

Digitale Bildverarbeitung

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

Dr. rer. nat. Johannes Riesterer

Kanten

Kanten sind durch schnelle Änderungen des Farbwertes gekennzeichnet. Sie sind damit Extremstellen der ersten Ableitung.

Intensität und Gradient entlang eines Bildschnittes

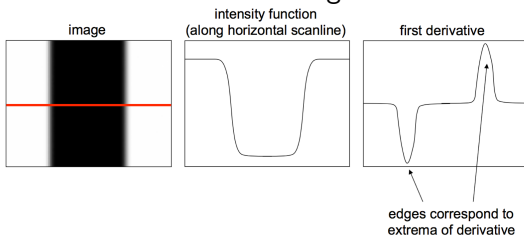


Figure: Quelle: ai.stanford.edu

Gradientenbasierte Kantenerkennung

Bei der Detektion von Kanten mit Hilfe des Gradienten ist Rauschen ein Problem, da sich hier ebenfalls der Farbwert schnell ändert.

Rauschen

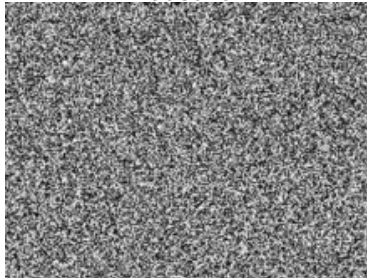


Figure: Quelle: Wikipedia

Gradientenbasierte Kantenerkennung

Idee: Wende einen Filter an, der das Rauschen reduziert und bilde dann den Gradienten. Bilde also den Gradienten

$$\frac{\partial(u * f)(x)}{\partial x}$$

wobei f ein Faltungskern ist.

Ableitung von Faltungen

Es gilt

$$\frac{\partial(u * f)(x)}{\partial x} = (u * f')(x)$$

Gradientenbasierte Kantenerkennung

Welcher Filter ist gut geeignet?

Kantenerkennung nach Canny

Es gibt Kanten auf unterschiedlichen Skalen ("grobe Kanten" und "feine Kanten"). Wähle daher einen parameterabhängigen Faltungskern f_σ . Zu einem Originalbild u_0 bekommen wir eine ganze Klasse von Bildern

$$u(x, \sigma) = u_0 * f_\sigma(x) .$$

Kantenerkennung nach Canny

Die Stellen der Kanten soll sich bei wachsendem σ nicht verändern und ebenso sollen auch keine Kanten hinzukommen. Deswegen soll in einem Kantenpunkt x_0 von u_0 gelten:

$$\frac{\partial^2}{\partial x^2} > 0 \Rightarrow \frac{\partial}{\partial \sigma} u(x_0, \sigma) > 0$$

$$\frac{\partial^2}{\partial x^2} = 0 \Rightarrow \frac{\partial}{\partial \sigma} u(x_0, \sigma) = 0$$

$$\frac{\partial^2}{\partial x^2} < 0 \Rightarrow \frac{\partial}{\partial \sigma} u(x_0, \sigma) < 0$$

Kantenerkennung nach Canny

Für einen allgemeinen Punkt soll daher gelten:

$$\begin{aligned}\frac{\partial^2}{\partial x^2} u(x, \sigma) &= \frac{\partial}{\partial \sigma} u(x, \sigma) \\ u(x, 0) &= u_0(x)\end{aligned}$$

Kantenerkennung nach Canny

Diese partielle Differentialgleichung hat die eindeutige Lösung

$$u(x, \sigma) = (u_0 * G^{\sqrt{2\sigma}})(x)$$

wobei $G^{\sqrt{2\sigma}}$ der Gaußfilter ist.

