EE/CS: Image Processing Fall 2020
Instructor: Prof. Zeyun Yu

Term Project

Assigned: November 19, 2020 (Thursday)
Deadline: December 11, 2020 (Friday)

General Guidelines:

1. You are encouraged to work with another student on a problem of your preference. If your team consists of three members or more, you will need to justify for the workload of your project and define each member's role in the project. If you decide to work individually on your project, you do not necessarily get any extra credit for working alone.
2. You may choose a topic from the list below or any other image-related problems.
3. Your final score will be given based on the **difficulty** (scale 0−10) of the problem you choose and the **performance** (scale 0−10) of your final project.
4. Please choose the topic and form a team as early as you can. The deadline for project proposal (a few sentences by email to me would be sufficient) is November 26th (Thursday). If you choose to work on your own topic, please consult with me for the difficulty scale of your topic.
5. Things to submit:
    a. Project slides (due on CANVAS, December 11th): used as your final project report. Your slides should include front page (title & team members), problem statement, methods, results/demo, and conclusion. Each team member's role should be clearly defined in the slides.
    b. Source code (due on CANVAS, December 11th): as always, please include a ReadMe file.
6. Final project demonstration:
    a. Each team will make a presentation in class on December 10th. Each team member should present his/her contributions of the project.
7. Materials provided on CANVAS:
    a. This project description
    b. Some sample project ideas from previous years
    c. Some sample images you may use in your projects

Some Project Ideas (all with difficulty scale of 10):
Note: If you need any image as input to test your project, please let me know.

1. **Local contrast enhancement**

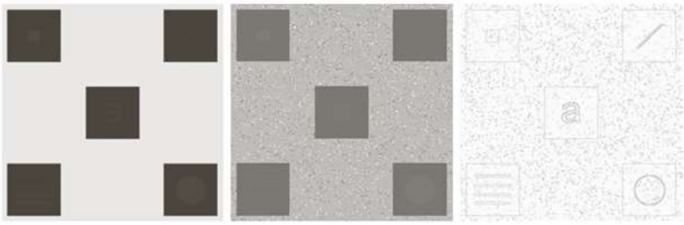
   The purpose of this project is to use image statistics for local contrast enhancement with the following scheme (Lecture 6):

$$g(i,j)=\begin{cases} E\cdot f(i,j) & if \quad m'_{ij}\le k_0 m_G \quad \& \quad k_1\sigma_G\le\sigma'_{ij}\le k_2\sigma_G \\ f(i,j) & else \end{cases}$$

   Please compare with the global histogram equalization method, meaning that you will need to implement the global histogram equalization as well.

2. **Local histogram equalization**

   Implement the local histogram equalization scheme on an image:



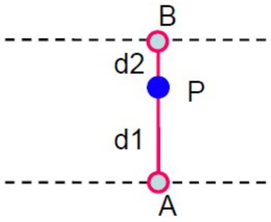
   Please compare with the global histogram equalization method, meaning that you will need to implement the global histogram equalization as well.

3. **Image interpolation**

   Given two images of the same size, interpolate a new image of the same size between the two given images. A simple way is by linear interpolation as follows:

$$I_P=\frac{d_2}{d_1+d_2}I_A+\frac{d_1}{d_1+d_2}I_B$$

Linear interpolation



   In this project, I'd like to see an improved version by considering a neighborhood of A and B and find a better way to calculate the intensity value for P. One possibility is to use a similar idea to the bilateral filtering: the weight of A and B is high if and only if their intensity difference is small.

4. **Shift-detection using image correlation**

   Given two images f_1 and f_2:

$$f_2(x,y)=f_1(x-x_0,y-y_0)$$

Their corresponding Fourier transforms F_1 and F_2 will be related by

$$F_2(\xi,\eta) = e^{-j2\pi(\xi x_0+\eta y_0)} * F_1(\xi,\eta).$$

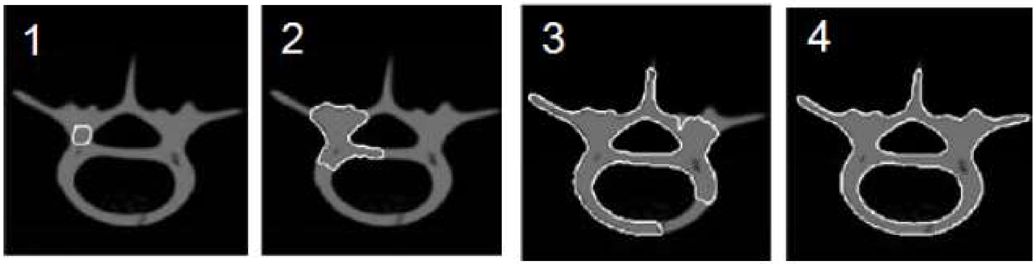
The cross-power spectrum of two images f and f' with Fourier transforms F and F' is defined as

$$\frac{F(\xi,\eta)F'^{*}(\xi,\eta)}{|F(\xi,\eta)F'(\xi,\eta)|} = e^{j2\pi(\xi x_0+\eta y_0)}$$

By taking inverse Fourier transform of the representation in the frequency domain, we will have a function that is an impulse; that is, it is approximately zero everywhere except at the displacement that is needed to optimally register the two images.

5. **Region-growing image segmentation**

   Implement a region-growing method to segment a feature of interest. The region-growing starts from a user-specified seed, which grows based on a 4- or 8-neighborhood. The stopping criterion can be simply a threshold of the image intensity (or gradient magnitude) value.



   Note: If you can come up with a smarter stopping criterion for the curve evolution, you will get 10 points for the difficulty scale.

6. **Watershed-based image segmentation**

   Implement the watershed-based image segmentation approach to segment a feature of interest. You may use the original image intensities or the gradient magnitude map as the basis for the watershed method.

7. **Local Otsu's method for image segmentation**

   Implement the local Otsu's method by partitioning the image into a number of sub-regions and applying the Otsu's method in each region individually. You should test your code on the noisy, shaded image shown on Slide 19 of Lecture 21.
   Please also compare your local method with the global Otsu's method.

8. **Motion de-blurring in frequency domains**

   Implement the Wiener filtering to de-blur a corrupted image that was caused by a motion of camera. You may refer to Lecture 9 for details of how to generate and recover an motion-induced blurred image. The motion can be assumed to be linear as discussed in Lecture 9, and some level of noise (such as Gaussian noise) should be added to the motion-blurred image as the input to your program.

9. **Model-based de-blurring in frequency domains**

Implement the Wiener filtering to de-blur a corrupted image that was caused by the filter defined as:

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}}$$

You may refer to Lecture 9 for details of how to generate and recover such images. Some level of noise (such as Gaussian noise) should be added to the blurred image as the input to your program.

10. **Reducing periodic noise in frequency domains**

Implement a method to reduce periodic noise in an image by using the band-reject filter in the frequency domain. In this project, you need to first generate a pattern for periodic noise and add the noise to a noise-free image, and then you implement the band-reject filter to remove noise. The pattern for periodic noise can be in one of several forms: such as a pattern like 1-0-1-0-1-0...., or a patter defined with a sine/cosine function.