Assignment8

Shufan Xia

(Dated: Apr.6th, 2020)

1. 7.9 IMAGE DECONVOLUTION

a) Reproduce the blurred photo:

First read "blur.txt" which records the brightness at each (x,y) on the photo by ***np.loadtxt()***. The photo is plotted by making a density plot by ***imshow()*** from the data in "blur.txt".

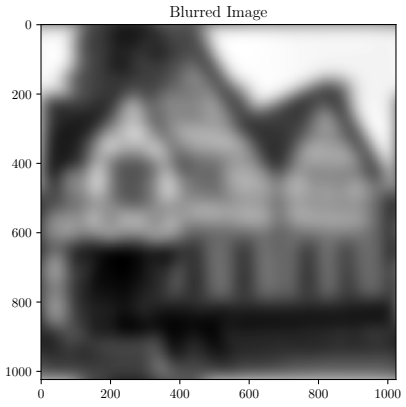

FIG. 1: The original blurred image

b) Define and plot the point spreading function:

The point spread function in xy coordinate is defined as:

$$f(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

where $\sigma = 25$. Since at later point, we need to do a FFT on $f(x,y)$, we need to treat $f(x,y)$ a periodic function along both x and y axis over the length of the photo. I need to make some transformation on x and y so that x and y values looks like Fig2

Therefore, x value repeats a half cycle over its period L=1024, and is symmetric with respect to x=512. I found this transformation works:

$$x = \begin{cases} 512(x//512+1) - x & \text{x//512 if odd} \\ x - 512(x//512) & \text{x//512 if even} \end{cases} \tag{2}$$

Take x = 0 to 1023 and y=0 to 1023 because there is 1024×1024 data on the gird of the original data file and index in python starts from 0, calculate $f(x,y)$ and plot the function in a 2D density plot, see Fig3.

c) Deconvolve the image: To reconstruct the deblurred image, we need the result of Fourier Transform of the deblurred image or the discrete 2D brightness function. Let's say the coefficient of the 2D Fourier Transform

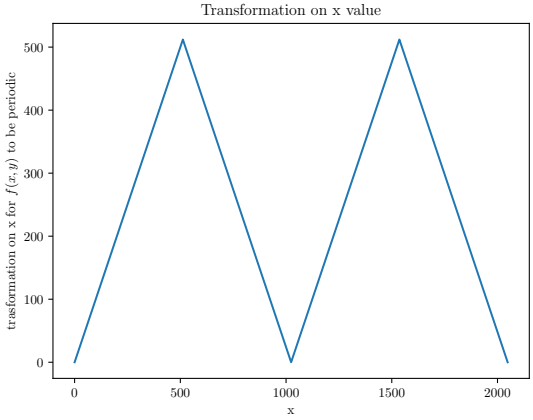

FIG. 2: For the point spread function to be periodic over the length of the image and the value of x and y repeat at the corner of the images when L=1024, a transformation on x and y like above is necessary

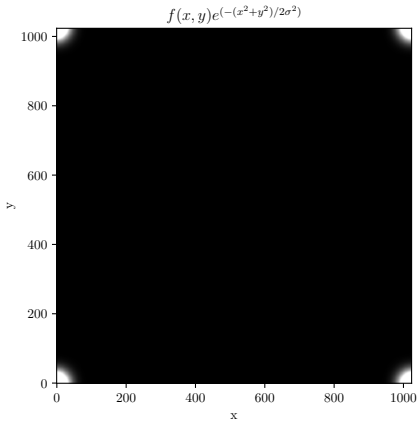

FIG. 3: The point spread function: $f(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$. $f(x,y)$ is periodic over the length of the image, and negative value x and y repeat at the four corners of the image

is a_k for the deblurred image, b_k for the original blurred image, and f_k for the point spread function. Newman shows:

$$a_k = \frac{b_k}{L^2 f_k} \tag{3}$$

I used ***numpy.fft.rfft2()*** to find the Fourier Transform of the discrete density function of the original blurred data $b(x,y)$, and the discrete point spread func-

tion $f(x, y)$ to find b_k and f_k. Applying Eq3, I found the Fourier Transform of the brightness function of the deblurred image, $a(x, y)$. Using textbf*numpy.fft.rfft2()*, I did a reversed Fourier Transform to reconstruct the 2D brightness function $a(x, y)$. Using ***imshow()*** to plot the 2D function, I got the deblurred image:

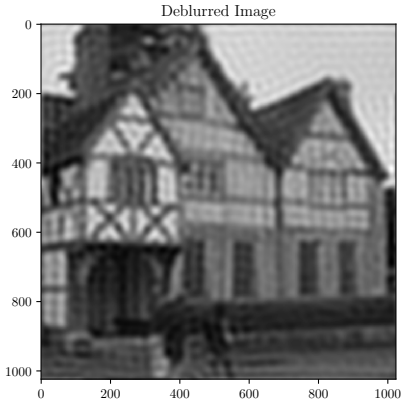

FIG. 4: Image after deblurring

A Gaussian function has very small Fourier Transform coefficients for high frequency components. To avoid division by zero error, the coefficients of very high frequency components of the Gaussian functions are left alone; therefore, we can not the recover the a_k, and are only able to use b_k when reconstructing the unblurred photo. Therefore, we lose the original high frequency signals which give us the sharp features of a photo.

2. SURVEY QUESTIONS

The homework took me 5 hours. I learned use FFT from **numpy** to do Fourier Transform and inverse Fourier transform in 2D. I learned the process to blurring and deblurring an image.