Weierstraß-Institut für Angewandte Analysis und Stochastik

J. Polzehl

Structural adaptive smoothing: Images, fMRI and DWI

Leibniz Gemeinschaft

DFG-Forschungszentrum MATHEON
Mathematik für Schlüsseltechnologien

1 Introduction

2 Propagation-Separation approach
- Heuristics
- Algorithm
- Propagation Condition - Selection of Parameters

3 Examples: Images
- 3D Image enhancement
- Locally smooth images
- Color images

4 Functional MRI
- Experimental questions & design
- Analysis of block-design experiments
- Examples

5 Diffusion weighted imaging
- Experimental questions
- Modeling diffusion weighted data
- Accessing diffusion data
- Smoothing diffusion weighted data
- Fibre tracking examples
- Outlook

Introduction

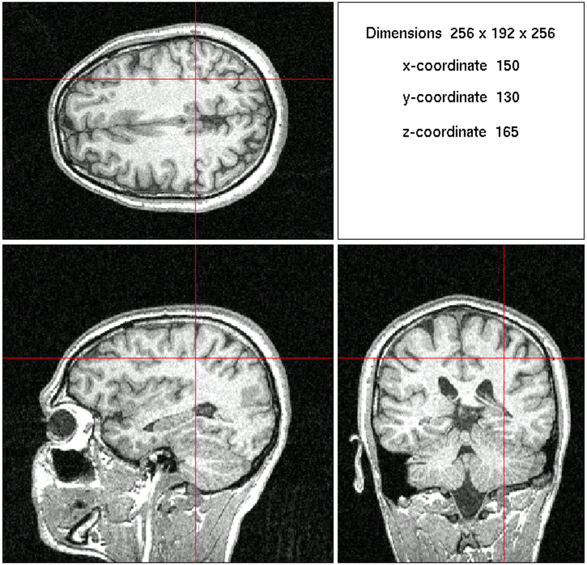
Collaboration

Joint Work with:

- Karsten Tabelow, WIAS and Matheon
- Vladimir Spokoiny, WIAS
- Henning Voss, Weill Medical College, Cornell University

Cooperation

- Citigroup Biomedical Imaging Center, Weill Medical College, Cornell University
- Berlin NeuroImaging Center, Department of Neurology, Charite
- Helios Klinikum Berlin Buch
- University of Tromsœ
- ...



3D-MR-Image

Data:

observations

$Z_i = (X_i, Y_i)$

Image:

structure + distortions

$Y_i = f(X_i) + \varepsilon_i$ or

$Y_i = P_{f(X_i)}$

Interesting structures:

discontinuities and homogeneous regions.

Propagation-Separation approach

Main idea:

Describe the data locally by a simple model P_ϑ, parameterized by ϑ.

Structural assumption:

For each voxel (pixel) X_i there exists a local vicinity $U(X_i)$ such that

$$Y_j \sim P_{\vartheta(X_i)} \quad \forall X_j \in U(X_i).$$

Goal:

Determine for each X_i the region

$$U(X_i) = \{X_j : \vartheta(X_j) \approx \vartheta(X_i)\}$$

from the data and use this information to estimate $\vartheta(X_i)$.

Propagation-Separation approach

Main idea:

Describe the data locally by a simple model P_ϑ, parameterized by ϑ.

Structural assumption:

For each voxel (pixel) X_i there exists a local vicinity $U(X_i)$ such that

$$Y_j \sim P_{\vartheta(X_i)} \quad \forall X_j \in U(X_i).$$

Goal:

Determine for each X_i the region

$$U(X_i) = \{X_j : \vartheta(X_j) \approx \vartheta(X_i)\}$$

from the data and use this information to estimate $\vartheta(X_i)$.

Propagation-Separation approach

Main idea:

Describe the data locally by a simple model P_ϑ, parameterized by ϑ.

Structural assumption:

For each voxel (pixel) X_i there exists a local vicinity $U(X_i)$ such that

$$Y_j \sim P_{\vartheta(X_i)} \quad \forall X_j \in U(X_i).$$

Goal:

Determine for each X_i the region

$$U(X_i) = \{X_j : \vartheta(X_j) \approx \vartheta(X_i)\}$$

from the data and use this information to estimate $\vartheta(X_i)$.

Localization by weights:

A local model in x is described by a diagonal matrix

$$W(x) = \text{diag}\{w_1(x), \ldots w_n(x)\}.$$

Observations with positive weights contribute to estimates of $\vartheta(x)$.

Example: Kernel weights $w_j(x) = K(\frac{X_j - x}{h})$.

Estimation by weighted (localized) likelihood

$$\hat{\vartheta}(x) = \sum_j^n w_j(x) Y_j / \sum_j^n w_j(x)$$

or local least squares.

Localization by weights:

A local model in x is described by a diagonal matrix

$$W(x) = \text{diag}\{w_1(x), \dots w_n(x)\}.$$

Observations with positive weights contribute to estimates of $\vartheta(x)$.

Example: Kernel weights $w_j(x) = K(\frac{X_j - x}{h})$.

Estimation by weighted (localized) likelihood

$$\hat{\vartheta}(x) = \sum_j^n w_j(x) Y_j / \sum_j^n w_j(x)$$

or local least squares.

Localization by weights:

A local model in x is described by a diagonal matrix

$$W(x) = \text{diag}\{w_1(x), \ldots w_n(x)\}.$$

Observations with positive weights contribute to estimates of $\vartheta(x)$.

Example: Kernel weights $w_j(x) = K(\frac{X_j - x}{h})$.

Estimation by weighted (localized) likelihood

$$\hat{\vartheta}(x) = \sum_j^n w_j(x) Y_j / \sum_j^n w_j(x)$$

or local least squares.

Ideal choice of weights: $w_j(x) > 0 \Longleftrightarrow \vartheta(X_i) \approx \vartheta(x)$

Structural adaptation: Recover the structure (weights) from the data and structural assumptions

Idea: Allow for data-dependent weights $w_j(x)$. Generate weights such that

- Propagation within regions with similar parameters
- Separation between regions if parameters are different.

Ideal choice of weights: $w_j(x) > 0 \Longleftrightarrow \vartheta(X_i) \approx \vartheta(x)$

Structural adaptation: Recover the structure (weights) from the data and structural assumptions

Idea: Allow for data-dependent weights $w_j(x)$. Generate weights such that

- Propagation within regions with similar parameters
- Separation between regions if parameters are different.

Ideal choice of weights: $w_j(x) > 0 \Longleftrightarrow \vartheta(X_i) \approx \vartheta(x)$

Structural adaptation: Recover the structure (weights) from the data and structural assumptions

Idea: Allow for data-dependent weights $w_j(x)$. Generate weights such that

- Propagation within regions with similar parameters
- Separation between regions if parameters are different.

Oracles (for P_ϑ fixed)

Structural assumption: $\vartheta(x)$ is nearly constant in some vicinity $U(x)$.

- $U(x)$ known in x:
  - define a local model by

$$W(x) = (w_1(x), \ldots, w_N(x)) \quad \text{with} \quad w_i(x) = I_{X_i \in U(x)}.$$

  - compute local likelihood estimate $\hat{\vartheta}(x)$.

- Estimates $\hat{\vartheta}(x)$ given:
  - for (i, j) with $\rho(x, X_j) \le h$ test a hypothesis of homogeneity $H : \vartheta(X_j) = \vartheta(x)$.
  - test statistics $T_j(x)$,
  - use $T_j(x)$ to assign a weight $w_j(x)$.

Both oracles can be used to design an iterative procedure that

- recovers the homogeneity structure and
- efficiently estimates the local parameters.

Oracles (for P_ϑ fixed)

Structural assumption: $\vartheta(x)$ is nearly constant in some vicinity $U(x)$.

- $U(x)$ known in x:
  - define a local model by

$$W(x) = (w_1(x), \dots, w_N(x)) \quad \text{with} \quad w_i(x) = I_{X_i \in U(x)}.$$

  - compute local likelihood estimate $\hat{\vartheta}(x)$.

- Estimates $\hat{\vartheta}(x)$ given:
  - for (i,j) with $\rho(x, X_j) \leq h$ test a hypothesis of homogeneity $H : \vartheta(X_j) = \vartheta(x)$.
  - test statistics $T_j(x)$,
  - use $T_j(x)$ to assign a weight $w_j(x)$.

Both oracles can be used to design an iterative procedure that

- recovers the homogeneity structure and
- efficiently estimates the local parameters.

Oracles (for P_ϑ fixed)

Structural assumption: $\vartheta(x)$ is nearly constant in some vicinity $U(x)$.

- $U(x)$ known in x:
  - define a local model by

$$W(x) = (w_1(x), \ldots, w_N(x)) \quad \text{with} \quad w_i(x) = I_{X_i \in U(x)}.$$

  - compute local likelihood estimate $\hat{\vartheta}(x)$.

- Estimates $\hat{\vartheta}(x)$ given:
  - for (i, j) with $\rho(x, X_j) \leq h$ test a hypothesis of homogeneity $H : \vartheta(X_j) = \vartheta(x)$.
  - test statistics $T_j(x)$,
  - use $T_j(x)$ to assign a weight $w_j(x)$.

Both oracles can be used to design an iterative procedure that

- recovers the homogeneity structure and
- efficiently estimates the local parameters.

Generating new weights (local likelihood)

Estimate the structure: Given estimates $\hat{\vartheta}_i = \hat{\vartheta}(X_i)$
assign a weight w_{ij} for every pair (i,j) with $\rho(X_i, X_j) \le h$ by testing

$$H : \vartheta_j = \vartheta_i$$

Test statistic (local likelihood):

$$T_{ij} = N_i \, \mathcal{K}(\hat{\vartheta}_j, \hat{\vartheta}_i) \quad \text{with} \quad N_i = \sum_l w_{il}$$

Define new weights using two kernel functions K_{loc} and K_{st} as

$$w_{ij} = K_{\text{loc}}(\boldsymbol{l}_{ij}) \, K_{\text{st}}(\boldsymbol{s}_{ij})$$

Localization penalty: $\boldsymbol{l}_{ij} = \frac{\rho(X_i, X_j)}{h}$ Statistical penalty: $\boldsymbol{s}_{ij} = \frac{T_{ij}}{\lambda}$

Generating new weights (local likelihood)

Estimate the structure: Given estimates $\hat{\vartheta}_i = \hat{\vartheta}(X_i)$ assign a weight w_{ij} for every pair (i,j) with $\rho(X_i, X_j) \le h$ by testing

$$H : \vartheta_j = \vartheta_i$$

Test statistic (local likelihood):

$$T_{ij} = N_i \, \mathcal{K}(\hat{\vartheta}_j, \hat{\vartheta}_i) \qquad \text{with} \qquad N_i = \sum_l w_{il}$$

Define new weights using two kernel functions K_{loc} and K_{st} as

$$w_{ij} \quad = \quad K_{\text{loc}}\left(\boldsymbol{l}_{ij}\right) K_{\text{st}}\left(\boldsymbol{s}_{ij}\right)$$

Localization penalty: $\boldsymbol{l}_{ij} = \frac{\rho(X_i, X_j)}{h}$ Statistical penalty: $\boldsymbol{s}_{ij} = \frac{T_{ij}}{\lambda}$

Generating new weights (local likelihood)

Estimate the structure: Given estimates $\hat{\vartheta}_i = \hat{\vartheta}(X_i)$
assign a weight w_{ij} for every pair (i,j) with $\rho(X_i, X_j) \leq h$ by testing

$$H : \vartheta_j = \vartheta_i$$

Test statistic (local likelihood):

$$T_{ij} = N_i \, \mathcal{K}(\hat{\vartheta}_j, \hat{\vartheta}_i) \qquad \text{with} \qquad N_i = \sum_l w_{il}$$

Define new weights using two kernel functions K_{loc} and K_{st} as

$$w_{ij} \quad = \quad K_{\text{loc}}\left(\boldsymbol{l}_{ij}\right) K_{\text{st}}\left(\boldsymbol{s}_{ij}\right)$$

Localization penalty: $\boldsymbol{l}_{ij} = \frac{\rho(X_i, X_j)}{h}$ Statistical penalty: $\boldsymbol{s}_{ij} = \frac{T_{ij}}{\lambda}$

Initialization: For every i, set $\hat{\vartheta}_i^{(0)} = n^{-1} \sum_i Y_i$, $N_i^{(0)} = n$ and $W_i^{(0)} = I$. Set $k = 1$ and initialize $h^{(1)}$.

Adaptation: For every i, j, compute

$$\boldsymbol{l}_{ij} = \rho(X_i, X_j)/h^{(k)}, \quad \boldsymbol{s}_{ij} = \lambda^{-1} N_i^{(k-1)} \mathcal{K}(\hat{\vartheta}_j^{(k-1)}, \hat{\vartheta}_i^{(k-1)})$$

and weights $W_i = \text{diag}\{w_{i1}, \dots, w_{in}\}$ as $w_{ij} = K_{\text{loc}}(\boldsymbol{l}_{ij}) K_{\text{st}}(\boldsymbol{s}_{ij})$,

Local estimation: compute new local MLE estimates

$$\hat{\vartheta}_i^{(k)} = \hat{S}_i^{(k)} / \hat{N}_i^{(k)} \qquad \text{with} \qquad \hat{N}_i^{(k)} = \sum_l w_{il}, \quad \hat{S}_i^{(k)} = \sum_l w_{il} Y_l$$

Stopping: Stop if $h^{(k)} \geq h_{max}$, otherwise increase k by 1, set $h^{(k)} = a_h h^{(k-1)}$ and continue with the adaptation step.

