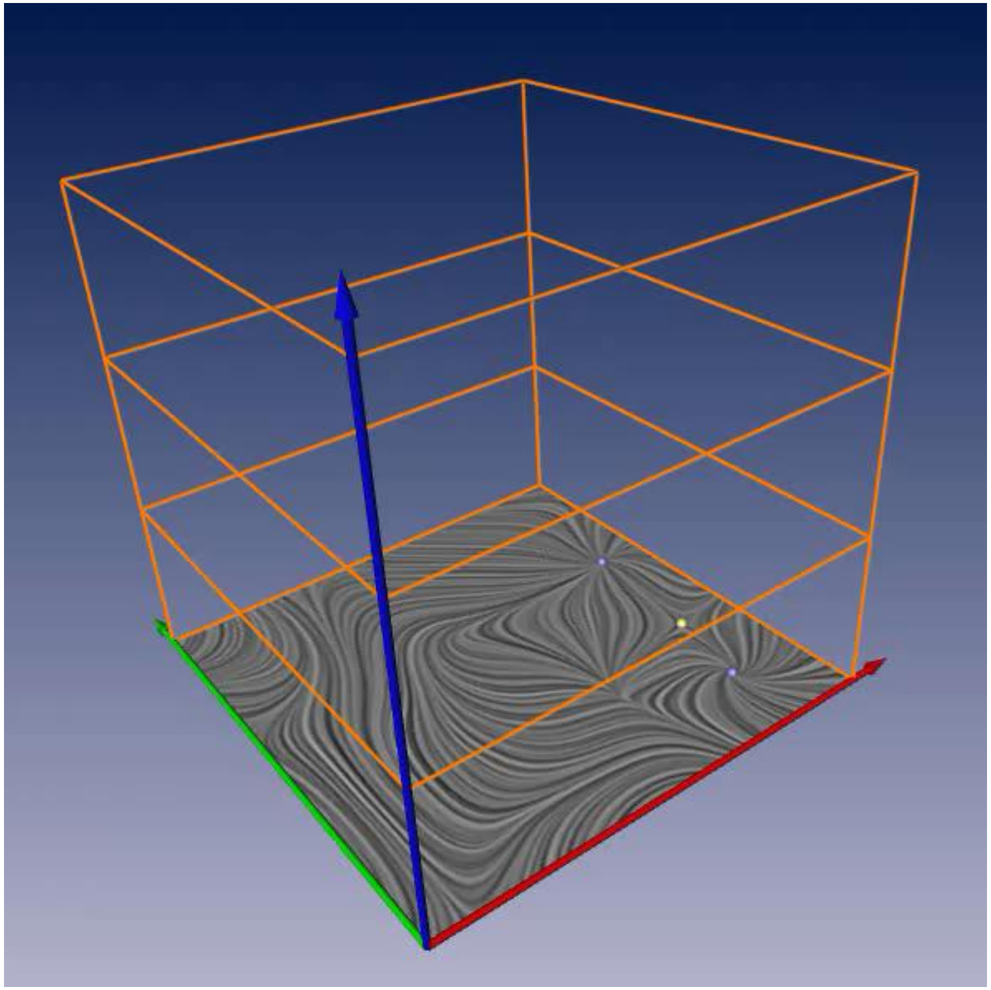
# *Feature Flow Fields Tutorial*

Speaker: Maik Schulze
University of Magdeburg, Germany

Visual Computing Research Group:
http://vc.cs.ovgu.de/

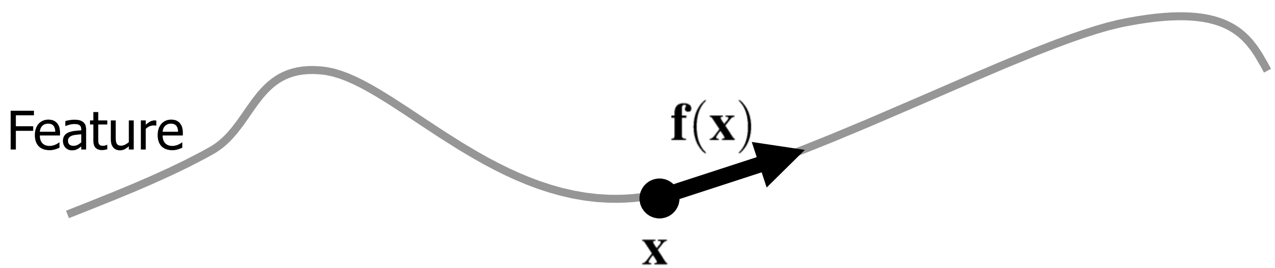
Based on slides from:
Holger Theisel
Tino Weinkauf

# Motivation



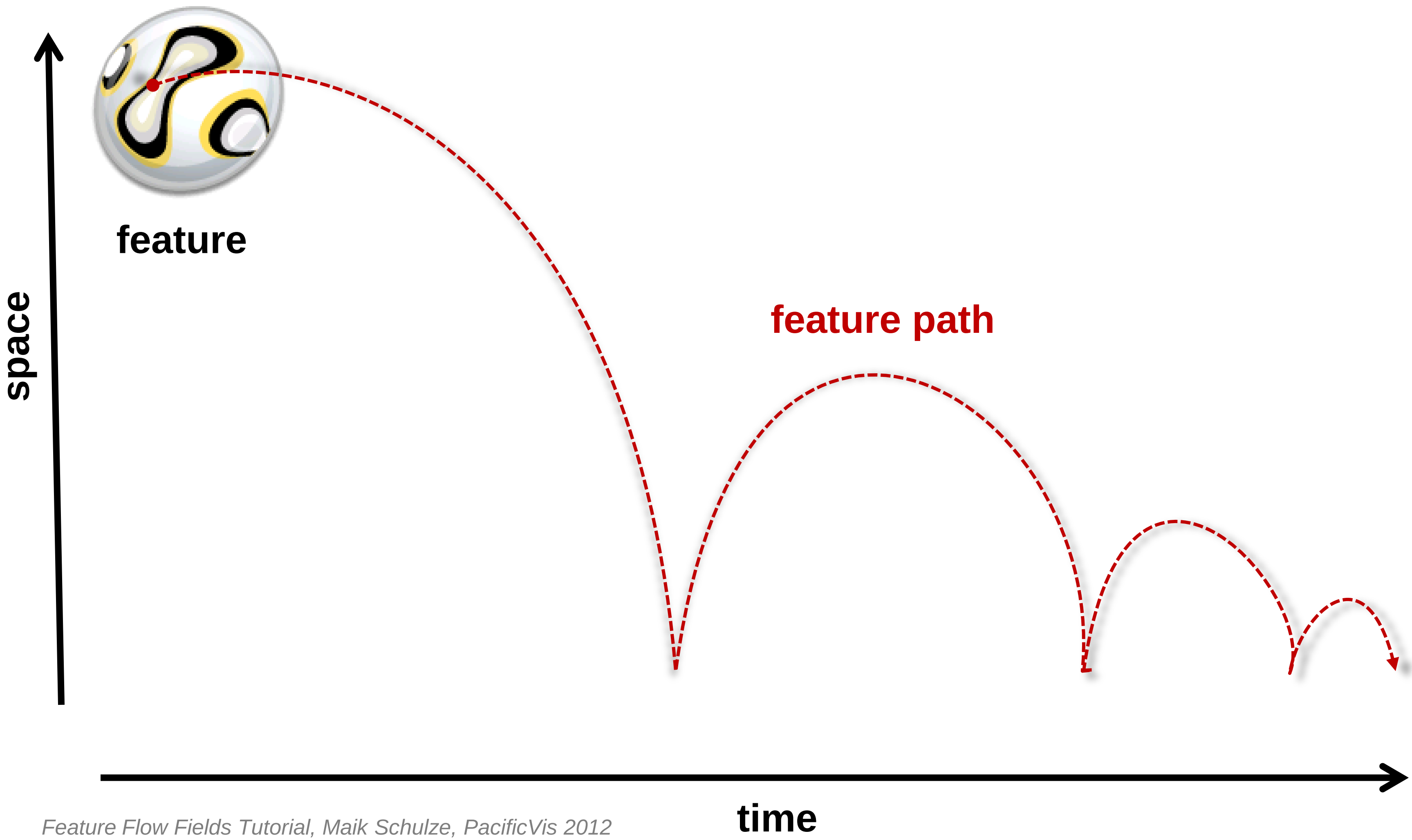
# *Motivation*

- Capture topological information over time
  - Critical points
  - Periodic orbits
  - Vortex axes
- Data set can be scalar, vector or tensor field
- Represent dynamic behavior of features as integration in steady vector field
  - Stream lines, stream surfaces
  - Numerical stream line/surface integration is well-understood



