

# Robotik

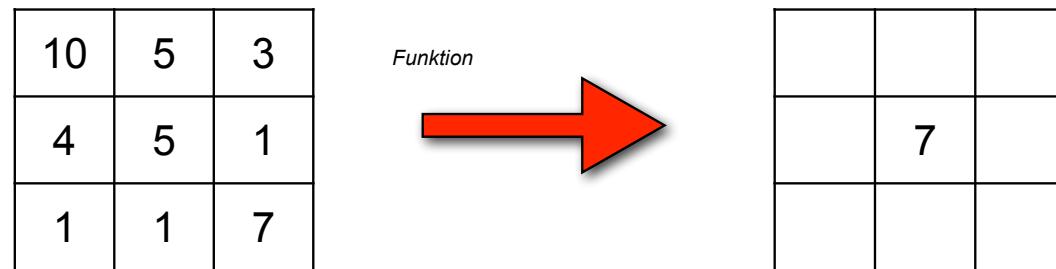
Priv.-Doz. Dr. Thomas Wiemann  
Institut für Informatik  
Autonome Robotik

SoSe 2021



## Wdh.: Bildfilter

- ▶ Ersetze die Pixel eines Bildes durch einen neuen Wert der durch funktionale Auswertung der Nachbarpixel entsteht



- ▶ Lineare Kernel sind die einfachsten und wirkungsvollsten
  - Ersetze einen Pixel durch eine Linearkombination der Nachbarn
  - Kodiere die Parameter in einer  $N \times N$ -Matrix (Kernel)
  - „Faltung des Bildes mit einem Kernel“

$$\begin{array}{|c|c|c|} \hline 10 & 5 & 3 \\ \hline 4 & 5 & 1 \\ \hline 1 & 1 & 7 \\ \hline \end{array}
 \otimes
 \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0.5 & 0 \\ \hline 0 & 1.0 & 0.5 \\ \hline \end{array}
 =
 \begin{array}{|c|c|c|} \hline & & \\ \hline & 7 & \\ \hline & & \\ \hline \end{array}$$

## Wdh.: Verfahren 1. Ordnung

Betrachten den Gradienten der Bildfunktion, d.h.

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{pmatrix}$$

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

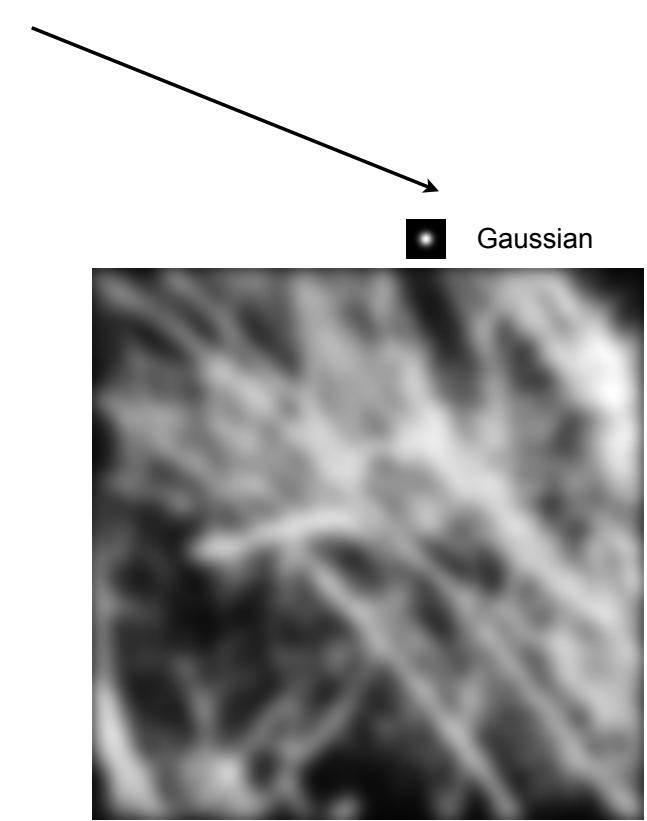
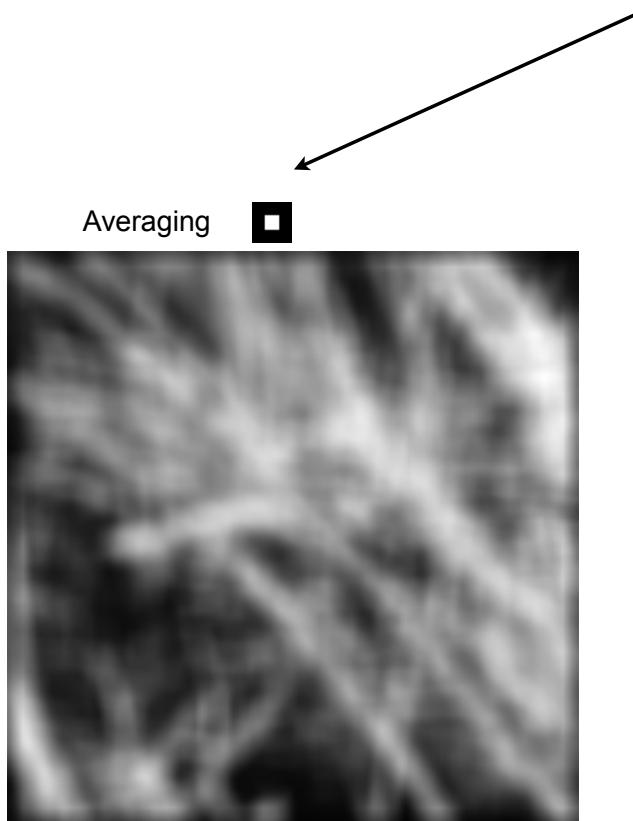
ist Maß für die Stärke des Anstiegs

Richtung des Anstiegs:  $\Phi = \arctan\left(\frac{\partial f / \partial y}{\partial f / \partial x}\right)$

