Vector Field Data Summarization and Exploration

Han-Wei Shen[1], Teng-Yok Lee[1], Abon Chaudhuri[1], Kewei Lu[1], Tom Peterka[2]
[1]The Ohio State University
[2]Argonne National Laboratory

Fractal Dimensions

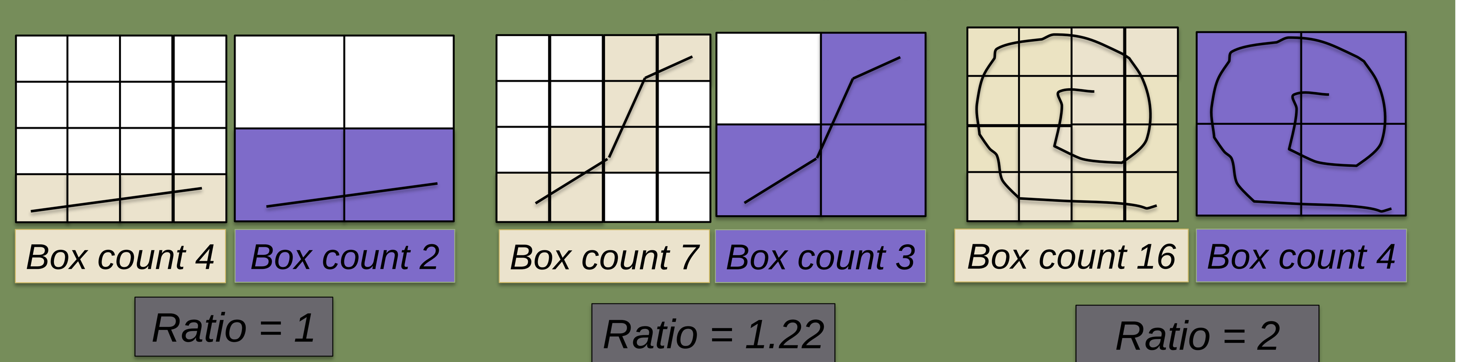
Streamline's Geometric Complexity

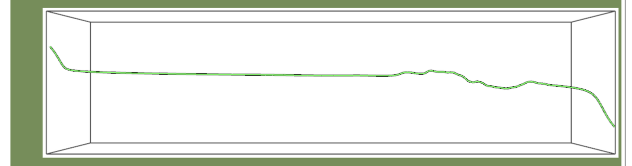
- ***Observation:*** Different flow features are represented by streamlines of different geometric complexity. A metric which quantifies geometric complexity can be useful
- ***Proposed solution:*** We adapt a fractal dimension based metric called box counting ratio [1] which can capture geometric complexity of streamlines

Fractal Dimension $= \lim_{\delta \to 0} \frac{\log(N_\delta(F))}{-\log(\delta)}$

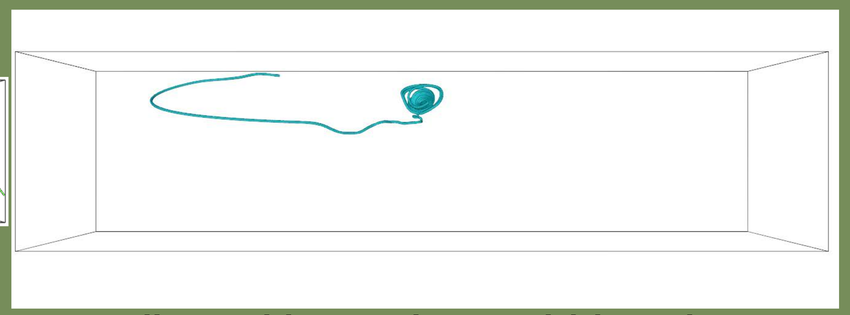
Where F is an object, N is the number of cells covered by F, δ is scale of measurement

Box Counting Ratio$=log_2 \frac{\textit{box count at finer resolution}}{\textit{box count at coarser resolution}}$



