

1. Grundlagen Digitaler Bildverarbeitung

Prof. Dr.-Ing. Thomas Schultz

URL: <http://cg.cs.uni-bonn.de/schultz/>

E-Mail: schultz@cs.uni-bonn.de

Büro: Friedrich-Hirzebruch-Allee 6, Raum 2.117

14./21. Oktober 2024

Kurze Vorstellung meiner Person



Informatik-Doktorand (2006-09)

MPI Informatik, Saarbrücken

Postdoctoral Fellow (2009-11)

University of Chicago, IL, USA



Wissenschaftl. MA (2011-14)

MPI Intelligente Systeme, Tübingen

Juniorprofessor (2013-2017)

Informatik, Universität Bonn

Professor (seit 2017)

B-IT und Informatik, Universität Bonn

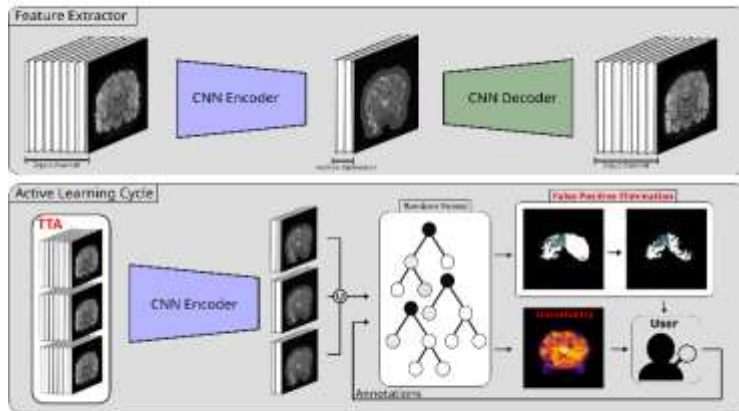


Kurze Vorstellung unserer Gruppe

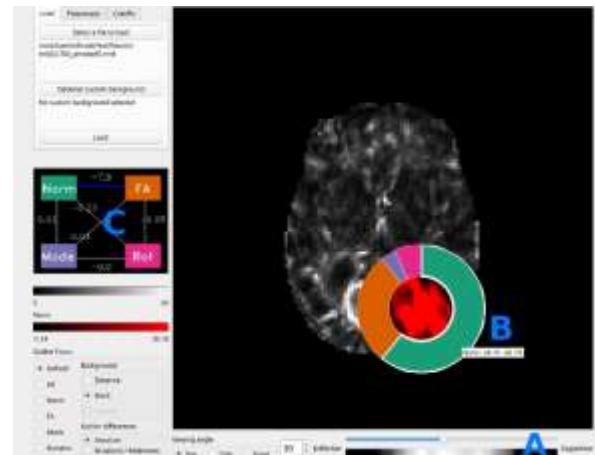
- **Arbeitsgebiet: Visualisierung und Medizinische Bildanalyse**
 - Aktuell 1 Postdoktorand, 3 aktive Doktorand*innen, 3 HiWis, 6 aktive Abschlussarbeiten (MSc/BSc)



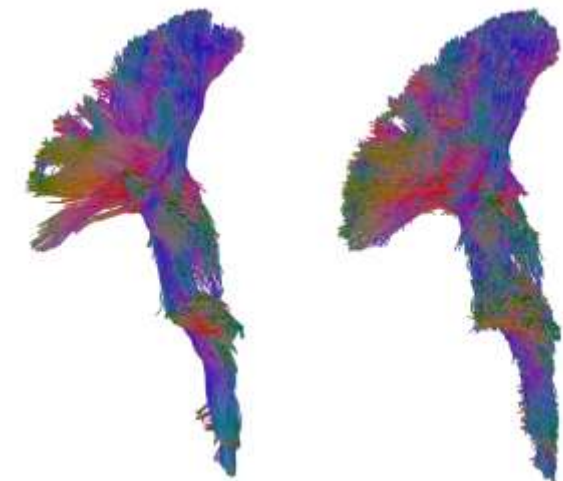
Publikationen aus Abschlussarbeiten heraus



[Lennartz et al., UNSURE 2024]



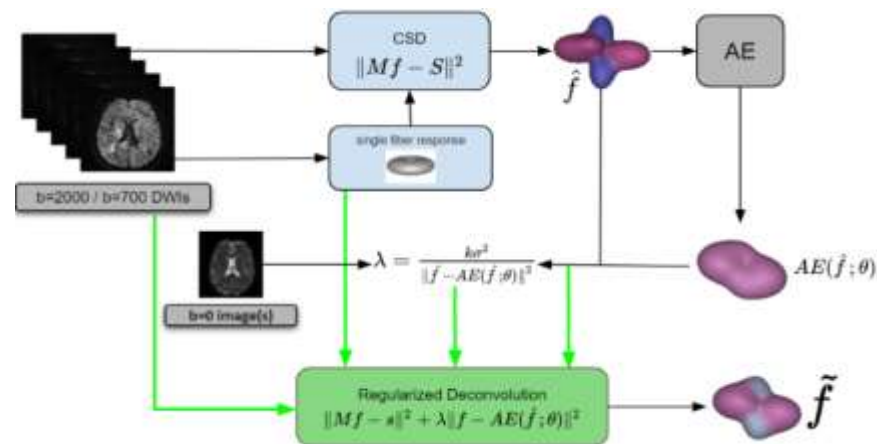
[Bareth et al., EuroVis SP 2023]



[Grün et al., VCBM 2021]



[Mueller et al., OMIA 2020]



[Patel et al., MICCAI 2018]

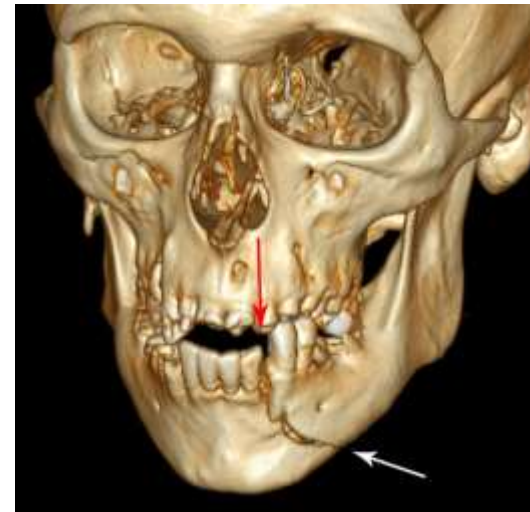


[Soundararajan et al., EuroVis 2015]

1.1 Geplante Inhalte dieser Vorlesung

Motivation: Bildgebung in der Medizin

- **Bildgebende Verfahren** liefern der Medizin ansonsten nicht zugängliche Informationen zu *Struktur* und *Funktion*, insbesondere im Körperinneren
- **Algorithmen** sind für die Rekonstruktion, (Vor-)verarbeitung und Visualisierung unerlässlich
 - *Beispiel*: Computertomographie



3D-Rekonstruktion
aus CT-Scan



Fusioniertes
PET/CT

Motivation: Bildgestützte Diagnostik

In vielen Bereichen der **medizinischen Diagnostik** ist die Bildgebung Teil der alltäglichen Routine.



Pädiatrische Echokardiografie



Zahnmedizinische Panoramaröntgenaufnahme

Motivation: Bildgestützte Intervention

Bildgebung unterstützt häufig die **Planung** oder sogar **intraoperativ** die **Durchführung** von Interventionen wie chirurgischen Eingriffen oder Strahlentherapien



Motivation: Autonome Bildgestützte Diagnostik

2018 wurde mit IDx-DR das erste autonome diagnostische System in den USA zugelassen (Erkennung diabetischer Retinopathie)

- 2023: Mehr als 200 KI-basierte Produkte in der Radiologie

<https://www.eyediagnostics.co/>



Autonomous AI
that instantly
detects disease

Increase patient access to early
disease detection.

[Learn More](#)

Motivation: Bildgestützte Gesundheitsforschung

Viele bildgebende Verfahren sind sicher genug um damit freiwillige Teilnehmer von **Gesundheitsstudien** zu untersuchen

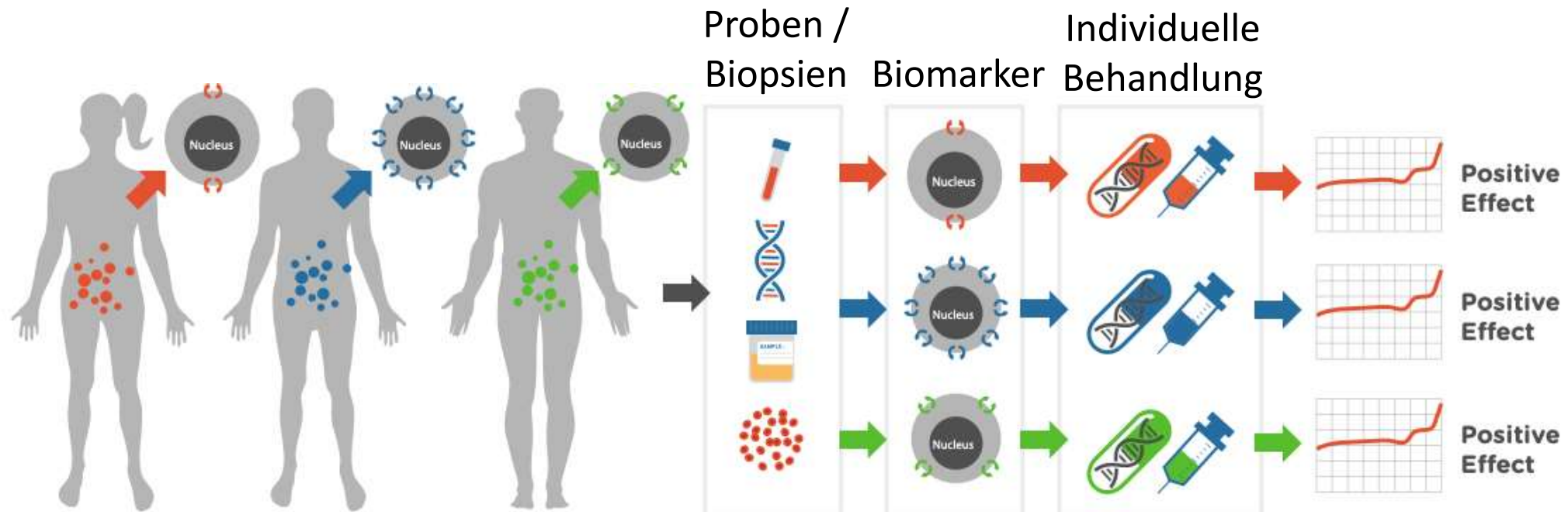
- „**nichtinvasive**“ Bildgebung
- *Beispiel:* Rheinland-Studie zu gesundem Altern am DZNE in Bonn



Ausblick: Präzisionsmedizin

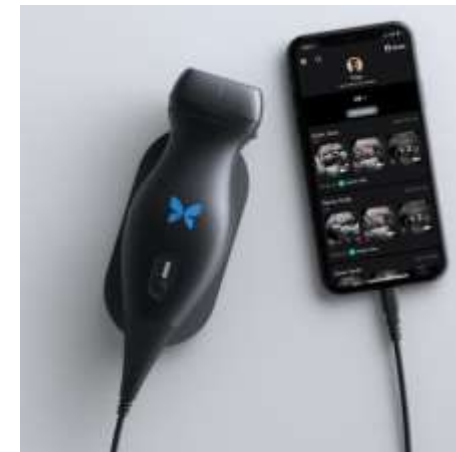
Als **Präzisionsmedizin** bezeichnet man eine stärkere Anpassung medizinischer Behandlungen an den individuellen Patienten

- *Beispiel*: Therapieentscheidungen in der Tumormedizin
- Aktuell primär auf Grundlage von Genanalysen
- *Hoffnung*: Durch Bildgebung Phänotypen stärker berücksichtigen

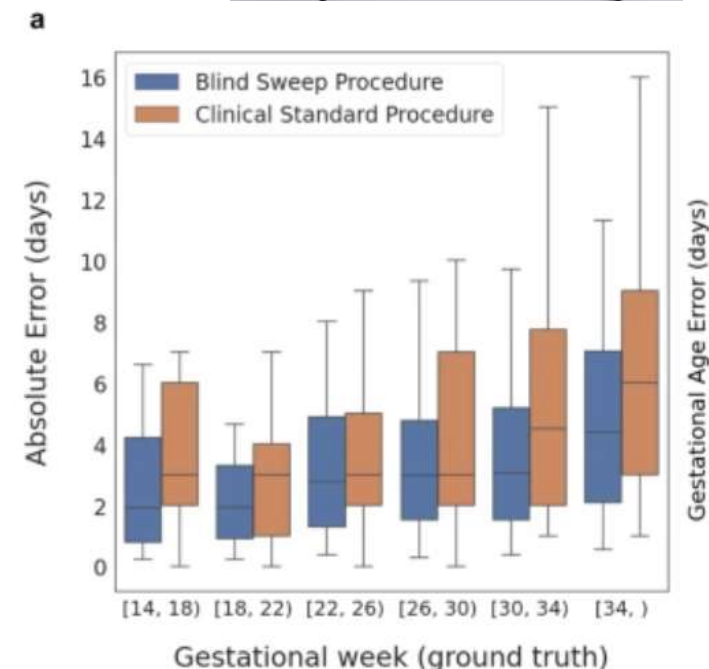


Ausblick: Medizinische Versorgung im Globalen Süden

Eine automatische Befundung mittels tragbaren Sensoren und Computern könnte die Lage in medizinisch unterversorgten Gegenden verbessern



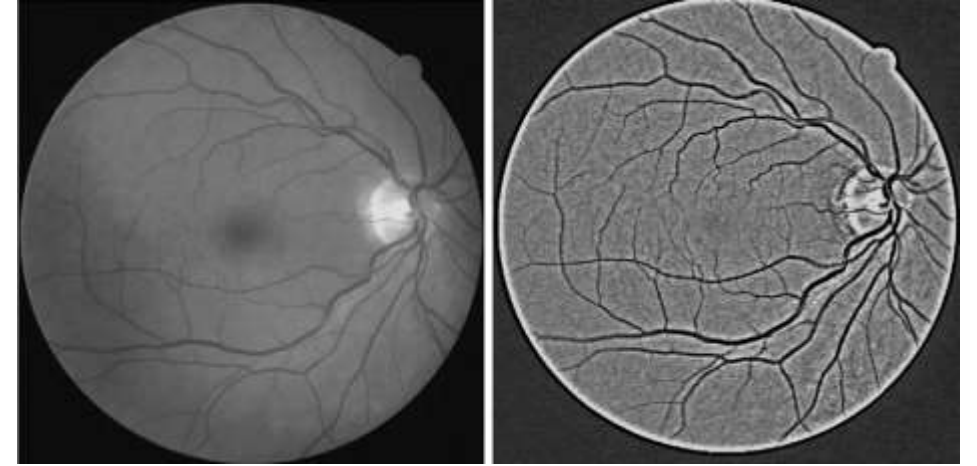
<https://gm-medical.com/>



1. Grundlagen Digitaler Bildverarbeitung

Vorverarbeitung von Bildern zur

- Kontrastverbesserung
- Rauschunterdrückung
- Schärfung
- Kantenerkennung



Kontrastverstärkung in Retina-Aufnahme

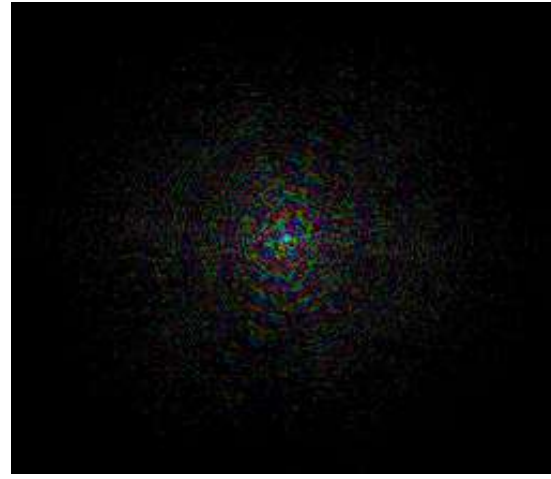
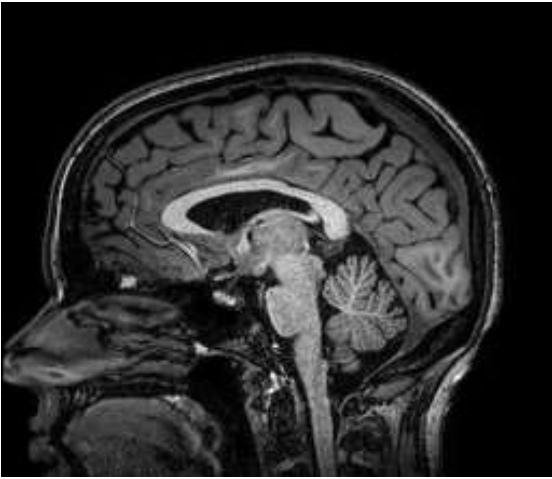


Zunehmende Glättung eines MRT-Scans

2. Grundlagen der Signaltheorie

Verständnis von **Theorie** und **praktischen Konsequenzen** von

- Abtasttheorem und Aliasing
- Fourier-Transformation
- Vergrößerung / Verkleinerung der Bildauflösung



MRT-Bild und seine Fourier-Transformation



Fehlerhaft / sinnvoll reduzierte Bildauflösung ¹⁴

3. Bildgebende Verfahren

Funktionsweise und Bildcharakteristika von

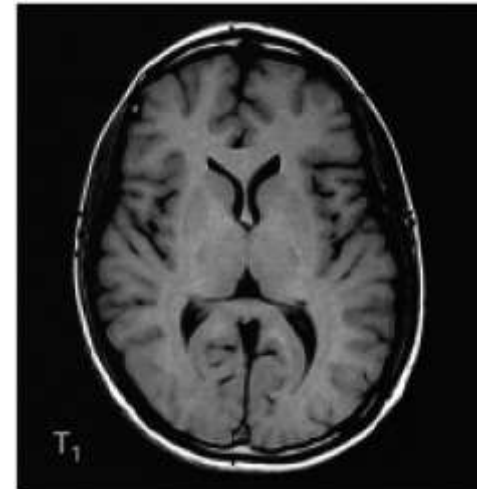
- Röntgenbildgebung
- Computertomographie
- Emissionstomographie
- Magnetresonanztomographie
- Ultraschall
- (Optischer Kohärenztomographie)