Kantenerkennung nach Canny

Die Kantenerkennung nach Canny faltet ein gegebenes Bild u zuerst mit einem Gaußkernel G^σ . Danach wird der Betrag der Ableitung und seine Richtung berechnet:

$$\begin{aligned} p(x) &= \|\nabla(u * G^\sigma)(x)\| \\ &= \sqrt{\left(\frac{\partial}{\partial x_1}(u * G^\sigma)(x)\right)^2 + \left(\frac{\partial}{\partial x_2}(u * G^\sigma)(x)\right)^2} \\ \theta(x) &= \angle \nabla(u * G^\sigma)(x) = \arctan\left(\frac{\frac{\partial}{\partial x_2}(u * G^\sigma)(x)}{\frac{\partial}{\partial x_1}(u * G^\sigma)(x)}\right) \end{aligned}$$

Kantenerkennung nach Canny

Als Kanten werden lokale Maxima von $p(x)$ in Richtung $(\sin \theta(x), \cos \theta(x))$

Kanten als lokale Maxima in Kantenrichtung

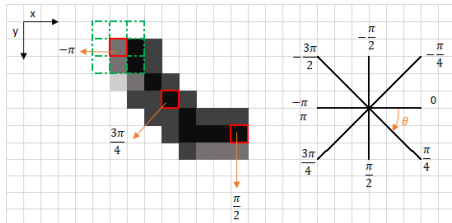


Figure: Quelle: towardsdatascience.com

Kantenschärfen mit Laplace

Durch die Operation $u - \tau \Delta u$ werden die Kanten hervorgehoben.

Kanten als lokale Maxima in Kantenrichtung

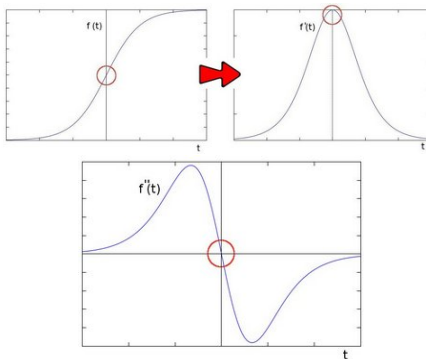


Figure: Quelle:OpenCV

Kantenschärfung

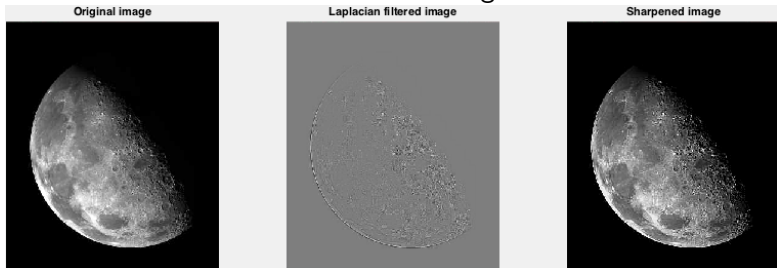


Figure: Quelle: Stackoverflow

Segmentierung

Die Erzeugung von inhaltlich zusammenhängenden Regionen durch Zusammenfassung benachbarter Pixel oder Voxel entsprechend einem bestimmten Homogenitätskriterium bezeichnet man als Segmentierung.

Segmentierung

Szene → Bildaufnahme → Bildvorverarbeitung → Segmentierung
→ Merkmalsextraktion → Klassifizierung → Aussage

