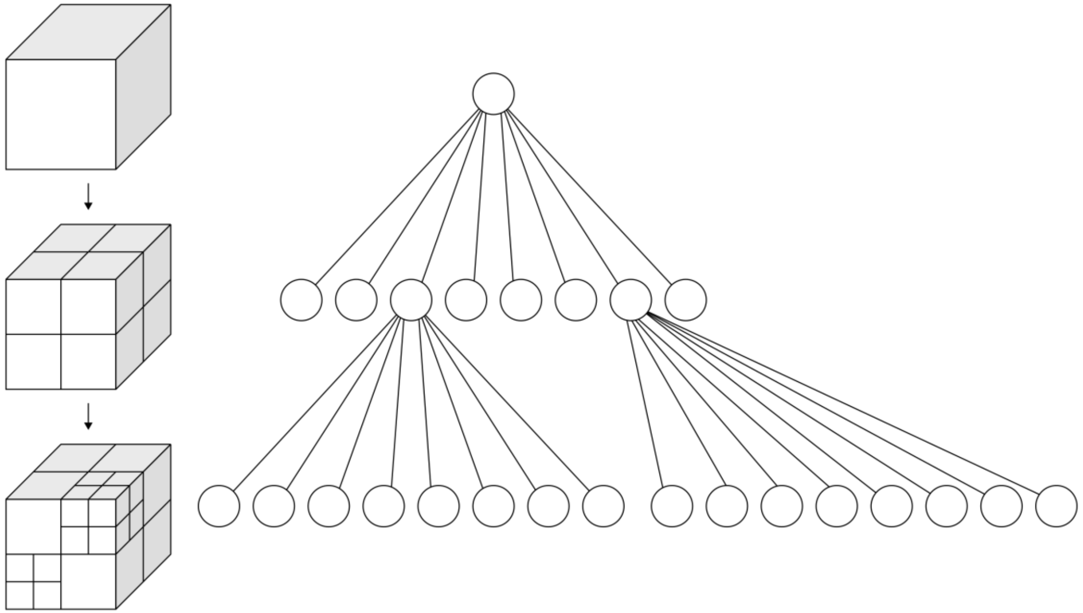
Adding more complexity to blobs

- Great compression – but requires convolving the blobs together for the final result so this is less useful for simulation where you would like your nodes to be independent



Managing variable complexity with spatial partitioning

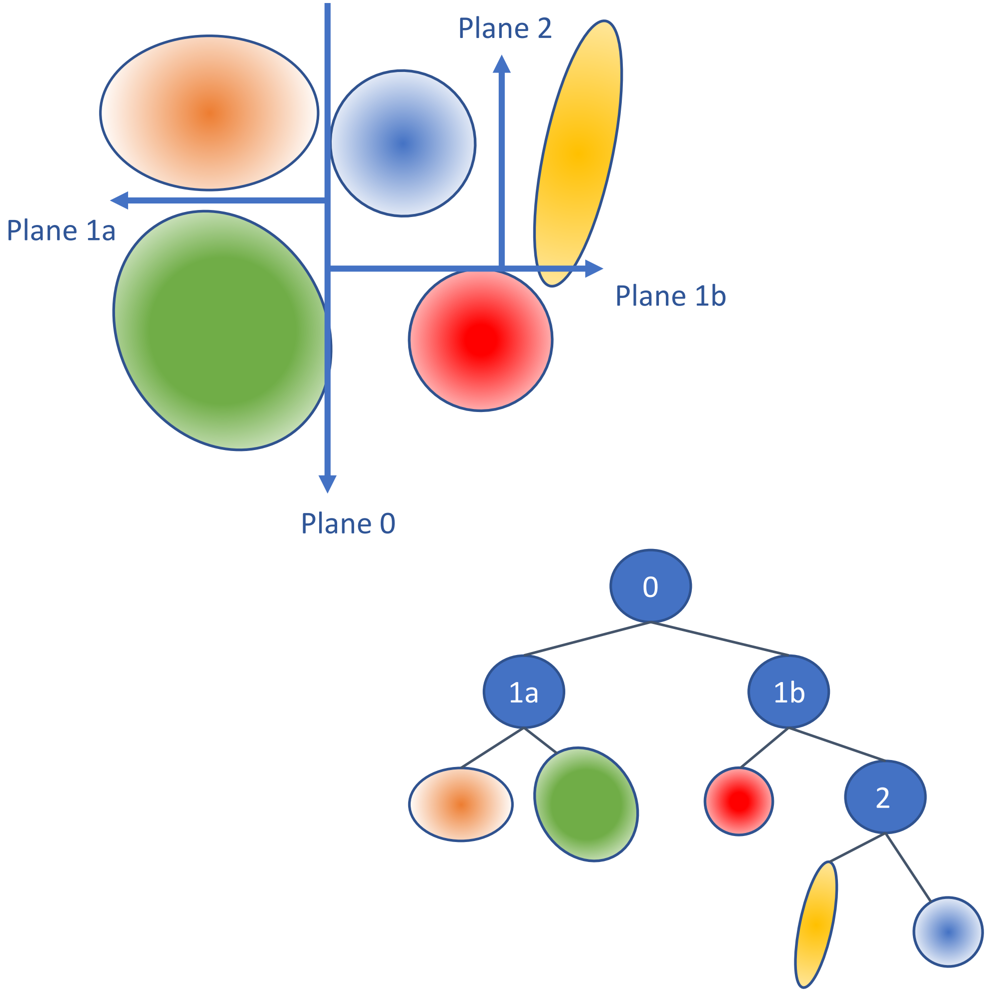
- OctTree allows for variable resolution and storing multiple levels of detail
- Deltas calculated per cube in the octree – allows for time series data to be variably spatially run length encoded for unchanging regions
- Spatial partitioning allows for parts of the data set to be processed in parallel



