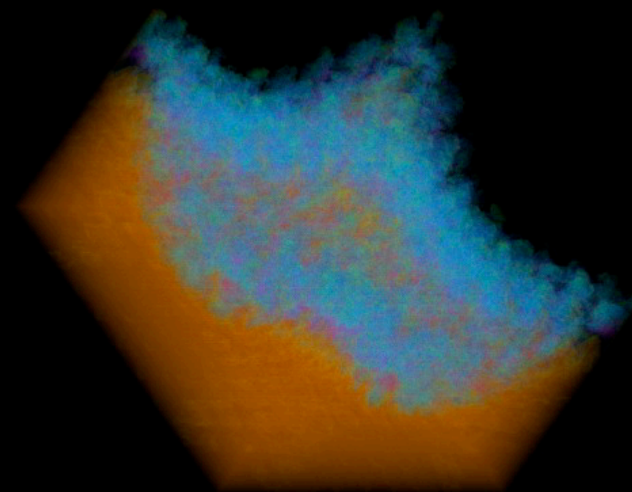
Histogram-based I/O Optimization for Visualizing Large-scale Data



Yuan Hong, The Ohio State University

Tom Peterka, Argonne National Laboratory

Han-Wei Shen, The Ohio State University

Tom Peterka

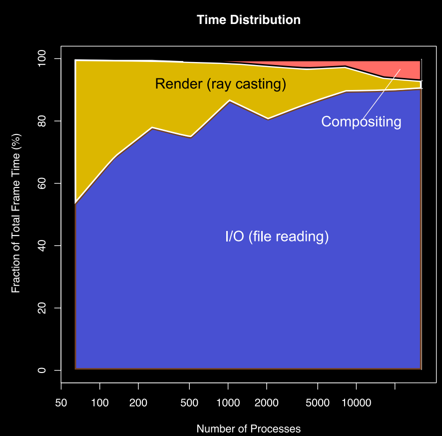
tpeterka@mcs.anl.gov

Mathematics and Computer Science Division

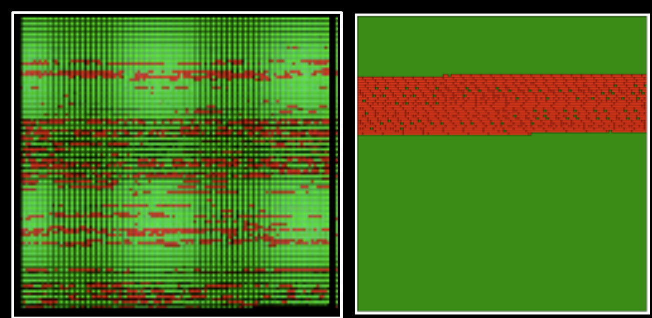
I/O Optimization for Visualization

Motivation

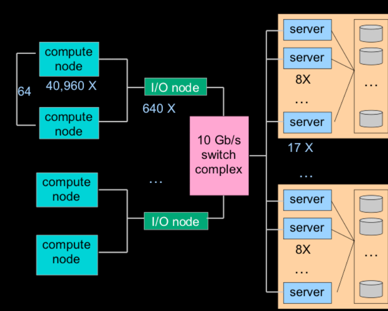
Performance of parallel visualization bound by data movement



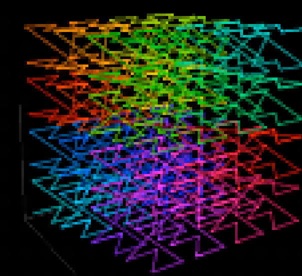
Visualization techniques resulting in sparse traversal can exacerbate the problem



Parallel I/O is necessary, but not sufficient



Effect of space-filling curves diminishes as process count increases



Idea

Consider both visibility culling and spatial locality when ordering data. Sample a variety of view directions and construct a histogram of visible blocks, independent of transfer function. Reorder data accordingly to balance load across file servers and produce contiguous access.

Related Literature

Background

Visibility culling

Gao et al., Visibility Culling Using Plenoptic Opacity Functions for Large Volume Visualization, Vis '03.

Zhang et al., Visibility Culling Using Hierarchical Occlusion Maps, SIGGRAPH '97.

Out of core methods

Pascucci and Frank, Global Static Indexing for Real-Time Exploration of Very Large Regular Grids, SC01.

Isenburg and Lindstrom, Streaming Meshes, Vis '05.

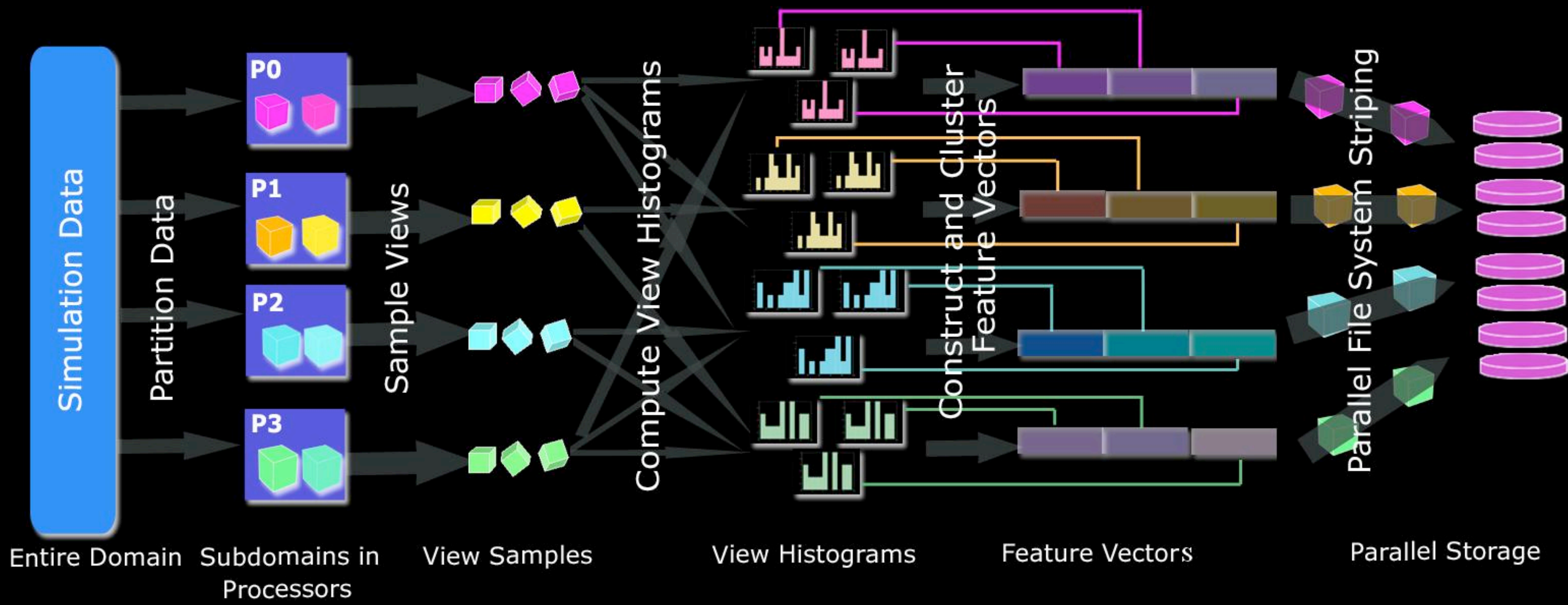
Collective I/O

Thakur et al., Optimizing Noncontiguous Access in MPI-IO, Parallel Computing '02.

Smirni et al., Algorithmic Influences on I/O Access Patterns and Parallel File System Performance, ICPADS '97.

Algorithm

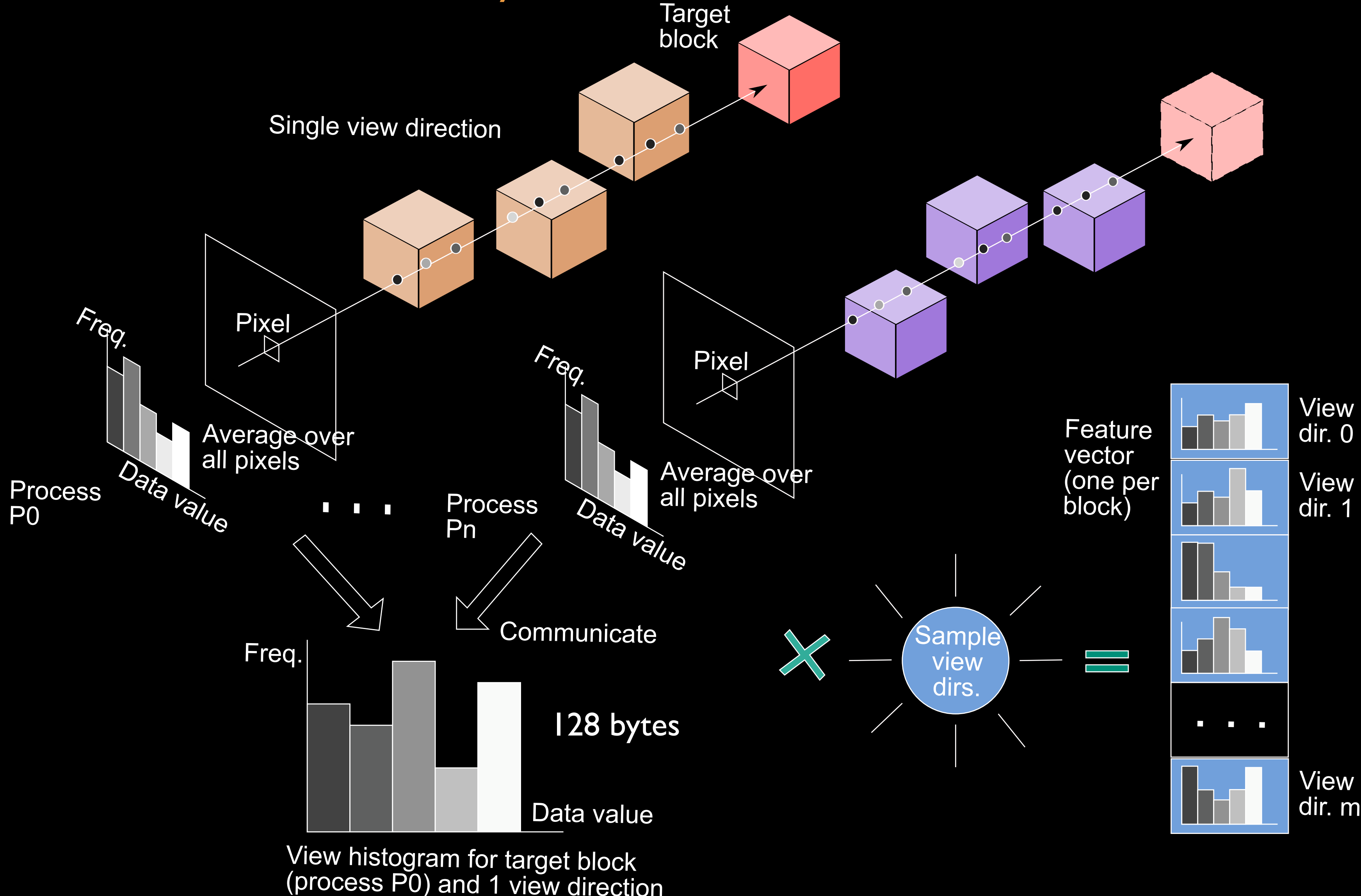
Overview



Algorithm overview consists of: partitioning data, sampling views on a view sphere, computing view histograms for each view direction, concatenating view histograms into feature vectors, grouping similar feature vectors into clusters, and striping data blocks onto parallel storage according to the clusters.

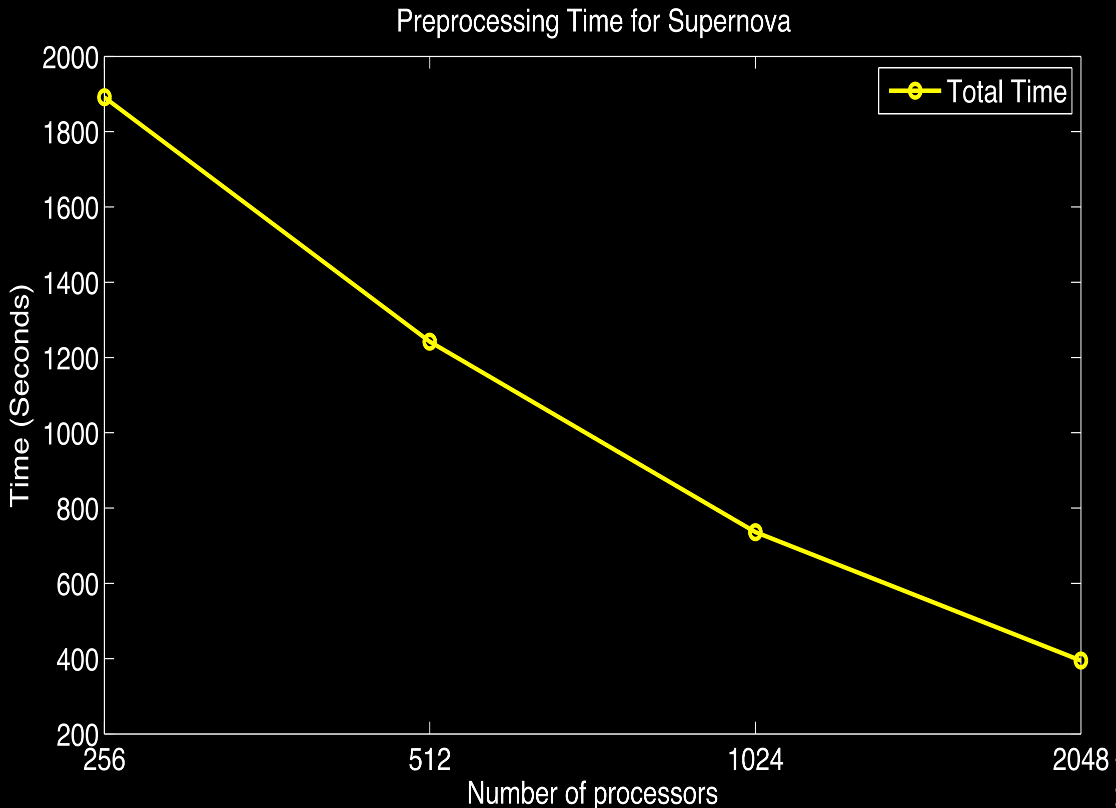
Compute View Histograms and Feature Vectors

