Image Processing and Imaging

Image Enhancement

Dominik Söllinger
Fachbereich Computerwissenschaften
Universität Salzburg

Winter term 2022/23

Image Enhancement Introduction

- 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

Spatial Domain Methods (1)

- $f(x, y)$ 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.

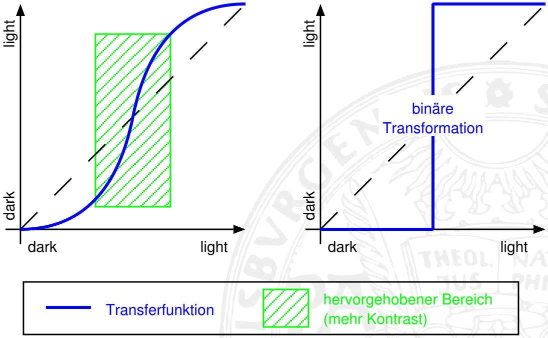


Figure: Grey-scale transformation

Spatial Domain Methods (2)

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 $w_i = -1 \; i = 1, \ldots 9$ 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

$$\text{Result} \begin{cases} = 0 & \text{all pixels identical} \\ > 0 & \text{central pixel is larger than surrounding} \\ < 0 & \text{central pixel is smaller than surrounding} \end{cases}$$

	...	$x-1$	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

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

Outline

Outline

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

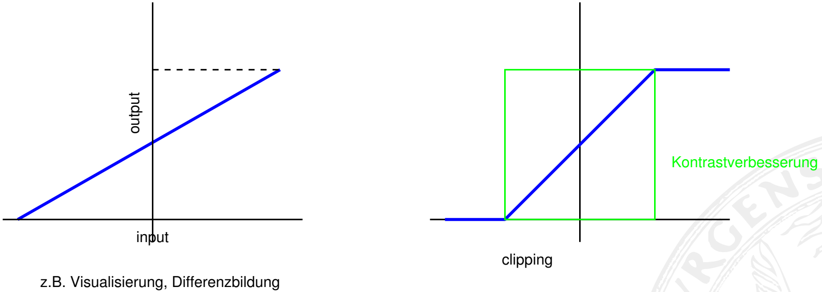


Figure: Modification of grey-scale range

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

Outline

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

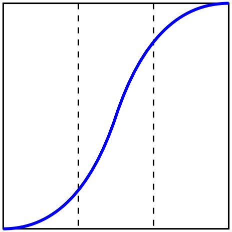


Figure: Contrast Modification

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

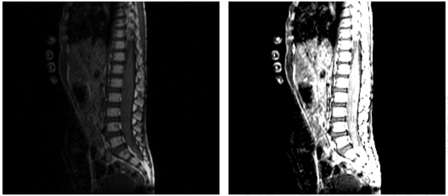


Figure: Contrast modification of a CT

Contrast can be modified by using simple transfer functions like

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

Further typical contrast modification techniques:

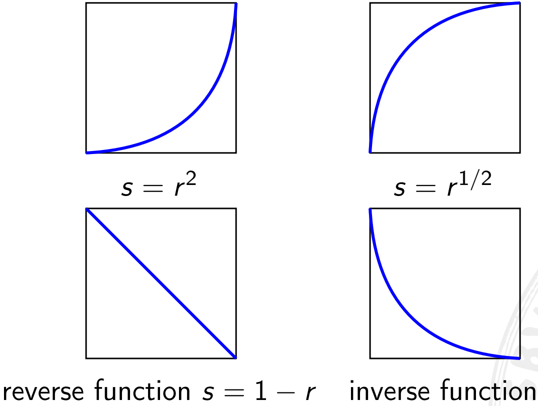


Figure: Typical contrast modification techniques

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

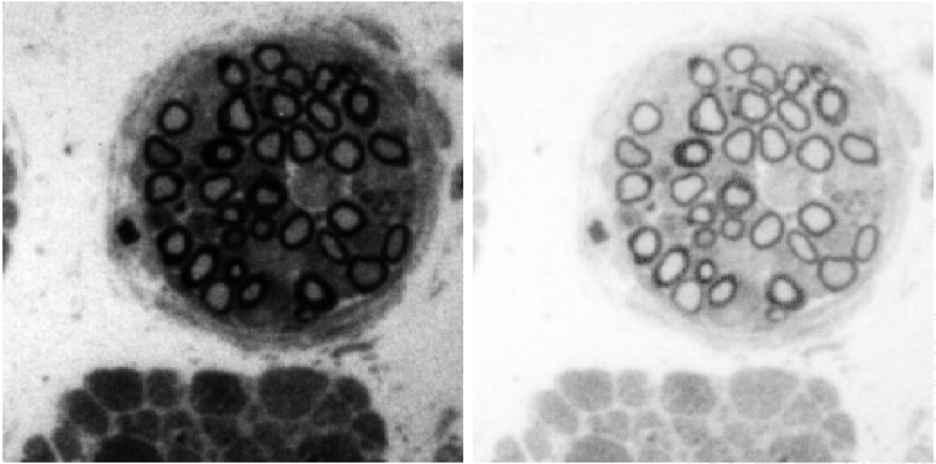


Figure: Contrast modification of a Myelin

Histogram Modification (1)

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*.

