Lab 5 – Image Restoration

Goal: Introduction to image restoration.

Introduction

In practical imaging systems the acquired image often suffers from effects of blurring and noise. Image restoration algorithms are aimed to restore the original undistorted image from its blurry and noisy version. The lab experiment demonstrates the evolution of restoration algorithms.

Salt and pepper noise:

We describe salt and pepper noise with the parameter p. The noise is defined on a per-pixel basis in the image; for each independent pixel in the image, exactly one of the following occurs:

1. pixel value changes to 255 with probability $\frac{p}{2}$
2. pixel value changes to 0 with probability $\frac{p}{2}$
3. pixel keeps its original value with probability $1-p$

In this experiment, we will try to deal with salt and pepper noise!

Blurring kernel:

In this scenario, the original image gets blurred by a kernel with DFT coefficient $H(u,v)$. The image may also suffer from additive noise with spectral component $S_{nn}(u,v)$. In this experiment, we will try to reconstruct the original image from a blurry and noisy image using several different approaches:

1. Simple Inverse Filter

$$H^{I}(u,v)=\frac{1}{H(u,v)},$$

2. Pseudo Inverse Filter

$$H^{PI}(u,v)=\begin{cases}\dfrac{1}{H(u,v)}, & H(u,v)>\varepsilon\\ 0, & H(u,v)\le\varepsilon\end{cases}$$

3. Wiener Filter

$$H^{WI}(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_{nn}(u,v)}{S_{ff}(u,v)}},$$

where $S_{ff}(u,v)$ is the spectral density of the original signal and $S_{nn}(u,v)$ is the spectral density of the noise.

Now let's assume that the noise is white, that is, it has a constant spectral density for all spatial frequencies:

$$S_{nn}(u,v) = S_n^2,$$

Where S_n^2 is a constant.

Furthermore, let's assume that the spectral density of the original signal is in inverse proportion to the square of the spatial frequency (the squared distance from an origin in the frequency space):

$$S_{ff}(u,v) \propto \frac{1}{u^2 + v^2},$$

that is,

$$S_{ff}(u,v) = \frac{1}{k} \cdot \frac{1}{u^2 + v^2},$$

where k is a constant.

Substituting these two assumptions into the general formula of the Wiener Filter, we obtain:

$$H^{WI}(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_{nn}(u,v)}{S_{ff}(u,v)}} = \frac{H^*(u,v)}{|H(u,v)|^2 + kS_n^2(u^2 + v^2)}.$$

In the Preliminary report you will be asked to prove one more formula regarding Wiener Filter.

Preliminary report

Prepare the following tasks and python files and bring them to the lab:

1. Read the introduction section in this file.
2. In a Colab python notebook, implement the image acquisition and degradation **by salt and pepper noise with parameter p.** You are only allowed to use the NumPy library (with no loops!) in order to add the noise. You should make a **brute force implementation.** (Hint: you may want to create a 3-level gray mask, where each level symbolizes one of the salt and pepper noise options that were described in the introduction. Create a random i.i.d. array from a uniform distribution and then quantize it in a clever way. Finally, apply the mask to the image)

   Load a **uint8 grayscale image** and check your implementation with p = 5%. Additionally, print a small patch (crop) from the image such that the noise can be clearly seen. You are allowed to use non-NumPy libraries for loading and printing the image.
3. Define a 1D and 2D median filter. How does a median filter work? Is it a linear operator? Is it a shift invariant operator? Prove your answer.
4. Consider the 1D discrete step function:

$$step[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

   Apply two different operators on the signal by **manual calculation** (no software tools):

   a. LSI mask (convolution) with "Gaussian" kernel $(0.25 \quad 0.5 \quad 0.25)$
   b. Median filter 1D of size 3

   You may solve the exercise in the space domain (not the frequency/Z domain) using a graphical method (moving window). Attach an **handwritten** sketch of the 3 signals (input and 2 outputs) for n values between [-5, 5]. Using your results, explain the **edge conservation property** of the median filter on 2D images.
5. Read about the python method **scipy.signal.medfilt2d**. Explain what it does, giving a short explanation of the method's parameters. Explain the meaning of kernel_size parameter. What is a valid value for kernel_size? (Hint: it is not just any integer)

6. Derive the following formula for Wiener filter:

$$H^{WI}(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + kS_n^2(u^2+v^2)} = \frac{H^*(u,v)}{|H(u,v)|^2 + \alpha\sigma_n^2(u^2+v^2)},$$

where σ_n^2 is the variance of the noise and α is a constant. Derive a relation between the constants k and α.

7. Implement in python notebook (in Colab) the image acquisition and degradation process by **blurring kernel and additive noise**:
    a. Read **grayscale** image from file as **float64**.
    b. Blur the image using an averaged filter 5X5 (use *cv2.filter2D*).
    c. Add Gaussian noise to the blurred image (use *skimage.util.random_noise*).

    Plot the original image and the resulting images of the steps (with titles).
8. Attach the Colab outputs to the preliminary report. It should include the code of all the functions and images (plots): input; salt and pepper noisy; blurred; blurred and noisy.

    (It might be 3-5 pages. Do not add irrelevant and unwanted content!)

Description of the experiment

Open the notebook in colab.

Open the python functions file that you prepared at home.

Follow the instructions in the 5 sections of the lab:

Part 1 – Median Filter

Part 2 - Inverse Filter

Part 3 – Pseudo Inverse Filter

Part 4 – Wiener Filter

Part 5 – Deep learning (DnCNN)

Part 6 – Comparison

Final report

Submit the results of testing and demonstrations from the 5 parts of the experiment, with the image of your choice. Explain your results in each step.

Answer all the questions appeared in the final report notebook at the end of each section,

***Pay attention to all submission guidelines in the lab1 manual file.**

Submit a single PDF file with all your answers.

Verify that the PDF contains all your content.

References

1. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.medfilt2d.html
2. https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d
3. R. C. Gonzalez and R. E. Woods, Digital Image Processing. Prentice-Hall, Inc., 2002, Second Edition (Library Dewey number 621.368 GON).
4. R. C. Gonzalez and R. E. Woods, Digital Image Processing. Addison-Wesley Publishing Company, Inc., 1992 (Library Dewey number 621.368 GON).
5. https://en.wikipedia.org/wiki/Structural_similarity