CZ4003 COMPUTER VISION - 2020

Lab 1: Point Processing + Spatial Filtering + Frequency Filtering + Imaging Geometry

by
Barnabas Lim
U1722494H

School of Computer Science and Engineering
50, Nanyang Avenue, Singapore 639798, Tel:65 790 5488 Fax:65 792 4062

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Contents

1.1 2.1 Contrast Stretching (*2.1 Part a, b, c, d, e*)

1.1.1 Read and Evaluate Image (*2.1 Part a*)

```
clear all;
Pc = imread('mrttrainbland.jpg');
whos Pc
P = rgb2gray(Pc);
whos P
```

Figure 1: Code to Read and evaluate image

```
Name      Size          Bytes  Class    Attributes

Pc        320x443x3     425280  uint8

Name      Size          Bytes  Class    Attributes

P         320x443       141760  uint8
```

Figure 2: Results of evaluated Image

1.1.2 Code for Contrast Stretching (*2.1 Part b, c, d, e*)

```
% 2.1 Part c)
min_P=double(min(P(:)))%13
max_P=double(max(P(:)))%204

% 2.1 Part d) contrast stretching
P2 = (double(P(:,:))-min_P).*(255/(max_P-min_P));
```

*Figure 3: Contrast Stretching Code implementation (**2.1 Part d**)*

$$S_r = \frac{255 * (r - r_{min})}{r_{max} - r_{min}}$$

Equation 1: Contrast Stretching

Using Equation 1, we were able to carry out contrast stretching. The concept of contrast stretching has similar fundamental principles to histogram equalisation. In both point processing technique, they aim to utilize the maximum range of grey level available (spreading out the histogram). Also, both point processing techniques (contrast stretching and histogram equalization) maintains the relative Grey level of the pixels.

The main difference between contrast stretching and histogram equalization is that contrast stretching is linear while histogram equalisation is nonlinear. Another difference is that contrast stretching does not try to evenly spread out the histogram but rather scale the grey levels to occupy the maximum available grey levels

1.1.3 Results & Evaluation for Contrast Stretching (*2.1 Part b, c, d, e*)

*Figure 4: Original Image (**2.1 Part b**)*

*Figure 5: Contrast Stretched Image (**2.1 Part e**)*

Table 1: Min Max Grey levels (original)

*(**2.1 Part c**)*

Min	13
Max	204

Table 2: Min Max Grey levels (Contrast Stretching)

*(**2.1 Part d**)*

Min	0
Max	255

As you can see after contrast stretching, there is larger contrast in the image compared to the original. In terms of grey levels, they have been scaled to occupy the maximum range of grey level such that the Min = 0 and Max = 255.

1.2 2.2 Histogram Equalization (*2.2 Part a, b, c*)

1.2.1 Code for Histogram Equalization (*2.2 Part b*)

<table>
<tr>
<td>

```
% 2.2 Part b) Histogram Equalization
P3 = histeq(P,255);
```

Figure 6: Histogram equalization Code implementation

(***2.2 Part b***)

</td>
<td>

$$S_k = \frac{(L-1)}{MN}\sum_{j=0}^{k} P_r(j)$$

Equation 2: Histogram Equalization

</td>
</tr>
</table>

1.2.2 Results & Evaluation for Histogram Equalization (Histogram) (*2.2 Part a, b, c*)

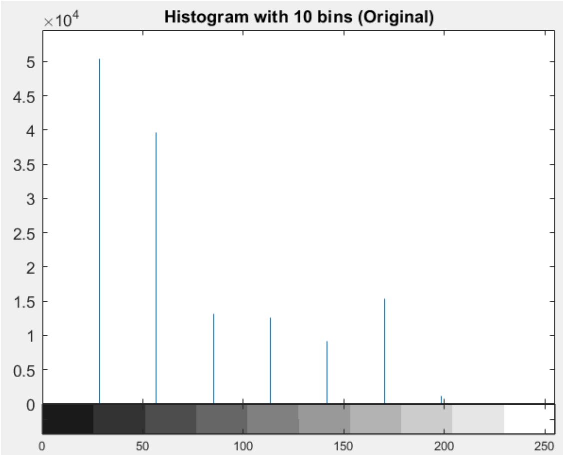


*Figure 7: Grey Level Histogram 10 bins (Original) (**2.2 Part a**)*

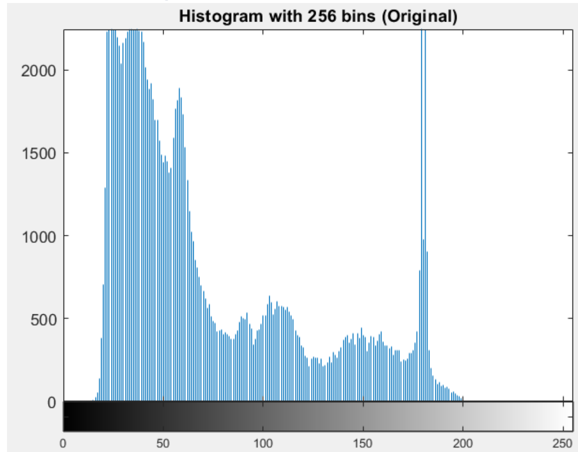


*Figure 8: Grey Level Histogram 256 bins (Original)(**2.2 Part a**)*

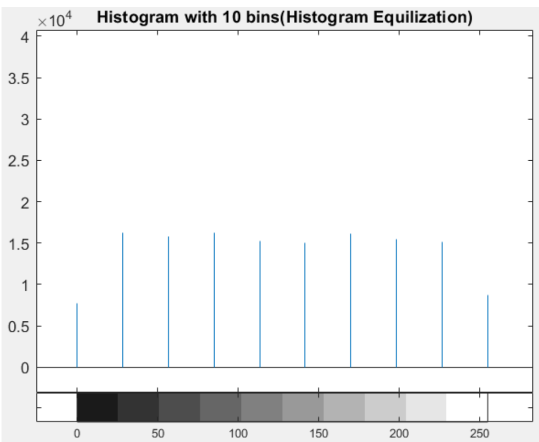


*Figure 9: Grey Level Histogram 10 bins (Histogram Equalization) (**2.2 Part b**)*

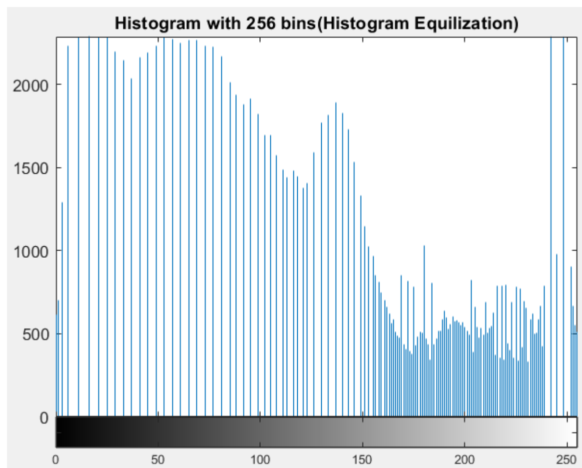


*Figure 10: Grey Level Histogram 256 bins (Histogram Equalization) (**2.2 Part b**)*