Initialization: For every i, set $\hat{\vartheta}_i^{(0)} = n^{-1} \sum_i Y_i$, $N_i^{(0)} = n$ and $W_i^{(0)} = I$. Set $k = 1$ and initialize $h^{(1)}$.

Adaptation: For every i, j, compute

$$\boldsymbol{l}_{ij} = \rho(X_i, X_j)/h^{(k)}, \quad \boldsymbol{s}_{ij} = \lambda^{-1} N_i^{(k-1)} \mathcal{K}(\hat{\vartheta}_j^{(k-1)}, \hat{\vartheta}_i^{(k-1)})$$

and weights $W_i = \operatorname{diag}\{w_{i1}, \ldots, w_{in}\}$ as $w_{ij} = K_{\text{loc}}(\boldsymbol{l}_{ij}) K_{\text{st}}(\boldsymbol{s}_{ij})$,

Local estimation: compute new local MLE estimates

$$\hat{\vartheta}_i^{(k)} = \hat{S}_i^{(k)}/\hat{N}_i^{(k)} \qquad \text{with} \qquad \hat{N}_i^{(k)} = \sum_l w_{il}, \quad \hat{S}_i^{(k)} = \sum_l w_{il} Y_l$$

Stopping: Stop if $h^{(k)} \geq h_{max}$, otherwise increase k by 1, set $h^{(k)} = a_h h^{(k-1)}$ and continue with the adaptation step.

Initialization: For every i, set $\hat{\vartheta}_i^{(0)} = n^{-1} \sum_i Y_i$, $N_i^{(0)} = n$ and $W_i^{(0)} = I$. Set $k = 1$ and initialize $h^{(1)}$.

Adaptation: For every i, j, compute

$$\boldsymbol{l}_{ij} = \rho(X_i, X_j)/h^{(k)}, \quad \boldsymbol{s}_{ij} = \lambda^{-1} N_i^{(k-1)} \mathcal{K}(\hat{\vartheta}_j^{(k-1)}, \hat{\vartheta}_i^{(k-1)})$$

and weights $W_i = \text{diag}\{w_{i1}, \ldots, w_{in}\}$ as $w_{ij} = K_{\text{loc}}(\boldsymbol{l}_{ij}) K_{\text{st}}(\boldsymbol{s}_{ij})$,

Local estimation: compute new local MLE estimates

$$\hat{\vartheta}_i^{(k)} = \hat{S}_i^{(k)} / \hat{N}_i^{(k)} \qquad \text{with} \qquad \hat{N}_i^{(k)} = \sum_l w_{il}, \quad \hat{S}_i^{(k)} = \sum_l w_{il} Y_l$$

Stopping: Stop if $h^{(k)} \geq h_{max}$, otherwise increase k by 1, set $h^{(k)} = a_h h^{(k-1)}$ and continue with the adaptation step.

Initialization: For every i, set $\hat{\vartheta}_i^{(0)} = n^{-1}\sum_i Y_i,\ N_i^{(0)} = n$ and $W_i^{(0)} = I$. Set $k = 1$ and initialize $h^{(1)}$.

Adaptation: For every i, j, compute

$$\boldsymbol{l}_{ij} = \rho(X_i, X_j)/h^{(k)}, \quad \boldsymbol{s}_{ij} = \lambda^{-1} N_i^{(k-1)}\ \mathcal{K}(\hat{\vartheta}_j^{(k-1)}, \hat{\vartheta}_i^{(k-1)})$$

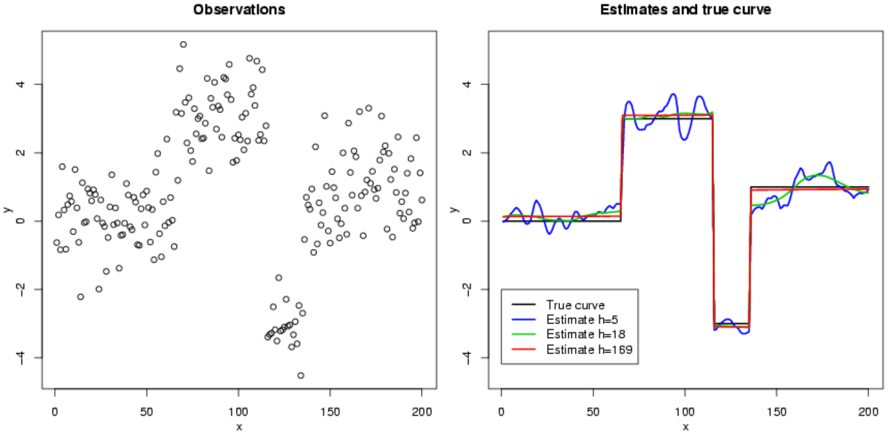
and weights $W_i = \mathrm{diag}\{w_{i1}, \dots, w_{in}\}$ as $w_{ij} = K_{\mathrm{loc}}(\boldsymbol{l}_{ij}) K_{\mathrm{st}}(\boldsymbol{s}_{ij})$,

Local estimation: compute new local MLE estimates

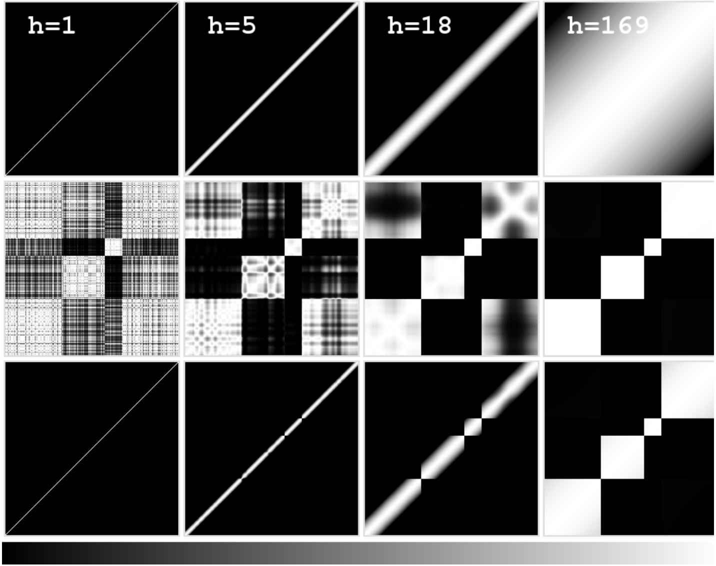
$$\hat{\vartheta}_i^{(k)} = \hat{S}_i^{(k)}/\hat{N}_i^{(k)} \qquad \text{with} \qquad \hat{N}_i^{(k)} = \sum_l w_{il}, \quad \hat{S}_i^{(k)} = \sum_l w_{il} Y_l$$

Stopping: Stop if $h^{(k)} \geq h_{max}$, otherwise increase k by 1, set $h^{(k)} = a_h h^{(k-1)}$ and continue with the adaptation step.

Model: $Y_i = f(X_i) + \varepsilon_i$ $\quad \mathsf{E}\,\varepsilon_i = 0 \;\; \mathsf{Var}\,\varepsilon_i = 1$



Data and estimates for bandwidths $h = 5, 18, 169$. True curve in black.



Propagation of weights

$K_{\text{loc}}(l_{ij})$

$K_{\text{st}}(s_{ij})$

w_{ij}

Scale: $(0, 1)$

Propagation condition:

$\hat{\vartheta}^{(k)}(x)$	PS estimate of $\vartheta(x)$	from iteration k,
$\tilde{\vartheta}^{(k)}(x)$	Kernel estimate of $\vartheta(x)$	using bandwidth $h^{(k)}$

Let $\vartheta(x) \equiv \vartheta$. Select parameters such that $\forall k$ and some $\alpha > 0$

$$\mathbf{E}\,|\hat{\vartheta}^{(\mathbf{k})}(\mathbf{x}) - \vartheta| \leq (\mathbf{1} + \alpha)\,\mathbf{E}\,|\tilde{\vartheta}^{(\mathbf{k})}(\mathbf{x}) - \vartheta|.$$

Main parameters:

- λ : scale in the stochastic penalty

$$s_{ij} = \lambda^{-1} N_i^{(k-1)}\, \mathcal{K}(\hat{\vartheta}_i^{(k-1)}, \hat{\vartheta}_j^{(k-1)})\,.$$

  λ can be selected using the Propagation condition. Selection depends only on the model, not on the data or structure.

- Maximal bandwidth h_{max} : controls smoothness within homogeneous regions and maximal possible variance reduction. Stopping criterion.

Propagation condition:

$\hat{\vartheta}^{(k)}(x)$	PS estimate of $\vartheta(x)$	from iteration k,
$\tilde{\vartheta}^{(k)}(x)$	Kernel estimate of $\vartheta(x)$	using bandwidth $h^{(k)}$

Let $\vartheta(x) \equiv \vartheta$. Select parameters such that $\forall k$ and some $\alpha > 0$

$$\mathbf{E}\, |\hat{\vartheta}^{(\mathbf{k})}(\mathbf{x}) - \vartheta| \leq (\mathbf{1} + \alpha)\, \mathbf{E}\, |\tilde{\vartheta}^{(\mathbf{k})}(\mathbf{x}) - \vartheta|.$$

Main parameters:

- λ : scale in the stochastic penalty

$$\mathbf{s}_{ij} = \lambda^{-1} N_i^{(k-1)}\, \mathcal{K}(\hat{\vartheta}_i^{(k-1)}, \hat{\vartheta}_j^{(k-1)})\,.$$

  λ can be selected using the Propagation condition. Selection depends only on the model, not on the data or structure.

- Maximal bandwidth h_{max} : controls smoothness within homogeneous regions and maximal possible variance reduction. Stopping criterion.

Propagation condition:

$\hat{\vartheta}^{(k)}(x)$	PS estimate of $\vartheta(x)$	from iteration k,
$\tilde{\vartheta}^{(k)}(x)$	Kernel estimate of $\vartheta(x)$	using bandwidth $h^{(k)}$

Let $\vartheta(x) \equiv \vartheta$. Select parameters such that $\forall k$ and some $\alpha > 0$

$$\mathbf{E}\,|\hat{\vartheta}^{(\mathbf{k})}(\mathbf{x}) - \vartheta| \leq (\mathbf{1} + \alpha)\;\mathbf{E}\,|\tilde{\vartheta}^{(\mathbf{k})}(\mathbf{x}) - \vartheta|.$$

Main parameters:

- λ : scale in the stochastic penalty

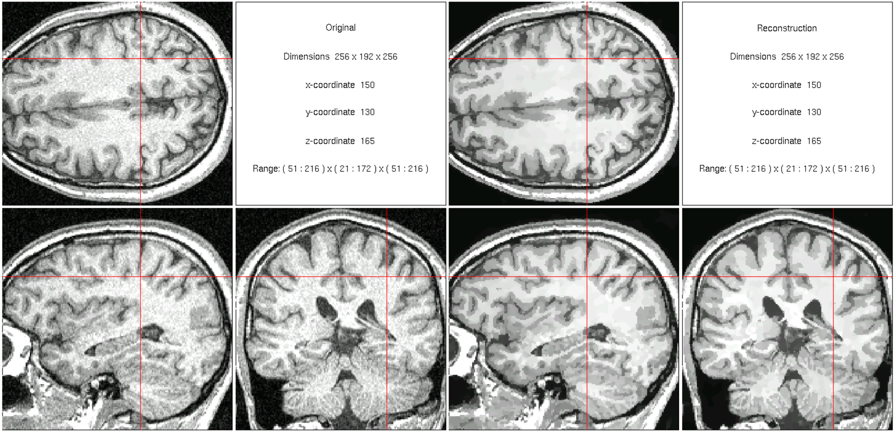
$$\mathbf{s}_{ij} = \lambda^{-1} N_i^{(k-1)}\,\mathcal{K}(\hat{\vartheta}_i^{(k-1)}, \hat{\vartheta}_j^{(k-1)})\,.$$

  λ can be selected using the Propagation condition. Selection depends only on the model, not on the data or structure.

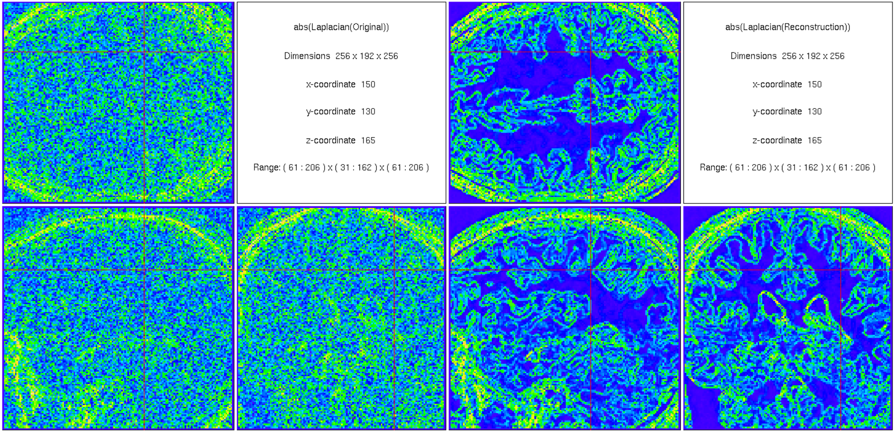
- Maximal bandwidth h_{max} : controls smoothness within homogeneous regions and maximal possible variance reduction. Stopping criterion.

Examples: Images

Structural assumption: Local homogeneity, constant model.

