CS 519: Scientific Visualization

Tensor Visualization

Eric Shaffer

Some slides adapted Alexandru Telea, Data Visualization Principles and Practice
1. What is a tensor?
   - Describe in terms of principal component analysis

2. Basic tensor visualization
   - Component visualization
   - Anisotropy visualization
   - Major eigenvector visualization

3. Application: Fiber tracking
   - Basic fiber tracking
   - Stream tubes
   - Hyperstreamlines
What is a tensor?

Explanation 1: Dimensionality
- scalar: a 0D array of values e.g. 1 value
- vector: a 1D array of values e.g. 3 values
- tensor: a 2D matrix of values e.g. $3 \times 3 = 9$ values

Explanation 2: Analysis
- scalar: magnitude (of some signal at a point in space)
- vector: magnitude and direction (of some signal at some point in space)
- tensor: variation of magnitude (of some signal at some point in space)
What is a tensor?

**Explanation 3: As a function**

- **scalar:** at \( x \in \mathbb{R}^3 \), measure some value \( s \in \mathbb{R} \)
- **vector:** at \( x \in \mathbb{R}^3 \), measure some magnitude and direction \( v \in \mathbb{R}^3 \)
- **tensor:** at \( x \in \mathbb{R}^3 \) and in a direction \( v \in \mathbb{R}^3 \), measure some magnitude \( s \in \mathbb{R} \)

**Fields**

So we have different kinds of fields (i.e. *functions* of a variable \( x \in \mathbb{R}^3 \)):

**Scalar fields** \( s : \mathbb{R}^3 \to \mathbb{R} \)

**Vector fields** \( v : \mathbb{R}^3 \to \mathbb{R}^3 \)

**Tensor fields** \( T : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R} \)
**Curvature in 1D**

- take a curve \( c \subseteq \mathbb{R}^3 \)
- locally, \( c \) can be described as a function \( y = f(x) \)

\[ y = f(x) \]

- curvature of \( f \) \( C(x) = \frac{\partial^2 f}{\partial x^2} \) (2\textsuperscript{nd} derivative of \( f \))

- analytically: \( C(x) = \) how quickly the normal \( n_c \) changes around \( x \)
  (why? Because the tangent to \( c \) is \( \frac{\partial f}{\partial x} \) and its change is \( \frac{\partial^2 f}{\partial x^2} \))
Curvature in 2D

- take a surface $S \subset \mathbb{R}^3$
- at each $x_0 \in S$
  - take a coordinate system $xyz$ with $x,y$ tangent to $S$ and $z$ along $n_S$
  - locally, $S$ can be described as a function $z = f(x,y)$

How to describe 2D curvature?

- 1D analogy: how quickly the normal $n_S$ changes around $x_0$
- problem: we have a surface – in which direction to look for change?

We must compute

$$C(x,s) = \frac{\partial^2 f(x)}{\partial s^2}$$
for any direction $s$. 

- slice plane $P$
- curve $C$
- surface $S$
The Curvature Tensor

\[ C(x, s) = \frac{\partial^2 f(x)}{\partial s^2} \]

- recall our definition of a tensor \( T : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R} \)? The above is precisely that

Also note that

\[ \frac{\partial^2 f}{\partial s^2}(x_0) = s^T H s. \]

where \( H \) is the so-called Hessian of \( f \)

\[
H = \begin{pmatrix}
\frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\
\frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2}
\end{pmatrix}
\]

In other words, if we have \( H \), we can compute the curvature tensor
- at any point \( x_0 \)
- in any direction \( s \)
The Curvature Tensor

However, there's a problem with the previous definition

- we need to construct local coordinate systems at every point on \( S \)
- not obvious how to do that….

**General solution:**

Describe \( S \) as an implicit function (i.e. the zero-level isosurface of a function)

\[
S = \left\{ x \in \mathbb{R}^3 \mid f(x) = 0 \right\} \quad \text{for a given } f : \mathbb{R}^3 \to \mathbb{R}
\]

Then, we still have

\[
\frac{\partial^2 f}{\partial s^2}(x_0) = s^T H s \quad \text{where } H \text{ is the } 3 \times 3 \text{ Hessian matrix}
\]

**Conclusion**

- A curvature tensor is fully described by a 3x3 matrix of 2\(^{nd}\) order derivatives
The Diffusion Tensor

- consider an anisotropic material (e.g. tissue in the human brain)
- water diffuses in this tissue
  - strongly along neural fibers
  - weakly across fibers

Actual image of a dissected human brain

Diffusion tensor

\[ D(x, s) = \frac{\partial^2 f(x)}{\partial s^2} \]

diffusivity at a point \(x\) in a direction \(s\)

Diffusion tensor: measured by a technique called **DT-MRI** (diffusion tensor magnetic resonance imaging)
The Diffusion Tensor

First visualization try

- compute hessian $H = \{h_{ij}\}$ in $\mathbb{R}^3$
- select some slice of interest
- visualize all components $h_{ij}$ using e.g. color mapping

Simple, but not very useful

- we get a lot of images (9)…
- we see the tensor is symmetric…
- …but we don’t really care about diffusion along $x, y, z$ axes!
• fix some point $x_0$ on the surface
• compute $C(x_0,s)$ for all possible tangent directions $s$ at $x_0$
• denote $\alpha = \text{angle of } s \text{ with local coordinate axis } x_0$

So we have

$$\frac{\partial^2 f}{\partial s^2} = s^T H s = h_{11} \cos^2 \alpha + (h_{12} + h_{21}) \sin \alpha \cos \alpha + h_{22} \cos^2 \alpha$$

Now, let’s look for the values of $\alpha$ for which this function is extremal!
Our curvature (as function of $\alpha$) is extremal when $\frac{\partial C}{\partial \alpha} = 0$

