Computer Vision

Kanten, Schwellen & Regionen - Segmentierung & Kantendetektion

Alexandra Posekany

December 2025

Einführung Segmentierung: Vom Pixel zum Objekt

Arten der Segmentierung

Bildsegmentierung ist die **Identifikation** und **Isolation** eines Bildes in unterschiedliche Teilbereiche (Regionen). Diese entsprechen **strukturellen Einheiten** (Segmente).

Pixelbasierte Segmentierung - Thresholding

Input: Graustufenbild

Output: Binäres Bild (Schwarz-Weiß-Bild)

Schwellenwert-Methoden erstellen binäre Bilder und klassifizieren Pixel basierend darauf, ob ihre Intensität über oder unter einem bestimmten Grenzwert liegt. Histogramme, die die Häufigkeit bestimmter Pixelwerte in einem Bild darstellen, werden häufig zum Definieren von Schwellenwerten verwendet.

Pixelbasierte Segmentierung - Thresholding

Filterung anhand der Helligkeit, indem ein Schwellwert (Threshold) überschritten oder unterschritten wird

Formel: $g(x, y) = \begin{cases} 255 & \text{wenn } f(x, y) > T \\ 0 & \text{sonst} \end{cases}$

Anwendung: Trennung von Vordergrund (Objekt) und Hintergrund, wenn der Kontrast hoch ist.

Thresholding mithilfe des Histogramms

Histogramme können beispielsweise die Werte von Hintergrundpixeln ableiten und so zur Isolierung von Objektpixeln beitragen.

Wann ist Thresholding sinnvoll?

- wenn die zu unterscheidenden Strukturen hinreichend weit von einander entfernt sind (Effektgröße)
- wenn das Hintergrundrauschen nicht zu stark ist (Variation, Noise)

Thresholding 2.0 – Wenn Global versagt

Das Problem: Globales Thresholding (z.B. > 127) versagt bei ungleichmäßiger Beleuchtung (Schatten auf einem Blatt Papier).

Adaptive Thresholding

Der Schwellenwert wird für jeden Pixel individuell berechnet, basierend auf seinen Nachbarn (z.B. Mittelwert der 11x11 Umgebung).

Effekt: Funktioniert wie eine lokale Kontrastverstärkung.

Otsu's Methode

Automatische Suche nach dem perfekten globalen Schwellenwert.

Prinzip: Suche im Histogramm das "Tal" zwischen zwei "Bergen" (Vordergrund/Hintergrund). Minimiert die Varianz innerhalb der Klassen.

Adaptives Thresholding

Otsu's Methode

Kantendetektion

Die kantenbasierte Segmentierung ist ein fundamentales Verfahren in der Bildverarbeitung, das darauf abzielt, Objekte in einem Bild zu isolieren, indem ihre Grenzen (Kanten) identifiziert werden. Kantenerkennungsmethoden identifizieren die Grenzen von Objekten oder Klassen durch die Erkennung von Helligkeits- oder Kontrastunterbrechungen.

Ablauf des Canny-Algorithmus

1. **Bildglättung** (Rauschunterdrückung, Noise Reduction): Gauß-Filter (Weichzeichnen), um Rauschen zu entfernen.
2. Berechnung der **Gradienten** (Kantenfilterung, Gradient Calculation): Finden der Intensitätsgradienten (ähnlich Sobel).
3. **Non-Maximum Suppression** (Kantenverdünnung): Nur lokale Maxima bleiben stehen. Macht die Kante 1 Pixel breit.
4. **Schwellenwertverfahren** (Hysterese, Hysteresis Thresholding): Zwei Schwellenwerte (Min/Max).

- Starke Kanten (>Max) werden sofort akzeptiert.
- Schwache Kanten (zwischen Min und Max) werden nur akzeptiert, wenn sie an einer starken Kante hängen.