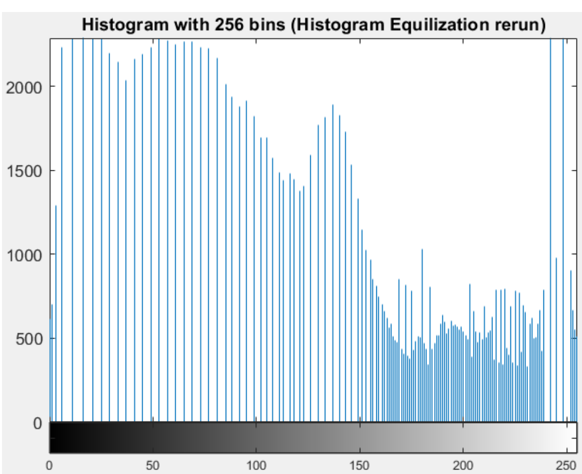


*Figure 11: Grey Level Histogram 256 bins (Histogram Equalization rerun) (**2.2 Part c**)*

1.2.2.1 What are the differences between 10 bins (Figure 7) and 256bins (Figure 8)? (2.2 Part a)

Difference is that there are more bins. For 10 bins, each bin contains greyscale of range 25, while for 256 bins, each bin contains greyscale of range 1.
Another difference is regarding the frequency in each bin. For 10 bins, the larger number of a bin ~5E+4, while for 10 bins, the larger number of a bin 2935.

1.2.2.2 Are the histograms equalized? (2.2 Part b)

Yes. The histogram is equalized as observed by the histogram being more evenly spread out from 0 to 255. This is even more evident in the 10-bin histogram (Figure 9) of the histogram equalized image. As you can see the 10 bin histogram (Figure 9) is much more even compared to before the histogram equalization (Figure 7).

1.2.2.3 What are the similarities and differences between the latter two histograms? (2.2 Part b)

The 10-bin histogram after histogram equalization (Figure 9) and the 256-bin histogram after histogram equalization (Figure 10) are similar in that both occupy the maximum range of grey level available from 0 to 255.

The main difference is regarding the frequency of each bin. For 10-bin histogram (Figure 9), the frequency for each bin is around the range of 1.5E+04 while For 256-bin histogram (Figure 10), the frequency for each bin is around the range of 500 to 3000.

Another difference is that spacing of each bin. For 10-bin histogram (Figure 9), the bins are evenly spaced out while for 256-bin histogram (Figure 10), the bins are more spaced out for grey levels between 0 to 150 when the frequency is high between 1000 to 3000 and the bins are less spaced out for grey levels between 150 to 240 when the frequency is low at around 500.

1.2.2.4 What are the similarities and differences between the 256-bin histograms before and after histogram equalization (EXTRA)?

The similarities are that the maximum number among the bin is still 2953. This is expected because the larger bins will likely not be combined because it already has a number larger than the target height of 555.92 (MN/255). Also, if is important to note that bins can only combined but cannot be split.

Another similarity is that the relative order of the pixel grey levels remains relatively intact. This means that if old n^{th} pixel Grey level< old m^{th} pixel Grey level, then new n^{th} pixel grey level <= old m^{th} pixel Grey level. As such, at a glance, the general shape of the histograms before equalization and after seems to be preserved

The difference between the two histogram is that the bins are more spread out after the equalisation. The bins range from Grey level 0 to Grey level 255. While before equalisation the bins range from Grey level 13 to Grey level 204.

Another difference is that after equalisation there is more empty bins between each filled bin as compared to before equalisation.

*1.2.2.5 Does the histogram become more uniform? Give suggestions as to why this occurs. (**2.2 Part c**)*

No. In fact the histogram remains the same after redoing histogram equalisation. This is because the bins have already been shifted based on their cumulative probability. Rerunning histogram equalisation will only allocate the bins to the same bins.

1.2.3 Results & Evaluation for Histogram Equalization (Image) (*2.2*)

*Figure 12: Original Image (**2.2)***

*Figure 13: Histogram Equalized Image (**2.2)***

*Figure 14: Histogram Equalized Image Rerun (**2.2)***

As you can see after histogram equalization, there is larger contrast in the image compared to the original. In terms of Grey levels, they have been scaled to occupy the maximum range of Grey level such that the Min = 0 and Max = 255.

As you can see there is no difference after a rerun of histogram equalization. This is because the bins have already been shifted based on their cumulative probability. Rerunning histogram equalisation will only allocate the bins to the same bins.

The image quality appears to be better than the results of contrast stretching.
A possible explanation for this is because histogram equalization reduces complexity of the image by joining grey levels that appear less frequently with their neighbouring Grey levels. Theoretically this will may lead to some loss in detail, but this effect may not be obvious because these grey levels occurs less frequently.

Another possible explanation is that since histogram equalization allocates new grey levels based on cumulative probability as compared to by linear scaling, it can more evenly spread out the grey levels in the histogram. This is evident by the fact that even though both histogram equalization and contrast stretching can preserve the relative **grey level order** of each pixel, histogram equalization does not preserve the relative **grey level difference** of each pixel while contrast stretching can because contrast stretching is a linear transformation while histogram equalization is a non-linear transformation.

1.3 2.3 Linear Spatial Filtering (*2.3 Part a i) ii), c, e*)

1.3.1 Code for Generating 5x5 Gaussian Filter (*2.3 Part a i) ii)*)

```
% 2.3 Part a) Gaussian Filters
h=@(X,Y,Sigma) exp(-(X.^2.+Y.^2)./(Sigma^2.*2))./(2*pi*Sigma^2);
sizes=5
x=-(sizes-1)/2:(sizes-1)/2;y=-(sizes-1)/2:(sizes-1)/2;
[X,Y] = meshgrid(x,y)
% 2.3 Part a)i) Gaussian Filters σ = 1.0
h1=h(X,Y,1)
h1=h1./sum(h1,'all')
% 2.3 Part a)ii) Gaussian Filters σ = 2.0
h2=h(X,Y,2)
h2=h2./sum(h2,'all')
```

*Figure 15: Generating Gaussian Filter Code(**2.3 Part a i) ii)**)*

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

Equation 3: Gaussian Filter Equation

1.3.2 View 5x5 Gaussian Filter (*2.3 Part a i) ii)*)

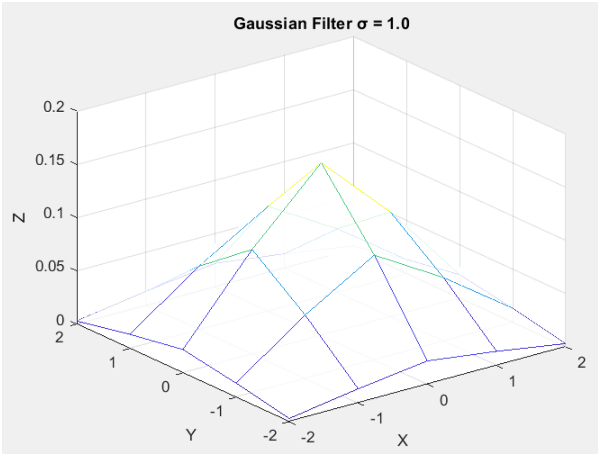


*Figure 16: Gaussian Filter σ = 1.0 (**2.3 Part a i**)*

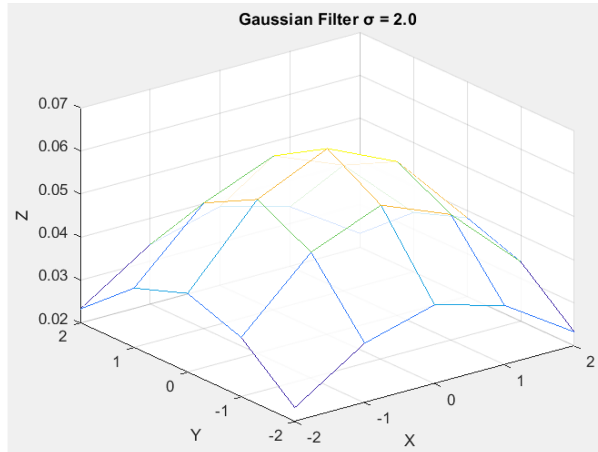


*Figure 17: Gaussian Filter σ = 2.0 (**2.3 Part a ii**)*

1.3.3 Results & Evaluation of gaussian filter on Gaussian Noise (*2.3 Part c*)

*Figure 18: Gaussian Noise Original, Gaussian Filter σ = 1.0, Gaussian Filter σ = 2.0 (Left to Right) (**2.3 Part c**)*

