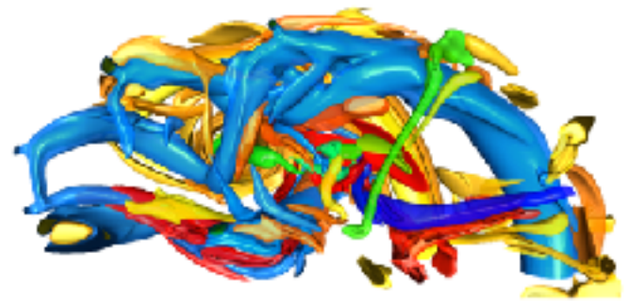
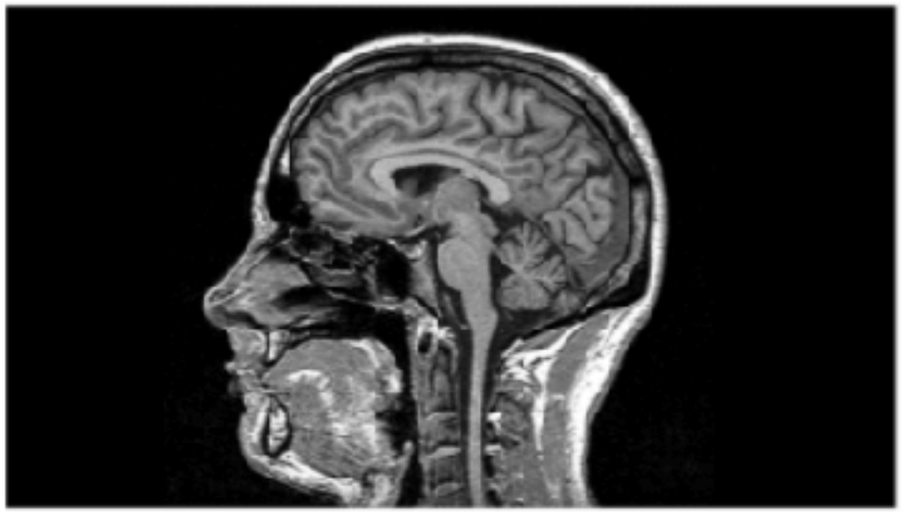
Scientific Visualization for Computational Science

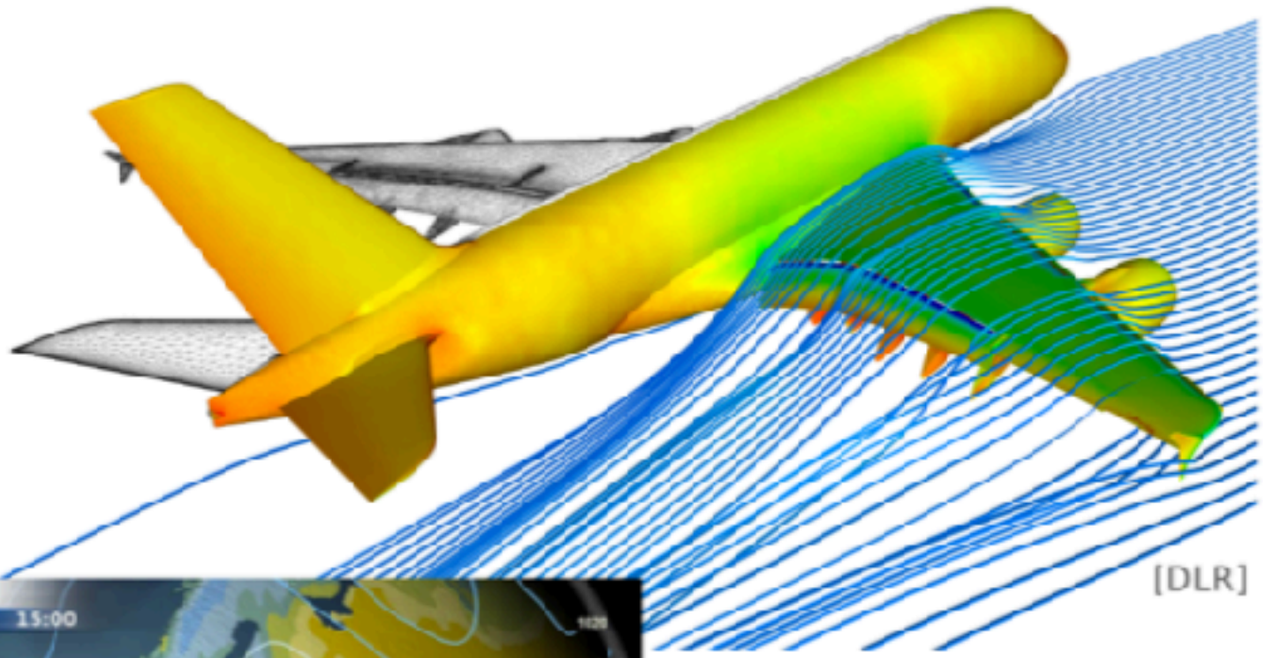
Christoph Garth
University of Kaiserslautern

International Summer Workshop on Visual Computing
Universidad de los Andes, Bogotá, June 2018



Scientific Visualization Examples





[DLR]

[Bruckner]

[wetter.net]

[science museum]

Visualization

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more efficiently.

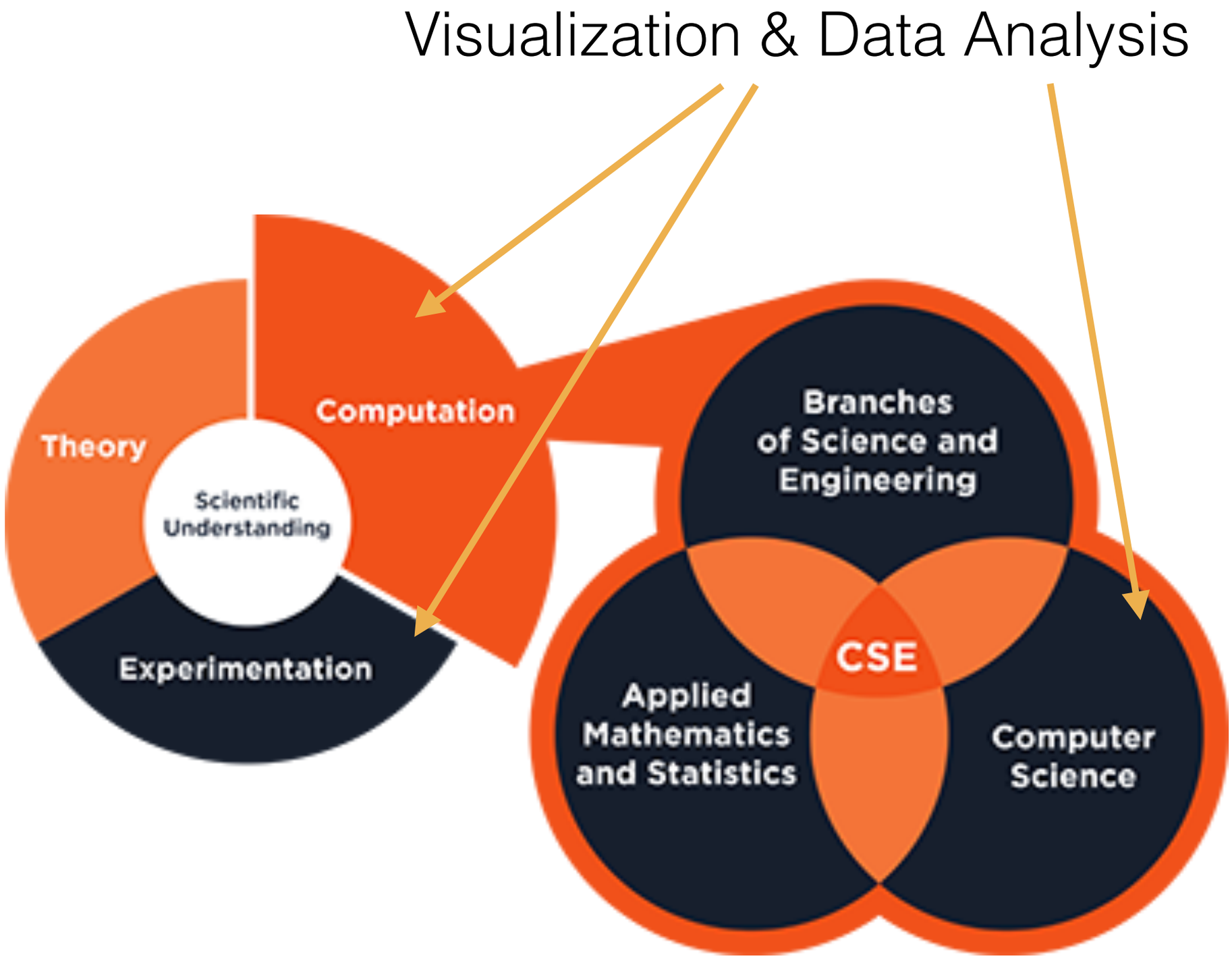
Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods.

[Munzner 2014]

Key tasks:

- exploration and discovery
- hypothesis formation and confirmation
- communication of results, sharing of insight

Visualization in Computational Science



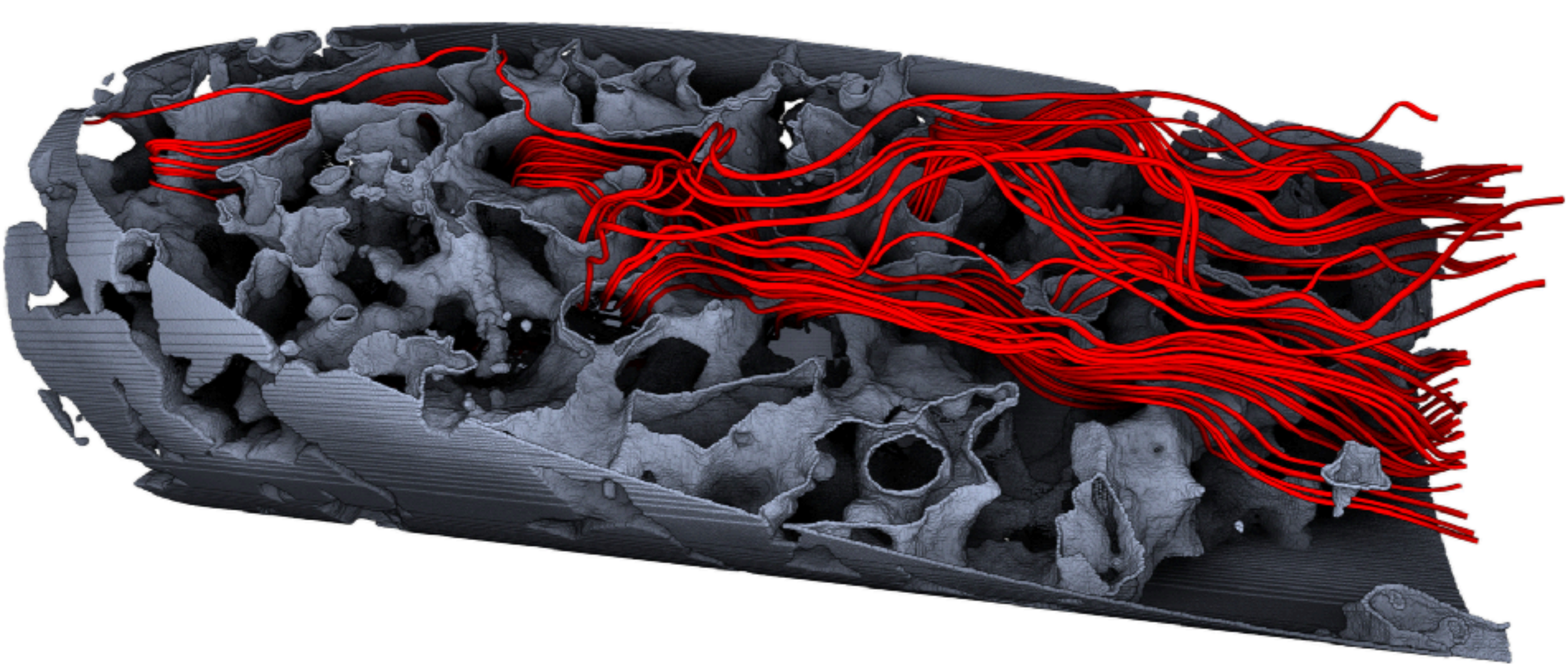
Large-Scale Computational Science

Modern workflows leverage massively parallel machines.

- clusters
  (10s - 1000s of proc.)
- supercomputers
  (1,000,000s of proc.)

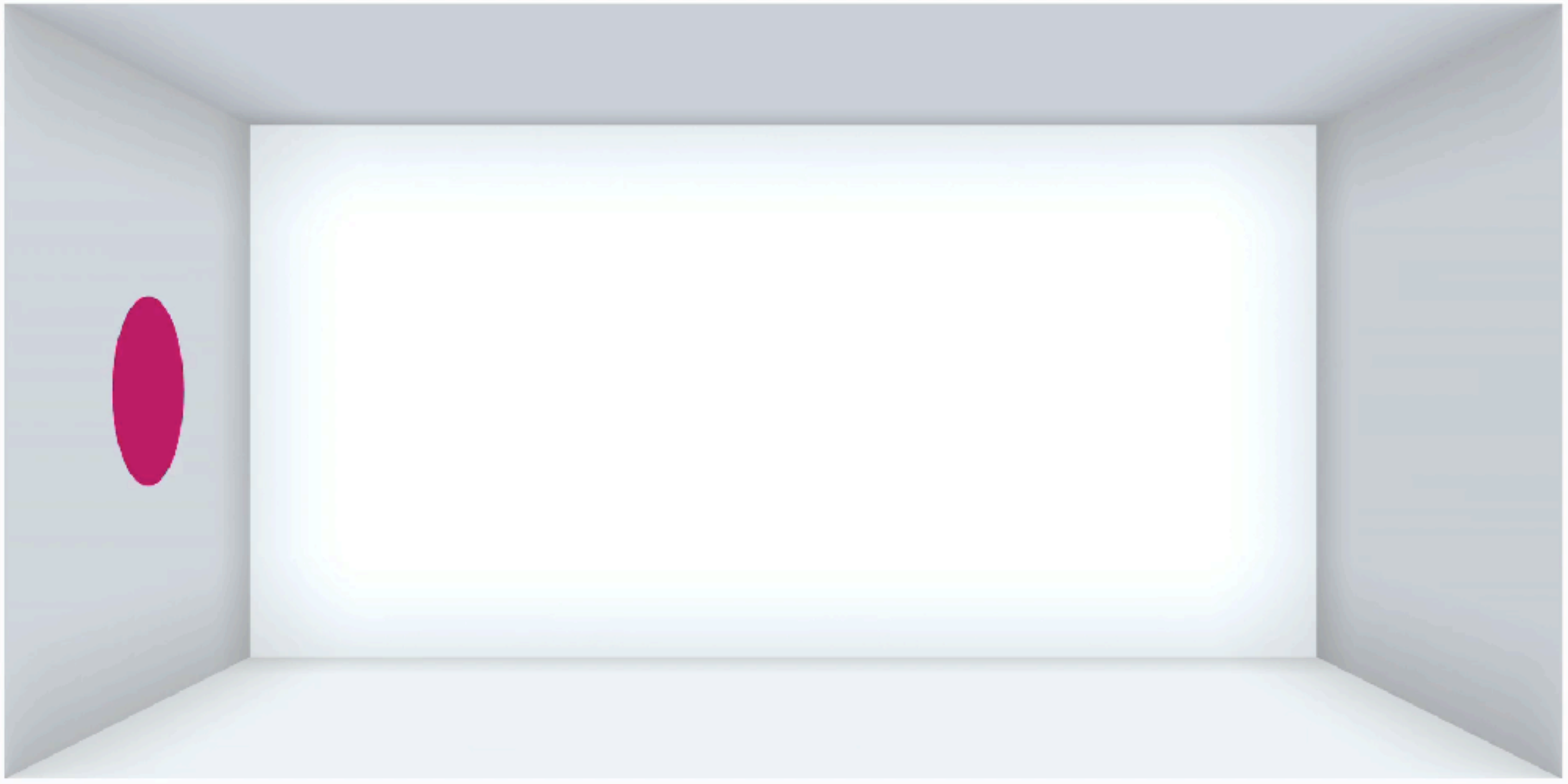
This allows to consider problems of almost arbitrary complexity, scale and resolution (Exascale era).

Scientific Visualization Examples



Flow of water through Karst limestone (data: TACC): 500 GB

Scientific Visualization Examples



Jet flow (DNS, data: Scientific Computation Group @ TUK): 1TB

Modern Simulation Data

State-of-the-art results from Computational Science

Structural richness of the data; simple feature definitions are not sufficient; quantification of relevance & simplification needed.

Traditional scalar, vector, tensor visualization techniques do not apply.

“Uncertainty” modelled along with data – lost in visualization unless explicitly cared for.

Efficient algorithms are a must, or, alternative techniques are needed (→ *in situ* visualization).

Overview

A quick tour through some modern paradigms:

- Topology-based and Feature-based Techniques
- Parallel Visualization Algorithms
- In Situ Visualization

Necessarily incomplete.

Feature-Based Visualization

Features in Scientific Visualization

Feature (working definition): a subset of data that is interesting or relevant in the sense that it facilitates one or more of the three tasks of visualization.

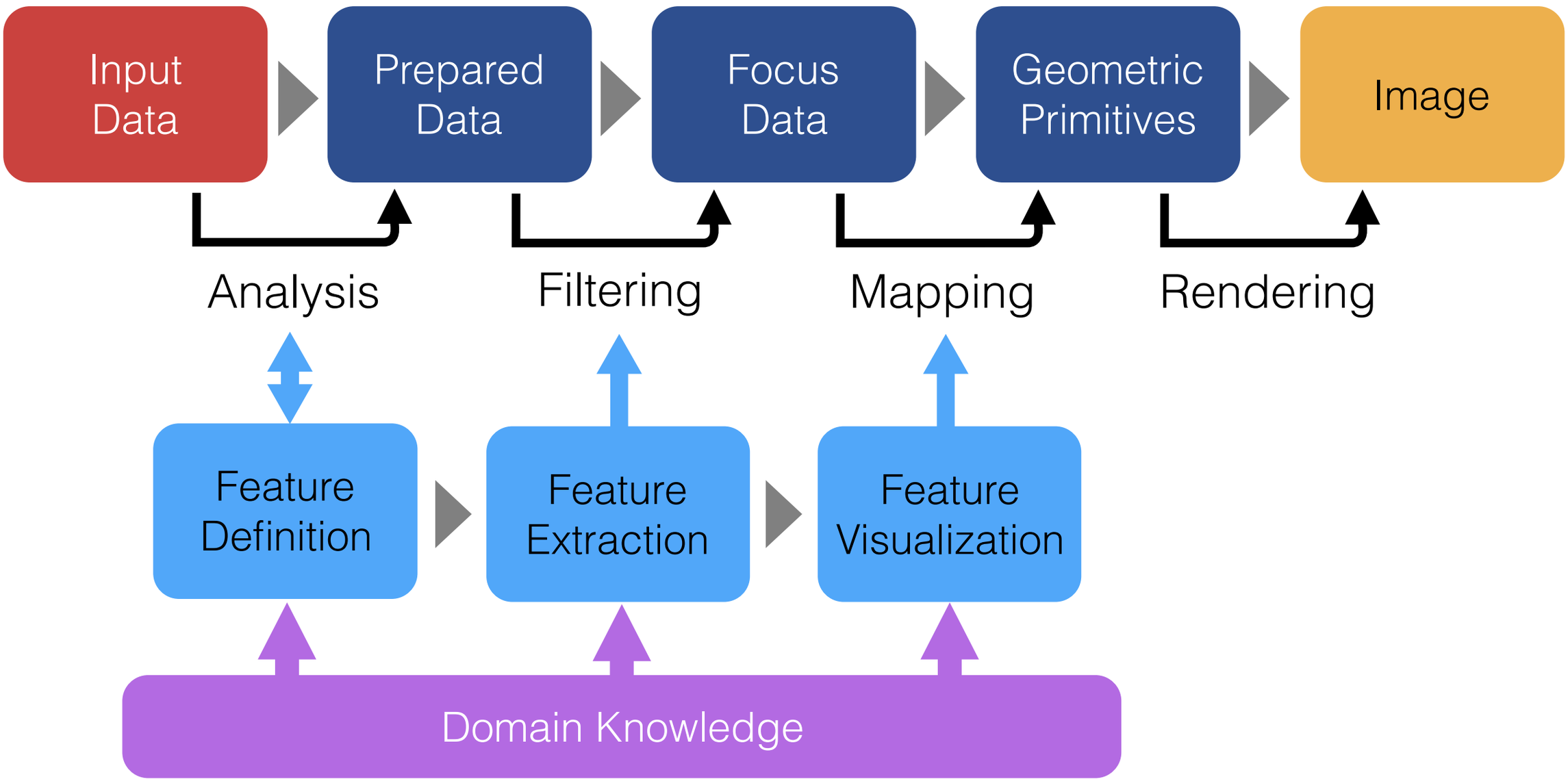
For example, a region of a dataset where some functional relation holds

Defining relevant features specific problems requires describing an (efficiently) computable feature criterion.

