Big Data Visual Analytics (CS 661)

Instructor: Soumya Dutta

Department of Computer Science and Engineering

Indian Institute of Technology Kanpur (IITK)

email: soumyad@cse.iitk.ac.in

Two Make Up Classes

- Due to an official travel, I won't be able to take in person classes on April 21st and April 23rd (the last two lectures)
- So, we will have two make up classes to cover those two classes
- Make up class on April 5th (Saturday) 2:00-3:15pm at RM-101
- Make up class on April 12th (Saturday) 2:00-3:15pm at RM-101
- The quiz syllabus will remain the same, i.e., up to the class on April 2nd
- We will have final project evaluation during the final exam week
  - Possibly between 27-30th

Quiz on April 7th Monday at LH-20, 2-3pm

- Quiz will have both subjective and MCQ type questions
- Quiz duration: 1 Hr. (2:00pm – 3:00pm) @ LHC-20
- Please come on time
- Bring your IITK ID
- Syllabus: Everything starting from Lecture 13 (Status Refresher) up to topics discussed until April 2nd

Final Class Project: What is Expected?

- You will build a visual-analytics interface/software to solve a problem from an application domain
- You are expected to have knowledge about your domain of application and the tasks you have picked must reflect that
- Your tasks should be meaningful, and you should be able to explain why you are doing something
- You should be able to tell a coherent story about your data through your visualization interface
  - Random set of plots will not work
  - You need to justify why you picked certain type of plots over others, i.e., design aspects of the tool

Final Class Project: What is Expected?

- You are expected to show meaningful patterns from your data through your interface
- Since this is a visualization focused course, so even if you do a very sophisticated modeling, your visualization interface still will be the focus for grading
  - How you are presenting your results will matter
- You are expected to write a comprehensive report of your work that describes the details of the methodologies and visualization techniques that you have used
  - You should justify your design choices for your interface
    - Such as: Why a bar chart and not a Pie chart!

Final Class Project: How Will It be Graded?

- Grading will be done on your overall idea, quality of work, final presentation, and report
- Grading will be done for individual group members
- Total marks: 300
  - Overall quality: 100
  - Presentation: 100
  - Report: 100
- Your group will have to give a presentation + individual Q&A
- Schedule of presentations will be shared soon
- Guidelines for submitting code and report will be shared soon

Study Materials

- Multimodal Data Fusion Based on Mutual Information, Bramon et al., IEEE TVCG.
- In Situ Adaptive Spatio-Temporal Data Summarization, Dutta et al., IEEE BigData.

Multimodal Spatial Data Fusion

Specific Mutual Information Measures

- $Surprise = I_1(x;Y) = \sum_{y \in Y} p(y|x) log \frac{p(y|x)}{p(y)}$
- $Predictability = I_2(x;Y) - \sum_{y \in Y} p(y) log p(y) + \sum_{y \in Y} p(y|x) log p(y|x)$
- $Entanglement = I_3(x;Y) = \sum_{y \in Y} p(y|x) I_2(y;X)$

I_1, I_2, and I_3 Fusion

- I_1 Fusion:

$$z = \begin{cases} x, & \text{if } I_1(x;Y) > I_1(y;X) \\ y, & \text{otherwise.} \end{cases}$$

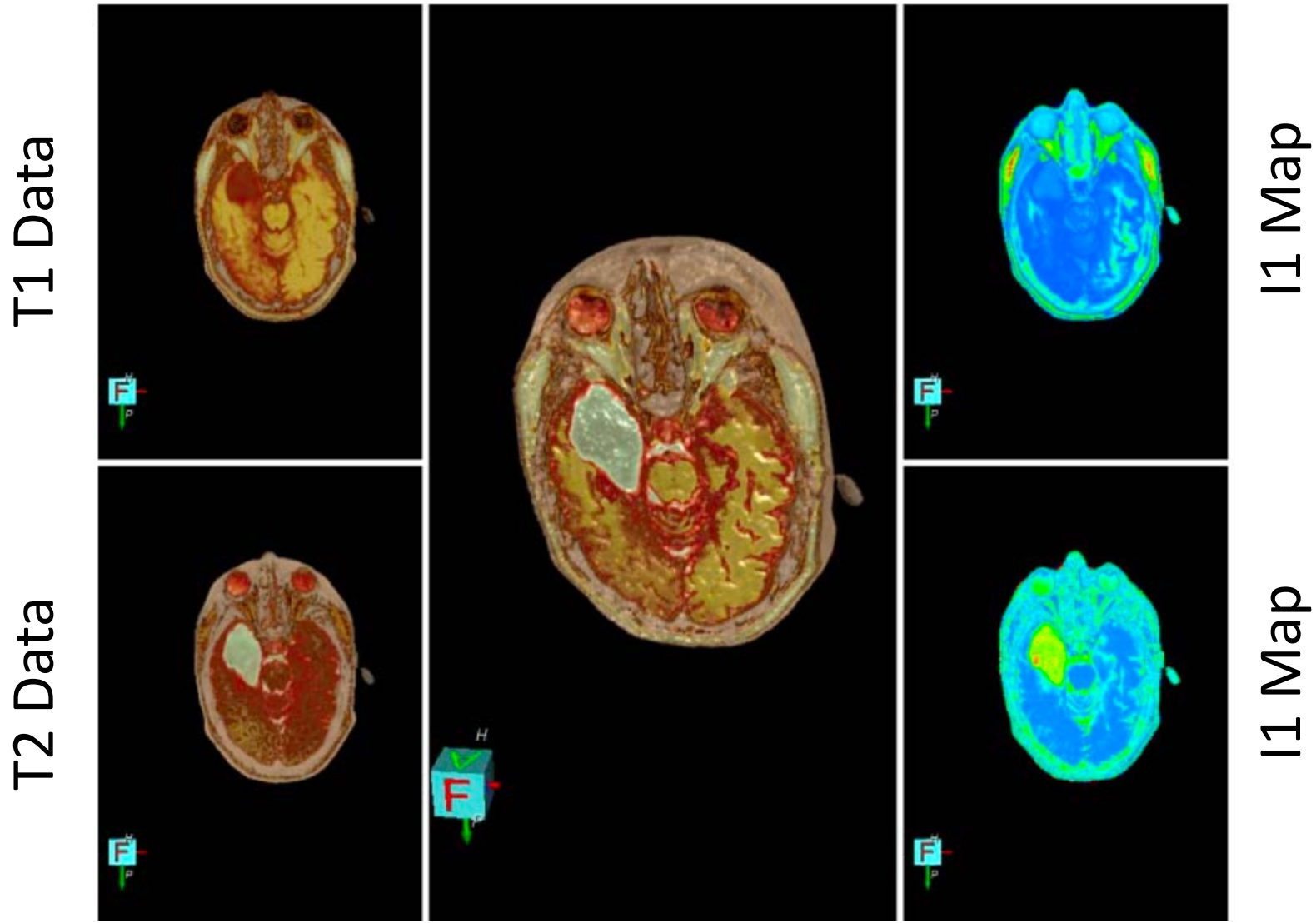
- I_2 Fusion:

$$z = \begin{cases} x, & \text{if } I_2(x;Y) > I_2(y;X) \\ y, & \text{otherwise.} \end{cases}$$

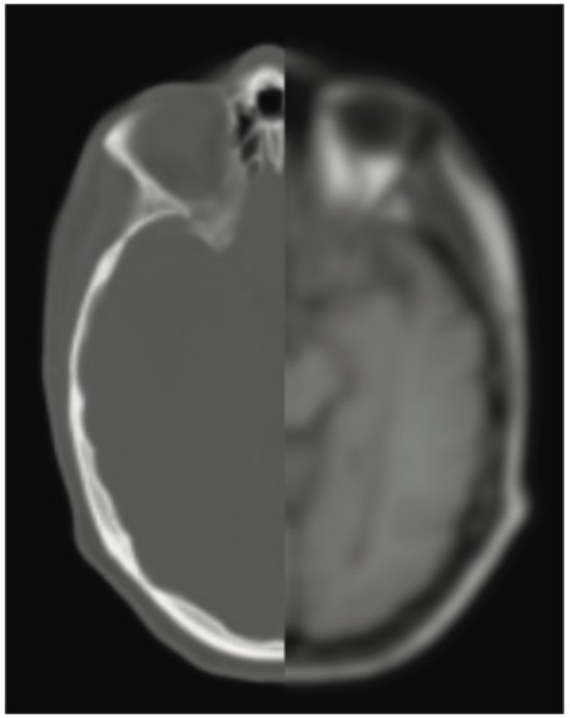
- I_3 Fusion:

$$z = \begin{cases} x, & \text{if } I_3(x;Y) < I_3(y;X) \\ y, & \text{otherwise.} \end{cases}$$

