Diffusion Tensor Visualization

and the Teem Software that Makes it Go

Gordon Kindlmann
gk@bwh.harvard.edu

Laboratory of Mathematics in Imaging
Department of Radiology
Brigham & Women's Hospital
Harvard Medical School

Outline

Visualization:

DWI Data source

Space of Tensor Shape

Volume Rendering

Superquadric Glyphs

Glyph packing

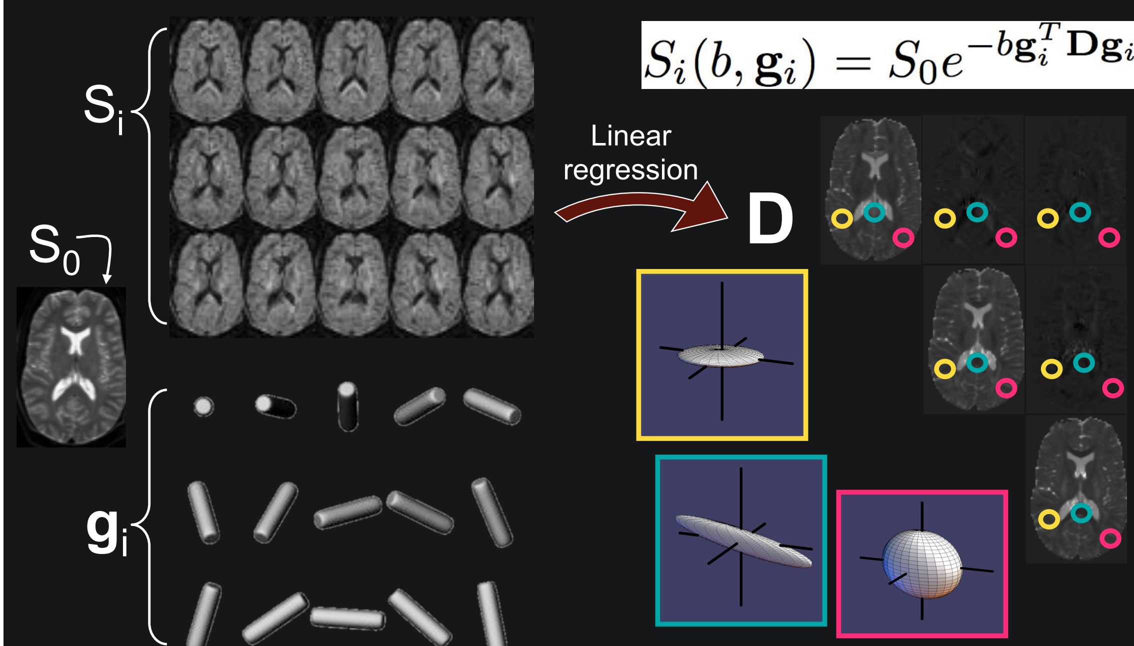
Outline

<u>Visualization</u>:

- **DWI Data source**
- Space of Tensor Shape
- Volume Rendering
- Superquadric Glyphs
- Glyph packing

Diffusion weighted, tensor images

Single Tensor Model (Basser et al. 1994)



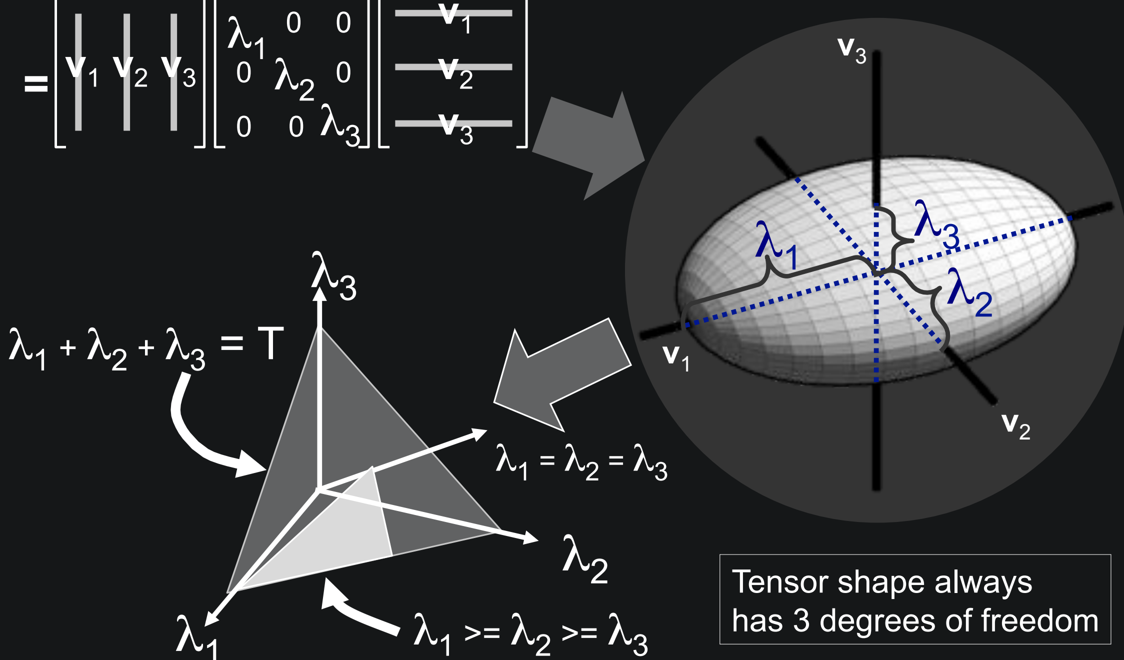
Outline

<u>Visualization</u>:

- DWI Data source
- **Space of Tensor Shape**
- Volume Rendering
- Superquadric Glyphs
- Glyph packing

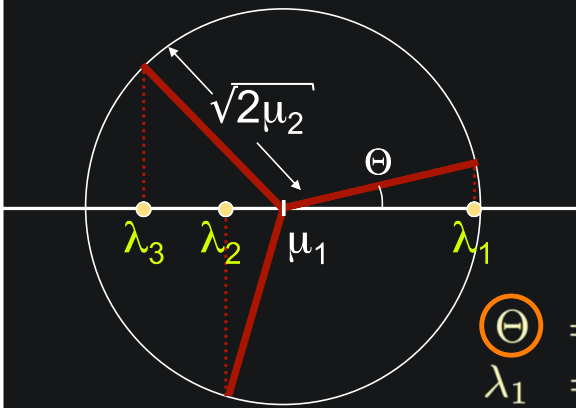
Eigenvalues == Shape

$$\mathbf{D} = \mathbf{R}\Lambda\mathbf{R}^{-1} = \begin{bmatrix} | & | & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} - \mathbf{v}_1 - \\ - \mathbf{v}_2 - \\ - \mathbf{v}_3 - \end{bmatrix}$$



Tensor shape always has 3 degrees of freedom

Cardano's Formula



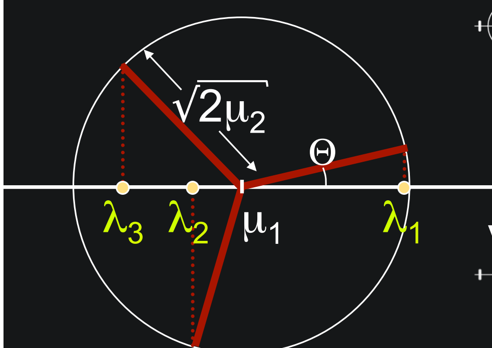
Nickalls, 1993

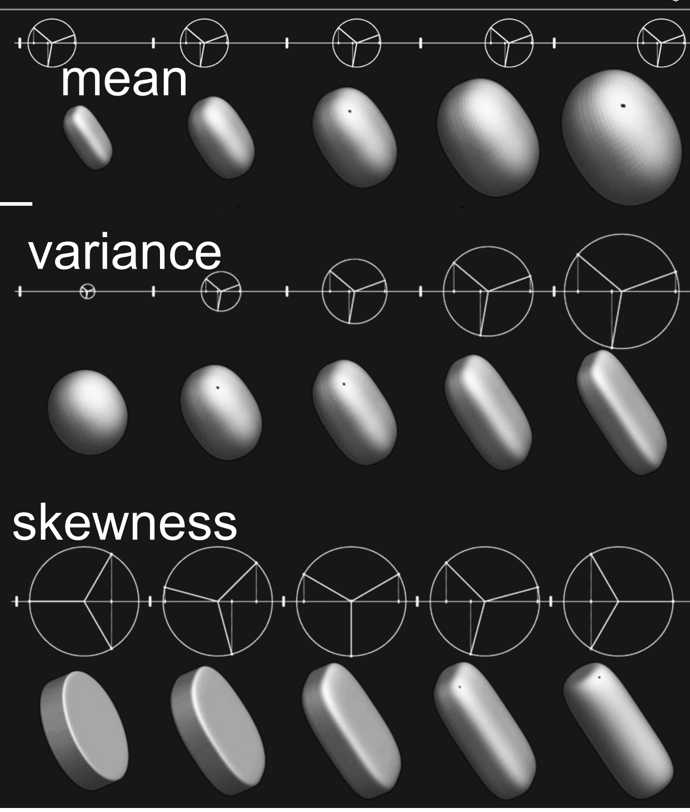
Tensor eigenvalues computed as solutions of a cubic polynomial:

$$
\begin{aligned}
\Theta &= \cos^{-1}(\sqrt{2}\alpha_3)/3 \\
\lambda_1 &= \mu_1 + \sqrt{2\mu_2}\cos(\Theta) \\
\lambda_2 &= \mu_1 + \sqrt{2\mu_2}\cos(\Theta - 2\pi/3) \\
\lambda_3 &= \mu_1 + \sqrt{2\mu_2}\cos(\Theta + 2\pi/3)
\end{aligned}
$$

$$
\begin{aligned}
\mu_1 &: \text{mean}(\lambda_1, \lambda_2, \lambda_3) \\
\mu_2 &: \text{variance}(\lambda_1, \lambda_2, \lambda_3) \\
\alpha_3 &: \text{skewness}(\lambda_1, \lambda_2, \lambda_3)
\end{aligned}
$$

Eigenvalue wheel

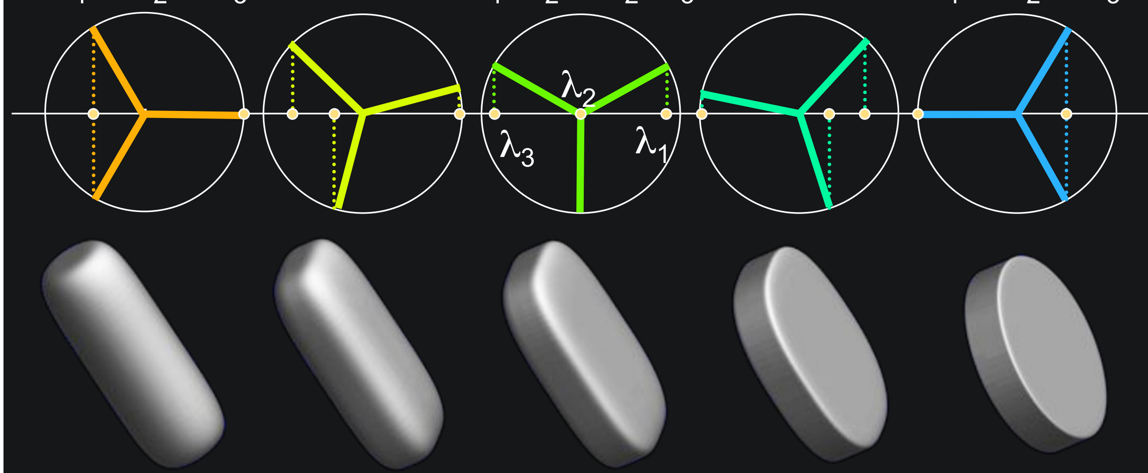




Eigenvalue “sorting”, 2nd order isotropy

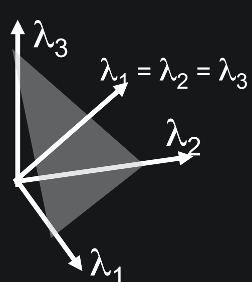
$\Theta = \cos^{-1}(\sqrt{2}\ \text{skew})/3 \Rightarrow \Theta \in [0, \pi/3] \Rightarrow \lambda_1 \geq \lambda_2 \geq \lambda_3$

linear	“orthotropic”	planar
$\Theta = 0$	$\Theta = \pi/6$	$\Theta = \pi/3$
skew = $-1/\sqrt{2}$	skew = 0	skew = $1/\sqrt{2}$
$\lambda_1 > \lambda_2 = \lambda_3$	$\lambda_1 - \lambda_2 = \lambda_2 - \lambda_3$	$\lambda_1 = \lambda_2 > \lambda_3$



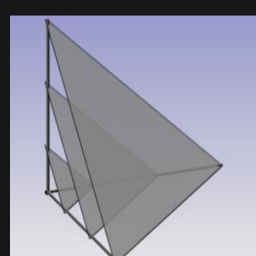
Tensor invariants as orthogonal shape parameterizations

Cylindrical or spherical coordinates (Ennis & Kindlmann 2005)

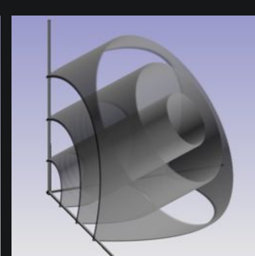


$\text{tr}(\mathbf{D})$ = Dxx+Dyy+Dzz
$|\mathbf{D}| = \text{sqrt}(\text{tr}(\mathbf{D}^T\mathbf{D}))$

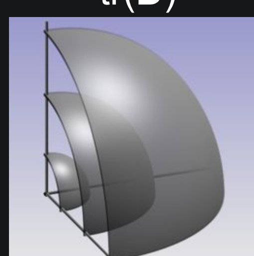
$\mathbf{E}$ = deviatoric($\mathbf{D}$)
= $\mathbf{D}$ - trace($\mathbf{D}$)*$\mathbf{I}$/3



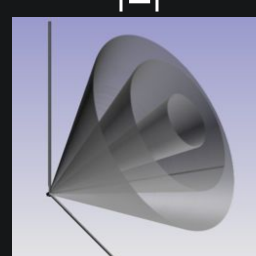
tr(**D**)



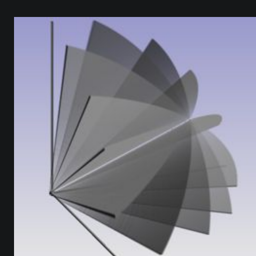
|**E**|



|**D**|

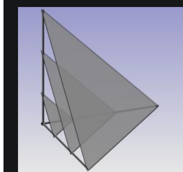


|**E**|/|**D**| ~= FA
FA = **Fractional**
…… **Anisotropy**



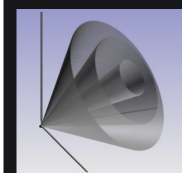
mode(**E**)
= det(**E**/|**E**|)
(Criscione ‘00)
Mode measures
Linear vs. planar
anisotropy

Biological Meaning of Tensor Shape

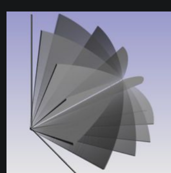


Size: **bulk mean diffusivity** (“ADC”)

- ADC strictly speaking diffusivity along **one** direction
- Note: same across gray+white matter, high in CSF
- Indicator of acute ischemic stroke

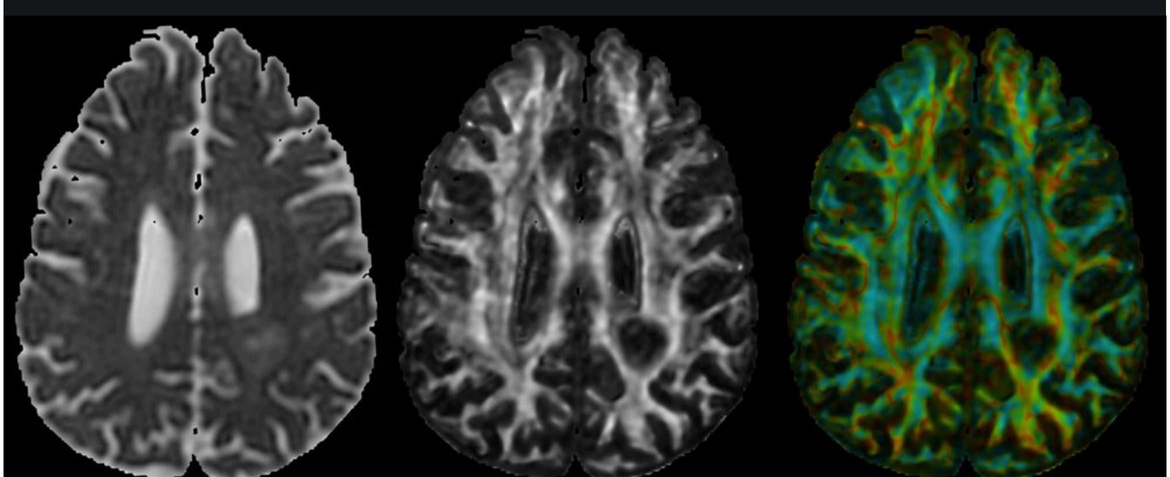


- Anisotropy (e.g. FA): directional microstructure
  - High in white matter, low in gray matter and CSF
  - Increases with myelination, decreases in some diseases (Multiple Sclerosis)



- Mode: linear versus planar
  - Partial voluming of adjacent orthogonal structures
  - Fine-scale mixing of diverse fiber directions
  - Tensor fitting error increases with planarity (Tuch 2002)

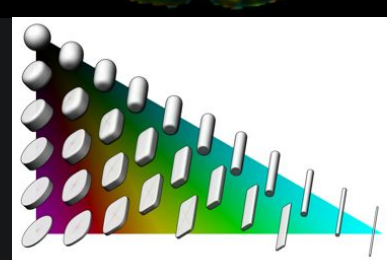
Tensor shape on one slice



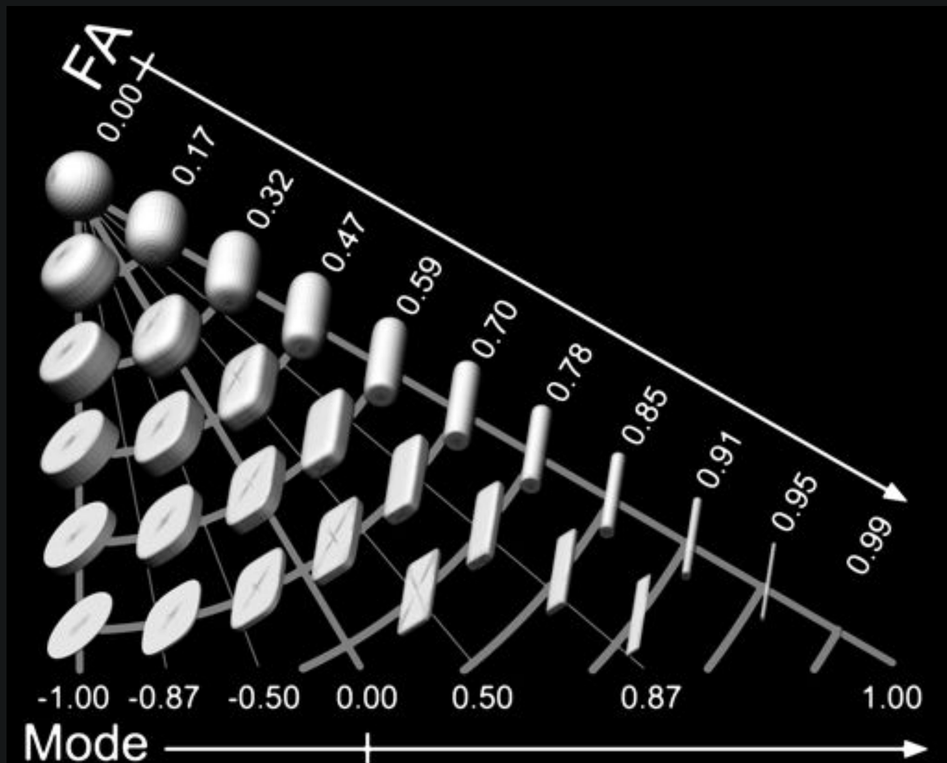
Trace

Fractional **A**nisotropy

“Anisotropy” is a bivariate quantity



FA and Mode again



Outline

<u>Visualization</u>:

