Liste der noch zu erledigenden Punkte

Kapitel 1

1.Overview

- „image society“ (webpages: 1995 text-based, 2005 image based, 2015 video based . . .)

  \- data transfer rates ↑, compression rates ↑
  critical shift: reading → watching

- „Photoshop“-ing (remove wrinkles, bumps, . . .)

- Images in medicine („medical image proscessing“), x-ray, CT, MRI, ultrasound, . . . („modalities“).
  different questions:

  1.) Layout!

     align bottom measurments $\overset{?}{\Rightarrow}$ image
     expl: tomography
     ⇒ difficult mathematical problems

  2.) Image enhancements
     – denoising
       simple pixels/lines: „sandpaper“ interpolation
       global noise: smoothing

     so richtig?

     – 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 ⇒ Wort? , compare Skizze
       ↗ object following in a movie

<u>Our Focus:</u>

\- mathematical models/methods/ideas

\- (algorthms)

\- ((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:

Discrete / Digital

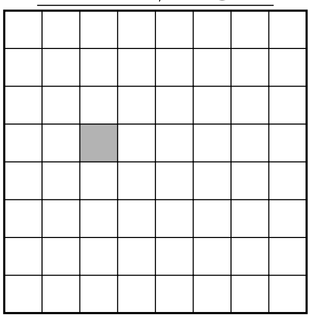


Abbildung 2.1: Discrete Image

Continous / Analogue

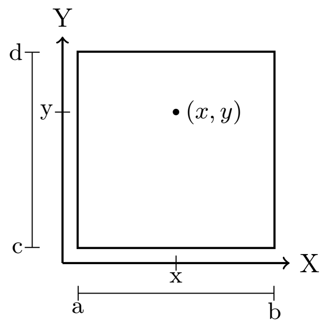


Abbildung 2.2: Continous Image

	Discrete / Digital	Continous / Analogue
object:	matrix	function
tools:	linear algebra (SVD, . . .)	analysis (differentrage, integrate, . . .)
pros:	(finite storage) storage, complexity	freedom, tools, motions?P.4 (e.g. edge discontinuity)
cons:	limitations: zooming, rotations, . . .	storage (infinite amout of data)

arguably, one has:

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

In general we will say:

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

$$u : \Omega \to F,$$

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

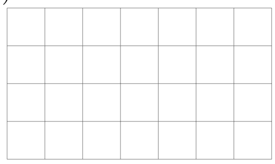
$d = 4$ (3D + time)

F . . . *range of colours*

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

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

$F = \{0, 1\}$ . . . 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 differentirate, . . . the image u.

Still: need to assume that also F ist continuous (not as $\{0, 1\}$, $\{0, 1, \dots, 255\}$ or $\mathbb{N}$) since otherwise the only differentiable (actually, the only continuous) functions $u : \Omega \to F$ 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

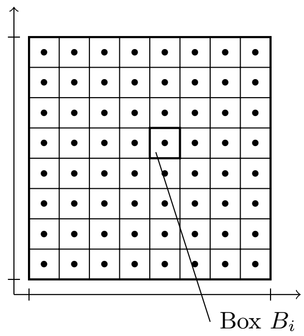
2.2 Switching between discrete and continuous images

continuous $\to$ 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 = midpoint of box B_i
  - strategy 2: use mean value

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

$\Omega =$



$\Rightarrow$ 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)

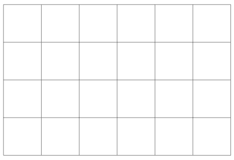
strategy 2: more complex but also more „democratic“ (actually closer to the way how CCD Sensors in digital cameras work)

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 $\to$ continous

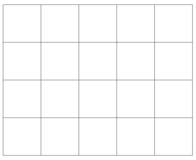
This is of course more tricky . . .

- 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)?



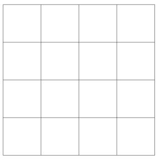
<u>idea 1:</u> just take the function value of the nearest midpoint („nearest neighbour interpolation“)

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



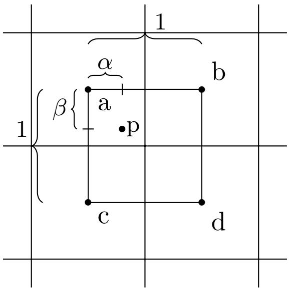
$\Rightarrow$ $u(x) = u(x_i)$ for all $x \in B_i$
$\Rightarrow$ each box is uni-color
$\Rightarrow$ the continuous image is essentially still discrete

<u>idea 2</u>: (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)

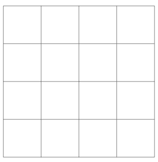
Similarly: $f = (1-\alpha)c + \alpha d$

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

$$\Rightarrow u(x) : = (1-\beta) \cdot e + \beta \cdot f$$
$$= (1-\beta)[(1-\alpha)a + \alpha b] + \beta[(1-\alpha)c + \alpha d]$$
$$= \underbrace{(1-\alpha)(1-\beta)}\, a + \underbrace{\alpha(1-\beta)}\, b + \underbrace{(1-\alpha)\beta}\, c + \underbrace{\alpha\beta}\, d$$
$$\in [0,1] \wedge \sum = 1$$

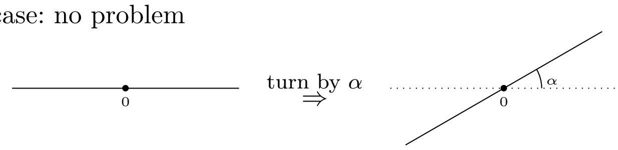
$\Rightarrow$ <u>convex combination</u> of the function values a, b, c, d at the the surrounding 4 midpoints (on which points is the nearest, instead of taking just a, b, c or d - depending)
$\Rightarrow$ 2D linear interpolation, *bi-linear interpolation* (can be interpreted as spline interpolation with bilinear basis splines).

Beispiel 2.2. Rotate image



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

- continuous image case: no problem



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

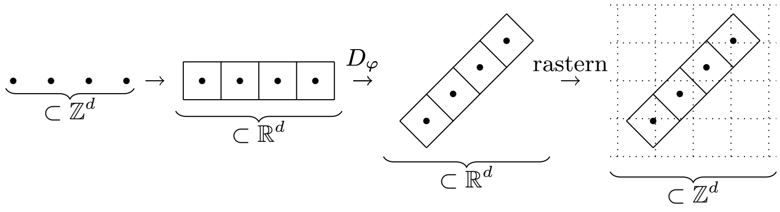
2D rotation matrix

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

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

- discrete image case: problem !

  For $x \in$ domain of notated image, in general $D_{-\varphi}\, x \notin$ domain of original image[1]
  Way out: $v(x) :=$ *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 → discrete: midpoint or average, discrete → continuous: nearest neighbour or bilinear?)

[1] 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 $u : \Omega \to F$ a discrete discrete image. The function

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

with

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

is called *histogramm* of the image u.

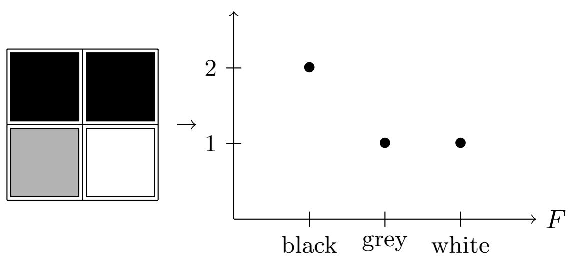
$H_u(k)$ counts how often colour k appears in u.

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

or

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

Beispiel 3.2.

$u =$ 

If u ist a continous image, H_u can be understood as a measure (generalized function)[1]. Another way to write this:

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

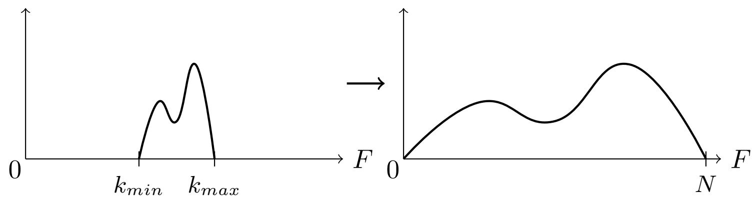
hier fehlt noch das Kronecker underarrow

Matlab-Code

[1] 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 histogramm over all of F.
Simple idea:



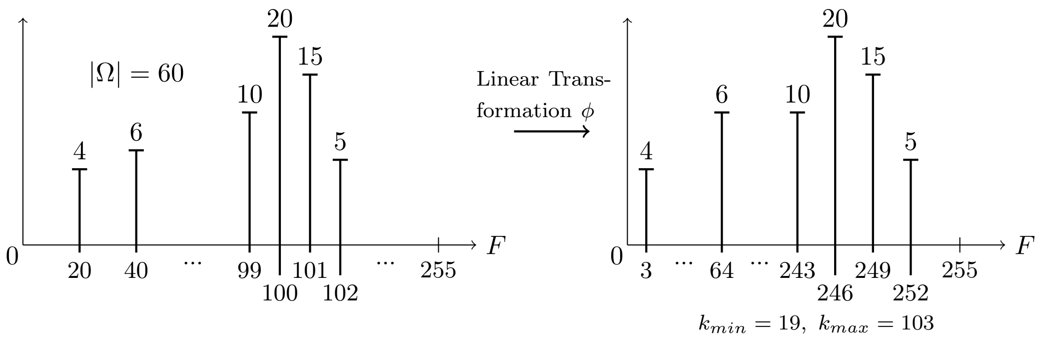
$$\begin{tikzcd}
\Omega \arrow[r, bend left, "\text{originales Bild}"] \arrow[rr, bend right, "\text{Bild mit besserem Kontrast}"'] & F \arrow[r, bend left, "\text{Zauberei } T"] & F
\end{tikzcd}
\qquad
\begin{array}{l} v = T \circ u \\ v(k) = T(u(k)) \end{array}$$

5

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

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

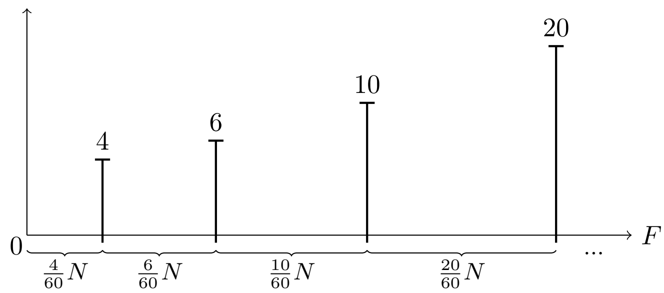
Where $[\ \cdot\ ]$ means ...rounding to the nearest integer (assumuning that $F = \{0, 1, \ldots, N\}$).
Example histogram:



A bit more sophisticated:

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

such that colour ranges that occur more frequently in u can occupy a larger range of colours in u. ($\Rightarrow$ visibility $\uparrow$)
Example histogramm spread out according to frequency of occurence:



$\Rightarrow$ „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 histogramm 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

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

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 histogramm splits into two „characteristic“ parts

For $t \in F$ put

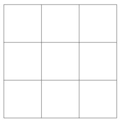
$$\text{black} := \{k \in F : k \leq t\}$$
$$\text{white} := \{k \in F : k > t\}$$

and

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

How to find the threshold t:

1.) Shape based methods
If the histogramm is „biomodal“
Put $t := \dfrac{k_{\max_1} + k_{\max_2}}{2}$
or $t := k_{\min}$

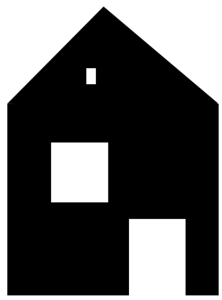


Kapitel 4

4.Basic Morphological Operations

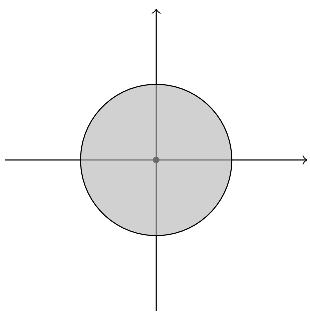
B/W Bild:

$A =$



Structural element :

$B =$



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.

$A + B =$

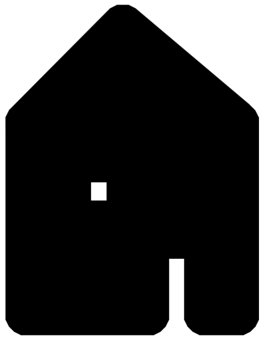


Image created in Matlab through:

```
I=imread('Bild1.png');
se=strel('disk',40,8);
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
imshow(I2);
```

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

This is called <u>erosion</u> .
You can imagine that you search for the points in which the structural element fits.

$A - B =$

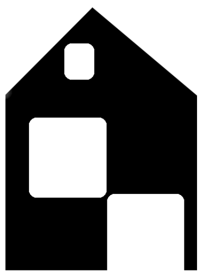


Image created in Matlab thorugh:

```
I=imread('Bild1.png');
se=strel('disk',20,8);
I2=imcomplement(imerode(imcomplement(I),se));
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.

$A \bullet B =$

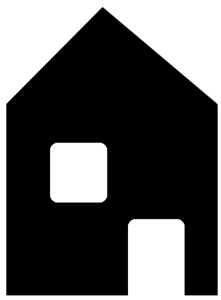


Image created in Matlab thorugh:

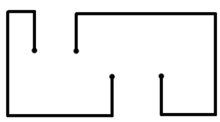
```
I=imread('Bild1.png');
se=strel('disk',20,8);
I2=imcomplement(imdilate(imcomplement(I),se));
I3=imcomplement(imerode(imcomplement(I2),se));
imshow(I3);
```

The inverse also exists:

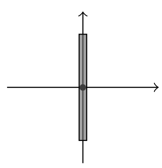
$$A \circ B := (A - B) + B$$

This is called opening .
This time with a new example:

$A =$



$B =$



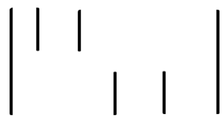
$$A \circ B =$$ 

Image created in Matlab thorugh:

```
I=imread('Bild2.png');
se=strel('line',10,90);
I2=imcomplement(imerode(imcomplement(I),se));
I3=imcomplement(imerode(imcomplement(I2),se));
imshow(I3);
```

Kapitel 5

5.Entrauschen: Filter und Co

5.1 Noise

<u>Noise</u> = Unwanted disturbances in an image. Mostly becaue of

- point wise
- random
- independent

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

$$\underset{\text{clean}}{\boxed{f_0}} \xrightarrow{+\text{ noise}} \underset{\text{disturbed image}}{\boxed{f}} \xrightarrow{\text{denoise}} \underset{\text{result}}{\boxed{u}}$$

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

$$||f - f_0|| \ldots \text{noise}$$
$$||u - f_0|| \ldots \underline{\text{absolute error}}$$
$$\frac{||u - f_o||}{||f - f_0||} \ldots \underline{\text{relative error}} \text{ compared to the noise}$$
$$\frac{||u - f_o||}{||f_0||} \ldots \text{relative error compared to the signal}$$

Typically the chosen norm is:

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

or in the discrete:

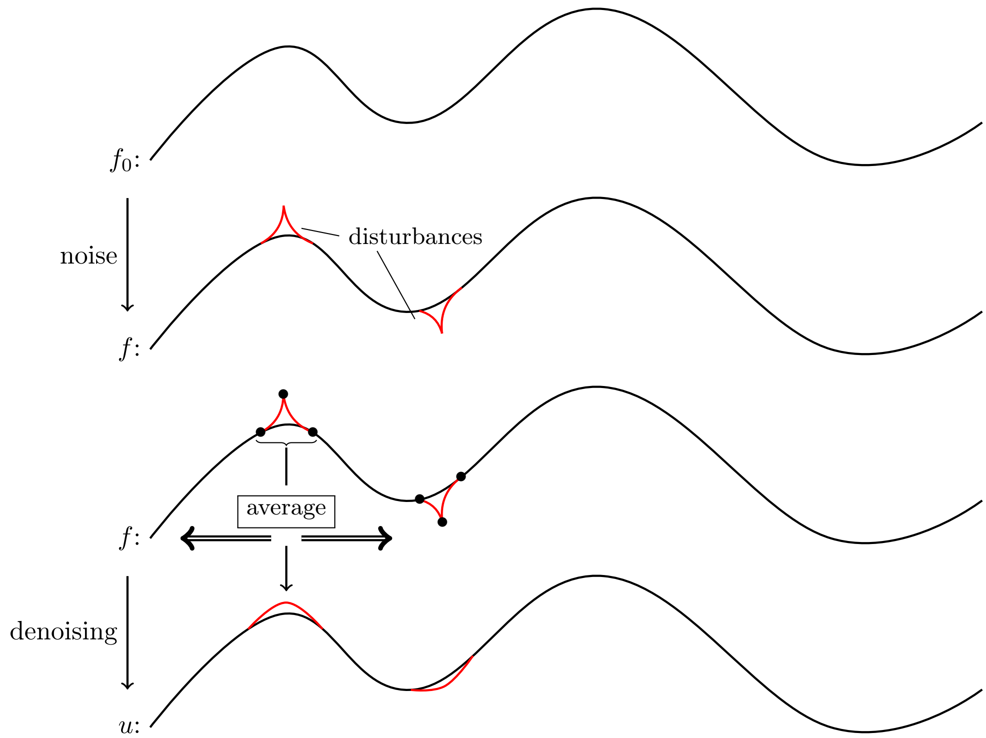
$$||f||_2 = \sqrt{\sum_{x \in \Omega} |f(x)|^2}$$

Closely connected is the <u>Signal to noise ratio</u> (SNR):

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

5.2 smoothing filter

Idea: (to simplify in 1D)

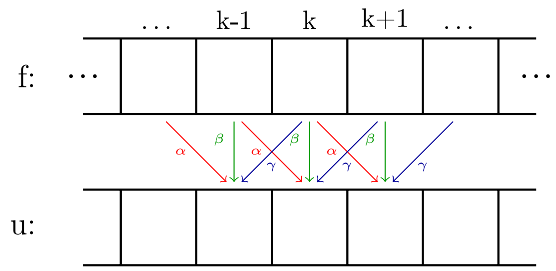


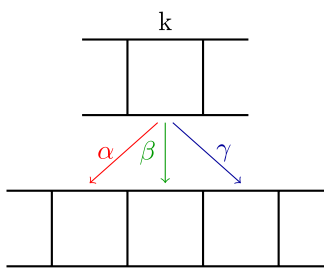
$$u(k) := \alpha \cdot f(k-1) + \beta \cdot f(k) + \gamma \cdot f(k+1) \tag{5.1}$$

where:

$$\alpha + \beta + \gamma = 1 \tag{5.2}$$

More precisely (5.1) means:





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

$u = m \boxtimes f$, this is called <u>Correlation</u> .

where:

$$\boxed{(m \circledast f)(k) = \sum_{i \in supp(m)} m(i) f(k+i)} \tag{5.3}$$

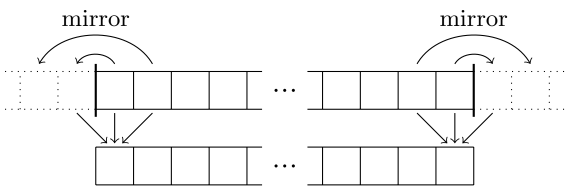
and:

$$\text{m}= \begin{array}{c|c|c|c|c} \ldots & \text{-1} & 0 & 1 & \ldots \\ \hline \cdots & \alpha & \beta & \gamma & \cdots \\ \hline \end{array} \quad \text{called } \underline{\text{mask}} .$$

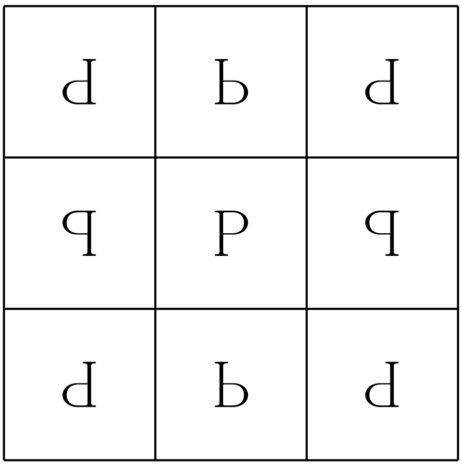
If you set $j := k + i$ in (5.1), then $i = j - k$, which means:

$$\boxed{(m \circledast f)(k) = \sum_{i \in supp(m)} m(j-k) f(j)} \tag{5.4}$$

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



in 2D:



Formula (5.4) might remind one of the <u>convolution</u> :

Layout!

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

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

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

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

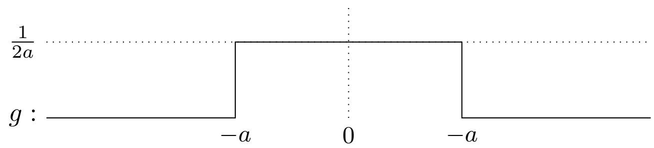
$\boxast$ und $*$ definiert man auf: $\ell^1(\mathbb{Z}^d) := \left\{ f = (f_i)_{i \in \mathbb{Z}^d} : \underbrace{\sum_{i \in \mathbb{Z}^d} |f_i|}_{:=||f||_1} < \infty \right\}$

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:

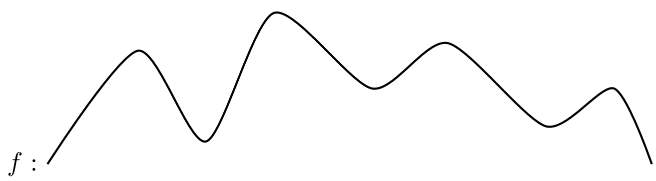
$$L^1(\mathbb{R}^d) := \left\{ f : \mathbb{R}^d \to \mathbb{R} : \underbrace{\int_{\mathbb{R}^d} |f| \, dx}_{:=||f||_1} < \infty \right\}$$

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

Beispiel für den kontinueirlichen Fall:

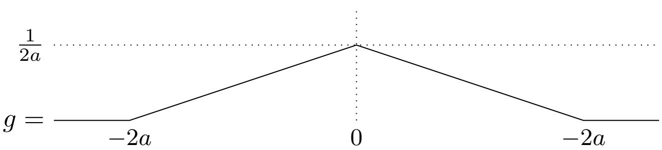


Hierbei gilt $\int_{\mathbb{R}} g(x) dx = 1$



$g \boxast f =$ gleitendes Mittel .

$g \boxast g = \tilde{g} * g = g * g =$



Layout!

Weitere Eigenschaften der Faltung:
Für alle $f, g \in L^1$ or ℓ^1

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

Somit ist:

$$g \mapsto f * g$$

ein linearer Operator.
Formt ℓ^1 bzw. L^1 eine Algebra mit neutralem Element δ?
ℓ^1?:

$$\delta\colon\quad \cdots \begin{array}{|c|c|c|c|c|}\hline 0 & 0 & 1 & 0 & 0\\ \hline\end{array} \cdots$$

$$\underset{\text{Pos } 0}{\uparrow}$$

Ja!
L^1?: Für ein solches Element muss gelten:
$\forall f \in L^1 : d * f = f$
$\forall x \in \mathbb{R} : \displaystyle\int_{\mathbb{R}^d} \underbrace{\delta(x-y)}_{=0 \forall x \neq y} f(y)dy = f(x)$

Diese Funktion wird <u>Dirac-Impuls</u> gennant ist aber kein Element von L^1.
<u>Nun zu Masken in 2D:</u>

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

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}.

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

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

<u>Symmetrischer Fall:</u>

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

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

$$\text{Erinnerung:}\quad \underset{\text{Sauberes Bild}}{\boxed{f_0}} \xrightarrow{+\text{ Rauschen}} \underset{\text{Gestörtes Bild}}{\boxed{f}} \xrightarrow{\text{Entrauschen}} \underset{\text{Resultat}}{\boxed{u}}$$

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

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

Layout!

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

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

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

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

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

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

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

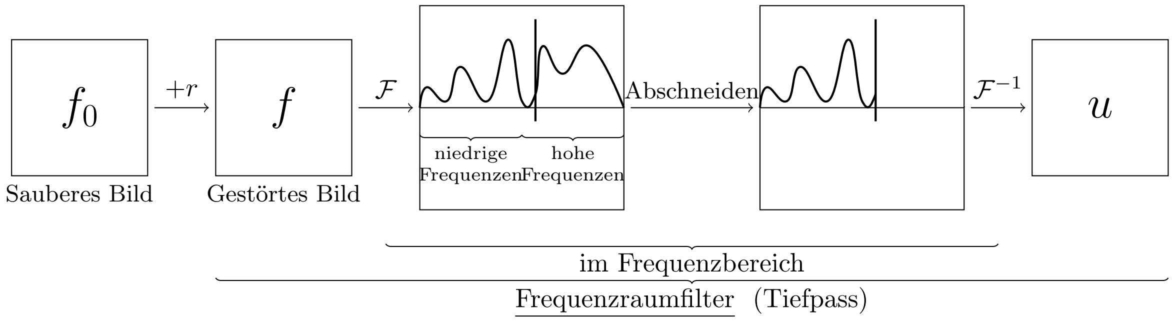
5.3 Frequenzfilter

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:

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

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

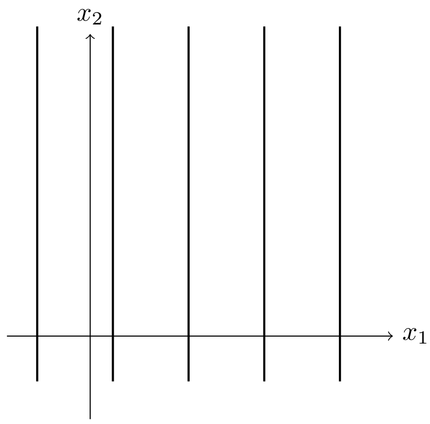
Wobei $x \in \mathbb{R}^d$.

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

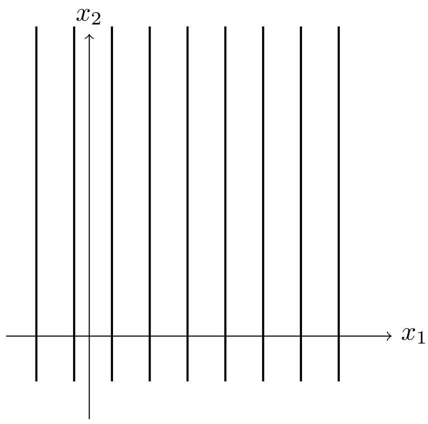
$$f(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \left(\frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} f(y) e^{-i\langle z,y\rangle} dy \right) e^{i\langle z,x\rangle} dz$$

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 $e_z(x) = e^{i\left\langle \binom{z_1}{z_2}, \binom{x_1}{x_2} \right\rangle} = e^{i(z_1x_1 + z_2x_2)}$
Beispiele in $2D$:
(Hier stellen die Linien, Punkte mit konstantem wert dar)

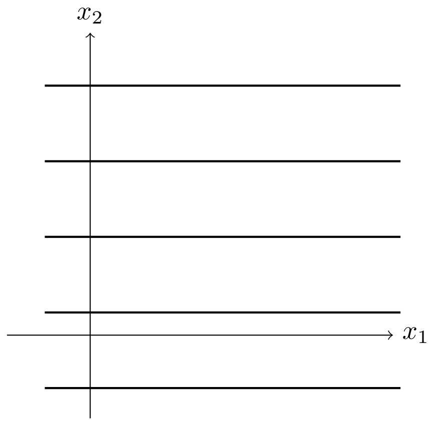
$z = \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right),\ e_z(x) = e^{ix_1}$:



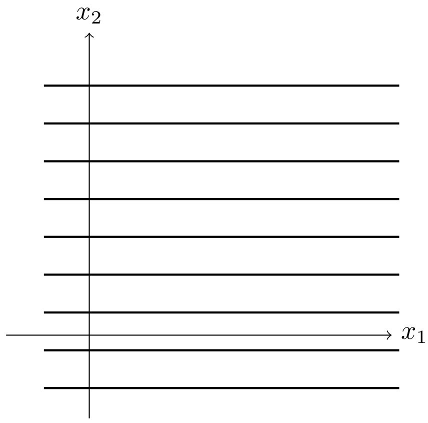
$z = \left(\begin{smallmatrix}2\\0\end{smallmatrix}\right),\ e_z(x) = e^{i2x_1}$:



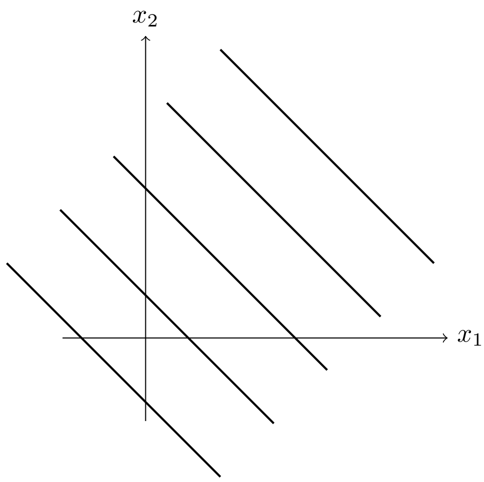
$z = \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right),\ e_z(x) = e^{ix_2}$:



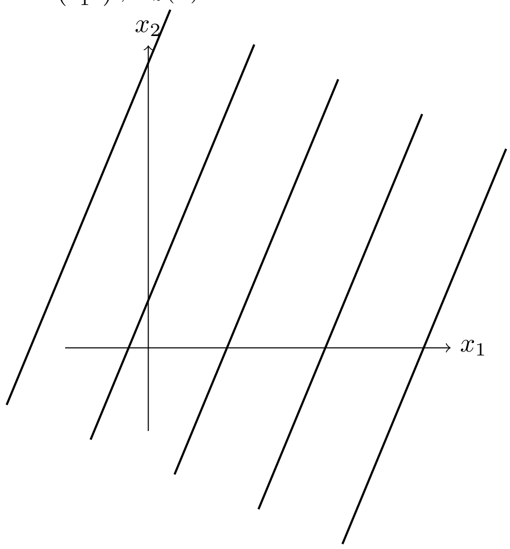
$z = \left(\begin{smallmatrix}0\\2\end{smallmatrix}\right),\ e_z(x) = e^{i2x_2}$:



$z = \left(\begin{smallmatrix}1\\1\end{smallmatrix}\right),\ e_z(x) = e^{i(x_1+x_2}$:



$z = \left(\begin{smallmatrix}-2\\1\end{smallmatrix}\right),\ e_z(x) = e^{i(-2x_1+x_2)}$:



$f \in L^2(\mathbb{R}^d) = \{f : \mathbb{R}^d \to \mathbb{R} | \int_{\mathbb{R}^d} |f|^2 \, dx < \infty\}$ 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 <u>Banachraum</u>

Ein vollständiger normierter Banachraum mit Skalarproduk 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

$$\left|\left|\hat{f}\right|\right|_2 = ||f||_2 \tag{5.8}$$

und sogar

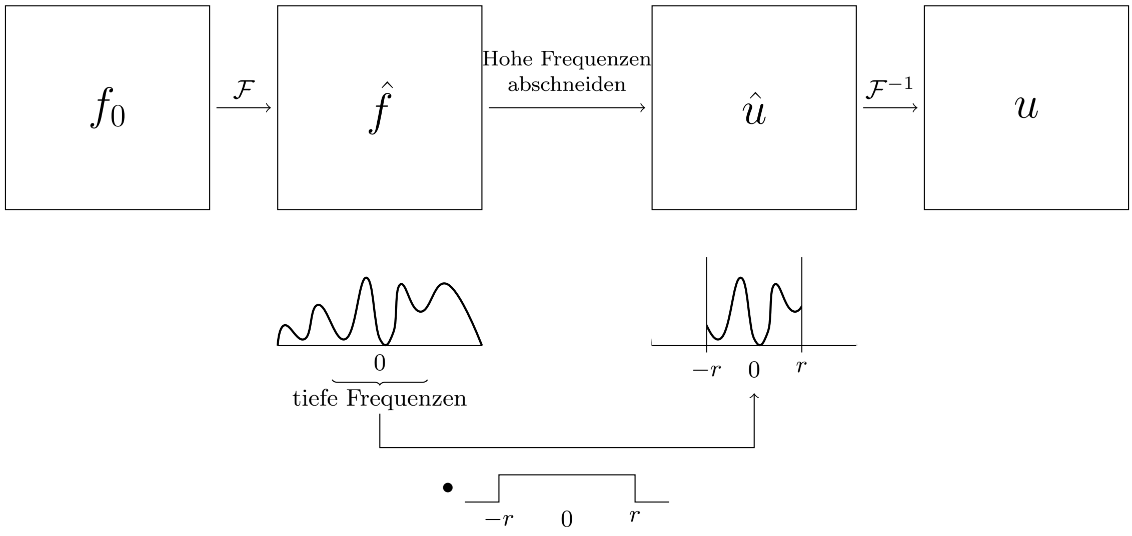
$$\left\langle \hat{f}, \hat{g} \right\rangle_2 = \langle f, g\rangle_2 \tag{5.9}$$

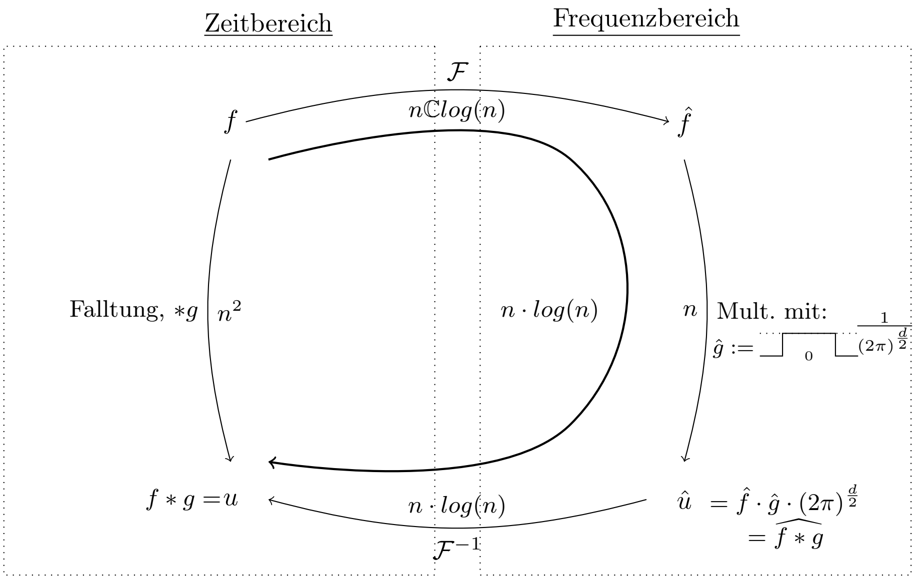
für alle $f, g \in L^2(\mathbb{R}^d)$.
Weitere Eigenschaften der Fouriertransformation:

- $f \in L^1(\mathbb{R}^d) \Rightarrow \hat{f}$ stetig und $\underset{|z|\to\infty}{lim} \hat{f}(z) = 0$
- $\mathcal{F} : L^1(\mathbb{R}^d) \to C(\mathbb{R}^d)$ ist eine lineare Abbildung
- $\mathcal{F} : L^1(\mathbb{R}^d) \to C(\mathbb{R}^d)$ ist eine beschränkte/stetige Abbildung
- Verschiebung $\overset{\mathcal{F}}{\to}$ Modulation, d.h.
$$g(x) = f(x + a) \Rightarrow \hat{g}(z) = e^{i\langle a,z\rangle}\hat{f}(z)$$
- Modulation $\overset{\mathcal{F}}{\to}$ Verschiebung, d.h.
$$g(x) = e^{i\langle x,a\rangle} f(x) \Rightarrow \hat{g}(z) = \hat{f}(z - a)$$
- Skalierung $\overset{\mathcal{F}}{\to}$ inverse Skalierung, d.h.
$$g(x) = f(cx) \Rightarrow \hat{g}(z) = \frac{1}{|c|}\hat{f}(\frac{z}{|c|})$$
- Konjugation: $g(x) = \overline{f(x)} \Rightarrow \hat{g}(z) = \overline{\hat{f}(-z)}$
  Folglich: f reelwertig $\Rightarrow \hat{f}(z) = \overline{\hat{f}(-z)}$
-
$$\text{Grundmode: } \hat{f}(0) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} f(x)dx$$
$$\text{Analog: } f(0) = \frac{1}{(2\pi)^{\frac{d}{2}}} \int_{\mathbb{R}^d} \hat{f}(x)dx$$
- Differentation $\overset{\mathcal{F}}{\to}$ Multiplikation mit Potenzen von z, d.h.
$$g(x) = \frac{\partial^{\alpha_1+\cdots+\alpha_d}}{\partial x_1^{\alpha_1}\cdots\partial x_d^{\alpha_d}} f(x) \Rightarrow \hat{g}(z) = i^{\alpha_1+\cdots+\alpha_d} z_1^{\alpha_1}\cdots z_d^{\alpha_d}\hat{f}(z)$$
- Unkehrung des letzten Punktes:
$$g(x) = x_1^{\alpha_1}\cdots x_d^{\alpha_d} f(x) \Rightarrow \hat{g}(z) = i^{\alpha_1+\cdots+\alpha_d}\frac{\partial^{\alpha_1+\cdots+\alpha_d}}{\partial x_1^{\alpha_1}}\hat{f}(z)$$
-
$$\text{Faltungssatz: } \mathcal{F}(f * g) = (2\pi)^{\frac{d}{2}}\mathcal{F}(f)\cdot\mathcal{F}(g),\ \widehat{f * g} = (2\pi)^{\frac{d}{2}}\hat{f}\cdot\hat{g}$$
$$\text{Analog: } \mathcal{F}(f\cdot g) = \frac{1}{(2\pi)^{\frac{d}{2}}}\mathcal{F}(f) * \mathcal{F}(g),\ \widehat{f\cdot g} = \frac{1}{(2\pi)^{\frac{d}{2}}}\hat{f} * \hat{g}$$

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

Zur Erinnerung:





Genauer:

$$\mathcal{F}u = \hat{v} =$$

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

$$\hat{g}(0) = (\mathcal{F}g)(0) = \frac{1}{2}$$

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

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

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

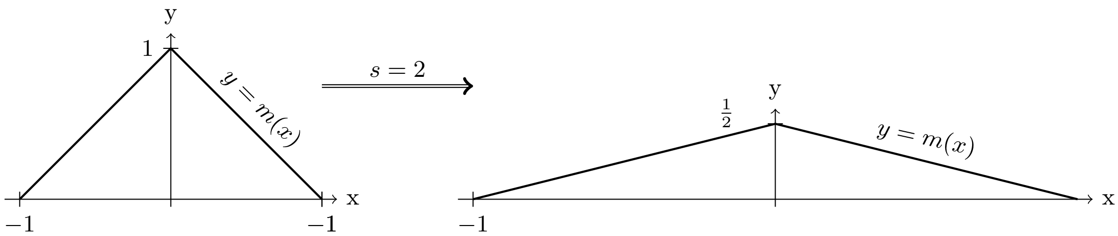
'glättet mehr als': $\frac{1}{9}$

1	1	1
1	1	1
1	1	1

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

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

Bsp (in $d = 1$):



Bsp: Gauß-Kern $G(x) = \frac{1}{(2\pi)^{\frac{d}{2}}} e^{\frac{-|x|^2}{2}}$
Skalierung mit Fehler $s > 0$

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

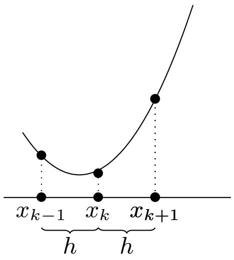
Skalierung $s \widehat{=}$ Standardabweichung σ

hier noch mehr im Skript p. 45

5.5 Differenzenfilter

Bisher: Glättung $\widehat{=}$ Mittelwert bilden $\widehat{=}$ Summe/Integrale
Jetzt: Schärfen $\widehat{=}$ Differenzen/Kontraste hervorheben $\widehat{=}$ Differenzen/Ableitungen

Diskretisierung von Ableitungen durch Differenzenquotienten



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

Vorwärts: $u(h) = \dfrac{f(k+1) - f(k)}{h}$ $\quad u = \dfrac{1}{h}\begin{array}{|c|c|c|}\hline 0 & -1 & 1\\\hline\end{array} \circledast f$

Rückwärts: $u(h) = \dfrac{f(k) - f(k-1)}{h}$ $\quad u = \dfrac{1}{h}\begin{array}{|c|c|c|}\hline 0 & -1 & 1\\\hline\end{array} \circledast f$

Zentral: $u(h) = \dfrac{f(k+1) - f(k-1)}{2h}$ $\quad u = \dfrac{1}{2h}\begin{array}{|c|c|c|}\hline 0 & -1 & 1\\\hline\end{array} \circledast f$

2. Abbleitung:

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

Also folgt $u := \begin{array}{|c|c|c|}\hline 1 & -2 & 1\\\hline\end{array} \circledast f$ und $\dfrac{1}{h^2}\begin{array}{|c|c|c|}\hline 1 & -2 & 1\\\hline\end{array} = \dfrac{1}{h}\begin{array}{|c|c|c|}\hline 0 & -1 & 1\\\hline\end{array} * \dfrac{1}{h}\begin{array}{|c|c|c|}\hline -1 & 1 & 0\\\hline\end{array}$

Denn:

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

In 2D: $\dfrac{\partial}{\partial x} \widehat{=} \begin{array}{|c|c|c|}\hline 0 & -1 & 1\\\hline\end{array}$, $\dfrac{\partial}{\partial y} \widehat{=} \begin{array}{|c|}\hline 0\\\hline -1\\\hline 1\\\hline\end{array}$, $\dfrac{\partial^2}{\partial x^2} \widehat{=} \begin{array}{|c|c|c|}\hline 1 & -2 & 1\\\hline\end{array}$, $\dfrac{\partial^2}{\partial y^2} \widehat{=} \begin{array}{|c|}\hline 1\\\hline -2\\\hline 1\\\hline\end{array}$.

<u>Diskreter Laplace Operator</u> :

$$
\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \widehat{=} \begin{array}{|c|c|c|}\hline 1 & -2 & 1\\\hline\end{array} + \begin{array}{|c|}\hline 1\\\hline -2\\\hline 1\\\hline\end{array} = \begin{array}{|c|c|c|}\hline 0 & 1 & 0\\\hline 1 & -4 & 1\\\hline 0 & 1 & 0\\\hline\end{array}
$$

5.6 Glättungsfilter und partielle Differentialgleichungen

Wir haben gesehen: $m = \dfrac{1}{5}\begin{array}{|c|c|c|}\hline 0 & 1 & 0\\\hline 1 & 1 & 1\\\hline 0 & 1 & 0\\\hline\end{array}$ ist unter allen 5-Punkt Filtern der am besten glättende.

