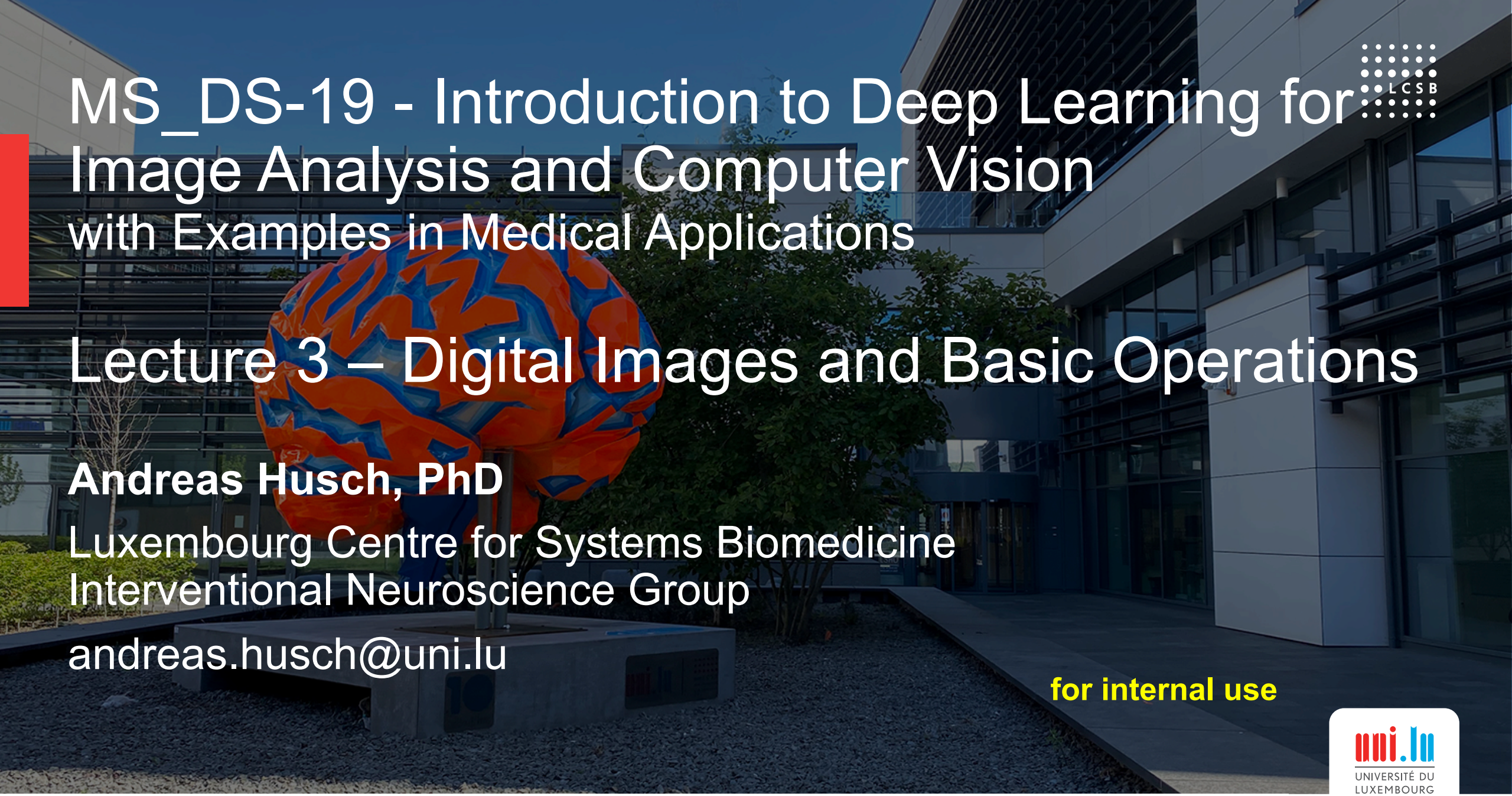


MS_DS-19 - Introduction to Deep Learning for Image Analysis and Computer Vision

with Examples in Medical Applications

Lecture 3 – Digital Images and Basic Operations

Andreas Husch, PhD

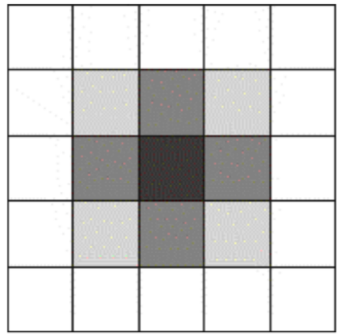
Luxembourg Centre for Systems Biomedicine
Interventional Neuroscience Group

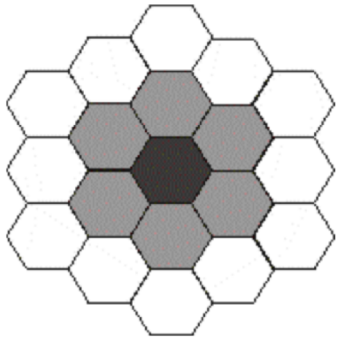
andreas.husch@uni.lu

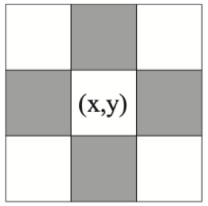
for internal use

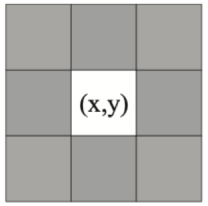
Recap last lecture

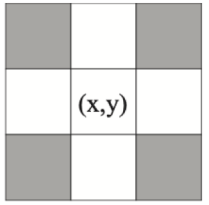
- Grids
- Neighborhoods
- Connectivities
- Point and Neighborhood Operations incl. Histogram Ops.
- Filtering

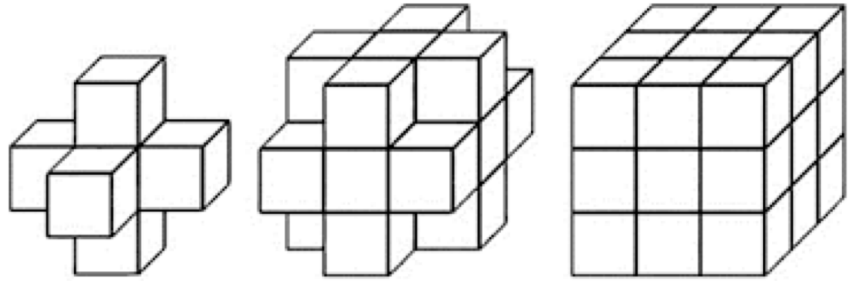


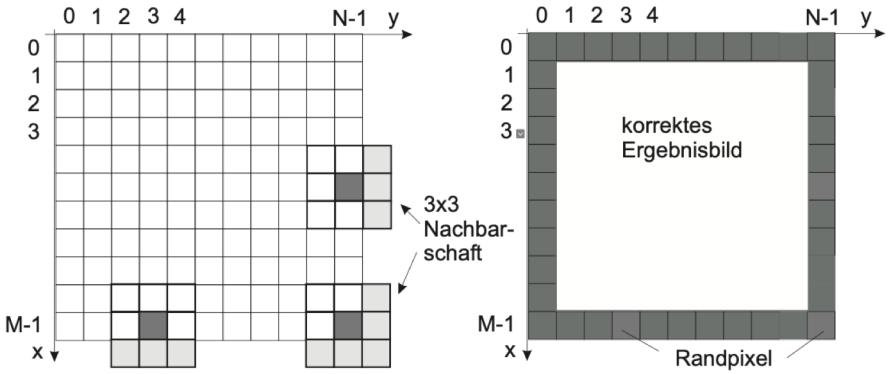


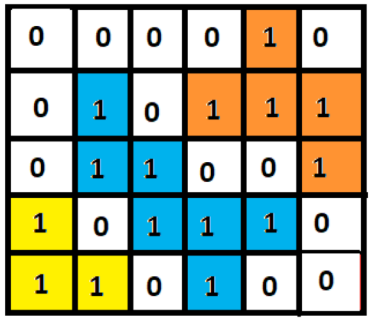




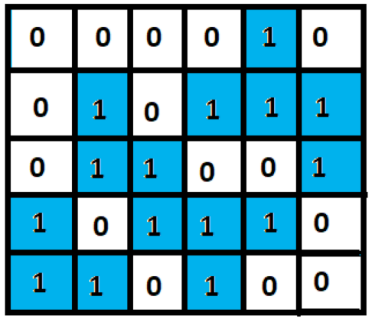








4-connectivity map



8-connectivity map

Image enhancement

- Subjective image enhancement - for the human viewer, so that he or she can, for example, more easily and reliably grasp the image content.
- Objective image enhancement - for machine evaluation, (e.g. noise suppression or contour sharpening).

Example (subjective)

- **Subjective** image enhancement, e.g. gamma correction

$$f(x,y) = c \cdot b^{\gamma}(x,y) \quad c, \gamma \in \mathbf{R} > 0$$

original

LUT–transform γ= 0.5

Q.: How could we implement this in a line of MATLAB code?

Example (objective)

- **Objective** Image enhancement, e.g. nonlinear noise reduction

$$f(x,y) = MEDIAN(b(r,s)_{\forall(r,s)\in N(x,y)}) \quad \text{with} \quad N(x,y) = R_{xy}^{33}$$

original + salt & pepper – noise

noisy image median filtered

Q.: How could we implement this in MATLAB code? How does it compare to an average filter? (Exercise)

Taxonomies for image operations I

- Argument range
  - Global argument of the function is the whole image
  - Local argument of the function is a local area for example a neighborhood N
- Point-based
  - Argument of the function is a pixel (point operation) Processing form.
- Homogeneous
  - Same function definition for all image positions (x,y)
- Inhomogeneous
  - Calculation depends on the image positions (x,y).

Image enhancement

- Goal: Processing in such a way that the resulting image is better suited for a specific application/evaluation.
- Application background: Methods for image enhancement (enhancement techniques) are (usually) always problem-oriented; (e.g. evaluation of satellite images vs. X-ray images).
- Conclusion: ***No* general method** can be given *(cf. no free lunch),* which would be equally suitable for any application. Under certain circumstances, a certain method destroys (eliminates) information that is relevant for a (different) special evaluation.

Terms

- Image restoration - refers to the restoration of the original, i.e. restoration with respect to recording errors, noise, etc.

- Image enhancement - refers to the (restored) image signal and serves to emphasize (enhance) certain image contents such as edges, suppression of local gray value fluctuations, etc. for subsequent evaluation.

Interferences

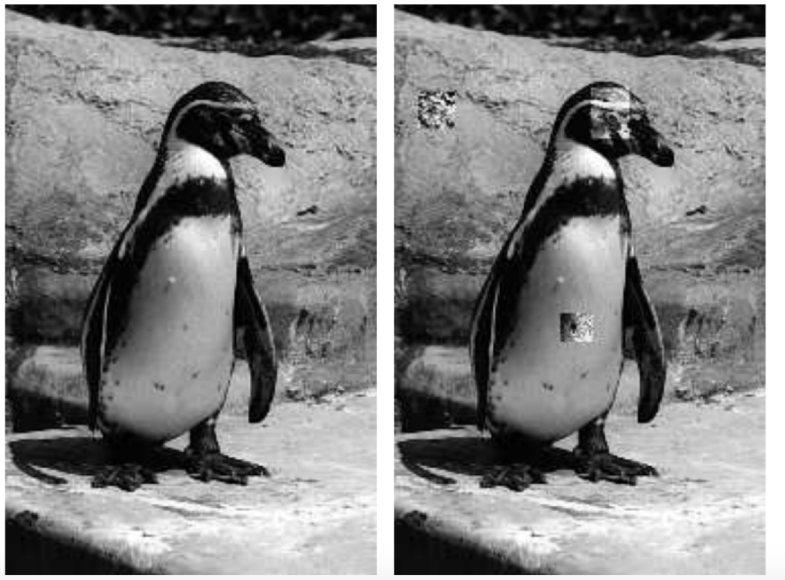
In practice, image information is *disturbed* by interfering information.

Interferences: specifications

Disturbance	Cause	Solution
Noise	Sensor and transmission system	Image smoothing (e.g. average, low pass)
Ramps (in the brightness curve)	inhomogeneous illumination	Local adaptation (e.g. moving average)
Blur	aberration, movement	Image sharpening (e.g. contrast sharpening)
Geometric Distortions	Imaging geometry	Equalization (e.g. affine mapping)

Global vs. local enhancement

- The methods discussed so far were global
- The transformation function was based on the intensity value distribution of the entire image.
- Global enhancement effects could be achieved
- Local enhancement/suppression must take into account the local distribution of pixel values.



Additive image noise

- Let an image $g(x,y)$ be disturbed by noise
- The noise components $\eta(x,y)$ are described by a random variable Z_η: with the properties $E(Z\eta) = 0$ (zero-mean) and $\sigma^2_{Z_\eta}$ (variance).
- Original image: $g_0(x,y)$ (unbiased).
- Noise component: $\eta(x,y)$ (assumption: additively superimposed)
- Disturbed image: $g(x,y) = g_0(x,y) + \eta(x,y)$.

Filtering - **Image smoothing**

- Reducing noise effects by smoothing operators
- The goal is to reduce noise power by:
  - Local averaging (linear image operation/function)
  - Median filtering (non-linear image operation/function)
  - Low-pass filter and (frequency domain function, approximation through spatial domain function, e.g. Gaussian filter)
  - Ideal filters

Filtering by Image Smoothing - **Moving average**

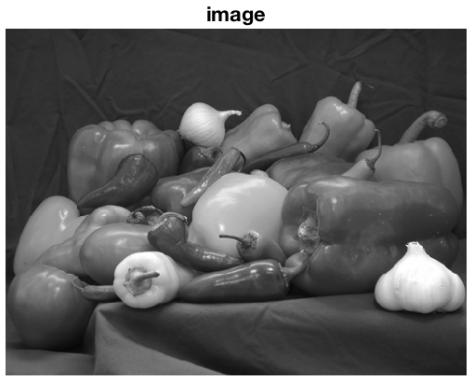
- Given. : $\mathrm{N}\times\mathrm{N}$ Input image $\mathrm{G} = (\mathrm{g}(\mathrm{x},\mathrm{y}))$ (gray value image) $\mathrm{m}\times\mathrm{n}$ neighborhood $N_{xy} = R_{xy}^{mn}$

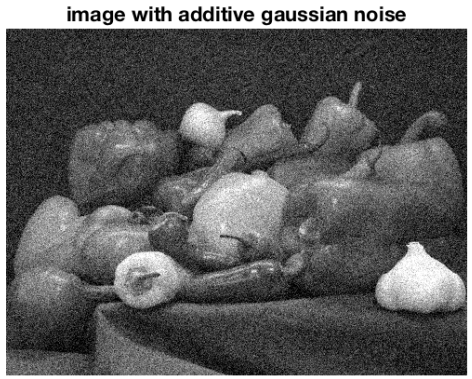
  the smoothed image F is obtained by applying the relation pixel by pixel:

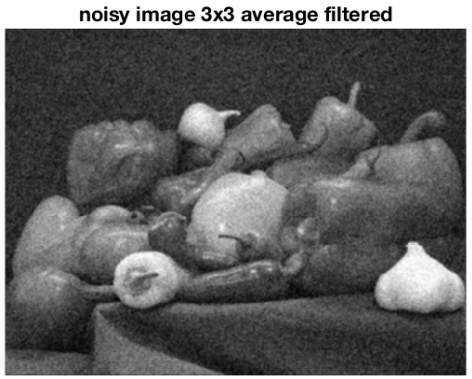
$$T_{GM} : f(x,y) = \frac{1}{m \cdot n} \sum_{(r,s)\in N_{xy}} g(r,s) \;\; \forall (x,y) \in \mathbf{G}$$

  and thus the smoothed output image $\mathrm{F} = (\mathrm{f}(\mathrm{x},\mathrm{y}))$.

Example: Averaging over N(x,y)







- **Averaging** results in **blurring**.
- The **amount of blurring** is proportional to the **neighborhood size** of the averaging filter kernel.

Example: Averaging over N(x,y)

- Reduction of blurring effect: effect can be reduced for a given neighborhood by applying a thresholding procedure.
- Aspect: Areas with **strong changes** in gray values should remain **unchanged**, because here original variations are expected: e.g. **edges**.
- Methods:

$$f(x,y) = \begin{cases} \frac{1}{n^2}\sum_{(r,s)\in N_{xy}} g(r,s) & \text{if } g(x,y) - \frac{1}{n^2}\sum_{(r,s)\in N_{xy}} g(r,s) < T \\ g(x,y) & \text{otherwise} \end{cases}$$

- T : non-negative threshold.

Filtering - Median filter I

- Simple **averaging** has two major **disadvantages**:
    1. Edges are blurred (blurring),
    2. Other (sharp) details are also blurred,
    3. Ringing effects occur.

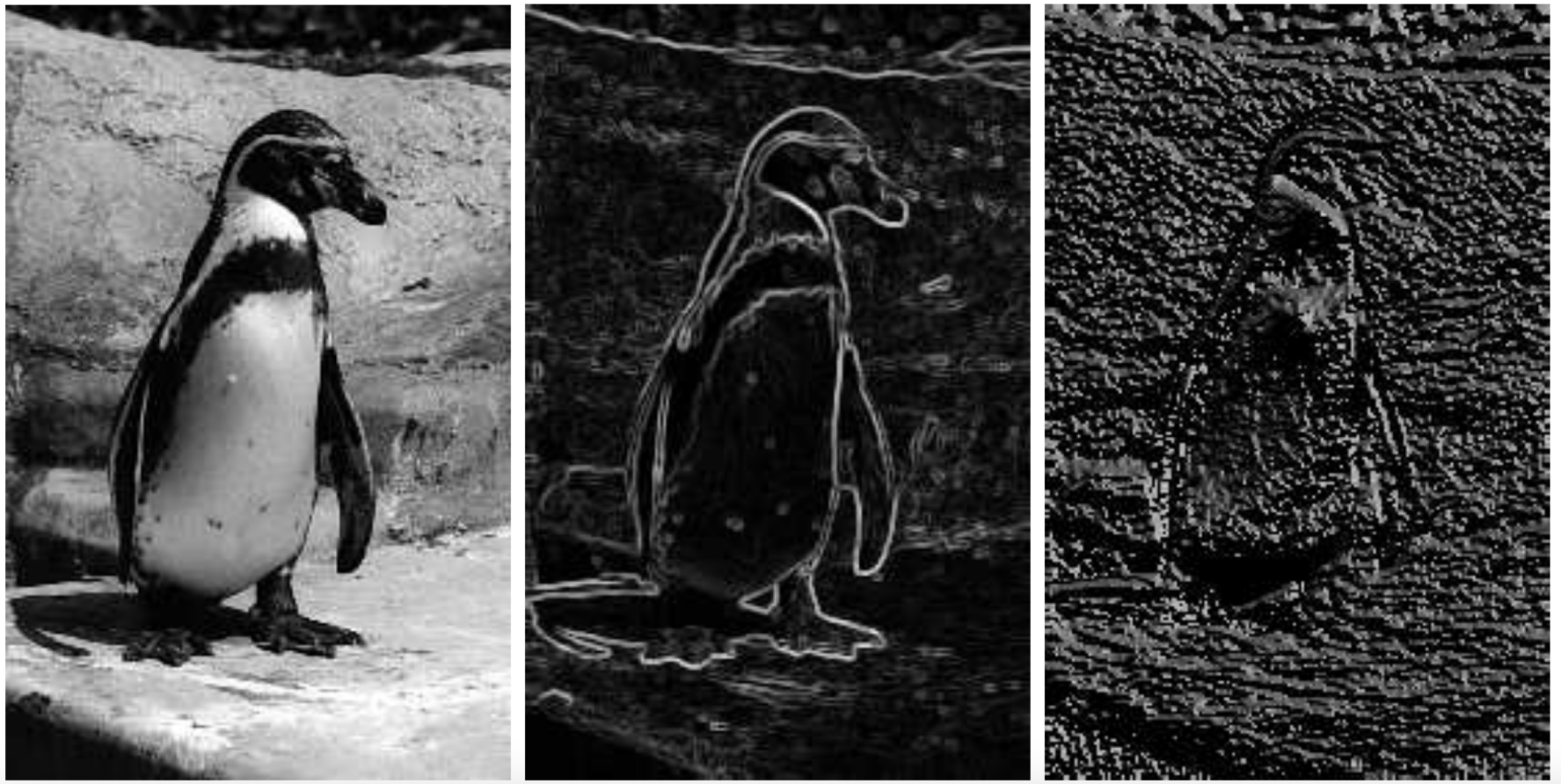
Filtering - Median filter II

- Figure : Image smoothing with median filter (median with 3 × 3 and 5 × 5 neighborhood, applied 5×).

Filtering - Median filter III

- Method: Each pixel is replaced by the median of the pixels *in its neighborhood.*
- Advantage: Method is especially effective for spikes and peaks and when edges are to be preserved (as a characteristic).

Image sharpening - Concept of image sharpening II



We need methods do detect ***edges*** in images

Edge Detection – the Image Gradient

- Gray image B(x,y): the analysis of the gray value gradient with the partial derivatives in x and y direction provides the gradient operators for edge extraction.
- Observation of the changes of the gray value gradient in x-direction (Bx) and y -direction (By) in the form of the partial derivatives:

$$B_x(x,y) = \frac{\delta}{\delta x}$$
$$B_y(x,y) = \frac{\delta}{\delta y}$$

- one thus obtains the components of the 2-D gradient:

$$\nabla B(x,y) = (B_x(x,y), B_y(x,y)).$$

- The gradient can be considered as a vector with length(norm) $|\nabla B(x,y)|$and direction $\varphi(\nabla B(x,y))$.

Differentiation in pictures

- The derivative f′(x)measures the rate of change of the function values of a function f(x).
- An edge represents a significant change in the function values (gray values) in the gray value progression of a gray value image B(x,y).
- The Euclidean norm can be used to calculate the length of the gradient:

$$G_E(B(x,y)) = \| \bigtriangledown B(x,y) \|_2 = \sqrt{B_x^2(x,y) + B_y^2(x,y)}\,.$$

Gradient operator I

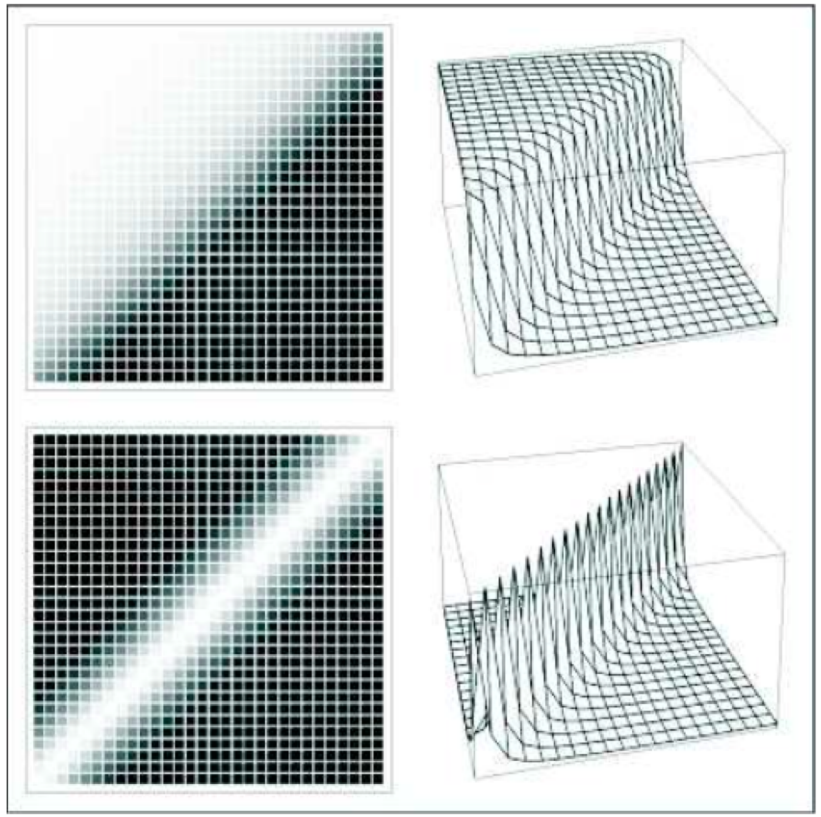


Image with an intensity step shown as 2D image and as surface(top) and gradient magintude (bottom)

Gradient operator II

- The gradient operator $G_E(B(x,y))$ provides a gradient magnitude image G_E (B).
- Thus, the gradient operator returns the maximum increase of the function area (gray values) at each point, so that large values indicate an edge.
- The gradient vector $\nabla B(x,y)$ **(Nabla operator)** points in the direction of the steepest slope in the function mountain,
- the value of the gradient magnitude corresponds to $\|\nabla B(x,y)\|_2$ (directional derivative, see Fig. 8).

