

Computergraphik 1

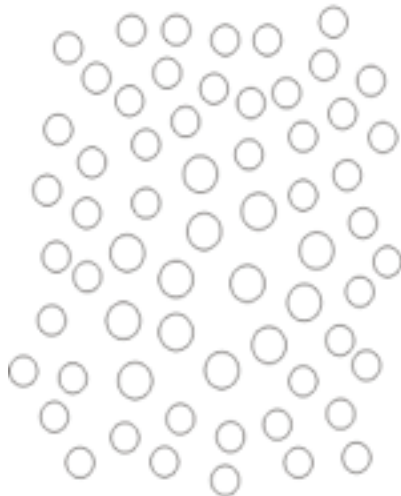
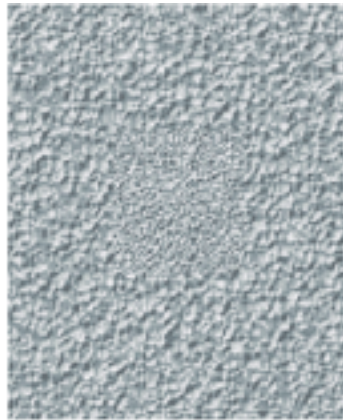
2. Teil: Bildverarbeitung

Bildverbesserung 2

Themen jetzt

- Lineare Bildverbesserung
 - Kantenoperatoren
 - Kantenattribute
 - Kanten und Rauschen
- Nichtlineare Filter
 - Rangordnungsfilter
 - Diffusionsfilter
- Segmentierung
 - Schwellwertbasiert
 - Shadingkorrektur

Was sind Kanten?



Diskontinuitäten zwischen Regionen.

Problem:

Die Region selbst und damit ihr **Homogenitätskriterium** muss bekannt sein, um Diskontinuität definieren zu können.

Homogenität einer Funktion:

- Grauwerte
- Textur
- Form

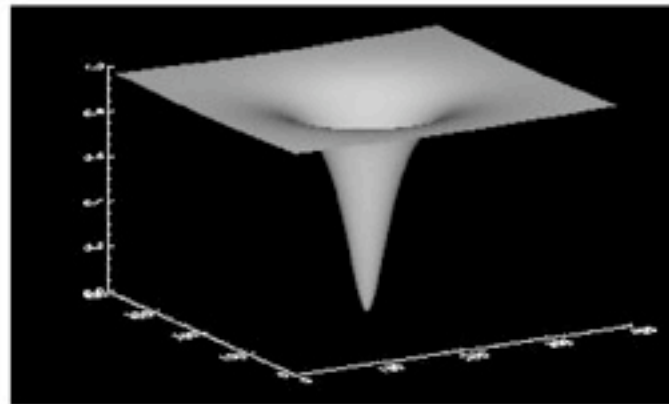
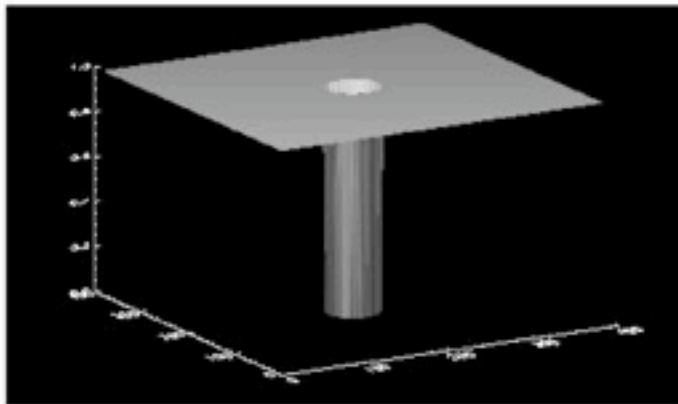
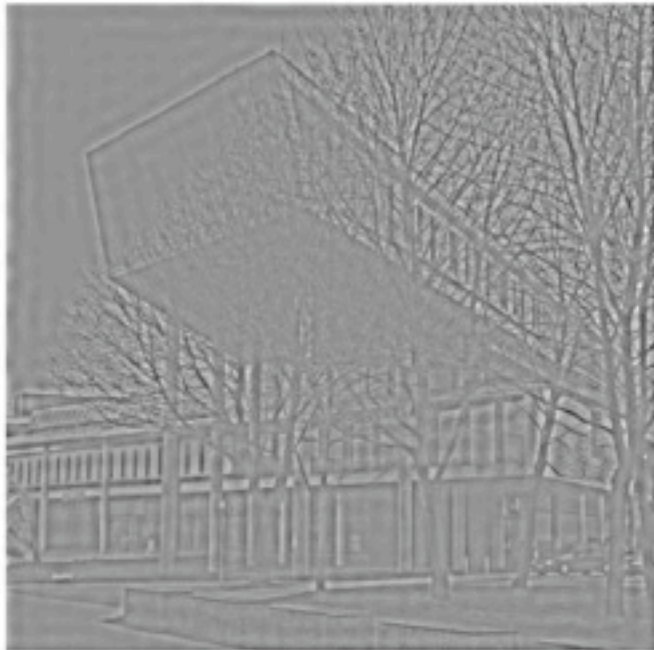
Kantenhervorhebung durch Frequenzraumfilterung

- Kanten weisen mehr hochfrequente Anteile auf als homogene Gebiete
- ▶ Hochpassfilterung

$$H_{F_{\max}}(u, v) = \begin{cases} 1 & , \text{ falls } u^2 + v^2 \geq F_{\max}^2 \\ 0 & , \text{ sonst.} \end{cases}$$



Filterung mit Idealem Hochpass- (links) und mit Butterworth-Hochpassfilter (rechts)



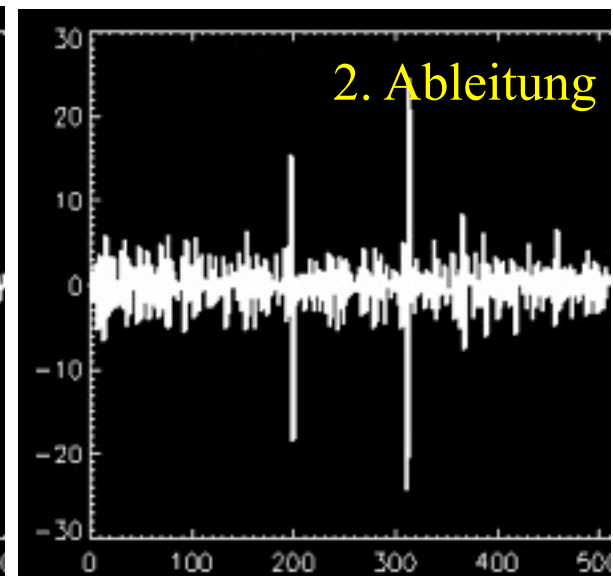
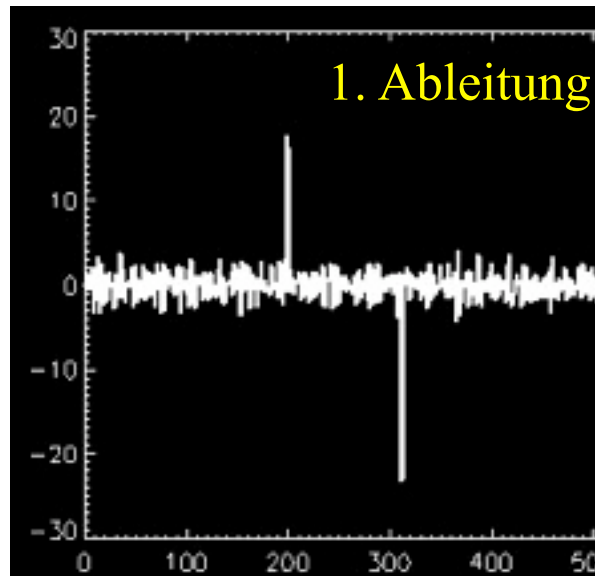
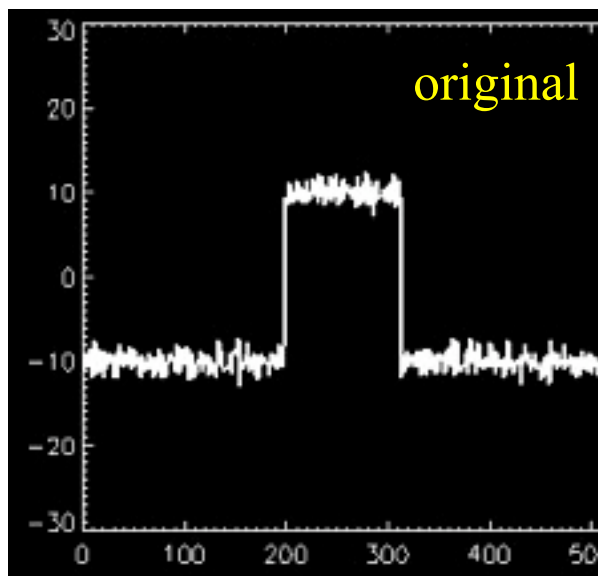
Butterworth-Hochpassfilter



$$H_{F_{\max}}(u, v) = \begin{cases} \frac{1}{1 + \left(\frac{F_{\max}^2}{(u^2 + v^2)^k} \right)} & u \neq 0 \vee v \neq 0 \\ 0 & \text{sonst} \end{cases}$$

Kanten im Ortsraum

Kanten und Rauschen haben **ähnliche Charakteristika** im Frequenzraum → Kantendetektor verstärkt Rauschen.

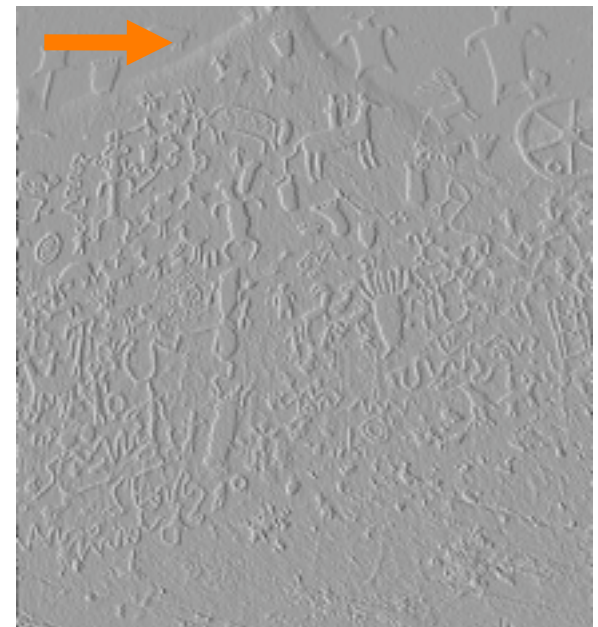
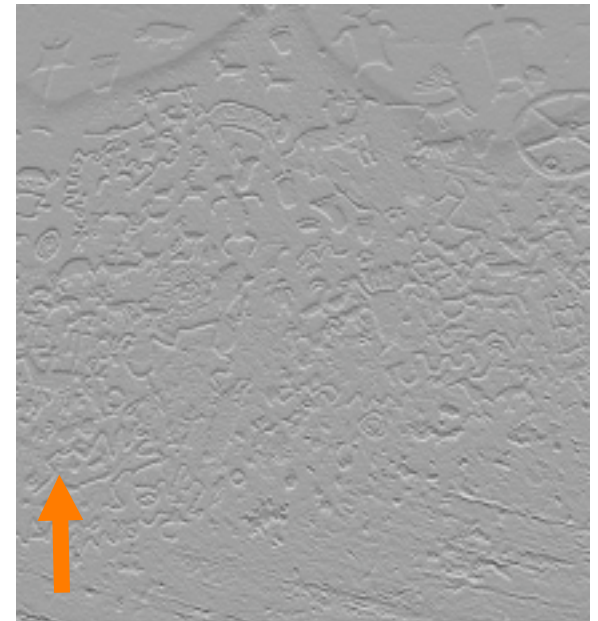


Kanten im 2-D Raum

Differenzbildung in n -Richtung



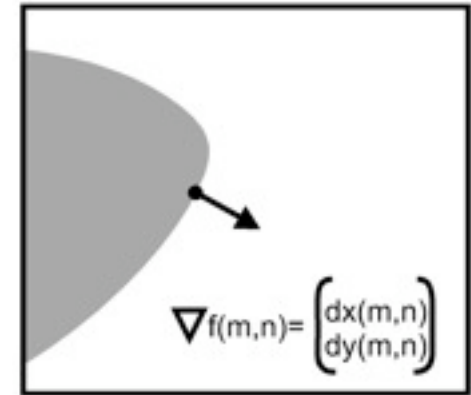
Pseudo-3D
Eindruck:



Differenzbildung in m -Richtung

Gradientenschätzung über Konvolution

- Gradient im N -dimensionalen Raum ist ein N -dimensionaler Vektor aus N partiellen Ableitungen.
- Jede partielle Ableitung kann durch eine Differenz abgeschätzt werden, die durch Konvolution berechnet werden kann.
- Beispiele:

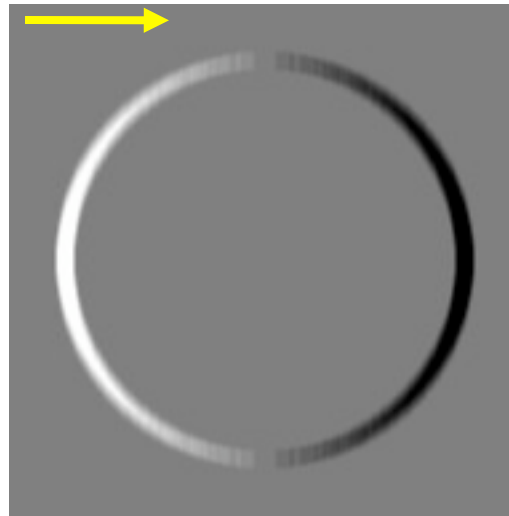
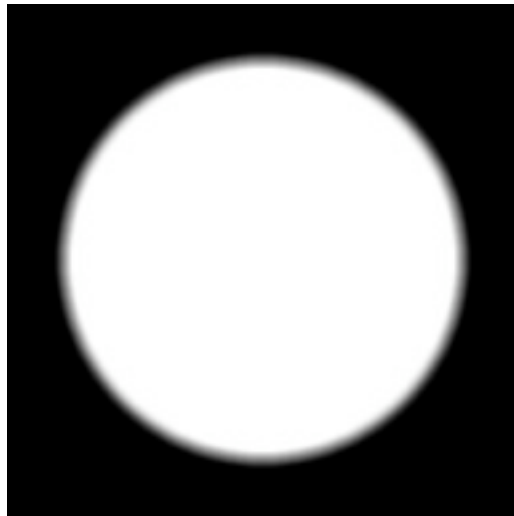


$$\vec{G}(m,n) = \begin{pmatrix} G_x(m,n) \\ G_y(m,n) \end{pmatrix}, \quad G_x(m,n) \approx [f * g_x][m,n], \quad G_y(m,n) \approx [f * g_y][m,n]$$

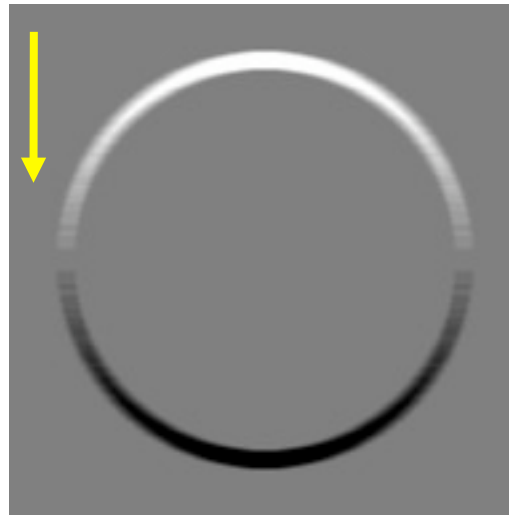
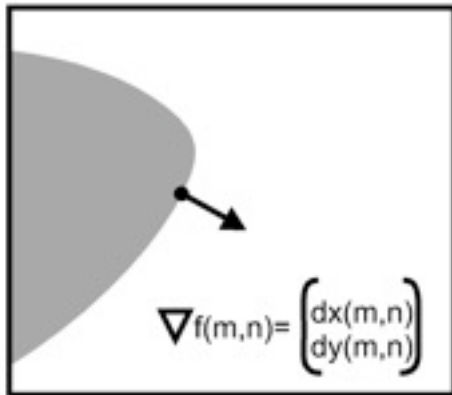
$$g_x = \begin{bmatrix} -1 & 1 \end{bmatrix}, \quad g_y = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad g_{R1} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad g_{R2} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \text{ (Robert's Gradient)}$$

$$g_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}, \quad g_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \text{ (symmetrischer Konvolutionskern)}$$

Elemente des Gradienten

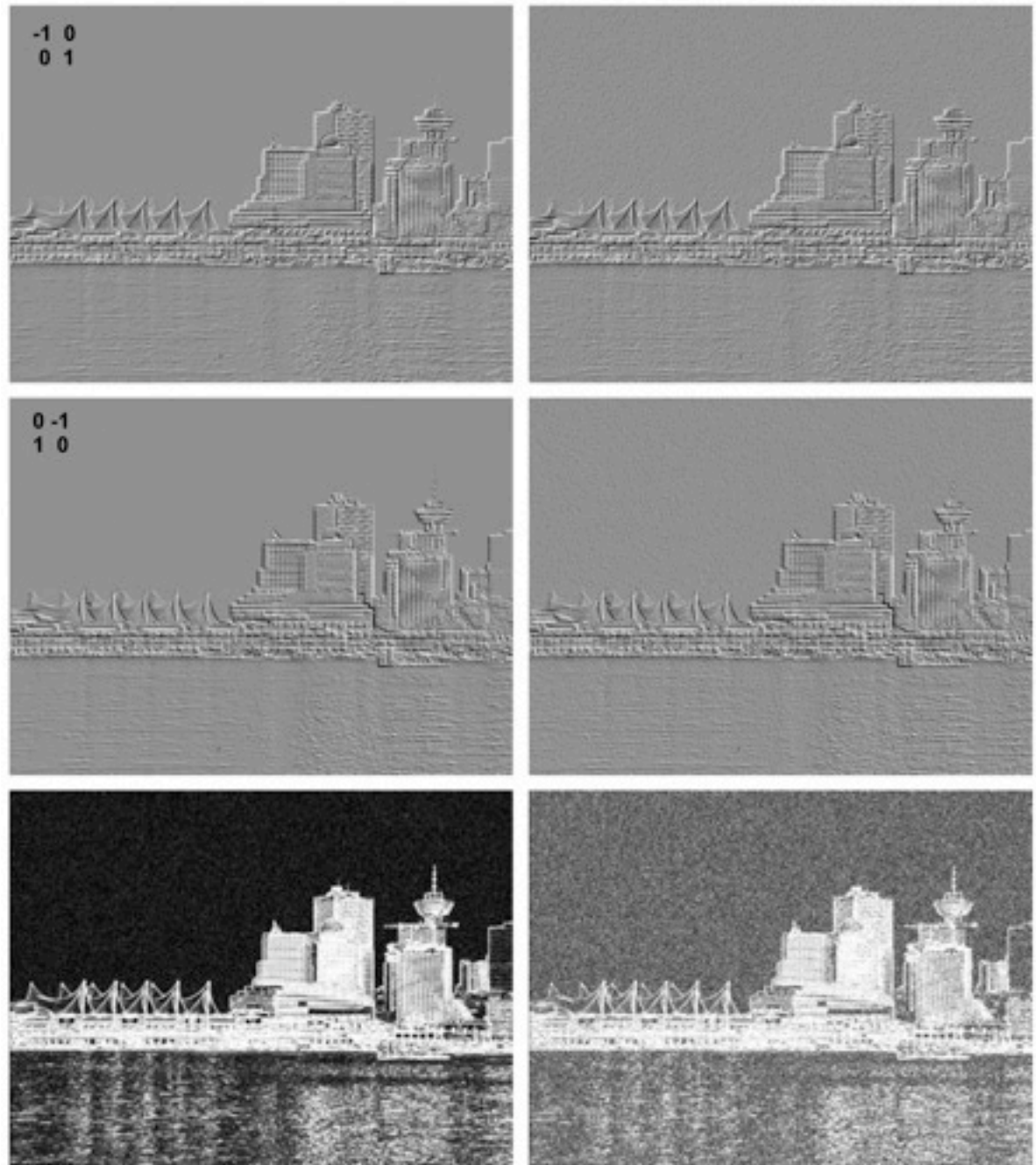


Betrag : $\text{sqrt}(G_x^2 + G_y^2)$
Richtung: $\text{tan}^{-1}(G_y/G_x)$



Robert's Gradient

Die beiden Komponenten des Robert's Gradient sowie der Betrag des Gradienten (unten) auf einem rauscharmen (links) und einem verrauschten (rechts) Bild



Rauschen und Kanten

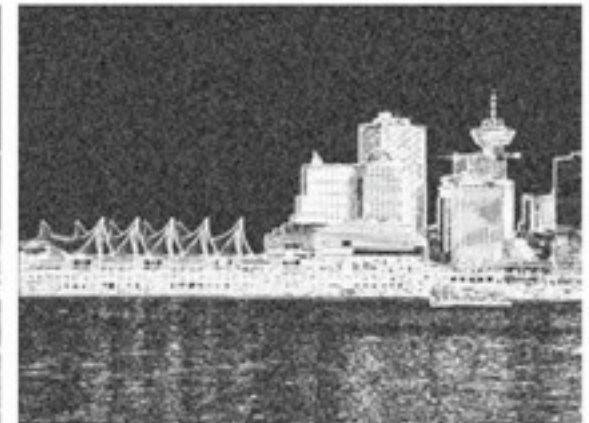
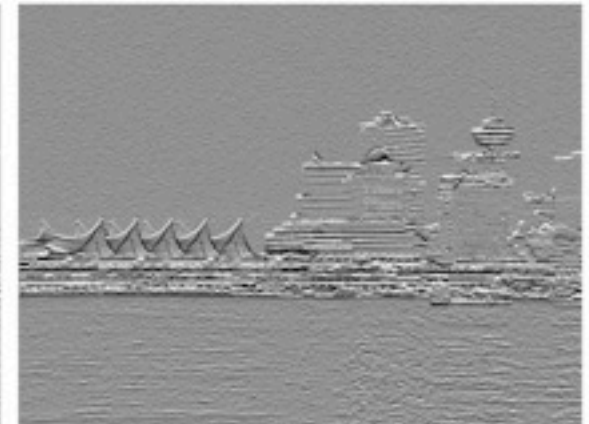
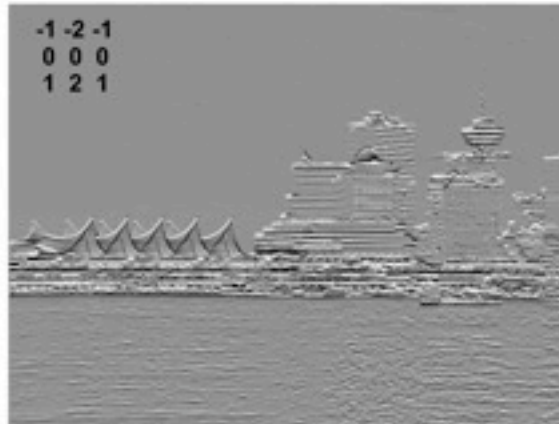
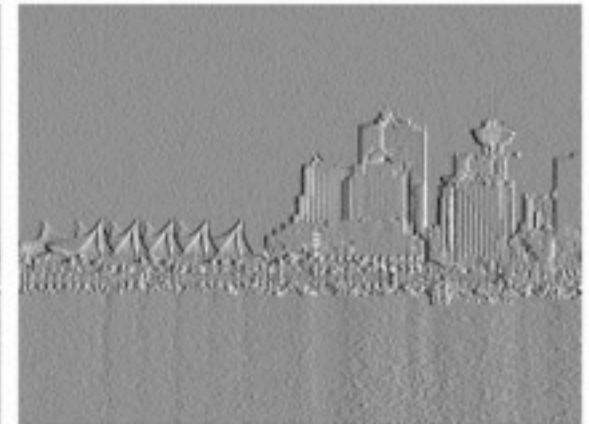
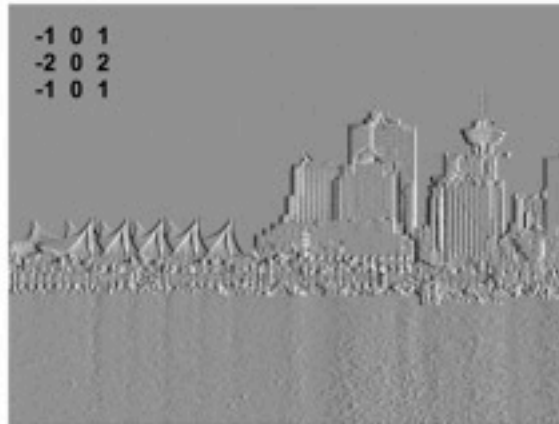


Hintergrundartefakte (Rauschen) werden verstärkt.

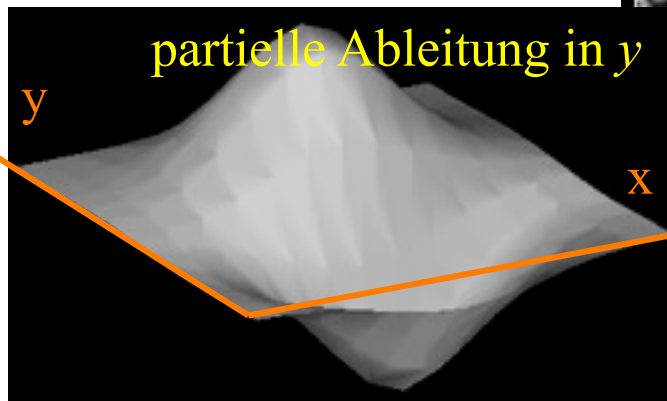
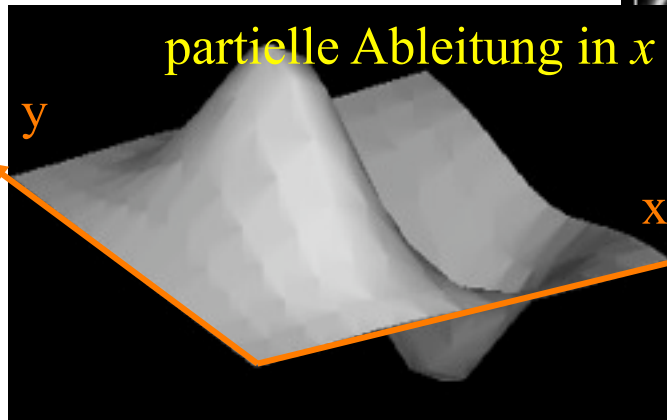
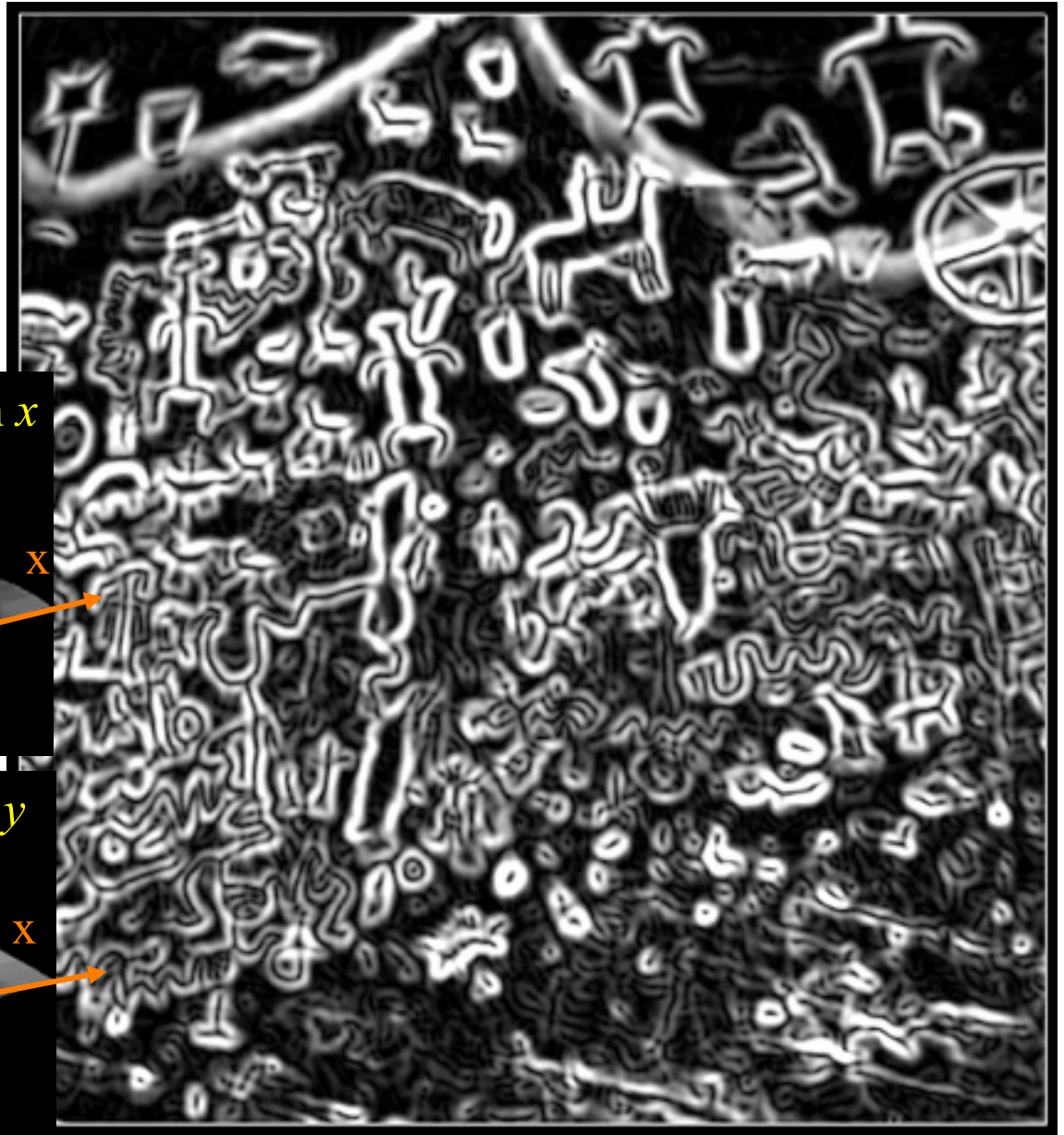
Sobel Operator

$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ und $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

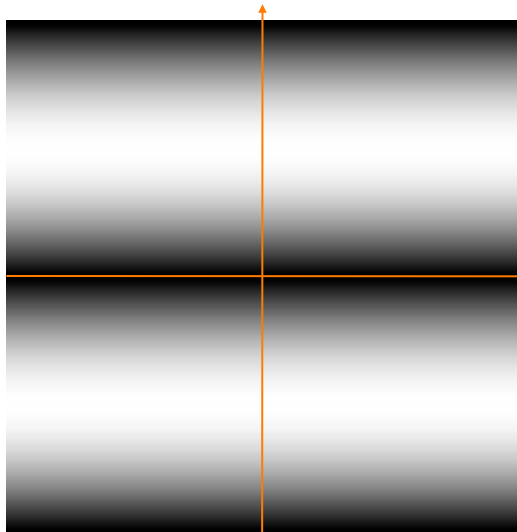
Faltungskern
kombiniert
Glättung und
Differenzierung in
einem Operator.