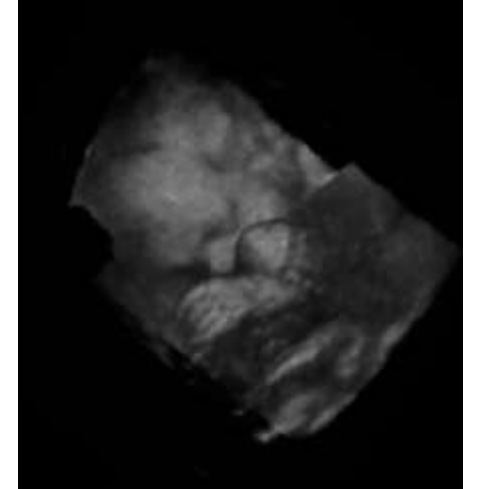
Röntgenbild



CT-Scanner



MRT-Scan

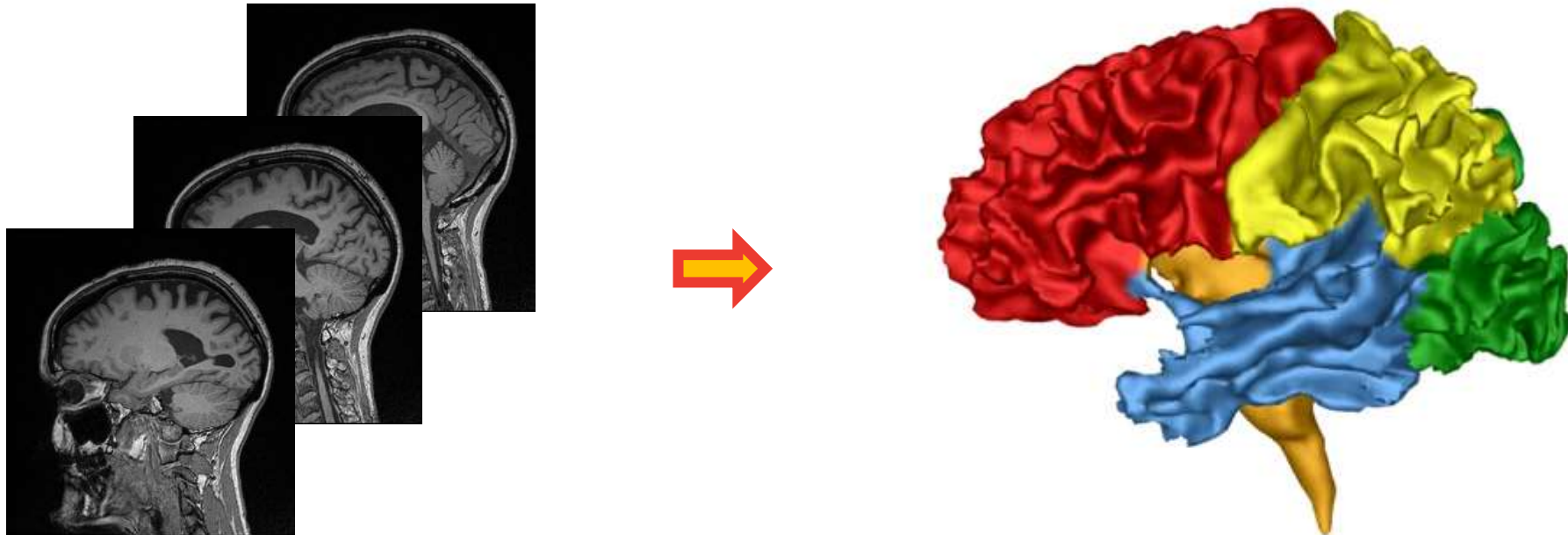


3D-Ultraschall

4. Bildsegmentierung

Abgrenzung von Bildobjekten (Organe, Tumore, etc.)

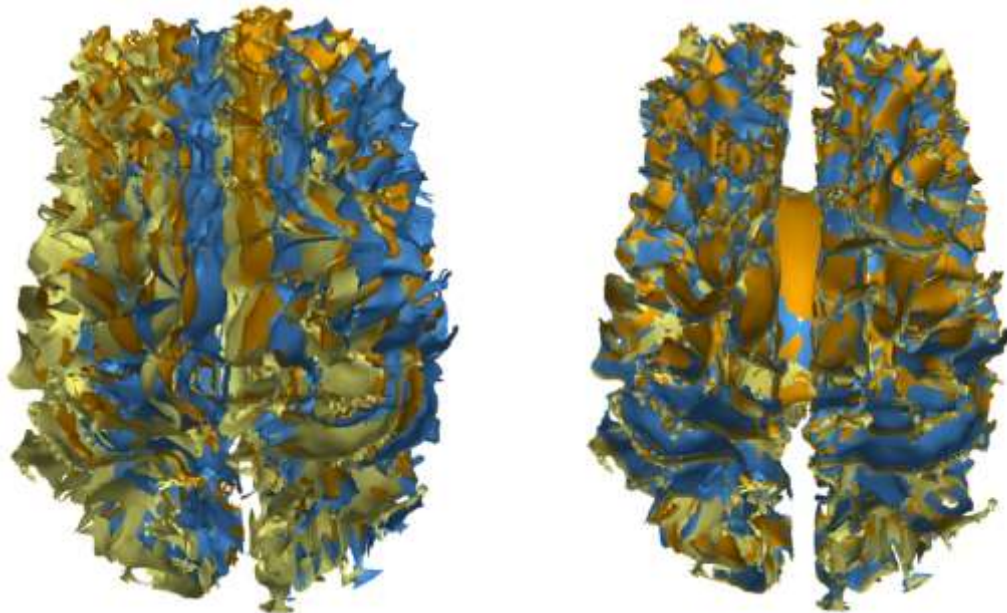
- Schwellenwert-Verfahren
- Aktive Konturen / Deformierbare Modelle
- Formmodelle



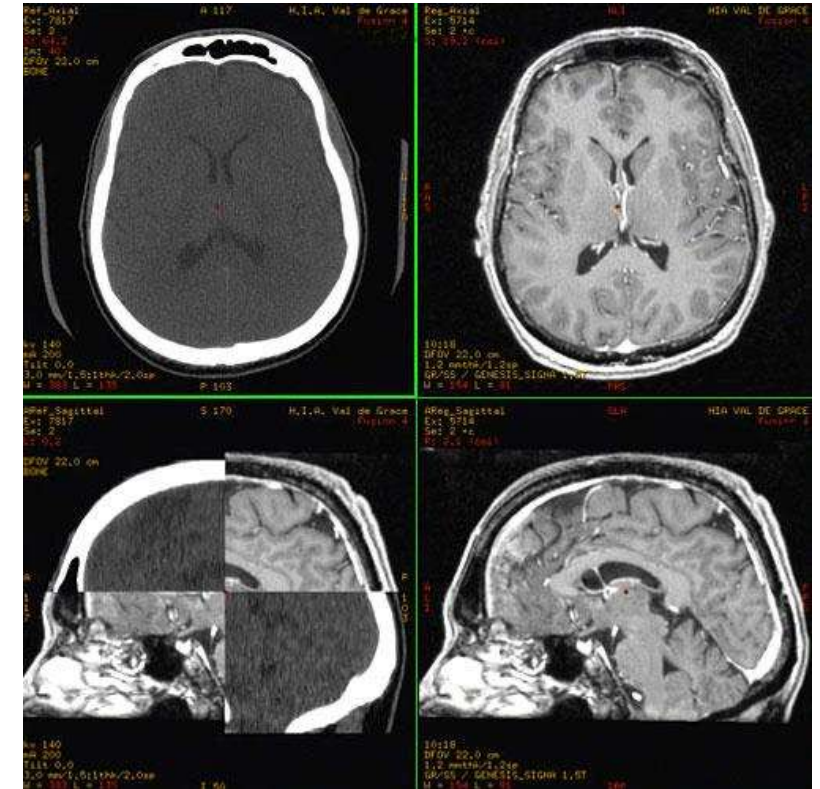
5. Bildregistrierung

Bilder in Korrespondenz bringen

- Bildtransformationen
- Kostenfunktionen
- Optimierungsalgorithmen



Registrierung wiederholter Scans

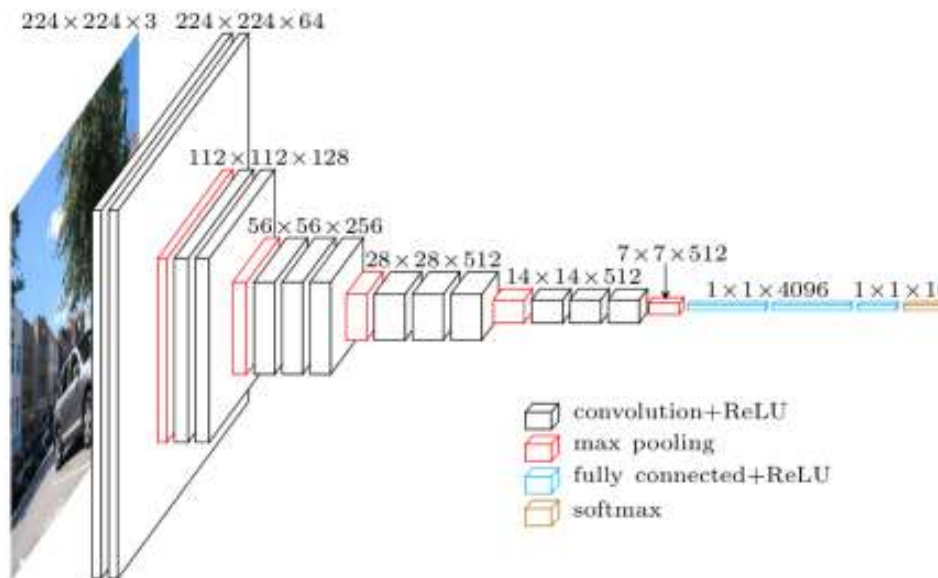


Registrierung von CT/MRT

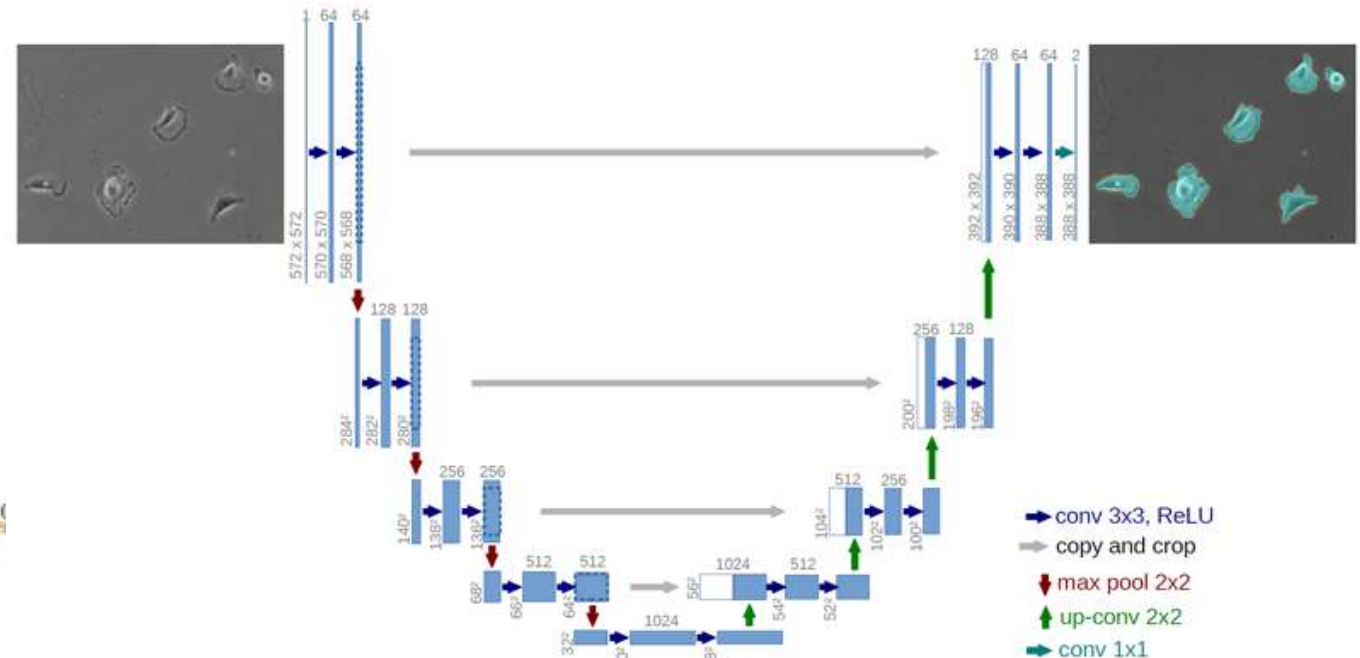
6. Bildanalyse mittels Deep Learning

Funktionsweise und Anwendungsbeispiele neuronaler Netze

- Faltungsnetzwerke (CNNs)
- Bildklassifizierung
- Bildsegmentierung



Bildklassifizierung [Simonyan/Zisserman, 2014]



Bildsegmentierung

[Ronneberger et al., MICCAI 2015]

1.2 Organisatorisches

Vorlesungen und eCampus

- **Vorlesungen** finden jeden Montag um 14 c.t. im HSZ HS3 statt
- Wir nutzen **eCampus**
 - um Ihnen Folien und ggf. weitere Materialien zur Verfügung zu stellen
 - für die Übungsabgaben
 - als öffentliches Forum für Fragen und Diskussionen
 - falls nötig, um ZOOM-Links / Vorlesungsvideos zu verbreiten
 - Ich versuche die Vorlesung wann immer möglich in Präsenz zu halten!
 - Bitte tragen Sie sich als Kursmitglied ein um an den Übungen teilzunehmen und ggf. wichtige Ankündigungen zu erhalten

Übungsbetrieb

- **Übungen**

- Voraussetzung um zur Klausur zugelassen zu werden
 - *Kriterium:* $\geq 50\%$ der Punkte insgesamt, mindestens eine Präsentation
 - Bilden Sie bitte möglichst **Gruppen von drei Studierenden**, jedoch nicht mehr.
 - Gruppieren Sie sich im Laufe des Semesters falls nötig gern um.
- Es wird 11 reguläre Übungsblätter geben
 - Plus Probeklausur am Ende, die zur Zulassung *nicht* mitzählt
- Übungen werden jeden Montag veröffentlicht, eine Woche später eingereicht, am Donnerstag besprochen
- Globalübung **Donnerstags um 14 c.t.** im HSZ, HS 4
 - Diese Woche: Sprechstunde für eventuelle Rückfragen

Hinweise zu den Übungen

- **Praktische Erfahrung mit Implementierung und Anwendung** relevanter Algorithmen zählt zu unseren zentralen Lernzielen
 - Sie profitieren davon, obwohl die Klausur keine Programmieraufgaben im engeren Sinne enthält
 - Diese Fähigkeiten benötigen Sie spätestens für eine Projektgruppe oder BSc-Arbeit in verwandten Bereichen
- **(Online-)Recherchen sind Teil des Lernprozesses**
 - und kein Anzeichen dafür, dass die Aufgabe falsch gestellt wäre
- Wir tolerieren keine **Plagiate!**
 - Für Lösungen, die in sehr ähnlicher Form von mehreren Gruppen eingereicht wurden oder offensichtlich kopiert sind, können wir keine Punkte vergeben.

Kreditpunkte

- Am Semesterende wird es eine **Klausur** geben
 - Vorläufige Planung: 10. Februar und 2. April
 - Genaue Daten und Zeiten gebe ich bekannt, sobald sie feststehen
- Die bestandene Klausur ist im Wahlpflichtbereich des **BSc Informatik** 6 ECTS wert
 - Haben wir Teilnehmer*innen aus anderen Studiengängen?

1.3 Punktweise Bildtransformationen und Histogramme

Wie können wir Bilder (mathematisch) beschreiben?

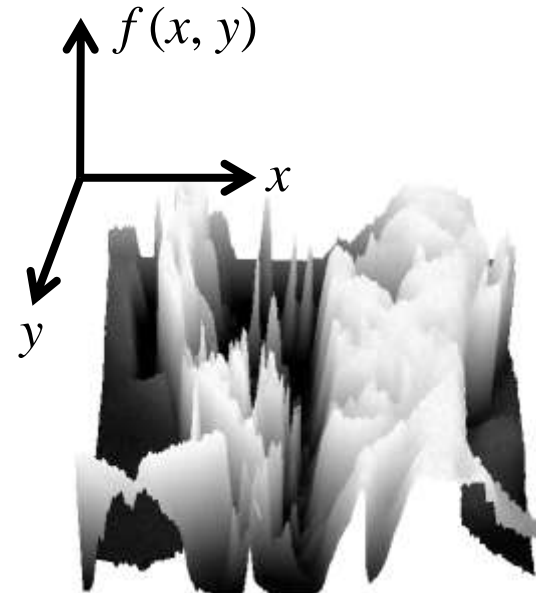


Bilder als Funktionen

- Bilder können als **Funktion** (2D-Signal) beschrieben werden:

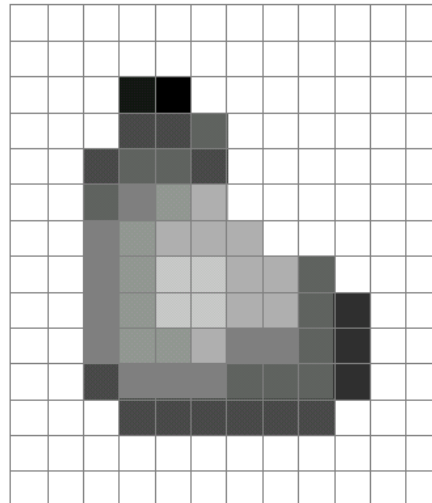
$$f: \mathbb{R}^2 \rightarrow \mathbb{R}$$

- $f(x,y)$ ist die **Intensität** des Bildes an Position (x,y)
- $\mathbf{f}(x,y)$ kann ein Vektor sein, z.B. im Fall von Farbbildern
 - *Beispiel:* RGB = Intensitäten der rot/grün/blau-Kanäle



Bilder als Matrizen

- **Digitale Bilder** sind eine Diskretisierung der Intensitäts-Funktion
 - **Abtastung (Sampling)** = räumliche Diskretisierung in **Pixel**
 - **Quantisierung** = Diskretisierung der Werte (z.B. Rundung auf ganze Zahlen)
 - *Beispiel:* 8bit “Tiefe”, 0 = schwarz, 255 = weiß
 - Farbbilder haben mehrere **Kanäle**



=

255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

Filterung von Bildern

- Ein **Bildfilter** $T: f \mapsto g$ ist ein Operator, der ein Eingabebild f auf ein Ausgabebild g abbildet
- *Beispiele:*



$$g(x, y) = f(x, y) + 20$$



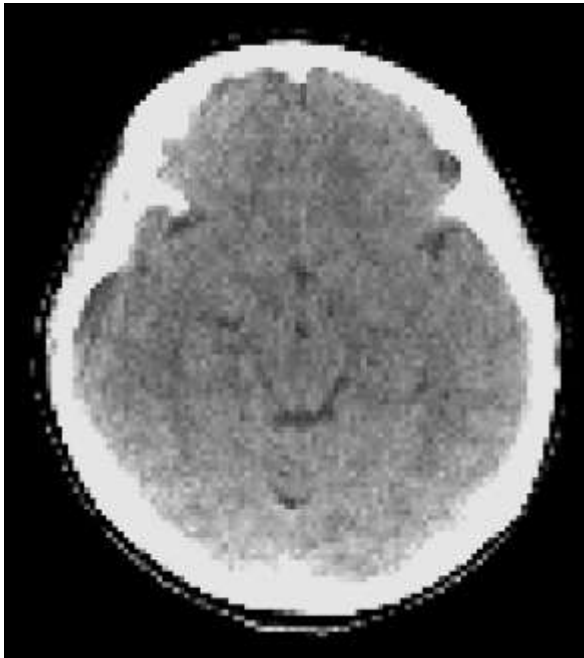
$$g(x, y) = f(-x, y)$$

- Ein **punkt-** oder **pixelweiser Operator** lässt sich schreiben als
$$g(x, y) = t(f(x, y))$$

