Superquadric Glyphs for Symmetric Second-Order Tensors

Thomas Schultz, Gordon L. Kindlmann
Computer Science Dept, Computation Institute
University of Chicago

Symmetric Tensor Representations

\[
D = \begin{pmatrix}
3.08 & 1.21 & 0.77 \\
1.21 & 3.85 & 1.98 \\
0.77 & 1.98 & 5.06
\end{pmatrix}
\]

\[
D = \begin{pmatrix}
-0.33 & -0.74 & 0.59 \\
-0.59 & -0.74 & -0.74 \\
-0.74 & 0.59 & 0.33
\end{pmatrix}
\]

\[
\begin{pmatrix}
7 & 0 & 0 \\
0 & 3 & 0 \\
0 & 0 & 2
\end{pmatrix}
\begin{pmatrix}
-0.33 & -0.59 & -0.74 \\
-0.74 & -0.59 & 0.59 \\
0.59 & -0.74 & 0.33
\end{pmatrix}
\]

Eigenvalues tell us about Positive Definiteness

[Kindlmann 2004]
Symmetric Tensor Representations

\[ D = \begin{pmatrix} 2.03 & 2.52 & 0.19 \\ 2.52 & 2.22 & 2.71 \\ 0.19 & 2.71 & 4.74 \end{pmatrix} \]

Why care about indefinite tensors?

Indefinite Symmetric Tensors arise in many fields:

- Hessians
- Stress Tensors
- Rate-of-deformation Tensors
- Geometry Tensors
Principles for Tensor Glyph Design

Faithful and expressive visualization requires:

- **Preservation of Symmetry**: Glyph should have same symmetries as the tensor
  \[ D = TDT^{-1} \iff G(D) = TG(D) \]

- **Continuity**:
  \[ D_1 \approx D_2 \iff \text{appearance}(G(D_1)) \approx \text{appearance}(G(D_2)) \]

- **Disambiguity**:
  \[ D_1 \neq D_2 \iff \text{appearance}(G(D_1)) \neq \text{appearance}(G(D_2)) \]

Principles for Tensor Glyph Design

Natural for a wide range of applications:

- **Invariance under scaling**:
  \[ G(D) = s(\|D\|)B\left(\frac{D}{\|D\|}\right) \]

- **Invariance under projection to eigenplanes**:
  \[ G(PDP^T) = PG(D) \]
Glyph Coloring

We color each point \( \mathbf{x} \) on the glyph by

\[
\text{sign}(\mathbf{x}^T \mathbf{D} \mathbf{x})
\]

\[
\rightarrow \quad + \text{ red} \quad - \text{ blue}
\]

Satisfies

✓ Preservation of Symmetry
✓ Continuity
✓ Disambiguity

Our Fundamental Glyph Equation

\[
G(\mathbf{D}) = s(\|\mathbf{D}\|) \mathbf{R} \tilde{\Lambda} B(\tilde{\lambda}_i)
\]

scaling \quad rotation \quad \text{basis shape}

normalized eigenvalues
Explicit Ellipses

\[ \mathbf{D} \mathbf{x} \text{ for } \|\mathbf{x}\|=1 \]

- Symmetry Preservation
- Disambiguity

Implicit Ellipses

\[ \mathbf{x}^\top \mathbf{D}^{-2} \mathbf{x} = 1 \]

“sign preserving”

Unbounded Surfaces
“Exp-Ellipses”

\[ \exp(D) x \text{ for } ||x||=1 \]

\checkmark\ Disambiguity

Reynolds Glyphs

\[ (x^TD^{-2}x)x \text{ for } ||x||=1 \]

\checkmark\ Invariance under Projection
\checkmark\ Occlusions in 3D
New Glyph for 2D symmetric tensor

- Meet disambiguity, symmetry preservation req.
- Glyph shape shows eigenvalue sign differences
- Convex indicates same sign (positive-definite, negative-definite)
- Concave indicates different sign (indefinite)

From 2D to 3D tensor glyphs

Understand space of 3D symmetric tensor shapes

Scaffold (prototype) 3D glyphs with 2D glyphs

Find mapping from 3D tensor shapes to 3D superquadrics
Space of 3D tensor shape

Eigenvalue sorting \( \lambda_1 \geq \lambda_2 \geq \lambda_3 \) \( \Rightarrow \) **Lune of tensor shape**

\[ \lambda_1 = \lambda_2 = \lambda_3 > 0 \]

