



CONFERENCE 4 – 7 December 2018
EXHIBITION 5 – 7 December 2018
Tokyo International Forum, Japan
SA2018.SIGGRAPH.ORG

Sponsored by

GPU-Based Large-Scale Scientific Visualization

Johanna Beyer, Harvard University

Markus Hadwiger, KAUST

Course Website:

<http://johanna-b.github.io/LargeSciVis2018/index.html>





COURSE OVERVIEW - TOPICS

1. Introduction to scalable volume visualization
 - Focus on volume data
 - General scalability and out-of-core techniques
2. Scalable GPU volume rendering
 - Virtual texturing
 - GPU virtual memory architectures
3. Ray-guided volume rendering
 - Visibility-driven data processing
 - Empty-space skipping
4. Display-Aware visualization and data processing



COURSE OVERVIEW - MATERIAL

Course webpage (updated material):

<http://johanna-b.github.io/LargeSciVis2018/index.html>

State-of-the-Art in GPU-Based Large-Scale Volume Visualization

[J. Beyer, M. Hadwiger, H. Pfister; Computer Graphics Forum, 2015]

<https://dl.acm.org/citation.cfm?id=3071497>



COURSE OVERVIEW - SCHEDULE

- Part 1 – Introduction & Basics of Scalable Volume Visualization
Markus Hadwiger [60 min]
- Part 2 – Scalable Volume Visualization Architectures
Johanna Beyer [45 min]
- Break



COURSE OVERVIEW - SCHEDULE

- Part 3 – GPU-Based Ray-Guided Volume Rendering & Empty Space Skipping
Johanna Beyer [60 min]
- Part 4 – Display-Aware Visualization and Processing
Markus Hadwiger [45 min]
- Part 5 – Outlook and Summary
Johanna Beyer, Markus Hadwiger [15 min]



Part 1 -

Introduction & Basics of

Scalable Volume Visualization



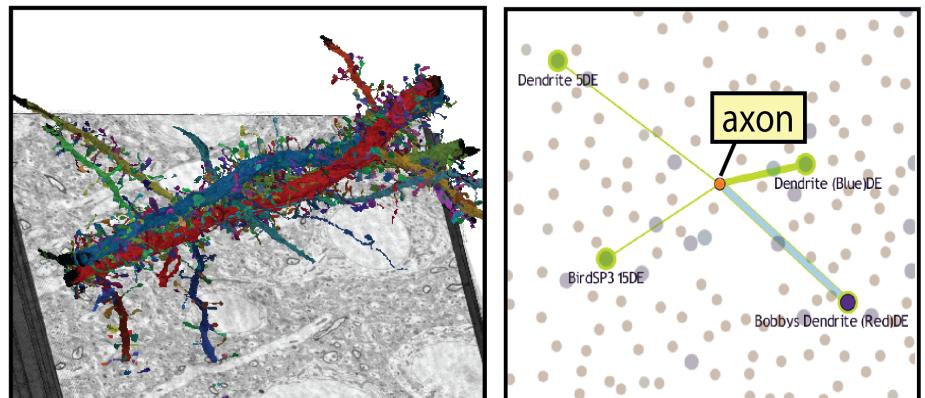
Motivation

BIG DATA

“In information technology, big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications. The challenges include capture, curation, storage, search, sharing, analysis, and visualization.”

‘Big Data’ on wikipedia.org

Our interest:
Very large 3D volume data



Example: Connectomics (neuroscience)



DATA-DRIVEN SCIENCE (E-SCIENCE)



MEDICINE

Digital Health Records



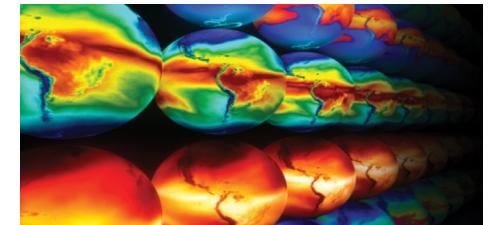
BIOLOGY

Connectomics



ENGINEERING

Large CFD Simulations



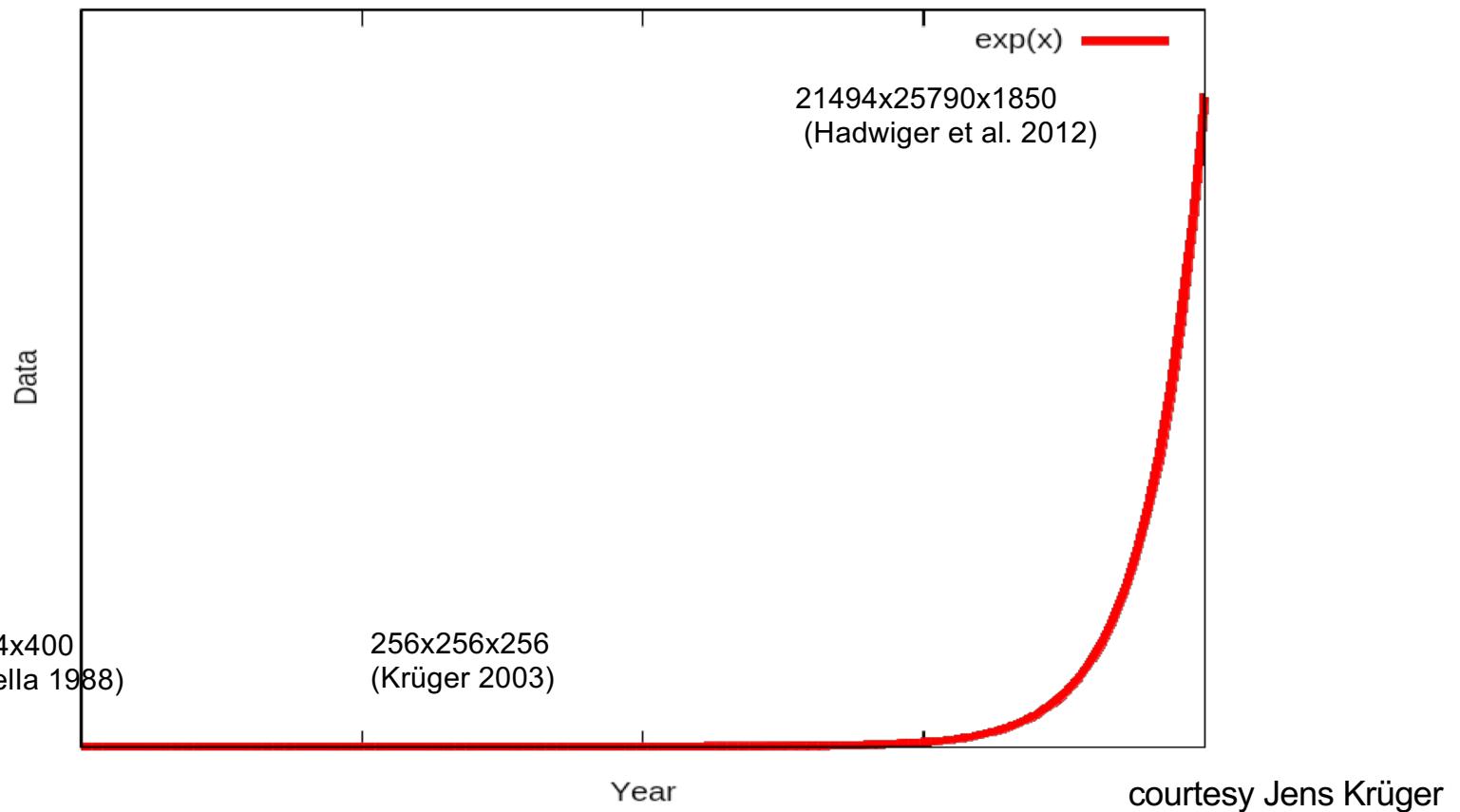
EARTH SCIENCES

Global Climate Models

courtesy Stefan Bruckner



VOLUME DATA GROWTH



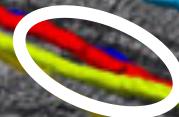


DATA SIZE EXAMPLES

year	paper	data set size	comments
2002	Guthe et al.	512 x 512 x 999 (500 MB) 2,048 x 1,216 x 1,877 (4.4 GB)	multi-pass, wavelet compression, streaming from disk
2003	Krüger & Westermann	256 x 256 x 256 (32 MB)	single-pass ray-casting
2005	Hadwiger et al.	576 x 352 x 1,536 (594 MB)	single-pass ray-casting (bricked)
2006	Ljung	512 x 512 x 628 (314 MB) 512 x 512 x 3396 (1.7 GB)	single-pass ray-casting, multi-resolution
2008	Gobbetti et al.	2,048 x 1,024 x 1,080 (4.2 GB)	'ray-guided' ray-casting with occlusion queries
2009	Crassin et al.	8,192 x 8,192 x 8,192 (512 GB)	ray-guided ray-casting
2011	Engel	8,192 x 8,192 x 16,384 (1 TB)	ray-guided ray-casting
2012	Hadwiger et al.	18,000 x 18,000 x 304 (92 GB) 21,494 x 25,790 x 1,850 (955 GB)	ray-guided ray-casting visualization-driven system
2013	Fogal et al.	1,728 x 1,008 x 1,878 (12.2 GB) 8,192 x 8,192 x 8,192 (512 GB)	ray-guided ray-casting
2018	Beyer et al.	21,494 x 25,790 x 1,850 (955 GB) images + 10,747 x 12,895 x 1,850 (489 GB) segmentation	ray-guided ray-casting, empty space skipping

The Connectome

How is the Mammalian Brain Wired?



