Digital Image Processing Report
Project 2: Image enhancement in frequency domain

2019233181 Jiale Xu

November 13, 2019

Part 1: Frequency Domain Filtering

1. Description

Enhance the blurred image ("lenna.mat") by applying an IHPF, BHPF, GHPF in the frequency domain and Laplacian filter in the spatial domain. Note: the equation as below should be implemented when filters are applied

$$g(x, y) = f(x, y) + c\left[\nabla^2 f(x, y)\right]$$

Here, $n = 2$ for BHPF. And some other parameters needed to be set, for example $c = 10$, cutoff frequency is 100.

- Please display the result by using *subplot()* in Matlab.
- Please discuss the difference of the four sharpening filters (Spatial Laplacian, IHPF, BHPF and GHPF). (You can adjust the parameters c and cutoff frequency to get different images to compare, but all the images you compared should be shown in your report and ppt.)
- *fft2()*, *ifft2()* and *fftshift()* can be used in this project.
- IHPF, BHPF, GHPF should be generated from the formulas in slides/textbook.

2. Contents

2.1 Laplacian filter

The definition of Laplacian is

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{1}$$

Where

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

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \tag{3}$$

Finally, we have

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y) \tag{4}$$

As the following figure has shown, (a) is the Laplacian kernel used to implement Eq.(4), (b) is the kernel used to implement an extension of this equation that includes the diagonal terms, (c) and (d) are two other Laplacian kernels. **In this project, we adopt (d) as the Laplacian kernel.**

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

a b c d

2.2 Frequency highpass filters

Ideal highpass filter (IHPF):

$$H_{HP}(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases} \tag{5}$$

Butterworth highpass filter (BHPF):

$$H(u,v) = \frac{1}{1+[D_0/D(u,v)]^{2n}} \tag{6}$$

Gaussian highpass filter (GHPF):

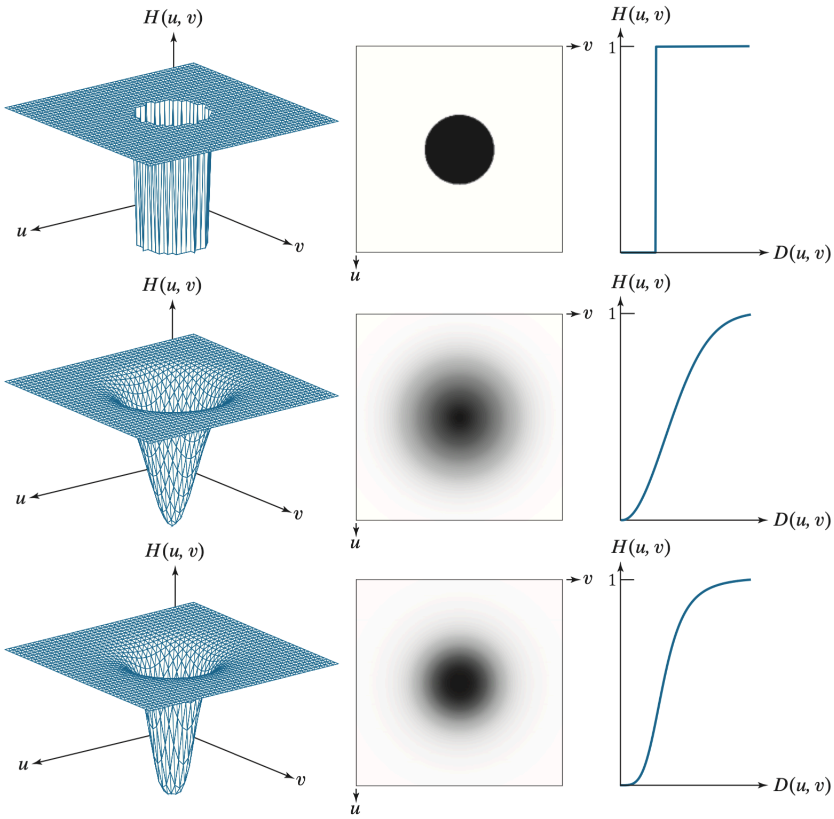
$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2} \tag{7}$$

Where in Eq.(5)(6)(7), D_0 is the cutoff frequency, and

$$D(u,v) = \left[(u-P/2)^2 + (v-Q/2)^2\right]^{1/2}$$

$$P = 2M$$

$$Q = 2N$$

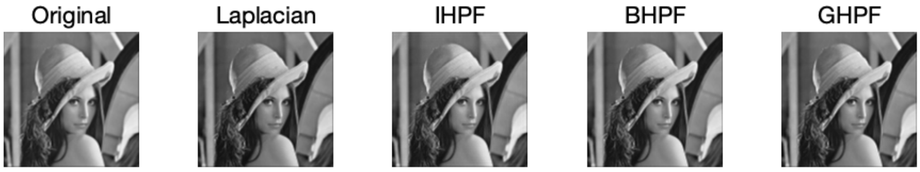


The process of filtering in the frequency domain can be summarized as follows:

1. Given an input image $f(x,y)$ of size $M \times N$, obtain the padding sizes P and Q, P = 2M and Q = 2N.
2. Form a padded image $f_p(x,y)$ of size $P \times Q$ using zero-, mirror-, or replicate padding.
3. Multiply $f_p(x,y)$ by $(1)^x + y$ to center the Fourier transform on the $P \times Q$ frequency rectangle.
4. Compute the DFT, $F(u,v)$, of the image from Step 3.
5. Construct a real, symmetric filter transfer function, $H(u,v)$, of size $P \times Q$ with center at $(P/2, Q/2)$.
6. Form the product $G(u,v) = H(u,v)F(u,v)$ using elementwise multiplication: $G(i,k) = H(i,k)F(i,k)$ for $i = 0, 1, 2, ..., M-1$ and $k = 0, 1, 2, ..., N-1$.
7. Obtain the filtered image (of size $P \times Q$) by computing the IDFT of $G(u,v)$:
$$g_p(x,y) = \left(\text{real}\left[\Im^{-1}\{G(u,v)\}\right]\right)(-1)^{x+y}$$
8. Obtain the final filtered result, $g(x,y)$, of the same size as the input image, by extracting the $M \times N$ region from the top, left quadrant of $g_p(x,y)$.

3. Results

In this project, we set $n = 2$ for BHPF, $c = 1$, and the cutoff frequency $D_0 = 100$. Here is the Result:

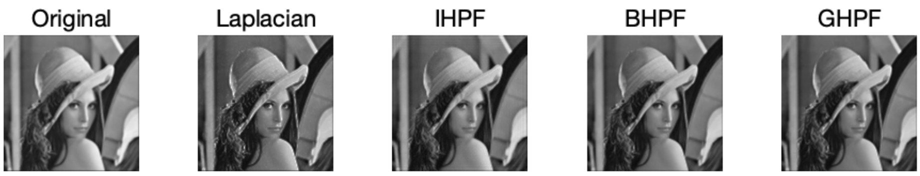


When $c = 10, D_0 = 100$, the result is:

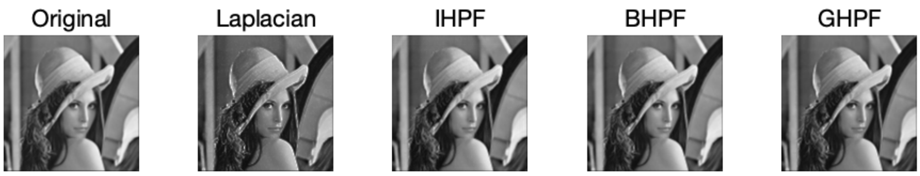
4. Discussion

What's the difference of the four sharpening filters (Spatial Laplacian, IHPF, BHPF and GHPF)?
When $c = 2, D0 = 50$, the result is:

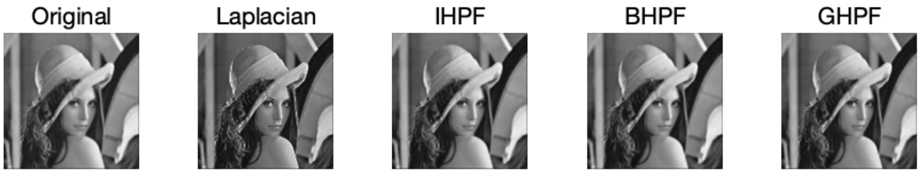
When $c = 2, D0 = 100$, the result is:



When $c = 2, D0 = 150$, the result is:



When $c = 2, D0 = 200$, the result is:



1. As the result shows, the detail in the image is unmistakably clearer and sharper than in the original image. Adding the Laplacian to the original image restored the overall intensity variations in the image. Adding the Laplacian increased the contrast at the locations of intensity discontinuities. The net result is an image in which small details were enhanced and the background tonality was reasonably preserved.
2. As the result shows, the ideal highpass filter produced results with severe distortions caused by ringing, while the other two frequency filters didn't.
3. The higher the curoff frequency is, the smoother the filtered image is.
4. GHPF produces smoother image than the other two frequency domain filters.

Part 2: Pseudo-color Enhancement

Problem 1: Transform the gray image to color image in spatial domain

1. Description

Transform the gray image to color image in spatial domain.

1. "Lenna"
2. "Remote Sensing Image"
   This method is often used in Remote Sensing Image, so please use "RSI.mat" to implement pseudo-color enhancement in frequency domain.

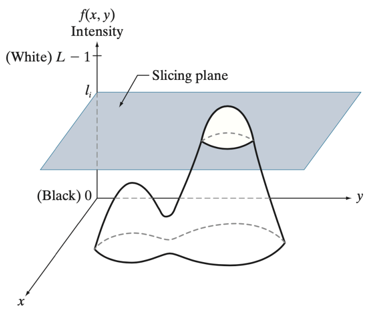
2. Contents

We use *intensity-slicing* technique to accomplish pseudo-color enhancement in spatial domain. It can be summarized as follows:

Let $[0, L1]$ represent the grayscale, let level l_0 represent black $[f(x, y) = 0]$, and level l_{L1} represent white $[f(x, y) = L1]$. Suppose that P planes perpendicular to the intensity axis are defined at levels $l_1, l_2, \cdots, l_P$. Then, assuming that $0 < P < L1$, the P planes partition the grayscale into $P + 1$ intervals, $I_1, I_2, \cdots, I_{P+1}$. Intensity to color assignments at each pixel location (x, y) are made according to the equation

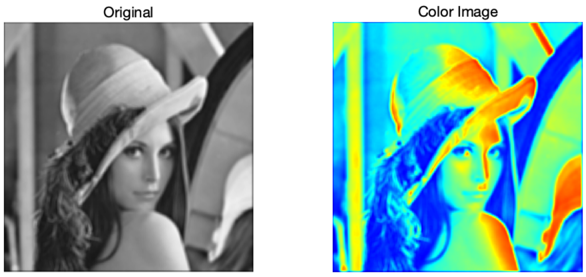
$$\text{if } f(x, y) \in I_k, \text{ let } f(x, y) = c_k \tag{8}$$

where c_k is the color associated with the k-th intensity interval I_k, defined by the planes at $l = k-1$ and $l = k$.



3. Results

Here is the result (use “jet” colormap):



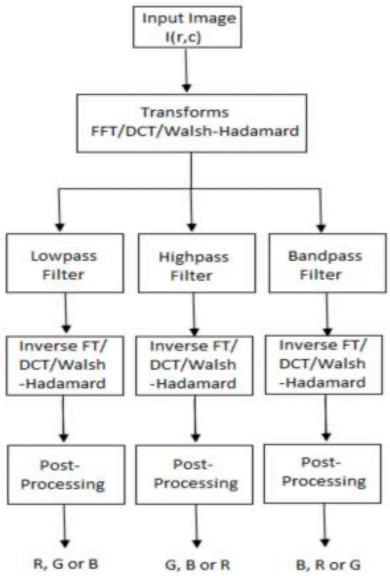
Problem 2: Transform the gray image to color image in frequency domain

1. Description

Transform the gray image to color image in frequency domain.

2. Contents

The pipeline is as the following figure shown:



First of all, we use Fourier transformation to transform the original image into frequency domain.
In the frequency domain, we adopt lowpass, bindpass and high pass Gaussian filters to process the low-frequency component, middle-frequency component, and high-frequency component respectively. Then we use inversed Fourier transform to transform the three components into spatial domain.
In the post processing procedure, we adopt the following techniques:

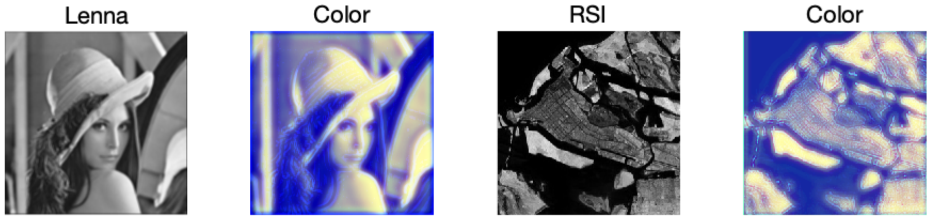
1. **Histogram equalization.** We adopt histogram equalization to the inversed-Fourier-transformed low-pass component to obtain the R channel of the RGB image.

2. **Histogram matching.** We adopt histogram matching to the inversed-Fourier-transformed bindpass component and highpass component, to make their histograms match the histogram of R, and obtain G and B channels.

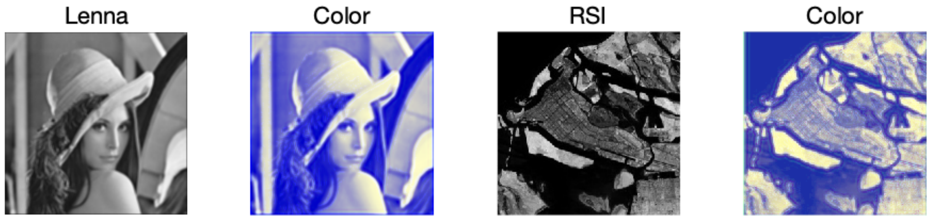
3. **Mean filtering.** Since there is too much noise in the B channel, we adopt mean filtering to it.

3. Results

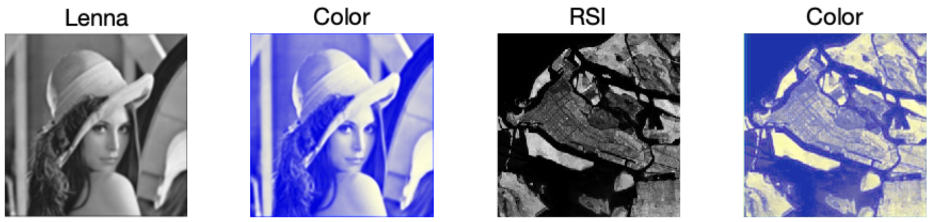
When $D_0 = 50, W = 60$, the result is:



When $D_0 = 100, W = 60$, the result is:



When $D_0 = 150, W = 60$, the result is:



When $D_0 = 200, W = 60$, the result is: