Over Two Decades of Integration-Based, Geometric Vector Field Visualization

Tony McLoughlin[1], Robert S. Laramee[1], Ronald Peikert[2], Frits H. Post[3], and Min Chen[1]

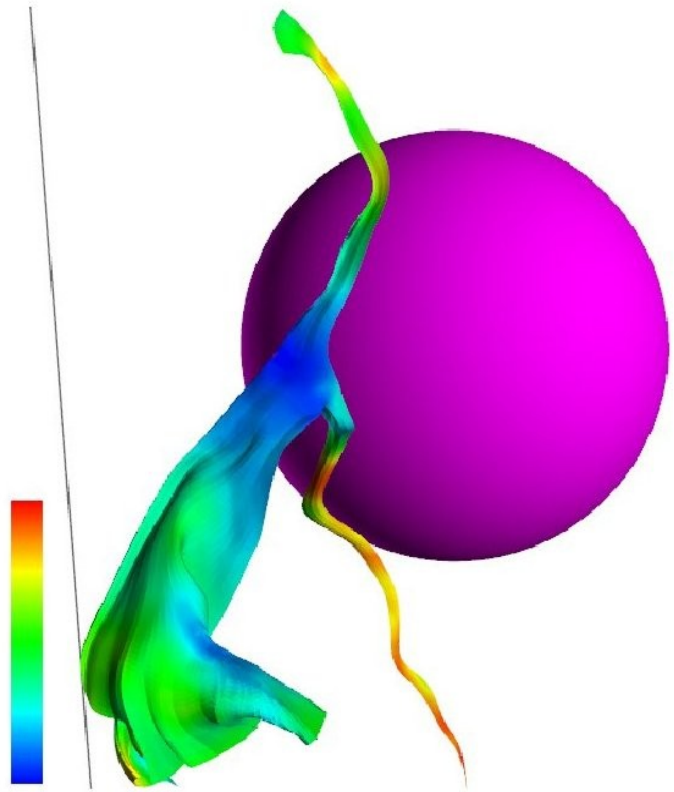
[1]The Visual and Interactive Computing Group
Computer Science Dept.
Swansea University
Swansea, Wales, UK
{R.S.Laramee,cstony, M.Chen} "at" swansea.ac.uk

[2]Dept. of Computer Science
ETH Zentrum
Zürich, Switzerland
peikert "at" inf.ethz.ch

[3]Computer Graphics and CAD/CAM Group
Delft University of Technology
Delft, The Netherlands
frits.post "at" its.tudelft.nl

Overview

- Part 1: Robert S Laramee
  - Introduction, Challenges
  - Classification
  - Integration-Based Geometric Vector Field Visualization
  - Point-Based Seeding in 2D and 2.5D
- Part 2: Tony McLoughlin
  - Effective Particle Tracing
  - Point-Based Seeding in 3D
- Part 3: Ronald Peikert
  - Curve-Based Seeding
  - Planar-Based Seeding
- Conclusions and Future Work



What is Flow Visualization?

- A classic topic within scientific visualization
- Depiction of vector quantities (as opposed to scalar quantities)
- Applications include automotive simulation, aerodynamics, turbo machinery, meteorology, oceanography, medical visualization

Challenges:

- To effectively visualize both *magnitude* + *direction,* often simultaneously
- Large, time-dependent data sets
- Interaction, seeding, and placement,
- Computation time and irregular grids
- Perception

Computational vs. Experimental Flow Visualization

Computational Flow Visualization -using computers

- data resulting from flow *simulation*, *measurements*, or flow modelling, e.g., computational fluid dynamics (CFD)
- computer-generated images and animations, often mimicking experimental flow visualization

Visualization of actual fluids, e.g. water and air

- dye injection
- interferometry
- Schlieren/shadows
- flow topology graphs
- etc.

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Data Characterized by Many Dimensions

Spatial dimensions:

- 2D (planar flow, simplified or synthetic)
- 2.5D (boundary flow, flow on surface)
- 3D (real-world flow)

Temporal dimension:

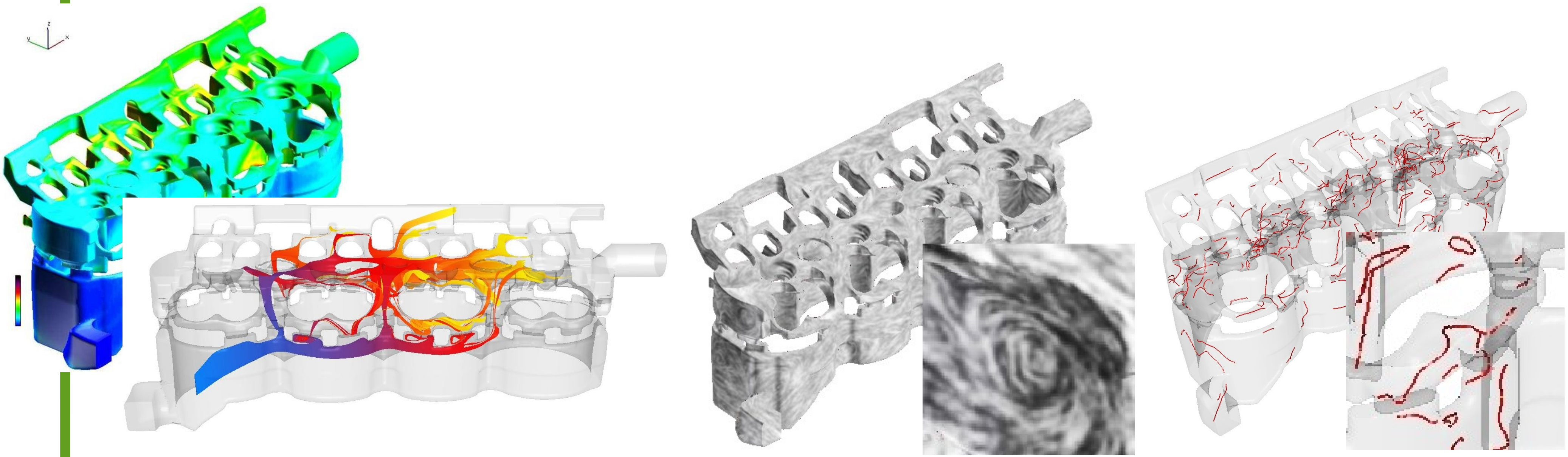
- steady flow -one time step (or instantaneous or static flow)
- time-dependent flow -multiple time steps (or unsteady or transient, real-world)
- ***caution*** is advised in the context of animation