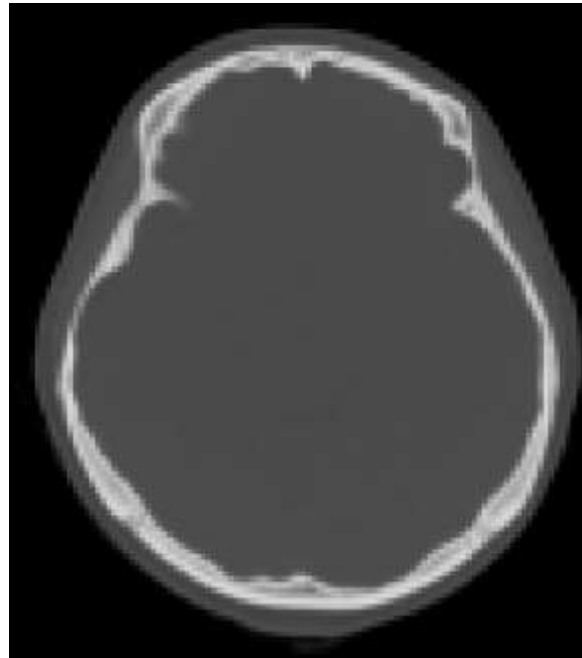
Fensterung

Als **Fensterung** bezeichnet man eine Einschränkung des Wertebereichs medizinischer Bilder auf ein “Fenster”

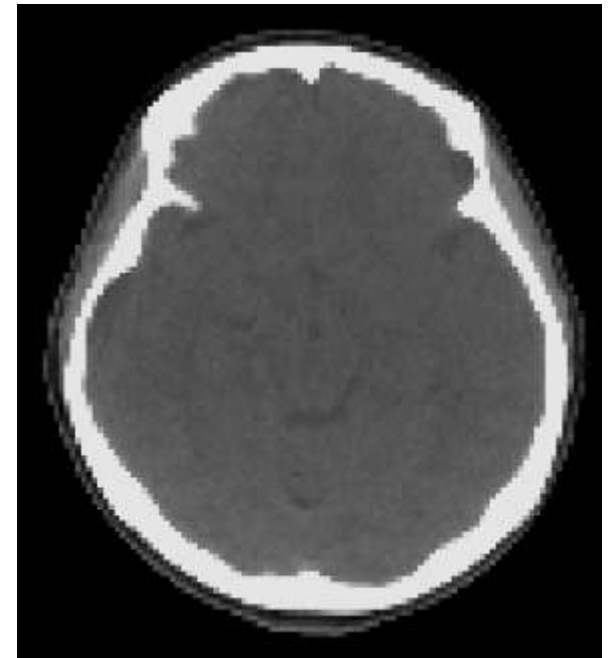
- Größere / kleinere Werte werden weiß / schwarz dargestellt
- Optimiert den Kontrast für Strukturen innerhalb des Fensters



Gehirnfenster



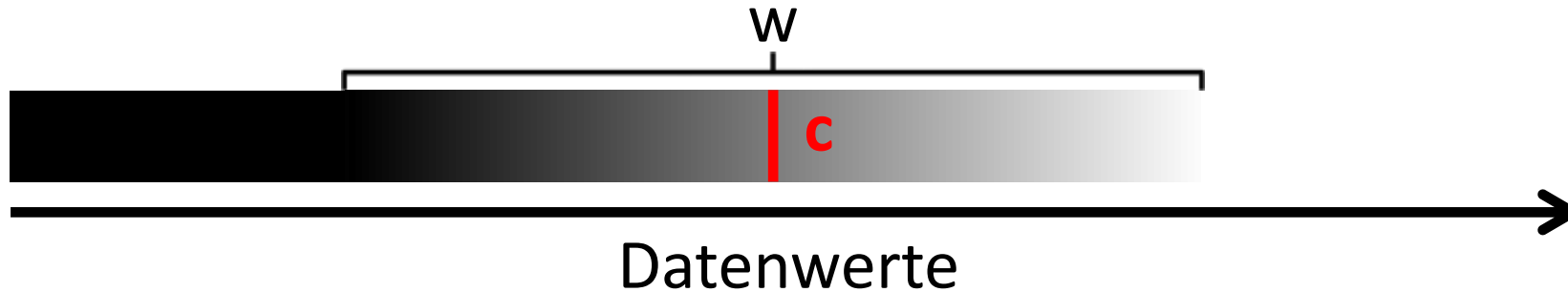
Knochenfenster



Blutungs-Fenster

Spezifikation von Fenstern

Fenster beschreibt man durch ein Zentrum c und eine Breite $w \geq 1$.



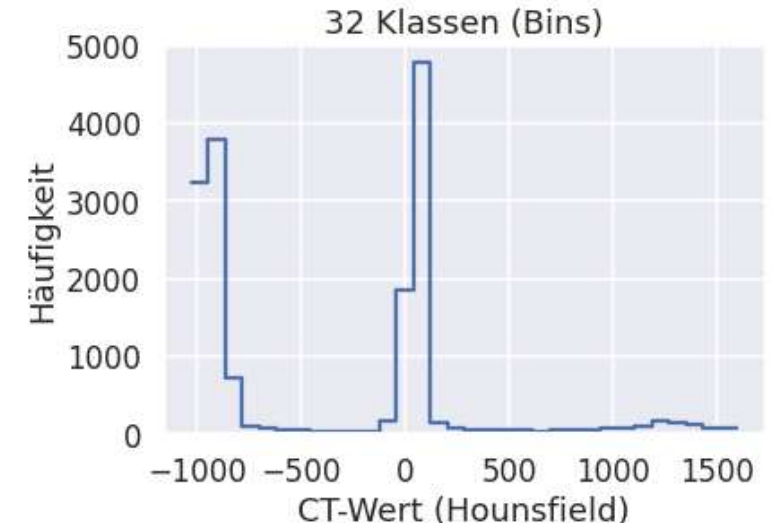
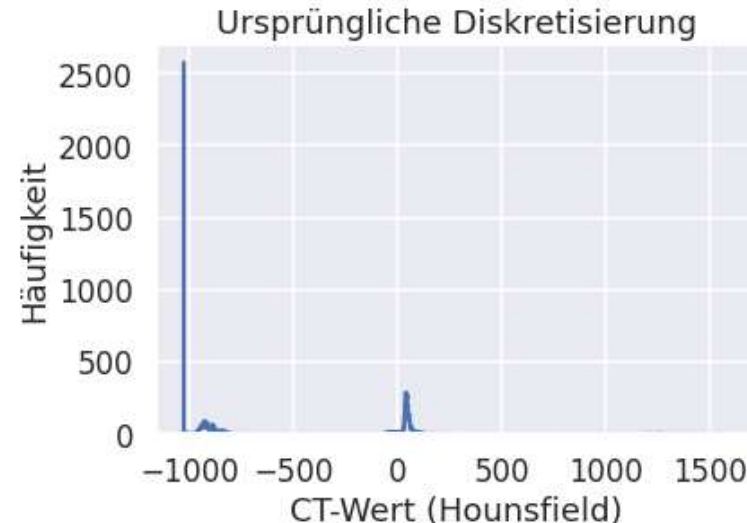
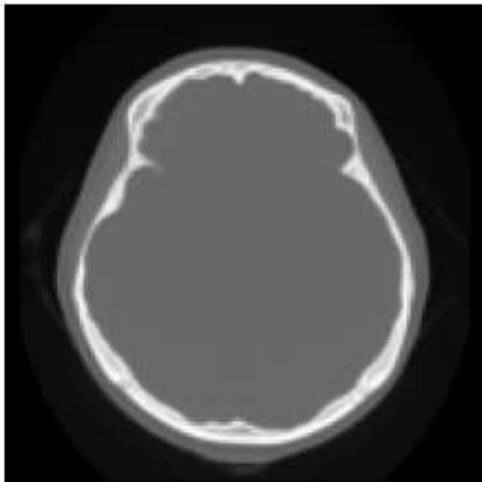
Für Datenwert x und auf $C \in [0,1]$ normierte Graustufen beschreibt der folgende Pseudocode die Fensterung:

```
if (x <= c - 0.5 - (w-1)/2) then C := 0
else if (x > c - 0.5 + (w-1)/2) then C := 1
else C := (x - (c - 0.5)) / (w-1) + 0.5
```

- Ausführung in Fließkomma-Arithmetik, Skalierung/Rundung am Ende
- $w=1$ führt zu einer Binarisierung (Schwellenwertbildung)

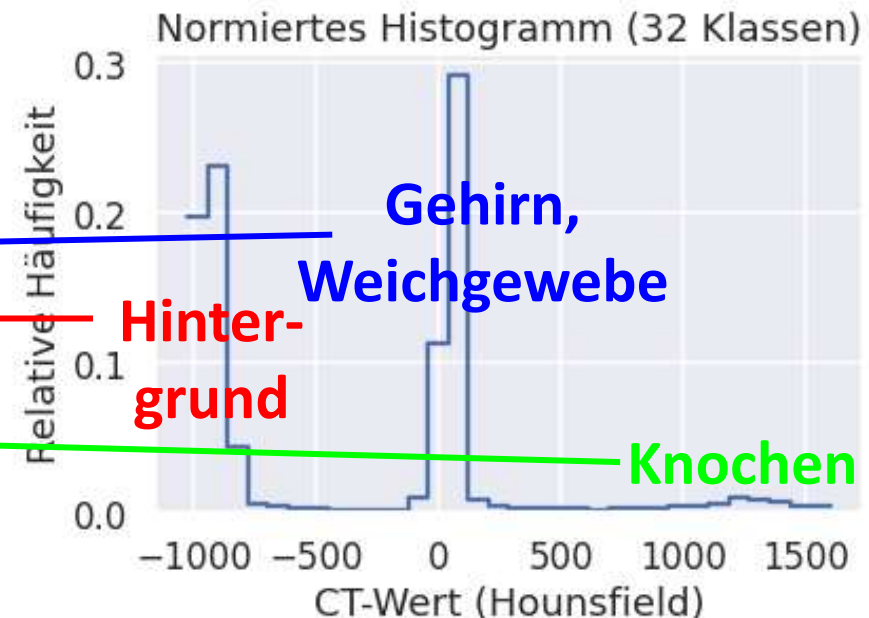
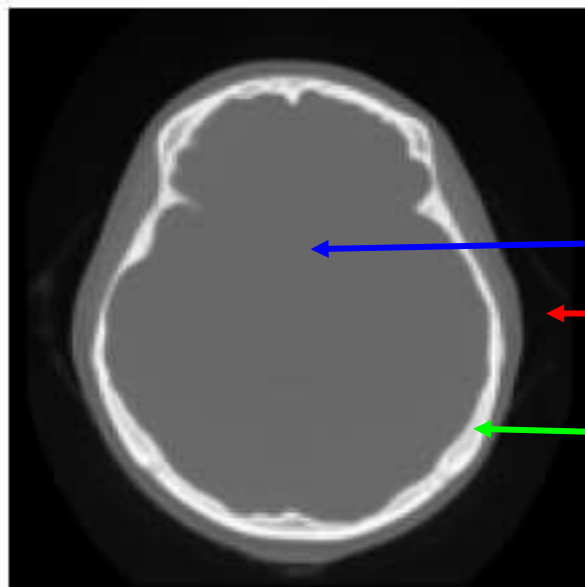
Histogramme

- **Histogramme** beschreiben die Häufigkeitsverteilung von Intensitätswerten in einem Bild
 - $H(I)$ nutzt die gegebene (ganzzahlige) Diskretisierung der Werte und zählt wie viele Pixel den Wert I haben
 - *Alternativ*: Grobere Diskretisierung („Binning“) fasst ähnliche Werte zusammen
 - Das mit der Zahl N der Pixel normierte Histogramm $H_n(I) := \frac{1}{N} H(I)$ kann man als *Wahrscheinlichkeitsverteilung* interpretieren



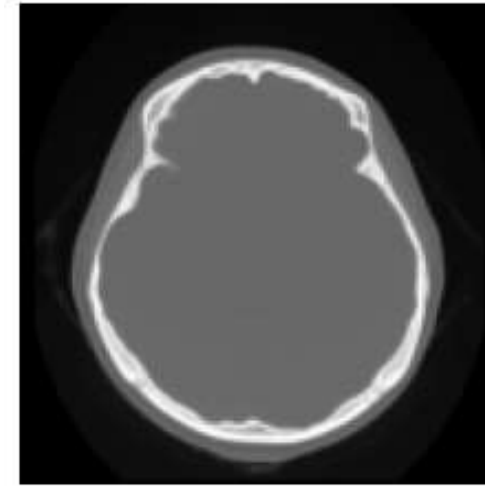
Histogramm-Analysen

- Häufig haben verschiedene Materialien oder Gewebetypen charakteristische Intensitäten, die als **Gipfel im Histogramm** erkennbar sind
 - **Histogramm-Analysen** bieten Anhaltspunkte für geeignete Fenster
 - **Täler im Histogramm** bieten sich als Schwellenwerte an

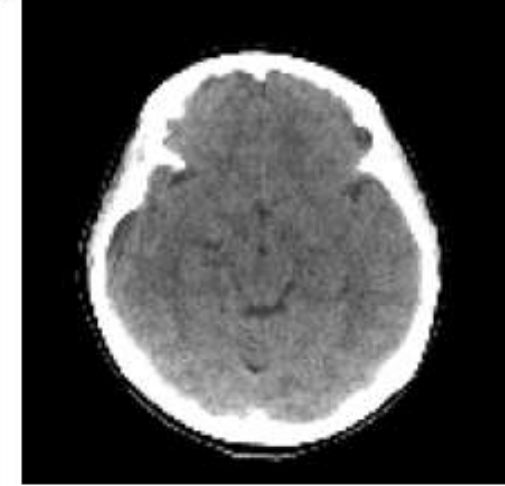


Fensterung als Histogrammtransformation

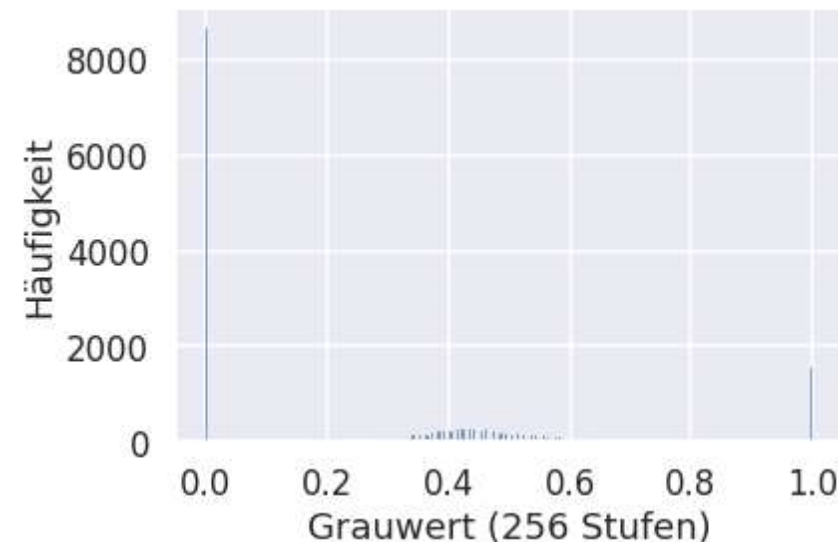
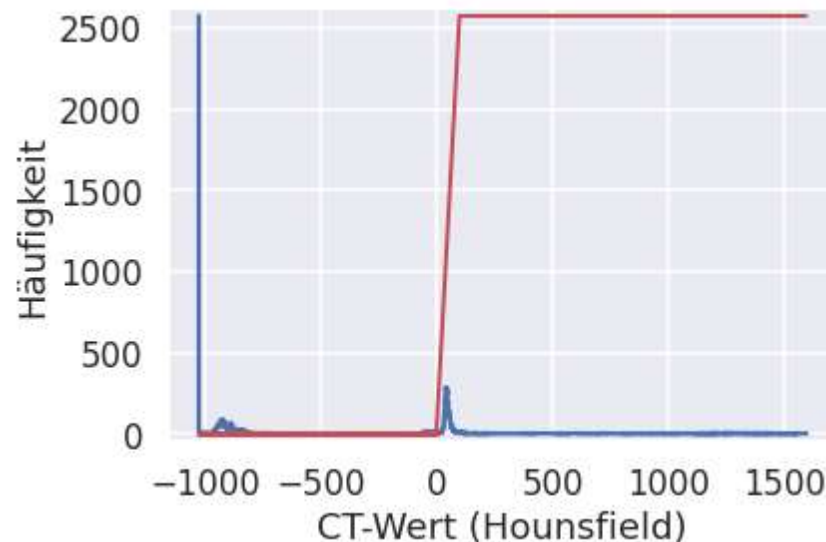
- Fensterung lässt sich durch eine stückweise lineare **Transformationskennlinie** beschreiben
 - **Begrenzt** und **spreizt** das Histogramm auf das Fenster
 - Führt aufgrund der Diskretisierung oft zu **Lücken im Histogramm**



Original

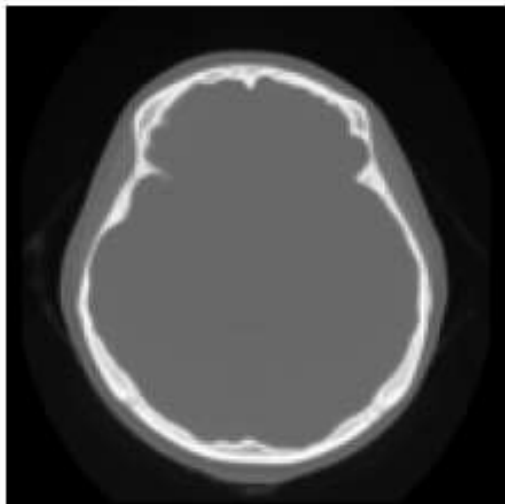


Gehirnfenster

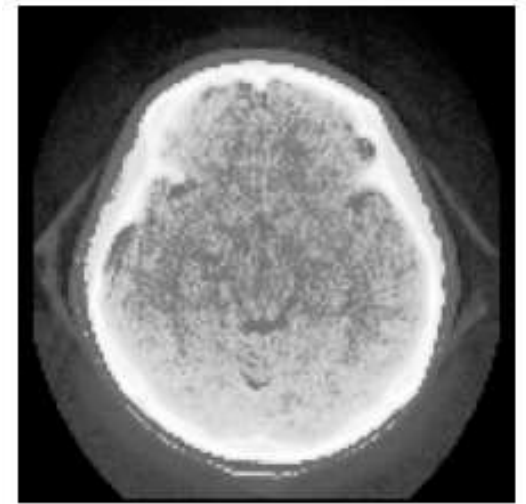
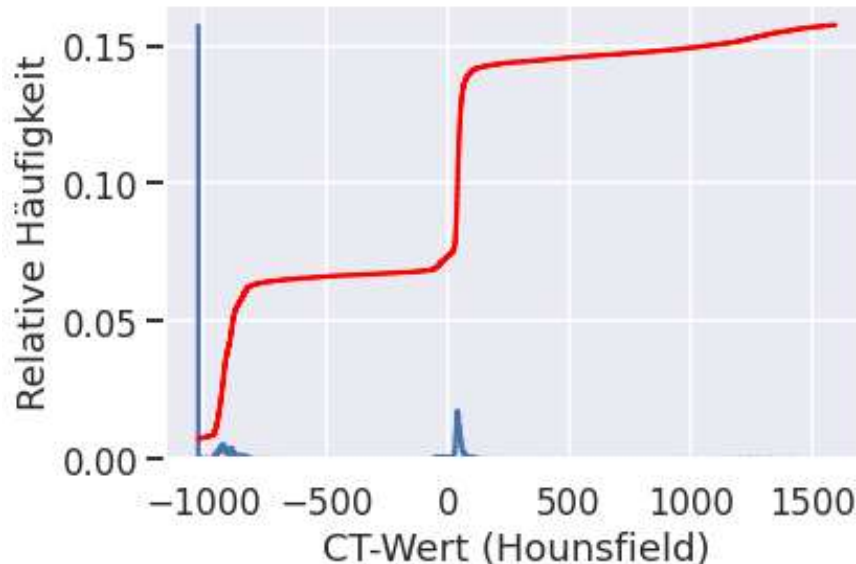


