Data Visualisation
CSE2002

Vector Visualization Techniques

- Vector data is a three-dimensional representation of direction and magnitude.
- a vector is a tuple of n scalar components $v=(v_1,v_2, v_3,..v_n)$, $v_i \in \mathbb{R}$
- Vector data often results from the study of fluid flow, or when examining derivatives, $\rightarrow$ rate of change, of some quantity, a position, force in $\mathbb{R}^n$.
- visualization techniques for vector data sets
  - Vector glyphs or Hedgehogs and oriented glyphs
  - Vector color coding
  - Warping
  - Displacement plots
  - Time animation
  - Streamlines or Streaming objects
  - texture-based vector visualization

- **Main applications of Vector field Visualization**
  - Motion of Fluids (gas, liquid)
  - Geometric boundary conditions
  - Velocity (flow) field v(x,t)
  - Pressure p
  - Temperature T
  - Vorticity
  - Density
  - Conservation of mass, energy, and momentum
  - Navier-Stokes equations
  - CFD (Computational Fluid Dynamics)

Vector Visualization Techniques

- One important application domain of vector visualization is **computational fluid dynamics**(CFD).
- The solution of a CFD simulation consists of several datasets, each for a different time step
- For each time step, several attributes are computed and stored into the solution dataset $\rightarrow$ velocity, pressure, density, flow divergence, and vorticity.
- Divergence and vorticity are important quantities for vector field visualization.

- **Divergence:**
  - Given a vector field v:$\mathbb{R}^3 \rightarrow \mathbb{R}^3$, the divergence of v $= (v_x, v_y, v_z)$ is a scalar quantity div v $= \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}$
  - if v is a flow field that transports mass, div v characterize the increase or loss of mass at a given point - p in the vector field in unit time.
  - A positive divergence at p $\rightarrow$ mass would spread from p outward. $\rightarrow$ positive divergence points are called **sources**
  - A negative divergence at p $\rightarrow$ mass get sucked into p. $\rightarrow$ negative divergence points are called **sinks**
  - A zero divergence at p $\rightarrow$ mass is transported without getting spread or sucked $\rightarrow$ without compression or expansion

- **Divergence**

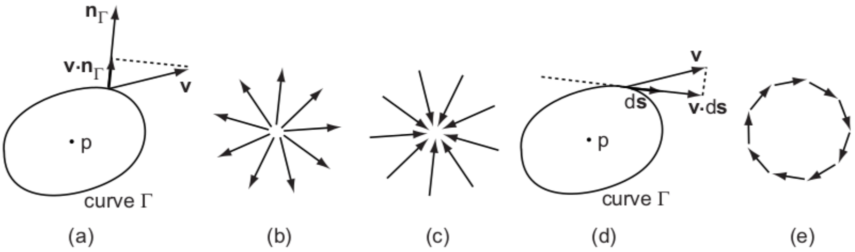
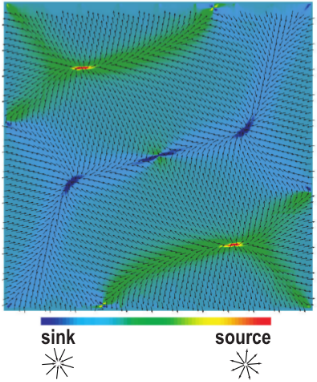


Figure 6.1. Divergence and curl in 2D. (a) Divergence construction. (b) Source point. (c) Sink point. (d) Rotor construction. (e) High-vorticity field.

- **Divergence**

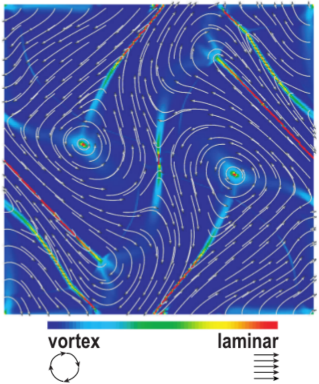
- Divergence of 2D flow field using blue-to-red colormap
- The vector fields are visualized with arrow glyphs
- Red Area $\rightarrow$ High Positive Divergence or Sources
- Blue Area $\rightarrow$ High Negative Divergence or Sinks
- if we correlate the divergence and vector glyphs, we get the image of a flow field that emerges from the sources and end up in the sinks



