Vector Visualization
Vector Visualization

- Divergence and Vorticity
- Vector Glyphs
- Vector Color Coding
- Displacement Plots
- Stream Objects
- Texture-Based Vector Visualization
- Simplified Representation of Vector Fields
Vector Function

\[ f: \mathbb{R}^3 \rightarrow \mathbb{R}^3 \]

(usually in 3-D)

\[ f: \mathbb{R}^2 \rightarrow \mathbb{R}^2 \]

(simple case: 2-D)
Vector: $\vec{V}$

$\vec{V} = V_x \hat{i} + V_y \hat{j} + V_z \hat{k}$

or

$\vec{V} = (V_x, V_y, V_z)$

or

$\vec{V} = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} = \begin{pmatrix} f_x(x,y,z) \\ f_y(x,y,z) \\ f_z(x,y,z) \end{pmatrix}$

Scalar: $S$

$s = f(x,y,z)$
Example in 2-D

Vector: $\vec{V}$

$\vec{V} = \begin{pmatrix} V_x \\ V_y \end{pmatrix}$

$\vec{V} = \begin{pmatrix} f_x(x, y) \\ f_y(x, y) \end{pmatrix} = \begin{pmatrix} y \\ x \end{pmatrix}$

Scalars:

$s = f(x, y)$

$s = e^{-(x^2+y^2)}$

exp:

$s = x + y$
• **Divergence and Vorticity**
• **Vector Glyphs**
• **Vector Color Coding**
• **Displacement Plots**
• **Stream Objects**
• **Texture-Based Vector Visualization**
• **Simplified Representation of Vector Fields**
Gradient of a Scalar

\[ V = \nabla s = \left( \frac{\partial s}{\partial x}, \frac{\partial s}{\partial y}, \frac{\partial s}{\partial z} \right) \]

Exp: 2D

\[ s = e^{-(x^2 + y^2)} \]

\[ V_x = \frac{\partial s}{\partial x} = -2xe^{-(x^2 + y^2)} \]

\[ V_y = \frac{\partial s}{\partial y} = -2ye^{-(x^2 + y^2)} \]
Divergence of a Vector

\[ \nabla \cdot \vec{V} = \lim_{\Gamma \to 0} \frac{1}{|\Gamma|} \int_{\Gamma} (\vec{V} \cdot \vec{n}_\Gamma) ds \]

\( \Gamma \) is closed hypersurface (closed curve in 2D and closed surface in 3D)

\(|\Gamma|\) is the area enclosed by \( \Gamma \) (area in 2D and volume in 3D)

Divergence in 2D. (a) Divergence construction. (b) Source point. (c) Sink point.
Divergence of a Vector

- Divergence computes the flux that the vector field transports through the imaginary boundary $\Gamma$, as $\Gamma \to 0$
- Divergence of a vector is a scalar
- A positive divergence point is called source, because it indicates that mass would spread from the point (in fluid flow)
- A negative divergence point is called sink, because it indicates that mass would get sucked into the point (in fluid flow)
- A zero divergence denotes that mass is transported without compression or expansion.
Divergence of a Vector

\[ \nabla \cdot \vec{V} = \frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial x} + \frac{\partial V_z}{\partial x} \]

Example:
\[ \vec{V} = (x, y) \]
\[ \nabla \cdot \vec{V} = 1 + 1 = 2 \]
Positive divergence source

\[ \vec{V} = (-x, -y) \]
\[ \nabla \cdot \vec{V} = -1 - 1 = -2 \]
Negative divergence sink

\[ \vec{V} = (y, x) \]
\[ \nabla \cdot \vec{V} = 0 + 0 = 0 \]
Divergence Free
Divergence of a Vector
Vorticity of a Vector

\[ \nabla \times \vec{V} = \lim_{\Gamma \to 0} \frac{1}{|\Gamma|} \int_{\Gamma} (\vec{V} \cdot d\vec{s}) \]

\( \Gamma \) is closed hypersurface (closed curve in 2D and closed surface in 3D)

\(|\Gamma|\) is the area enclosed by \( \Gamma \) (area in 2D and volume in 3D)

Vorticity in 2D. (d) Rotor construction. (e) High-vorticity field.
Vorticity of a Vector