Histogrammausgleich: Definition

- Ein **Histogrammausgleich** $t_{HA}(I)$ (engl. *histogram equalization*) strebt eine Kontrastoptimierung durch Gleichverteilung aller Intensitäten an
 - Die Transformationskennlinie ergibt sich aus dem **kumulativen Histogramm** $H_k(I) := \sum_{i=0}^I H_n(i) = P(i \leq I)$
 - Bei Skalierung der Ausgabe auf $[0, I_{\max}]$: $t_{HA}(I) := I_{\max} \times H_k(I)$



Original



Histogrammausgleich

Histogrammausgleich: Theoretische Begründung

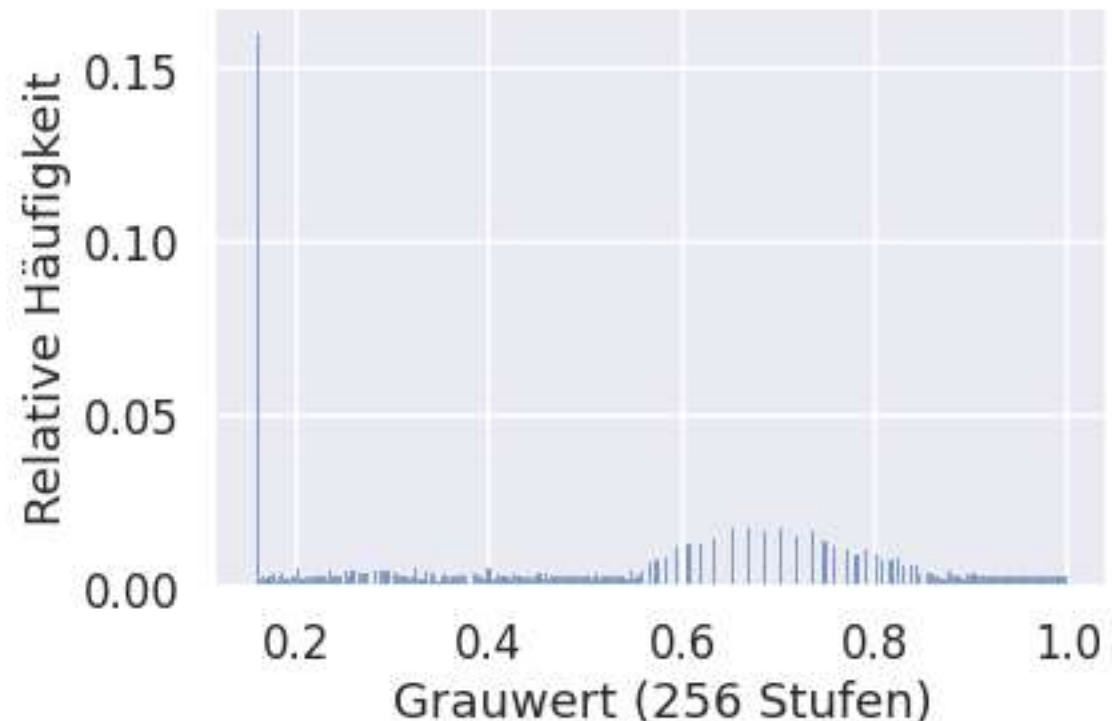
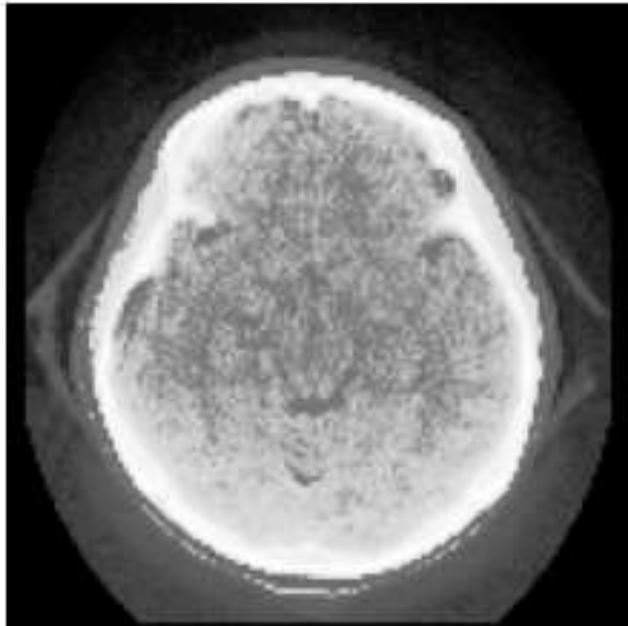
- **Erinnerung:** $t_{HA}(I) := I_{\max} \times H_k(I)$
- **Begründung:** Wenn H_k invertierbar ist, erreicht $t_{HA}(I)$ die gewünschte Gleichverteilung auf $[0, I_{\max}]$:

$$\begin{aligned} P(t_{HA}(I) \leq J) &= P(I_{\max} \times H_k(I) \leq J) \\ &= P\left(I \leq H_k^{-1}\left(\frac{J}{I_{\max}}\right)\right) = H_k\left(H_k^{-1}\left(\frac{J}{I_{\max}}\right)\right) = \frac{J}{I_{\max}} \end{aligned}$$

Definitionsgemäß ist $H_k(I) = P(i \leq I)$

Histogrammausgleich in der Praxis

- Dominante Werte im Ausgangshistogramm kann $t_{HA}(I)$ nicht vollständig ausgleichen
- Wegen Rundung der Ergebnisse und möglicher Lücken im Ausgangshistogramm ist $t_{HA}(I)$ in der Praxis **nicht invertierbar!**



Zusammenfassung: Punktweise Bildtransformationen

- Wir können Bilder als **kontinuierliche Funktionen** darstellen, digitale Bilder als **ganzzahlige Matrizen**
- **Bildfilter** sind Operatoren, die Bilder auf Bilder abbilden
- **Histogramme** geben die Häufigkeitsverteilung von Intensitätswerten in Bildern an
- **Punktweise Bildfilter** transformieren die Intensität an derselben Stelle. Sie dienen insb. zur Kontrastverstärkung
 - **Fensterung** relevanter Intensitätsbereiche
 - **Histogrammausgleich** mittels kumulativer Histogramme

1.4 Lineare Bildfilterung

Motivation: Reduzierung von Bildrauschen

- Wie können wir Bildrauschen reduzieren, wenn wir mit einer Kamera eine statische Szene aufnehmen?

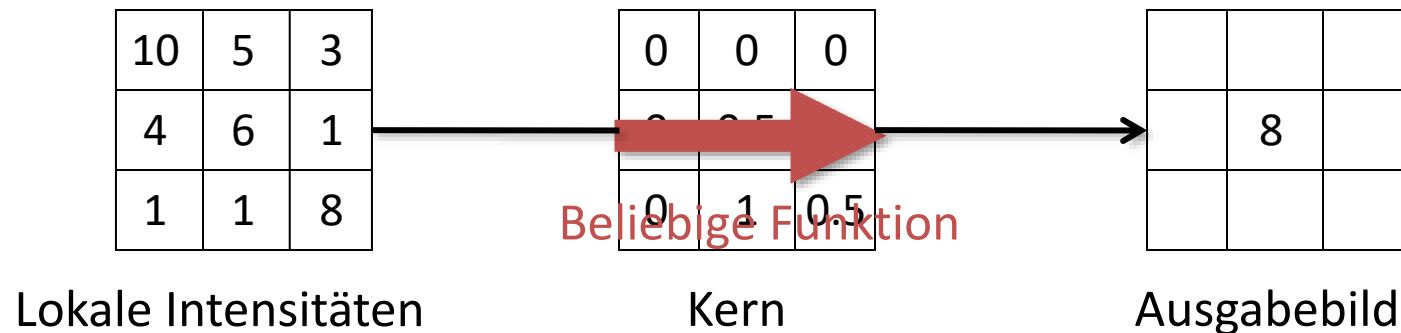


Idee: Mitteln wiederholter Aufnahmen

Was, wenn uns nur eine Aufnahme zur Verfügung steht?

Lokale Bildfilterung

- **Lokale Bildfilter** berechnen neue Pixelwerte als Funktion der Werte in einer lokalen Nachbarschaft
 - Lokalität reduziert den Rechenaufwand
- *Beispiel:* Linearkombination benachbarter Werte
 - Die Gewichte der Linearkombination werden als “Kern” des Filters bezeichnet



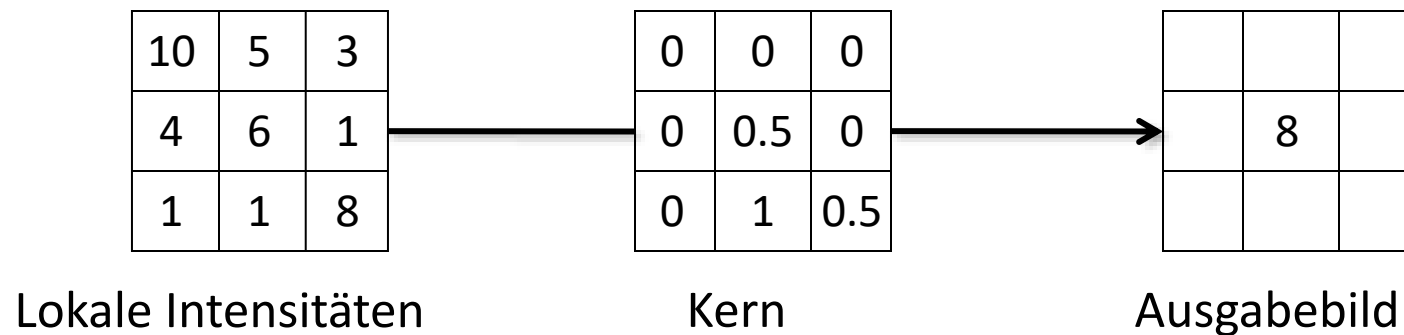
Lineare Filterung

- Ein Filter T ist **linear** wenn für Bilder f und g gleicher Größe und pixelweiser Addition/Skalierung folgendes gilt:

$$T(f + g) = T(f) + T(g)$$

$$T(\alpha f) = \alpha T(f)$$

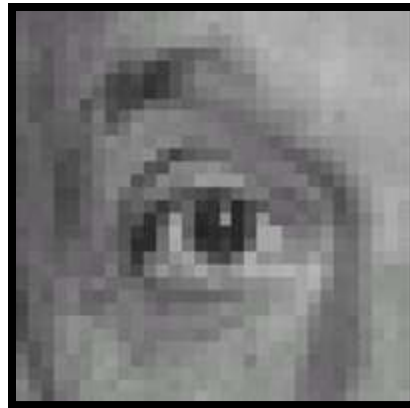
- Beispiel:* Lokale Linearkombinationen mit festen (vom Bildinhalt unabhängigen) Werten



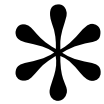
Mittelwertfilter: Erstes Beispiel

$$\frac{1}{9} \begin{matrix} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \\ H \end{matrix} * \begin{matrix} \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 0 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 90 & 90 & 90 & 90 & 90 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 90 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \\ G \end{matrix} = \begin{matrix} \begin{matrix} & & & & & & & & & \\ & 0 & 10 & 20 & 30 & 30 & 30 & 20 & 10 & \\ & 0 & 20 & 40 & 60 & 60 & 60 & 40 & 20 & \\ & 0 & 30 & 60 & 90 & 90 & 90 & 60 & 30 & \\ & 0 & 30 & 50 & 80 & 80 & 90 & 60 & 30 & \\ & 0 & 30 & 50 & 80 & 80 & 90 & 60 & 30 & \\ & 0 & 20 & 30 & 50 & 50 & 60 & 40 & 20 & \\ 10 & 20 & 30 & 30 & 30 & 30 & 20 & 10 & & \\ 10 & 10 & 10 & 0 & 0 & 0 & 0 & 0 & & \\ & & & & & & & & & \end{matrix} \\ F \end{matrix}$$

Mittelwertfilter: Zweites Beispiel

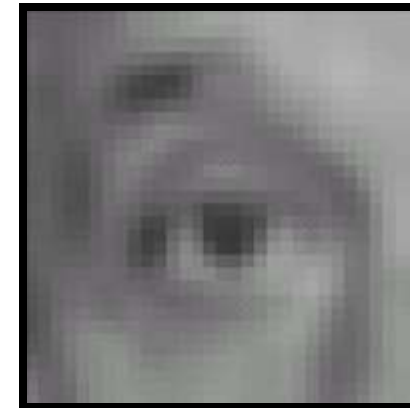


Original



$\frac{1}{9}$

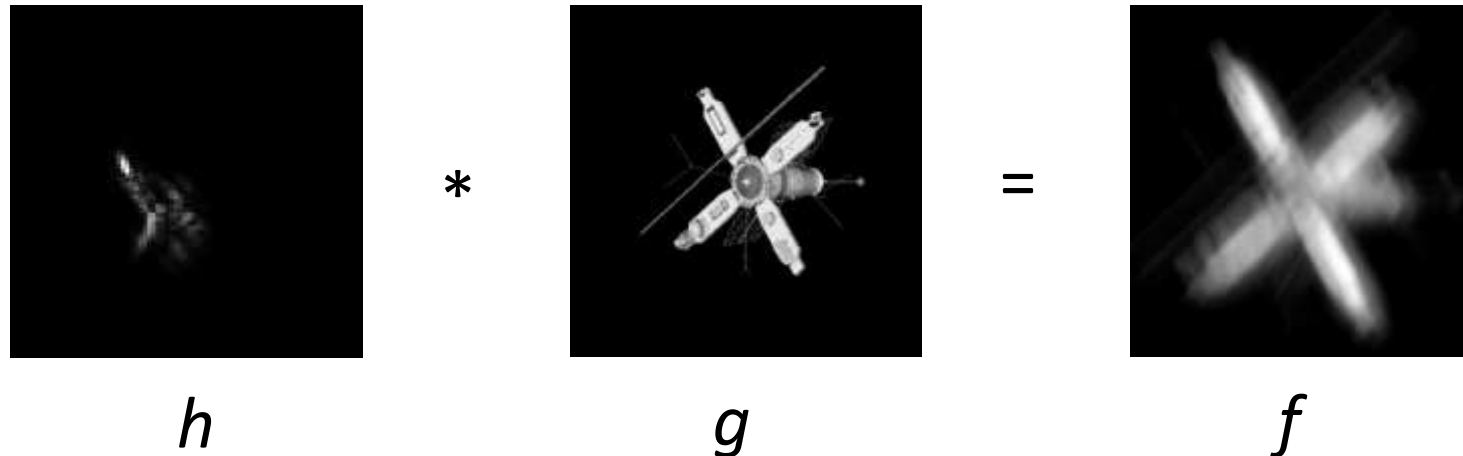
1	1	1
1	1	1
1	1	1



Gefiltert

Faltung: Grundidee

- Die **Faltung** zweier Funktionen h, g ergibt eine neue Funktion f
 - *Vorstellung*: f ergibt sich als gewichtete Summe unterschiedlich verschobener Kopien von g . Die Gewichte sind die Werte von h .
 - *Beispiel*: “Verwackelte” Bildaufnahme
 - Häufig hat h einen kleineren Träger¹ als g und wird als **Faltungskern** bezeichnet.
 - Prinzipiell sind h und g jedoch austauschbar (Faltung ist *kommutativ*)!



¹ Erinnerung: Träger := Abgeschlossene Hülle der Nichtnullstellenmenge

Definition der Faltung im kontinuierlichen Fall

- Die Faltung zweier Funktionen g und h ist definiert durch

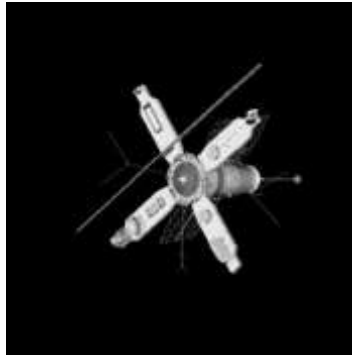
$$f(x) = (h * g)(x) = \int_{-\infty}^{\infty} h(\xi) \cdot g(x - \xi) d\xi$$

- *Beachte:* Negative ξ entsprechen einer Verschiebung von g nach links

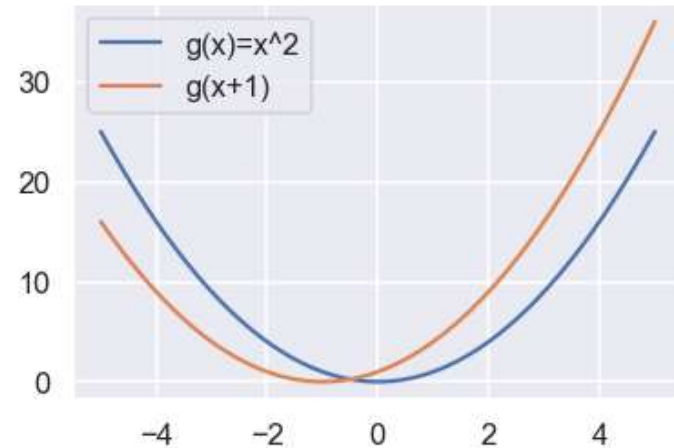


h

*



g



- Die Filterung von 2D-Bildern erfordert 2D-Faltungen. In diesem Fall sind \mathbf{x} und ξ Vektoren: $(h * g)(\mathbf{x}) = \iint h(\xi) g(\mathbf{x} - \xi) d\xi$

Separierbare Faltung in 2D/3D

- Für **separierbare Kerne** h vereinfacht sich das 2D-Faltungsintegral

$$(h * g)(\mathbf{x}) = \iint h(\xi)g(\mathbf{x} - \xi)d\xi$$

zu

$$(h * g)(\mathbf{x}) = \iint h_1(\xi_1)h_2(\xi_2)g(x_1 - \xi_1, x_2 - \xi_2)d\xi$$

- Faktorisierung von h ermöglicht Auswertung durch zwei 1D-Integrale
- In der Bildverarbeitung nutzt man häufig $h_1=h_2$
- Generalisiert entsprechend auch für 3D-Bilder

Diskrete Faltung

- **Definition:** Faltung diskreter 1D-Funktionen h und g :

$$f(i) = (h * g)(i) = \sum_{u=-k}^k h(u) \cdot g(i - u)$$

– Hierbei ist h ein Faltungskern der Größe $(2k + 1)$

- **Definition:** Faltung diskreter 2D-Funktionen h und g :

$$f(i, j) = (h * g)(i, j) = \sum_{u=-k}^k \sum_{v=-k}^k h(u, v) \cdot g(i - u, j - v)$$

Vergleich von Faltung und Kreuzkorrelation

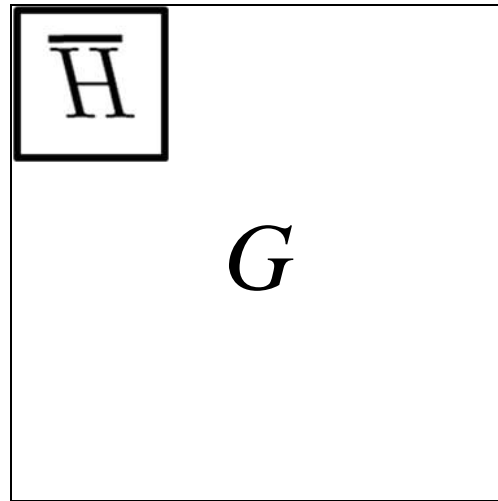
- Die in den ersten Beispielen verwendete **Kreuzkorrelation**

$$f(i, j) = (h \otimes g)(i, j) = \sum_{u=-k}^k \sum_{v=-k}^k h(u, v) \cdot g(i + u, j + v)$$

entspricht einer Faltung mit gespiegeltem Kern h .

- Faltungen haben gegenüber Kreuzkorrelationen den Vorteil, dass sie *kommutativ* und *assoziativ* sind
- Für spiegelsymmetrische Kerne ergeben Faltung und Kreuzkorrelation identische Ergebnisse
 - *Beispiel*: Mittelwertfilter

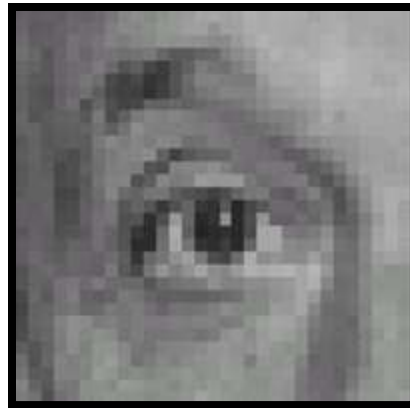
Illustration: Faltung



Randbedingungen

- **Problem:** Wenn wir eine Faltung am Bildrand auswerten wollen, ragt der Kern über das Bild hinaus
- **Lösungen:**
 - Wir schneiden den Rand ab (“gültiger” Teil der Faltung)
 - **Dirichlet-Randbedingung:** Wir nehmen jenseits des Bildrands feste Werte an (häufig Null/schwarz)
 - **Neumann-Randbedingung:** Wir nehmen an, dass entlang des Bildrands die Ableitung in Richtung der äußeren Normalen null ist
 - Entspricht einer Spiegelung der Werte am Bildrand
 - **Periodische Randbedingung:** Wir setzen das Bild in alle Richtungen periodisch fort
 - Ergibt sich bei der Berechnung mittels Faltungstheorem, sehen wir später

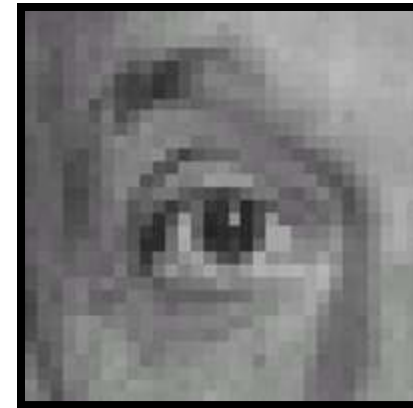
Faltungsbasierte Filterung: Beispiel 1



Original

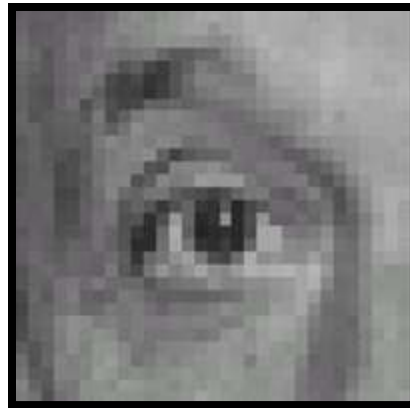


0	0	0
0	1	0
0	0	0



Unverändertes Bild

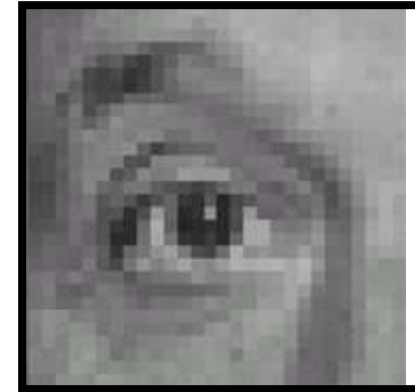
Faltungsbasierte Filterung: Beispiel 2



Original



0	0	0
1	0	0
0	0	0

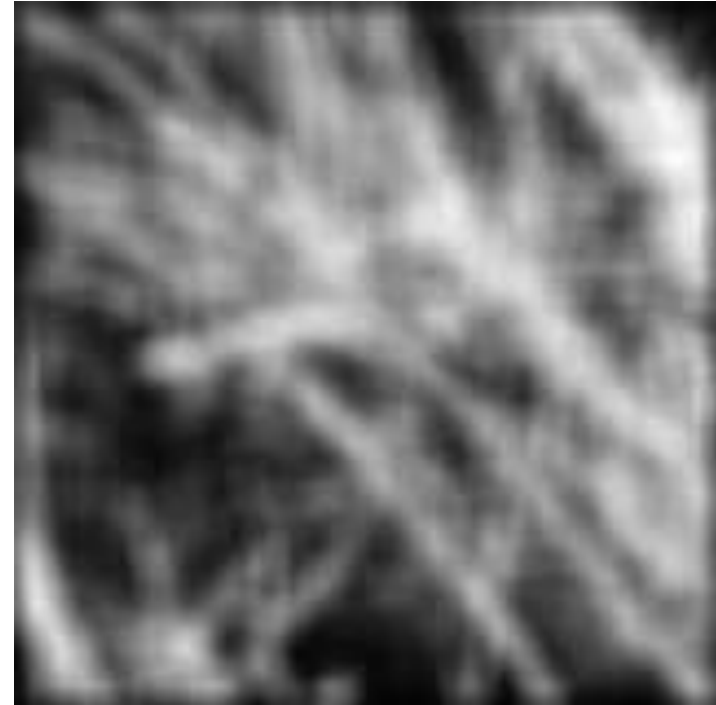


Um 1 Pixel
nach links verschoben

Nachteil des Mittelwert-Filters



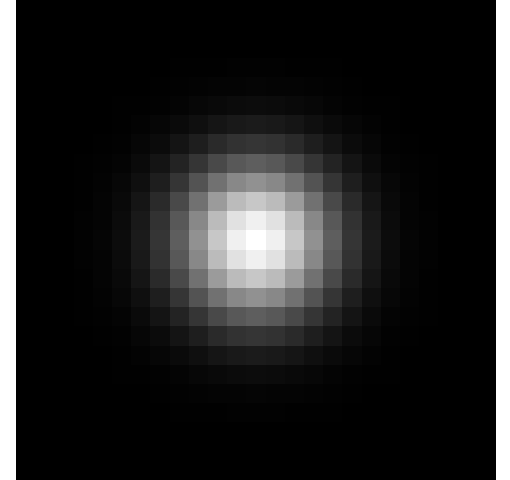
$$* \square =$$



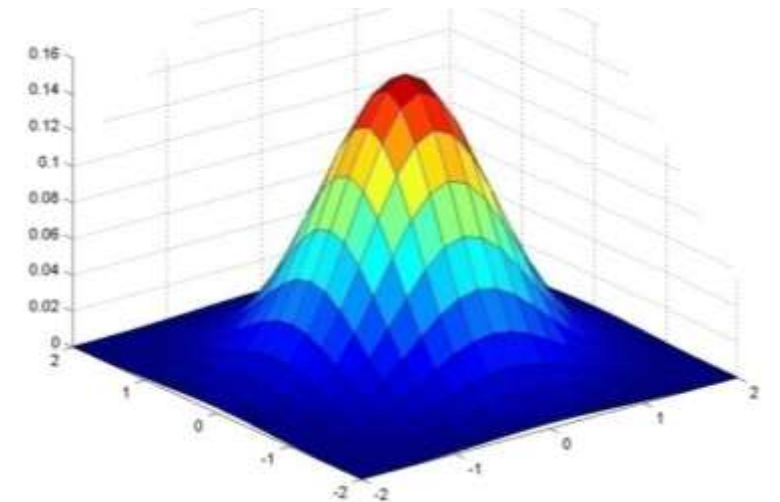
Filterung mit Gauss-Kern

- Eine rotationssymmetrische Tiefpass-Filterung erreicht man durch Filterung mit dem **Gauss-Kern**

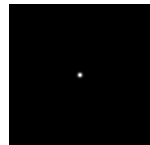
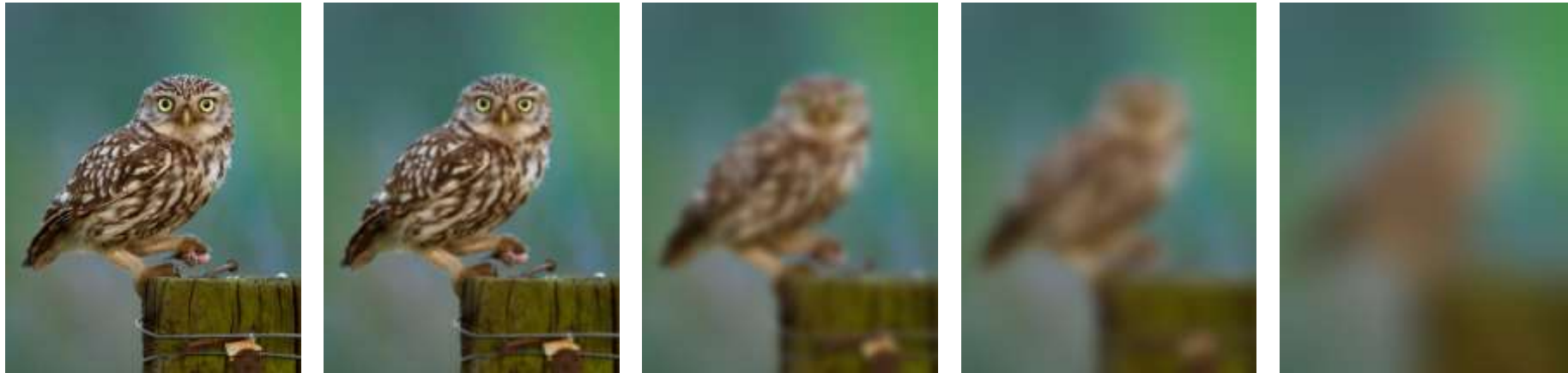
$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



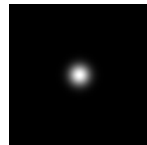
- Dämpft hohe Ortsfrequenzen
- Um Rechenzeit zu sparen schneidet man den Filter meist ab, z.B. nach 3σ
- **Quiz:** Wie wirkt sich ein höheres σ auf das gefilterte Bild aus?



Gauss-Filter: Effekt der Bandbreite



$\sigma = 1$ pixel



$\sigma = 5$ pixel



$\sigma = 10$ pixel



$\sigma = 30$ pixel

Eigenschaften des Gauss-Filters

- Faltung zweier Gauss-Verteilungen mit Standardabweichungen σ_1 und σ_2 ergibt einen Gauss mit Std.-Abweichung $\sqrt{\sigma_1^2 + \sigma_2^2}$



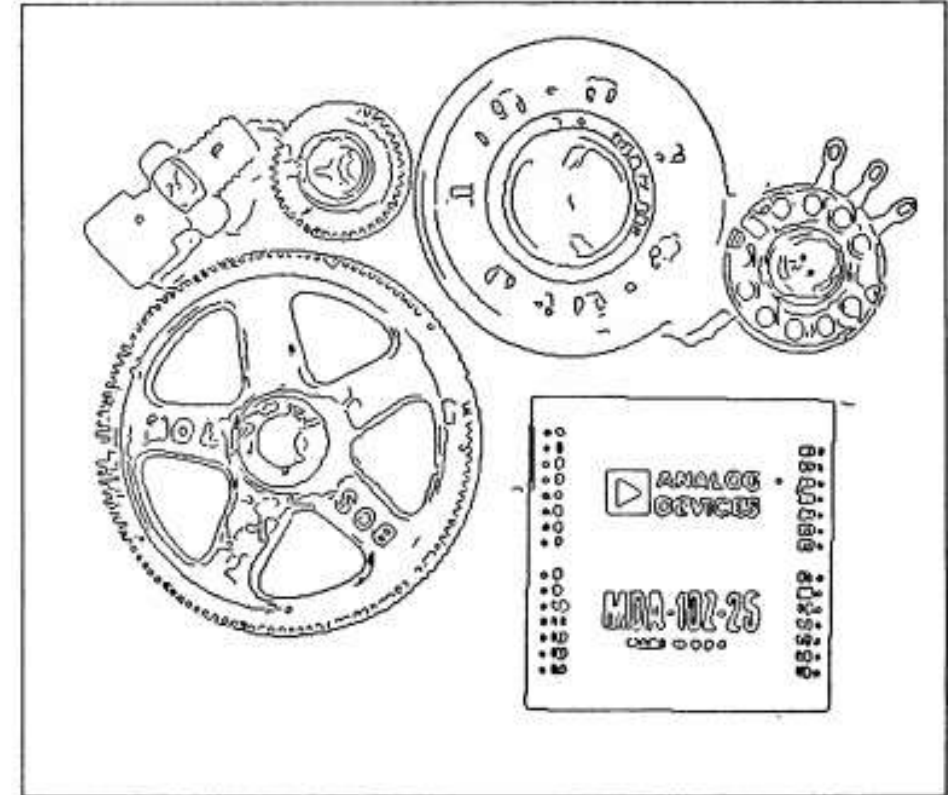
- Zusammen mit der Assoziativität der Faltung ergibt sich daraus, dass wiederholtes Gauss-Filtern dasselbe Ergebnis liefert wie ein einmaliges Filtern mit einem entsprechend breiteren Gauss

Zusammenfassung: Lineare Bildfilterung

- **Lineare Bildfilter** lassen sich durch **Kreuzkorrelationen** oder **Faltungen** ausdrücken
 - *Unterschiede:*
 - Spiegelung des Kerns
 - Faltung ist assoziativ und kommutativ
 - Matrixdarstellung von Faltungskernen
- Beliebte lineare Filter zur **Entrauschung** sind
 - Mittelwert-Filter
 - Gauss-Filter

1.5 Kantenerkennung

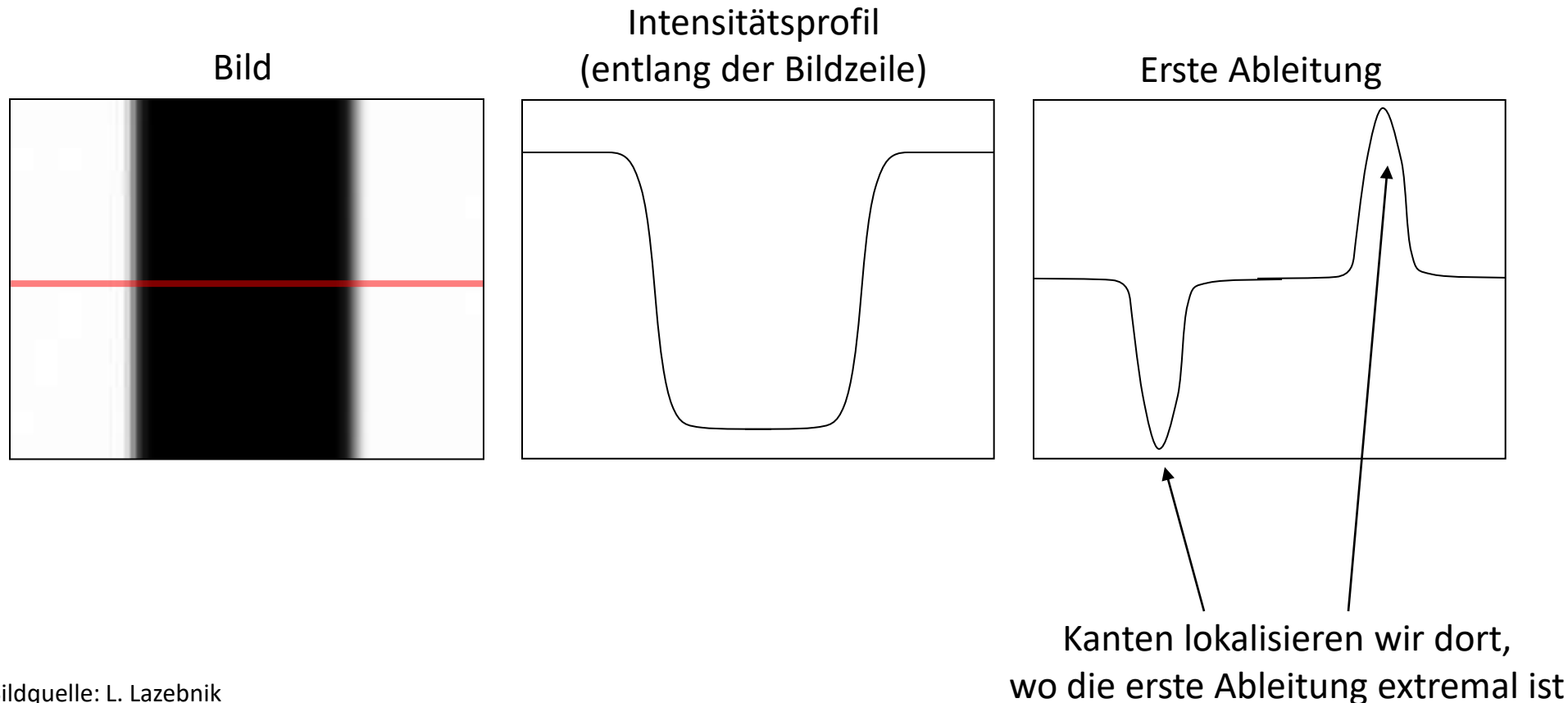
Motivation: Kantenerkennung



- **Ziel der Kantenerkennung:** Extrahiere aus einem 2D-Bild eine Menge von Kurven entlang kontraststarker Kanten
 - *Bedeutung:* Kanten folgen häufig dem Umriss von Objekten oder anderen wichtigen Bildinhalten

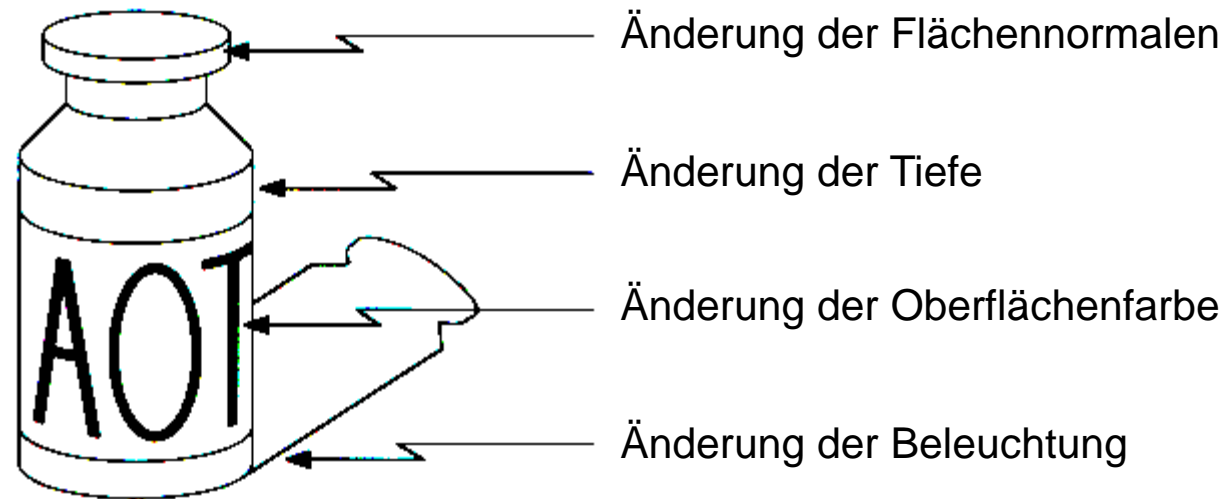
Was macht eine Kante aus?