Daniel Berger, MIT

The Connectome

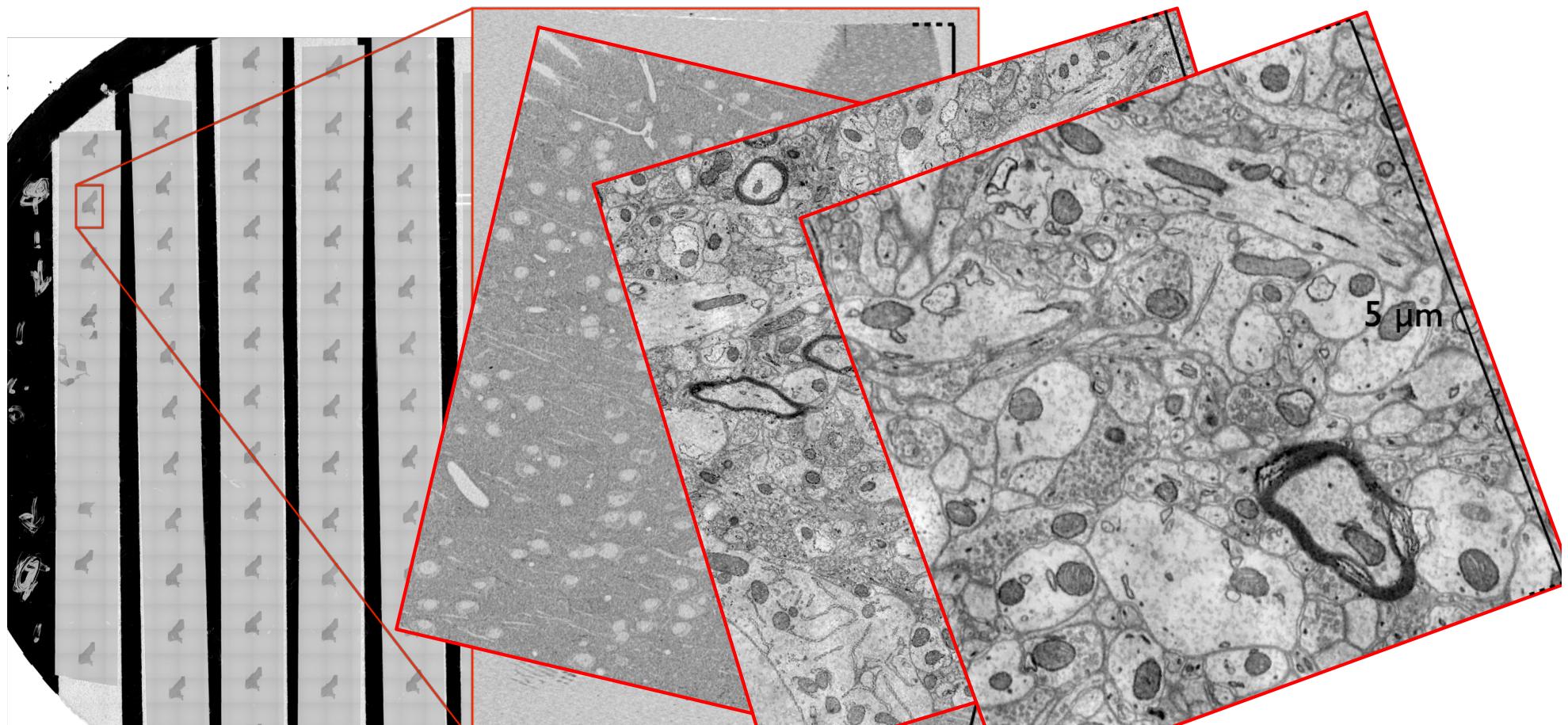
How is the Mammalian Brain Wired?

~60 μm^3
1 Teravoxel
 $21,500 \times 25,800 \times 1,850$

Bobby Kasthuri, Harvard



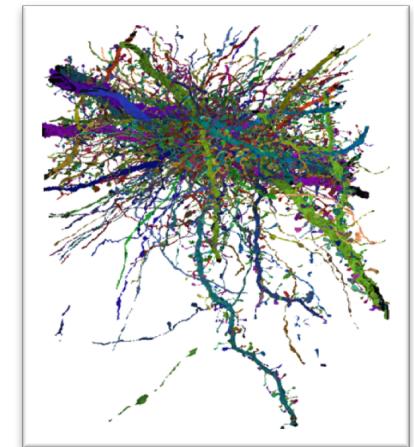
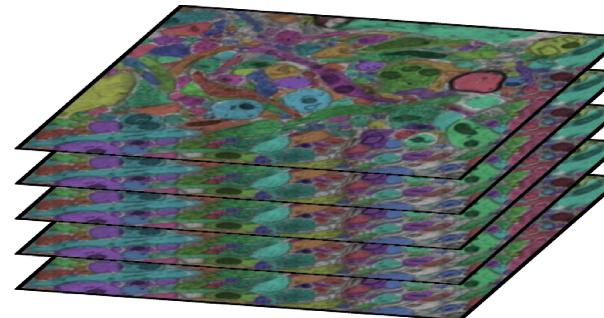
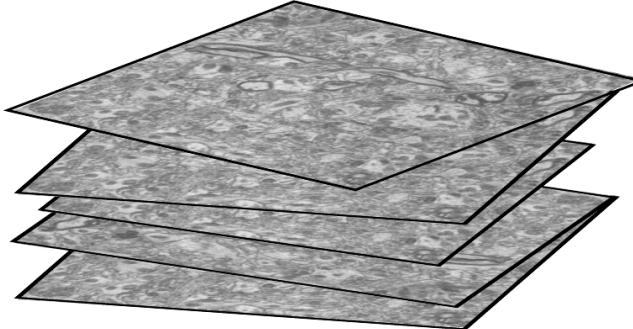
ELECTRON MICROSCOPY (EM) IMAGES





PETAVOXEL MICROSCOPY VOLUMES

- Huge amount of raw data (terabytes to petabytes)
- Takes months to years to scan, align, segment
- How to visualize and analyze this?



