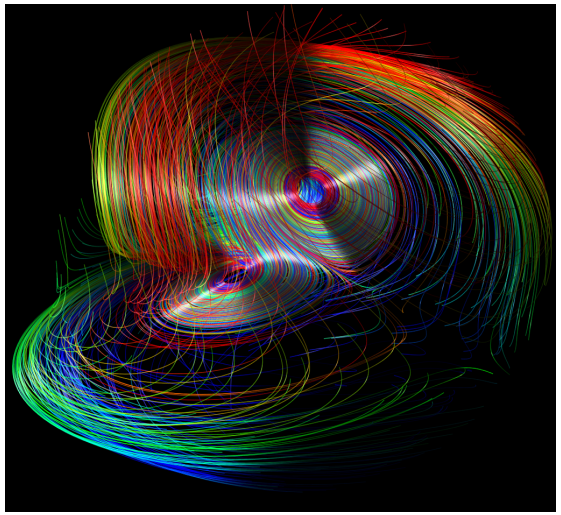
Visualization & Interface Design Innovation

Parallel Clustering for Visualizing Large Scientific Line Data

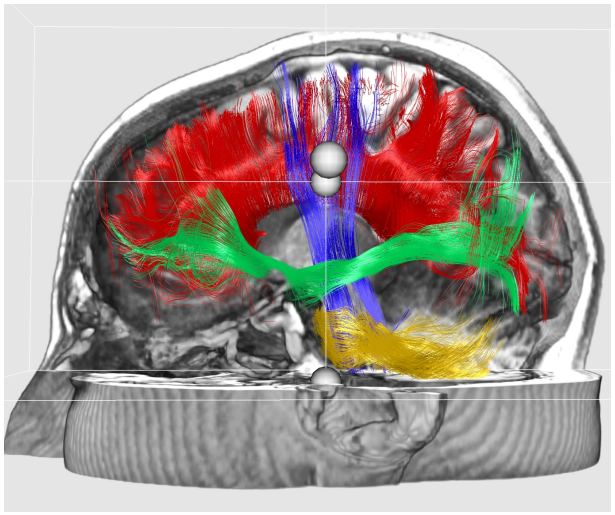
Jishang Wei, *University of California, Davis*
Hongfeng Yu, *Sandia National Laboratories*
Jacqueline H. Chen, *Sandia National Laboratories*
Kwan-Liu Ma, *University of California, Davis*

Background

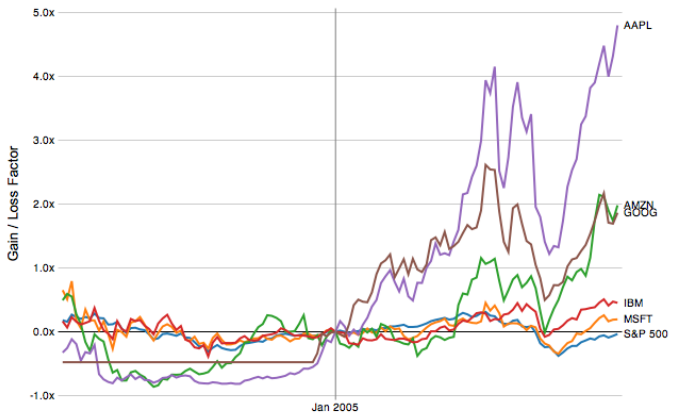
- Line data in scientific simulations and experiments
  - Line: an ordered sequence of multi-dimensional data points
  - Examples: vector field lines, white matter fibers, time series curves



