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

Tony McLoughlin¹, Robert S. Laramee¹, Ronald Peikert², Frits H. Post³, and Min Chen¹

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

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

³Computer 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.
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:
\[ \mathbf{v} = \frac{d\mathbf{x}}{dt} \]

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
### Survey Overview

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

<table>
<thead>
<tr>
<th>Seeding Object</th>
<th>2D Data Domain</th>
<th>2.5D Data Domain</th>
<th>3D Data Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensionality</td>
<td>Steady</td>
<td>Unsteady</td>
<td>Steady</td>
</tr>
<tr>
<td>0D</td>
<td>[TB96]</td>
<td>[JL01]</td>
<td>[LH96]</td>
</tr>
<tr>
<td></td>
<td>[JL97a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[HL97b]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[JL01]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[VLP90]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[LH06]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[LHS08]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1D</td>
<td>[HP92]</td>
<td>[VLP90]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[VLP90]</td>
<td>[VLP90]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[VLP90]</td>
<td>[VLP90]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[VLP90]</td>
<td>[VLP90]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[VLP90]</td>
<td>[VLP90]</td>
<td></td>
</tr>
<tr>
<td>2D</td>
<td>[Bl92]</td>
<td>[VW93b]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[BHR94]</td>
<td>[LMG97]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[SBH91]</td>
<td>[IGS+94]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[IGS+94]</td>
<td>[IGS+94]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[IGS+94]</td>
<td>[IGS+94]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[IGS+94]</td>
<td>[IGS+94]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[IGS+94]</td>
<td>[IGS+94]</td>
<td></td>
</tr>
</tbody>
</table>

[http://cs.swan.ac.uk/~csbob/]
Geometric Flow Visualization: Some Terminology

Stream vs. Path vs Streak vs Time lines

Streamline
- everywhere tangent to flow at instantaneous time, $t_0$ (blue/aqua)

Pathline
- path traced by a particle over time, $t$ (red/maroon)

Streakline
- line traced by continuous injection at location, $x_0$ (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
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 as 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)
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
End of Part I

- Thank you for your attention! Any questions?

We would like to thank the following:


- PDF versions of STAR and MPEG movies available at:
  
  http://cs.swan.ac.uk/~csbob

- Next up: Tony McLoughlin and Part II