# EXPERIMENT 4

# Frequency Filtering

- *Write C++/Image-J modular functions to perform the following operations on the 512 × 512 grayscale test images, e.g. lena_gray_512.jpg, jetplane.jpg, lake.jpg, livingroom.jpg, mandril_gray.jpg, pirate.jpg, walkbridge.jpg.*
    - *FFT2 (takes input image filename as the argument; gives 2D FFT coefficients as output)*
    - *IFFT2 (takes 2D FFT coefficients as input argument; gives the back-projected/ reconstructed image as output)*
    - *Perform Ideal, Gaussian, and Butterworth low-pass and high-pass filtering, taking cut-off frequency, D0, and image filename as input arguments) respectively with Ideal_LPF, Ideal_HPF, Gaussian_LPF, Gaussian_HPF, Butterworth_LPF and Butterworth_HPF*

***Display the (shifted) magnitude spectrums of the input, the filter and the filtered output. You may make use of the tracker/slider function to choose images, filter types and cut-off frequencies.***


Group 16

Arnab Mondal – 14EC35031

Sandeep Mishra – 14EC35033

# INTRODUCTION

The objective of this experiment is to apply following Frequency Filtering operations on an image:

1. Ideal Low-pass Filter

2. Ideal High-pass Filter

3. Butterworth Low-pass Filter

4. Butterworth High-pass Filter

5. Gaussian Low-pass Filter

6. Gaussian High-pass Filter

Each of the above filter helps in modification or extracting features by manipulating frequency spectrum of the image. The image is Fourier transformed, multiplied with the filter function and then re-transformed into the spatial domain. Suppressing low frequencies result in a enhancement of edges, while suppressing high frequencies results in a smoother image in the spatial domain. All frequency filters can also be implemented in the spatial domain and, if there exist a simple kernel for the desired filter effect, it is computationally less expensive to perform the filtering in the spatial domain. Frequency filtering is more appropriate if no straightforward kernel can be found in the spatial domain, and may also be more efficient. Functions for various filters are as follows:

- *Ideal Low-pass Filter* : In the following function, C x and C y are the cutoff frequencies for the frequency spectrum in x-direction and y-direction respectively.

$$H(\omega_x, \omega_y) = \begin{cases} 1 & \text{if } \omega_x \leq C_x, \omega_y \leq C_y \\ 0 & otherwise \end{cases}$$

- *Ideal High-pass Filter* : In the following function, C x and C y are the cutoff frequencies for the frequency spectrum in x-direction and y-direction respectively.

$$H(\omega_x, \omega_y) = \begin{cases} 1 & \text{if } \omega_x \geq C_x, \omega_y \geq C_y \\ 0 & otherwise \end{cases}$$

- *Butterworth Low-pass Filter* : In the following function, C and n are the parameters that decide the spread and steepness of the function

$$H(\omega_x, \omega_y) = \frac{1}{1 + \left( \frac{\omega_x^2 + \omega_y^2}{c_2} \right)^{2n}}$$

- *Butterworth High-pass Filter* : In the following function, C and n are the parameters that decide the spread and steepness of the function

$$H(\omega_x, \omega_y) = 1 - \frac{1}{1 + \left( \frac{\omega_x^2 + \omega_y^2}{c_2} \right)^{2n}}$$

- *Gaussian Low-pass Filter* : In the following function, σ is the square root of variance (variance = $\sigma^2$)

$$H(\omega_x, \omega_y) = e^{-\frac{\omega_x^2 + \omega_y^2}{2\sigma^2}}$$

- *Gaussian High-pass Filter :* In the following function, σ is the square root of variance (variance = $\sigma^2$ )

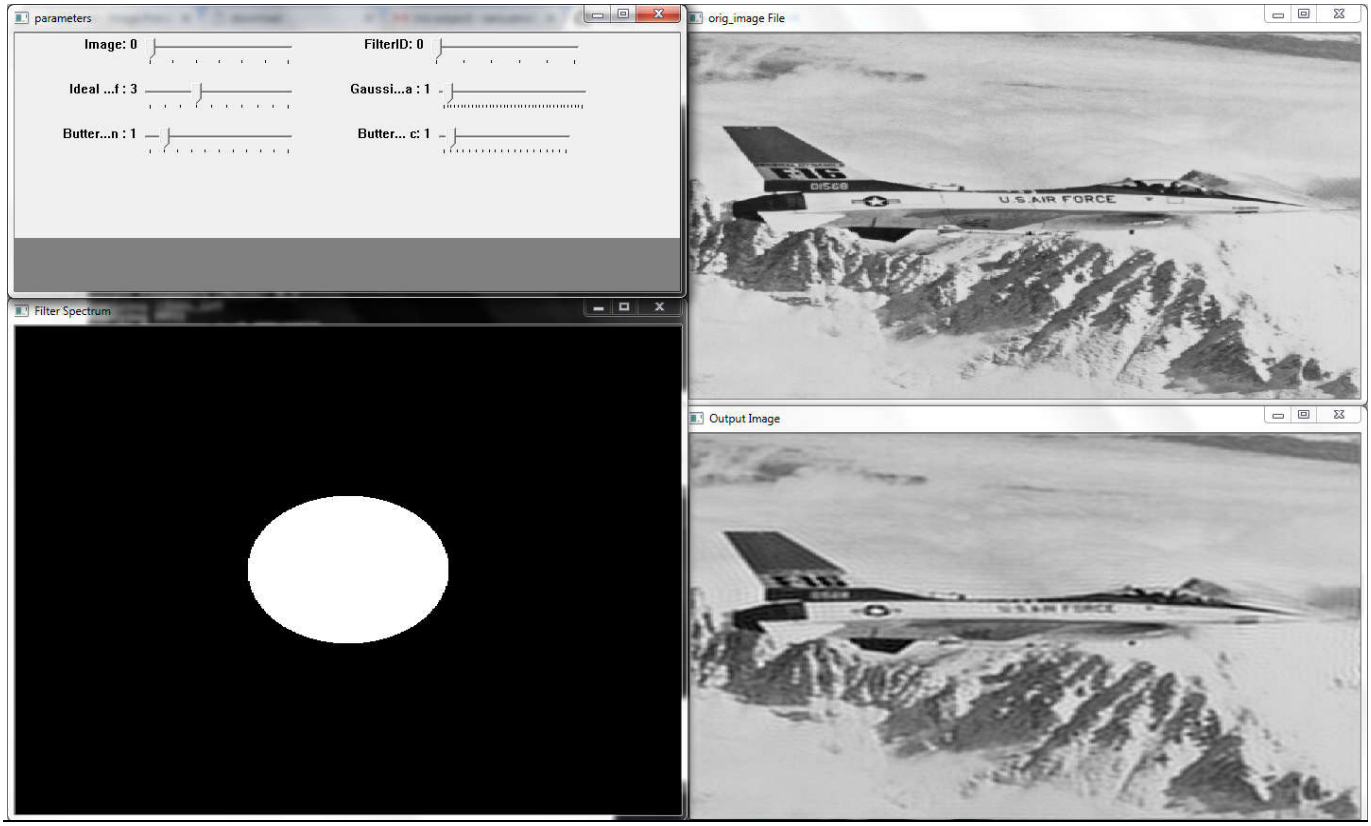
$$H(\omega_x, \omega_y) = 1 - e^{-\frac{\omega_x^2 + \omega_y^2}{2\sigma^2}}$$