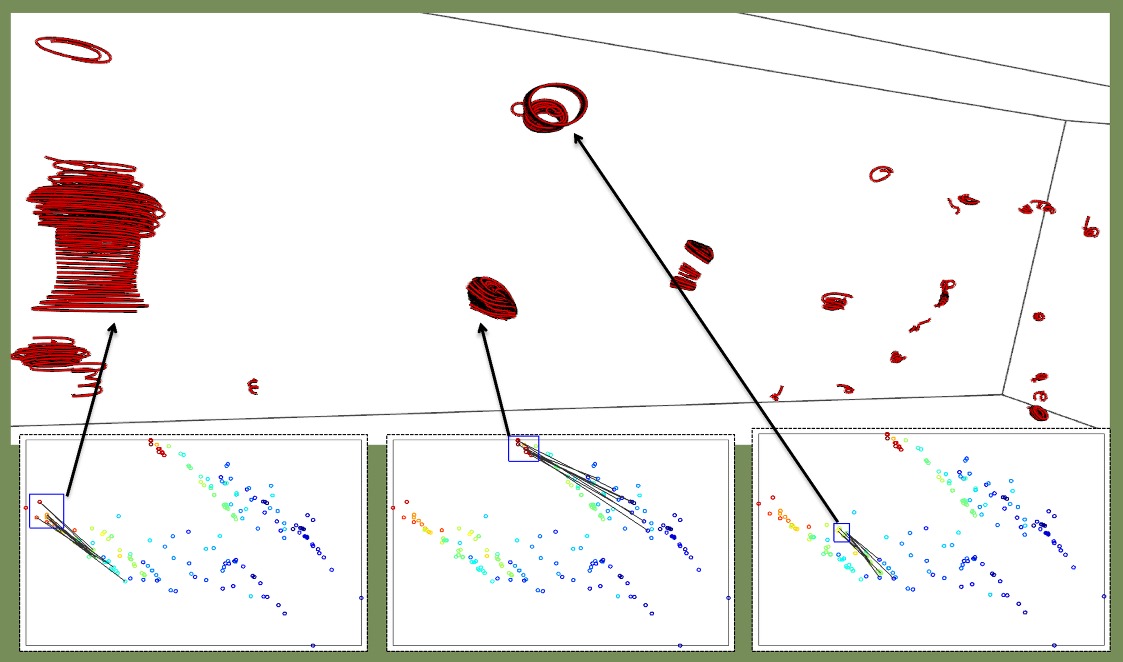


Fairly straight streamlines have box counting ratio close to 1



Streamlines with complex part(s) have box counting ratios well above 1. The complex parts alone may have box counting ratio above 2. Range is: 0-3.

Case Study: Solar Plume

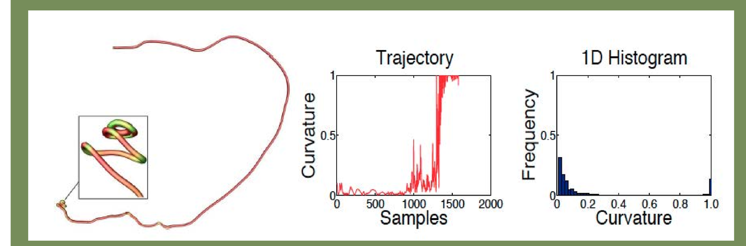


Solar Plume: Triage of complex segments extracted based on box counting ratio. Top: Extracted feature segments, Bottom: clusters on the feature space correspond to various clustered streamline segments for this dataset

Feature Histogram

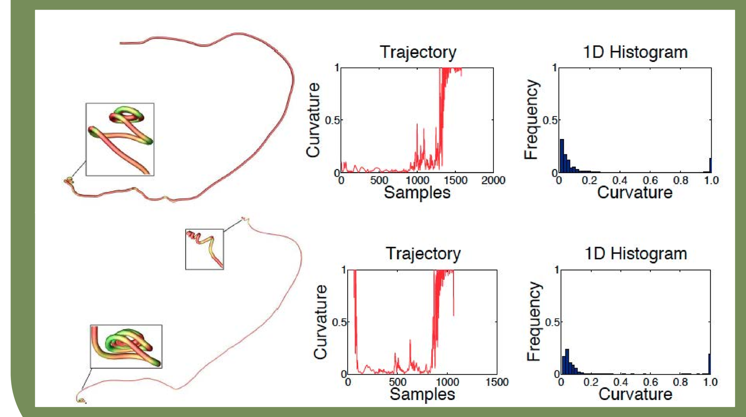
Histogram Construction

- Generate K streamlines within a local region (or the entire vector field)
- Uniformly sample the streamlines
- Construct a histogram of a desired attribute from the samples



