Pattern Recognition
Chapter 3:
Image Acquisition and Preprocessing

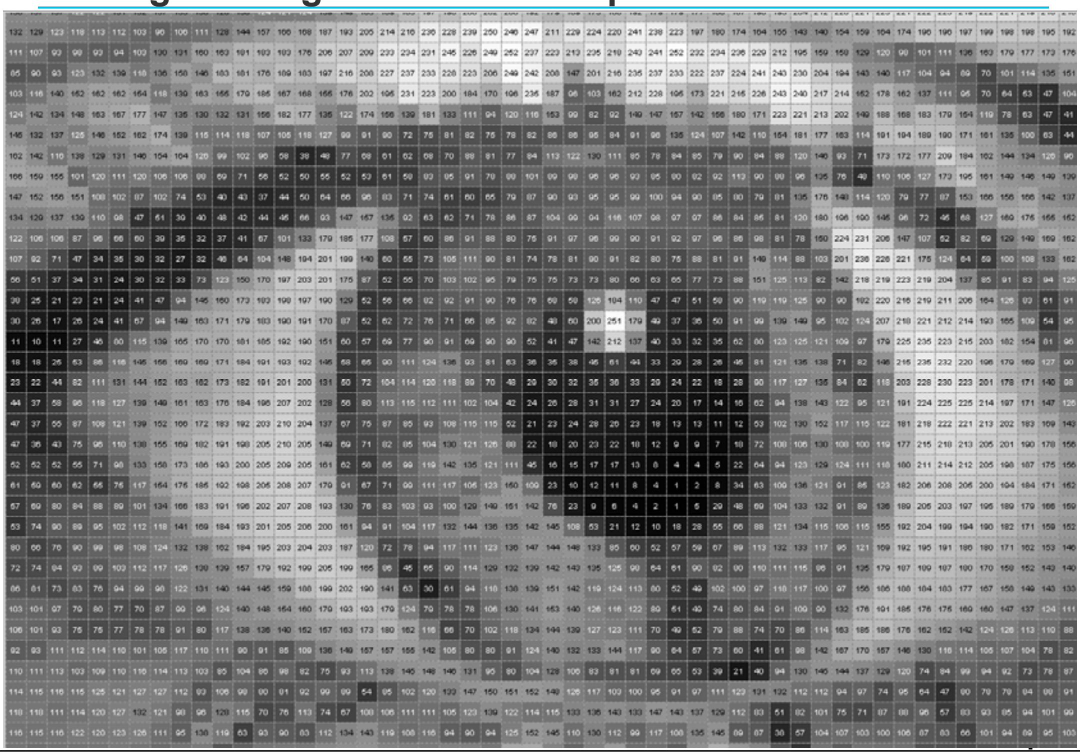
Prof. Dr.-Ing. Uwe Sörgel
soergel@ifp.uni-stuttgart.de

Contents

Level model of model-based image analysis

top-down ↓ / bottom-up ↑	**Recognition/Evaluation** Features → Objects	high-level
	Segmentation Image → Features	mid-level
	Preprocessing Image → Image	low-level
	Image acquisition Scene → Image	

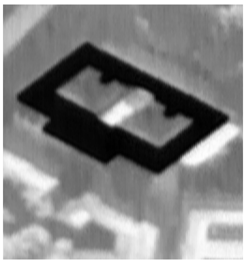
The digital image as a matrix of pixels

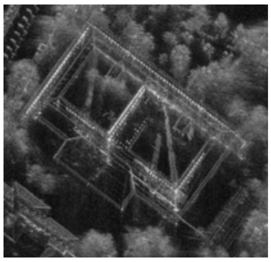


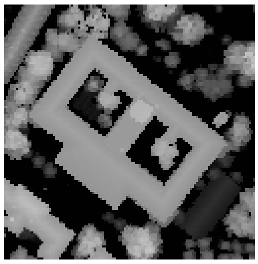
Many entities can be visualized as image