DWI Data source

Space of Tensor Shape

Volume Rendering

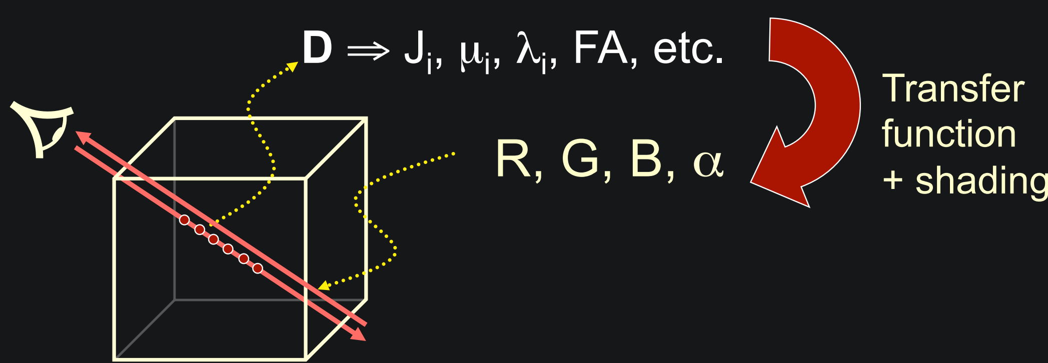
Superquadric Glyphs

Glyph packing

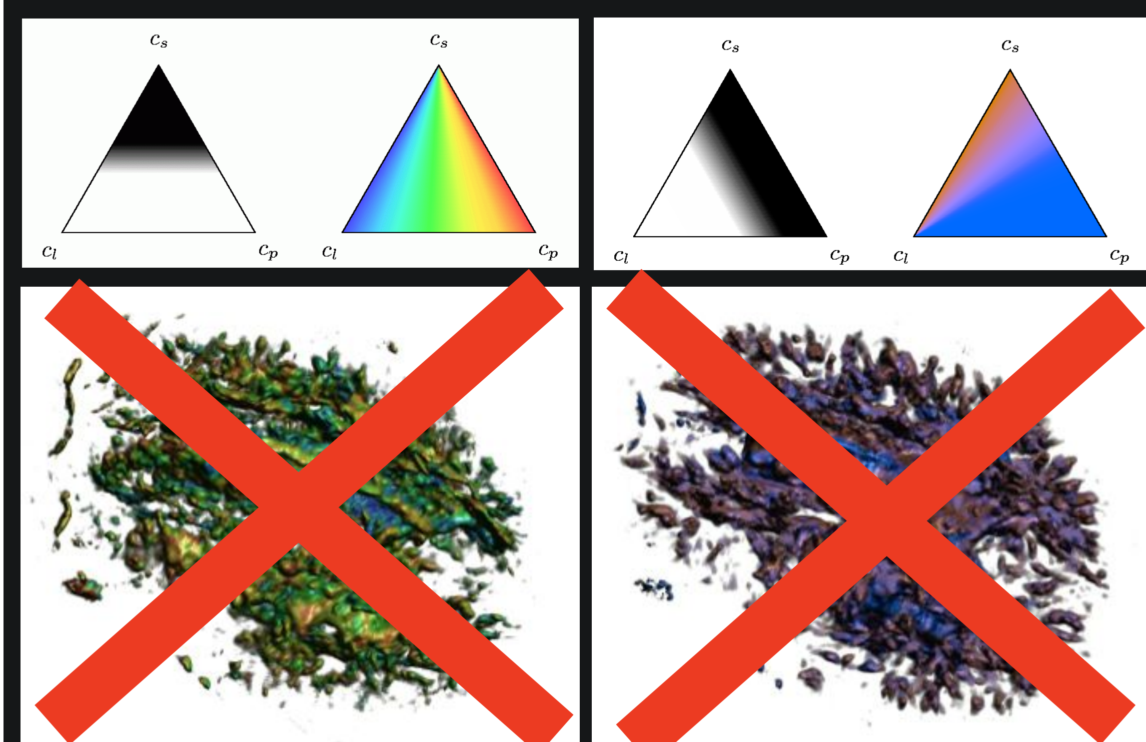
Direct Volume Rendering

Simple algorithm

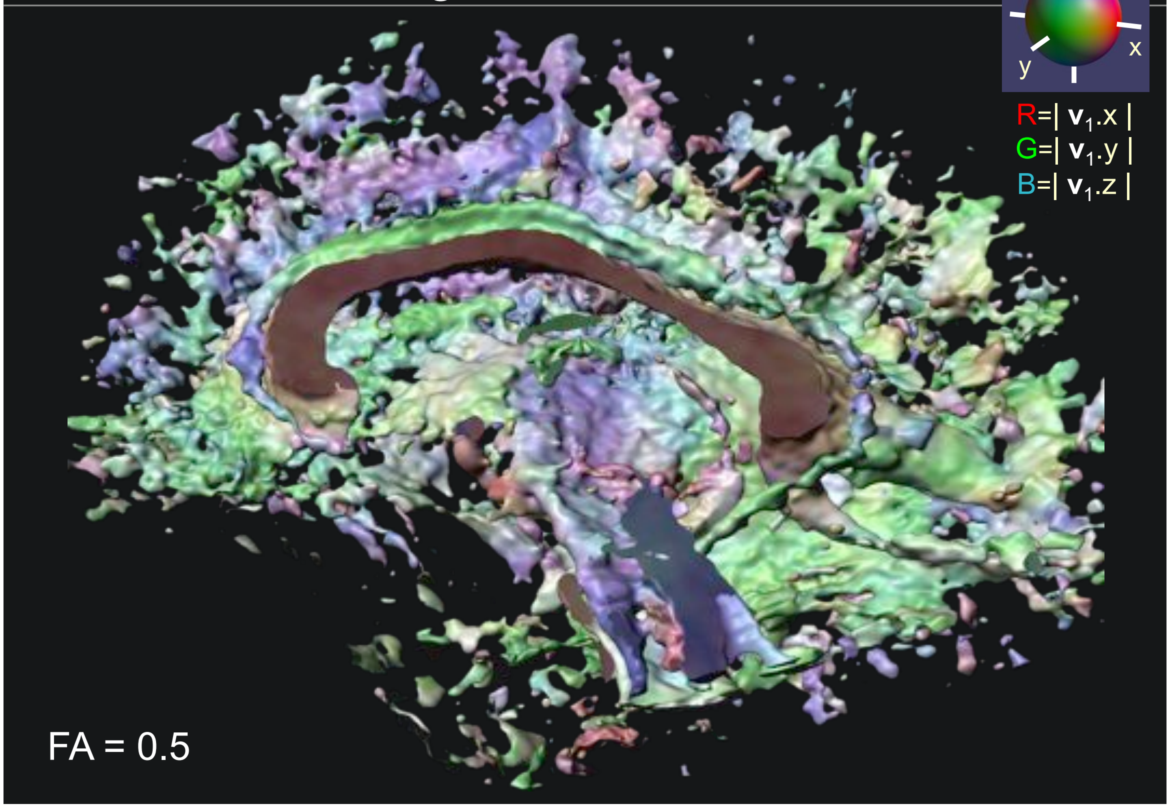
- Cast rays through volume
- Measure tensor, tensor properties
- Assign colors and opacities
- Modulate colors with shading
- Composite with "over" operator



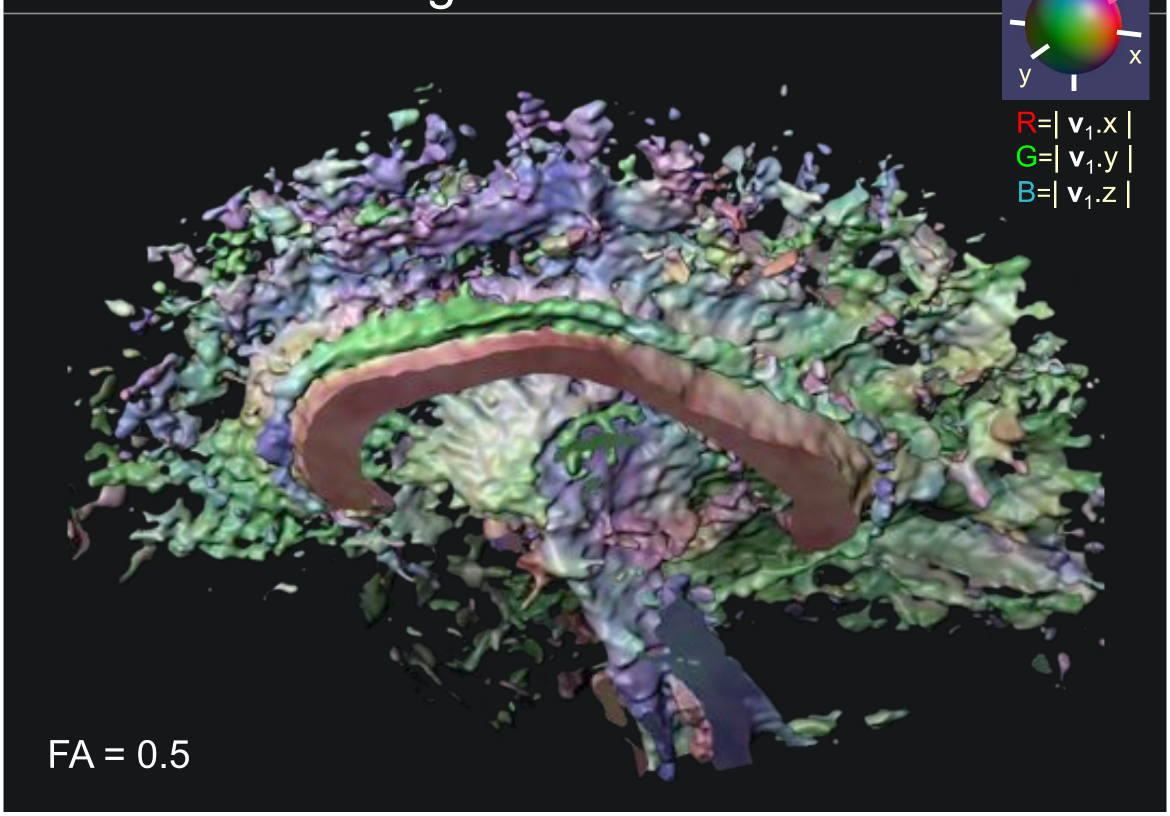
Barycentric color maps: results

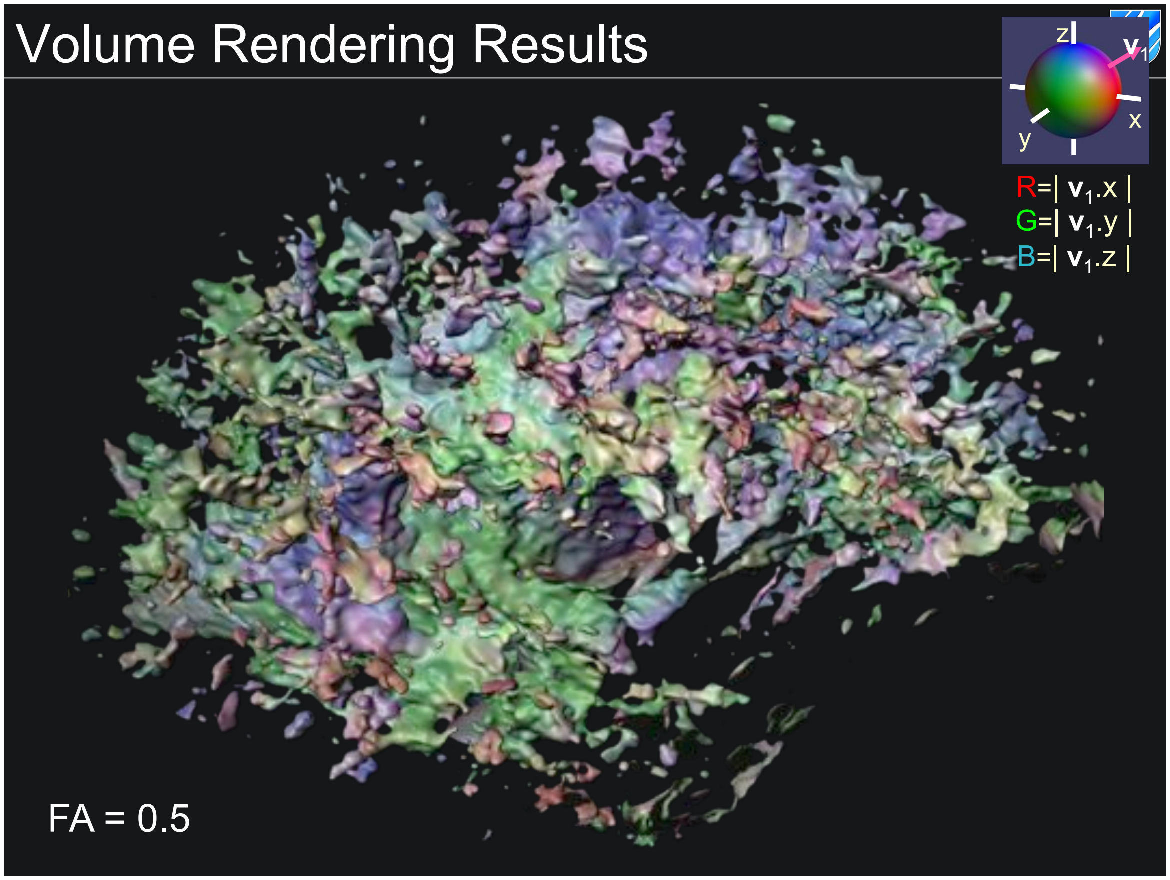


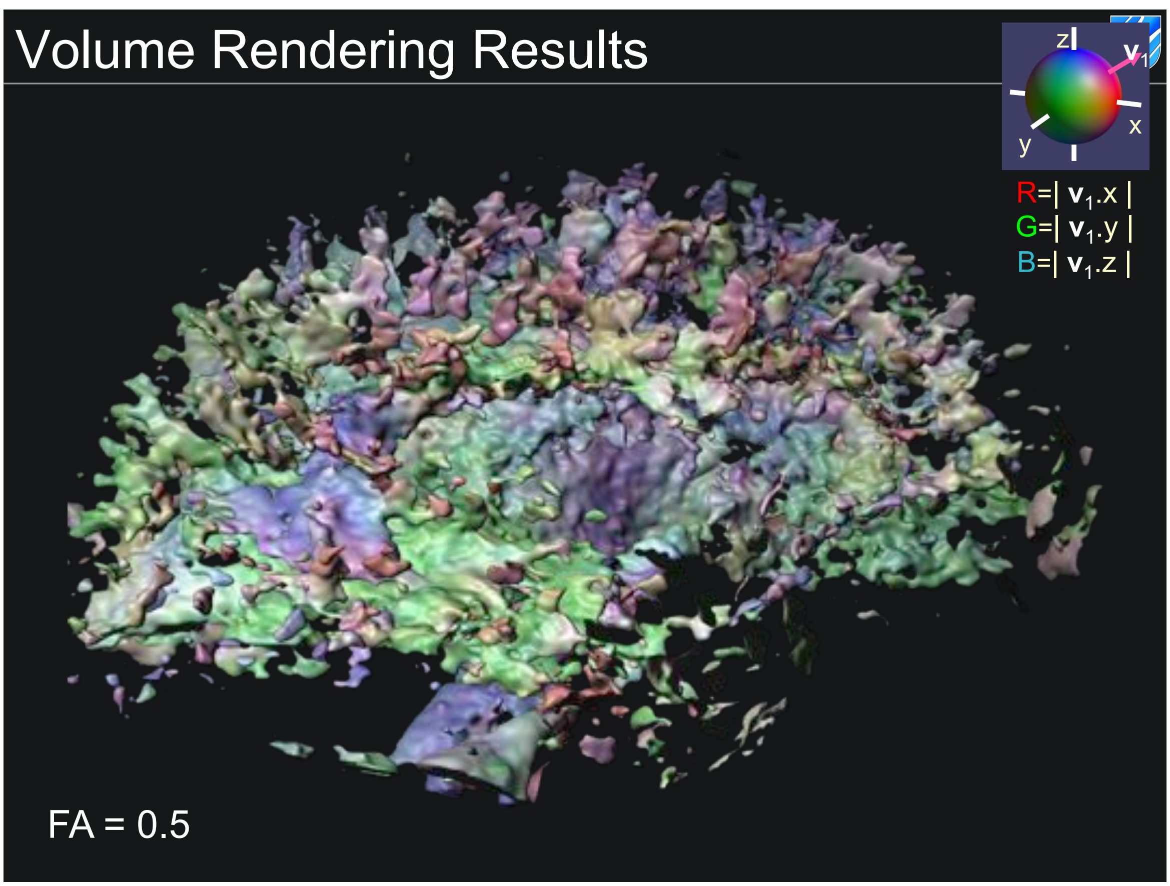
Volume Rendering Results



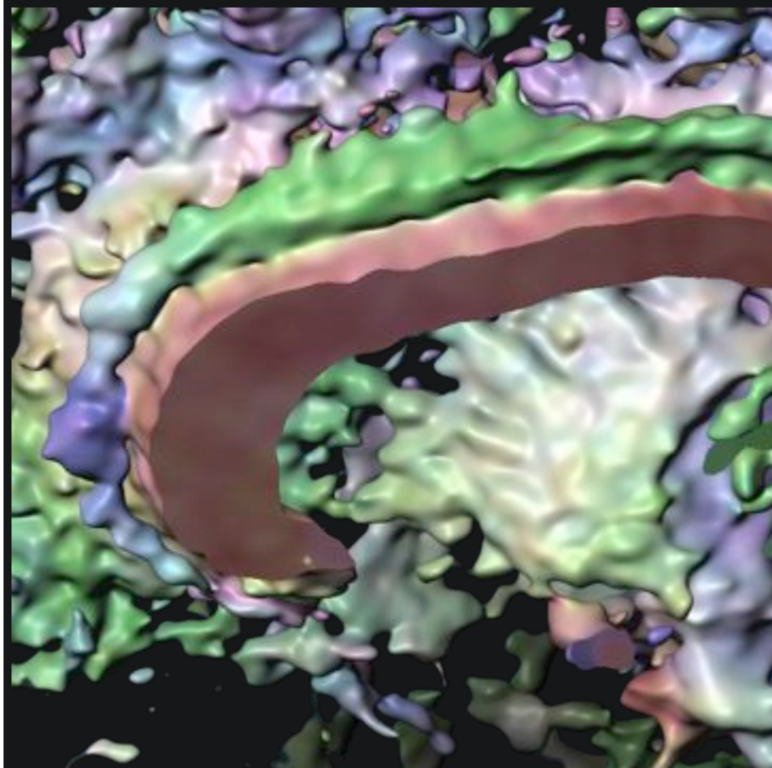
Volume Rendering Results



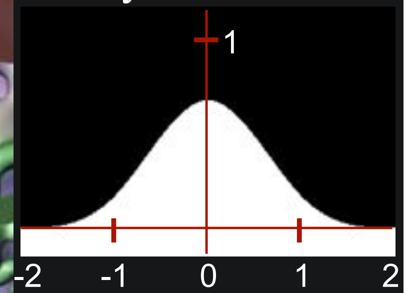

Volume Rendering Results
z
v_1
x
y
R=| v_1.x |
G=| v_1.y |
B=| v_1.z |
FA = 0.5


Volume Rendering Results
z
v_1
x
y
R=| v_1.x |
G=| v_1.y |
B=| v_1.z |
FA = 0.5

Visualizing kernel differences

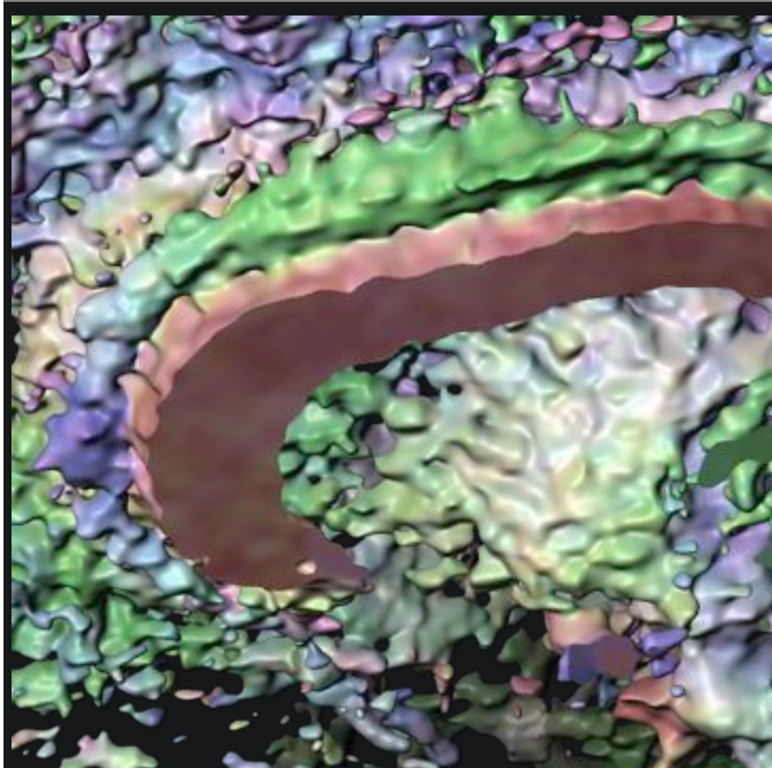


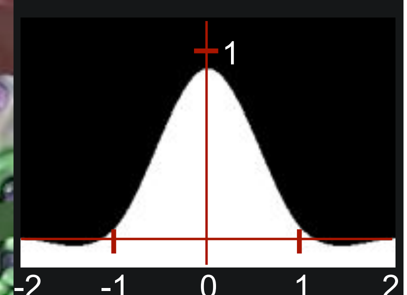
Mitchell-Netravali BC-splines: Simple, tunable, always C^1



