


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 übernommen 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
hier noch mehr im Skript p. 45	25
noch einmal schauen was 5.10 ist	27
Ab hier Livetex 24.11	28
Vergleich kontinuierlicher mit dem diskreten Fall.	28
...	29
Bild zu isotrop und anisotrop. (Kann man sich sparen?)	29
..	29
)	29
Ab hier livetex	29
Wiederholung von letzter Woche.	29
unterpunkte	29
bild	29
oder Mischunngen	29

 Bild	30
--	-------	----

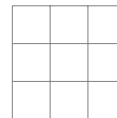
Kapitel 1

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:

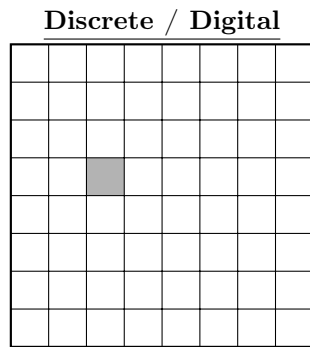


Abbildung 2.1: Discrete Image

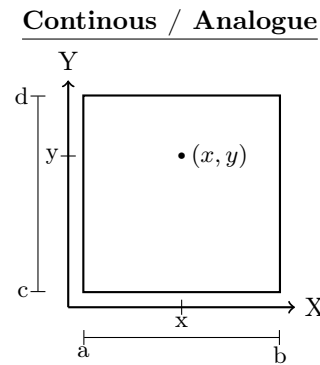


Abbildung 2.2: Continuous Image

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 $\underbrace{2D + \text{time}}_{\text{movie}}$)

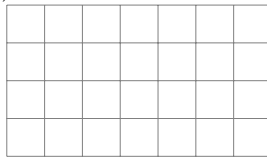
$d = 4$ (3D + time)

$F \dots$ 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 Ω)

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

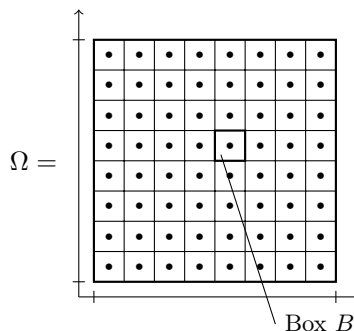
- gray scaled image, or
- treating R,G & B layer separately

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)

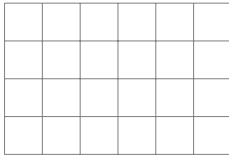
strategy 2: more complex but also more „democratic“ (actually closer to the way how CCD Sensors in digital cameras work)

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

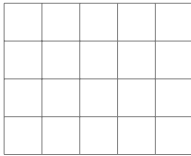
This is of course more tricky ...

- 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)?



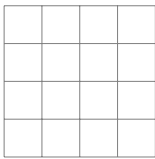
idea 1: just take the function value of the nearest midpoint („nearest neighbour interpolation“)

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



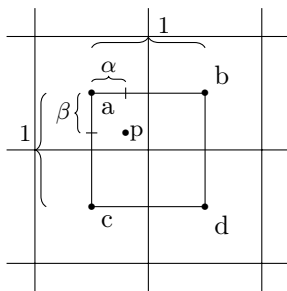
$\Rightarrow u(x) = u(x_i)$ for all $x \in B_i$
 \Rightarrow each box is uni-color
 \Rightarrow the continuous image is essentially still discrete

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 1×1)

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

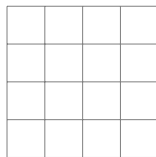
Then: The same 1D-interpolation between e and f
 $\Rightarrow u(x) := (1 - \beta) \cdot e + \beta \cdot f$

$$= (1 - \beta)[(1 - \alpha)a + \alpha b] + \beta[(1 - \alpha)c + \alpha d]$$

$$= \underbrace{(1 - \alpha)(1 - \beta)a + \alpha(1 - \beta)b}_{\in [0, 1] \wedge \sum = 1} + \underbrace{(1 - \alpha)\beta c + \alpha\beta d}_{\in [0, 1] \wedge \sum = 1}$$

\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)

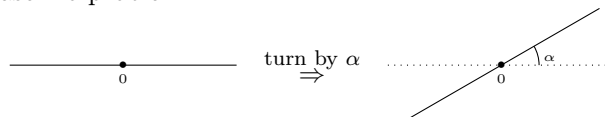
\Rightarrow 2D linear interpolation, *bi-linear interpolation* (can be interpreted as spline interpolation with bilinear **basis** splines).



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

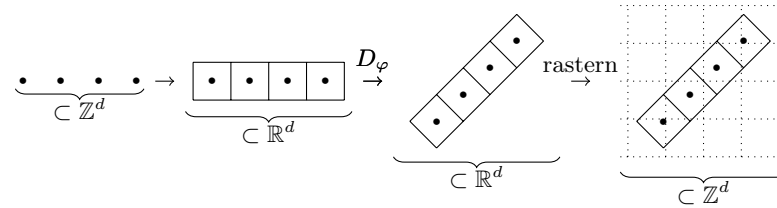
$$y = D_\varphi^{-1} x = D_{-\varphi} x$$

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

- discrete image case: problem !

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

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?)

¹it'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 .

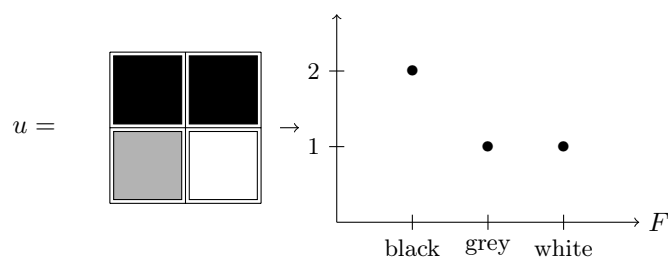
$H_u(k)$ counts how often colour k appears in u .

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

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)¹.

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

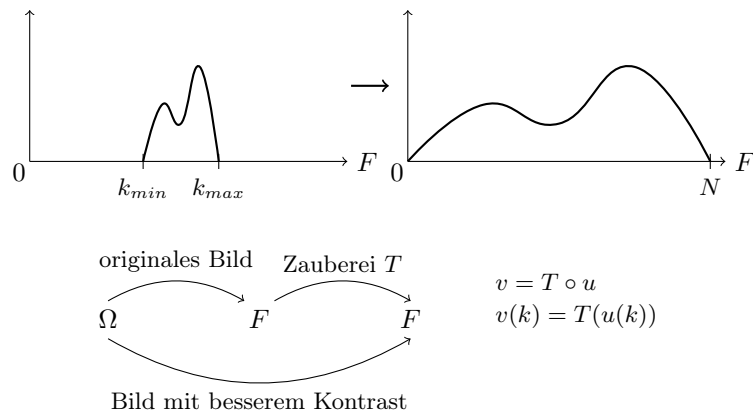
Matlab-Code

¹density 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:



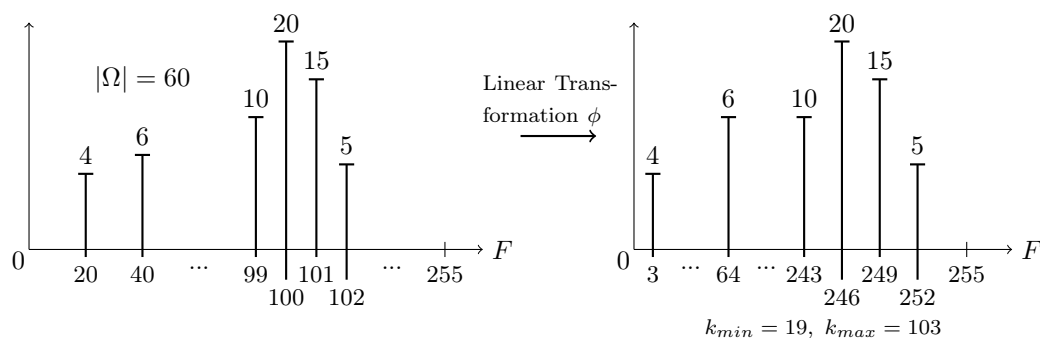
5

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

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

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

Example histogram:

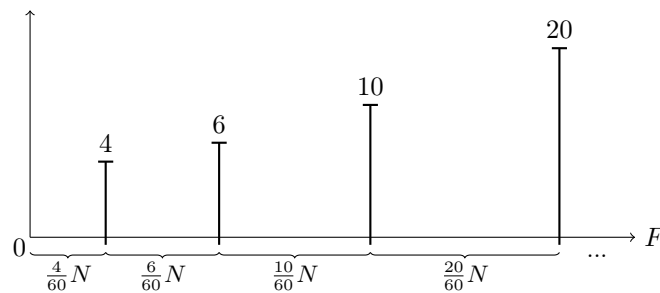


A bit more sophisticated:

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

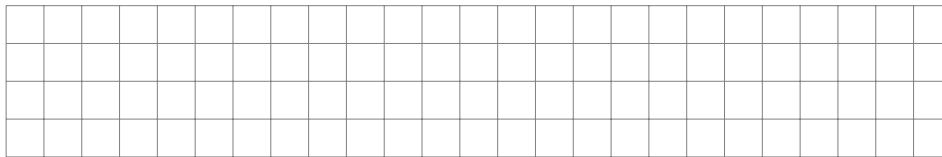
such that colour ranges that occur more frequently in u can occupy a larger range of colours in v .
 (\Rightarrow visibility \uparrow)

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“

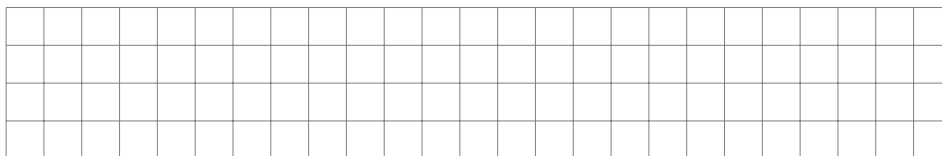
Exercise ?!

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

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

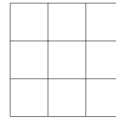
How to find the threshold t :

1.) Shape based methods

If the histogram is „biomodal“

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

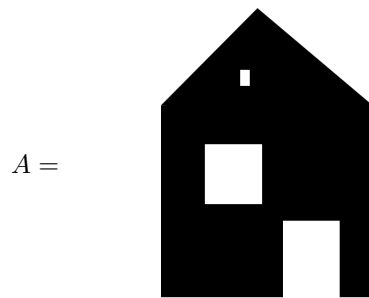
or $t := k_{\min}$



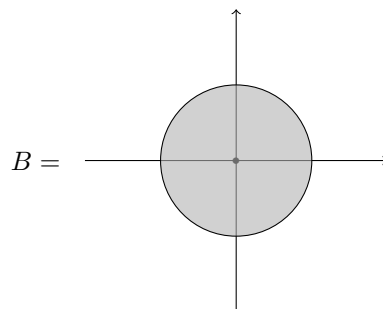
Kapitel 4

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.

$$A + B =$$

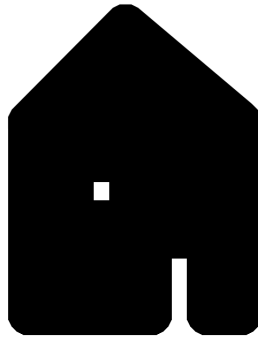


Image created in Matlab through:

```

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.

$$A - B =$$

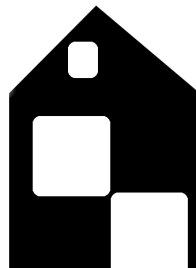


Image created in Matlab thorough:

```

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);

