

# Elementare Bildverarbeitungsoperationen

## - Kantenerkennung -

1 Einführung

2 Gradientenverfahren

3 Laplace-Verfahren

4 Canny-Verfahren

5 Literatur



## 1 Einführung

Kantenerkennung basiert auf *linearer Filterung*.

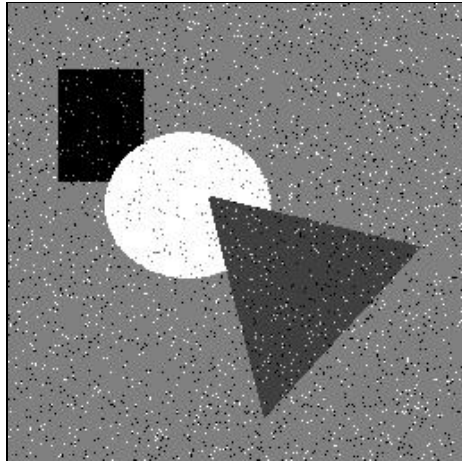
### Definition

Ein Pixel nennt man *Kantenpixel*, wenn sich seine Nachbarschaft durch hohe Graustufenvariationen auszeichnet.

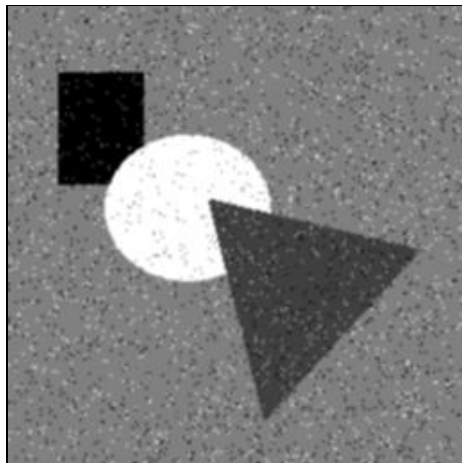
Eine Menge von Kantenpixeln nennt man *Kante*, wenn die Kantenpixel paarweise über einen Pfad in der Menge verbunden sind.

Den **Prozess der Kantenerkennung** kann man in drei Schritte untergliedern:

1. Reduktion des Bildrauschens
2. Kanten hervorhebung
3. Kantenlokalisierung

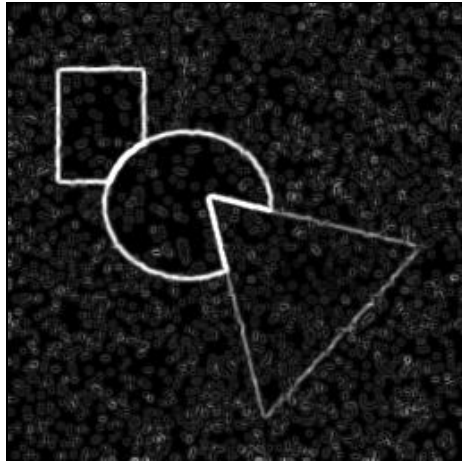


Wir betrachten ein verrauschtes Bild.



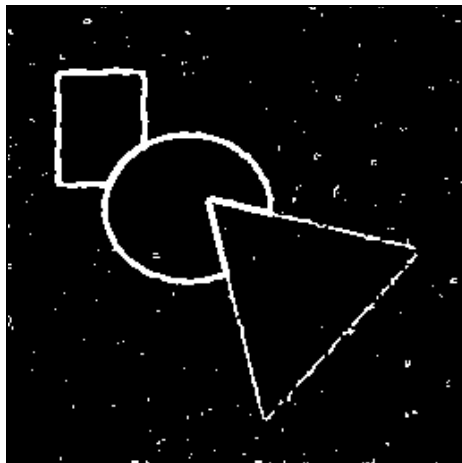
## Reduktion des Bildrauschens

Das Bildrauschen sollte so stark wie möglich unterdrückt werden, ohne die eigentlichen Kanten stark zu glätten.