1.3.4 Results & Evaluation of gaussian filter on Speckled Noise (*2.3 Part e*)

*Figure 19: Speckle Noise Original, Gaussian Filter σ = 1.0, Gaussian Filter σ = 2.0 (Left to Right) (**2.3 Part e**)*

1.3.4.1 How effective are the filters in removing speckled noise?

(***2.3 Part c***) The filters are effective at removing the gaussian noise. The higher the σ the better the gaussian noise is removed.

(***2.3 Part e***) The filters can reduce speckled noise to some extend but is not very effective. Higher σ reduces more of the speckled noise.

1.3.4.2 What are the trade-offs between using either of the two filters, or not filtering the image at all?

(***2.3 Part c***) The trade-off is that higher σ removes more gaussian noise but more edges are blurred/ loss as a result.

(***2.3 Part e***) The trade-off is that higher σ removes more speckled noise but the more edges blurred/ loss as a result.

1.3.5 Comparison & Evaluation

1.3.5.1 Are the filters better at handling Gaussian noise or speckle noise?

(***2.3 Part e***) The filters are better at handle Gaussian Noise compared to Speckled Noise

1.4 2.4 Linear Spatial Filtering (*2.4 c, e*)

1.4.1 Results of 3x3 Median filter on Gaussian Noise (*2.4 Part c*)

*Figure 20: Gaussian Noise Original, Median Filter 3X3, Median Filter 5X5 (Left to Right) (**2.4 Part c**)*

1.4.2 Results of 5x5 Median filter on Speckled Noise (*2.4 Part e*)

*Figure 21: Speckle Noise Original, Median Filter 3X3, Median Filter 5X5 (Left to Right) (**2.4 Part e**)*

1.4.3 Comparison & Evaluation

1.4.3.1 How effective are the 3x3 and 5x5 Median filters in removing gaussian noise?

The filters are effective at removing the gaussian noise but are very effective at removing the speckle noise.

For Gaussian noise there is a slight improvement when we move from a 3x3 to 5x5 median filter. For Speckled Noise there is a huge improvement mainly because speckled noise tends to be white dots which are easily filtered by median filters.

1.4.3.2 What are the trade-offs between using either of the two filters, or not filtering the image at all?

Salt and pepper noise or speckled noise can be effectively removed from the images by 3x3 and 5x5 median filters. Edges are well preserved.

However, noise that with grey level that exhibit gaussian distribution will be less effectively removed because gaussian noises tend to be closer to the median. But there are still smoothing of the surface due to gaussian noise.

1.4.3.3 Are the filters 3x3 and 5x5 median filter better at handling Gaussian noise or speckle noise?

The x3 and 5x5 median filters are better at handling Speckled Noise then Gaussian Noise.

1.5 2.5 Suppressing Noise Interference Patterns (*2.5 Part b, c, d, e*)

Figure 22: Original Image

1.5.1 Results of Power Spectrum, S, FFTShift (*2.5 Part b*)

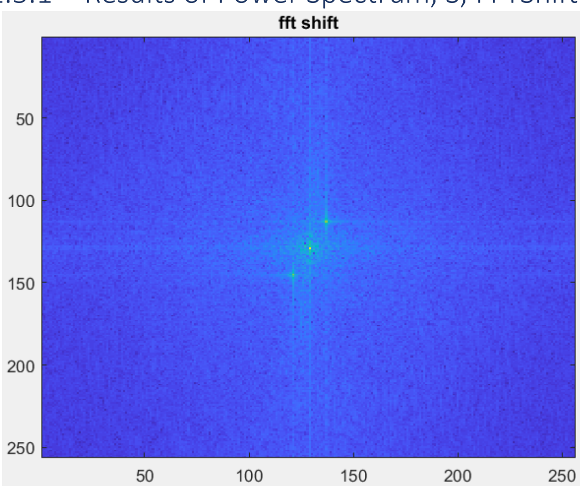


*Figure 23: FFT Shift of power spectrum (**2.5 Part b**)*

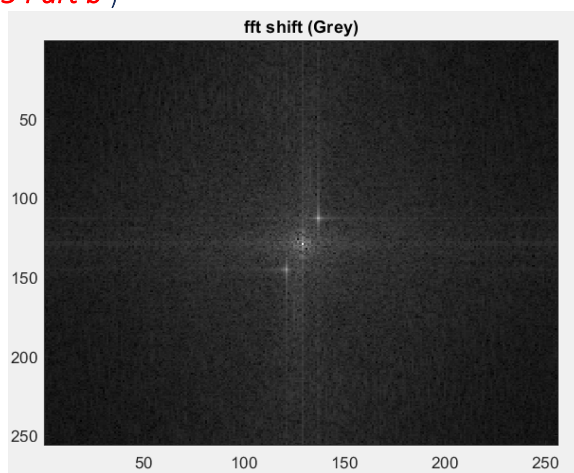


Figure 24: Figure 17: FFT Shift of power spectrum (Grey)

1.5.2 Results of Power Spectrum, S, FFT (*2.5 Part c*)

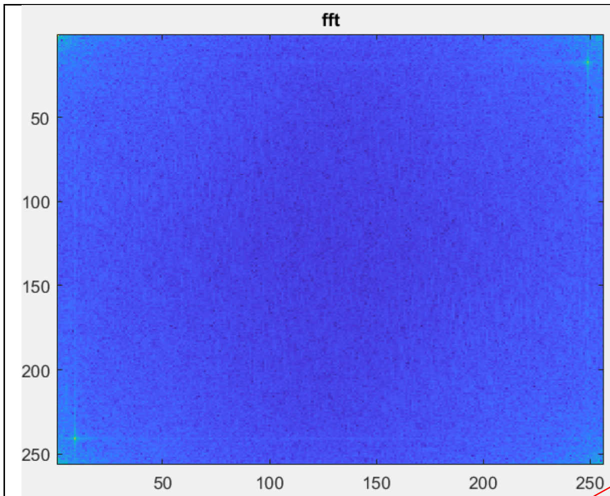


*Figure 25: FFT of power spectrum (**2.5 Part c**)*

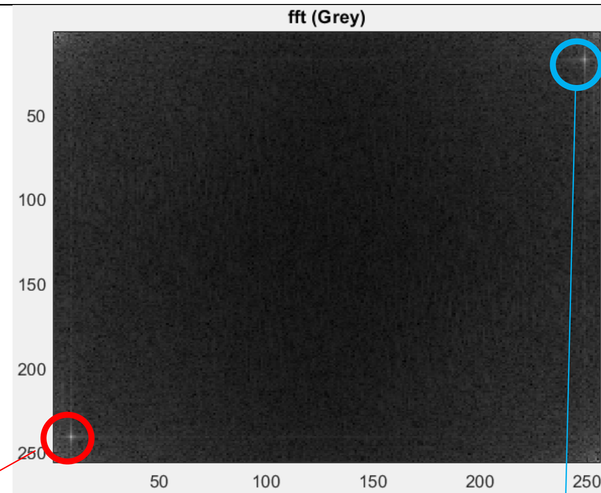


Figure 26: FFT of power spectrum (Grey)