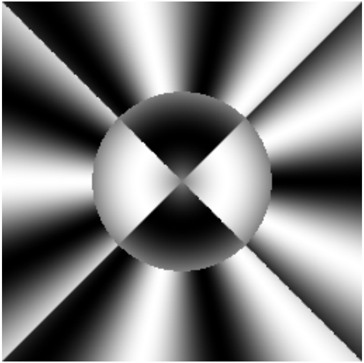


Reconstruction of the 3D MR-Image (Detail)



Absolute values of Laplacian filter (2D) for the same images

Effects of specified model



Start Back

Piecewise smooth original 256×256

$Y_i = f(X_i) + \varepsilon_i, \ -1 \le f(x) \le 1, \ \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

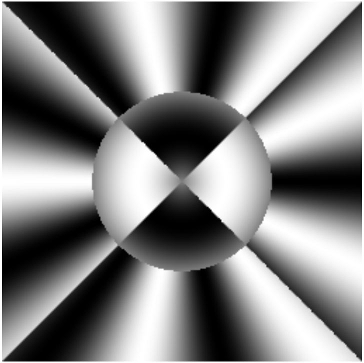
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat\vartheta_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat\vartheta_i^{(k-1)} - \hat\vartheta_j^{(k-1)})^T B_i^{-1} (\hat\vartheta_i^{(k-1)} - \hat\vartheta_j^{(k-1)})$

Effects of specified model



▸ Start ▸ Back

Piecewise smooth original 256 × 256

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

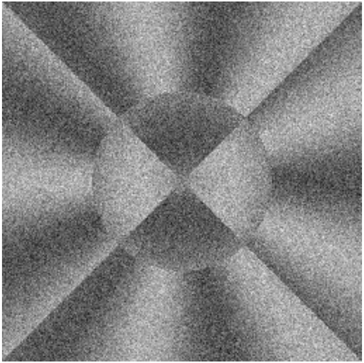
Local quadratic model:

$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with
$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)} x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Noisy image

$$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

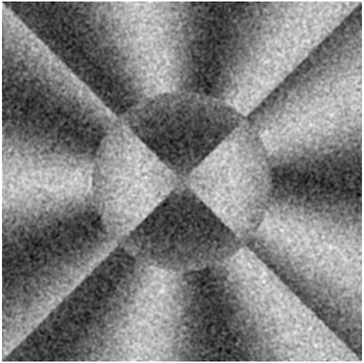
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

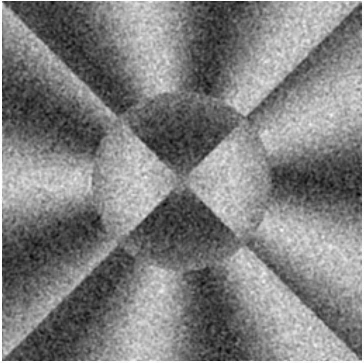
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

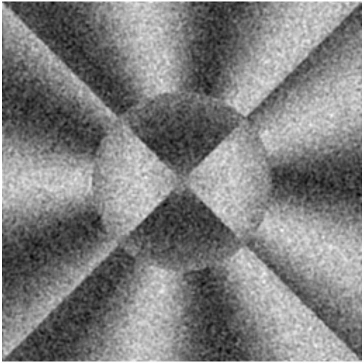
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



▸ Start ▸ Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

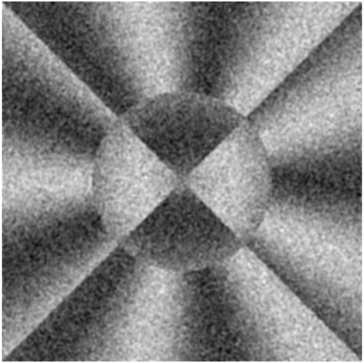
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)} x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i, \ -1 \le f(x) \le 1, \ \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

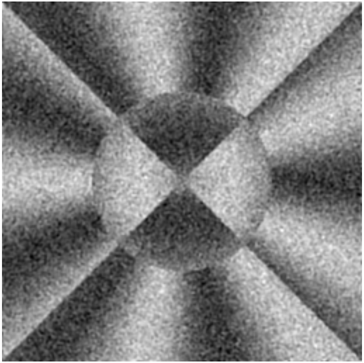
$$f(X_j) = \Psi(X_j - x)^T \vartheta(x) \text{ with}$$

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \ -1 \leq f(x) \leq 1, \ \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

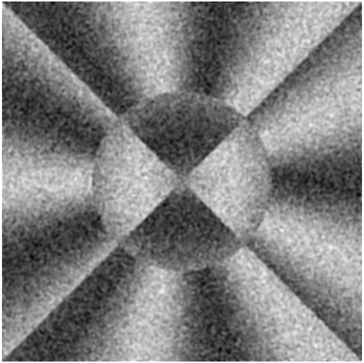
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)} x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \ -1 \leq f(x) \leq 1, \ \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

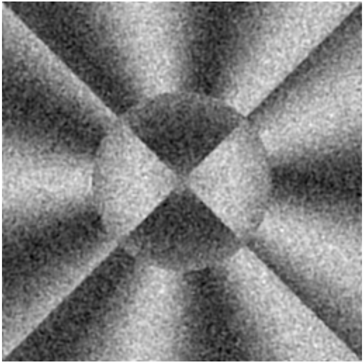
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



▸ Start ▸ Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \ -1 \leq f(x) \leq 1, \ \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

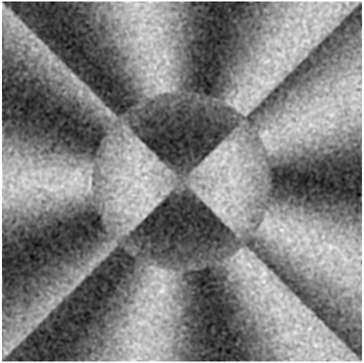
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i, \; -1 \le f(x) \le 1, \; \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

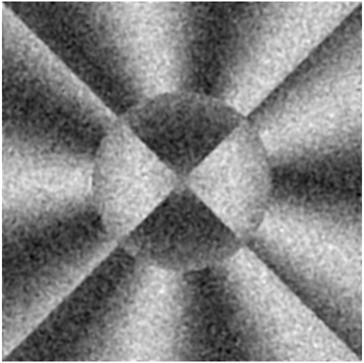
Local quadratic model:

$$f(X_j) = \Psi(X_j - x)^T \vartheta(x) \text{ with}$$
$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



▸ Start ▸ Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

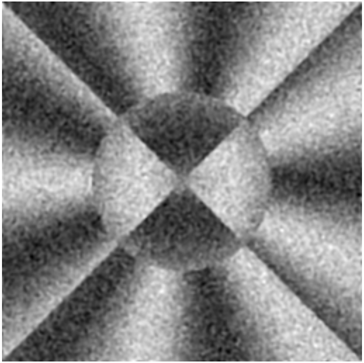
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i, \ -1 \leq f(x) \leq 1, \ \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

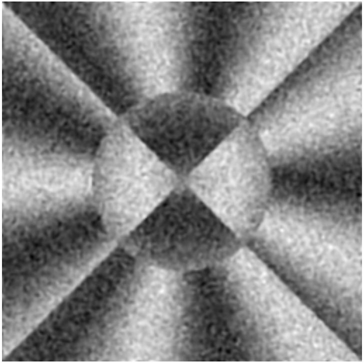
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start   Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i, \; -1 \le f(x) \le 1, \; \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

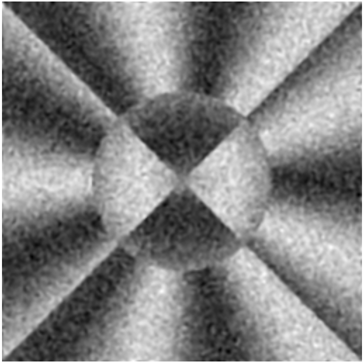
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

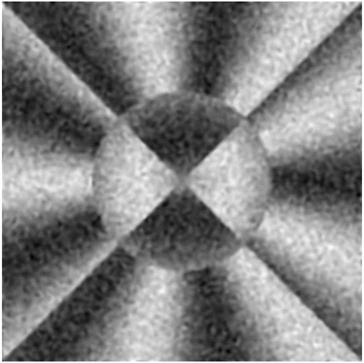
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)} x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



▸ Start ▸ Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \ -1 \le f(x) \le 1, \ \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

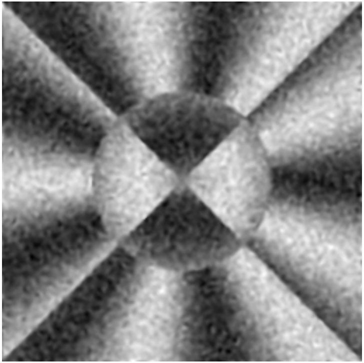
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



▸ Start ▸ Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

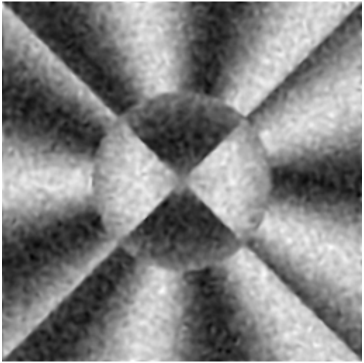
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat\vartheta_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat\vartheta_i^{(k-1)} - \hat\vartheta_j^{(k-1)})^T B_i^{-1} (\hat\vartheta_i^{(k-1)} - \hat\vartheta_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i, \; -1 \le f(x) \le 1, \; \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

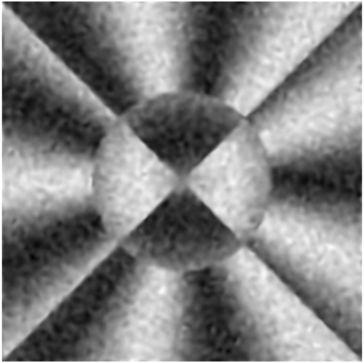
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

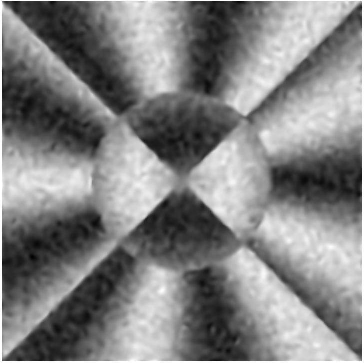
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

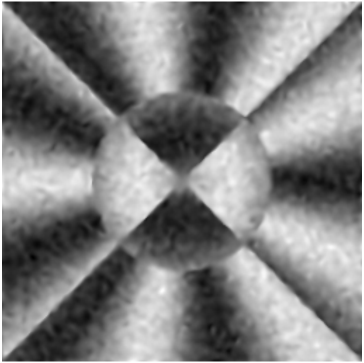
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



▸ Start ▸ Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i, \ -1 \leq f(x) \leq 1, \ \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

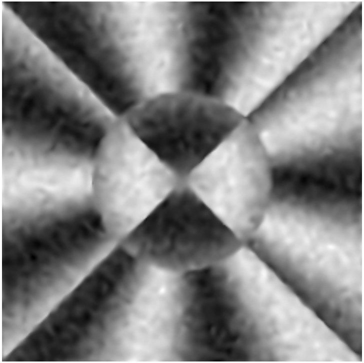
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \ -1 \leq f(x) \leq 1, \ \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

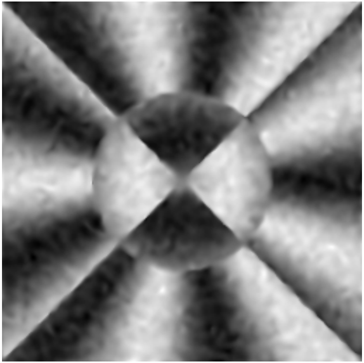
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i, \ -1 \le f(x) \le 1, \ \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

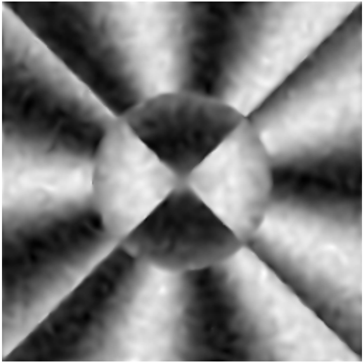
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i,\ -1 \leq f(x) \leq 1,\ \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

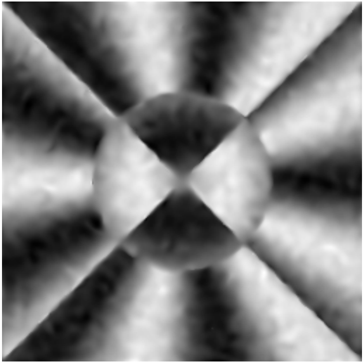
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



▸ Start ▸ Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \le f(x) \le 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

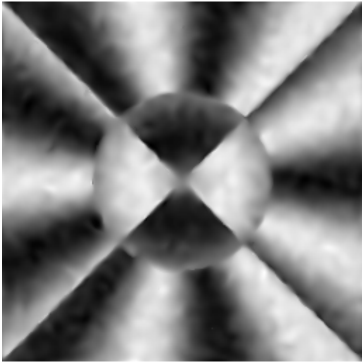
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i,\ -1 \leq f(x) \leq 1,\ \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

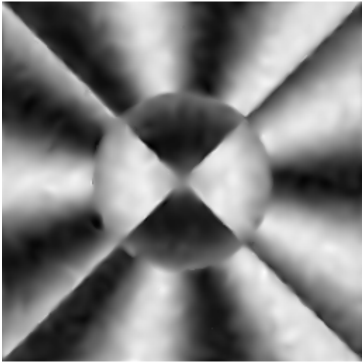
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

