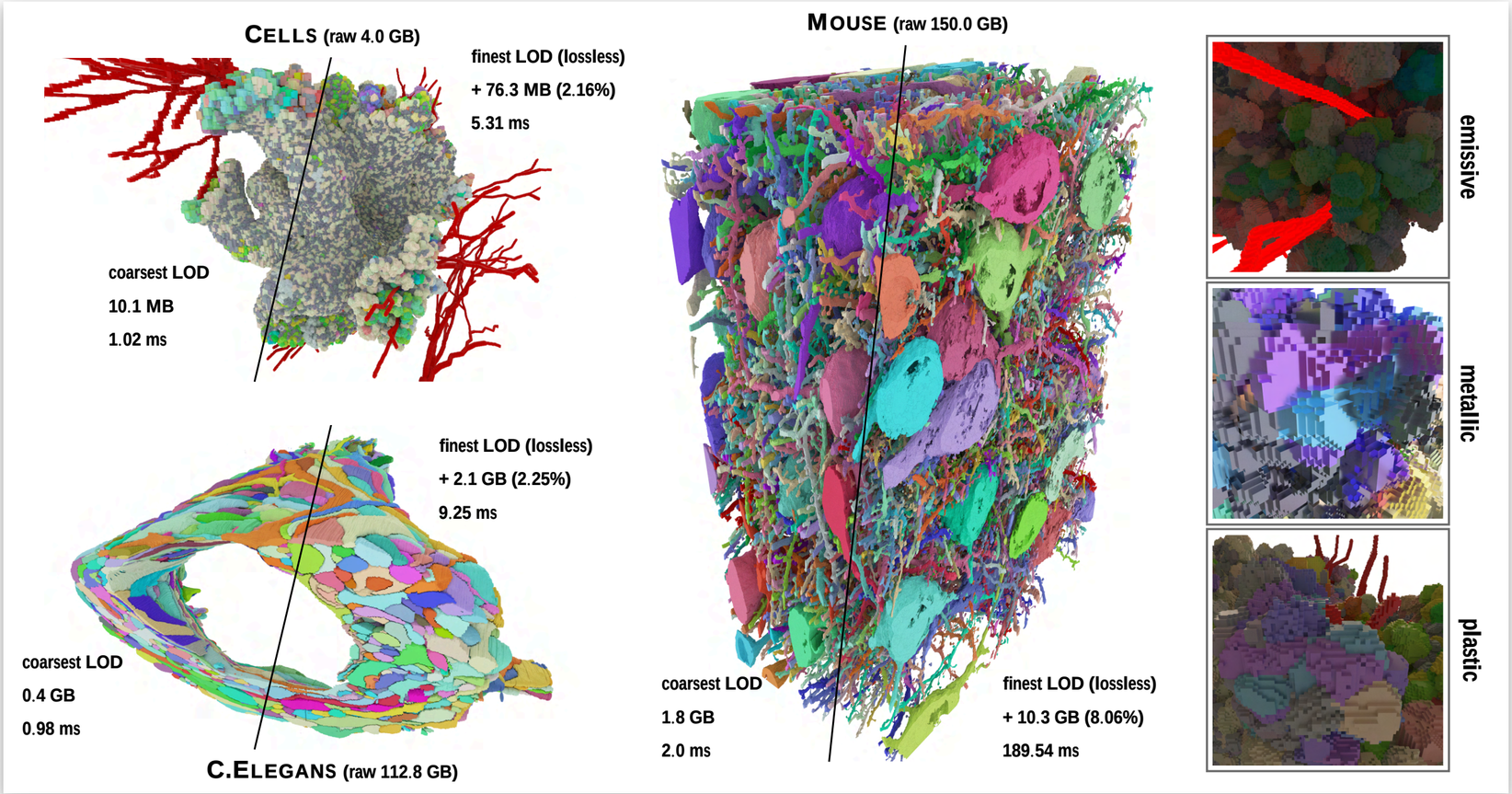


SVDAG Compression for Segmentation Volume Path Tracing

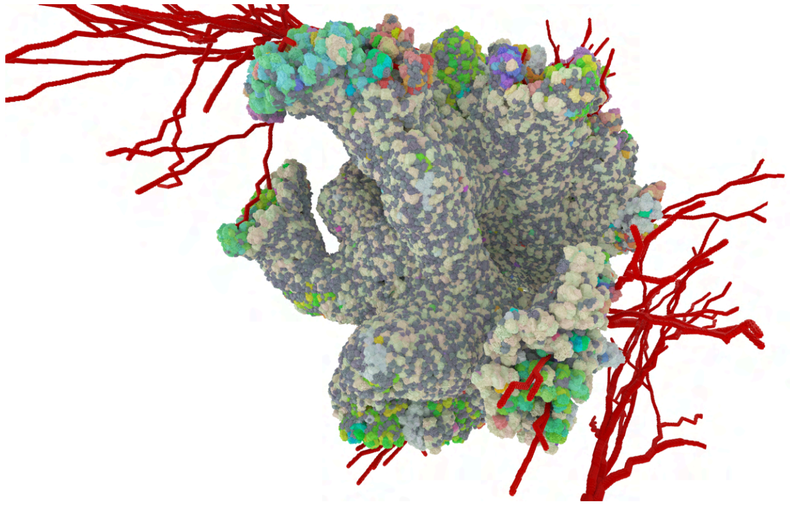
Mirco Werner†, Max Piochowiak†, and Carsten Dachsbacher

(† joint first authors)

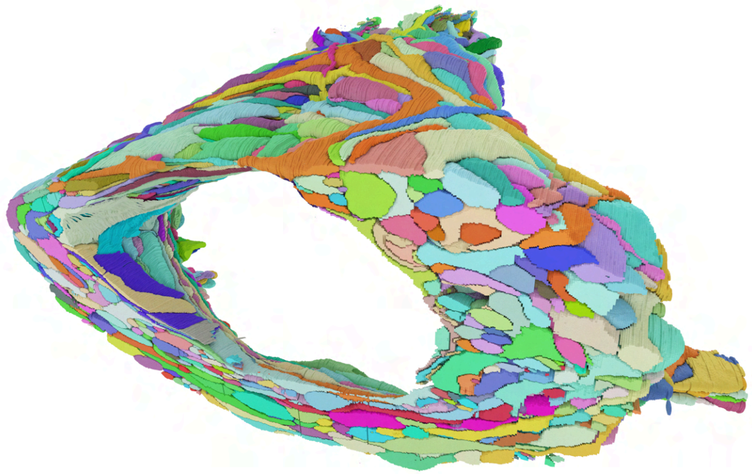
Karlsruhe Institute of Technology

Segmentation Volumes

CELLS (Rosenbauer et al. 2020)



C.ELEGANS (Witvliet et al. 2021)

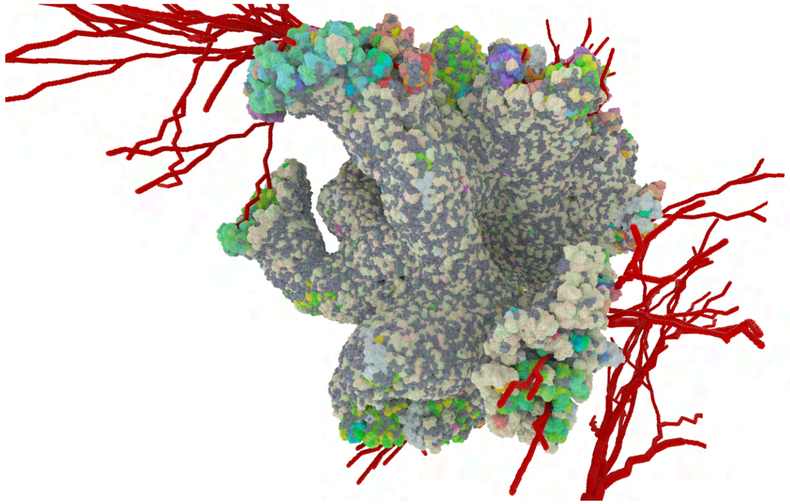


MOUSE (Motta et al. 2019)

Segmentation Volume: $V(x) = l \in \mathbb{N}$

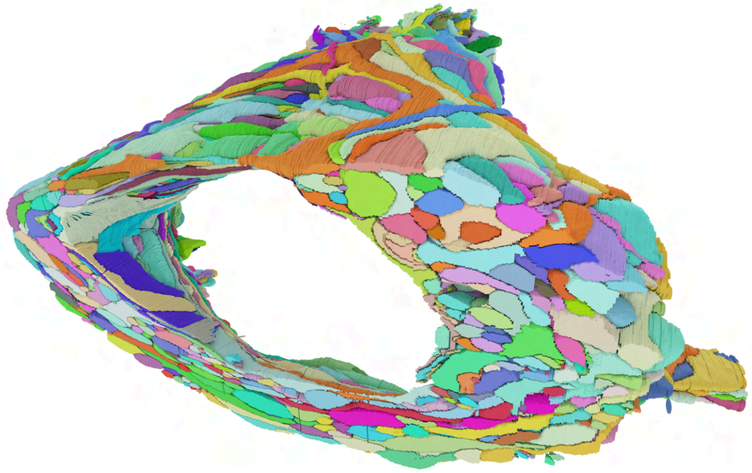
Segmentation Volumes

CELLS (Rosenbauer et al. 2020)



4 GB (32 bit/voxel)

C.ELEGANS (Witvliet et al. 2021)



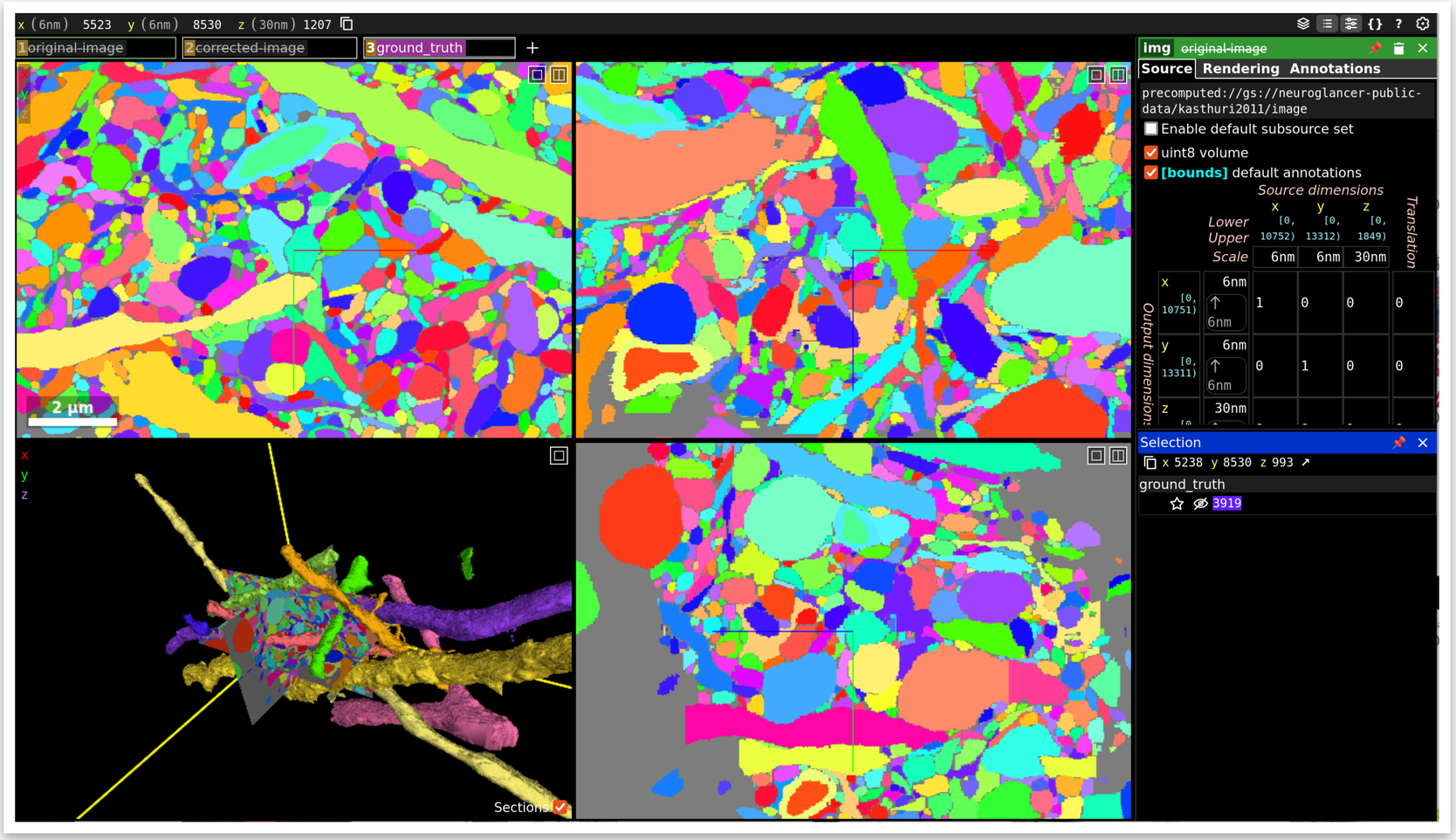
113 GB (8 bit/voxel)

MOUSE (Motta et al. 2019)

150 GB (8 bit/voxel)

Segmentation Volume: $V(x) = l \in \mathbb{N}$

Typical Rendering Workflow

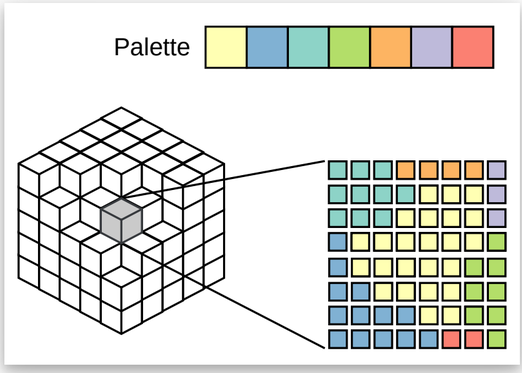


Google Neuroglancer

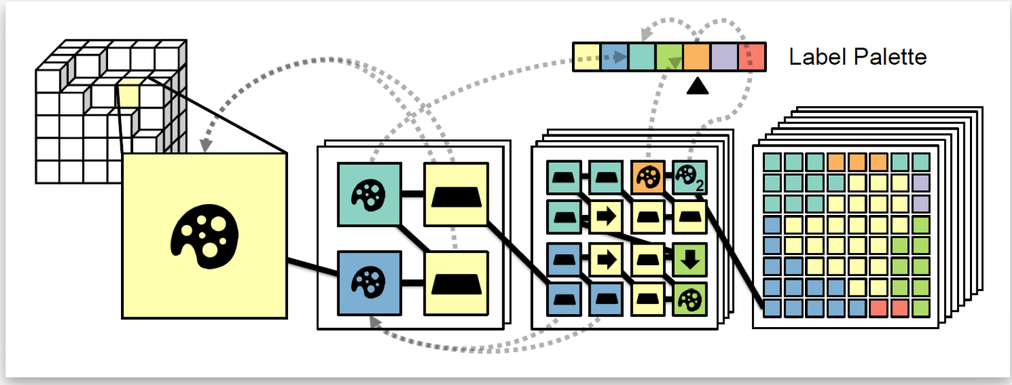
Typical Rendering Workflow

Brick-Wise Compression

- stores voxels in bricks
- label palette per brick
- $\lceil log_2(\|palette\|) \rceil$ bits per voxel

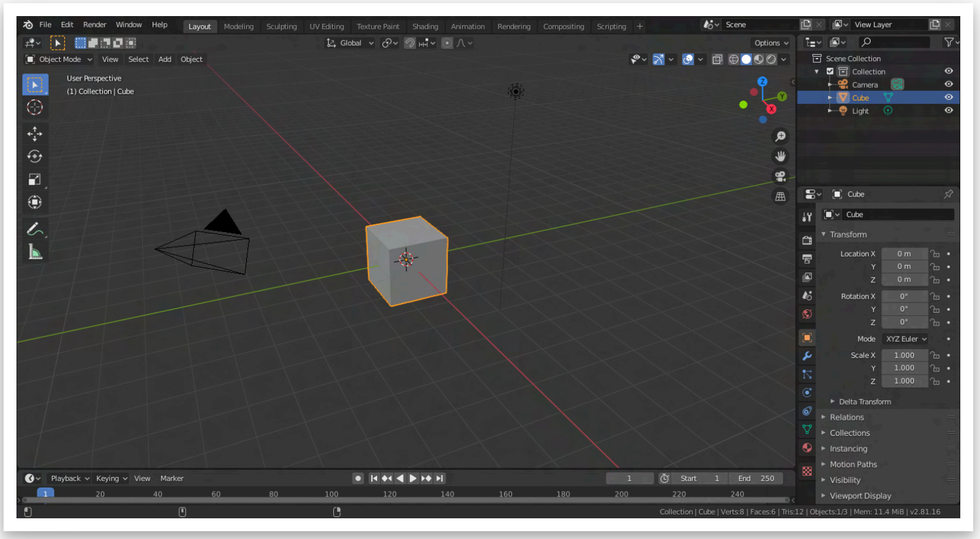


Neuroglancer Precomputed Format

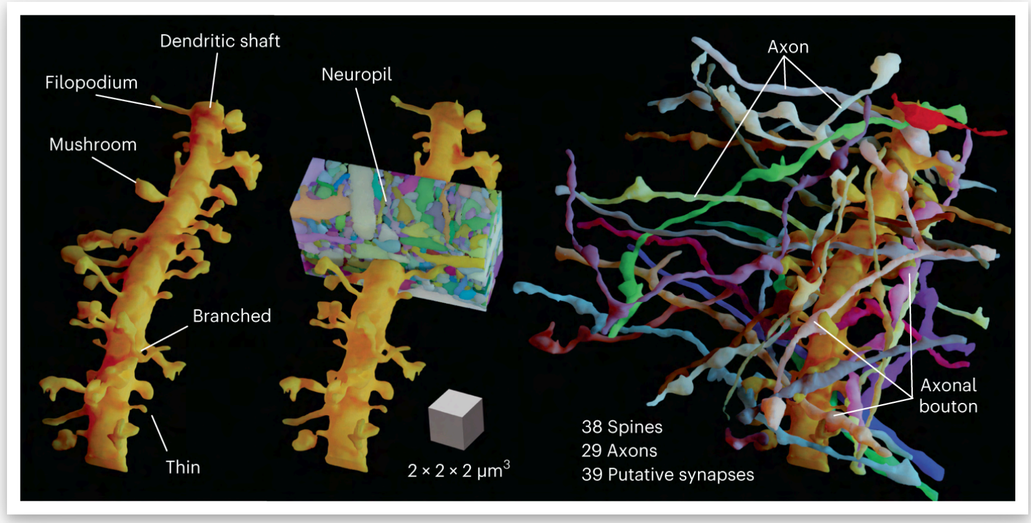


[Piochowiak*2023]
Fast Compressed Segmentation Volumes for Scientific Visualization, IEEE Vis