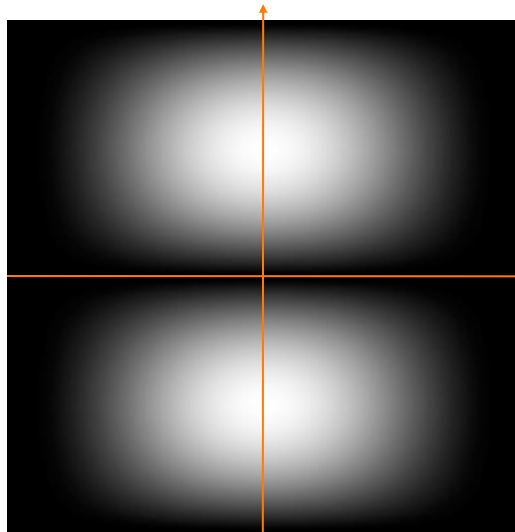
Ableitungen der Gaussfunktion



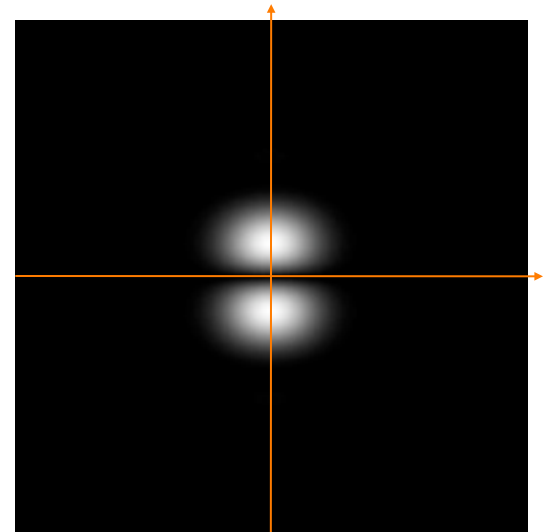
Vergleich der Differenzfilter im Frequenzbereich



Einfache Differenz



Sobelfilter



Approximation der
ersten Ableitung der
Gaußfunktion

Sobel vs. Gauß



3x3 Sobel

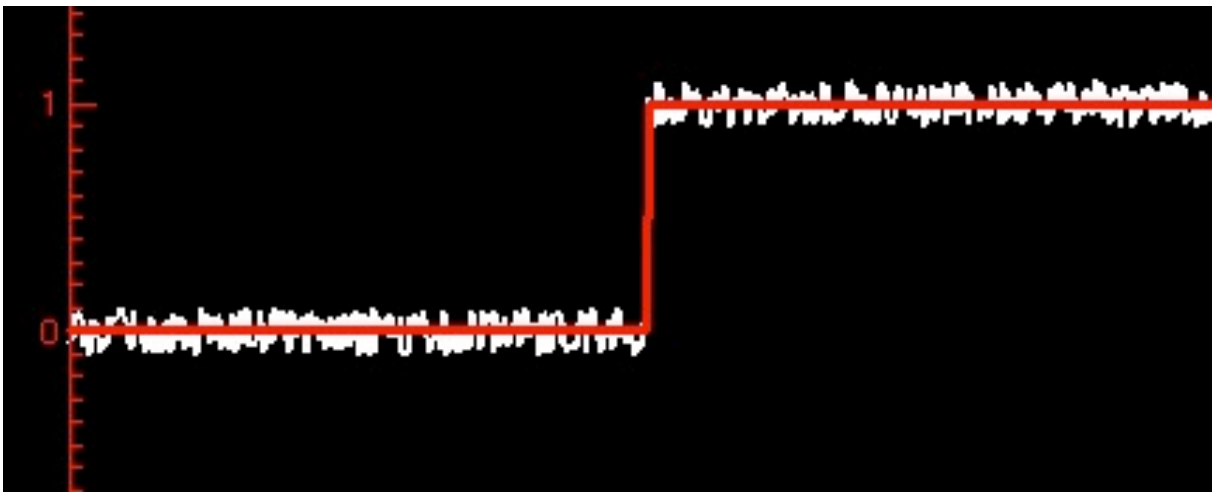


13x13 Gauß

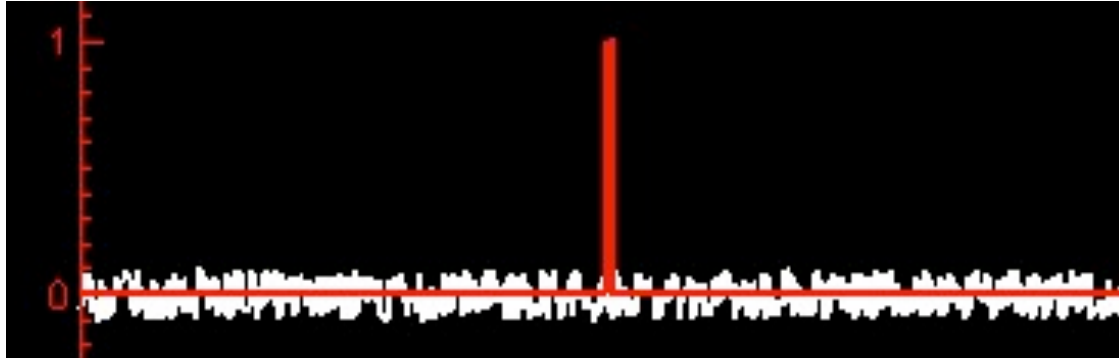


Optimale Kantenfilterung

- Annahme: Bild ist homogen mit Ausnahme von Funktionswertänderungen an idealen Kanten.
- Repräsentation im Frequenzraum
 - homogener Bereich mit Grauwert g :
$$F(0,0) = g \text{ und } F(u,v)=0 \text{ für } u,v \neq 0$$
 - Kante: $F(0,0) = 0 \text{ und } F(u,v) \neq 0 \text{ für } u,v \neq 0$
 - Rauschen: $F(u,v) \neq 0$



Kanten und Rauschen

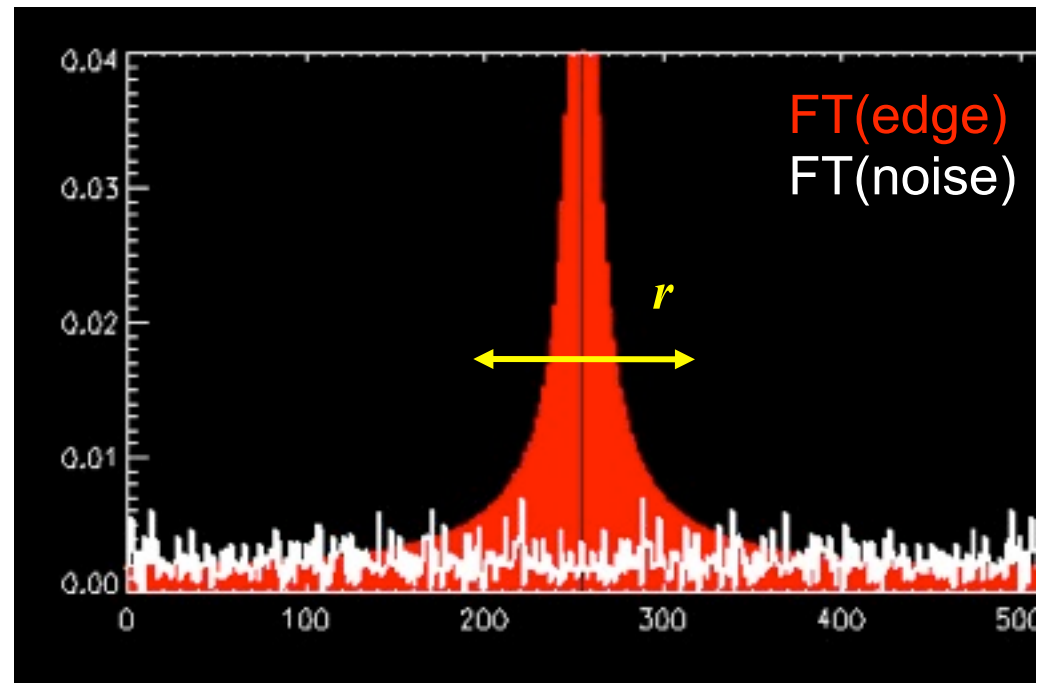


Doppelkante: beide Kantenorte müssen unterscheidbar sein.

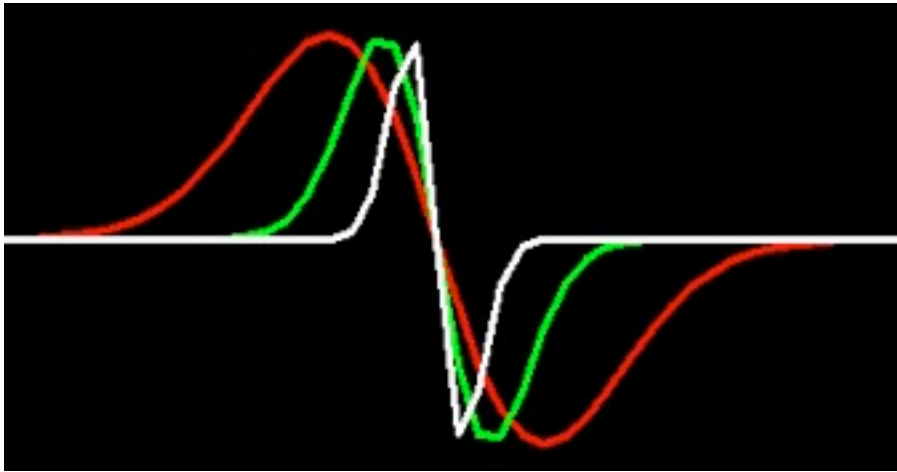
Filter suchen mit:

$$F(0,0)=0$$

$$F(u,v)=w, \text{ falls } (u^2+v^2)<r^2$$
$$0, \text{ sonst}$$

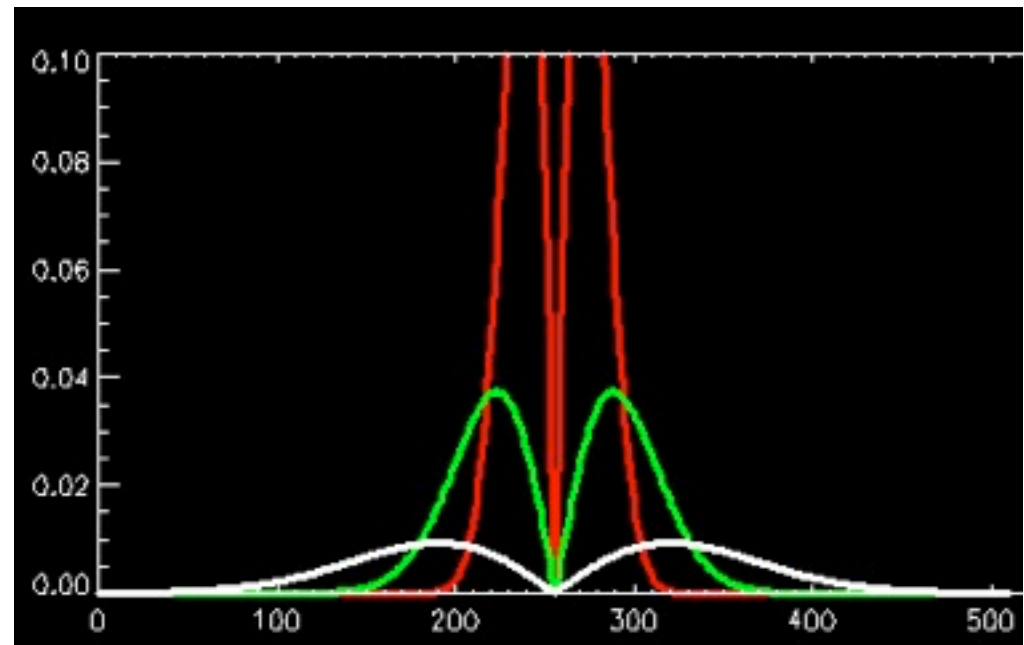


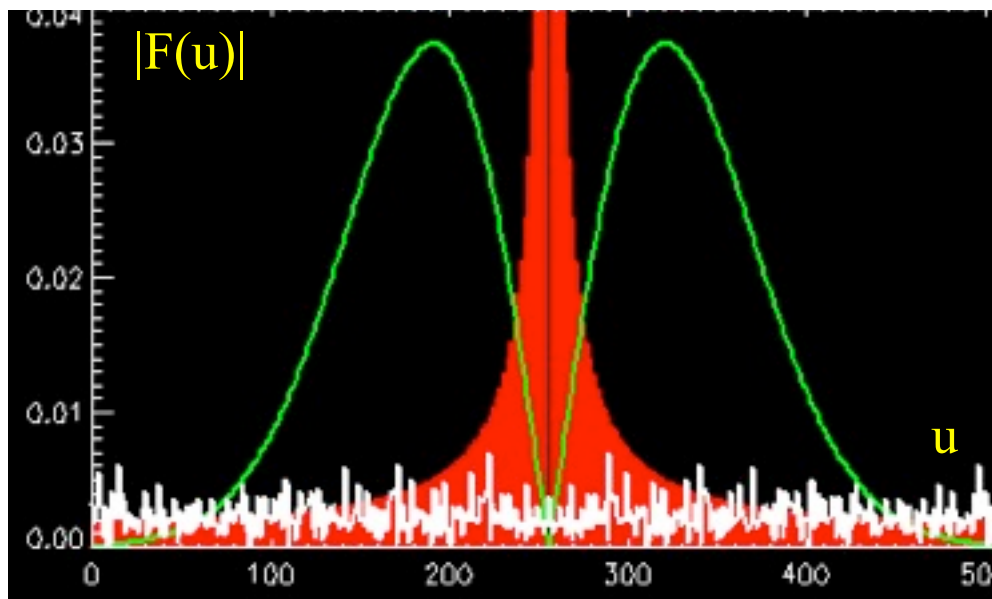
Filter



Gaußfilter mit unterschiedlicher Varianz (im Ortsraum).

Repräsentation
im Frequenzraum

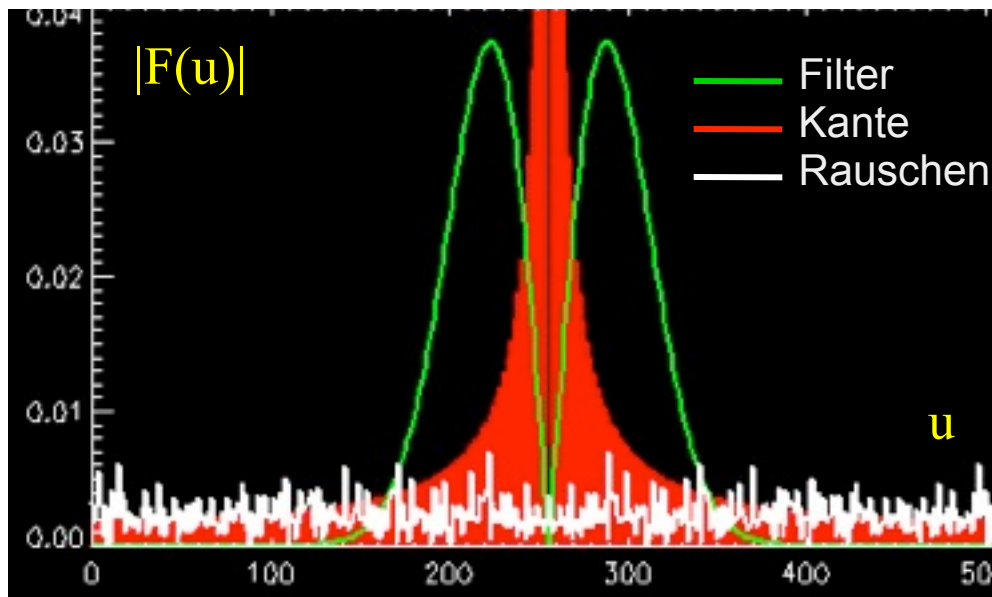




$$\sigma=1.0$$

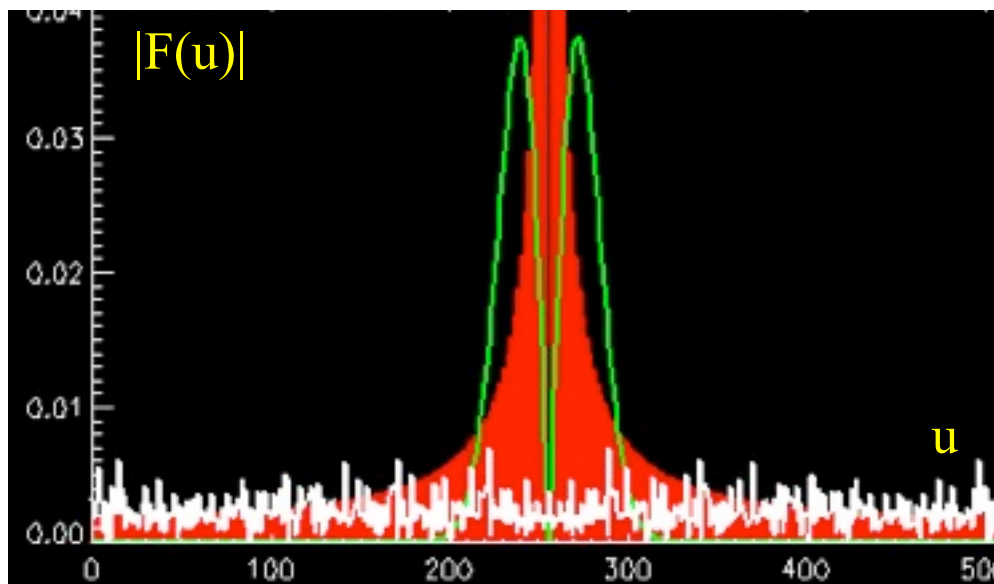
Filter unterdrückt
Rauschen ungenügend

Filter



$$\sigma=2.0$$

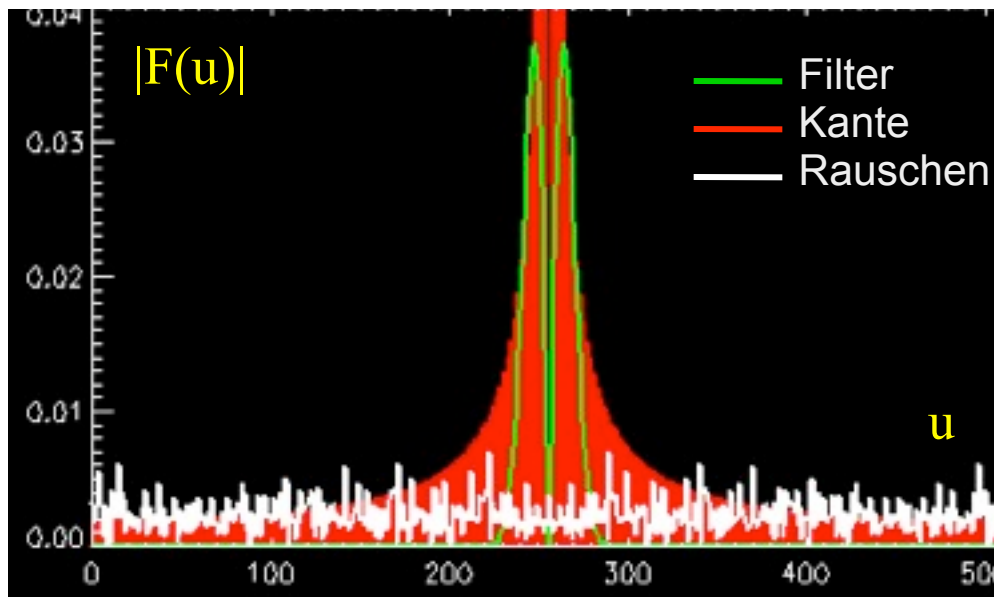
Filter ist nahezu optimal.



$\sigma=4.0$

Filter ist nahezu optimal.

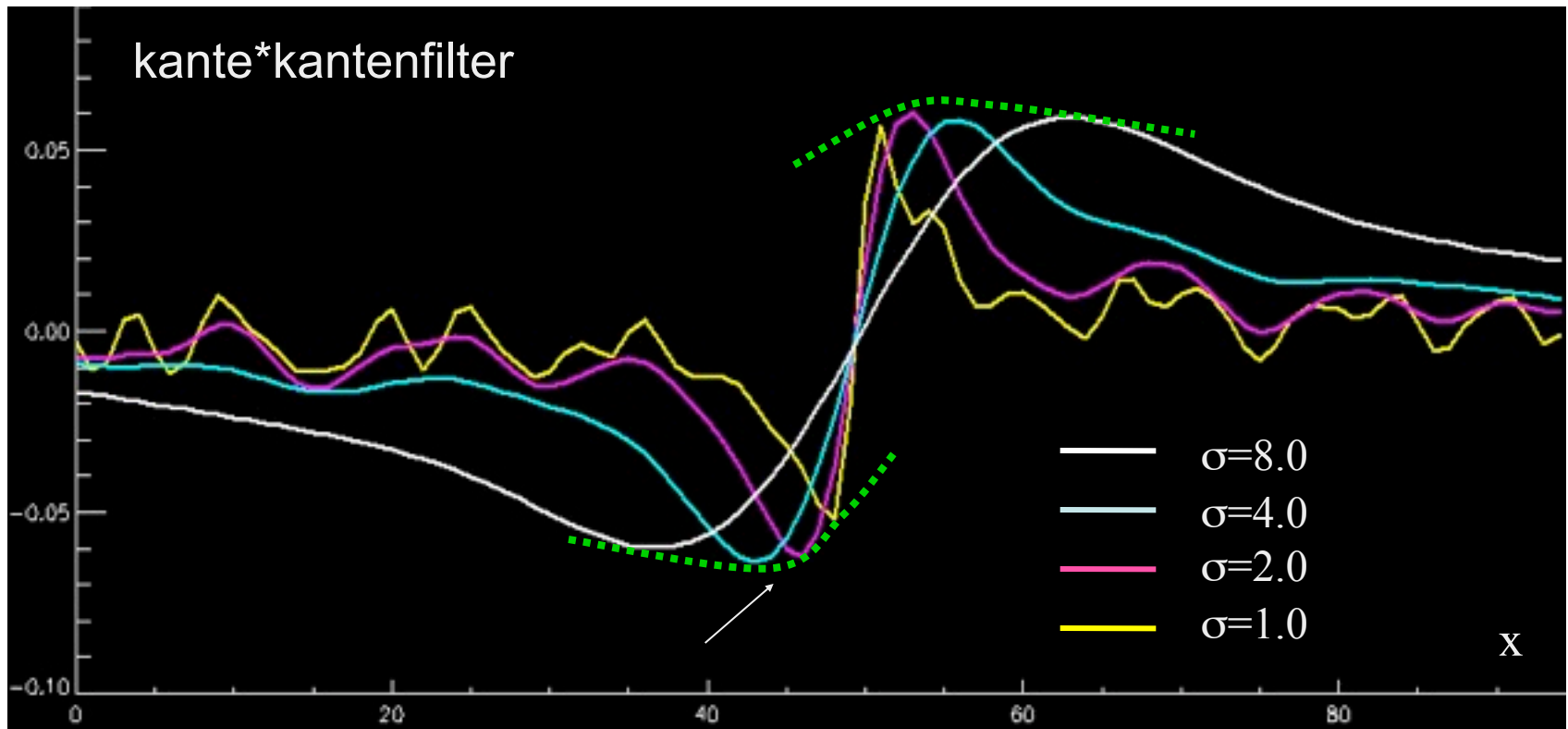
Filter



$\sigma=8.0$

Filter unterdrückt Kante.

Filterresultat



Zweidimensionales optimales Filter

- Rauschunterdrückung und Kanten hervorhebung gemäß Frequenzraumcharakteristik.
- Filterung nur entlang größter Steigung
- Canny Edge Enhancement:
 - Richtung der Kante schätzen.
 - Sequenz von Filtern mit unterschiedlicher Glättungscharakteristik in Gradientenrichtung berechnen.