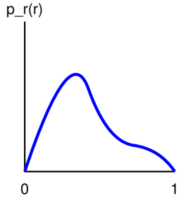


Figure: Continuous histogram

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:

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

Outline

Histogram-Equalisation (1)

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

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

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

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

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

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

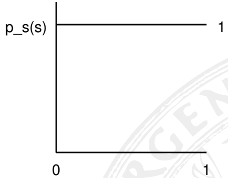
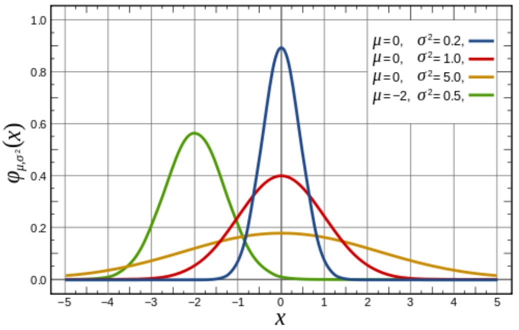


Figure: equalised histogram

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

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



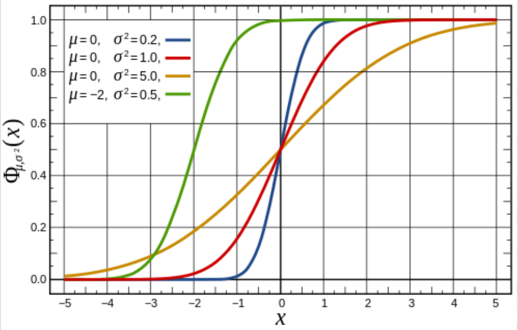


Figure: Gaussians and their cumulative distribution functions

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

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

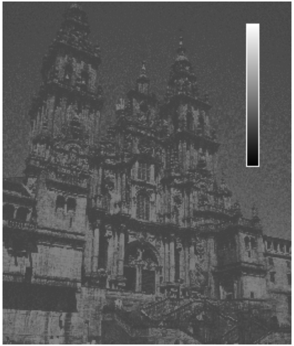
n_k ... number of occurence of grey-scale k
n ... number of pixel
L ... number of grey-scales

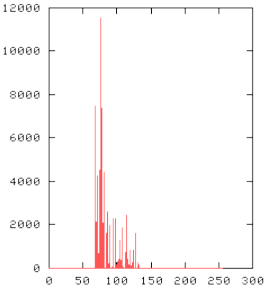
Discrete Equalisation

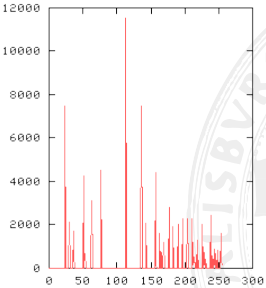
$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p_r(r_j) \qquad k = 0, \dots, L-1, \quad r_k = T^{-1}(s_k)$$

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

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.

From AHE to CLAHE

- 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) (1)

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)

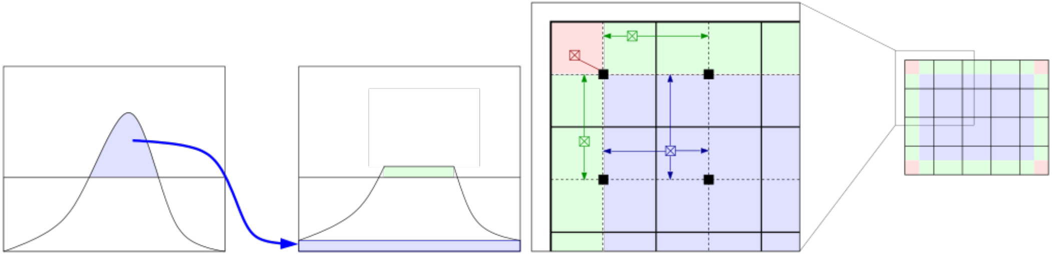


Figure: CLAHE: histogram clipping and interpolation

- 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

Gamma Correction - Why Gamma

- 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 $\rightarrow$ 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

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:

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

- 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

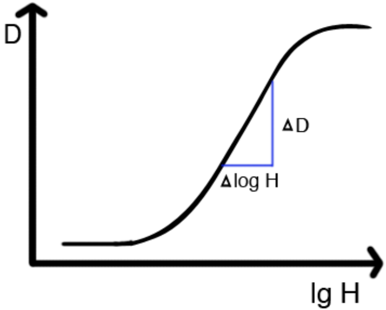


Figure: Relation between exposure (H) and resulting density (D) on the film material

- Gamma 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 $\bar{\gamma_c} = \frac{1}{\gamma_c}$:

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

- The result is then:

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

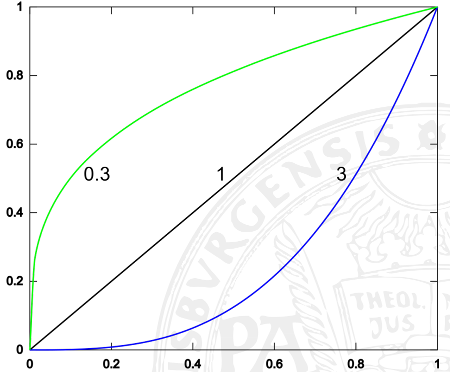


Figure: Gamma curve and inverse Gamma curve

- We implicitely 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

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

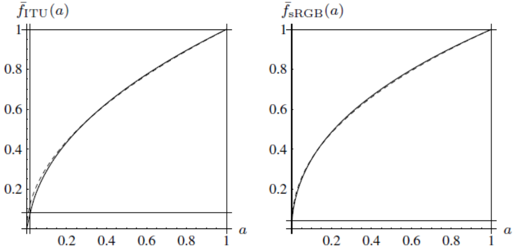


Figure: Left: ITU Gamma correction, right: sRGB Gamma

original image

$\gamma = 2.22$

$\gamma = 0.64$

$\gamma = 0.45$

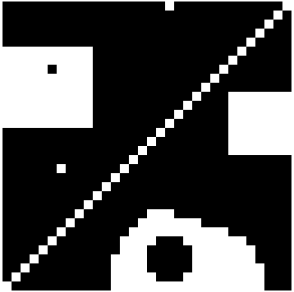
Outline

Outline

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 ... number of pixels in S):

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



$$\rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \rightarrow$$

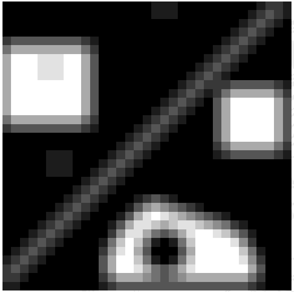


Figure: Averaging with 3×3 mask

Neighbourhood Averaging - Problem

Problem: Edges get significantly softened (*blurring*)