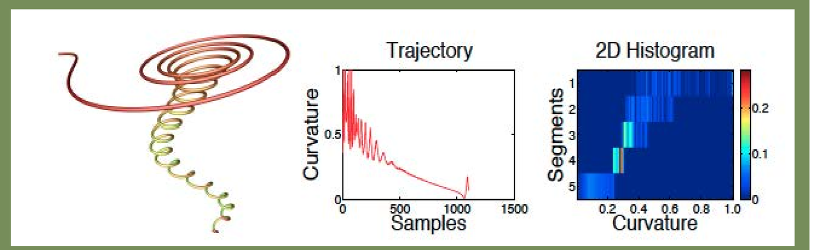
Why not 1D histogram

- The order of the attribute values may need to be preserved



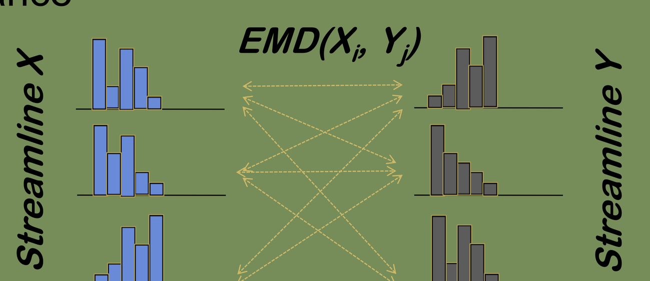
From 1D to 2D histograms

- Preserve the order of the attribute values by dividing streamlines into segments



Distance between histograms

- Calculate the distance between every pair of 1D histograms using Earth Mover's Distance
- Using Dynamic Time Warping to map streamline segments and get the final distance

