Neuroimage Processing

Instructor: Moo K. Chung

mkhung@wisc.edu

Lecture 04-05.

Image Smoothing

September 25, 2009

Gaussian Kernel Smoothing

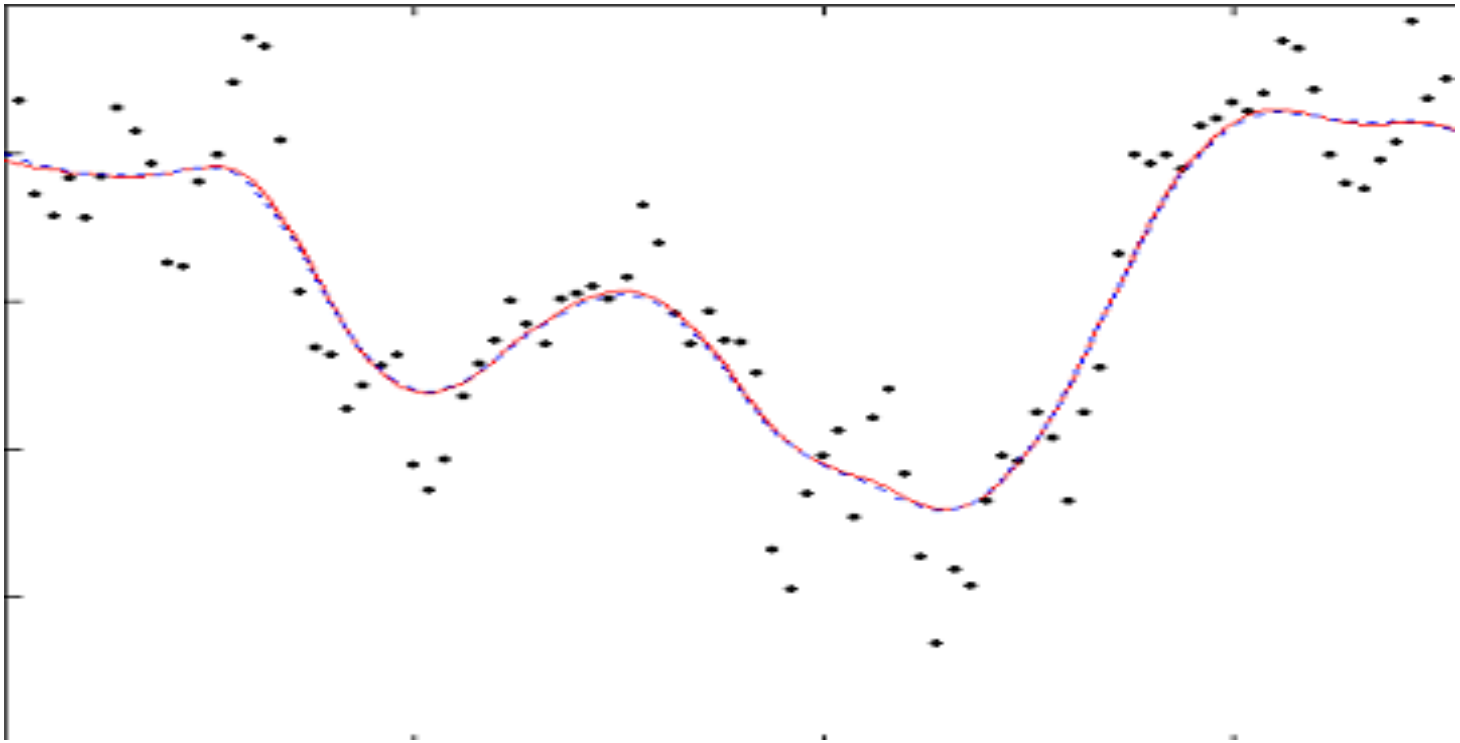
We will study basic properties of Gaussian kernel smoothing and numerical implementation issues.

Kernel Smoothing, Convolution, Linear Filter

$$Y(t) = \int K(t,s)X(s)\,ds.$$

output kernel input

$$Y(t) = K * X(t) = \int K(t-s)X(s)\,ds.$$

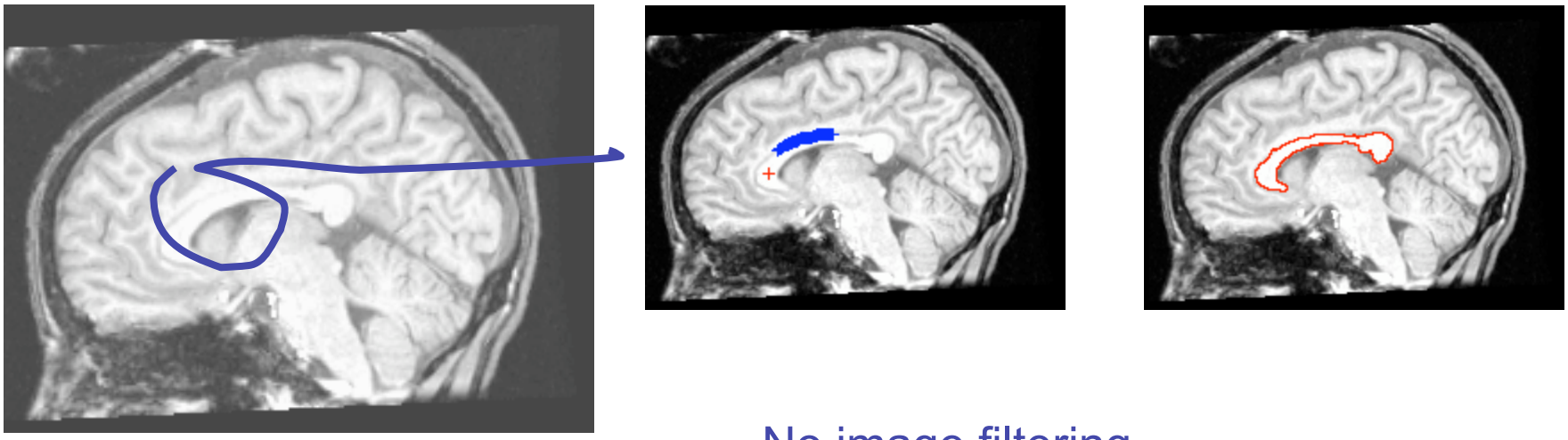


2D example

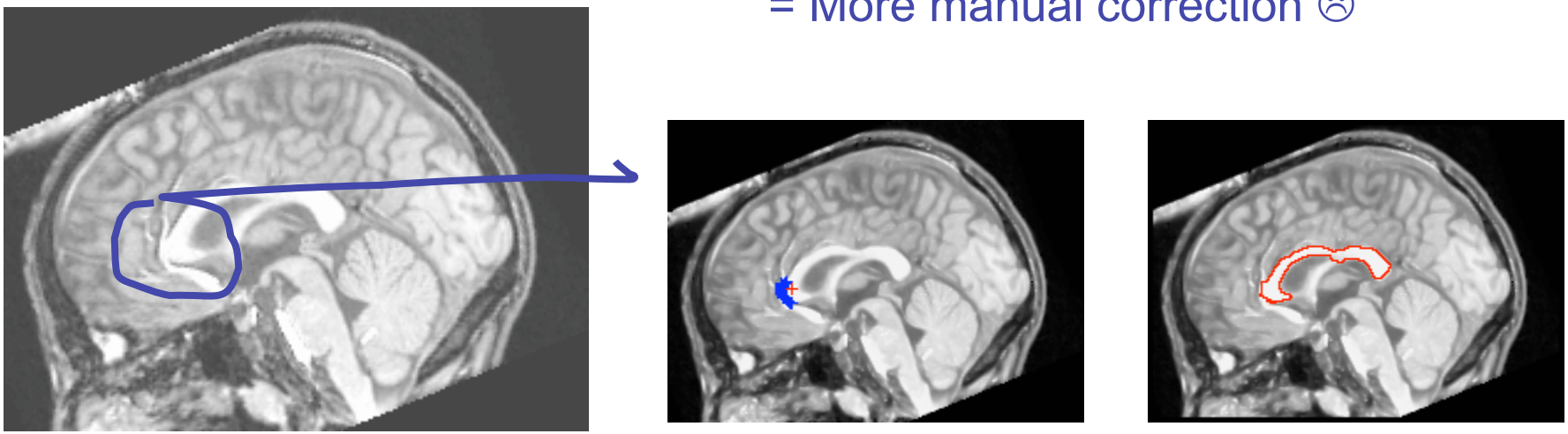

California State University
Hello
San Bernardino
World
Department of Computer Science


Hello
World

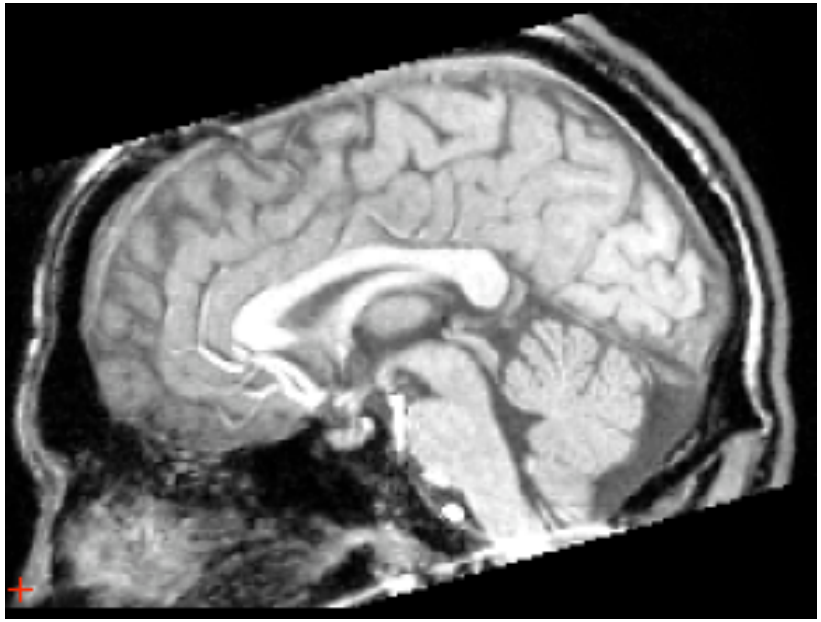
Motivation for image smoothing: Improve performance of PDE based segmentation - level set



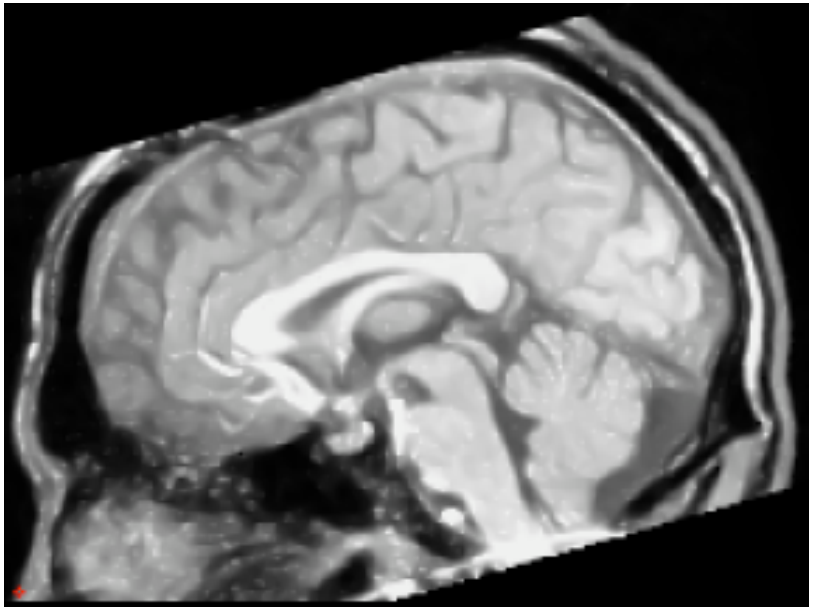
No image filtering
= More manual correction ☹



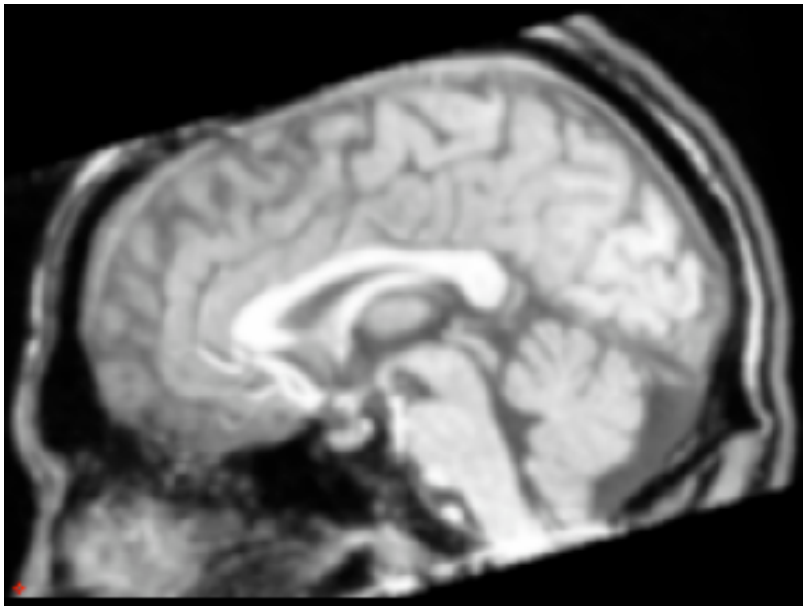
Malladi & Sethian's Min/Max Flow
This is basically a PDE smoother.



Original



Min/Max Flow



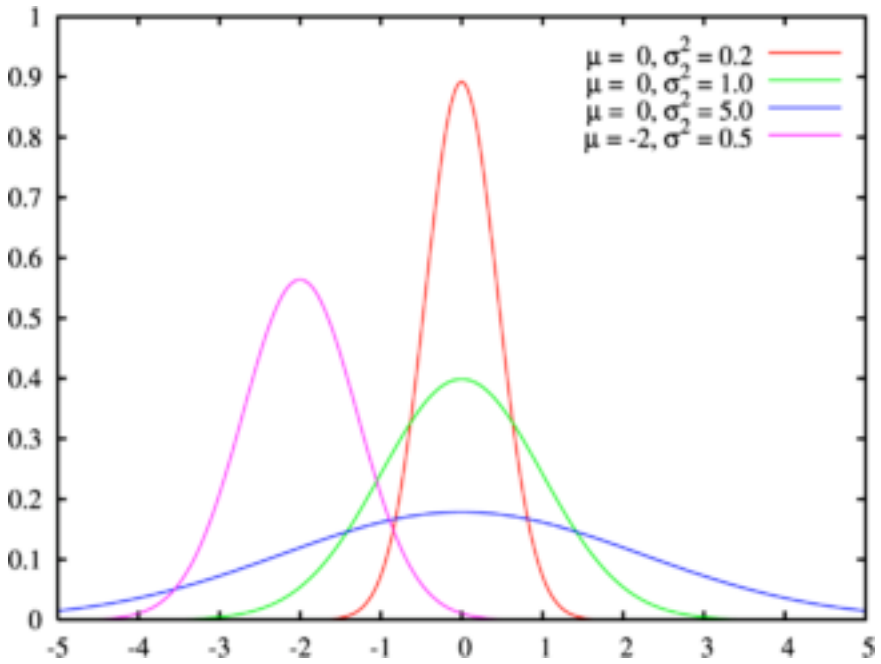
Gaussian

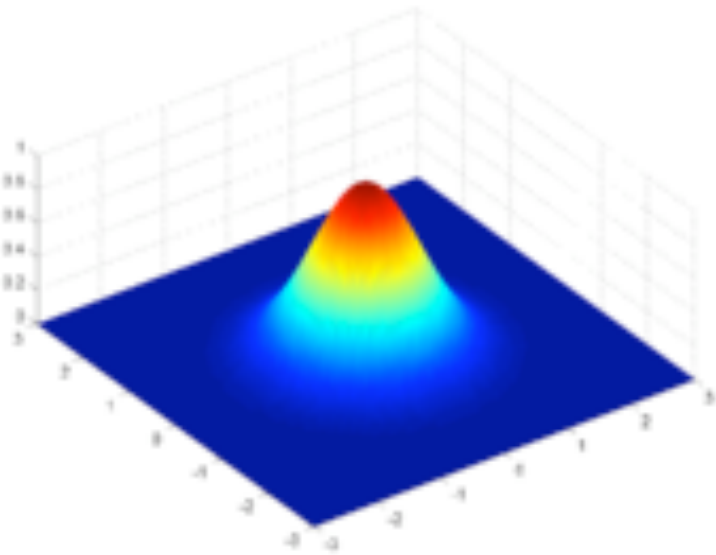
Thomas Hoffmann

Shape of kernel

Unimodal, Symmetric (Isotropic), normalized

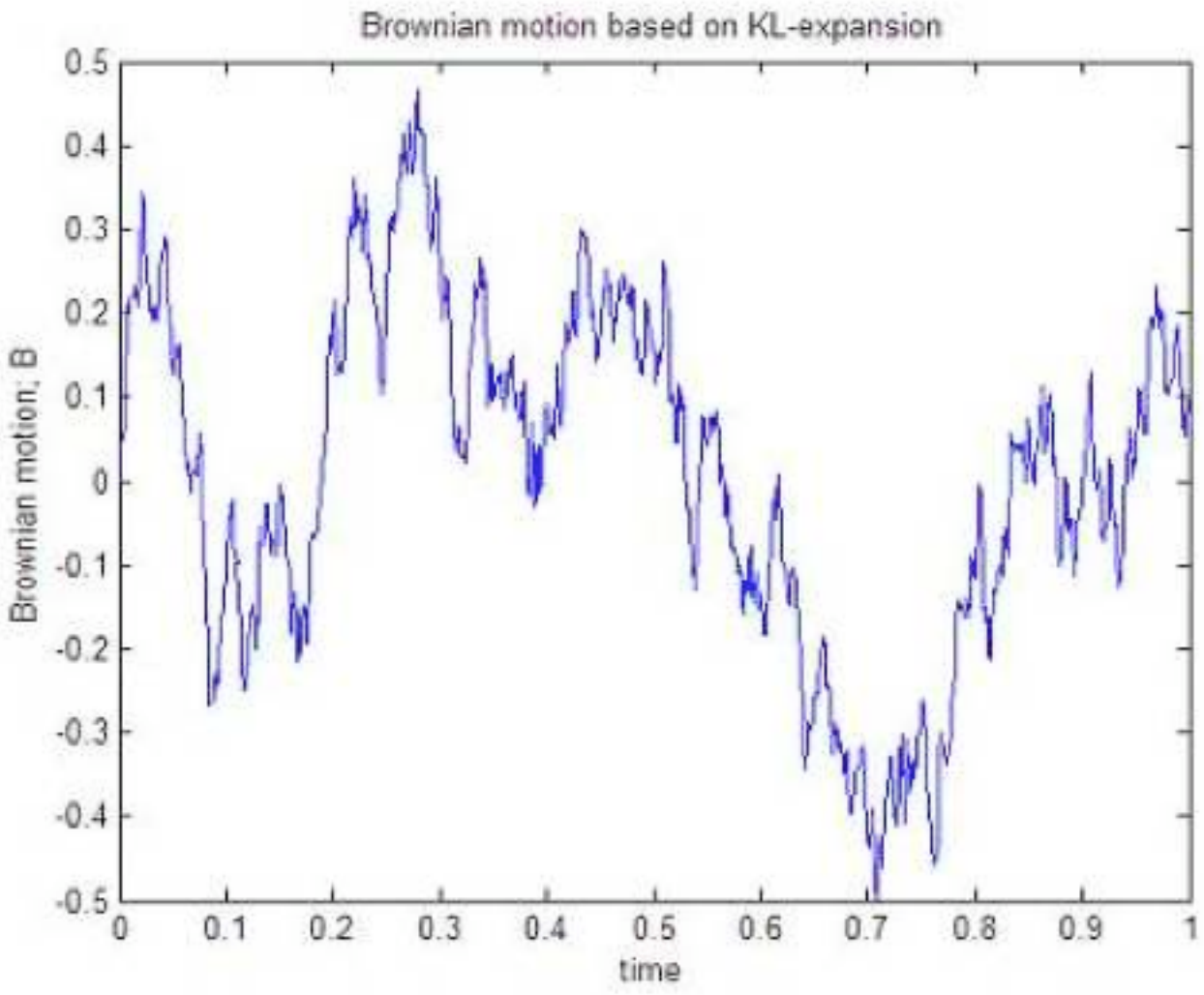
1D and 2D Gaussian kernel





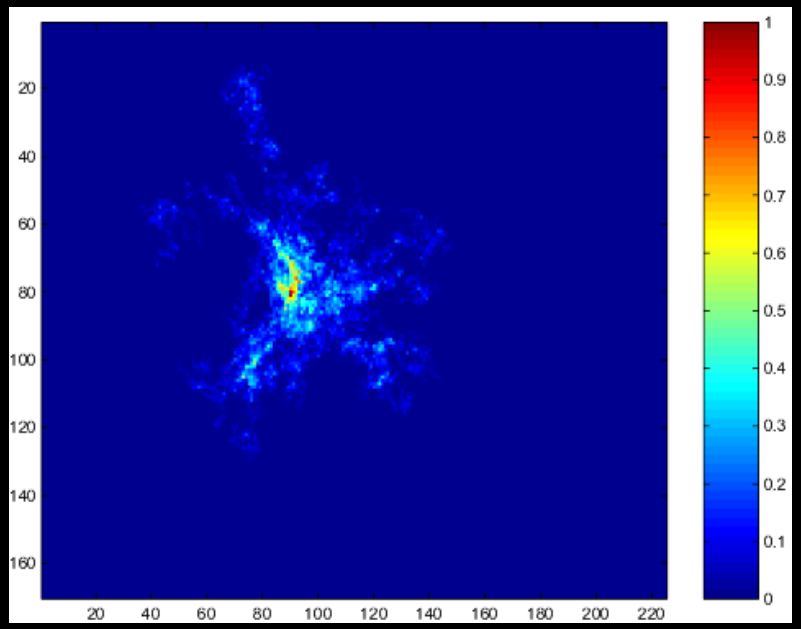
Quiz: The cross section of 2D Gaussian kernel ?

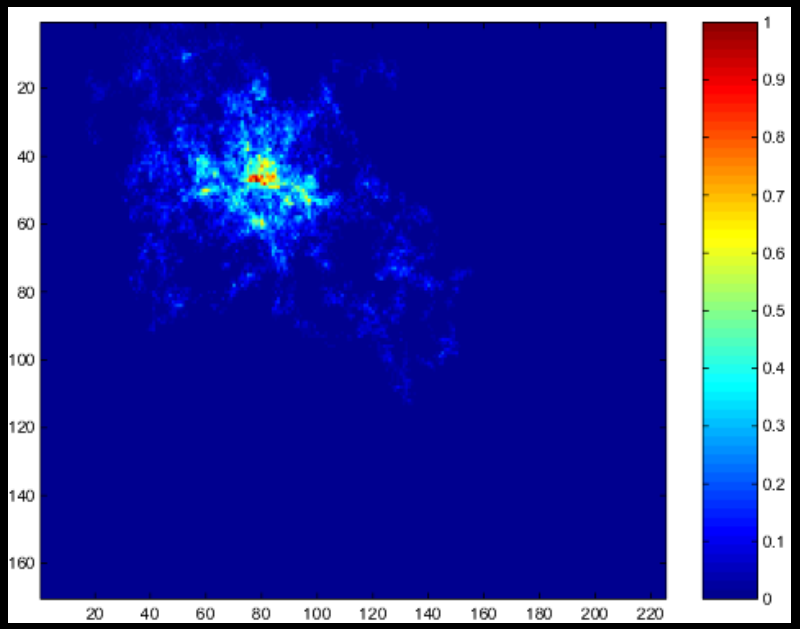
1D Brownian motion

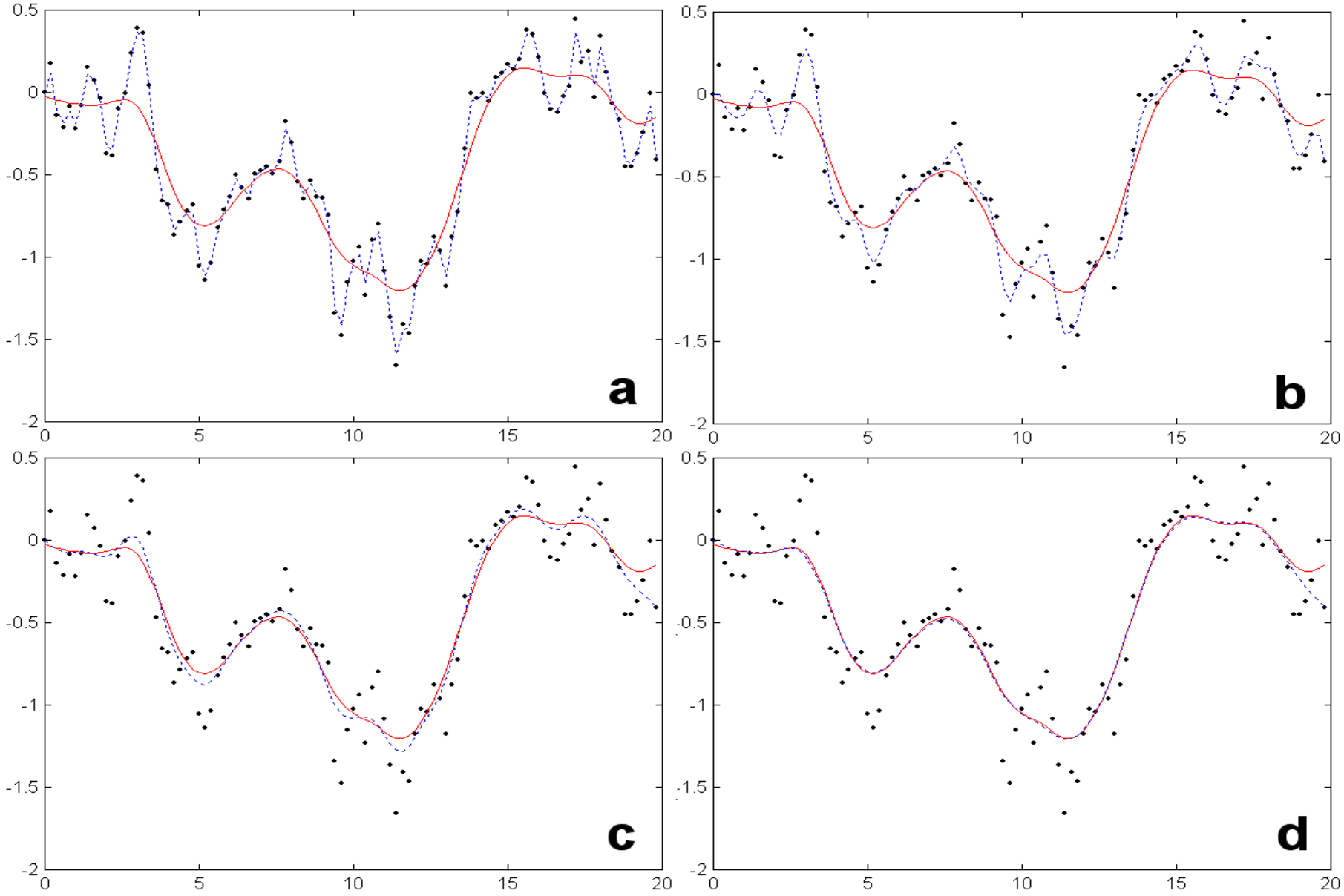


Brownian motion simulation ---> Gaussian kernel

$$\text{Probability} = \frac{\#\text{ random walk hitting a target voxel}}{\#\text{ total random walk}}$$







Red= Gaussian kernel smoothing

Blue = Diffusion smoothing after 5, 25 and 50 iterations

We will investigate the linear system

$$Y(t) = \int K(t,s)X(s)\,ds, \tag{4.1}$$

where K is called the kernel of the integral. Given the input signal X, Y represents the smoothed signal. We assume the kernel to be unimodal, isotropic and normalized as

$$\int K(t)\,dt = 1.$$

When the kernel K is isotropic, it has radial symmetry and should be invariant under rotation. So it has the form

$$K(t,s) = f(\|t-s\|)$$

for some function f. Since the kernel only depends on the difference of the arguments, with the abuse of notation, we can simply write K as

$$K(t,s) = K(t-s).$$

Then (4.1) can be written as

$$Y(t) = K * X(t) = \int K(t-s)X(s)\, ds$$

and it is called *kernel smoothing*. We may further assume K to be dependent on some parameter σ such that

$$\lim_{\sigma \to 0} K(t, s; \sigma) \to \delta(t-s),$$

the Dirac-delta function (Dirac, P.A.M., 1958). The Dirac-delta function is a special case of *generalized functions* or *distributions* (Gelfand and Shilov, 1964; Stakgold, 1997). It is usually defined as $\delta(t) = 0$ if $t \neq 0$ and $\int \delta(t)\, dt = 1$. This is also referred to as the *impulse function* in engineering literature.

