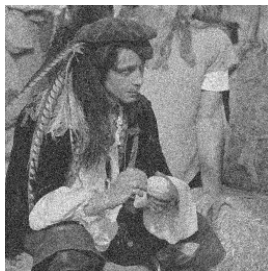


## Numerische Mathematik I

Wintersemester 2021/22

### Programmieraufgabe 1

In dieser Programmieraufgabe werden wir eine Reihe von linearen und nicht-linearen Filtern in Python implementieren, um das Rauschen auf den folgenden drei Bildern zu entfernen.



Die unten geforderten Funktionen sollten selbstständig implementiert werden. Insbesondere ist es nicht ausreichend, z.B. einfach die Funktion `skimage.filters.median` zu verwenden, wenn ein Medianfilter implementiert werden soll. Um die Bilder einzulesen, verwende `skimage` aus dem `scikit-image`-Paket.

1. Implementiere jeweils eine Funktion, welche den gewichteten Mittelwert und den gewichteten Median

$$\bar{A}_W := \sum_{k=1}^n \sum_{l=1}^m W(k, l) A(k, l) \quad \text{und} \quad \tilde{A}_W := \text{median}_W(A)$$

für eine beliebige Matrix  $A \in \mathbb{R}^{n \times m}$  und die Gewichte  $W \in \mathbb{R}^{n \times m}$  mit  $W(k, l) \geq 0$  und  $\sum_{k=1}^n \sum_{l=1}^m W(k, l) = 1$  berechnet. Für den Median darf die Funktion `sort` von NumPy verwendet werden. Um die beiden Funktionen zu testen, erzeuge 100 Zufallsmatrizen unterschiedlicher Größe und vergleiche die Ergebnisse mit den NumPy-Funktionen `mean` und `median` für den gleichgewichteten Fall. Gebe hierfür die maximale absolute Abweichung aus.

2. Implementiere den gewichteten Mittelwertfilter

$$U(k, l) := \sum_{(i, j) \in \mathcal{N}(k, l)} B(i, j) W(i - k, j - l)$$

mit  $\mathcal{N}(k, l) := \{(i, j) : |i - k| \leq s, |j - l| \leq s\}$  für ein Bild  $B \in \mathbb{R}^{n \times m}$  und Gewichte  $W \in \mathbb{R}^{(2s+1) \times (2s+1)}$  mit  $W(k, l) \geq 0$  und  $\sum_{k=-s}^s \sum_{l=-s}^s W(l, k) = 1$ . Beachte, dass die Gewichte  $W$  hier nicht gespiegelt werden! Für die Pixel am Rand setze das Bild geeignet fort. Glätte alle drei Bilder, mit geeigneten rechteckigen Mittelwertfiltern und geeigneten Gauß-Filtern und gebe die Ergebnisse auf den Bildschirm aus. Gib die Gewichte für einen Gauß-Filter ebenfalls als Bild aus.

3. Implementiere den gewichteten Medianfilter

$$U(k, l) := \text{median}_W(B(i, j))_{(i, j) \in \mathcal{N}(k, l)}$$

mit  $\mathcal{N}(k, l) := \{(i, j) : |i - k| \leq s, |j - l| \leq s\}$  für ein Bild  $B \in \mathbb{R}^{n \times m}$  und Gewichte  $W \in \mathbb{R}^{(2s+1) \times (2s+1)}$  mit  $W(k, l) \geq 0$  und  $\sum_{k=-s}^s \sum_{l=-s}^s W(l, k) = 1$ . Für die Pixel am Rand setze das Bild geeignet fort. Glätte alle drei Bilder, mit (ungewichteten) Median-Filtern geeigneter Größe und gewichteten Median-Filtern mit Gauß-Gewichten geeigneter Varianz und gebe die Ergebnisse auf den Bildschirm aus.

4. Implementiere den bilateralen Filter für Gauß-Gewichte. Für die Pixel am Rand setze das Bild geeignet fort. Glätte alle drei Bilder mit passenden Parametern und gebe die Ergebnisse auf den Bildschirm aus.

Verwende außer `skimage`, `numpy` und `matplotlib` keine weiteren Pakete.