

BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

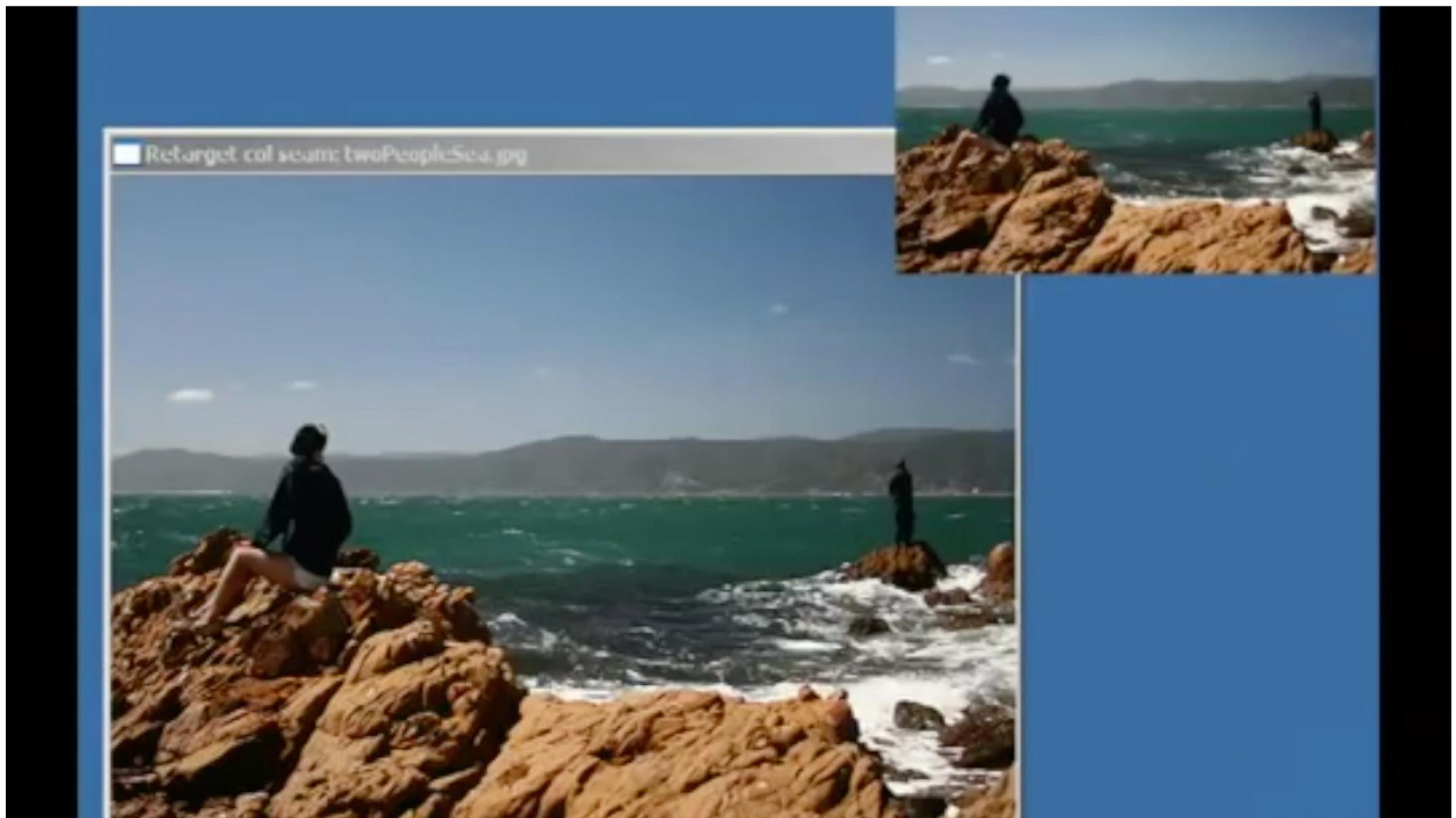
Projekt: Seam Carving

Einführung in Bildverarbeitung

Bachelor Medieninformatik
Wintersemester 2019/20

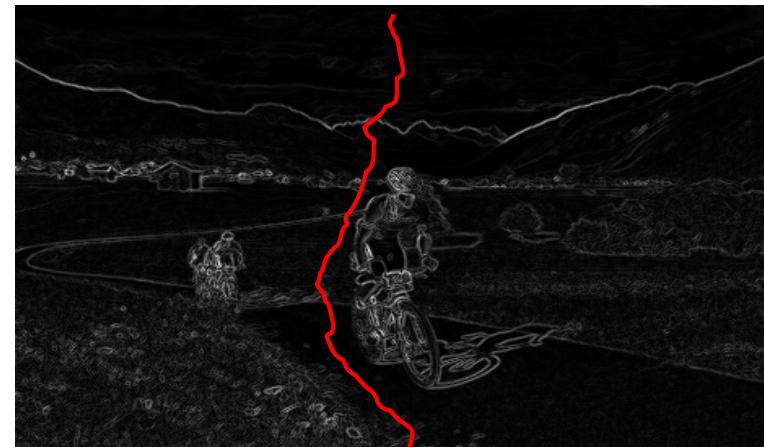
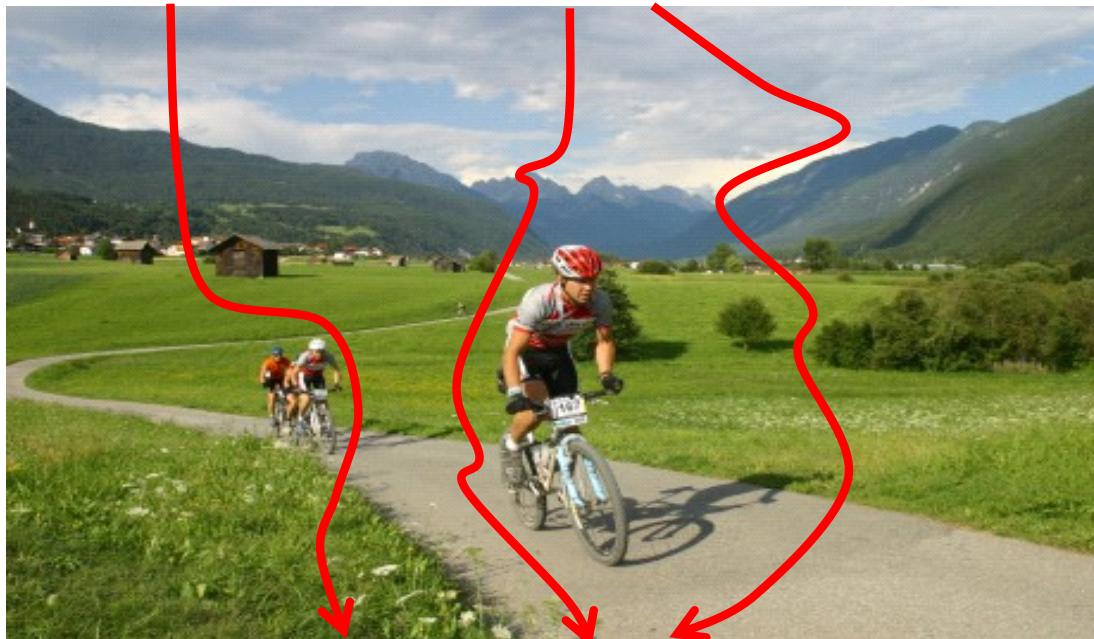
Prof. Dr.-Ing. Kristian Hildebrand
khildebrand@beuth-hochschule.de