Approximation der Ableitung durch diskrete Differenzen:

$$\frac{\partial f}{\partial x} \approx \frac{f(x+1, y) - f(x-1, y)}{2}$$

## Wdh.: Weichzeichnen

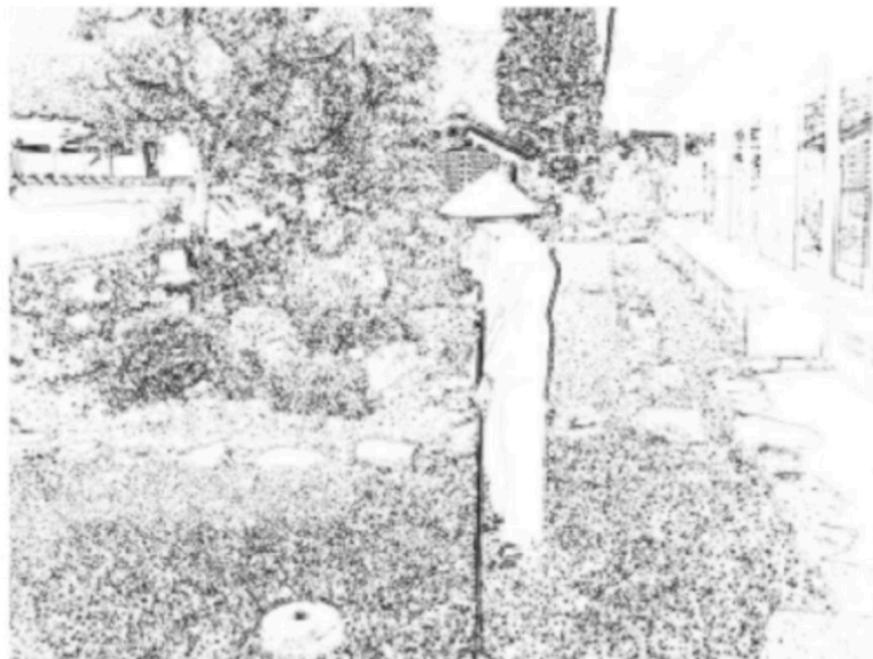


## Wdh.: Laplace-Filter

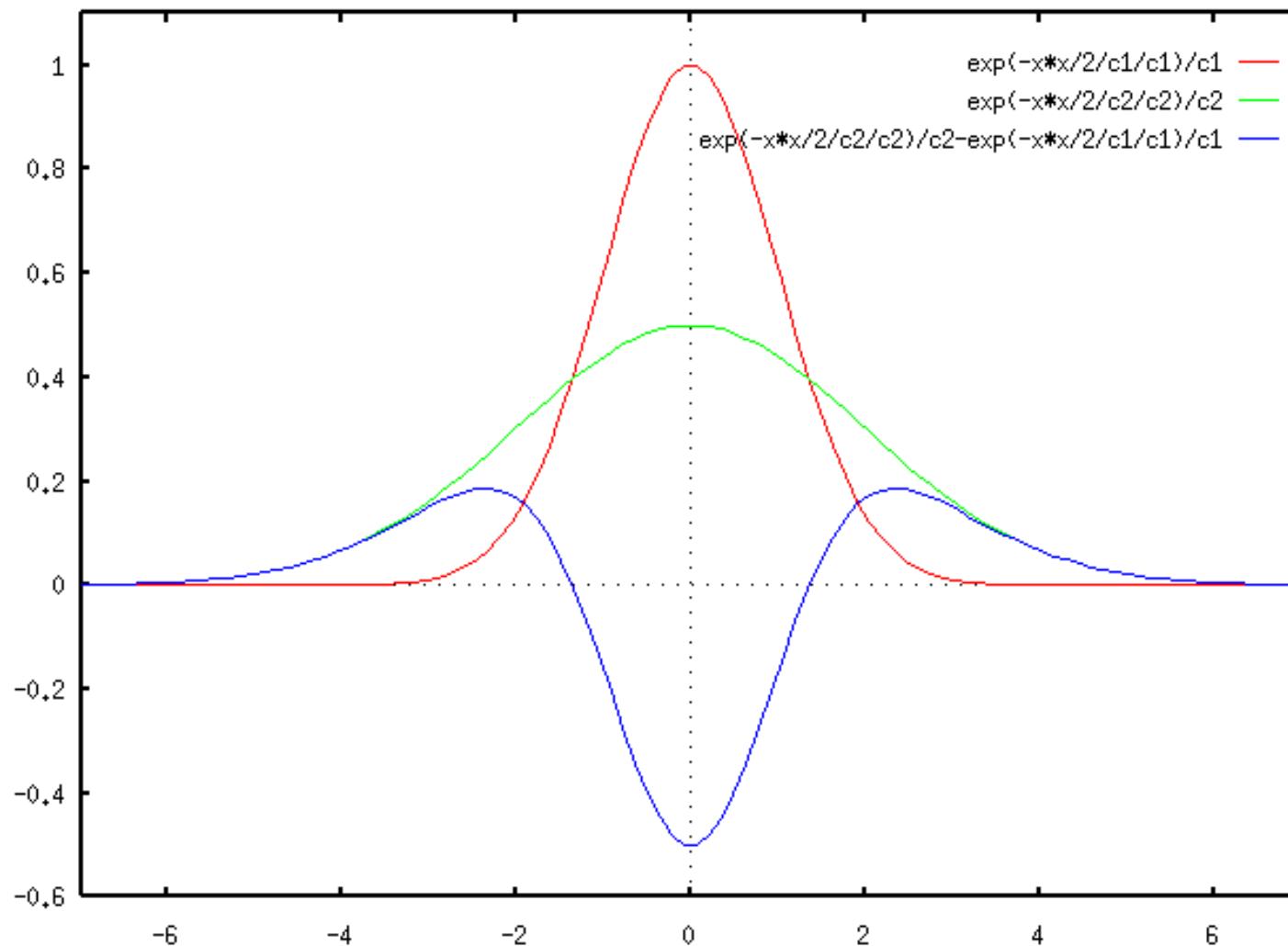
- ▶ Bisher haben wir Filter erster Ordnung (Gradienten) betrachtet
- ▶ Filter zweiter Ordnung -> Laplace Operator

$$\begin{aligned}\frac{\partial^2 f(x, y)}{\partial x^2} &\approx \frac{\partial(f(x+1, y) - f(x, y))}{\partial x} \\ &\approx f(x+1, y) - f(x, y) - (f(x, y) - f(x-1, y)) \\ &= f(x+1, y) - 2f(x, y) + f(x-1, y)\end{aligned}$$

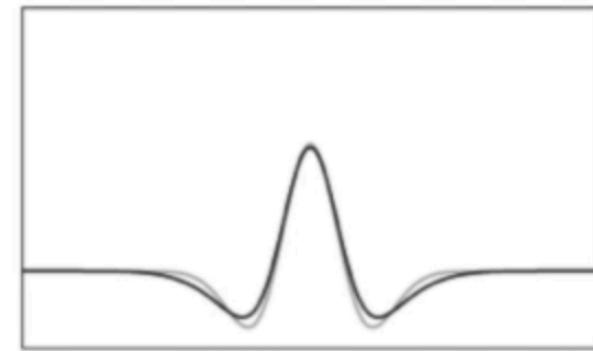
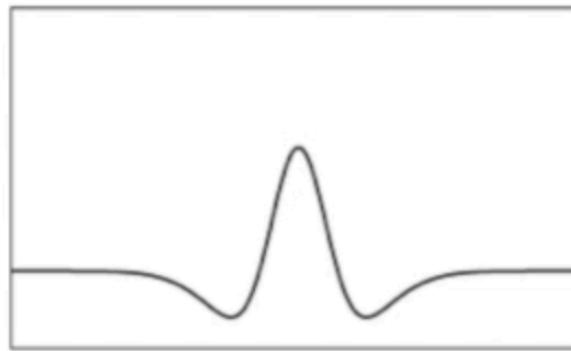
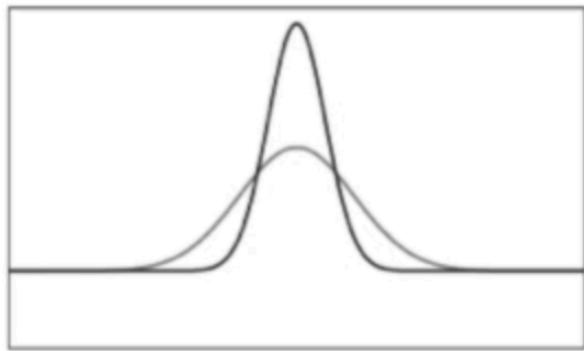
$$\mathbf{H}^L = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