Time	Event Type	Location	Value
1	Start	0, 0, 0	0.0000
2	End	1, 0, 0	0.0000
3	Start	0, 1, 0	0.0000
4	End	1, 1, 0	0.0000
5	Start	0, 0, 1	0.0000
6	End	1, 0, 1	0.0000
7	Start	0, 1, 1	0.0000
8	End	1, 1, 1	0.0000
9	Start	1, 0, 0	0.0000
10	End	2, 0, 0	0.0000
11	Start	1, 1, 0	0.0000
12	End	2, 1, 0	0.0000
13	Start	1, 0, 1	0.0000
14	End	2, 0, 1	0.0000
15	Start	1, 1, 1	0.0000
16	End	2, 1, 1	0.0000
17	Start	2, 0, 0	0.0000
18	End	3, 0, 0	0.0000
19	Start	2, 1, 0	0.0000
20	End	3, 1, 0	0.0000
21	Start	2, 0, 1	0.0000
22	End	3, 0, 1	0.0000
23	Start	2, 1, 1	0.0000
24	End	3, 1, 1	0.0000
25	Start	3, 0, 0	0.0000
26	End	4, 0, 0	0.0000
27	Start	3, 1, 0	0.0000
28	End	4, 1, 0	0.0000
29	Start	3, 0, 1	0.0000
30	End	4, 0, 1	0.0000
31	Start	3, 1, 1	0.0000
32	End	4, 1, 1	0.0000
33	Start	4, 0, 0	0.0000
34	End	5, 0, 0	0.0000
35	Start	4, 1, 0	0.0000
36	End	5, 1, 0	0.0000
37	Start	4, 0, 1	0.0000
38	End	5, 0, 1	0.0000
39	Start	4, 1, 1	0.0000
40	End	5, 1, 1	0.0000
41	Start	5, 0, 0	0.0000
42	End	6, 0, 0	0.0000
43	Start	5, 1, 0	0.0000
44	End	6, 1, 0	0.0000
45	Start	5, 0, 1	0.0000
46	End	6, 0, 1	0.0000
47	Start	5, 1, 1	0.0000
48	End	6, 1, 1	0.0000
49	Start	6, 0, 0	0.0000
50	End	7, 0, 0	0.0000
51	Start	6, 1, 0	0.0000
52	End	7, 1, 0	0.0000
53	Start	6, 0, 1	0.0000
54	End	7, 0, 1	0.0000
55	Start	6, 1, 1	0.0000
56	End	7, 1, 1	0.0000
57	Start	7, 0, 0	0.0000
58	End	8, 0, 0	0.0000
59	Start	7, 1, 0	0.0000
60	End	8, 1, 0	0.0000
61	Start	7, 0, 1	0.0000
62	End	8, 0, 1	0.0000
63	Start	7, 1, 1	0.0000
64	End	8, 1, 1	0.0000
65	Start	8, 0, 0	0.0000
66	End	9, 0, 0	0.0000
67	Start	8, 1, 0	0.0000
68	End	9, 1, 0	0.0000
69	Start	8, 0, 1	0.0000
70	End	9, 0, 1	0.0000
71	Start	8, 1, 1	0.0000
72	End	9, 1, 1	0.0000
73	Start	9, 0, 0	0.0000
74	End	10, 0, 0	0.0000
75	Start	9, 1, 0	0.0000
76	End	10, 1, 0	0.0000
77	Start	9, 0, 1	0.0000
78	End	10, 0, 1	0.0000
79	Start	9, 1, 1	0.0000
80	End	10, 1, 1	0.0000
81	Start	10, 0, 0	0.0000
82	End	11, 0, 0	0.0000
83	Start	10, 1, 0	0.0000
84	End	11, 1, 0	0.0000
85	Start	10, 0, 1	0.0000
86	End	11, 0, 1	0.0000
87	Start	10, 1, 1	0.0000
88	End	11, 1, 1	0.0000
89	Start	11, 0, 0	0.0000
90	End	12, 0, 0	0.0000
91	Start	11, 1, 0	0.0000
92	End	12, 1, 0	0.0000
93	Start	11, 0, 1	0.0000
94	End	12, 0, 1	0.0000
95	Start	11, 1, 1	0.0000
96	End	12, 1, 1	0.0000
97	Start	12, 0, 0	0.0000
98	End	13, 0, 0	0.0000
99	Start	12, 1, 0	0.0000
100	End	13, 1, 0	0.0000
101	Start	12, 0, 1	0.0000
102	End	13, 0, 1	0.0000
103	Start	12, 1, 1	0.0000
104	End	13, 1, 1	0.0000
105	Start	13, 0, 0	0.0000
106	End	14, 0, 0	0.0000
107	Start	13, 1, 0	0.0000
108	End	14, 1, 0	0.0000
109	Start	13, 0, 1	0.0000
110	End	14, 0, 1	0.0000
111	Start	13, 1, 1	0.0000
112	End	14, 1, 1	0.0000
113	Start	14, 0, 0	0.0000
114	End	15, 0, 0	0.0000
115	Start	14, 1, 0	0.0000
116	End	15, 1, 0	0.0000
117	Start	14, 0, 1	0.0000
118	End	15, 0, 1	0.0000
119	Start	14, 1, 1	0.0000
120	End	15, 1, 1	0.0000
121	Start	15, 0, 0	0.0000
122	End	16, 0, 0	0.0000
123	Start	15, 1, 0	0.0000
124	End	16, 1, 0	0.0000
125	Start	15, 0, 1	0.0000
126	End	16, 0, 1	0.0000
127	Start	15, 1, 1	0.0000
128	End	16, 1, 1	0.0000
129	Start	16, 0, 0	0.0000
130	End	17, 0, 0	0.0000
131	Start	16, 1, 0	0.0000
132	End	17, 1, 0	0.0000
133	Start	16, 0, 1	0.0000
134	End	17, 0, 1	0.0000
135	Start	16, 1, 1	0.0000
136	End	17, 1, 1	0.0000
137	Start	17, 0, 0	0.0000
138	End	18, 0, 0	0.0000
139	Start	17, 1, 0	0.0000
140	End	18, 1, 0	0.0000
141	Start	17, 0, 1	0.0000
142	End	18, 0, 1	0.0000
143	Start	17, 1, 1	0.0000
144	End	18, 1, 1	0.0000
145	Start	18, 0, 0	0.0000
146	End	19, 0, 0	0.0000
147	Start	18, 1, 0	0.0000
148	End	19, 1, 0	0.0000
149	Start	18, 0, 1	0.0000
150	End	19, 0, 1	0.0000
151	Start	18, 1, 1	0.0000
152	End	19, 1, 1	0.0000
153	Start	19, 0, 0	0.0000
154	End	20, 0, 0	0.0000
155	Start	19, 1, 0	0.0000
156	End	20, 1, 0	0.0000
157	Start	19, 0, 1	0.0000
158	End	20, 0, 1	0.0000
159	Start	19, 1, 1	0.0000
160	End	20, 1, 1	0.0000
161	Start	20, 0, 0	0.0000
162	End	21, 0, 0	0.0000
163	Start	20, 1, 0	0.0000
164	End	21, 1, 0	0.0000
165	Start	20, 0, 1	0.0000
166	End	21, 0, 1	0.0000
167	Start	20, 1, 1	0.0000
168	End	21, 1, 1	0.0000
169	Start	21, 0, 0	0.0000
170	End	22, 0, 0	0.0000
171	Start	21, 1, 0	0.0000
172	End	22, 1, 0	0.0000
173	Start	21, 0, 1	0.0000
174	End	22, 0, 1	0.0000
175	Start	21, 1, 1	0.0000
176	End	22, 1, 1	0.0000
177	Start	22, 0, 0	0.0000
178	End	23, 0, 0	0.0000
179	Start	22, 1, 0	0.0000
180	End	23, 1, 0	0.0000
181	Start	22, 0, 1	0.0000
182	End	23, 0, 1	0.0000
183	Start	22, 1, 1	0.0000
184	End	23, 1, 1	0.0000
185	Start	23, 0, 0	0.0000
186	End	24, 0, 0	0.0000
187	Start	23, 1, 0	0.0000
188	End	24, 1, 0	0.0000
189	Start	23, 0, 1	0.0000
190	End	24, 0, 1	0.0000
191	Start	23, 1, 1	0.0000
192	End	24, 1, 1	0.0000
193	Start	24, 0, 0	0.0000
194	End	25, 0, 0	0.0000
195	Start	24, 1, 0	0.0000
196	End	25, 1, 0	0.0000
197	Start	24, 0, 1	0.0000
198	End	25, 0, 1	0.0000
199	Start	24, 1, 1	0.0000
200	End	25, 1, 1	0.0000
201	Start	25, 0, 0	0.0000
202	End	26, 0, 0	0.0000
203	Start	25, 1, 0	0.0000
204	End	26, 1, 0	0.0000
205	Start	25, 0, 1	0.0000
206	End	26, 0, 1	0.0000
207	Start	25, 1, 1	0.0000
208	End	26, 1, 1	0.0000
209	Start	26, 0, 0	0.0000
210	End	27, 0, 0	0.0000
211	Start	26, 1, 0	0.0000
212	End	27, 1, 0	0.0000
213	Start	26, 0, 1	0.0000
214	End	27, 0, 1	0.0000
215	Start	26, 1, 1	0.0000
216	End	27, 1, 1	0.0000
217	Start	27, 0, 0	0.0000
218	End	28, 0, 0	0.0000
219	Start	27, 1, 0	0.0000
220	End	28, 1, 0	0.0000
221	Start	27, 0, 1	0.0000
222	End	28, 0, 1	0.0000
223	Start	27, 1, 1	0.0000
224	End	28, 1, 1	0.0000
225	Start	28, 0, 0	0.0000
226	End	29, 0, 0	0.0000
227	Start	28, 1, 0	0.0000
228	End	29, 1, 0	0.0000
229	Start	28, 0, 1	0.0000
230	End	29, 0, 1	0.0000
231	Start	28, 1, 1	0.0000
232	End	29, 1, 1	0.0000
233	Start	29, 0, 0	0.0000
234	End	30, 0, 0	0.0000
235	Start	29, 1, 0	0.0000
236	End	30, 1, 0	0.0000
237	Start	29, 0, 1	0.0000
238	End	30, 0, 1	0.0000
239	Start	29, 1, 1	0.0000
240	End	30, 1, 1	0.0000
241	Start	30, 0, 0	0.0000
242	End	31, 0, 0	0.0000
243	Start	30, 1, 0	0.0000
244	End	31, 1, 0	0.0000
245	Start	30, 0, 1	0.0000
246	End	31, 0, 1	0.0000
247	Start	30, 1, 1	0.0000
248	End	31, 1, 1	0.0000
249	Start	31, 0, 0	0.0000
250	End	32, 0, 0	0.0000
251	Start	31, 1, 0	0.0000
252	End	32, 1, 0	0.0000
253	Start	31, 0, 1	0.0000
254	End	32, 0, 1	0.0000
255	Start	31, 1, 1	0.0000
256	End	32, 1, 1	0.0000
257	Start	32, 0, 0	0.0000
258	End	33, 0	



COURSE SCOPE

Course focus

- (Single) GPUs in standard workstations
- Scalar volume data; single time step
- But a lot applies to more general settings...

Orthogonal techniques (will not cover details)

- Parallel and distributed rendering, clusters, supercomputers, ...
- Compression



RELATED BOOKS AND SURVEYS

Books

- Real-Time Volume Graphics, Engel et al., 2006
- High-Performance Visualization, Bethel et al., 2012

Surveys

- GPU-Based Large-Scale Volume Visualization: Beyer et al. '15
- Parallel Visualization: Wittenbrink '98, Bartz et al. '00, Zhang et al. '05
- Real Time Interactive Massive Model Visualization: Kasik et al. '06
- Vis and Visual Analysis of Multifaceted Scientific Data: Kehrer and Hauser '13
- Compressed GPU-Based Volume Rendering: Rodriguez et al. '14
- Web-based Visualization: Mwalongo et al. '16
- In-Situ Methods, Infrastructures, and Applications in High Performance Computing: Bauer et al. '16

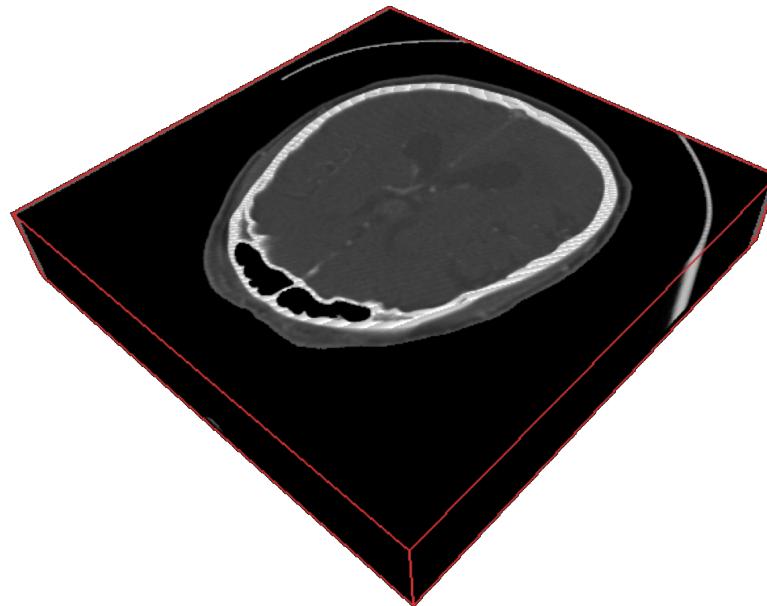


Fundamentals



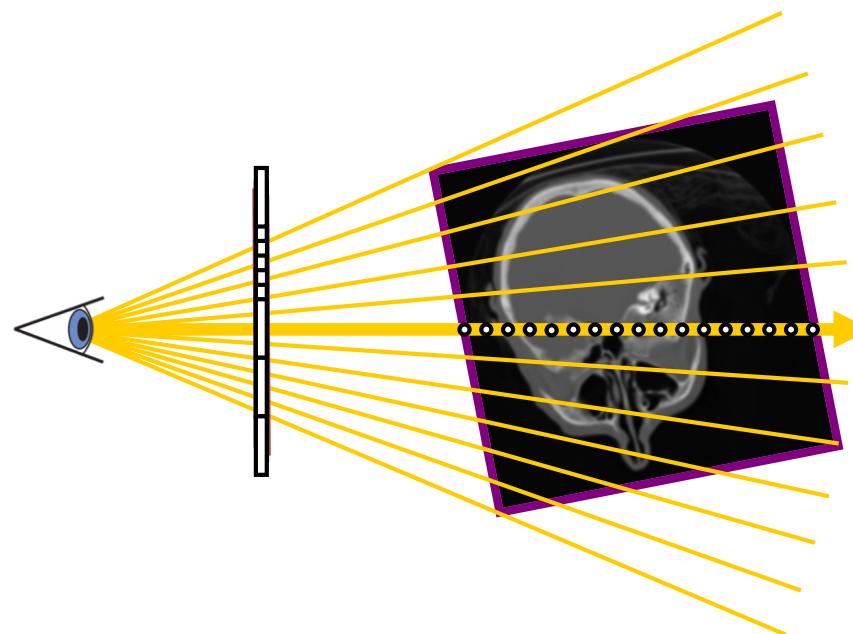
VOLUME RENDERING (1)

