Sampling and Visualizing Creases with Scale-Space Particles

Gordon L. Kindlmann
THE UNIVERSITY OF CHICAGO
COMPUTER SCIENCE

Raúl San José, Carl-Fredrik Westin
Laboratory for Mathematics in Imaging
Brigham & Women's Hospital
Harvard Medical School

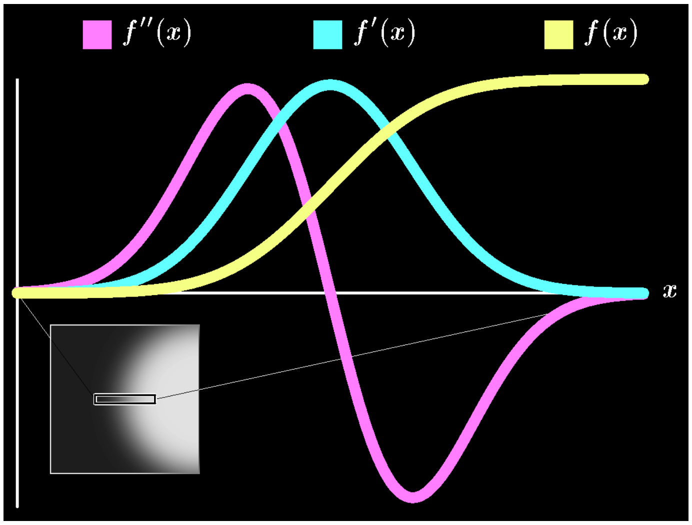
Stephen M. Smith
Center for Functional MRI of the Brain
John Radcliffe Hospital
Oxford University
FMRIB

Background

- Four papers that contribute to the **strategy** and **methodology** of the scale-space particles
- Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering (1998)
- Curvature-Based Transfer Functions... (2003)
- Diffusion Tensor Visualization with Glyph Packing (2006)
- Delineating White Matter Structure in Diffusion Tensor MRI with Anisotropy Creases (2007)

Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering (Kindlmann & Durkin)

Volume Visualization Symposium, Visualization '98

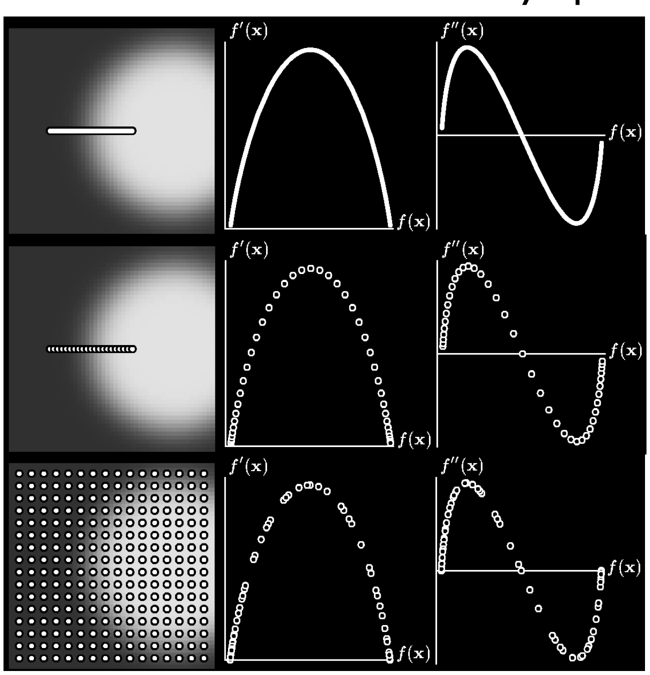


Edge as maxima of f', 0-crossing of f"

- Used histograms to measure relationship between scalar values and directional derivatives along gradient direction
- gradient $\mathbf{g} = \nabla f$
- Hessian $\mathbf{H} = \nabla \otimes \nabla f$
- f' = $|\mathbf{g}|$
- f" = $\frac{1}{|\mathbf{g}|^2} \mathbf{g} \cdot \mathbf{H}\mathbf{g}$

Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering (Kindlmann & Durkin)

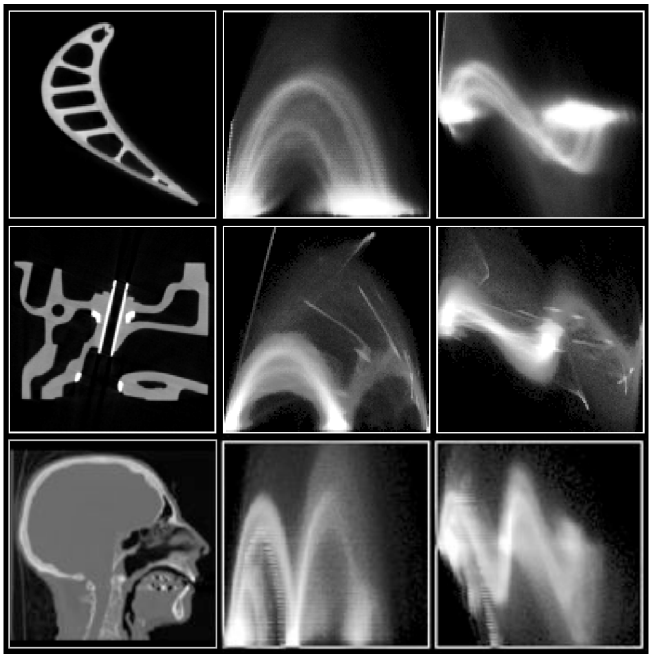
Volume Visualization Symposium, Visualization '98



- Used histograms to measure relationship between scalar values and directional derivatives along gradient direction
- gradient $\mathbf{g} = \nabla f$
- Hessian $\mathbf{H} = \nabla \otimes \nabla f$
- f' = $|\mathbf{g}|$
- f" = $\frac{1}{|\mathbf{g}|^2} \mathbf{g} \cdot \mathbf{H}\mathbf{g}$

Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering (Kindlmann & Durkin)

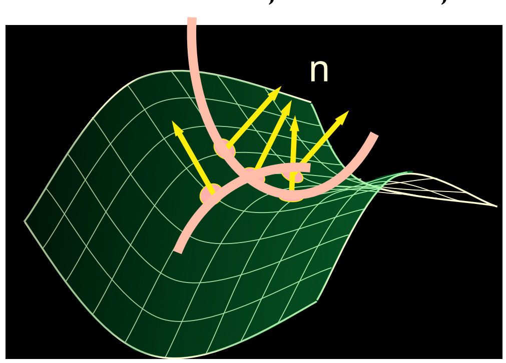
Volume Visualization Symposium, Visualization '98

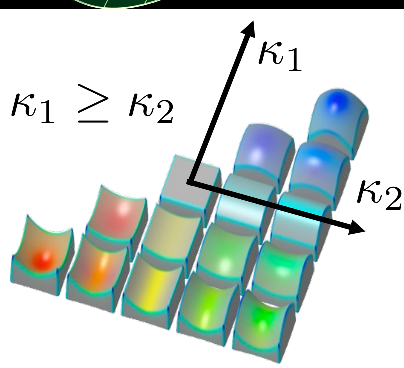


- Used histograms to measure relationship between scalar values and directional derivatives along gradient direction
- gradient $\mathbf{g} = \nabla f$
- Hessian $\mathbf{H} = \nabla \otimes \nabla f$
- f' = $|\mathbf{g}|$
- f'' = $\frac{1}{|\mathbf{g}|^2}\mathbf{g} \cdot \mathbf{H}\mathbf{g}$

Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications

Kindlmann, Whitaker, Tasdizen, Möller; Visualization '03

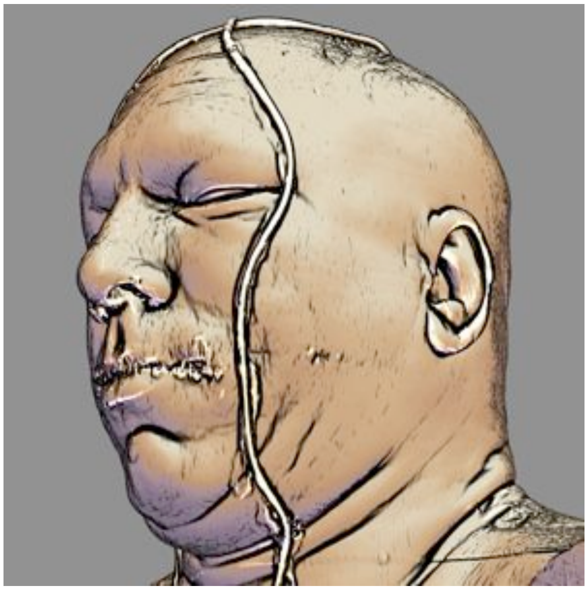


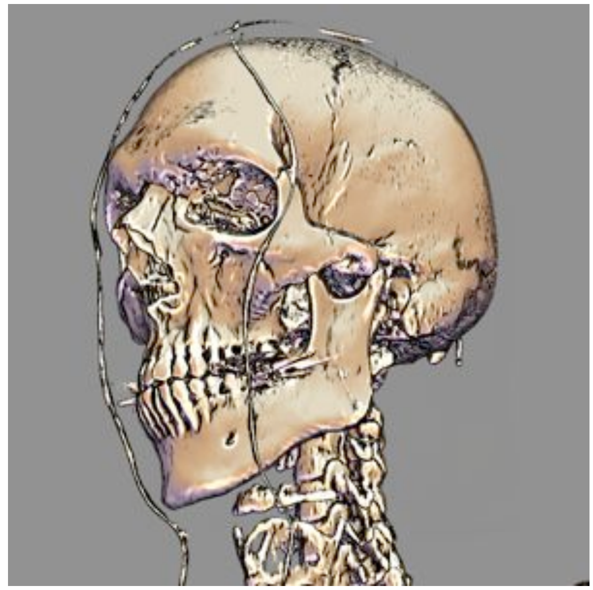


- Applied implicit surface curvature information to direct volume rendering
- gradient $\mathbf{g} = \nabla f$
- "normal" $\mathbf{n} = -\mathbf{g}/|\mathbf{g}|$
- $\mathbf{P} = \mathbf{I} - \mathbf{n} \otimes \mathbf{n}$
- Hessian $\mathbf{H} = \nabla \otimes \nabla f$
- $\mathbf{G} = \mathbf{PHP}/|\mathbf{g}|$
- $[\mathbf{G}] = \begin{bmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications

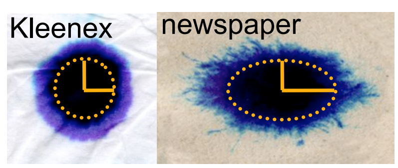
Kindlmann, Whitaker, Tasdizen, Möller; Visualization '03





Diffusion Tensor MRI background

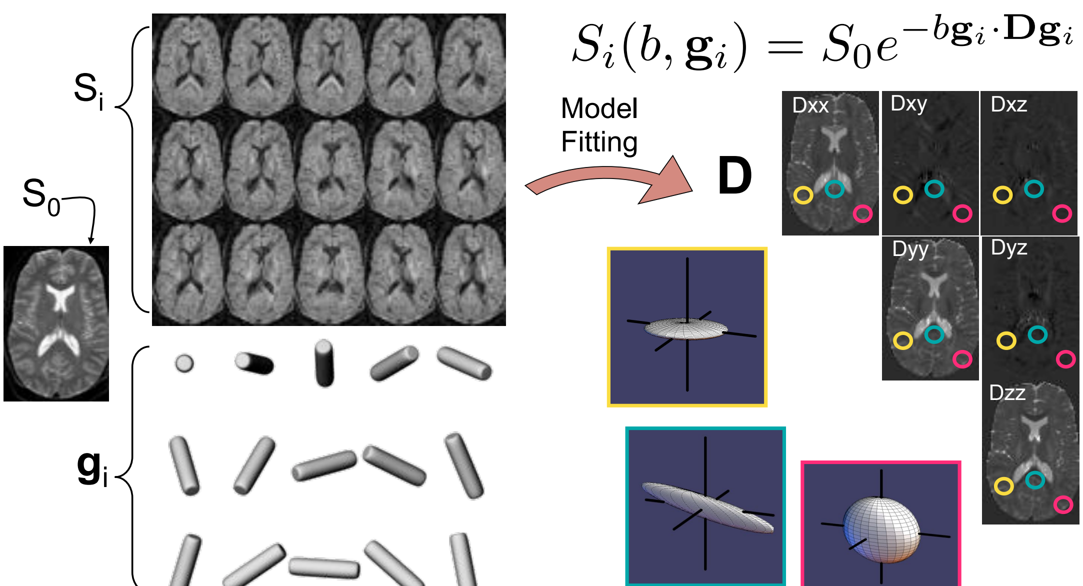
Rate of water diffusion in nervous system is directionally dependent



<u>Anisotropy</u>

Tensor Model:
$d(\mathbf{v}) = \mathbf{v} \cdot \mathbf{D}\mathbf{v}$

$$S_i(b, \mathbf{g}_i) = S_0 e^{-b\mathbf{g}_i \cdot \mathbf{D}\mathbf{g}_i}$$

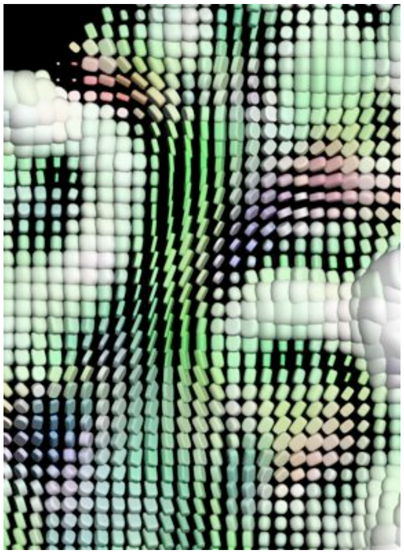


Diffusion Tensor Visualization with Glyph Packing

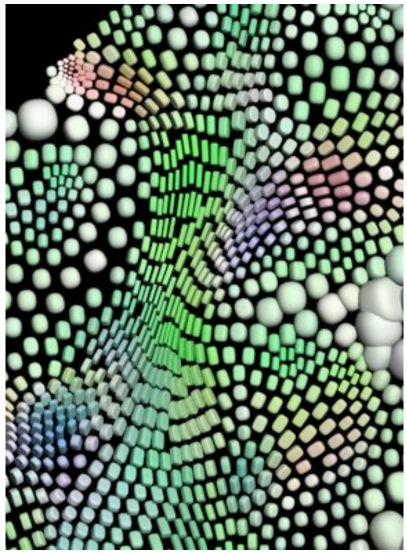
(Kindlmann & Westin, Visualization '06)

Tries to show underlying structures with texture of glyphs

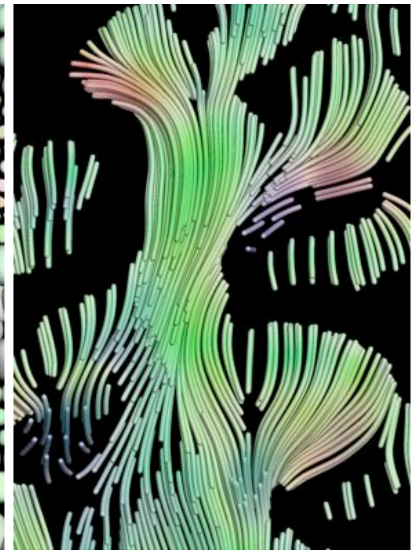
Created with energy-minimizing **particle system**



Glyphs on grid



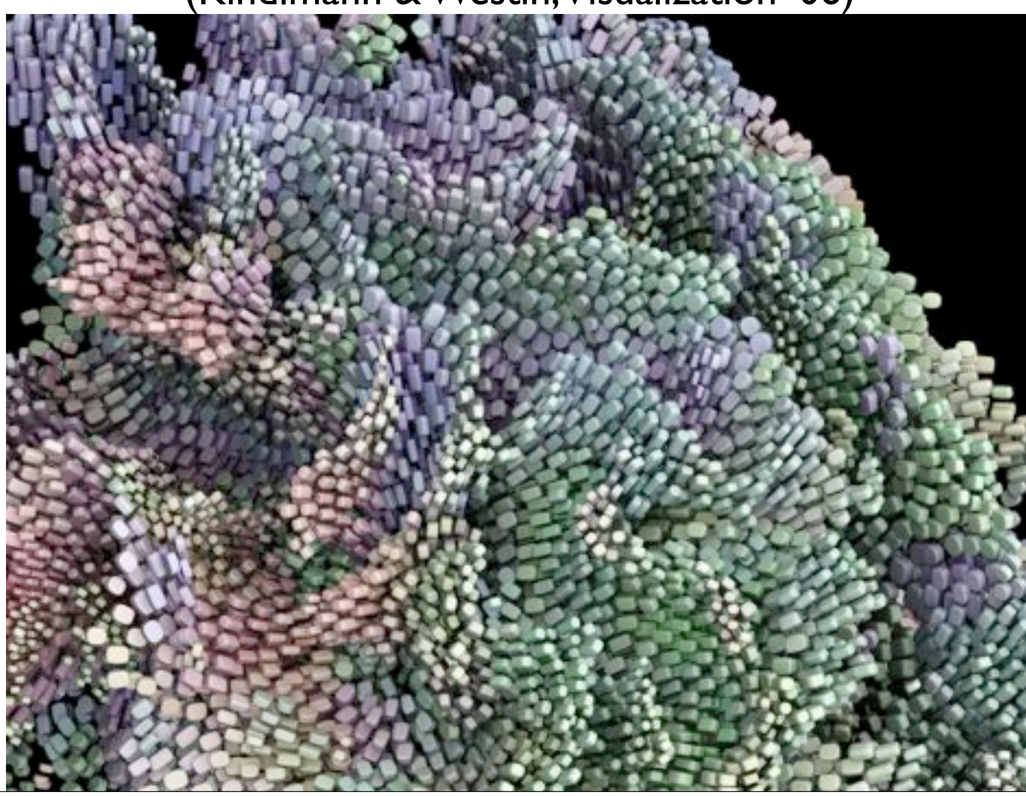
Glyph Packing



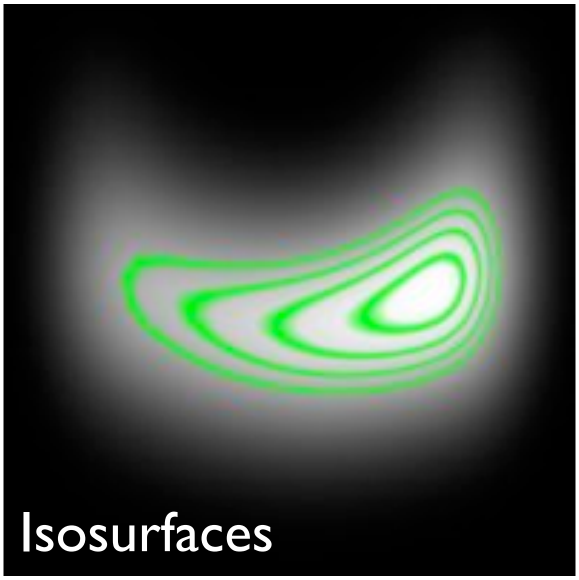
Tractography

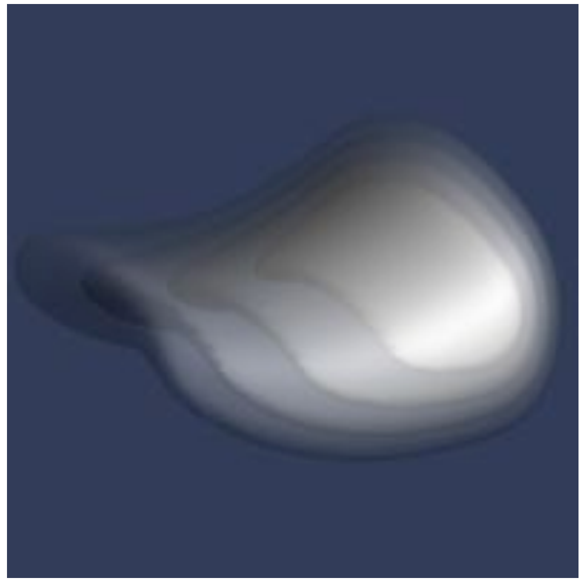
Diffusion Tensor Visualization with Glyph Packing

(Kindlmann & Westin, Visualization '06)