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

$$\boxed{\underset{:=u^{(0)}}{f}} \xrightarrow{m\circledast} \boxed{u^{(1)}} \xrightarrow{m\circledast} \boxed{u^{(2)}} \cdots$$

$$\begin{aligned}
\Rightarrow u^{(n+1)} - u^{(n)} &= (\text{Unterschied zwischen 'Zeit' Punkt } n \text{ und } n+1) \\
&= \underbrace{m \circledast u^{(n)}}_{u^{n+1}} - \underbrace{\delta \circledast u^{(n)}}_{u^{(n)}} \text{ mit } \delta = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \\
&= (m - \delta) \circledast u^{(n)} \\
&= \left(\frac{1}{5} \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} - \frac{1}{5} \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 5 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \right) \circledast u^{(n)} \\
&= \frac{1}{5} \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} u^{(n)}
\end{aligned}$$

noch einmal schauen was 5.10 ist (5.10)

Somit gilt insgesamt:

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

Kontinuierlich: Funktion u

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

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

$$\frac{\partial u}{\partial t} = \Delta u \tag{5.12}$$

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

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

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 unteranderem von konstanten Funktionen oder linearen Funktionen $u(x_1, x_2) = ax_1 + bx_2$ erfüllt.

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

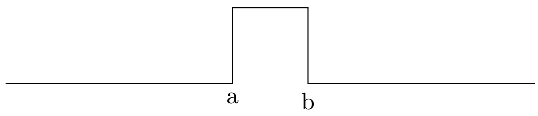
$$u(x,t) = \left(G_{\sqrt{2t}} * u^{(0)} \right)(x)$$

Wobei $\sqrt{2t}$ für eine Skalierung um diesen Wert steht.
Zu zeigen ist: $\dfrac{\partial u}{\partial t} = \Delta u$

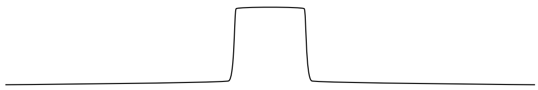
$$\begin{aligned}
\frac{\partial}{\partial t} \left(G_{\sqrt{2t}} * u^{(0)} \right) &= \Delta \left(G_{\sqrt{2t}} * u^{(0)} \right) \\
\overset{\text{mit Satz}}{\Longrightarrow} \left(\frac{\partial}{\partial t} G_{\sqrt{2t}} \right) * u^{(0)} &= \left(\Delta G_{\sqrt{2t}} \right) * u^{(0)}
\end{aligned}$$

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

$t = 0$:

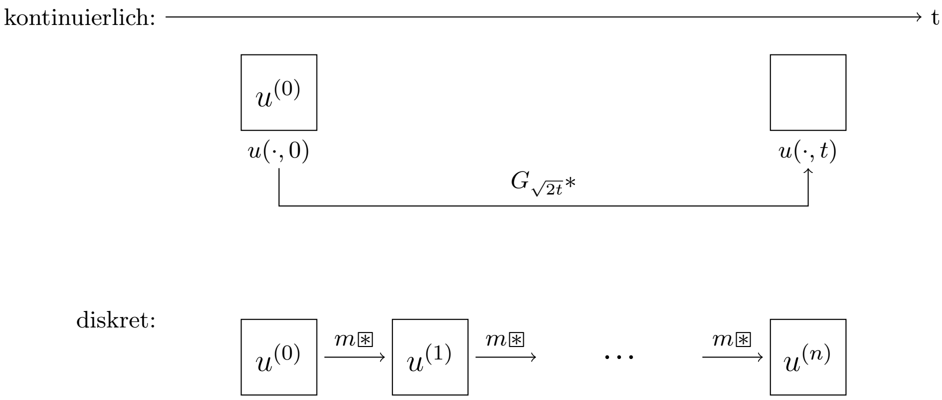


$t > 0$:



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:



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.:

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

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

Diffusionsgleichung:

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

Ansatz für M:

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

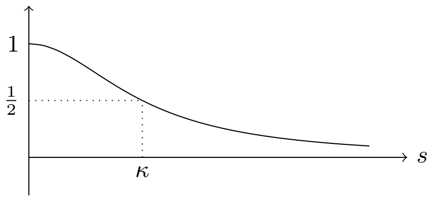
b) $M = g(||\nabla u(x,y)||) \cdot I$

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

$\Rightarrow$ Perona & Malik (1990)

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

$$g_\kappa(s) = \frac{1}{1+\left(\frac{s}{\kappa}\right)^2}$$

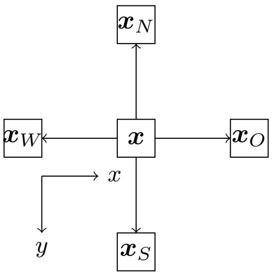


- Kante mit $||\nabla u|| < \kappa$ werden gelättet ($g > \frac{1}{2}$)
- Kante mit $||\nabla u|| \geq \kappa$ werden nicht geglättet ($g \leq \frac{1}{2}$)

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



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

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 $f(\mathbf{x})$

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

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

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

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: $f \overset{\log}{\mapsto} \log f \overset{\text{bil Filter}}{\mapsto} \log u \overset{\exp}{\mapsto} u$

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:

$$||u - f||_2 + \lambda \cdot ||\nabla||_2^2 \overset{u \in U}{\longrightarrow} \min \tag{5.15}$$

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 Gradiente an (gewollten Kanten) zu stark bestraft. ($\Rightarrow$ optimales u hat geglättete Kanten)
Ausweg: Wähle $||\nabla u||_2 = \sqrt{s.o.}$ oder $||\nabla u||_1 = \int |\nabla u(x)| dx = \int_\Omega (|\frac{\partial u}{\partial x}(x)| + |\frac{\partial u}{\partial y}(x)|) dx$

$$||u - f||_2 + \lambda \cdot ||\nabla||_1 \overset{u \in U}{\longrightarrow} \min \tag{5.16}$$

Genannt <u>Rudin–Osher–Fatemi-Funktional</u> (ROF)
Allgemeiner Ansatz bei Variationsproblemen:

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

Notwendiges Kriterium:
Falls $J : U \to \mathbb{R}$ in $u \in U$ ein lokales Minimum besutzt, dann gilt für jede Richtung $v \in U$:

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

Dies ist die Verallgemeinerte Richtungsableitung (Gateux-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:

$$\boxed{\text{Filter}} \xrightarrow{\text{wiederholte Anwendung}} \boxed{\text{partielle Differentialgleichung}} \xleftarrow{\text{Euler-Lagrange}} \boxed{\text{Variationsrechnung}}$$

Kapitel 6

6.Kantenerkennung

6.1 Gradientenfilter

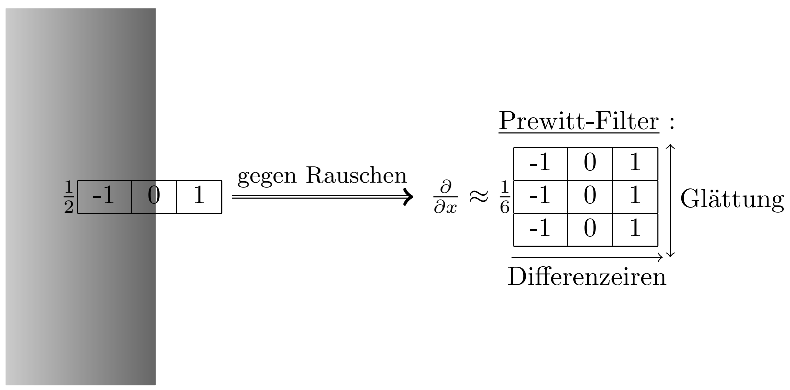
Wir suchen Stellen $\boldsymbol{x}$ mit großem Gradienten:

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

Approximation der Gradienten über zentrale Differenzen:

$$\frac{\partial}{\partial x} \approx \frac{1}{2}\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline\end{array} \text{ bzw. } \frac{\partial}{\partial y} \approx \frac{1}{2}\begin{array}{|c|}\hline -1 \\ \hline 0 \\ \hline 1 \\ \hline\end{array} \tag{6.1}$$

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



Alternative: $\frac{\partial}{\partial x} \approx \frac{1}{2}\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline\end{array} \circledast \frac{1}{4}\begin{array}{|c|}\hline -1 \\ \hline 0 \\ \hline 1 \\ \hline\end{array} = \frac{1}{8}\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline\end{array} =: D_x$, genannt <u>Sobel-Filter</u> . Eine stärkere Glättung kann mittels anderer vertikaler Filter mit Binomialkoeffizienten erzeilt werden.

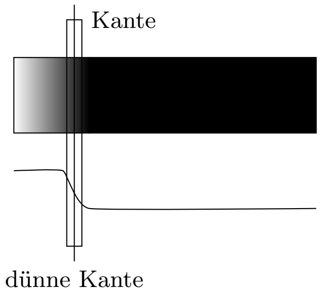
Entsprechen wird $\frac{\partial}{\partial y} D_y := D_x^T$ definiert.

$$\nabla u(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial u}{\partial x}(\boldsymbol{x}) \\ \frac{\partial u}{\partial y}(\boldsymbol{x}) \end{pmatrix} \approx \begin{pmatrix} (D_x \circledast u)(\boldsymbol{x}) \\ (D_y \circledast u)(\boldsymbol{x}) \end{pmatrix} \tag{6.2}$$

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

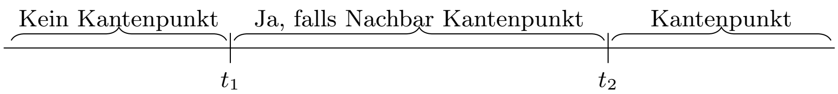


Mathematisch: $\boldsymbol{x}$ wird Kantenpunkt falls:

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

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

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



wobei $t_1,\ t_2$ thresholds sind.
$\boldsymbol{x}$ ist also ein Kantenpunkt, falls $|\nabla u(\boldsymbol{x})| \geq t_2$ oder ($|\nabla u(\boldsymbol{x})| \in [t_1, t_2]$ und $\boldsymbol{x}$ 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:

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

BWimg: Binärbild

u: Graustufenbild

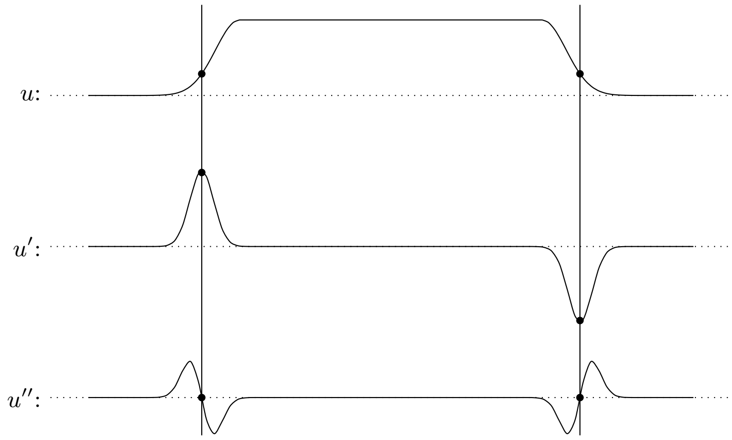
canny: Algorithmus

$t_1,\ t_2$: Sind gewählt wie oben

sigma: Parameter für den Gaußkern aus 1)

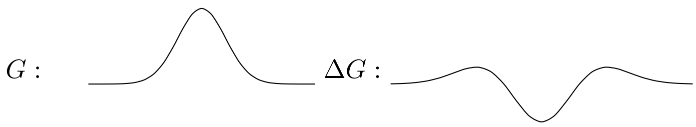
6.2 Die zweite Ableitung

Zunächst in 1D:

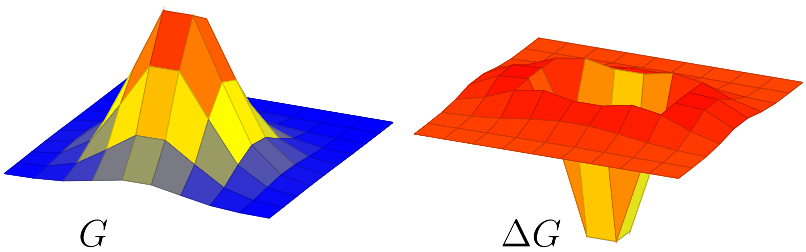


Test für kantenpunkte $u''(\boldsymbol{x}) = 0$ und $|u'(\boldsymbol{x})| > \text{threshold}$.
Wichtig: Vorglätten!, da die 2. Ableitung noch anfälliger gegenüber Rauschen als die 1. Ableitung ist.

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:

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

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

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

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

Kapitel 7

7.Schaerfen und Entfalten

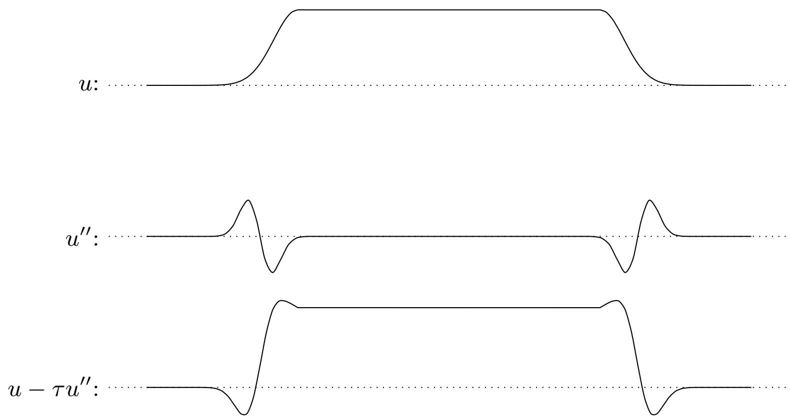
(Gegenteil von Kapitel 5)

Gegeben: unscharfes Bild

Gesucht: Version mit vielen erkennbaren Details

7.1 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 $\Delta statt$ 2. Ableitung

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

7.2 Kantenverstärkende Diffusion

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

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

lbbel=- senkrecht zu ∇u (entlang der Kante): $\Lambda_2 = 1$

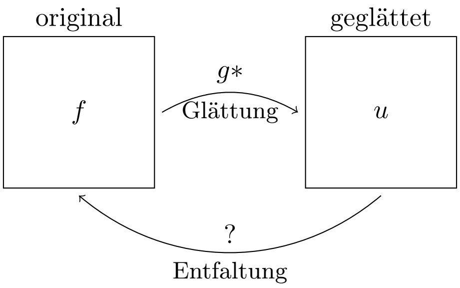
$\Rightarrow M$ hat EW λ_1 zum EV $v_1 = \dfrac{\nabla u}{|\nabla u|} \begin{pmatrix} \frac{\partial u}{\partial x}(\boldsymbol{x}) \\ \frac{\partial u}{\partial y}(\boldsymbol{x}) \end{pmatrix}$ und EW λ_2 zum EV $v_2 = \dfrac{1}{|\nabla u|} \begin{pmatrix} -\frac{\partial u}{\partial y}(\boldsymbol{x})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

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

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

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

7.3 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:

$$
\begin{aligned}
& g * f = u \\
& (g * f)(j) = u(j),\ j \in \Omega \\
& \sum_k g(j-k) f(k) = u(j),\ j \in \Omega \\
& \Rightarrow \Omega \times \Omega \text{ Gleichungssystem}
\end{aligned}
$$

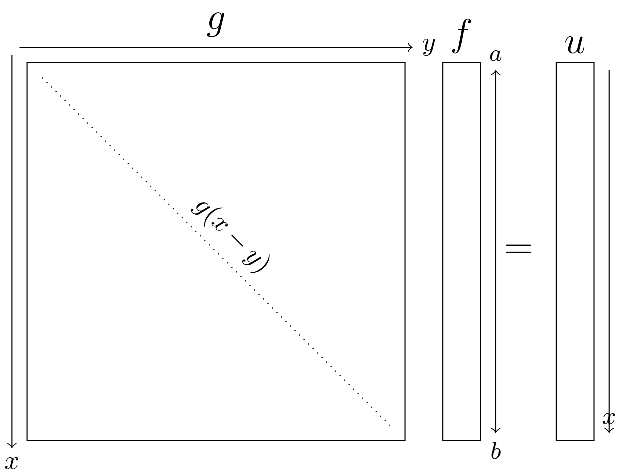
$$
\underbrace{
\begin{array}{c}
\\
j=0 \\
j=1 \\
\\
\\
j=n
\end{array}
\begin{array}{c}
\begin{array}{cccc} k=0 & k=1 & \qquad & k=n \end{array} \\
\begin{pmatrix}
g(0) & g(-1) & & g(-n) \\
g(1) & g(0) & g(-1) & \\
 & \ddots & \ddots & \ddots \\
g(n) & & &
\end{pmatrix}
\end{array}
}_{\underline{\text{Toeplitz-Matrix}}}
\begin{pmatrix} f(0) \\ f(1) \\ \vdots \\ f(n) \end{pmatrix}
=
\begin{pmatrix} u(0) \\ u(1) \\ \vdots \\ u(n) \end{pmatrix}
$$

2. Kontinuierlicher Fall:

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

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:

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

Und erhalten:

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

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 $z \to \infty$.

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:

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

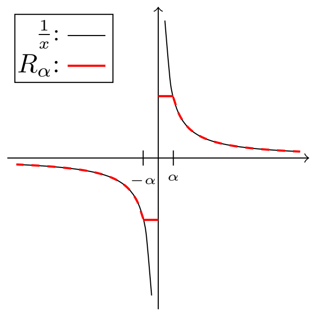
Problembehebung:

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

1. Ansatz:

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

wobei $\alpha > 0$.



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

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

und lasse $\alpha \to 0$.

2. Ansatz: Variationsrechnung:
   1. Wunsch: $g * f \approx u$
   2. Wunsch $||f||_2$ klein

Minimiere nun:

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

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

über die Wahl von $f \in U := L^2(\mathbb{R}^d)$.
Idee: $\mathcal{F}$ anwenden $\Rightarrow *$ wird zu $\mathbb{C}$ und $||\mathbb{C}||_2$ bleibt unverändert.

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

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:

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

Das Argument (Winkel) taucht nur in $\circledast$ auf

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

$$\begin{aligned}
&\Rightarrow 0 = arg(\circledast) = arg(\hat{g}(z)\ \overline{\hat{u}(z)}) + arg(t) \\
&\Rightarrow arg(t) = -arg(\hat{g}(z)\ \overline{\hat{u}(z)}) = arg(\overline{\hat{g}(z)}\ \hat{u}(z)) \\
&\Rightarrow I(t) = ((2\pi)^{\frac{d}{2}}|\hat{g}(z)|^2 + \lambda)|t|^2 - (2\pi)^{\frac{d}{2}}2\mathbb{C}|\overline{\hat{g}(z)}\ \hat{u}(z)|\ |t| + |\hat{u}(z)|^2 \xrightarrow{|t|\in\mathbb{R}} min
\end{aligned}$$

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

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

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

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

Dieses Verfahren wird $\underline{L^2 \text{ deblurring}}$ gennant. Es gibt auch einen alternativen, algebraischen Zugang:

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

$\Rightarrow$ lineares Ausgleichsproblem.

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

$$\Rightarrow \left(A^*A + |\lambda|\,I\right) f = A^*u$$

$$\Rightarrow f = \left(A^*A + |\lambda|\,I\right)^{-1} A^*u$$

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:

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

$\Rightarrow$ Dämpfung höher wenn Frequenz höher.
Dieses Verfahren nennt sich $\underline{H^1 \text{ deblurring}}$

Kapitel 8