Kompassfilter

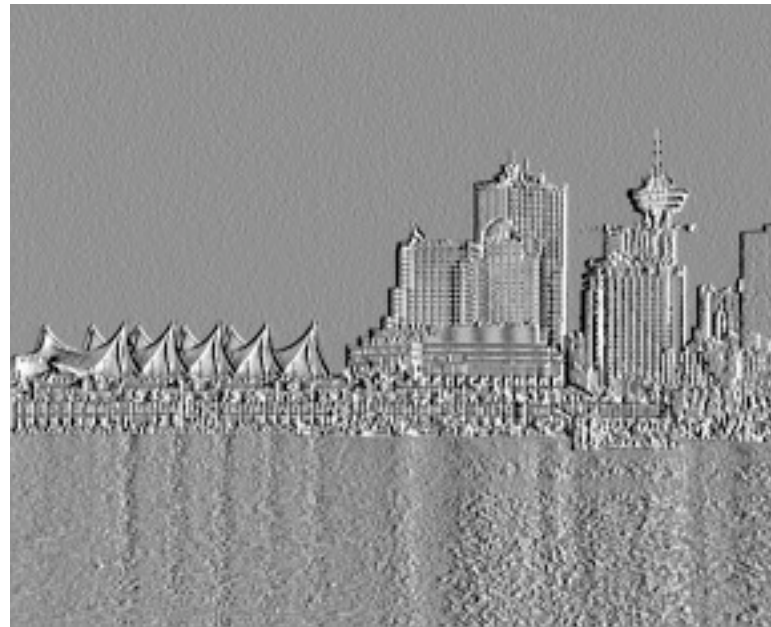
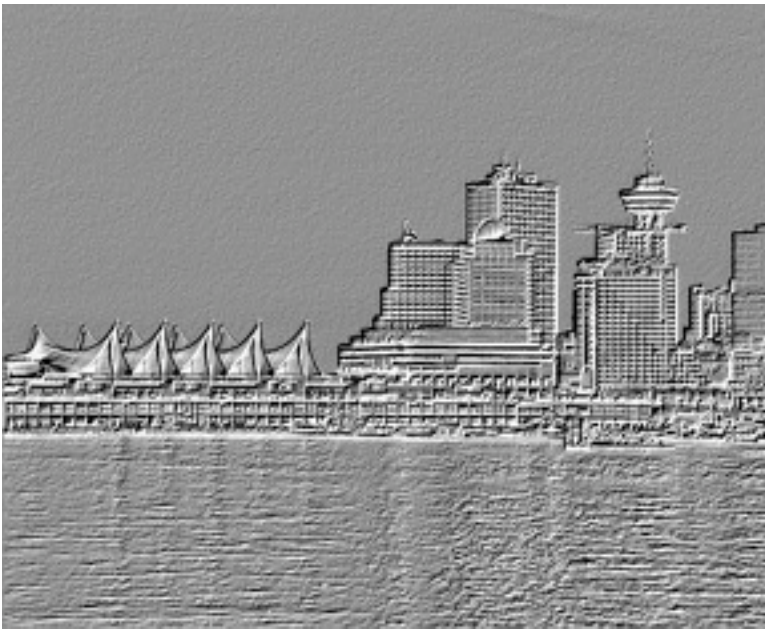
- Dienen der Hervorhebung von Kanten in einer bestimmten Richtung.
- Prewitt-Operator

$$h_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} \quad h_2 = \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix} \quad h_3 = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad h_8 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix}$$

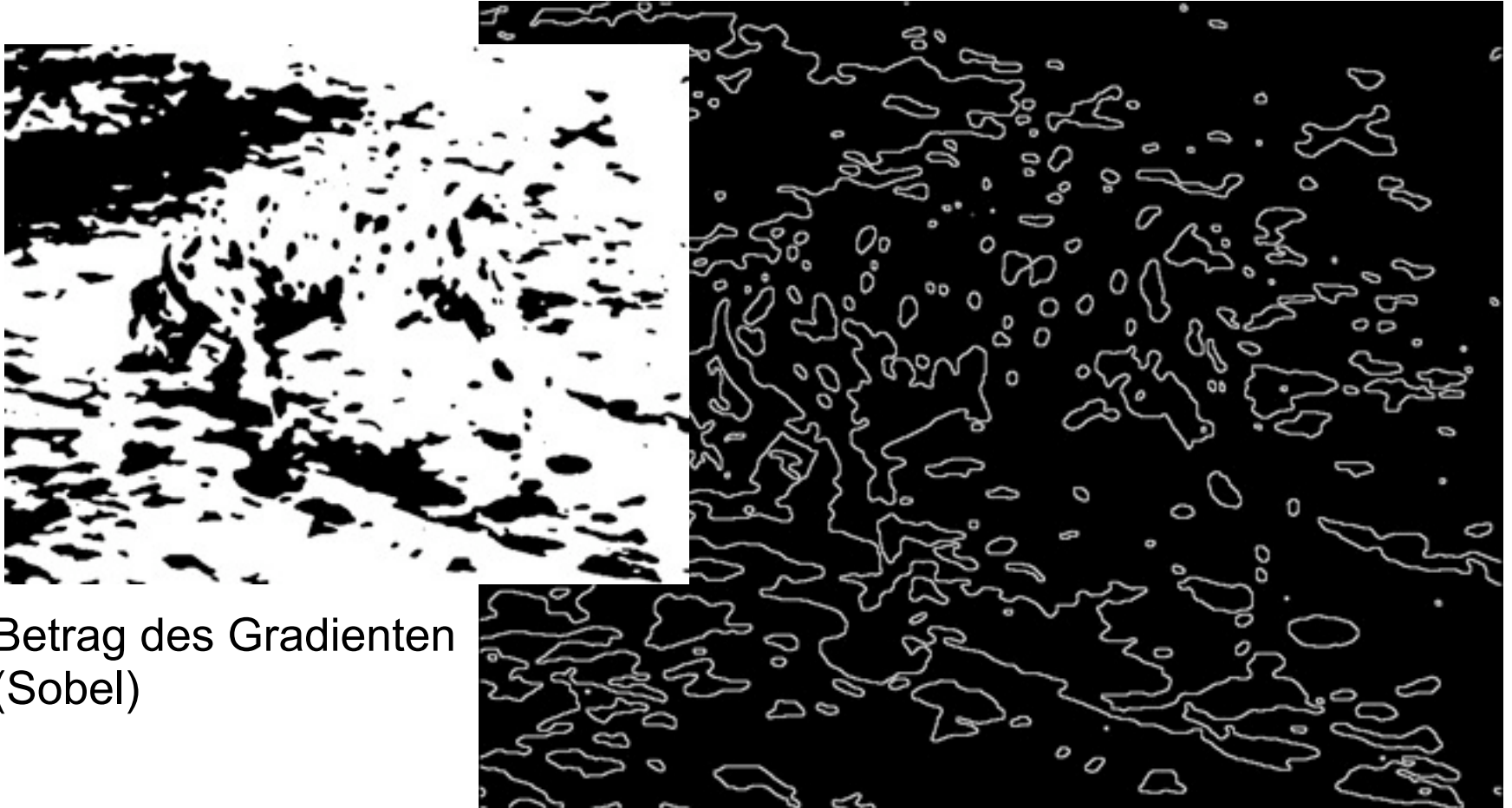
- Kirsch-Operator

$$h_1 = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{pmatrix} \quad h_2 = \begin{pmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{pmatrix} \quad h_3 = \begin{pmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{pmatrix} \quad h_8 = \begin{pmatrix} 3 & 3 & 3 \\ 3 & 0 & -5 \\ 3 & -5 & -5 \end{pmatrix}$$

Kirschoperator

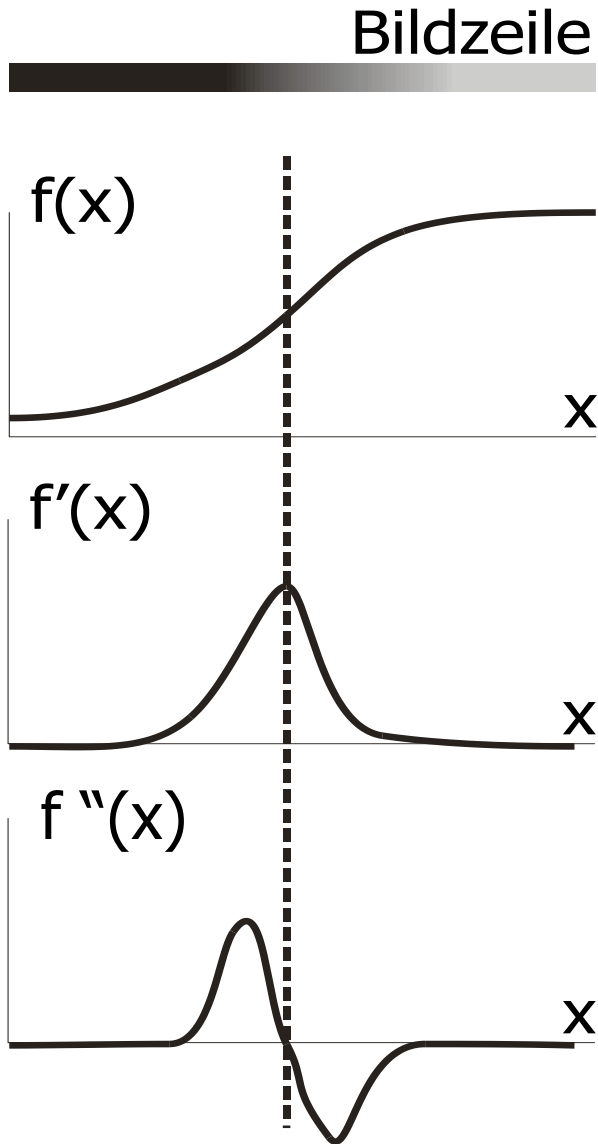


Ecken und Regionengrenzen



Betrag des Gradienten
(Sobel)

Wo ist die Regionengrenze?



Vorzeichenwechsel ist leichter zu erkennen, als ein Minimum oder Maximum.

- Gradient (Länge) als Kennzeichen für die Wichtigkeit einer Kante
- zweite Ableitung für den Ort der Kante (Nulldurchgang)

Operatoren zur Berechnung der zweiten Ableitung:

- Laplace Filter
- Marr-Hildreth Filter (LoG Filter, Mexican Hat)
- DoG (Difference of Gaussians)

Laplace Funktion

- Summe der partiellen zweiten Ableitungen: $\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$
- Nulldurchgänge der Laplacefunktion produzieren zusammenhängende Kurven entlang von Bildkanten.
- Approximation durch Kombination einer doppelten Differenzbildung in x- und y-Richtung,
 - z.B. Faltung eines Differenzoperators $[-1 \ 1]$ mit sich selbst: $[-1 \ 2 \ -1]$
 - ergibt für $[-1 \ 1] * [-1 \ 1] + [-1 \ 1]^T * [-1 \ 1]^T$:

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

- Summe **aller** partiellen Ableitungen:

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} + \frac{\partial^2 f(x, y)}{\partial x \partial y} + \frac{\partial^2 f(x, y)}{\partial y \partial x} : \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

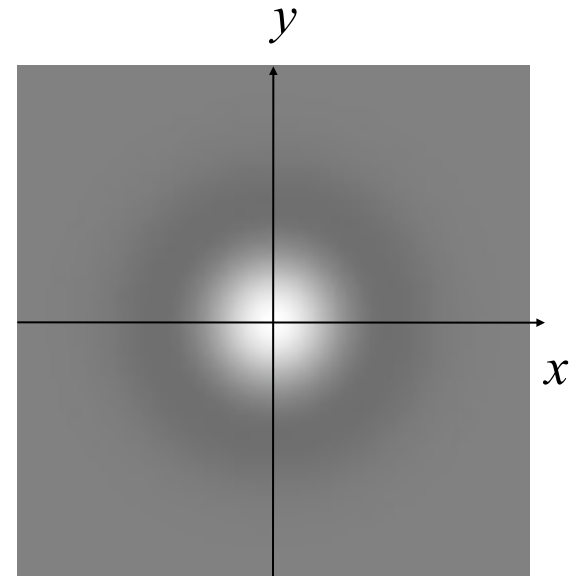
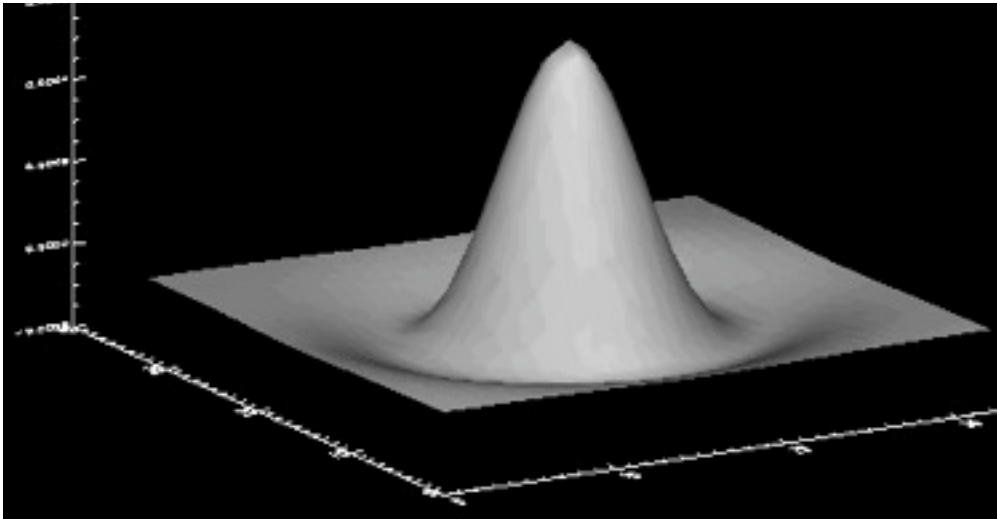
Laplace-Operator



- Nulldurchgänge des Laplace-Operators. Dunkle Pixel bezeichnen negative, helle Pixel bezeichnen positive Werte.

Marr-Hildreth-Filter = LoG-Filter

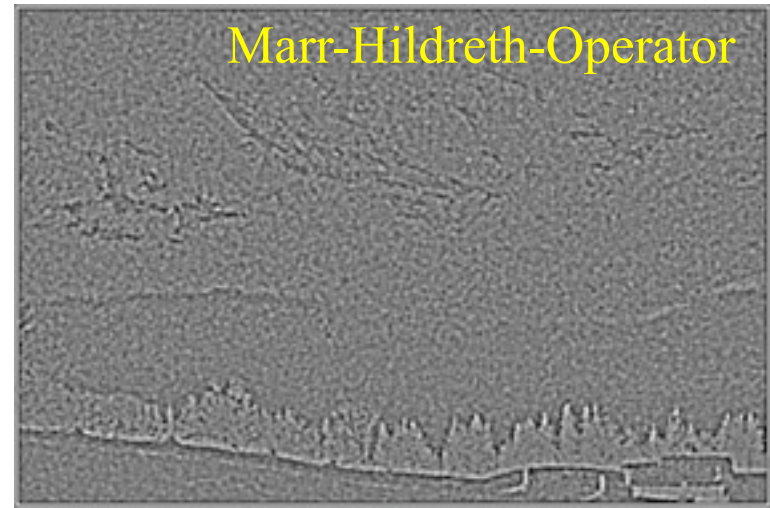
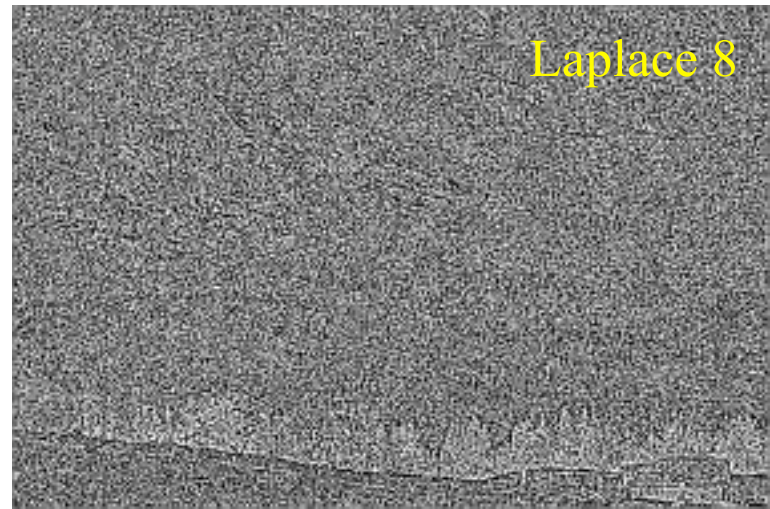
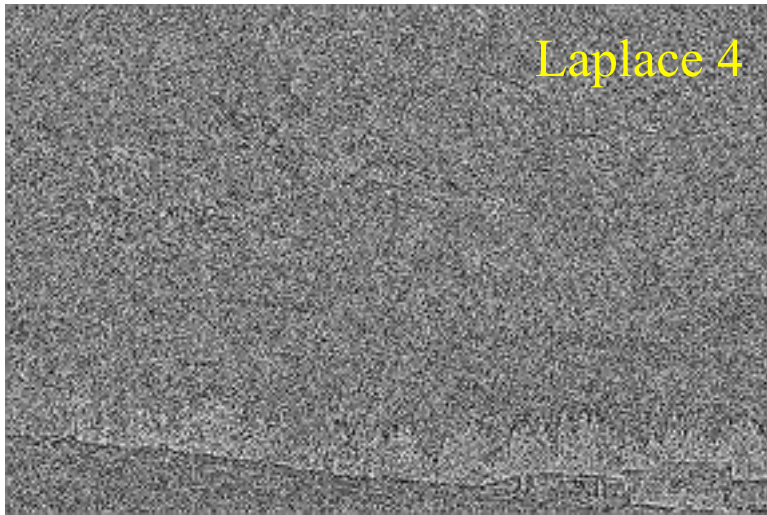
LoG-Filter: Laplacian-of-Gaussian,
d.h. der **Faltung mit dem Laplacefilter** geht eine
Glättung mit einer **Gaußfunktion** voraus.



$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2} \right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2} \right)$$



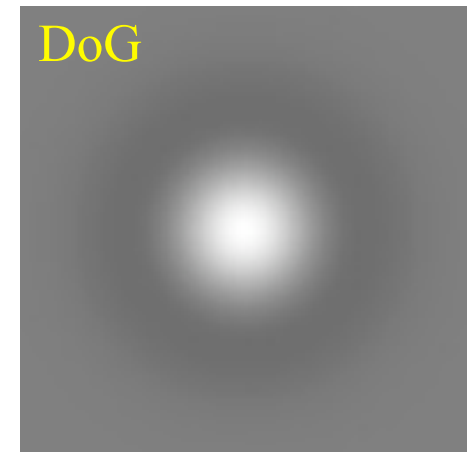
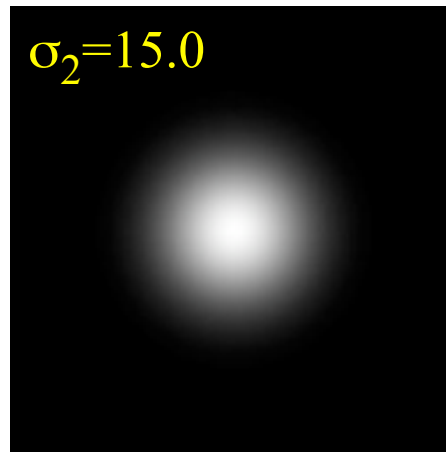
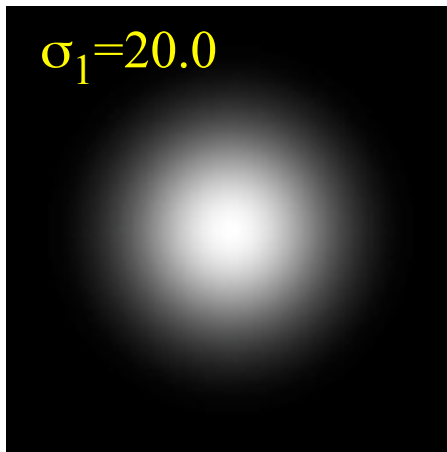
Auch genannt: „Mexican hat“ filter



DoG (Difference of Gaussians)

Subtraktion zweier Gaußfilter mit unterschiedlicher Varianz σ^2 :

$$\text{DoG}(x,y,\sigma_1,\sigma_2) = [\sqrt{2\pi}\cdot\sigma_1]^{-1} \cdot \exp(-(x^2+y^2)/2\sigma_1^2) \\ - [\sqrt{2\pi}\cdot\sigma_2]^{-1} \cdot \exp(-(x^2+y^2)/2\sigma_2^2)$$



DoG und LoG-Filter sind in ihrer Wirkungsweise vergleichbar.

..nochmal kurz zusammengefasst:

- Rauschen und Kanten haben im Frequenzbereich ähnliche Attribute.
- Kanten hervorhebung in Orts- und Frequenzraum, Kompassfilter
- Kanteneigenschaften
- Zweite (partielle Ableitung): Laplace-Filter, LoG-Filter,...

Nichtlineare Filterung




- **Rauschen** und **Kanten** haben im Frequenzbereich ähnliche Attribute.
- Ist ein nichtlineares Filter denkbar, das für Rauschen und Kanteneigenschaften unterschiedlich sensitiv ist?
 - Rauschen sind räumlich gleichverteilte Grauwertvariationen.
 - Grauwertvariationen an Kanten sind nicht räumlich gleichverteilt.
 - Filter muss diesen Unterschied berücksichtigen.

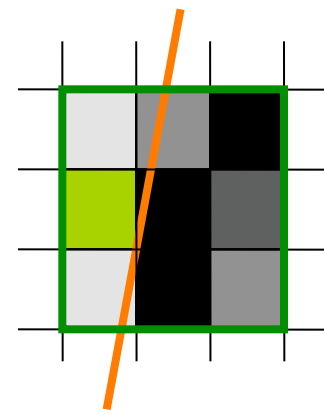
Rangordnungsfilter

- **Vorgehen:**
 - Sortierung der Elemente in einer Filtermaske
 - Auswahl des an einer bestimmten Stelle einsortierten Werts
 - Eintragung des ausgewählten Werts in die zentrale Position
- **Eigenschaften:**
 - Es entstehen keine neuen Werte
 - Filter ist nichtlinear, nicht kommutativ, nicht assoziativ

Gebräuchlichstes
Rangordnungsfilter ist das
Medianfilter

26 3.	132 8.	112 5.
25 2.	102 4.	142 9.
17 1.	122 7.	117 6.

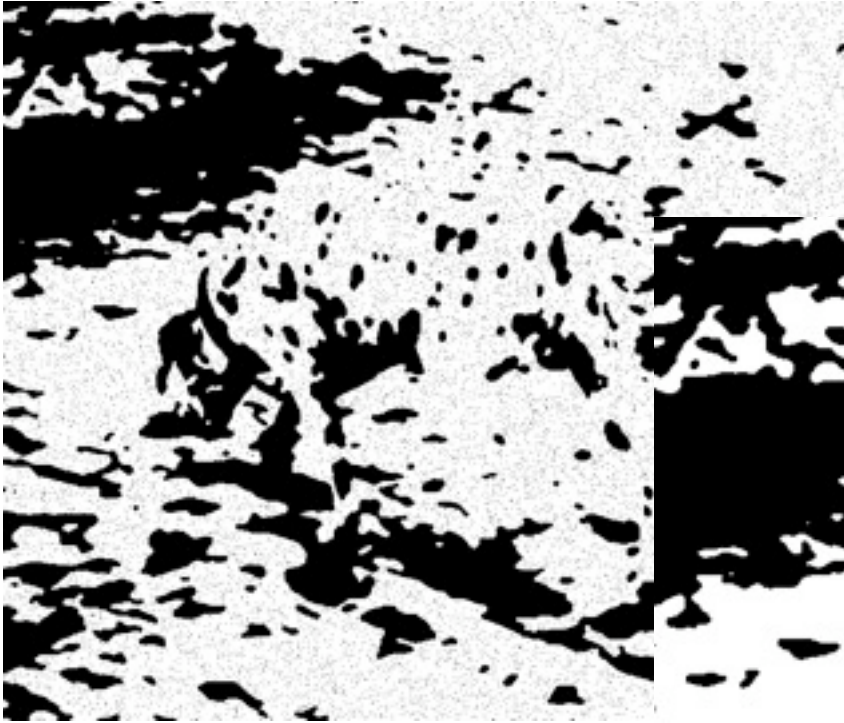
-  erster Rang
-  mittlerer Rang (Median)
-  letzter Rang



Medianfilter

- Annahmen:
 - Medianfilter ist quadratisch mit ungerader Seitenlänge.
 - Kantensignal ist größer als das Rauschsignal
 - Kante im Filterbereich verläuft in diesem Bereich (nahezu) gerade.
 - Grauwert ist (nahezu) konstant in einer Umgebung von Größe des Filters.
- Kante verläuft durch Filterbereich:
 - Der Mittelpunkt des Filters liegt auf der gleichen Seite der Kante wie die Mehrzahl der Pixel (z.B. rechts von der Kante).
 - Pixel von dieser Seite (z.B. rechts) wird selektiert (**kantenerhaltend**).
- Keine Kante im Filterbereich:
 - Median nähert sich dem Erwartungswert mit Anzahl der Stichproben (**rauschunterdrückend**).

Medianfilter

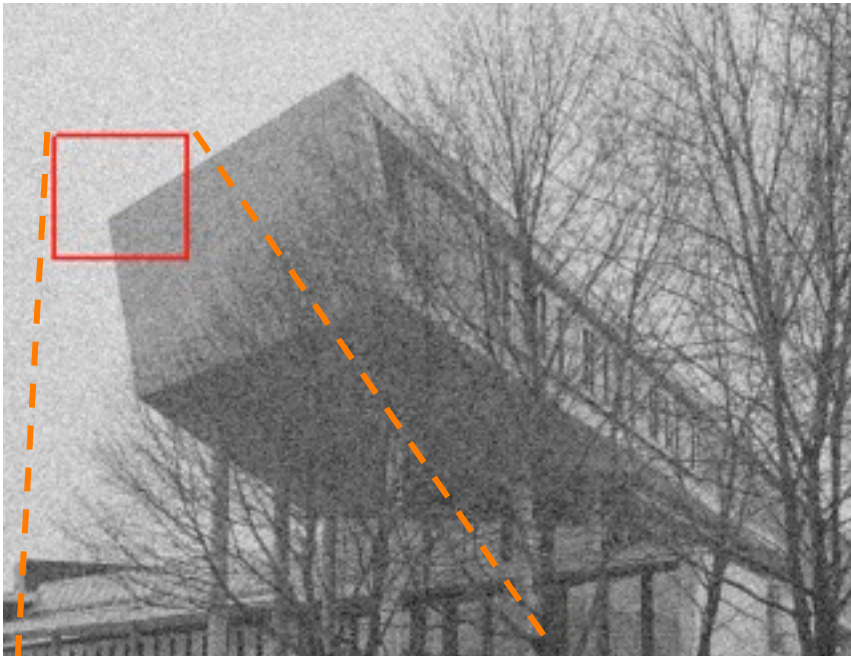


Median zur Entfernung
von Salt&Pepper-
Rauschen in einer 3x3
Umgebung.

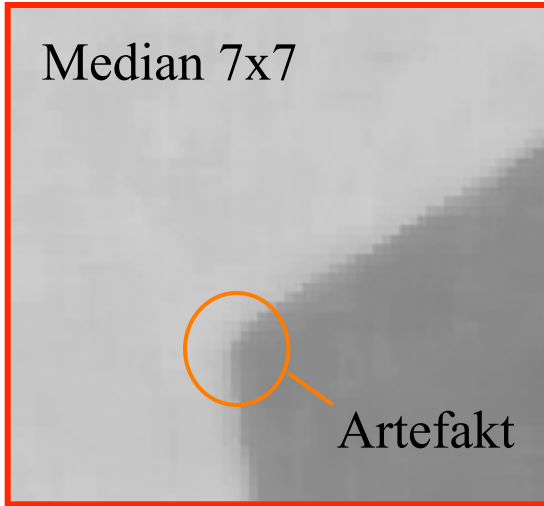
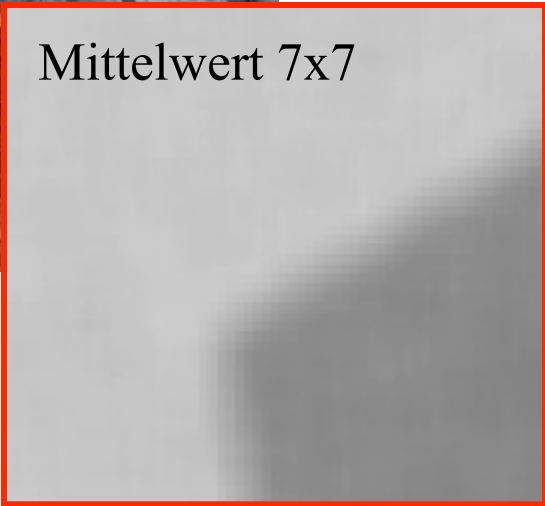
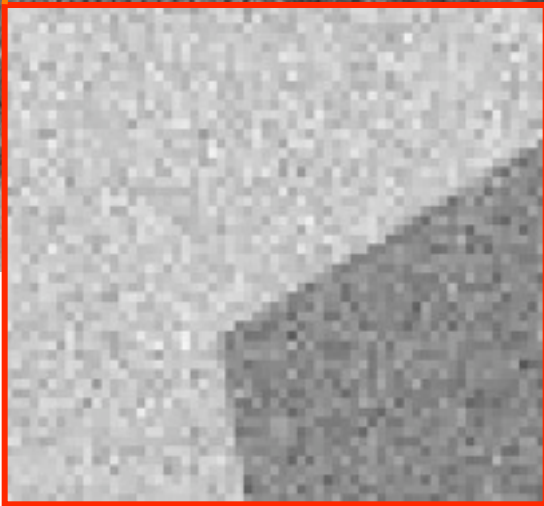
Medianfilter



- Durch Medianfilterung (rechts) kann Impulsrauschen im Gegensatz zur Mittelwertfilterung (Mitte) beseitigt werden.



Vergleich Median vs. Mittelwert

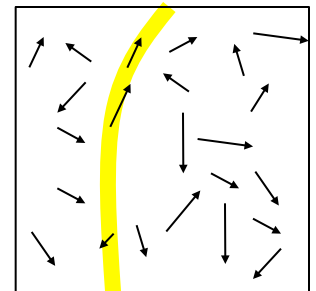
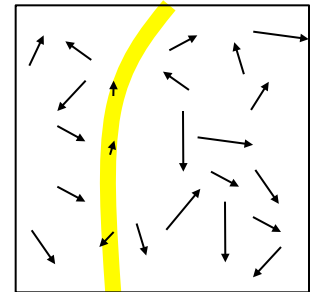
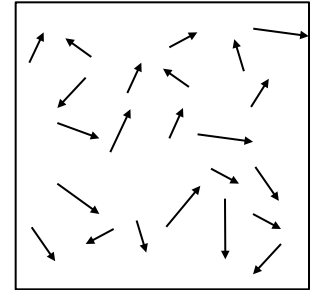


Diffusionsfilter

- Diffusionsprozess ist Modell für die Filterung
- Diffusion:
 - Flüssigkeit mit lokal variierender Dichte
 - Lokaler Ausgleich in Abhängigkeit vom Dichtegradienten
- Übertragung auf ein Bild:
 - Dichte ist die Bildfunktion
 - Dichtegradient ist Grauwertgradient

Diffusion

- Homogene Diffusion: Szene besteht aus einem einzigen Material und die Diffusion ist proportional zum Dichtegradienten dieses Materials.
- Inhomogene Diffusion: Szene enthält Barrierematerialien, wo die Diffusion erschwert wird.
- Anisotrope, inhomogene Diffusion: Diffusion an Barrierematerialien ist richtungsabhängig.



Fick'sches Gesetz

$$\vec{j}(x, y) = -\mathbf{D}(x, y) \times \nabla u(x, y) = -\mathbf{D}(x, y) \times \begin{pmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{pmatrix}$$

\mathbf{D} – Diffusionstensor

∇u - Dichtegradient

- Der Diffusionstensor spezifiziert die Art des Fluss \vec{j} in Abhängigkeit vom Dichtegradient ∇u .

- Die Konzentrationsveränderung durch \vec{j} ist:
$$\frac{\partial u(x, y, t)}{\partial t} = -\operatorname{div} \vec{j}(x, y, t)$$

- Der Divergenzoperator ist definiert durch:

$$\operatorname{div} \vec{j} = \left(\frac{\partial j_x(x, y, t)}{\partial x} + \frac{\partial j_y(x, y, t)}{\partial y} \right)$$

Fick'sches Gesetz (Namensherkunft, Quelle: [Wikipedia](#))

- **Adolf Eugen Fick** (* [3. September 1829](#) in [Kassel](#), † [21. August 1901](#) in [Blankenberghe, Flandern](#)) war ein [deutscher Physiologe](#).
- Er stellte [1855](#) die nach ihm benannten Gesetze der [Diffusion](#) auf, die eine wesentliche Grundlage zur Entwicklung der [Kontaktlinse](#) bildeten.
- Die Adolf-Fick-Stiftung vergibt alljährlich den Adolf-Fick-Preis an einen herausragenden [Physiologen](#). Der Preis wird als die wichtigste Auszeichnung auf dem Gebiet der deutschsprachigen Physiologie angesehen.



Berechnung der Diffusion

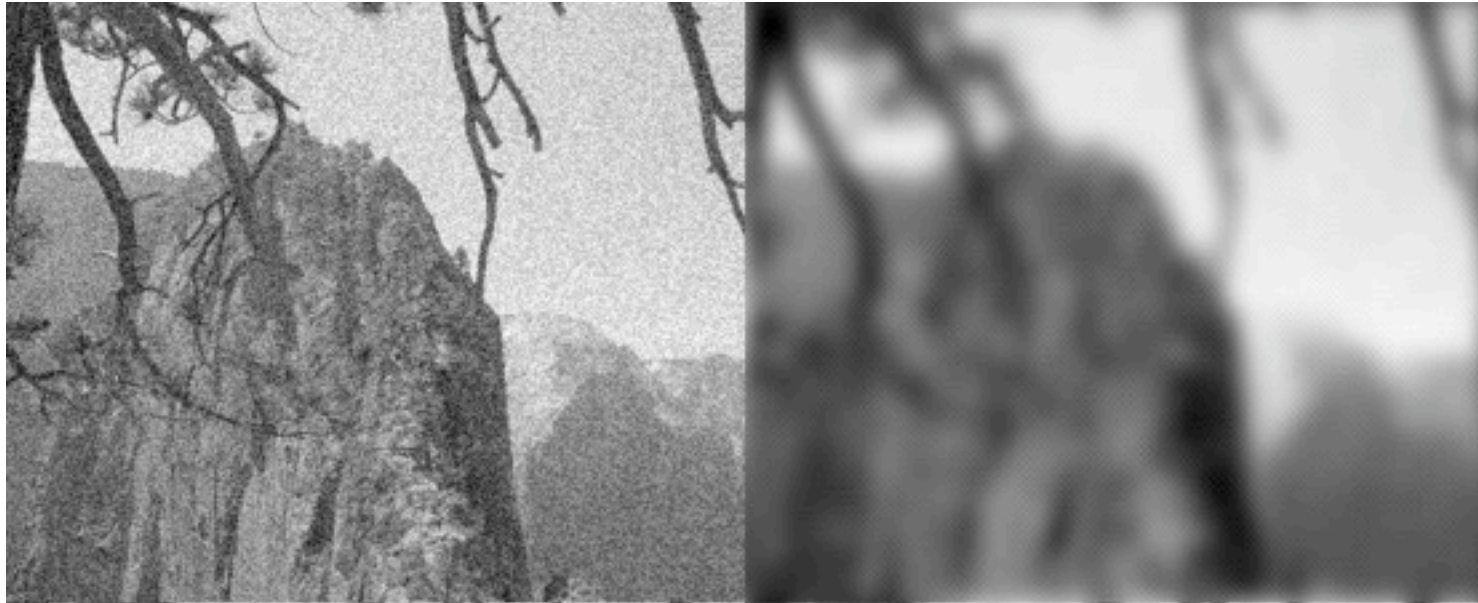
- Die Belegung des Diffusionstensors gibt an, ob es sich um homogene, inhomogene oder anisotrope, inhomogene Diffusion handelt.
- Die Differentialgleichung lässt sich im Allgemeinen nicht analytisch lösen.
- Iterative Approximation:

$$u(x, y, t_{i+1}) = u(x, y, t_i) + (t_{i+1} - t_i) \frac{\partial u(x, y, t_i)}{\partial t}.$$

Die Intervallgröße muss sorgfältig gewählt werden, damit der Schätzfehler durch die Approximation nicht zu groß wird.

Homogene Diffusion

$$\mathbf{D} = \begin{pmatrix} \varepsilon & 0 \\ 0 & \varepsilon \end{pmatrix} \quad \text{und} \quad \text{div } \vec{j} = \left(\varepsilon \frac{\partial f(x, y, t)}{\partial x} + \varepsilon \frac{\partial f(x, y, t)}{\partial y} \right) \vec{j}$$



Inhomogene Diffusion

- **Ziel:** keine (wenig) Diffusion an Kanten.
- **Problem:** Kantenverlauf ist nicht bekannt.
- **Annahme:** Kante = hoher Gradient, d.h. Diffusionstensor muss von der Gradientenlänge abhängig sein.

$$\mathbf{D} = \begin{pmatrix} \varepsilon \left(\|\nabla u\|^2 \right) & 0 \\ 0 & \varepsilon \left(\|\nabla u\|^2 \right) \end{pmatrix} \quad \text{mit} \quad \varepsilon \left(\|\nabla u\|^2 \right) = \varepsilon_0 \frac{\lambda^2}{\|\nabla u\|^2 + \lambda^2}$$

Der Parameter λ steuert, wie stark der Gradient die Diffusion beeinflusst.

Inhomogene Diffusion

Rauschen an
den Kanten
bleibt erhalten.

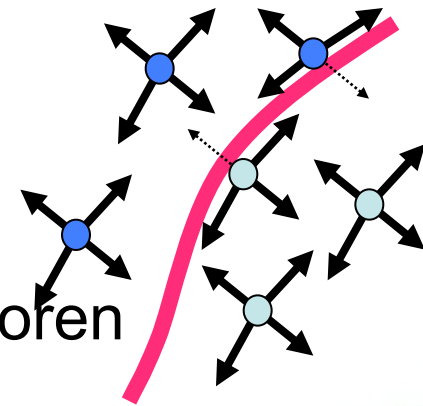


Anisotrope, inhomogene Diffusion

- Ziel: keine Diffusion über Kanten hinaus, aber entlang von Kanten.
- Diffusionstensor muss Kantenrichtung mit berücksichtigen.
- Diffusionstensor kann zerlegt werden in:

$$\mathbf{D} = \begin{pmatrix} e_{1,1} & e_{2,1} \\ e_{1,2} & e_{2,2} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} e_{1,1} & e_{1,2} \\ e_{2,1} & e_{2,2} \end{pmatrix}$$

- Adäquate Bestimmung der Eigenvektoren und Eigenwerte löst das Problem.



Eigenvektoren für die anisotrope Diffusion

$$\begin{aligned} \begin{pmatrix} e_{1,1} & e_{1,2} \end{pmatrix} &= \frac{\nabla u}{\|\nabla u\|}, & \lambda_1 &= \varepsilon \left(\|\nabla u\|^2 \right) \\ \begin{pmatrix} e_{2,1} & e_{2,2} \end{pmatrix} &= \begin{pmatrix} e_{1,2} & -e_{1,1} \end{pmatrix} & \lambda_2 &= 1. \end{aligned} \quad \text{mit} \quad \varepsilon \left(\|\nabla u\|^2 \right) = \varepsilon_0 \frac{\lambda^2}{\|\nabla u\|^2 + \lambda^2}$$

- Erster Eigenvektor in Richtung des Gradienten und Eigenwert sinkt mit Gradientenlänge: keine Diffusion über Kanten hinweg.
- Zweiter Eigenvektor parallel zur Kante und Eigenwert konstant: Diffusion parallel zur Kante ist von der Kantenstärke unabhängig.

Anisotrope Diffusion

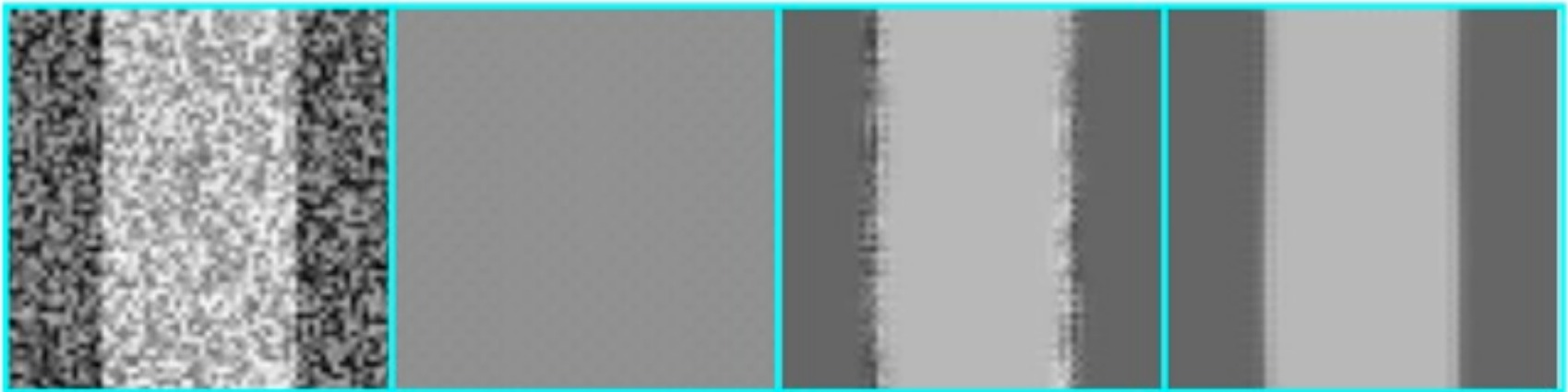


Anistrophe Diffusion



Diffusion im Vergleich

- Homogene Diffusion: vollständige Homogenisierung
- Inhomogene Diffusion
 - Isotrop: Erhält Kanten, aber dort auch Rauschen.
 - Anisotrop: unterdrückt Rauschen auch an Kanten.
- Modell für Dichtegradient und Kante sind gleich
- Es gibt keine 100%-ige Kantenentscheidung: Jede Diffusion führt nach genügend vielen Iterationen zur vollständigen Homogenisierung



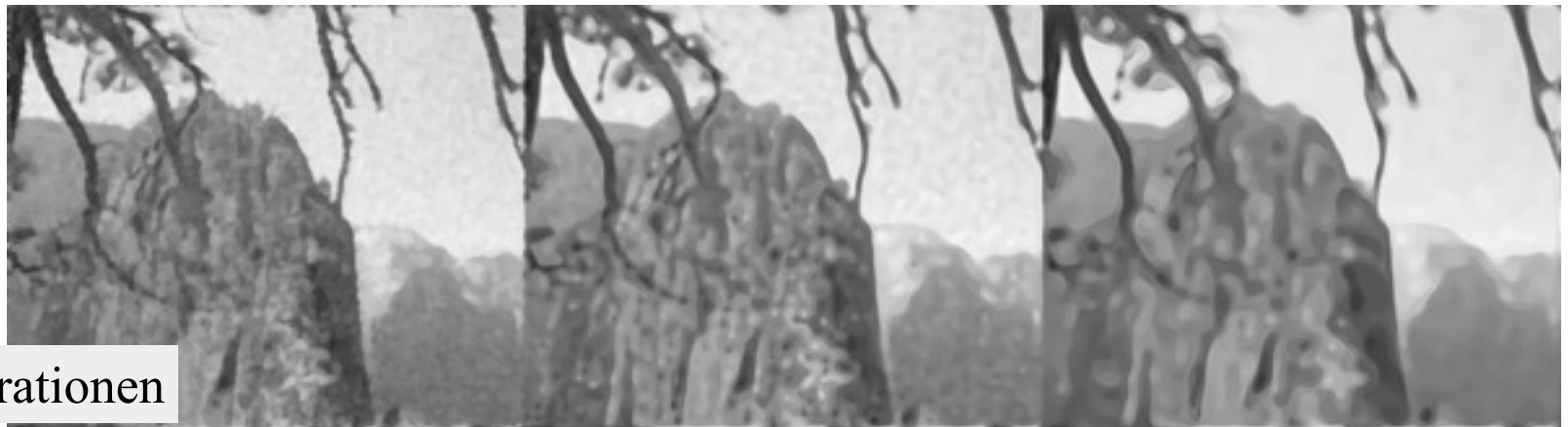
Resultat



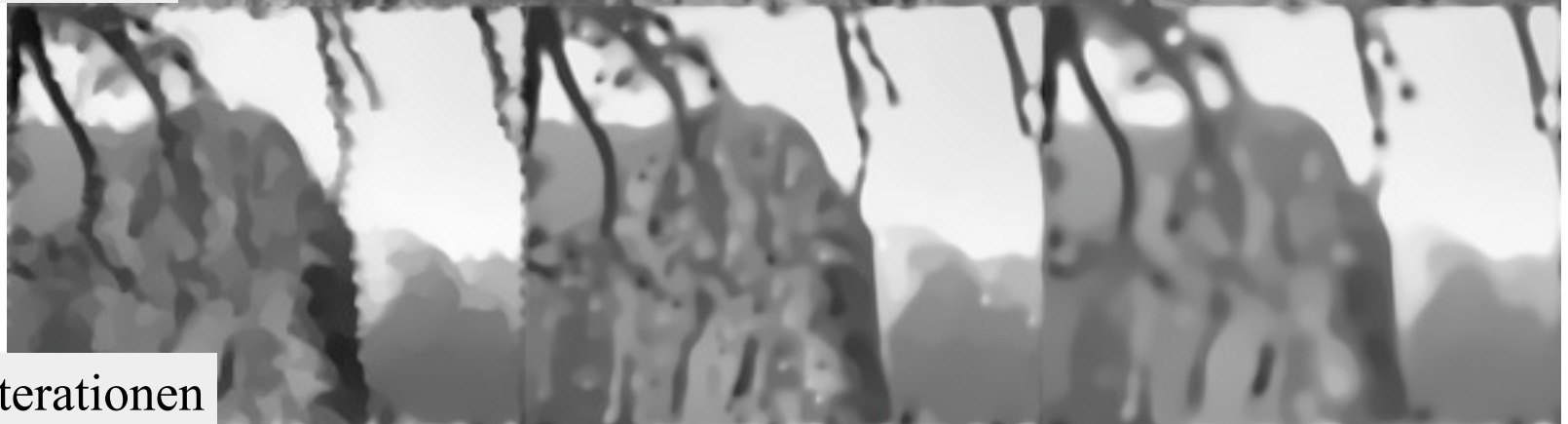
$\sigma=2.0$

$\sigma=5.0$

$\sigma=9.0$



20 Iterationen



200 Iterationen

..Also nochmal kurz:

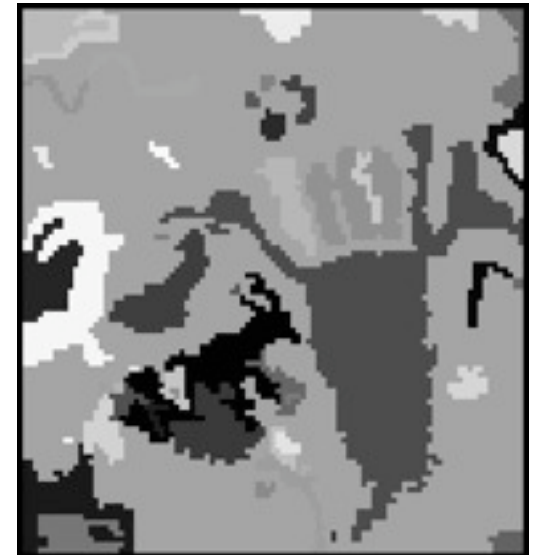
- Durch nichtlineare Operatoren können Kanten und Rauschen getrennt behandelt werden.
- Medianfilterung und Kantenmodell der Medianfilterung.
- Diffusionsfilter
 - Diffusionsmodell
 - Homogene, inhomogene und anisotrope Diffusion
 - Parametrisierung

Themen jetzt gleich

- Segmentierung
 - Thresholding
 - Region labeling
 - Flood fill
 - Relaxation labeling
- Regionenbasierte Segmentierung
 - Multiskalenstrategien
 - Region Merging
 - Split-and-Merge
 - Textursegmentierung
- Kantenbasierte Segmentierung
 - Edge Linking und Canny Edge Operator
 - Nulldurchgänge
 - Wasserscheidentransformation

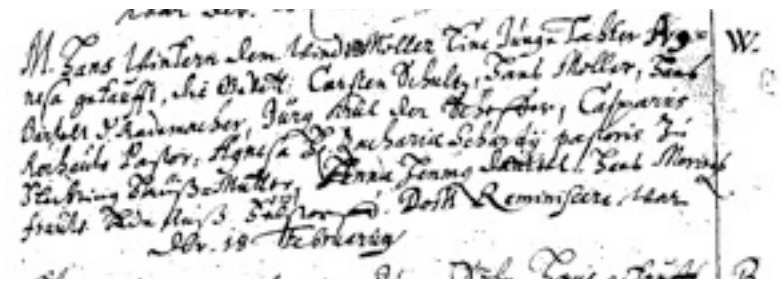
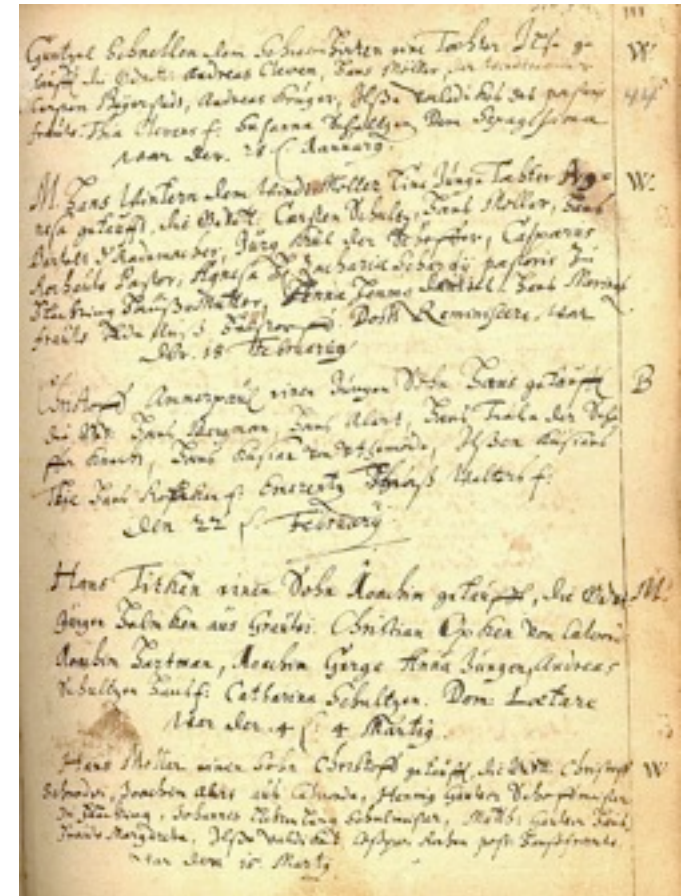
Segmentierung

- Ziel: Pixelaggregate schaffen, denen Bedeutung zugeordnet werden kann
- Pixel gehören zusammen, wenn sie einem zusammenhängenden Gebiet angehören.
- Homogenitätsbedingung ist vom Bildinhalt unabhängig (Segmentierung ist datenbasiert).
- Modellbasierte Segmentierung: Zwitter zwischen Segmentierung und Analyse.



Segmentierung

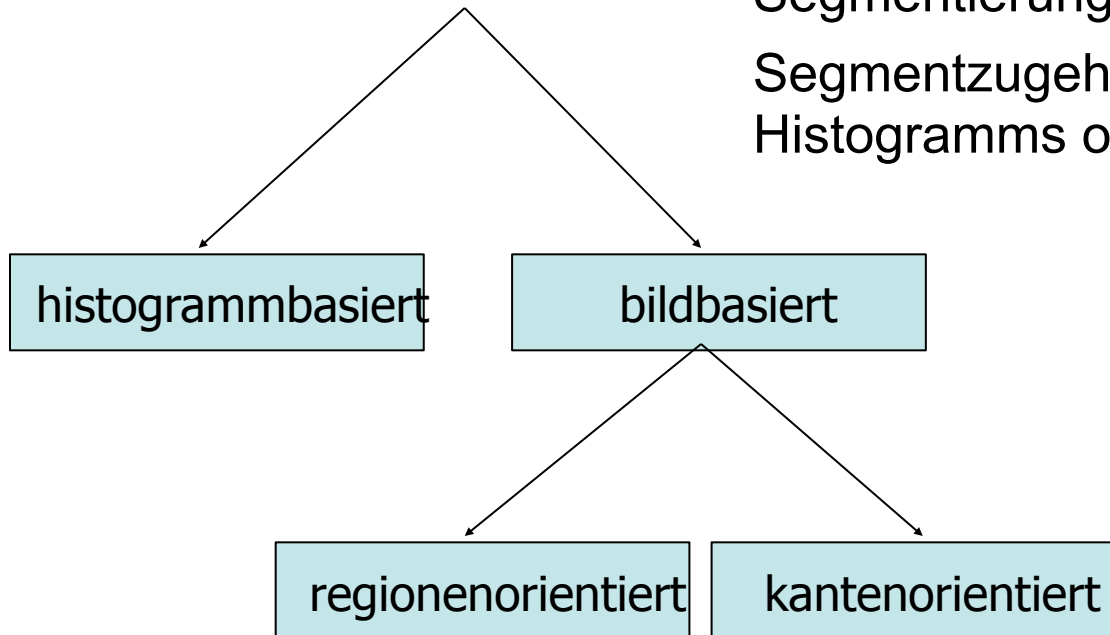
- Zerlegung eines Bildes in semantische Einheiten.
- **Segmente:** Träger der Bedeutung von Strukturen eines Bildes.
- Eigenschaften einer Segmentierung:
 - **vollständig:** jedes Pixel ist einem Segment zugeordnet.
 - **überdeckungsfrei:** ein Pixel ist genau einem Segment zugeordnet.
 - **zusammenhängend:** jedes Segment bildet ein zusammenhängendes Gebiet.



Segmentierungsmethoden

Histogrammbasierte oder bildbasierte
Segmentierung:

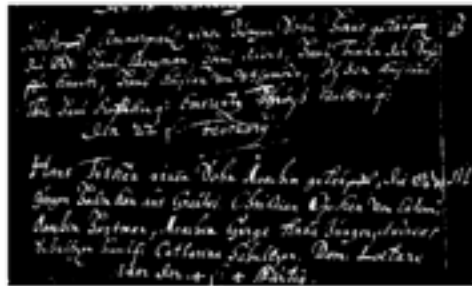
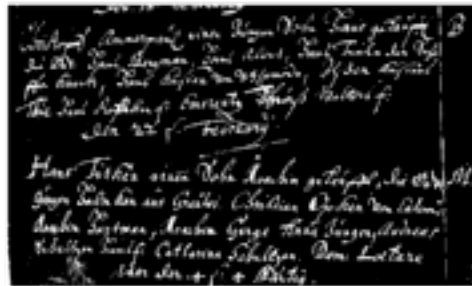
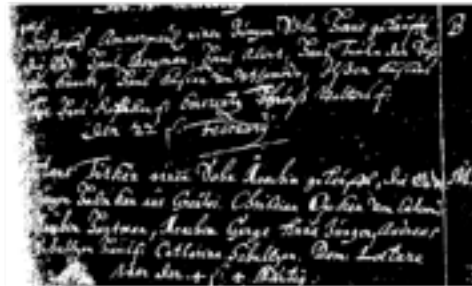
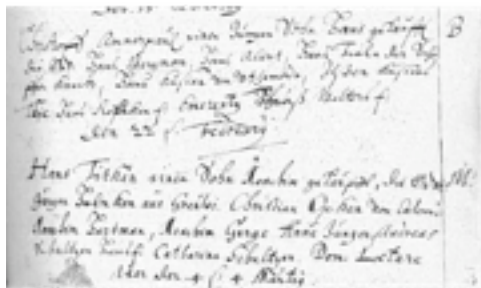
Segmentzugehörigkeit wird anhand des
Histogramms oder des Bildes entschieden



Regionen- oder Kantenorientierung:

Segmente werden durch ihre Grenzen oder ihr
Inneres definiert

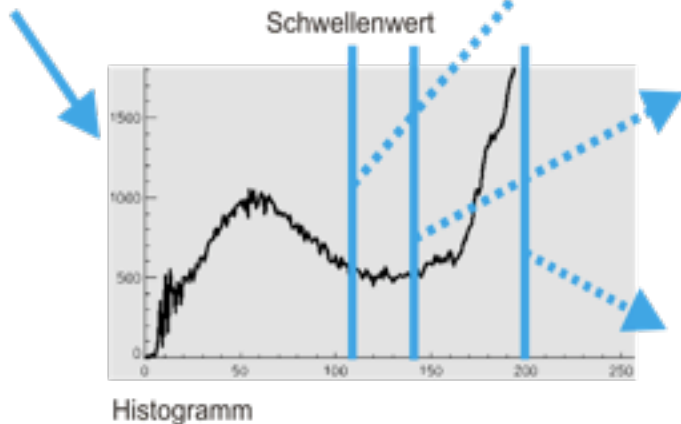
Histogrammbasierte Segmentierung



Annahme: Bild besteht aus zwei Anteilen die sich durch ihren Grauwert unterscheiden.

Aufgabe: Schwellenwert zwischen den beiden Grauwerten finden.

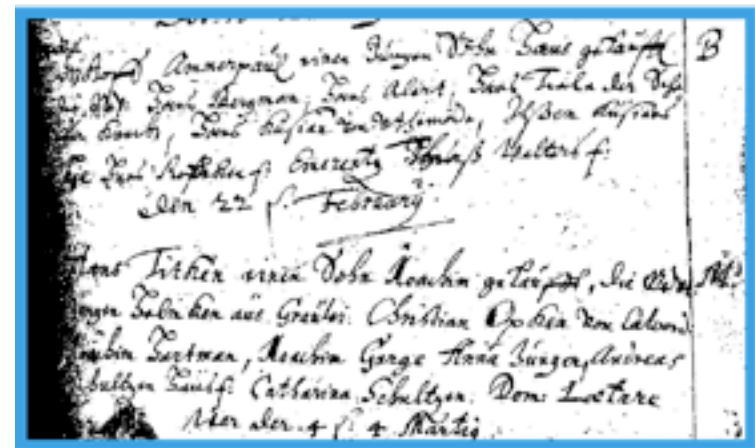
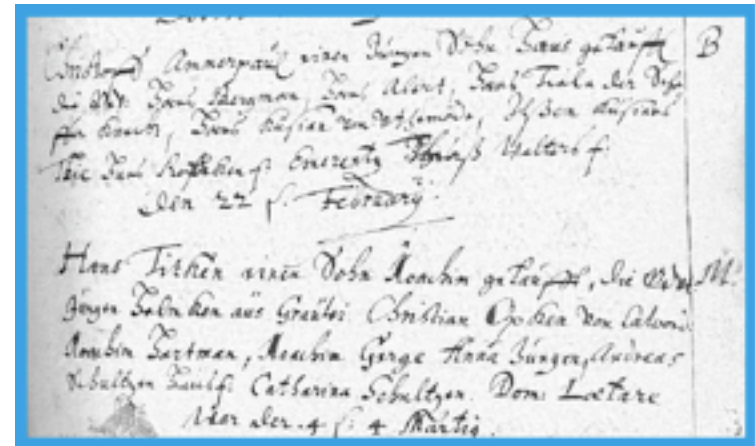
$$b(i,j) = \begin{cases} 1, & \text{falls } f(i,j) > T \\ 0, & \text{sonst} \end{cases}$$



Shading

Helligkeitsvariationen zerstören die bimodale Verteilung der Häufigkeiten.

- Schwellenwert ist nicht mehr für das gesamte Bild definierbar.

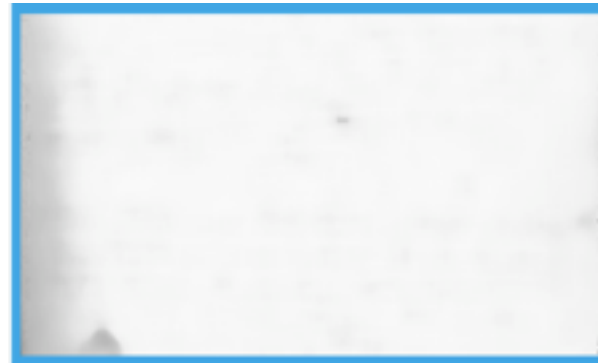
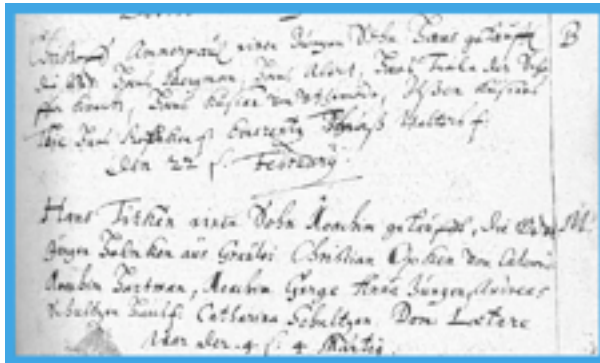


Berücksichtigung von Shading

- Homogenes Bild unter gleichen Bedingungen aufnehmen.
- Shading-Bild aus dem Bild selbst bestimmen.
- Variierende Schwellenwerte.

Bestimmung des Shadingbilds

- Falls die überwiegende Mehrzahl des Bildes aus Vorder- oder Hintergrund-pixeln besteht, kann das Shadingbild durch einem Rangordnungsfilter erzeugt werden.
- Das Rangordnungsfilter muss mindestens so groß sein, dass immer mindestens ein Vordergrund- und ein Hintergrundpixel enthalten ist.



13×13 Maximumsfilter

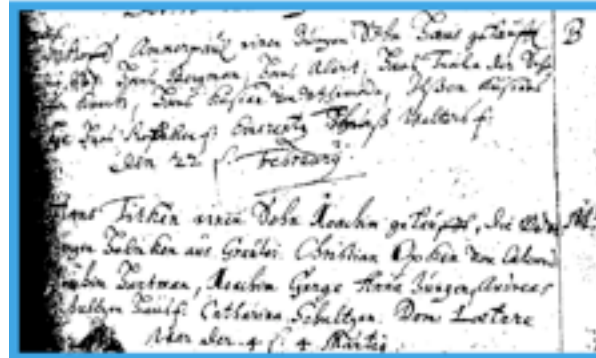
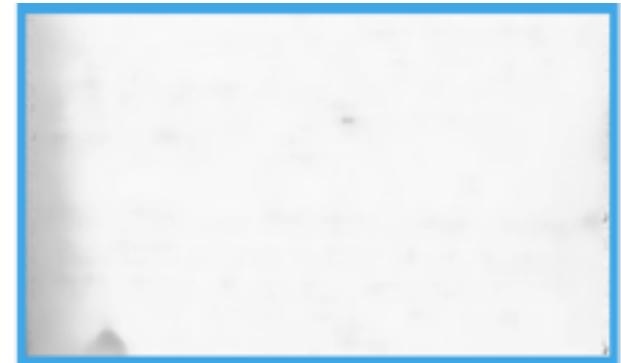
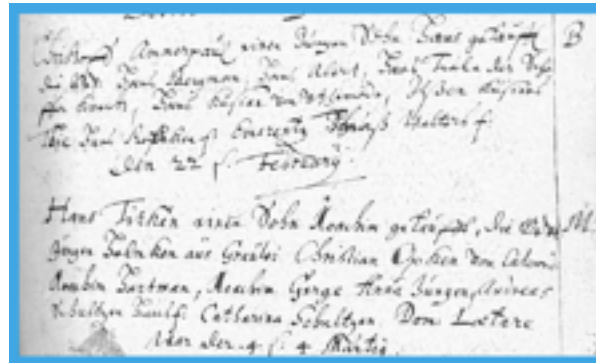
Shading-Korrektur

Berechne aus dem Hintergrundbild Shadingfunktion $s(i,j)$

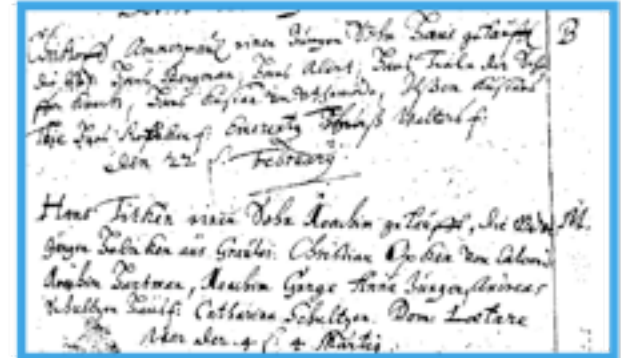
Korrektur:

$$f'(i,j) = f(i,j) \cdot s(i,j)$$

Segmentierung auf dem korrigierten Bild.

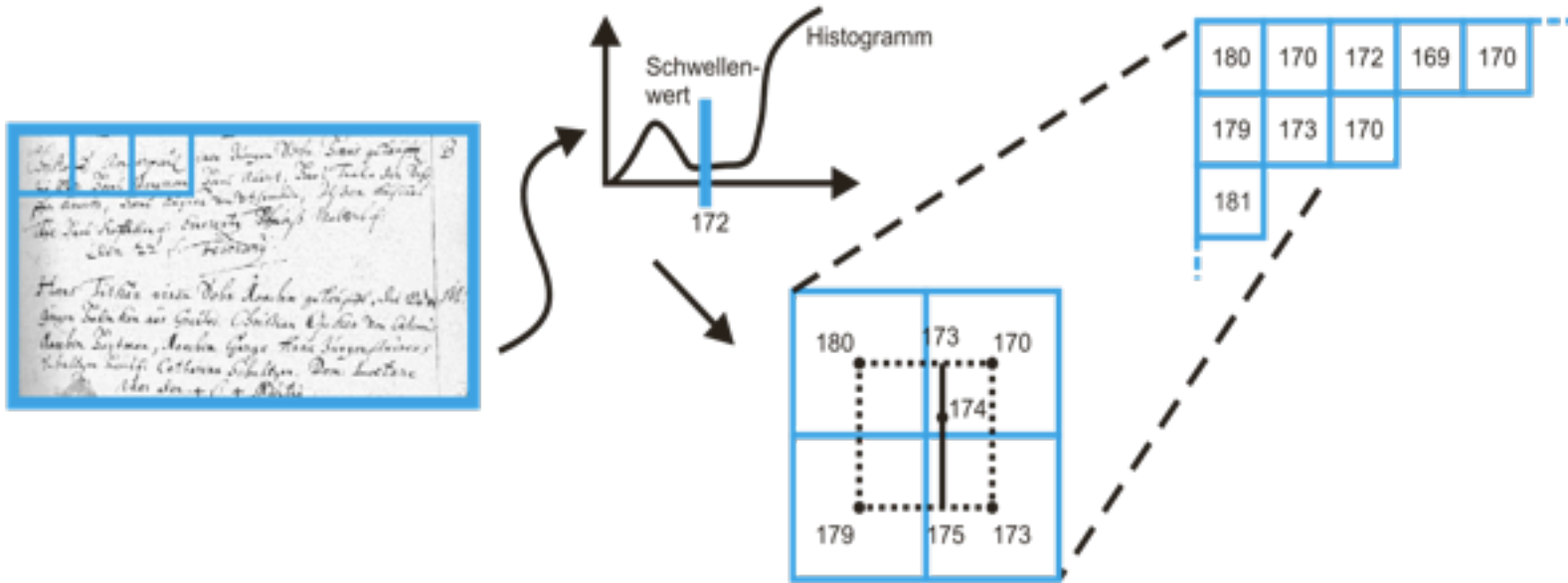


ohne Korrektur



mit Korrektur

Variierende Schwelle



Lokale Schwellenwerte aus Histogramme in Teilregionen.

(Lineare) Interpolation von Schwellenwerten $T(i,j)$ an allen anderen Punkten.

Segmentierung durch $f(i,j) > T(i,j)$.

Region Labeling

- Schwellenwert zerlegt das Bild in Vordergrund und Hintergrundsegmente.
- **Region Labeling** bestimmt Ort und Anzahl aller zusammenhängenden Gebiete im Binärbild b :

```
region.initialize() // Region der Größe M,N erzeugen und
label=1             // mit Null initialisieren, Startlabel=1
for (i,j) = 0, (M,N) do // Doppelschleife über i und j
    if region.labels(i,j) = 0 then // dieser Ort ist noch nicht
                                    Teil einer Region
        label = label+1             // neues Label
    vergeben
    region.flood_fill(i,j,label) // zusammenhängendes
                                    Gebiet um
    (i,j) mit                        Label füllen
```

Flood Fill

```
flood_fill(i,j,label) // Variablen zur Auswertung der
                       // Zusammenhangsbedingung sind global verfügbar
if f(i,j) erfüllt Zusammenhangsbedingung then
    region(i,j) = label // Region an (i,j) mit label
                       // versehen

    flood_fill(i-1,j,label) // Nachbarpixel untersuchen
    flood_fill(i,j-1,label)
    flood_fill(i+1,j,label)
    flood_fill(i,j+1,label)
```

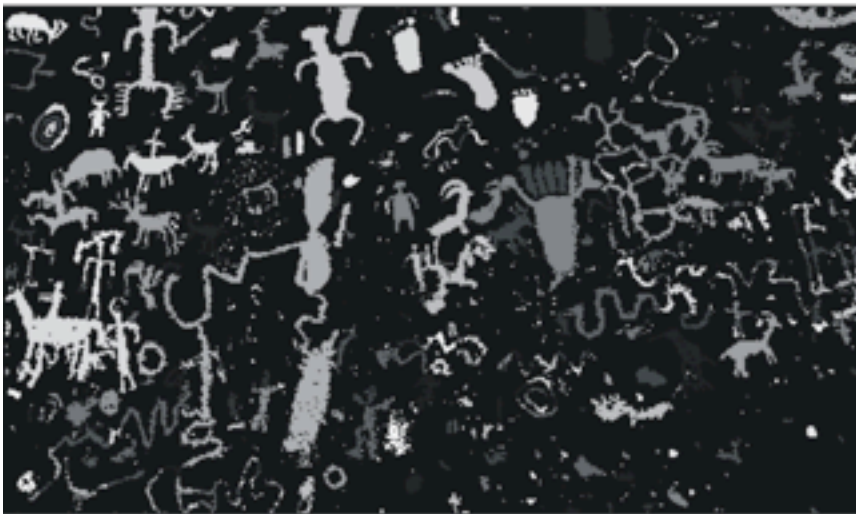
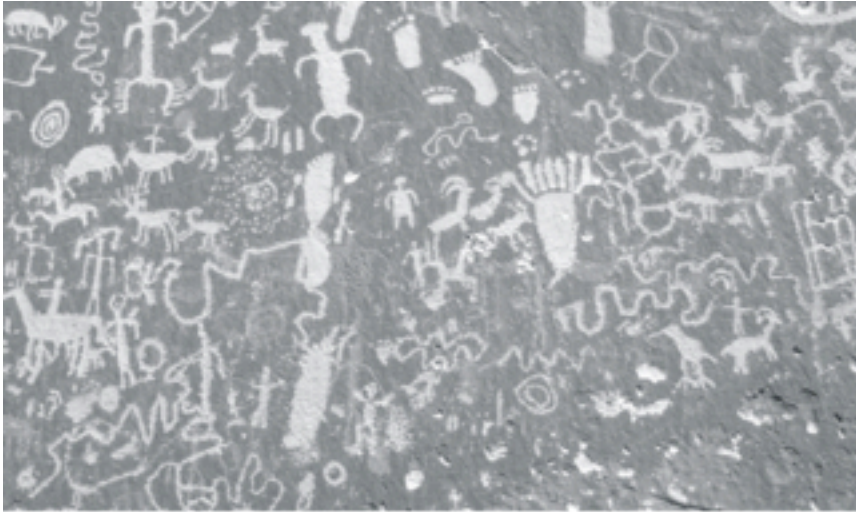
Region labeling + flood fill:

- Vollständige Segm.?
- Überdeckungsfrei?
- Zusammenhängend?

Bsp. für Zusammenhangsbedingung:

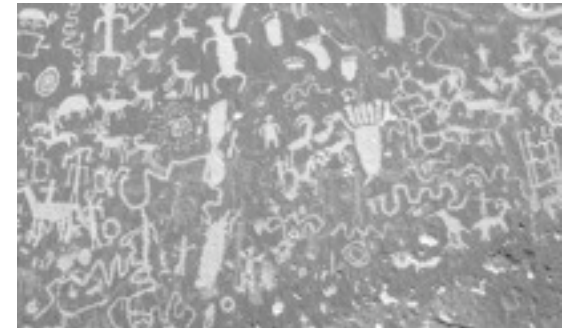
- hat den gleichen Grauwert wie Saatpunkt

Resultat



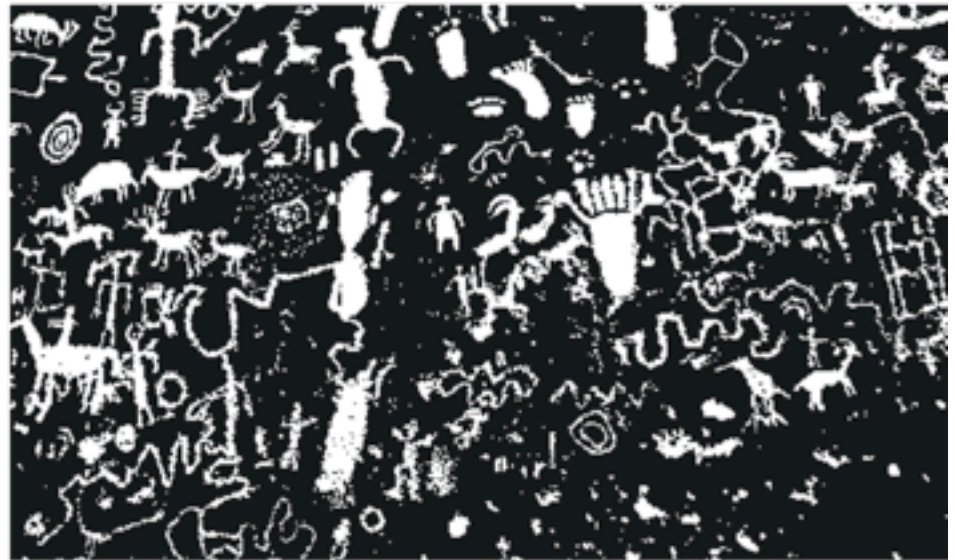
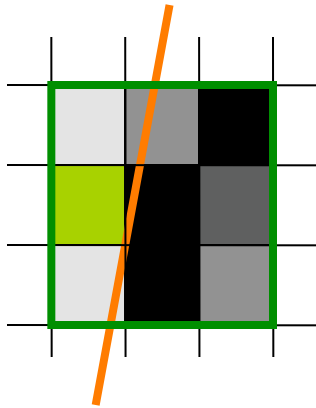
Nachverarbeitung

- Trennung nach Grauwerten wird nicht perfekt sein.
- Schwellenwertbild enthält falsche Regionen:
 - kleine fälschlicherweise als Segmente identifizierte Regionen.
 - Störungen am Rand von Regionen.
- Nachverarbeitung
 - Medianfilterung auf den Labels
 - Entfernung von zu kleinen Regionen
 - Relaxation Labeling

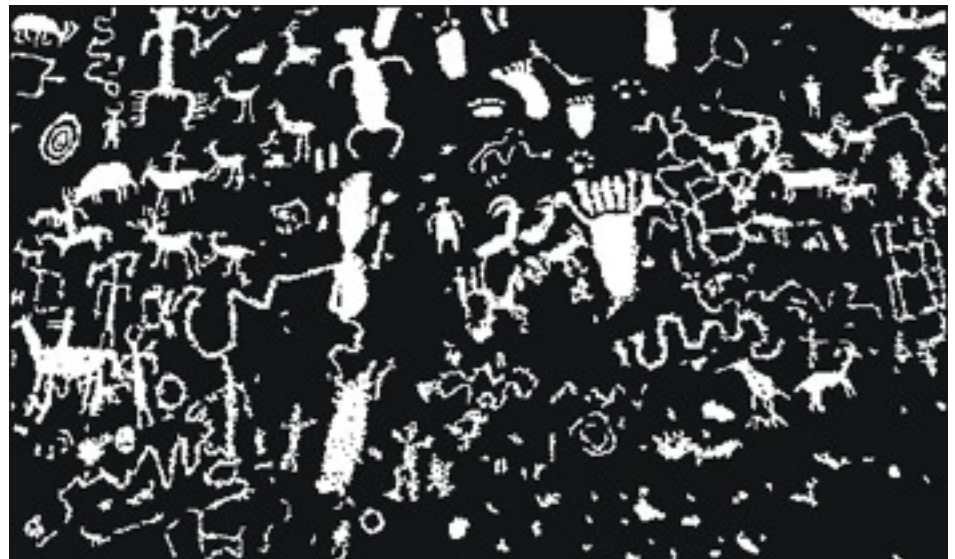
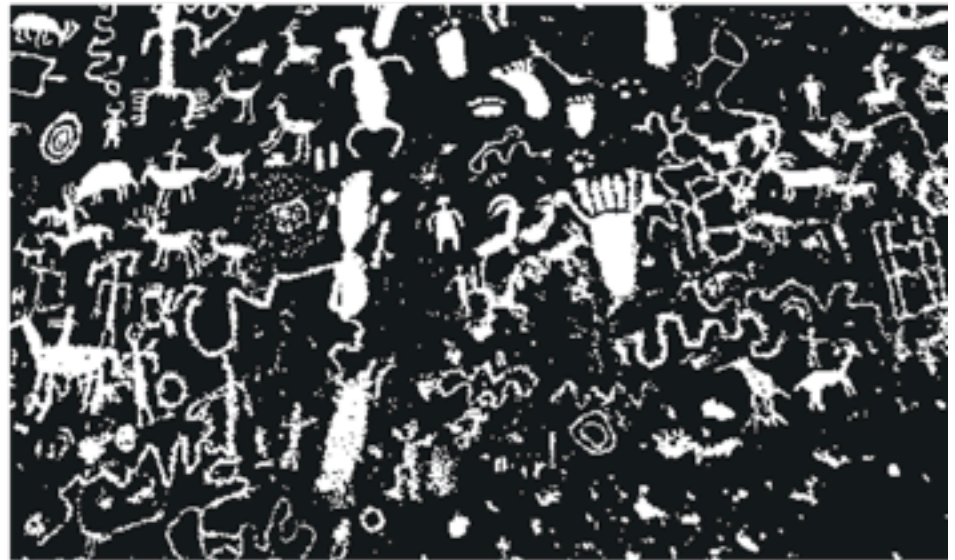


Medianfilterung auf Labeln

Wieso geht das
überhaupt?
→ siehe Def.
Medianfilter



Entfernung kleiner Gebiete



Relaxation Labeling

- Jedes Pixel erhält jedes Label mit einer vorgegebenen Wahrscheinlichkeit
(Bsp.: Schwellenwertresultat, Wahrscheinlichkeiten „70% weiß“ und „30% schwarz“ für weiße Pixel; umgekehrt für schwarze Pixel)
- Benachbarte kompatible Pixel unterstützen sich.
- Relaxationsprozess: Zuordnungswahrscheinlichkeiten ändern sich mit dem Maß der Unterstützung.
- Zu definieren:
 - Kompatibilität
 - Einfluß der Kompatibilität auf die Labelwahrscheinlichkeiten.

Labelwahrscheinlichkeit

- Aufzählung aller Pixel in Liste $p_0, p_1, \dots, p_{NM-1}$.
- K verschiedene Label
- Initiale Labelwahrscheinlichkeit P^0 für jedes Pixel, und jedes Label l_k vergeben, z.B.

$$P^0(p_i, l_k) = \begin{cases} 0.8 & , \text{ falls } l_k = l(p_i) \\ 0.2 & , \text{ sonst} \end{cases}$$

Labelwahrscheinlichkeit darf nicht 0 oder 1 sein
(Gewissheiten werden nicht verändert)

Kompatibilität

- Ein Pixel p_i mit Label l_k hat eine Kompatibilität r mit einem Pixel p_j , dessen Label l_l sei:

$$r((p_i, l_k), (p_j, l_l))$$

- Kompatibilitätskoeffizient für Binärbilder (K=2 Label) z.B.

$$r((p_i, l_k), (p_j, l_l)) = r(l_k, l_l) = \begin{cases} 1 & , \text{ falls } l_k = l_l \\ 0 & , \text{ sonst.} \end{cases}$$

(d.h., nur gleiche Label unterstützen sich)

Unterstützung eines Pixels

- Unterstützung $q^{(n)}$ von Pixel p_i durch Pixel p_j zur Iteration n

$$q_j^{(n)}(p_i, l_k) = \sum_{l=0}^{K-1} P^{(n)}(p_j, l_l) \cdot r((p_i, l_k), (p_j, l_l))$$

- Unterstützung von p_i durch alle Pixel

$$Q^{(n)}(p_i, l_k) = \sum_{j=0}^{NM-1} c_{ij} q_j^{(n)}(p_i, l_k)$$

mit Einflussparameter c_{ij} , z.B.

$$c_{ij} = \begin{cases} 1/8 & , \text{ falls } p_j \in N_8(p_i) \\ 0 & , \text{ sonst.} \end{cases}$$

Iterationsschritt

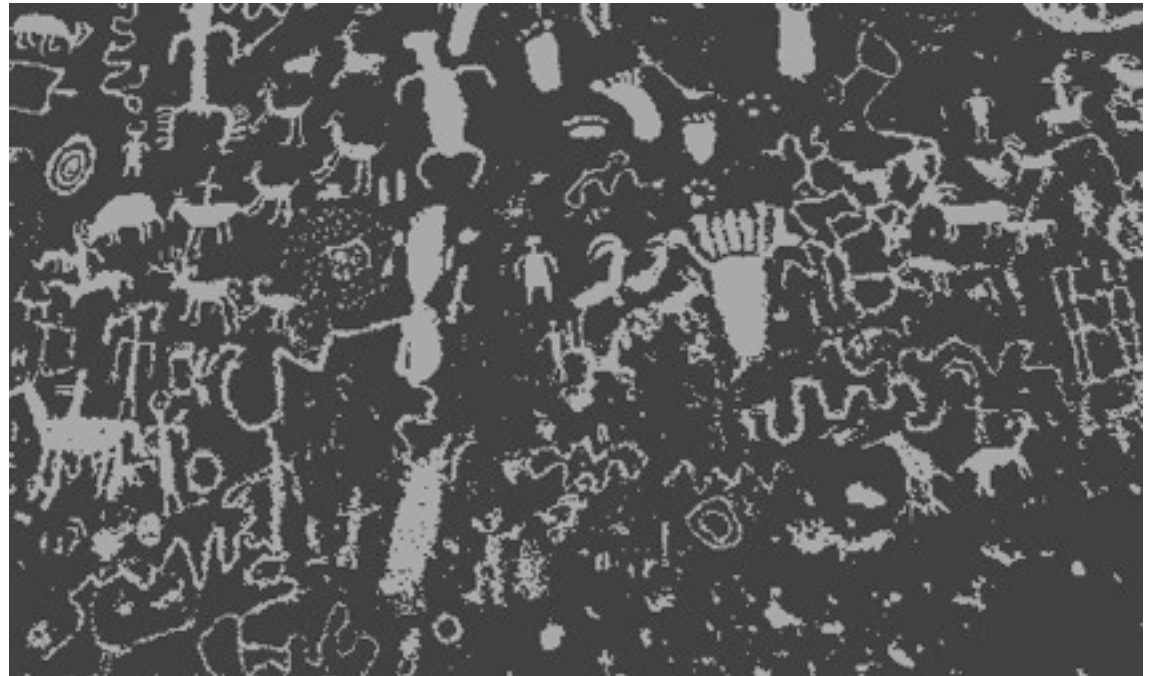
$$P^{(n+1)}(p_i, l_k) = \frac{P^{(n)}(p_i, l_k) [1 + Q^{(n)}(p_i, l_k)]}{\sum_{l=0}^{K-1} P^{(n)}(p_i, l_l) [1 + Q^{(n)}(p_i, l_l)]}$$



Confidence Map

- Um das Konvergenzverhalten zu beobachten, kann eine **Confidence Map** erzeugt werden.
- Confidence Map: Gibt für jedes Pixel die Zuverlässigkeit der derzeitigen Entscheidung an (weiss = sicher)
- Für 2-Label-Segmentierung: Differenz zwischen gewählten Label und nicht gewähltem Label (evtl. gewichtet mit der Anzahl der Pixel mit diesem Label).

Initiale Confidence Map

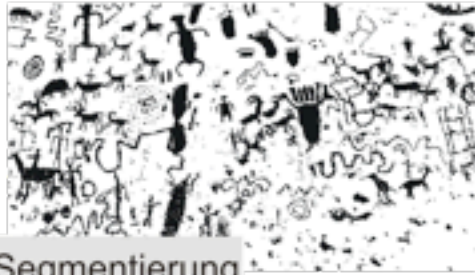


Konvergenz

Iteration 0

1

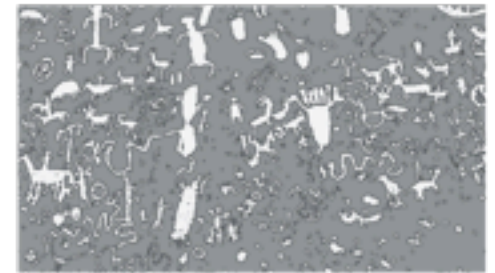
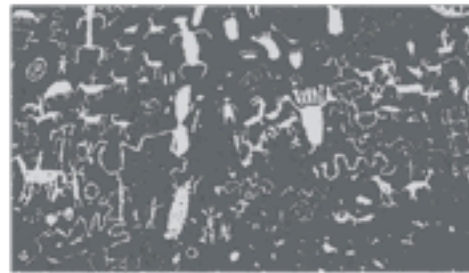
2



Segmentierung



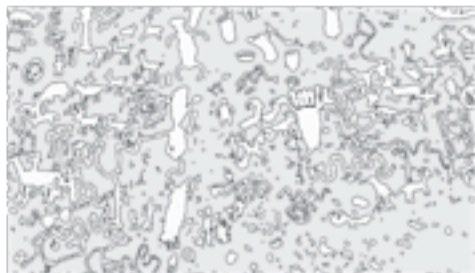
Confidence Map



4

10

20



..und nochmal kurz zusammengefasst:

- Zuerst:
 - Schwellenwertsegmentierung
 - Shadingkorrektur
- Dann:
 - Region Labeling
 - Nachverarbeitung
 - Relaxation Labeling

Regionenbasierte Segmentierung

- Multiskalenstrategien
- Region Merging
- Split-and-Merge
- Textursegmentierung

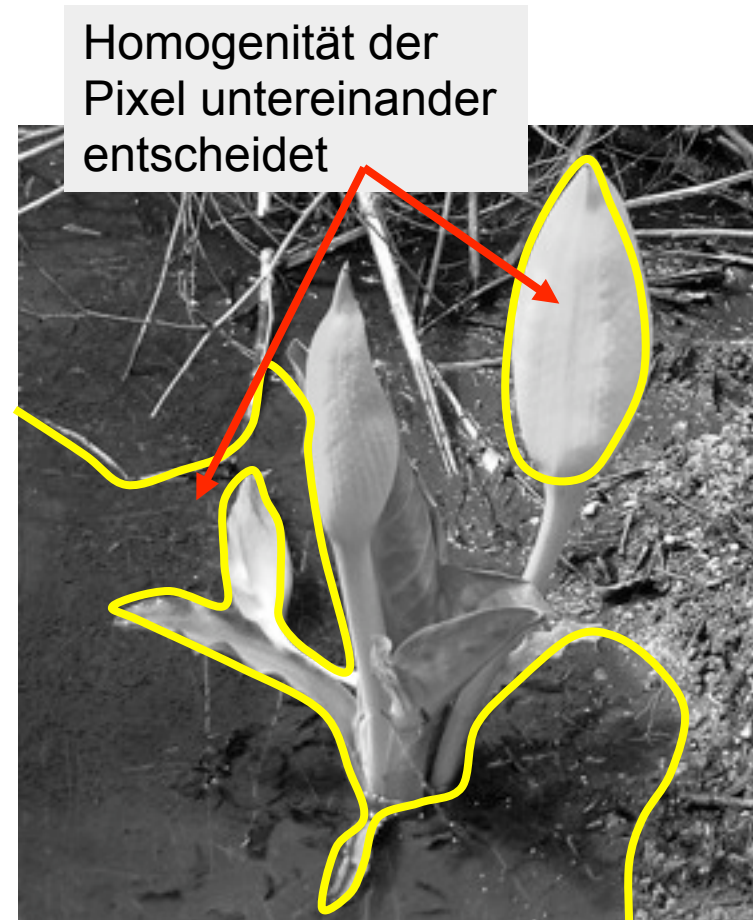
Regionenbasierte Segmentierung

Homogenität im Inneren des Segments.

Homogenitätsbedingung wird **bei der Segmentierung** ausgewertet.

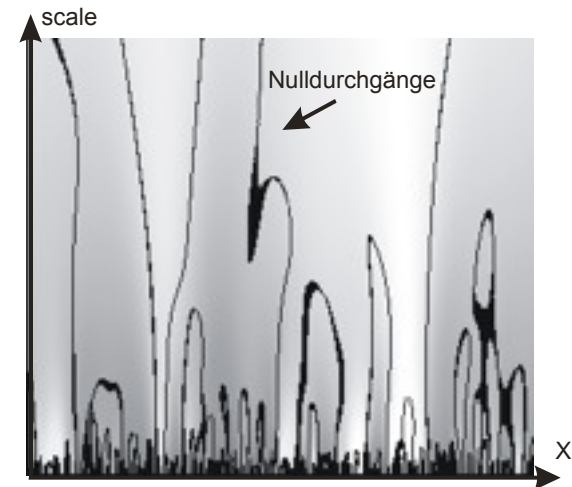
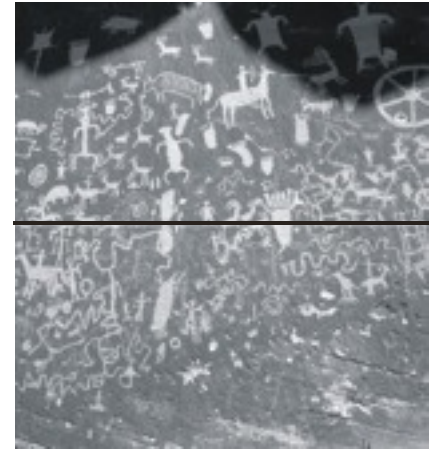
Homogenität ist **relativ** zu den Attributen eines Segments definiert.

Globale Zusammenhänge über **Multiskalenstrategie**.



Multiskalenstrategie

- Relative Kriterien für Homogenität können über unterschiedliche Entfernungen verschieden wirken.
- Segmentierung nach Multiskalenstrategie wertet Kriterien auf unterschiedlichen Skalierungen aus.
- Multiskalenstrategie
 - explizit auf einer Multiskalenrepräsentation
 - implizit in den Segmentierungsalgorithmus integriert.