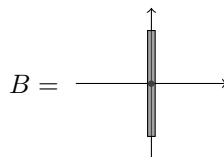
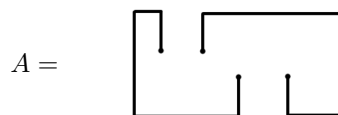
```

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. Entrauschen: Filter und Co

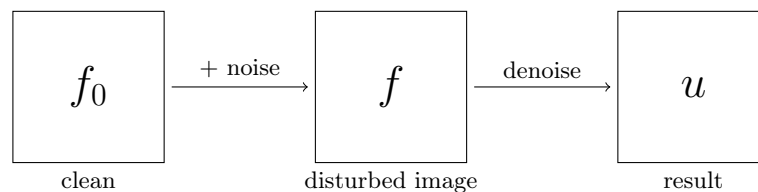
5.1 Noise

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:

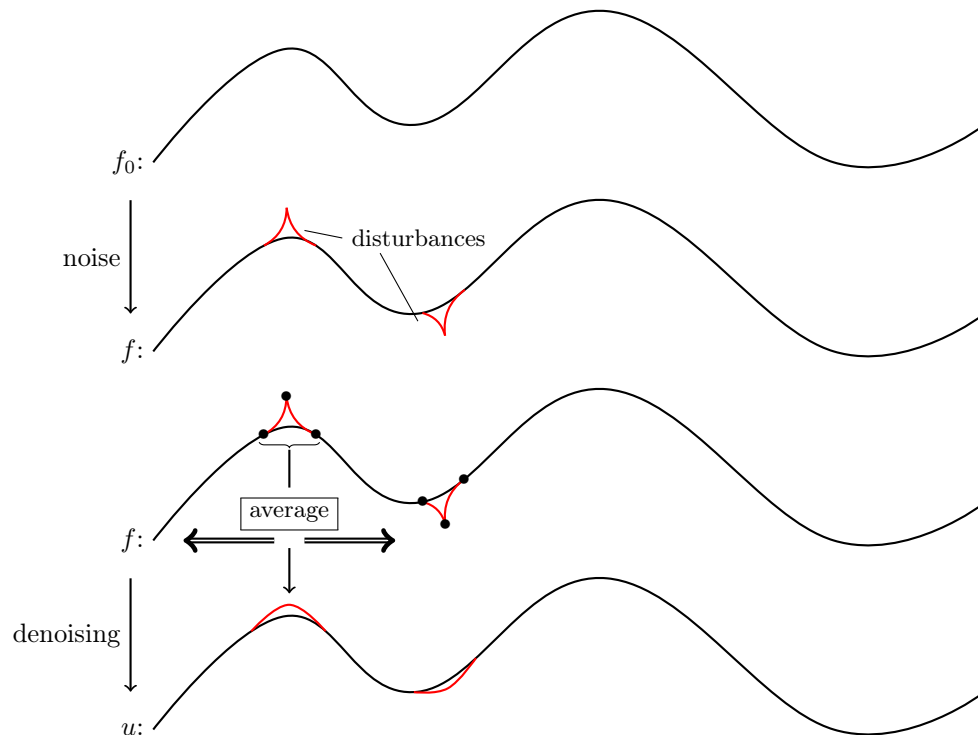
$$\|f\|_2 = \sqrt{\sum_{x \in \Omega} |f(x)|^2}$$

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)

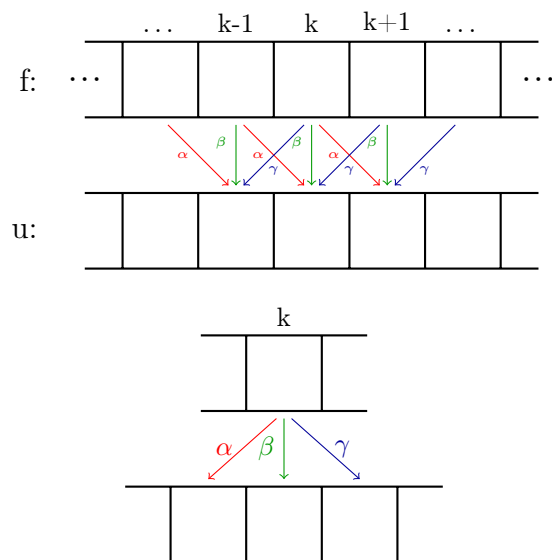


$$u(k) := \alpha \cdot f(k-1) + \beta \cdot f(k) + \gamma \cdot f(k+1) \quad (5.1)$$

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:

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

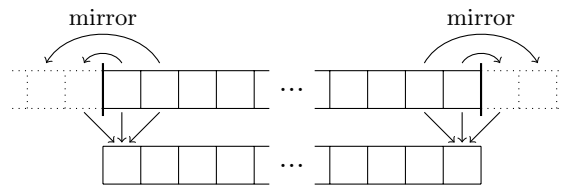
and:

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

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. $\tilde{f} \boxtimes \tilde{g} = \tilde{\tilde{f}} * \tilde{\tilde{g}} \stackrel{\boxed{3}}{=} \widetilde{(\tilde{f} * \tilde{g})} = \widetilde{f \boxtimes g}$, Compatibility with reflection

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

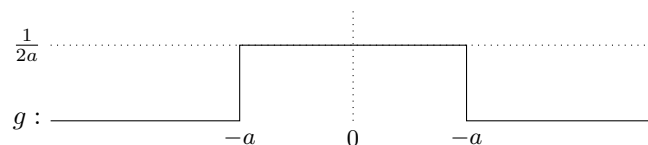
Man kann zeigen (Übung): $f, g \in \ell^1 \Rightarrow f * g \in \ell^1$ und $\|f * g\|_1 \leq \|f\|_1 \cdot \|g\|_1$. Wobei oft die Gleichheit gilt.

Alles gilt auch in der Kontinuierlichen Version:

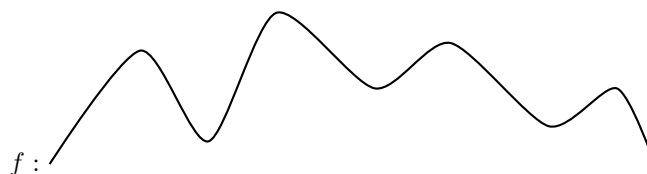
$$L^1(\mathbb{R}^d) := \left\{ f : \mathbb{R}^d \rightarrow \mathbb{R} : \underbrace{\int_{\mathbb{R}^d} |f| dx}_{:= \|f\|_1} < \infty \right\}$$

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

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 ℓ^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 ℓ^1 bzw. L^1 eine Algebra mit neutralem Element δ ?

ℓ^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} .

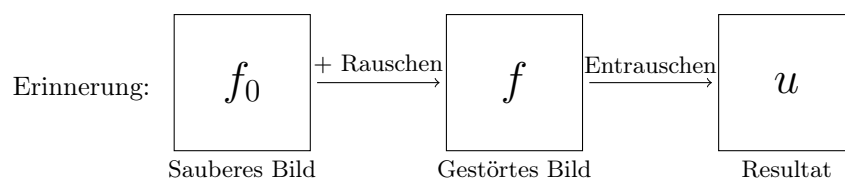
$$\Rightarrow u_{ij} = \alpha f_{i-1,j} + \beta f_{i,j-1} + \gamma f_{ij} + \delta f_{i,j+1} + \epsilon f_{i+1,j}$$

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

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

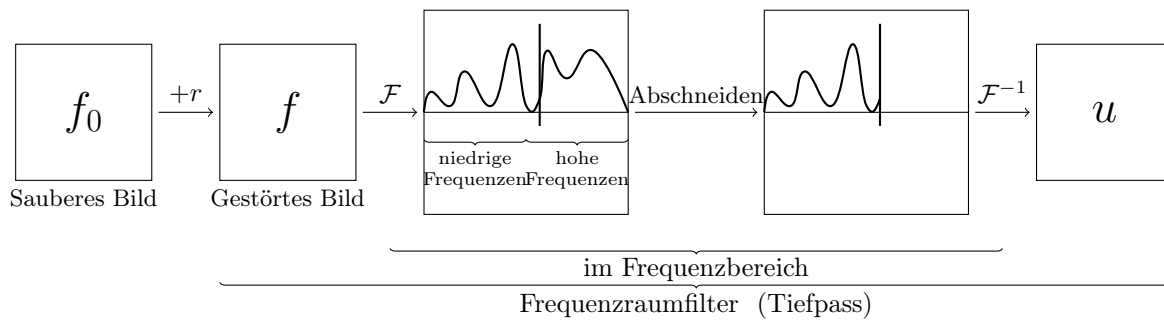
\Rightarrow gezieltes entfernen

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.

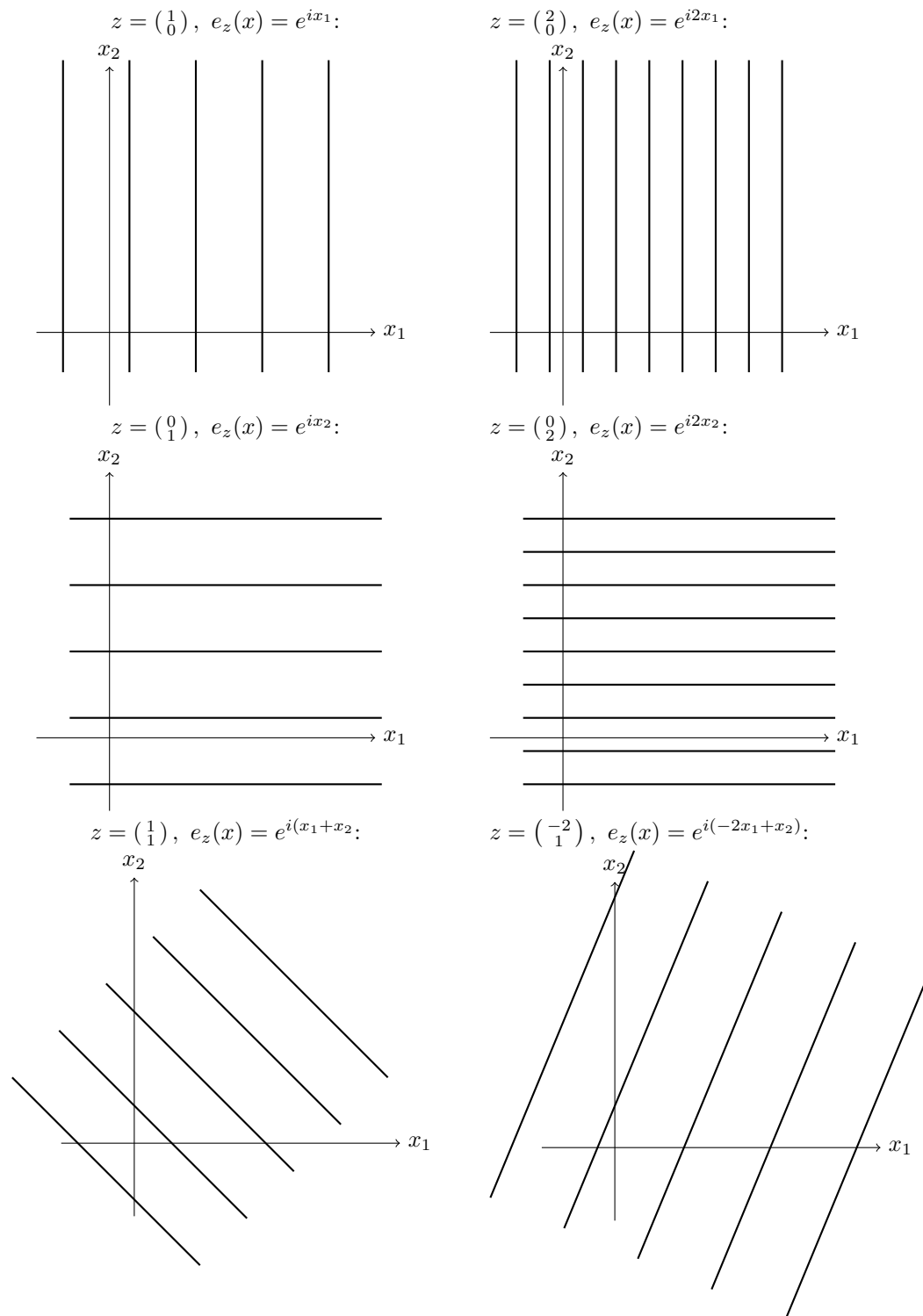
$$f(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \left(\frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} f(y) e^{-i\langle z, y \rangle} dy \right) e^{i\langle z, x \rangle} dz$$

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 $\Rightarrow \hat{f}(z) = \hat{f}(-z)$

-

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

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

- 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:

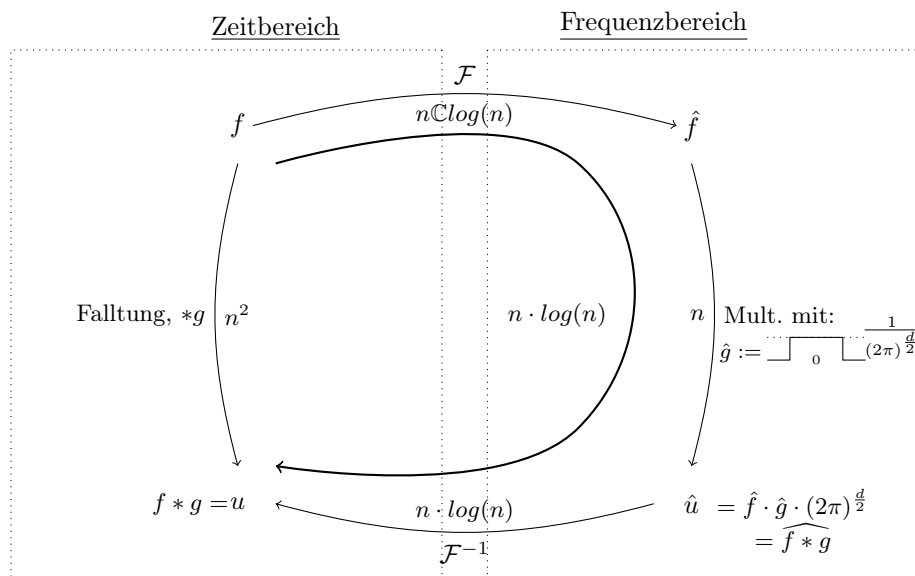
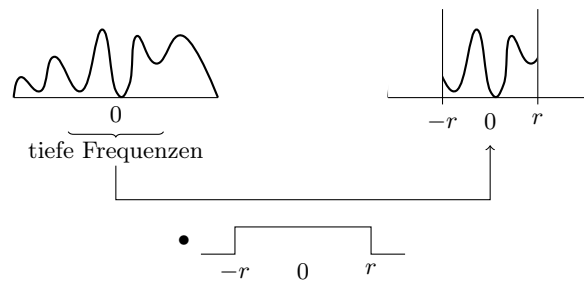
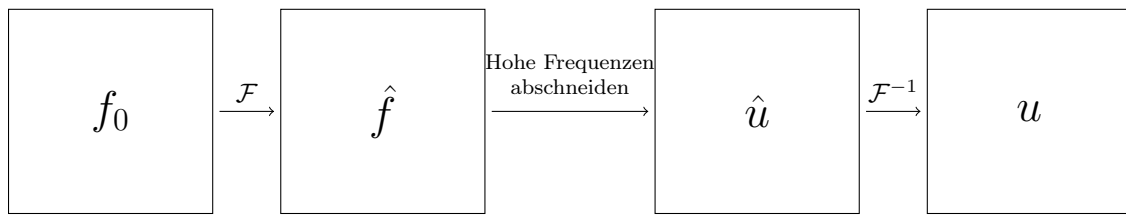
$$g(x) = x_1^{\alpha_1} \dots x_d^{\alpha_d} f(x) \Rightarrow \hat{g}(z) = i^{\alpha_1 + \dots + \alpha_d} \frac{\partial^{\alpha_1 + \dots + \alpha_d}}{\partial x_1^{\alpha_1}} \hat{f}(z)$$

-

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

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

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 :

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

$$= \left(\frac{1}{(2\pi)^{\frac{1}{2}}} e^{-\frac{x_1^2}{2}}\right) \cdot \dots \cdot \left(\frac{1}{(2\pi)^{\frac{1}{2}}} e^{-\frac{x_d^2}{2}}\right) = G(x_1) \cdot \dots \cdot G(x_d)$$

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

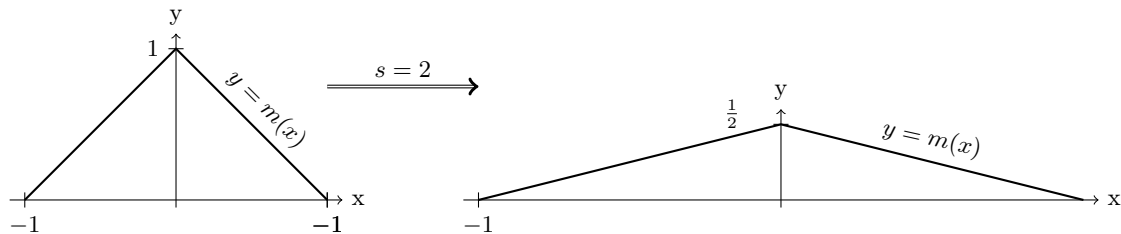
'glättet mehr als': $\frac{1}{9}$

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 σ

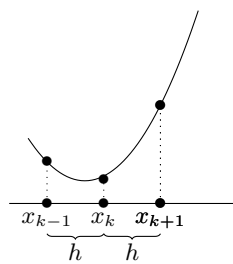
hier noch mehr im Skript p. 45

5.5 Differenzenfilter

Bisher: Glättung $\hat{=}$ Mittelwert bilden $\hat{=}$ Summe/Integrale

Jetzt: Schärfen $\hat{=}$ Differenzen/Kontraste hervorheben $\hat{=}$ Differenzen/Ableitungen

Diskretisierung von Ableitungen durch Differenzenquotienten



(hier bedeutet $f(k) = f(x_k)$)

Vorwärts: $u(h) = \frac{f(k+1) - f(k)}{h} \quad u = \frac{1}{h} \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \circledast f$

Rückwärts: $u(h) = \frac{f(k) - f(k-1)}{h} \quad u = \frac{1}{h} \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \circledast f$

Zentral: $u(h) = \frac{f(k+1) - f(k-1)}{2h} \quad u = \frac{1}{2h} \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \circledast f$

2. Ableitung:

$$\begin{aligned}
 u(h) &\approx \frac{f'(k+1) - f'(k)}{h} \text{ (vorwärts)} \\