(B,C) = (1,0)
Uniform cubic B-spline, also C^2

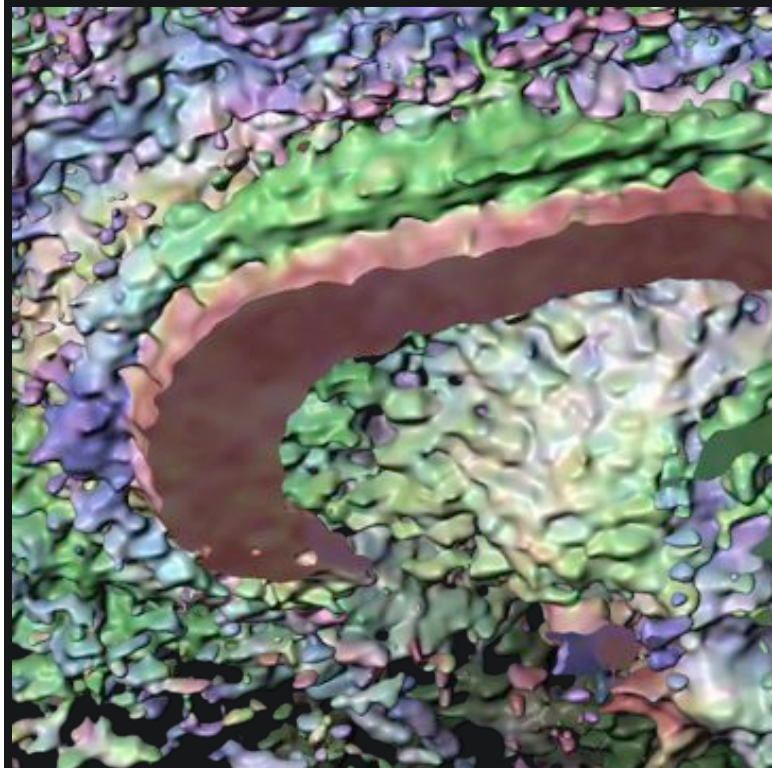
Visualizing kernel differences

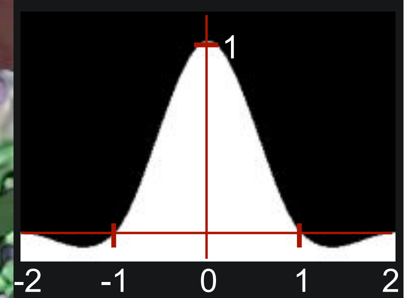




(B,C) = (1/3,1/3)
Blurs a little

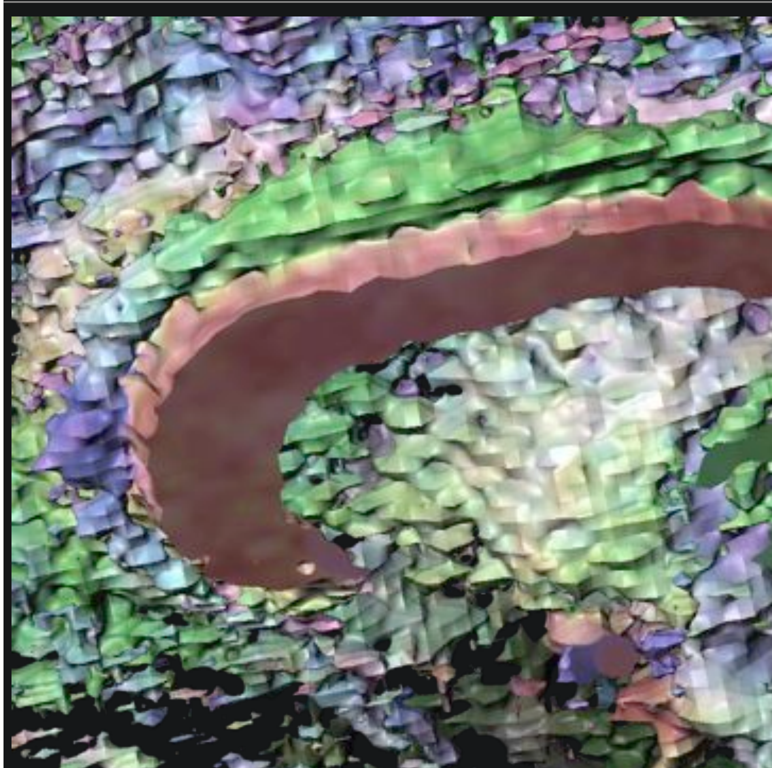
Visualizing kernel differences

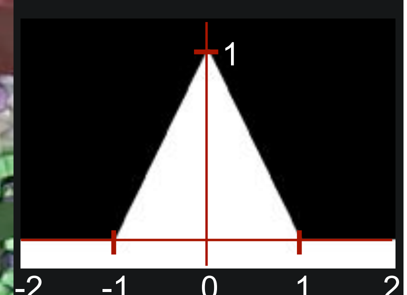




(B,C) = (0,1/2)
Catmull-Rom
Interpolates

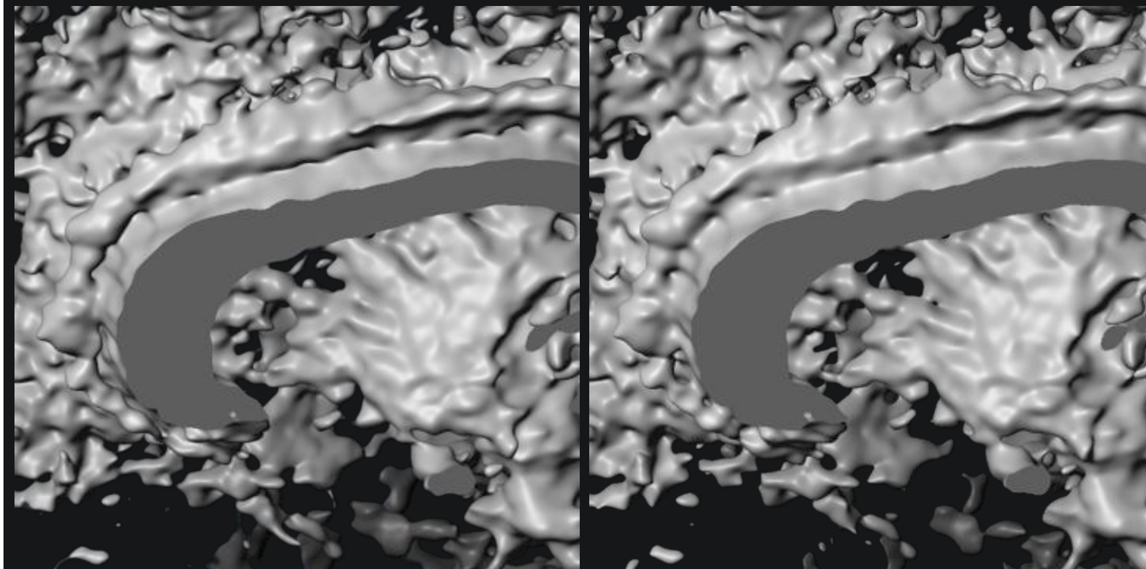
Visualizing kernel differences





Linear : Not C^1
⇒ nasty edges,
but can see
each sample

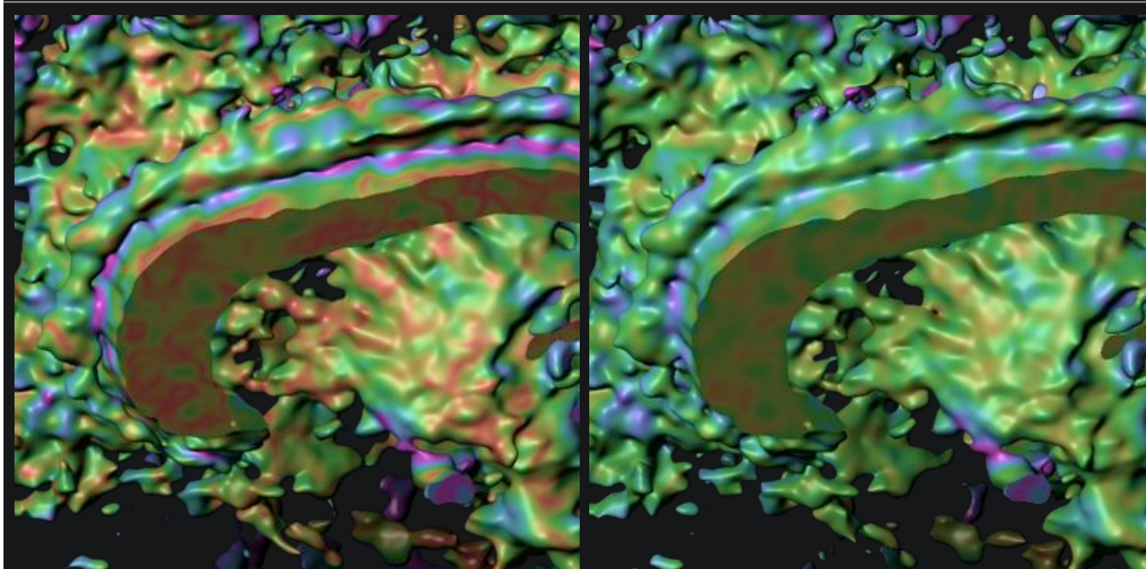
Reconstruction+invariants don’t commute



Reconstruct tensors, then Calculate FA

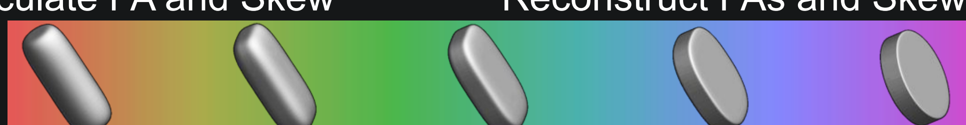
Calculate FA, then Reconstruct FAs

Reconstruction+invariants don’t commute



Reconstruct tensors, then Calculate FA and Skew

Calculate FA and Skew, then Reconstruct FAs and Skews



Outline

Visualization:

DWI Data source

Space of Tensor Shape

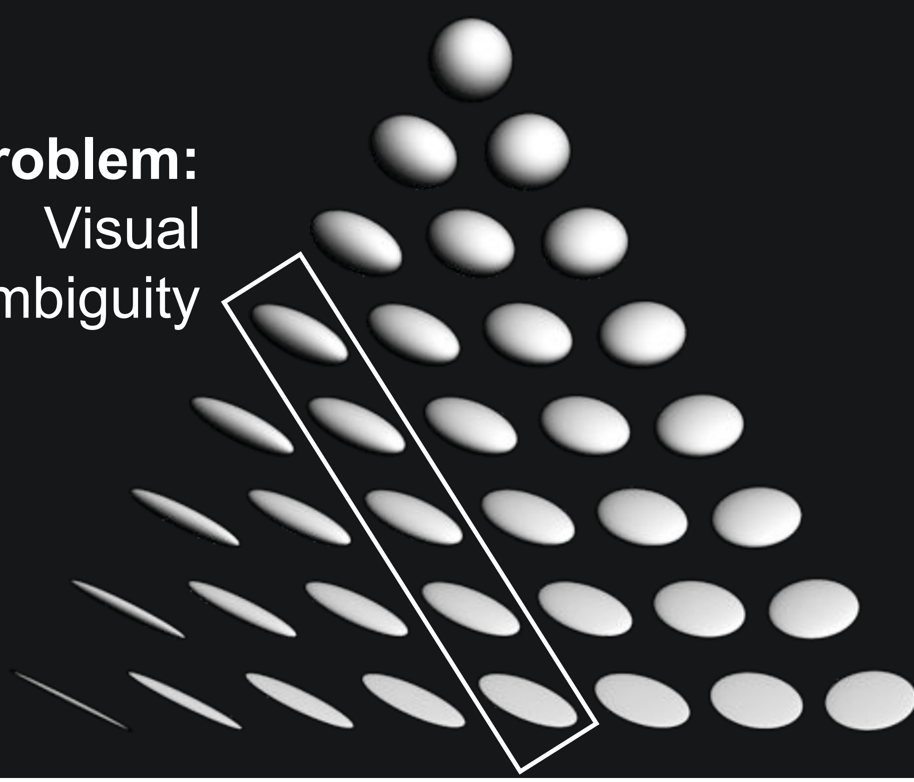
Volume Rendering

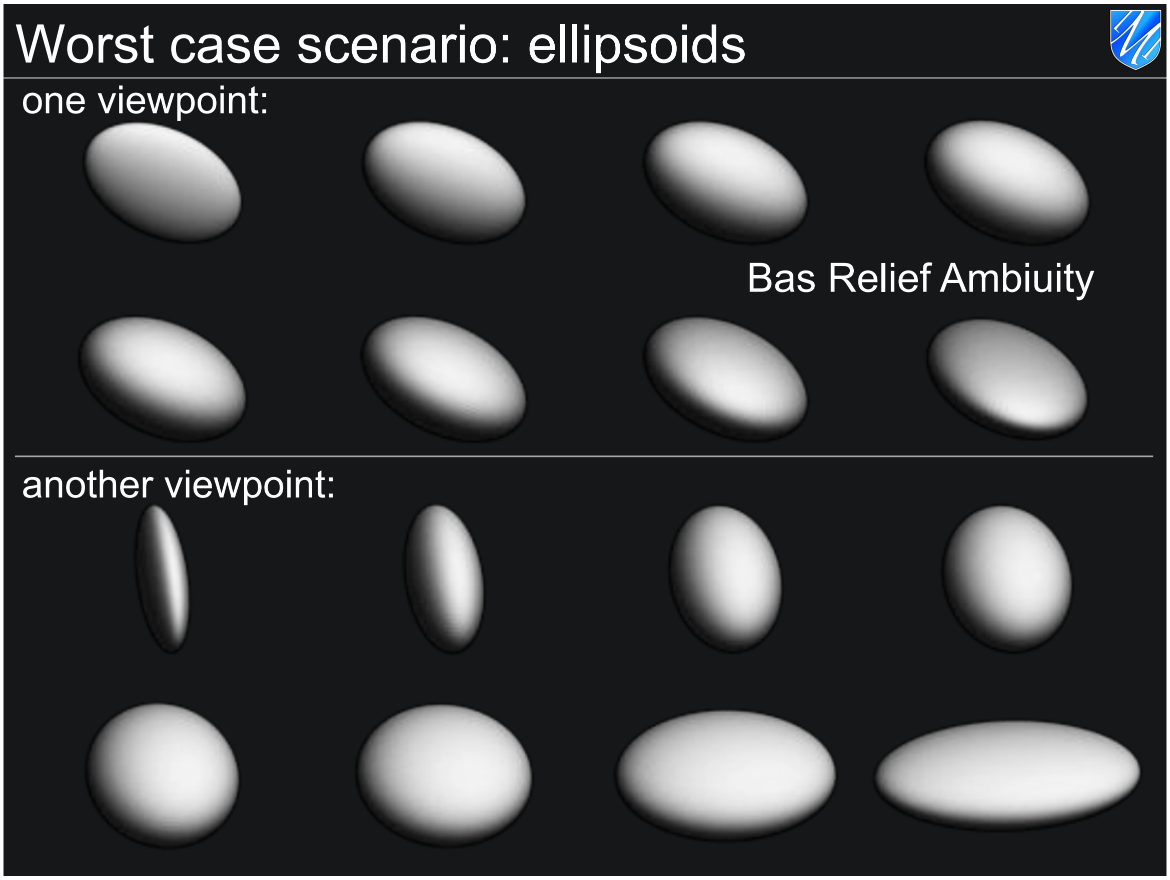
Superquadric Glyphs

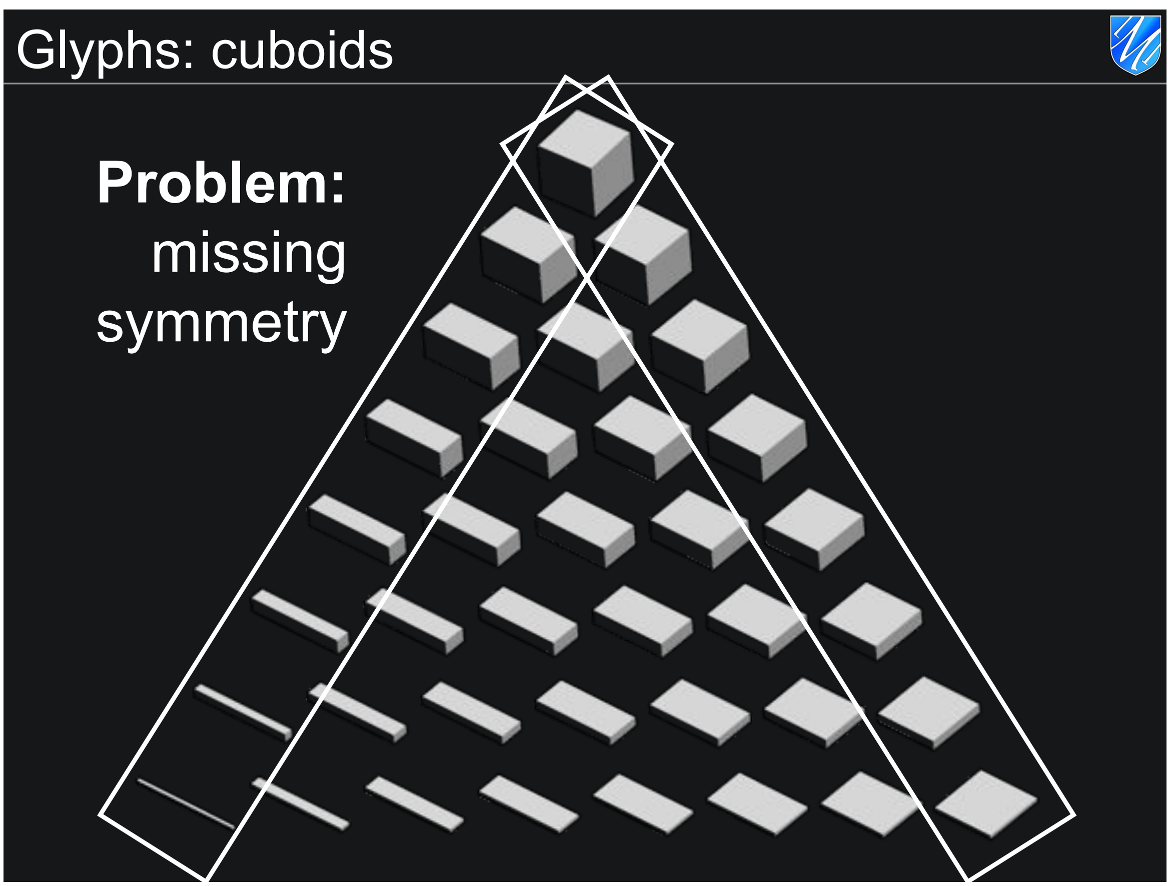
Glyph packing

Glyphs: ellipsoids

Problem:
Visual
ambiguity




Worst case scenario: ellipsoids
one viewpoint:
Bas Relief Ambiuity
another viewpoint:


Glyphs: cuboids
Problem:
missing
symmetry

Eigensystem symmetry

$\lambda_1 = \lambda_2$

$\Downarrow$

2-D eigenspace(λ_1)

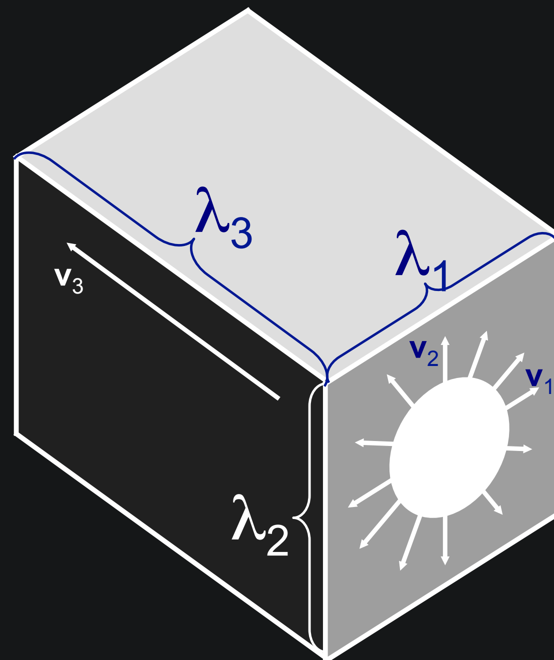
$\Downarrow$

$\mathbf{v_1}$, $\mathbf{v}_2$ not unique

(continuous rotational symmetry)

