Übung 4.2

Lernziele

Die Studierenden...

- können Kanten mittels Laplace-Filter detektieren
- können Gauss- mit Laplace-Filtern kombinieren, um Rauschen bei der Kanten-Detektion zu unterdrücken
- **kennen den Vorteil von Bild-Pyramiden bei Detektions-Aufgaben**

Aufgaben

1. Die Funktion `cv2.resize()` nimmt als Parameter keine Skalierungs-Faktoren, sondern Ziel-Grössen (Anzahl Spalten, Anzahl Zeilen). Programmieren Sie eine Funktion, die ein Bild anhand eines Skalierungs-Faktors gleichmässig skaliert. Beachten Sie, dass `cv2.resize()` **nur Integer** als Ziel-Grössen annimmt.
2. Laden Sie das Bild «Saturn_1.jpg» ein und detektieren Sie die Kanten mit einem 5x5 LoG-Filter bei verschiedenen Skalierungsfaktoren. Bei welchem Faktor werden die Kanten am besten detektiert? Zur Visualisierung der Unterschiede können Sie die Histogramme der Kanten-Bilder plotten.

Bonus

Mit Hilfe von Bild-Pyramiden lassen sich Bilder in unterschiedlichen Auflösungen mit dem gleichen Filter untersuchen. Programmieren Sie eine Funktion, die den besten Skalierungs-Faktor ermittelt, um Kanten mit LoG oder Laplace-Filtern zu detektieren.

Strukturierung:

- For-Schleife (verschiedene Skalierungs-Faktoren durch-iterieren):
  - Bild skalieren
  - Kanten detektieren
  - Standardabweichung ermitteln (vorher Kanten-Bilder normalisieren / auf Bereich [0;1] skalieren)
  - Skalierungs-Faktor in einer Liste speichern
  - Standardabweichung in einer Liste speichern
- Nach for-Schleife den Skalierungsfaktor ausgeben, bei dem die maximale Standardabweichung im Kanten-Bild ermittelt wurde

Da eine hohe Standardabweichung bei extrem kleiner Skalierung auch nicht mehr hilft, sollte das Beurteilungskriterium nicht nur der Kontrast (Standardabweichung) sein. Stattdessen sollte ein neuer «Kontrast-Score» berechnet werden, der auch die Auflösung berücksichtigt.

1. Kontrast-Gewinn berechnen: Standardabweichung des Original-Kantenbildes von den Standardabweichungen der skalierten Kantenbilder abziehen:

   $$\sigma_{Gewinn} = \sigma_i - \sigma_0$$ mit *i* den Indizes der skalierten Bilder

2. Kontrast-Score berechnen: Kontrast-Gewinn mit Skalierungsfaktoren multiplizieren
3. Skalierungsfaktor mit dem höchsten Kontrast-Score ausgeben

   → So werden Skalierungen bevorzugt, die proportional zur Skalierung einen höheren Kontrast-Gewinn erzielen