

# BIV-Übung 5

## 1 Kanten

**Achtung:** Verwenden Sie für Faltungen die Funktion `conv2` (siehe auch den Hilfetext dazu, insbesondere ist die Option `same` nützlich) und **geben Sie Faltungsmasken selbst ein!**

*Tipp:* Nützlich zur Anzeige: `imshow(meinBild, [])`. Damit wird automatisch eine Grauwertskalierung auf den maximalen Kontrast durchgeführt.

- Berechnen Sie von dem Bild `Dreieck.png` durch Berechnung der ersten Ableitung einmal die horizontalen und einmal die vertikalen Kanten. Welche Werte des jeweiligen Ergebnisbildes sind positiv, welche negativ? Warum? Überlegen Sie sich, in welche Richtung der Gradient auf den Seiten des Dreiecks zeigt.
- Lassen Sie sich den *Betrag* des Gradienten des genannten Bildes berechnen (mit Hilfe den Ergebnissen der vorigen Teilaufgabe!) und anzeigen.  
*1c+d erst für 25.4. - also erstmal überspringen*
- Wenden Sie den Laplace-Operator auf das Bild an und lassen Sie sich das Ergebnis anzeigen. Können Sie die Kanten sehen? Wo genau liegen die Kanten?
- Verbessern Sie die Schärfe des Bildes `Mond.png` durch Scharfzeichnen (siehe Foliensatz „Lokale Operatoren“, Kapitel 3.3). Verwenden Sie für die Faltung die Funktion `conv2`.

## 2 Kanten und Rauschen

In dieser Aufgabe geht es um das Bild `BIVn.png`, das Sie in den Dateien zur Aufgabe auf der Moodle-Seite finden. Ziel ist, die Kanten der Zeichen zu bestimmen.

- Wie viele verschiedene Grauwerte hat das Bild?
- Berechnen Sie ein „Kantenbild“, in dem die Stärke der Kanten zu sehen ist.
- Glätten Sie das Bild mit verschiedenen starken Gauß-Filtern und lassen Sie sich die Ergebnisse anzeigen. Übernehmen Sie die Bilder in die Auswertung und beschriften Sie sie mit der Stärke der Glättung (z.B. Größe Maske und Anzahl der hintereinander durchgeführten Faltungen).
- Berechnen Sie von diesen geglätteten Bildern die „Kantenbilder“ (Stärke der Kanten).
- Versuchen Sie es jetzt auch umgekehrt: das Kantenbild aus Teilaufgabe ?? glätten. Warum ist das Ergebnis ein anderes, obwohl die Faltung kommutativ ist?
- Optional: Erstellen Sie ein Farbbild, in dem starke Kanten (stärker als eine von Ihnen gewählte Schwelle) als grüne Überlagerung über dem Originalbild dargestellt sind.

## 3 Reduktion von Rauschen

**Achtung:** Verwenden Sie für Faltungen die Funktion `conv2` und geben Sie Faltungsmasken selbst ein!

- Vermindern Sie das Rauschen des Bildes `HausNoisy.png` durch Filterung mit einem Mittelwerts- (Rechteck-), Gauß- und Median-Filter, jeweils der Größe<sup>1</sup> 5. Implementieren Sie die Filter (sofern das möglich ist) so, dass die Berechnung möglichst schnell ist! Eine nützliche MATLAB-Funktion ist `medfilt2`.

Wo sind Artefakte (Bildfehler) zu erkennen? (Achten Sie insbesondere auf die Rahmen des linken Fensters im Turm bei dem Mittelwert-Filter im Vergleich zum Gauß-Filter).

---

<sup>1</sup>Mit „Größe“ ist die Größe in jede Raum-Richtung gemeint, z.B. hat also in 2D ein 5x5 Filter die Größe 5.

- b) Nun soll eine einfache *inhomogene* Rauschunterdrückung implementiert und auf das Bild angewendet werden.

Die Idee ist, an den Stellen, an denen „deutliche“ Kanten sind, die Werte des Originalbildes zu verwenden; an allen übrigen Stellen sollen die Werte eines homogen (z.B. mit einem Gaußfilter) geglätteten Bildes verwendet werden.

1. Welche Berechnung (aus der Aufgabe zu Kanten) liefert einen Wert, der angibt, wie „deutlich“ eine Kante ist? Berechnen Sie ein solches Bild und übernehmen Sie es – wie alle anderen berechneten Bilder – in die Auswertung.
2. Lassen Sie sich ein Bild berechnen, das nur „deutliche“ Kanten enthält: an diesen Stellen soll das Bild weiß (1) sein, überall sonst schwarz (0). Sie berechnen damit also ein „binäres Kantenbild“.
3. Setzen Sie nun die oben beschriebene Idee um: Benutzen Sie das gerade berechnete binäre Kantenbildes um das Originalbild und das mit dem homogen Gaußfilter geglättete Bild zu kombinieren.
4. Vergleichen Sie die Ergebnisse der Gaußfilterung und dieser inhomogenen Filterung (z.B. im Bereich der Dachziegeln).

## 4 Verkleinern eines Bildes

In den Dateien zur Aufgabe befindet sich die Bilddatei `ringe.png`.

- a) Verkleinern Sie das Bild mit der MATLAB-Funktion `imresize(bild, 0.1, 'nearest')`. Bei diesem Aufruf wird nur jeder 10-te Pixel in beide Richtungen in das verkleinerte Bild übernommen. Was beobachten Sie?
- b) Glätten Sie das Original-Bild mehrfach (mindestens 40x bei einer Maske der Größe 3; probieren Sie es aus!) und führen Sie danach dann wieder die Verkleinerung durch. Vergleichen Sie die Ergebnisse mit der vorigen Teilaufgabe.