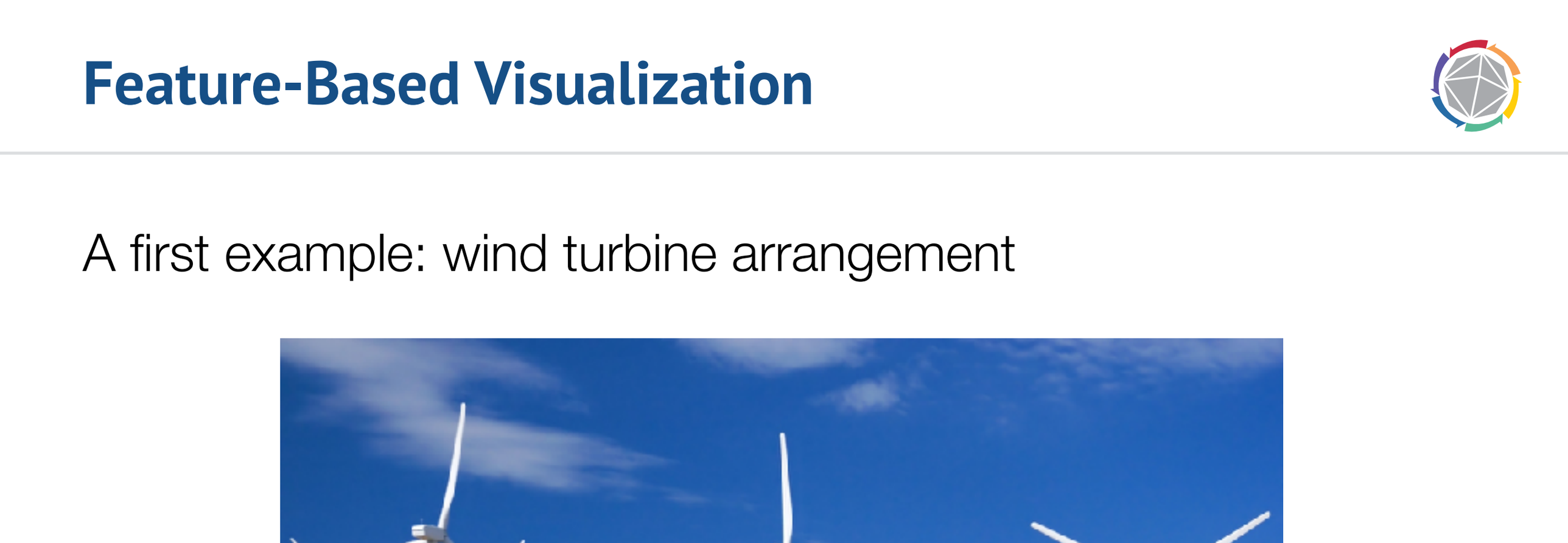
Feature-based visualization is a particular mental framework to reason about visualization.

Features in the Visualization Pipeline

A model for the visualization process: the visualization pipeline.



Feature-Based Visualization

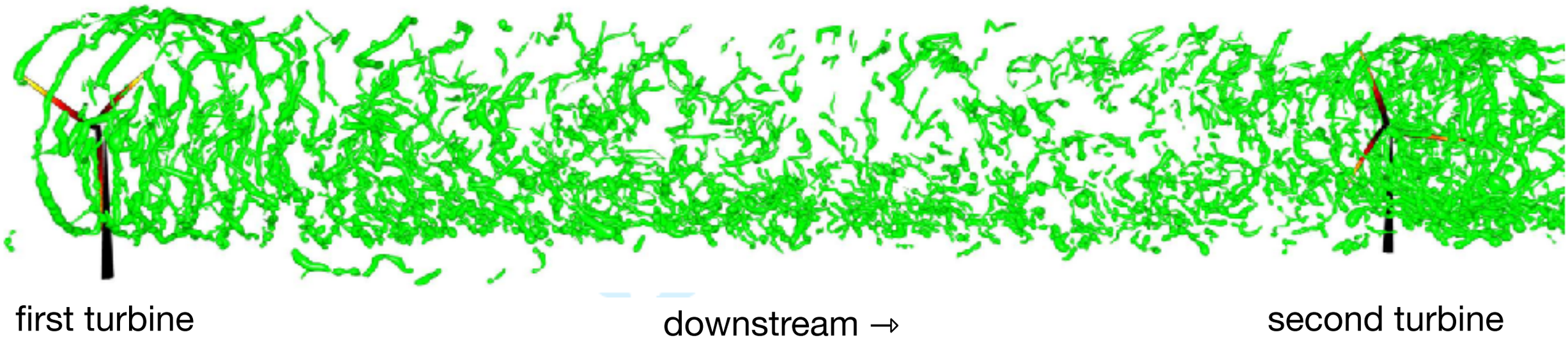


A first example: wind turbine arrangement

(image: leanfiberopticdesign.com)

Feature-Based Visualization

A first example: wind turbine arrangement



data from R. Linn, E. Koo, Los Alamos National Laboratory

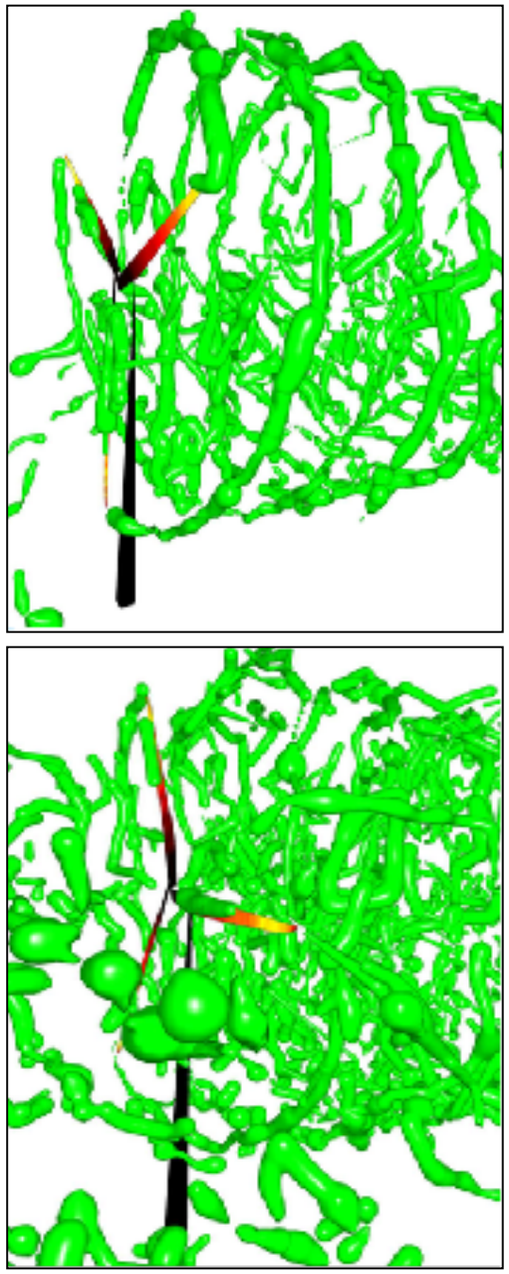
Vortices from first turbine hit second turbine rotor blades and induce heightened stresses, leading to maintenance overhead and failure.

S. Shafii, H. Obermaier, R. Linn, E. Koo, M. Hlawitschka, CG, B. Hamann, K. I. Joy: *Visualization and Analysis of Vortex-Turbine Intersections in Wind Farms.* IEEE TVCG 19(9):1579-1591, 2013

Feature-Based Visualization

Feature definition:

Vortical regions transported along wind vector field from upstream to downstream turbine.



Feature extraction:

- extraction of vortical regions
- computation of particle paths along wind field
- filter for particles connecting upstream and downstream turbines, contained in vortical regions

S. Shafii, H. Obermaier, R. Linn, E. Koo, M. Hlawitschka, CG, B. Hamann, K. I. Joy: *Visualization and Analysis of Vortex-Turbine Intersections in Wind Farms.* IEEE TVCG 19(9):1579-1591, 2013

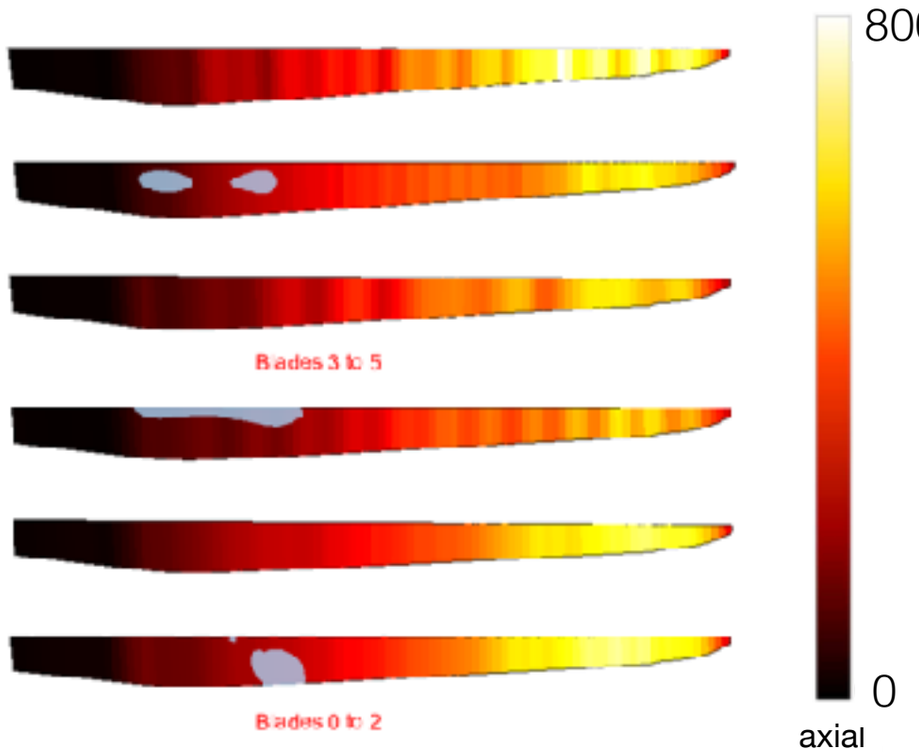
Feature-Based Visualization

Feature visualization: improved mapping

Intersections on (moving) blades are hard to see.
Where do vortices hit? How much force do they induce?

naïve (3D)

intersections (3D)

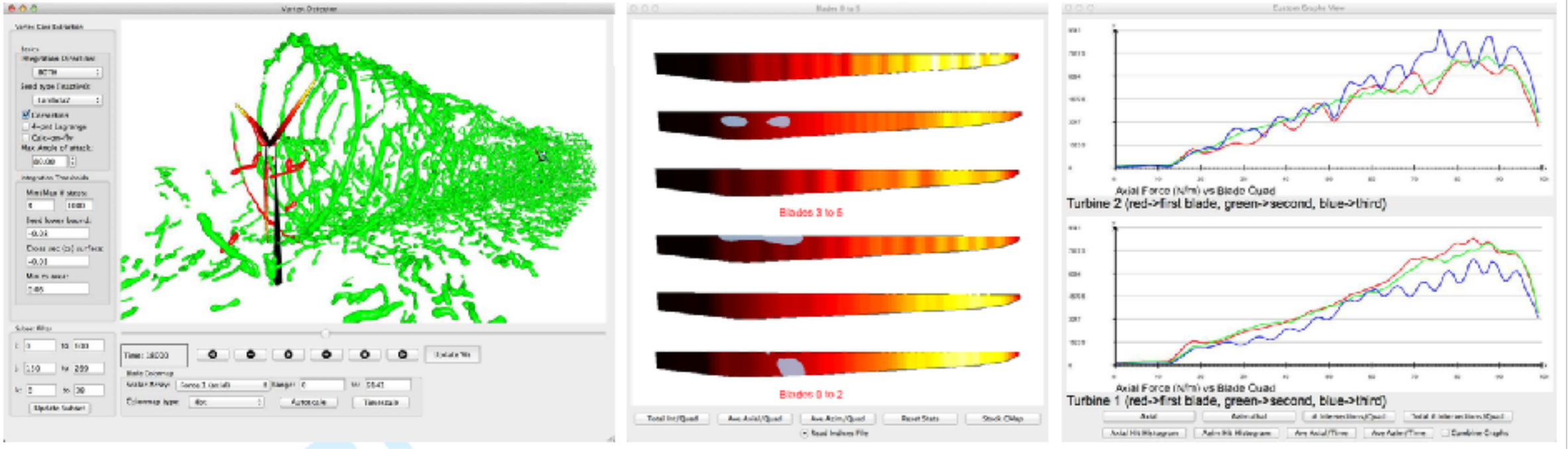


intersections (2D)

S. Shafii, H. Obermaier, R. Linn, E. Koo, M. Hlawitschka, CG, B. Hamann, K. I. Joy: *Visualization and Analysis of Vortex-Turbine Intersections in Wind Farms.* IEEE TVCG 19(9):1579-1591, 2013

Feature-Based Visualization

Feature-based visualization system:
coupling of feature definition, extraction, and visualization.



In this case, augmented with statistical feature information.

S. Shafii, H. Obermaier, R. Linn, E. Koo, M. Hlawitschka, CG, B. Hamann, K. I. Joy: *Visualization and Analysis of Vortex-Turbine Intersections in Wind Farms.* IEEE TVCG 19(9):1579-1591, 2013

Topological Techniques in Visualization

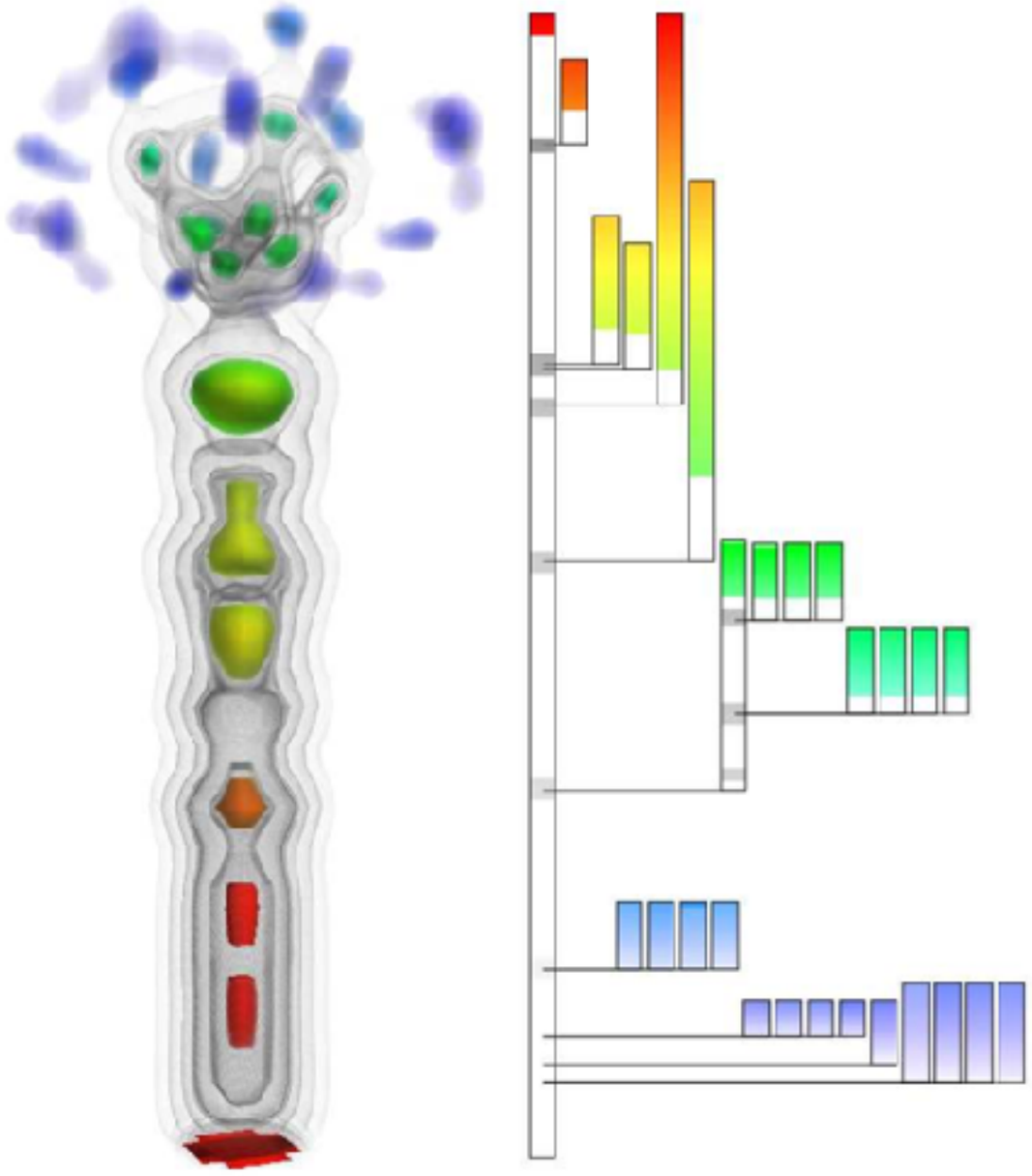
Topology is used to describe similarity up to smooth deformation.

Fundamentally, the mathematical notion of **topological equivalence** is applied to treat a complex dataset as a simpler, equivalent one; hence,

- to **abstract** complex datasets into comprehensible representation, and
- to **simplify** the structure in a meaningful way.

Fits well with the feature-based visualization mental model.

Contour Tree: extrema and basins



[Weber et al.: *Topology-controlled volume rendering.* IEEE TVCG 13(2):330-41, 2007.]

Topology of Area-Preserving Maps

Nuclear fusion in a Tokamak-type reactor:
Visualization of plasma confinement through closed fieldlines.