- In principle, any mathematical function of type $g=f(x,y)$ can be coded as image.
  - Height with respect to some reference
  - Chemical concentration
  - Temperature
  - Pressure
- Some examples related to our field:


Aerial image


Infrared image (IR)


Radar image (SAR)


Laserscan height

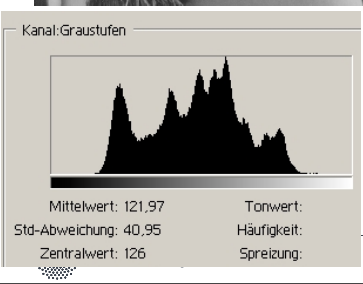
Level model of model-based image analysis

top-down ↓ / bottom-up ↑		
	Recognition/Evaluation Features → Objects	high-level
	Segmentation Image → Features	mid-level
	Preprocessing Image → Image	low-level
	Image acquisition Scene → Image	

Contents

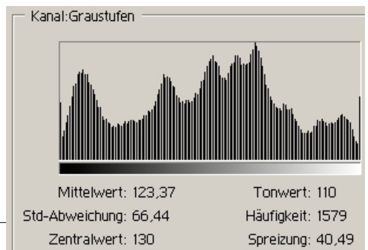
- Image Acquisition
- Preprocessing
  - Image enhancement
  - Image restauration
- Calculation of derivations

Example for histogram normalisation



original

On both ends 0.5% of values forced into saturation



Problem statement of image restauration

- The „true“ image $g(x,y)$ is corrupted by some distortion $n(x,y)$.
- For example, additive zero mean Gaussian noise $\eta(x,y)$:

$$g_R(x,y) = g(x,y) + \eta(x,y)$$

- There are also other kinds of distortions like „Salt-and-Pepper“ noise (randomly distributed white and black noise pixels)

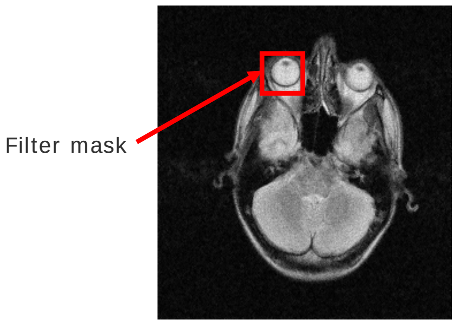
Original

Gaussian noise

„Salt-and-Pepper“

Local filter operation

- Point operations only consider the current pixel value
- Global transformations use the whole image
- *Local filter operations* are a mixture of both:
  - Neighbourhood around current pixel is considered.
  - Neighbourhood is defined via a window (filter mask).
  - Filter mask is continuously moved across the image.
  - In each position value of one pixel (usually the central pixel) is determined.



Different types of local filters

- Linear Filters
  - Linear digital filters $h(x,y)$ carry out convolution on the image $g(x,y)$.
  - The coefficients of filter matrix represent weights, which are multiplied with corresponding pixel values, before adding all products.
  - For homogeneous filters the coefficients are constant, independent of image location (shift invariant)

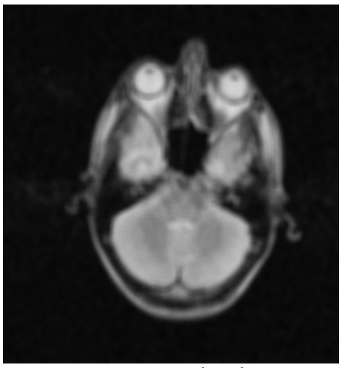
$$h(x,y)=\begin{bmatrix}-1 & -2 & -1\\ -2 & 12 & -2\\ -1 & -2 & -1\end{bmatrix}$$

  - Non-linear filters, for example.:
    - Rank filter
      - Median filter
      - Morphological filter
    - Diffusion filter

