





















# Liste der noch zu erledigenden Punkte

 Layout! . . . . .	2
 so richtig? . . . . .	2
 Wort? . . . . .	2
 Skizze . . . . .	2
 skipped: Very fast intro: Matlab and images . . . . .	2
 motions?P.4 . . . . .	3
 Matlab stuff . . . . .	4
 basis . . . . .	5
 hier fehlt noch das Kronecker underarrow . . . . .	7
 Matlab-Code . . . . .	7
 Layout S.12 u . . . . .	9
 Exercise ?! . . . . .	9
 Layout! . . . . .	17
 Im Skript hier noch Beispiele und soetwas p. 32f . . . . .	17
 Layout! . . . . .	18
 Layout! . . . . .	20
 Kapitel sollte noch fehlergelesen werden. Es könnte noch einiges aus dem Skript über- nommen werden. Es braucht etwas Layout . . . . .	20
 hier fehlt der rest aus einer Vorlesung . . . . .	20
 siehe auch p. 41 . . . . .	20
 allerhand noch im Skript und ein Tafelfoto . . . . .	25

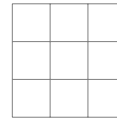
# Kapitel 1

## 1. Overview

- „image society“ (webpages: 1995 text-based, 2005 image based, 2015 video based ...)
  - data transfer rates  $\uparrow$ , compression rates  $\uparrow$
  - critical shift: reading  $\rightarrow$  watching
- „Photoshop“-ing (remove wrinkles, bumps, ...)
- Images in medicine („medical image processing“), x-ray, CT, MRI, ultrasound, ... („modalities“).  
different questions:

### 1.) Layout!

align bottom    measurements  $\xrightarrow{?}$  image  
                  expl: tomography  
                   $\Rightarrow$  difficult mathematical problems



### 2.) Image enhancements

- denoising
  - simple pixels/lines: „sandpaper“ interpolation
  - global noise: smoothing
- grayscale
  - histogramm balancing (spreading)
- distortion
  - makes straight lines (in real world) straight (in the images)
- edge detection
  - contour enhancement
- segmentation
  - detect and separate parts of the image
- registration
  - sequence of images of the same object  $\Rightarrow$  Wort?, compare Skizze
  - $\nearrow$  object following in a movie

so richtig?

### Our Focus:

- mathematical models/methods/ideas
- (algorithms)
- ((implementation))

skipped: Very fast intro: Matlab and images

# Kapitel 2

## 2. What is an image?

### 2.1 Discrete and continuous images

There are (at least) two different points of view:



Abbildung 2.1: Discrete Image

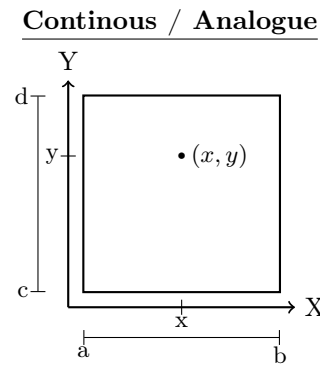


Abbildung 2.2: Continuous Image

**object:** matrix  
**tools:** linear algebra (SVD, ...)  
**pros:** (finite storage) storage, complexity  
**cons:** limitations: zooming, rotations, ...

**function**  
 analysis (differentiation, integrate, ...)  
 freedom, tools, motions? P.4  
 (e.g. edge discontinuity)  
 storage (infinite amount of data)

arguably, one has:

- real life  $\Rightarrow$  continuous „images“ (objects)
- digital cameras  $\Rightarrow$  discrete images

In general we will say:

**Definition 2.1** ((mathematical) image). A (mathematical) *image* is a function

$$u : \Omega \rightarrow F,$$

where:  $\Omega \subset \mathbb{Z}^d$  (discrete) or  $\Omega \subset \mathbb{R}^d$  (continuous) ... *domain*

$d = 2$  (typical case 2D),  $d = 3$  („3D image“ = body or  $\underbrace{2D + \text{time}}_{\text{movie}}$ )

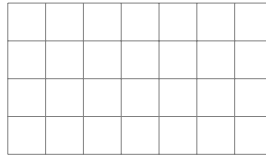
$d = 4$  (3D + time)

$F \dots$  range of colours

$F = \mathbb{R}$  or  $[0, \infty]$  or  $[0, 1]$  or  $\{0, \dots, 255\}$ , ... grayscale (light intensity)

$F \subset \mathbb{R}^3 \dots$  RGB image (colored)

$F = \{0, 1\} \dots$  black/white



3 Layers

$\Rightarrow$  colored images:w

### Matlab stuff

Large parts of the course: analytical approach (i.e. continuous domain  $\Omega$ )

Since we want to differentiate, ... the image  $u$ .

Still: need to assume that also  $F$  is continuous (not as  $\{0, 1\}$ ,  $\{0, 1, \dots, 255\}$  or  $\mathbb{N}$ )

since otherwise the only differentiable (actually, the only continuous) functions  $u : \Omega \rightarrow F$  are *constant* functions  $\Leftrightarrow$  single-colour images

Also: We usually take  $F$  one-dimensional ( $F \subset \mathbb{R}$ ). Think of it as either

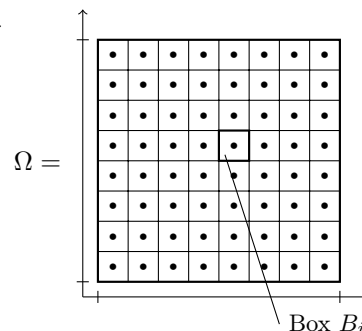
- gray scaled image, or
- treating R,G & B layer separately

## 2.2 Switching between discrete and continuous images

**continuous  $\rightarrow$  discrete:**

- divide the continuous image in small squared pieces (boxes) (superimpose grid)
- now: represent each box by *one* value
  - strategy 1: take function value  $u(x_i)$   
for  $x_i = \text{midpoint of box } B_i$
  - strategy 2: use mean value

$$\frac{1}{|B_i|} \int_{B_i} u(x) dx$$



$\Rightarrow$  discrete image

strategy 1: simple (and quick) but problematic ( $u(x_i)$  might represent  $u|_{B_i}$  badly; for  $u \in L^p$ , single point evaluation not even defined)

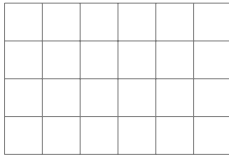
strategy 2: more complex but also more „democratic“ (actually closer to the way how CCD Sensors in digital cameras work)

often the image value of the box  $B_i$  gets also digitized, i.e. fitted (by scaling & rounding) into range  $\{0, 1, \dots, 255\}$

**discrete  $\rightarrow$  continuous**

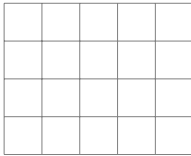
This is of course more tricky ...

- Again: each pixel of the discrete image corresponds to a „box“ of the continuous image (that is still to be constructed)
- Usually: pixel value  $\mapsto$  function value at the *midpoint* of the box
- Question: How to get the other function values (in the box)?