**Positive Definite**

**Negative Definite**

**Indefinite** (mixed eval signs)

Space of 3D tensor shape

Mercator projection

Kindlmann “Superquadric Tensor Glyphs”, Vissym ’04
Space of 3D tensor shape

Cubic projection

unfold, shear → unit square of tensor shape

From 2D to 3D tensor glyphs

Understand space of 3D tensor shapes

Scaffold (prototype) 3D glyphs with 2D glyphs

Find mapping from 3D tensor shapes to 3D superquadrics
From 2D to 3D tensor glyphs

Understand space of 3D tensor shapes

Scaffold (prototype) 3D glyphs with 2D glyphs

Find mapping from 3D tensor shapes to 3D superquadrics

Invariance under eigenplane projection

⇒ orthogonal 2D glyphs form scaffold for 3D glyphs

Can use these 3D shapes to guide next step ...
From 2D to 3D tensor glyphs

Understand space of 3D tensor shapes

**Scaffold** (prototype) 3D glyphs with 2D glyphs

Find mapping from 3D tensor shapes to 3D superquadrics
\( \lambda_i = 0 \) hides discontinuity.

Slight shape problem.
Meet the glyphs ...

Halos: improve visibility, colormap of tensor trace

The Glyph Rendering Pipeline

\[ G(D) = s(\|D\|) R \tilde{A} B(\tilde{\lambda_i}) \]

4. Scale and rotate (modelview matrix)

1. Map \( \lambda_i \) to (u,v)

3. Generate Base Glyph

2. Map (u,v) to (\( \alpha, \beta \))

5. Color code sign (fragment shader)
Using a Palette of Base Geometry

Precompute and re-use superquadric geometry on a grid in $(\alpha, \beta)$ space

Implementation & Performance

- Geometry-based implementation
- Performance:
  - 3000 full-resolution glyphs
  - 1800x1000 viewport
  - 25fps (incl. halos) on NVIDIA Quadro FX 1800

Most code is in Teem; please see tutorial on http://www.ci.uchicago.edu/~schultz/sphinx/
Why not use Reynolds Glyphs?

Colored Reynolds Glyphs clearly show negative eigenvalues…

…unless you’re viewing them from an unfortunate direction.

Why Prefer Superquadrics?

Superquadrics remain legible from a wide range of directions.
What About Traceless Tensors?

Existing Traceless Superquadric Glyphs
[Jankun-Kelly/Mehta, Vis06]

Our new Superquadric Glyphs

Hessians vs. Geometry Tensors

Geometry Tensors

Hessians

See how geometry tensors are related to Hessians

\[ G = (I-\nn\nn^T)H(I-\nn\nn^T)/||g|| \]
Inspecting Smoothing Image Filters

Original Data  Gaussian Smoothing  Total Variation Flow

Equal Remaining Variance

What are Differences?

Let’s look at Hessians…
Inspecting Smoothing Image Filters

Gaussian Filtering **Smoothes Hessians.**
Leads to “smearing out” into flat regions and **cancellation effects.**

TV flow creates **flat regions** and **sharp edges** between them.
Results, turbulent jet flow

LIC

Results, turbulent jet flow

LIC, modulate contrast by velocity
Results, turbulent jet flow

Rate-of-Deformation Tensor = \( (\nabla \mathbf{v} + \nabla^T \mathbf{v}) / 2 \)

Ellipses with \( \lambda \rightarrow \exp(\lambda) \)

Results, turbulent jet flow

Rate-of-Deformation Tensor = \( (\nabla \mathbf{v} + \nabla^T \mathbf{v}) / 2 \)

New glyphs, \( s(\|D\|) \propto \|D\| \)
Results, turbulent jet flow

Rate-of-Deformation Tensor = \((\nabla \mathbf{v} + \nabla^T \mathbf{v})/2\)

New glyphs, \(s(\|D\|) \propto \|D\|^{1/2}\)

Results, double point stress field

Can now see tensors underlying hyperstreamlines \(s(\|D\|) \propto \|D\|^{1/10}\)
Conclusions

• Presented a new set of tensor glyphs
  • Can visualize all symmetric 3D 2nd-order tensors
• Glyphs design guided by principles
  • Need disambiguity, avoid misleading symmetries
  • Eigenplane projection invariance ⇒ scaffolds
• Future: still no satisfactory glyphs for:
  • General 2nd-order (non-symmetric) tensors
  • Rotations

Acknowledgements

• Funding: DAAD
• Flow data: Wolfgang Kollmann, UC Davis
• Symmetry and Continuity discussion:
  2009 Dagstuhl Scientific Visualization Seminar 09251

Thank you