Gradient operator III

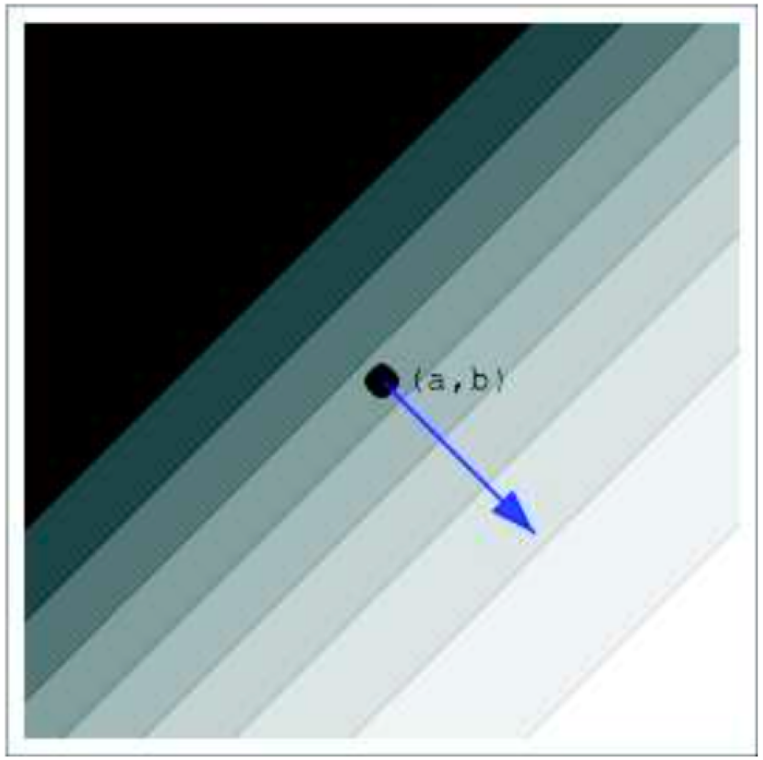


Figure : Gradient vector perpendicular to the edge direction

- A determination of edge points can then be achieved by evaluating the gradient operator.

Difference quotients I

- Since images represent discrete functions, the partial derivatives can be **approximated** by difference quotients.
- Different approaches can be used (pixel spacing d =Δx =Δy =1):

$$B_x(x,y) \approx \frac{B(x+d,y)-B(x,y)}{d}$$
$$B_y(x,y) \approx \frac{B(x,y+d)-B(x,y)}{d}$$

(forward difference)

$$B_x(x,y) \approx \frac{B(x,y)-B(x-d,y)}{d}$$
$$B_y(x,y) \approx \frac{B(x,y)-B(x,y-d)}{d}$$

(Backward difference)

$$B_x(x,y) \approx \frac{B(x+d,y)-B(x-d,y)}{d}$$
$$B_y(x,y) \approx \frac{B(x,y+d)-B(x,y-d)}{d}$$

(Central difference)

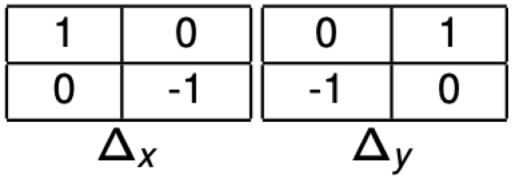
Discrete Gradient Operators I

- For the determination of edge points often **only** the **magnitude** G(B(x,y)) of the gradient is considered:
  - $G(x,y) = |\nabla(B(x,y)|$
  - i.e. we get a scalar output (a simple image) instead of a vectoral output

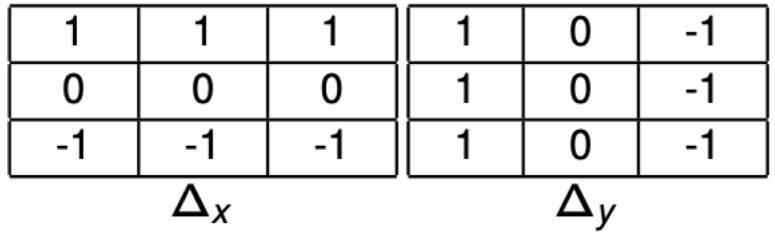
- For the discrete approximation of the gradient (-magnitude) one could use various difference functions, which can be represented efficiently with the help of the convolution operation.
  - Well known: **Roberts**, **Prewitt, Sobel** and **Kirsch** *gradient operators*
- **Everything becomes very simple now** ☺

Discrete Gradient Operators in 2D

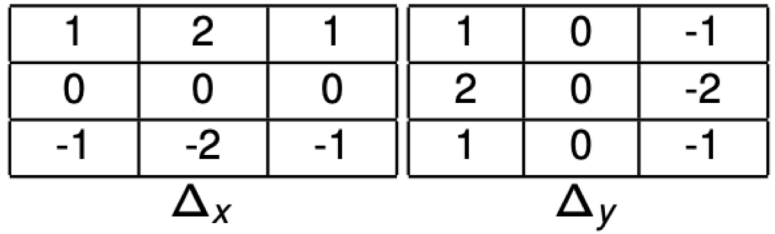
Filter Kernels:



Roberts Cross



Prewitt



Sobel

Rember: we want to compute an approximation of the differential quotient for each voxel!

Kirsch operator

- Determination of the **direction** of the maximum gradient
- Consideration of discrete directions: ”compass gradients”
  - e.g. Kirsch operator: eight different directions are defined,
- the direction φ of the maximum gradient $K_\varphi = \max\left(K_i\right)_{\forall i=0,\ldots 7}$ is selected.

Kirsch operator

$K_0 =$

-3	-3	5
-3	0	5
-3	-3	5

$K_1 =$

-3	5	5
-3	0	5
-3	-3	-3

$K_2 =$

5	5	5
-3	0	-3
-3	-3	-3

$K_3 =$

5	5	-3
5	0	-3
-3	-3	-3

$K_4 =$

5	-3	-3
5	0	5
5	-3	-3

$K_5 =$

-3	-3	-3
5	0	-3
5	5	-3

$K_6 =$

-3	-3	-3
-3	0	-3
5	5	5

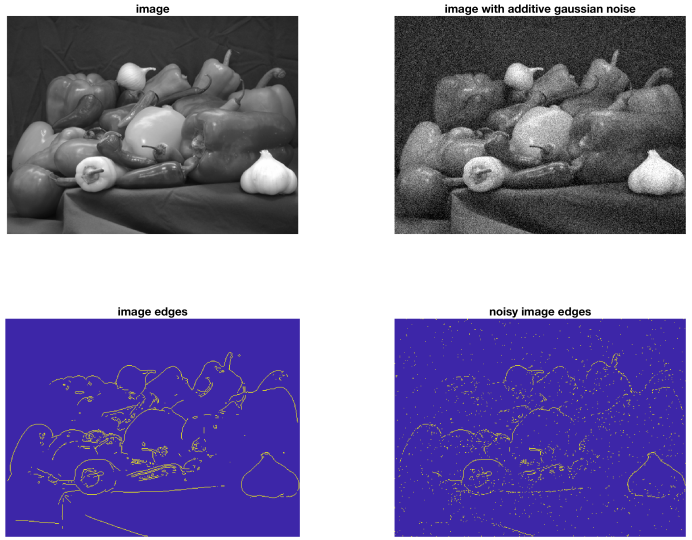
$K_7 =$

-3	-3	-3
-3	0	5
-3	5	5

We simply run **8** convolution filtering operations with the above filter kernels and select the maxium response

Notes on the gradient

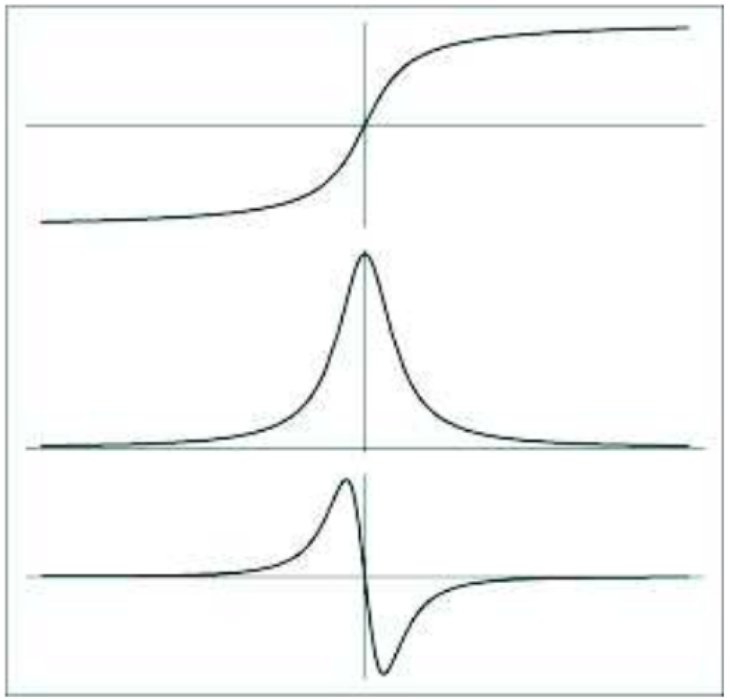
- Gradient operators are sensitive to function (gray level/intensity) changes, so they are also **sensitive to noise**!



- A *combination* of **image smoothing and gradient computation** can address this problem.

Edge detection with 2nd derivative I

- Relative maxima of the 1st derivative, which indicate **edge point**s are located at the **zero-crossings** of the **2nd derivative**.



- Figure : One-dimensional soft edge (top), 1st derivative (middle), 2nd derivative (bottom).

Edge detection with 2nd derivative III

- Zero of the 2nd derivative is, necessary not sufficient condition for a relative maximum of the 1st derivative.
- Example:

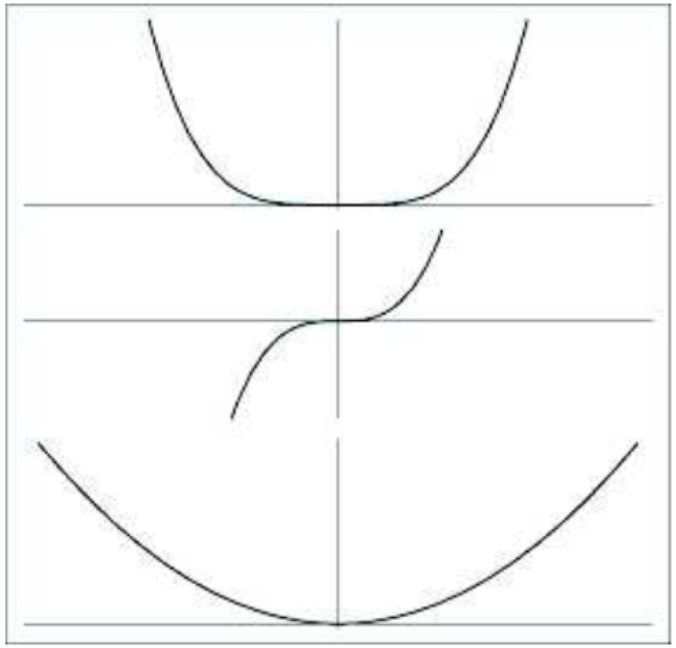


Figure : Example image without edge (top), 1st and 2nd derivative (with zero).