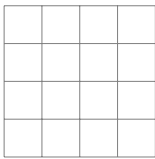
idea 1: just take the function value of the nearest midpoint („nearest neighbour interpolation“)

For each  $x \in B_i : u(x) := u(x_j)$  where  $|x - x_j| = \min_k |x - x_k|$



$\Rightarrow u(x) = u(x_i)$  for all  $x \in B_i$   
 $\Rightarrow$  each box is uni-color  
 $\Rightarrow$  the continuous image is essentially still discrete

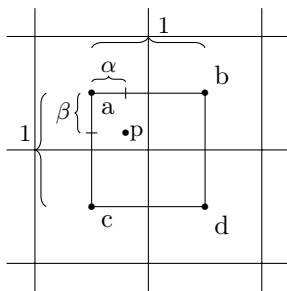
idea 2: (bi-) linear interpolation



Let  $a, b, c, d \dots$  function values at 4 surrounding adjacent midpoints ( $\nearrow$  figure)

$\alpha, \beta, 1 - \alpha, 1 - \beta \dots$  distance to dotted lines ( $\nearrow$  figure, w.l.o.g, bob is  $1 \times 1$ )

interpolation (linear) on the dotted line between  $a$  and  $b$ :



$$e := a + \alpha(b - a) = (1 - \alpha)a + \alpha b$$

(1D - interpolation, convex combination)

Similarly:  $f = (1 - \alpha)c + \alpha d$

Then: The same 1D-interpolation between  $e$  and  $f$

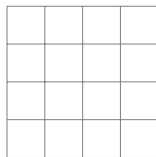
$$\Rightarrow u(x) := (1 - \beta) \cdot e + \beta \cdot f$$

$$\begin{aligned} &= (1 - \beta)[(1 - \alpha)a + \alpha b] + \beta[(1 - \alpha)c + \alpha d] \\ &= \underbrace{(1 - \alpha)(1 - \beta)a}_{\in [0, 1]} + \underbrace{\alpha(1 - \beta)b}_{\in [0, 1]} + \underbrace{(1 - \alpha)\beta c}_{\in [0, 1]} + \underbrace{\alpha\beta d}_{\in [0, 1]} \end{aligned}$$

$\in [0, 1] \wedge \sum = 1$

$\Rightarrow$  convex combination of the function values  $a, b, c, d$  at the the surrounding 4 midpoints (on which points is the nearest, instead of taking just  $a, b, c$  or  $d$  - depending)

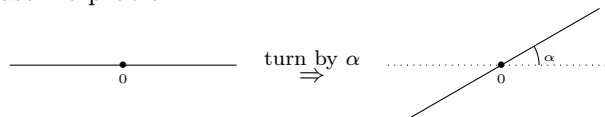
$\Rightarrow$  2D linear interpolation, *bi-linear interpolation* (can be interpreted as spline interpolation with bilinear **basis** splines).



**Beispiel 2.2.** Rotate image

by angle  $\phi \neq k \cdot \frac{\pi}{2}$

- continuous image case: no problem



$$x = D_\varphi y \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, D_\varphi = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix}$$

2D rotation matrix

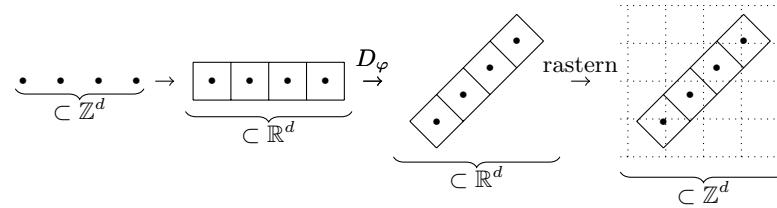
$$y = D_\varphi^{-1} x = D_{-\varphi} x$$

$$\Rightarrow v(x) := u(y) = u(D_{-\varphi} x) \quad \forall x \in \text{domain of the rotated image}$$

- discrete image case: problem !

For  $x \in \text{domain of notated image}$ , in general  $D_{-\varphi} x \notin \text{domain of original image}$ <sup>1</sup>

Way out:  $v(x) := \text{interpolation}$  between the  $u(\cdot)$  of the 4 surrounding pixels of  $D_{-\varphi}$



Something to think about:

What happens in the limit (?) if we, starting with an image (discrete or continuous), repeatedly switch between discrete and continuous, non-stop ... ?

Does the answer depend on the way of switching ? (continuous  $\rightarrow$  discrete: midpoint or average, discrete  $\rightarrow$  continuous: nearest neighbour or bilinear?)

---

<sup>1</sup>it's not an integer

# Kapitel 3

## 3. Histogramm and first applicatsion

### 3.1 The histogramm

**Definition 3.1** (histogram). Let  $\Omega \subset \mathbb{Z}^d$ ,  $F \subset \mathbb{R}$  discrete and  $u : \Omega \rightarrow F$  a discrete discrete image. The function

$$H_u : F \rightarrow \mathbb{N}_0 \quad (:= \mathbb{N} \cup \{0\})$$

with

$$H_u(k) := \# \{x \in \Omega : u(x) = k\}, \quad k \in F$$

is called *histogramm* of the image  $u$ .

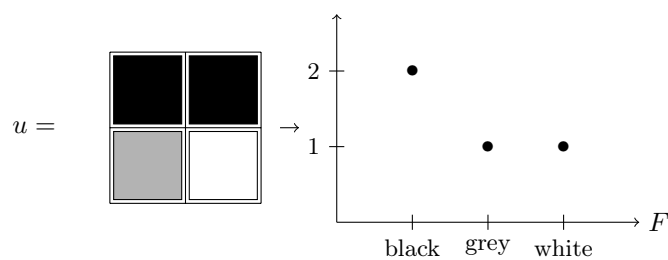
$H_u(k)$  counts how often colour  $k$  appears in  $u$ .

$$\sum_{k \in F} H_u(k) = |\Omega| = \text{number of pixels in the whole image}$$

or

$$\frac{H_u(k)}{|\Omega|} = \begin{array}{l} \text{relative frequency of colour } k \text{ in image } u \\ \text{(relative H\u00e4ufigkeit)} \end{array}$$

**Beispiel 3.2.**



If  $u$  is a continuous image,  $H_u$  can be understood as a measure (generalized function)<sup>1</sup>.

Another way to write this:

$$H_u(k) = \sum_{x \in \Omega} \delta_{u(x)}(k), \quad k \in F \qquad H_u(k) = \int_{\Omega} \delta_{u(x)}(k) dx, \quad k \in F$$

hier fehlt noch das Kronecker underarrow

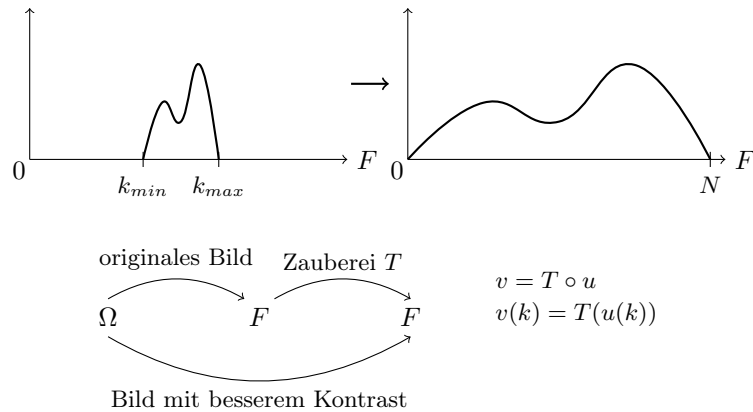
Matlab-Code

<sup>1</sup>density of a probability distribution

### 3.2 Application: contrast enhancement

If the image only uses a small part of the available colour/grayscale „palette“  $F$ , then its contrast can be improved by „spreading“ the histogram over all of  $F$ .