O. Mallo, R. Peikert, C. Sigg, F. Sadlo, Illuminated Lines Revisited, 2005



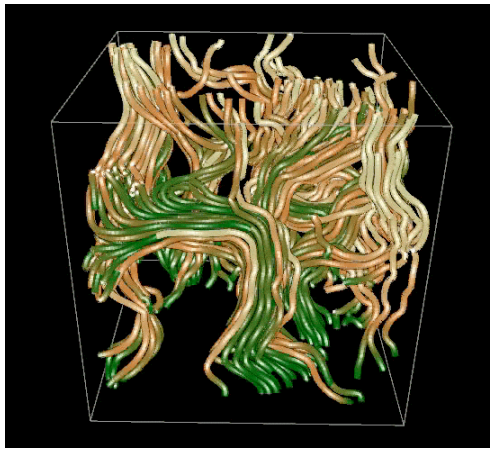
Generated by Pierre Fillard, Neurospin CEA



Jeffrey Heer, Michael Bostock, and Vadim Ogievetsky, A Tour Through the Visualization Zoo, 2010

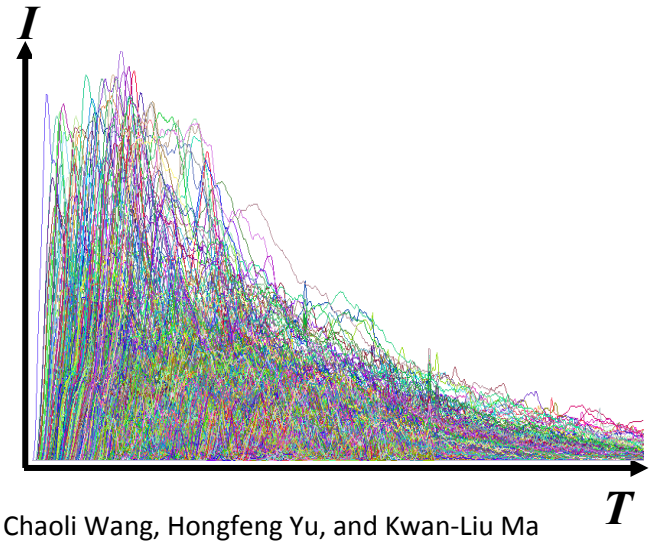
Motivation

- Challenges to visualize large line data
  - Visual clutter, clustering first, then visualizing
  - Large data, using a parallel machine to handle heavy workload
- Our contribution
  - A parallel design of model-based clustering for categorizing and visualizing large line data with multiple CPUs and GPUs



http://www.absoluteastronomy.com/topics/Drishti

O'Donnell. Cerebral White Matter Analysis Using Diffusion Imaging. 2006.



Chaoli Wang, Hongfeng Yu, and Kwan-Liu Ma Importance-Driven Time-Varying Data Visualization. 2008

Model-based Clustering

- What is model-based clustering
  - Assume that data can be divided into K groups, and each has a probabilistic model to describe the data within it
  - Recover model parameters from data
  - Assign a data object to a cluster with highest probability
- Why is model-based clustering
  - Cluster lines of different lengths
  - Process large data efficiently
- Model-based clustering of line data
  - Polynomial regression model
  - Recover model parameters using Expectation-Maximization algorithm

Parallel Model-based Clustering

- Distribute line data to multiple compute nodes
  - Keep workload balanced and minimize communication costs between compute nodes
  - Use a sorted balancing algorithm to ensure the total number of data points on each compute node roughly the same
- Preprocess line data on each compute node
  - Smooth and sample local lines on each compute node
  - Use GPUs to accelerate the preprocessing

Parallel Model-based Clustering

- Cluster lines using multiple CPUs
  - On each compute node, Initialize K component model parameters
  - Iterate between two steps
    - Expectation step: on each compute node, estimate local lines' probabilistic membership in different clusters
    - Maximization step: on each compute node, calculate the K model parameters globally
  - Assign each local line to a cluster with highest membership probability on each CPU node