Application of linear local filters

- Examples for typical application of some linear local filter
  - Smoothing
  - Enhancement of
    - Salient points
    - Edges
- Can not be achieved by point operations→ Context required

Original



Smoothed

Gradient image

Box filter (rectangular filter)

• The most simple smoothing filter is the box or rectangular filter of size 3 x 3.

$$h(x,y)=\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}\Rightarrow g'(x,y)=g*h=\frac{1}{9}\sum_{i=-1}^{i=1}\sum_{j=-1}^{j=1}g(x+i,y+j)$$

Example

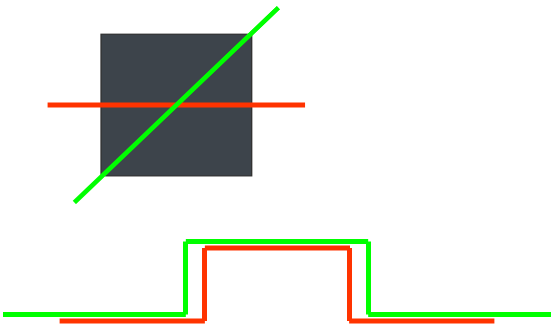
g	..	**93**	..
:	113	116	104
15	99	101	125
:	0	107	105

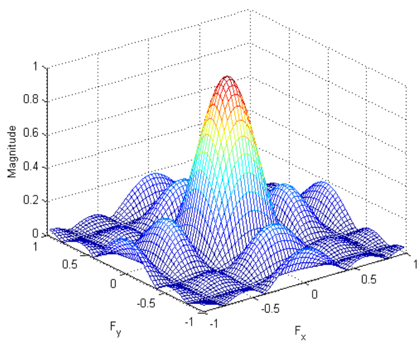
Element “101” at position (93,15)

$$g'(93,15)=\frac{1}{9}\left(113+116+104+99+101+125+0+107+105\right)=\underline{\underline{96.7\approx 97}}$$

Disadvantages of the box filter

• Box filters are no ideal low-pass filters:





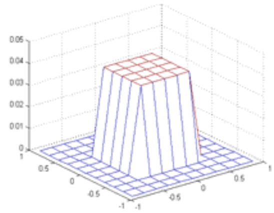
Transfer function

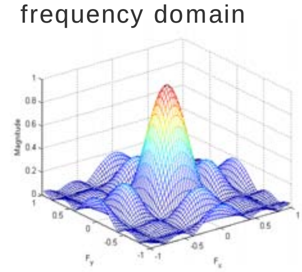
- The box filter is rotation variant
  - Orientation dependency (diagonal longer than edges)
  - Non-uniform suppression of high frequencies
  - Undesired sidelobes may cause artifacts

Comparison of box and Gauss operator

spatial domain | frequency domain

box operator

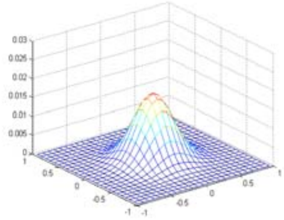


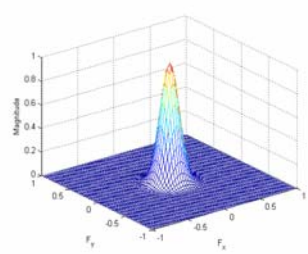


disadvantage: not rotation invariant, does not suppress all high frequencies
advantage: recursive implementation possible

Gauss operator

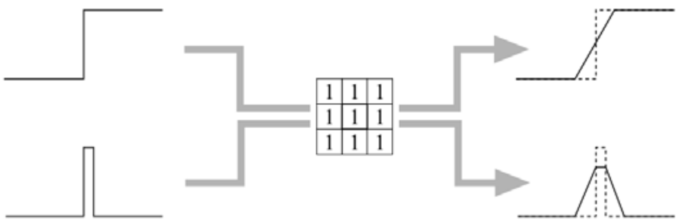
$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$





advantage: rotation invariant, suppresses all high frequencies
disadvantage: only approx. recursive implementation -> binomial filter

