Image Restoration Project

A large component of this course will be an image restoration project. Teams of several students will write a Genetic Algorithm (GA) program in Matlab, Python, C to restore the corrupted Lena image below. Teams will be given the following information:

The corrupted image that teams must attempt to restore. The original uncorrupted Lena image. The form of the corruption source:

N(row, col)=NoiseAmp*sin([2*pi*NoiseFreqRow*row] + [2*pi*NoiseFreqCol*col])

NoiseAmp [0.00, 30.0], **NoiseFreqRow [0.00, 0.01]**, **NoiseFreqCol [0.00, 0.01]**

Teams must code a simple GA that optimizes the three unknown parameters NoiseAmp, NoiseFreqRow, and NoiseFreqCol such that the restoration error (the difference between the original and GA-optimised restored image) is minimised. To make things easy, we will measure the restoration error, thus:

Restoration error = (Ioriginal + NoiseGA)-Icorrupted

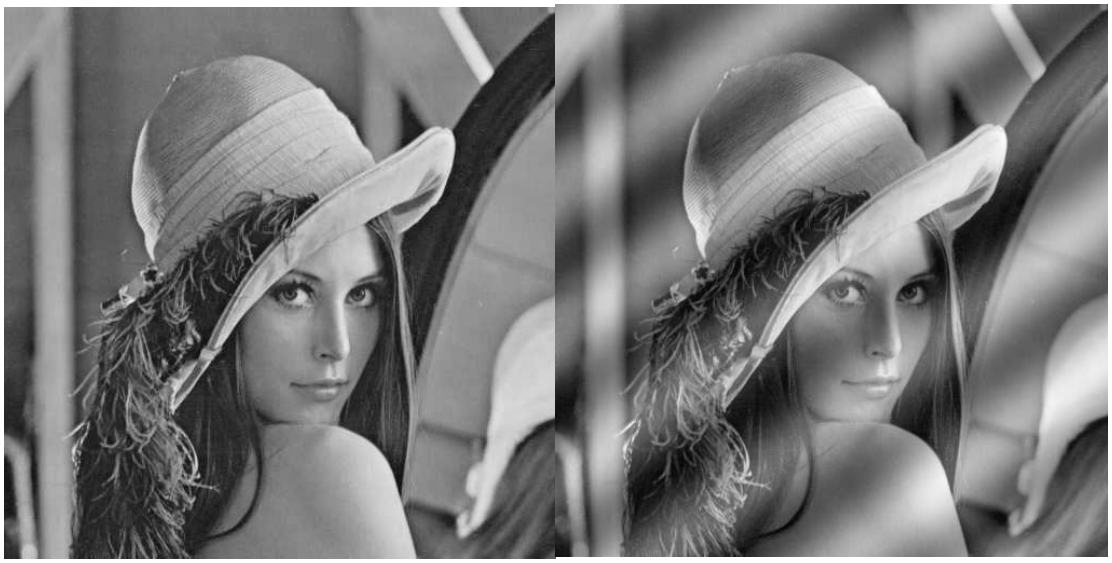
where Ioriginal is the original uncorrupted Lena image, Icorrupted is the corrupted image, and NoiseGA is the modelled GA corruption noise using the noise equation above.

Forming Teams

Each team will have at about two to four members (depending on class size), with one member being proficient at coding in a computer language (Python, C, Java, Matlab, etc.). Because class time will be limited, it is recommended that you code your GA in Matlab or another language that allows easy read/write/display of images. At least one team member should have access to your chosen programming language at their lab, because it will be necessary for coding work to be continued outside of class hours (yes, **homework**!!).

When you choose your team members, ensure that you all have a time each week when all members are available to meet for coding work outside class times.Reports will be written in English, so it is recommended that at least one member has good English writing skills.

Original (left) image and Corrupted (right) image



NB:

N1.- **Index** start from **1** (**ONE**) as in Matlab : **N(row, col**

N2.- Before saving your **float arrays** as **images**, **clip** them to **(0, 255)** and then convert them to **uint8**