Review of Flow Vis for Lower Dimensional Data

- **Direct:** overview of vector field, minimal computation, e.g. glyphs (arrows), 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
Vector Field Visualization in 3D
Review of Data Structure

Regular (uniform), rectilinear, and structured grids

Alternative:
tetrahedral volume elements: unstructured
Direct Method (Arrow Plot)

Source: http://docs.enthought.com/mayavi/mayavi/mlab.html
Texture-Based Method

Volume LIC

- Victoria Interrante and Chester Grosch (IEEE Visualization 97).
- A straightforward extension of LIC to 3D flow fields.
- Low-pass filters *volumetric noise* along 3D streamlines.
- Uses *volume rendering* to display resulting 3D LIC textures.
- Very time-consuming to generate 3D LIC textures.
- Texture values offer no useful guidance for transfer function design due to lack of intrinsic physical info that can be exploited to distinguish components.

⇒ Very challenging to clearly show flow directions and interior structures through a dense texture volume.
Sparse noise + Hybrid Hanning-Ramp kernel (Zhanping Liu and et al., *Journal of Image and Graphics* 2001)
Geometric-Based Methods

Streamlines:

Theory: \( \mathbf{s}(t) = \mathbf{s}_0 + \int_{0 \leq u \leq t} \mathbf{v}(\mathbf{s}(u)) \, du \)

Practice: Numerical integration such as Euler, RK2, RK4, etc.

Important: interpolation scheme, \textbf{seeding}!!

Chen et al. Vis 2007
Geometric-Based Methods
streamribbons, streamtubes, stream surfaces, flow volumes
streamribbon:
a ribbon (surface of fixed width) always tangent to the vector field
shows rotational (or twist) properties of the 3D flow
Streamribbon generation:

- Start with a 3D point $\mathbf{x}_{i=0}$ and a 2$^{\text{nd}}$ one $\mathbf{y}_{i=0}$ in a particular dist. $d$, i.e. $|\mathbf{x}_i-\mathbf{y}_i|^2 = d^2$
- Loop:
  - Integrate from $\mathbf{x}_i$ to yield $\mathbf{x}_{i+1}$
  - Do an integration step from $\mathbf{y}_i$ to yield $\mathbf{z}$
    renormalize the distance between $\mathbf{x}_i$ & $\mathbf{z}$ to $d$, i.e. $\mathbf{y}_{i+1} = \mathbf{x}_i + d \cdot (\mathbf{z}-\mathbf{x}_i)/|\mathbf{z}-\mathbf{x}_i|$
- End streamribbon integration if necessary
What about **Stream Surfaces**?

- The computation of stream surfaces is similar to streamribbon.
- However, now the seeding points are typically more than two.
- Also, during the integration, we may need to adaptively add or remove seeds (i.e. handling divergence, convergence, and shear).
- Triangulating the stream surface between neighboring streamlines is easy to achieve.
- What is the challenge?
Where to put seeds to start the integration?

Seeding along a straight-line
Allow user exploration
Source: D.Weiskopf et al. 2007

Seeding along the direction that is perpendicular to the flow leads to stream surface with large coverage
Source: Edmunds et al. EuroVis2012
How about automatic stream surface placement?

Where to start?  
How to proceed?  
Render

Source: Edmunds et al. TPCG 2012
Rendering of stream surfaces

- Stream arrows
  (Löffelmann et al. 1997)
- Texture advection on stream surfaces
  (Laramee et al. 2006)
Rendering of stream surfaces

