

EBME/CSDS 361/461. Homework 5. Spring.
David Wilson, PhD
Department of Biomedical Engineering
Homework: Fourier

All homework assignments must be completed individually (no teams/groups).

Your responses to each question to each should include:

- A short discussion of each problem. **Code/images alone will not suffice.** A brief description/explanation should accompany each image/algorithm.
- Any processed images
- All MATLAB code used to generate results and images. NOTE: MATLAB/Python code may be randomly tested by TA for functionality!
- Be sure to comment your MATLAB/Python Scripts thoroughly! We need to understand your approach to each problem.

Please submit the following files in Canvas:

- Your responses to each question in PDF format, titled “[caseID]_HW05.pdf”
Be sure to include all generated images in your PDF report.
- All generated images, MATLAB/Python code in one zip file titled “[caseID]_HW05_Code.zip”

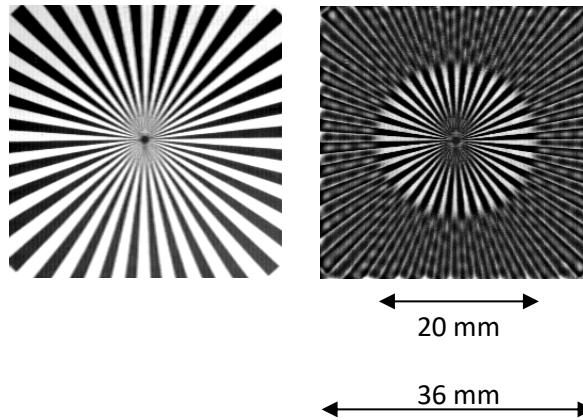
Please read the question descriptions carefully. Be sure to address every question. We try to give you sufficient information to get you successfully started on this simple, but interesting, project.

The following files will be provided by the TA.

- edges.tif
- wheel.tif
- FigP0405(HeadCT_corrupted).tif

Questions

1. The image on the left is filtered in the Fourier domain to obtain the image on the right. The outer data in the image on the right is much depressed, even though there is structure visible. The data in the middle is unaffected. *Note: No digital image processing is required for this problem!*
 - a. Is the filter a high-pass or low-pass filter?
 - b. Sketch a generic plot of the transfer function of this type of filter. *Be sure to label axes & standard units!*
 - c. Determine the cut-off frequency for this filter in 1/mm. Some hints are:
 - i. Determine the circumference at the cutoff in millimeters.
 - ii. Determine the number of cycles.



2. The DFT of a flat disk of amplitude 1 is $F(x,y)=f_1(x,y) + jf_2(x,y)$. The DFT of a flat square of amplitude 2 is $G(x,y)=g_1(x,y) + jg_2(x,y)$. What is the **magnitude** of the DFT of a flat disk of amplitude 4 added to a square of amplitude -2? This is a really simple problem that you will probably see on an exam. Don't try to make it difficult!
3. Using **Bandreject Filtering for Image Denoising**. Image FigP0405 (HeadCT_corrupted).tif is a tomography image of a human head, heavily corrupted by sinusoidal noise in at least two directions.
 - a. Clean up the image using **bandreject filtering**. To simplify your project you may ignore padding in this case. *Hint:* Since the noise is sinusoidal, it will show in the spectrum as impulses. Display the spectrum as a guide to where to set up the band of your filter. If you use function `impixelinfo` to determine the coordinates of the impulses interactively, keep in mind that this function lists the column coordinates (v) first and the row coordinates (u) second.
 - b. Use bandpass filtering to extract the noise pattern.
4. We will **now investigate low-pass filtering** in the frequency domain using the images "wheel.tif" and "edges.tif". For **both** images do the steps below.
 - a. Apply an ideal low-pass filter at a frequency which is 1/3 the maximum frequency.
 - b. Apply a low-pass Gaussian filter with a width at half maximum amplitude which is 1/4 the maximum frequency. *Recall the equation for a 2D Gaussian filter you had to use in Hw2 Prob 4a.*
 - c. Print out these images for report
 - i. The 2 original images before filtering in Cartesian domain
 - ii. The 2 original images before filtering in frequency domain
 - iii. The 2 images after ideal low-pass filtering in frequency domain
 - iv. The 2 images after ideal low-pass filtering in Cartesian domain
 - v. The 2 images after Gaussian low-pass filtering in frequency domain
 - vi. The 2 images after Gaussian low-pass filtering in Cartesian domain
 - vii. Create an image of the ideal low-pass filter in the space domain
 - viii. Create an image of the Gaussian low pass filter in the space domain
 - d. Compare the two filters and explain results: *Which yields less ringing artifacts? Which yields better smoothing/blurring?*

Functions of particular interest

MATLAB Functions:

fft2(), fftshift(), ifft2(), ifftshift()

Python Functions:

numpy.fft.fft2(), numpy.fft.fftshift(), numpy.fft.ifft2(), numpy.fft.ifftshift()