# Difference of Gaussians (DoG) (1)

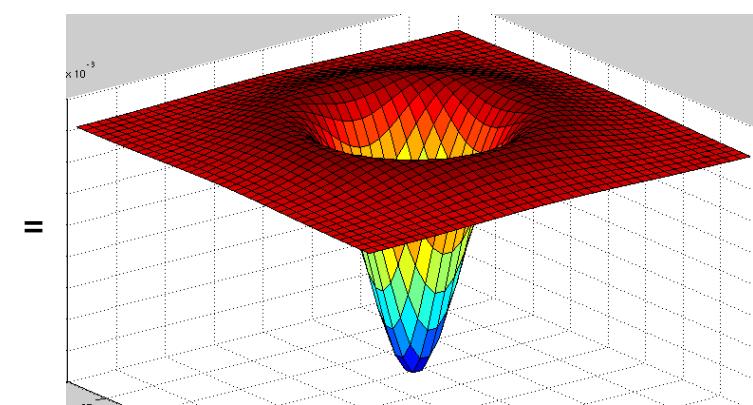
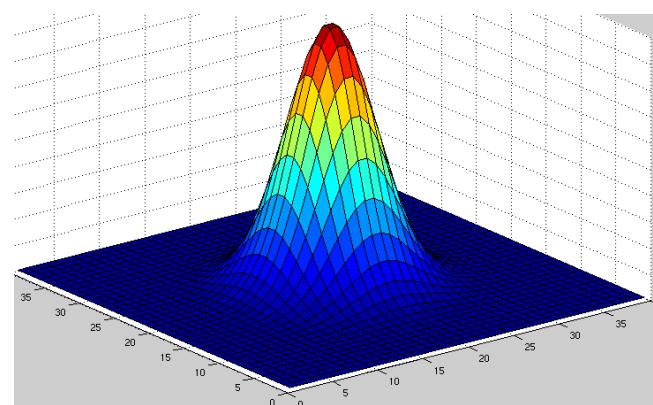
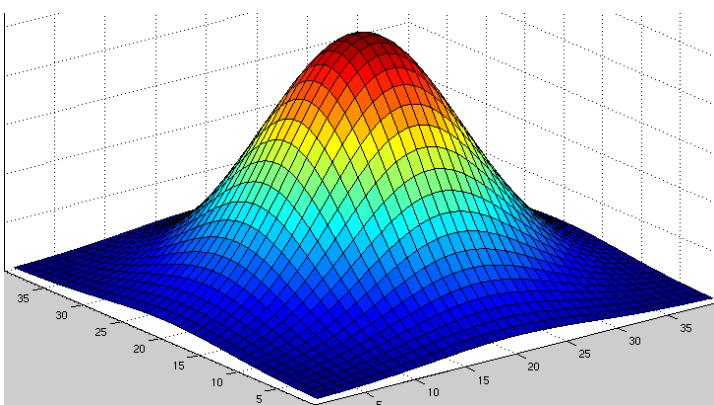


# Approximation des Laplace-Filters



**Abbildung 3.10:** ..; Links: Zwei Gaußverteilungen mit unterschiedlicher Varianz  $\sigma_1$  (schwarz) und  $\sigma_2 > \sigma_1$  (grau). Mitte: Subtraktion der Gaußverteilungen,  $\mathcal{N}(\mu, \sigma_1) - \mathcal{N}(\mu, \sigma_2)$ , ergibt einen Dog-Filter. Rechts: Vergleich des Dog-Filters (schwarz) mit einem Laplace-Filter (grau), der approximiert werden soll.

## Difference of Gaussians (2)



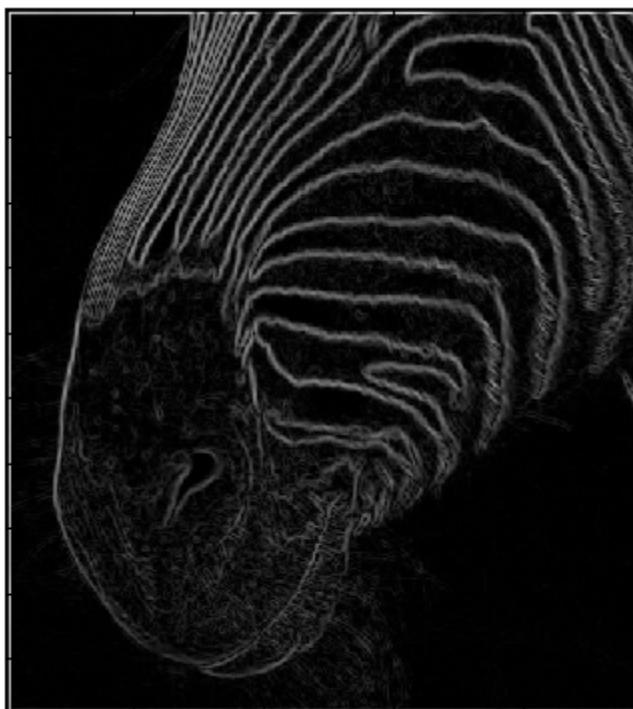
## Beispiel



Kantenerkennung

# Einfluss der Standardabweichung

- ▶ Höhere Standardabweichung resultiert in stärkerer Weichzeichnung
  - Eliminiert Kanten, die durch Bildrauschen erzeugt werden
  - Verbreitert Kanten
  - Entfernt Details

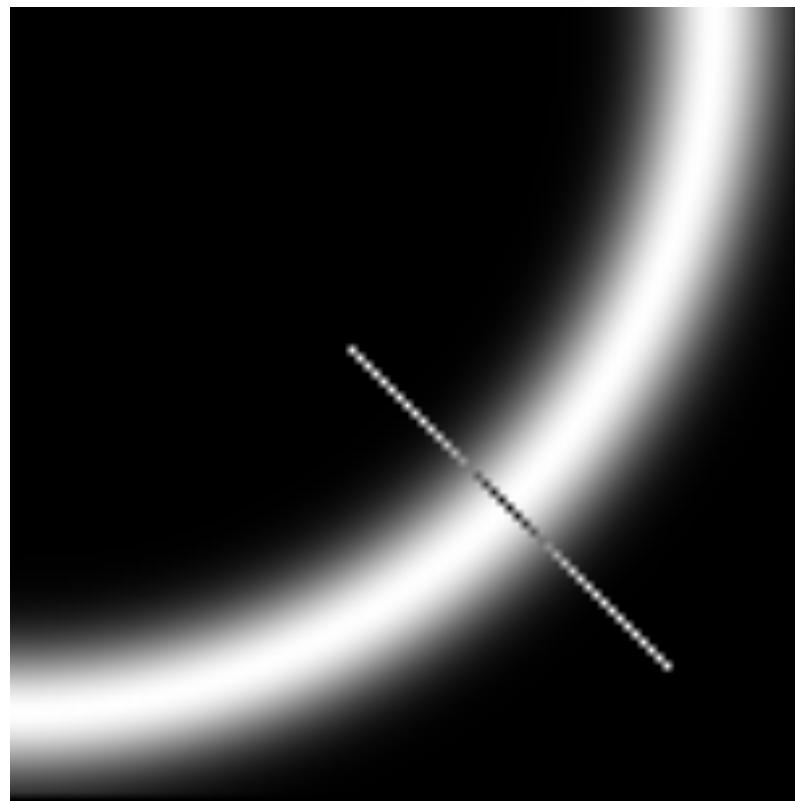


# Canny Edge Detection

- ▶ Nicht lokal, keine geschlossene Form
- ▶ Algorithmus:
  1. Gaußverrauschung zur Rauschunterdrückung
  2. Non-Maximum-Supression
    - Dünne Linien mit mehr als einem Pixel Breite werden in einem Pixel zusammengefasst
  3. Anwendung von Schwellwerten
    - Akzeptiere starke Kanten mit hohem Schwellwert
    - Akzeptiere schwache Kanten mit niedrigen Schwellwert wenn sie mit Kanten verbunden sind, die den hohen überstanden haben

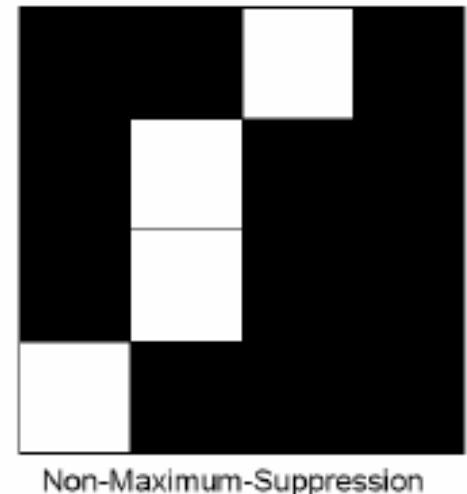
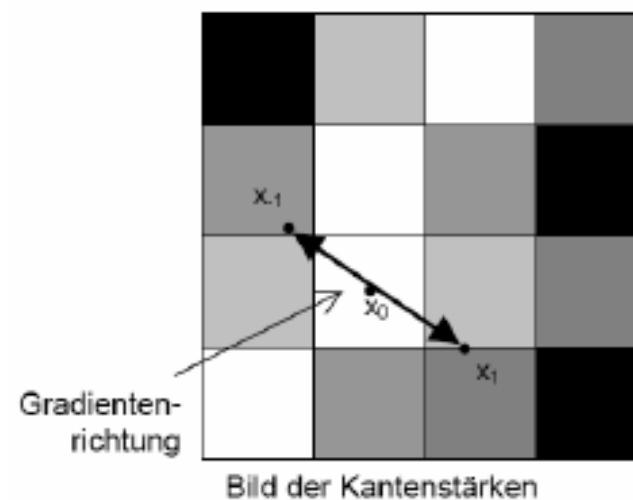
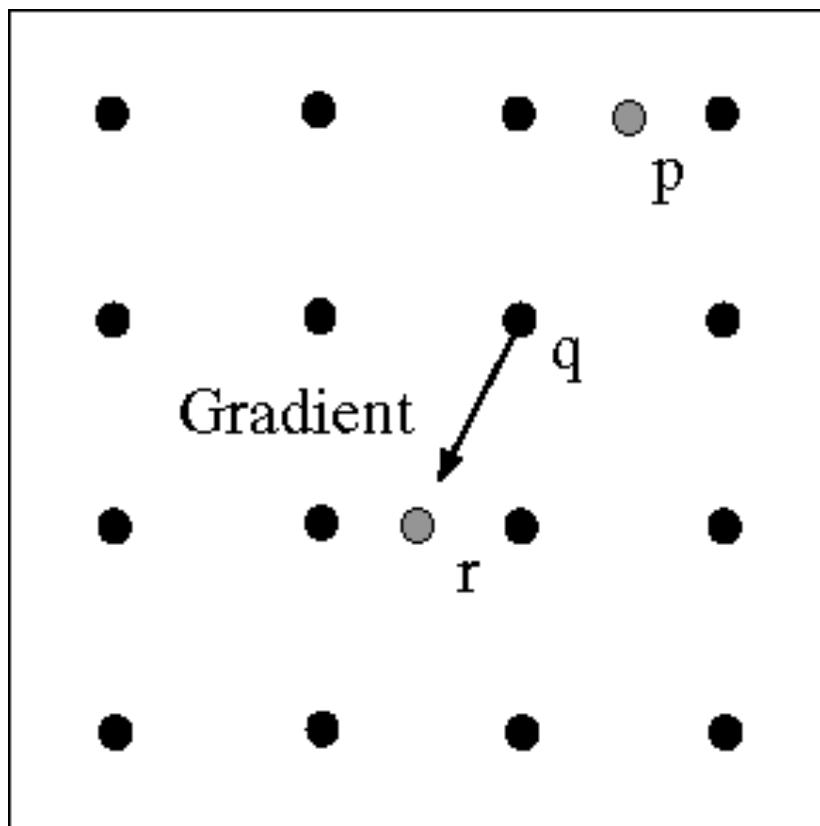
## Canny Edge Detection - Schritt 2

- ▶ Non-Maximum-Suppression:



## Canny Edge Detection - Schritt 2

- Der Wert von  $q$  muss größer sein als die interpolierten Werte von  $p$  und  $r$