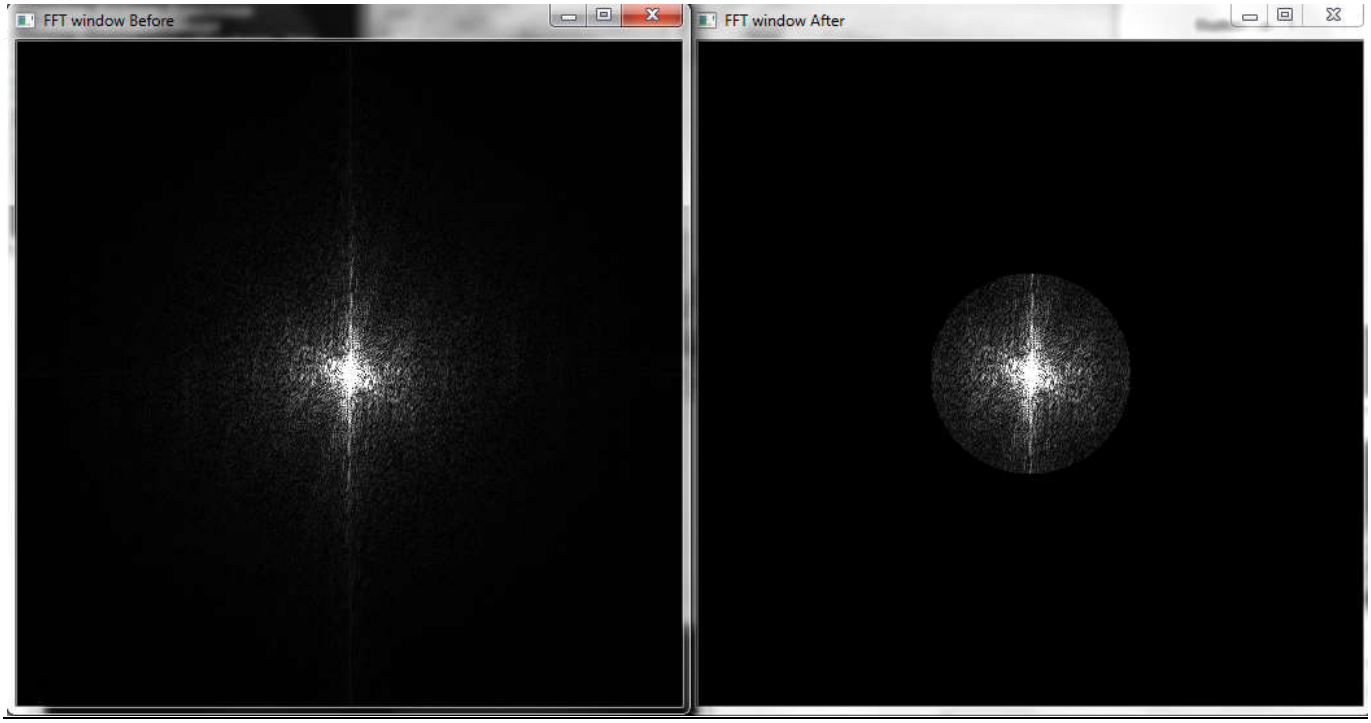
# ALGORITHM

2-D FFT requires 1-D FFT on rows first and then on columns. To do this, we first perform FFT on each row using Divide and Conquer strategy. Then we transpose this coefficient matrix and perform FFT on rows again. Same strategy is used for Inverse FFT. We calculate filter coefficients by using the functions as mentioned above.