- Vorticity computes the rotation flux around a point.
- Vorticity of a vector is a vector.
- The magnitude of vorticity expresses the speed of angular rotation.
- The direction of vorticity indicates direction perpendicular to the plane of rotation.
- Vorticity signals the presence of vortices in vector field.
\[ \nabla \times \vec{V} = \begin{pmatrix} \frac{\partial V_z}{\partial y} - \frac{\partial V_y}{\partial z} \\ \frac{\partial V_x}{\partial z} - \frac{\partial V_z}{\partial x} \\ \frac{\partial V_y}{\partial x} - \frac{\partial V_x}{\partial y} \end{pmatrix} \]

**Exp: in 2-D**

\[ V_z = 0 \]
\[ \frac{\partial}{\partial z} = 0 \]

For \( \vec{V} = (x, y) \)

\[ (\nabla \times \vec{V})_x = 0 \]
\[ (\nabla \times \vec{V})_y = 0 \]
\[ (\nabla \times \vec{V})_z = 0 - 0 = 0 \]
Vorticity of a Vector
• Divergence and Vorticity
• Vector Glyphs
• Vector Color Coding
• Displacement Plots
• Stream Objects
• Texture-Based Vector Visualization
• Simplified Representation of Vector Fields
Vector Glyph

\[ l = (\vec{x}, \vec{x} + k\vec{V}(\vec{x})) \]

- Vector glyph mapping technique associates a vector glyph (or icon) with the sampling points of the vector dataset.
- The magnitude and direction of the vector attribute is indicated by the various properties of the glyph: location, direction, orientation, size and color.
- Many variations of framework:
  - Lines (convey direction)
  - 3D cone (convey direction + orientation)
  - Arrow (convey direction + orientation)
Vector Glyph

Line glyph, or hedgehog glyph

Sub-sampled by a factor of 8 (32 X 32)

Original (256 X 256)

Velocity Field of a 2D Magnetohydrodynamic Simulation
Vector Glyph

Line glyph, or hedgehog glyph

Sub-sampled by a factor of 4 (64 X 64)

Original (256 X 256)
Sub-sampled by a factor of $2$ ($128 \times 128$)

Original ($256 \times 256$)

Problem with a dense representation using glyph:
(1) clutter
(2) miss-representation
Vector Glyph

Random Sub-sampling Is better
Vector Glyph: 3D

Simulation box: 128 X 85 X 42; 456,960 data point, 100,000 glyphs
Problem: visual occlusion
Vector Glyph: 3D

Simulation box: 128 X 85 X 42; 456,960 data point, **10,000 glyphs**

Less occlusion
Vector Glyph: 3D

Simulation box: 128 X 85 X 42; 456,960 data point, 100,000 glyphs, 0.15 transparency
Less occlusion
Vector Glyph: 3D

Simulation box: 128 X 85 X 42; 456,960 data point
3D velocity isosurface
Vector Glyph

- Glyph method is simple to implement, and intuitive to interpretation.

- High-resolution vector datasets must be sub-sampled in order to avoid overlapping of neighboring glyphs.

- Glyph method is a sparse visualization: does not represent all points.

- Occlusion

- Subsampling artifacts: difficult to interpolate.

- Alternative: color mapping method is a dense visualization.
• Divergence and Vorticity
• Vector Glyphs
• **Vector Color Coding**
• Displacement Plots
• Stream Objects
• Texture-Based Vector Visualization
• Simplified Representation of Vector Fields
Vector Color Coding

- Similar to scalar color mapping, vector color coding is to associate a color with every point in the data domain.

- Typically, use HSV system (color wheel):
  - Hue is used to encode the direction of the vector, e.g., angle arrangement in the color wheel.
  - Value of the color vector is used to encode the magnitude of the vector.
  - Saturation is set to one.
2-D Velocity Field of the MHD simulation:

Orientation, Magnitude
Vector Color Coding

2-D Velocity Field of the MHD simulation:

Orientation only; no magnitude
Vector Color Coding

- Dense visualization

- Lacks of intuitive interpretation; take time to be trained to interpret the image
• Divergence and Vorticity
• Vector Glyphs
• Vector Color Coding
• Displacement Plots
• Stream Objects
• Texture-Based Vector Visualization
• Simplified Representation of Vector Fields
Displacement Plots

• Vector glyphs
  – can be understood in terms of displaying trajectories
  – Each glyph shows both the start and end points of the trajectory, \( p \) and \( p + v(p) \Delta t \)