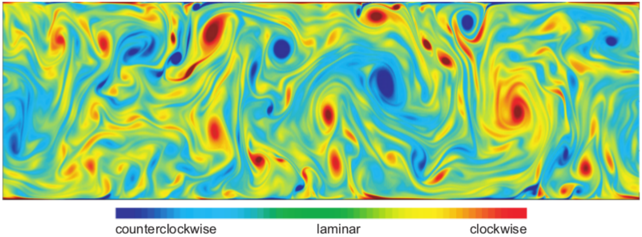
- **Vorticity**
  - Given a vector field $v : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, the vorticity of v, also called curl or rotor of v[2], is the quantity vector $\text{rot } v = (\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z}, \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x}, \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y})$
  - The vorticity vector characterizes the speed and direction of rotation of a given vector field at every point.
  - An informal definition of a vortex is a region where the vector field locally circles around a point called the vortex center.

- **Vorticity**

- The absolute value of the vorticity of a velocity field from a magnetohydrodynamic (MHD) simulation using blue-to-red colormap
- The field itself is visualized with arrow-capped stream tubes.
- Blue areas indicate low-vorticity, laminar regions.
- Red areas indicate high-vorticity regions



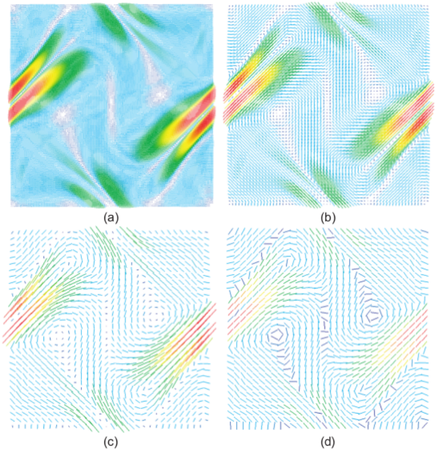
- **Vorticity**
- The following image visualizes the vorticity of a more complex turbulent 2D flow
- Blue and red indicates counter-clockwise and clockwise spinning vortices respectively
- Green indicates low-vorticity, laminar regions.
- The image clearly conveys the high complexity of the flow.



- **Vector Glyphs**
  - These are simplest, fastest and more popular techniques for visualizing vector fields
  - it associates a mapping between a vector glyph or vector icon with every simple point of the vector dataset
  - Various properties of the icon, such as location, direction, orientation, size, and color, are adjusted to reflect the value of the vector attribute it represents.
  - **Line glyphs**
    - Lines essentially show the position, direction, and magnitude of a set of vectors.
    - Given a vector dataset defined on a sampled domain D, we associate a line $l = (x, x+kv(x))$ with every sample point $x \in D$ that has a vector attribute $v(x)$.
    - The parameter k represents the scaling factor used to map the vector magnitudes to the geometric domain.
    - Oriented line glyphs are sometimes also called **hedgehogs** $\rightarrow$ the particularly spiky appearance of the visualization.

- **line glyph**

- line glyph, or hedgehog, visualization of a 2D vector field MHD dataset
- The images show the hedgehog visualization of the vector field uniformly subsampled in both x and y dimensions at a rate of 2 (a), a rate of 4 (b), a rate of 8 (c) and a rate of 8 but line glyphs all are scaled to the same length.



- **Line Glyphs**
  - The high-resolution vector datasets must be subsampled in order to be visualized with hedgehogs.
  - The clarity of hedgehog visualizations depends strongly on the glyph scaling factor.
  - Glyph should be as large as possible, since larger glyphs have an easier perceivable direction, but not too large
  - If we scale all glyphs to the same size → removes clutter, but eliminates the use of the glyph size (length) as a visual cue for the vector field magnitude.
  - linear scaling factor
  - non-liner scaling factor kv → has constrained minimal and maximal values or has a logarithmic, instead of linear → prevent clutter and guarantee glyph visibility → but drops the one-to-one relationship between vector magnitude and glyph length

- **cone and arrow glyphs**
  - For the same 2D vector field, visualized with 3D cone and arrow glyphs
  - These glyphs are have an advantage to convey a signed direction
  - line glyphs are able to convey unsigned direction
  - These glyphs also take more space to draw $\rightarrow$ they increase the clutter or require lower-resolution datasets.
  - An interesting compromise between arrows and lines is to use Gouraud shaded lines.
  - By using even more complex glyph shapes, we can encode more attributes than the vector field itself.
  - These are used to provide correlations between several scalar and vector fields.