Seam Carving



Ziel: Entfernen eines Pixelpfades

- Was macht einen geeigneten Pixelpfad aus?
 - Wo sind wichtige Änderungen im Bild?



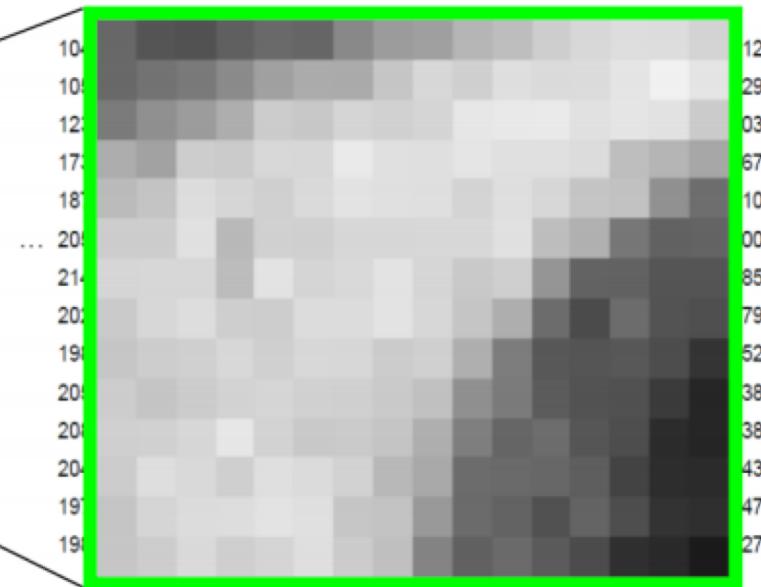
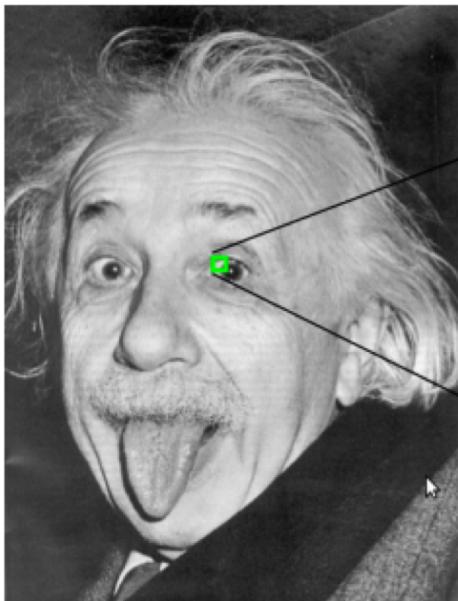
Welche Informationen im Bild sind wichtig?
Wo ist die geringste Energie, um einen Pfad zu entfernen?

