Visualization and Imaging

Summer Term 2015

Notations

- Let $\Omega \subset \mathbb{R}^n$ be a Euclidean subspace, $\mathcal{M} \subset \mathcal{R}^n$ as smooth manifold, $T\mathcal{M}$ the corresponding tangent bundle, $I \subset \mathbb{R}$ an interval. A time-dependent vector field can be denoted by

$$u : \mathcal{M} \times I \to T\mathcal{M}, \quad u(x,t) \in T_x\mathcal{M},$$
$$u : \Omega \times I \to \mathbb{R}^n, \quad u(x,t) \in \mathbb{R}^n.$$

- A steady vector field can be denoted by

$$u : \mathcal{M} \to T\mathcal{M}, \quad u(x) \in T_x\mathcal{M},$$
$$u : \Omega \to \mathbb{R}^n, \quad u(x) \in \mathbb{R}^n.$$

Notations

- **Path line**: A particle at $x_0 \in \mathcal{M}$ at time t_0 follows a path line that $x_{path}(\cdot; x_0, t_0))$ that satisfies

$$\frac{dx_{path}(t; x_0, t_0)}{dt} = u(x(t; x_0, t_0), t)$$

- **Stream line (at time instance τ)**: A particle at $x_0 \in \mathcal{M}$ at time t_0 follows a stream line that $x_{stream}(\cdot; x_0, t_0))$ that satisfies

$$\frac{dx_{stream}(t; x_0, t_0)}{dt} = u(x(t; x_0, t_0), \tau)$$

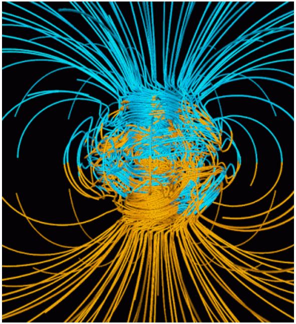
- **Streak line**: A set of particle is released at $x_0 \in \mathcal{M}$ at times $s \in [t_0, t]$. The streak line contains those points at time instance t that have been released x_0 at time s:

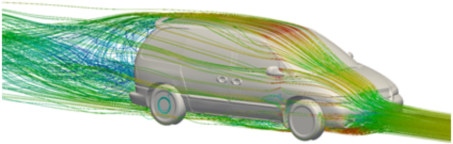
$$x_{streak}(s; x_0, t) = x_{path}(t; x_0, s)$$

Illustration: http://crcv.ucf.edu/projects/streakline_eccv/

- For steady vector fields all definitions are identical

Example





For visualization use shading/lighting from before

Example

Streak Lines

Computation of Path Lines

- Integral equation for path lines $x(t)$:

$$x(t) = \int_{t_0}^{t} u(x(s), s) ds.$$

- Forward discretization:

$$x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} u(x(s), s) ds \approx x(t_{k+1}) = x(t_k) + u(x(t_k), t_k)\Delta t$$

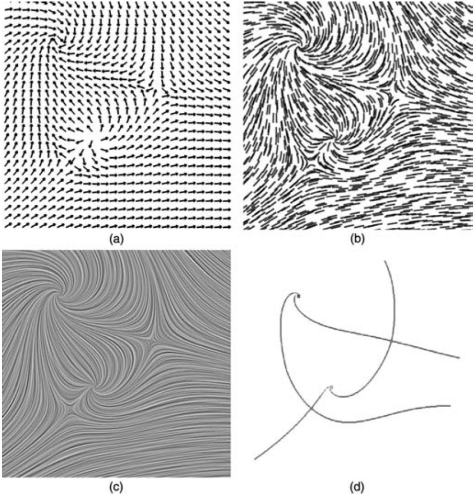
- Backward discretization:

$$x(t_k) = x(t_{k+1}) - \int_{t_k}^{t_{k+1}} u(x(s), s) ds \approx x(t_k) = x(t_{k+1}) - u(x(t_{k+1}), t_{k+1})\Delta t$$

- For the evaluation of $u(x(t_k), t_k)$, interpolation is necessary

Texture Visualization of Vector Flows

(c) High Resolution Visualization by LIC



Texture Visualization of Vector Flows

Spot Noise Visualization

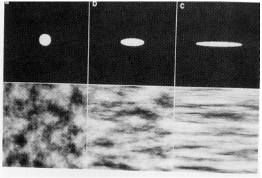
J.J. van Wijk: Spot Noise: Texture Synthesis for Data Visualization, Computer Graphics (1991). Filtered image $D : \mathbb{R}^n \to \mathbb{R}$ can be represented by

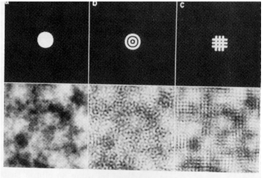
$$D(x) = \sum_{i=1}^{N} a_i h(x - x_i)$$

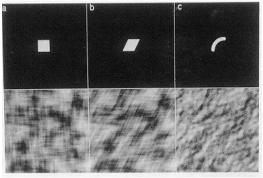
where a_i are randomly distributed intensities (with mean value zero) and $x_i \in \Omega$ randomly distributed points. The spot function $h : \Omega \to \mathbb{R}$ may contain information on the vector field (e.g., streched in direction of the vector field $u(x_i)$ at spot x_i by a factor $1 + |u(x_i)|$ and in orthogonal direction by $1/(1 + |u(x_i)|)$).

Texture Visualization of Vector Flows

Sample Spots







Texture Visualization of Vector Flows

General Space-Time Correlation via Integration

Compute filtered slice $D_t : \mathbb{R}^n \to \mathbb{R}$ by

$$D_t(x) = \int_{-\infty}^{\infty} K(s) I(Z(s; x, t)) ds$$

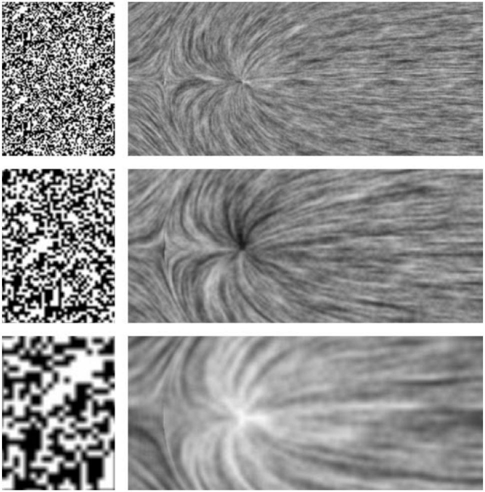
where $K : \mathbb{R} \to \mathbb{R}$ is some convolution kernel (along time), $I : \mathbb{R}^n \to \mathbb{R}$ denotes an intensity (which is chosen to be constant along some trajectory $Y(\cdot; x, t)$), and $Z(\cdot; x, t)$ denotes another trajectory with $Z(t; x, t) = (x, t)$.

Texture Visualization of Vector Flows

- **Line Integral convolution (LIC)**:
  Choose $Y(s; x, t) = (x, s)$ and $I(x, t) = \Phi(x)$. Φ is a random field with values $\Phi(x) \in [0, 1]$ (e.g., white noise). The trajectory $Z(s; x, t) = (\bar{x}(s; x, t), s)$ is the path line of the vector field u.

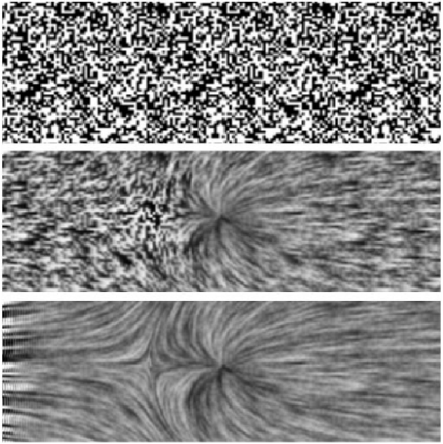
Texture Visualization of Vector Flows

Illustration of Initialization



Texture Visualization of Vector Flows

Illustration of Speed



Texture Visualization of Vector Flows

Comparison of Spot Noise and LIC (see, e.g., W. de Leeuw, R. van Liere: Comparing LIC and Spot Noise)

spot noise	LIC
random spot intensity	random input texture
spot function	kernel shape
spot scaling	kernel length variation
standard spots	DDA convolution
bent spots	streamline convolution
spot advection	texture advection

