Spatial Visualization: Vector and Tensor Fields

CS 6630, Fall 2016 — Alex Lex
Aaron Knoll, guest lecturer

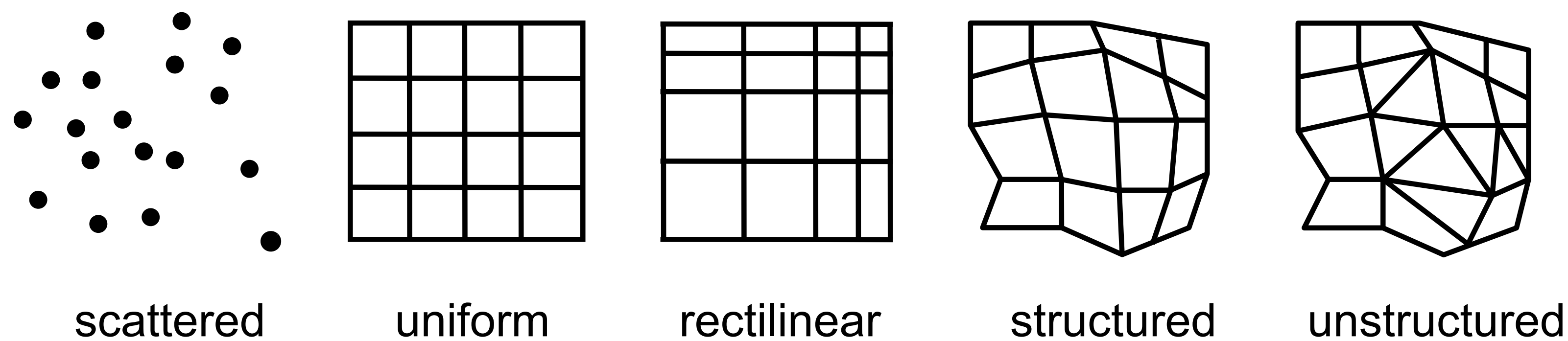
Slides thanks to:
Joshua Levina, Clemson University
Guoning Chen, University of Texas at Houston
Gordon Kindlmann, University of Chicago
Robert Laramee, Swansea University
Christoph Garth, University of Kaiserslautern

www.sci.utah.edu

Vector field (flow) visualization

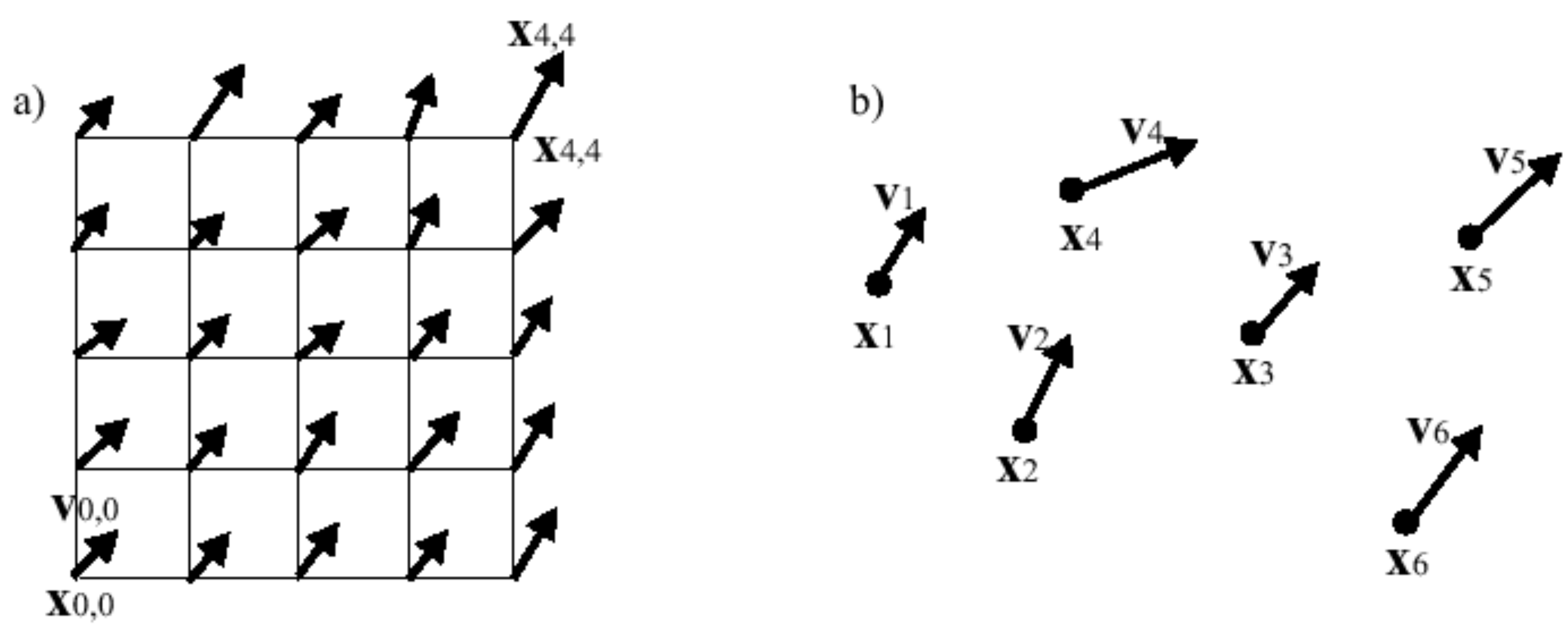
Vector fields

- Vector data on a 2D or 3D grid



- Additional scalar data may be defined per grid point

- Example on a regular grid (a) or scattered data points (b)



More formally

scalar field

$$s : \mathbb{E}^n \rightarrow \mathbb{R}$$

vector field

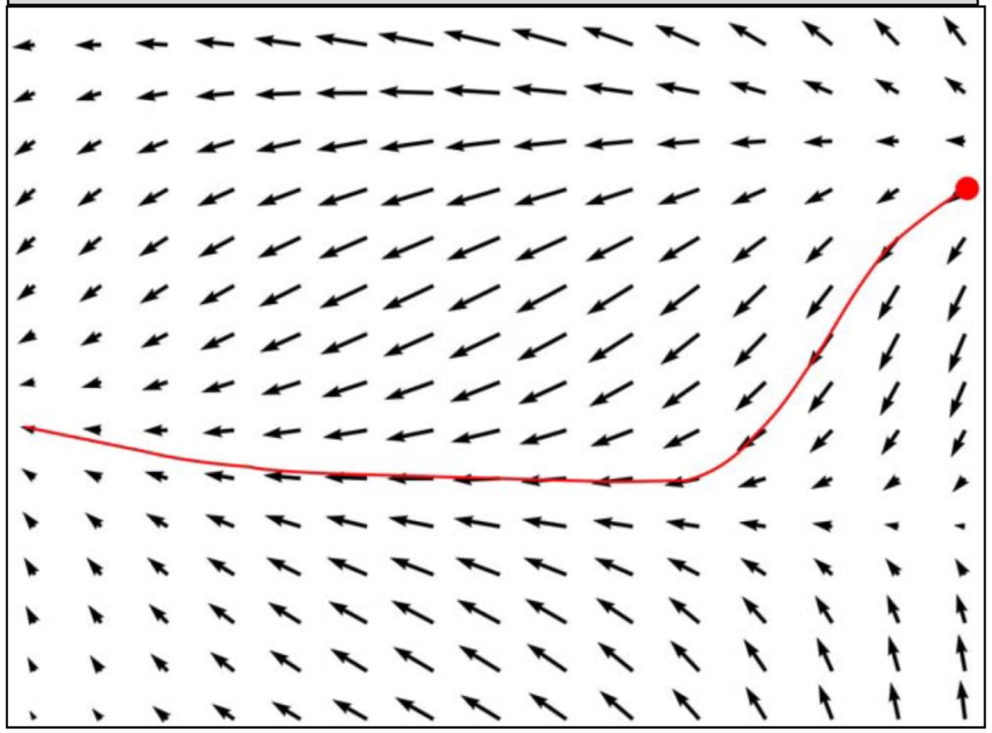
$$\mathbf{v} : \mathbb{E}^n \rightarrow \mathbb{R}^m$$

- m=n *usually* — but not always.
- *The vector is the element of the field*
  (in contrast to multifields)
- Typically, the vector field can be expressed as an ordinary differential equation (ODE), e.g.,

$$\frac{d\varphi(x)}{dt} = V(x)$$

- Solving (integrating) this ODE results in **flow,** i.e. the set of particle trajectories in this field.
- *Flow vis is about how we select and show these trajectories.*

A simple 2D **steady** vector field



- Main application of vector field visualization is flow visualization
  - Motion of fluids (gas, liquids)
  - Geometric boundary conditions
  - Velocity (flow) field $\boldsymbol{v}(\boldsymbol{x},t)$
  - Pressure p
  - Temperature T
  - Vorticity $\nabla\times\boldsymbol{v}$
  - Density ρ
  - Conservation of mass, energy, and momentum
  - Navier-Stokes equations
  - CFD (Computational Fluid Dynamics)

Experimental flow visualization

Milestones in Flight History
Dryden Flight Research Center

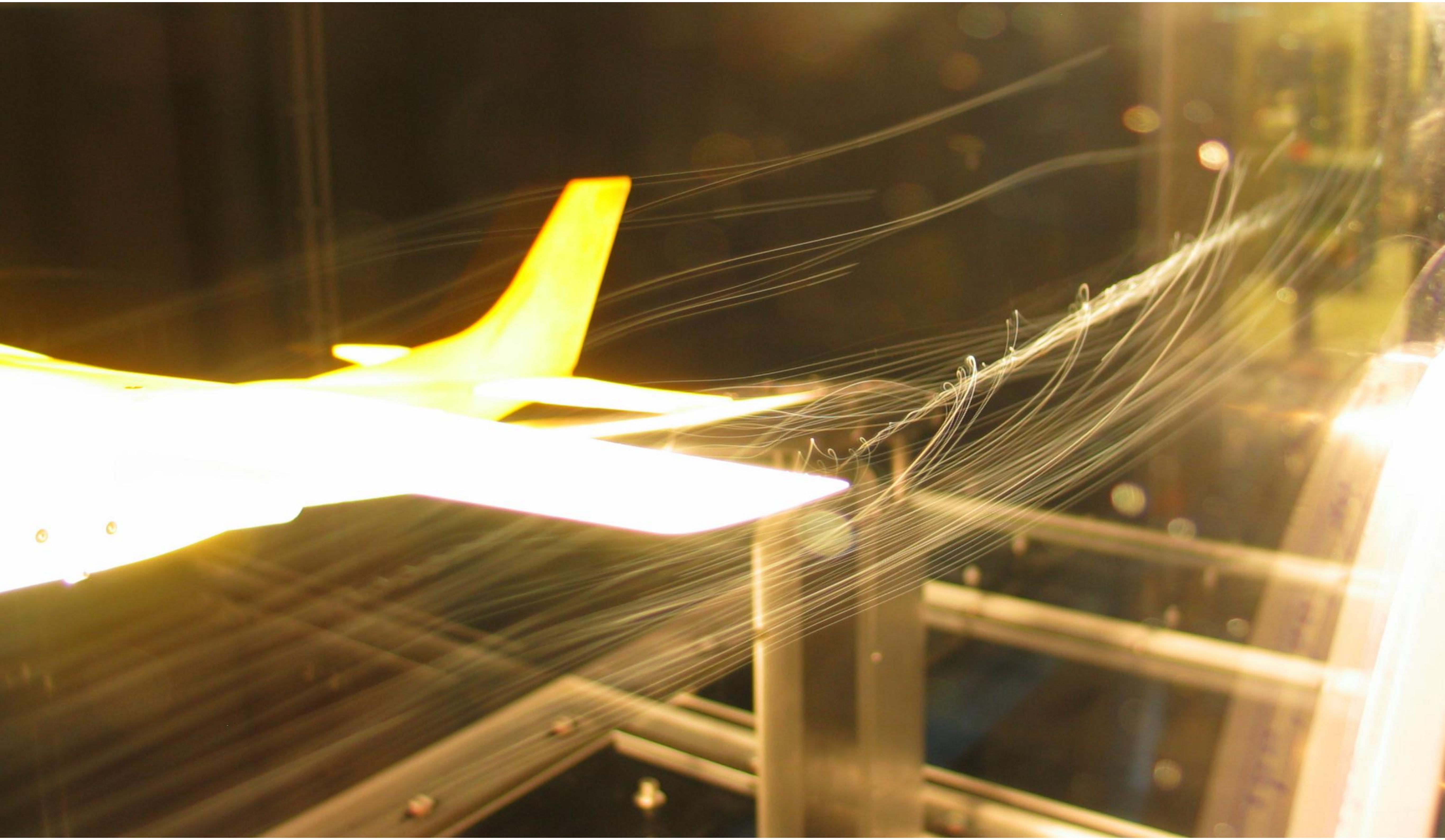
L-1011

Airliner Wing Vortice Tests at Langley

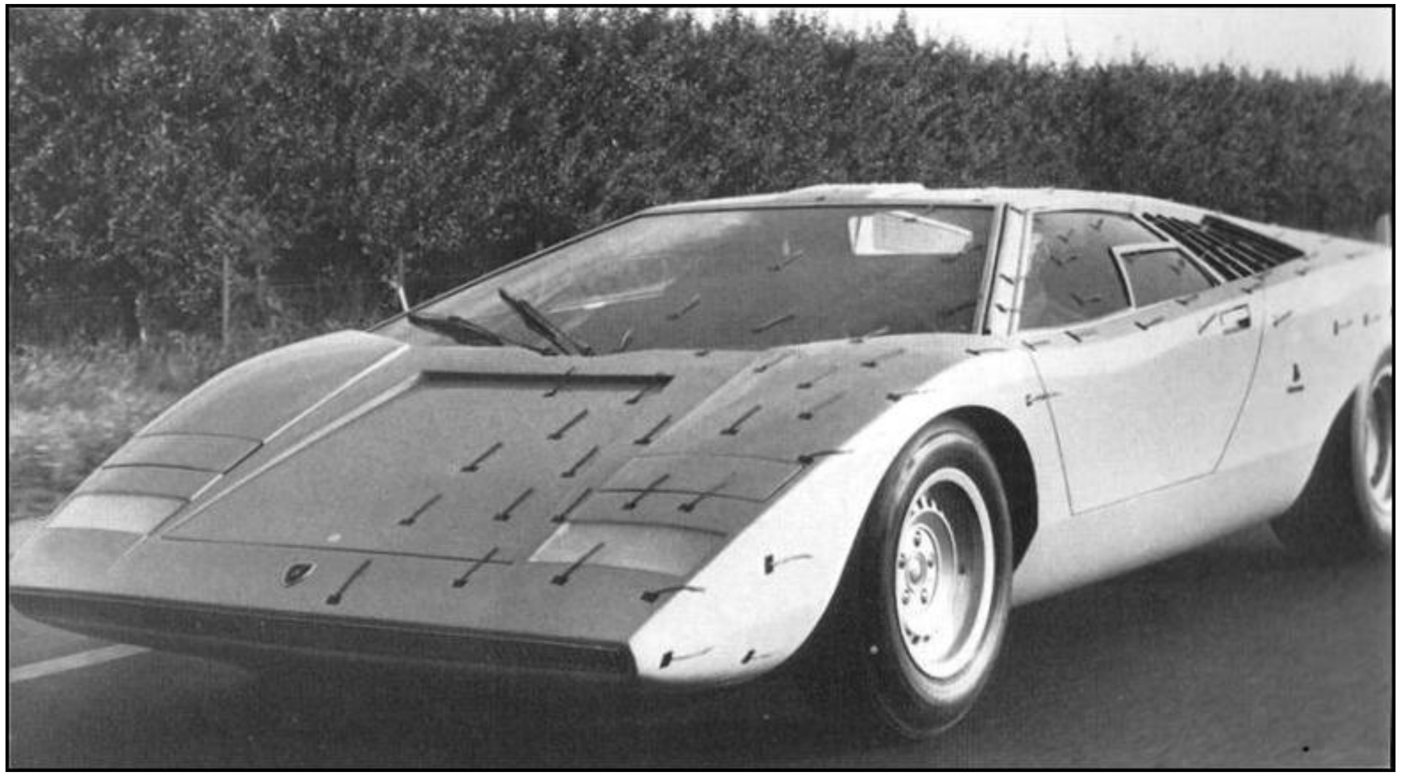
Circa 1970s

Smoke angel
A C-17 Globemaster III from the 14th Airlift Squadron, Charleston Air Force Base, S.C. flies off after releasing flares over the Atlantic Ocean near Charleston, S.C., during a training mission on Tuesday, May 16, 2006. The "smoke angel" is caused by the vortex from the engines.
(U.S. Air Force photo/Tech. Sgt. Russell E. Cooley IV)

http://de.wikipedia.org/wiki/Bild:Airplane_vortex_edit.jpg



A wind tunnel model of a Cessna 182 showing a wingtip vortex.
Tested in the RPI (Rensselaer Polytechnic Institute) Subsonic Wind Tunnel.
By Ben FrantzDale (2007).



http://autospeed.com/cms/A_108677/article.html

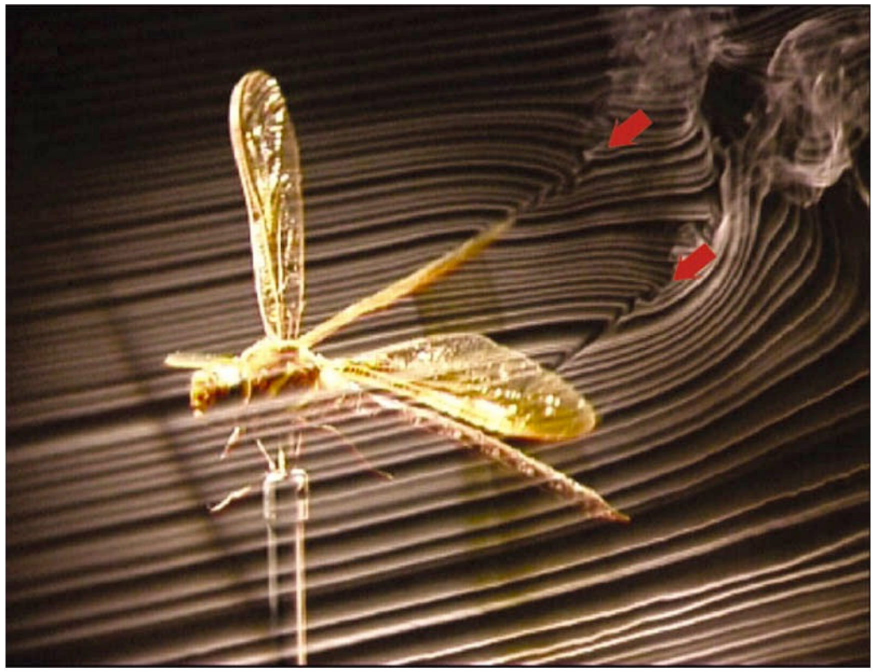
Wool Tufts

http://autospeed.com/cms/A_108677/article.html

Smoke Injection

http://autospeed.com/cms/A_108677/article.html

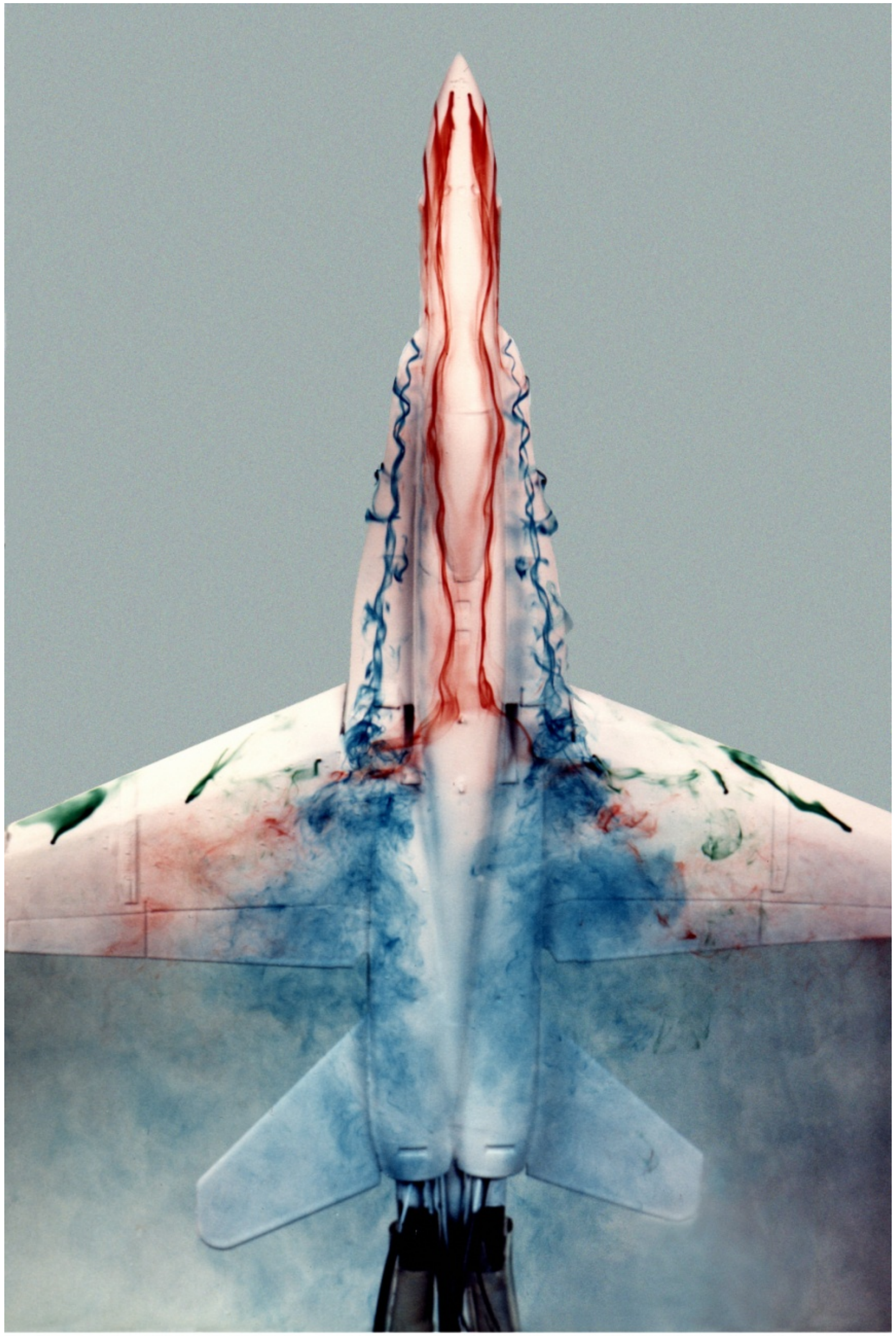
Smoke Nozzles



[NASA, J. Exp. Biol.]

http://autospeed.com/cms/A_108677/article.html

Streaklines in Experimental Flow Vis



NASA Dryden Flight Research Center Photo Collection
http://www.dfrc.nasa.gov/gallery/photo/index.html
NASA Photo: ECN-33298-03 Date: 1985

1/48-scale model of an F-18 aircraft in Flow Visualization Facility (FVF)

Dryden Flight Research Center ECN 33298-47 Photographed 1985
F-18 water tunnel test in Flow Visualization Facility NASA/Dryden

Computational fluid dynamics

Fluid dynamics

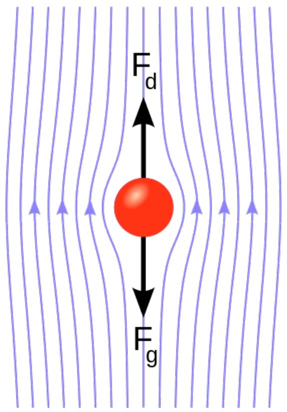
- **Navier-Stokes equations:** a set of PDE's modeling the behavior of fluids.
  Example for compressible fluids:

$$\underbrace{\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right)}_{1} = \underbrace{-\nabla p}_{2} + \underbrace{\nabla\cdot\left(\mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - \frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{I}\right)}_{3} + \underbrace{\mathbf{F}}_{4}$$

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) = 0$$

Continuity equation

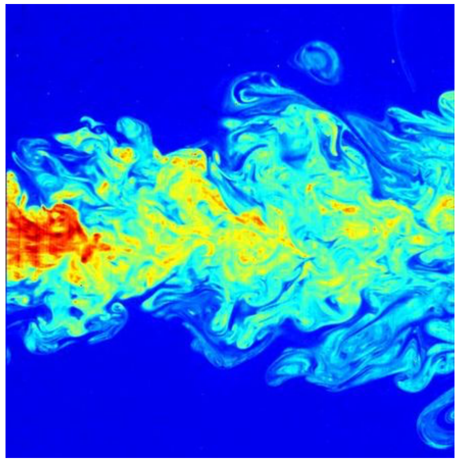
where u is the fluid velocity, *p* is the fluid pressure, *ρ* is the fluid density, and *μ* is the viscoscity.



Laminar flow

- **Conservation of mass**, **momentum**, **energy** (relate to 2nd law of thermodynamics).
- **Viscosity** is the measure of the fluid's resistance to deformation, from shear or tensile stress.
  (A stress tensor with 9 degrees of freedom!)
- Flow can be **steady** (time derivative $\frac{\partial \rho}{\partial t} = 0$) or **unsteady** (or **transient,** i.e. high time derivative)
- Also **laminar** (flows in predictable, parallel layers) or **turbulent** (eddies, vortices, random chaos).
- **Reynolds number** indicates the turbulence of flow = inertial forces / viscous forces.

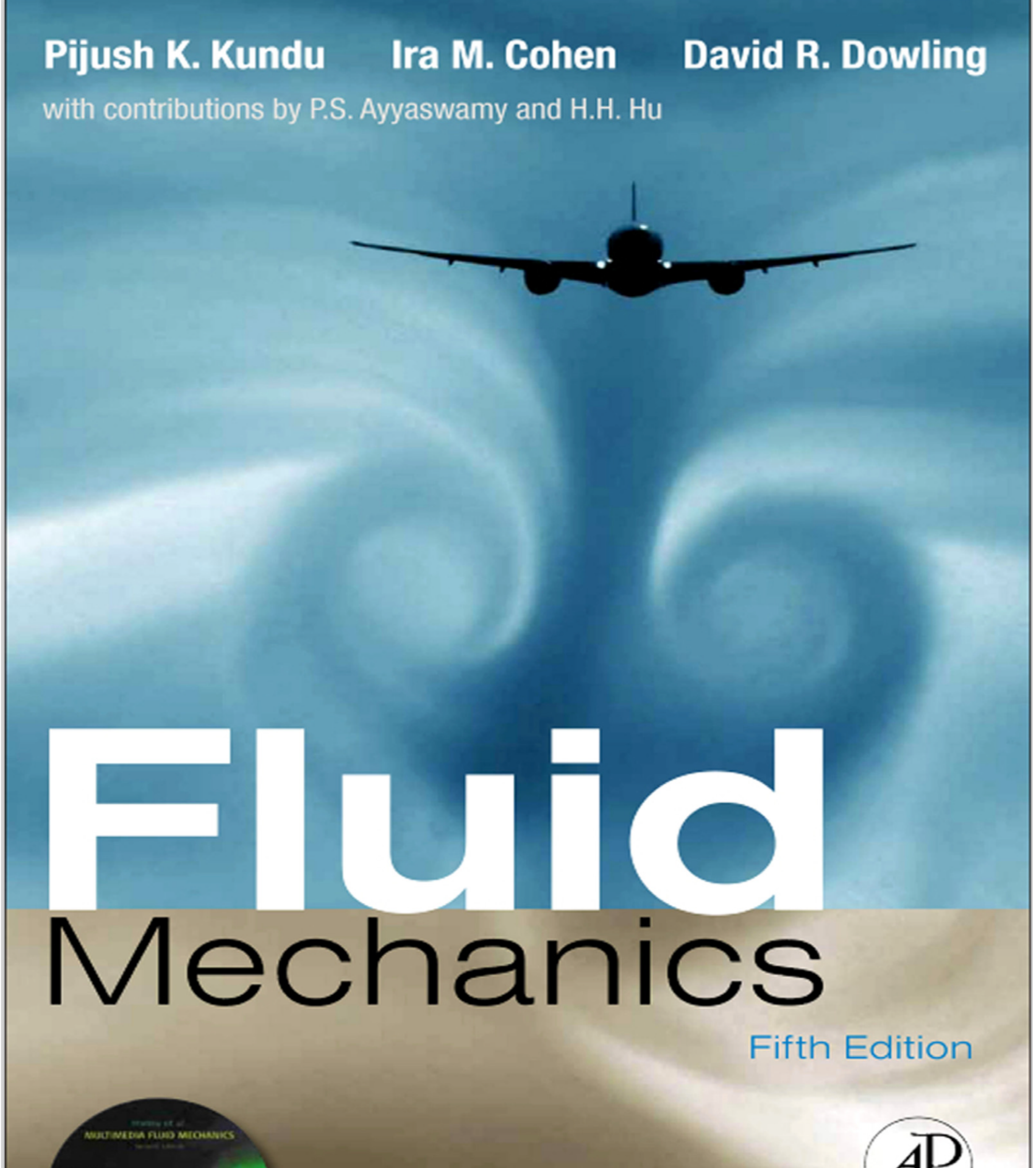
$$\mathrm{Re} = \frac{\text{inertial forces}}{\text{viscous forces}} = \frac{\rho\mathbf{v}L}{\mu} = \frac{\mathbf{v}L}{\nu}$$



Turbulent flow

https://www.comsol.com/multiphysics/navier-stokes-equations
https://en.wikipedia.org/wiki/Navier–Stokes_equations
https://en.wikipedia.org/wiki/Fluid_dynamics
https://en.wikipedia.org/wiki/Chaos_theory

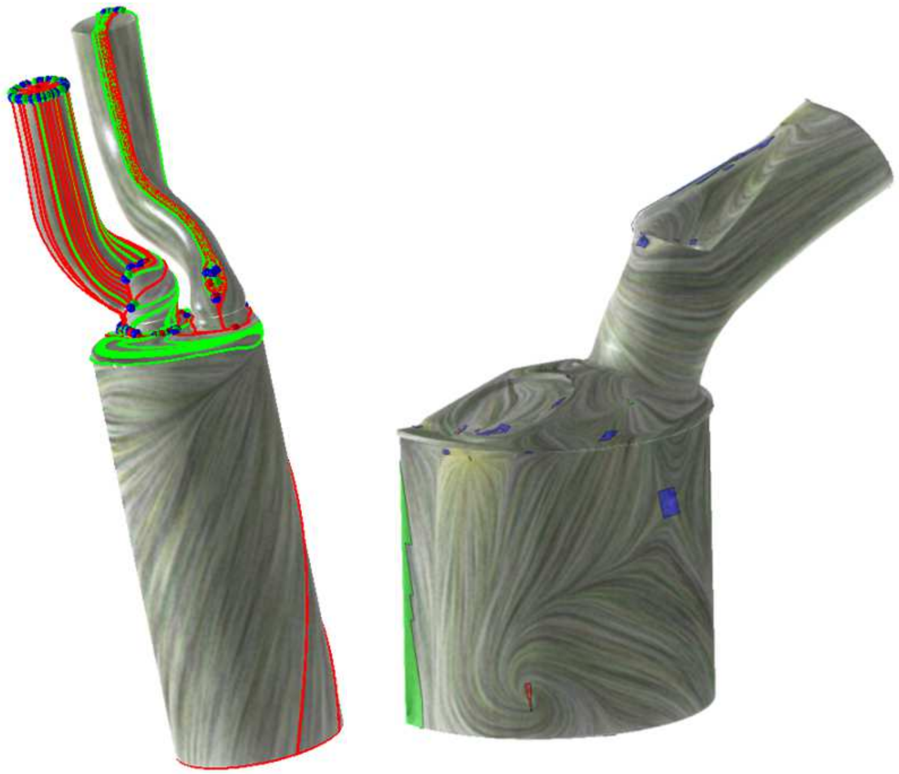
SCI www.sci.utah.edu



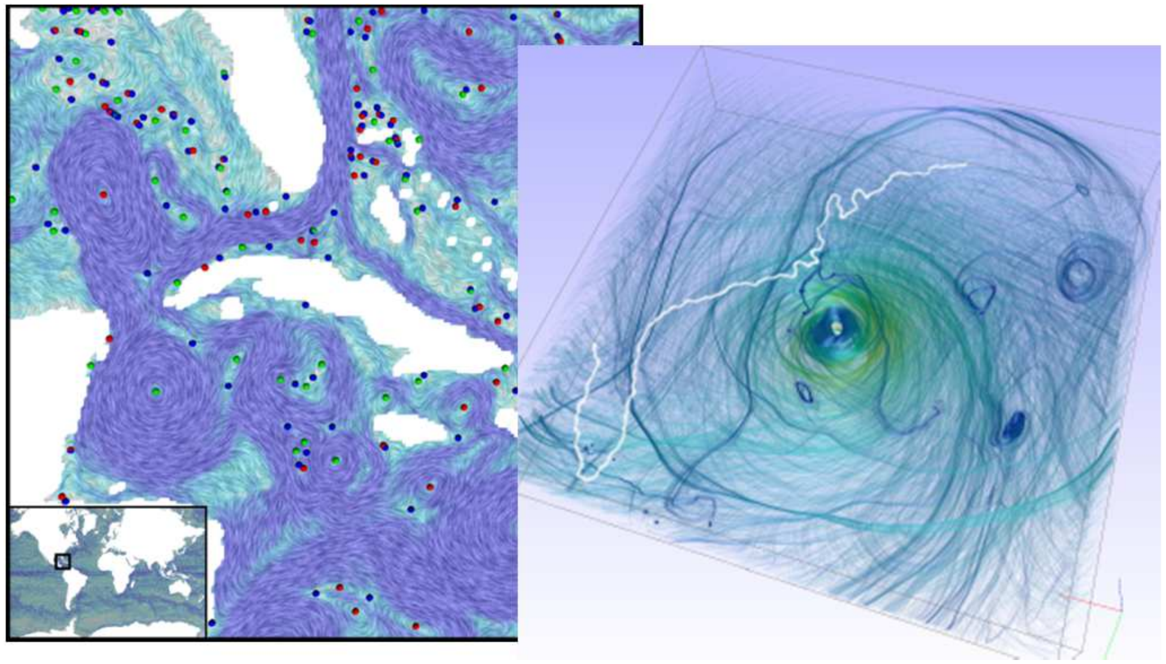
http://www3.nd.edu/~fthomas/Kundu_Fluid_Mechanics.pdf

www.sci.utah.edu

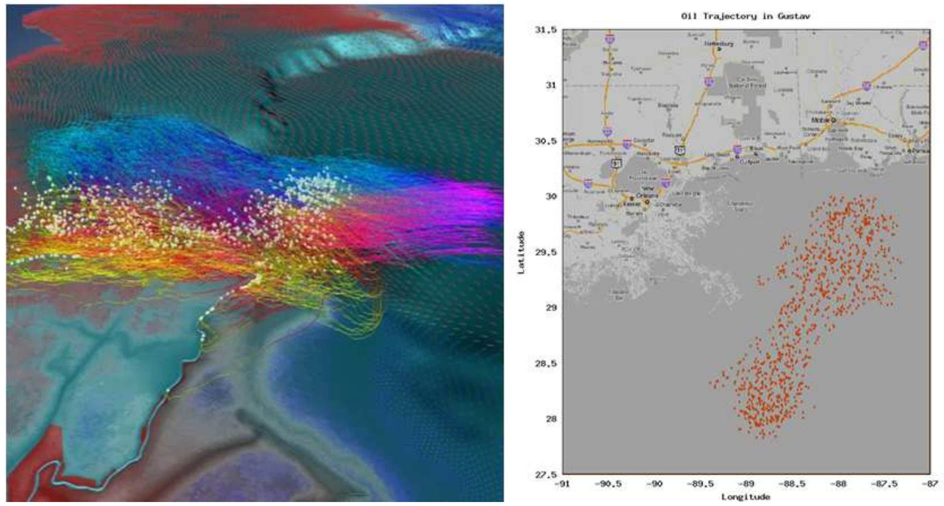
Vector Fields in Engineering and Science



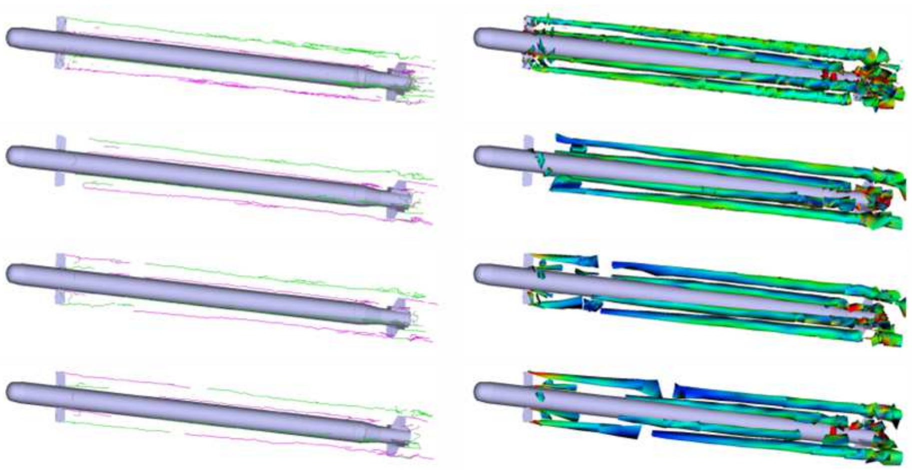
Automotive design
[**Chen** et al. TVCG07,TVCG08]



Weather study [Bhatia and Chen et al. TVCG11]

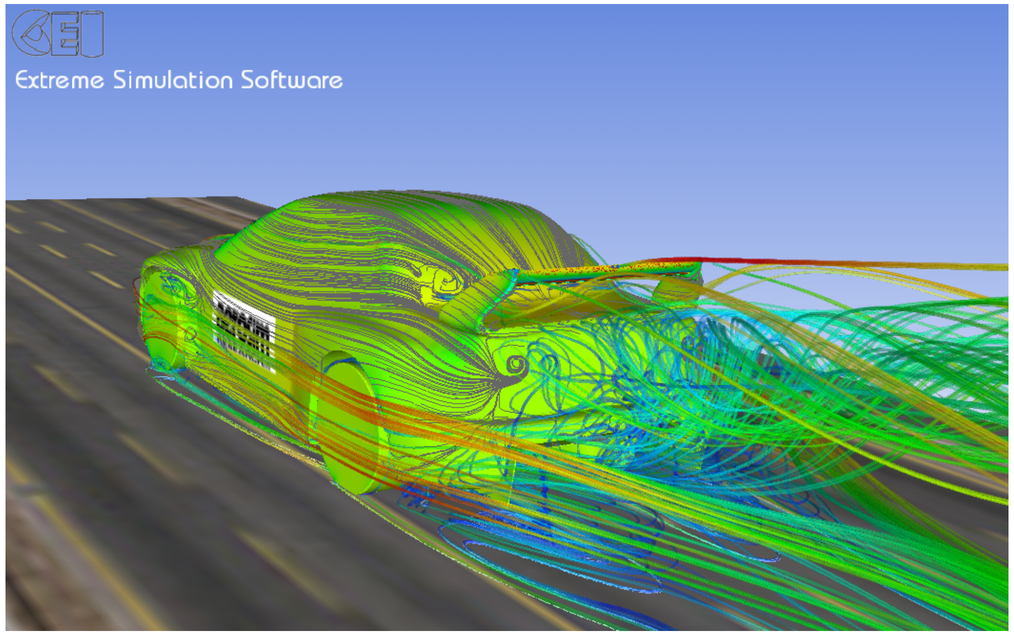


Oil spill trajectories [Tao et al. EMI2010]



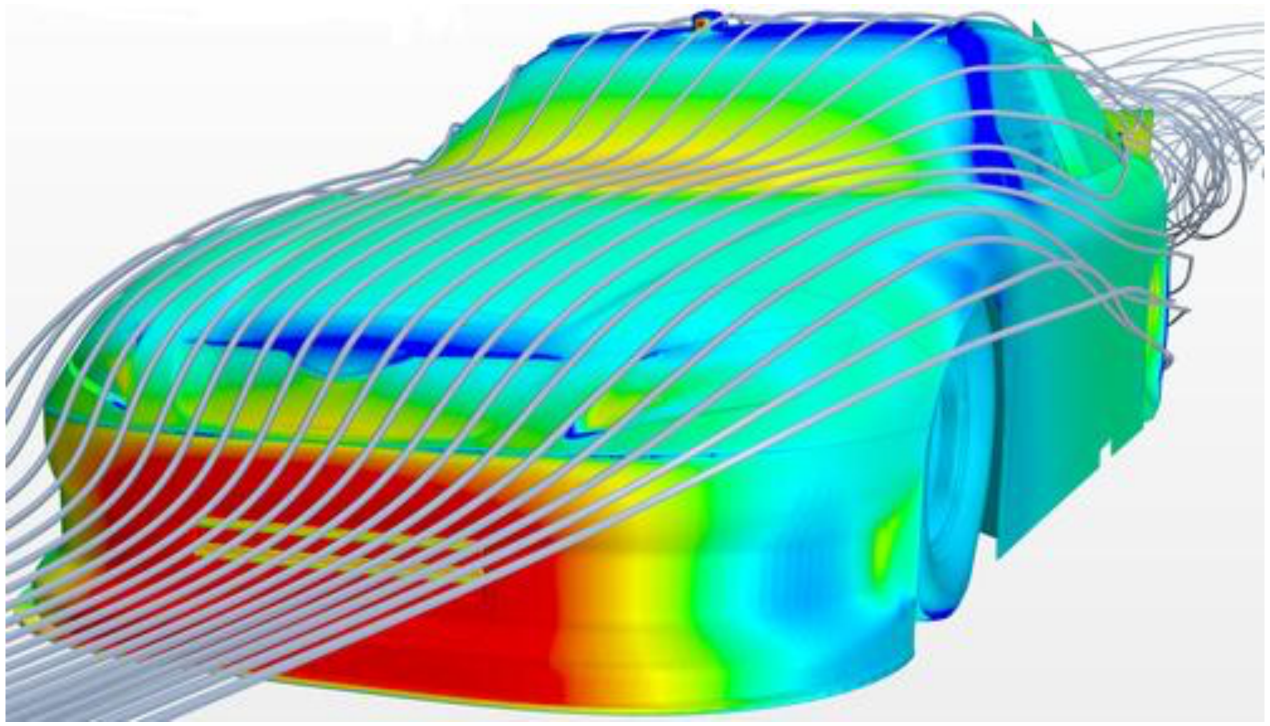
Aerodynamics around missiles [Kelly et al. Vis06]

Automotive body CFD simulations

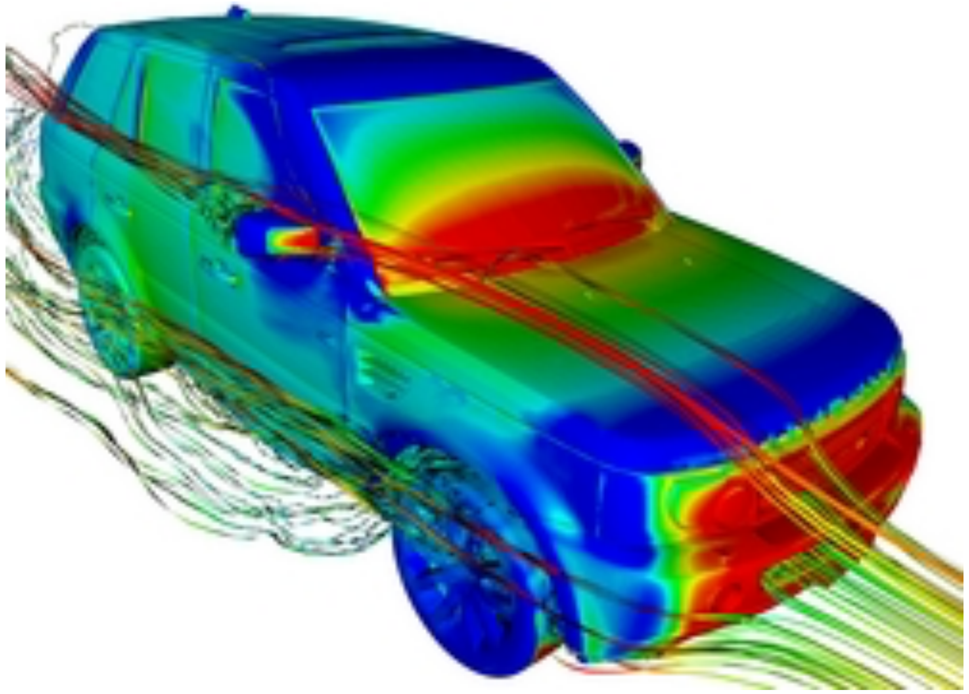


Flow visualization in Ensight
http://gallery.ensight.com/keyword/external%20aero;simulation/

F1 RANS simulation
http://www.symscape.com/blog/car-design-cfd

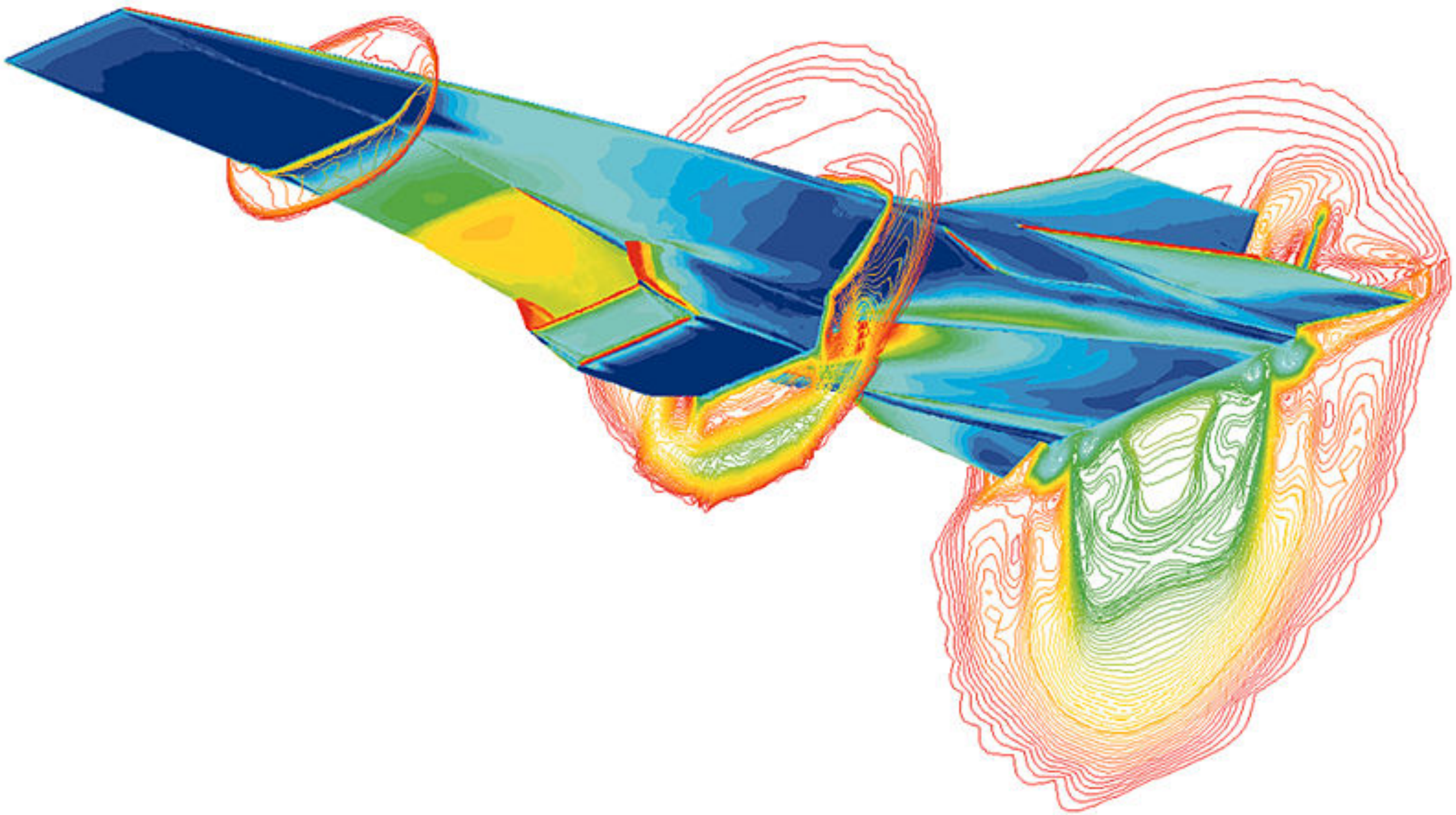


Michael Waltrip NASCAR
flow analysis in CD-adapco Star-CCM CFD tools

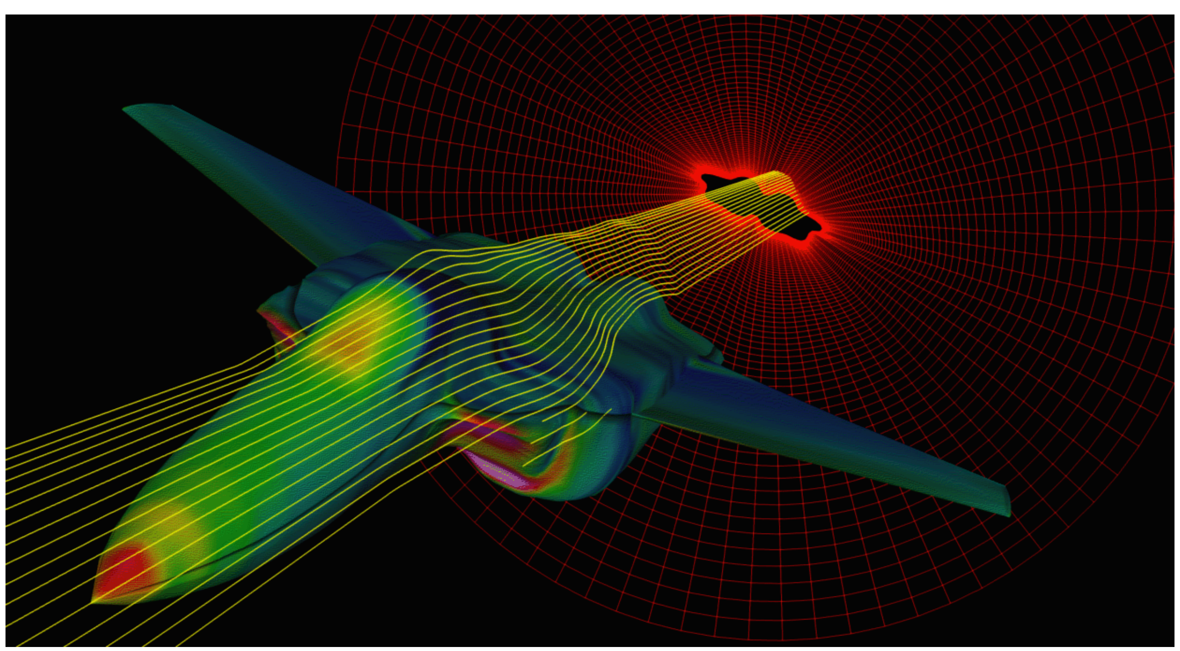


Jaguar Land Rover External Aerodynamic Simulation by Exa's PowerFLOW Software

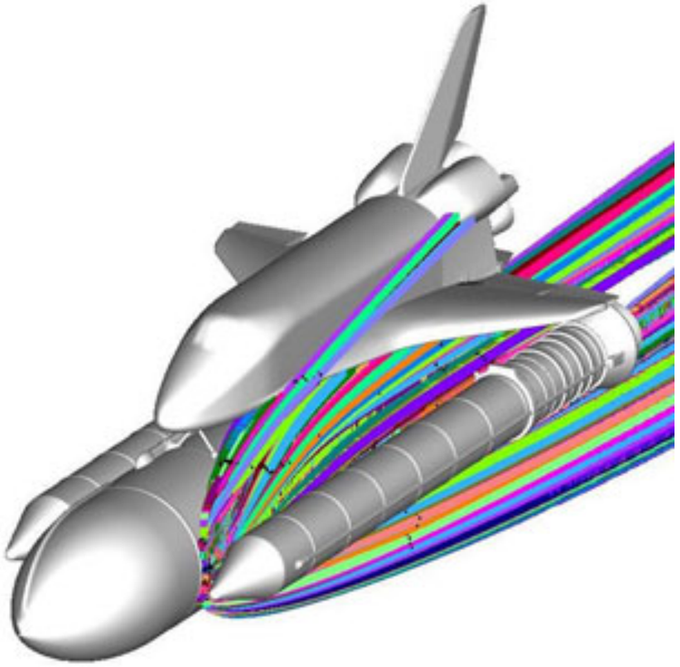
Aerospace



A simulation of the Hyper-X scramjet vehicle in operation at Mach-7. http://www.airports-worldwide.com/articles/article0523.php



