HOMEWORK 2
FREQUENCY DOMAIN PROCESSING

CSED551 – COMPUTATIONAL PHOTOGRAPHY
FALL 2024
SUNGHYUN CHO (S.CHO@POSTECH.AC.KR)

OVERVIEW

In this homework, you need to implement filtering in the frequency domain:

- Ideal lowpass filter
- Gaussian lowpass filter
- Unsharp masking using the convolution theorem

IDEAL LOWPASS FILTER

Ideal lowpass filter is a filter that removes all the frequency components whose frequencies are larger than a certain threshold. You need to implement the ideal lowpass filter in the frequency domain. Specifically, your program takes an input image and a threshold value that indicates the radius of the lowpass filter in the frequency domain. Then, it outputs a filtered result. You also need to implement boundary handling to avoid color bleeding across the image boundaries. For implementing your program, you can use the example code in the course material. Your program needs to handle RGB images. To process RGB images, you can first split the color channels, and process each color channel separately. One test image will be given. You can use your own additional test images too.

In your report, you are required to:

- Explain how you implemented your program
- Visualize the Fourier transform of an input image, the ideal lowpass filter in the frequency domain, and the filtering result in the frequency domain
- Show the results of different lowpass filter sizes
- Discuss the effect of the boundary handling & discuss how large the padding size should be with respect to the filter size

GAUSSIAN LOWPASS FILTER

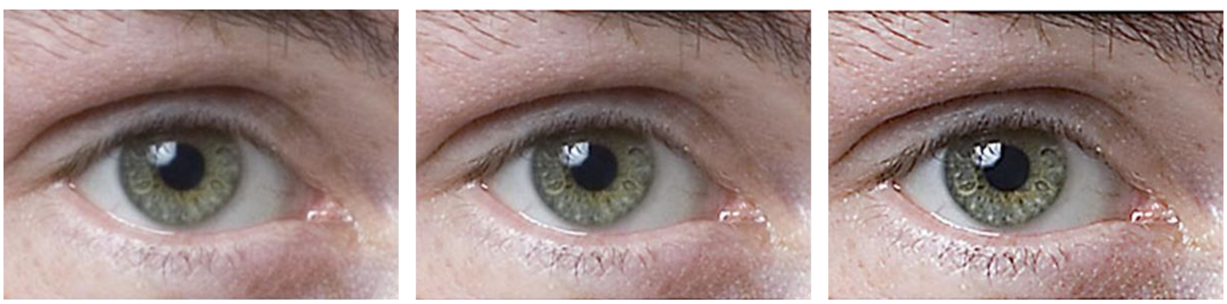
Gaussian lowpass filter is a filter that gradually attenuates high frequency components instead of simple thresholding. You need to implement the Gaussian lowpass filter in the frequency domain. Specifically, your program takes an input image and a threshold value that indicates the filter size, i.e., the standard deviation of the Gaussian function in the frequency domain. Then, it outputs a filtered result. You also need to implement boundary handling to avoid color bleeding across the image boundaries. For implementing your program, you can use the example code in the course material.

In your report, you are required to:

- Explain how you implemented your program
- Visualize the Fourier transform of an input image, the Gaussian lowpass filter in the frequency domain, and the filtering result in the frequency domain
- Show the results of different filter sizes
- Compare the results of the ideal and Gaussian lowpass filters

UNSHARP MASKING & CONVOLUTION THEOREM

Unsharp masking is an image sharpening technique, which was originally developed in 1930s. Originally, unsharp masking was done using glass plates and films. Nowadays, it is provided by many image editing applications such as Adobe Photoshop. For more details, you can refer to: https://en.wikipedia.org/wiki/Unsharp_masking.



Input image | Unsharp masking (small α) | Unsharp masking (large α)

Unsharp masking can be formulated as:

$$Y = X + \alpha(X - G * X)$$

where X, Y and G are an input image, its sharpened result, and a low-pass filter, respectively. G is typically a Gaussian filter. α is a parameter to control the sharpening strength. The size and shape of G are controlled by the standard deviation parameter σ.

In this homework, you need to implement two versions of unsharp masking.

- Use spatial domain to implement unsharp masking

- Use the frequency domain (use FFTs)
  - You need to implement all operations including convolution, addition, subtraction, and scaling in the frequency domain.

In your report, you are required to:

- Explain how you implemented your code
- Show input images, parameters, and their corresponding outputs
- Present a detailed discussion on the effects of the parameters, i.e., show how results change as the parameters change.
- Present a detailed discussion on the results of each step, e.g., the FFT results of images and convolution kernels.
- A comparison between the spatial-domain and frequency-domain implementations including their results and computation times with respect to different parameter values.

You can use your own test images.

REQUIREMENTS

You must upload a single zip file that contains the following to the LMS:

- code/ - a directory containing all your code for this assignment
- images/ - a directory containing your input images and their results
- report.pdf – your report as a PDF file

You can use any programming language for your implementation. I recommend you use one of the followings:

- C++ and OpenCV
  - OpenCV is a powerful computer vision library that provides many useful features.
- Matlab
  - You will need to install Image Processing Toolbox
- Python
  - There are several useful libraries, e.g., a python binding of OpenCV, scipy, numpy, …

Due: Oct. 20^{th}, 23:59

Penalty for late submission

- 1 day: 70%
- 2 days: 30%
- 3 days: 0%

EVALUATION

- 20 points for the ideal lowpass filtering
- 20 points for the Gaussian lowpass filtering
- 40 points for the unsharp masking (You must implement two versions. 20 points for each.)
- 20 points for the report

Your homework will be scored based on your report, and I am not going to compile or run your code. Thus, your report must include all necessary details of your implementation and results.