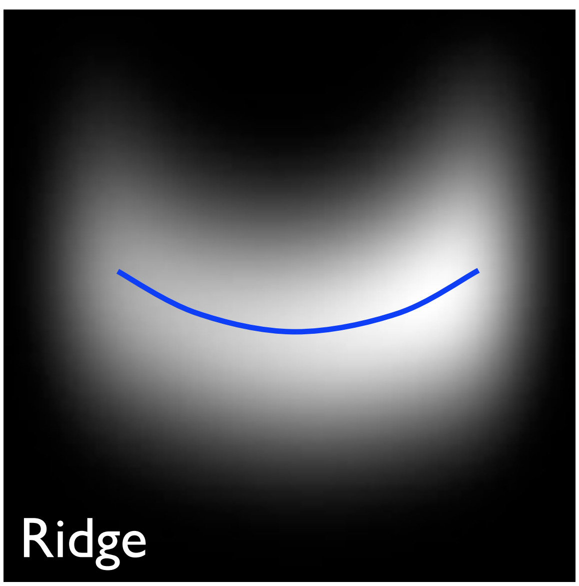
Ridge & Valley background

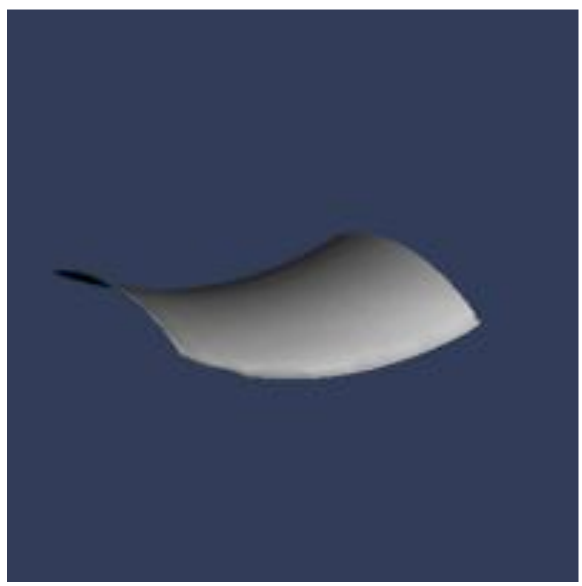




Isosurfaces collectively show global structure
Individual isosurface depends on isovalue **parameter**

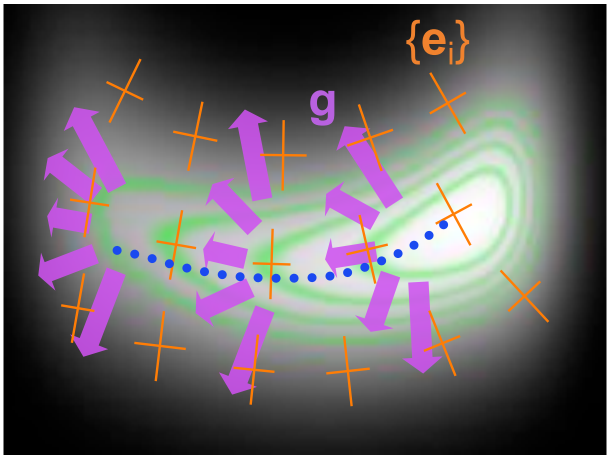
Ridge & Valley background





Ridges & valleys (collectively, **creases**) more intrinsic
No parameter dependence, more invariant
⇒ Can be better at capturing essential structure

Ridge/Valley (Crease) background



Eberly 1994

Constrained extremum

Gradient **g**

Hessian eigensystem $\mathbf{e}_i, \lambda_i$

Crease: **g** orthogonal to one or more $\mathbf{e}_i$

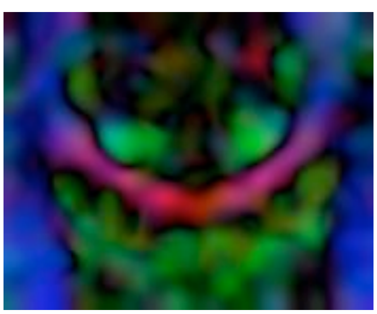
Eigenvalue gives strength

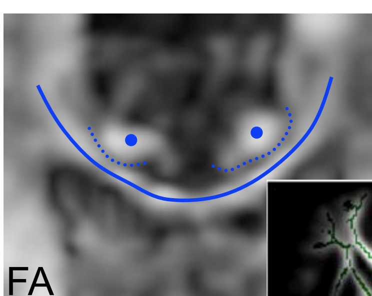
Ridge surface: $\mathbf{g} \cdot \mathbf{e}_3 = 0$; $\lambda_3 <$ thresh

Ridge line: $\mathbf{g} \cdot \mathbf{e}_3 = \mathbf{g} \cdot \mathbf{e}_2 = 0$; $\lambda_3, \lambda_2 <$ thresh

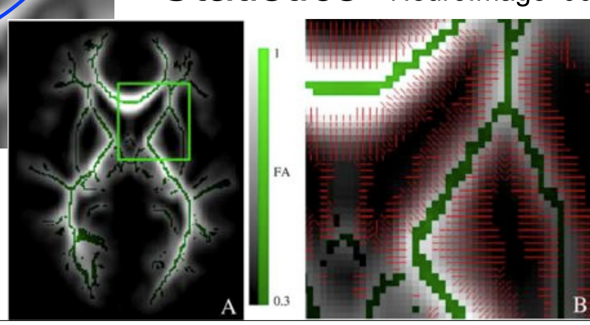
Valley surface: $\mathbf{g} \cdot \mathbf{e}_1 = 0$; $\lambda_1 >$ thresh

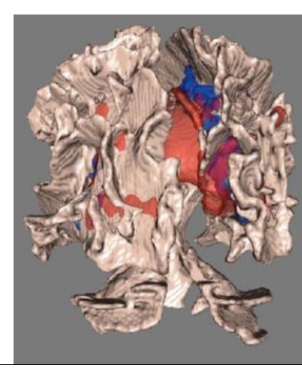
Delineating White Matter Structure in Diffusion Tensor MRI with Anisotropy Creases (Kindlmann et al. Medical Image Analysis 2007)