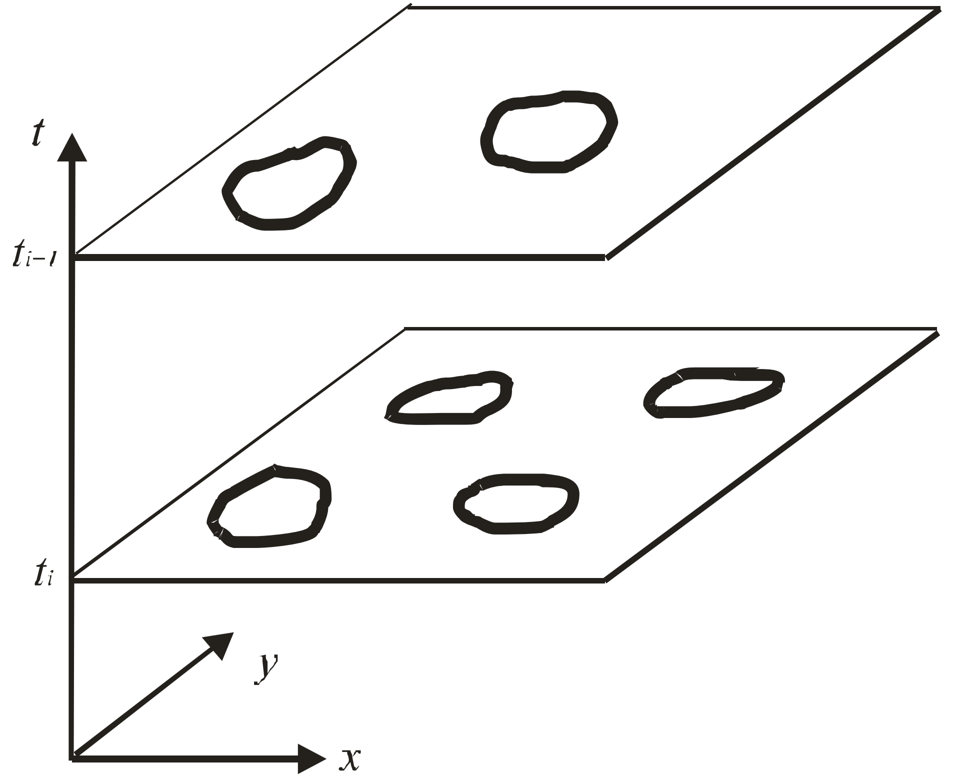
# *Feature Tracking*

Simple Example: Tracking the ball



# Feature Tracking

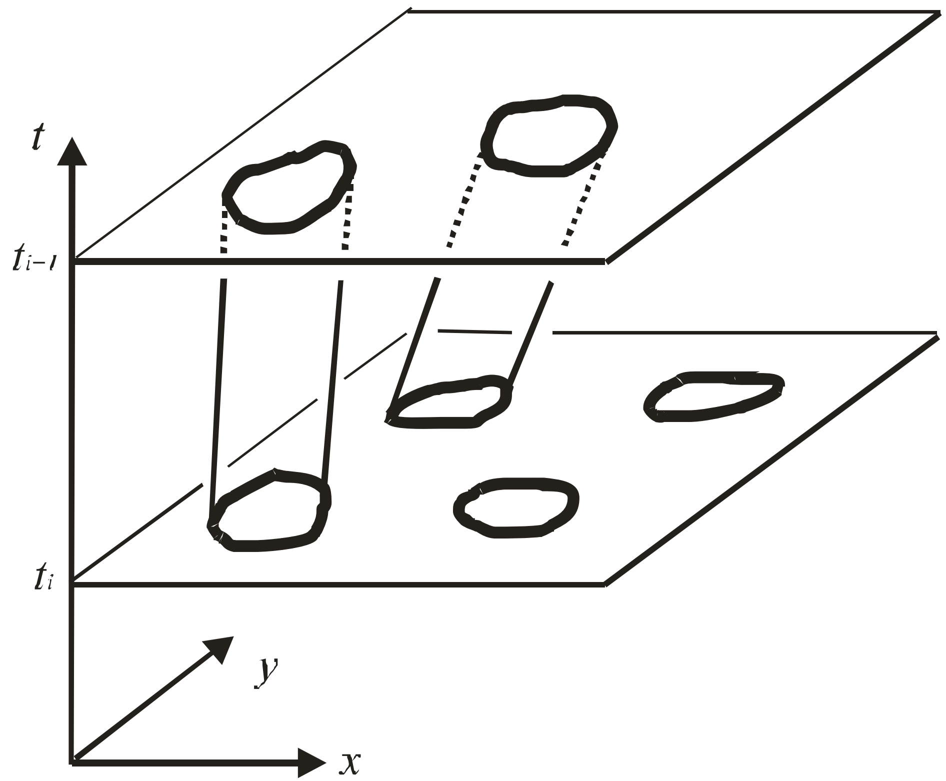
1. Extract features at different time steps



# *Feature Tracking*

1. Extract features at different time steps

2. Find corresponding features

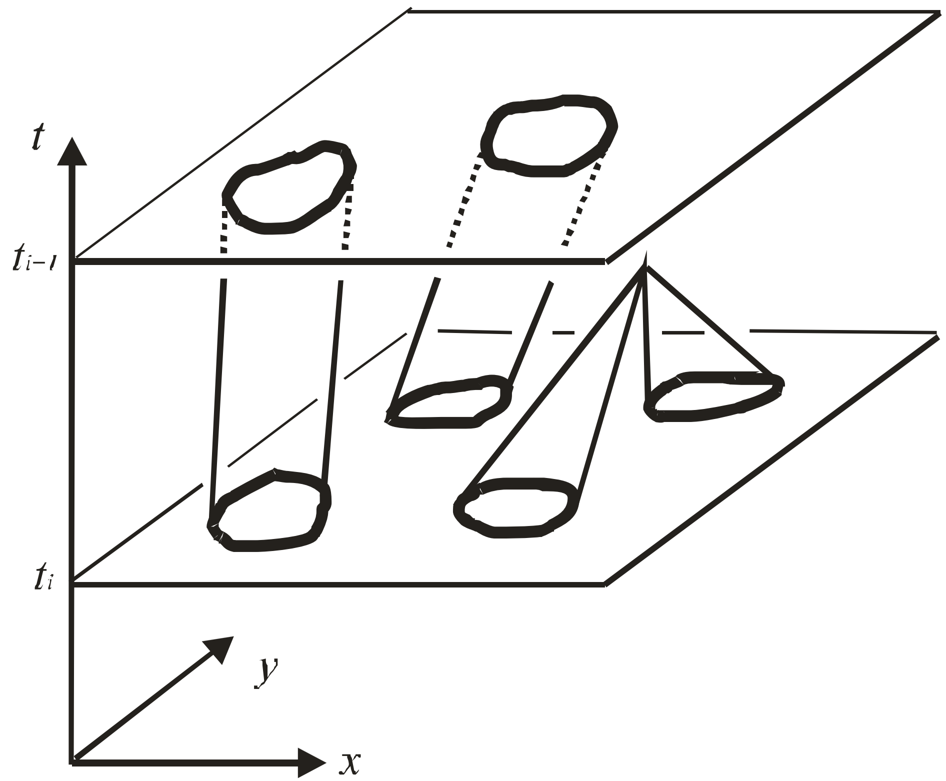


# *Feature Tracking*

1. Extract features at different time steps

2. Find corresponding features

3. Event detection
   - birth, death
   - entry, exit
   - split, merge



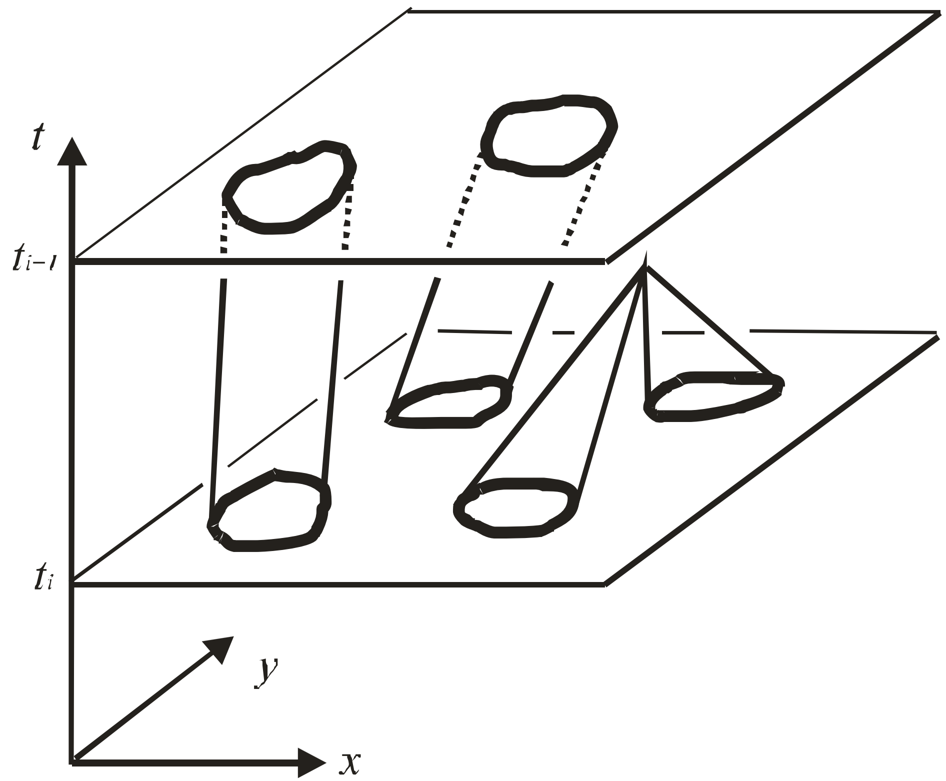
# Feature Tracking

1. Extract features at different time steps

2. Find corresponding features

3. Event detection
   - birth, death
   - entry, exit
   - split, merge

4. Visualization



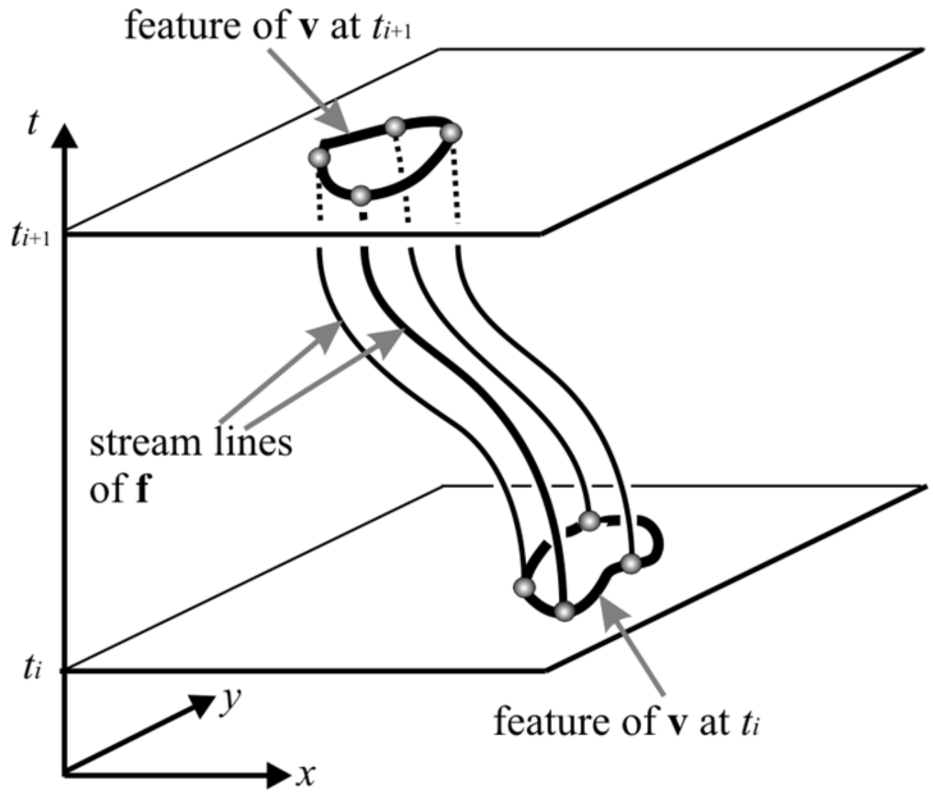
# Feature Tracking with FFF

- Consider a point **x** known to be part of a feature
  - FFF **f** is a well-defined vector field at **x**
  - **f** points into direction where the feature moves to
  - Streamline integration in **f** at **x** yields curve with all points on the feature

$$\mathbf{v}(x,y,t) = \begin{pmatrix} u(x,y,t) \\ v(x,y,t) \end{pmatrix}$$

$$\mathbf{f}(x,y,t) = \begin{pmatrix} f(x,y,t) \\ g(x,y,t) \\ h(x,y,t) \end{pmatrix}$$

**Feature Flow Field (FFF)**



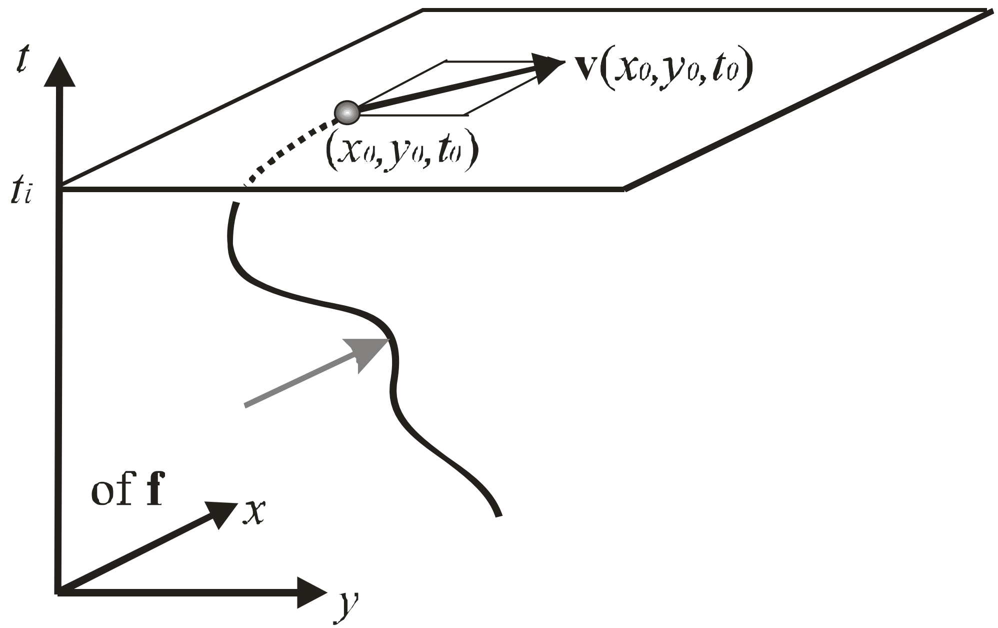
# Application:
# Critical Point Tracking