Typical Rendering Workflow

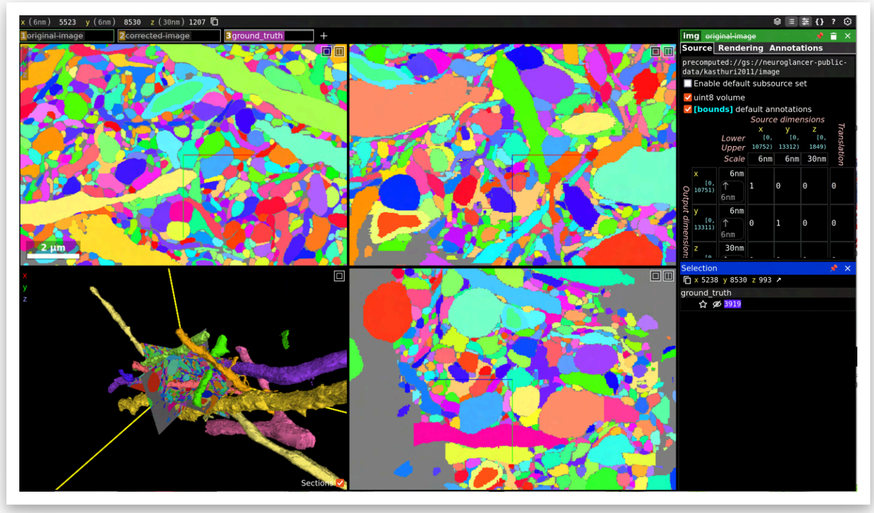


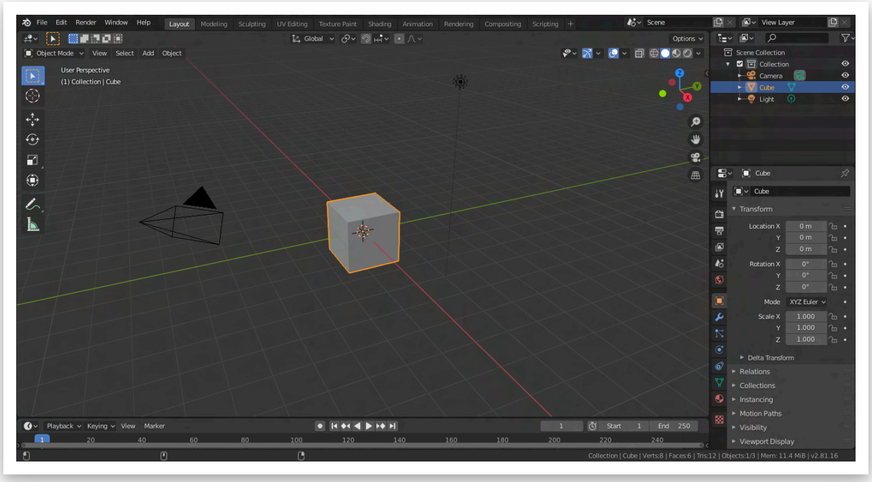
Blender manual CC-BY-SA 4.0

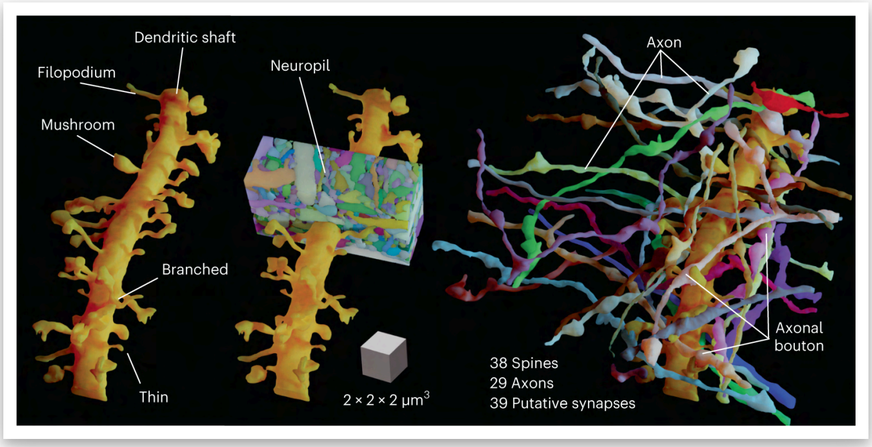


[Velicky*2023]
Dense 4D nanoscale reconstruction of living brain tissue, Nature methods

Typical Rendering Workflow

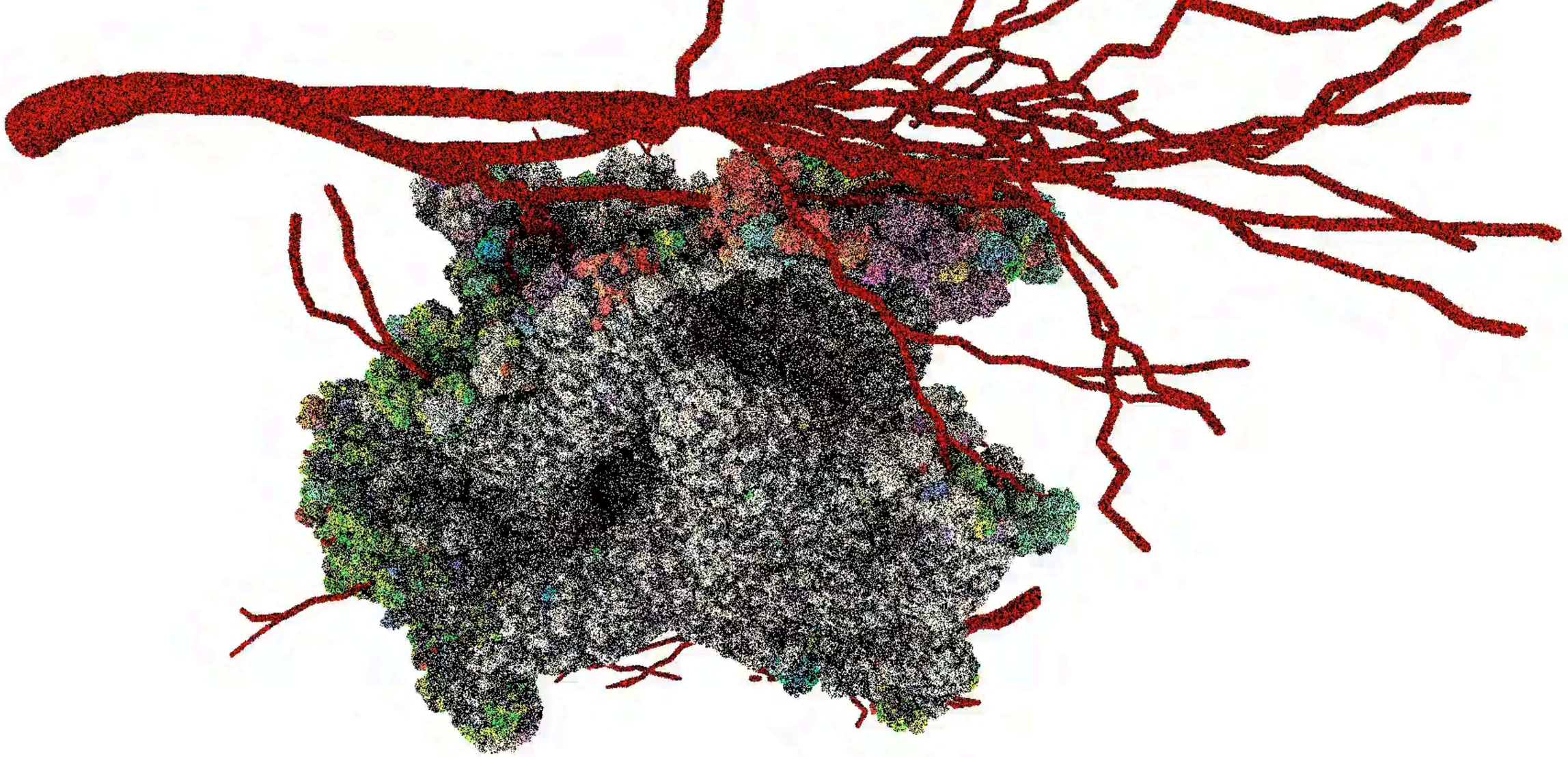






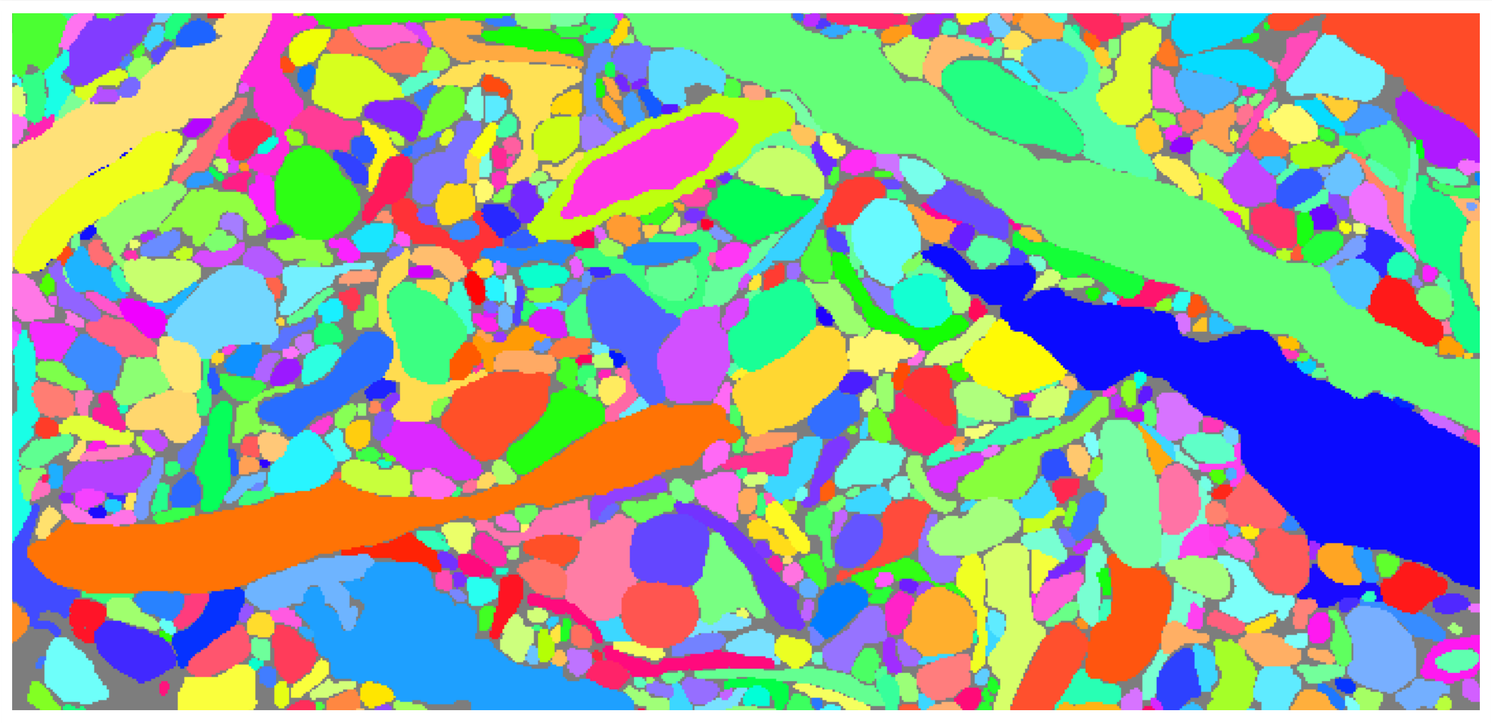
Compression for Segmentation Volumes

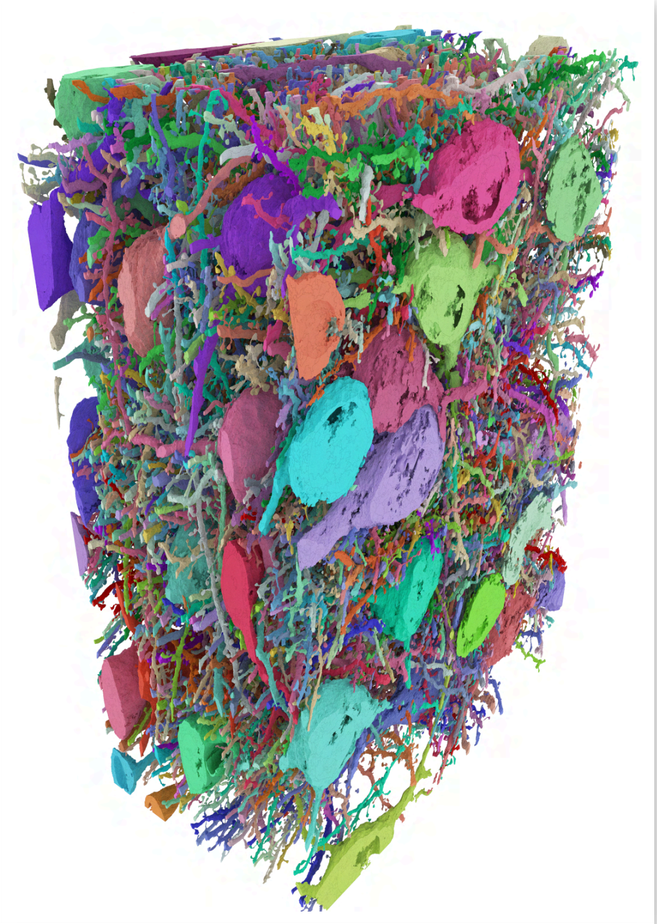
(our data structure)



Compression for Segmentation Volumes - Observations

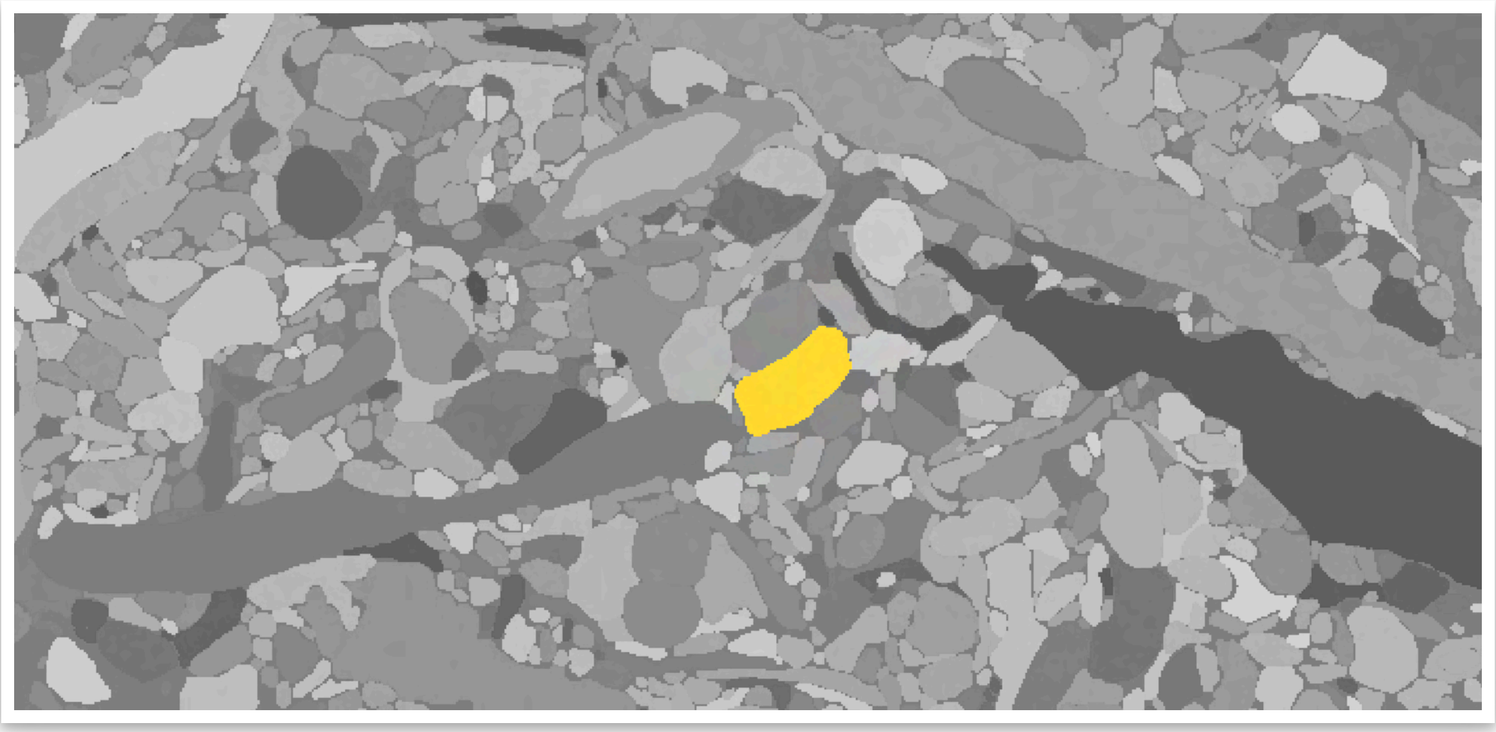
- key observation: large uniform areas with the same label

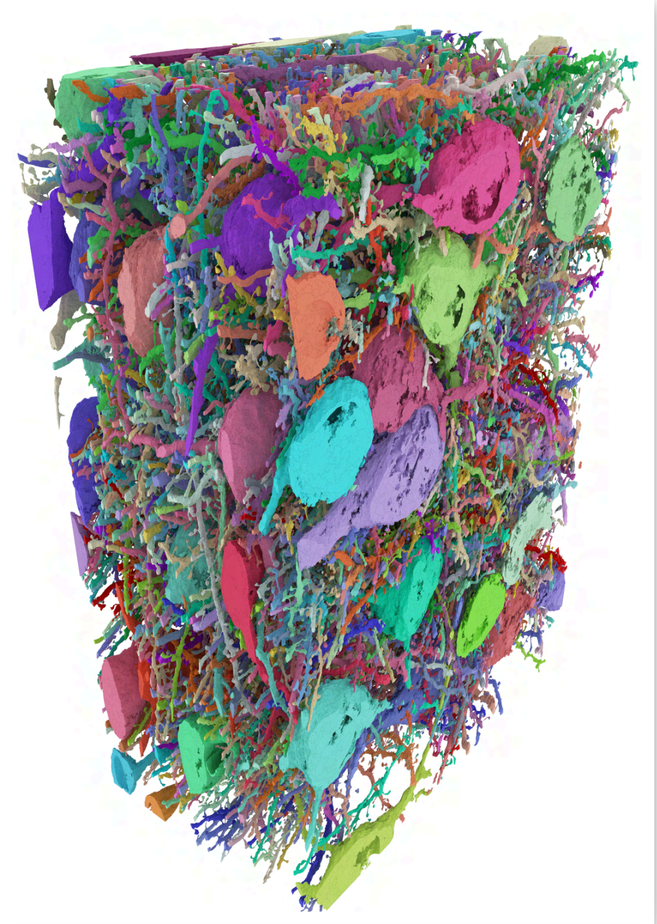




Compression for Segmentation Volumes - Observations

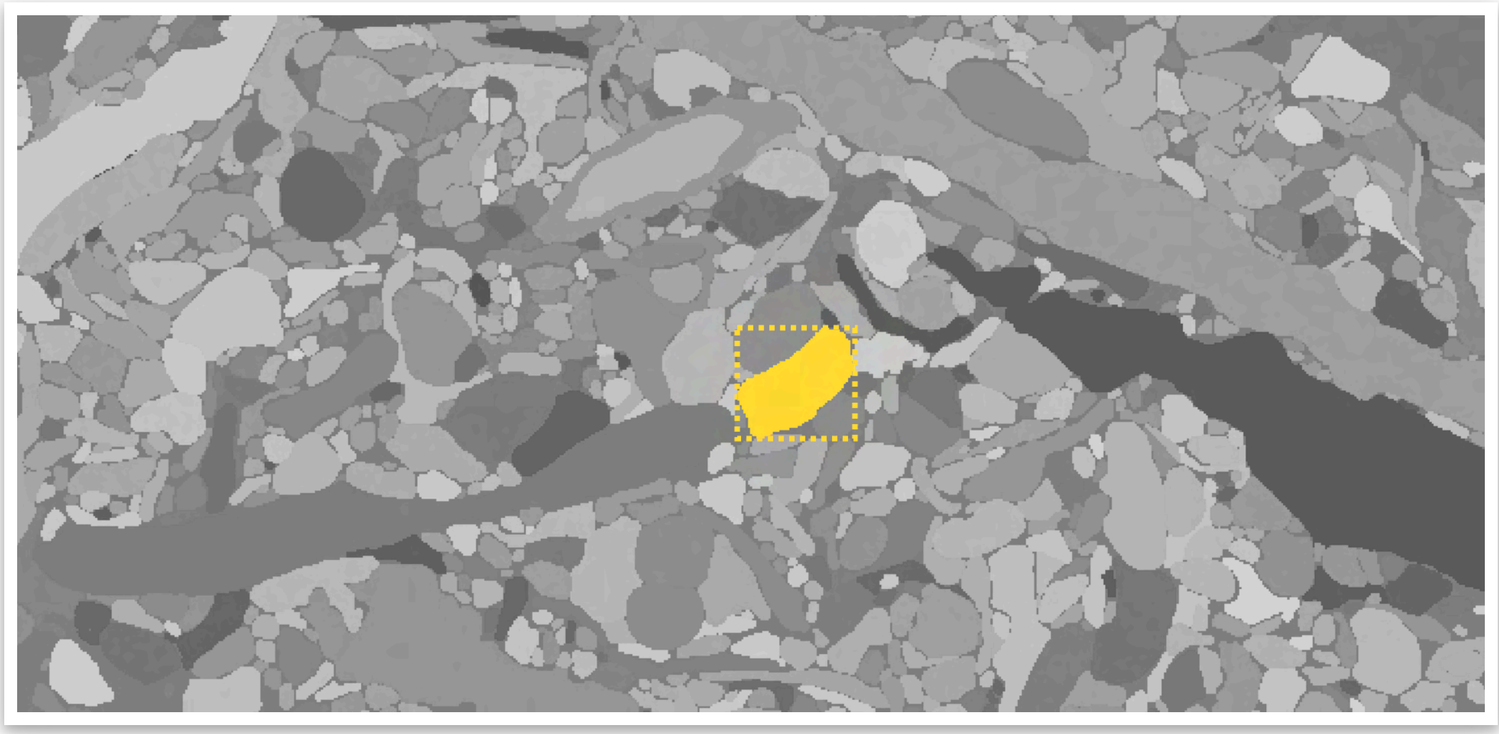
- key observation: large uniform areas with the same label
- compression idea: decouple label and voxel occupancy





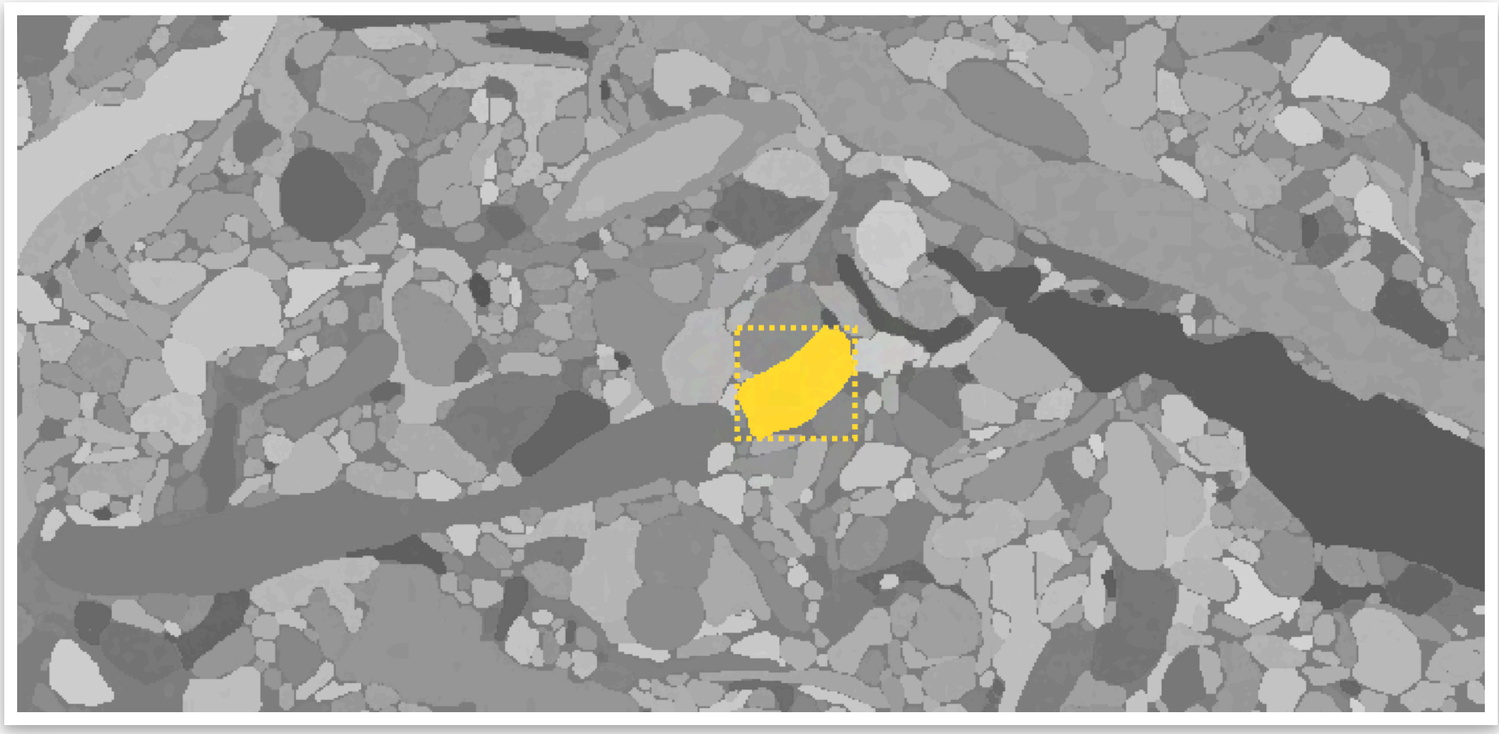
Compression for Segmentation Volumes - Observations

- key observation: large uniform areas with the same label
- compression idea: decouple label and voxel occupancy
  - AABB per label region



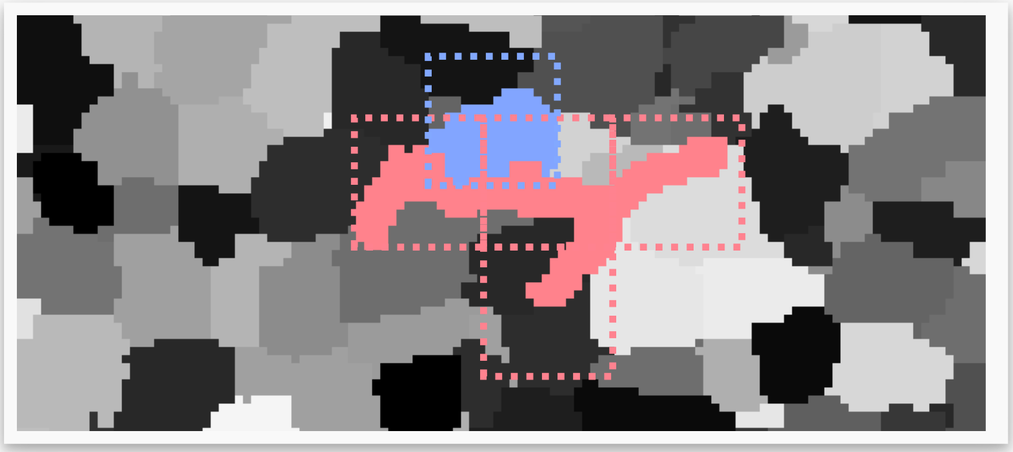
Compression for Segmentation Volumes - Observations

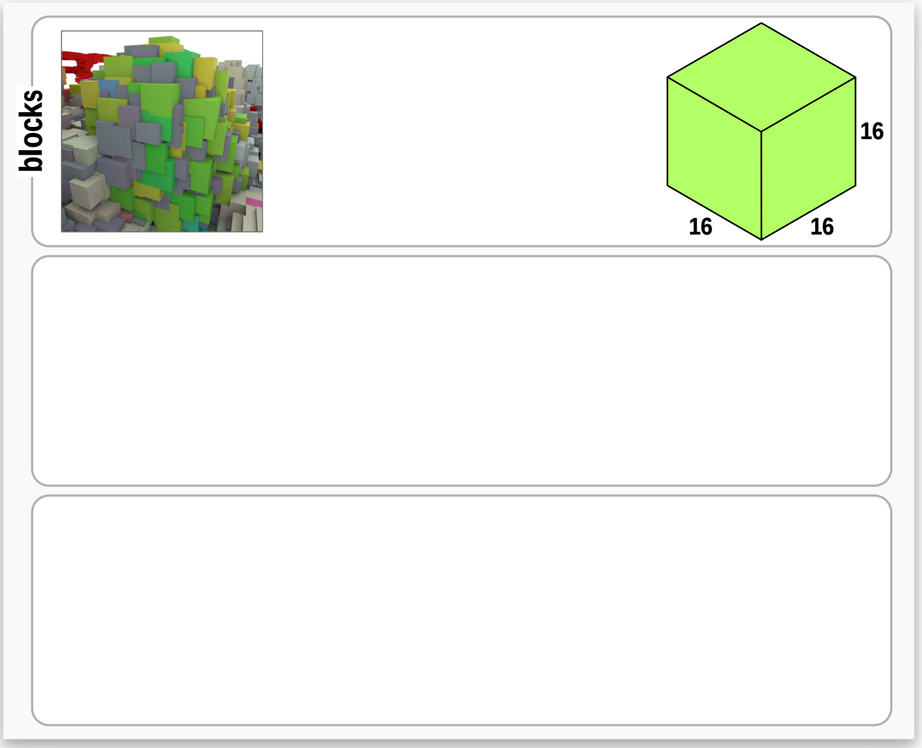
- key observation: large uniform areas with the same label
- compression idea: decouple label and voxel occupancy
  - AABB per label region
  - voxel occupancy becomes binary attribute inside region



Data Structure - Level 1 (blocks) ➡ decouple label from voxel occupancy

- split AABBs into blocks of fixed size (16^3)

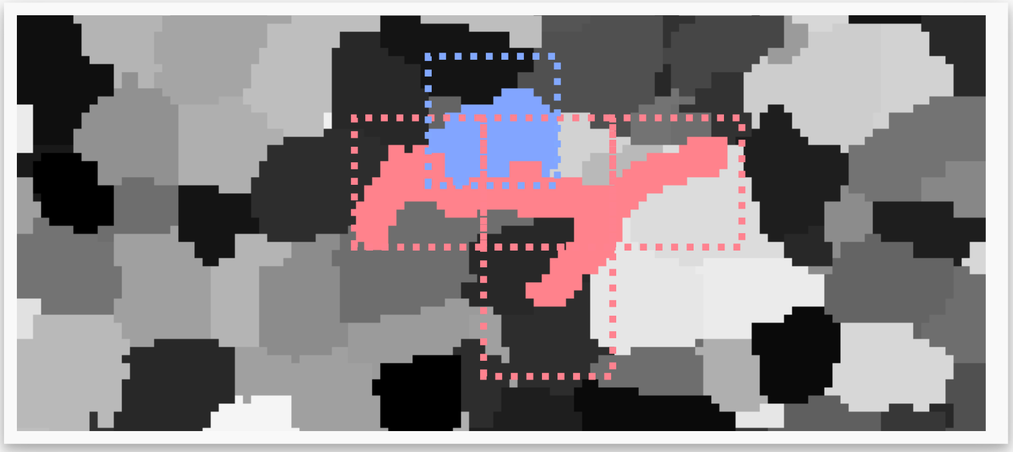


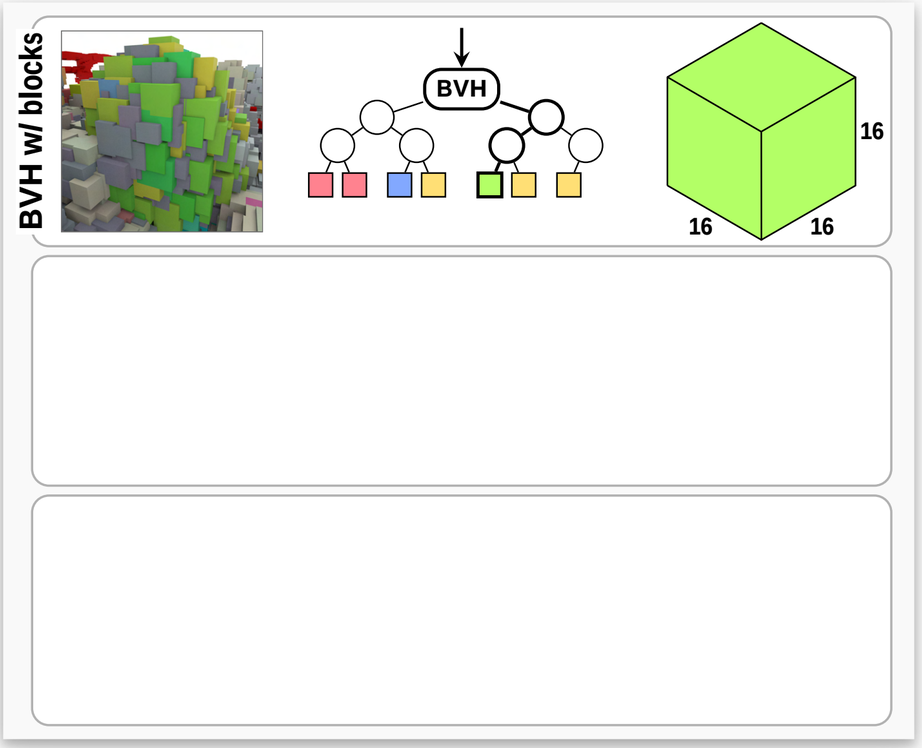


Data Structure - Level 1 (BVH w/ blocks)

➡ decouple label from voxel occupancy

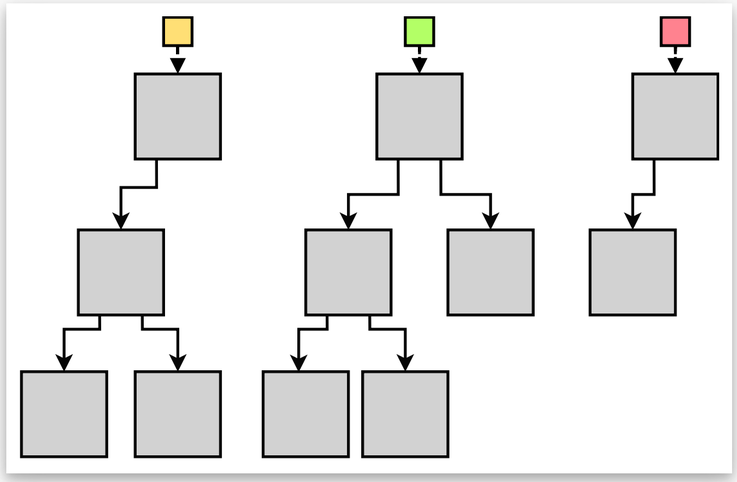
- split AABBs into blocks of fixed size (16^3)
- BVH for hardware-accelerated raytracing
  - blocks = BVH leaves, store label information

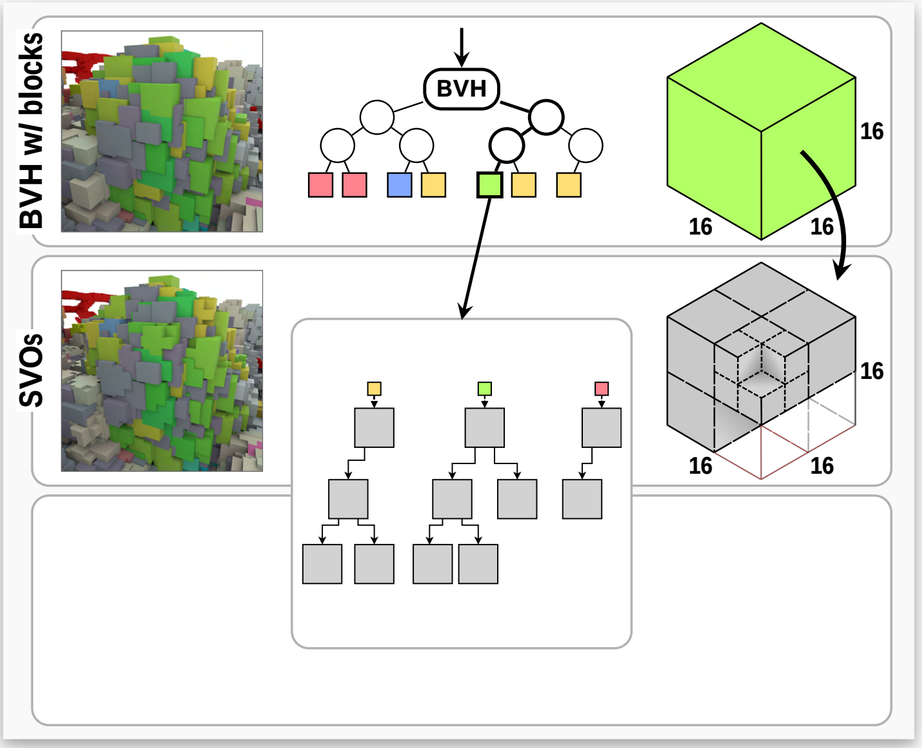




Data Structure - Level 2 (SVOs) ➡ encode voxel occupancy

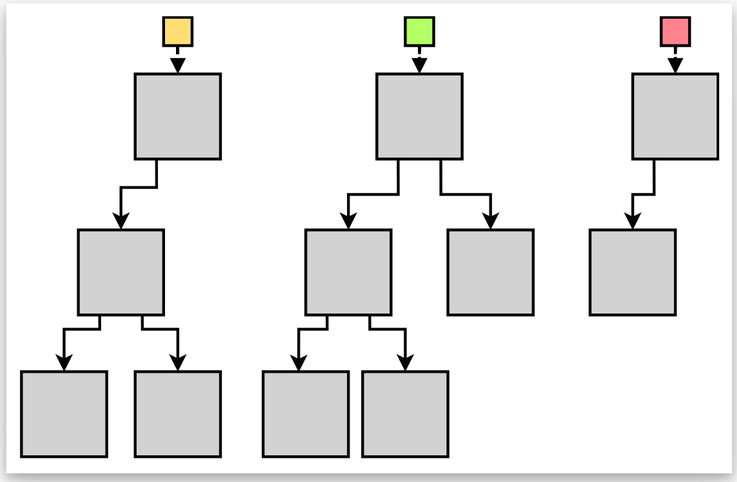
- sparse voxel octree (SVO) [Laine*2010] in each block

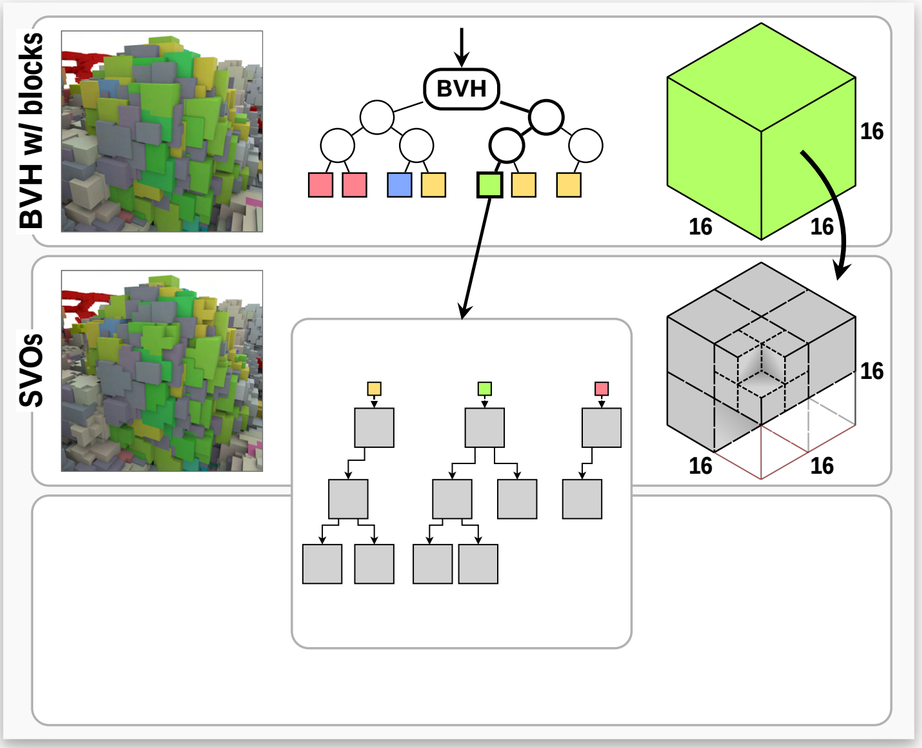




Data Structure - Level 2 (SVOs) ➡ encode voxel occupancy

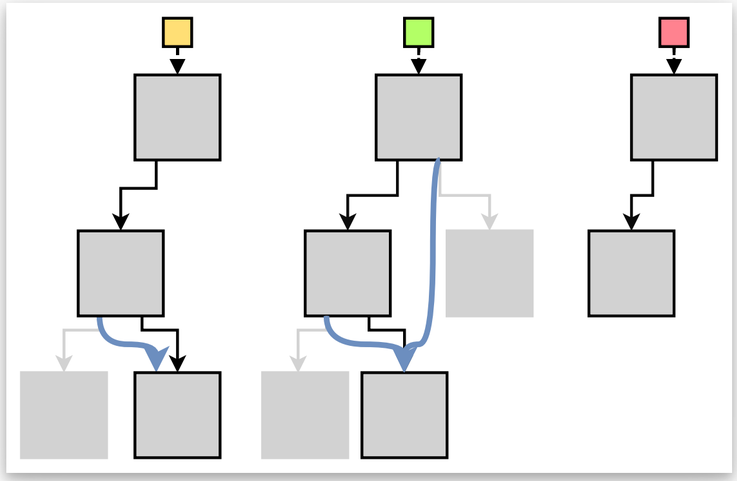
- sparse voxel octree (SVO) [Laine*2010] in each block
  - identical subtrees

