EQ2330 – Image and Video Processing

Project 1: Image Enhancement

due Nov. 28, 11:59 pm

Students should submit their project reports electronically. Additional files are available on canvas which contains LaTeX template for the report, images and matlab files. Please upload your document in PDF format to canvas before submission deadline. Note that unlike the assignments, the project is graded. A template for the report is available as additional files in canvas. Include the "relevant" parts of the code in the report in a clear and concise manner as shown in the report.

Note that the even though the primary focus of the grading will be the report, the code will be checked if necessary to corroborate the results. Thus your submission should also include relevant Matlab source codes. The report should be named *projectX_groupY.pdf* and only one submission per group is allowed. Follow the below formatting for the source code submission which should be a zip file named *projectX_groupY.zip*. If the source code spans over multiple files and directory structures, the code that needs to be run should be named as *main*, should be placed in the parent directory. You could choose to place it in the one directory deeper with name 1,2,3... if you want to separate subquestions of the project.

1 Introduction

The goal of this project is to investigate spatial and frequency domain image enhancement techniques. Completion of all tasks in the project description is required to pass the project. Use the "Lena" image 512×512 to demonstrate your results in the report.

2 Spatial domain processing

2.1 Histogram equalization

This task is about the global histogram equalization.

- Plot the histogram (with 8 bits resolution) of the input image using the `hist` command.
  Hint: use `im(:)` to create an one dimensional vector.

- Simulate a low-contrast image by reducing the dynamic range of the image, e.g.:

$$g(x, y) = \min(\max(\lfloor a \cdot f(x, y) + b \rceil, 0), 255),$$

  where $\lfloor \cdot \rceil$ denotes the round operator, $0 < a < 1$, $0 < b < 255(1 - a)$. Use $a = 0.2$, $b = 50$. Plot the histogram again and comment on the difference.
  Hint: To view the image of lower constrast, you may need to specify the interval for `imagesc`, e.g., `imagesc(im, [0 255]);`.

- Implement the histogram equalization algorithm from section 3.3.1 in [1]. Apply your algorithm to the contrast-reduced image. Plot the histogram of the enhanced image.
  Hint: Use `cumsum` to calculate CMF.

- *Question:* Why is the histogram not flat after the equalization?

2.2 Image denoising

This task is to investigate the denoising effect of spatial smoothing filters and order- statistics filters.

- Use the provided `mynoisegen(·)` to generate Gaussian (zero-mean and variance 64) and salt & pepper noises ($p(0) = p(255) = .05$), and apply the noises to the input image. The two noise types should be investigated separately. The Gaussian noise is additive, and for the salt & pepper noise, use

```
im_saltp = im;
n = mynoisegen(saltpepper, 512, 512, .05, .05);
im_saltp(n==0) = 0;
im_saltp(n==1) = 255;
```

  Use `help mynoisegen` for more information. Plot the histograms of the noisy images and compare to the clean image.

- Implement the 3×3 mean filter and apply to the noisy images. Plot and compare the histograms.
  Hint: Use the matlab function `conv2(·)`.

- Implement the 3×3 median filter and apply to the noisy image. Again, plot and compare the histograms. Note that `conv2(·)` cannot be used here anymore.

- *Question:* Explain the difference between the mean filter and the median filter and their denoising effects on different noise types.

3 Frequency domain filtering

The last task is to investigate the frequency domain filtering. The frequency domain filtering is particularly interesting for the deblurring application, since the convolution corresponds to the multiplication in the frequency domain. The goal of this task is, therefore, to implement a deblurring algorithm which is capable of restoring an out-of-focus image. Before you start working on the task, you should read section 4.6.3 in [1](also [2]) for some common pitfalls in the implementation of the frequency domain filtering.

- The blurred image can be generated using the degradation model

  $$g(x, y) = h(x, y) * f(x, y) + \eta(x, y),$$

  where the notation and more detailed explanation can be found on page 221 in [1] (and 312 in [2]). The noise term $\eta(x, y)$ is considered to be the quantization noise introduced by representing (quantizing) the out-of-focus image using the 8-bit resolution,

  $$g(x, y) = \min(\max(\lfloor h(x, y) * f(x, y) \rceil, 0), 255).$$

  Plot and compare the Fourier spectra of the image before and after the degradation. *Hint:* The provided `h = myblurgen('gaussian', r)` can be used to generate `h(x, y)`. For the report, use the Gaussian blur kernel with radius `r = 8`. The blurred image should be generated using `conv2(im, h, same)` to ensure that the size of the image preserves.

- Use `fftshift(·)` to center the spectra. To be specific, the Fourier spectra means the log of the amplitude of the 2-dimensional FFT of the image.

- Design an image deblurring algorithm to enhance the distorted image. The function should take the degraded image $g(x, y)$, the blur function $h(x, y)$ and the noise variance $Var(\eta)$ as the input parameters. Be prepared to demonstrate your function using other input images.
  Hint: The enhancement algorithm may assume that the variance of the noise is known, but not the actual realization of the noise (or its spectrum). The noise variance can be evaluated using the difference image between the out-of-focus image and the quantized image.
  Hint: Two useful algorithms are described in section 5.8-5.9 in [1] [2]
  Hint&question Be careful with the implied periodicity of the DFT, which may cause problems when the input image has sharp edges on the boundaries. Your algorithm may not work if this is not properly treated. Explain why?
  Hint Two out-of-focus images are provided for testing purposes. The images are generated using the Gaussian kernel with $r = 8$, and the noise variance is 0.0833.

References

[1] R. C. Gonzales and R.E. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, New Jersey, second edition, 2002.

[2] R. C. Gonzales and R.E. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, New Jersey, third edition, 2008.