This is equivalent to a system of equations

\[
\begin{cases}
  h_{11} \cos \alpha + h_{12} \sin \alpha = \lambda \cos \alpha \\
  h_{21} \cos \alpha + h_{22} \sin \alpha = \lambda \sin \alpha,
\end{cases}
\]

which in matrix form is $Hs = \lambda s$ or $(H - \lambda I)s = 0$

Since we’re looking for the non-trivial solution $s \neq \mathbf{0}$ this means

$$\det(H - \lambda I) = (h_{11} - \lambda)(h_{22} - \lambda) - h_{12}h_{21} = 0$$

Solving the above 2$\text{nd}$ order equation in $\lambda$ yields

• two real values $\lambda_1, \lambda_2$ eigenvalues (principal values) of tensor

Plugging $\lambda_1, \lambda_2$ into $Hs = \lambda s$ yields

• two direction vectors $s_1, s_2$ eigenvectors (principal directions) of tensor

**Summarizing**

• Given a 2x2 tensor, we can compute its principal directions and values
  • directions: those in which tensor has extremal (minimal, maximal) values
    Can be shown that eigendirections are orthogonal to each other
  • values: the actual minimal and maximal values

*For full details, see Sec. 7.1*
How about a 3x3 tensor, like the diffusion tensor?

- 3 eigenvalues, 3 eigenvectors (computed similarly, see Sec. 7.1)

Say we order eigenvalues (and their vectors) as $\lambda_1 > \lambda_2 > \lambda_3$.

- $\lambda_1, s_1$: **major** eigenvector i.e. direction of strongest diffusion
- $\lambda_2, s_2$: **medium** eigenvector (no particular meaning)
- $\lambda_3, s_3$: **minor** eigenvector i.e. direction of weakest diffusion

What if two or more eigenvalues are equal (so we cannot fully order them all)?

<table>
<thead>
<tr>
<th>Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a,b)</td>
<td>all values ordered: unique eigendirections</td>
</tr>
<tr>
<td>c,d)</td>
<td>equal eigenvalues: eigendirections not determined (any two orthogonal vectors tangent to surface are valid eigendirections)</td>
</tr>
</tbody>
</table>
How to use PCA for visualization?

Visualize mean diffusivity $\mu = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3)$

white: strong mean diffusivity
black: weak mean diffusivity
Principal Component Analysis

Linear diffusivity

\[ c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \]

Fractional anisotropy

\[ FA = \sqrt{\frac{3}{2} \sqrt{\frac{\sum_{i=1}^{3} (\lambda_i - \mu)^2}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}}, \quad \text{where} \quad \mu = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3) \]

Relative anisotropy

\[ RA = \sqrt{\frac{3}{2} \frac{\sum_{i=1}^{3} (\lambda_i - \mu)^2}{\lambda_1 + \lambda_2 + \lambda_3}} \]

All above measures estimate how much ‘fiber-like’ is the current point

white: strong fibers
Exploit the directional information in the eigenvectors

• major eigenvector $e_1$: along the **strongest** diffusion direction
• for DTI tensors, it thus indicates fiber directions

Directional color coding

• like for vectors (see Module 4)
• use simple colormap
  \[
  R = |e_1 \cdot x|, \\
  G = |e_1 \cdot y|, \\
  B = |e_1 \cdot z|.
  \]
• use vector glyphs / hedgehogs
• seed only points where $c_1, FA$ or $RA$ are large enough (other points don’t cover fibers)

• OK, but takes training to grasp
Vector PCA

**Directional color coding (2\textsuperscript{nd} variant)**
- like before, but simply color points by direction
- no glyphs drawn
- no occlusion/clutter
- direction coded \textbf{only} by color – less intuitive images
So far, we only visualized the major eigenvector $e_1$
- so we reduced a tensor field to a vector field
- we **threw away** existing information ($e_2, e_3$)

**Ellipsoid glyph**: Use all eigenvalues + eigenvectors
- orient glyph along eigensystem ($e_1, e_2, e_3$)
- scale it by eigenvalues ($\lambda_1, \lambda_2, \lambda_3$)

Shapes that an ellipsoid glyph can assume:
- **Strong fiber-like structures**
- **Fiber sheet**
- **Isotropic diffusion**
Tensor Glyphs

Can use other glyph shapes besides ellipsoids

- Ellipsoids
- Cuboids
- Cylinders
- Superquadrics
Tensor Glyphs

Zoom-in on brain DT-MRI dataset

a) ellipsoids
b) cuboids
c) cylinders
d) superquadrics

Superquadrics look arguably most ‘natural’

For full details, see Sec. 7.5
Fiber Tracking

Reuse some other vector visualization methods
- consider major eigenvector field
- trace streamlines
  - **seed:** in regions with high anisotropy (i.e. where fibers are)
  - **stop:** when anisotropy gets too low (i.e. when we leave fibers)

This method is also called **tractography**
Generalize stream tubes

- trace stream tubes in major eigenvector field (like so far)
- use an **elliptic** cross-section
  - oriented along medium + minor eigenvectors
  - scaled with medium + minor eigenvalues

Tube cross-section shows diffusion across fibers

- Thin, round tubes: we’re in a fiber **bundle**
- Thick, flat tubes: we’re in a fiber **sheet**
- Thick, round tubes: we’re **exiting** a fiber
Tensor Visualization Summary

- fundamentally harder than vector visualization
  - 9 values per point (!)
  - classical vector visualization problems (occlusion, seeding, etc)

- methods
  - reduce tensors to scalars (tensor components, PCA or anisotropies)
  - directional and/or color coding of major eigenvector
  - tensor glyphs
  - streamlines, stream tubes
  - hyperstreamlines