I1, I2, and I3 Fusion



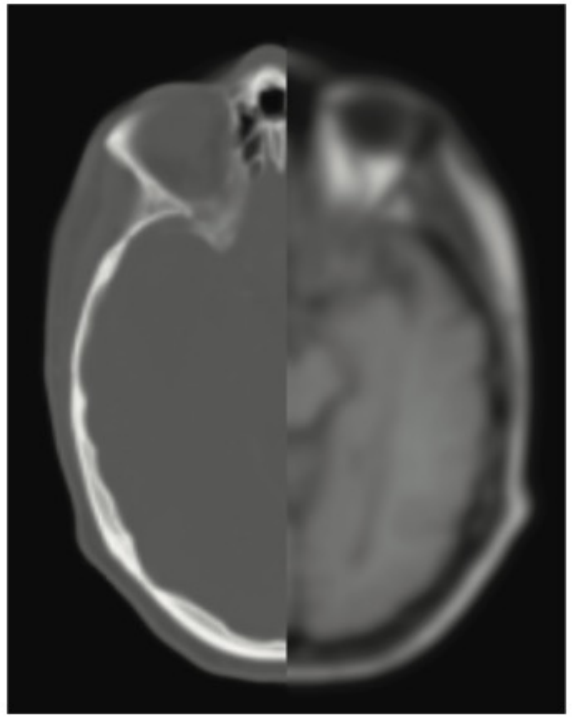
I1, I2, and I3 Fusion



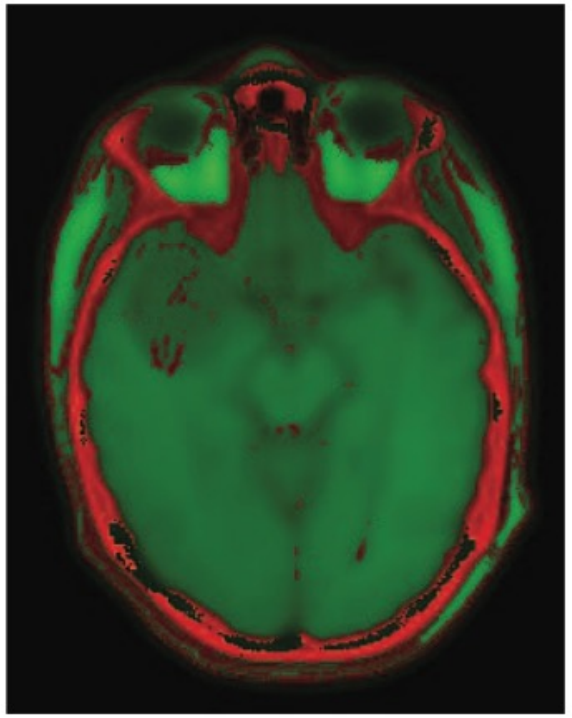
CT-T1 Data

Fusion after smoothing is applied

I1, I2, and I3 Fusion



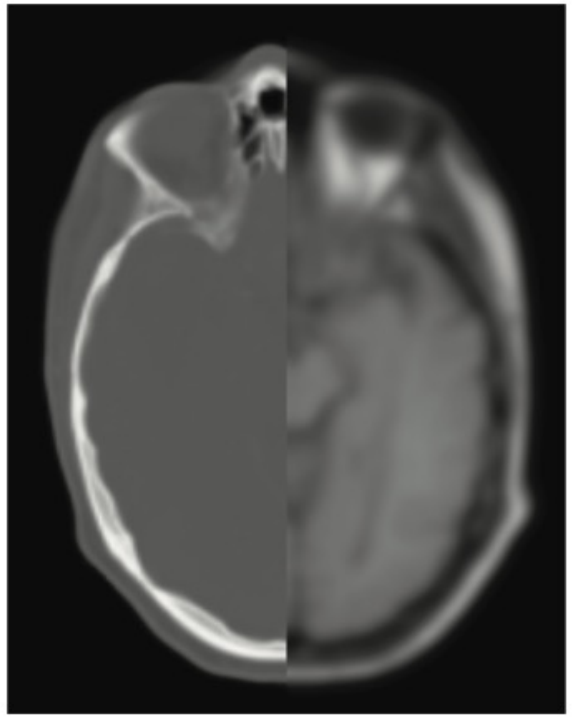
CT-T1 Data



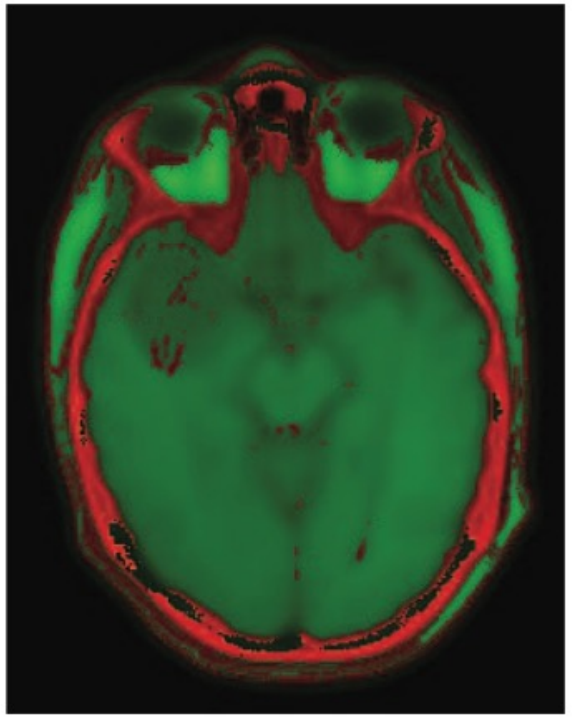
I1 Fusion

Fusion after smoothing is applied

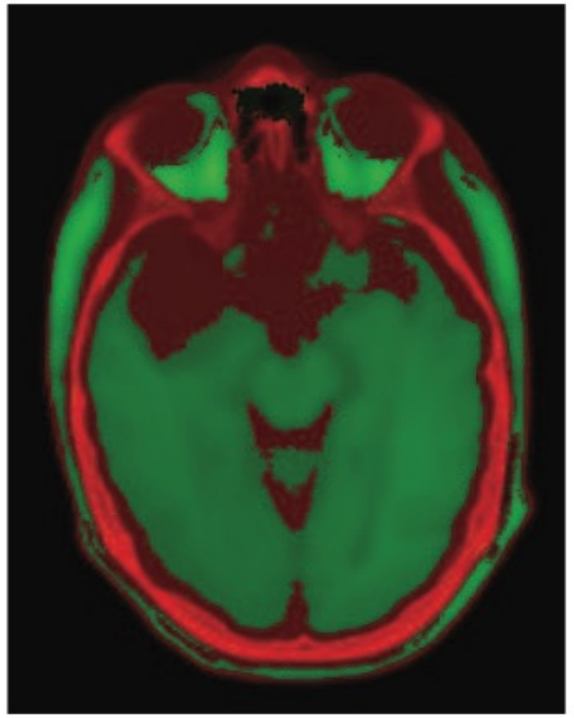
I1, I2, and I3 Fusion



CT-T1 Data



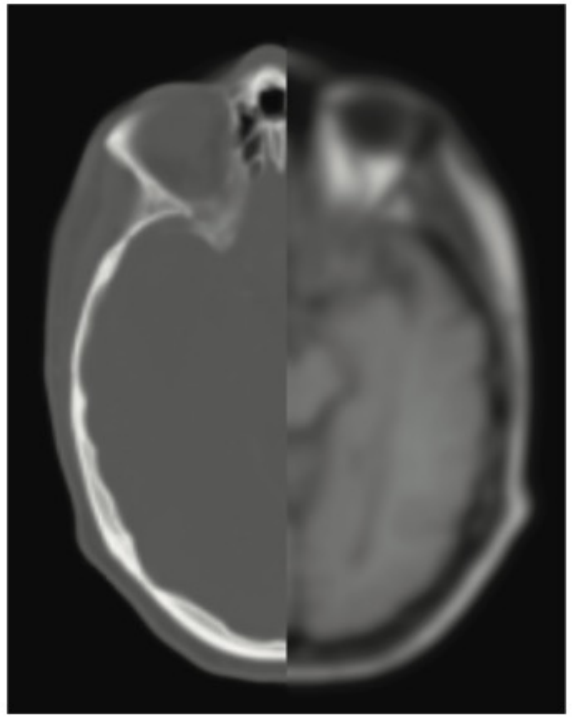
I_1 Fusion



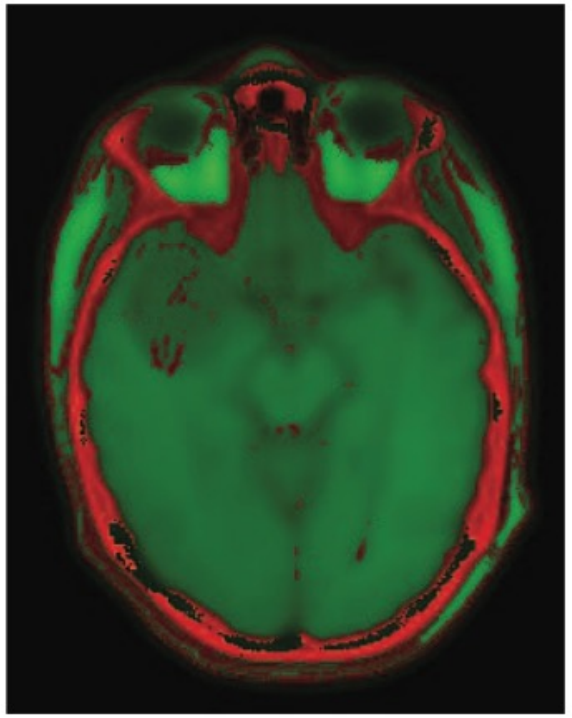
I_2 Fusion

Fusion after smoothing is applied

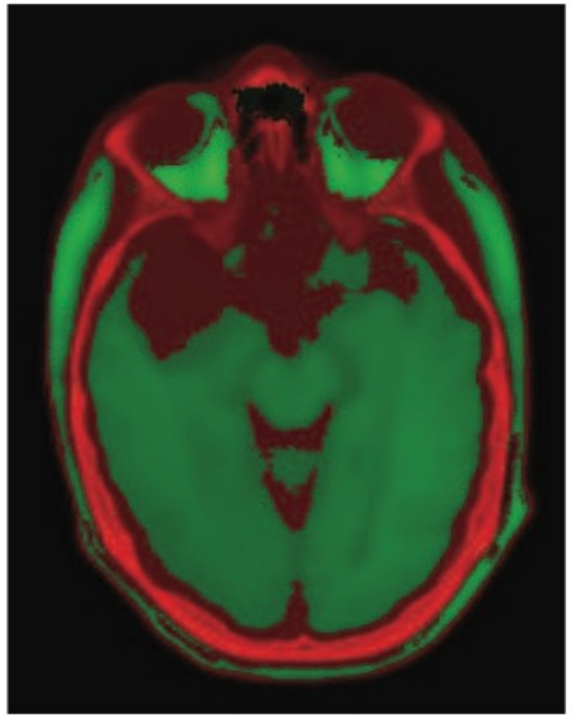
I1, I2, and I3 Fusion



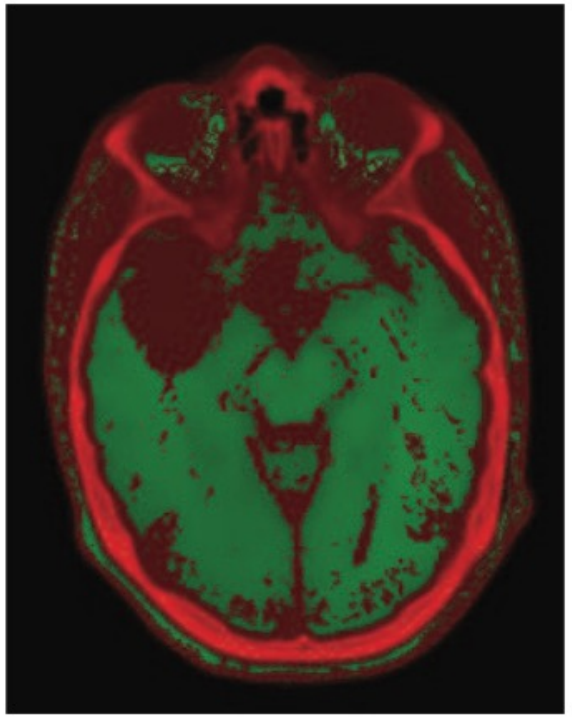
CT-T1 Data



I_1 Fusion



I_2 Fusion



I_3 Fusion

Fusion after smoothing is applied

Temporal Data Fusion

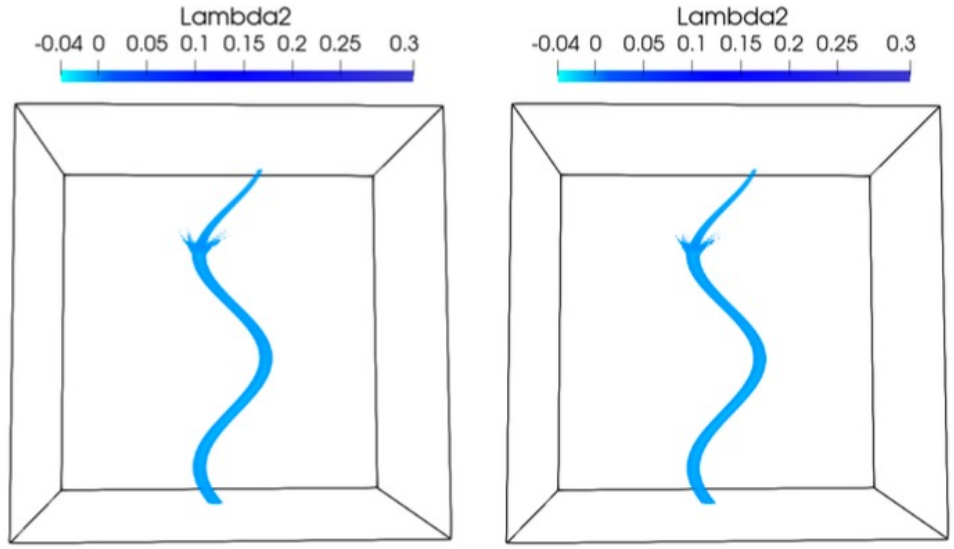
Temporal Data Fusion using Surprise (I_1)