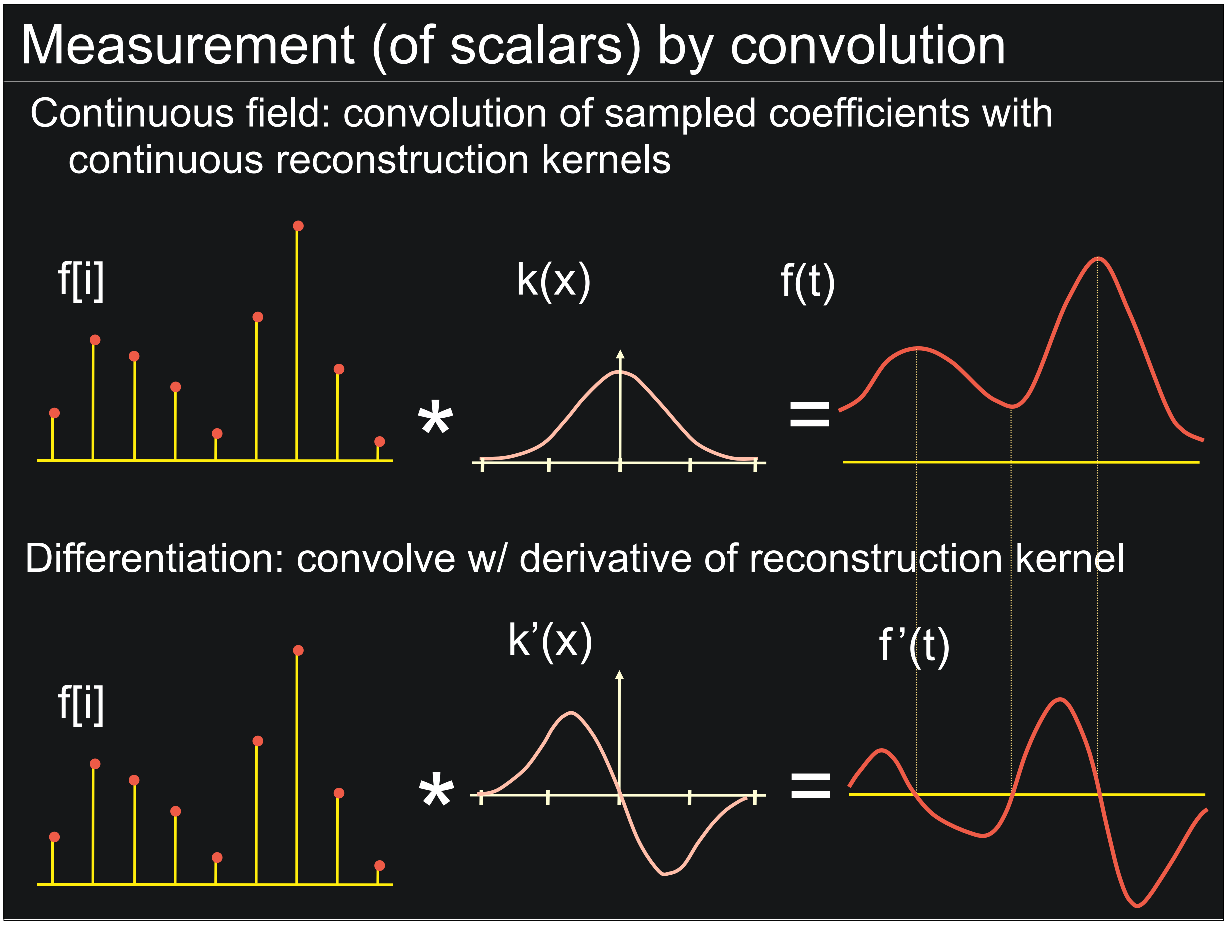


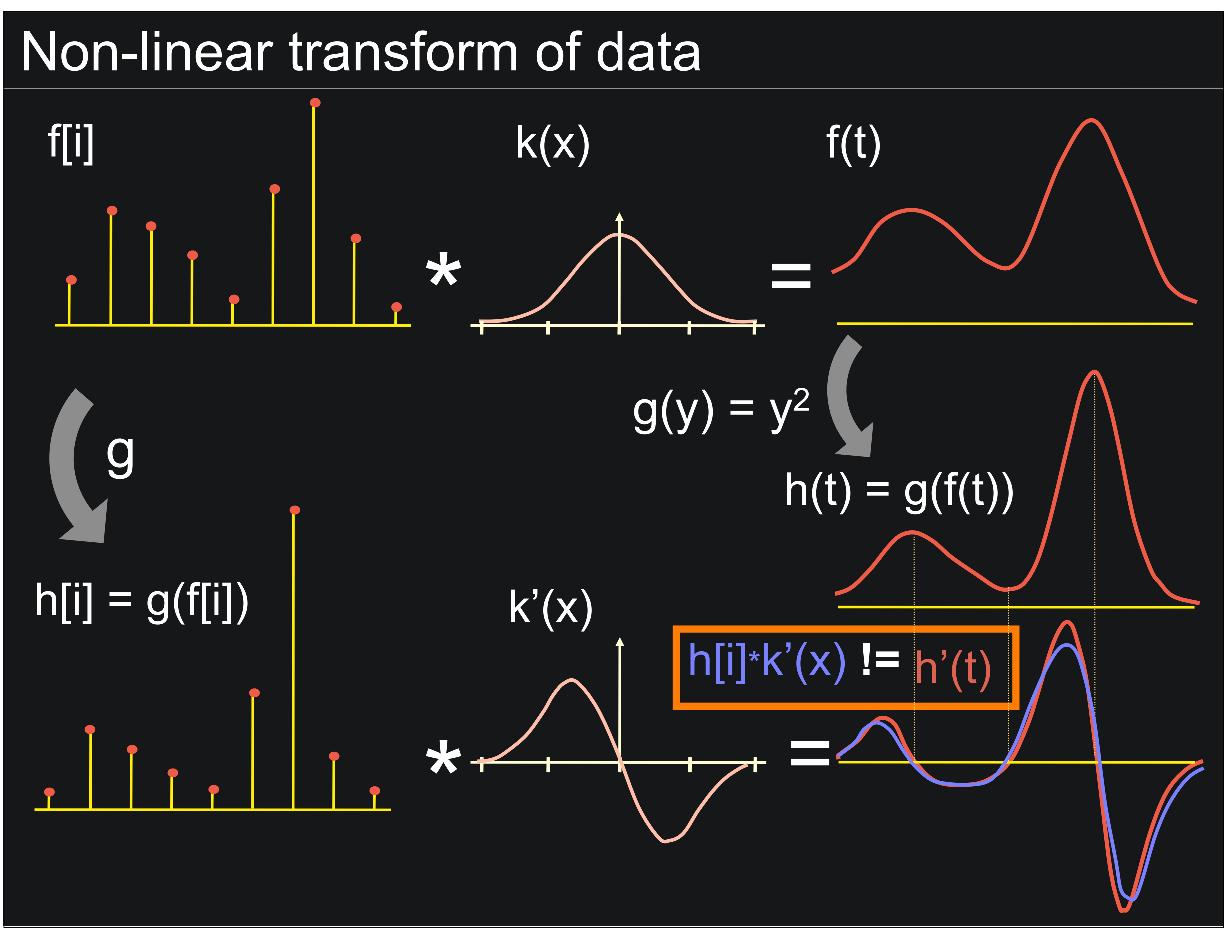


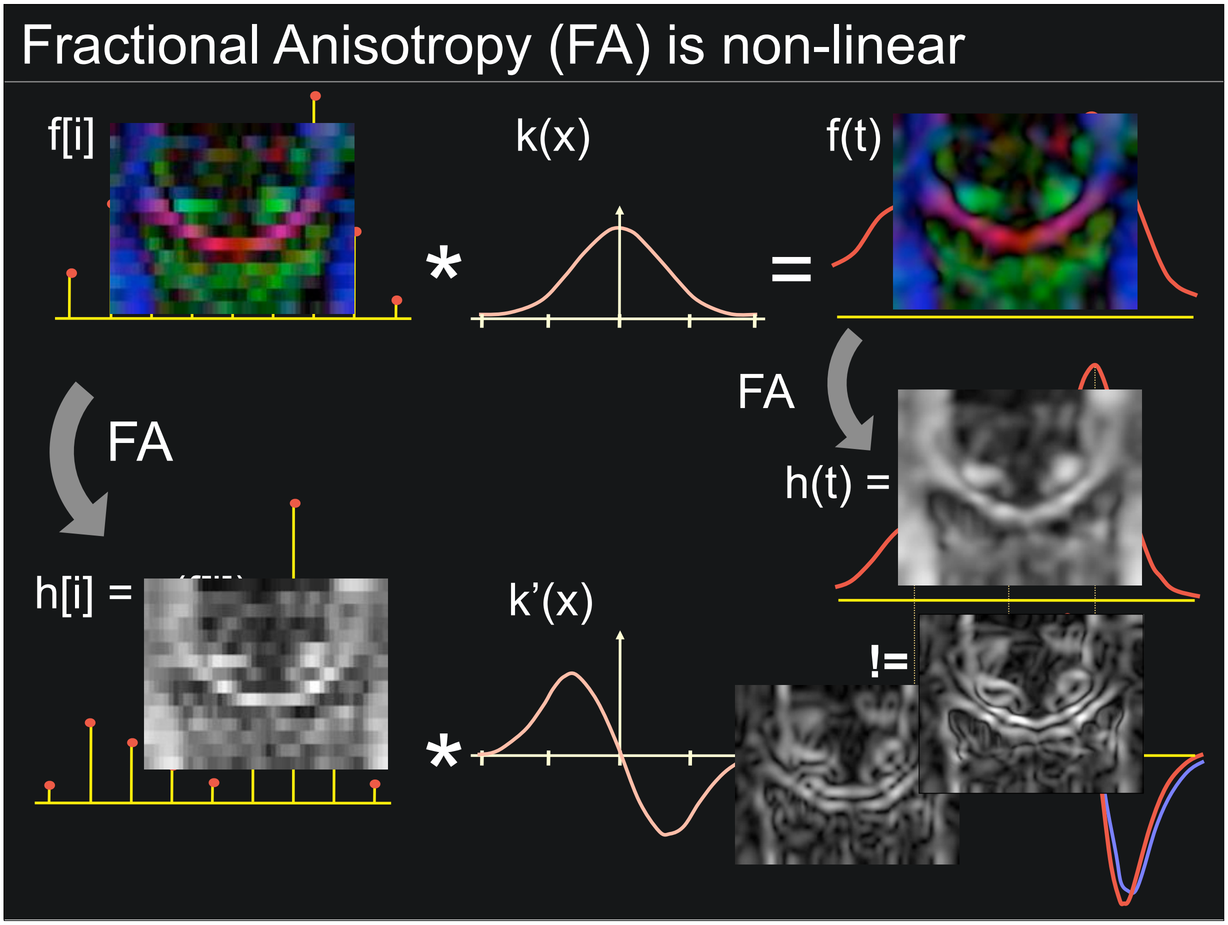
- Goal: automatically delineate large-scale white matter structures
- "Sulci for white matter"
- Ridges, valleys: major paths and interfaces in between
- **Shape**, not connectivity
- Smith et. al. "Tract-Based Spatial Statistics" NeuroImage '06

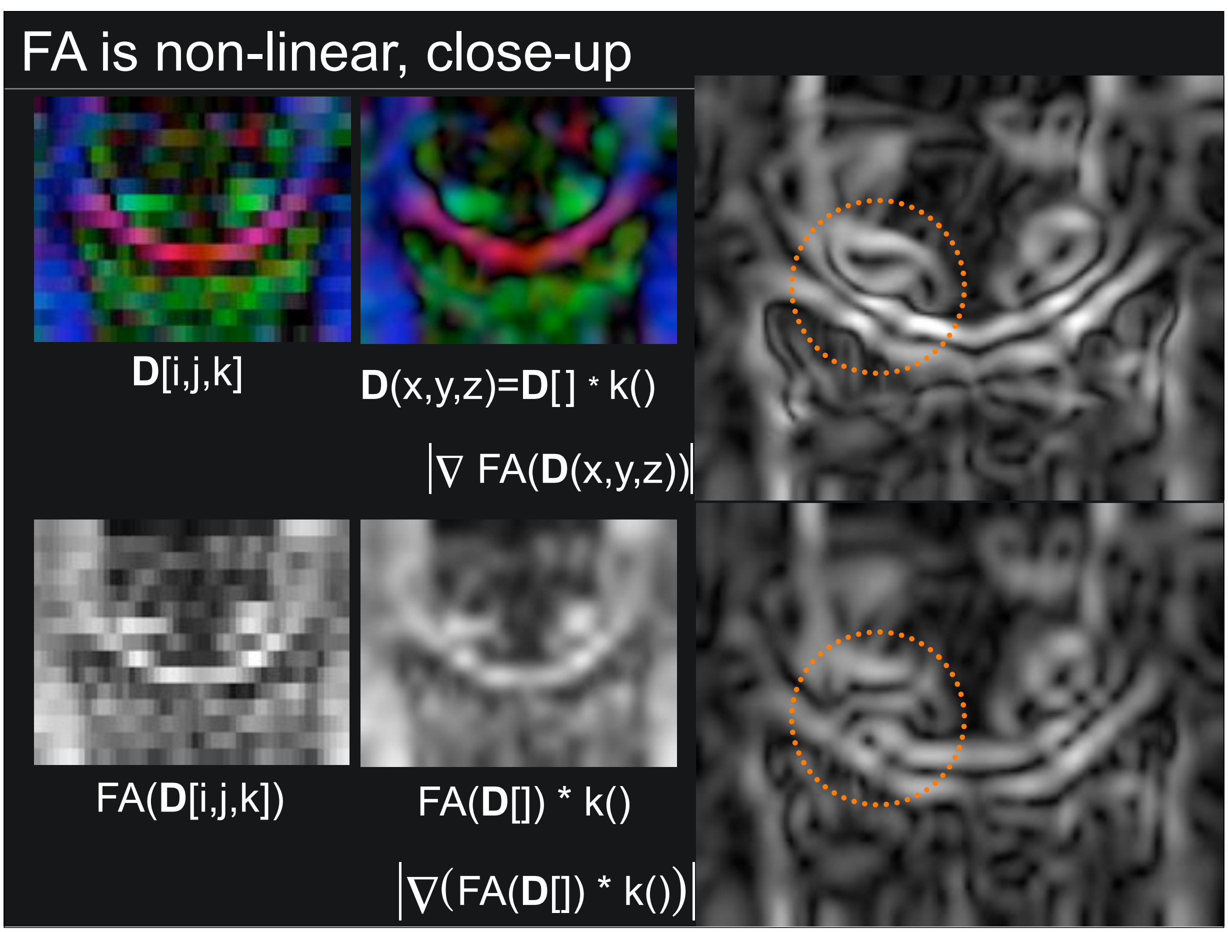





Measurement (of scalars) by convolution
Continuous field: convolution of sampled coefficients with continuous reconstruction kernels
f[i]
k(x)
f(t)
*
=
Differentiation: convolve w/ derivative of reconstruction kernel
f[i]
k'(x)
f'(t)
*
=