# Main Idea

$$\mathbf{v}(x,y,t) = \begin{pmatrix} u(x,y,t) \\ v(x,y,t) \end{pmatrix}$$

$$\Downarrow$$

$$\mathbf{f}(x,y,t) = \begin{pmatrix} f(x,y,t) \\ g(x,y,t) \\ h(x,y,t) \end{pmatrix}$$

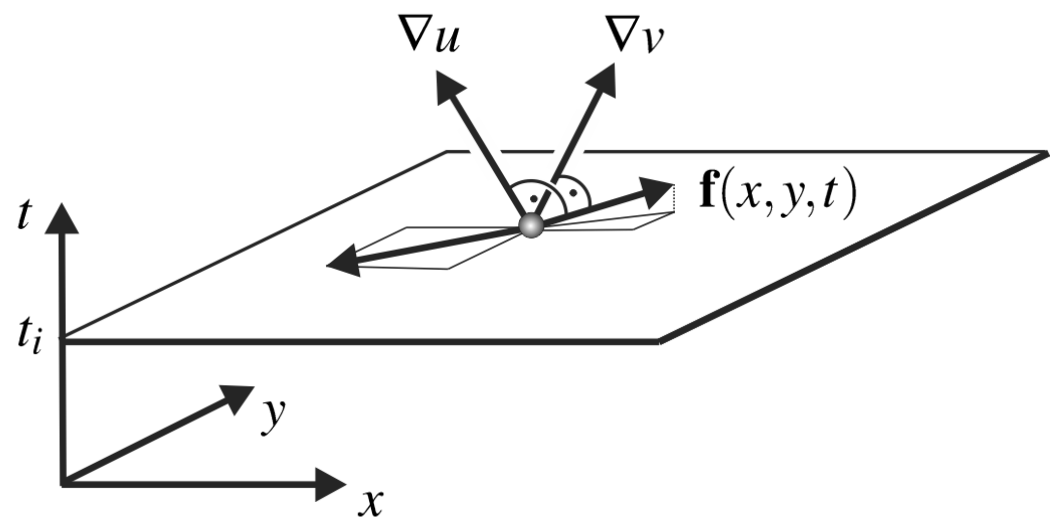



*Feature flow field* **f** *shall point into the direction where the original vector field* **v** *is not changing in terms of both magnitude and direction.*

# Tracking Critical Points in 2D

$$\mathbf{v}(x,y,t) = \begin{pmatrix} u(x,y,t) \\ v(x,y,t) \end{pmatrix}$$

unsteady 2D vector field




feature flow field

$$\mathbf{f}(x,y,t) = \begin{pmatrix} f(x,y,t) \\ g(x,y,t) \\ h(x,y,t) \end{pmatrix}$$

$$\mathbf{f}(x,y,t) = (\nabla u)^T \times (\nabla v)^T = \begin{pmatrix} \det(\mathbf{v}_y, \mathbf{v}_t) \\ \det(\mathbf{v}_t, \mathbf{v}_x) \\ \det(\mathbf{v}_x, \mathbf{v}_y) \end{pmatrix}$$

# *Tracking Critical Points in 2D*

- Given: unsteady 2D field $\mathbf{v}(x,y,t) = \begin{pmatrix} u(x,y,t) \\ v(x,y,t) \end{pmatrix}$

- Goal: Find trajectories of all critical points

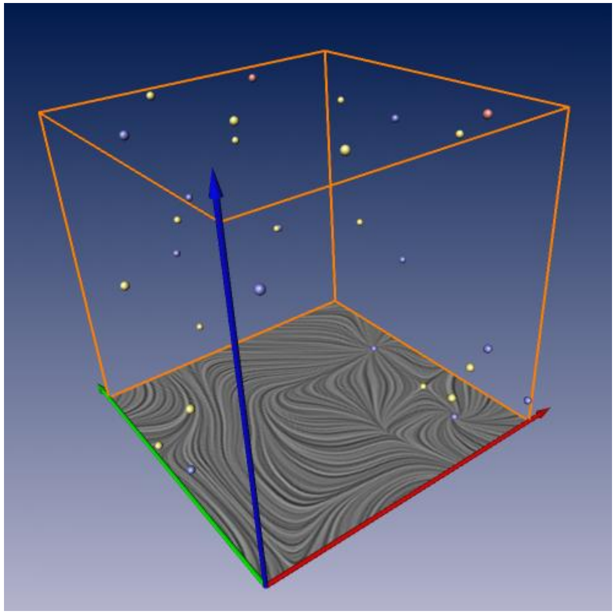
# Tracking Critical Points in 2D

1. Find all seed points
   - Domain boundaries

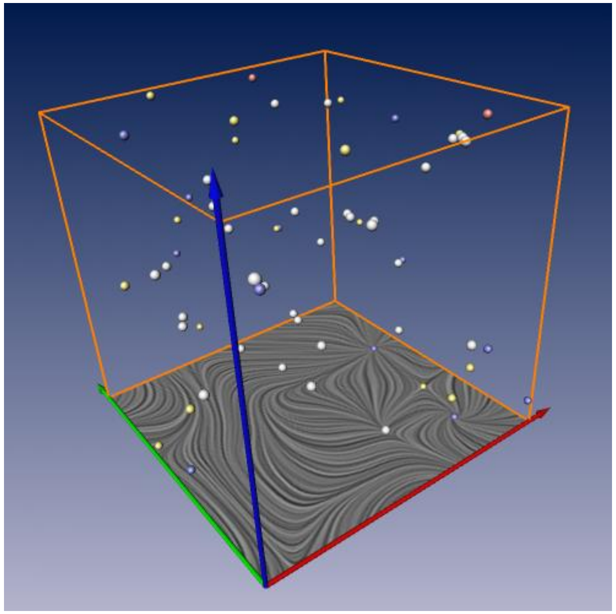
$$\mathbf{v}(x,y,t_{min}) = (0,0)^T \text{ and } \mathbf{v}(x,y,t_{max}) = (0,0)^T$$
$$\mathbf{v}(x,y_{min},t) = (0,0)^T \text{ and } \mathbf{v}(x,y_{max},t) = (0,0)^T$$
$$\mathbf{v}(x_{min},y,t) = (0,0)^T \text{ and } \mathbf{v}(x_{max},y,t) = (0,0)^T$$

   - Fold bifurcations

$$[\ \mathbf{v}(\mathbf{x}) = (0,0)^T\ ,\ \ \det(\mathbf{J_v}(\mathbf{x})) = 0\ ]$$



On domain boundaries

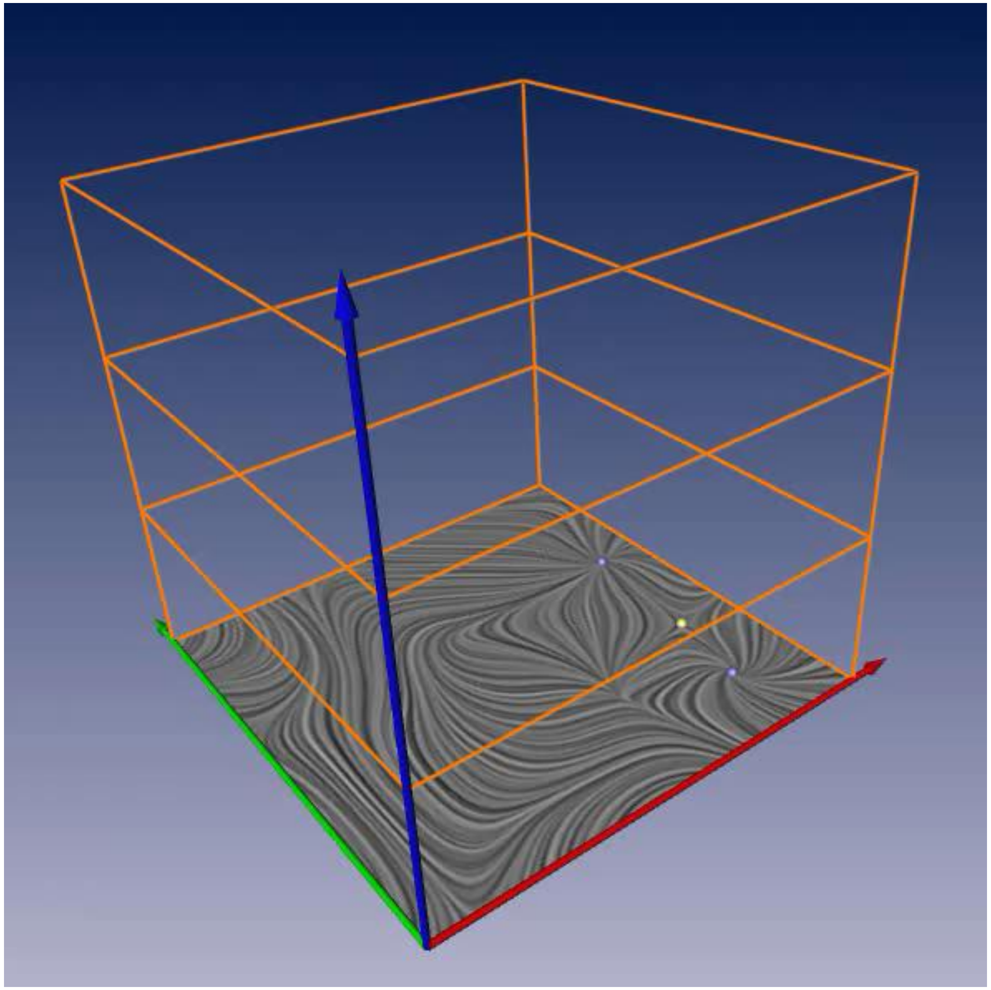


All seed points

# Tracking Critical Points in 2D

$$\mathbf{v}(x,y,t) = \begin{pmatrix} u(x,y,t) \\ v(x,y,t) \end{pmatrix}$$

$$\mathbf{f}(x,y,t) = \begin{pmatrix} \det(\mathbf{v}_y, \mathbf{v}_t) \\ \det(\mathbf{v}_t, \mathbf{v}_x) \\ \det(\mathbf{v}_x, \mathbf{v}_y) \end{pmatrix}$$