## Canny Edge Detection - Schritt 2

- ▶ Linienausdünnung



Originalbild

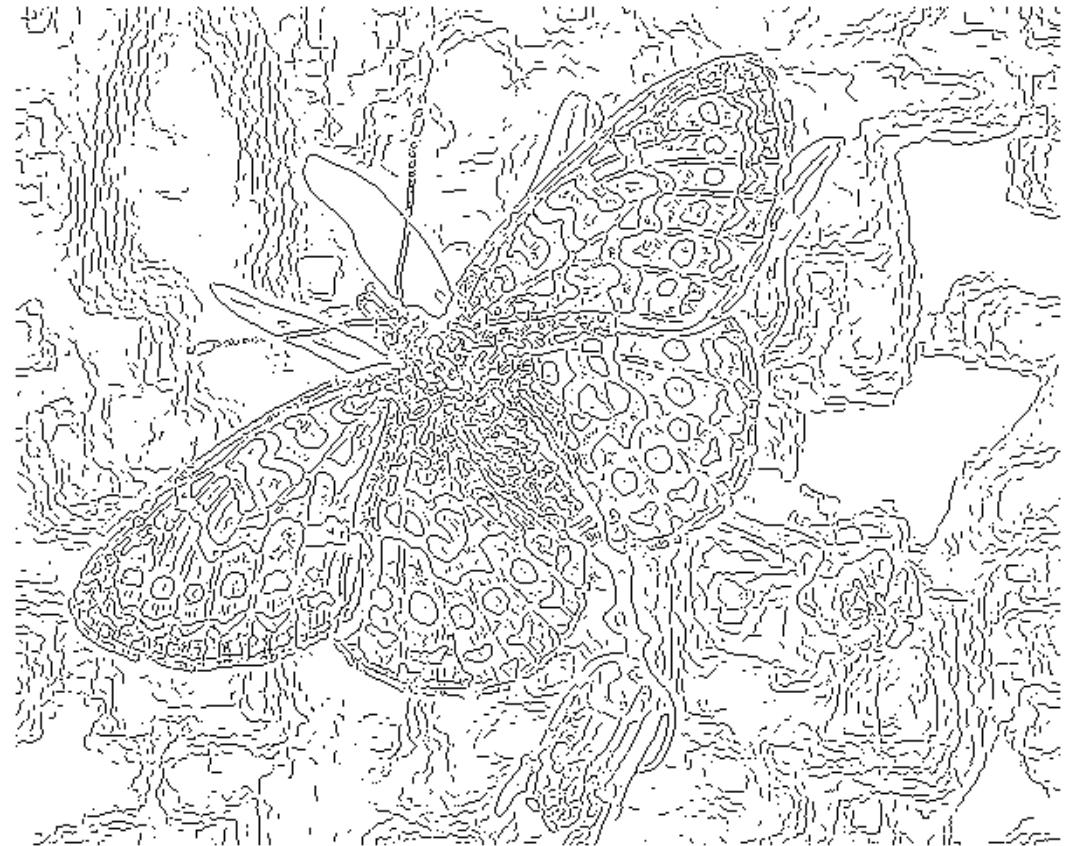


DoG-Filter



Ausgedünnte Linien

## Canny Edge Detection - Schwellwerte



$$\sigma = 1$$

## Canny Edge Detection - Schwellwerte



$$\sigma = 4$$

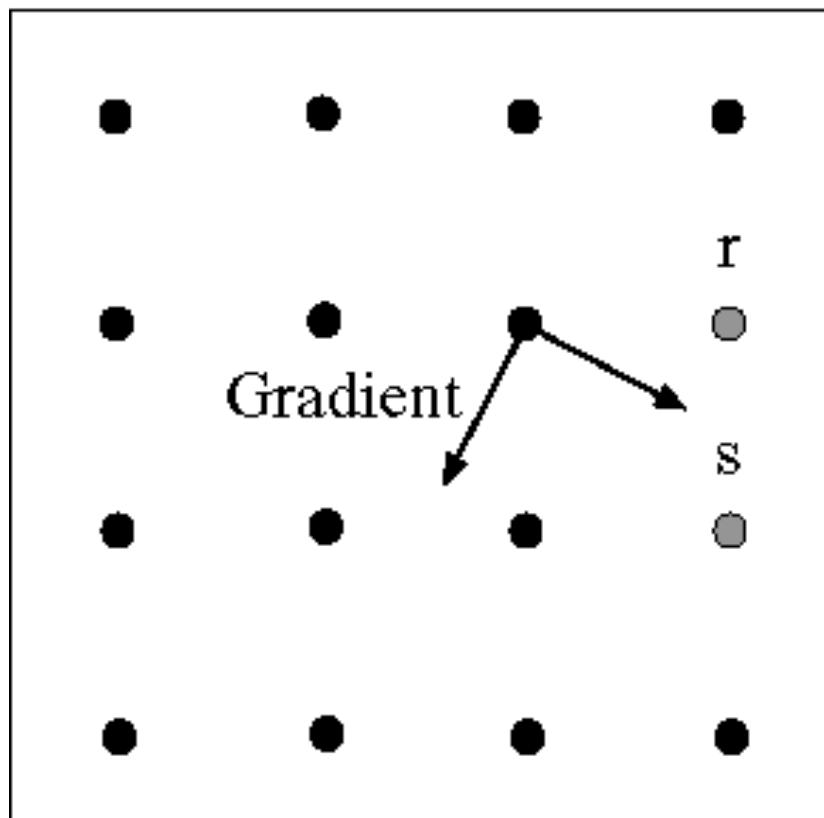
## Canny Edge Detection - Schwellwerte



$$\sigma = 10$$

## Canny Edge Detection - Schritt 3

- ▶ Verbinde starke und schwache Kanten



Annahme:  $p$  ist Kantelpunkt:

Betrachte die Normale des Gradienten. Mach Vorhersage über mögliche Verlängerungspunkte ( $r$  oder  $s$ )

# Hysterese

- ▶ Hysterese: Betrachte neben dem aktuellen Zustand auch die Historie vergangener Zustände um die Entwicklung vorherzusagen
- ▶ Verzögerung / Beschleunigung der Entwicklung erkennen
- ▶ Hier verwendet, um die Schwellenwerte zum Kantenfinden anzupassen
- ▶ Idee: Verwalte zwei Schwellwerte  $k_h$  und  $k_l$ 
  - Finde starke Kanten mittels  $k_h$  um einen Kantenzug zu starten
  - Finde schwache Kanten mittels  $k_l$ , die den Kantenzug fortführen
- ▶ Das typische Verhältnis ist ca.  $k_h/k_l = 2$