Disadvantage of linear smoothing filters

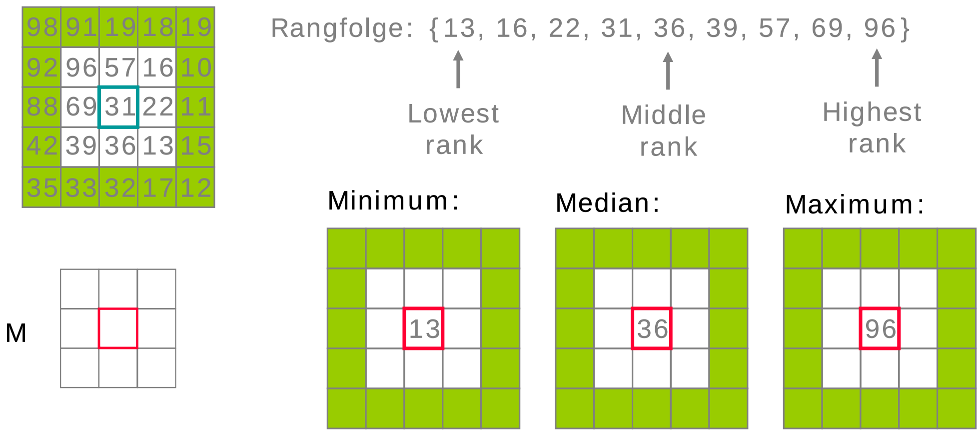


Blur at edges and lines

Salt-and-Pepper-Noise not removed

Rank filter

- The grey values inside mask are sorted in ascending manner.
- The choice of the rank (i.e., index) depends on the desired purpose
  - Minimum filter: Lowest rank
  - Maximum filter: Highest rank
  - Median filter: Middle rank



Example of application of median filter

- Removes Salt-and-Pepper noise
- Mostly quite convincing results

Original image | Original image + Salt-and-Pepper noise | After Median filtering (3x3)

Contents

- Image Acquisition
- Preprocessing
  - Image enhancement
  - Image restauration
- Calculation of derivations

The derivations of an image and their meanings

1. Derivation

$$g_x(x,y)=\frac{\partial g(x,y)}{\partial x},\ g_y(x,y)=\frac{\partial g(x,y)}{\partial y}$$

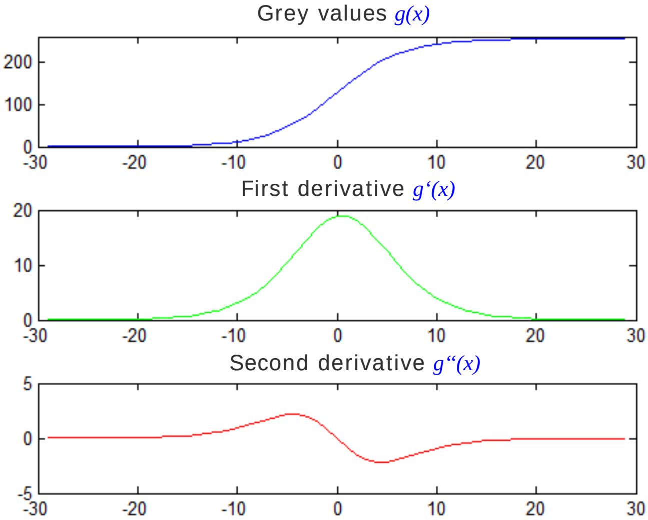
- Digital images: Change of grey values → edges, points
- Digital surface models: normal vectors, height jumps

2. Derivation

$$g_{xx}(x,y)=\frac{\partial^2 g(x,y)}{\partial x^2},\ g_{yy}(x,y)=\frac{\partial^2 g(x,y)}{\partial y^2},\ g_{xy}(x,y)=\frac{\partial^2 g(x,y)}{\partial x \partial y}$$