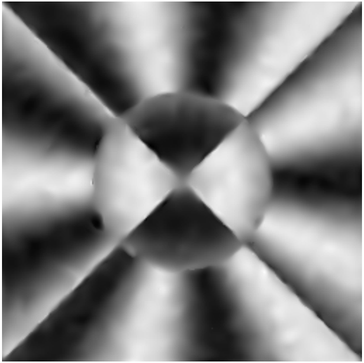
Local quadratic model:

$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with
$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x^2_{(1)}, x_{(1)}x_{(2)}, x^2_{(2)}\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \ -1 \leq f(x) \leq 1, \ \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

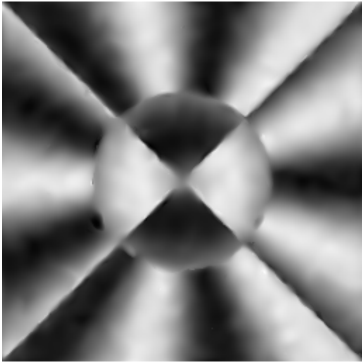
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

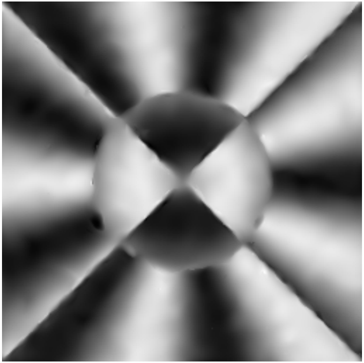
Local quadratic model:

$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with
$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \; -1 \leq f(x) \leq 1, \; \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

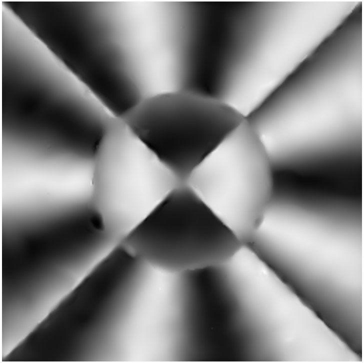
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x^2_{(1)}, x_{(1)} x_{(2)}, x^2_{(2)}\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



▸ Start ▸ Back

Reconstruction

$$Y_i = f(X_i) + \varepsilon_i, \; -1 \le f(x) \le 1, \; \sigma = .25$$

Local constant model:

$$f(x) = \vartheta(x)$$

Local quadratic model:

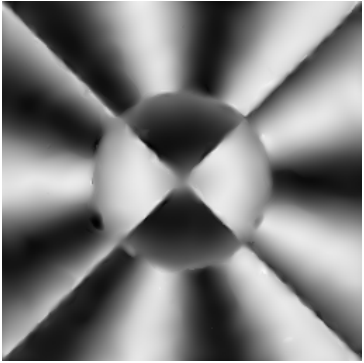
$$f(X_j) = \Psi(X_j - x)^T \vartheta(x) \text{ with}$$

$$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)} x_{(2)}, x_{(2)}^2\right)^T$$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



▸ Start ▸ Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i, \ -1 \le f(x) \le 1, \ \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

Local quadratic model:

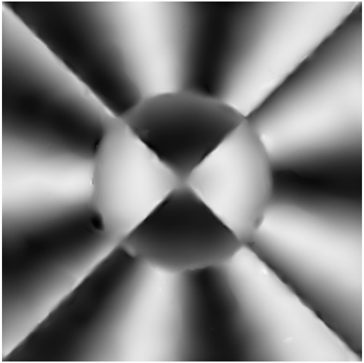
$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$

Effects of specified model



Start Back

Reconstruction

$Y_i = f(X_i) + \varepsilon_i,\ -1 \leq f(x) \leq 1,\ \sigma = .25$

Local constant model:

$f(x) = \vartheta(x)$

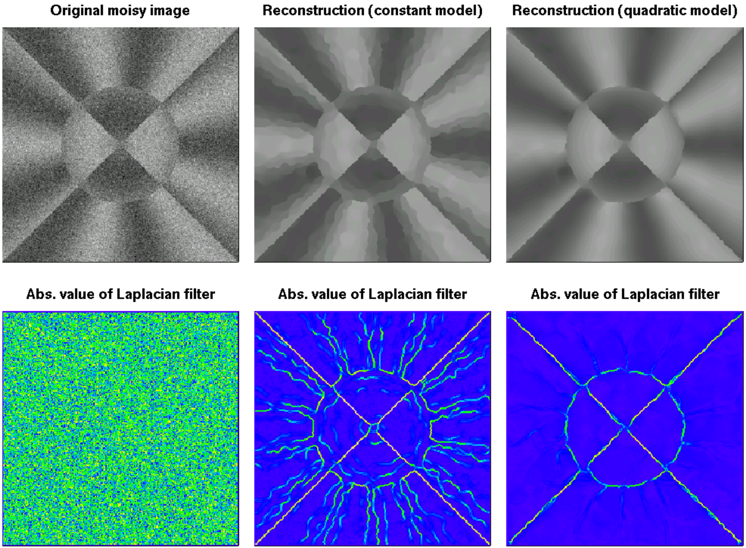
Local quadratic model:

$f(X_j) = \Psi(X_j - x)^T \vartheta(x)$ with

$\Psi(x) = \left(1, x_{(1)}, x_{(2)}, x_{(1)}^2, x_{(1)}x_{(2)}, x_{(2)}^2\right)^T$

Estimates: $\hat{\vartheta}_i = B_i^{-1} \sum_j w_{ij} \Psi(X_j - x)^T Y_j, \quad B_i = \sum_j w_{ij} \Psi(X_j - x)^T \Psi(X_j - x)$

Penalty: $s_{ij}^{(k)} = \frac{1}{2\sigma^2 \lambda} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T B_i^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$



Laplacian filter for noisy image and local constant and quadratic reconstructions.

Color images

Properties

- Vector of 3 (4) values per pixel
- usually in RGB space
- spatial correlation
- correlation of noise between channels

Model

$$Y_i = \vartheta(X_i) + \varepsilon_i, \quad \mathsf{E}\,\varepsilon_i = 0, \quad \mathsf{Var}\,\varepsilon_i = \Sigma_i$$

$$\Sigma_i = \begin{pmatrix} \sigma_{Ri} & 0 & 0 \\ 0 & \sigma_{Gi} & 0 \\ 0 & 0 & \sigma_{Bi} \end{pmatrix} R \begin{pmatrix} \sigma_{Ri} & 0 & 0 \\ 0 & \sigma_{Gi} & 0 \\ 0 & 0 & \sigma_{Bi} \end{pmatrix}$$

$$\sigma^2_{*i} = \eta_{*0} + \eta_{*1}\vartheta_*(X_i)$$

Statistical penalty

$$s_{ij}^{(k)} = \frac{N_i^{(k-1)}}{\lambda C(g, h^{(k-1)})} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T \Sigma^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$$

Color images

Properties

- Vector of 3 (4) values per pixel
- usually in RGB space
- spatial correlation
- correlation of noise between channels

Model

$$
\begin{aligned}
Y_i &= \vartheta(X_i) + \varepsilon_i, \quad \mathsf{E}\,\varepsilon_i = 0, \quad \mathsf{Var}\,\varepsilon_i = \Sigma_i \\
\Sigma_i &= \begin{pmatrix} \sigma_{Ri} & 0 & 0 \\ 0 & \sigma_{Gi} & 0 \\ 0 & 0 & \sigma_{Bi} \end{pmatrix} R \begin{pmatrix} \sigma_{Ri} & 0 & 0 \\ 0 & \sigma_{Gi} & 0 \\ 0 & 0 & \sigma_{Bi} \end{pmatrix} \\
\sigma^2_{*i} &= \eta_{*0} + \eta_{*1}\vartheta_*(X_i)
\end{aligned}
$$

Statistical penalty

$$s_{ij}^{(k)} = \frac{N_i^{(k-1)}}{\lambda C(g, h^{(k-1)})} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})^T \Sigma^{-1} (\hat{\vartheta}_i^{(k-1)} - \hat{\vartheta}_j^{(k-1)})$$

Color images

Properties

- Vector of 3 (4) values per pixel
- usually in RGB space
- spatial correlation
- correlation of noise between channels

Model

$$\begin{aligned} Y_i &= \vartheta(X_i) + \varepsilon_i, \quad \mathsf{E}\,\varepsilon_i = 0, \quad \mathsf{Var}\,\varepsilon_i = \Sigma_i \\ \Sigma_i &= \begin{pmatrix} \sigma_{Ri} & 0 & 0 \\ 0 & \sigma_{Gi} & 0 \\ 0 & 0 & \sigma_{Bi} \end{pmatrix} R \begin{pmatrix} \sigma_{Ri} & 0 & 0 \\ 0 & \sigma_{Gi} & 0 \\ 0 & 0 & \sigma_{Bi} \end{pmatrix} \\ \sigma_{*i}^2 &= \eta_{*0} + \eta_{*1}\vartheta_*(X_i) \end{aligned}$$

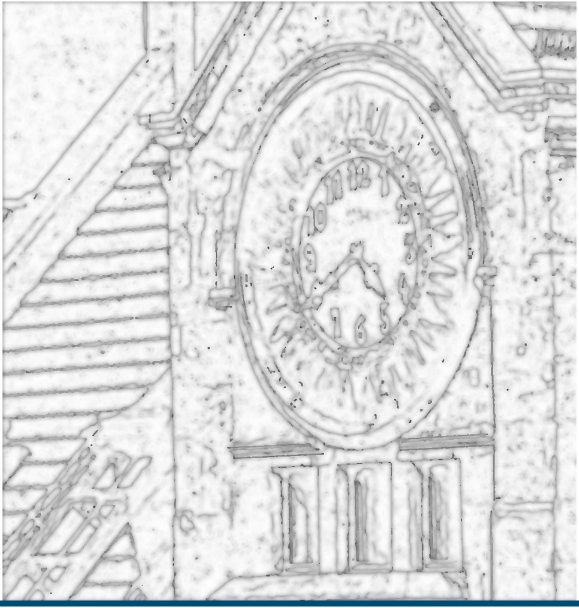
Statistical penalty

$$\boldsymbol{s}_{ij}^{(k)} = \frac{N_i^{(k-1)}}{\lambda C(g, h^{(k-1)})} (\hat\vartheta_i^{(k-1)} - \hat\vartheta_j^{(k-1)})^T \Sigma^{-1} (\hat\vartheta_i^{(k-1)} - \hat\vartheta_j^{(k-1)})$$

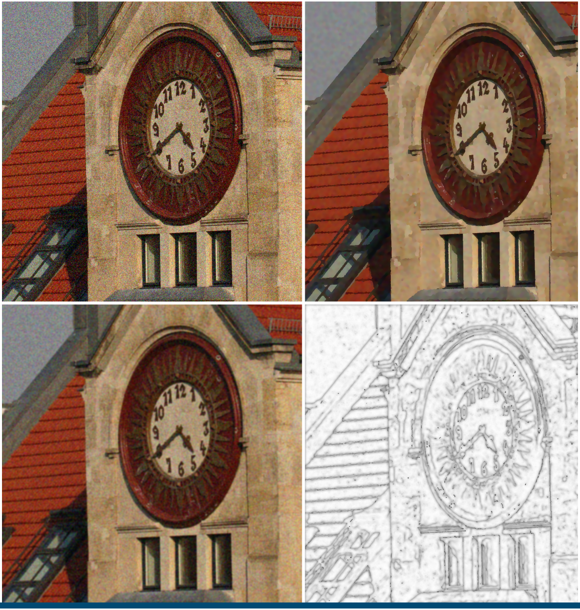
Noisy original (artificial noise)

Adaptive reconstruction $h = 10$

Non-adaptive filter with MSE optimal bandwidth



N_i, number of pixel in $U(X_i)$



Upper left: noisy original (artificial noise)

Upper right: Adaptive reconstruction $h = 10$

Lower left: Non-adaptive filter with MSE optimal bandwidth

Lower right: N_i, number of pixel in $U(X_i)$

Functional MRI

Goal: Insight into the working brain

- knowing "where" a cognitive function is located
- a subject is asked to perform some tasks (designed experiment)
- block design experiments: periods with stimulus alternated with rest periods
- event related experiments: Stimuli (Events) at random or pre-defined times
- investigate contrasts between stimuli
- detect correlations between brain activation and the task the subject performs
- multi-subject comparisons (withingroup variations, classification of diseases, ...)

Functional MRI

Goal: Insight into the working brain

- knowing "where" a cognitive function is located
- a subject is asked to perform some tasks (designed experiment)
- block design experiments: periods with stimulus alternated with rest periods
- event related experiments: Stimuli (Events) at random or pre-defined times
- investigate contrasts between stimuli
- detect correlations between brain activation and the task the subject performs
- multi-subject comparisons (withingroup variations, classification of diseases, ...)

Functional MRI

Goal: Insight into the working brain

- knowing "where" a cognitive function is located
- a subject is asked to perform some tasks (designed experiment)
- block design experiments: periods with stimulus alternated with rest periods
- event related experiments: Stimuli (Events) at random or pre-defined times
- investigate contrasts between stimuli
- detect correlations between brain activation and the task the subject performs
- multi-subject comparisons (withingroup variations, classification of diseases, ...)

Functional MRI

Goal: Insight into the working brain

- knowing "where" a cognitive function is located
- a subject is asked to perform some tasks (designed experiment)
- block design experiments: periods with stimulus alternated with rest periods
- event related experiments: Stimuli (Events) at random or pre-defined times
- investigate contrasts between stimuli
- detect correlations between brain activation and the task the subject performs
- multi-subject comparisons (withingroup variations, classification of diseases, ...)