FAST, http://www.openchannelfoundation.org



http://www.cesc.zju.edu.cn/learningcenter.htm

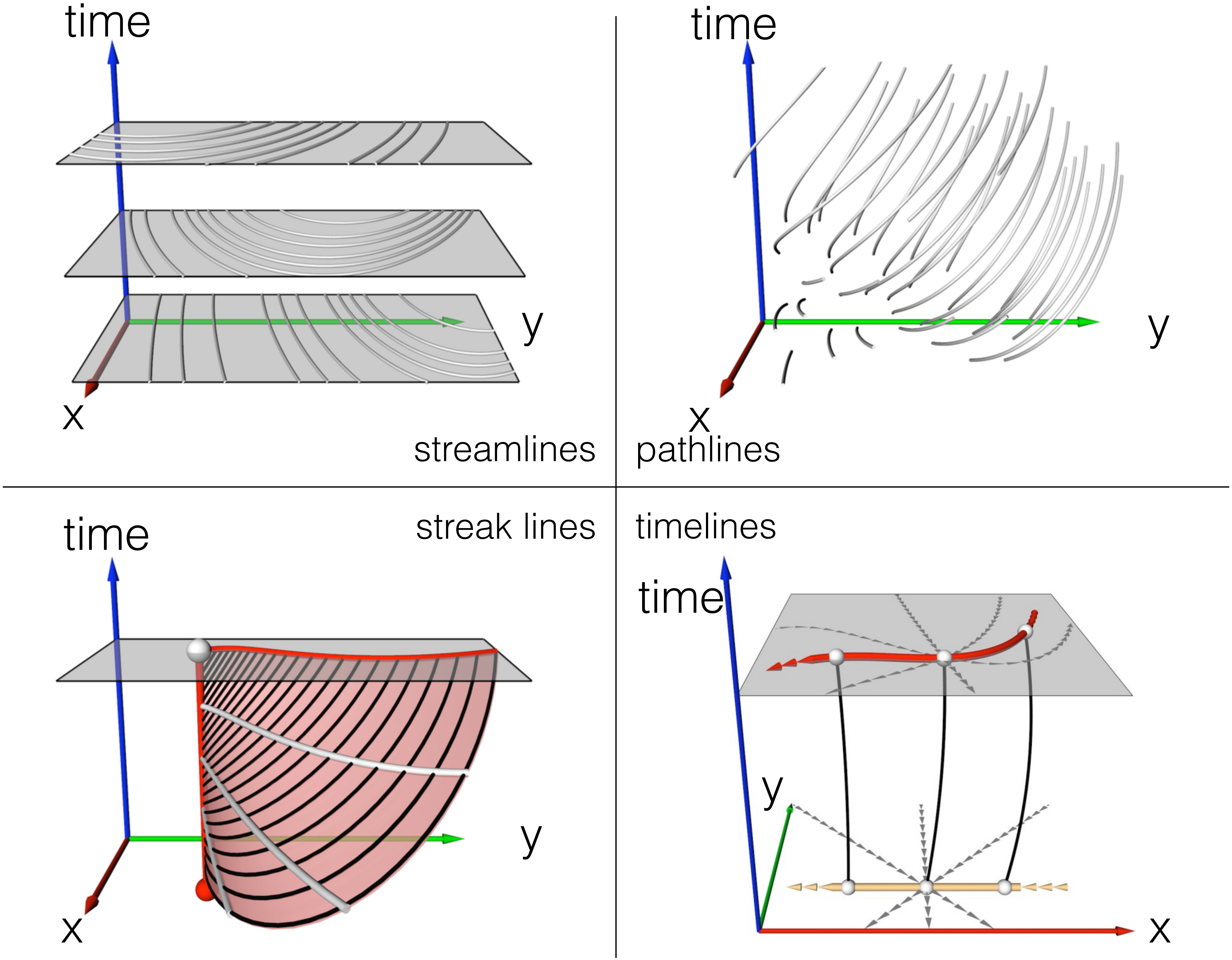
Flow visualization

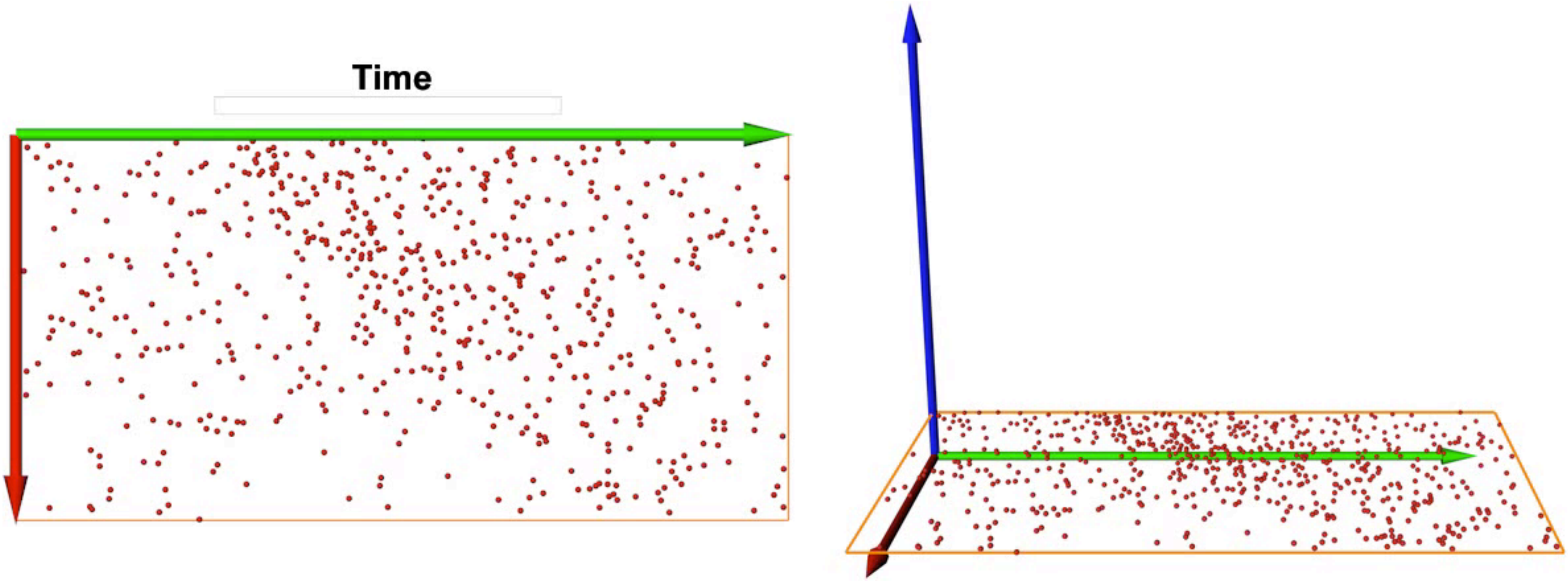
Approaches to flow vis

- "How?"
  - **Characteristic curves of the vector field (streamlines, pathlines, streaklines, timelines, Lagrangian coherent structures / FTLE)**
  - Texture-based (LIC, spot noise)
  - Direct + geometry-based (hedehogs, glyphs)
  - Direct + heuristic (magnitude, Laplacian, FTLE)
  - Physically-based (Schlieren imaging, virtual rheoscopic fluids)
- "Where?"
  - Flow in 2D
  - Flow on surfaces
  - Flow in 3D space

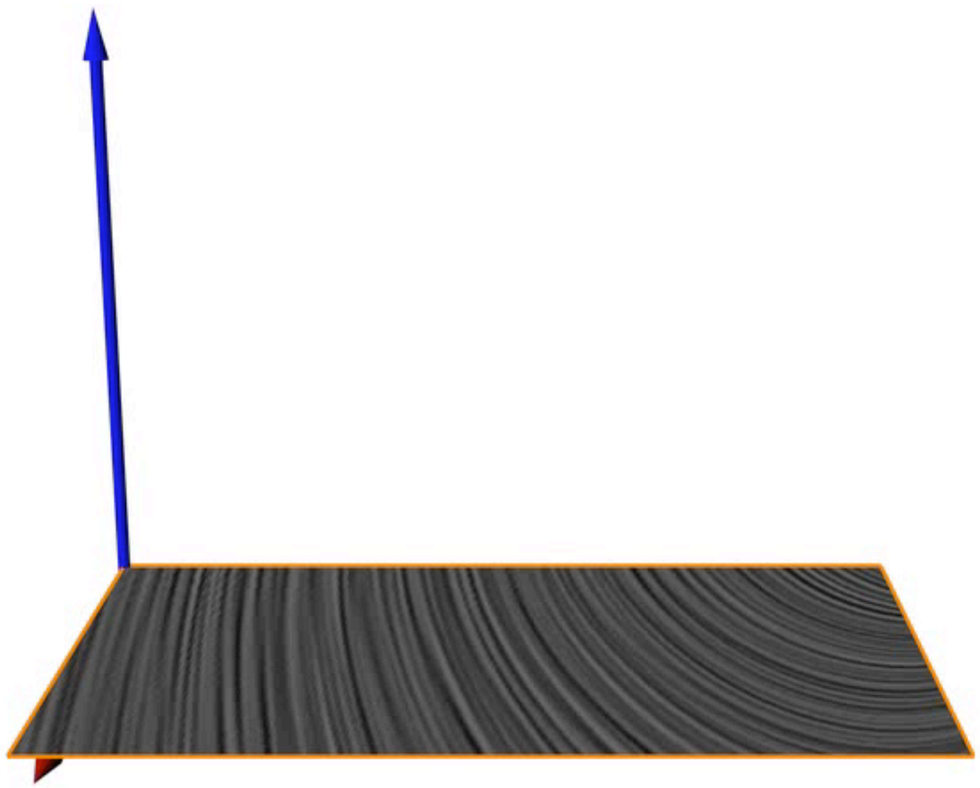
Characteristic Curves of a Vector Field

- **Streamlines:** curve parallel (tangent) to the vector field in each point for a fixed time
- **Pathlines:** describes motion of a particles over time through a vector field
- **Streaklines:** trace of dye that is released into the flow at a fixed position
- **Timelines**: describes motion of particles set out on a line over time through a vector field


time
y
x
streamlines
time
y
x
pathlines
time
streak lines
y
x
timelines
time
y
x



2D time-dependent vector field
particle visualization

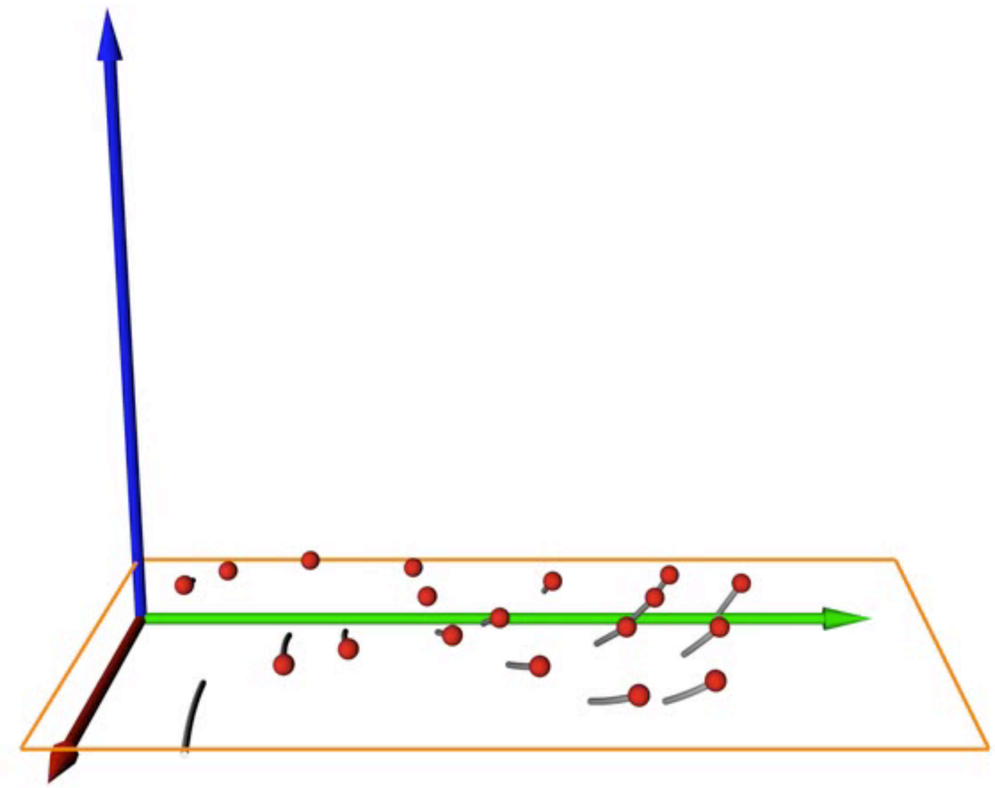


streamlines

curve parallel to the vector field in each point for a **fixed time**

describes motion of a massless particle in an **steady** flow field

$$\mathbf{s}(t) = \mathbf{s}_0 + \int_{0 \le u \le t} V(\mathbf{s}(u))\, du$$

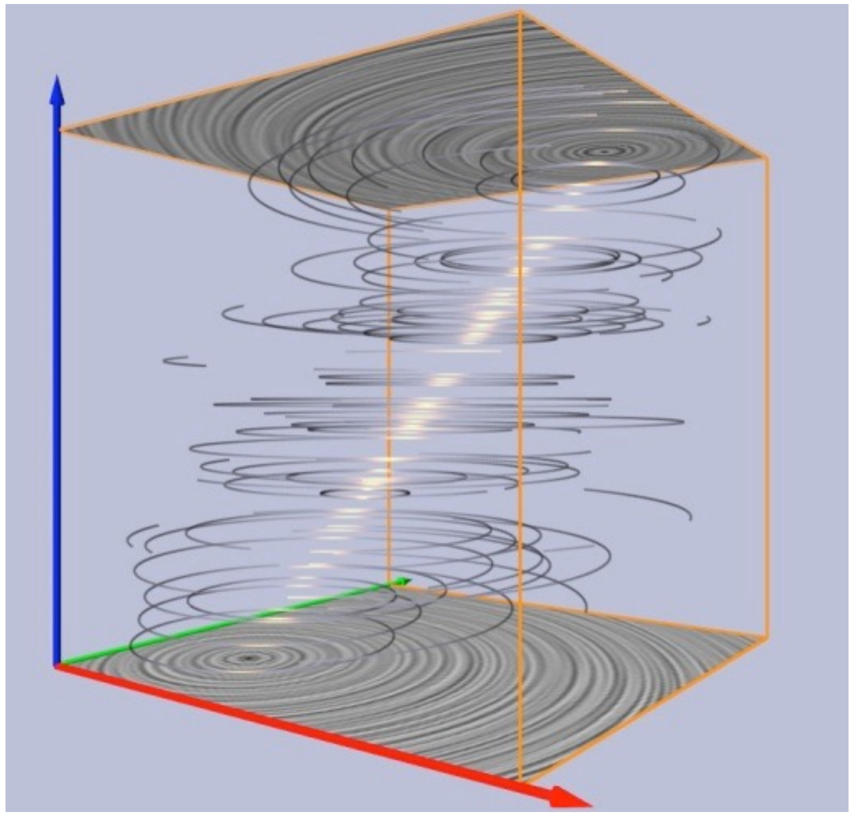


pathlines

curve parallel to the vector field in each point **over time**

describes motion of a massless particle in an **unsteady** flow field

$$\mathbf{s}(t) = \mathbf{s}_0 + \int_{0 \le u \le t} V(\mathbf{s}(u), \mathbf{u})\, du$$

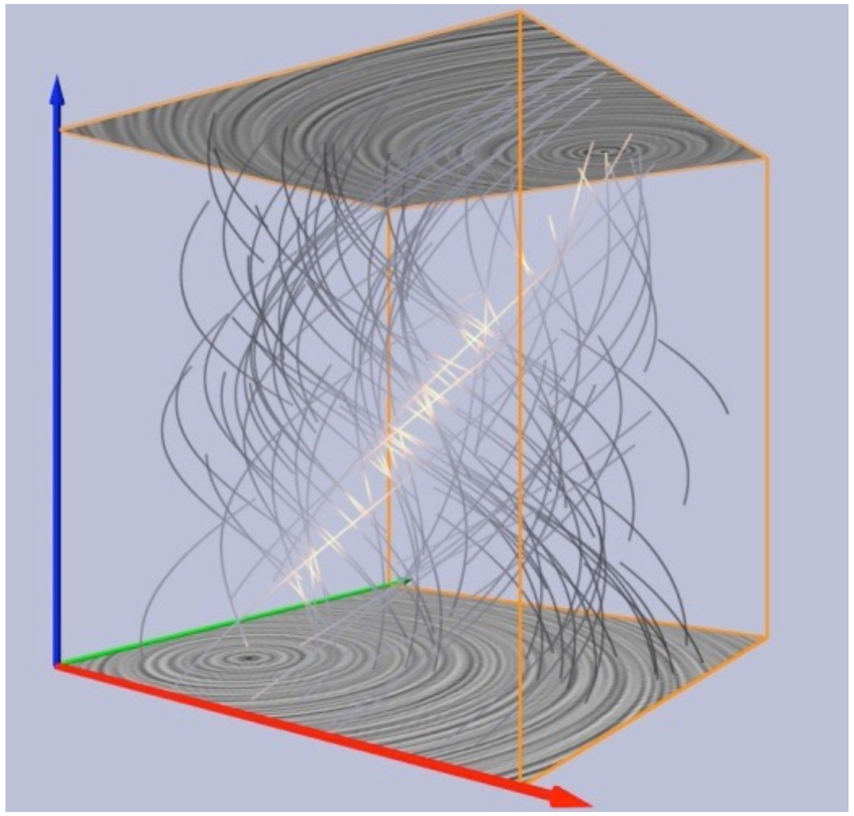


streamlines

curve parallel to the vector field in each point for a **fixed time**

describes motion of a massless particle in an **steady** flow field

$$\boldsymbol{s}(t) = \boldsymbol{s}_0 + \int_{0 \le u \le t} V(\boldsymbol{s}(u))\, du$$



pathlines

curve parallel to the vector field in each point **over time**

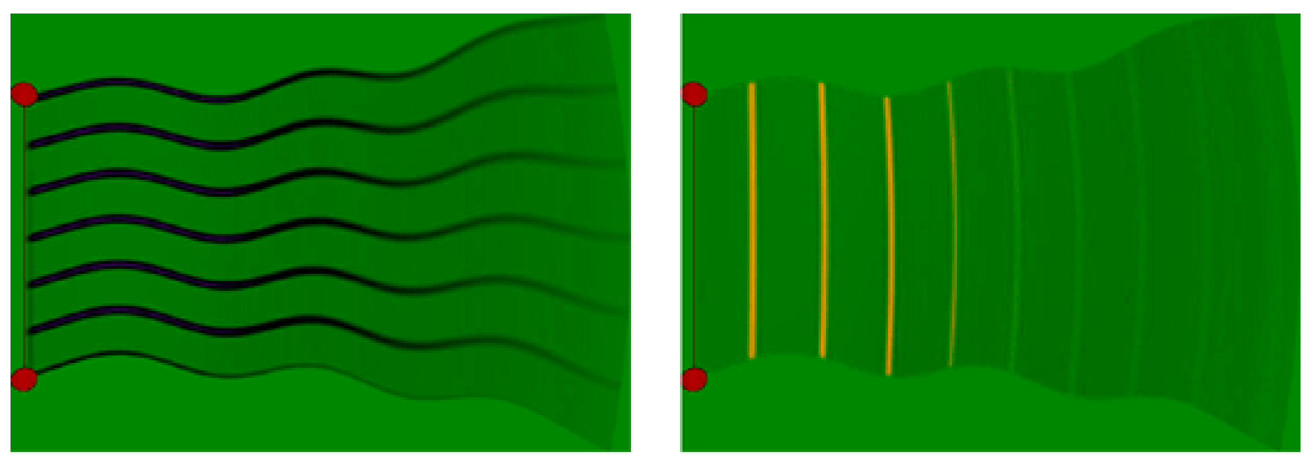
describes motion of a massless particle in an **unsteady** flow field

$$\boldsymbol{s}(t) = \boldsymbol{s}_0 + \int_{0 \le u \le t} V(\boldsymbol{s}(u), \boldsymbol{u})\, du$$

Weinkauf and Theisel, TVCG 2010

Other feature curve

- **Timelines**
  - Union of the current positions of particles released at the same time in space



(a) Coloring fixed rows in the array reveals streak lines.

(b) Coloring fixed columns in the array reveals time lines.

Source: doi.ieeecomputersociety.org

- **Stream and Path lines:**
  - Through all non-critical points (**x**,t) in space-time there is exactly one stream/path line passing through it.

- **Streak and Time lines:**
  - Many streak/time lines through every point (of the spatial domain)
  - ➔ makes it difficult to describe streak/time lines as tangent curves of some vector field
    - But it is possible. We may discuss it in a later session.

- Stream, Path, and Streak lines $\boxed{\quad\quad}$ in a steady vector field.

?

- **Stream and Path lines:**
  - Through all non-critical points (**x**,t) in space-time there is exactly one stream/path line passing through it.

- **Streak and Time lines:**
  - Many streak/time lines through every point (of the spatial domain)
  - ➔ makes it difficult to describe streak/time lines as tangent curves of some vector field
    - But it is possible. We may discuss it in a later session.

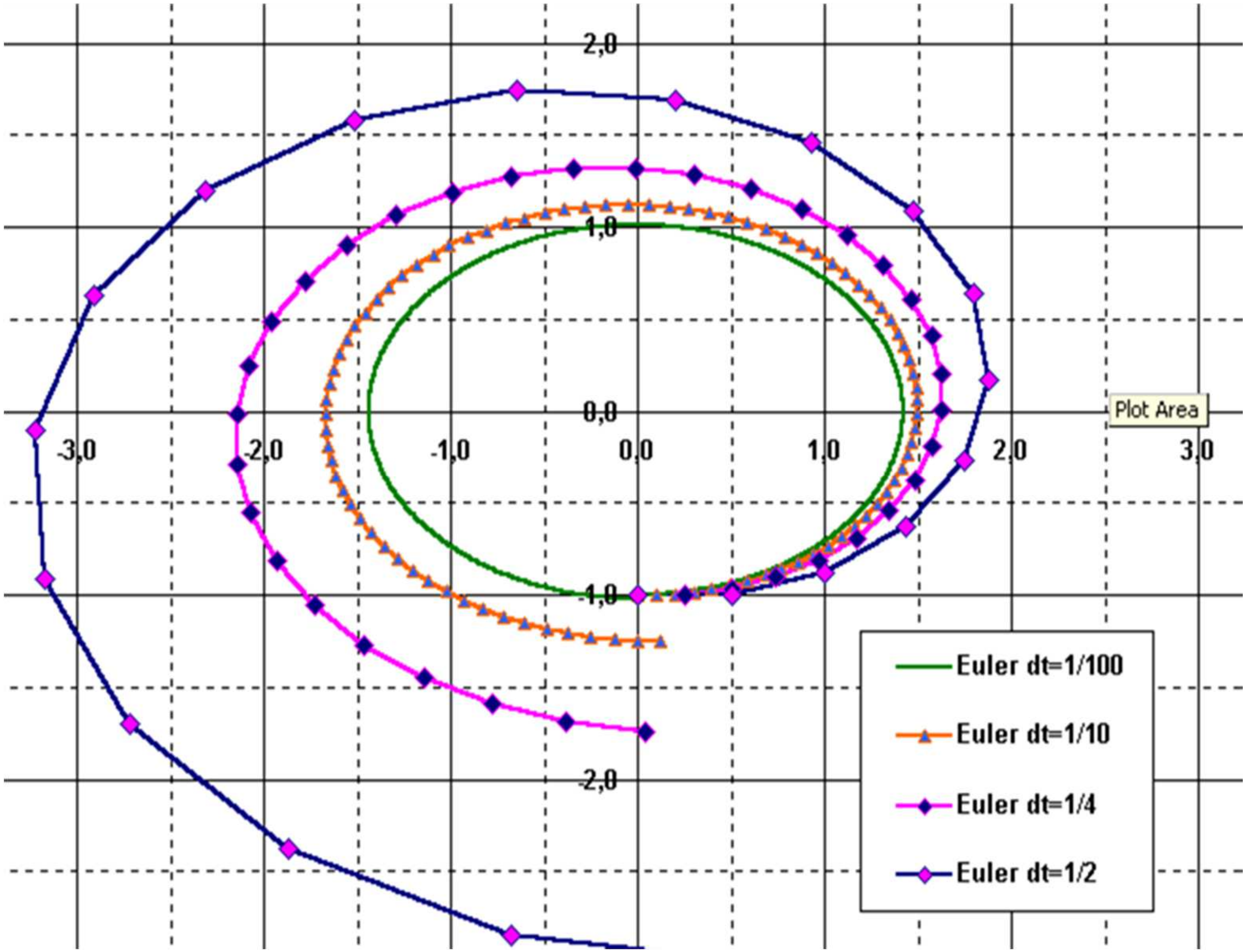
- Stream, Path, and Streak lines coincide in a steady vector field.

Integration Techniques

Comparison Euler, Step Sizes

Euler

quality is proportional to d*t*

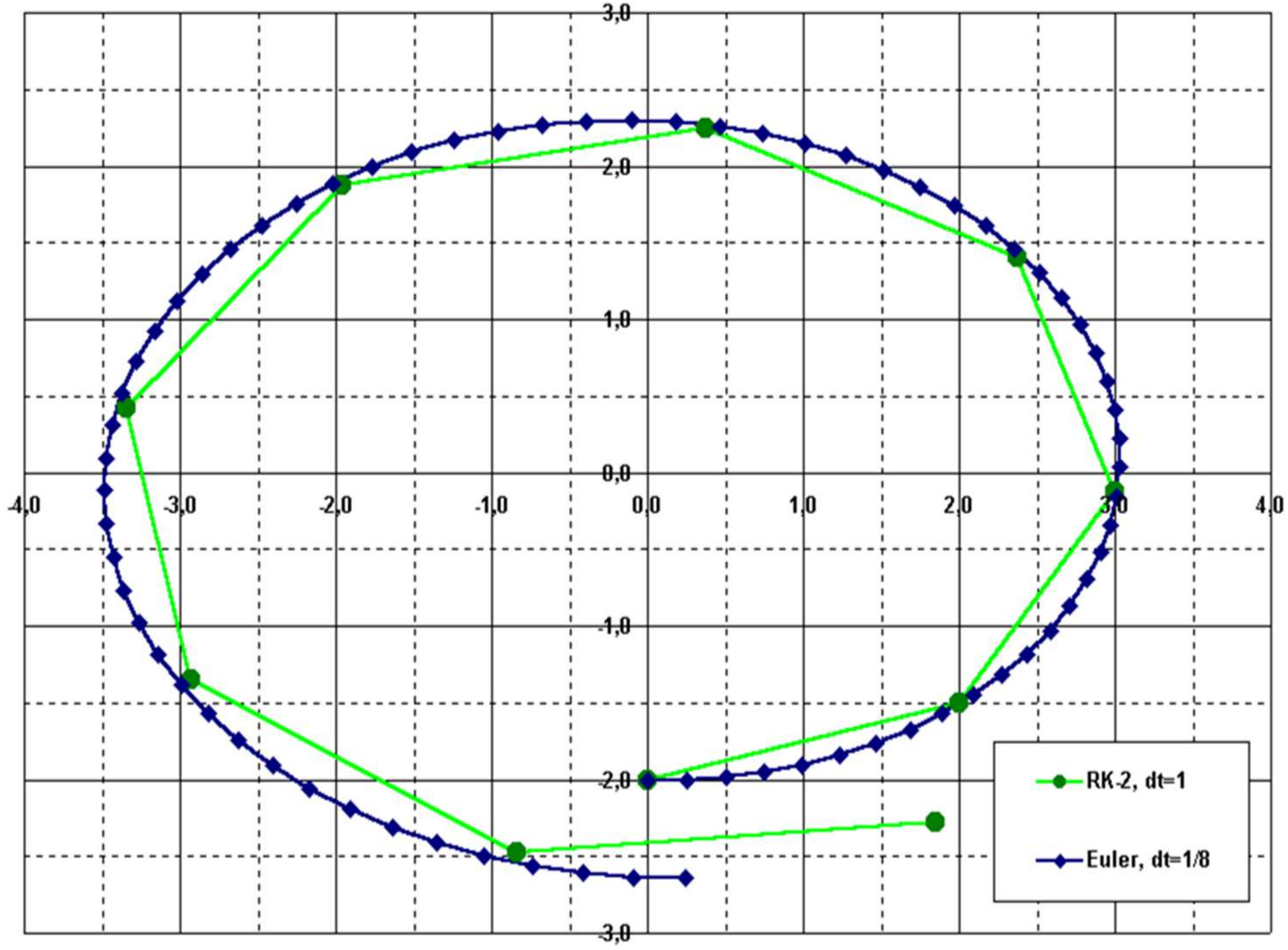


Euler Example – Error Table

dt	#steps	error
1/2	19	~200%
1/4	36	~75%
1/10	89	~25%
1/100	889	~2% ✓
1/1000	8889	~0.2%

RK-2 – A Quick Round

RK-2: even with dt = 1 (9 steps)
better
than Euler
with dt = 1/8
(72 steps)



RK-4 vs. Euler, RK-2

Even better: fourth order RK:

- four vectors **a**, **b**, **c**, **d**
- one step is a convex combination:
  $\mathbf{s}_{i+1} = \mathbf{s}_i + (\mathbf{a} + 2\cdot\mathbf{b} + 2\cdot\mathbf{c} + \mathbf{d})/6$
- vectors:

  $\mathbf{a} = \mathrm{d}t\cdot\mathbf{v}(\mathbf{s}_i)$ … original vector

  $\mathbf{b} = \mathrm{d}t\cdot\mathbf{v}(\mathbf{s}_i+\mathbf{a}/2)$ … RK-2 vector

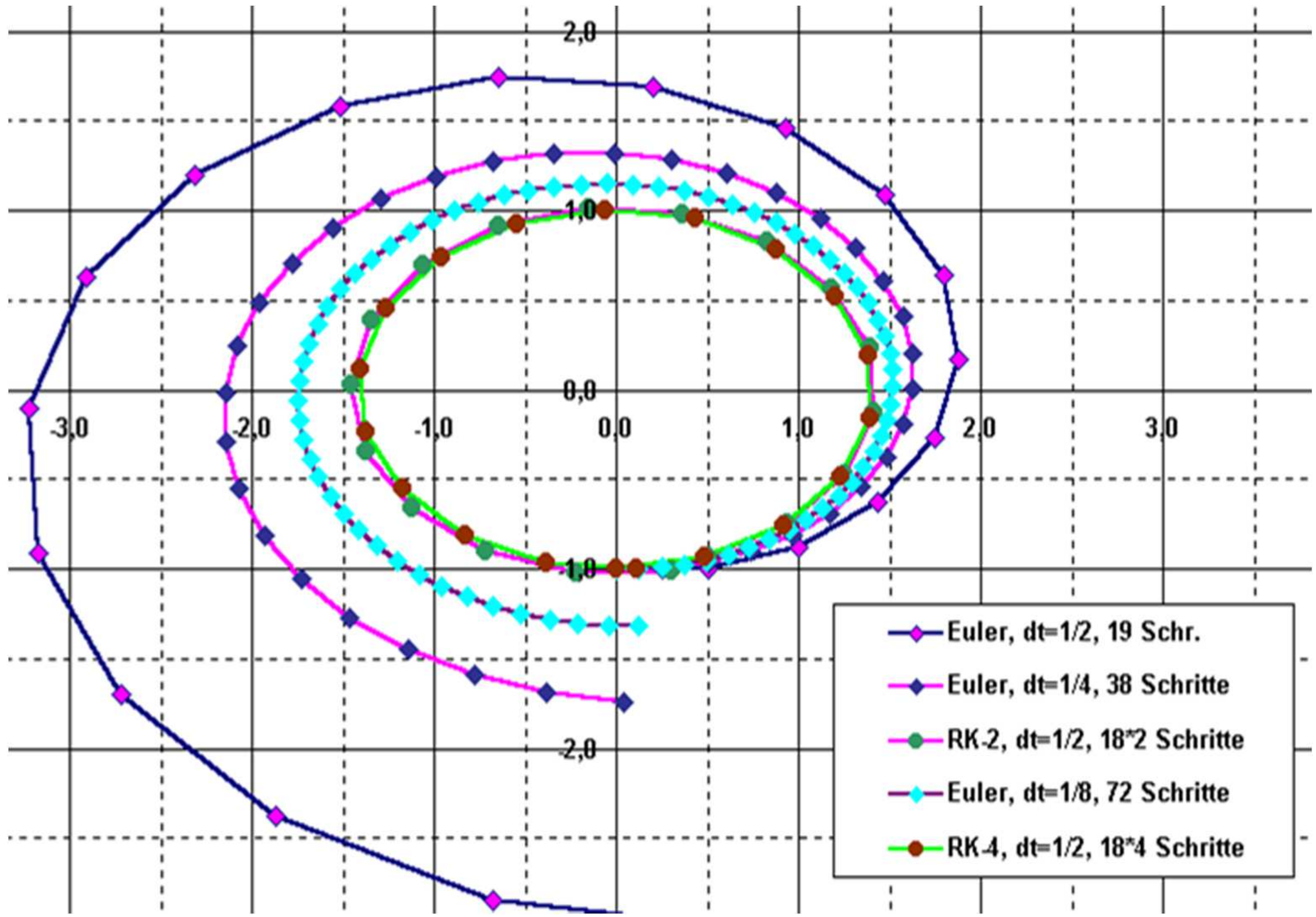
  $\mathbf{c} = \mathrm{d}t\cdot\mathbf{v}(\mathbf{s}_i+\mathbf{b}/2)$ … use RK-2 …

  $\mathbf{d} = \mathrm{d}t\cdot\mathbf{v}(\mathbf{s}_i+\mathbf{c})$ … and again

Euler vs. Runge-Kutta

RK-4: pays off only with complex flows

Here approx. like RK-2



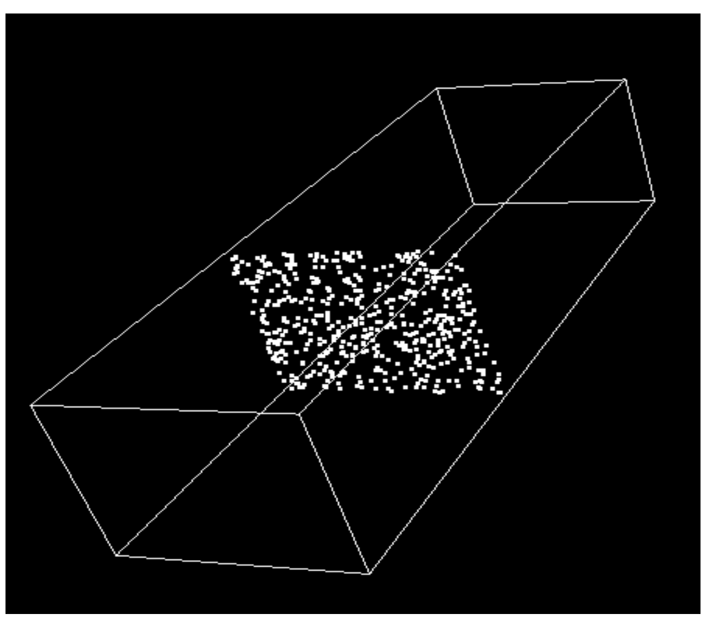
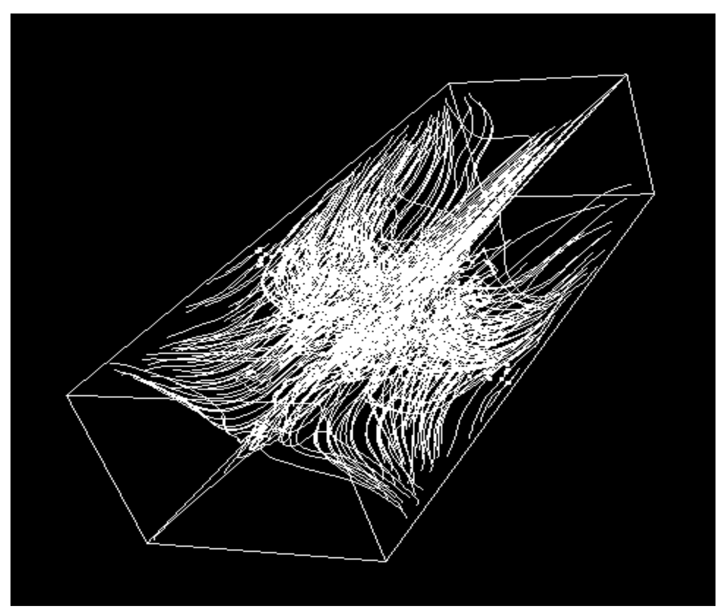
Integration, Conclusions

Summary:

- analytic determination of streamlines usually not possible
- hence: numerical integration
- various methods available
  (Euler, Runge-Kutta, etc.)
- Euler: simple, imprecise, esp. with small d*t*
- RK: more accurate in higher orders
- furthermore: adaptive methods, implicit methods, etc.

Streamline Placement
(in 2D)

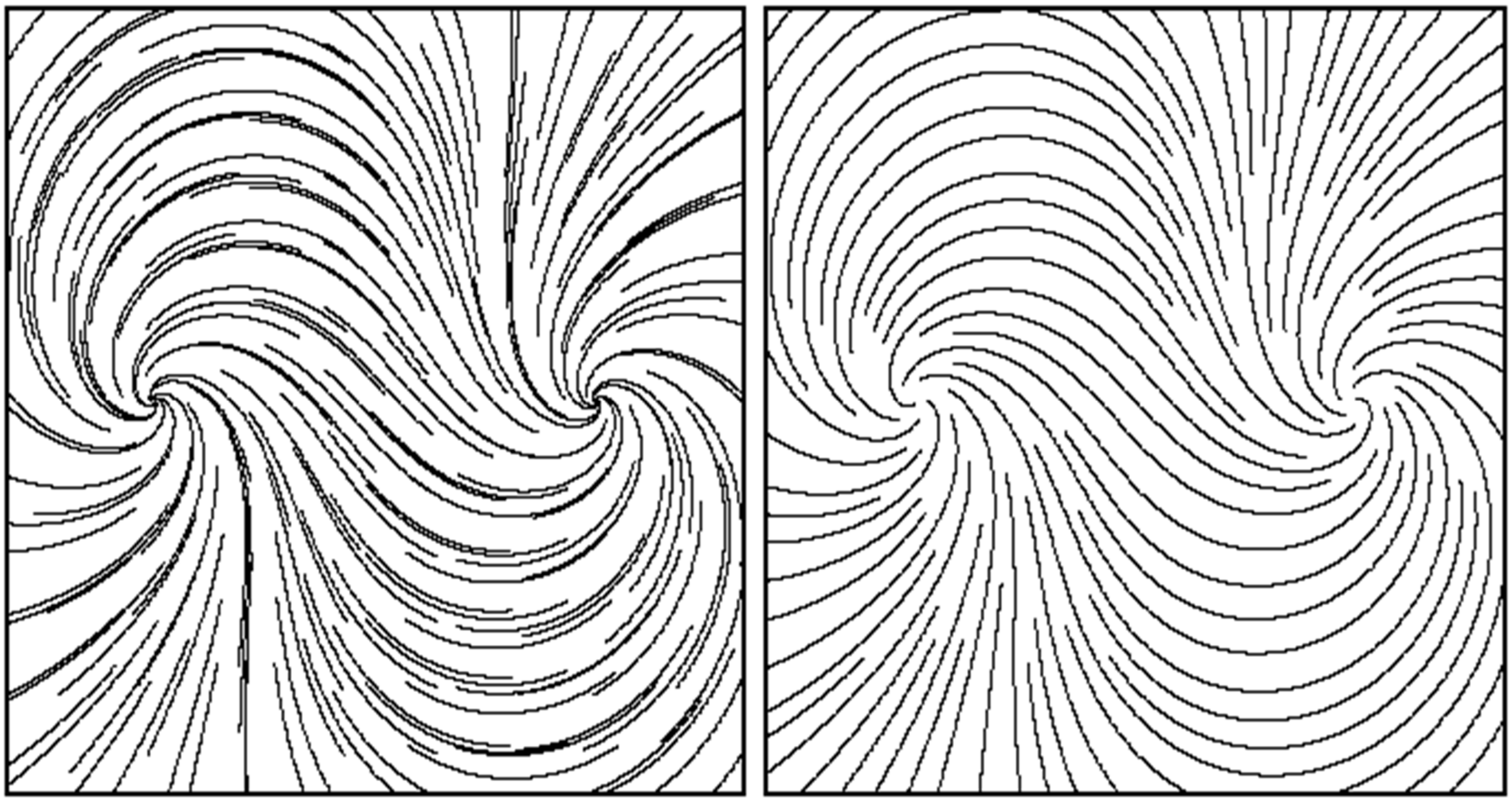
- **Seeding of integral lines:**
- which stream/path/streak/time lines to visualize?
- too few: important details get lost
- too many: overload, visual clutter
- simple approaches:
  - start on regular grid points
  - start randomly
- It has to be the right number at the right places!!!

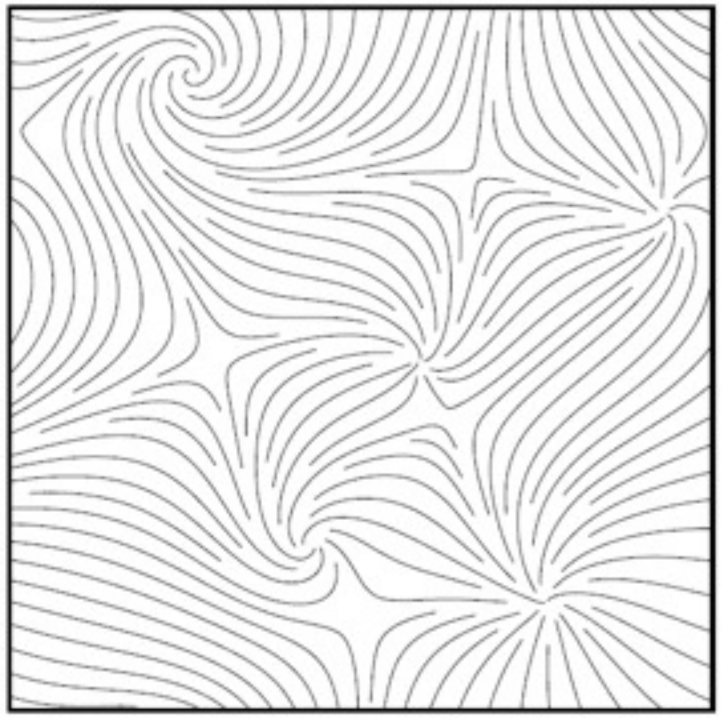



Problem: Choice of Seed Points

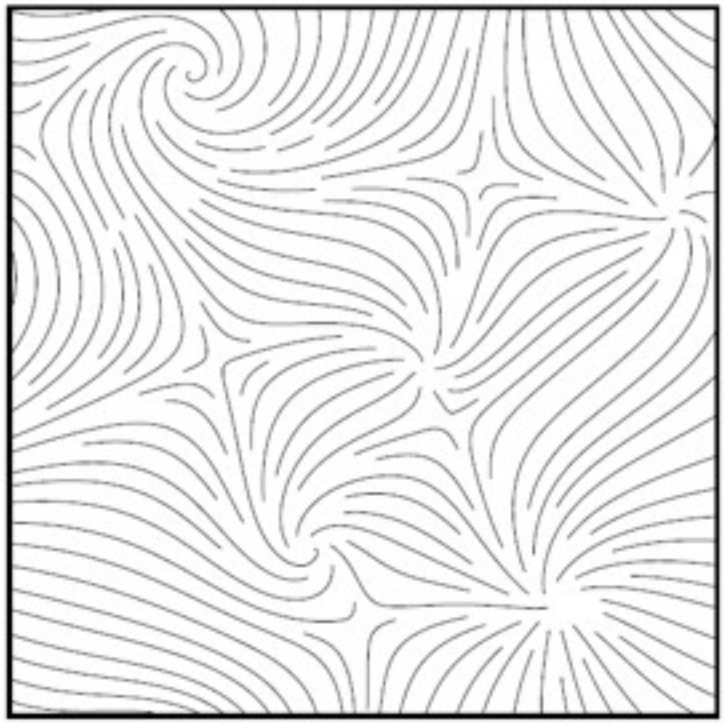
Streamline placement:

- If regular grid used: very irregular result



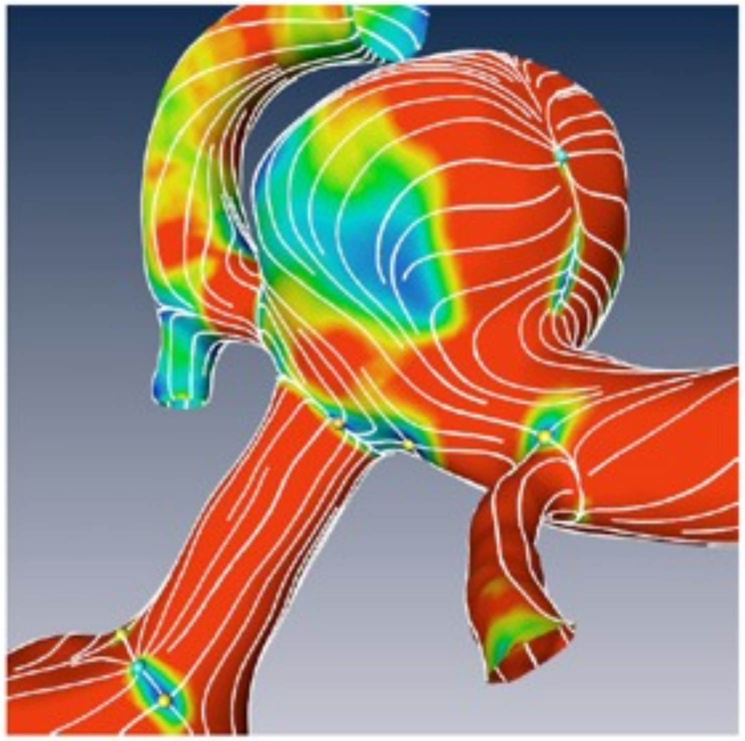


Turk and Banks, 1996



Jobard et al., 1997

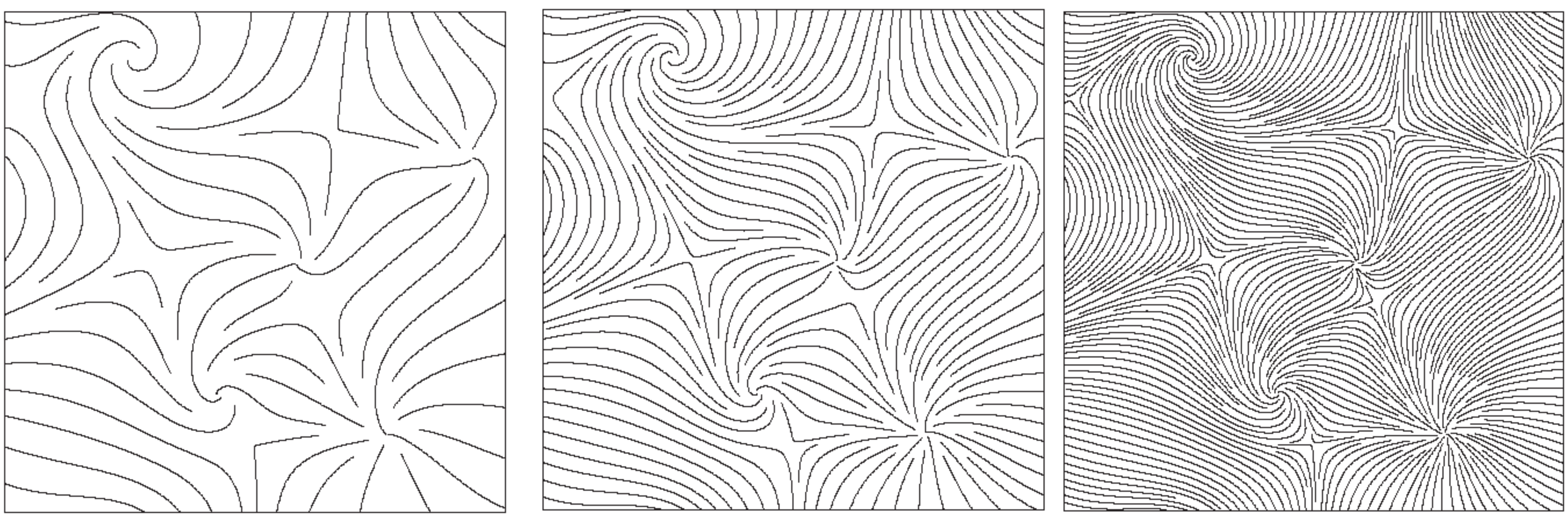
Mebarki et al., 2005



Rosanwo et al., 2009

Streamline seeding

- 2D: evenly spaced stream lines
- Turk/Banks 96:
    - Start with “streamlets” (very short stream lines)
    - Apply a series of energy-decreasing elementary operations: combine, delete, create, lengthen, shorten streamlets
    - Energy: difference between low-pass filtered version of current placements and uniform grey image



Main idea: the distribution of ink on the screen should be even [Turk and Banks 96]

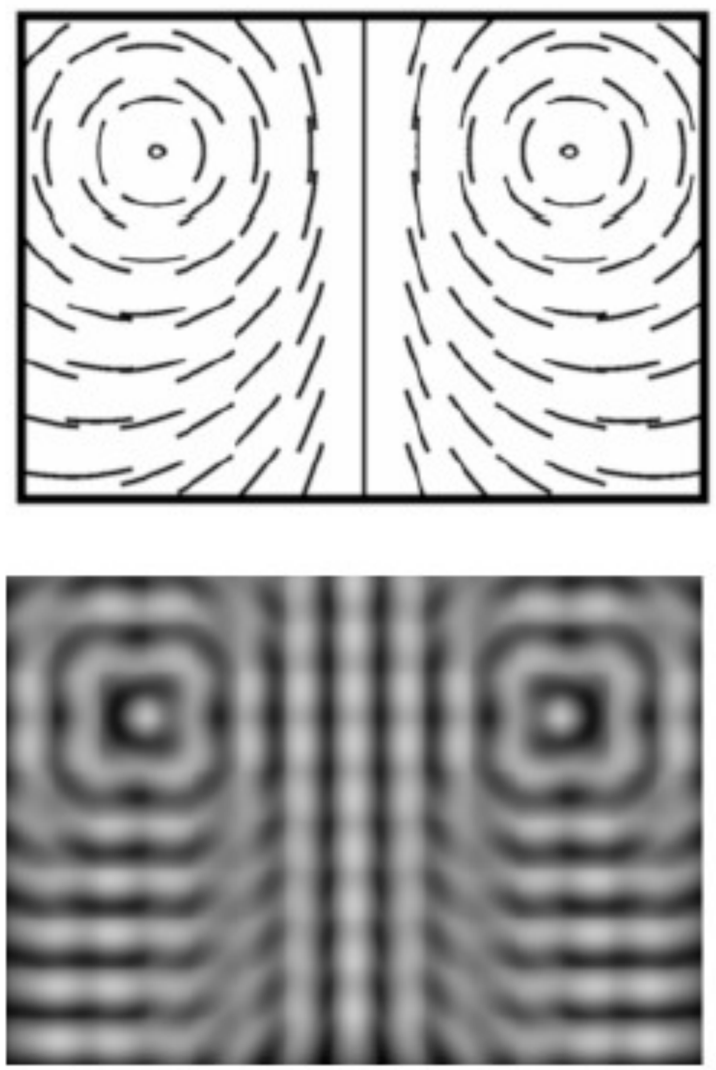


Figure 2: (a) Short streamlines with centers placed on a regular grid (top); (b) filtered version of same (bottom).

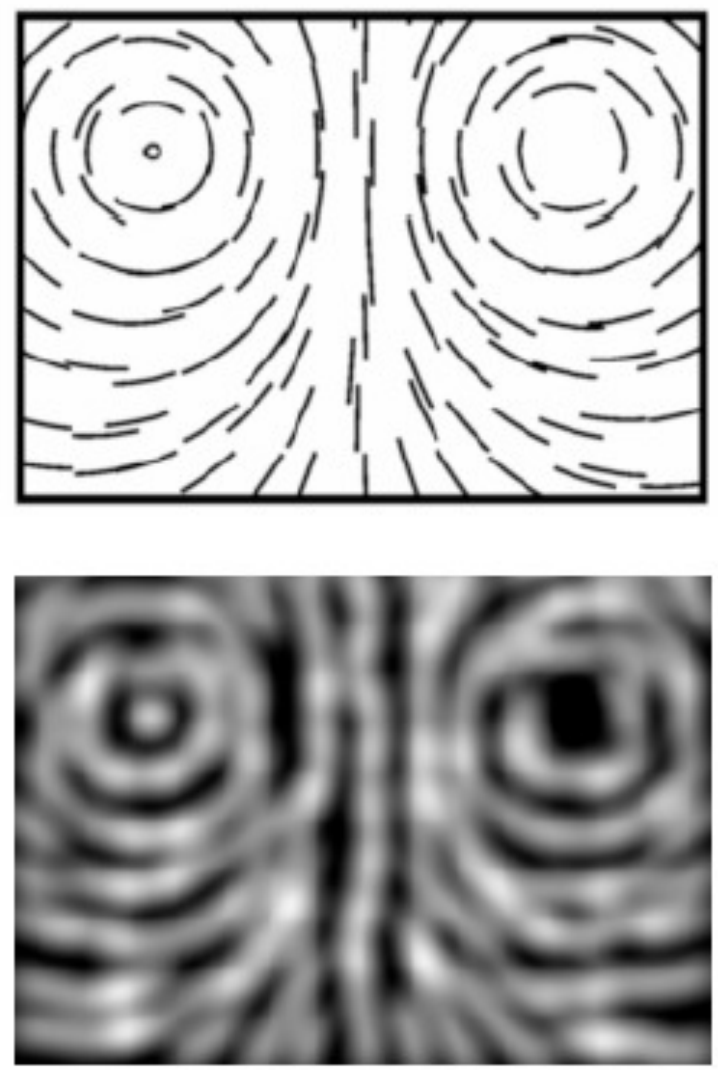


Figure 3: (a) Short streamlines with centers placed on a jittered grid (top); (b) filtered version showing bright and dark regions (bottom).

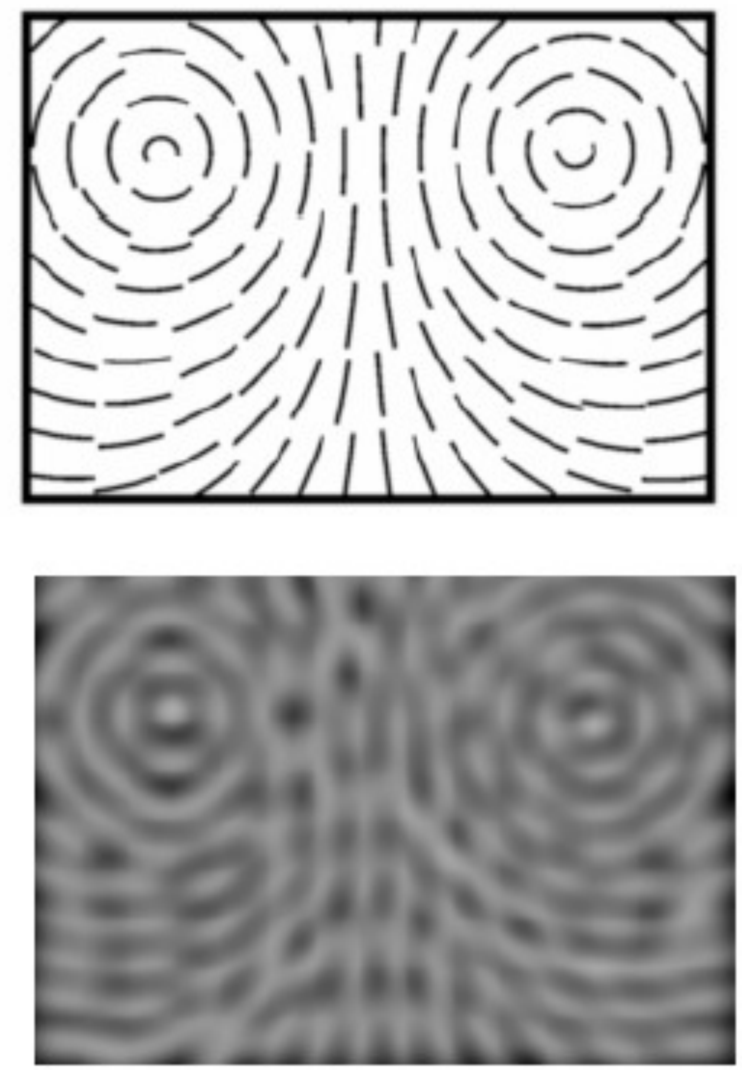
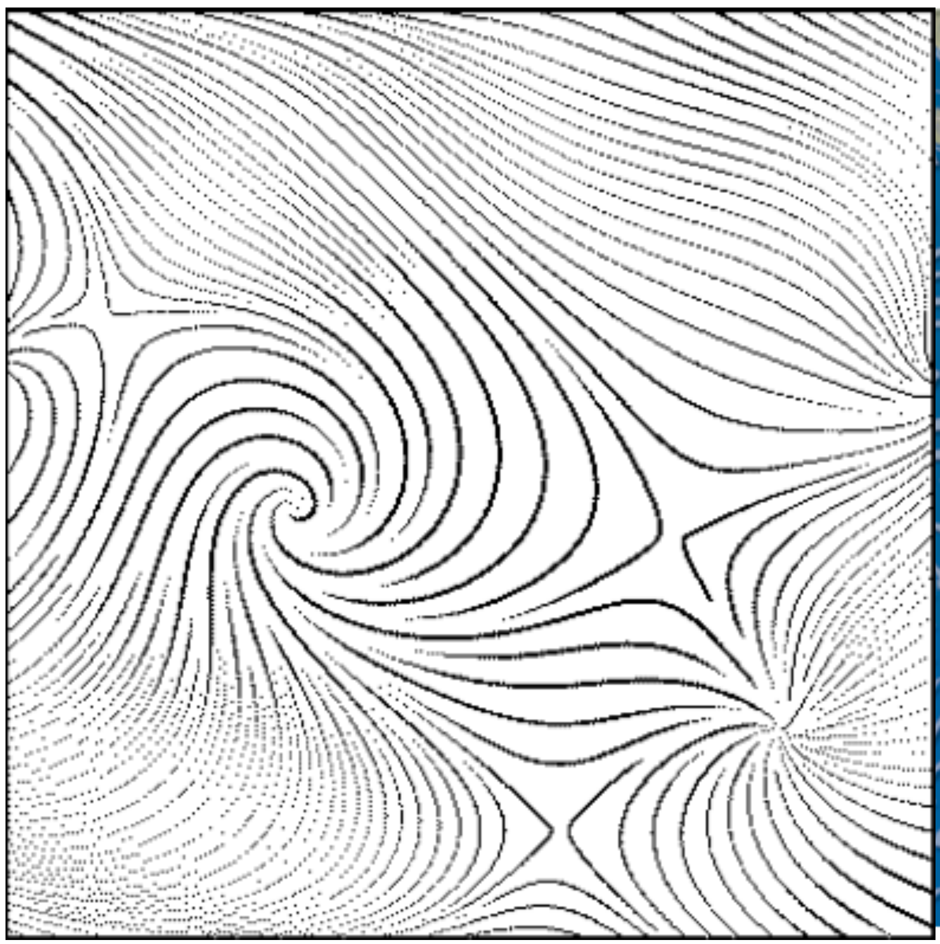


Figure 4: (a) Short streamlines placed by optimization (top); (b) filtered version showing fairly even gray value (bottom).

