Lecture 11:
Linear Filters I:
System Theory

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Contents

Motivation

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

- In the previous lecture we have considered filters based on point operations.
  They ignore the neighbourhood structure of each pixel.
- Let us now study filters that can take into accout the spatial context of the pixels.
- The simplest of these filters are linear and shift invariant.
  One can show that they can be described by convolutions.
- The behaviour of these filters can be nicely analysed in the Fourier domain:
  This turns convolutions into simple multiplications.
- Thus, linear shift invariant filters are elegant and transparent in theory and practise.

Linear System Theory

Linear Filter

- A filter L is called *linear,* if it satisfies the *superposition principle*

$$L(\alpha f + \beta g) \;=\; \alpha L f + \beta L g$$

for all (continuous or discrete) images f, g, and for all real numbers α, β.

Linear Shift Invariant (LSI) System

- A *shift (translation) invariant* filter acts identically at all locations.
- More formally, a shift-invariant filter L satisfies

$$L T_b f \;=\; T_b L f$$

for all translations T_b with $(T_b f)(x) := f(x-b)$.

- A filter that is both linear and shift invariant is also called an *LSI system.*

Impulse Response of a Discrete LSI System

- The *impulse response (Impulsantwort)* of a discrete LSI filter $\boldsymbol{L}$ is the result of filtering a *discrete Dirac delta impulse:*

$$\boldsymbol{h} = \boldsymbol{L}\boldsymbol{\delta}_0$$

where $\boldsymbol{\delta}_0 = (\delta_{0,i})$ with the *Kronecker symbol*

$$\delta_{i,j} \;:=\; \begin{cases} 1 & \text{for } i = j, \\ 0 & \text{else.} \end{cases}$$

- Since every discrete signal $\boldsymbol{f} = (f_1, ..., f_N)^\top$ can be represented as linear combination of N unit impulses $\boldsymbol{\delta}_1$,...,$\boldsymbol{\delta}_N$, linearity and shift invariance imply

$$\boldsymbol{L}\boldsymbol{f} \;=\; \boldsymbol{L}\sum_{i=1}^{N} f_i \boldsymbol{\delta}_i \;=\; \sum_{i=1}^{N} f_i\, \boldsymbol{L}\boldsymbol{\delta}_i \;=\; \sum_{i=1}^{N} f_i\, \boldsymbol{L}\,(\boldsymbol{T}_i \boldsymbol{\delta}_0) \;=\; \sum_{i=1}^{N} f_i\, \boldsymbol{T}_i\,(\boldsymbol{L}\boldsymbol{\delta}_0).$$

- This shows: *Any LSI system $\boldsymbol{L}$ is fully characterised by its impulse response $\boldsymbol{L}\boldsymbol{\delta}_0$.*

Linear System Theory (3)

Example: Stock Market Price Averaged over the Last 200 Days

$$u_i \;:=\; \frac{1}{200}\sum_{k=0}^{199} f_{i-k}\,.$$

- This averaging can be represented as a discrete convolution (cf. Lecture 2):

$$u_i \;=\; \sum_{k=-\infty}^{\infty} f_{i-k}\, w_k \;=\; (\boldsymbol{f} * \boldsymbol{w})_i$$

  with the convolution mask

$$w_k := \begin{cases} \frac{1}{200} & \text{for } k \in \{0, ..., 199\}, \\ 0 & \text{else.} \end{cases}$$

- Such a convolution filter is linear and shift invariant (Assignment H1, Problem 3).
- Its impulse response $\boldsymbol{h} = \boldsymbol{L}\boldsymbol{\delta}_0 = \boldsymbol{\delta}_0 * \boldsymbol{w}$ is given by the convolution mask:

$$h_i \;=\; \sum_{k=-\infty}^{\infty} \delta_{0,\,i-k}\, w_k \;=\; w_i \qquad \forall i.$$

Linear System Theory (4)

Repetition from Lecture 2: Convolution

- discrete convolution in 1-D:

$$(\boldsymbol{f} * \boldsymbol{w})_i \;:=\; \sum_{k=-\infty}^{\infty} f_{i-k}\, w_k$$

- discrete convolution in 2-D:

$$(\boldsymbol{f} * \boldsymbol{w})_{i,j} \;:=\; \sum_{k=-\infty}^{\infty}\sum_{\ell=-\infty}^{\infty} f_{i-k,\,j-\ell}\, w_{k,\ell}$$

Linear System Theory (5)

- continuous convolution in 1-D:

$$(f * w)(x) \;:=\; \int_{-\infty}^{\infty} f(x-x')\, w(x')\, dx'$$

- continuous convolution in 2-D:

$$(f * w)(x,y) \;:=\; \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x-x',\, y-y')\, w(x',y')\, dx'\, dy'$$

Signals with finite extension can be mirrored and extended periodically.