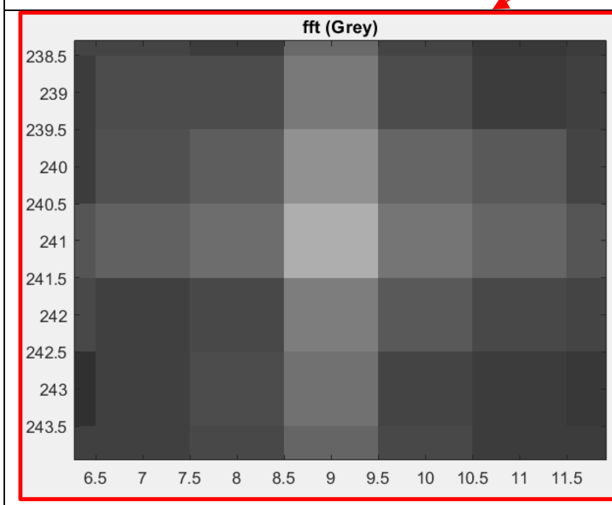


*Figure 27: Bottom left peak (**2.5 Part c**)*

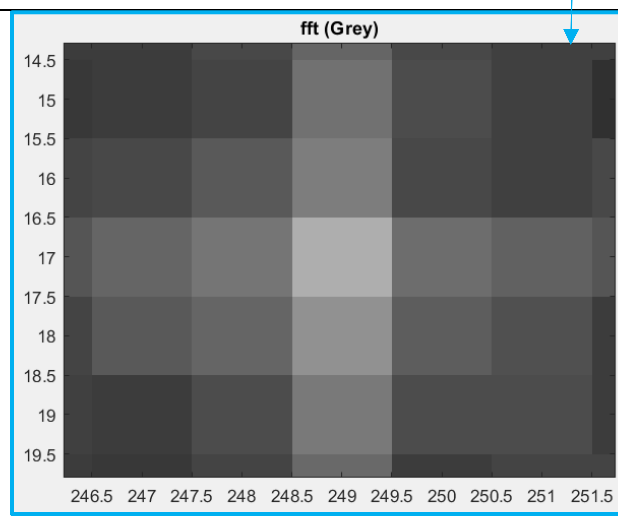


*Figure 28: Top right peak (**2.5 Part c**)*

*Table 3: 5x5 Spectrum filter (Bottom left) (**2.5 Part c**)*

u	[243,239]
v	[7,11]

*Table 4: 5x5 Spectrum filter (top right) (**2.5 Part c**)*

u	[15,19]
v	[247,251]

1.5.3 Filtering specific patterns using Fourier transform, F (*2.5 Part d*)

```
F(15:19,247:251)=0;
F(239:243,7:11)=0;
```

Figure 29: Code to filter specific patterns using amplitude spectrum

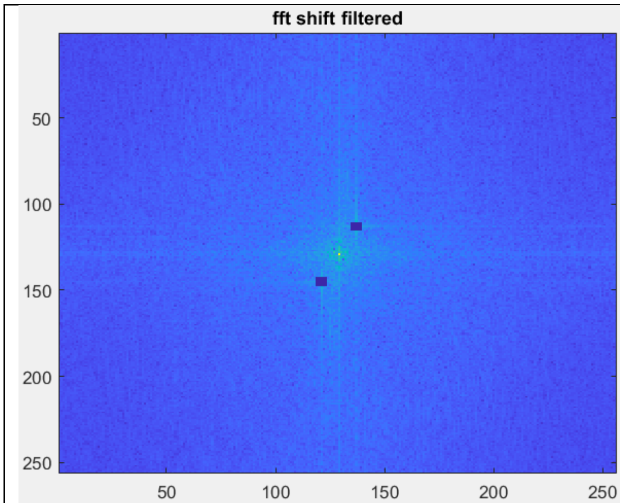


*Figure 30: FFT Shift of power spectrum (Filtered) (**2.5 Part d**)*

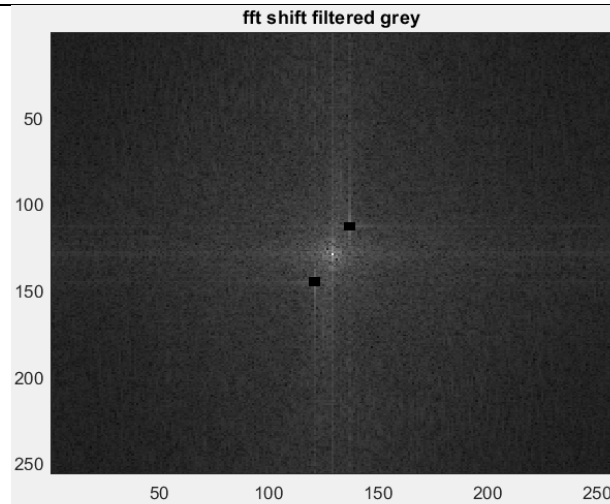


Figure 31: FFT Shift of power spectrum (Filtered)(Grey)

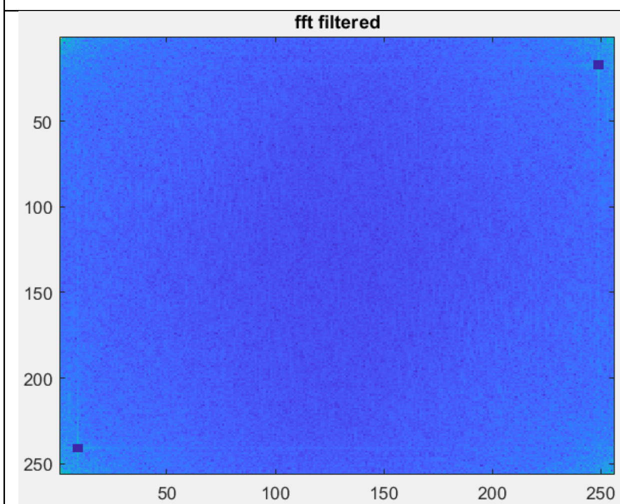


*Figure 32: FFT of power spectrum (Filtered) (**2.5 Part d**)*

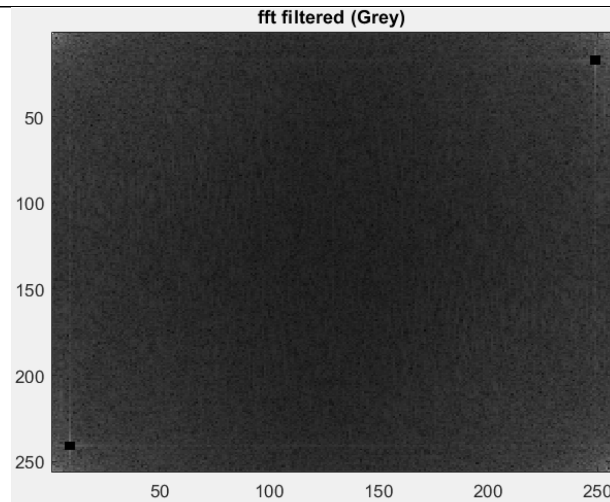


Figure 33: FFT of power spectrum (Filtered)(Grey)