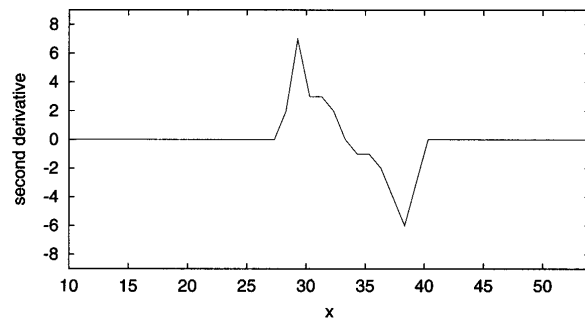
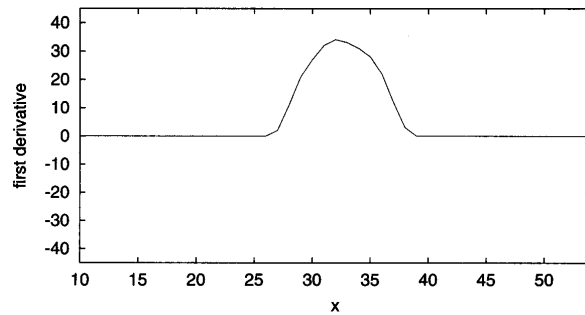
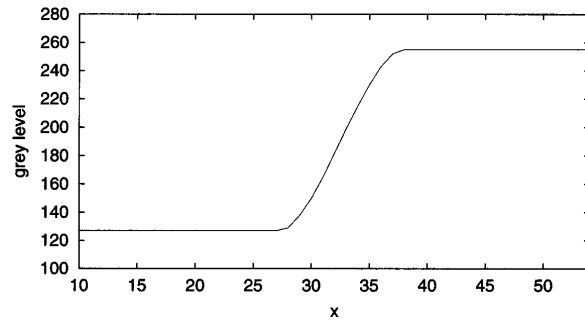
## Kantenhervorhebung

In diesem Schritt wird ein (linearer) Filter ( $\rightarrow$  *High Pass Filter*) auf das Bild angewendet, der Kanten verstärkt und andere Bildstrukturen abschwächt.



## Kantenlokalisierung

Abschließend entscheidet man, welche hervorgehobenen Pixel echte Kantenpixel sind und erstellt ein *Binärbild*, in dem Kantenpixel auf 1 (weiß) gesetzt sind.



**Was zeichnet eine Kante im zweidimensionalen Bild aus?**

Betrachtung der 1. und 2. partiellen Ableitungen

**2 Gradientenverfahren**

Für  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ ,  $f \in \mathcal{C}^1$  gilt  $\forall (x, y) \in \mathbb{R}^2$ :

$$\frac{\partial f}{\partial x}(x, y) = \lim_{h \rightarrow 0} \frac{f(x+h, y) - f(x-h, y)}{2h} \stackrel{h=1}{\approx} \frac{1}{2} [f(x+1, y) - f(x-1, y)]$$

Entsprechendes gilt für  $\frac{\partial f}{\partial y}(x, y)$  und es ergibt sich für den Gradienten:

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

Um das Bildrauschen zu reduzieren, berechnen wir den Gradienten über einer  $3 \times 3$ -Nachbarschaft. Dies führt zu folgender Darstellung mit Hilfe der Faltung:

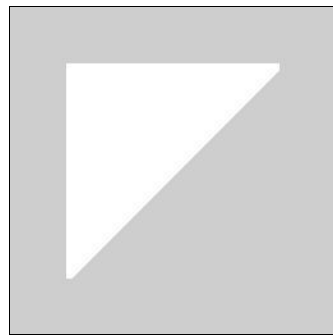
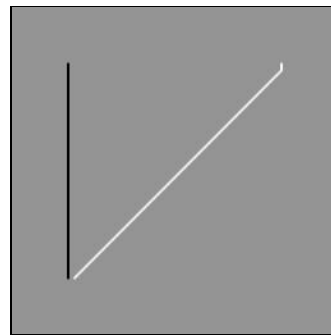
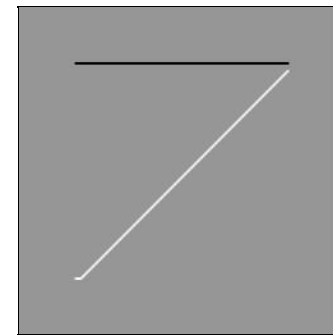
$$\nabla f(x, y) \approx \begin{bmatrix} (h_x * f)(x, y) \\ (h_y * f)(x, y) \end{bmatrix}.$$