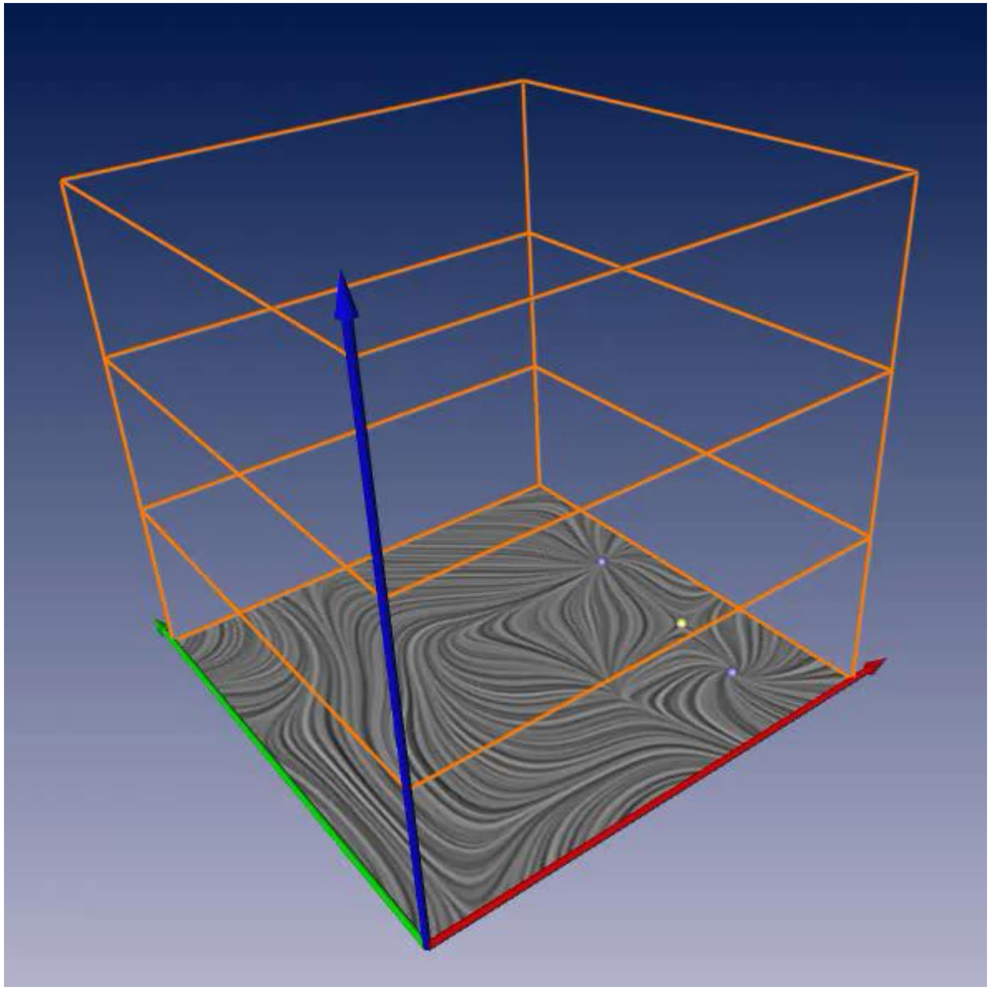
- Optimization
  - One sweep through the data
  - Only two time slices at once in memory

# Tracking Critical Points in 2D

$$\mathbf{v}(x,y,t) = \begin{pmatrix} u(x,y,t) \\ v(x,y,t) \end{pmatrix}$$

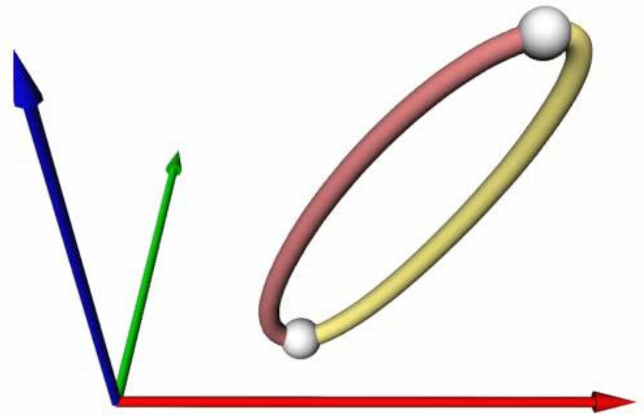
$$\mathbf{f}(x,y,t) = \begin{pmatrix} \det(\mathbf{v}_y, \mathbf{v}_t) \\ \det(\mathbf{v}_t, \mathbf{v}_x) \\ \det(\mathbf{v}_x, \mathbf{v}_y) \end{pmatrix}$$

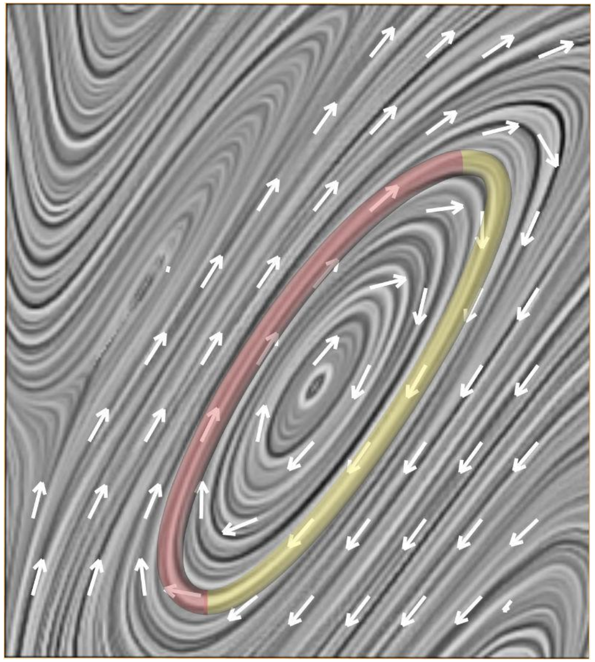
- Optimization
  - One sweep through the data
  - Only two time slices at once in memory

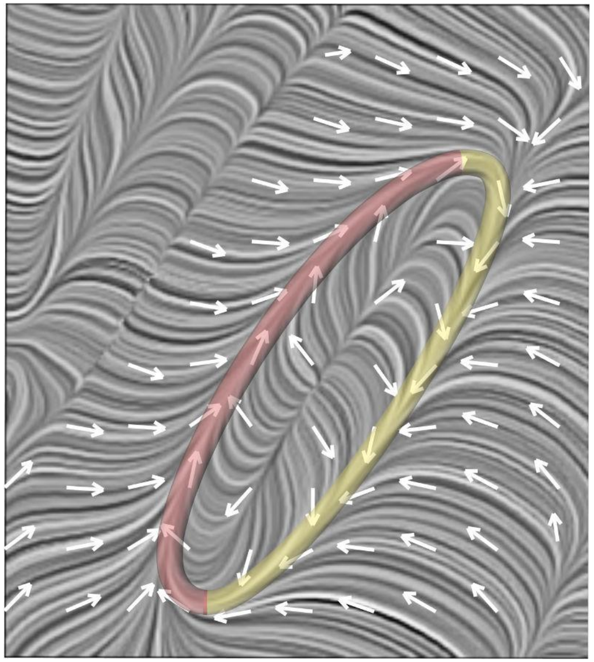


# *Stability of FFF*

- Streamlines in **f** have possibly diverging behavior
- In long integration times, streamline diverges from actual feature line



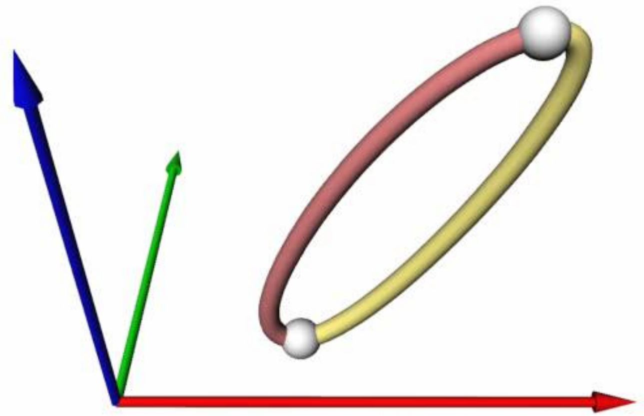


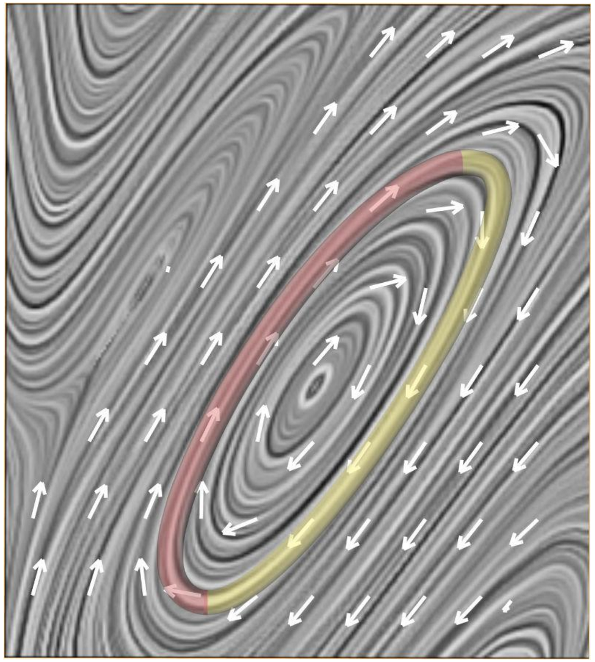


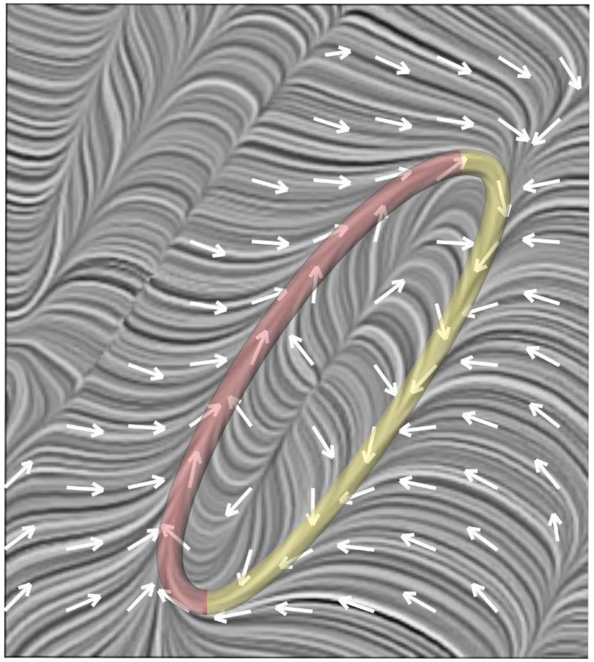
*Feature Flow Fields Tutorial, Maik Schulze, PacificVis 2012*

# *Stability of FFF*

- Streamlines in **f** have possibly diverging behavior
- In long integration times, streamline diverges from actual feature line