1.5.4 Results & Evaluation of Filtering specific patterns using Fourier transform, F (*2.5 Part e*)

```
% 2.5 Part e) Compute the inverse Fourier transform using ifft2 and display the resultant image
figure;colormap('gray');imshow(uint8(ifft2(F)))
```

Figure 34: Inverse Fourier transform to generate filtered image

Figure 35: Original Image

*Figure 36: Filtered Image (**2.5 Part e**)*

1.5.4.1 *Comment on the result and how this relates to step (c). Can you suggest any way to improve this? (2.5 Part e)*

As shown in Figure 36, by setting F(u,v)=0 for the two peaks, we are able to remove most of the interference pattern. This is because when we set F(u,v)=0 in step c we are removing a specific sets of building block ('atom') (e.g. Figure 37) of the image that corresponds to the interference patterns. When we do an inverse Fourier Transform the specific atom with "weight" =0 will have zero influence on the image. We can improve this process by filtering out other peaks in the frequency domain refer to "Improvement Attempts ***(2.5 Part** e)*".

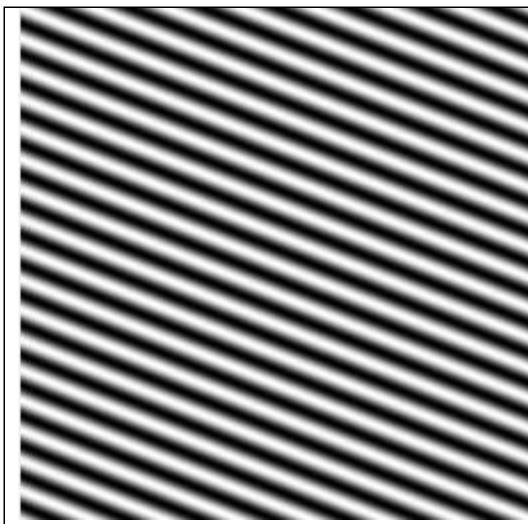


Figure 37: Atom $\alpha_{(u,v)}\ (x,y)$ M=N=256, u=17 v=249

"Weight" (Complex number)

atom $\alpha_{(u,v)}(x,y)$

Synthesis

$$f(x,y) = \sum_{v=0}^{N-1}\sum_{u=0}^{M-1} \boxed{F(u,v)}\boxed{\exp\left(j2\pi\left(\frac{xu}{M}+\frac{yv}{N}\right)\right)}$$

Analysis

$$F(u,v) = \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} f(x,y)\exp\left(-j2\pi\left(\frac{xu}{M}+\frac{yv}{N}\right)\right)$$

1.5.5 Improvement Attempts (*2.5 Part e*)

In this section, we attempt to improve the filtering by identifying other peaks in the power spectrum (Figure 38). A total of 4 attempts were made. Filter V2 (Figure 39) and Filter V3 (Figure 39) attempts to improve the filter by removing other peaks in the power spectrum. Filter 3 extends the range of filter. Filter V5 (Figure 39) and 6 (Figure 39) attempts combines results from Filter V3 with a circular band filter on the shifted Fourier Transform trying to remove atoms with the similar frequency.

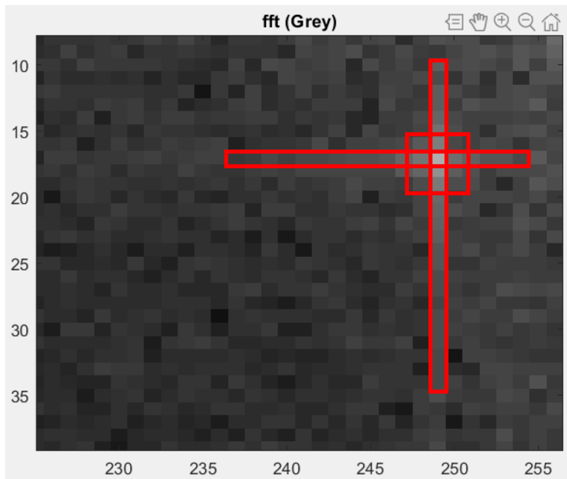
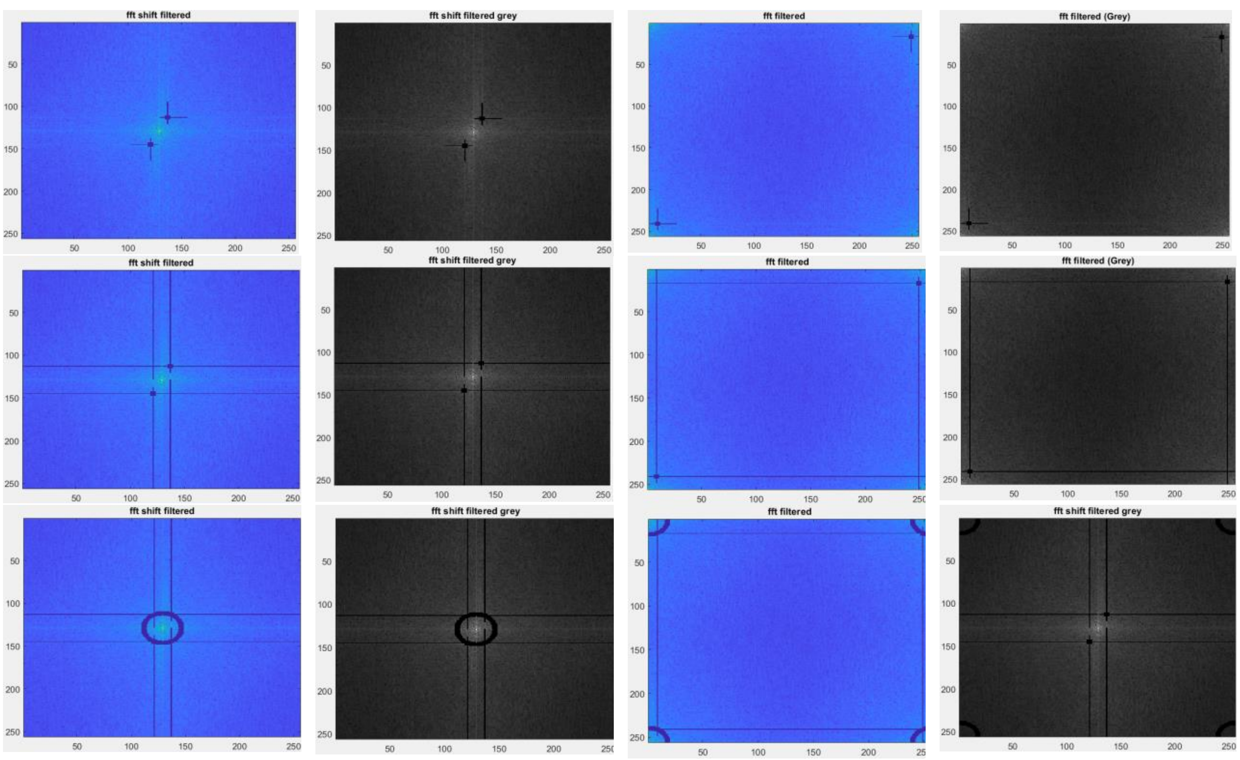