Assign optical properties (color, opacity) via *transfer function*



VOLUME RENDERING (2)

Ray-casting





SCALABILITY

Traditional HPC, parallel rendering definitions

- Strong scaling (“more nodes are faster for same data”)
- Weak scaling (“more nodes allow larger data”)

Our interest/definition: output sensitivity

- Running time/storage proportional to size of output instead of input
 - Computational effort scales with visible data and screen resolution
 - Working set independent of original data size



SOME TERMINOLOGY

Output-sensitive algorithms

- Standard term in occlusion culling (of geometry)

Ray-guided volume rendering

- Determine working set via ray-casting
- Actual visibility; not approximate as in traditional occlusion culling

Visualization-driven pipeline

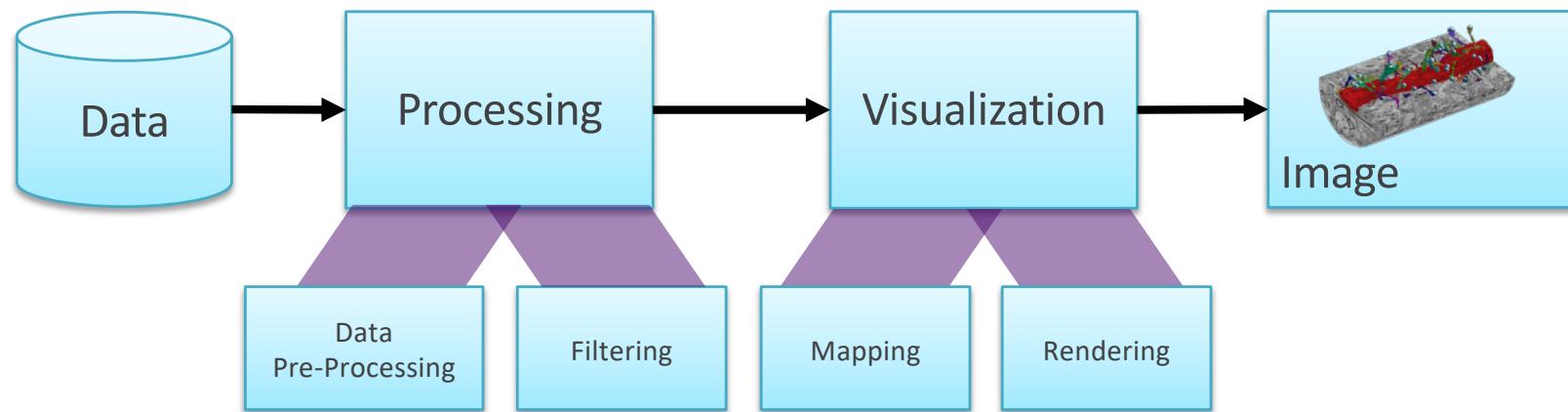
- Drive entire visualization pipeline (including processing) by actual on-screen visibility

Display-aware techniques

- Image processing, ... for current on-screen resolution

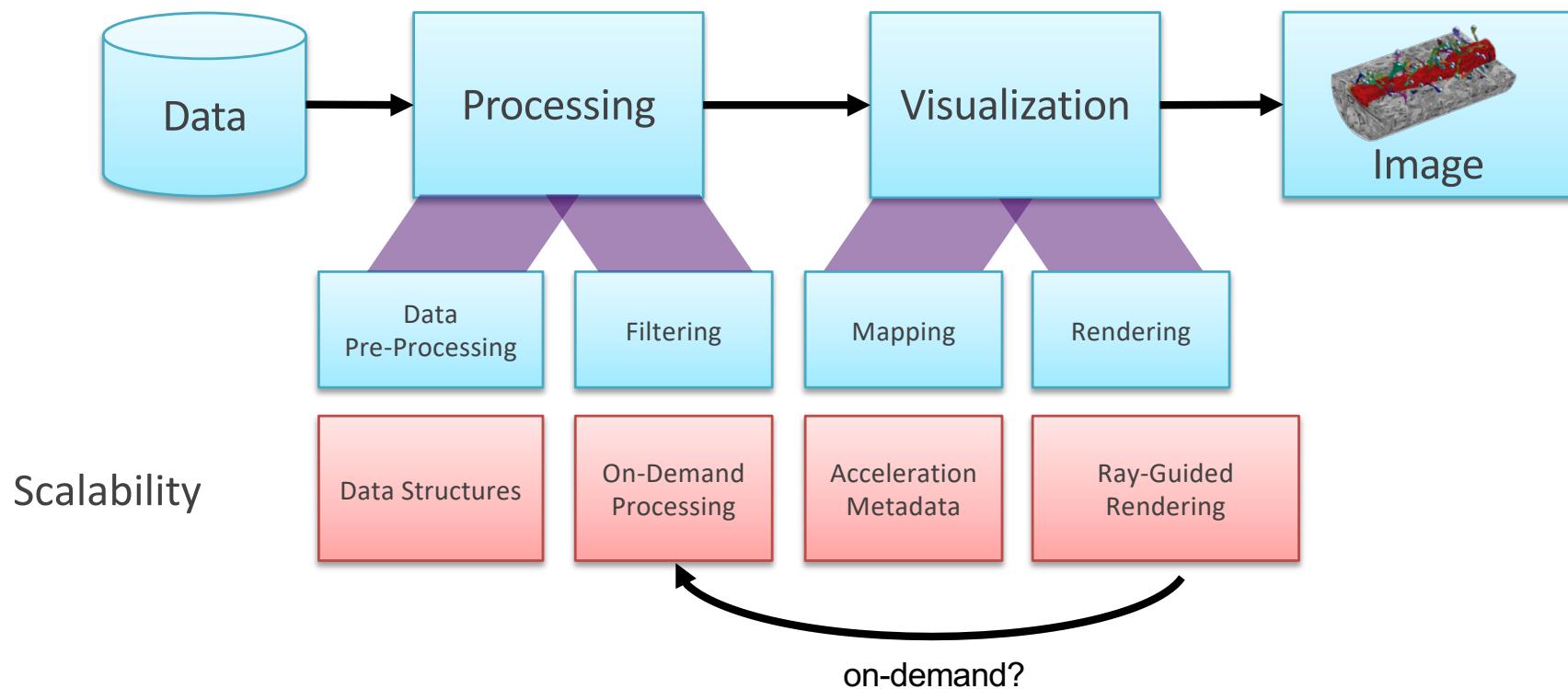


LARGE-SCALE VISUALIZATION PIPELINE





LARGE-SCALE VISUALIZATION PIPELINE





Basic Scalability Issues



SCALABILITY ISSUES

Scalability issues	Scalable method
Data representation and storage	Multi-resolution data structures
	Data layout, compression
Work/data partitioning	In-core/out-of-core
	Parallel, distributed
Work/data reduction	Pre-processing
	On-demand processing
	Streaming
	In-situ visualization
	Query-based visualization



SCALABILITY ISSUES

Scalability issues	Scalable method
Data representation and storage	Multi-resolution data structures Data layout, compression
Work/data partitioning	In-core/out-of-core Parallel, distributed
Work/data reduction	Pre-processing On-demand processing Streaming In-situ visualization Query-based visualization



DATA REPRESENTATIONS

Data structure	Acceleration	Out-of-Core	Multi-Resolution
Mipmaps	-	Clipmaps	Yes
Uniform bricking	Cull bricks (linear)	Working set (bricks)	No
Hierarch. bricking	Cull bricks (hierarch.)	Working set (bricks)	Bricked mipmap
Octrees	Hierarchical traversal	Working set (subtree)	Yes (interior nodes)

Additional issues

- Data layout (linear order, Z/Morton order, ...)
- Compression

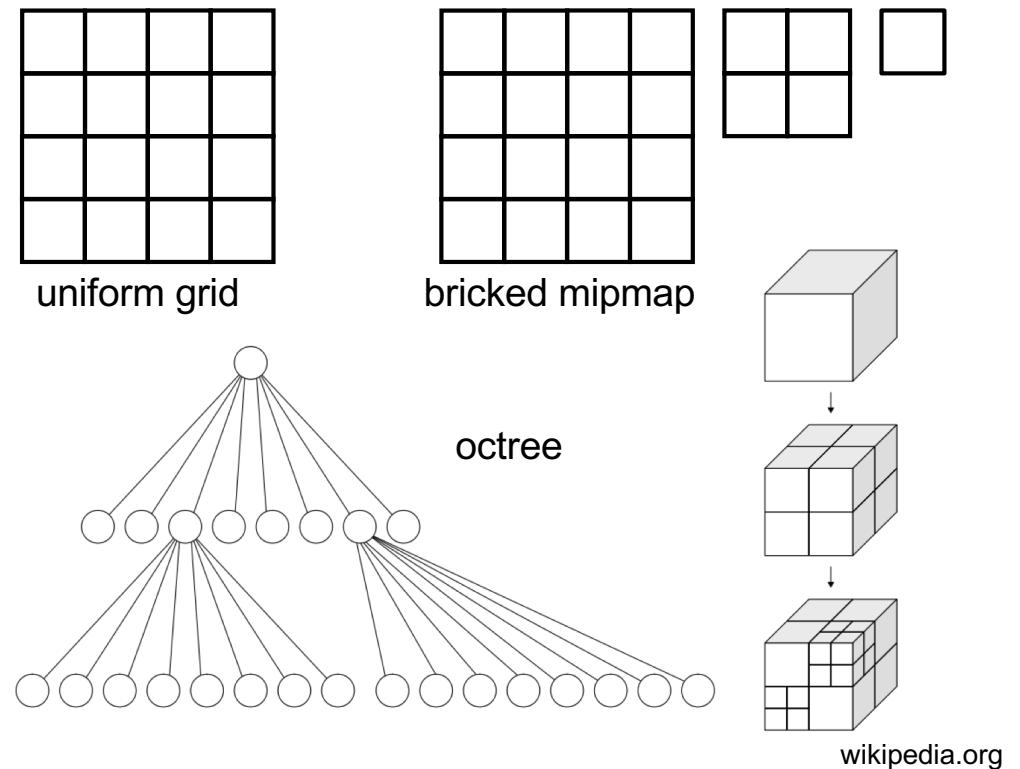
UNIFORM VS. HIERARCHICAL DECOMPOSITION

Grids

- Uniform or non-uniform

Hierarchical data structures

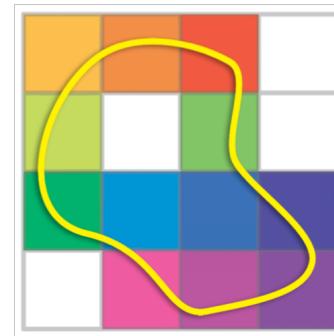
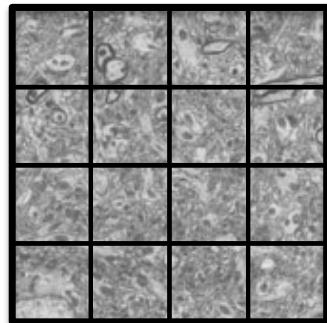
- Pyramid of uniform grids
 - Bricked 2D/3D mipmaps
- Tree structures
 - Quadtree, octree, kd-tree



BRICKING (1)

Object space (data) decomposition

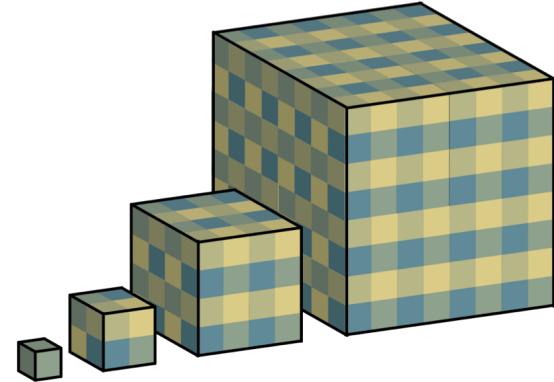
- Subdivide data domain into small bricks
- Re-orders data for spatial locality
- Each brick is now one unit (culling, paging, loading, ...)



BRICKING (2)

What brick size to use?

- Small bricks
 - + Good granularity:
Better culling efficiency, tighter working set, ...
 - More bricks to cull, more overhead for ghost voxels,
one rendering pass per brick is infeasible
- Traditional out-of-core volume rendering: **large** bricks (e.g., 256^3)
- Modern out-of-core volume rendering: **small** bricks (e.g., 32^3)
 - Task-dependent brick sizes
(small for rendering, large for disk/network storage)



Analysis of different brick sizes: [Fogal et al. 2013]

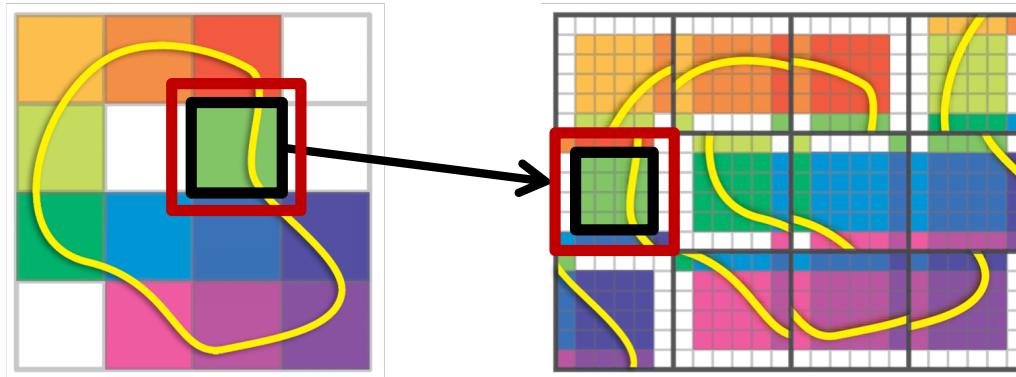
FILTERING AT BRICK BOUNDARIES

Duplicate voxels at border (“ghost” voxels)

- Need at least one voxel overlap
- Large overhead for small bricks

Otherwise costly filtering at brick boundary

- Except with hardware support: *sparse textures*





PRE-COMPUTE ALL BRICKS?

Pre-computation might take very long

- Brick on demand? Brick in streaming fashion (e.g., during scanning)?

Different brick sizes for different tasks (storage, rendering)?

- Re-brick to different size on demand?
- Dynamically fix up ghost voxels?

Can also mix 2D and 3D

- E.g., 2D tiling pre-computed, but compute 3D bricks on demand

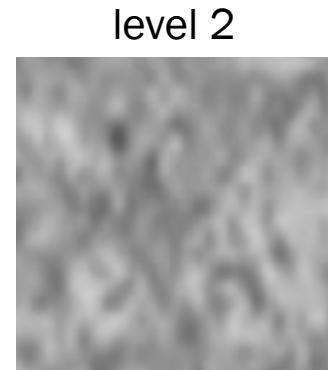
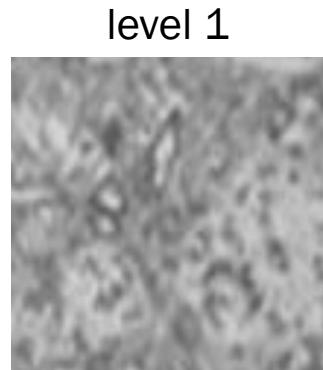
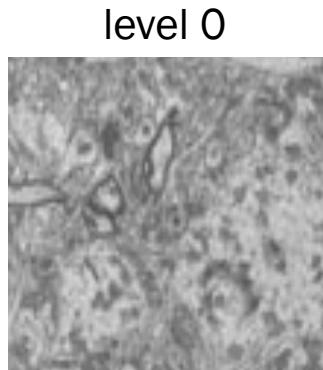
MULTI-RESOLUTION PYRAMIDS (1)

Collection of different resolution levels

- Standard: dyadic pyramids (2:1 resolution reduction)
- Can manually implement arbitrary reduction ratios

Mipmaps

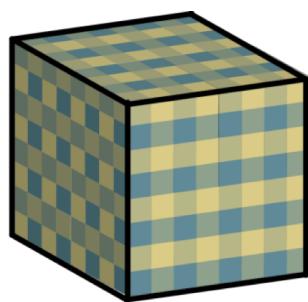
- Isotropic



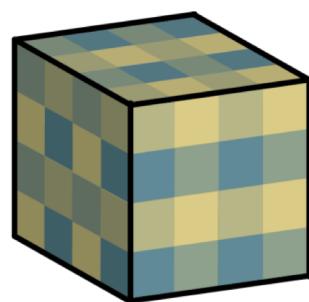
MULTI-RESOLUTION PYRAMIDS (2)

3D mipmaps

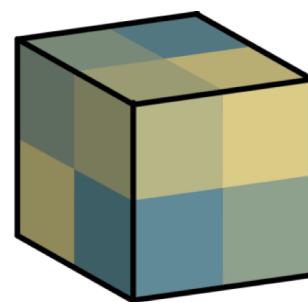
- Isotropic



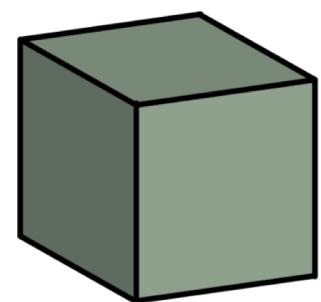
level 0
(8x8x8)



level 1
(4x4x4)



level 2
(2x2x2)

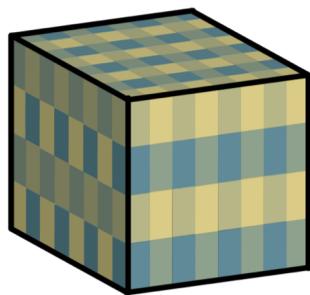


level 3
(1x1x1)

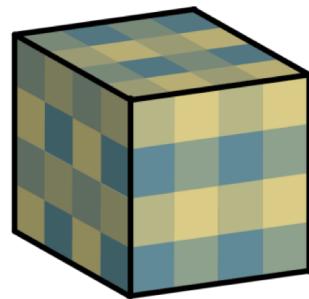
MULTI-RESOLUTION PYRAMIDS (3)

Scanned volume data are often anisotropic

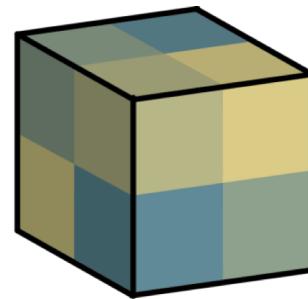
- Reduce resolution anisotropically until isotropy reached



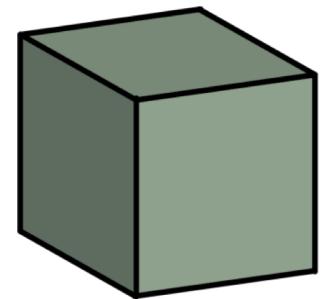
level 0
(8x8x4)



level 1
(4x4x4)



level 2
(2x2x2)

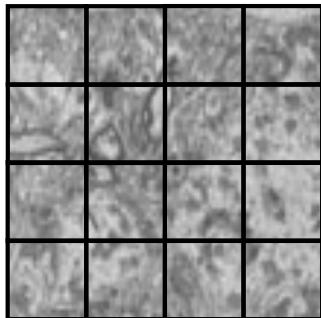


level 3
(1x1x1)

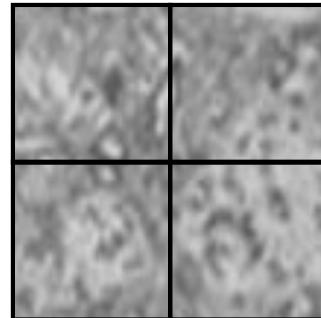
BRICKING MULTI-RESOLUTION PYRAMIDS (1)