5. **Kantenverbindung** (Edge Linking)

Canny-Algorithmus

Kantenbasierte Segmentierung

2. Schritt: **Kantenfindung** (Sobel filter, Laplacian, Canny filter)

Gradienten

Die Mathematik der Kante und der Bild-Gradient
WAS IST EINE KANTE MATHEMATISCH?
Intuition: Helligkeitswechsel.
Kante
Signal f(x)
(Helligkeit)
x
1. Ableitung f'(x)
(Gradient)
Hohe Änderung = Hohe Ableitung = Kante.
Maximum an der Kante.
x
2. Ableitung f''(x)
(Laplace)
Nulldurchgang (Zero-Crossing) an der Kante.
x
DER BILD-GRADIENT
Gy (Änderung in Y)
Gradient
∇f = [Gx, Gy]T
Gx (Änderung in X)
Bedeutung: Wir müssen Änderungen in X-Richtung und Y-Richtung getrennt berechnen.
Magnitude (Stärke):
$|G| = \sqrt{G_x^2 + G_y^2}$
$\approx |G_x| + |G_y|$ (für Speed)
Richtung (Winkel):
$\theta = \arctan(G_y/G_x)$
Die Kantenrichtung steht senkrecht zum Gradienten!

Kantenbasierte Segmentierung

3. Schritt: **Non-Maximum Suppression** (Kantenverdünnung)

   Das Ergebnis aus Schritt 2 liefert oft breite, verschwommene Kanten. Wir wollen aber dünne, präzise Linien.

   Der Algorithmus wandert entlang der Gradientenrichtung. Wenn ein Pixel nicht das lokale Maximum (der stärkste Wert) in seiner Nachbarschaft ist, wird es auf Null gesetzt (unterdrückt). Das Resultat sind "dünne" Kanten (1 Pixel breit).

Kantenbasierte Segmentierung

4. Schritt: **Schwellenwertverfahren (Hysterese)**

Entscheidung: Was ist eine "echte" Kante und was ist nur eine leichte Helligkeitsschwankung?

Man definiert zwei Schwellenwerte (Thresholds): T_{high} und T_{low}.

- Werte über Thigh sind sichere Kanten.
- Werte unter Tlow werden verworfen.
- Werte dazwischen sind "schwache Kanten". Sie werden nur dann akzeptiert, wenn sie direkt mit einer "sicheren Kante" verbunden sind.

Kantenbasierte Segmentierung

5. Schritt: **Kantenverbindung** (Edge Linking)

Nach der Filterung hat man oft noch Lücken in den Konturen. Um ein Objekt wirklich zu segmentieren (also vom Hintergrund zu trennen), muss die Kante geschlossen sein, also einzelne Kanten zu einer **Verkettung** (Kantenzug, Polygon) erweitern.

- Edge Relaxation ('erleichtert' die Kantenzugbedingung, indem die lokale Umgebung berücksichtigt wird)
- Graph searching (sucht Pixel, die 'Kandidaten' sind, um einzelne Kanten zu Pfaden zu verbinden, wobei Kosten minimiert werden)
- Hough transformation (robustes globales Verfahren zur Erkennung von Geraden, Kreisen oder beliebigen anderen parametrisierbaren geometrischen Figuren)

Hough Transformation

Vergleich Sobel vs. Canny

Region Growing

Ausgehend von einem **Seed Point** wird eine gesamte zusammenhängende Region als identifiziert, indem alle Pixel mit hinreichend ähnlichem Graustufenwert als dazugehörend definiert und auf den Seed Point Wert gesetzt werden.

Herausforderungen von Region Growing

- **Ähnlichkeitsbereich zu klein** nicht die gesamte gesuchte Region wird erkannt und markiert
- **Ähnlichkeitsbereich zu groß** Überlaufen des segmentierten Bereichs in Nachbarregionen, die eigenetlich abgetrennt werden sollen (Problem von Fehldiagnosen von Tumoren etc.)

similarity range too small

similarity range too large

Ablauf von Region Growing

Initialisierung

1. Wahl vom Startpunkt (Seed): Manuell oder automatisch wird die initiale Region gewählt.

Iterativer Prozess für jede bestehende Region:

2. Untersuche benachbarte Pixel oder Pixelgruppen, etwa die 4 oder 8 Nachbarn.
3. Vergleiche die Eigenschaften (z.B. Grauwerte) der Nachbarn mit der Region.
4. Füge Nachbarn zur Region hinzu, wenn sie ein Homogenitätskriterium erfüllen (innerhalb des Graustufenintervalls liegen).
5. Aktualisiere die Eigenschaften der Region (z.B. Durchschnittswert, Indikatormatrix der inneren Punkte).