- **cone and arrow glyphs**

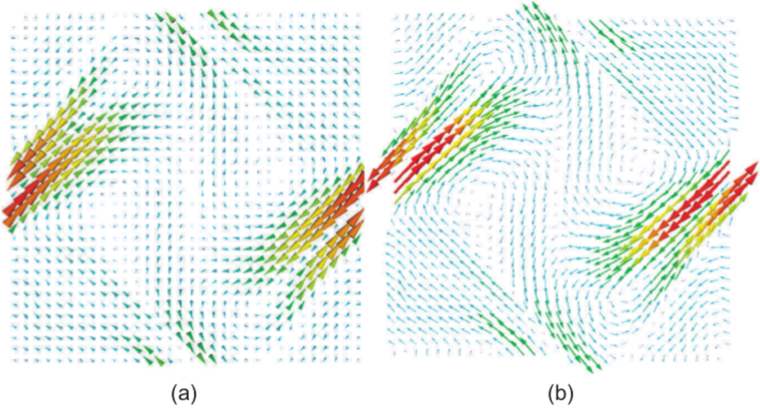


Figure 6.5. Different glyph types. (a) Cones. (b) Arrows.

- **Glyphs**
  - the power of expression of glyphs, number of attributes they can encode and minimal screen size needed by a glyph is an important characteristic of glyph-based visualizations.
  - Since a glyph takes more space than just a pixel, we cannot draw one glyph at every pixel of a given dataset $\rightarrow$ every pixel represents an data value.
  - All glyph visualizations share this inherent discreteness.
  - This affects the inverse image-to-data mapping.

- **Vector glyphs in 2D**
- Problems in Vector glyphs in 2D

- **Vector glyphs in 2D**

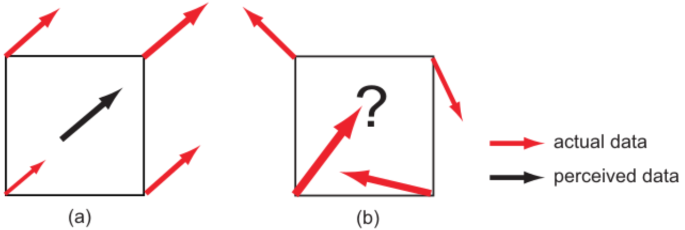
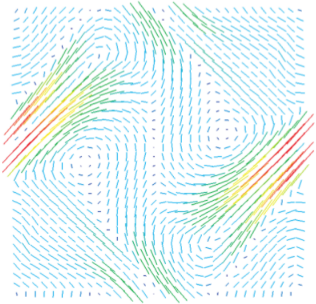


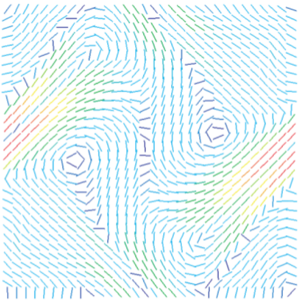
Figure 6.6. Visual interpolation of vector glyphs. (a) Small data variations are easily interpolated. (b) Large data variations create more problems.

- **Vector glyphs in 2D**
  - In the first case Fig-6.6(a)
  - we can easily interpolate mentally the displayed arrow glyphs and make the conclusion
  - 1. The vector field has an upper-right direction and orientation
  - 2. Increases in magnitude in this direction
  - In the second case Fig-6.6(b)
  - The vector field varies greatly between the vertices of the considered cell $\rightarrow$ it is harder to mentally interpolate between these four vector glyphs and get an idea of how the field actually behaves over the considered surface.
  - 1. More difficult to interpolate between directions and orientations
  - Glyph techniques produce a purely discrete visualization
  - we do not have to mentally interpolate between drawn pixels $\rightarrow$ the graphics hardware has done this task

- **Vector glyphs in 2D**
  - The regular pattern of the sample points present in uniform and rectilinear grids.