## Canny Edge - Beispiel



Starke Kanten

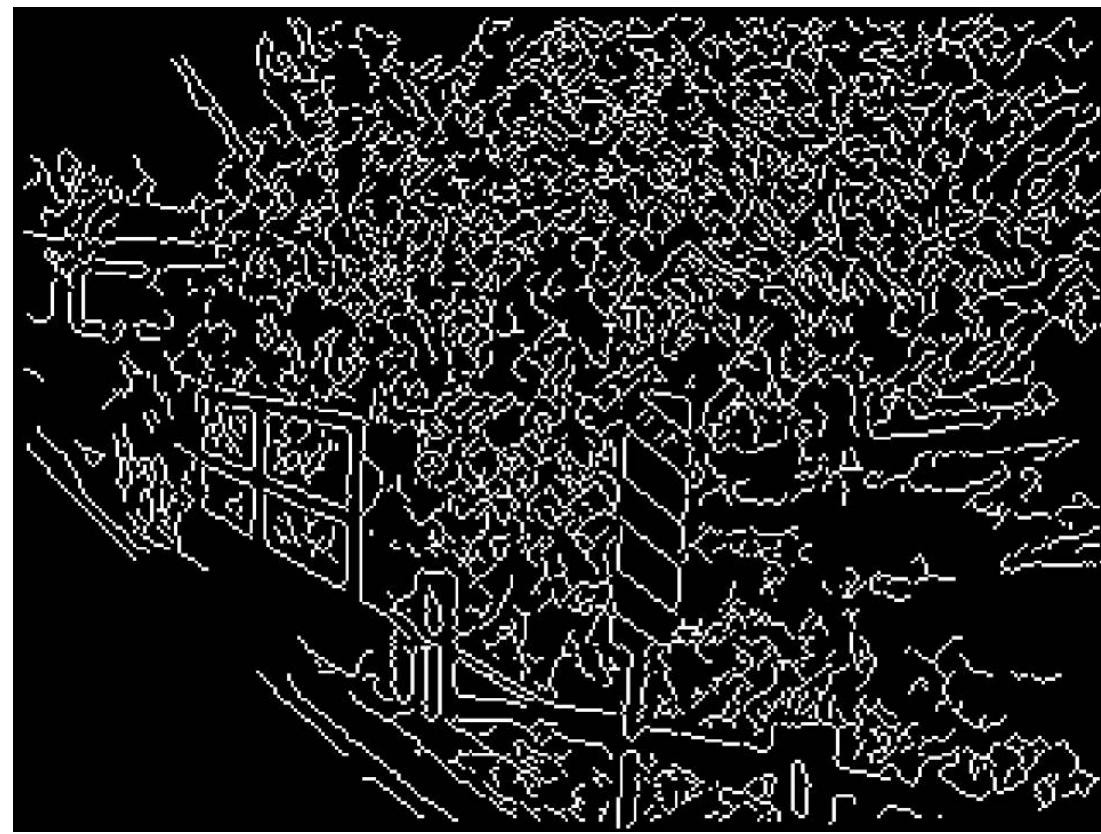


Schwache Kanten

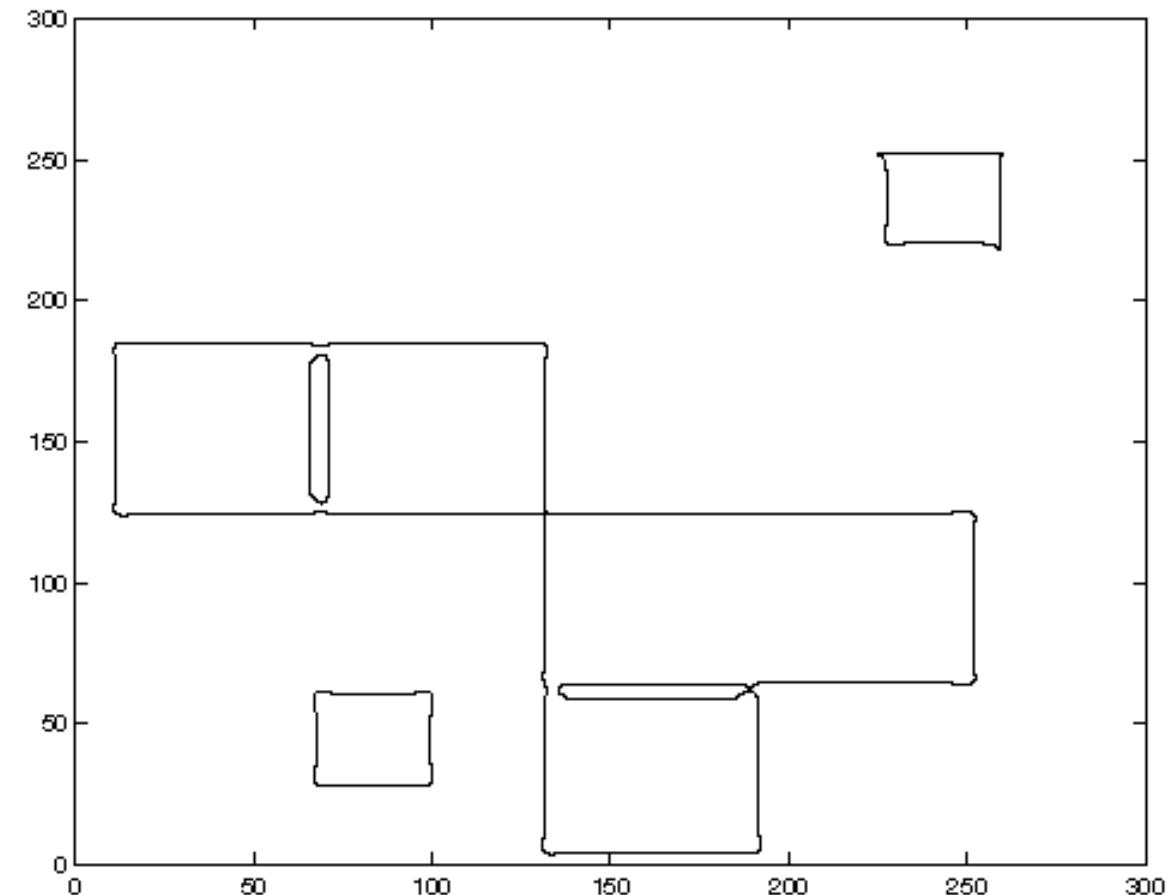


Canny Edge

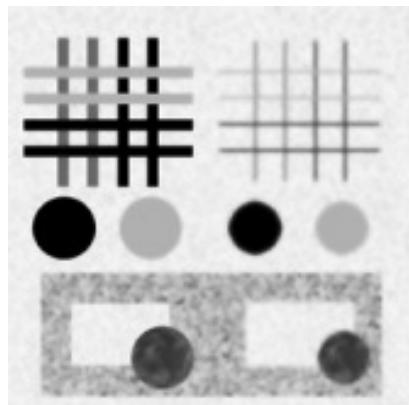
## Canny Edge - Beispiel



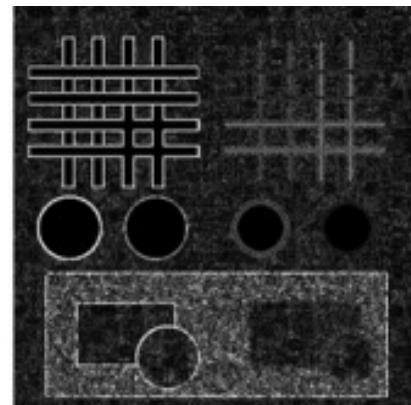
# Seiteneffekte an Ecken



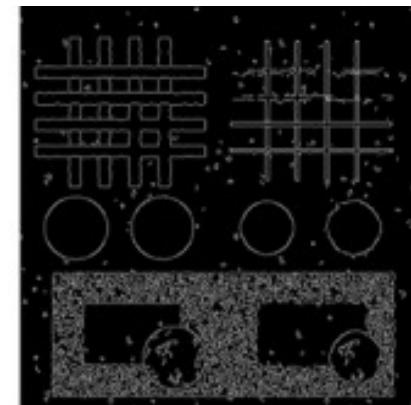
# Zusammenfassung Kantenfindung



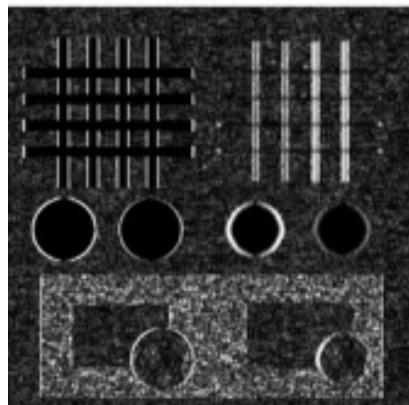
Original



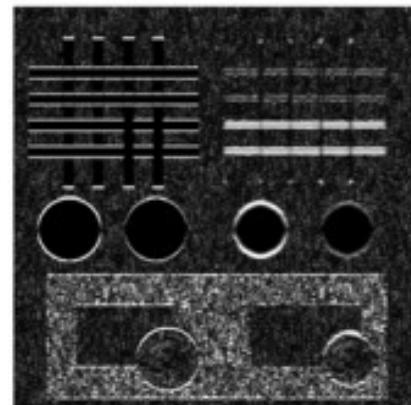
Laplacian



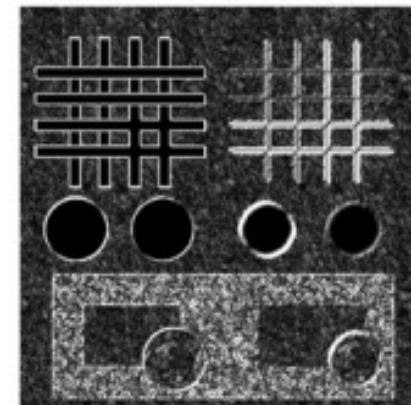
Canny



Sobel X



Sobel Y



Sobel X+Y

Uchida, Seiichi. "Image processing and recognition for biological images." *Development, growth & differentiation* 55.4 (2013): 523-549.

## Von Kanten zu Ecken

- ▶ Kanten haben einen hohen Grauwertgradienten in einer Richtung
- ▶ Kanten bestehen aus Linienzügen von vielen Pixeln
- ▶ Die bisher vorgestellten Filter versagen an Ecken
- ▶ Ecken zeichnen haben einen starken Gradienten in zwei Richtungen
- ▶ Ecken / Herausstehende Punkte sind genau lokalisiert
- ▶ Anforderungen an einen Operator zur Eckenerkennung:
  - Regionen mit homogener Textur werden als nicht interessant erkannt
  - Kantenpunkte sollen einen starken Gradienten in einer Richtung aufweisen