Each level is bricked individually

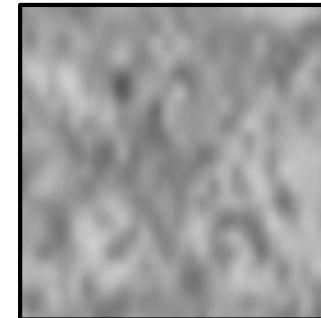
- Use same brick resolution (# voxels) in each level



level 0



level 1



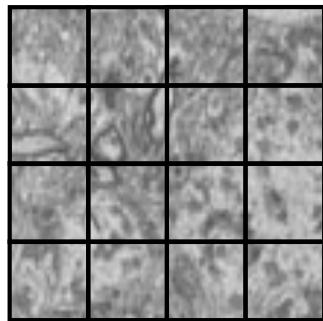
level 2

spatial
extent

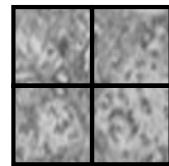
BRICKING MULTI-RESOLUTION PYRAMIDS (2)

Virtual memory: Each brick will be a “page”

- “Multi-resolution virtual memory”: every page lives in some resolution level



4x4 pages



2x2 pages



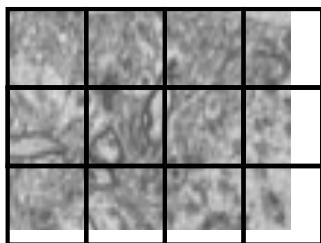
memory extent

1 page

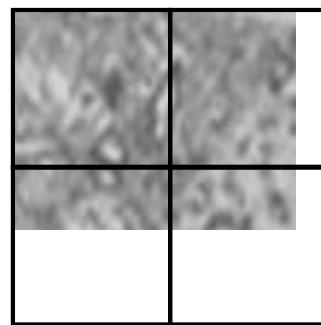
BRICKING MULTI-RESOLUTION PYRAMIDS (3)

Beware of aspect ratio and partially-filled pages

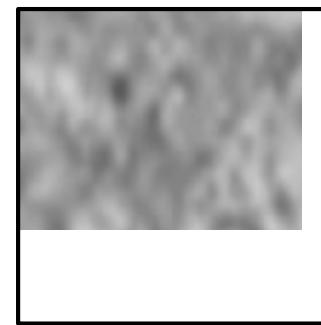
- Reduce total resolution in voxels; compute number of pages (ceil); iterate



4x3 pages



2x2 pages



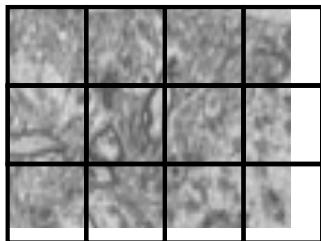
1 page

spatial
extent

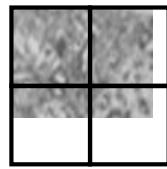
BRICKING MULTI-RESOLUTION PYRAMIDS (3)

Beware of aspect ratio and partially filled pages

- Reduce total resolution in voxels; compute number of pages (ceil); iterate



4x3 pages



2x2 pages



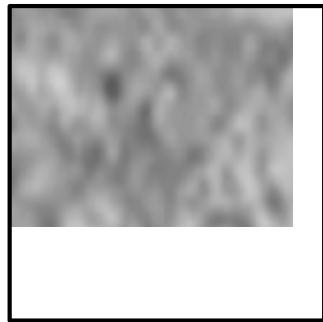
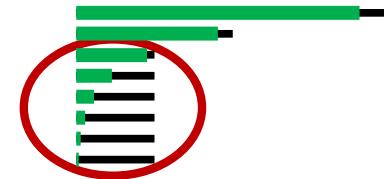
1 page

memory
extent

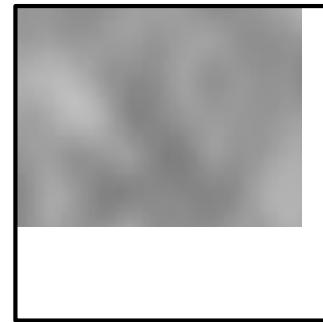
BRICKING MULTI-RESOLUTION PYRAMIDS (4)

Tail of pyramid

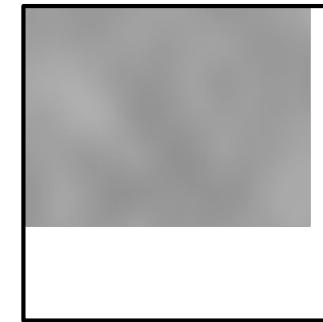
- Below size of single page; can cut off early



1 page



1 page



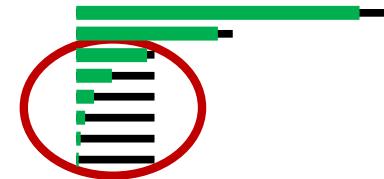
spatial
extent

1 page

BRICKING MULTI-RESOLUTION PYRAMIDS (4)

Tail of pyramid

- Below size of single page; can cut off early



memory
extent

- **GL_ARB_sparse_texture** treats tail as single unit of residency (implementation-dependent definition of tail !)

1 page

1 page

1 page

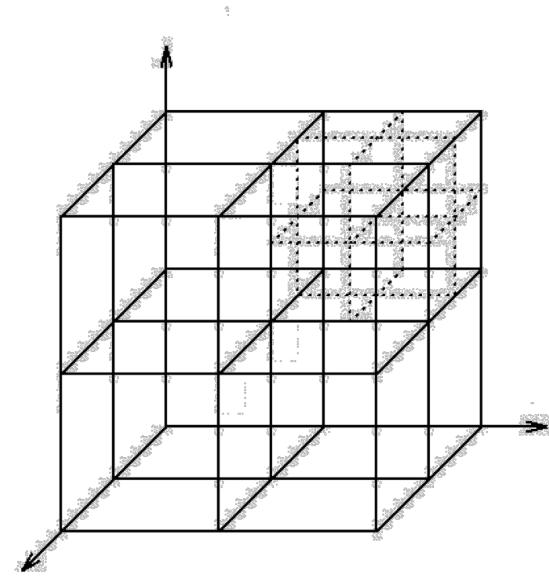
OCTREES FOR VOLUME RENDERING (1)

Multi-resolution

- Adapt resolution of data to screen resolution
 - Reduce aliasing
 - Limit amount of data needed

Acceleration

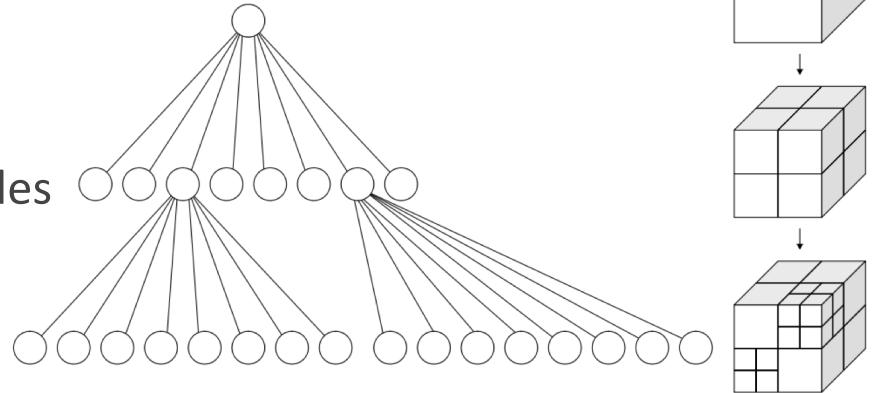
- Hierarchical empty space skipping
- Start traversal at root
 - (but different optimized traversal algorithms:
kd-restart, kd-shortstack, etc.)



OCTREES FOR VOLUME RENDERING (2)

Representation

- Full octree
 - Every octant in every resolution level
- Sparse octree
 - Do not store voxel data of empty nodes



wikipedia.org

Data structure

- Pointer-based
 - Parent node stores pointer(s) to children
- Pointerless
 - Array to index full octree directly



SCALABILITY ISSUES

Scalability issues	Scalable method
Data representation and storage	Multi-resolution data structures Data layout, compression
Work/data partitioning	In-core/out-of-core Parallel, distributed
Work/data reduction	Pre-processing On-demand processing Streaming In-situ visualization Query-based visualization



WORK/DATA PARTITIONING

- Out-of-core techniques
- Domain decomposition
- Parallel and distributed rendering

OUT-OF-CORE TECHNIQUES (1)

Data too large for GPU memory

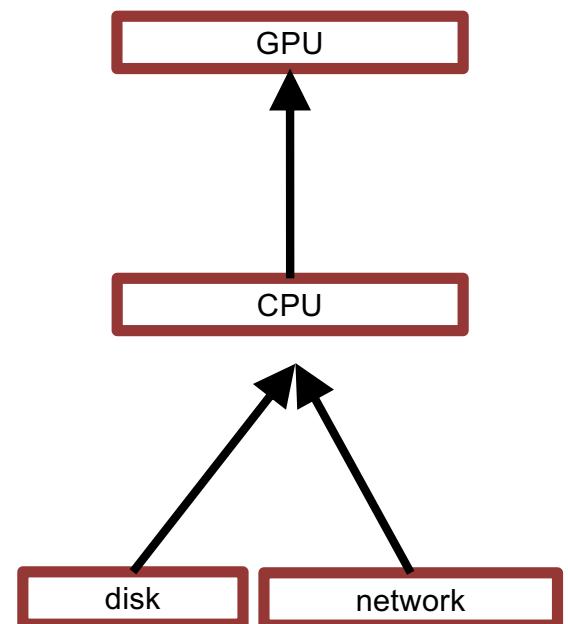
- Stream volume bricks from CPU to GPU on demand

Data too large for CPU memory

- Stream volume bricks from disk on demand

Data too large for local disk storage

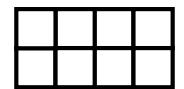
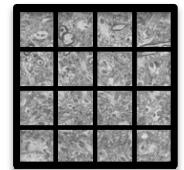
- Stream volume bricks from network storage



OUT-OF-CORE TECHNIQUES (2)

