CS-663 Assignment 5 Q6

Soham Naha (193079003)
Akshay Bajpai (193079002)
Mohit Agarwala (19307R004)

November 15, 2020

6

Consider two different Laplacian filter kernels $k_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ and $k_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$. Write down the formulae for their N, N-point Discrete Fourier Transforms in the report. Compute their N, N-point Discrete Fourier Transforms in MATLAB with $N = 201$, i.e. where the spatial and frequency indices range from -100 to 100 in both canonical directions. Display the magnitude of the DFT on a log scale using the `imshow` and `surf` functions in MATLAB along with a colorbar. Besides the plots, include the code snippet for the DFT computation and display in the report. Comment on the difference in the Fourier transforms of the two kernels. [10+5=15 points]

The first kernel given is as $k_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
Here, $N = 201$.
Thus, for the above k_1, we will appropriately zero-pad to get a new k_1 which is a NN matrix. Hence,$DFT(k_1)$ at a particular frequency (u, v) is given as:

$$K_1(u, v) = 2 * \exp(-\frac{j2\pi(u+v)(N+1)}{2N})\{\cos(\frac{2\pi u}{N}) + \cos(\frac{2\pi v}{N}) - 2\} \tag{1}$$

Similarly, for the kernel $k_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$ with N=201, the $DFT(k_2)$ of the padded kernel k_2 at frequency (u, v) will be given as the equation :

$$K_1(u, v) = 2*\exp(-\frac{j2\pi(u+v)(N+1)}{2N})\{4-\cos(\frac{2\pi u}{N})-\cos(\frac{2\pi v}{N})-\cos(\frac{2\pi(u+v)}{N})-\cos(\frac{2\pi(u-v)}{N})\} \tag{2}$$

The same equations have been used to calculate the DFT of the kernels and their plots. **myDFT.py**

```python
import numpy as np
import matplotlib.pyplot as plt
import cmath
from mpl_toolkits.mplot3d import axes3d

def computeDFT(N, matrix,KernelType, verbose=True):
    center = (N+1)/2
    fac = 2*np.pi/N
    X = np.linspace(-(N-1)/2, (N-1)/2, num=N)
    Y = np.linspace(-(N-1)/2, (N-1)/2, num=N)

    my_ft_lap = np.zeros((N,N)).astype(np.complex64)
    for u in range(N):
        for v in range(N):
            if KernelType=="Normal":
                my_ft_lap[u,v] = 2*cmath.exp(-1j*fac*center*(u+v))*(np.cos(fac*u)+np.cos(fac*v)-2)
            elif KernelType.startswith("Diagonal"):
                my_ft_lap[u,v] = 2*cmath.exp(-1j*fac*center*(u+v))*(4 -
                            np.cos(fac*u)-np.cos(fac*v)-np.cos(fac*(u+v))-np.cos(fac*(u-v)))

```

```python
    my_ft_lap = np.fft.fftshift(my_ft_lap)
    magnitude = np.log(np.abs(my_ft_lap)+1)

    if verbose:
        plt.figure()
        # plt.colorbar(cmap="jet")
        plt.title("Log Magnitude plot of kernel "+KernelType)
        plt.imshow(magnitude,extent=[-100,100,-100,100])
        plt.set_cmap("jet")
        plt.colorbar()
        #plt.xlim(xmin = np.min(X), xmax=np.max(X))
        #plt.ylim(ymin = np.min(Y), ymax=np.max(Y))
        plt.savefig("../images/"+ KernelType +"LogMagnitude.png", cmap="jet", bbox_inches="tight")

        fig = plt.figure()
        ax = plt.axes(projection ='3d')
        plt.set_cmap("jet")
        x,y = np.meshgrid(X,Y)
        surf = ax.plot_surface(x, y, magnitude, cmap="jet")
        fig.colorbar(surf, ax=ax)
        plt.title("Surface Plot of "+KernelType + " Laplacian Kernel")
        #plt.show()
        plt.savefig("../images/"+ KernelType +"SurfacePlot.png", cmap="jet",
                    bbox_inches="tight")

if __name__=="__main__":
    N = 201
    laplacianMatrix1 = np.array([[0,1,0],[1,-4,1],[0,1,0]])
    laplacianMatrix2 = np.array([[-1,-1,-1],[-1,8,-1],[-1,-1,-1]])
    computeDFT(N, laplacianMatrix1, "Normal")
    computeDFT(N, laplacianMatrix2, "Diagonal Added")
```