Functional MRI

Goal: Insight into the working brain

- knowing "where" a cognitive function is located
- a subject is asked to perform some tasks (designed experiment)
- block design experiments: periods with stimulus alternated with rest periods
- event related experiments: Stimuli (Events) at random or pre-defined times
- investigate contrasts between stimuli
- detect correlations between brain activation and the task the subject performs
- multi-subject comparisons (withingroup variations, classification of diseases, ...)

Source: www.wikipedia.org

- High noise level
- Weak signals
- Resolution 64x64x26 voxel
- Voxelsize 3-6 mm
- Time series 100 ... 1000 scans
- Time resolution of 2 s
- Correlation in space and time

- Extremely strong magnetic field ($\geq 1.5T$)
- Measure relaxation of proton spins caused by a small overlaid magnetic field
- 3D Images obtained by Fourier transform

How does it work

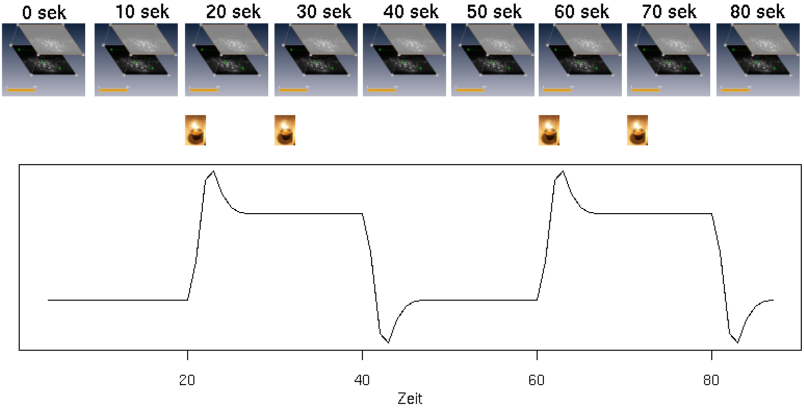
- Active neurons need oxygen
- BOLD-effect: **B**lood **O**xygen **L**evel **D**ependent
- Time series of 3D images every 2 seconds
- Observe change in signal in activated voxels

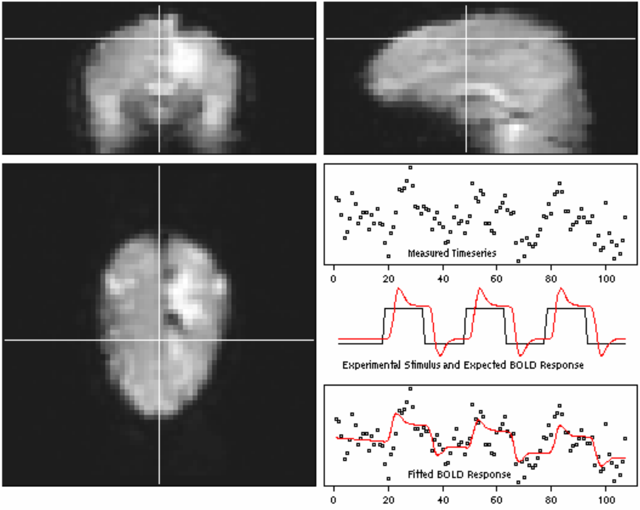
How does it work

- Active neurons need oxygen
- BOLD-effect: **B**lood **O**xygen **L**evel **D**ependent
- Time series of 3D images every 2 seconds
- Observe change in signal in activated voxels

How does it work

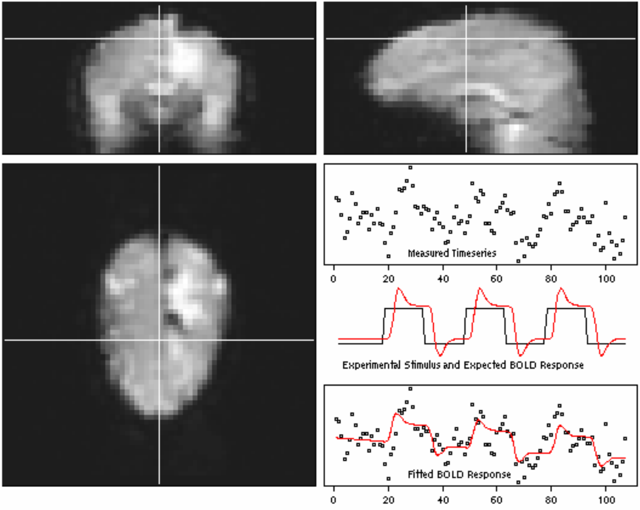
- Active neurons need oxygen
- BOLD-effect: **B**lood **O**xygen **L**evel **D**ependent
- Time series of 3D images every 2 seconds
- Observe change in signal in activated voxels





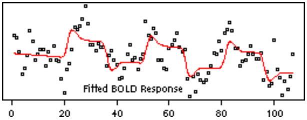
Data: H. Voss, Weill Medical College of Cornell University

- High noise level!
- Weak signals
- Resolution 64x64x26 voxel (Voxelsize 4mm)
- Time series 100 ... 1000 scans every 2s
- Correlation in time (modeled as AR(1))
- Correlation in space (modeled as convolution with a Gaussian kernel)



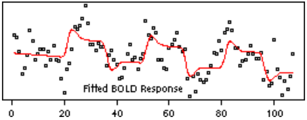
Data: H. Voss, Weill Medical College of Cornell University

- High noise level!
- Weak signals
- Resolution 64x64x26 voxel (Voxelsize 4mm)
- Time series 100 ... 1000 scans every 2s
- Correlation in time (modeled as AR(1))
- Correlation in space (modeled as convolution with a Gaussian kernel)



Data Preparation

- Registration
- Motion correction
- Normalization

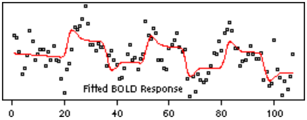


Data Preparation

- Registration
- Motion correction
- Normalization

Linear Model

- $Y = X\beta + \epsilon$
- X contains hemodynamic response and trends
- Evaluate and smooth AR(1)
- Prewhitening
- $\tilde{Y} = \tilde{X}\beta + \tilde{\epsilon}$

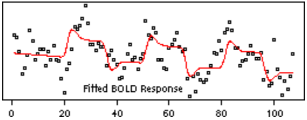


Data Preparation

- Registration
- Motion correction
- Normalization

Linear Model

- $Y = X\beta + \epsilon$
- X contains hemodynamic response and trends
- Evaluate and smooth AR(1)
- Prewhitening
- $\tilde{Y} = \tilde{X}\beta + \tilde{\epsilon}$



Data Preparation

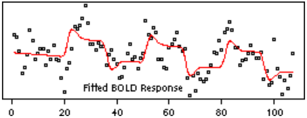
- Registration
- Motion correction
- Normalization

Linear Model

- $Y = X\beta + \epsilon$
- X contains hemodynamic response and trends
- Evaluate and smooth AR(1)
- Prewhitening
- $\tilde{Y} = \tilde{X}\beta + \tilde{\epsilon}$

Thresholding

- define t-statistic for parameters or contrast of parameters
- severe multiple test problem

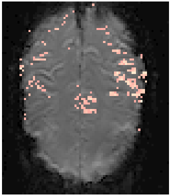


Data Preparation

- Registration
- Motion correction
- Normalization

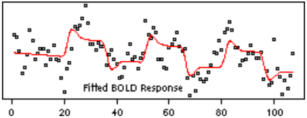
Linear Model

- $Y = X\beta + \epsilon$
- X contains hemodynamic response and trends
- Evaluate and smooth AR(1)
- Prewhitening
- $\tilde{Y} = \tilde{X}\beta + \tilde{\epsilon}$



Thresholding

- define t-statistic for parameters or contrast of parameters
- threshold using FDR

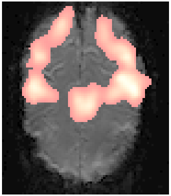


Data Preparation

- Registration
- Motion correction
- Normalization
- Smoothing fMRI images (Gaussian filter)

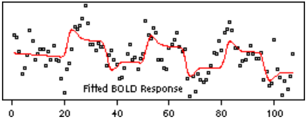
Linear Model

- $Y = X\beta + \epsilon$
- X contains hemodynamic response and trends
- Evaluate and smooth AR(1)
- Prewhitening
- $\tilde{Y} = \tilde{X}\beta + \tilde{\epsilon}$



Thresholding

- define t-statistic for parameters or contrast of parameters
- threshold using RFT (EC)



Data Preparation

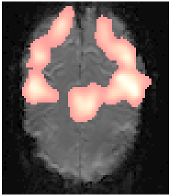
- Registration
- Motion correction
- Normalization

Linear Model

- $Y = X\beta + \epsilon$
- X contains hemodynamic response and trends
- Evaluate and smooth AR(1)
- Prewhitening
- $\tilde{Y} = \tilde{X}\beta + \tilde{\epsilon}$

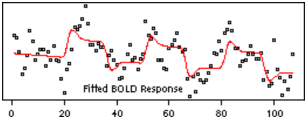
Smoothing β

non-adaptive



Thresholding

- define t-statistic for parameters or contrast of parameters
- threshold using RFT (EC)



Data Preparation

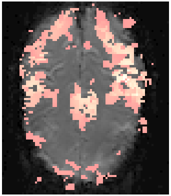
- Registration
- Motion correction
- Normalization

Linear Model

- $Y = X\beta + \epsilon$
- X contains hemodynamic response and trends
- Evaluate and smooth AR(1)
- Prewhitening
- $\tilde{Y} = \tilde{X}\beta + \tilde{\epsilon}$

Smoothing β

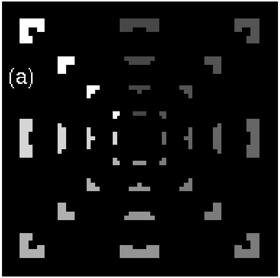
- adaptive!



Thresholding

- define t-statistic for parameters or contrast of parameters
- threshold using RFT (EC)

Simulation experiment

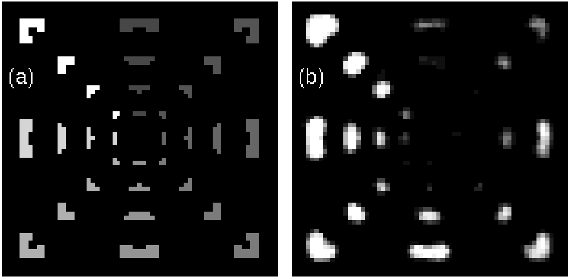


(a) Signal location in Phantom data (3D) with different signal size
26 slices with 15 containing activations

(b) Relative proportion of detection using non adaptive smoothing

(c) Relative proportion of detection using Structural Adaptive Smoothing

Simulation experiment

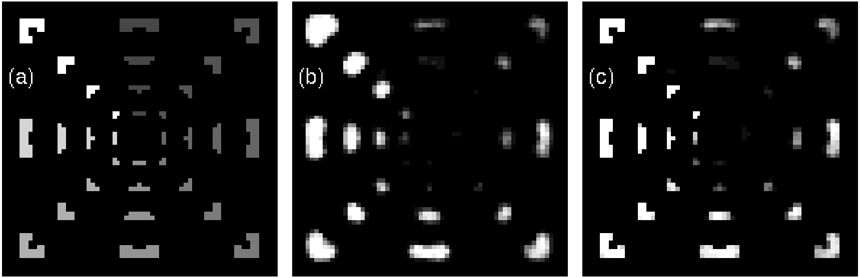


(a) Signal location in Phantom data (3D) with different signal size
26 slices with 15 containing activations

(b) Relative proportion of detection using non adaptive smoothing

(c) Relative proportion of detection using Structural Adaptive Smoothing

Simulation experiment



(a) Signal location in Phantom data (3D) with different signal size
26 slices with 15 containing activations

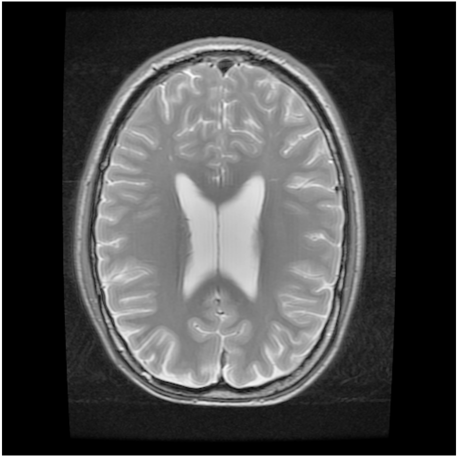
(b) Relative proportion of detection using non adaptive smoothing

(c) Relative proportion of detection using Structural Adaptive Smoothing

Diffusion Weighted Imaging

Accessing the brain structure I

- Human brain has a rich structure at different scales: gray matter, white matter, CSF



Data: H.U. Voss, Weill Medical College, Cornell University New York

Accessing the brain structure II

- Fiber structure of white matter
- Highly anisotropic water diffusion
  (hindered (within neurons) or restricted (within fiber bundles) diffusion)
- Structures of interest: 1μm ... 10mm
- Diseases/injury may lead to changes in diffusion abilities
- Brain development / Myelination
- Neuronal connectivity

Accessing the brain structure II

- Fiber structure of white matter
- Highly anisotropic water diffusion
  (hindered (within neurons) or restricted (within fiber bundles) diffusion)
- Structures of interest: 1μm ... 10mm
- Diseases/injury may lead to changes in diffusion abilities
- Brain development / Myelination
- Neuronal connectivity