Non-linear transform of data
f[i]
k(x)
f(t)
*
=
g
g(y) = y²
h(t) = g(f(t))
h[i] = g(f[i])
k'(x)
h[i]*k'(x) != h'(t)
*
=


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

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

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

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

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

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

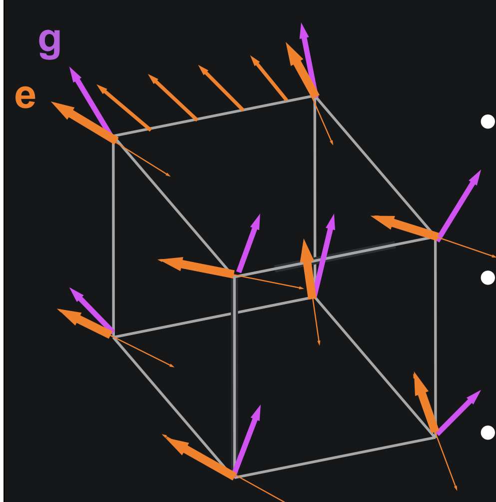
$$\nabla Q = \frac{\nabla S - \nabla J_2}{9}$$

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

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

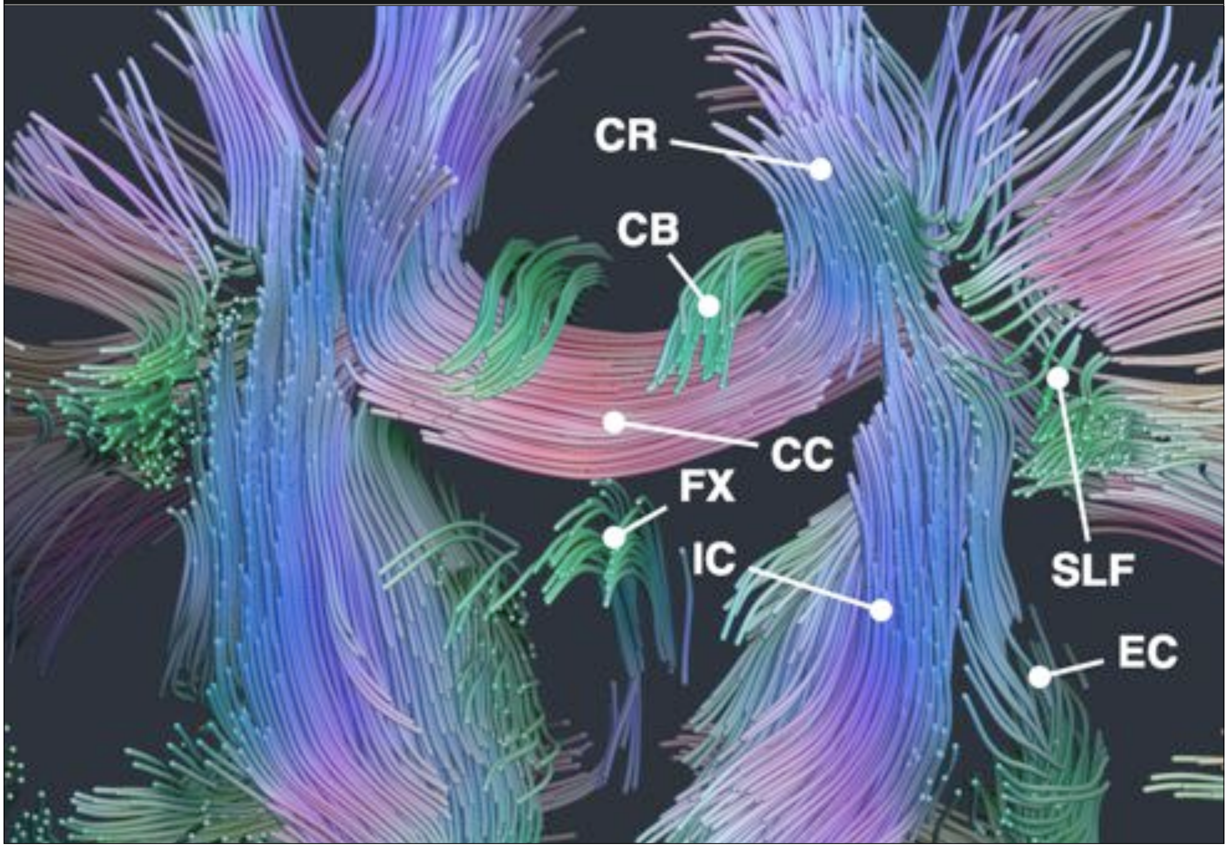
Hessian(FA) more complicated, but similarly derived

Modified Marching Cubes for Surfaces

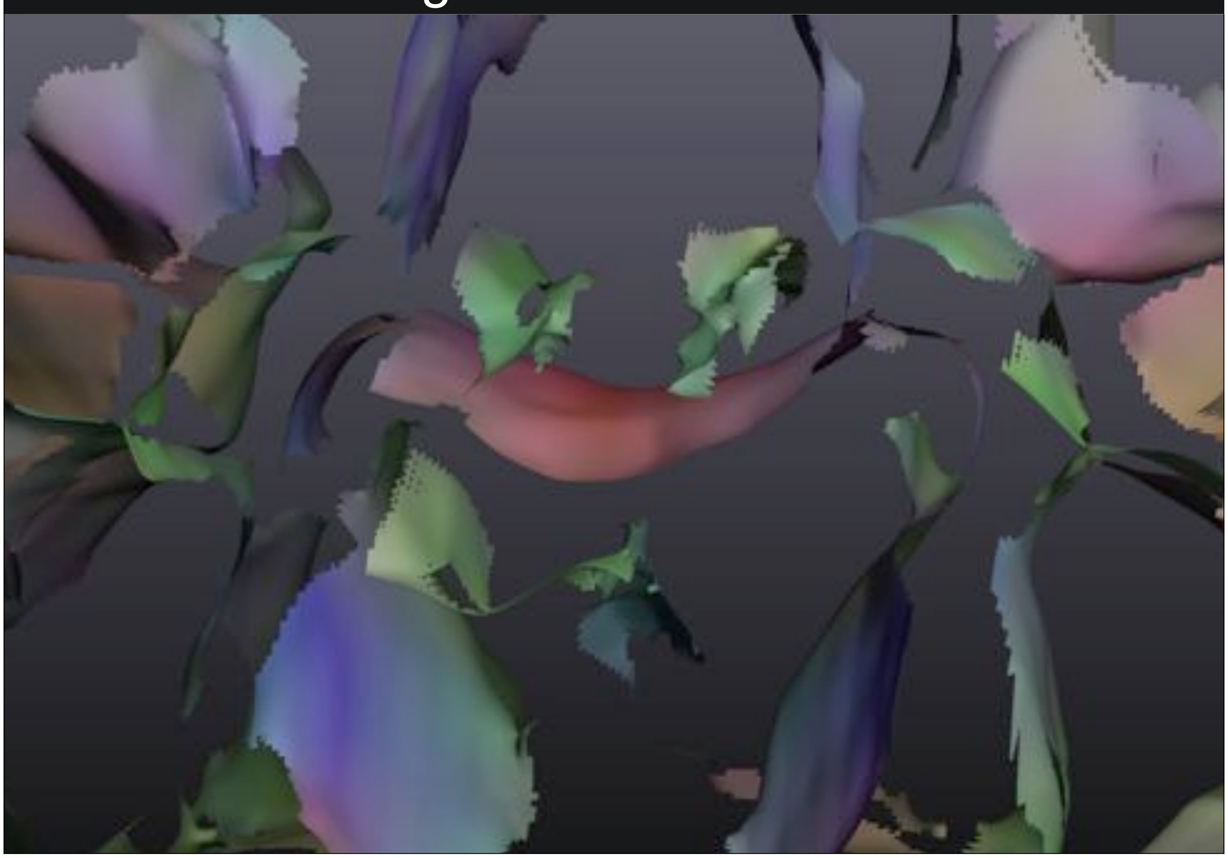


- Crease surface is isosurface (zero-crossing) of $\mathbf{g}\cdot\mathbf{e}_i$, but…
- Eigenvectors lack sign: enforce intra-voxel sign consistency
- Propagate eigenvector at one corner to all others
- $\mathbf{g}\cdot\mathbf{e}$ dot products, then MC case table
- Schultz et al. TVCG 09: smarter