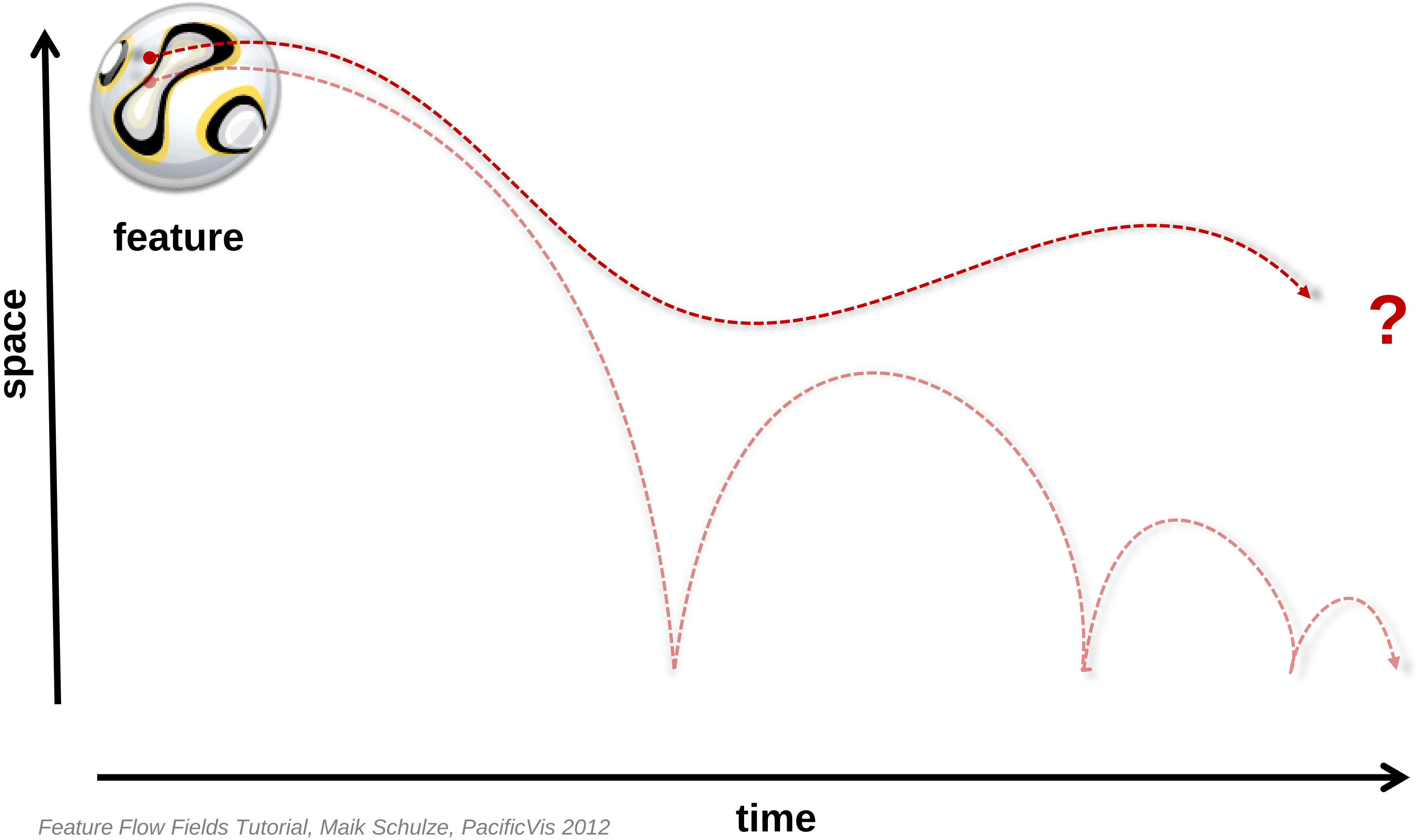


Feature Flow Fields Tutorial, Maik Schulze, PacificVis 2012

# Stable FFF

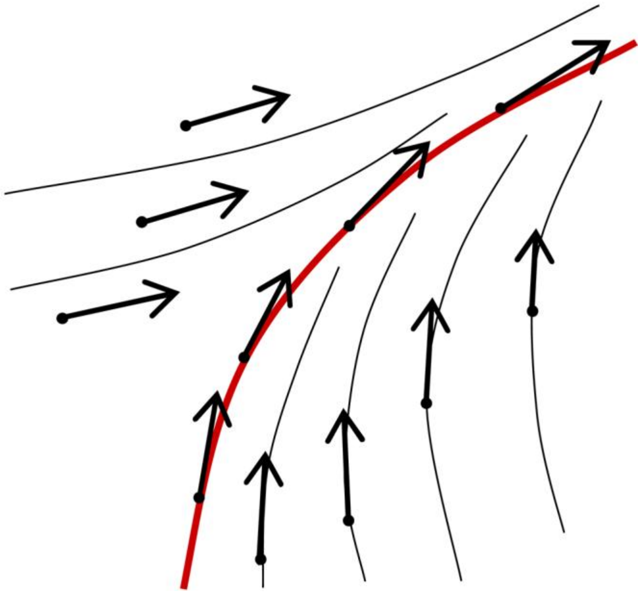
# Feature Tracking

Simple Example: Tracking the ball

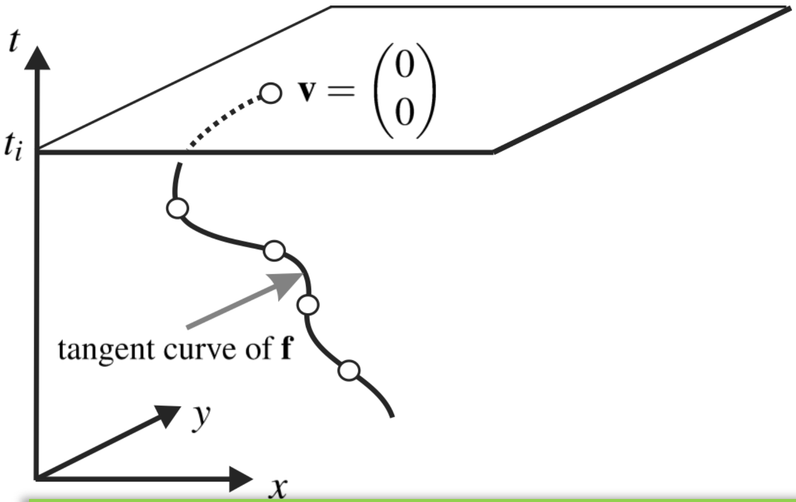


# Stable Feature Flow Fields

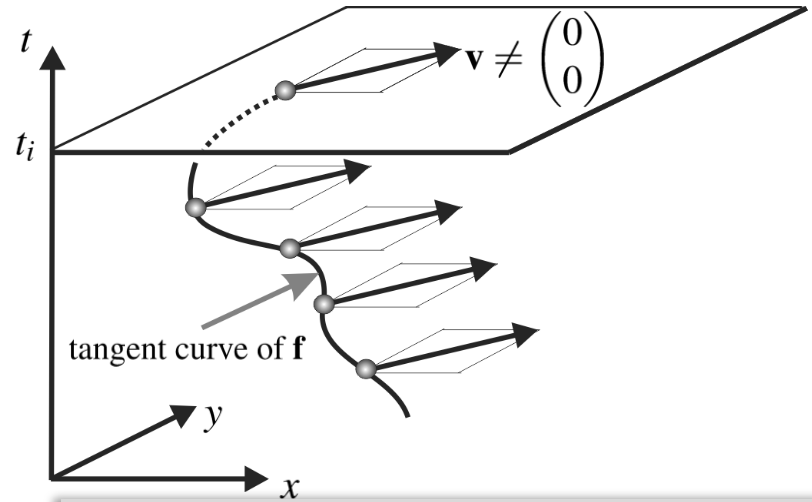
- Mathematically correct
- Numerically stable
  - Converging flow behavior near the feature line
  - Automatic correction of small errors
- Feature tracking still stream line integration
- Seeding must not be as precisely computed as in FFF



# *Stable Tracking Critical Points in 2D*



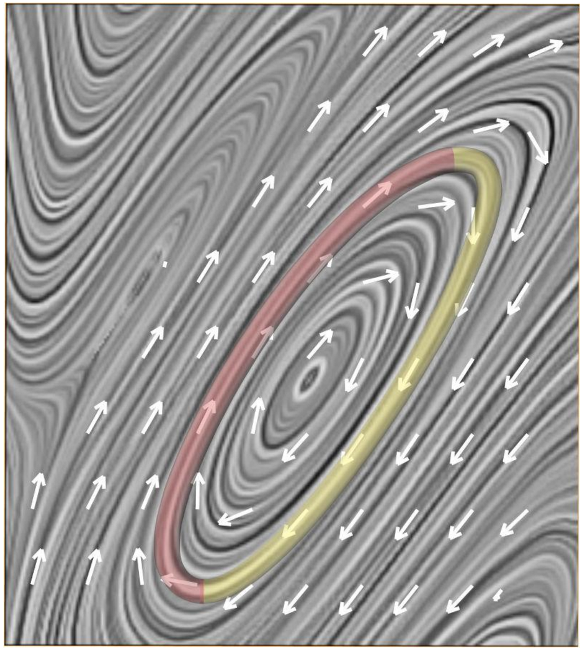

**Required**

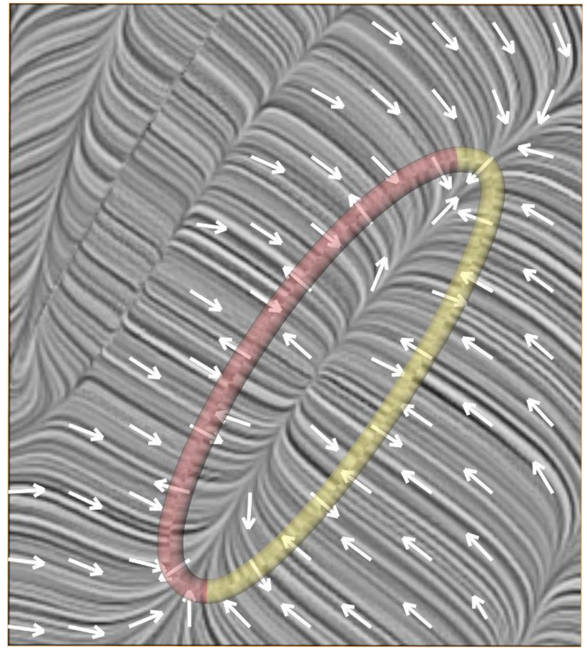


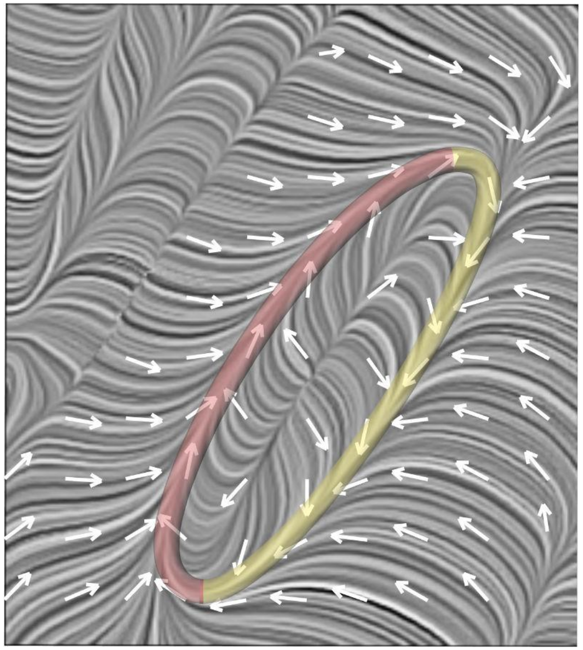

**Not necessary**

$$\mathbf{f}(x,y,t) = (\nabla u)^T \times (\nabla v)^T = \begin{pmatrix} \det(\mathbf{v}_y, \mathbf{v}_t) \\ \det(\mathbf{v}_t, \mathbf{v}_x) \\ \det(\mathbf{v}_x, \mathbf{v}_y) \end{pmatrix}$$