Results



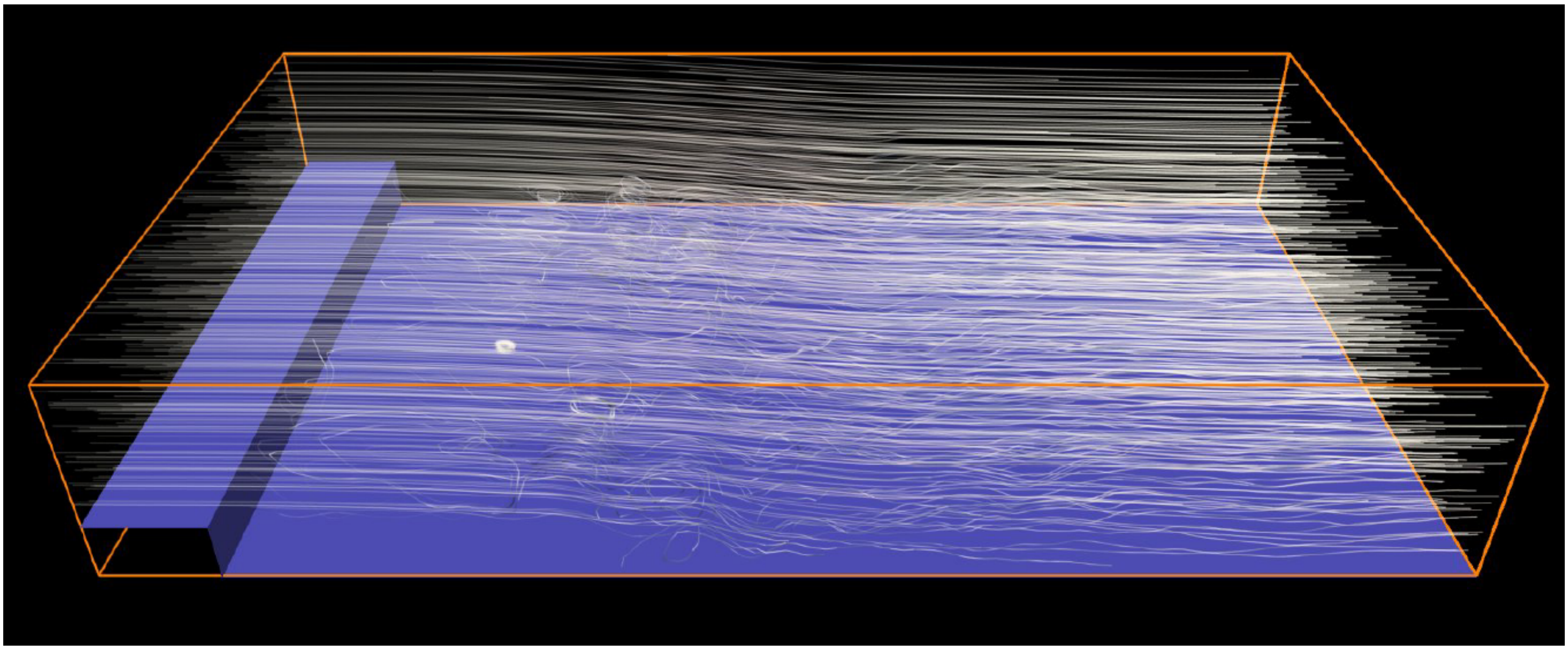
Tapering at streamline ends

Optimized arrow plots

[Turk and Bank '96]

Streamline Seeding in 3D

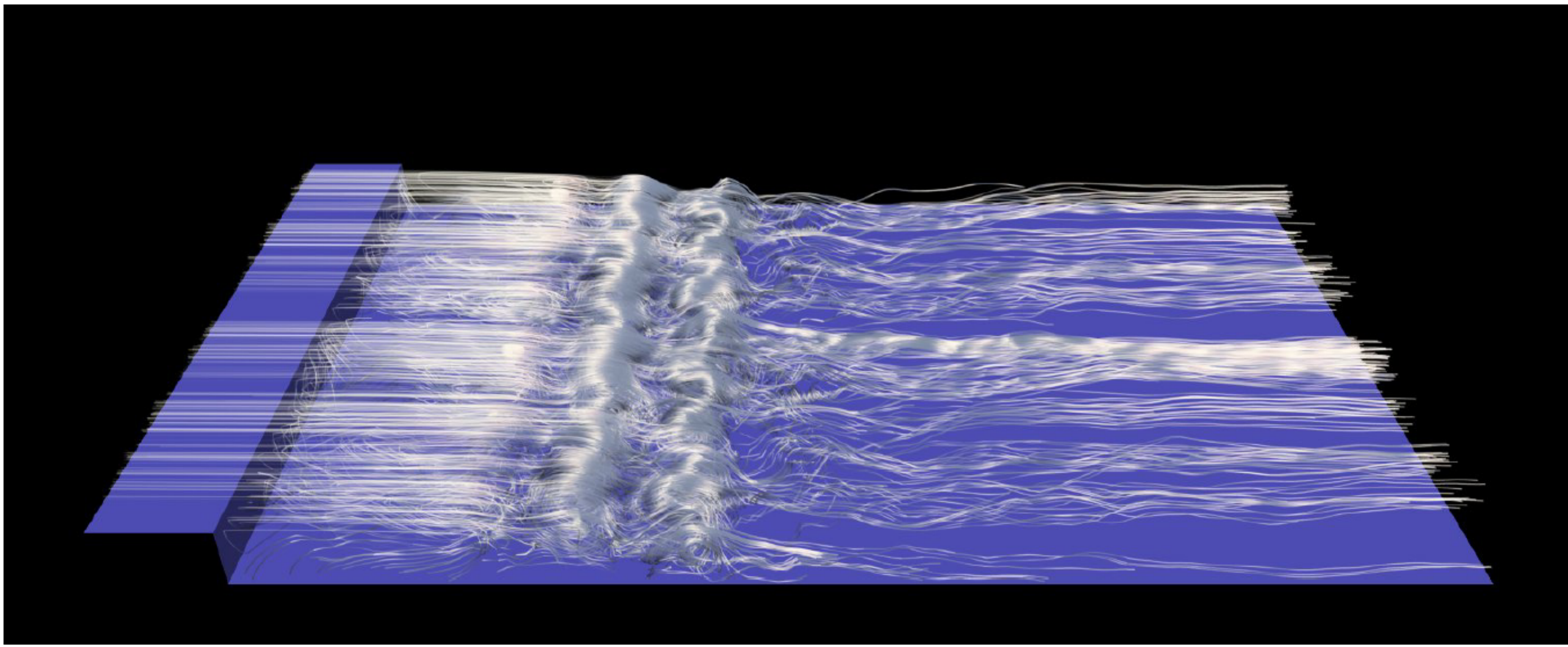
- Evenly-spaced does not make sense
- Start on uniform grid



Weinkauf 2003

Streamline Seeding in 3D

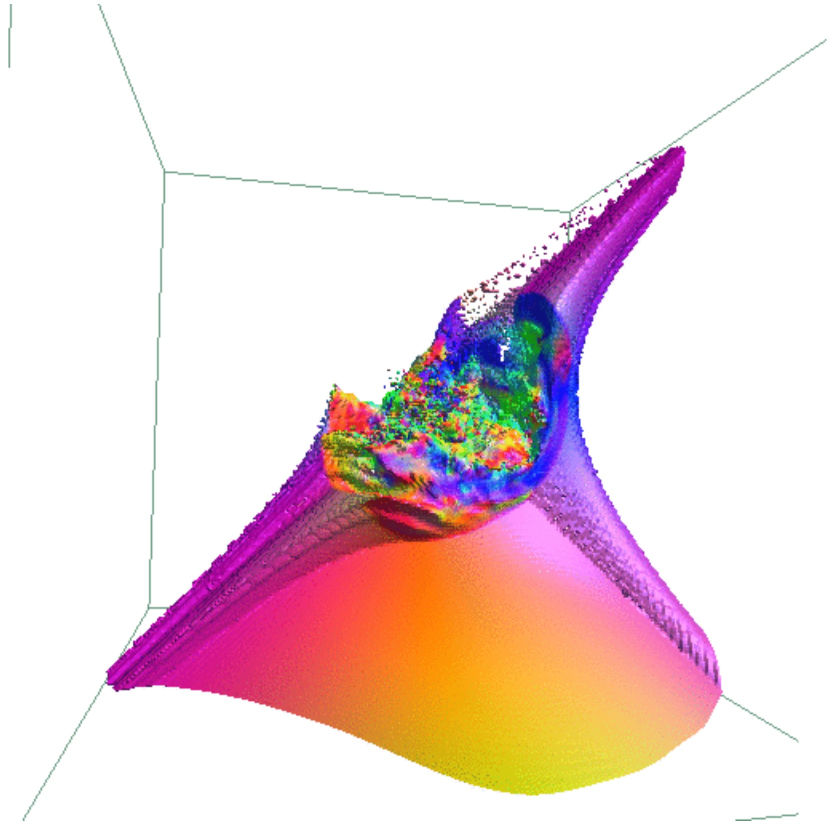
- Evenly-spaced does not make sense
- Start in regions of high vector field curvature (i.e., close to critical points)



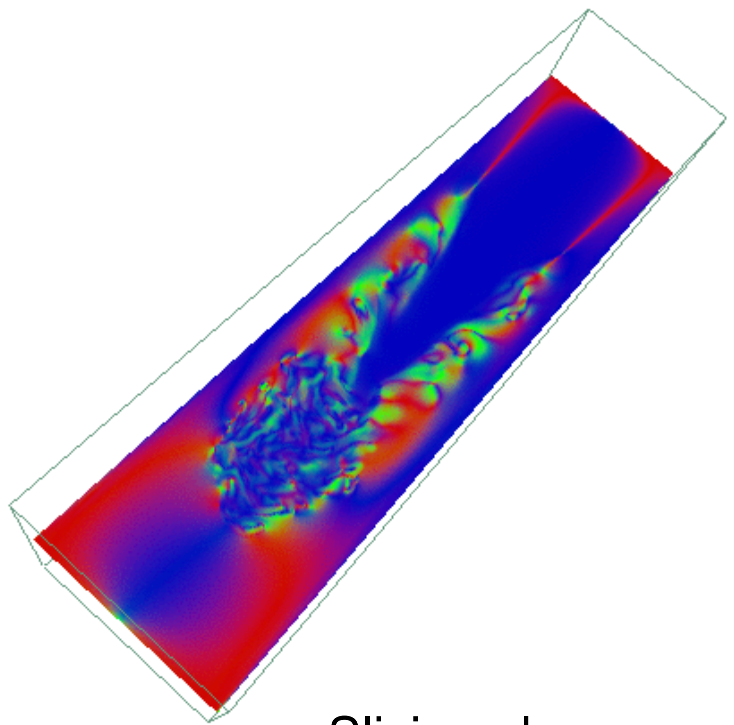
Weinkauf 2003

Seeds in Image Space

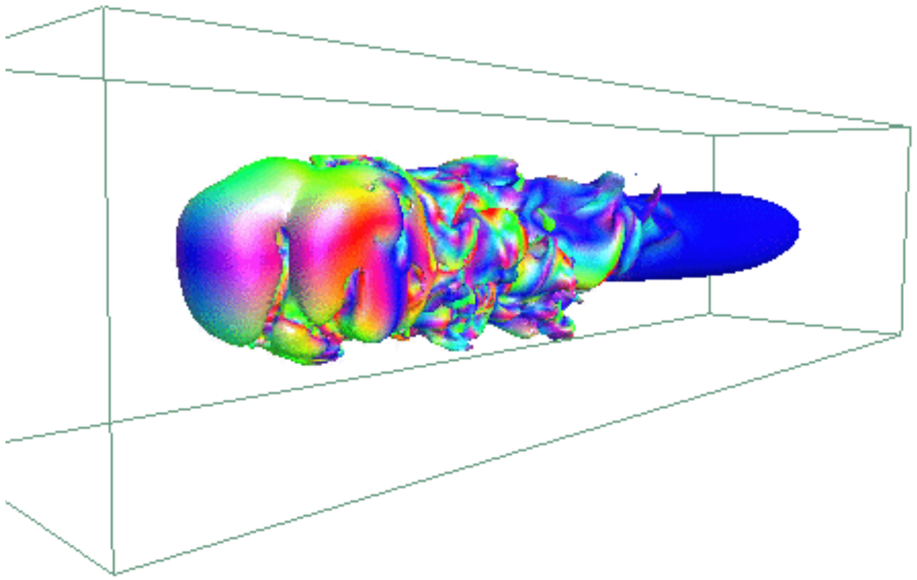
- Need to un-project the seeds back to 3D object space for streamline integration
- Utilize depth maps generated from other visualization techniques



Stream surface

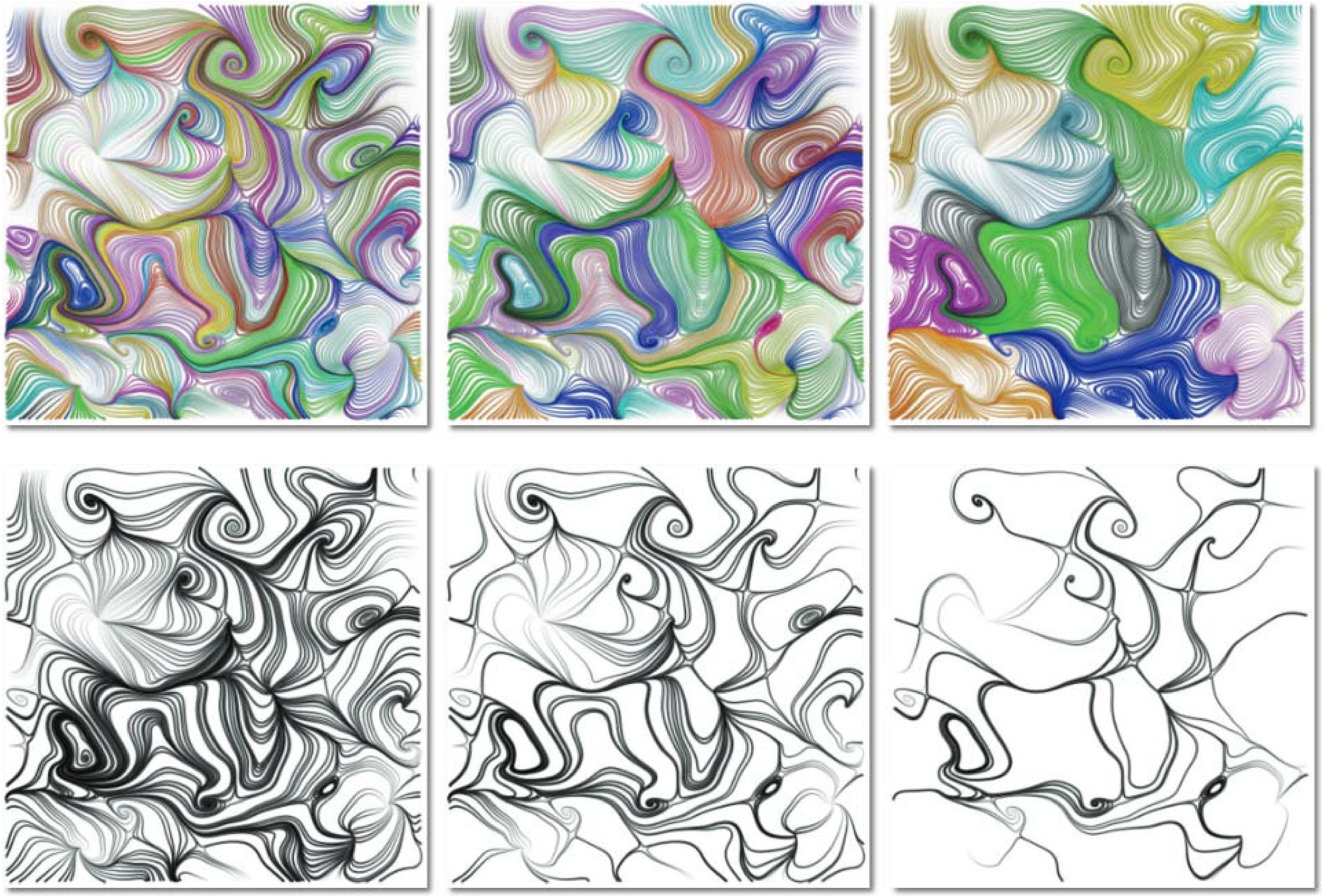


Slicing plane



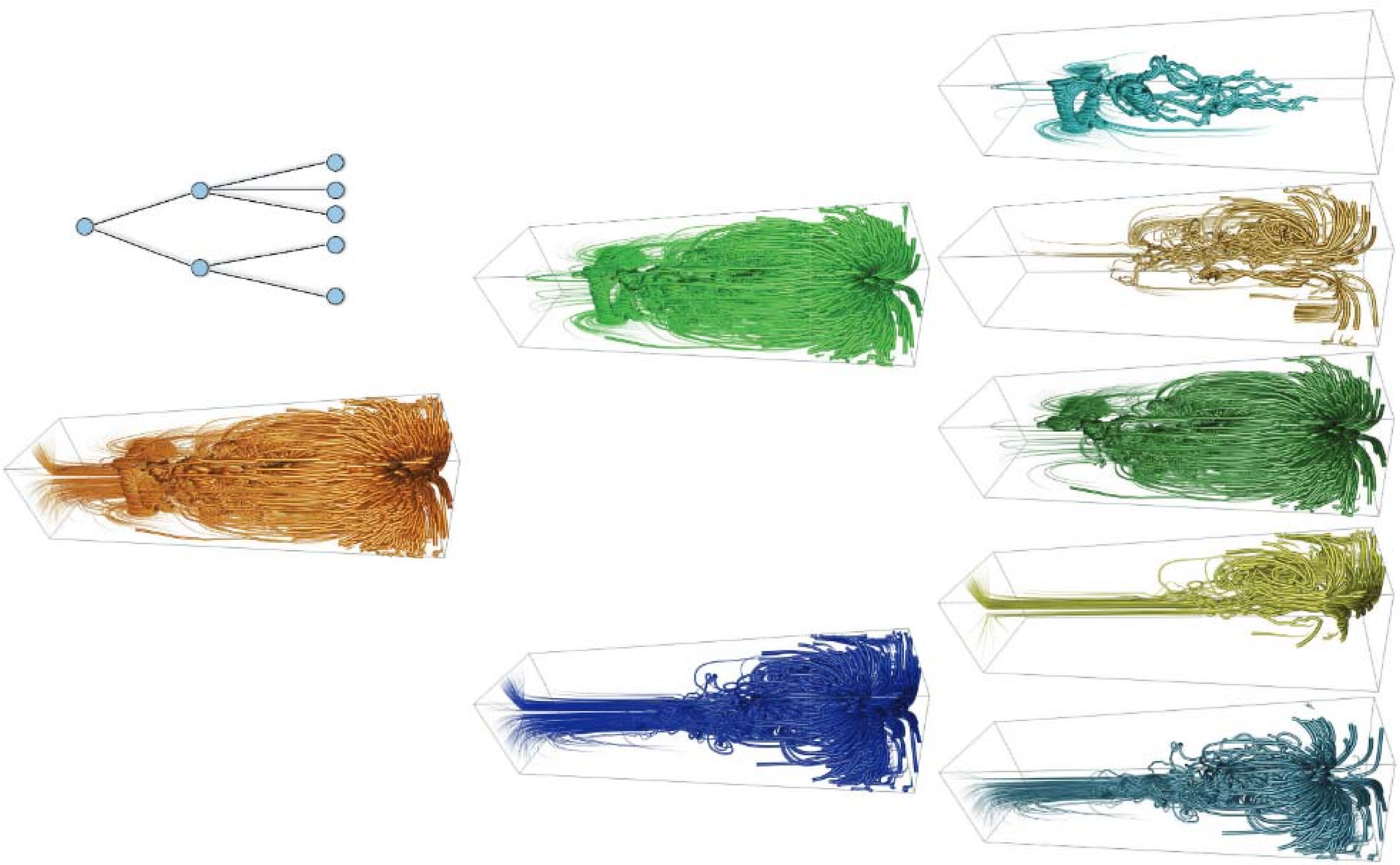
Isosurface

Streamline Bundling



[Yu et al. 2012]

Streamline Bundling



[Yu et al. 2012]

Illuminated Streamlines

Use lighting to improve spatial perception of lines in 3D.

This can to some extend reduce the 3D cluttering issue.

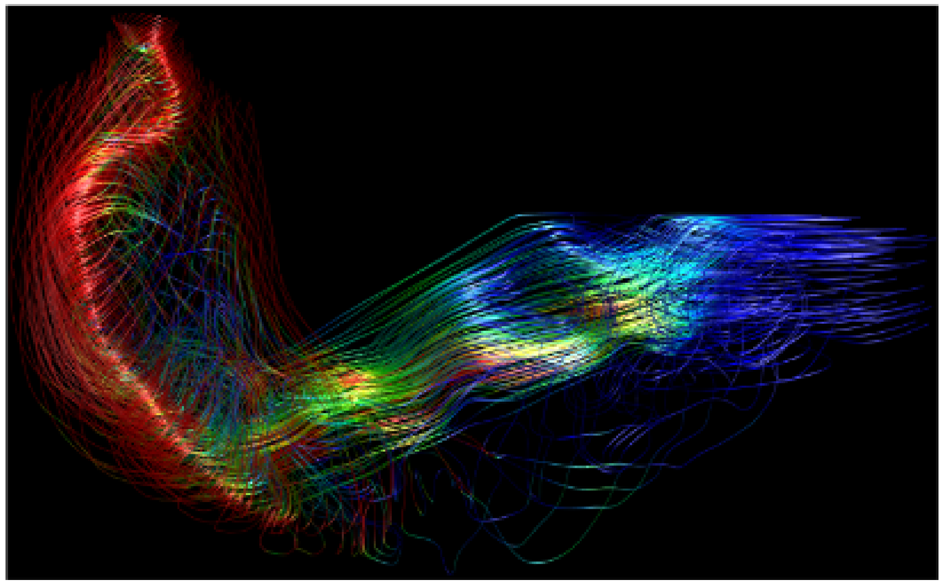
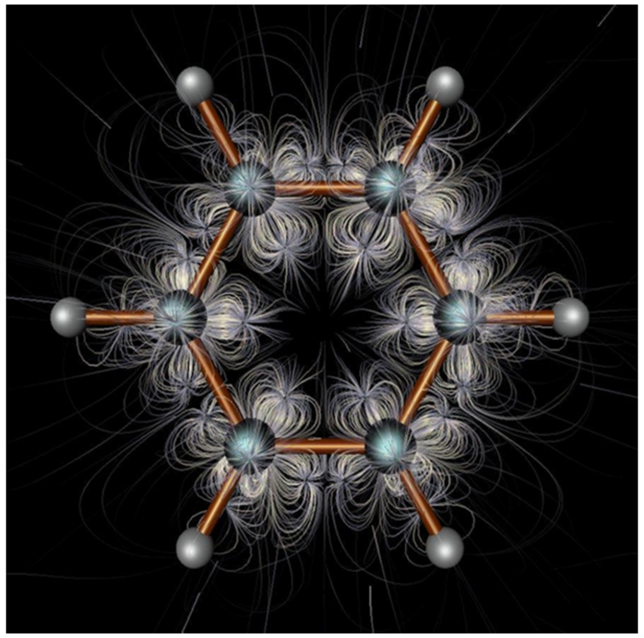


Figure 1: Flow in a Francis draft tube visualized by streamlines regularly seeded on a cone and colored by speed. Streamlines are illuminated based on cylinder averaging. In the vertical part of the tube, a vortex rope is visible.



Open Source: http://www.scivis.ethz.ch/research/projects/illuminated_streamlines

[Zockler et al. 96, Mallo et al. 2005]

Opacity Optimization for 3D Line Fields

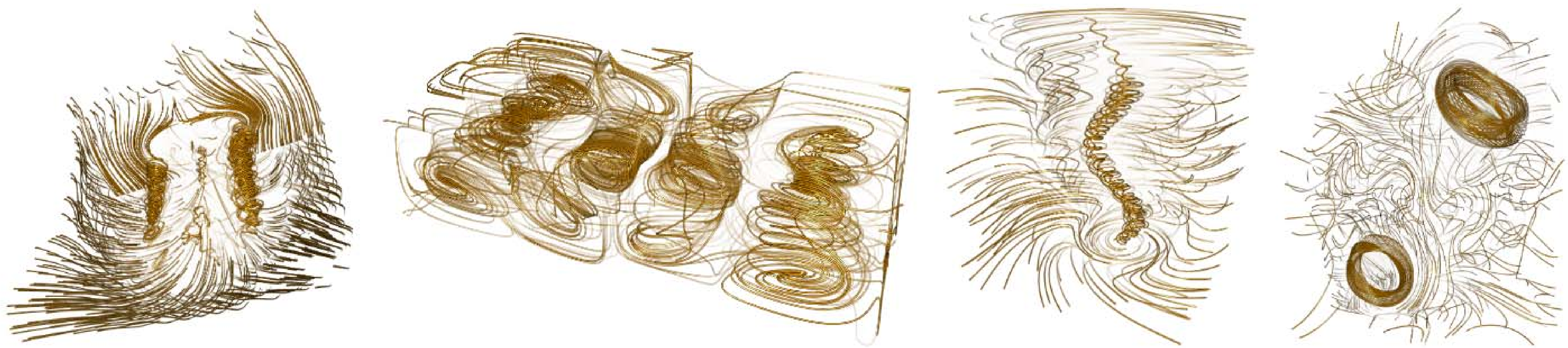


Figure 1: *Applications of our interactive, global line selection algorithm. Our bounded linear optimization for the opacities reveals user-defined important features, e.g., vortices in rotorcraft flow data, convection cells in heating processes (Rayleigh-Bénard cells), the vortex core of a tornado and field lines of decaying magnetic knots (from left to right).*

(a) Given is a set of polylines.

(b) Discretize polylines into n segments (here: $n = 6$).

(c) Compute per-segment opacity α_i by energy minimization.

(d) Interpolate opacities between adjacent segments for final rendering.

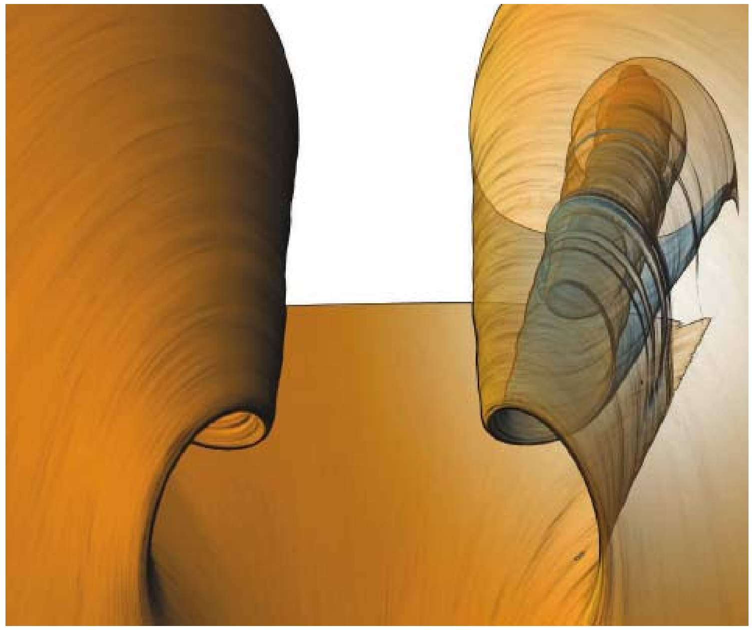
Idea: make less important sections of streamlines transparent to fix occlusion, remove clutter.
Gunthe et al. SIGGRAPH 2013

[Gunthe et al. 2013]

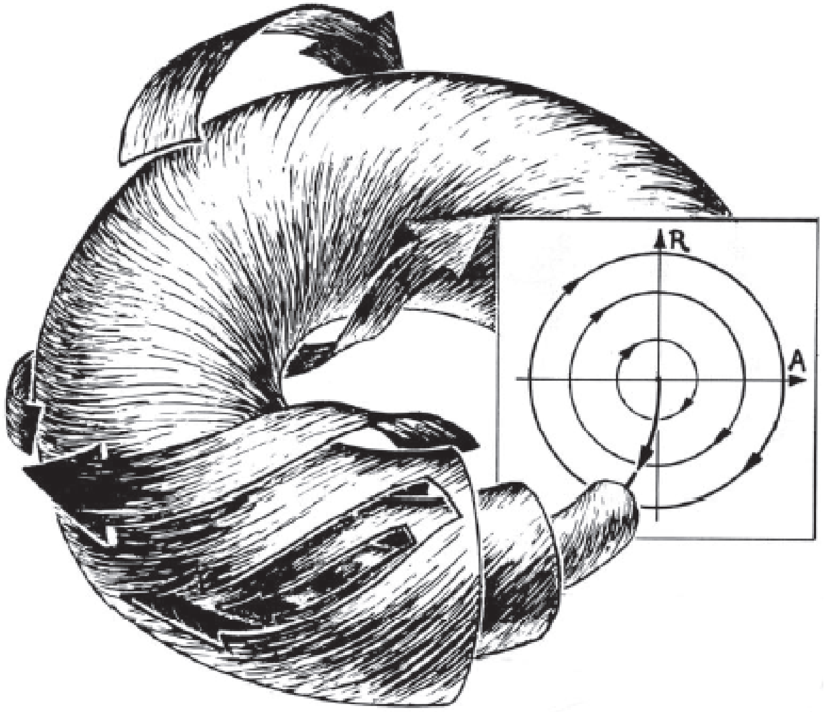
Rendering of stream surfaces

Illustrative visualization

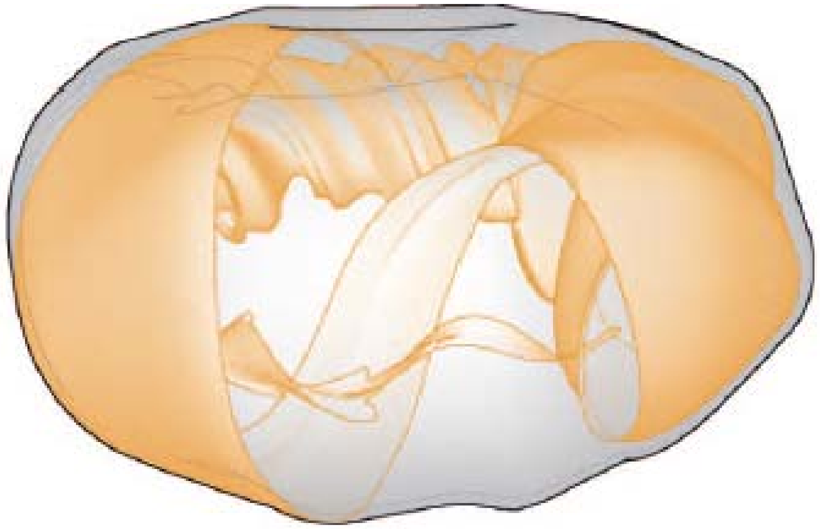
- Using transparency and surface features such as silhouette and feature curves.



[Hummel et al. 2010]

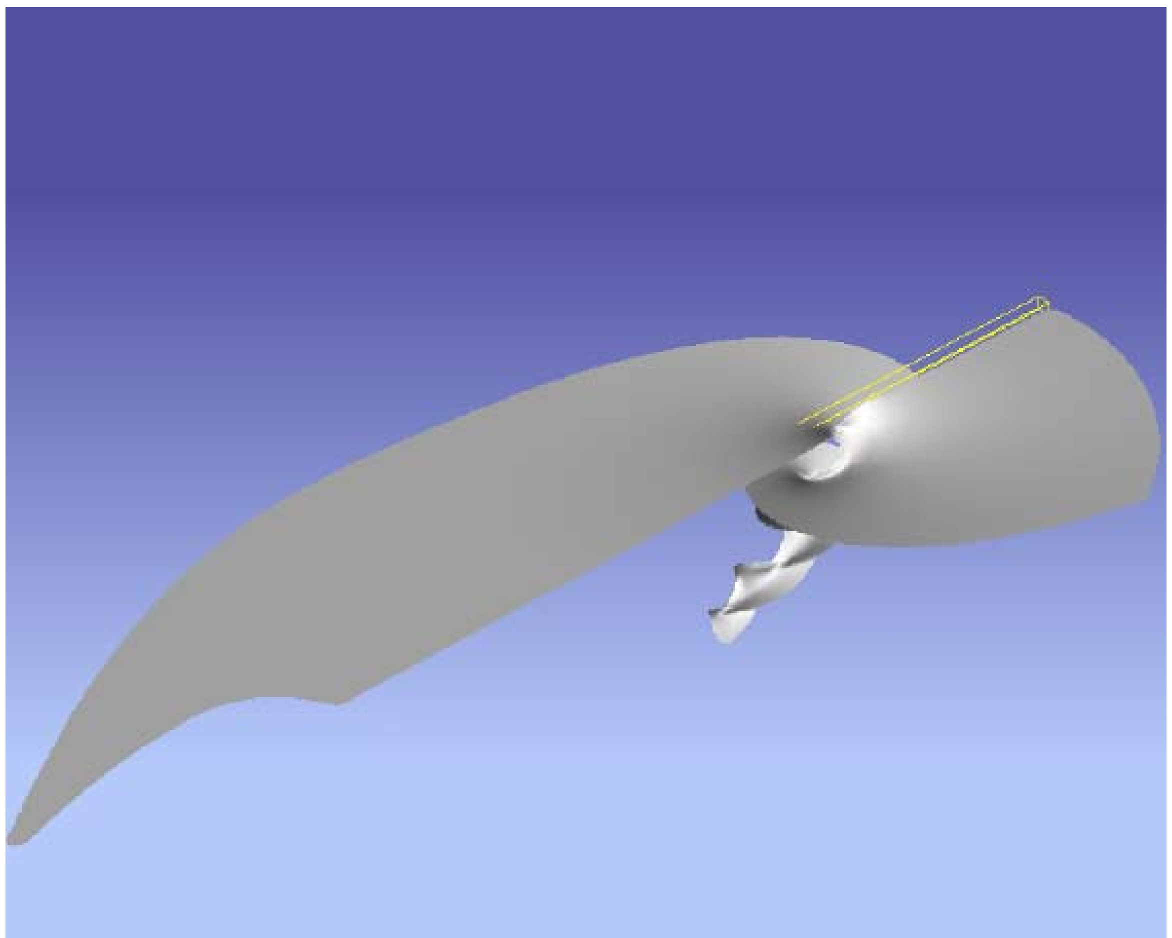


Abraham/Shaw's illustration, 1984

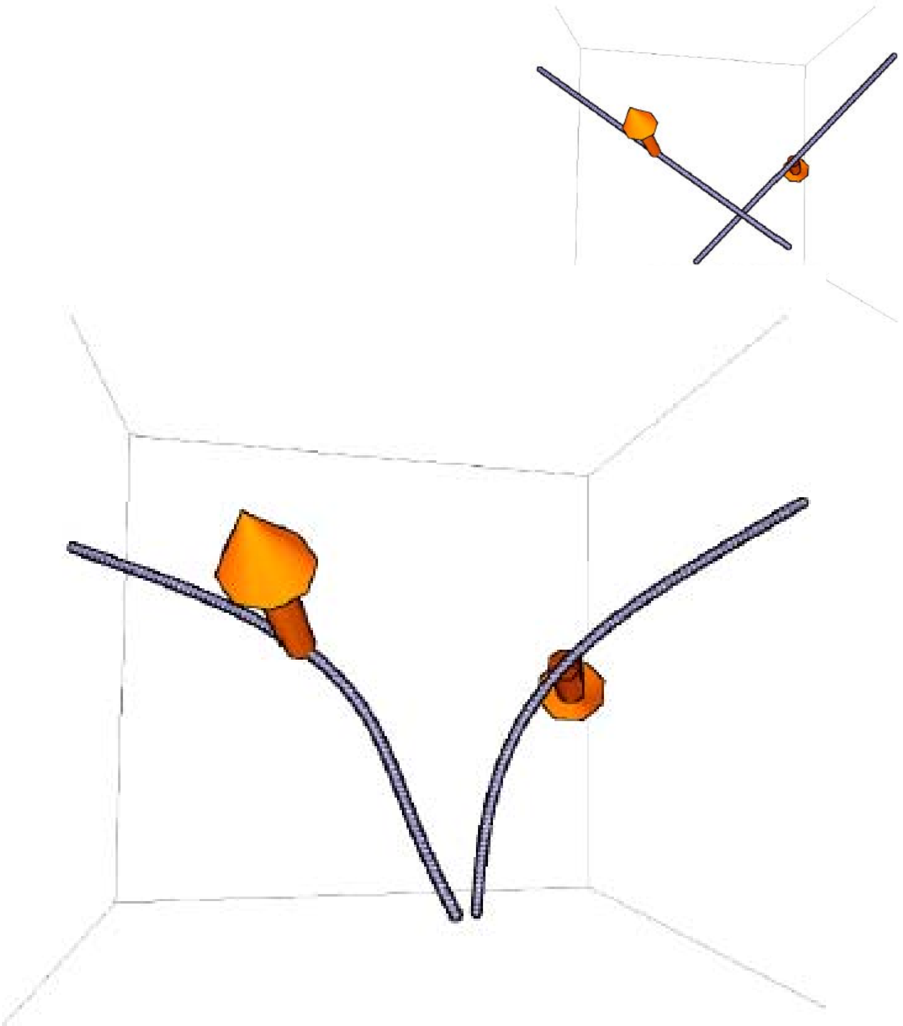


[Born et al. Vis2010]

Where to put seeds to start the integration?



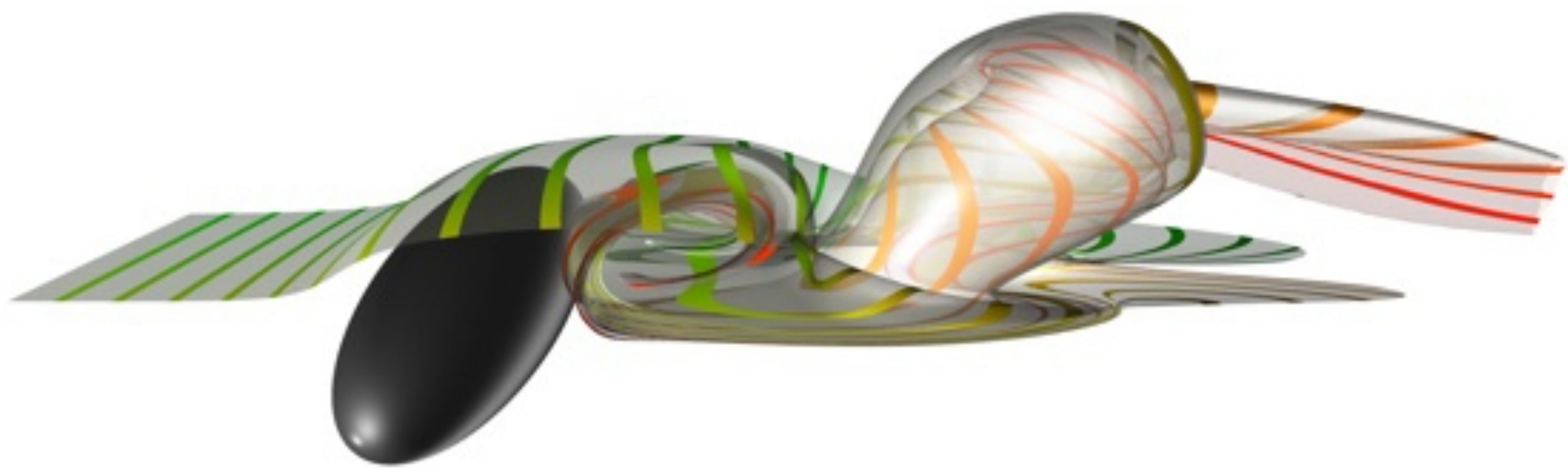
Seeding along a straight-line
Allow user exploration
[Weiskopf et al. 2007]



Seeding along the direction that is perpendicular to the flow leads to stream surface with large coverage
[Edmunds et al. EuroVis2012]

Time and streak surfaces

http://www.vacet.org/gallery/images_video/Krishnan_TimeStreakSurfaces.mp4



Hari Krishnan, Christoph Garth, Ken Joy. Time and Streak Surfaces for Flow Visualization in Large Time-Varying Data Sets. IEEE Visualization 2009.

Approaches to flow vis

- "How?"
  - Characteristic curves of the vector field (streamlines, pathlines, streaklines, timelines, Lagrangian coherent structures / FTLE)
  - **Texture-based (LIC, spot noise)**
  - Direct + geometry-based (hedehogs, glyphs)
  - Direct + heuristic (magnitude, Laplacian, FTLE)
  - Physically-based (Schlieren imaging, virtual rheoscopic fluids)
- "Where?"
  - Flow in 2D
  - Flow on surfaces
  - Flow in 3D space

Overview — Texture-Based Methods

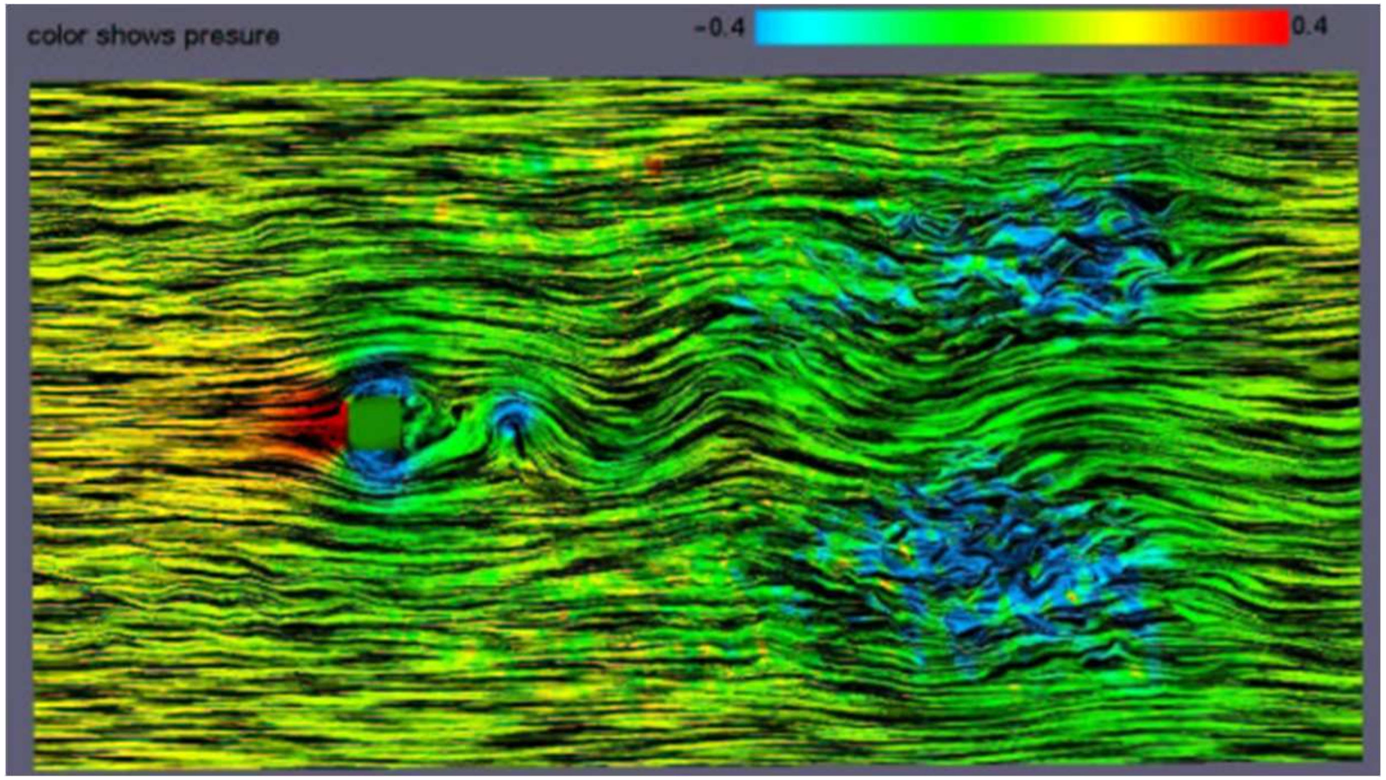
➢ Spot Noise

✧ One of the first texture-based techniques (*Van Wijk, Siggraph1991*).

✧ **Basic idea**: *distribute a set of intensity function, or spot, over the domain, that is wrapped by the flow over a small step*.

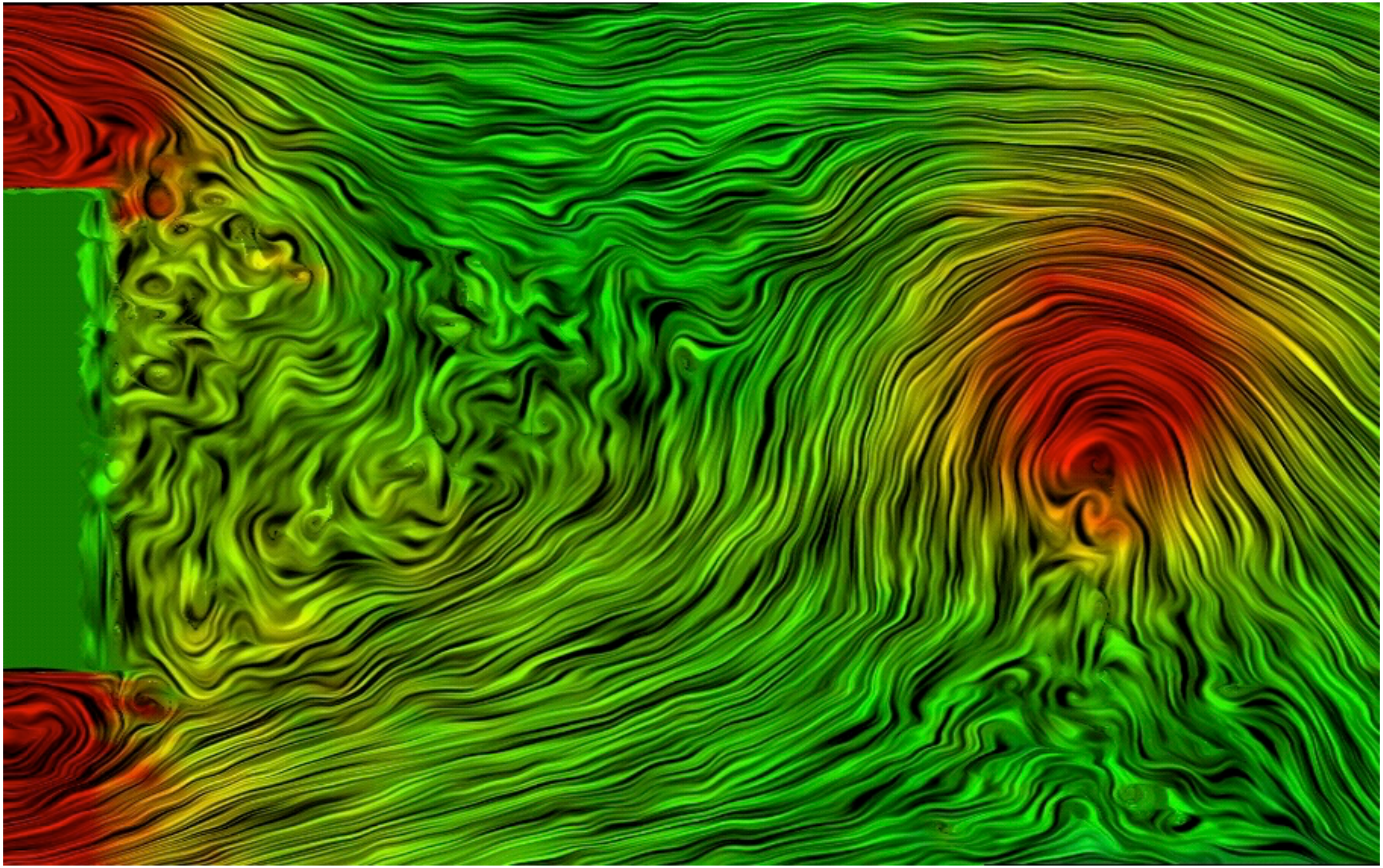
✧ **Pro**: mimic the smear effect of oil; encode magnitude; can be applied for both steady and unsteady flow.

✧ **Con**: tricky to implement; low quality; computationally expensive.



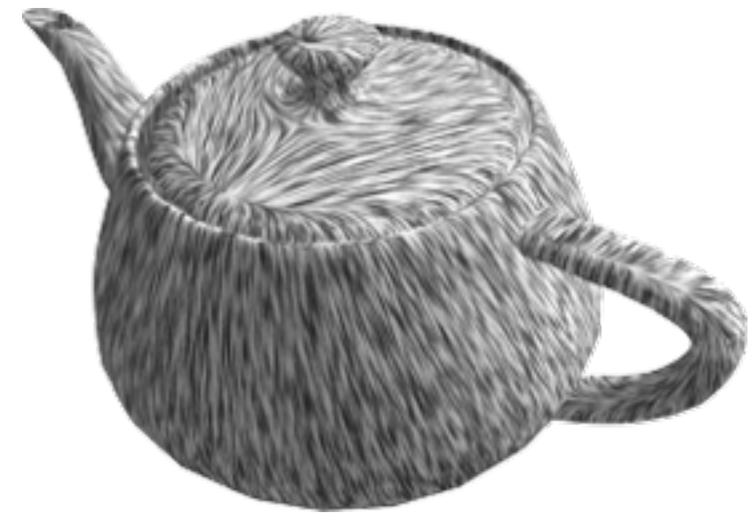
[De Leeuw and Van Liere]

Spot noise

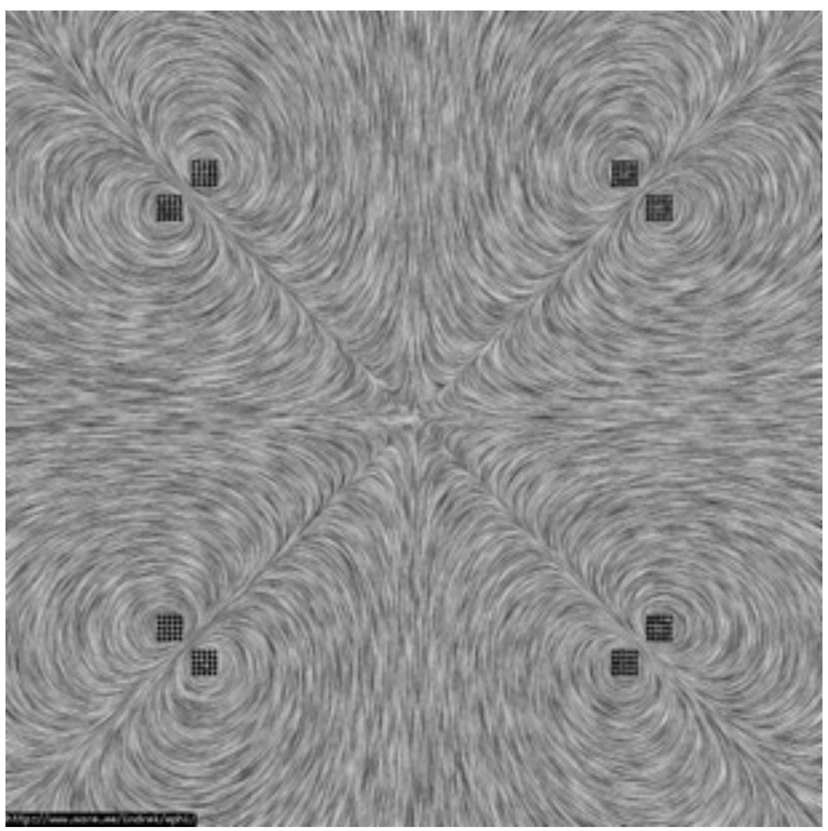


- Image: Wim de Leeuw. http://homepages.cwi.nl/~robertl/movies/flow1.mpg

LIC – Line Integral Convolution

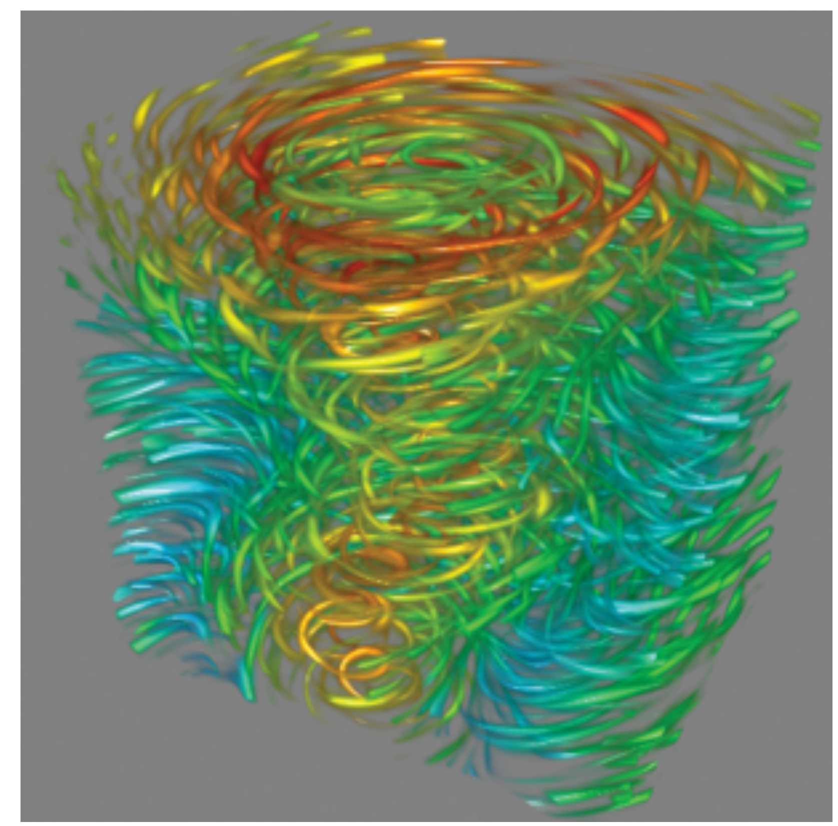


- (Cabral/Leedom, Siggraph 1993)
- A global method to visualize vector fields



2D vector field

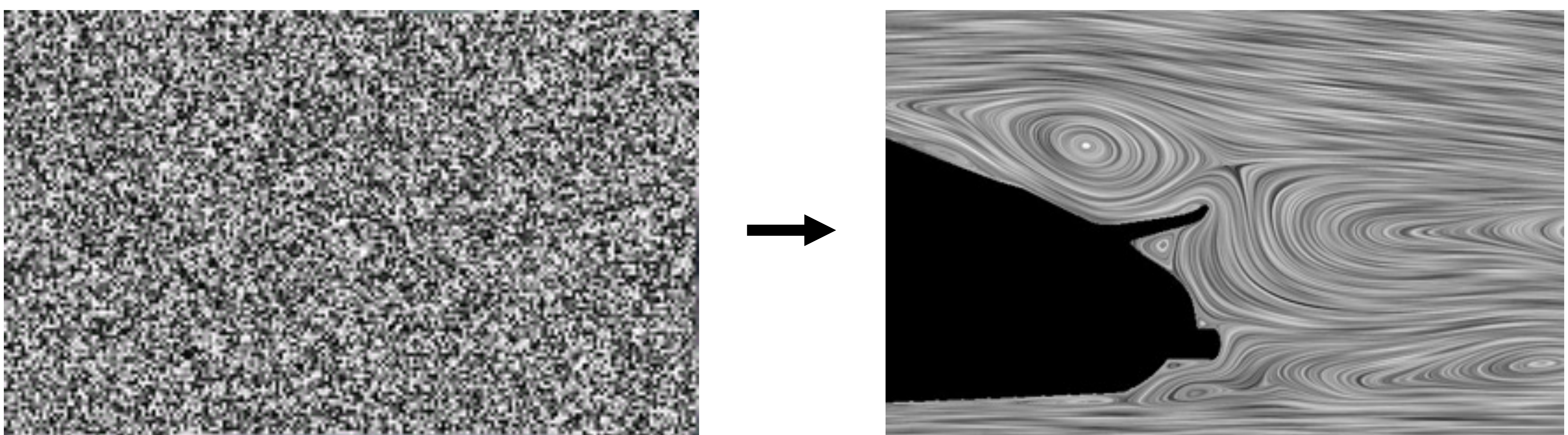
vector field on surface
(often called 2.5D)



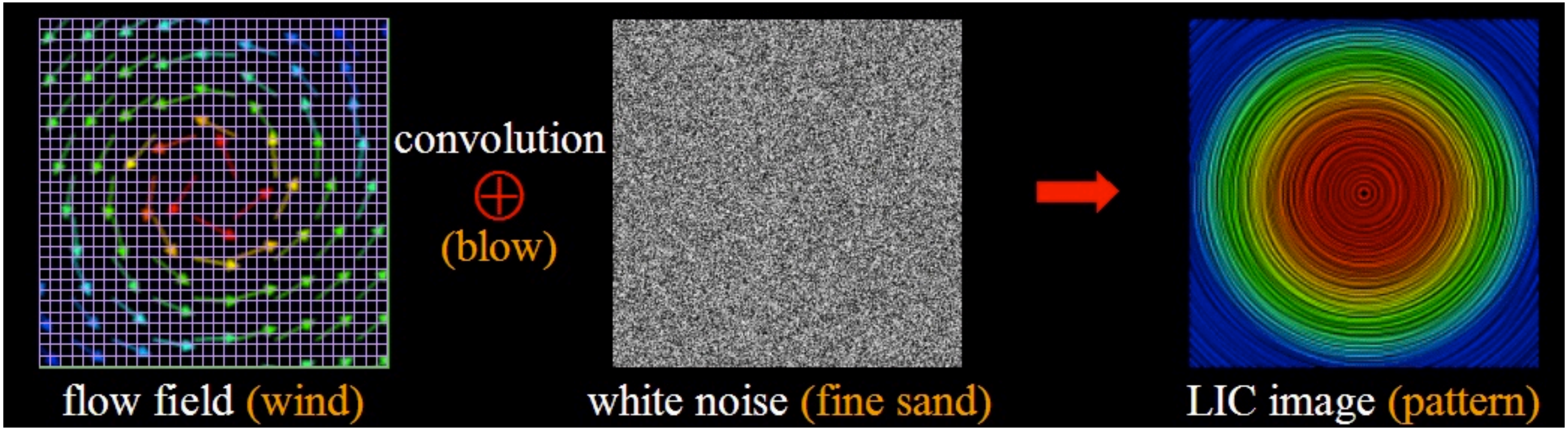
3D vector field

Idea of LIC

- **Global** visualization technique; not only one particle path
- Start with a **random texture**
- **Smear** out this texture along the path lines in a vector field, results in
  - **Low correlation** of intensity values **between** neighboring lines,
  - But **high correlation along** them

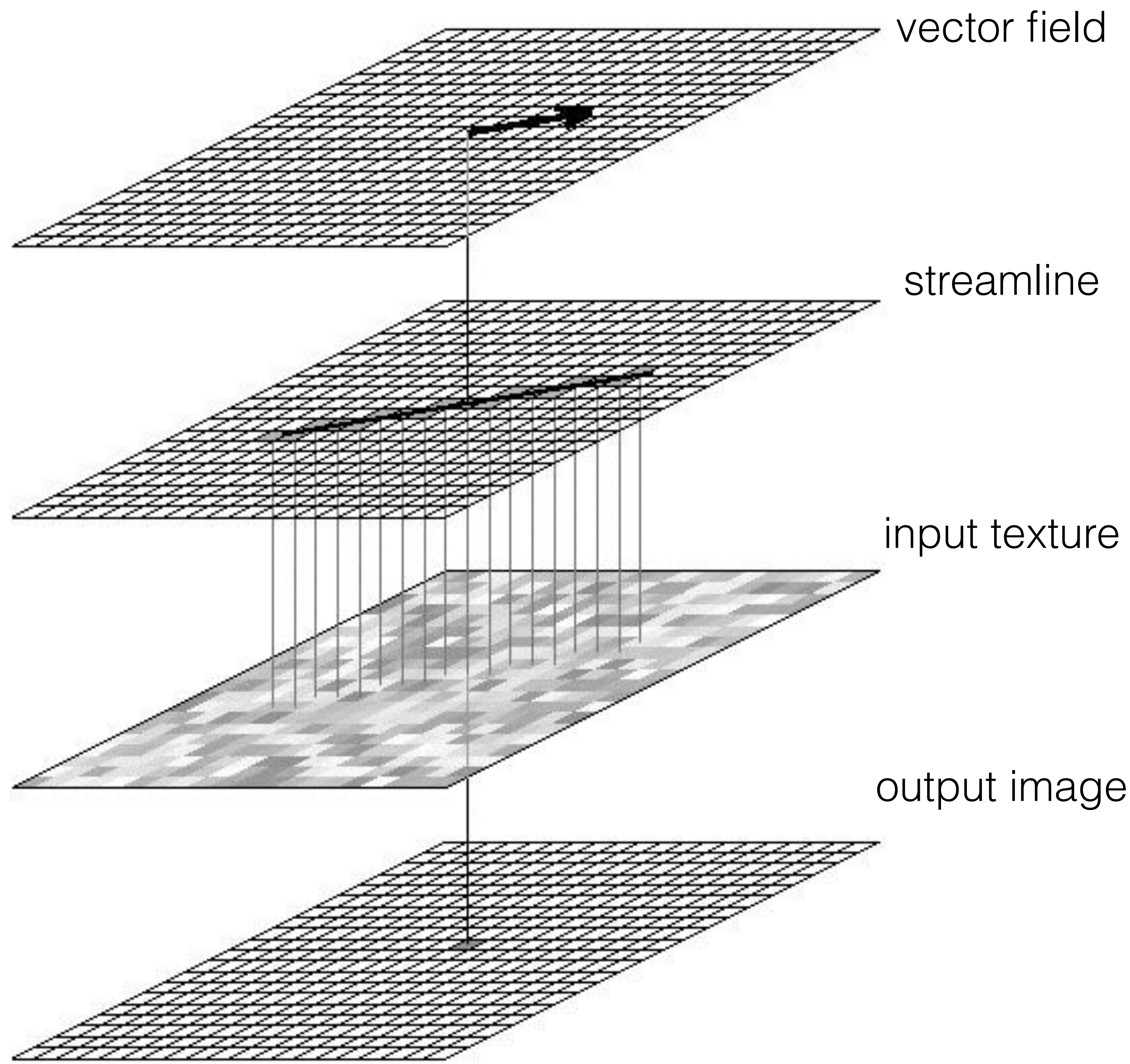


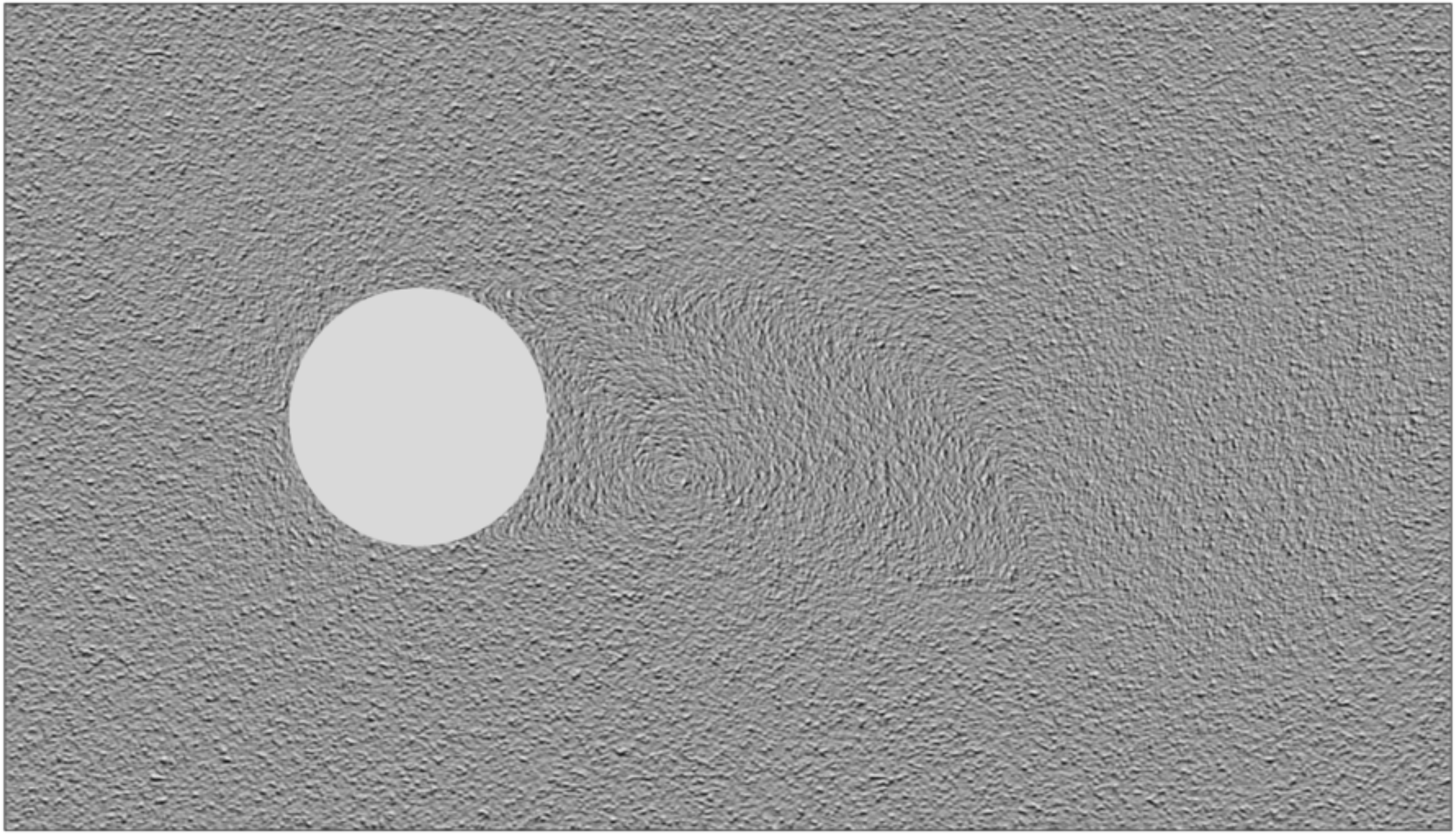
Idea of LIC


convolution
(blow)
flow field (wind)
white noise (fine sand)
LIC image (pattern)

Algorithm for 2D LIC

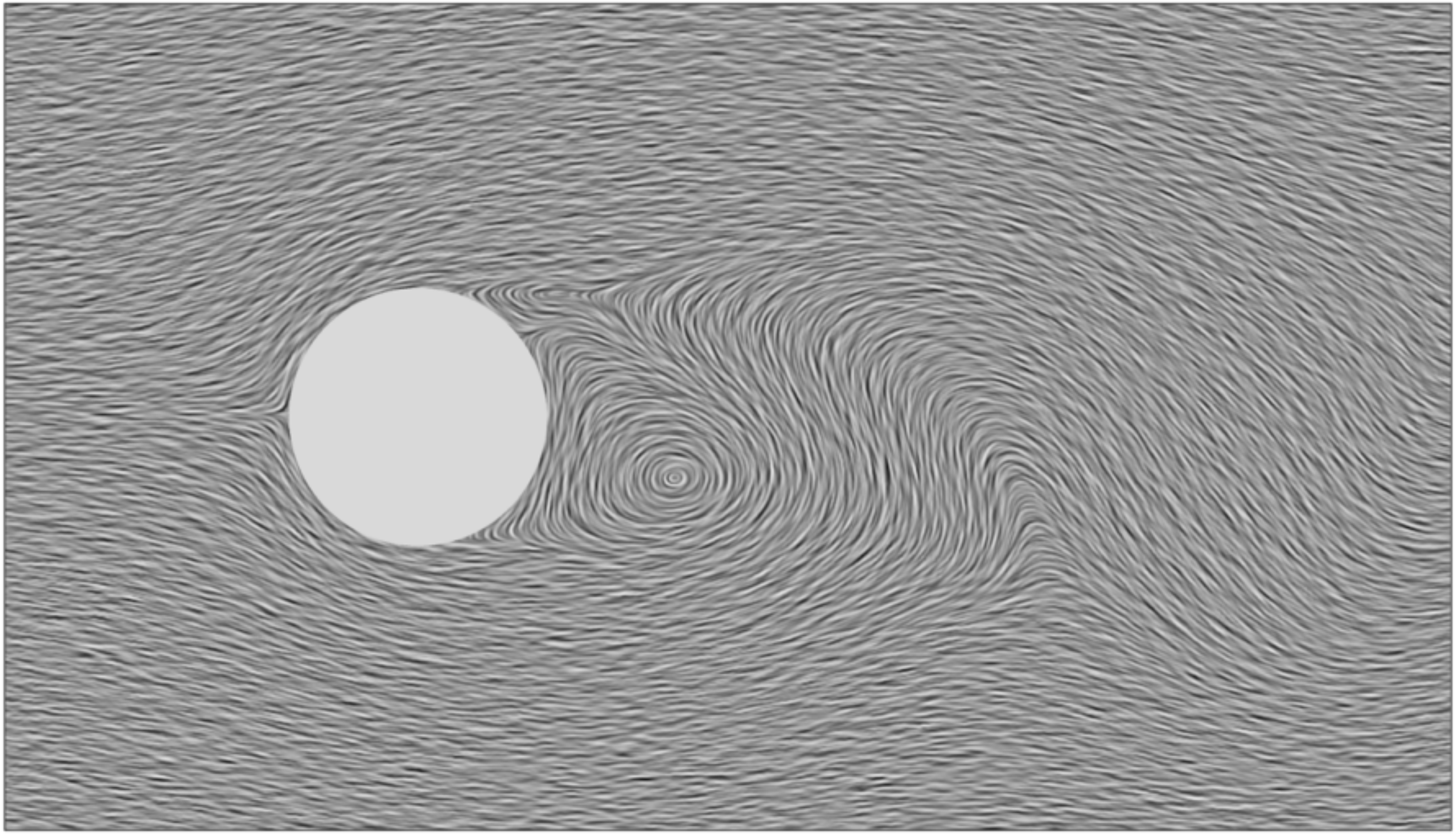
- Convolve a random texture along the streamlines





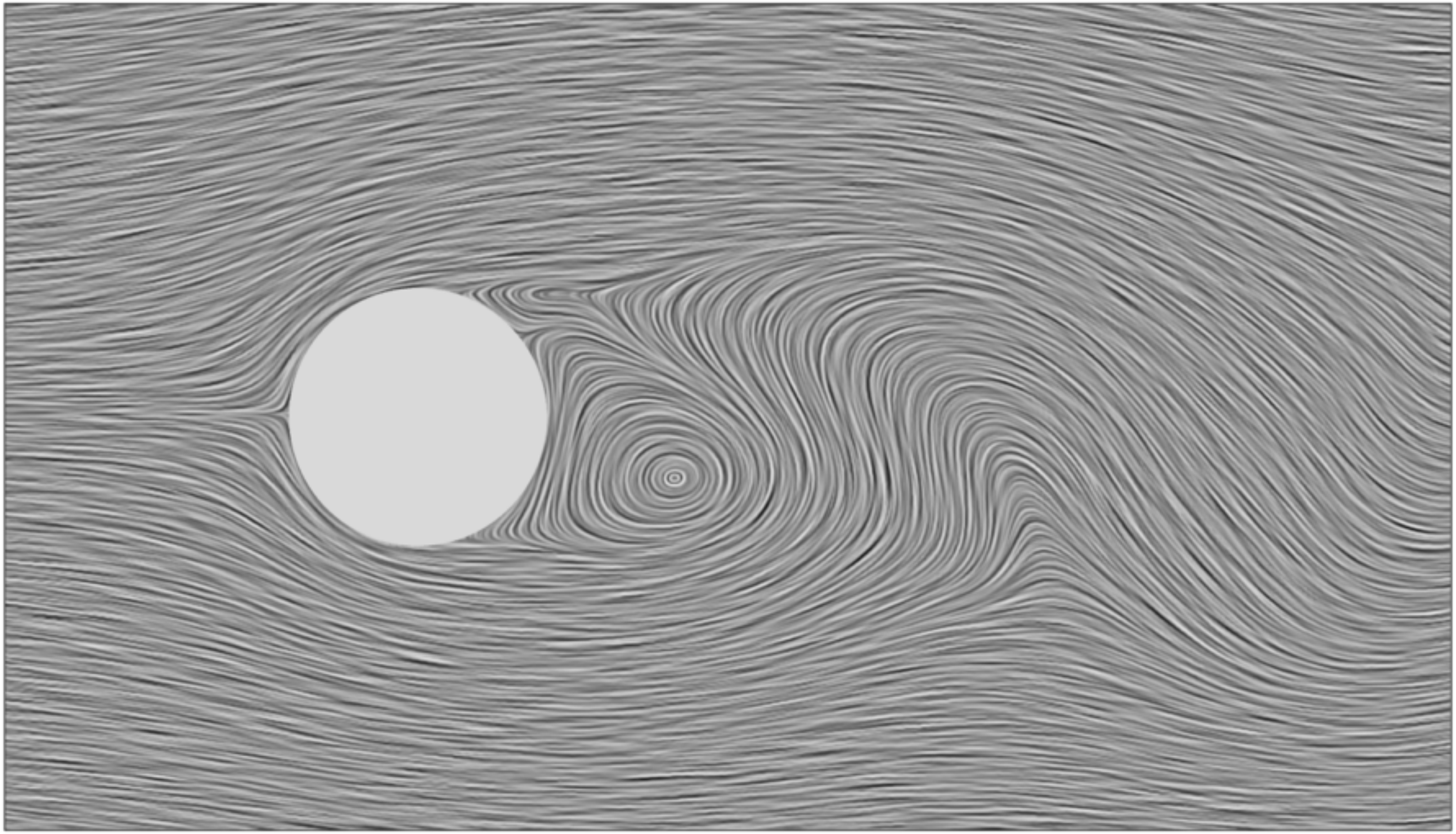
Filter length influences the quality of LIC images
filter length = 1

2D flow behind a cylinder



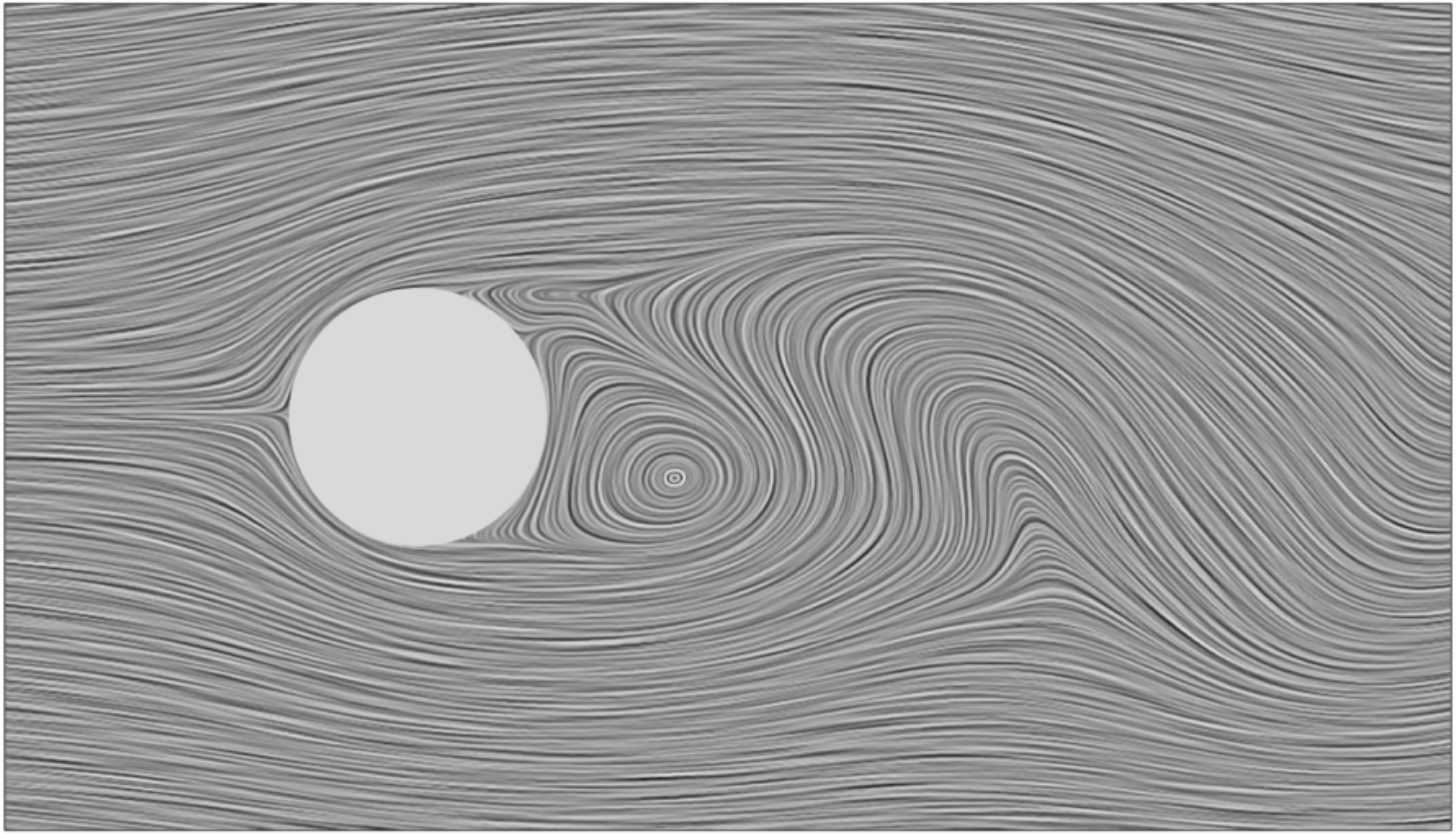
Filter length influences the quality of LIC images
filter length = 10

2D flow behind a cylinder



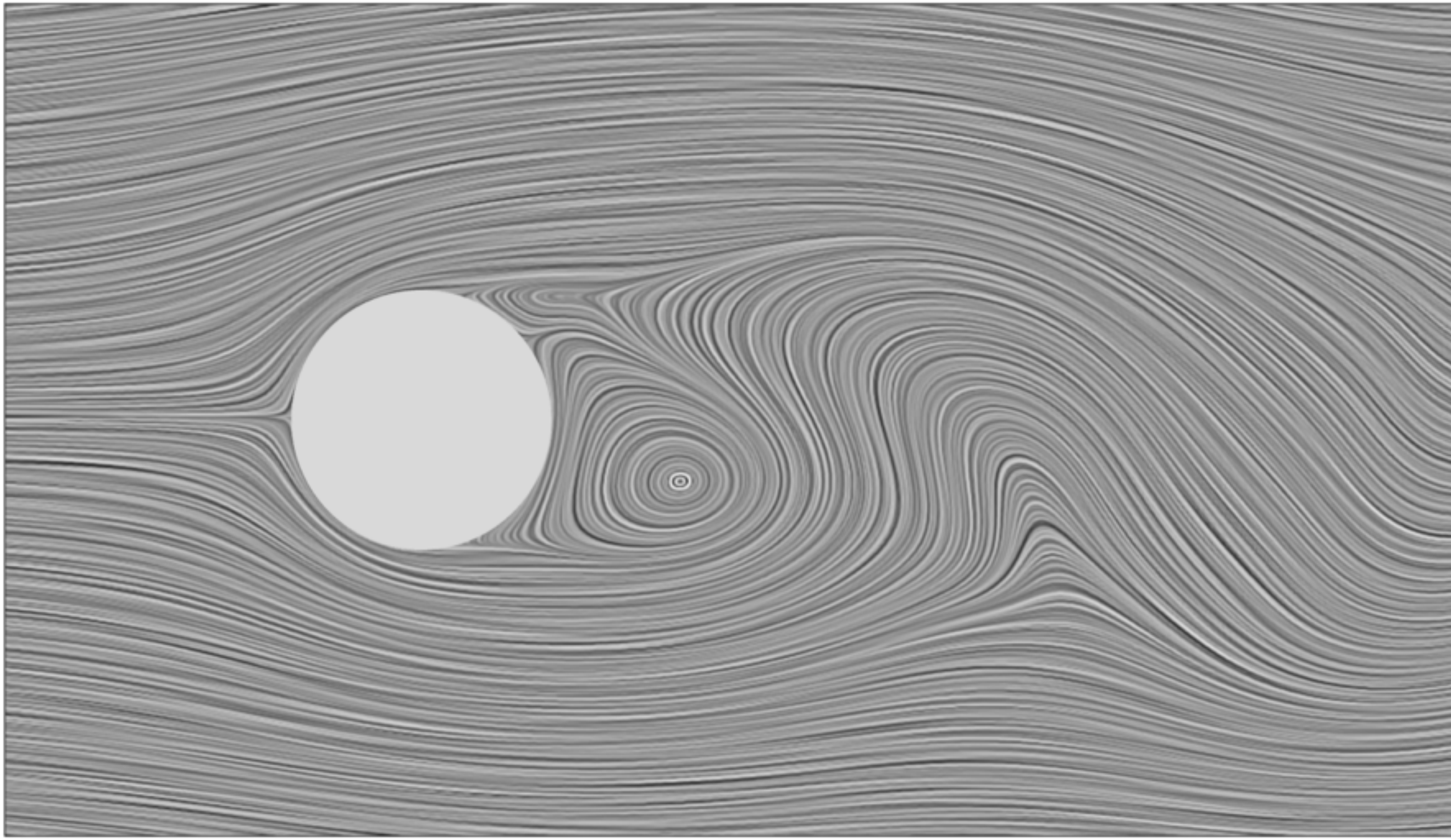
Filter length influences the quality of LIC images
filter length = 25

2D flow behind a cylinder



Filter length influences the quality of LIC images
filter length = 50

2D flow behind a cylinder

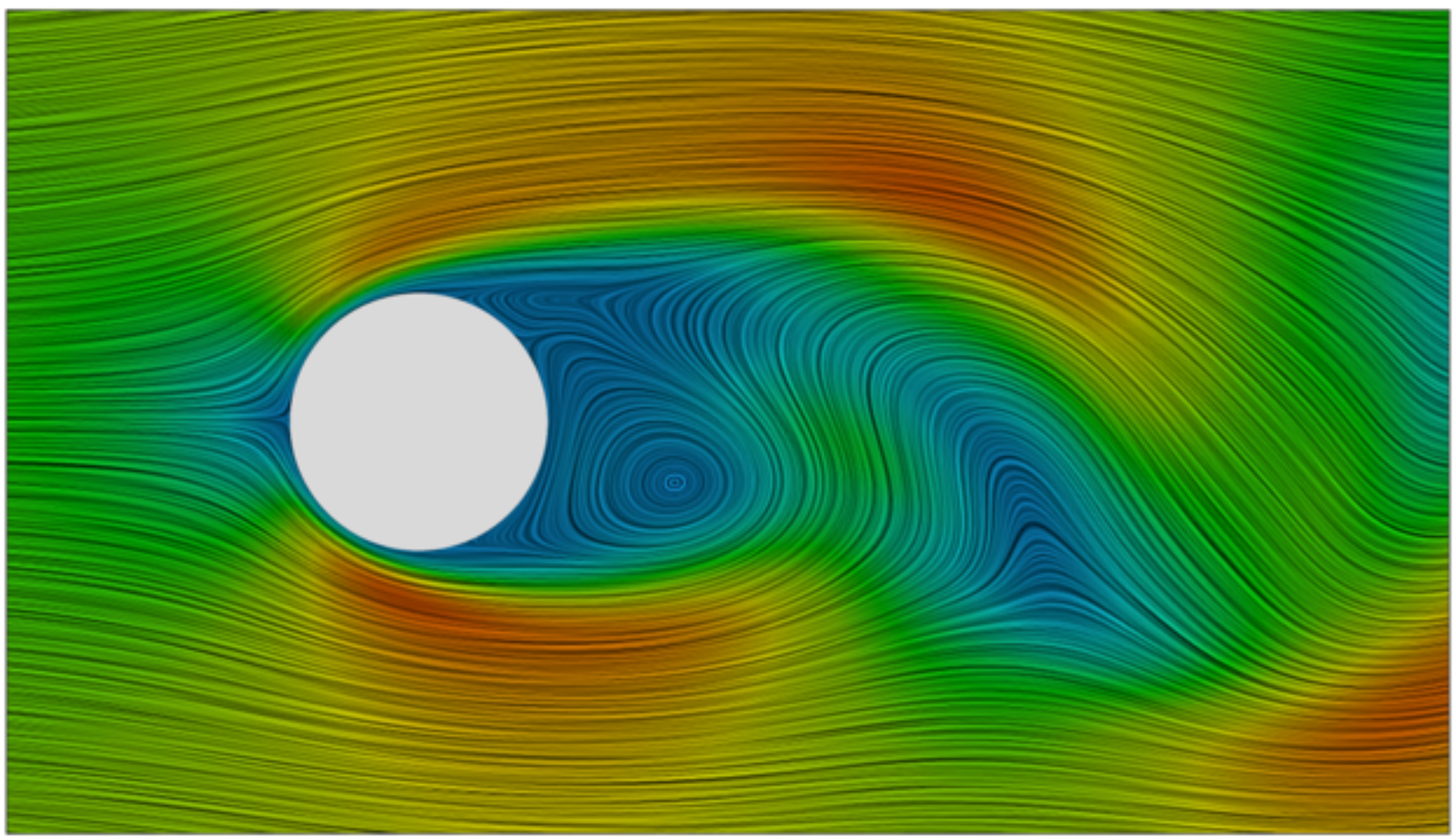


Filter length influences the quality of LIC images
filter length = 100

2D flow behind a cylinder

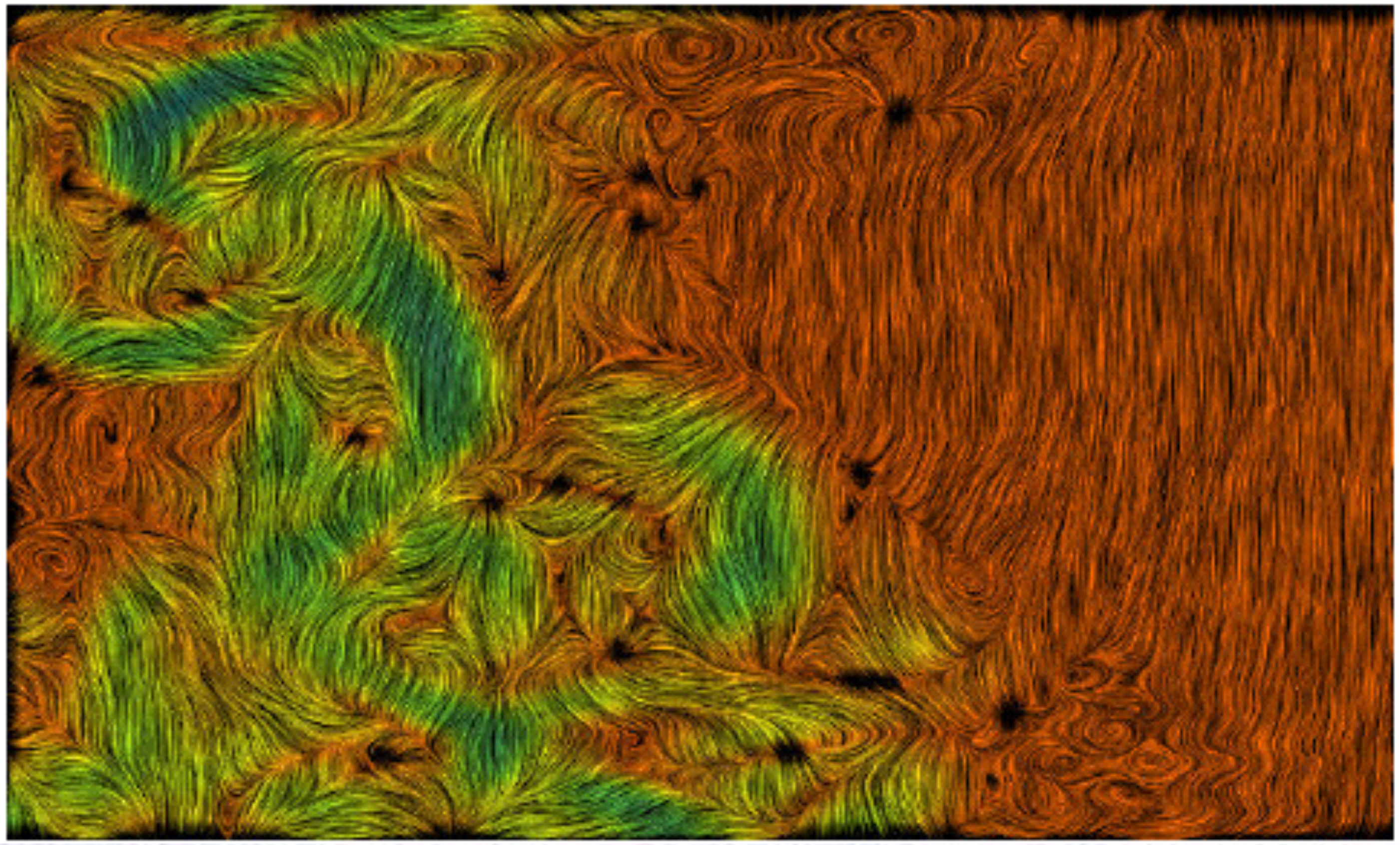
LIC Color Coding

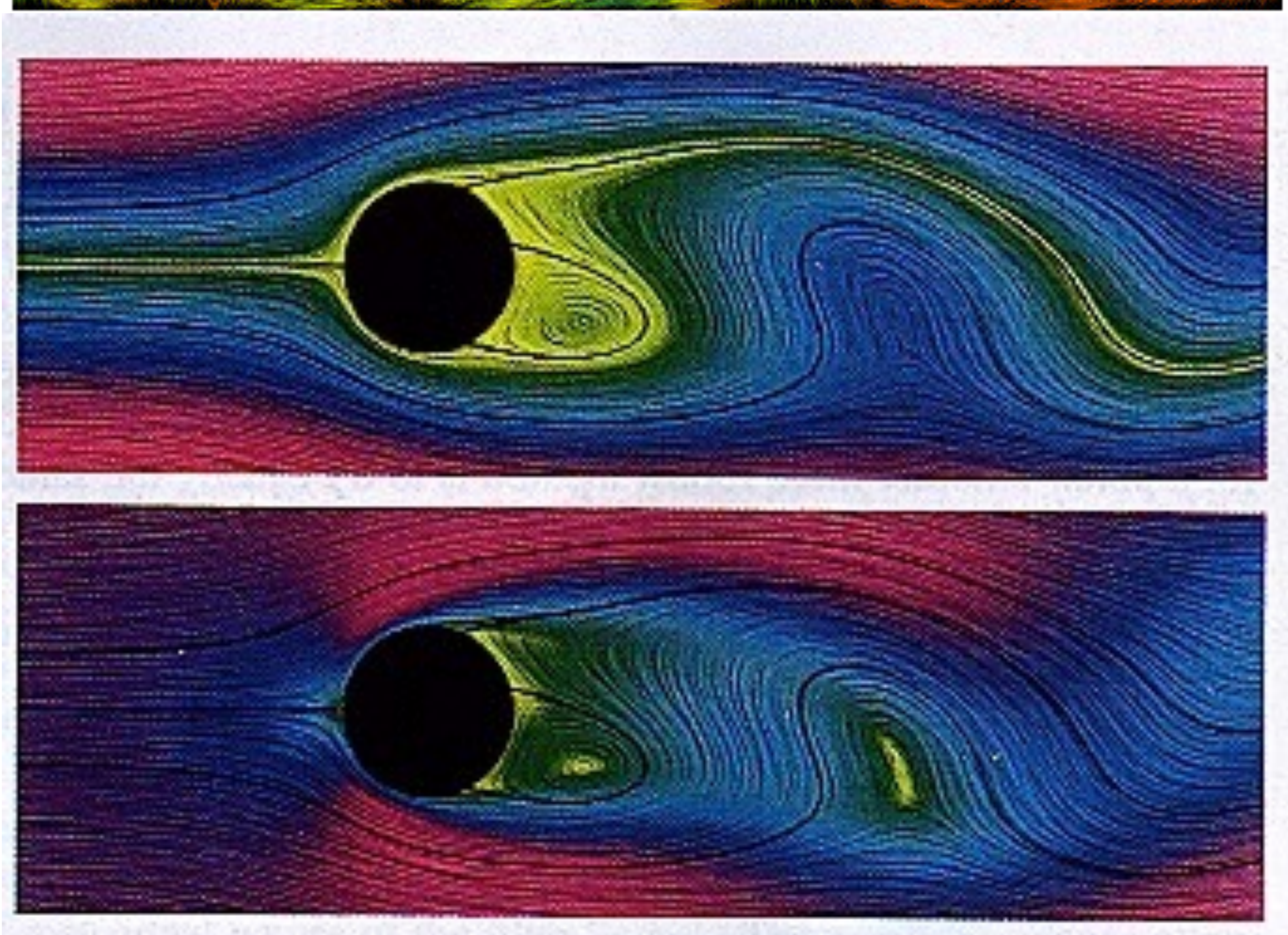
- Usually, LIC does not use the color channel
  - Could use color to encode scalar quantities



Velocity magnitude encoded using color

2D flow behind a cylinder

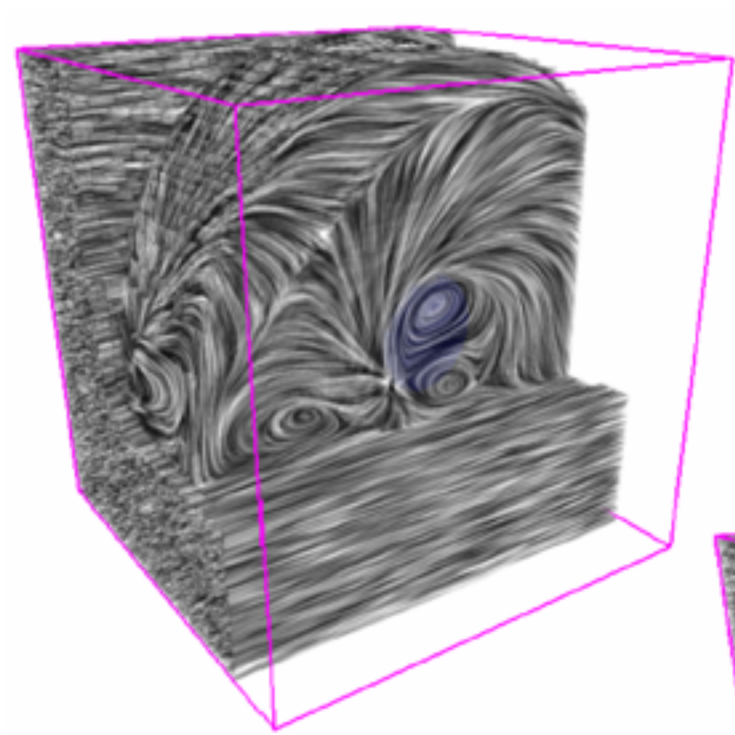


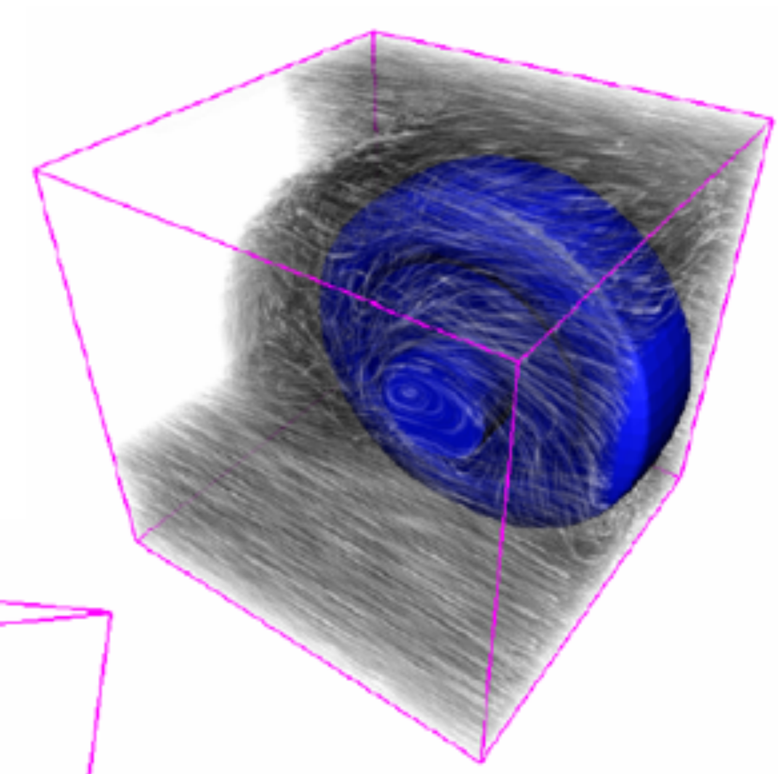


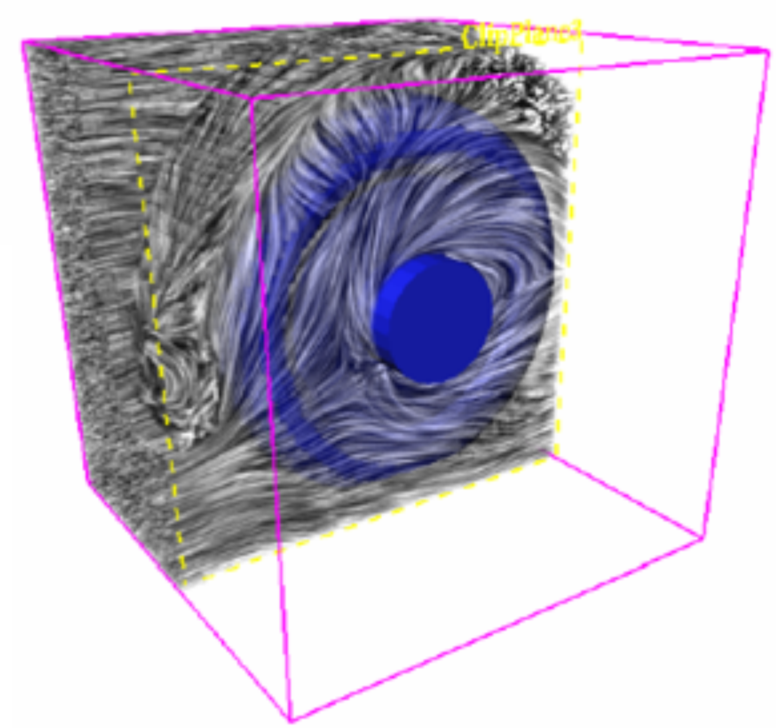
LIC and color coding of velocity magnitude

LIC for 3D Flows

- LIC concept easily extendable to 3D
- Problem: rendering!







3D LIC can only reveal interesting structures if some data is discarded.

Images courtesy of [Rezk-Salama et al. 99]

Overview — Texture-Based Methods

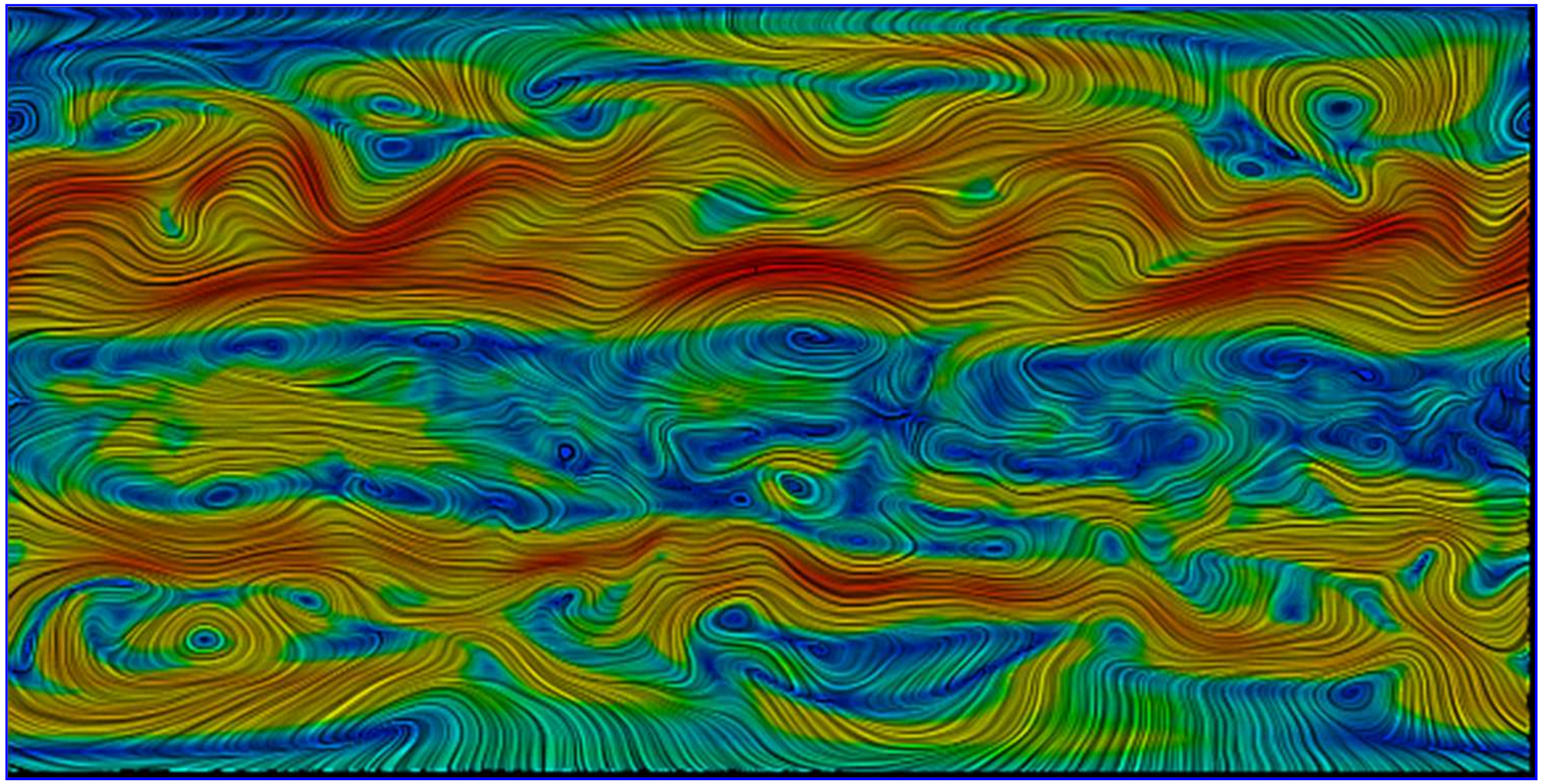
➢ Unsteady Flow LIC (UFLIC)

✧ The first texture-based **unsteady** flow visualization method (by *Han-Wei Shen and David Kao, IEEE Visualization 97 & IEEE TVCG 98*).

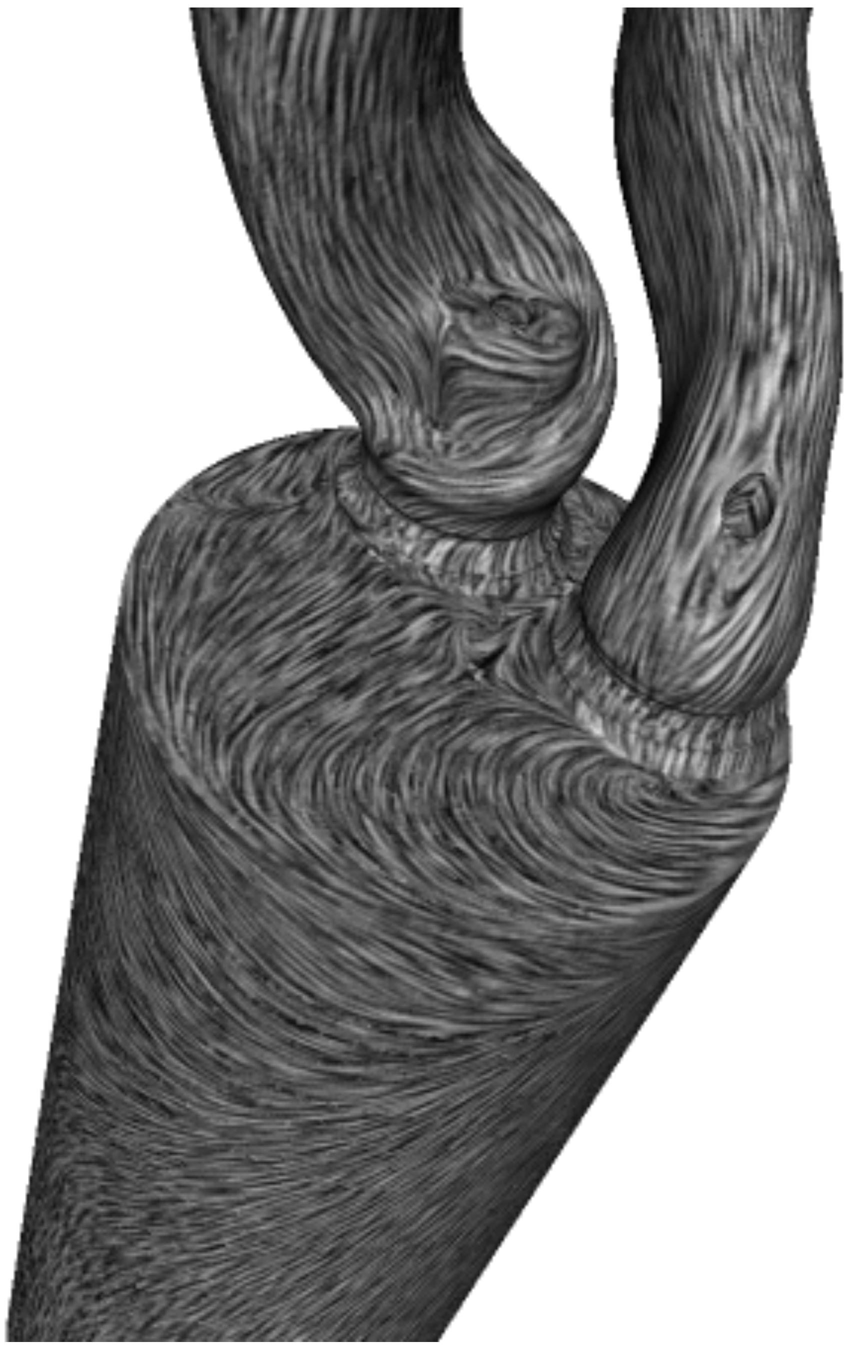
✧ **Basic idea:** Time-accurately scatters particle values of *successively fed-forward textures* along **pathlines** *over several time steps* to convey the footprint / contribution that a particle leaves at downstream locations as the flow runs forward.

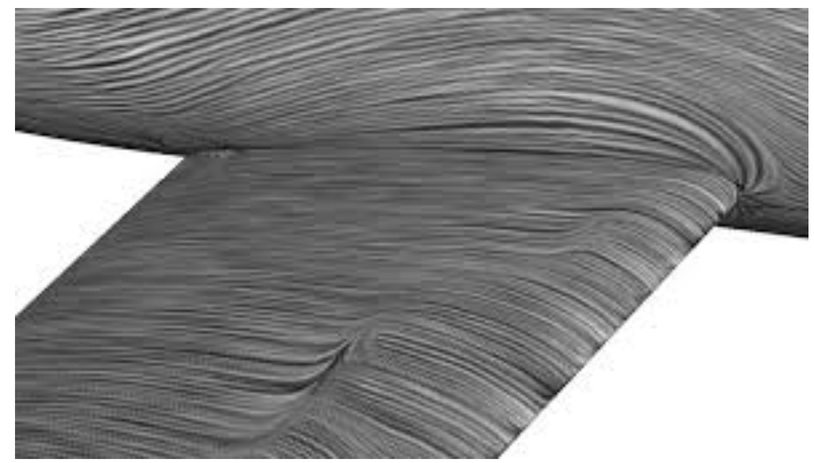
✧ **Pro**: High temporal coherence & high spatial coherence & hardware-independent.

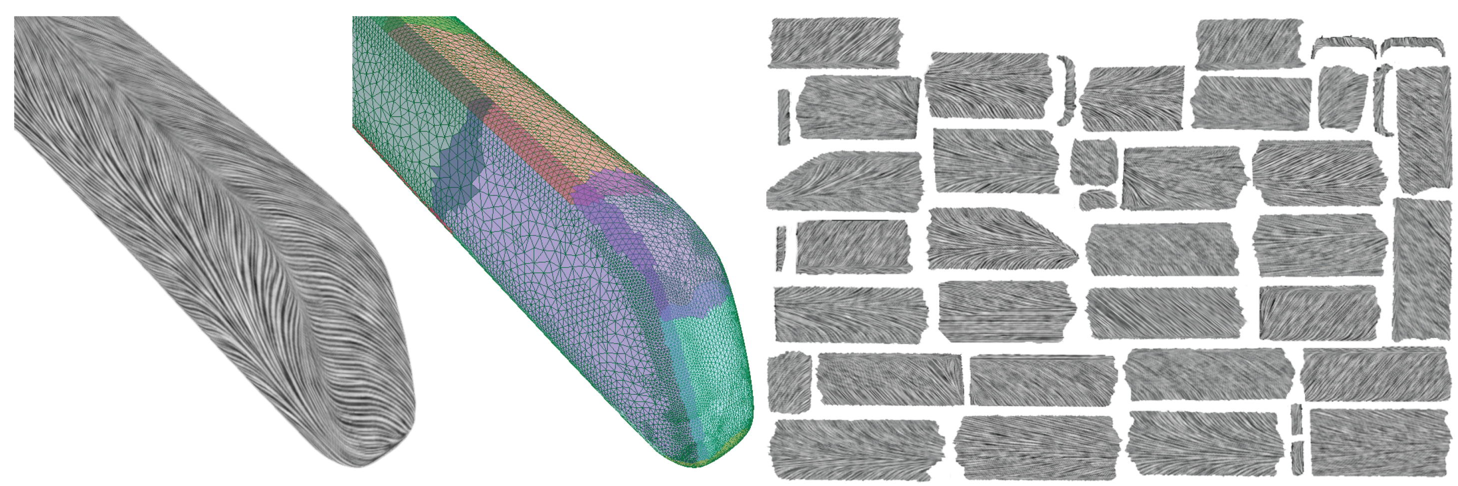
✧ **Con**: Low computational performance due to *multi-step* (≈ 100) *pathline integration.*



GPU-accelerated UFLIC on arbitrary surfaces







Flow Charts: Visualization of Vector Fields on Arbitrary Surfaces. G Li, X Tricoche, D Weiskopf, C Hansen.
IEEE TVCG 2008.