- Wie findet man einen optimalen Pixelpfad schnell?
 - Finden des optimalen Pfades durch Dynamische Programmierung (*nächste Woche*)

Lernziele

- Was sind Bilder?
 - Bild als Matrix vs Bild als Graph?
 - Einfluß von Farträumen
- Faltung / Convolution
- Filterung von Bildern
- Bildgradienten

Was sind Bilder?



D. Schlesinger, TU Dresden

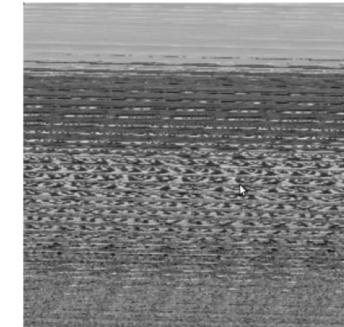
Matrizen

Darstellung auch als Vektoren

$$A_{n \times m} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

Demo! - Bilder mit Scipy

Bilder sind Matrizen



D. Schlesinger TU Dresden

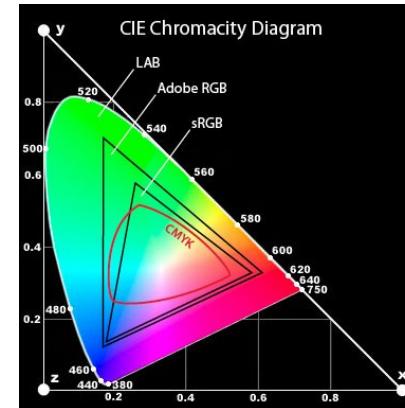
- ähnliche Vektoren sind nicht unbedingt ähnliche Bilder
- ähnliche Bilder sind nicht unbedingt ähnliche Vektoren
- Darstellung, geometrische Transformationen, Farbtransformationen (RGB vs. LAB Farbraum)



