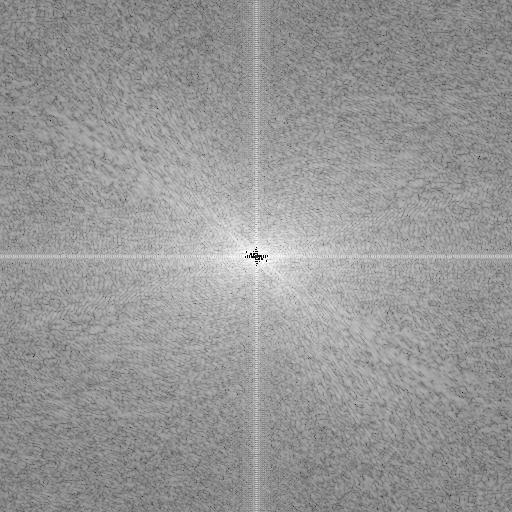
John McCormack  
Assignment #5  
4/15/2015  
  
fft.py

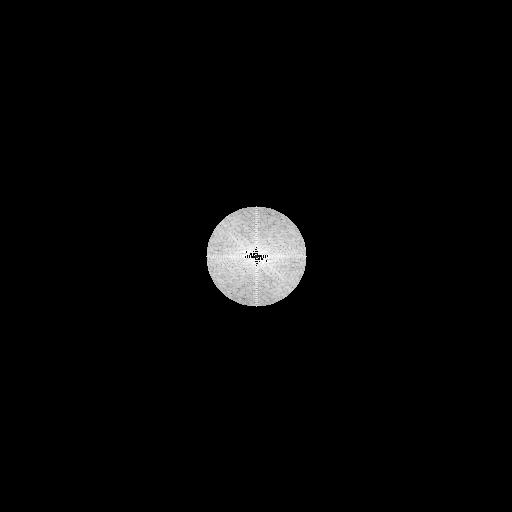
This program is used to generate the FFT of the image, and then to filter it. An object is instantiated with the image path. Each filter is then applied using a different method. A graph method was included as part of a test run, but is not necessary to the codes completion. Two private methods, \_\_pad and \_\_fft, are called upon instantiation. These generate the self.paddedimage and self.fft instance variables. There are several other instance variables included for convenience. A more in depth overview of the code can be found in the comments of the file.



The Original Image



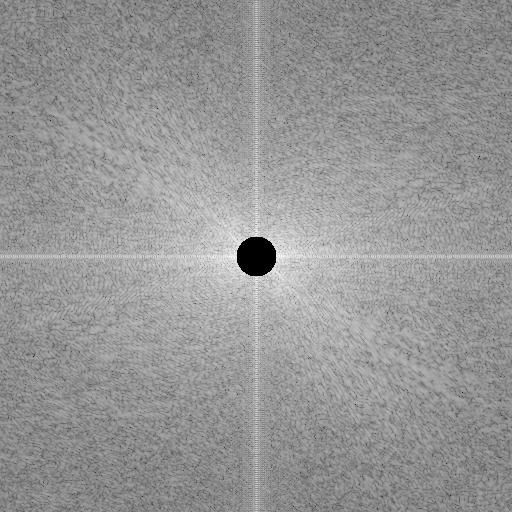
The padded FFT of the image



The FFT of the Ideal Low Pass Filter



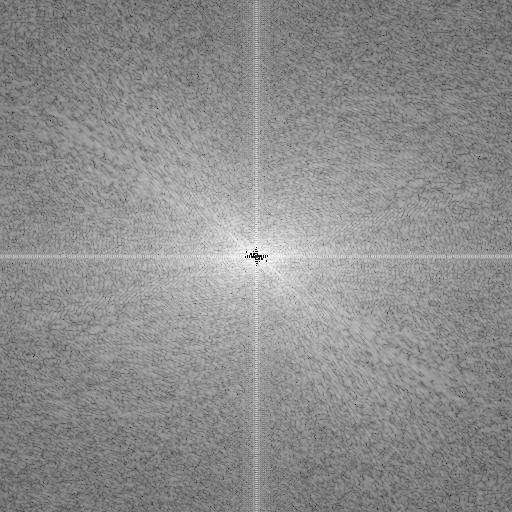
The result of the Ideal Low Pass Filter. It is easy to see the ringing effect here.