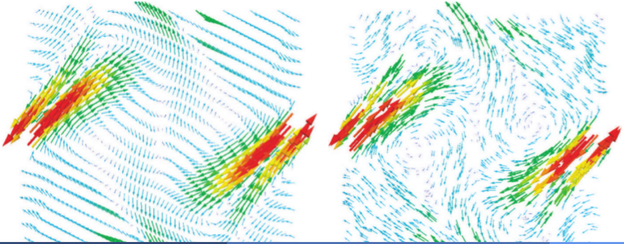
(c)



(d)

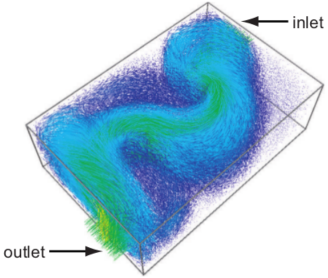
  - The problem is visible in the central area of the above figure
  - In these regions, the perception of the diagonal orientation of the vector glyphs is weakened by the regular vertical pattern of the uniformly distributed sampling points.

- **Vector glyphs in 2D**
  - The above problem affects the dense visualizations to a much lesser degree.
  - Solution - 1: Subsampled on a rectilinear grid rotated approximately 30 degrees with respect to the original dataset grid.
  - The undesired visual interference between the grid lines and glyph directions is clearly visible.
  - Subsampling the dataset using a randomly distributed set of points, the problem can be alleviated



- **Vector glyphs in 3D**

- Vector glyphs can be used to visualize 3D vector fields
- The beside figure shows an arrow glyph visualization of a 3D vector dataset sampled on a uniform grid containing 128 X 85 X 42 data points
- It describes the flow of water in a box-shaped basin
- It has inlet $\rightarrow$ located upper-right
- It has outlet $\rightarrow$ located lower-left



- Two obstacles that cause the sinuous behavior of the flow.
- Randomly subsampling the dataset to 100,000 points

- **Vector Color Coding**
  - Dense visualization (color mapped surfaces) has more advantages compared to sparse visualization (vector glyphs)
  - Can we develop a dense visualization for vector fields?
  - **Vector Color Coding**
  - It associates a color with every point of a given surface, on which we have defined a vector dataset
  - The color is used to encode the vector **orientation** and **direction** attributes.
  - The simple way to understand vector color coding is to use HSV (Hue, Saturation and Value) to represent color
  - Colors in HSV can be represented using Color Wheel

- **Vector Color Coding**
  - Every distinct **hue** corresponds to a different angle of the color wheel → red is 0^0, magenta is 60^0, blue is 120^0, cyan is 180^0, green is 240^0 and yellow is 300^0.
  - Saturation is represented as the distance from the wheel center to a given color point
  - Value is usually represented as a separate one-dimensional "luminance" parameter.
  - the vector magnitude can be encoded as the luminance → long vectors result in full color whereas shorter vectors tend to be represented as black.

- **Vector Color Coding on 2D surfaces**
  - Assume we have a color wheel of unit radius.
  - All vectors in the 2D dataset are scaled so that the longest one has unit length.
  - Under this assumption, every vector is represented by the color it points to if it is placed at the center of the color wheel.
  - The vector orientation is encoded in the hue.
  - The vector length in the value.
  - The saturation parameter is set to one. $\rightarrow$ we use only fully saturated colors

