Fatemeh Darbehani [redacted]

ENSC474 – SFU – Spring 2017

Assignment 6

Low-pass and High-pass Filters

- In this assignment I implemented an ideal rectangular low-pass filter and ideal rectangular high-pass filter in frequency domain.
- The low-pass filter will create a rectangle with the size of cutoff frequencies specified by the user that is bright (1) in the inside and dark(0) in the outside.
- The high-pass filter will create a rectangle with the size of cutoff frequencies specified by the user that is dark(0) in the inside and bright(1) in the outside.

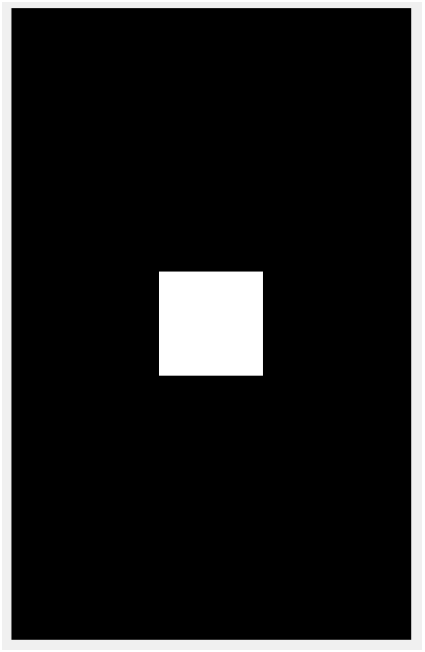

Figure1: Frequency Domain Low-pass Filter with cutoff of 100

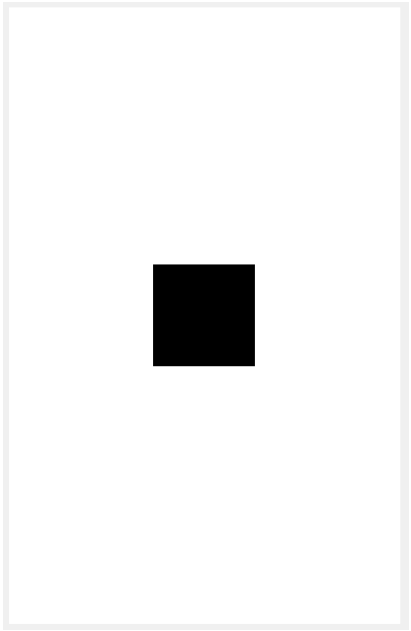

Figure2: Frequency Domain High-pass Filter with cutoff of 100

Low-pass and High-pass Filters

- Fast Fourier Transform of the image is calculated using inbuilt fft2 MATLAB function.
- To transfer low frequency components to the center, I used fftshift which is another MATLAB inbuilt function.
- Multiplying the transformed image by the low-pass filter will cut off all high frequencies.
- Multiplying the transformed image by the high-pass filter will cut off all low frequency components.

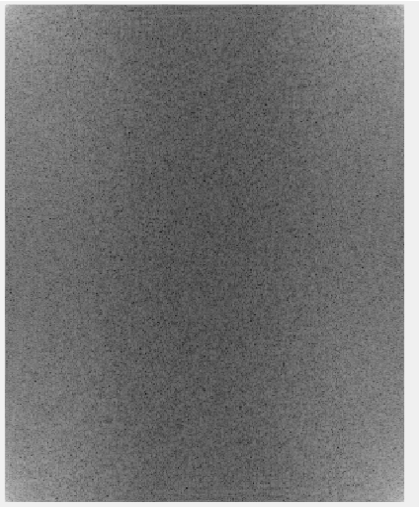

Figure3: FFT of the image before shifting

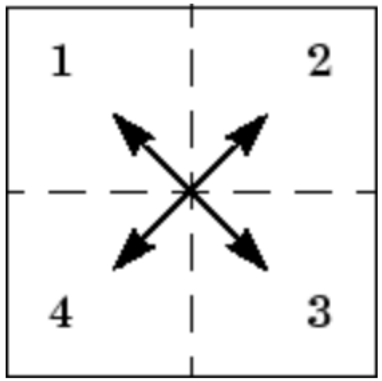

Figure4: Effect of fftshift function

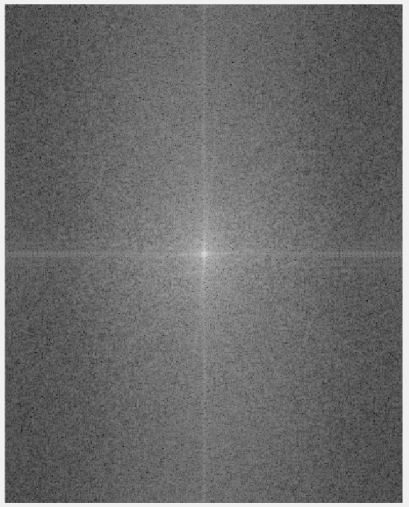

Figure5: FFT of the image after shifting

Low-pass and High-pass Filters

- After filtering, I used fftshift again to take high frequencies back to the center.
- I used ifftshift2 function to find the inverse Fourier transform of the filtered image.