- Digital images: At edges the 2. derivation vanishes
- Digital surface models: Curvature, torsion

High pass filters: Edge detection

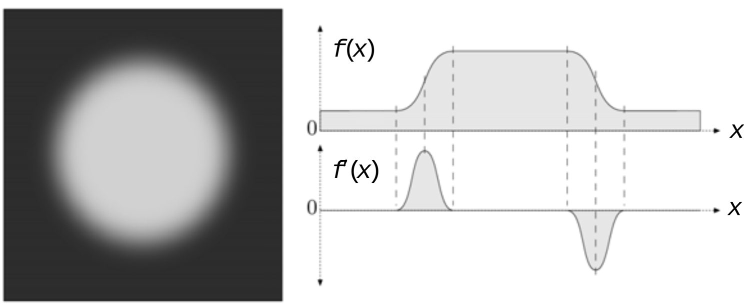


edge = strong straight change of grey values $g(x)$

gradient operator:
1. derivative $g'(x)$
➔ Search for maxima

Laplace-Operator:
2. derivative $g''(x)$
➔ Search for zero crossings

Gradient based edge detection

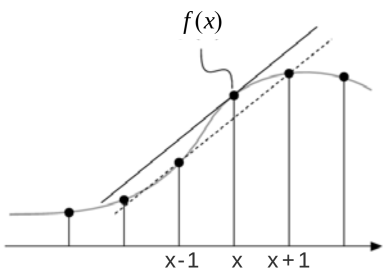


Discrete approximation of gradient by differencing

- Take left and right value to end up at grid position.
- Σ (filter coeff.) = 0 (i.e., high-pass filter)

$$f'(x) = \frac{\partial f(x)}{\partial x} \approx \frac{f(x+1) - f(x-1)}{2} =$$

$$= 0.5 \cdot \left(f(x+1) - f(x-1) \right)$$



Prewitt-Operator

Simple realization:

$$f'(x) \approx 0.5 \cdot \big(f(x+1) - f(x-1) \big) \longrightarrow h_x = 0.5 \cdot \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$f'(y) \approx 0.5 \cdot \big(f(y+1) - f(y-1) \big) \longrightarrow h_y = 0.5 \cdot \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

However, *differencing enhances noise*:

→ noise suppression by low pass filtering (smoothing) in across direction!