Classify data in all view directions



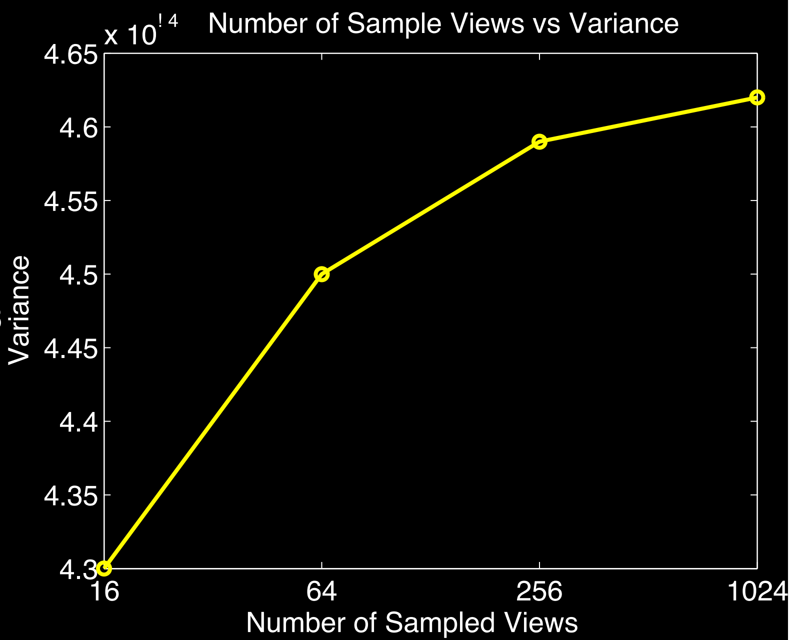
Feature Vector Computational Cost

Scalable parallel implementation



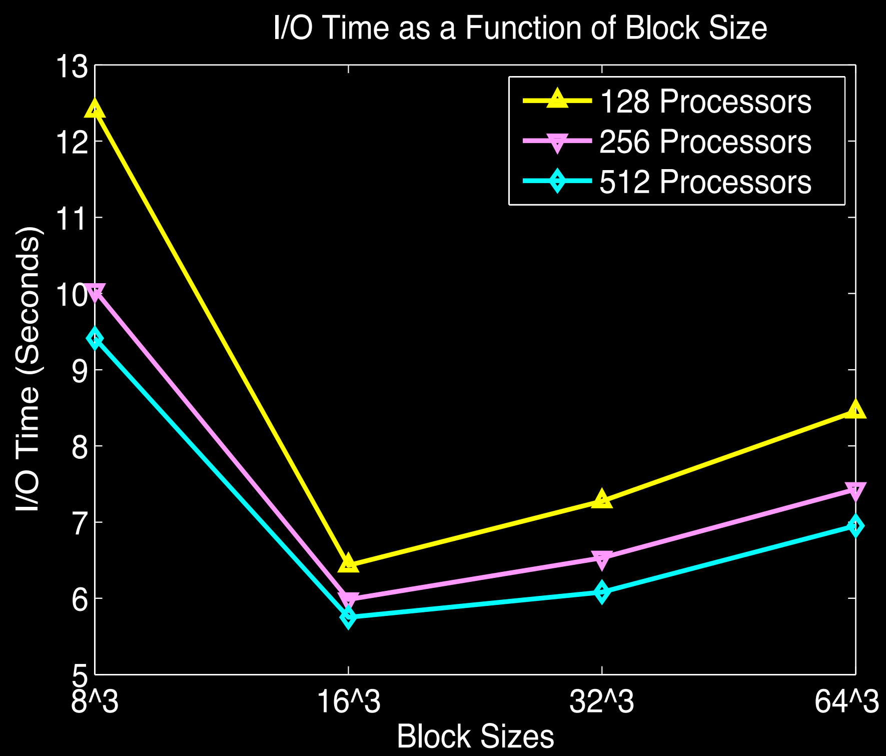
Total preprocessing time for supernova dataset, from 256 to 2048 cores, on Argonne's BG/P system. The dataset is 276 GB, and 1024 views were sampled in under seven minutes.

