

# HOMework 2

## IMAGE FILTERING

CSED551 – COMPUTATIONAL PHOTOGRAPHY  
FALL 2022

SUNGHYUN CHO ([S.CHO@POSTECH.AC.KR](mailto:S.CHO@POSTECH.AC.KR))

### OVERVIEW

In this homework, you need to implement uniform and non-uniform image filters:

- Unsharp masking
  - Using spatial domain filtering
  - Using frequency domain filtering
- Edge-aware filtering
  - Bilateral filtering

### UNSHARP MASKING

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](https://en.wikipedia.org/wiki/Unsharp_masking).



Input image



Unsharp masking  
(small  $\alpha$ )



Unsharp masking  
(large  $\alpha$ )

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.  $\alpha$  is a parameter to control the sharpening strength. The size and shape of  $G$  are controlled by the standard deviation parameter  $\sigma$ .

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 can implement all operations including convolution, addition, subtraction, and scaling in the frequency domain.

In your report, you are required to include:

- A detailed discussion on the effects of the parameters, i.e., show how results change as the parameters change.
- 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.

You can use your own test images.

## EDGE-AWARE IMAGE FILTERING

Low-pass filters such as Gaussian filters are often used for removing high-frequency details such as small-scale textures and noise from an image. However, such filters blur entire images, so other structural edges get blurred too. Often, we want to blur out small-scale details and noise while keeping important edges sharp. To this end, many edge-aware filtering methods have been proposed. Bilateral filtering is one of the most popular methods among them. Bilateral filtering was first introduced by Tomasi and Manduchi in 1998, and quickly became one of the most widely used tools thanks to its effectiveness. Lots of applications and variants have been proposed as well. More details can be found in the lecture slides, and in the paper below.

- Tomasi and Manduchi, Bilateral filtering for gray and color images, ICCV 1998

In this homework, your goal is to implement the bilateral filter.

In your report, you need to discuss the effects of the parameters, i.e., show how results change as the parameters change.

You also need to compare the results of the bilateral filter and a Gaussian filter with sample images.

You can use your own test images.

## REQUIREMENTS

Besides the requirements mentioned above, your report must:

- Explain how you implemented the algorithms and your code.
- Show input images, parameters, and their corresponding results.
- Provide a brief analysis on different parameters of each algorithm, i.e., show how results change as parameters change.

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. 30<sup>th</sup>, 23:59

Penalty for late submission

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

## RUBRIC

- 40 points for the unsharp masking (You must implement two versions. 20 points for each.)
- 40 points for the bilateral filtering.
- 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.