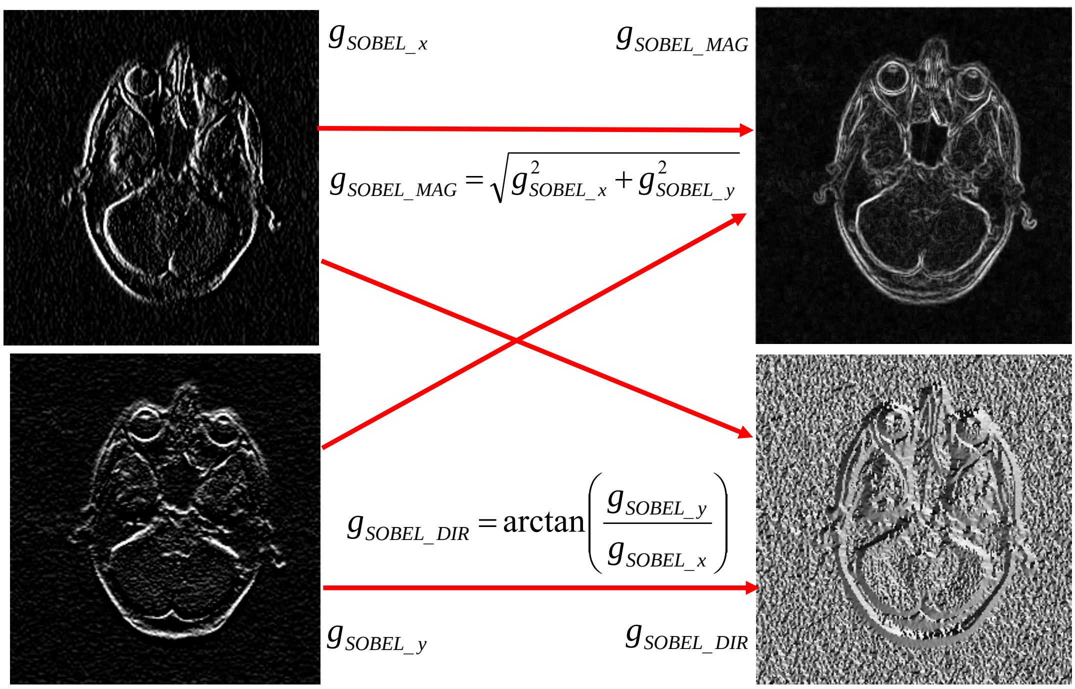
gradient →, smoothing ↓

$$h_x = \frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

smoothing →, gradient ↓

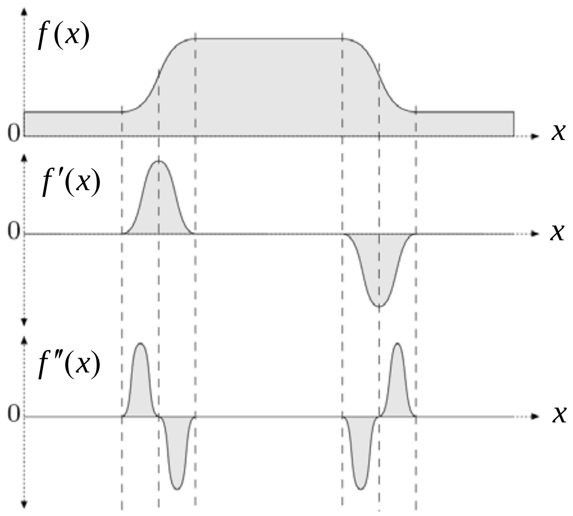
$$h_y = \frac{1}{6} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Amplitude and direction of gradient



Laplace operator

- Alternative to gradient filtering (1. derivative)
- Search for zero crossings of 2nd derivative.



Laplace operator

Laplace Operator Δ computes sum of 2nd partial derivatives of a continuous function $g(x,y)$ for the variables x und y:

$$\Delta g(x,y) = \frac{\partial^2 g(x,y)}{\partial x^2} + \frac{\partial^2 g(x,y)}{\partial y^2}$$

Discrete realization of 2nd derivative: difference of difference of neighboring pixels per row and column

$$\frac{\partial^2 g(x,y)}{\partial x^2} \approx \frac{g(x+1,y)-g(x,y)}{(x+1)-x} - \frac{g(x,y)-g(x-1,y)}{x-(x-1)}$$

$$\approx g(x+1,y) - 2\cdot g(x,y) + g(x-1,y)$$

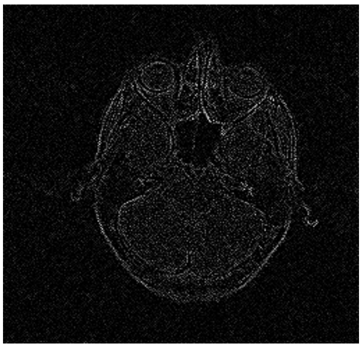
$$\frac{\partial^2 g(x,y)}{\partial y^2} \approx g(x,y+1) - 2\cdot g(x,y) + g(x,y-1)$$

Laplace-Operator for 2D

- It is better to add the two filter masks for each direction in order to yield the result in only a single run:

$$h_\Delta = h_{\Delta x} + h_{\Delta y} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

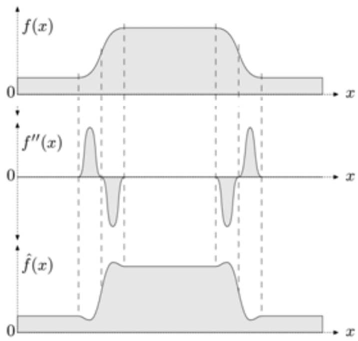
Original



Result

Unfortunately, besides desired object contours also noise is enhanced.