Accessing the brain structure II

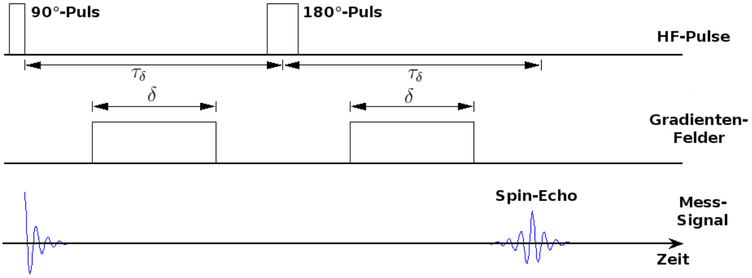
- Fiber structure of white matter
- Highly anisotropic water diffusion
  (hindered (within neurons) or restricted (within fiber bundles) diffusion)
- Structures of interest: 1μm ... 10mm
- Diseases/injury may lead to changes in diffusion abilities
- Brain development / Myelination
- Neuronal connectivity

Accessing the brain structure II

- Fiber structure of white matter
- Highly anisotropic water diffusion
  (hindered (within neurons) or restricted (within fiber bundles) diffusion)
- Structures of interest: 1μm ... 10mm
- Diseases/injury may lead to changes in diffusion abilities
- Brain development / Myelination
- Neuronal connectivity

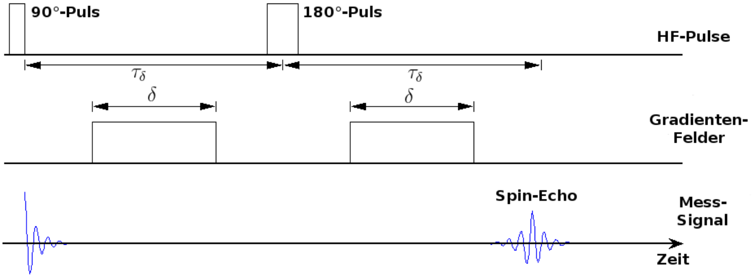
Accessing the brain structure II

- Fiber structure of white matter
- Highly anisotropic water diffusion
  (hindered (within neurons) or restricted (within fiber bundles) diffusion)
- Structures of interest: 1μm ... 10mm
- Diseases/injury may lead to changes in diffusion abilities
- Brain development / Myelination
- Neuronal connectivity



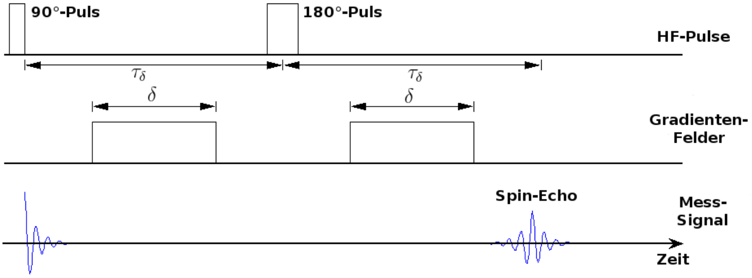
Source: www.wikipedia.org

- Based on MR: homogeneous magnetic field B_0
- RF-pulse at Larmor-frequency: Phase coherence
- Magnetic field gradient in direction $\vec{b}$: de-phasing $\rightarrow$ signal loss
- reverse spin rotation by 180° RF-pulse
- Magnetic field gradient in direction $\vec{b}$: re-phasing $\rightarrow$ signal recovery ...
- ... attenuated because of possible diffusion of spins along $\vec{b}$



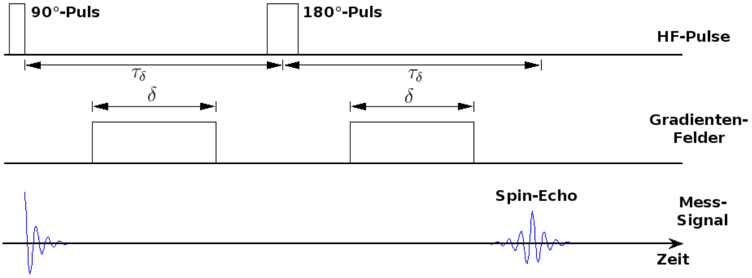
Source: www.wikipedia.org

- Based on MR: homogeneous magnetic field B_0
- RF-pulse at Larmor-frequency: Phase coherence
- Magnetic field gradient in direction $\vec{b}$: de-phasing $\longrightarrow$ signal loss
- reverse spin rotation by 180° RF-pulse
- Magnetic field gradient in direction $\vec{b}$: re-phasing $\longrightarrow$ signal recovery ...
- ... attenuated because of possible diffusion of spins along $\vec{b}$



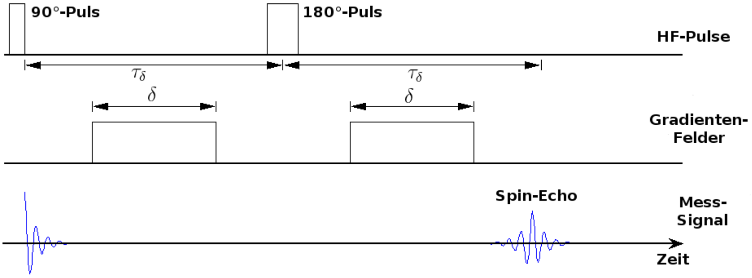
Source: www.wikipedia.org

- ▷ Based on MR: homogeneous magnetic field B_0
- ▷ RF-pulse at Larmor-frequency: Phase coherence
- ▷ Magnetic field gradient in direction $\vec{b}$: de-phasing $\rightarrow$ signal loss
- ▷ reverse spin rotation by 180° RF-pulse
- ▷ Magnetic field gradient in direction $\vec{b}$: re-phasing $\rightarrow$ signal recovery ...
- ▷ ... attenuated because of possible diffusion of spins along $\vec{b}$



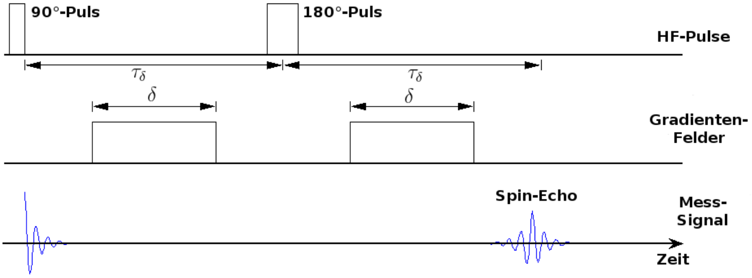
Source: www.wikipedia.org

- Based on MR: homogeneous magnetic field B_0
- RF-pulse at Larmor-frequency: Phase coherence
- Magnetic field gradient in direction $\vec{b}$: de-phasing $\rightarrow$ signal loss
- reverse spin rotation by 180° RF-pulse
- Magnetic field gradient in direction $\vec{b}$: re-phasing $\rightarrow$ signal recovery ...
- ... attenuated because of possible diffusion of spins along $\vec{b}$



Source: www.wikipedia.org

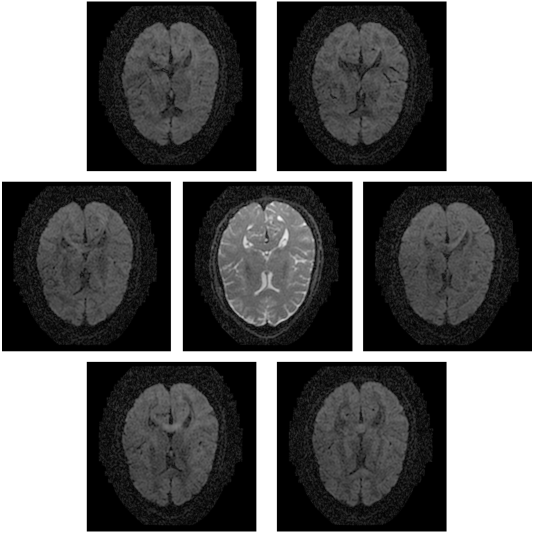
- Based on MR: homogeneous magnetic field B_0
- RF-pulse at Larmor-frequency: Phase coherence
- Magnetic field gradient in direction $\vec{b}$: de-phasing $\rightarrow$ signal loss
- reverse spin rotation by 180° RF-pulse
- Magnetic field gradient in direction $\vec{b}$: re-phasing $\rightarrow$ signal recovery ...
- ... attenuated because of possible diffusion of spins along $\vec{b}$



Source: www.wikipedia.org

- Based on MR: homogeneous magnetic field B_0
- RF-pulse at Larmor-frequency: Phase coherence
- Magnetic field gradient in direction $\vec{b}$: de-phasing $\rightarrow$ signal loss
- reverse spin rotation by 180° RF-pulse
- Magnetic field gradient in direction $\vec{b}$: re-phasing $\rightarrow$ signal recovery ...
- ... attenuated because of possible diffusion of spins along $\vec{b}$

- Measuring at different gradient vectors b



- 3D - Diffusion weighted image (DWI) and compare with image at zero gradient

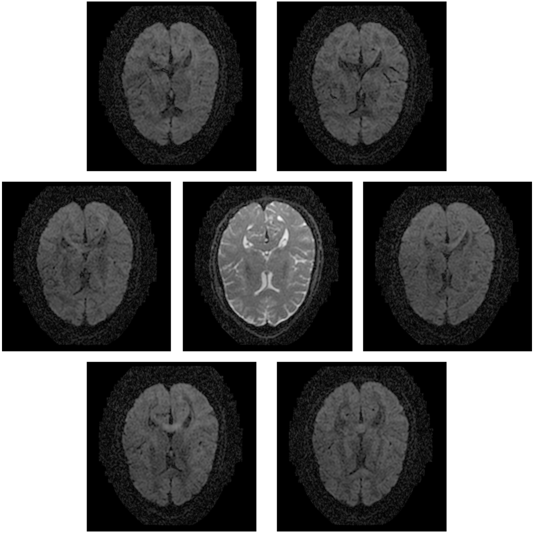
$$S_b = S_0 \exp(-D * b)$$

Data: H. Voss, Weill Medical College of Cornell University

- Measuring at different gradient vectors b

- 3D - Diffusion weighted image (DWI) and compare with image at zero gradient

$$S_b = S_0 \exp(-D * b)$$



Data: H. Voss, Weill Medical College of Cornell University

Data transformation

- Apparent Diffusion Coefficient (ADC)

$$D = -\ln \frac{S_b}{S_0}$$

- Measure $n_{grad} = 25 \ldots 100$ gradient directions b
- Data dimensionality: $n_x \times n_y \times n_z \times n_{grad}$

Data properties

- Complicated noise model for D: non-linear transformation
- Bias!
- Spatial correlation!

Data transformation

- Apparent Diffusion Coefficient (ADC)

$$D = -\ln \frac{S_b}{S_0}$$

- Measure $n_{grad} = 25 \dots 100$ gradient directions b
- Data dimensionality: $n_x \times n_y \times n_z \times n_{grad}$

Data properties

- Complicated noise model for D: non-linear transformation
- Bias!
- Spatial correlation!

Diffusion Tensor Imaging (DTI)

- Diffusion Tensor $\mathcal{D}$ as lowest order model

$$S_b = S_0 \exp(-b * \mathcal{D} * b^T)$$

- $\mathcal{D}$ is symmetric semi-definite 3×3 matrix
- for $A = (a_{ij})_{i,j=1,\dots,3}$ define $\overrightarrow{A} = (a_{11}, a_{22}, a_{33}, 2a_{12}, 2a_{13}, 2a_{23})^T$ and $\overleftarrow{A} = (a_{11}, a_{22}, a_{33}, a_{12}, a_{13}, a_{23})^T$
- Linear Model

$$D_i = X \overleftarrow{\mathcal{D}}_i + \varepsilon_i$$

with X containing $\mathbf{b} = \overrightarrow{(bb^T)}$ as columns.

Diffusion Tensor Imaging (DTI)

- Diffusion Tensor $\mathcal{D}$ as lowest order model

$$S_b = S_0 \exp(-b * \mathcal{D} * b^T)$$

- $\mathcal{D}$ is symmetric semi-definite 3×3 matrix
- for $A = (a_{ij})_{i,j=1,\ldots,3}$ define $\overrightarrow{A} = (a_{11}, a_{22}, a_{33}, 2a_{12}, 2a_{13}, 2a_{23})^T$ and $\overleftarrow{A} = (a_{11}, a_{22}, a_{33}, a_{12}, a_{13}, a_{23})^T$
- Linear Model

$$D_i = X \overleftarrow{\mathcal{D}}_i + \varepsilon_i$$

with X containing $\mathbf{b} = \overrightarrow{(bb^T)}$ as columns.