Simulation Data Attributes a.k.a. Data Dimensions:

- velocity
- temperature
- pressure
- and many more…

Flow Visualization Classification

- **direct:** overview of vector field, minimal computation, e.g. glyphs, color mapping
- **texture-based:** covers domain with a convolved texture, e.g., Spot Noise, LIC, ISA, IBFV(S)
- **geometric:** a discrete object(s) whose geometry reflects flow characteristics, e.g. streamlines
- **feature-based:** both automatic and interactive feature-based techniques, e.g. flow topology



Geometric Flow Visualization

The computation of discrete objects whose shape is directly related to underlying geometry

Velocity is described by:

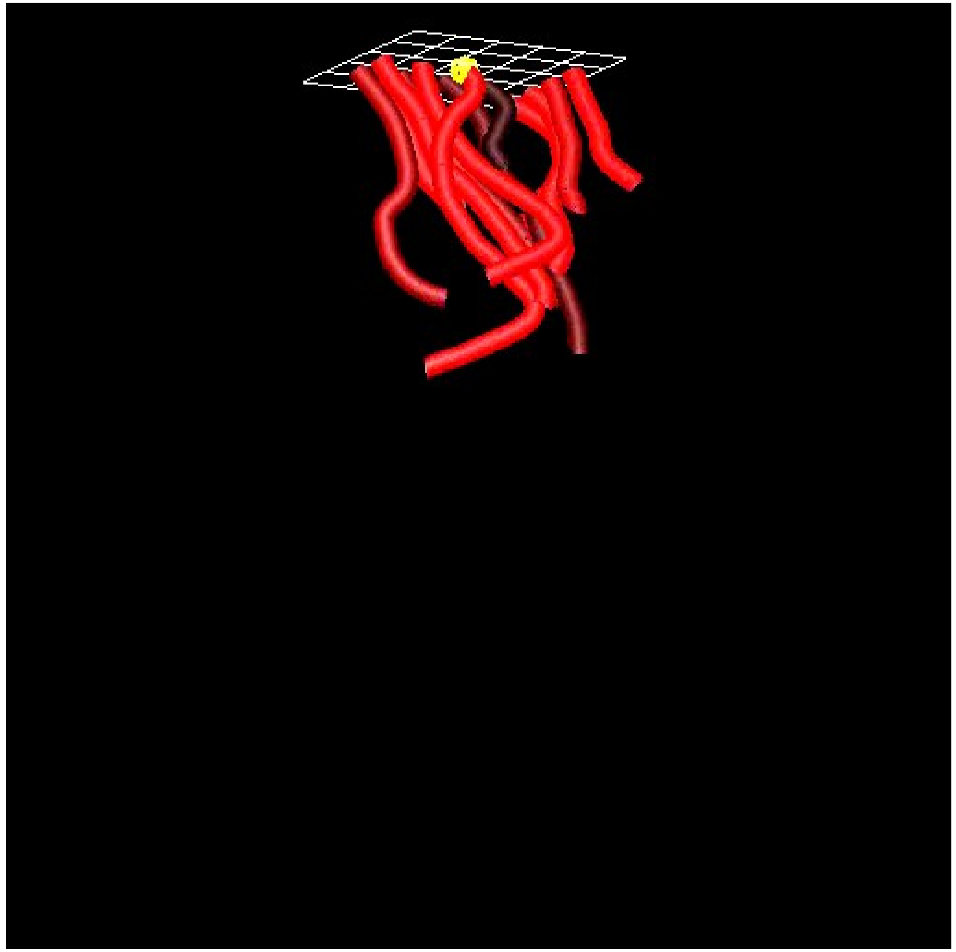
$\boldsymbol{v} = d\boldsymbol{x}/dt$

Displacement described by:

$d\mathbf{x} = \mathbf{v} \cdot dt$

Integrate in order to solve for position:

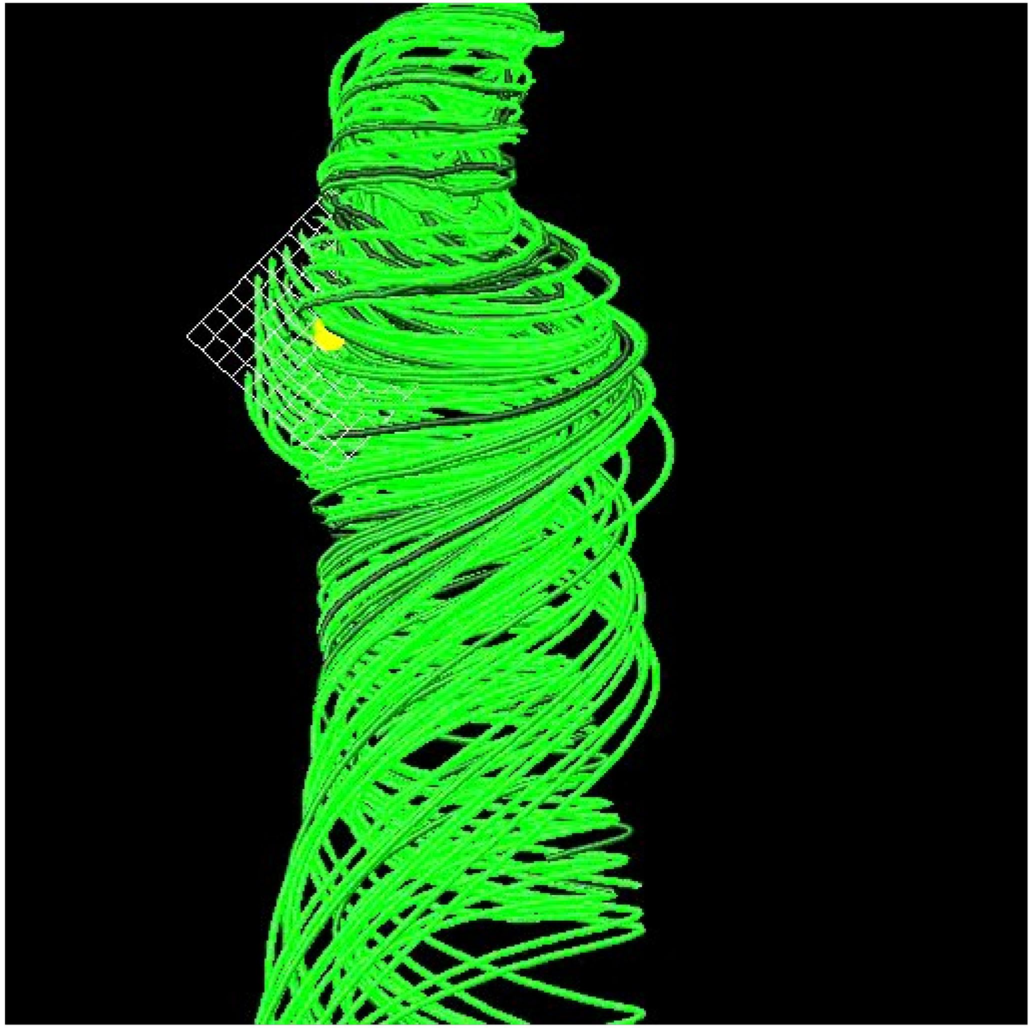
$$x(t, x_0) = \int_0^{\lambda} v(\lambda)\, d\lambda$$