Overview — Texture-Based Methods

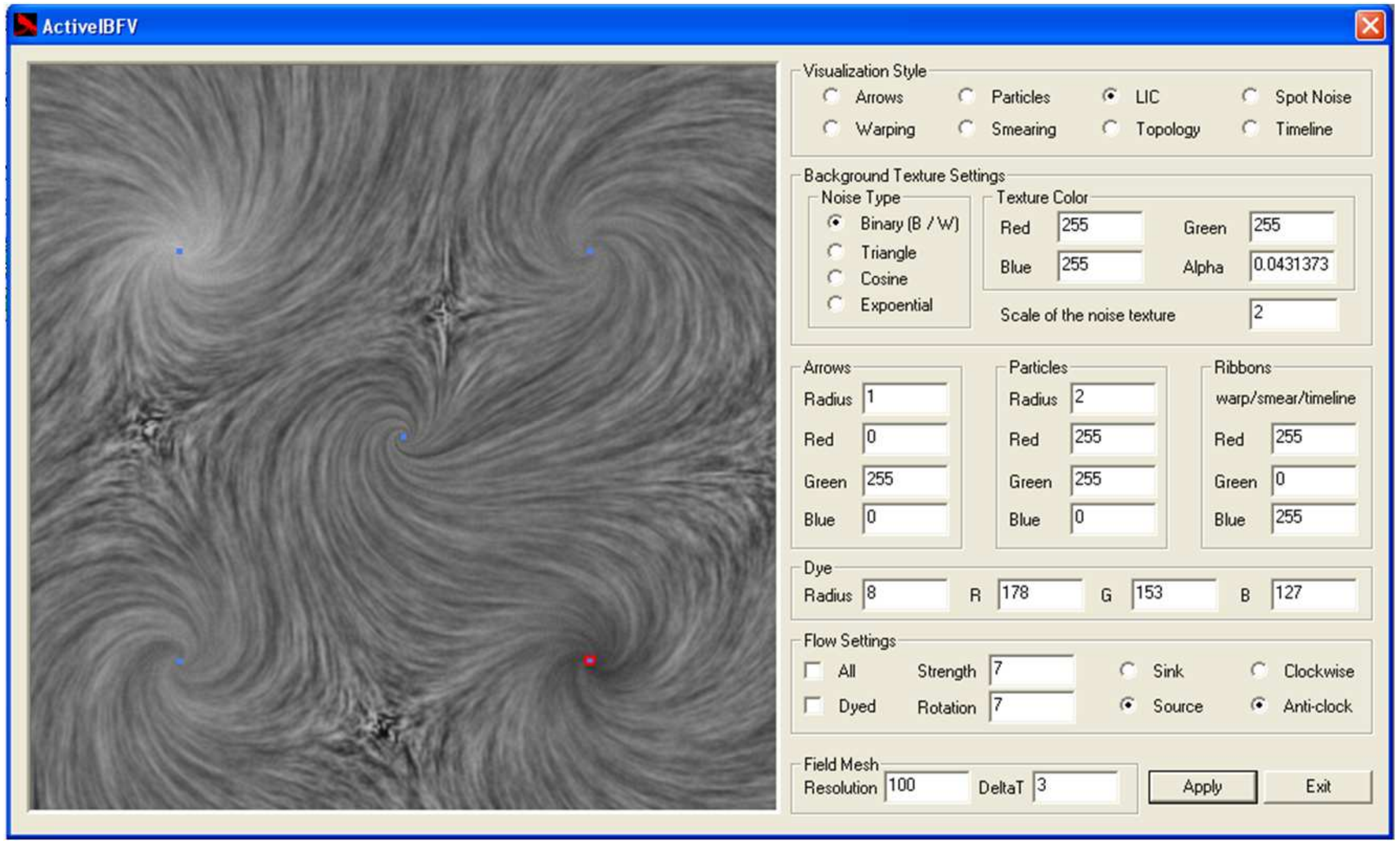
➢ Image-Based Flow Visualization (IBFV)

✧ One of the most versatile and the easiest-to-implement hardware-based methods (by *Jarke J. van Wijk, SIGGRAPH02*).

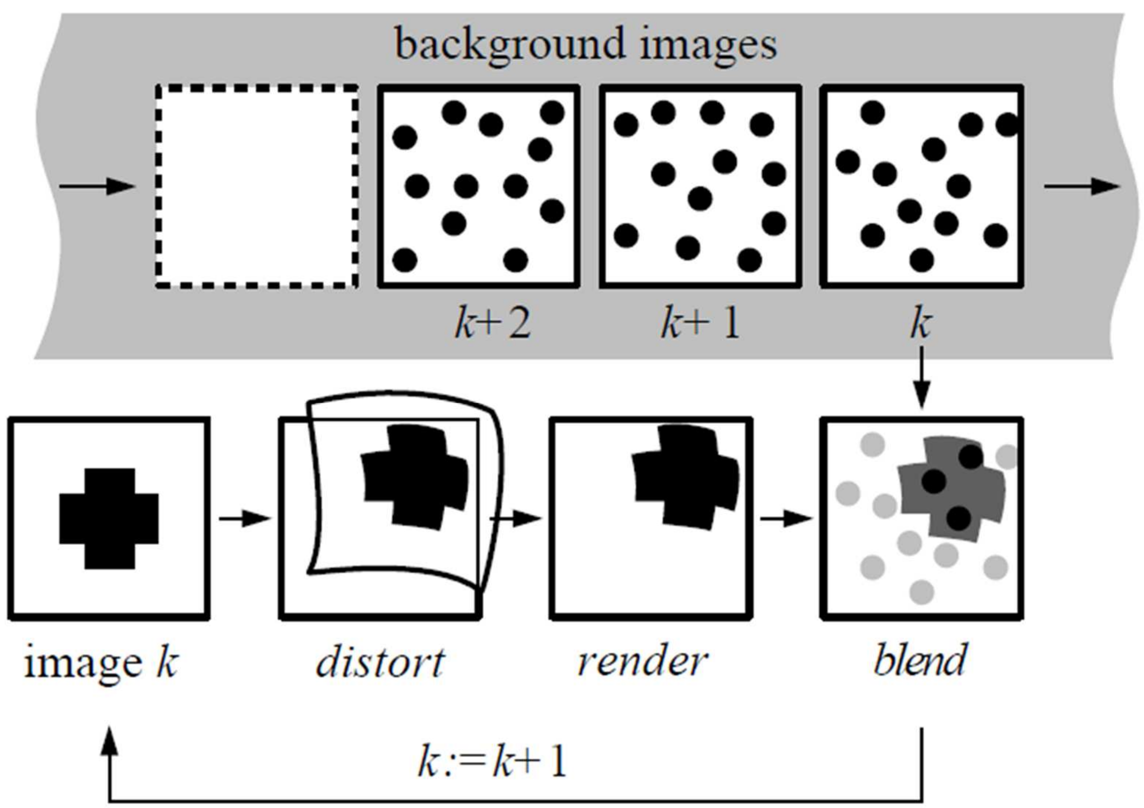
✧ **Basic idea:** Designs a sequence of *temporally-spatially low-pass filtered* noise textures and cyclically blends them with an iteratively advected (using *forward single-step pathline integration*) image (which is initially a BLACK rectangle).

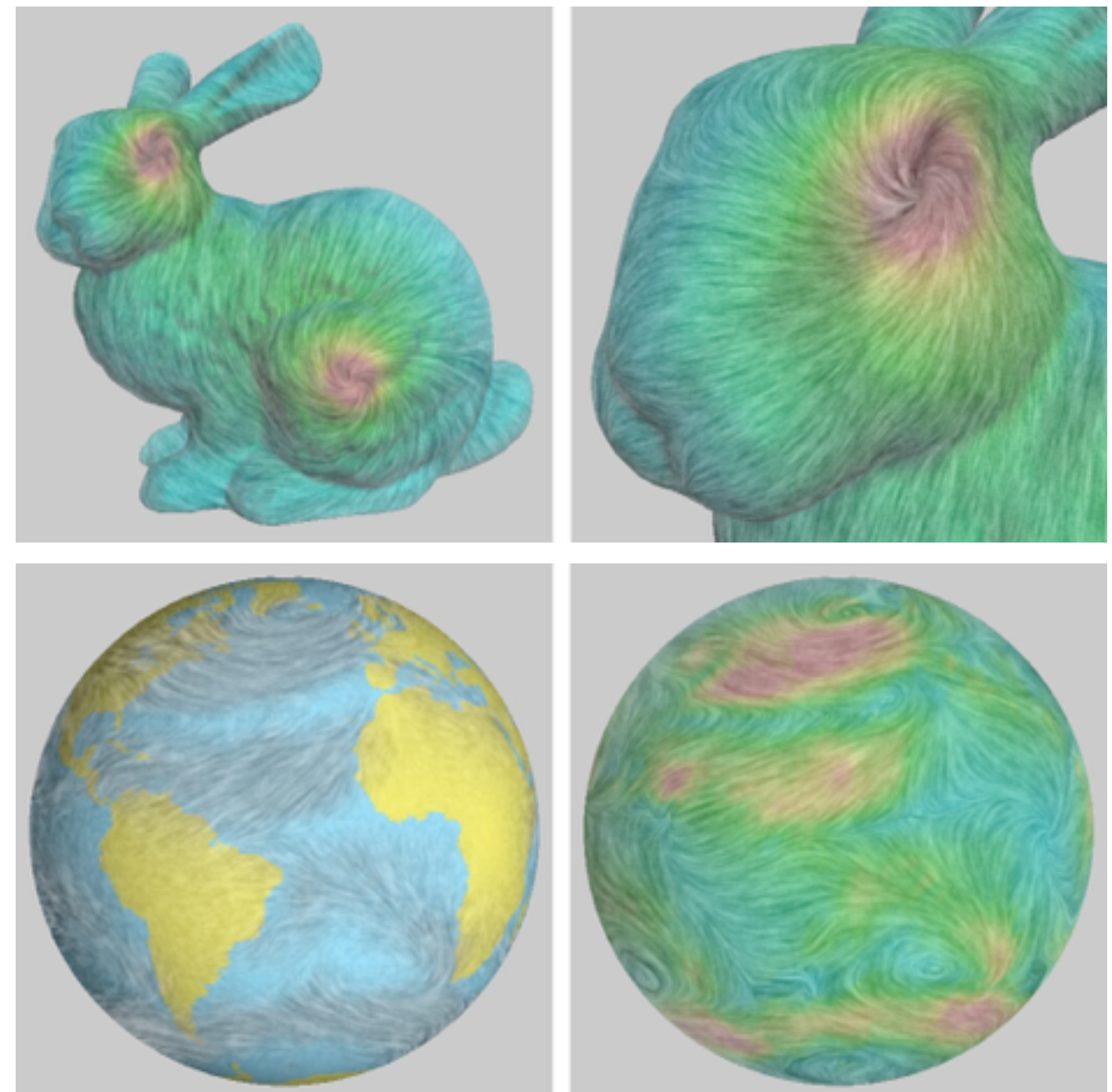
✧ **Pro:** Interactive frame rates and easy simulation of many visualization techniques.

✧ **Con:** Good temporal coherence and insufficient spatial coherence (*noisy or blurred*).



IBFV: Image-Based Flow Visualization (Advect Dye in Image-Space)



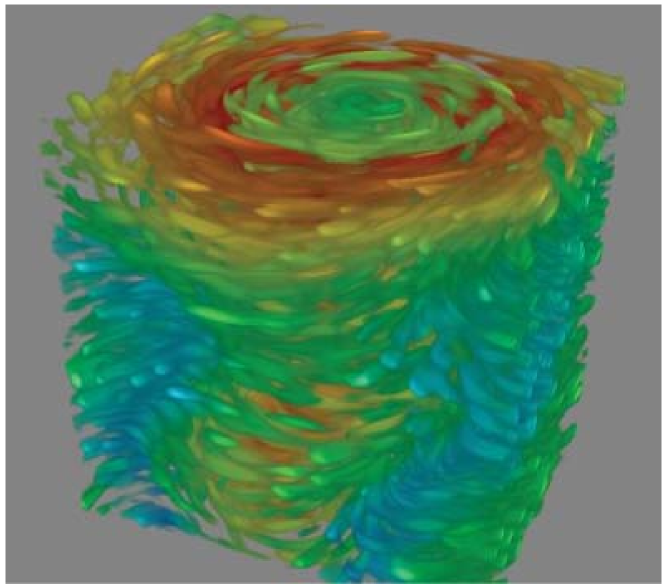


http://www.win.tue.nl/~vanwijk/ibfv/

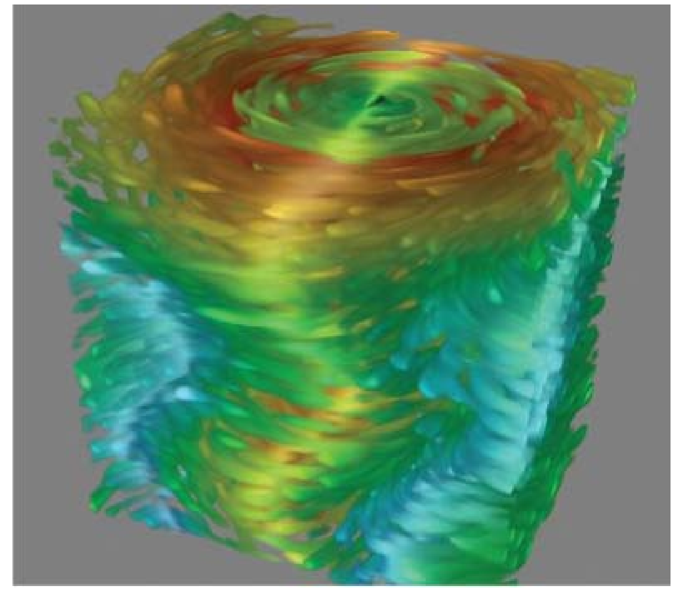
http://www.win.tue.nl/~vanwijk/ibfvs/

Volumetric LIC

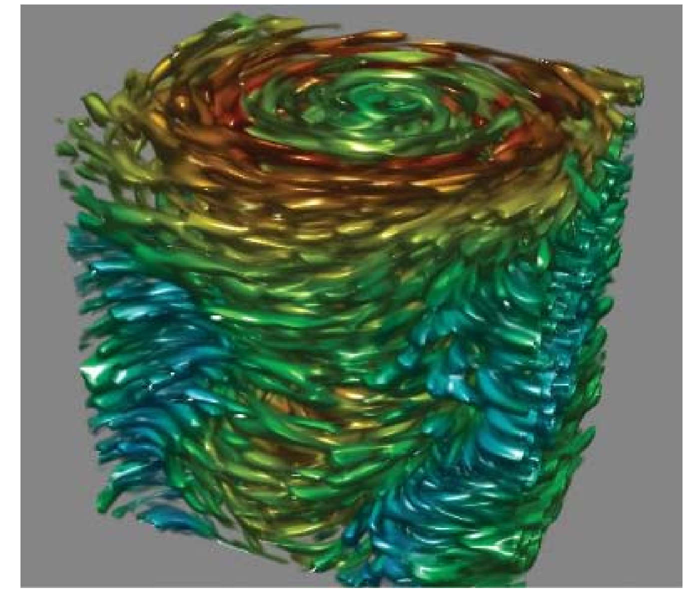
Recent Advances in 3D Texture-based Method



without illumination

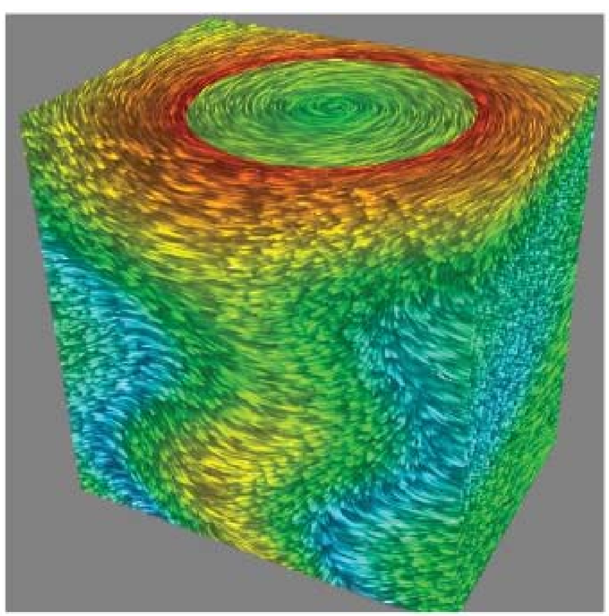


with illumination

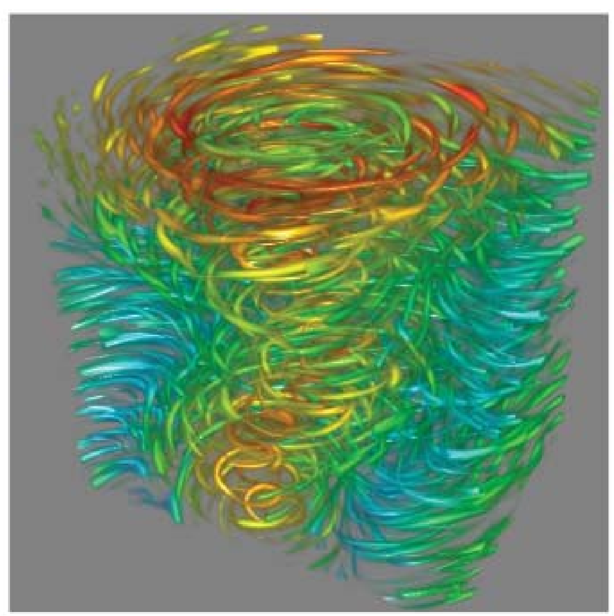


Gradient-based illumination

Codimension-2 illumination

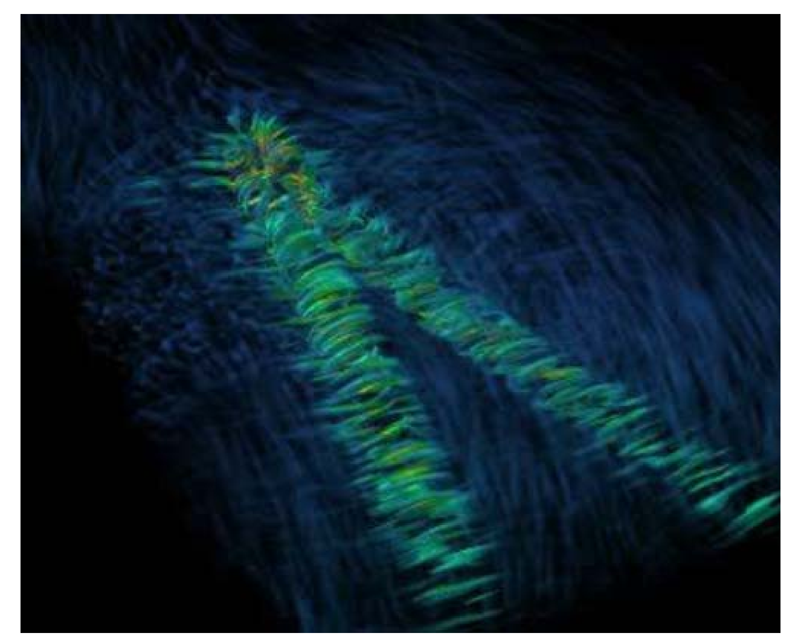


Dense (white noise)



Sparse noise

Different seeding strategies



Feature enhancement

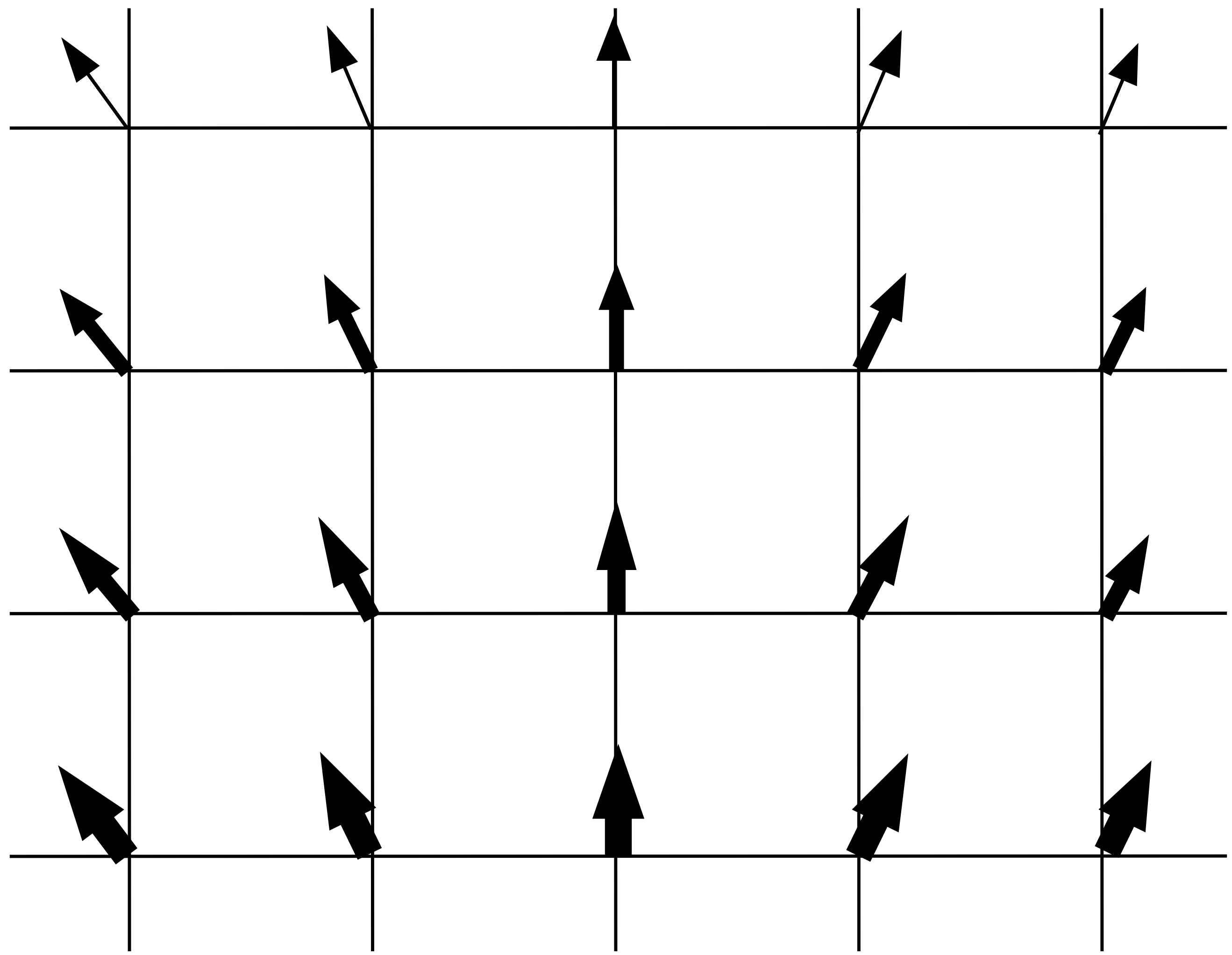
[Falk and Weikopf 2008]

Approaches to flow vis

- “How?”
    - Characteristic curves of the vector field (streamlines, pathlines, streaklines, timelines, Lagrangian coherent structures / FTLE)
    - Texture-based (LIC, spot noise)
    - **Direct + geometry-based (hedehogs, glyphs)**
    - Direct + heuristic (magnitude, Laplacian, FTLE)
    - Physically-based (Schlieren imaging, virtual rheoscopic fluids)
- “Where?”
    - Flow in 2D
    - Flow on surfaces
    - Flow in 3D space

- **Arrow plots:**
- also called hedgehog plots
- represent velocity as arrows at regular locations, e.g., place arrows at grid points
- ➔ overloading possible
- arrows: (scaled) unit length or encode magnitude
- well-established for 2D

- Arrows visualize
  - Direction of vector field
  - Orientation
  - Magnitude:
    - Length of arrows
    - Color coding



- [Kirby et al 99]: multiple values of 2d flow data by layering concept related to painting process of artists

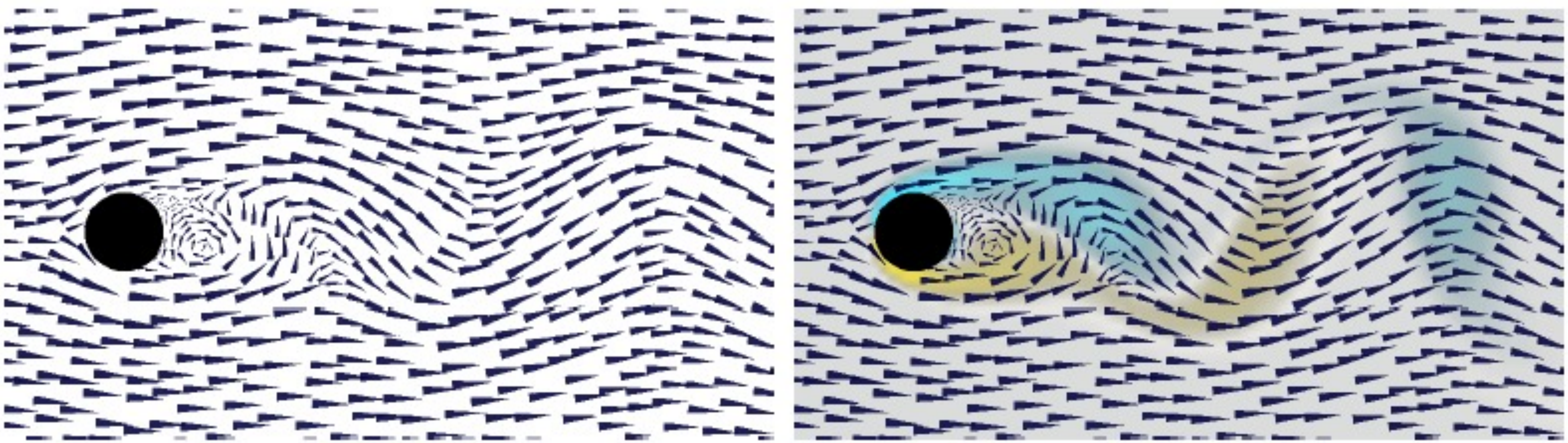
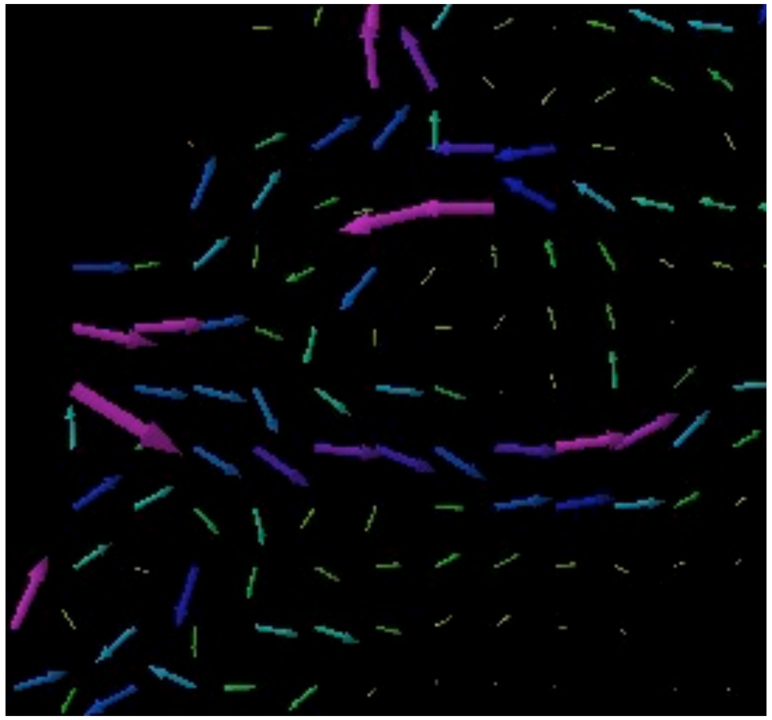
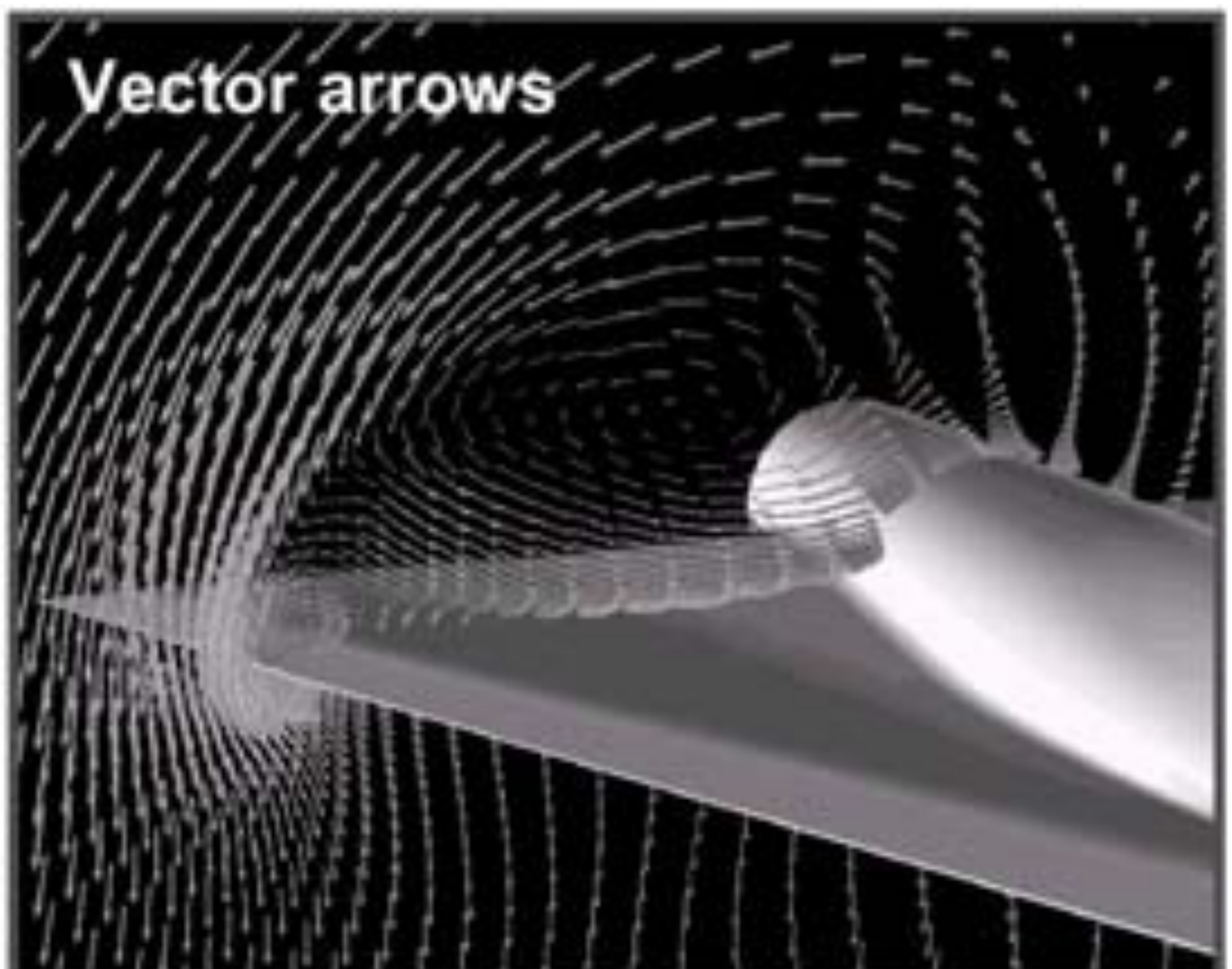
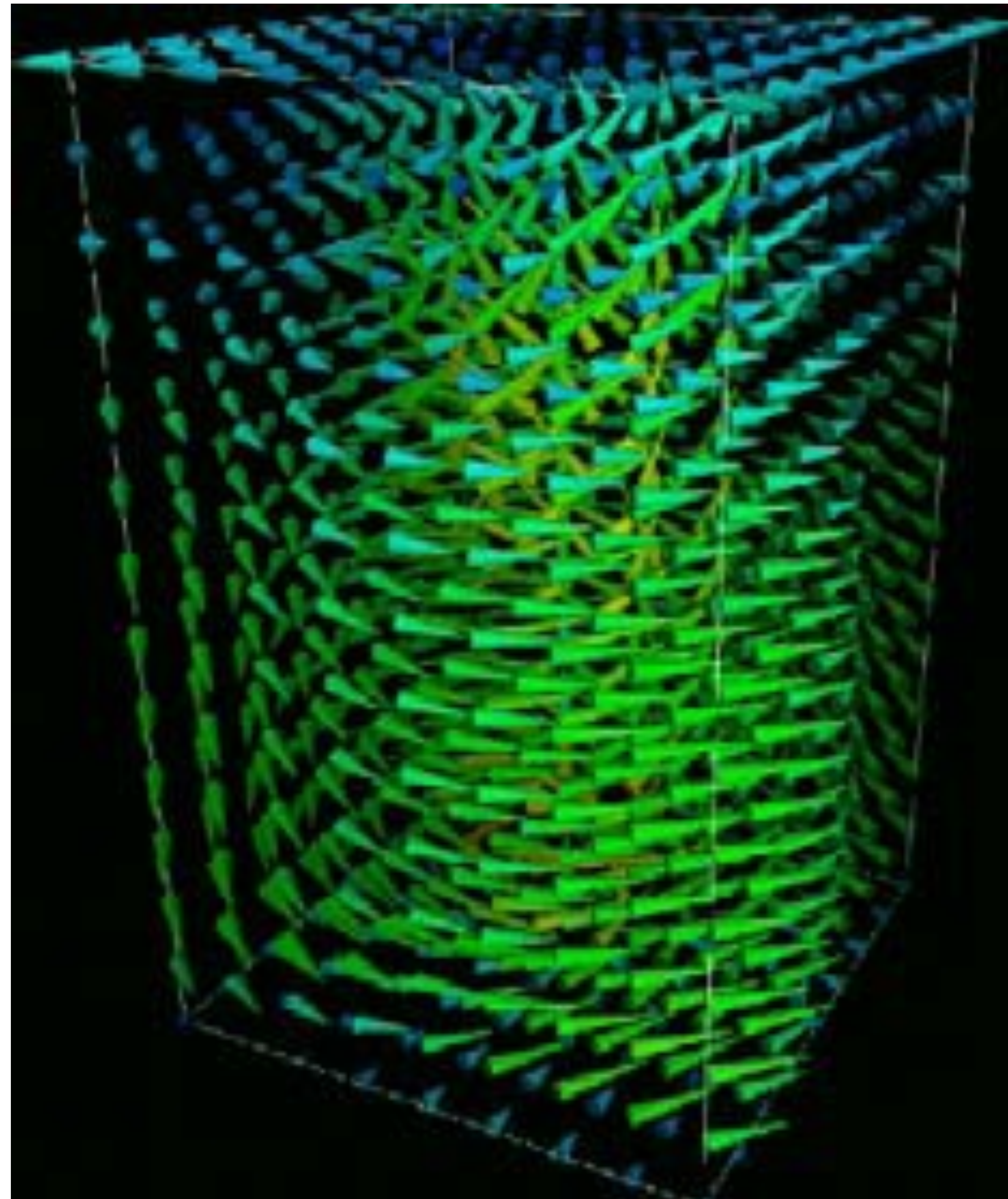


Figure 1: Typical visualization methods for 2D flow past a cylinder at Reynolds number 100. On the left, we show only the velocity field. On the right, we simultaneously show velocity and vorticity. Vorticity represents the rotational component of the flow. Clockwise vorticity is blue, counterclockwise yellow.

Arrows in 3D





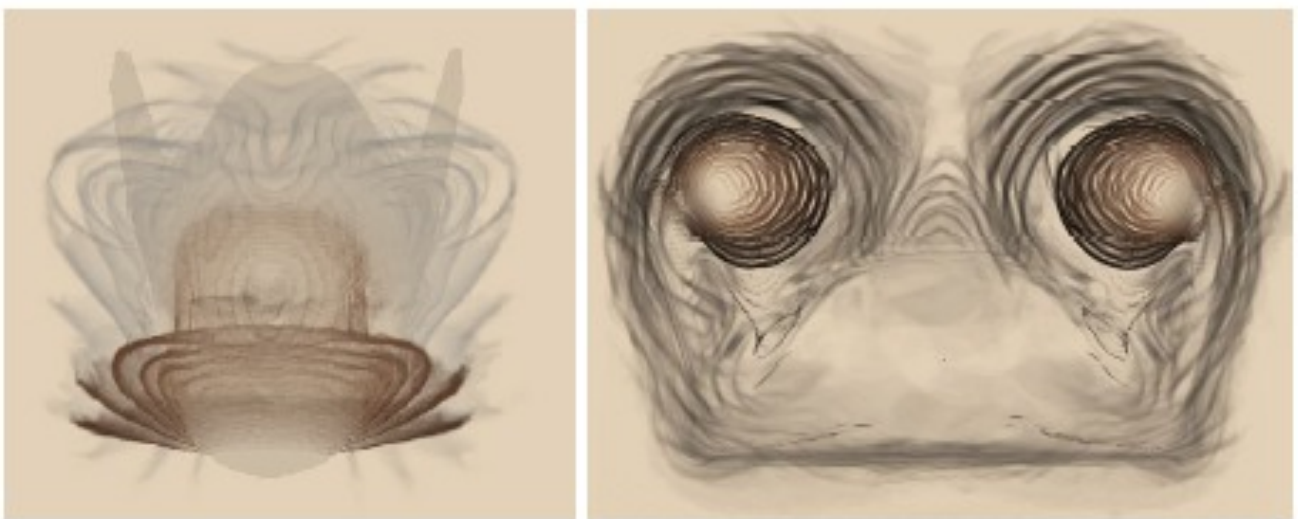


- Advantages and disadvantages of glyphs and arrows:
  - + Simple
  - + 3D effects
  - \- Inherent occlusion effects
  - \- Poor results if magnitude of velocity changes rapidly
    (Use arrows of constant length and color code magnitude)

Approaches to flow vis

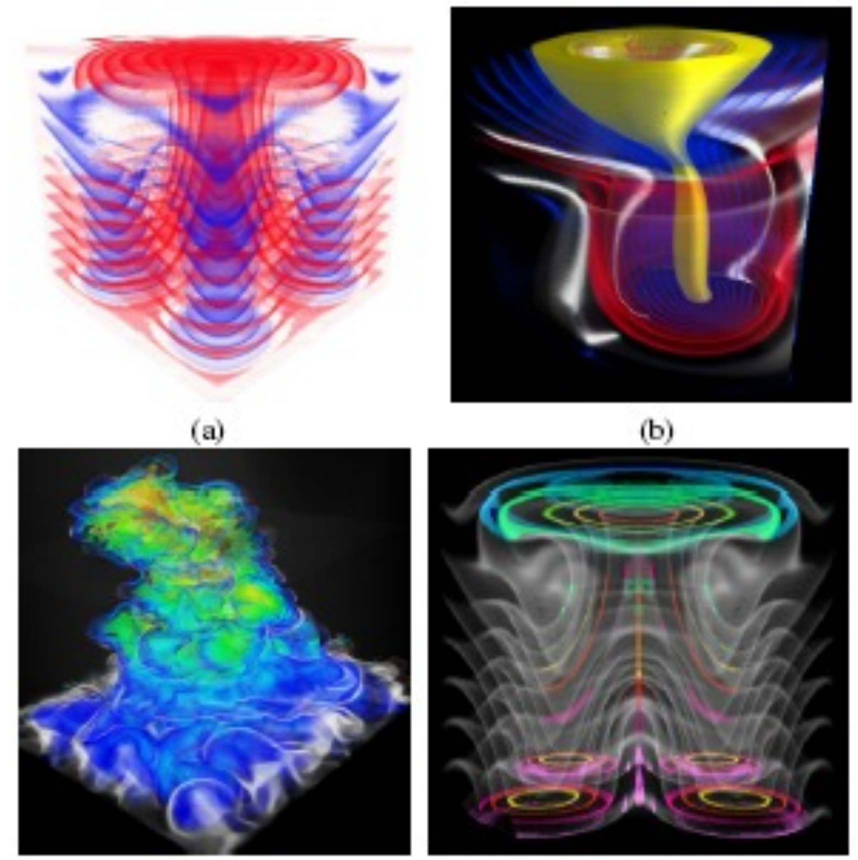
- "How?"
  - Characteristic curves of the vector field (streamlines, pathlines, streaklines, timelines, Lagrangian coherent structures / FTLE)
  - Texture-based (LIC, spot noise)
  - Direct + geometry-based (hedehogs, glyphs)
  - **Direct + heuristic (magnitude, Laplacian, FTLE)**
  - Physically-based (Schlieren imaging, virtual rheoscopic fluids)
- "Where?"
  - Flow in 2D
  - Flow on surfaces
  - Flow in 3D space

- Volume illustration for flow visualization [Svakine et al 05]



(a) (b)

Figure 3: Volume illustrations of flow around the X38 spacecraft. (a) is an illustration of density flow and shock around the bow, while (b) highlights the vortices created above the fins of the spacecraft.

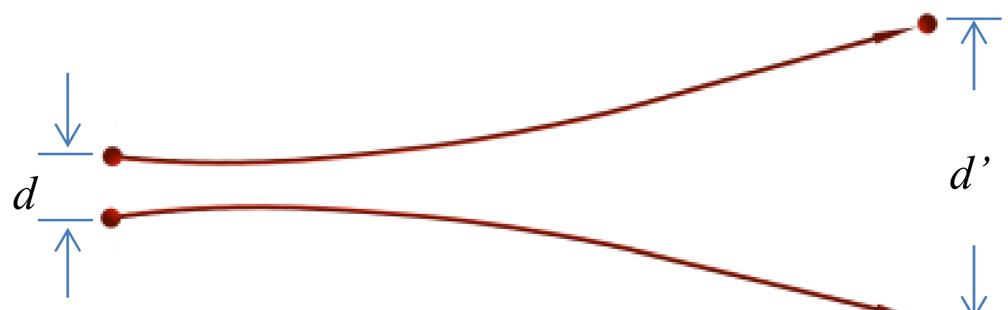


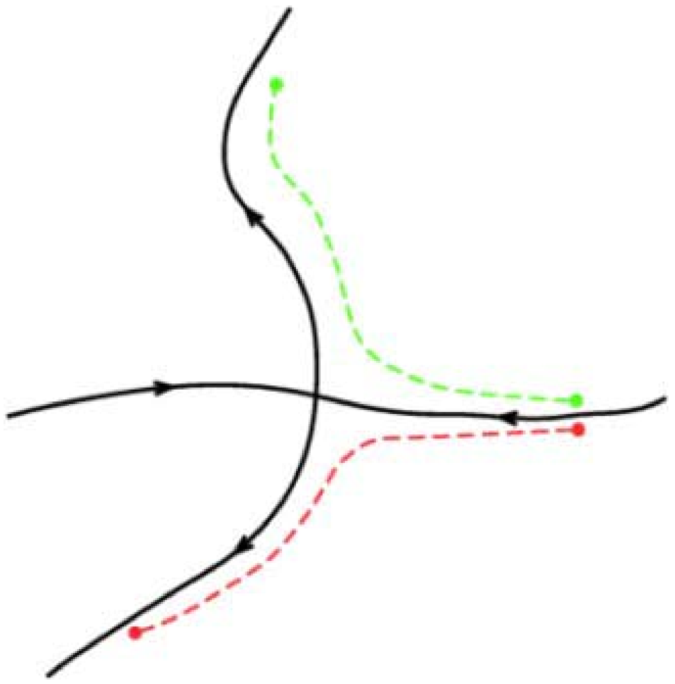
(a) (b) (c) (d)

Figure 6: Use of two-dimensional transfer function with the Laplacian operator and other flow quantities. (a) shows heat inflow (red) and outflow (blue). (b) shows all values of the Laplacian of velocity magnitude in the tornado dataset. (c) visualizes the cloud TKE using the Laplacian to highlight boundaries (white) and velocity for silhouetting. (d) highlights emerging flow structures in the convection dataset using banding of the second derivative magnitude of the temperature field.

Finite-Time Lyapunov Exponent

- Some observation
  - Observe particle trajectories
  - Measure the **divergence** between trajectories, i.e. how much flow stretch





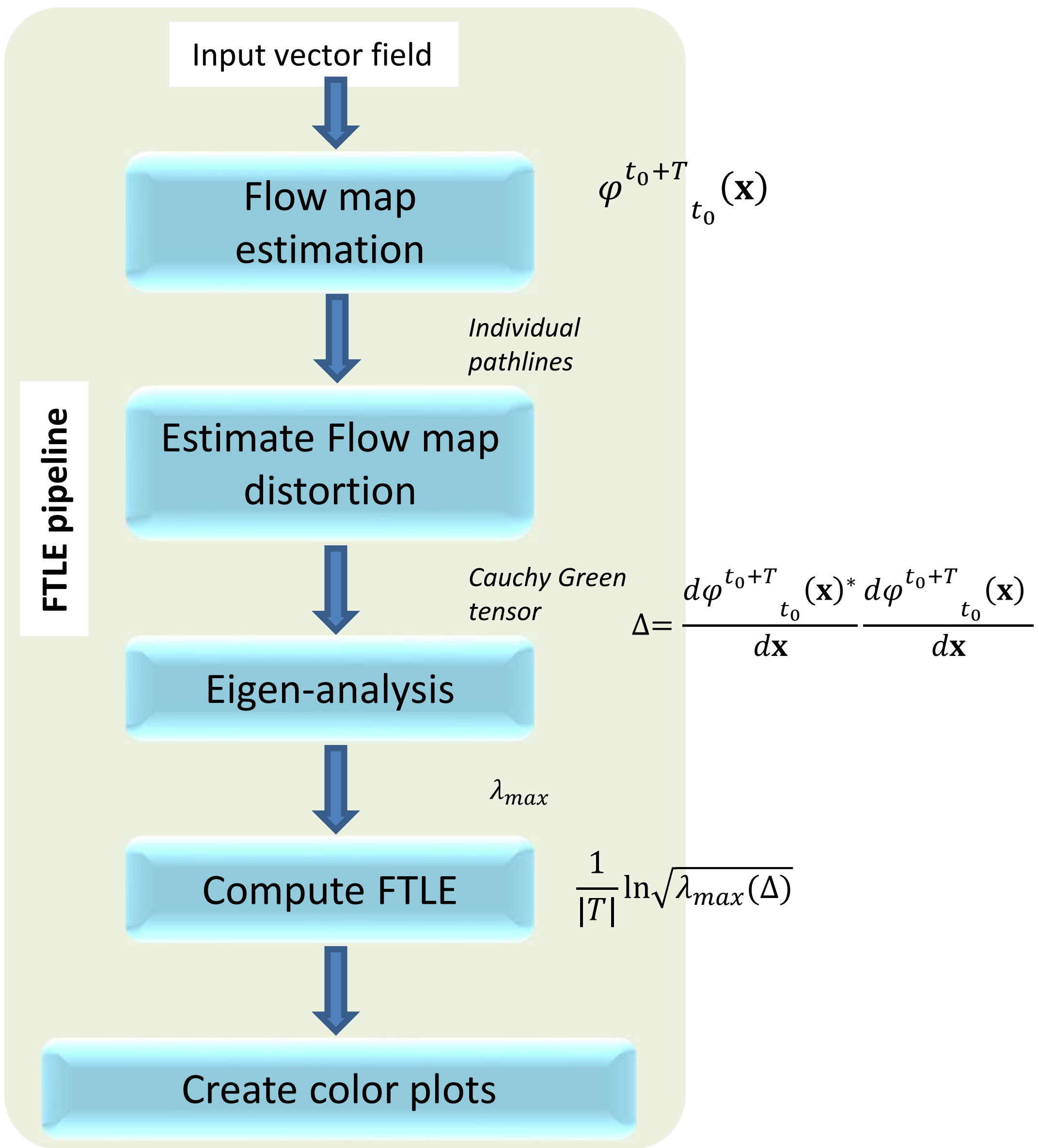
[Shadden]

Finite-Time Lyapunov Exponent

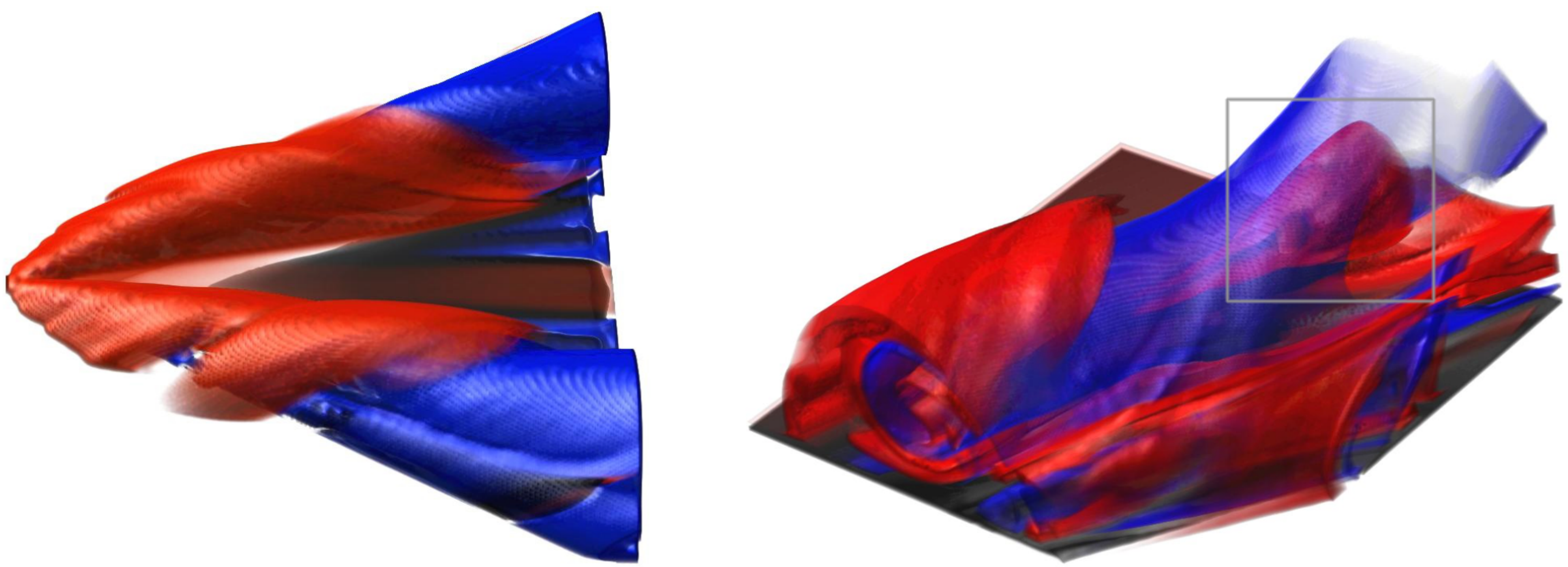
- Description
  - Lyapunov exponents describe rate of separation or stretching of two infinitesimally close points over time in a dynamical system
  - FTLE refers to the largest Lyapunov exponent for only a **limited time** and is measured **locally**
  - *Largest exponent is governing the behavior of the system, smaller ones can be neglected*
  - Ridge lines of FTLE correspond to **"Lagrangian Coherent Structures" (LCS)**
  - **i.e., sources and sinks**

Finite-Time Lyapunov Exponent

A computation framework

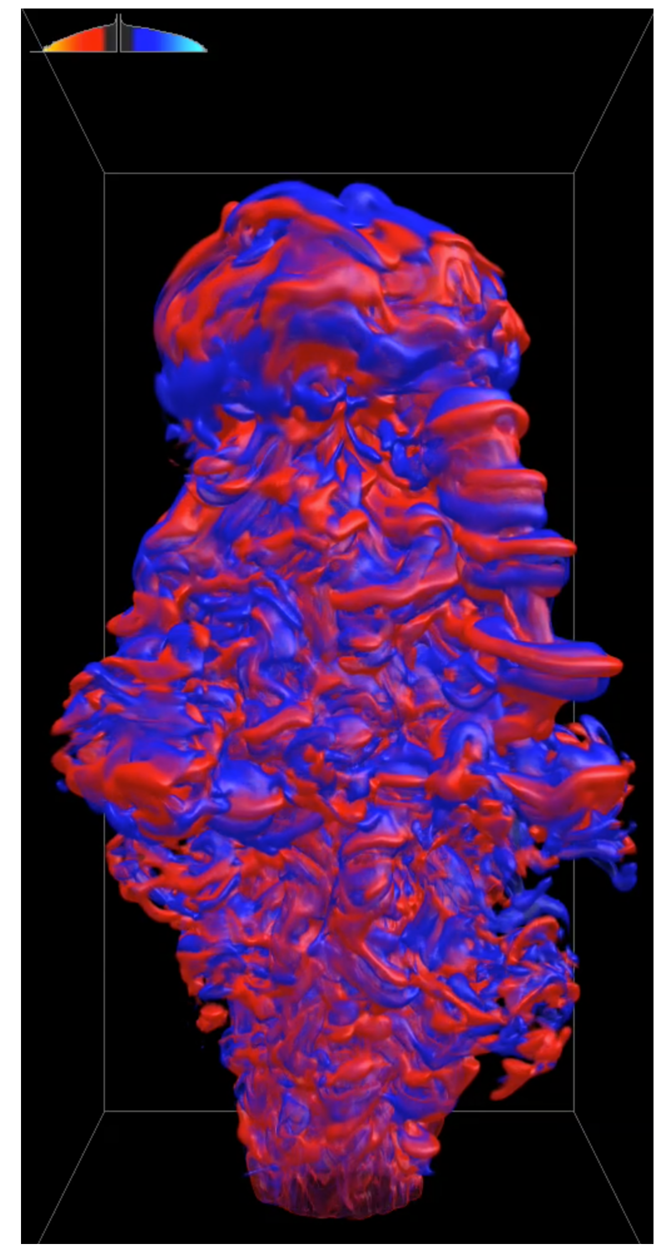


FTLE volumes - sources and sinks



Efficient Computation and Visualization of Coherent Structures in Fluid Flow Applications

C Garth, F Gerhardt, X Tricoche, H Hagen. IEEE Visualization 2007.