- The Diffusion Tensor only models some data properties
- Diffusion Weighted Imaging is capable to probe microscopic structures well beyond typical image resolutions through molecule displacement!
- At voxel resolution the overall displacement distribution of all molecules in a voxel is observed.
- Structure not described by Diffusion Tensor (Crossing fibres etc.)
- Use all ADC's for as many gradients as possible ... HARDI, Q-ball

- The Diffusion Tensor only models some data properties
- Diffusion Weighted Imaging is capable to probe microscopic structures well beyond typical image resolutions through molecule displacement!
- At voxel resolution the overall displacement distribution of all molecules in a voxel is observed.
- Structure not described by Diffusion Tensor (Crossing fibres etc.)
- Use all ADC's for as many gradients as possible ... HARDI, Q-ball

- The Diffusion Tensor only models some data properties
- Diffusion Weighted Imaging is capable to probe microscopic structures well beyond typical image resolutions through molecule displacement!
- At voxel resolution the overall displacement distribution of all molecules in a voxel is observed.
- Structure not described by Diffusion Tensor (Crossing fibres etc.)
- Use all ADC's for as many gradients as possible ... HARDI, Q-ball

- The Diffusion Tensor only models some data properties
- Diffusion Weighted Imaging is capable to probe microscopic structures well beyond typical image resolutions through molecule displacement!
- At voxel resolution the overall displacement distribution of all molecules in a voxel is observed.
- Structure not described by Diffusion Tensor (Crossing fibres etc.)
- Use all ADC's for as many gradients as possible ... HARDI, Q-ball

- The Diffusion Tensor only models some data properties
- Diffusion Weighted Imaging is capable to probe microscopic structures well beyond typical image resolutions through molecule displacement!
- At voxel resolution the overall displacement distribution of all molecules in a voxel is observed.
- Structure not described by Diffusion Tensor (Crossing fibres etc.)
- Use all ADC's for as many gradients as possible ... HARDI, Q-ball

DTI - eigenvalues and anisotropy

- Only invariant information useful for medical analysis
- Consider eigenvalue decomposition of the diffusion tensor $\mathcal{D} = V \Lambda V^T$
- Main Diffusion direction by first eigenvector of $\mathcal{D}$
- Anisotropy (diffusion ellipsoid)
- Consider Anisotropy Indices:

$$FA = \frac{\sqrt{3}\sqrt{(\lambda_1 - \langle\lambda\rangle)^2 + (\lambda_2 - \langle\lambda\rangle)^2 + (\lambda_3 - \langle\lambda\rangle)^2}}{\sqrt{2}\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

- Barycentric coordinates

$$c_l = \frac{\lambda_1 - \lambda_2}{3\langle\lambda\rangle} \quad c_p = \frac{2(\lambda_2 - \lambda_3)}{3\langle\lambda\rangle} \quad c_s = \frac{\lambda_3}{\langle\lambda\rangle}$$

DTI - eigenvalues and anisotropy

- Only invariant information useful for medical analysis
- Consider eigenvalue decomposition of the diffusion tensor $\mathcal{D} = V \Lambda V^T$
- Main Diffusion direction by first eigenvector of $\mathcal{D}$
- Anisotropy (diffusion ellipsoid)
- Consider Anisotropy Indices:

$$FA = \frac{\sqrt{3}\sqrt{(\lambda_1 - \langle\lambda\rangle)^2 + (\lambda_2 - \langle\lambda\rangle)^2 + (\lambda_3 - \langle\lambda\rangle)^2}}{\sqrt{2}\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

- Barycentric coordinates

$$c_l = \frac{\lambda_1 - \lambda_2}{3\langle\lambda\rangle} \quad c_p = \frac{2(\lambda_2 - \lambda_3)}{3\langle\lambda\rangle} \quad c_s = \frac{\lambda_3}{\langle\lambda\rangle}$$

DTI - eigenvalues and anisotropy

- Only invariant information useful for medical analysis
- Consider eigenvalue decomposition of the diffusion tensor $\mathcal{D} = V\Lambda V^T$
- Main Diffusion direction by first eigenvector of $\mathcal{D}$
- Anisotropy (diffusion ellipsoid)
- Consider Anisotropy Indices:

$$FA = \frac{\sqrt{3}\sqrt{(\lambda_1 - \langle\lambda\rangle)^2 + (\lambda_2 - \langle\lambda\rangle)^2 + (\lambda_3 - \langle\lambda\rangle)^2}}{\sqrt{2}\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

- Barycentric coordinates

$$c_l = \frac{\lambda_1 - \lambda_2}{3\langle\lambda\rangle} \quad c_p = \frac{2(\lambda_2 - \lambda_3)}{3\langle\lambda\rangle} \quad c_s = \frac{\lambda_3}{\langle\lambda\rangle}$$

DTI - eigenvalues and anisotropy

- Only invariant information useful for medical analysis
- Consider eigenvalue decomposition of the diffusion tensor $\mathcal{D} = V \Lambda V^T$
- Main Diffusion direction by first eigenvector of $\mathcal{D}$
- Anisotropy (diffusion ellipsoid)
- Consider Anisotropy Indices:

$$FA = \frac{\sqrt{3}\sqrt{(\lambda_1 - \langle\lambda\rangle)^2 + (\lambda_2 - \langle\lambda\rangle)^2 + (\lambda_3 - \langle\lambda\rangle)^2}}{\sqrt{2}\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

- Barycentric coordinates

$$c_l = \frac{\lambda_1 - \lambda_2}{3\langle\lambda\rangle} \quad c_p = \frac{2(\lambda_2 - \lambda_3)}{3\langle\lambda\rangle} \quad c_s = \frac{\lambda_3}{\langle\lambda\rangle}$$

DTI - eigenvalues and anisotropy

- Only invariant information useful for medical analysis
- Consider eigenvalue decomposition of the diffusion tensor $\mathcal{D} = V \Lambda V^T$
- Main Diffusion direction by first eigenvector of $\mathcal{D}$
- Anisotropy (diffusion ellipsoid)
- Consider Anisotropy Indices:

$$FA = \frac{\sqrt{3}\sqrt{(\lambda_1 - \langle\lambda\rangle)^2 + (\lambda_2 - \langle\lambda\rangle)^2 + (\lambda_3 - \langle\lambda\rangle)^2}}{\sqrt{2}\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

- Barycentric coordinates

$$c_l = \frac{\lambda_1 - \lambda_2}{3\langle\lambda\rangle} \quad c_p = \frac{2(\lambda_2 - \lambda_3)}{3\langle\lambda\rangle} \quad c_s = \frac{\lambda_3}{\langle\lambda\rangle}$$

DTI - eigenvalues and anisotropy

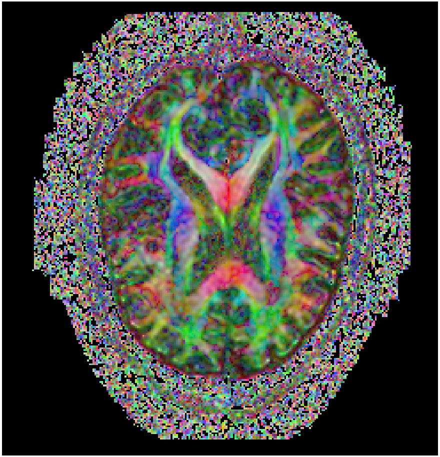
- Only invariant information useful for medical analysis
- Consider eigenvalue decomposition of the diffusion tensor $\mathcal{D} = V \Lambda V^T$
- Main Diffusion direction by first eigenvector of $\mathcal{D}$
- Anisotropy (diffusion ellipsoid)
- Consider Anisotropy Indices:

$$FA = \frac{\sqrt{3}\sqrt{(\lambda_1 - \langle\lambda\rangle)^2 + (\lambda_2 - \langle\lambda\rangle)^2 + (\lambda_3 - \langle\lambda\rangle)^2}}{\sqrt{2}\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}$$

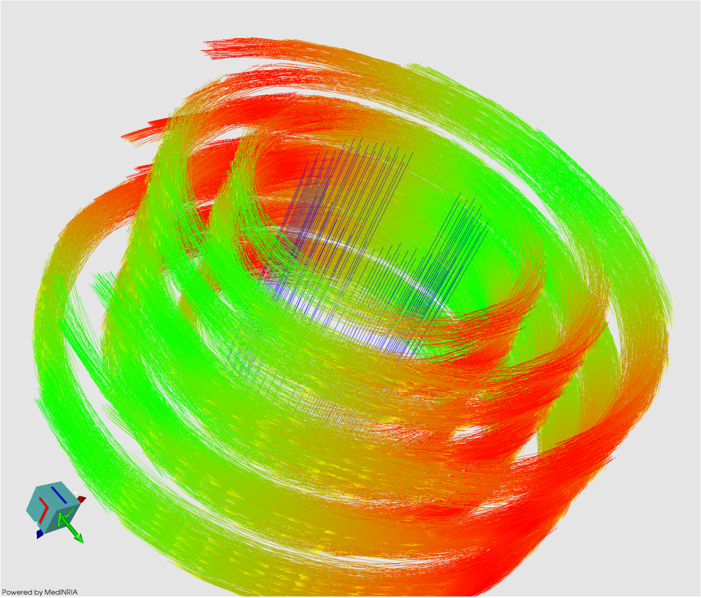
- Barycentric coordinates

$$c_l = \frac{\lambda_1 - \lambda_2}{3\langle\lambda\rangle} \quad c_p = \frac{2(\lambda_2 - \lambda_3)}{3\langle\lambda\rangle} \quad c_s = \frac{\lambda_3}{\langle\lambda\rangle}$$

DTI - Results



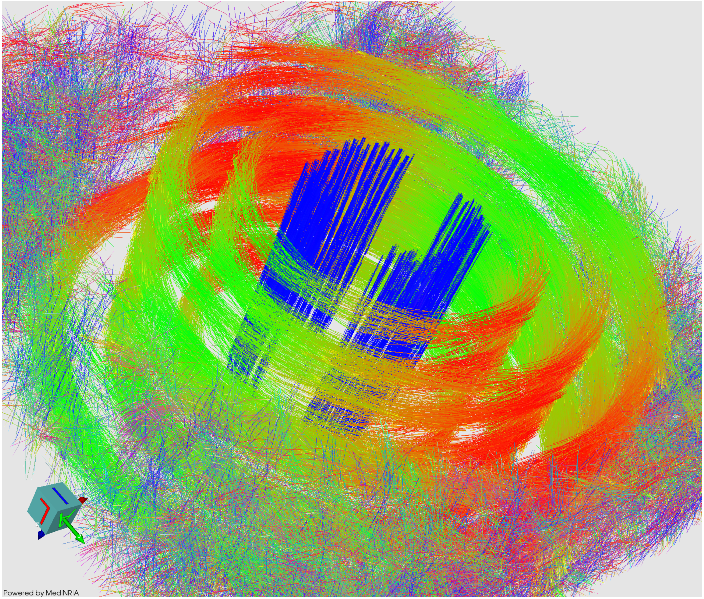
- Main diffusion direction of water in tissue (color)
- Degree of anisotropy of diffusion (brightness)



DTI - Fiber tracking

Artificial example: Fiber tracking based on true tensors.

Tracking performed using MedINRIA

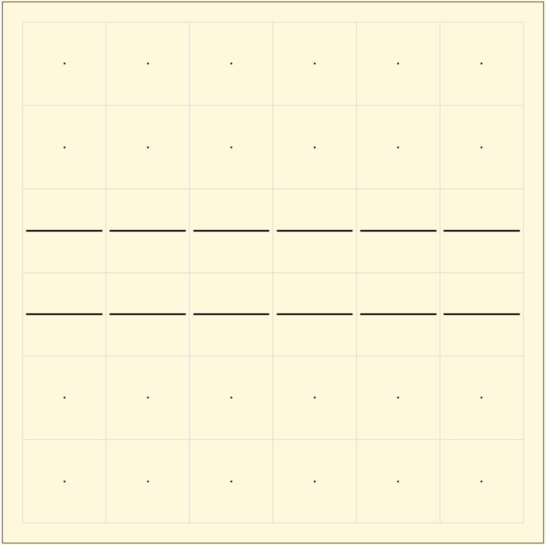


DTI - Fiber tracking

Artificial example: Fiber tracking based on estimated tensors

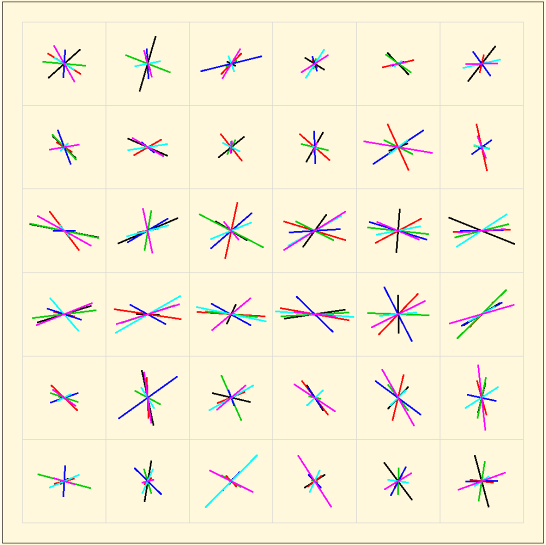
Tracking performed using MedINRIA

Effect of noise in DTI



- Vector field: diffusion direction
- Fiber tracking
- Connectivity, diagnosis

Effect of noise in DTI



- Vector field disturbed by noise
- Fiber tracking still possible?
- Medical information?