Geometric Flow Visualization

Advantages:

- Intuitive,
- Clearer perception of characteristics,
- Applicable to 3D/4D



Disadvantages:

- Placement,
- Perception: visual complexity in 3D and 4D,
- irregular grids: Sometimes difficult implementation

Robert S. Laramee
r.s.laramee@swansea.ac.uk

Survey Overview

- red = seeding
- green = perceptual challenges
- yellow = performance

Seeding Object Dimensionality	2D Data Domain Steady	2D Data Domain Unsteady	2.5D Data Domain Steady	2.5D Data Domain Unsteady	3D Data Domain Steady	3D Data Domain Unsteady
0D	[TB96] [JL97a] [JL97b] [JL01] [VKP00] [LJL04] [MAD05] [LM06] [LHS08]	[JL00]	[vW92] [vW93a] [MHHI98]		[BS87]$_{pt}$ [RBM87]$_{pt}$ [Bun89]$_{pt}$ [BMP*90]$_{pt}$ [KM92]$_{pt}$ [USM96]$_{pt}$ [LPSW96]$_{pt}$ [SvWHP97]$_{pt}$ [SdBPM98]$_{pt}$ [SRBE99]$_{pt}$ [NJ99]$_{pt}$ [VP04]$_{pt}$ [HP93] [ZSH96] [FG98] [MT*03] [LWSH04] [MPSS05] [LGD*05] [LH05] [YKP05] [CCK07] [LS07]	[Lan93]$_{pt}$ [Lan94]$_{pt}$ [KL96]$_{pt}$ [TGE97]$_{pt}$ [TGE98]$_{pt}$ [TE99]$_{pt}$ [SGvR*03] [KKKW05]$_{pt}$ [BSK*07]$_{pt}$ [BL92] [WS05] [HE06] [GKT*08]
1D					[Hul92] [vW93b] [BHR*94] [LMG97] [SBH*01] [GTS*04] [LGSH06]	[STWE07]
2D					[SVL91] [MBC93] [XZC04]	[BLM95]

Geometric Flow Visualization: Some Terminology

Stream vs. Path vs Streak vs Time lines

Streamline

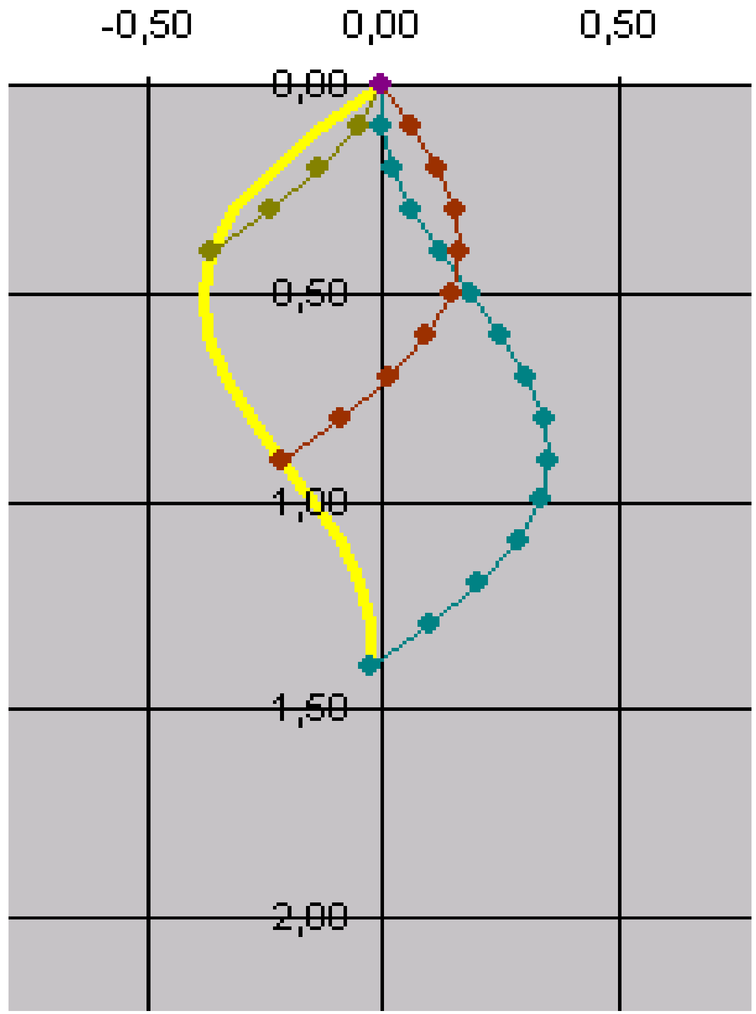
- everywhere tangent to flow at instantaneous time, **`t0`** (blue/aqua)