Gaußpyramide

- Das Originalbild wird fortlaufend durch eine „reduce“-Operation skaliert.
- Jedes Pixel der nächsthöheren Skalierungsstufe repräsentiert 4 Pixel der aktuellen Stufe.
- Vor der Reduktion wird der Frequenzumfang durch Filterung vermindert:

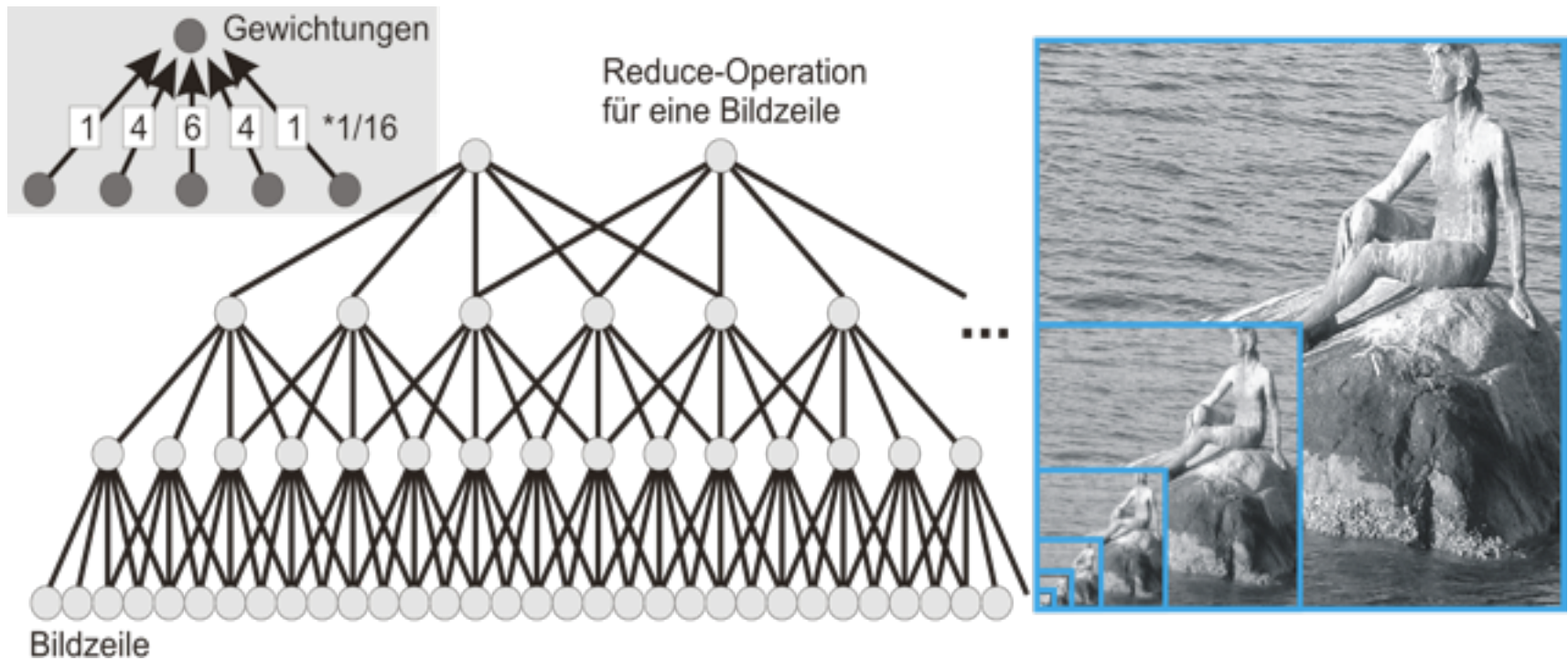
– Gaußfilter

$$\frac{1}{16} \begin{pmatrix} 0.87 & 3.91 & 6.44 & 3.91 & 0.87 \end{pmatrix}$$

– Binomialfilter

$$\frac{1}{16} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

Gaußpyramide



Expand-Operation

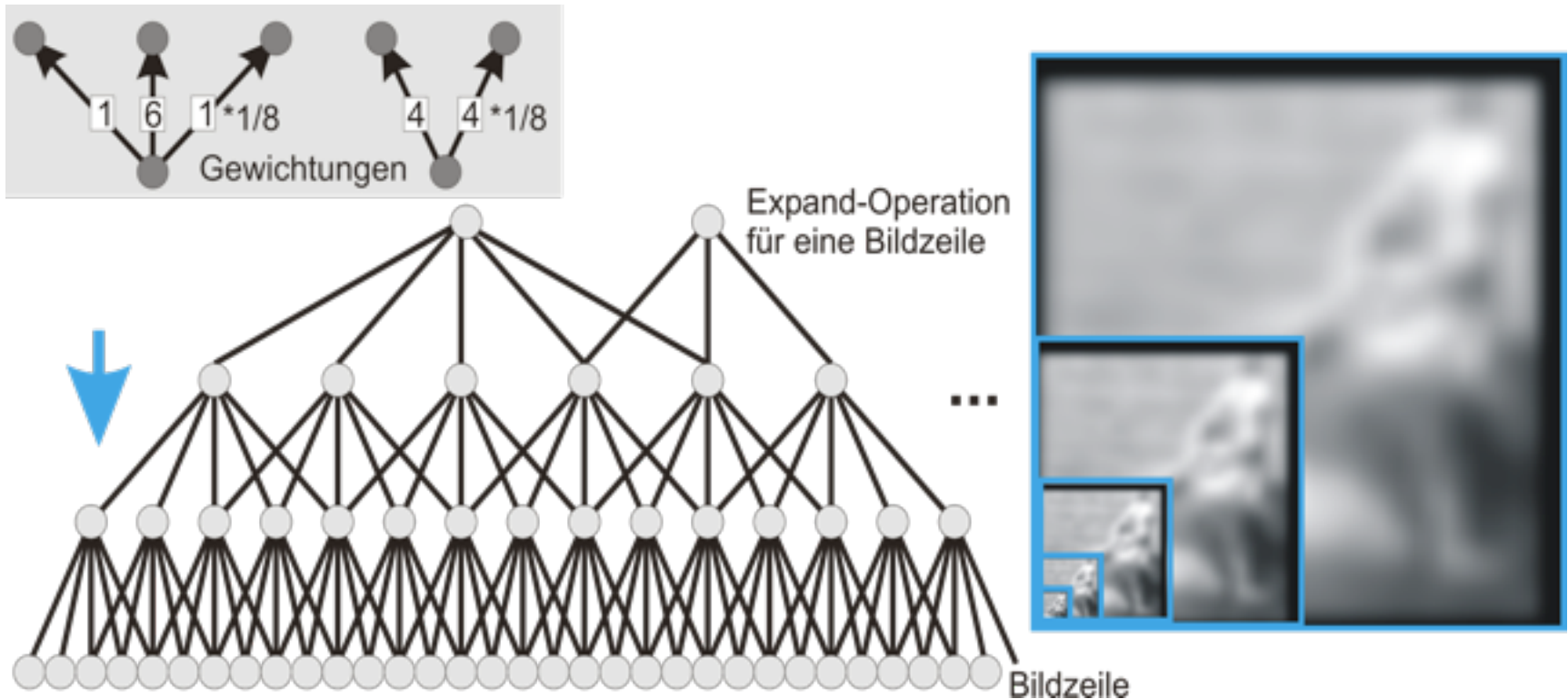
- Um die vorherige Skalierungsstufe zu erzeugen, wird eine „expand“-Operation definiert.
- Pixel der neuen Skalierungsstufe werden durch Interpolation erzeugt:
 - Pixelorte, die auf beiden Skalierungsstufen existieren:

$$\frac{1}{8.18} (0.87 \quad 6.44 \quad 0.87) \text{ bzw. } \frac{1}{8} (1 \quad 6 \quad 1)$$

- Pixelorte, die nur auf der vorherigen Skalierungsstufe existieren:

$$\frac{1}{7.82} (3.91 \quad 3.91) \text{ bzw. } \frac{1}{8} (4 \quad 4)$$

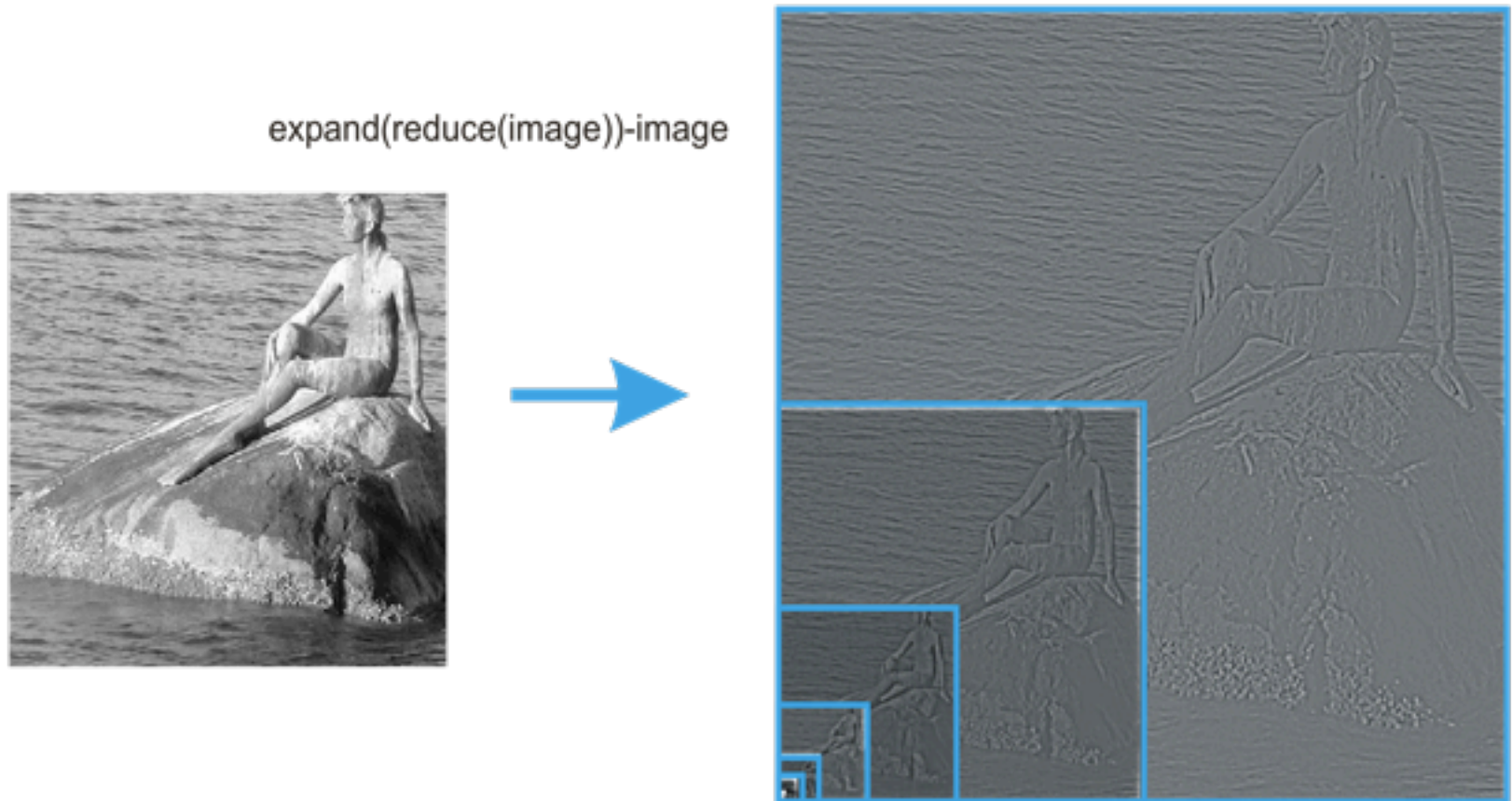
Expand-Operation



Die Expand-Operation ist nicht verlustfrei

Laplace-Pyramide

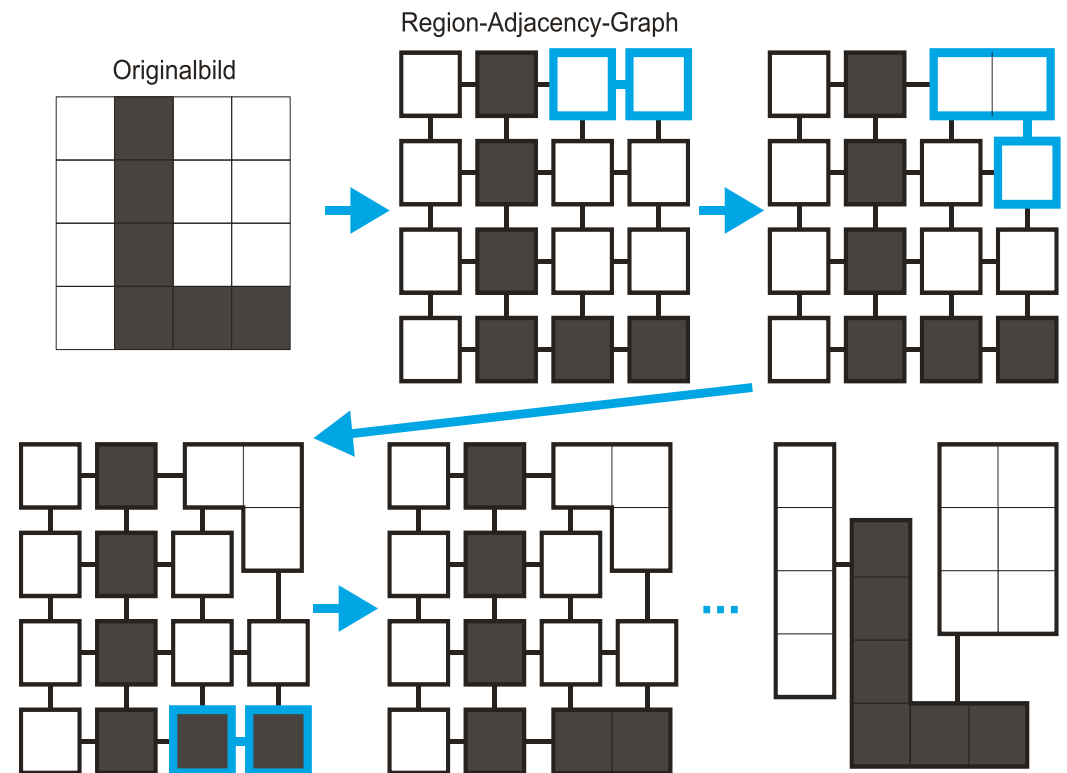
- Jede Skalierungsstufe s enthält nur den Unterschied $f_s - \text{expand}(\text{reduce}(f_s))$
- Redundanzfreie Repräsentation



Region Merging

- Initial wird jedes Pixel zu einem Segment erklärt.
- Zwei benachbarte Regionen werden zusammengefasst, wenn sie auch gemeinsam das Homogenitätskriterium erfüllen.
- Die Segmentierung ist beendet, wenn keine zwei Regionen mehr existieren, die zusammengefasst werden können.

- Zwischenergebnisse werden in einem Region Adjacency Graph (RAG) gespeichert.



Region Merging

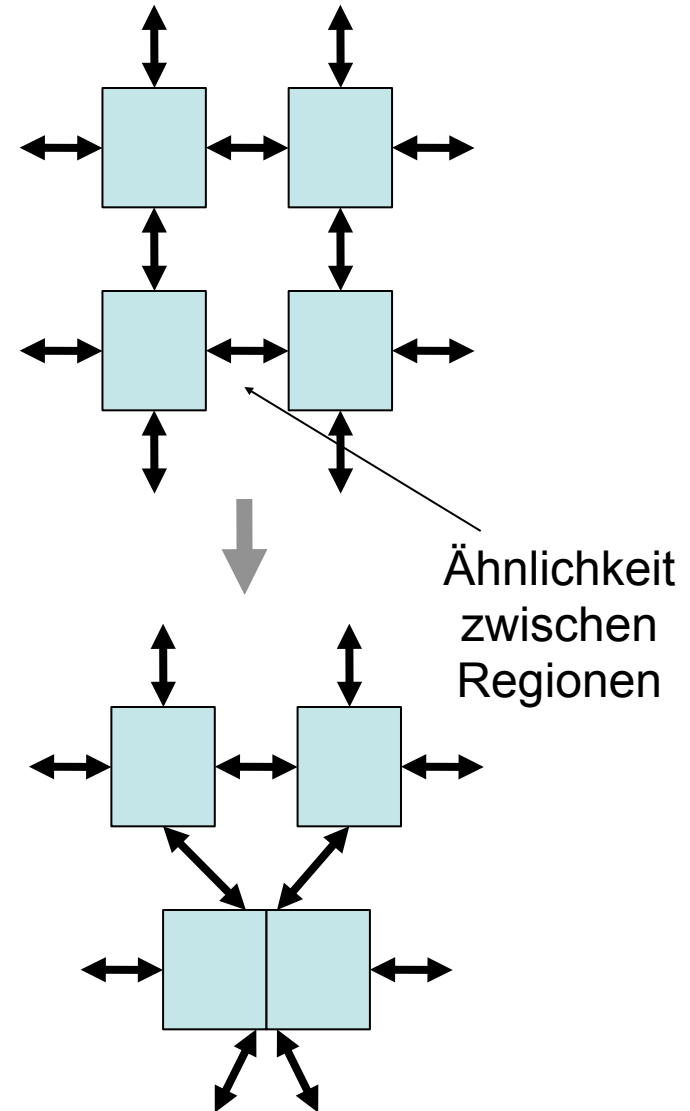
„von Pixeln zu Regionen“:

```
stopMerge = false
while not stopMerge do
  (r1, r2) =
  MaxSimilarity(region)
  if sim(r1, r2) > T then
    region.merge(r1, r2)
  else
    stopMerge = true
```

Bsp. f. Ähnlichkeitskriterium:

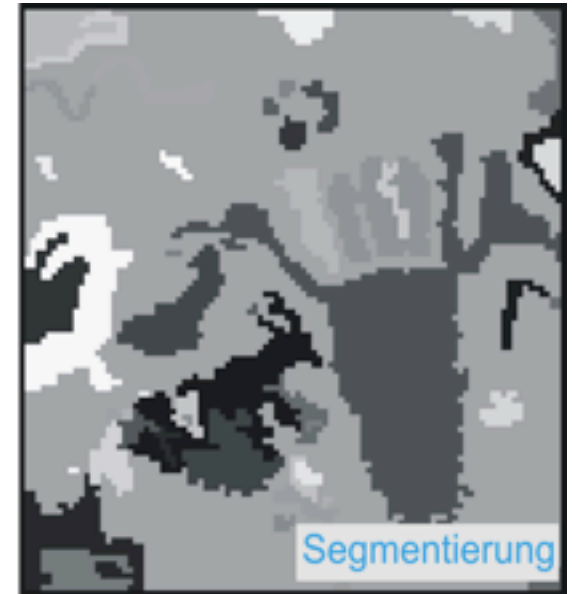
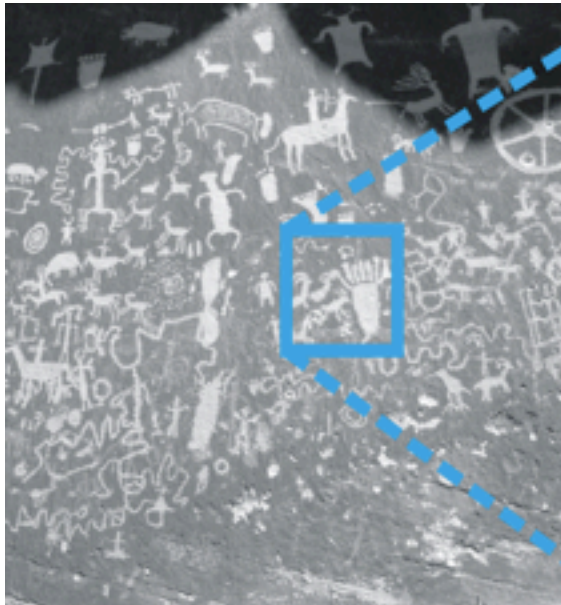
maximaler Grauwertunterschied
zwischen Pixeln von r1 und r2.

Region Labeling kann in den
Prozess integriert werden.



Region Merging

- Homogenitätskriterium
 - Grauwertdifferenz innerhalb der Region.
 - Wahrscheinlichkeit, dass beide Regionen die gleichen Grauwertverteilungen haben.



Region Merging und Multiskalenstrategie

Modellannahme:

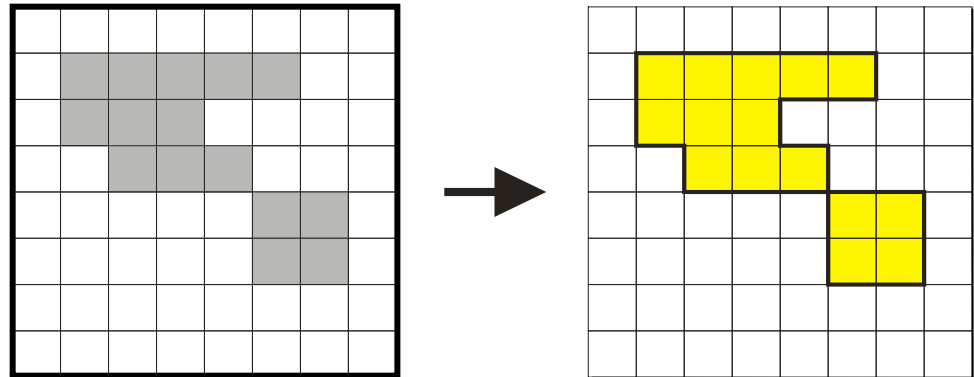
Die größte Skalierungsstufe, auf der sich segmentierungsrelevante Eigenschaften manifestieren, ist bekannt.

Prozess:

- Region Merging auf grober Skalierung
- Übertragung des Resultats auf die nächstfeinere Stufe.
- Alle Pixel, die zu Pixeln eines anderen Segments benachbart sind, werden nochmals geprüft.
- Verfahren endet, wenn die die feinste Skalierungsstufe erreicht ist

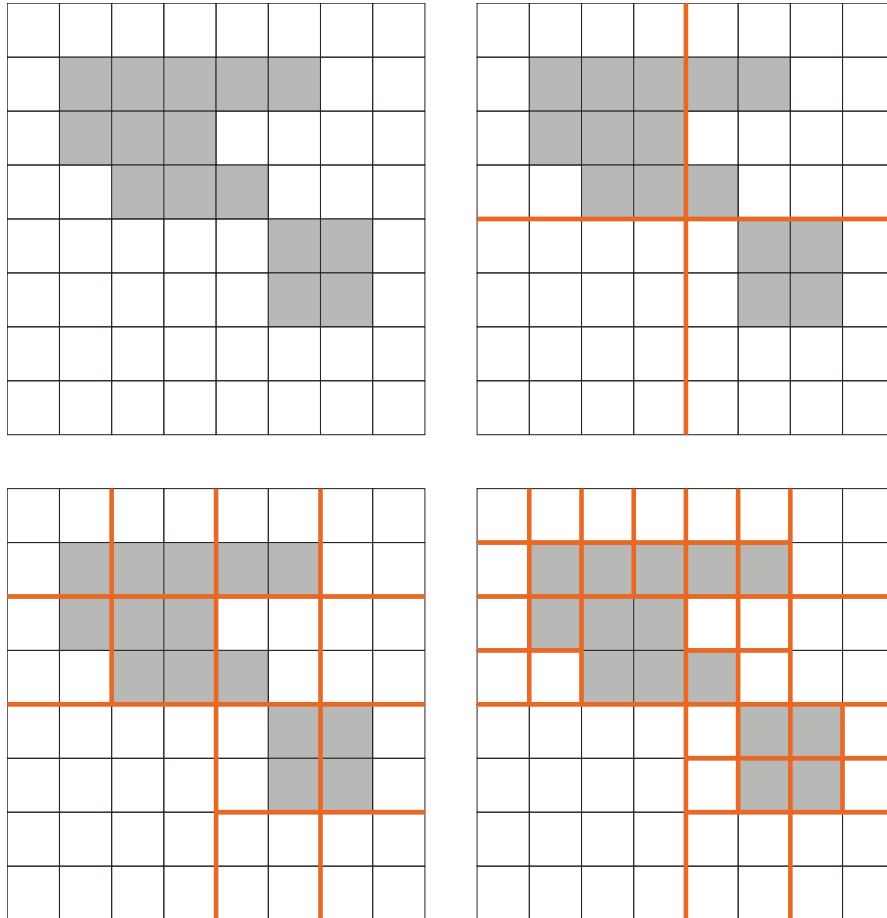
Split & Merge-Algorithmus

Regionenbasiertes
Verfahren



- **Startbedingung:** Das gesamte Bild ist ein Segment.
- Ein Segment wird solange in 4 Untersegmente zerlegt, wie es ein gegebenes Homogenitätskriterium nicht erfüllt
- Benachbarte Segmente werden zusammengefasst, wenn sie auch nach der Zusammenfassung das Homogenitätskriterium erfüllen.
- **Resultat** ist eine vollständige, überdeckungsfreie Zerlegung des Bildes (Segmentierung gemäß Definition)

Zerlegungsschritt

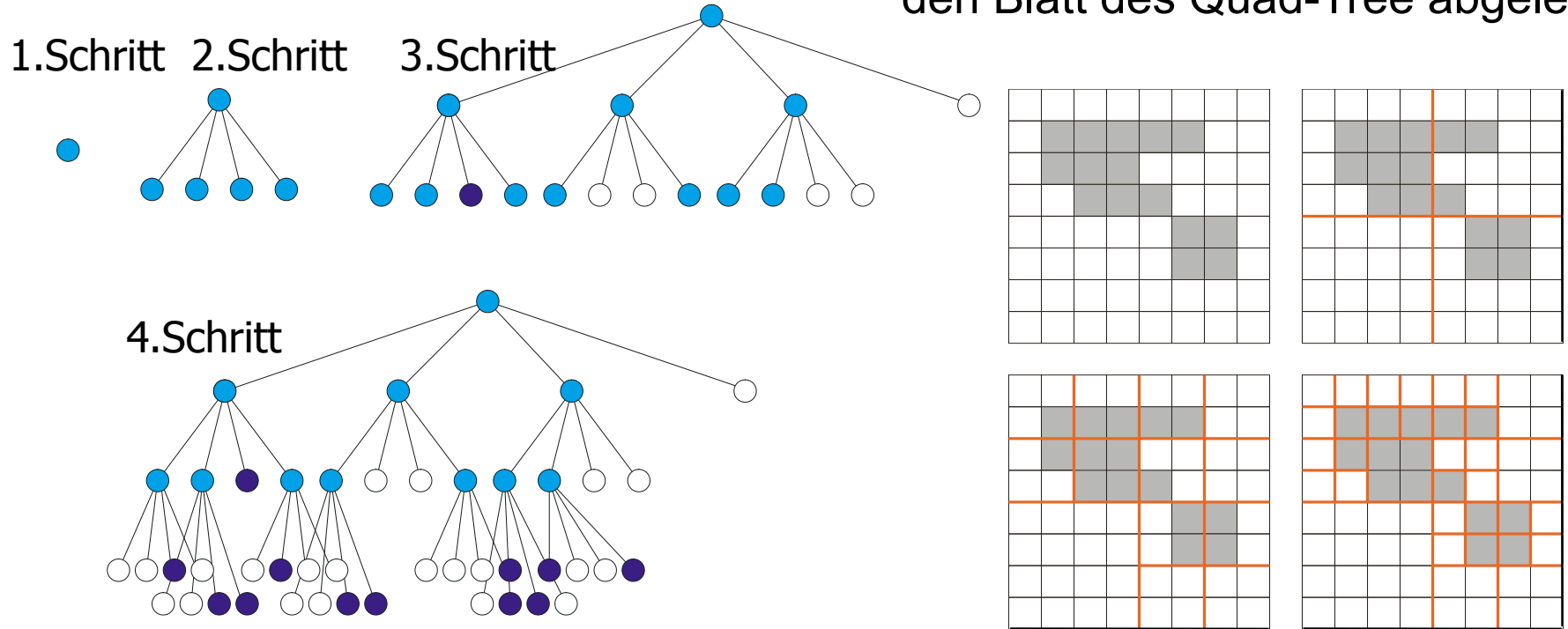


Zerlegung terminiert
spätestens auf
Pixelebene.

Problem: Datenstruktur
zur Dokumentation der
aktuellen Zerlegung

Zerlegungsschritt (Quad-Tree Repräsentation)

Wert des Homogenitätsmerkmals
einer Region wird im entsprechen-
den Blatt des Quad-Tree abgelegt



Split & Merge

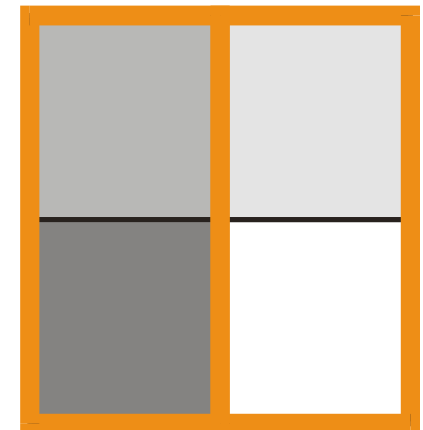
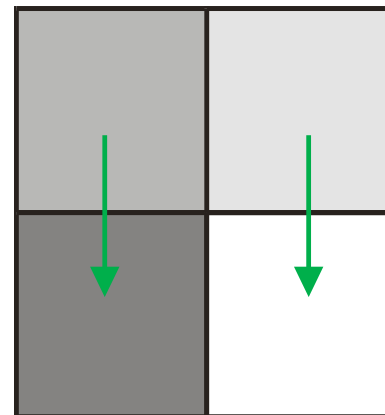
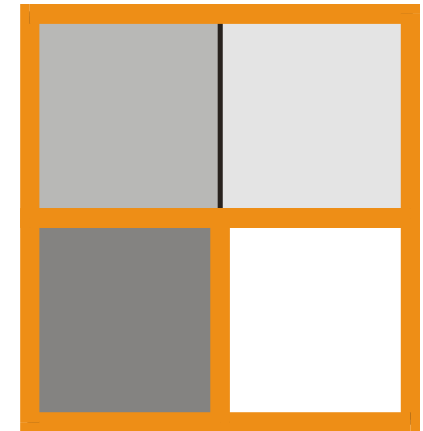
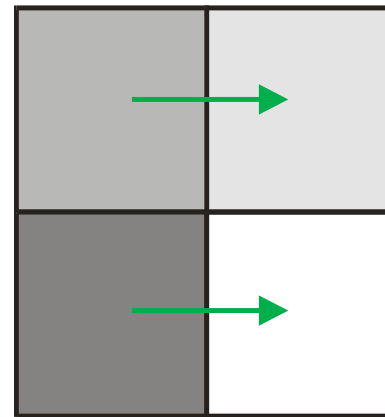
Resultat: Zerlegung des gesamten Bilds in Regionen.