8.Restauration und Inpainting

Problem: Lücken im Bild, etwa

1. Kratzer
2. Scannerzeile kaputt
3. Defekt in der Kamera
4. Bewusst entferntes Objekt

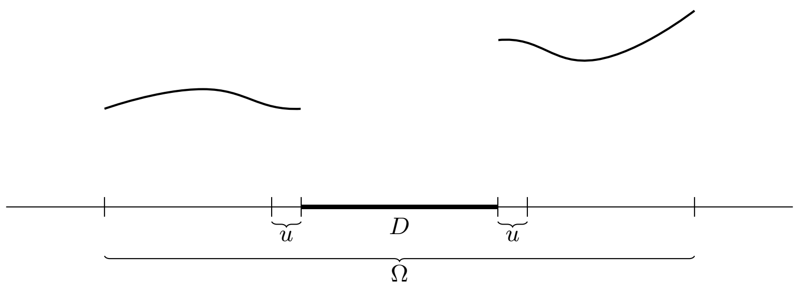
sollen sinvoll und unauffällig geschlossen werden.
Sei $f : \Omega \to F$ unser Bild jedoch mit Defekt, d.h. fehlenden Funktionswerten in $D \subset \Omega$.

1. Fall: Jeder Punkt aus D hat Nachbarn in $\Omega \backslash D$.
$\Rightarrow$ Lücken mittels Interpolation aus benachbarten Werten in $\Omega \backslash D$ schließen.

2. Fall: D hat innere Punkte. Diesen Fall werden wir im folgenden näher betrachten.

8.1 Frequenzraum-Ansatz

Zur Illustration in 1D:



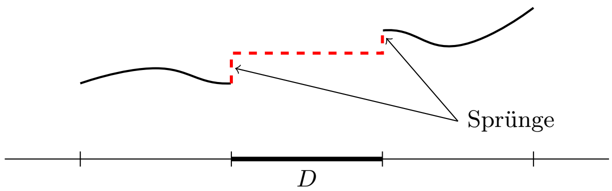
Betrachte Umgebung $u \subset \Omega \backslash D$ und errechne den Mittelwert:

$$m := \frac{1}{|u|} \int_u f(x)dx$$

von f auf u.

Algorithmus:

[1] Initialisiere f auf D mittels konstanter Funktion m:

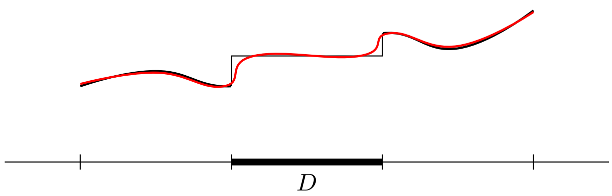


$\Rightarrow$ Sprünge am Rand von D.
Idee: Sprünge $\widehat{=}$ hochfrequente Anteile $\Rightarrow$ wende Tiefpass filter an.

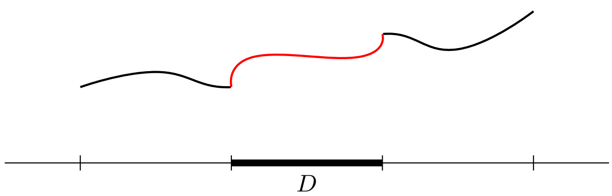
Iteriere

Abbruch falls $||f-g|| < \textit{threshold}$

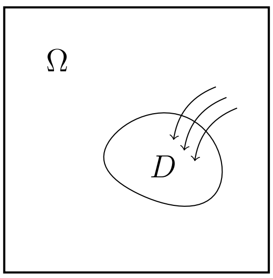
[2] $f \xrightarrow{\mathcal{F}} \hat{f} \xrightarrow{\text{Lösche hohe Frequenzen}} \hat{g} \xrightarrow{\mathcal{F}^{-1}} g$



[3] Ersetze f <u>innerhalb</u> von D durch g.



8.2 PDE-Transport-Diffusions-Ansatz



Idee: Informationen aus $\Omega \backslash D$ nach D "hineintragen".

Referenzen:
- Weichert 1998
- Bornemann & März 2007

Skalierung der Diffusion:

$$\frac{\partial u}{\partial t} = div(M \nabla u)$$

$$\text{mit } M = \begin{pmatrix} | & | \\ v_1 & v_2 \\ | & | \end{pmatrix} \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} \begin{pmatrix} - & v_1^T & - \\ - & v_2^T & - \end{pmatrix} \in R^{2\times 2}$$

Wobei $v_1 \perp v_2$ die Eigenvektoren des sogenannten <u>doppelt geglätteten Strukturrensors</u>

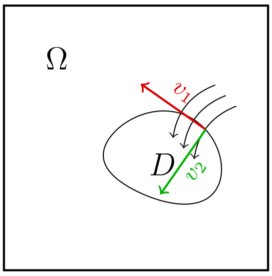
$$J = G_\rho * \left[\underbrace{\begin{pmatrix} | \\ \nabla(G_\sigma * u) \\ | \end{pmatrix}}_{2\times 1} \cdot \underbrace{\begin{pmatrix} - & \nabla^T(G_\sigma * u) & - \end{pmatrix}}_{1\times 2} \right]$$

sind.

$\Rightarrow$ v_1 Richtung mit maximalem Kontrast mit EW μ_1

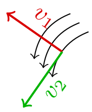
v_2 Richtung mit maximalem Kontrast mit EW μ_2, genannt <u>Kohärentsrichtung</u> .

Hierbei ist $\mu_1 \geq \mu_2$.



Fälle:

$\mu_1 \gg \mu_2 \approx 0$



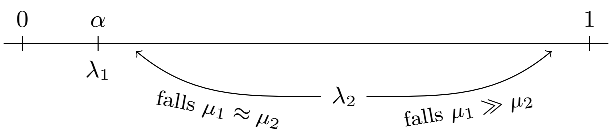
$\mu_1 \approx \mu_2 \approx 0$

Lokal keine Struktur.

$\mu_1 \approx \mu_2 \gg 0$

Kanten.

Die Werte λ_1 und λ_2 werden wie folgt gewählt:
$\alpha \in (0,1)$ wird festgehalten.



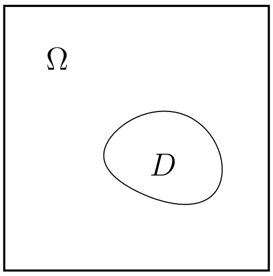
$$\lambda_1 := \alpha, \ \lambda_2 := \alpha + (1-\alpha)(1 - g(\mu_1 - \mu_2))$$

wobei g wie bei Perona Malik gewählt wird, also:

$$g(x) = \frac{1}{1 + \frac{s^2}{\kappa^2}}$$

Dieses wird <u>Kohärenz verstärkende Diffusion</u> genannt.

8.3 Variationsansatz



geg.: f auf $\Omega\backslash D$

ges.: u auf Ω

Wunsch 1: $u = f$ auf $\Omega\backslash D$

Wunsch 2: $||\nabla u||$ klein auf Ω

Daraus folgt:

$$J(u) := ||\nabla u||_2^2 \to \min$$

auf

$$U := \{u \in W^{1,2}(\Omega) : u|_{\Omega\backslash D} = f\}$$

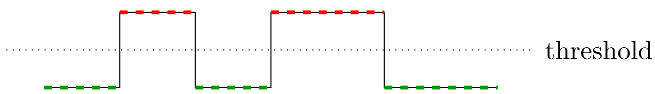
Angenommen, $u \in U$ minimiert J, dann folgt für beliebige $v \in W^{1,2}(\Omega)$ mit $v|_{\Omega\backslash D} = 0$:

$$\begin{aligned}
0 &= \lim_{t\to 0} \frac{J(u+tv) - J(u)}{t} = \lim_{t\to 0} \frac{1}{t} \int_\Omega ||\underbrace{\nabla(u+tv)(x)}_{\nabla u(x) + t\nabla v(x)}||^2 - ||\nabla u(x)||^2 \, dx \\
&= \lim_{t\to 0} \frac{1}{t} \int_\Omega \langle \nabla u(x) + t\nabla v(x), \nabla u(x) + t\nabla v(x)\rangle - \langle \nabla u(x), \nabla u(x)\rangle \, dx \\
&= \lim_{t\to 0} \frac{1}{t} \int_\Omega t^2 \, ||\nabla v(x)||^2 + 2t \langle \nabla u(x), \nabla v(x)\rangle \, dx = \lim_{t\to 0} \int_\Omega t \, ||\nabla v(x)||^2 + 2 \langle \nabla u(x), \nabla v(x)\rangle \, dx \\
&= 2 \int_D \langle \nabla u(x), \nabla v(x)\rangle \, dx \overset{\text{Greensche Formel}}{=} 2\left(\overbrace{\int_{\delta D} \frac{\partial u}{\partial n} \underbrace{v(x)}_{0} \, ds(x)}^{0} - \int_D \Delta u(x) v(x) dx\right) \\
&= 2 \int_D \Delta u(x) v(x) dx \Rightarrow \nabla u = 0 \text{ in } D
\end{aligned}$$

Kapitel 9

9.Segementierung

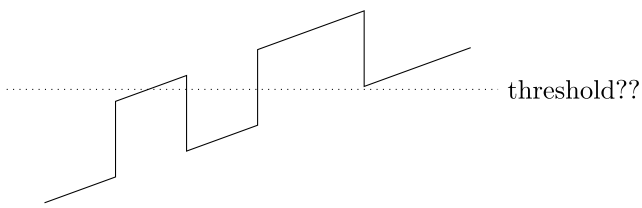
Dieses ist die Zerlegung eines Bildes in verschiedene Objekt.
Eine einfach methode hierfür ist das Historgramm thresholding :



So kann ein Bild in mehrere Objekte zerlegt werden.
Hierbei können jedoch diverse Probleme auftreten, die durch preprocessing vermindert werden sollten. Einige der preprocessing methoden sind:

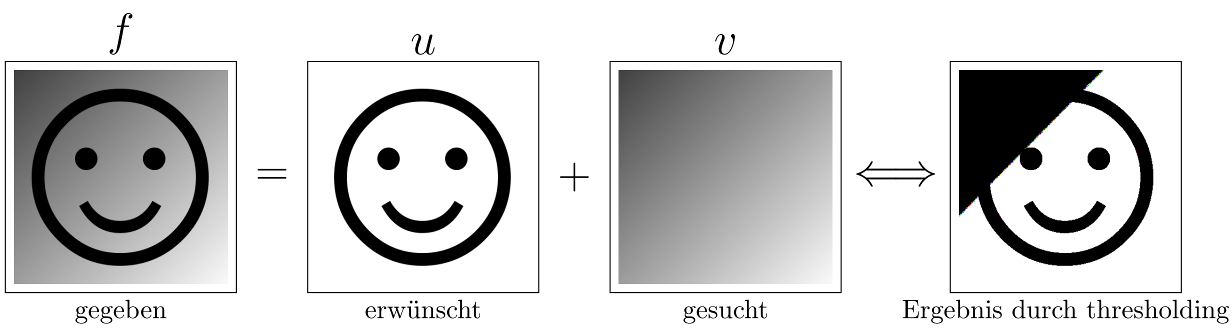
- Entrauschen $\nearrow$ 5.7
- Farbraum optimal ausnutzen $\nearrow$ 3.2
- Beleuchtungsausgleich

Dieser Beleuchtungsausgleich wurde nohc nicht vorher besprochen, das Problem:



Anstatt eines „geraden" Bildes ist das Bild, etwa durch Beleuchtung, „gekippt" und der Ansatz mittels Histogramthresholding würde nicht das gewünschte Ergebniss erzielen.

In 2D könnte dies so aus sehen.



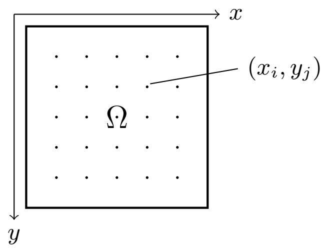
Rechts ist das Bild, das durch das Beschreibene Histogramm thresholding dargestellt wurde zu sehen. Methoden um durch preprocessing den gesuchten Gradienten zu entfernen werden im folgenden beschrieben.

9.1 Beleuchtungsausgleich

Einfachster Fall: v konstant. In diesem Fall kann man etwa eine „Leeraufnahme“ machen, $v = f$ setzen und dieses v in allen folgenden Aufnahmen subtrahieren.

Normalfall: v ändert sich bei Jeder Aufnahme. Hierbei gibt es mehrere Ansätze:

a) Lineare Regression :



Wir Unterstellen das der Verlauf affin-linear ist, d.h.:

$$v(x, y) = ax + by + c$$

Diese Parameter a, b, c gilt es nun zu schätzen.

Dazu soll gelten

$$\forall (x, y) \in \Omega : ax + by + c \approx f(x, y)$$

um dieses zu erfüllen wird eine Stichprobe von endlich vielen Punkten (x_i, y_i) aus Ω gewählt und durch diese ein Gleichungs System gebildet.

$$\begin{gathered} ax_1 + by_1 + c \approx f(x_1, y_1) \\ \vdots \\ ax_n + by_n + c \approx f(x_n, y_n) \end{gathered}$$