Pathline

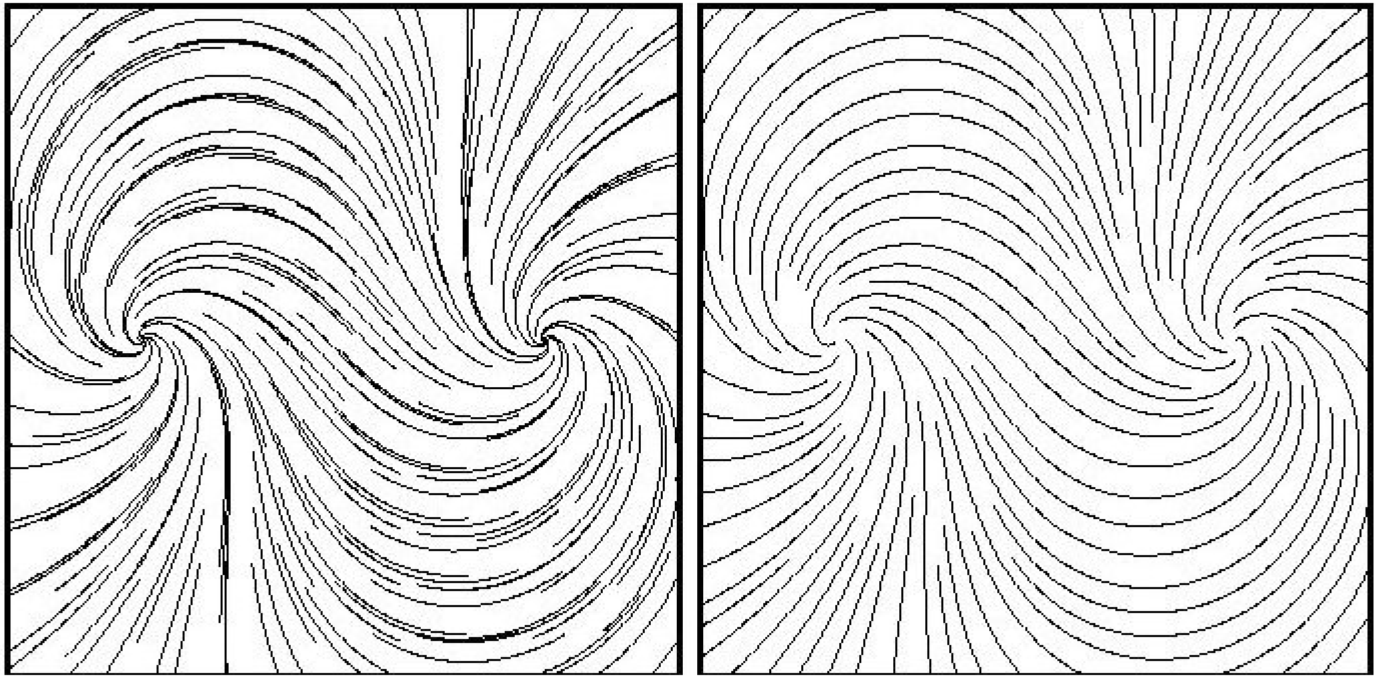
- path traced by a particle over time, **`t`** (red/maroon)

Streakline

- line traced by continuous injection at location, $\mathbf{x}_0$ (light green)



Point-Based Seeding: Problem



Regularly spaced seeds do not result in regularly spaced streamlines.

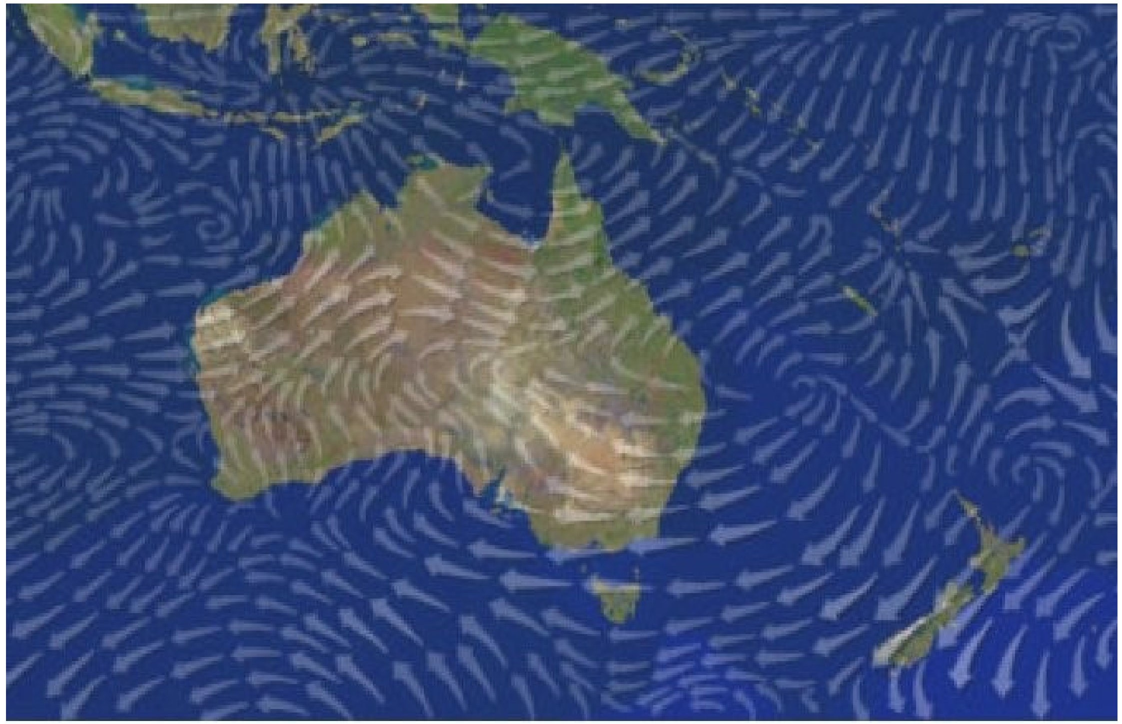
Point-Based Seeding in 2D, Steady-State Vector Fields

Image Guided Streamlines (Turk and Banks '96)

- Distribute streamlines evenly in image space

- Algorithm:

place streamlines (randomly),
DO shift streamlines,
   IF (improved position)
   THEN (accept change)
UNTIL no more improvements



Point-Based Seeding in 2D, Steady-State Vector Fields

Evenly-Spaced Streamlines
(Jobard and Lefer '97)

- Distribute streamlines evenly in image space quickly

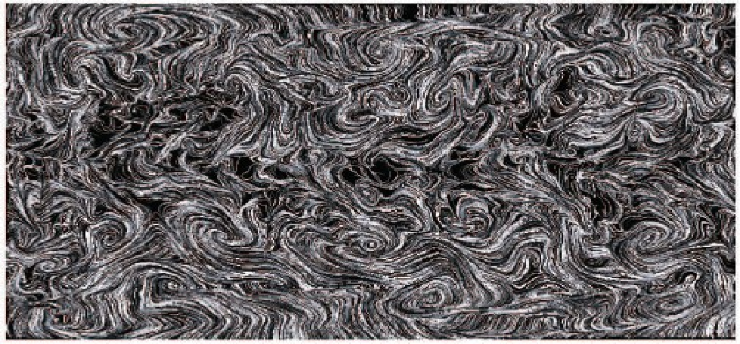
Implementation:

- Place initial streamline (randomly),
- Perform streamline-driven search of image space for new seeds.