RGB - Farbraum



LAB - Farbraum

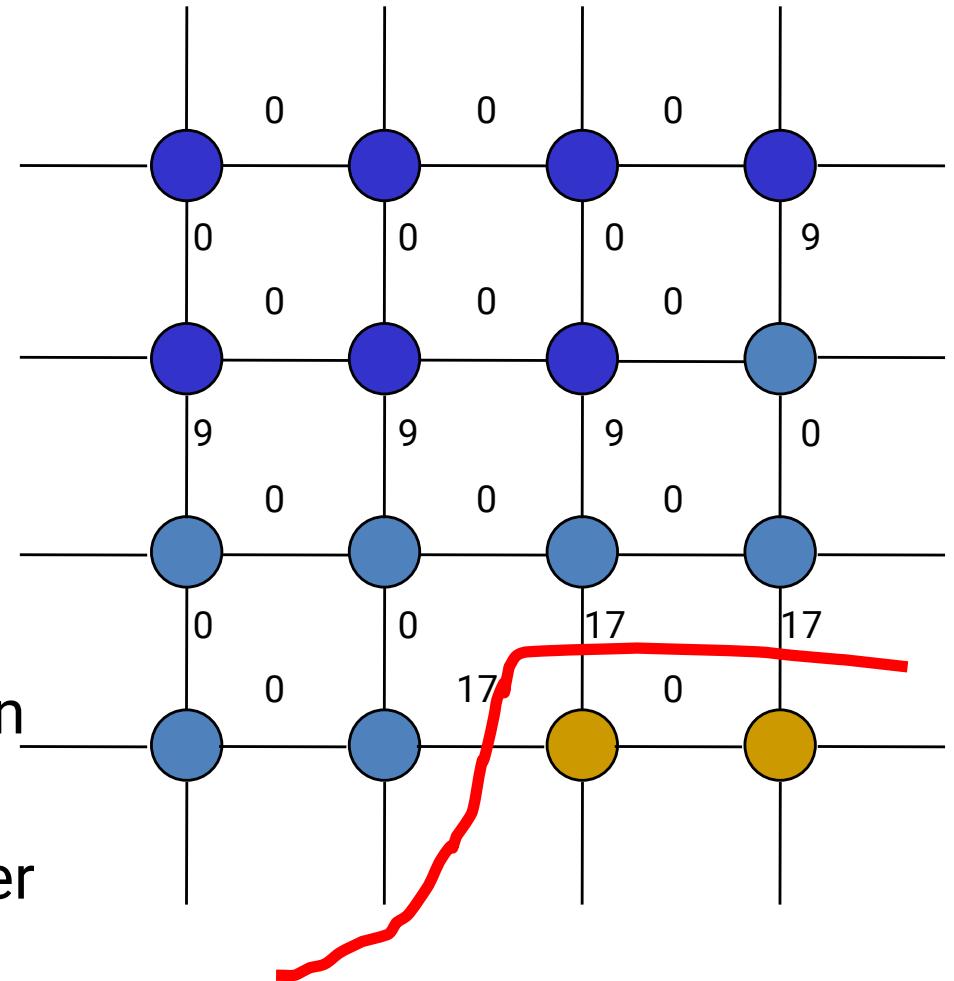


<https://www.photo.net/learn/using-lab-color-adjustments/>

Distanzen zwischen Farben immer besser im LAB – Farbraum berechnen (Differenzen korrespondieren zur menschlichen Wahrnehmung)

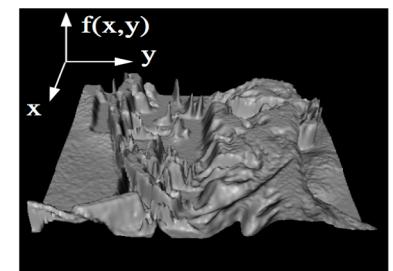
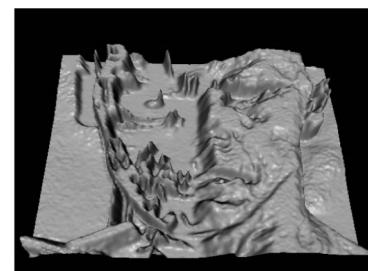
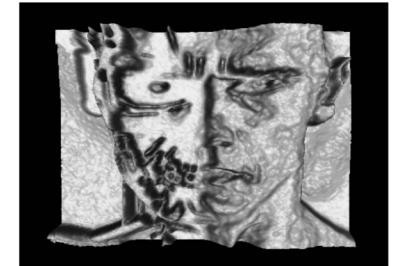
Bilder sind Graphen

- Pixel sind Knoten mit 4- (oder 8-) Nachbarschaft
- Kanten sind Gewichte
 - z.B. Helligkeitsunterschiede zwischen Pixeln
- Graph ist das Gitter
 - attributierte, gelabelte Graphen etc.
 - Wird eher für höhere Stufen der Verarbeitung (bei Computer Vision) verwendet



Bilder sind Funktionen

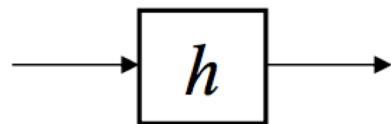
- sind z.B. konvex, stetig und differenzierbar
- $I(x,y)$ gibt die Helligkeit an der Stelle (x,y)
- Veränderung des Bildes durch Transformationsfunktion



N. Snavely



$$\bullet \quad g(x) = h(f(x))$$



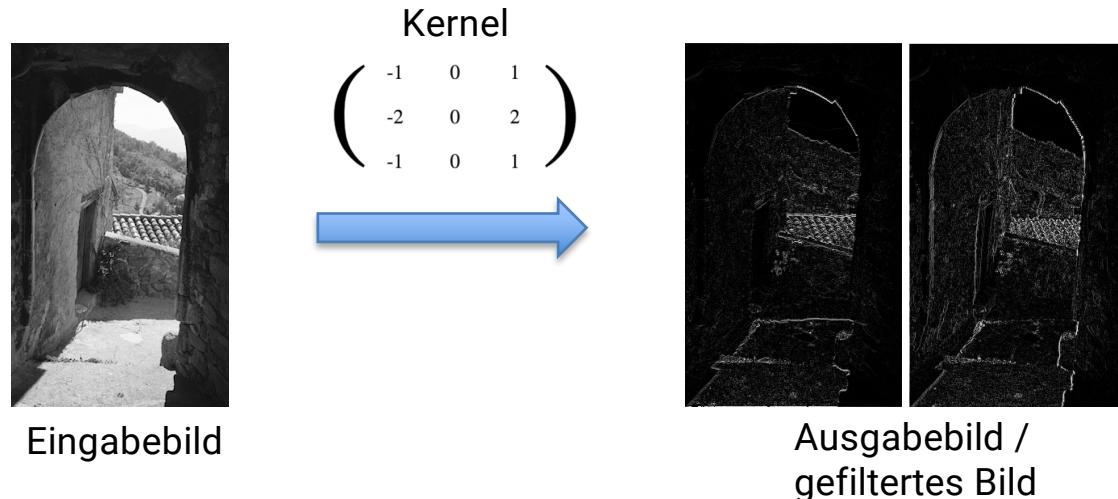
Fredo Durand, MIT

Invertieren
Histogrammanpassung
Kontrastanpassung
etc.

FFT Demo

Faltung des Bildes (Convolution)

- Dient zur Filterung der Frequenzcharakteristika des Bildes
 - genereller Ansatz zur Erzeugung von Bildeffekten
 - Filterung des Bildes durch ein Strukturelement (Kernel)



- Funktioniert durch die Bestimmung des neuen Farbwertes (Grauwert) eines zentralen Pixels als gewichtete Summe seiner Nachbarn

Faltung des Bilder (Convolution)

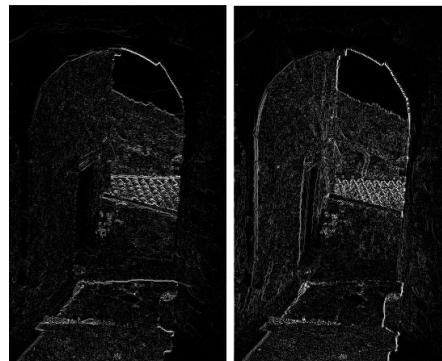
- Faltung ist dabei die Multiplikation eines Pixels und seiner Nachbarn mit der Kernelmatrix

$$I' = \sum_{u,v} I(x-u, y-v) g(u, v)$$



$$g(u,v) = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

Eingabebild



Demo

<http://setosa.io/ev/image-kernels/>

Was passiert an den Rändern?

- padding
- zero

Jetzt einmal selbst coden!
Rechner raus ☺

Beispielhafte Anwendungsbereiche der Convolution (Faltung)

Morphologische Operationen

- einfache Operationen auf der Form des Bildinhaltes
 - wird meist auf Binärbildern ausgeführt
 - Binärbilder erhält man z.B. über einen Schwellwert auf den Helligkeiten des Bildes **Vorverarbeitungsschritt**
 - Strukturelement (Kernel) verändert die Form

Hinweis: OpenCV – Tolle Bildverarbeitungsbibliothek



Closing



Opening

Morphologische Operationen

Original



Erosion



Erosion:

Pixel wird 1, wenn alle
Pixel unter dem Kernel
1 sind
(Rauschentfernung)

Dilation



Dilation:

Pixel wird 1, wenn min. ein
Pixel unter dem Kernel 1
ist
(Rauschunterdrückung,
erst Erosion (Noise), dann
Korrektur durch (Dilation))

Opening



Opening:

Zusammengefasste
Operation von Erosion
gefolgt von Dilation

Demo in Scipy/Numpy

Closing



Closing:

Zusammengefasste
Operation von Dilation
gefolgt von Erosion

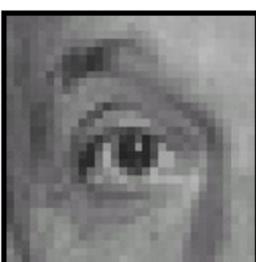
Lineare Filter

 $*$

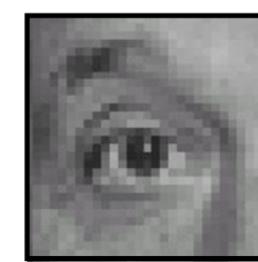
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

 $=$ 

Identische Transformation

 $*$

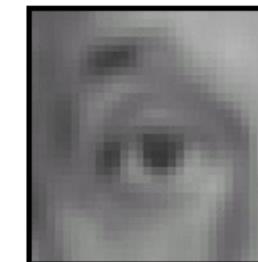
$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

 $=$ 

Bildverschiebungstransformation

 $*$

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

 $=$ 

Blur / Glättungsfilter (mit Durchschnittsfilter)

 $*$

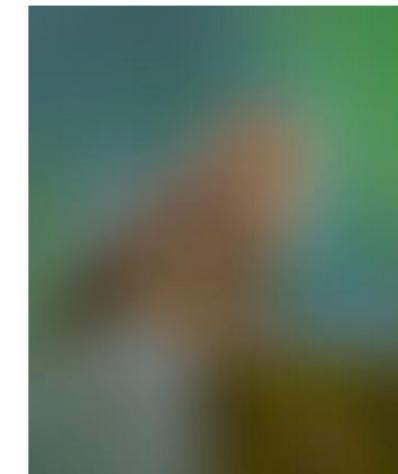
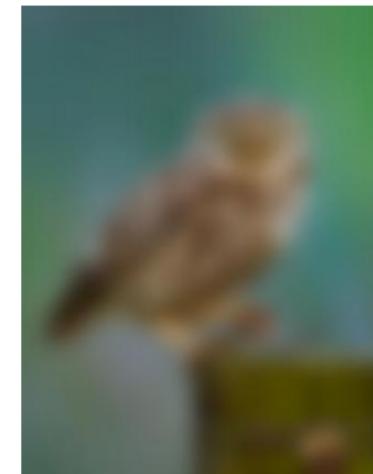
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$- \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

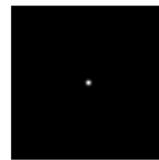
 $=$ 

Schärfungsfilter

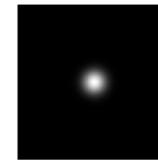
Gauss-Filter



N. Snavely



$\sigma = 1$ pixel



$\sigma = 5$ pixels



$\sigma = 10$ pixels

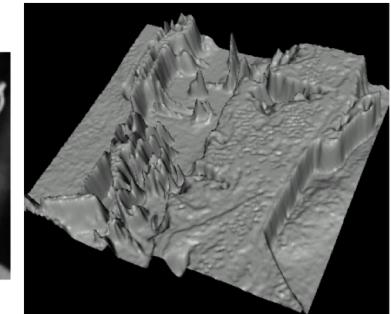


$\sigma = 30$ pixels

Bildgradienten

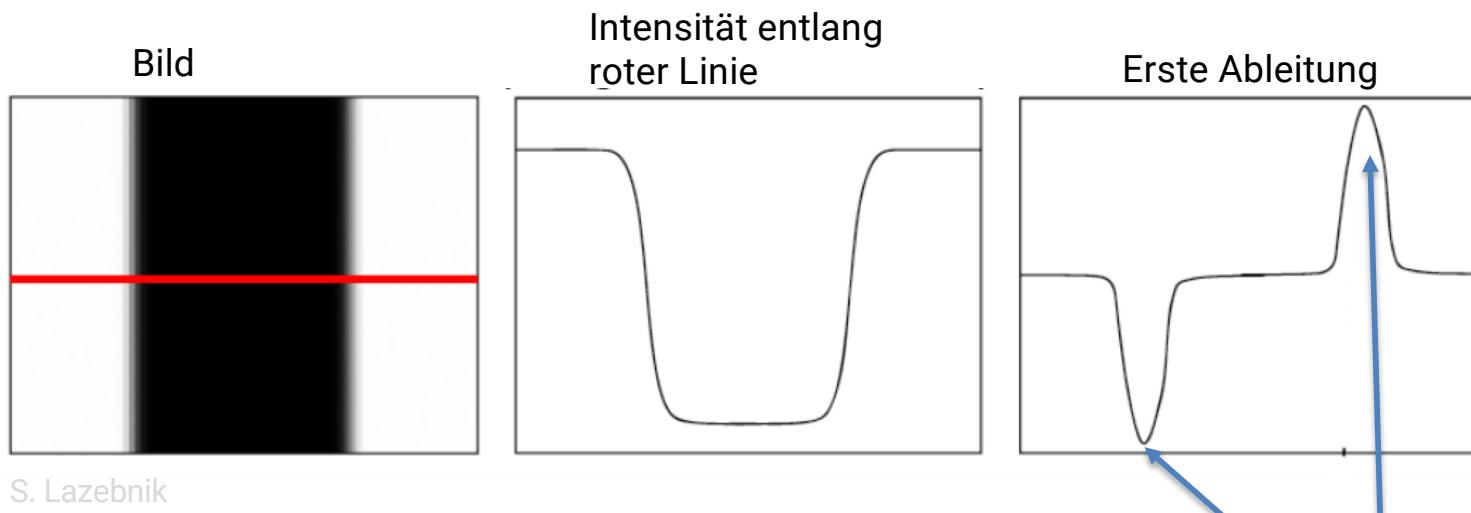
Bildgradienten

- Kanten sehen aus wie steile Abhängigkeiten



N. Snavely

- Orte plötzlicher Änderung der Bildintensität



S. Lazebnik

Kanten korrespondieren mit Extrema der 1. Ableitung

Bildgradienten

- Berechnung der diskreten Ableitung über finite Differenzen

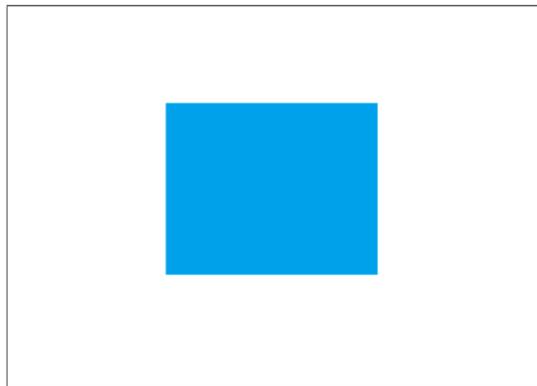
$$\frac{\partial f(x, y)}{\partial x} \approx \frac{f[x + 1, y] - f[x, y]}{1} \quad \text{partielle Ableitung in x-Richtung}$$

$$\frac{\partial f(x, y)}{\partial y} \approx \frac{f[x, y + 1] - f[x, y]}{1} \quad \text{partielle Ableitung in y-Richtung}$$

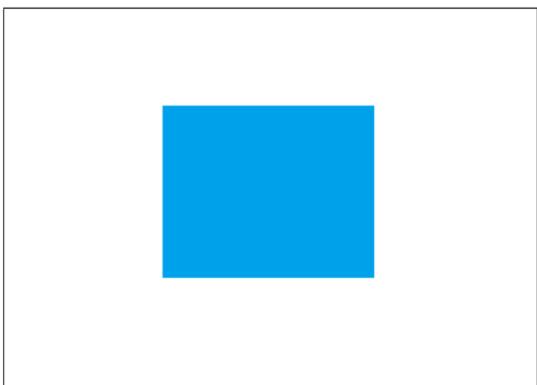
- Wie könnte dafür ein Convolution-Filter aussehen?