- An adaptive Spatiotemporal data fusion using information theoretic measure 'Surprise (I_1)'

$$Surprise = I_1(x;Y) = \sum_{y \in Y} p(y|x) log \frac{p(y|x)}{p(y)}$$

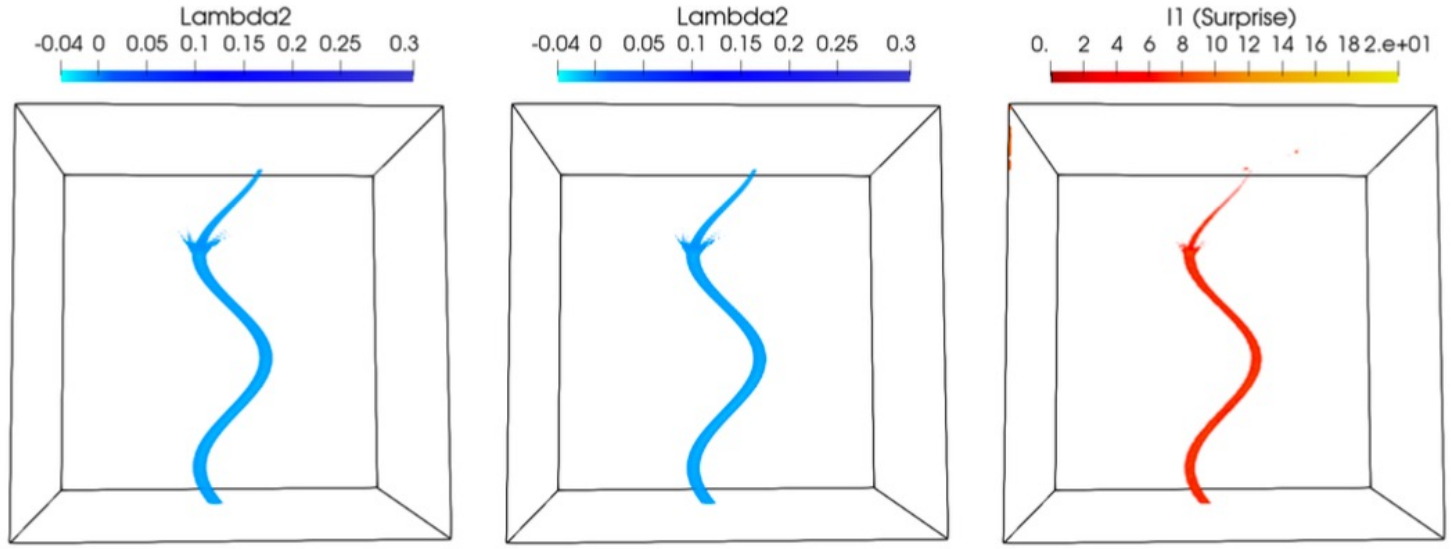
- Given a time window, identify which time step is the most informative for each location in 3D domain -- How?
  - Pick the data from the time step when the value of Surprise is maximum within a time window
- Generate a new fused 3D field where different location has values from different time steps

Temporal Data Fusion using Surprise (I_1)



(a) T=25 (b) T=26

Temporal Data Fusion using Surprise (I_1)

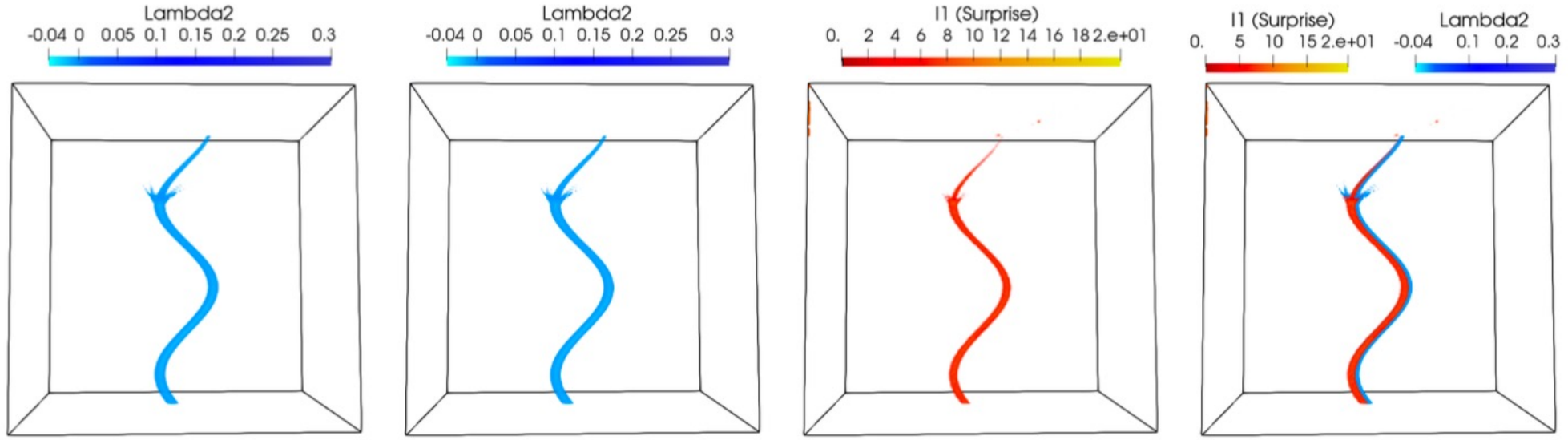


(a) T=25

(b) T=26

(c) $I1field$ generated using tornado data at T=25 and 26.

Temporal Data Fusion using Surprise (I_1)



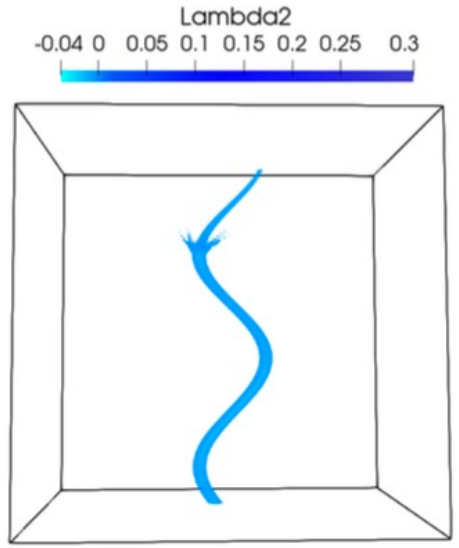
(a) T=25

(b) T=26

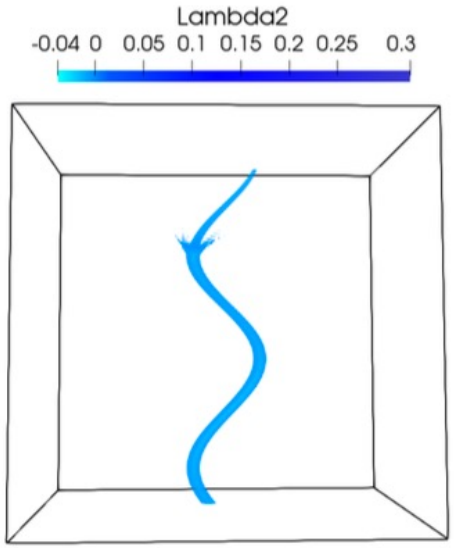
(c) $I1field$ generated using tornado data at T=25 and 26.

(d) $I1field$ overlapped with tornado data at T=25

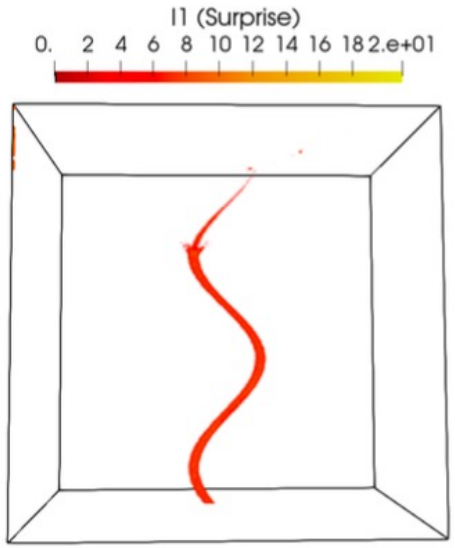
Temporal Data Fusion using Surprise (I_1)



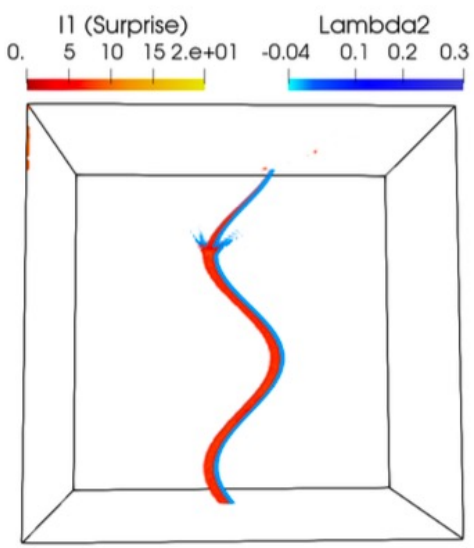
(a) T=25



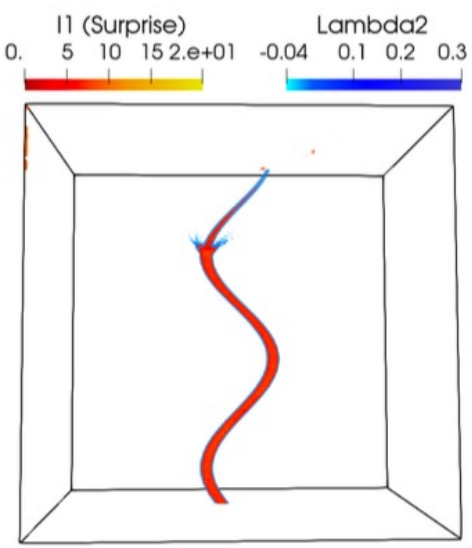
(b) T=26



(c) $I1field$ generated using tornado data at T=25 and 26.



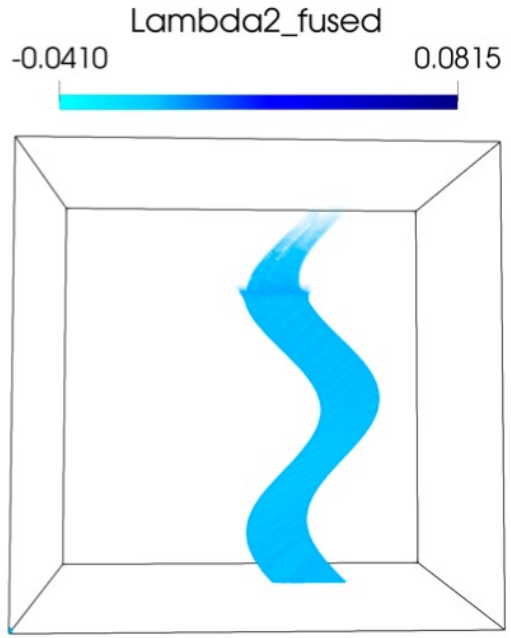
(d) $I1field$ overlapped with tornado data at T=25



(e) $I1field$ overlapped with tornado data at T=26

Time-varying Feature-based Data Fusion using Information Fields

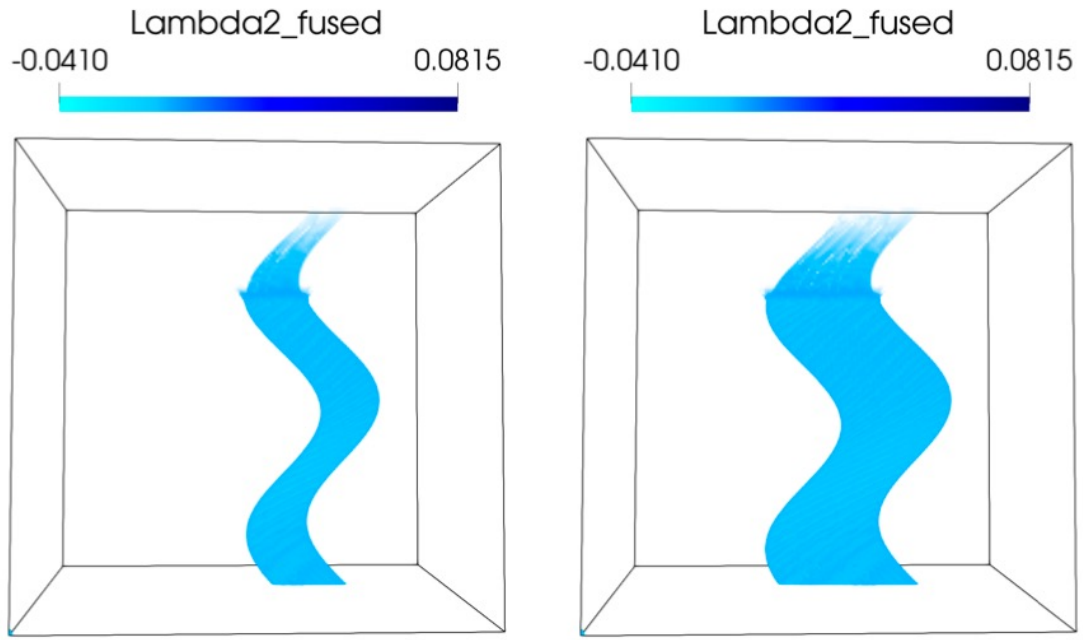
$$Val(p) = max(I_1^t(p)), \forall\, t = t_{start}, .., t_{end}$$



(a) TDSF for T=1-15

Time-varying Feature-based Data Fusion using Information Fields

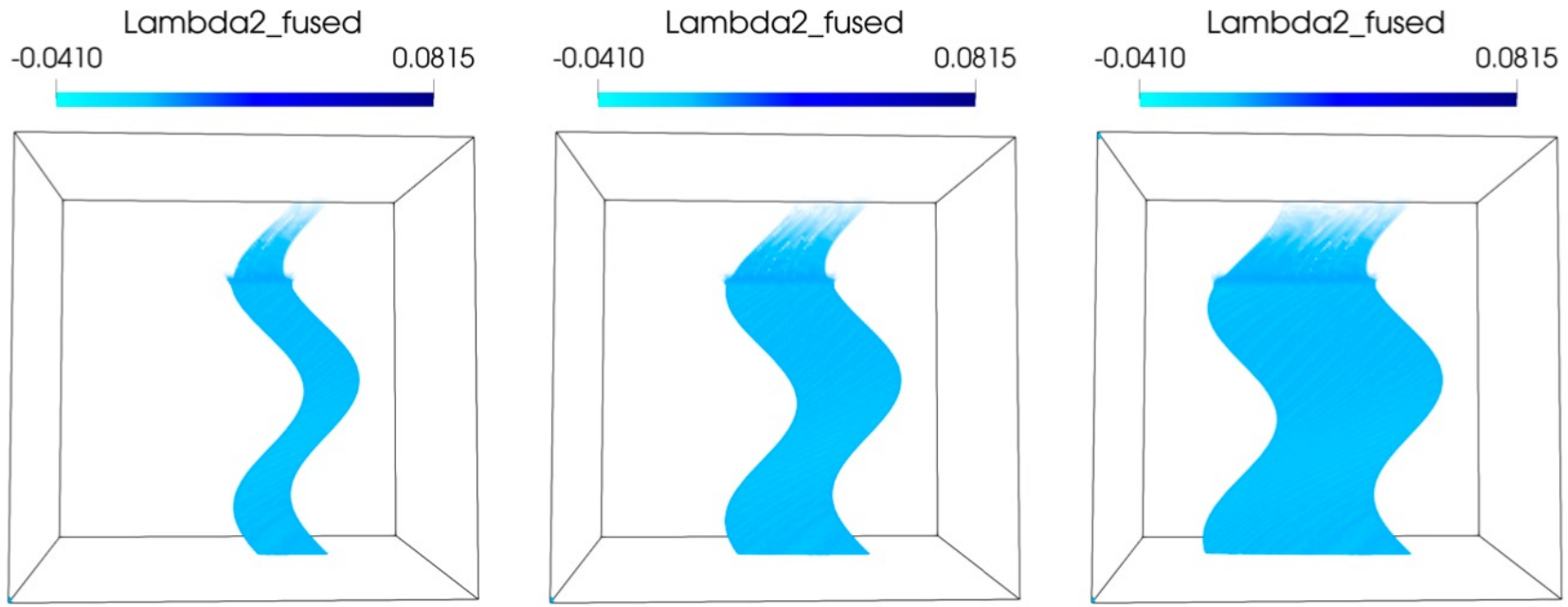
$$Val(p) = max(I_1^t(p)), \forall\, t = t_{start}, .., t_{end}$$



(a) TDSF for T=1-15 (b) TDSF for T=1-30

Time-varying Feature-based Data Fusion using Information Fields

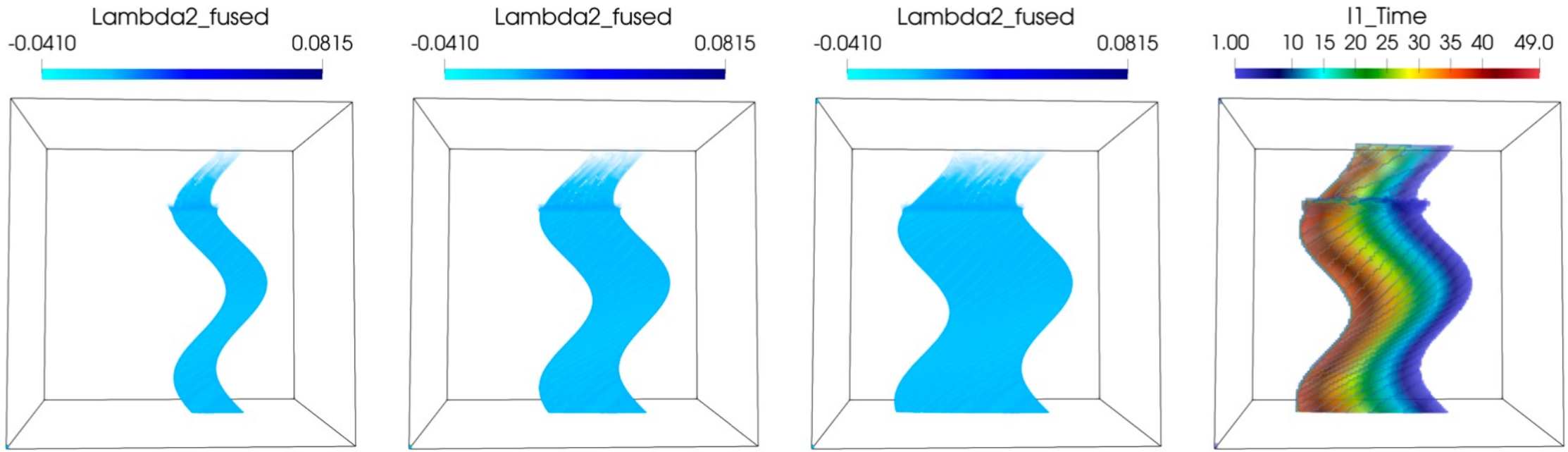
$$Val(p) = max(I_1^t(p)), \forall\, t = t_{start}, .., t_{end}$$



(a) TDSF for T=1-15 (b) TDSF for T=1-30 (c) TDSF for T=1-50

Time-varying Feature-based Data Fusion using Information Fields

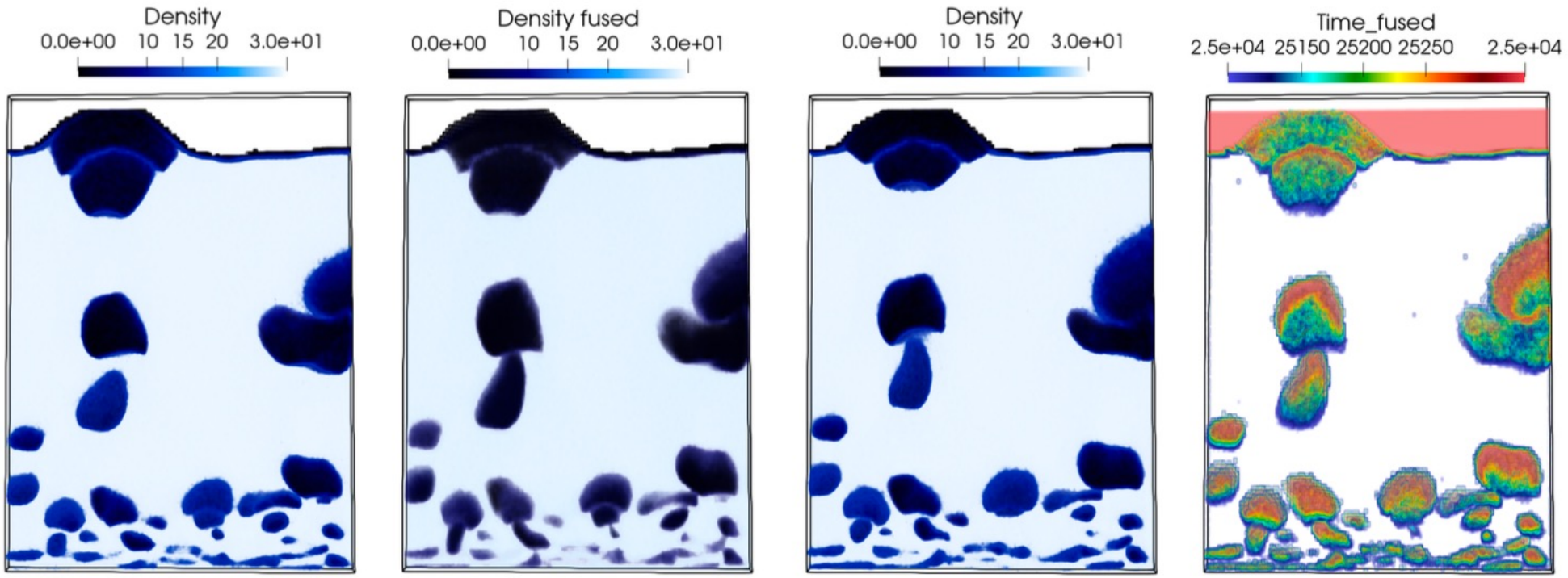
$$Val(p) = max(I_1^t(p)), \forall\, t = t_{start}, .., t_{end}$$



(a) TDSF for T=1-15 (b) TDSF for T=1-30 (c) TDSF for T=1-50 (d) TSSF for T=1-50

Time-varying Feature-based Data Fusion using Information Fields

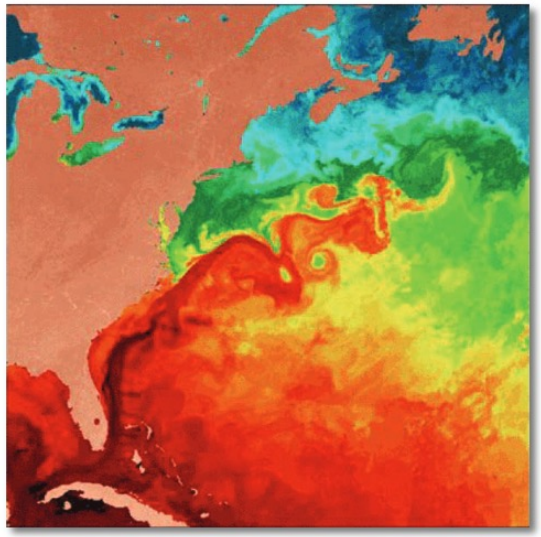
$$Val(p) = max(I_1^t(p)), \forall\, t = t_{start}, .., t_{end}$$



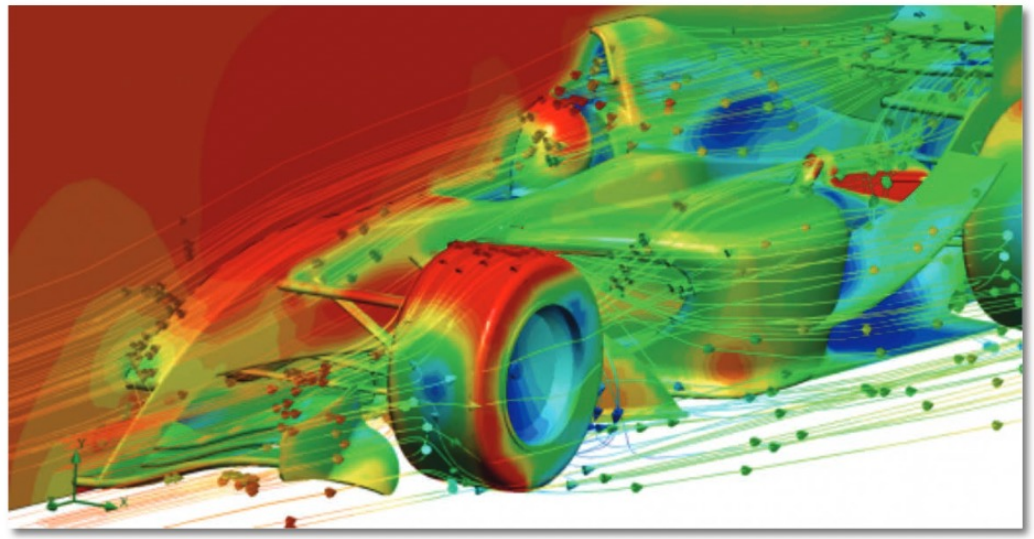
(a) Density field at T=25090. (b) TDSF for T=25090-25340. (c) Density field at T=25340. (d) TSSF for T=25090-25340.

