

Bildverarbeitung Praktikum

Übung 5, Abgabe 21.05.2020

Dr. Christina Gillmann

May 14, 2020

1 Kanten-basierte Segmentierung

Implementieren sie die kantenbasierte Segmentierung. Sie müssen hierzu folgende Schritte durchlaufen.

a) Wenden sie einen Kantendetektor an. Die Wahl des Kantendetektors überlasse ich ihnen.

b) Schreiben sie einen Algorithmus, der bei ihren detektierten Kanten bei einem beliebigen Kantenpixel startet und dann entlang der Nachbarkanten läuft bis er wieder beim Startpixel landet. Hierzu müssen sie folgendes beachten.

1. Die Nachbarschaft kann unterschiedlich definiert werden (siehe Vorlesung 1). Sie können gerne versch. Ansätze ausprobieren.

2. Sie müssen es schaffen sich zu merken welche Kanten bereits durchlaufen wurden. Verwenden sie am besten ein Bild mit sehr deutlichen Kanten.

2 Region-Growing Segmentierung

Implementieren sie eine region-growing Segmentierung. Gehen sie dabei wie folgt vor.

a) Sie müssen einen Startpunkt (Pixel) definieren, der in der Region liegt, die sie segmentieren möchten.

b) Sie müssen einen Schwellenwert definieren, der beschreibt wie weit sich Pixel von der Region unterscheiden dürfen.

c) Ausgehend von dem Startpunkt müssen sie die Nachbarschaft des Pixels betrachten (auch hier können sie mit versch. Definitionen der Nachbarschaft arbeiten). Wenn die betrachteten Pixel weniger unterschiedlich zum Startpixel sind als der Schwellenwert angibt, dann wird der Pixel zur Region gezählt. Beachten sie dabei zwei Dinge:

1. Wenn sie Pixel zur Region hinzufügen, müssen sie eine Variable mit dem aktuellen Mittelwert der Region speichern. Dieser wird dann für das weitere Entscheiden über die Zugehörigkeit weiterer Pixel verwendet.

2. Sie müssen eine Liste führen in der alle Pixel liegen, die sich am aktuellen Rand ihrer Region befinden. Die Liste wird erweitert, wenn sie neue Pixel in einer Nachbarschaft in

betracht ziehen. Die Liste wird verkürzt, wenn sie eine Entscheidung getroffen haben, ob der Pixel zur Region gehört oder nicht. In dieser Liste darf es keine Duplikate geben. Dies erreichen sie durch:

```
1 np.unique(X)
```