Assignment 2

General Info

- The work can be done anywhere where MATLAB and the related toolboxes are available.
- A **written report** is required. The report is free form but should include results, figures, and any code you wrote, as well as discussions of what you observe.
- Reports should be submitted as an electronic copy in **PDF** format on **Canvas** before the due date.
- Late submissions will not be accepted.
- Better documentation and clearer discussions of your work improves our ability to fairly mark your report. Make sure your report is well structured, organized, and clear. Your report has to follow the order of questions. Give all relevant code right before the results and discussion of each part, not in an appendix.
- This assignment has 6 questions.

1. Color Images: Display, Colormaps and Histogram

Here you will learn about the different colormaps used to display color images and how to generate image histograms. Run the following MATLAB code:

```
clear all; close all;
load('flujet','X','map');
imagesc(X);colormap(map);
axis off
```

(a) Show the image displayed in the resulting figure.

(b) What is the size of the matrices X and map?

(c) Explain what X and map contain and what they represent.

(d) Change the map matrix such that the color of the background becomes black. Display the new image with your new colormap.

(e) Write your own MATLAB function that computes the histogram of a color image:

hist = colorhist(image, n1, n2, n3)

where, “hist” is a one-dimensional vector and n1, n2, n3 denote the number of bins for the R, G, B channel, respectively. The algorithm should scan the R channel first, then the G channel and finally the B channel. Then, concatenate the histograms of each color channel into the resulting vector, “hist”.

(f) Perform your “colorhist” function on the images *TheCourtesan.bmp* and *TheHagueSchool.bmp*. Is it possible to discriminate these two images from each other based on their color histograms?

2. Homomorphic Filtering

Homomorphic filtering is a technique for separating multiplicative components of a signal with the help of the log transform. In the image domain, the log transform may be separated into low-frequencies (representing illumination) and high-frequencies (reflectance in the scene).

(a) Run the file 'Q2_student.m'. Explore all the functions that are being called and explain, in detail, all the parameters that are being passed into 'homomorphic.m'.

(b) Use the code to extract all the low-frequency illumination from the *in_the_tunnel.png* image.

(c) Explain how the 'highboost.m' filter operates.

3. Estimation by Mathematical Modelling

Here you will explore deriving a model of motion blur and understand its effects on images.

(a) Open the file 'Q3_student.m' and run it in MATLAB. Briefly describe the code and what it does.

(b) Change the parameter 'st' to 0.001 and 0.1 to add noise to the degraded signal. How does this affect the reconstruction? Why?

(c) Set the parameter 'USE_PSUEDO' to 1 and run the code with the 'st' parameters 0, 0.001, and 0.1. What is happening? Explain the results.

(d) Suppose you want to develop an app for your iPhone that uses the information from the phone's accelerometer to remove blurring caused by camera motion. For the sake of simplicity, assume that the acceleration is constant and only in the *x* and *y* direction. Extend 'Q3_student.m' to estimate motion in the *x* and *y* directions. Then, try to remove the motion blur from *'blur.bmp'* using the following parameters a_x=30 a_y=40. To achieve this, use the reconstruction portion of the code, read *'blur.bmp'* as a double, take its Fourier transform, and pass it into the reconstruction portion as I_motion_fn. (Hint: use the pseudo filter, and play with the threshold to achieve a good result)

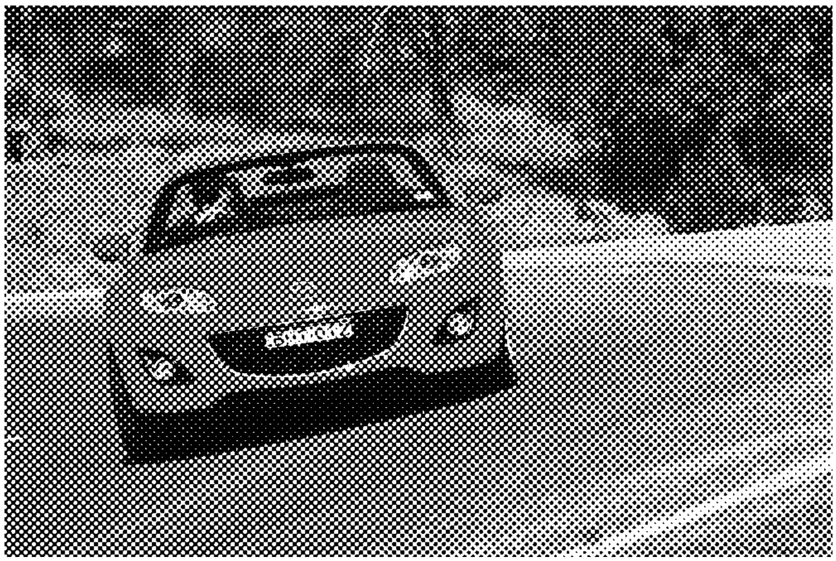
(e) Replace the pseudo-inverse filter with the Wiener filter that was derived in class (implement your own, do not use deconvwnr). Repeat part (c) with the new filter. Play with the Noise-to-

Signal ratio of the Wiener filter and find an optimal parameter for each noise case. Explain your observations.

(f) Use the Wiener filter to restore *'blur.bmp'* as was done in part (d). Do the results improve? Why or why not?

4. Restoration of Halftone Images

Here you will learn about reversing a special type of image degradation. You are an image processing expert helping a local police department. A detective has reopened a cold case and part of the evidence is a newspaper print of a car. You have been asked to do some CSI style investigation to see if you can learn the suspect's license plate number.



Fortunately, you find some code online to deal with this type of 'halftone' noise and decide to use it to restore the image. Unfortunately, the code is not commented and you need to study it to figure out what it does.

(a) Open the file 'Q4_student.m', run it, and play with the parameters. Probe the outputs to figure out the code functionality. Write your comments describing what the code is doing exactly (leave comments after each '%' symbol and submit the code).

(b) The default parameters of the code do not perform too well. You start thinking back to what your EECE 570 professor told you about filter design, and decide to dig deeper into the code. What can you do to make the results clearer? Can you reproduce something as good as or better than *'halftone_fixed.jpg'*? Explain what you did to improve the results. (Hint: look at notch.m)

5. Geometric Image Manipulation

Here you will learn about techniques to geometrically manipulate images.

(a) Open the file ‘Q5_student.m’ and run it in MATLAB. Describe what the code is doing in detail (leave comments after each ‘%’ symbol and submit the code).

(b) This code was used to degrade ‘swirled.bmp’ modify the code to “un-swirl” the image.

6. Removing Occlusions from Video

In this question, we will be exploring different ways of removing noisy objects from a video sequence with a still background.

(a) Run Q6_student.m and briefly explain what the “accumulate difference” and “reconstruct the background image” sections are doing.

(b) Implement two new reconstruction techniques by computing the mean and the median in the temporal direction of the video. Measure the mean absolute error between the original image and the three reconstructed images. Plot the absolute difference in each case.

(c) Change the ‘nz’ parameter in the code to 0.05 and describe its effect on the video. Repeat step (b) and comment on your observations.

(d) To improve the ‘accumulated difference’ method in the presence of noise, change the ‘T’ parameter and comment on what it does.

<u>End of assignment 2</u>