



# Image Processing & Understanding

## Grundlagen der Bildverarbeitung

### Praktische Aufgaben - Aufgabe 2

Wintersemester 16/ 17  
Bildverarbeitung und Bildverstehen  
Prof. Klaus Tönnies,  
Tim König, Johannes Steffen

**Abgabe:** Die Abgabe erfolgt spätestens am 15.11.2016 um 08:00 Uhr. Jedes Team muss genau 4 Dateien einreichen: *p02\_myfft2.m*, *p02\_fft2\_test.m*, *p02\_modify.m*, *p02\_modify\_test.m*. Allgemeine Details zur Einreichung finden Sie auf der Webseite!

1. Schreiben Sie eine Funktion *img\_myfft = p02\_myfft2(img)* mit einer Bildmatrix (Grauwertbild vom Typ *double(!)*) als Eingabeparameter, die das Bild in den Fourierraum transformiert und so zurückgibt, dass das Fourierspektrum um den Nullpunkt zentriert ist, d.h. niedrige Frequenzen sind im Bildmittelpunkt und die hochfrequenten Anteile sind am Bildrand. Die Berechnungsvorschrift der (nicht-zentrierten) 2D-Fouriertransformation für einen Bildpunkt  $f(m, n)$  und die Fouriertransformierte  $F$  lautet:

$$F(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \exp^{-i2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)}.$$

*Hinweis 1:* Die Transformation in den Frequenzraum soll von Ihnen eigenständig und nicht mittels schon vorhandener Funktionen, wie beispielsweise *fft* oder *fft2*, erfolgen.

*Hinweis 2:* Beachten Sie, dass die Indizierung von Arrays und Matrizen in Matlab/Octave einsbasiert ist!

*Hinweis 3:* Versuchen Sie, wenn möglich, ohne For- oder While-Schleifen auszukommen (es gibt jedoch keinen Punktabzug bei sinnvollem Gebrauch dieser).

2. Schreiben Sie ein Matlab/Octave - Skript *p02-fft2-test*, in dem folgende Schritte ausgeführt werden:

- Einlesen des Bildes *p02-Bild01.tif* und anschließende Typenkonvertierung zum Typ *double*
- Transformation des eingelesenen Bildes in den Frequenzraum mittels Ihrer Funktion *p02-myfft2* aus Aufgabe 1 (*img-myfft*).
- Nutzen Sie zusätzlich die vorhandenen Matlab/Octave-Funktionen um das eingelesene Bild in den Frequenzraum zu transformieren und zu zentrieren (*img-fft*).
- Lassen Sie sich die Amplitudenspektren beider fouriertransformierten Bilder visuell anzeigen. Logarithmieren und spreizen Sie die Amplitudenspektren zur besseren Erkennbarkeit.
- Transformieren Sie beide Bilder zurück in den Ortsraum und testen Sie ihr transformiertes Bild *img-myifft* mittels folgender Bedingung:

$$\max(\text{abs}(\text{imag}(\text{img-myifft}(:)))) / \max(\text{abs}(\text{img-myifft}(:))) < \text{sqrt}(\text{eps}).$$

Falls die Bedingung erfüllt ist, geben Sie den Realteil des Bildes in einer Figure-Umgebung aus. Falls nicht ist Ihre Berechnung des fouriertransformierten Bildes ungenau oder falsch.

3. Schreiben Sie eine Funktion *img-mod = p02-modify(img-fft, x, y)*, der ein bereits zentriertes fouriertransformiertes Bild *img-fft* und zwei Koordinaten übergeben werden. Die Funktion soll ein neues Bild *img-mod* erzeugen, das an der durch die Koordinaten spezifizierten Stelle den gleichen Wert wie *img-fft* hat und ansonsten Null ist. Transformieren Sie *img-mod* anschließend in den Ortsraum und geben Sie es aus der Funktion zurück.
4. Testen Sie in einem Skript *p02-modify-test* Ihre Funktion aus Aufgabe 3 mit den Koordinaten (2,5), (-2,5), (-2,-5), (16, -4) und einem zentrierten fouriertransformierten Bild *img-fft*. Geben Sie die Ergebnisse in einer Figure-Umgebung mit 4 Teilbildern aus. Zeigen Sie zur besseren Erkennbarkeit nur den Realteil an und spreizen Sie die Intensitätswerte adäquat.

Geben Sie anschließend der Funktion aus Aufgabe 3 die Werte aller Koordinaten, die einen euklidischen Abstand kleiner als 20 vom zentrierten Ursprung haben, nacheinander als Eingabe. Addieren Sie die Teilergebnisse, zeigen Sie das Ergebnis an und interpretieren Sie dieses in den Kommentaren.

## Hinweise

- Hilfreiche Funktionen, die Sie verwenden *können* (aber nicht müssen!): *fft*, *fft2*, *ifft*, *ifft2*, *ifft2*, *fftshift*, *ifftshift*, *log*, *abs*, *im2double*, *subfigure*.