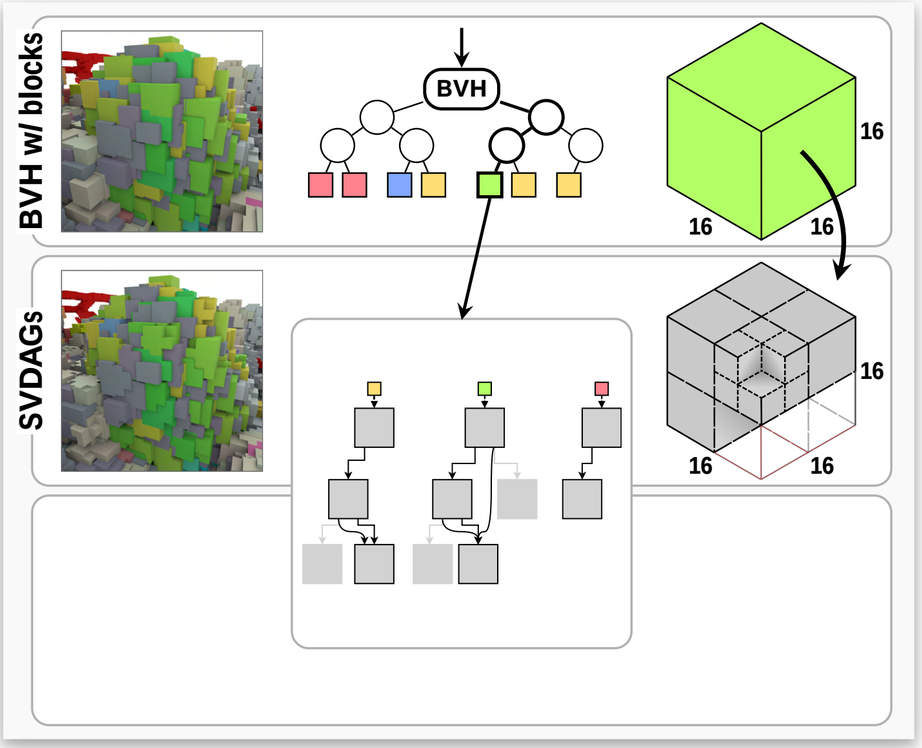




Data Structure - Level 2 (SVDAGs) ➡ encode voxel occupancy

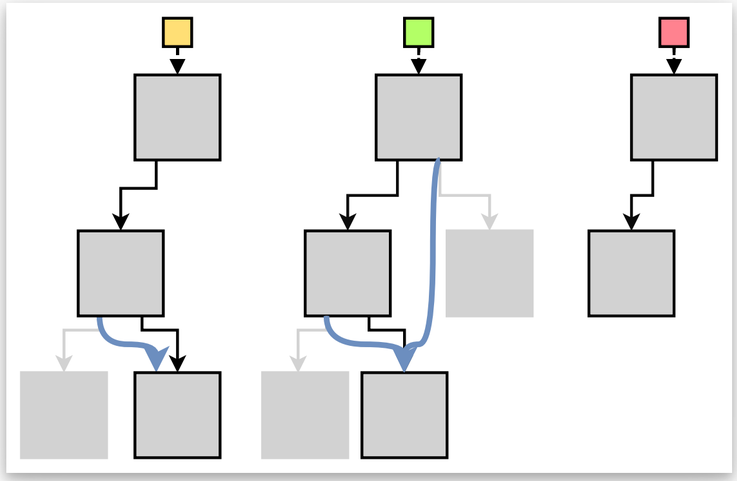
- sparse voxel directed acyclic graph (SVDAG) [Kämpe*2013] in each block

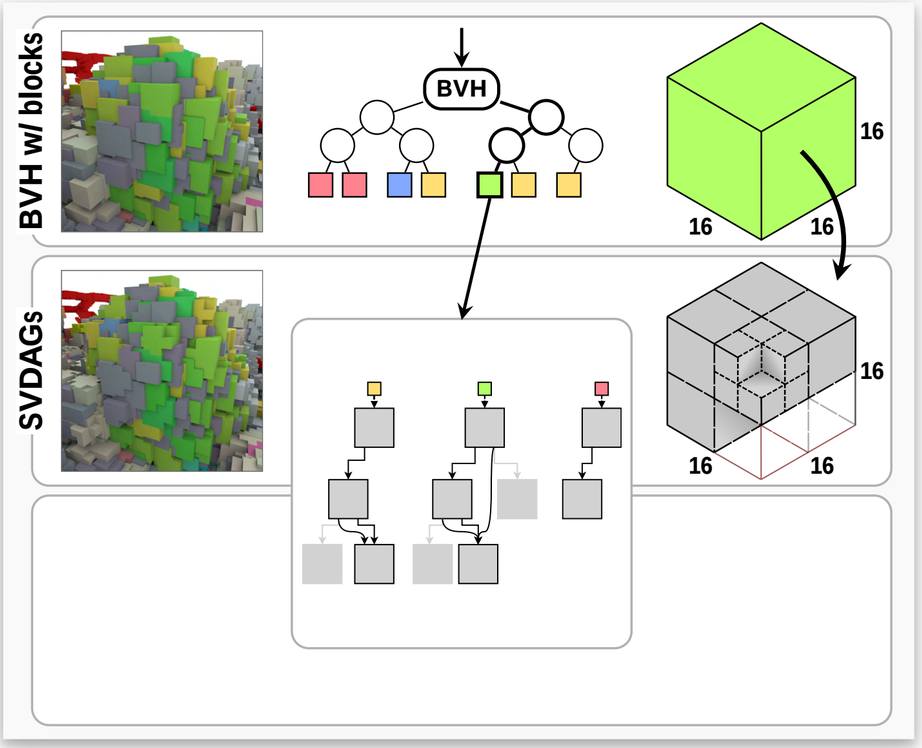




Data Structure - Level 2 (SVDAGs) ➡ encode voxel occupancy

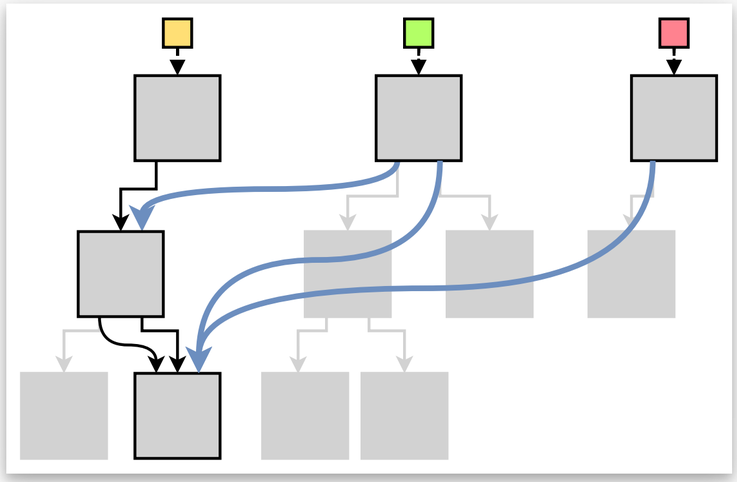
- sparse voxel directed acyclic graph (SVDAG) [Kämpe*2013] in each block
  - identical subtrees across blocks with different labels
  - (no label information stored)

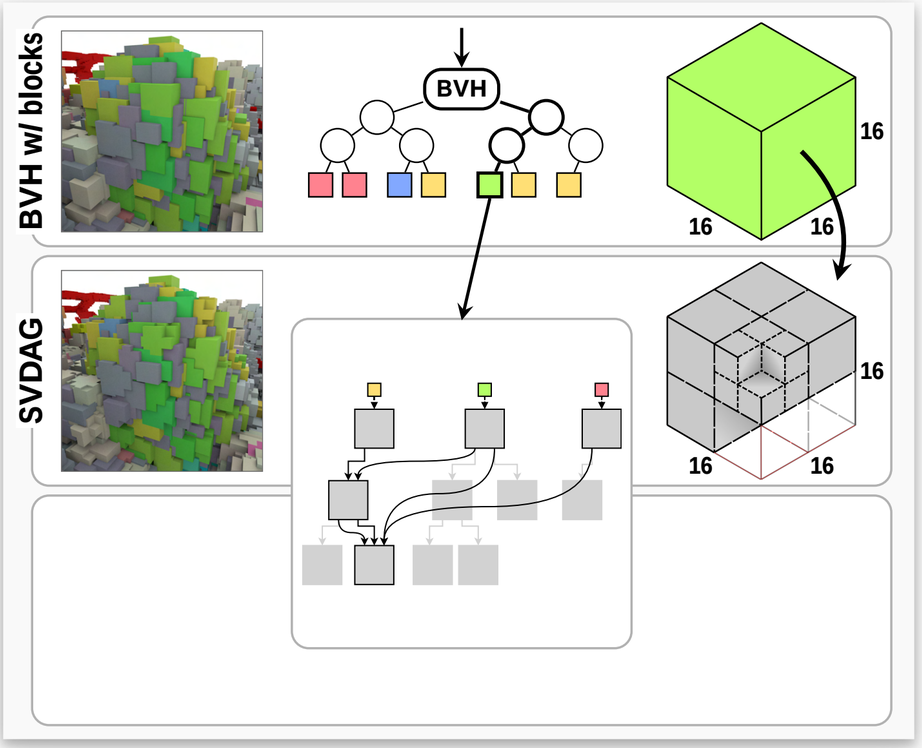




Data Structure - Level 2 (SVDAG) ➡ encode voxel occupancy

- single large SVDAG across all blocks

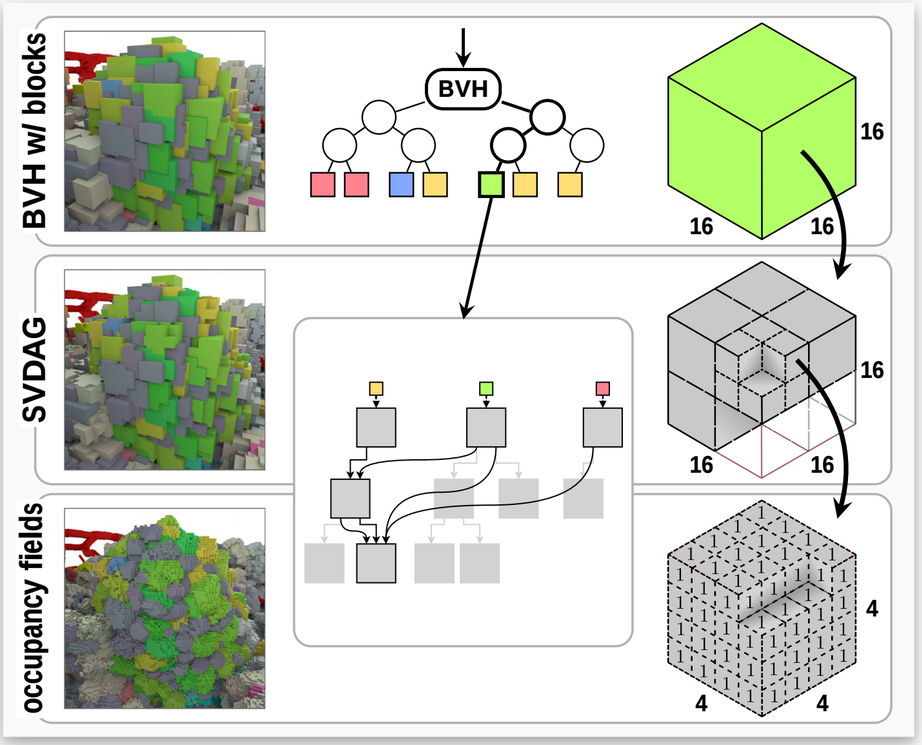




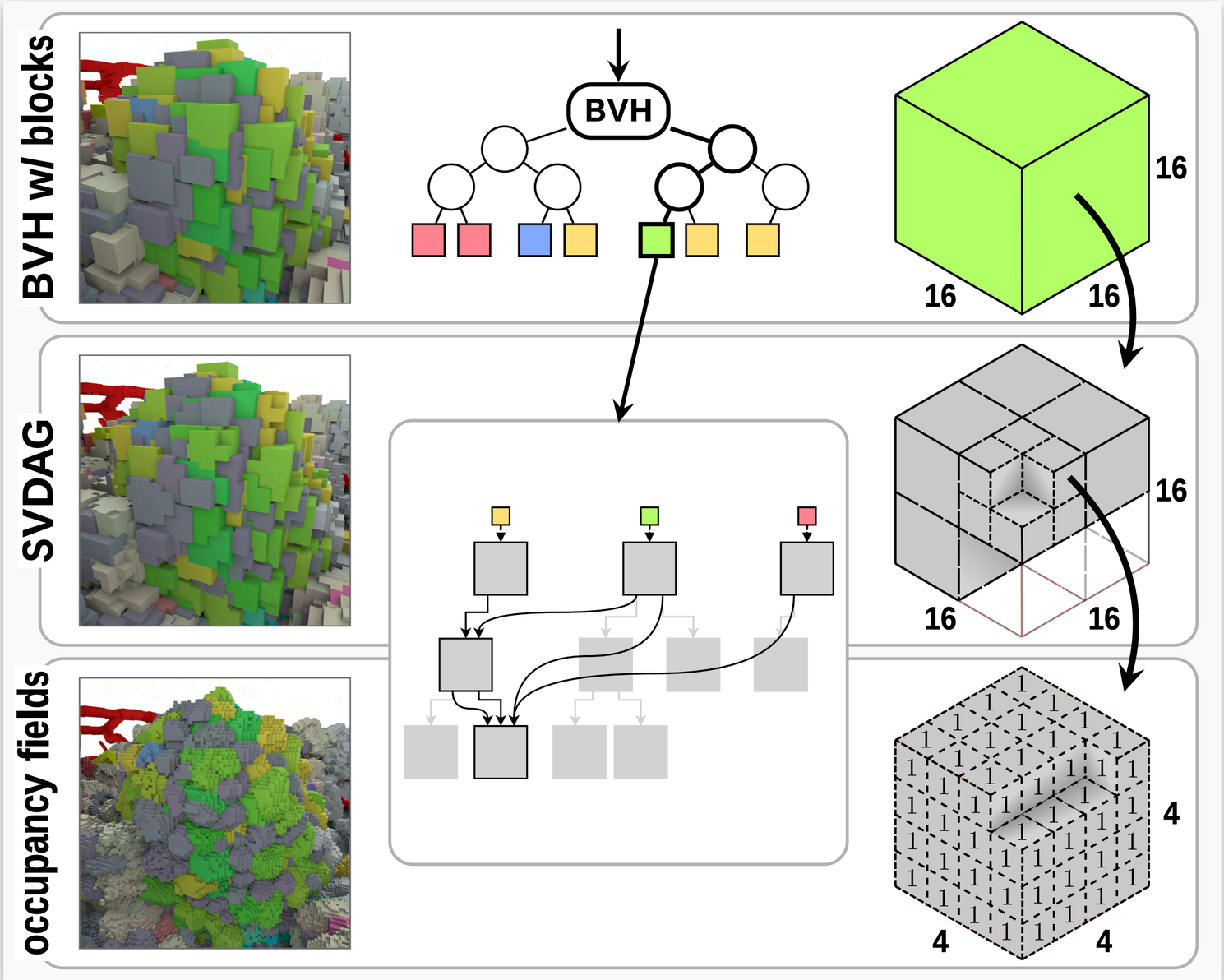
Data Structure - Level 3 (Occupancy Fields)

➡ improve traversal performance

- store occupancy directly in 4^3 SVDAG leaf in bitmask
  - $4^3 = 64$ bits
- less VRAM accesses