Segmentierung

Schwelwert

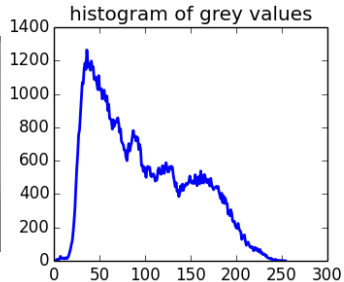


Figure: Quelle: scikit-image.org

Segmentierung über Intensität-Schwelwert

Schwelwert

coins > 100

coins > 150

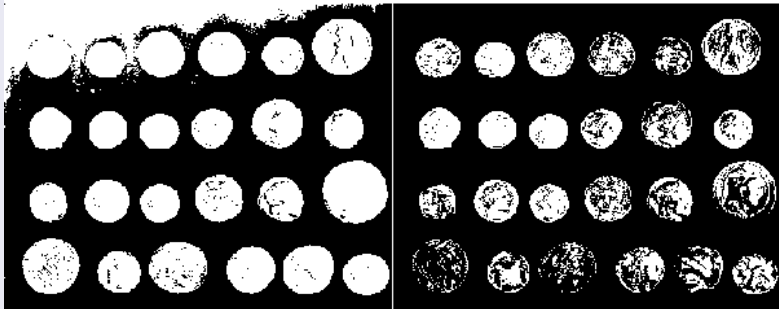


Figure: Quelle: scikit-image.org

Kantenbasierte Segmentierung

1. Kantenerkennung nach Canny

Canny
Canny detector

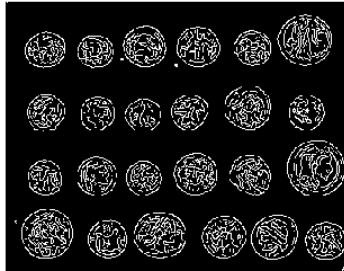


Figure: Quelle: scikit-image.org

Kantenbasierte Segmentierung

2. Löcher Füllen mit morphologischem Filter

Canny + morphologischem Filter
Filling the holes

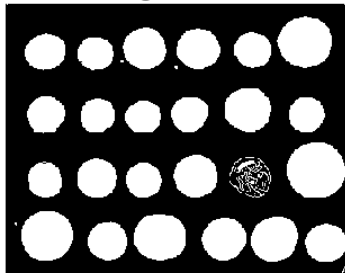


Figure: Quelle: scikit-image.org

Kantenbasierte Segmentierung

3. Kleine Objekte entfernen mit morphologischem Filter

Canny + morphologischem Filter + morphologischem Filter
Removing small objects

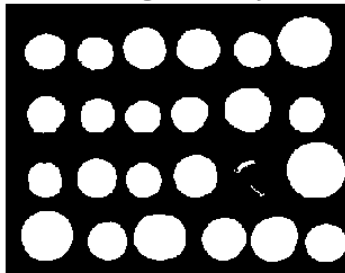


Figure: Quelle: scikit-image.org

Segmentierung über K-Means

Ziel von "k"-Means ist es, den Datensatz so in "k" Partitionen zu teilen, dass die Summe der quadrierten Abweichungen von den Cluster-Schwerpunkten minimal ist. Mathematisch entspricht dies der Optimierung der Funktion

$$J = \sum_{i=1}^k \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

mit den Datenpunkten \mathbf{x}_j und den Schwerpunkten $\boldsymbol{\mu}_i$ der Cluster S_i .

Segmentierung über K-Means

- Initialisierung: Wähle k zufällige Mittelwerte ("Means"): $\mathbf{m}_1^{(1)}, \dots, \mathbf{m}_k^{(1)}$ aus dem Datensatz.
- Zuordnung: Jedes Datenobjekt wird demjenigen Cluster zugeordnet, bei dem die Cluster-Varianz am wenigsten erhöht wird.

$$S_i^{(t)} = \left\{ \mathbf{x}_j : \|\mathbf{x}_j - \mathbf{m}_i^{(t)}\|^2 \leq \|\mathbf{x}_j - \mathbf{m}_{i^*}^{(t)}\|^2 \text{ für alle } i^* = 1, \dots, k \right\}$$

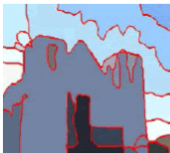
- Aktualisieren: Berechne die Mittelpunkte der Cluster neu

$$\mathbf{m}_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j$$

Segmentierung des Farbraumes über K-Means



(a)



(b)



(c)

Figure 1: (a) is the original image; (b) and (c) are the segmentation results.

Figure: Quelle: towardsdatascience.com

Segmentierung über K-Means

Ist die Anzahl der Clusterzentren unbekannt, kann man den Algorithmus mit verschiedenen Anzahlen von Zentren k ausführen und die Fehler miteinander vergleichen. Es gibt meistens einen Punkt, ab dem sich der Fehler nicht mehr signifikant ändert. Diesen kann man zum Beispiel wählen. (Elbowmethod).

Elbow Method

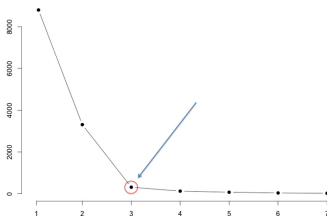


Figure: Quelle: mubaris.com