Wiederhole den Prozess, bis kein weiteres Wachstum mehr möglich ist (while Schleife)

Abbruchkriterien

- Vorgegebene Anzahl an Iterationen: Ein einfaches Kriterium ist die Festlegung einer maximalen Anzahl von Durchläufen, z.B. 5 oder 10 Iterationen. Dies garantiert eine begrenzte Rechenzeit, kann aber zu suboptimalen Ergebnissen führen.
- Ausschluss aller Nachbarpixel: Der Algorithmus endet, wenn keine Nachbarpixel mehr die Kriterien für eine Aufnahme in die Region erfüllen

Flood Fill

The Flood Fill Algorithm: Filling Connected Areas
Start Point (Seed)
1. Select Start & Target Color
User chooses a starting pixel and the fill color.
2. Check & Change Color
Match Target?
If pixel matches target color, change to new color.
3. Spread to Neighbors (Recursion/Queue)
Algorithm checks adjacent pixels and repeats the process.
4. Stop at Boundary
Stops when a different color (boundary) or already filled pixel is hit.
Real-World Example
Used in image editing for filling shapes, mazes, and game development.
Before
After

Watershed Algorithmus

Nachbearbeitung nach der Segmentierung

Die Nachbearbeitung dient dazu, Fehler zu korrigieren, die Genauigkeit zu erhöhen und die Segmentierungsergebnisse für die weitere Analyse oder klinische Anwendung zu optimieren, um folgende Probleme zu beheben:

- **Unvollständige Segmentierung**: Viele Segmentierungsverfahren, etwa Schwellenwertverfahren, erzeugen keine vollständig zusammenhängenden Segmente, besonders bei stärkerem Rauschen.
- **Randungenauigkeiten**: Die Genauigkeit an den Rändern der segmentierten Objekte lässt oft zu wünschen übrig, besonders bei komplexen anatomischen Strukturen.
- **Eckige Kanten**: Einige Verfahren wie Erosion und Dilation erzeugen teilweise eckige Kanten, die eine Glättung erfordern.
- **Trennung überlappender Objekte**: Besonders bei der Zellsegmentierung kann die Trennung einzelner Zellen eine aufwändige Nachbearbeitung erfordern.

Morphologische Operationen

Morphologische Operationen sind eine Reihe von Algorithmen, die zur Analyse und Bearbeitung der Form und Struktur von Objekten in Bildern verwendet werden. Sie nutzen spezielle Strukturelemente, um Pixel in Bildern zu beeinflussen, für Anwendungen wie Objekterkennung, Rauschunterdrückung und Bildsegmentierung.

Grundoperationen der Morphologie

- Dilatation (Erweiterung): Fügt Pixel zu den Grenzen von Objekten hinzu. Diese Operation vergrößert die weißen Regionen eines binären Bildes.
- Erosion (Abtragung): Entfernt Pixel von den Grenzen von Objekten. Sie verkleinert die weißen Regionen, indem sie kleinere Strukturen entfernt.
- Öffnen: Eine Sequenzoperation, die zuerst eine Erosion gefolgt von einer Dilation ausführt. Diese Technik kann kleinere Objekte oder Störungen aus dem Bild entfernen.
- Schließen: Das Gegenteil von Öffnen, beginnend mit einer Dilation gefolgt von einer Erosion. Diese Methode füllt kleinere Löcher innerhalb von Objekten.

Dilation, Erosion

- **Dilatation** (Ausdehnung, Erweiterung)

Das aktive Pixel wird auf den **Maximumwert** der Umgebungspixel gesetz. Die **Region** wird dadurch **erweitert**

$$\begin{bmatrix} 11 & 45 & 23 \\ 108 & ? & 0 \\ \mathbf{210} & 71 & 98 \end{bmatrix} \Rightarrow \begin{bmatrix} 11 & 45 & 23 \\ 108 & \mathbf{210} & 0 \\ \mathbf{210} & 71 & 98 \end{bmatrix}$$

- **Erosion** (Abtragung, Verkleinerung)

Das aktive Pixel wird auf den **Minimumwert** der Umgebungspixel gesetz. Die **Region** wird dadurch **verkleinert**

$$\begin{bmatrix} 11 & 45 & 23 \\ 108 & ? & \mathbf{0} \\ 210 & 71 & 98 \end{bmatrix} \Rightarrow \begin{bmatrix} 11 & 45 & 23 \\ 108 & \mathbf{0} & 0 \\ \mathbf{210} & 71 & 98 \end{bmatrix}$$

Kombination der Methoden

- **Opening**, erst Erosion, dann Dilation

kleine Objekte und Hintergrundrauschen werden entfernt, große Objekte bleiben unbetroffen

- **Closing**, erst Dilation, dann Erosion

verbindet große Strukturen, indem ‘Löcher’ oder ‘Spalten’ geschlossen werden, die morphologisch nicht relevant sind

Opening - Strukturauftrennung

Original

0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	0	0	255	255	0	0	0	0
0	0	0	0	255	0	0	0	0	0
0	0	0	0	255	0	0	0	0	0
0	0	0	0	255	0	0	0	0	0
0	0	0	0	255	0	0	0	0	0
0	0	0	255	255	0	0	0	0	0
0	0	255	255	255	255	255	0	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0

Schritt 1

0	0	0	255	255	255	255	0	0	0
0	0	0	255	255	255	255	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	255	255	255	0	0	0	0
0	0	0	255	255	255	255	0	0	0
0	0	0	255	255	255	255	0	0	0

Schritt 2

0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	255	255	255	255	255	0	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0

Opening - Artefaktentfernung

Original

0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	255	0	0	0	0	0	0	0
0	0	255	0	0	0	0	255	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	255	255	255	255	255	0	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0

Schritt 1

0	0	0	255	255	255	255	0	0	0
0	0	0	255	255	255	255	0	0	0
0	0	0	0	0	255	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	255	255	255	0	0	0	0
0	0	0	255	255	255	255	0	0	0
0	0	0	255	255	255	255	0	0	0

Schritt 2

0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	0	0	255	255	255	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	255	255	255	255	255	0	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0
0	0	255	255	255	255	255	255	0	0

Closing - Spaltenschließung

Original

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
255	255	255	0	0	0	0	255	255	255
255	255	255	255	0	255	255	255	255	255
255	255	255	255	0	255	255	255	255	255
255	255	255	255	0	255	255	255	255	255
255	255	255	255	0	255	255	255	255	255
255	255	255	255	0	255	255	255	255	255
255	255	255	0	0	0	255	255	255	255
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Schritt 1

0	0	0	0	0	0	0	0	0	0
255	255	255	255	0	0	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	0	255	255	255	255	255
0	0	0	0	0	0	0	0	0	0

Schritt 2

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
255	255	255	0	0	0	0	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
255	255	255	0	0	0	255	255	255	255
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Transformationen von Bildern durch Matrizentransformationen

- Affine Transformationen
  - Skalierungen
  - Drehungen
  - Spiegelungen
- Digitale Transformation (Filterprozesse)
  - Filterprozesse
    - Räumlicher Hochpassfilter
    - Räumlicher Tiefpassfilter
  - Fouriertransformation
    - Fourier Hochpassfilter
    - Fourier Tiefpassfilter

Affine Transformationen

Identität $$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Spiegelung $$M = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Skalierung $$S = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

Drehung/Rotation $$R = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

Forward Mapping

Ermitteln der Gesamttransformationsmatrix T

$$T = S \cdot M \cdot R$$

Iteration Sie über alle Pixel (x,y) des Originalbildes

Berechnen Sie die neuen Koordinaten (x, y') durch Multiplikation mit Gesamttransformationsmatrix T

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = T \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Vom alten Bild ausgehend werden die Werte des neuen Bildes Schritt für Schritt ermittelt

Problem von Forward Mapping

Backward Mapping

Anstatt vom Quellbild ins Zielbild zu rechnen, für jeden Pixel (x',y') des Zielbildes auf die Koordinate im Originalbild (x,y) mithilfe der inversen Transformationsmatrix zurückrechnen:

$$\begin{pmatrix} x \\ y \end{pmatrix} = T^{-1} \cdot \begin{pmatrix} x' \\ y' \end{pmatrix}$$

Interpolation, wenn nichtganzahlige Koordinaten entstehen

Nearest Neighbor Interpolation (Runden auf den nächsten Nachbarn)

Bilineare Interpolation (gewichteter Mittelwert der 4 umliegenden Pixel)

Rand schwarz oder weiß auffüllen

Filterprozesse

Filterprozesse werden in der Bildbearbeitung eingesetzt, wobei Matrix-Kernels als Transformationsbasis zum Einsatz kommen.

- zur Rauschreduktion (Tiefpassfilter)
- zur Schärfung (Hochpassfilter)

Filter sind lokale Transformationen

Der räumliche Filter ist eine Koeffizientenmatrix, die einen kleinen Bereich des Eingabebildes auf einem einzelnen Bildpunkt des Ausgabebildes abbildet. Der Filter muß dann über das ganze Bild 'laufen', um das Eingabebild mathematisch zu 'falten'.

Prinzip der digitalen Filterung

In der digitalen Bildverarbeitung wird ein Bild als zweidimensionales Signal dargestellt (Signal = G[x,y]). G ist der Graustufenwert, der die aufgenommene reflektierte bzw. emittierte Strahlungsintensität misst. Die x- und y-Werte sind ganzzahlige (integer) Ortsvariablen von G. Da ein typischer Standard ist mit 8 Bit Genauigkeit diese Intensität zu speichern, stehen $2^8 = 256$ mögliche Zahlenwerte zur Verfügung (0, 1, . . . , 255).

In Matlab arbeiten wir daher mit dem Format $\underbrace{u}_{\text{unsigned}}\ \underbrace{int}_{\text{integer}}$ 8, also uint8.

Was ist Rauschen?

In medizinischen Bildern entsteht Rauschen durch verschiedene physikalische und elektronische Prozesse. Es äußert sich als zufällige Schwankungen der Pixelwerte und beeinträchtigt die Bildqualität, indem es Details verschleiert und die diagnostische Genauigkeit verringert.

- Elektronisches Rauschen: Entsteht durch thermische Bewegung von Elektronen in elektronischen Komponenten.
- Quantengerauschen: Ist eine Folge der diskreten Natur von Licht und führt zu statistischen Schwankungen in der Anzahl der detektierten Photonen.

Warum ist Rauschreduktion wichtig?

- Verbesserung der Bildqualität: Rauschreduktion macht Details besser sichtbar und erleichtert die Diagnose.
- Erhöhung des Signal-Rausch-Verhältnisses: Ein höheres Signal-Rausch-Verhältnis verbessert die Genauigkeit von quantitativen Messungen.
- Vorbereitung für weitere Bildverarbeitungsschritte: Rauschreduktion ist oft ein erster Schritt in komplexeren Bildverarbeitungspipelines.

Tiefpassfilter zur Rauschreduktion

- **Mittelwertfilter**: Ersetzt den Wert eines Pixels durch den Durchschnitt der Werte seiner Nachbarn. Glättet das Bild, aber kann auch Details verwischen.
- **Medianfilter**: Ersetzt den Wert eines Pixels durch den Median der Werte seiner Nachbarn. Effektiv bei der Entfernung von Impulsrauschen (Ausreißern).
- **Gaußfilter**: Gewichtet die Beiträge der Nachbarn nach einer Gaußverteilung. Glättet das Bild und bewahrt dabei besser Details.

Schärfung

- Hervorhebung von Details: Schärfung kann feine Strukturen wie Kanten und Details hervorheben, die für die Diagnose wichtig sind.
- Verbesserung der visuellen Wahrnehmung: Schärfere Bilder sind für den menschlichen Betrachter oft einfacher zu interpretieren.

Methoden zur Schärfung

- **Unscharfmaskierung**: Ein leicht unscharfes Bild wird vom Originalbild subtrahiert. Dadurch werden Kanten verstärkt.
- **Hochpassfilter**: Verstärken hohe Frequenzanteile, die für scharfe Kanten verantwortlich sind.

Matrix-Kernels und ihre Verwendung

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

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

Kantendetektion $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

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

Gaußfilter (3x3) $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

Idee und Zusammenhang von Hochpass und Tiefpass

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\text{Identität}} - \underbrace{\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}}_{\text{Mittelwert (Tiefpass)}} = \underbrace{\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}}_{\text{Kantendetektion (Hochpass)}}$$

Schritt von 3x3 zu 5x5

Transformation	3x3	5x5
Identität	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
Mittelwertfilter (TP)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
Kantendetektion (HP)	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$

Idee von Gaußfiltern

Pascalsches Dreieck

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

Gauß Kernel

$$\begin{matrix} 1\times \\ 2\times \\ 1\times \end{matrix} \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}}_{\text{Gewicht: } \frac{1}{16}}$$

$$\begin{matrix} 1\times \\ 4\times \\ 6\times \\ 4\times \\ 1\times \end{matrix} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 1 & 4 & 6 & 4 & 1 \\ 1 & 4 & 6 & 4 & 1 \\ 1 & 4 & 6 & 4 & 1 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}}_{\text{Gewicht: } \frac{1}{256}}$$

Prinzip hinter Gaussfilter

FIGURE 24-7
Separation of the Gaussian. The Gaussian is the only PSF that is circularly symmetric *and* separable. This makes it a common filter kernel in image processing.