Simple idea:



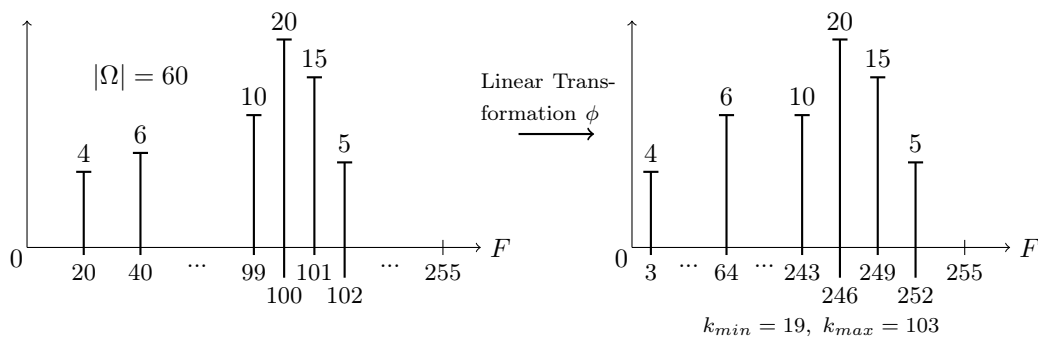
5

The above simple idea („contrast stretching“) corresponds to

$$\begin{aligned} \varphi : k_{\min} &\mapsto 0 \\ k_{\max} &\mapsto N \\ &\text{and linear in between} \\ \text{i.e.} \quad \varphi(k) &= \left[ \frac{k - k_{\min}}{k_{\max} - k_{\min}} \cdot N \right] \end{aligned}$$

Where  $[ \cdot ]$  means ...rounding to the nearest integer (assuming that  $F = \{0, 1, \dots, N\}$ ).

Example histogram:



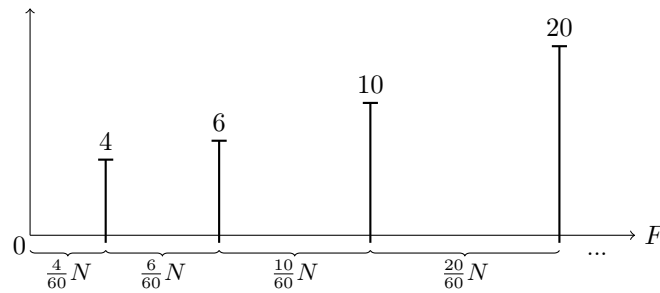
A bit more sophisticated:

$$\begin{aligned} \varphi : (k_{\min} &\mapsto 0) \\ k_{\max} &\mapsto N \\ &\text{and **non** linear in between} \end{aligned}$$

such that colour ranges that occur more frequently in  $u$  can occupy a larger range of colours in  $v$ .  
 ( $\Rightarrow$  visibility  $\uparrow$ )

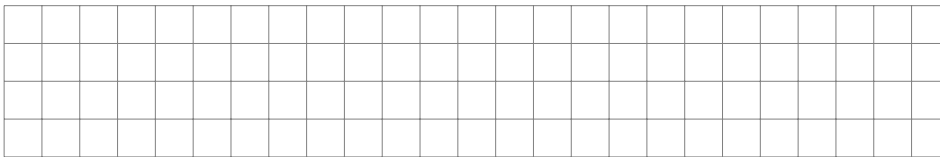
Example histogram spread out according to frequency of occurrence:





$\Rightarrow$  „density“ is equalized over  $F = \{0, \dots, N\}$

Ideal would be:



#### Layout S.12 u

Note: The new colours (i.e the location of the bars in the histogram of  $u$ ) only depend on the frequencies / height of the bars in  $H_u$  but not on the colours/location of the bars in  $H_u$

Finally: The formula

$$\varphi(k) = \left\lceil \frac{N}{|\Omega|} \sum_{l=0}^k H_u(l) \right\rceil$$

This process is called „histogramm equalization“

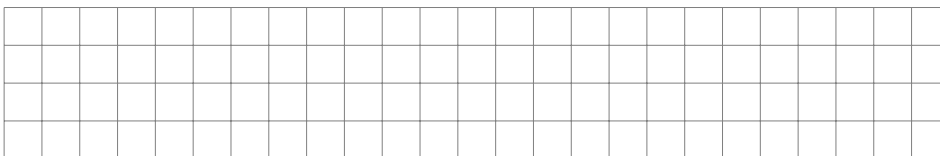
#### Exercise ?!

### 3.3 Another application: conversion to b/w

Task: convert grayscale image to black white

- interesting for object detection/*segmentation* ...!

Idea: Find a threshold  $t \in T$  s.t. the histogram splits into two „characteristic“ parts



For  $t \in F$  put

$$\text{black} := \{k \in F : k \leq t\}$$

$$\text{white} := \{k \in F : k > t\}$$

and

$$\tilde{u} := \begin{cases} 0, & u(x) \in \text{black} \\ 1, & u(x) \in \text{white} \end{cases} \quad \tilde{F} = \{0, 1\}$$

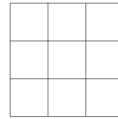
How to find the threshold  $t$ :

## 1.) Shape based methods

If the histogram is „biomodal“

$$\text{Put } t := \frac{k_{\max_1} + k_{\max_2}}{2}$$

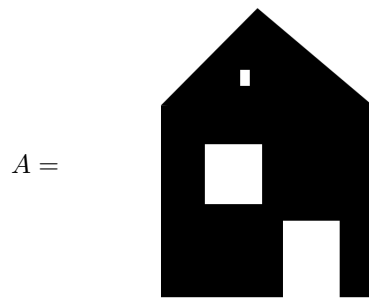
or  $t := k_{\min}$



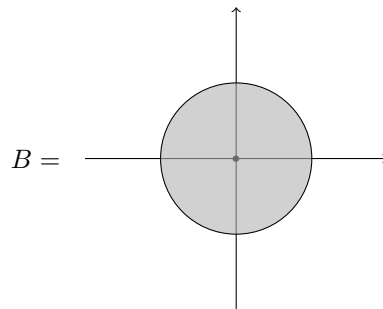
# Kapitel 4

## 4. Basic Morphological Operations

B/W Bild:



Structural element :



### 4.1 Operations on A and B

$$A + B := \{a + b : a \in A, b \in B\}$$

This is called dilation.

You might imagine that at every dark point in the image  $A$  the Structurelement is applied.

$$A + B =$$

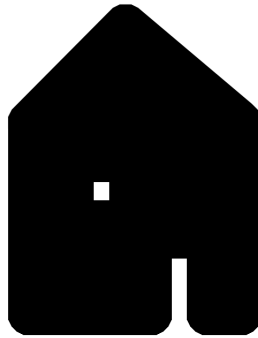


Image created in Matlab through:

---

```

1 I=imread('Bild1.png');
2 se=strel('disk',40,8);
3 I2=imcomplement(imdilate(imcomplement(I),se));%I am using the complement of the image
   here so that the structural element is applied to the dark parts of the image
4 imshow(I2);

```

---

$$A - B := \{a : a + B \subset A\}$$

This is called erosion.

You can imagine that you search for the points in which the structural element fits.

$$A - B =$$

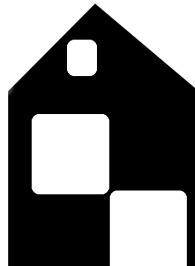


Image created in Matlab thorough:

---

```

1 I=imread('Bild1.png');
2 se=strel('disk',20,8);
3 I2=imcomplement(imerode(imcomplement(I),se));
4 imshow(I2);

```

---

One may quickly realize that  $A \neq (A + B) - B$ , so a new Operation is introduced:

$$A \bullet B := (A + B) - B$$

This is called closing and is used to e.g. remove noise. In the example image you might notice that the upper window is missing.