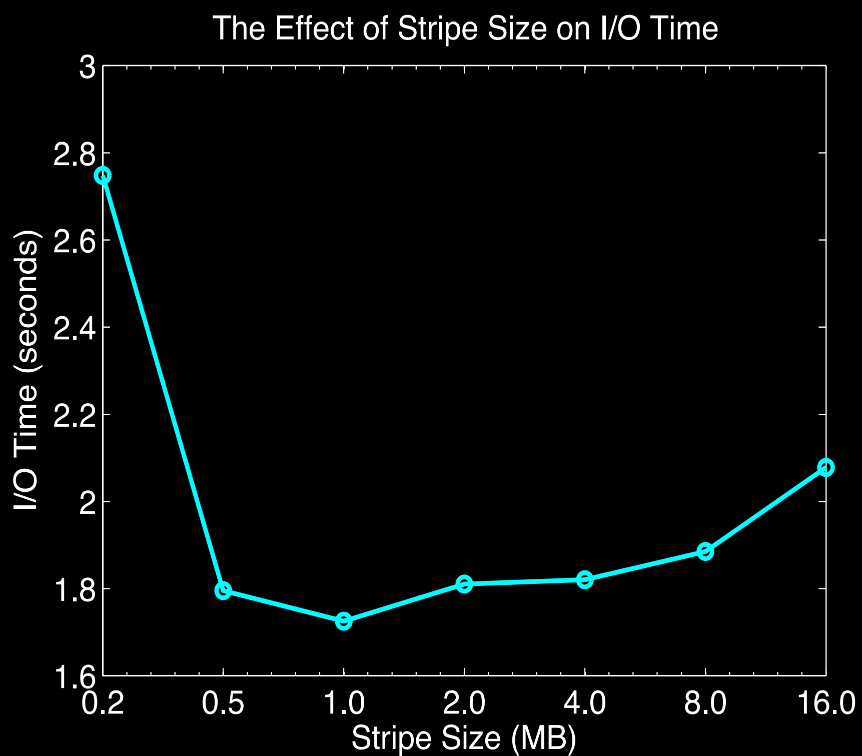
The variance across all histogram bins and all view directions as a function of the number of view directions. The variance changes slowly after 256 sampled views, indicating that more samples are not necessary. Viswoman dataset.



Organizing Data in Storage

Layout parameters





Block size of 16^3 has best I/O performance for Viswoman dataset, irrespective of process count. Block size is chosen to be a multiple of the read buffer size, 16 KB in our default MPI-IO implementation.

I/O time vs. stripe size for Viswoman dataset. Optimal stripe size is that of average cluster size that results from clustering feature vectors.

End-to-End Performance

Test conditions, datasets, total and component time

Test conditions:

System: IBM BG/P at Argonne National Laboratory, PVFS file system

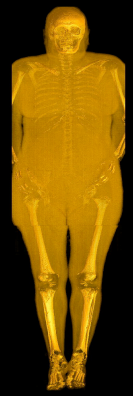
Viswoman dataset: 512x512x1728, 2-byte short ints, 16^3 blocks

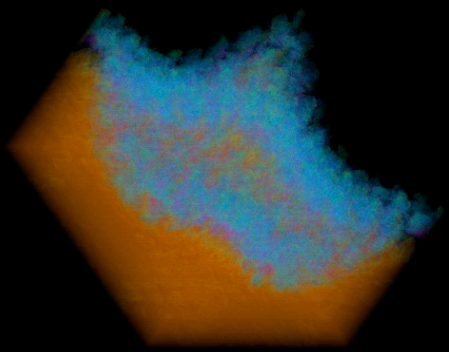
Richtmyer-Meshkov Instability (RMI) dataset: 2048x2048x1920, 1-byte chars, 32^3 blocks

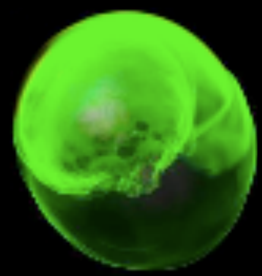
Supernova dataset: 3456x3456x3456 supersampled, 4-byte floats, 16^3 block size

Viswoman volume rendering performance with histogram-optimized method

# Procs	I/O time (s)	Render time (s)	Composite time (s)	Total time (s)
64	4.37	1.02	1.20	6.59
128	3.66	0.46	0.80	4.92
256	3.43	0.33	0.80	4.56
512	1.77	0.20	0.60	2.57
1024	0.91	0.12	0.50	1.53