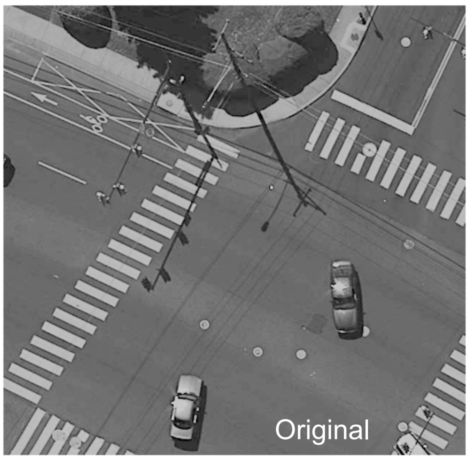
Laplace operator: Image sharping

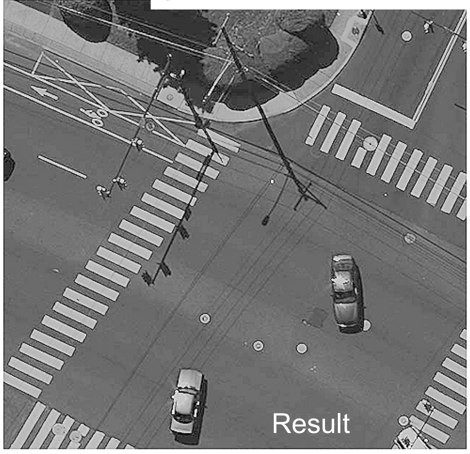


- "Sharper" image by adding high-pass signal:

$$h_{Sharping} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

"Image" - k · Laplacian = Sharpening





Laplace operator in presence of noise

- The Laplace operator is very noise sensitive because difference operation enhances noise – now we have two of them in a row…

LoG operator (Laplace of Gaussian)

- In order to mitigate noise, it is beneficial to smooth the image (G*image) before subsequent convolution with Laplace Operator (Δ(G*image)).
- Alternatively, - same as for gradient approach - we rather should calculate the 2nd derivation of Gaussian first and use this result for convolution ((ΔG)*image):

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$LoG: \Delta G = \frac{\partial^2 G(x,y)}{(\partial x)^2} + \frac{\partial^2 G(x,y)}{(\partial y)^2}$$

$$\frac{\partial G(x,y)}{\partial x} = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \cdot \frac{-x}{\sigma^2}$$

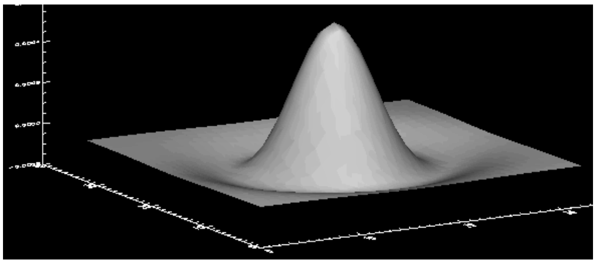
$$\frac{\partial^2 G(x,y)}{(\partial x)^2} = \frac{-1}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} + \frac{1}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \cdot \frac{x^2}{\sigma^2}$$

$$= \frac{1}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \left[\left(\frac{x}{\sigma}\right)^2 - 1 \right]$$

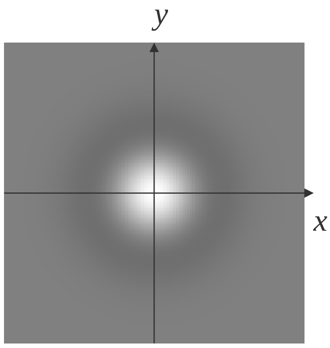
$$\frac{\partial^2 G(x,y)}{(\partial y)^2} = \frac{1}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \left[\left(\frac{y}{\sigma}\right)^2 - 1 \right]$$