Texture Visualization of Vector Flows

Comparison of Spot Noise and LIC

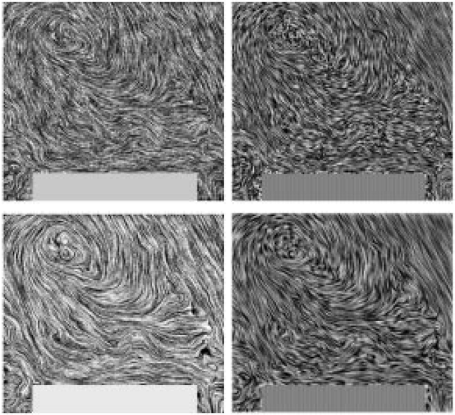


Figure 9: LIC (left) and spot noise (right) images with equal pixel coverage using a filter length of 20 and a spot radius of 0.005. (top) and using a filter length of 40 and a spot radius of 0.007. (bottom)

Texture Visualization of Vector Flows

- **Image Based Flow Visualization (IBFV)**:
  Choose $Y(s; x, t) = (x, s)$ and, e.g., $I(x, t) = \Psi((t + \Phi(x)) \bmod 1)$ for some function Ψ. The trajectory $Z(s; x, t) = (\bar{x}(s; x, t), s)$ is the path line of the vector field u.
  https://www.youtube.com/watch?v=OQ12UjmVt1M
- **Lagrangian Eulerian Advection (LEA)**:
  Choose $Y(s; x, t) = (\bar{x}(s; x, t), s)$ to be the path line of the vector field u and, e.g., $I(x, t) = \Psi((t + \Phi(x)) \bmod 1)$ for some function Ψ. The trajectory $Z(s; x, t) = (x, s)$ is parallel to the time axis.
- LEA and IBFV are dual.

Texture Visualization of Vector Flows

Illustration of IBFV for different Ψ

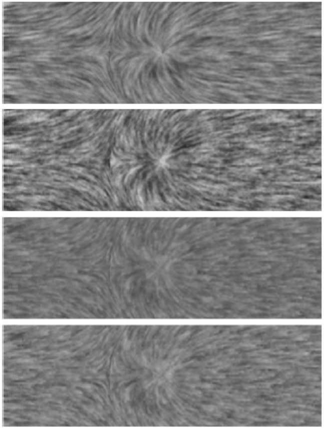


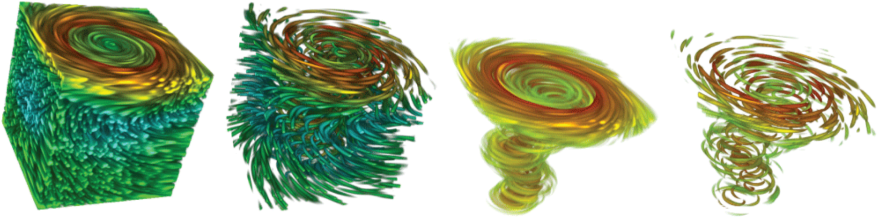
Figure 5: Different profiles $w(t)$. From top to bottom: (a) cosine, (b) square, (c) exponential decay, (d) saw tooth.

Texture Visualization of Vector Flows

- **Dynamic Line Integral convolution (DLIC)**:
  Choose $Y(s;x,t) = (\bar{y}(s;x,t), s)$ to be the path line of a secondary vector field v. and $I(x,t) = \Phi(x)$. Φ is a random field with values $\Phi(x) \in [0,1]$ (e.g., white noise). The trajectory $Z(s;x,t) = (\bar{x}(s;x,t), s)$ is the path line of the vector field u.
- DLIC makes sense for the representation of electric fields that are driven by time-dependent charge distributions.

Example

3D texture visualization



(a) (b) (c) (d)

Notations

- Let $\Omega \subset \mathbb{R}^m$, $I \subset \mathbb{R}$ an interval. A time-dependent second-order tensor field can be denoted by

$$\mathbf{T} : \Omega \times I \to \mathbb{R}^{n \times p}, \quad u(x,t) \in \mathbb{R}^{n \times p}.$$

  We choose $\mathbf{n} = \mathbf{p}$ and (mostly) **symmetric** tensors (6-D).