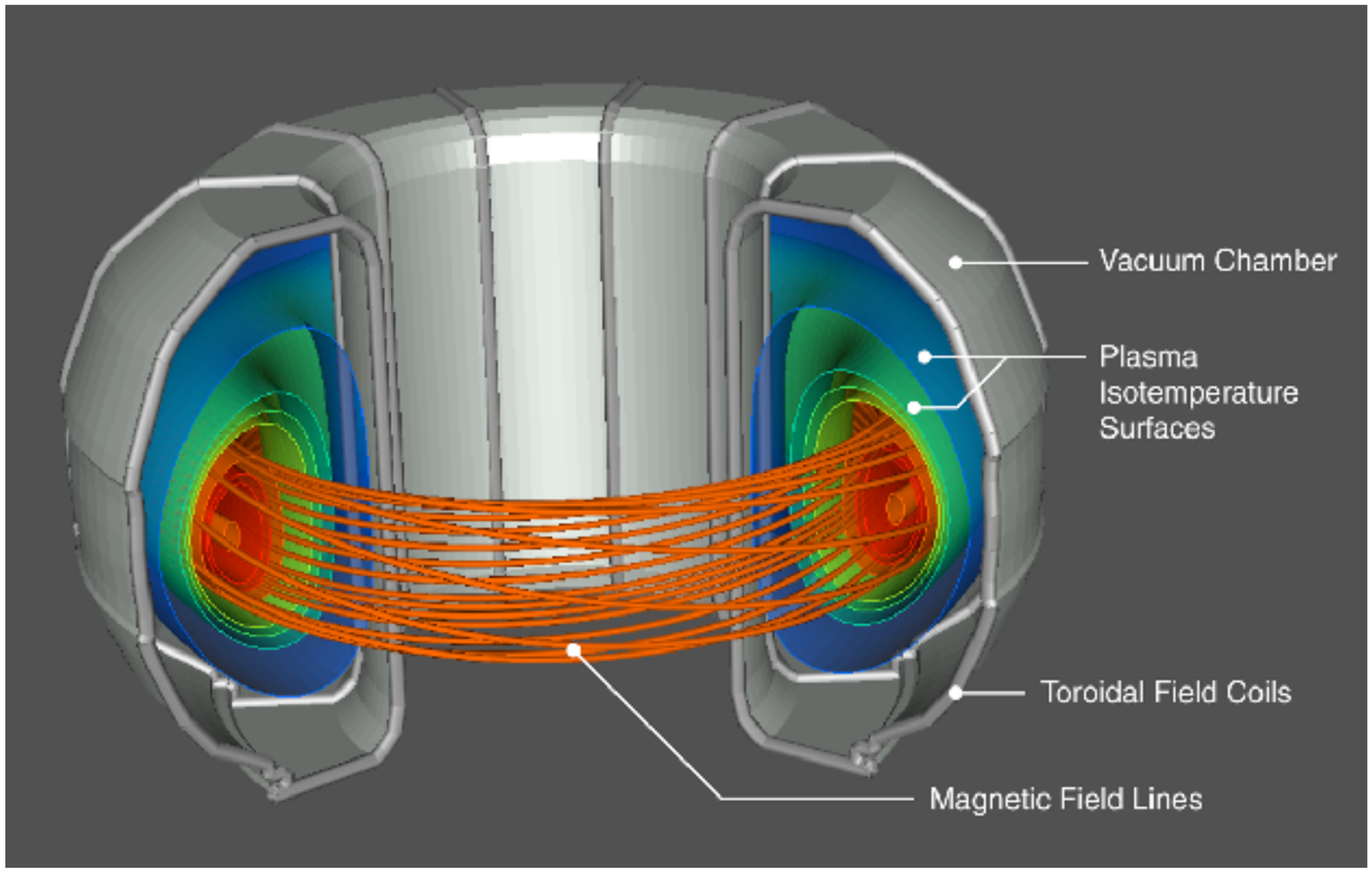
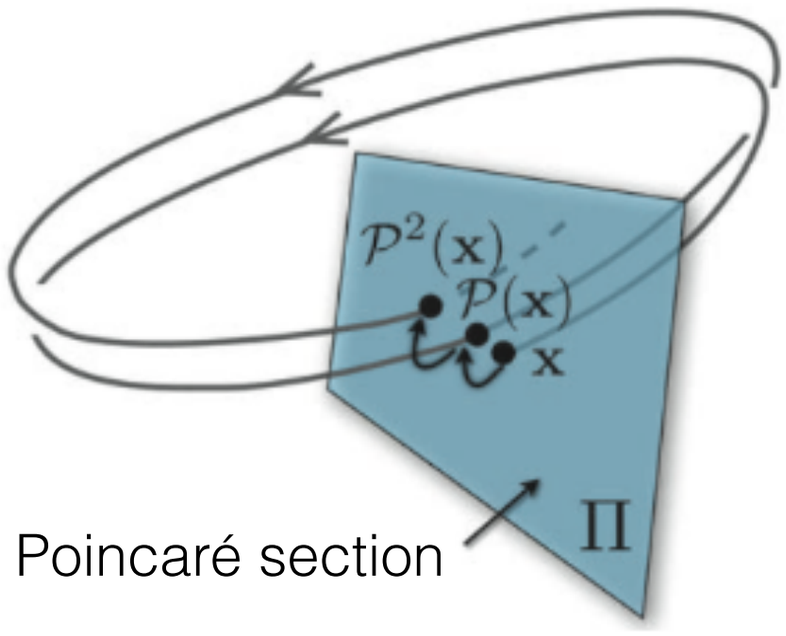


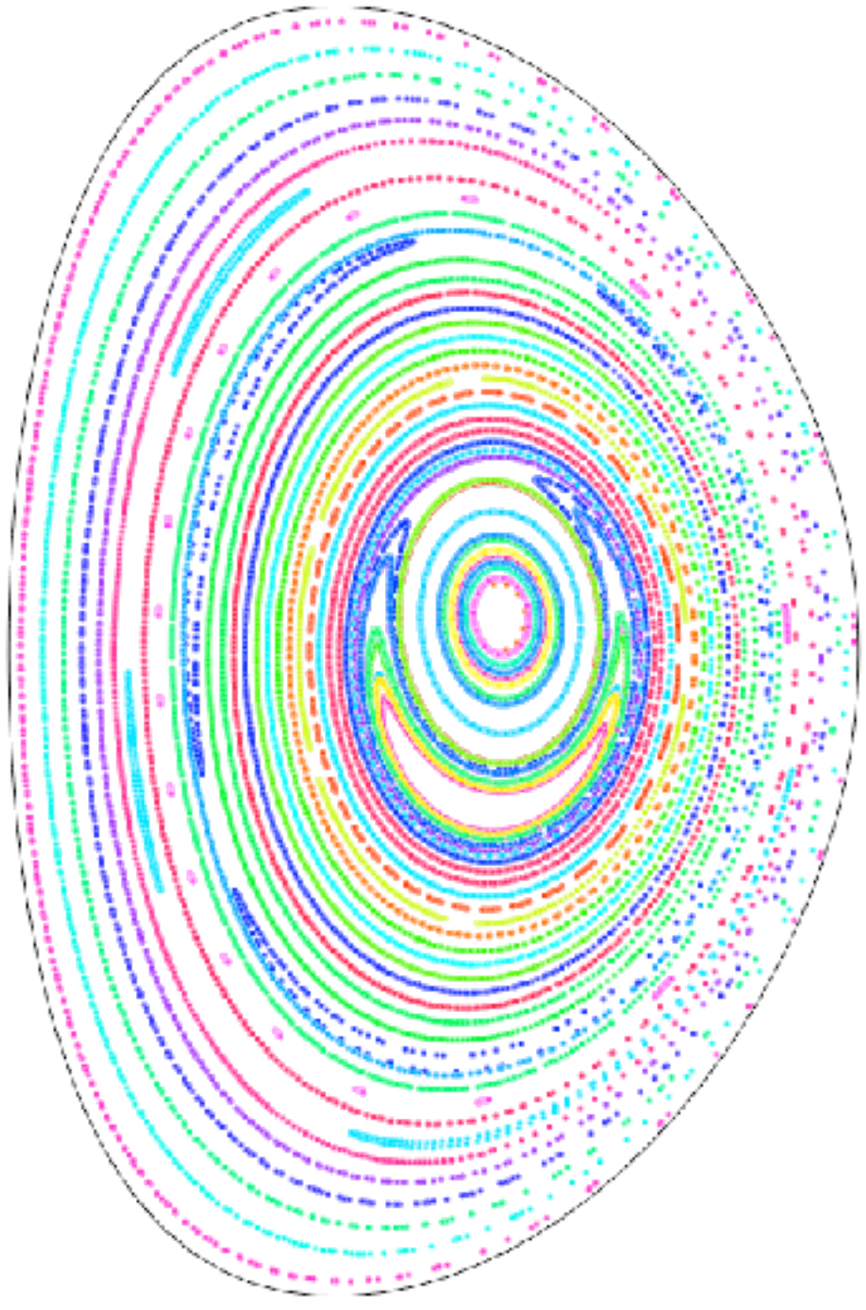
Image courtesy A. Sanderson, U. of Utah

Topology of Area-Preserving Maps

The magnetic field is described by a vector field; however, classical vector field topology is not applicable.

Typical visualization:
puncture plot (Poincaré map)

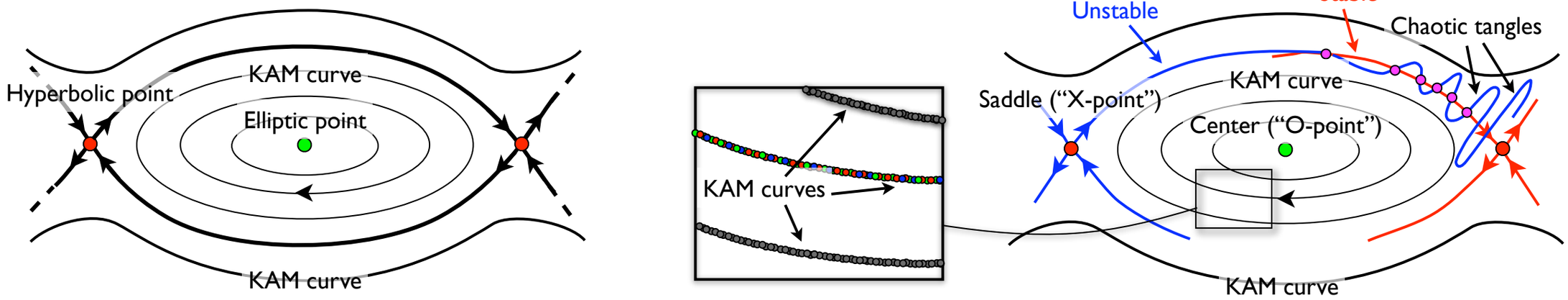




X. Tricoche, **CG**, A. Sanderson: *Visualization of Topological Structures in Area-Preserving Maps.* IEEE TVCG 17(12):1765–1774, 2011.

Topology of Area-Preserving Maps

Around periodic orbits, the return map exhibits a distinct topological structure:



non-chaotic

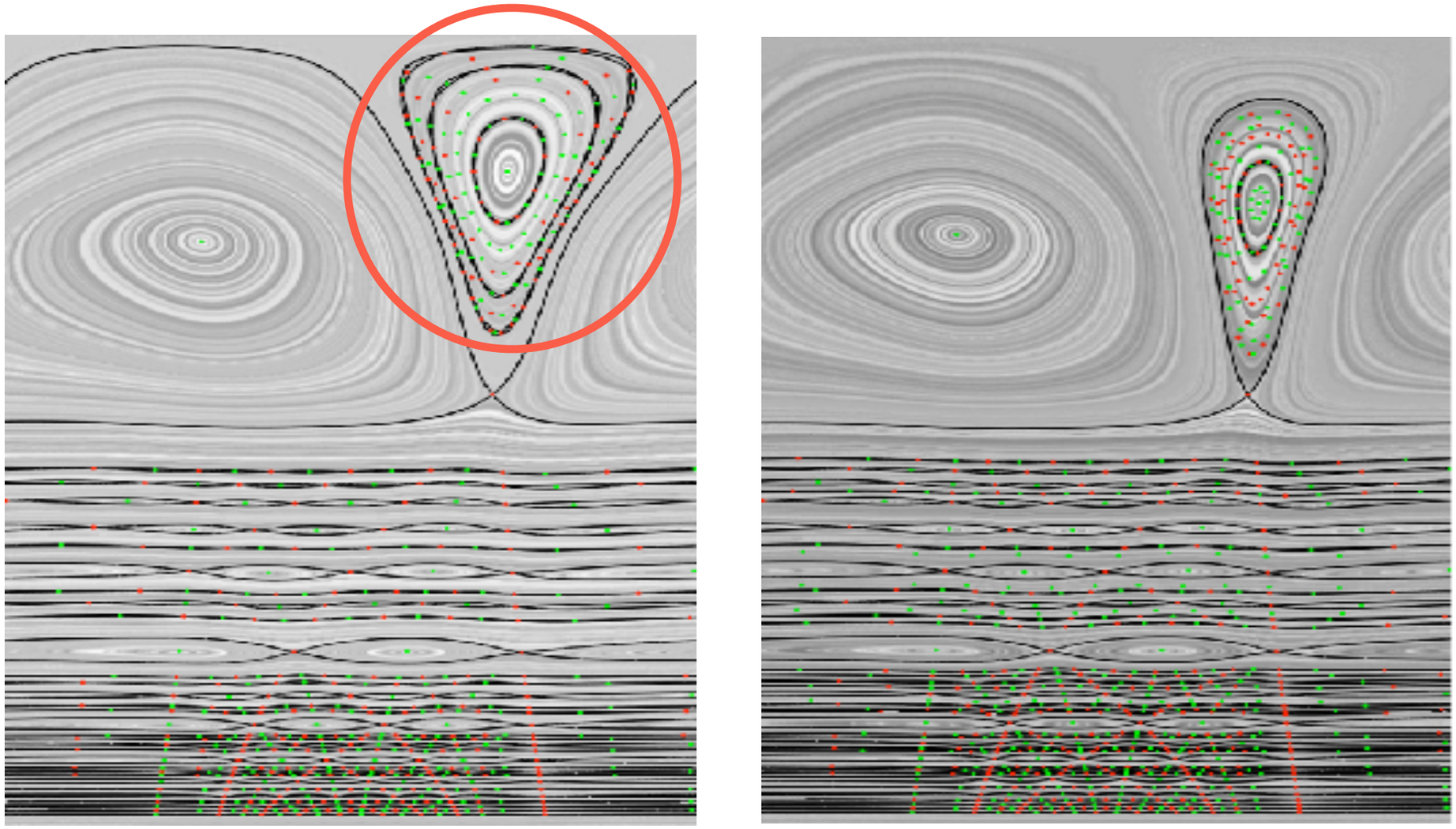
transition to chaos

This can be taken advantage of to enforce topological consistency for numerically inconsistent and challenging data.

X. Tricoche, **CG**, A. Sanderson: *Visualization of Topological Structures in Area-Preserving Maps.* IEEE TVCG 17(12):1765–1774, 2011.

Topology of Area-Preserving Maps

Visualization result: magnetic island reconnection



before reconnection

after reconnection

data: computed using NIMROD code, courtesy S. Kruger, Tech-X

Poincaré map topology + Context (orbital averaging)

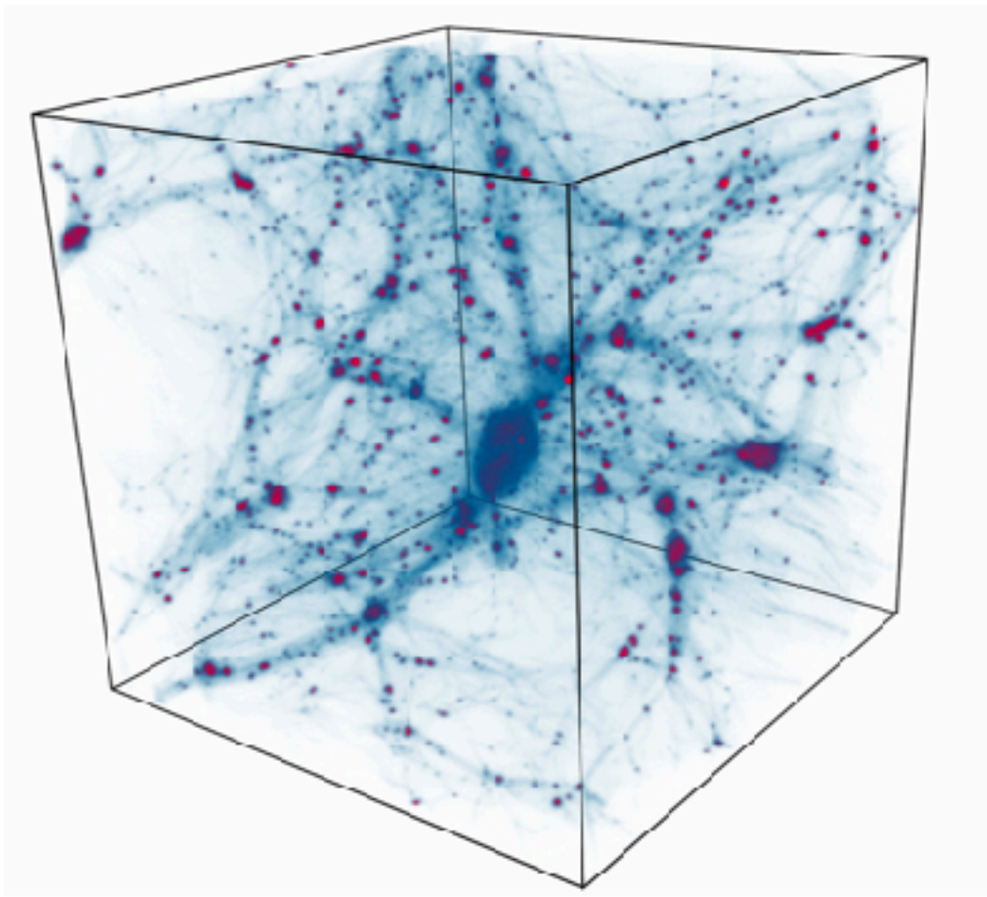
X. Tricoche, **CG**, A. Sanderson: *Visualization of Topological Structures in Area-Preserving Maps.* IEEE TVCG 17(12):1765–1774, 2011.

Topological Features

In many applications, features can be described as superlevel set components.