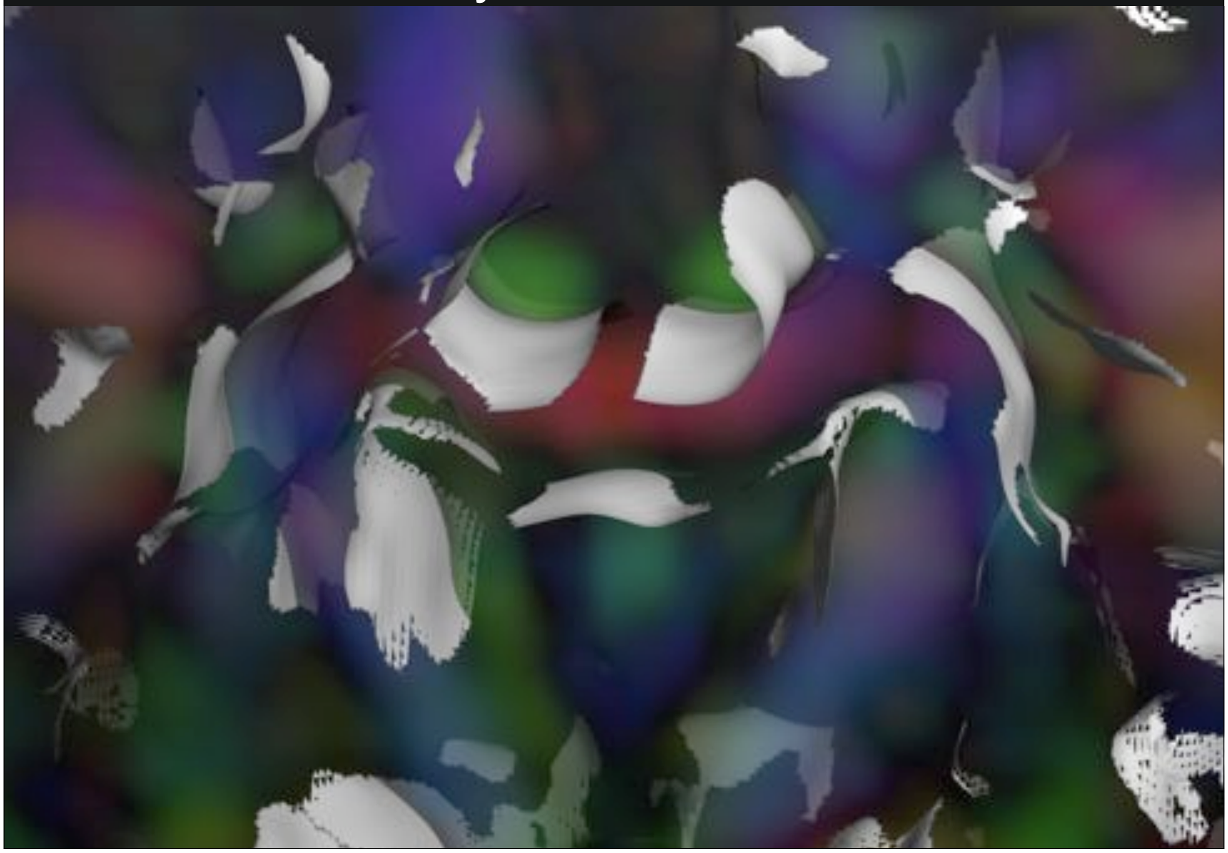
Coronal slab: tractography



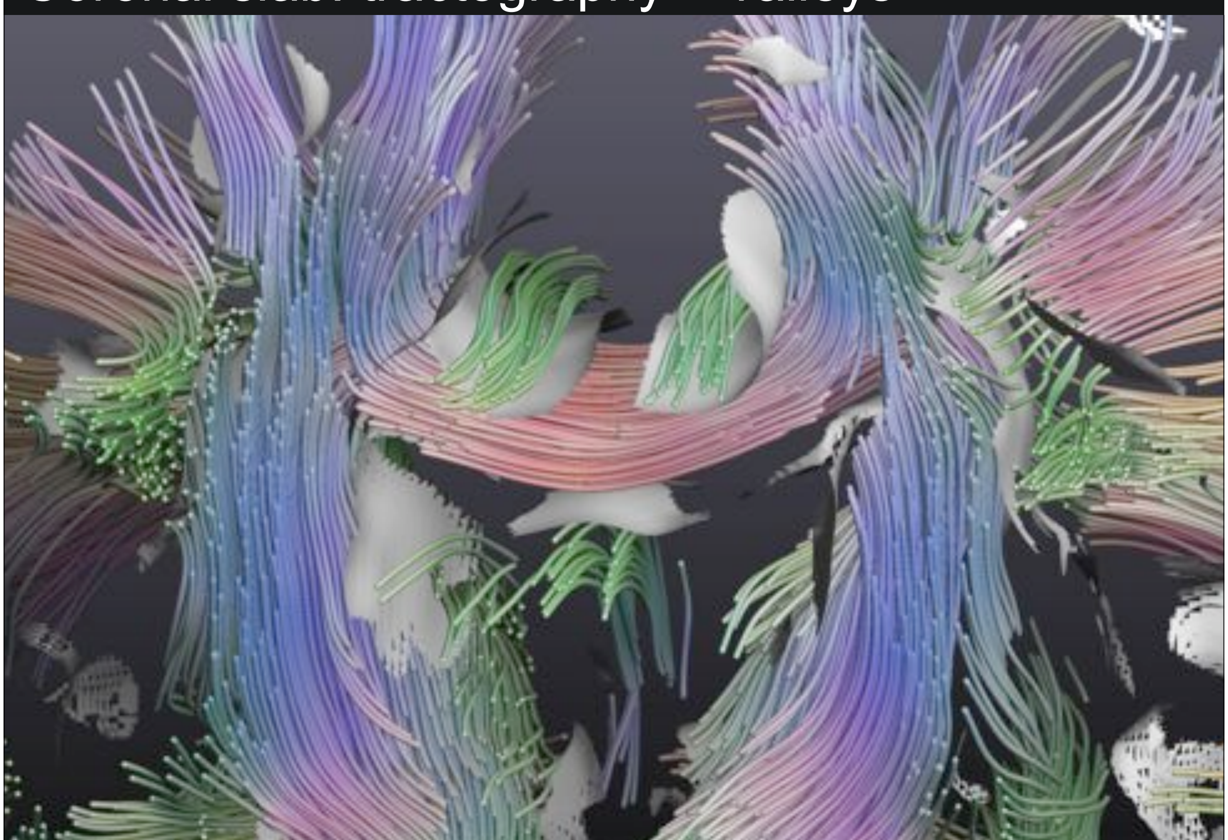
Coronal slab: ridge surfaces



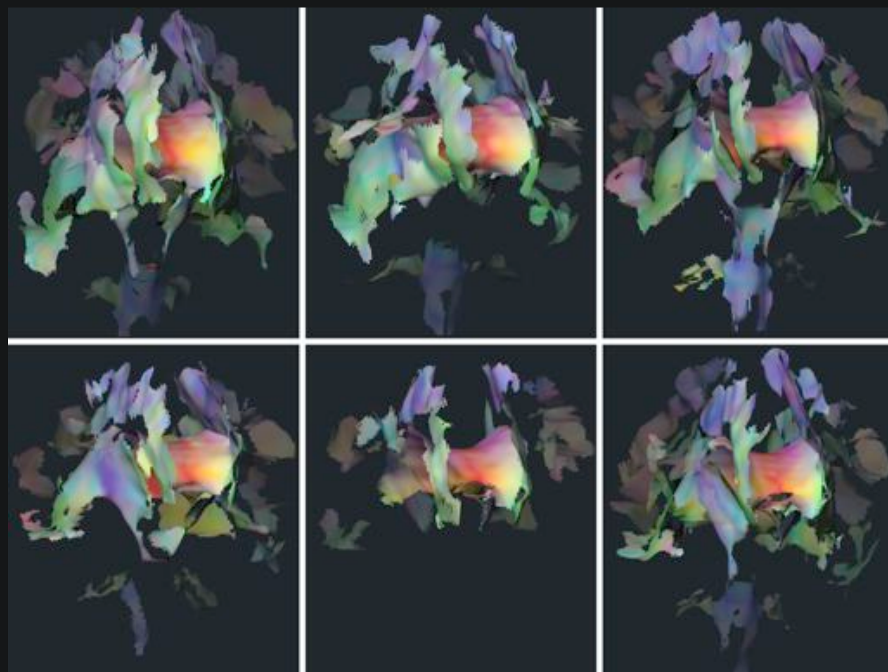
Coronal slab: valley surfaces



Coronal slab: tractography + valleys



6 cases, ridges w/ connected components

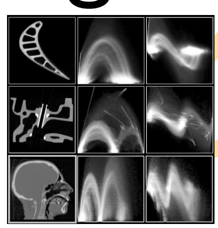


Non-linearity of FA makes it interesting to study

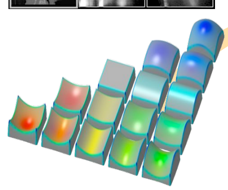
Problem of **scale-dependence** in feature extraction

Background

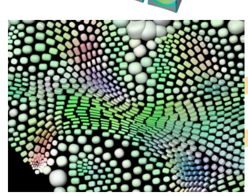
Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering (1998)



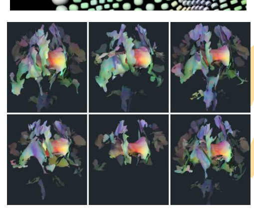
Curvature-Based Transfer Functions... (2003)



Diffusion Tensor Visualization with Glyph Packing (2006)



Delineating White Matter Structure in Diffusion Tensor MRI with Anisotropy Creases (2007)



Borrowing mathematical tools of Computer **Vision**

Leveraging **Hessian** for better local structure

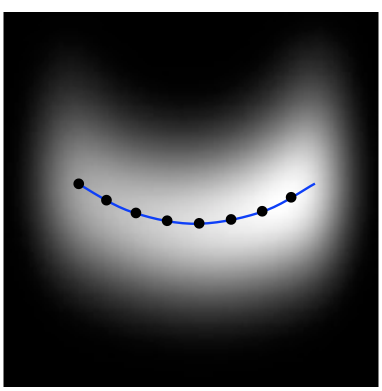
Using energy-minimizing **particle systems** to capture 3D structure

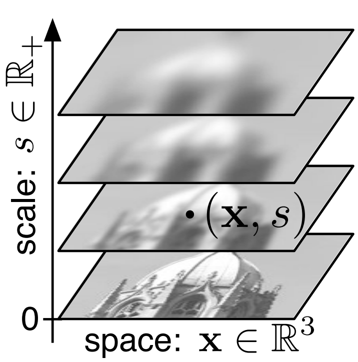
Visualization as tool for creating & evaluating methods of **anatomic feature** sampling & extraction

Scale-Space Particles (SSP)

- Basic Idea: use particle systems to **sample** crease features, and do so in **scale-space**
- Scale-space: process image and all blurrings of it
- Works in continuous domains of space and scale (no Marching-Cubes type discretization)
- Simple to implement: teach particles rules, let them go
- Variety of anatomic features could be creases (ridges & valleys, lines & surfaces)
- Biomedical research applications: group studies of brain, clinical quantification of anatomic structure, high-throughput image-based phenotyping