  - Punkte an Ecken und herausragende Punkte sollen signifikante Gradienten in beiden Richtungen aufweisen
- ▶ „Harris Operator“

# Harris Operator (1)

- ▶ Betrachte in einem Grauwertbild  $I$  eine Region der Größe  $(u, v)$
- ▶ Verschiebe die Region im Bild und betrachte die gewichtete Summe der Grauwertunterschiede:

$$S_I(x, y) = \sum_u \sum_v w(u, v) (I(u, v) - I(u - x, v - y))^2$$

- ▶ Leite die Harris-Matrix  $\mathbf{A}$  durch eine Taylorreihenapproximation von  $S_I$  ab:

$$S_I(x, y) \approx S_I(0, 0) + (x, y) \nabla S_I + \frac{1}{2}(x, y) \mathbf{A} \begin{pmatrix} x \\ y \end{pmatrix}$$

- ▶ Vereinfacht ergibt sich also

$$S_I(x, y) \approx \frac{1}{2}(x, y) \mathbf{A} \begin{pmatrix} x \\ y \end{pmatrix} = 0$$

## Harris Operator (2)

- ▶  $S_I$  ist eine Funktion der Grauwerte von  $I$
- ▶ Definition der Ableitungen  $I_x$  und  $I_y$

$$\mathbf{I}_x := \partial \mathbf{I} / \partial x$$

$$\mathbf{I}_y := \partial \mathbf{I} / \partial y$$

- ▶ Somit ergibt sich:

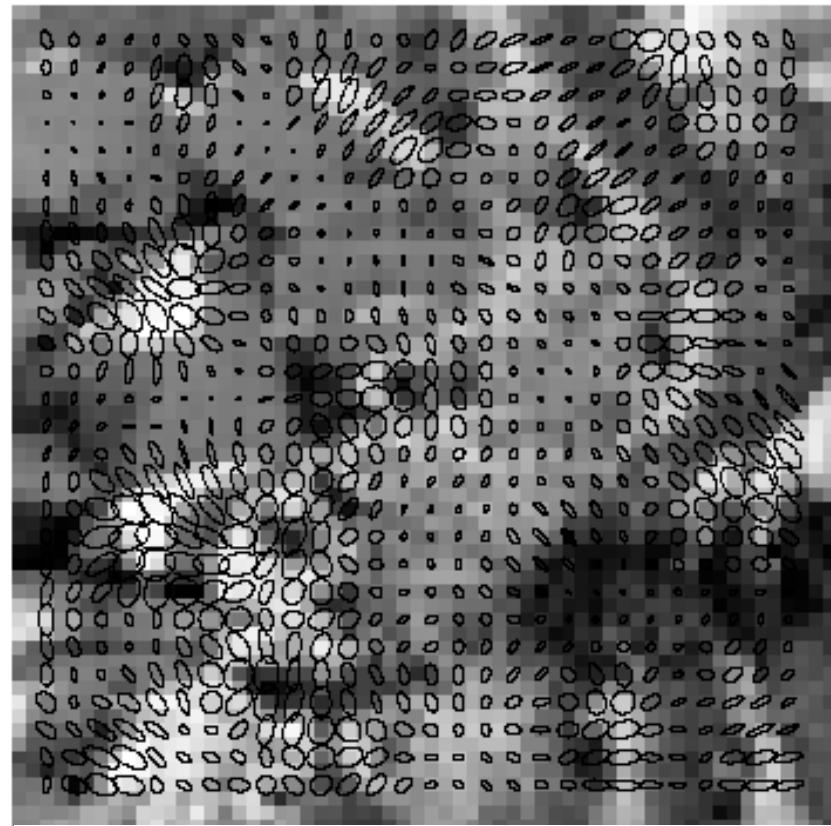
$$\mathbf{A} = \sum_u \sum_v w(u, v) \begin{pmatrix} \mathbf{I}_x^2 & \mathbf{I}_x \mathbf{I}_y \\ \mathbf{I}_x \mathbf{I}_y & \mathbf{I}_y^2 \end{pmatrix}$$

