

Image Processing and Imaging

Image Enhancement

Dominik Söllinger
Fachbereich Computerwissenschaften
Universität Salzburg

Winter term 2022/23



1 Spatial Domain Methods

2 Contrast Manipulation & Modification

- Changing the Amplitude
- Contrast Modification
- Histogram Modification
- Histogram-Equalisation
- Explicit Histogram Specification
- Gamma Correction

3 Image Smoothing & Denoising

- Neighbourhood Averaging
- Median Filtering

4 Image Sharpening

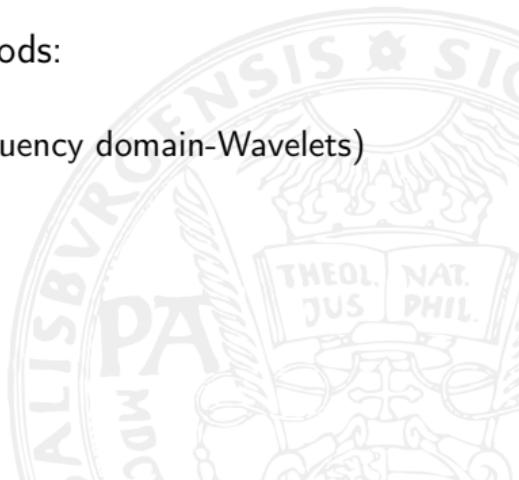
5 Transformation-Based Techniques

- Fourier Transform
- Filtering in Frequency Domain
- Wavelet Transformation
- Fourier vs. Wavelet
- Further Wavelet Transform variants



- Aim is to pre-process images in order to make them better suited for specific applications
- Applications might include human viewing but this is not the most important one
- More important is the preparation for subsequent image processing operations

- Two different application domains of the enhancement methods:
 - spatial domain methods (Bildraum)
 - transform domain (e.g., frequency domain-Fourier, time-frequency domain-Wavelets)



Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



- $f(x, y)$ is the intensity function of the original image
- $g(x, y) = T(f(x, y))$ is the *enhanced* image
- T represents an operator applied to $f(x, y)$ in a specific neighbourhood of (x, y) :
 - often, a squared image region centered in (x, y)
 - center of this image region is moved from pixel to pixel

Simplest case:

- 1×1 neighbourhood, i.e. g only depends on the value of f at position (x, y) ab
- T is called **grey-scale transformation** or transfer function, represented as $s = T(v)$ with $v = f(x, y)$ and $s = g(x, y)$

In the figure, the x-axis shows the original grey-scales, while the y-axis shows the values after transformation.

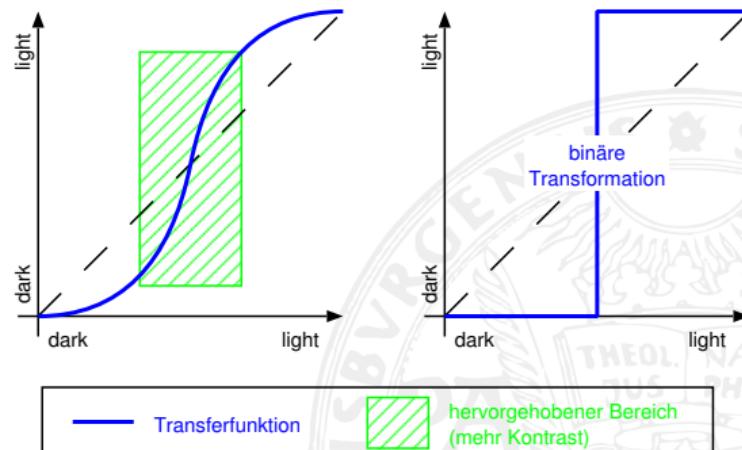


Figure: Grey-scale transformation

Larger neighbourhood:

- Different types of functions are often denoted as **masks**, templates, windows or filter
- Usually a small 2-D array (e.g. 3×3 pixel) is shifted across the image
- Computing the enhanced value at each pixel position
- Coefficients in the array are chosen as to emphasize or suppress certain image properties

Example:

- Image with constant intensity with isolated pixels exhibiting different intensity ("pop noise")
- Mask is $w_i = -1 \quad i = 1, \dots, 9$ except for $w_5 = 8$
- Each entry of the mask is multiplied with the pixels positioned below the entry and all the results are added up
- For an area of constant intensity we get 0 as a response
- Mask is shifted across the image pixel by pixel

Spatial Domain Methods (3)

Result $\left\{ \begin{array}{ll} = 0 & \text{all pixels identical} \\ > 0 & \text{central pixel is larger than surrounding} \\ < 0 & \text{central pixel is smaller than surrounding} \end{array} \right.$

	...	$x - 1$	x	$x + 1$...
$y - 1$...	o	o	o	...
y	...	o	x	o	...
$y + 1$...	o	o	o	...

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figure: Image and mask

$$T[f(x, y)] = w_1 f(x - 1, y - 1) + w_2 f(x, y - 1) + w_3 f(x + 1, y - 1) + \dots + w_9 f(x + 1, y + 1)$$

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Outline

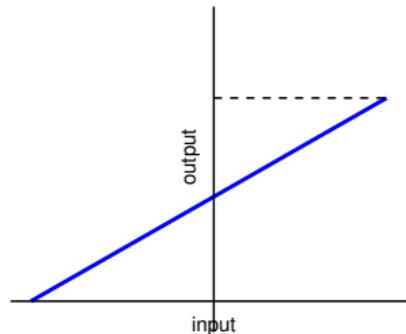
- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Changing the Amplitude

Changing the range of grey-scales:

- Visualisation of difference images after prediction or MC
- Clipping is often used in case of a small number of pixels is found at the tails of the histogram
 - Contrast is improved additionally



z.B. Visualisierung, Differenzbildung

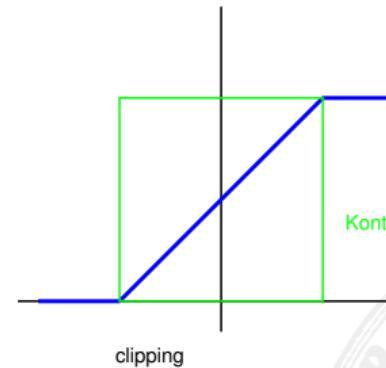


Figure: Modification of grey-scale range

Local vs. global: All techniques discussed here can be applied to the entire image - globally - or to parts / tiles of the image - locally.

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Contrast Modification

Contrast is increased in areas where the slope of the transfer function (or its tangent) is larger than 1.

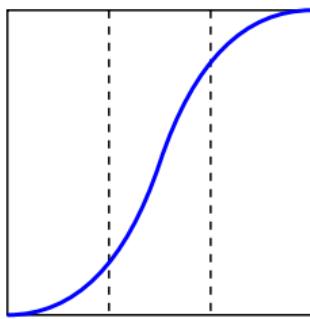


Figure: Contrast Modification

Example: Contrast modification of a computer tomography by applying the logarithm as transfer function

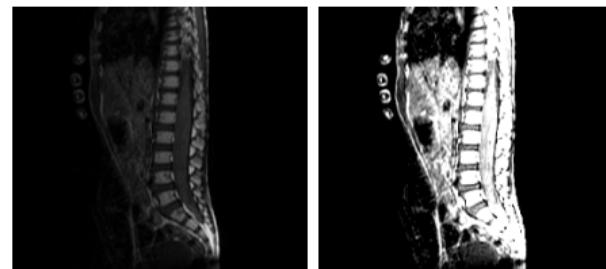


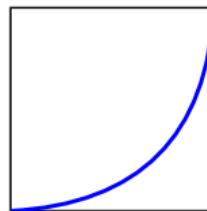
Figure: Contrast modification of a CT

Contrast Modification - Transfer Functions

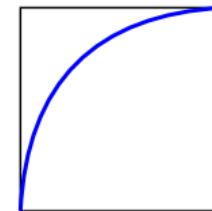
Contrast can be modified by using simple transfer functions like

$$s = r^p \quad p = 2, 3, 1/2$$

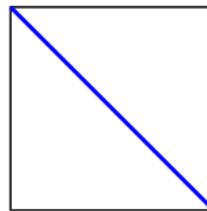
Further typical contrast modification techniques:



$$s = r^2$$



$$s = r^{1/2}$$



$$\text{reverse function } s = 1 - r \quad \text{inverse function}$$

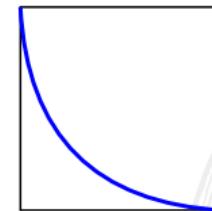


Figure: Typical contrast modification techniques

Contrast Modification - Myelin Example

A further example is displayed applying the logarithm function to a Myelin image (similar $s = r^{1/2}$).

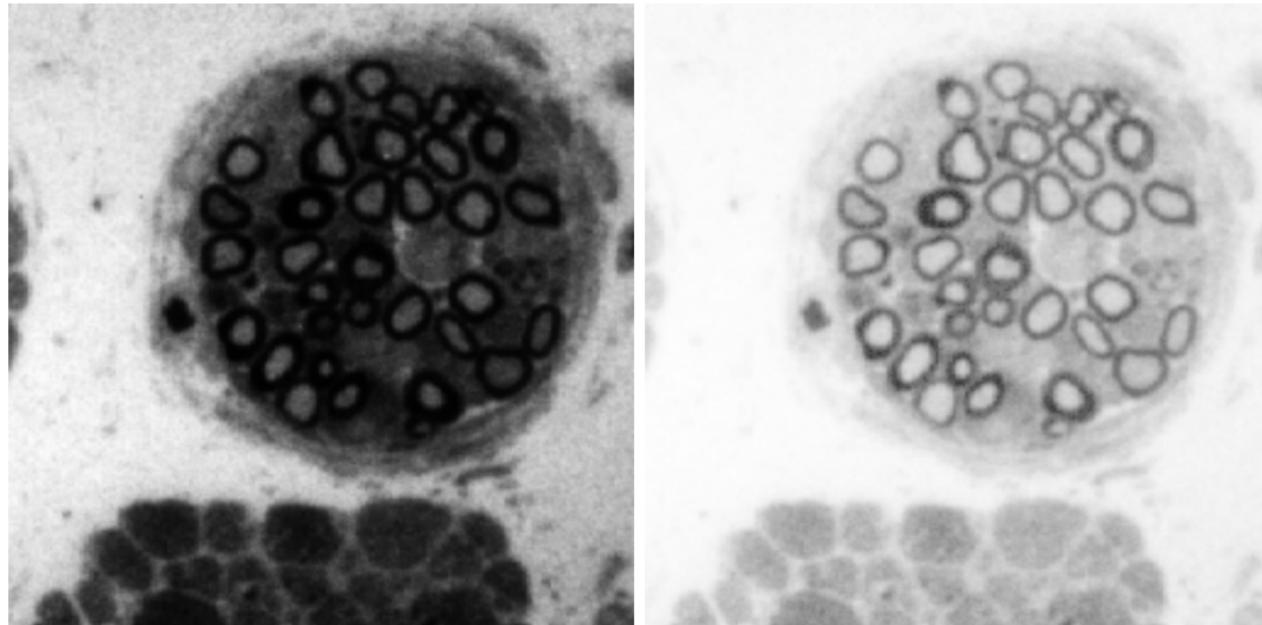


Figure: Contrast modification of a Myelin

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Histogram

Distribution of relative frequency of grey-values in an image (global image description).

Let r be the grey-value of a pixel and $0 \leq r \leq 1$ with $r = 0$ = black and $r = 1$ = white.

We consider the transfer function $s = T(r)$ with the properties

- (a) T is monotonically increasing on $(0, 1)$
- (b) $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$

The inverse transform from s to r is $r = T^{-1}(s)$ with $0 \leq s \leq 1$ with identical properties (a) and (b).

- Consider the grey-values as being continuous random variables
- We can represent the original and transformed grey-value distributions by considering their corresponding density functions $p_r(r)$ and $p_s(s)$
- Density functions describe the overall impression of the image

Histogram Modification (2)

Remark: These density functions may be interpreted as a *continuous histogram*.

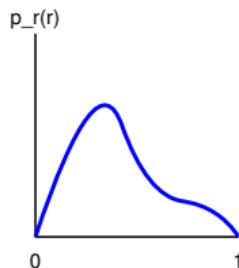


Figure: Continuous histogram

From elementary probability theory and statistics we know:

- If we know $p_r(r)$, $T(r)$
- And $T^{-1}(s)$ satisfies the condition (a)
- Then the density function of the transformed grey-values is given by:

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \quad [\text{see proof}] \quad (1)$$

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - **Histogram-Equalisation**
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Histogram-Equalisation (1)

We consider the following transfer function (called cumulative distribution function):

$$s = T(r) = \int_0^r p_r(w)dw \quad 0 \leq r \leq 1$$

When computing the derivative with respect to r we result in (fundamental theorem of calculus):

$$\frac{ds}{dr} = p_r(r) \quad (2)$$

When inserting equation (2) into equation (1)
we get:

$$p_s(s) = \left[p_r(r) \frac{1}{p_r(r)} \right]_{r=T^{-1}(s)} = 1 \quad 0 \leq s \leq 1. \quad (3)$$

The result is a uniform density, constant 1. This result is entirely independent of the inverse function.

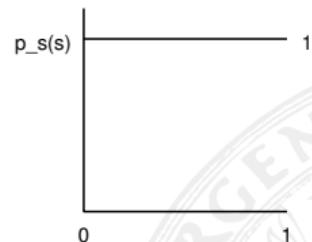


Figure: equalised histogram

Histogram-Equalisation (2)

- Histogram equalisation is achieved by applying the cumulative distribution function (CDF) as grey-value transfer function
- In the equalised histogram all occurrence probabilities are equal to 1
- Attention: here we are in the – *idealised* – continuous case!
- Examples of CDF and corresponding densities:

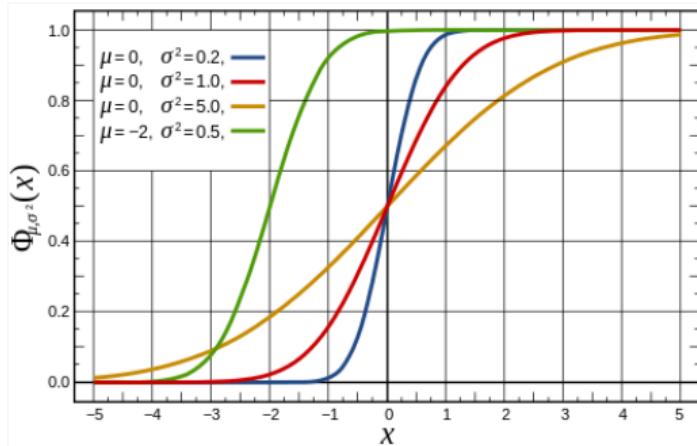
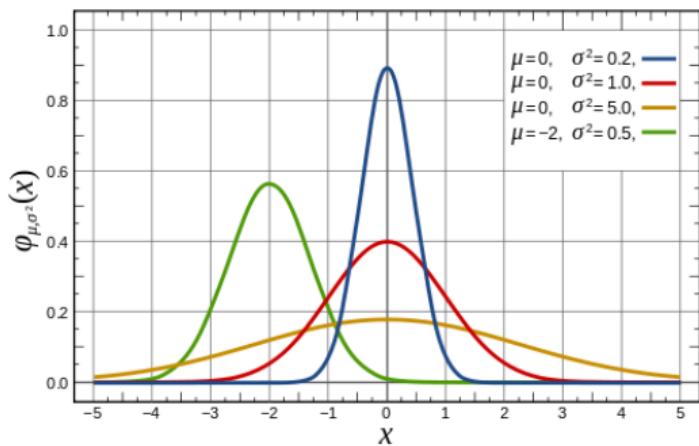


Figure: Gaussians and their cumulative distribution functions

Histogram-Equalisation Example

Example:

$$p_r(r) = \begin{cases} -2r + 2 & 0 \leq r \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$$s = T(r) = \int_0^r (-2w + 2)dw = -r^2 + 2r$$

$$r = T^{-1}(s) = 1 - \sqrt{1-s}$$

Problem: For natural images no “Function” $p_r(r)$ exists.

Discretisation

$$p_r(r_k) = \frac{n_k}{n} \quad 0 \leq r_k \leq 1, \quad k = 0, 1, \dots, L - 1$$

n_k ... number of occurrence of grey-scale k

n ... number of pixel

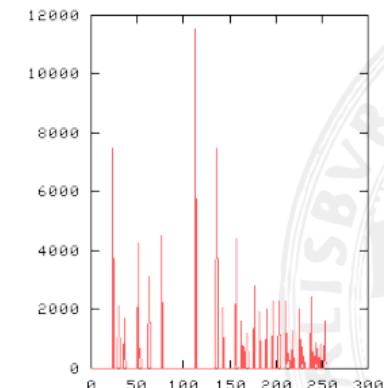
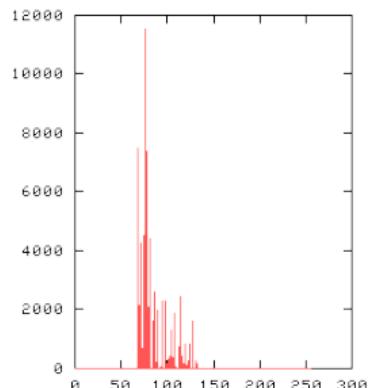
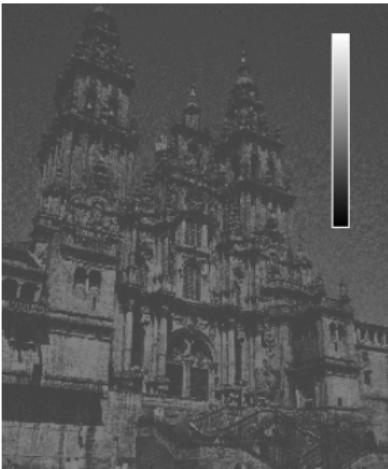
L ... number of grey-scales

Discrete Equalisation

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j) \quad k = 0, \dots, L - 1, \quad r_k = T^{-1}(s_k)$$

Remark: The inverse function is not required. $T(r_k)$ can be derived from pixel statistics.
Due to discretisation the result is an approximation only.

Discrete Histogram-Equalisation Example



Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Explicit Histogram Specification (1)

- Let $p_r(r)$ be the original and $p_z(z)$ the target density function
- First step - the original image is histogram equalised:

$$s = T(r) = \int_0^r p_r(w)dw$$

- Assuming the target image to be available, it could be histogram equalised as well:

$$v = G(z) = \int_0^z p_z(w)dw$$

- $z = G^{-1}(v)$ would result in the target pixel values.
- $p_s(s)$ and $p_v(v)$ have identical uniform densities (gleichmäßige Dichte)
- It is possible to use s (from the equalised original) instead of v in the inverse process
- $z = G^{-1}(s)$ exhibits the desired target density

Explicit Histogram Specification (2)

Procedure:

- 1 Equalise original image $\rightarrow s$
- 2 Specify the desired target density and obtain $G(z)$
- 3 $z = G^{-1}(s) \rightarrow z = G^{-1}(T(r))$

Problem: The inverse function cannot be computed directly in the discrete case. Thus, the inverse function is obtained by a mapping grey-scale to grey-scale (table lookup).

Application: Optimisation for specific output devices, for which the optimal target histogram is known, e.g. for large plotters etc.

Remark: Techniques discussed so far can also be applied to $n \times m$ neighbourhoods – in case it is only a specific region which is of interest, this leads to better results.

- Global histogram equalisation works well when the distribution of the pixel values (i.e. the histogram) is similar throughout the image
- If the image contains areas which are significantly lighter or darker than the overall histogram, the contrast of those regions will not be sufficiently enhanced.

Adaptive histogram equalisation (**AHE**):

- Transforms each pixel with a transformation function derived from the pixel's neighbourhood
 - Can be a fixed square, can be more involved, the computation may be weighted, etc.
- CDF computed from pixels in the neighbourhood is used
- In case the neighbourhood is a very homogeneous area:
 - Histogram will be very peaked and the transformation function will map a narrow range of pixel values to the whole range of the result image
 - Resulting in an over-amplification of **noise**

Contrast Limited Adaptive Histogram Equalisation **CLAHE**:

- Contrast is limited in each neighbourhood
- Slope of the transformation function determines the contrast amplification
- This slope is proportional to the slope of the CDF (which is locally proportional to the histogram value of the pixel)
- Histogram is clipped at some predefined histogram value (see figure)
- Limits the slope of the contrast enhance CDF and thus, the amount of contrast enhancement
- Clip-value is often chosen to be 3 times the grey mean value
- Due to intensity / luminance loss, it is better to redistribute the lost parts to the other histogram bins

Contrast Limited Adaptive Histogram Equalisation (CLAHE) (2)

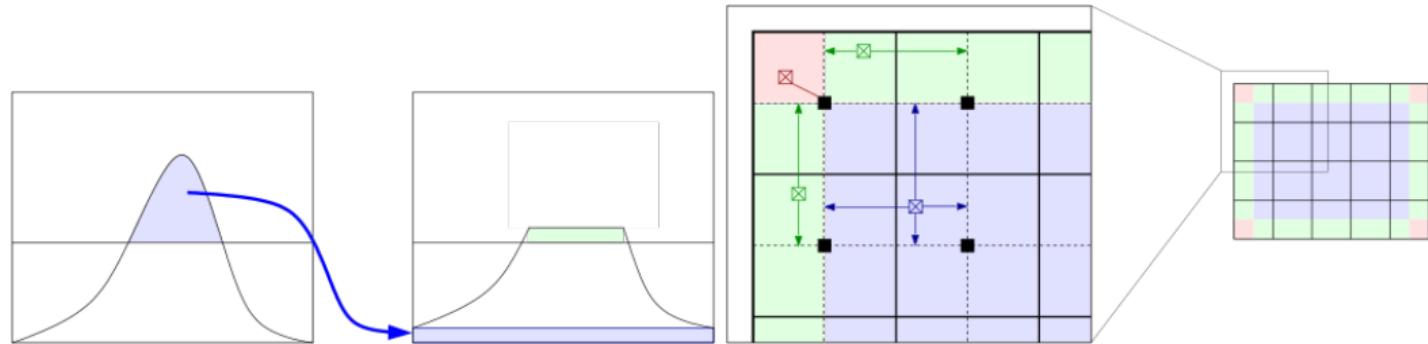
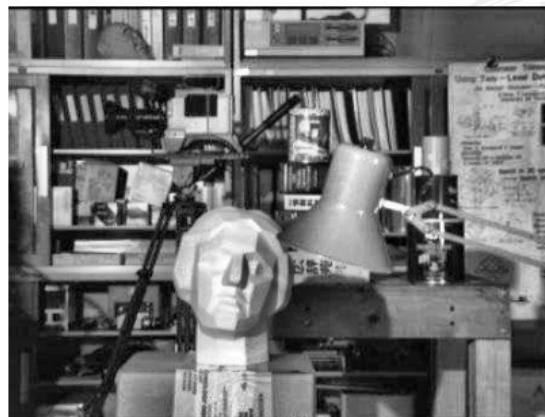
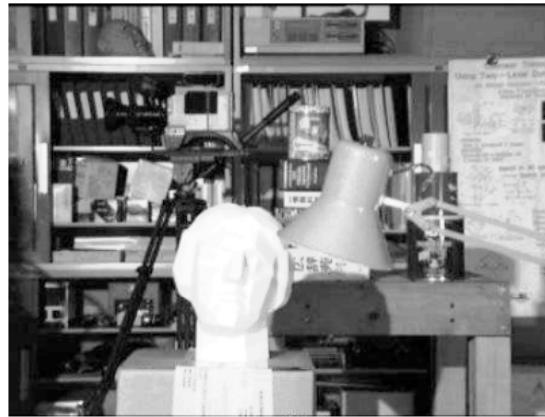
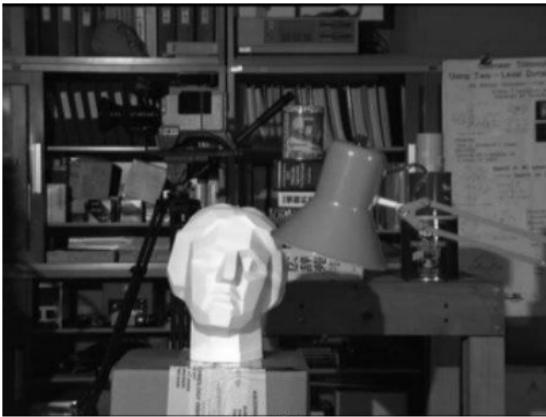


Figure: CLAHE: histogram clipping and interpolation

- Computation of CLAHE involves the determination of the transformation function at each pixel
- T is usually approximated only, by computing transformation functions for fixed tiles of an image grid
- Actual output pixel for a specific location is then computed using up to four transformation functions and appropriate bilinear or linear interpolation techniques

Histogram Equalisation - AHE - CLAHE Comparison



Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



- Brightness/intensity values in digital images are on a linear scale
- Perceived brightness/intensity by the human eye increases faster in dark areas and slower in bright areas
- Following an exponential function → non-linear
- Another reason: CRT screens
- Brightness is roughly correlated to U^2 (voltage of the electron tube)
- Again, the linear scale of the brightness values in images does not fit the non-linear scale of the output device
- TV stations applied a correction to the video material so that no correction in the TV set was necessary
- We need something to correct this in digital images / viewing devices

Gamma Correction - Gamma Function

Can be achieved using Gamma correction:

- Tries to compensate the different characteristics of capture and viewing devices
- Consistent impression of images on different devices
- Gamma originates from analog film - relation between the incoming light intensity and the level of blackening on the film material:
 - Slope between H and D in the linear area is denoted as Gamma

Gamma Function:

$$b = f_\gamma(a) = a^\gamma \text{ for } a \in \mathbb{R}, \gamma > 0$$

- γ is denoted as Gamma value
- If a is between $[0, 1]$, f_γ stays within $[0, 1]$ as well, independent of γ
- Function always passes $(0, 0)$ and $(1, 1)$
- Can be controlled by one parameter only:

γ

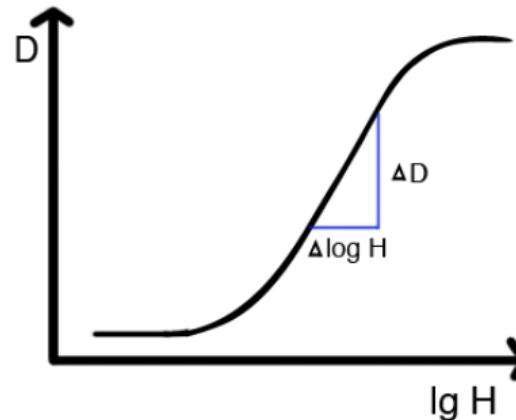


Figure: Relation between exposure (H) and resulting density (D) on the film material

Gamma Correction - Transfer Function

- Gamma function $f_\gamma(a)$ can be inverted
- Result is a Gamma function $f_{\bar{\gamma}}(b)$ with $\bar{\gamma} = \frac{1}{\gamma}$
- Typical Gamma values: 1.8-2.8 for CRT screens, 2.2 for NTSC TV sets and 2.8 for PAL, 0.45 for capturing devices (1/2.2)
- Camera with known transfer characteristic: $o = I^{\gamma_c}$, o...Output, I...light intensity

Gamma Correction:

- To compensate, we want to have an i which is proportional to I
- Inverse Gamma function is used (as transfer function) with $\bar{\gamma}_c = \frac{1}{\gamma_c}$:

$$i = f_{\bar{\gamma}_c}(o) = o^{1/\bar{\gamma}_c}$$

- The result is then:

$$i = o^{1/\bar{\gamma}_c} = (I^{\gamma_c})^{1/\bar{\gamma}_c} = B^{\gamma_c \frac{1}{\bar{\gamma}_c}} = B^1$$

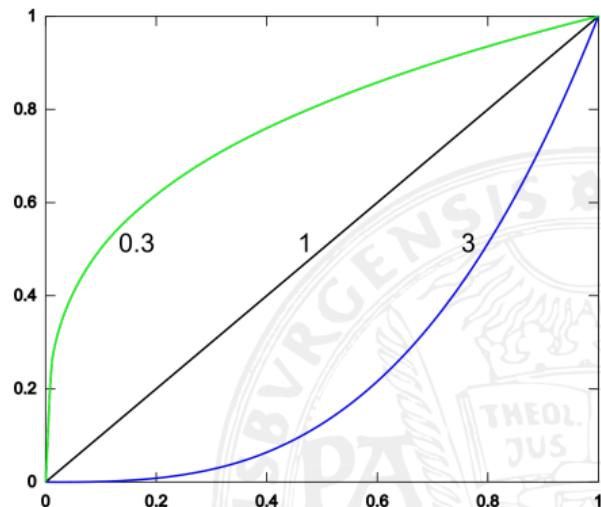


Figure: Gamma curve and inverse Gamma curve

- We implicitly assumed that all values are within $[0, 1]$
- Usual pixel values are between $[0, 255]$

Steps to perform a Gamma correction on images:

- 1 Map pixel values a linear to $\hat{a} \in [0, 1]$
- 2 Apply the Gamma function to \hat{a} : $\hat{b} = f_\gamma(\hat{a}) = \hat{a}^\gamma$
- 3 Map \hat{b} back linear to $b \in [0, 255]$

Problem:

- For γ values < 1 , there is a steep slope around 0 (see Fig.)
- Usually γ values < 1 are used during compensation (e.g. 0.45 for NTSC cameras)
- Results in undesired amplification of noise in pixel values close to zero

Gamma Correction - Modified Version

Practical solution: Replace the Gamma function with a linear one around 0 ($0 \leq a \leq a_0$):

$$\bar{f}_{(\gamma, a_0)}(a) = \begin{cases} s \cdot a & 0 \leq a \leq a_0 \\ (1 + d) \cdot a^\gamma - d & a_0 \leq a \leq 1 \end{cases}$$

with

$$s = \frac{\gamma}{a_0(\gamma - 1) + a_0^{(1-\gamma)}} \text{ and } d = \frac{1}{a_0^\gamma(\gamma - 1) + 1} - 1$$

- For an optimal approximation, small values of a_0 should be used
- Typical values for ITU (International Telecommunications Union) standard:
 - $\gamma = \frac{1}{2.222} \approx 0.45$ and $a_0 = 0.018$
 - Values for the sRGB standard are similar

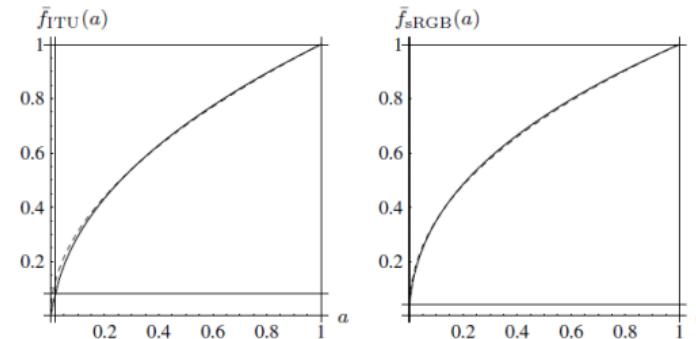
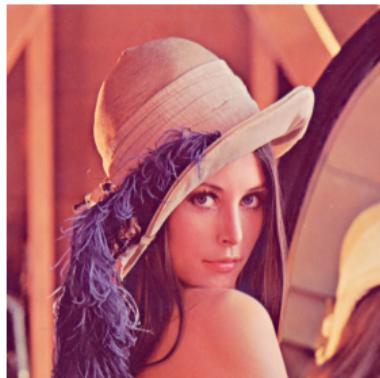
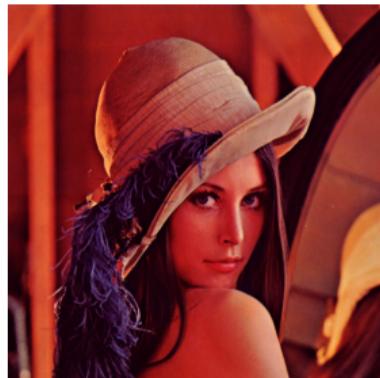


Figure: Left: ITU Gamma correction, right: sRGB Gamma

Gamma Correction - Examples



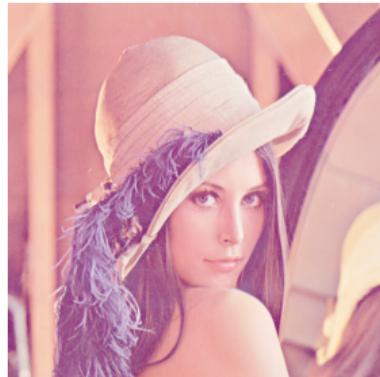
original image



$\gamma = 2.22$



$\gamma = 0.64$



$\gamma = 0.45$

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants

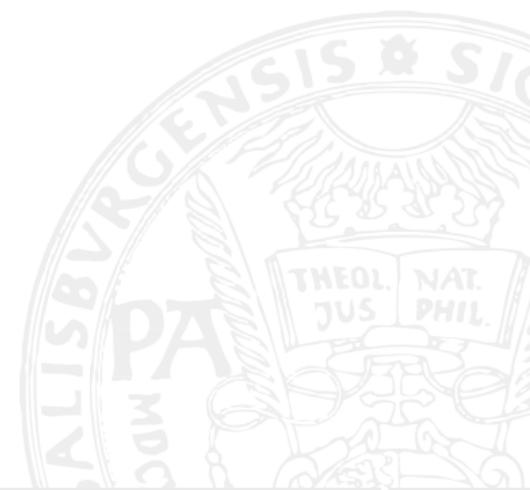


Image Smoothing & Denoising - Neighbourhood Averaging

Aim: Effects caused by transmission errors or sampling errors should be corrected. These effects are **local errors** (in the ideal case single independent pixels).

Most popular technique: *Neighbourhood Averaging*

$g(x, y)$ is obtained by computing averages in a neighbourhood S ($M \dots$ number of pixels in S):

$$g(x, y) = \frac{1}{M} \sum_{(n,m) \in S} f(n, m)$$

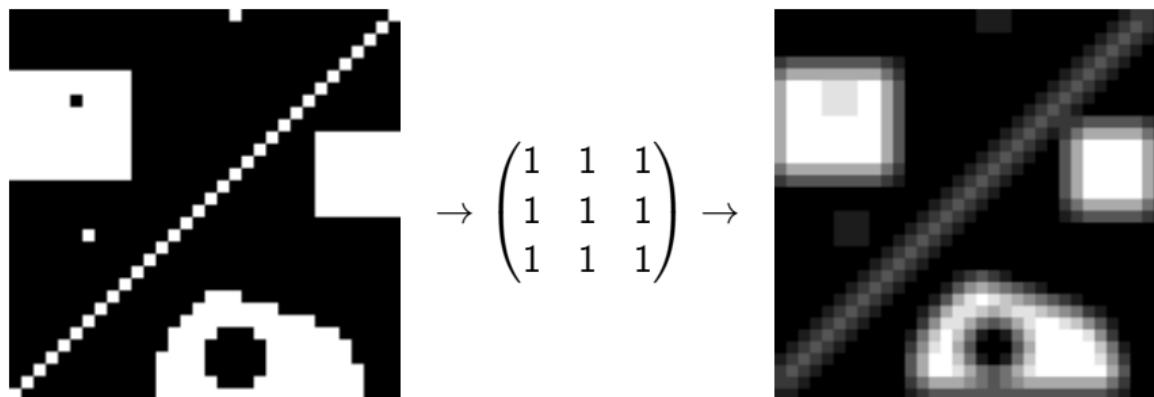


Figure: Averaging with 3×3 mask

Problem: Edges get significantly softened (*blurring*)

- Can be handled by applying thresholding (with threshold T)
- If the difference between original and “enhanced” pixel value is too large:
- Averaging is avoided and the original value is set

$$\hat{g}(x, y) = \begin{cases} g(x, y) & |f(x, y) - g(x, y)| < T \\ f(x, y) & \text{otherwise} \end{cases}$$

Outline

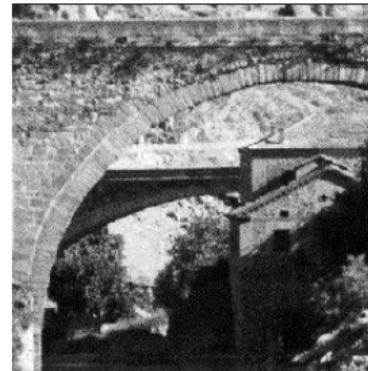
- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



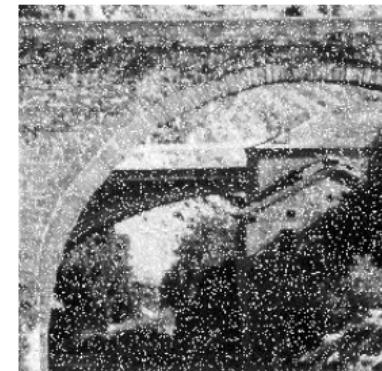
Median Filtering

Median is used:

- Instead of computing an average in the neighbourhood
- Statistical outliers in the neighbourhood are not included in the generation of the enhanced pixel value
- Especially for denoising (e.g. pop noise) the median-based approach is often preferable