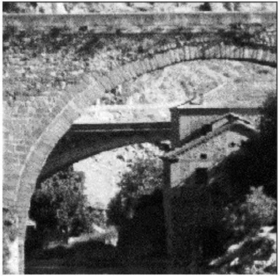
- Can be handled by applying thresholding (with threshold T)
- If the difference between original and “enhanced” pixel value is too large:
- Averaging is avoided and the original value is set

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

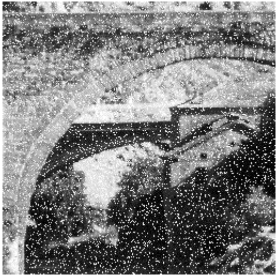
Outline

Median is used:

- Instead of computing an average in the neighbourhood
- Statistical outlayers 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 x 5 Averaging

5 x 5 Median

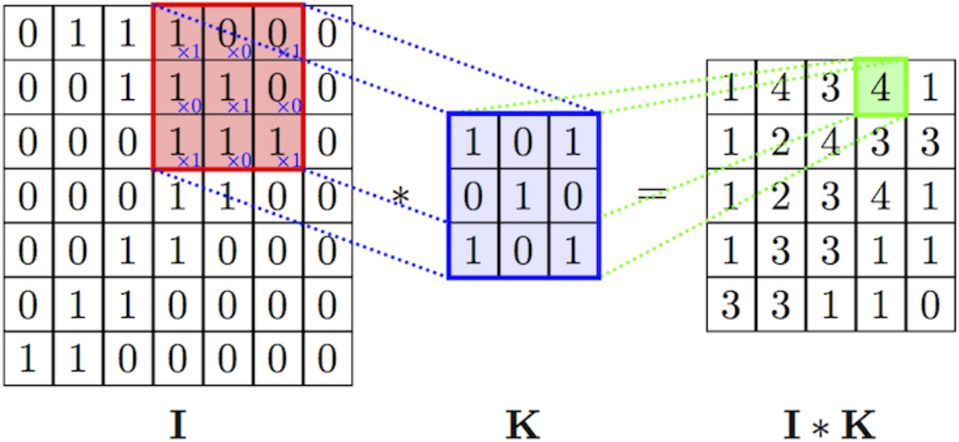


Figure: Image Convolution with 3x3 kernel example

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)

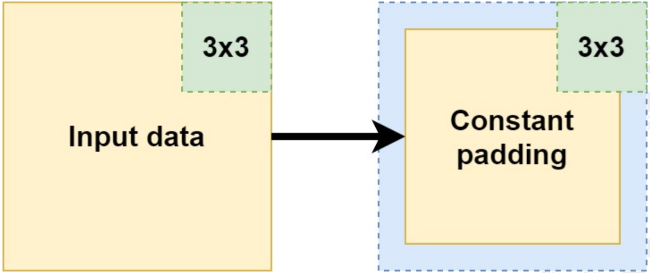


Figure: Boundary Handling - Constant padding

3	5	1
3	6	1
4	7	9

No padding

5	3	3	5	1	1	5
5	3	3	5	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

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"

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

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

In image processing $mag(G)$ 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* $g(x,y) = |G[f(x,y)]|$:

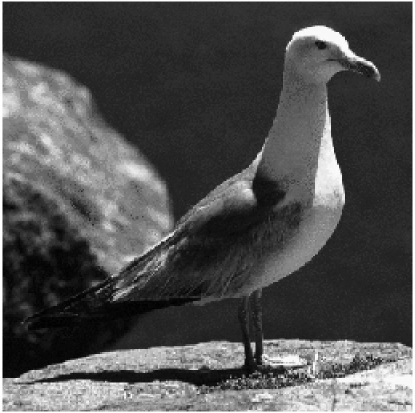
$g(x,y) = G[f(x,y)]$	$g(x,y) = \begin{cases} G[f(x,y)] & G \geq S \\ f(x,y) & \text{otherwise} \end{cases}$
$g(x,y) = \begin{cases} T_{\text{otherwise}} & G \geq S \\ f(x,y) & \text{otherwise} \end{cases}$	$g(x,y) = \begin{cases} T_1 & G \geq S \\ T_2 & \text{otherwise} \end{cases}$

Figure: Types of Gradient visualisation

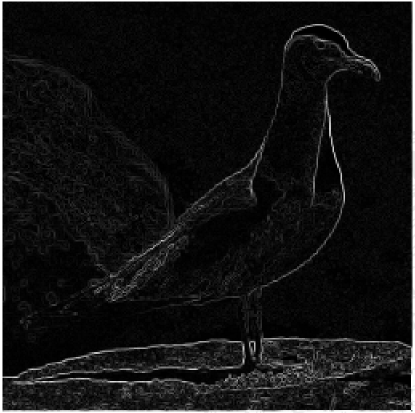
Roberts Operator:

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

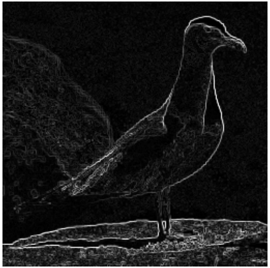
Overall, the value of the Gradient is proportional to the difference among pixels grey-values – large values for edges, small values for smooth or uniform areas.



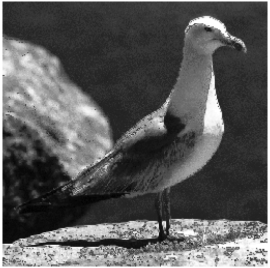
Original



Roberts Gradient image



gradient image



gradient above threshold

value for gradient if above threshold

value for above and below threshold

Outline

Transformation-Based Techniques

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:

$$f(x) = \sum_n < f(x), \psi_n(x) > \psi_n(x)$$

- Functions $\psi_n(x)$ are orthogonal basis functions
- $< f(x), \psi_n(x) >$ are the transform coefficients which determine the weight of each basis function to represent a given signal "well"
- For an application the coefficients $< f(x), \psi_n(x) >$ 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 $\psi_n(x) = e^{-\pi inx} = \cos(nx) - i\sin(nx)$
- Frequencies of periodic signals are considered
- A Fourier coefficient $< f(x), \psi_n(x) >$ represents the strength / energy of the frequency n in a signal
- Obviously, not all signals may be represented efficiently using this approach

f

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 chords.

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

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

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

Inversion can be computed if:

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

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

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

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

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

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

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

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

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

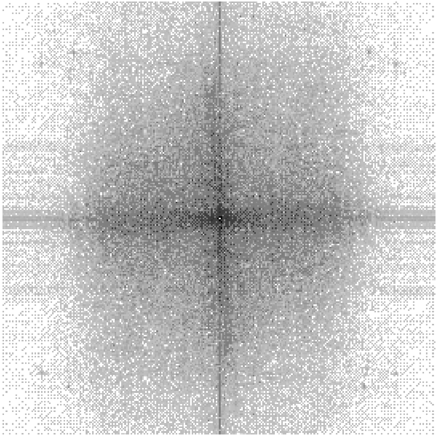
Two-dimensional:

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

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

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: Origial image (a) and its Fourier Spectrum (Magnitude) (b)

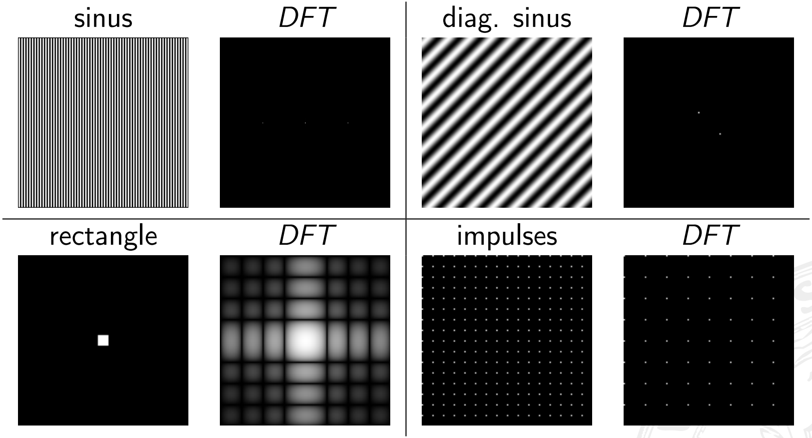


Figure: DFT Transformations

- $\hat{f}(0,0)$ 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**

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

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

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.:

$$e^{-2\pi i(ux+vy)} = e^{-2\pi i u x} e^{-2\pi i v y}$$

- **Translation**

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

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

  - 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):

$$|\hat{f}(u,v)e^{-2\pi i(ux_{\mathbf{0}}/M+vy_{\mathbf{0}}/N)}| = |\hat{f}(u,v)|$$

Properties of the 2D Fourier Transform (3)

- **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:

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

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**

$$af(x,y) = a\hat{f}(u,v) \text{ contrasting to } \qquad f(ax,by) = \frac{1}{ab}\hat{f}(u/a, v/b)$$

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

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

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

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:

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

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

$f(x) * g(x)$ ... 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 Tuckey
- 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

$$g(x, y) = h(x, y) * f(x, y) \tag{14}$$

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

$\hat{h}(u, v)$ ... 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 $\hat{h}(u, v) \cdot \hat{f}(u, v)$

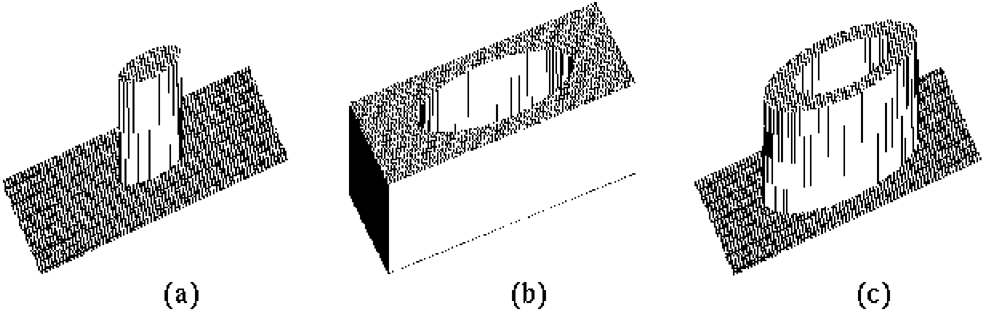


Figure: Different types of filters

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*)

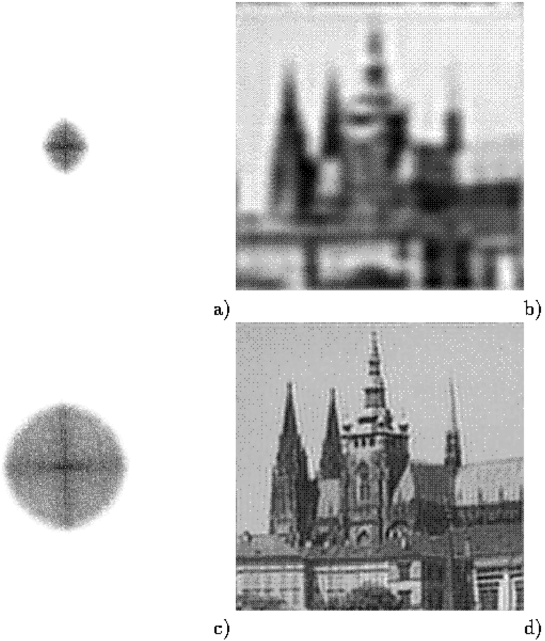
Problems with Ideal Low Pass Filter:

- 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.


a)
b)
c)
d)

Highpass Filter and the Butterworth Filter (HP)

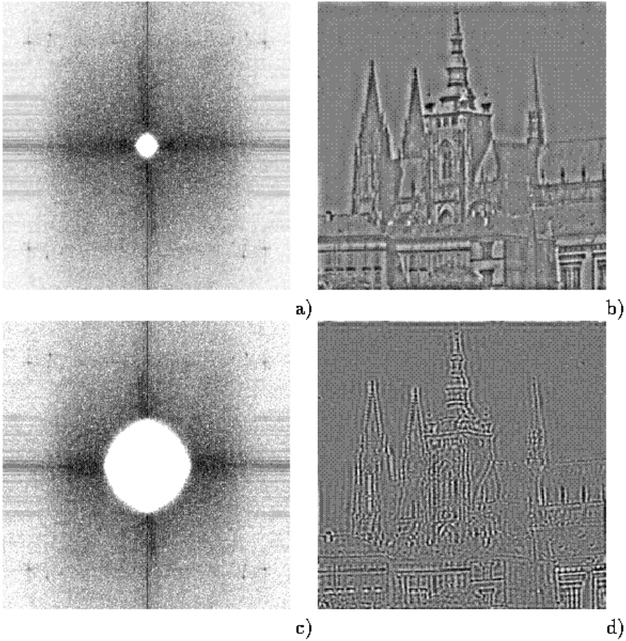
Highpass Filter: By analogy to lowpass filters, highpass filters allow high frequencies to pass, thus, edges and sharp transitions get emphasised.
$\hat{h}(u,v)$ is the *Ideal Highpass Filter* (IHPF):

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

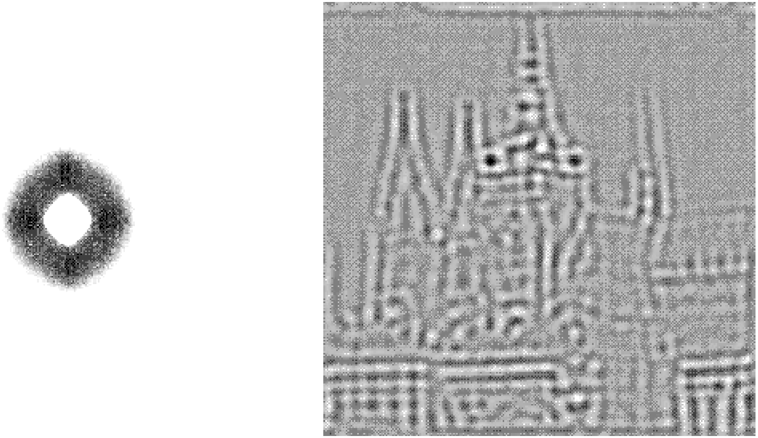
Butterworth Filter (BHPF):

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

- The Butterworth highpass filter ist 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


a)
b)
c)
d)

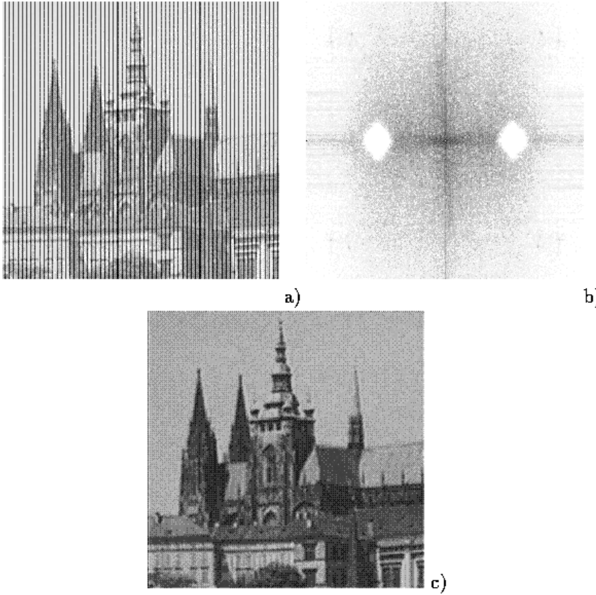
Bandpass Filter:
A specific (middle) frequency band is determined to pass and $\hat{h}(u, v)$ is designed correspondingly:



a) b)

Figure: Bandpass Filter Example

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



a) b) c)

Outline

Drawback of the Fourier Transform (FT):

- We only know WHICH frequencies occur in an signal (image) but NOT WHERE the frequencies occur
- Hence no filtering on a local scale is possible
- We can circumvent the problem by applying a "windowed" FT also called *Short Term Fourier Transform* (STFT)
- Tradeoff between frequency and time/spatial resolution.
- Problem(s): Window sizes have to be determined a priori.

A better solution to this problem is given by the **Wavelet Transform**.

Wavelet Transformation - Motivation

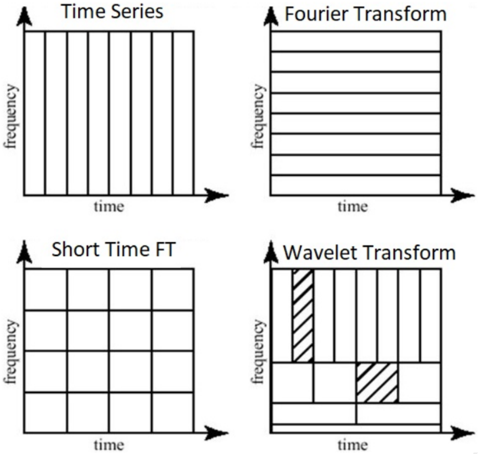


Figure: Schematic overview of the time and frequency resolutions of the different transformations. Image source: Link

The Fourier Transform used sines and cosines of different frequencies to analyze/decomposed a signal. What if we used function instead? For instance, a function such as ...

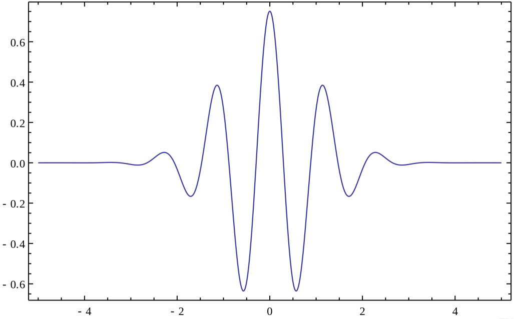


Figure: Morlet Wavelet

We call these special type of function "wavelet", or more precisely "mother wavelet".

By convolving the mother wavelet with a signal, we can determine how "much" of the wavelet is contained in the signal. But how can we choose the frequency and then determine its location?

