

Praktische Übung: Computer Vision für medizinische und industrielle Anwendungen

Versuch 3: Regionenbasierte Segmentierung

Aufgabe 1 - Region-Growing

- a) Schreiben Sie eine Funktion `regionGrowing(I, xStart, yStart, threshold)`, die einen Region-Growing-Algorithmus zur Segmentierung implementiert, der auf einem Graustufenbild `I` arbeitet. Ausgehend vom Saatpunkt `(xStart, yStart)` wird jeweils der ähnlichste benachbarte Pixel zur Region hinzugefügt. Hierzu werden alle angrenzenden Bildpunkte mit dem Mittelwert der bereits segmentierten Region verglichen. Der Algorithmus stoppt, sobald keine Nachbarn mehr gefunden werden können, deren Ähnlichkeit unter dem angegebenen Schwellwert liegt.

Hinweise zur möglichen Implementierung:

- Die Nachbarn können in einer separaten Matrix/Liste vorgehalten werden.
- Achten Sie auf die Randbehandlung.
- In MATLAB kann auf zwei Arten auf Arrays zugegriffen werden: über lineare Indizes (`M(ind)`) und Subscript-Werte (`M(x, y)`). Während die (x, y) -Koordinaten einfacher mit den entsprechenden Matricelementen in Verbindung gebracht werden können, sind die eindimensionalen Indizes hilfreich, wenn man direkt auf eine beliebige Menge an Bildpunkten zugreifen möchte. Die Befehle `sub2ind` und `ind2sub` können verwendet werden, um zwischen den Varianten zu wechseln.
- Zeilen/Spalten in einer Matrix können gelöscht werden, indem den entsprechenden Elementen ein leeres Array zugewiesen wird (z.B. `M(1, :) = []`).

- `ismember` überprüft, ob ein Element in einem Array vorhanden ist. Je nach Länge der Liste, kann eine wiederholte Abfrage allerdings recht lange dauern. Denken Sie über andere Varianten nach, um zu überprüfen, ob ein Bildpunkt schon bekannt ist.
- Beachten Sie gegebenenfalls die Datentypen, auch wenn Sie nicht explizit in MATLAB angegeben werden. Mit `uint8` können z.B. keine negativen oder gebrochenen Zahlen abgebildet werden. Mit `im2double` kann ein Bild in Fließkommadarstellung konvertiert werden.

Sie können sich für die Implementierung auch an dem folgenden Pseudo-Code orientieren:

Input: Image $I \in \{0, \dots, 255\}^{m \times n}$

Input: Seedpoint $(x_{start}, y_{start}) \in \{(x, y) | 1 \leq x \leq m \wedge 1 \leq y \leq n\}$

Input: Schwellwert $\in \mathbb{R}^+$

Output: Segmentierung $S \in \{0, 1\}^{m \times n}$

- 1: Initialisiere Segmentierung: $S \leftarrow 0^{m \times n}$
- 2: Initialisiere Liste verfügbarer Punkte: $apList \leftarrow (x_{start}, y_{start})$
- 3: Initialisiere Mittelwert der segmentierten Region: $regionMean \leftarrow I(x_{start}, y_{start})$
- 4: **while** $apList \neq \emptyset$ **do**
- 5: Bestimme nächsten Punkt: $p \leftarrow \underset{x \in apList}{\operatorname{argmin}} |I(x) - regionMean|$
- 6: **if** $|I(p) - regionMean| > \text{Schwellwert}$ **then**
- 7: **return**
- 8: **end if**
- 9: Entferne p aus den verfügbaren Punkten: $apList \leftarrow apList \setminus p$
- 10: Füge p zu der segmentierten Region: $S(p) \leftarrow 1$
- 11: Bestimme **neue!** Nachbarn von p : $nbList$
- 12: Aktualisiere verfügbare Punkte: $apList \leftarrow apList \cup nbList$
- 13: Aktualisiere $regionMean$
- 14: **end while**

- b) Schreiben Sie ein eigenes Testskript, welches ein Bild einliest und interaktiv den Startpunkt und Schwellwert abfragt. MATLAB stellt hierzu verschiedene Möglichkeiten zur Verfügung (`ginput`, `getpts`, `input`, `inputdlg`). Stellen Sie anschließend geeignet das Bild und die Segmentierung dar. Mit dem Befehl `contour` kann die Segmentierung z.B. als Kurve auf dem Originalbild dargestellt werden.

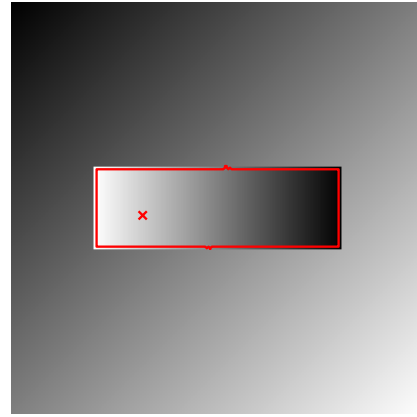


Abbildung 1: Beispielhafte Ergebnisse des Region-Growing-Algorithmus

Aufgabe 2 - Watershed-Transformation

- Machen Sie sich mit der MATLAB-Funktion zur Wasserscheidentransformation vertraut (`watershed`).
- Nutzen Sie die Funktion, um die Lungenbläschen im Bild `Emphysem.png` zu segmentieren. Die Wasserscheidentransformation auf dem Originalbild wird nicht das gewünschte Ergebnis liefern. Überlegen Sie sich geeignete Schritte zur Vorverarbeitung. Die Funktion `plotWatershed.m` kann verwendet werden, um anschließend das Ergebnis darzustellen.

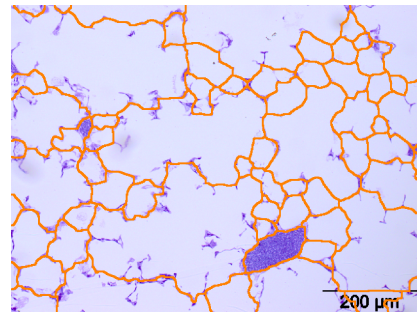
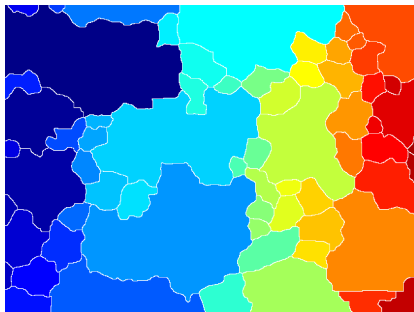


Abbildung 2: Beispielhafte Ergebnisse der Wasserscheidentransformation

Kontrollfragen

- Warum kann das Rechteck im Bild `rectangle.png` nicht segmentiert werden? Welche weiteren Ähnlichkeitsmaße könnten im Region Growing verwendet werden, um das Problem zu lösen?
- Warum muss ein Bild in den meisten Fällen vorverarbeitet werden, um die Wasserscheidentransformation zur Segmentierung verwenden zu können?