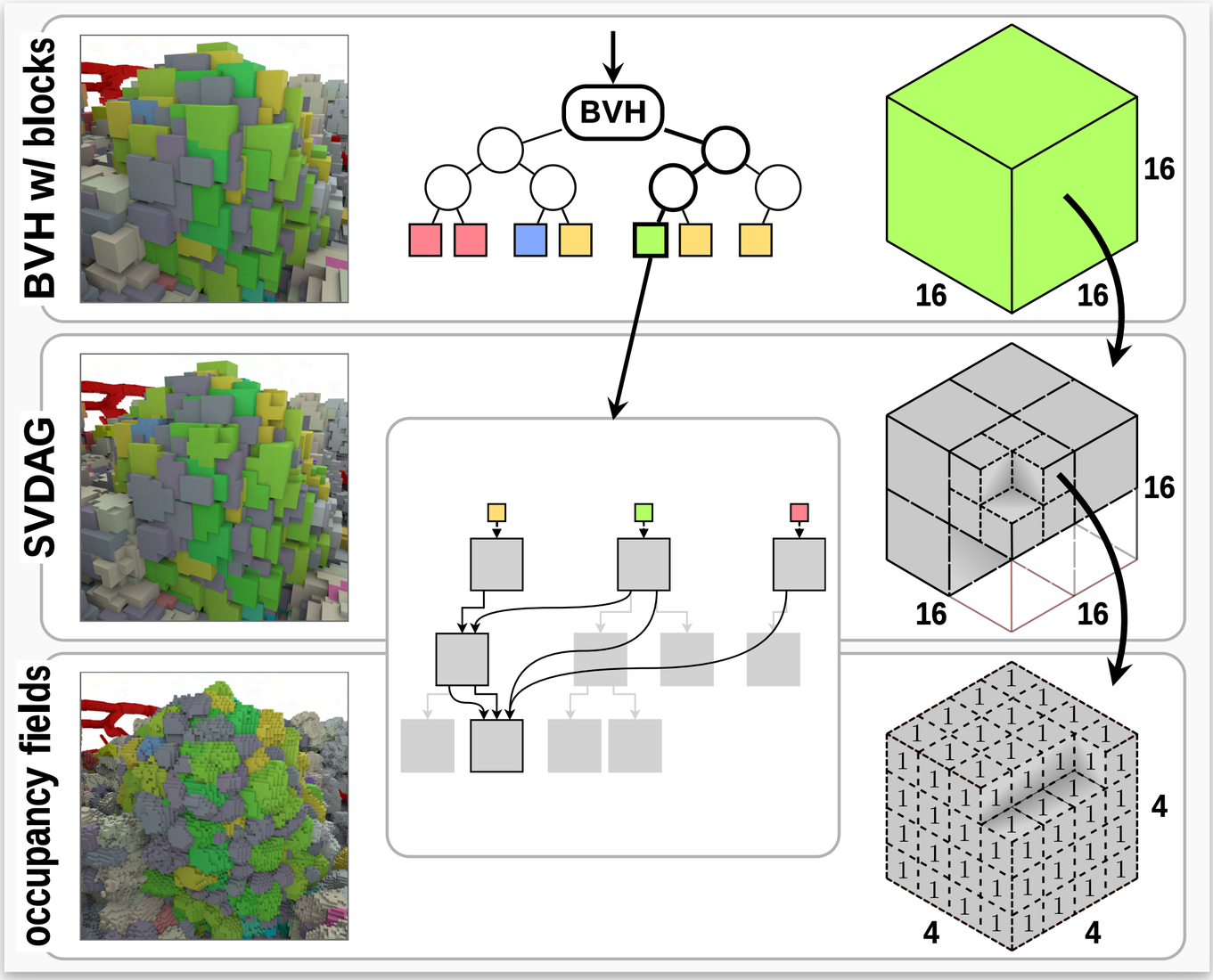
Data Structure



Compressed Segmentation Volume Path Tracing

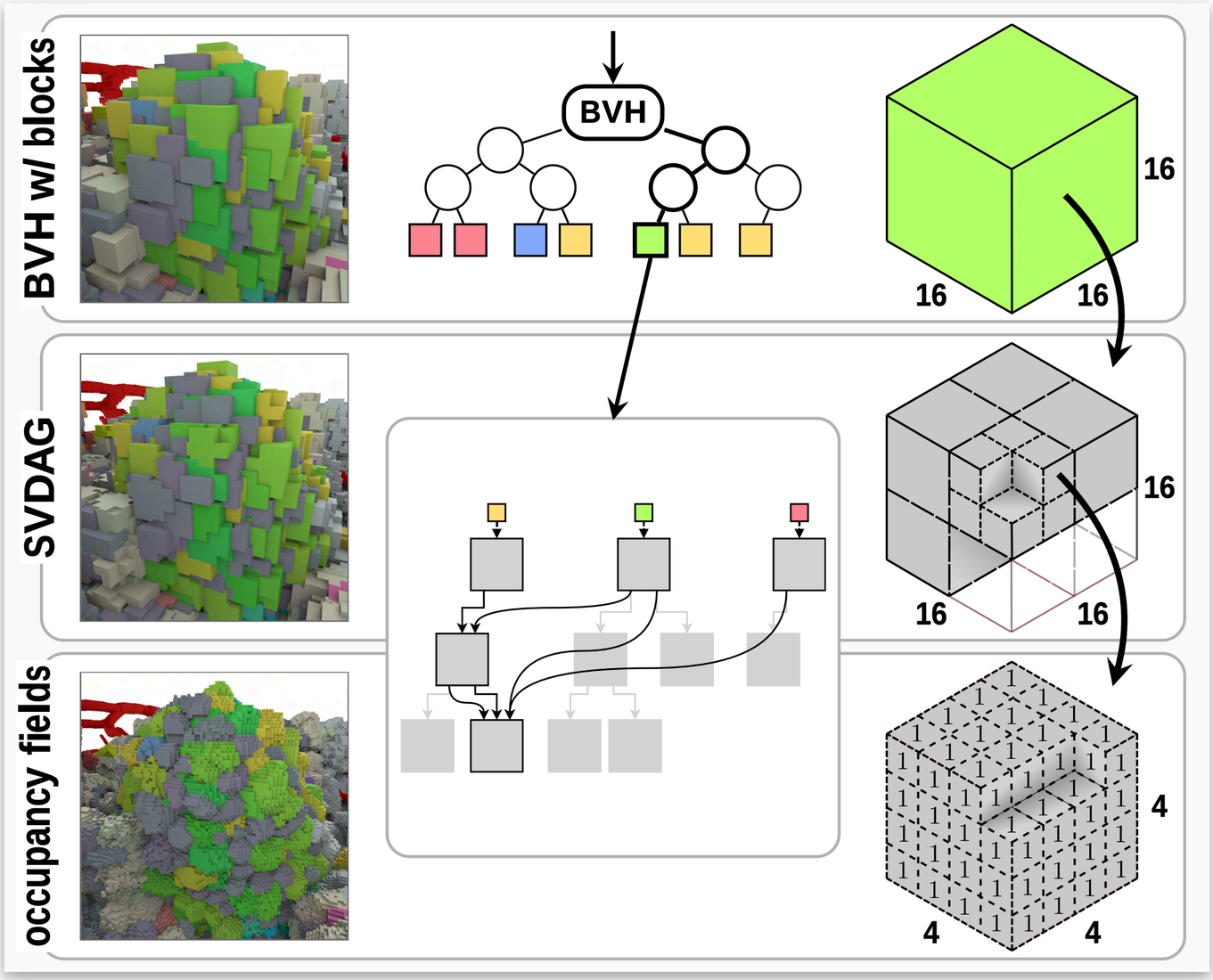
(traversal of our data structure)

GPU Traversal



- hardware-accelerated ray tracing
  - invokes custom intersection shader

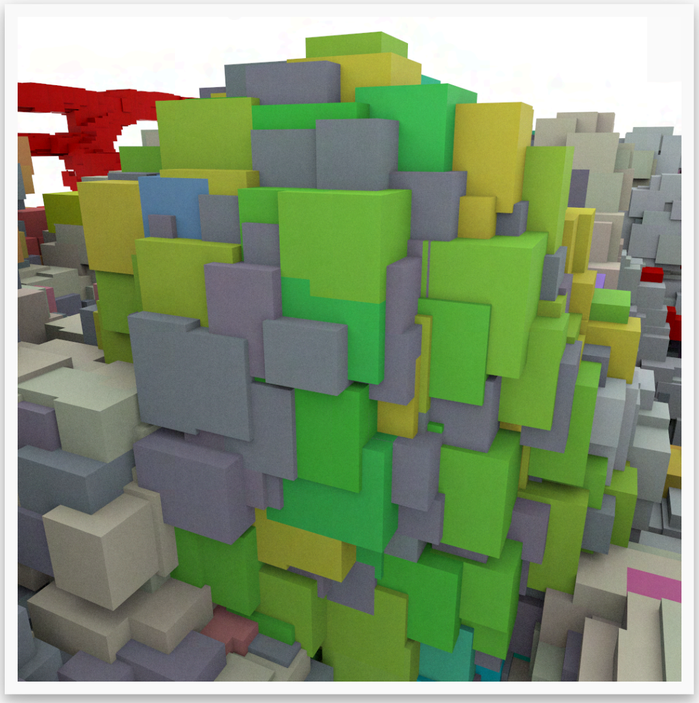
GPU Traversal



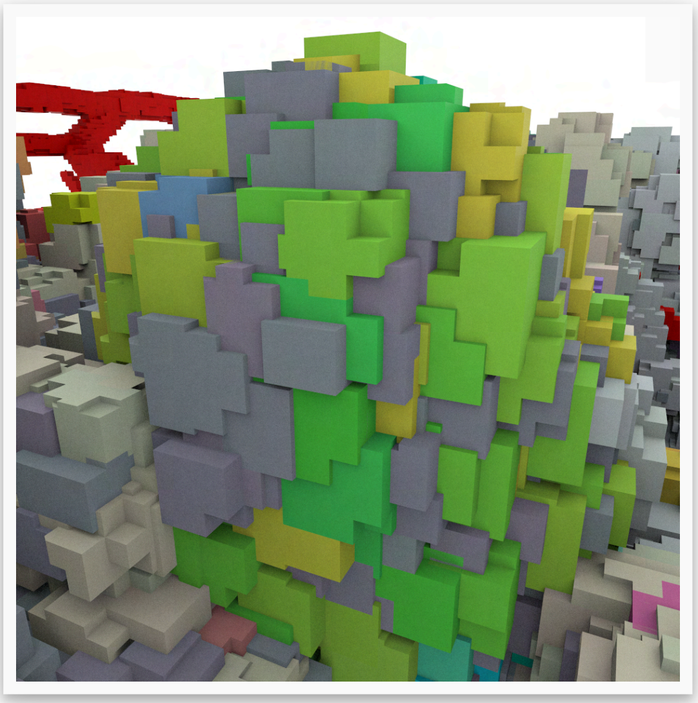
- hardware-accelerated ray tracing
  - invokes custom intersection shader

- query label and pointer to the SVDAG root node
- (multi-level) 3D-DDA traversal of SVDAG and occupancy field

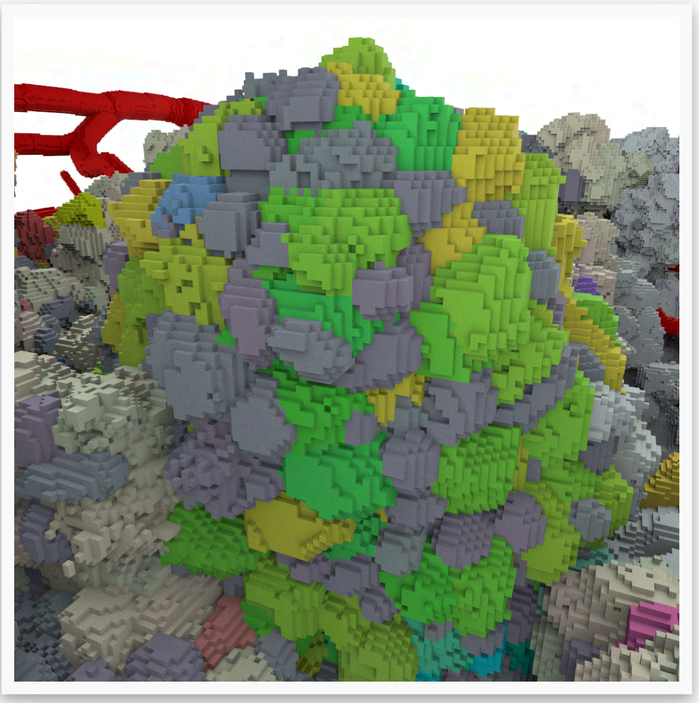
Level of Detail



block
(16^3)



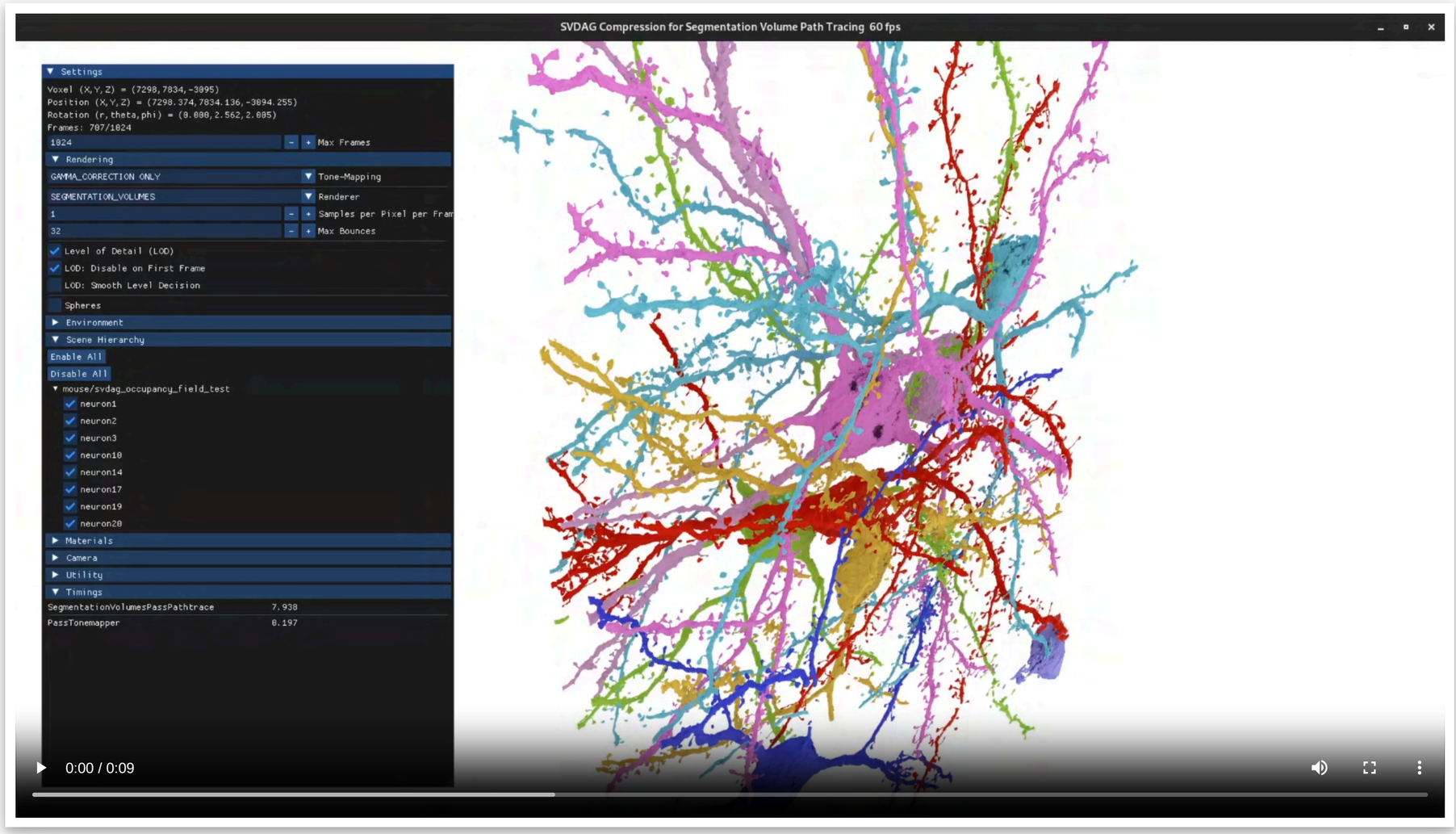
SVDAG leaf
(4^3)

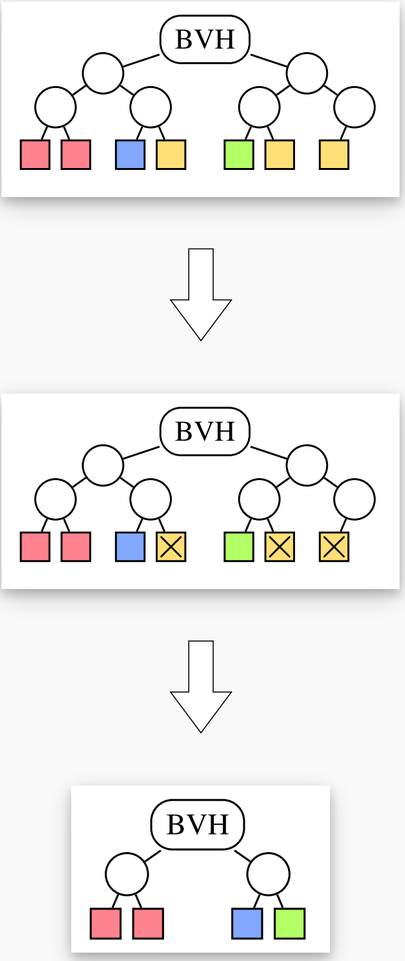


occupancy field
(1^3)

Transfer Function Editing

Transfer Function Editing

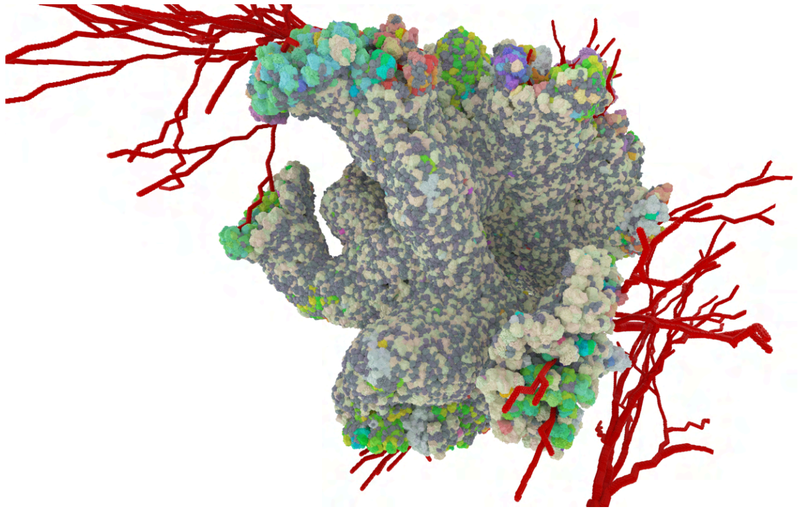




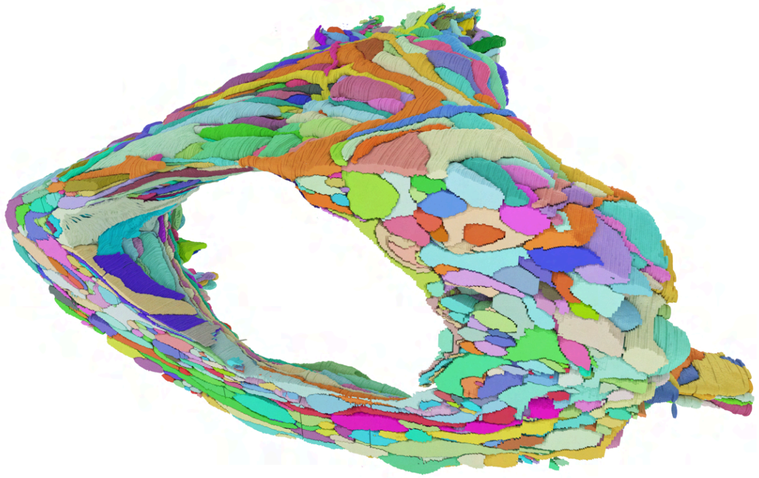
Results

Datasets

CELLS (raw 4.0 GB)



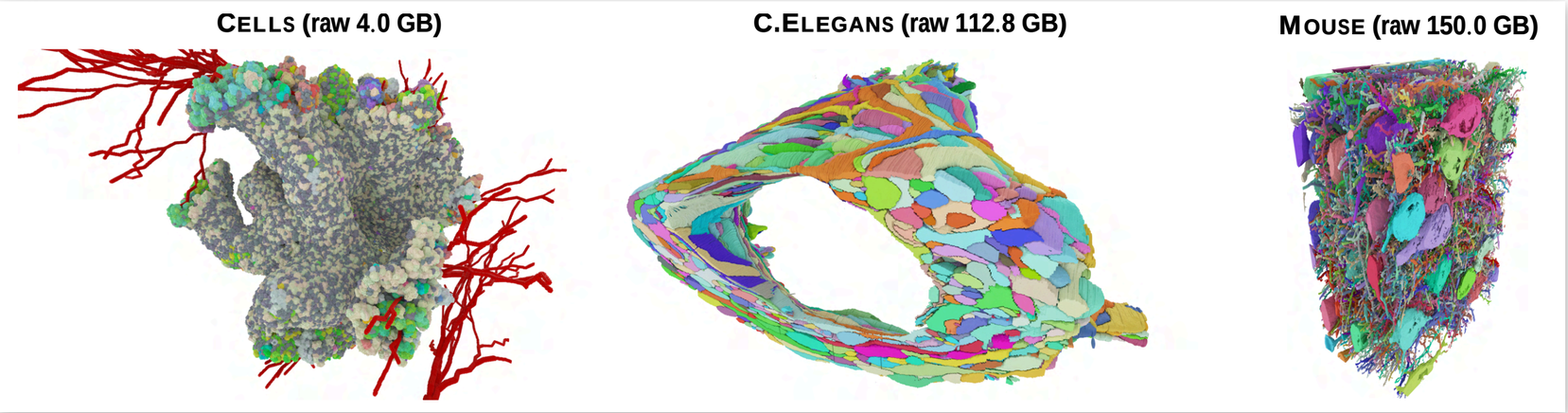
C.ELEGANS (raw 112.8 GB)



MOUSE (raw 150.0 GB)

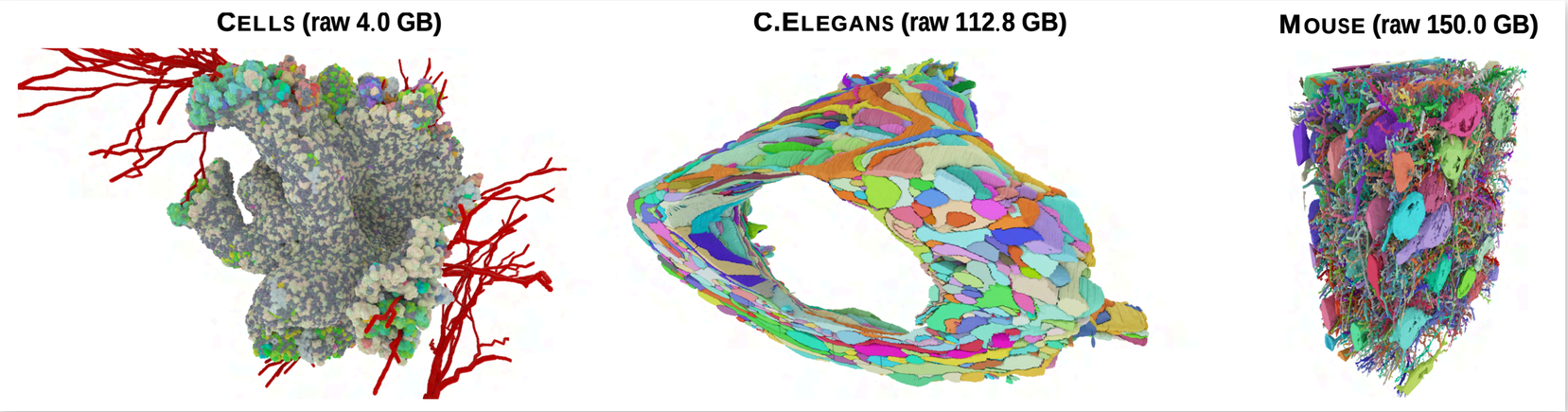
Memory (mem) and Compression Rate (CR)

	blocks + BVH + SVOs mem. [MB]	CR
CELLS	253.97	6.35%
C.ELEGANS	14322.37	12.7%
MOUSE	62911.29	41.95%



Memory (mem) and Compression Rate (CR)

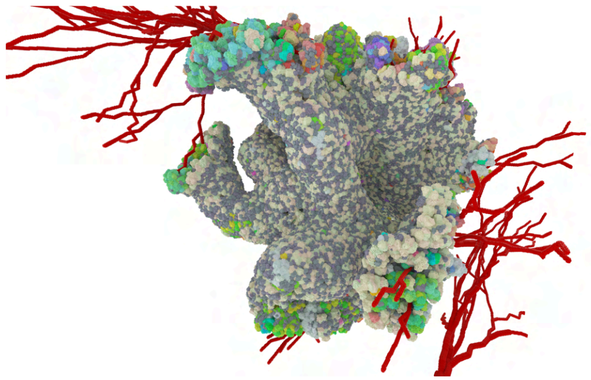
	blocks + BVH + SVOs mem. [MB]	CR
CELLS	253.97	6.35%
C.ELEGANS	14322.37	12.7%
MOUSE	62911.29	41.95%



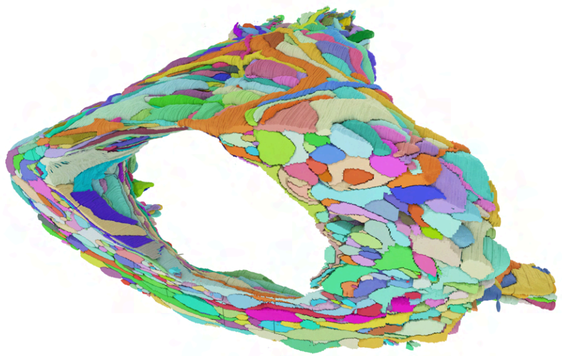
Memory (mem) and Compression Rate (CR)

	blocks + BVH + SVOs mem. [MB]	CR
CELLS	253.97	6.35%
C.ELEGANS	14322.37	12.7%
MOUSE	62911.29	41.95%

CELLS (raw 4.0 GB)



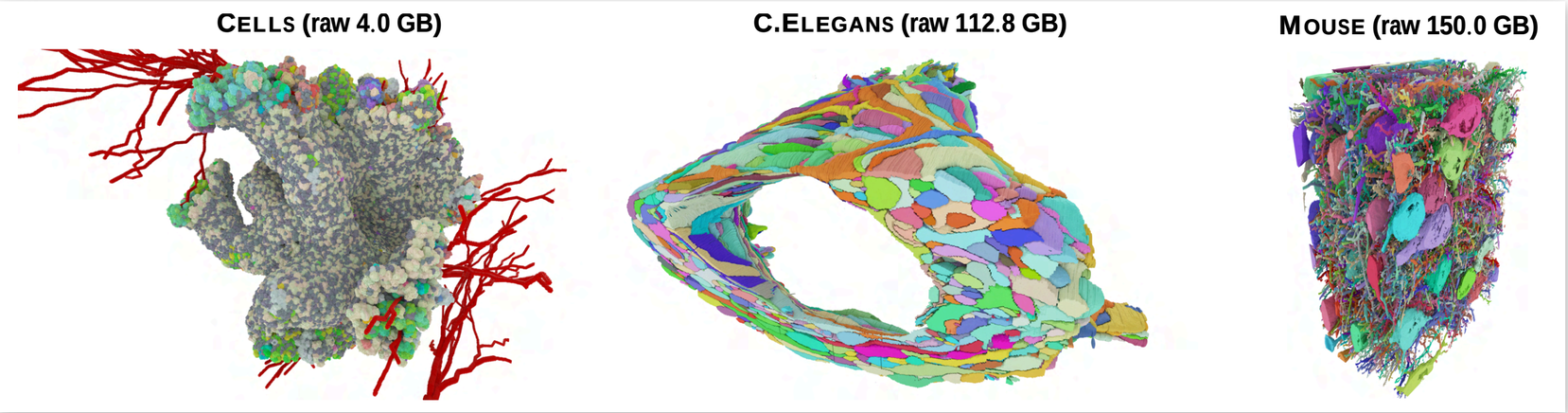
C.ELEGANS (raw 112.8 GB)



MOUSE (raw 150.0 GB)

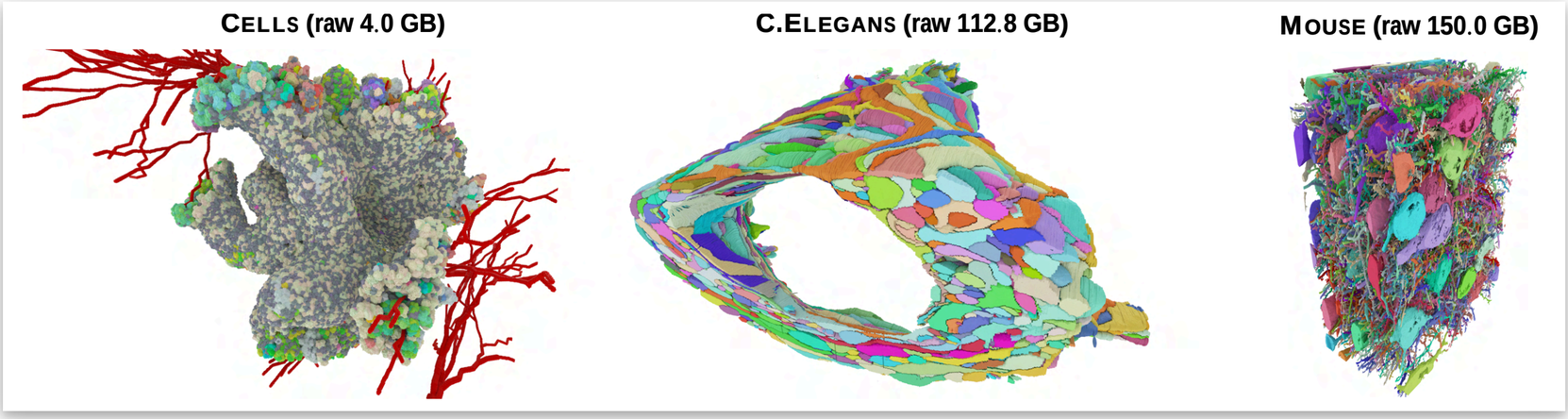
Memory (mem) and Compression Rate (CR)

	blocks + BVH + SVOs mem. [MB]	CR
CELLS	253.97	6.35%
C.ELEGANS	14322.37	12.7%
MOUSE	62911.29	41.95%



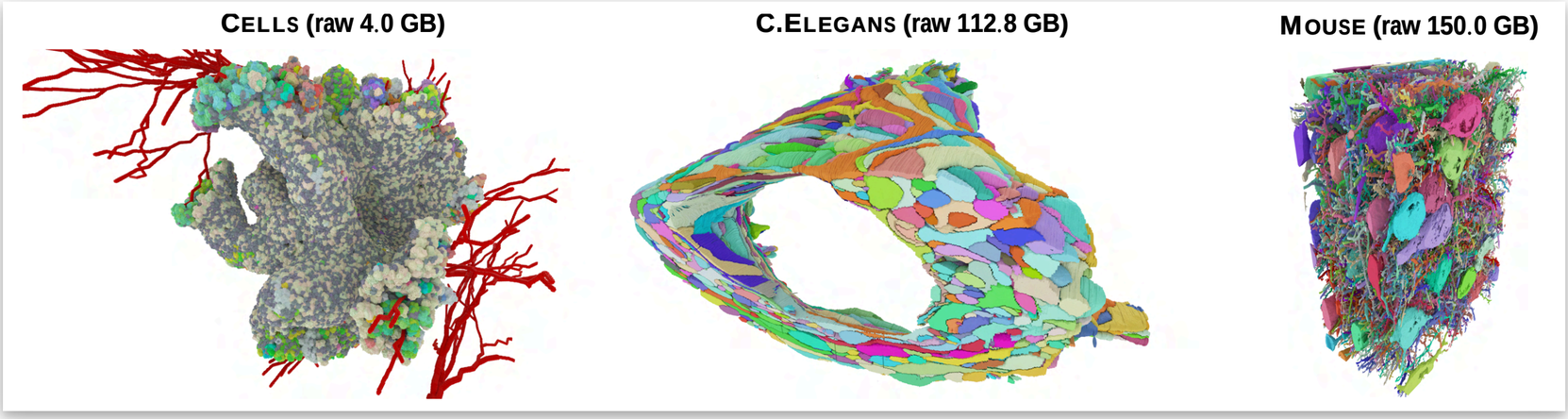
Memory (mem) and Compression Rate (CR)

	blocks + BVH + SVOs		blocks + BVH + SVDAG	
	mem. [MB]	CR	mem. [MB]	CR
CELLS	253.97	6.35%	86.34	2.16%
C.ELEGANS	14322.37	12.7%	2534.28	2.25%
MOUSE	62911.29	41.95%	12095.48	8.06%



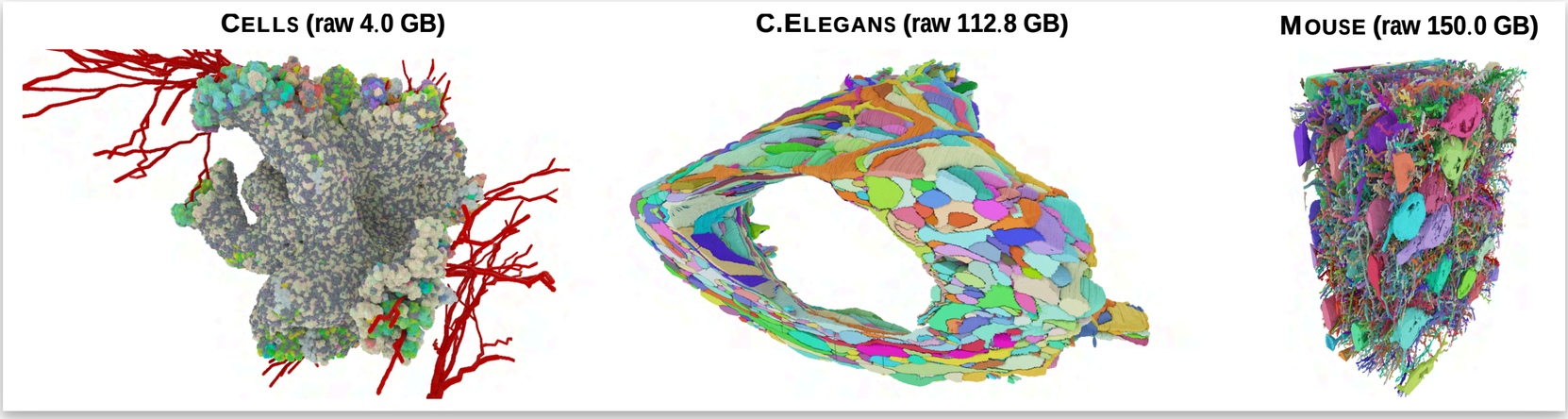
Memory (mem) and Compression Rate (CR)

	blocks + BVH + SVOs		blocks + BVH + SVDAG	
	mem. [MB]	CR	mem. [MB]	CR
CELLS	253.97	6.35%	86.34	2.16%
C.ELEGANS	14322.37	12.7%	2534.28	2.25%
MOUSE	62911.29	41.95%	12095.48	8.06%



Memory (mem) and Compression Rate (CR)

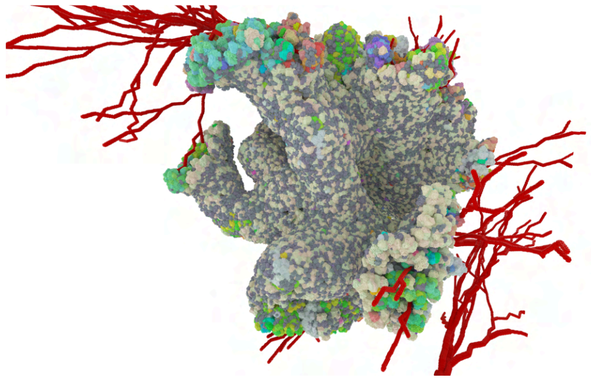
	blocks + BVH + SVOs		blocks + BVH + SVDAG	
	mem. [MB]	CR	mem. [MB]	CR
CELLS	253.97	6.35%	86.34	2.16%
C.ELEGANS	14322.37	12.7%	2534.28	2.25%
MOUSE	62911.29	41.95%	12095.48	8.06%



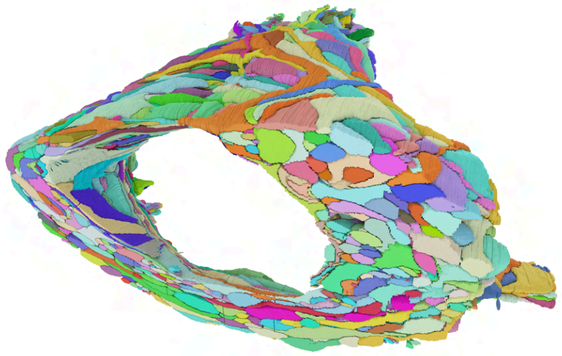
Rendering Performance

		SVOs [ms] finest LOD	SVOs [ms] coarsest LOD
1 bounce	CELLS	8.19	0.4
	C.ELEGANS	–	–
	MOUSE	–	–
32 bnc.	CELLS	11.42	0.81
	C.ELEGANS	–	–
	MOUSE	–	–

CELLS (raw 4.0 GB)



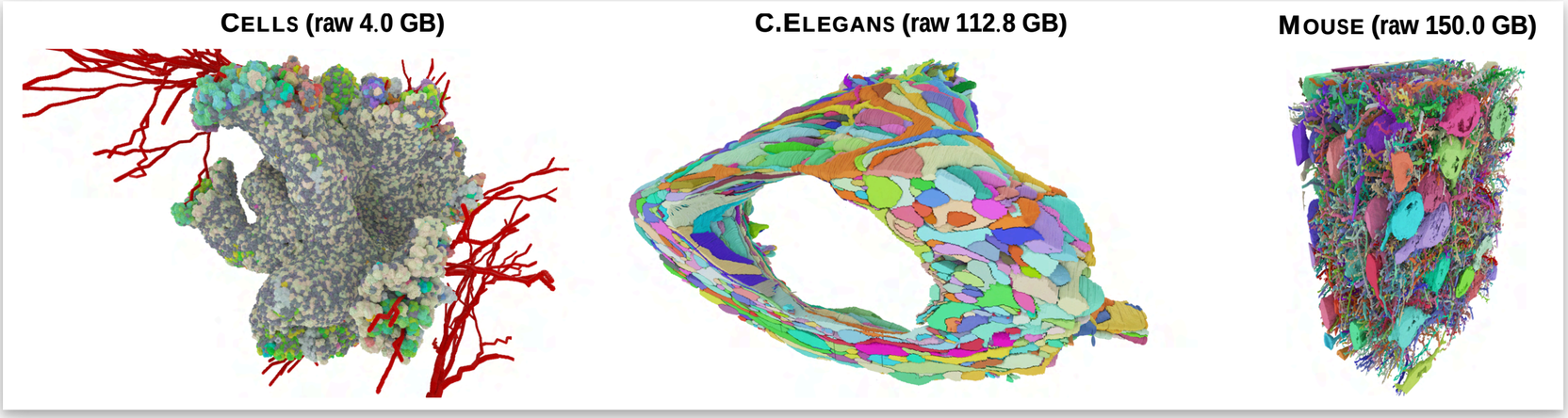
C.ELEGANS (raw 112.8 GB)



MOUSE (raw 150.0 GB)

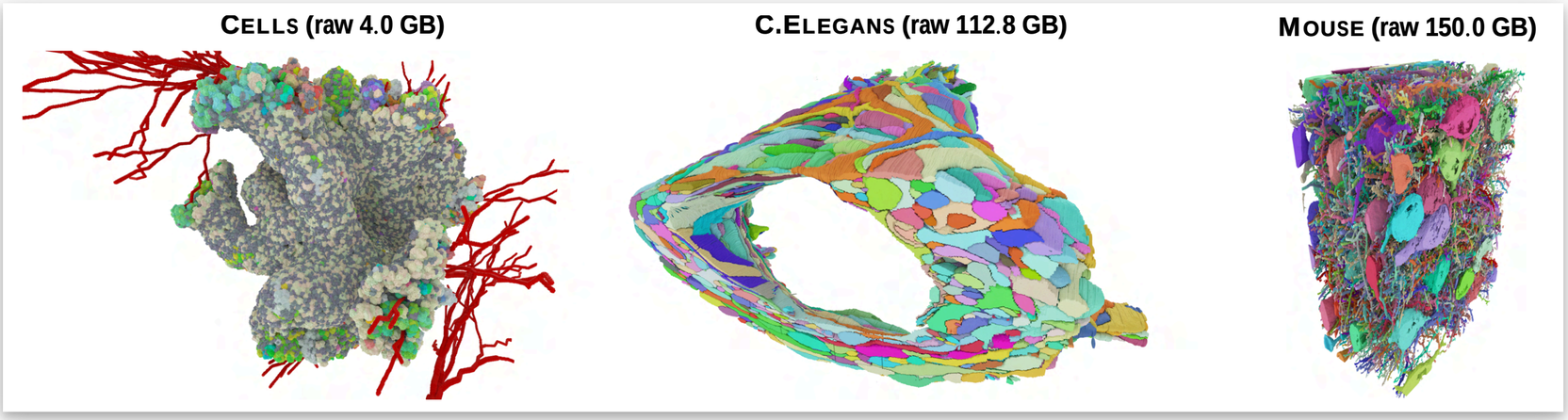
Rendering Performance

		SVOs [ms]		SVDAG [ms]	
		finest LOD	coarsest LOD	finest LOD	coarsest LOD
1 bounce	CELLS	8.19	0.4	2.95	0.51
	C.ELEGANS	–	–	5.61	0.55
	MOUSE	–	–	126.73	0.98
32 bnc.	CELLS	11.42	0.81	5.31	1.02
	C.ELEGANS	–	–	9.25	0.98
	MOUSE	–	–	189.54	2.0



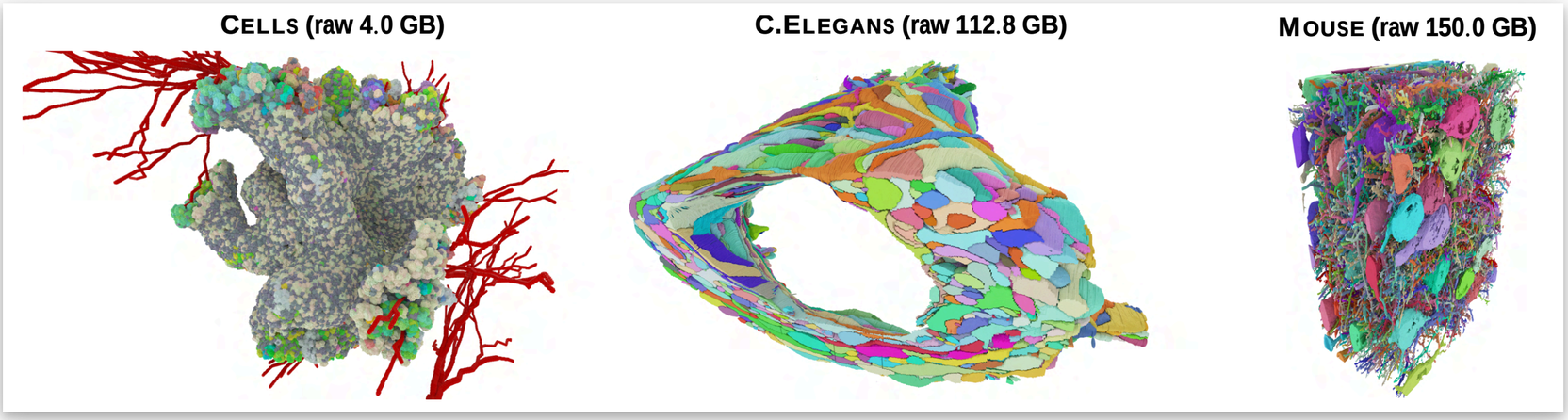
Rendering Performance

		SVOs [ms]		SVDAG [ms]	
		finest LOD	coarsest LOD	finest LOD	coarsest LOD
1 bounce	CELLS	8.19	0.4	2.95	0.51
	C.ELEGANS	–	–	5.61	0.55
	MOUSE	–	–	126.73	0.98
32 bnc.	CELLS	11.42	0.81	5.31	1.02
	C.ELEGANS	–	–	9.25	0.98
	MOUSE	–	–	189.54	2.0