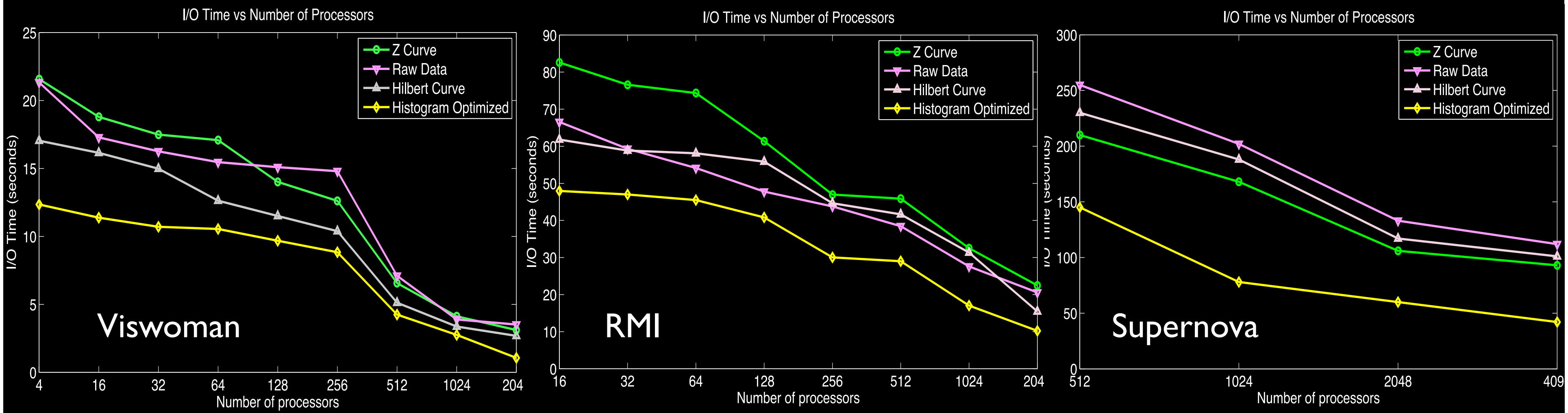




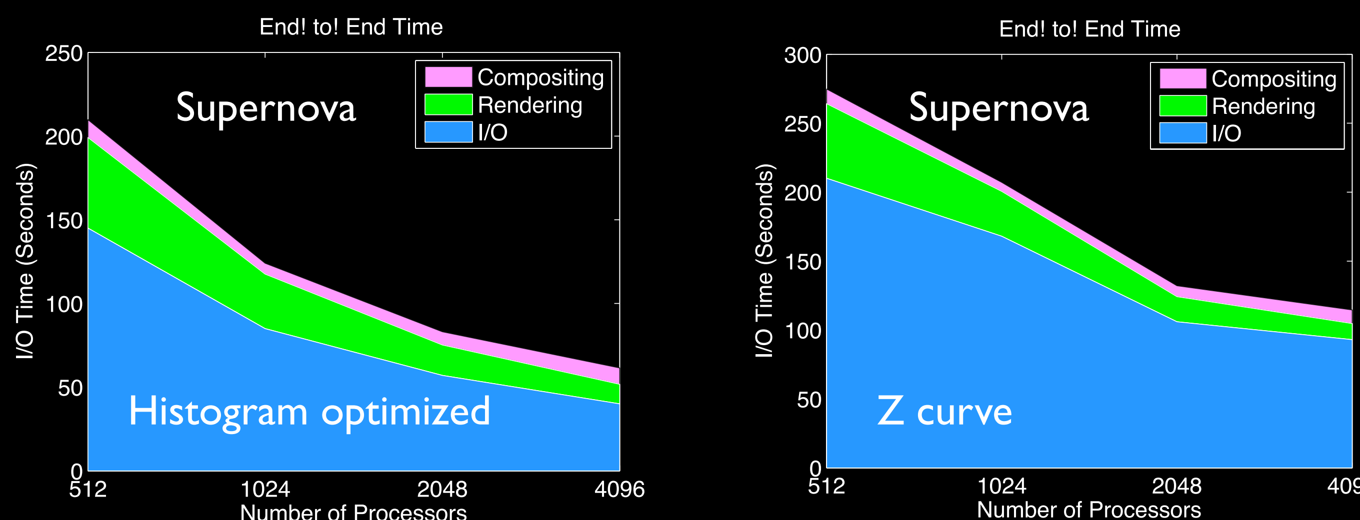


Comparison to Space-Filling Curves

I/O time for three datasets

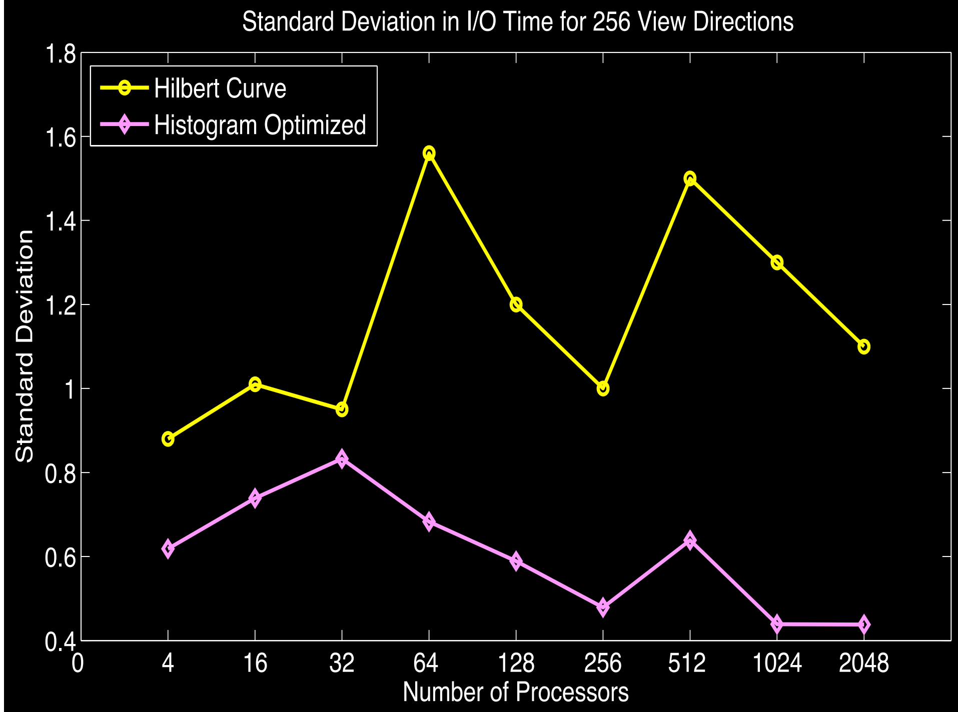


Top: I/O time for three datasets. Bottom: compositing, rendering, I/O time for supernova. In all test cases, the histogram-optimized