Illustrative visualization

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

Source: Hummel et al. 2010

Source: Born et al. Vis2010

Abraham/Shaw’s illustration, 1984
Geometric FlowVis in 3D

**flow volume**: a volume whose surface is everywhere tangent to the flow

**streamtube**: shows convergence and divergence of flow (similar to streamribbon)
# Relation to Seed Objects

<table>
<thead>
<tr>
<th>Object</th>
<th>Object Dimensionality</th>
<th>Seed Object Dimensionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Streamline,...</td>
<td>1D</td>
<td>0D (point)</td>
</tr>
<tr>
<td>Streamribbon</td>
<td>2.5D</td>
<td>1D (line segment)</td>
</tr>
<tr>
<td>Streamtube</td>
<td>2.5D</td>
<td>1D (circle)</td>
</tr>
<tr>
<td>Stream surface</td>
<td>2.5D</td>
<td>1D (curve)</td>
</tr>
<tr>
<td>Flow volume</td>
<td>3D</td>
<td>2D (patch)</td>
</tr>
</tbody>
</table>
Feature-Based Methods

Topology of 3D *Steady* Flows
3D Flow Topology

• Fixed points

• Can be characterized using 3D Poincaré index

• Both line and surface separatrices exist
3D Cycles

• Similar principle as in 2D
  – Isolate closed cell chain in which streamline integration appears captured
  – Start stream surface integration along boundary of cell-wise region
  – Use flow continuity to exclude reentry cases

Source: http://www.stsci.edu/~lbradley/seminar/attractors.html
3D Cycles
3D Topology Extraction

• Cell-wise fixed point extraction:
  – Compute root of linear / trilinear expression
  – Compute Jacobian at found position
  – If type is saddle compute eigenvectors

• Extract closed streamlines

• Integrate line-type separatrices

• Integrate surface separatrices as stream surfaces
Saddle Connectors

Topological representations of the Benzene data set.
(left) The topological skeleton looks visually cluttered due to the shown separation surfaces.
(right) Visualization of the topological skeleton using connectors.
Source: Weinkauf et al. VisSym 2004
Saddle Connectors

• Multiple separating surfaces may lead to occlusion problems
• Idea: reduce visual clutter by replacing stream surfaces with streamlines of interest

• Saddle Connector:
  – Separating surfaces intersection integrated from two saddle points of opposite indices (inflow vs. outflow surface)
  – Intersection is a streamline

Source: Theisel et al. Vis 03
Saddle Connectors

Flow behind a circular cylinder:
13 fixed points and 9 saddle connectors have been detected and visualized. Additional LIC planes have been placed to show the correspondence between the skeleton and the flow.
Source: Theisel et al. 2003
Time-Dependent (Unsteady) Flow Visualization
What is Different?

Steady (time-independent) flows:
- flow itself constant over time
- \( \mathbf{v}(\mathbf{x}) \), e.g., laminar flows
- simpler case for visualization

Time-dependent (unsteady) flows:
- flow itself changes over time
- \( \mathbf{v}(\mathbf{x},t) \), e.g., turbulent flow
- more complex case
Mathematical Framework

• An unsteady **vector field**
  – is a continuous vector-valued function \( \tilde{v}(x, t) \) on a manifold \( X \)
  – can be expressed as a system of ODE \( \frac{dx}{dt} = \tilde{v}(x, t) \)
  – is a map \( \varphi : R \times X \rightarrow X \)
  – with initial condition \( x(t_0) = x_0 \), its solution is called an integral curve, trajectory, or orbit.
Stream, Path, and Streaklines

Terminology:

- **Streamline**: a curve that is everywhere tangent to the steady flow (release 1 massless particle)
  \[ \mathbf{s}(t) = \mathbf{s}_0 + \int_{0 \leq u \leq t} \mathbf{v}(\mathbf{s}(u)) \, du \]

- **Pathline**: a curve that is everywhere tangent to an unsteady flow field (release 1 massless particle)
  \[ \mathbf{s}(t) = \mathbf{s}_0 + \int_{0 \leq u \leq t} \mathbf{v}(\mathbf{s}(u), u) \, du \]

- **Streakline**: a curve traced by the continuous release of particles in unsteady flow from the **same position in space** (release infinitely many massless particles)

Each is equivalent in steady-state flow
Difference Between Streamlines and Pathlines