Linear System Theory (6)

Important Properties of the Convolution

(cf. Homework H1, Problem 3, and Lecture 4)

- **Linearity:**

$$(\alpha f + \beta g) * w \;=\; \alpha\,(f * w) + \beta\,(g * w)$$

for all signals/images f, g and all real numbers α, β.

- **Shift Invariance:**

$$T_b(f * w) = (T_b f) * w$$

for all translations T_b.

- **Commutativity:**

$$f * w \;=\; w * f.$$

Function and convolution kernel play an equal role.

- **Associativity:**

$$(f * v) * w \;=\; f * (v * w).$$

Successive convolution with kernels v and w comes down to a single convolution with the kernel $v * w$.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Linear System Theory (7)

- **Distributivity:**

$$(f+g)*w \;=\; f*w \;+\; g*w.$$

- **Differentiation:**

$$(f*w)' \;=\; f'*w \;=\; f*w'.$$

  Thus, either the signal or the kernel is differentiated.

- **Differentiability:**

  Convolving with a smooth kernel makes a signal smoother:
  If $f \in C^0(\mathbb{R})$ and $w \in C^n(\mathbb{R})$, then $(f*w) \in C^n(\mathbb{R})$.

- **Convolution Theorem of the Fourier Transform:**

$$\mathcal{F}[f*w] \;=\; \mathcal{F}[f] \cdot \mathcal{F}[w].$$

  This allows an efficient convolution if the kernels have a large support region.

Linear System Theory (8)

Importance of Convolutions in Linear System Theory

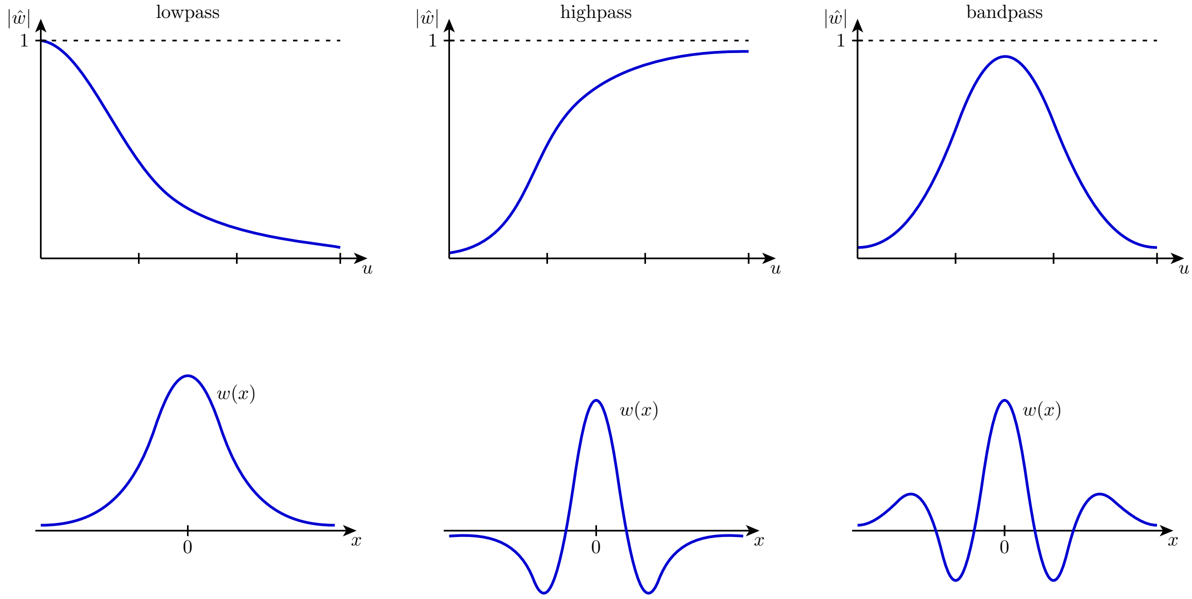
- Any convolution is linear and shift invariant, i.e. it creates an LSI system.
- More importantly, it can be shown that even the reverse is true:
  An LSI system always performs a convolution !
- The convolution mask is given by the impulse response.