# *Stable Tracking Critical Points in 2D*







$$\mathbf{f}(x,y,t) = (\nabla u)^T \times (\nabla v)^T$$

$$\mathbf{g} = \frac{\mathbf{f}}{\|\mathbf{f}\|} \times \begin{pmatrix} \det(\mathbf{v}, \mathbf{v}_x) \\ \det(\mathbf{v}, \mathbf{v}_y) \\ \det(\mathbf{v}, \mathbf{v}_t) \end{pmatrix}$$

$$\mathbf{h} = \mathbf{f} + \alpha\, \mathbf{g}$$

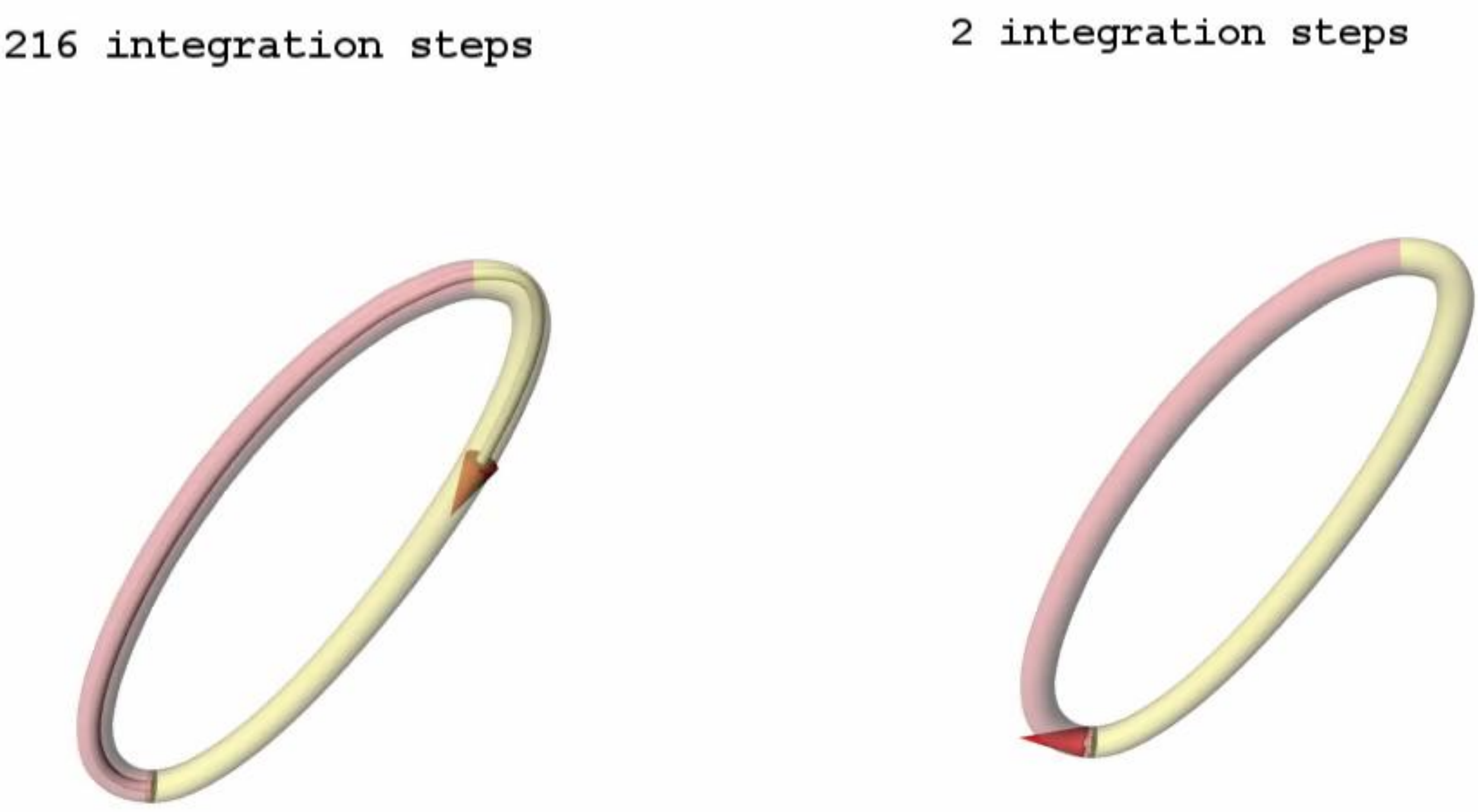
acts as sink around $\mathbf{v} = \mathbf{0}$

Original FFF

$$\mathbf{v} = \mathbf{0} \;\Rightarrow\; \mathbf{g} = \mathbf{0}$$

***Stable Feature Flow Field***

# Stable Tracking Critical Points in 2D

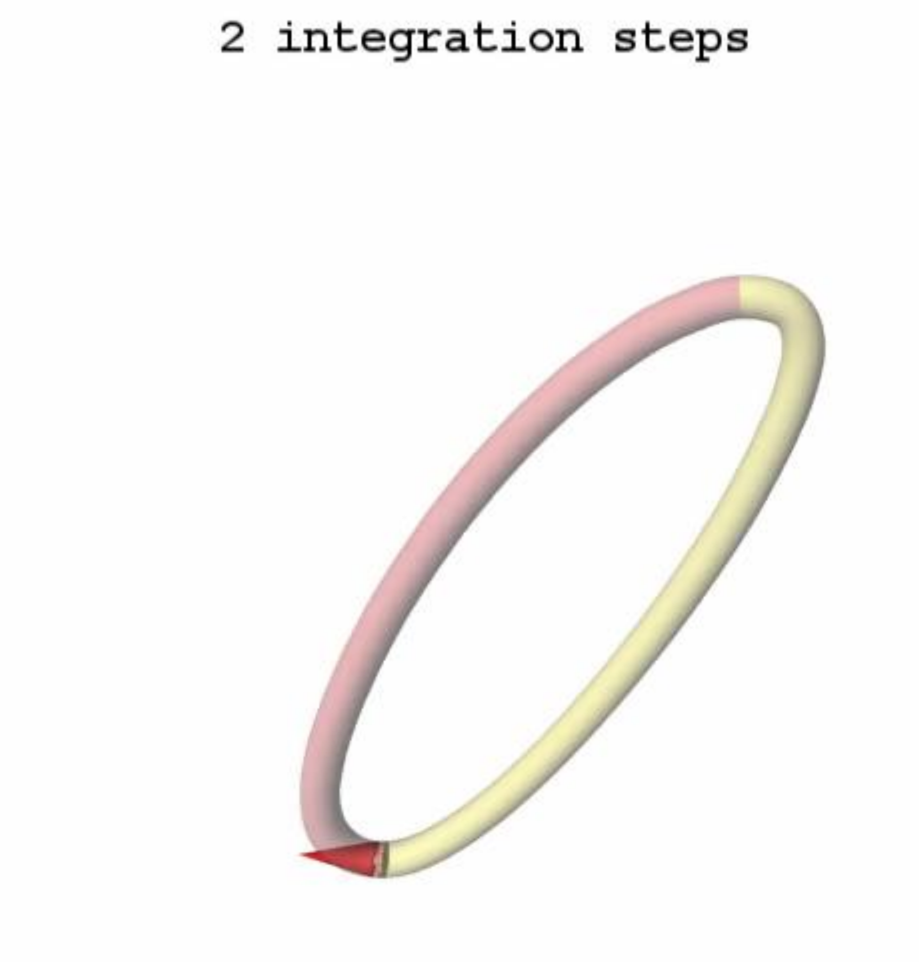



Feature Flow Fields

Stable Feature Flow Fields

# *Stable Tracking Critical Points in 2D*


Feature Flow Fields



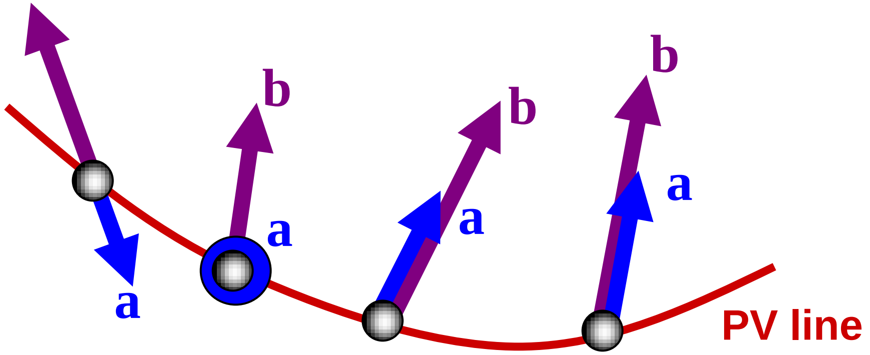

Stable Feature Flow Fields

# Stable Feature Flow Fields for Parallel Vector Lines

# *Parallel Vectors with Stable FFF*

- **Parallel Vectors (PV) operator**
  - Generic approach to feature extraction
  - Swirling motion cores, ridges, …

$$\mathbf{w}_1 \parallel \mathbf{w}_2$$



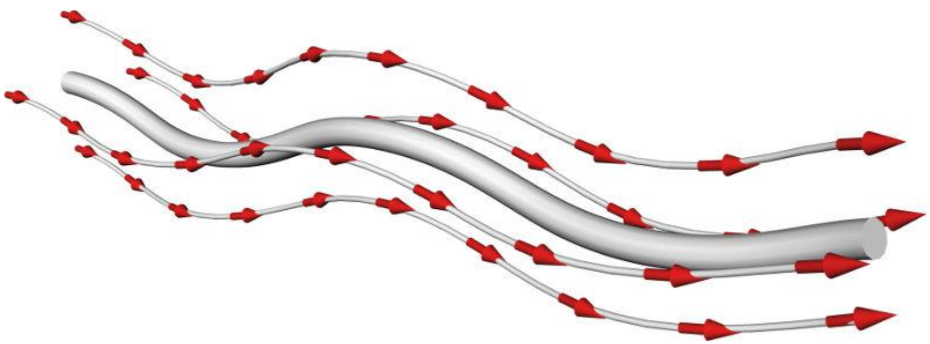

- Every PV problem can be reformulated using FFF

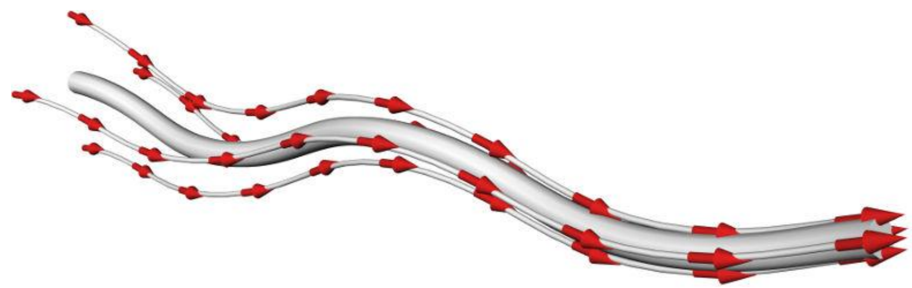
$$\mathbf{f} = \begin{pmatrix} \det(\mathbf{q}_y, \mathbf{q}_z, \mathbf{a}) \\ \det(\mathbf{q}_z, \mathbf{q}_x, \mathbf{a}) \\ \det(\mathbf{q}_x, \mathbf{q}_y, \mathbf{a}) \end{pmatrix} \quad \text{with} \quad \mathbf{q} = \mathbf{w}_1 \times \mathbf{w}_2$$

# *Parallel Vectors with Stable FFF*

Correction Field

$$\mathbf{g} = \frac{\mathbf{f}}{\|\mathbf{f}\|} \times \begin{pmatrix} \det(\mathbf{q}, \mathbf{q}_x, \mathbf{a}) \\ \det(\mathbf{q}, \mathbf{q}_y, \mathbf{a}) \\ \det(\mathbf{q}, \mathbf{q}_z, \mathbf{a}) \end{pmatrix}$$