Answer: Simply scaling and shifting of the mother wavelet.

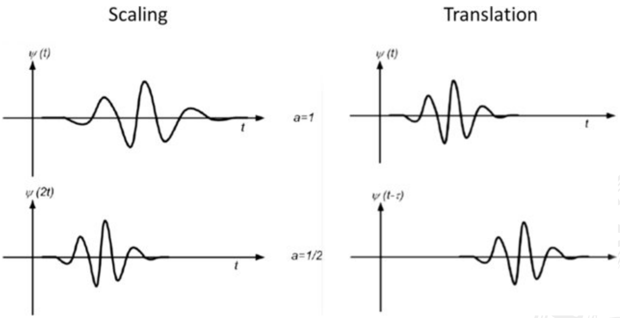


Figure: Example of shifting and scaling of a wavelet

From a chosen mother wavelet $\psi(s)$, we can generate multiple scaled/shifted version $\psi_{a,b}(s)$ (sometimes called child wavelets).

$$\psi_{a,b}(s) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{s-b}{a}\right) \tag{16}$$

We can then calculate the wavelet coefficients $W_{a,b}$ from the signal $f(t)$ as follows:

$$W_{a,b}(f) = \int_{-\infty}^{\infty} f(t)\cdot\psi_{a,b}(t)dt \tag{17}$$

Note that since we consider the continuous wavelet transform $a \in \mathbb{R}^+$ and $b \in \mathbb{R}$.

Important to know:

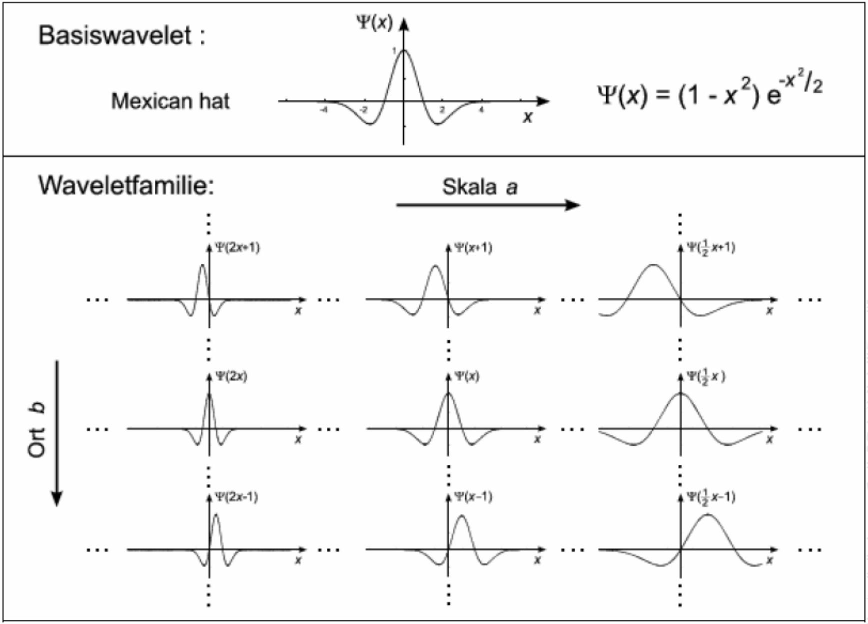
- "Continuity" in the context of the CWT refers to the wavelet function (and the fact a and b can be real numbers). We can still use the CWT to analyze a discrete signal $f(x)$.
- By changing a, we target different local "frequency bands".
- By choosing a large a, we focus on low frequency bands
- By making a small, we focus on high frequencies (the details in a signal).
- A mother wavelet is simply a blueprint from which we create different scaled/shifted versions.

Some examples for mother wavelets are:

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

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

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


Basiswavelet :
Mexican hat
Ψ(x)
x
Ψ(x) = (1 - x²) e^{-x²/2}
Waveletfamilie:
Skala a
Ort b
Ψ(2x+1)
Ψ(x+1)
Ψ(½x+1)
Ψ(2x)
Ψ(x)
Ψ(½x)
Ψ(2x-1)
Ψ(x-1)
Ψ(½x-1)

Th drawback of the CWT is that the representation of the signal is often redundant, since a and b are continuous over $\mathbb{R}$. The coefficients tend to be highly correlated.

However, it turns out that the original signal can be completely reconstructed by a sampled version of $W_{a,b}$. Typically, we sample $W_{a,b}$ f in a dyadic grid, i.e.,

$$a = 2^m \text{ and } b = n2^m$$

where $m, n \in \mathbb{Z}$.

Hence, we can compute our DWT coefficients as follows:

$$W_{m,n}(f) = \frac{1}{\sqrt{2^m}} \int_{-\infty}^{\infty} f(t) \cdot \psi(\frac{t - n2^m}{2^m})dt \tag{21}$$

$$= 2^{-m/2} \int_{-\infty}^{\infty} f(t) \cdot \psi(2^{-m}t - n)dt \tag{22}$$

$$\tag{23}$$

This choice of a and b finally brings us to the concept of multi-resolution analysis (MRA) — the design method of most practically relevant DWTs.

Stated in Layman's terms, the concept of MRA shows that any signal can be decomposed into scales with different time and frequency resolution. For high frequencies, good time and poor frequency resolution is provided. While for low frequencies, the opposite is the case.

The MRA provides the theoretical basis for the calculation of wavelet coefficients with so-called **filterbanks** → Fast Wavelet Transform (FWT).

The basic idea the MRA relies upon is that a function space can be decomposed into a sequence of nested subspaces.

Let v_{j-1} and v_j be a set of functions sampled with a sampling rate of 2^{j-1} and 2^j, respectively. As v_j contains functions sampled at a higher sampling rate, some of those functions can not be contained in v_{j-1}. We define this set of functions to be w_{j-1}.

Hence, $v_j = v_{j-1} \oplus w_{j-1}$ and $v_{j-1} \cap w_{j-1} = \{\emptyset\}$

Let $f_J \in v_J$, $f_{J-1} \in v_{J-1}$ and $d_{J-1} \in w_{J-1}$ where J is some scalar.

$$\begin{aligned} f_J(t) &= d_{J-1}(t) + f_{J-1}(t) \\ &= d_{J-1}(t) + d_{J-2}(t) + f_{J-2}(t) \\ &= \ldots = \sum_{j=0}^{J-1} d_j(t) + f_0(t) \end{aligned}$$

Using this mathematical framework, we express any signal $f_J(t)$ as follows:

$$f_J(t) = \sum_{j=0}^{J-1} \sum_k W_{j,k}(t)\psi_{j,k}(t) + \sum_k s_{0,k}\varphi_{0,k}(t)$$

where

ψ ... Mother wavelet
φ ... Father wavelet (Scaling function)

The mathematical explanation might be confusing. It's often easier to study the MRA from the perspective of a filter bank.

What can be shown is that wavelet coefficients can be efficiently computed by iteratively filtering of the signal $x[n]$ using a lowpass ($h[n]$) and highpass filters ($g[n]$).

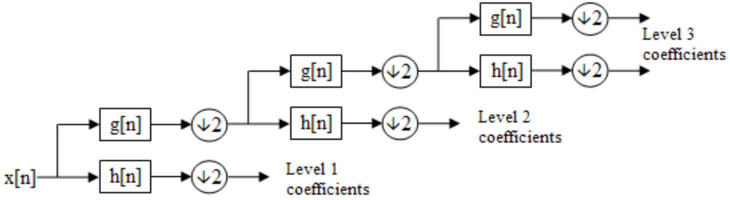
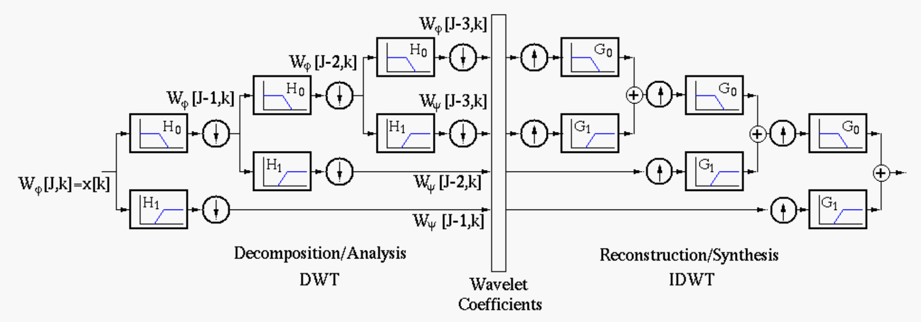


Figure: Example of a wavelet filter bank

How to find the filter coefficients?
The filter coefficents simply depend on the chosen wavelet family (father and mother wavelet).

Some examples of commonly used families of Wavelets can be found here: Link

Note that at each level, the high pass filter yields the wavelet coefficients $W_{m,n}$ and the lowpass filter yields an approximation (down-scaled) version of the image.



Why is it called "Fast" Wavelet Transform?

Due to the downsampling step, half of the signal is lost at each level. Hence, computing the coefficients becomes faster and faster after each level. As a result, the FWT can be computed in $O(n)$. Note that this is even faster, than the FFT ($O(nlog(n))$).

Note that to reconstruct the full signal, we only need to keep the detail coefficients and the last approximation.

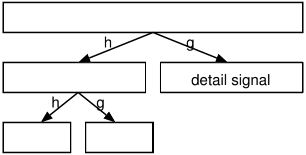
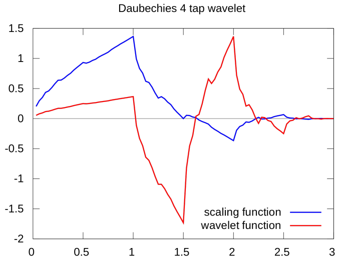
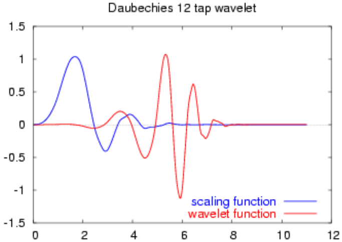


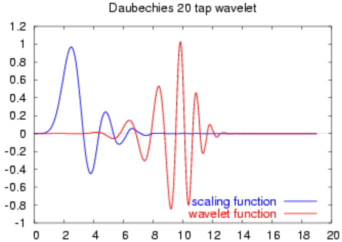
Figure: 1D Wavelet Transform

- 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







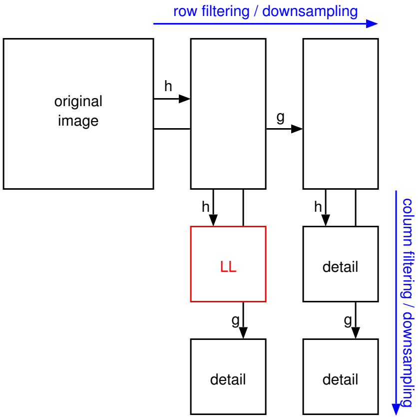


Figure: 2D Wavelet Transform

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

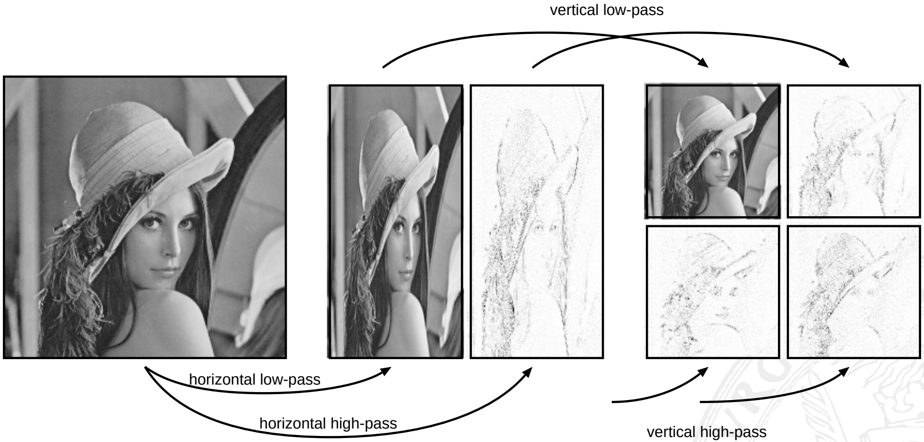


Figure: 2D Wavelet Transform: Visualisation 1. Level

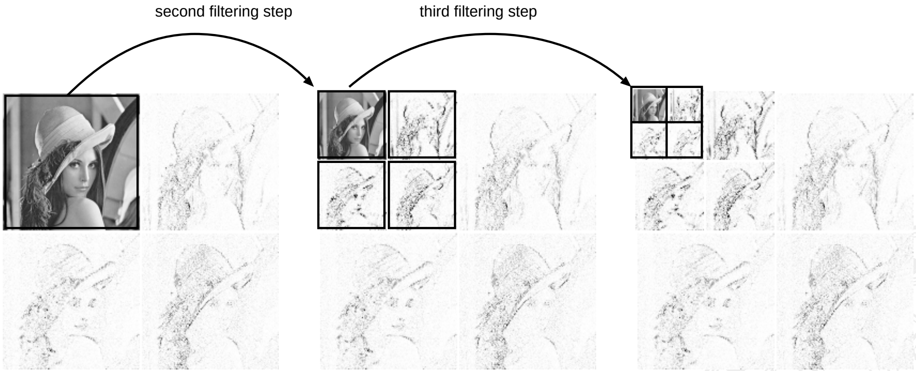


Figure: 2D Wavelet Transform: Visualisation 2.+3. Level; in-place transformation due to subsampling

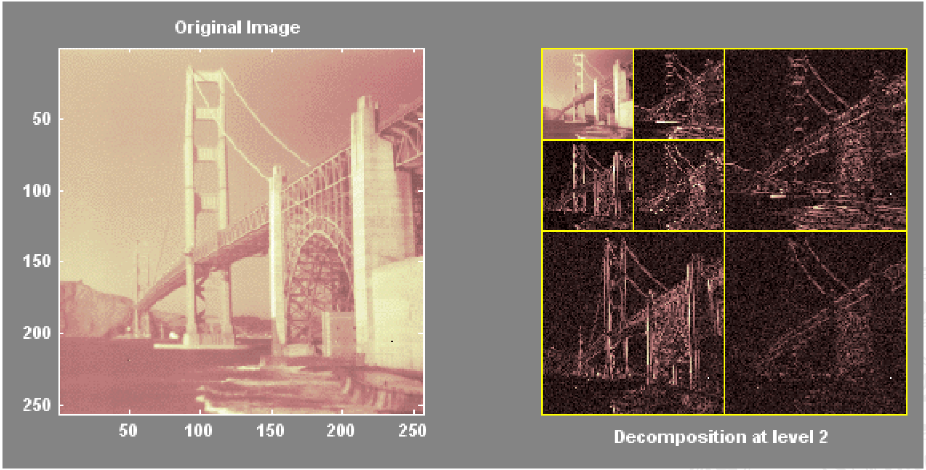


Figure: Wavelet Transform example

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 grayscale 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.

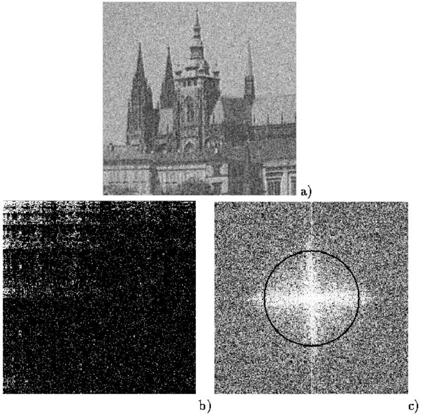


Figure: Denoising in the Wavelet/Fourier domain

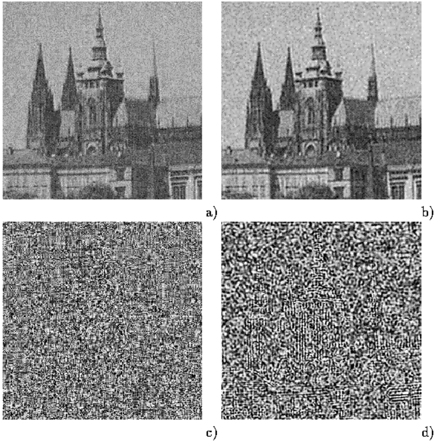


Figure: Denoising in Wavelet/Fourier domain: Results

Outline

Arrangement of the different frequencies for both transform domains:

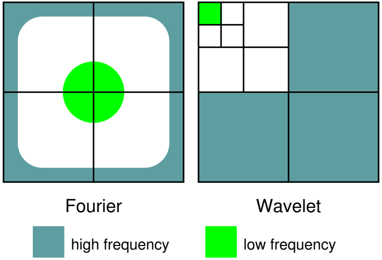


Figure: Fourier and Wavelet Transform

- **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

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

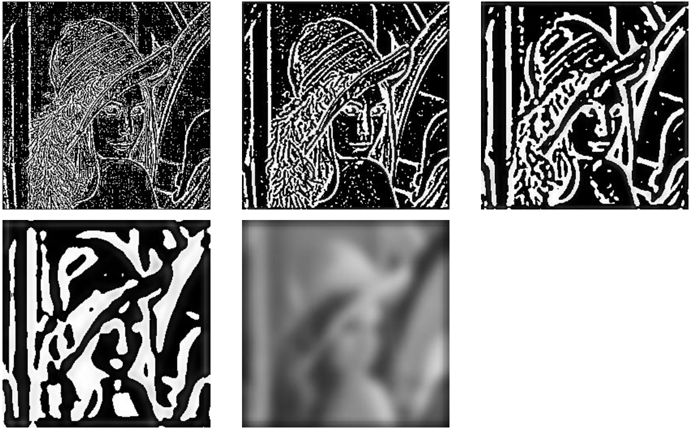


Figure: A trous Visualisation

- 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")