Bildgradienten

- Convolution-Filter $[-1,1]$ (horizontal), $[-1,1]^T$ (vertikal)



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



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

Finite Differenzen Filter

Prewitt: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

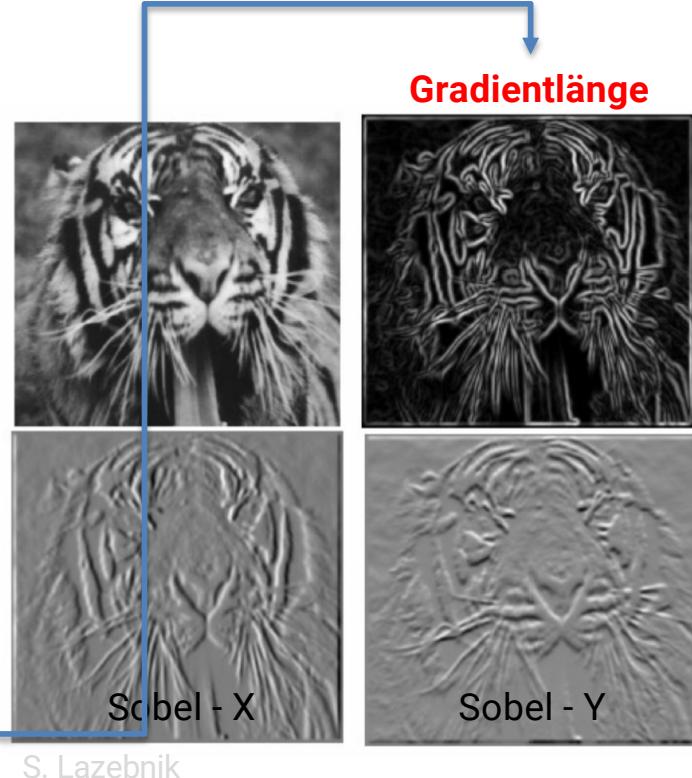
Sobel: $M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Roberts: $M_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$; $M_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

K. Grauman

Demo

Wichtige Änderungen im Bild!



S. Lazebnik

Bildgradient als 2D Vektor:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

$$\nabla f = \left[\frac{\partial f}{\partial x}, 0 \right]$$

$$\nabla f = \left[0, \frac{\partial f}{\partial y} \right]$$

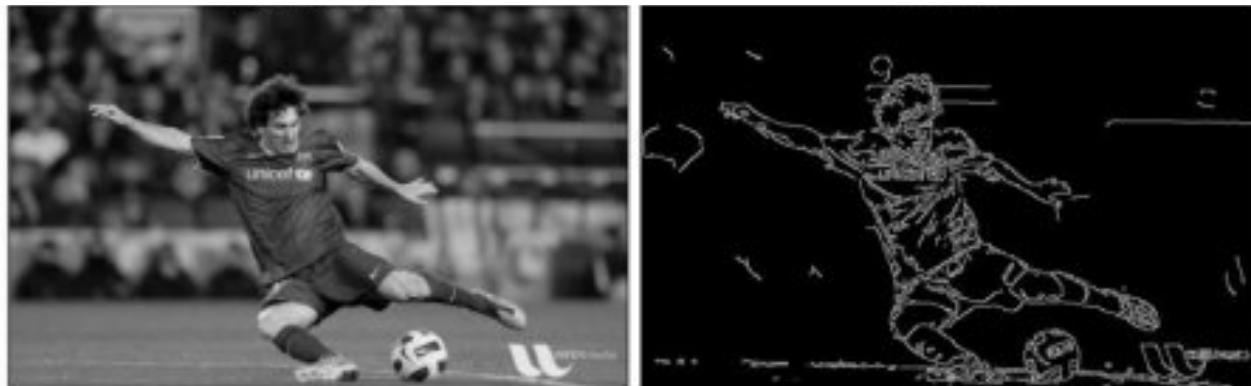
$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$

Gradientenrichtung:
Richtung der Kantennormalen

$$\theta = \tan^{-1}\left(\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}}\right)$$

Gradientenlänge: $\|\nabla f\| = \sqrt{\frac{\partial f^2}{\partial x} + \frac{\partial f^2}{\partial y}}$
Länge des Richtungsvektors

Canny-Edge Stabile Kantenextraktion



Canny-Edge / Stabile Kantenextraktion

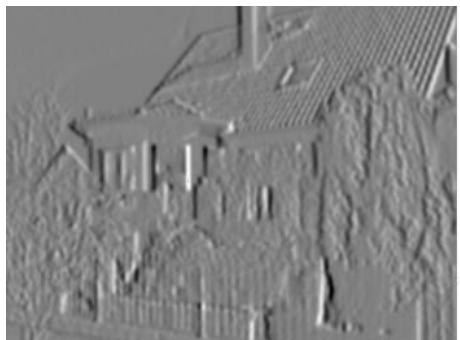
- 4 Schritte
 1. Rauschfilterung durch Gauss-Filter