FFT of the Ideal High Pass Filter



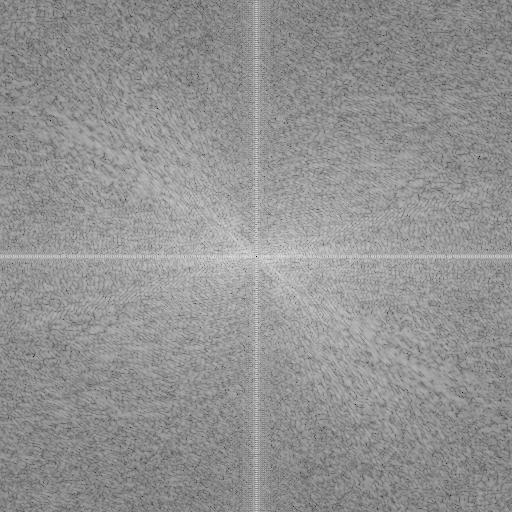
The result of the Ideal High Pass Filter. Again, the ringing is clear.



The Gaussian Low Pass FFT, the difference is harder to see.



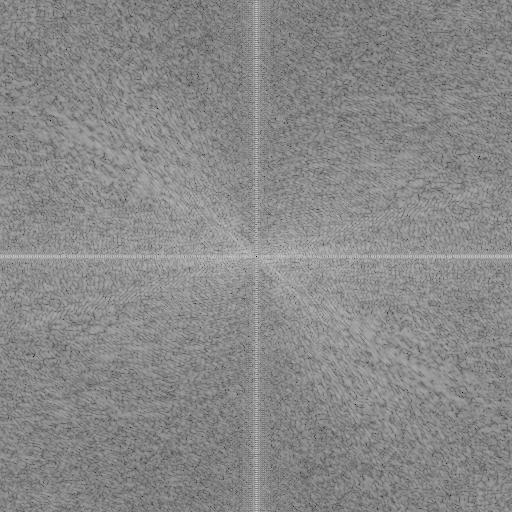
The result of the Gaussian Low Pass is significantly more subtle.



The FFT of the Gaussian High Pass Filter



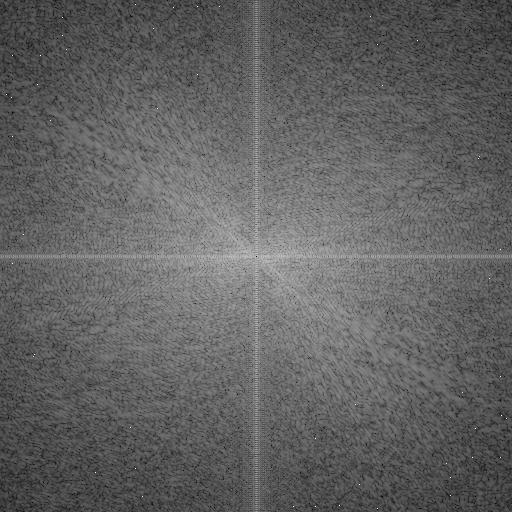
The output of the Gaussian HP Filter, the ringing is gone.



The FFT of the Laplacian



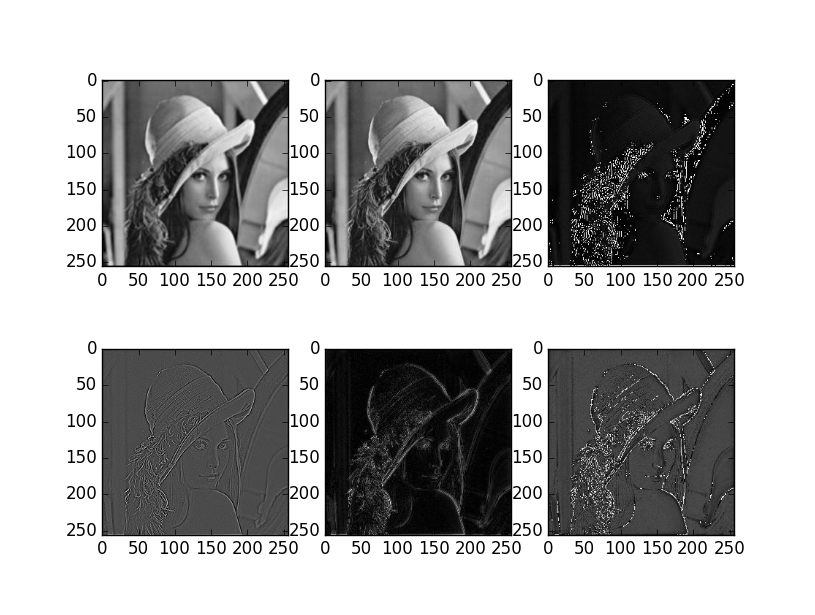
The Laplacian Filter



The FFT of the Gaussian, followed by the Laplacian



The resulting image is a blurring of the laplacian.



The comparison of the spatial and FFT images.

The two Gaussian blurs are very similar. Black represents and exact match. There are minimal differences. With the laplacian, the differences were more distinct. The Spatial filter shifted the image, I tried to account for this but might have done so incorrectly, as it remained shifted.