

# Computer Aided Medical Procedures II

## Image Processing: Filtering

Ben Glocker, Andreas Keil, and Nassir Navab

25 April 2007

chair for computer aided medical procedures & augmented reality

department of computer science | technische universität münchen

# Definition of Image

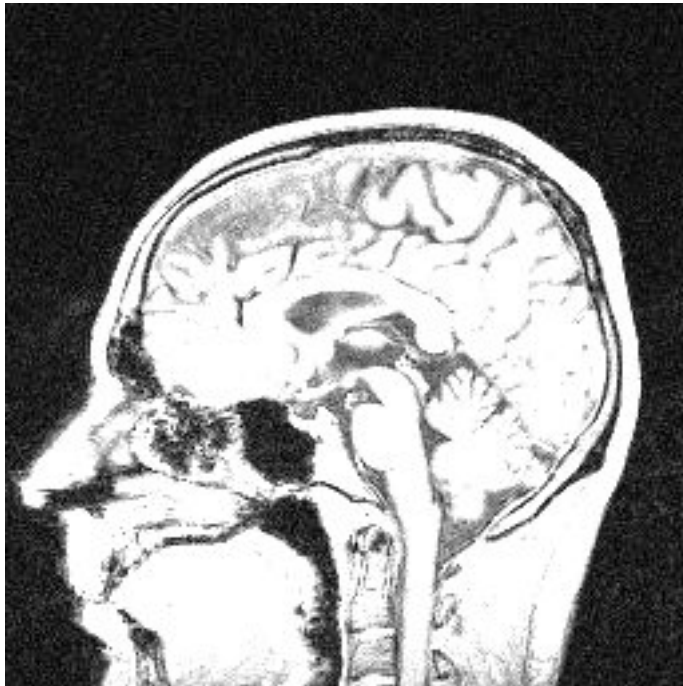
A function  $I: D \rightarrow \mathbf{R}$  with  $D$  being a subset of either  $\mathbf{R}^d$  or  $\mathbf{N}^d$  and  $d$  being the image dimension (2 or 3) is called *image* (or *volume* or *data*).

- Values are intensities (no vector-valued or color images in this presentation)
- Continuous definition but discrete values (e.g. 0, ..., 255) in practice
- Data points  $\mathbf{x} \in \mathbf{N}^d$  are often referred to as “pixels”, if  $d = 3$  they are usually called “voxels”

# Discrete Image



# Windowing



*Bad window for brain matter contrast*



*Good window for brain matter contrast*

## Windowing (continued)

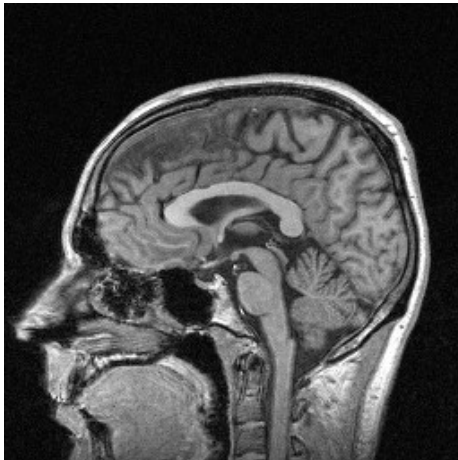
- Adjusting brightness and contrast = “windowing”
  
- Window: Range of values mapped onto intensities (black – white), described by
  - level = center of window =  $(\text{maxDisplay} + \text{minDisplay}) / 2$
  - width = range width =  $\text{maxDisplay} - \text{minDisplay}$or
  - white =  $\text{maxDisplay} = \text{level} + \text{width}/2$
  - black =  $\text{minDisplay} = \text{level} - \text{width}/2$

## Windowing (continued)

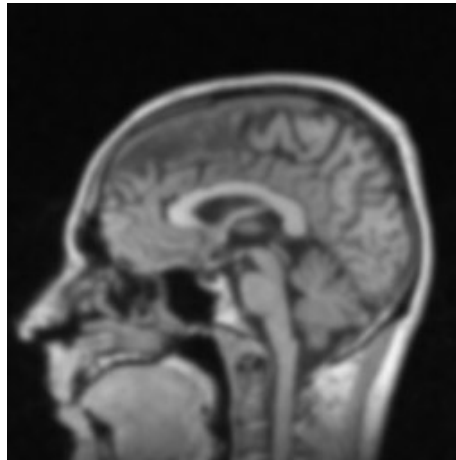
- Point operation / gray value transformation
- Usually, image data remains unchanged while mapping from values to intensities is adjusted
- Example: Preset windows used for better visibility of certain anatomical structures in CT workstations

# Filtering

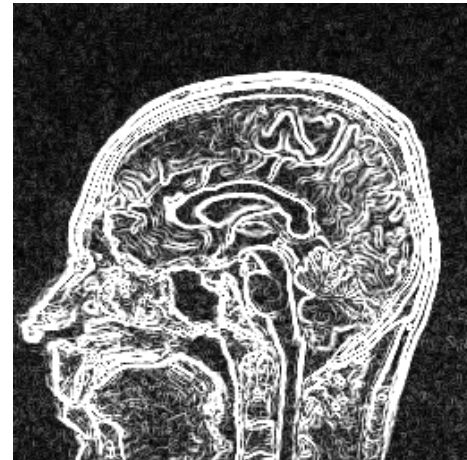
- Smoothing and edge detection are filtering operations



*Original image*



*Smoothed image*



*Edge-filtered image*

- Classification in
  - Linear filtering (in discrete or continuous domain)
  - Advanced filtering (median, anisotropic diffusion, Fourier)

# Linear Filtering in Continuous and Discrete Domain

- Convolution with kernel (continuous)

$$(w * g)(\mathbf{x}) = \int_{\mathbf{R}^d} w(\boldsymbol{\xi})g(\mathbf{x} - \boldsymbol{\xi}) d\boldsymbol{\xi}$$

or mask (discrete)

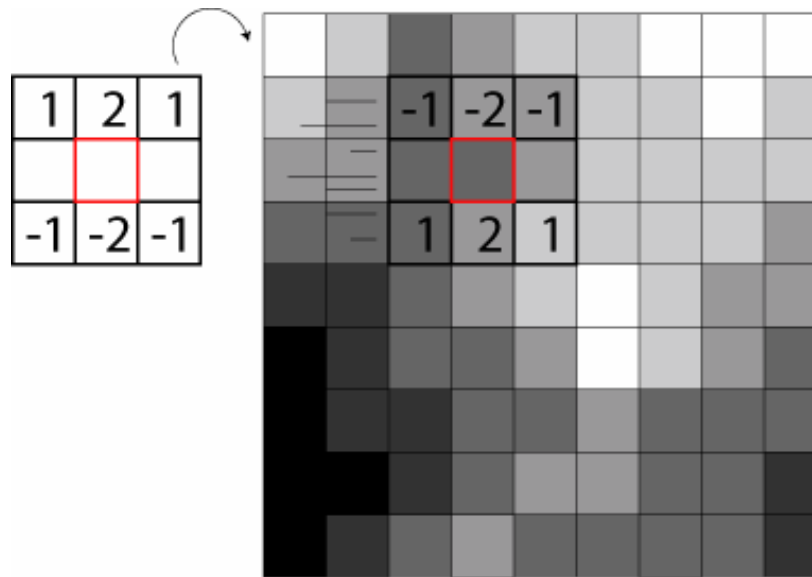
$$(w * g)(\mathbf{x}) = \sum_{\xi_1=-a}^a \cdots \sum_{\xi_d=-a}^a w(\boldsymbol{\xi})g(\mathbf{x} - \boldsymbol{\xi}) d\boldsymbol{\xi}$$

# Linear Filtering in Discrete Domain

- Discrete Convolution

$$(w * g)(\mathbf{x}) = \sum_{\xi_1=-a}^a \cdots \sum_{\xi_d=-a}^a w(\xi)g(\mathbf{x} - \xi) d\xi$$

is equivalent to correlating a flipped mask with the image:



# Linear Filtering in Continuous and Discrete Domain (continued)

Correspondence Convolution  $\leftrightarrow$  Correlation

□ Convolution:

$$(w * g)(\mathbf{x}) = \int_{\mathbf{R}^d} w(\boldsymbol{\xi})g(\mathbf{x} - \boldsymbol{\xi}) d\boldsymbol{\xi}$$

□ Correlation:

$$(w * g)(\mathbf{x}) = \int_{\mathbf{R}^d} w(\boldsymbol{\xi})g(\mathbf{x} + \boldsymbol{\xi}) d\boldsymbol{\xi}$$

$\Rightarrow$  Convolution is a correlation with a flipped mask

Attention: Convolutions are commutative, correlations are not!  
(Proof?)