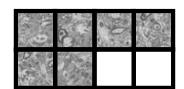
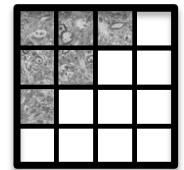
Preparation

- Subdivide spatial domain
 - May also be done “virtually”, i.e., data re-ordering may be delayed
- Allocate cache memory (e.g., large 3D cache texture)



Run-Time

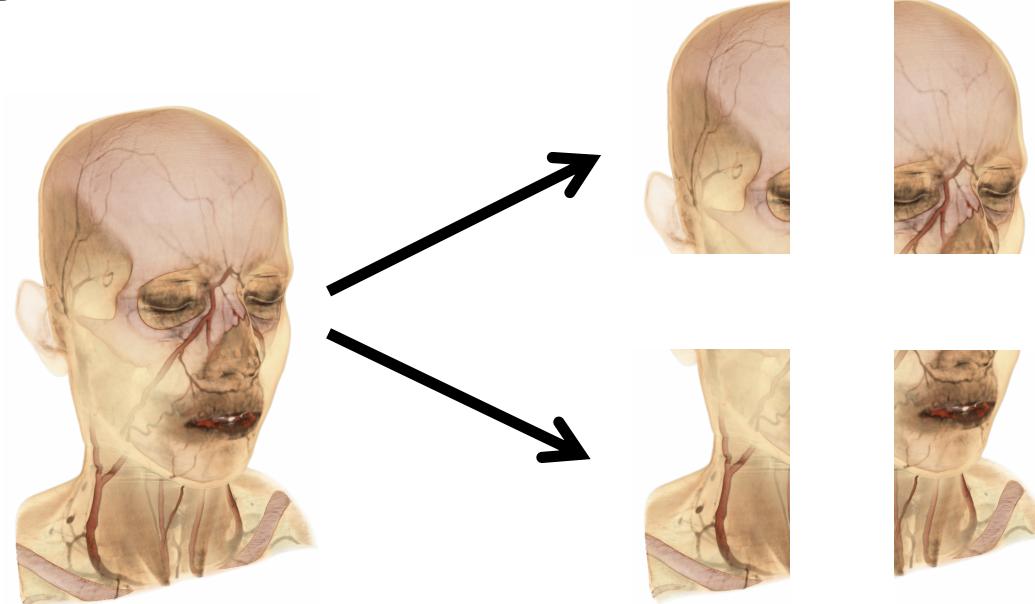
- Determine **working set**
- Page working set into cache memory
- Render from cache memory



DOMAIN DECOMPOSITION (1)

Subdivide image domain (image space)

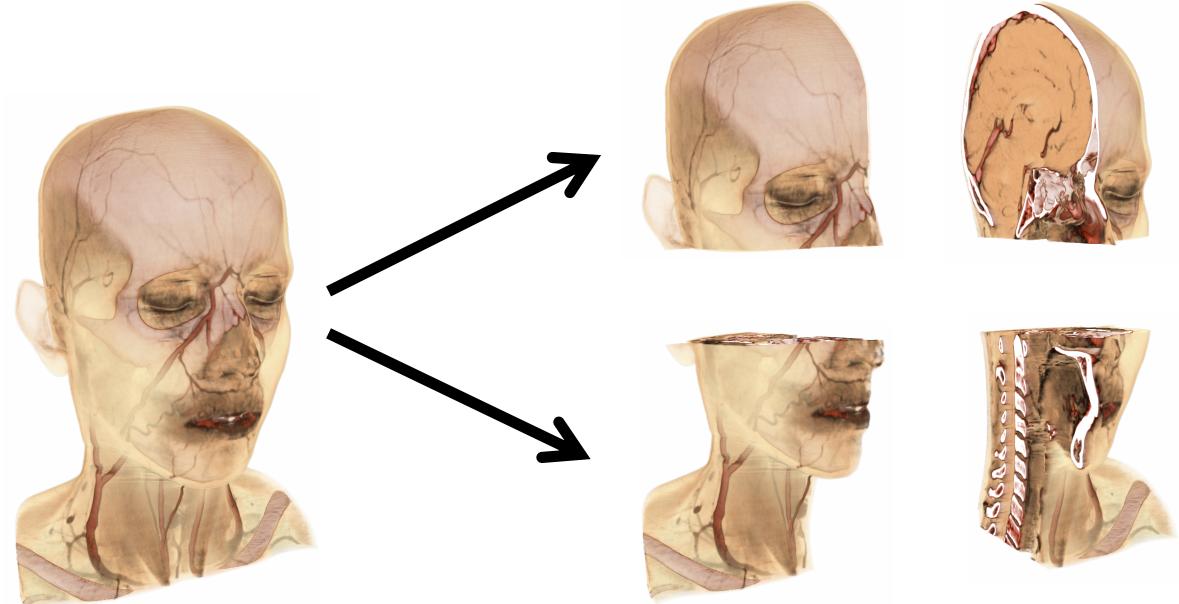
- “Sort-first rendering” [Molnar, 1994]
- View-dependent



DOMAIN DECOMPOSITION (2)

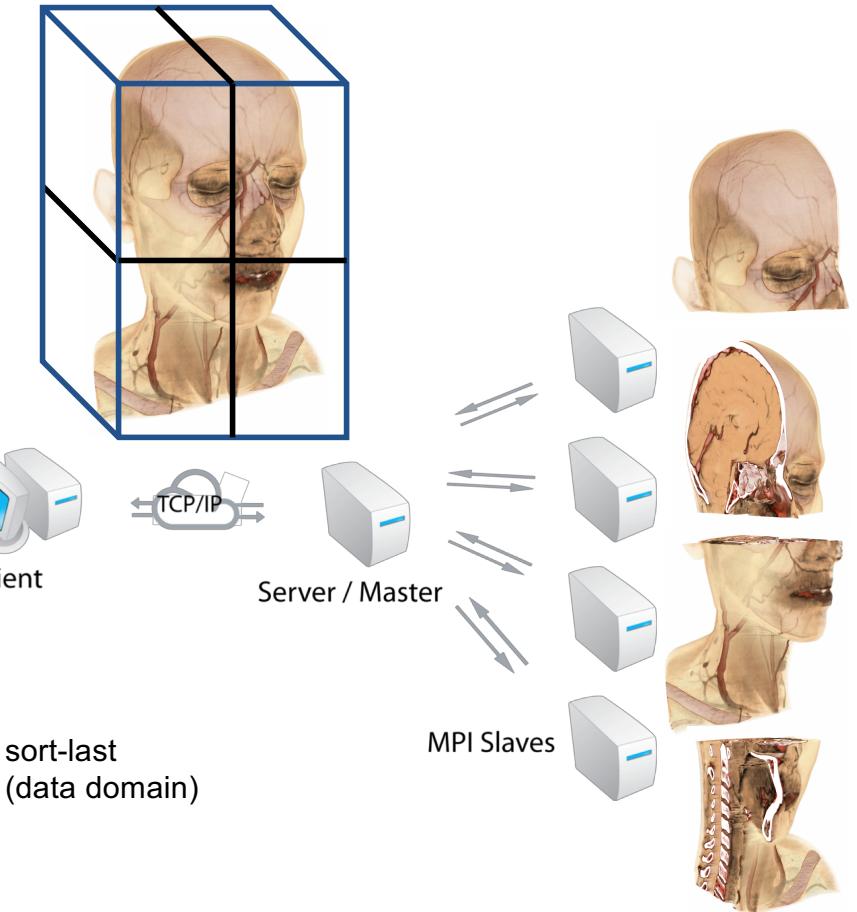
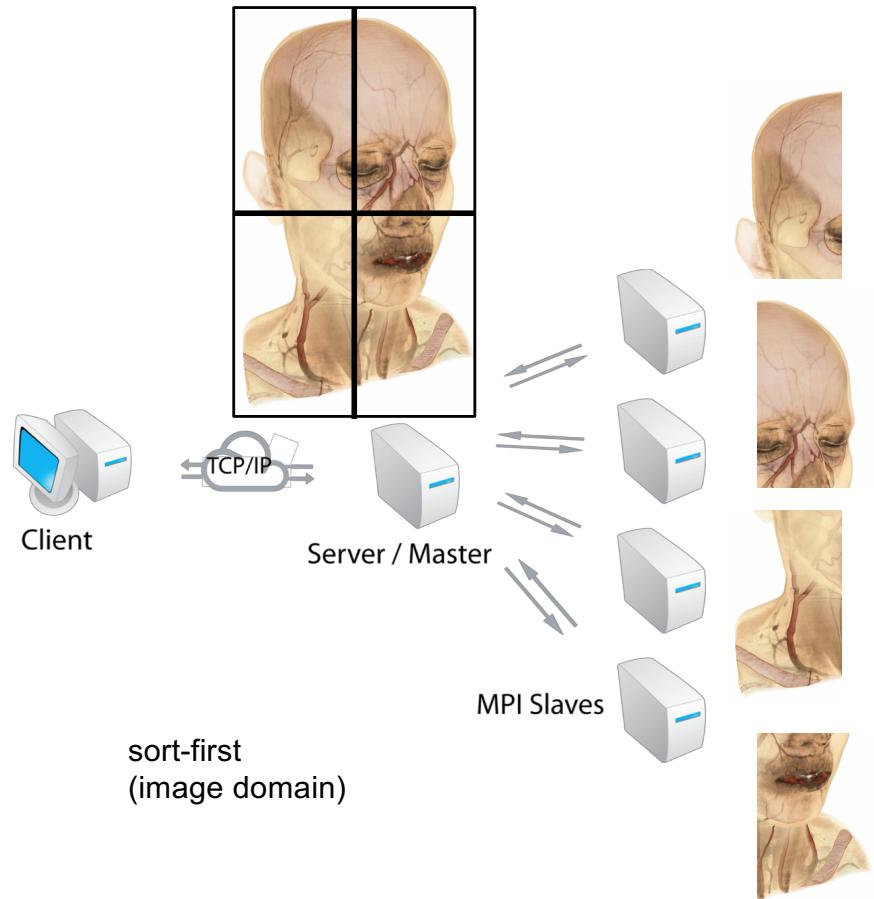
Subdivide data domain (object space)