Conversely,

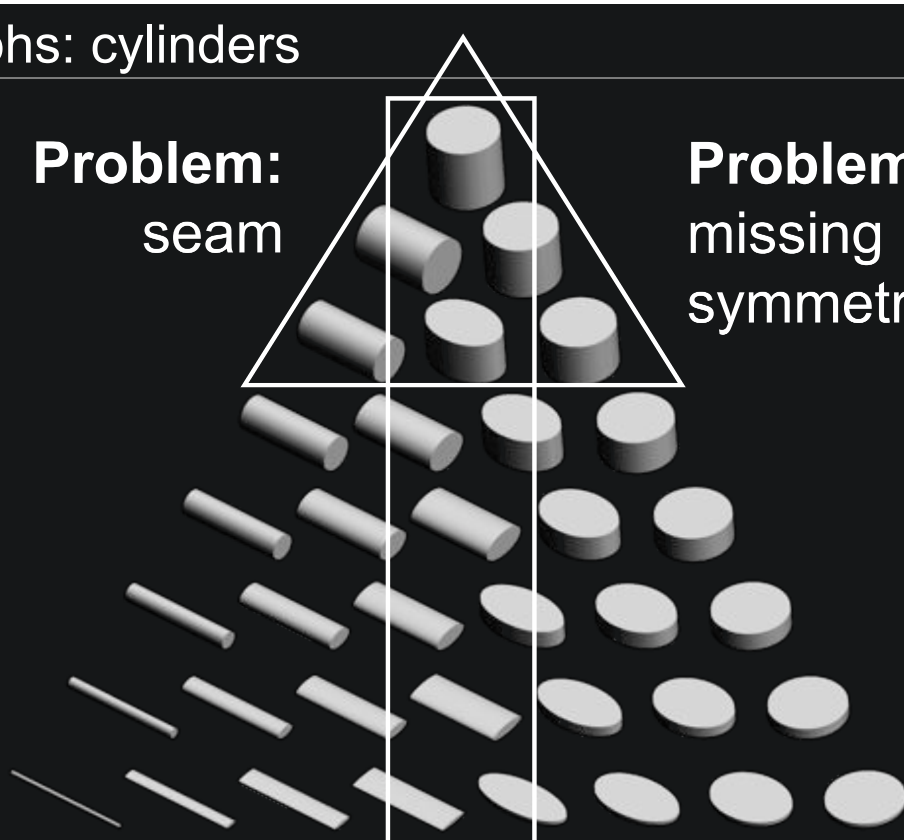
$\mathbf{v_1}$, $\mathbf{v}_3$ distinct



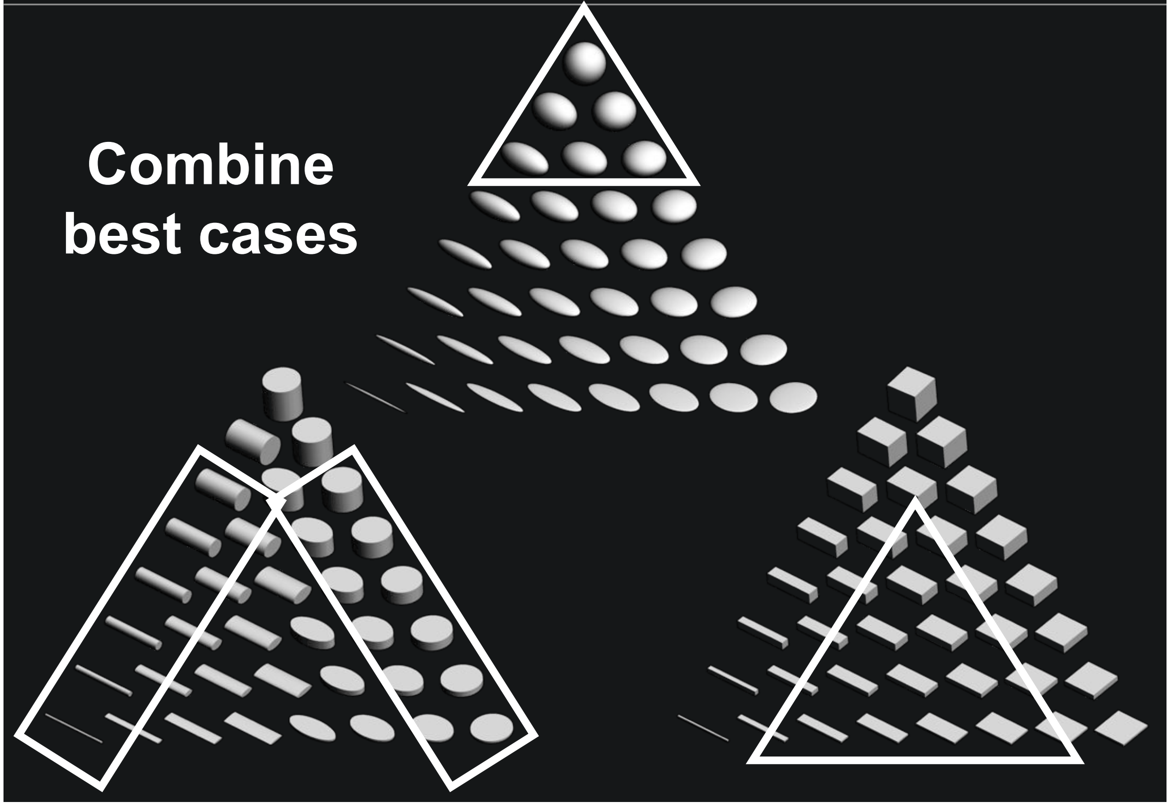
Glyphs: cylinders

Problem:
seam

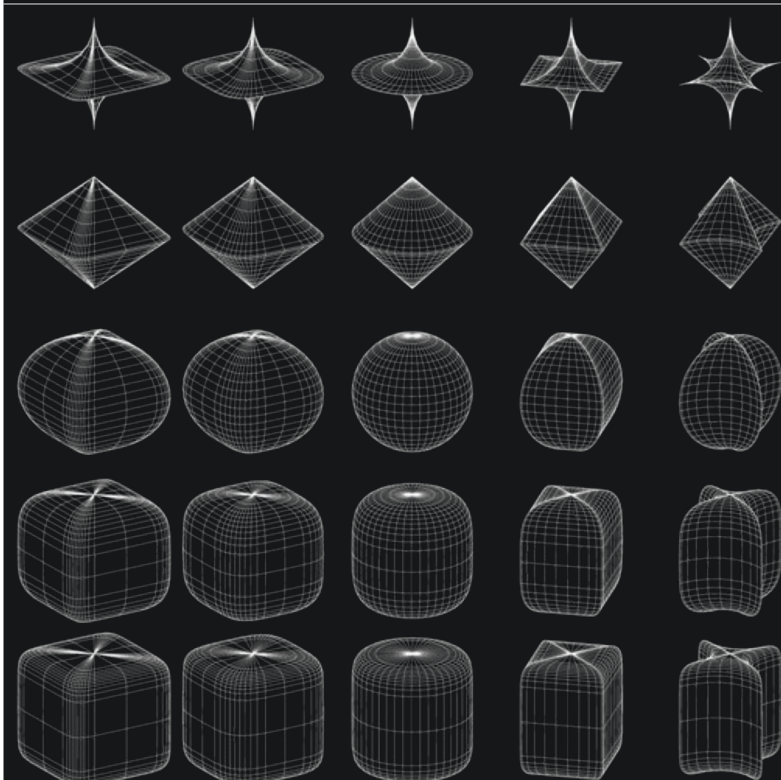
Problem:
missing symmetry



Basic idea:



How: **superquadrics**



Barr, 1981

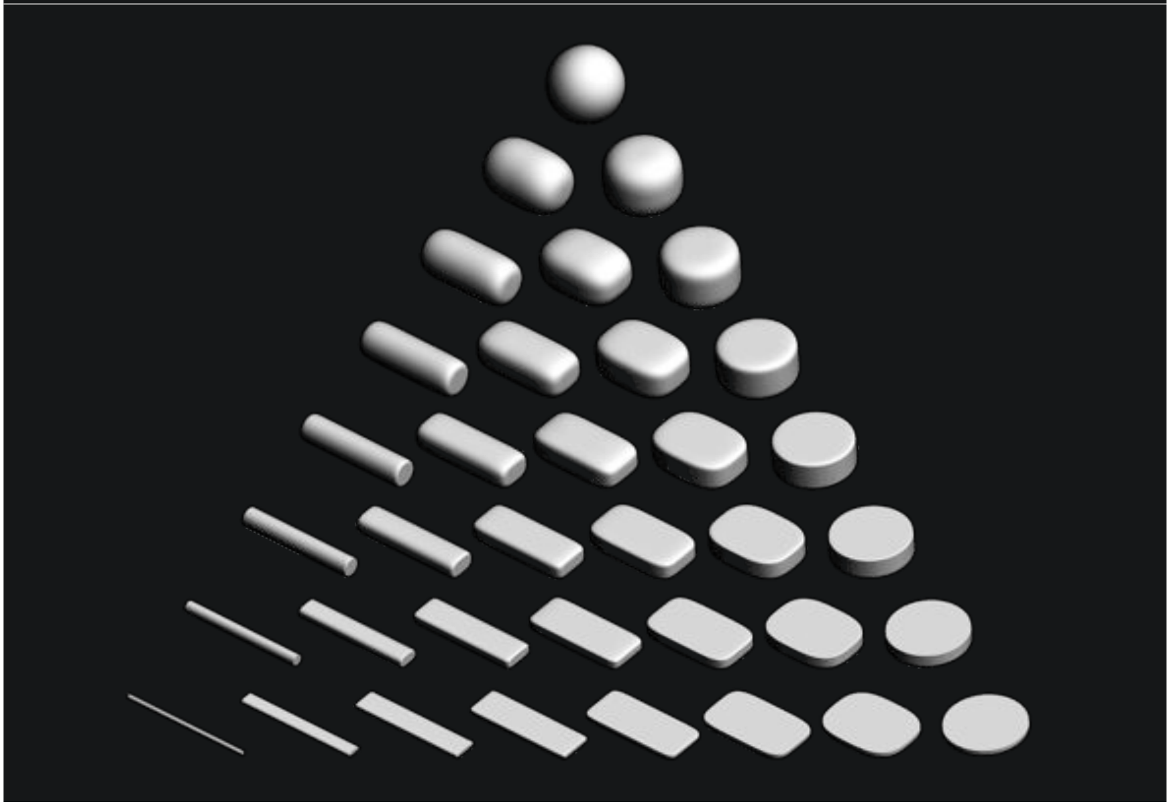
For visualization:
Shaw et al.,
1998, 1999

Ebert at al.,
2000, 2001

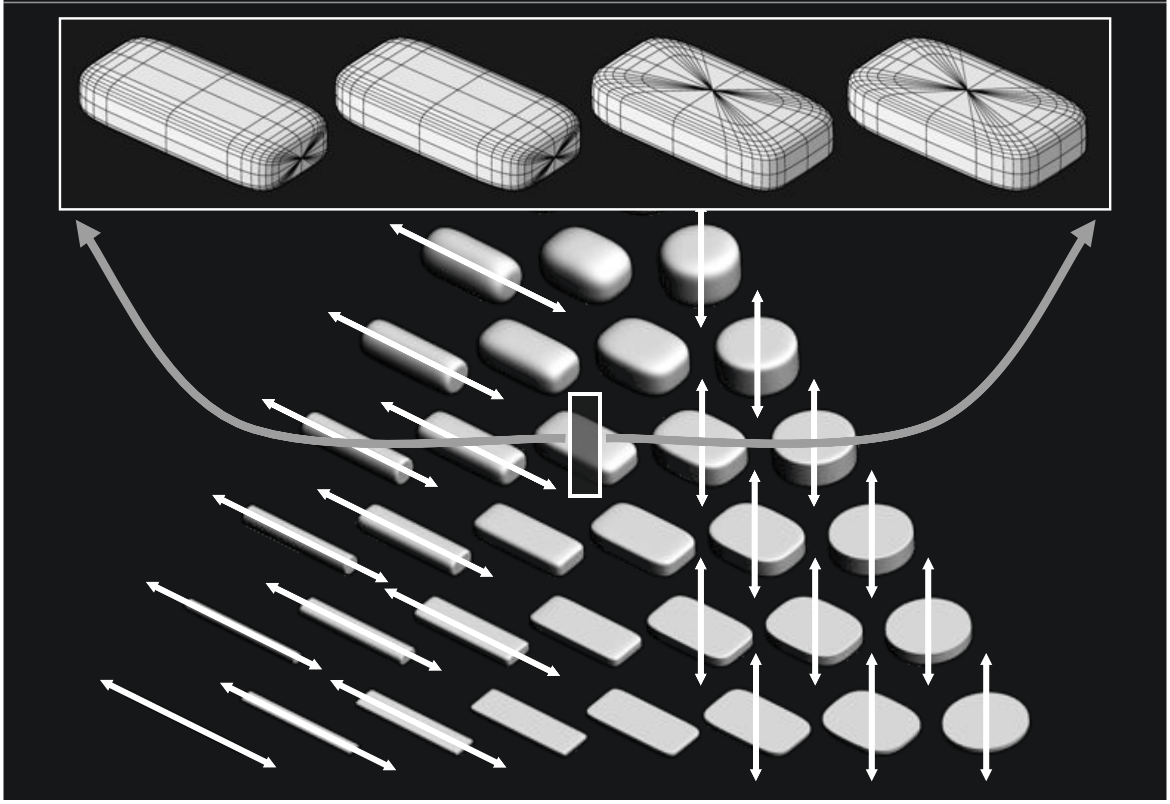
- Variable glyph geometry
- Tensor glyph, not just multi-value