Viscous Fingering

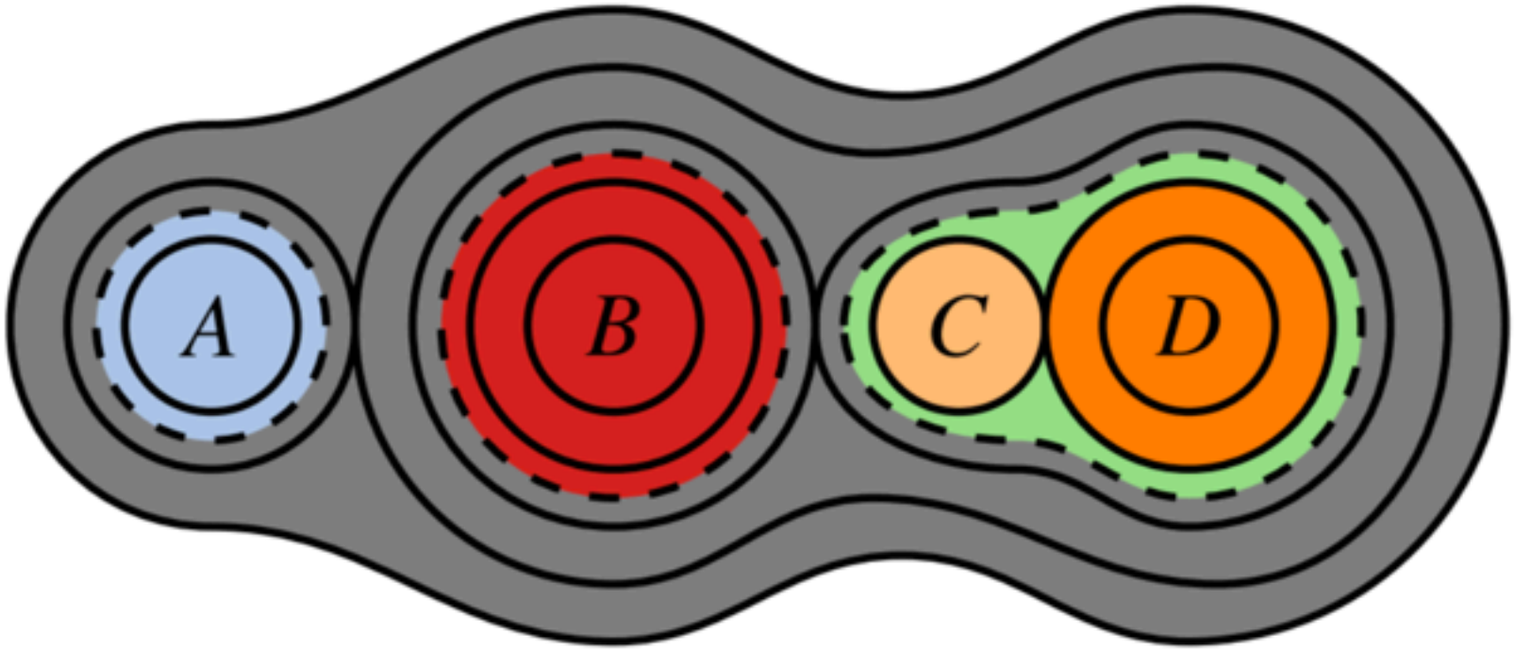
Fluid Simulations

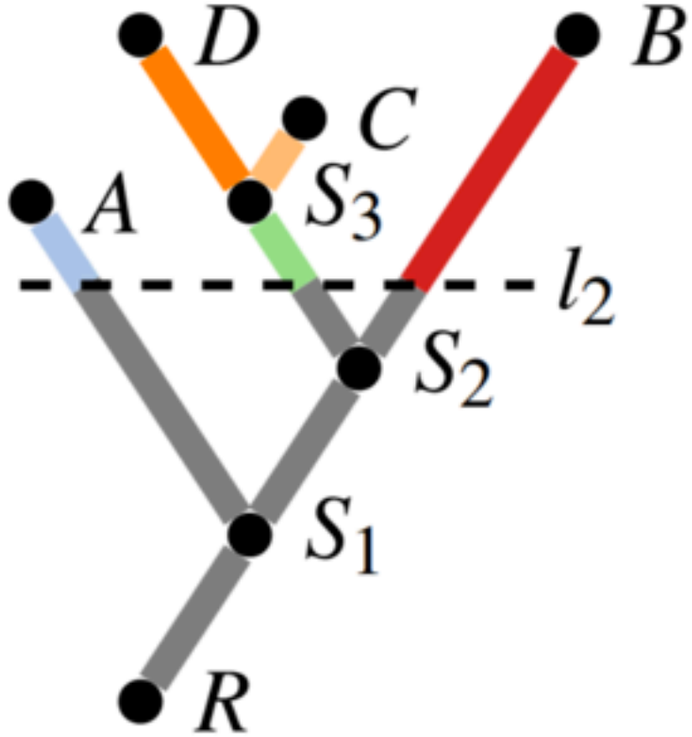


Dark Matter Halos

Topological Features

Simple example: Merge tree analysis

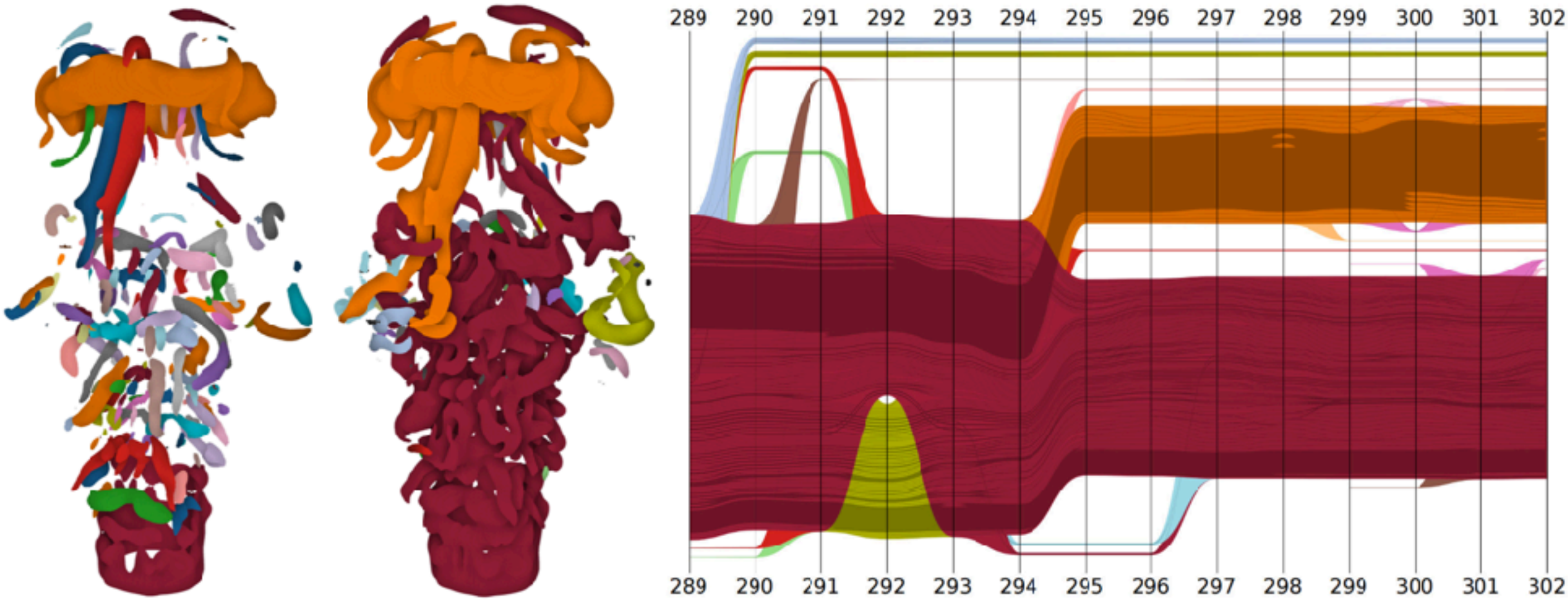




Evolution over time?

Tracking of Topological Features

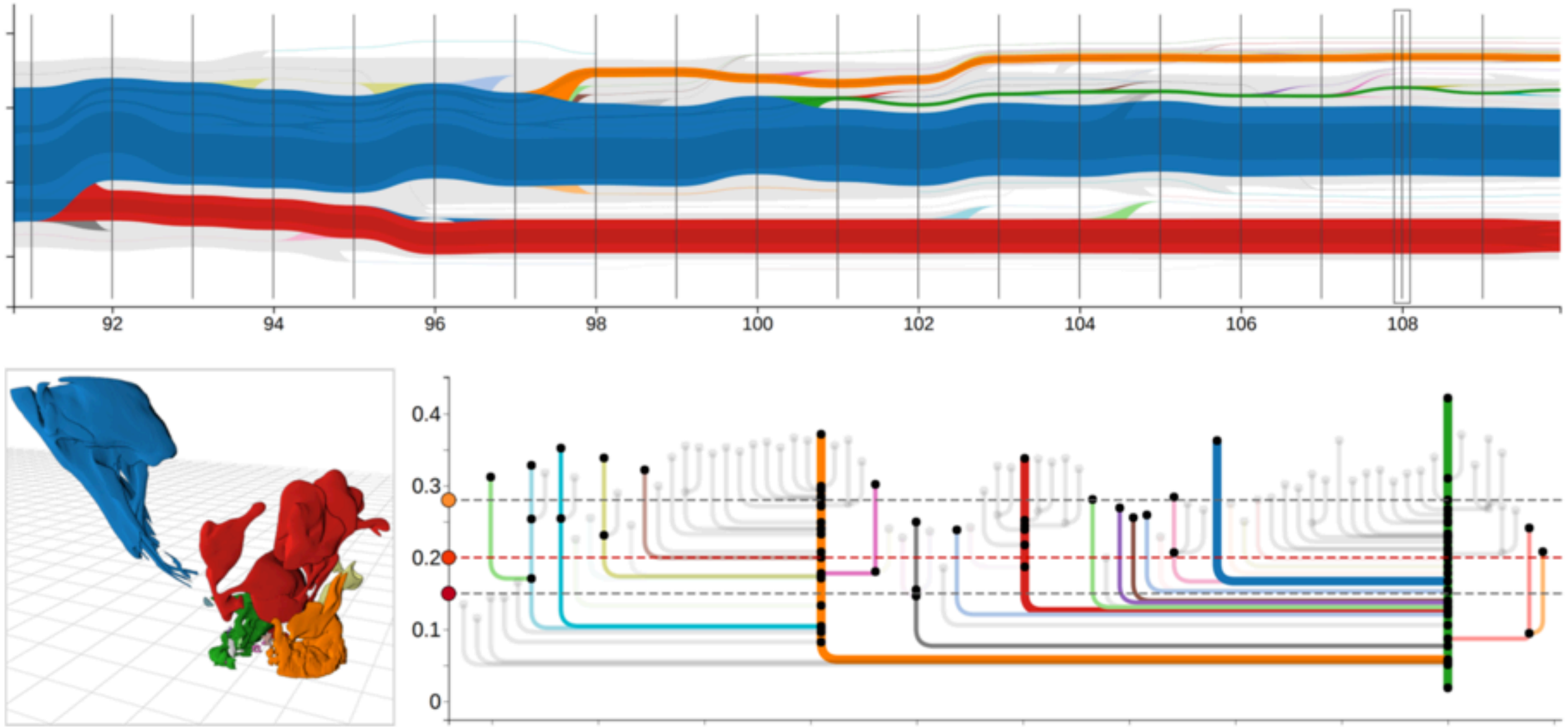
Use topological abstraction to draw a nested graph (y-axis) over time (x-axis).



[Lukasczyk et al., Nested Tracking Graphs, EuroVis 2017]

Tracking of Topological Features

Case study: Asteroid Ocean Impact



[Lukasczyk et al., Nested Tracking Graphs, EuroVis 2017]

Parallel Algorithms for Large-Scale Data

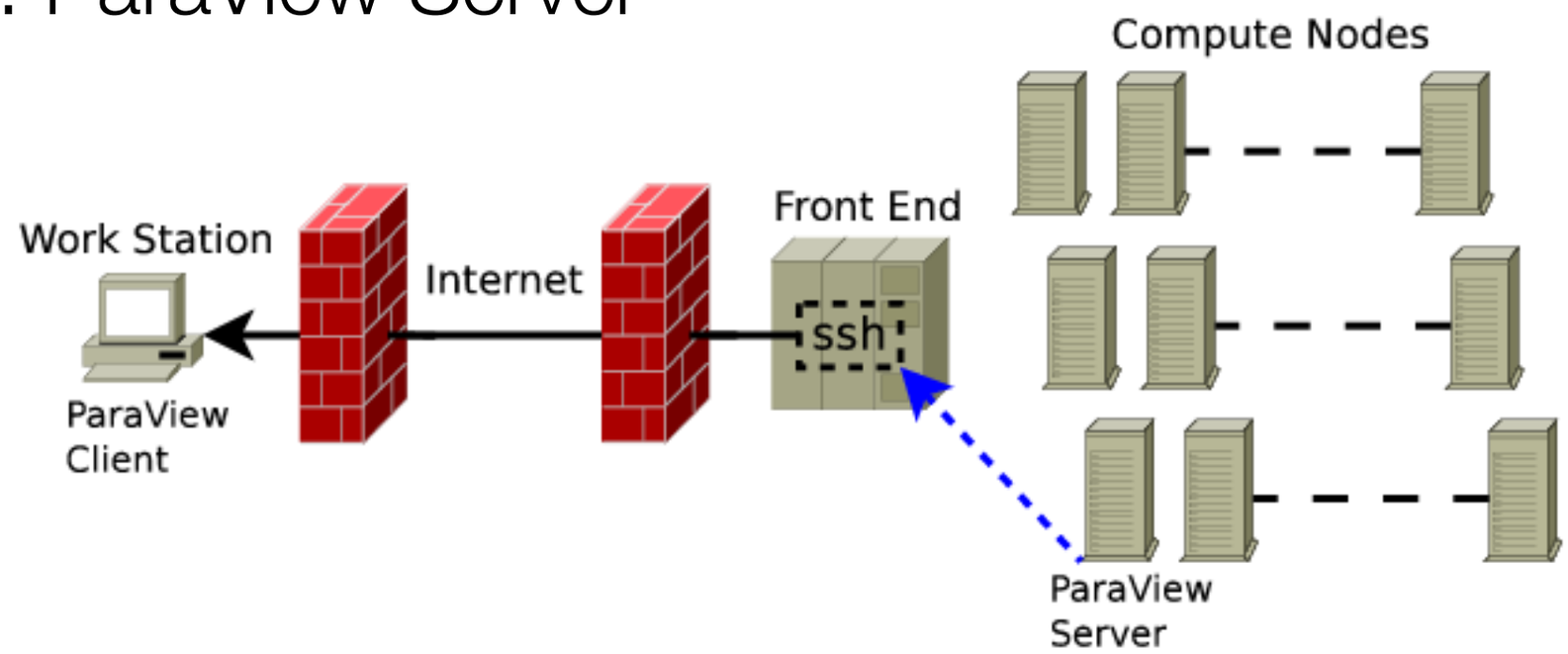
The Data Size Problem

Data is too large to be analyzed on a commodity workstation.

→ Analyze / visualize the data where it is stored,
directly on the cluster / supercomputer

→ **remote visualization**

Example: ParaView Server



Scalable Algorithms

Large-scale data: want to use supercomputing resources also for visualization, i.e. run visualization on many processors.

This requires **scalable, parallel** algorithms.

Scalable algorithms, in a nutshell (p = number of processors):

Time on one processor: $T_1(n)$
Time on p processors: $T_p(n)$

Overhead: $p * T_p(n) - T_1(n)$

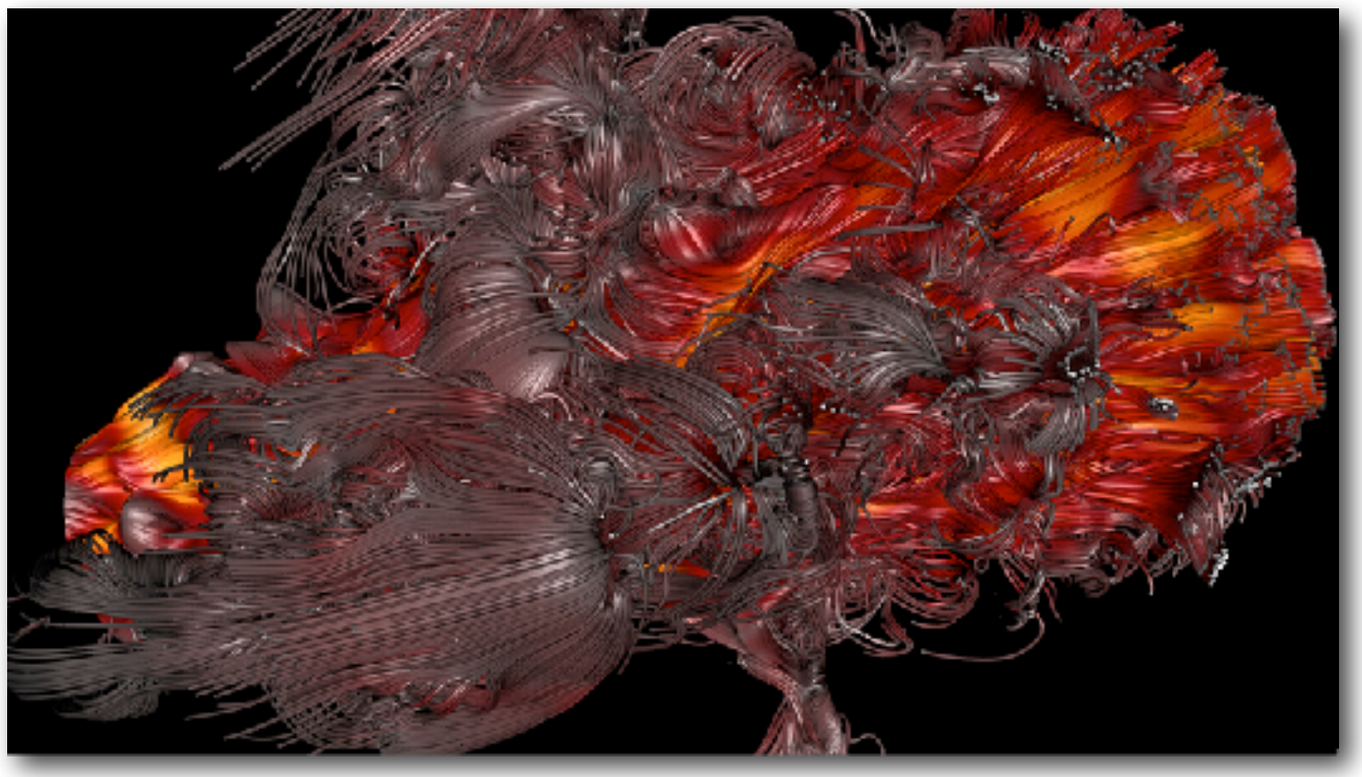
Scalable: overhead grows “slowly” (sub-linear) as a function of p.