- “Sort-last rendering” [Molnar, 1994]
- View-independent





SORT-FIRST VS. SORT-LAST





SCALABILITY ISSUES

Scalability issues	Scalable method
Data representation and storage	Multi-resolution data structures Data layout, compression
Work/data partitioning	In-core/out-of-core Parallel, distributed
Work/data reduction	Pre-processing On-demand processing Streaming In-situ visualization Query-based visualization



ON-DEMAND PROCESSING

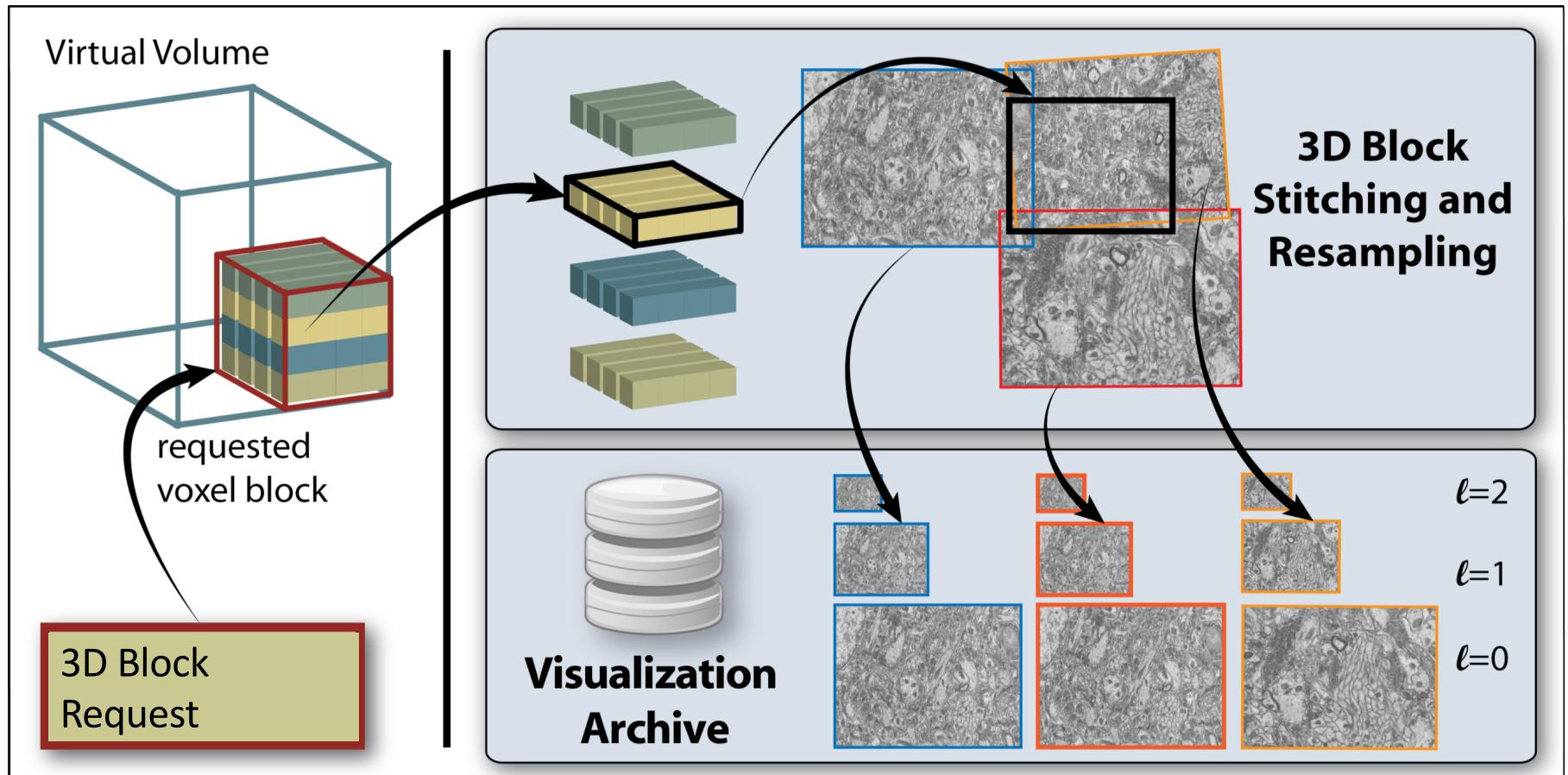
First determine what is visible / needed: working set

Then process only this working set

- Basic processing
 - Noise removal and edge detection
 - Registration and alignment
 - Segmentation, ...
- Basic data structure building
 - Construct pages/bricks/octree nodes only on demand?

EXAMPLE: 3D BRICK CONSTRUCTION FROM 2D EM STREAMS

[Hadwiger et al., IEEE Vis 2012]



EXAMPLE: DENOISING & EDGE ENHANCEMENT

Edge enhancement for EM data

Caching scheme

- Process only currently visible bricks
- Cache result for re-use

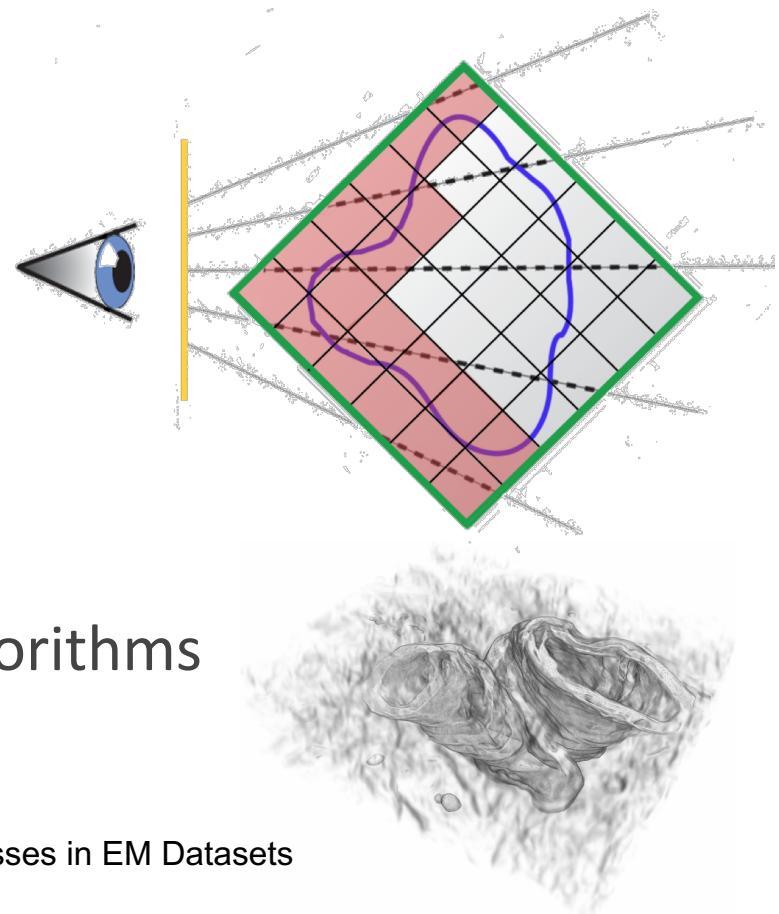
GPU Implementation

- CUDA and shared memory for fast computation

Different noise removal and filtering algorithms

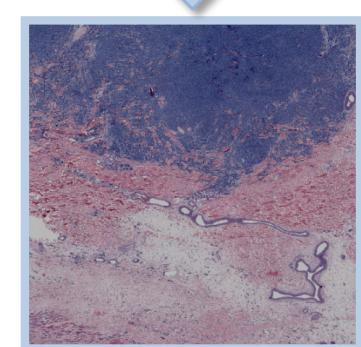
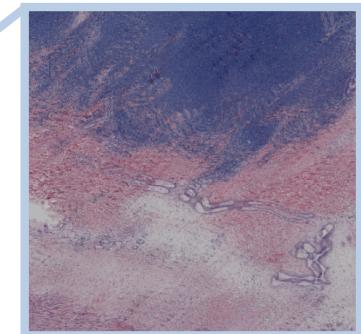
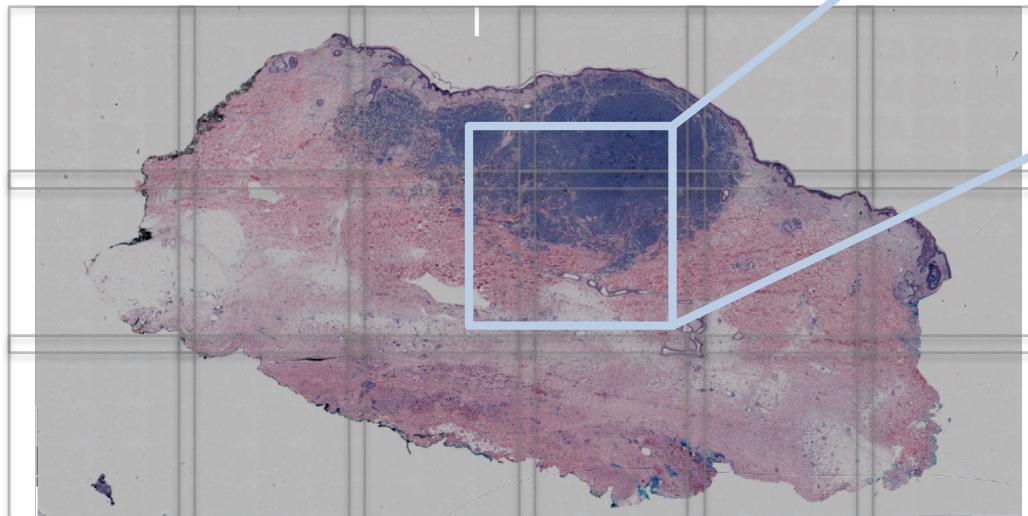
[Jeong et al., IEEE Vis 2009]

Scalable and Interactive Segmentation and Visualization of Neural Processes in EM Datasets



EXAMPLE: REGISTRATION & ALIGNMENT

Registration at screen/brick resolution



[Beyer et al., CG&A 2013]

Exploring the Connectome – Petascale Volume Visualization of Microscopy Data Streams



Questions?