Basic method

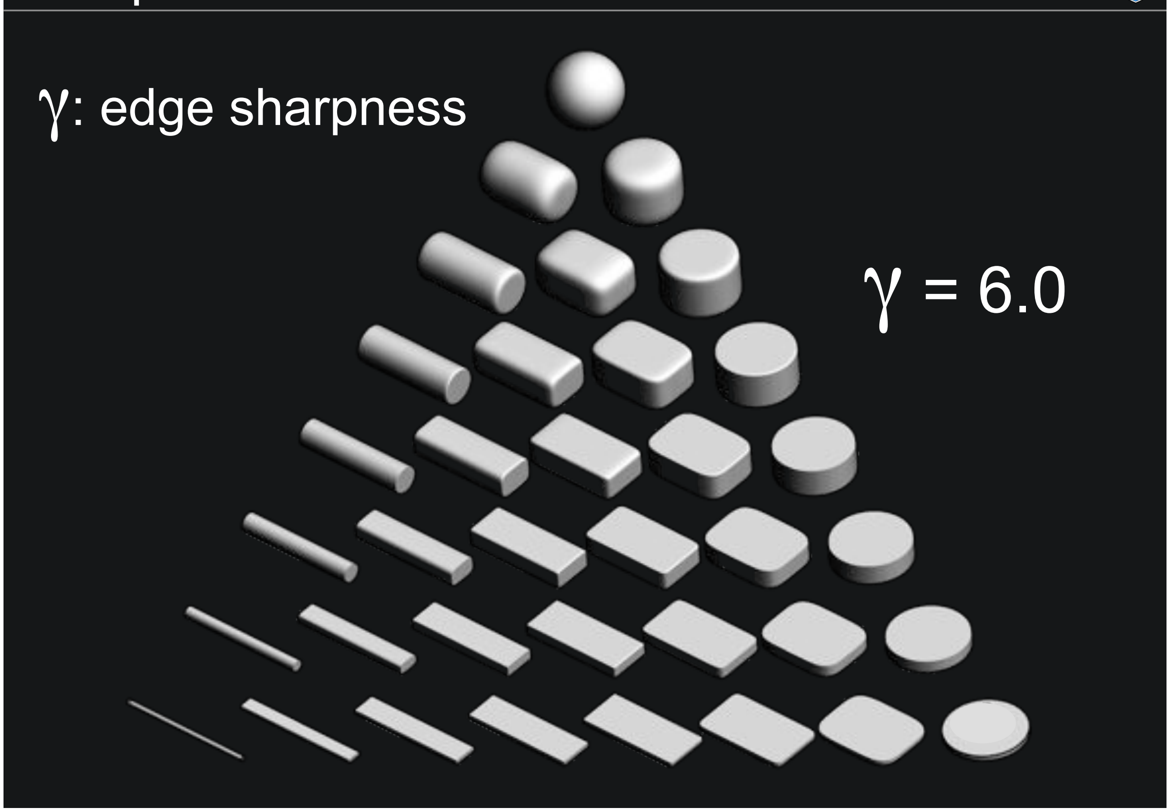
Glyphs: superquadrics



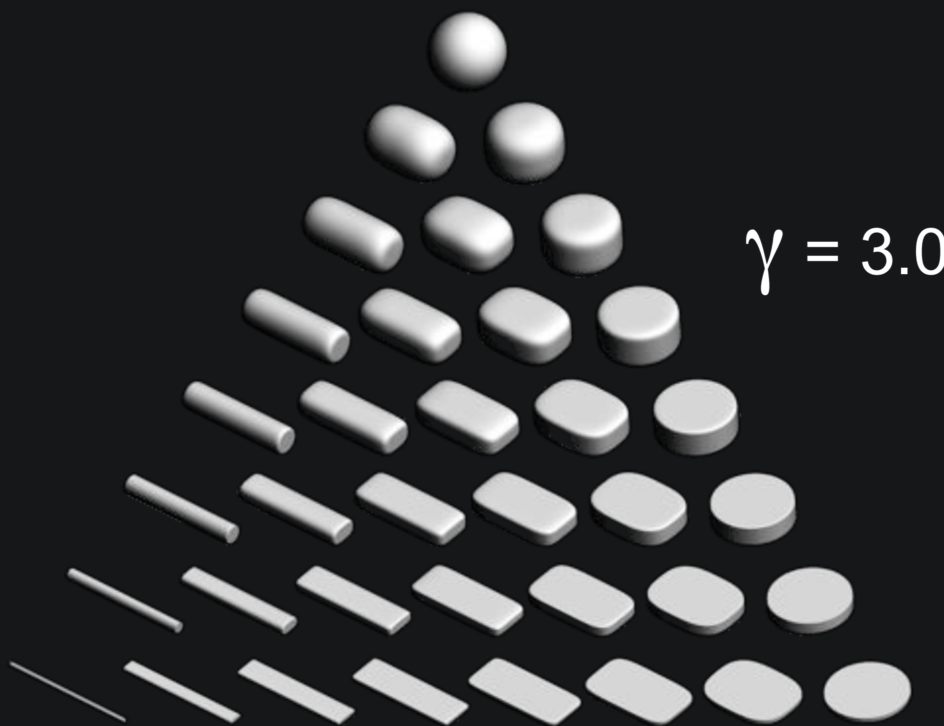
Seam: hidden



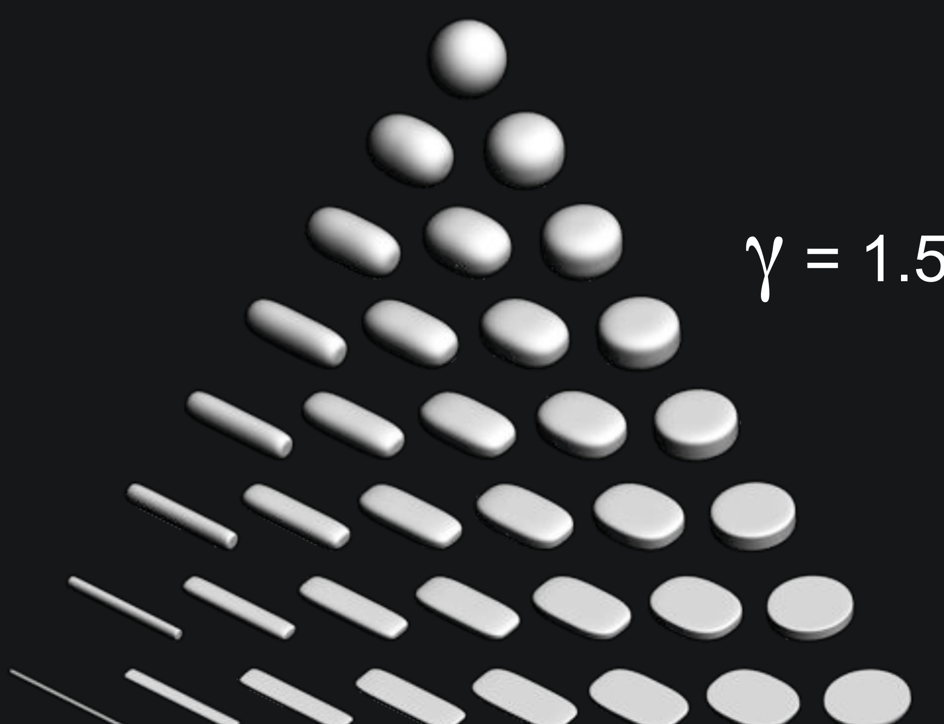
One parameter



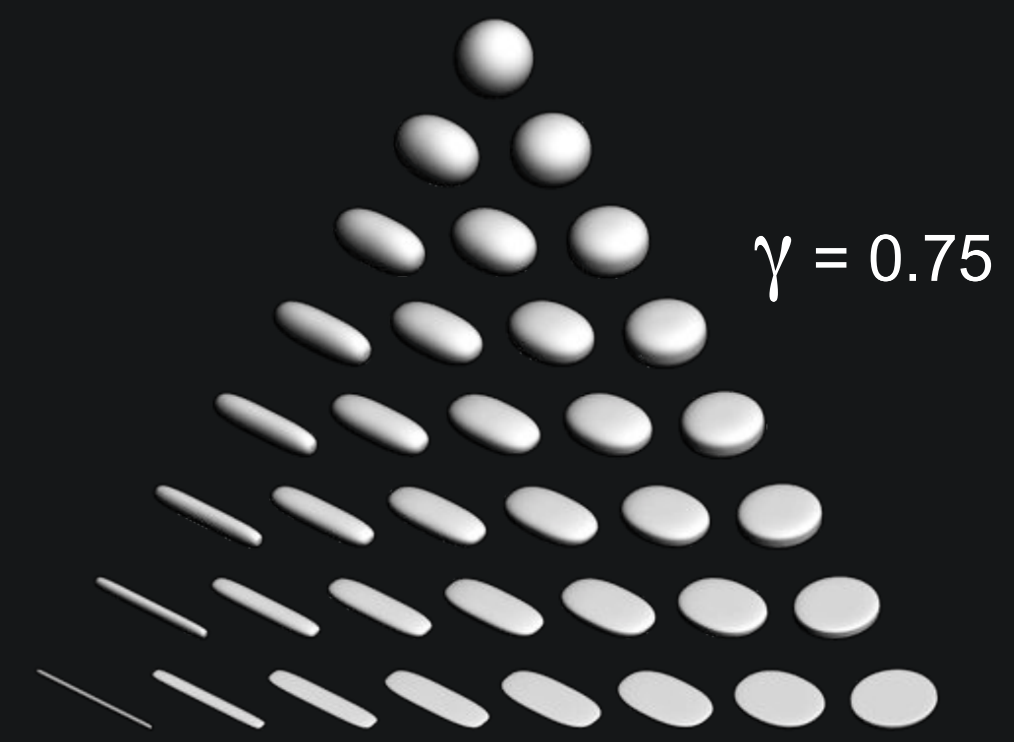
One parameter



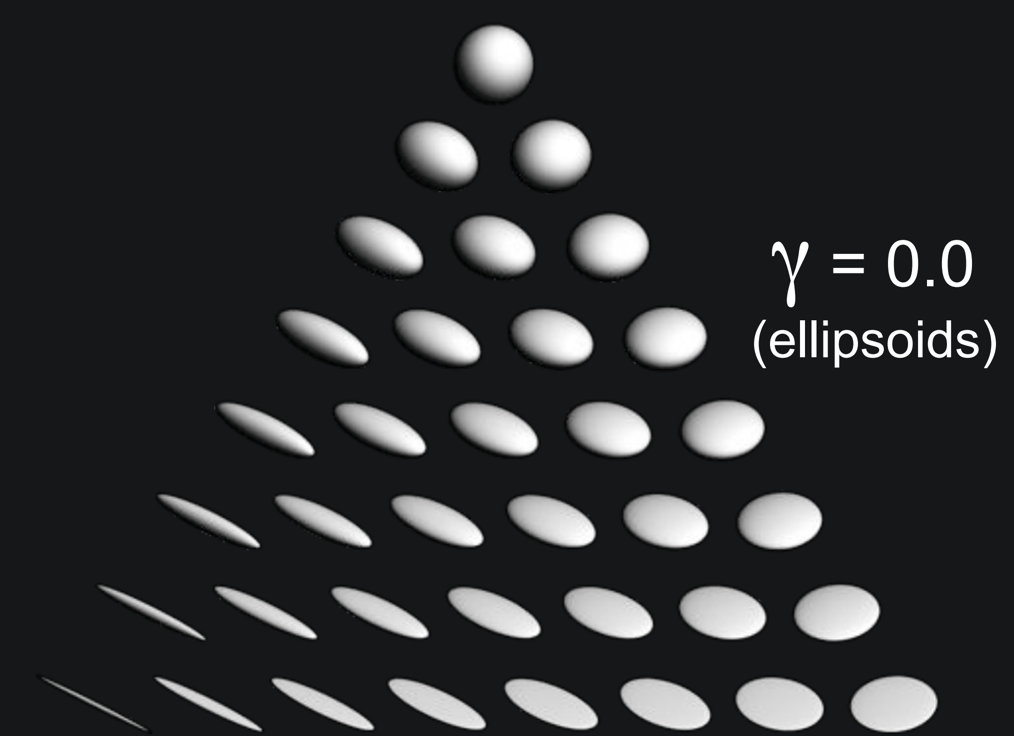
One parameter



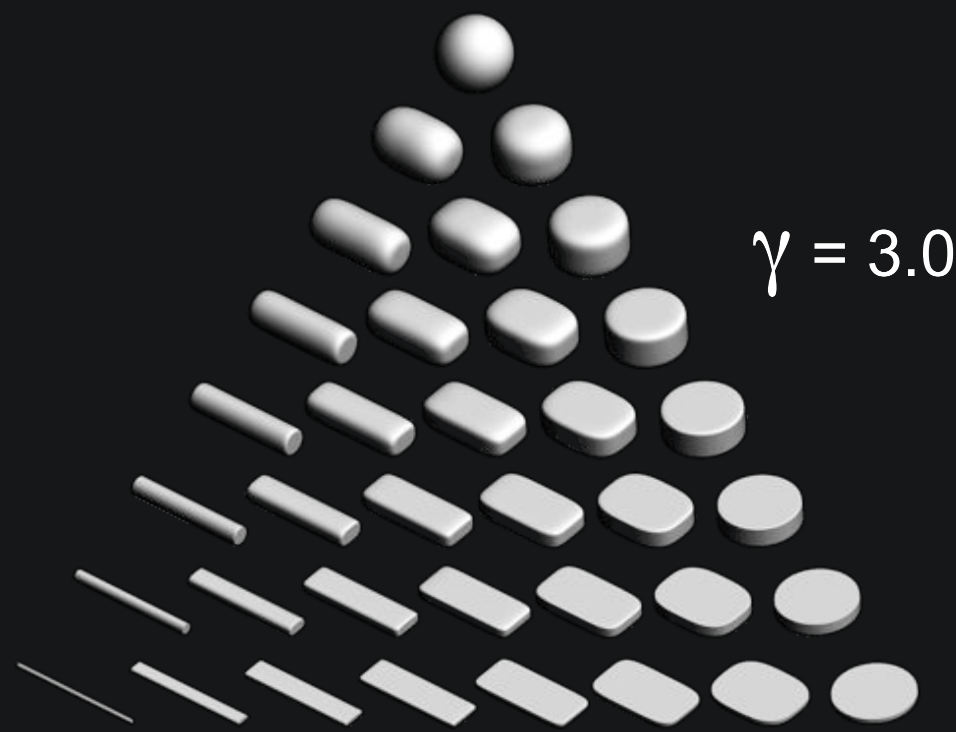
One parameter



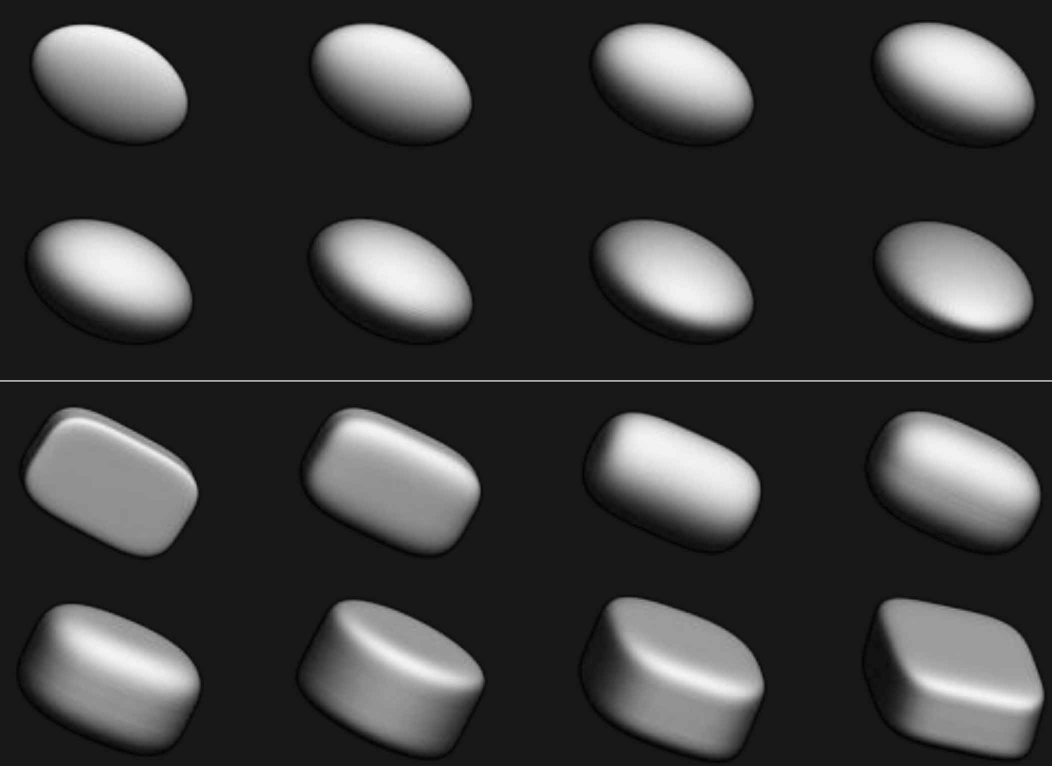
One parameter



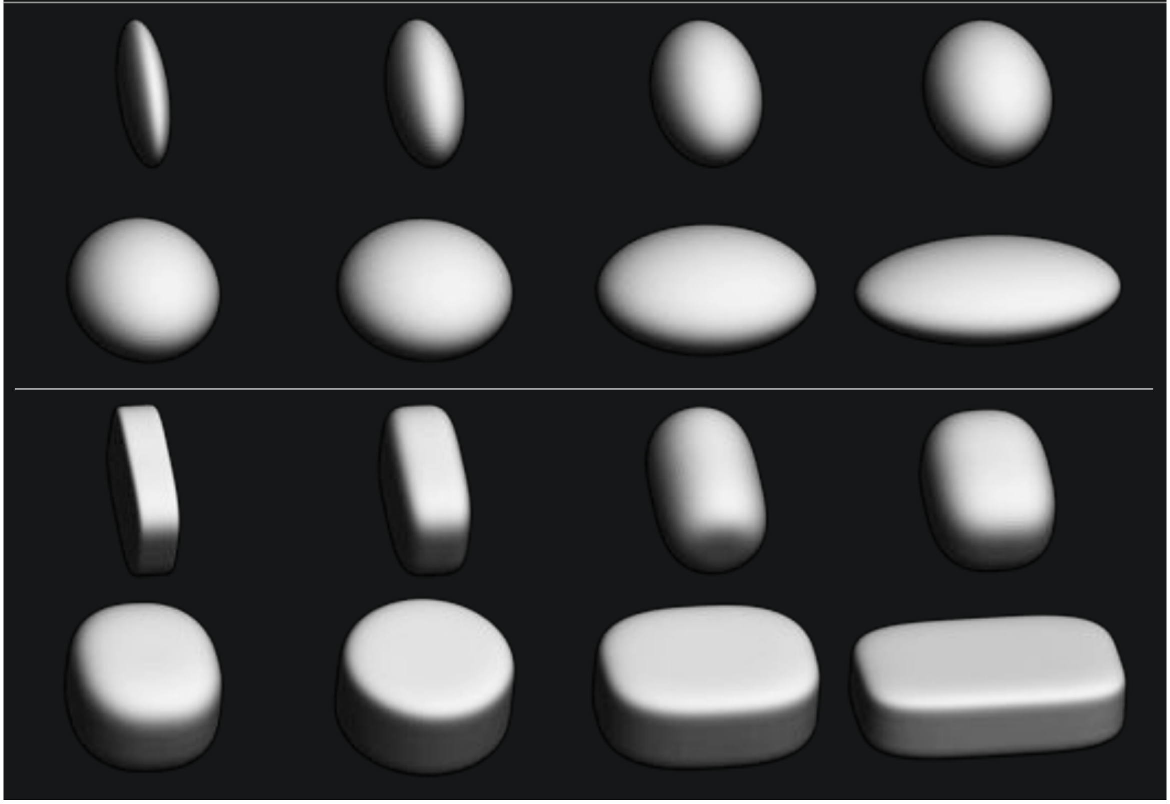
Good default choice

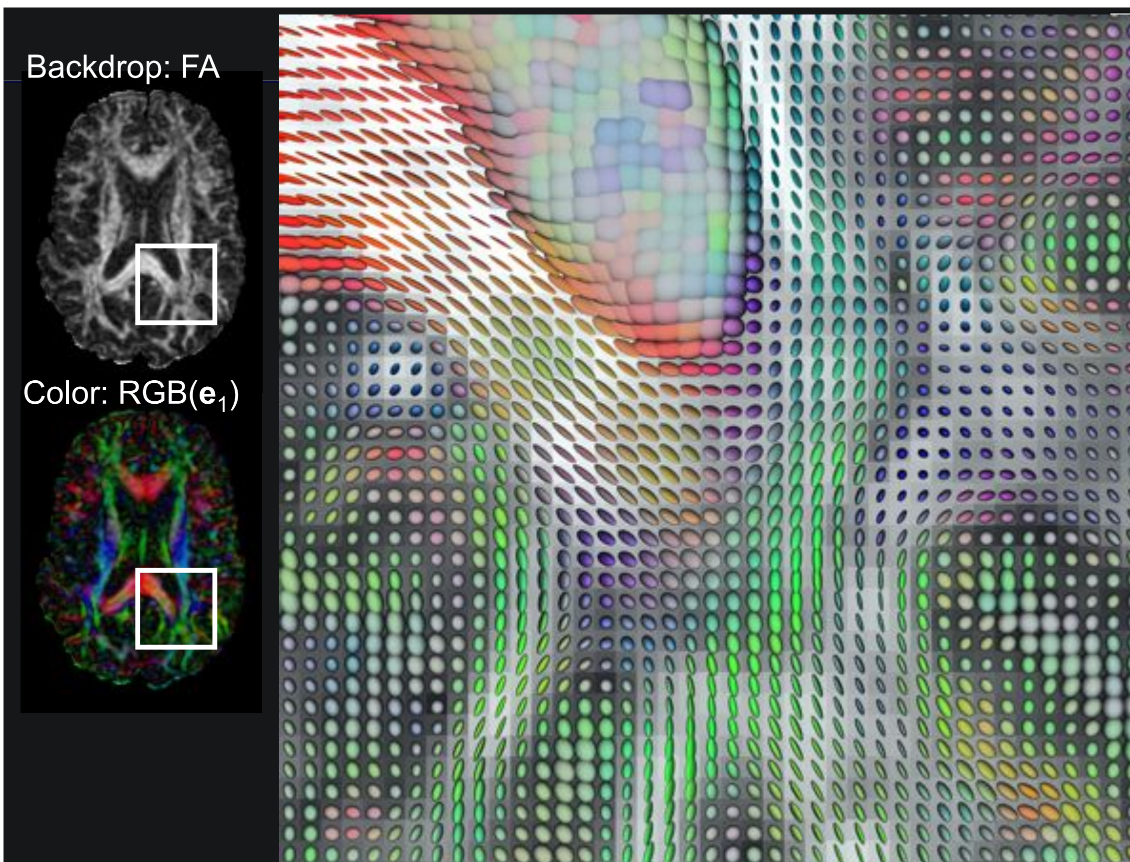


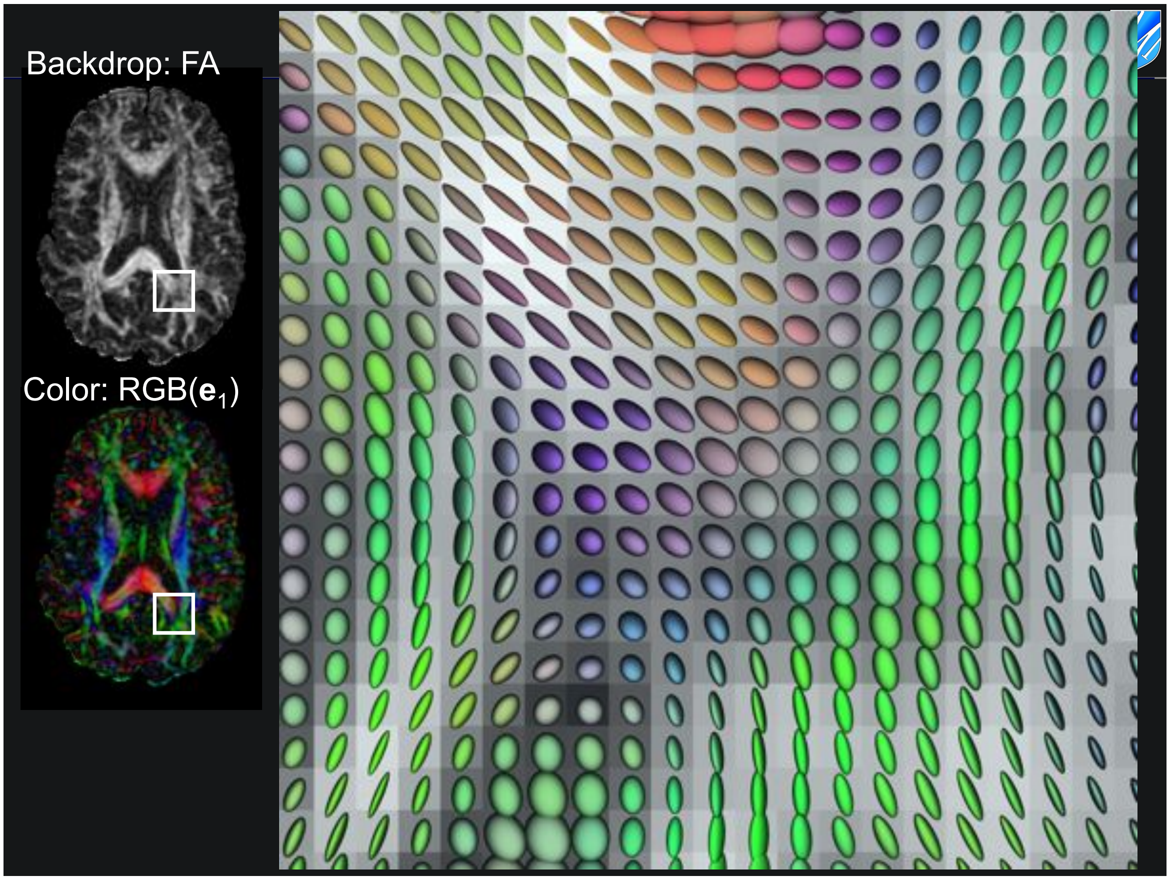
Worst case scenario, revisited

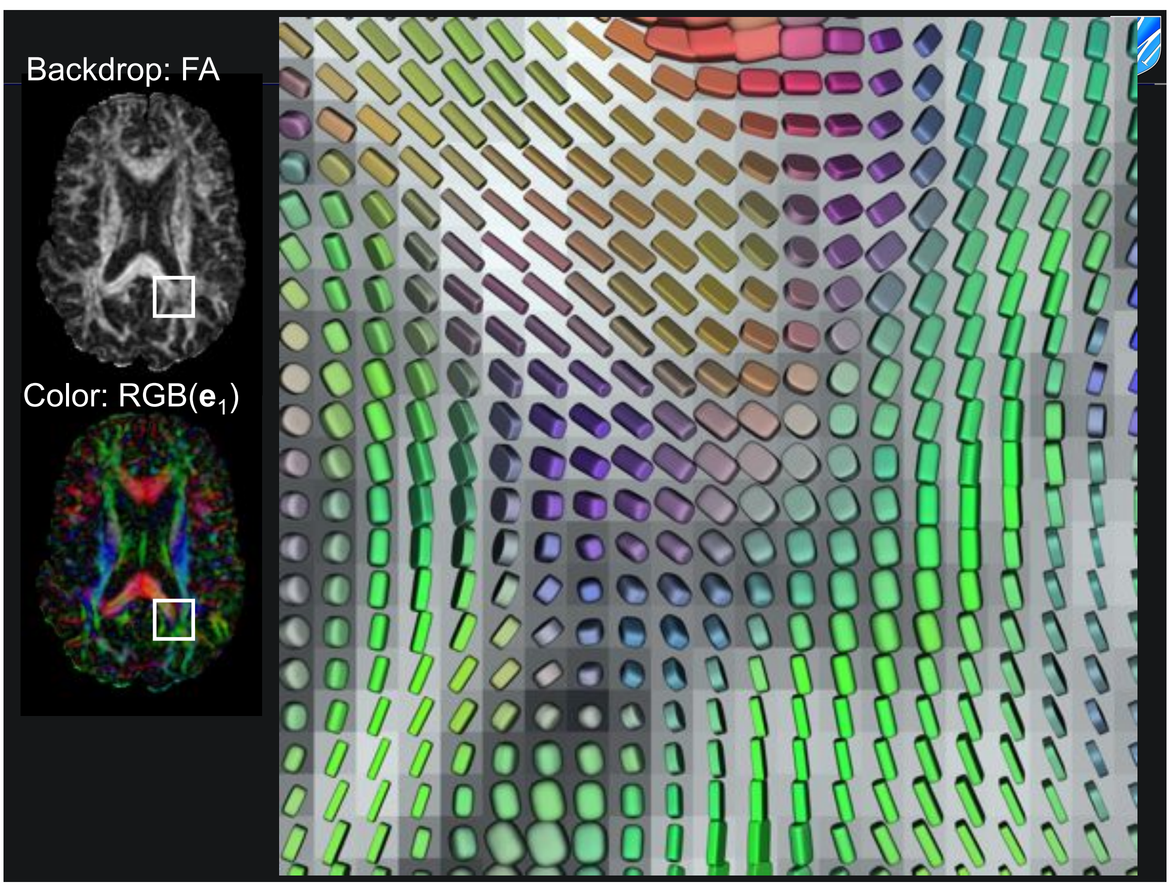


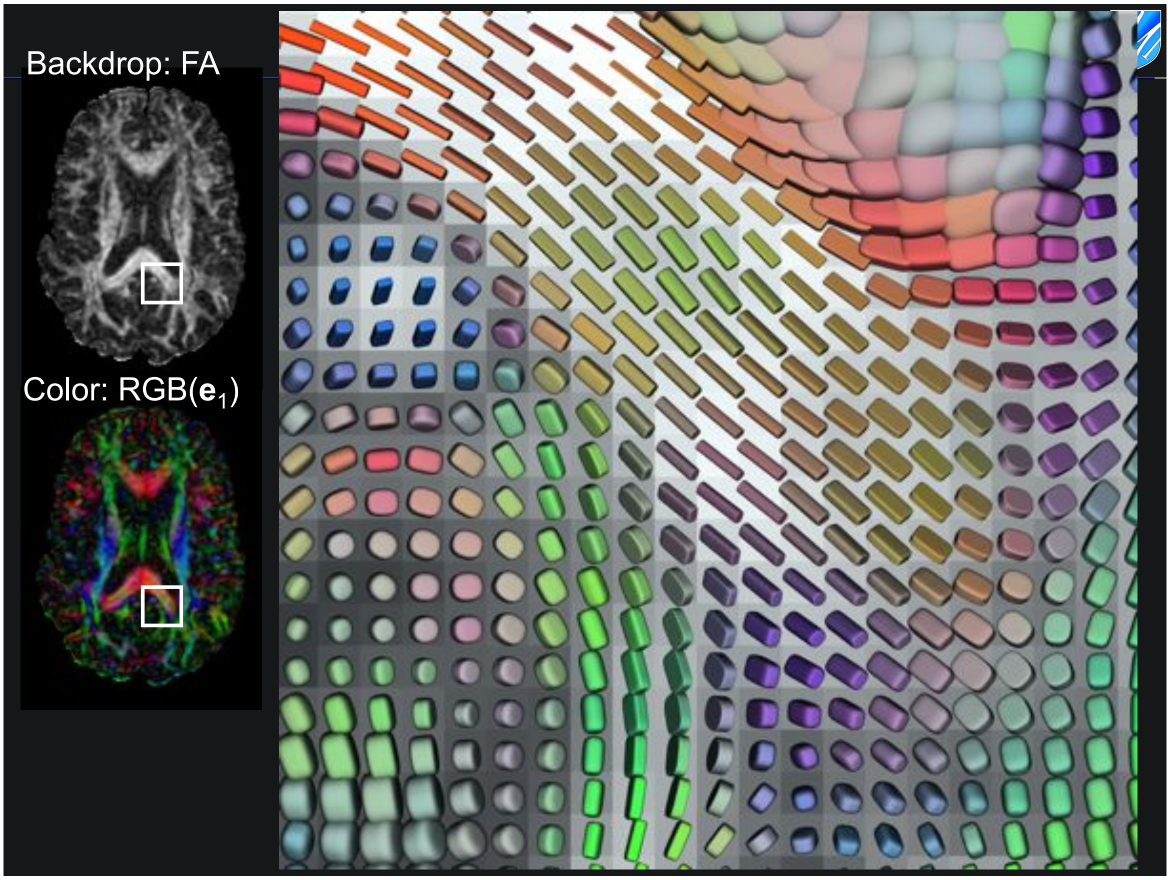
Worst case scenario, revisited

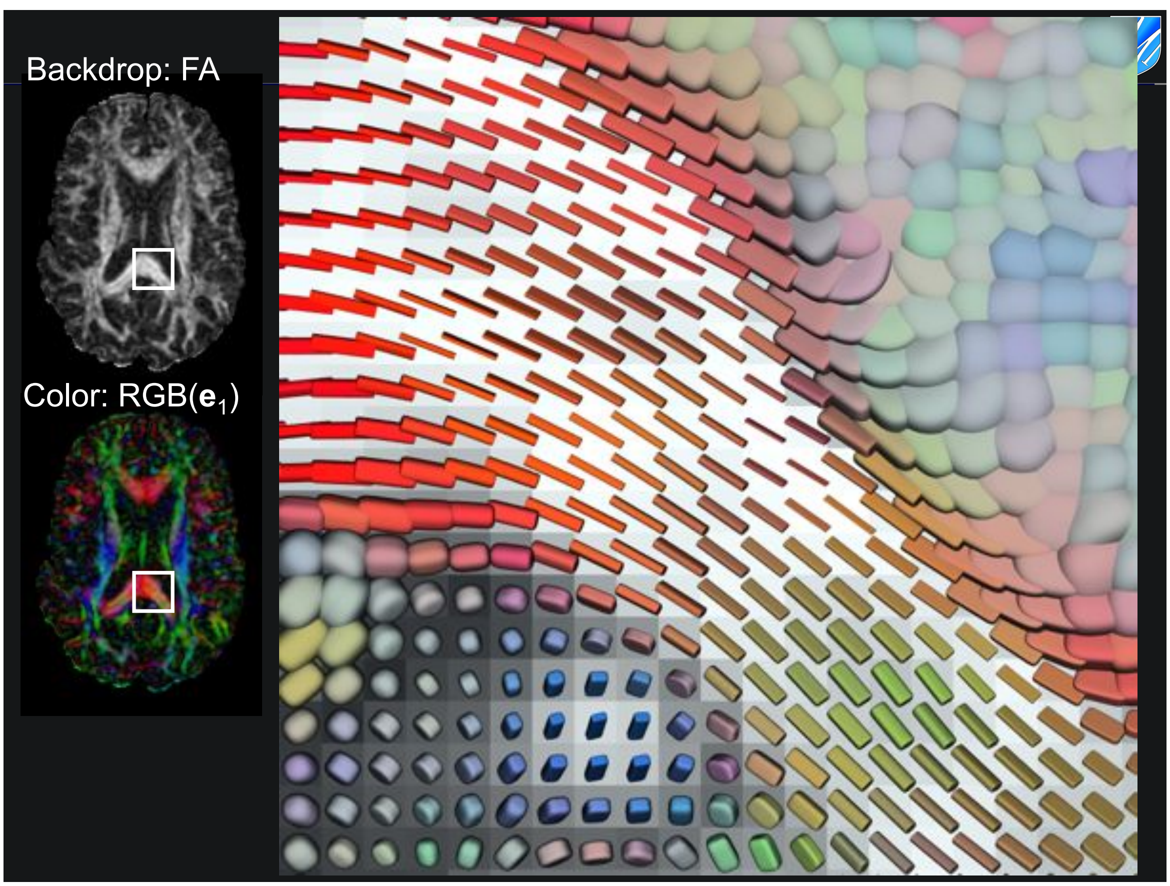


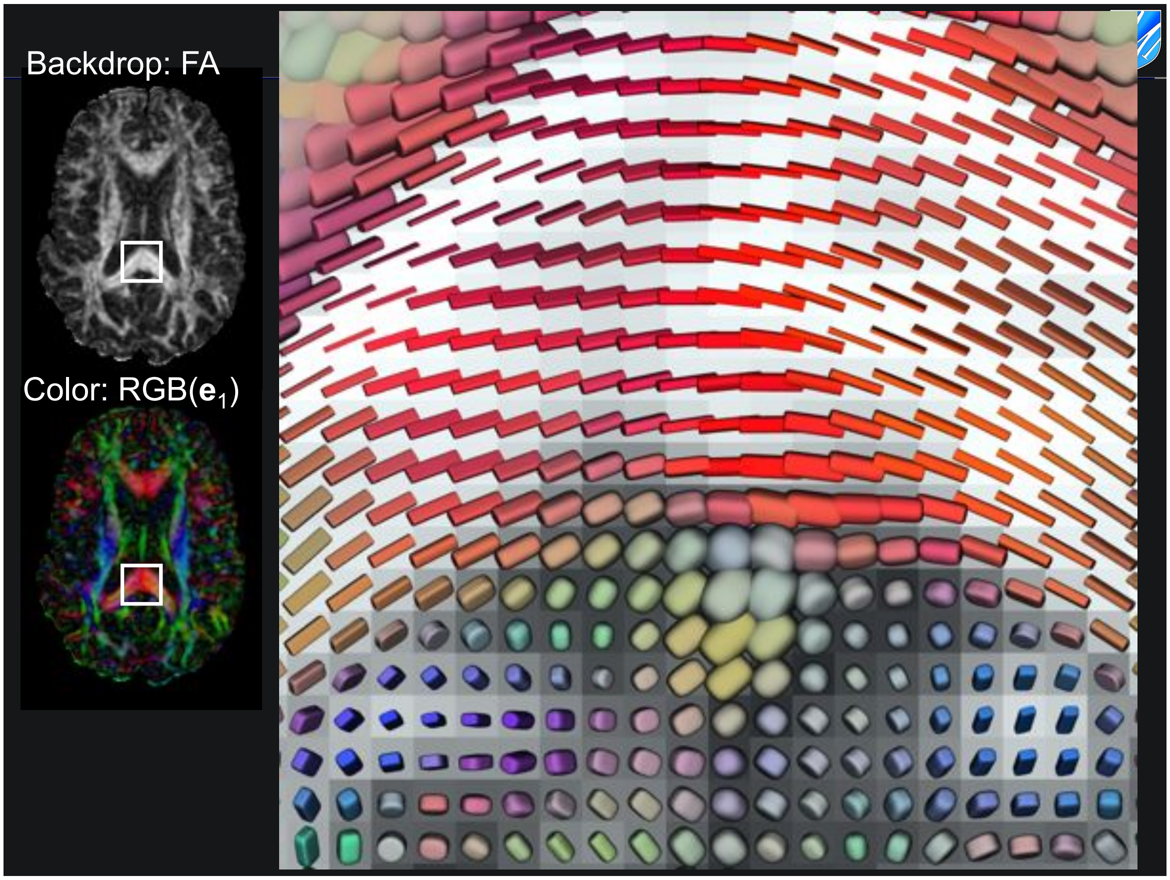


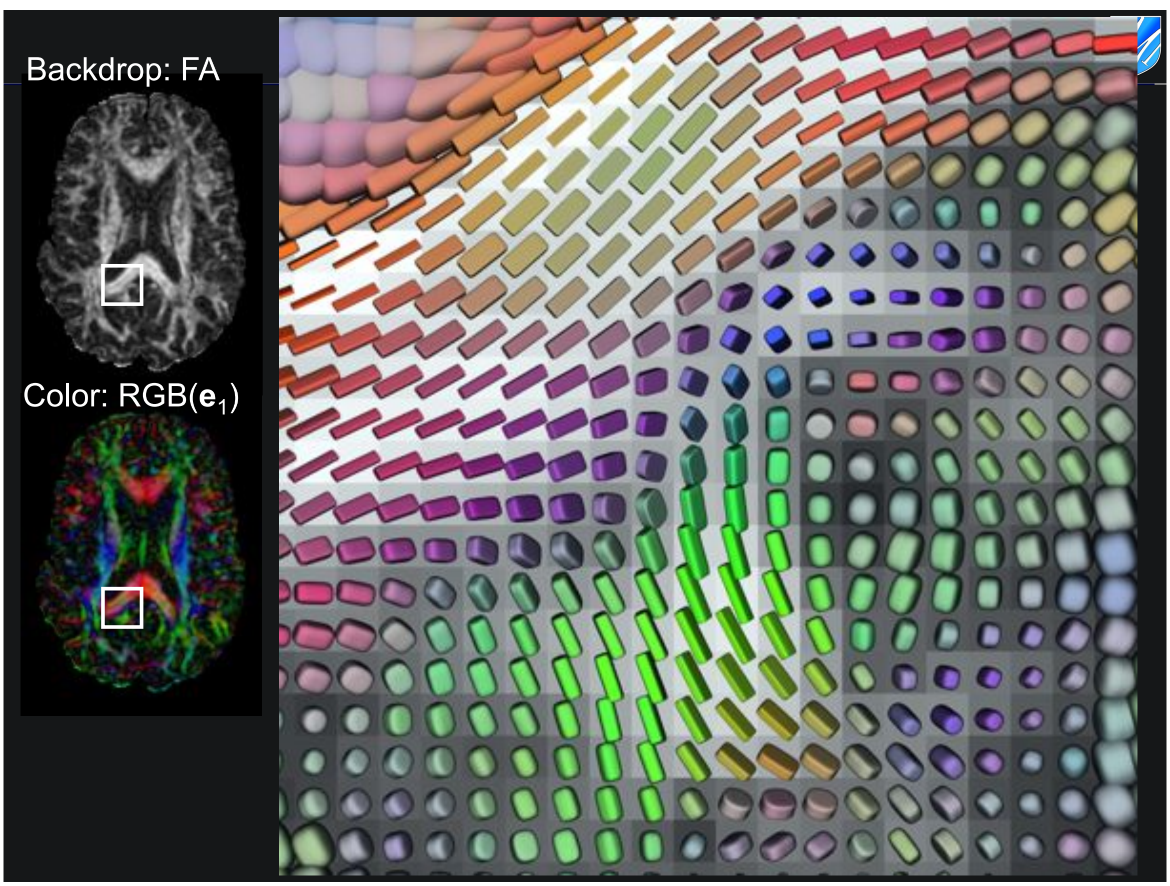

Backdrop: FA
Color: RGB(e1)

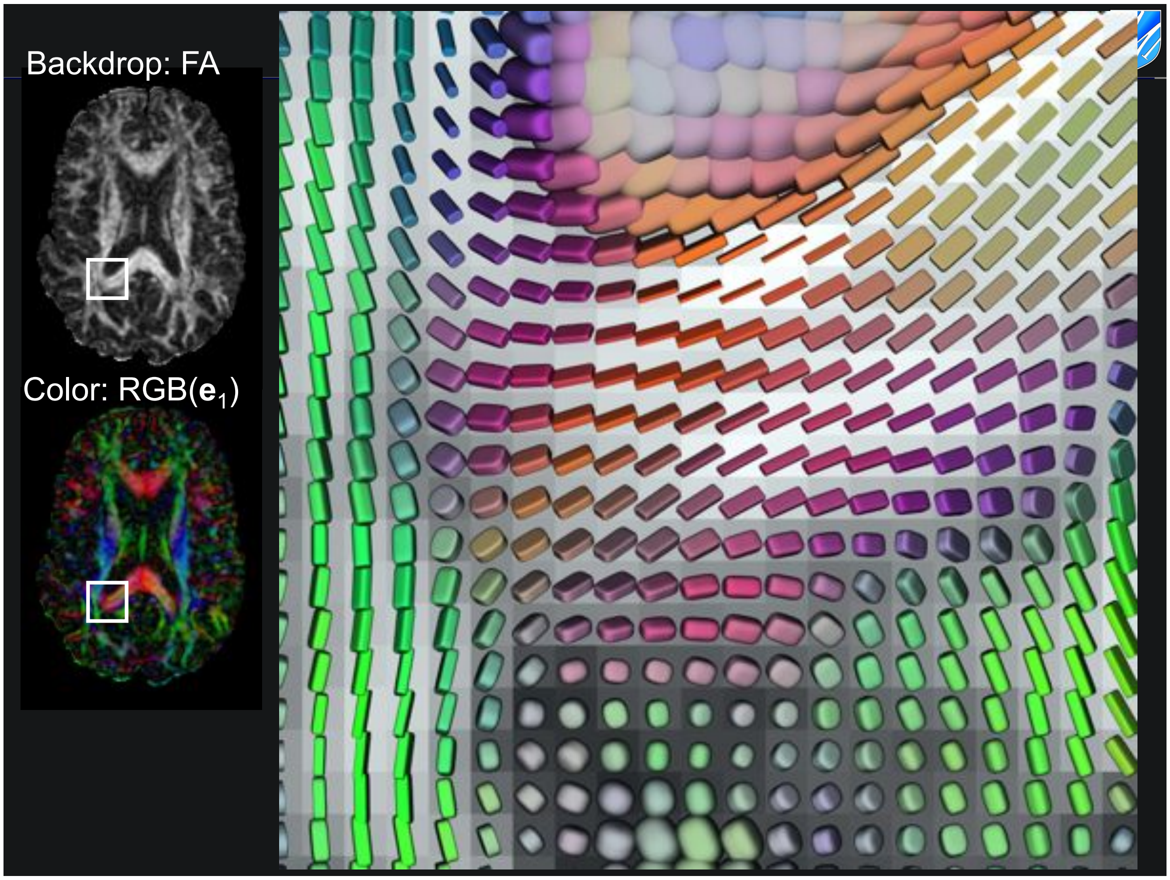

Backdrop: FA
Color: RGB(e1)


Backdrop: FA
Color: RGB($\mathbf{e}_1$)


Backdrop: FA
Color: RGB($\mathbf{e}_1$)


Backdrop: FA
Color: RGB($\mathbf{e}_1$)


Backdrop: FA
Color: RGB($\mathbf{e}_1$)



Outline