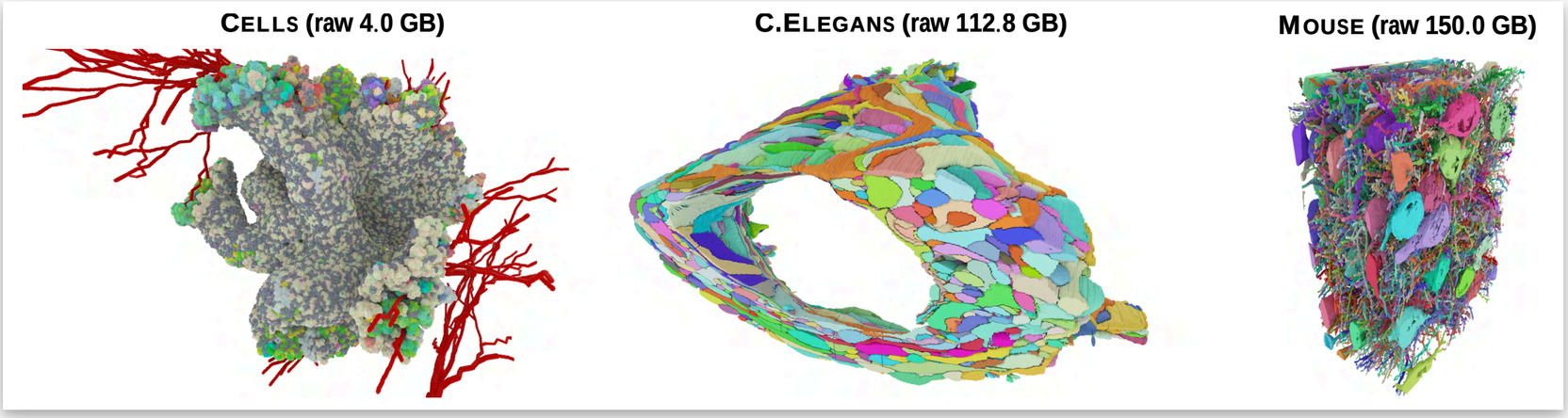
Rendering Performance

		SVOs [ms]		SVDAG [ms]	
		finest LOD	coarsest LOD	finest LOD	coarsest LOD
1 bounce	CELLS	8.19	0.4	2.95	0.51
	C.ELEGANS	–	–	5.61	0.55
	MOUSE	–	–	126.73	0.98
32 bnc.	CELLS	11.42	0.81	5.31	1.02
	C.ELEGANS	–	–	9.25	0.98
	MOUSE	–	–	189.54	2.0



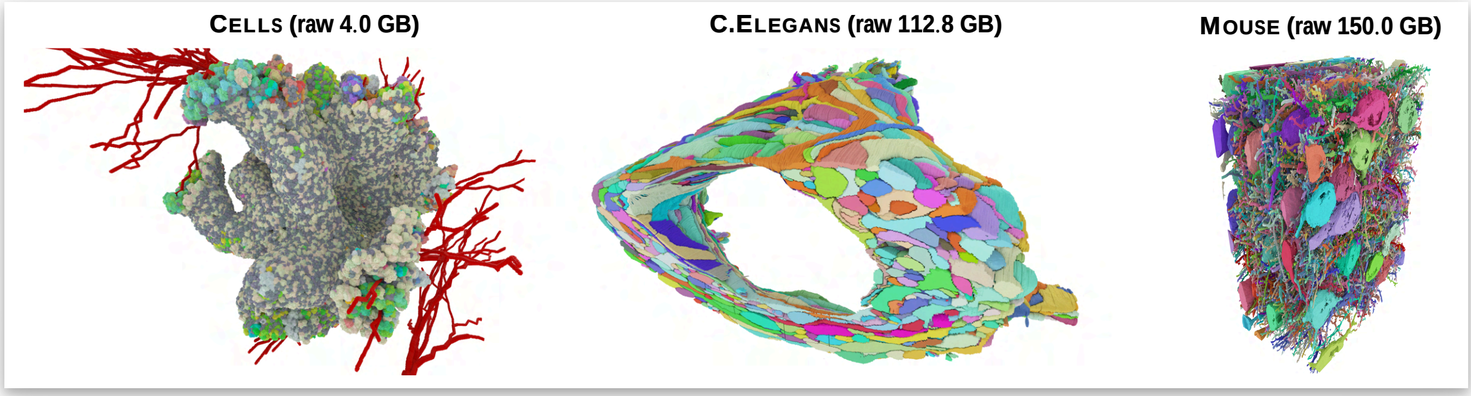
Rendering Performance

		SVOs [ms]		SVDAG [ms]	
		finest LOD	coarsest LOD	finest LOD	coarsest LOD
1 bounce	CELLS	8.19	0.4	2.95	0.51
	C.ELEGANS	–	–	5.61	0.55
	MOUSE	–	–	126.73	0.98
32 bnc.	CELLS	11.42	0.81	5.31	1.02
	C.ELEGANS	–	–	9.25	0.98
	MOUSE	–	–	189.54	2.0



Rendering Performance

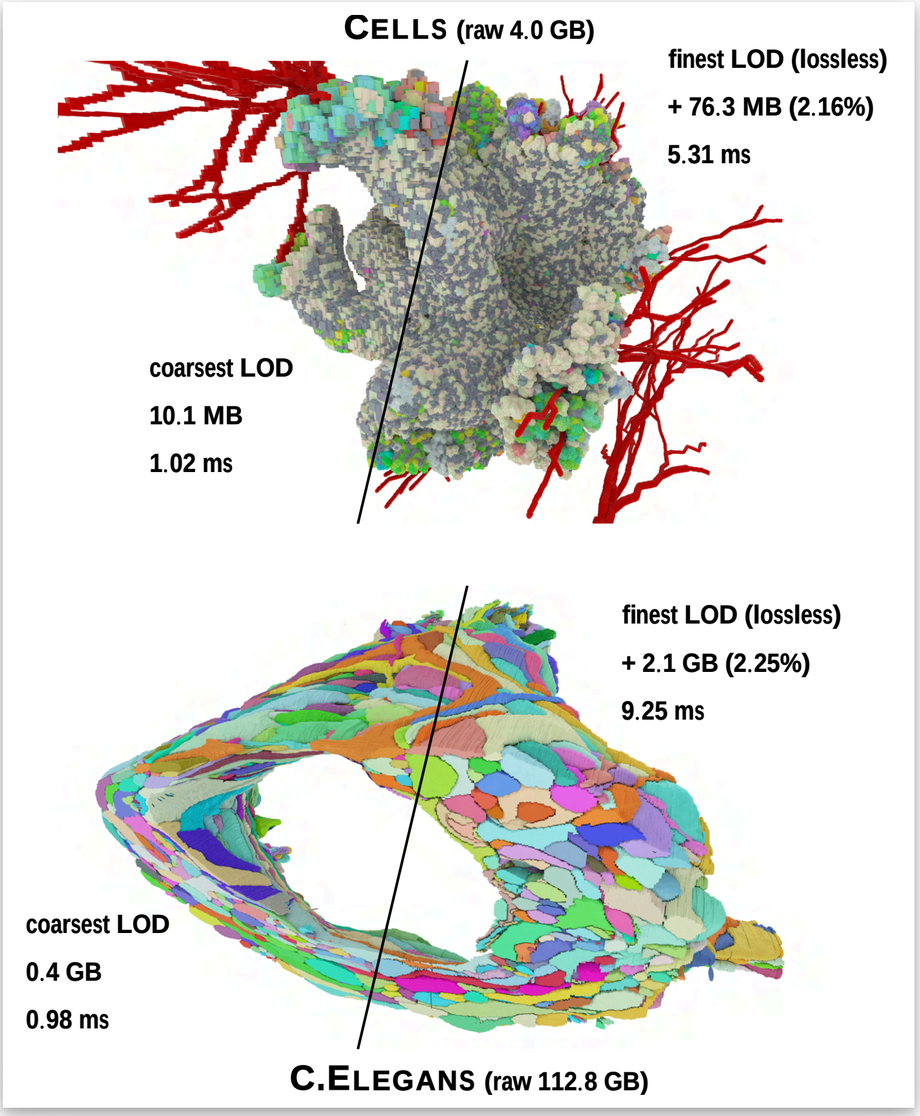
		SVOs [ms]		SVDAG [ms]	
		finest LOD	coarsest LOD	finest LOD	coarsest LOD
1 bounce	CELLS	8.19	0.4	2.95	0.51
	C.ELEGANS	–	–	5.61	0.55
	MOUSE	–	–	126.73	0.98
32 bnc.	CELLS	11.42	0.81	5.31	1.02
	C.ELEGANS	–	–	9.25	0.98
	MOUSE	–	–	189.54	2.0



Conclusion

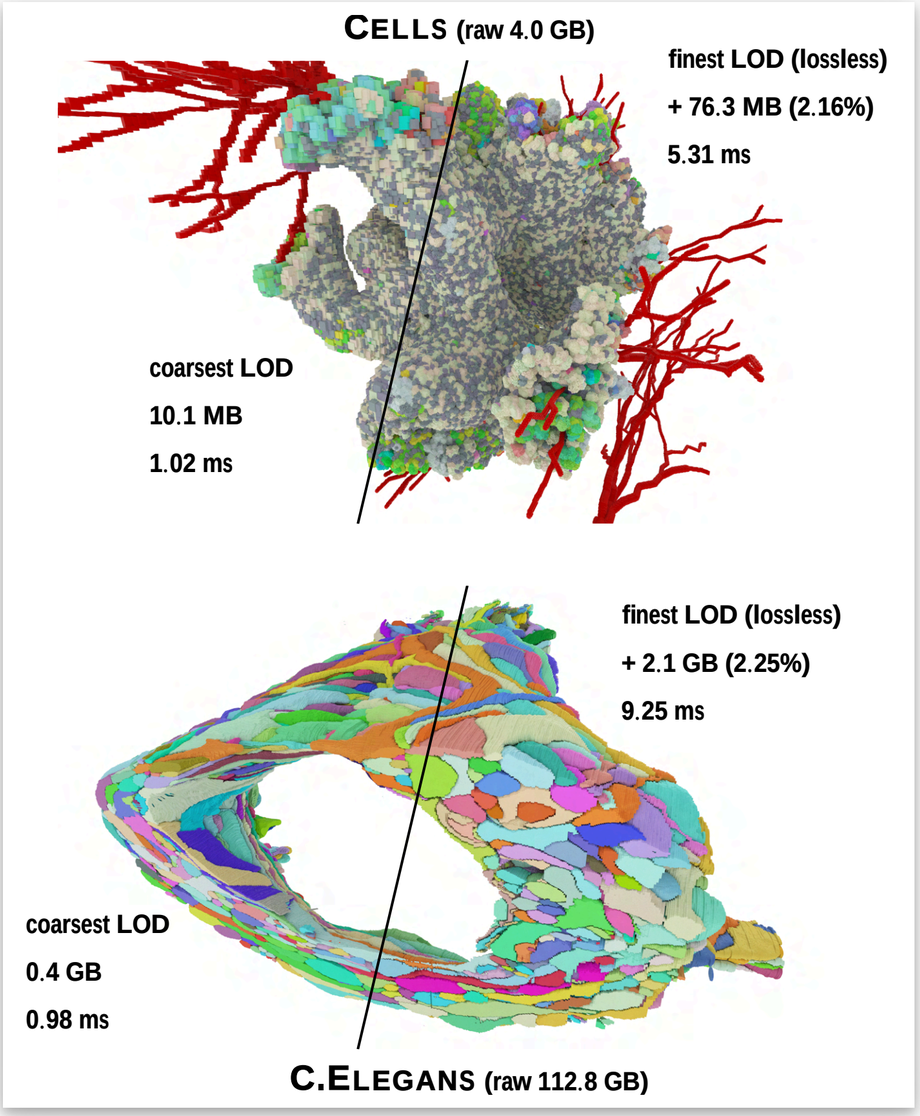
Conclusion

- introduced a
  - lossless SVDAG compression method



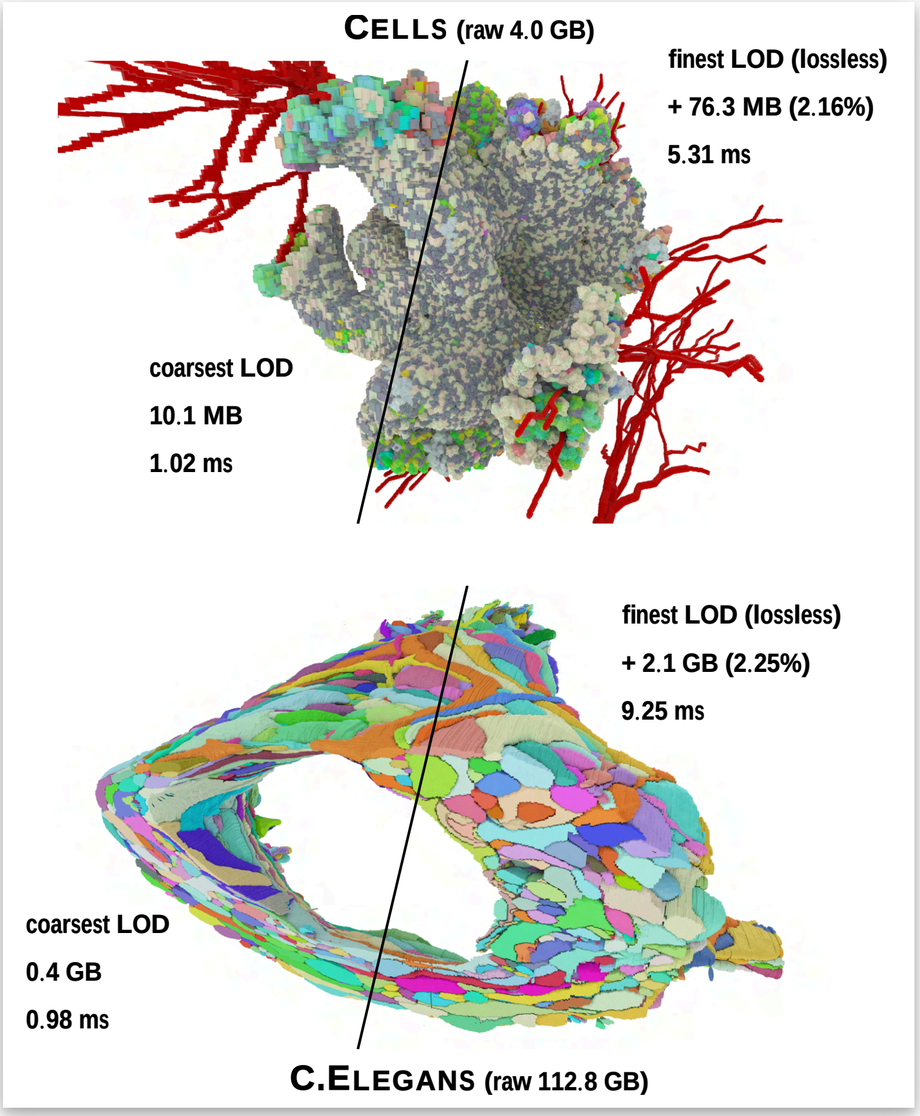
Conclusion

- introduced a
  - lossless SVDAG compression method
  - and a hardware-accelerated path tracing framework



Conclusion

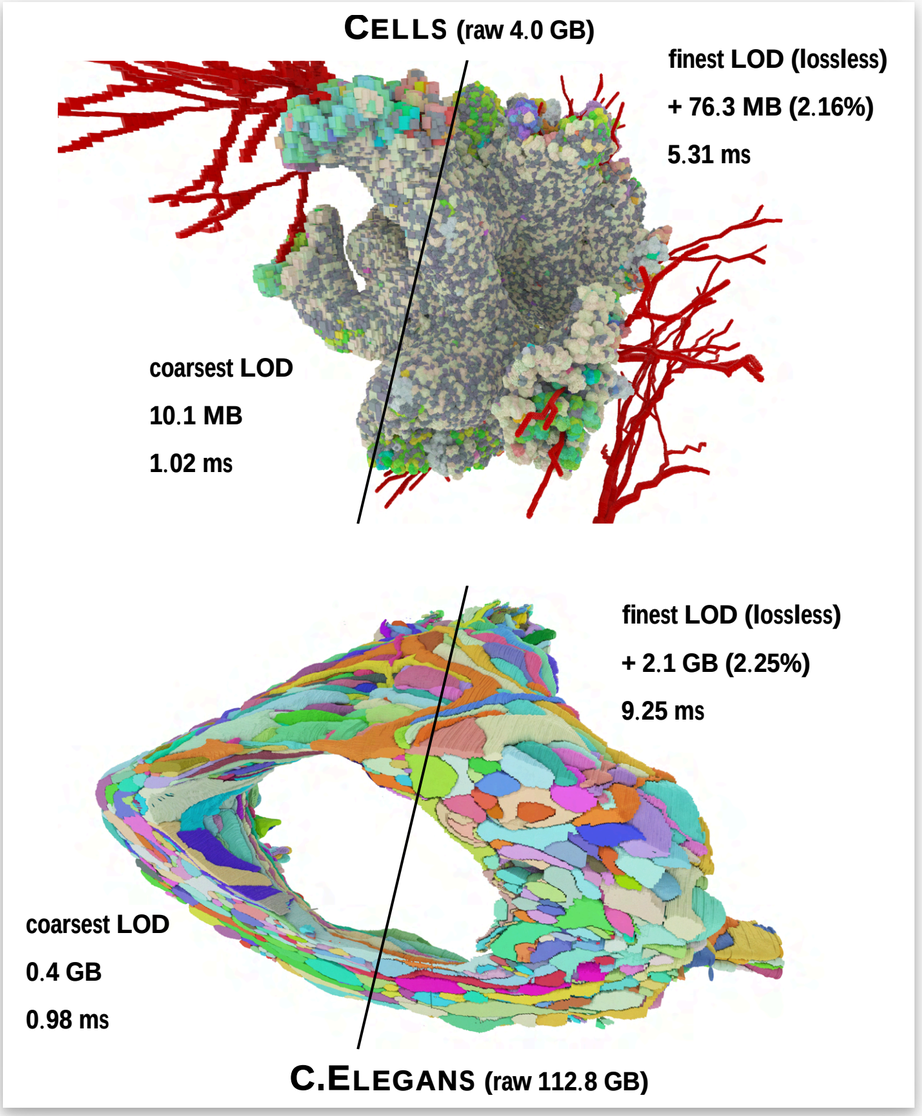
- introduced a
  - lossless SVDAG compression method
  - and a hardware-accelerated path tracing framework
- for interactive rendering of segmentation volumes



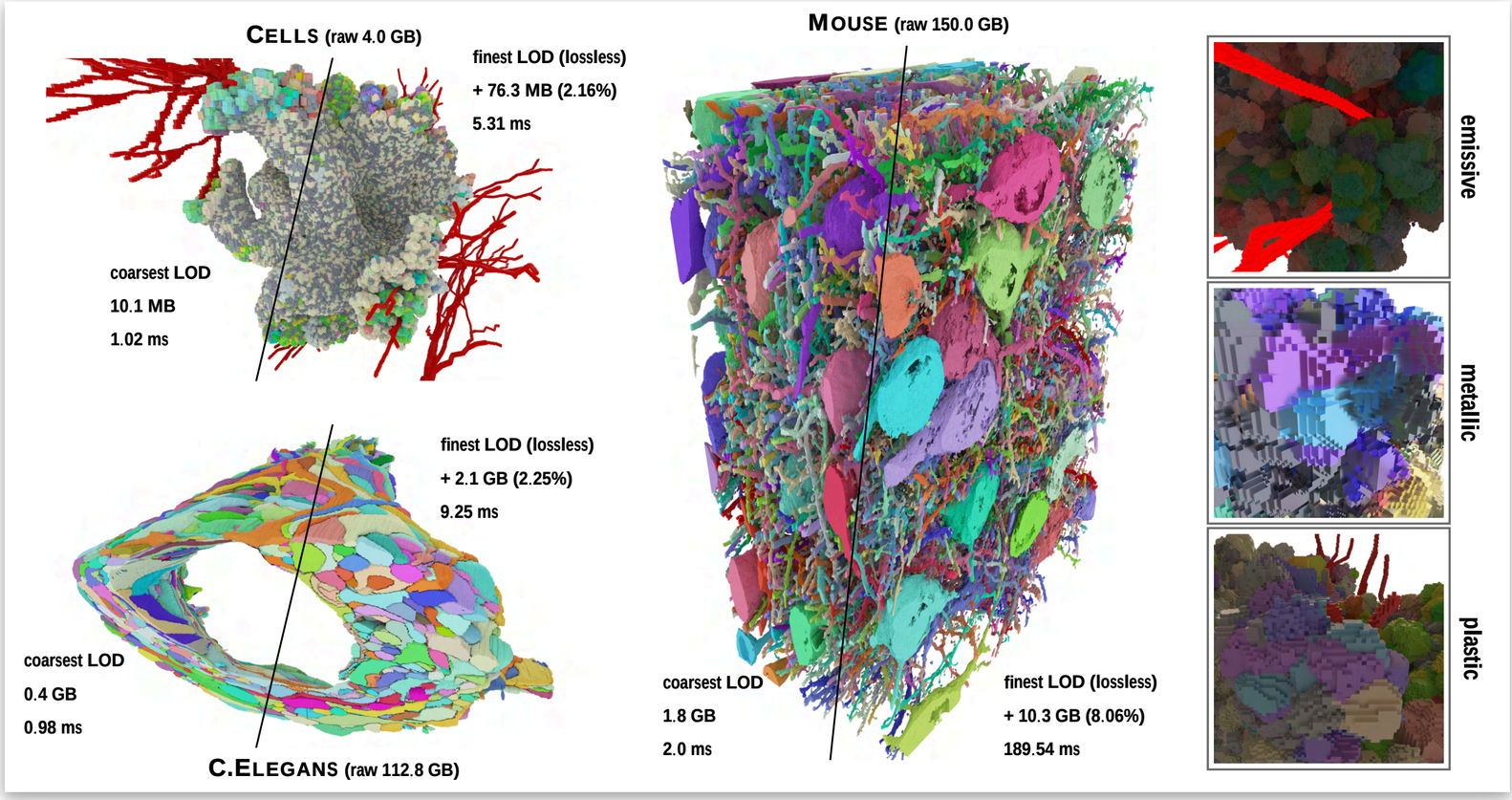
Conclusion

- introduced a
  - lossless SVDAG compression method
  - and a hardware-accelerated path tracing framework
- for interactive rendering of segmentation volumes

- strong compression and rendering performance



Thank you!