- **Vector Color Coding on 2D surfaces**

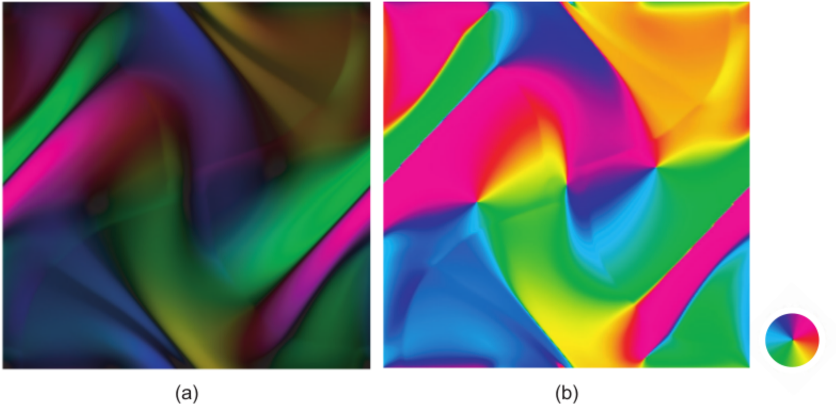


Figure 6.10. Vector color coding. (a) Orientation and magnitude. (b) Orientation only.

- **Vector Color Coding on 2D surfaces**
    - This image does not suffer from the sampling problems
    - Low-vector-magnitude regions can be easily detected as dark (low value) areas.
    - High-vector-magnitude regions show up as brightly colored areas.
    - This visualization is highly abstract.
    - The inverse mapping from hue to vector orientation takes more time. $\rightarrow$ users have to be trained extensively to interpret such images.
    - If we are interested only in the vector orientation and not the magnitude, we can set the value component to one, and we obtain the visualization shown in Figure 6.10(b).
    - The orientation patterns of the vector field are easier to distinguish $\rightarrow$ image is brighter

- **Displacement Plots**
  - Vector glyphs has been understood in terms of displaying trajectories.
  - The vector glyph with the origin at some point p can be seen as the trajectory p would follow in v(p) over a short time interval Δt.
  - The vector glyph shows both the start and end points of the trajectory $\rightarrow$ p and p+v(p)Δt.
  - Displacement plots take a different approach by showing only the end points of such trajectories.
  - Given a surface S $\in$ D inside the domain D of a vector field, where S is discretized as a set of sample points p_i, a displacement plot of S is a new surface S' given by the set of sample points $p'_i = p_i + kv'(p_i)$
  - v' is a vector field that controls the displacement of the surface S and k is the displacement factor

- **Displacement Plots**
- A natural interpretation of a displacement plot is to think of it as being the effect of displacing, or warping, a given surface in the vector field.

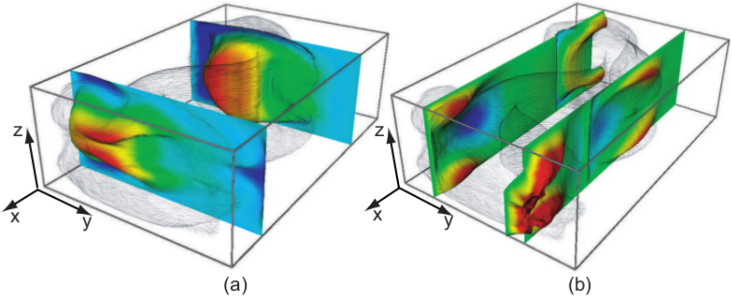


Figure 6.12. Displacement plots of planar surfaces in a 3D vector field.

- **Displacement Plots**
  - Both examples uses k=20
  - The displacement plots are colored by the vector field component on which the input surface is perpendicular.
  - Blue shows the minimal (negative) displacement
  - Red is the maximal (positive) displacement
  - Green indicates a non-displaced point with vector value close to zero.
  - The red areas, warped forward in the direction of the x-axis, indicate regions where the fluid flow strongly follows the inlet-to-outlet direction → figure-a
  - Blue regions are also interesting, as these indicate a backward flow that goes against the main stream

- **Displacement Plots**
  - displacement plots are sometimes also called warped plots
  - Advantages of Displacement Plots:
  - 1. they produce a visually continuous result
  - 2. these plots produce a more abstract, less intuitive visualization
  - Red is the maximal (positive) displacement
  - Green indicates a non-displaced point with vector value close to zero.
  - A displacement plot shows the motion of an object in the direction perpendicular to its surface.