Point-Based Seeding in 2D, Steady-State Vector Fields

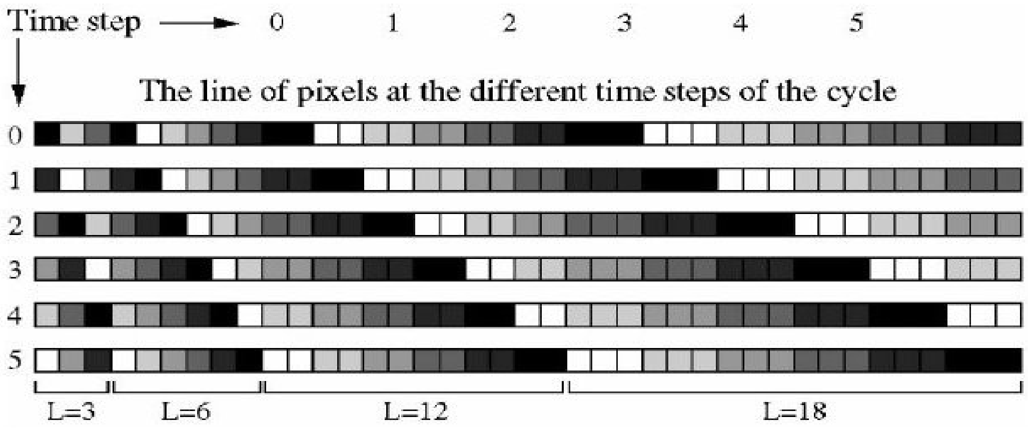
High Quality Animation of 2D, Steady Vector Fields (Lefer et al. '04)

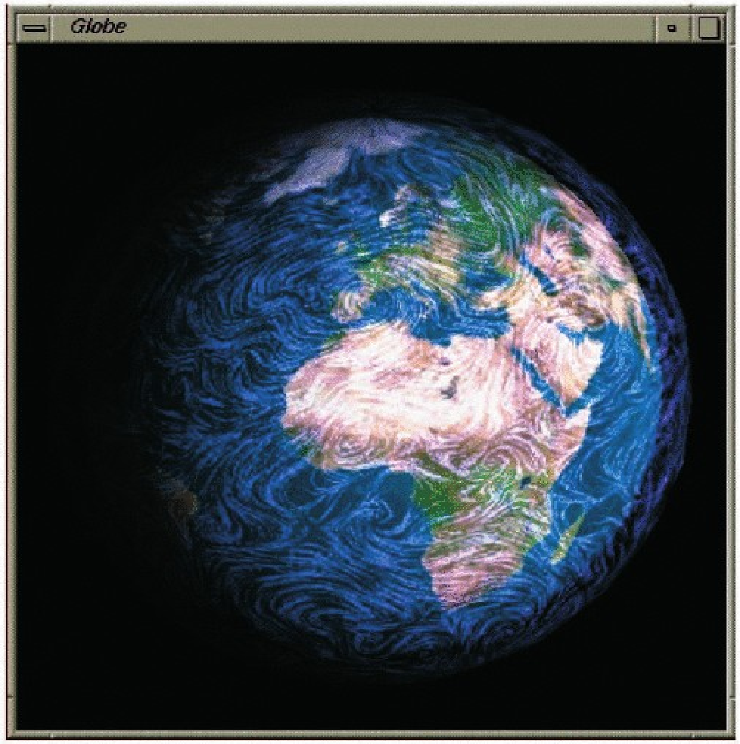
- A dense, animation of flow



Implementation:

- Texture-mapped streamlines





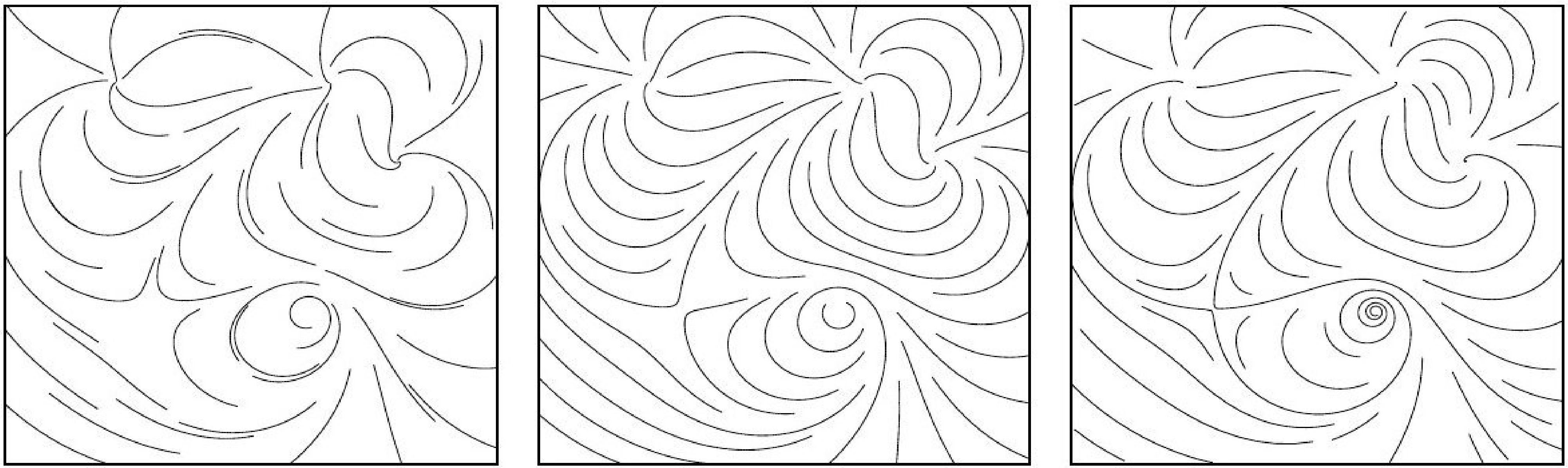
Point-Based Seeding in 2D, Steady-State Vector Fields

Flow-Guided Streamline Seeding (Verma et al. '00)

- Emphasize critical points in flow field

Implementation:

- Extract critical points
- Apply dense seeding template



Robert S. Laramee
r.s.laramee@swansea.ac.uk

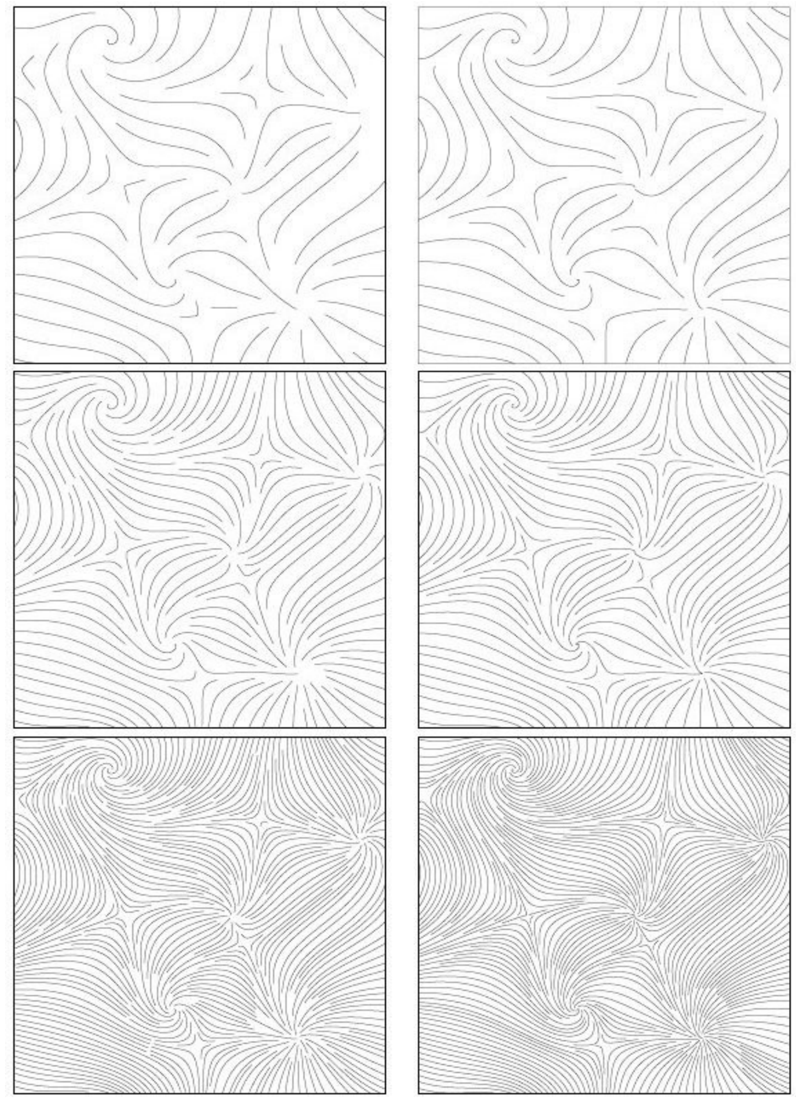
Point-Based Seeding in 2D, Steady-State Vector Fields

Farthest Point Seeding for Efficient Placement of Streamlines (Mebarki et al. '05)

- Longer, more coherent streamlines

Implementation:

- Seed in largest empty spaces



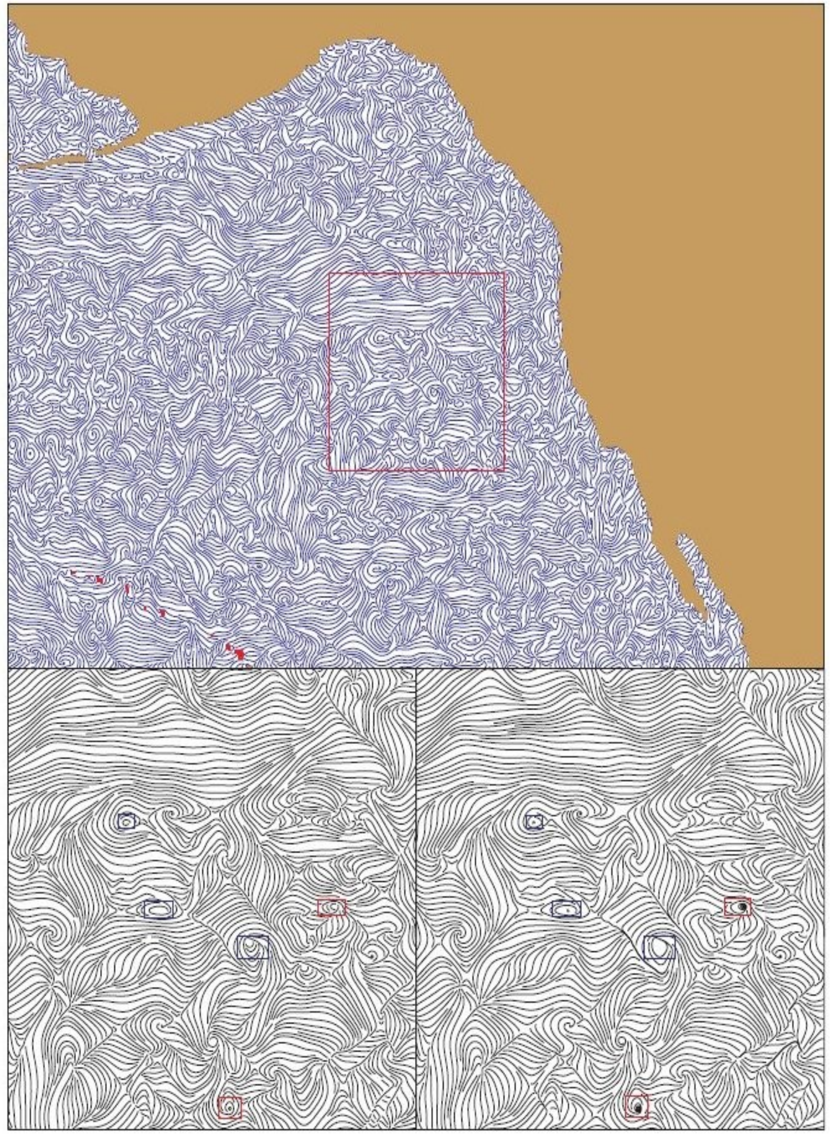
Point-Based Seeding in 2D, Steady-State Vector Fields