Flow (Vector) Data Analysis and Visualization

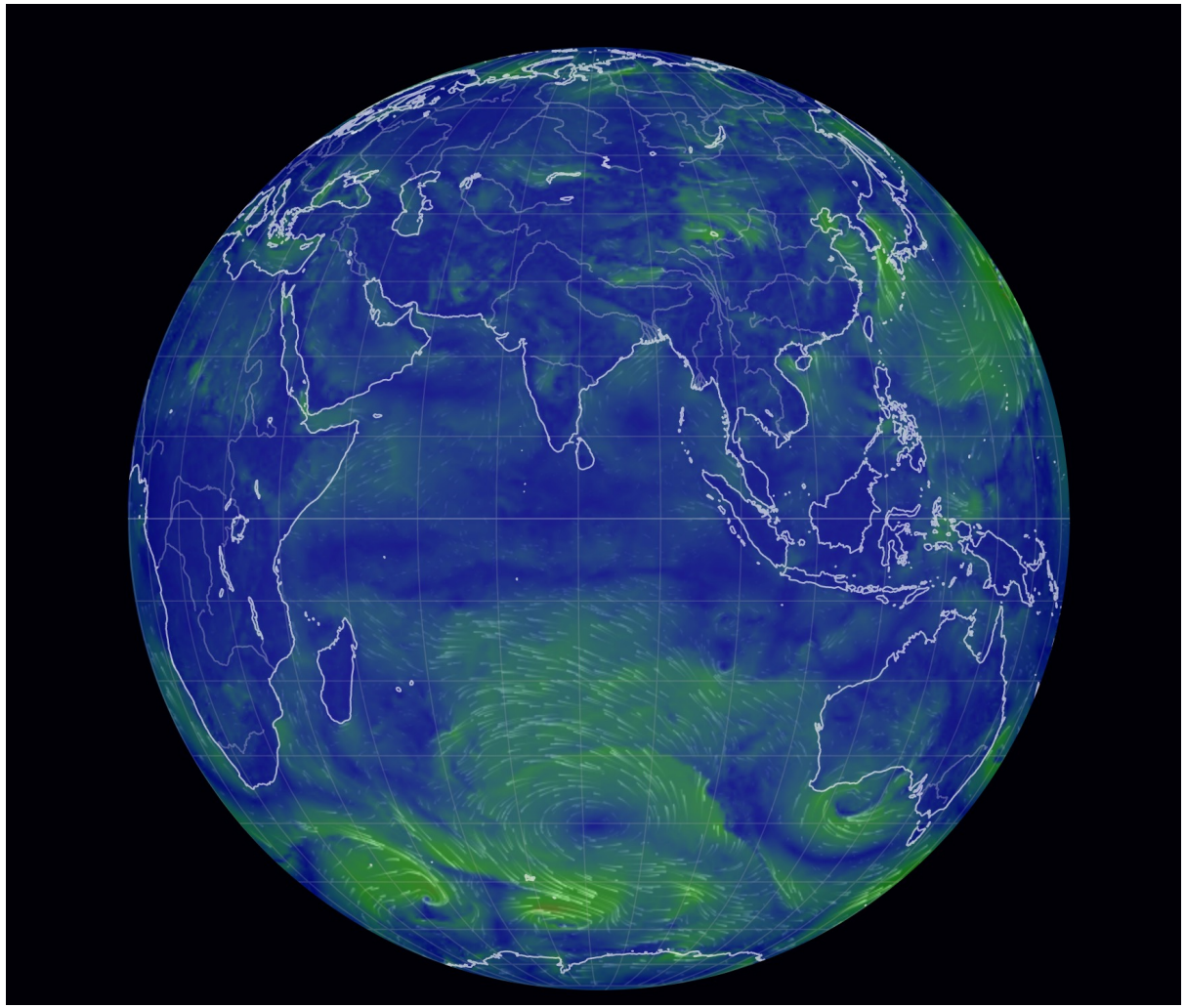
Flow (Vector) Fields







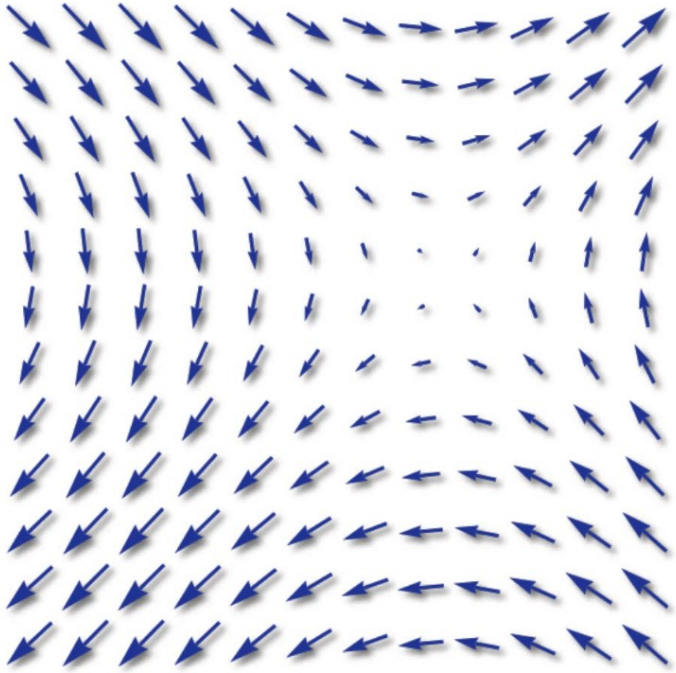
Real Life Example of Flow Visualization

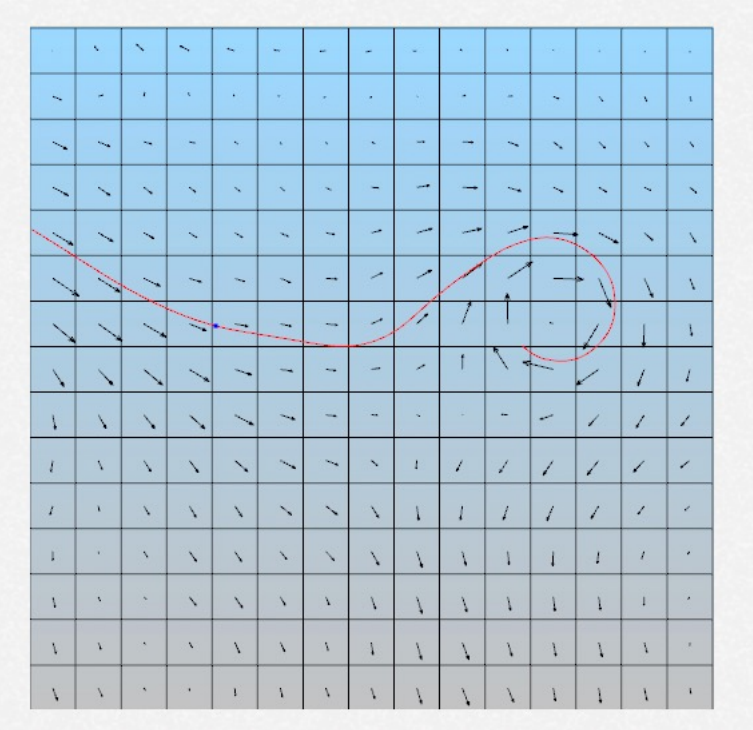


Demo: https://earth.nullschool.net/

Flow (Vector) Fields

- A vector field F(U) = V
  - U: Field domain (x,y) in 2D; (x,y,z) in 3D
  - V: vector (u,v) in 2D; (u,v,w) in 3D
- Like scalar fields, vector data are defined at grid points in a mesh





Flow Visualization

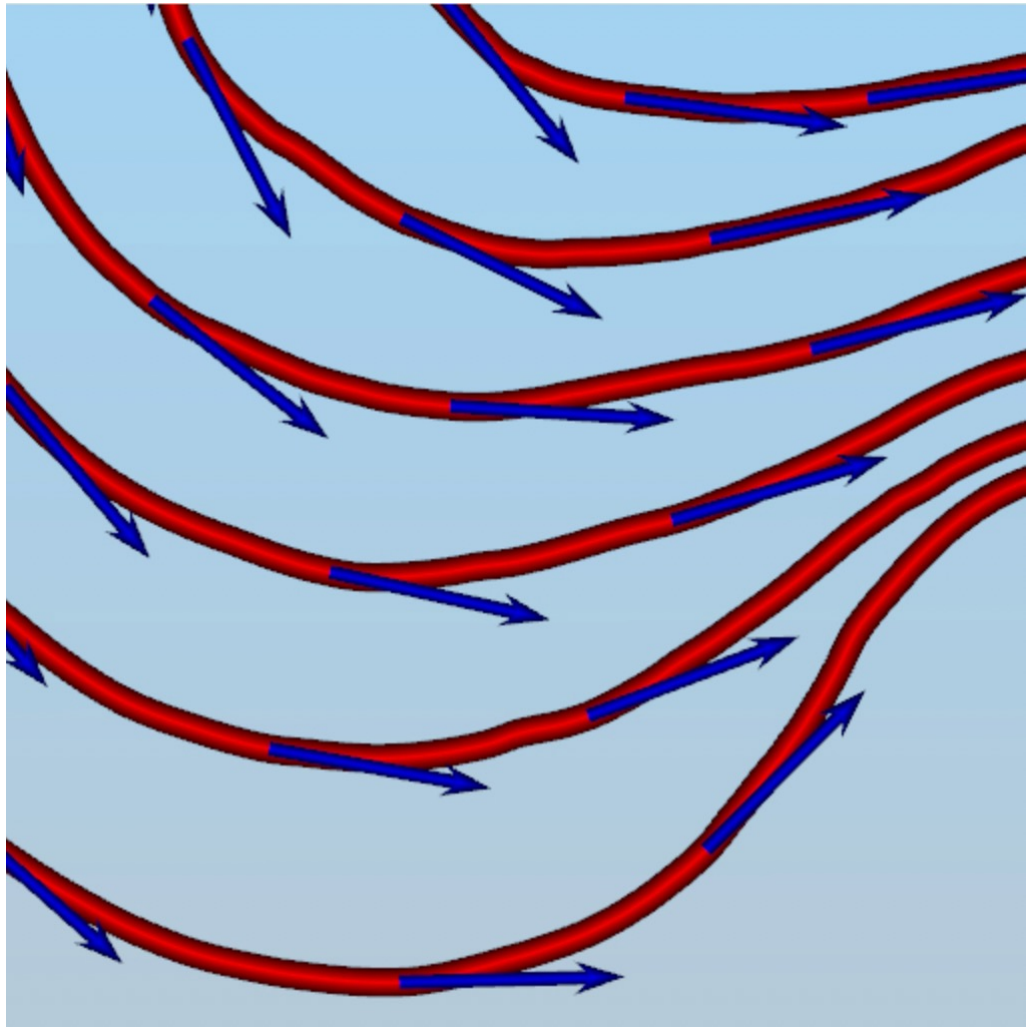
- A challenging task since flow patterns are often invisible
  - Air, water flow patterns
- Flow visualization is used to make flow patterns visible so that we can visually acquire qualitative and quantitative flow information
- In a ***vector field*** (or ***flow field***), vectors are defined over a discrete grid
  - Liner interpolation can be used to obtain vector at any location in the domain
- Steady flow field: Fluid properties and vectors at a point in the system do not change over time
- Unsteady flow field: Time affects the behavior of the flow, and the vectors change over time