Figure 38: FFT top right Bright points



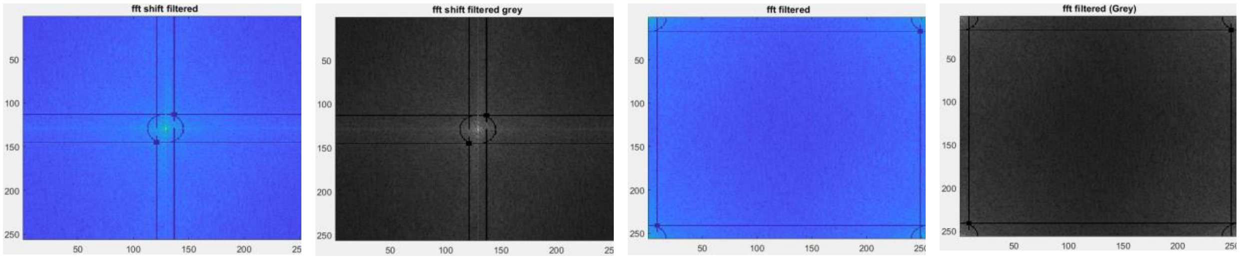


Figure 39: fft shift filter v2, fft filter v2 (1st row left to right), fft shift filter v3 , fft filter v3 (2nd row left to right)

*fft shift filter v5, fft filter v5 (3st row left to right), fft shift filter v6 , fft filter v6 (4nd row left to right) (**2.5 Part e**)*

1.5.6 Improvement Attempts Results (*2.5 Part e*)

*Table 5: Image Results using various filters (**2.5 Part e**)*

 *Figure 40: Original Image*	 *Figure 41: Filtered Image*	 *Figure 42: Filtered Image V2*	 *Figure 43: Filtered Image V3*
 *Figure 44: Filtered Image V5*	 *Figure 45: Filtered Image V6*		

By observation, we can see that resultant image from filter V2 (Figure 43)and V6 (Figure 45) is the best.

1.6 "Free" the primate by filtering out the fence (*2.5 Part f*)

Figure 46: Original Image

1.6.1.1 FFT and FFTshift spectrum analysis

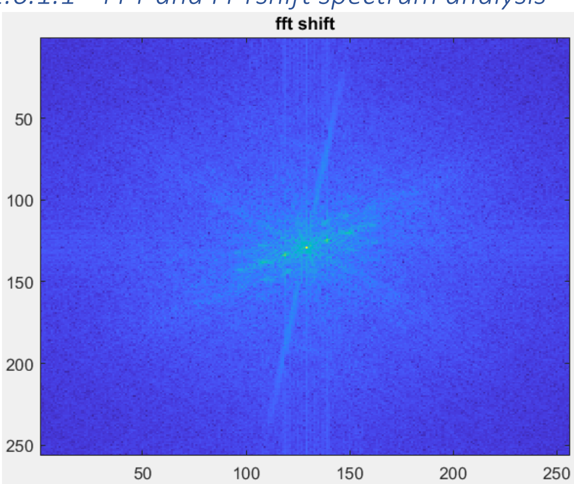


*Figure 47: FFT Shift of power spectrum (**2.5 Part f**)*

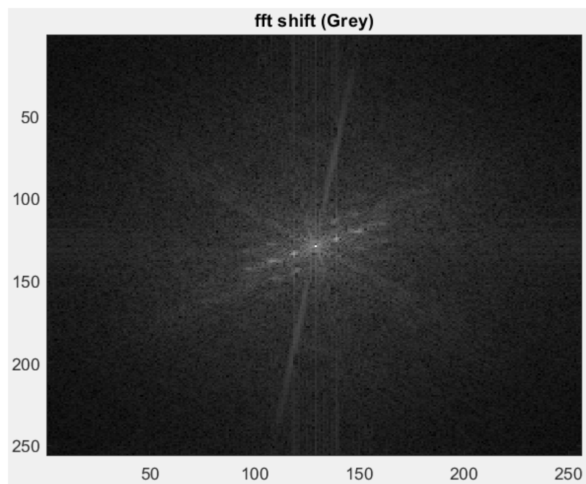


Figure 48: Figure 17: FFT Shift of power spectrum (Grey)

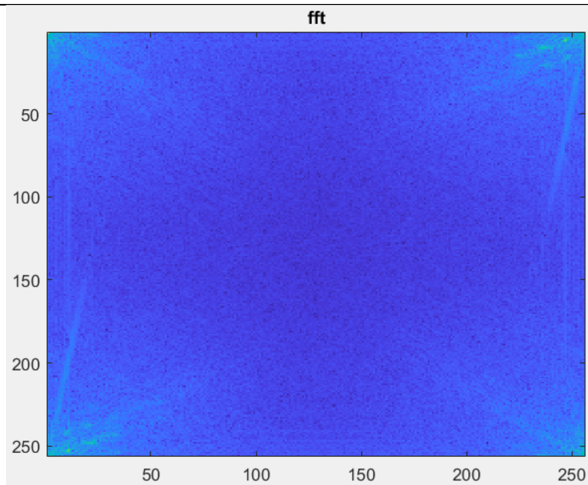


*Figure 49: FFT of power spectrum (**2.5 Part f**)*

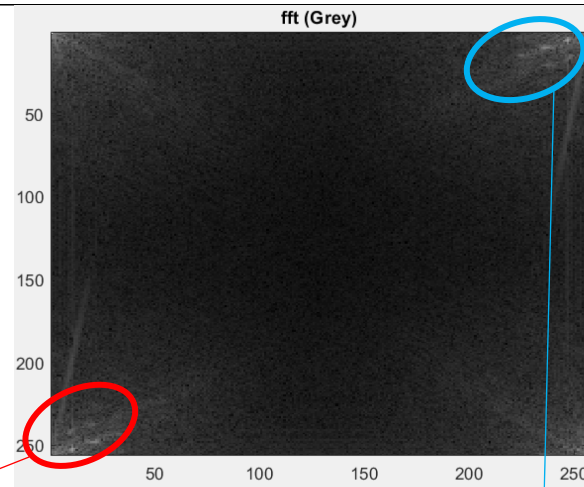


Figure 50: FFT of power spectrum (Grey)

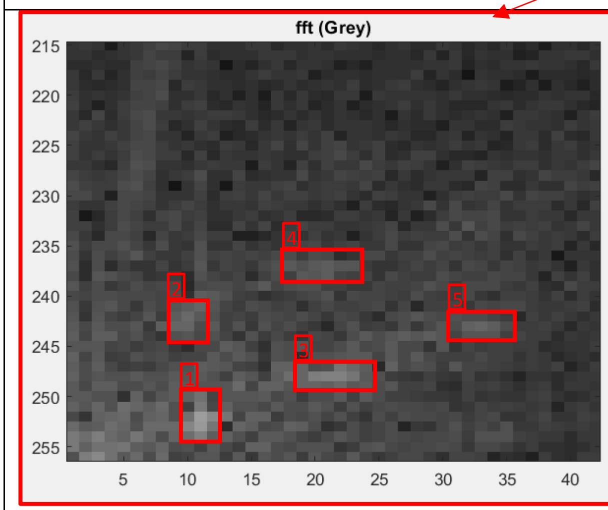


*Figure 51: Bottom left peak (**2.5 Part f**)*

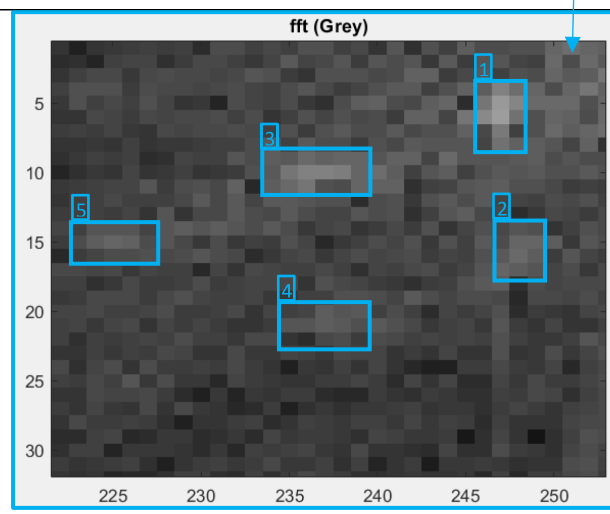


*Figure 52: Top right peak (**2.5 Part f**)*

*Table 6: 5x5 Spectrum filter (Bottom left) (**2.5 Part f**)*

U1	[250,254]
V1	[10,12]
U2	[241,244]
V2	[9,11]
U3	[247,249]
V3	[19,24]
U4	[236,238]
V4	[18,23]
U5	[242,244]
V5	[31,35]

*Table 7: 5x5 Spectrum filter (top right) (**2.5 Part f**)*

U1	[4,8]
V1	[246,248]
U2	[14,17]
V2	[247,249]
U3	[9,11]
V3	[234,239]
U4	[20,22]
V4	[235,239]
U5	[14,16]
V5	[223,227]

1.6.1.2 Filtering Attempts

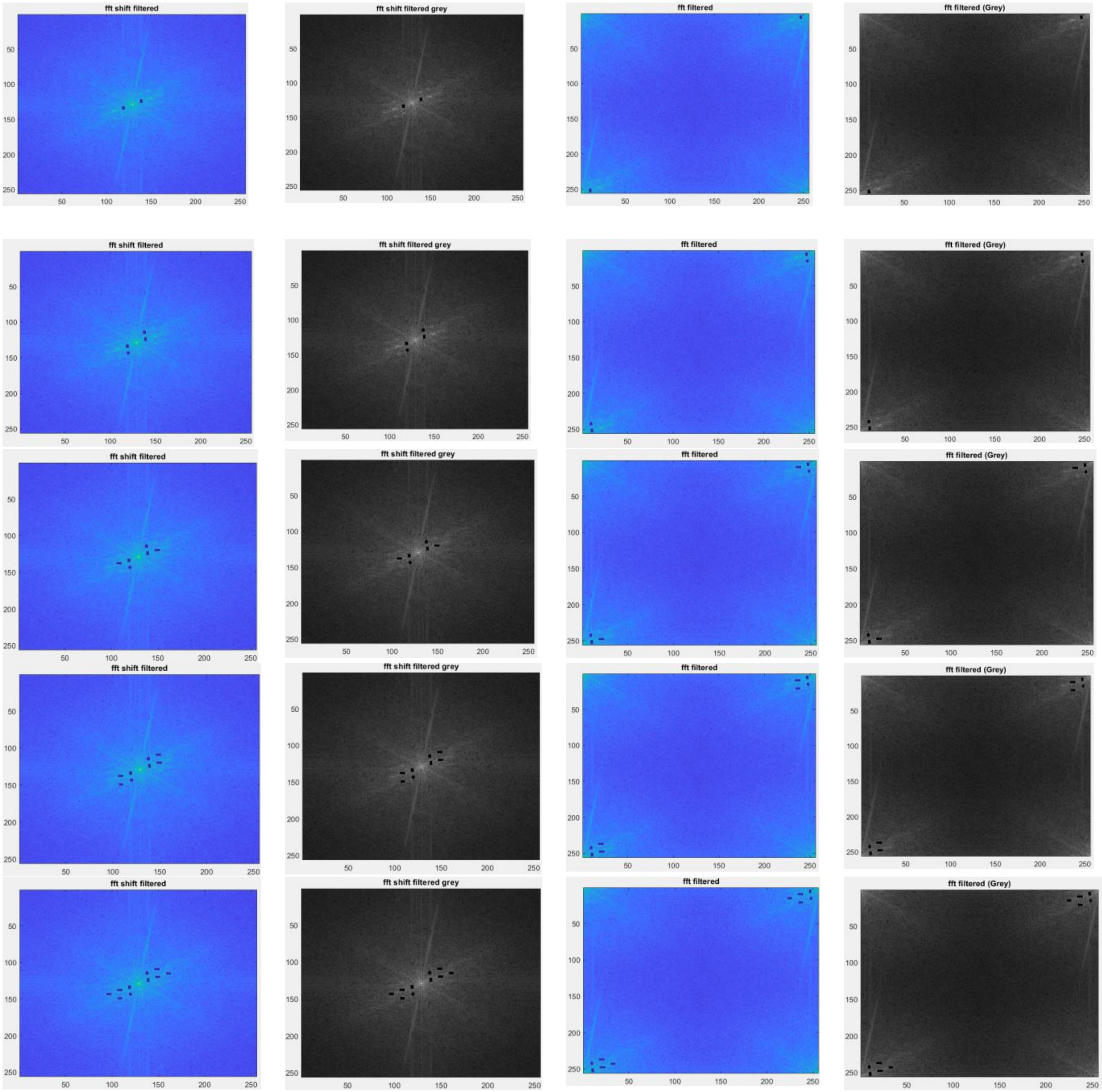


Figure 53: fft shift filter 1, fft filter 1 (1st row left to right), fft shift filter 2 , fft filter 2 (2nd row left to right)

fft shift filter 3, fft filter 3 (3st row left to right), fft shift filter 4 , fft filter 4 (4nd row left to right),

fft shift filter 5 , fft filter 5 (5nd row left to right) (**2.5 Part f**)

1.6.1.3 Results

*Table 8: Image Results using various filters (**2.5 Part f**)*

 *Figure 54: Original Image*	 *Figure 55: Filtered Image 1*	 *Figure 56: Filtered Image 2*	 *Figure 57: Filtered Image 3*
 *Figure 58: Filtered Image 4*	 *Figure 59: Filtered Image 5*		

By Observation, Filter image 3 (Figure 57) has the best trade-off between gate removal and detail retention. While Filtered image 5 (Figure 59) has the best removal of gate but with slightly more detail lost.

1.7 Undoing Perspective Distortion of Planar Surface (*2.6*)

$$\boxed{\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}}$$

Equation 4: 2D planar projective transform

$$\boxed{x_{im} = \frac{m_{11}X_w + m_{12}Y_w + m_{13}}{m_{31}X_w + m_{32}Y_w + 1}, \mathrm{y}_{im} = \frac{m_{21}X_w + m_{22}Y_w + m_{23}}{m_{31}X_w + m_{32}Y_w + 1}}$$

Equation 5: Alternative form of projective transform

X_w and Y_w are the coordinates of a point in the input image

x_{im} and y_{im} are the coordinates of the transformed point in the output image.