Importance of the Fourier Transform in Linear System Theory

- We have seen that LSI systems are fully characterised by convolutions.
- Convolutions in the spatial domain become multiplications in the Fourier domain.
  Thus, Fourier analysis is perfectly suited for LSI filters.
- For large convolution kernels, it is more efficient to perform the computation in the Fourier domain.
- LSI filters are often studied in the Fourier domain, in order to understand their frequency behaviour.
- Often one even starts designing LSI filters in the Fourier domain.
  Afterwards one transforms them back to the spatial domain.

Linear System Theory (9)

Basic Types of LSI Filters



Top: Basic types of convolution kernels in the Fourier domain. **Bottom:** Corresponding kernels in the spatial domain. Author: T. Schneevoigt.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Linear System Theory (10)

Often one uses the following taxonomy to characterise LSI filters:

- *Lowpass filters:* Low frequencies are less attenuated than high ones.
- *Highpass filters:* High frequencies are less attenuated than low ones.
- *Bandpass filters:* A specific frequency band is hardly attenuated.

Let us now study these three LSI filter types in more detail.

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Lowpass Filters

Goals

- smooth an image by eliminating noise and unimportant small-scale details
- design in the spatial domain: convolution with a weighted averaging mask
- design in the Fourier domain: attenuate high frequencies

M I A

Design in the Spatial Domain: Box Filters

- uses convolution mask of size $(2m+1) \times (2m+1)$ with weights $\frac{1}{(2m+1)^2}$
- can also be implemented efficiently for large masks in the spatial domain:
  - filter is separable
  - Shifting the 1-D mask by one pixel to the right removes one grey value at the left end, and it adds one at the right end:

$$\begin{aligned} u_{i+1} &= \frac{1}{2m+1} \sum_{k=-m}^{m} f_{i+1-k} \\ &= \frac{1}{2m+1} \Big(f_{i+m+1} - f_{i-m} + \sum_{k=-m}^{m} f_{i-k} \Big) \\ &= \frac{1}{2m+1} \Big(f_{i+m+1} - f_{i-m} \Big) + u_i \end{aligned}$$

  - total complexity is linear and independent (!) of the mask size:
    1 addition, 1 subtraction, 1 multiplication per pixel (in 1-D)

Lowpass Filters (3)

- Often results with a single box filter do not look too convincing:
  - not rotationally invariant:
    – prefers horizontal and vertical structures
  - not satisfactory in the frequency domain:
    – continuous FT of a box function is a sinc function (Lecture 4)
    – nonmonotone behaviour: Fourier spectrum has multiple extrema
    – attenuation of high freqencies only with $1/|u|$
- However, we will see later that iterated box filtering can be useful.

Lowpass Filters (4)

Optimality in the Fourier Domain: The Ideal Lowpass

- All frequency components (u, v) with $u^2 + v^2 > T^2$ are set to 0.
- not satisfactory in the spatial domain:
  – sinc-like rotation invariant convolution kernel in the spatial domain
  – creates visible ringing artifacts

Lowpass Filters (5)

Gaussian Convolution Kernels

- m-dimensional Gaussian:

$$K_\sigma(\boldsymbol{x}) \;:=\; \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{|\boldsymbol{x}|^2}{2\sigma^2}\right).$$

  The “width” σ is called *standard deviation,* σ^2 is the *variance.*

- creates Gaussian with reciprocal variance in Fourier domain (Lecture 4)
- good compromise: one maximum in both spatial and frequency domain
- the only convolution kernel that is both separable and rotationally invariant
- Iterated Gaussian convolution creates a new Gaussian where the variances sum up.
- Gaussian convolution can be implemented efficiently in numerous ways.

Lowpass Filters (6)

Top left: Noisy original image. **Top right:** Filtering with a 11×11 box filter creates horizontal and vertical artifacts. **Bottom left:** The ideal lowpass with $T^2 = 500$ suffers from ringing artifacts. **Bottom right:** Smoothing with a Gaussian with $\sigma = 3$ gives better results. Author: J. Weickert.

Lowpass Filters (7)

Approximation Possibility 1: Sampling in the Spatial Domain

- exploit separability and symmetry in order to achieve high efficiency
- restrict sampling to interval $[-k\sigma, k\sigma]$ (high accuracy for $k \geq 3$)
- renormalise sum of coefficients to 1
- Advantage: simple and flexible (σ can be tuned continuously)
- Disadvantage: computational complexity increases with σ
- good for small values of σ

Lowpass Filters (8)

Approximation Possibility 2: Multiplication in the Fourier Domain

- cf. Lecture 5:
  - use FFT to transform the image into the Fourier domain
  - multiply with the Fourier transform of the Gaussian
    (a Gaussian with inverse variance)
  - use FFT for backtransformation
- Advantages:
  - almost linear complexity: $\mathcal{O}(N^2 \log N)$ for an $N \times N$ image
  - computational complexity does not increase with σ
- Disadvantages:
  - wraparound errors (unless image is mirrored)
  - standard FFT requires image sizes of powers of 2
- good for large values of σ

Lowpass Filters (9)

Approximation Possibility 3: Binomial Kernels

Binomial kernels and the variance of the approximated Gaussians.

Normalisation	Filter Coefficients	Variance σ^2
1	1	0
1/2	1 1	1/4
1/4	1 2 1	1/2
1/8	1 3 3 1	3/4
1/16	1 4 6 4 1	1
1/32	1 5 10 10 5 1	5/4
1/64	1 6 15 20 15 6 1	3/2
1/128	1 7 21 35 35 21 7 1	7/4
1/256	1 8 28 56 70 56 28 8 1	2

- Binomial kernels approximate Gaussians.
- Separability and symmetry can be exploited.
- Iterated binomial kernels create binomial kernels.

Lowpass Filters (10)

- Advantage:
  - even possible in integer arithmetics:
    division by powers of 2 comes down to bit shifts
- Disadvantages:
  - computational complexity increases with σ
  - σ cannot be tuned continuously

MIA 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Approximation Possibility 4: Iterated Box Filters

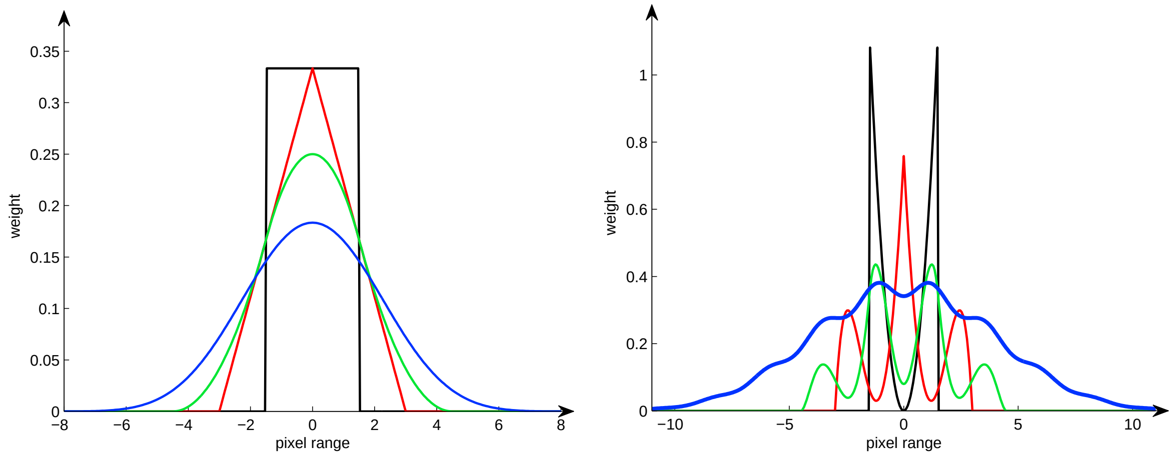


Illustration of the central limit theorem of statistics. **Left:** Iterated box filters approximate a Gaussian. The graph shows the box filter (black) and the results after 1 (red), 2 (green), and 5 (blue) iterations. **Right:** Iterating a more complicated filter also approximates a Gaussian, but the convergence is slower. The graph shows a parabola-shaped filter (black) and the results after 1 (red), 2 (green), and 10 (blue) iterations. Author: T. Schneevoigt.