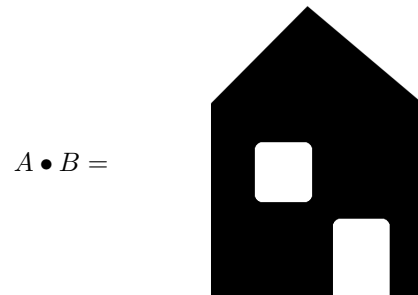


Image created in Matlab thorough:

---

```

1 I=imread('Bild1.png');
2 se=strel('disk',20,8);
3 I2=imcomplement(imdilate(imcomplement(I),se));
4 I3=imcomplement(imerode(imcomplement(I2),se));
5 imshow(I3);

```

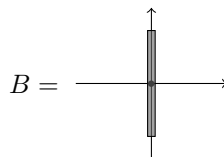
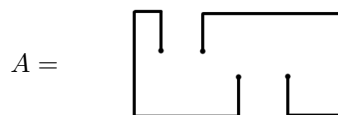
---

The inverse also exists:

$$A \circ B := (A - B) + B$$

This is called opening.

This time with a new example:



$$A \circ B = \begin{array}{cccc} | & | & | & \\ | & & & \\ & & & | \\ & & | & | \\ & & & | \end{array}$$

Image created in Matlab thorough:

---

```

1 I=imread('Bild2.png');
2 se=strel('line',10,90);
3 I2=imcomplement(imerode(imcomplement(I),se));
4 I3=imcomplement(imerode(imcomplement(I2),se));
5 imshow(I3);

```

---

# Kapitel 5

## 5. Entauschen: Filter und Co

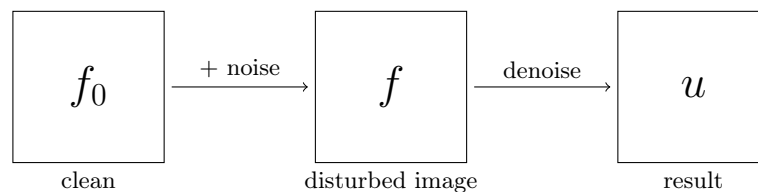
### 5.1 Noise

Noise = Unwanted disturbances in an image. Mostly because of

- point wise
- random
- independent

We consider *noise* to be an additive disturbances (for multiplicative noise use *log*).

*Notation:*



The quality of the denoised image  $u$  compared to the original image  $f_0$  is described by norms:

$$\begin{aligned}
 \|f - f_0\| &\dots \text{noise} \\
 \|u - f_0\| &\dots \text{absolute error} \\
 \frac{\|u - f_0\|}{\|f - f_0\|} &\dots \text{relative error compared to the noise} \\
 \frac{\|u - f_0\|}{\|f_0\|} &\dots \text{relative error compared to the signal}
 \end{aligned}$$

Typically the chosen norm is:

$$\|f\| = \|f\|_2 = \sqrt{\int_{\Omega} |f(x)|^2 dx}$$

or in the discrete:

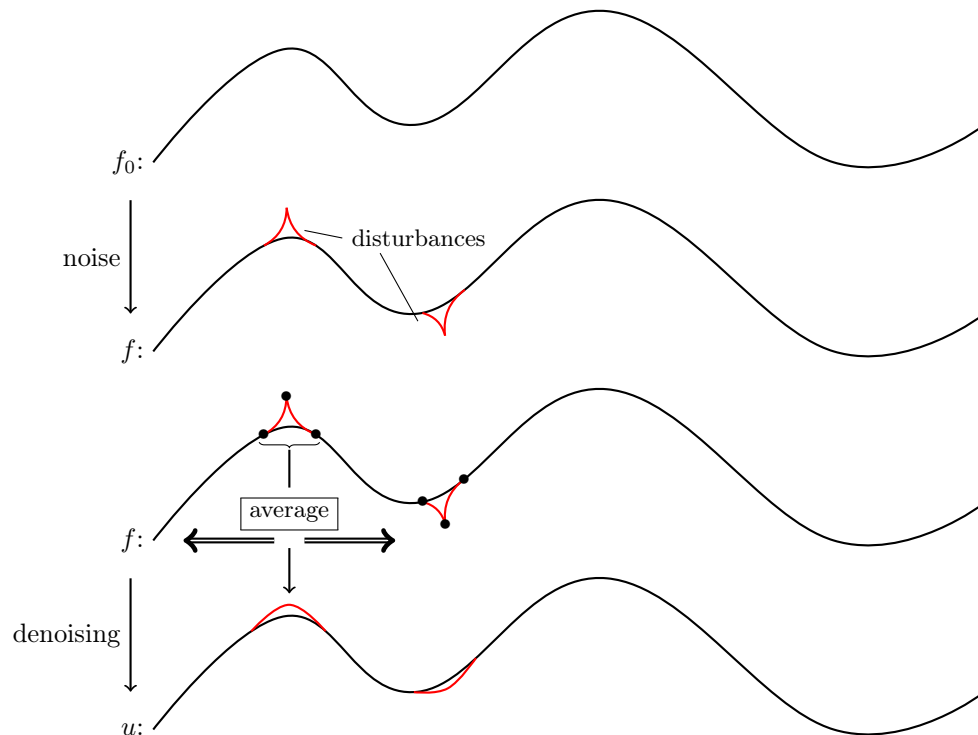
$$\|f\|_2 = \sqrt{\sum_{x \in \Omega} |f(x)|^2}$$

Closely connected is the Signal to noise ratio (SNR):

$$\log\left(\underbrace{\frac{\|f_0\|_2}{\|u - f_0\|_2}}_{\in [1, \infty)}\right) \in [0, +\infty), \text{ where } 0 \text{ is bad and } +\infty \text{ is good.}$$

## 5.2 smoothing filter

Idea: (to simplify in 1D)

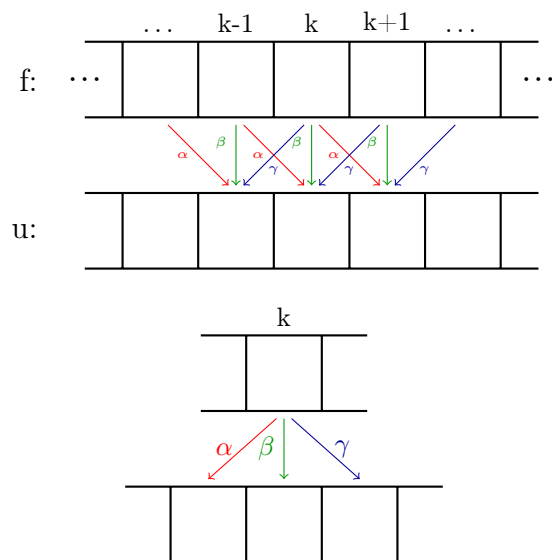


$$u(k) := \alpha \cdot f(k-1) + \beta \cdot f(k) + \gamma \cdot f(k+1) \quad (5.1)$$

where:

$$\alpha + \beta + \gamma = 1 \quad (5.2)$$

More precisely (5.1) means:



With (5.1) there is a mapping  $f \mapsto u$ , we write

$$u = m \boxtimes f, \text{ this is called } \underline{\text{Correlation}} .$$



where:

$$(m \boxtimes f)(k) = \sum_{i \in \text{supp}(m)} m(i) f(k+i) \quad (5.3)$$

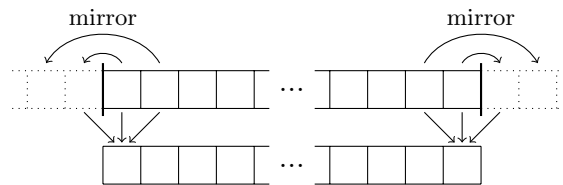
and:

$$m = \begin{array}{ccccc} & \dots & -1 & 0 & 1 & \dots \\ \dots & \alpha & \beta & \gamma & \dots & \text{called } \underline{\text{mask}} \end{array}$$

If you set  $j := k + i$  in (5.1), then  $i = j - k$ , which means:

$$(m \boxtimes f)(k) = \sum_{i \in \text{supp}(m)} m(j-k) f(j) \quad (5.4)$$

To apply the mapping onto the boundary the image is reflected, in 1D:



in 2D:

d	b	d
q	P	q
d	b	d

Formula (5.4) might remind one of the convolution :

Layout!

$$(g * f)(k) = \sum_{j \in \mathbb{Z}} g(\underbrace{k-j}_{\text{Difference to (5.4)}}) \cdot f(j) \quad (5.5)$$

If you set  $g(i) := m(-i) =: \tilde{m}(i)$ , which corresponds to a reflection of the Mask, then

$$m \boxtimes f = g * f = \tilde{m} * f$$

Im Skript hier noch Beispiele und soetwas p. 32f

Properties of the convolution:

1.  $(f * g) * h = f * (g * h)$ , Associativity
2.  $f * g = g * f$ , Commutativity
3.  $\tilde{f} * \tilde{g} = \widetilde{f * g}$ , Compatibility with reflection

Properties of the correlation:

1.  $f \boxtimes (g \boxtimes h) = \tilde{f} * (\tilde{g} * h) \stackrel{\boxed{1}}{=} (\tilde{f} * \tilde{g}) * h \stackrel{\boxed{3}}{=} (\widetilde{f * g}) * h = (f * g) \boxtimes h \neq (f \boxtimes g) \boxtimes h$ , not associative!
2.  $f \boxtimes g = \tilde{f} * g \stackrel{\boxed{2}}{=} g * \tilde{f} = \tilde{\tilde{g}} * \tilde{f} \stackrel{\boxed{3}}{=} (\widetilde{\tilde{g} * f}) = \widetilde{g \boxtimes f} \neq g \boxtimes f$ , not commutative!
3.  $\tilde{f} \boxtimes \tilde{g} = \tilde{\tilde{f} * \tilde{g}} \stackrel{\boxed{3}}{=} \widetilde{(\tilde{f} * g)} = \widetilde{f \boxtimes g}$ , Compatibility with reflection

$$\boxtimes \text{ und } * \text{ definiert man auf: } \ell^1(\mathbb{Z}^d) := \left\{ f = (f_i)_{i \in \mathbb{Z}^d} : \underbrace{\sum_{i \in \mathbb{Z}^d} |f_i|}_{:= \|f\|_1} < \infty \right\}$$

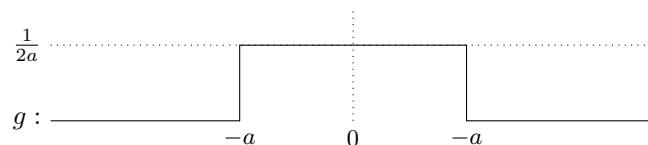
Man kann zeigen (Übung):  $f, g \in \ell^1 \Rightarrow f * g \in \ell^1$  und  $\|f * g\|_1 \leq \|f\|_1 \cdot \|g\|_1$ . Wobei oft die Gleichheit gilt.

Alles gilt auch in der Kontinuierlichen Version:

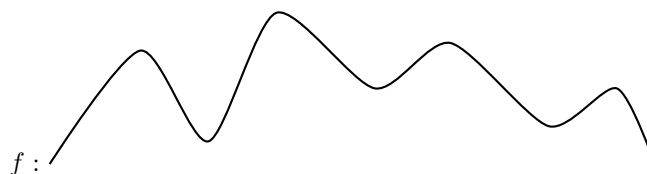
$$L^1(\mathbb{R}^d) := \left\{ f : \mathbb{R}^d \rightarrow \mathbb{R} : \underbrace{\int_{\mathbb{R}^d} |f| dx}_{:= \|f\|_1} < \infty \right\}$$

$$f, g \in L^1(\mathbb{R}^d) : (g * f)(x) = \int_{\mathbb{R}^d} g(x - y) f(y) dy, \quad y, x \in \mathbb{R}^d$$

Beispiel für den kontinuierlichen Fall:



Hierbei gilt  $\int_{\mathbb{R}} g(x) dx = 1$



$g \boxtimes f = \underline{\text{gleitendes Mittel}}$ .



## Layout!

Weitere Eigenschaften der Faltung:

Für alle  $f, g \in L^1$  or  $\ell^1$

$$\left. \begin{aligned} (g_1 + g_2) * f &= (g_1 * f) + (g_2 * f) \\ (\alpha g) * f &= \alpha(g * f) \end{aligned} \right\} = \text{Linearität}$$

Somit ist:

$$g \mapsto f * g$$

ein linearer Operator.

Formt  $\ell^1$  bzw.  $L^1$  eine Algebra mit neutralem Element  $\delta$ ?

$\ell^1$ ?:

$$\delta: \quad \cdots \quad \boxed{0} \quad \boxed{0} \quad \boxed{1} \quad \boxed{0} \quad \boxed{0} \quad \cdots$$

↑  
Pos 0

Ja!

$L^1$ ?: Für ein solches Element muss gelten:

$$\forall f \in L^1 : d * f = f$$

$$\forall x \in \mathbb{R} : \int_{\mathbb{R}^d} \underbrace{\delta(x-y)}_{=0 \forall x \neq y} f(y) dy = f(x)$$

Diese Funktion wird Dirac-Impuls genannt ist aber kein Element von  $L^1$ .

Nun zu Masken in 2D:

$$u = m \boxtimes f \text{ mit } m = \begin{array}{|c|c|c|} \hline \alpha & & \\ \hline \beta & \gamma & \delta \\ \hline \epsilon & & \\ \hline \end{array}$$

wobei  $\alpha + \beta + \gamma + \delta + \epsilon = 1$

Kurzschreibweise:  $u_{ij} := u(x)$  wobei  $x = \begin{pmatrix} i \\ j \end{pmatrix} \in \mathbb{Z}^2$ , analog für  $f_{ij}$ .