An example of 1D kernel is an isotropic Gaussian kernel defined as

$$f(t) = \frac{1}{\sqrt{2\pi}} e^{t^2/2}.$$

Let's scale the kernel by the parameter σ:

$$K_\sigma(t) = \frac{1}{\sigma} f\Big(\frac{t}{\sigma}\Big).$$

This is the density function of the normal distribution with mean 0 and variance σ^2. The *bandwidth* σ defines the spread of kernel. The nD isotropic Gaussian kernel is defined as the product of n 1D kernel. Let $\mathbf{x} = (x_1, \cdots, x_n)' \in \mathbb{R}^n$. Then the nD kernel is given by

$$\begin{aligned} K_\sigma(\mathbf{x}) &= K_\sigma(x_1) K_\sigma(x_2) \cdots K_\sigma(x_n) \\ &= \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\Big(\frac{x_1^2 + x_2^2 + \cdots + x_n^2}{2\sigma^2}\Big) \end{aligned}$$

Gaussian kernel smoothing is probably the most widely used image smoothing technique in brain imaging. The numerical implementation utilizes the idea of kernel factorization.

4.3 Diffusion Equation

We show that the Gaussian kernel estimator $K_\sigma * X$ is the unique solution to a diffusion equation

$$\frac{\partial f}{\partial t} = f, f(\mathbf{x}, t = 0) = X(x)$$

with $t = \sigma^2/2$. With respect to the spherical coordinates, the n-dimensional Gaussian kernel is given by

$$K_\sigma(r) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\big[- \frac{r^2}{2\sigma^2}\big],$$

where $r = \sum_{i=1}^n x_i^2$. The Laplacian Δ in both the Cartesian and spherical coordinates at the fixed radius r are given by

$$\Delta = \sum_{i=1}^n \frac{\partial^2}{\partial x_i^2} = \frac{\partial^2}{\partial r^2} + \frac{n-1}{r}\frac{\partial}{\partial r}.$$

The algebraic manipulation can show that

$$\frac{1}{\sigma}\frac{\partial K_\sigma}{\partial \sigma} = \Delta K_\sigma. \tag{4.3}$$

Although (4.3) does not look like a diffusion equation, it is if we change the variables to $t = \sigma^2/2$. Using the differential $dt = \sigma d\sigma$, (4.3) transforms to

$$\frac{\partial K_\sigma}{\partial t} = \Delta K_\sigma.$$

By applying the convolution on both sides, we get

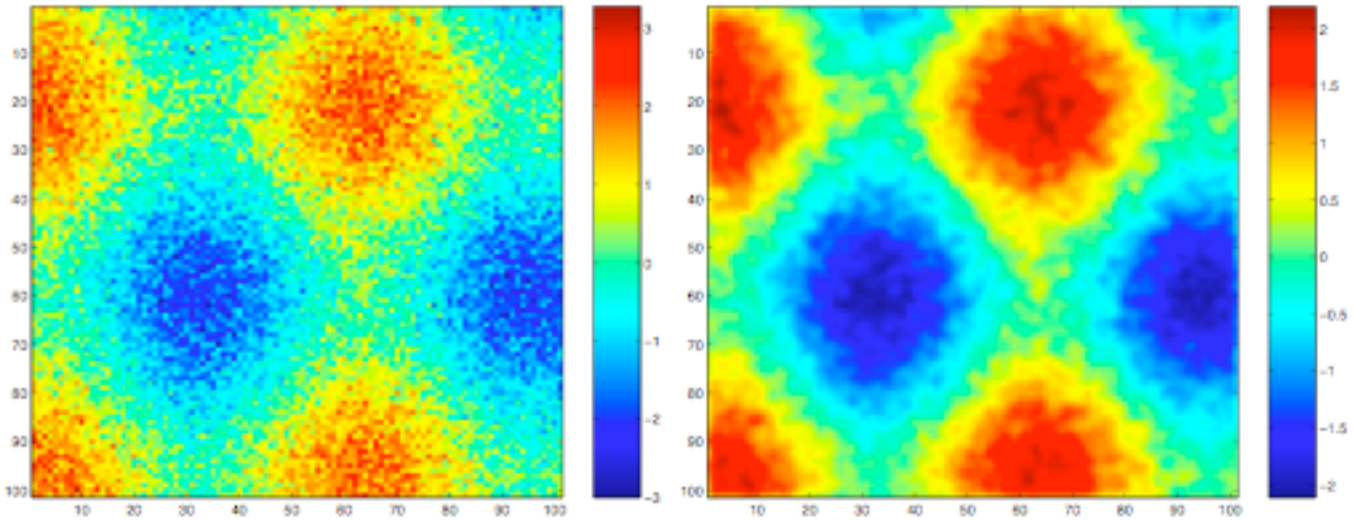
$$\frac{\partial K_\sigma * X(\mathbf{x})}{\partial t} = \Delta\big[K_\sigma * X(\mathbf{x})\big].$$

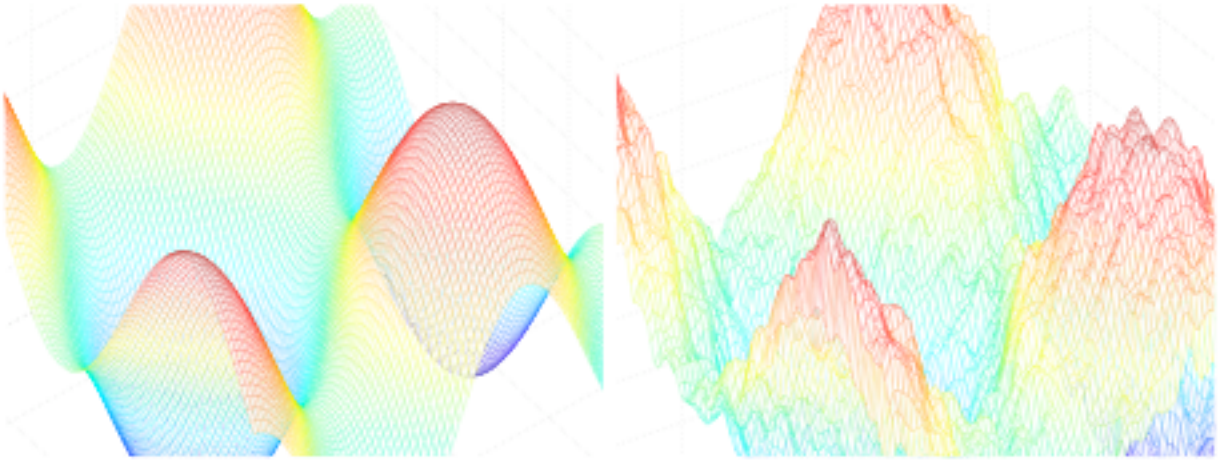
Hence $K_\sigma * X(\mathbf{x})$ is a solution of an isotropic heat equation

$$\frac{\partial f}{\partial t} = \Delta f \tag{4.4}$$

with initial condition $f(\mathbf{x}, 0) = X(\mathbf{x})$. Note that $\lim_{t \to 0} K_\sigma * X(x) = X(x)$ so the solution satisfies the initial condition. So if we diffuse observation $Y(x)$ for time duration of $t = \sigma^2/2$. it should be equivalent to kernel smoothing with bandwidth σ. This is a remarkable result that has many important ramifications in various imaging applications. The solution to (4.4) is identical to Gaussian kernel smoothing estimate so it should inherit all the statistical properties of kernel smoothing estimator and vice versa.

2D simulation results





4.1 Kernel Smoothing Estimator

Given noise observation

$$X(t) = \mu(t) + \epsilon(t),$$

where ϵ is a mean zero random field and μ is unknown signal. Then we estimate μ via kernel smoothing

$$\widehat{\mu}(t) = K_H * X(t) = \int K_H(t-s)X(s)\, ds$$

Some asymptotic properties are

$$\lim_{H\to 0} \widehat{\mu}(t) = \int \delta(t-s)X(s)\, ds = X(t)$$

and

$$\lim_{H\to \infty} \widehat{\mu}(t) = 0.$$

Also

$$\mathbb{E}\widehat{\mu}(t) = K_H * \mu(t) \to \mu(t) \text{ as } H \to 0$$

The kernel estimator becomes more unbiased as $H \to 0$. Assuming $|\mu| \leq \infty$,

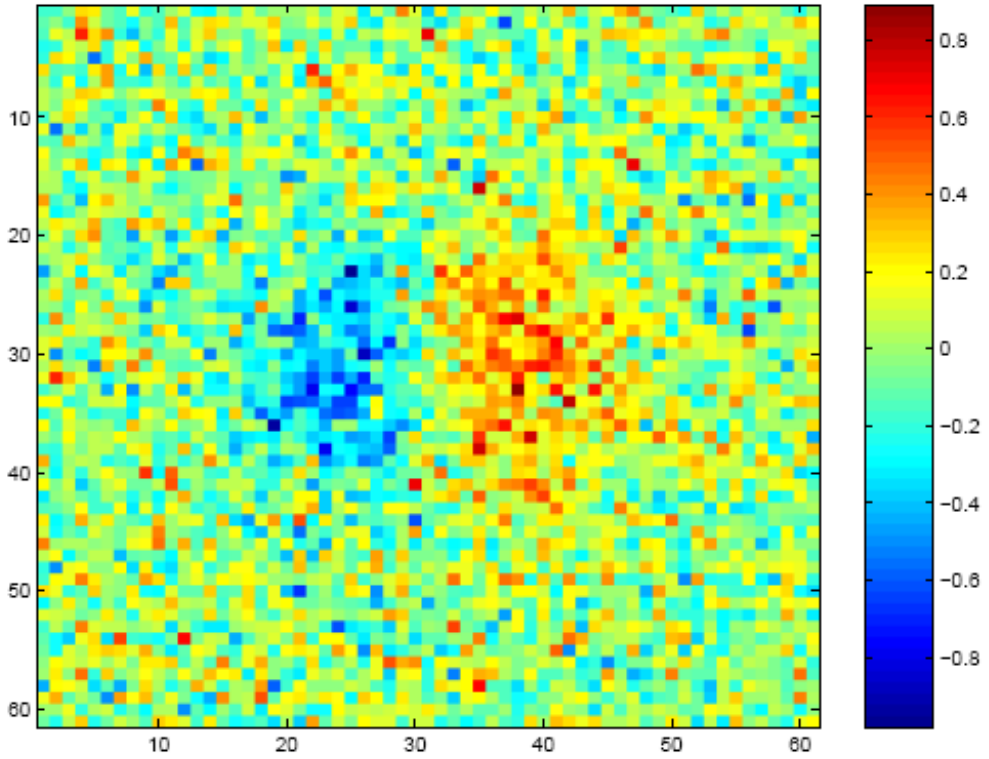
$$\mathbb{E}\widehat{\mu}(t) \leq \int K_H(t) \sup \mu(t)\, dt \leq \sup \mu(t).$$

Similarly we can bound from below so that

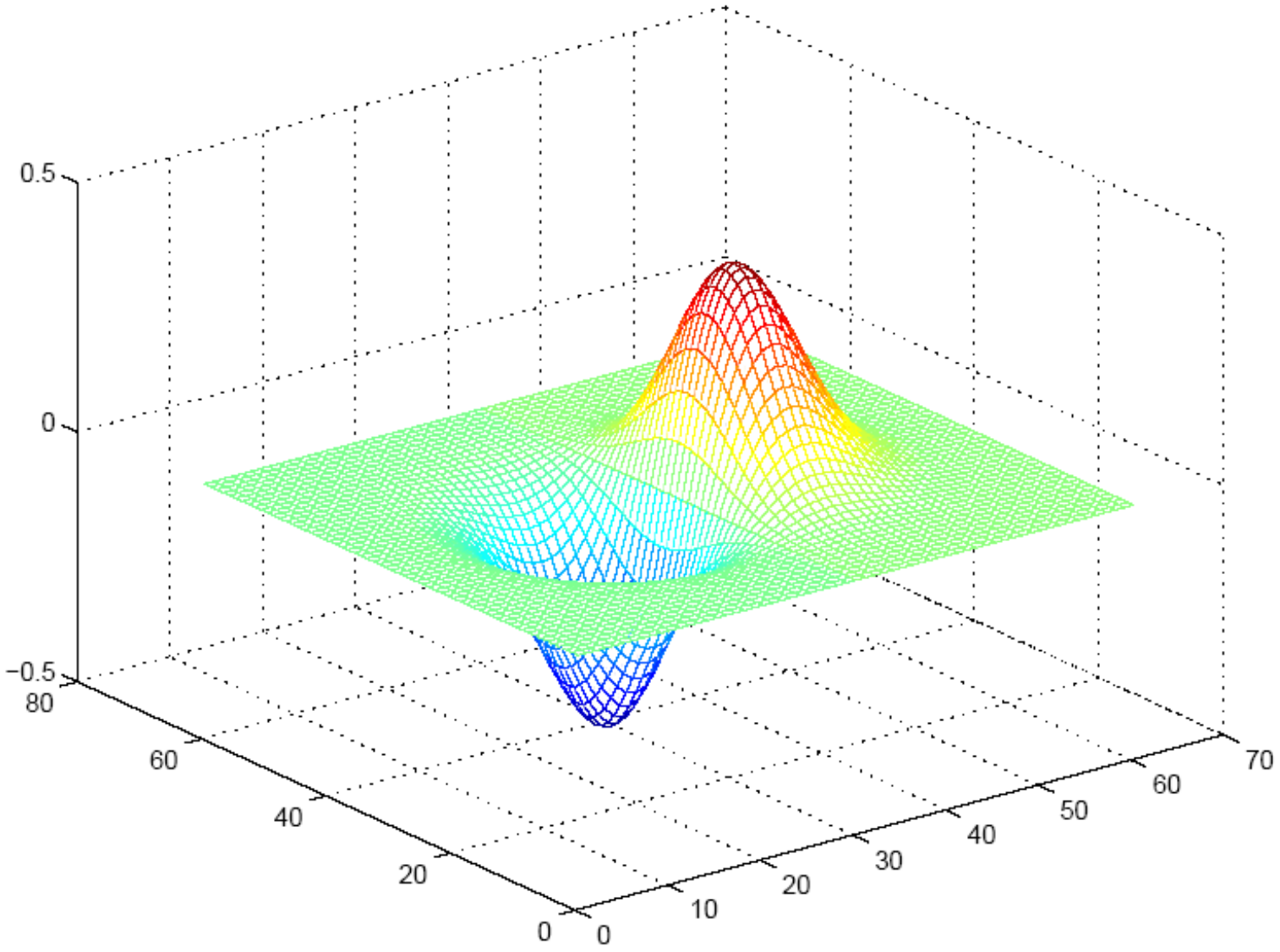
$$\inf \mu(t) \leq \mathbb{E}\widehat{\mu}(t) \leq \sup \mu(t).$$

Another interesting property is

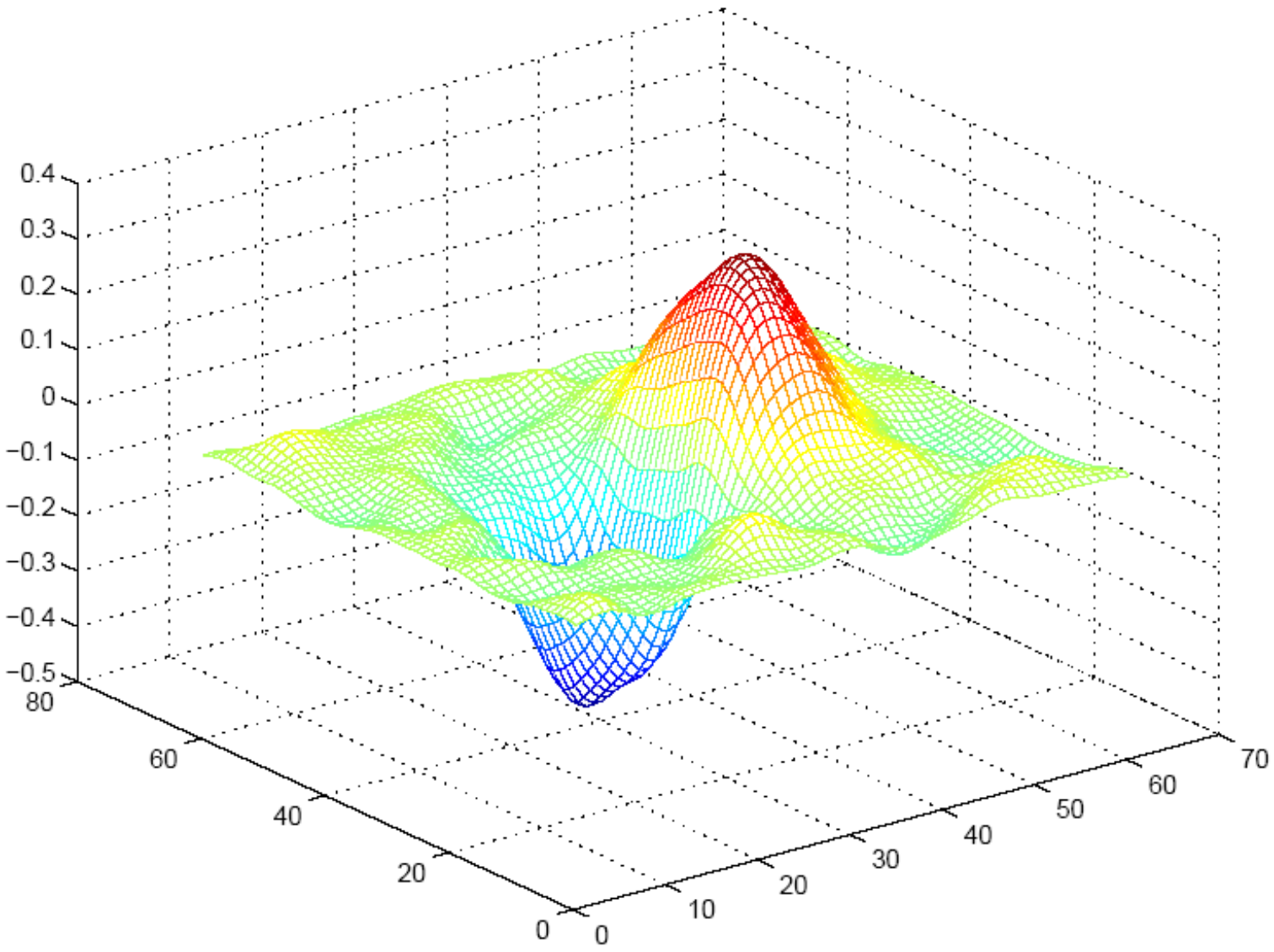
$$\begin{aligned} \int K_H * X(t)\, dt &= \int_t \int_s K_H(t-s) X(s)\, ds\, dt \\ &= \int X(s)\, ds. \end{aligned}$$



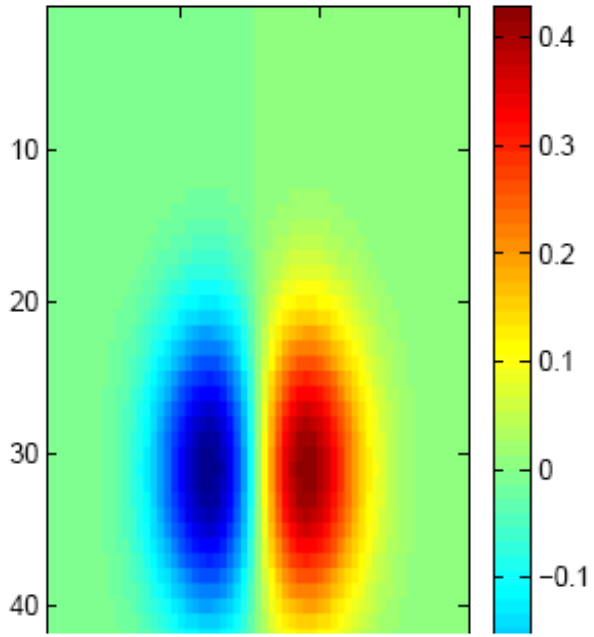
obervation = signal + noise



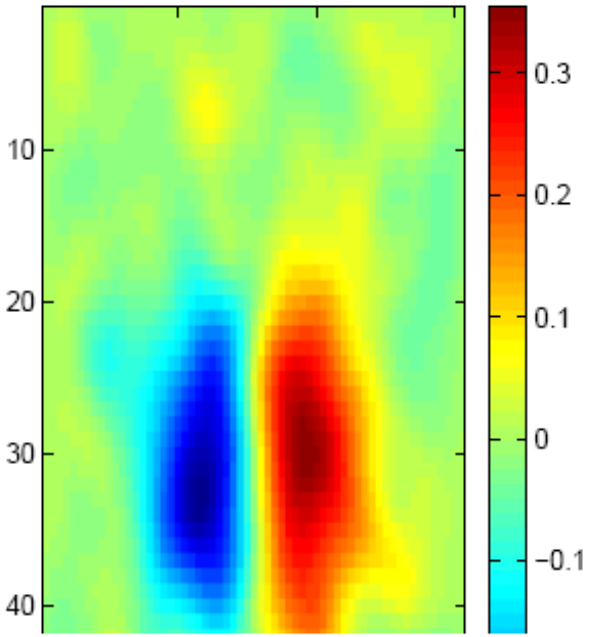
Signal



Prediction

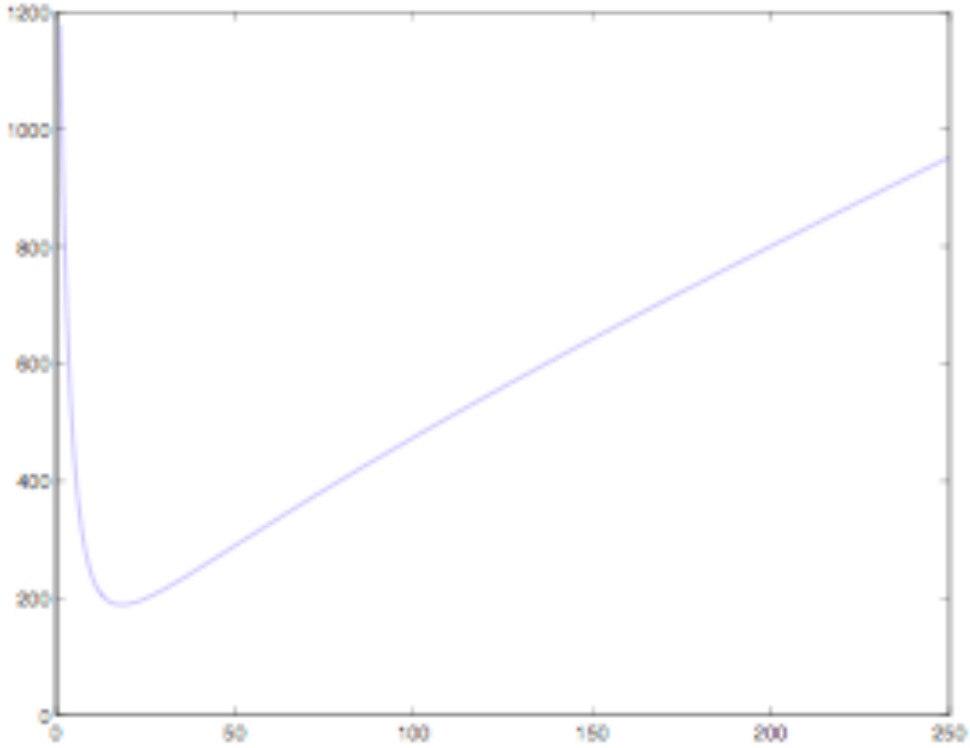


Signal



Prediction

Optimal bandwidth
choose sigma that minimizes the integrated squared error



Many technique uses some sort of cross-validation

4.4 Iterated Kernel Smoothing

When we diffuse measurement Y for duration $t = \sigma^2/2$, we obtain $K_\sigma * Y$. Now we diffuse $K_\sigma * Y$ for additional $t = \sigma^2/2$ and get $K_\sigma * (K_\sigma * Y)$. The

total diffusion duration is σ^2. It should be equivalent to kernel smoothing with bandwidth $\sqrt{2}\sigma$, i.e.