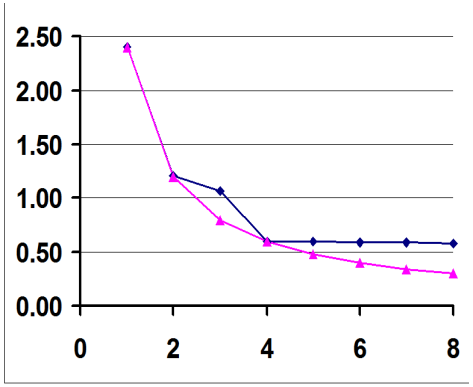
Experiment Settings

- Cluster: 8 computer nodes, each node contains
  - One Intel quad-core 3.00GHz CPU with 4GB of memory
  - One NVIDIA GeForce GTX 285 GPU.
- Datasets:
  - 10,000 streamlines from the vector field of a solar plume simulation
  - 1,000,000 time series curves correlating multiple variables generated from a combustion simulation

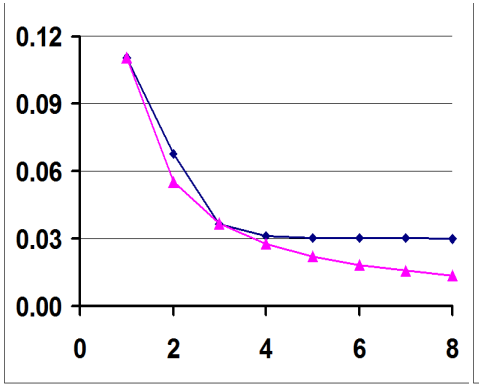
case	Data set	Number of lines	Number of computer nodes							
			1	2	3	4	5	6	7	8
1	solar plume	10,000	X	X	X	X	X	X	X	X
2	combustion	10,000	X	X	X	X	X	X	X	X
3	combustion	100,000	X	X	X	X	X	X	X	X
4	combustion	1,000,000				X	X	X	X	X

Table : Setup of experiments. Entries marked with “x” represent experiment runs.