• Displacement plots (comparison)
  – Different approach by showing only the end points of such trajectories
  – Given a surface \( S \), where \( S \) is discretized as a set of sample points \( P_i \)
  – A displacement plot of \( S \) is a new surface \( S^1 \) given by:

\[
p'_i = p_i + kv'(p_i)
\]
Displacement Plots

• Displacement plot
  – Think of it as being the effect of displacing or warping, a given surface in the vector field
  – Advantage: produce a visually continuous result
  – Disadvantage: more abstract, less intuitive
    (particularly poor when displacement is along surface)

• Several elements control the quality
  – Displacement factor $K$
  – The shape and position of the surface to be warped
    • Plane, & other geometric objects.
Displacement plots of planar surfaces in a 3D vector field. 
(*color* shows the dis. length along the surface normal direction)
Displacement plots using a box & a spherical surface

(color shows velocity magnitude)
• Divergence and Vorticity
• Vector Glyphs
• Vector Color Coding
• Displacement Plots
• Stream Objects
• Texture-Based Vector Visualization
• Simplified Representation of Vector Fields
Stream Objects

- Vector glyphs – trajectories over a short $\Delta T$

- Displacement plots -- trajectories at the end points

- Stream objects -- using trajectories computed for longer time intervals
Streamlines

- Streamline is a curved path over a given time interval of a trace particle passing through a given start location or seed point.

\[
S = \{ \vec{\rho}(\tau), \tau \in [0, T] \}
\]

\[
\vec{\rho}(\tau) = \int_{t=0}^{\tau} \vec{V}(\vec{\rho})dt
\]

where

\[
\vec{\rho}(0) = \vec{\rho}_0 , \text{ these e f point}
\]
Streamlines

All lines are traced up to the same maximum time $T$
Seed points (gray ball) are uniformly sampled
Color is used to reinforce the vector magnitude
Streamlines: Issues

- Require numerical integration, which accumulates errors as the integration time increases

Euler Integration

\[ \vec{p}(\tau) = \int_{t=0}^{\tau} \vec{V}(\vec{p}) \, dt = \sum_{i=0}^{\tau/\Delta t} \vec{V}(\vec{p}_i) \Delta t \]

where

\[ \vec{p}_i = \vec{p}_{i-1} + \vec{V}_{i-1} \Delta t \]

- Euler integration: fast but less accurate
- Runge-Kutta integration: slower but more accurate
Streamlines: Issues

• Need to find optimal value of time step $\Delta t$
• Choose number and location of seed points
• Trace to maximum time or maximum length
• Trace upstream or downstream
• Saved as a polyline on an unstructured grid
Stream Tubes

• Stream tubes
  – Can be constructed by sweeping a circular cross-section along the streamline curve
  – The thickness or radius, parameter of stream tubes can also be used to convey some extra information
    • The degree of freedom has some limitations
Stream tubes

- Add a circular cross section along the streamline curves, making the lines thicker

Tracing downstream: the seed points are on a regular grid
Stream tubes

Tracing upstream: the arrow heads are on a regular grid
Stream tubes with radius and luminance modulated by normalized tube length
Streamlines and Tubes in 3D Datasets

• Choosing an appropriate sampling strategy is more critical when tracing streamline in 3D datasets compared to 2D
  – solves the coverage
  – density
  – continuity issues

• Stream tubes have advantage
  – Providing some *shading* and *occlusions* cues
  – Allow us to better determine their *actual relative position* and orientation in 3D vector visualization
Stream Objects in 3-D

Input: 128 X 85 X 42

Undersampling: 10 X 10 X 10

Opacity 1

Maximum Length
Stream Objects in 3-D

Input: 128 X 85 X 42

Undersampling: 3 X 3 X 3

Opacity 1

Maximum Length
Stream Objects in 3-D

Input: 128 X 85 X 42

Undersampling: 3 X 3 X 3

Opacity 0.3

Maximum Time
Stream Objects in 3-D

Stream tubes
Seed area at the flow inlet
Stream Ribbons

- Stream ribbons
  - Created by launching two streamlines from two seed points close to each other
  - Surface created by the lines of minimal length with endpoints on the two streamline is called a stream ribbon
  - If the two streamlines stay relatively close to each other, then the stream ribbon’s twisting around its center curve gives a measure of the twisting of the vector field around the direction of advection
Stream Ribbons

Two thick Ribbons

Vorticity is color coded

Vector Glyph
Stream Ribbons

Stream ribbons representing streamlines and twisted according to the local streamline-aligned vorticity
Stream surfaces