MIA 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

- central limit theorem of statistics:
  iterated averaging kernels (symmetric, normalised, nonnegative) converge to Gaussians
- Iterating a box filter three times is already a reasonable approximation.
- It can be shown that n iterations of a box filter $(b_i)_{i\in\mathbb{Z}}$ of length $2\ell+1$,

$$b_i = \begin{cases} \frac{1}{2\ell+1} & \text{for } -\ell \le i \le \ell, \\ 0 & \text{else,} \end{cases}$$

  approximate a Gaussian with variance

$$\sigma^2 \;=\; n \cdot \frac{\ell^2+\ell}{3}.$$

- Advantage: linear complexity, independent of σ
- Disadvantage: σ cannot be tuned continuously

Highpass Filters

Goals

- remove low-frequent background perturbations
- perhaps even sharpen blurry image structures by enhancing high frequencies

Remarks

- An important class of highpass filters consists of derivative filters.
  They are useful for detecting edges (next lecture).
- While lowpass filters act stabilising, highpass filters may act destabilising, if they enhance high frequencies.

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Design in the Spatial Domain

- Example: highpass filter as difference between identity and a lowpass filter
- Using e.g. a 3×3 box filter as lowpass filter creates the highpass stencil

$$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \quad - \quad \begin{array}{|c|c|c|}\hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \end{array} \quad = \quad \begin{array}{|c|c|c|}\hline -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \\ \hline -\frac{1}{9} & \frac{8}{9} & -\frac{1}{9} \\ \hline -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \\ \hline \end{array}$$

where the *stencil notation* depicts the weights in the pixels

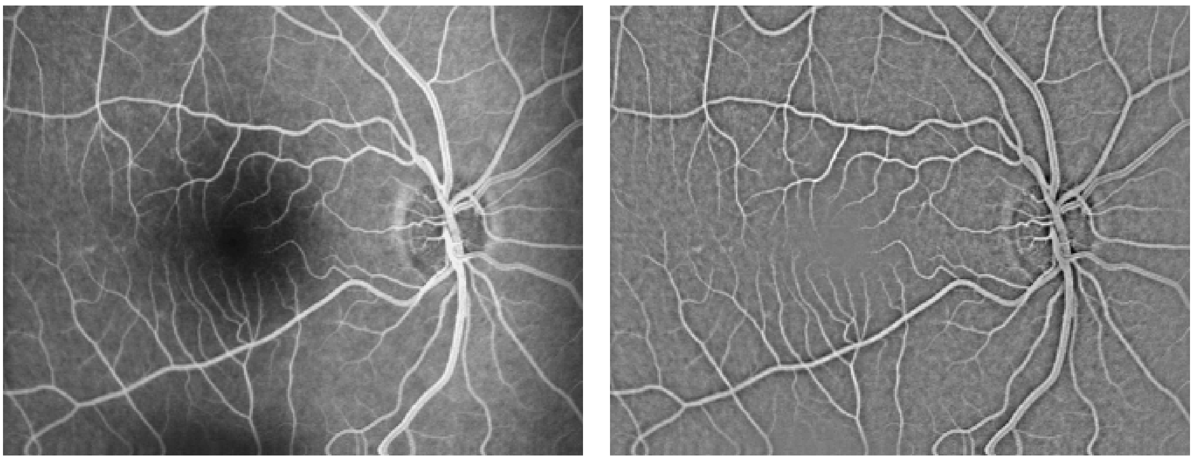
$i-1,\ j+1$	$i,\ j+1$	$i+1,\ j+1$
$i-1,\ j$	$i,\ j$	$i+1,\ j$
$i-1,\ j-1$	$i,\ j-1$	$i+1,\ j-1$

M I A 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Highpass Filters (3)

- Displaying the filtered image often requires an affine rescaling of the grey values (cf. Lecture 10):
  - For many of these filters, the average grey value becomes 0 (e.g. if the lowpass filter preserves the average grey value).
  - Thus, negative values are common.

