

Bildverarbeitung Praktikum

Übung 2, Abgabe 05.05.2022

Dr. Daniel Wiegrefe

April 19, 2023

1 Fourier Transformation

In dieser Aufgabe sollen Sie die diskrete Fouriertransformation auf ein 1-Dimensionales Signal berechnen. Nutzen Sie hierfür folgende Funktionen:

$$\sin(x) \tag{1}$$

und

$$\sin(x) + (3 * \sin(2 * x + 1) - 1) \tag{2}$$

Diskretisieren diese Funktionen im Bereich von 0 bis 4π mit einer ausreichenden Abtastrate. Berechnen Sie nun mittels der DFT die Fouriertransformierte der beiden Funktionen. Plotten Sie zudem den Realanteil der Ergebnisse, um die Frequenzen der Funktionen zu bestimmen.

Hinweis: Sie sollen die DFT selbst programmieren und keine Library dafür nutzen. Nutzen Sie hierzu die Summenformel Variante der DFT.

2 Box Filter

Für diese Aufgabe benötigen Sie eine weitere Bibliothek in Python, die Sie durch die Anaconda Shell wie folgt laden können.

```
1 conda install -c conda-forge opencv
```

OpenCV ist eine Bibliothek, die Ihnen sehr viele Möglichkeiten der Bildverarbeitung liefert. Unter anderem die Funktion

```
1 result = cv2.filter2D(img,-1,kernel)
```

Hiermit können Sie einen beliebigen Kernel auf ein Bild anwenden. Hierzu müssen Sie sich einen Kernel erstellen. Dies können Sie durch ein neues Numpy Array erreichen, wie auch in Aufgabe 1.

a) Erstellen sie einen Box Kernel (Größe 5x5) und wenden Sie ihn auf ein beliebiges Bild an. Vergessen sie die Normalisierung nicht.

b) Manipulieren Sie den Wert des zentralen Eintrags in Ihrem Kernel. Nutzen Sie dabei immer größere Werte. Achten Sie darauf, dass Sie auch hier entsprechend normalisieren müssen. Was fällt bei beliebig großen Werten auf?

3 Implementierung des Gaußfilters

Wie bei allen anderen Programmiersprachen Funktionen zu erstellen. Diese können direkt in Ihren Programmcode eingefügt werden. Eine kurze Zusammenfassung finden sie hier:

<https://pythonbuch.com/funktion.html>

Sie sollen eine Funktion schreiben, die Ihnen einen Gaußfilter mit beliebiger Größe (Input als Parameter der Funktion) erstellt.

Nutzen Sie die Funktion `filter2D` von `openCV` um verschiedene Filter auf ein Bild anzuwenden.

```
1 result = cv2.filter2D(img, -1, kernel)
```

a) Schreiben Sie eine Funktion, die Ihnen beliebig große Gaußfilter erzeugt.

b) Nutzen Sie Ihre Funktion um Gaußfilter der Größe 3, 7, und 15 auf ein beliebiges Bild anzuwenden. Was fällt Ihnen auf?