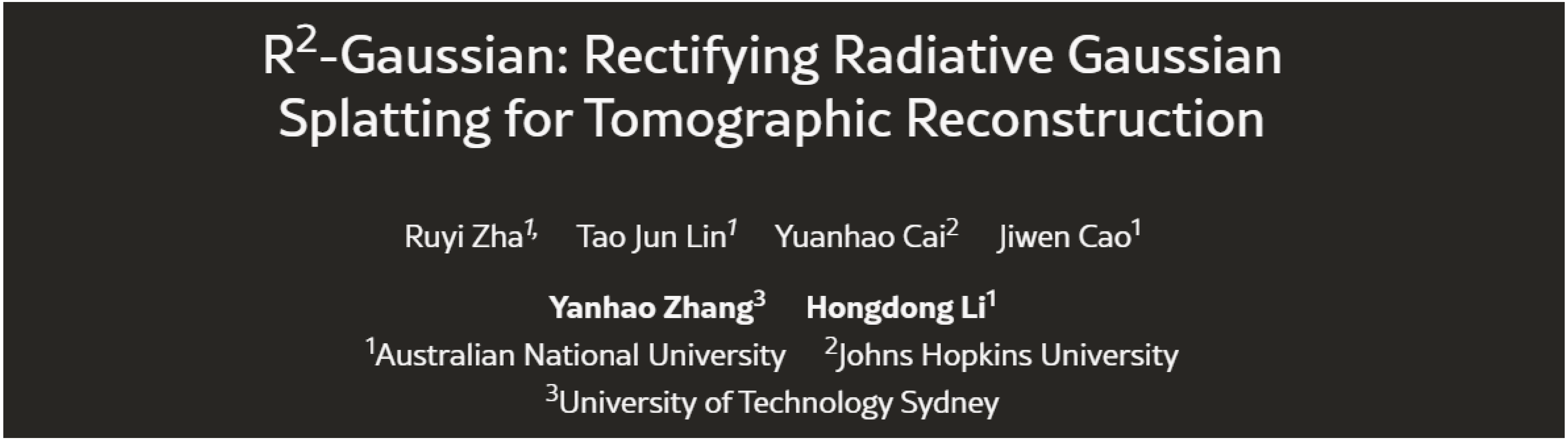
Simulation & processing using trees

- KD Trees for homogenous nodes
- KD Trees for connected nodes

- Replace per voxel style marching algorithms with significantly faster approaches using 10,000x 1,000,000x fewer nodes with precomputed connectivity
- Connectivity of different classifications (bone, tissue, material properties)
- Connectivity of variable density



Volume not surface optimisation



R^2-Gaussian: Rectifying Radiative Gaussian Splatting for Tomographic Reconstruction

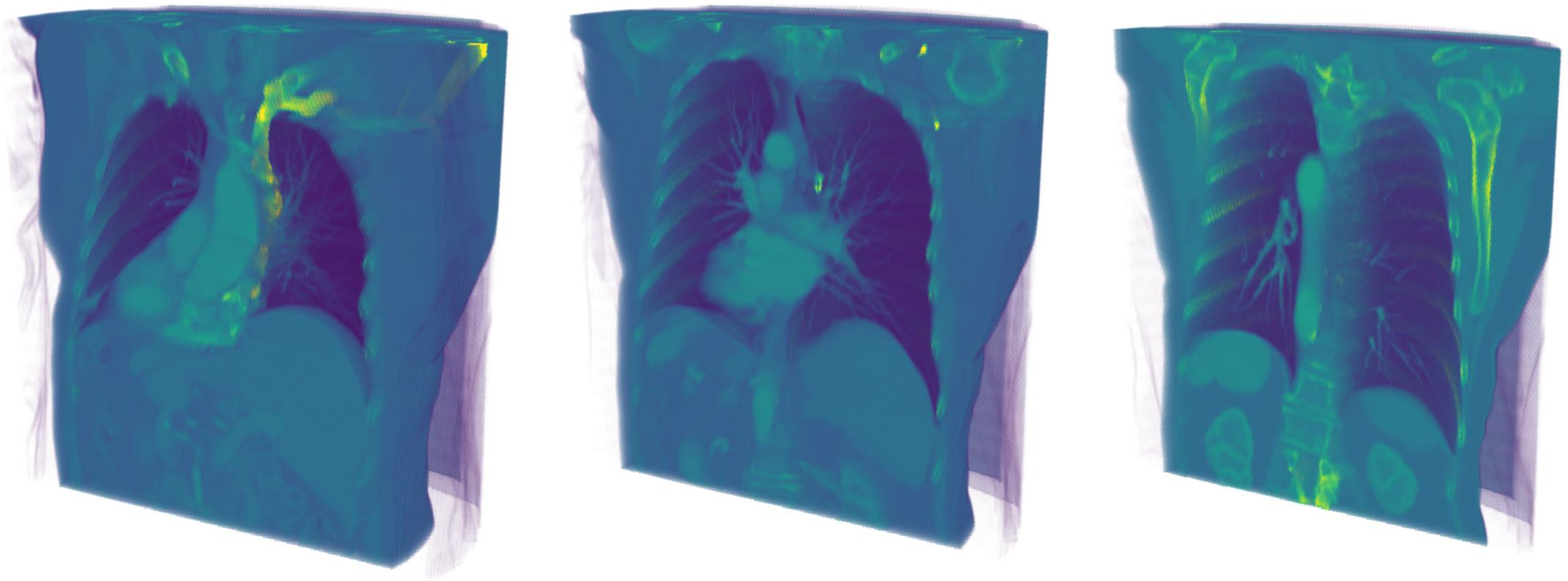
Ruyi Zha[1,] Tao Jun Lin[1] Yuanhao Cai[2] Jiwen Cao[1]

