Project 3: Phase Retrieval

Introduction. This project will teach you how to implement the Gerchberg-Saxton phase retrieval algorithm to reconstruct images from their autocorrelations. ***This project is due Wednesday, October 18***.

Data Description. You are given a dataset (P3Data.mat) which contains two variables. The first variable (data_test) contains 2000 64x64 pixel grayscale images of hand-written numbers. The other variable (data_test_corr_2D) is the autocorrelation of the original images. Note that these are 127x127 pixels because the correlation operation expands the image. You will only use the data_test_corr_2D variable in this project. The other dataset is just for you to double check your results. Note that they are different dimensions, so it would be best to just do a qualitative comparison.

Phase Retrieval Approach. Your goal is to follow the instructions in the attached supplementary methods from Katz et al. to implement phase retrieval on the data to get back to the original image. Remember that the Wiener-Khinchin Theorem states:

$$\mathcal{F}\{I \star I\} = \|\mathcal{F}\{I\}\|^2$$

where I is the image and $\star$ is the correlation operator. Thus, you will begin by taking the FFT of the autocorrelation data and taking the square root of the result. Therefore, your starting data is the magnitude in the Fourier domain. Your starting phase can just be random values between 0 and 2π.

Next, carefully follow the algorithm outlined in the Katz paper (see the supplementary materials). If implemented correctly, it should yield robust results relatively quickly. Refer to the Fienup paper as needed for clarification on the methods.

Deliverables. Include your Matlab code and a short write-up addressing the following points:

1) Include representative images of your results.
2) Did the algorithm ever fail? If so, include an image of a failed reconstruction. If you run the algorithm again on the same image. Does it consistently fail?
3) How many iterations does it typically take to converge to the right solution?
4) Add noise to the autocorrelation data. At what SNR does the algorithm fail to be reliable? Include representative images at a few different noise levels.