method performs better than canonical organization and space-filling curves.



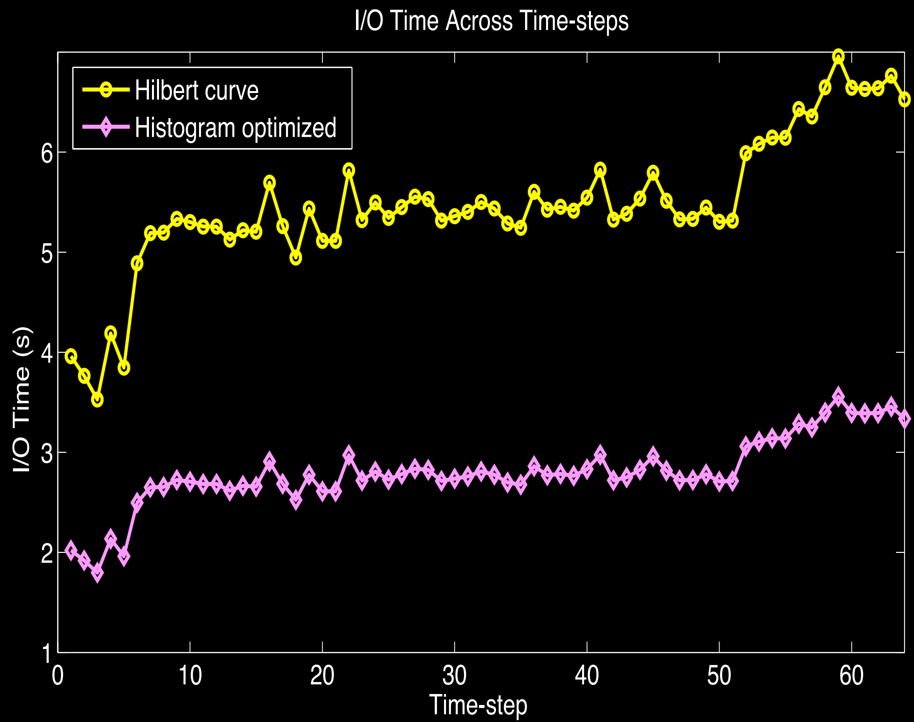
Comparison to Hilbert Curve

Across view directions



Standard deviation of I/O time in RMI across 256 random view directions demonstrates consistent performance over variety of view conditions.