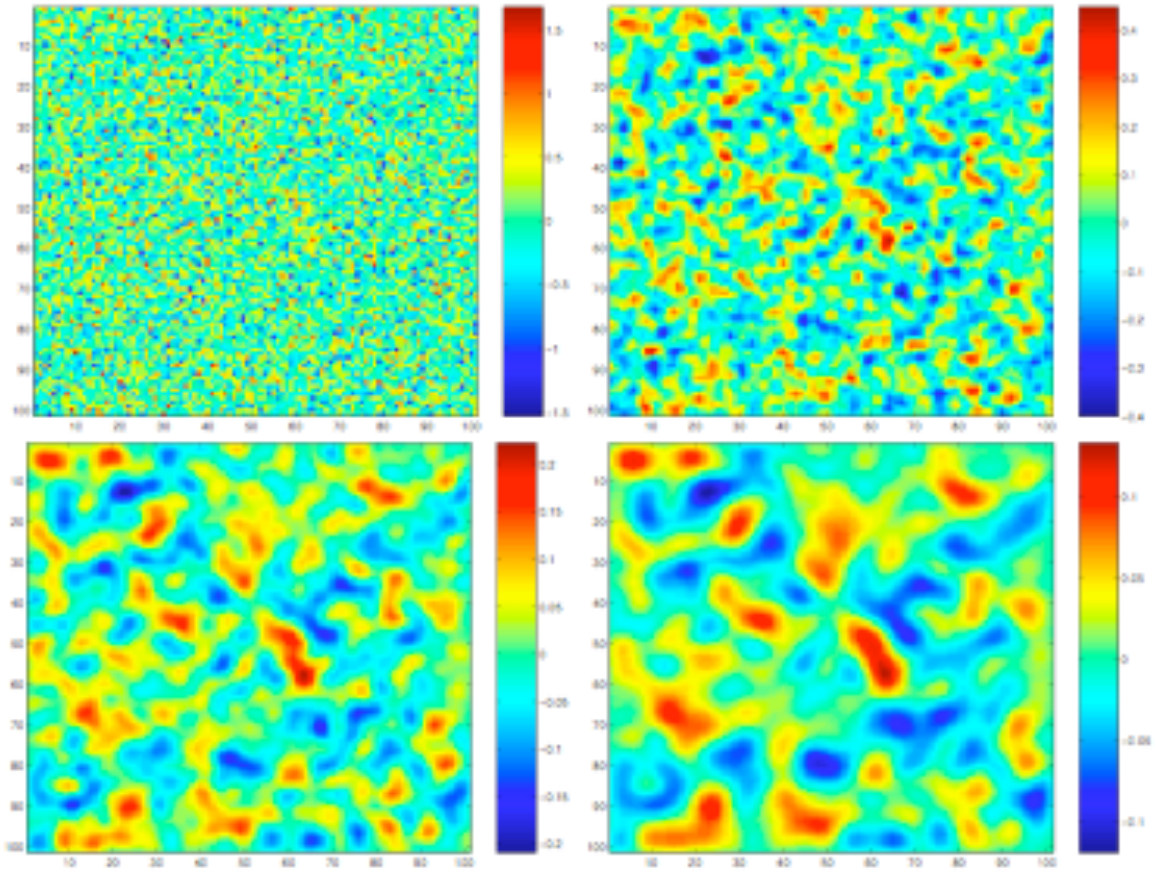
$$K_\sigma * K_\sigma * Y = K_{\sqrt{2}\sigma} * Y.$$

Define m iterative convolutions as

$$K_\sigma^{(m)} = \underbrace{K_\sigma * \cdots * K_\sigma}_{m \text{ times}}.$$

From the inductive argument, we then have $K_\sigma^{(m)} = K_{\sqrt{m}\sigma}$. An alternate proof can be obtained by noting that $K_\sigma^{(m)}$ is the density of the sum of m independent and identically distributed Gaussian random variables with mean zero and variance σ^2. Based on this identity, kernel smoothing with larger bandwidth $\sqrt{m}\sigma$ can be decomposed into m iterated kernel smoothing with smaller bandwidth σ. The iterated kernel smoothing approach has been used to smooth data in irregular grid system such as cortical surfaces.

Simulating Gaussian random field

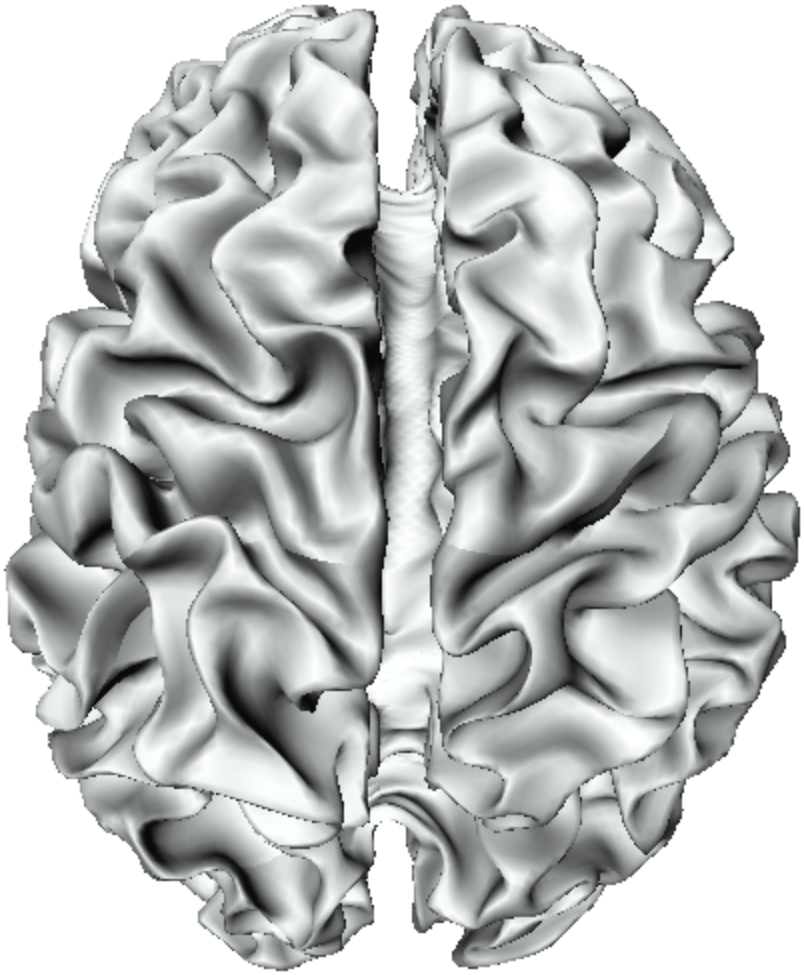


N(0, 0.4^2) Gaussian white noise
Iterative kernel smoothing with sigma=0.4 and 1,4, 9 iterations

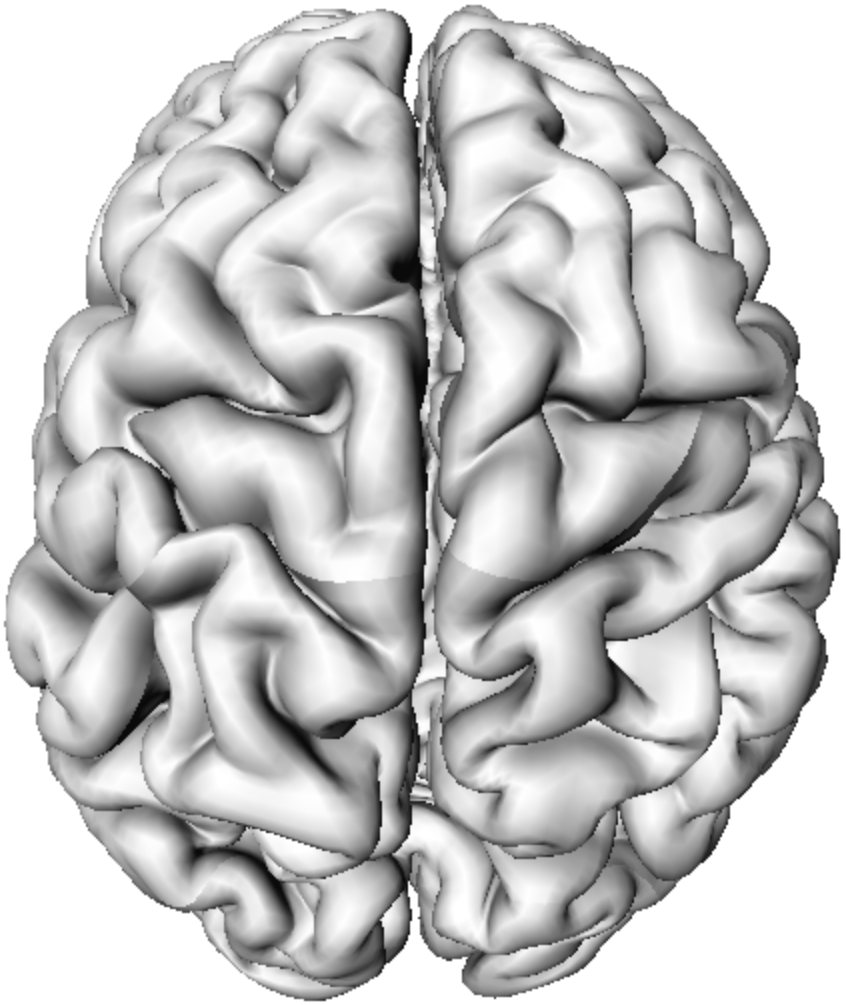
MATLAB Demonstration

Smoothing volume data was easy. How about smoothing surface data?

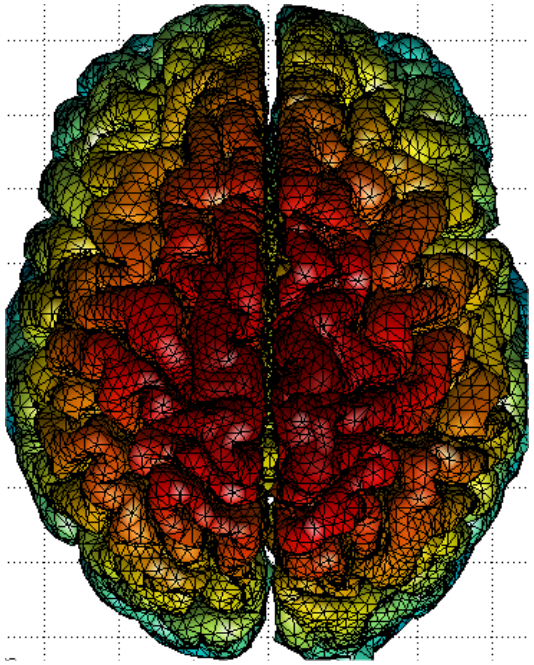
Final surface extraction result



Inner surface

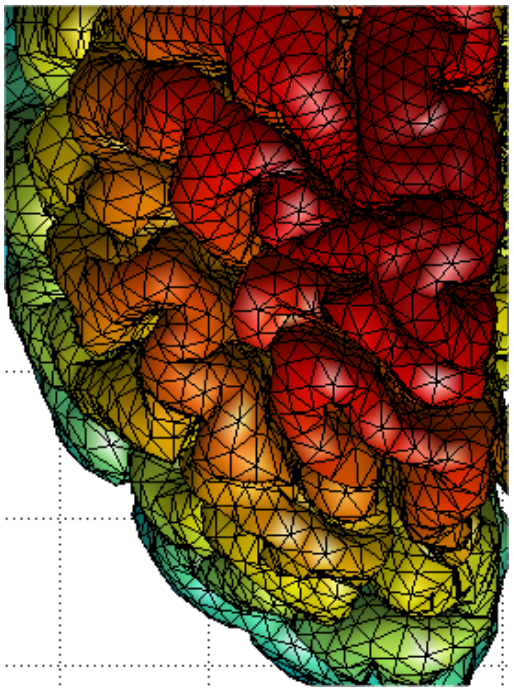


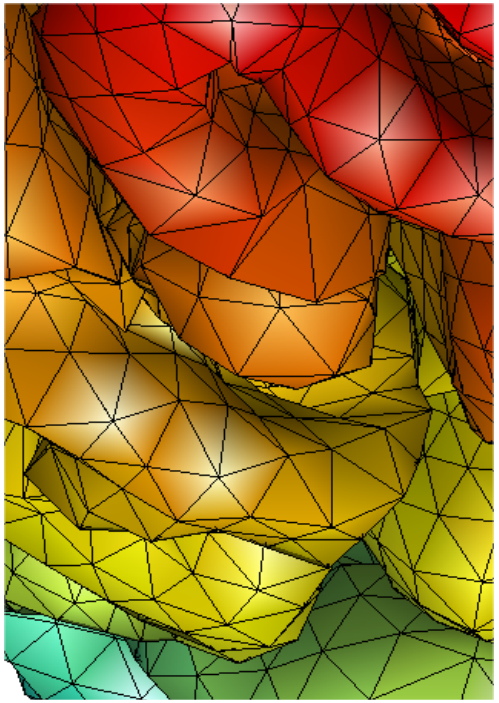
Outer surface



Polygonal mesh data structure

Basis of most surface rendering tools for 3D computer games: 3D Max Studio, Maya





Data structure for polygonal mesh

Coordinates for subject 1

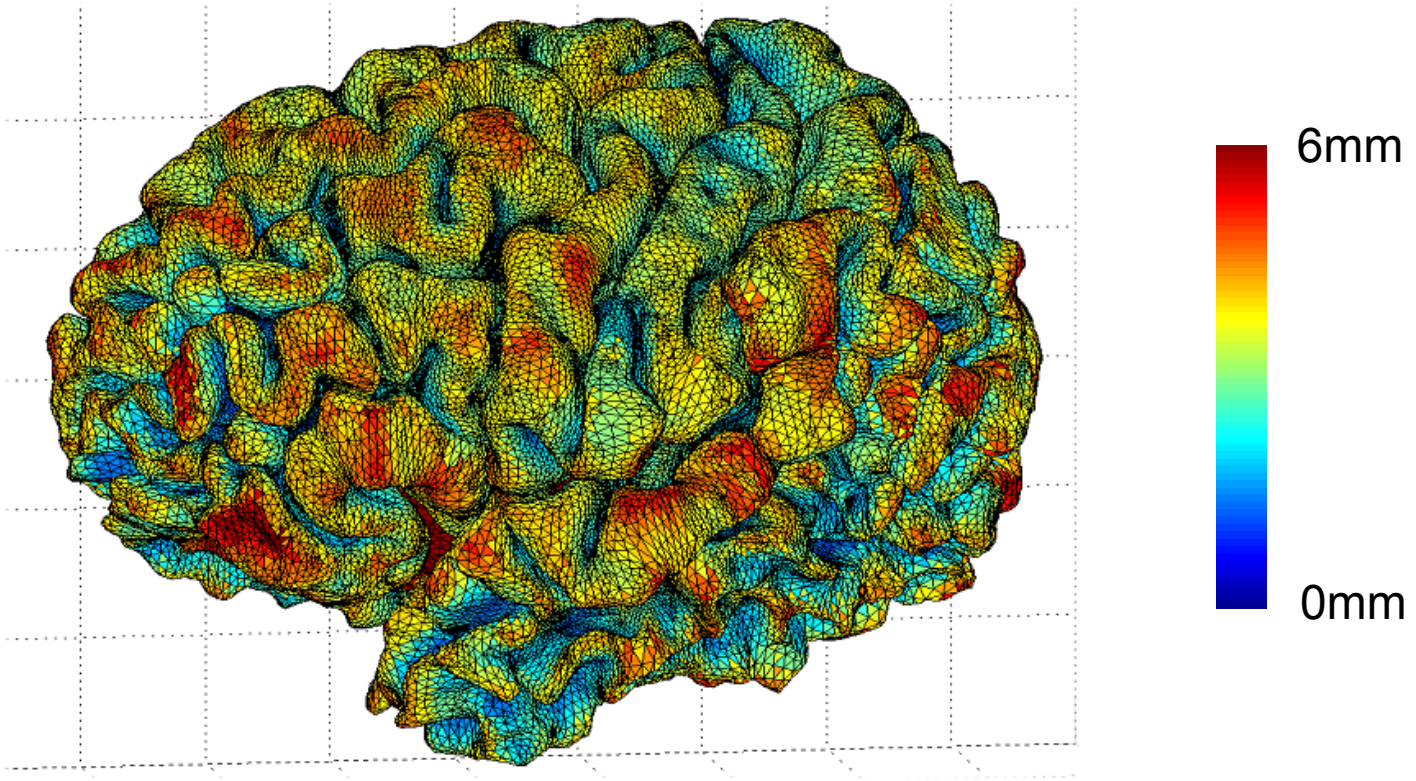
Vertex	1	2	3	4	5	6	….	40962
x	57.1876	41.0450	-53.1115	-38.1080	1.8440	-0.2458		
y	21.6388	-56.3448	29.8912	-65.5394	22.9715	9.4176		
z	2.9667	21.1399	-5.5088	23.6724	21.5146	16.9014		
Thickness	5.0	4.9	3.0	2.1	3.4	4.5		

Coordinates for subject 2

Vertex	1	2	3	4	5	6	….	40962
x	53.4240	41.0552	-61.4073	-43.2099	1.6256	-3.9101		
y	22.5535	-56.7731	20.9221	-65.9948	22.7979	29.7043		
z	7.1866	22.4754	-0.1368	21.3962	20.2838	-10.8959		
Thickness	5.5	3.4	2.7	5.1	3.7	4.5		

Corresponding vertices have approximate anatomical homology.

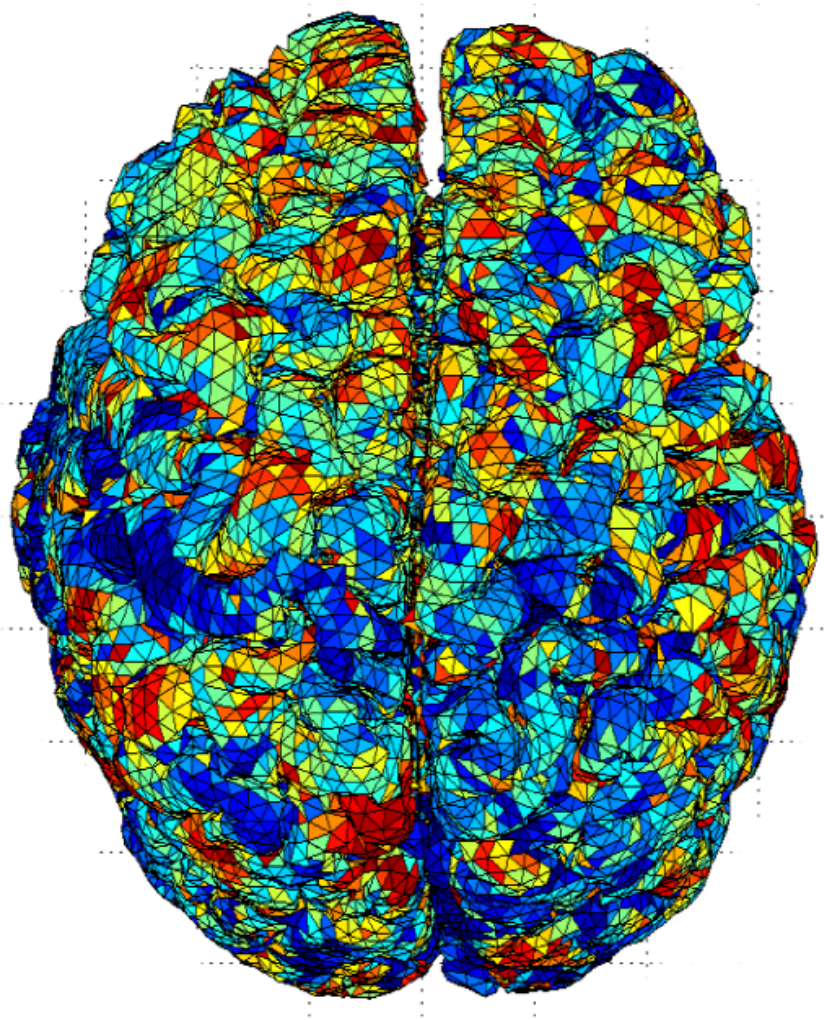
Cortical thickness measurements

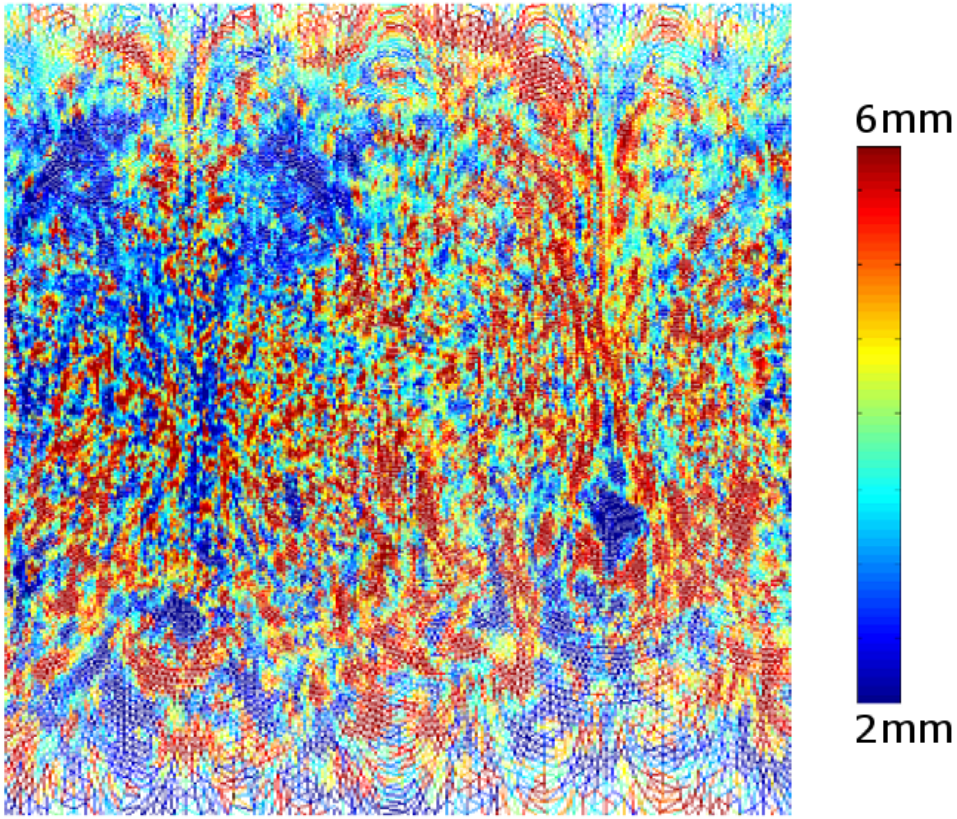


$$Y(p) = \theta(p) + \epsilon(p), p \in \partial\Omega$$

where θ is the mean thickness and ϵ is a zero mean Gaussian random field.

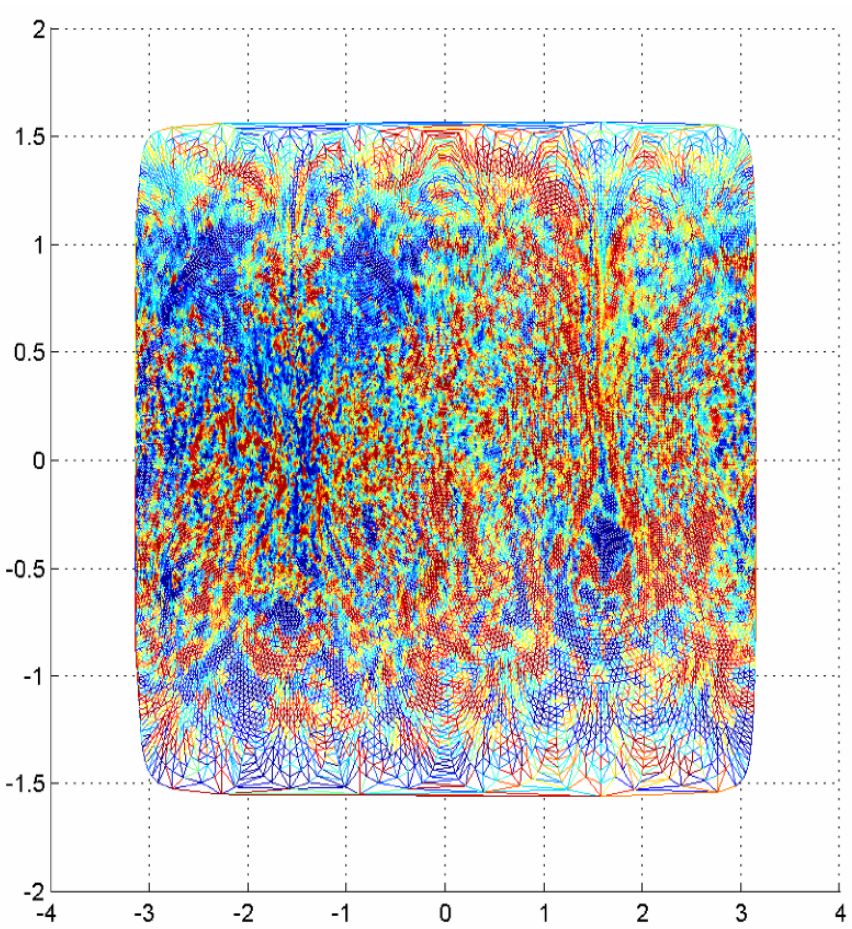
Noisy thickness measures from triangle mesh



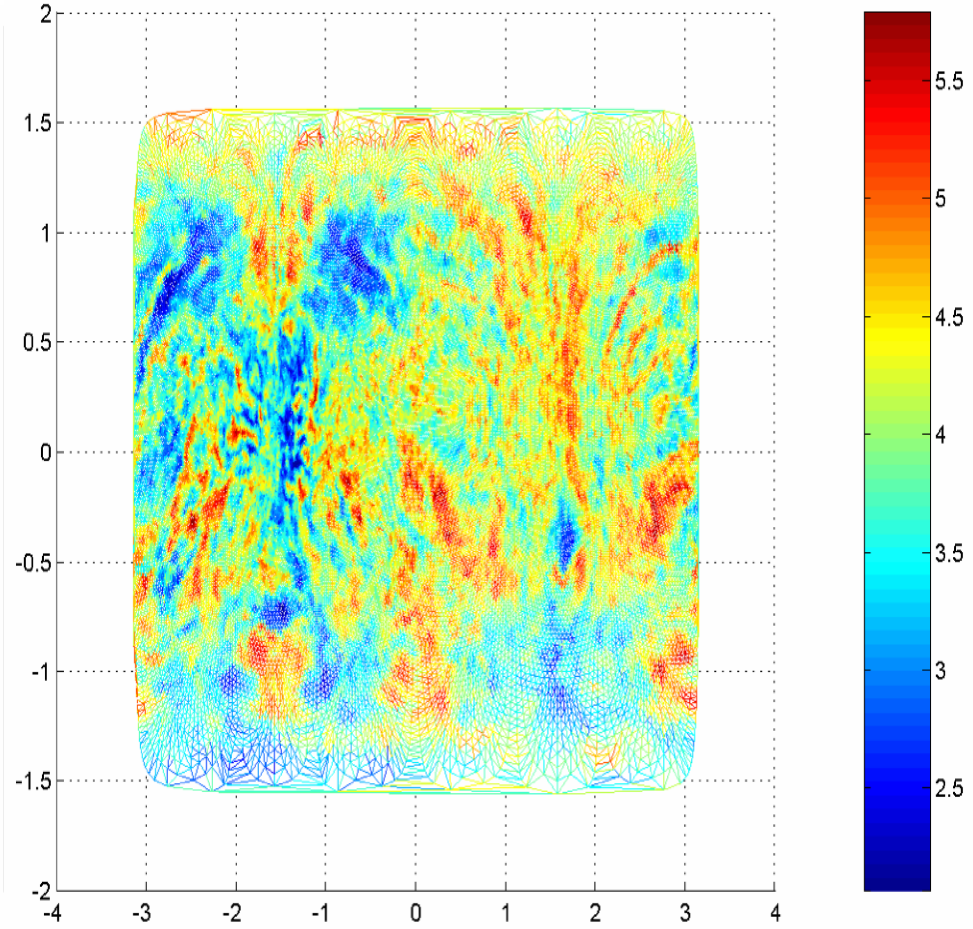


Flattened map:
Spherical projection
(visualization only)

Flat map of cortical thickness



Original thickness



Heat kernel smoothing

Why do we smooth images before statistical analysis?

1. To increase the signal-to-noise ratio (SNR).

2. Random field based multiple comparison correction requires very smooth Gaussian random field assumption.

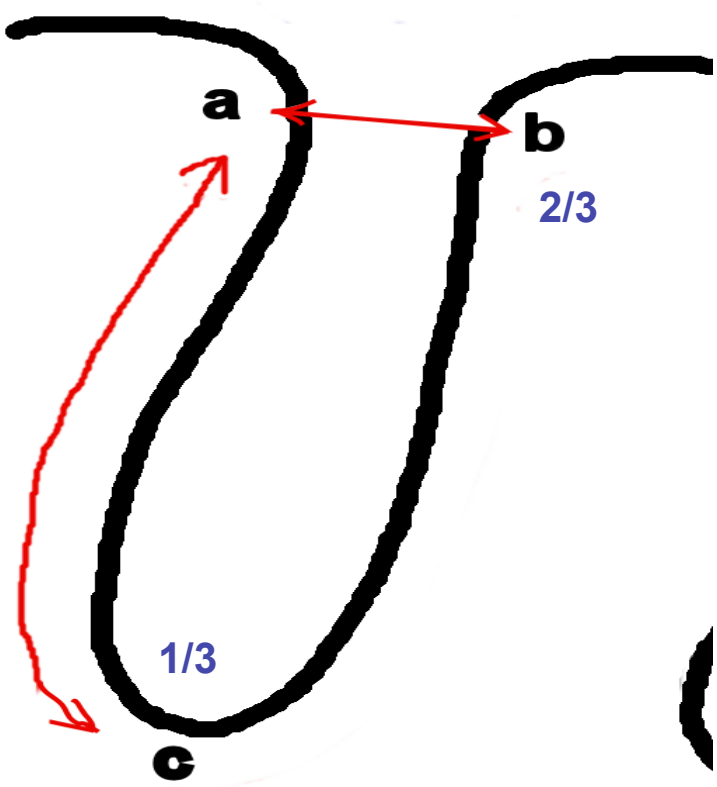
Why Gaussian kernel smoothing?

1. It is a standard technique in imaging.
2. Computationally fast.
3. Computationally easy to implement.

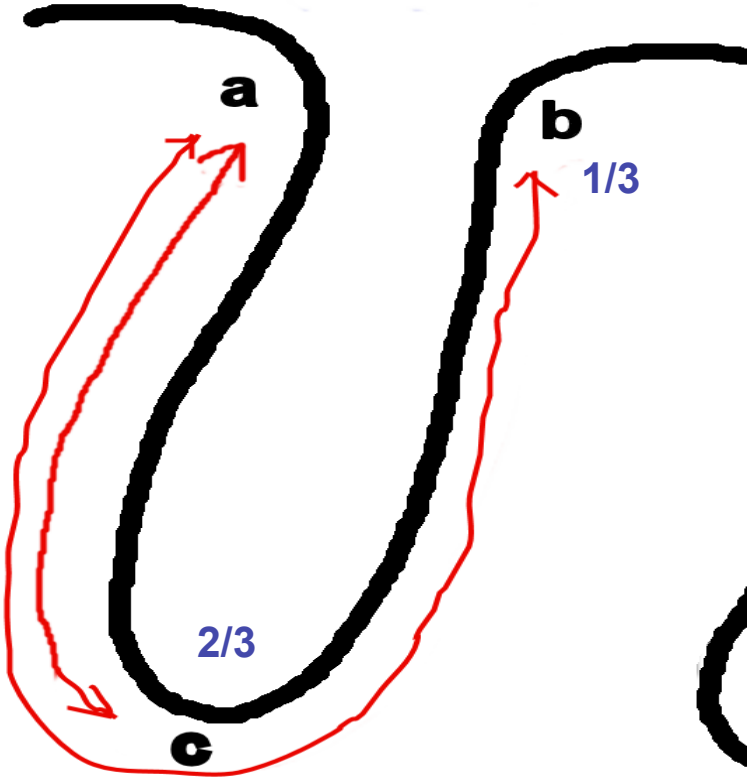
How to perform “kernel smoothing” on anatomical boundary?

Difficulty of performing smoothing along boundary

Due to curved geometry, the shortest distance between two points is not a straight line. So we may incorrectly assign less weights to the closer measurements.



Improper kernel weighting



Proper kernel weighting

Heat kernel smoothing

Gaussian kernel smoothing is a special case of heat kernel smoothing. It is the solution of the isotropic diffusion equation.

$$K_\sigma * Y(p) = \int_{\partial\Omega} K_\sigma(p,q) Y(q)\ d\mu(q)$$

Estimate heat kernel K and simply perform integral convolution on surface.

Iterated heat kernel identity

Theorem 4 *Heat kernel smoothing with large bandwidth can be decomposed into multiple kernel smoothing with smaller bandwidth via*

$$K_\sigma^{(k)} * f = \underbrace{K_\sigma * \cdots * K_\sigma}_{k \text{ times}} * f = K_{\sqrt{k}\sigma} * f.$$

Example.
Single Gaussian kernel smoothing with the bandwidth of 10 mm FWHM.
= 100 repeated applications of Guassian kernel smoothing with the bandwidth of 1 mm FWHM.

1st order approximation of heat kernel

$$\widetilde{W}_\sigma(p, q_i) = \frac{\exp\left[-\frac{d^2(p,q_i)}{2\sigma^2}\right]}{\sum_{j=0}^{m} \exp\left[-\frac{d^2(p,q_j)}{2\sigma^2}\right]}$$

Heat kernel smoothing is performed iteratively with smaller FWHM.

Algorithm 1

For $i = 1$ *to* n *do*

Find a set of neighboring vertices $N(q_i)$ *of* q_i.

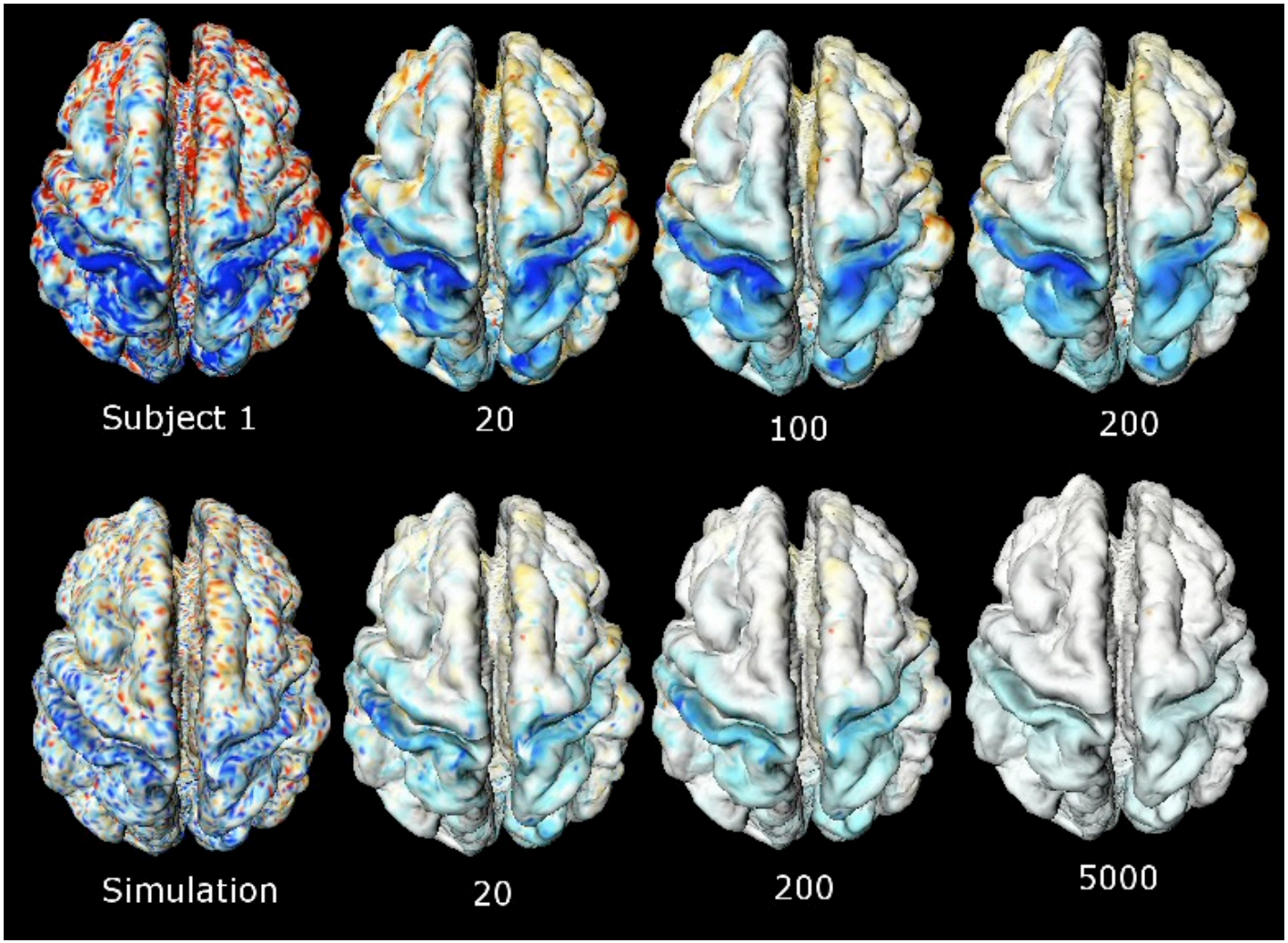
Compute the weighted average and store $Z(q_i) \leftarrow W_\sigma * Y(q_i)$.

End.

Update $Y \leftarrow Z$.

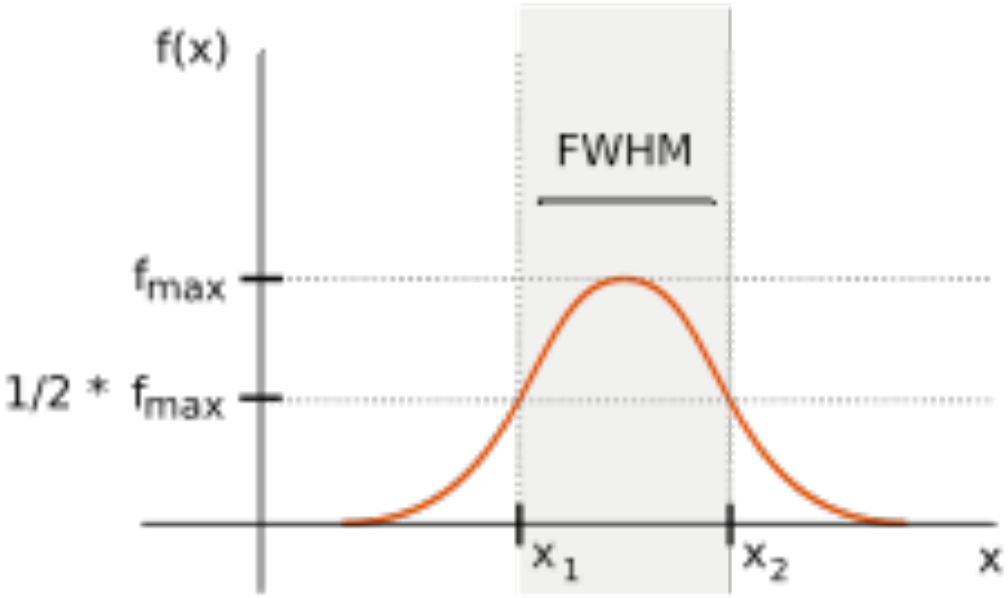
Repeat this procedures k*-times.*

Heat kernel smoothing on cortical thickness



MATLAB Demonstration of Heat Kernel Smoothing

Full width at half maximum (FWHM)



$$f(t) = \frac{1}{\sqrt{2\pi}} e^{t^2/2}.$$

Let's scale the kernel by the parameter σ:

$$K_\sigma(t) = \frac{1}{\sigma} f\left(\frac{t}{\sigma}\right).$$

This is the density function of the normal distribution with mean 0 and variance σ^2. The *bandwidth* σ defines the spread of kernel. The full width at the half maximum (FWHM) of Gaussian kernel K_σ is given by $2\sqrt{2\ln 2}\sigma$. This has been widely used unit for measuring smoothness with respect to kernel smoothing in brain imaging.

Motivation for spatially adaptive anisotropic smoothing

Need for smoothing data while preserving boundary information

The Canadian Journal of Statistics
Vol. 28, No. 2, 2000, Pages 225–240
La revue canadienne de statistique

Differential equation models for statistical functions[1]

James O. RAMSAY

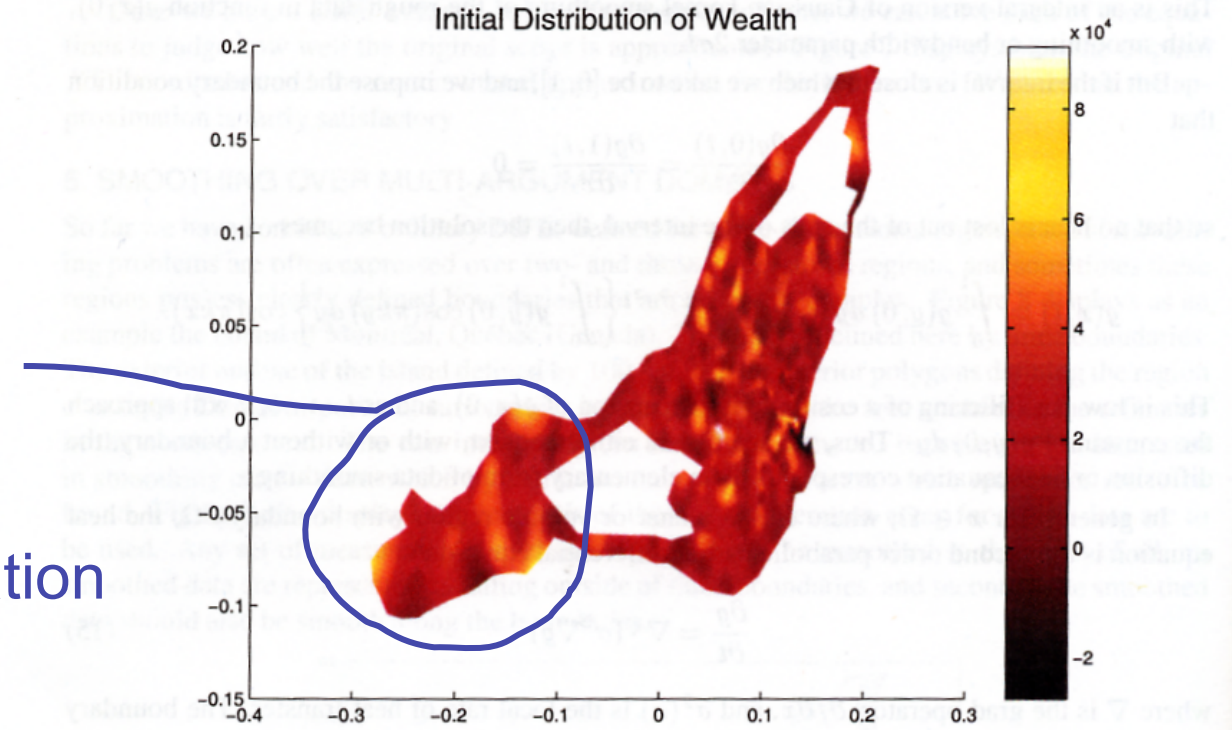


FIGURE 10: The distribution of wealth on the Island of Montréal. This defines the initial state $g(x, 0)$ of the system described by partial differential equation (15) and boundary condition (16).

Wealth concentration in Montreal area

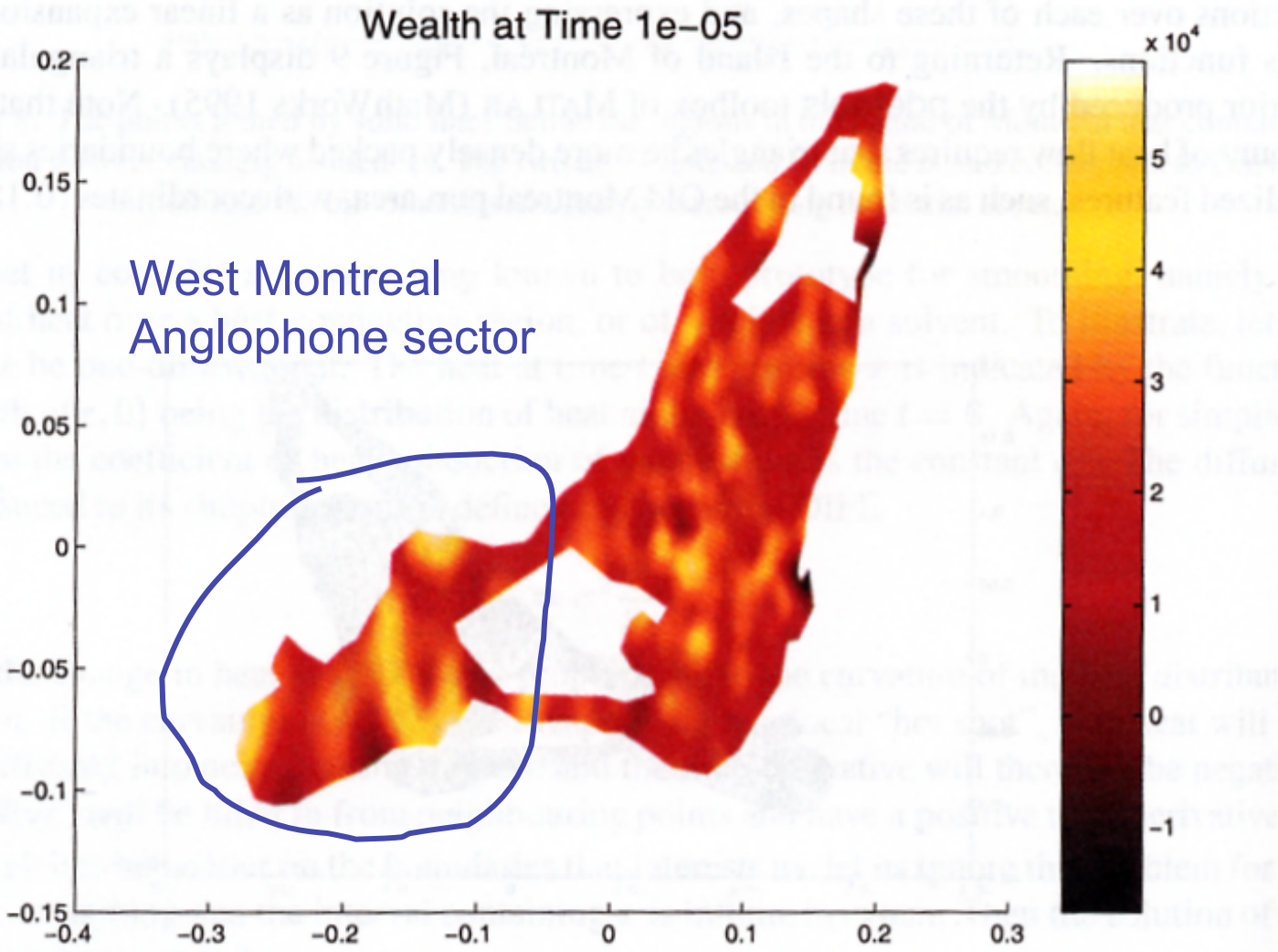


FIGURE 11: The distribution of wealth at time 0.00001.

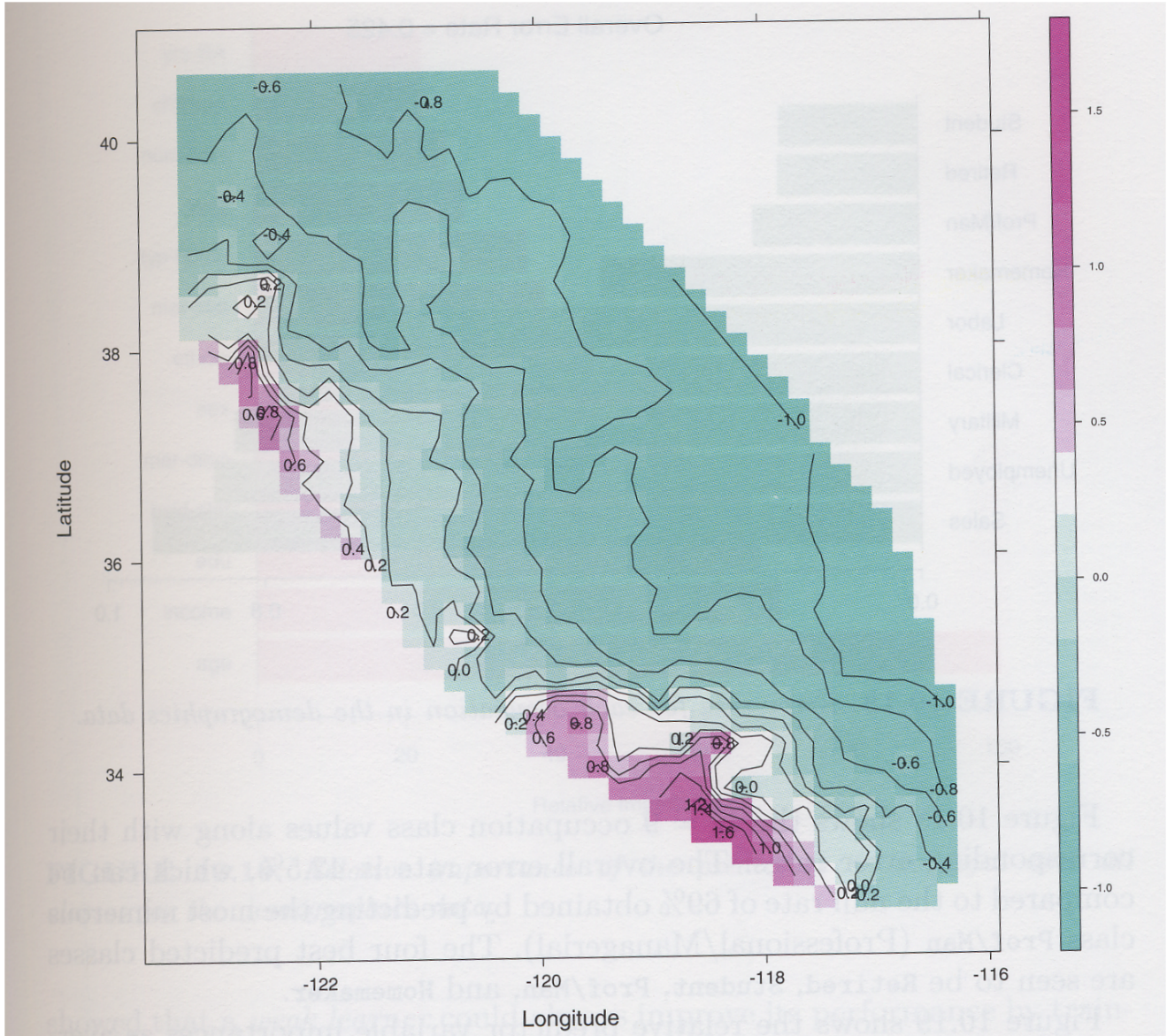


FIGURE 10.17. *Partial dependence of median house value on location in California.*

Hastie

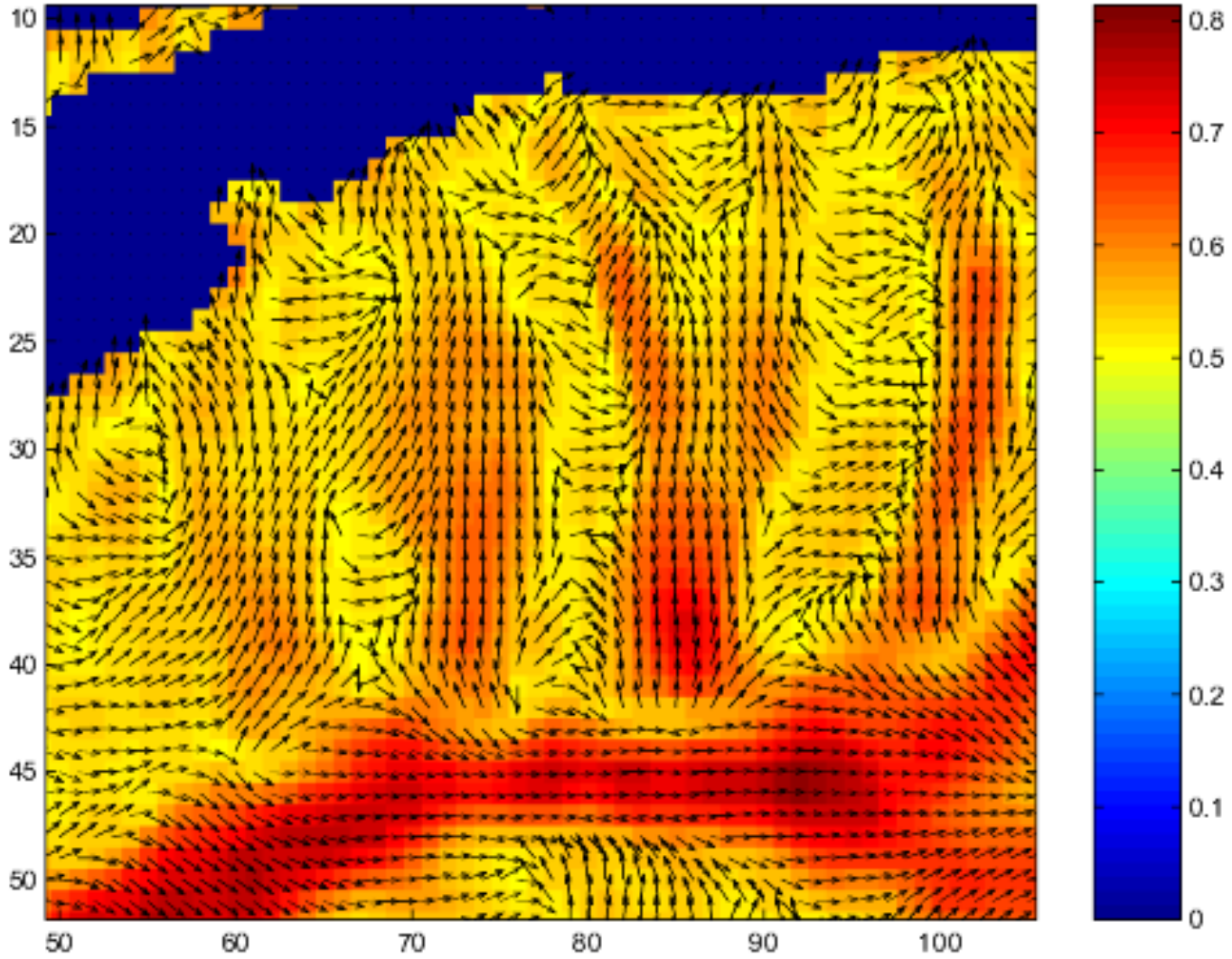
Smoothing while preserving edge

NOISY IMAGE

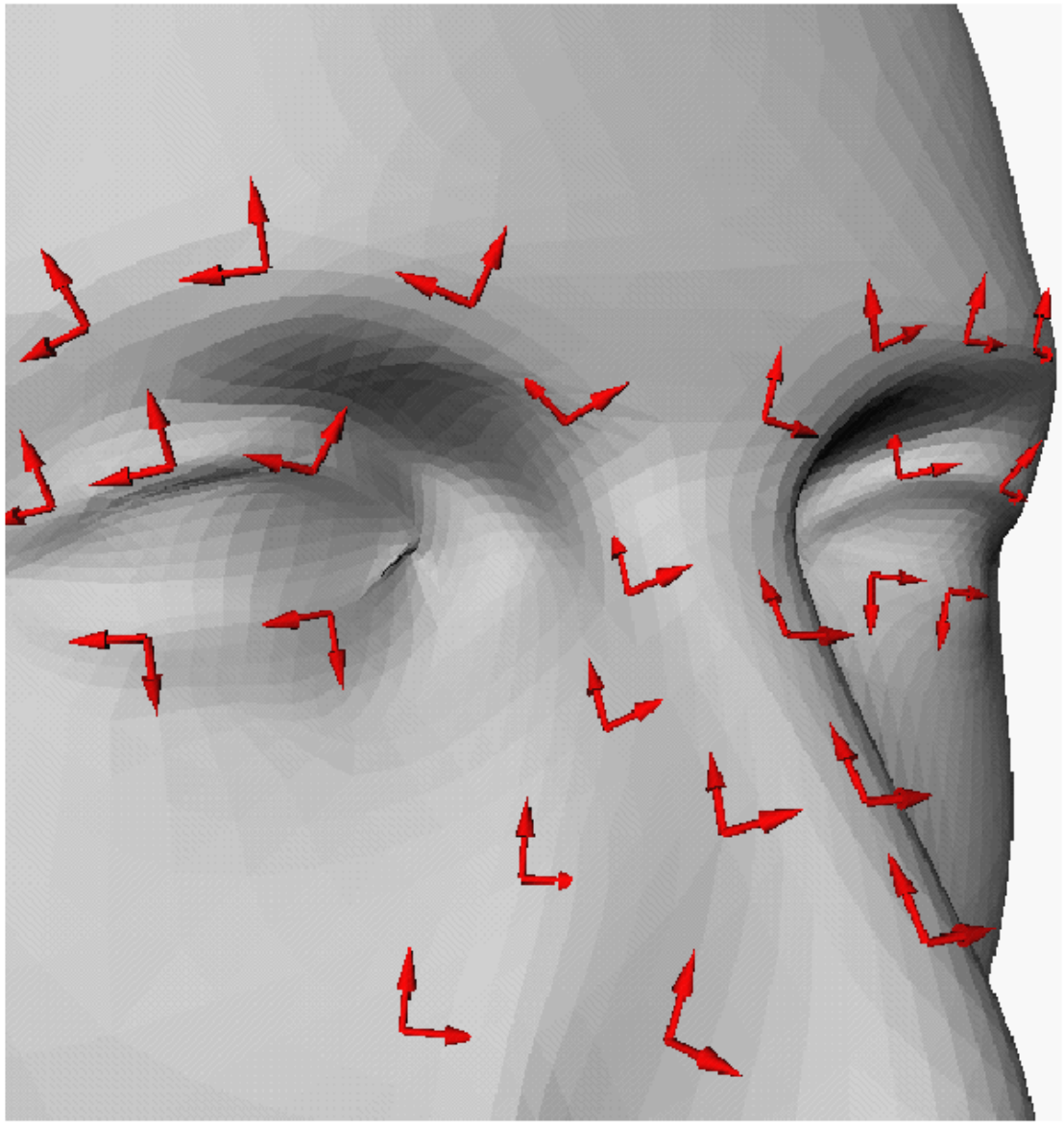
DENOISED IMAGE

Figure 10. Image denoising using Beltrami flow [Kimmel et al].

Diffusion tensor imaging (DTI): Smoothing along vector or tensor fields



Arrows = Principal eigenvectors
Colors = Principal eigenvalues of diffusion coefficient matrix.



Smoothing along tensor fields

Principal curvature direction
Meyer *et al.*

Two main approaches

- Anisotropic kernel smoothing
- Anisotropic diffusion equation

Kernel smoothing

- Isotropic kernel smoothing: weighted averaging where weights are isotropic

- Anisotropic kernel smoothing: weights are not isotropic. It contains directional information

Anisotropic kernel

The isotropic kernel under linear transform $\mathbf{x} \to H\mathbf{x}$ changes the shape of the kernel to an anisotropic kernel

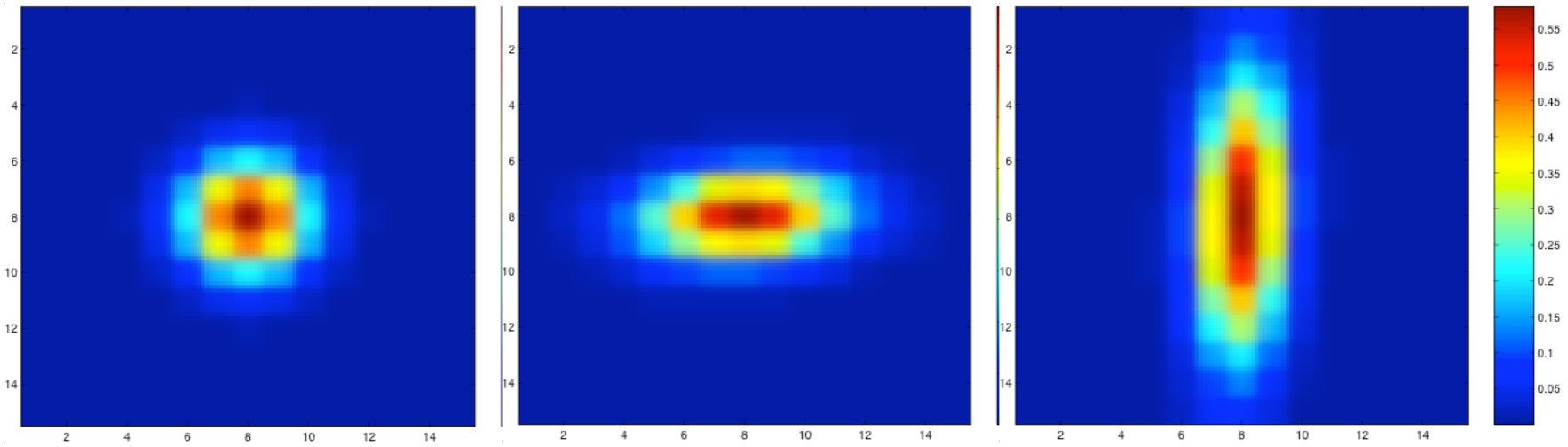
$$K_H(t) = \frac{1}{\det H} K(H^{-1}\mathbf{x}).$$

$\det H$ is the Jacobian determinant of the transformation that normalize the kernel. Note that this is the density of n-dimensional multivariate normal with the covariance matrix HH', i.e. $N(0, HH')$. The matrix H is called the *bandwidth matrix* and it measures the amount of smoothing. It can be shown that K_H is the Dirac-delta function when all the eigenvalues $\lambda_1, \cdots, \lambda_n$ of H go to zero, i.e.

$$\lim_{\lambda_1, \cdots, \lambda_n \to 0} K_H(\mathbf{x}) = \delta(\mathbf{x}).$$

From now on we $H \to 0$ if all $\lambda_i \to 0$ and $H \to \infty$ if all $\lambda_i \to \infty$.

Kernel weights

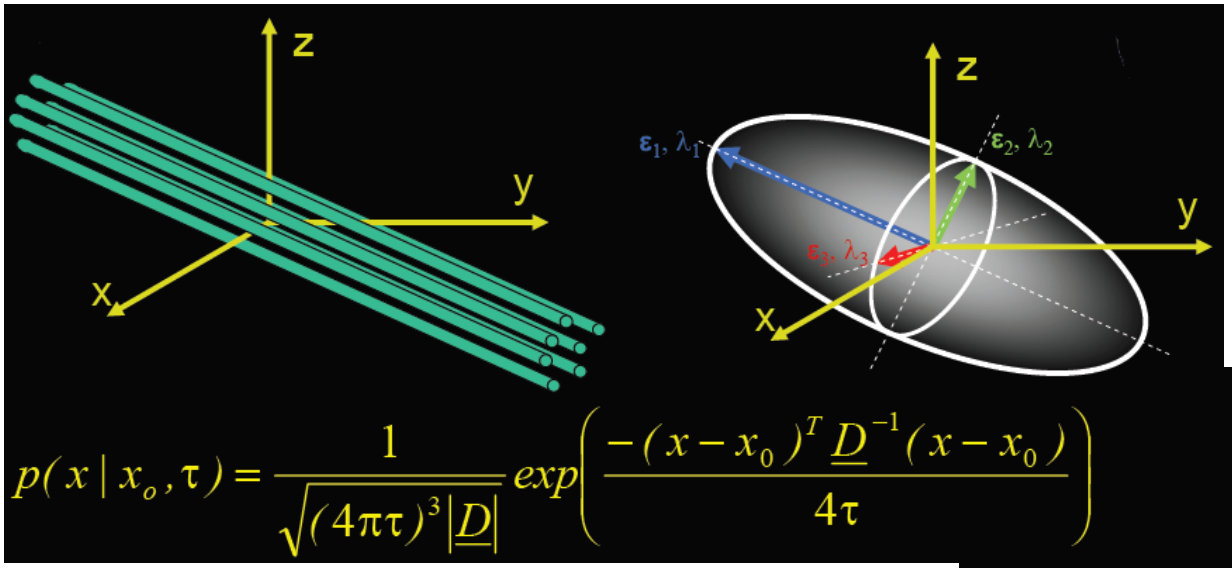


Isotropic kernel

Anisotropic kernel

Diffusion Tensor Imaging

Lecture 5-6 topic
(6 hours)



$$p(x \mid x_o, \tau) = \frac{1}{\sqrt{(4\pi\tau)^3 |\underline{D}|}} exp\left(\frac{-(x - x_0)^T \underline{D}^{-1} (x - x_0)}{4\tau} \right)$$

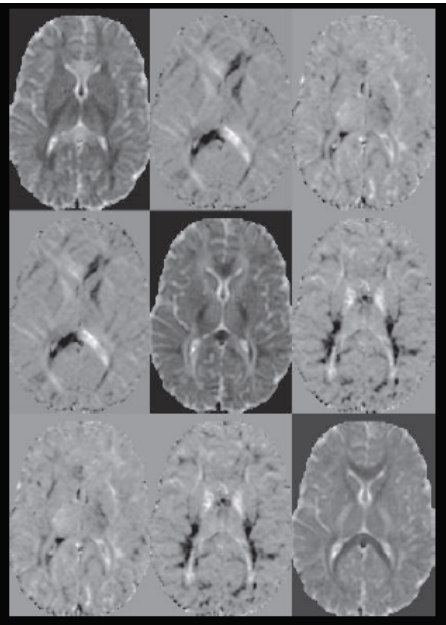
$$\mathbf{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix} = \mathbf{E} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \mathbf{E}$$

Diffusion Tensor

Eigenvalues

Matrix of 3 eigenvectors

$$\mathbf{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix} =$$



$$D_{xx}, D_{yy}, D_{zz} > 0 \qquad D_{xy} = D_{yx}; D_{xz} = D_{zx}; D_{yz} = D_{zy}$$

Please read following two papers for lecture 5

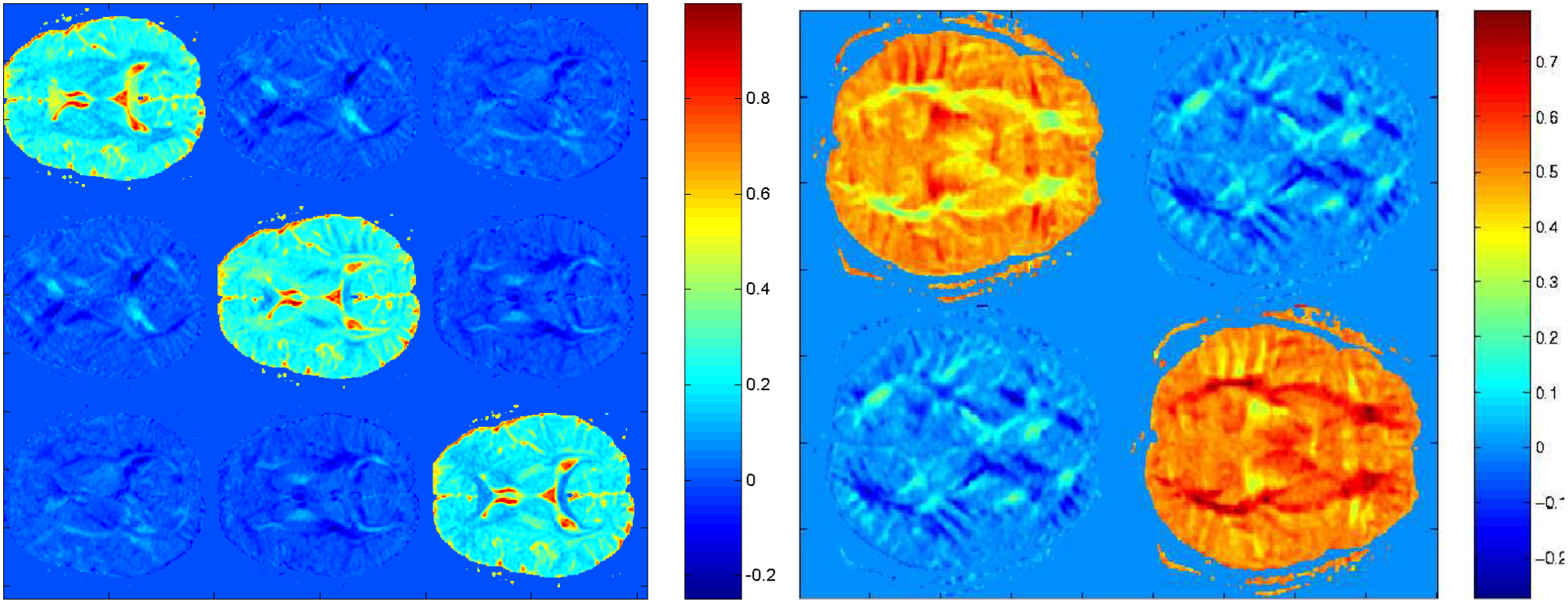
Alexander.2007….. : review paper

Chung.2010….. : method paper dealing with tract shape analysis

Application to DTI data

$$D = (d_{ij})$$

6 diffusion coefficient matrix D_xx, D_xy, D_xz, D_yy, D_yz, D_zz



- Diffusion coefficient measures the diffusion of water molecules.
- The principal eigenvector = direction of water molecules.
- This gives indirect information about white matter fibers.

Tractography

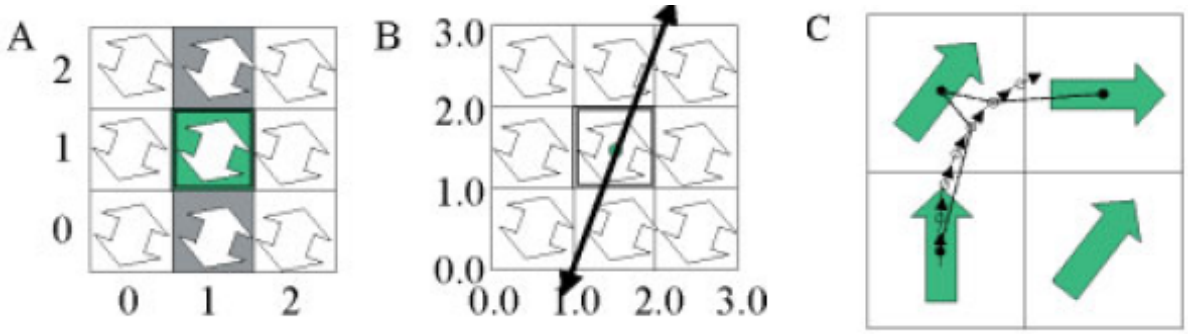
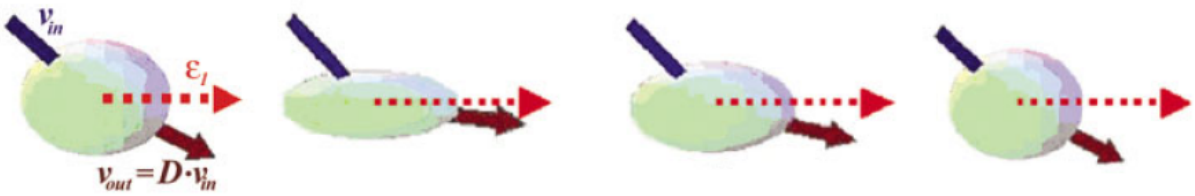
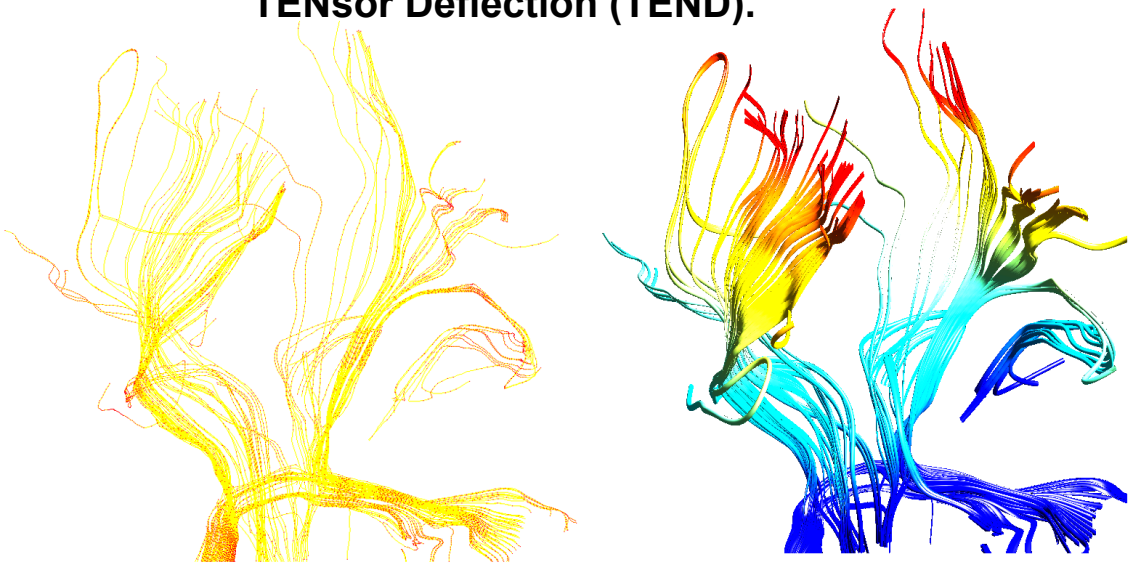


Figure 2. Schematic diagram of the linear line-propagation approach. Double-headed arrows indicate fiber orientations at each pixel. Tracking is initiated from the center pixel. In the discrete number field (A), the coordinate of the seed pixel is {1, 1}. If it is judged that the vector is pointing to {1, 2} and {1, 0}, shaded pixels are connected. In the continuous number field (B), the seed point is {1.50, 1.50} and a line, instead of a series of pixels, is propagated. (C) shows an example of the interpolation approach to perform nonlinear line propagation. Large arrows indicate the vector of the largest principal axis. For every step size, a distance-weighted average of nearby vectors is calculated. In this example, the vector orientations of two nearest pixels are averaged as the line is propagated



TENsor Deflection (TEND).



Mori and van Zijl NMR Biomed 2002

Camino software package

Second order Runge-Kutta algorithm with TEND (Lazar et al., HBM 2003).

Camino DTI toolbox

http://en.wikipedia.org/wiki/Camino_(diffusion_MRI_toolkit)

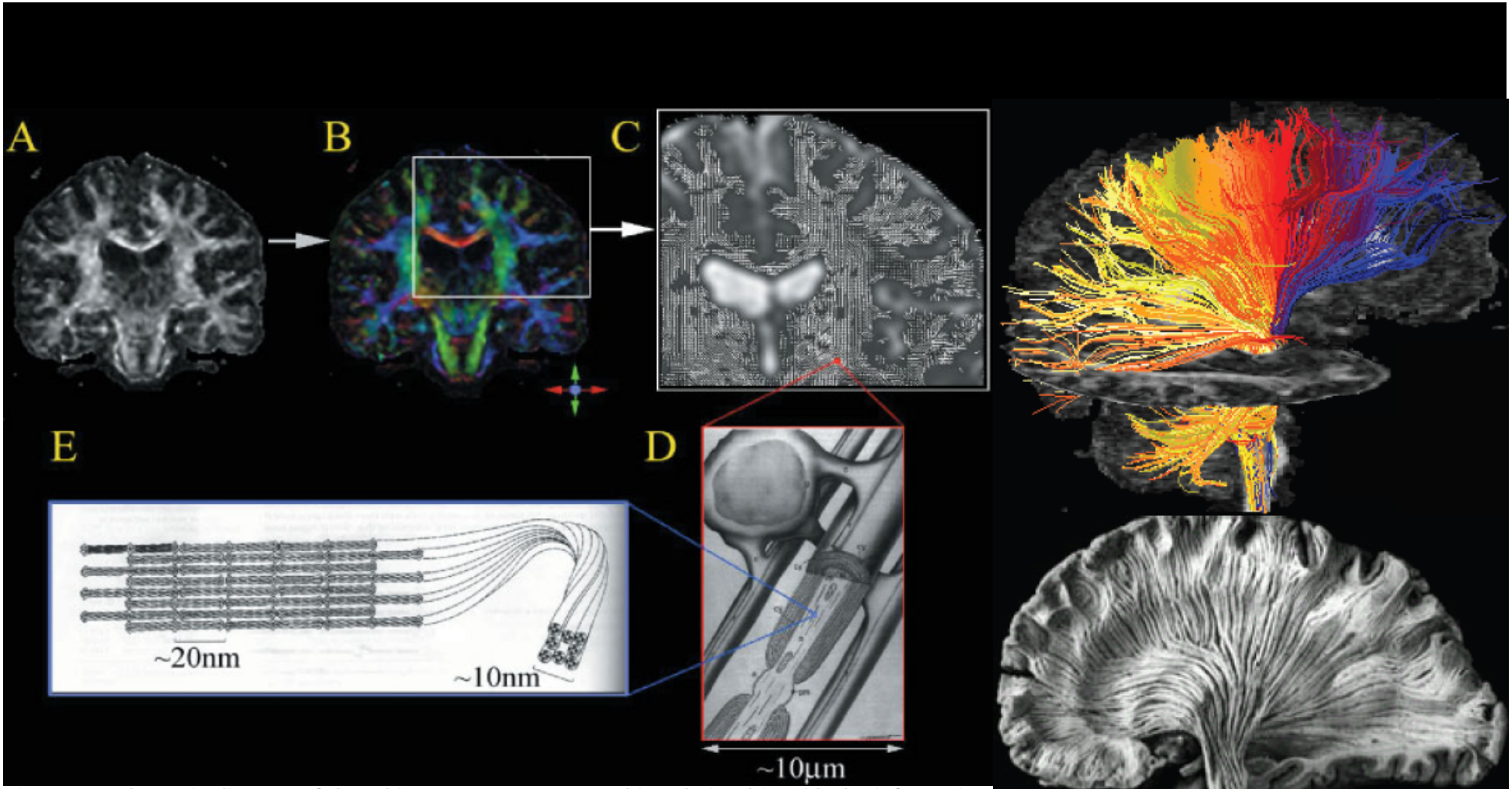
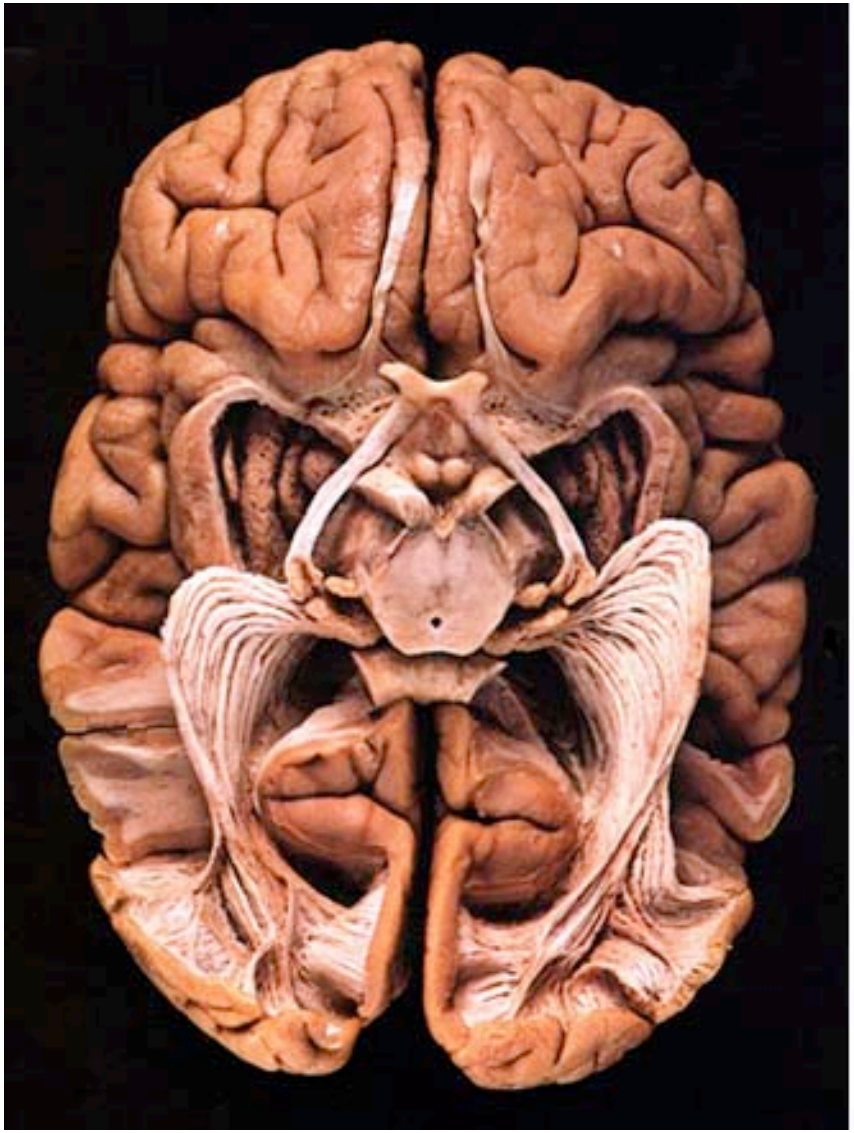
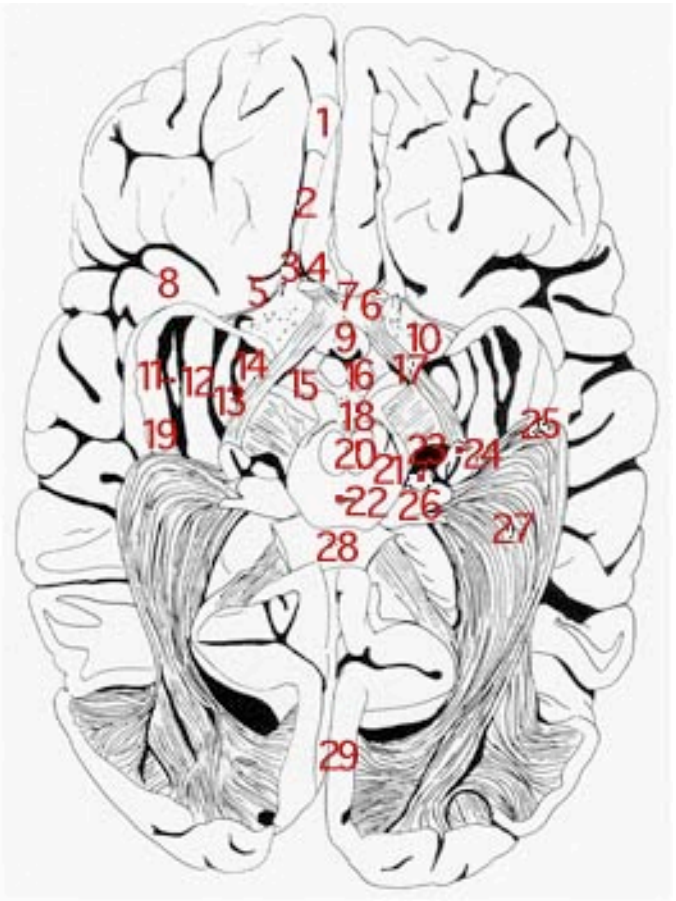