# Smoothing in Discrete Domain

- Calculation of mean in 8 neighborhood is equivalent to applying the linear filter

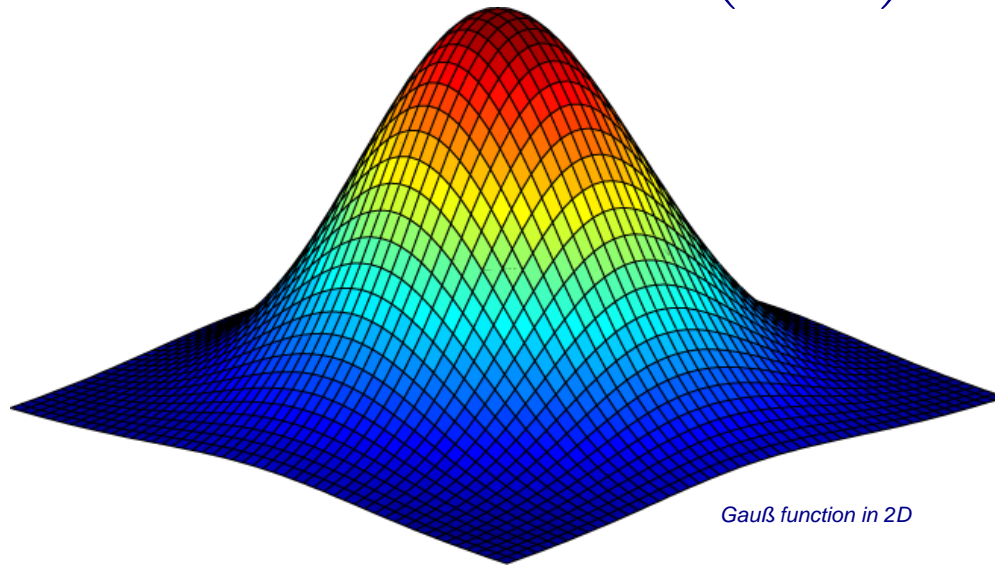
$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

- Some discrete approximations of the Gaussian function with integer values

$$\frac{1}{16} \cdot \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array} \quad \text{or} \quad \frac{1}{115} \cdot \begin{array}{|c|c|c|c|c|} \hline 2 & 4 & 5 & 4 & 2 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 5 & 12 & 15 & 12 & 5 \\ \hline 4 & 9 & 12 & 9 & 4 \\ \hline 2 & 4 & 5 & 4 & 2 \\ \hline \end{array}$$

# Smoothing in Continuous Domain

- Gauß function  $G_\sigma(\mathbf{x}) = (2\pi\sigma^2)^{-\frac{d}{2}} \exp\left(-\frac{|\mathbf{x}|^2}{2\sigma^2}\right)$ ,  $\mathbf{x} \in \mathbf{R}^d$



Gauß function in 2D

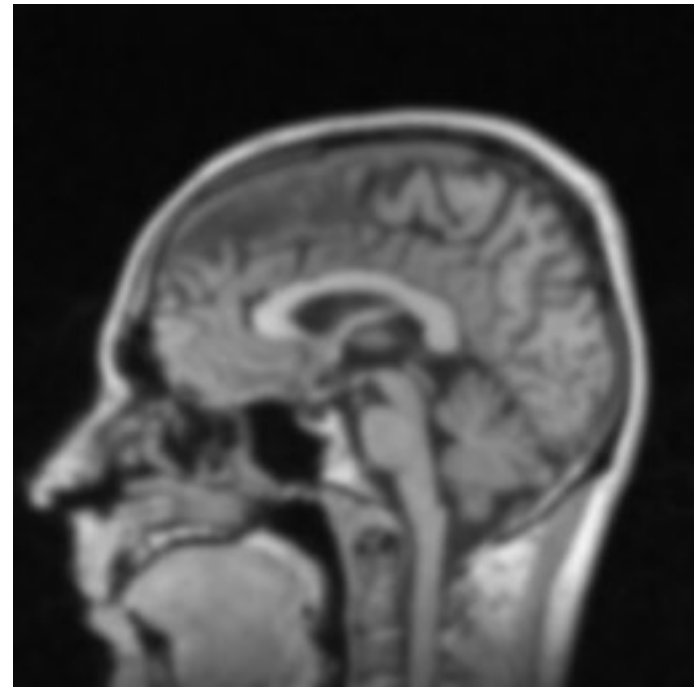
- Smoothing by convolution  $I_\sigma = G_\sigma * I$

# Smoothing (continued)

Full image:



*Original Head MR Image*



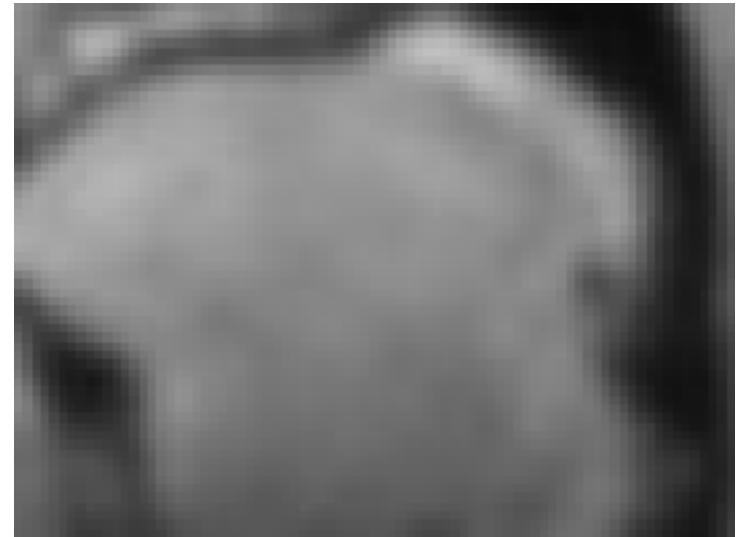
*Smoothed Image*

# Smoothing (continued)

Desired smoothing inside homogeneous regions:



*Original Head MR Image*



*Smoothed Image*

# Smoothing (continued)

Undesired smoothing across regions/edges:



*Original Head MR Image*



*Smoothed Image*

# Equivalence to Heat Diffusion, Anisotropic Diffusion Filtering

- Observation: Convolution with Gauß function is solution to heat diffusion equation

$$\partial_t u = c \cdot \Delta_{\mathbf{x}} u$$

- Idea: Create smoothing filter based on heat diffusion with locally varying conductance parameter  $c(\mathbf{x})$

$$\begin{aligned}\partial_t I &= \operatorname{div}_{\mathbf{x}}(c(\mathbf{x}) \cdot \nabla_{\mathbf{x}} I) \\ &= \nabla_{\mathbf{x}}^{\top}(c(\mathbf{x}) \cdot \nabla_{\mathbf{x}} I) \\ &= \nabla_{\mathbf{x}}^{\top} c(\mathbf{x}) \cdot \nabla_{\mathbf{x}} I + c(\mathbf{x}) \cdot \Delta_{\mathbf{x}} I\end{aligned}$$

# Equivalence to Heat Diffusion, Anisotropic Diffusion Filtering (continued)

- Conductance with gradient dependant, decreasing function  $g$

$$c(\mathbf{x}) = g(|\nabla I(\mathbf{x})|)$$

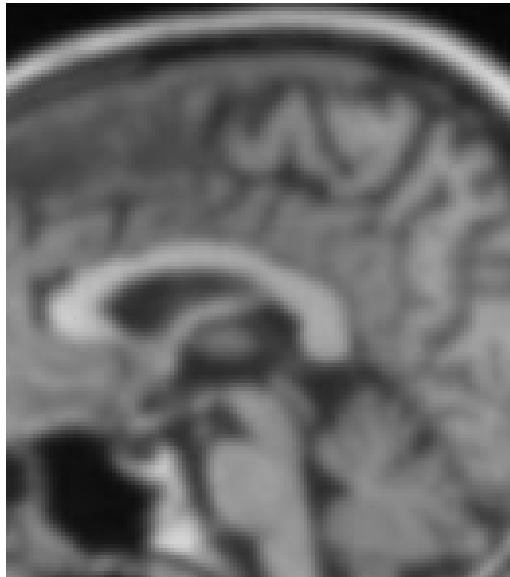
⇒ High gradient magnitudes yield low conductance and therefore less smoothing

- Result: Better preservation of edges

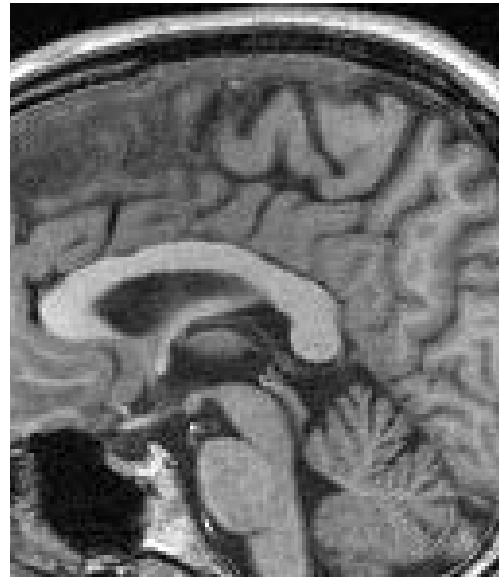
- Examples for decreasing  $g$ 's:

$$g(|\nabla I|) = \frac{1}{1 + \left(\frac{|\nabla I|}{K}\right)^2} \quad \text{or} \quad g(|\nabla I|) = e^{-\left(\frac{|\nabla I|}{K}\right)^2}$$

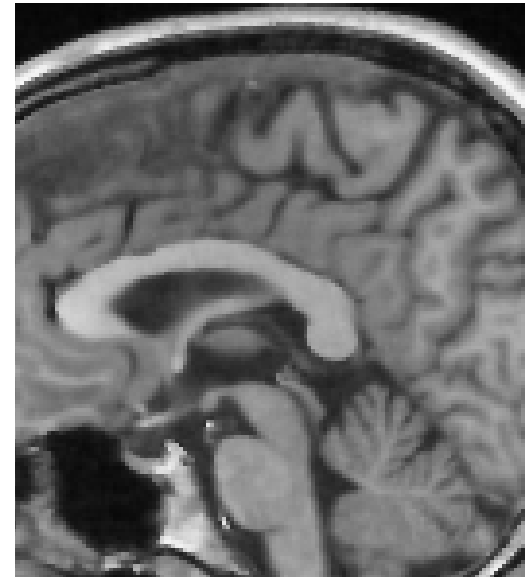
# Equivalence to Heat Diffusion, Anisotropic Diffusion Filtering (continued)