Edge detection with Laplace Op.

- Functions of two variables: Laplace operatorΔ as a direction-independent differential operator of 2nd order:

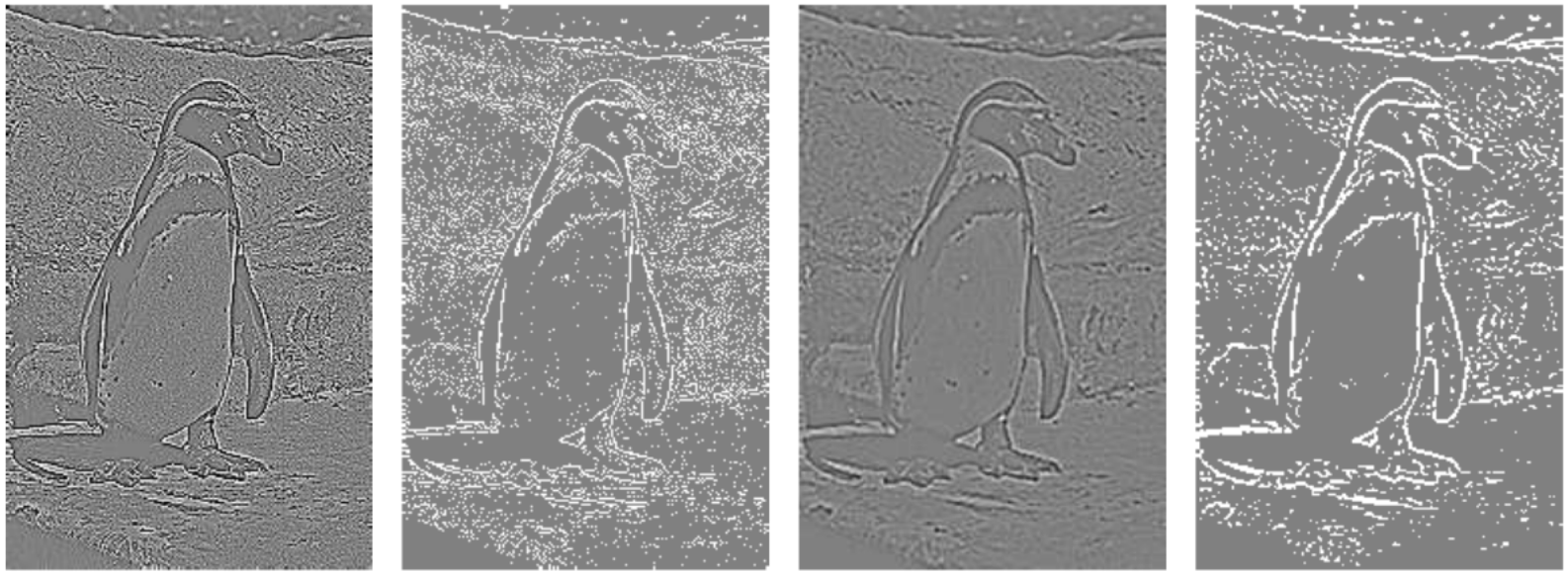
$$\Delta B(x,y) = \nabla^2 B(x,y) = \frac{\partial^2}{\partial x^2} B(x,y) + \frac{\partial^2}{\partial y^2} B(x,y).$$

- Discrete computation via convolution operator with the following kernel:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

- Zero crossings of the Laplace operator are candidates for edge points.

Laplace examples



Problems

- Real image data shows a **very strong sensitivity** to noise when usinig the laplace filter for edge detection.
- Idea: first use a smoothing step (e.g. low-pass or average filter) if possible.

Edge detection according to Marr and Hildreth I

- Method: Consideration in scientific literature
- Notes: Edge detection in the human visual system is based on a similar principle.
- similar principle.
  1. smoothing (by convolution) by the Gaussian function gσ
  2. combination with Laplace operator Δ;
  3. edge criterion (usually): Zeros of the Laplace operator:

$$\begin{aligned} g_\sigma(x,y) &= \frac{1}{2\pi\sigma^2}\exp\left(\frac{-(x^2+y^2)}{2\sigma^2}\right) \\ \Delta(g_\sigma * B) &= 0 \end{aligned} .$$

LoG Filter I

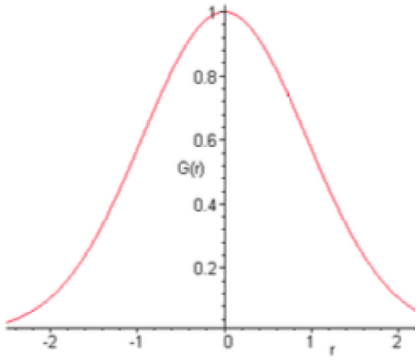
- "Laplace on Gaussian" (LoG filter, "Laplacian of Gaussian filter"):

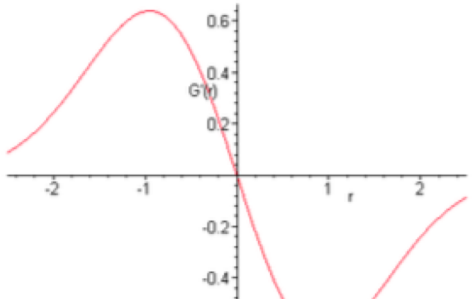
$$\Delta(g_\sigma * B) = \Delta(g_\sigma) * B\,.$$

- can be represented as a convolution with the kernel $\Delta(g_\sigma)$.
- The **operators** (*Gauss and Laplace*) are **linear**, so they can also be interchanged in order

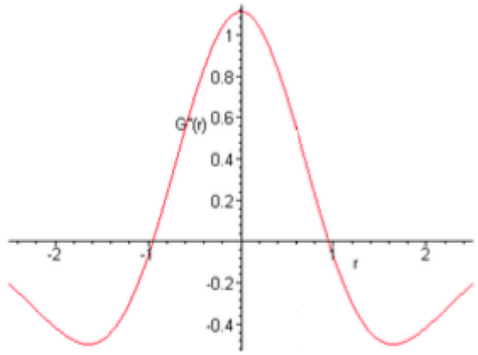
LOG-Operator: Laplacian of Gaussian II

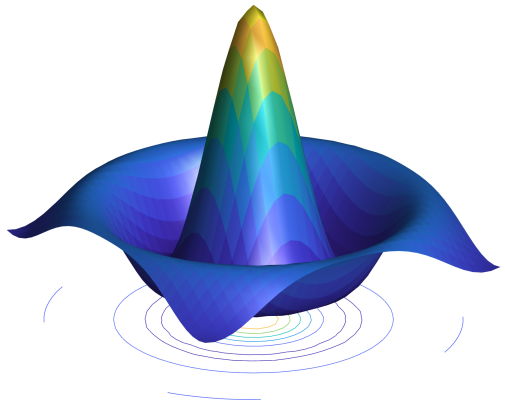
a)





c)





LOG, also known as **Marr-Hildreth** operator or the **Mexican Hat**

LoG Filter III

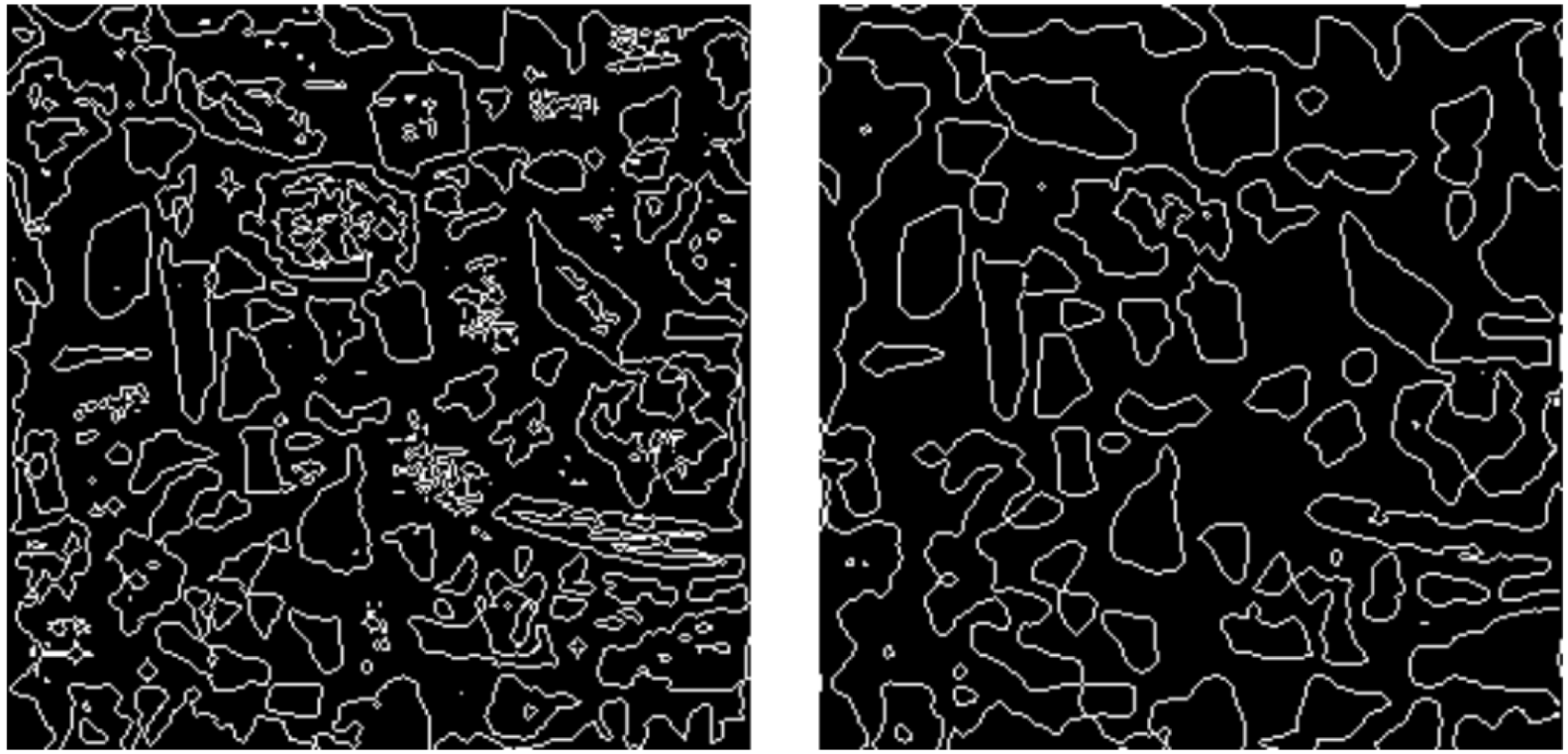


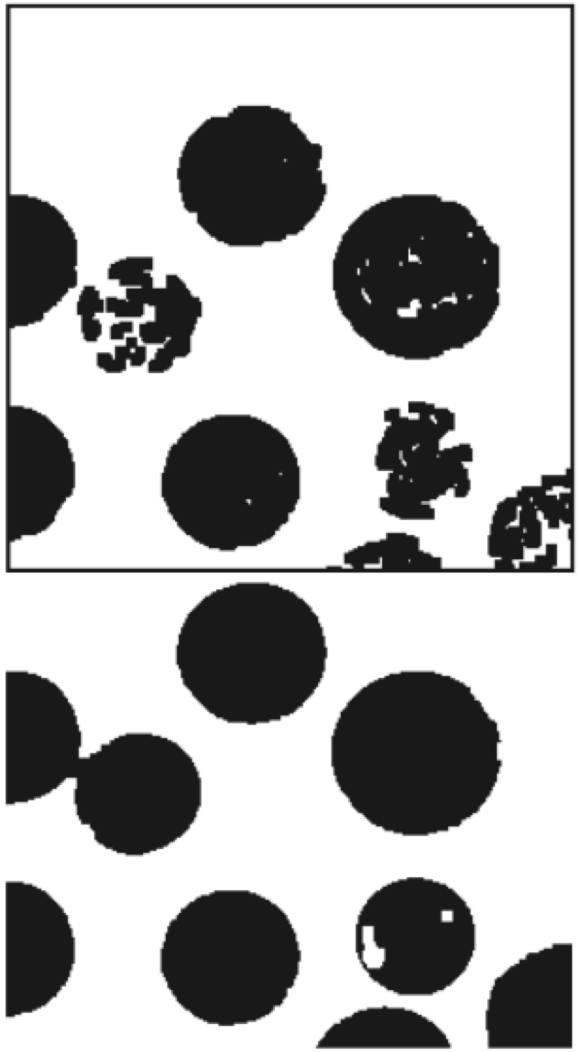
Figure : Edge detection with the Marr-Hildreth operator:
(a) weak smoothing (small sigma), (b) strong smoothing (large sigma).