Visualizing Flow Fields using Flowlines

- Streamline
- Pathline
- Streakline
- Timeline

Flowlines: Streamline

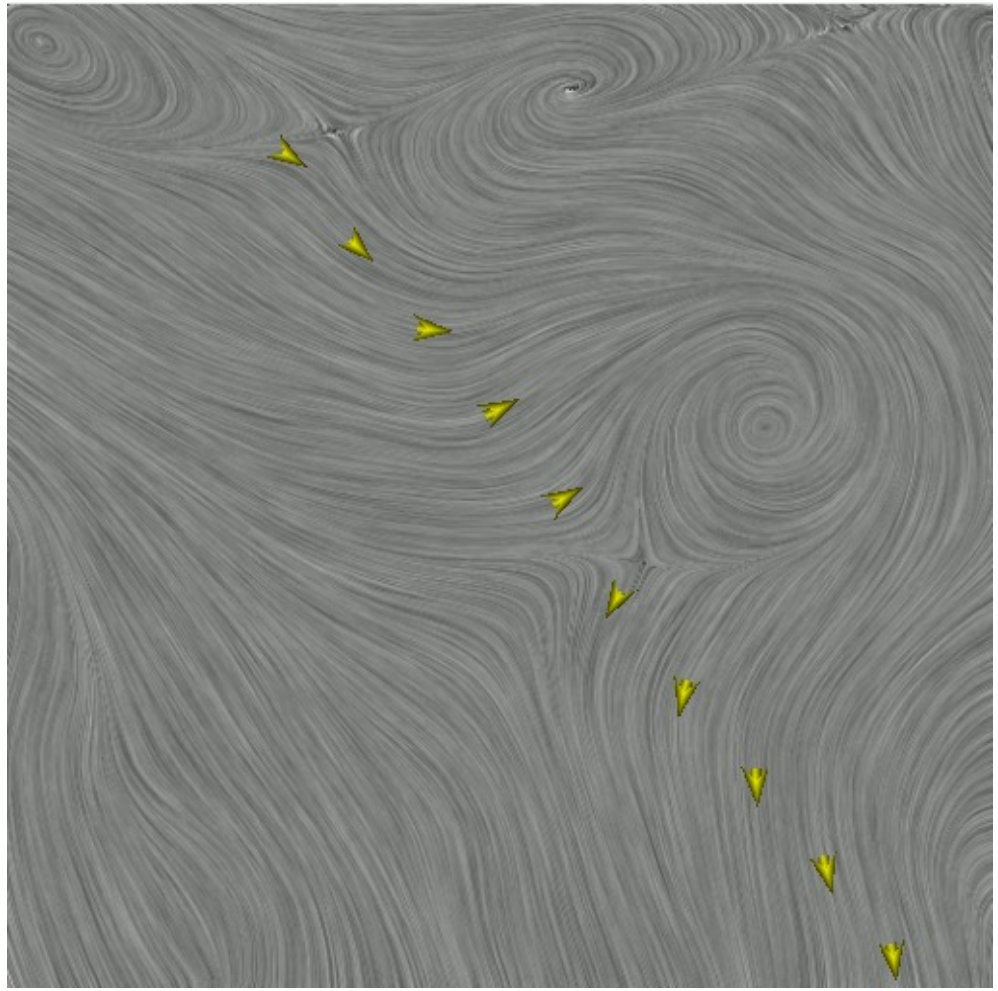
- A streamline is the path that a massless particle will follow if released in a steady flow field
- Streamline: A streamline is a curve tangent to the flow field everywhere
- Streamlines are drawn in steady flow



Red lines are streamlines

Flowlines: Pathline

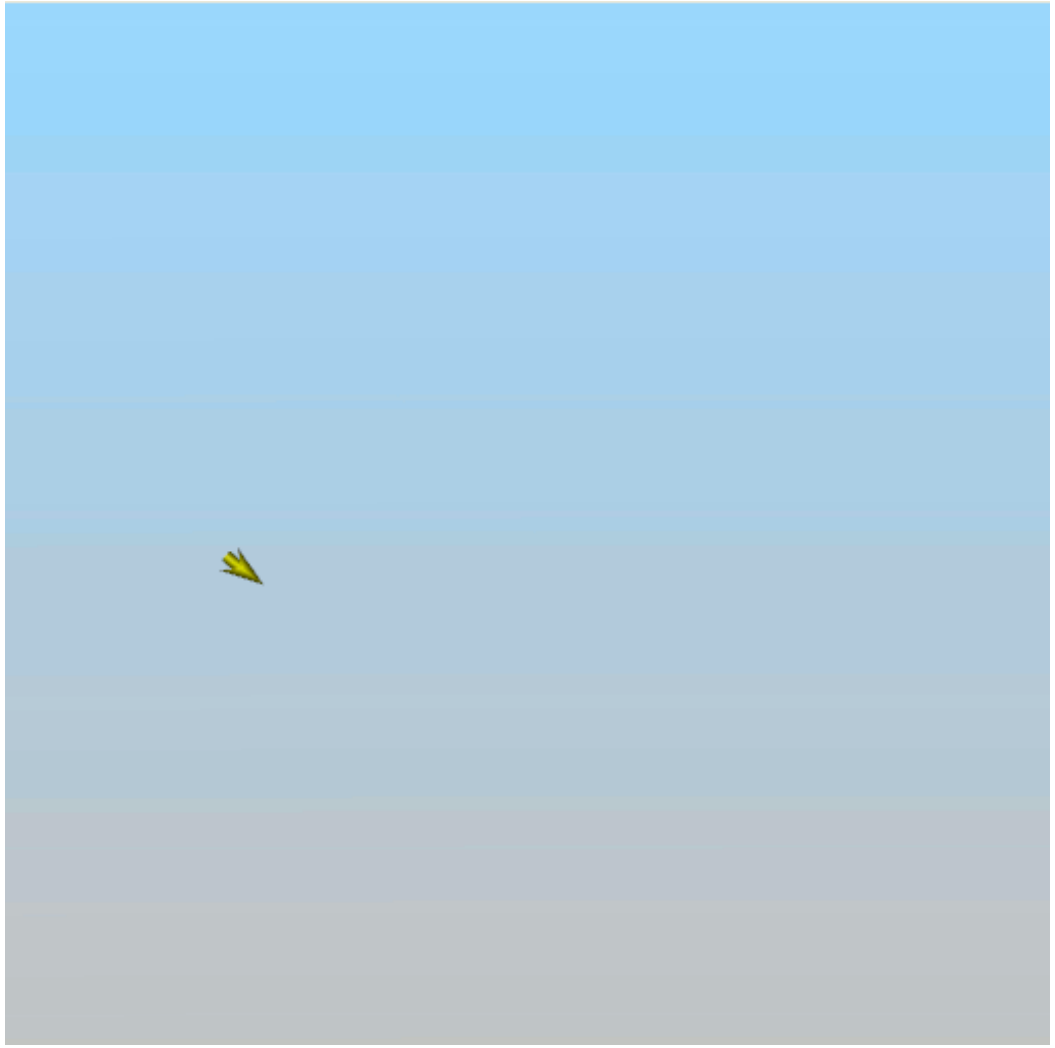
- Pathline: A pathline is the trajectory that an individual fluid particle will follow in an unsteady flow field
- The concept of pathline is similar to that of streamline except that the underlying flow field is unsteady



Yellow lines are pathlines

Flowlines: Streakline

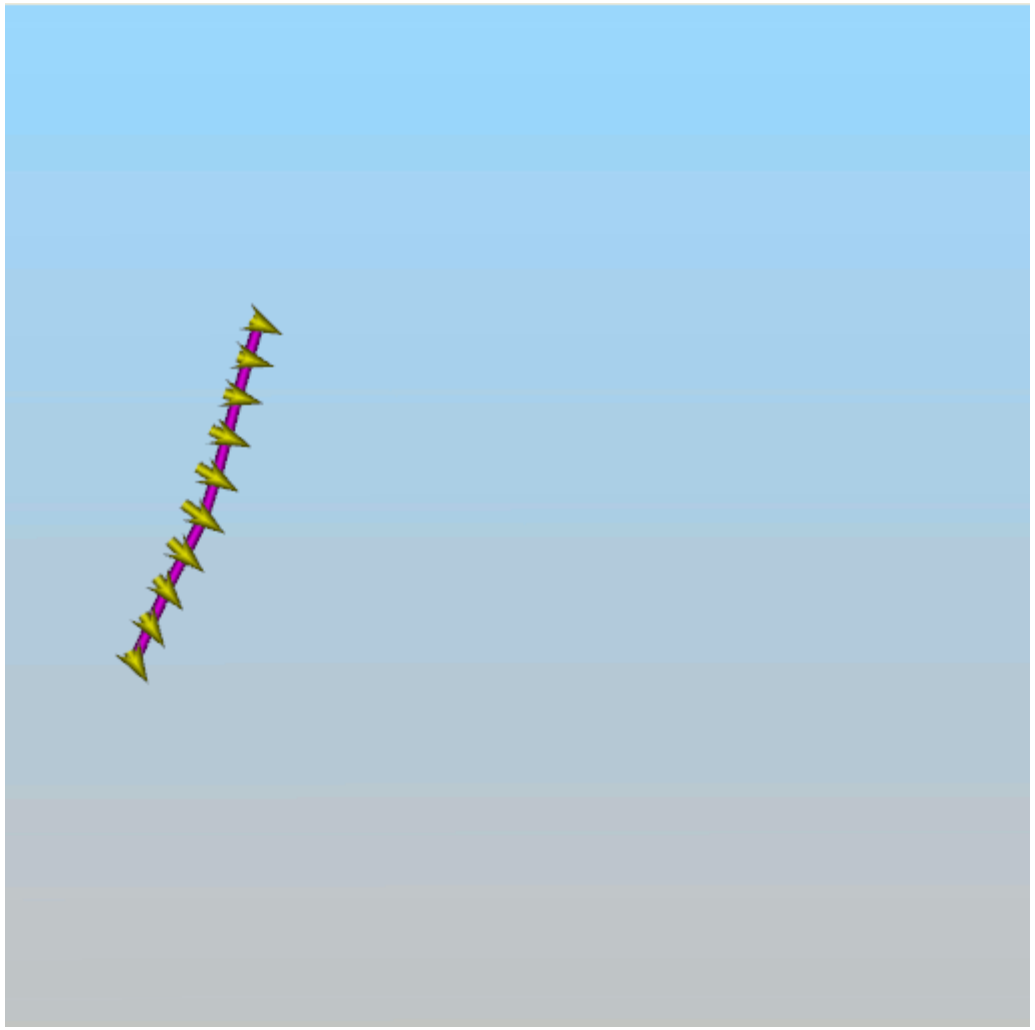
- Streakline: A streakline is the locus of points of all the fluid particles that have passed continuously through a particular spatial point in the past
- If we place multiple small balls into the water flow at the same position but at different time steps, the streakline is the path by connecting all the balls in the placement order



Green line is are streakline

Flowlines: Timeline

- Timeline: A timeline is a line formed by a set of fluid particles that were marked at a previous instant in time, creating a curve that is displaced over time as the particles move
- Imagine that we place several small balls into a water flow and allow the balls to follow the flow. At a certain time, step, the path that connects all the balls is a timeline.