Original



Noisy image



5 × 5 Averaging



5 × 5 Median

Image Filtering / Convolution Example

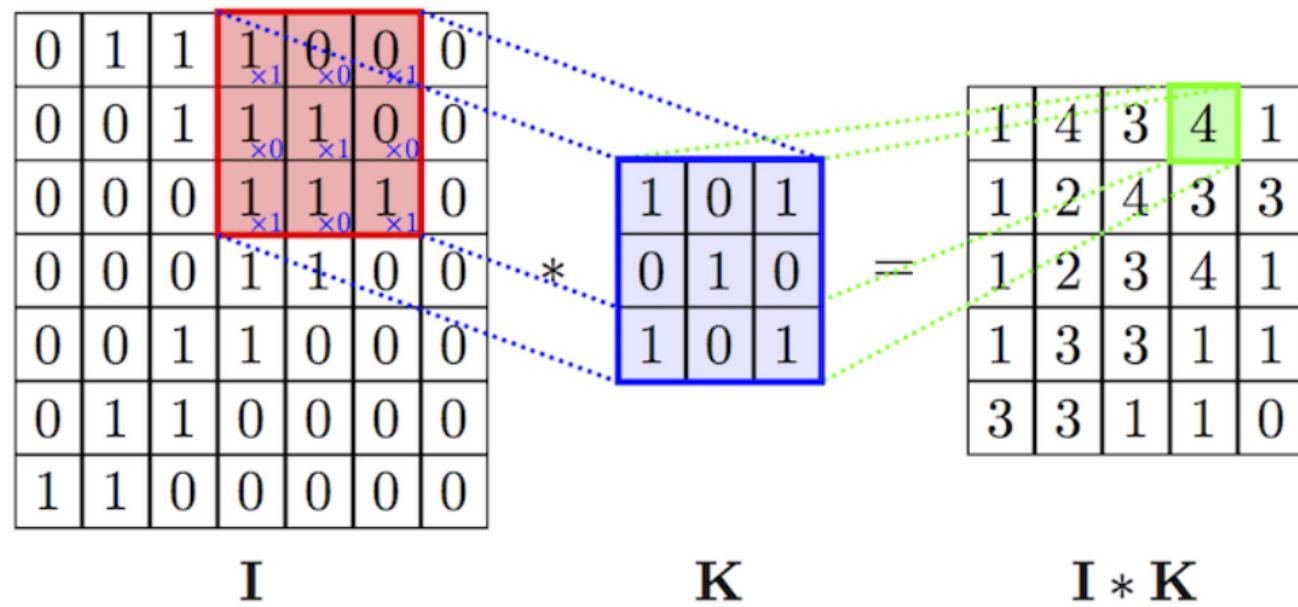


Figure: Image Convolution with 3x3 kernel example

Image Filtering / What about the Boundaries? (1)

Problem with the previous example: Filtered image is smaller than the input image

In general: output should be the same size as the input

- Filter kernel has to be placed at the image boundaries as well
- Which values should be used for the kernel elements “outside” the image?
- Several strategies (boundary handling):
 - Padding with constant value (special case: zero padding)
 - Replication (either one line or several lines)
 - Reflection (or mirroring)
 - (Cyclic) wrap (repeat)

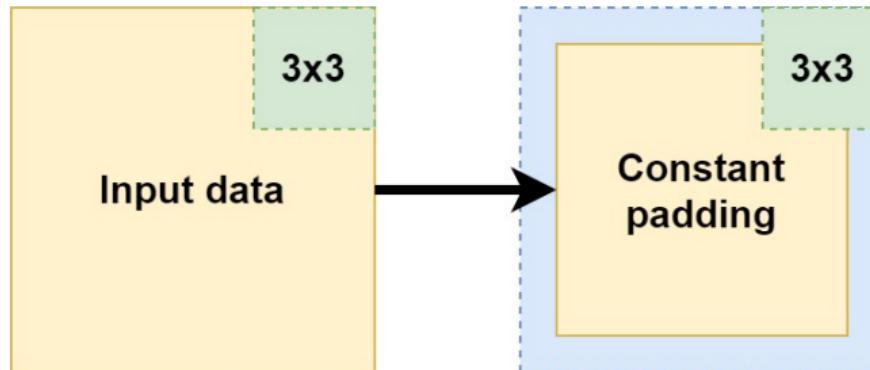


Figure: Boundary Handling - Constant padding

Image Filtering / What about the Boundaries? (2)

3	5	1
3	6	1
4	7	9

No padding

5	3	3	5	1	1	1	5
5	3	3	5	1	1	1	5
6	3	3	6	1	1	6	
7	4	4	7	9	9	7	
7	4	4	7	9	9	7	

(1, 2) replication padding

Figure: Boundary Handling - Replication Padding

3	5	1
3	6	1
4	7	9

No padding

1	6	3	6	1	6	3
1	5	3	5	1	5	3
1	6	3	6	1	6	3
9	7	4	7	9	7	4
1	6	3	6	1	6	3

(1, 2) reflection padding

Figure: Boundary Handling - Reflection Padding

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Image Sharpening

Aim: emphasize edges

Idea: difference among pixels suggests the existence of an edge

Averaging and sharpening are based on two antagonistic mathematical concepts:

- Averaging: details are “integrated”
- Sharpening: details are “differentiated”

$$\text{Gradient } G[f(x, y)] = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

1 G points into the direction of the largest growth of $f(x, y)$

$$2 |G[f(x, y)]| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \sim \text{mag}(G)$$

$\text{mag}(G)$... magnitude of f , is equal to the largest growth rate of $f(x, y)$

In image processing $\text{mag}(G)$ is often denoted as Gradient for simplicity.

Discretisation: derivatives are approximated by differences

$$|G[f(x, y)]| = \sqrt{[f(x, y) - f(x + 1, y)]^2 + [f(x, y) - f(x, y + 1)]^2}$$

As an alternative, absolute values can be used instead of the square root (more efficient implementation).

There are several possibilities how to visualise the *Gradient image* $g(x, y) = |G[f(x, y)]|$:

$g(x, y) = G[f(x, y)]$	$g(x, y) = \begin{cases} G[f(x, y)] & G \geq S \\ f(x, y) & \text{otherwise} \end{cases}$
$g(x, y) = \begin{cases} T_{\text{otherwise}} & G \geq S \\ f(x, y) & \text{otherwise} \end{cases}$	$g(x, y) = \begin{cases} T_1 & G \geq S \\ T_2 & \text{otherwise} \end{cases}$

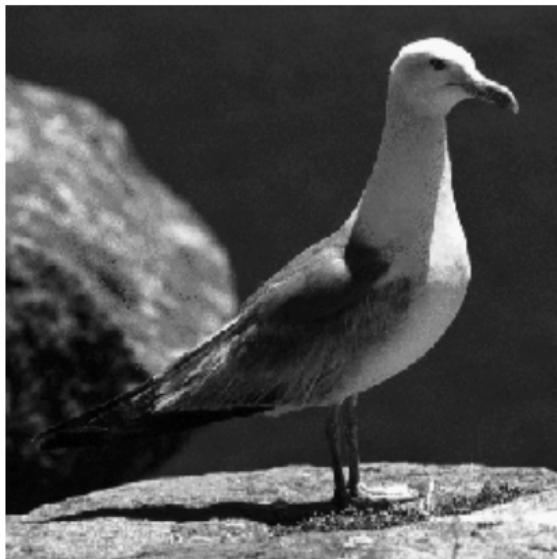
Figure: Types of Gradient visualisation

Image Sharpening - Roberts Operator

Roberts Operator:

$$|G[f(x, y)]| = \max \{|f(x, y) - f(x + 1, y + 1)|, |f(x + 1, y) - f(x, y + 1)|\}$$

Overall, the value of the Gradient is proportional to the difference among pixels grey-values – large values for edges, small values for smooth or uniform areas.



Original



Roberts Gradient image

Image Sharpening - Roberts Operator - Examples of Gradient Visualisation



gradient image



gradient above threshold



value for gradient if above threshold



value for above and below threshold

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Transformations used in image processing are unitary(orthogonal and regular) transformations and are used for:

Feature extraction to describe certain properties in an efficient manner (e.g. frequencies: high - edges, low - luminance). The aim is to be able to conduct certain operations more efficiently in the transformed domain (e.g. denoising).

Compression concentration of information

Efficient calculations e.g. a *dense matrix* is transformed into a *sparse matrix*, since more efficient algorithms do exist for sparse matrices (in sparse matrices – sparsely populated matrices – many coefficients are equal to zero).

In many cases the concept to represent a signal using orthogonal basis functions is used.

Background: Vectors in 2 dimensional space can be represented by a set of orthogonal (i.e. the inner product is zero) basis-vectors (orthogonal basis):

$$(x, y) = \alpha(1, 0) + \beta(0, 1).$$

- $\{(1, 0), (0, 1)\}$ are the orthogonal basis-vectors
- α and β are the coefficients which determine the weight of each basis-vector to represent the vector (x, y)
- Orthogonality of the vectors facilitates a minimal number of basis-vectors

This concept can be generalised to functions and signals, respectively:

$$f(x) = \sum_n \langle f(x), \psi_n(x) \rangle \psi_n(x)$$

- Functions $\psi_n(x)$ are orthogonal basis functions
- $\langle f(x), \psi_n(x) \rangle$ are the transform coefficients which determine the weight of each basis function to represent a given signal “well”
- For an application the coefficients $\langle f(x), \psi_n(x) \rangle$ are computed and processed further
- Basis functions $\psi_n(x)$ are orthogonal \rightarrow required number to represent the signal is minimal

Fourier transform:

- Basis functions are $\psi_n(x) = e^{-\pi i n x} = \cos(nx) - i \sin(nx)$
- Frequencies of periodic signals are considered
- A Fourier coefficient $\langle f(x), \psi_n(x) \rangle$ represents the strength / energy of the frequency n in a signal
- Obviously, not all signals may be represented efficiently using this approach

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Fourier Transform

Developed by French Fourier who was very interested in music (violin) and wanted to know how sounds are created by changing the length of the cords.

Let $f(x)$ be a continuous function,
 $\hat{f}(u)$ is the Fourier transform of $f(x)$
with respect to frequency u .

$$\hat{f}(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} dx \quad (4)$$

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(u) e^{2\pi i u x} du \quad (5)$$

The Fourier transform of a real function is usually of complex values.

$$\hat{f}(u) = \Re(u) + i\Im(u)$$

$$\hat{f}(u) = |\hat{f}(u)| e^{i\Phi(u)}$$

$$|\hat{f}(u)| = \sqrt{\Re^2(u) + \Im^2(u)} \quad \Phi(u) = \tan^{-1} \left(\frac{\Im(u)}{\Re(u)} \right)$$

Inversion can be computed if:

- $f(x)$ is continuous and can be integrated
- $\hat{f}(u)$ can be integrated as well

- $|\hat{f}(u)|^2 \dots$ Power-Spectrum (Spektraldichte)
- $|\hat{f}(u)| \dots$ Fourier-Spectrum (Frequenzspektrum)
- $\Phi(u) \dots$ Phase angle
- $u \dots$ Frequency variable
(since $e^{2\pi i u x} = \cos 2\pi u x + i \sin 2\pi u x$)

Discrete Fourier Transform

Interpret the integral as the summation of discrete terms:

- $\hat{f}(u)$ is composed of an infinite sum of Sine- and Cosine terms,
- With parameter u determining the frequency of the Sine/Cosine pair

Discrete Fourier Transform (DFT):

$\{f(0), f(1), \dots, f(N - 1)\}$ are N uniformly sampled points of a continuous function

$$\hat{f}(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-2\pi i u x} \quad u = 0, \dots, N-1 \quad (6)$$

$$f(x) = \sum_{u=0}^{N-1} \hat{f}(u) e^{2\pi i u x / N} \quad x = 0, \dots, N-1 \quad (7)$$

Two-dimensional:

$$\hat{f}(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i (ux/M + vy/N)} \quad (8)$$

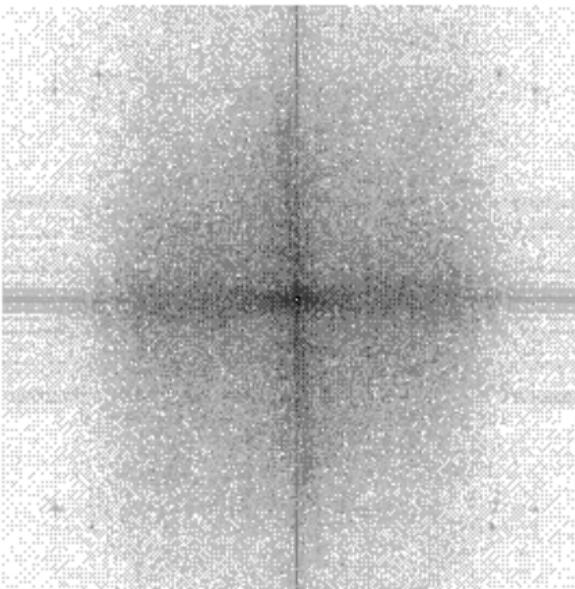
$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \hat{f}(u, v) e^{2\pi i (ux/M + vy/N)} \quad (9)$$

Discrete Fourier Transform - Example

Logarithmic scale: $D(u, v) = \log(1 + |\hat{f}(u, v)|)$ better than $|\hat{f}(u, v)|$ for display purposes, since values decrease rapidly for increasing frequency



a)



b)

Figure: Original image (a) and its Fourier Spectrum (Magnitude) (b)

Discrete Fourier Transform - Example of Signals (Images)

