HOMEWORK 1
IMAGE FILTERING

CSED551 – COMPUTATIONAL PHOTOGRAPHY
SPRING 2025

SUNGHYUN CHO (S.CHO@POSTECH.AC.KR)

OVERVIEW

In this homework, you need to implement basic image filtering operations:

- Gaussian filters
  - Different kernel sizes
  - Different boundary handling schemes
  - Separable filters
- Histogram equalization

GAUSSIAN FILTERING

Gaussian filters are the most representative low-pass filters. In this homework, you need to implement different versions of Gaussian filtering operations as specified below.

- The function prototype of your Gaussian filtering function must be as follows:
  - `def filterGaussian(image, kernel_size, kernel_sigma, border_type, separable)`
    - Input parameters:
      - `image`: input RGB image
      - `kernel_size`: an odd integer to specify the kernel size. If this is 5, then the actual kernel size is 5×5.
      - `kernel_sigma`: a positive real value to control the shape of the filter kernel.
      - `border_type`: extrapolation method for handling image boundaries. Possible values are:
        - `cv2.BORDER_CONSTANT`
        - `cv2.BORDER_REPLICATE`

        - `cv2.BORDER_REFLECT`
        - `cv2.BORDER_WRAP`
        - `cv2.BORDER_REFLECT_101`
        - For details on the parameter values above, refer to OpenCV: Operations on arrays.
    - `separable`: Boolean value. If `separable` == `true`, then the function performs Gaussian filtering using two 1D filters. Otherwise, the function performs Gaussian filtering using a normal 2D convolution operation.
- Output:
    - The function must return a filtered RGB image.

- Do not use the convolution functions provided by the image processing libraries such as OpenCV. You need to implement your own code to sample filter kernel elements, and to perform convolution operations. You can use other functions provided by the image processing libraries such as OpenCV's copyMakeBorder.
- Use your own test images and report their results in your report.
- Report the computation times of `filterGaussian()` with respect to different values for the parameters `kernel_size` and `separable`. Then, discuss how the parameters affect the computation time in your report.
- Visualize the filter kernels of different parameter values for `kernel_size` and `sigma` in your report. Also, discuss how `kernel_size` must be set with respect to `sigma`.
- Show the filtering results of different values for `border_type` in your report. Discuss how they are different from each other. Also, explain what extrapolation methods you think are the best and worst, and why.

HISTOGRAM EQUALIZATION

Histogram equalization is a technique to enhance the brightness and contrast of an image. You need to implement two versions: a grayscale-version, and a color-version. For the color version, you can simply apply the grayscale-version to each color channel independently.

A few grayscale and color images are provided as test examples. You need to include their results in your report. You SHOULD also find your own example images and include their results in your report.

Many image processing and computer vision libraries including OpenCV provide histogram equalization. However, in this assignment, you MUST NOT use such functions, but implement it on your own.

In your report, you should discuss how the histogram equalization changes the brightness and colors of input images, any limitations or artifacts, and how to improve the method.

ADDITIONAL 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.

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:

- Python
  - There are several useful libraries, e.g., a python binding of OpenCV, scipy, numpy, …
- C++ and OpenCV
  - OpenCV is a powerful computer vision library that provides many useful features.

Due: March 16th, 23:59

Penalty for late submission

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

EVALUATION

- 40 points for the Gaussian filtering
- 40 points for the histogram equalization
- 20 points for the report

Your homework will be evaluated based on your report, and the TA does not compile or run your code. Thus, your report must include all necessary details of your implementation and results.