AU=V

$$\begin{bmatrix} X_w^1 & Y_w^1 & 1 & 0 & 0 & 0 & -x_{im}^1 X_w^1 & -x_{im}^1 Y_w^1 \\ 0 & 0 & 0 & X_w^1 & Y_w^1 & 1 & -y_{im}^1 X_w^1 & -y_{im}^1 Y_w^1 \\ X_w^2 & Y_w^2 & 1 & 0 & 0 & 0 & -x_{im}^2 X_w^2 & -x_{im}^2 Y_w^2 \\ 0 & 0 & 0 & X_w^2 & Y_w^2 & 1 & -y_{im}^2 X_w^2 & -y_{im}^2 Y_w^2 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ X_w^n & Y_w^n & 1 & 0 & 0 & 0 & -x_{im}^n X_w^n & -x_{im}^n Y_w^n \\ 0 & 0 & 0 & X_w^n & Y_w^n & 1 & -y_{im}^n X_w^n & -y_{im}^n Y_w^n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{31} \\ m_{32} \end{bmatrix} = \begin{bmatrix} x_{im}^1 \\ y_{im}^1 \\ x_{im}^2 \\ y_{im}^2 \\ . \\ . \\ x_{im}^n \\ y_{im}^n \end{bmatrix}$$

Equation 6: Using correspondence 4 points in input image (corner coordinate) and output image(desired corner coordinates)

Part a,b,c aims to determine the projective transformation unknowns/

<table>
<tr><td></td><td></td></tr>
<tr><td>2.6 Part b) Input coordinates
<code>[X Y] = ginput(4);</code></td><td>Note selection to be made from top left-hand corner in a clockwise direction
x_w^i, y_w^i</td></tr>
<tr><td>2.6 Part b) desired output coordinates
<code>%begins Top left rotate clockwise</code>
<code>X_im=[0,210,210,0];</code>
<code>Y_im=[0,0,297,297];</code></td><td>Book dimensions
210mm X 297mm

x_{im}^i, y_{im}^i</td></tr>
<tr><td>2.6 Part c)

<code>A = [</code>
<code>   [X(1), Y(1), 1, 0, 0, 0, -Xim(1)*X(1), -Xim(1)*Y(1)];</code>
<code>   [0, 0, 0, X(1), Y(1), 1, -Yim(1)*X(1), -Yim(1)*Y(1)];</code>
<code>   [X(2), Y(2), 1, 0, 0, 0, -Xim(2)*X(2), -Xim(2)*Y(2)];</code>
<code>   [0, 0, 0, X(2), Y(2), 1, -Yim(2)*X(2), -Yim(2)*Y(2)];</code>
<code>   [X(3), Y(3), 1, 0, 0, 0, -Xim(3)*X(3), -Xim(3)*Y(3)];</code>
<code>   [0, 0, 0, X(3), Y(3), 1, -Yim(3)*X(3), -Yim(3)*Y(3)];</code>
<code>   [X(4), Y(4), 1, 0, 0, 0, -Xim(4)*X(4), -Xim(4)*Y(4)];</code>
<code>   [0, 0, 0, X(4), Y(4), 1, -Yim(4)*X(4), -Yim(4)*Y(4)];</code>
<code>];</code></td><td>$\begin{bmatrix} X_w^1 & Y_w^1 & 1 & 0 & 0 & 0 & -x_{im}^1 X_w^1 & -x_{im}^1 Y_w^1 \\ 0 & 0 & 0 & X_w^1 & Y_w^1 & 1 & -y_{im}^1 X_w^1 & -y_{im}^1 Y_w^1 \\ X_w^2 & Y_w^2 & 1 & 0 & 0 & 0 & -x_{im}^2 X_w^2 & -x_{im}^2 Y_w^2 \\ 0 & 0 & 0 & X_w^2 & Y_w^2 & 1 & -y_{im}^2 X_w^2 & -y_{im}^2 Y_w^2 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ X_w^n & Y_w^n & 1 & 0 & 0 & 0 & -x_{im}^n X_w^n & -x_{im}^n Y_w^n \\ 0 & 0 & 0 & X_w^n & Y_w^n & 1 & -y_{im}^n X_w^n & -y_{im}^n Y_w^n \end{bmatrix}$</td></tr>
</table>

2.6 Part c) desired output coordinates `v = [Xim(1); Yim(1); Xim(2); Yim(2); Xim(3); Yim(3); Xim(4); Yim(4)];`	$\begin{bmatrix} x_{im}^{1} \\ y_{im}^{1} \\ x_{im}^{2} \\ y_{im}^{2} \\ . \\ . \\ x_{im}^{n} \\ y_{im}^{n} \end{bmatrix}$
2.6 Part c) projective transformation `u = A \ v;`	$u = A^{-1}v$

2.6 Part c)
`U = reshape([u;1], 3, 3)';`
	$U = \begin{Bmatrix} m_{11} & m_{21} & m_{31} \\ m_{12} & m_{22} & m_{32} \\ m_{13} & m_{23} & 1 \end{Bmatrix}^{T} = \begin{Bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{Bmatrix}$
	`U =`
	`1.4324  1.4858  -246.3794`
	`-0.4302  3.5491  -33.8779`
	`0.0001  0.0050  1.0000`
	Figure 60: Projective Transform Matrix

2.6 Part c) `w = U*[X'; Y'; ones(1,4)];`	$w = U * \begin{bmatrix} x(1) & x(2) & x(3) & x(4) \\ y(1) & y(2) & y(3) & y(4) \\ 1 & 1 & 1 & 1 \end{bmatrix}$ $w = \begin{bmatrix} k_1 x_{im}^{1} & k_2 x_{im}^{2} & k_3 x_{im}^{3} & k_4 x_{im}^{4} \\ k_1 y_{im}^{1} & k_2 y_{im}^{2} & k_3 y_{im}^{3} & k_4 y_{im}^{4} \\ k_1 & k_2 & k_3 & k_4 \end{bmatrix}$ $kx_{im}^{i} = m_{11} \times x(i) + m_{12} \times y(i) + m_{13} \times 1$ $ky_{im}^{i} = m_{21} \times x(i) + m_{22} \times y(i) + m_{23} \times 1$ $k_i = m_{31} \times x(i) + m_{32} \times y(i) + 1$ $i = 1,2,3,4$

2.6 Part c)

```
w = w ./ (ones(3,1) * w(3,:));
```

$$w = w./\left\{\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\begin{bmatrix}k_1 & k_2 & k_3 & k_4\end{bmatrix}\right\}$$

$$= w./\begin{bmatrix}k_1 & k_2 & k_3 & k_4\\k_1 & k_2 & k_3 & k_4\\k_1 & k_2 & k_3 & k_4\end{bmatrix}$$

```
%begins Top left rotate clockwise
X_im=[0,210,210,0];
Y_im=[0,0,297,297];
```

```
w =

    0.0000  210.0000  210.0000    0.0000
         0    0.0000  297.0000  297.0000
    1.0000    1.0000    1.0000    1.0000
```

Does the transformation give you back the 4 corners of the desired image? **(2.6 Part c)**

Yes

2.6 Part d)

```
T = maketform('projective', U');
P2 = imtransform(P, T, 'XData', [0 210], 'YData', [0 297]);
```

Figure 61: Results