<u>Visualization</u>:

DWI Data source

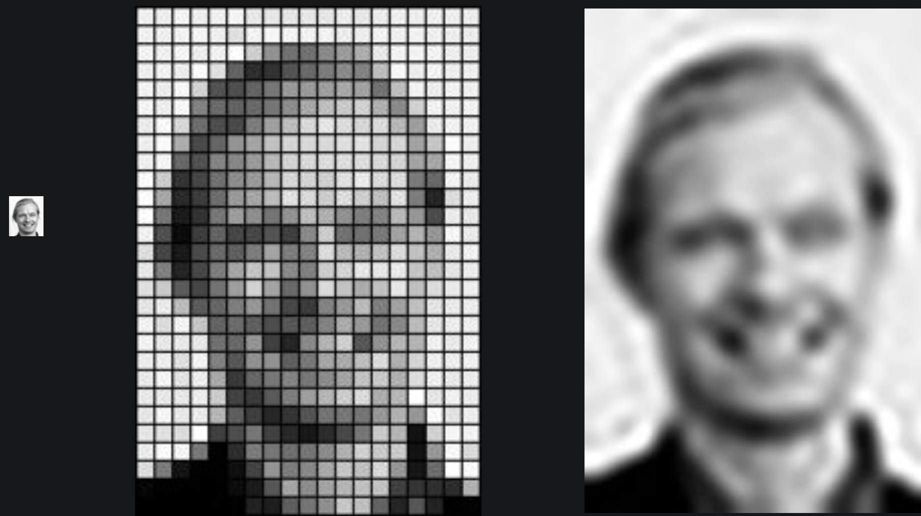
Space of Tensor Shape

Volume Rendering

Superquadric Glyphs

Glyph packing

Appearance matters



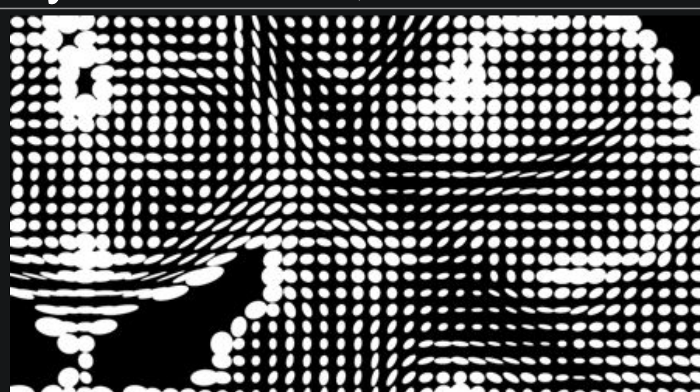
Display method affects ability to perceive data

Don’t introduce arbitrary or ill-defined structure

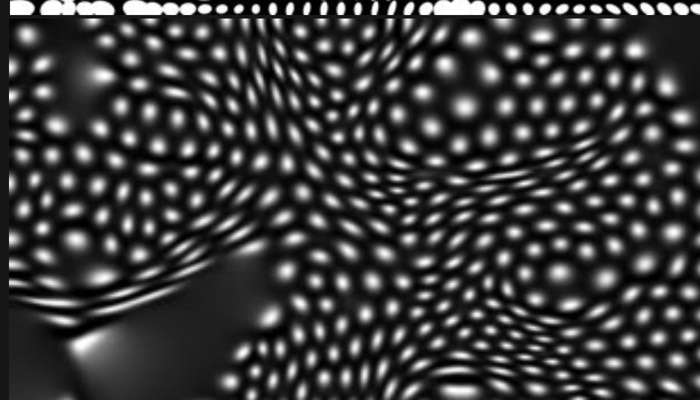
Textures tuned by tensor data, cont.

Real data

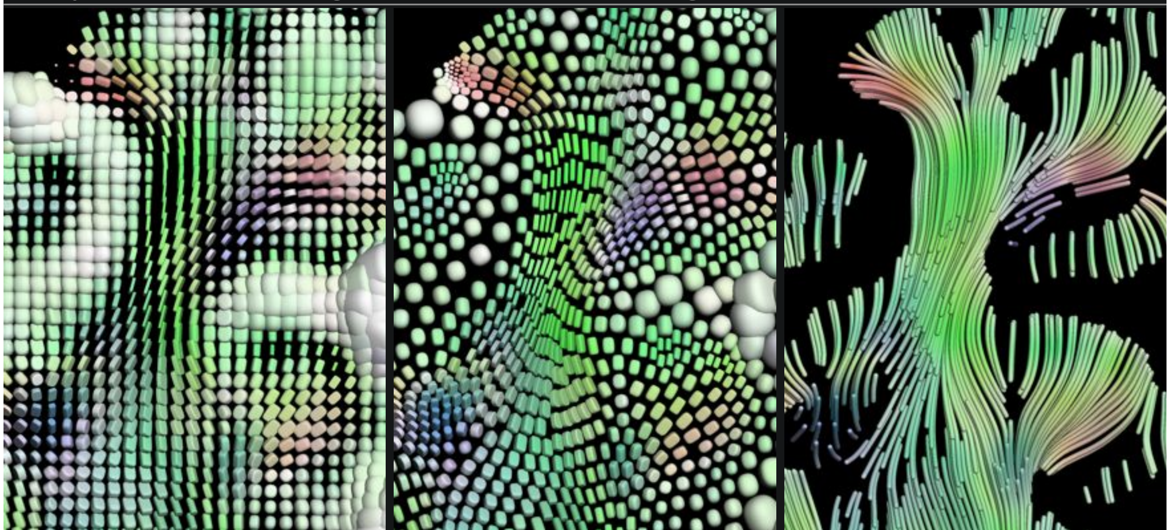
Ellipses

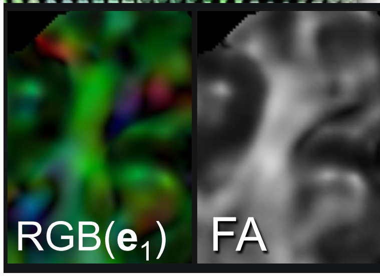


Texture



Glyph packing compared to grid, fibers



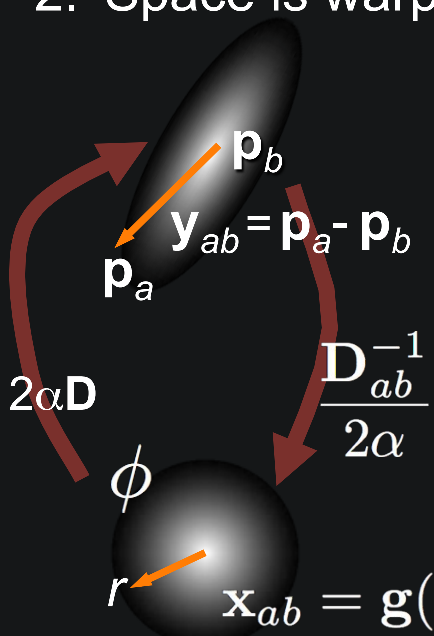


Formation of **discrete** glyphs into a **continuous** texture

Need tensor interpolation

Tensor-based potential energy

1. Particles push each other away
2. Space is warped by the local tensor



$$E_{ab} = \phi(r_{ab}) = \phi(|\mathbf{g}(\mathbf{y}_{ab})|)$$

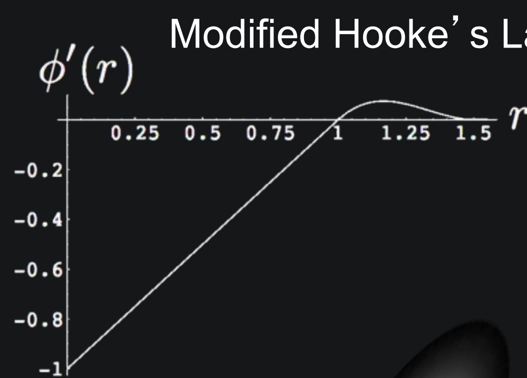
$$\mathbf{D}_{ab} = \mathbf{D}\left(\frac{\mathbf{p}_a + \mathbf{p}_b}{2}\right)$$

$$\mathbf{x}_{ab} = \mathbf{g}(\mathbf{y}_{ab}) = \frac{\mathbf{D}_{ab}^{-1}\mathbf{y}_{ab}}{2\alpha};\ r_{ab} = |\mathbf{x}_{ab}|$$