Multiskalenstrategie ist integriert

Homogenitätsmerkmale wie bei
Region Merging

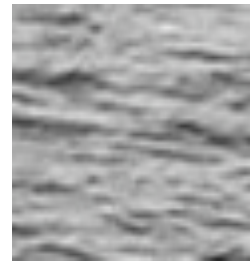
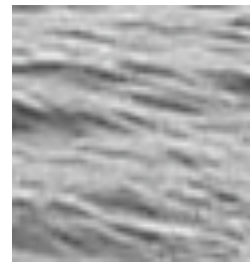
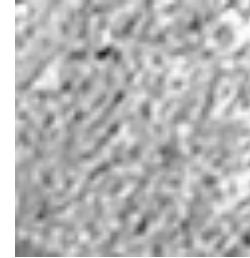
Probleme (Region Merging und
Split & Merge):

- Merge-Schritt ist bei relativem Homogenitätsmerkmal **nicht immer eindeutig**
- **Minimale Segmentzahl** wird nicht immer gefunden



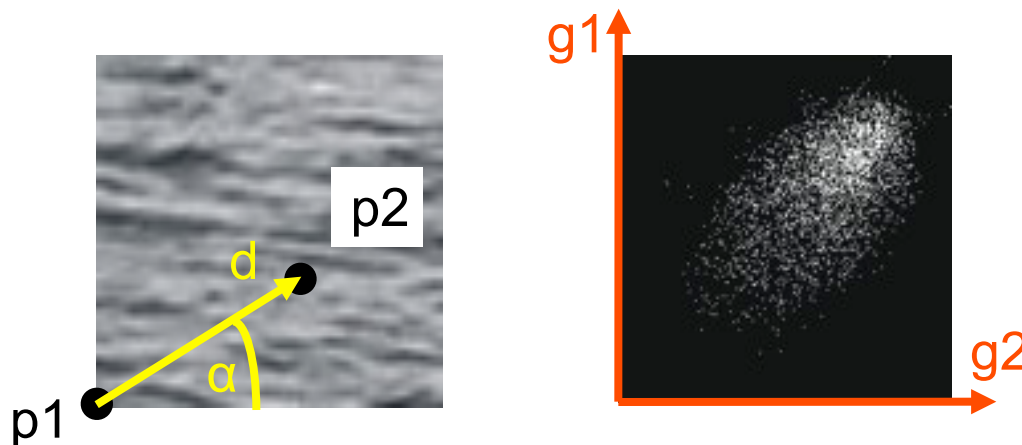
Textur als Homogenitätsmerkmal

- Textur – Musterung der Oberfläche
 - Es existiert keine Definition von Textur
 - Es gibt eine große Anzahl von Texturmaßen
 - Textur ist inhärent skalenabhängig
 - Textur ist eine Eigenschaft einer Gruppe von Pixeln.
- Texturmaße
 - strukturell (Zusammensetzung aus Texturelementen – texel)
 - stochastisch (eine charakterisierbare Grauwertverteilung)
 - spektral (charakteristische Frequenzattribute)



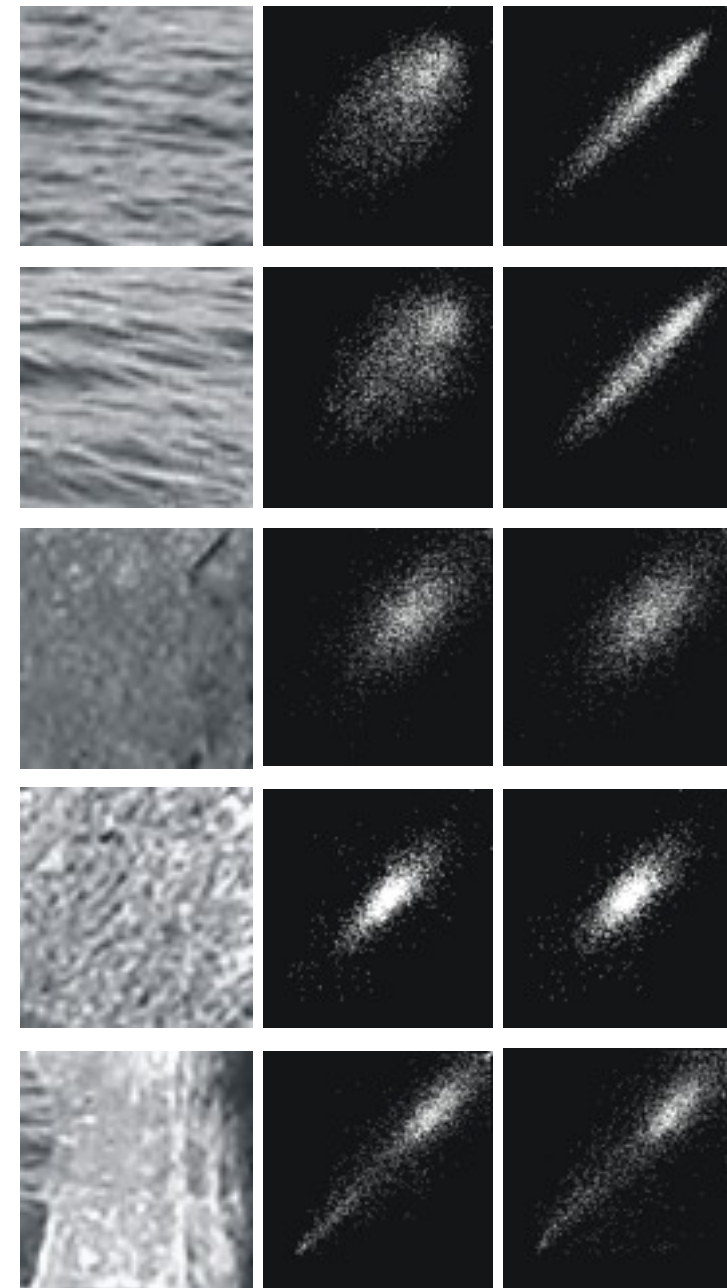
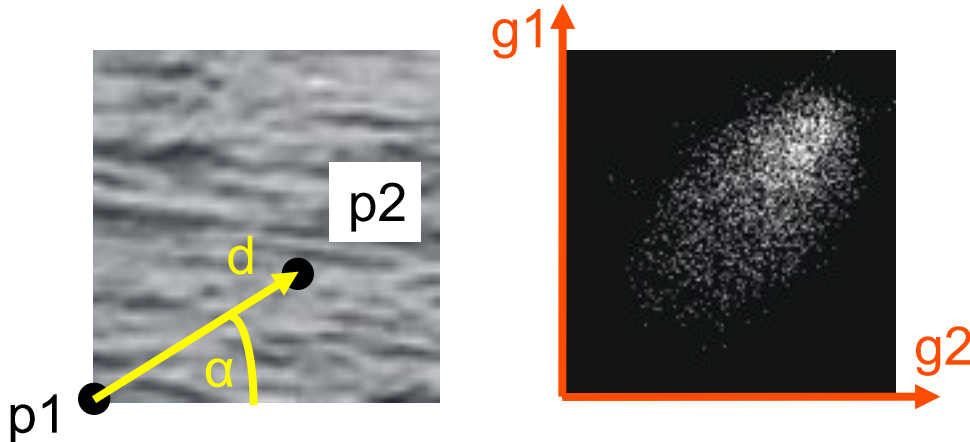
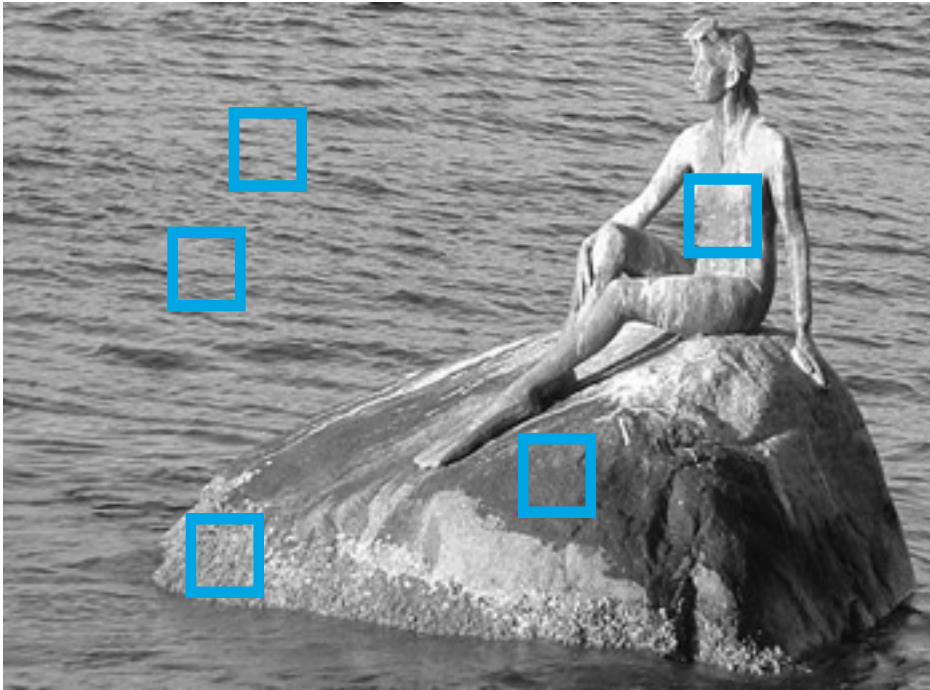
Haralick'sche Texturmaße

- **Co-Occurrence-Matrix** = Zweidim. Histogramm für Pixelpaare.
Pixel p_1 und p_2 sind ein Paar, wenn sie Abstand d haben und auf einer Linie mit einem gegebenen Winkel α zur x -Achse liegen.
- Co-Occurrence-Matrix repräsentiert die Korrelation zwischen Pixeln.
(Wahrscheinlichkeit P , dass p_1 und p_2 Grauwerte g_1 und g_2 haben)
- Meist sind Pixel nicht über große Entfernungen korreliert, daher sind Werte $d=1$, $d=2$ üblich.
- Falls Korrelation über größere Entfernung vermutet wird, dann sollte eine Multiskalenstrategie angewendet werden.



[Robert Haralick](#)

Co-Occurrence-Matrix



$d=1, \alpha=90^\circ, \alpha=0^\circ$

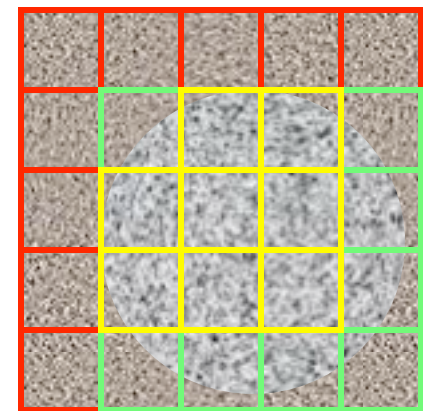
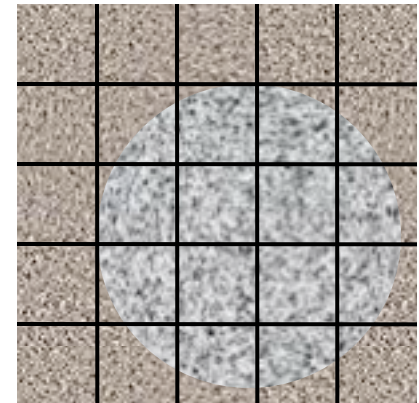
Haralick'sche Texturmaße

$\sum_{g_1=0}^{K-1} \sum_{g_2=0}^{K-1} P_{\Delta,\alpha}^2(g_1, g_2)$	Energie
$\sum_{g_1=0}^{K-1} \sum_{g_2=0}^{K-1} (g_1 - g_2)^2 \times P_{\Delta,\alpha}(g_1, g_2)$	Kontrast
$\sum_{g_1=0}^{K-1} \sum_{g_2=0}^{K-1} P_{\Delta,\alpha}(g_1, g_2) \times \log[P_{\Delta,\alpha}(g_1, g_2)]$	Entropie
$\sum_{g_1=0}^{K-1} \sum_{g_2=0}^{K-1} \frac{P_{\Delta,\alpha}(g_1, g_2)}{1 + g_1 - g_2 }$	Homogenität

- Lassen sich aus der Co-Occurrence-Matrix berechnen
- Liefern aussagekräftige Kennwerte für verschiedene Texturen

Besonderheiten Texturmerkmal

- Berechnung der Texturmerkmale auf der Basis willkürlicher Regionen.
- Segmentierung
- Berechnung der Zuverlässigkeit
- Erneute Segmentierung

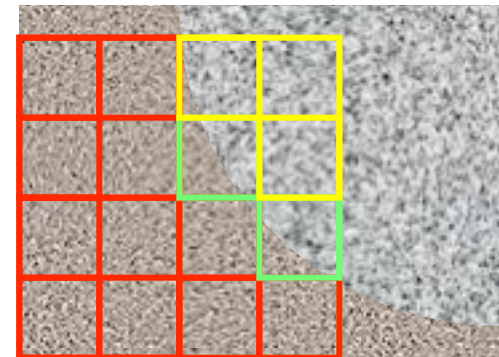
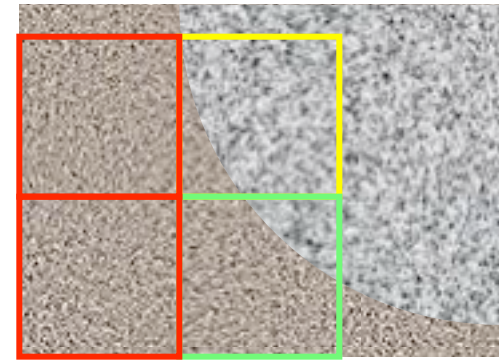


Zuverlässigkeit Texturmerkmal

Persistenz:

Falls eine Region in Teile zerlegt wird und das Merkmal in den Teilregionen berechnet werden kann, dann sollte das Texturmerkmal in der Teilregion dieselben Werte annehmen, wie das der Ursprungsregion.

Achtung: In der Regel ist das berechnete Maß eine Schätzung, deren Güte von der Regionengröße abhängt.

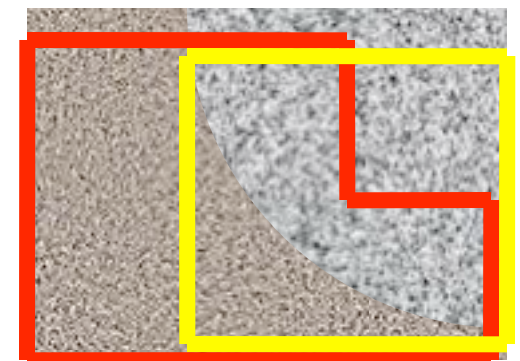
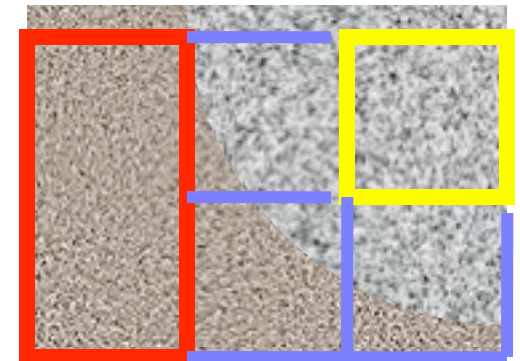


Strategie: Erneute Segmentierung

- Texturmerkmale der Gesamtsegmente erneut berechnen
- Segmente mit hoher Unzuverlässigkeit
 - allen benachbarten zuverlässigen Segmenten zuordnen
 - ähnlichste Kombination selektieren.

Berechenbarkeit von Texturmerkmalen beliebiger Regionen ist vorteilhaft.

Kann über mehrere Auflösungsstufen erfolgen.



Die beiden Extremfälle

Naive Texturesegmentierung

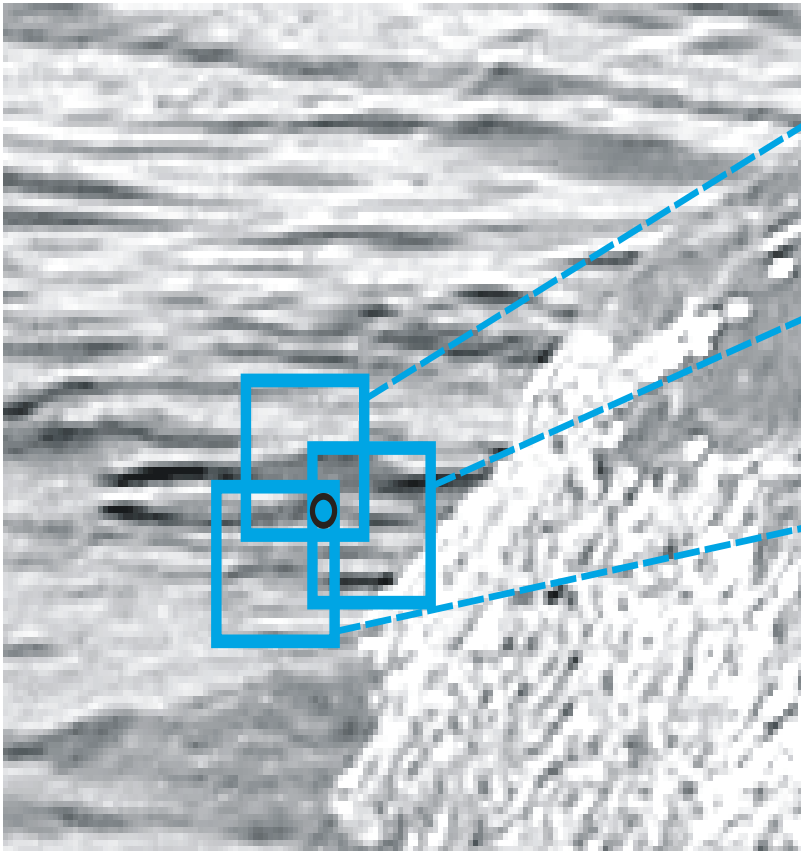
Modell:

- Skalierungsstufe der Textur ist bekannt.
- Blockgröße für die Texturberechnung ist bekannt.
- Geeignetes Texturmaß ist bekannt

Algorithmenskizze:

- Berechne für jedes Pixel das Texturmaß.
- Führe eine homogenitätsbasierte Segmentierung auf den Texturmaßen durch.
- Jedes Pixel erhält für jedes Label, dessen Block das Pixel überdeckt, eine von Null verschiedene Wahrscheinlichkeit.
- Nachverarbeitung ähnlich wie bei Schwellenwertsegmentierung.

Textursegmentierung (naiv)



2.37
1.22

2.82
2.14

2.09
0.87

Summe der Amplituden
mit niedriger und mit
hoher Frequenz

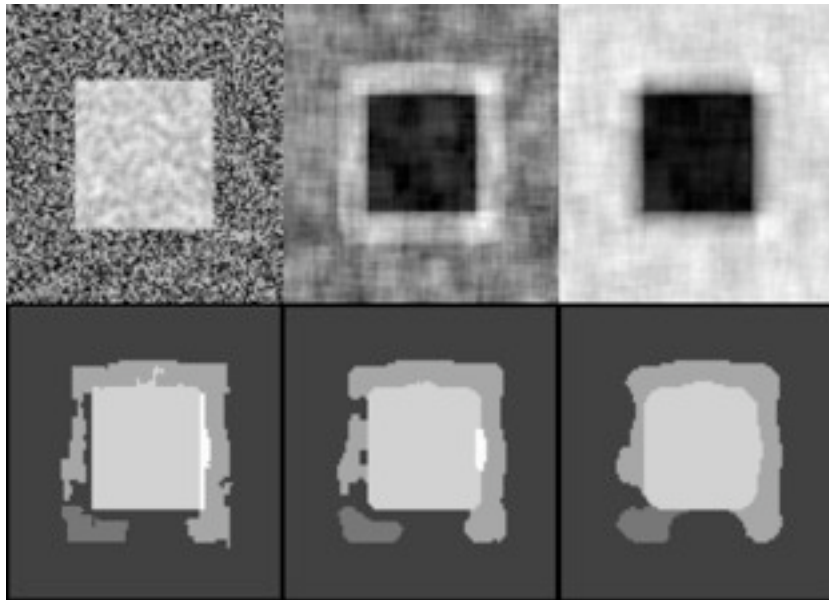
Beispiel:

Jedes Pixel wird von mehreren Blöcken überdeckt.

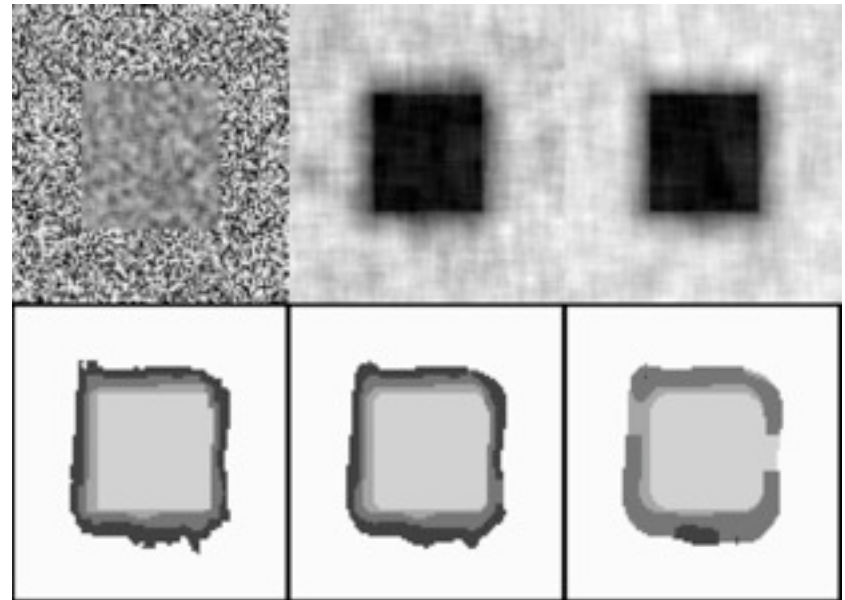
Sind die Texturen unterschiedlich, dann erhalten diese Blöcke unterschiedliche Label.

Nachverarbeitung Medianfilterung

Segmentierung von zwei verschiedenen Texturen

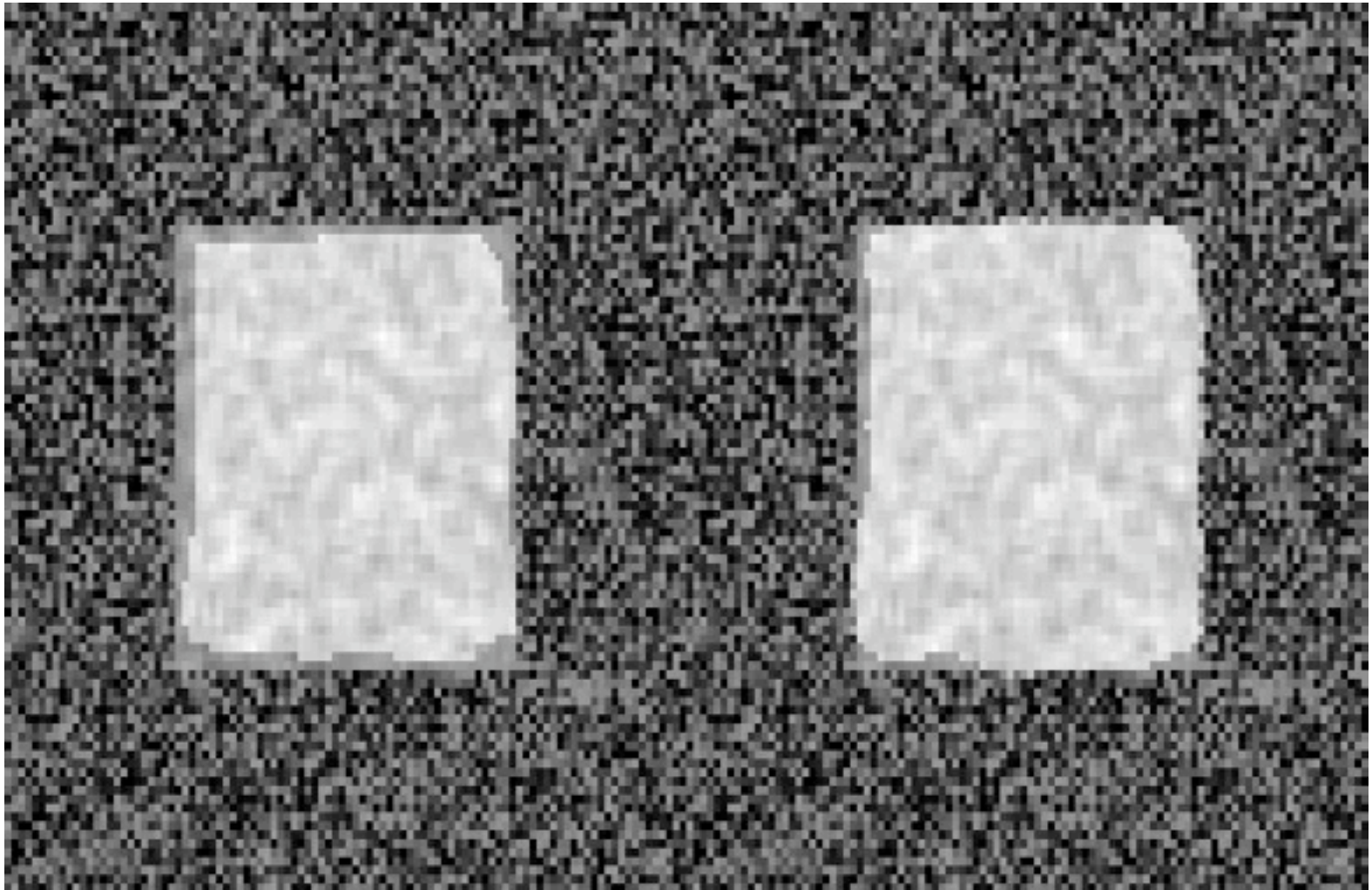


Resultat Median 5×5 Median 11×11



Resultat Median 5×5 Median 11×11

Nachverarbeitung Relaxation Labeling



...Also nochmal bis hierhin...

- Multiskalenstrategie
- Regionenbasierte Segmentierung (Region Merging, Split & Merge)
- Datenstrukturen RAG und Quad-Tree
- Texturen und Textursegmentierung

Kantenbasierte Segmentierung

- Edge Linking und Canny Edge Operator
- Nulldurchgänge
- Wasserscheidentransformation

Segmentierung durch Kantenerkennung

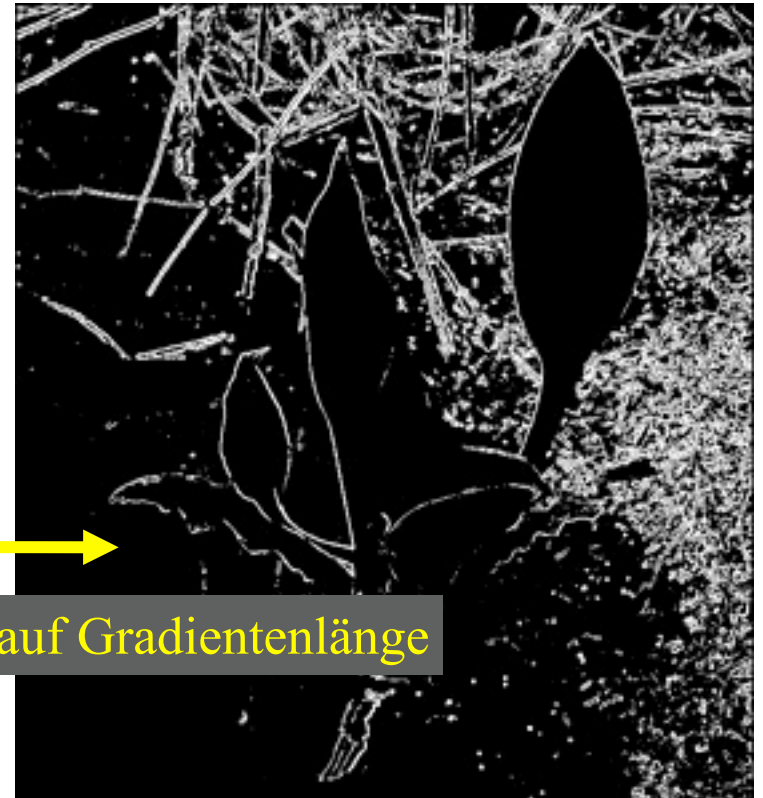
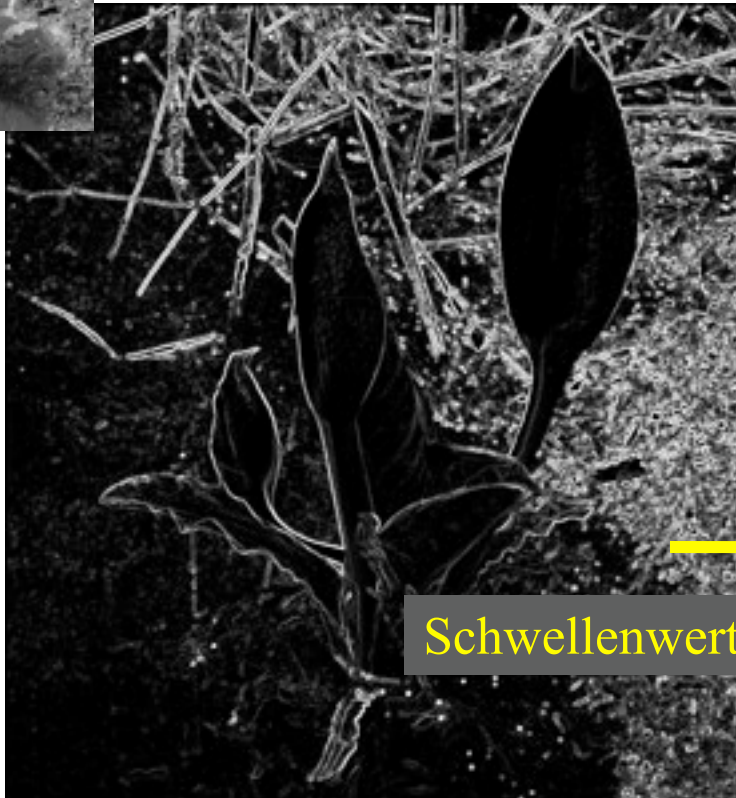


Vorteil: Kantenmerkmale sind robuster gegenüber Shading

Einfache Methode:

- Gradientenberechnung
- Kantenpunktdetektion (z.B. Schwelle auf Gradientenlänge)
- Region Labeling basierend auf Kantenpunkten.

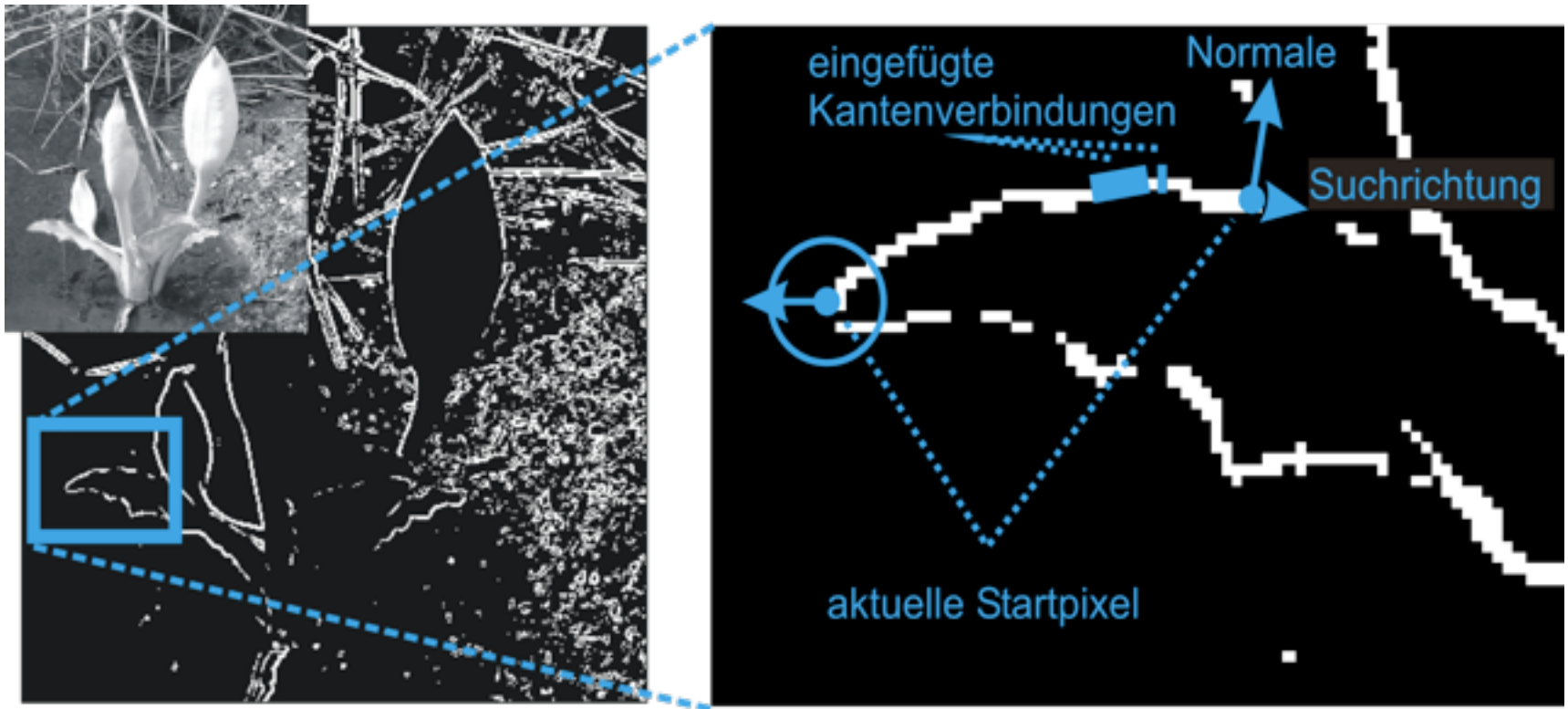
Segmentierung durch Kantenerkennung



Schwellenwert auf Gradientenlänge

Problem: Kantenpunkte sind nicht Ränder zusammenhängender Gebiete.

Edge Linking



Anfangs sind alle Kantenpixel frei und nicht untersucht. Edge Linking sucht sich das nächste nicht untersuchte und freie Kantenpixel und versucht es mit anderen Pixeln zu einem Kantenzug zu verknüpfen.

Edge Linking

1. Suche das nächste Kantenpixel, welches noch nicht als „untersucht“ markiert wurde und erkläre es zum Startpixel eines Kantenzugs.
2. Falls sich in der Umgebung des Kantenpixels in einer der beiden Richtungen orthogonal zur Kantenrichtung unmarkierte Kantenpixel befinden, die eine ähnliche Gradientenrichtung und –stärke aufweisen:
 - a. Markiere die Pixel als zum selben Kantenzug gehörend.
 - b. Erkläre diese Pixel zu neuen Startpixeln.
 - c. Gehe zu 2.
3. Falls sich in der Umgebung markierte Pixel befinden, die den obigen Bedingungen genügen, dann wurde eine Verzweigung von Kanten gefunden.
4. Falls ein Kantenpixel gefunden wurde, gehe zurück zu Schritt 1.

Canny Edge Operator

Ziele:

- möglichst viele Kanten fehlerfrei vom Hintergrund unterscheiden zu können (niedrige Rate von Fehldetektionen).
- (unverzweigte) Kanten genau zu lokalisieren.
- für jede Kante genau eine Detektorantwort zu liefern.

Canny Operator besteht aus Kantenhervorhebung und Erzeugung von Kantenzügen.

Optimale Kantenhervorhebung ist eine Filterung mit einer 1-D abgeleiteten Gaußfunktion orthogonal zur Kante.

Geringfügig schlechtere Ergebnisse erzielt man mit einem 2-D Gradientenoperator auf der Basis abgeleiteter Gaußfunktionen.

Canny Edge Operator

- Es wird immer das Pixel mit größter Gradientenlänge selektiert.
- Startpixel können nur Pixel sein, deren Gradientenlänge oberhalb einer Signifikanzschwelle T_1 liegt.
- Neue Kantenpixel werden in den Kantenzug eingefügt, wenn ihre Gradientenlänge größer als T_2 ist ($T_2 < T_1$).
- Das Verfahren endet, wenn keine neuen Startpixel gefunden werden.

Nulldurchgänge

Die Orte der Nulldurchgänge der zweiten Ableitung sind Ränder zusammenhängender Gebiete.

Methode:

- Laplace-Operator
- Nulldurchgänge bestimmen:

$$\nabla^2(f(i,j)) \cdot \nabla^2(\text{shift}(f(i,j))) \leq 0$$

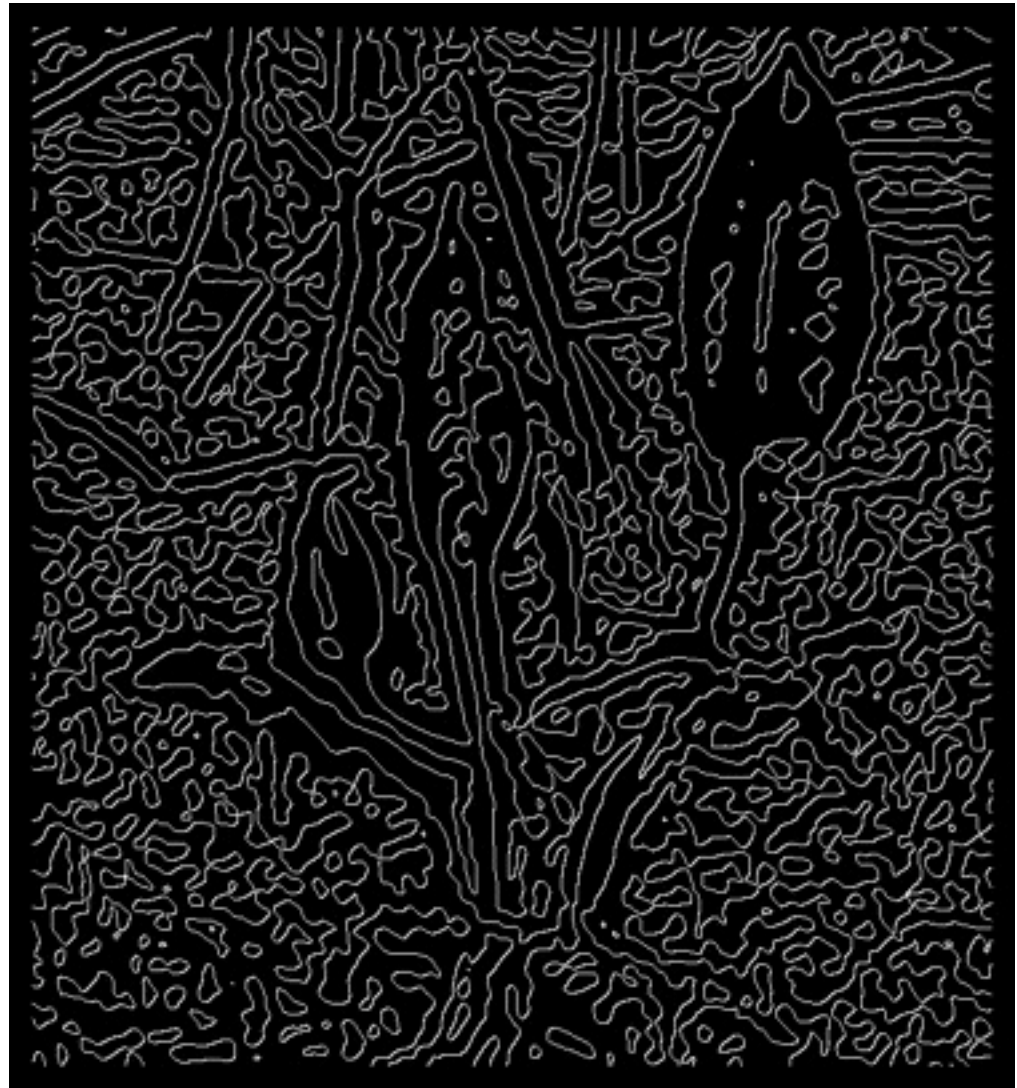
(shift: Verschiebung des Bilds um ein Pixel in jede Richtung)



$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} + \frac{\partial^2 f(x, y)}{\partial x \partial y} + \frac{\partial^2 f(x, y)}{\partial y \partial x} : \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Nulldurchgänge

Kombination des Laplace-Operators mit Glättungsoperator (z.B. als LoG-Operator) reduziert die Anzahl der Nulldurchgänge



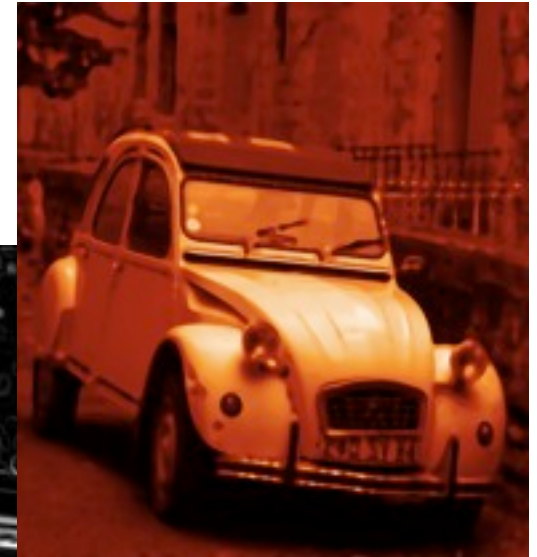
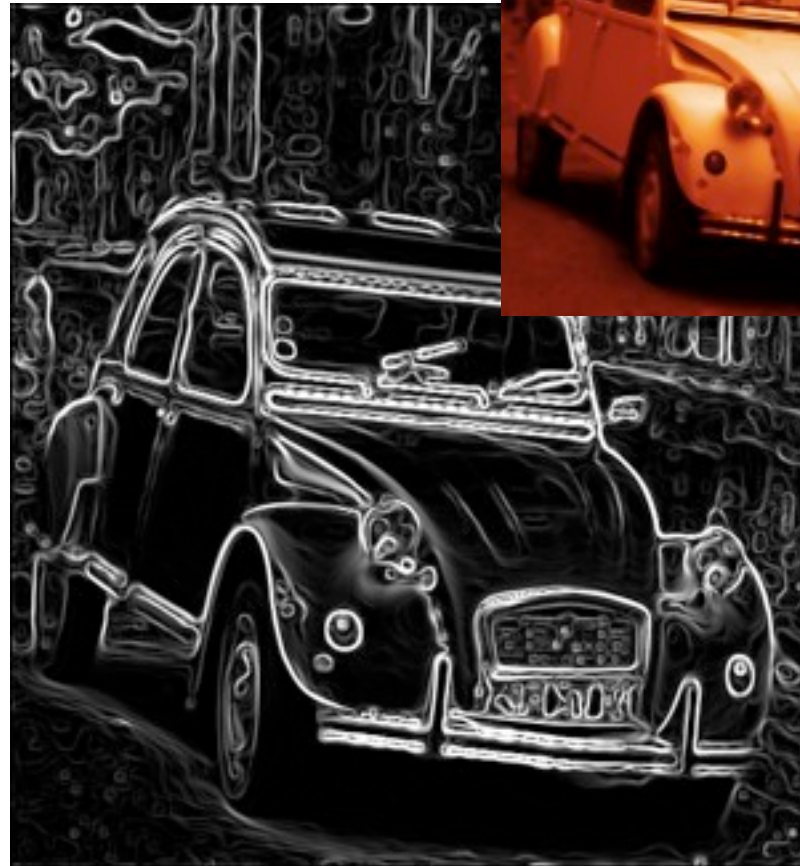
Wasserscheidentransformation

- Wasserscheide: Menge aller Orte, die die Grenzen der Entwässerung in unterschiedliche Senken sind.
- Beispiel: Wasserscheide zwischen Nordsee und Mittelmeer verläuft entlang des Kamms der Berner Alpen.
- Wasserscheide in der Segmentierung: Generiere Höhenprofil so, dass Wasserscheiden gerade die gesuchten Segmentgrenzen sind.



Wasserscheiden

- Wasserscheiden sollen an Kanten verlaufen.
- Wasserscheiden sind „Gebirgskämme“
- ▶ Wasserscheiden sind die Längen der Grauwertgradienten.



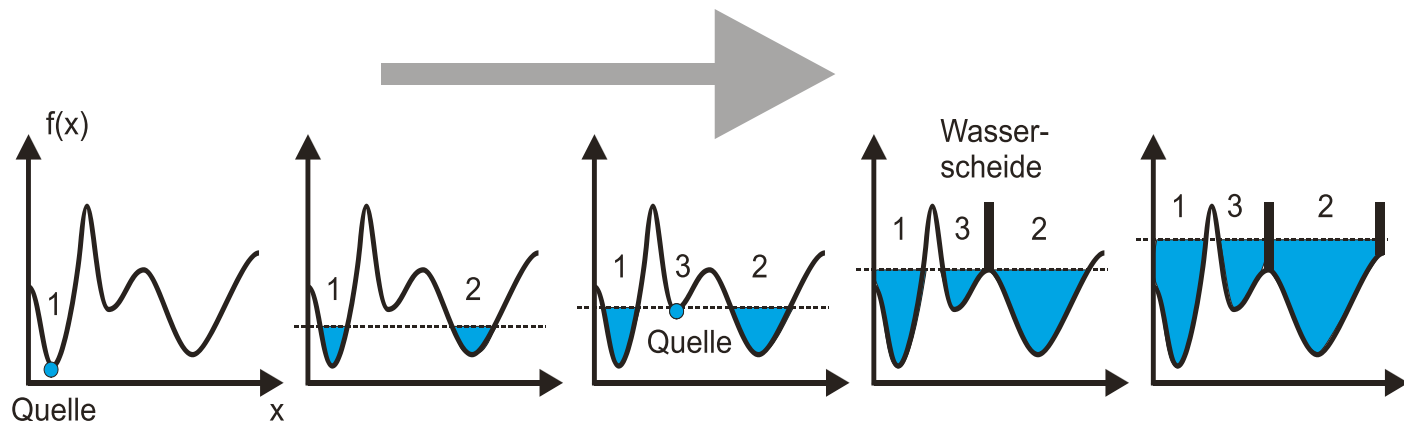
Wasserscheidentransformation

- Berechnung

Es fällt „Regen“ auf jedes Pixel. Anhand des Gradienten wird entschieden, wohin der Regen entwässert wird.

- Flutung

Die „Welt“ wird von den Senken her geflutet. Immer wenn Wasser aus zwei Senken zusammen fließt, entsteht eine Wasserscheide.



Flutungsalgorithmus (Skizze)

Jedes neu überflutete Pixel (m_f, n_f) ist

- **in Isolation:**

Es nicht zu anderen überfluteten Pixeln der Höhen $h < h_{\text{aktuell}}$ benachbart.

Isolierte Pixel sind Kerne von neuen Segmenten.

- **Erweiterung:**

Es ist zu anderen überfluteten Pixeln der Höhen $h < h_{\text{aktuell}}$ mit gleichem Label benachbart.

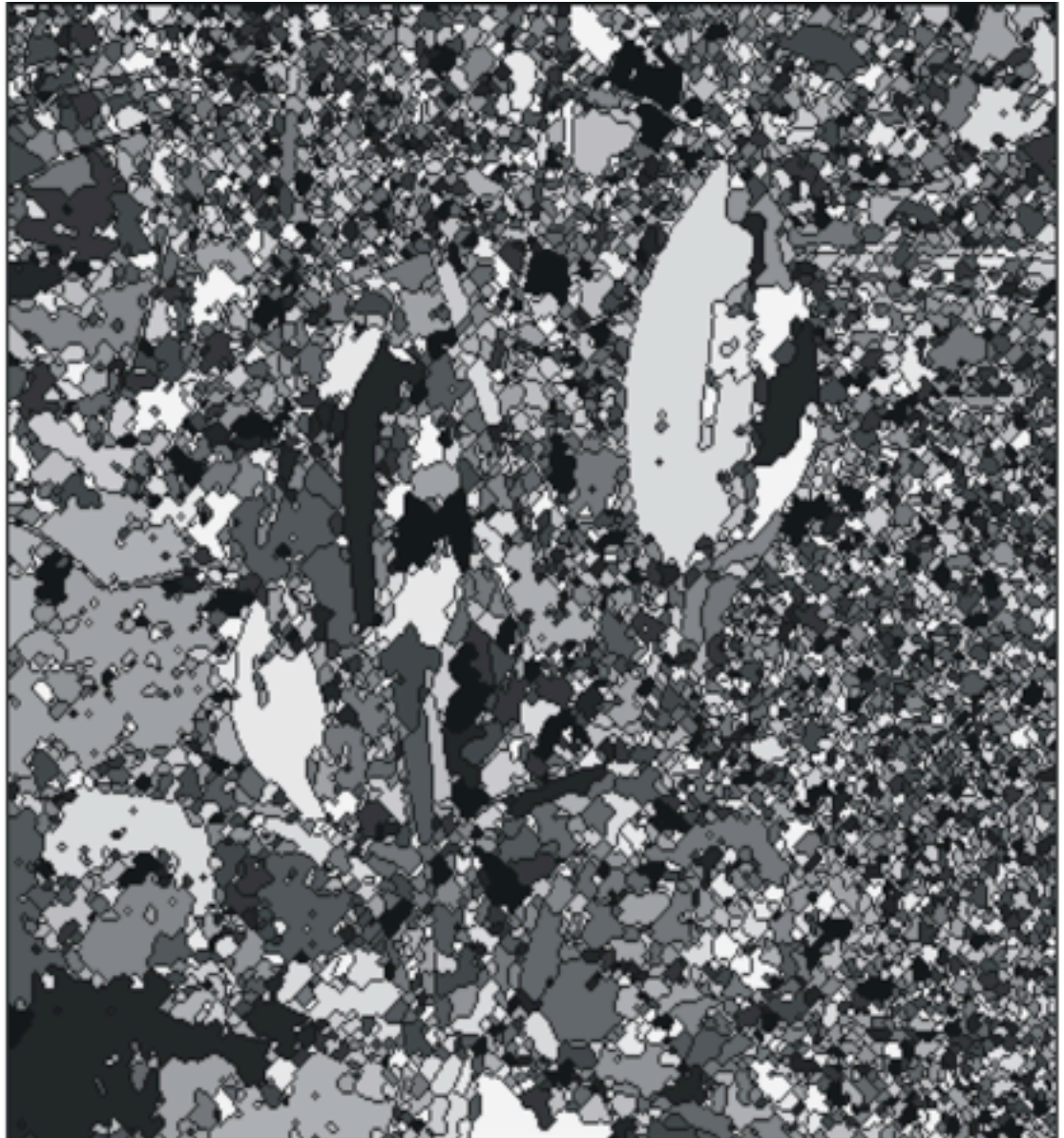
Das Pixel wird dem Segment mit diesem Label zugeordnet.

- **Wasserscheide:**

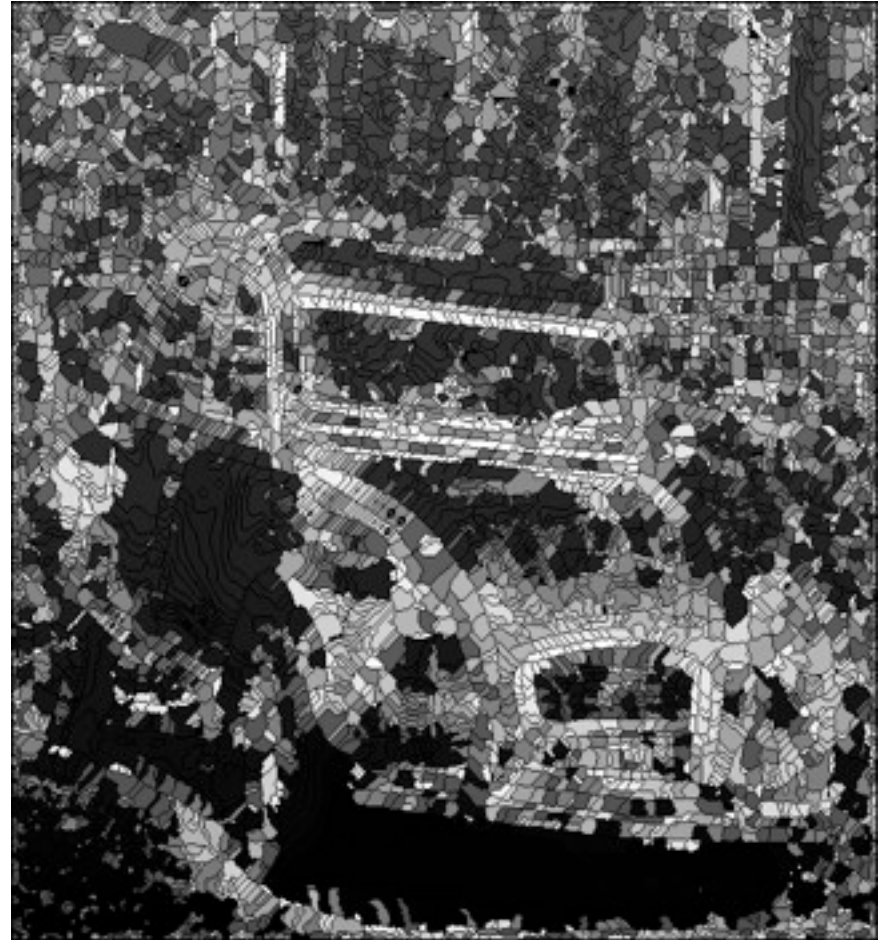
Es ist zu überfluteten Pixeln von mindestens zwei Regionen benachbart.

Dem Pixel wird das Label „Wasserscheide“ zugeordnet.

Resultat der WST

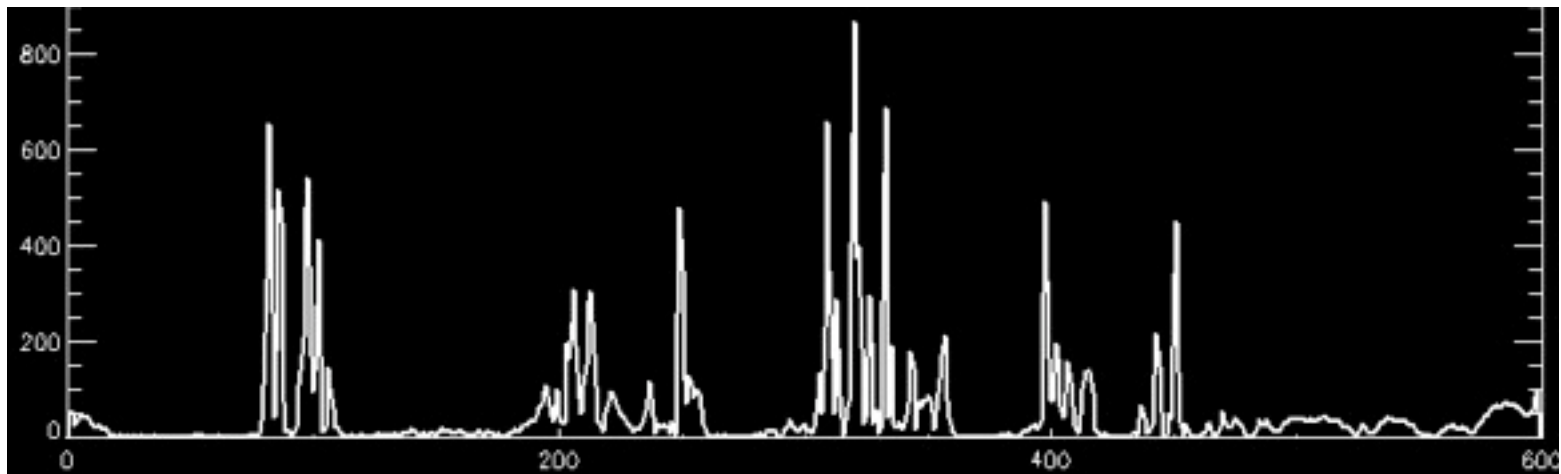


Resultat



Problem Übersegmentierung

- Für die WST ist jedes lokale Minimum eine Senke.
- Die meisten Senken werden durch Rauschen verursacht.
- Senken durch Rauschen sind weniger tief als die von Kanten.

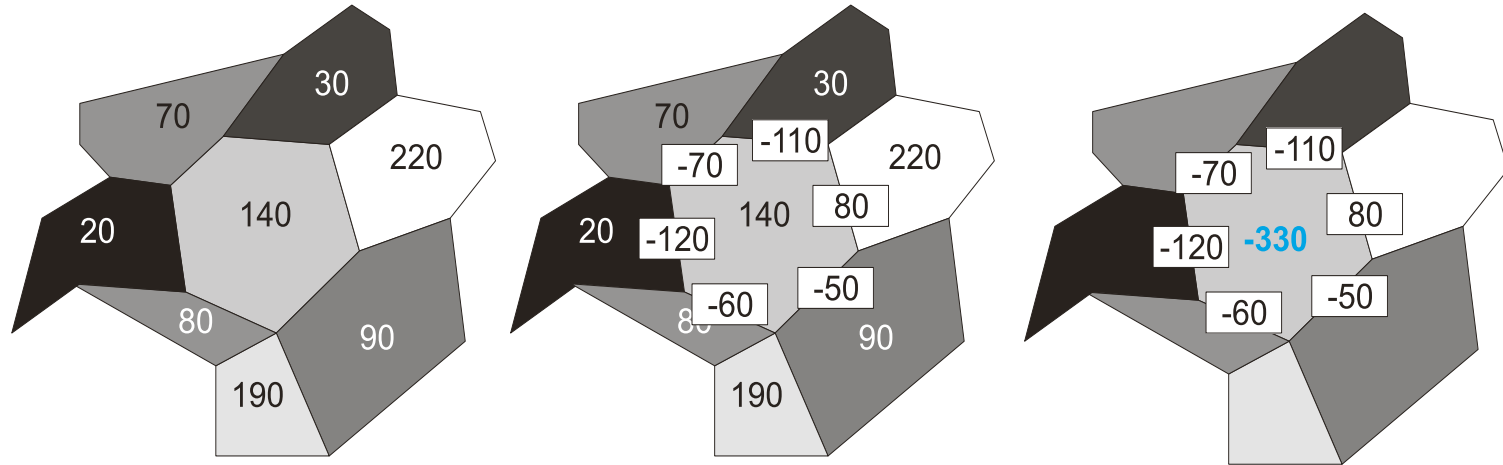


Hierarchische WST

Multiskalenstrategie:

- Wasserscheidentransformation auf dem WST-Resultat.
- Jede Region erhält ihren durchschnittlichen Grauwert als Funktionswert.
- Die erste WST wird hauptsächlich durch Rauschen verursachte Senken finden.
- „Wahre“ Senken sollten über mehrere Stufen der Hierarchie erhalten bleiben.

Gradienten für die hWST



- Zu jeder der benachbarten Regionen wird die Differenz berechnet.
- Die Länge des Gradienten ist die durchschnittliche Differenz zu allen Regionen.
- Die Richtung ergibt sich aus den (mit der Regionengröße gewichteten) Vektoren zwischen dem Schwerpunkt der Region und den Schwerpunkten aller benachb. Regionen.

...nochmal zum langsam mitmeißeln...

- Edge Linking und Canny Edge Operator
- Nulldurchgänge zur Regionensegmentierung
- Wasserscheidentransformation
 - Ausführung
 - Probleme
 - Multiskalenstrategie

Literatur

- Klaus D. Tönnies: "Grundlagen der Bildverarbeitung", ISBN 3-8273-7155-4
- <http://www.pearson-studium.de/main/main.asp?page=booksites/selectchapter&isbn=3827371554&PSZielgruppe=Student>