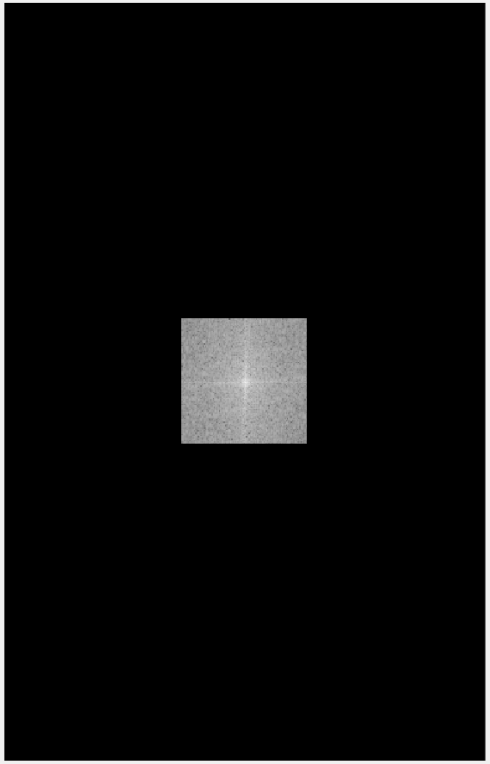

Figure6: Low-pass filtered image in frequency domain

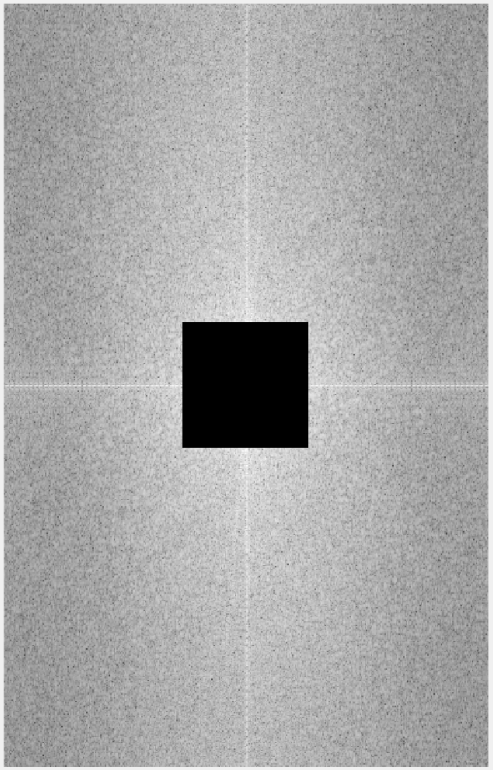

Figure7: High-pass filtered image in frequency domain

Low-pass and High-pass Filters

- The following 4 images were used to create 8 low-frequency and 8 high-frequency images to test my functions.

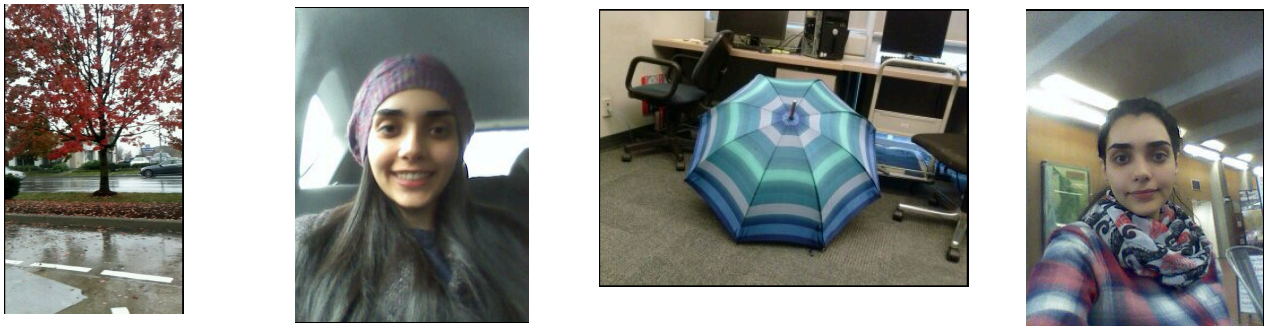

Figure8: Low-pass filtered images with cutoff of 5, 10, 450, and 50 from left to right

