Lab 2

Convolution, Fourier

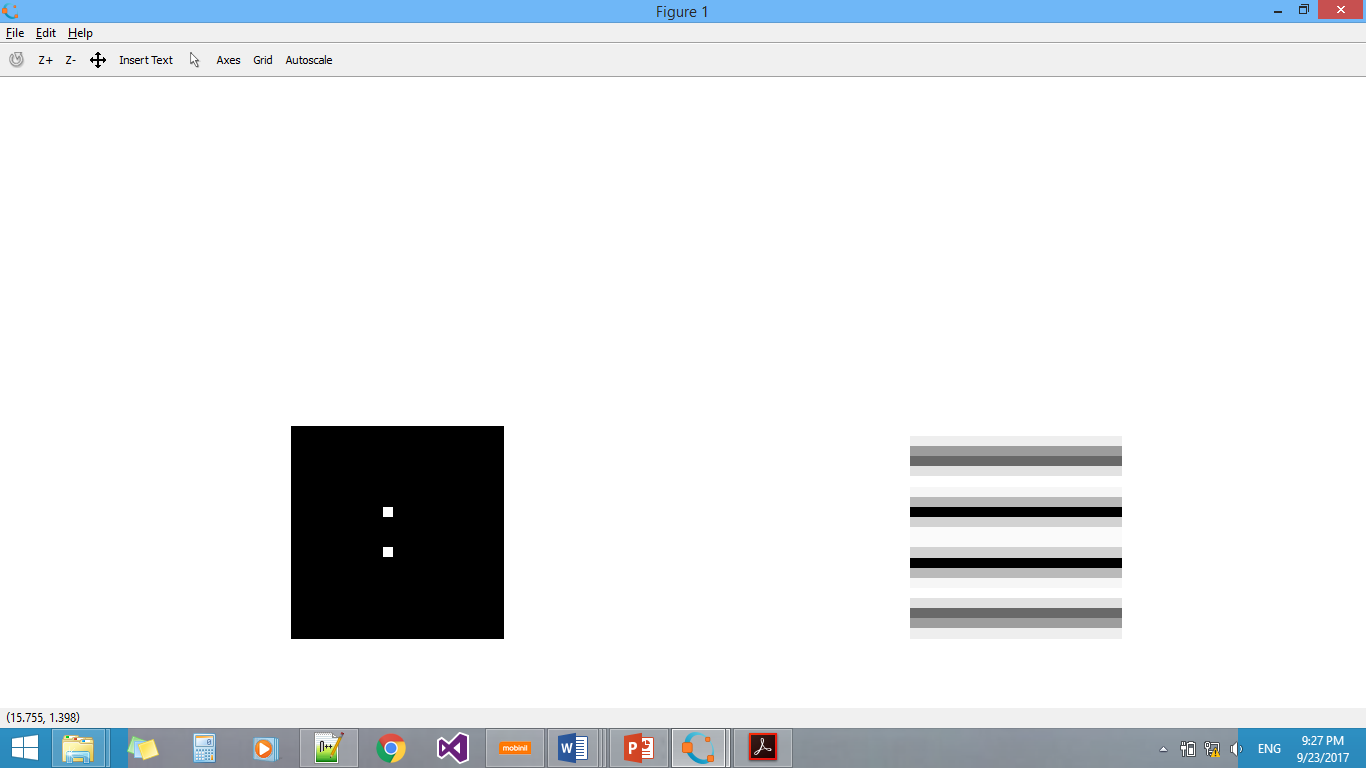
# Objective:

* Learn the concept of Convolution in space domain.
* Learn the concept of Inverse Fourier Transform.
* Learn the concept of Multiplication in frequency domain.

# Example: Inverse Fourier Transform

In this example we show the transformation for an image from frequency domain to spatial domain using inverse Fourier transform.

* Create a zeros matrix of size 21x21, then change the following pixels to be ones:
  + Pixels (11,10) and (11,12)
* Get the inverse Fourier transform of the image using .
* The result of ***ifft2*** contains complex and very small values that must be pre-processed before displaying.
* This is done by getting the ***abs*** of the image , then get the ***log( image +1).***
* Display each of the above images and its inverse using ***imshow ( image ,[])*** , we use this empty matrix to adjust the display range.



* What image gives an inverse Fourier transformation to be something like this?

# Example: Multiplication in frequency domain

In this exercise we transfer the image and the filter to the frequency domain, hence we can perform a multiplication operation, then we inverse transfer the result.

* Read any image from your choice then convert it to gray level.
* Calculate the Fourier transform for it using ***fft2***  , then shift it using ***fftshift***
* Generate the filter .
* Calculate the Fourier transform of the filter using ***fft( filter , rows(image), columns(image))***.
* Perform a multiplication operation between the image and the filter in frequency domain using np.multiply***.***
* Display the result use the ***abs***  then  ***log ( image +1 )*** before the imshow.
* Then you apply inverse fourier transformation using ***ifft2.***
* To display the result use the ***abs***  then  ***log ( image +1 )*** before the imshow.

# Experiment 1:Convolution in space domain (30minutes )

In this exercise we experiment convolution in space domain for 2d images and linear filters.

* Read any image from your choice then convert it to gray level.
* Generate another version of the image after adding salt and paper noise with density=0.05.
* Convolve in space domain using the function ***convolve2d*** the following filters with the images and write your comments about the results:
  + On the image after adding noise.
  + On the image without noise.
  + On the image without noise.
  + On the image without noise.
* After applying each filter display

1. gray scaled original image
2. image after adding noise to it
3. The four images after applying the four filters

Images in **one single figure** using ***show\_images*** instruction.

* What are the uses of these filters?
* For the second filter, how can we make the output more descriptive?

Useful New Functions and Attributes

|  |  |  |
| --- | --- | --- |
| Name | Attribute or Function | Usage |
| np.log | Function | to get log for matrix (element-wise) |
| np.abs | Function | to get absolute value for matrix (element-wise) |
| np.multiply | Function | to multiply two matrices (element-wise) |
| np.power | Function | to power a matrix (element-wise) |

For Fourier Transform functions, check: <https://docs.scipy.org/doc/scipy/reference/fftpack.html>