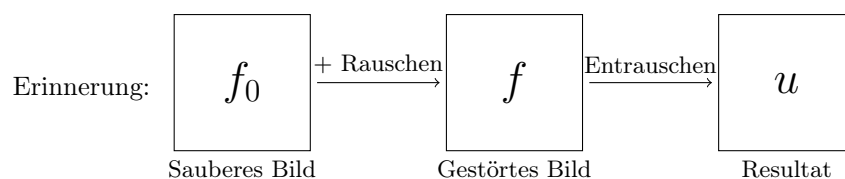
$$\Rightarrow u_{ij} = \alpha f_{i-1,j} + \beta f_{i,j-1} + \gamma f_{ij} + \delta f_{i,j+1} + \epsilon f_{i+1,j}$$

$$u = m \boxtimes f = \tilde{m} * f \text{ mit } \tilde{m} = \begin{array}{|c|c|c|} \hline & \epsilon & \\ \hline \delta & \gamma & \beta \\ \hline & \alpha & \\ \hline \end{array}$$

Symmetrischer Fall:

$$\tilde{m} = \begin{array}{|c|c|c|} \hline \alpha & & \\ \hline \alpha & \gamma & \alpha \\ \hline & \alpha & \\ \hline \end{array} \text{ mit } \gamma = 1 - 4\alpha$$

$$u_{ij} = (1 - 4\alpha)f_{ij} + \alpha(f_{i-1,j} + f_{i,j-1} + f_{i,j+1} + f_{i+1,j}) \quad (5.6)$$



Annahme:  $f_{ij} = f_{ij} + r_{ij}$  mit  $r_{ij} \sim N(0, \sigma^2)$  iid.

z.z.:  $\text{Var}(u_{ij}) \leq \text{Var}(f_{ij})$

$$\text{Var}(f_{ij}) = E(\underbrace{f_{ij} - \overbrace{E f_{ij}}^{f_{ij}^0}}_{r_{ij}})^2 = \sigma^2$$

Layout!

$$\begin{aligned} \text{Var}(u_{ij}) &= E(u_{ij} - E u_{ij})^2 = E((1 - 4\alpha)(\underbrace{f_{ij} - f_{ij}^0}_{r_{ij}}) + \alpha(\underbrace{(f_{i-1,j} - f_{i-1,j}^0)}_{r_{i-1,j}}) + \dots + \underbrace{(f_{i+1,j} - f_{i+1,j}^0)}_{r_{i+1,j}}))^2 \\ &= E((1 - 4\alpha)^2 r_{ij}^2 + \alpha^2(r_{i-1,j}^2 + r_{i,j-1}^2 + r_{i,j+1}^2 + r_{i+1,j}^2) + 2(1 - 4\alpha)\alpha r_{ij} r_{i-1,j} \dots) \\ &= (1 - 4\alpha)^2 \underbrace{E r_{ij}^2}_{\sigma^2} + \alpha^2(E r_{i-1,j}^2 + \dots + E r_{i+1,j}^2) + 2(1 - 4\alpha)\alpha \underbrace{E(r_{ij} r_{i-1,j})}_{\underbrace{E r_{ij} E r_{i-1,j}}_0} + \underbrace{\dots}_0 \\ &= (1 - 4\alpha)^2 \sigma^2 + \alpha^2 4\sigma^2 = (1 - 8\alpha + 16\alpha^2 + 4\alpha^2) \sigma^2 \end{aligned}$$

Da  $0 \leq \alpha$  und  $0 \leq 1 - 4\alpha \Rightarrow 0 \leq \alpha \leq \frac{1}{4}$ :

$$(1 - 8\alpha + 16\alpha^2 + 4\alpha^2) \sigma^2 = 1 + \underbrace{\underbrace{20\alpha}_{\geq 0} \underbrace{(\alpha - \frac{2}{5})}_{< 0}}_{\leq 1}$$

$\Rightarrow \text{Var}(u_{ij}) \leq \text{Var}(f_{ij})$  für  $\alpha \in [0, \frac{1}{4}]$

Dabei gilt:  $\text{Var}(u_{ij}) \xrightarrow{\alpha} \min \iff 1 - 8\alpha + 20\alpha^2 \xrightarrow{\alpha} \min \iff -8 + 40\alpha = 0 \iff \alpha = \frac{1}{5}$

$$\Rightarrow \text{bester Filter : } \begin{array}{|c|c|c|} \hline & \frac{1}{5} & \\ \hline \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \hline & \frac{1}{5} & \\ \hline \end{array}$$

Kapitel sollte noch fehlergelesen werden. Es könnte noch einiges aus dem Skript übernommen werden. Es braucht etwas Layout

## 5.3 Frequenzfilter

Ansatz: Rauschen  $\approx$  hochfrequente Anteile des Bildes/Signals

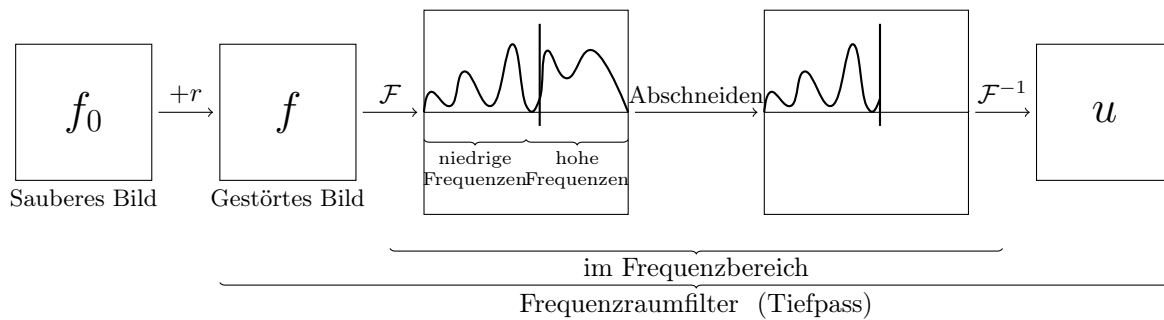
$\Rightarrow$  gezieltes entfernen

Wichtiges Instrument: Fouriertransformation (FT)

$$\mathcal{F} : f \mapsto \hat{f} \text{ mit } \hat{f}(z) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} dx$$

hier fehlt der rest aus einer Vorlesung

siehe auch p. 41



Wobei  $z \in \mathbb{R}^d, f \in L^1(\mathbb{R}^d)$ .

Falls auch  $\hat{f} \in L^1(\mathbb{R}^d)$  ist, dann lässt sich  $f$  wie folgt mittels der inversen Fouriertransformation aus  $\hat{f}$  rekonstruieren:

$$\mathcal{F}^{-1} : \hat{f} \mapsto f$$

$$\hat{f}(z) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} f(x) e^{i\langle z, x \rangle} dx \quad (5.7)$$

Wobei  $x \in \mathbb{R}^d$ .

Man hat also  $\mathcal{F}^{-1} \mathcal{F} f$ , d.h.