Example: Integration-Based Visualization

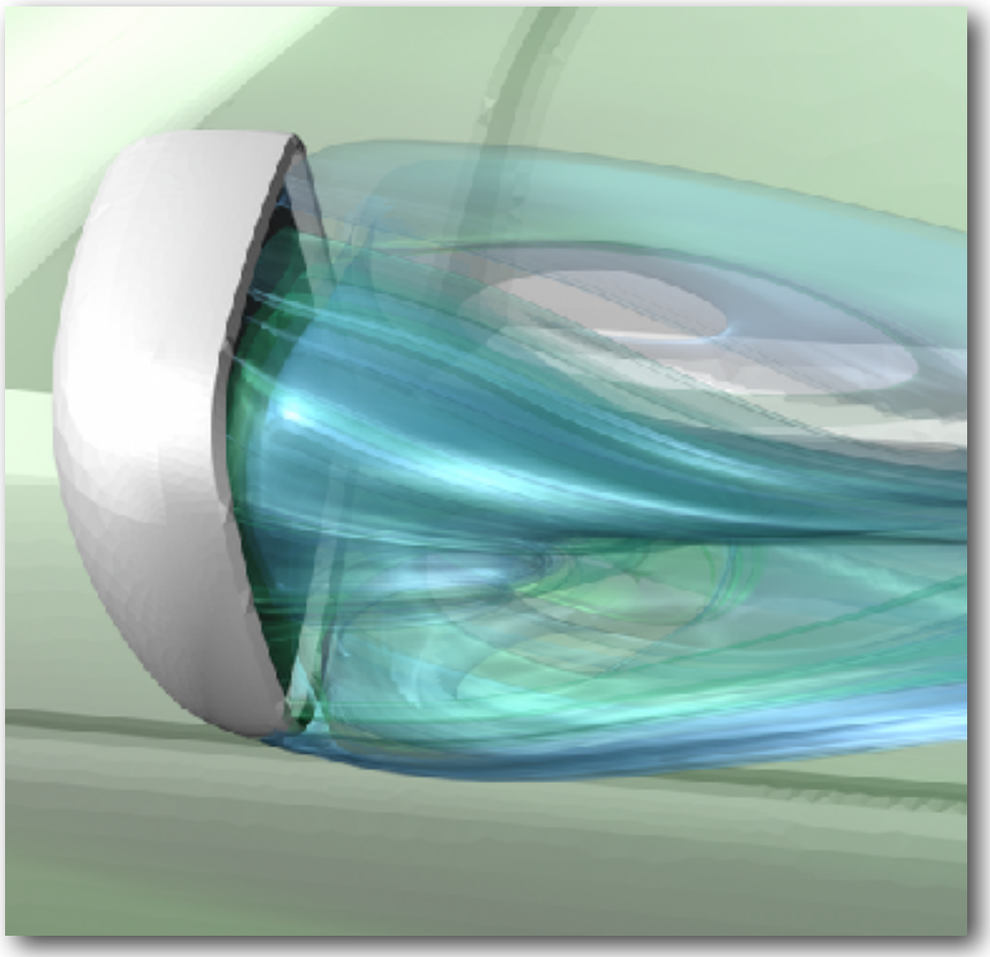
Integration-based Vector Field Visualization

- Integral Curves and Surfaces

20K STREAMLINES

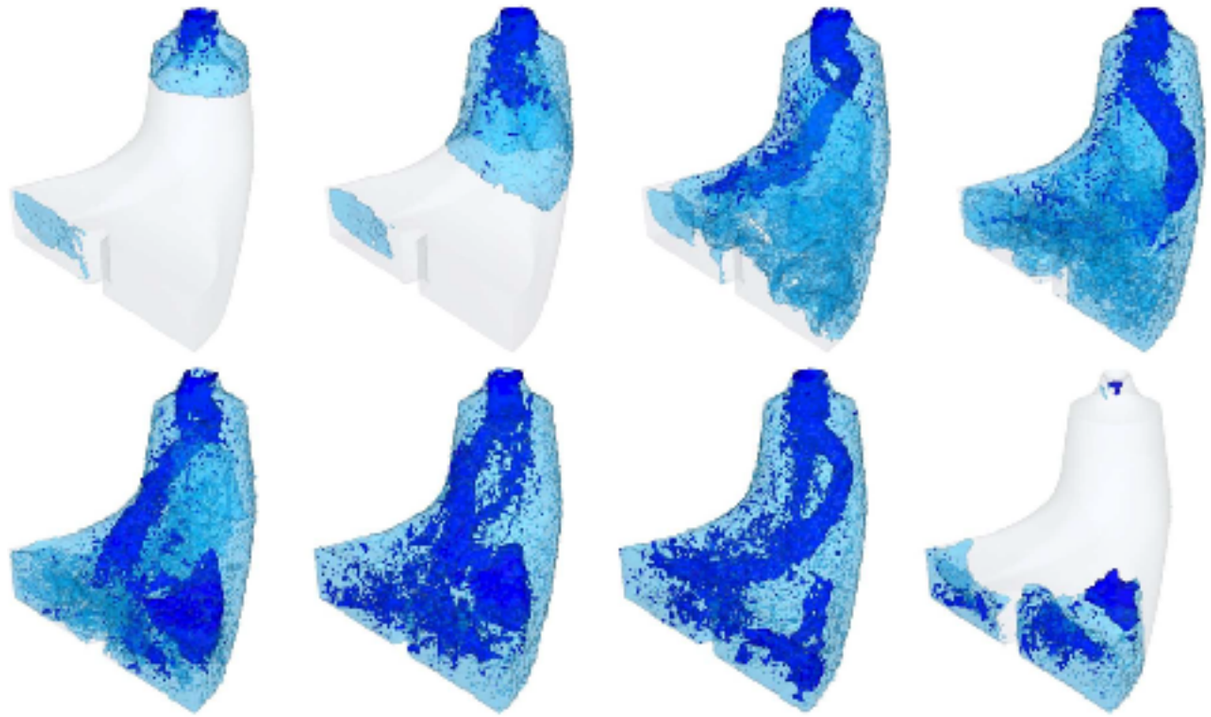


27K STREAMLINES



Example: Integration-Based Visualization

Pathline Predicates: integral curves in combination with Boolean predicates and set operations



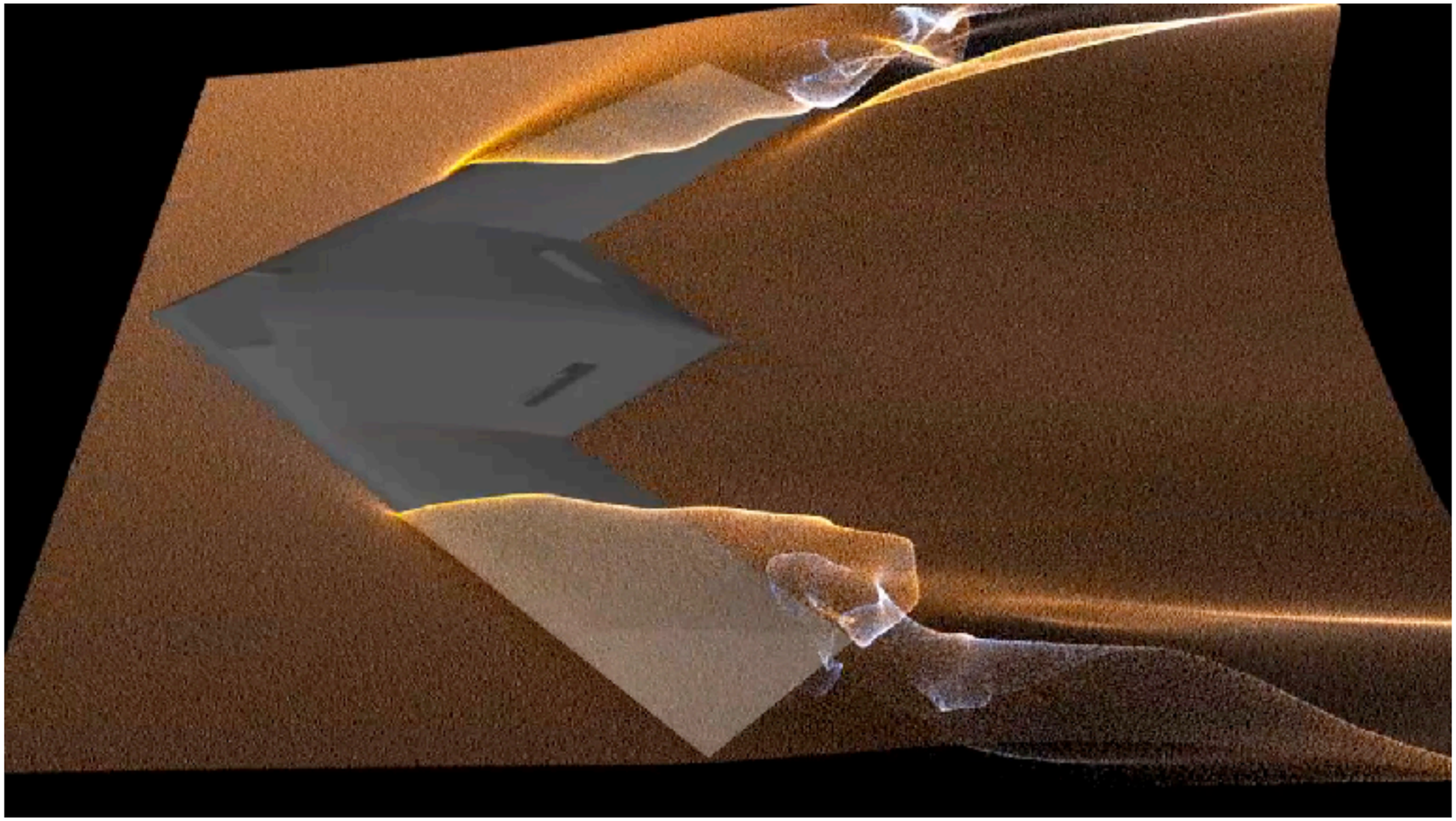
pathline predicates over 250,000 pathlines

Turbine dataset courtesy R. Peikert, ETH Zürich

Definition and extraction of features as relevant sets of curves.

T. Salzbrunn, **CG**, G. Scheuermann, J. Meyer. *Pathline predicates and unsteady flow structures.* The Visual Computer 24(12):1039–1051, 2008

Example: Integration-Based Visualization



(UAV data courtesy M. Rütten, DLR Göttingen)

Typical (naïve) flow visualization: 1M particles moving with a flow.

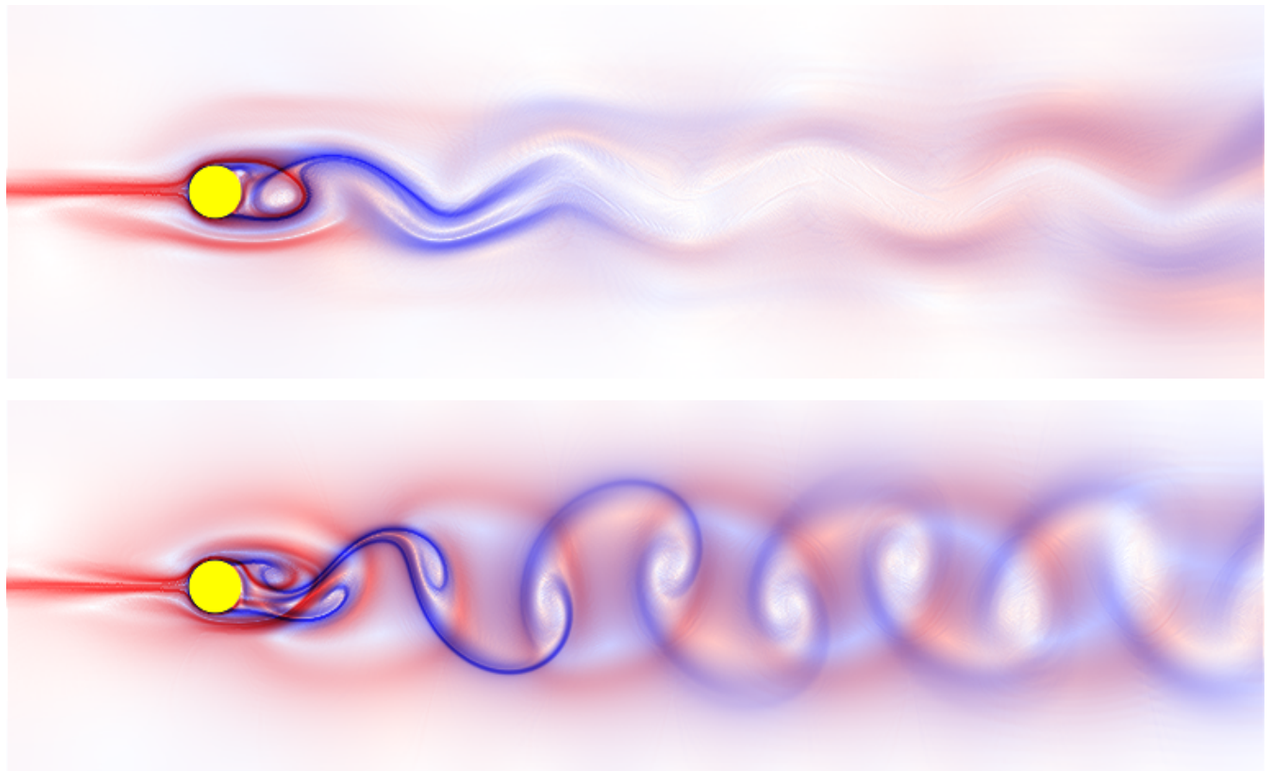
Real-time due to GPU parallelization (all data resides in GPU memory).

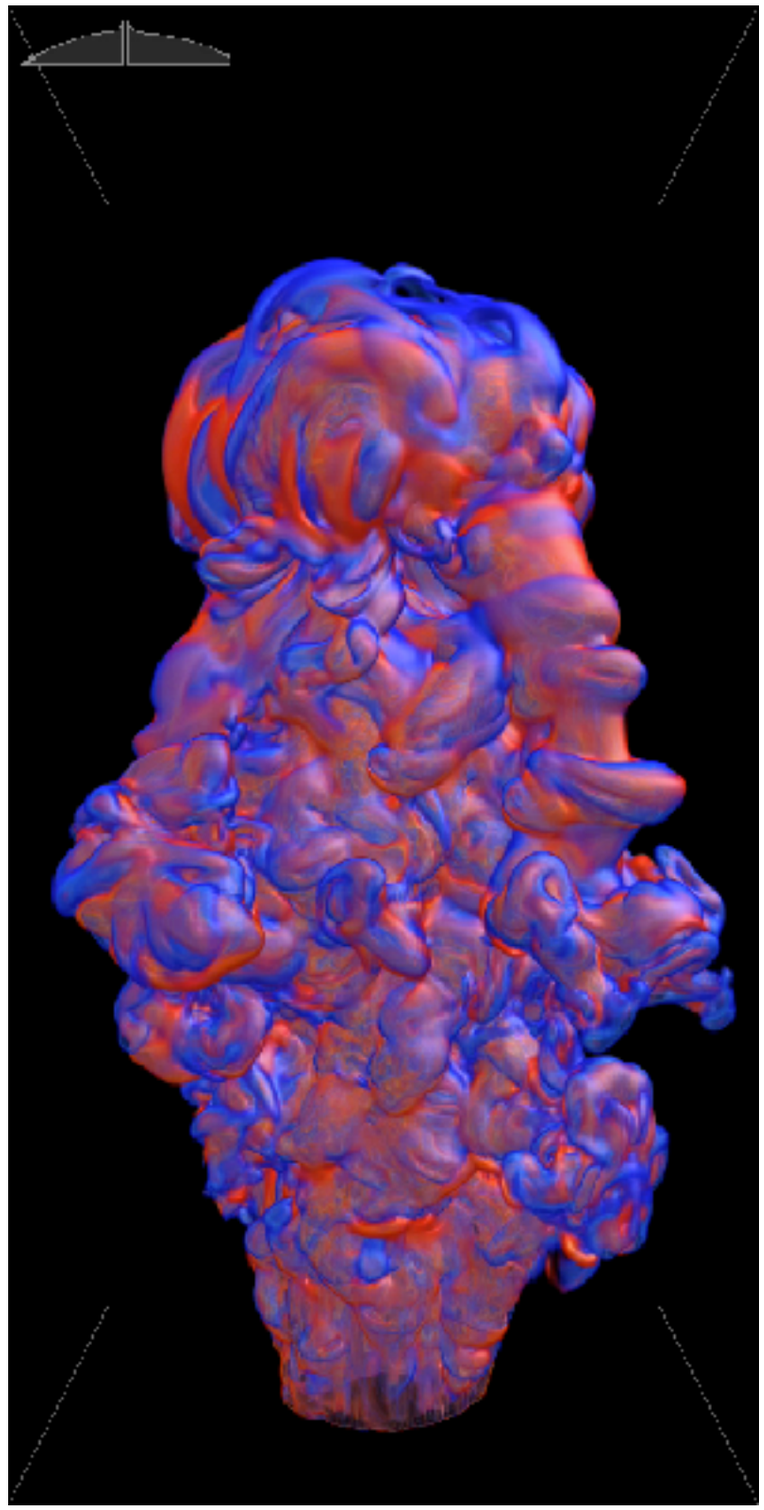
CG, K. I. Joy: *Fast, Memory-Efficient Cell Location in Unstructured Grids for Visualization*. IEEE TVCG 16(6):1541-1550.

Example: Integration-Based Visualization

Integration-based
Vector Field Visualization

- Lagrangian Methods





5 billion pathlines

Parallel Algorithms

Integral Curves / Particles: conceptually simple

- Trace particle through vector field,
  choice of numerical method and interpolation scheme

- Curves are independently computed,
  parallelization is straightforward for in-core data

Large data:

- Data in blocks, all blocks do not fit in core of single machine

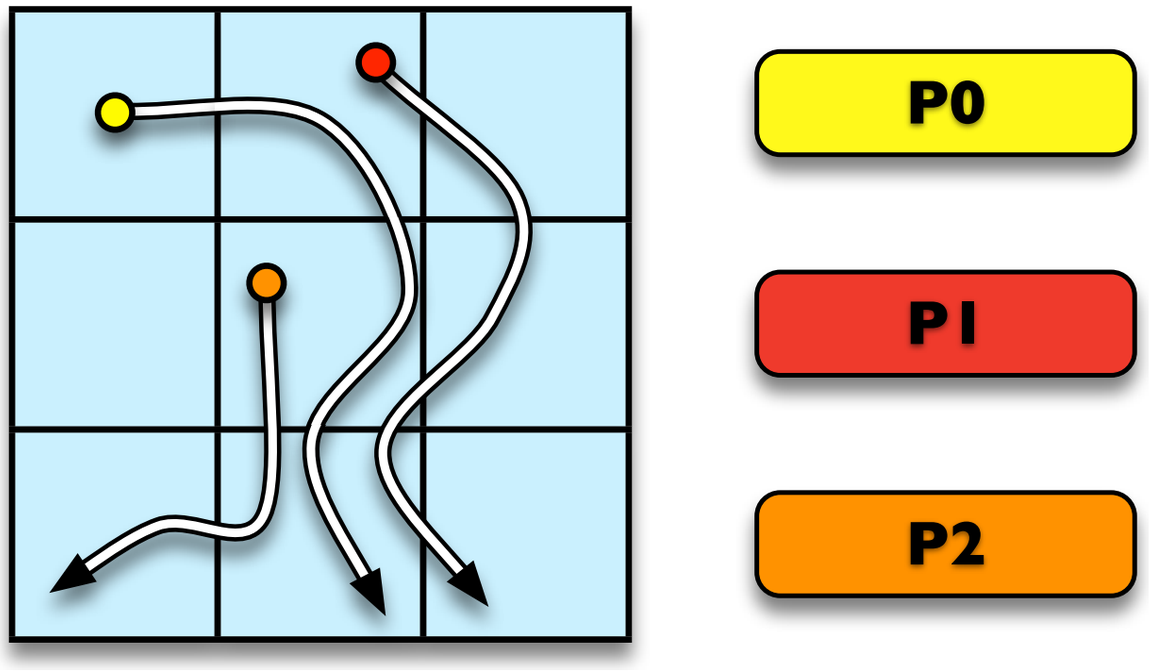
- Particles need to come together with correct
  data blocks to allow propagation.