Method Overview





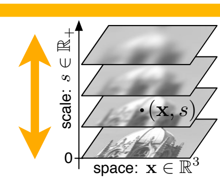
- Particles move/live/die to lower total system energy:

$$\underset{\{(\mathbf{x}_i,s_i)\},N}{\operatorname{argmin}} \; \mathscr{E} = \underset{\{(\mathbf{x}_i,s_i)\},N}{\operatorname{argmin}} \; (1-\alpha)\sum_{i=1}^{N} E_i + \frac{\alpha}{2}\sum_{i,j=1}^{N} E_{ij}$$

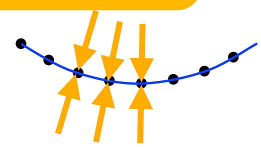
  - Inter-particle energy E_{ij} : maintains ~uniform sampling distance
  - Image-particle energy E_i : draws particles towards **scale** of maximal feature strength
- Within an iteration, particles are spatially constrained (no energy) to stay within crease features

Methods components

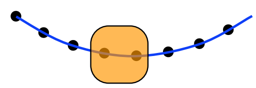
- Interpolation through scale



- Crease constraint enforcement, particle-image energy



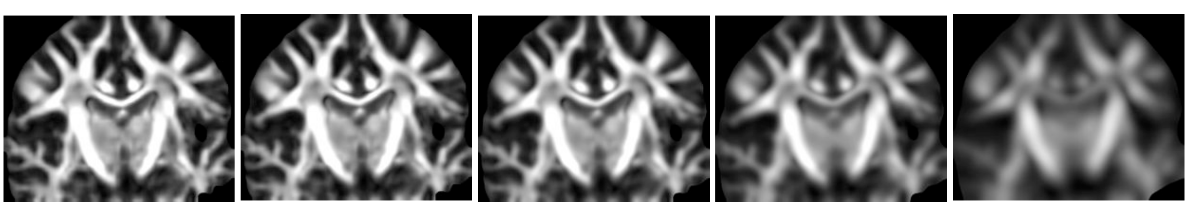
- Inter-particle energy



- System mechanics
- Particle visualization

Scale interpolation

- Want access to image at any blurring level



- Slow: re-blur at exactly desired blurring level
- Fast: interpolate between discrete pre-blurrings
- Spatial reconstruction: signal processing wisdom
- Reconstruction across scale, between pre-computed blurrings: Computer Vision wisdom (Lindeberg)

Diffusion = Gaussian blurring

- **Continuous domain**: *L*(*x*;*t*) is signal *L*(*x*) after time *t* of homogenous and isotropic diffusion

$$L(x;t) = (f(\cdot) \star g(\cdot;t))(x) = \int g(\xi;t) f(x-\xi) d\xi$$
$$g(\xi;t) = \exp(-\xi^2/2t)/\sqrt{2\pi t}$$

- 1st time derivative determined by 2nd space derivative

$$\frac{\partial L(x;t)}{\partial t} = \frac{1}{2}\frac{\partial^2 L(x;t)}{\partial x^2}$$

- What is analog for **Discrete domain**?

Lindeberg's Discrete Gaussian

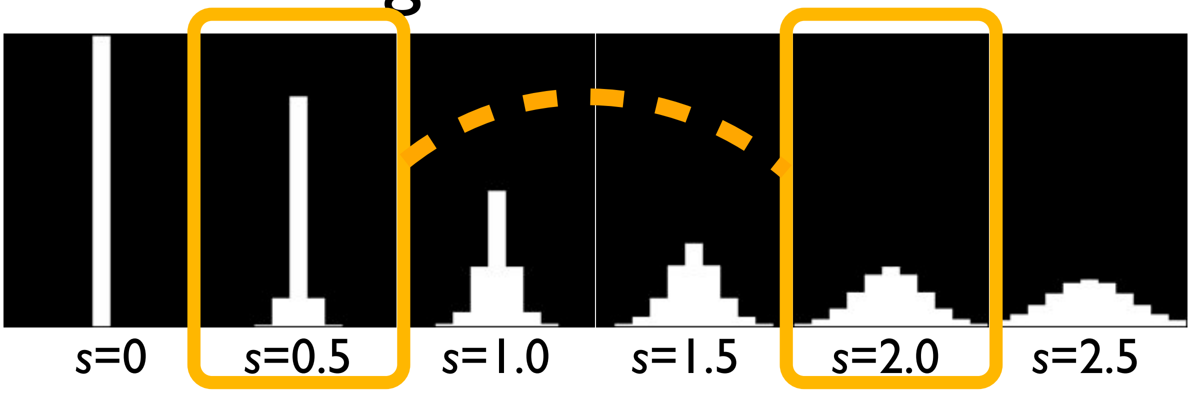
- "Scale-Space for Discrete Signals" IEEE PAMI 1990
- Formulated very nice analog to continuous Gaussian

$$L[i;t] = (f \star K[\cdot;t])[i] = \sum_n K[n;t] f[i-n]$$
$$K[n;t] = \exp(-t) I_n(t)\,;\; s = \sqrt{t} = \text{“}\sigma\text{”}$$

- $I_n(t)$ modified Bessel function of order *n*

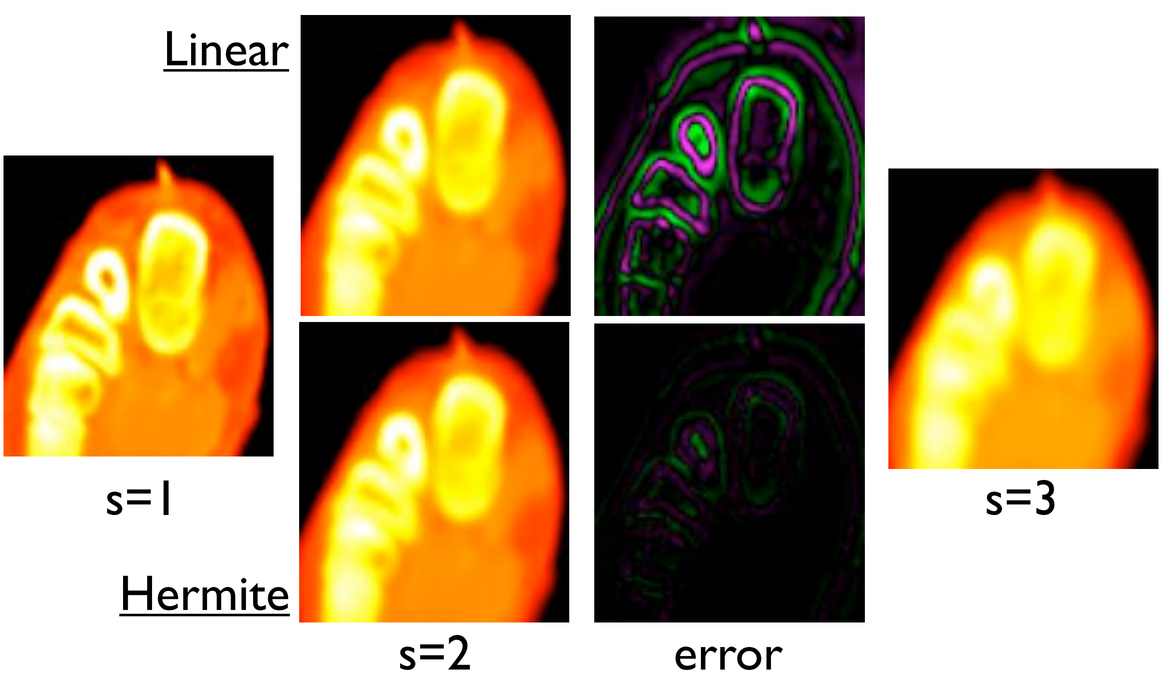
$$\frac{\partial K[i;t]}{\partial t} = (K[\cdot;t] \star \begin{bmatrix}1 & -2 & 1\end{bmatrix})[i]$$
$$\Rightarrow \frac{\partial L[i;t]}{\partial t} = \frac{1}{2}(L[\cdot;t] \star \begin{bmatrix}1 & -2 & 1\end{bmatrix})[i]$$
$$\Rightarrow \frac{\partial L[i;s^2]}{\partial s} = s\,(L[\cdot;s^2] \star \begin{bmatrix}1 & -2 & 1\end{bmatrix})[i]$$

Lindeberg's Discrete Gaussian

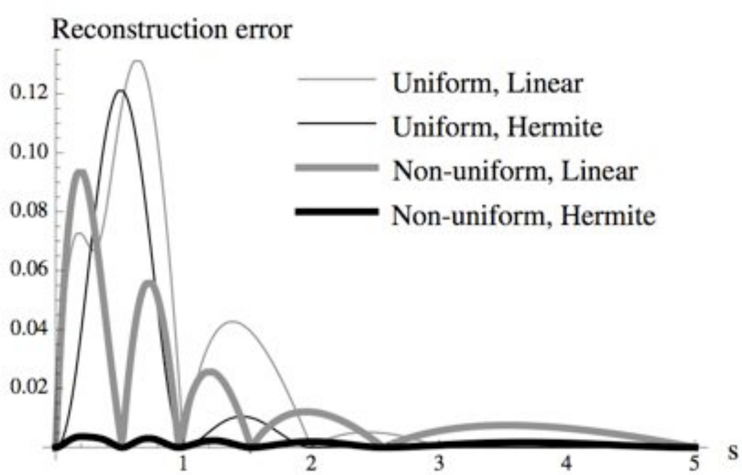


- **Not** same as sampling continuous Gaussian
- $s = 0 \rightarrow [\ldots 0\ \ 0\ \ 1\ \ 0\ \ 0 \ldots]$, not $[\ldots 0\ \ 0\ \ +\infty\ \ 0\ \ 0 \ldots]$
- Automatically sums to unity
- Basis for scale interpolation: exact derivative along s
  → can do Hermite spline instead of linear blend

Scale Interpolation Accuracy



Scale Interpolation Accuracy



- Measured error as squared difference between interpolated K[] and true K[], summed over support
- Optimized scale sample locations by gradient descent on error

Hermite-spline scale interpolation makes scale-space practical for real-world 3D volumes

Constraint enforcement

- Translate feature definition into update in iterative constraint solver; (R)idge, (V)alley; (L)ine, (S)urface
- Based on Hessian eigenvectors $\mathbf{v}_i$, eigenvalues λ_i