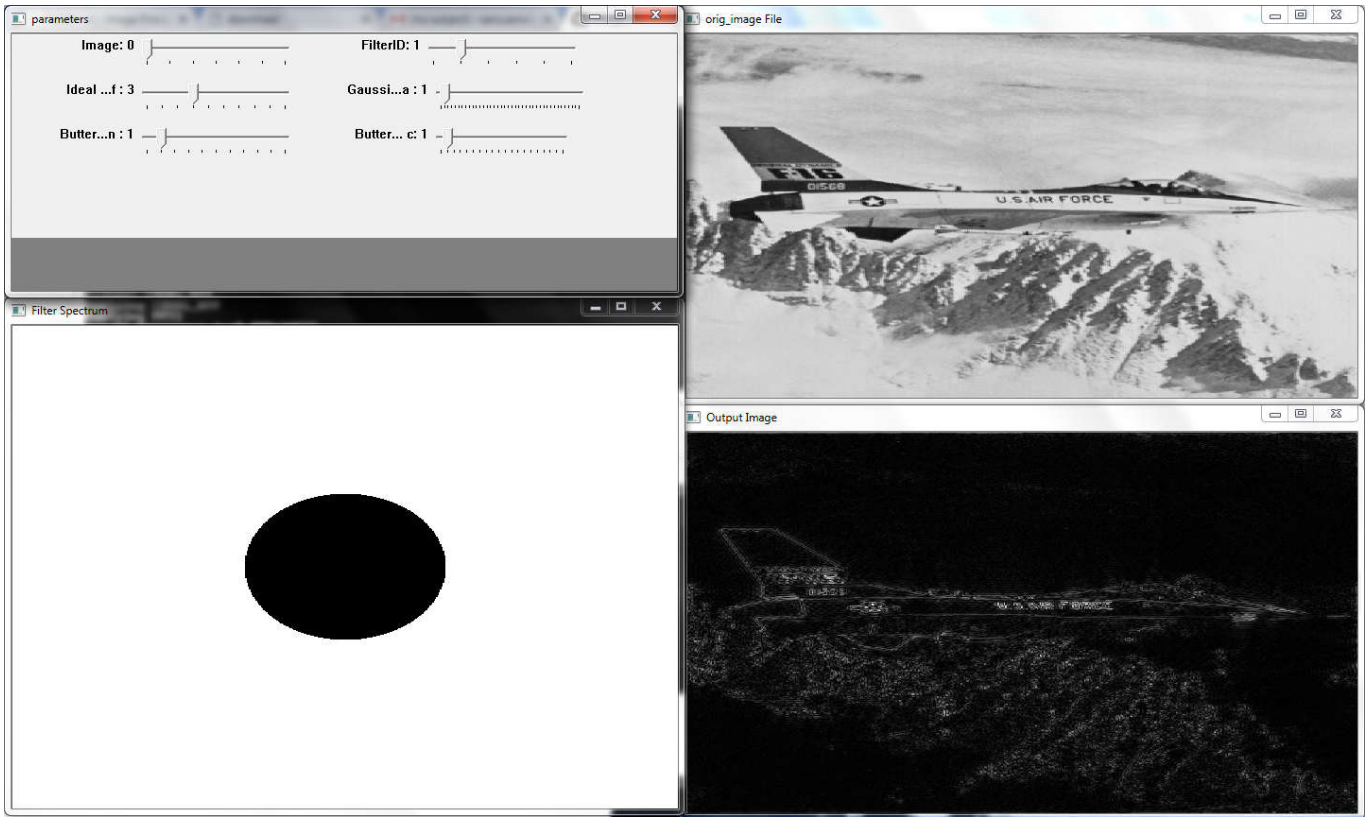
# OUTPUT RESULTS

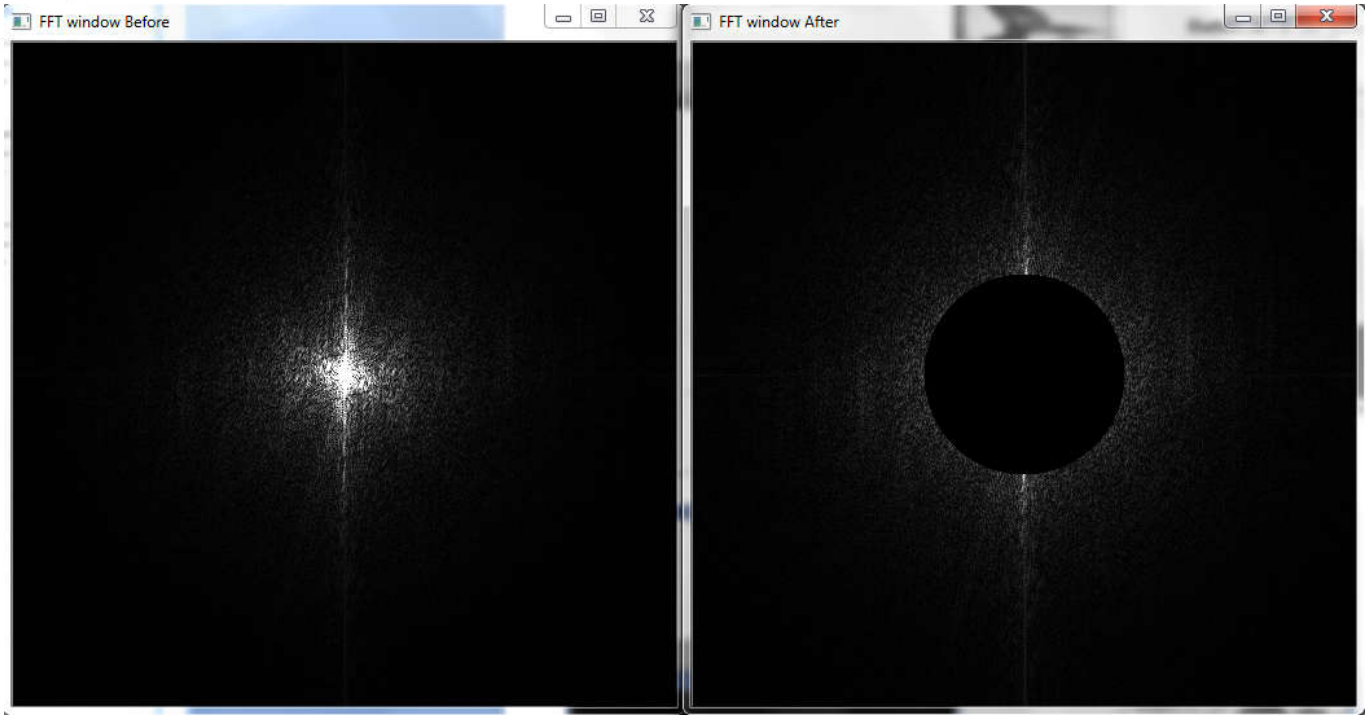
Ideal Low-pass Filter



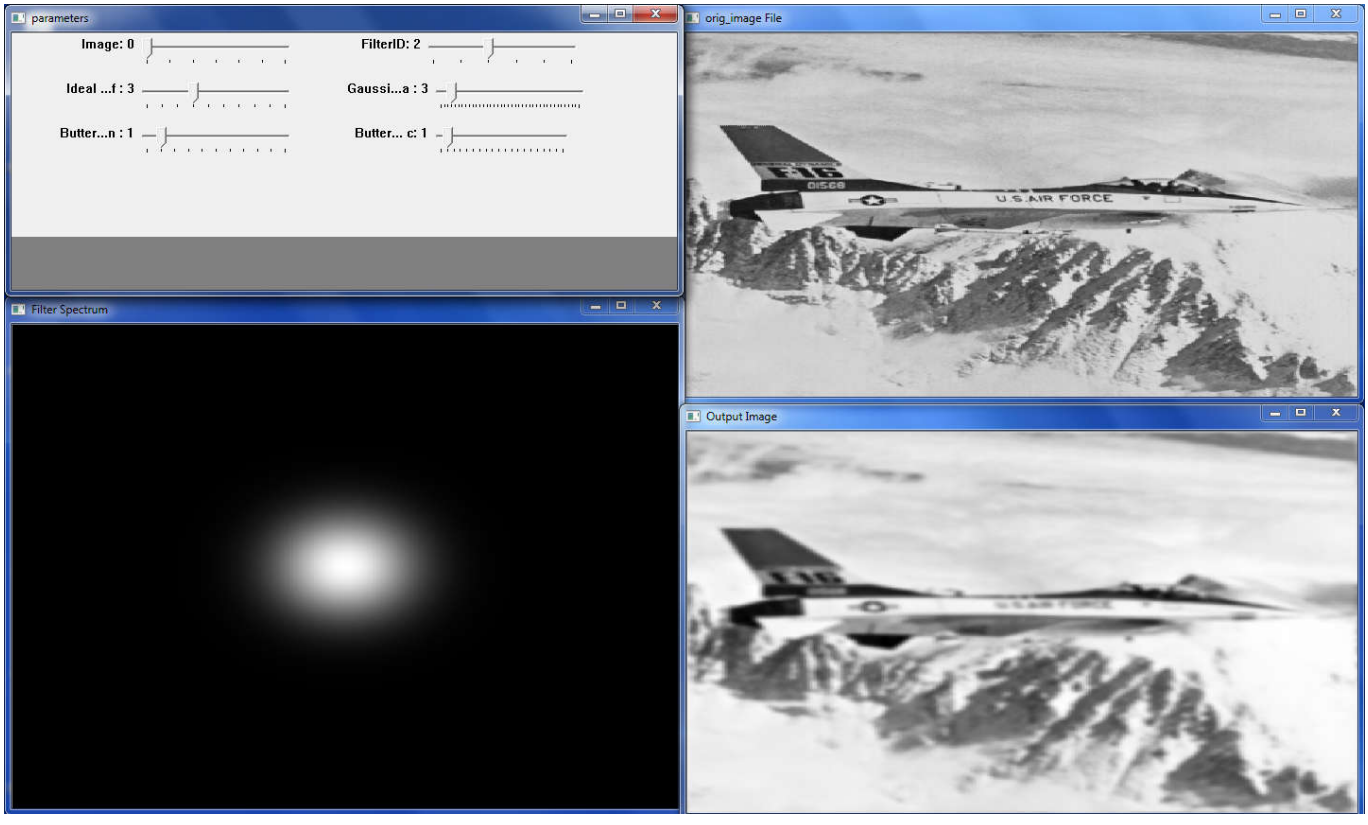


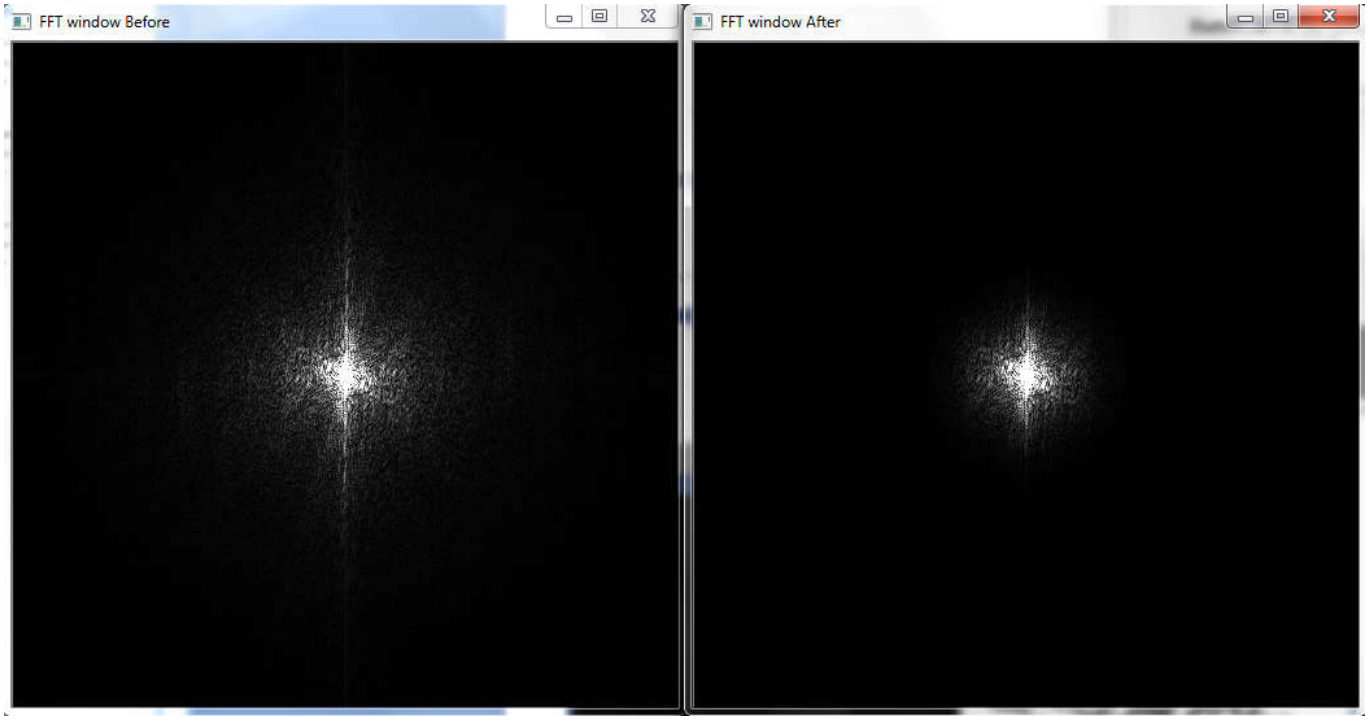

## Ideal High-pass Filter



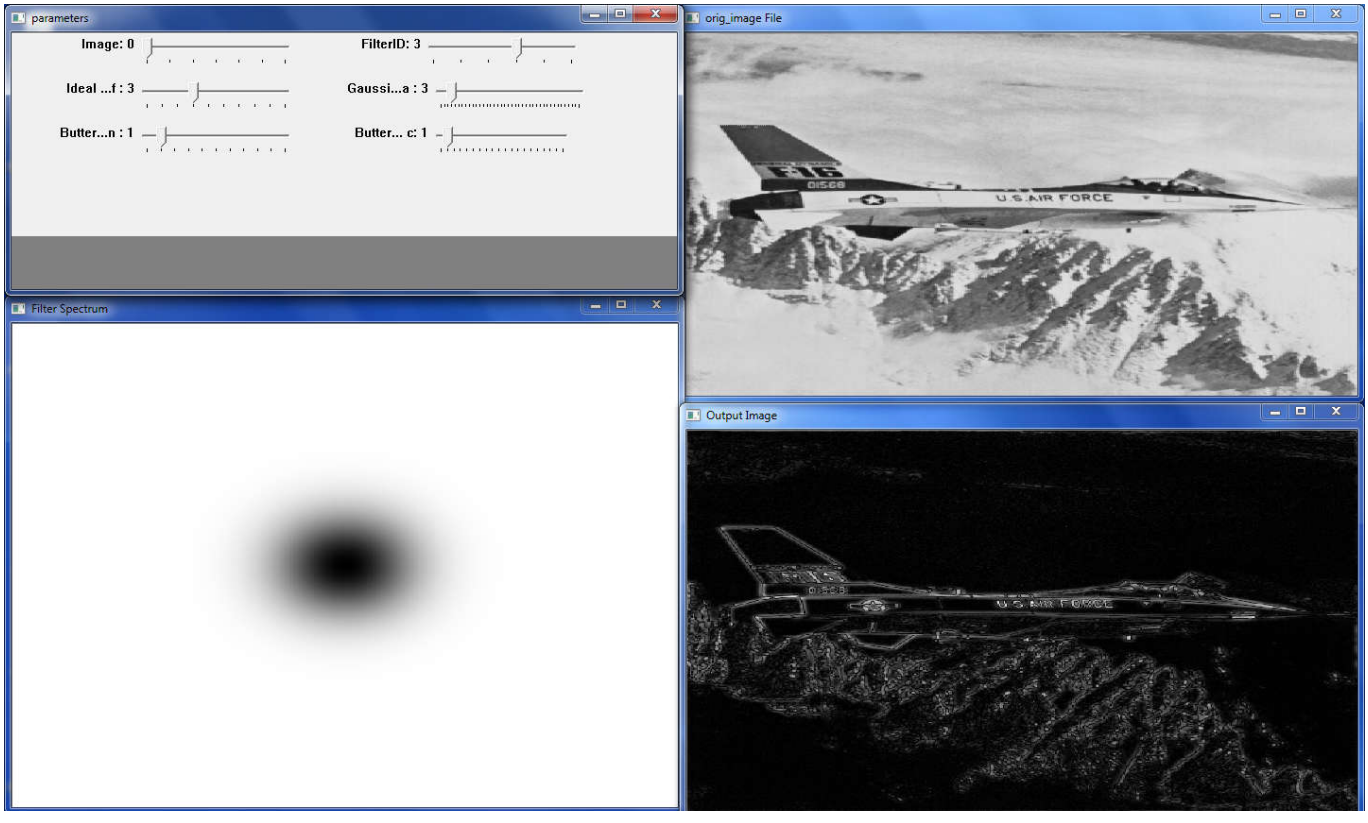


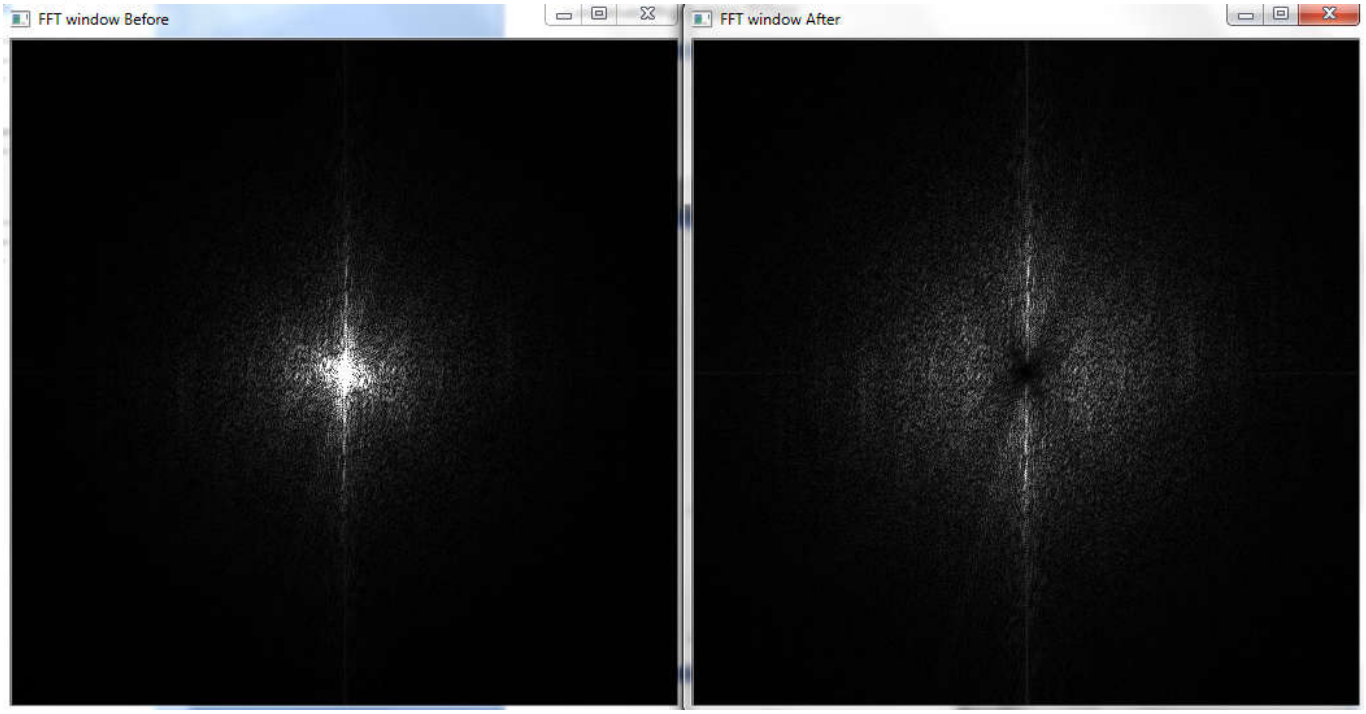

## Gaussian Low-pass Filter



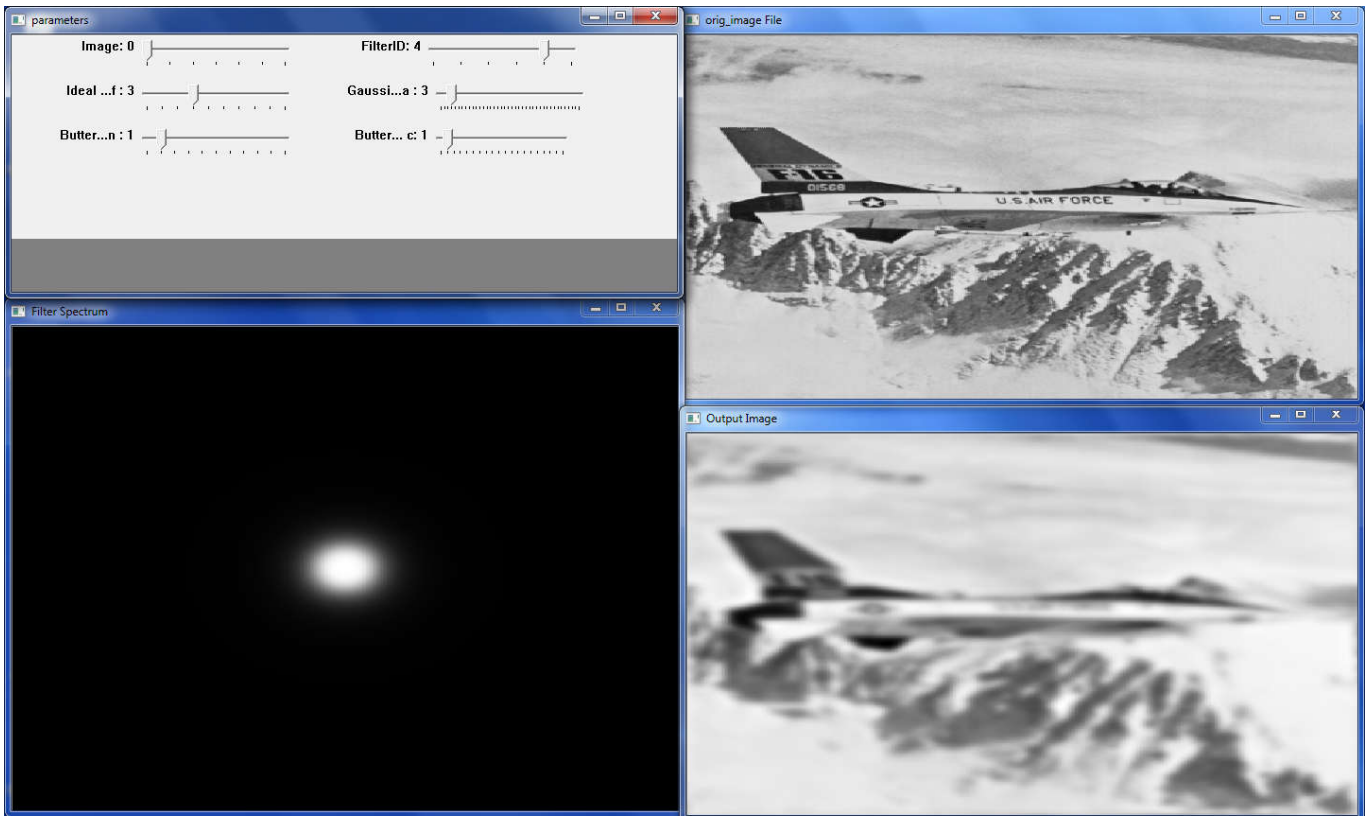


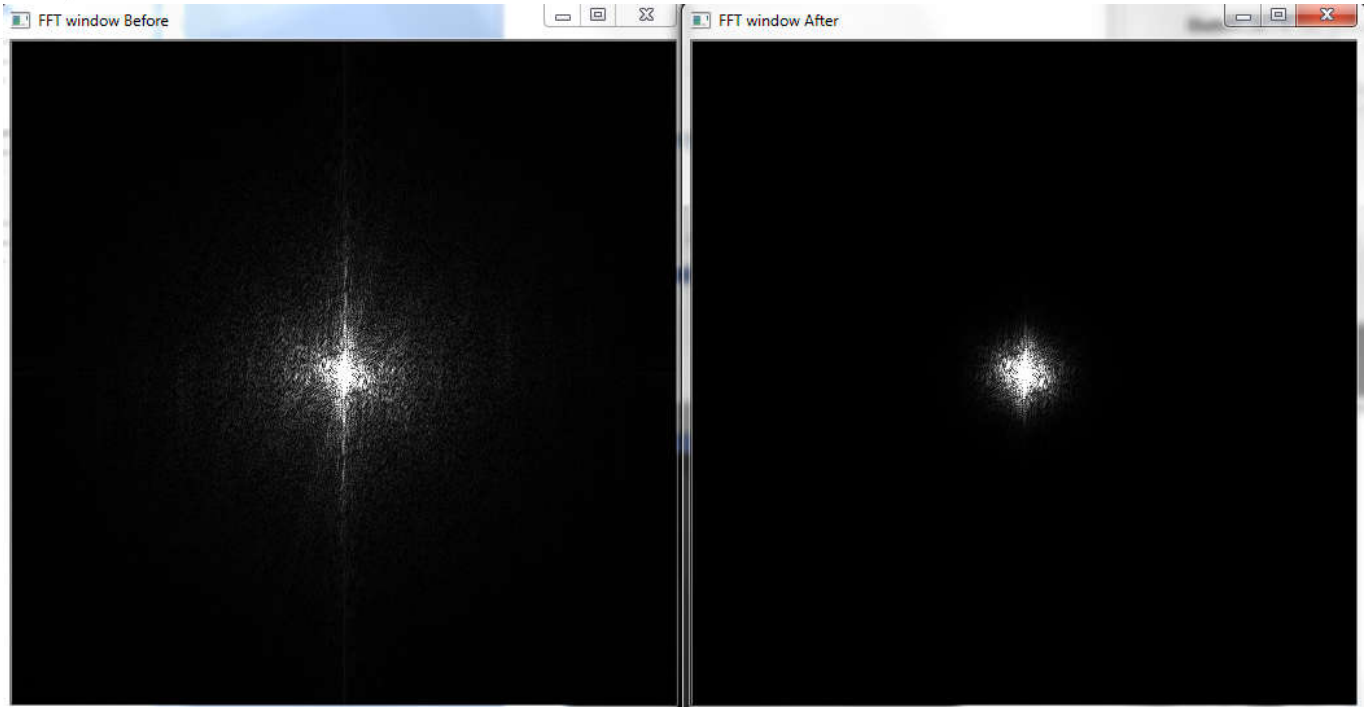

## Gaussian High-pass Filter



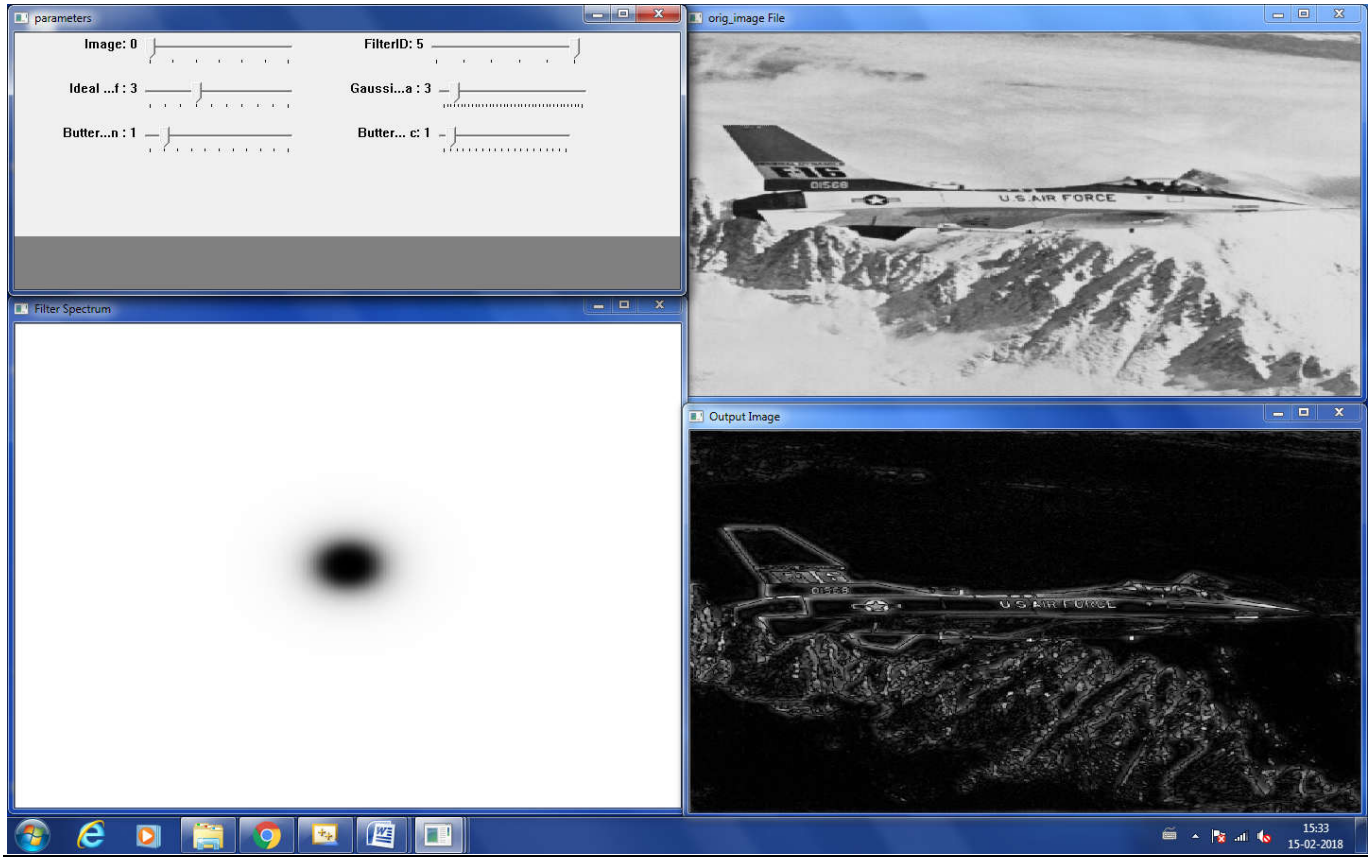


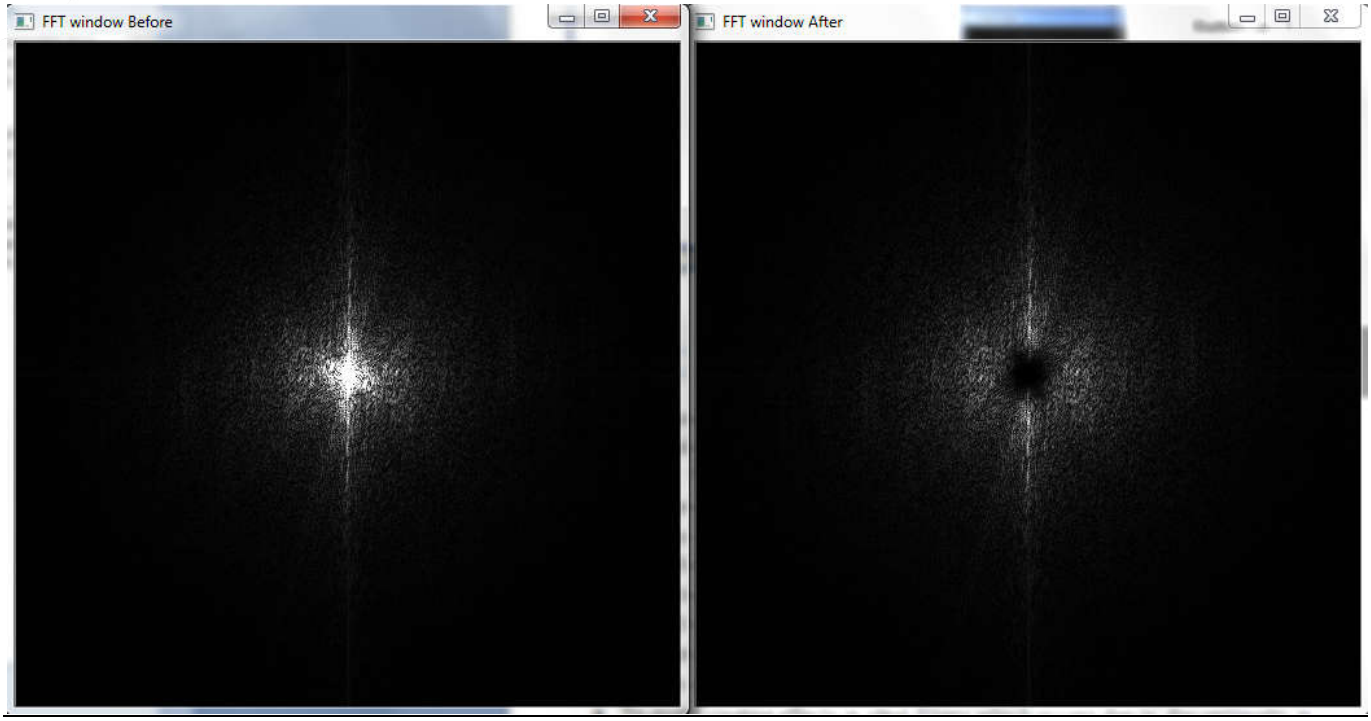

## Butterworth Low-pass Filter






## Butterworth High-pass Filter






# DISCUSSIONS

- Frequency domain filtering provides more control over frequency. For example, in an ideal low-pass filter we can remove high frequency components easily. Thus they are easier for filtering operations.
- Frequency filtering are more computationally expensive than the spatial filters because of the requirement of fourier transform and inverse fourier transform. Thus they are use only in case we can't create a mask in spatial domain for an operation.
- Ideal filters suffer from ringing (Gibbs) phenomenon. It means some similar to oscillation response are observed in the output. It is due to non-continues nature of the filter.
- To avoid ringing effects in ideal filters which occurs due to discontinuity in filter response, Butterworth and Gaussian filters are used which are continuous. The parameters (like σ or order) decide how close they are to the ideal filters by affecting the slope in transition band.

# SOURCES

[1] https://classes.soe.ucsc.edu/ee264/Fall11/LecturePDF/8-SpectralFiltering.pdf

[2] https://en.wikipedia.org/wiki/Butterworth

[3] https://en.wikipedia.org/wiki/Gaussian_filter

[4] http://www.programming-techniques.com/2013/01/low-pass-filters-blurring-in-image.html