Figure 1. Schematic diagram of the white matter structure and its relationship with the information provided by DTI-based images, such as anisotropy maps (A), have sufficient resolution to segment white and gray matter. By incorporating DTI orientation information, white matter can be parcellated into various tracts using a color-coded map (B) or a vector map (C). The image resolution is sufficient to delineate large white matter tracts, which mostly consist of neuroglia and axons that are largely running parallel. A pixel thus contains bundles of axons and neuroglial cells (D). Note that the size of a pixel (C) is on the order of mm but that the size of the cells (D) is on the order of μm. The axon is filled with neuronal filaments (E) running along its longitudinal axis, which may contribute in superimposing anisotropy on the direction of water diffusion. In the color-coded map, red indicates fibers running along the right–left direction, green inferior–superior, and blue anterior–posterior (perpendicular to the plane). The figures (D) and (E) were reproduced from Carpenter[49] and Alberts *et al.*[64] respectively with permission

Mori and van Zijl NMR Biomed 2002

White matter fibers

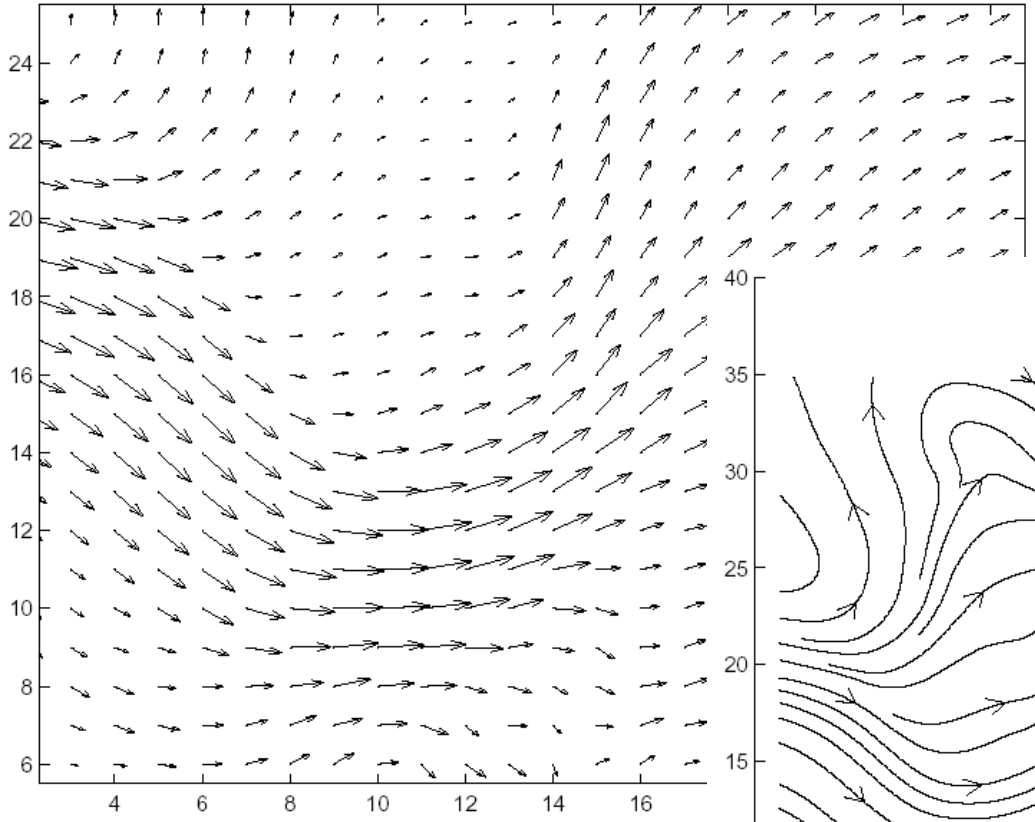




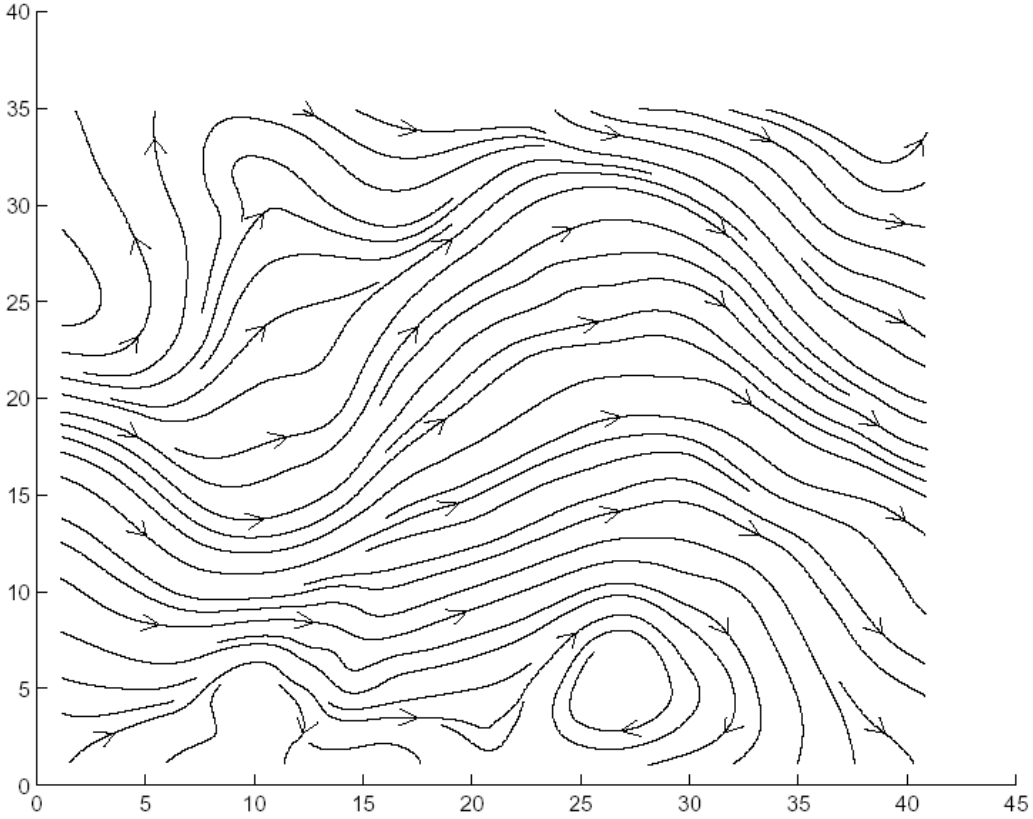
1. Olfactory bulb
2. Olfactory tract
3. Olfactory trigone
4. Medial olfactory stria
5. Lateral olfactory stria
6. Optic nerve

www.vh.org

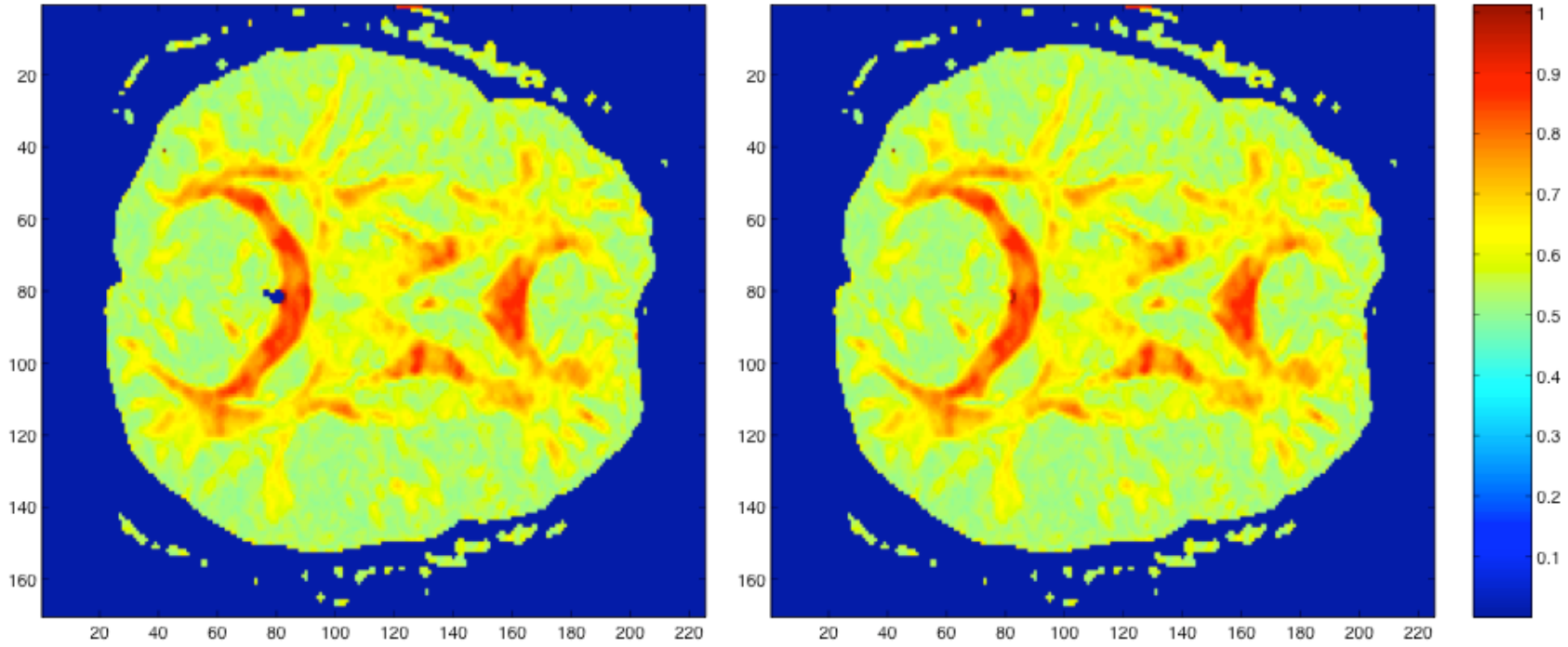
For given vector fields there exists a family of curves whose tangent is given by the vector fields.



Stream lines generated by built in MATLAB function



Principal eigenvalues



Diffusion tensor imaging (DTI) is a new technique that provides the directional information of water diffusion in the white matter of the brain. The directional information is usually represented as a symmetric positive definite 3×3 matrix $D = (d_{ij})$ which is usually termed as the *diffusion tensor* or *diffusion coefficients*. The diffusion tensors are usually normalized by the transpose, i.e. $D/\mathrm{tr}D$. This normalization guarantees that the sum of eigenvalues of D to be 1.

The eigenvectors and eigenvalues are obtained by solving

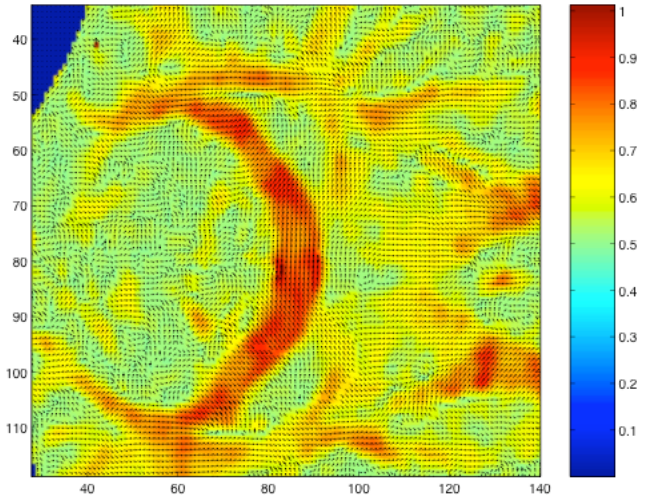
$$D\mathbf{V} = \lambda\mathbf{V}.$$

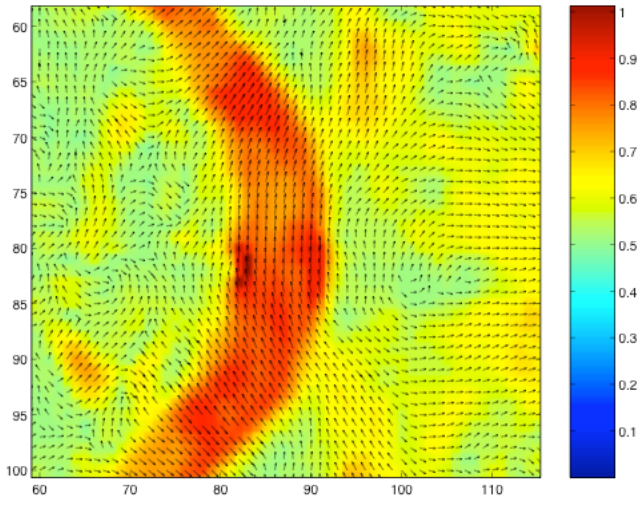
Consider a vector field $\mathbf{V}$ which is the principal eigenvector of D with $\|\mathbf{V}\| = 1$ with the corresponding principal eigenvalue λ. Now suppose that we would like to smooth signals along the vector fields $\lambda\mathbf{V}$ such that we smooth more along the larger vector fields. Suppose that the stream line or flow $\mathbf{x} = \psi(t)$ corresponding to the vector field is given by

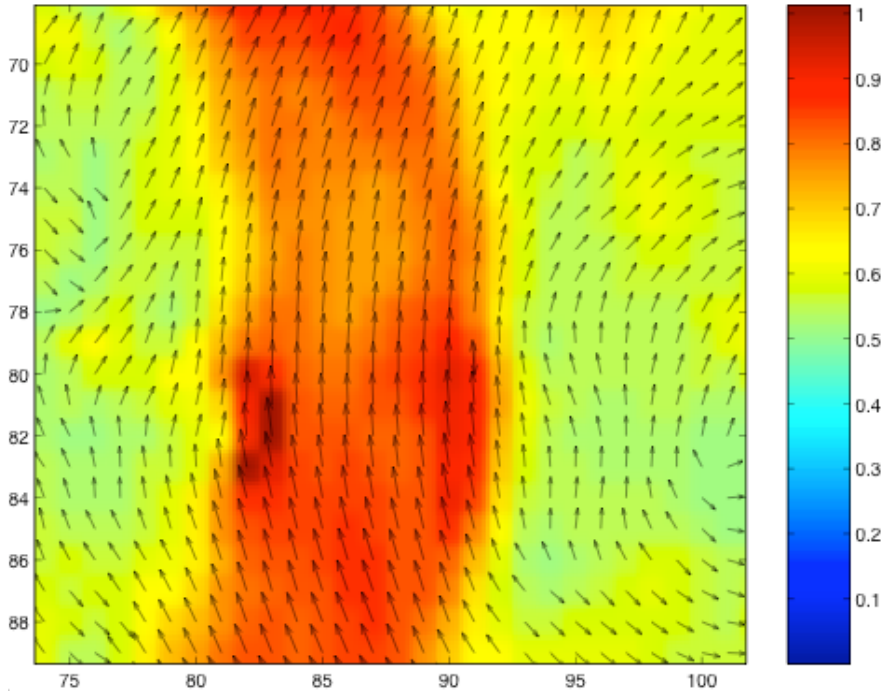
$$\frac{d\psi}{dt} = (\lambda\mathbf{V}) \circ \psi(t).$$

This ordinary differential equation gives a family of integral curves whose tangent vector is $\lambda\mathbf{V}$ (Betounes, 1998).

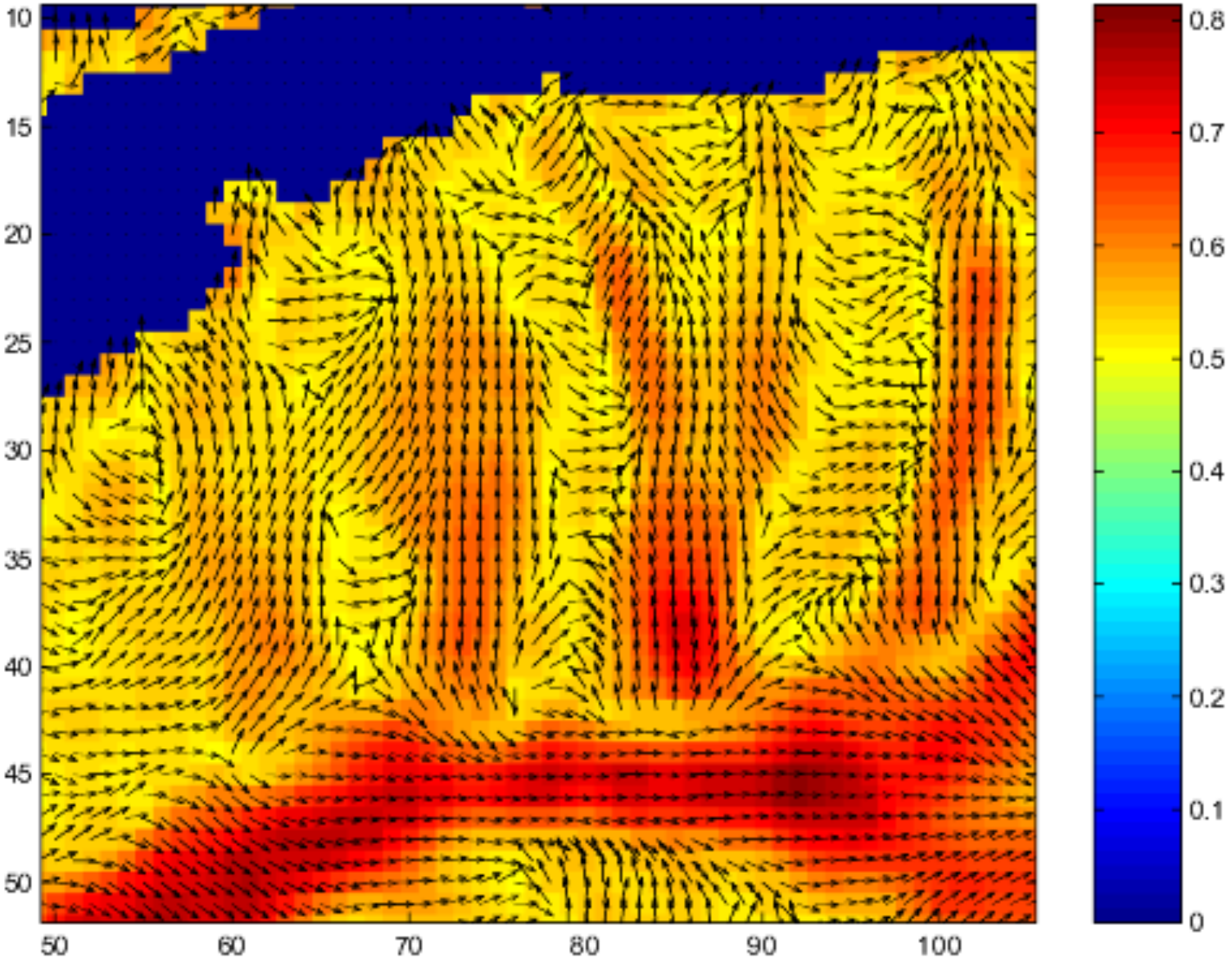
Principal eigenvectors







Smooth along principal eigenvectors



The stream line $\psi(t)$ corresponding to the vector field is given by

$$\frac{d\psi}{dt} = \mathbf{V}.$$

This ordinary differential equation gives a family of curves whose tangent vector is $\mathbf{V}$. The line element is

$$d\psi^2 = V_1^2 dx_1^2 + \cdots V_n^2 dx_n^2.$$

So $g_{ij} = V_i^2 \delta_{ij}$. Riemannian metric tensor

Choose $HH' = 2tG, G = (g_{ij})$. In this way we smooth more along the larger metric distance.

Relation to Diffusion equation

Let $HH' = 2tD$. Then $K_H(\mathbf{x}) = K_{\sqrt{2t}D^{1/2}(\mathbf{x})}$. Call this kernel $K_t(\mathbf{x})$

$$K_t(\mathbf{x}) = \frac{\exp(-\mathbf{x}' D^{-1}\mathbf{x}/4t)}{(4\pi t)^{n/2}(\det D)^{1/2}}.$$

$f(\mathbf{x}, t) = K_t * g(\mathbf{x}), t \in \mathbb{R}^+$

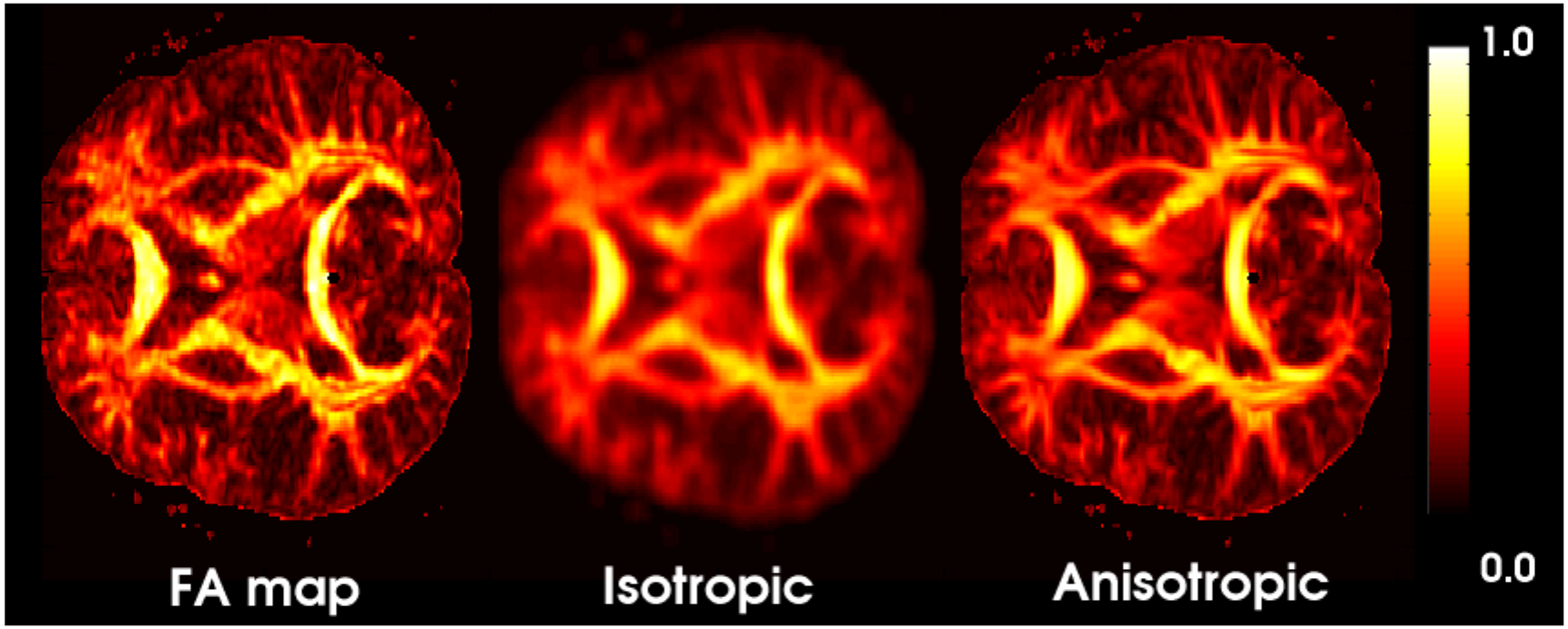
Theorem: $f(\mathbf{x}, t) = K_t * g(\mathbf{x})$ is a unique solution of

$$\frac{\partial f}{\partial t} = \nabla \cdot (D \nabla f) \tag{2}$$

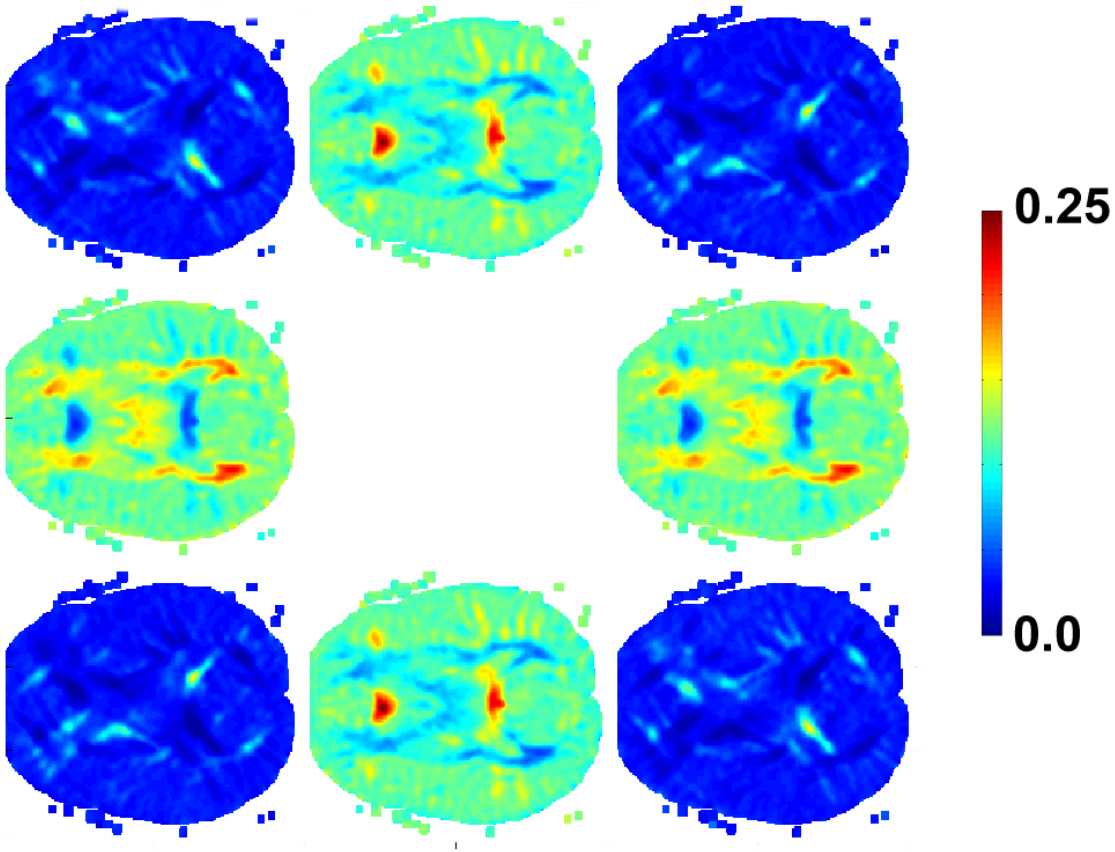
with initial condition $f(\mathbf{x}, 0) = g(\mathbf{x})$. In vector-free notation,

$$\frac{\partial f}{\partial t} = \sum_{i,j=1}^{n} d_{ij} \frac{\partial^2 f}{\partial x_i \partial x_j} \tag{3}$$

The natural Riemmanian metric tensor associated with diffusion process is $G = D$ (De Lara, Annals of Probability, 1995).