 &\approx \frac{\frac{f(k+1) - f(k)}{h} - \frac{f(k) - f(k-1)}{h}}{h} \text{ (rückwärts)} \\
 &= \frac{f(k+1) - 2f(k) + f(k-1)}{h^2}
 \end{aligned}$$

Also folgt $u := \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \circledast f$ und $\frac{1}{h^2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} = \frac{1}{h} \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} * \frac{1}{h} \begin{bmatrix} -1 & 1 & 0 \end{bmatrix}$

Denn:

$$\begin{aligned}
 &\frac{1}{h} \begin{bmatrix} -1 & 1 & 0 \end{bmatrix} \\
 &= \frac{1}{h} \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} * \left(\frac{1}{h} \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} * f \right) \\
 &= \left(\frac{1}{h} \begin{bmatrix} 0 & 1 & -1 \end{bmatrix} * \frac{1}{h} \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} \right) * f \\
 &= \left(\frac{1}{h} \begin{bmatrix} -1 & 1 & 0 \end{bmatrix} \circledast \frac{1}{h} \begin{bmatrix} 1 & -1 & 0 \end{bmatrix} \right) * f \\
 &= \frac{1}{h^2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} * f \\
 &= \frac{1}{h^2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} \circledast f
 \end{aligned}$$

In 2D: $\frac{\partial}{\partial x} \hat{=} \begin{bmatrix} 0 & -1 & 1 \end{bmatrix}$, $\frac{\partial}{\partial y} \hat{=} \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$, $\frac{\partial^2}{\partial x^2} \hat{=} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$, $\frac{\partial^2}{\partial y^2} \hat{=} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$.

Diskreter Laplace Operator :

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \hat{=} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

5.6 Glättungsfiler und partielle Differentialgleichungen

Wir haben gesehen: $m = \frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ ist unter allen 5-Punkt Filtern der am besten glättende.

Idee: Rauschen weiter verringern indem man $m \circledast$ wiederholt anwendet \Rightarrow Folge von Bildern:

$$\boxed{\begin{array}{c} f \\ := u^{(0)} \end{array}} \xrightarrow{m \boxtimes} \boxed{u^{(1)}} \xrightarrow{m \boxtimes} \boxed{u^{(2)}} \dots$$

$\Rightarrow u^{(n+1)} - u^{(n)} = (\text{Unterschied zwischen 'Zeit' Punkt } n \text{ und } n+1)$

$$\begin{aligned} &= \underbrace{m \boxtimes u^{(n)}}_{u^{n+1}} - \underbrace{\delta \boxtimes u^{(n)}}_{u^{(n)}} \text{ mit } \delta = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \\ &= (m - \delta) \boxtimes u^{(n)} \\ &= \left(\frac{1}{5} \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} - \frac{1}{5} \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 5 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \right) \boxtimes u^{(n)} \\ &= \frac{1}{5} \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} u^{(n)} \end{aligned}$$

noch einmal schauen was 5.10 ist (5.10)

Somit gilt insgesamt:

$$\underbrace{u^{(n+1)} - u^{(n)}}_{\cong \frac{\partial u}{\partial t}} = \frac{1}{5} \underbrace{\begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}}_{\cong \Delta u} \quad (5.11)$$

Kontinuierlich: Funktion u

$$u(x, t) \quad x \in \mathbb{R}^2, \quad t \text{ Zeit}$$

(5.11) ist eine Diskretisierung (1 Zeitschritt im Eulerverfahren) der partiellen Differentialgleichungen

$$\frac{\partial u}{\partial t} = \Delta u \quad (5.12)$$

Bekannt als Wärmegleichung oder Diffusionsgleichung.
Zum Zeitpunkt $t = 0$ möge die Anfangsbedingung

$$u(x, 0) = u^{(0)} = f(x) \quad (5.13)$$

gelten. Vorranschreiten der Zeit t repräsentiert Diffusion.

Für einen stationären Zustand, also keine Änderung $\frac{\partial u}{\partial t}$ dann muss auch $\Delta u = 0$ gelten.

Diese wird unter anderem von konstanten Funktionen oder linearen Funktionen $u(x_1, x_2) = ax_1 + bx_2$ erfüllt.

Es existiert auch eine explizite Formel für die Lösung der Diffusionsgleichung (5.12) mit Anfangsbedingung (5.13):

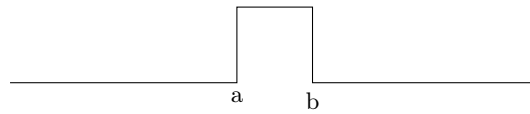
$$u(x, t) = \left(G_{\sqrt{2t}} * u^{(0)} \right) (x)$$

Wobei $\sqrt{2t}$ für eine Skalierung um diesen Wert steht.

Zu zeigen ist: $\frac{\partial u}{\partial t} = \Delta u$

$$\begin{aligned} &\frac{\partial}{\partial t} \left(G_{\sqrt{2t}} * u^{(0)} \right) = \Delta \left(G_{\sqrt{2t}} * u^{(0)} \right) \\ &\stackrel{\text{mit Satz}}{\implies} \left(\frac{\partial}{\partial t} G_{\sqrt{2t}} \right) * u^{(0)} = (\Delta G_{\sqrt{2t}}) * u^{(0)} \end{aligned}$$

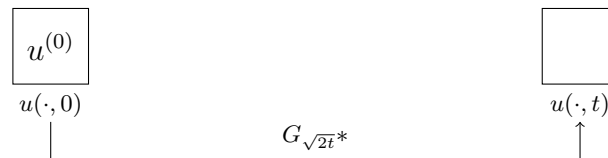