- Kanten erkennt man daran, dass sich die Bildintensität senkrecht zu ihnen schnell verändert



Woher kommen Kanten?

- Kanten entstehen durch verschiedene Faktoren:



Ableitungen in Bildern

- Wie können wir Ableitung in einem *digitalen* Bild $F[x,y]$ bilden?
 - Option 1: Rekonstruktion eines kontinuierlichen Bildes f , für das wir herkömmliche Ableitungen berechnen können
 - Option 2: Berechnung diskreter Ableitungen (finiter Differenzen)

$$\frac{\partial f}{\partial x}[x, y] \approx F[x + 1, y] - F[x, y]$$

Wie würde man das als Faltungskern schreiben?

$$\frac{\partial f}{\partial x} : \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}$$

H_x

$$\frac{\partial f}{\partial y} : \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array}$$

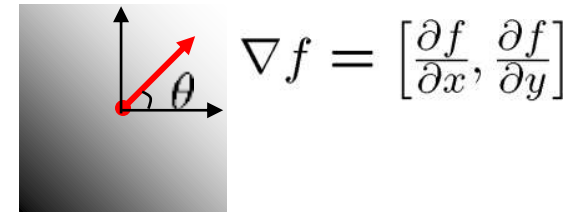
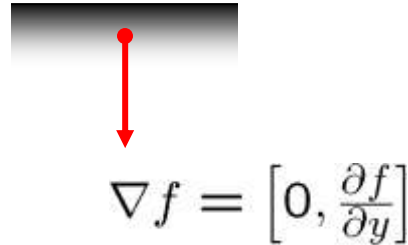
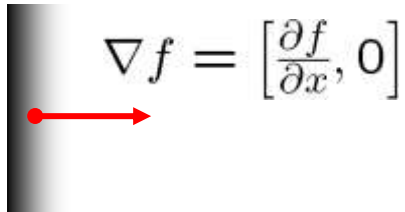
H_y

Hinweis: Wir definieren die Ecke oben links als Ursprung (Matrix-Notation)

Bildgradienten

- Der *Gradient* eines Bildes ist definiert als $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$

Der Gradient weist in die Richtung der steilsten Intensitätsveränderung



Die Norm des Gradienten gibt die *Kantenstärke* an:

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

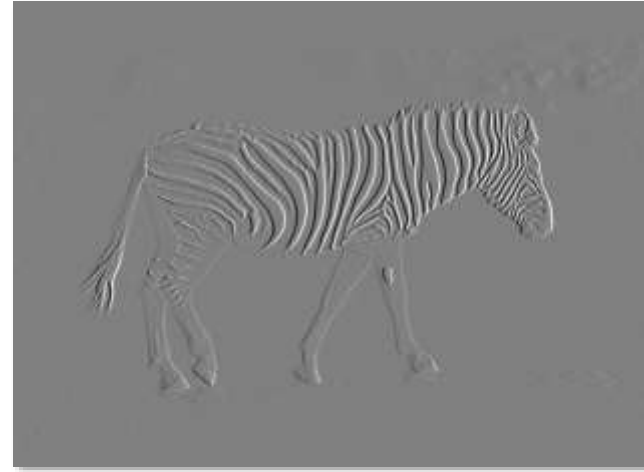
Die Richtung des Gradienten ist $\theta = \tan^{-1} \left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)$

- Frage:** In welcher Richtung verläuft die Kante?

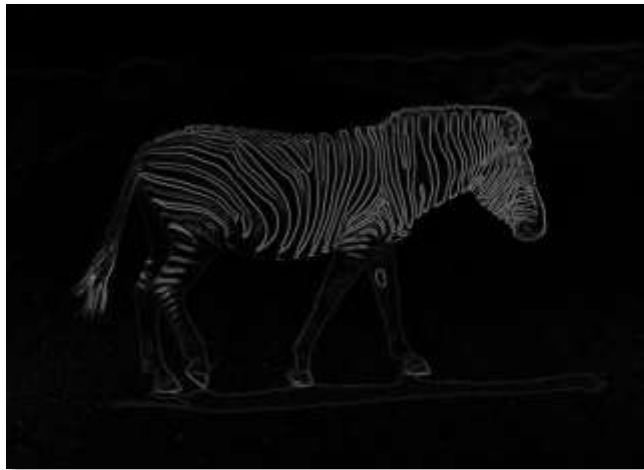
Illustration: Bildgradient



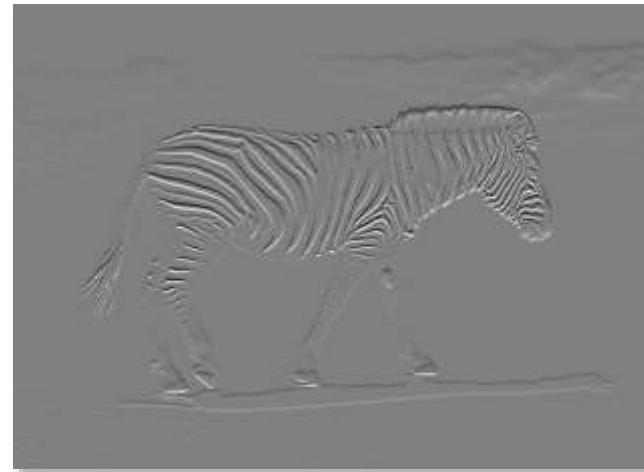
f



$\frac{\partial f}{\partial x}$

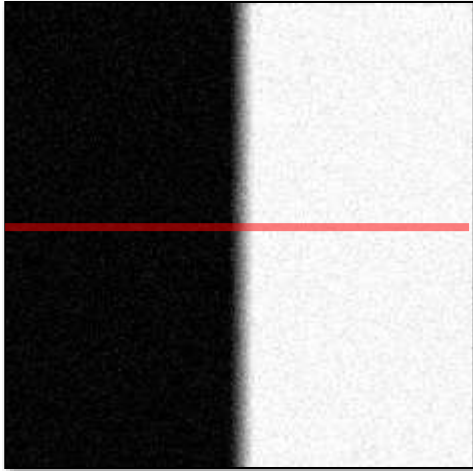


$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$



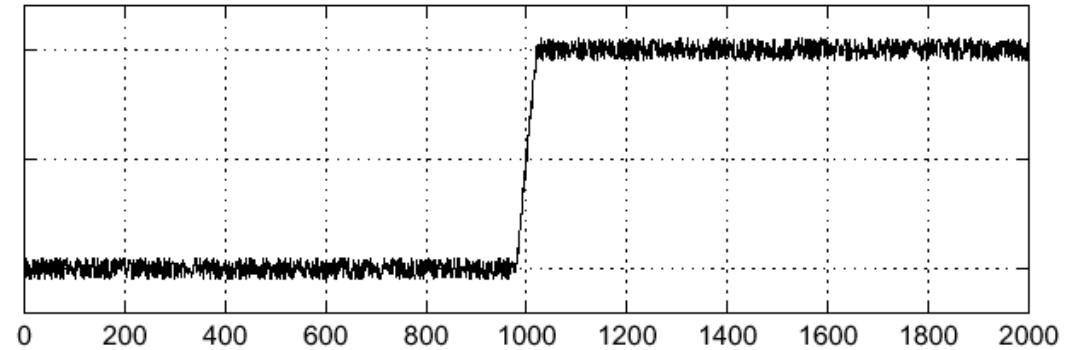
$\frac{\partial f}{\partial y}$

Auswirkung von Rauschen

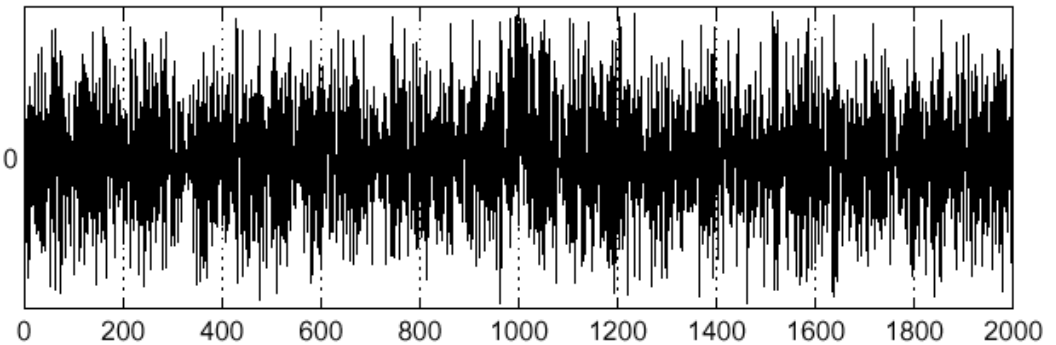


Verrauschtes Eingabebild

$$f(x)$$

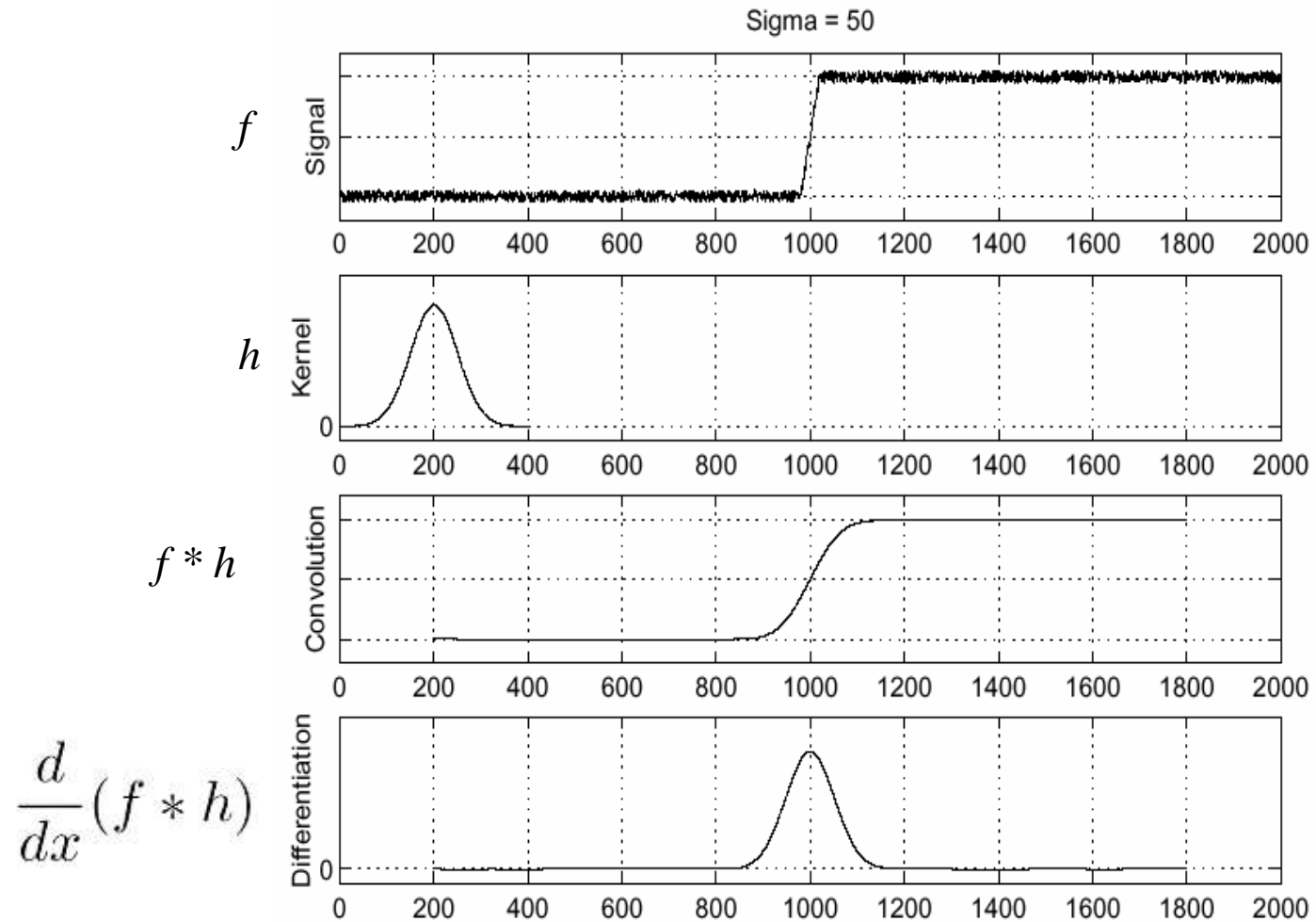


$$\frac{d}{dx}f(x)$$



Wo ist die Kante?

Lösung: Vor Kantenerkennung immer glätten!



Wir bestimmen Kanten als Extrema von $\frac{d}{dx}(f * h)$

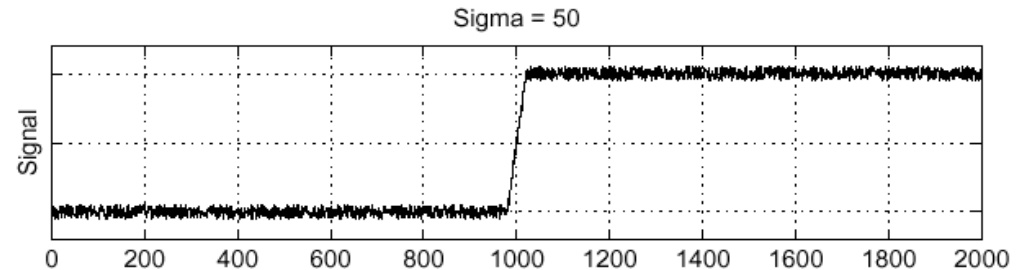
Glättung und Ableitung in Einem

- Wir wissen bereits, dass
 1. Ableitungen als Faltungen implementiert werden können
 2. Faltungen assoziativ und kommutativ sind

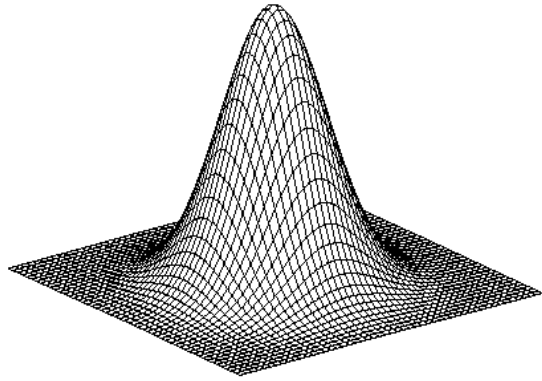
$$\frac{d}{dx}(f * h) = f * \frac{d}{dx}h$$

- Damit können wir beide Operationen kombinieren:

f

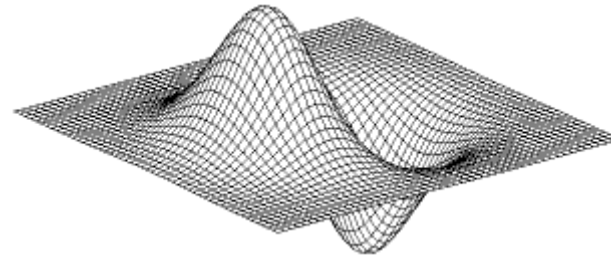


Kantenerkennung in zwei Dimensionen



Gauss

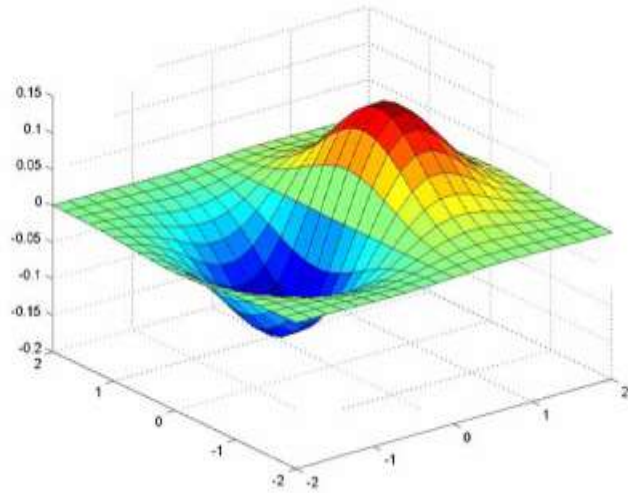
$$h_{\sigma}(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



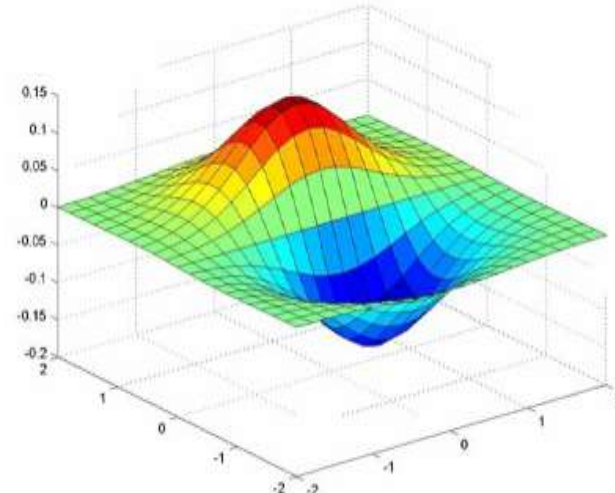
Ableitung des Gauss (x)

$$\frac{\partial}{\partial x} h_{\sigma}(u, v)$$

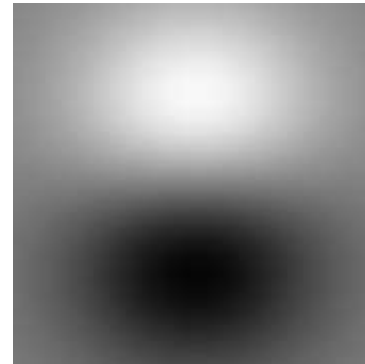
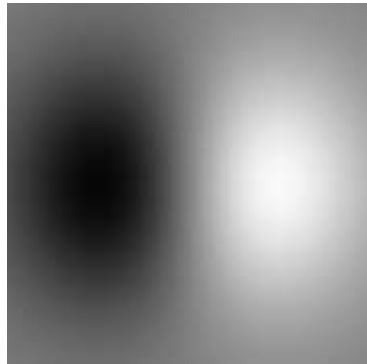
Ableitungen des Gauss-Filters



x-Richtung



y-Richtung



Der Sobel-Operator

- Aufgrund seines großen Trägers ist der Gauss-Filter rechenaufwendig. Der **Sobel-Operator** ist eine beliebte und relative günstig zu berechnende Approximation seiner Ableitung:

$$\frac{1}{8} \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

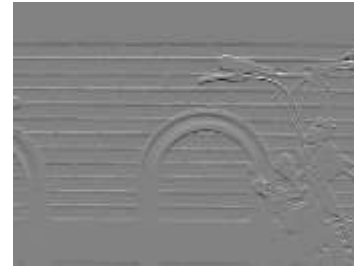
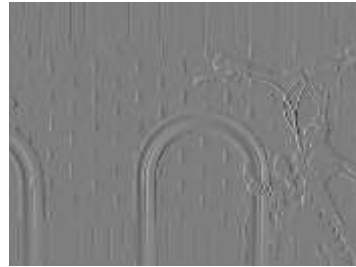
s_x

$$\frac{1}{8} \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

s_y

- Anmerkung:* Häufig findet man Definitionen ohne den Faktor $1/8$
 - Ohne diesen Faktor haben die berechneten Bildgradienten eine zu hohe Norm
 - Der Faktor $1/8$ geht von einem Abstand von eins zwischen den Pixeln aus
 - Konstante Faktoren sind bei der Kantenerkennung jedoch häufig vernachlässigbar

Sobel-Operator: Beispiel



Grundlage der Kantenerkennung: Der Gradient



Eingabebild



Norm der Gradienten

Problem: Genaue Lokalisierung der Kante

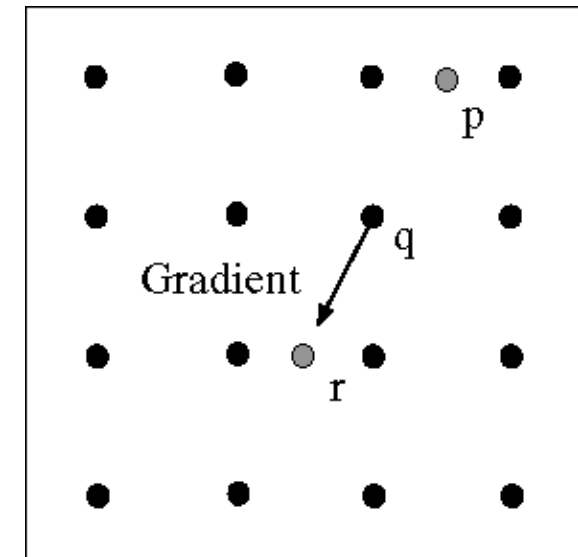
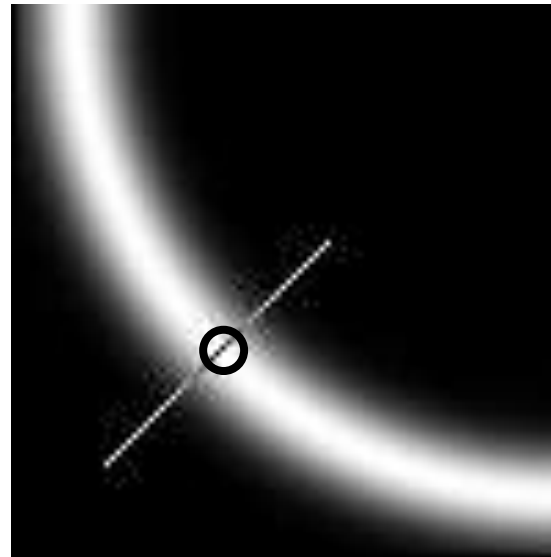
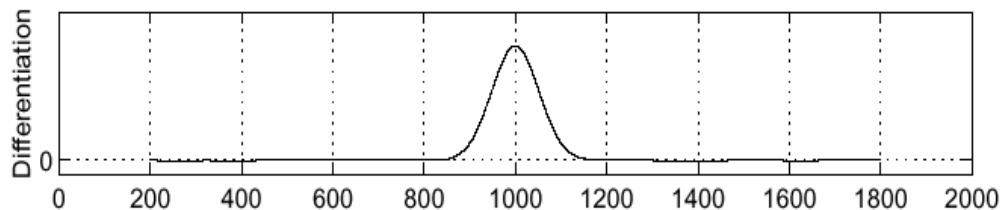


Wo ist die Kante?

Norm der Gradienten

Lösung: Unterdrückung der Nicht-Maxima

- Die **Nicht-Maxima-Unterdrückung** setzt Pixel, an denen die Gradientennorm entlang der Gradientenrichtung kein lokales Maximum ist auf Null
 - Erfordert eine Interpolation an den Punkten p und r
 - Englisch „non maximum suppression“

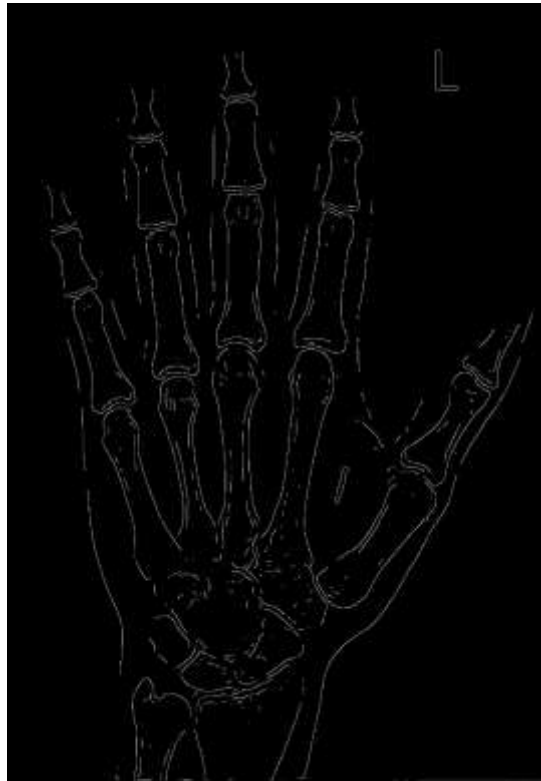


Letzter Schritt: Verkettung und Schwellenwertbildung

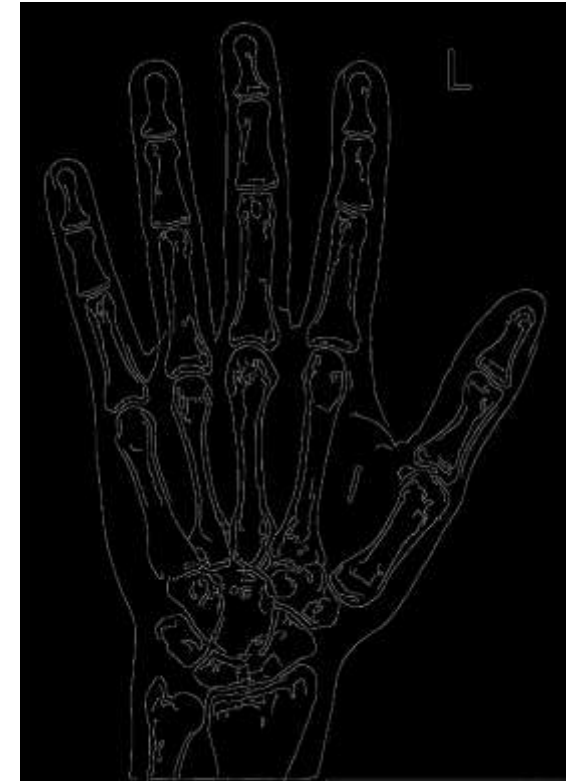
- Canny's Algorithmus **verkettet** Pixel zu Kanten (*linking*) und nutzt dabei **zwei Schwellenwerte** (*Hysteresis*):
 - Höherer Wert zum Start einer neuen Kante, kleinerer Wert zur Fortsetzung



Skelettierung (unterer Wert)



Skelettierung (oberer Wert)



Kombination

Canny-Kantenerkennung: Rolle von σ



Original



Canny mit $\sigma = 2$



Canny mit $\sigma = 3$

- Canny's Algorithmus berechnet Gradienten durch Faltung mit der Ableitung von Gauss-Kernen der Bandbreite σ
 - großes σ erfasst gröbere Kanten ("large-scale" / "hohe Skala")
 - kleines σ erfasst feinere Kanten ("small-scale" / "niedrige Skala")

Zusammenfassung: Canny-Algorithmus zur Kantenerkennung

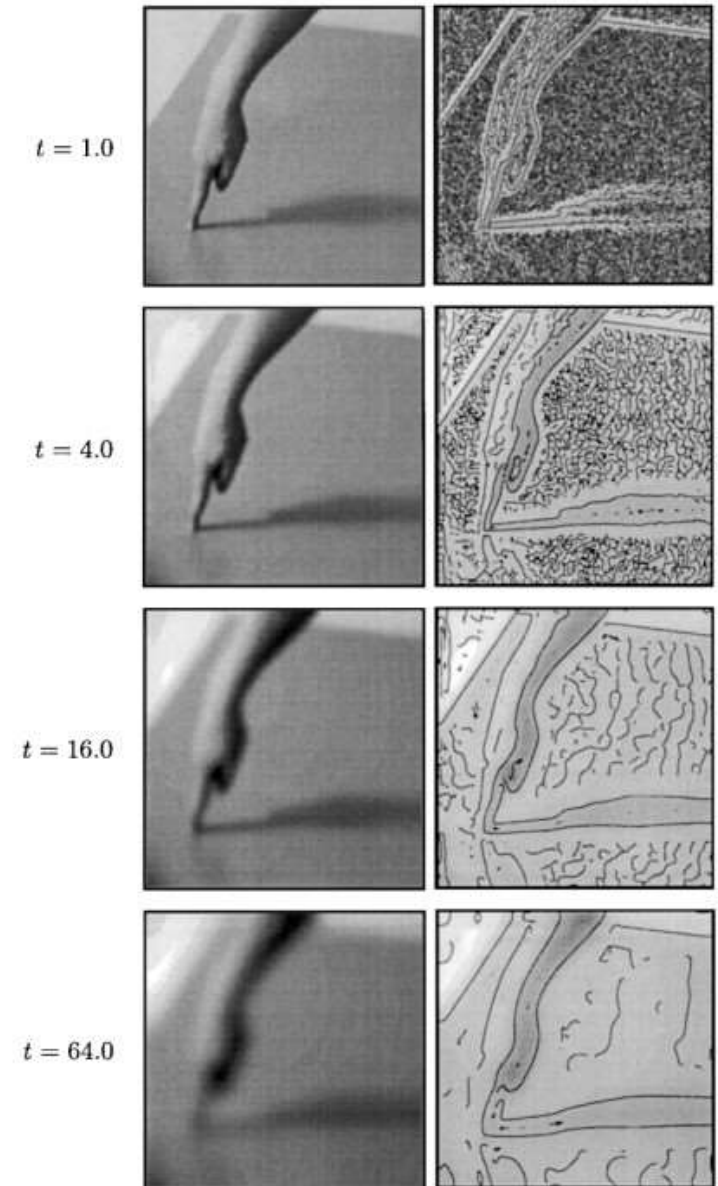


Die Schritte des **Canny-Algorithmus'** sind:

1. Filterung des Bildes mit Ableitungen des Gauss-Kerns
 - Skalen-Parameter σ
2. Berechnung von Stärke und Richtung des Gradienten
3. Unterdrückung von nicht-Maxima
4. Verkettung und Filterung
 - Kombination eines oberen und unteren Schwellenwerts der Kantenstärke

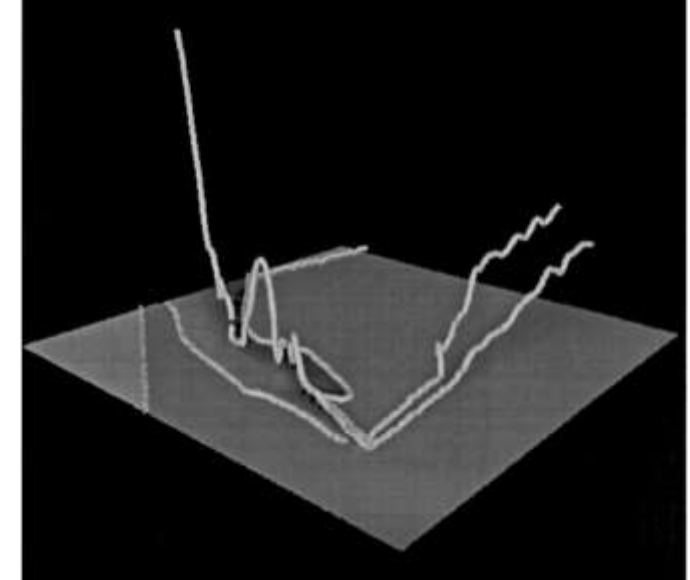
Grundidee der Skalenraumanalyse

- **Problem:** Bilder enthalten häufig Strukturen verschiedener Größe, die mit einem festen σ nicht adäquat erkannt werden
- Grundidee der **Skalenraumanalyse:** Untersuche die Familie aller möglichen geglätteten Bilder
 - $\{T_t f \mid t \geq 0\}$ mit Ausgangsbild f , Glättungsoperator T_t , Glättungsparameter t
 - Häufige Wahl von T_t : Gauss-Glättung mit $\sigma = \sqrt{2t}$

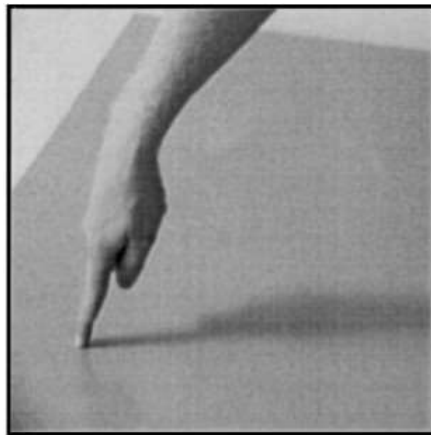


Details: Kantenerkennung im Skalenraum

- Die Kanten aller Skalen zusammengenommen ergeben im $(2D+t)$ Skalenraum zunächst Flächen
- Auf diesen Flächen wählen wir lokale Maxima der **Kantenstärke** $\sqrt{t} \|\nabla(\tau_t f)\|^2$ entlang der t -Achse aus
 - Faktor \sqrt{t} kompensiert Kontrastabschwächung



Kanten im Skalenraum



Ausgangsbild



2D-Projektionen der 50/20/10 stärksten Kanten

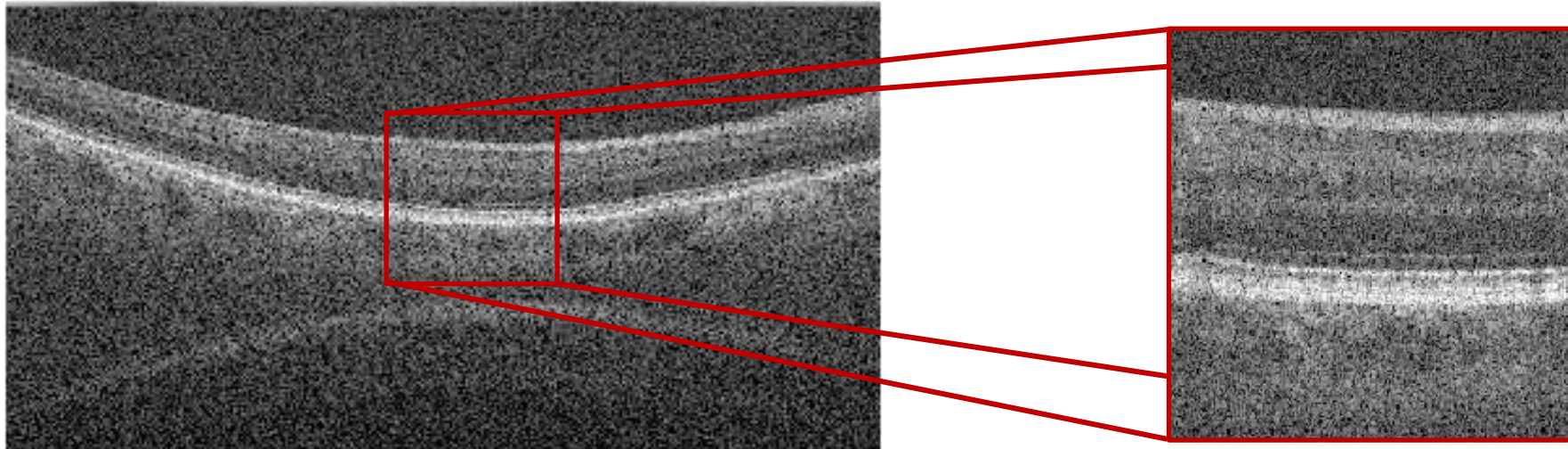
Zusammenfassung: Kantenerkennung

- **Kanten** sind plötzliche Änderungen der Bildintensität
- Wir berechnen sie über den **Gradienten**, konkret mittels
 - Finiten Differenzen,
 - Faltung mit Ableitungen des Gauss-Kerns oder
 - Sobel-Operator
- Der **Canny-Algorithmus** zur Kantenerkennung ist weit verbreitet
 - Als Nutzer kann man gewünschte Skala und Stärke einstellen
- Eine **Skalenraumanalyse** betrachtet das Bild auf allen Skalen gleichzeitig und wählt pixelweise die passende Skala aus

1.6 Nichtlineare Bildfilterung

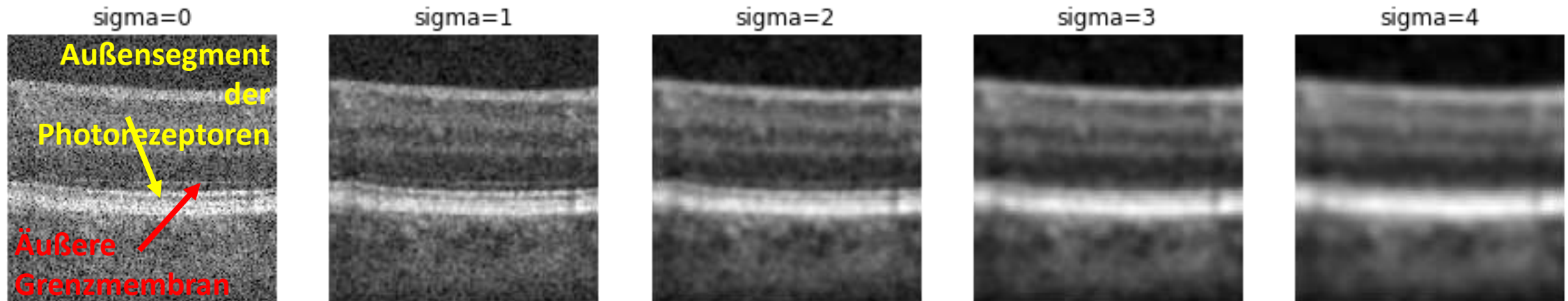
Motivation

- **Nichtlineare Bildfilter** ermöglichen es beim Entfernen von Rauschen relevante Bildstrukturen zu erhalten, insbesondere Kanten
- *Beispiel:* Querschnitt der Retina, aufgenommen mittels Optischer Kohärenz-Tomographie (OCT)



Warum Gauss-Glättung oft nicht reicht

- **Problem:** Gauss-Glättung entfernt nicht nur Rauschen, sondern auch relevante Bildstrukturen
 - Gewichte des Filterkerns sind in jeder Richtung gleich
 - In der Nähe von Kanten mitteln wir über Pixel, die verschiedene Strukturen zeigen
 - Wenn wir wüssten, welche Pixel zu welchem Objekt gehören könnten wir das vermeiden – wissen wir aber meist nicht!