	0.04	0.25	1.11	3.56	8.20	13.5	16.0	13.5	8.20	3.56	1.11	0.25	0.04
0.04	0	0	0	0	0	1	1	1	0	0	0	0	0
0.25	0	0	0	1	2	3	4	3	2	1	0	0	0
1.11	0	0	1	4	9	15	18	15	9	4	1	0	0
3.56	0	1	4	13	29	48	57	48	29	13	4	1	0
8.20	0	2	9	29	67	111	131	111	67	29	9	2	0
13.5	1	3	15	48	111	183	216	183	111	48	15	3	1
16.0	1	4	18	57	131	216	255	216	131	57	18	4	1
13.5	1	3	15	48	111	183	216	183	111	48	15	3	1
8.20	0	2	9	29	67	111	131	111	67	29	9	2	0
3.56	0	1	4	13	29	48	57	48	29	13	4	1	0
1.11	0	0	1	4	9	15	18	15	9	4	1	0	0
0.25	0	0	0	1	2	3	4	3	2	1	0	0	0
0.04	0	0	0	0	0	1	1	1	0	0	0	0	0

Kernel-Abtastung

Kernel-Animation

Beispiel: Tiefpassfilter mit Mittelwert

$$\text{Der } 3 \times 3 \text{ Mittelwertfilter } \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{wird angewendet auf die Pixelmatrix } \begin{bmatrix} 10 & 0 & 11 & 12 \\ 8 & 9 & 10 & 0 \\ 10 & 14 & 0 & 10 \\ 12 & 14 & 15 & 11 \end{bmatrix}$$

Beispiel: Tiefpassfilter mit Mittelwert

$$\begin{matrix} 10 & 0 & 11 & 12 \\ 8 & 9 & 10 & 0 \\ 10 & 14 & 0 & 10 \\ 12 & 14 & 15 & 11 \end{matrix}$$

Daraus ergibt sich als erster Pixel

$$(10*1+0*1+11*1+8*1+9*1+10*1+10*1+14*1+0*1)/9 = 8$$

Beispielmatrix nach 3x3 Mittelwertfilter: $\begin{bmatrix} ? & ? & ? & ? \\ ? & 8 & 7,2 & ? \\ ? & 10,2 & 9,2 & ? \\ ? & ? & ? & ? \end{bmatrix}$

Umgang mit Randwerten

- Extension: Die Randwerte werden orthogonal nach außen hin weiter kopiert, soweit wie notwendig.
- Spiegeln: Entlang des Randes werden die Werte nach außen gespigelt.
- Image Crop: Jedes Pixel des (resultierenden) Bildes, das Informationen von außerhalb braucht, wird abgeschnitten.
- Kernel Crop: Jeder Kernelwert, der Informationen von außerhalb des Bildes braucht, wird abgeschnitten.
- Konstanter Wert: Alle Werte außerhalb werden auf dieselbe Zahl gesetzt, z.B. 0 (schwarz).

Was wir wissen (sollten)

Die Fouriertransformation ist ein mathematisches Verfahren, das ein Signal oder eine Funktion aus dem Zeit- oder Ortsbereich in den Frequenzbereich überführt. Sie zerlegt ein komplexes Signal in eine Summe von Sinus- und Kosinuswellen unterschiedlicher Frequenzen und Amplituden. Dies ermöglicht es, die Frequenzkomponenten des ursprünglichen Signals zu analysieren und zu verstehen.