Es bleibt somit z.z.: $\frac{\partial}{\partial t} G_{\sqrt{2t}} = \Delta G_{\sqrt{2t}}$.

$t = 0:$  $t > 0:$ 

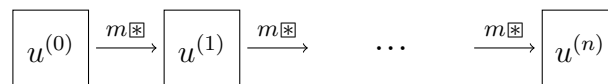
Bemerkenswert ist das, für $t = 0$ die Funktion nicht stetig ist, aber für alle $t > 0$ die Funktion beliebig oft differenzierbar ist.

Insgesamt lässt sich die Idee darstellen als:

kontinuierlich: $\xrightarrow{\hspace{10cm}}$ t



diskret:



Ab hier Livetex 24.11

Wiederholung Diffusionsgleichung letzte Woche:

Vergleich kontinuierlicher mit dem diskreten Fall.

5.7 Isotrope und anisotrope Diffusion

Haben gesehen: Glättung/Diffusion verringert rauschen

Aber: Auch Kanten/Details werden verwischt.

Ausweg: Diffusion steuern, so dass sie an Kanten weniger stark glättet.

an Kanten Stellen mit großer Änderungsrate in x - oder y -Richtung, oder beides, d.h.:

$$|\frac{\partial u}{\partial x}|^2 + |\frac{\partial u}{\partial y}|^2 = \left\| \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix} \right\|^2 [\nabla u]$$

$$\text{Plan: } \nabla u \begin{cases} \text{groß} & \Rightarrow \text{Diffusion} \searrow \\ \text{klein} & \Rightarrow \text{Diffusion normal} \end{cases} \quad (5.14)$$

Diffusionsgleichung:

$$\frac{\partial u}{\partial t} = \Delta u = \frac{\partial u}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial u}{\partial y} = \dots = \text{div}(M)(\nabla u)$$

Ansatz für M :

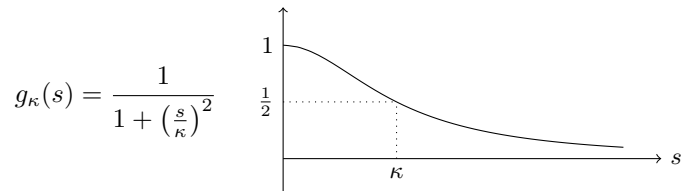
a) $M = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \Rightarrow$ übliche Diffusion

b) $M = g(|\nabla u(x, y)|) \cdot I$

$$g(s) = \frac{1}{\left(\frac{s}{\kappa}\right)^2 + 1} \text{ mit Parameter } \kappa > 0$$

\Rightarrow Perona & Malik (1990)

c) $M = \begin{pmatrix} g(|\frac{\partial u}{\partial x}|) & 0 \\ 0 & g(|\frac{\partial u}{\partial y}|) \end{pmatrix}$

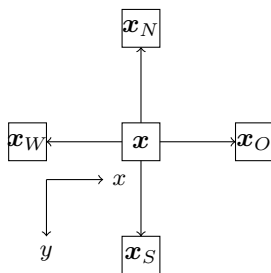


- Kante mit $\|\nabla u\| < \kappa$ werden geglättet ($g > \frac{1}{2}$)
- Kante mit $\|\nabla u\| \geq \kappa$ werden nicht geglättet ($g \leq \frac{1}{2}$)

Bild zu isotrop und anisotrop. (Kann man sich sparen?)

Im diskreten Fall: $\mathbf{x} \in \mathbb{Z}^2$ $\mathbf{x}_W = \mathbf{x} + \begin{pmatrix} -1 \\ 0 \end{pmatrix}$, usw.

Sei $M = \begin{pmatrix} c_1(\mathbf{x}) & 0 \\ 0 & c_2(\mathbf{x}) \end{pmatrix}$



$$\begin{aligned} \text{div}(M \cdot \nabla u(\mathbf{x})) &= \left(\frac{\partial}{\partial x} \frac{\partial}{\partial x} \right) \left[\begin{pmatrix} c_1(\mathbf{x}) & 0 \\ 0 & c_2(\mathbf{x}) \end{pmatrix} \begin{pmatrix} \frac{\partial u}{\partial x}(\mathbf{x}) \\ \frac{\partial u}{\partial y}(\mathbf{x}) \end{pmatrix} \right] \\ &= \left(\frac{\partial}{\partial x} \frac{\partial}{\partial x} \right) \begin{pmatrix} c_1(\mathbf{x}) \cdot \frac{\partial u}{\partial x}(\mathbf{x}) \\ c_2(\mathbf{x}) \cdot \frac{\partial u}{\partial y}(\mathbf{x}) \end{pmatrix} \\ &\approx \left(\frac{\partial}{\partial x} \frac{\partial}{\partial x} \right) \begin{pmatrix} c_1(\mathbf{x}) \cdot \dots \\ c_2(\mathbf{x}) \cdot \dots \end{pmatrix} \end{aligned}$$

Ab hier livetex

Wiederholung von letzter Woche.

Bei Salt, Pepper einmal vorglätten, aber nur bei der u in der Steuerungsfunktion.

5.8 Bilaterale Filter

Anderer Ansatz für selbes Problem:

$$u(\mathbf{x}) := \text{gewichtetes Mittel aus allen } f(\mathbf{y}) \text{ mit}$$

a) \mathbf{y} ist nahe bei \mathbf{x} und

b) $f(\mathbf{y})$ ist nahe bei $f(\mathbf{x})$

$$u(\mathbf{x}) = \frac{1}{w(\mathbf{x})} \int_{\Omega} g(\|\mathbf{x} - \mathbf{y}\|) f(\mathbf{y}) d\mathbf{y} \quad \text{unterpunkte}$$

Oft: $g, h \dots$ Gauß-Kerne (\Rightarrow „nichtlinearer Gauß-Filter“)

Manchmal: $g, h \dots$ charakteristische Funktionen bild . (\Rightarrow „SUSAN-Filter“) oder Mischungen

Effekt: Falls Höhe (Kante) > Filterradius (h) \Rightarrow Kante bleibt erhalten.

Numerische sehr aufwändig:

- keine reine Faltung (\Rightarrow keine FFT-Implementierung möglich)
- Normierung $w(\mathbf{x})$ in jedem Punkt neu berechnen.

Manchmal: $f \xrightarrow{\log} \log f \xrightarrow{\text{bil Filter}} \log u \xrightarrow{\exp} u$

5.9 Entrauschen mittels Variationsrechnung

Erinnerung: **Bild**

Wunsch 1: $u \approx f$ (Datenkonsistenz)

Wunsch 2: u sei „glatt“ (Regularitätsbedingung)

Mathematische Umsetzung der Wünsche:

Wunsch 1: $\|u - f\|_2 = \sqrt{\int_{\Omega} |u(x) - f(x)|^2 dx}$ sei klein.

Wunsch 2: $\|\nabla u\|_2 = \sqrt{\int_{\Omega} |\nabla u(x)|^2 dx} = \sqrt{\int_{\Omega} (\frac{\partial u}{\partial x}(x)^2) + (\frac{\partial u}{\partial y}(x)^2) dx}$ sei klein