- Stream ribbons can be used to visualize how the curve would be advected in the vector field
- Stream ribbons can be generalized to compute so-called stream surfaces of the vector field
  - The intuitive property: the flow described by the vector field is always tangent to the surface
  - Easier to follow visually
  - Can be constructed in several ways
• Divergence and Vorticity
• Vector Glyphs
• Vector Color Coding
• Displacement Plots
• Stream Objects
• Texture-Based Vector Visualization
• Simplified Representation of Vector Fields
Texture-Based Vector Visualization

• Discrete
  -- Vector glyphs; Streamlines; Stream surfaces

• Continuous: Texture-Based Vector Visualization
  – Create a texture signal that encodes the direction and magnitude of a vector field by
    • Luminance, graininess, color and pattern structure
  – Challenge: how to encode the vector direction in the texture parameters
    • graininess
Texture-Based Vector Visualization

Vector magnitude: Color
Vector direction: Graininess
Texture-Based Vector Visualization

• **LIC**: Line Integrated Convolution

• LIC is a process of blurring or filtering the texture (noise) image along the streamlines.

• Due to blurring, the pixels along a streamline are getting smoothed; the graininess of texture is gone.

• However, between neighboring streamlines, the graininess of texture is preserved, showing contrast.
LIC employs a low-pass filter to convolve an input noise texture along pixel-centered symmetrically bi-directional streamlines to exploit spatial correlation in the flow direction. LIC provides a global dense representation of the flow, analogous to the resulting pattern of a tract of wind-blown sand.
Basic Idea of 2D LIC

\[ \frac{d}{d \tau} \rho(\tau) = \mathbf{u}(\rho(\tau)) \]

\[ \rho(\tau + \Delta \tau) = \rho(\tau) + \int_{\tau}^{\tau + \Delta \tau} \mathbf{u}(\rho(\tau)) d\tau \]

\[ \sum (\text{texture}[i] \times \text{weight}[i]) \]

\[ \sum \text{weight}[i] \]

A point in the flow field — the counterpart of a pixel in the output LIC image.

Locate a set of pixels hit by the streamline index the input noise for the texture values.

Obtain the value of the target pixel in the LIC image via texture convolution.

Weighting is governed by a low-pass filter.
Texture-Based Vector Visualization

Line Integral Convolution visualization
• Divergence and Vorticity
• Vector Glyphs
• Vector Color Coding
• Displacement Plots
• Stream Objects
• Texture-Based Vector Visualization
• Simplified Representation of Vector Fields
Simplified Representation of Vector Fields

• Do not visualize all the data points in the same way
  – Regions that exhibit important characteristic for an application area, should be visualized in different ways compared to the less important ones
    • Vortices, Speed extrema, Separation lines between regions of lamina flow
  • Simplified version
    – The sheer size of the data
Feature Detection Methods

• Feature detection methods reduce the vector field to a set of features of interest
  – Feature type, position, extent, and strength

• Feature defined
  – Analytically by a feature energy-like function
  – A set of examples or patterns

• Some problems
  – Hard to define precise numerical criteria to detect such features
  – Appear at different spatial scales in vector fields
  – No clear spatial separation between a feature and a non-feature area
Field Decomposition Methods

• Partition the vector dataset into
  – regions of strong intra-region (internal)
  – weak inter-region similarity

• Core: similarity metric $f$ that defines how similar two regions are
  – Different metric -> different decomposition
    • A frequently used: compares the direction and magnitude of the vector data

• Usually perform a top-down partitioning or bottom-up agglomerative clustering of dataset
Simplified vector field visualization via bottom-up clustering of a 2D field and a 3D field.
Field Decomposition Methods

AMG: the Algebraic MultiGrid — The idea of producing a hierarchy of bases that approximates a given vector field at several levels of detail

Climate dataset decomposition, five coarsest levels
Domains and flow texture overlaid with curved arrow icons
Conclusion

• A number of visualization methods for vector fields
  – From simple visual representations, straightforward implementation: vector glyph
  – To multiscale textures animated in real time, complex implementations: LIC: Line Integrated Convolution

• Another classification method: based on the dimensionality of the data domain
  – 2D surface: planar or curved ones
  – 3D volumetric vector fields, more challenging
    • Inherent occlusion of the visualization primitives

• The challenge of creating insightful visualization of 3D time-dependent vector fields describing complex phenomena is still an active area of research