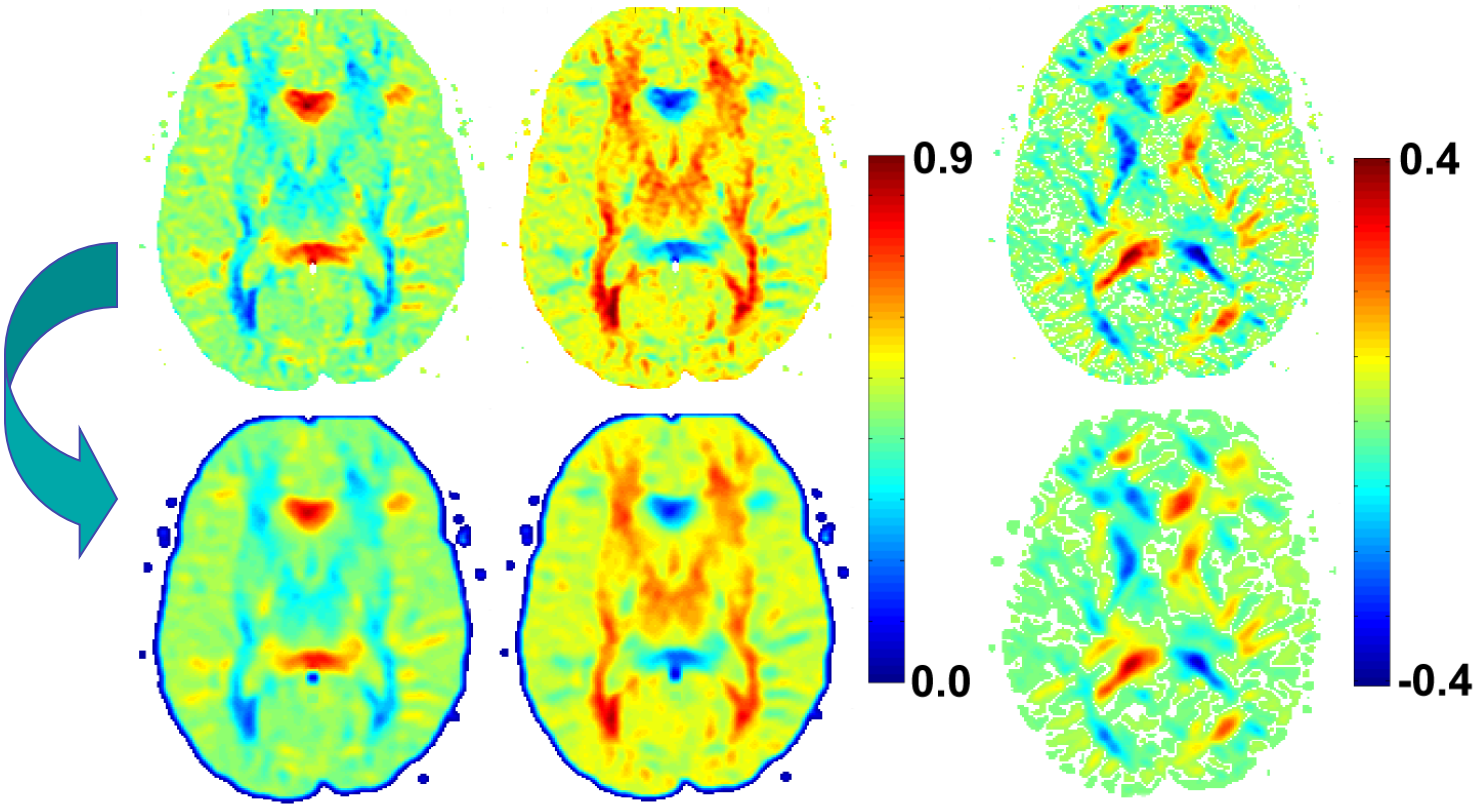

1.0
0.0
FA map
Isotropic
Anisotropic

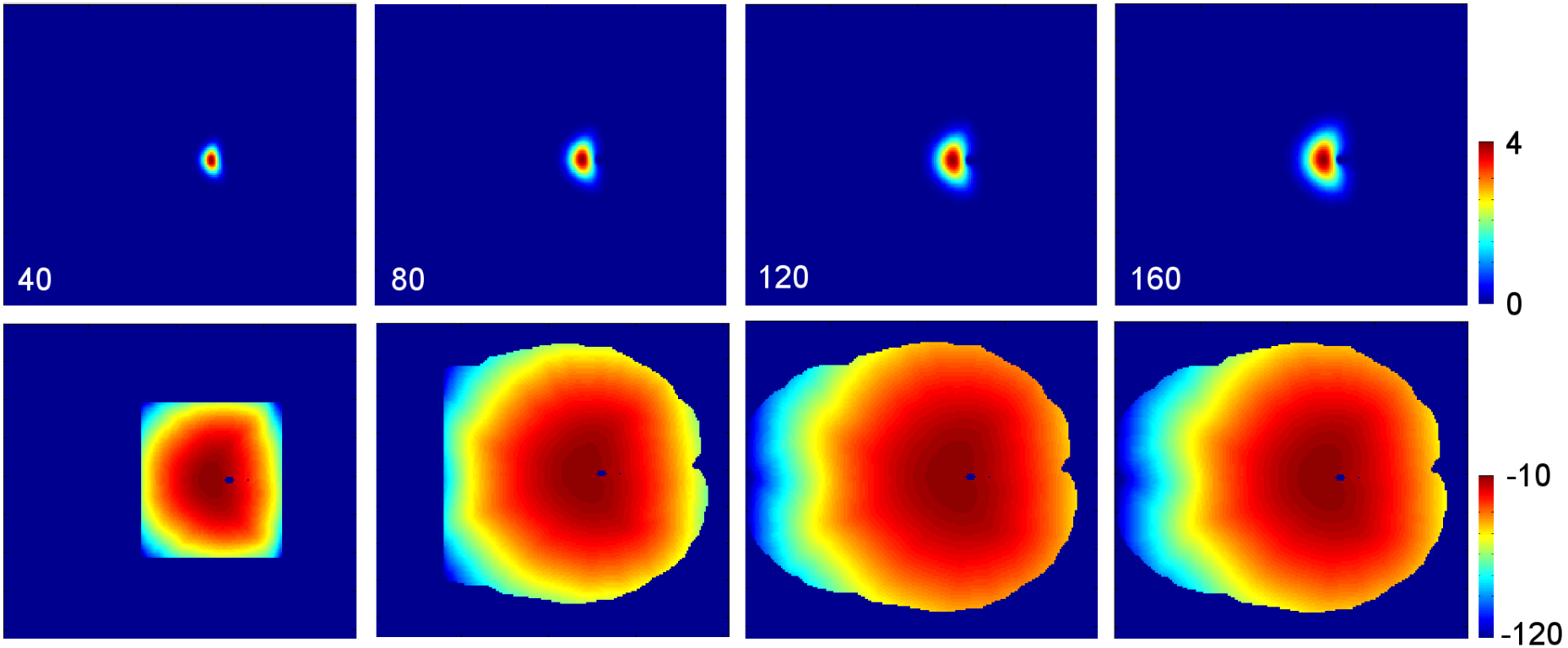
Estimated anisotropic weights

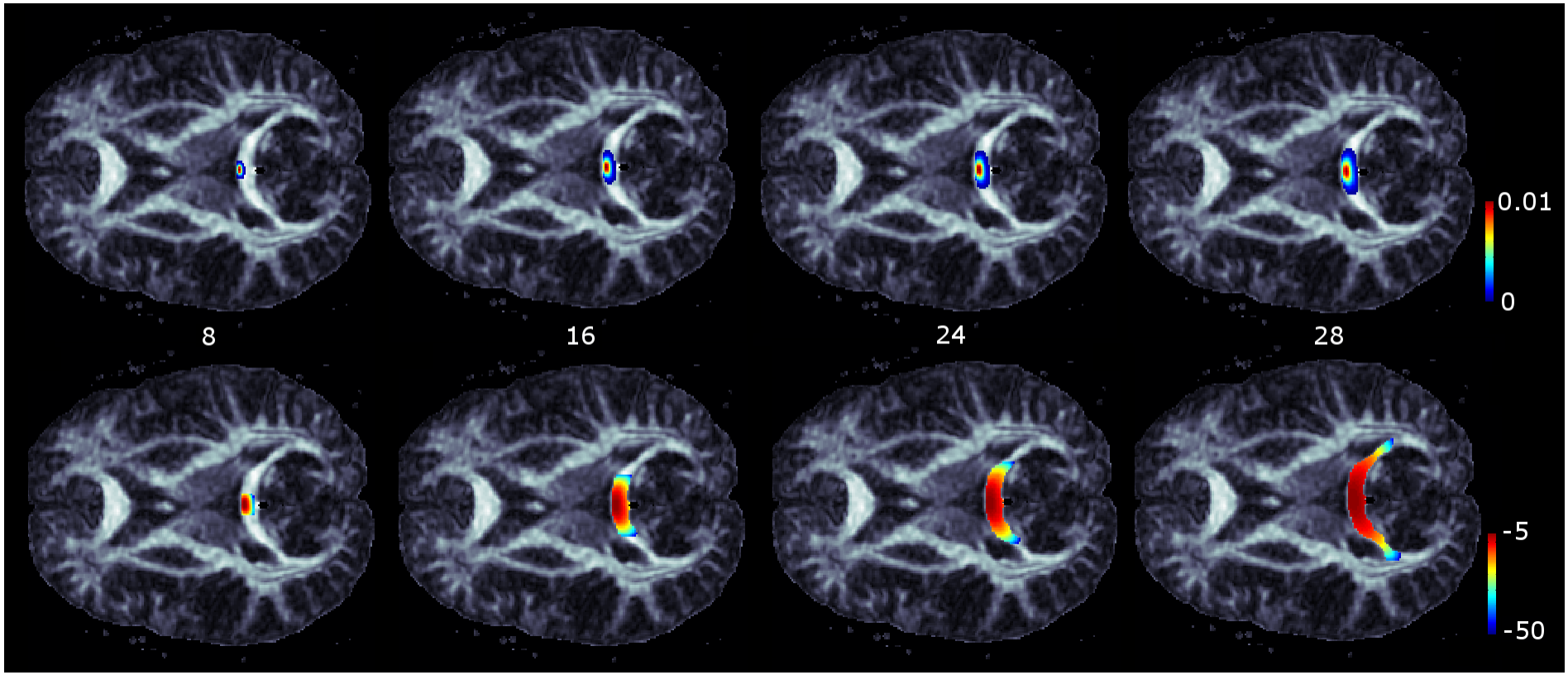


- Due to image noise, images are not smooth enough.
- Isotropic smoothing on the Cholesky factors of DTI is needed to improve the performance.

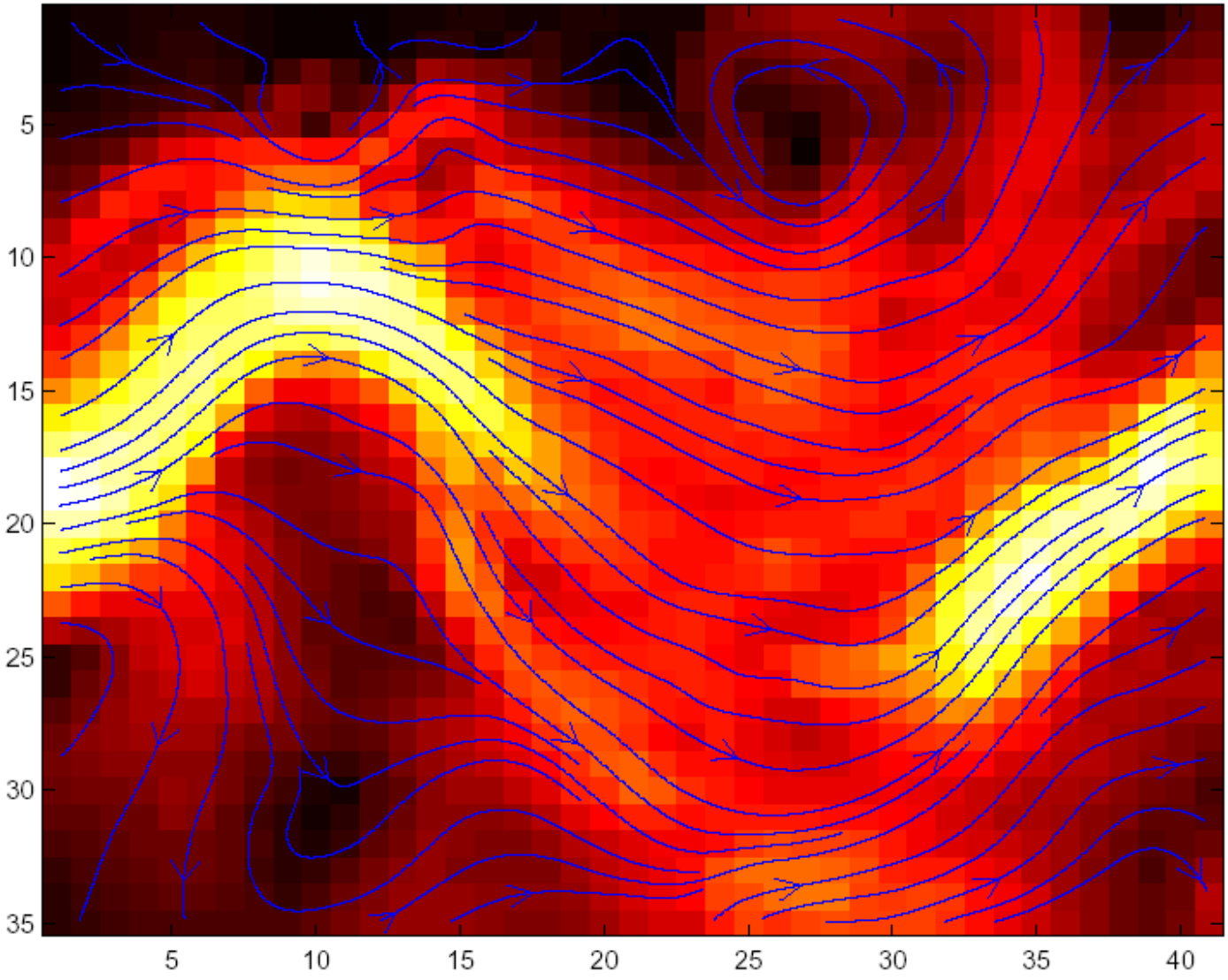
Smooth Cholesky Factors
and reconstruct diffusion coefficients



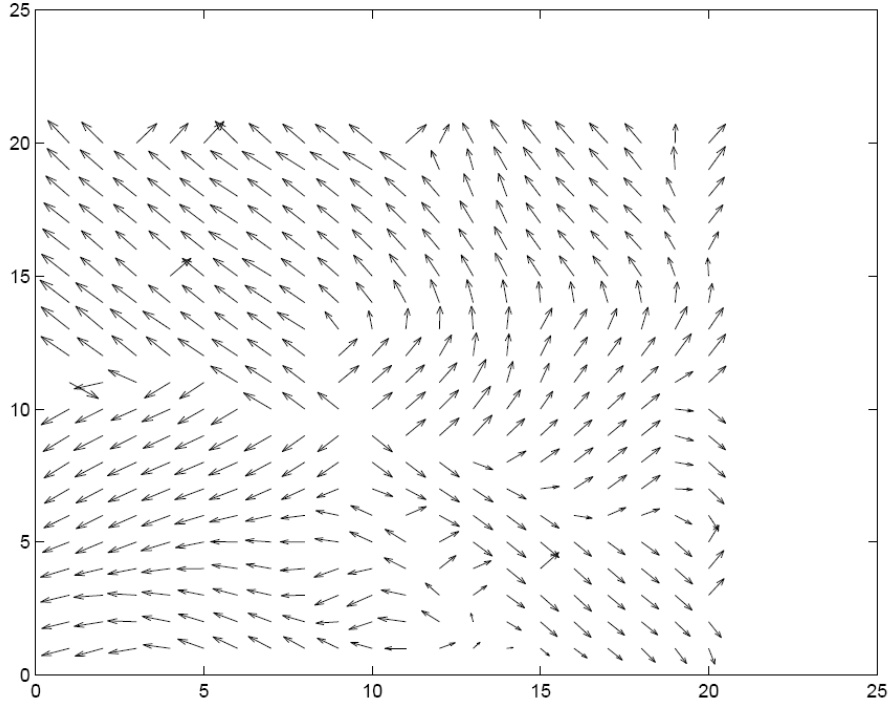

40
80
120
160
4
0
-10
-120

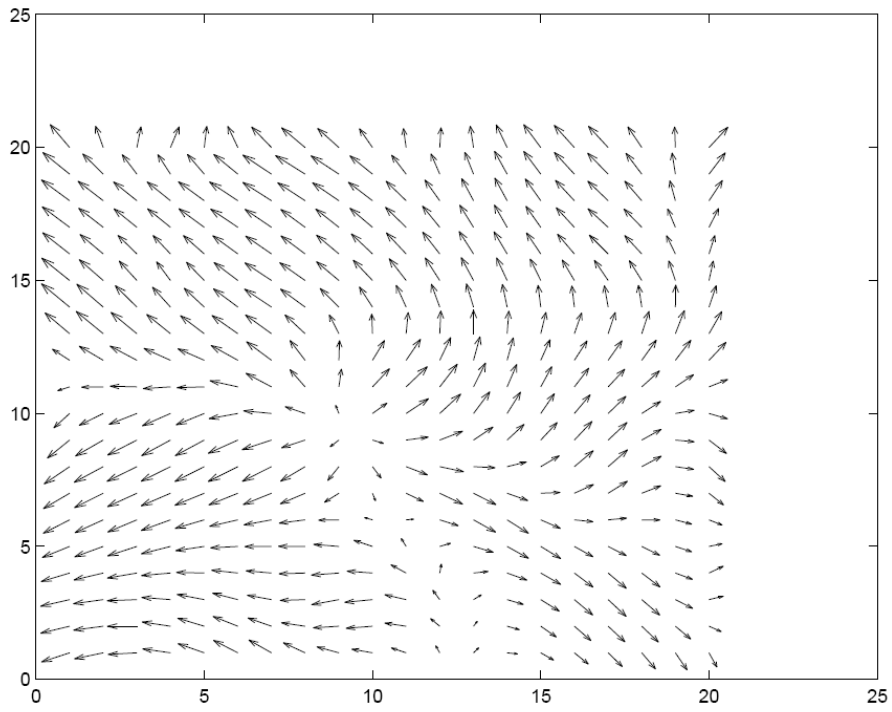

8
16
24
28
0.01
0
-5
-50

Smoothing vector fields: spline approach

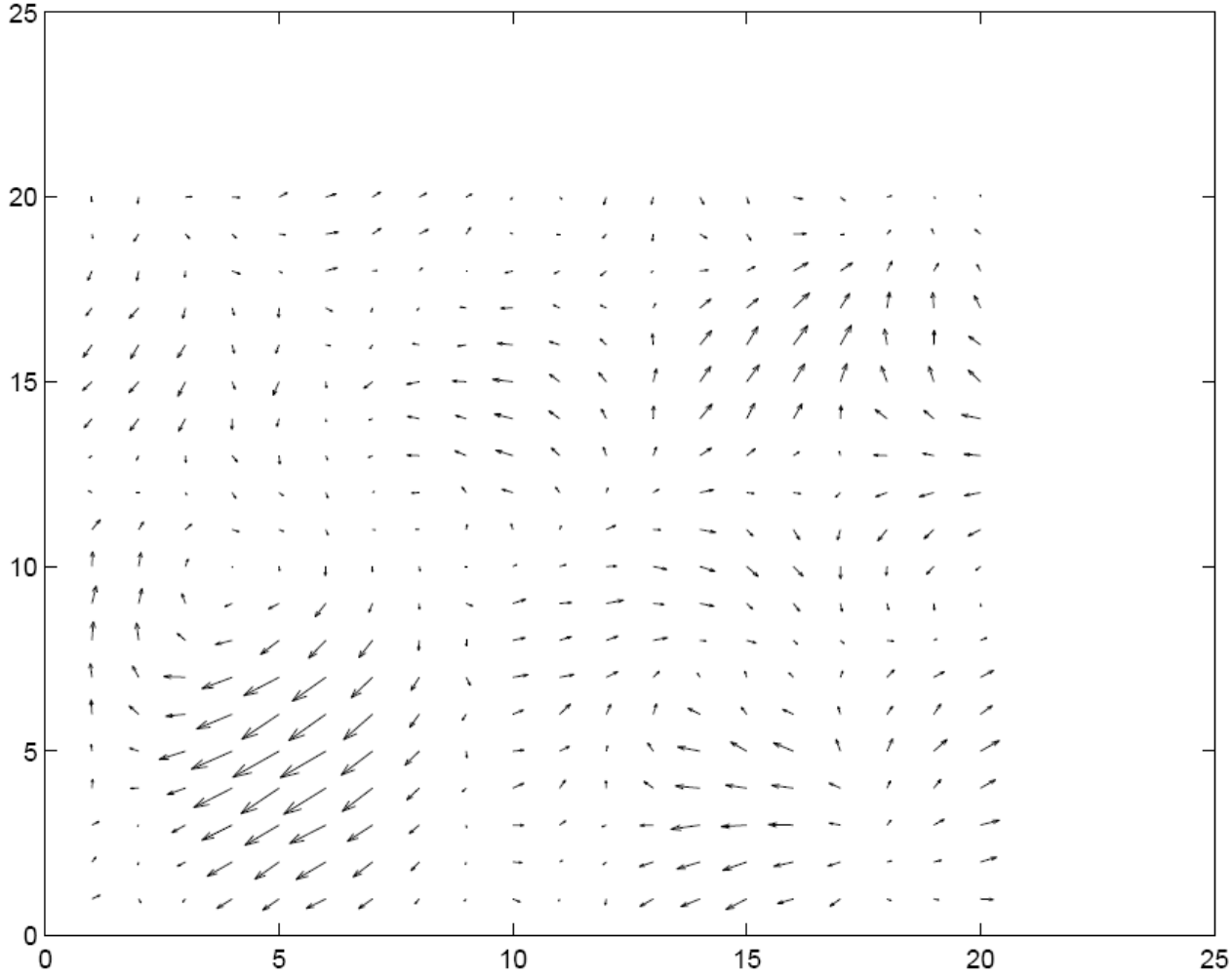


Thin-plate spline smoothing of vector fields.
This is a special case of div-curl spline.





Div-curl spline smoothing



Vector fields are decomposed into rotational (curl) and irrotational components (div).

Helmholtz theorem

The motivation for the vector spline can be seen from the decomposition of 2D vector field f. We assume that field f vanishes at infinity. f can be decomposed into divergent free (solenoidal) and rotation free (irrotational) parts using the Helmholtz decomposition:

$$f = f_{sol} + f_{irr},$$

where $f_{sol} = \nabla \times \psi$ and $f_{irr} = \nabla\phi$ for vector potential field ψ and stream function ϕ. Note that $\nabla \cdot f_{sol} = \nabla \cdot (\nabla \times \psi) = 0$ and $\nabla \times f_{irr} = \nabla \times (\nabla\phi) = 0$.