Structural adaptive smoothing the DWI

- ▷ Structural assumption: local constant diffusion tensor
- ▷ Idea 1: Use Diffusion tensor to assess spatial information!

$$\begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix}$$

$$s_{ij}^{(k)} = \frac{1}{\lambda(h^{(k-1)}, g)} \Delta(\hat{D}_i^{(k-1)}, \hat{D}_j^{(k-1)})/(\hat{\sigma}_i^{(k-1)})^2$$

- ▷ Idea 2: Use anisotropy information contained in Diffusion Tensor $\hat{D}_i$ using an ellipsoidal localization kernel $K_l(., \hat{D}_i)$

Structural adaptive smoothing the DWI

- Structural assumption: local constant diffusion tensor
- Idea 1: Use Diffusion tensor to assess spatial information!

$$\begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix}$$

$$s_{ij}^{(k)} = \frac{1}{\lambda(h^{(k-1)}, g)} \Delta(\hat{D}_i^{(k-1)}, \hat{D}_j^{(k-1)})/(\hat{\sigma}_i^{(k-1)})^2$$

- Idea 2: Use anisotropy information contained in Diffusion Tensor $\hat{D}_i$ using an ellipsoidal localization kernel $K_l(., \hat{D}_i)$

Structural adaptive smoothing the DWI

- ▷ Structural assumption: local constant diffusion tensor
- ▷ Idea 1: Use Diffusion tensor to assess spatial information!

$$\begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix}$$

$$s_{ij}^{(k)} = \frac{1}{\lambda(h^{(k-1)}, g)} \Delta(\hat{D}_i^{(k-1)}, \hat{D}_j^{(k-1)})/(\hat{\sigma}_i^{(k-1)})^2$$

- ▷ Idea 2: Use anisotropy information contained in Diffusion Tensor $\hat{D}_i$ using an ellipsoidal localization kernel $K_l(., \hat{D}_i)$

Structural adaptive smoothing of DWI's

- Define local weighting schemes

$$w_{ij}^{(k)} = K_l(l_{ij}^{(k)}, \hat{D}_i^{(k-1)})K_s(s_{ij}^{(k)})$$

- Smooth DWI's
- Re-estimate diffusion tensor and it's variance
- Iterate

Structural adaptive smoothing of DWI's

- Define local weighting schemes

$$w_{ij}^{(k)} = K_l(l_{ij}^{(k)}, \hat{D}_i^{(k-1)})K_s(s_{ij}^{(k)})$$

- Smooth DWI's
- Re-estimate diffusion tensor and it's variance
- Iterate

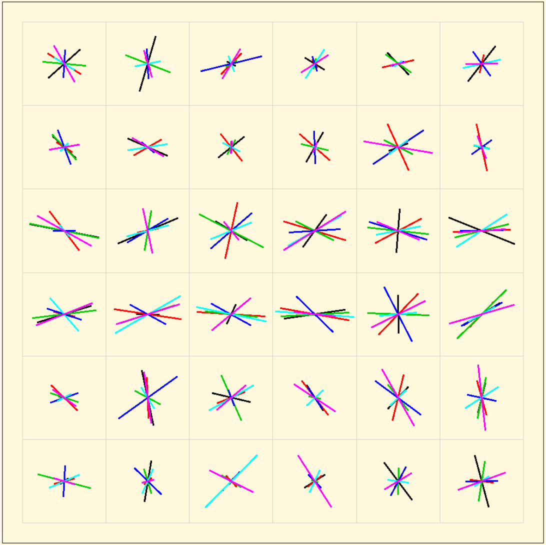
Structural adaptive smoothing of DWI's

- Define local weighting schemes

$$w_{ij}^{(k)} = K_l(l_{ij}^{(k)}, \hat{D}_i^{(k-1)})K_s(s_{ij}^{(k)})$$

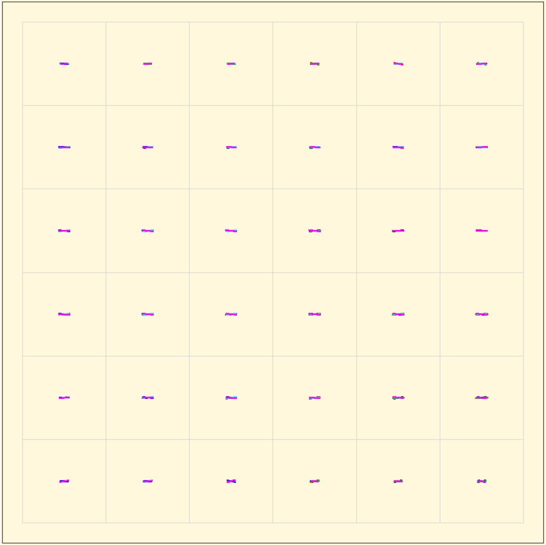
- Smooth DWI's
- Re-estimate diffusion tensor and it's variance
- Iterate

Simulation experiment



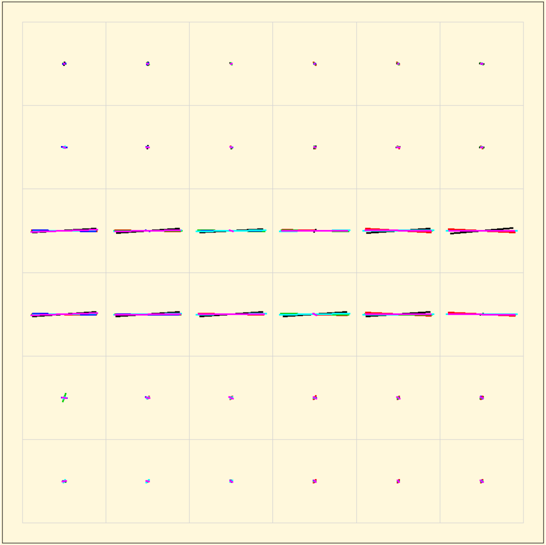
- Diffusion direction measured with noise

Simulation experiment

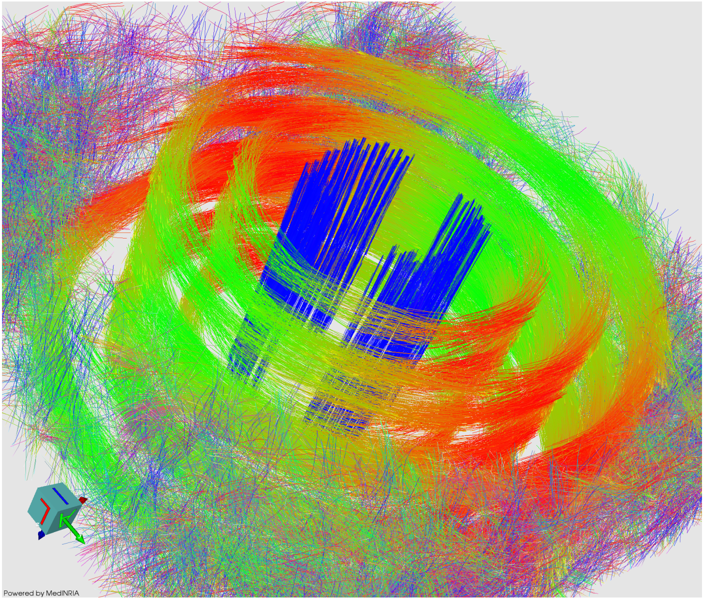


- Reconstruction using non adaptive smoothing

Simulation experiment



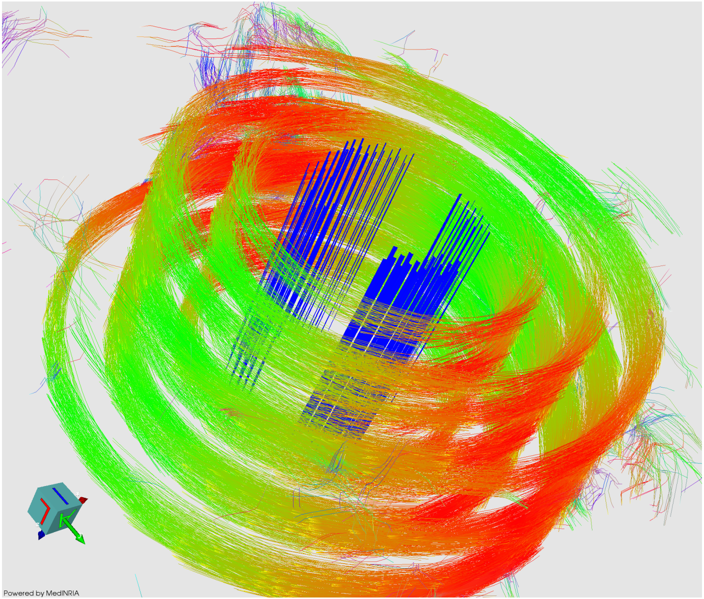
- Reconstruction using Structural Adaptive Smoothing



DTI - Fiber tracking

Artificial example: Fiber tracking based on estimated tensors.

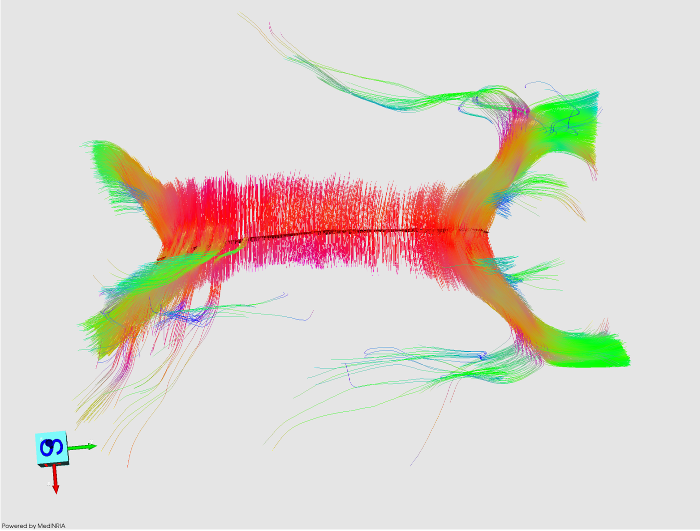
Tracking performed using MedINRIA



DTI - Fiber tracking

Artificial example: Fiber tracking based on smoothed tensors

Tracking performed using MedINRIA

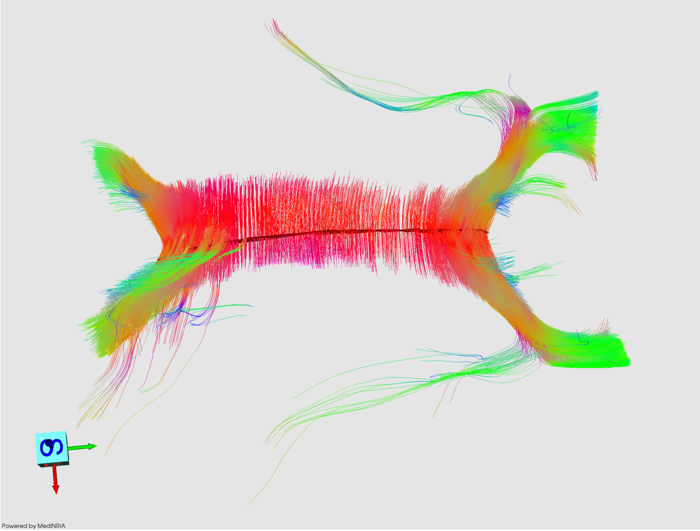


DTI - Fiber tracking

Experimental data: Fiber tracking based on estimated tensors.

Data recorded by and from H. U. Voss.

Tracking performed using MedINRIA



DTI - Fiber tracking

Experimental data: Fiber tracking based on smoothed tensors

Data recorded by and from H. U. Voss.

Tracking performed using MedINRIA

Images

- ▷ Joint demosaicing and smoothing of RAW images.

fMRI

- ▷ high resolution fMRI
- ▷ fMRI for multi-subject studies

Diffusion weighted imaging

- ▷ corrections for Rician bias within the smoothing algorithm
- ▷ improved estimates of diffusion tensors and their variances
- ▷ generalizations bejoint the tensor model (e.g. Q-ball imaging)

Publications (Imaging)

- Polzehl, J. and Spokoiny, V. (2006).
  Propagation-separation approach for local likelihood estimation.
  Probab. Theory Relat. Fields, 135:335-362.
- Polzehl, J. and Spokoiny, V. (2006).
  Structural adaptive smoothing by Propagation-Separation-methods
  to appear in *Handbook of Computational Statistics, Vol. III.*
- Polzehl, J., Tabelow, K. (2007).
  Adaptive Smoothing of Digital Images: The R Package adimpro.
  Journal of Statistical Software, 19/1:17.

Publications (fMRI, DTI)

- Tabelow, K., Polzehl, J., Voss, H. and Spokoiny, V. (2006).
  Analyzing fMRI experiments with structural adaptive smoothing procedures
  Neuroimage, 33:55-62.
- Voss, H.U., Tabelow, K., Polzehl, J., Tchernichovski, O., Maul, K., Salgado-Commissariat, D., Ballon, D. and Helekar, S.A. (2007).
  Functional MRI of the zebra finch brain during song stimulation suggests a lateralized response topography
  Proceedings of the National Academy of Sciences, 104:10667–672.
- Tabelow, K., Polzehl, J., Uluğ, A.M., Dyke, J.P., Watts, R., Heier, L.A., and Voss, H.U. (2006).
  Accurate localization of brain activity in presurgical fMRI by structure adaptive smoothing
  IEEE TMI, to appear.
- Polzehl J. and Tabelow, K. (2007).
  fmri: A package for analysing fMRI data.
  RNews, to appear.
- Tabelow, K., Polzehl, J., Spokoiny, V., Voss, H. (2007).
  Diffusion tensor imaging - spatial adaptive smoothing
  Submitted to Neuroimage

Software

- R-Package *adimpro*: Image processing and adaptive smoothing
- Plugin for AMIRA-Visualization system (ZIB)
- R-Package *fMRI*: Analysis of fMRI time series
- R-Package *dti*: Diffusion tensor imaging (in preparation)

Thank you!

Software

- R-Package *adimpro*: Image processing and adaptive smoothing
- Plugin for AMIRA-Visualization system (ZIB)
- R-Package *fMRI*: Analysis of fMRI time series
- R-Package *dti*: Diffusion tensor imaging (in preparation)

Thank you!