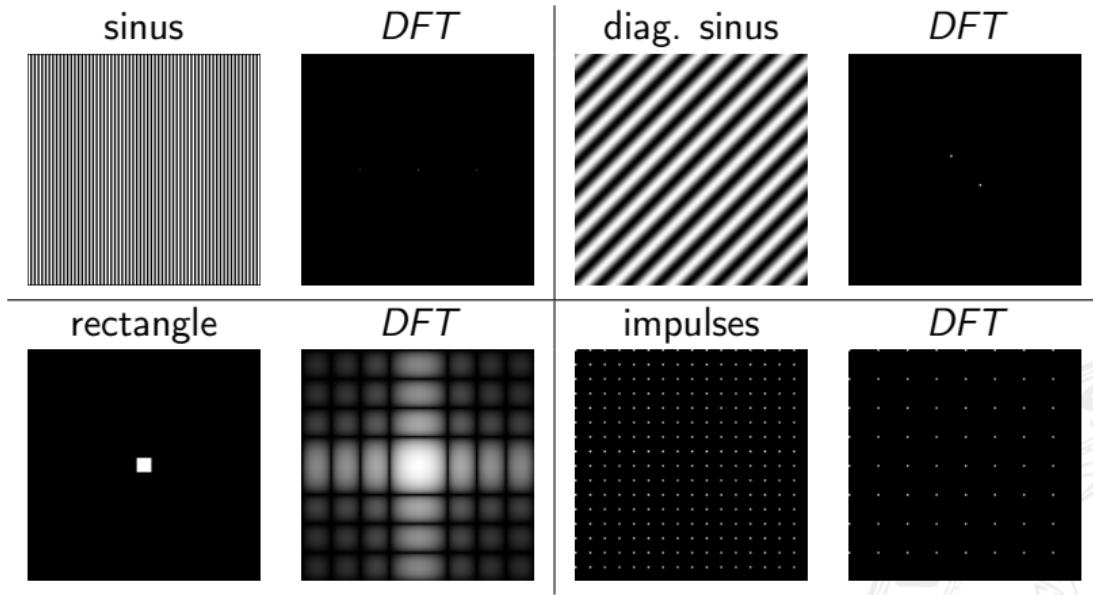


Figure: DFT Transformations

Properties of the 2D Fourier Transform (1)

- $\hat{f}(0,0)$ is identical to the average grey-value of all pixels:

$$\hat{f}(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)$$

- Separability

$$\hat{f}(u,v) = \frac{1}{M} \sum_{x=0}^{M-1} \left(\frac{1}{N} \sum_{y=0}^{N-1} f(x,y) e^{-2\pi i vy/N} \right) e^{-2\pi i ux/M}$$

$$f(x,y) = \sum_{u=0}^{M-1} \left(\sum_{v=0}^{N-1} \hat{f}(u,v) e^{2\pi i vy/N} \right) e^{2\pi i ux/M}$$

The two dimensional transform may be implemented as a consecutive conduct of two one-dimensional transforms, i.e. applying a DFT to all rows and subsequently to all columns (or vice versa).

The foundation of this property is the separability of the underlying basis functions, i.e.:

$$e^{-2\pi i(ux+vy)} = e^{-2\pi iux} e^{-2\pi ivy}$$

Properties of the 2D Fourier Transform (2)

■ Translation

$$\hat{f}(u - u_0, v - v_0) = f(x, y) e^{2i\pi(u_0x/M + v_0y/N)} \quad (10)$$

$$f(x - x_0, y - y_0) = \hat{f}(u, v) e^{-2i\pi(ux_0/M + vy_0/N)} \quad (11)$$

- The origin of the Fourier Transform $(0, 0)$ can be moved to the center of the frequency plane $(M/2, N/2)$ by multiplying $f(x, y)$ with $(-1)^{x+y}$
- Set $u_0 = M/2$ and $v_0 = N/2$:

$$\hat{f}(u - M/2, v - N/2) = f(x, y) e^{i\pi(x+y)} = (-1)^{x+y} f(x, y)$$

- A shift in $f(x, y)$ does not affect $|\hat{f}(u, v)|$ (shift invariance of the DFT):

$$|\hat{f}(u, v) e^{-2\pi i(ux_0/M + vy_0/N)}| = |\hat{f}(u, v)|$$

■ Periodicity

$$\hat{f}(u, v) = \hat{f}(u + N, v) = \hat{f}(u, v + M) = \hat{f}(u + aN, v + bM)$$

■ Symmetry In case $f(x, y)$ is real-valued:

$$\hat{f}(u, v) = \hat{f}^*(-u, -v) \quad |\hat{f}(u, v)| = |\hat{f}(-u, -v)|$$

Caused by the conjugate symmetry property around the origin, half of the transform coefficients are redundant. Symmetry and periodicity facilitate to keep the entire period and to shift the origin of the transform domain into $(M/2, N/2)$ as described before.

Properties of the 2D Fourier Transform (4)

■ Linear combination

$$k_1 f(x, y) + k_2 g(x, y) \Leftrightarrow k_1 \hat{f}(u, v) + k_2 \hat{g}(u, v)$$

■ Scaling

$$af(x, y) = a\hat{f}(u, v) \text{ contrasting to}$$

$$f(ax, by) = \frac{1}{ab} \hat{f}(u/a, v/b)$$

Scaling can be shown as follows (1-dim.):

- for $f(x)$: $\hat{f}(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i ux} dx$
- for $f(ax)$: $\hat{f}(u) = \int_{-\infty}^{\infty} f(ax) e^{-2\pi i ux} dx$
- Multiplication of the integral and the exponent by a/a leads to:
 $1/a \int_{-\infty}^{\infty} f(ax) e^{-2\pi i a x (u/a)} a dx$
- Applying a substitution of variables $s = ax$ ($ds = adx$):
 $1/a \int_{-\infty}^{\infty} f(s) e^{-2\pi i s (u/a)} ds$
- This expression is evidently equal to $\frac{1}{a} \hat{f}\left(\frac{u}{a}\right)$

A contracted function ($a > 1$) consequently exhibits a Fourier transform with reduced amplitude and horizontal stretching in frequency space

Laplacian and Convolution (Theorem)

Laplacian:

$$\nabla^2 f(x, y) = \frac{\partial f}{\partial x^2} + \frac{\partial f}{\partial y^2}$$
$$\widehat{\nabla^2 f(x, y)} = -(2\pi)^2(u^2 + v^2)\hat{f}(u, v)$$

Convolution: Convoluting the mask $h(x)$ with the image $f(x)$ is defined as

$$h(x) * f(x) = \int_{-\infty}^{\infty} h(\alpha)f(x - \alpha)d\alpha$$

Convolution Theorem:

$$f(x) * g(x) \Leftrightarrow \hat{f}(u) \cdot \hat{g}(u) \tag{12}$$

$$f(x) \cdot g(x) \Leftrightarrow \hat{f}(u) * \hat{g}(u) \tag{13}$$

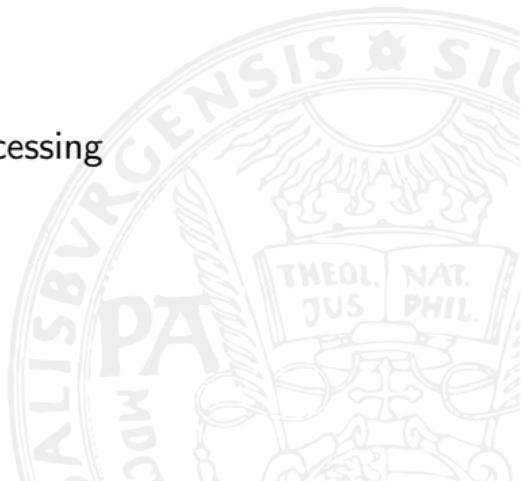
$f(x) * g(x)$... exhibits increasing computational complexity with increasing size of the mask f .
 $\hat{f}(u) \cdot \hat{g}(u)$... no increasing complexity if mask f is known

The convolution theorem can be applied for:

- Reduction of complexity of convolution: Fourier Transforms of f and g are computed and the results multiplied, the product is inverse Fourier transformed
 - Pays off with a mask size larger than 20^2 pixels
- Filtering in frequency domain



- FFT was published in 1968 by Cooley and Tukey
- Relies on an idea of C.F. Gauss in the area of matrix factorisation
- The computational complexity of the DFT when applied to N data points is (N^2)
 - Too high even for today's advanced hardware
- FFT reduces complexity to $(N \log N)$
- Enabler of an application of Fourier techniques in signal processing



Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - **Filtering in Frequency Domain**
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



$$g(x, y) = h(x, y) * f(x, y) \quad (14)$$

$$\hat{g}(u, v) = \hat{h}(u, v) \cdot \hat{f}(u, v) \quad (15)$$

$\hat{h}(u, v)$... Transfer function

$g(x, y)$... Shifting the mask $h(x, y)$ across the image $f(x, y)$

Procedure ($f(x, y)$ is given):

- Compute $\hat{f}(u, v)$
- choose $\hat{h}(u, v)$ in a way, that the resulting image emphasises certain properties
- Compute the enhanced image by applying the inverse Fourier transform to $\hat{h}(u, v) \cdot \hat{f}(u, v)$

Different Types of Filters

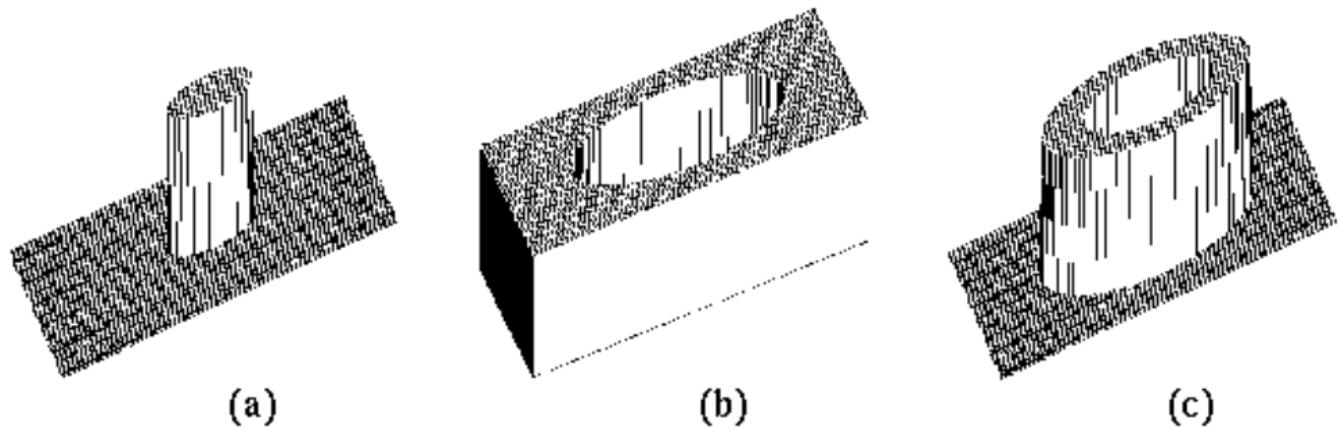


Figure: Different types of filters

Lowpass Filter: Edges and sharp transitions are phenomena of high frequency nature. If these parts are suppressed in the frequency domain, the image gets smoothed.

$\hat{h}(u, v)$ is the *Ideal Lowpass Filter* (ILPF)

$$\hat{h}(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases}$$

- D_0 is the so-called Cut-off Frequency
- $D(u, v) = (u^2 + v^2)^{1/2}$ is the distance between (u, v) and the origin
- Applying $\hat{h}(u, v) \cdot \hat{f}(u, v)$ zeros high frequency parts (edges), low frequency parts are retained.
- Filters of this type affect real- and imaginary parts but do not change the phase (*zero-phase shift*)

Problems with Ideal Low Pass Filter:

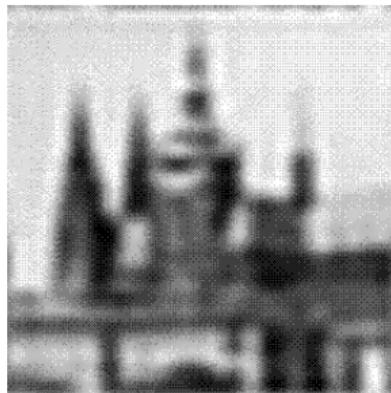
- Can not be implemented in electronic hardware
- Cutting the frequencies very sharply results in artefacts (*ringing*):
 - The shape of $h(x, y)$ (which determines the rings when convolved with a bright spot) depends on the value of D_0
 - Radii of the resulting rings are inverse proportional to the value of D_0
 - Small D_0 generates a low number of broad rings (*strong ringing*)
 - With increasing D_0 the number of rings increases but their breadth decreases.

Butterworth Filter (BLPF):

$$\hat{h}(u, v) = \frac{1}{1 + (D(u, v)/D_0)^{2n}}$$

The Butterworth Lowpass Filter is a transfer function of order n . There is no discontinuity and thus, less artefacts occur.