Meaning from Binary Images

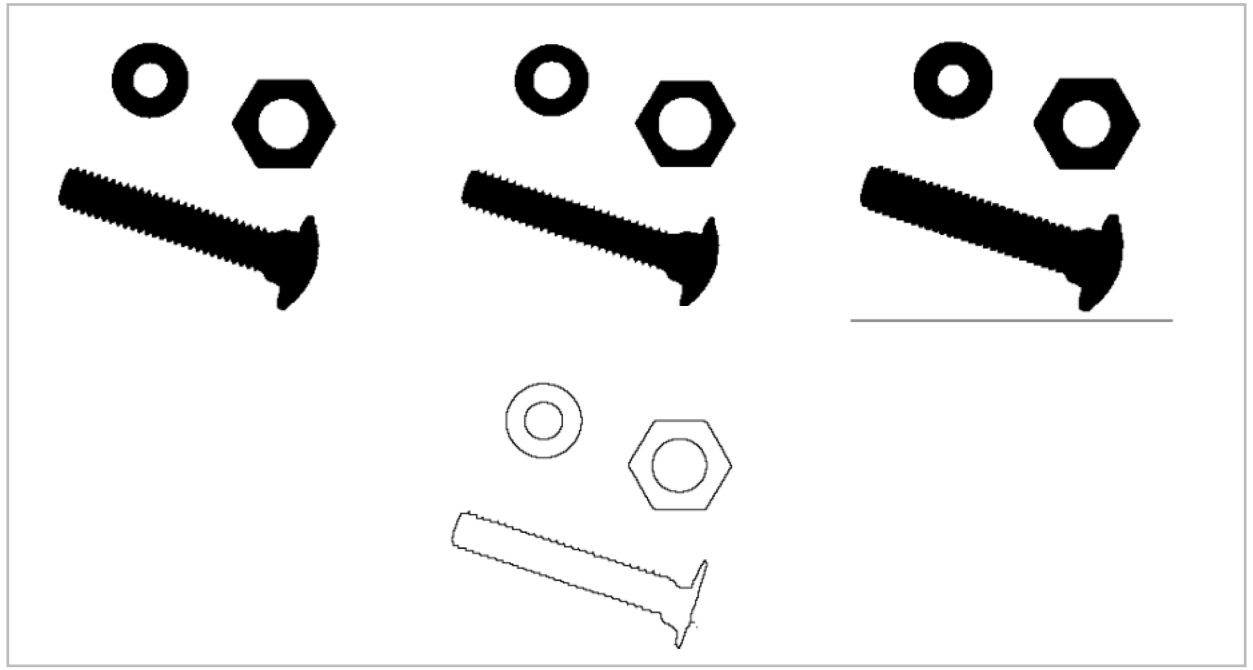
- Many image processing tasks involve the **computation of binary** images from color or grayscale data for further **inference**
- Assignments to the binary values of an image point, for example
  - 1. **"black and white"** image, i.e. the captured image has only two intensities
  - 2. **segmentation** image, where e.g. all "1" pixels represent object pixels and neighboring object pixels p,q (selectable neighborhood N(p)) belong to one object (context),
  - 3. **mask** image, where only the areas masked by "1" pixels are of interest for subsequent processing.

Binary image enhancement

- Elimination of disturbances
- Holes and concavities
- Singularities and convexities
- Smoothing of contours
- Determination of object contours
  Line thinning, skeleton lines

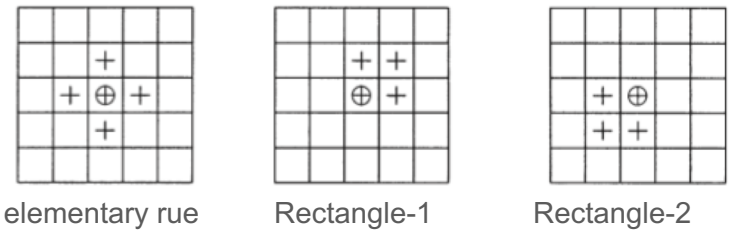


(Binary) Morphological Operators

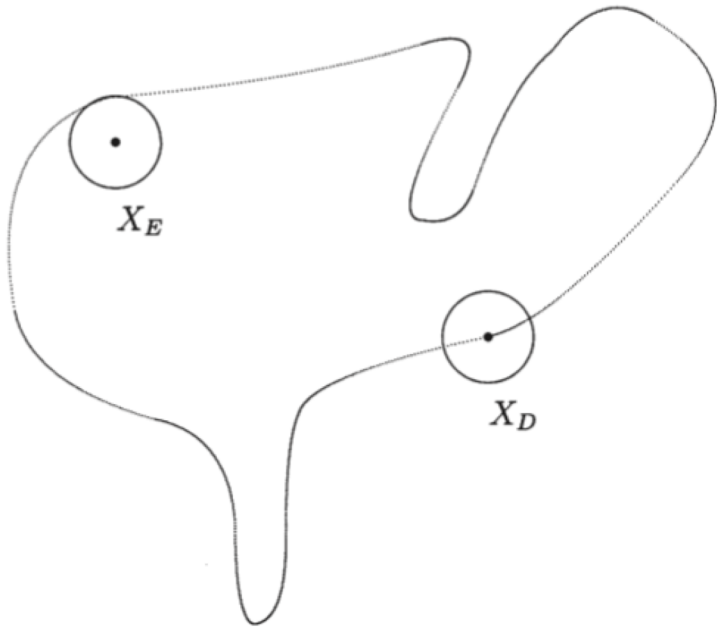


Dilation and **Erosion** I

- 1. “attaching to “(dilation) and “peeling off” (erosion) an object pixels by means of moving a **structural element**
- 2. basic operations for mathematical morphology (algebra)



elementary rue Rectangle-1 Rectangle-2



Dilation and Erosion II

- **Dilation**

$$d(i,j) := \begin{cases} 1 & \mid \; b(i,j) = 0 \wedge \exists (r,s) : (b(r,s) = 1 \wedge X^*(r,s) = 1) \\ b(i,j) & \text{otherwise} \quad (r,s) \in N_{X^*}(i,j) \end{cases}$$

- **Erosion**

$$e(i,j) := \begin{cases} 0 & \mid \; b(i,j) = 1 \wedge \exists (r,s) : (b(r,s) = 0 \wedge X(r,s) = 1) \\ b(i,j) & \text{otherwise} \quad (r,s) \in N_{X^*}(i,j) \end{cases}$$

- Compact description of erosion and dilatation

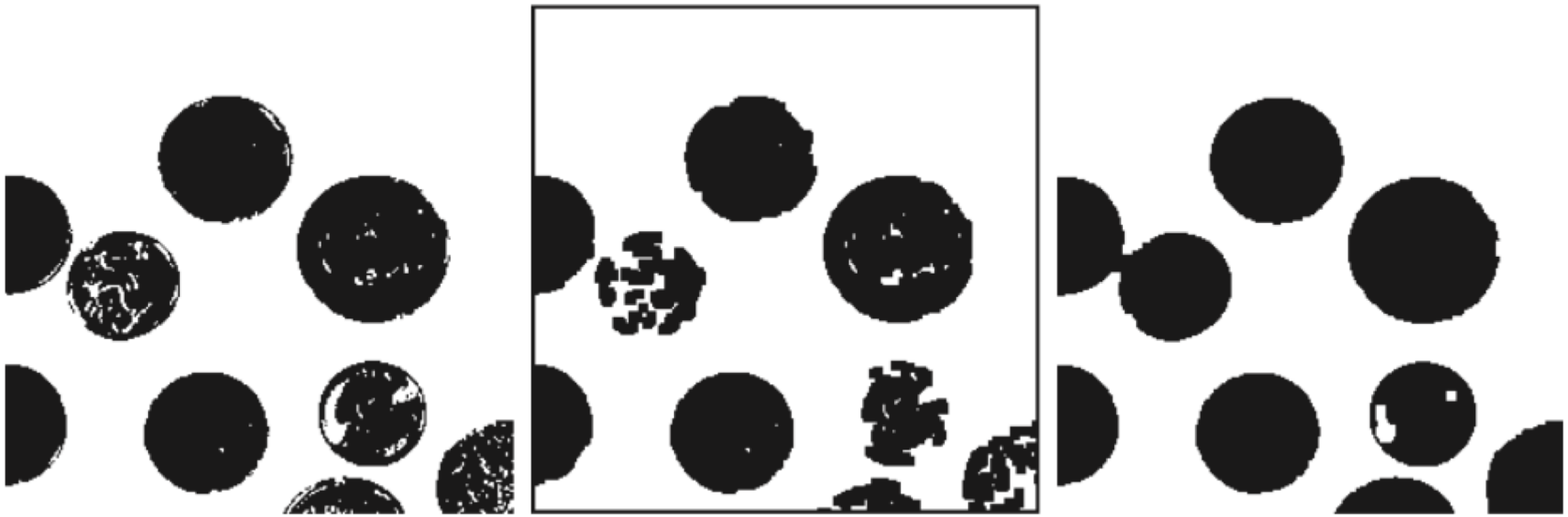
Dilation and Erosion III

- Dilatation and erosion are still dual:
- The dilation of the complement image is equal to the complement image of the eroded image!

$$\overline{B} \oplus X = \overline{B \ominus X}$$

Open and close I

- **Deleting** singularities or **separating** objects
- **Closing holes** or **connecting** objects



Open and close II

- Open:

$$o = b \circ X$$
$$o = ((b \ominus X)_{i-\text{ times}} \oplus X)_{i-\text{times}}$$

- Close:

$$s = b \bullet X$$
$$s = ((b \oplus X)_{i-\text{times}} \ominus X)_{i-\text{ times}}$$

Gray value morphology

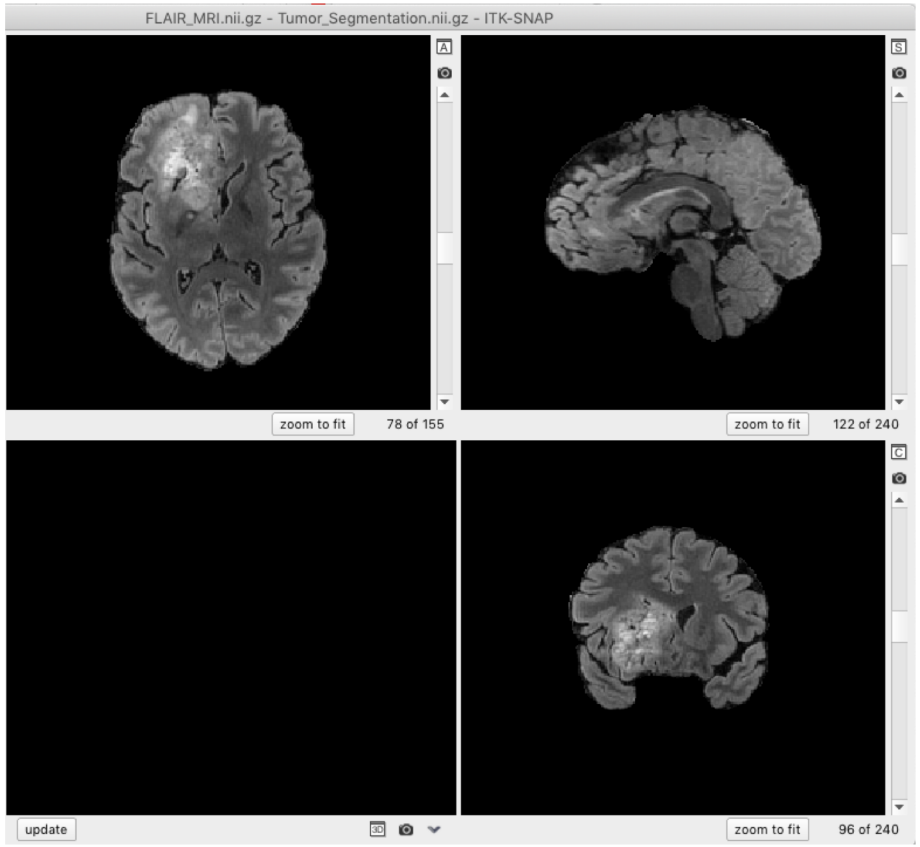
- Generalization of dilation and erosion.
- Morphological operations $\oplus, \ominus$ on gray-scale images, e.g.
  - Smoothing or sharpening
- Determination of object contours

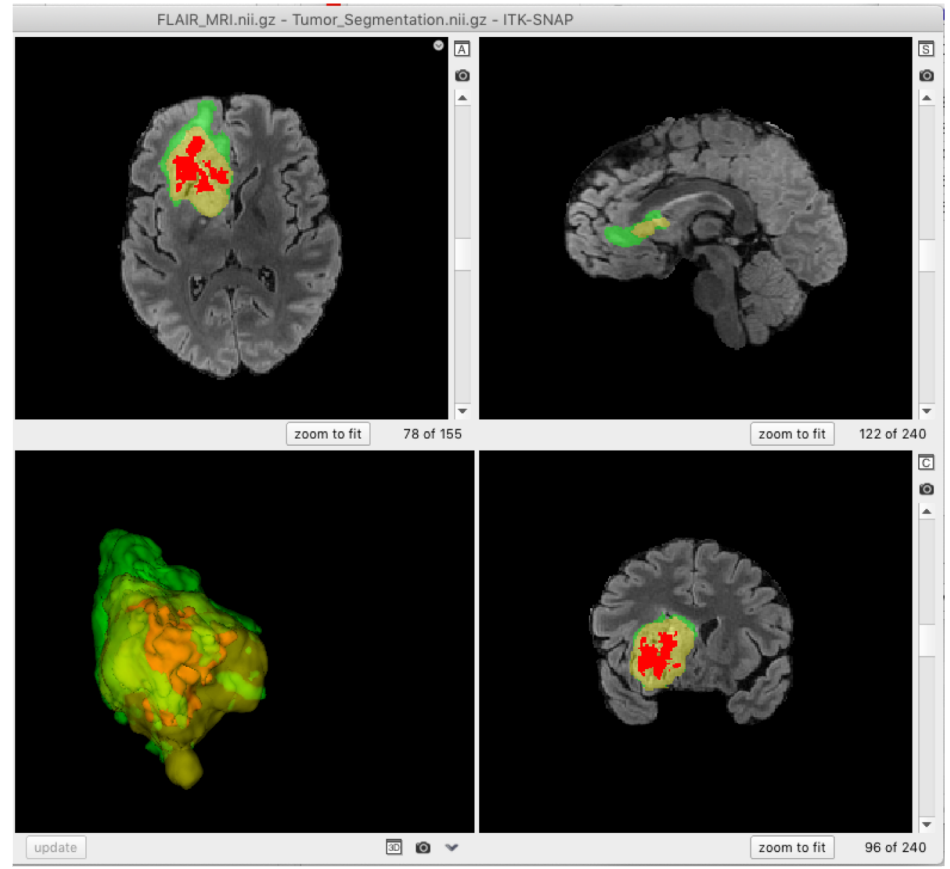
$$
\begin{aligned}
e &= b \ominus X \\
e(i,j) &= \min_{\forall (r,s) \in N(i,j)} (b(r,s) - X(r,s)) \\
d &= b \oplus X \\
d(i,j) &= \max_{\forall (r,s) \in N(i,j)} (b(r,s) + X^*(r,s))
\end{aligned}
$$

Edge extraction

- **Edge extraction** using image smoothing and **erosion / dilation.**

Image **Segmentation**





Segmentation: Paradigms

- The **goal of segmentation**: to **divide an image into meaningful sub-areas to be classified as objects or background**.
  Segmentation methods are based on two complementary concepts:

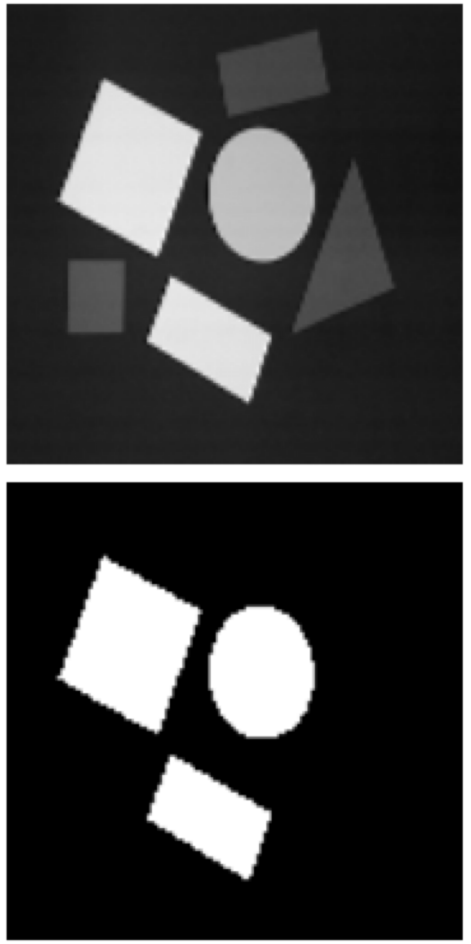
  1. objects (signatures) are characterized by a homogeneity H(g) (common property such as gray level, color or texture similarity),

  2. local signal changes $\Delta(g)$ (gray values, etc.) are indications of object boundaries.

- Methods / Paradigms:
  - Search for contiguous homogeneous areas or
  - Look for the contours(edges) that delimit homogeneous areas.
- We consider both segmentation paradigms: first **area-oriented** methods and then **edge-oriented**.

Area oriented methods

- The goal of area-oriented image segmentation is to divide the image plane into regions that correspond to the objects of interest or the background.
- The objects are to be characterized by a uniformity criterion, e.g. that there is object-specific homogeneity.
- The segmentation is called complete because the regions found are contiguous and cover the whole image (i.e. the segmentation map is a *partition*).



Threshold operators

- For each pixel: Assignment to an object or background is made solely by comparing its gray value to a threshold θ: thresholding.
- The threshold θ can be fixed/constant or (more rarely) dynamically adjusted.
- Multiple thresholds θ_i,i=1,2,...4 can also be used: more than two segmented classes possible.
- Threshold operators can be implemented very efficiently; because they are simple and effective, they are among the most important tools in image analysis.

Simple thresholding methods I

- Let the input image be the gray image function, $B: \mathscr{Z} \times \mathscr{Z} \rightarrow G(x, y \in \mathscr{Z})$, with G={0,1,2,...,$g_{max}$}(e.g. g_{max} =255).
- Binary output image S(x,y)=T_s(B(x,y)) to threshold s ∈G:

$$S(x,y) = T_s(B(x,y)) = \begin{cases} 1 & \text{for } B(x,y) \geq s, \\ 0 & \text{for } B(x,y) < s \end{cases}$$

- In image processing systems, a 2-level image is often generated:

$$\tilde{S}(x,y) = \tilde{T}_s(B(x,y)) = \begin{cases} g_{max} & \text{for } B(x,y) \geq s, \\ 0 & \text{for } B(x,y) < s \end{cases}$$

Simple Threshold Methods II

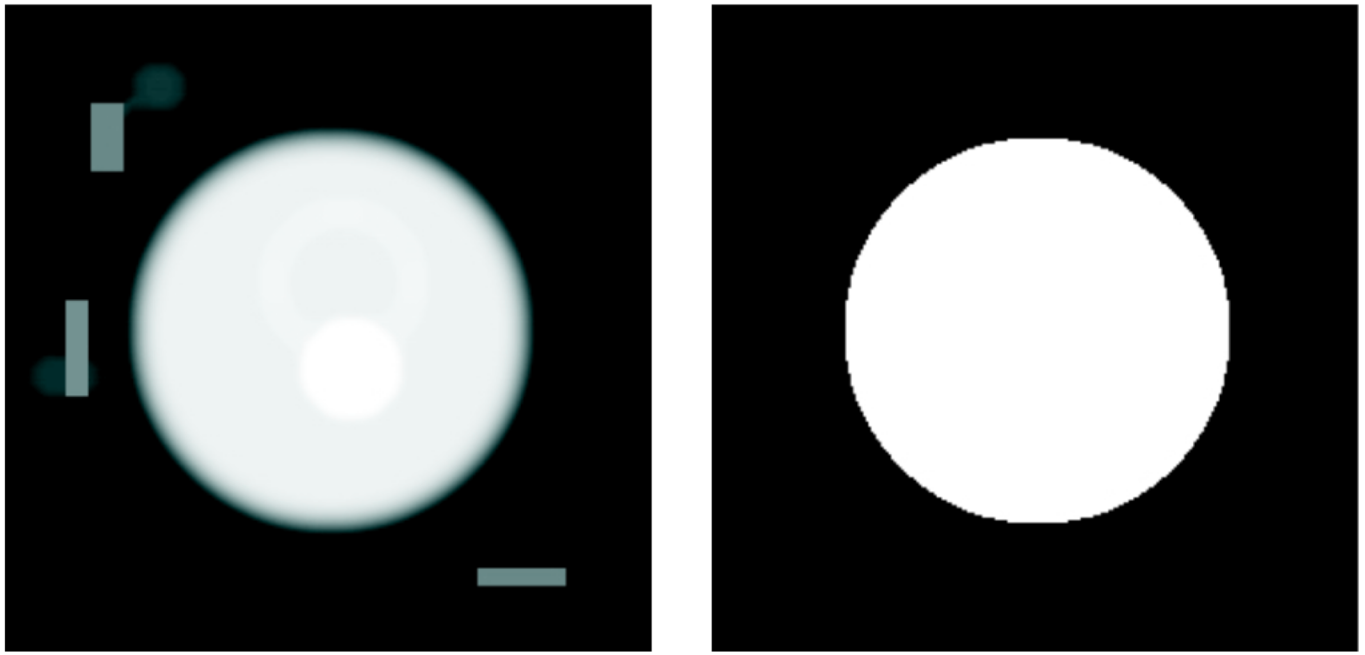
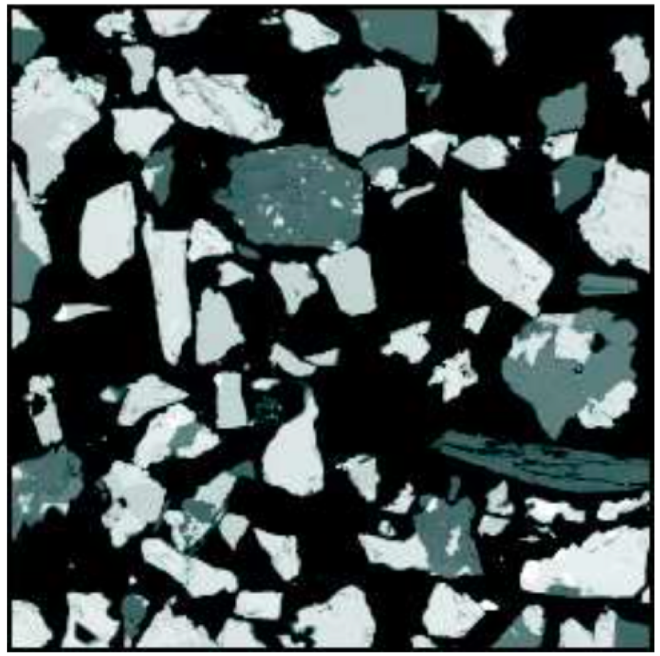
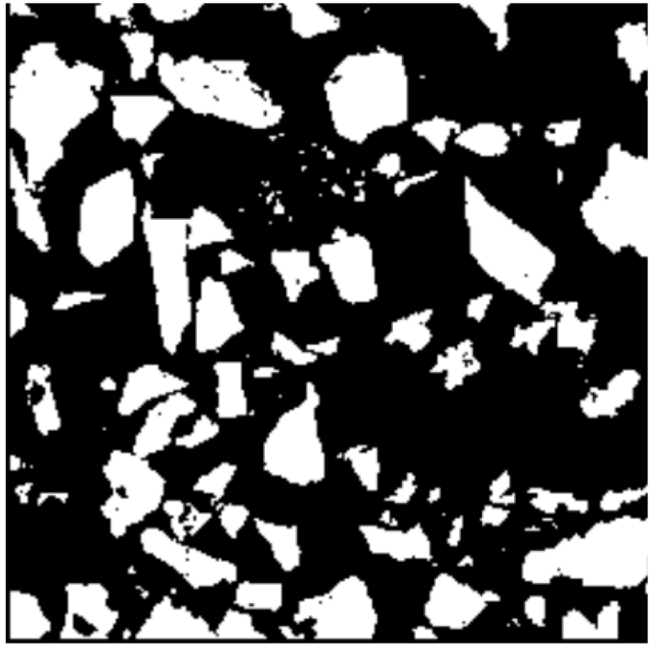


Figure : Threshold operator with a fixed threshold ($\theta = 130$, gray values large circle $g \simeq 230$).

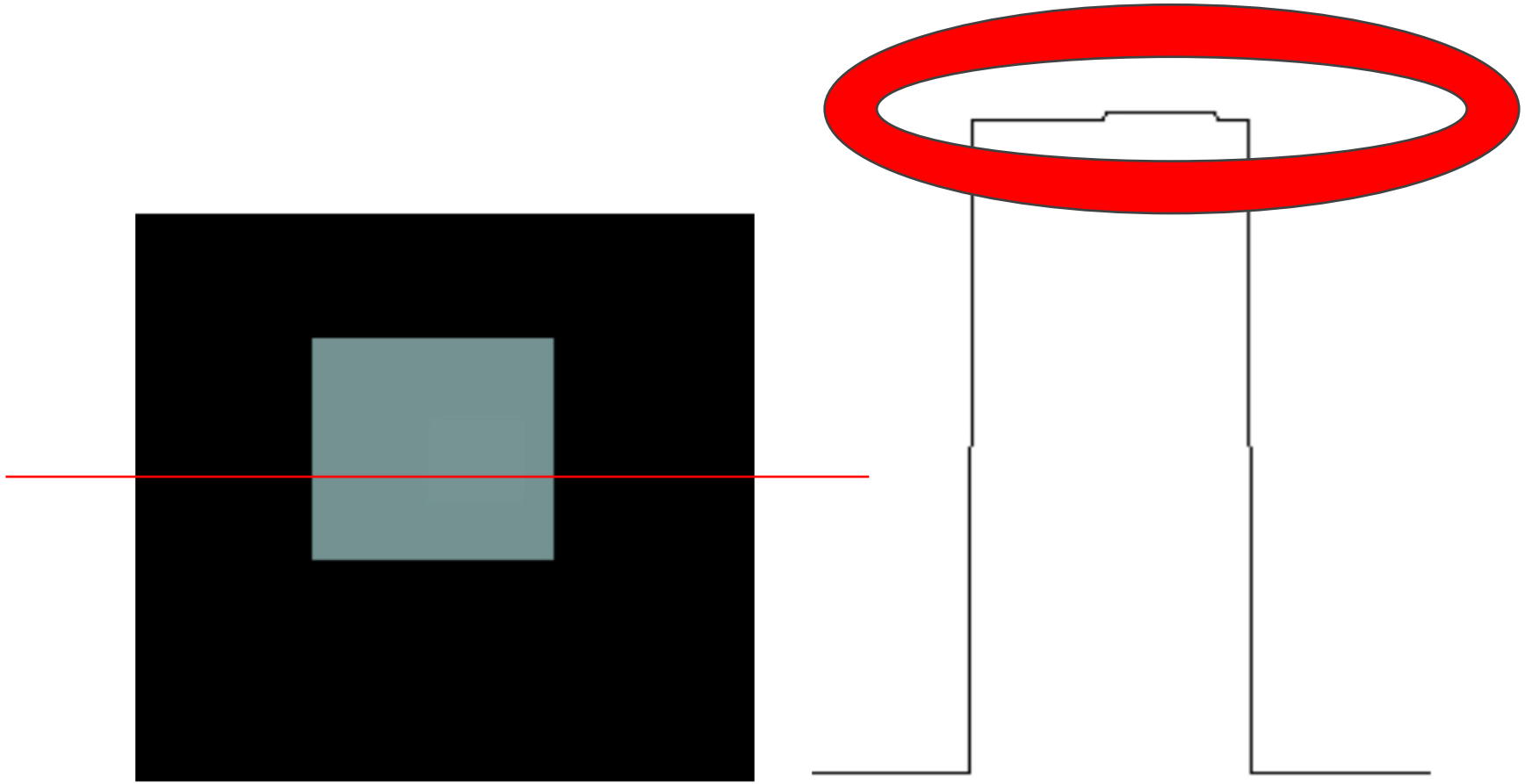
Simple Threshold Methods III

- Practical problems: Inhomogeneity of gray values results in blurred object borders and holes.
- Often post-processing required (for example by using morphological operators).





Threshold I choice



Choice of threshold II

- The **choice of threshold** is an **essential problem** in the application of the threshold operator.

- Example image B(x,y) in Figure B has a square with constant gray value 120 on a black background and another smaller square with constant gray value 121 inside it

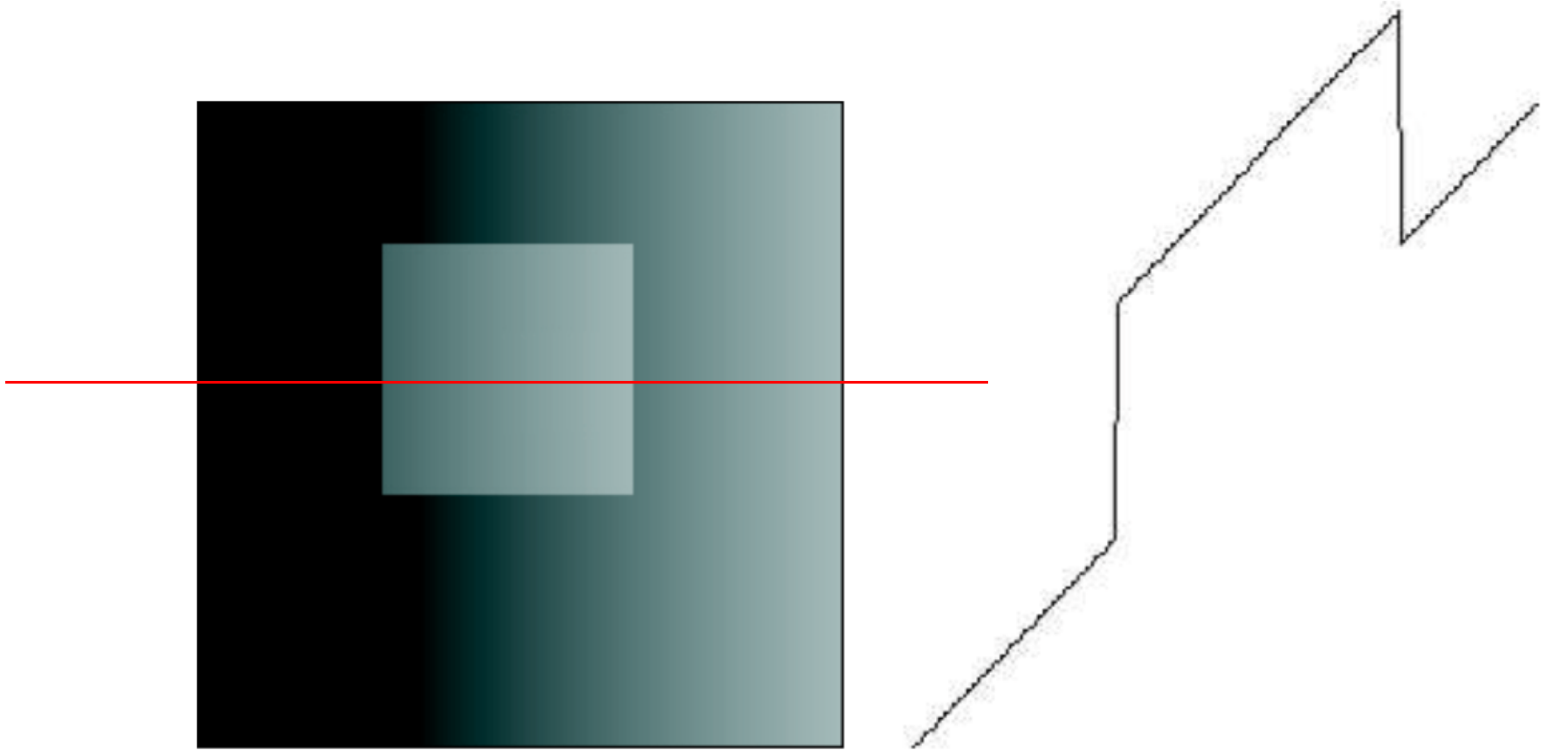
Choice of threshold III

- Figure : Threshold operator for: a) s = 120 and b) s = 121.
- A "small" change in the threshold causes a "large" change in the result image ("ill posed problem").

Choice of threshold IV

- Likewise: a "small" change in the input image can have a significant effect in the result.
  - Increasing the gray values of the image B(x,y)by the value 1: Bˆ(x,y) = B(x,y)+1, (no visible difference).
  - The threshold operator now returns a result image with S3(x,y)=T121(Bˆ(x,y))in which the large square has been segmented again.
- This makes segmentation "difficult" and it is important to **use additional information**.
- In real image data, **noise is always present**, which complicates the segmentation.
- Further difficulties in threshold segmentation arise from the superimposition of a background signal on the image data.

Example: Superposition II



- Figure : a) test image with superimposed gray wedge, b) line gray value profile. => fixed threshold segmentation not possible anymore!

Example: Superposition II

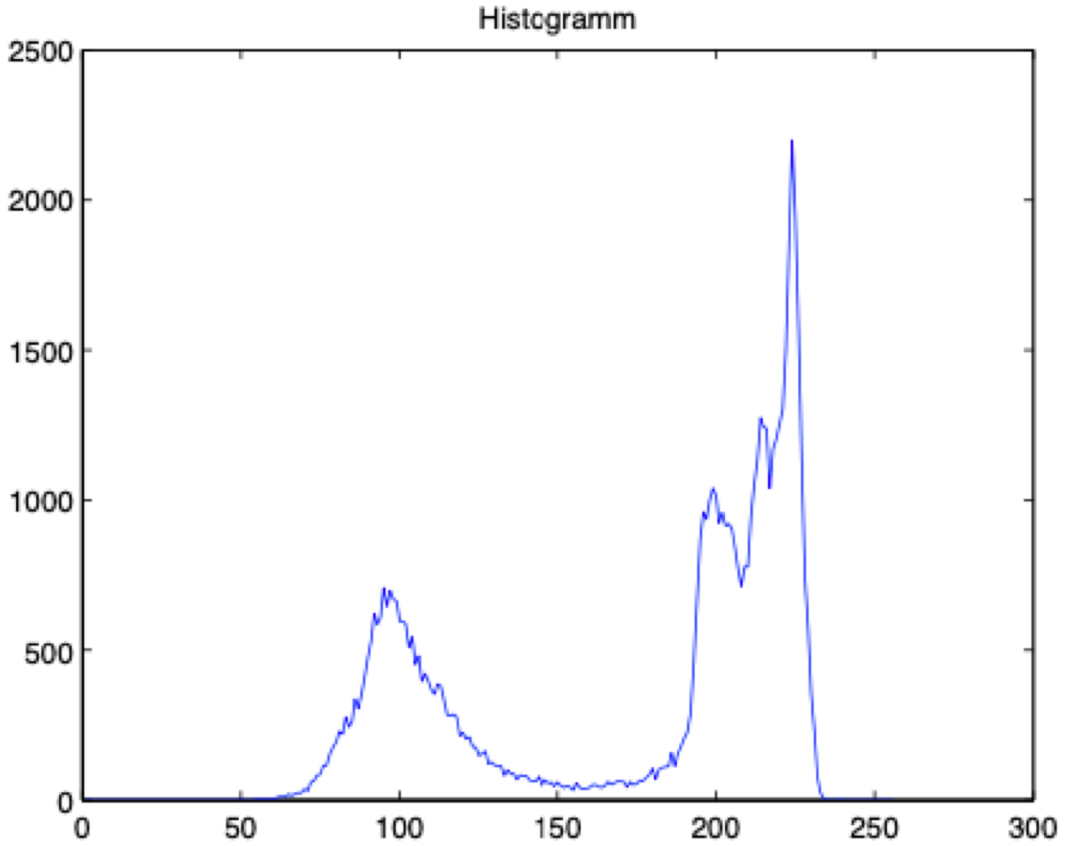
- Figure : a) test pattern and b) result threshold operator for s = 120

Threshold determination

- In realy image processing systems often **interactive** threshold selection
  - The setting of a suitable threshold is done manually by "trial and error".
- For the **systematic** determination of threshold values, one usually looks at the the **gray scale histogram**.
- For images with "light" objects on a dark background, the histogram shows two distinct maxima (bimodal).
- A suitable threshold is now sensibly placed exactly between the two maxima.

Automatic threshold determination

- For example, one can use a statistically motivated approach (Otsu method).
- With a bimodal histogram, one determines a value s that maximizes the interclass variance.
- This can be understood as an "optimal" separation of the two maxima in the histogram.



Threshold value determination according to Otsu *(extra slides for self study/reference)*

- Let pi be the probability for the occurrence of the gray value g ∈ G.
- If the two classes C_1=[0,s) and C_2=[s,g_{max}] are,then the probabilities for the occurrence of the classes P(C_1) and P(C_2) as well as the associated mean values $\overline{g}_1(s), \overline{g}_2(s)$ can be determined.
- With the total mean value , $\overline{g}$ the following formula for the variance of the two classes is obtained:

$$V_1(s) = P(C_1)(\overline{g}_1(s) - \overline{g})^2 + P(C_2)(\overline{g}_2(s) - \overline{g})^2.$$

- The determination of the maximum is done with the (known) methods of differential calculus.

Adaptive and local thresholding methods

- For objects on inhomogeneous background, the following method can be applied.
- Original image B is first lowpass filtered: $\tilde{B}$, (background signal).
- Subtraction B- $\tilde{B}$, eliminates the inhomogeneous background
- approximately;
- then a thresholding procedure is performed.
- The segmentation is thus adaptive to some extent with respect to the low-frequency background.
- Test image from Figure 6 can be adaptively segmented exactly (with strong smoothing: mean filter) with window size 137.
- An alternative approach is to divide the image into regions.
- For each region, an adapted threshold is determined. This local threshold can be determined from the local histogram, for example.

Region growing based method

- 1. **starting from one image point** *(seed)* of a homogeneous region further pixels are **successively added.**
- 2. in each step all neighbor pixels of the already assigned region pixels are examined.
- 3. If such a neighbor pixel fulfills a homogeneity criterion H, it is added to the region **(cf. thresholding!).**
- 4. a complete segmentation of the image into *N* non-overlapping regions requires a seed point for **each** region.
- 5. the growth process ends when all image points have been assigned to one region.

Growth procedure for a region I

- 1. definition of a neighborhood relation Z (mostly 8-neighborhood (N_8) or 4-neighborhood (N_4)).
- Example: image matrix B with marked starting point, relation Z: 8-neighborhood .
- 2. a neighbor pixel is added exactly if it has the gray value of the start point: $\mathrm{H} : \mathrm{g}(\mathrm{r},\mathrm{s}) \in 6,\ (\mathrm{r},\mathrm{s}) \in \mathrm{N}_8(\mathrm{x},\mathrm{y})$.

$$B = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 3 & 1 & 0 \\ 0 & 1 & 6 & 7 & 0 & 3 & 6 & 6 \\ 7 & 1 & 6 & 6 & 2 & 3 & 1 & 0 \\ 0 & 7 & 6 & 6 & 6 & 4 & 1 & 0 \\ 0 & 1 & 6 & \underline{\mathbf{6}} & 6 & 6 & 1 & 0 \\ 0 & 1 & 6 & 6 & 6 & 6 & 6 & 0 \\ 6 & 4 & 2 & 7 & 0 & 3 & 1 & 7 \\ 0 & 1 & 2 & 1 & 0 & 3 & 1 & 0 \end{bmatrix}$$

Growth procedure for a region II

- **First iteration**: eight neighboring pixels are added and marked

$$B_1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 3 & 1 & 0 \\ 0 & 1 & 6 & 7 & 0 & 3 & 6 & 6 \\ 7 & 1 & 6 & 6 & 2 & 3 & 1 & 0 \\ 0 & 7 & \mathbf{6} & \mathbf{6} & \mathbf{6} & 4 & 1 & 0 \\ 0 & 1 & \mathbf{6} & \underline{\mathbf{6}} & \mathbf{6} & 6 & 1 & 0 \\ 0 & 1 & \mathbf{6} & \mathbf{6} & \mathbf{6} & 6 & 6 & 0 \\ 6 & 4 & 2 & 7 & 0 & 3 & 1 & 7 \\ 0 & 1 & 2 & 1 & 0 & 3 & 1 & 0 \end{bmatrix}$$

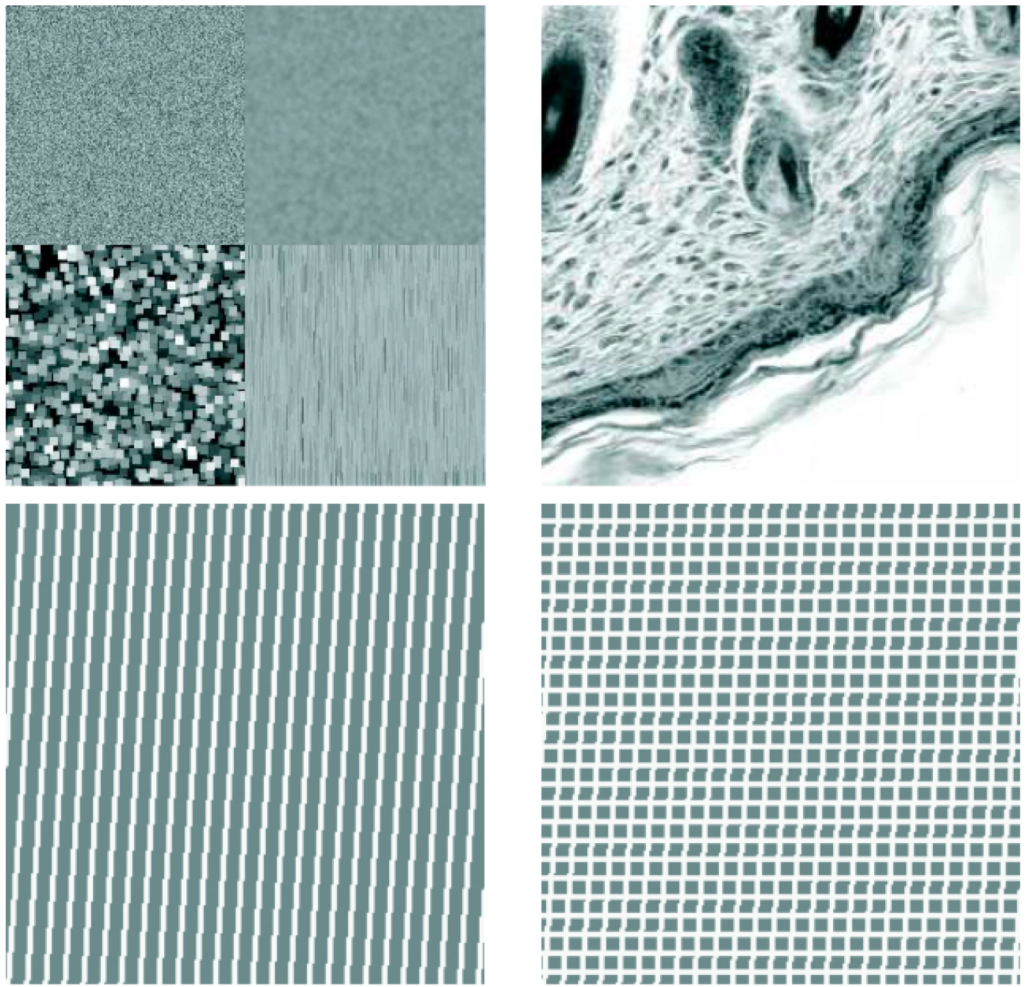
Growth procedure for a region III

- **Second iteration:** grow the region around all neighboring pixels with gray value 6 (image B2)

- Termination: The process ends when no more pixels can be added.

$$B_2 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 3 & 1 & 0 \\ 0 & 1 & 6 & 7 & 0 & 3 & 6 & 6 \\ 7 & 1 & \mathbf{6} & \mathbf{6} & 2 & 3 & 1 & 0 \\ 0 & 7 & \mathbf{6} & \mathbf{6} & \mathbf{6} & 4 & 1 & 0 \\ 0 & 1 & \mathbf{6} & \underline{\mathbf{6}} & \mathbf{6} & \mathbf{6} & 1 & 0 \\ 0 & 1 & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & 6 & 0 \\ 6 & 4 & 2 & 7 & 0 & 3 & 1 & 7 \\ 0 & 1 & 2 & 1 & 0 & 3 & 1 & 0 \end{bmatrix} \quad B_3 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 3 & 1 & 0 \\ 0 & 1 & \mathbf{6} & 7 & 0 & 3 & 6 & 6 \\ 7 & 1 & \mathbf{6} & \mathbf{6} & 2 & 3 & 1 & 0 \\ 0 & 7 & \mathbf{6} & \mathbf{6} & \mathbf{6} & 4 & 1 & 0 \\ 0 & 1 & \mathbf{6} & \underline{\mathbf{6}} & \mathbf{6} & \mathbf{6} & 1 & 0 \\ 0 & 1 & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & \mathbf{6} & 0 \\ 6 & 4 & 2 & 7 & 0 & 3 & 1 & 7 \\ 0 & 1 & 2 & 1 & 0 & 3 & 1 & 0 \end{bmatrix}$$

Texture segmentation I



Texture examples; left: periodic and stochastic structures, right: real image with texture (microscopy)

Texture examples (periodic structures)

Texture Segmentation II

- Objects in images often have a characteristic gray value or surface **structure** patterns, th: texture.
- Texture: intuitive term; no exact definition possible.
- Typical for textures: periodicity (actually: more or less regularity) of basic patterns or stochastic elements.
- Texture is a local property in the image.

  Texture analysis is local: window size or resolution scale play an important role.
- Here: consideration of methods based on statistical models (2nd order statistics).

End Lecture 3