EE225b – Digital Image Processing
Homework 15 – Phase-only image reconstruction

Overview:

In this assignment, you explore the possibility of reconstructing an image from its Fourier transform phase.

Assignment specifics:

On the class website, there is a file `Phase.dat` which contains samples of the phase information of an image, stored in matrix form. The original image was 96 pixels wide and 64 pixels tall. The phase matrix contains 192 by 128 evenly spaced complex-valued samples of the discrete space Fourier transform (DSFT). The magnitude of each sample is normalized to one.

$$(1,1) \text{ of the matrix corresponds to } \omega_x = 0 \text{ and } \omega_y = 0$$

$$(row, col) \text{ corresponds to } \omega_x = 2\pi(col - 1) / width \text{ and } \omega_y = 2\pi(row - 1) / height$$

Remember, in MATLAB, indices start with 1 and the row value (y) is listed before the column value (x). Implement the phase-only reconstruction algorithm described in Figure 1.30 (see the attached PDF file in the class website), and do your best to reconstruct the original image.

The algorithm in the book is described in terms of the DSFT and continuous ω_x, ω_y. In this lab you are working with samples of the DSFT, and 2-D discrete Fourier transforms (2D-DFT).

The iterative procedure requires an initial guess. How does your initial guess affect the algorithm? What did you choose for an initial guess? Why?

The iterative procedure also requires a termination condition. The algorithm in the book requires the phase of the reconstructed image to be equal to the phase given to you. This is not entirely practical from a implementation point of view. What termination criterion did you choose? Why?

On the class website, there is another file `Magnitude.dat` which contains samples of the DSFT of a different image, with the phase information removed. Modify the iterative procedure to do magnitude-only reconstruction, and test it on this file. How well does it work?

Please submit a written lab writeup in class on the due date. Also please submit all your Matlab `.m` files and a bitmap (`bmp`) of your final reconstructed image, via email to ee225bsp19@gmail.com. Email submissions must be received before class on the due date. There should be an executable Matlab script `Lab1.m` which will generate all your results.

Note:
1. For each problem, you need to:
   a. [1] Email your source code (zip it) to ee225bsp19@gmail.com. If the question asks for any implementations.
      i. Make sure it is executable because I need to run your code to give you a score. Either MATLAB or Python is okay.

b. [2] Submit a single PDF file (not word or other formats) on Gradescope which contains:

    i. your answer for each problem;

    ii. your source code;

    iii. your output image.

Make sure to prepare your solution to each problem on a separate page. On Gradescope, please select and match each page to the corresponding problems. Please do make sure that you have BOTH your output images AND codes included in the Gradescope submission AND email me your executable codes, otherwise you will lose credits.

Here are some helpful Matlab commands:

`X = fft2(x)`	Computes the 2D-DFT of the matrix `x`
`x = ifft2(X)`	Computes the inverse 2D-DFT of the matrix `X`
`I = uint8(x)`	`I` is a matrix of integers ranging from 0..255
`imshow(I)`	Displays `I` as a grayscale image in the current figure
`I = imread('small.bmp', 'bmp')`	Reads the image file small.bmp and stores it in matrix `I`
`imwrite(I, 'result.bmp', 'bmp')`	Writes the matrix I to the image file result.bmp
`stuff = load('Phase.dat', '-mat')`	Loads the contents of `Phase.dat` stored in matrix format into the structure `stuff`
`stuff.ImagePhase`	Access the `ImagePhase` matrix stored in `stuff`