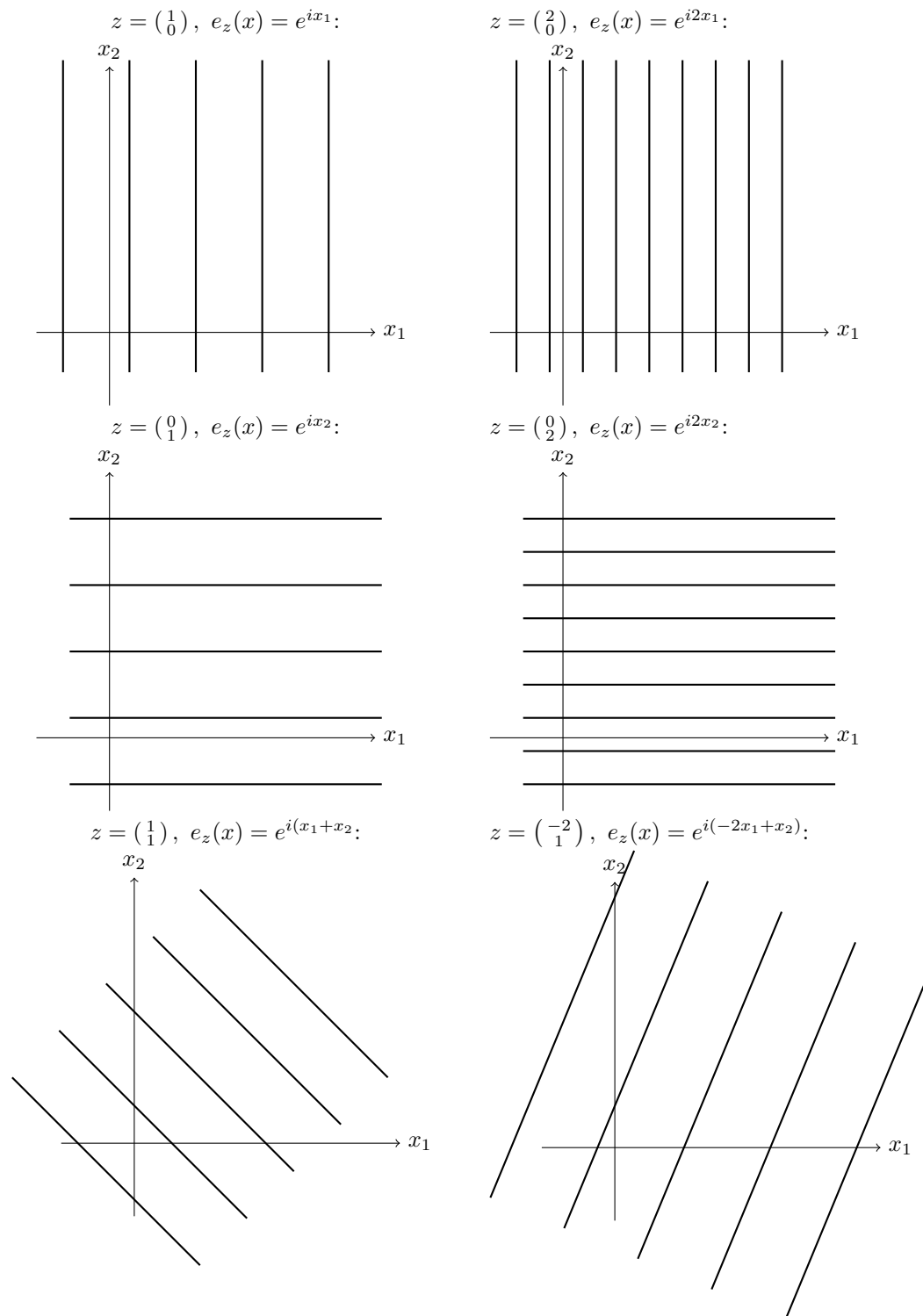
$$f(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \left( \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} f(y) e^{-i\langle z, y \rangle} dy \right) e^{i\langle z, x \rangle} dz$$

Sei nun  $e_z(x) := e^{i\langle z, x \rangle}$ ,  $x \in \mathbb{R}^d$  mit Parameter  $z = \begin{pmatrix} z_1 \\ \vdots \\ z_d \end{pmatrix}$ .

Also  $e_z(x) = e^{i\langle \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}, \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rangle} = e^{i(z_1 x_1 + z_2 x_2)}$

Beispiele in 2D:

(Hier stellen die Linien, Punkte mit konstantem Wert dar)



$f \in L^2(\mathbb{R}^d) = \{f : \mathbb{R}^d \rightarrow \mathbb{R} \mid \int_{\mathbb{R}^d} |f|^2 dx < \infty\}$  ist

- ein normierter Raum mit  $+$ ,  $\alpha \cdot$  und  $\| \cdot \|_2 := \sqrt{\int_{\mathbb{R}^d} |f(x)|^2 dx}$
- ein Skalarproduktraum mit  $\langle f, g \rangle := \int_{\mathbb{R}^d} f \bar{g} dx$ , wobei  $\|f\|_2^2 = \langle f, f \rangle$
- ein vollständiger Raum, also Banachraum

Ein vollständiger normierter Banachraum mit Skalarprodukt heißt Hilbertraum.  
 $\mathcal{F}$  kann auch als Abbildung auf  $L^2(\mathbb{R}^d)$  betrachtet werden. Dann gilt:

$$\hat{f} = \mathcal{F}f \in L^2(\mathbb{R}^d)$$

und

$$\|\hat{f}\|_2 = \|f\|_2 \quad (5.8)$$

und sogar

$$\langle \hat{f}, \hat{g} \rangle_2 = \langle f, g \rangle_2 \quad (5.9)$$

für alle  $f, g \in L^2(\mathbb{R}^d)$ .

Weitere Eigenschaften der Fouriertransformation:

- $f \in L^1(\mathbb{R}^d) \Rightarrow \hat{f}$  stetig und  $\lim_{|z| \rightarrow \infty} \hat{f}(z) = 0$
- $\mathcal{F} : L^1(\mathbb{R}^d) \rightarrow C(\mathbb{R}^d)$  ist eine lineare Abbildung
- $\mathcal{F} : L^1(\mathbb{R}^d) \rightarrow C(\mathbb{R}^d)$  ist eine beschränkte/stetige Abbildung
- Verschiebung  $\xrightarrow{\mathcal{F}}$  Modulation, d.h.

$$g(x) = f(x + a) \Rightarrow \hat{g}(z) = e^{i\langle a, z \rangle} \hat{f}(z)$$

- Modulation  $\xrightarrow{\mathcal{F}}$  Verschiebung, d.h.

$$g(x) = e^{i\langle x, a \rangle} f(x) \Rightarrow \hat{g}(z) = \hat{f}(z - a)$$

- Skalierung  $\xrightarrow{\mathcal{F}}$  inverse Skalierung, d.h.

$$g(x) = f(cx) \Rightarrow \hat{g}(z) = \frac{1}{|c|} \hat{f}\left(\frac{z}{|c|}\right)$$

- Konjugation:  $g(x) = \overline{f(x)} \Rightarrow \hat{g}(z) = \overline{\hat{f}(-z)}$   
 Folglich:  $f$  reelwertig  $\Rightarrow \hat{f}(z) = \overline{\hat{f}(-z)}$

-

$$\text{Grundmode: } \hat{f}(0) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} f(x) dx$$

$$\text{Analog: } f(0) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \hat{f}(x) dx$$

- Differentiation  $\xrightarrow{\mathcal{F}}$  Multiplikation mit Potenzen von  $z$ , d.h.

$$g(x) = \frac{\partial^{\alpha_1 + \dots + \alpha_d}}{\partial x_1^{\alpha_1} \dots \partial x_d^{\alpha_d}} f(x) \Rightarrow \hat{g}(z) = i^{\alpha_1 + \dots + \alpha_d} z_1^{\alpha_1} \dots z_d^{\alpha_d} \hat{f}(z)$$

- Umkehrung des letzten Punktes:

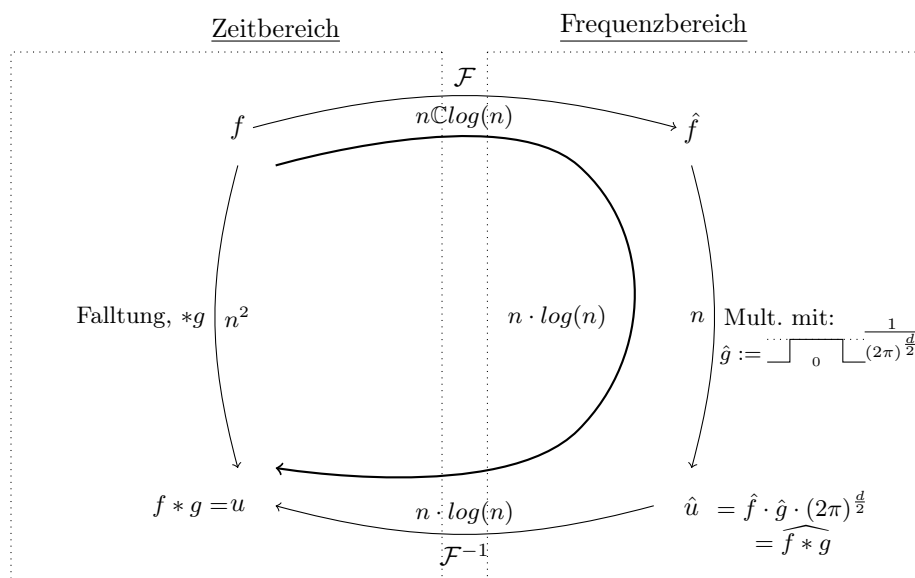
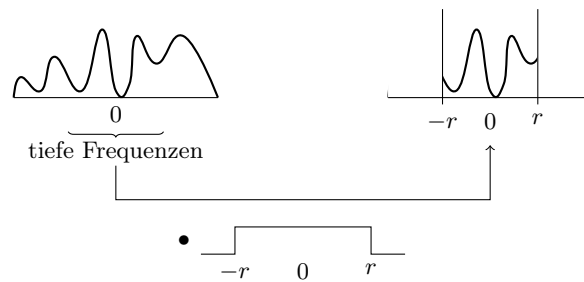
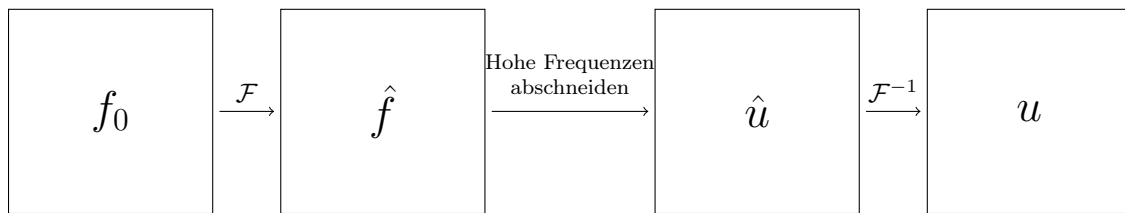
$$g(x) = x_1^{\alpha_1} \dots x_d^{\alpha_d} f(x) \Rightarrow \hat{g}(z) = i^{\alpha_1 + \dots + \alpha_d} \frac{\partial^{\alpha_1 + \dots + \alpha_d}}{\partial x_1^{\alpha_1}} \hat{f}(z)$$

-

$$\text{Faltungssatz: } \mathcal{F}(f * g) = (2\pi)^{\frac{d}{2}} \mathcal{F}(f) \cdot \mathcal{F}(g), \widehat{f * g} = (2\pi)^{\frac{d}{2}} \hat{f} \cdot \hat{g}$$

$$\text{Analog: } \mathcal{F}(f \cdot g) = \frac{1}{(2\pi)^{\frac{d}{2}}} \mathcal{F}(f) * \mathcal{F}(g), \widehat{f \cdot g} = \frac{1}{(2\pi)^{\frac{d}{2}}} \hat{f} * \hat{g}$$

d.h.: Faltung  $\xrightarrow{\mathcal{F}}$  Multiplikation und umgekehrt

**Zur Erinnerung:**

Genauer:

$$\begin{aligned}
 \mathcal{F}u &= \hat{v} = \\
 g(x) &= \frac{1}{(2\pi)^{\frac{d}{2}}} (\mathcal{F}^{-1} \chi_{[-r,r]^d})(x) \\
 &= \frac{1}{(2\pi)^{\frac{d}{2}}} \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \chi_{[-r,r]^d}(z) e^{i\langle z, x \rangle} dz \\
 &\stackrel{1d}{=} \frac{1}{2\pi} \int_{-\infty}^{\infty} \chi_{[-r,r]} e^{izx} dz \\
 &= \frac{1}{2\pi} \int_{-r}^r e^{izx} dz = \frac{1}{2\pi} \left. \frac{e^{izx}}{ix} \right|_{z=-r}^r \\
 &= \frac{1}{2\pi ix} (e^{irx} - e^{-irx}) = \frac{1}{\pi x} \sin(rx) \\
 \hat{g}(0) &= (\mathcal{F}g)(0) = \frac{1}{2}
 \end{aligned}$$



Es ist zu bemerken, dass  $g$  eine Art Tensor Struktur besitzt, was in etwa bedeutet das sich die Funktion in beliebigen Dimensionen als Produkt der Funktion in einer Dimensionen darstellen lässt.

Gauß-Kern :

$$\begin{aligned} G(x) &= \frac{1}{(2\pi)^{\frac{d}{2}}} e^{-\frac{|x|^2}{2}} \Rightarrow G\left(\begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}\right) = \frac{1}{(2\pi)^{\frac{d}{2}}} e^{-\frac{x_1^2 + x_2^2 + \dots + x_d^2}{2}} \\ &= \left(\frac{1}{(2\pi)^{\frac{1}{2}}} e^{-\frac{x_1^2}{2}}\right) \cdot \dots \cdot \left(\frac{1}{(2\pi)^{\frac{1}{2}}} e^{-\frac{x_d^2}{2}}\right) = G(x_1) \cdot \dots \cdot G(x_d) \end{aligned}$$

allerhand noch im Skript und ein Tafelfoto

## 5.4 Filterbreite und Glättung

klar ist:  $\frac{1}{25}$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

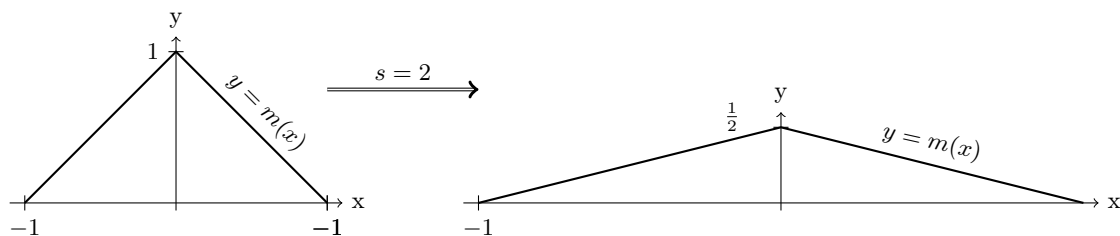
'glättet mehr als':  $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Im Kontinuierlichen: Sei  $m \in L^1(\mathbb{R}^d)$  und  $s > 0$ . Setze

$$m_s(x) := \frac{1}{s^d} m\left(\frac{x}{s}\right), \quad x \in \mathbb{R}^d$$

Bsp (in  $d = 1$ ):



Bsp: Gauß-Kern  $G(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} e^{-\frac{|x|^2}{2}}$

Skalierung mit Fehler  $s > 0$

$$\Rightarrow G_s(x) = \frac{1}{s^d} G\left(\frac{x}{s}\right) = \frac{1}{s^d} \frac{1}{(2\pi)^{\frac{d}{2}}} e^{-\frac{|x|^2}{2s^2}} = \frac{1}{(2\pi s^2)^{\frac{d}{2}}} e^{-\frac{|x|^2}{2s^2}}$$

Skalierung  $s \hat{=}$  Standardabweichung  $\sigma$