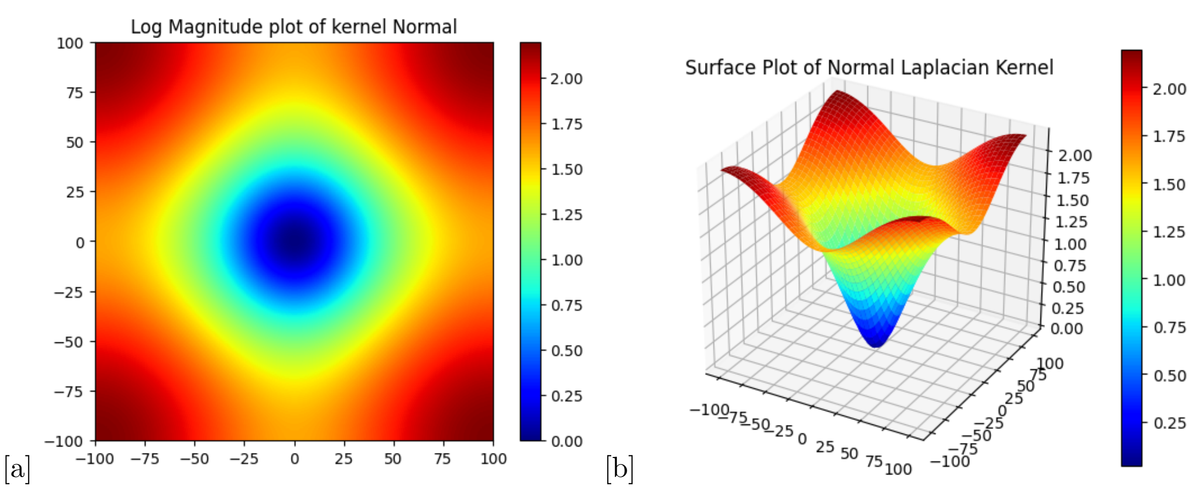

Figure 1: (a) Log Magnitude Plot of k_1 (b) Surface Plot of k_1

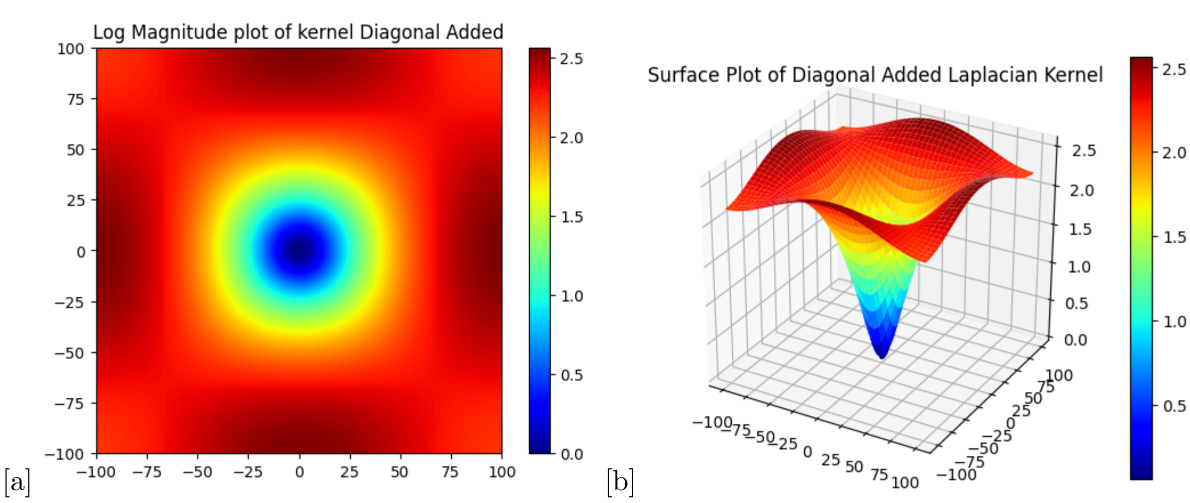

Figure 2: (a) Log Magnitude Plot of k_2 (b) Surface Plot of k_2

Discussions :

For an ideal high-pass (differentiating) system, we would want the frequency response to be cylindrical with low values at low frequencies, with sharp-rise at the cut-off frequency and beyond.
From the images/contour plots, we can observe that both the kernels have circular contours in the low frequency ranges, thus it acts ideally in this frequency range.
The gradient of the cone in the center frequency regions for laplacian kernel k_2 is quite sharp as compared to that of the kernel k_1. So, the kernel k_2 sharpens the image better than the kernel k_1, as the cutoff frequency D_0 changes as the shape of the filter changes.
The frequency response of both the plots become parabolic at the end frequency regions, with the structure of k_1 facing upwards, while that of k_2 forming a paraboloid type of structure, with almost constant magnitude like that of ideal highpass filter.
We observe that k_2 is a better high-pass than the kernel k_1.