Lowpass Filter Example



a)



c)

b)



Highpass Filter and the Butterworth Filter (HP)

Highpass Filter: By analogy to lowpass filters, highpass filters allow high frequencies to pass, thus, edges and sharp transitions get emphasised.

$\hat{h}(u, v)$ is the *Ideal Highpass Filter* (IHPF):

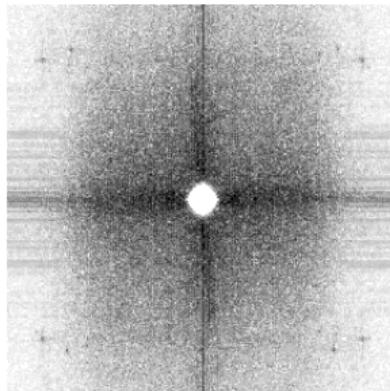
$$\hat{h}(u, v) = \begin{cases} 0 & D(u, v) < D_0 \\ 1 & D(u, v) \geq D_0 \end{cases}$$

Butterworth Filter (BHPF):

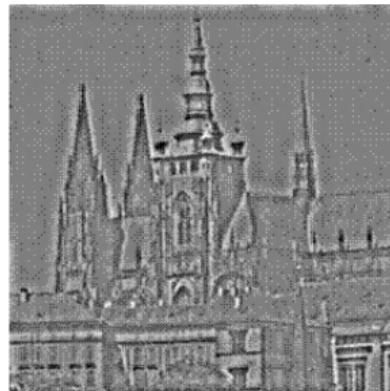
$$\hat{h}(u, v) = \frac{1}{1 + (D_0/D(u, v))^{2n}}$$

- The Butterworth highpass filter is a transfer function of order n .
- Edges and sharp transitions are kept and less artefacts occur.
- In order to retain a certain amount of lower frequencies, a constant value can be added to the transfer function (*High Frequency Emphasis*).
- Additionally, histogram equalisation can be applied to improve the result

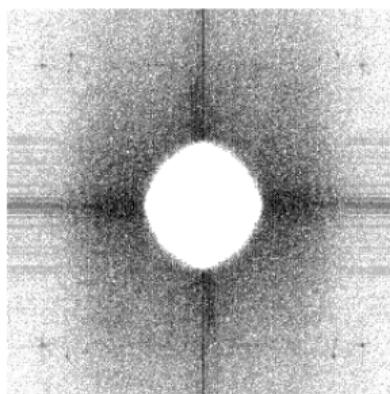
Highpass Filter Example



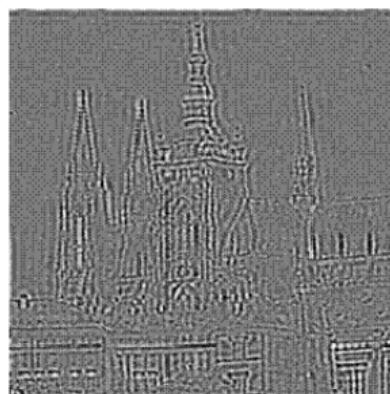
a)



b)



c)

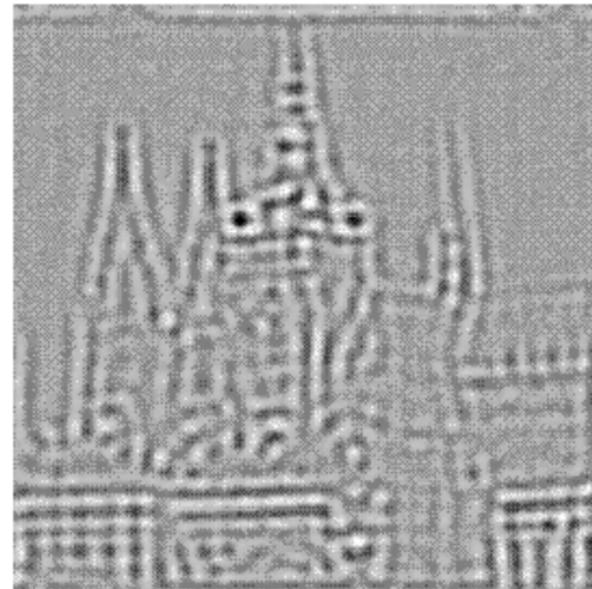
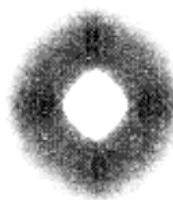


d)

Bandpass Filter

Bandpass Filter:

A specific (middle) frequency band is determined to pass and $\hat{h}(u, v)$ is designed correspondingly:



a)

b)

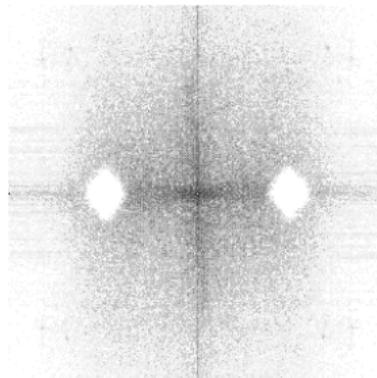
Figure: Bandpass Filter Example

Bandpass Filter for Specific Frequency Bands

More specific filtering techniques take specific properties of eventual disturbances into account:



a)



b)



c)

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - **Wavelet Transformation**
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Using the Fourier transform, it is not possible to perform frequency filtering on a local scale.

Windowed Fourier transform

- Short term Fourier transform (STFT)
- Cuts the signal into pieces (by applying smooth window functions)
- Applies the Fourier Transform to the single pieces
- PROBLEM: Width of the window functions have to be determined a priori
- Results in frequency information loss no matter how the fixed window size is chosen
- SOLUTION: A comprehensive solution to this dilemma is given by the Wavelet Transform:

$$W_{a,b}(f) = |a|^{-1/2} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt \quad (16)$$

$$\psi_{a,b}(s) = |a|^{-1/2} \psi\left(\frac{s-b}{a}\right) \quad (17)$$

The functions in equation (17) are named "Wavelets", $\psi(s)$ is the *mother wavelet*.

Wavelet Transformation - Mother Wavelets

Examples for mother wavelets are:

$$\psi(s) = (1 - s^2)e^{\frac{s^2}{2}} \quad \text{Mexican Hat} \quad (18)$$

$$\psi(s) = \frac{\sin(2\pi s) - \sin(\pi s)}{\pi s} \quad \text{Shannon Wavelet} \quad (19)$$

$$\psi(s) = \begin{cases} 1 & 0 \leq s \leq 1/2 \\ -1 & 1/2 \leq s \leq 1 \\ 0 & \text{other} \end{cases} \quad \text{Haar Wavelet} \quad (20)$$

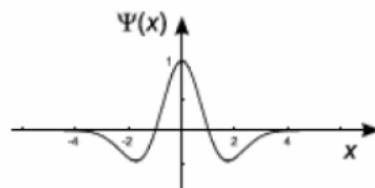
Wavelet transform depends on two parameters (a and b):

- Changing a , the wavelets in equation (17) describe different local “frequency bands”
- Large a describe broad, rather low frequent functions,
- Small a describe slim, fine detailed functions, rather representing high frequencies on a local scale
- Changing parameter b translates the time-space center of the function (which is in $s = b$)
- All wavelets are consequently translated and scaled versions of the mother wavelet

Wavelet Transformation - Mexican Hat Wavelet Example

Basiswavelet :

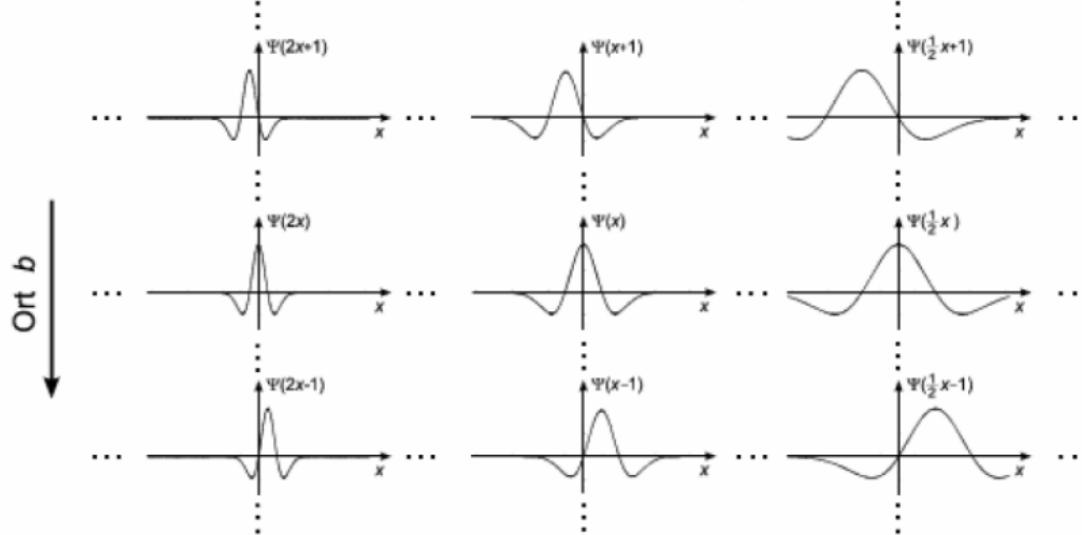
Mexican hat



$$\Psi(x) = (1 - x^2) e^{-x^2/2}$$

Waveletfamilie:

Skala a



Multiresolution Analysis - Idea

Representation of signals as different levels of approximation (i.e. resolution) and the differences among those resolutions:

- Orthogonal basis functions are being applied, parameters a and b are discretised
- $a = a_0^m$, $b = nb_0a_0^m$ with $m, n \in \mathbb{Z}$ and $a_0 > 1$, $b_0 > 1$
- Common choice is $a_0 = 2$ and $b_0 = 1$

$$W_{m,n}(f) = 2^{-m/2} \int_{-\infty}^{\infty} f(t)\psi(2^{-m}t - n) dt$$

A MRA is obtained by a series of approximation and detail spaces nested into each other, functions $\phi(t)$ and $\psi(t)$ are the respective orthonormal basis for these nested subspaces.

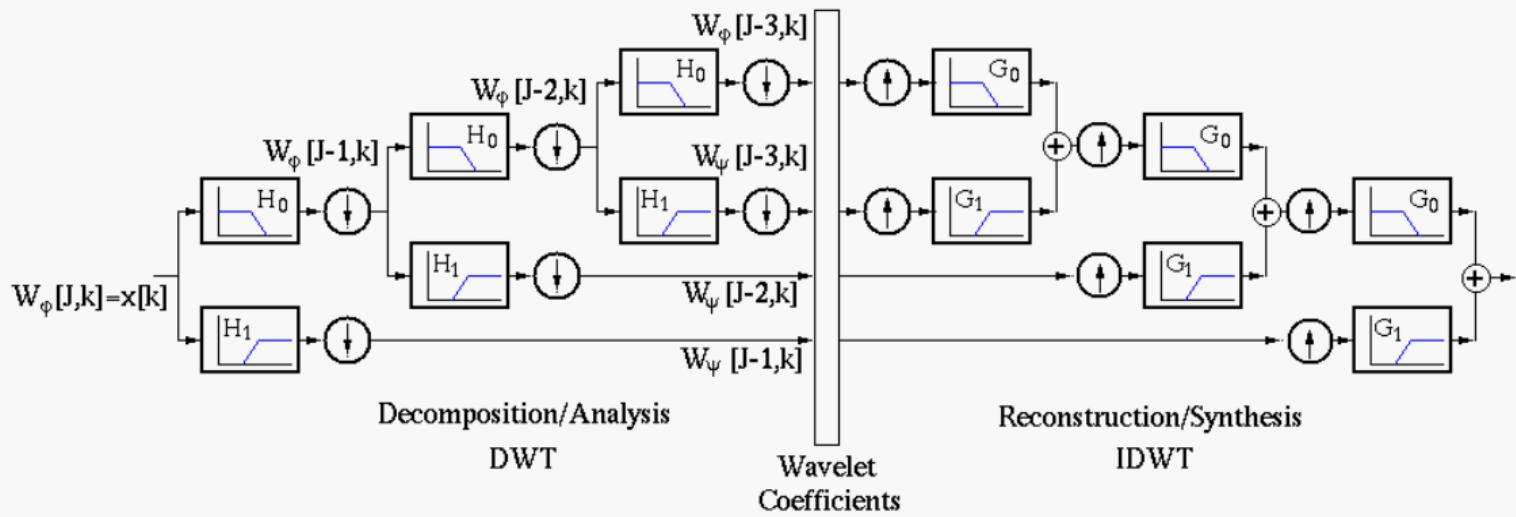
$$\phi(t) = \sum_n h(n)\phi(2t - n) \quad (21) \quad g(n) = (-1)^n h(1 - n)$$

$$\psi(t) = \sum_n g(n)\phi(2t - n) \quad (22) \quad \phi(t) \dots \text{scaling function}$$

$\psi(t) \dots \text{wavelet function}$

'Scaling Equation' relates scaling function (dilated by factor 2) to its dilated/translated versions (functions with lower "frequency" are represented by identical ones with higher frequency).

Important: The sequence $h(n)$ determines the resulting functions in a unique manner (i.e. given $h(n)$, $\phi(t)$ and $\psi(t)$ are uniquely defined).



- Applying $h(n)$ and $g(n)$ and down-sampling (**skipping every second coefficient**)
- Iterative, recursive filtering of the low pass part leads to same coefficients as a Wavelet transform → MRA represents a Wavelet transform without “Wavelets”
- High pass coefficients correspond to Wavelet coeffs., low pass coefficients to scaling coeffs.
- Reconstruction/Synthesis by upsampling (**filling a 0's after every coefficient**) and applying the “inverse” filters $\bar{h}(n)$ and $\bar{g}(n)$ (orthogonal to the orginal ones)

Fast Wavelet Transform

A fast wavelet transform in the one-dimensional case (complexity $\mathcal{O}(N)$ with N ... number of elements in the input signal):

$$\text{input} = (a, b, c, d, e, f, \dots) \quad h(n) = (1, 2, 3, 4)$$

$$WT_1 = a + 2b + 3c + 4d$$

$$WT_2 = b + 2c + 3d + 4e$$

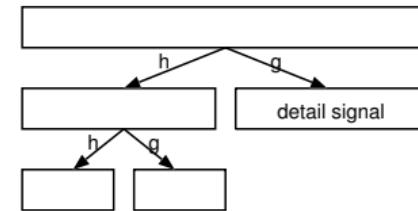


Figure: 1D Wavelet Transform, note that only the detail spaces and the last approximation space are kept as coefficients

Variants to handle border sample points in case of one-dimensional computation:

- Periodisation, signal extension, mirroring, zero-padding

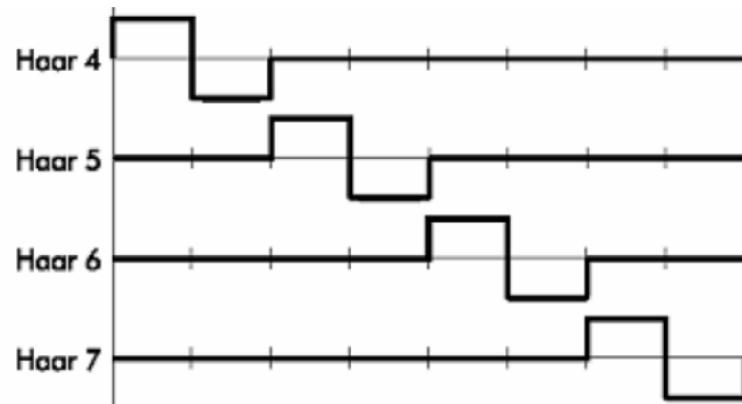
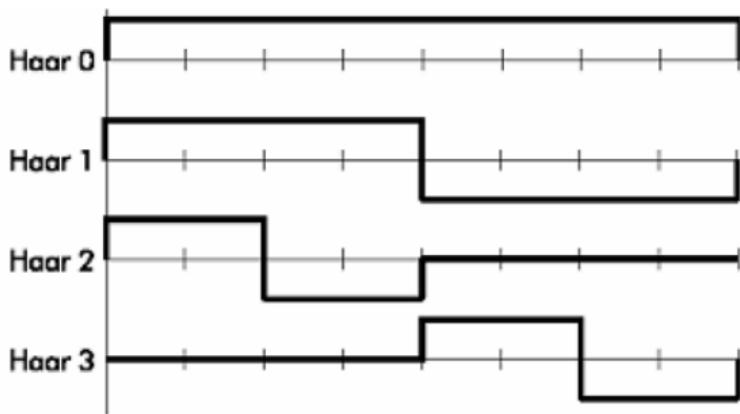
Further information on how to derive the FWT from the DWT:

<http://fourier.eng.hmc.edu/e161/lectures/wavelets/node7.html>

General information about DWT and its applications:

<https://de.mathworks.com/help/wavelet/ug/fast-wavelet-transform-fwt-algorithm.html>

Fast Wavelet Transform - Haar Wavelet Example (1)



- Discrete input signal with 8 values
- Wavelets are shifted and scaled version of the original Haar Wavelet
- FWT coefficients: $h(n) = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$, $g(n) = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]$
- Approximation space by using $h(n)$ and detail space by using $g(n)$ (nested)
 - Recall the scaling and wavelet function from the previous slide!
- Reconstruction with orthogonal filters to $h(n)$ and $g(n)$: $\bar{h}(n) = [\frac{2}{\sqrt{2}}, \frac{2}{\sqrt{2}}]$ and $\bar{g}(n) = [\frac{2}{\sqrt{2}}, -\frac{2}{\sqrt{2}}]$

Fast Wavelet Transform - Haar Wavelet Example (2)

Input	a	b	c	d	e	f	g	h
level 1: h(n)	$\bar{ab} := \frac{a}{2} + \frac{b}{2}$	$\bar{cd} := \frac{c}{2} + \frac{d}{2}$	$\bar{ef} := \frac{e}{2} + \frac{f}{2}$	$\bar{gh} := \frac{g}{2} + \frac{h}{2}$				
level 1: g(n)	$\frac{a}{2} - \frac{b}{2}$	$\frac{c}{2} - \frac{d}{2}$	$\frac{e}{2} - \frac{f}{2}$	$\frac{g}{2} - \frac{h}{2}$				
level 2: h(n)		$\bar{abcd} := \frac{\bar{ab}}{2} + \frac{\bar{cd}}{2}$			$\bar{efgh} := \frac{\bar{ef}}{2} + \frac{\bar{gh}}{2}$			
level 2: g(n)		$\frac{\bar{ab}}{2} - \frac{\bar{cd}}{2}$				$\frac{\bar{ef}}{2} - \frac{\bar{gh}}{2}$		
level 3: h(n)			$\bar{abcde}fgh := \frac{\bar{abcd}}{2} + \frac{\bar{efgh}}{2}$					
level 3: g(n)				$\frac{\bar{abcd}}{2} - \frac{\bar{efgh}}{2}$				

Input	12	4	6	8	4	2	5	7
level 1: h(n)	8		7		3		6	
level 1: g(n)	4		-1		-1		-1	
level 2: h(n)		7.5			4.5			
level 2: g(n)		0.5			-1.5			
level 3: h(n)			6					
level 3: g(n)			1.5					

a (12)	$1.5+6+0.5+4=12$
b (4)	$6+1.5+0.5-4=4$
c (6)	$6+1.5-0.5-1=6$
d (8)	$6+1.5-0.5-(-1)=8$
e (4)	$6-1.5-1.5+1=4$
f (2)	$6-1.5-1.5-1=2$
g (5)	$6-1.5-(-1.5)-1=5$
h (7)	$6-1.5-(-1.5)-(-1)=7$

Table: FWT Haar Wavelet example (done with $h(n) = [\frac{1}{2}, \frac{1}{2}]$ and $g(n) = [\frac{1}{2}, -\frac{1}{2}]$) for easier calculation, for reconstruction $\bar{h}(n) = [1, 1]$ and $\bar{g}(n) = [1, -1]$ have then to be applied

2D Wavelet Transform

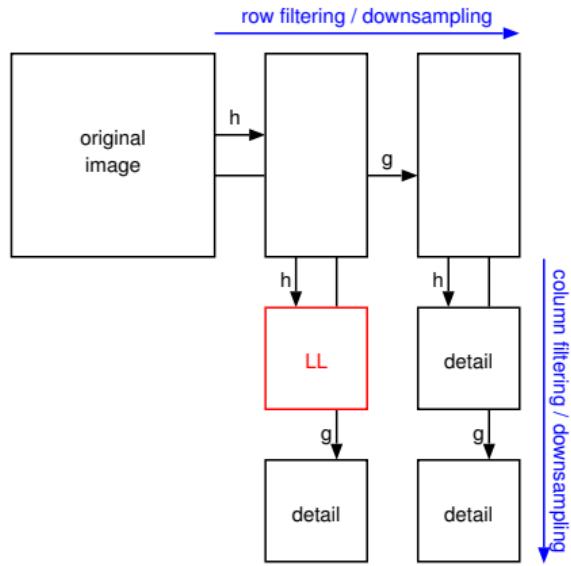


Figure: 2D Wavelet Transform

A transform of a two-dimensional function is necessary (for images)

By analogy to the Fourier case (enabled by separable functions):

- Image is first transformed along the rows using a one-dimensional transform
- Subsequently the already transformed columns are transformed using again a one-dimensional transform
- Usually, as it is the case for the one-dimensional transform, downsampling (with a factor 2) is applied

2D Wavelet Transform - Visualisation 1. Level

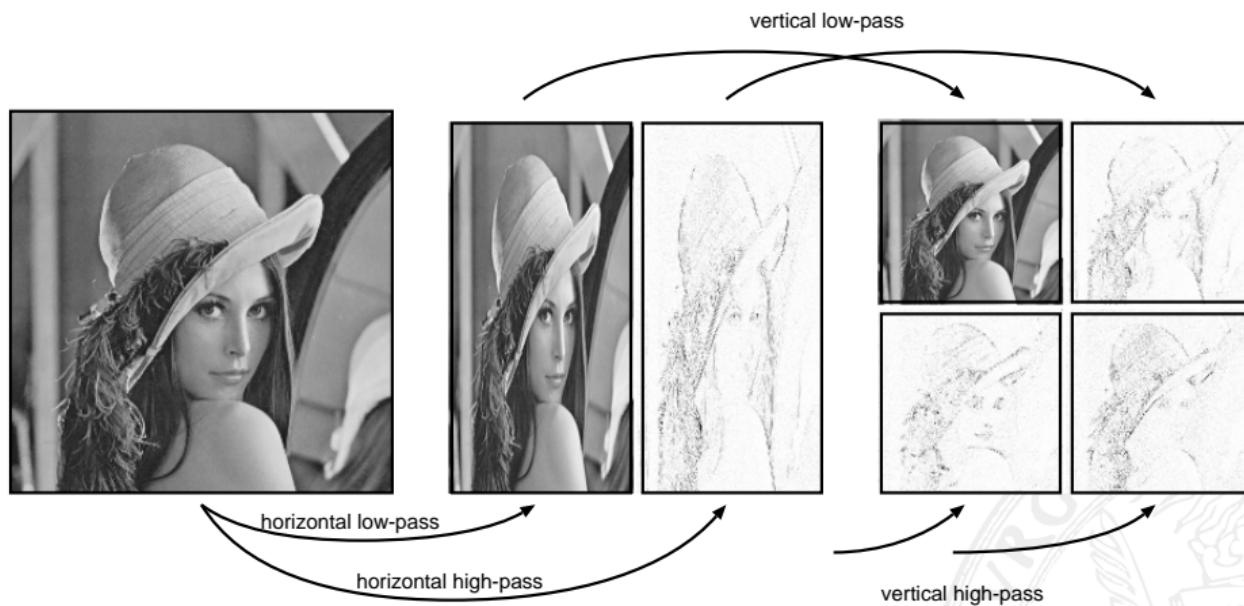


Figure: 2D Wavelet Transform: Visualisation 1. Level

2D Wavelet Transform - Visualisation 2. and 3. Level

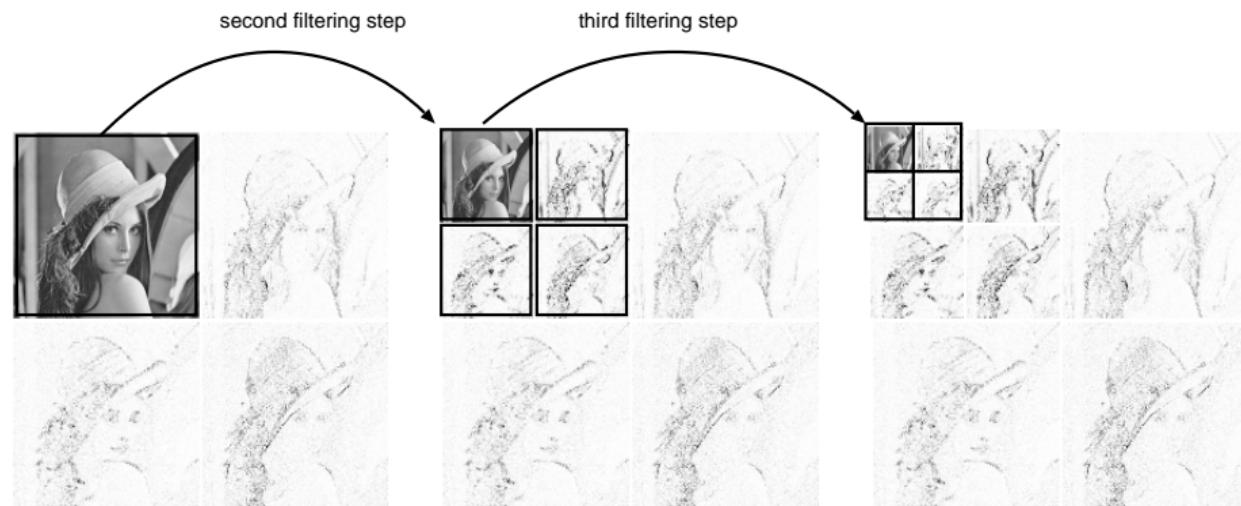


Figure: 2D Wavelet Transform: Visualisation 2.+3. Level; in-place transformation due to subsampling