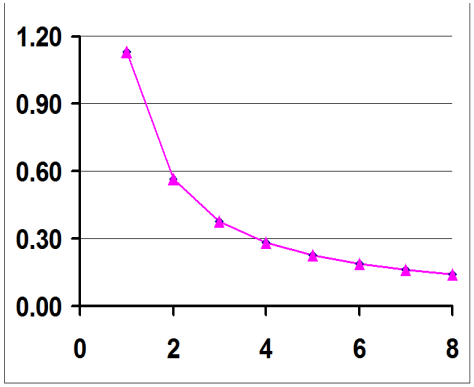
Clustering Performance Results



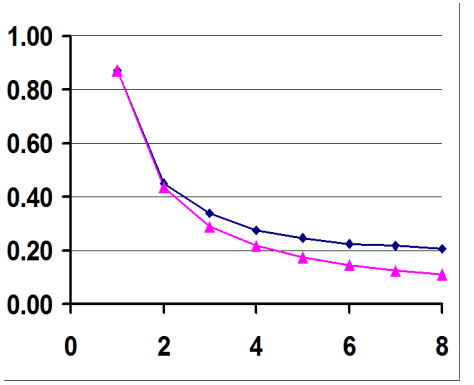
Case 1 smoothing time



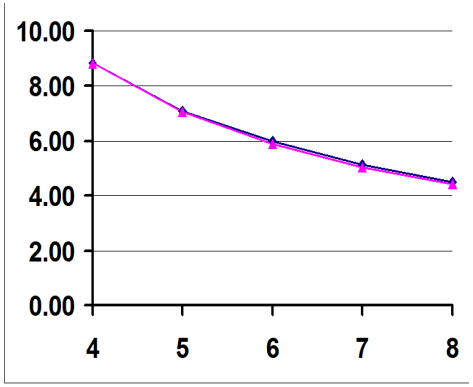
Case 1 resampling time



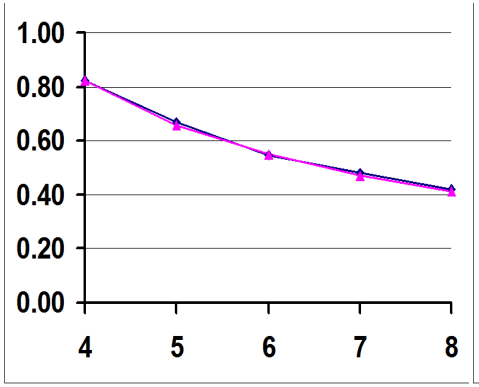
Case 1 E-Step time



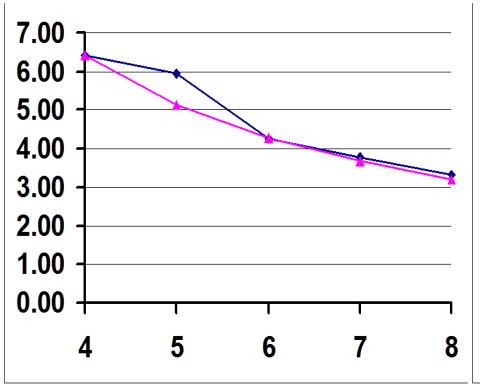
Case 1 M-Step time



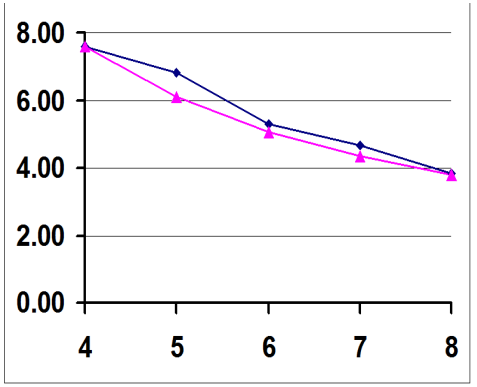
Case 4 smoothing time



Case 4 resampling time



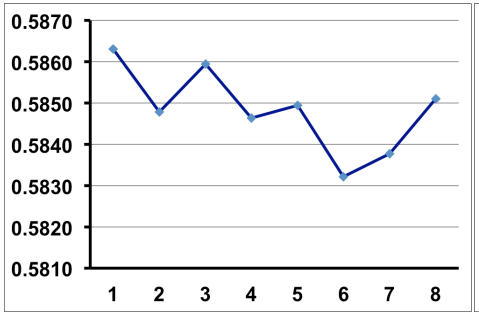
Case 4 E-Step time



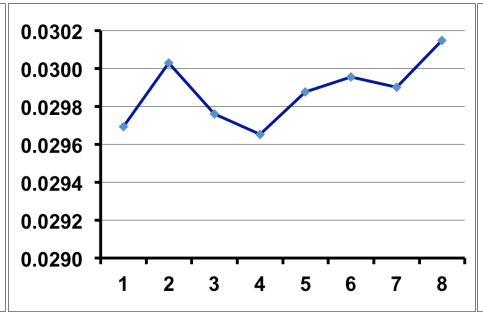
Case 4 M-Step time

Speedups of scalability study. In each plot, the horizontal axis: number of nodes; the vertical axis: running time in second; :real speed-up time : ideal speed-up time.

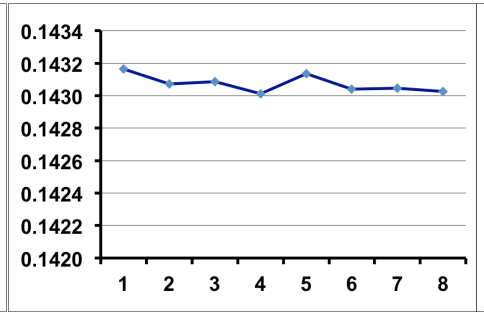
Clustering Performance Results



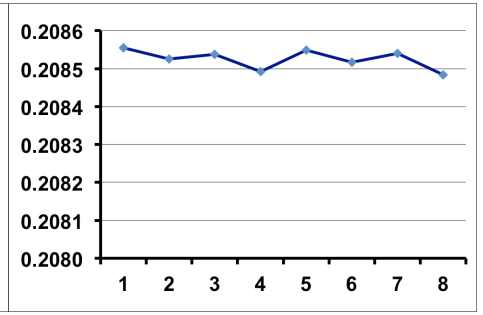
Case 1 smoothing time(0.53%)