A moving center over time: (left) streamlines; (right) path lines
Source: Weinkauf et al. Vis2010
**Streaklines**

- Not tangent curves to the vector fields
- Union of the current positions of particles released at the same point in space

**timelines**

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

Source: wikipedia

Weinkauf et al. 2010
Characteristics of Integral Lines

Advantages:
• **Implementation**: various easy-to-implement streamline tracing algorithms (integration)
• **Intuitive**: interpretation is not difficult
• **Applicability**: generally applicable to all vector fields, also in three-dimensions

Disadvantages:
• **Perception**: too many lines can lead to clutter and visual complexity
• **Perception**: depth is difficult to perceive, no well-defined normal vector
• **Seeding**: optimal placement is very challenging (unsolved problem)

Source: Marchesin et al. Vis 2010
**Path surface:**
The extension of the path lines into unsteady 3 dimensional flows

Illustrative path surfaces: texture + transparency [Hemmel et al. 2010]
Path surface:
Its computation can use time line advection

Source: Garth Vis09 Tutorial
Streak Surfaces: Challenges

**Streak surfaces** are an extension of streak lines (next higher dimension)

**Challenges:**

- Computational cost: surface advection is very expensive
- Surface completely dynamic: entire surface (all vertices) advect at each time-step
- Mesh quality and maintaining an adequate sampling of the field.
  - Divergence
  - Convergence
  - Shear
- Large size of time-dependent (unsteady) vector field data, out-of-core techniques
A Streak Surface Computation Pipeline

Divergence: Quad Splitting

Shear

Seed Rake
- Integrate Vertices
  - Remove Terminated Vertices and Quads
    - Divergence → Convergence → Shear
      - Update Sampling Rate and Mesh
        - Boundary Conditions
          - End of Simulation?
            - Render
              - No
                - Iterate
              - Yes
                - Stop Integration
                  - Render
Time Lines on Streak Surfaces

Vortex formation behind an ellipsoidal body

Source: Garth et al. Vis 2008
Texture-Based Methods

Unsteady flow LIC (UFLIC): forward scattering + collecting

IBFV: texture advection in forward direction + hardware acceleration
Topology-based Method

• Parameter dependent topology:
  • “Fixed points” (no more fixed) move, appear, vanish, transform
  • Topological graph connectivity changes

• Structural stability: topology is stable w.r.t. small but arbitrary changes of parameter(s) if and only if
  • 1) Number of fixed points and closed orbits is finite and all are hyperbolic
  • 2) No saddle-saddle connection
Bifurcations

• Transition from one stable structure to another through *unstable state*
• Bifurcation value: parameter value inducing the transition
• Local vs. global bifurcations
Local Bifurcations

- Transformation affects local region
- **Fold bifurcation**: saddle + sink/source

1D equivalent:

- Source
- Sink
- Unstable
- No fixed point
Local Bifurcations

- Transformation affects local region
- **Hopf bifurcation**: sink/source + closed orbit
Global Bifurcations

- Affects overall topological connectivity
- Basin bifurcation

Saddle-saddle connection
Global Bifurcations

- Modifies overall topological connectivity
- Homoclinic bifurcation

Saddle-saddle connection

repelling cycle (source)
Global Bifurcations

- Modifies overall topological connectivity
- Periodic blue sky

![Diagram showing bifurcations with sink, source, and homoclinic connection marked.](image-url)
2+1D Topology

- Time-wise interpolation
- Cell-wise tracking over 2+1D grid
- Detect local bifurcations
- Track associated separatrices (surfaces)
2+1D Topology
2+1D Topology
• Feature Flow Field
  – Track path of critical points by streamline integration in vector field defined over (n+1)D space-time domain

\[
\vec{f}(x, y, t) = \begin{pmatrix} u(x, y, t) \\ v(x, y, t) \end{pmatrix} \quad \vec{g}(x, y, t) = \nabla u \times \nabla v
\]

– The value of \( \vec{f} \) (e.g. \( \vec{0} \)) is constant along streamlines of \( \vec{g} \)
2+1D Topology
2+1D Topology

• Combined with stream surface integration in space-time domain: moving separatrices
Additional Readings


• Tino Weinkauf and Holger Theisel. Streak Lines as Tangent Curves of a Derived Vector Field. IEEE Visualization 2010.
Acknowledgment

Thanks for the materials

- Prof. Robert S. Laramee, Swansea University, UK
- Dr. Christoph Garth, University of Kaiserslautern, Germany
Final Project Topic Overview
Advanced texture-based flow visualization

- IBFVS – Image-based Flow Visualization on Curved Surfaces

1) Initialize image
2) Calculate the texture coordinates for all vertices
3) Texture mapping using previous saved image
4) Blend with fresh white noise
5) Save current image for next frame texture mapping
6) Blend with shaded surface
7) Other post image process
8) If current state is changed, say camera position is changed, the projected vector field is changed subsequently, go back to 2), otherwise, go back to 3)
Advanced texture-based flow visualization

- ISA – Image-Space Texture Advection

1) Project geometric mesh and its associated vector field onto the image space to get a velocity image
2) Perform edge detection to avoid unnatural continuities
3) Compute the advection of the mesh in image space using the similar method as IBFV
4) Map the previous image to the distorted mesh
5) Blend fresh noise to keep contrast
6) Blend extra gray value to highlight the discontinuities at the edges detected in 2)
7) Overlap with the attributions of the surfaces
Advanced texture-based flow visualization

Volume LIC
Vector field topology

- ECG versus MCG
Vector field topology

• 3D topology

Saddle connectors:
Vector field topology

Out-of-core vector field analysis
new research, no much work on this topic yet
Time-varying vector field analysis

• Stable feature flow for feature tracking

Time-varying vector field analysis

- FTLE analysis
  - Lagrangian coherent structure (LCS)

Ridges in Lyapunov exponent
Transient view

Confluences

LCS = interface
Time-varying vector field analysis

- FTLE analysis
  - FTLE is the “growth of perturbation after advection time $T$".

For flow:

$$\varphi_{t_0}^t (x) : x_{t_0} \mapsto x_t \quad x_{t_0} \rightarrow x_t$$

$$FTLE(x, t_0, T) = \frac{1}{|T|} \ln \left( \frac{\Delta}{\delta} \right)$$

$$\|A\|_2 = \sqrt{\lambda_{\text{max}}(A^T A)}$$
3D (time-varying) vector field visualization

- Geometric-based method
Scalar field analysis

- Reeb graph computation

Scalar field analysis

• Morse-Smale complex computation

Asymmetric tensor field visualization
Asymmetric tensor field visualization

Diffusion tensor imaging


GPU-based real-time volume rendering

- GPU-based ray-casting + iso-surface rendering

Open source for your reference: Exposure render
Illustrative visualization

Illustrative visualization of scalar fields

Illustrative visualization of vector fields
Information visualization

• Large scale dynamic graph visualization


Open source: https://gephi.org/
http://www.caida.org/tools/visualization/walrus/
Information visualization

• Higher dimensional data visualization

What is a person’s life span given gender, annual income, living area, marriage status, medical care, smoking or not, ...

What will determine/affect the temperature? Location (x, y, z), time in a day, humidity, pressure, cloudy or not, windy or not, human activities, ...
Information visualization

• Higher dimensional data visualization

Climatic predictors

<table>
<thead>
<tr>
<th>WetDays</th>
<th>TempJuly</th>
<th>TempJan</th>
<th>TempAnn</th>
<th>RHJuly</th>
</tr>
</thead>
</table>

Parallel coordinates

What else?
name a good project with some merit of visualization then we can discuss.
3B: Be creative! Be aggressive! Be ambitious!