2D Wavelet Transform - Full Example

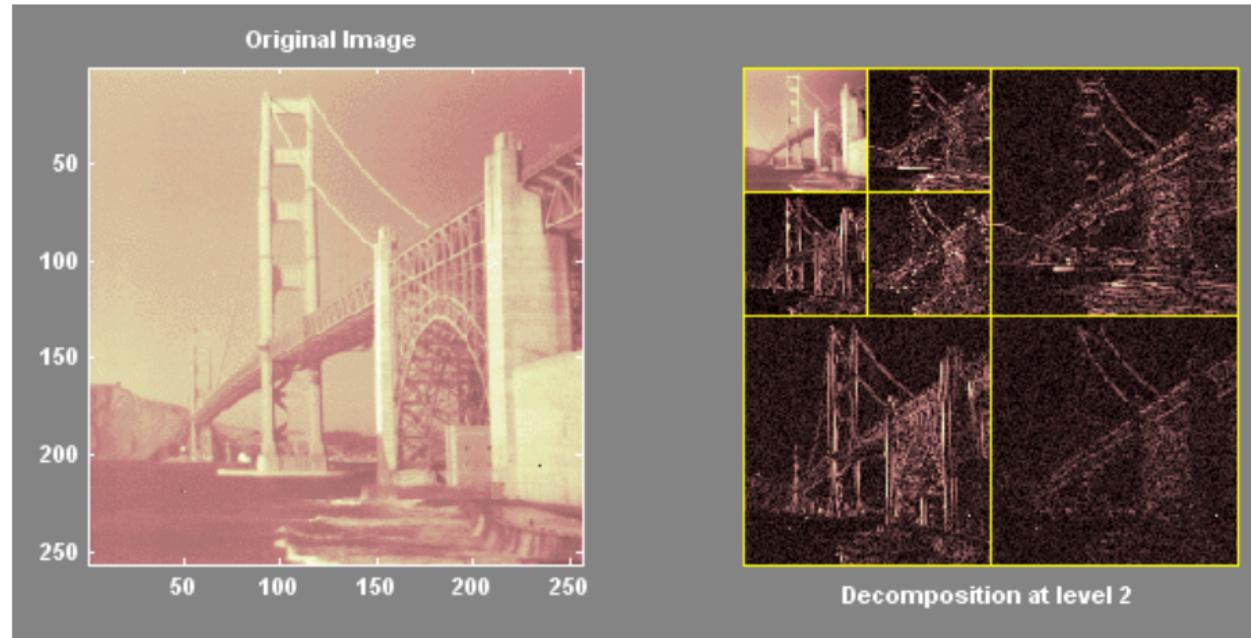
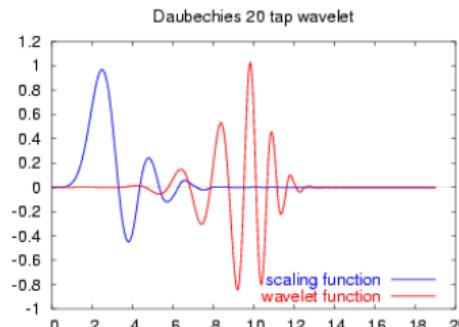
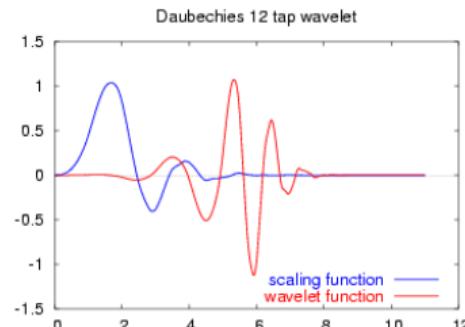
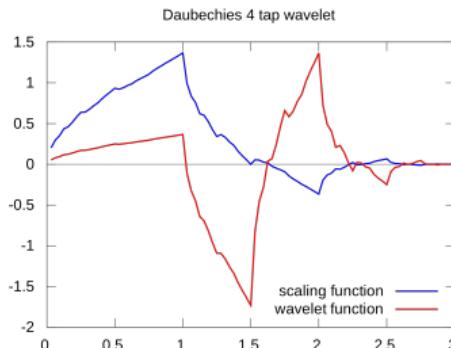


Figure: Wavelet Transform example

2D Wavelet Transformation - Daubechies Wavelets

- Special Wavelets to fulfill desired properties
- Orthogonal Wavelets with a compact support
- Forming a bi-orthogonal filter bank
- Widely used in signal analysis and compression
- Can be implemented easily and efficiently using FWT



Filtering in the Wavelet Domain:

Lowpass Filter setting detail-subbands to 0.

Highpass Filter setting LL-subband (and low frequency detail-subbands) to 0.

Bandpass Filter Subband of interest for the application has to be retained, the rest is set to 0.

Remark: Setting the LL-Subband to 0, removes the graya-scale information entirely, only high frequency edge information is retained.

Denoising: is achieved by thresholding in the wavelet domain. Only coefficients above a certain threshold are retained.

Application: Wavelet Transform is used for signal decorrelation in in the context of compression in

- JPEG2000 and
- MPEG-4 VTC (visual texture coding).

Furthermore, wavelets are well suited for many tasks in signal- and image analysis.

Denoising in the Wavelet Domain - Example

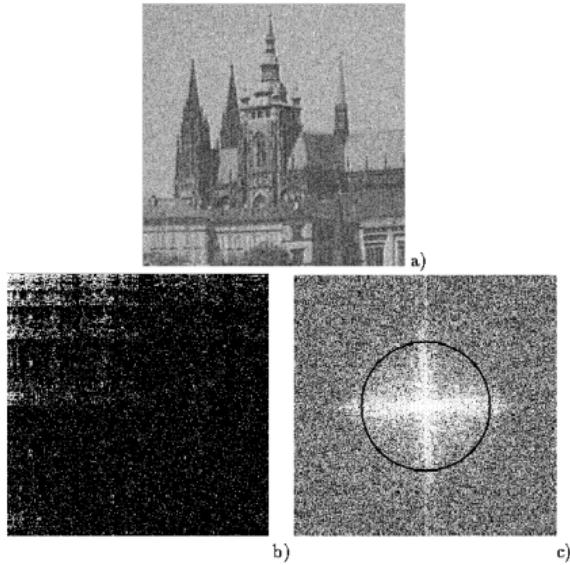


Figure: Denoising in the Wavelet/Fourier domain

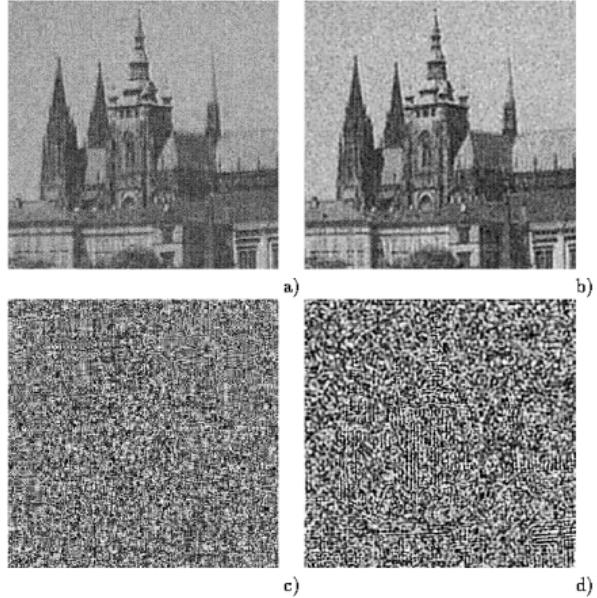


Figure: Denoising in Wavelet/Fourier domain:
Results

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - **Fourier vs. Wavelet**
 - Further Wavelet Transform variants



Fourier vs. Wavelet

Arrangement of the different frequencies for both transform domains:

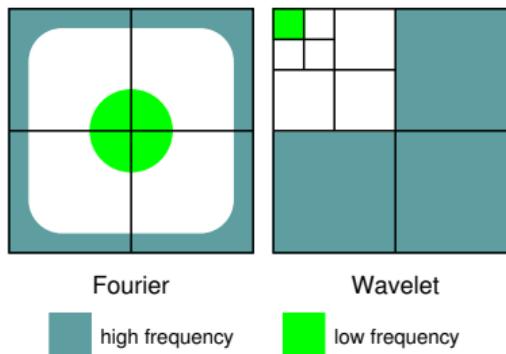


Figure: Fourier and Wavelet Transform

- **Fourier transform:** a coefficient represents the global frequency content of the entire image with frequencies u and v .
- **Wavelet transform:** a coefficient represents the local frequency content at scale 2^i in a certain neighbourhood in the image
- In case an entire frequency band needs to be processed, Fourier methods are more appropriate, for local phenomena wavelet transforms are a better choice
- According to signal theory, frequency and location can not be exactly determined at the same time

Outline

- 1 Spatial Domain Methods
- 2 Contrast Manipulation & Modification
 - Changing the Amplitude
 - Contrast Modification
 - Histogram Modification
 - Histogram-Equalisation
 - Explicit Histogram Specification
 - Gamma Correction
- 3 Image Smoothing & Denoising
 - Neighbourhood Averaging
 - Median Filtering
- 4 Image Sharpening
- 5 Transformation-Based Techniques
 - Fourier Transform
 - Filtering in Frequency Domain
 - Wavelet Transformation
 - Fourier vs. Wavelet
 - Further Wavelet Transform variants



Wavelet Packet Transform (WP)

Basic Idea:

- Iteration of the decomposition is not restricted to the low-pass sub-band
- It is applied to all wavelet sub-bands recursively
- This leads to a much better (frequency) resolution, especially of the high frequency image part

Best Basis selection:

- Technique to choose the specific sub-tree for representing the signal
- Allows to represent the signal in the most compact manner
- Application: in compression techniques (FBI-standard, J2K Part II), sub-trees are selected by optimising (information) cost functions

Local Discriminant Bases:

- Technique to choose the specific sub-tree for representing the signal
- Allows to represent the signal in a way that allows to distinguish among different signal classes
- Idea is to select the most discriminative features for a classification problem
- Application: texture classification is the most effective application area

Problem: Classical fast wavelet transform suffers from shift variance due to the downsampling stage in the transform

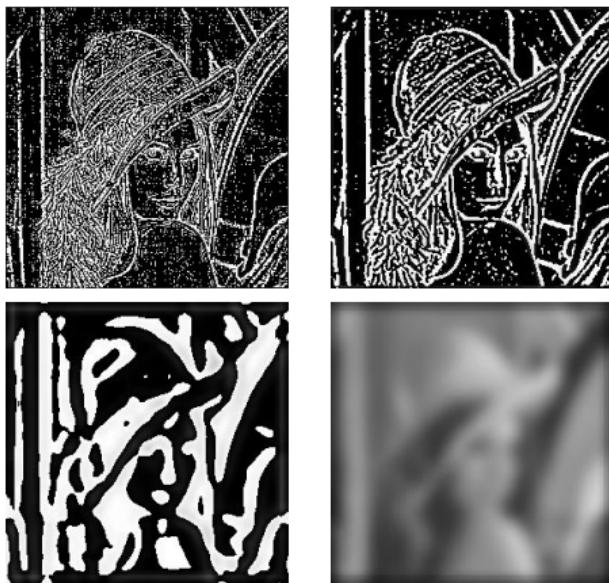


Figure: A trous Visualisation

- Shift invariance is achieved by omitting downsampling, by sacrificing the compact and redundant-free representation
- In this manner, each decomposition level produces data of the same amount as the original signal
- Contrasting to the CWT the fast wavelet transform (DWT) can be used
- Scaling is coarse (powers of two, "octaves")

- Direct computation of the transform coefficients with $O(N^2)$ complexity
- For this algorithm, an actual wavelet in explicit formulation is required to compute the inner products between signal and basis function
 - Note the difference to the DWT algorithm
- CWT is usually only used in one-dimensional application due to the large quantities of data produced and the high computational complexity