*Gaussian Smoothing*



*Original*



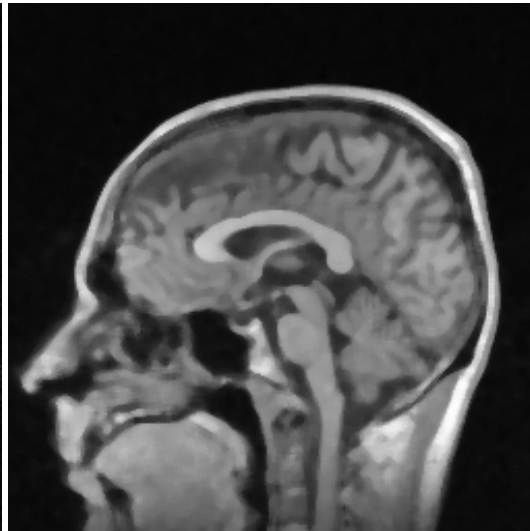
*Anisotropic Diffusion Filtering*

# Median Filtering

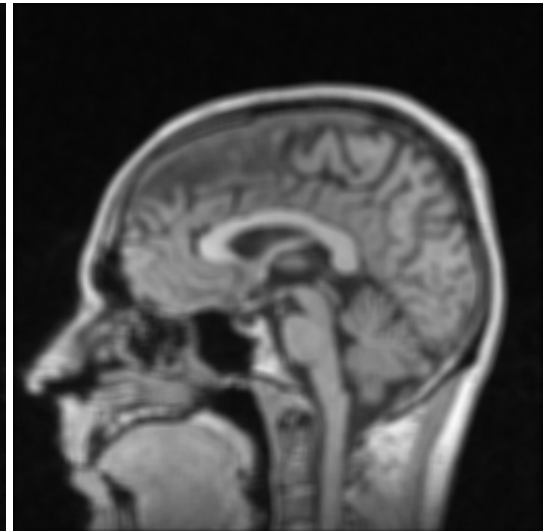
- Another nonlinear type of filter
- Sort intensity values from the pixel's neighborhood and keep the middle one
- Less blurring compared to averaging or Gaussian smoothing
- Cannot be represented using convolution or a mask



*Original*



*Median Filtering with a 5x5 mask*



*Gaussian Smoothing with a 5x5 mask*

# Edge Detection

- Approximation of continuous differentiation operators with finite differences

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1}))}{x_{i+1} - x_{i-1}} \Rightarrow \frac{1}{2} \cdot \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline \end{array}$$

→ Discrete filtering

## Edge Detection (continued)

- Sobel filter masks (first order differential including smoothing)

1	0	-1
2	0	-2
1	0	-1

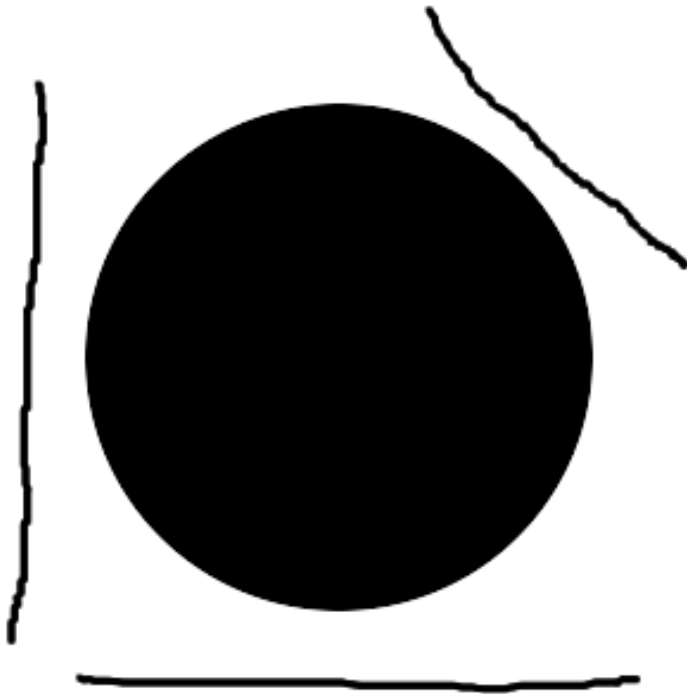
1	2	1
0	0	0
-1	-2	-1

- Laplacian filter mask (second order differential)