Fouriertransformation graphisch

Frequenzen

a Ton
b Klang (Trompete)
c Klang (Klavier)
d Geräusch (Wasserfall)
Auslenkung
Lautstärke
0 2 4 6 8 10 12
Zeit (ms)
0 1 2 3 4 5 6
Frequenz (kHz)

Fouriertransformation von Bildern

Bei der Anwendung der Fouriertransformation auf Bilder wird das Konzept auf zweidimensionale Daten erweitert:

- Bildzerlegung: Ein Bild wird als zweidimensionales Signal betrachtet und in seine Frequenzkomponenten zerlegt.
- Ortsfrequenzen: Statt zeitlicher Frequenzen werden hier Ortsfrequenzen analysiert, die die Änderungsrate der Pixelintensitäten in verschiedenen Richtungen des Bildes repräsentieren.
- Anwendung: Die Fouriertransformation wird zuerst zeilenweise und dann spaltenweise auf das Bild angewendet.
- Frequenzdarstellung: Das Ergebnis ist eine Darstellung des Bildes im Frequenzraum, die Informationen über die Verteilung der Ortsfrequenzen enthält.
- Bildverarbeitung: Im Frequenzraum können bestimmte Bildverarbeitungsoperationen wie Filterung oder Rauschunterdrückung effizienter durchgeführt werden.

Fourierfilterung

Anstatt auf Basis der Pixelwerte wird auf Basis der Frequenzen nach Durchführung einer Fouriertransformation gefiltert.

Frequenz im Raum

1 keine räumliche Frequenz der Helligkeitswerte in x- und y-Richtung; d.h. flaues, kontrastloses Bild mit weitgehend Einheits-DN-Werten => räumliche Frequenz = 0 in beide Richtungen 2. geringe räumliche Frequenz, also schwache räumliche Frequenz in x und y-Richtung; d.h. sanfte Übergänge im Kontrast 3. hohe räumliche Frequenz in x und y-Richtung; d.h. extreme Kontraste (Muster) zwischen den Pixelgruppen => Wiederholung von abrupten Variationen der Helligkeitswerte in x- und y-Richtung

X
Y
Ortsbereich
X, Y-Ortskoordinaten
Frequenzbereich
mit den einzelnen
Ortsfrequenzen

How to interpret FFT images

Gaussian Noise → No pattern

DC term
Greyscale-values: from 0 to 255

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

sine wave (low frequency)

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

sine wave (medium frequency)

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

sine wave (high frequency)

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

sine wave - horizontal
(medium frequency)

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

sine wave – diagonal
(medium frequency)

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

superposition of
3 sine waves with different
frequencies

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

superposition of numerous sine wave results in peak

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

combination of sine waves in vertical and diagonal directions

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

superposition of sine waves
in vertical and diagonal
directions

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

superposition of sine waves
in horizontal and vertical
directions

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

How to interpret FFT images

superposition of sine waves including noise

brightness of pixel in fft is proportional to amplitude

Source: http://qsimaging.com/ccd_noise_interpret_ffts.html

Die räumliche Funktion f(x,y), also unsere Graustufenwerte, werden also Fourier transformiert durch

$$f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u,v) e^{j2\pi(ux+vy)} \, du \, dv.$$

Da $e^{ix} = \cos x + i \sin x$ gilt, bedeutet das die Transformation in Sinus und Cosinusfunktionen unterschiedlicher Frequenzen.

Das bedeutet also, dass ein Bild folgendermaßen aufgesplittet wird:

Trnaformationen und Filter

Mithilfe des Frequenzraums können wir nun Hochpassfilter oder Tiefpassfilter definieren, indem das Zentrum oder der Rand weggelassen wird. Diese Transformationen sind nicht auf lokale Bildinformationen sondern globale Bildinformation aufgebaut.

Überblick über Bildbearbeitung

Eigenschaft	Beschreibung
Modale Daten	Diskrete Werte (z.B. CT-Werte, PET-Counts), Kontinuierliche Werte (z.B. Intensitätswerte in MRT-Bildern)
Mehrdimensionale Daten	2D-Bilder (Röntgen, einzelne CT-Schnitte), 3D-Volumen (CT, MRT), 4D-Daten (Zeitreihen, funktionelle Bildgebung)
Rauschen	Zufällige Variationen in den Pixelwerten, Beeinträchtigt die Bildqualität
Artefakte	Verzerrungen im Bild, verursacht durch physikalische oder technische Faktoren
Großer Datenumfang	Hohe Auflösung, viele Schnittebenen, Erfordert effiziente Speicher- und Verarbeitungsmethoden
Spezialisierte Dateiformate	DICOM (Digital Imaging and Communications in Medicine), NIfTI (Neuroimaging Informatics Technology Initiative)