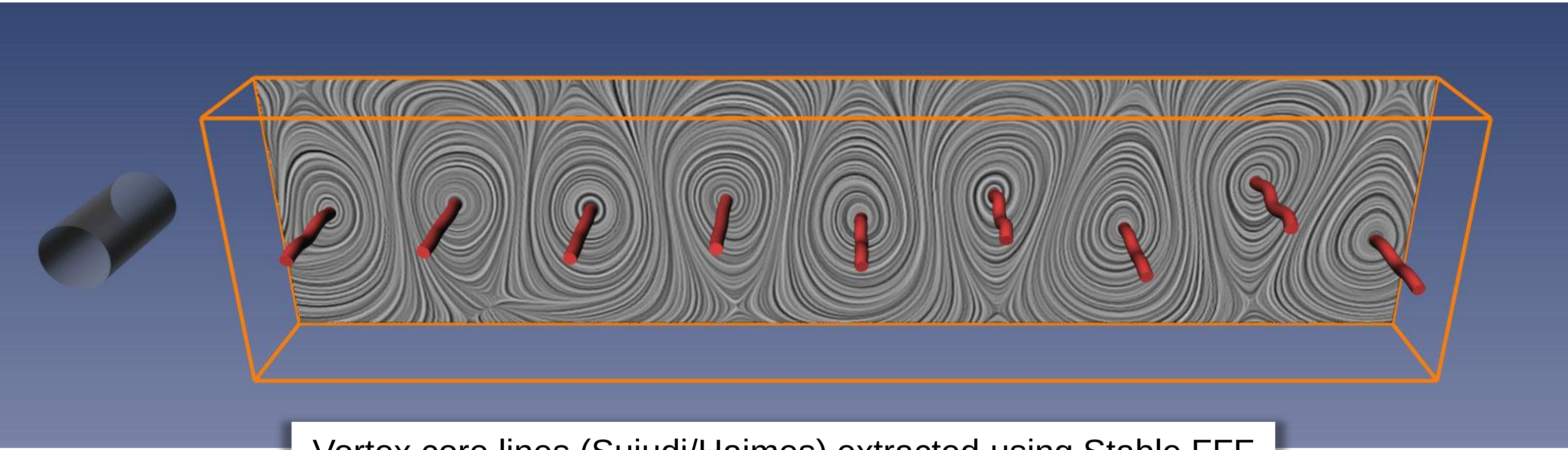
$$\mathbf{f} = \begin{pmatrix} \det(\mathbf{q}_y, \mathbf{q}_z, \mathbf{a}) \\ \det(\mathbf{q}_z, \mathbf{q}_x, \mathbf{a}) \\ \det(\mathbf{q}_x, \mathbf{q}_y, \mathbf{a}) \end{pmatrix}$$

$$\boxed{\mathbf{h} = \mathbf{f} + \alpha\, \mathbf{g}}$$

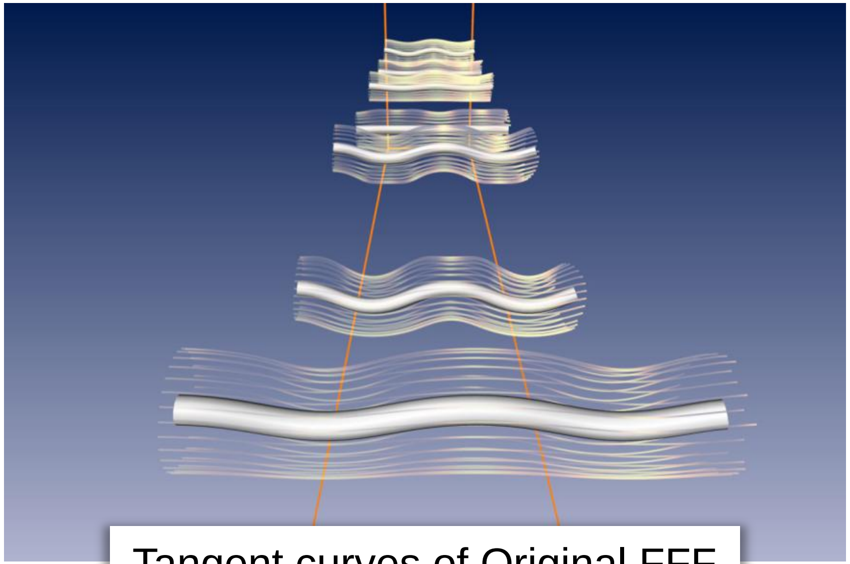
Original FFF

***Stable Feature Flow Field***

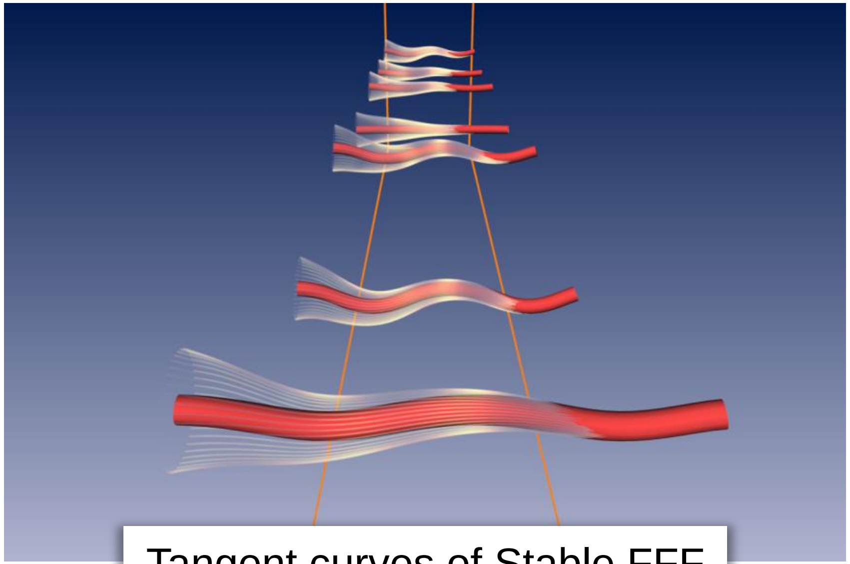
# Parallel Vectors with Stable FFF



Vortex core lines (Sujudi/Haimes) extracted using Stable FFF



Tangent curves of Original FFF



Tangent curves of Stable FFF

# *Summary*

- Dynamic behavior of certain features can be described by streamline integration in Feature Flow Fields
  - Critical points
  - Periodic orbits
  - Vortex axes

- Numerical stability can be improved by Stable Feature Flow Fields

# *Feature Flow Fields Tutorial*

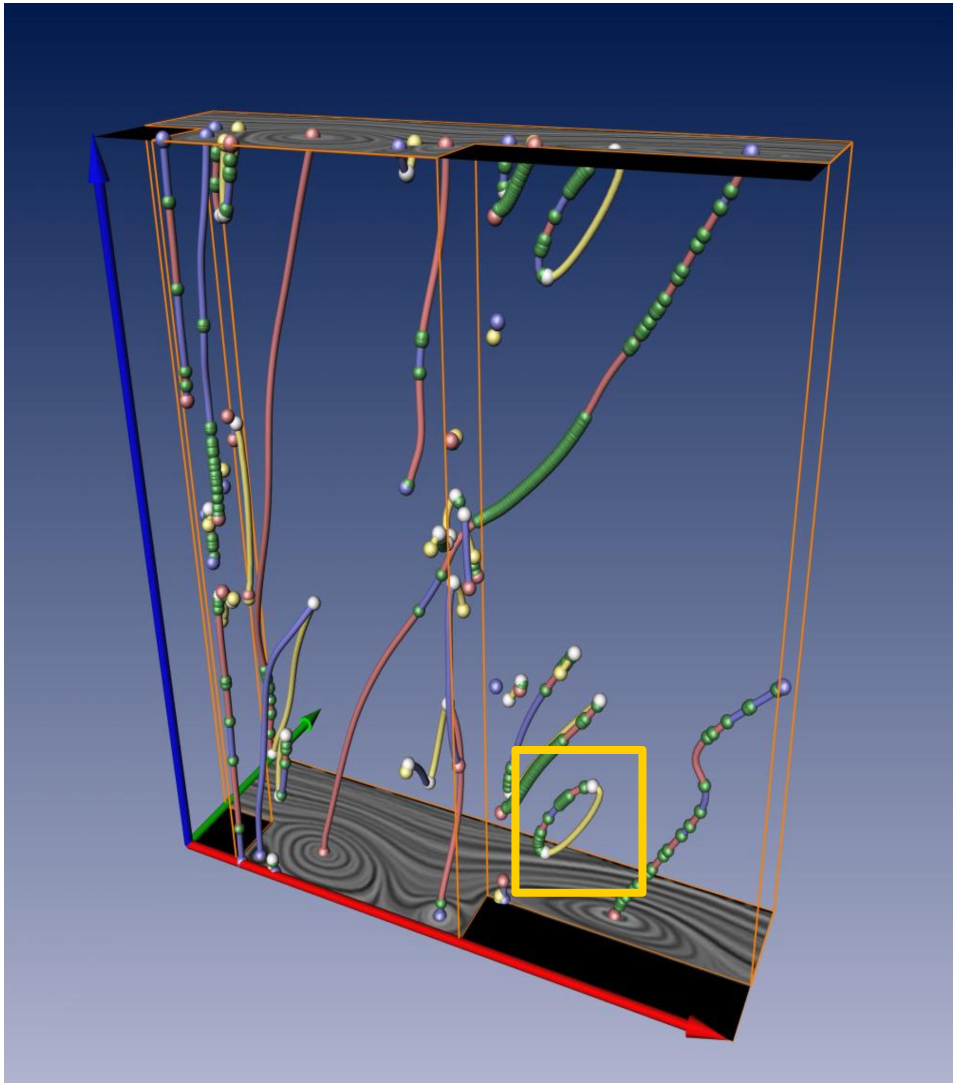
Speaker: Maik Schulze
University of Magdeburg, Germany

Visual Computing Research Group:
http://vc.cs.ovgu.de/

Based on slides from:
Holger Theisel
Tino Weinkauf

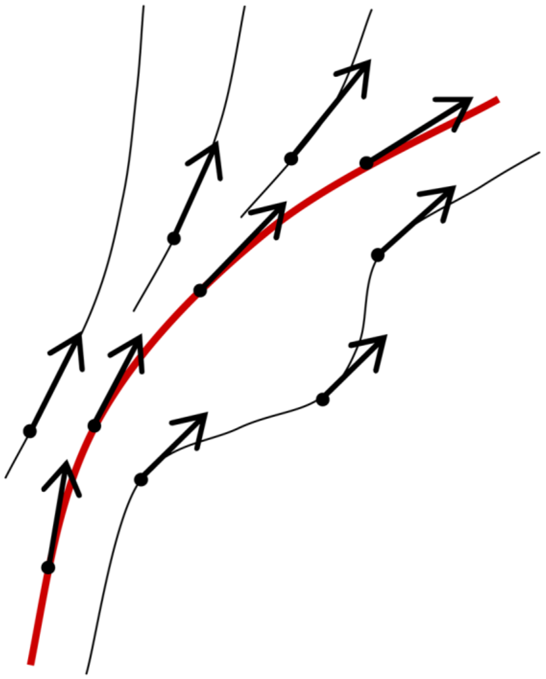
# Stability of FFF

# Stability of FFF



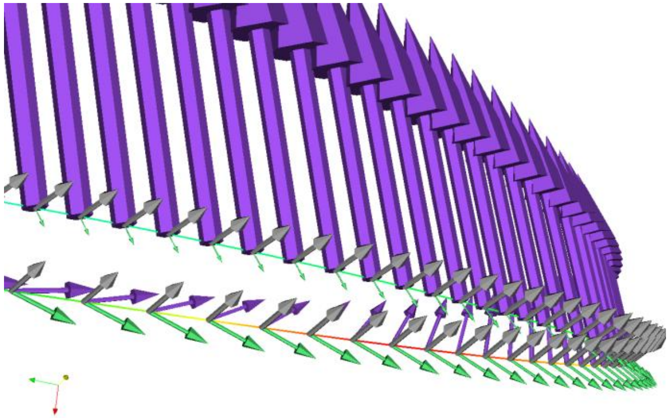
# *Stability of FFF*

- Mathematically correct
- Possibly numerically unstable
  - Possibly divergent behavior near feature line
  - Numerical errors may move integration off the feature