In Matrix form ergibt sich:

$$\underbrace{\begin{pmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} a \\ b \\ c \end{pmatrix}}_{w} \approx \underbrace{\begin{pmatrix} f(x_1, y_1 \\ \vdots \\ \vdots \\ \vdots \\ f(x_n, y_n) \end{pmatrix}}_{z}$$

Die optimale Lösung dieses Problems kann über die Normalengleichung berechnet werden.

$$A^T A w = A^T z$$

Dauraus erhält man w, somit auch a, b, c und schlussendlich $v \Rightarrow u = f - v$.
Anschließend wird noch ein Histogram stretching durchgeführt um das finale Bild zu erhalten.

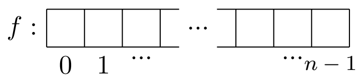
b) Polynomiale Regression : Ähnlich zur linearen Regression nun wird hierbei keine affin-lineare Funktion, sondern ein polynom genutzt. Für ein Polynom zwieten gerades kann etwa die Funktion

$$v(x, y) = ax^2 by^2 cxy + dx + ey + f$$

gewählt werden. Wieder entsteht ein Gleichungssytem:

$$\underbrace{\begin{pmatrix} x_1^2 & y_1^2 & x_1y_1 & x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^2 & y_n^2 & x_ny_n & x_n & y_n & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix}}_{w} \approx \underbrace{\begin{pmatrix} f(x_1, y_1 \\ \vdots \\ \vdots \\ \vdots \\ f(x_n, y_n) \end{pmatrix}}_{z}$$

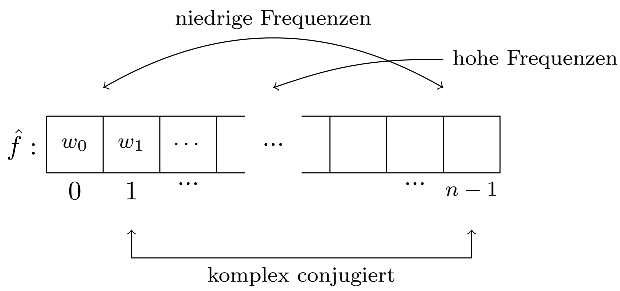
c) <u>Trigonometrisches Polynom</u> Hierbei wird v in den niedrigfrequenten Anteilen von f gesucht.



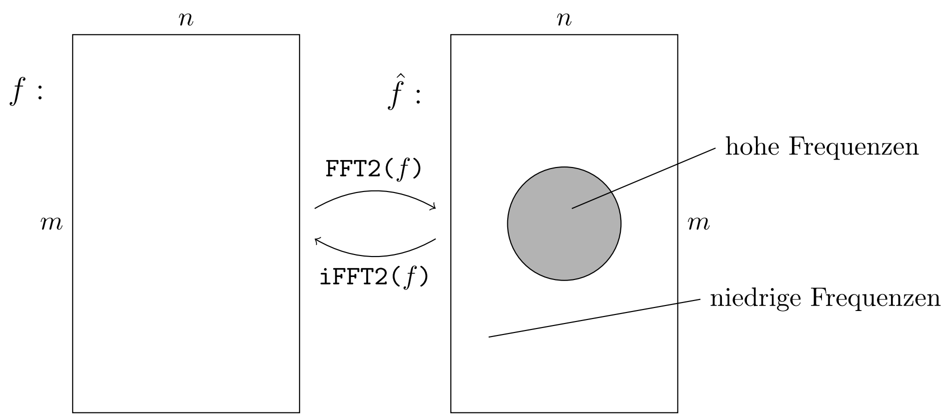
Es ergibt sich $\hat{f}$:

$$\hat{f}_k = \sum_{m=0}^{n-1} f_k (\underbrace{e^{-i2\pi \frac{k}{n}}}_{w_k})^m \quad , \ k = 0, 1, ..., n-1$$

Mittels der FFT (Fast Fourier Transformation) ergibt sich $\hat{f}$ zu:



Durch entfernen dieser niedrigen Frequenzen ergibt sich u. Ähnliches funktioniert auch in 2D:



9.2 Thresholding als Variationsproblem

geg: Bild $u : \Omega \to F = [0, 1]$ und Schwellenwert $t \in (0, 1)$

$$\Rightarrow: \quad \left.\begin{array}{l} \Omega_0 = \{x \in \Omega : u(x) \leq t\} \to \text{schwarz} \\ \Omega_1 = \{x \in \Omega : u(x) > t\} \to \text{weiß} \end{array}\right\} \text{soll in ein Variationsproblem umformuliert werden.}$$

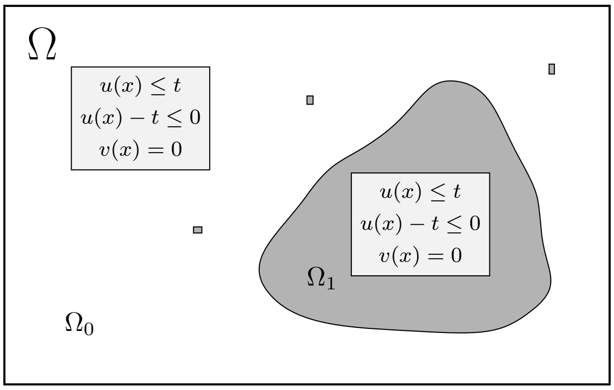
Setze dazu:

$$J(v) := -\int_\Omega (u(x) - t)\mathbb{C}v(x)dx \to \; min \tag{9.1}$$

mit $v \in U := \big\{v \in \Omega \to \{0,1\} \;\; (\text{oder } [0,1])\big\}$, dies ist jedoch kein Vektorraum.
Die Lösung dieses Problems ist offenbar

$$v = \chi_{\Omega_1}, \;\text{d.h.} \begin{cases} 0, & x \in \Omega_0 \\ 1, & x \in \Omega_1 \end{cases}$$

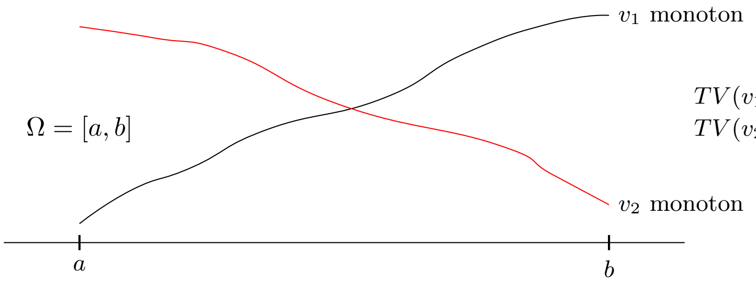
illustiert hier:



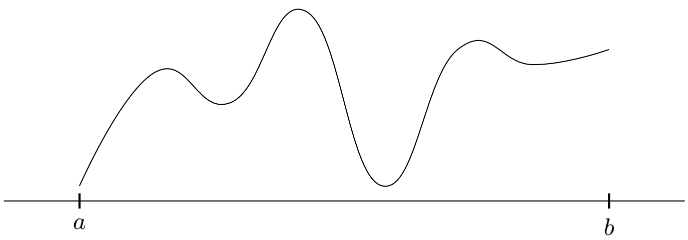
Diese Herangehensweise ist jedoch kompliziert, um sie zu begründen betrachten wir die Flecken die neben der großen Masse in der obing Illustration zu sehen sind und etwa durch Rauschen entstanden seien könnten. Durch Verallgemeinerung des in 9.1 gegebenen Funktionals können wir die Zerlegung von Ω in Ω_0 und Ω_1 weniger Anfällig gegenüber Rauschen und sonstigen kleinen Strukturen machen.
Dazu sei

$$TV(v) = \int_\Omega |\nabla v(x)|\, dx \tag{9.2}$$

die sogennate <u>Totalvariation</u> einer Funktion auf Ω.
Illustration in 1D:



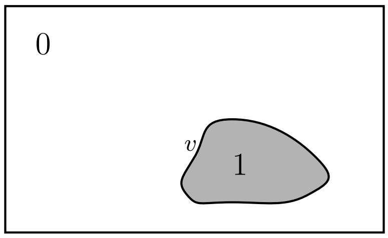
$TV(v_1) = v_1(b) - v_1(a)$
$TV(v_2) = |v_2(b) - v_2(a)|$



Hier ist die Totalvariation die Summe aller Höhenanstiege, wobei sowohl ab- und anstiege positiv gezählt werden.

In Abschnitt 11.2 werden wir weiterhin sehen, dass die Totalvariation sich auch auf nicht stetigen Funktionen etwa $\chi_{(0,\infty)}$ berechnen lässt, obwohl für diese der Gradient nicht definiert ist. Für die eben genannte Funktion beträgt die Totalvariationetwa 1.

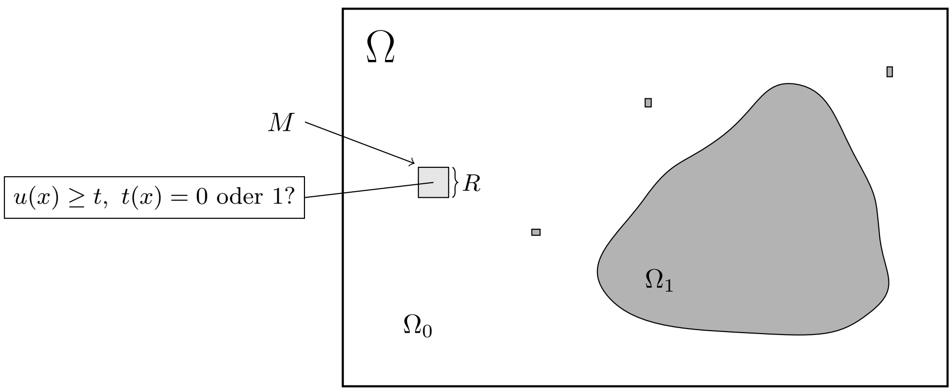
Nun in 2D:



$TV(v)$ ist hier die Länge der Kante die Ω_1 von Ω_0 trennt.
Die Idee ist nun $TV(v)$ als Strafterm zu 9.1 hinzu zu addieren,

$$\widetilde{J}(v) := \underbrace{- \int_\Omega (u(x) - t) \mathbb{C} v(x) dx}_{J(x)} + \lambda \mathbb{C} TV(v) \to \ min \tag{9.3}$$

wobei $v \in U$ ist. Der Effekt dieser Herangehensweise wird nun illustriert.



$$\left.\begin{array}{llccc} \text{Falls:} & v|_M = 0: & a & + & \lambda\mathbb{C}b \\ \text{Falls:} & v|_M = 1: & a - dR^2 & + & \lambda\mathbb{C}(b + 4R) \end{array}\right\} \text{Also } M \to 0 \iff 4\lambda - dR < 0 \iff R > \tfrac{4\lambda}{d}$$

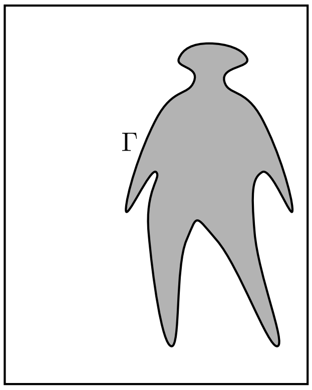
Das heißt das kleine Segment M wird durch 9.3 „erkannt“ falls seine Kantenlänge $R > \frac{4\lambda}{d}$ ist. Somit können die Abmessung der kleinsten zu segmentierenden Strukturen über λ gesteuert werden.

9.3 Segmentierung nach Mumford und Shah

Wieder: Variationsrechnung

Diesmal: Ohne Vorkenntnis des thresholds

Idee: Bild zerlegen in „glatte“ Teile getrennt durch Sprünge an deren Rändern.



geg.: Bild u

ges.: Stückweise glattes Bild v mit Randkurve Γ

1. Wunsch: $u \approx v$ auf ganz Ω

2. Wunsch: ∇v klein auf $\Omega\backslash\Gamma$

$$\Rightarrow J(v,\Gamma) := \underbrace{||u-v||^2_{2,\Omega}}_{\text{1. Wunsch}} + \lambda \underbrace{||\nabla v||^2_{2,\Omega\backslash\Gamma}}_{\text{2. Wunsch}} \rightarrow \ min$$

Wie im letzten Abschnitt 9.2 soll nun noch die Segmentierung sehr kleiner Strukturen vermieden werden, indem man zu J einen entsprechenden Strafterm addiert.

$$\widetilde{J}(v,\Gamma) := ||u-v||^2_{2,\Gamma} + \lambda\, ||\nabla v||^2_{2,\Omega\backslash\Gamma} + \mu \text{ Länge}(\Gamma) \rightarrow \ min$$

Dieses wird Mumford-Shah-Funktional (1989) genannt.

λ bestimmt die „Flachheit“ von v auf $\Omega\backslash\Gamma$

μ ist proportional zur Größe der kleinsten zu segmentierenden Struktur.

Die numerische Lösung dieses Problemes ist jedoch sehr kompliziert, da neben v auch die Kurve Γ variiert wird, desßhalb existert eine „vereinfachte“ Version das dieses Problem approximiert.
Mumford-Shah (1989):

$$\widetilde{J}(v,\Gamma) := ||u-v||^2_{2,\Gamma} + \lambda\, ||\nabla v||^2_{2,\Omega\backslash\Gamma} + \mu \text{ Länge}(\Gamma)$$

Strekalovskiy & Cremers (2014):

$$\widetilde{J}(v,\Gamma) := \int_\Omega \left[|u(x)-v(x)|^2 + \min(\lambda\, |\nabla c(x)|^2, \mu) \right] dx$$

Die minimierung und μ simulieren den Sprung an der Randkurve Γ.