	R L	R S	V L	V S
defined	$\mathbf{g}\cdot\mathbf{v}_2 = \mathbf{g}\cdot\mathbf{v}_3 = 0$	$\mathbf{g}\cdot\mathbf{v}_3 = 0$	$\mathbf{g}\cdot\mathbf{v}_1 = \mathbf{g}\cdot\mathbf{v}_2 = 0$	$\mathbf{g}\cdot\mathbf{v}_1 = 0$
λ sign	$\lambda_3 \le \lambda_2 < 0$	$\lambda_3 < 0$	$0 < \lambda_2 \le \lambda_1$	$0 < \lambda_1$
strength h	$-\tilde{\lambda}_2$	$-\tilde{\lambda}_3$	$\tilde{\lambda}_2$	$\tilde{\lambda}_1$
$\sim$ tangent $\mathbf{T}$	$\mathbf{v}_1\mathbf{v}_1^T$	$\mathbf{v}_1\mathbf{v}_1^T + \mathbf{v}_2\mathbf{v}_2^T$	$\mathbf{v}_3\mathbf{v}_3^T$	$\mathbf{v}_2\mathbf{v}_2^T + \mathbf{v}_3\mathbf{v}_3^T$

$$\mathbf{x} \leftarrow \mathbf{x} + c\dot{\mathbf{x}};\ \dot{\mathbf{x}} = (\mathbf{I} - \mathbf{T})\mathbf{g}$$

$\dot{\mathbf{x}} = 0 \Rightarrow$ On the feature

Accurate enough for small updates

Particle-Image Energy

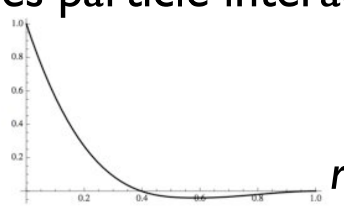
	RL	RS	VL	VS
defined	$\mathbf{g}\cdot\mathbf{v}_2=\mathbf{g}\cdot\mathbf{v}_3=0$	$\mathbf{g}\cdot\mathbf{v}_3=0$	$\mathbf{g}\cdot\mathbf{v}_1=\mathbf{g}\cdot\mathbf{v}_2=0$	$\mathbf{g}\cdot\mathbf{v}_1=0$
λ sign	$\lambda_3<\lambda_2<0$	$\lambda_3<0$	$0<\lambda_2<\lambda_1$	$0<\lambda_1$
strength h	$-\tilde{\lambda}_2$	$-\tilde{\lambda}_3$	$\tilde{\lambda}_2$	$\tilde{\lambda}_1$
$\sim$ tangent $\mathbf{T}$	$\mathbf{v}_1\mathbf{v}_1^T$	$\mathbf{v}_1\mathbf{v}_1^T+\mathbf{v}_2\mathbf{v}_2^T$	$\mathbf{v}_3\mathbf{v}_3^T$	$\mathbf{v}_2\mathbf{v}_2^T+\mathbf{v}_3\mathbf{v}_3^T$

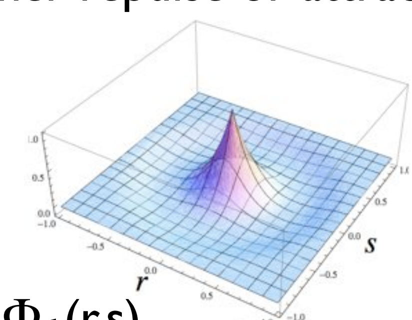
- Derivatives naturally go to 0 as scale increases
- Scale-normalized derivatives $$\begin{aligned}\tilde{\nabla} f(\mathbf{x},s) &= s\,\nabla f(\mathbf{x},s)\\ \tilde{\mathbf{H}} f(\mathbf{x},s) &= s^2\,\mathbf{H} f(\mathbf{x},s)\end{aligned}$$
- Particle-Image energy $E_i = -\gamma h(\mathbf{x}_i, s_i)$
- Draws particles towards scale at which feature appears strongest: feature localization along scale

Inter-Particle Energy

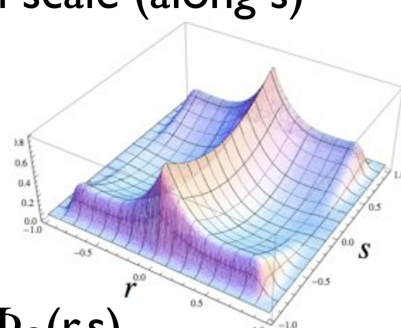
$$E_{ij} = \Phi(r_{ij}, s_{ij}) = \Phi\left(\frac{|\mathbf{x}_i - \mathbf{x}_j|}{\sigma_r}, \frac{s_i - s_j}{\sigma_s}\right)$$

- Potential function Φ(r,s) guides particle interaction
- Repulsion in space (along r)
- Role of potential well
- Either repulse or attract in scale (along s)





$\Phi_1(r,s)$



$\Phi_2(r,s)$

Particle Visualization

- Glyph indicates particle's location in space and scale, and shape of local Hessian
- Visualization to debug/evaluate feature extraction
- Tensors $\mathbf{D}_1$ and $\mathbf{D}_2$ visualized with tensor glyphs

$$\begin{aligned}
\lambda' &= \frac{(|\lambda_1| + |\lambda_2| + |\lambda_3|)}{10} \\
\lambda_i'' &= \frac{1}{\max(\lambda', |\lambda_i|)} \\
\mu_i &= \frac{\lambda_i''}{\max(\lambda_1'', \lambda_2'', \lambda_2'')} \\
\mathbf{D}_1 &= \sum_i \mu_i \mathbf{v}_i \otimes \mathbf{v}_i
\end{aligned}$$

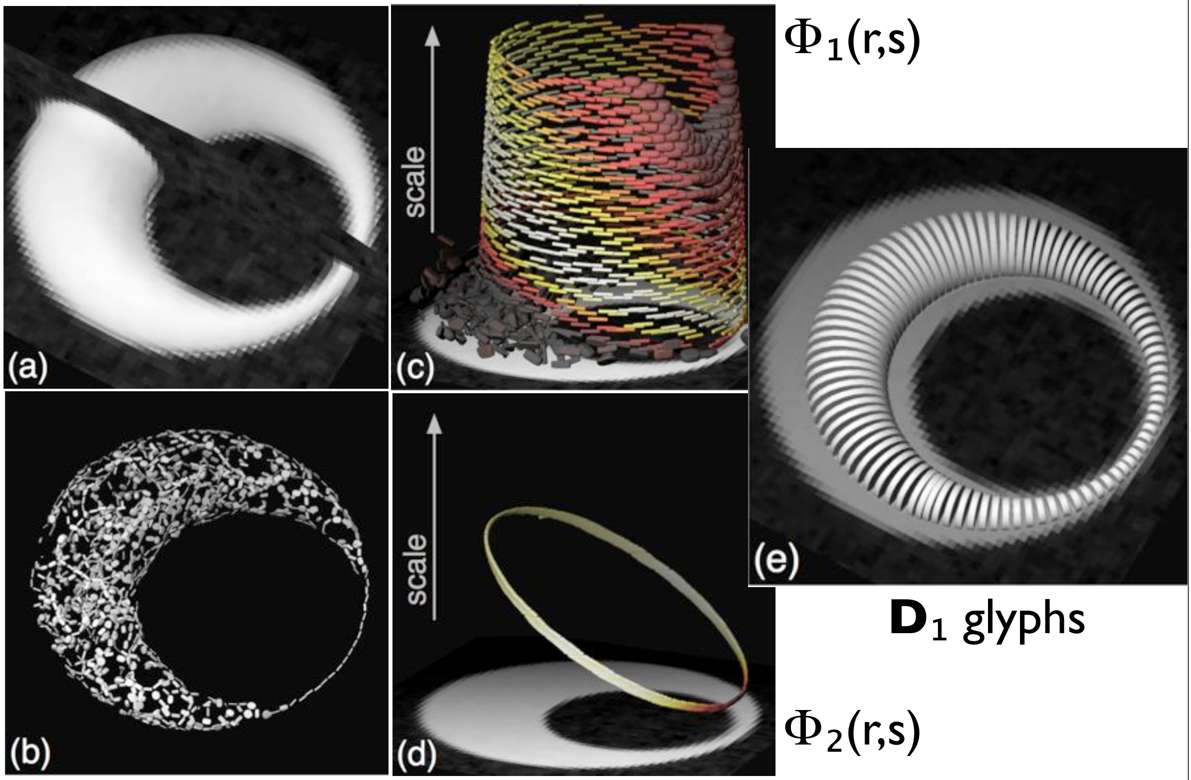
	R L	R S	V L	V S
defined	$\mathbf{g}\cdot\mathbf{v}_2 = \mathbf{g}\cdot\mathbf{v}_3 = 0$	$\mathbf{g}\cdot\mathbf{v}_3 = 0$	$\mathbf{g}\cdot\mathbf{v}_1 = \mathbf{g}\cdot\mathbf{v}_2 = 0$	$\mathbf{g}\cdot\mathbf{v}_1 = 0$
λ sign	$\lambda_3 \leq \lambda_2 < 0$	$\lambda_3 < 0$	$0 < \lambda_2 \leq \lambda_1$	$0 < \lambda_1$
strength h	$-\tilde{\lambda}_2$	$-\tilde{\lambda}_3$	$\tilde{\lambda}_2$	$\tilde{\lambda}_1$
$\sim$ tangent $\mathbf{T}$	$\mathbf{v}_1\mathbf{v}_1^T$	$\mathbf{v}_1\mathbf{v}_1^T+\mathbf{v}_2\mathbf{v}_2^T$	$\mathbf{v}_3\mathbf{v}_3^T$	$\mathbf{v}_2\mathbf{v}_2^T+\mathbf{v}_3\mathbf{v}_3^T$

$$\mathbf{D}_2 = \mathbf{T} + s_i(\mathbf{I} - \mathbf{T})$$

Results

Simple demo:
Isosurfaces vs Creases
Adding particles
(no scale-space)

Results



Results

Lung, Brain, from paper

Software

- Teem (http://teem.sf.net): collection of C libraries
- Weakened LGPL license
- Particle system in library called "pull"
  - No documentation except the source
- All interpolation (space and scale) in underlying library called "gage"

Discussion

- Not "visualization" per se
  - Particle systems from Graphics, Visualization
  - Glyphs from Visualization
  - Scale-space, Crease definition from Vision
  - Working on anatomical feature extraction
  - Visualization to put itself out of business
- Thank you!