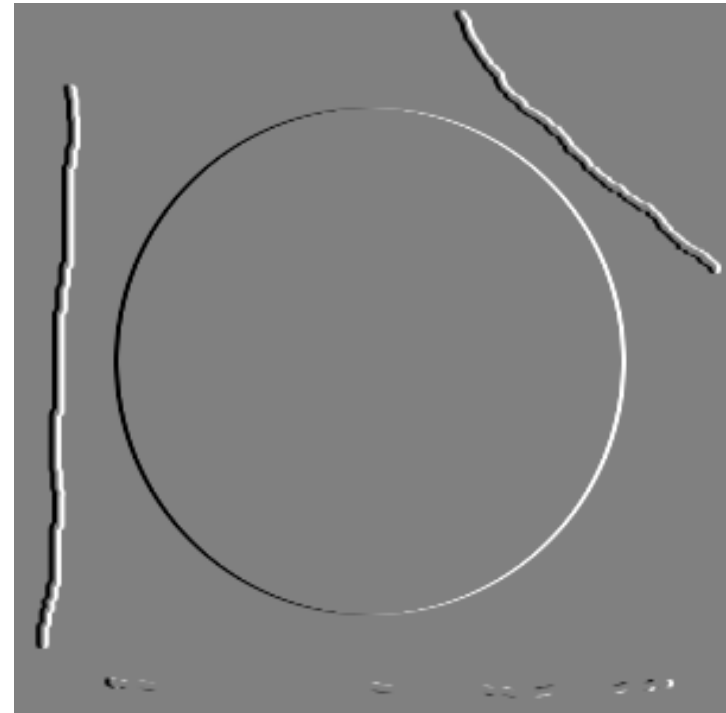
0	1	0
1	-4	1
0	1	0

# Edge Detection (continued)

Example for vertical Sobel operator:



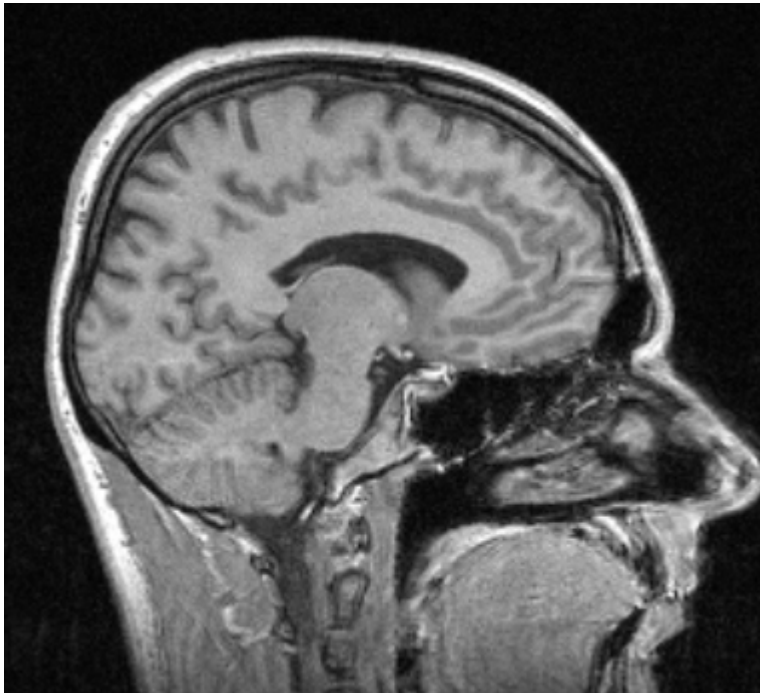
*Artificial example (original)*



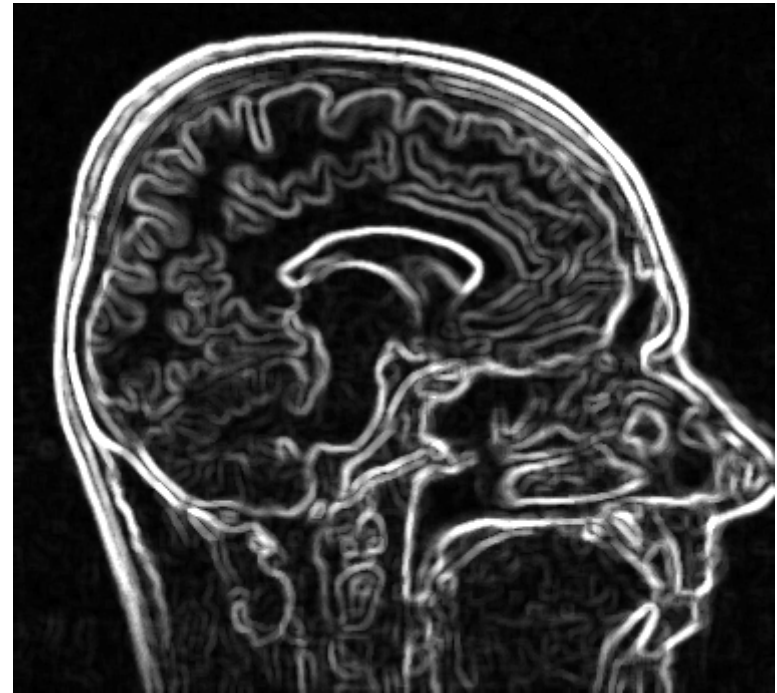
*Result of vertical Sobel*

## Edge Detection (continued)

Example for gradient magnitude operator (non-linear!):