- A steady second-order tensor field can be denoted by

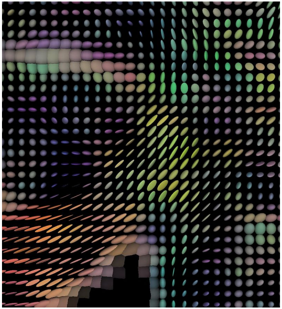
$$\mathbf{T} : \Omega \to \mathbb{R}^{n \times p}, \quad u(x) \in \mathbb{R}^{n \times p}.$$

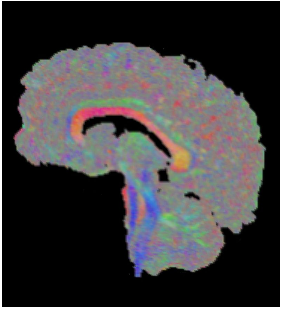
- **Examples**: Deformation tensor (not symmetric), Stress tensor (force per unit area in a point $x \in \Omega$ acting on surfaces of arbitrary direction going through x, symmetric), Diffusion Tensor (DT-MRI: diffusion direction of water molecules in body tissue at a location $x \in \Omega$, symmetric), Gradiometry (symmetric)

Shape Spaces and Invariants

Representation of tensor fields via ellipsoids/glyphs (eigenvalues and eigenvectors of $\mathbf{T}(x)$ determine the major axes and lengths of the ellipsoid at point x) or representation by slices and colors to indicate main directions.

In general, different **anisotropy metrics** influence the shape and representation of tensor information.





Shape Spaces and Invariants

Tensor invariants are properties of the tensor or functions that stay invariant under orthogonal coordinate transformations

- eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$
- determinant
- trace
- characteristic polynomial

Shape Spaces and Invariants

Shape Space The vector space with eigenvalues $\lambda_1, \lambda_2, \lambda_3$ denoting its threes axes is called shape space. Coordinates in a different reference frame are called shape descriptors.

Basis in Shape Space Any set of three linearly independent invariants $I_i(\lambda_1, \lambda_2, \lambda_3)$, $i = 1, 2, 3$, define a basis in shape space via ∇I_i, $i = 1, 2, 3$. Each invariant describes a two-dimensional subset (level set) in shape space.

Shape Spaces and Invariants

Glyphs Iconic figures that illustrate tensor information via shape, color, transparency, etc.

- Ellipsoids (isosurface $\frac{x_1^2}{\lambda_1^2} + \frac{x_2^2}{\lambda_2^2} + \frac{x_3^2}{\lambda_3^2} = 1$ or the set $\{\mathbf{T}y | y \in \mathbb{R}^3, |y| = 1\}$)
- Deformation glyph (the set $\{(1 + \mathbf{T})y | y \in \mathbb{R}^3, |y| = 1\}$), allows representation of negative eigenvalues
- Anisotropy measures: linear anisotropy, planar anisotropy, isotropy (sphere):

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}, \quad c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}, \quad c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$$

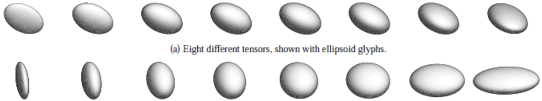
Fractional Anisotropy:

$$FA = \sqrt{\frac{3}{2}} \frac{\sqrt{(\lambda_1 - \bar{\lambda})^2 + (\lambda_2 - \bar{\lambda})^2 + (\lambda_3 - \bar{\lambda})^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

with $\bar{\lambda} = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3)$.

Shape Spaces and Invariants

Ambiguity of Ellipsoidal representation:



(a) Eight different tensors, shown with ellipsoid glyphs.

(b) Same eight glyphs, but with a different viewpoint.

Figure 5: *From some viewpoints, ellipsoids poorly convey tensor shape.*

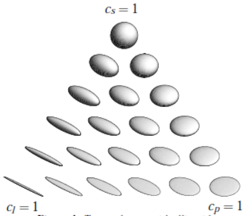


Figure 4: *Tensor shapes, with ellipsoids.*

Shape Spaces and Invariants

- Superquadrics:

$$q_3(x) = \left(x_1^{\frac{2}{\alpha}} + x_2^{\frac{2}{\alpha}}\right)^{\frac{\alpha}{\beta}} + x_3^{\frac{2}{\beta}}$$

$$q_1(x) = \left(x_3^{\frac{2}{\alpha}} + x_2^{\frac{2}{\alpha}}\right)^{\frac{\alpha}{\beta}} + x_1^{\frac{2}{\beta}}$$