Bilaterale Filterung

- Der **bilaterale Filter** mittelt nur Pixel, die sowohl räumlich nah beieinander sind, als auch ähnliche Intensitäten haben

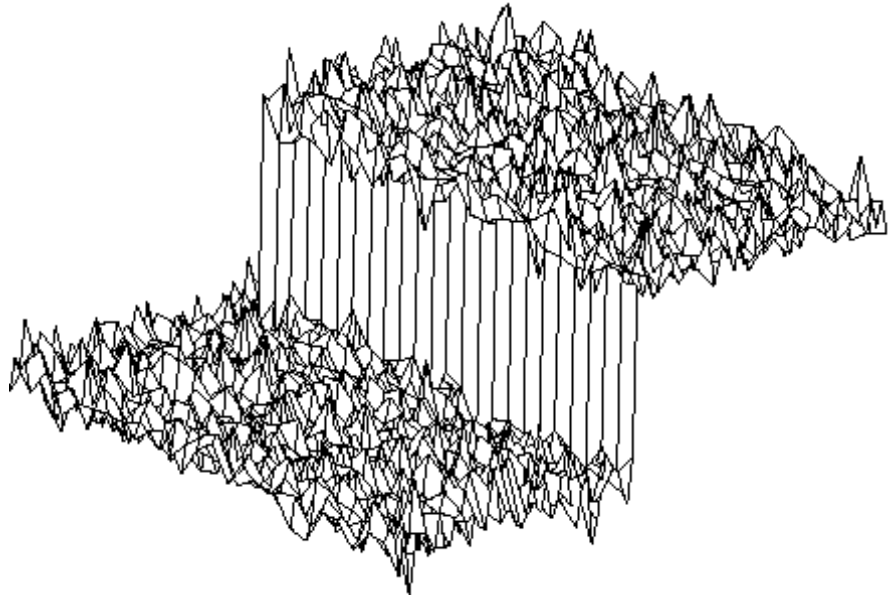
- Separate Parameter σ_c, σ_s steuern entsprechende Gewichte

$$c(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\sigma_c^2}\right), s(g(\mathbf{x}), g(\mathbf{x}')) = \exp\left(-\frac{|g(\mathbf{x}) - g(\mathbf{x}')|^2}{\sigma_s^2}\right)$$

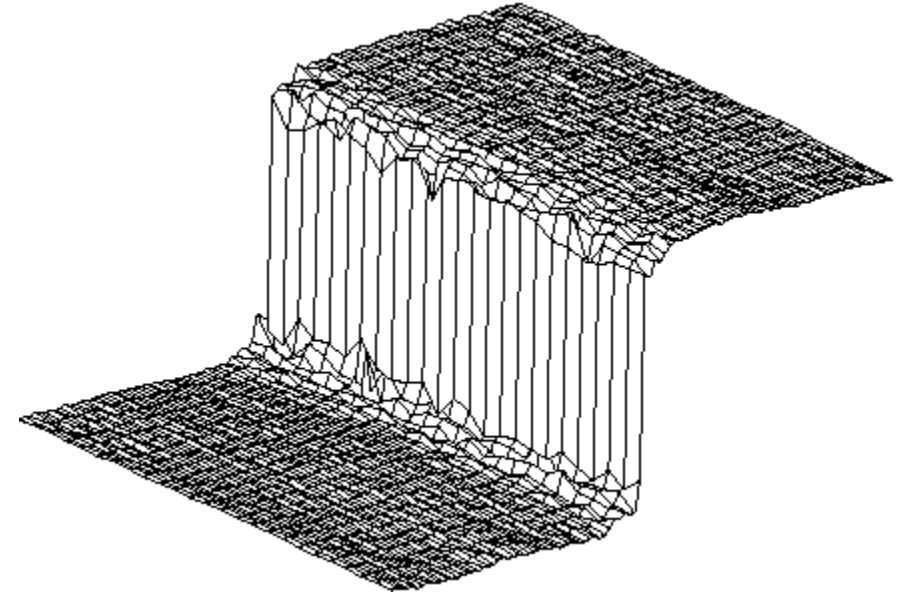
- Das Ergebnis $f(\mathbf{x})$ einer bilateralen Filterung von $g(\mathbf{x})$ in einer gegebenen Nachbarschaft $\omega_{\mathbf{x}}$ um jeden Punkt \mathbf{x} ergibt sich durch Multiplikation beider Gewichte und entsprechende Normierung:

$$f(x) = \frac{\sum_{\mathbf{x}' \in \omega_{\mathbf{x}}} g(\mathbf{x}') c(\mathbf{x}, \mathbf{x}') s(g(\mathbf{x}), g(\mathbf{x}'))}{\sum_{\mathbf{x}' \in \omega_{\mathbf{x}}} c(\mathbf{x}, \mathbf{x}') s(g(\mathbf{x}), g(\mathbf{x}'))}$$

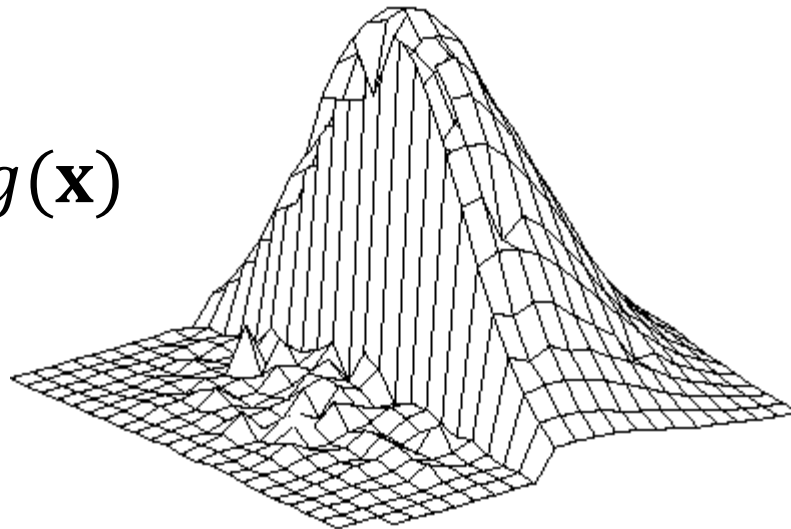
Bilaterale Filterung: Illustration



Verrauschtes Signal $g(\mathbf{x})$



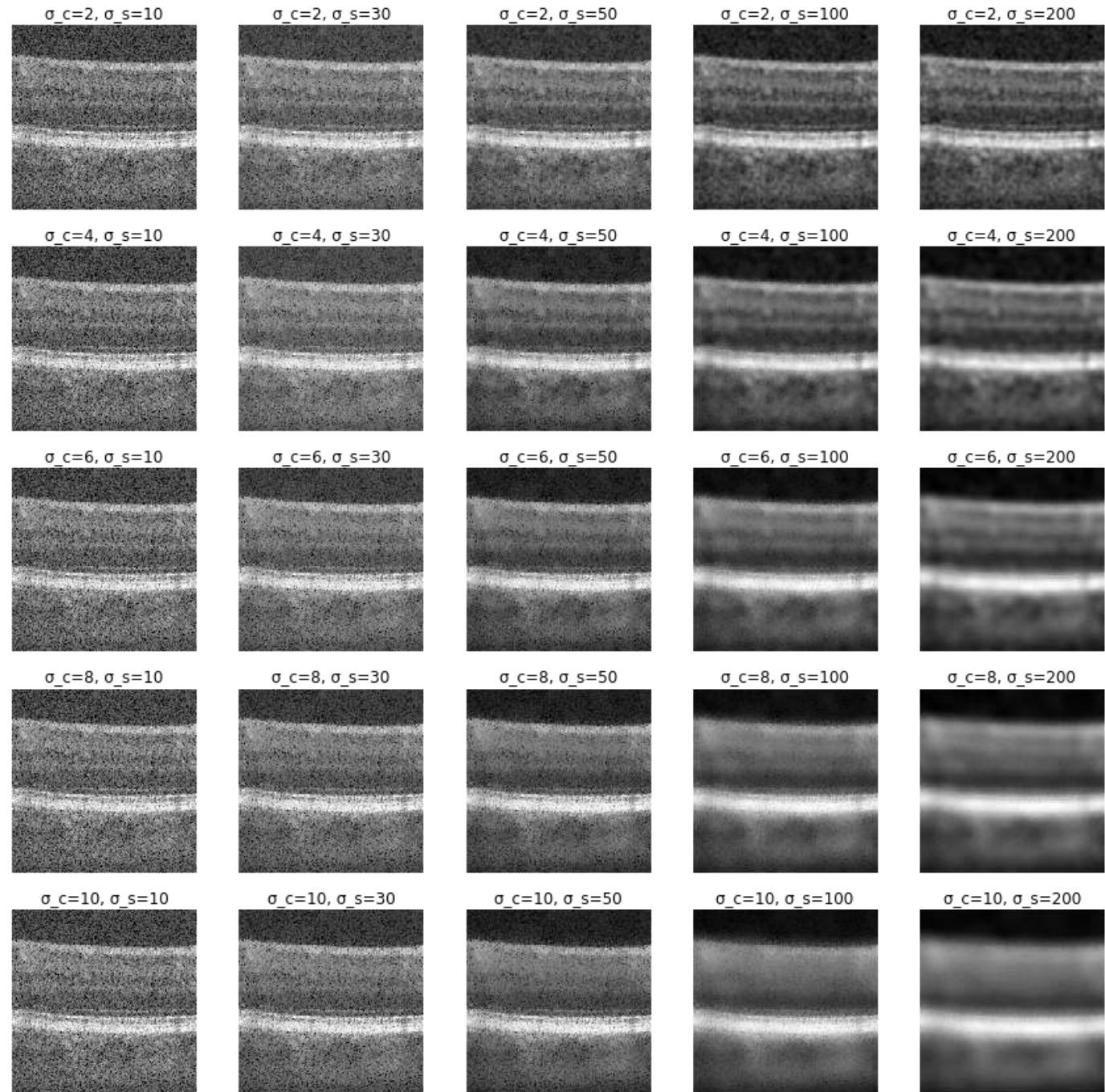
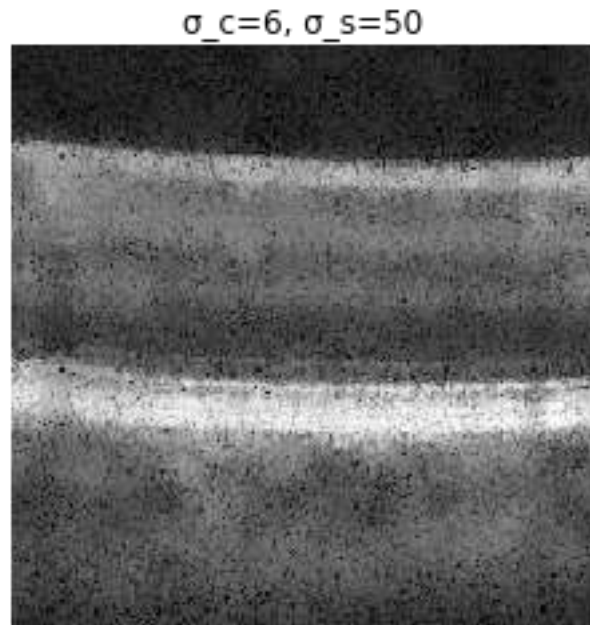
Gefiltertes Signal $f(\mathbf{x})$



Bilaterales Gewicht $c(\mathbf{x}, \mathbf{x}') \times s(g(\mathbf{x}), g(\mathbf{x}'))$

Bilateraler Filter: Parameter-Wahl

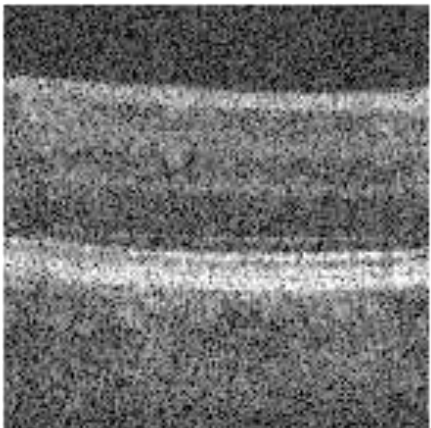
- Was, wenn σ_s hohe Werte annimmt?
- Wie würden Sie die Parameter hier setzen?



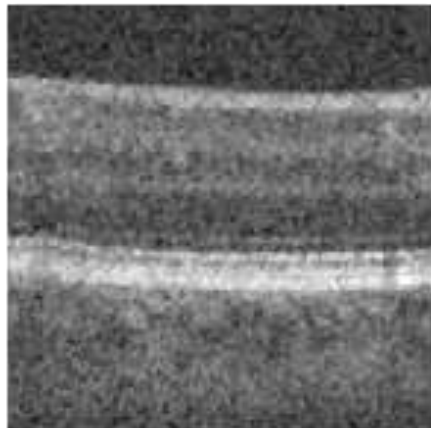
Median-Filter

- **Bemerkung:** Pixel, die viel heller oder dunkler sind als ihre Umgebung, werden von bilateralen Filtern nicht entauscht
 - *Beispiel: Specklemuster*, ein Interferenzphänomen, das in manchen Modalitäten (z.B. OCT) auftritt
- **Median-Filters** entfernen solche Ausreißer, indem sie jeden Pixel durch den Median einer lokalen Nachbarschaft ersetzen

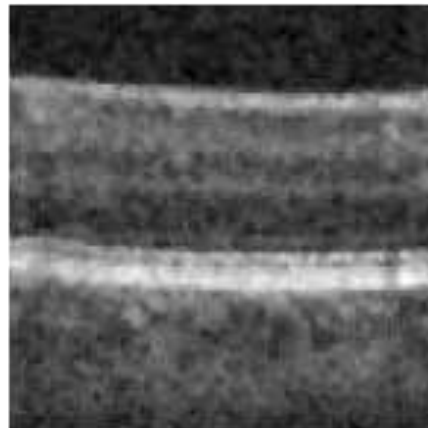
size=1



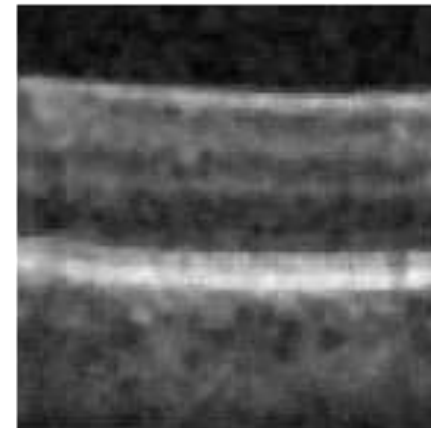
size=3



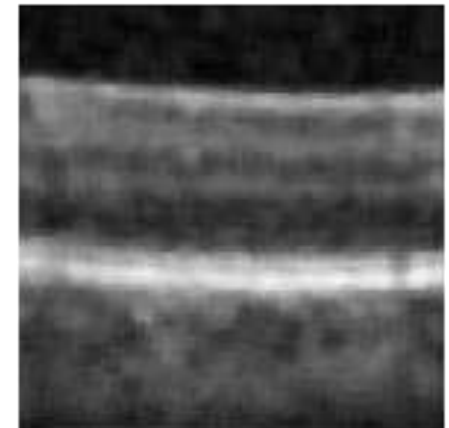
size=5



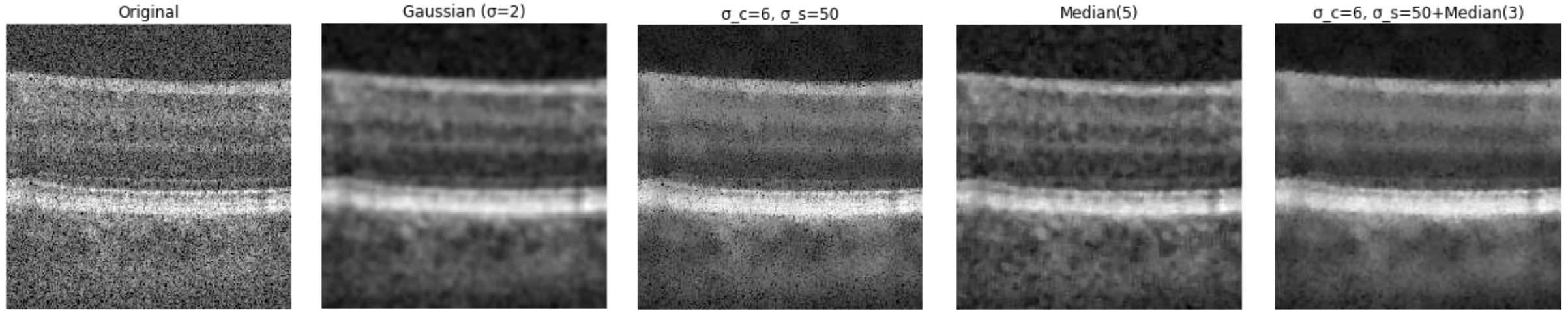
size=7



size=9



Visueller Vergleich verschiedener Filter



- *Hinweis:* In der Praxis kann es sinnvoll sein mehrere Filter zu kombinieren, z.B. durch einen Median-Filter Ausreißer zu entfernen, die nach einer bilateralen Filterung verbleiben

Zusammenfassung: Nichtlineare Bildfilter

- Im Gegensatz zu linearen Filtern haben **nichtlineare Bildfilter** die Möglichkeit sich an Bildinhalte anzupassen
- In der Praxis beliebte nichtlineare Filter sind u.a.
 - **Bilaterale Filter**, die eine kantenerhaltende Filterung ermöglichen
 - **Median-Filter** zur Beseitigung von Ausreißern

Zum Nach- und Weiterlesen

- Heinz Handels: Medizinische Bildverarbeitung. Vieweg+Teubner, 2. Auflage, 2009
- Ramesh Jain, Rangachar Kasturi, Brian G. Schunck: *Machine Vision*. McGraw-Hill 1995
- John Canny: *A Computational Approach to Edge Detection*. IEEE Trans. on Pattern Analysis and Machine Intelligence 8(6):679-698, 1986
- Carlo Tomasi, Roberto Manduchi: *Bilateral Filtering for Gray and Color Images*. In: Proc. Int'l Conf. on Computer Vision (ICCV), pp. 839-846, 1998