Parallel / Distributed Integration

Two basic parallelization/distribution schemes:

- **Parallelize over Seeds** (== Curves)

  distributed computation, data on demand

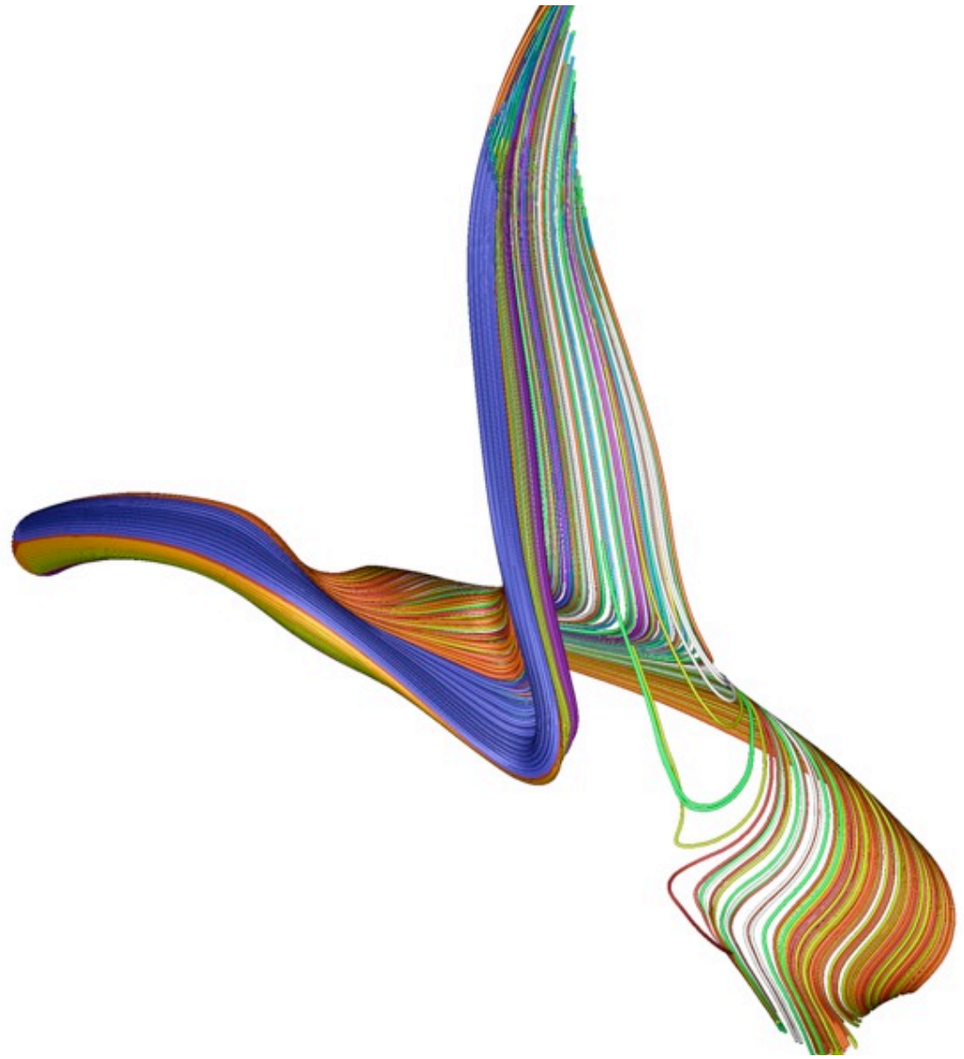


Parallel / Distributed Integration

Example:
Supernova Core Collapse (10^9 cells in 4096 blocks)

Parallelize over Seeds

→ lots of redundant I/O since same subset of data touched by many curves

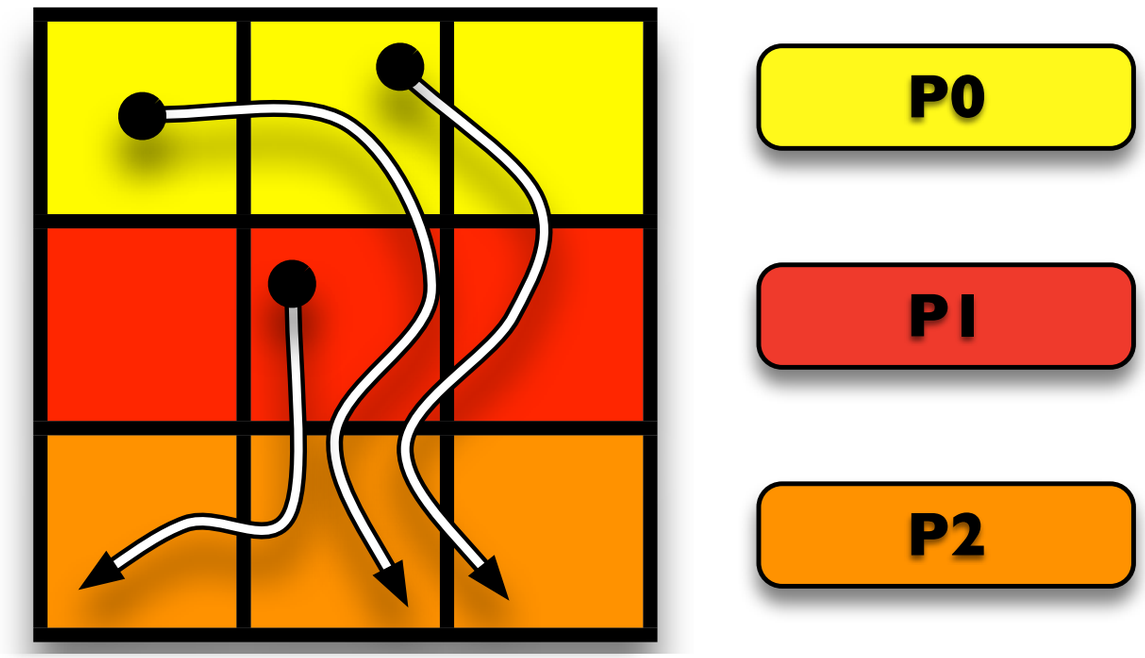


Parallel / Distributed Integration

Two basic parallelization/distribution schemes:

- **Parallelize over Blocks**

  distributed data, computation on demand

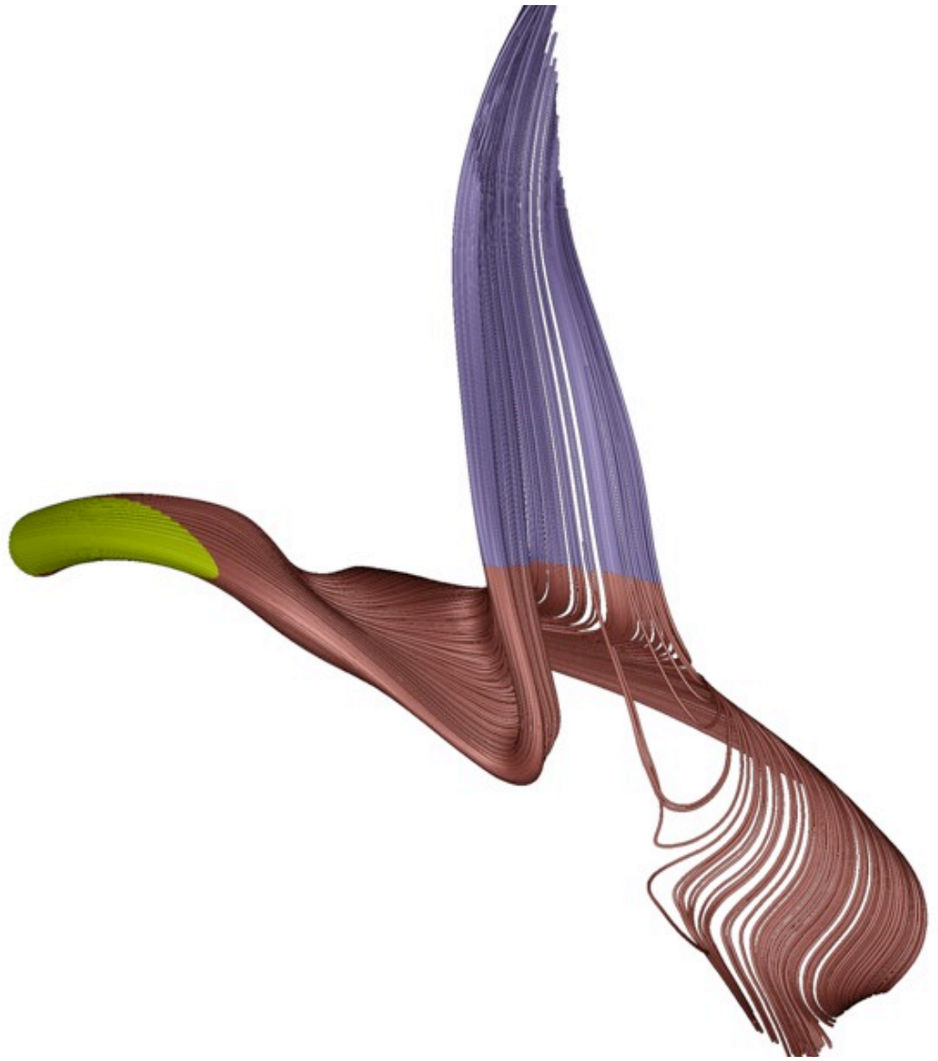


Parallel / Distributed Integration

Example:
Supernova Core Collapse (10^9 cells in 4096 blocks)

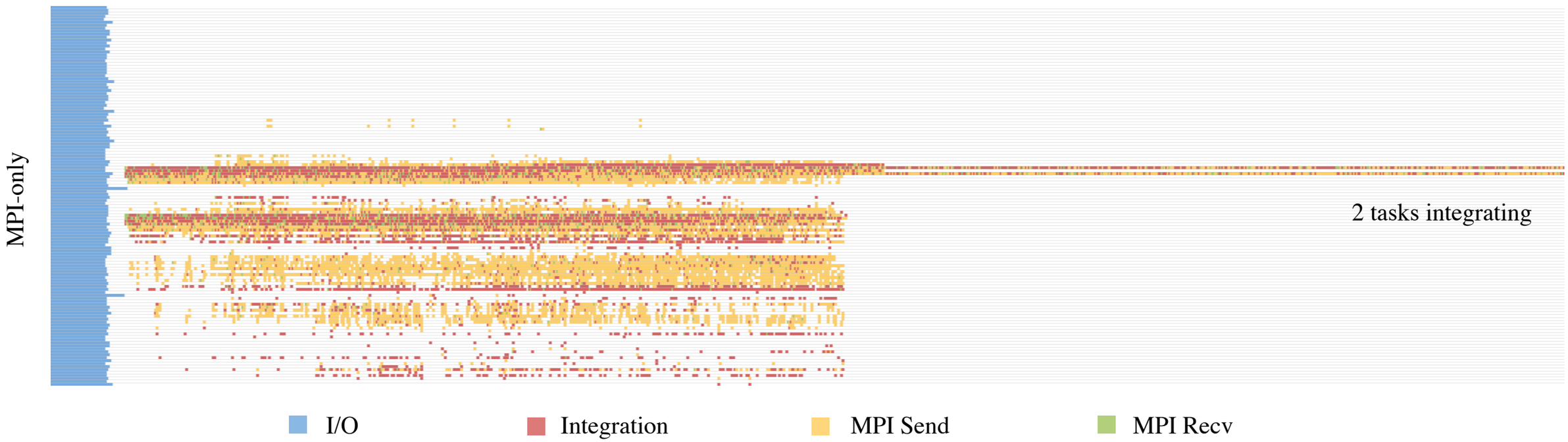
Parallelize over Blocks

→ only 3 processors used



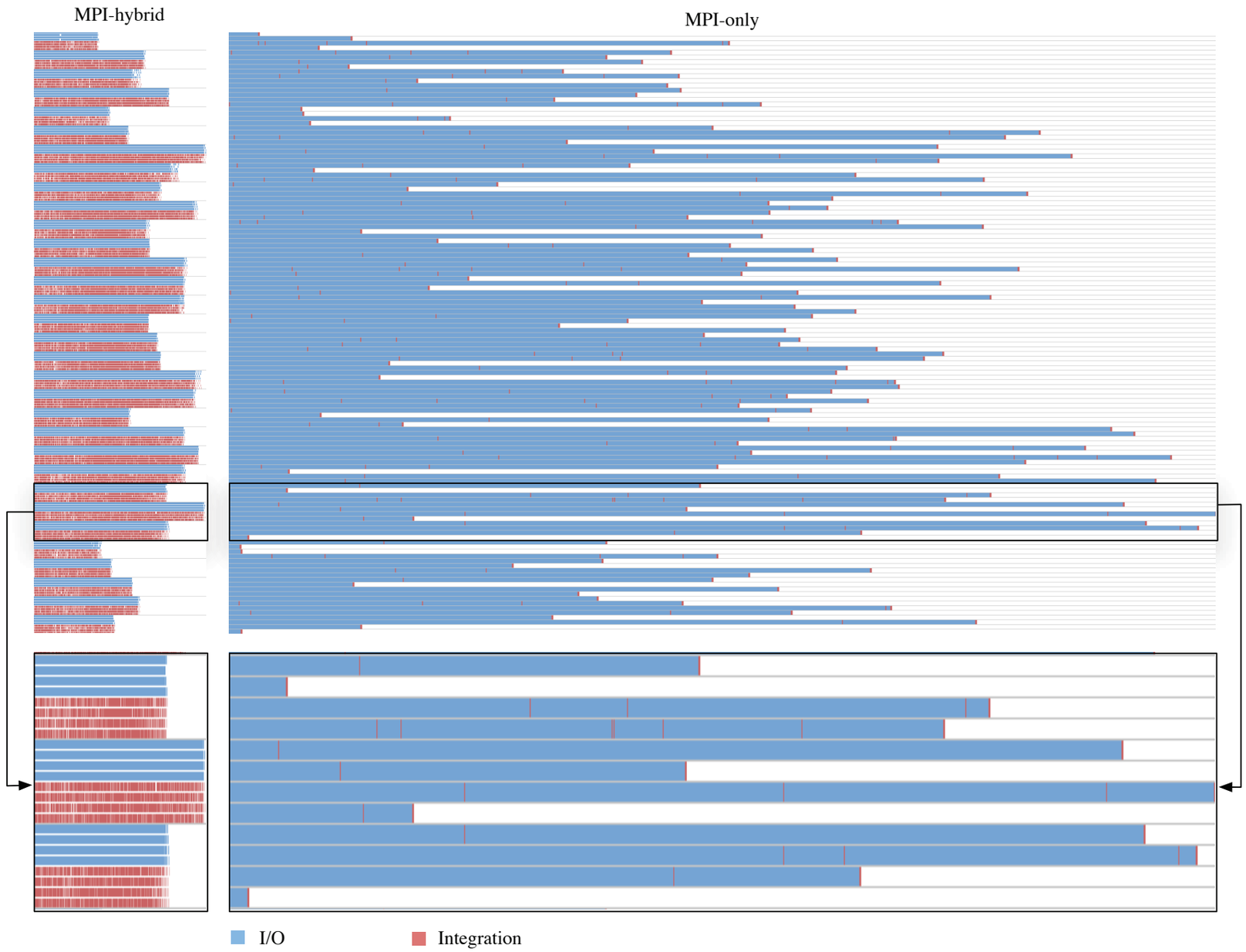
Parallel / Distributed Integration

Degenerate case Gantt chart



Better Parallel Integration

Adaptive Load Balancing and Hybrid Parallelization



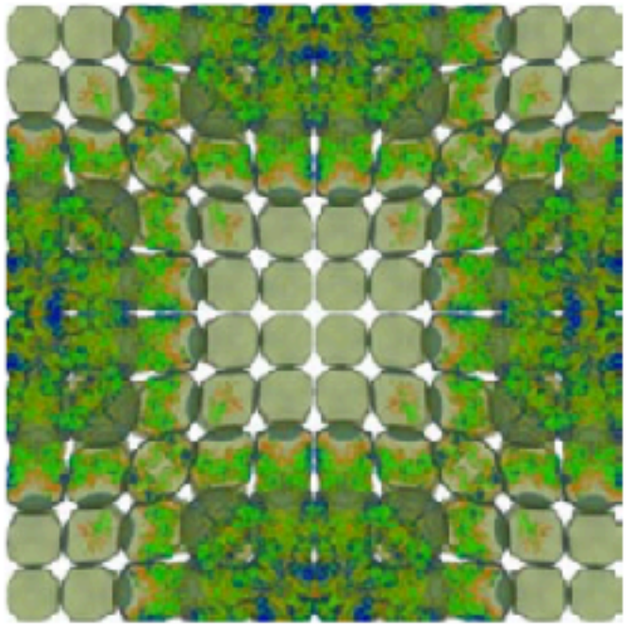
Pugmire, Garth, Childs, Joy, Bethel: *Scalable Computation of Integral Curves*. Proc. ACM Supercomputing, 2009.

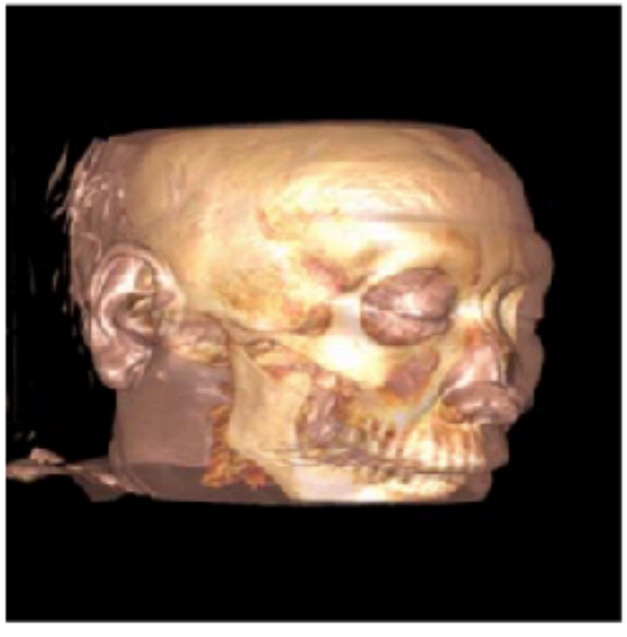
Parallel Algorithms

Parallelizing visualization algorithms can be **very** hard.

Interesting examples:

- Map-Reduce implementation of visualization algorithms (isosurface, volume rendering, …)





PPM Richtmyer-Meshkov instability (7.6GB)

#Isovalues	CLuE time 256M/256R	Cluster time 480M/480R	File Written
1	2min 55s	57s	1.78MB
2	2min 00s	57s	1.81MB
4	9min 35s	2min38S	35.26GB
8	10min 26s	2min37s	103.92GB
16	28min 29s	5min27s	252.38GB

SKULL (256MB)

#Isovalues	CLuE time 256M/256R	Cluster time 480M/480R	File Written
1	2min 22s	54s	1.94MB
2	2min 00s	56s	15.83MB
4	2min 02s	56s	31.22MB
8	2min 11s	58s	3.35GB
16	6min 06s	1min27s	18.74GB

[Vo et al., Parallel Visualization on Large Clusters using MapReduce, Proc. LDAV 2011]

Parallel Algorithms

Interesting examples:

- Parallel Contour Tree Computation
  [Carr et al.,*Hybrid Data-Parallel Contour Tree Computation*, dx.doi.org/10.2312/cgvc.20161299]

- Feature-Tracking in Large Simulations
  [Widanagamaachi et al., Tracking features in embedded surfaces: Understanding extinction in turbulent combustion, doi.org/10.1109/LDAV.2015.7348066]

- …

Community Effort: vtk-m

- “data-parallel” high-performance GPU implementations

Parallel Algorithms

Open-source tools with good general-purpose parallel implementations:

- Paraview
  paraview.org

- VisIt
  https://wci.llnl.gov/simulation/computer-codes/visit/

In Situ Techniques

I/O Bottleneck

The ability to generate data has increased tremendously; storing simulation output is becoming too expensive

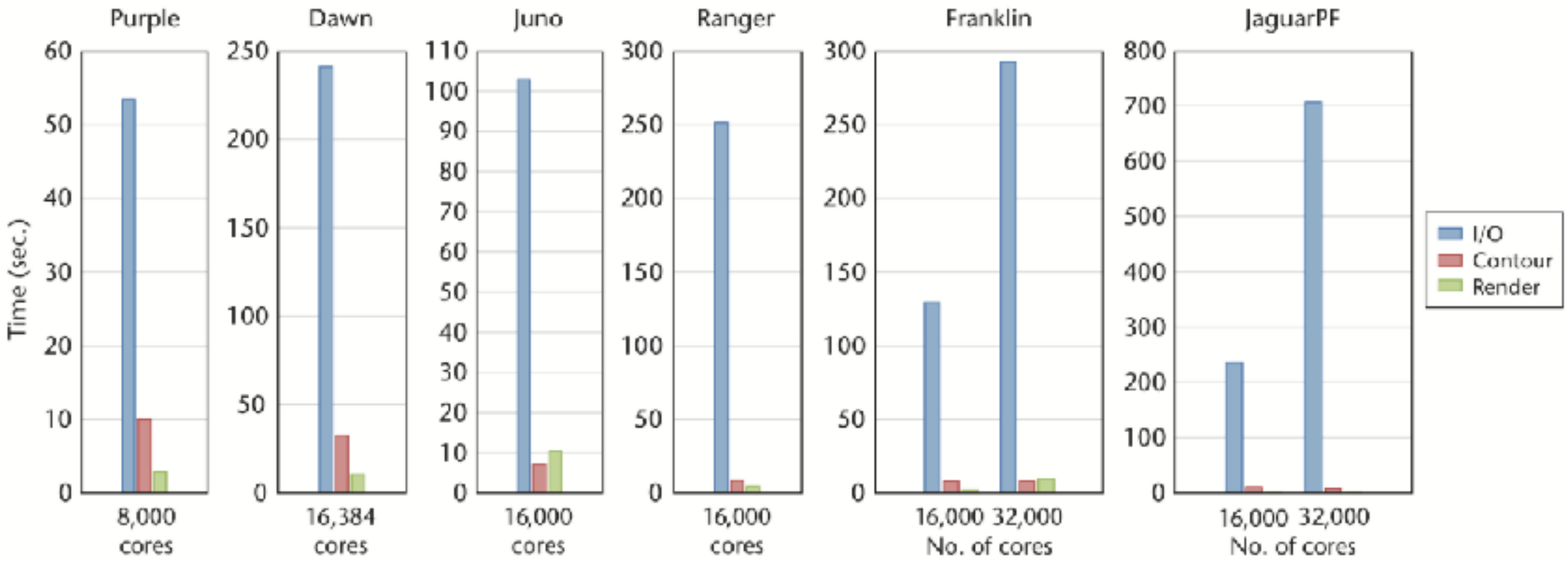
	2010	2018	Factor Change
System Peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Tf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 CPUs	1000 CPUs	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4444
Storage	15 PB	300 PB	20
Input/Output Bandwidth	0.2 TB/s	20 TB/s	100

from Vetter et al. "The key to Exascale Computing." HPC workshop, 2008

Visualization as pure **post-processing** is no longer practical.