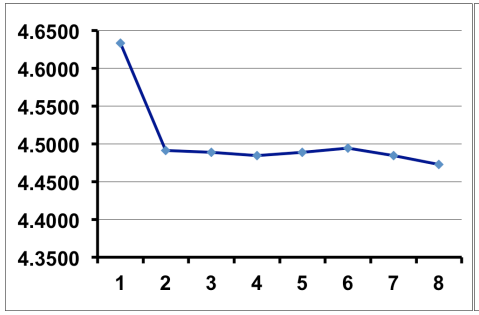
Case 1 resampling time(1.64%)



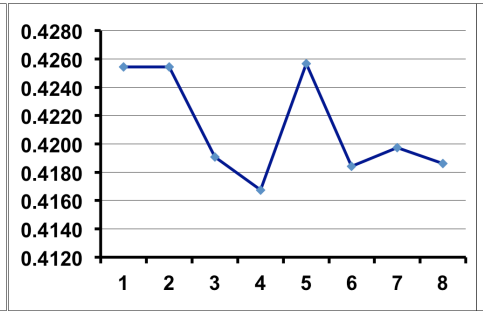
Case 1 E-Step time(0.11%)



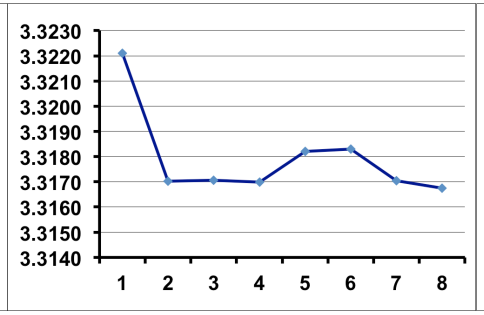
Case 1 M-Step time(0.03%)



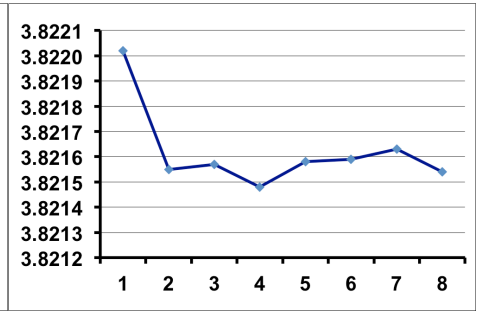
Case 4 smoothing time(3.46%)



Case 4 resampling time(2.09%)