An Advanced Evenly-Spaced Streamline Seeding Algorithm (Liu et al. '06)

- Faster than previous algorithms and can detect streamline loops

Implementation:

- faster streamline integrator
- double-queueing strategy- prioritizes streamlines near critical points
- efficient loop detection
- (Ocean Flow from Pacific Northwest)



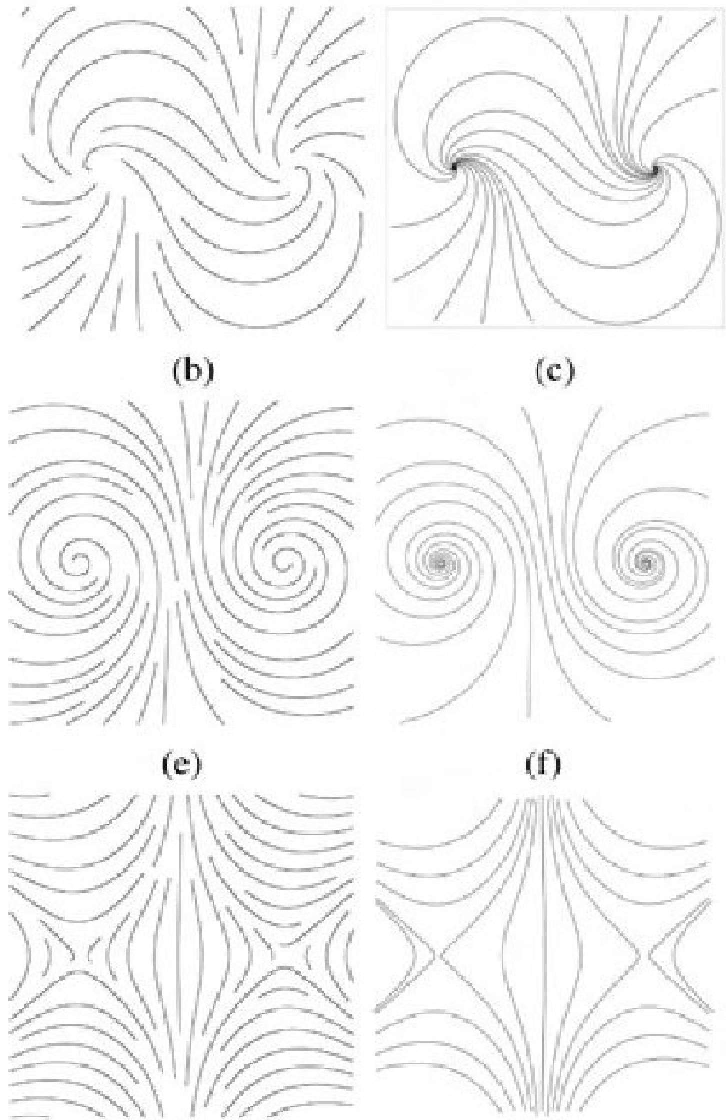
Point-Based Seeding in 2D, Steady-State Vector Fields

Illustrative Streamline Placement and Visualization (Li et al. '08)

- place minimal number of streamlines and capture features

Implementation:

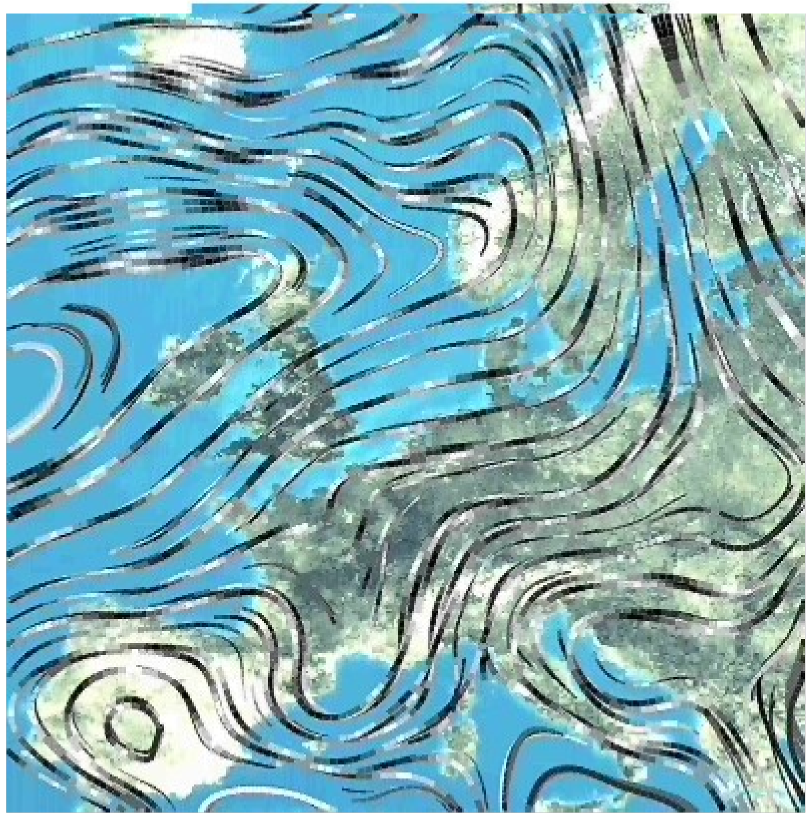
- derive a distance field
- compare sample points to existing streamline points
- trace new streamlines only when difference exceeds a threshold



Point-Based Seeding in 2D, Unsteady Vector Fields

Unsteady Flow Visualization by Animating Evenly-Spaced Streamlines (Jobard and Lefer. '00)