Code available on GitHub:

https://github.com/MircoWerner/SegmentationVolumeCompression

References

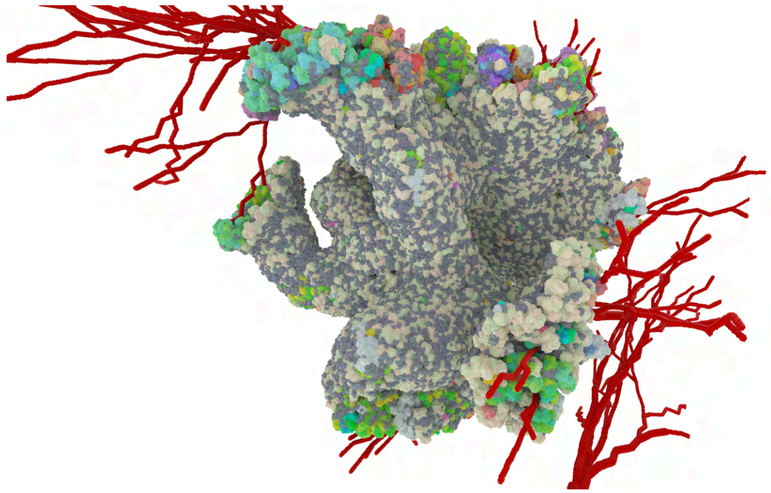
- [Rosenbauer*2020]
  - ROSENBAUER, J., BERGHOFF, M., and SCHUG, A. “Emerging Tumor Development by Simulating Single-cell Events”. bioRxiv (2020). DOI: 10.1101/2020.08.24.264150
- [Witvliet*2021]
  - WITVLIET, D., MULCAHY, B., MITCHELL, J. K., et al. “Connectomes across development reveal principles of brain maturation”. Nature 596.7871 (2021), 257-261. DOI: 10.1038/s41586-021-03778-8
- [Motta*2019]
  - MOTTA, A., BERNING, M., BOERGENS, K. M., et al. “Dense connectomic reconstruction in layer 4 of the somatosensory cortex”. Science 366.6469 (2019), eaay3134. DOI: 10.1126/science.aay3134
- [Velicky*2023]
  - VELICKY, P., MIGUEL, E., MICHALSKA, J.M. et al. Dense 4D nanoscale reconstruction of living brain tissue. Nat Methods 20, 1256-1265 (2023). https://doi.org/10.1038/s41592-023-01936-6
- [Laine*2010]
  - LAINE, S. and KARRAS, T. “Efficient sparse voxel octrees”. Proc. ACM SIGGRAPH Symposium on Interact. 3D Graph. and Games. New York, NY, USA: ACM, 2010, 55-63. DOI: 10.1145/1730804.1730814
- [Kämpe*2013]
  - KÄMPE, V., SINTORN, E., and ASSARSSON, U. “High resolution sparse voxel DAGs”. ACM Transactions on Graphics 32.4 (July 2013). DOI: 10.1145/2461912.2462024

Appendix

SVDAG Node Memory Layout

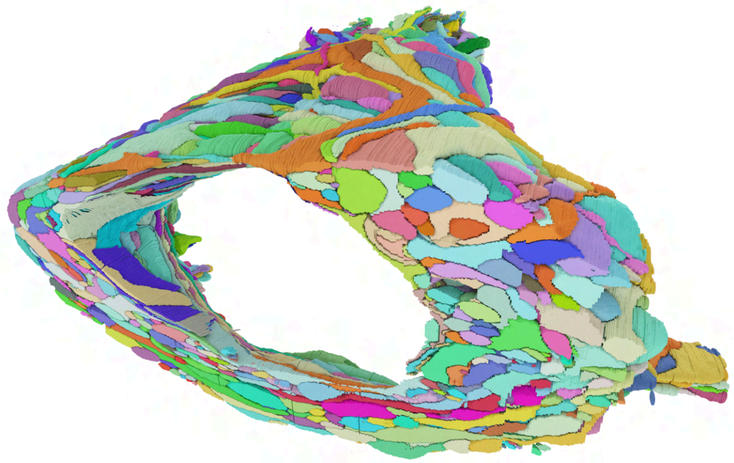
Datasets

CELLS (raw 4.0 GB)



$1000 \times 1000 \times 1000 = 1.0 \cdot 10^9$
15.39% solid
118251 labels

C.ELEGANS (raw 112.8 GB)



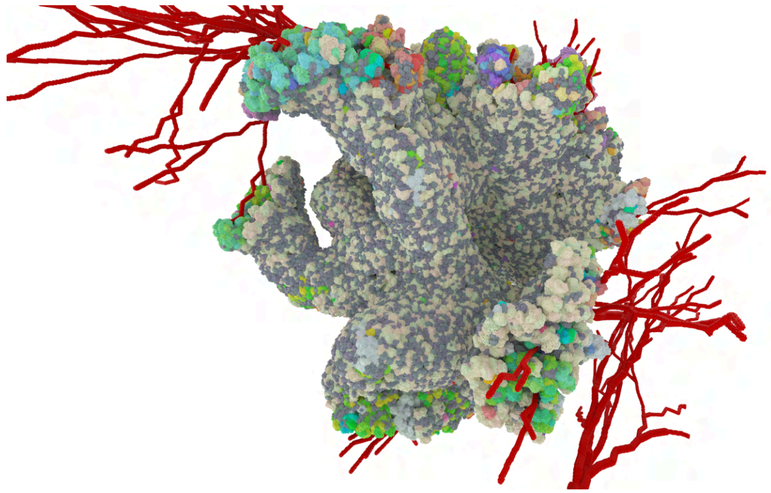
$661 \times 11044 \times 15448 = 112.77 \cdot 10^9$
5.45% solid
234 labels

MOUSE (raw 150.0 GB)

$5446 \times 8381 \times 3286 = 149.98 \cdot 10^9$
15.57% solid
96 labels

Datasets

CELLS (raw 4.0 GB)

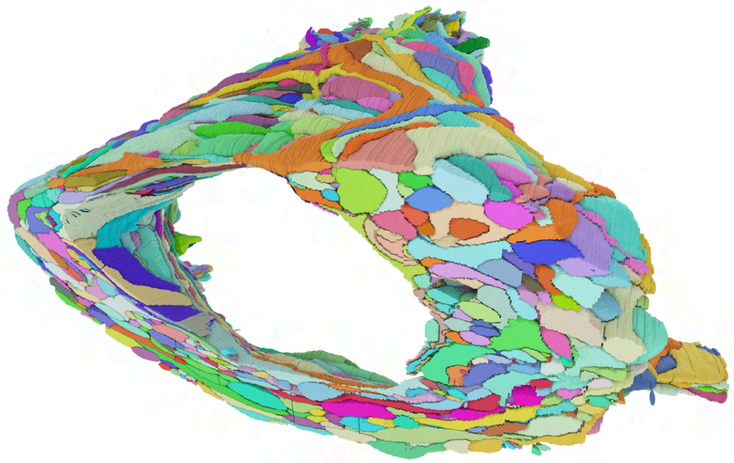


$1000 \times 1000 \times 1000 = 1.0 \cdot 10^9$

15.39% solid

118251 labels

C.ELEGANS (raw 112.8 GB)



$661 \times 11044 \times 15448 = 112.77 \cdot 10^9$

5.45% solid

234 labels

MOUSE (raw 150.0 GB)

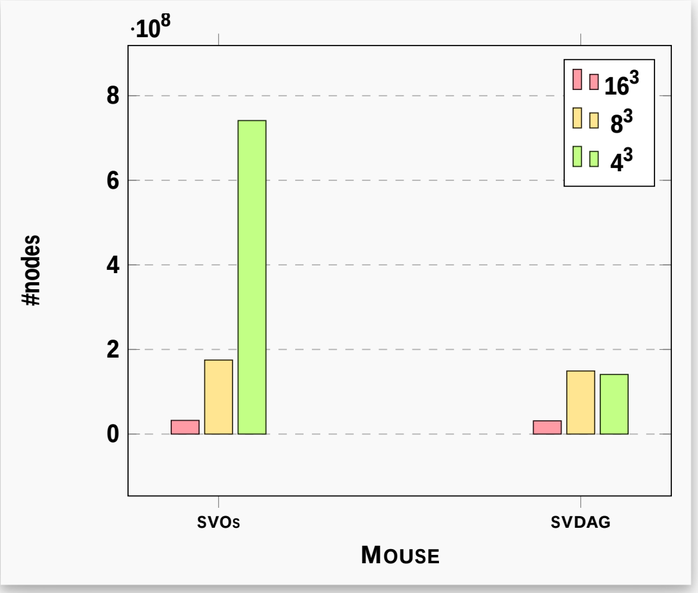
$5446 \times 8381 \times 3286 = 149.98 \cdot 10^9$

15.57% solid

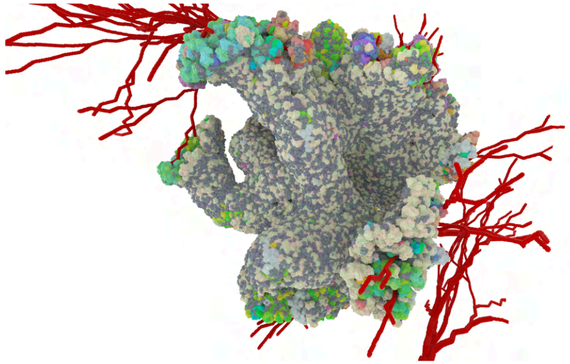
96 labels

Memory (mem) and Compression Rate (CR)

	AABBs + BVH + mem. [MB]	SVOs mem. [MB]	SVOs CR	SVDAG mem. [MB]	SVDAG CR
CELLS	5.63 + 4.42	+ 243.92	6.35%	+ 76.29	2.16%
C.ELEGANS	248.13 + 186.24	+ 13888.0	12.7%	+ 2099.91	2.25%
MOUSE	1032.11 + 786.44	+ 61092.74	41.95%	+ 10276.93	8.06%

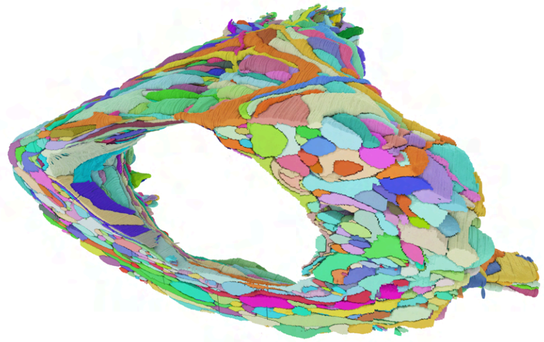


CELLS (raw 4.0 GB)



$1000 \times 1000 \times 1000 = 1.0 \cdot 10^9$
15.39% solid
118251 labels

C.ELEGANS (raw 112.8 GB)



$661 \times 11044 \times 15448 = 112.77 \cdot 10^9$
5.45% solid
234 labels

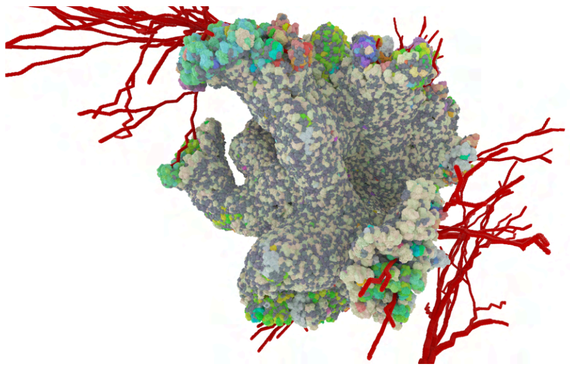
MOUSE (raw 150.0 GB)

$5446 \times 8381 \times 3286 = 149.98 \cdot 10^9$
15.57% solid
96 labels

Rendering Performance [ms] using finest (coarsest) LOD

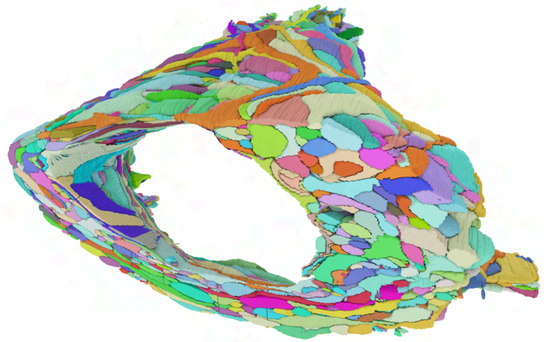
		SVOs	w/ occupancy field SVDAG	w/o occupancy field SVDAG
1 bounce	CELLS	8.19 (0.4)	2.95 (0.51)	5.87 (0.42)
	C.ELEGANS	–	5.61 (0.55)	12.75 (0.47)
	MOUSE	–	126.73 (0.98)	134.72 (0.78)
32 bnc.	CELLS	11.42 (0.81)	5.31 (1.02)	8.66 (0.85)
	C.ELEGANS	–	9.25 (0.98)	18.72 (0.83)
	MOUSE	–	189.54 (2.0)	203.65 (1.67)

CELLS (raw 4.0 GB)



$1000 \times 1000 \times 1000 = 1.0 \cdot 10^9$
15.39% solid
118251 labels

C.ELEGANS (raw 112.8 GB)



$661 \times 11044 \times 15448 = 112.77 \cdot 10^9$
5.45% solid
234 labels

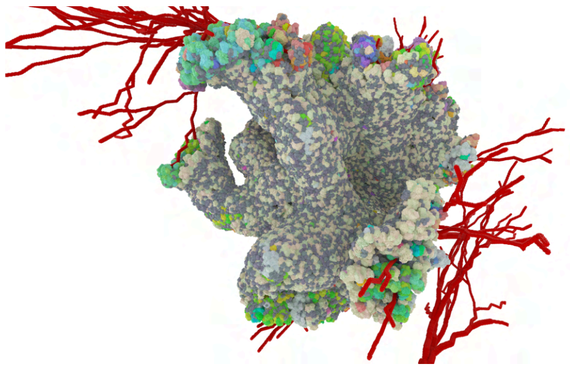
MOUSE (raw 150.0 GB)

$5446 \times 8381 \times 3286 = 149.98 \cdot 10^9$
15.57% solid
96 labels

Rendering Performance [ms] using finest (coarsest) LOD

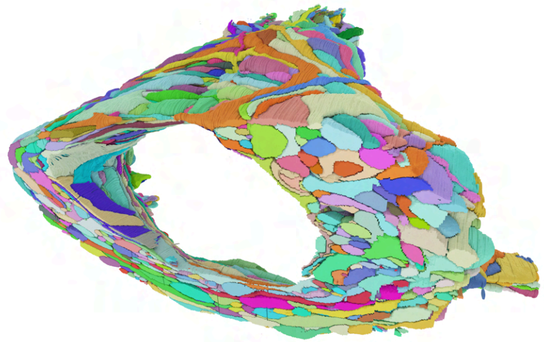
		SVOs	w/ occupancy field SVDAG	w/o occupancy field SVDAG
1 bounce	CELLS	8.19 (0.4)	2.95 (0.51)	5.87 (0.42)
	C.ELEGANS	–	5.61 (0.55)	12.75 (0.47)
	MOUSE	–	126.73 (0.98)	134.72 (0.78)
32 bnc.	CELLS	11.42 (0.81)	5.31 (1.02)	8.66 (0.85)
	C.ELEGANS	–	9.25 (0.98)	18.72 (0.83)
	MOUSE	–	189.54 (2.0)	203.65 (1.67)

CELLS (raw 4.0 GB)



$1000 \times 1000 \times 1000 = 1.0 \cdot 10^9$
15.39% solid
118251 labels

C.ELEGANS (raw 112.8 GB)



$661 \times 11044 \times 15448 = 112.77 \cdot 10^9$
5.45% solid
234 labels

MOUSE (raw 150.0 GB)

$5446 \times 8381 \times 3286 = 149.98 \cdot 10^9$
15.57% solid
96 labels

Compression Timings

	CELLS	C.ELEGANS	MOUSE
SVOS (AABB)			
CPU [s]	5.97	905.26	4127.16
SVDAGS (TYPE)			
CPU [s]	2.24	117.92	683.45
SVDAG (SHARED)			
CPU [ms]	408.49	33089.85	147721.0
GPU [ms]	101.96	2865.33	–

Table 5: Timings for each step to compress segmentation volumes: Computing AABBs and their SVOs from lists of solid voxel positions per label, merging them to SVDAGS (TYPE), and subsequently merging these partial SVDAGs to one SVDAG (SHARED). We provide a GPU implementation of the last step for fast merging of visualized types during rendering start-up or runtime.

CSGV [Piochowiak*2023] (memory)

	blocks + BVH + SVOs		blocks + BVH + SVDAG	
	mem. [MB]	CR	mem. [MB]	CR
CELLS	253.97	6.35%	86.34	2.16%
C.ELEGANS	14322.37	12.7%	2534.28	2.25%
MOUSE	62911.29	41.95%	12095.48	8.06%

CSGV mem. [MB]		CSGV CR	
15.46	(+544)	0.39%	(13.99%)
736.981	(+768)	0.65%	(1.33%)
1910.17	(+768)	1.27%	(1.79%)

Table 2: Memory and compression rate of CSGV [PD24]. Theoretical compression rate when including a GPU cache size to decompress visible regions to fitting levels of detail during rendering in ().

CSGV [Piochowiak*2023] (rendering performance)

		SVOs [ms]		SVDAG [ms]		CSGV
		finest LOD	coarsest LOD	finest LOD	coarsest LOD	
1 bounce	CELLS	8.19	0.4	2.95	0.51	23.21 (2.28)
	C.ELEGANS	–	–	5.61	0.55	87.18 (16.56)
	MOUSE	–	–	126.73	0.98	124.89 (8.47)
32 bnc.	CELLS	11.42	0.81	5.31	1.02	25.64 (2.29)
	C.ELEGANS	–	–	9.25	0.98	97.63 (17.06)
	MOUSE	–	–	189.54	2.0	136.16 (10.55)