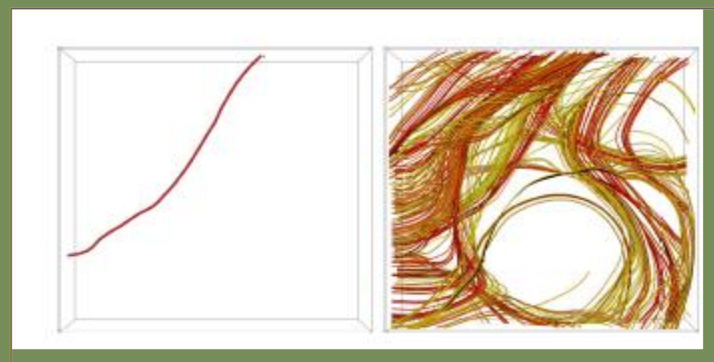


- ***DTW: minimum mapping cost***

$$Dist(H_1, H_2) = min_v(\sum_{l=0}^{L} d(x_{n_l}, y_{m_l}))$$

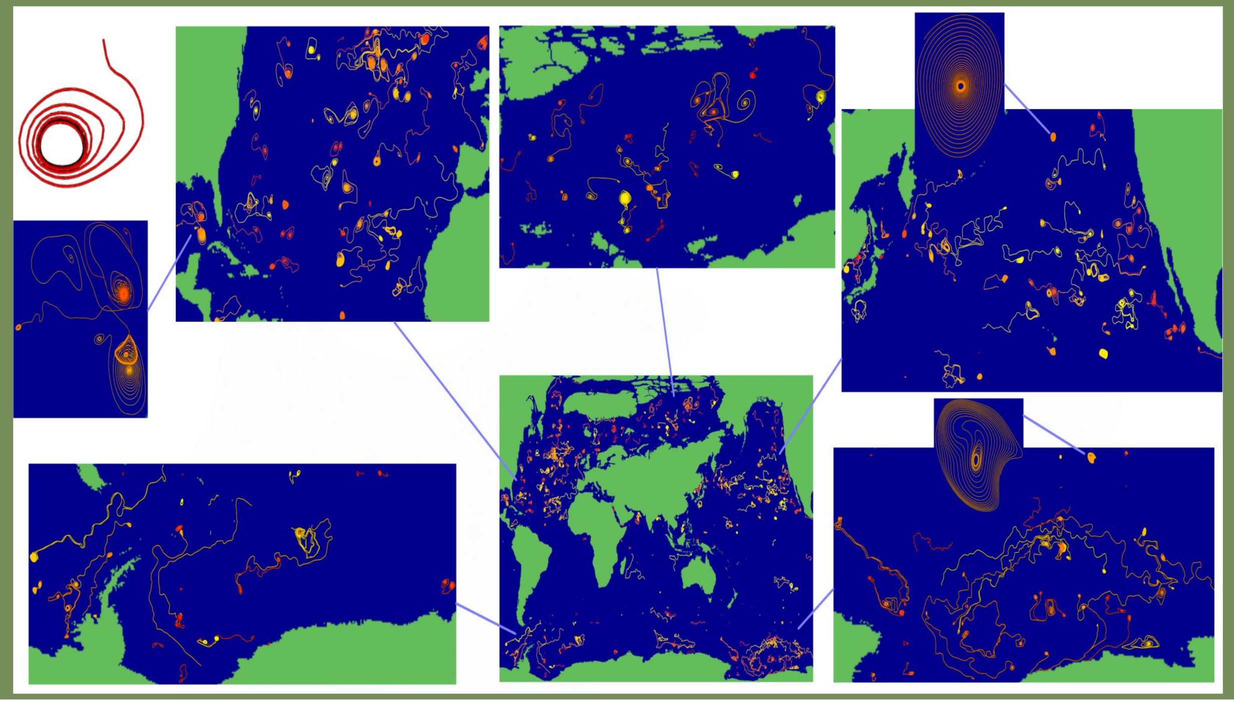
Application: Similar Streamline Query

- The user can specify some sample streamlines, and uses the distribution-based distance to find all similar streamlines



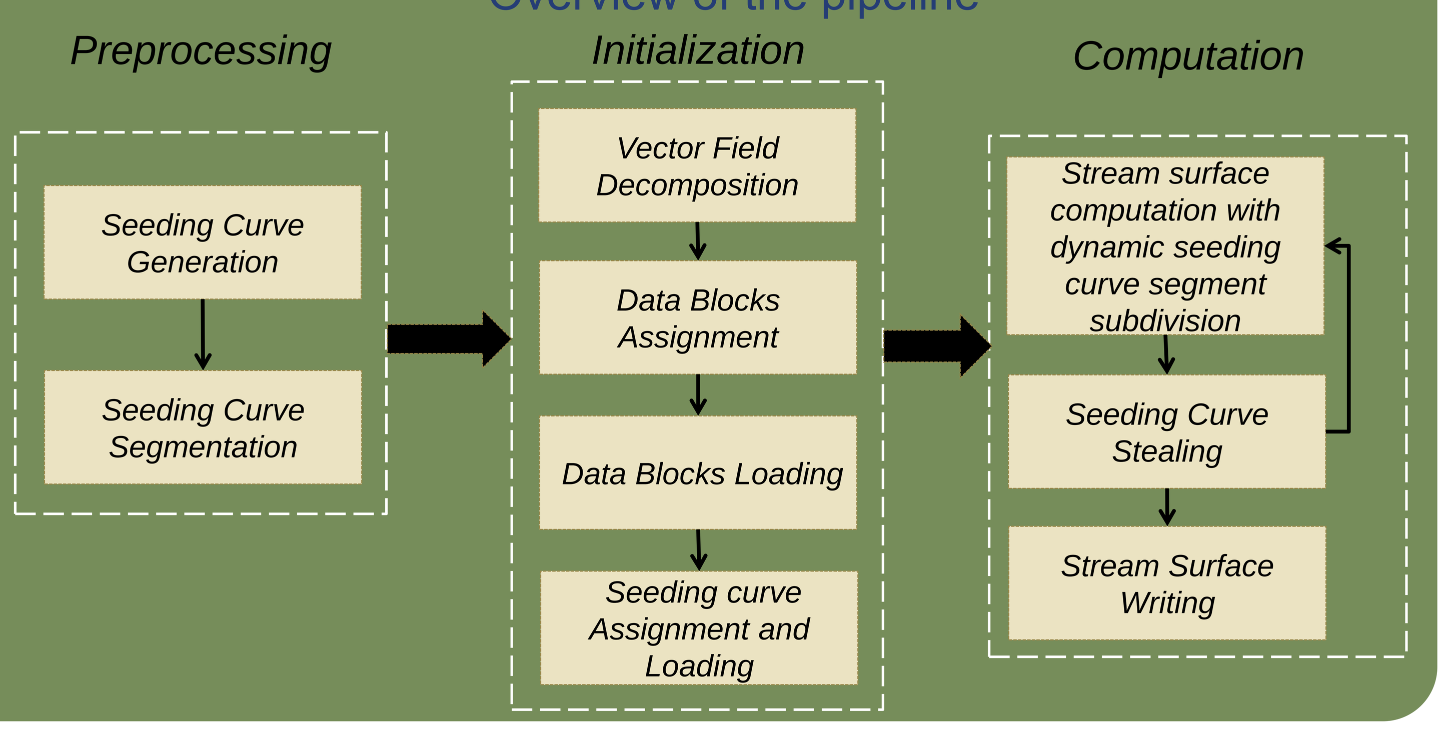
Sample Streamline ***Extracted Streamlines***