- ▶ Eine Ecke / Interessanter Punkt zeichnet sich durch eine Variation von  $S_I$  in beide Richtungen von  $(x, y)$  aus

## Harris Operator (3)

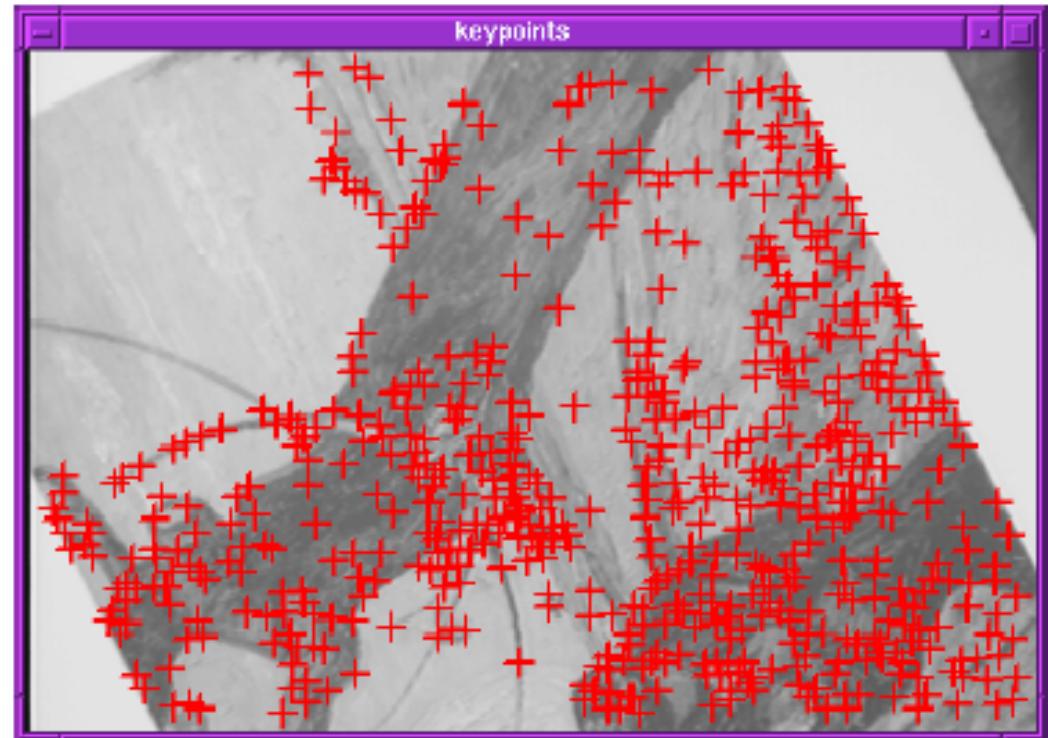
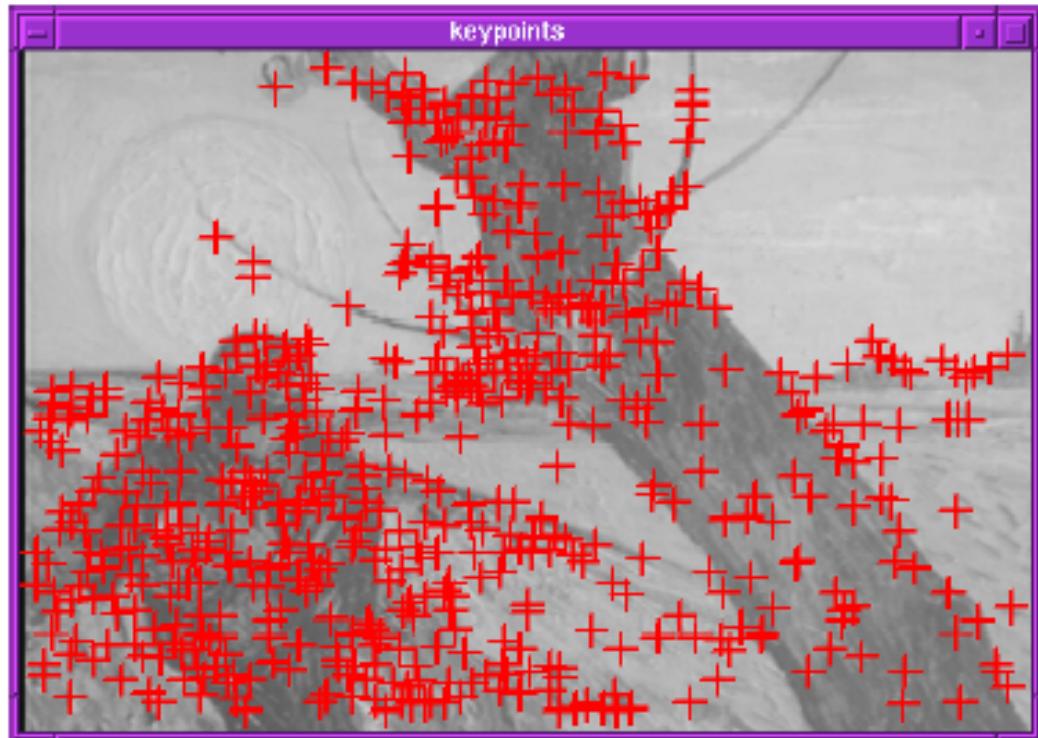
- ▶ Wie erkenne ich einen interessanten Punkt?
- ▶ Analyse der Eigenwerte  $\lambda_1$  und  $\lambda_2$  von  $A$
- ▶ Es gelten folgende Eigenschaften:
  1. Wenn  $\lambda_1 \approx 0$  und  $\lambda_2 \approx 0$  liegt kein interessanter Punkt vor.
  2. Eine Kante liegt vor, wenn  $\lambda_1 \approx 0$  und  $\lambda_2 = c_1$  und  $c_1 \gg 0$  ist.
  3. Eine Ecke liegt vor, wenn  $\lambda_1 = c_1$ ,  $\lambda_2 = c_2$  und  $c_1 \neq c_2$ , sowie  $c_1 \gg 0$  und  $c_2 \gg 0$ .
- ▶ Das Verfahren erfüllt also formal die eingangs gestellten Anforderungen
- ▶ A ist rotationssymmetrisch!

## Harris Corner Detection - Beispiel



Corners are detected where the product of the ellipse axis lengths reaches a local maximum.

## Harris Corner Detection - Beispiel



- ▶ Ursprünglich für Motion Tracking entwickelt
- ▶ Reduziert die Rechenzeit erheblich verglichen mit Pixel-Pixel-Tracking
- ▶ Translations- und Rotationsinvariant (nicht skalierungsinvariant)

# Harris Corners zu Kamerakalibrierung