Red line is timeline

Streamlines

- Streamline: A line that is tangential to the instantaneous velocity direction
- Velocity is a vector, and it has a magnitude and a direction
- Release a particle into the flow and perform numerical integration to compute the path of the particle

Computing Flowlines: Particle Tracing

- The path of a massless particle at position p at time t can be described by the following ordinary differential equation (ODE):

$$\frac{d\mathbf{p}}{dt} = v(\mathbf{p}(t))$$

steady flow

or

$$\frac{d\mathbf{p}}{dt} = v(\mathbf{p}(t), t)$$

unsteady flow

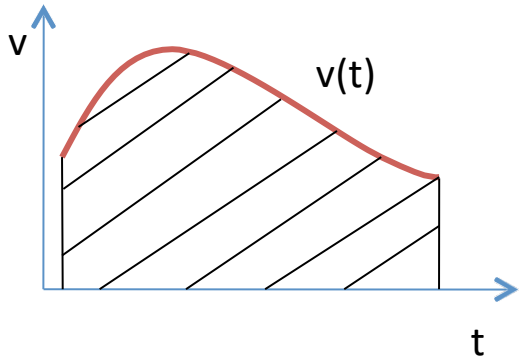
- And the positions of the particle can be computed by

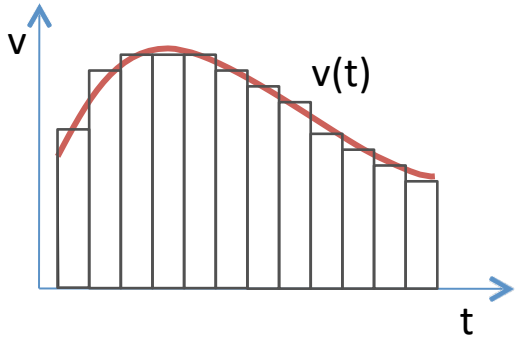
$$p(t + \Delta t) = p(t) + \int_{t}^{t+\Delta t} v(p(t), t)dt$$

Solved by numerical integration

Numerical Integration for Particle Tracing

- Discrete approximation of the continuous integration
- Calculate the area under the curve of v(t)
  - Trapezoidal approximation
  - Error related to the step size used





Euler's Method

- Typically, not recommended due to low accuracy

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \boldsymbol{v}(\boldsymbol{p}_k) \times \Delta t$$

- The equation can be derived from Taylor series expansion

$$y(t_0 + h) = y(t_0) + hy'(t_0) + \frac{1}{2}h^2y''(t_0) + ..$$

$$y(t_0 + h) = y(t_0) + hy'(t_0) + O(h^2) \longleftarrow \text{Error}$$

2nd Order Runge-Kutta (RK-2) Integration

- Improved accuracy than Euler's method, more commonly used

$$\mathbf{p}^* = \mathbf{p}_k + \mathbf{v}(\mathbf{p}_k)\,\Delta t$$

$$\mathbf{p}_{k+1} = \mathbf{p}_k + (\mathbf{v}(\mathbf{p}_k) + \mathbf{v}(\mathbf{p}^*))\ x\,\Delta t/2$$

4th Order Runge-Kutta (RK4) Integration

- Better accuracy, recommended method

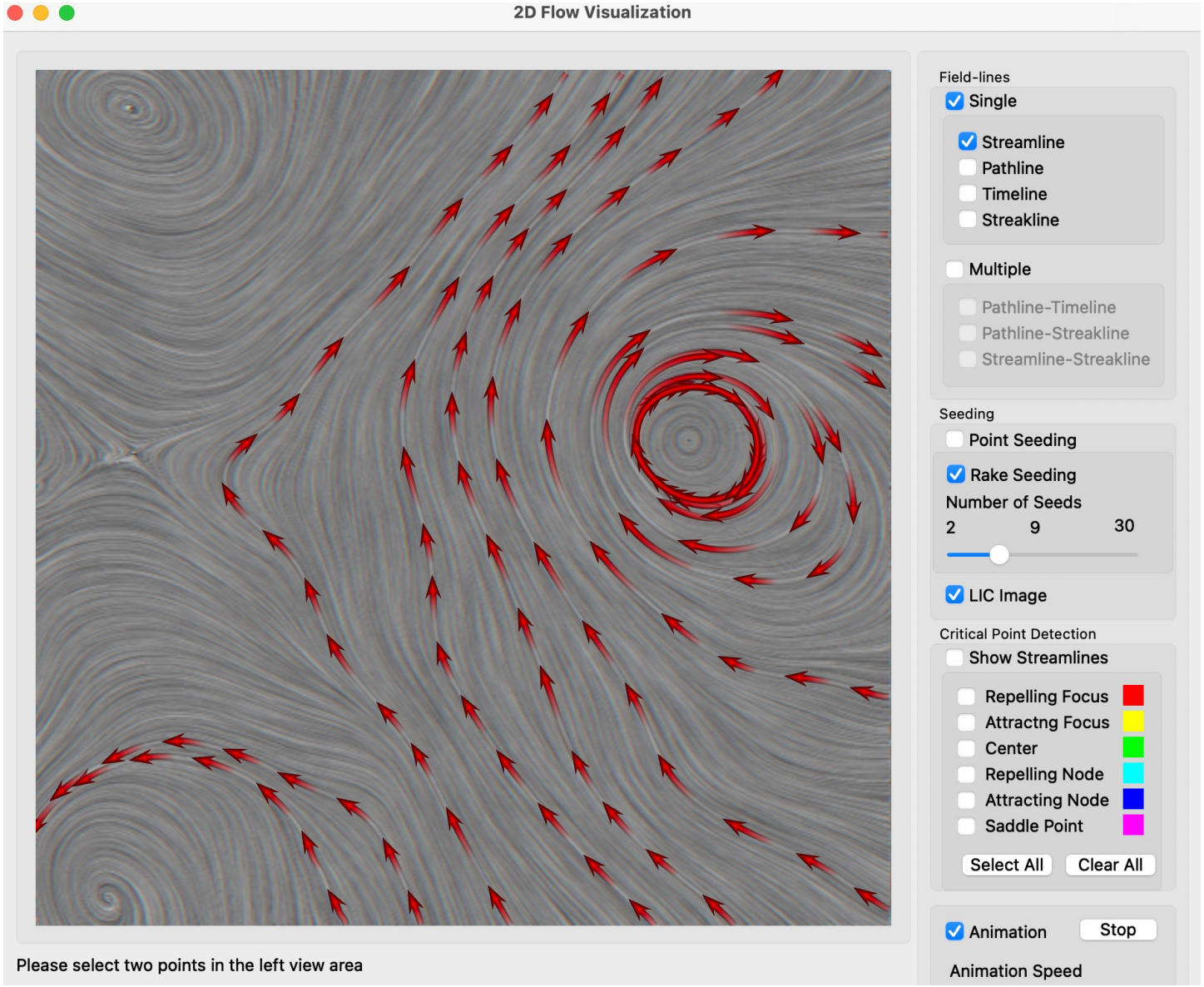
$$\boldsymbol{a} = \Delta t\ \boldsymbol{v}(\boldsymbol{p}_k),$$
$$\boldsymbol{b} = \Delta t\ \boldsymbol{v}(\boldsymbol{p}_k + \boldsymbol{a}/2),$$
$$\boldsymbol{c} = \Delta t\ \boldsymbol{v}(\boldsymbol{p}_k + \boldsymbol{b}/2)$$
$$\boldsymbol{d} = \Delta t\ \boldsymbol{v}(\boldsymbol{p}_k + \boldsymbol{c}),$$

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + (\boldsymbol{a} + 2\boldsymbol{b} + 2\boldsymbol{c} + \boldsymbol{d})/6$$

Particle Tracing Algorithm

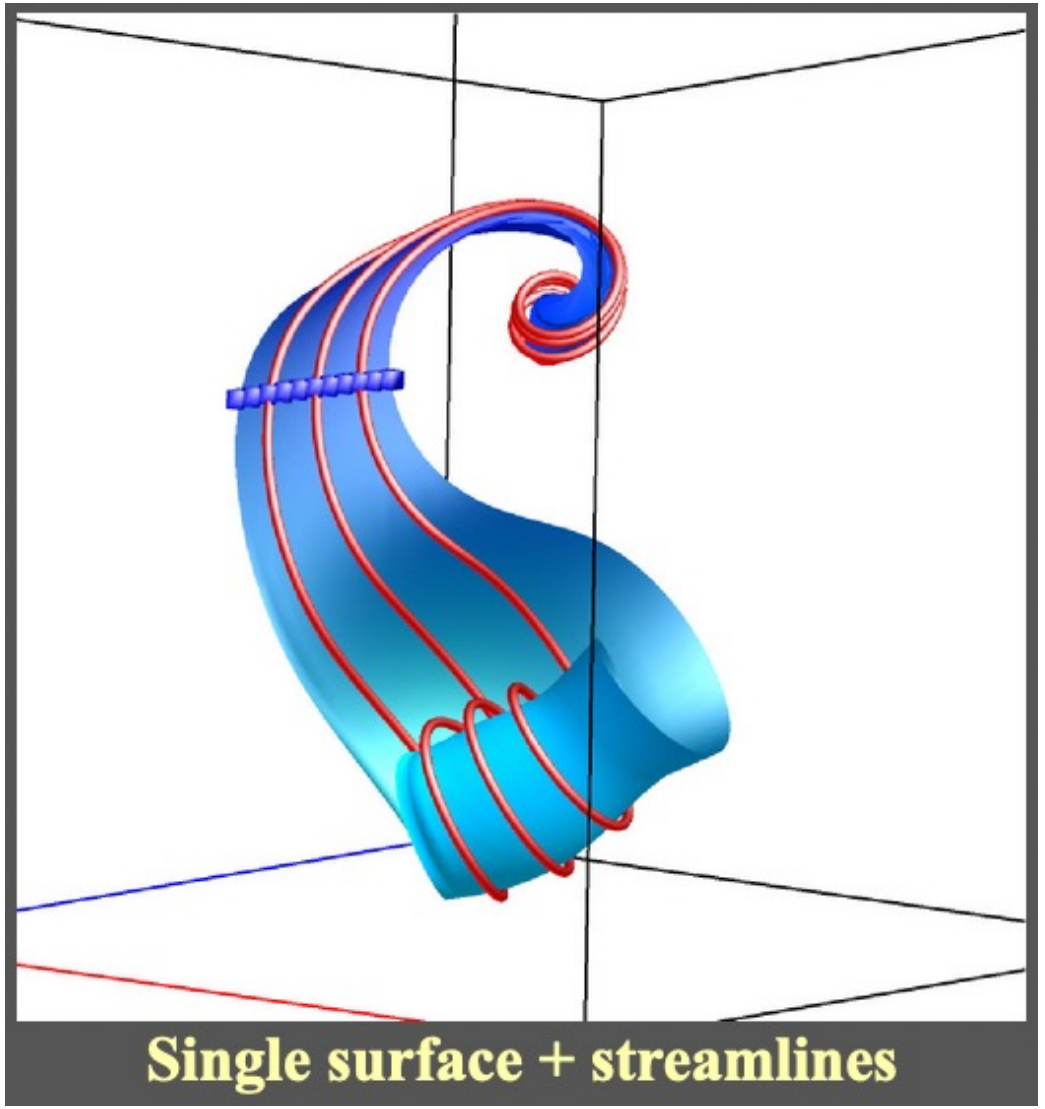
1. Specify a seed position p(0), t = 0
2. Perform cell search to locate the cell that contains the p(t)
3. Interpolate the velocity field to determine the velocity at p(t)
4. Advance the particle from p(t) to p(t+Δt) using a numerical integration method
5. Repeat from step 2 until the particle moves a certain distance or goes out of bound