*Original head MR image*



*Gradient magnitude of smoothed image*

# A Note on Efficiency: Smoothing and Edge Detection in One Step

- Order of linear(!) operations can be exchanged
- Application to smoothing and edge detection:  
$$D(G_\sigma * I) = (DG_\sigma) * I = G_\sigma * (DI)$$
- Middle formula is most efficient, since differential of Gaussian can be precomputed
- Example: Sobel filter is a combination of Gaussian smoothing and edge detection

- Another example: Laplacian of Gaussian

$$\Delta(G_\sigma * I) = (\Delta G_\sigma) * I$$

# An Application-Tailored Filter: The Frangi Filter

- A nonlinear filter employing Eigenvalue analysis for delineating tubular structures like blood vessels
- Bright tubular structure on dark background in 3D  $\Rightarrow$ 
  - One eigenvalue close to zero (corresponding eigenvector in the direction of the tubus)
  - Two other eigenvalues with large magnitude compared to the first one and negative sign
- Different diameters of blood vessels require computations on different scales

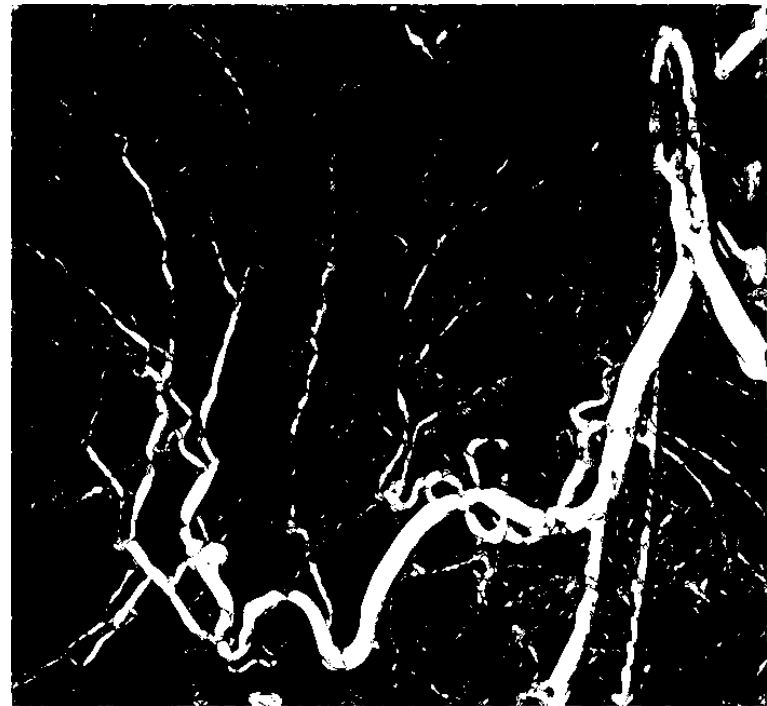
## The Frangi Filter (continued)

- Compute Hessian matrix  $H$  for every voxel
- Compute Eigenvalues and sort them so that  $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$
- Calculate 3 values:
  - Blobness measure  $R_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2 \lambda_3|}}$
  - Distinguishing plate-like and line-like structures  $R_A = \frac{|\lambda_2|}{|\lambda_3|}$
  - Structureness  $S = \|H\|_F = \sqrt{\sum_{i=1}^3 \lambda_i^2}$
- Vesselness measure at scale  $s$  is a combination of the three values above
- Compute maximum across different scales

# The Frangi Filter (continued)



*Pre-filtered DSA image of liver vessels*



*Frangi-filtered image*

A. F. Frangi, W. J. Niessen, K. L. Vincken, M. A. Viergever. Multiscale vessel enhancement filtering. MICCAI 1998, LNCS 1496