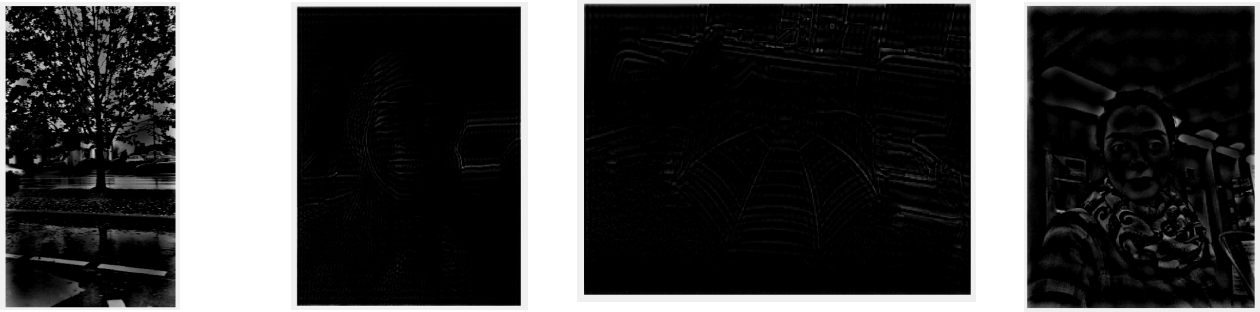

Figure9: High-pass filtered images with cutoff of 10, 60, 100, and 400 from left to

Rank Function

- Rank function first multiplies the shifted fft of the image by a low-pass filter that is 5% of the size of the image.
- Then, it find the ratio of sum of square of all low frequency components to sum of square of all frequencies.
- If the image is mostly high frequency, this ratio is small and if it is mostly lw frequency this ratio is big.

```
% ------------------------------------------------------------------- %
% Calculating Power of the original image and lowpass filtered image
% ------------------------------------------------------------------- %
img_squared = abs(img_fft).^2;
limg_squared = abs(Zl).^2;
sum_orig = sum(img_squared(:));
l_sum = sum(limg_squared(:));
% ------------------------------------------------ %
% Ratio of low frequencies over to all frequencies %
% ------------------------------------------------ %
l = (l_sum)/sum_orig;
l = abs(l);
```

Figure10: MATLAB code of low-freq/total-freq ratio

Rank Function

- The following bar chart displays the ranking of my 16 images.
- Note that all low-pass filtered images are ranked from 0 to 50 and all high-pass filtered images are ranked from 50 to 100.
- Also, the high-pass filtered image with cutoff of 10 has a rank of 56. This is expected as the image itself has both low and high frequencies and the high-pass filter is so small that is has a minimal effect on the image.

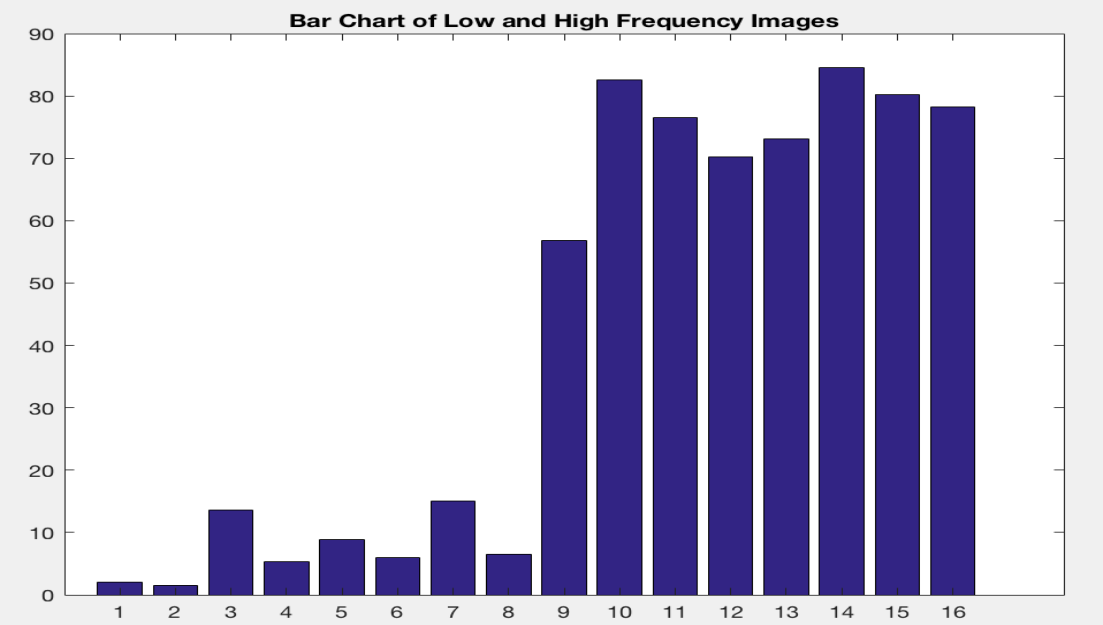

Figure 11: Bar chart of ranked images. Image 1 to 8 are predominantly low frequency while image 9 to 16 are predominantly high frequency

Conclusion

- In general, an image usually has high intensities at low frequencies and lower intensities at the higher frequencies
- In this rank function my cutoff frequency is 5% of the size of each image. This only includes a small range of frequencies but as we can observe it can include 50 to 100 percent of the frequency components of an image.
- If I used a bigger low-pass filter (higher cutoff), my ranks would be smaller and more images would be considered predominantly low frequency.