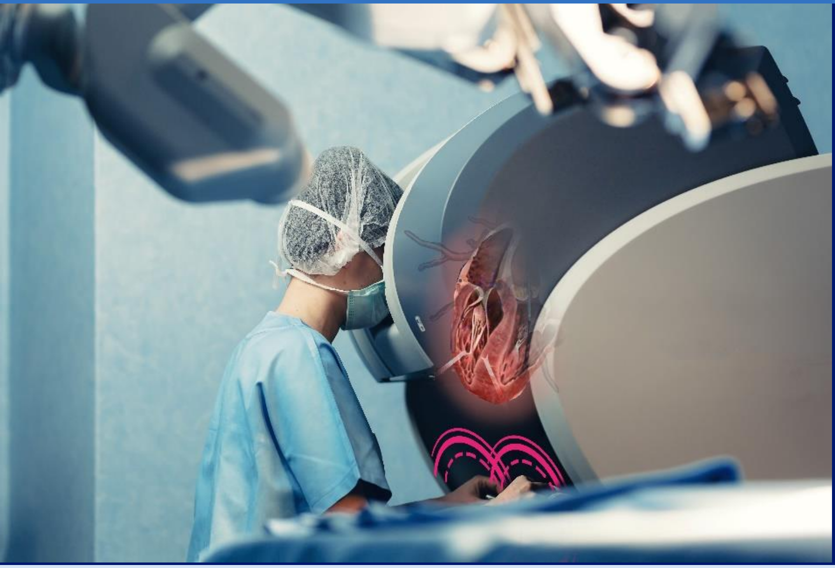
Computer- and robot-assisted Surgery



NCT
NATIONALES CENTRUM
FÜR TUMORERKRANKUNGEN
PARTNERSTANDORT DRESDEN
UNIVERSITÄTS KREBSCENTRUM UCC

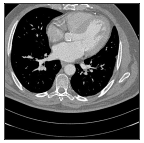
getragen von:
Deutsches Krebsforschungszentrum
Universitätsklinikum Carl Gustav Carus Dresden
Medizinische Fakultät Carl Gustav Carus, TU Dresden
Helmholtz-Zentrum Dresden-Rossendorf

Lecture 2

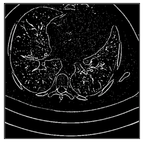
Basics of Computer Vision – Part 1

Process chain computer-assisted surgery

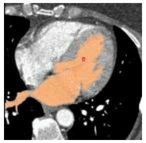
Image acquisition



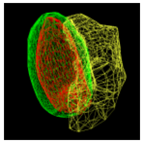
Pre-processing



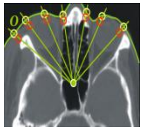
Segmentation



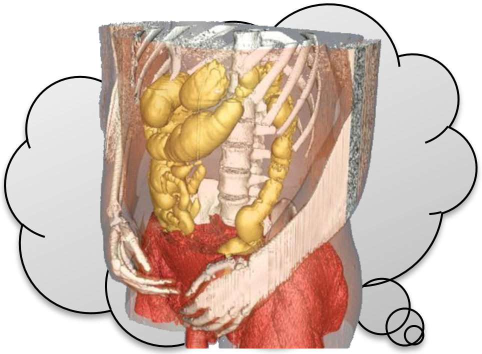
Modeling

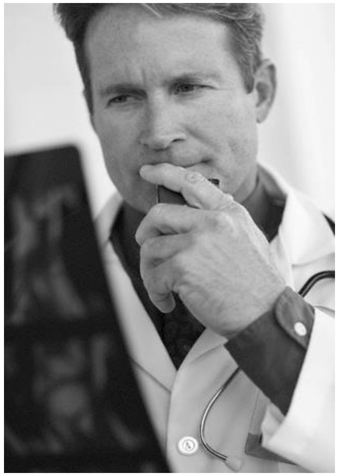


Registration



Visualization/
Assistance





Interaction and Feedback

- https://pingo.coactum.de -> 392473

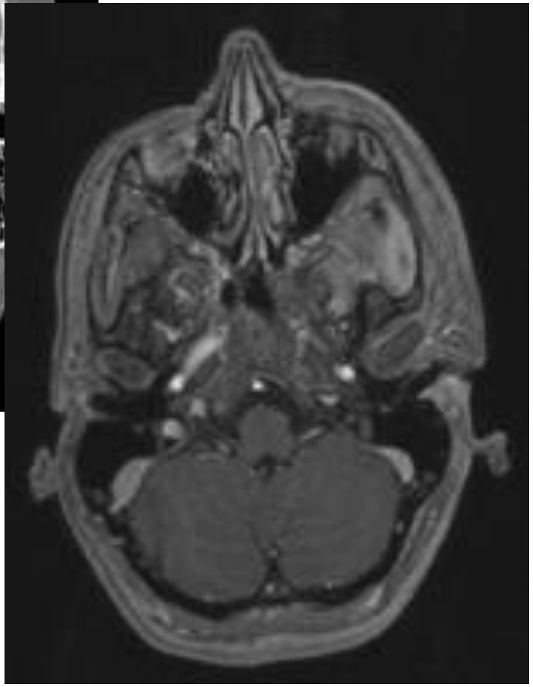
Contents

How can the quality of images be improved and relevant features be extracted for further processing steps?

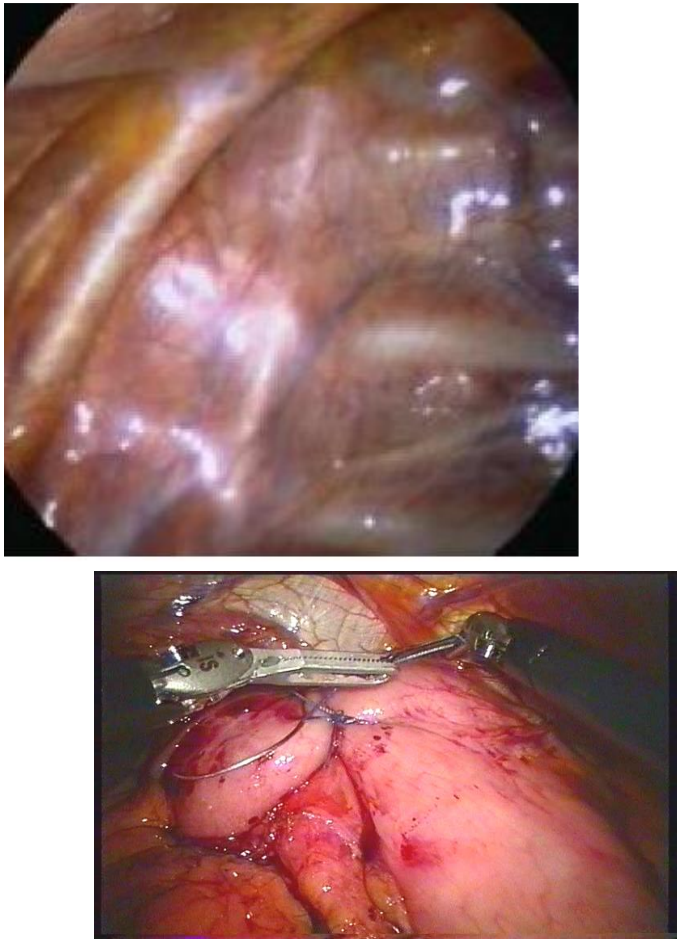
- Characteristics of images
- Point operations
- Local operations
  - Smoothing filters
  - Edge filters
  - Morphological operators

Image to be processed

CT



MRT



Endoscope

Endoscope

- Endoscope = tube- or pipe-shaped instrument that uses an optical system to provide images from the inside of the body
- Usage in diagnostics and surgery: Flexible and rigid endosopes
- Pros: Gentle surgery with low risks, minimal costs and quick recovery

Quelle: Storz

Minimally-invasive surgery


Quelle: Universität Heidelberg

- Complex interventions with complex anatomy
- Difficulties:

  Restricted field of view, reduced mobility, difficult hand-eye-coordination, discrepancies between enlarged 2D view and actual 3D environment, reduced tactile feedback

→ Requires high level of dexterity

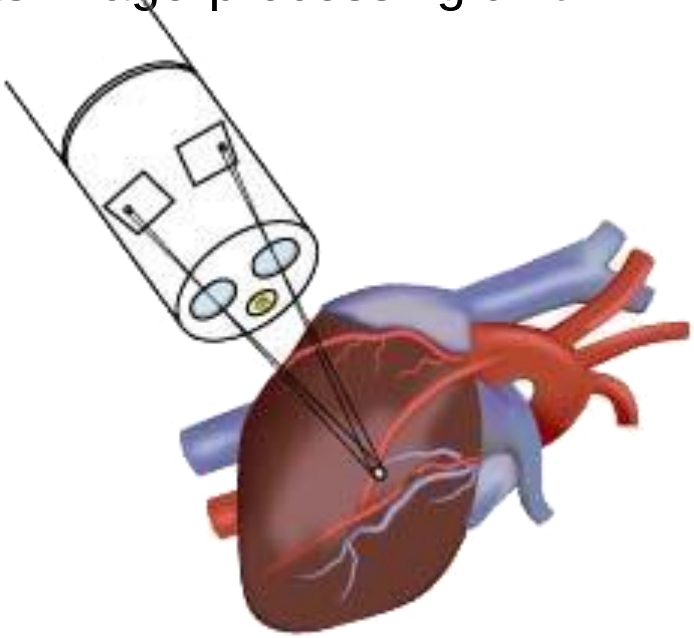
NCT

Computer-assisted endoscopy

- Motivation:
  Intraoperative support of the surgeon
  Goal is the usage of the endoscope as image-processing unit

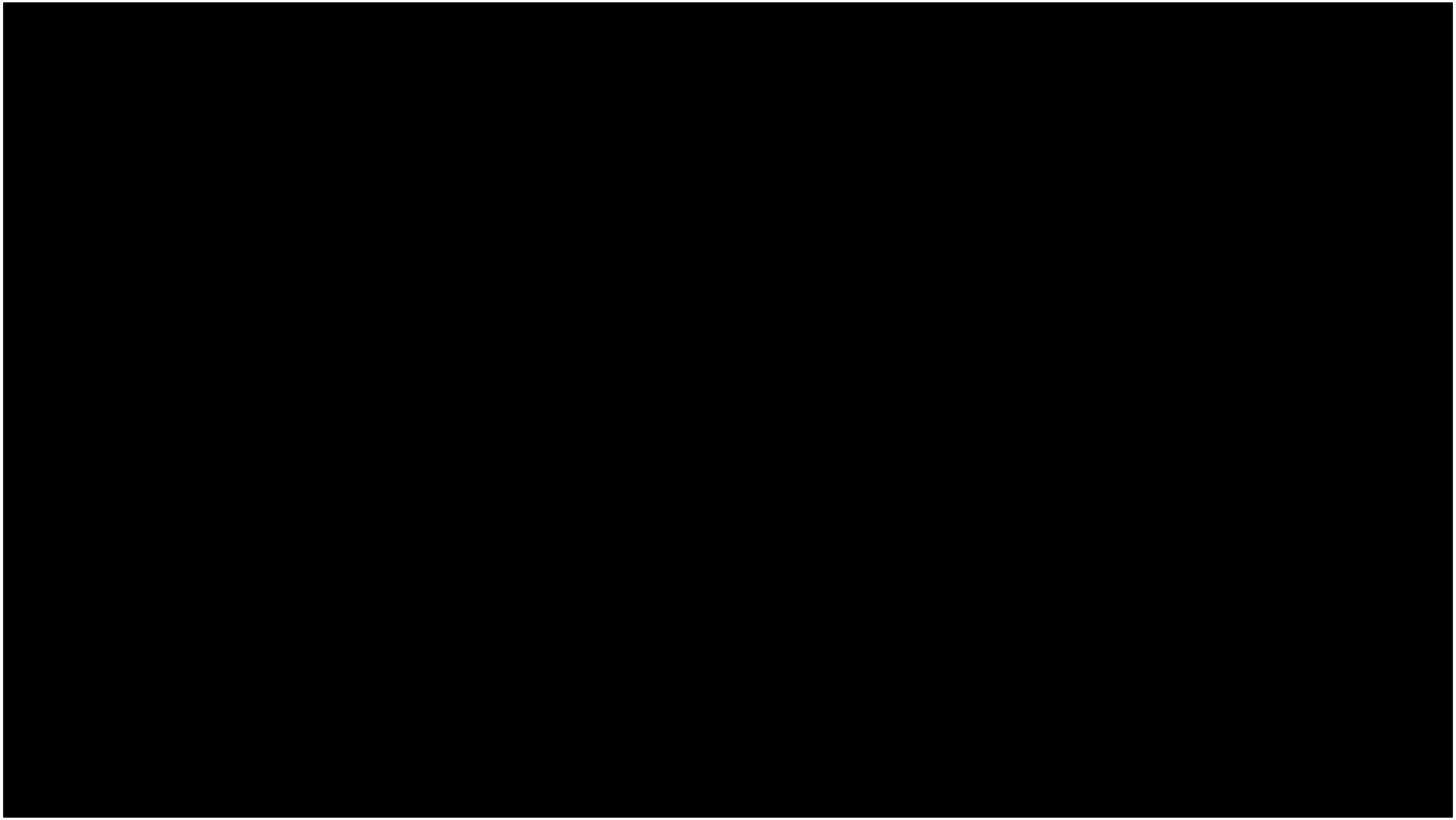
- Support:
  - Image processing
  - 3D modelling
  - Soft-tissue tracking
  - Registration
  - Navigation
  - Augmented reality

    …



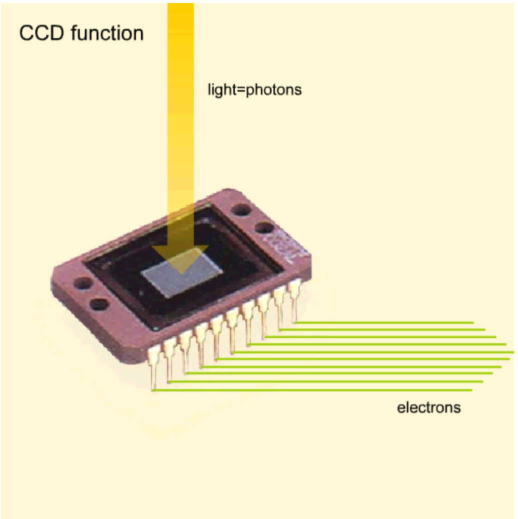
Quelle: D. Stoyanov, Imperial College

Laparoscopic Bowel Measurement



Camera system

- Camera head: Image acquisition and transformation into electronic signal
  - Photo-CCD
  - Lens/focus
  - Connection CCD/Lens
  - Cable



Quelle: Websurg

- Camera control unit
  - Receives signal from camera head
  - White balance, shutter control, image processing…
  - Transfer to screen

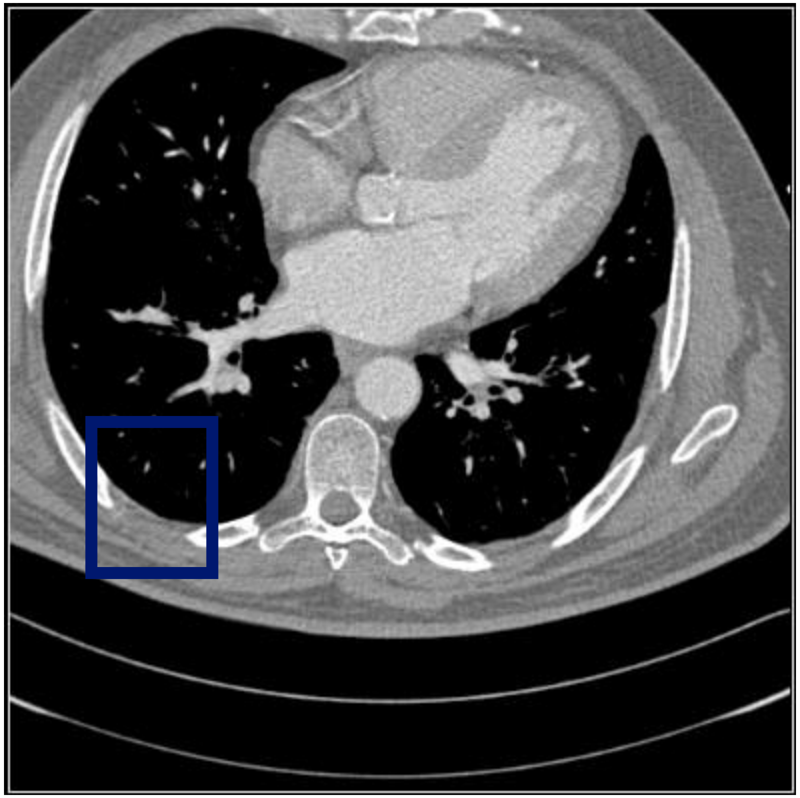
Goals of preprocessing

- Goals of preprocessing:
  - Correcting unclear or erroneous images (Image correction)
  - Preparation for next processing steps (Image improvement)
    - Highlight important information
    - Adaption of geometry, resolution, …

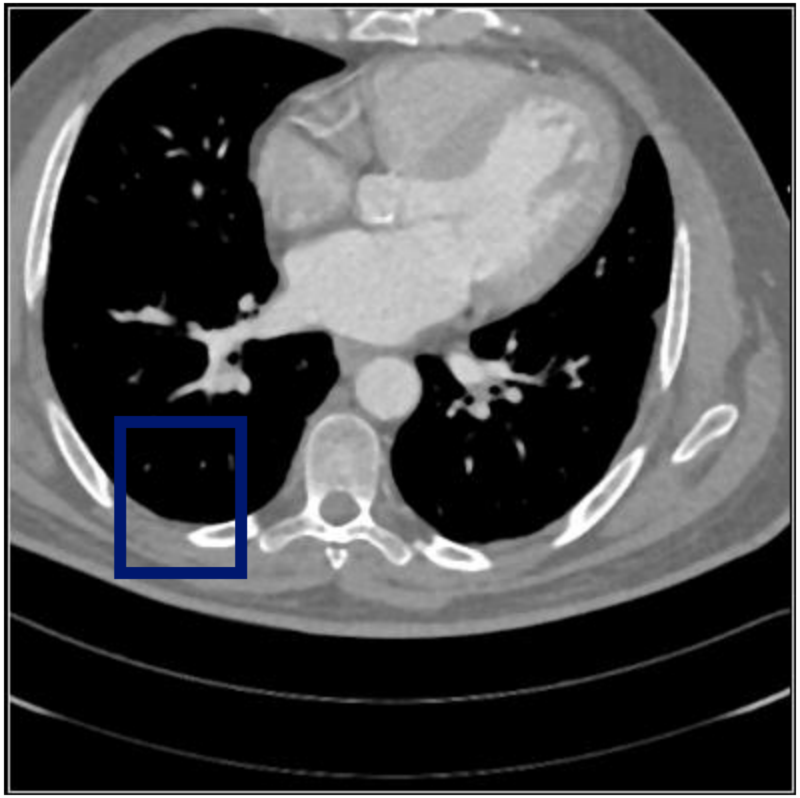
- According to law, original data has to be conserved

Preprocessing – Noise reduction

before

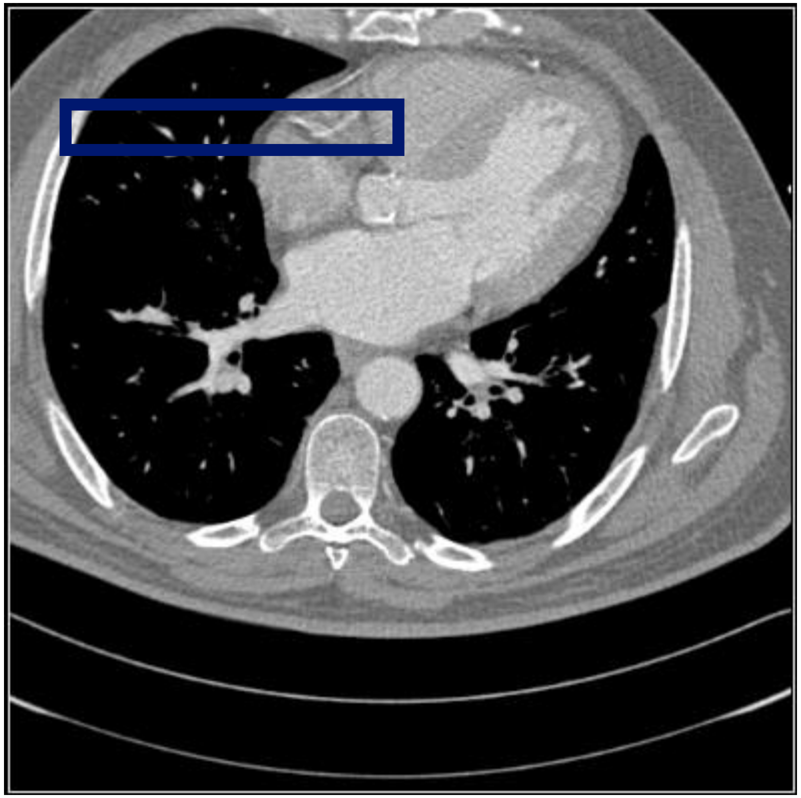


after



Bring out important information

before



after

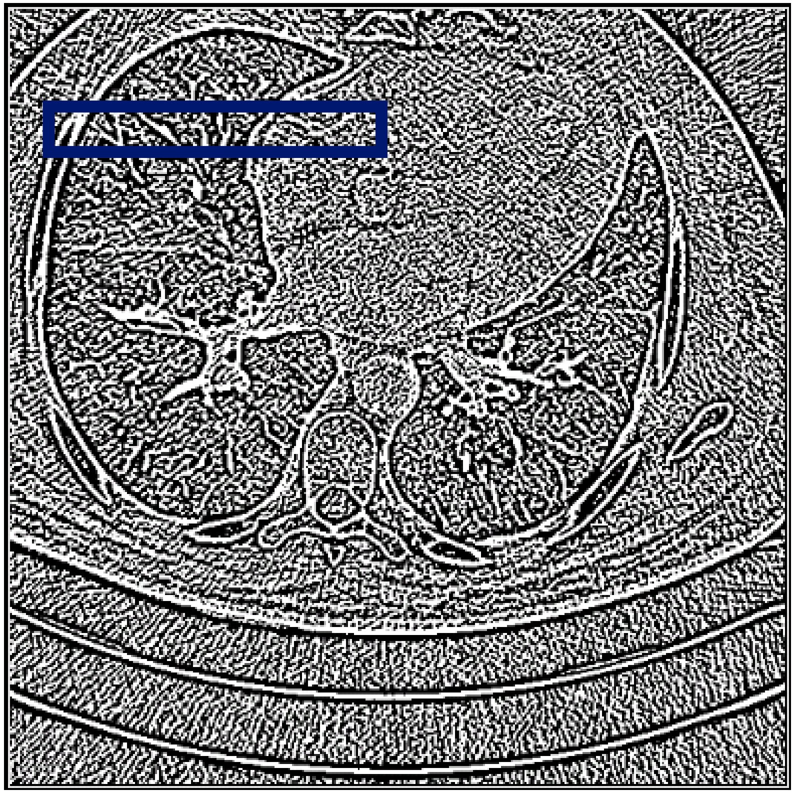
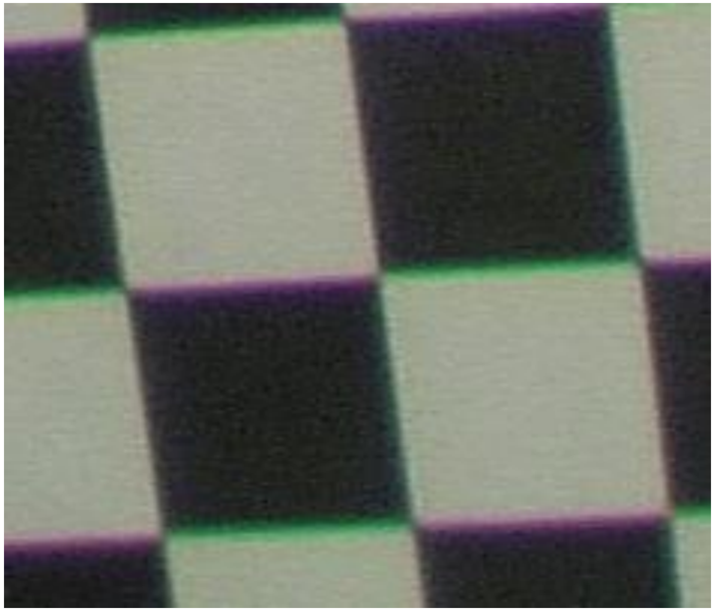
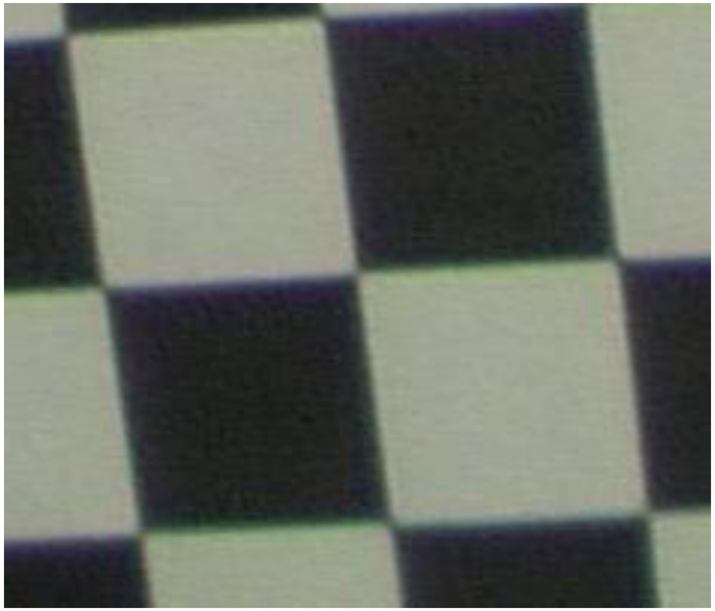


Image correction

before



after



Causes for errors in images

- Geometric distortions
- Defective sensor
- Bad/erroneous calibration
- Read error sensor matrix
- Noise
- Artifacts

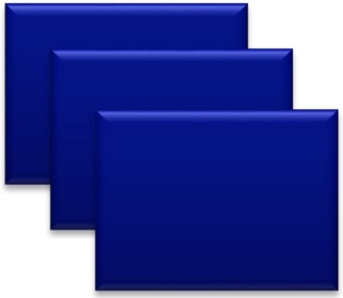
Error source	Error type
Optics	Geometrics Distortions
	Optical dispersion
	Color errors
	Blurring
Sensor	Inhomogeneity
	Color errors
	Transfer function
	Blurring
	Aliasing
A/D Converter	Quantization
	Aliasing
Amplifier	Noise

NCT

Image representation

2D grayscale image: Discrete function

$$\mathrm{Img} : [0..n] \times [0..m] \to [0..q]$$
$$(x, y) \mapsto \mathrm{G}(x, y) = g$$



Rows \ Column	0			*y*		m-1
0	80	0	100	70	0	80
	0	20	30	20	110	30
	25	79	136	100	30	0
x	20	20	30	50	90	85
	22	46	0	5	36	87
n-1	112	0	44	50	50	0

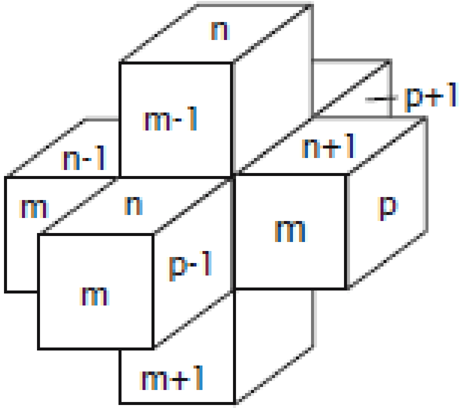
4-Neighbors

8-Neighbors

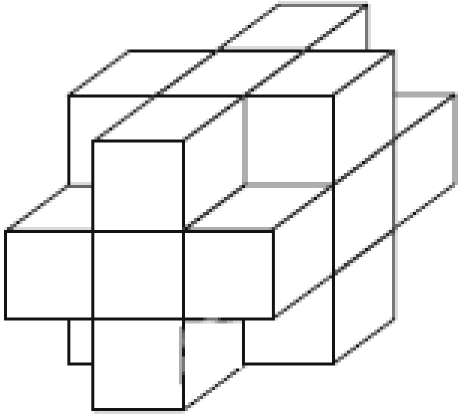
Image point (x, y)

Image representation

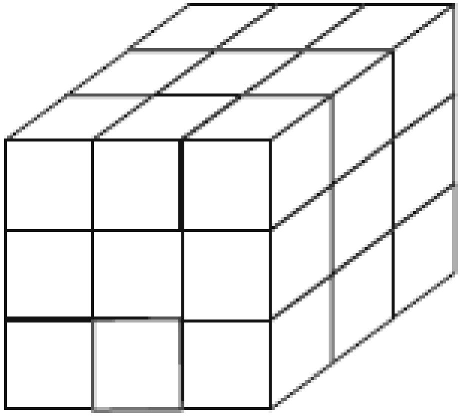
Neighborhoods in 3D



6-Neighborhood: connected surfaces



18-Neighborhood: connected borders



26-Neighborhood: connected corners

Source: http://www.springerlink.com/content/gl774141t3272413/fulltext.pdf

- Definition connected regions
- Important for e.g. segmentation

Image representation

- RGB-model:

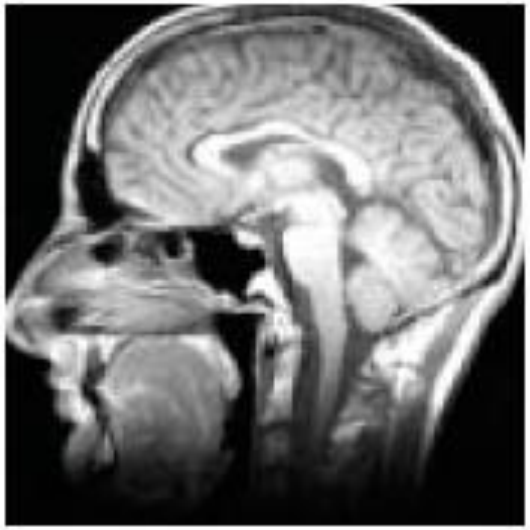
$$\begin{aligned} \mathrm{Img}\ :\ \ & [0..n] \times [0..m] \rightarrow [0..R] \times [0..G] \times [0..B] \\ & (x,\ y) \mapsto \mathrm{G}(x,\ y) = (r,\ g,\ b) \end{aligned}$$

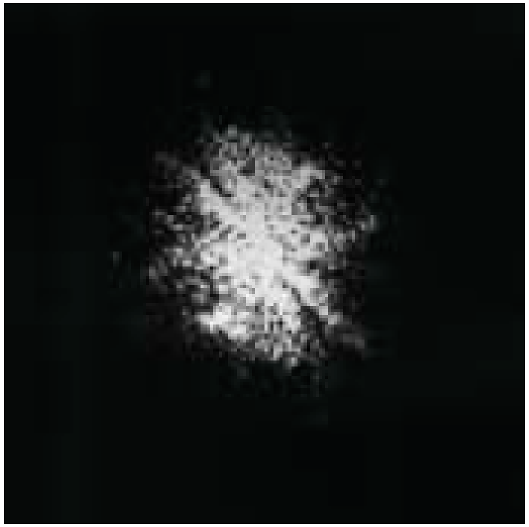
3 components: red, green and blue

usually 256 x 256 x 256 nuances = 16,8 Mio. colors

Image representation

- Alternative: Frequency space
- Fourier-Transform: Representing the images as a sum of sin- and cosine functions
- Two forms of information:
  - Amplitude: describes appearance of structure in image
  - Phase: describes position of structure in image





Characteristics of images

- Parameters, for characterizing/classifying global or local properties
- Local characteristics: Gradient, local contrast…
- Global characteristics:

- Average intensity

$$\overline{g} = \frac{1}{mn}\sum_{x=0}^{n-1}\sum_{y=0}^{m-1} g$$

- Intensity variance

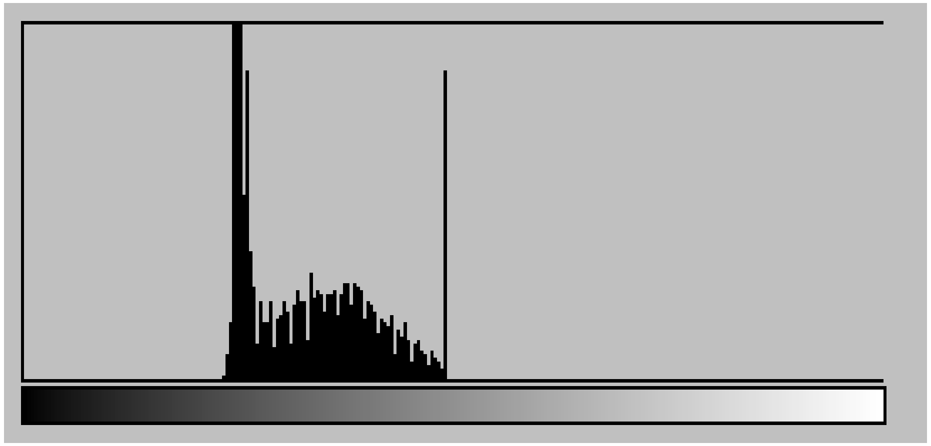
$$q = \frac{1}{mn}\sum_{x=0}^{n-1}\sum_{y=0}^{m-1}\left(g - \overline{g}\right)^2$$

Characteristics of images

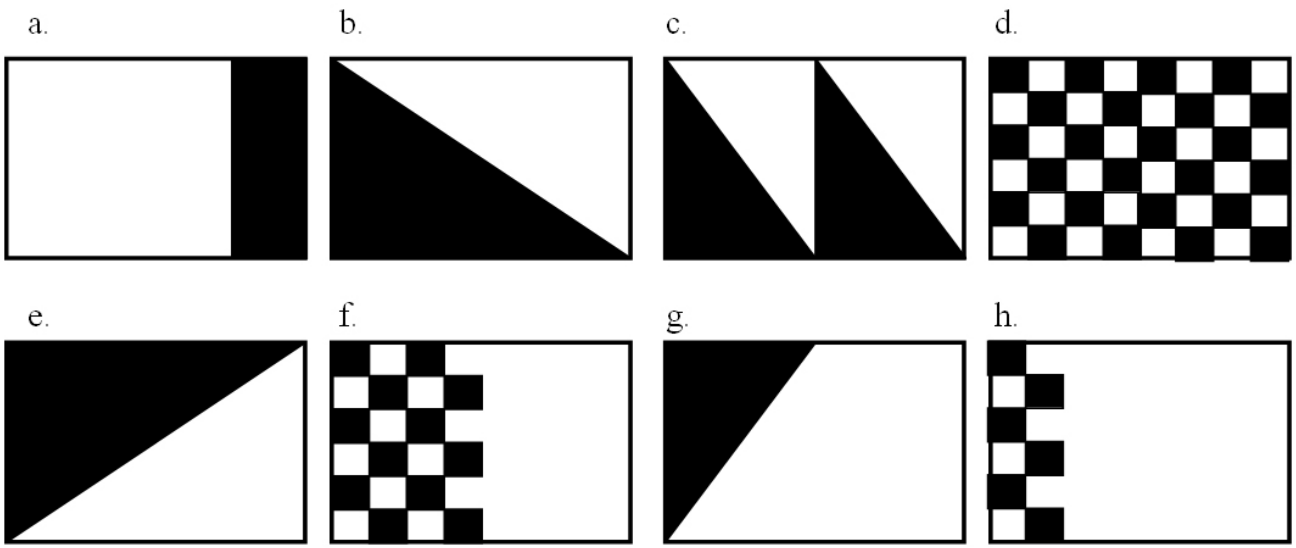
- Histogram: occurrence rate of features

$$h(g) = \# \left(x, y\right) : G(x, y) = g,\ g \in [0..q]$$

$$p(g) = \frac{h(g)}{mn} \quad = \text{Relative frequency:} \quad \sum_{g=0}^{255} p(g) = 1$$



Which images has the same histogram as a ?



A: c+g

B: g+h

C: f+g

D: none

Operations on images

- Point operation

  Results of transform f is only influenced by a single pixel

- Local operation

  Results of transform f is only influenced by the surroundings of a pixel

- Global operation

  Results of transform f is influenced by the entire image

Point operation

- Modification of a single pixel through operations that are only depend on value and position of that pixel

$$g'_{xy} = P_{xy}(g_{xy})$$

- Indices x, y of function P describe possible dependency of the function to the position of the pixel

Example: $$g'_{xy} = \begin{cases} 2g_x, if\ (xy)\ mod\ 2\ \equiv 0 \\ 4g_x, if\ (xy)\ mod\ 2\ \equiv 1 \end{cases}$$

Homogenous point operations

- Independent from position of the position of image point, only dependent on value

$$g'_{xy} = P(g_{xy})$$

Generally not invertible operation (compare threshold segmentation)

→ affine homogenous point operations

→ non-affine homogenous point operations

Affine point operations

- Definition:

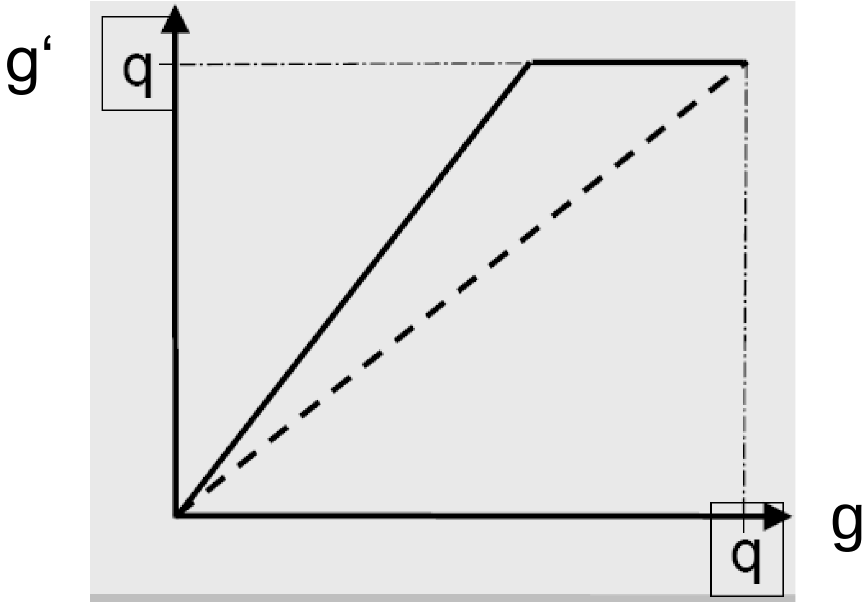
$$P : [0..q] \to [0..q]$$
$$g' = ag + b$$

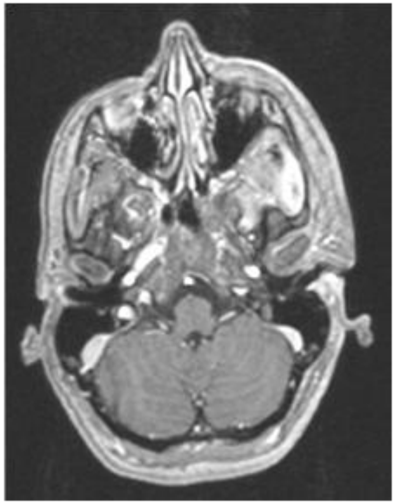
- Parameters a, b define function:

$a > 1, b = 0$	Contrast increase
$0 < a < 1, b = 0$	Contrast reduction
$a = 1, b > 0$	Brightness increase
$a = 1, b < 0$	Brightness reduction

Affine point operation

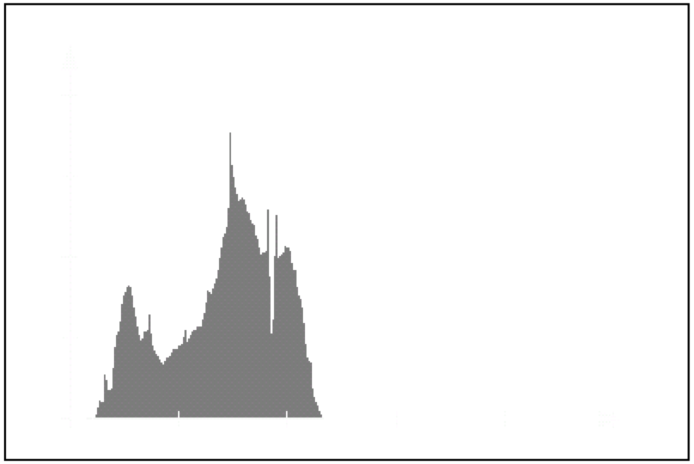
- Geometric visualization of P:
  Characteristic line of transformation
- E.g..: Contrast increase ($a > 1, b = 0$)

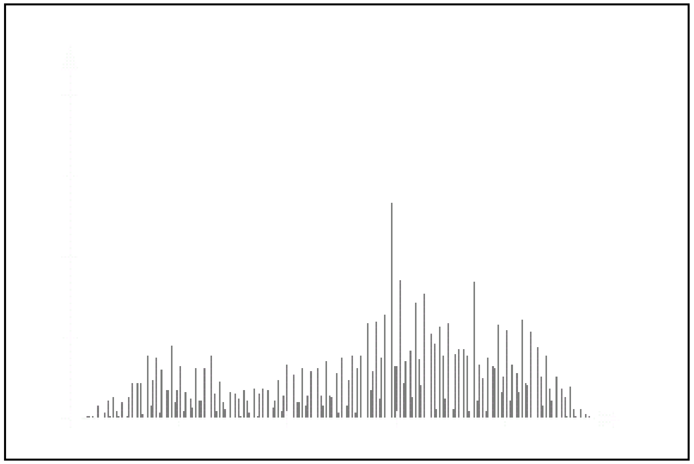




Increase contrast

- Histogram only has spikes in a small area of possible grayscale values
- By increasing contrast, the small area is spread onto a larger area
- Characteristics
  - Usage of the entire grayscale spectrum
  - Visual effect is improved





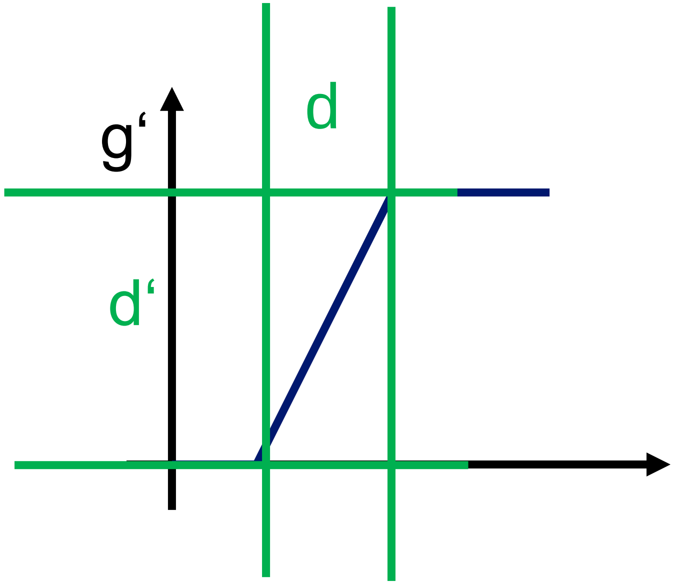
Lookup-table

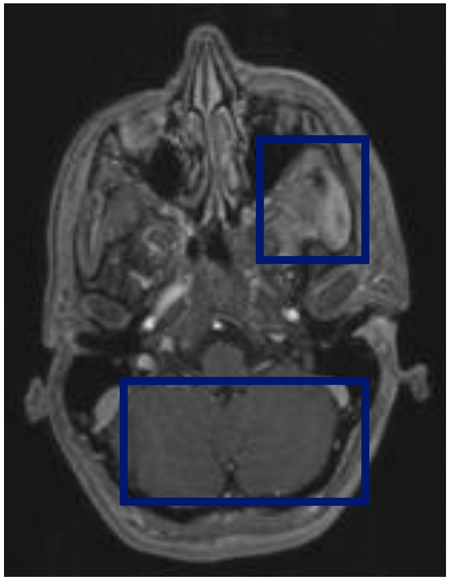
- Computation of the homogenous point operations can be computationally expensive
- Solution:
  - Compute once and create a Lookup-Table (LUT)
  - Generally computation time savings increase
    - Lower number of bits
    - Increasing image size
    - Complexity of the transformation

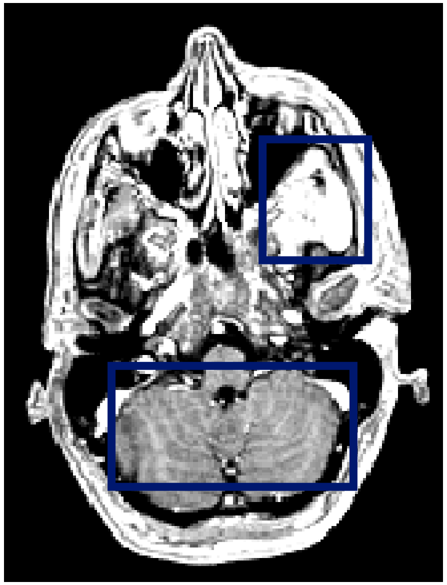
g	1	2	3	4	5	6	…
g‘	2	4	9	16	25	36	…

Contrast stretch

- Problem: For the human eye, small differences in grayscale values are hard to distinguish
- Solution: To make fine differences visible, the greyscale range of the select area is stretched.







- Information in other areas can become lost

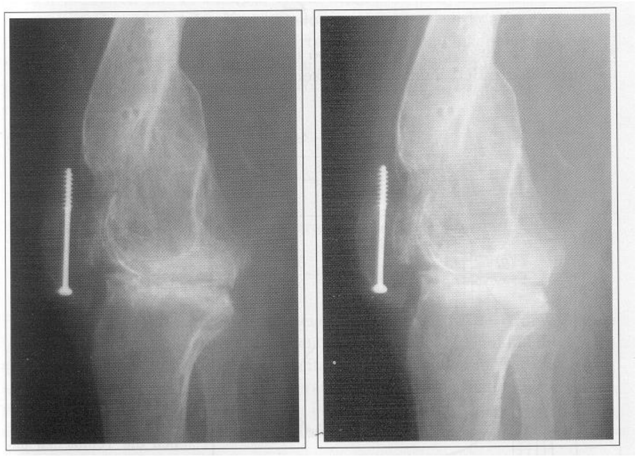
Non-affine: Monotone grayscale transformation

- Non-affine point operations: arbitrary function
- Example: non-linear, monotone grayscale transformation
- Results in the following LUT:

$$g' = \left(\sqrt{(G-1)*g}\right)$$

g	0	1	2	3	4	…	100	…	200	…	250	251	252	253	254	255
T(g)	0	16	23	28	32	…	160	…	226	…	252	253	254	254	254	255

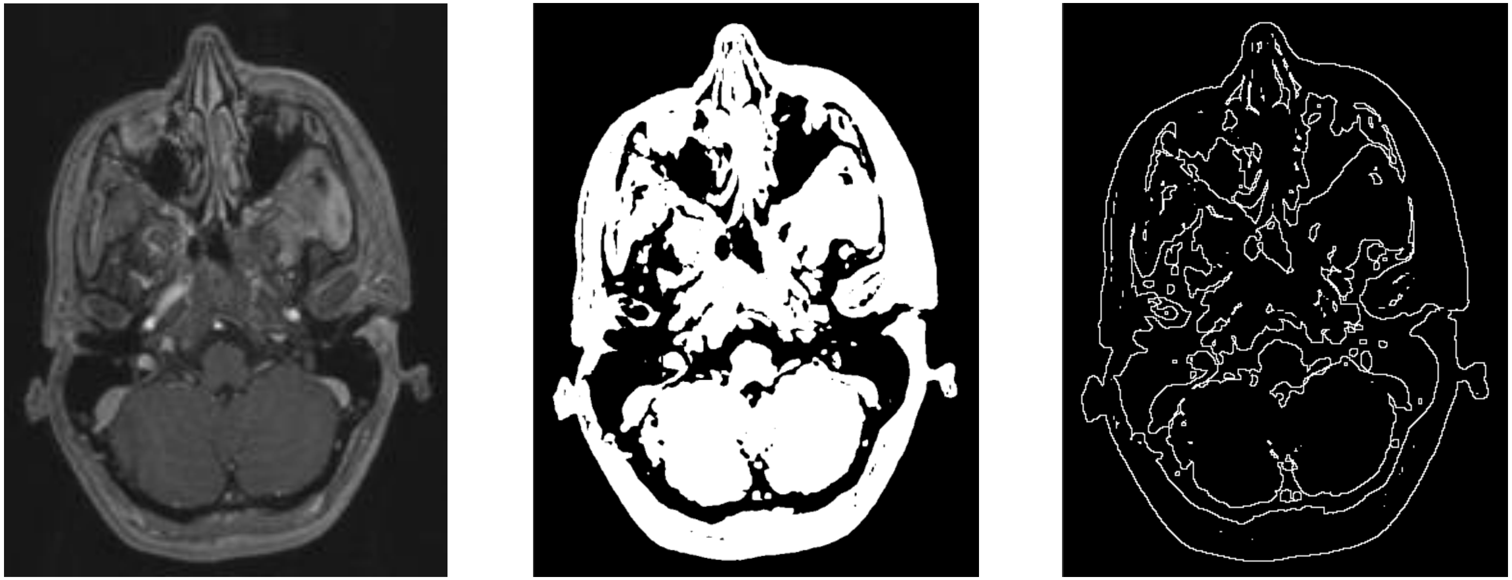
- Stretch in lower grayscale range, compression in upper grayscale range



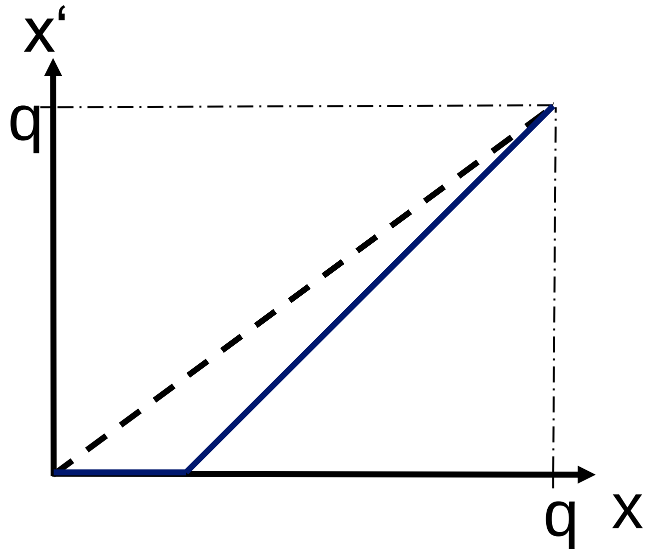
Binarization

- Replace each pixel with a maximum or minimum value, dependent on threshold

→ Simple segmentation, edge detector



This characteristic line of a transformation describes a …?



- A: Contrast increase and brightness reduction
- B: Inversion
- C: Contrast and brightness reduction
- D: Contrast reduction and brightness increase

Local operations

- Takes the neighborhood of the image point into consideration

- Execution through masks/windows:
  Size defined through the neighborhood

- Usage
  - Smoothing filter (Low-pass filter)
  - Edge filter (High-pass filter)
  - …

Local Operation in 1D

- Input g:

0	0	0	0	0	5	5	5	5	5

* * * * *

- Filter h:

0.2	0.2	0,2	0.2	0.2

- Output g‘:

		0							

- Formel:

$$g'(x) = \sum_{u=-r}^{r} h(u)g(x+u)$$

Local Operation in 1D

- Input g:

0	0	0	0	0	5	5	$\boxed{5}$	5	5

- Filter h:

*	*	*	*	*
0.2	0.2	0.2	0.2	0.2

- Output g‘:

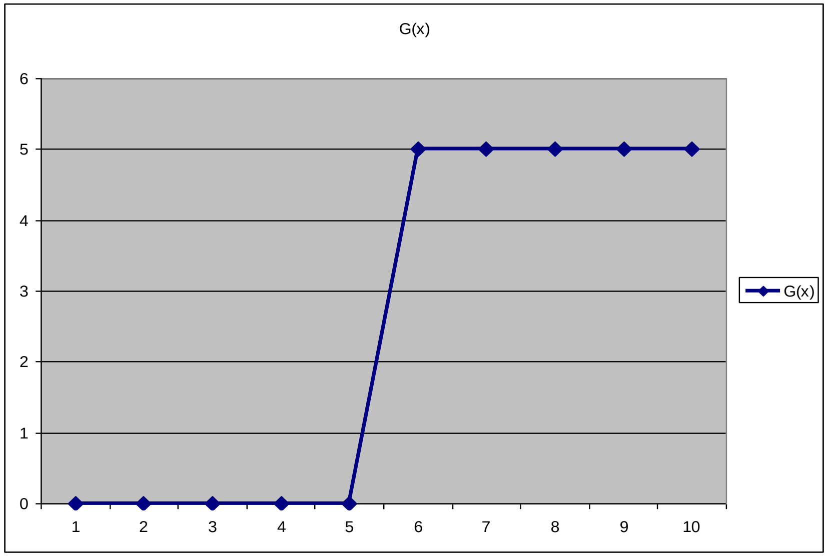
		0	1	2	3	4	$\boxed{5}$		

- Formel:

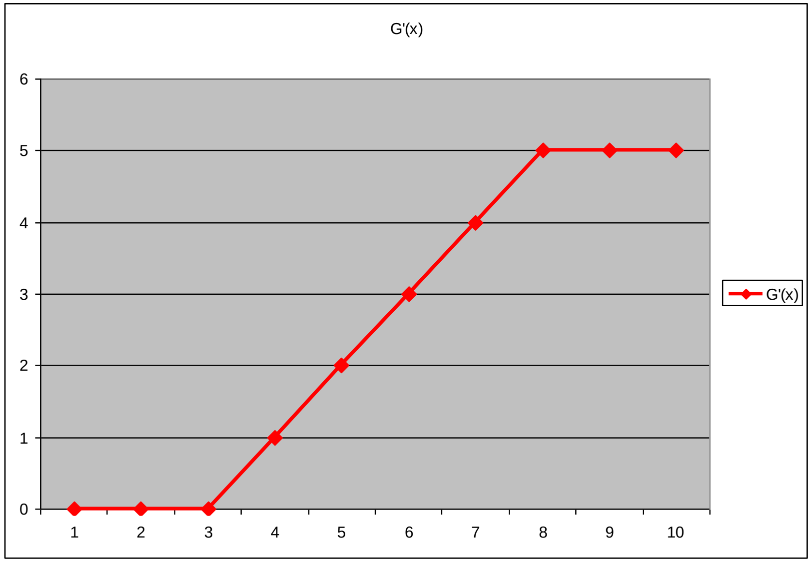
$$g'(x) = \sum_{u=-r}^{r} h(u)g(x+u)$$

Local Operation in 1D

0.2	0.2	0.2	0.2	0.2



Input



Output

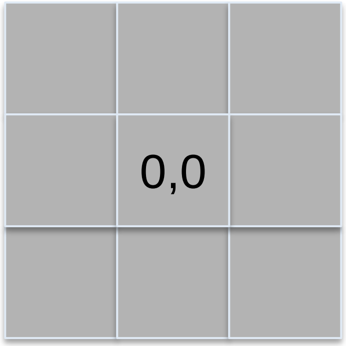
Local Operation in 2D

- Input: Image matrix g(x,y)
- Filter: Filter matrix h(2r+1, 2r+1)
- Output: Image matrix G'(x,y)
- Image filtering is a convolution with filter matrix/mask:

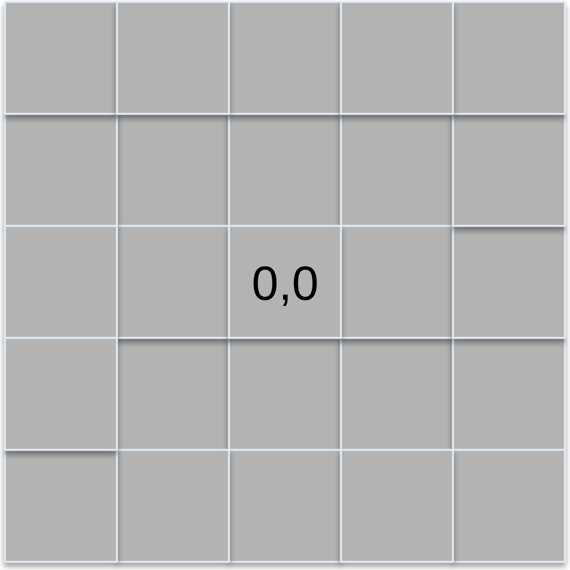
$$g'(x,y) = \sum_{u=-r}^{r} \sum_{v=-r}^{r} h(u,v)\, g(x+u, y+v)$$

Local Operation in 2D: Masks

- Discretization of the filter function
- Each entry in the mask is assigned a weight h(i,j)
- Center point of the mask has coordinate (0,0)



r = 1



r = 2

Example

Input

0	2	2	2	0
1	10	2	10	1
2	2	20	2	2
1	10	5	10	1
0	2	10	2	1

Output

$\frac{1}{10}$

1	1	1
1	2	1
1	1	1

Example

Input

0	2	2	2	0
1	10	2	10	1
2	2	20	2	2
1	10	5	10	1
0	2	10	2	1

Output

?	?	?	?	?
?	**5.1**	**5.4**	**5.1**	?
?	**5.5**	**9.1**	**5.5**	?
?	**6.2**	**6.8**	**6.3**	?
?	?	?	?	?

$\frac{1}{10}$

1	1	1
1	2	1
1	1	1

Image boundary

- Problem at image boundary: Filter not defined for all image points of the environment of border points
  - Border points are not transformed

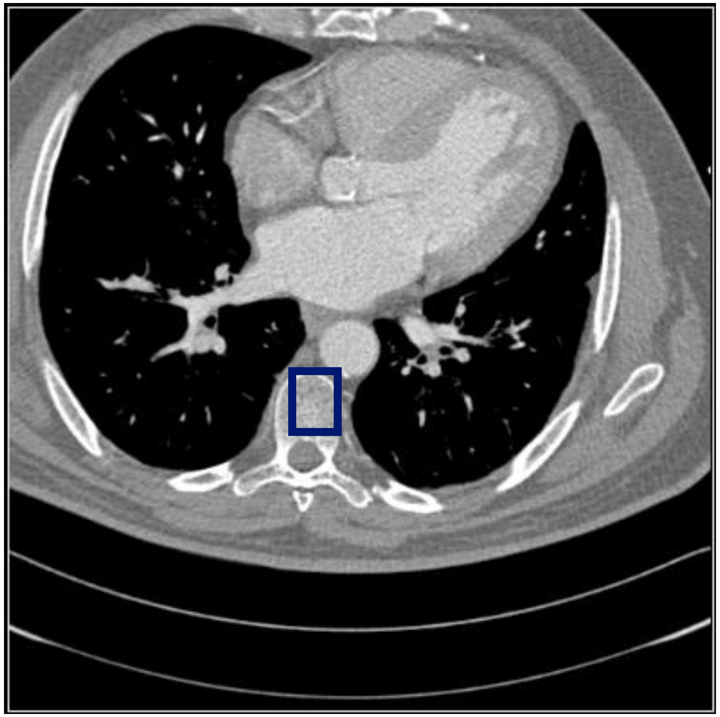
    Con: Image shrinks with iterative transformations
  - Image is extrapolated past the border

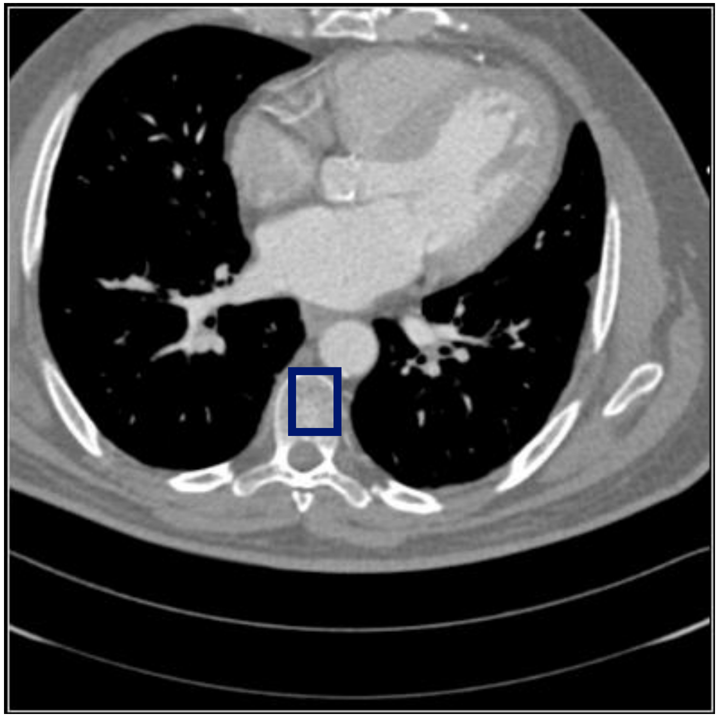
    Problem: Extrapolation errors can travel into the inside of the image when iterated
  - Mask is restricted so it doesn't exceed past the image boundaries
  - Image is extended periodically

    Problem: Images, that don’t have periodicity

Local operations: Smoothing filter

- Goal: Suppression of noise and image smoothing
- Local variations of the image function values are reduced
- Homogenization of the image function values





Box filter

- Averages out extreme points
- Smoothing effect proportional to mask size
- Fast computation
- Can cause „smearing“
  → Edges are flattend

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

0	1	1	1	0
1	1	1	1	1
1	1	3	255	1
1	2	1	1	1
0	2	3	1	1

What will happen?

Box filter computation

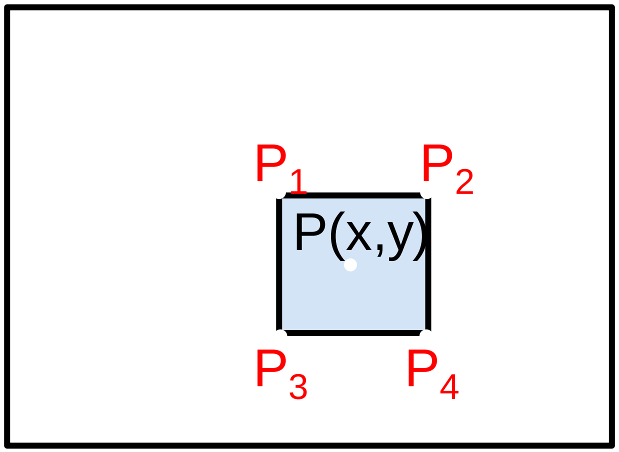
- Computation with constant computational effort: usage of integral images

$$I_{\sum}(x,y) = \sum_{i=0}^{i\leq x} \sum_{j=0}^{j\leq y} I(i,j)$$

- Simple computation of pixel sums out of four values from the integral image

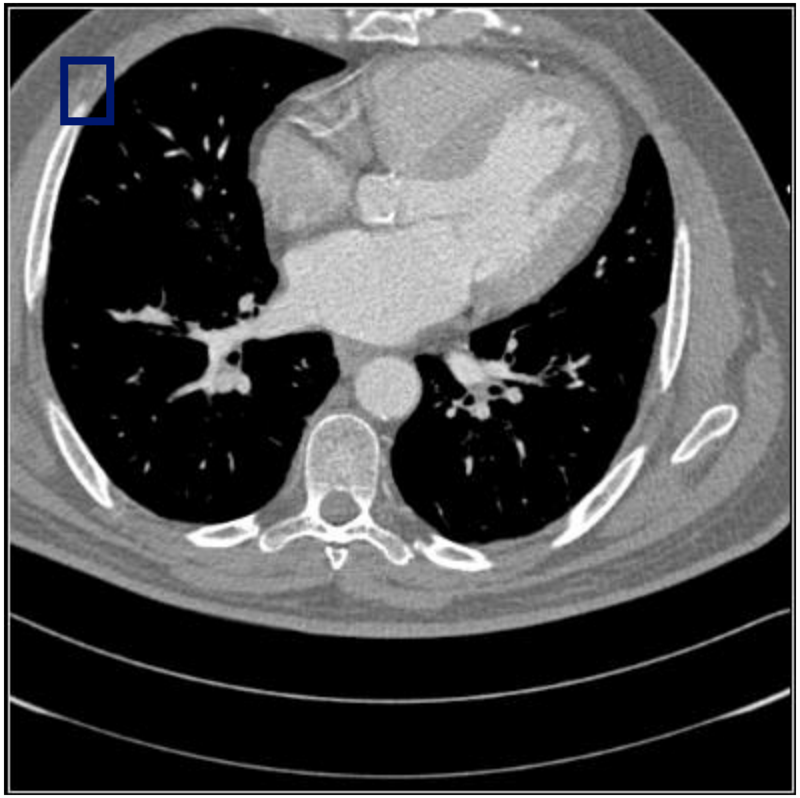
- New grayscale value (dependent on size of neighborhood N):

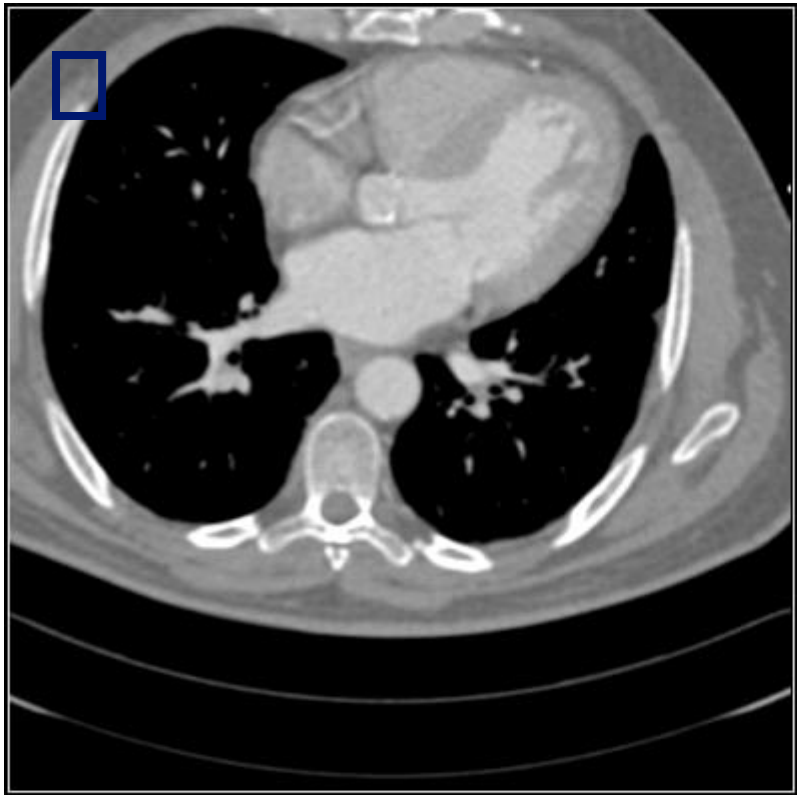
Integral image



$$g'\big(P(x,y)\big) = \frac{1}{size(N)}\left(I_{\sum}(P_1) + I_{\sum}(P_4) - I_{\sum}(P_2) - I_{\sum}(P_3)\right)$$

Example: Box filter





Gaussian filter

- Better smoothing filter than box filter
- Structure is inspired by Gaussian normal distribution
- Normal distribution can be approximated via binomial distribution
- Influence of the environment in dependence to the distance to the image center → no strong flattening of edges

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

$\sigma = 0.625$ → $\frac{1}{121}$

1	2	3	2	1
2	7	11	7	2
3	11	17	11	3
2	7	11	7	2
1	2	3	2	1

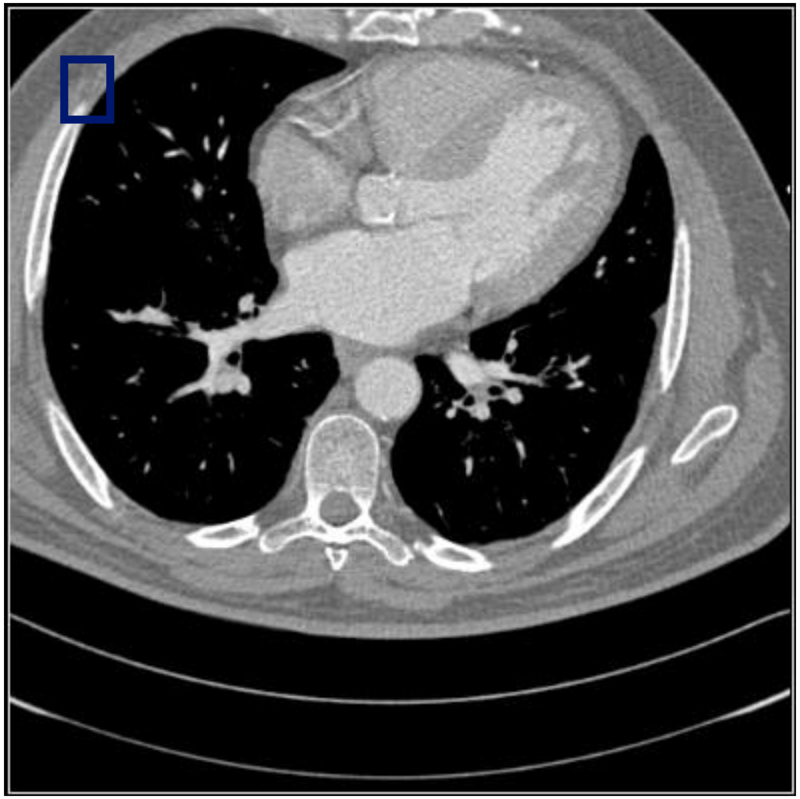
Gaussian filter computation

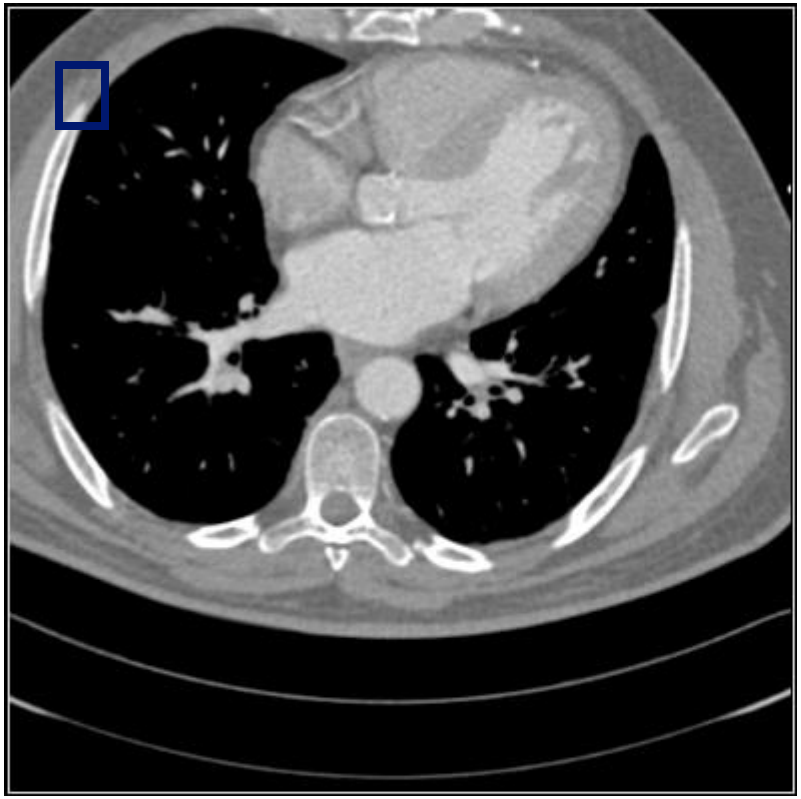
- Computation with reduced time effort:
  Separation = Combination of two 1D-Gaußfiltern
- First filter in horizontal direction with 1D-Filter
- Then filter in vertical direction with 1D-Filter

$$\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \cdot \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Example: Gaussian filter





Median filter

2	4	3
5	9	2
7	2	4

2,4,3,5,9,2,7,2,4

2,2,2,3,4,4,5,7,9

9 replaced with 4

- Center pixel is assigned the median of the grayscale values of the local neighborhood
- Robust against outliers
- Sharpness barely suffers

  Edges are mostly conserved
- Smoothing effect is less

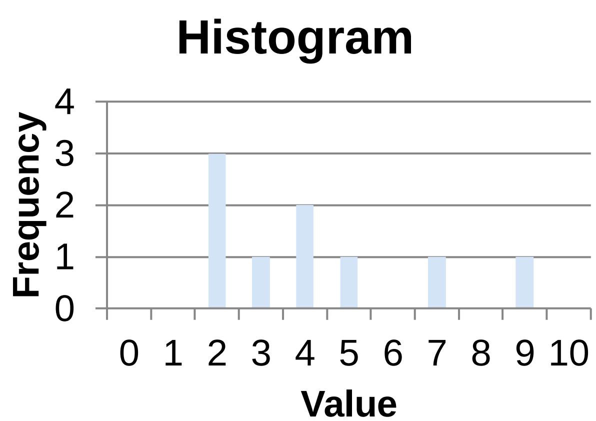
New:

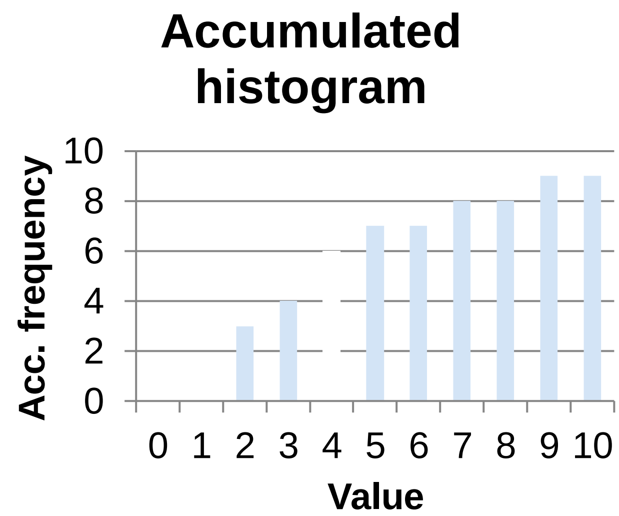
2	4	3
5	4	2
7	2	4

Median filter computation

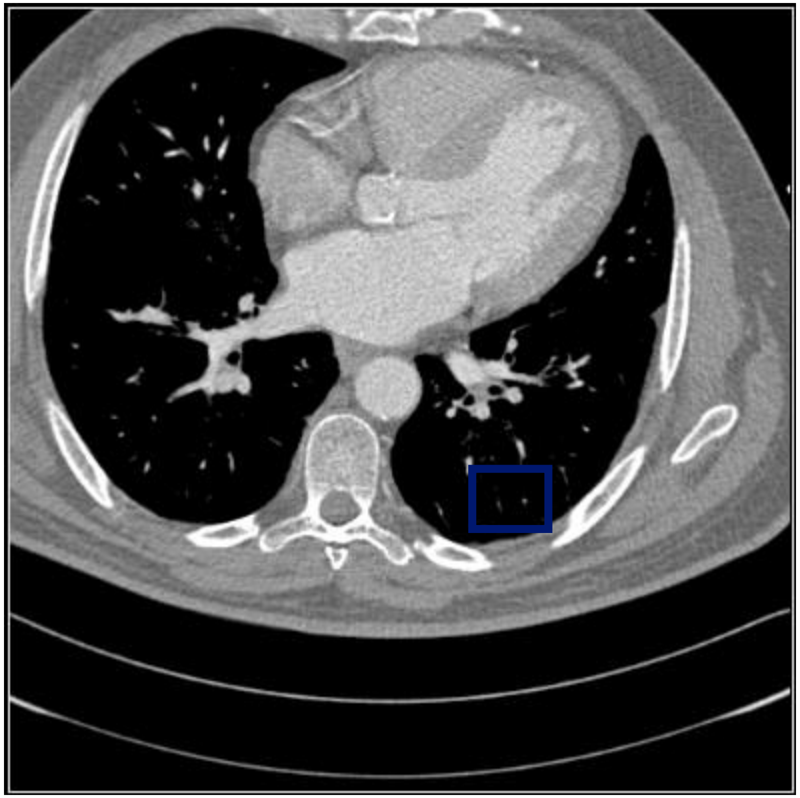
- Sorting costly → Usage of a histogram
- Grayscale image: Create histogram out of the neighborhood of the viewed pixel
- Accumulate entries in histogram, starting at smallest element, until sum is larger than half of the size of the neighborhood → Median

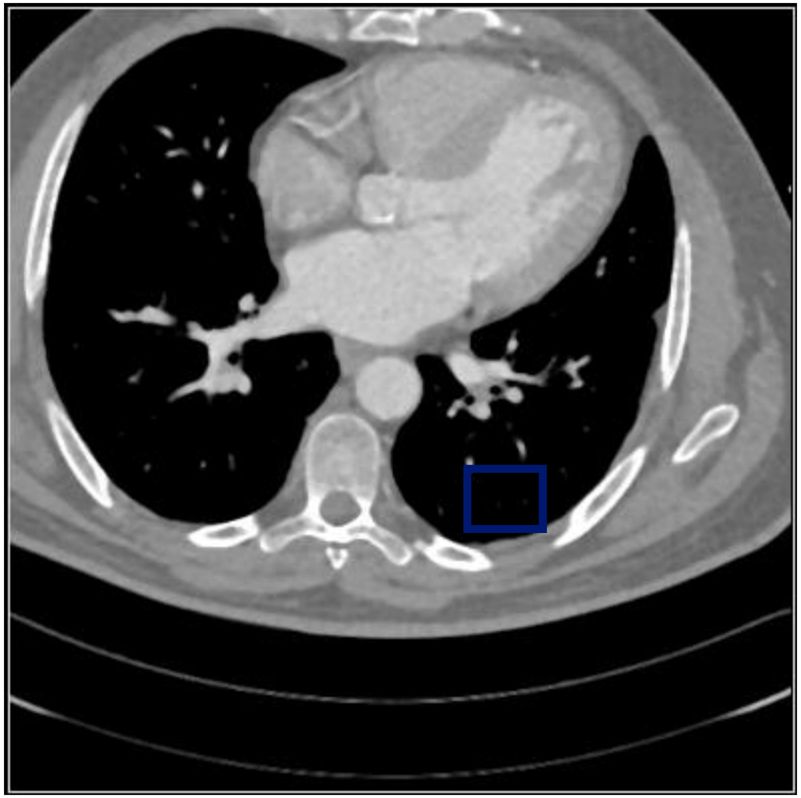
2	4	3
5	9	2
7	2	4





Example: Median filter





Problem of filters up to now

- In medical image data, the border between different types of tissue can be difficult to recognize.
- By applying any of the former filters, fine structures can be lost (Smearing of boundaries)
- Idea: Don't use static filter masks/weights, but instead adapt shape of mask or weights to the local image contents.

→ anisotropic smoothing filter
→ Bilateralfilter
→ „Mean Shift"-Filter

Anisotropic smoothing filter

- Anisotropic = varying in magnitude according to the direction

- Consists, for example, out of:
  - A Gaussian filter G (with constant variance)
  - A Conductance-term C

  Both elements are combined via multiplication

- Idea:

  To not smear edges, the Conductance-term weakens the Gaussian filter at the required positions

Bilateral filter

Idea: Combination of two filters, e.g. Gaussian filter

- Let p be the center pixel, $q \in N$ of the pixel neighborhood, I_p with I_q being their intensities (e.g. grayscale)

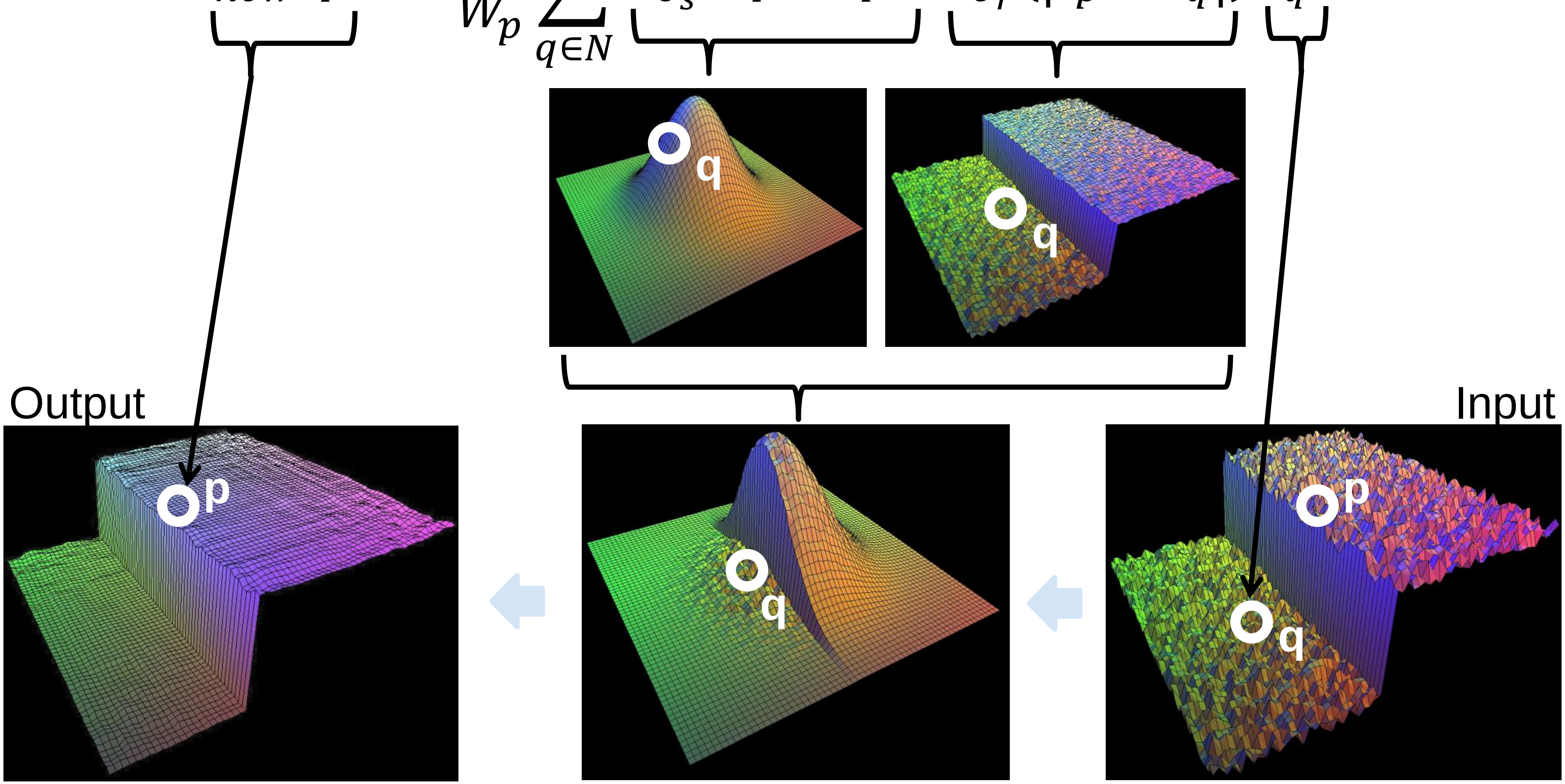
$$I_{new}(p) = \frac{1}{W_p} \sum_{q \in N} G_{\sigma_s}(\|p - q\|)\, G_{\sigma_r}(|I_p - I_q|)\, I_q$$

$$W_p = \sum_{q \in N} G_{\sigma_s}(\|p - q\|)\, G_{\sigma_r}(|I_p - I_q|)$$

- G_{σ_s}: Weight dependent on the distance between the current pixel and the center pixel
- G_{σ_r}: Weight dependent on the difference of intensity between the current pixel and the center pixel

Bilateral filter

$$I_{new}(p) = \frac{1}{W_p}\sum_{q\in N} G_{\sigma_s}(\|p-q\|)\, G_{\sigma_r}(|I_p - I_q|)\, I_q$$



Source: Durand et al.: „Fast bilateral filtering for the display of high-dynamic-range images"

Bilateral filter

- Edge conserving smoothing filter
- Non-linear: Relatively expensive for computation

- Joint-Bilateral Filtering: Combination of two images, smoothing of the first image using the intensity differences to the second image

„Mean shift“-Filter

Idea: Replace center pixel with the most probable value from a defined neighborhood

- Iterative method:
    1. Estimate a probability density function that describes the feature space on a defined search window
    2. The pixel is assigned the value of the local maximum of the probability density function
    3. Search window is moved to the position of the maximum

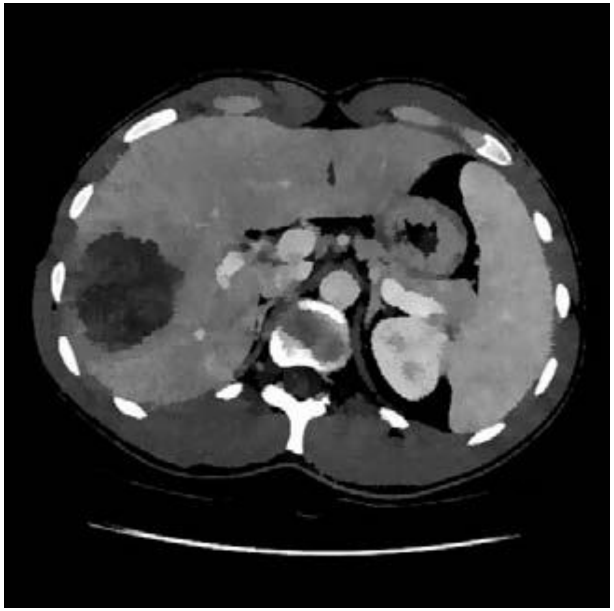
- Size of the search windows is reduced with each iteration („Parzen Window“-method)
- Feature space: spatial- and intensity-distances

Bilateral-filter vs. „Mean shift“-Filter

- Both operate using spatial- and intensity neighborhoods
- Main difference: Mean Shift is iterative with movable search window

Original

Bilateral filter



„Mean Shift“-Filter

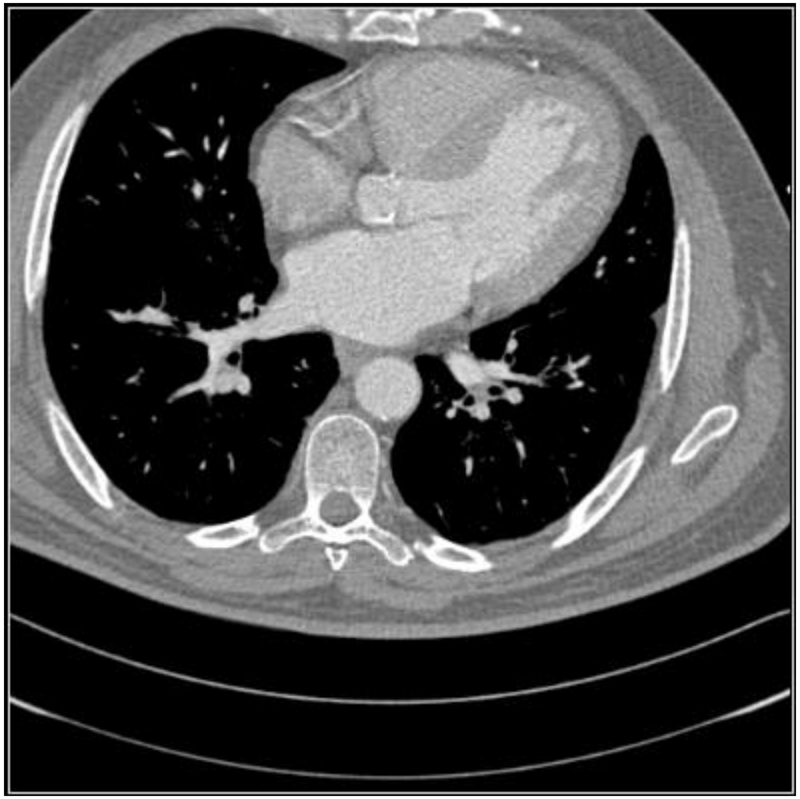
Quelle: Dominguez et al.: „Fast 3D Mean Shift Filter for CT Images”

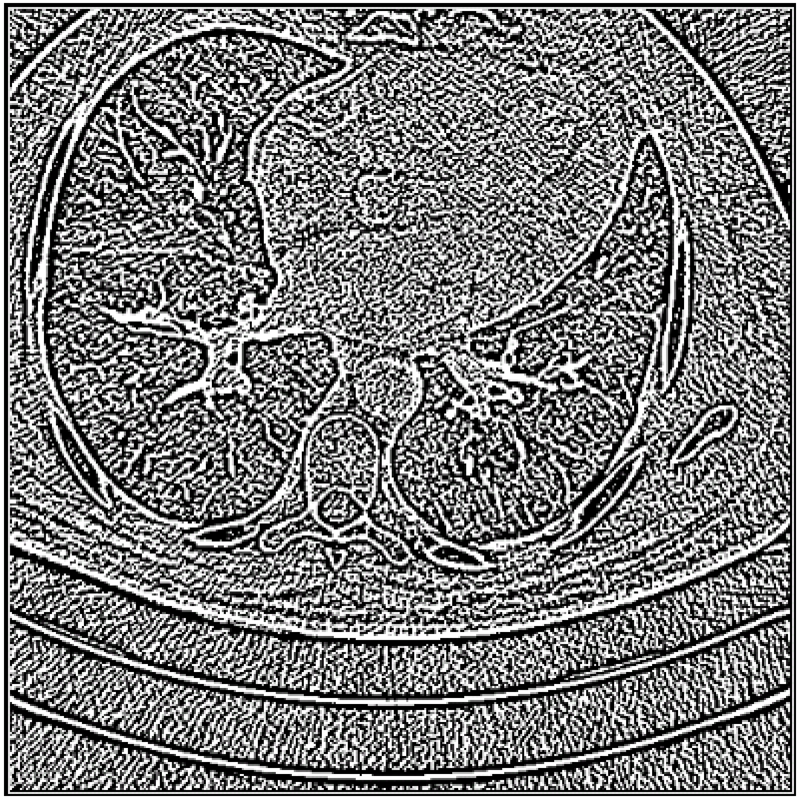
Which filter would most likely improve the image?

- A: Bilateral filter
- B: Median filter
- C: Gaussian filter
- D: Box filter

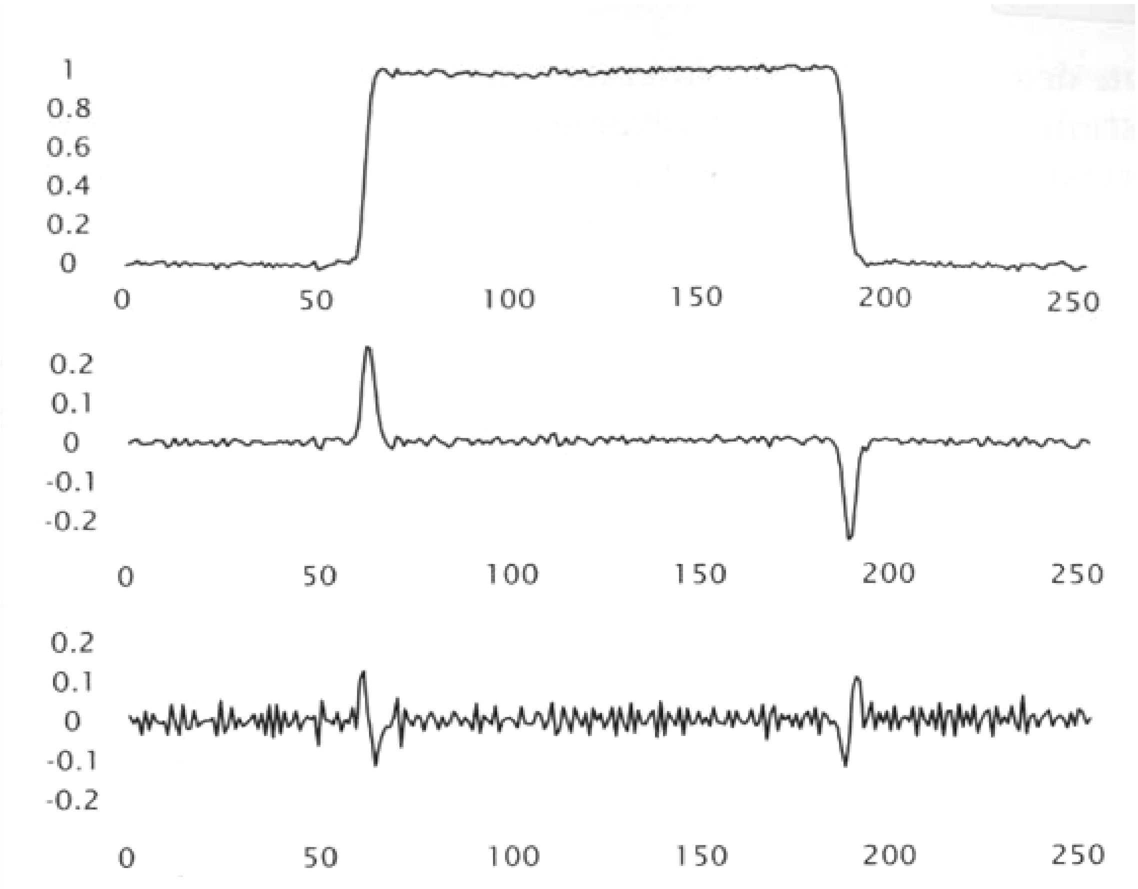
Edge filter

- Goal: Enhancing changes in grayscale between neighboring pixels





Edge filter - Derivatives



Gradient-based edge detection

- Edge: strong, local change in the image function G
  → Value of the gradient assumes local maximum

- Gradient operations:

Gradient:

$$\nabla G(x, y) = \left[\frac{\partial G(x, y)}{\partial x}, \frac{\partial G(x, y)}{\partial y} \right]^T$$

Value/Magnitude:

$$\left|\nabla G(x, y)\right| = \sqrt{\left(\frac{\partial G(x, y)}{\partial x}\right)^2 + \left(\frac{\partial G(x, y)}{\partial y}\right)^2}$$

Orientation:

$$\tan \Theta = \frac{\partial G(x, y)}{\partial y} \bigg/ \frac{\partial G(x, y)}{\partial x}$$

NCT

Gradient-based edge detection

- Gradient is always oriented in the direction of the strongest change in the image function
- Value of the gradient is a measurement for the strength of change in the image function
- Discrete approximation of value:

$$|\nabla G(x, y)| = |G(x+1, y) - G(x, y)| + |G(x, y+1) - G(x, y)|$$

Saves times as computing square root is expensive

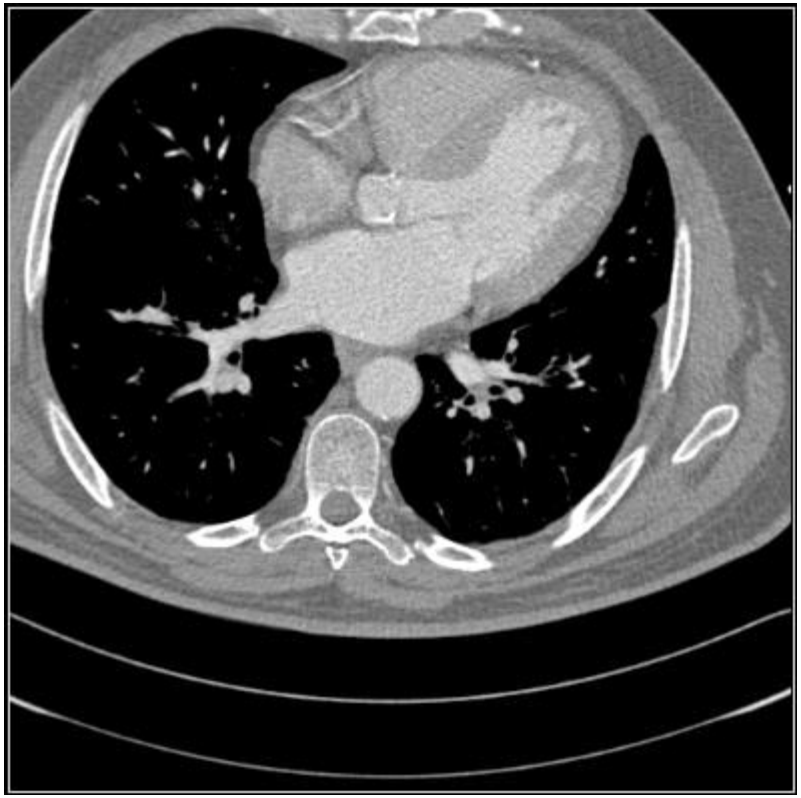
Edge detector - Prewitt Filter

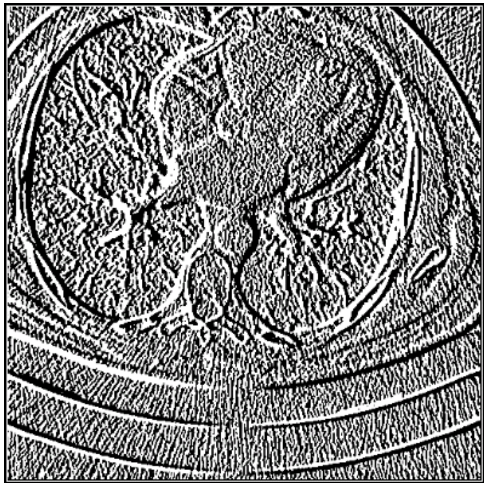
- Differences of pixel values are averaged with the same weight
- Prewitt-X filter enhances vertical, Prewitt-Y filter horizontal edges

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$P_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Example Prewitt filter





Prewitt filter

- Prewitt-Operator: Combination of the Prewitt filter for computing the gradient magnitude M

$$M \approx \sqrt{P_x^2 + P_y^2}$$

- Afterwards: Threshold filter

Sobel filter

- Gaussian-based weighting the difference of the pixel values to enhance edges

- Sobel-X filter enhances vertical, Sobel-Y Filter horizontal edges

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

NCT

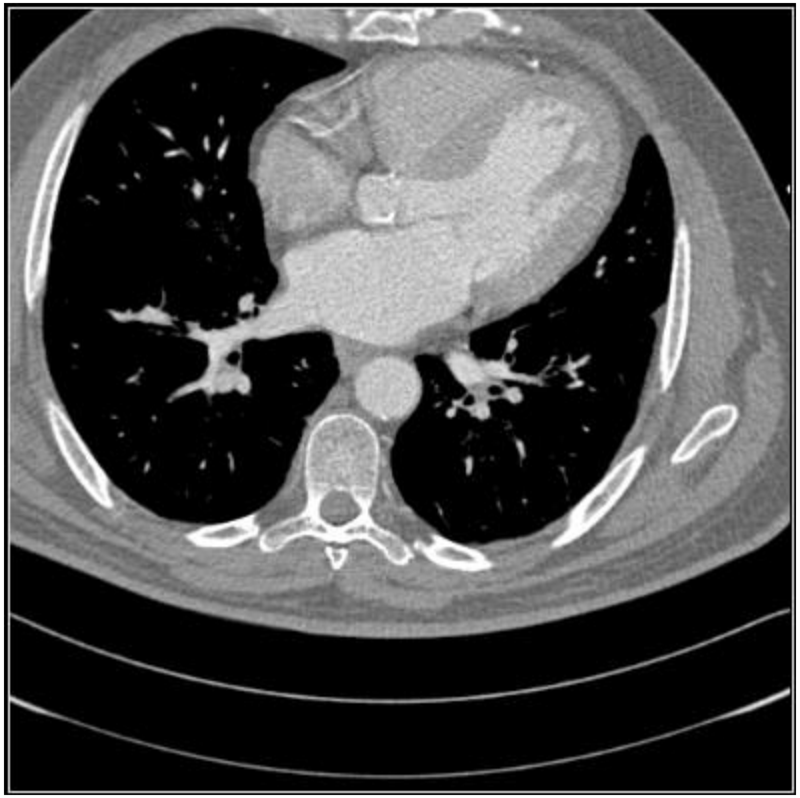
Sobel filter

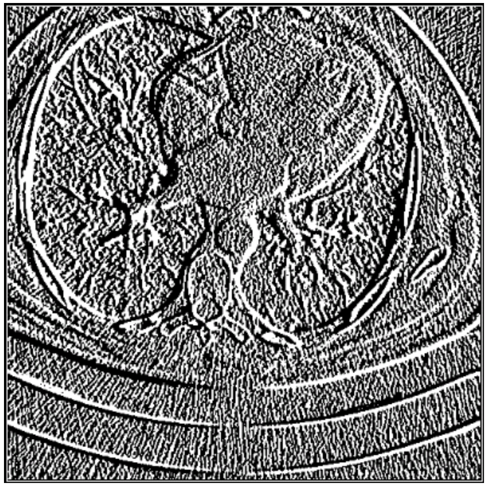
- Sobel-Operator: Combination of the Sobel filters for computing the gradient magnitude M

$$M \approx \sqrt{S_x^2 + S_y^2}$$

- Afterwards: Threshold filter

Example Sobel Filter





Laplace filter

- Edges are zero-crossings in the 2nd derivative
- Laplace-Operator:

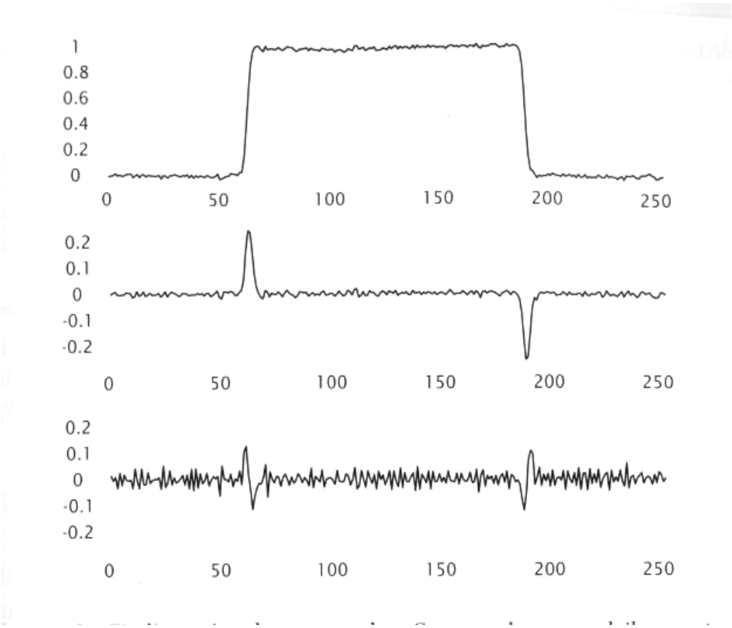
$$\nabla^2 G(x, y) = \frac{\partial^2 G(x, y)}{\partial^2 x} + \frac{\partial^2 G(x, y)}{\partial^2 y}$$

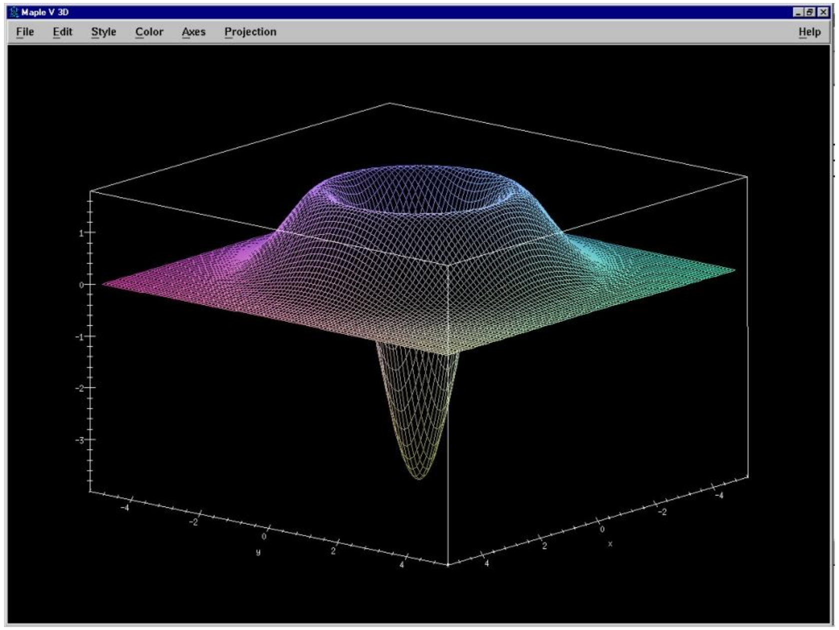
$$LP = \begin{array}{ccc} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{array}$$

- Orientation-independent edges enhancement
- Sensitive to noise (Pseudo-edges)

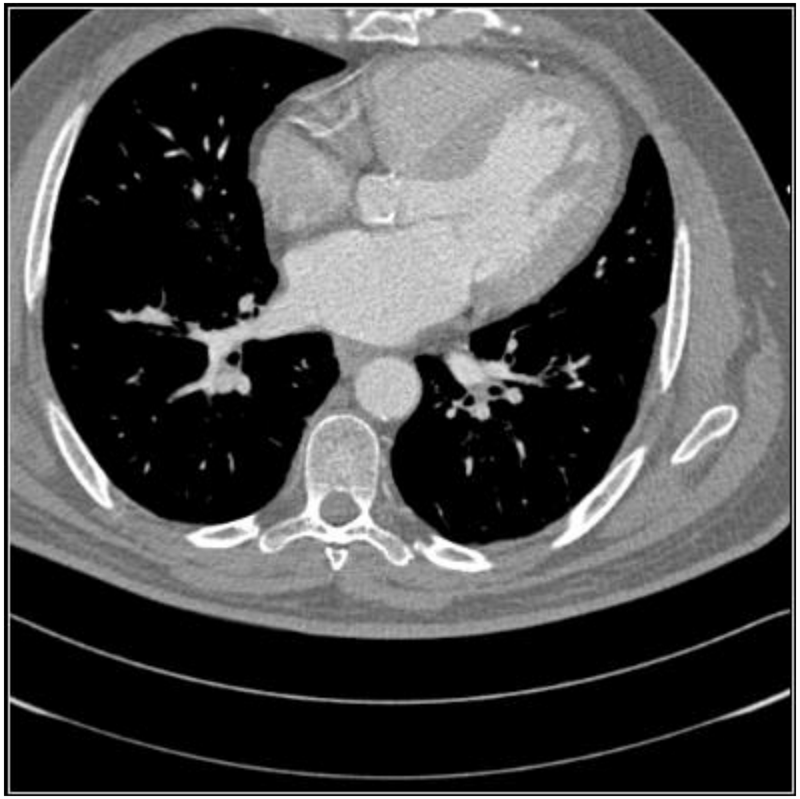
Combination: Laplace of Gaussian (LoG)

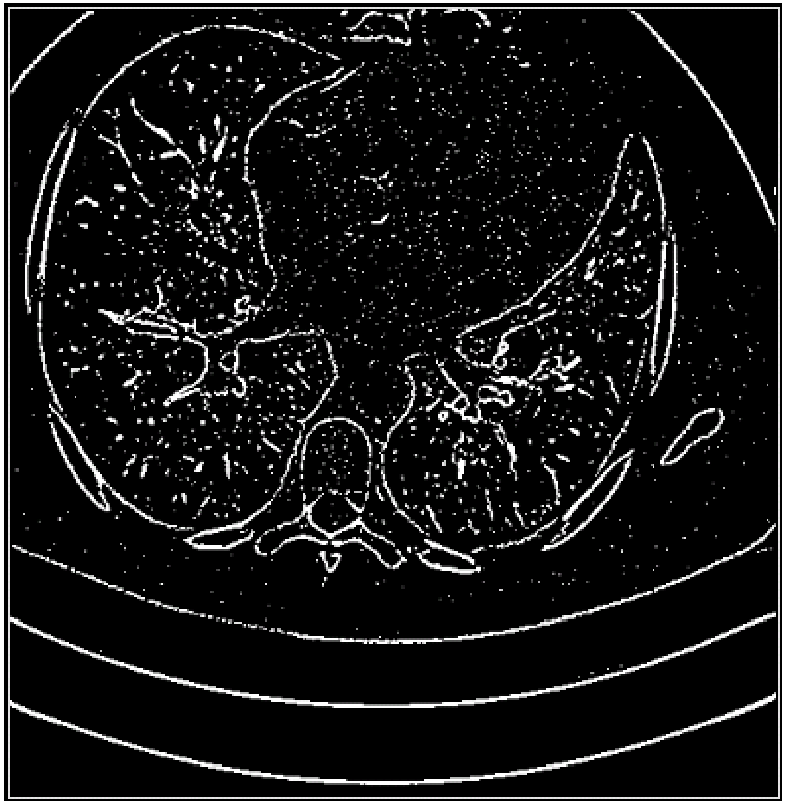
- Problem: 2nd derivative are sensitive to noise, therefore first smoothing using Gaussian filter, then filtering with Laplace filter





Example LoG





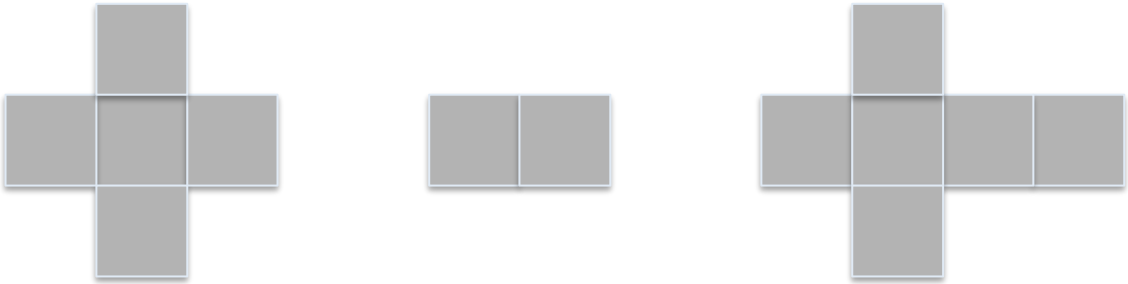
Morphological Operations

- Morphology = Study of shapes
- Morphological operations = binary neighborhood operations for changes of areas with structuring elements

- Application on binary images:
  - Removing stand-alone pixels
  - Removing single, thin lines
  - In a white object, a few pixels remained black

Morphological Operations

- Principle: A structuring element/mask is moved over the binary image



- Basic operations
  - Dilatation: Enlarge object
  - Erosion: Shrink object

Erosion

- Move mask H over the image B
- For each position, test if H is a subset of B

0	0	0	0	0
0	0	0	0	0
1	1	1	0	0
1	1	1	0	0
0	1	1	0	0

1 x	1

Mask H

Erosion →

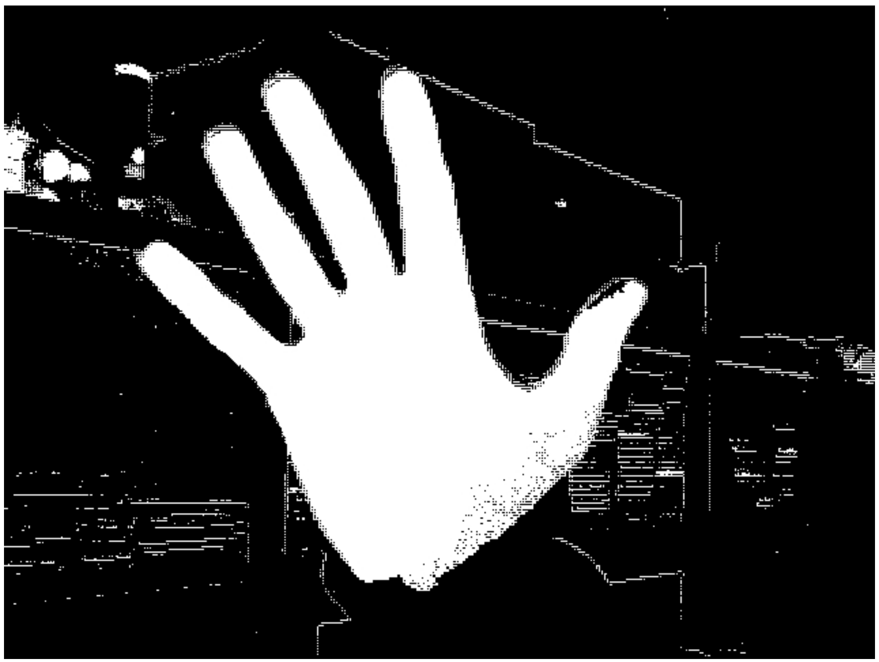
0	0	0	0	0
0	0	0	0	0
1	1	0	0	0
1	1	0	0	0
0	1	0	0	0

- Results depend on the size of the structuring element
- Objects shrink
- Thin lines disapear

NCT

Morphological Operations

- Example application of erosion



Input

Output

Dilatation

- Move mask H over the image B
- For each position, test if the intersection of H and B is not empty

0	0	0	0	0
0	0	0	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0

1 x	1

Mask H

$\xrightarrow{\text{Dilatation}}$

0	0	0	0	0
0	0	0	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0

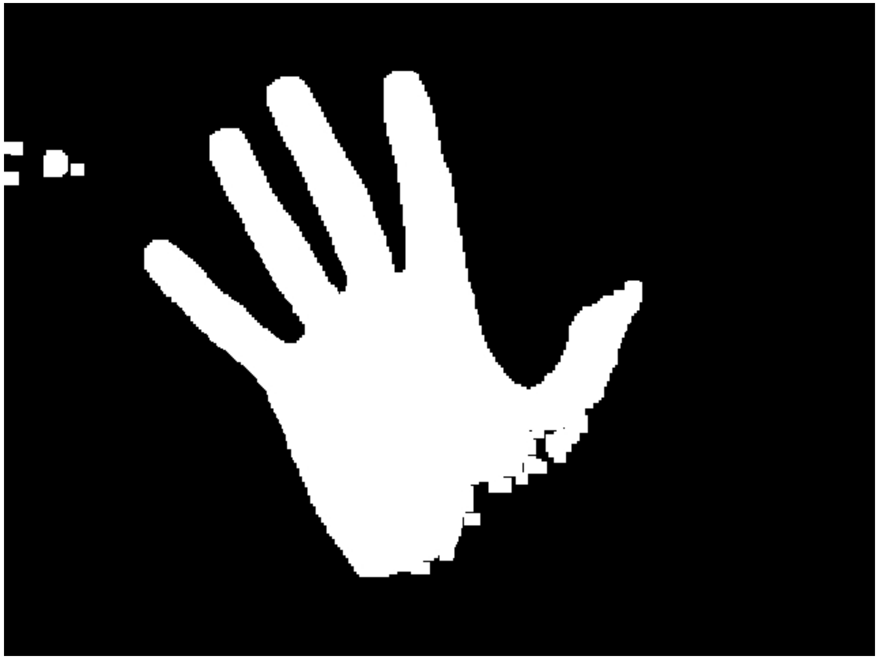
Results depend on the size of the structuring element

- Objects are enlarged/connected
- Holes are closed

Morphological Operations

- Example application of dilatation

Input



Output

Opening and Closing

- Erosion reduces spread of the objects, dilatation enhances spred

→ Distances are altered

- Avoidance: n-fold combination of erosion and dilatation
- Opening: Erosion followed by dilatation
- Closing: Dilatation followed by erosion

Opening

- „Extremities“ of the objects are eliminated
- Thin connection are removed
- Small structures are removed

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	1	1	1	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	0	0	1	1	1	0

1
1

Mask H

Opening

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	1	1	1	0
0	1	1	1	0	0	1	1	1	0
0	1	1	1	0	0	1	1	1	0

Closing

- Gaps in the perimeter are filled
- Close objects are connected
- Number of elements is only changed slightly

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	0	0	1	1	1	0
0	1	1	1	1	1	1	1	1	0

1
1

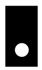
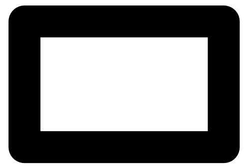
Mask H

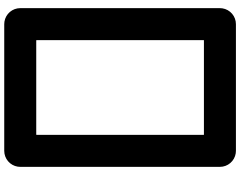
Closing →

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	0

NCT

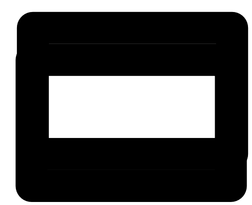
What is the result of this Closing opertion on the following image?

Mask  Image 

- A: 

C: 

- B: 

D: 

Summary – Computer Vision II

- Characteristics of images

- Point operations

- Local operations
  - Smoothing filters
  - Edge filters
  - Morphological operations

Literature

- Lehmann et al.: „Bildverarbeitung für die Medizin“
- Jähne: „Digitale Bildverarbeitung“
- Sonka et al. „Image Processing, Analysis and Machine Vision“
- OpenCV: opencvlibrary.sourceforge.net
- http://saravananthirumuruganathan.wordpress.com/2010/04/01/introduction-to-mean-shift-algorithm/
- Paris et al: „A gentle introduction to bilateral filtering and its applications”