Project # 1: Spatial and Frequency Filtering
CSC 391: Introduction to Computer Vision

Instructor: V. Paúl Pauca

due date: Wednesday, February 6th

---

1 Instructions

This assignment contains three main parts: spatial filtering, frequency analysis, and frequency filtering. It is designed for you to explore and gain deeper understanding of the underlying concepts in these topics by implementing code, experimenting on real images, and making observations and drawing conclusions about your results. For each of the three parts, you will need to write a Python script and then use this code to on the provided image data (see section below). Write down your experimentation and observations in a Google Doc, Word, or LaTeX document and include images with your explanations.

2 Image Data

We will use several images in this assignment which you can find in:
`http://csweb.cs.wfu.edu/~pauca/csc391-CV/ImageData`
Notice that images containing additional noise have suffix "noisy" as part of the file name.
You will find the following images:

- The puppy picture, original and noisy
- Drone images of vegetation and human activity, Madre de Dios, Peru.
- Underwater images of wildlife in the Lighthouse reef, Belize.
- Camera-trap images of wildlife from the Serengeti National Park, Tanzania.

3 Spatial Filtering

3.1 Implementation

For this part of the assignment, you need to implement a Python script that applies a spatial filter to an image and displays the results. More specifically, your program should:

- Allow you to define a square filter of any size $k \times k$.
- Allow you to apply the filter to any color image of size $m \times n$ with the resulting image also being in color.
- Display the original image, the filter, and filtered image.
- Save the images to JPG files.

3.2 Smoothing, Denoising, and Edge Detection Filters

3.2.1 Smoothing and denoising

Experiment with Gaussian and median filters of various sizes, .e.g 3×3, 9×9, 27×27 on the puppy noisy image data. Notice there is a tradeoff between denoising and resolution for both cases. Draw observations based on your experimentation that compare and contrast the performance of both filters on this tradeoff.

3.2.2 Edge detection

Edge detection is another type of operation of high importance in image processing and computer vision applications. Experiment with Cany edge detectors on the puppy image data, both original and noisy, and make observations on their performance on these datasets. Apply edge detection to one of the other datasets and explore its performance (i.e. field images such as camera-trap or underwater imagery contain larger amounts of noise). Do you think there might be better ways to extract edges from noisy data?

4 Frequency Analysis

4.1 Implementation

Implement a Python script that applies the 2-D DFT to image data and displays the resulting coefficients in the Fourier domain. Specifically, your program should:

- Allow you to work with any grayscale image of size $m \times n$. I.e. if the image is in color then transform it to grayscale before processing.
- Allow you to visualize the magnitude of the Fourier coefficients, resulting from the 2-D DFT of the image, as 3-D plots or 2-D images, with the low frequencies in the center of the plot. Plot also the log of the magnitude+1 to better observe the contribution of mid and high frequencies to your image.
- Save the plots in JPG files.

4.2 Frequency Analysis

Use your Python script to explore the presence and contribution of different spatial frequencies in a subsample of the image data (you choose the subsample). Some of the things to explore and ponder about when visualizing the magnitude of the Fourier coefficients include:

- Is there any kind of correlation between the image content and the decay of the magnitude of the Fourier coefficients? For example, plot the Fourier coefficients along the x-axis (y = 0) for various images, superimpose the plots and compare.
- For the same image in the dataset (original and noisy versions), can you see the effect of the noise on the magnitude of the Fourier coefficients?
- Some log plots of the magnitude of the Fourier coefficients (better seen as images) show higher (brighter) values around the axis, away from the center of the image. Why is that?
- What is the effect of removing (zero-ing out) Fourier coefficients on the image reconstructed using the inverse 2-D DFT? Observe what happens when Fourier coefficients in the center of the magnitude plot around $F(0, 0)$ (i.e. the low frequencies) are zero-ed out and compare to what happens when the Fourier coefficients away from the center (i.e. high frequencies) are zero-ed out. Observe what you get and draw conclusions as to how this might be useful for smoothing or edge detection.

5 Frequency Filtering

5.1 Implementation

Implement a Python script that applies Butterworth low-pass and high-pass frequency filtering. Specifically, your program should:

- Allow you to work with any grayscale image of size $m \times n$. I.e. if the image is in color then transform it to grayscale before processing.
- Allow you to compute the 2-D DFT of the image

- Allow you to dampen the Fourier coefficients using Butterworth's approach so that they decay quickly but smoothly towards zero around a selected cutoff frequency, instead of sharply like it would happen if Fourier coefficients are just zero-ed out.
- Plot the original and Butterworth-scaled magnitudes of the Fourier coefficients and their resulting images
- Save the plots in JPG files.

5.2 Low-pass and High-pass Filtering

Use your Python script to explore the difference between low-pass by zero-ing out coefficients (ideal low-pass or high-pass filtering) versus Butterworth low-pass and high-pass filtering on a subsample of the image data. Draw observations based on the results you obtain.

6 Expectations and Grading

For this assignment, I ask you to submit your Python scripts as well as a report in a Google Doc, Word or LaTeX file. Be sure to organize your work to show your experiments with the data, the results you obtain, and the observations and conclusions you draw from them.

A good report is one that is well-written and well-organized, showing results, specifying parameters used for these results and drawing important observations about what is happening.

A poor report is one that contains only images, no parameters or explanations about how they were obtained, and no significant observations made on the results obtained.