Sample Streamline ***Extracted Streamlines***



Parallel Stream Surface

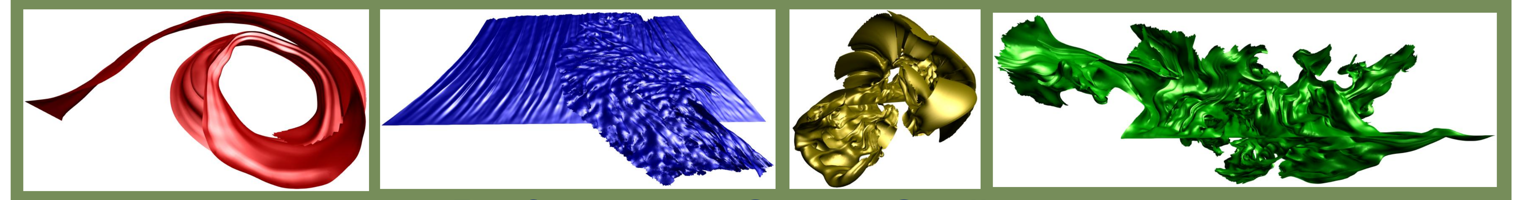
Overview of the pipeline



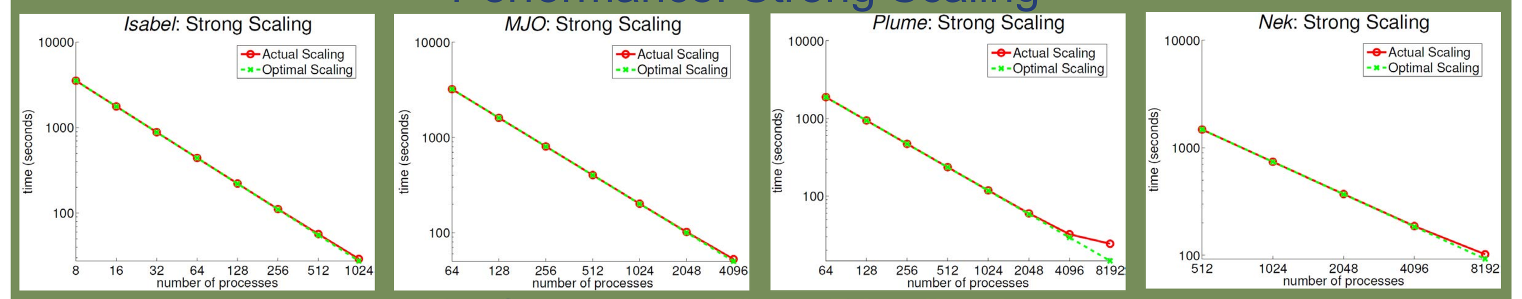
Parallelization Strategies

- *Cut seeding curves into segments, each process computes part of the stream surface*
- I/O strategies:
  - Data loading on demand from other processes over network instead of disk
  - Use cache to store the recently received data blocks
- Dynamic load balancing strategies:
  - Work stealing
  - Runtime seeding curve segment subdivision

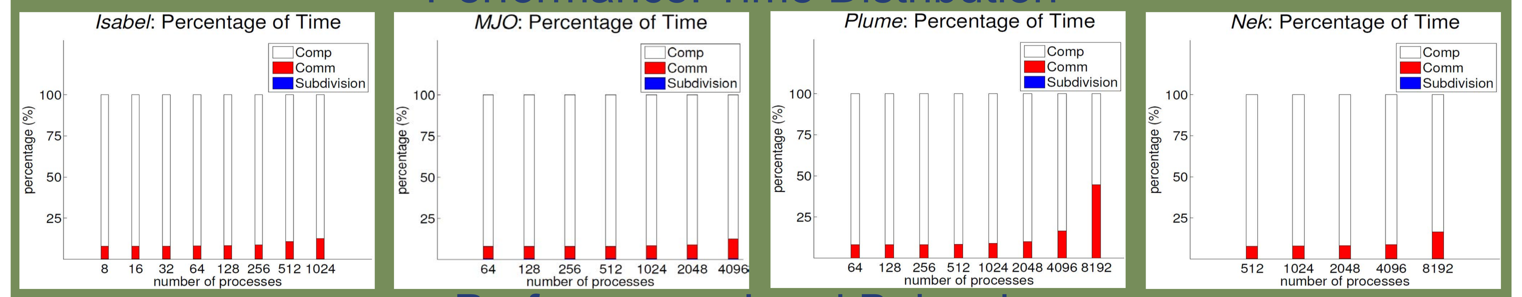
Data



Performance: Strong Scaling



Performance: Time Distribution



Performance: Load Balancing