In Situ Visualization

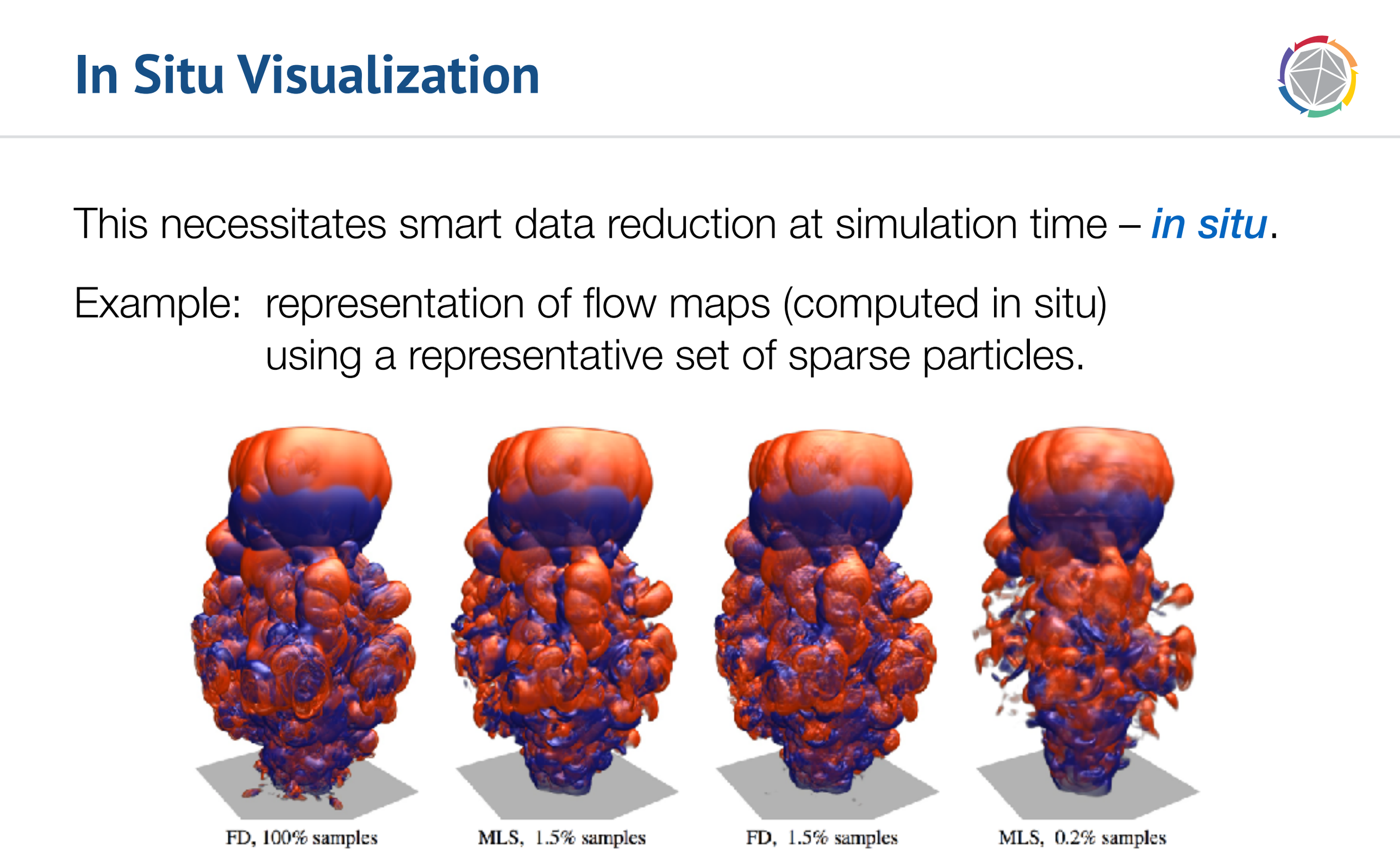
Re-thinking visualization as **post-processing**: for many practical problems, storing simulation output has become too expensive.



[from Childs et al.: *Extreme scaling of production visualization software on diverse architectures.* IEEE CG&A 30(3):22-31, 2010

I/O bandwidth does not scale with CPU & memory performance.

In Situ Visualization



This necessitates smart data reduction at simulation time – ***in situ***.

Example: representation of flow maps (computed in situ)
using a representative set of sparse particles.

FD, 100% samples | MLS, 1.5% samples | FD, 1.5% samples | MLS, 0.2% samples

A. Agranovsky, **CG**, K. I. Joy: *Extracting Flow Structures using Sparse Particles*. In Proc. Vision, Modeling, Visualization, 2011.

A. Agranovsky, D. Camp, **CG**, E. W. Bethel, K. I. Joy, H. Childs: *Improved Post Hoc Flow Analysis Via Lagrangian Representations.* In Proc. Large-Data Analysis and Visualization, 2014 (best paper)

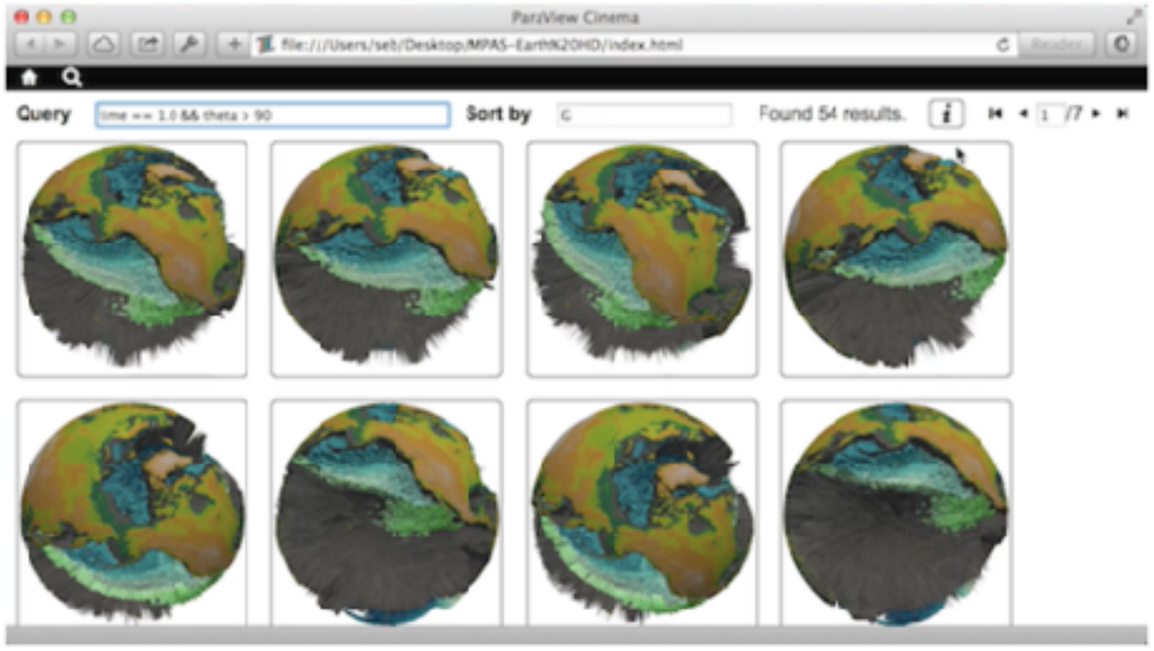
In Situ Visualization

Even more extreme: execute visualization algorithms (e.g. feature extraction) during simulation.

Example: Paraview Cinema (Ahrens et al. 2014)

- pre-specify all visualization parameters and store only images

- preserve limited flexibility by storing many different combinations of visualizations and viewpoints

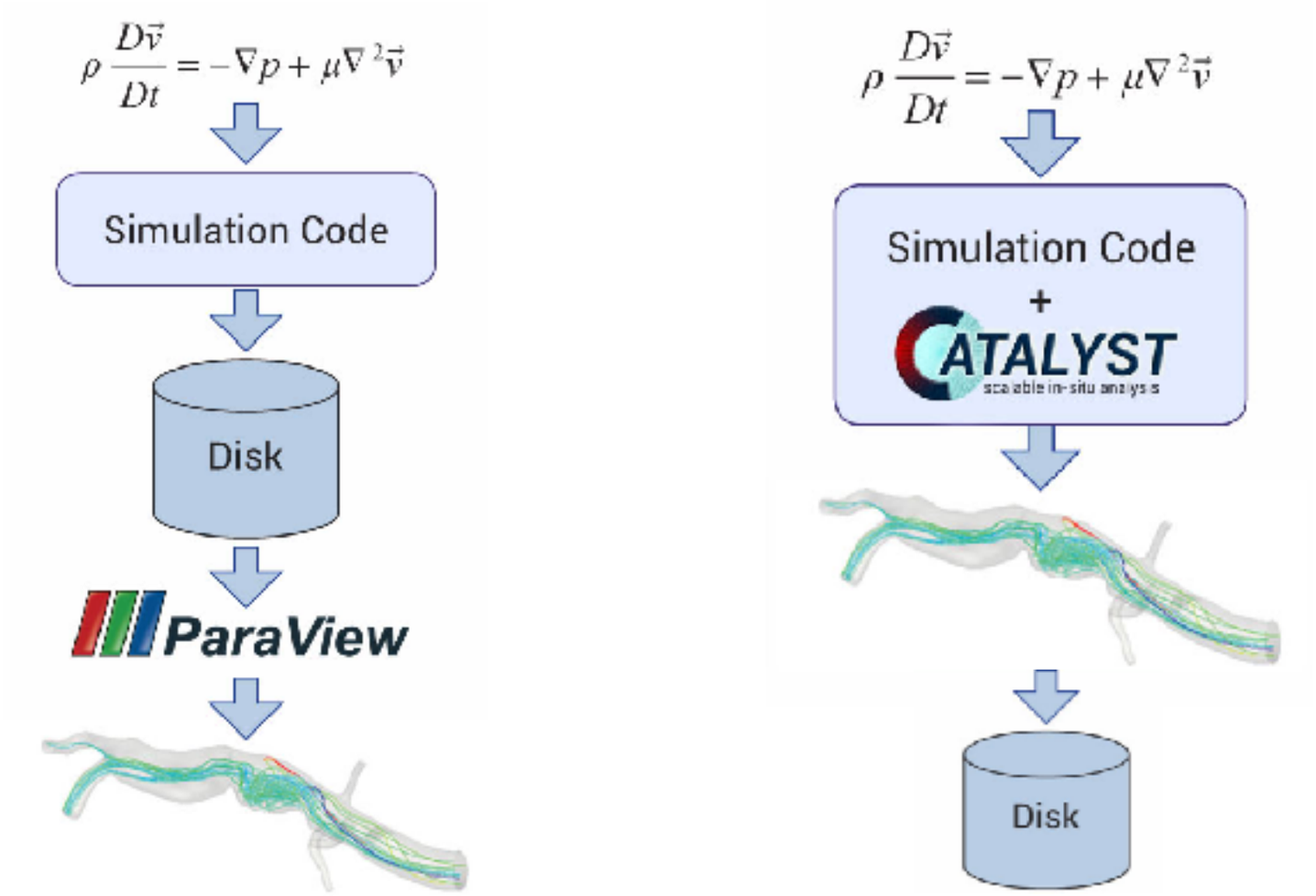


from Ahrens et al.: In Situ MPAS-Ocean Image-based Visualization. In Proc ACM Supercomputing (SC'14).

Video

In Situ Visualization

In situ visualization: compute visualization while simulating.



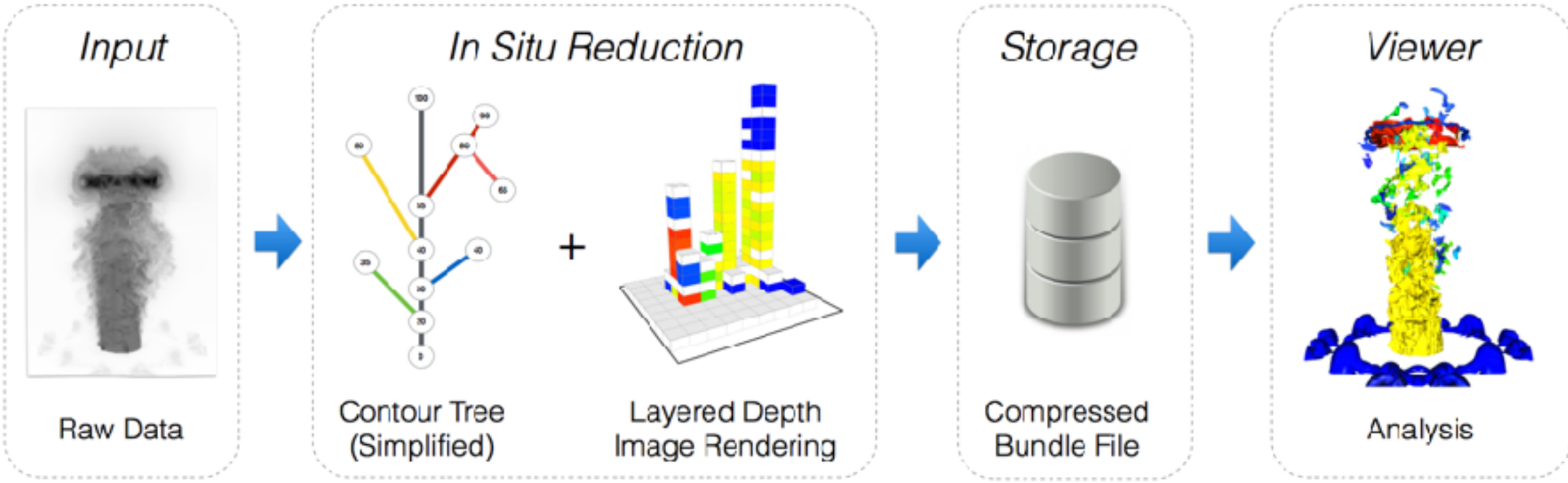
classical workflow

in-situ workflow

Problem: very limited interaction.

In Situ Visualization

Feature-based visualization example: combine **image-based approach** with **topological feature extraction.**

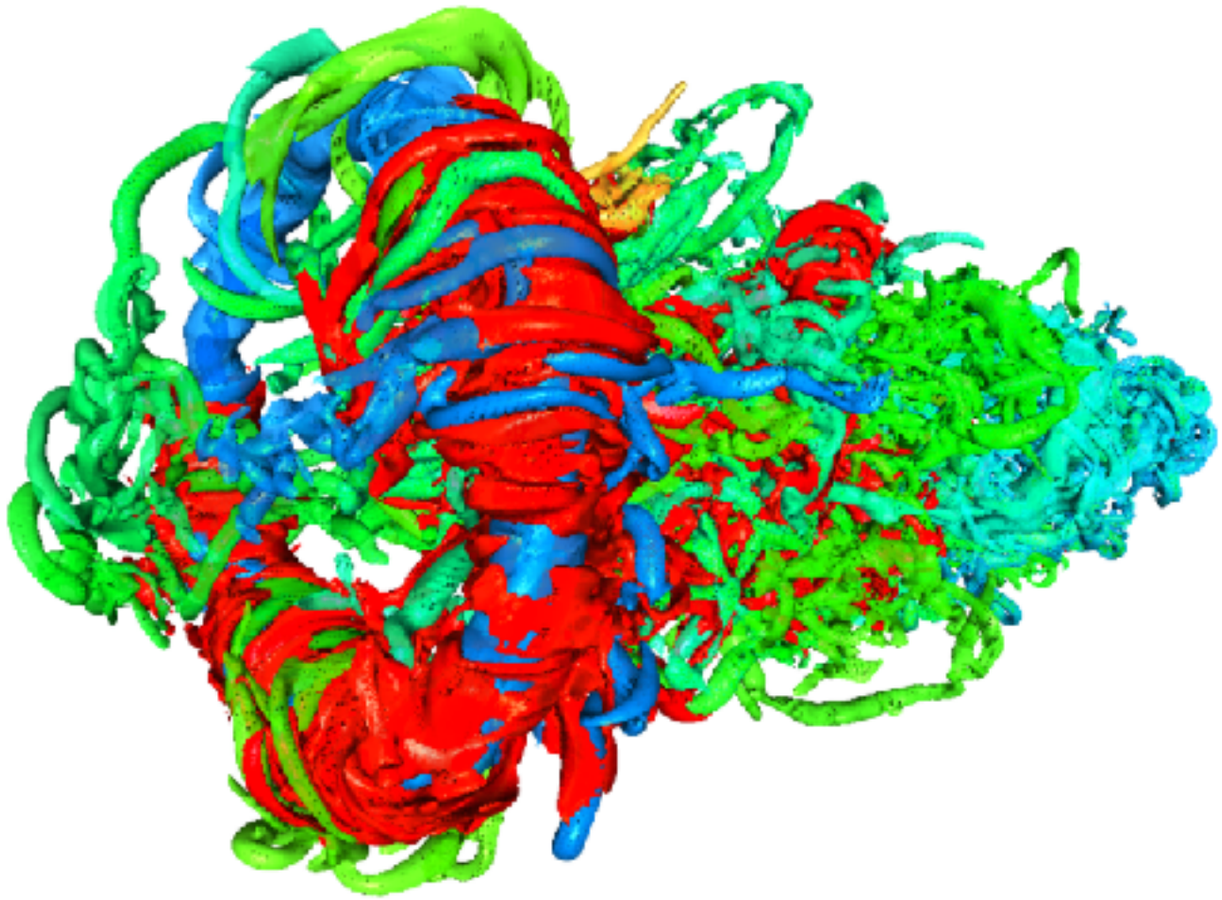


Raw data is converted into a topological representation, simplified, and rendered into a **layered depth image**.

T. Biedert, CG: Contour Tree Depth Images For Large Data Visualization. In Proc. EGPGV, 2015.

Contour Tree Depth Images

The depth image stores only a fixed number of the most important topological features, after simplification.

