CSE 4084 Multimedia Systems

HOMEWORK 3

For the problems below please prepare a report that contains all the explanations in every step and any intermediate results including images. Place your code at the end of the report.

Problem 1

This problem is about inverse filtering. You should review the corresponding slides to refresh your memory before attempting this problem. To help you understand how inverse filter is implemented and applied a MATLAB script has been provided (inverse\_filtering.m).

Once you open the script using MATLAB, you will see on Line 8 the statement “T=1e-1”. This defines the threshold value used in the inverse filter. The script simulates the blur due to motion and applies inverse filtering for its removal.

Use the image “original\_cameraman.jpg” for this problem and try different values of the threshold and see how it effects the performance of the inverse filter. Find the ISNR value when the threshold is set to 0.5 (up to two decimal digits).

Problem 2

In this problem you will implement the Constrained Least Squares (CLS) filter and examine its performance when the regularization parameter is set at different values. The original image (Cameraman256.bmp) and a set of MATLAB files have been provided for this problem (cls\_restoration.m, next2pow.m, wrapper.m). Follow the instructions below to finish this problem.

1. Place the original image and all the provided MATLAB files in the same directory.
2. The file “wrapper.m” is the entry or the “main” code. It loads the original image, applies a motion blur to it, and degrades the image by adding noise. The 17th line in “wrapper.m” sets the value of the regularization parameter “alpha”.
3. The MATLAB file “cls\_restoration.m” has an incomplete implementation of the CLS filter. You need to uncomment line 24 in the “cls\_restoration.m” and complete the implementation of the CLS filter.
4. After you complete the implementation of the CLS filter, you should run “wrapper.m” with different values of alpha: (0.0001, 0.001, 0.1, 1, 10, 100). For each value of alpha, compute the improvement in SNR (ISNR and put your results in a table. Note that the computation of ISNR involves three images: the original iage, the blurred and noisy image, and the restored image.

**Problem 3**

In this problem you will use Accumulative Difference Image (ADI) to calculate the motion of an object. The object is a bright rectangle moving with a constant speed in a dark background.

Your task is to find the speed of the object in the horizontal direction (x direction) and in the vertical direction (y direction), as well as the total space this object occupied while moving. The total space is defined as the total number of pixels that this object occupies at least once during its movement.

Use the code “motion\_ADI.m” fort his problem. The code has detailed comments regarding each functioning part. Basically, the code generates the reference frame and 10 consecutive frames containing the moving object.

All you need to do is decide on the appropriate threshold T in line 23 in the code and implement the three equations for ADI in the lecture notes regarding motion-based segmentation. Starting your code flowing line 37 and finish it before the end of the for-loop. The rest of the code will calculate the speed of the moving object and the total space it occupies for you. Report speed\_X\_Direction and speed\_Y\_Direction.