At each control point $p_j \in \mathbb{R}^d$, we have the principal eigenvector $V(p_j)$. Then we have the following stochastic model

$$V(p_j) = f(p_j) + \epsilon_j$$

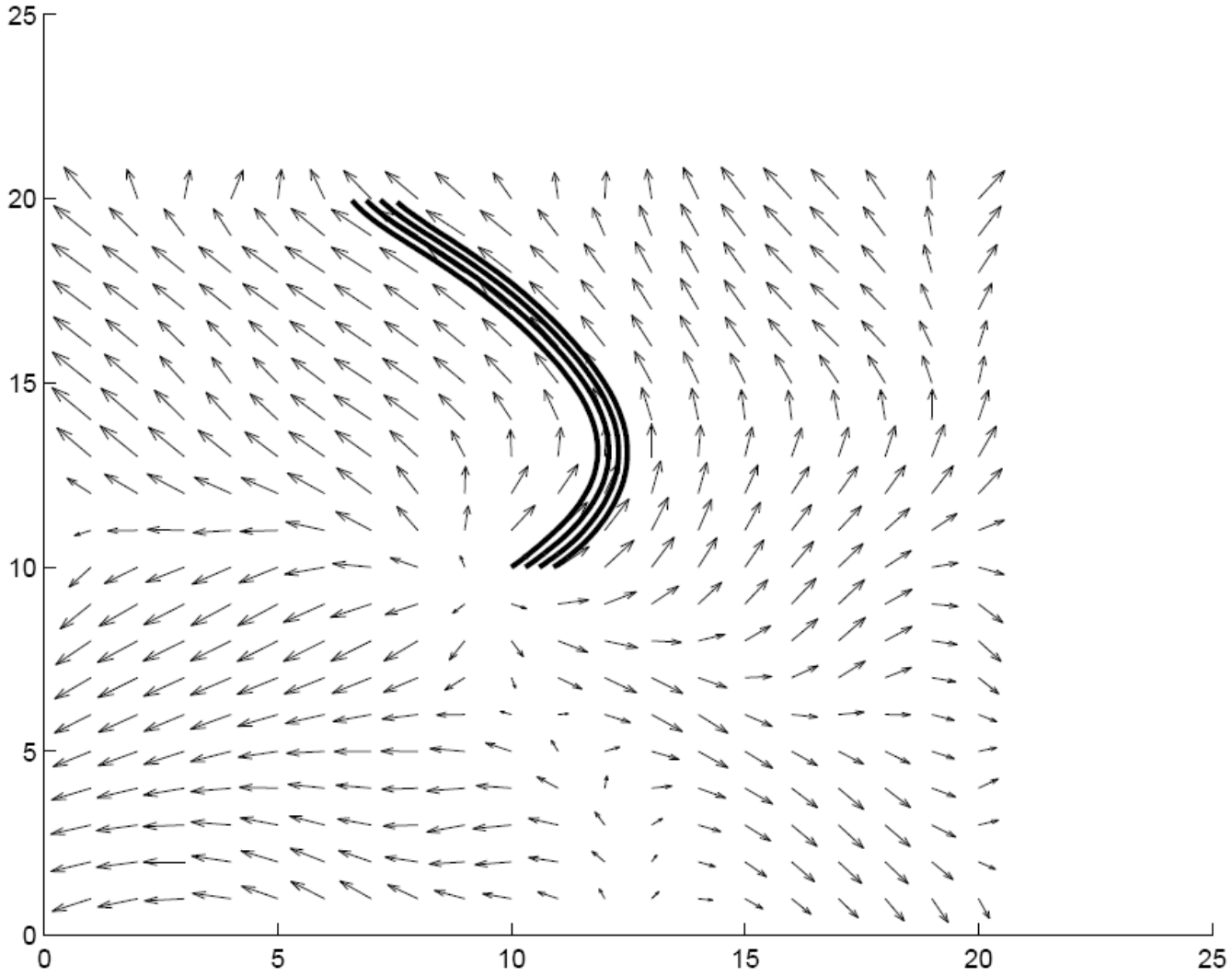
where ϵ_j is assumed to be a Gaussian white noise vector and f is the mean vector-valued function. Then we estimate the mean function f by minimizing

$$\sum_{j=1}^{n} \|V(p_j) - f(p_j)\|^2 + \lambda \int_{\mathbb{R}^2} \alpha \|\nabla(\nabla \cdot f)\|^2 + \beta \|\nabla(\nabla \times f)\|^2 \, dp.$$

This is called the second order vector spline. The first order vector spline has no gradient operator in the integral.

Div-curl spline is not efficient computationally.

Streamlines after thin-plate spline smoothing



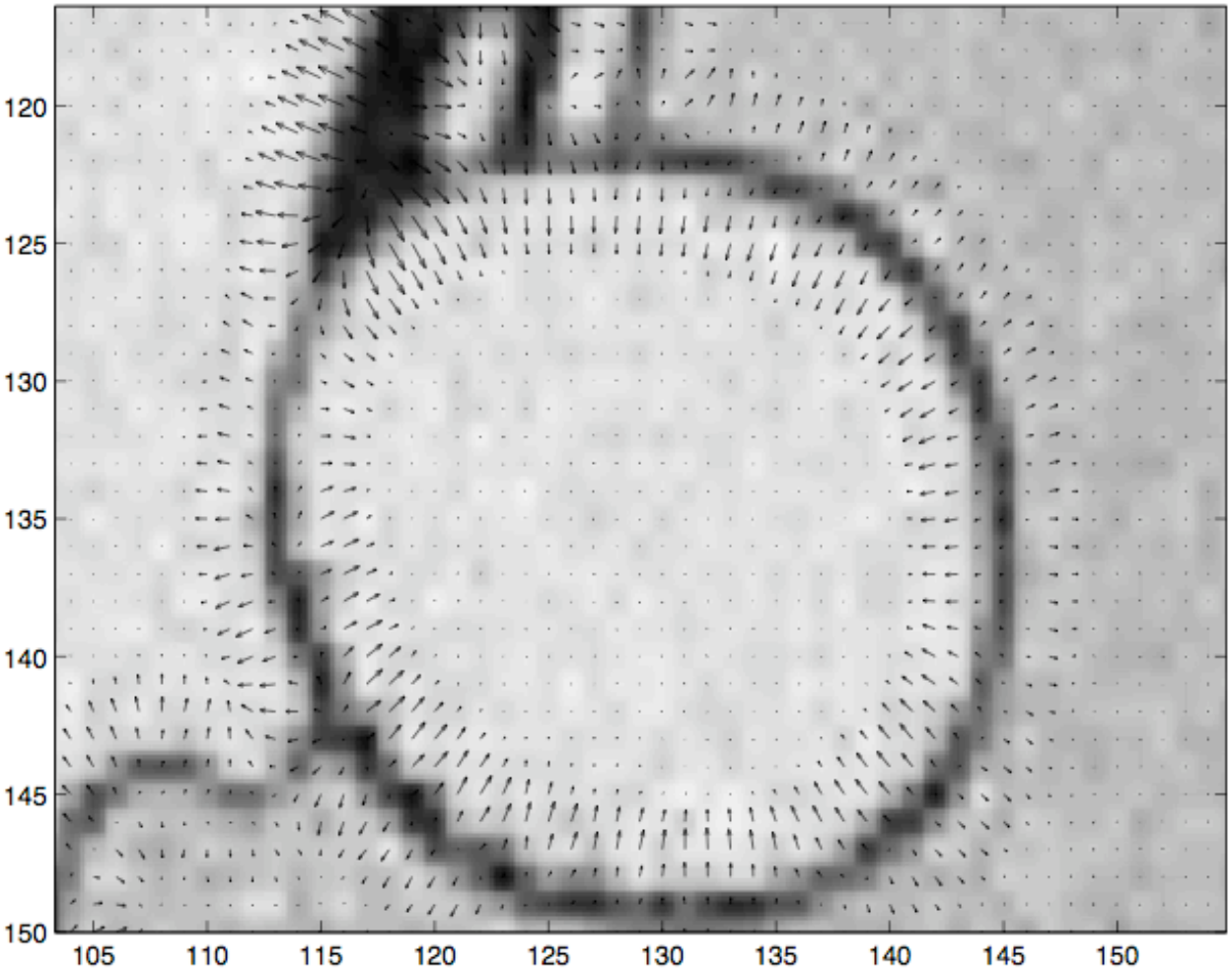
The concentration of water molecules follows the following anisotropic diffusion equation:

$$\frac{\partial C}{\partial t} = \nabla \cdot (D \nabla C)$$

We can use this idea for edge preserving image smoothing by taking D to be related to image gradient such that D obtains high value (more smoothing) in the interior and low value (less smoothing) near edges

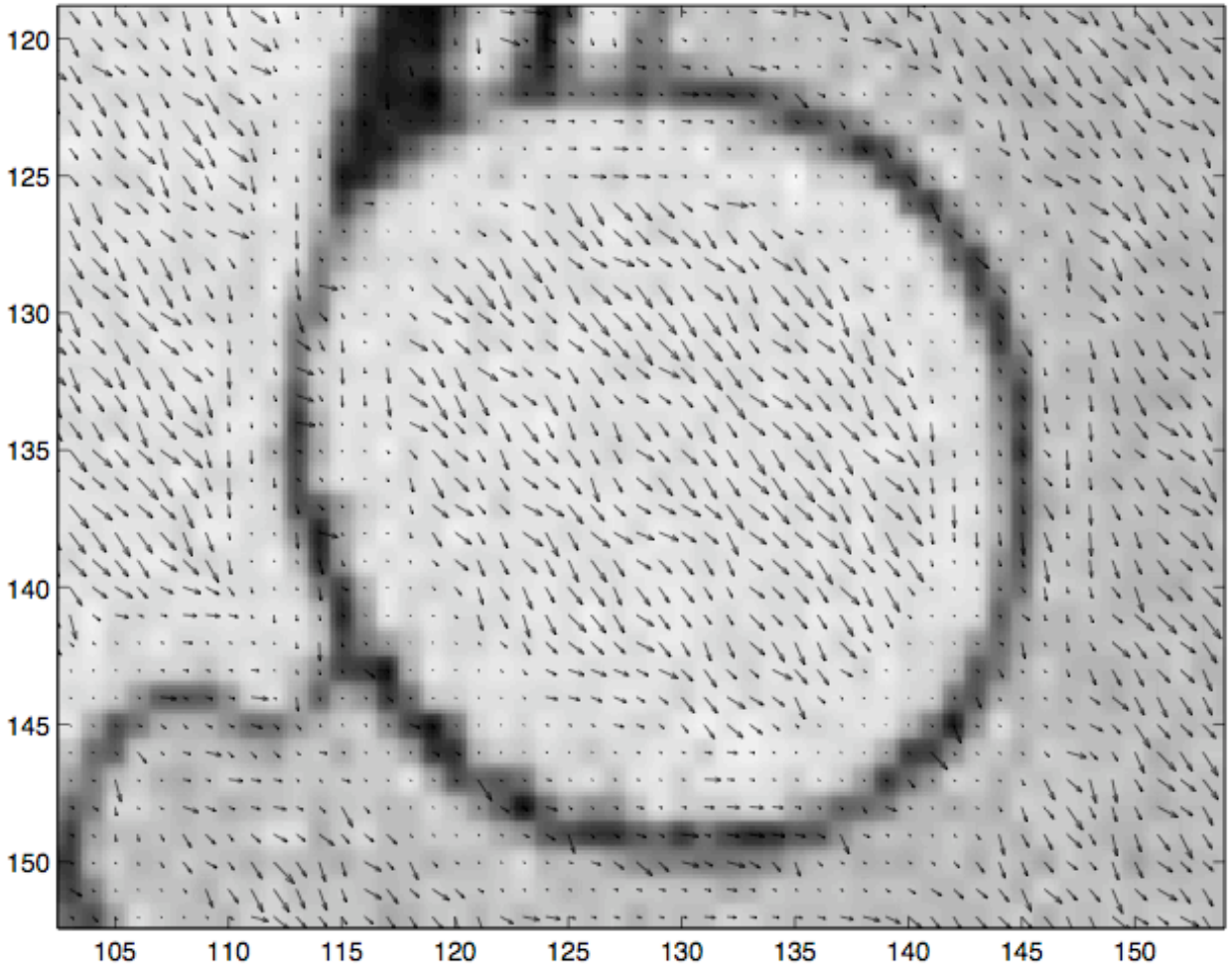
(Perona and malik, 1986).

Image gradient change



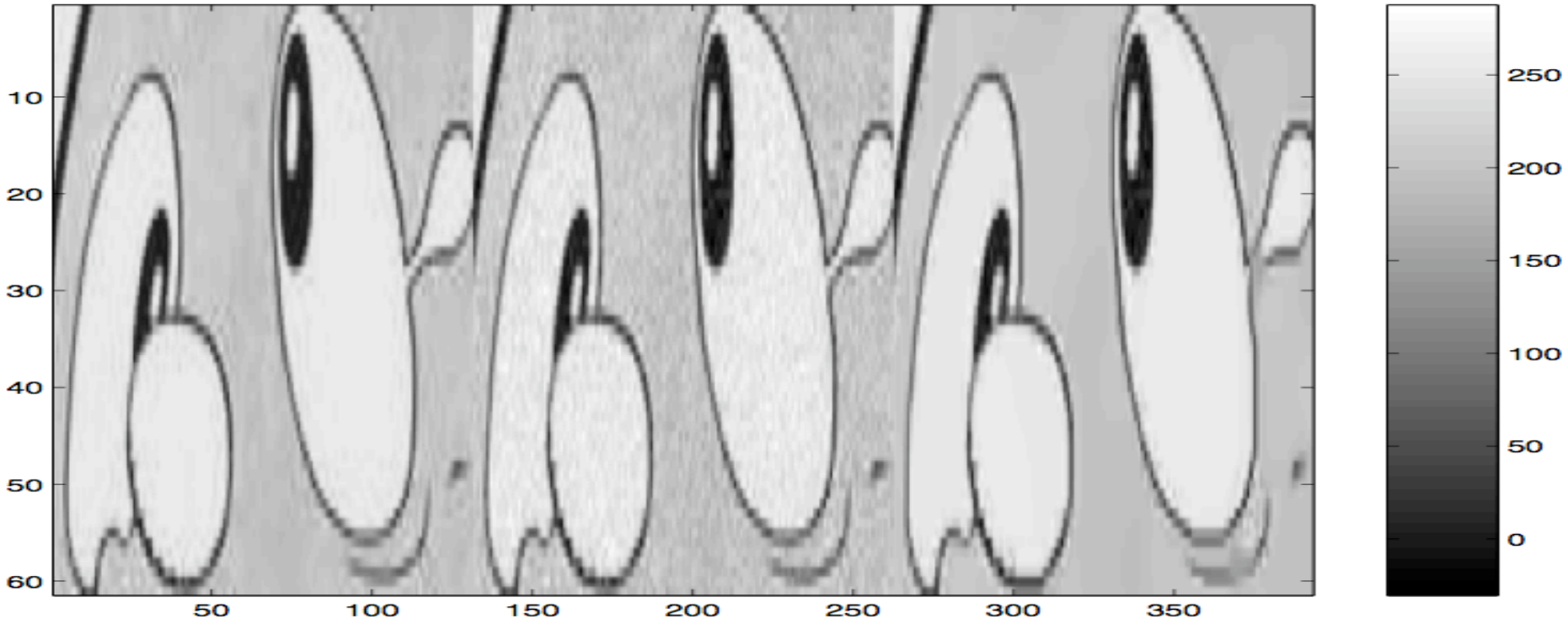
$$\frac{\partial}{\partial x}(K_\sigma * I) \qquad \frac{\partial}{\partial y}(K_\sigma * I)$$

Image gradient change



$$\frac{1}{1+\frac{\partial}{\partial x}(K_\sigma * I)} \qquad \frac{1}{1+\frac{\partial}{\partial y}(K_\sigma * I)}$$

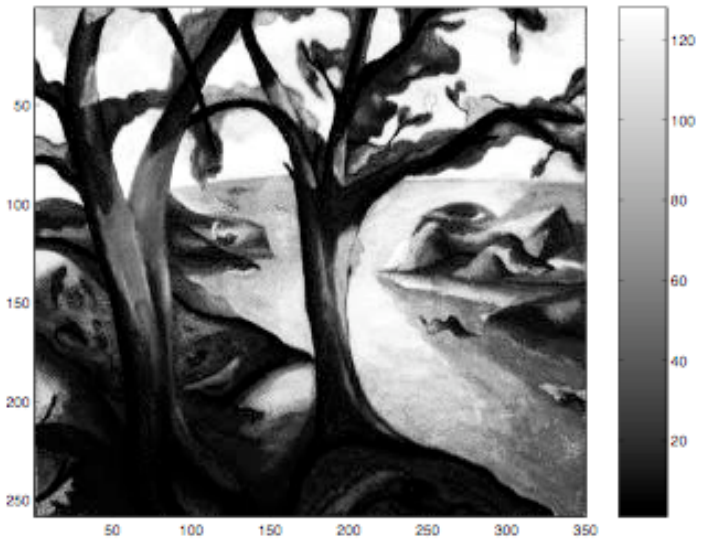
Anisotropic smoothing



$$D = \begin{pmatrix} \dfrac{1}{1 + \dfrac{\partial}{\partial x}(K_\sigma * I)} & 0 \\ 0 & \dfrac{1}{1 + \dfrac{\partial}{\partial y}(K_\sigma * I)} \end{pmatrix}$$

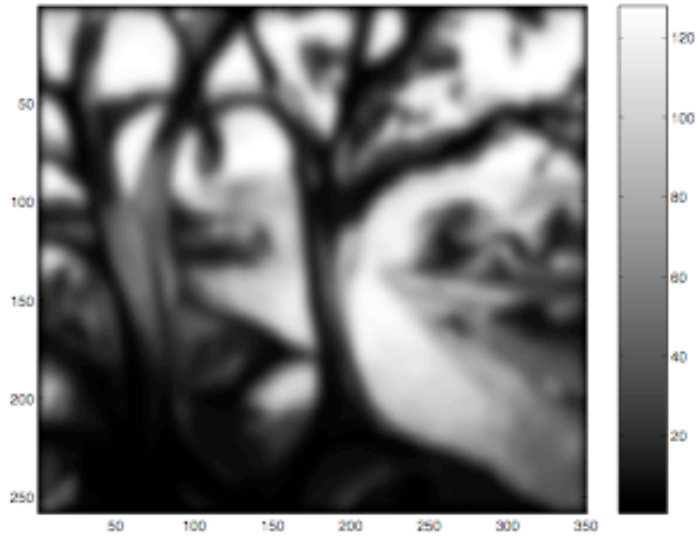
Diffusion coefficients can be used in solving either diffusion equation or kernel smoothing

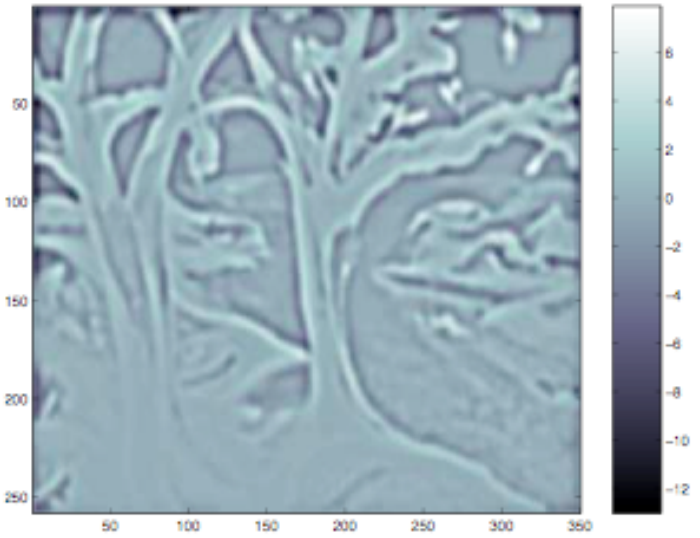
Obtaining edge information: Laplacian



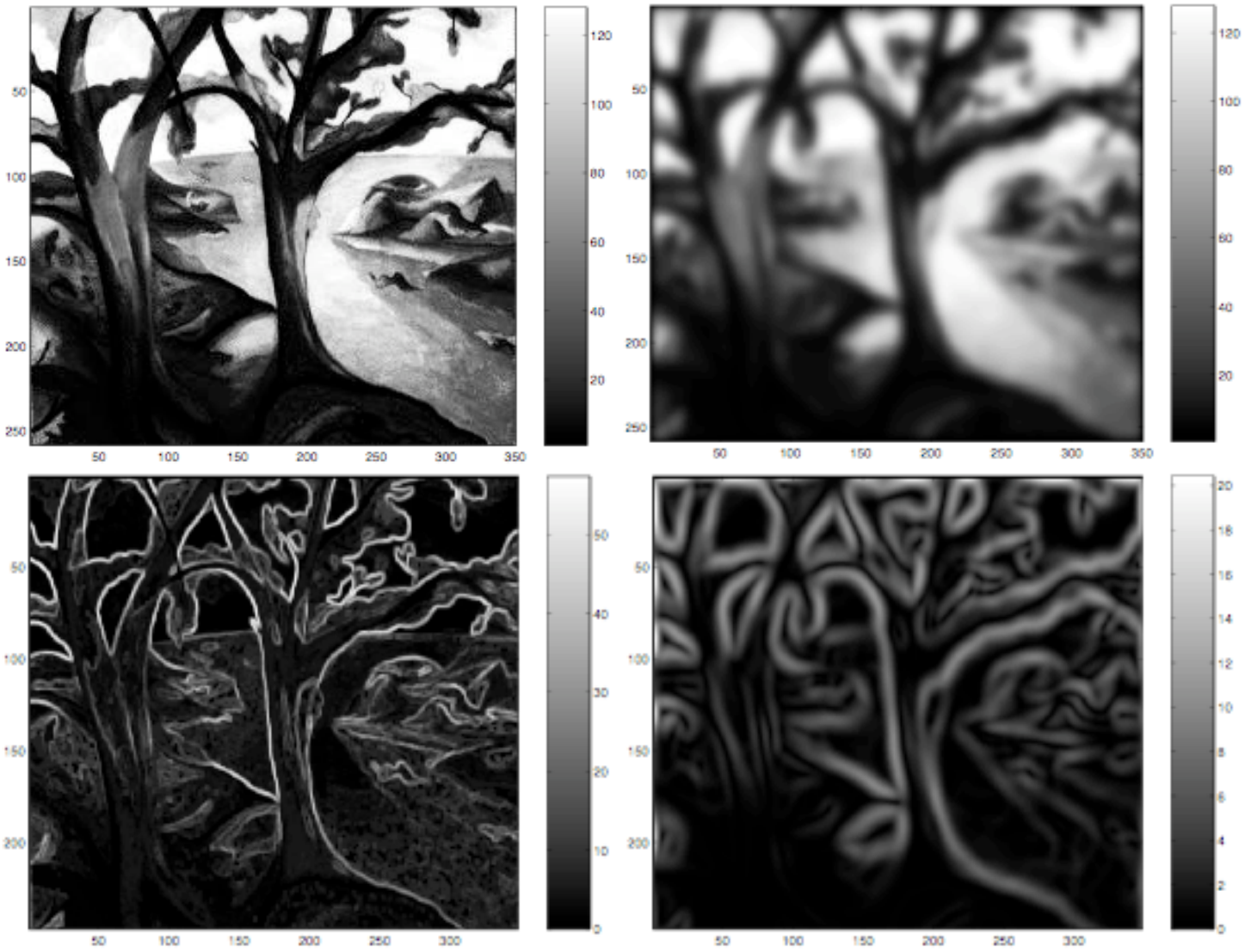
$$(w_{ij}) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$\Delta f = \frac{1}{\delta l^2} \sum_{i,j=-1,0,1} w_{ij} f(x_i, y_i)$$





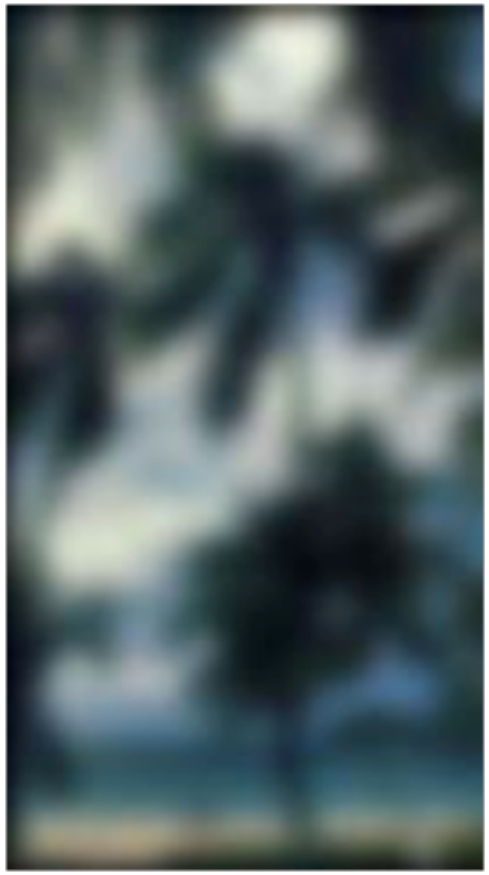
Obtaining edge information: sample variance



Without smoothing

With smoothing

Anisotropic diffusion equation



Original anisotropic diffusion Gaussian smoothing

Michael Bronstein

Smoothing data manifold

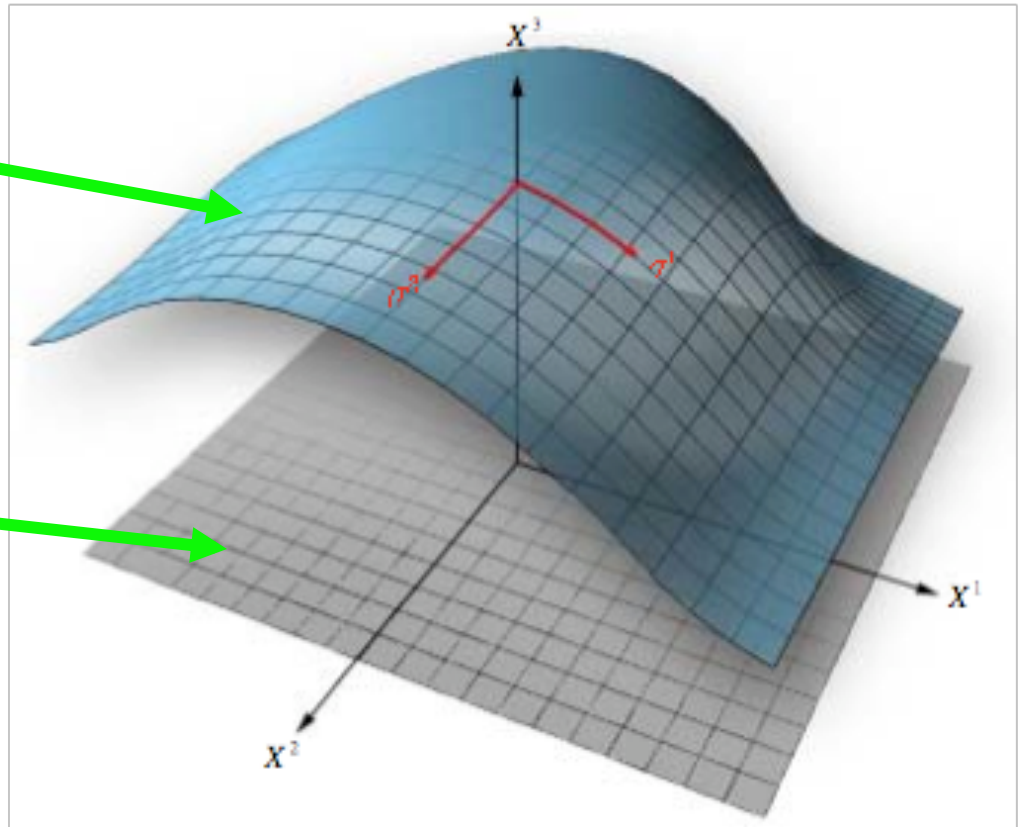
Figure 9. Beltrami flow application on Eugenio Beltrami's portrait [Kimmel et al].

Weights for smoothing is determined by the geodesic distance between two neighboring sample points. Metric tensor approach.

Riemannian metric tensor formulation

Isotropic smoothing in image intensity surface

Anisotropic smoothing in Image space



MATLAB demonstration

Lecture 6 Topics

Diffusion Tensor Image Analysis

Course Project

(50% of your final grade)

By October 23, please send me 1 page abstract of what you will do research and write about.
(approximate length 15 pages without figures, tables, references)

The project topic has to be consulted with me but if you already have some brain images, you can use them. However, methods you will use in the project must be related to course materials.

Possible Course Project

1. Install some software such as Camino or FreeSurfer and do image processing, and do some simple data analysis. You have to install it in Mac Pro we are currently purchasing.

2. Using the in-class data set, do processing and analysis with technique not covered in class

3. Get data set from other professors and follow some of procedures covered in class and do processing and analysis.