Highpass Filters (4)



Left: Vessel structure of the background of the eye. **Right:** Elimination of low-frequent background structures by subtracting a Gaussian-smoothed version from the original image. The greyscale range $[-94, 94]$ has been rescaled to $[0, 255]$ by an affine rescaling. Author: J. Weickert.

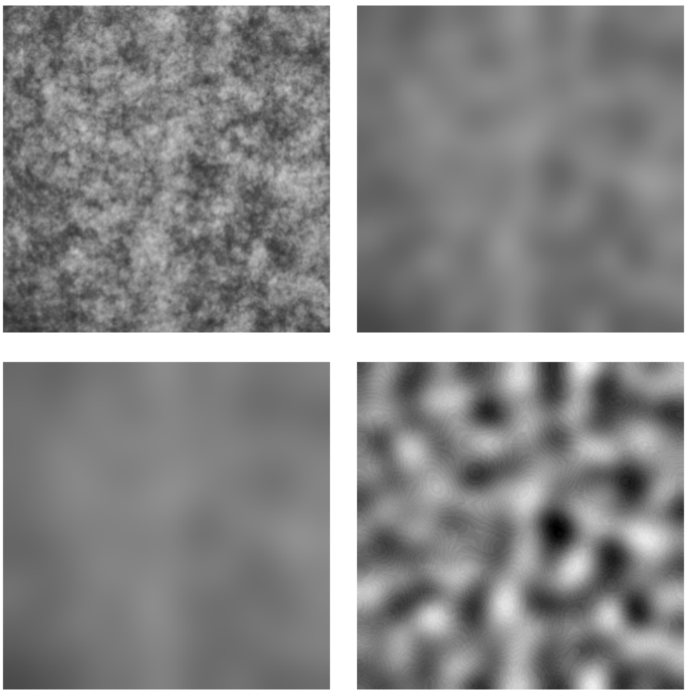
M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Bandpass Filters

- useful for extracting interesting image structures on certain scales
- Example: assessing the cloudiness of fabrics (Lecture 6)
- can be created by subtracting two lowpass filters
- If the lowpass filters are Gaussians, the resulting bandpass is called *DoG (difference of Gaussians).*

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32



(a) Top left: Fabric, 257×257 pixels. **(b) Top right:** After lowpass filtering with a Gaussian with $\sigma = 10$. **(c) Bottom left:** Lowpass filtering with $\sigma = 15$. **(d) Bottom right:** Subtracting (b) and (c) gives a bandpass filter that visualises cloudiness on a certain scale. The greyscale range has been affinely rescaled from $[-13, 13]$ to $[0, 255]$. Author: J. Weickert.

Summary

- Linear shift invariant (LSI) filters are fully characterised by their impulse response.
- can always be represented as convolutions
- impulse response: given by convolution mask
- The Fourier transform is very important for designing LSI filters.
- Lowpass filters are useful for smoothing data:
  - most important example: Gaussian convolution
  - Gaussian convolution can be implemented in many ways, e.g. in the spatial domain, in the Fourier domain, via binomial filters, via iterated box filters
- Highpass filters eliminate low-frequent perturbations and/or sharpen image structures.
- Bandpass filters are mainly used for extracting features at certain scales.

References

- K. R. Castleman: *Digital Image Processing*. Prentice Hall, Upper Saddle River, 1996.
  (a text book that focuses on LSI filters)
- R. C. Gonzalez, R. E. Woods: *Digital Image Processing*. Pearson, Upper Saddle River, Global Edition, 2017.
  (see in particular Sections 3.5–3.7 and Chapter 4)
- A. V. Oppenheim, R. W. Schafer: *Discrete-Time Signal Processing*. Prentice Hall, Englewood Cliffs, Third Edition, 2013.
  (Signal processing books usually provide an exhaustive treatment of LSI filters. This volume is one of the classical text books on signal processing.)
- W. M. Wells: Efficient synthesis of Gaussian filters by cascaded uniform filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 234–239, 1986.
  (Gaussian approximation by iterated box filters)
- P. Getreuer: A survey of Gaussian convolution algorithms. *Image Processing On Line*, Vol. 3, pp. 286–310, 2013.
  (`https://www.ipol.im/pub/art/2013/87/`)
  (compares many algorithms for Gaussian convolution)

M I A
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32