LoG-Operator (Laplace of Gaussian)

$$LoG(x,y) = \frac{1}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \left[\frac{x^2+y^2}{\sigma^2} - 2 \right]$$



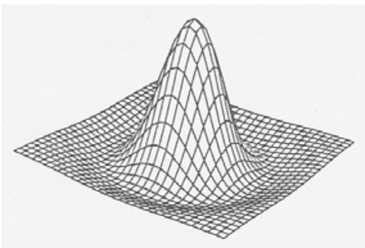
LoG Operator

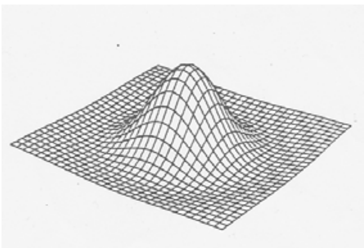


LoG-Operator (Laplace of Gaussian)

- Because of similarity with Sombrero this operator is sometime called Mexican-Hat-Operator.
- The result depends on the choice standard deviation σ.
- An edge is inferred from zero crossings of the 2nd derivation.
- In contrast to gradient methods, which always rely to some threshold, we yield always closed curves

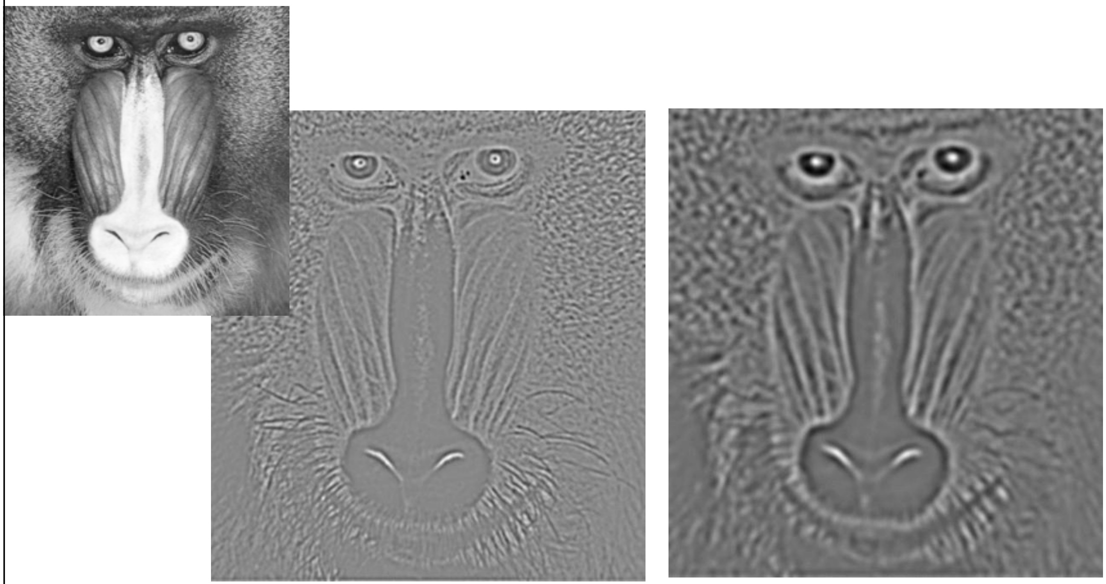
σ=4





σ=5

LoG-Operator: Example



LoG, $\sigma = 2.0$ LoG, $\sigma = 4.0$

DoG operator (Difference of Gaussians)

- The very good LoG operator can be quite good approximated by the difference of two low-pass filterings with Gaussians of different σ.
- However, the ration of standard deviations is recommended to be σ_1/σ_2=1.6.
- Similar processing I taking place in visual system of humans.

1d

2d