Force is spatial derivative of energy

$$E = \phi(|\mathbf{g}(\mathbf{y})|)$$

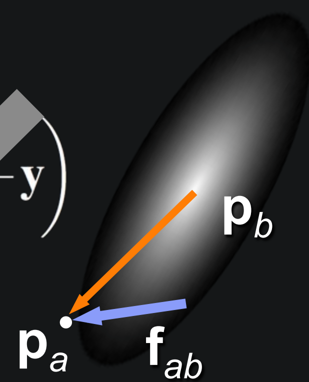
Modified Hooke's Law



$$\frac{dE}{d\mathbf{y}} = \frac{dE}{dr}\frac{dr}{d\mathbf{x}}\frac{d\mathbf{x}}{d\mathbf{y}}$$

$$= \phi'(|\mathbf{x}|)\frac{\mathbf{x}^T}{|\mathbf{x}|}\frac{1}{2\alpha}\left(\mathbf{D}^{-1} + \frac{d\mathbf{D}^{-1}}{d\mathbf{y}}\mathbf{y}\right)$$

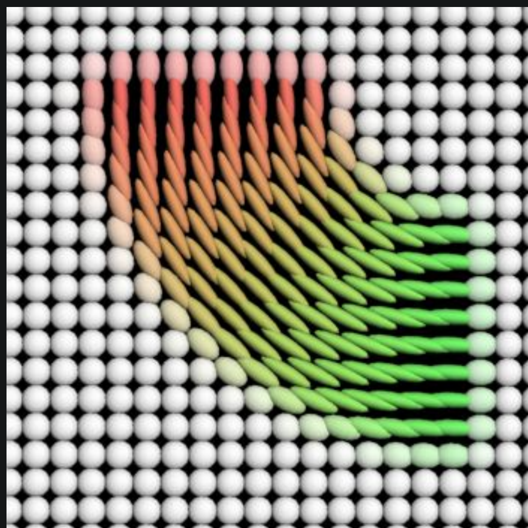
$$\mathbf{f}_{ab} = -\frac{\phi'(|\mathbf{x}_{ab}|)}{2\alpha}\mathbf{D}_{ab}^{-1}\frac{\mathbf{x}_{ab}}{|\mathbf{x}_{ab}|}$$

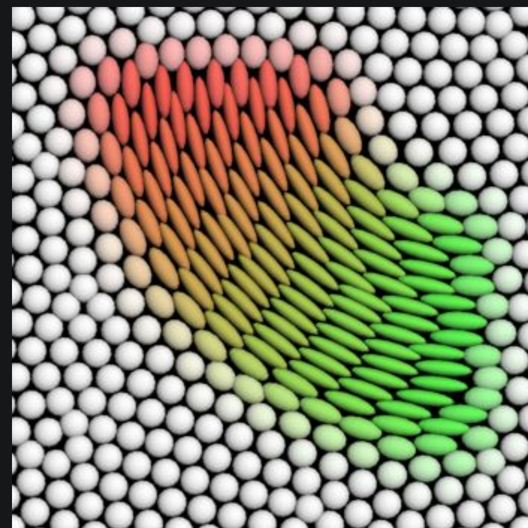


Implementation

- Spatial Binning
- Inverse approximation : $\mathbf{D}_{ab}^{-1} \approx \dfrac{\mathbf{D}^{-1}(\mathbf{p}_a) + \mathbf{D}^{-1}(\mathbf{p}_b)}{2}$
- Constraints on slices
- Solver: F=ma vs. Gradient descent
  - Order of faster convergence times than paper
- Probabilistically re-use probed tensors

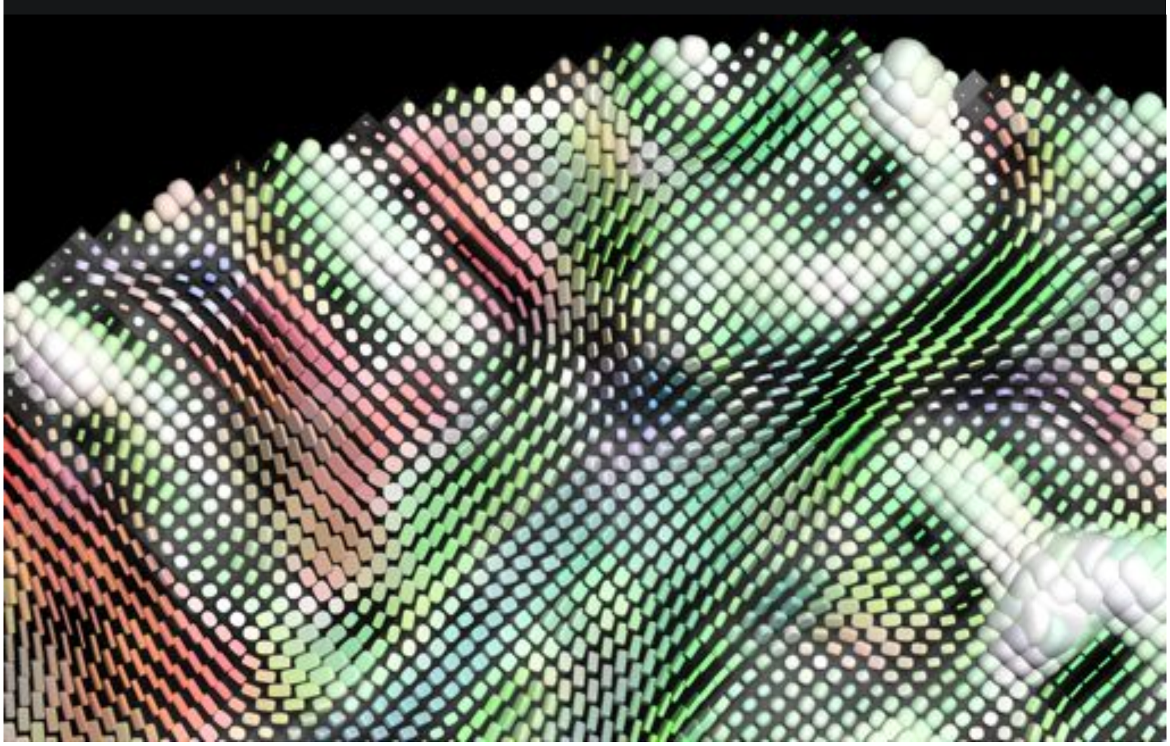
Results: synthetic data



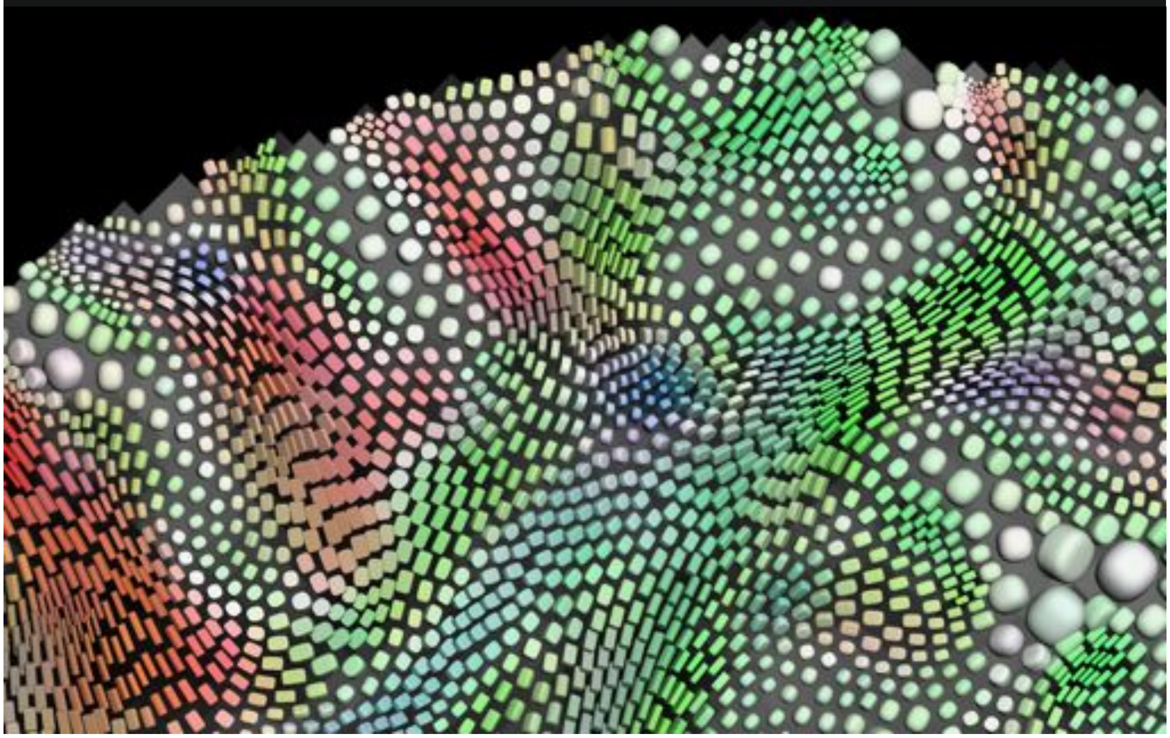


- Glyph overlap reduced
- Smoother orientation change
- Problem (?): new hexagonal grid
- Packing results can seed other glyph geometry

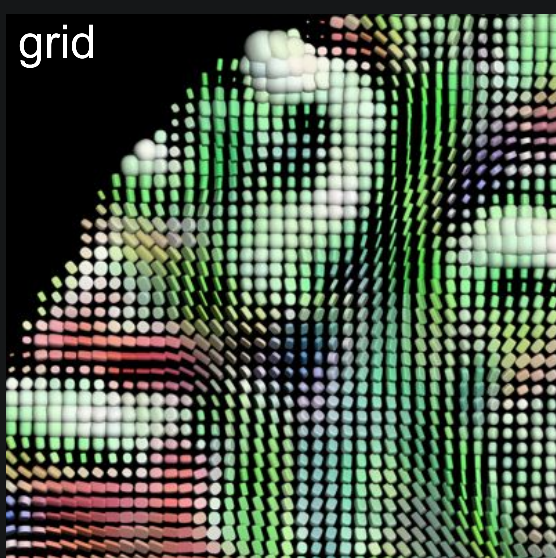
Results: Real data



Results: Real data



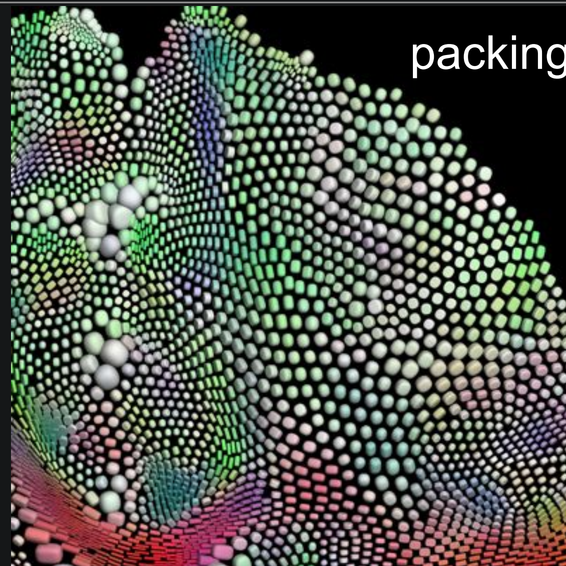
Results: healthy slice

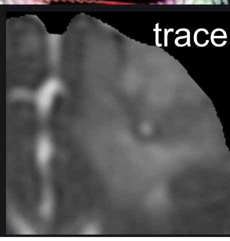


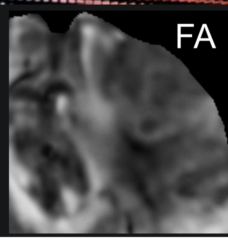
Tracts only in linear anisotropy; glyphs everywhere

Glyph packing complements tractography

Results: oligodendroglioma slice





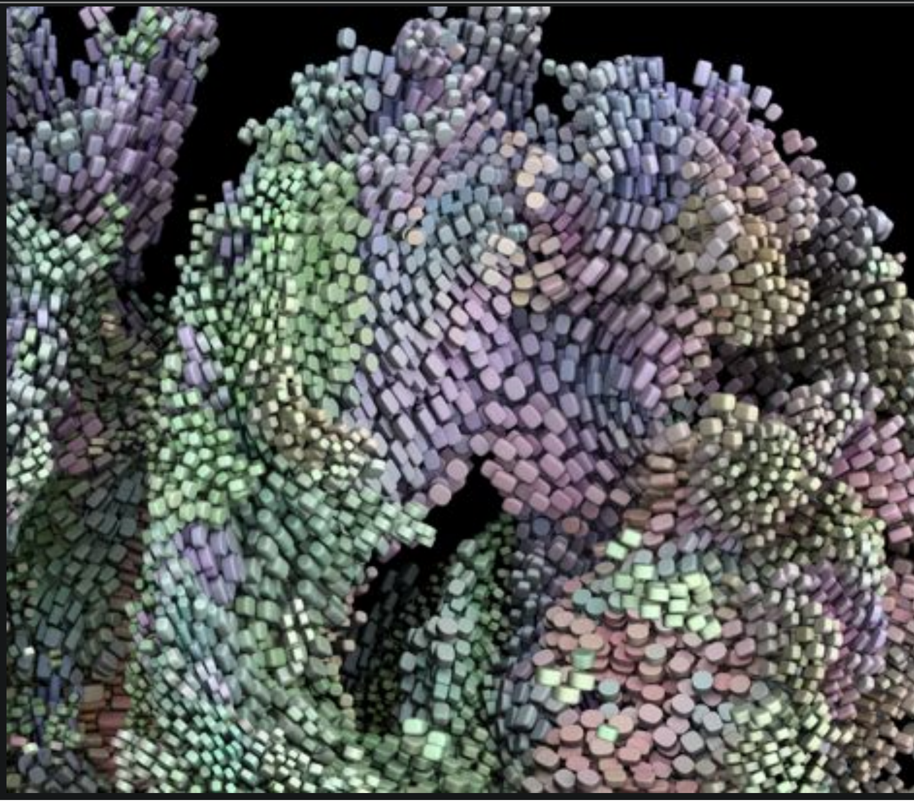


Boundary region of planar anisotropy

Results: volume of glyphs

Tarini et al. Vis '06 "Ambient Occlusion and Edge Cueing for Enhancing Real Time Molecular Visualization"

Results: volume of glyphs



Outline

Software:

Teem basics

Nrrd: for arrays

Gage: for convolution

Ten: for tensors

Deft: interactive

Outline

<u>Software:</u>

Teem basics

Nrrd: for arrays

Gage: for convolution

Ten: for tensors

Deft: interactive

Teem

Decided in 1999 to never again write "throw-away" code

Teem == the trail of crumbs of my research

Set of C libraries that play well together

Other people find it useful

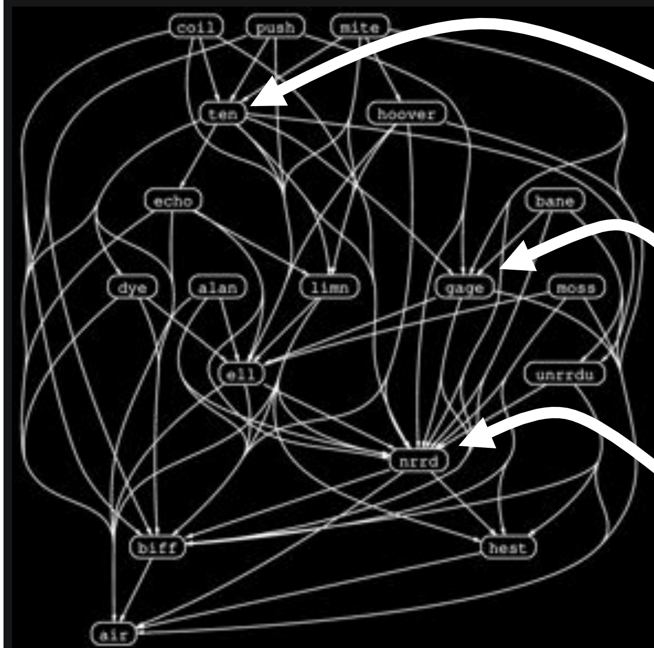
Part of SCIRun, Slicer, NA-MIC Toolkit

LGPL + Static linking (like FLTK)

http://teem.sourceforge.net

Teem

Coordinated set of C libraries for scientific data communication, processing, and visualization



Ten: DWI and DTI processing (tensor estimation, anisotropy metrics, fiber tractography)

Gage: convolution-based point sampling of scalar/ vector/tensor fields, and derived attributes

Nrrd: raster data representation, file I/O, and processing

Outline

Software:

Teem basics

Nrrd: for arrays

Gage: for convolution

Ten: for tensors

Deft: interactive

Nrrd

"Nearly Raw Raster Data"

Representation and manipulation of N-dimensional raster data

- File format
- Data structure
- Large set of operations

Associated "unu" command-line tool

Philosophy: "stark simplicity"

Generality it has is based on experience

Concept: scan-line ordering

N-D raster data has "scan-line ordering"

linear ordering in memory

```
(0, 0, 0)
(0, 0, 1)     } color
(0, 0, 2)
(0, 1, 0)
(0, 1, 1)     } color
(0, 1, 2)
 …                          } X
(0, 539, 0)
(0, 539, 1)
(0, 539, 2)
(1, 0, 0)
…                                  } Y
(2, 0, 0)
…
(379, 539, 0)
(379, 539, 1)
(379, 539, 2)
```

2-D color image →
3-D array color:X:Y
Each coordinate is an axis
Axes ordered: fast to slow; contiguous to distant

Fast: color
Medium: X
Slow: Y → (Y, X, color)

scalars, but each axis has "kind"

Important for data communication

NRRD DWI headers : Motivation

File format need

- Self-contained representation
- All image values (in original files)
- All DWI-specific parameters (b, $\mathbf{g}_i$, etc.)
- All coordinates (**including measurement frame!!**)
- DICOM is not a solution (DWI or DTI)
  - Manual sign flips/transposition ⇒ should **SAVE** answer

NA-MIC solution:

- NRRD **format** + key/value pair **convention**
- NRRD: image values + coordinates
  - **Explicit** representation of measurement frame
- Key/value pair convention: DWI-specific parameters

Now in use in multiple labs at Brigham & Women's, including for clinical neurosurgery

Example NRRD Header

```
NRRD0005
content: NAMIC01
type: short
dimension: 4
space: right-anterior-superior
sizes: 256 256 36 14
thicknesses:  NaN  NaN 3  NaN
space directions: (-0.9375,0,0) (0,-0.9375,0) (0,0,-3) none
centerings: cell cell cell none
kinds: space space space list
endian: little
encoding: raw
space units: "mm" "mm" "mm"
space origin: (125.00,124.10,79.3)
data file: S4.%03d 1 504 1 2
byte skip: -1
measurement frame: (0,-1,0) (1,0,0) (0,0,-1)
modality:=DWMRI
DWMRI_b-value:=800
DWMRI_gradient_0000:= 0 0 0
DWMRI_NEX_0000:=2
DWMRI_gradient_0002:= -0.8238094 -0.4178235 -0.3830949
DWMRI_gradient_0003:= -0.5681645 0.5019867 -0.6520725
 …
```

NRRD DWI header: data file possibilities

Volume interleaved, volume at a time:

```
data file: dwi%03d.gipl 1 12 1
byte skip: -1
sizes: 256 256 60 12
kinds: space space space list
```

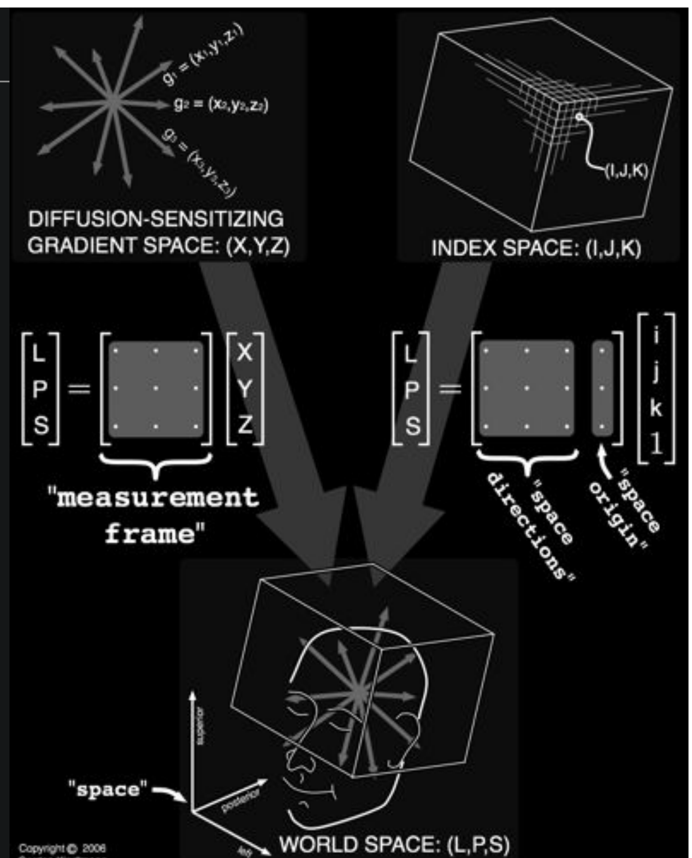
Volume interleaved: slice at a time

```
data file: MR_0028_%04d.dcm 1 448 1 2
byte skip: -1
sizes: 128 128 64 7
kinds: space space space list
```

Slice interleaved: slice at a time

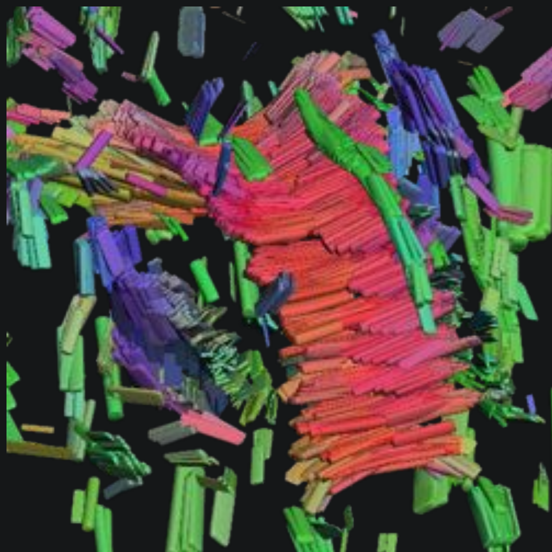
```
data file: test_%04d.dcm 1 448 1 2
byte skip: -1
sizes: 128 128 7 64
kinds: space space list space
```

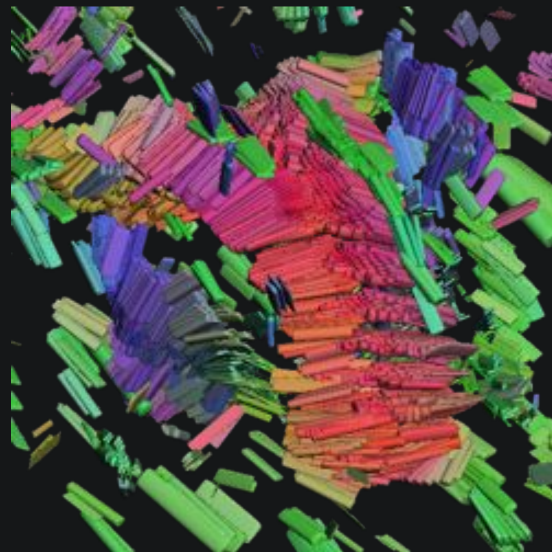
NRRD's full representation of image and data orientation



Measurement frame mysteries

Maps from gradient (X,Y,Z) to world (R,A,S)/(L,P,S)





`measurement frame:`

`(0,-1,0) (1,0,0) (0,0,-1)`

`measurement frame:`

`(0,1,0) (1,0,0) (0,0,1)`

Measurement frame bugs in pubs

Benger '06

Outline

Software:

Teem basics

Nrrd: for arrays

Gage: for convolution

Ten: for tensors

Deft: interactive

Gage

Convolution-based measurements

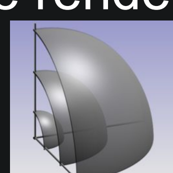
- Discrete data, continuous kernels

Scalars, 3-Vectors, DTIs, DWIs

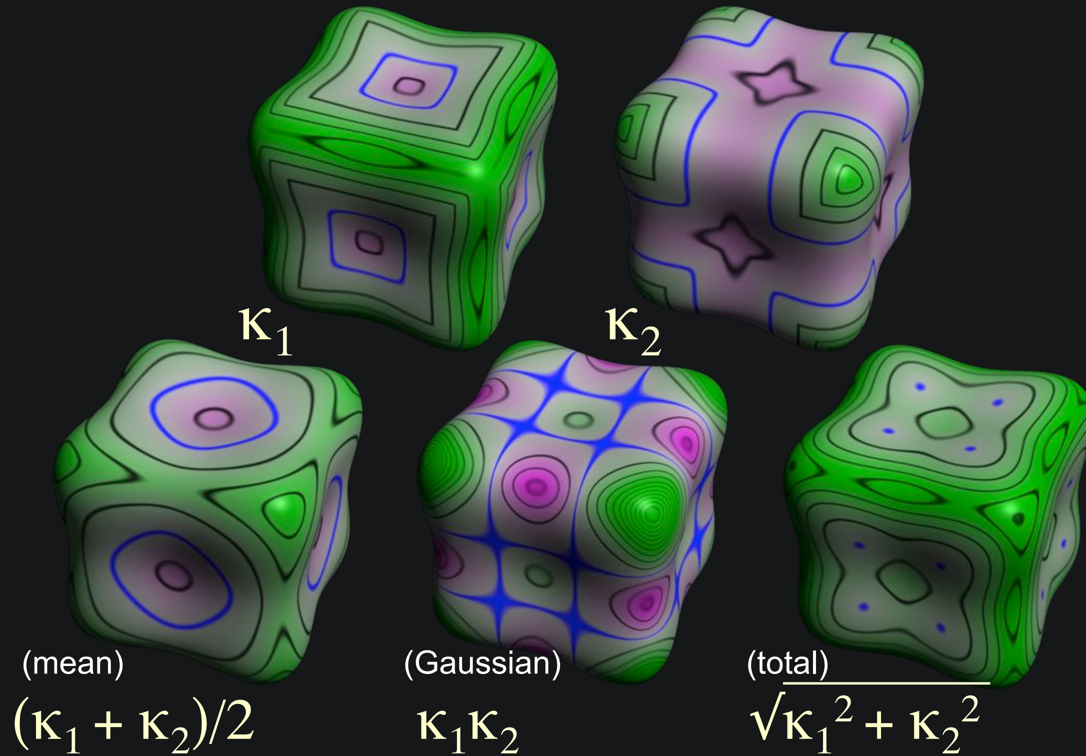
Values, 1st, 2nd derivatives

- e.g. curvature tensor in scalars

Basis of volume rendering, tractography



Curvature measures

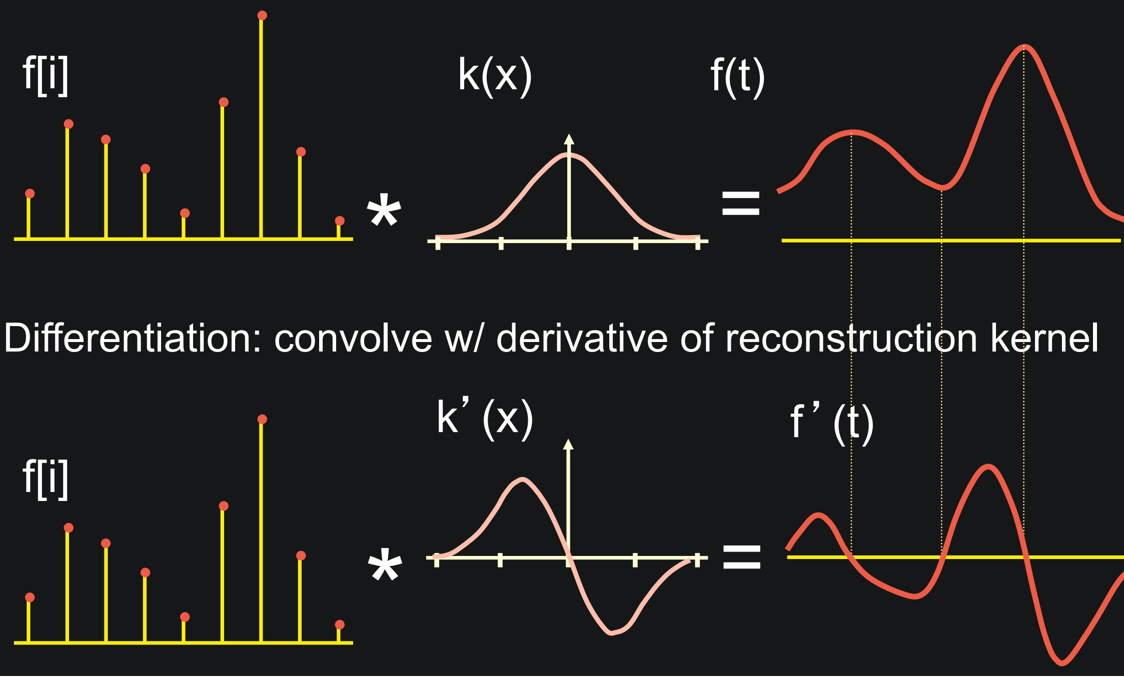


Volume NPR: results

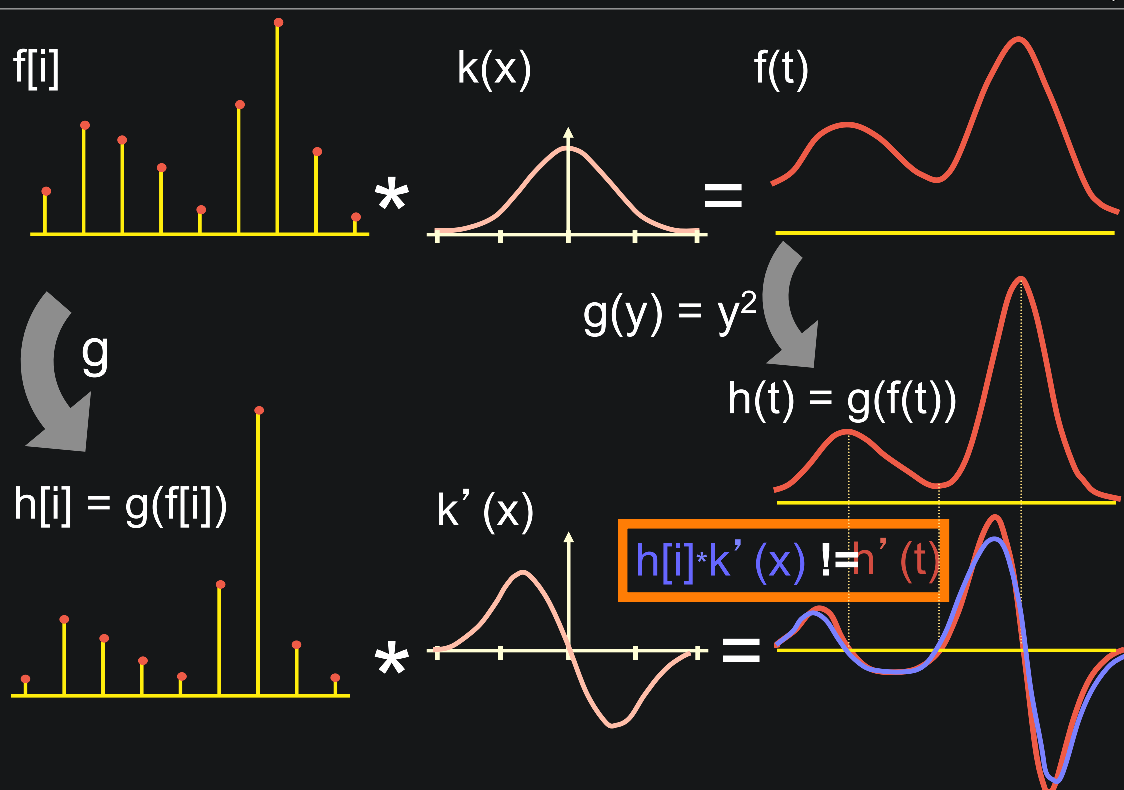
Measurement (of scalars) by convolution

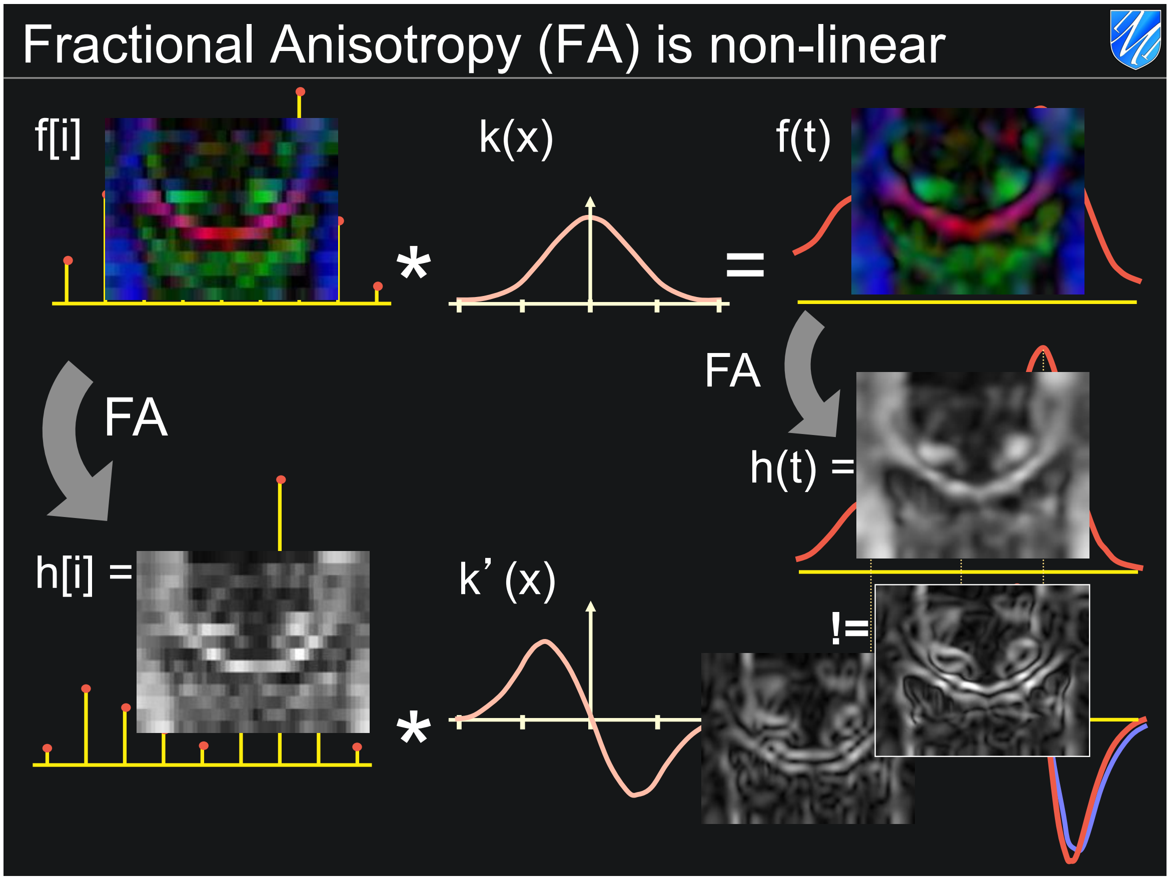
Continuous field: convolution of sampled coefficients with continuous reconstruction kernels

Differentiation: convolve w/ derivative of reconstruction kernel

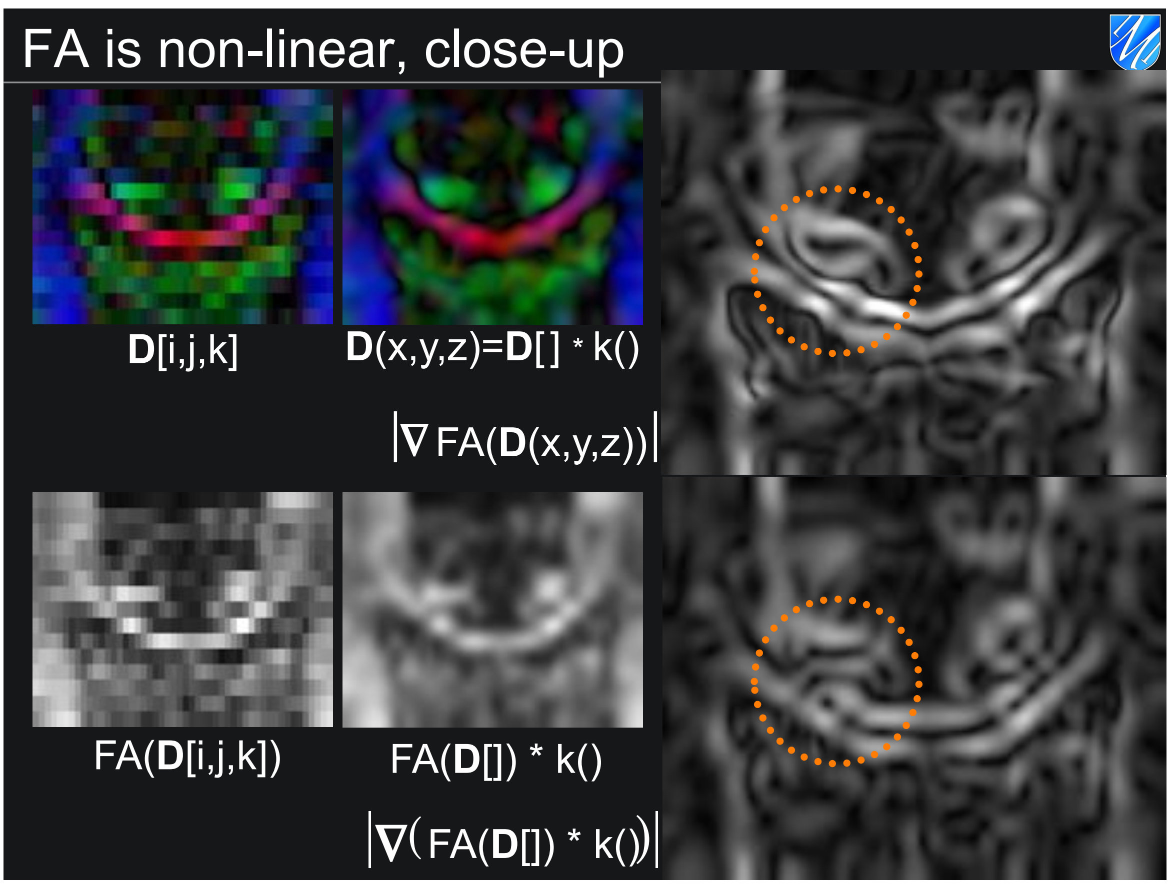


Non-linear transform of data





Fractional Anisotropy (FA) is non-linear
f[i]
k(x)
f(t)
*
=
FA
FA
h(t) =
h[i] =
k' (x)
!=
*



FA is non-linear, close-up
D[i,j,k]
D(x,y,z)=D[] * k()
|∇ FA(D(x,y,z))|
FA(D[i,j,k])
FA(D[]) * k()
|∇(FA(D[]) * k())|

FA from invariants, from coefficients

$$\mathrm{FA} \equiv \sqrt{\frac{3\,D{:}D}{2\,\mathbf{D}{:}\mathbf{D}}} \qquad D = \mathbf{D} - \langle \mathbf{D} \rangle \mathbf{I} \qquad \mathbf{D}{:}\mathbf{D} = \mathrm{tr}(\mathbf{D}\mathbf{D}^T)$$

$$J_2 = \begin{array}{c} D_{xx}D_{yy} + D_{xx}D_{zz} + D_{yy}D_{zz} \\ -D_{xy}^2 - D_{xz}^2 - D_{yz}^2 \end{array}$$

$$\mathrm{FA} = 3\sqrt{\frac{Q}{S}} \qquad Q = \frac{S - J_2}{9}$$

$$S = \mathbf{D}{:}\mathbf{D} = \begin{array}{c} D_{xx}^2 + D_{yy}^2 + D_{zz}^2 \\ +2D_{xy}^2 + 2D_{xz}^2 + 2D_{yz}^2 \end{array}$$

$$\nabla J_2 = \begin{array}{c} (D_{yy} + D_{zz})\nabla D_{xx} + (D_{xx} + D_{zz})\nabla D_{yy} + (D_{xx} + D_{yy})\nabla D_{zz} \\ -2D_{xy}\nabla D_{xy} - 2D_{xz}\nabla D_{xz} - 2D_{yz}\nabla D_{yz} \end{array}$$

$$\nabla Q = \frac{\nabla S - \nabla J_2}{9}$$

$$\nabla \mathrm{FA} = \frac{3}{2}\left(\sqrt{\frac{1}{SQ}}\nabla Q - \sqrt{\frac{Q}{S^3}}\nabla S\right)$$

$$\nabla S = \begin{array}{c} 2D_{xx}\nabla D_{xx} + 2D_{yy}\nabla D_{yy} + 2D_{zz}\nabla D_{zz} \\ +4D_{xy}\nabla D_{xy} + 4D_{xz}\nabla D_{xz} + 4D_{yz}\nabla D_{yz} \end{array}$$

Hessian(FA) more complicated, but similarly derived

Outline

Software:

Teem basics

Nrrd: for arrays

Gage: for convolution

Ten: for tensors

Deft: interactive

Ten

All things tensor-related:

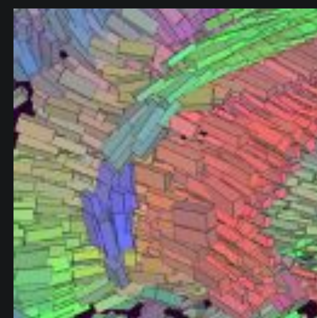
DWI gradient optimization

Tensor estimation

Anisotropy calculations

Fast 3x3 eigensolution

Tractography



Outline

<u>Software:</u>

Teem basics

Nrrd: for arrays

Gage: for convolution

Ten: for tensors

Deft: interactive

Deft: interactive tensor vis

Deft: written in C++/OpenGL

- Built on top of Teem

Compiled with Cmake

MIT License

Deft is **not** for clinicians

low-level data inspection: “xv for tensors” (glyphs)

Triangle strips, glyph palette

Algorithmic visualization

Tractography parameter space

Esto es Todo

Gracias

Preguntas, por favor?