# Improvement of FFF

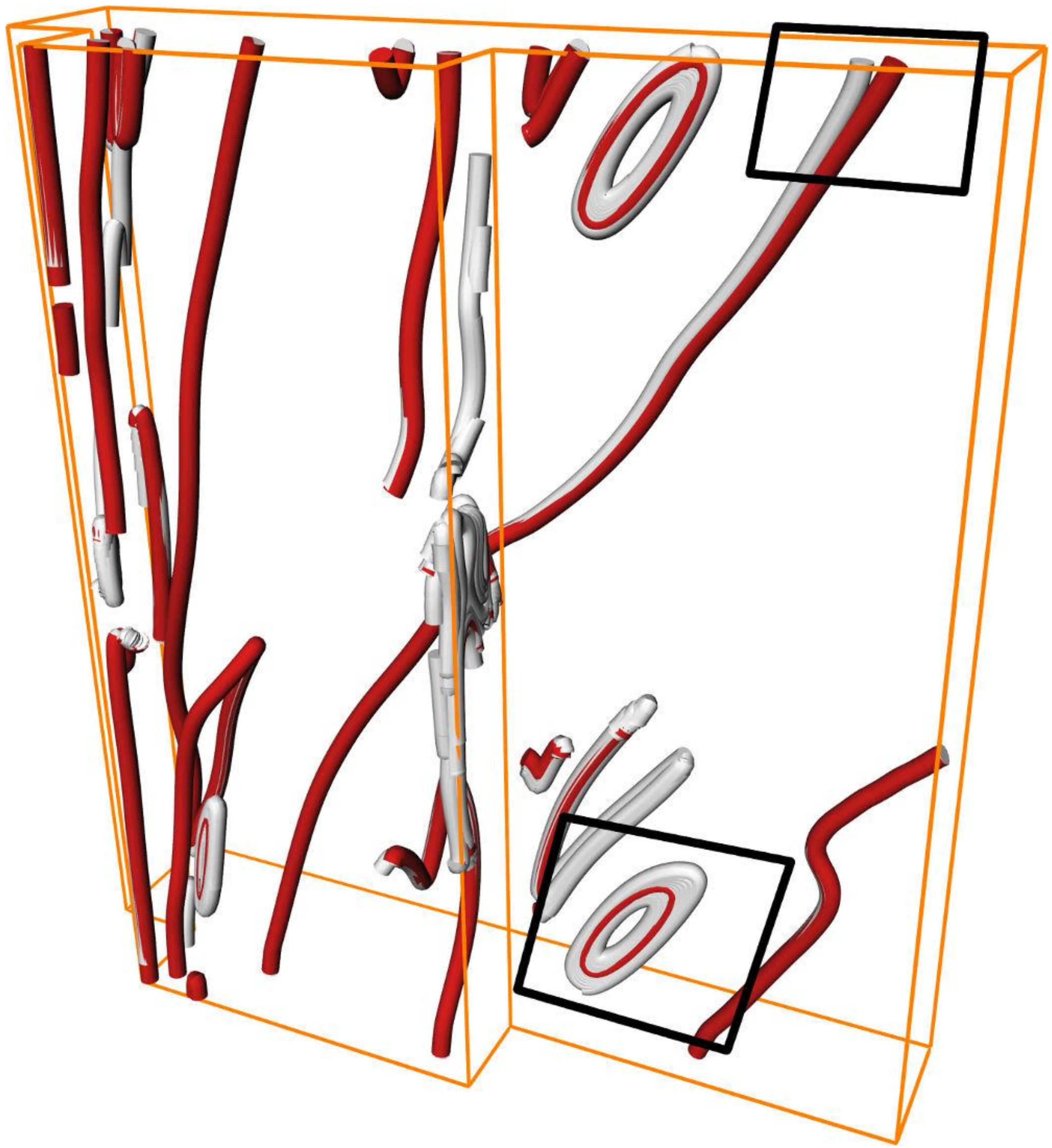
- Predictor-Corrector approach
  - Do integration step
  - Move particle to feature by root finding



**A. Van Gelder, A. Pang**
*Using PVsolve to Analyze and Locate Positions of Parallel Vectors*,
IEEE TVCG 15(4):682-695, 2009.

- Disadvantage
  - Root finding computationally expensive
  - Feature tracking not stream line integration anymore

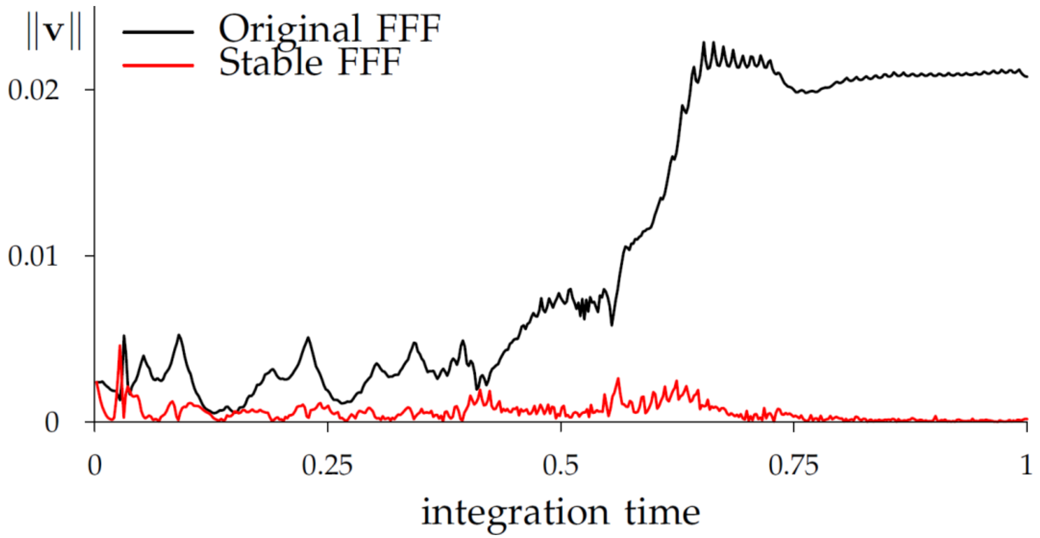
# *Comparison between FFF and Stable FFF*

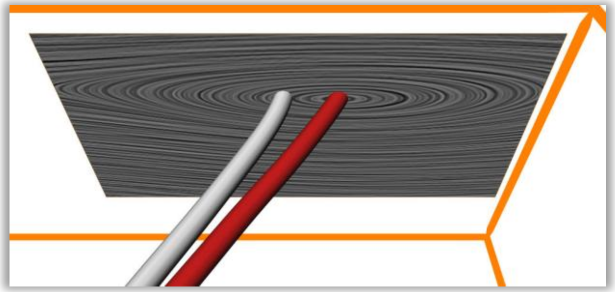


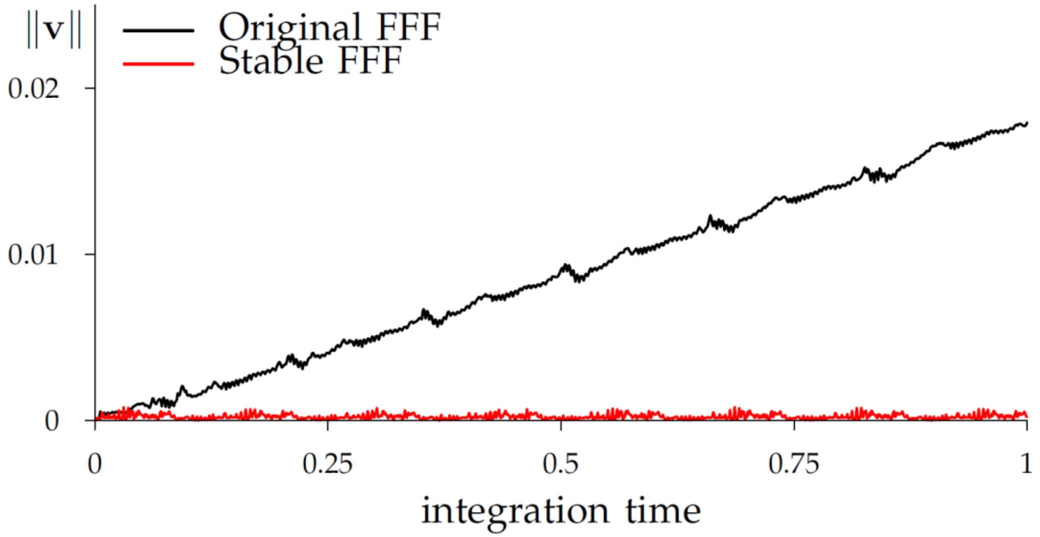
**Original FFF**

**Stable FFF**

# Comparison between FFF and Stable FFF






# Acknowledgements

- **Based on the following references:**

  - **Scientific Visualization** — **Tino Weinkauf,** MPI Saarbrücken, 2012
  - **Flow and Tensor Visualization** — **Holger Theisel,** University of Magdeburg, 2011
  - **Flow Visualization** — **Helwig Hauser,** University of Bergen, 2011

The project **SemSeg** acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number 226042.

# *Selected Work From Feature Flow Fields*


- **H. Theisel and H.-P. Seidel**
  *Feature Flow Fields*, VisSym 2003
- **H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel**
  *Stream Line and Path Line Oriented Topology for 2D Time-Dependent Vector Fields*, IEEE Vis 2004
- **T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel**
  *Feature Flow Fields in Out-Of-Core Settings*, TopoInVis 2005
- **X. Zheng and A. Pang**
  *Topological Lines in 3D Tensor Fields and Discriminant Hessian Factorization*,
  TVCG 2005
- **H. Theisel, J. Sahner, T. Weinkauf, H.-C. Hege, and H.-P. Seidel**
  *Extraction of Parallel Vector Surfaces in 3D Time-Dependent Fields and Application to Vortex Core Line Tracking*, IEEE Vis 2005
- **S. Depardon, J. J. Lasserre, L. E. Brizzi, and J. Borée**
  *Automated topology classification method for instantaneous velocity fields*, Experiments in Fluids 2007
- **T. Weinkauf and H. Theisel and A. Van Gelder and A. Pang**
  Stable Feature Flow Fields,
  TVCG 2010
- **J. Reininghaus, J. Kasten, T. Weinkauf, I. Hotz**
  *Efficient Computation of Combinatorial Feature Flow Fields*,
  TVCG 2011