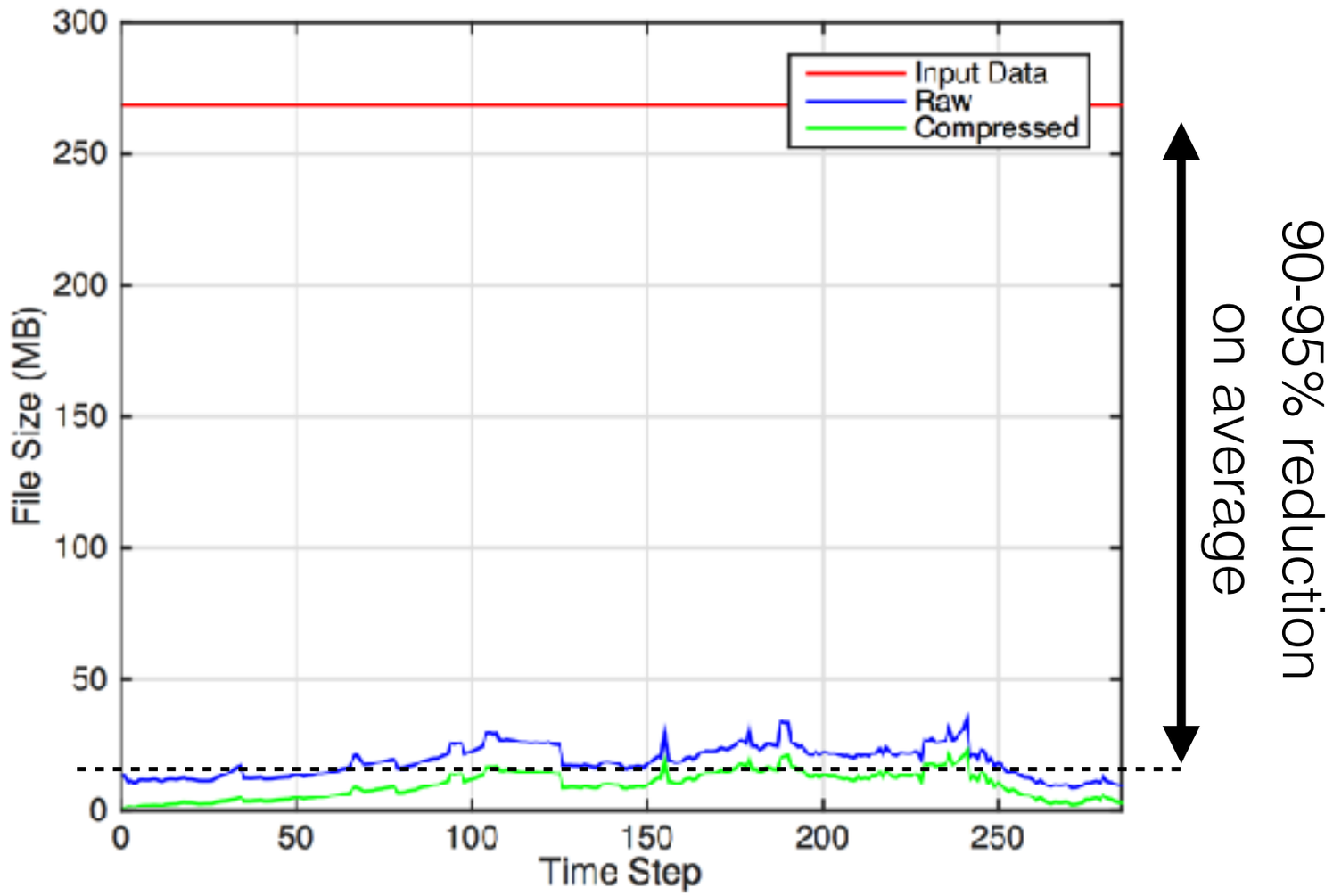


Vortices in a jet flow characterized via λ_2–criterion
(**100** branches stored, **1.5M** branches before simplification)

T. Biedert, **CG**: Contour Tree Depth Images For Large Data Visualization. In Proc. EGPGV, 2015.

Contour Tree Depth Images

Large storage savings are achievable.

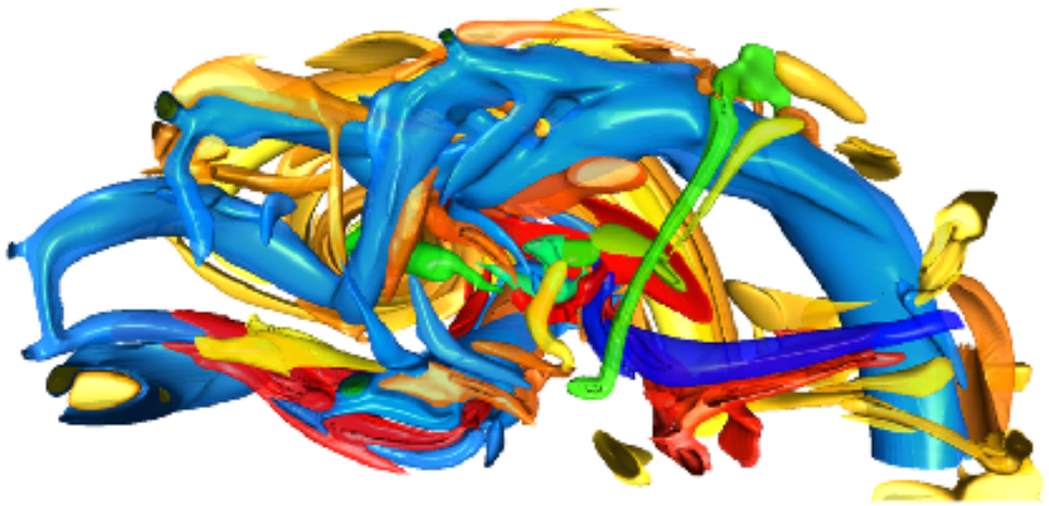


Data can be copied to and visualized on a standard PC.

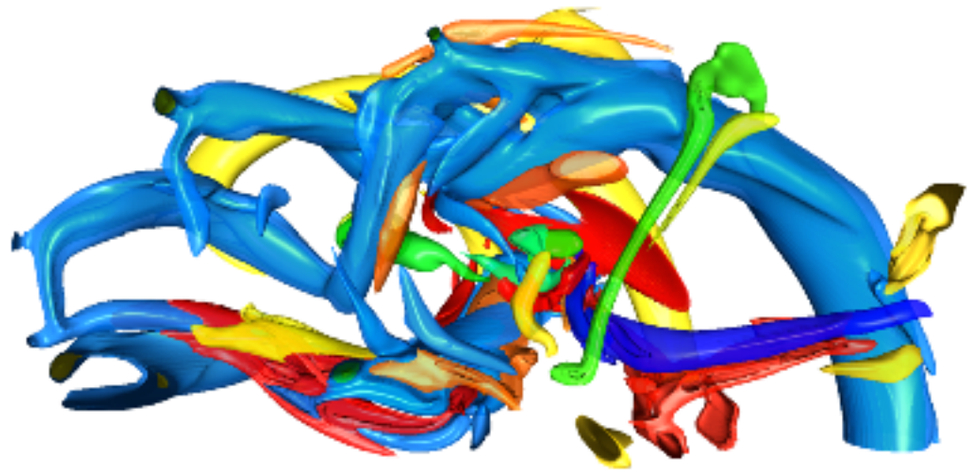
T. Biedert, **CG**: Contour Tree Depth Images For Large Data Visualization. In Proc. EGPGV, 2015.

Contour Tree Depth Images

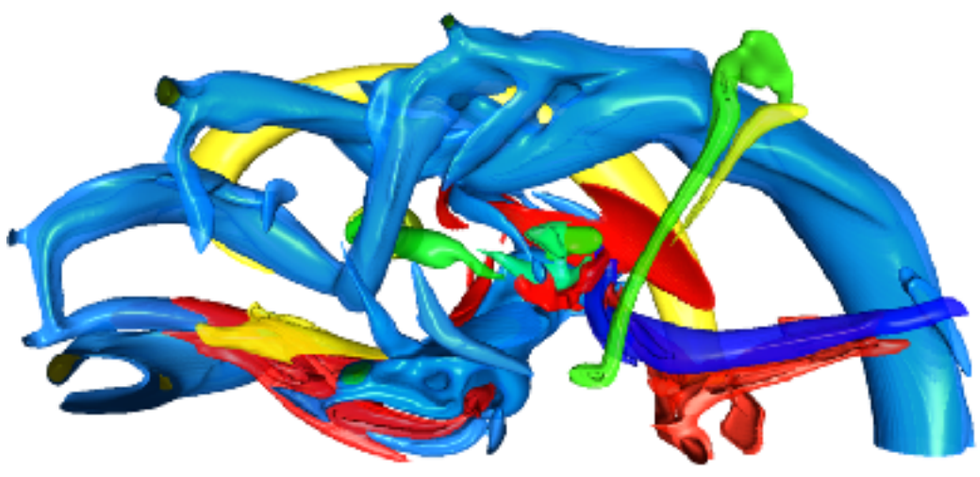
Flexibility for further filtering and simplification is preserved.



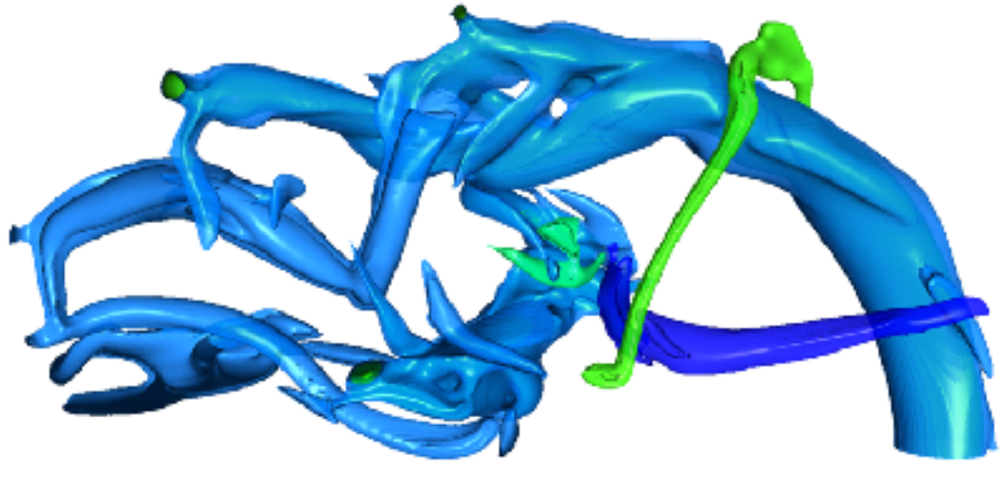
100 features



50 features



20 features



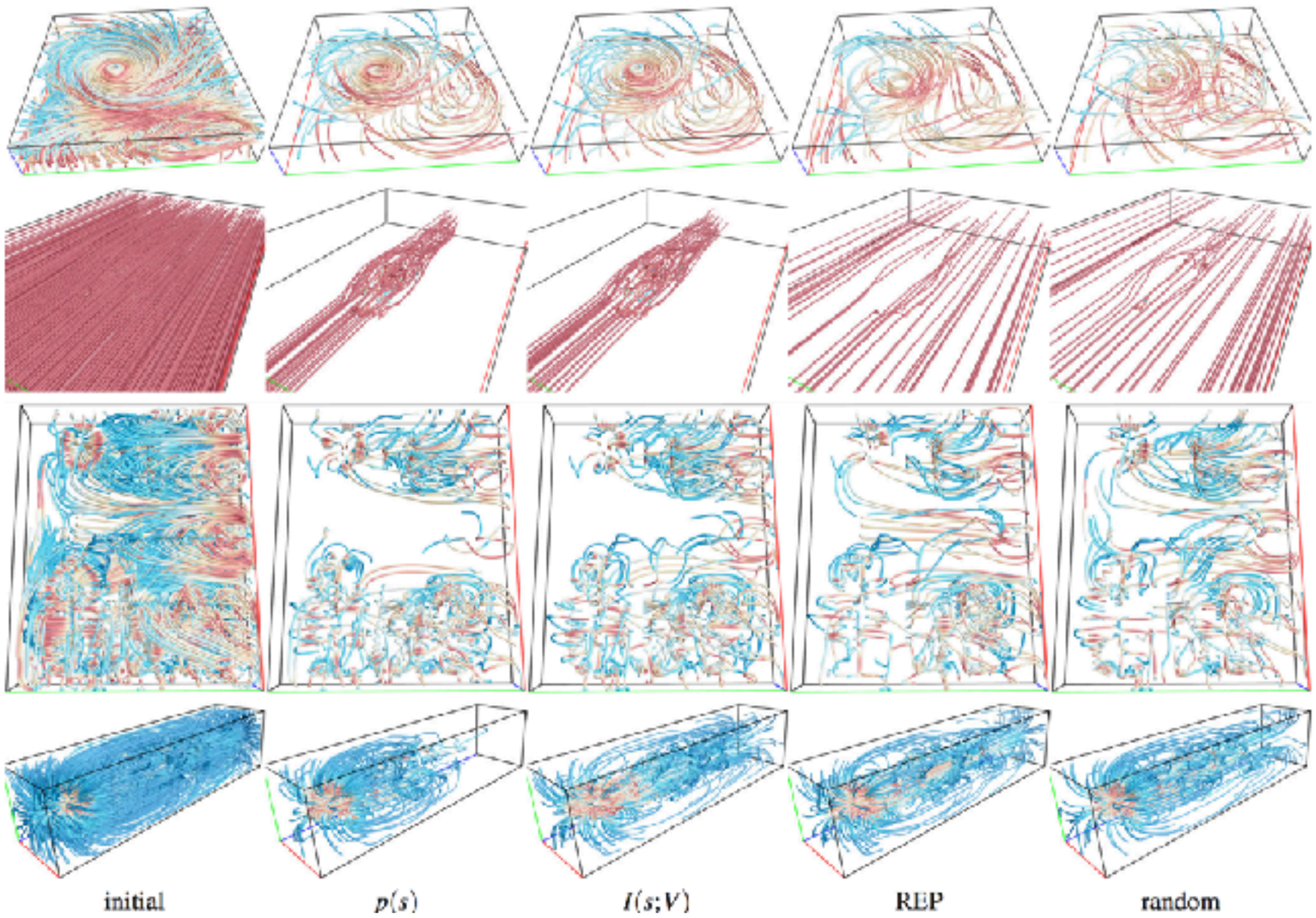
10 features

T. Biedert, **CG**: Contour Tree Depth Images For Large Data Visualization. In Proc. EGPGV, 2015.

In Situ Visualization

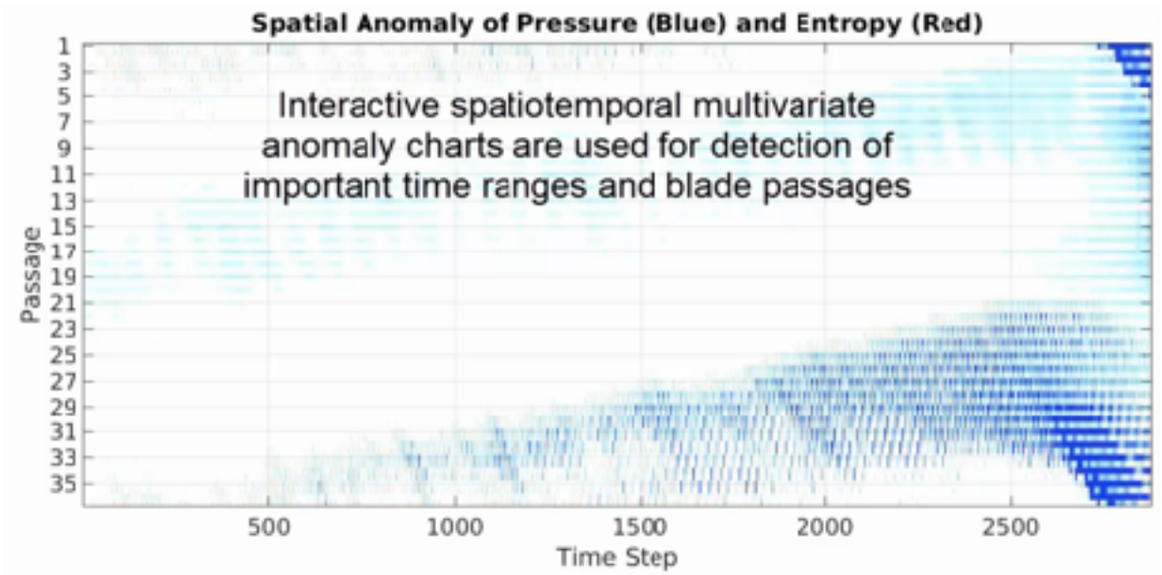
Various techniques to generate “good” images.

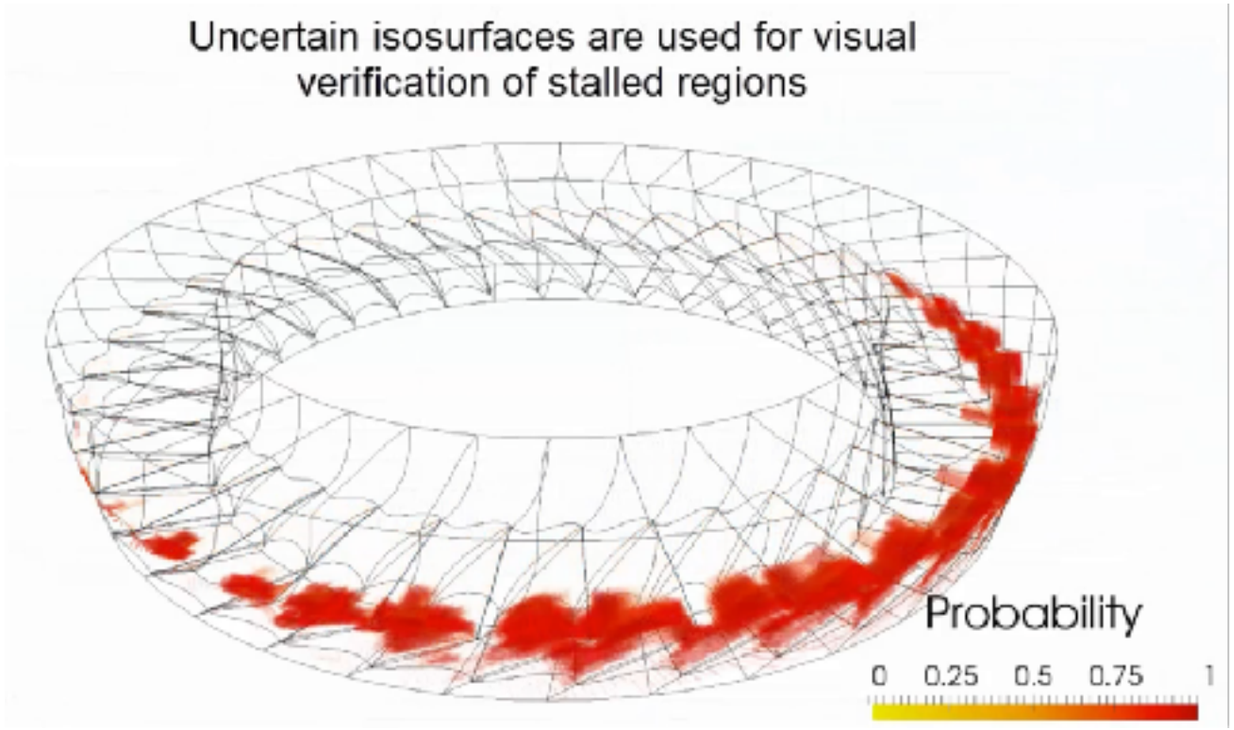
E.g. select “interesting” streamlines using information theory



In Situ Visualization

E.g. using distribution analysis to select interesting data subsets and store only these (data compression).





Dutta et al., In Situ Distribution Guided Analysis and Visualization of Transonic Jet Engine Simulations, IEEE TVCG 23(1):1077-2626, 2016

And many many more techniques for very specific scenarios.

Conclusion

Conclusion

Visualization of modern datasets (in the sense of this talk) is not a straightforward undertaking.

Research is ongoing on which techniques are general.

Axes to consider are:

- Visual scalability
- Cognitive scalabilty
- Computational scalability

Not addressed today: ensemble analysis, uncertainty visualization, high-dimensional point data, etc.