Dabei sind  $h_x$  und  $h_y$  z.B. gegeben durch die sogenannten *Prewitt-Filter*

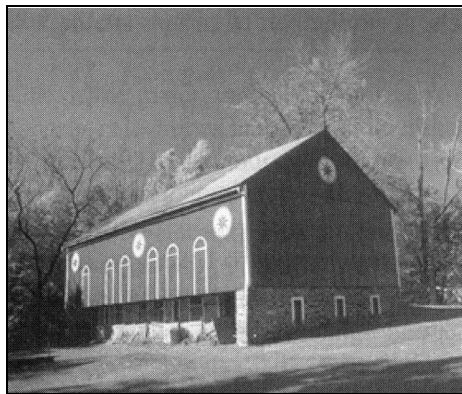
$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad h_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Durch zweifache Gewichtung der Pixel auf den Achsen ergeben sich die sogenannten *Sobel-Filter*.

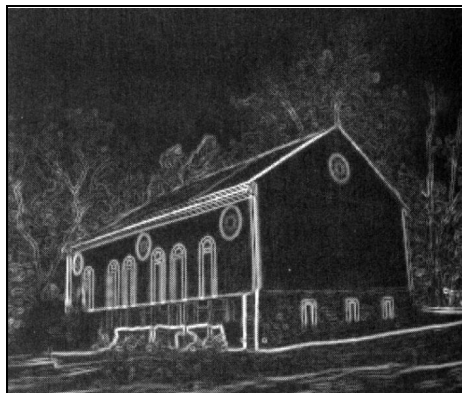
### Beispiel

 $f$  $h_x * f$  $h_y * f$

Der Gradientenbetrag  $|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \approx \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$  ist ein Maß für die Graustufenvariation und hebt somit Kanten hervor.



Originalbild



### Reduktion des Bildrauschens und Kantenhervorhebung

z.B. durch Anwendung der Prewitt-Filter und anschließende Bildung des Gradientenbetrags





### Kantenlokalisierung durch einfaches Thresholding

Über einfaches Thresholding (threshold (engl.): Schwellenwert) entscheidet man, ob das Pixel Kantenpixel ist oder nicht.

Betrachte dazu

$$g(x, y) = \begin{cases} 0, & |\nabla f(x, y)| < T, \\ 1, & |\nabla f(x, y)| \geq T, \end{cases}$$

wobei  $T$  ein fest vorgegebener Schwellenwert sei.

### Bemerkungen

- Der Erfolg der Kantenerkennung hängt entscheidend von der Wahl des Schwellenwertes  $T$  ab:

 $T = 50$  $T = 150$ 

- Kanten zwischen Bildelementen sind nicht immer scharf. Unscharfe Kanten führen zu kleineren Gradientenbeträgen und können so herausgefiltert werden.
- Rauschen kann sehr hohe Gradientenbeträge hervorrufen und kann so fälschlicherweise als Kante erkannt werden.

### 3 Laplace-Verfahren

Der Laplace-Operator  $\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y}$  ist ein Maß für die Rate der Graustufenvariation und hebt somit Kanten hervor.

#### 1. Kanten hervorhebung durch Bestimmung des Laplace-Bildes

$\nabla^2$  kann über einer  $3 \times 3$ -Nachbarschaft angenähert werden durch

$$\nabla^2 f \approx h * f, \text{ wobei } h = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}.$$

Berechne also für jedes Pixel  $\nabla^2 f(x, y)$  und führe die Informationen in einem Laplace-Bild zusammen.

#### 2. Kantenlokalisierung

erfolgt durch anschließende Suche nach Nulldurchgängen (*zero crossings*).