Yanhao Zhang[3] **Hongdong Li[1]**

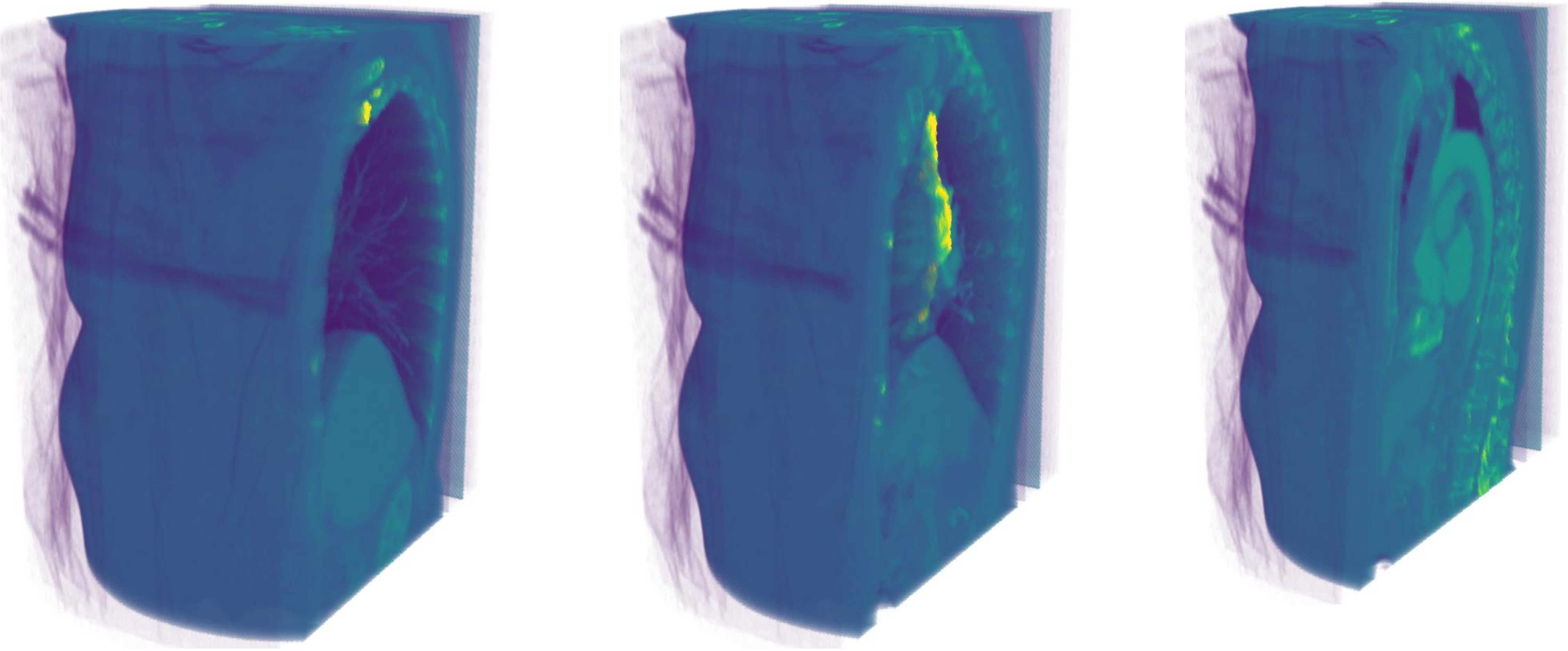
[1]Australian National University [2]Johns Hopkins University

[3]University of Technology Sydney

Volume not surface optimisation



Volume not surface optimisation



Nanoscanner