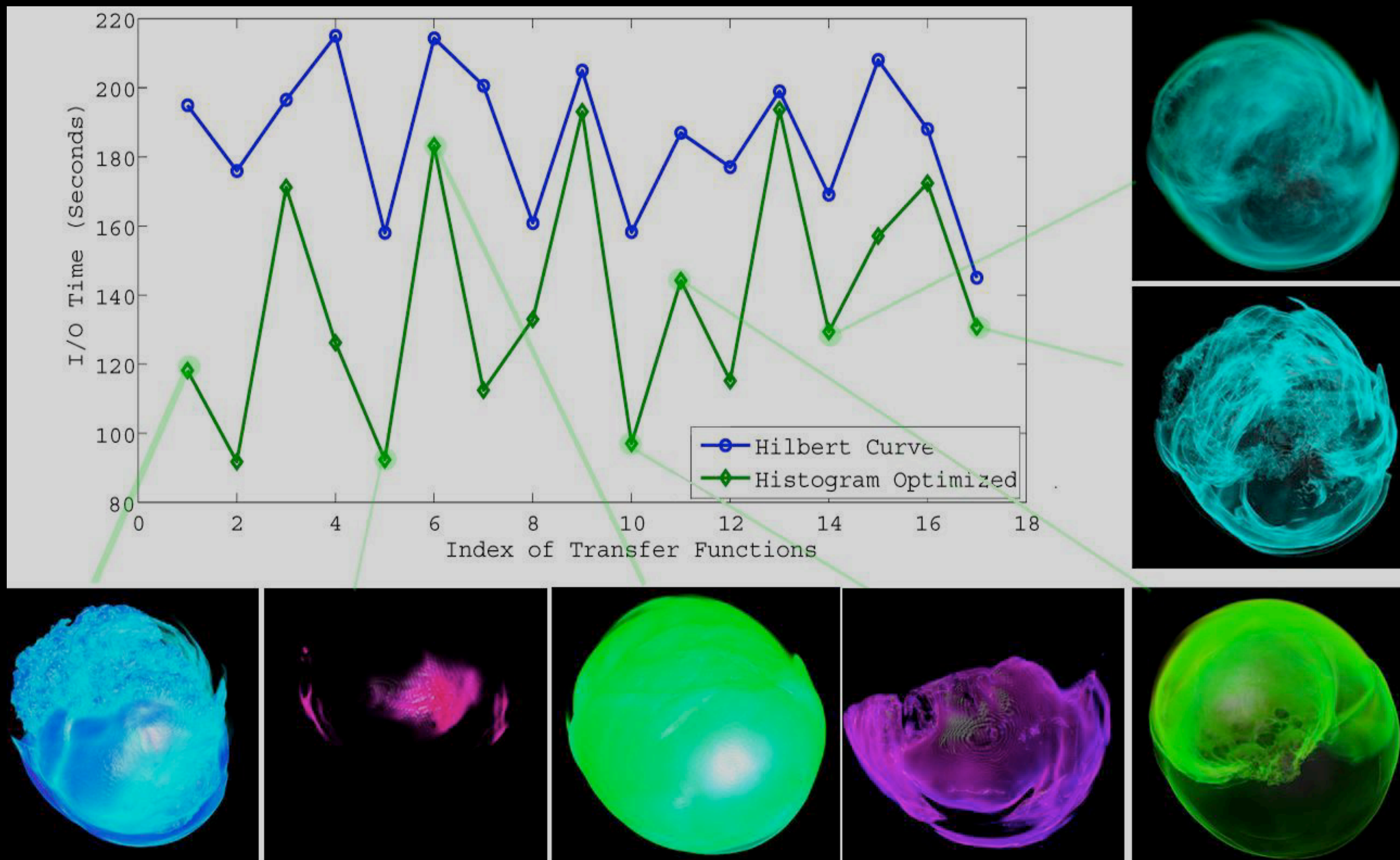
Across time-steps



I/O time across 64 time-steps of RMI with 512 processors demonstrates consistent performance over a time-varying dataset.

Independent of Transfer Function

Various opacities, single and multimodal



I/O time for histogram-optimized and Hilbert curve for supernova dataset rendered with a variety of transfer functions. Transfer functions were generated synthetically using a nonlinear computation that stochastically produces one or more modes.

Argonne NATIONAL LABORATORY
... for a brighter future

Histogram-based I/O Optimization for Visualizing Large-scale Data

Successes

- Data organization based on visibility culling and spatial locality
- Scalable feature classification time
- Improved volume rendering performance over space-filling curves
- Transfer function independence
- Heuristics for usage

Limitations / Future work

- Scale to higher number of processes
- Zoom
- Higher-dimension transfer functions
- Other storage and file systems

UltraVis
SciDAC Institute for Ultrascale Visualization
www.ultravis.org

U.S. Department of Energy

UChicago Argonne LLC

Office of Science
U.S. DEPARTMENT OF ENERGY

A U.S. Department of Energy laboratory managed by UChicago Argonne, LLC

Acknowledgments:
Argonne Leadership Computing Facility
US DOE SciDAC UltraVis Institute

Tom Peterka
tpeterka@mcs.anl.gov
Mathematics and Computer Science Division