If $c_l \geq c_p$:

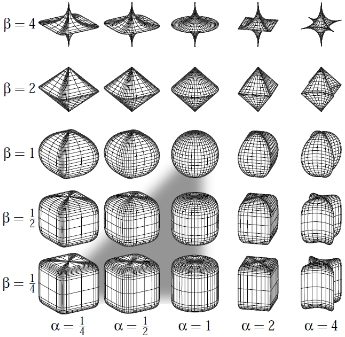
$$\alpha = (1 - c_p)^\gamma, \quad \beta = (1 - c_l)^\gamma, \quad q_1(x) = 1$$

If $c_l < c_p$:

$$\alpha = (1 - c_l)^\gamma, \quad \beta = (1 - c_p)^\gamma, \quad q_3(x) = 1$$

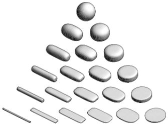
Shape Spaces and Invariants

Shapes of Superquadrics (Superquadric Tensor Glyphs, G. Kindlemann (2004)):

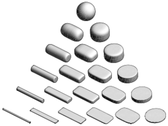


Shape Spaces and Invariants

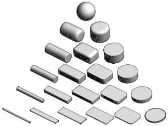
Tensor Shapes for superquadrics with different γ:



(a) $\gamma = 1.5$



(b) $\gamma = 3.0$



(c) $\gamma = 6.0$

Fibre Tracking

Streamline Integration: Let $e_1(x)$ be the normalized eigenvector to the largest eigenvalue λ_1 of $\mathbf{T}(x)$. Fibre trajectory

$$x(t) = \int_0^t e_1(x(s))ds.$$

Forward scheme

$$x(t_{k+1}) = x(t_k) + e_1(x(t_k))\Delta t.$$

Fibre Tracking: Choose initial point $x(t_0)$ to be a maximum of $c_l(x)$. Choose a threshold $K \in [0, 1]$ such that a point x is considered part of a fibre if $c_l(x) \geq K$. Compute fibre starting from $x(t_0)$ and use forward (or backward) scheme to compute $x(t_1)$. If $x(t_1) \geq K$, it is part of the fibre, iterate until the fibre stops.

Fibre Tracking

Smoothing along Fibre: Instead of $e_1(x)$ use a smoothed version $\bar{e}_1(x)$ in the streamline Integration. $\bar{e}_1(x)$ denotes the principal eigenvalue of a smoothed tensore field $\bar{T}(x)$. The smoothing should emphasize information along the fibre, e.g. by moving least squares (MLS): $\bar{T}(x)$ is obtained by minimizing the functional

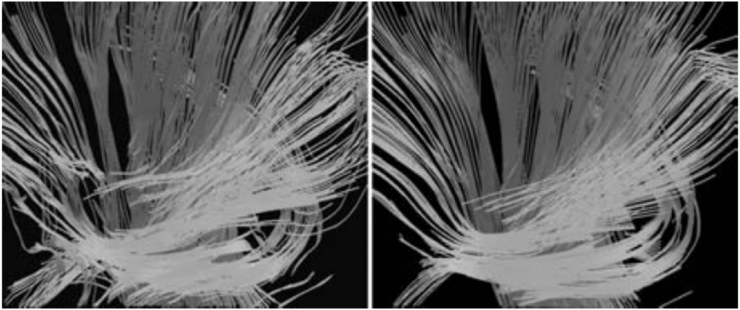
$$\mathcal{F}(F) = \int_{\mathbb{R}^3} K(y-x)(T(y) - F(y-x))^2 dy,$$

where $\bar{T}(x) = F(0)$, and the kernel K depends on the location x (e.g., by adapting the shape of K to the eigenvectors and eigenvalues of $\bar{T}(x(t_{i-1}))$ in the previous time step of the streamline integration)

The choice of K to be a scaled Gauss kernel relates to linear diffusion. *How could nonlinear diffusion be inlcuded in the above procedure?*

Fibre Tracking

Fibre Tracking result for unfiltered and MLS filtered setting



⟶ Do LIC representations of the eigenvector field e_1 reflect fibres?