Demo


2D Flow Visualization
Field-lines
Single
Streamline
Pathline
Timeline
Streakline
Multiple
Pathline-Timeline
Pathline-Streakline
Streamline-Streakline
Seeding
Point Seeding
Rake Seeding
Number of Seeds
2
9
30
LIC Image
Critical Point Detection
Show Streamlines
Repelling Focus
Attractng Focus
Center
Repelling Node
Attracting Node
Saddle Point
Select All
Clear All
Animation
Stop
Animation Speed
Please select two points in the left view area

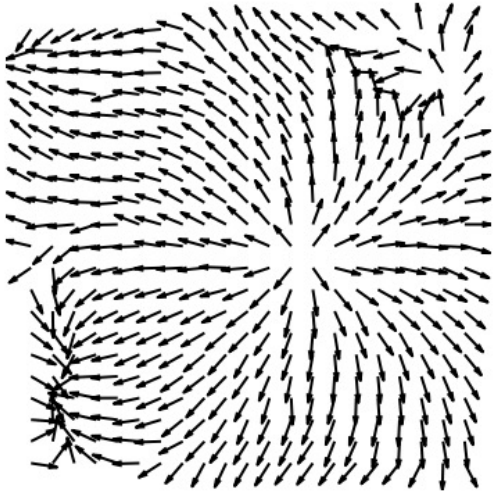
Stream Surface

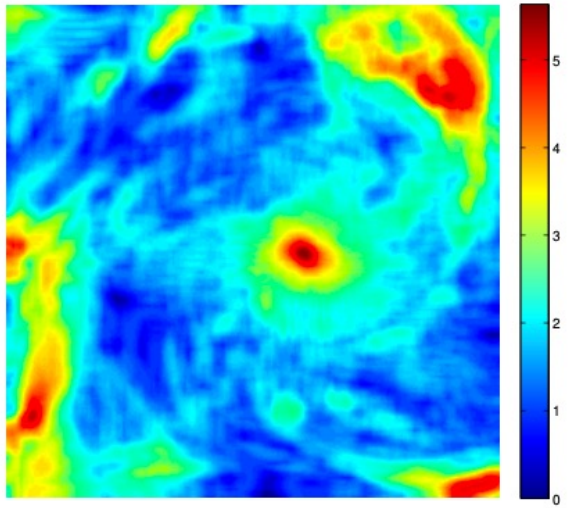
- A stream surface is a continuous surface that is everywhere tangent to the vector it passes, which can be obtained from streamlines traced from a densely seeded curve.



Seed Placement in Flow Field

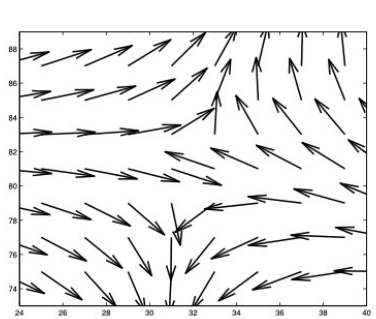
- Place seeds in the vector field where the field has more information
  - Orientation of the vectors are more diverse
- Information Entropy can help
  - Compute the entropy in the local neighborhood around each grid point
  - The value in the entropy field for a point indicates the degree of vector variation in its local neighborhood

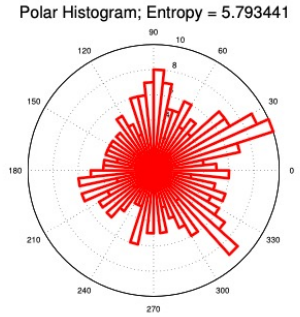


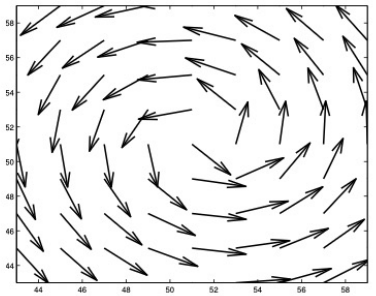


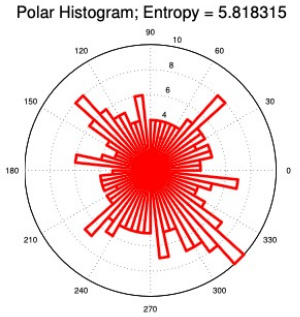
Seed Placement in Flow Field

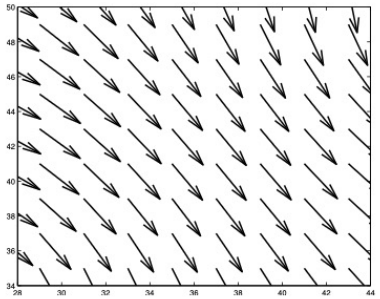
- How do we compute distribution of the vector field?
  - Can be done by first partitioning the range of the vectors, represented as a polar angle θ , $0 \leq \theta \leq 2\pi$ for two dimensional vectors into a finite number of bins x_i , $i = 1 \ldots n$

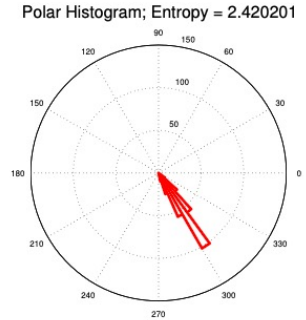


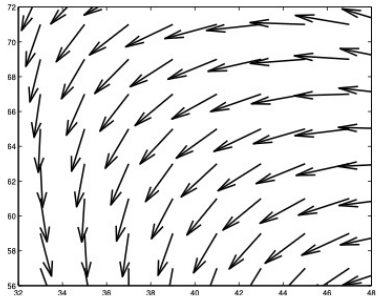


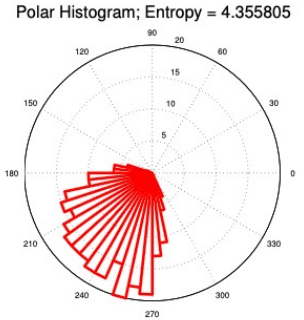




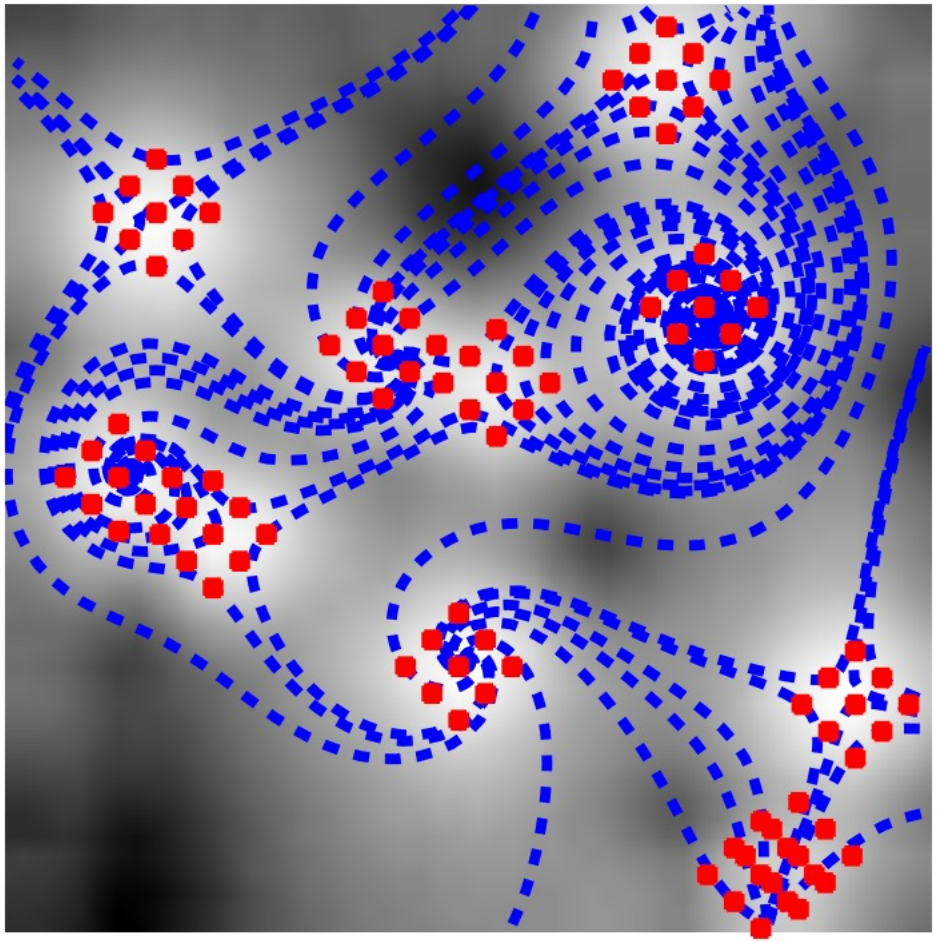


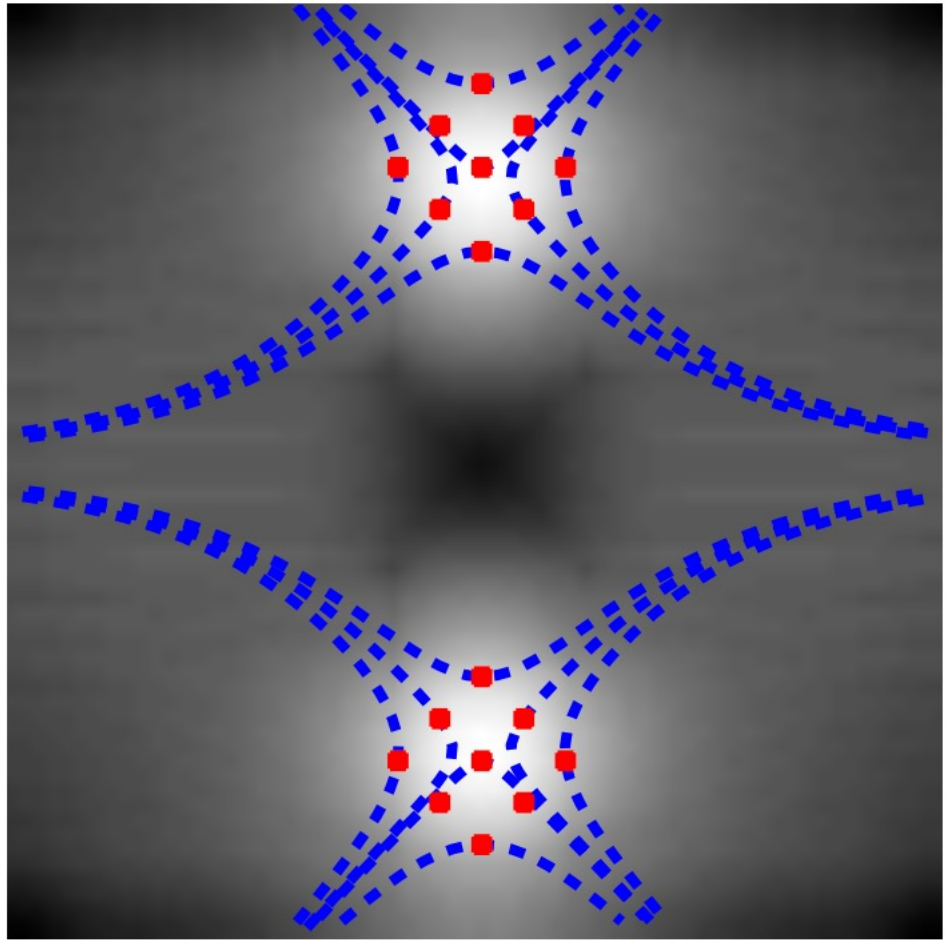






Seed Placement in Flow Field





Red points are seeds placed on high entropy regions; blue lines are streamlines