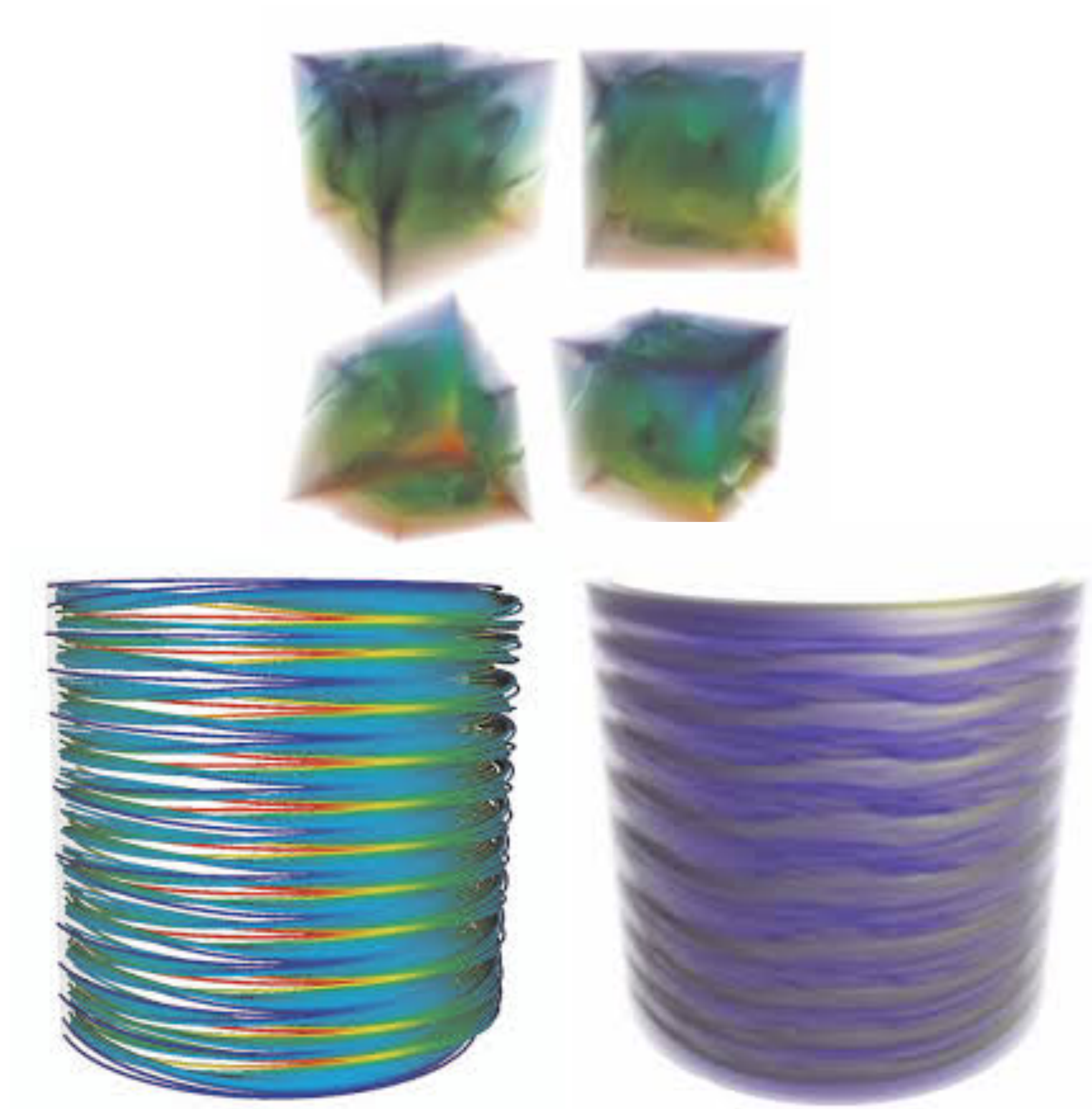


http://www.vacet.org/gallery/images_video/jet4-ftle-0.012.mp4

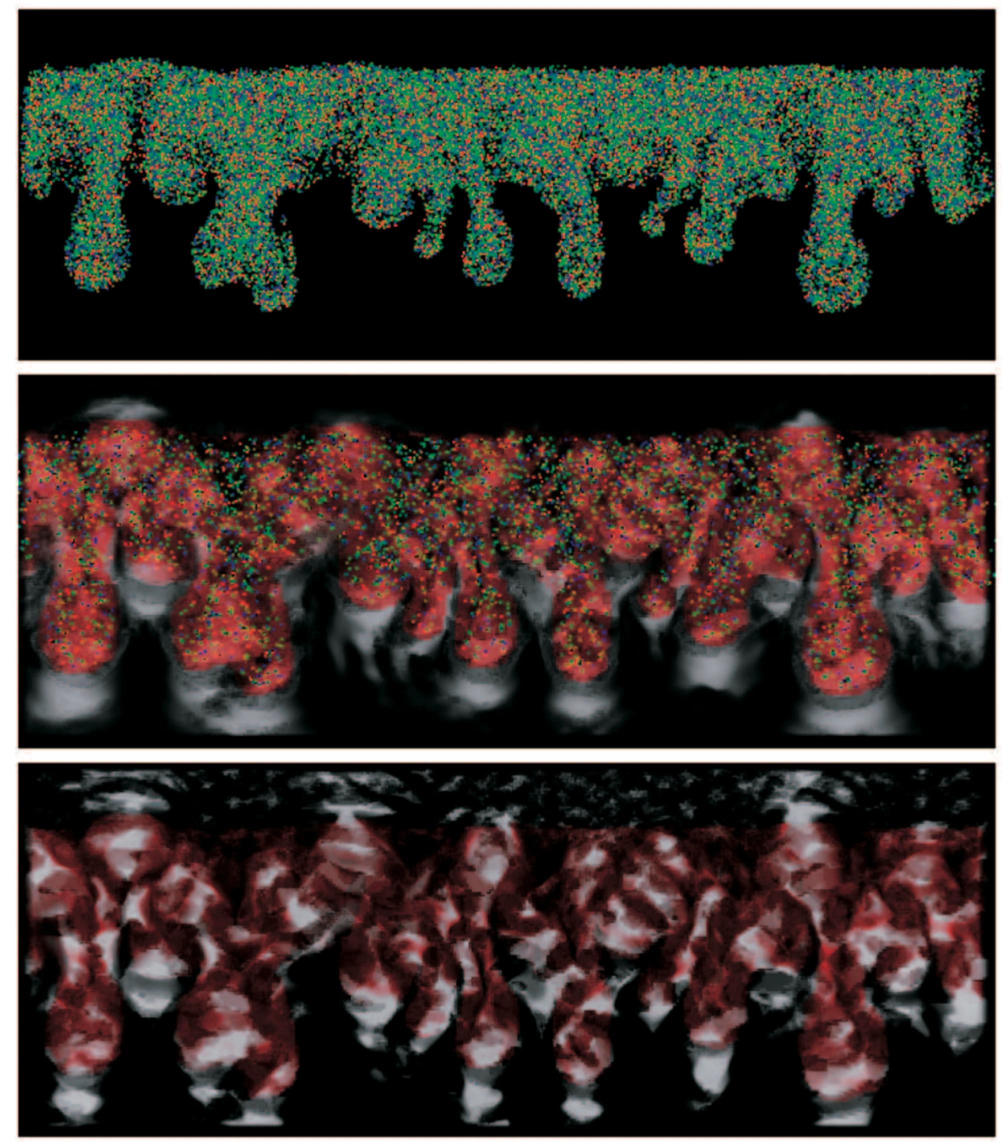
Approaches to flow vis

- "How?"
  - Characteristic curves of the vector field (streamlines, pathlines, streaklines, timelines, Lagrangian coherent structures / FTLE)
  - Texture-based (LIC, spot noise)
  - Direct geometry-based (hedehogs, glyphs)
  - Direct heuristic (magnitude, Laplacian, FTLE)
  - **Physically-based (Schlieren imaging, virtual rheoscopic fluids)**
- "Where?"
  - Flow in 2D
  - Flow on surfaces
  - Flow in 3D space

Virtual Rheoscopic Fluids



Barth et al. Virtual Rheoscopic Fluids for Flow Visualization, IEEE Vis 2007

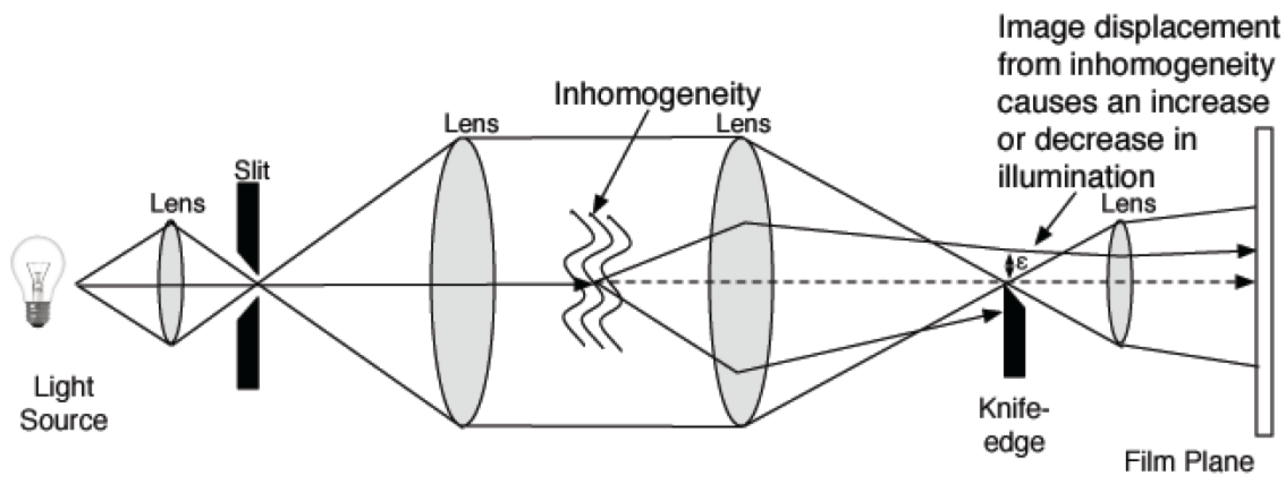


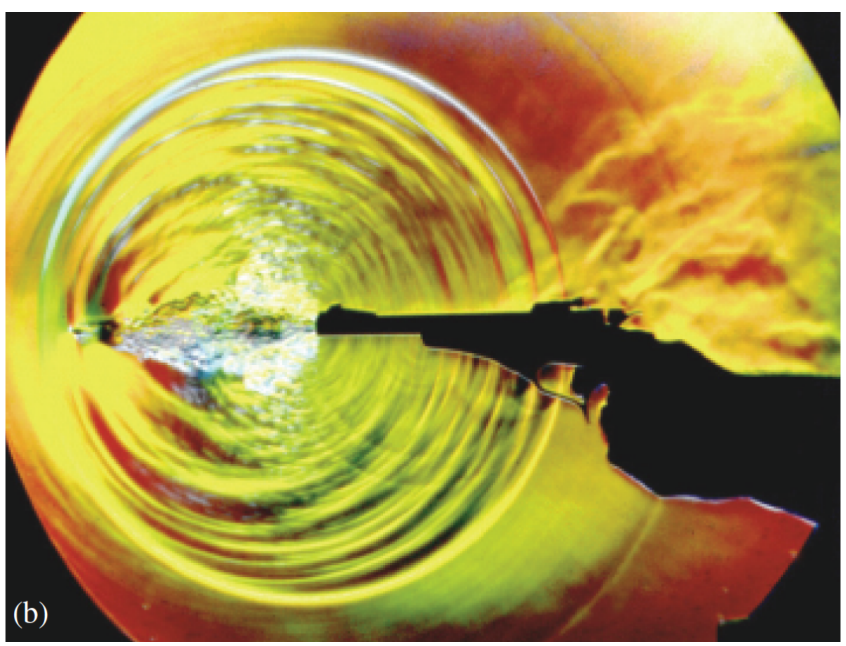
Hecht et al. Virtual Rheoscopic Fluids, IEEE Vis 2008

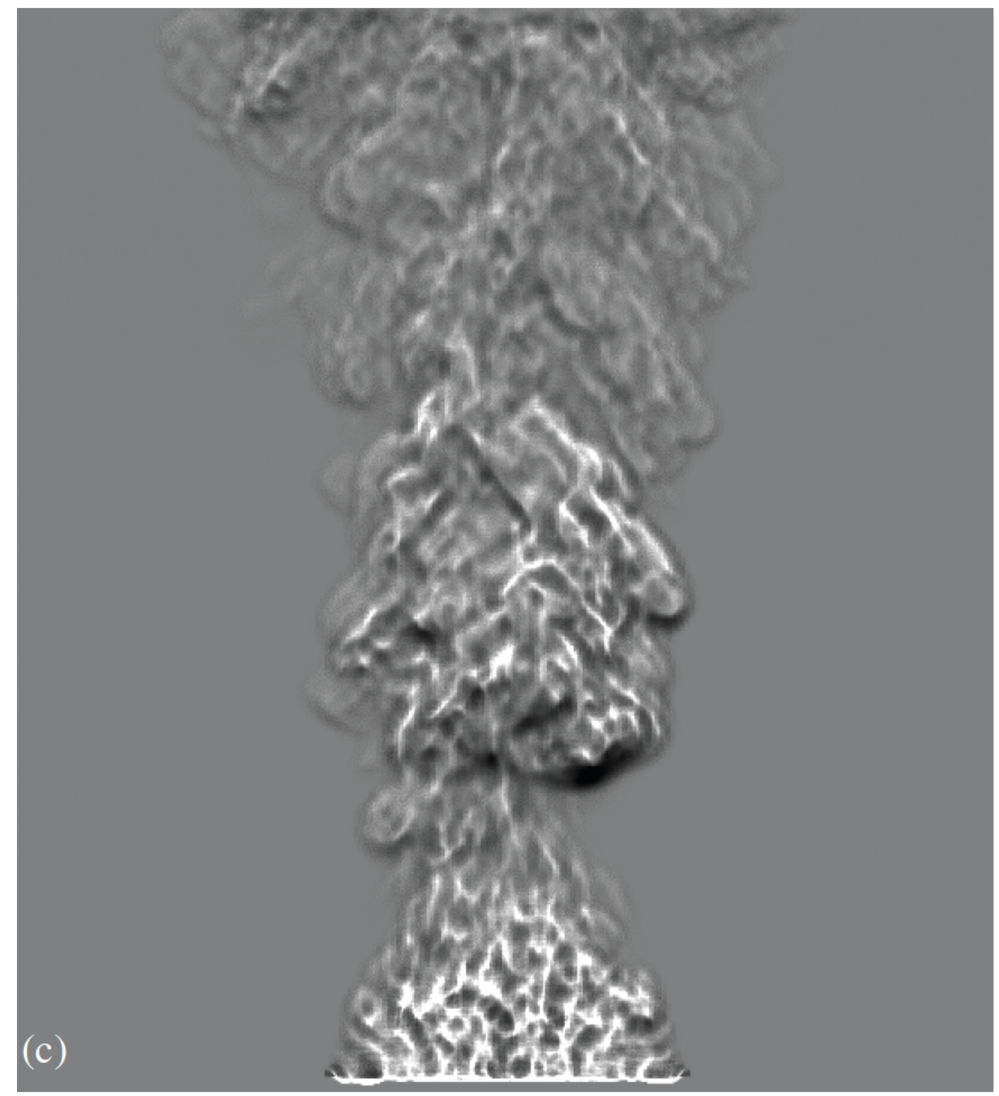
- Simulates the orientation of virtual microscope gold plate particles swimming in the vector field.
- Determine rheoscopic particle orientation via eigenvalues of the Jacobian (gradient tensor)

www.sci.utah.edu

Schlieren imaging







- Not *really* vector field visualization… but can show similar effects
- Uses precomputed index of refraction, and physically-based light transport (path tracing) to illustrate flow
- Brownlee et al. Physically-Based Interactive Schlieren Flow Visualization. IEEE Pacific Visualization 2010.

Tensor Field Visualization

scalar field

$$s : \mathbb{E}^n \rightarrow \mathbb{R}$$

$$s(\mathbf{x})$$

with $\mathbf{x} \in \mathbb{E}^n$

vector field

$$\mathbf{v} : \mathbb{E}^n \rightarrow \mathbb{R}^m$$

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} c_1(\mathbf{x}) \\ \vdots \\ c_m(\mathbf{x}) \end{pmatrix}$$

with $\mathbf{x} \in \mathbb{E}^n$

tensor field

$$\mathbf{T} : \mathbb{E}^n \rightarrow \mathbb{R}^{m \times b}$$

$$\mathbf{T}(\mathbf{x}) = \begin{pmatrix} c_{11}(\mathbf{x}) & \dots & c_{1b}(\mathbf{x}) \\ \vdots & & \vdots \\ c_{m1}(\mathbf{x}) & \dots & c_{mb}(\mathbf{x}) \end{pmatrix}$$

with $\mathbf{x} \in \mathbb{E}^n$

- Tensor: extension of concept of scalar and vector
- Tensor data: for a tensor of level k is given by $t_{i1,i2,\ldots,ik}(x_1,\ldots,x_n)$
- Second-order tensor often represented by matrix
- Examples:
  - Diffusion tensor (from medical imaging, see later)
  - Material properties (material sciences):
    - Conductivity tensor
    - Dielectric susceptibility
    - Magnetic permittivity
    - Stress tensor

Definition

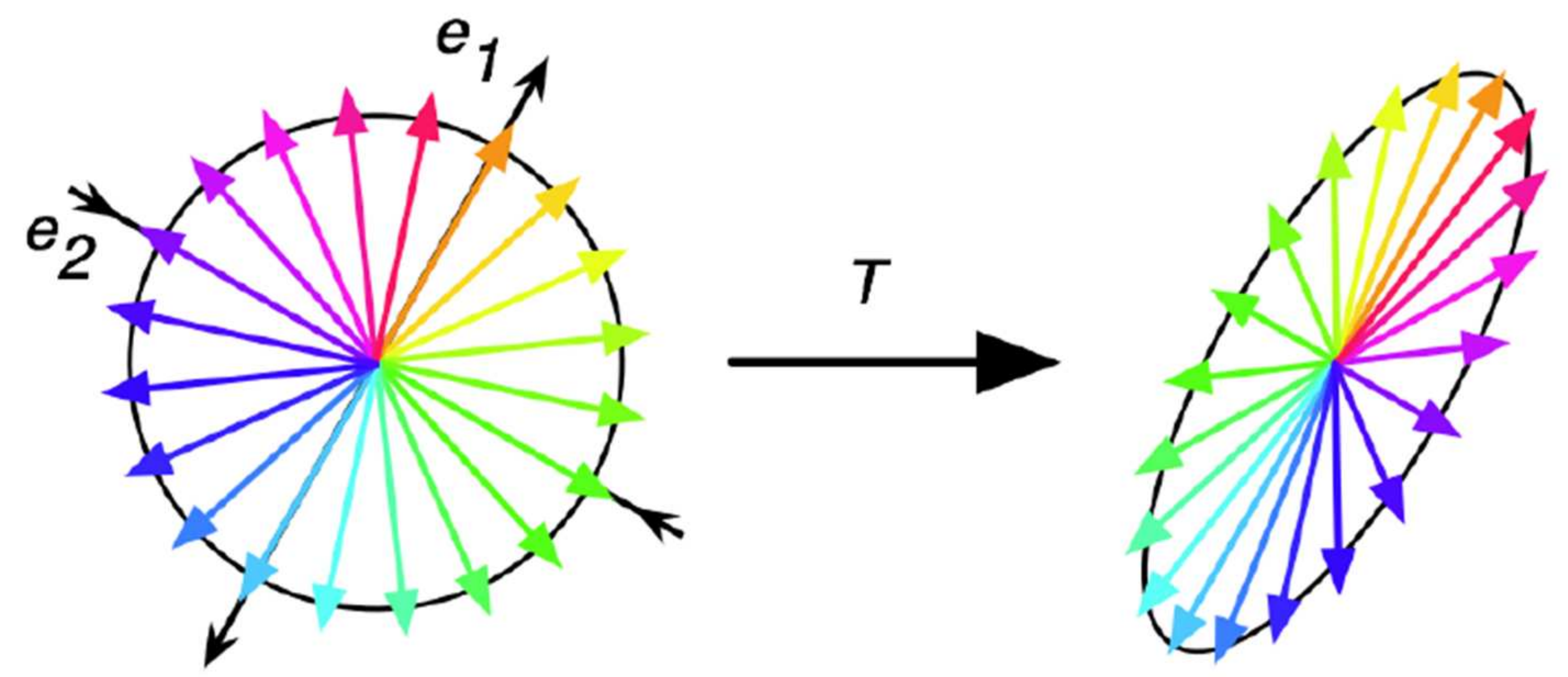
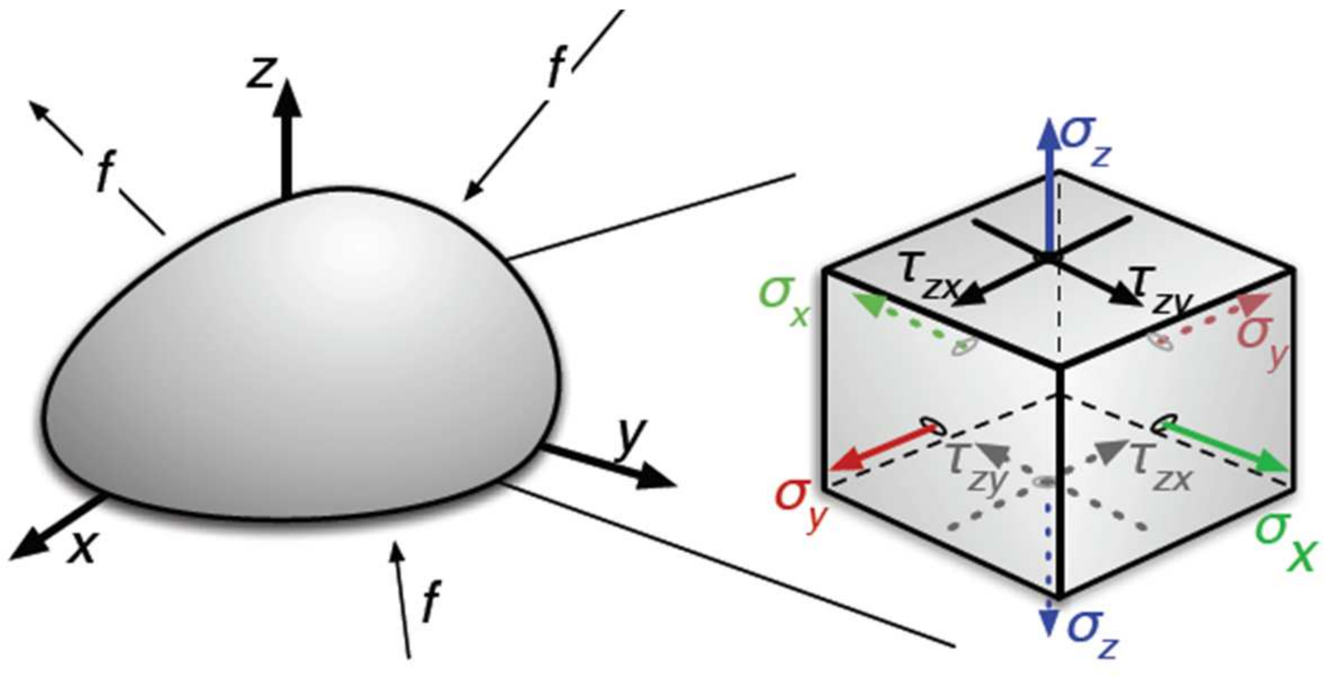


Illustration of a symmetric second-order tensor as linear operator. The tensor is uniquely determined by its action on all unit vectors, represented by the circle in the left image. The eigenvector directions are highlighted as black arrows. In this example one eigenvalue (l2) is negative. As a consequence all vectors are mirrored at the axis spanned by eigenvector e1. The eigenvectors are the directions with strongest normal deformation but no directional change.

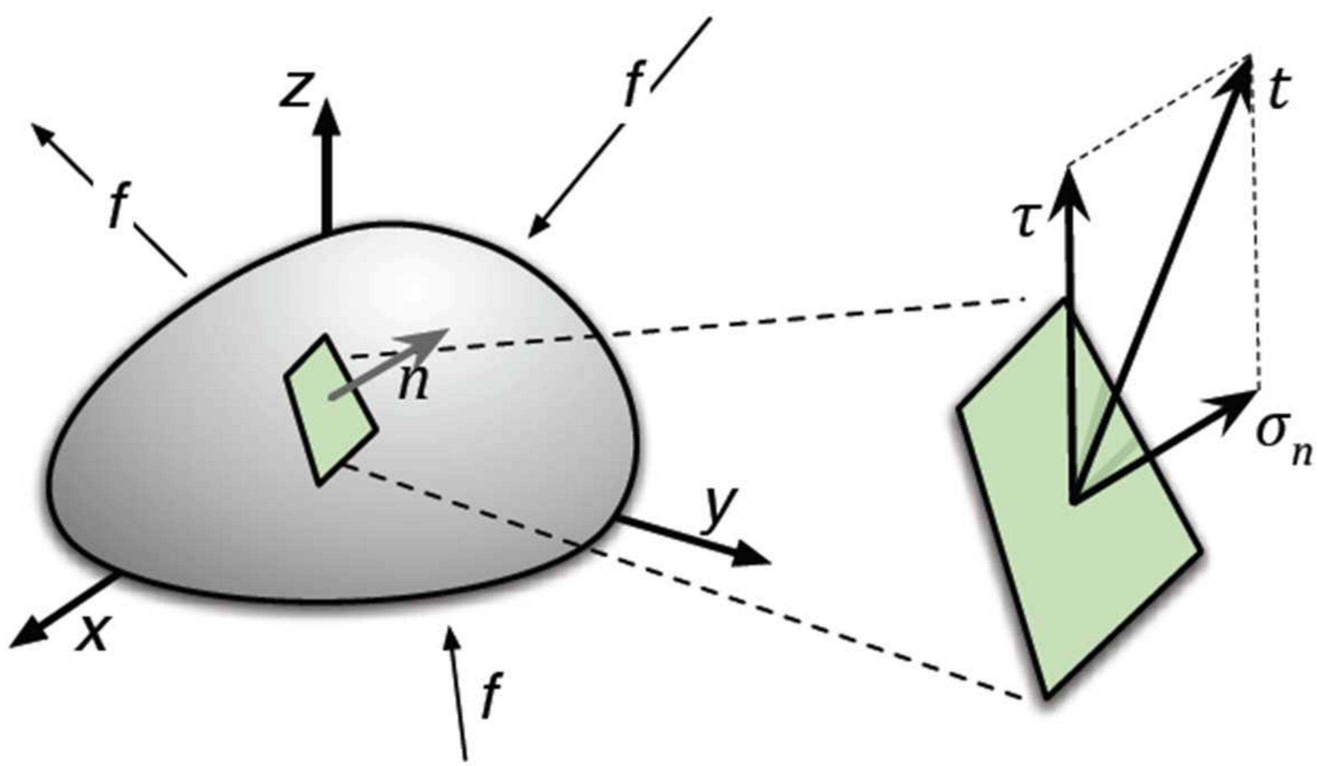
Applications

- Tensors describe entities that scalars and vectors cannot describe sufficiently, for example, the stress at a point in a continuous medium under load.
  - medicine,
  - geology,
  - astrophysics,
  - continuum mechanics
  - and many more

Tensors in Mechanical Engineering



(a)



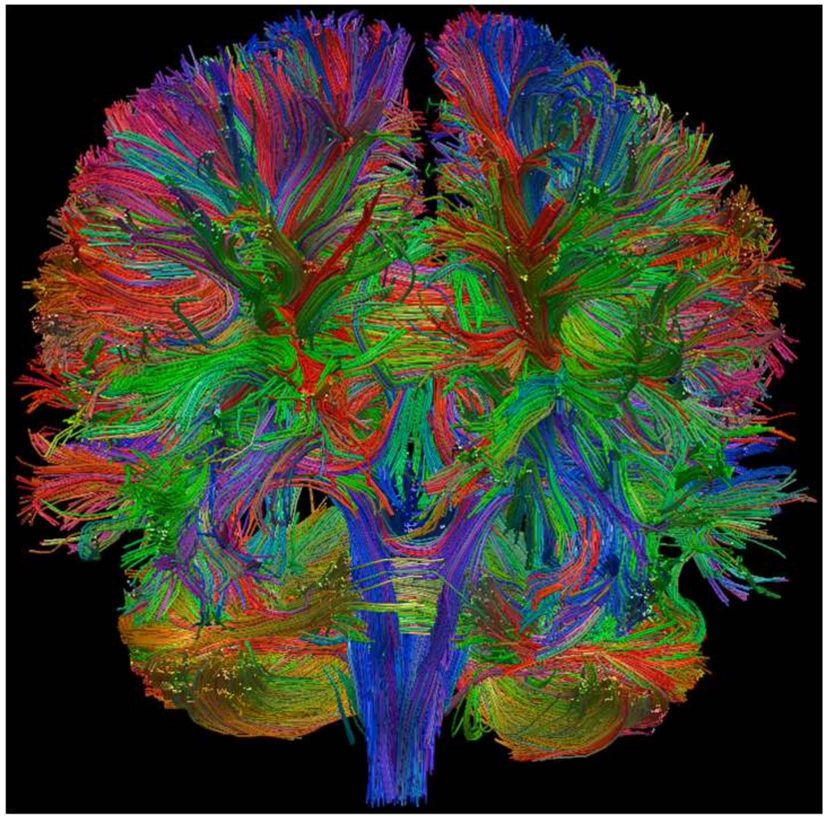
Stress tensors describe internal forces or stresses that act within deformable bodies as reaction to external forces

(a) External forces f are applied to a deformable body. Reacting forces are described by a three-dimensional stress tensor that is composed of three normal stresses *s* and three shear stresses τ.

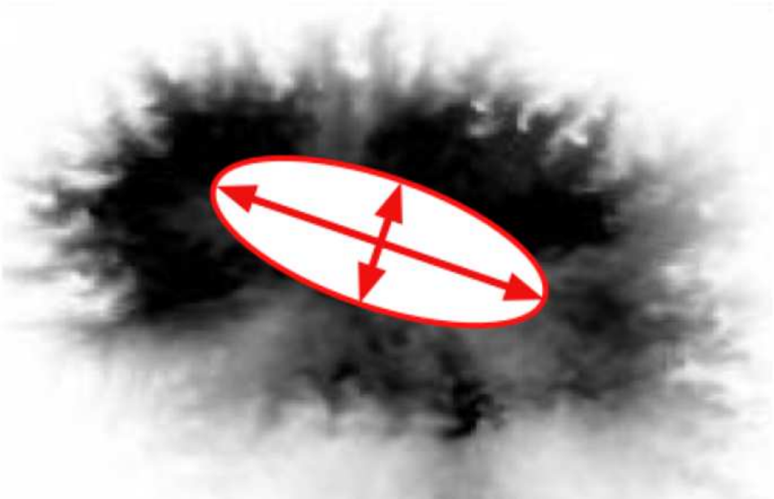
(b) Given a surface normal n of some cutting plane, the stress tensor maps *n* to the traction vector *t*, which describes the internal forces that act on this plane (normal and shear stresses).

- Typical second-order tensor: diffusion tensor
  - Diffusion: based on motion of fluid particles on microscopic level
  - Probabilistic phenomenon
  - Based on particle's Brownian motion
  - Measurements by modern MR (magnetic resonance) scanners
  - Diffusion tensor describes diffusion rate into different directions via symmetric tensor (probability density distribution)
  - In 3D: representation via 3×3 symmetric matrix

Diffusion Tensor Imaging (DTI)



- For medical applications, diffusion tensors describe the anisotropic diffusion behavior of water molecules in tissue.
- Here, the molecule motion is driven by the Brownian motion and not the concentration gradient.
- The tensor contains the following information about the diffusion: its strength depending on the direction and its anisotropy
- It is positive semi-definite and symmetric.

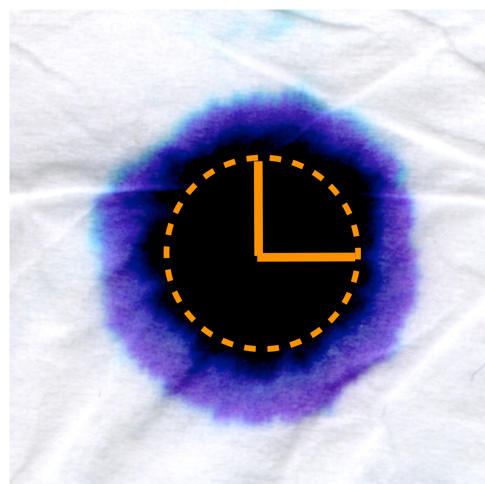


Note that in practice the positive definiteness of diffusion tensors can be violated due to measurement noise.

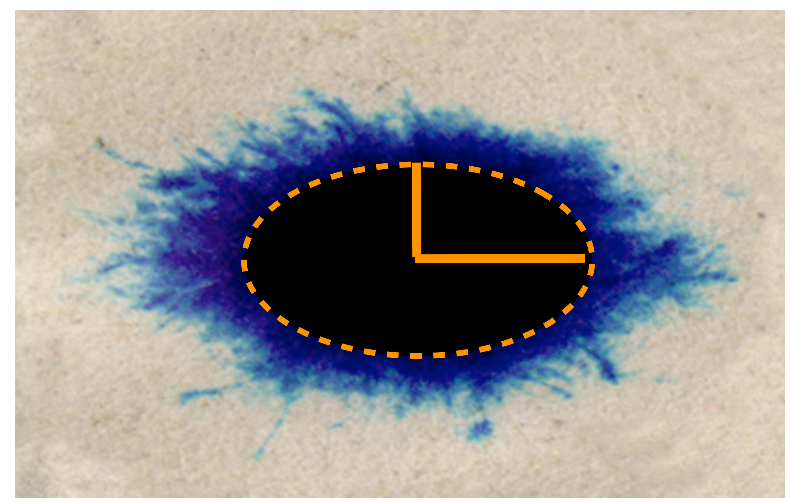
In the human brain, DTI measures movement of (mostly water) within neural axons of white matter. I.e., a method for tractography.

Diffusion in Biological Tissue

- Motion of water through tissue
- Sometimes faster in some directions than others

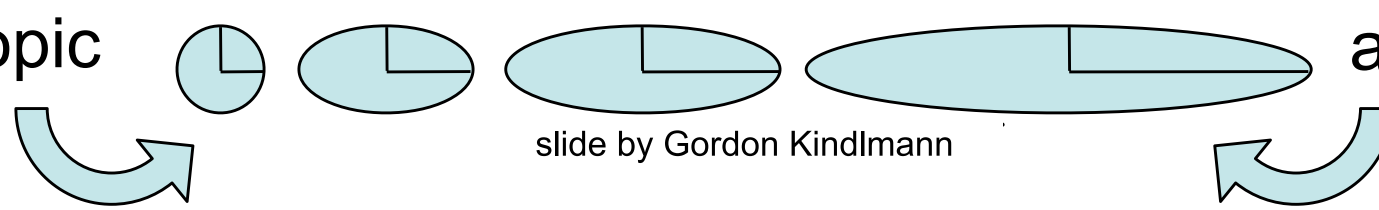


Kleenex



newspaper

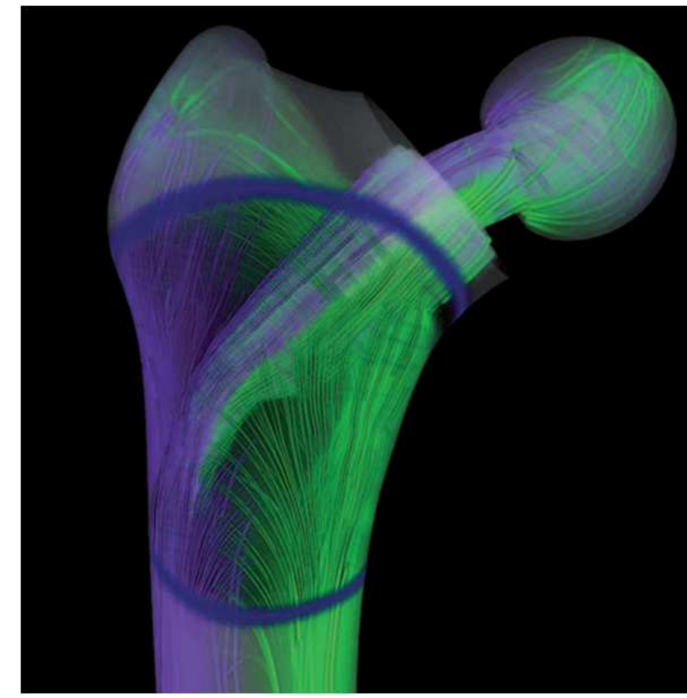
Anisotropy: diffusion rate depends on direction

isotropic  anisotropic

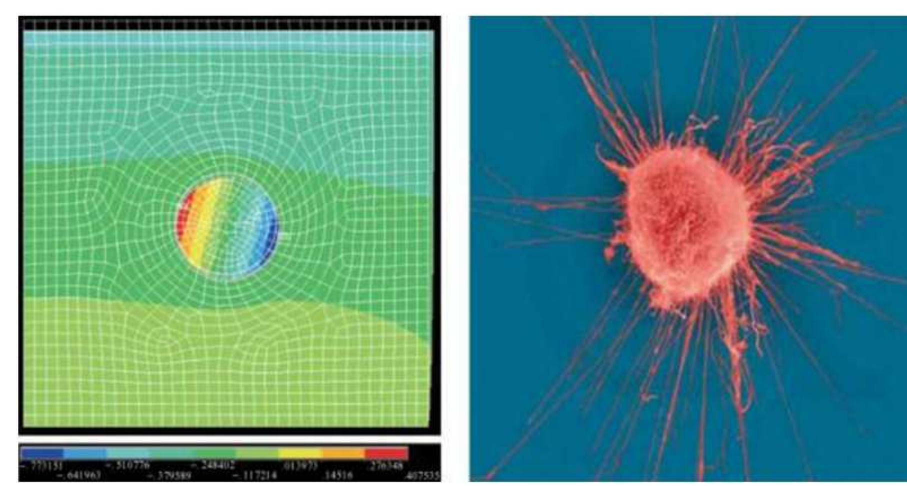
slide by Gordon Kindlmann

Tensors in Medicine

- Diffusion tensors are not the only type of tensor that occur in the medical context.
- In the context of implant design, stress tensors result from simulations of an implant’s impact on the distribution of physiological stress inside a bone.
- An application related to strain tensors is used in elastography where MRI, CT or ultrasound is used to measure elastic properties of soft tissues. Changes in the elastic properties of tissues can be an important hint to cancer or other diseases



[DICK et al. Vis09]

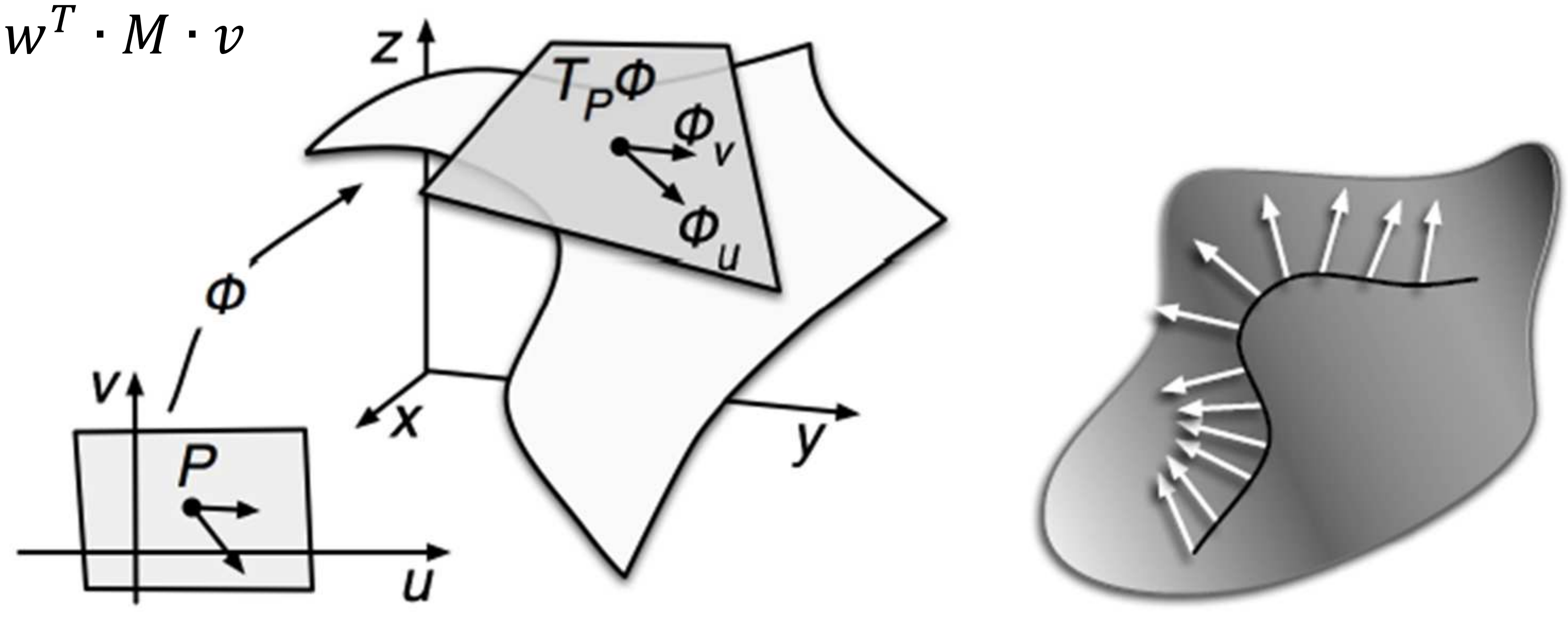


[SOSA-CABRERA et al., 2009]

Tensors in Geometry

- Curvature tensors - change of surface normal in any given direction
- Metric tensors - relates a direction to distances and angles; defines how angles and the lengths of vectors are measured independently of the chosen reference frame

$$T(v,w) = w^T \cdot M \cdot v$$



(a) (b)

Tensor Diagonalization

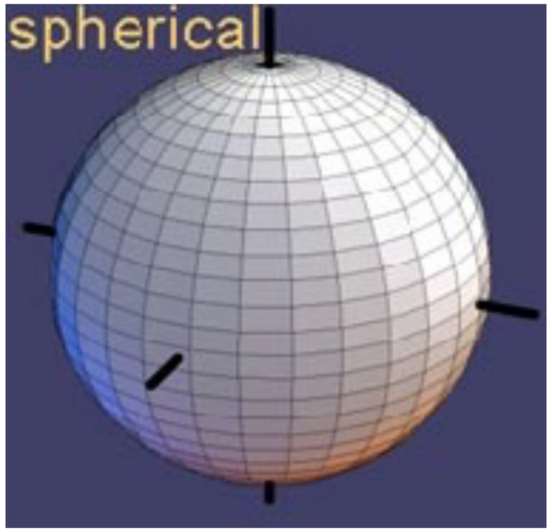
- The tensor representation becomes especially simple if it can be diagonalized.
- The complete transformation of T from an arbitrary basis into the eigenvector basis, is given by

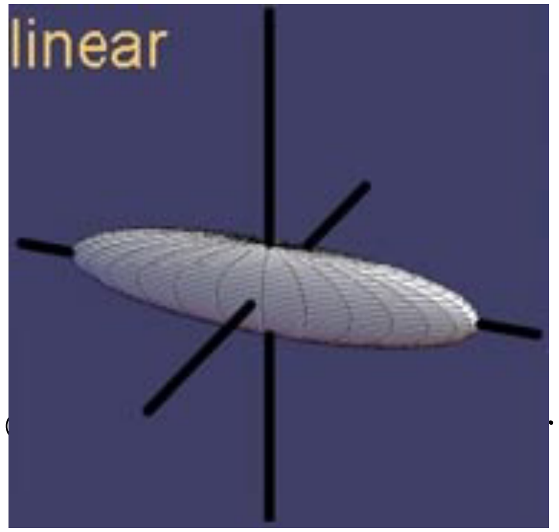
$$UTU^T = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

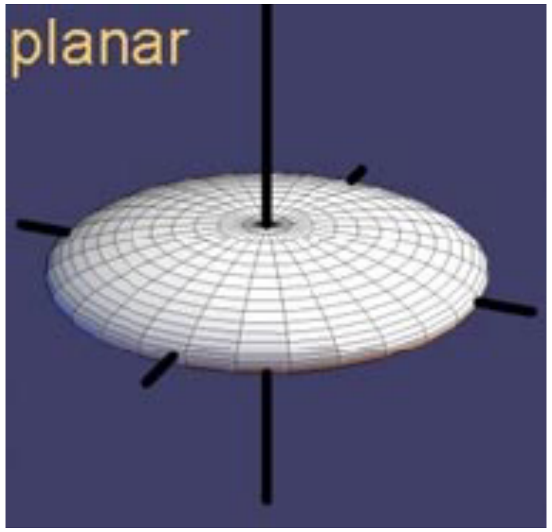
- The diagonal elements λ_1, λ_2, λ_3 are the eigenvalues and U is the orthogonal matrix that is composed of the eigenvectors, that is (e_1, e_2, e_3)

- The diagonalization generally is computed numerically via singular value decomposition (SVD) or principal component analysis (PCA).

- Symmetric diffusion matrix can be diagonalized:
  - Real eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$
  - Eigenvectors are perpendicular
- Isotropy / anisotropy:
  - Spherical: $\lambda_1 = \lambda_2 = \lambda_3$
  - Linear: $\lambda_2 \approx \lambda_3 \approx 0$
  - Planar: $\lambda_1 \approx \lambda_2$ and $\lambda_3 \approx 0$





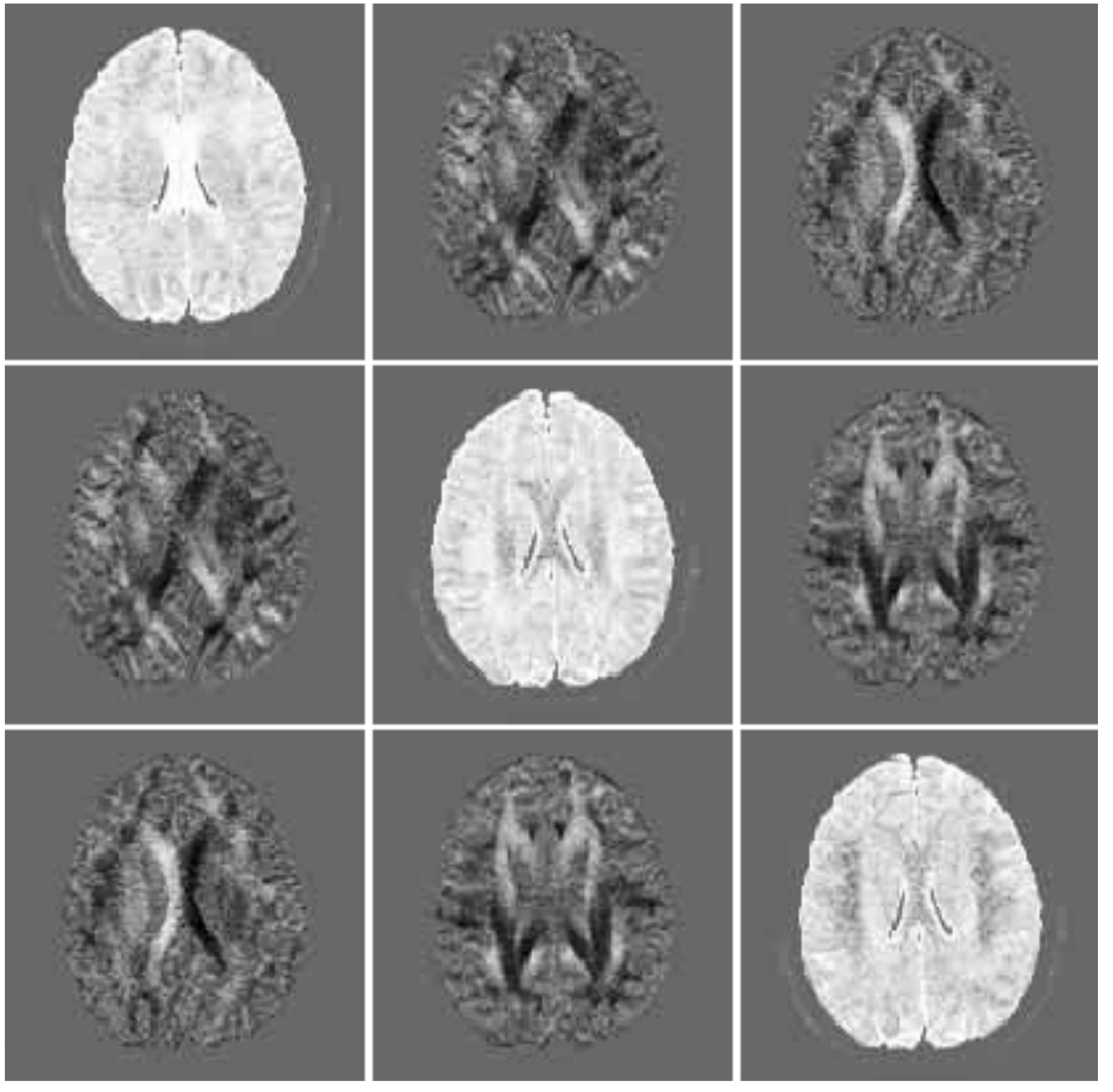


Challenges in Visualization

- Hard to achieve intuitive visualization
  - Tensors represent diverse quantities, ranging from the curvature of a surface to the diffusion of water molecules in tissue. There is no universal intuition similar to, e.g., arrows for vectors.
- Multi-variate nature makes it challenging.
  - The multi-variate nature of tensors affects all stages of the visualization pipeline, making each of them a challenging task, including interpolation, segmentation, and visualization.
- Perception issue: clutter and occlusion
- Highly application-dependent

- Direct
  - Color-coding and Glyphs
  - Hue-Balls and Lit-Tensors
- Geometry-Based
  - Hyperstreamlines, and tensorlines
- Texture-Based
  - HyperLIC
- Feature-Based
  - Segmentation, Topological Features

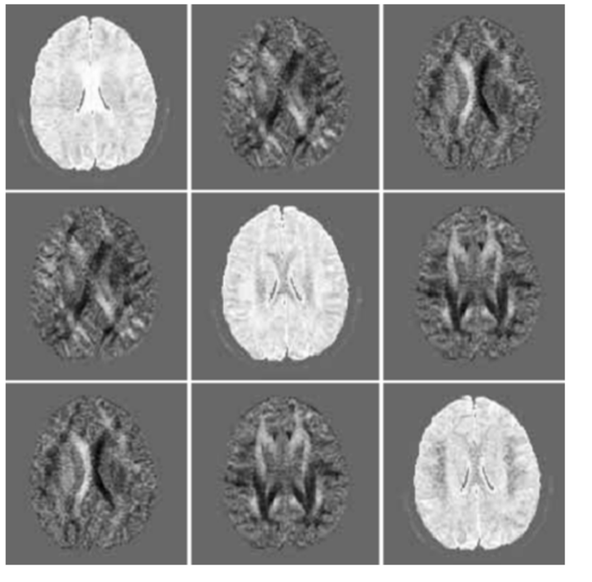
- Matrix of images
  - Slices through volume
  - Each image shows one component of the matrix



Pseudo-Colors

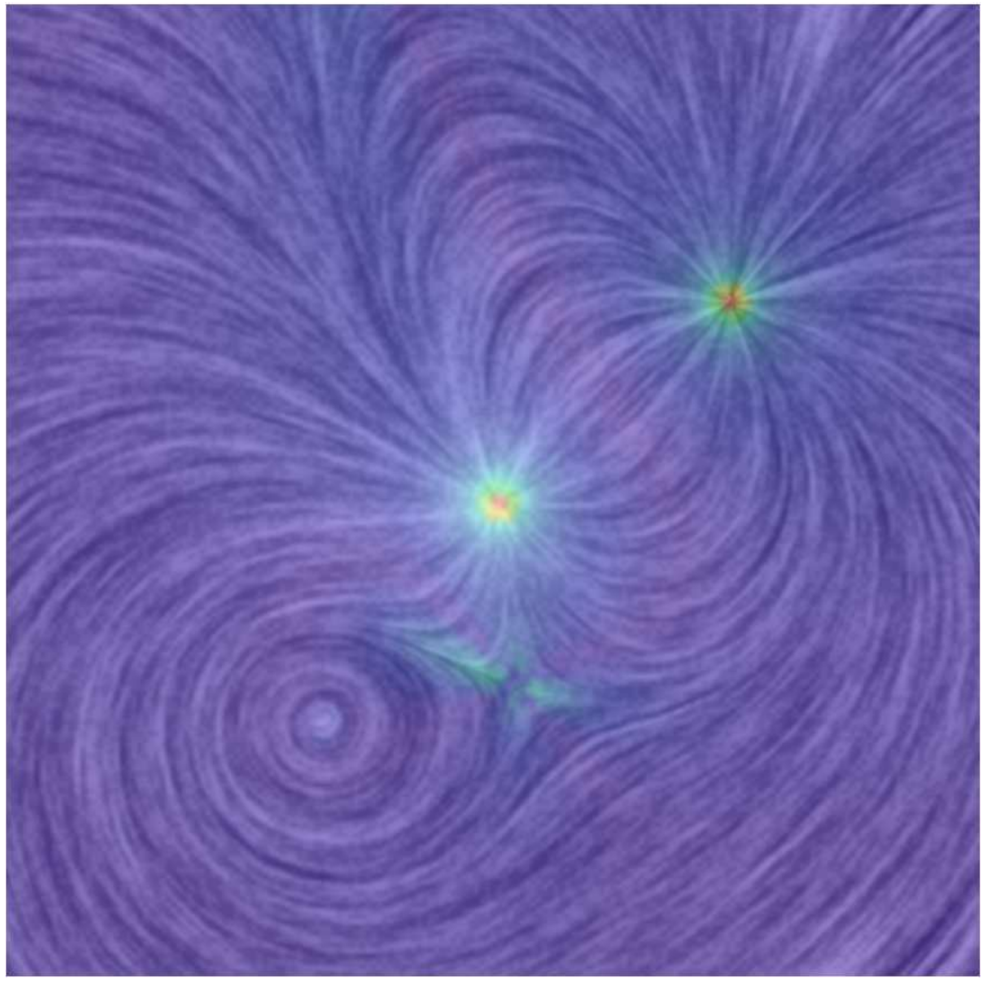
- Any derived scalar properties of the tensor can be mapped to color plots
- Assume a tensor T is defined at each vertex
  - Components (or entries) T_{ij}
  - Tensor magnitude

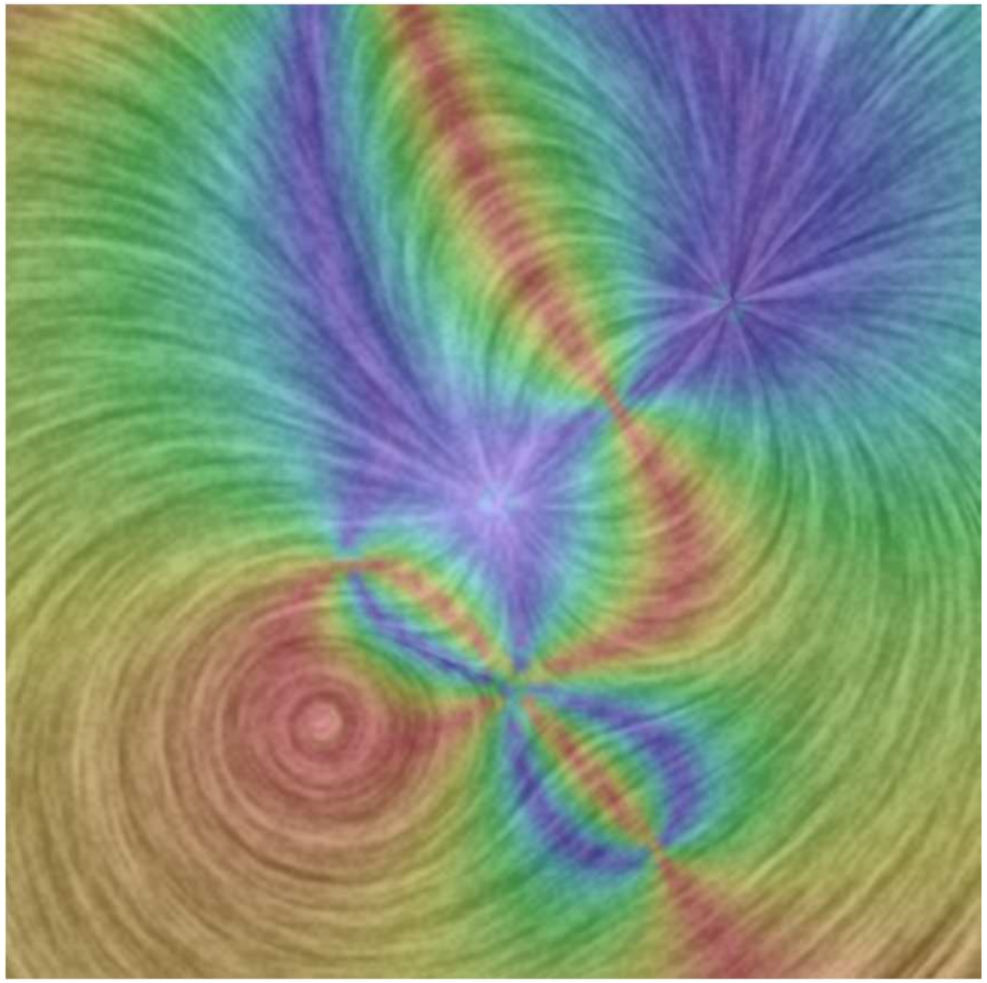
$$\|T\|_F = \sqrt{\frac{1}{2}\sum {T_{ij}}^2}$$



  - Trace, $tr(T) = \sum T_{ii}$. If T is the Jacobian of a flow field, this tells how much divergence it has.

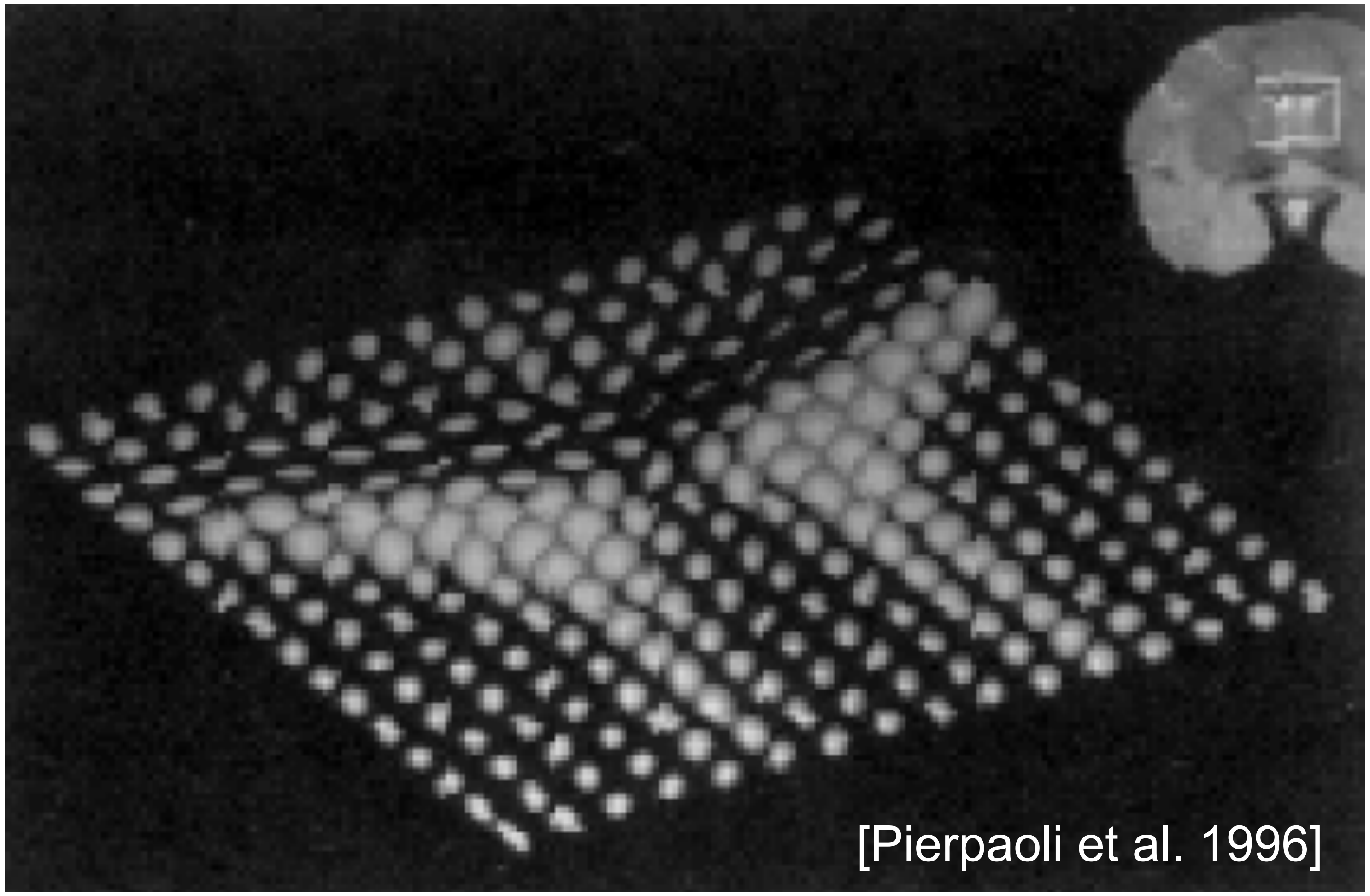
Pseudo-Colors



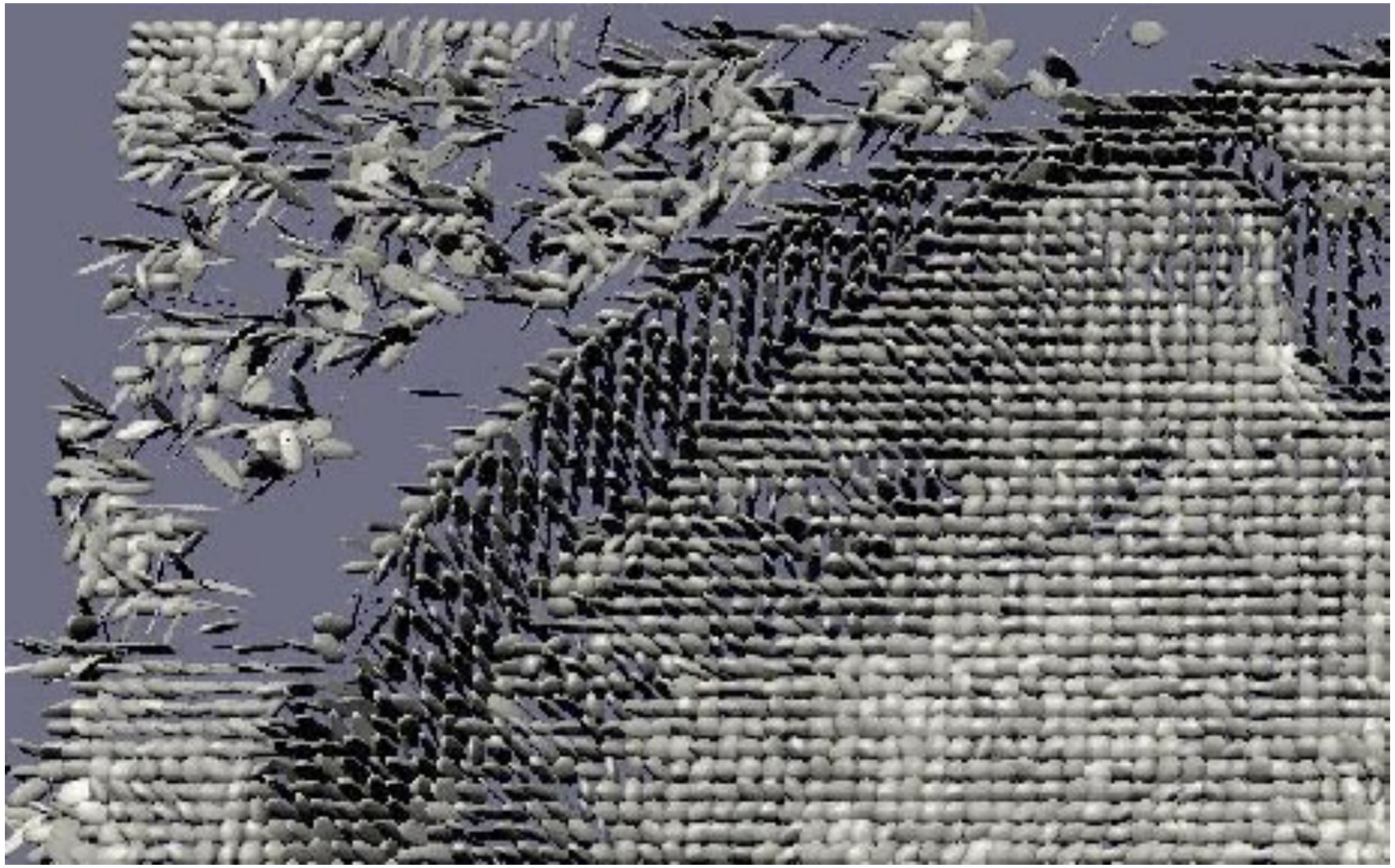


Divergence and curl of a vector field

- Uniform grid of ellipsoids
  - Second-order symmetric tensor mapped to ellipsoid
  - Sliced volume

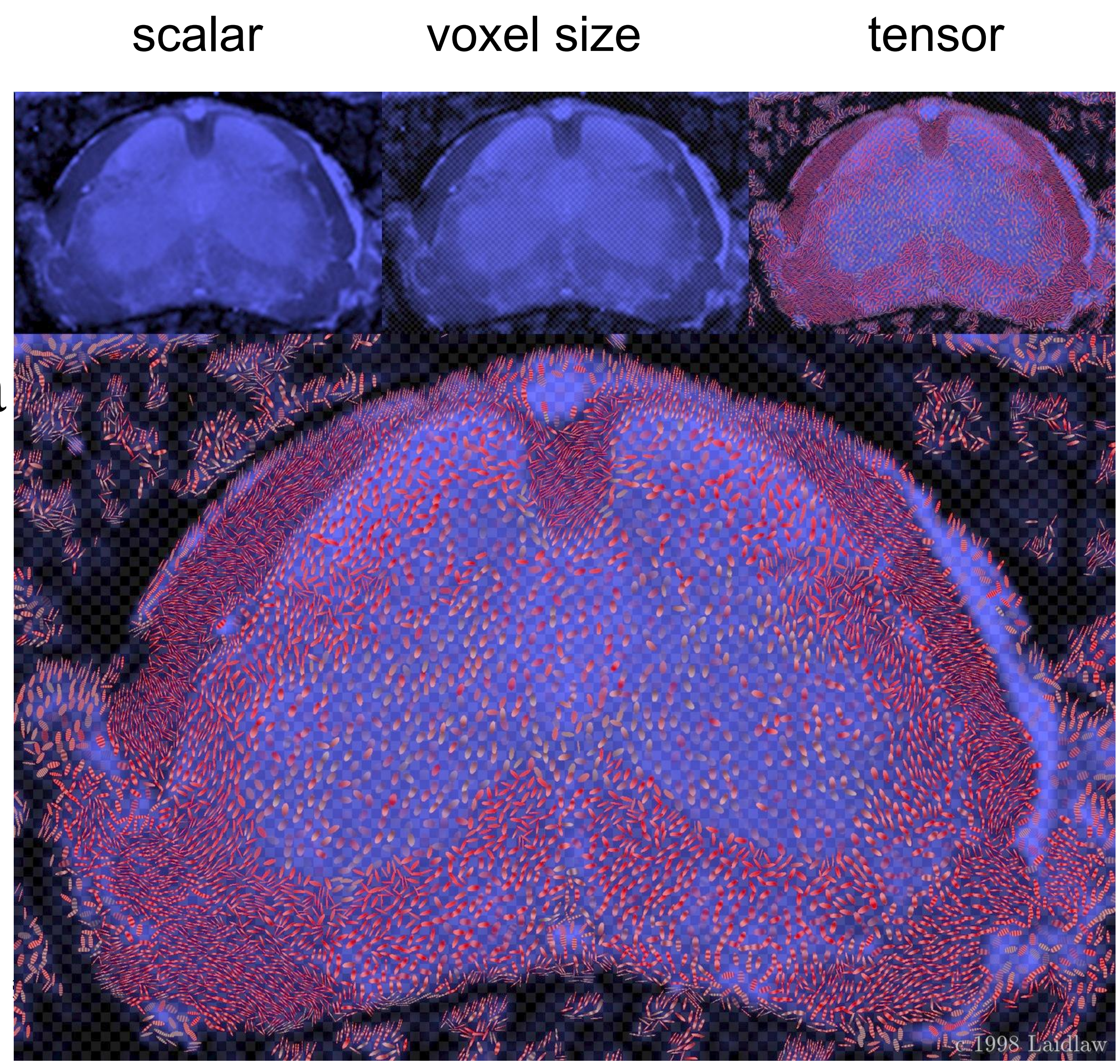


- Uniform grid of ellipsoids
  - Normalized sizes of the ellipsoids

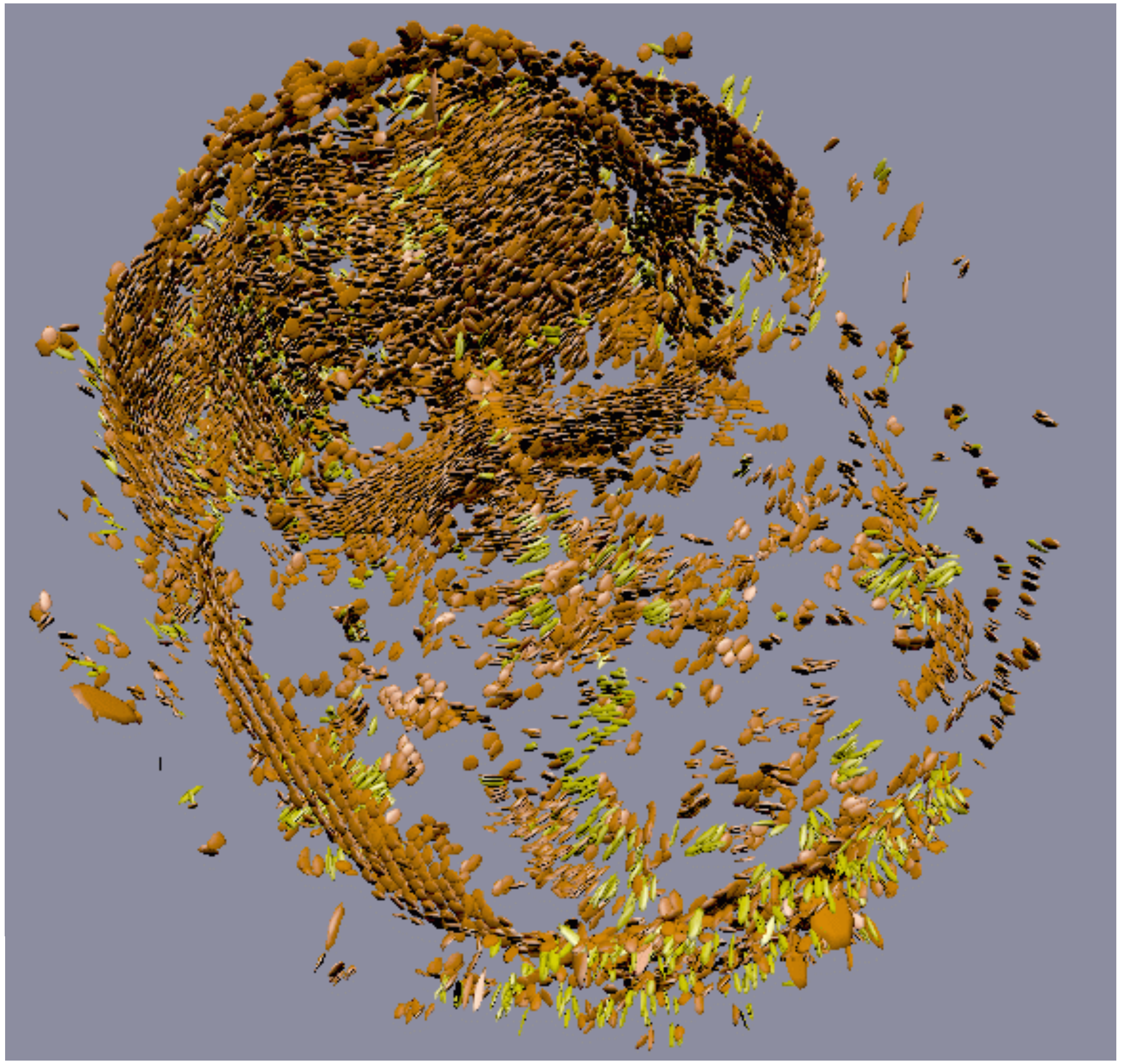


[Laidlaw et al. 1998]

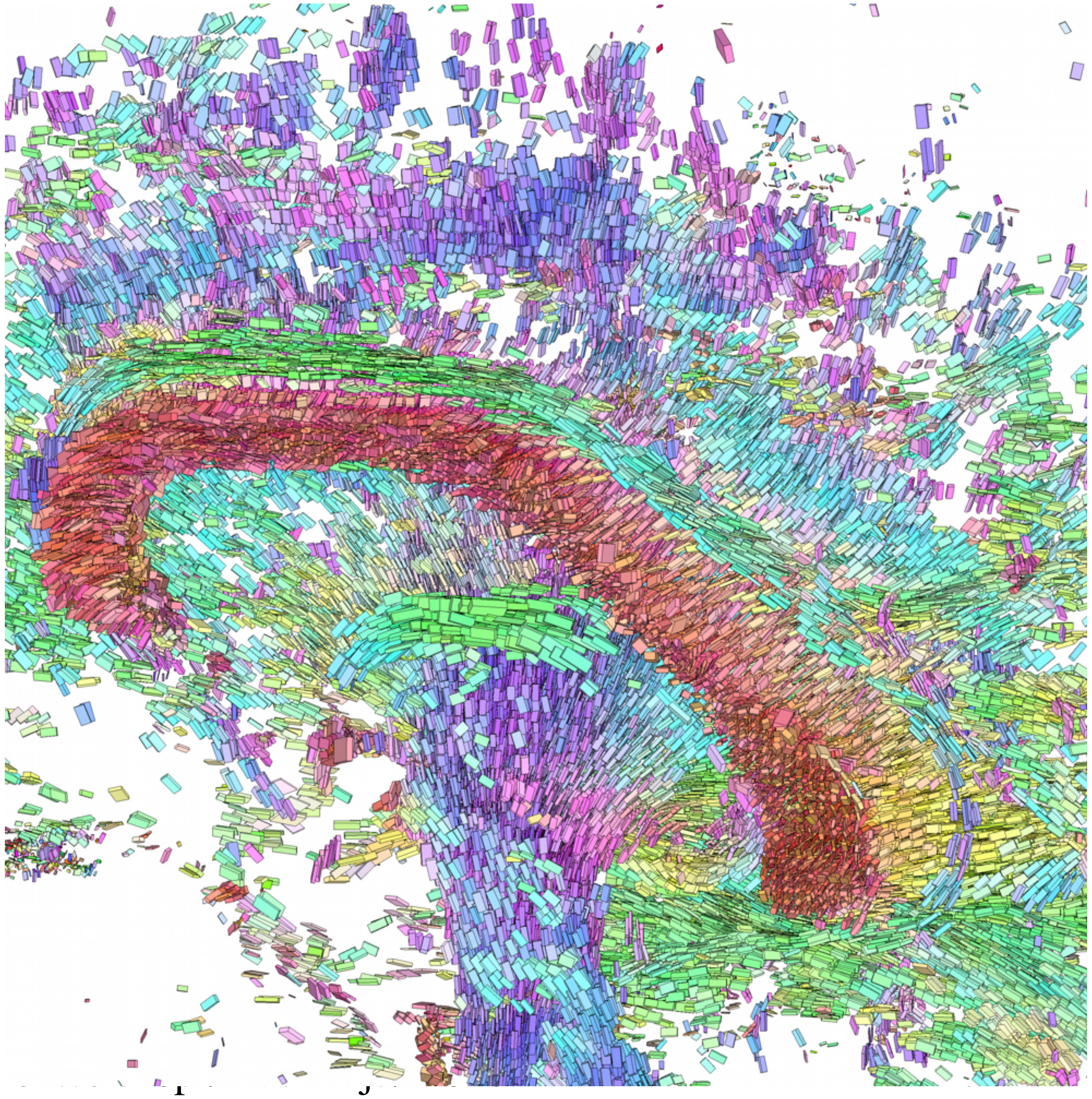
- Brushstrokes
  [Laidlaw et al. 1998]
  - Influenced by paintings
  - Multivalued data
  - Scalar intensity
  - Voxel size
  - Diffusion tensor
  - Textured strokes



- Ellipsoids in 3D
- Problems:
  - Occlusion
  - Missing continuity



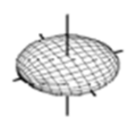
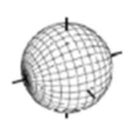
- Box glyphs
  [Johnson et al. 2001]



Glyphs for Tensors

Consider symmetric tensors at this moment. They have real eigenvalues and orthogonal Eigenvectors. Therefore, they can be intuitively represented as ellipsoids.

Three types of anisotropy:

- linear anisotropy 
- planar anisotropy 
- isotropy (spherical) 

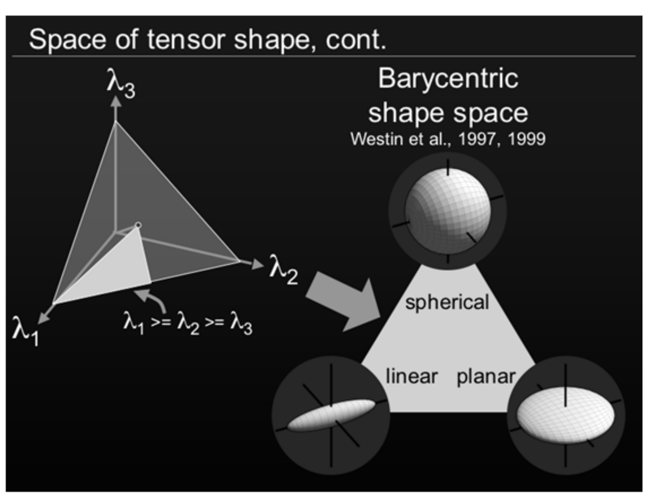
Anisotropy measure

$$c_l = (\lambda_1 - \lambda_2) \big/ (\lambda_1 + \lambda_2 + \lambda_3)$$

$$c_p = 2(\lambda_2 - \lambda_3) \big/ (\lambda_1 + \lambda_2 + \lambda_3)$$

$$c_s = 3\lambda_3 \big/ (\lambda_1 + \lambda_2 + \lambda_3)$$

$$\lambda_1 \geq \lambda_2 \geq \lambda_3$$



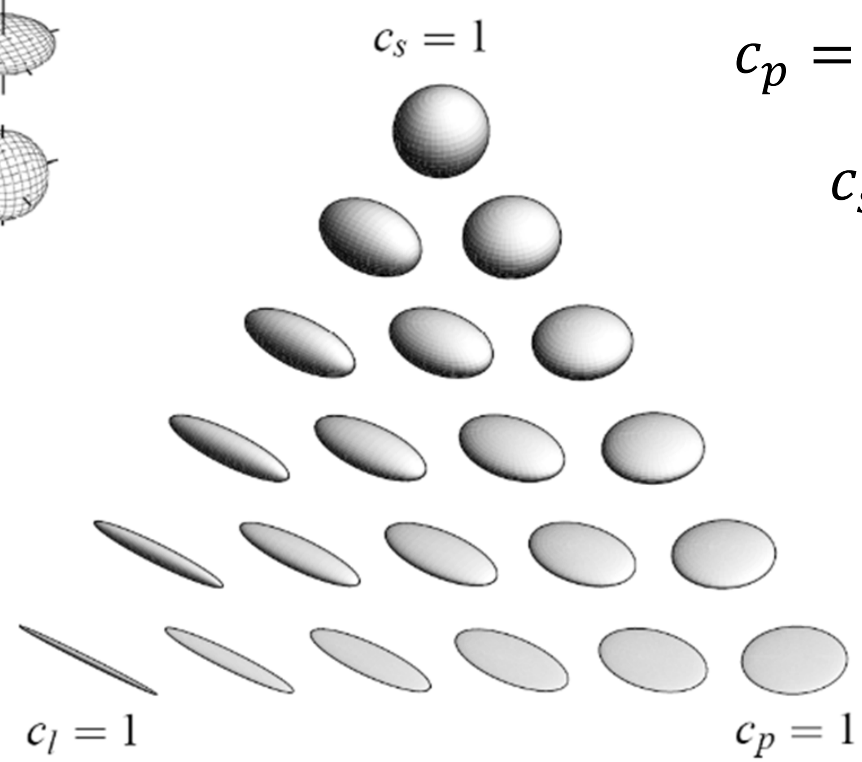
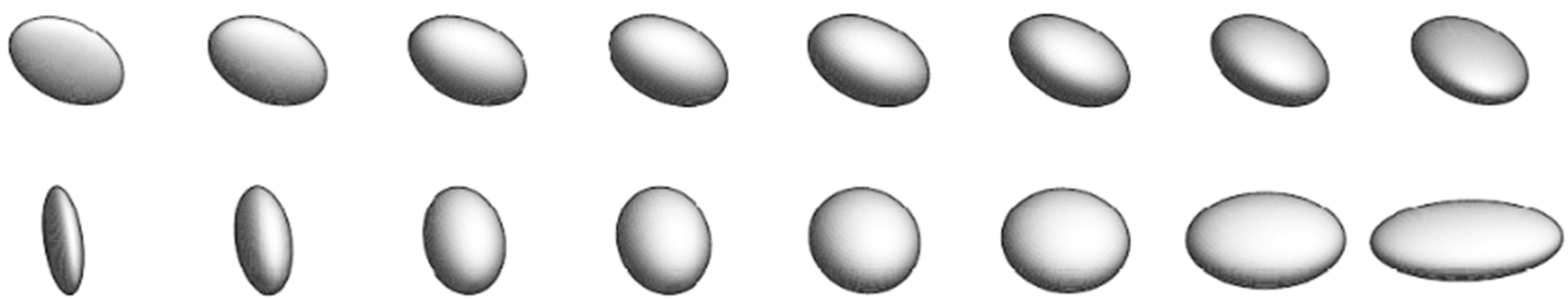


Image by G. Kindlmann

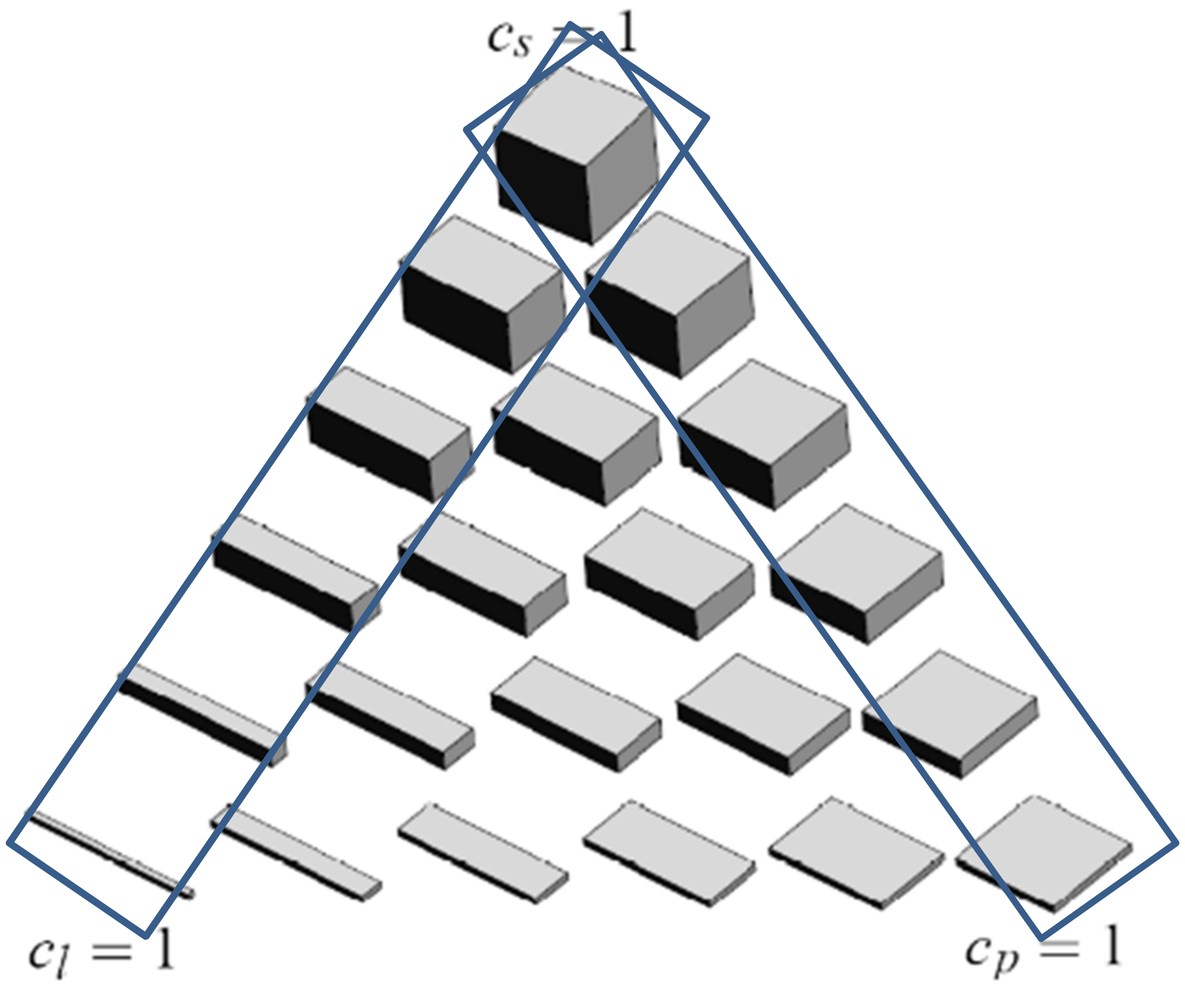
Problem of **ellipsoid** glyphs:
- Shape is poorly recognized in projected view

8 ellipsoids but in two different views (two rows)



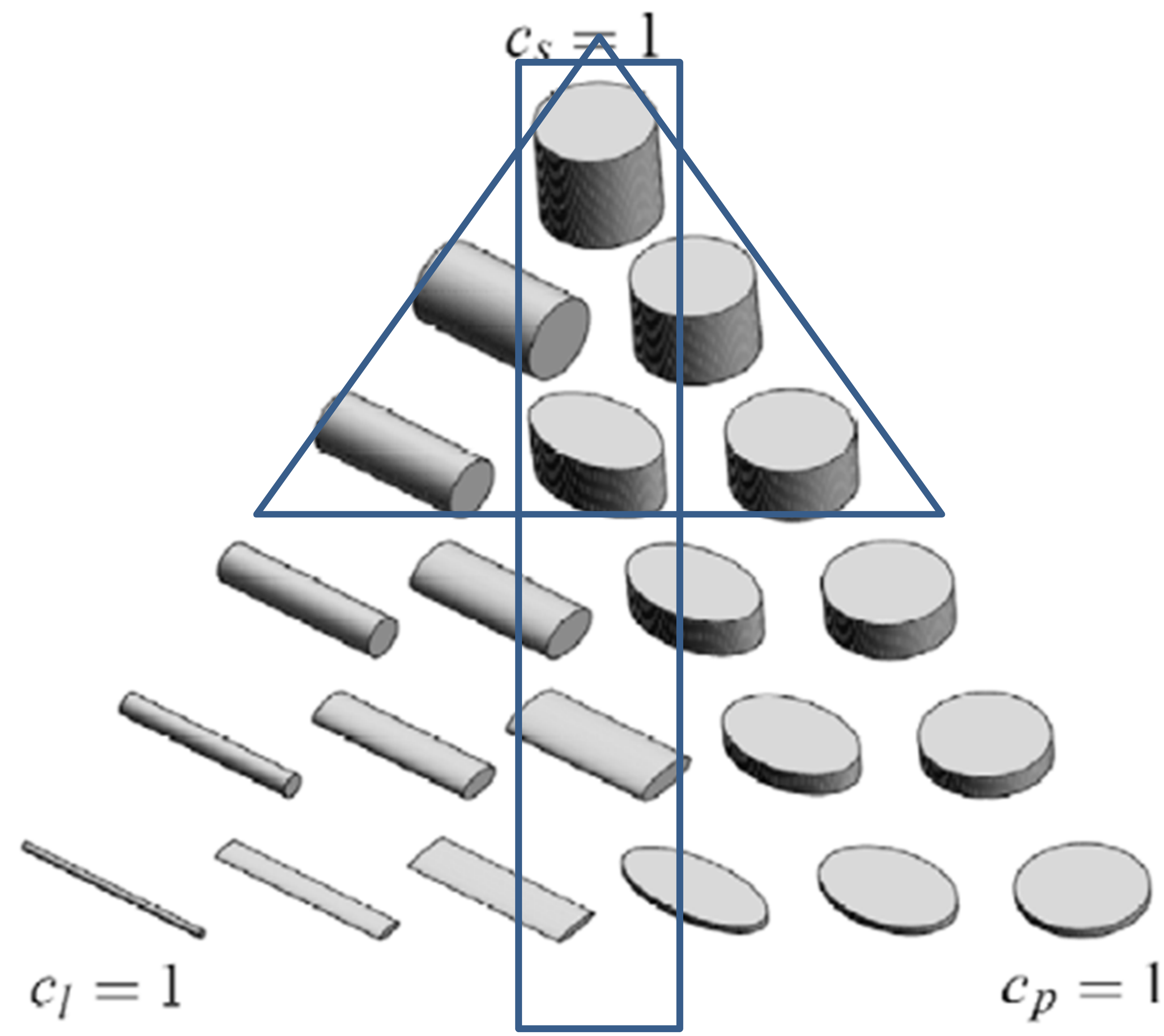
Problem of **cuboid** glyphs

- Missing symmetry



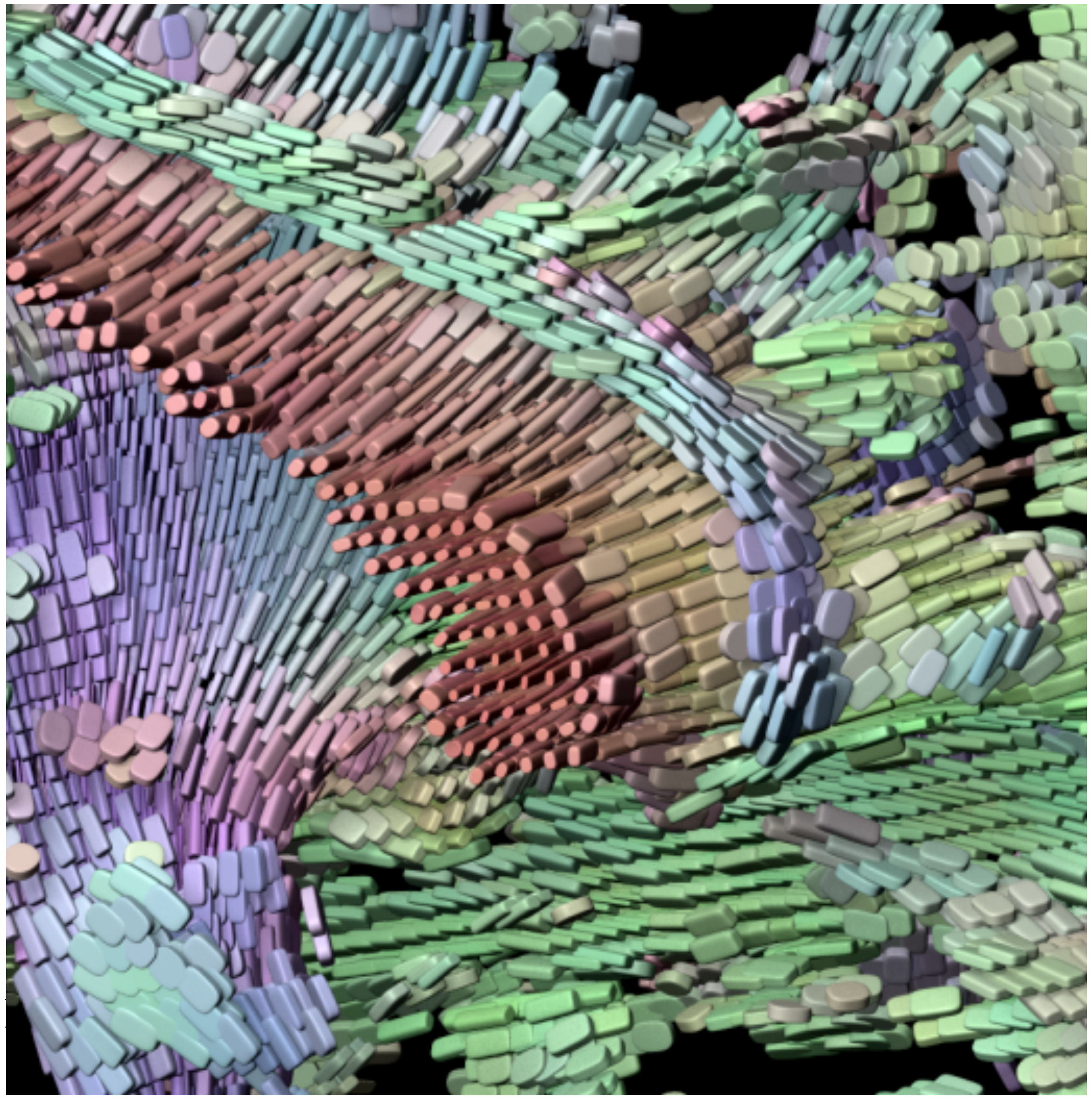
Problem of **cylinder** glyphs

- Seam at $c_l = c_p$
- Losing symmetric close to $c_s = 1$

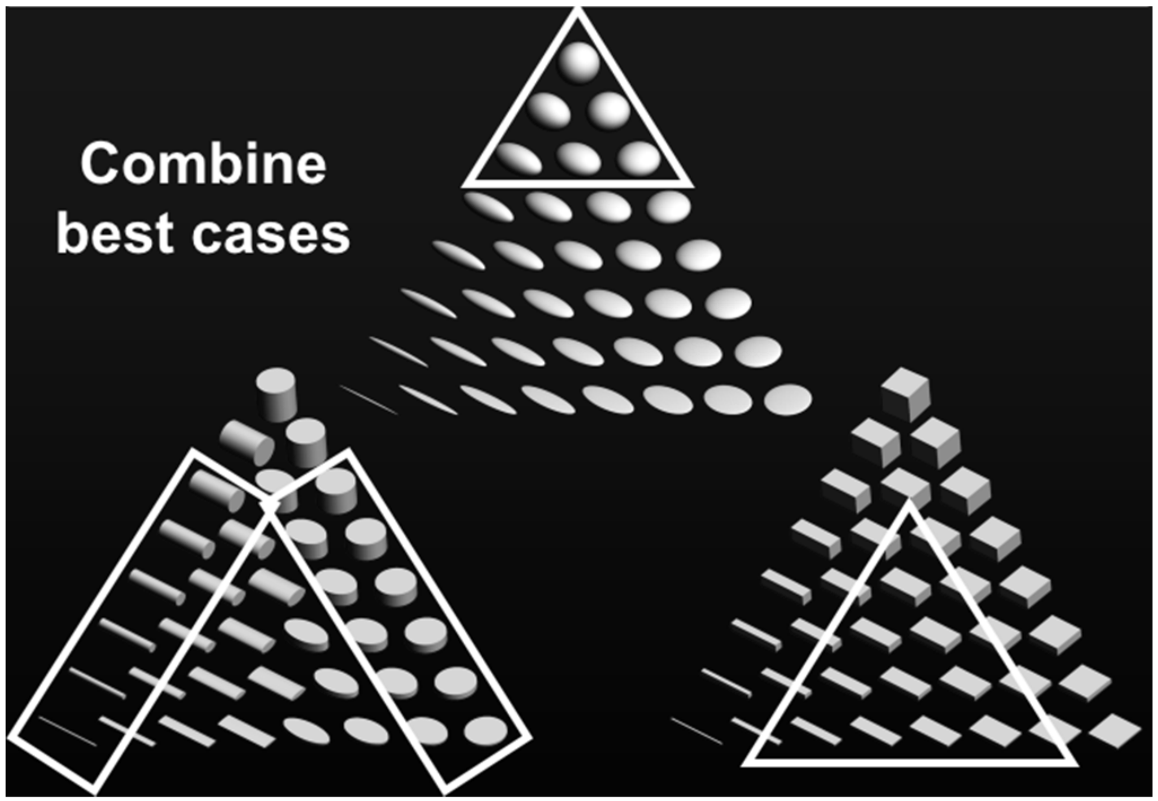


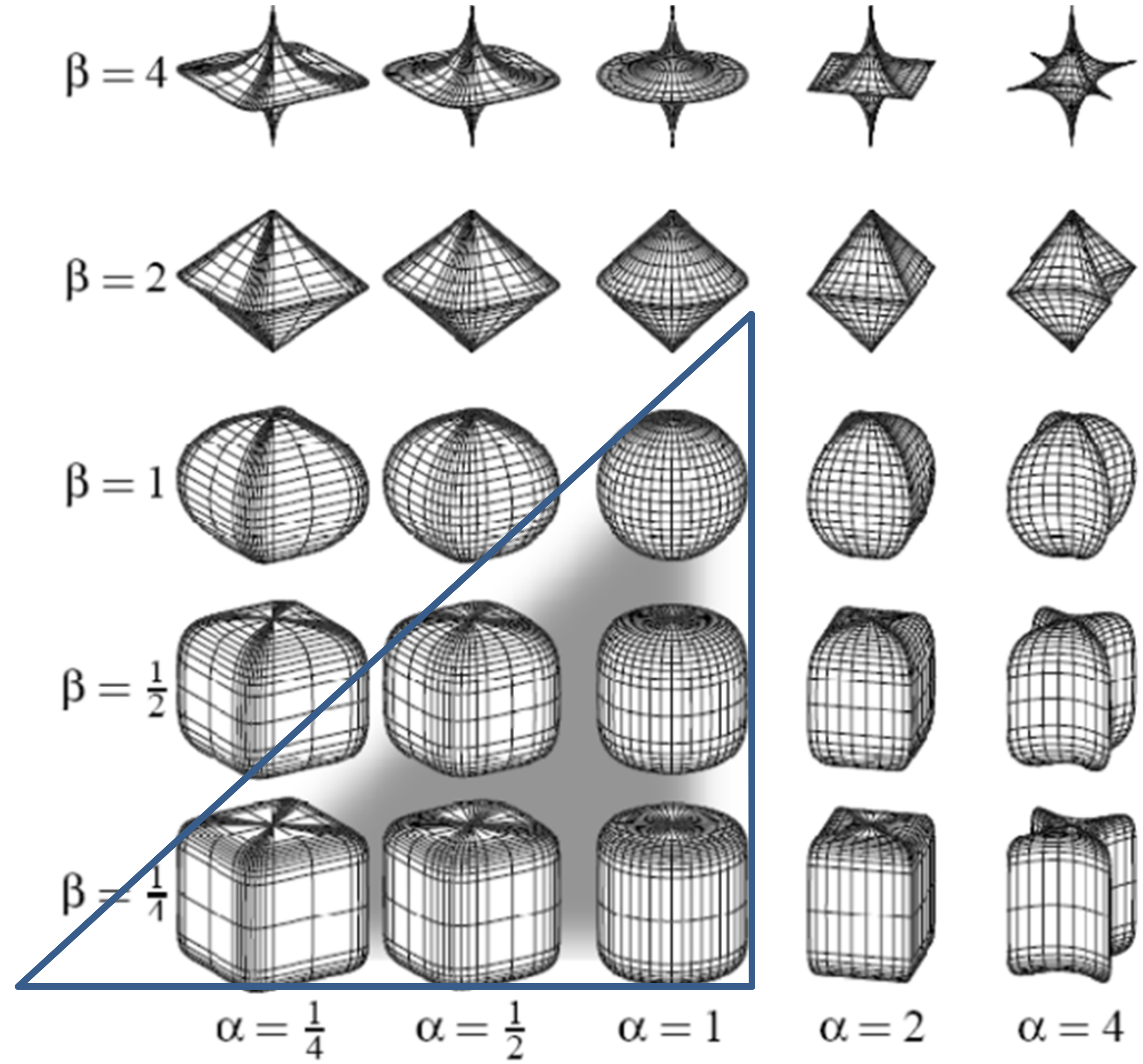
- Superquadric glyphs

  [Kindlmann 2004]



Combining advantages: **superquadrics**
Superquadrics with z as primary axis





$$q_z(\theta, \emptyset) = \begin{bmatrix} \cos^{\alpha}\theta \sin^{\beta}\emptyset \\ \sin^{\alpha}\theta \sin^{\beta}\emptyset \\ \cos^{\beta}\emptyset \end{bmatrix}$$
$$0 \leq \theta \leq 2\pi, 0 \leq \emptyset \leq 2\pi$$

Barr, 1981

Superquadrics for some pairs (α, β)

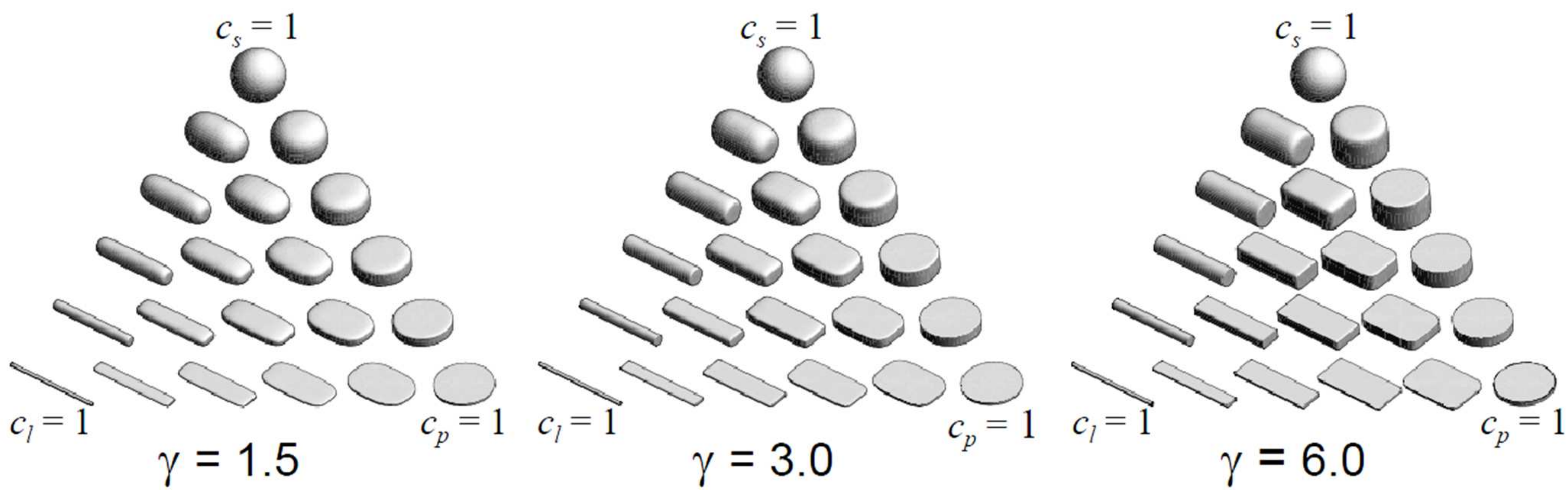
Shaded: sub-range used for glyphs

Superquadric glyphs (Kindlmann): Given c_l, c_p, c_s

- Compute a base superquadric using an **edge sharpness value γ**:

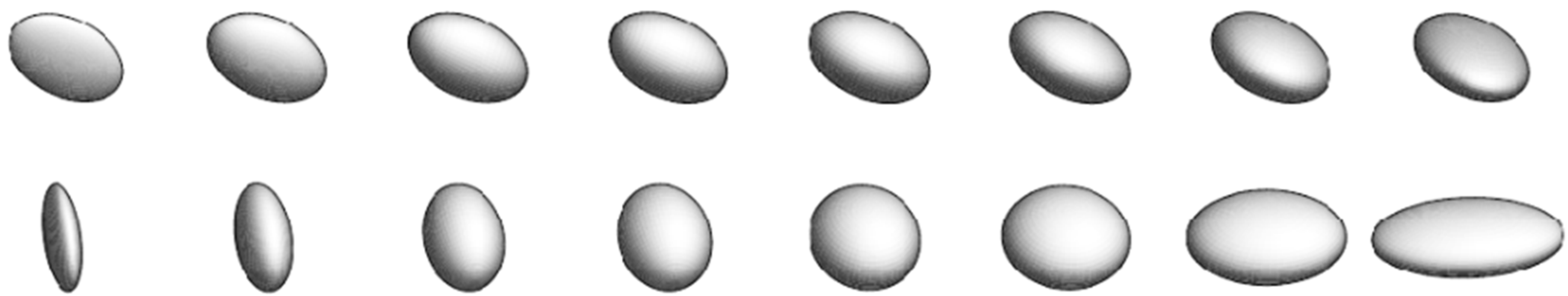
$$q(\theta, \emptyset) = \begin{cases} if\, c_l \geq c_p\colon\ q_z(\theta, \emptyset)\ with\ \alpha = \left(1 - c_p\right)^{\gamma}\ and\ \beta = (1 - c_l)^{\gamma} \\ if\, c_l < c_p\colon\ q_x(\theta, \emptyset)\ with\ \alpha = (1 - c_l)^{\gamma}\ and\ \beta = \left(1 - c_p\right)^{\gamma} \end{cases}$$

- Scale with c_l, c_p, c_s along $\mathrm{x}, \mathrm{y}, \mathrm{z}$ and rotate into eigenvector frame

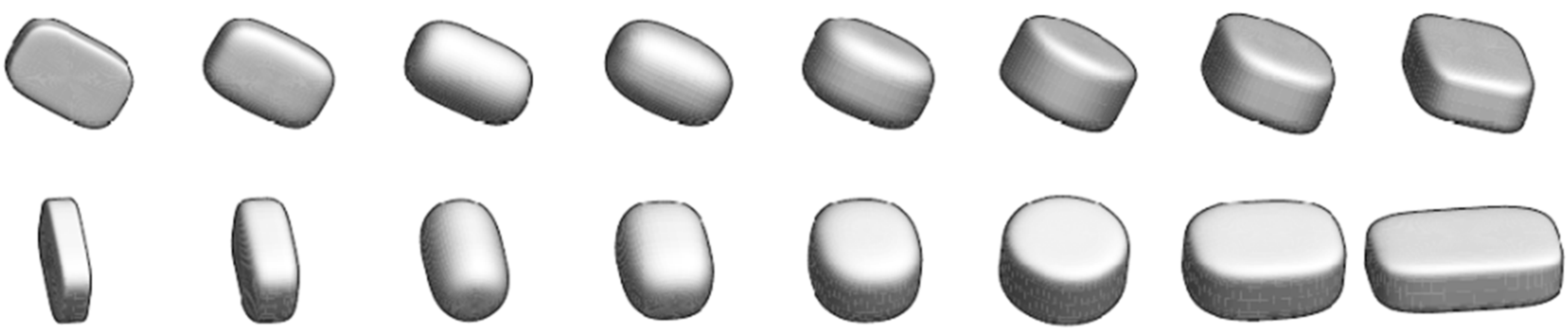


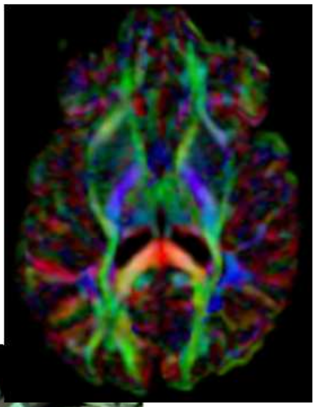
Comparison of shape perception (previous example)

- With ellipsoid glyphs

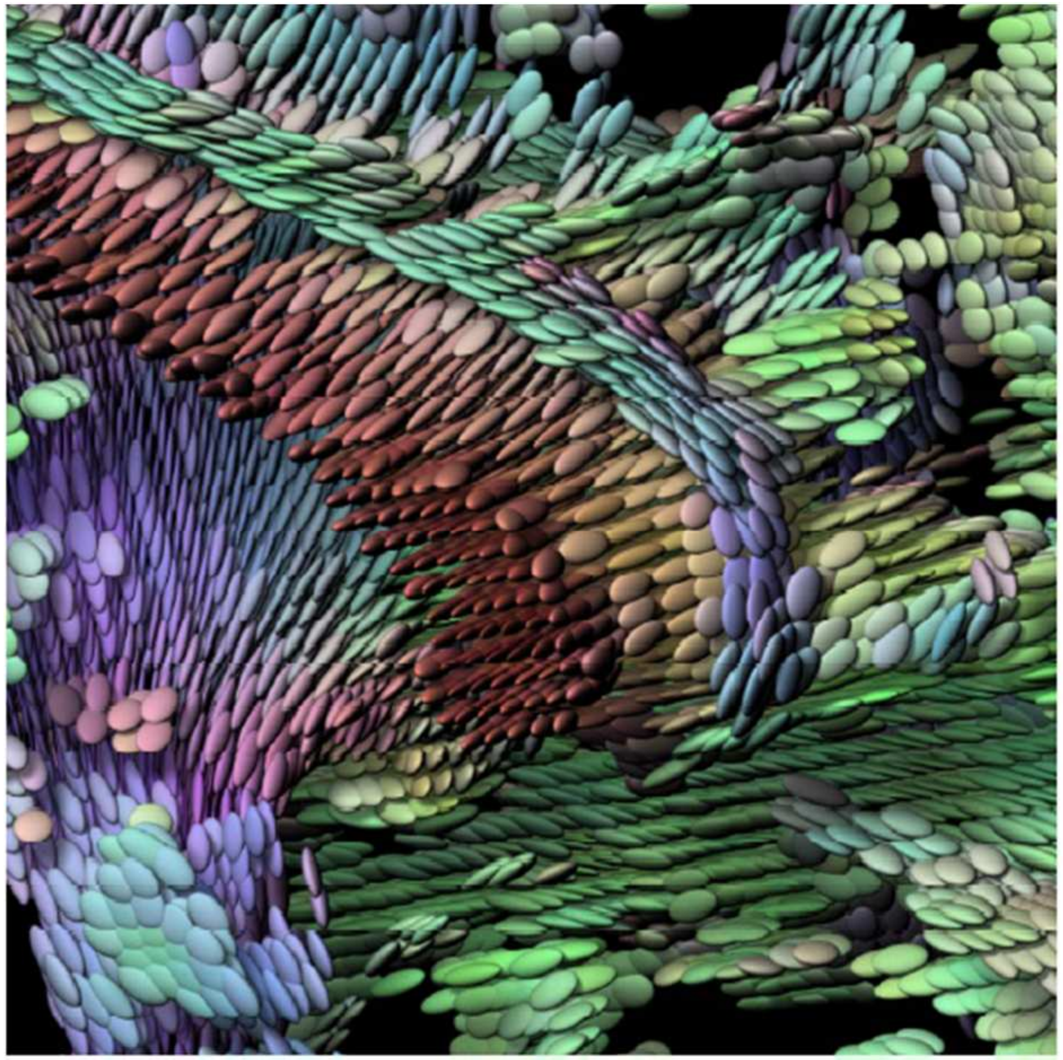


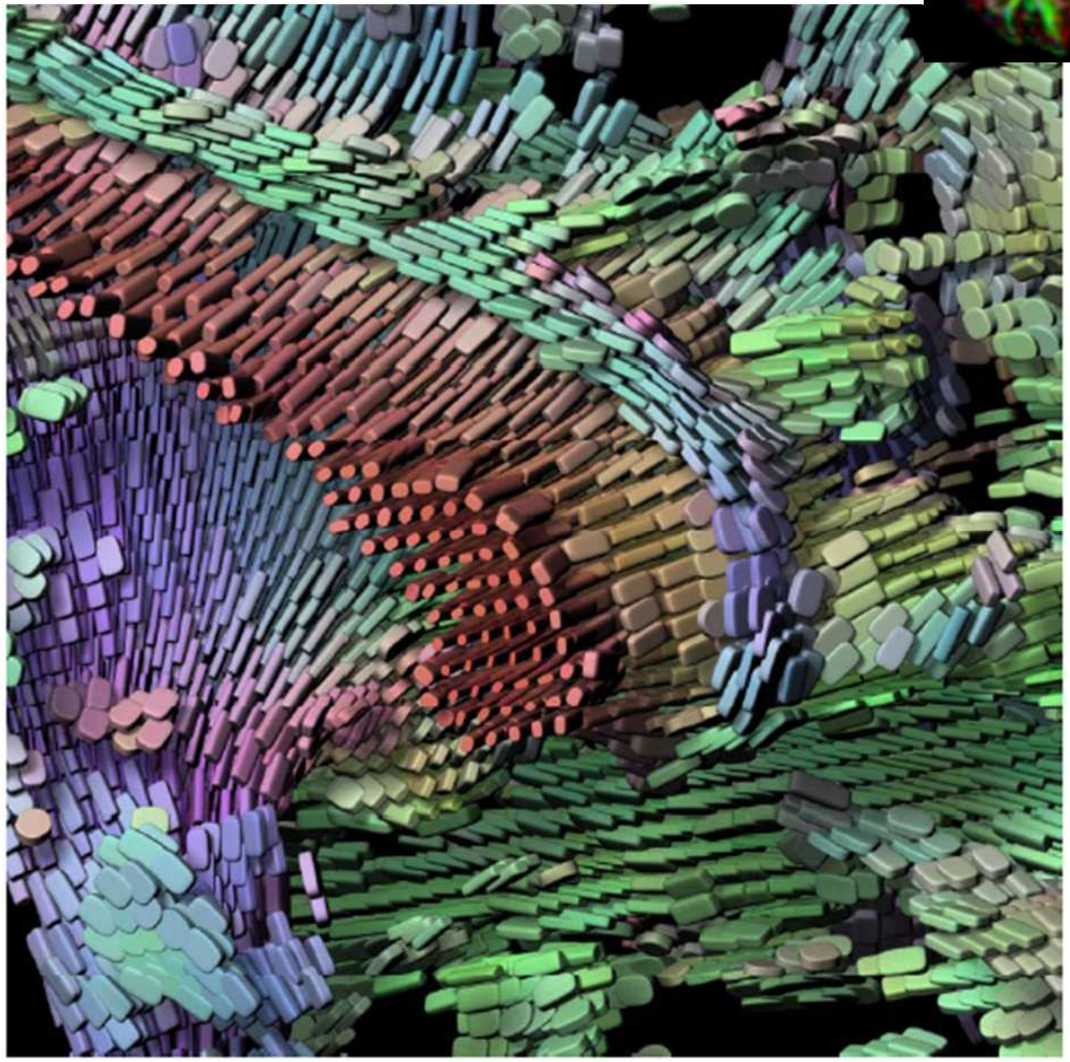
- With superquadrics glyphs





Comparison: Ellipsoids vs. superquadrics (Kindlmann)





Color map $\begin{pmatrix} R \\ G \\ B \end{pmatrix} = c_l \begin{bmatrix} |{e_x}^1| \\ |{e_y}^1| \\ |{e_z}^1| \end{bmatrix} + (1 - c_l) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

This is half of the brain, looking at the posterior part of the corpus callosum, which is the main bridge between the two hemispheres. And with the superquadrics, you can see that on the surface of the corpus callosum, the glyphs have more of a planar component, but on the inside, they're basically very linear.

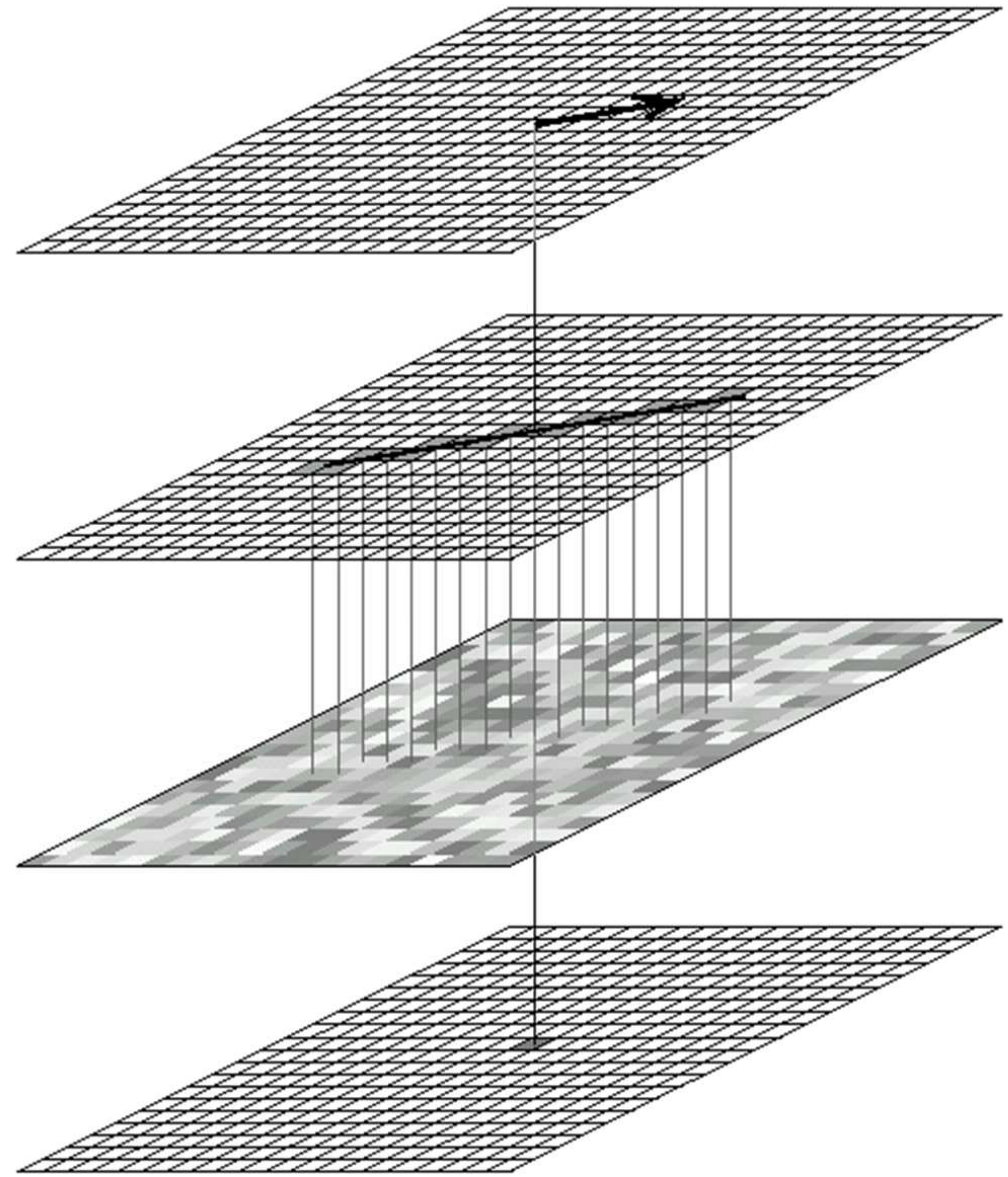
Texture-Based

HyperLIC [Zheng and Pang]

Instead using a 1D kernel along the streamline, HyperLIC uses a 2D kernel

$$
\begin{aligned}
I_o(P) &= \frac{\sum_{i=-N}^{N}\sum_{j=-N}^{N} k(i,j) I_n(P_{i,j})}{\sum_{i=-N}^{N}\sum_{j=-N}^{N} k(i,j)} \\
P_{i,j} &= P_{i-1,j} + \lambda_1(P_{i-1,j}) e_1(P_{i-1,j}) \Delta t \\
P_{0,j} &= P_{0,j-1} + \lambda_2(P_{0,j-1}) e_2(P_{0,j-1}) \Delta t \\
P_{0,0} &= P
\end{aligned}
$$

I_n is the input and I_o is the output
$\lambda_n(X), e_n(X), k(i,j), n = 1{,}2$ are the nth eigenvalues, eigenvectors and the weight function at point X. $\Delta t$is the integration step.



The LIC pipeline

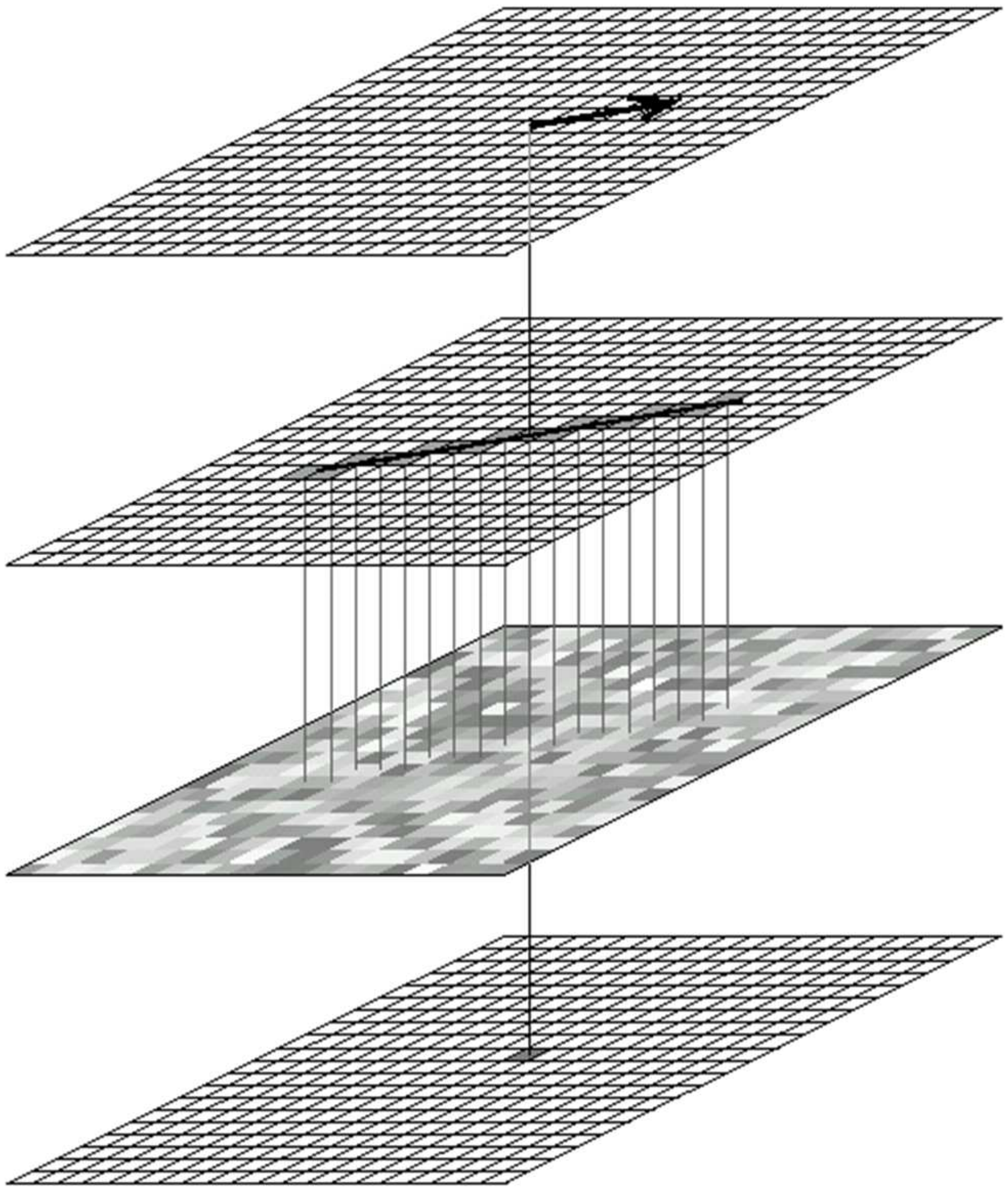
HyperLIC [Zheng and Pang]

Instead using a 1D kernel along the streamline, HyperLIC uses a 2D kernel

Decompose the computation

If we define

$$
\begin{aligned}
I_1(P) &= \frac{\sum_{j=-N}^{N} k_2(j) I_n(P_j)}{\sum_{j=-N}^{N} k_2(j)} \\
P_j &= P_{j-1} + \lambda_2(P_{j-1}) e_2(P_{j-1}) \Delta t \\
P_0 &= P
\end{aligned}
$$



The LIC pipeline

HyperLIC [Zheng and Pang]

Instead using a 1D kernel along the streamline, HyperLIC uses a 2D kernel

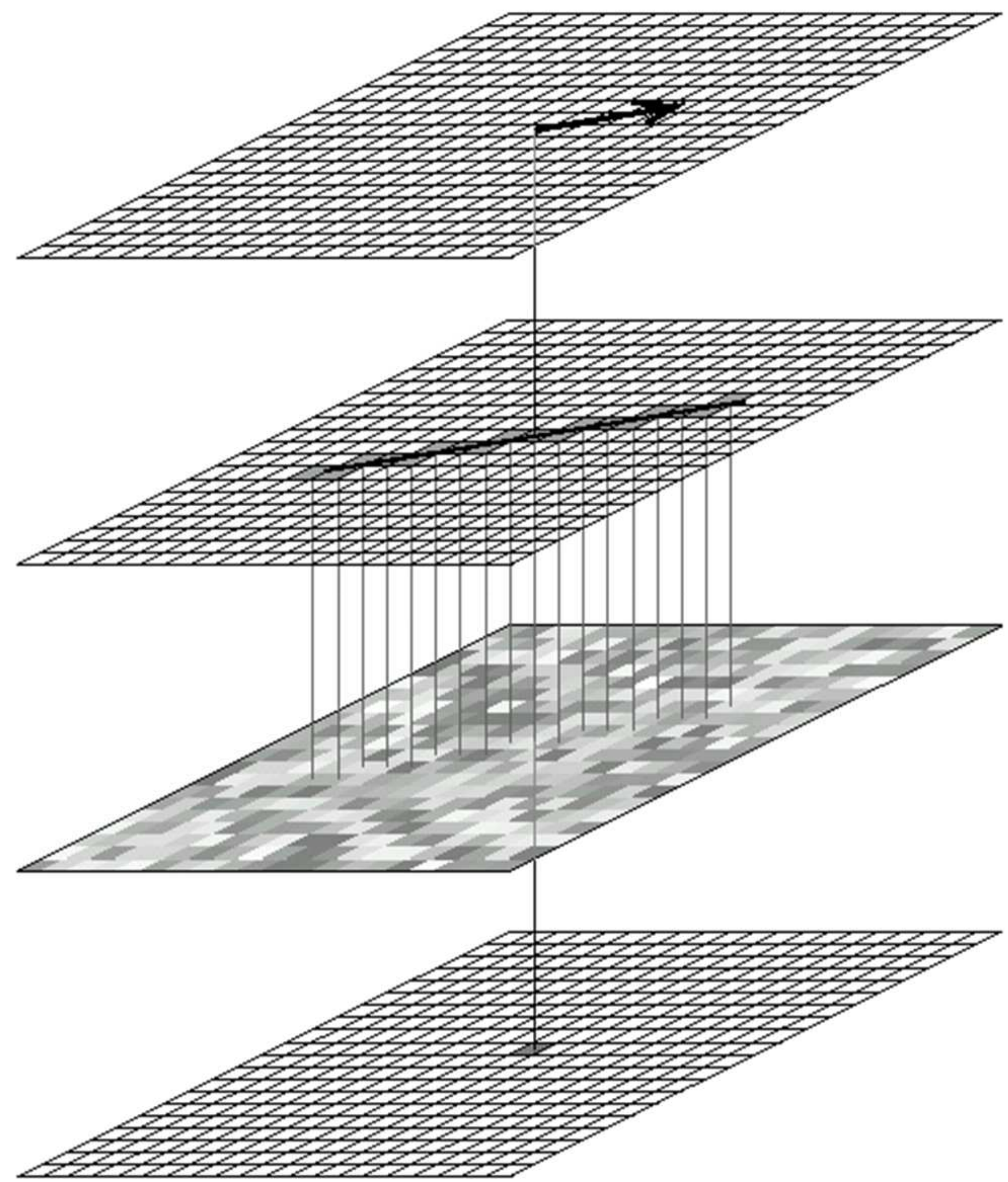
Decompose the computation

Then

$$
\begin{aligned}
I_o(P) &= \frac{\sum_{i=-N}^{N} k_1(i) I_1(P_i)}{\sum_{i=-N}^{N} k_1(i)} \\
P_i &= P_{i-1} + \lambda_1(P_{i-1}) e_1(P_{i-1}) \Delta t \\
P_0 &= P
\end{aligned}
$$

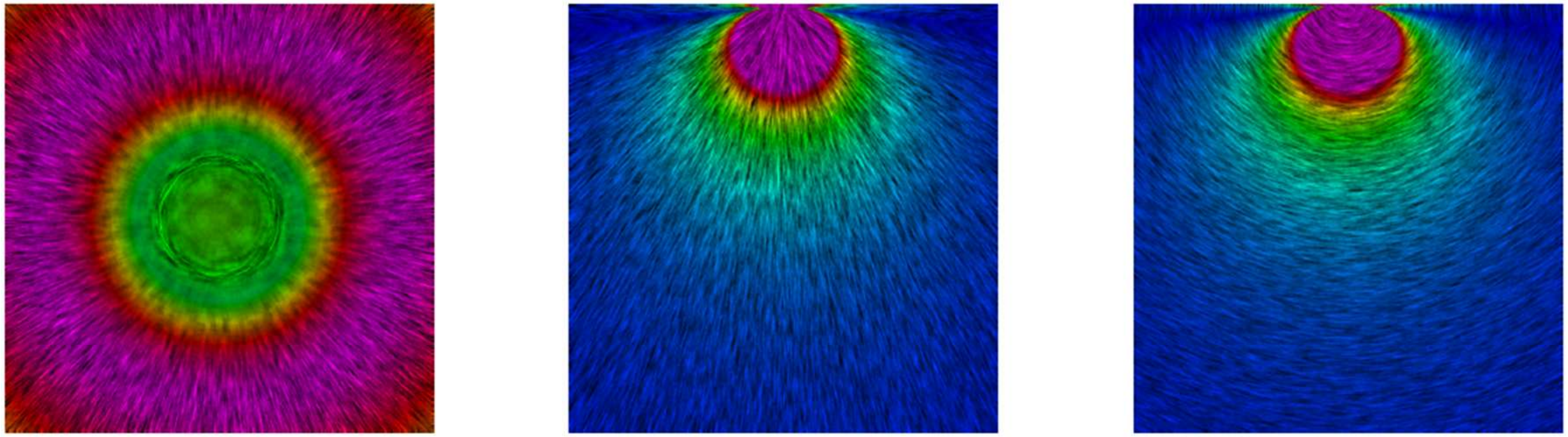
This is a two-pass process

I_1 and I_0 are the output images of the un-normalized LIC on $\lambda_2 e_2$ and $\lambda_1 e_1$ vector fields with input images I_n and I_1, respectively.

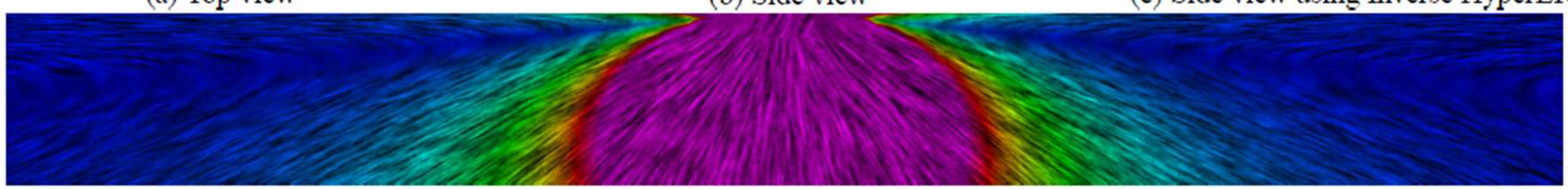


The LIC pipeline

Some Results



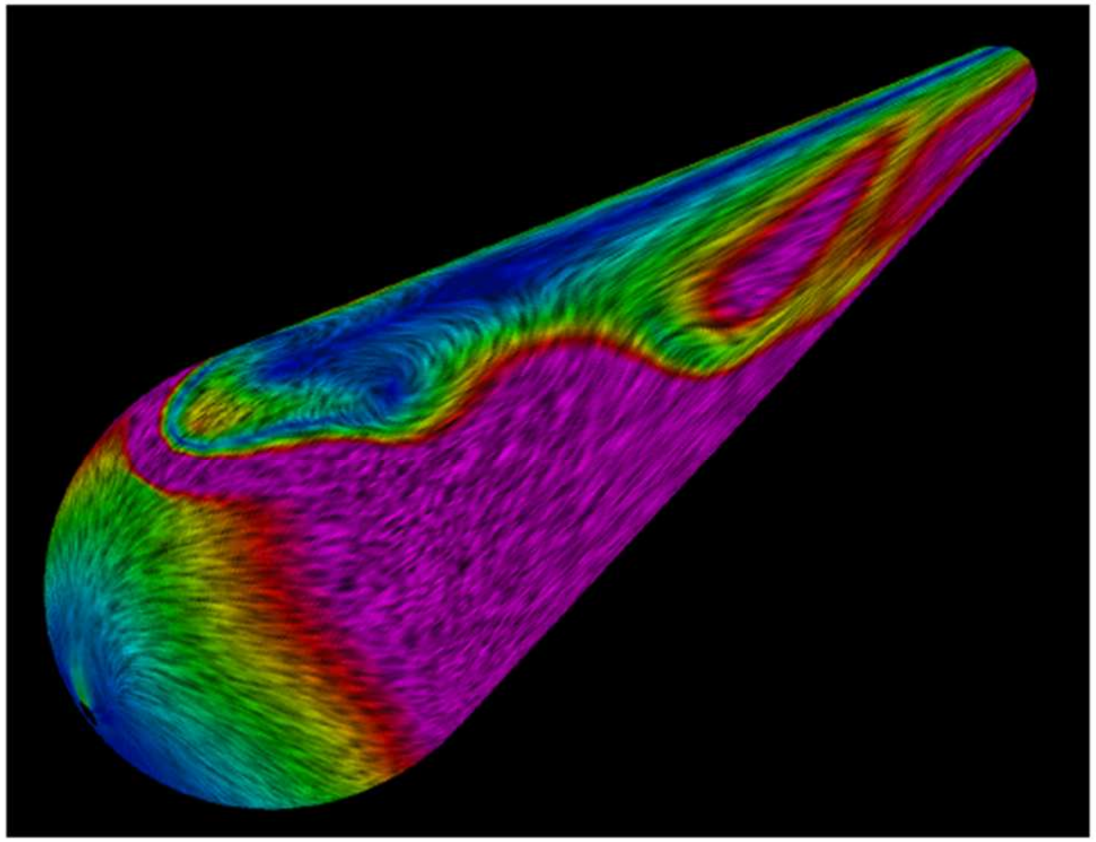
(a) Top view (b) Side view (c) Side view using Inverse HyperLIC



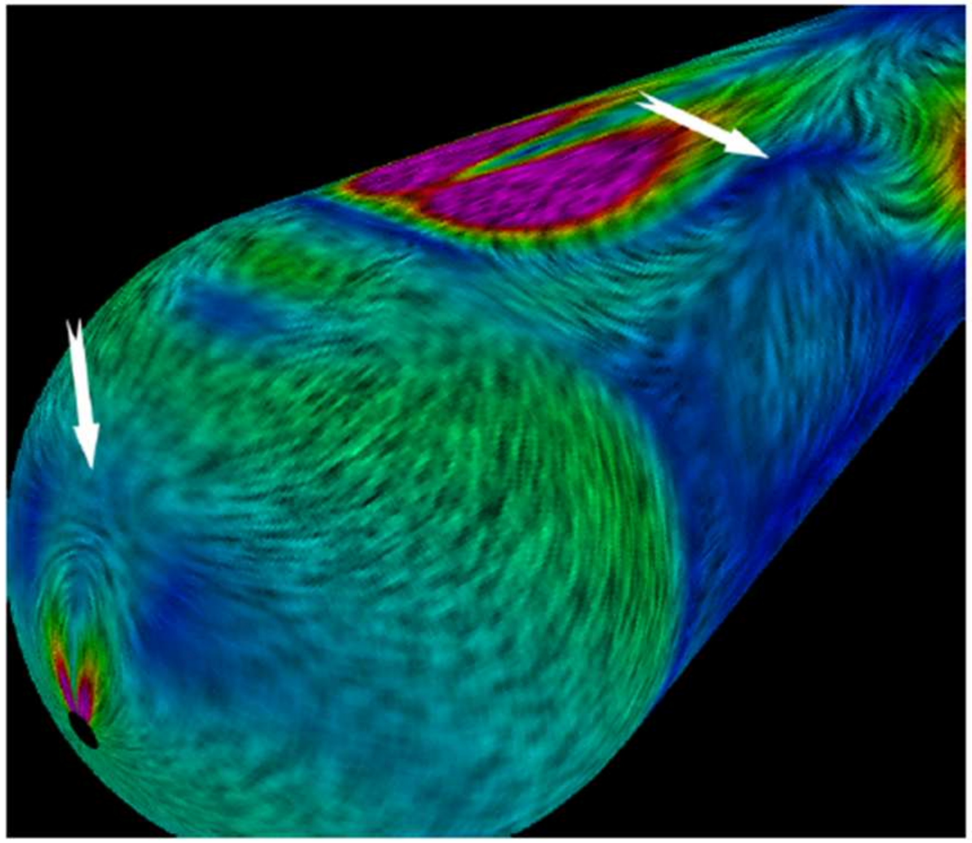
(d) Zoomed view of the top portion of (b)

A 2D slice from single point load stress tensors. It is taken from the middle of the volume and viewed from the point load direction. It is mostly composed of components from medium or minor eigenvectors. We see that the center of this slice is quite isotropic. Around the center is a ring formed by lines, which means tensors are highly anisotropic. It is the boundary where the minor eigenvalues are zero.

Some Results



(a) Inner layer



(b) Middle layer

Flow past a cylinder with hemispherical cap. HyperLIC of two different computational layers of the strain rate tensor. Arrows point to locations of degenerate wedge points

Feature-Based

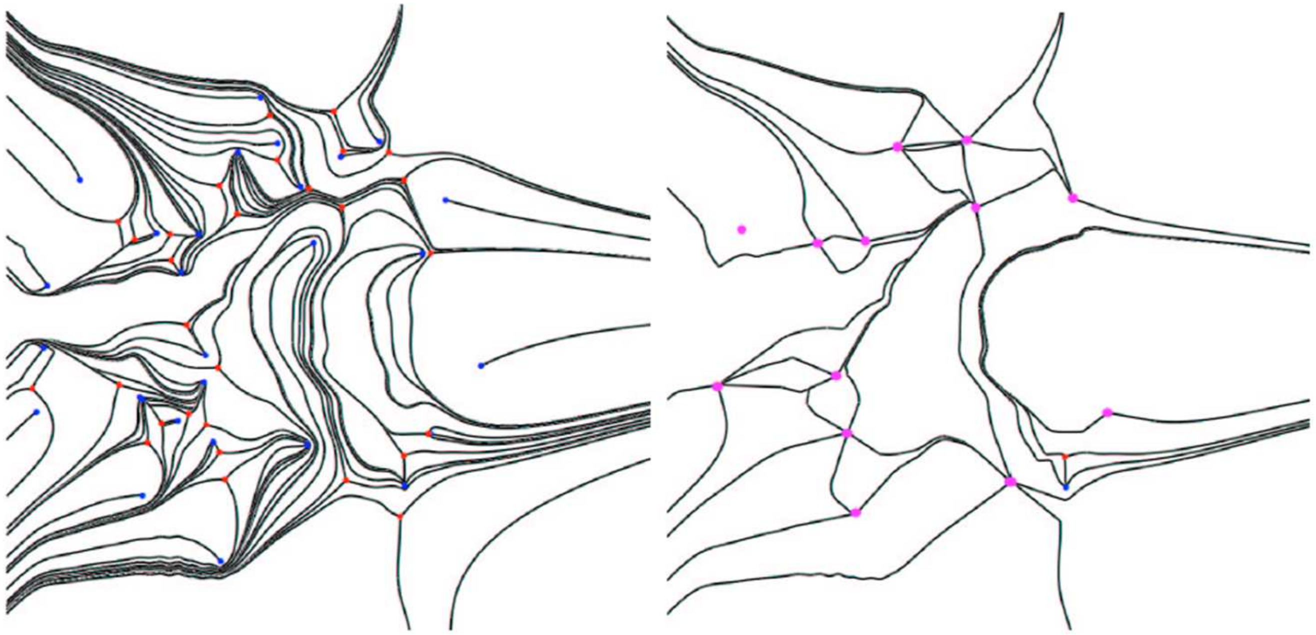
Tensor Field Segmentation

- The goal of tensor segmentation algorithms is to aggregate regions that exhibit similar data characteristics to ease the analysis and interpretation of the data.

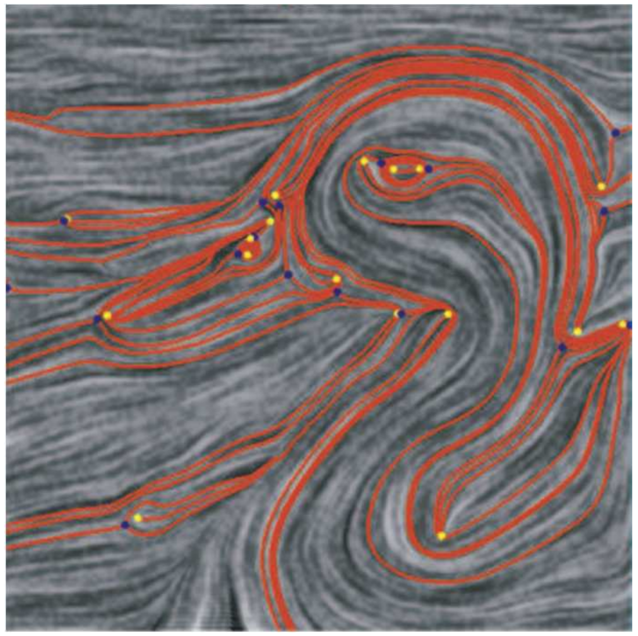
- Two classes
  - Segmentation or clustering based on certain similarity (or dis-similarity) metric
  - Topology-based

Tensor Field Segmentation

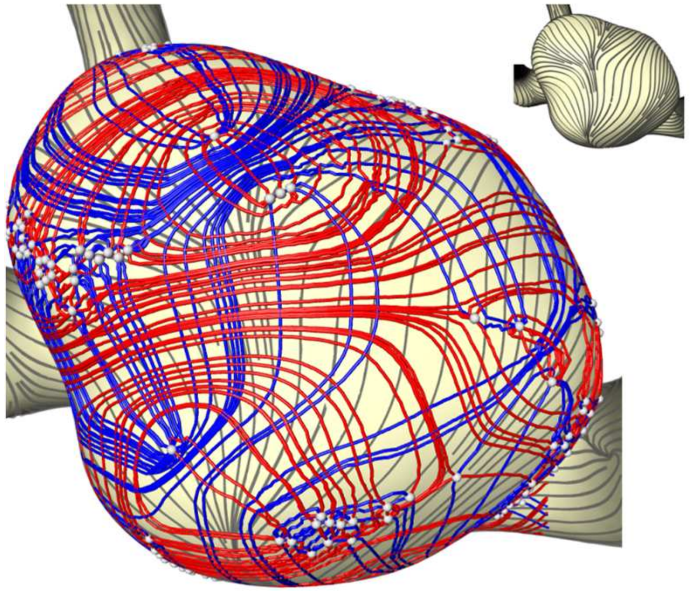
- Topology-Based Segmentation



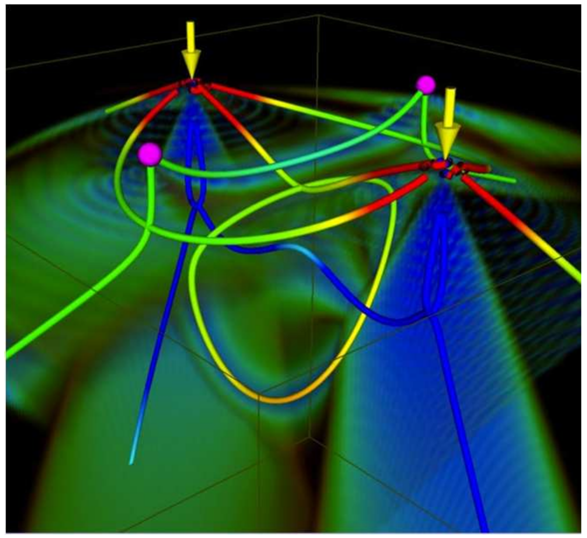
[Tricoche et al. VisSym2001]



[Zhang et al. TVCG 2007]

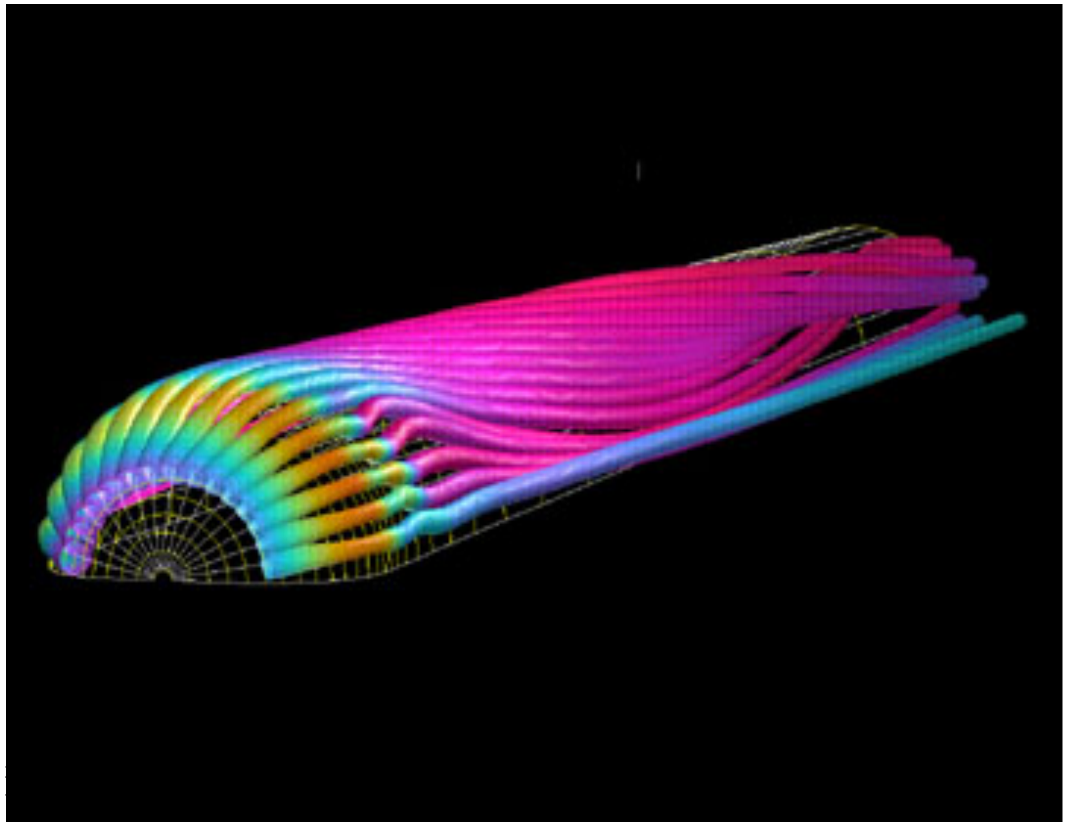


[Auer and Hotz EuroVis11]



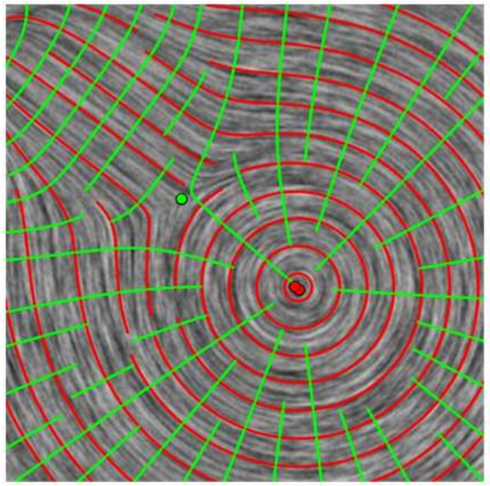
[Zheng and Pang, Vis04]

- Hyperstreamlines [Delmarcelle, Hesselink 1992/93]
  - Streamlines defined by eigenvectors
  - Direction of streamline by major eigenvector
  - Visualization of the vector field defined by major eigenvector
  - Other eigenvectors define cross-section



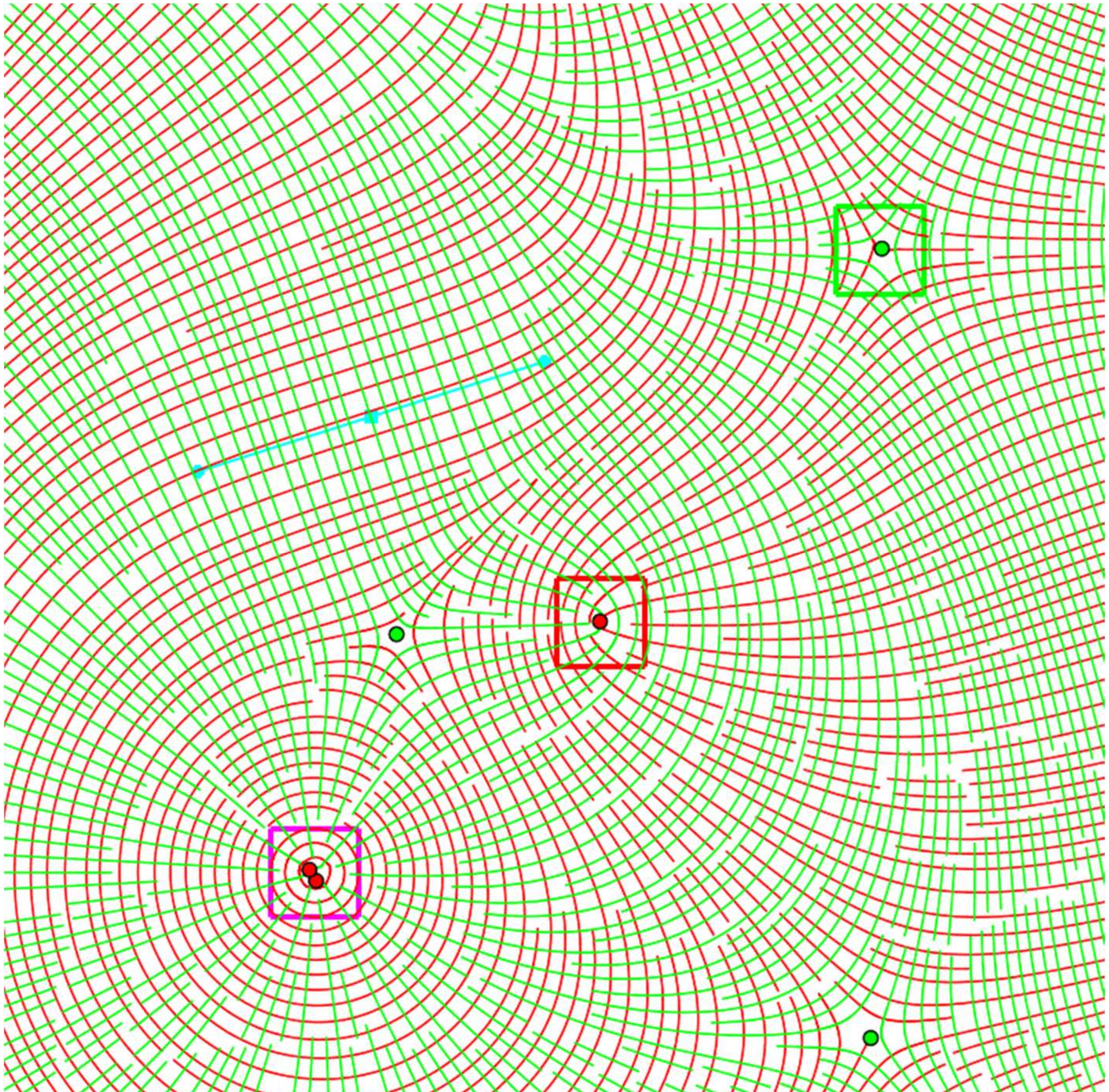
Hyperstreamlines

- Let **T**(**x**) be a (2nd order) symmetric tensor field
  - real eigenvalues, orthogonal eigenvectors



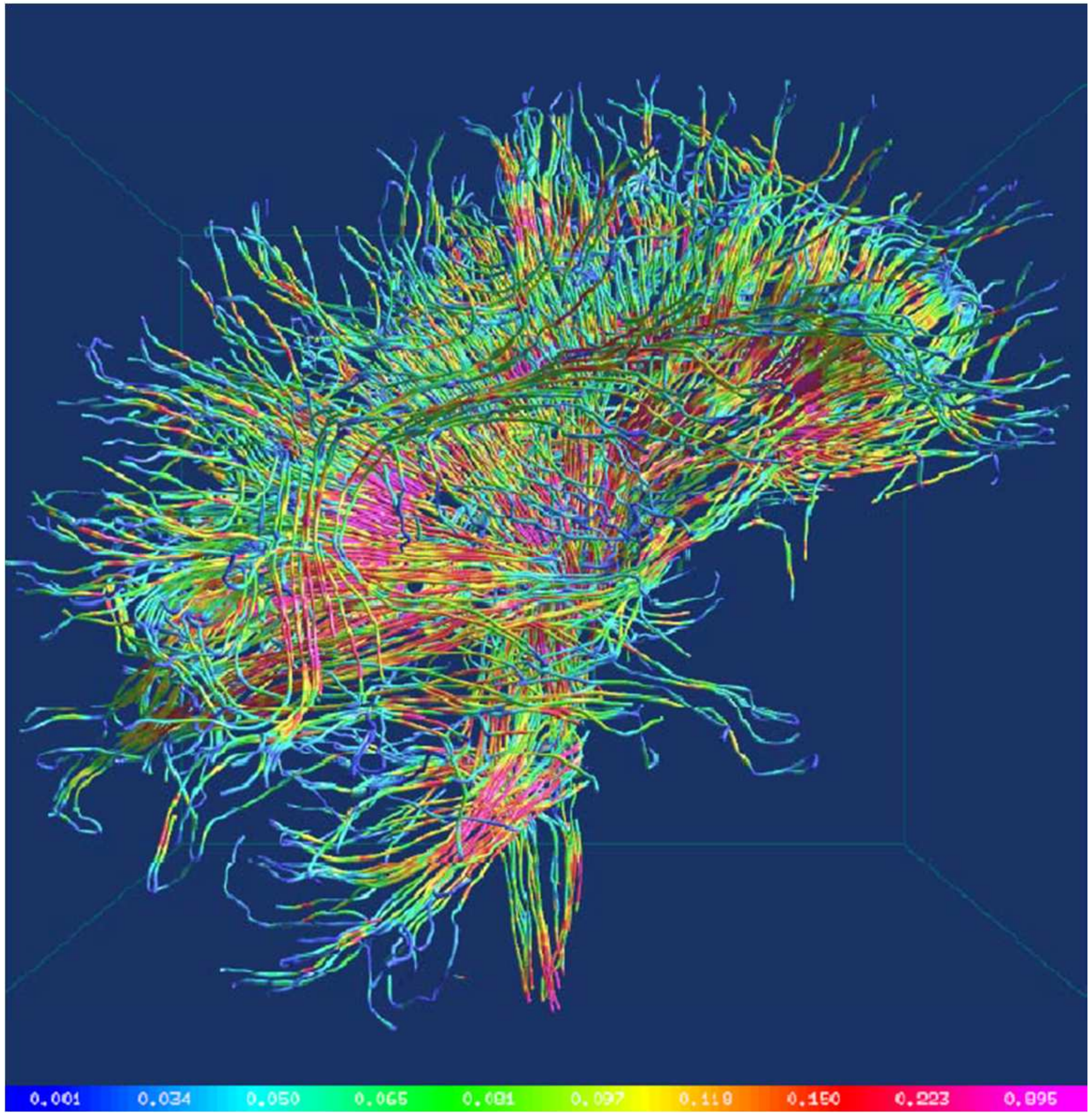
- Hyperstreamline: by integrating along one of the eigenvectors

- **Important: Eigenvector fields are not vector fields!**
  - eigenvectors have no magnitude and no orientation (are bidirectional)
  - the choice of the eigenvector can be made consistently as long as eigenvalues are all different
  - Hyperstreamlines can intersect only at points where two or more eigenvalues are equal, so-called **degenerate points**.



Red – major
Green – minor

Hyperstreamlines

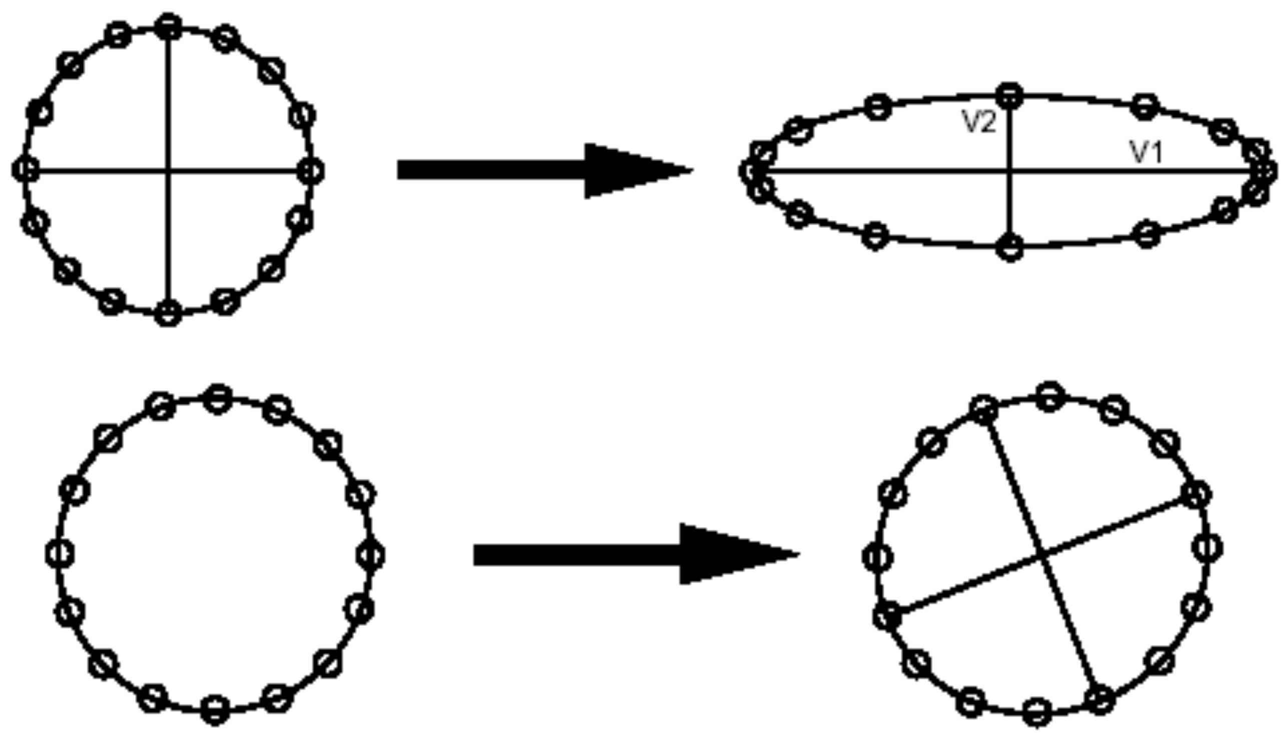


Hyperstreamlines rendered as tubes with elliptic cross section, radii proportional to 2nd and 3rd eigenvalue

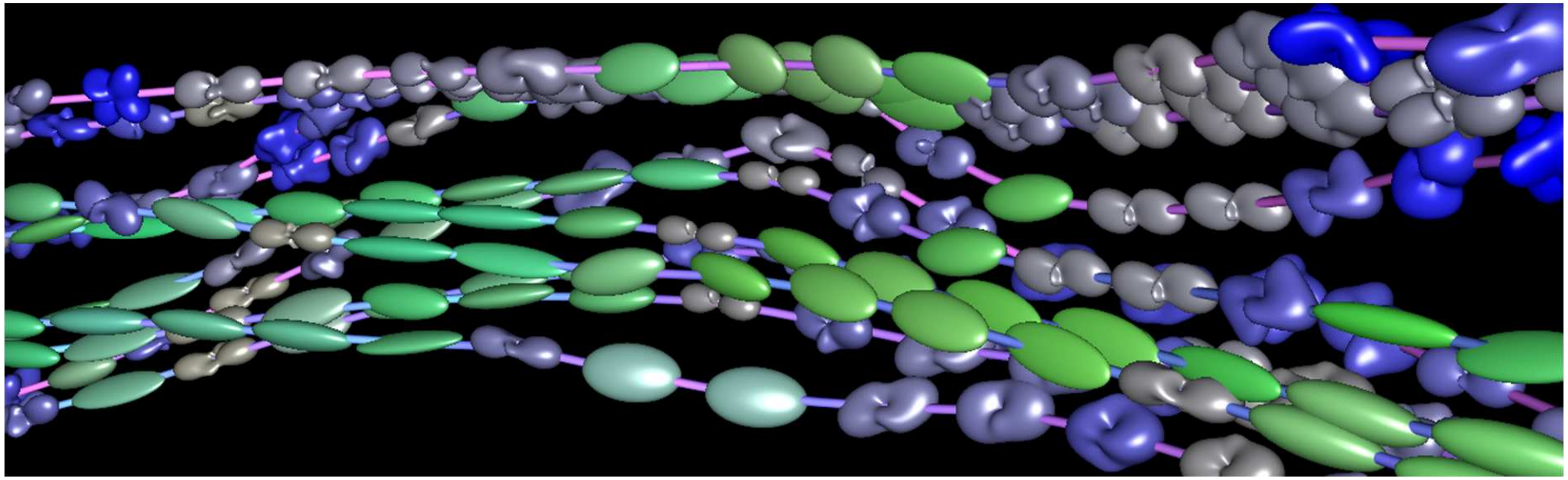
[Shen and Pang 2004]

Widely used in diffusion tensor imaging tractography

- Idea of hyperstreamlines:
  - Major eigenvector describes direction of diffusion with highest probability density



Hyperstreamlines



[Prckovska et al. 2010]

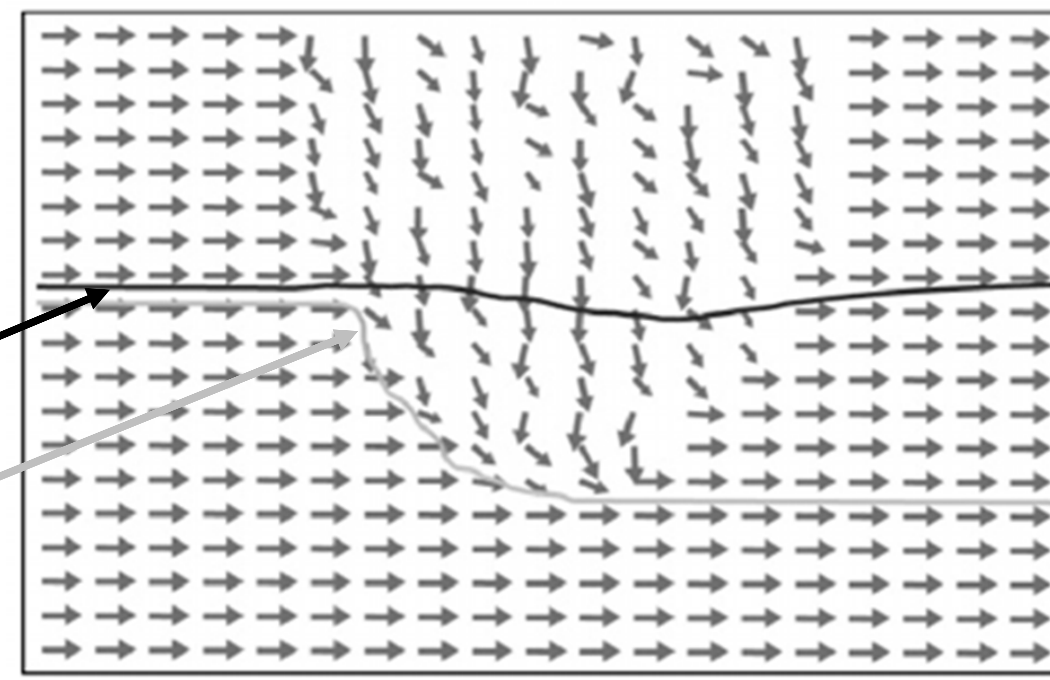
Hybrid visualization: hyperstreamlines + glyphs

Good for some non-symmetric tensor visualization where the rotational components can be encoded by the glyphs

Problem of Hyperstreamlines

- Ambiguity in (nearly) isotropic regions:
  - Partial voluming effect, especially in low resolution images (MR images)
  - Noise in data
  - Solution: tensorlines

    [Weinstein, Kindlmann 1999]

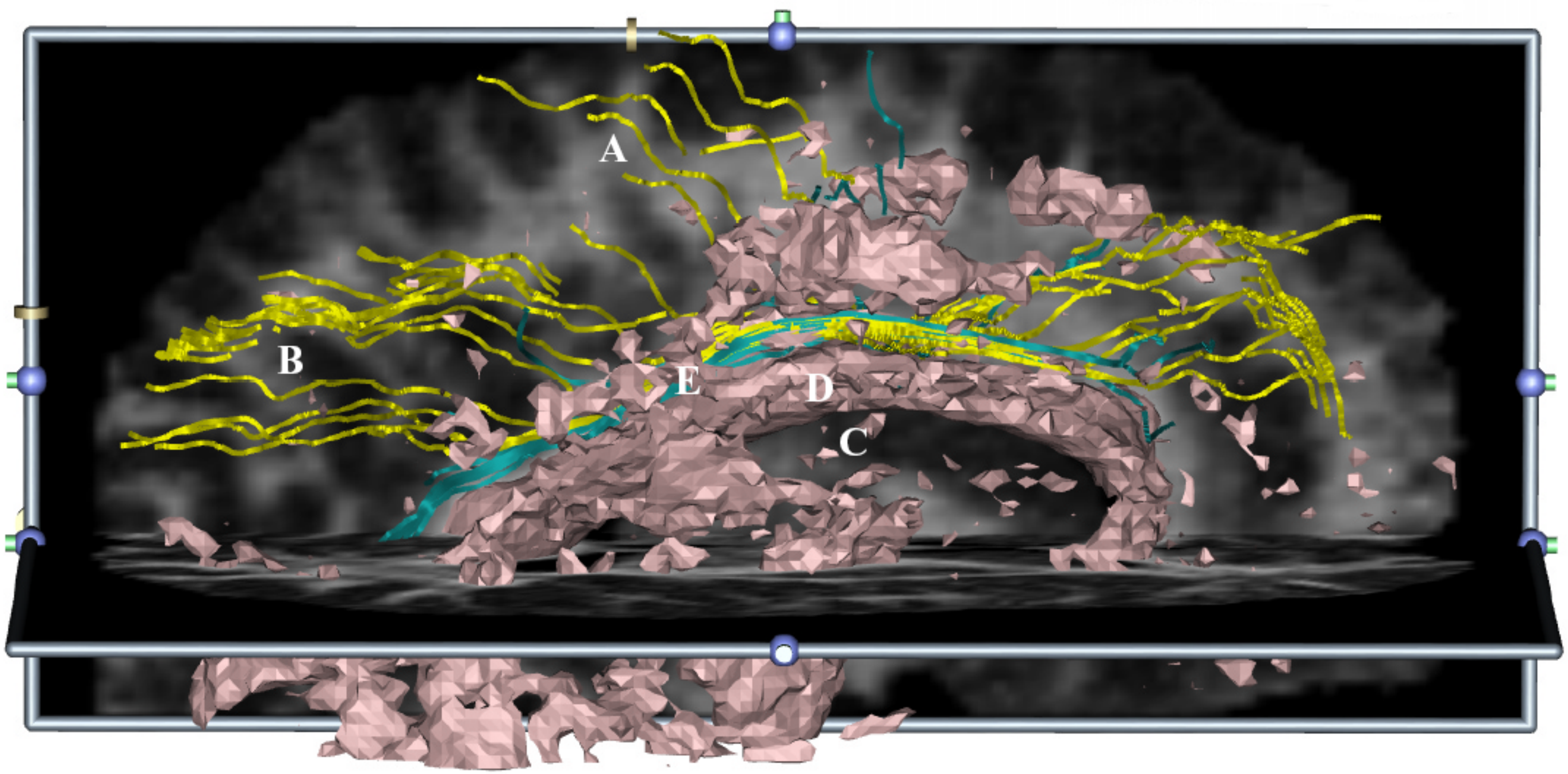


Tensorline
Hyperstreamline
Arrows: major eigenvector

- Advection vector
- Stabilization of propagation by considering
  - Input velocity vector
  - Output velocity vector (after application of tensor operation)
  - Vector along major eigenvector
- Weighting of three components depends on anisotropy at specific position:
  - Linear anisotropy: only along major eigenvector
  - Other cases: input or output vector

- Tensorlines [Weinstein, Kindlmann 1999]
  - Advection vector
  - Stabilization of propagation by considering
    - Input velocity vector
    - Output velocity vector (after application of tensor operation)
    - Vector along major eigenvector
  - Weighting of three components depends on anisotropy at specific position:
    - Linear anisotropy: only along major eigenvector
    - Other cases: input or output vector

- Tensorlines



Tensorlines: Yellow   Hyperstreamlines: Cyan

Summary

- Spatial visualization is diverse
  - Scalar field vis: how to *interpolate* space.
  - Vector, tensor field vis: how to *integrate* space.
  - Multifields: how to summarize heterogeneous phenomena.
  - Topology: how to represent connectivity.
- Until now, this field has been very technique-driven…
- Are there ways to make it more process-driven?
  (i.e., "design" for spatial visualization?)