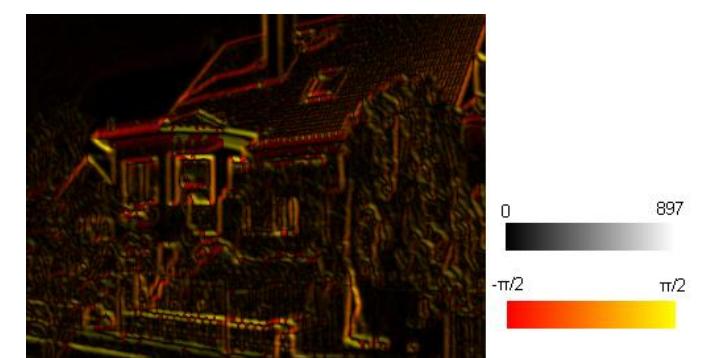
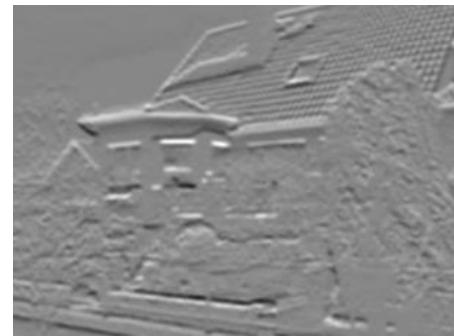
© wikipedia



2. Gradientlängen berechnen



© wikipedia

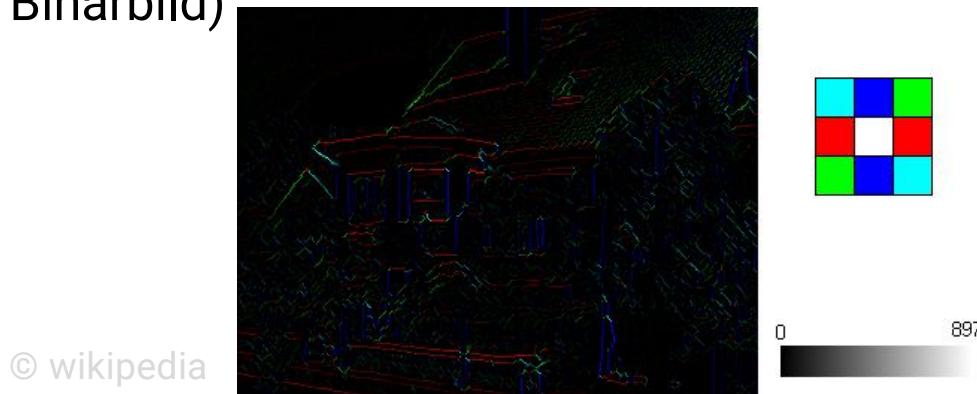


© wikipedia

Canny Edge

3. Non-maximum suppression:

- Kanten soll nicht dicker als ein Pixel breit sein
- Entlanglaufen auf der Kante und einen Erosionsansatz verfolgen
(nur nicht auf dem Binärbild)



© wikipedia

4. Hysterese:

- Schwellenwertanalyse, um herauszufinden, ob ein Pixel noch zu einer Kante gehört
- startet mit Intensität $> T_2$ und verfolgt Kante solange Intensität $> T_1$

Seam Carving

Energieminimierung

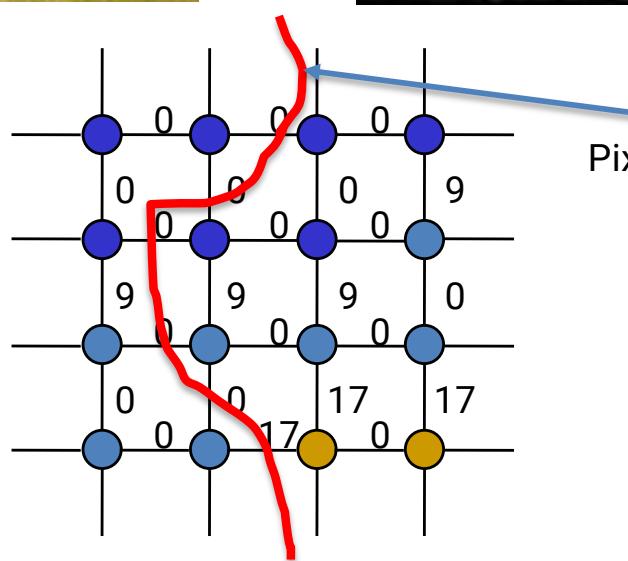
Ausgangsbild



Energie



Bild als Graph



Pixelpfad mit minimaler Energie

Das wars für heute!