- Extension to unsteady flow visualization



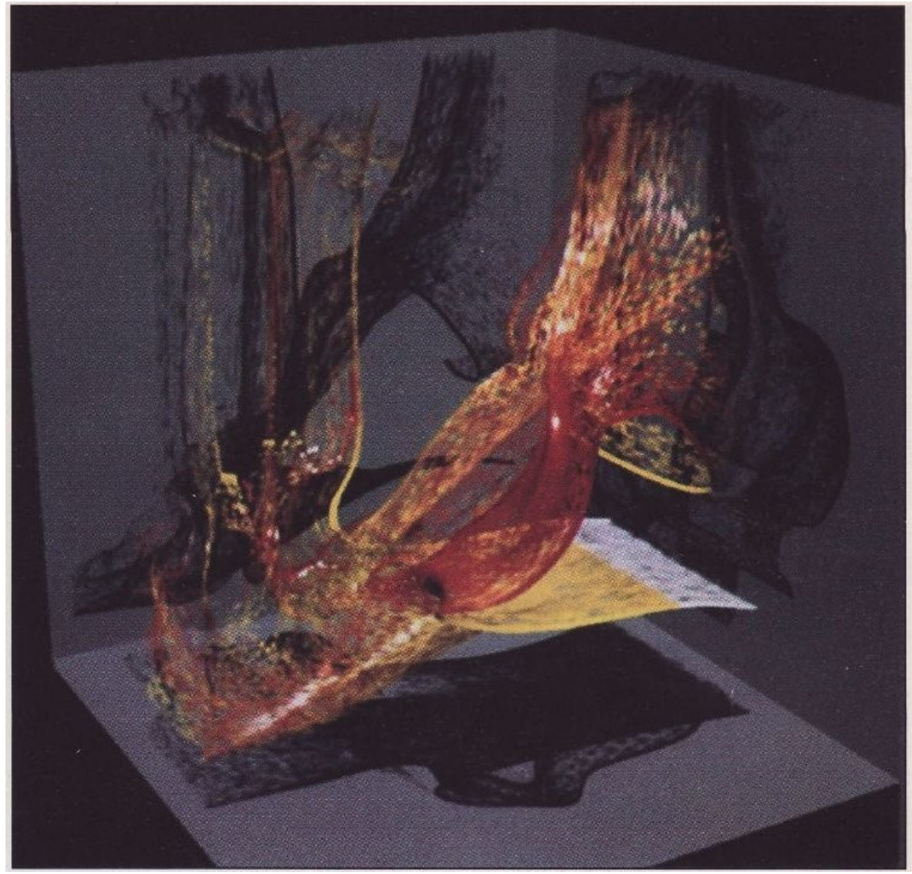
Implementation:

- evenly-spaced streamlines computed for each time step
- streamlines computed at previous time step are used a basis for current set

Point-Based Seeding on Surfaces, Steady-State Flow

Flow Visualization with Surface Particles
(Van Wijk '93)

- Efficient rendering and animation on surfaces



Implementation:

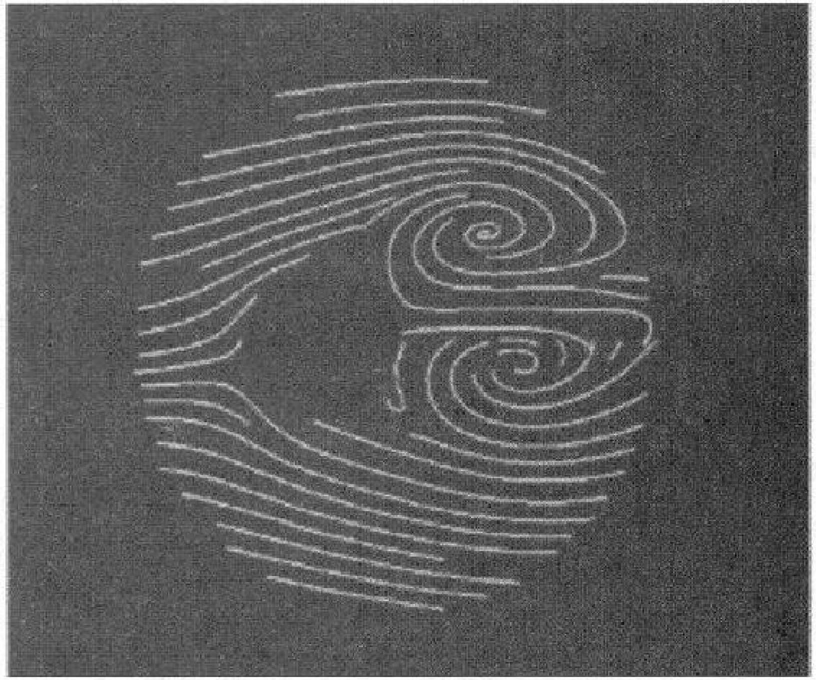
- Shading, filtering, scan conversion, occlusion including hidden surface removal
- (Thermal air flow through a TV cabin)

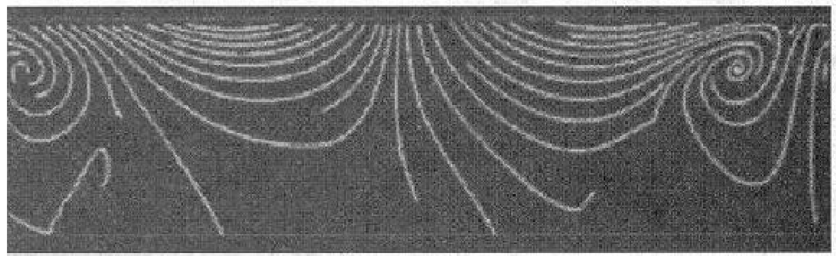
Robert S. Laramee
r.s.laramee@swansea.ac.uk

Streamline Seeding on Surfaces, Steady-State

Image-Guided Streamline Placement on Curvilinear Grid Surfaces (Mao et al. '98)

- Streamline placement for surfaces





Implementation:

- Map surface vectors to computational space of curvilinear grid
- Introduce a new energy function to handle distortion resulting from mapping

Streamline Seeding on Surfaces

Evenly-Spaced Streamlines for Surfaces: An Image-Based Approach (Spencer et al. '09)

- General streamline placement for surfaces

Implementation:

- Project vector field to image space
- Perform integration in image space

Evenly-Spaced Streamlines for Surfaces: An Image-Based Approach

Ben Spencer, Robert S. Laramee, Guoning Chen and Eugene Zhang

End of Part I

- Thank you for your attention! Any questions?

We would like to thank the following:

- G. Chen, R. Crawfis, H. Doleisch, C. Garth, B. Girod, H. Hauser, A. Helgeland, V. Interrante, B. Jobard, W. de Leeuw, H. Loeffelmann, F. H. Post, A. Telea, H. Theisel, X. Tricoche, V. Verma, J. J. van Wijk. T. Weinkauf, D. Weiskopf, R. Westermann, E. Zhang

- PDF versions of STAR and MPEG movies available at:

http://cs.swan.ac.uk/~csbob

- Next up: Tony McLoughlin and Part II