

Bildverarbeitung Praktikum

Übung 3, Abgabe 29.05.2024

Dr. Daniel Wiegrefe

May 15, 2024

1 Kantendetektion

In der Vorlesung haben wir verschiedene Ansätze zur Kantendetektion ausprobiert. Diese sollen Sie nun an einem beliebigen Bild ausprobieren (am besten alle auf dem gleichen Bild). Erstellen Sie dazu einen Plot, der mehrere Subplots erhält. Somit können Sie die Ergebnisse direkt vergleichen.

Implementieren Sie die Filter als Maske und nutzen Sie nicht die vorgefertigten Filter aus OpenCV und Co!

- a) Implementieren Sie den Laplace Kantendetektor.
- b) Implementieren Sie den Sobel Kantendetektor.
- c) Implementieren Sie den LoG Filter.
- d) Implementieren Sie den DoG Filter.
- e) Überlegen Sie sich welche Vor und Nachteile die jeweiligen Kantendetektoren haben.
- f) Skalieren Sie nun ihr genutztes Bild auf verschiedene Größen (z.B. 50 und 200 Prozent). Lassen Sie die Kantenfilter auf den verschiedenen Auflösungen laufen und vergleichen Sie die Ergebnisse.

2 Dilatation und Erosion

Erstellen Sie ein binären Bild (beispielsweise durch eine Segmentierung).

- a) Wenden Sie eine Dilatation an. Dabei benötigen Sie den entsprechenden Kernel und die Funktion

```
1 img_erosion = cv2.erode(img, kernel, iterations=1)
```

- b) Wenden Sie eine Erosion an. Auch hier benötigen Sie den Kernel, sowie die Funktion

```
1 img_dilation = cv2.dilate(img, kernel, iterations=1)
```

- c) Kombinieren Sie die Funktionen aus a) und b) um ein Opening zu implementieren.
- d) Kombinieren Sie die Funktionen aus a) und b) um ein Closing zu implementieren.
- e) Zeigen Sie alle Ergebnisse in einem Plot an!

3 Ausdünnung

Implementieren Sie den Algorithmus zur Ausdünnung aus der Vorlesung. Sie müssen dazu

- a) Die entsprechenden Filter anwenden.
- b) Die Filter so lange iterieren, bis sich am Bild nichts mehr ändert.

Wenden Sie ihren Algorithmus auf ein binäres Bild an.