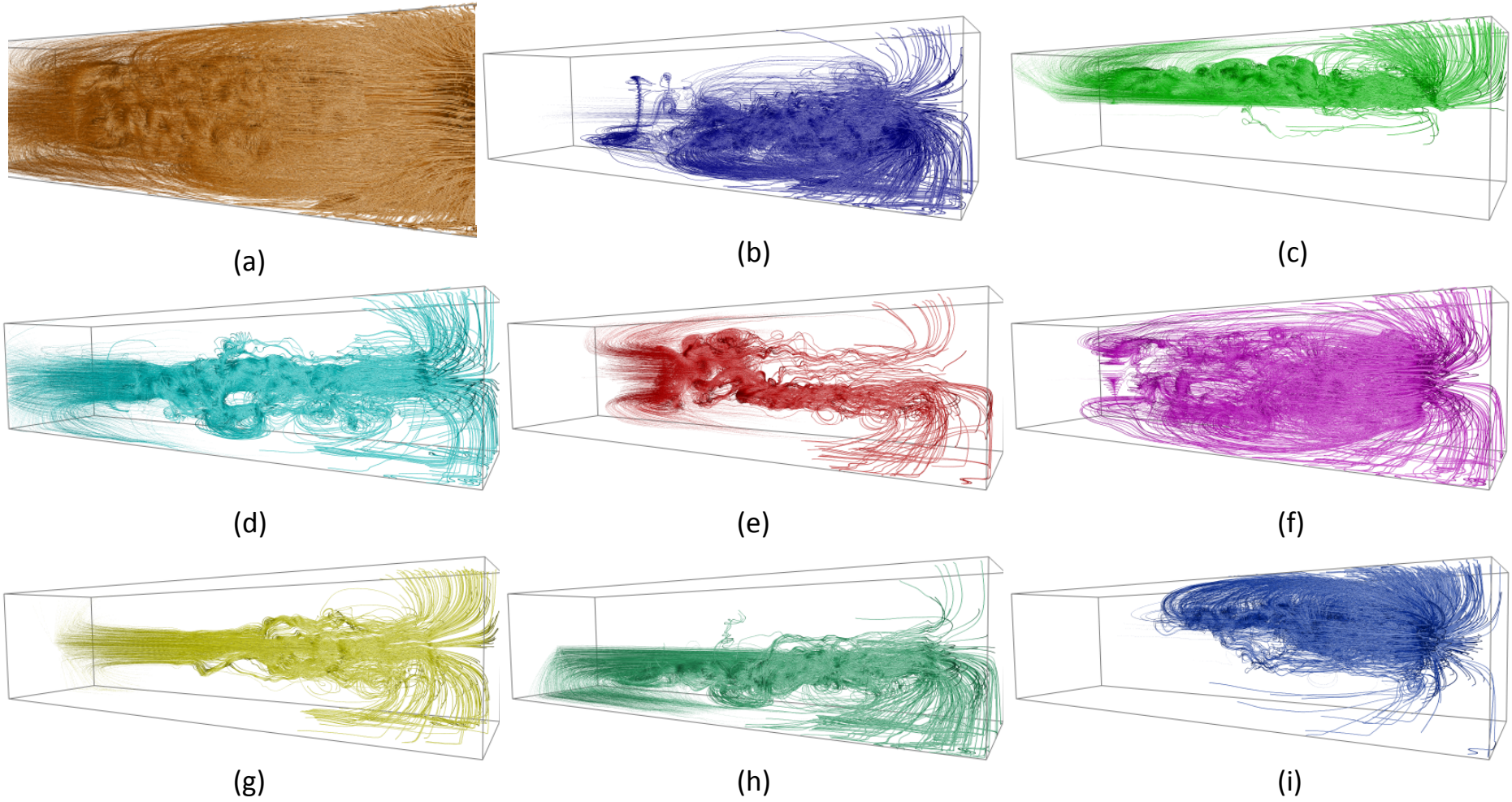
Case 4 E-Step time(0.16%)



Case 4 M-Step time(0.01%)

Workloads among 8 nodes for Cases 1 and 4. In each plot, the horizontal axis represents the node ID, and the vertical axis represents the running time in second. The percentage number associated with each plot is the difference ratio ($dr = (max\_time - min\_time)/max\_time$) between the maximum and minimum times among the nodes.