#### Bemerkungen

- Zur Reduktion des Bildrauschens wird der Laplace-Filter oftmals mit einem Gauss-Filter kombiniert (*Laplacian of Gaussian - LoG*):

Mit der Gauss-Funktion  $h : \mathbb{R}^2 \rightarrow \mathbb{R}$ ,  $h(x, y) = \exp(-\frac{1}{2} \frac{x^2 + y^2}{\sigma^2})$  ergibt sich der LoG-Filter

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

Dabei bestimmt  $\sigma$  die Breite des Filters ( $> 6\sigma$ ) sowie den Grad der Glättung (und damit die Anzahl erkannter Kanten).

Aufgrund der Form des Graphen von  $\nabla^2 h$  nennt man den LoG-Filter auch *Mexican-Hat-Filter*.

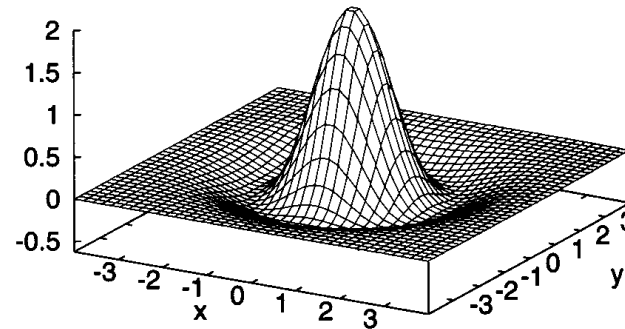
- Das Laplace-Verfahren liefert scharfe Kanten.



LoG-Filterung mit  $\sigma = 3.0$



LoG-Filterung mit  $\sigma = 5.0$



Graph von  $-\nabla^2 h$

### 4 Canny-Verfahren



Originalbild



**Reduktion des Bildrauschens mit Gauss-Filter**

**Kantenerhebung über den Gradientenbetrag**

Diese beiden Schritte können auch als Einheit durchgeführt werden.



### Kantenlokalisierung (1. Teilschritt)

#### Nicht-Maximum-Unterdrückung

Bereiche um lokale Maxima im Gradientenbetrag (d.h. um Kanten) werden reduziert auf scharfe, ein Pixel breite Kanten.



### Kantenlokalisierung (2. Teilschritt)

#### Hysterese

Hysterese bietet einen Lösungsansatz der bereits genannten Probleme beim einfachen Thresholding.

**Hysterese** arbeitet mit zwei Schwellenwerten  $T_{low}$  und  $T_{high}$ :

- Liegt ein lokales Maximum über  $T_{high}$ , wird es sofort als Kantenpixel erkannt.
- Liegt ein lokales Maximum unter  $T_{low}$ , wird es sofort als Kantenpixel abgelehnt.
- Liegt ein lokales Maximum zwischen  $T_{low}$  und  $T_{high}$ , so wird es akzeptiert, wenn es von ihm aus einen Pfad aus Pixeln mit  $f(x_i, y_i) > T_{low}$  zu einem Pixel mit  $f(x_0, y_0) > T_{high}$  gibt.



### Bemerkungen

- Hysterese verringert das Auftreten von unterbrochenen Kanten dramatisch:



$T = 100$



$T = 50$



$T_{low} = 50,$

$T_{high} = 100$

- Das Canny-Verfahren bietet einen sehr guten Kompromiss zwischen Reduktion des Bildrauschens und Kantenlokalisierung.
- Das Canny-Verfahren liefert scharfe Kanten.

### 5 Literatur

- J. Canny, *A Computational Approach to Edge Detection*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986
- N. Efford, *Digital Image Processing: A Practical Introduction using Java*, Addison-Wesley, 2000
- B. Jähne, *Digital Image Processing: Concepts, Algorithms and Scientific Applications*, Springer-Verlag, 2002
- D. A. Lyon, *Image Processing in Java*, Prentice Hall, 1999
- D. Marr und E. Hildreth, *Theory of Edge Detection*, 1980
- P. Soille, *Morphological Image Analysis: Principles and Applications*, Springer-Verlag, 1998
- Java-Applet auf <http://www-mm.informatik.uni-mannheim.de/veranstaltungen/animation/multimedia/segmentation/Applet/Segmentation.html>