Kombination:

$$\|u - f\|_2 + \lambda \cdot \|\nabla\|_2^2 \xrightarrow{u \in U} \min \quad (5.15)$$

U ... geeigneter Funktionenraum

$\lambda > 0$, fest („Kopplungskonstante“)

In diesem Bsp. empfiehlt sich als Suchraum

$$U = \{u : \|u\|_2 < \infty, \nabla u \text{ existiert}, \|\nabla u\|_2 < \infty\} =: W^{1,2}[]$$

1 ... Ableitung 1. Ordnung

2 ... 2-Norm

Im obigen Ansatz (5.15) stellt man fest, dass der Regularitätsterm

$$\|\nabla u\|_2^2 = \int_{\Omega} |\nabla u(x)|^2 dx = \int_{\Omega} (\frac{\partial u}{\partial x}(x)^2) + (\frac{\partial u}{\partial y}(x)^2) dx$$

die große Gradienten an (gewollten Kanten) zu stark bestraft. (\Rightarrow optimales u hat geglättete Kanten)

Ausweg: Wähle $\|\nabla u\|_2 = \sqrt{s.o.}$ oder $\|\nabla u\|_1 = \int |\nabla u(x)| dx = \int_{\Omega} (|\frac{\partial u}{\partial x}(x)| + |\frac{\partial u}{\partial y}(x)|) dx$

$$\|u - f\|_2 + \lambda \cdot \|\nabla\|_1 \xrightarrow{u \in U} \min \quad (5.16)$$

Genannt Rudin-Osher-Fatemi-Funktional (ROF)

Allgemeiner Ansatz bei Variationsproblemen:

$$J(u) := \underbrace{D(u, f)}_{\text{Datenkern}} + \lambda \underbrace{R(u)}_{\text{Regularitätsterm}} \xrightarrow{u \in U} \min$$

Notwendiges Kriterium:

Falls $J : U \rightarrow \mathbb{R}$ in $u \in U$ ein lokales Minimum besitzt, dann gilt für jede Richtung $v \in U$:

$$\lim_{\epsilon \searrow 0} \frac{J(u + \epsilon v) - J(u)}{\epsilon} = 0 \quad (5.17)$$

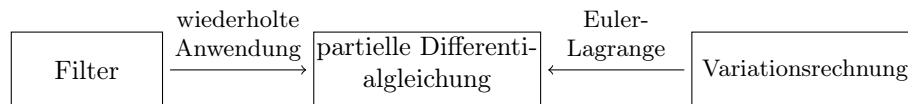
Dies ist die Verallgemeinerte Richtungsableitung (Gateaux-Ableitung).

Häufig ist J in Integralform gegeben, z.b.:

$$J(u) = \int_{\Omega} g(x, u(x), \nabla(x)) dx$$

Dann führt Bedingung (5.17) auf Gleichungen für bestimmte partielle Ableitungen von g und u , die sogenannte Euler-Lagrange-Gleichung für (5.17).

\Rightarrow partielle Differentialgleichung u . Fazit:



Kapitel 6

6. Kantenerkennung

6.1 Gradientenfilter

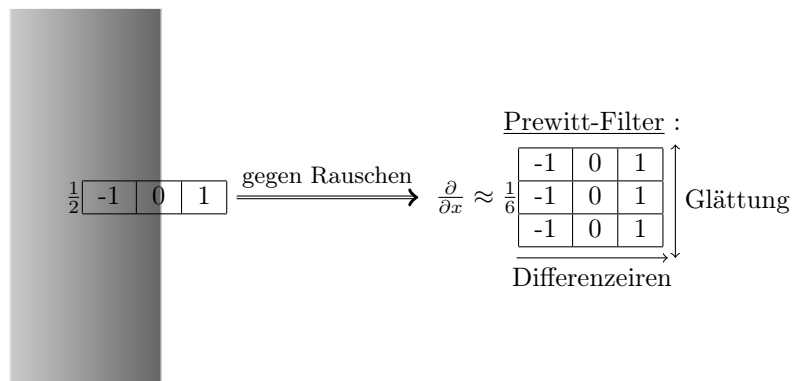
Wir suchen Stellen \mathbf{x} mit großem Gradienten:

$$\nabla u(\mathbf{x}) = \begin{pmatrix} \frac{\partial u}{\partial x}(\mathbf{x}) \\ \frac{\partial u}{\partial y}(\mathbf{x}) \end{pmatrix}$$

Approximation der Gradienten über zentrale Differenzen:

$$\frac{\partial}{\partial x} \approx \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \text{ bzw. } \frac{\partial}{\partial y} \approx \frac{1}{2} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \quad (6.1)$$

Um Rauschen zu verringern wird auch ein entrauschen Filter simultan angewendet:



Alternative: $\frac{\partial}{\partial x} \approx \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \boxtimes \frac{1}{4} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} =: D_x$, genannt Sobel-Filter. Eine stärkere Glättung kann mittels anderer vertikaler Filter mit Binomialkoeffizienten erzielt werden.

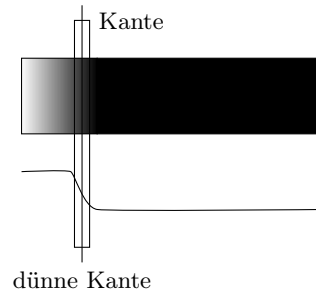
Entsprechen wird $\frac{\partial}{\partial y} D_y := D_x^T$ definiert.

$$\nabla u(\mathbf{x}) = \begin{pmatrix} \frac{\partial u}{\partial x}(\mathbf{x}) \\ \frac{\partial u}{\partial y}(\mathbf{x}) \end{pmatrix} \approx \begin{pmatrix} (D_x \boxtimes u)(\mathbf{x}) \\ (D_y \boxtimes u)(\mathbf{x}) \end{pmatrix} \quad (6.2)$$

Zur Erinnerung der Gradienten steht senkrecht auf Kanten und zeigt in Richtung heller (hoher) Werte, die Intensität wird beschrieben von $|\nabla u(\mathbf{x})|$, also dem Betrag des Gradienten.

Ein typischer Algorithmus kann etwa folgende Form annehmen:

1. Gradienten mittels Prewitt oder Sobel approximieren und Richtung auf Vielfache von 45° runden.
2. Non-maximum suppression (edge thinning). Da es potentiell viele Punkte mit hoher Steigung gibt kann es dazu kommen, dass Kanten sehr breit werden, dieses wird durch das edge thinning verhindert.



Mathematisch: \mathbf{x} wird Kantenpunkt falls:

$$|\nabla u(\mathbf{x})| \leq \max(|\nabla u(\mathbf{x}_+)|, |\nabla u(\mathbf{x}_-)|)$$

wobei \mathbf{x}_+ und \mathbf{x}_- Vorgänger und Nachfolger von \mathbf{x} in Gradientenrichtung sind.

3. Kandidat \mathbf{x} wird Kantenpunkt, falls:



wobei t_1 , t_2 thresholds sind.

\mathbf{x} ist also ein Kantenpunkt, falls $|\nabla u(\mathbf{x})| \geq t_2$ oder $(|\nabla u(\mathbf{x})| \in [t_1, t_2])$ und \mathbf{x} ist Nachbar eines Kantenpunktes).

Dieses wird hysteresis thresholding genannt und verhindert Abreißen von Kantenzügen.

Die am häufigsten verbreitete Version von 1) -3) ist der Canny-Algorithmus (1986).

Matlab:

```
1 BWimg=edge(u,'canny',[t_1, t_2],sigma);
```

BWimg: Binärbild

u: Graustufenbild

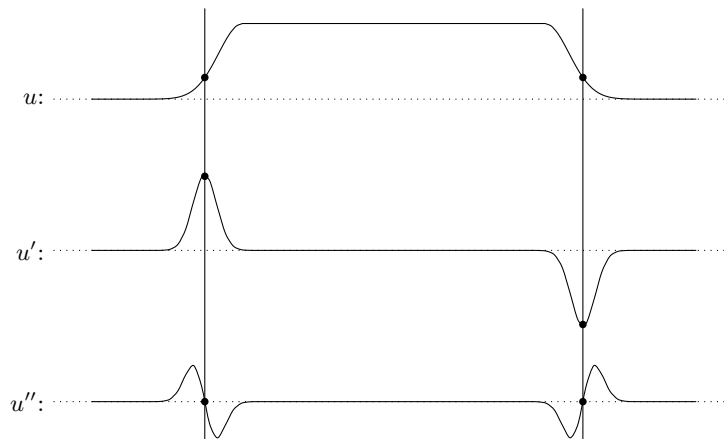
canny: Algorithmus

t_1 , t_2 : Sind gewählt wie oben

sigma: Parameter für den Gaußkern aus 1)

6.2 Die zweite Ableitung

Zunächst in 1D:



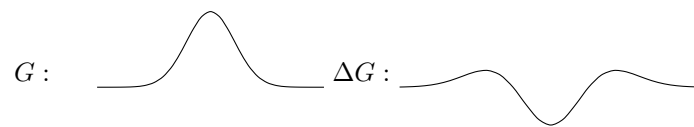
Test für kantenpunkte $u''(\mathbf{x}) = 0$ und $|u'(\mathbf{x})| > \text{threshold}$.

Wichtig: Vorglätten!, da die 2. Ableitung noch anfälliger gegenüber Rauschen als die 1. Ableitung ist.

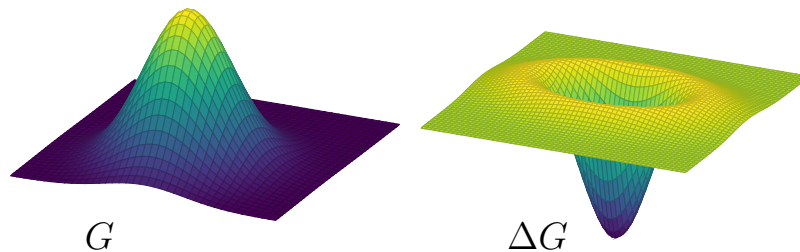
In 2D. Laplace Operator $\Delta u = \frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y}$ (Richtungsunabhängige Messung der 2. Ableitung)

Vorglätten: $\Delta(G * u) = (\Delta G) * u$, wobei ΔG vorher berechnet werden kann.

In 1D:



In 2D:



Dieses wird Laplacian of Gaussian method genannt.

Matlab:

```
1 BWing=edge(u,'log',thresh,sigma);
```

\Rightarrow alle $\mathbf{x} \in \Omega$ mit:

$$\Delta(G_{\text{sigma}} * u)(\mathbf{x}) \approx u, \text{ nicht auf Gleichheit sondern auf Vorzeichenwechsel testen.}$$

und: $|\nabla(G_{\text{sigma}} * u)| > \text{thresh}$

Kapitel 7

7.Schaerfen und Entfalten

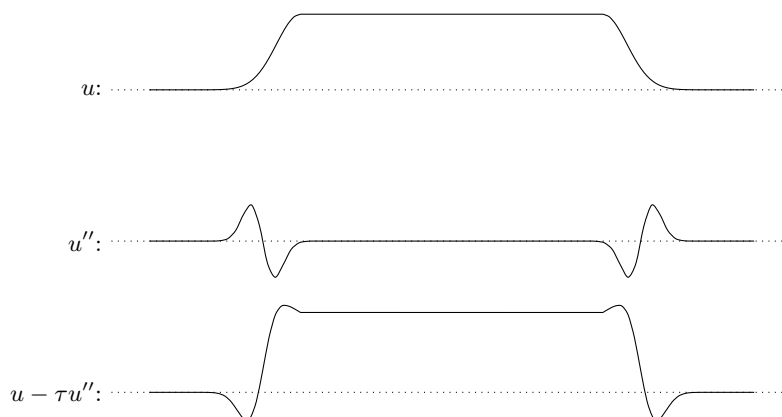
(Gegenteil von Kapitel 5)

Gegeben: unscharfes Bild

Gesucht: Version mit vielen erkennbaren Details

7.0.1 Laplace-Schärfen

Idee:



Zu sehen ist, dass durch die Subtraktion von u'' , skaliert mit einem Faktor $\tau > 0$ die Kanten hervorgehoben werden.

Hinweise zur Umsetzung:

label= $u - \tau u''$ reskalieren (Kontrast-stretching) falls der Farbraum verlassen wird.

lbbel= τ kann auch sehr klein gewählt werden und der Vorgang dafür wiederholt iteriert werden.

lcbel= In 2D Δ statt 2. Ableitung

ldbel= Vorglätten: $u - \tau \mathbb{C} \Delta (G * u)$

7.0.2 Kantenverstärkende Diffusion

Verallgemeinerte Diffusionsgleichung: $\frac{\partial u}{\partial t} = \text{div}(M \nabla u)$. Idee: M so wählen, so dass der Fluss:

label= Parallel zum Gradienten (d.h. durch die Kante verläuft): $\lambda_1 = \frac{1}{1 + \frac{|\nabla u(\mathbf{x})|^2}{\kappa^2}}$

lbbel= senkrecht zu ∇u (entlang der Kante): $\lambda_2 = 1$

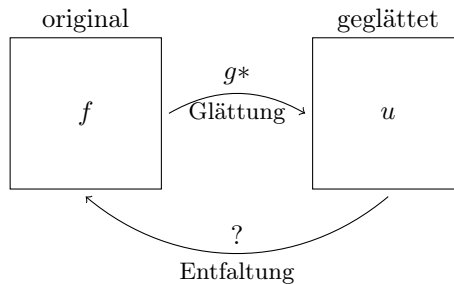
$\Rightarrow M$ hat EW λ_1 zum EV $v_1 = \frac{\nabla u}{|\nabla u|} \begin{pmatrix} \frac{\partial u}{\partial x}(\mathbf{x}) \\ \frac{\partial u}{\partial y}(\mathbf{x}) \end{pmatrix}$ und EW λ_2 zum EV $v_2 = \frac{1}{|\nabla u|} \begin{pmatrix} -\frac{\partial u}{\partial y}(\mathbf{x}) \\ \frac{\partial u}{\partial x}(\mathbf{x}) \end{pmatrix} \perp v_1$.

$\Rightarrow M \mathbb{C} \underbrace{\begin{pmatrix} v_1 & v_2 \end{pmatrix}}_{\text{orthogonale Matrix}} = \begin{pmatrix} \lambda_1 v_1 & \lambda_2 v_2 \end{pmatrix} = \begin{pmatrix} v_1 & v_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} \Rightarrow M^{-1} = M^T$

$\Rightarrow M = \begin{pmatrix} v_1 & v_2 \end{pmatrix} \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} \underbrace{\begin{pmatrix} v_1 & v_2 \end{pmatrix}^T}_{= \begin{pmatrix} v_1^T \\ v_2^T \end{pmatrix}} = \frac{1}{|\nabla u|^2} \begin{pmatrix} \lambda_1 \frac{\partial u}{\partial x}(\mathbf{x})^2 + \lambda_2 \frac{\partial u}{\partial y}(\mathbf{x})^2 & (\lambda_1 - \lambda_2) \frac{\partial u}{\partial x} \frac{\partial u}{\partial y}(\mathbf{x}) \\ (\lambda_1 - \lambda_2) \frac{\partial u}{\partial x} \frac{\partial u}{\partial y}(\mathbf{x}) & \lambda_2 \frac{\partial u}{\partial x}(\mathbf{x})^2 + \lambda_1 \frac{\partial u}{\partial y}(\mathbf{x})^2 \end{pmatrix}$

falls $\nabla u(\mathbf{x}) \neq 0$, sonst $M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

7.0.3 Entfaltung



Das heißt: $u = f * g$, wobei u, g gegeben sind und f gesucht ist.

Alternativ kann dies als die Invertierung des Faltungsoperator $f \mapsto g * f$ betrachtet werden.

1. Diskreter Fall:

$$g * f = u$$

$$(g * f)(j) = u(j), \quad j \in \Omega$$

$$\sum_k g(j - k) f(k) = u(j), \quad j \in \Omega$$

$\Rightarrow \Omega \times \Omega$ Gleichungssystem

$$\begin{array}{c}
 k=0 \quad k=1 \qquad \qquad \qquad k=n \\
 \begin{array}{c} j=0 \\ j=1 \\ \vdots \\ j=n \end{array} \begin{pmatrix} g(0) & g(-1) & & g(-n) \\ g(1) & g(0) & g(-1) & \\ & \ddots & \ddots & \ddots \\ g(n) & & & \end{pmatrix} \begin{pmatrix} f(0) \\ f(1) \\ \vdots \\ f(n) \end{pmatrix} = \begin{pmatrix} u(0) \\ u(1) \\ \vdots \\ u(n) \end{pmatrix}
 \end{array}$$

Toeplitz-Matrix

2. Kontinuierlicher Fall:

$$(g * f)(x) = u(x), \quad x \in \Omega$$

$$\int g(x-y)f(y)dy = u(x), \quad x \in \Omega$$

Integralgleichung für die gesuchte Funktion f

\Rightarrow Kontinuierliche Matrix:

Wobei $[a, b]$ die das Definitionsgebiet von f ist. Diese Problem is jedoch schlecht gestellt, da der Operator kompakt ist. (\nearrow Datei im Studip)

Wir versuchen es trotzdem zu lösen:

$$\begin{array}{ll}
 g * f = u & | \cdot \mathcal{F} \\
 \mathcal{F}(g * f) = \mathcal{F}u & \\
 (2\pi)^{\frac{d}{2}}(\mathcal{F}g) \cdot (\mathcal{F}f) = \mathcal{F}u & | \div (2\pi)^{\frac{d}{2}}(\mathcal{F}g) \\
 \mathcal{F}f = \frac{\mathcal{F}u}{(2\pi)^{\frac{d}{2}}\mathcal{F}g} & | \mathcal{F}^{-1}
 \end{array}$$

Und erhalten:

$$f = (2\pi)^{\frac{d}{2}} \mathcal{F}^{-1} \left(\frac{\mathcal{F}u}{\mathcal{F}g} \right) \quad (7.1)$$

Dieses kann jedoch zu Problemen führen, da etwa $g \approx 0$ werden kann. Je glatter g ist, desto stärker klingt $(\mathcal{F}g)(z)$ ab für $z \rightarrow \infty$.

Anders betrachtet:

Wenn $|\hat{g}(z)|$ für hohe Frequenzen klein ist, dann ist:

$$A : f \mapsto g + f$$

ein Tiefpassfilter. Nimmt man nun eine Funktion h mit hoher Frequenz und großer Amplitude, dann gilt:

$$A(f + h) = Af + \underbrace{Ah}_{\approx 0} \approx Af$$

Problembehebung:

Approximiere die Funktion $\frac{1}{x}$ durch

$$R_\alpha = \begin{cases} \frac{1}{x}, & |x| > \alpha \\ \frac{1}{\alpha}, & x \in [0, \alpha] \\ \frac{1}{-\alpha}, & x \in [-\alpha, 0] \end{cases}$$

1. Ansatz:

wobei $\alpha > 0$.

und ersetze $f = (2\pi)^{\frac{d}{2}} \mathcal{F}^{-1} \left(\frac{\hat{u}(z)}{\hat{g}(z)} \right)$ durch:

$$f = (2\pi)^{\frac{d}{2}} \mathcal{F}^{-1} (\hat{u}(z) R_\alpha(\hat{g}(z)))$$

und lasse $\alpha \rightarrow 0$.

2. Ansatz: Variationsrechnung:

1. Wunsch: $g * f \approx u$

2. Wunsch $\|f\|_2$ klein

Minimiere nun:

$$\Rightarrow J(f) := \|g * f - u\|_2^2 + \lambda \|f\|_2^2 \rightarrow \min$$

$$\iff \int_{\mathbb{R}^d} ((g * f)(x) - u(x))^2 + \lambda f(x)^2 dx \rightarrow \min$$

über die Wahl von $f \in U := L^2(\mathbb{R}^d)$.

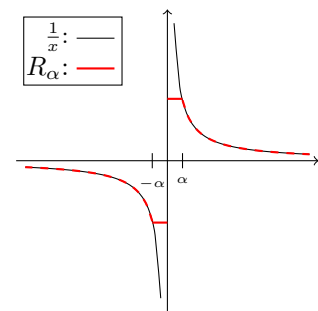
Idee: \mathcal{F} anwenden $\Rightarrow *$ wird zu \mathbb{C} und $\|C\|_2$ bleibt unverändert.

$$\begin{aligned} \Rightarrow J(f) &= \|g * f - u\|_2^2 + \lambda \|f\|_2^2 \\ &= \|g * \widehat{f} - u\|_2^2 + \lambda \|\hat{f}\|_2^2 \\ &= \|(2\pi)^{\frac{d}{2}} \hat{g} \hat{f} - \hat{u}\|_2^2 + \lambda \|\hat{f}\|_2^2 \\ &= \int_{\mathbb{R}^d} \left[\left((2\pi)^{\frac{d}{2}} \hat{g}(z) \hat{f}(z) - \hat{u}(z) \right)^2 + \lambda |\hat{f}(z)|^2 \right] dz \xrightarrow{f \in U} \min \end{aligned}$$

Strategie: Integral für jedes einzelne z minimieren. Daraus erhalten wir ein optimales \hat{f} und somit auch ein optimales f .

Also minimiere für jedes $z \in \mathbb{R}^d$

$$I(t) := |(2\pi)^{\frac{d}{2}} \hat{g}(z)t - \hat{u}(z)|^2 + \lambda |t|^2 \xrightarrow{t \in \mathbb{C}} \min$$



Später setzen wir $\hat{f}(z) := t_{min}$, nun zur minimierung:

$$\begin{aligned} I(t) &= ((2\pi)^{\frac{d}{2}} \hat{g}(z)t - \hat{u}(z))((2\pi)^{\frac{d}{2}} \overline{\hat{g}(z)} \bar{t} - \overline{\hat{u}(z)}) + \lambda t \bar{t} \\ &= (2\pi)^{\frac{d}{2}} \hat{g}(z) \overline{\hat{g}(z)} t \bar{t} + \lambda t \bar{t} - (2\pi)^{\frac{d}{2}} (\hat{g}(z) \overline{\hat{u}(z)} t + \overline{\hat{g}(z)} \hat{u}(z) \bar{t}) + \hat{u}(z) \overline{\hat{u}(z)} \\ &= ((2\pi)^{\frac{d}{2}} |\hat{g}(z)|^2 + \lambda) |t|^2 - (2\pi)^{\frac{d}{2}} \underbrace{2 \operatorname{Re}(\hat{g}(z) \overline{\hat{u}(z)} t)}_{\circledast} + |\hat{u}(z)|^2 \xrightarrow{t \in \mathbb{C}} \min \end{aligned}$$

Das Argument (Winkel) taucht nur in \circledast auf

\Rightarrow So wählen, das \circledast auf die positive reelle Achse fällt

$$\Rightarrow 0 = \arg(\circledast) = \arg(\hat{g}(z) \overline{\hat{u}(z)}) + \arg(t)$$

$$\Rightarrow \arg(t) = -\arg(\hat{g}(z) \overline{\hat{u}(z)}) = \arg(\overline{\hat{g}(z)} \hat{u}(z))$$

$$\Rightarrow I(t) = ((2\pi)^{\frac{d}{2}} |\hat{g}(z)|^2 + \lambda) |t|^2 - (2\pi)^{\frac{d}{2}} 2 \operatorname{Re}(\overline{\hat{g}(z)} \hat{u}(z) |t|) + |\hat{u}(z)|^2 \xrightarrow{|t| \in \mathbb{R}} \min$$

Dieses ist nun ein Polynom in $|t|$, sodass das minimum einfach bestimmt werden kann.

$$\begin{aligned} 0 &= \frac{d}{d|t|} \dots = 2\mathbb{C}((2\pi)^{\frac{d}{2}} |\hat{g}(z)|^2 + \lambda) |t| - (2\pi)^{\frac{d}{2}} 2 \operatorname{Re}(\overline{\hat{g}(z)} \hat{u}(z)) \\ \Rightarrow |t| &= \frac{(2\pi)^{\frac{d}{2}} 2 \operatorname{Re}(\overline{\hat{g}(z)} \hat{u}(z))}{2\mathbb{C}((2\pi)^{\frac{d}{2}} |\hat{g}(z)|^2 + \lambda)} = \frac{(2\pi)^{\frac{d}{2}} \mathbb{C}(\overline{\hat{g}(z)} \hat{u}(z))}{(2\pi)^{\frac{d}{2}} |\hat{g}(z)|^2 + \lambda} \text{ und } \arg(u) = \arg(\overline{\hat{g}(z)} \hat{u}(z)) \\ \Rightarrow t &= \frac{(2\pi)^{\frac{d}{2}} \overline{\hat{g}(z)} \hat{u}(z)}{(2\pi)^{\frac{d}{2}} |\hat{g}(z)|^2 + \lambda} =: \hat{f}(z) \end{aligned}$$

Wegen

$$\hat{f}(z) = (2\pi)^{\frac{d}{2}} \frac{\overline{\hat{g}(z)}}{(2\pi)^{\frac{d}{2}} |\hat{g}(z)|^2 + \lambda} \hat{u}(z)$$

gilt

$$f(z) = \mathcal{F}^{-1} \left(\frac{\overline{\hat{g}(z)}}{(2\pi)^{\frac{d}{2}} |\hat{g}(z)|^2 + \lambda} \right) * u \quad (7.2)$$

Dieses Verfahren wird L^2 deblurring genannt. Es gibt auch einen alternativen, algebraischen Zugang:

$$\begin{aligned} I(f) &= \|g * f - u\|_2^2 + \lambda \|f\|_2^2 \xrightarrow{f} \min \\ &\iff \left\| \begin{pmatrix} g * f - u \\ \sqrt{\lambda} f \end{pmatrix} \right\| \xrightarrow{f} \min \\ &\iff \left\| \begin{pmatrix} Af \\ \sqrt{\lambda} f \end{pmatrix} - \begin{pmatrix} u \\ 0 \end{pmatrix} \right\| = \left\| \begin{pmatrix} A \\ \sqrt{\lambda} \end{pmatrix} f - \begin{pmatrix} u \\ 0 \end{pmatrix} \right\| \xrightarrow{f} \min \quad (A = f \mapsto g * f) \end{aligned}$$

\Rightarrow lineares Ausgleichsproblem.

$$\begin{aligned} \Rightarrow (A^* \quad \sqrt{\lambda} I^*) \begin{pmatrix} A \\ \sqrt{\lambda} I \end{pmatrix} f &= (A^* \quad \sqrt{\lambda} I) \begin{pmatrix} u \\ 0 \end{pmatrix} \quad (\text{Normalengleichung}) \\ \Rightarrow (A^* A + |\lambda| I) f &= A^* u \\ \Rightarrow f &= (A^* A + |\lambda| I)^{-1} A^* u \end{aligned}$$

Die Inverse existiert, da $-|\lambda|$ nicht im Spektrum von $A^* A$ sein kann, denn das Spektrum von $A^* A$ ist positiv und reel.

3. Ansatz: noch einmal Variationsrechnung, diesmal mit anderen Wünschen

1. Wunsch: $g * f \approx u$

2. Wunsch: $\|\nabla f\|$ klein

Nach analoger Rechnung wie oben erhält man:

$$f = \mathcal{F} \left(\frac{\overline{\hat{g}(z)}}{(2\pi)^{\frac{d}{2}} |\hat{g}(z)|^2 + \lambda |z|^2} \right) * u \quad (7.3)$$

\Rightarrow Dämpfung höher wenn Frequenz höher.

Dieses Verfahren nennt sich H^1 deblurring