Visualization Results



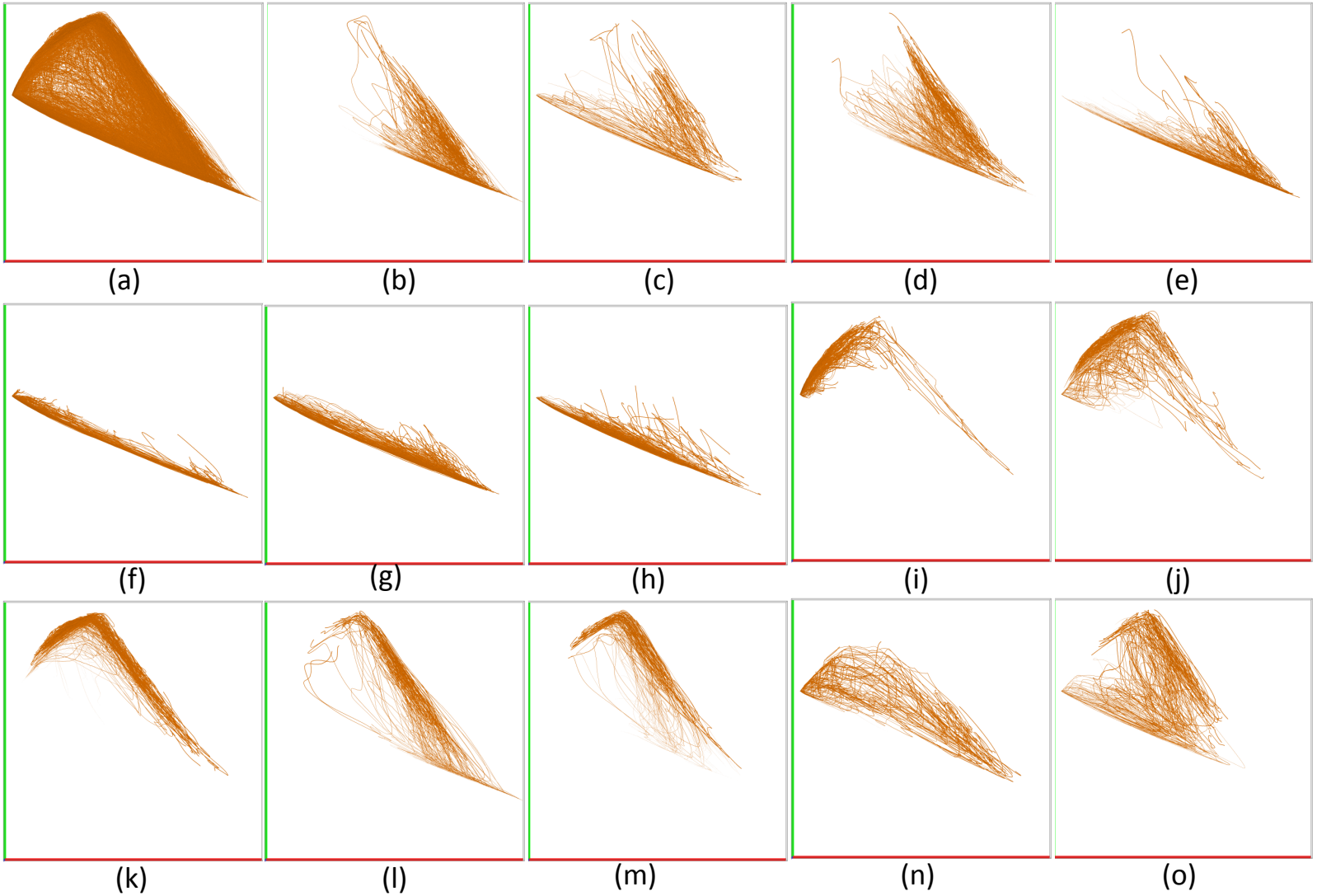
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Visualization of the streamlines generated from the solar plume velocity vector field. (a) shows the overview of all 10,000 streamlines. (b)-(i) show the eight different groups of streamlines.

Visualization Results



Visualization of the time series curves relating two variables, mixture fraction (the red axis) and temperature (the green axis), in the combustion simulation. (a) shows the overview of all 100,000 time series curves. (b)-(o) show the fourteen different groups of time series curves.

Conclusion and Future Work

- Our approach clusters large line data with multiple CPUs and GPUs
  - How to distribute the line data for balanced workload
  - How to effectively preprocess line data in CUDA
  - How to devise and implement the regression model-based clustering in MPI
- Future work:
  - Conduct clustering in situ and compress lines as much as possible
  - Visualize high dimensional lines

Acknowledgement

- This work has been sponsored in part by
  - the U.S. Department of Energy through the SciDAC program with Agreement No. DE-FC02-06ER25777 under Program Manager Dr. Lucy Nowell
  - the U.S. National Science Foundation through grants OCI-0749217, CCF-0811422, CCF-0850566, OCI-0749227, and OCI-0950008.

Questions or Comments?

Thank You!