ELL715
Assignment 3 Report

Naman Jhunjhunwala (2017MT10737),Ankit Kumar (2017MT10727)

February 8, 2020

1 Part 1: Image Filtering

1.1 Sample Images

1.2 Point Spread Function

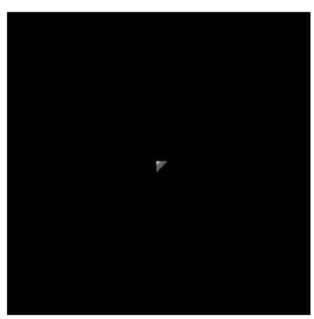


Figure 1: Image after applying IIR filter on a point image

After applying IIR filter we can see that the spread of point function is increased

1.3 IIR Filter

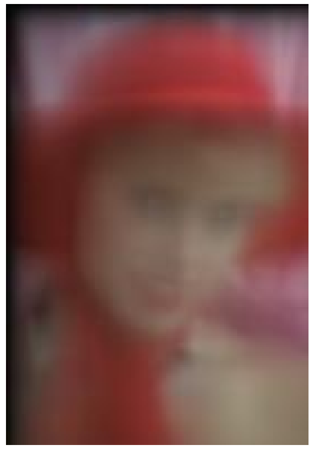


Figure 2: Sample image after applying IIR filter

We can see that IIR filter has blurred the image

1.4 Convolution

Figure 3: Sample image after convolving with h(m,n)

We can see that the filter used has blurred the image

1.5 Unsharp Masking

In this part we performed unsharp masking on image using following equation

$$y[m,n] = x[m,n] + alpha * (x[m,n] - a[m,n])$$

where $a[m,n]$ is the output after convolving the image with h[m,n]

alpha = 0.2 and 0.8

alpha = 1.5

1.6 Observations

- For the output image with alpha = 0.2 we can see that the image is sharpened by unsharp masking but some information has been lost due to conversion on intensity to eight bit
- For alpha = 0.8 and 1.5 image quality has decreased significantly even though image has sharpened

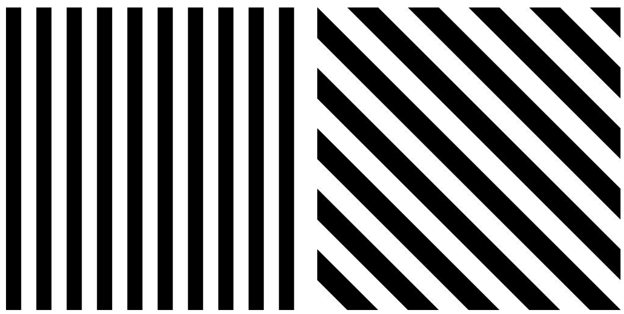
2 Part 2: Fourier Transform

2.1 Challenge Faced

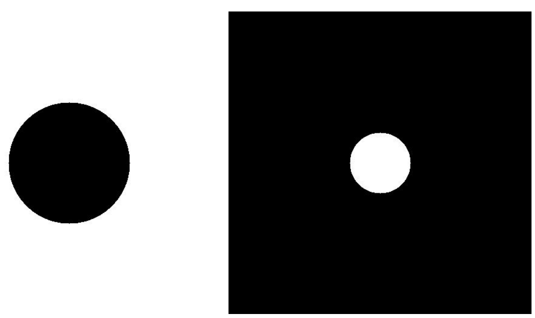
We implemented the Fourier transform ourselves. It had one major issue. The complexity of the naive implementation is $O(n^4)$. It's practically impossible to run it in case the image is significantly large, say $1000 * 1000$. Our image was $500 * 500$, but still, it was taking a lot of time, and hence we had to reduce it down. To do so, we used numpy matrix multiplication algorithms. Instead of writing down 4 nested for loops, we wrote the transform as a multiplication of 3 matrices, and used numpy to evaluate this multiplication. Since numpy is extremely fast in this, it helped reduce the taken by our code drastically.

2.2 Sample Images

Strips image and 45-degree rotated strips image:

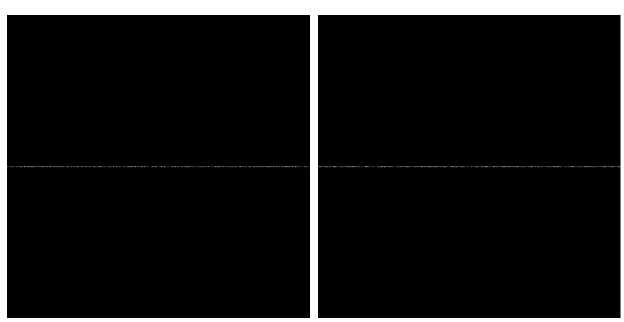


Black circle and white circle images:



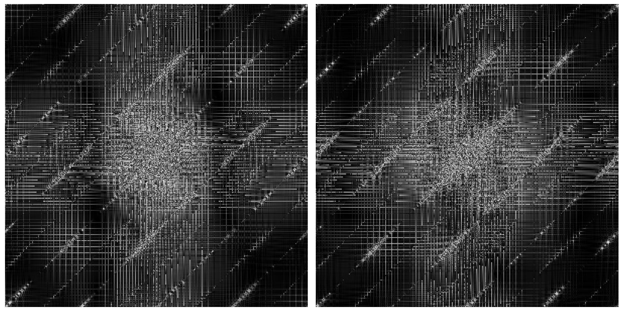
2.3 Fourier: Strips

The real and imaginary components respectively when plotted in image are as follows:



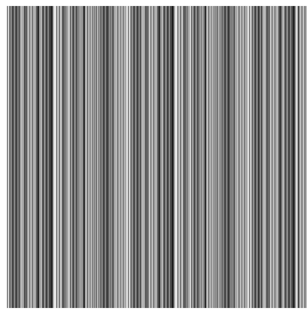
2.4 Fourier: Rotated Strips

The real and imaginary components respectively when plotted in image are as follows:



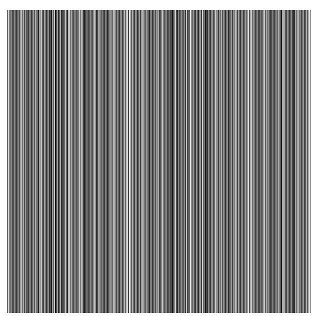
2.5 Low-Pass

Inverse Fourier Transform obtained after multiplying Fourier output of strips image with white circle is:



2.6 High-pass

Inverse Fourier Transform obtained after multiplying Fourier output of strips image with black circle is:



2.7 Observations

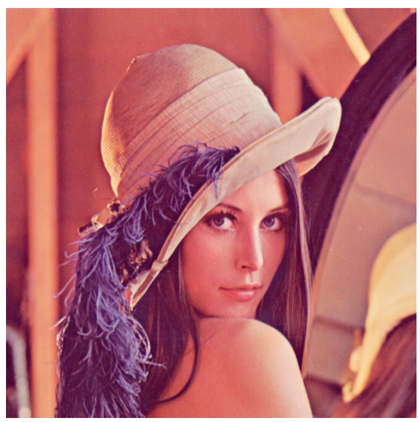
- We are able to see that most of the nonzero coefficients are present around center in frequency domain
- We are able to observe that most of the high frequency component(edges) are retained in output image which we get by applying high pass Black Circle
- We are able to see that the edges get smoothed when we applied low pass white circle

3 Part 3: DCT

All the images in this section are scaled by factor of 3

3.1 Sample images

lena gray scale



lena RGB

3.2 Gray Scale

a and b

c and d

e

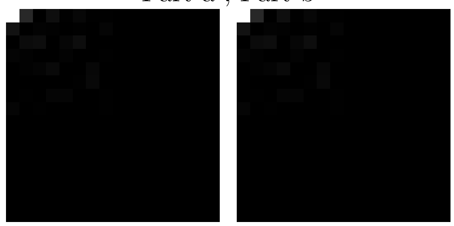
S No.	image	Block size	max coeff. kept	MSE/pixel
1	a	8	5	88.27
2	b	8	10	70.85
3	c	16	10	231.40
4	d	16	40	99.83
5	e	16	150	14.99

```
Mask a:
[[1. 1. 1. 0. 0. 0. 1. 0.]
 [1. 0. 1. 0. 0. 0. 0. 0.]
 [0. 1. 1. 1. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]]
Mask b:
[[1. 1. 1. 0. 0. 0. 0. 0.]
 [1. 0. 1. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0.]]
Mask c:
[[1. 1. 0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```

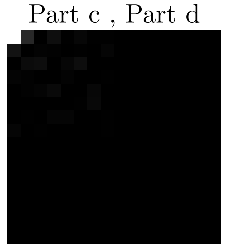
```
 [0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
Mask d:
[[1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 1. 0. 0. 0.]
 [1. 1. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 1. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0.]
 [1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
Mask e:
[[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 0.]
 [1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 0. 0. 1.]
 [1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1.]
 [1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 0. 0.]
 [1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1.]
 [1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 1. 1. 1. 0. 0. 1.]
 [1. 1. 1. 0. 1. 1. 1. 1. 1. 1. 0. 0. 0. 0. 1. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 1.]
 [1. 0. 1. 1. 0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 1.]
 [0. 0. 1. 0. 0. 1. 1. 1. 0. 1. 1. 1. 0. 0. 0. 0.]
 [1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0.]
 [0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 1. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0.]]
```

Average Energy

Part a , Part b



Part c , Part d



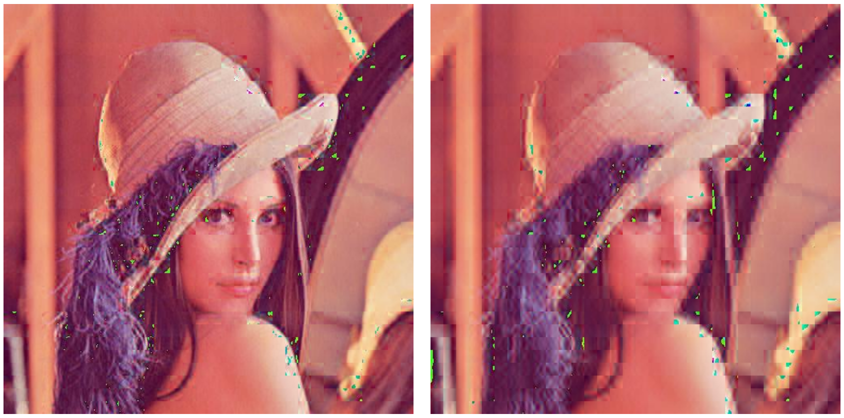
Part e

Observations :

- In part a and b not much information has been lost and part b is slightly better than part a (Also MSE per pixel of b is less than a), but a has better compression ratio than b as it is keeping only 5 coefficients in each block
- Part c has lost much information
- Part a and b are better than part d
- Part e has the best performance , loosing very less information but it has least compression ratio

3.3 Extra Credit : Coloured

a and b



c and d

e and f

S No.	image	Block size	max coeff.	MSE/pixel
1	a	8	5	379.8
2	b	8	10	452.24
3	c	16	10	509.62
4	d	16	40	96.97
5	e	8	30	913.3
6	f	16	150	107.5

4 References

- https://www.math.cuhk.edu.hk/ lmlui/dct.pdf
- https://en.wikipedia.org/wiki/Infinite_impulse_response
- https://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm