

Project 3

Image Enhancement in Frequency Domain

Md Zobaer Islam

PhD student, Oklahoma State University

CWID: A20270547



Part 1: Homomorphic Filter

Letter.bmp – MATLAB code for homomorphic filtering

```
clc; clear; close all;
fnts=20;
I=imread('letter.bmp');
[X, Y]=size(I);
ID=im2double(I);
IL = log(ID+1);
JF=fft2(IL); JFS=fftshift(JF); FS=JFS;
D0 = 15; yH = 2.5; yL = .46; c = 1;
for i=1:X
    for j=1:Y
        D=(i-(X/2+1))^2+(j-(Y/2+1))^2;
        H(i,j) = (yH-yL)*(1-exp(-c*(D^2/D0^2)))+yL;
    end
end
JFS=FS.*H;
JK=real(ifft2(ifftshift(JFS)));
JK = exp(JK)-1;
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
figure(1); imshow(I); title('Original Image', 'FontSize', fnts);
figure(2);imshow(JFR); title('Filtered Image', 'FontSize', fnts);
```

$$H(u, v) = (\gamma_H - \gamma_L) \left(1 - e^{-c(D^2(u,v)/D_0^2)} \right) + \gamma_L$$

Parameters used (after trial and error with different values):

$D_0=15$, $\gamma_H=2.5$, $\gamma_L=0.46$, $c = 1$

Letter.bmp

Parameters $D_0=15$, $\gamma_H=2.5$, $\gamma_L=0.46$, $c=1$

Original Image

Ind Ninety Six between Stockley
of Knox And State of Tennessee
Andrew Jackson of the County
of Alton aforesaid of the other part
Paid Stockley Donelson for a
sum of two thousand
and paid the receipt where
hath and by these presents
I alien enfeoff and confirm
Jackson his heirs and
certain traits or parcels of Land
and acres one thousand acre
more or less give and his

Filtered Image

Indixity Six between Stockley
of Knox. And State of Tennessee
Andrew Jackson of the County
State aforesaid of the other part
Paid Stockley Donelson for a
sum of two thousand
and paid the receipt thirt
eighth day by these presents
Full alien enfeoff and confirm
Jackson his heirs and
certain traits or parcels of La
and acres one thousand acre
more or less five and six

Tunnel.bmp – MATLAB code for homomorphic filtering

$$H(u, v) = (\gamma_H - \gamma_L) \left(1 - e^{-c(D^2(u,v)/D_0^2)} \right) + \gamma_L$$

```
clc; clear; close all;
fnts=20;
I=imread('tunnel.bmp');[X, Y]=size(I);
ID = im2double(I);
IL = log(ID+1);
JF=fft2(IL); JFS=fftshift(JF); FS=JFS;
D0 = 15; yH = .85; yL = .6; c = 1;
for i=1:X
    for j=1:Y
        D=(i-(X/2+1))^2+(j-(Y/2+1))^2;
        H(i,j) = (yH-yL)*(1-exp(-c*(D^2/D0^2)))+yL;
    end
end
JFS=FS.*H;
JK=real(ifft2(ifftshift(JFS))); JK = exp(JK)-1;
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
figure(1);imshow(I);title('Original Image', 'FontSize',fnts);
figure(2);imshow(JFR);title('Filtered Image', 'FontSize',fnts);
```

Parameters used (after trial and error with different values):

$D_0=15$, $\gamma_H=0.85$, $\gamma_L=0.6$, $c=1$

Tunnel.bmp

Parameters $D_0=15$, $\gamma_H=0.85$, $\gamma_L=0.6$, $c = 1$

Original Image



Filtered Image



The filtered image has more uniform intensity, thus more soothing to eye. In addition to that, more details of the darker parts are now visible.

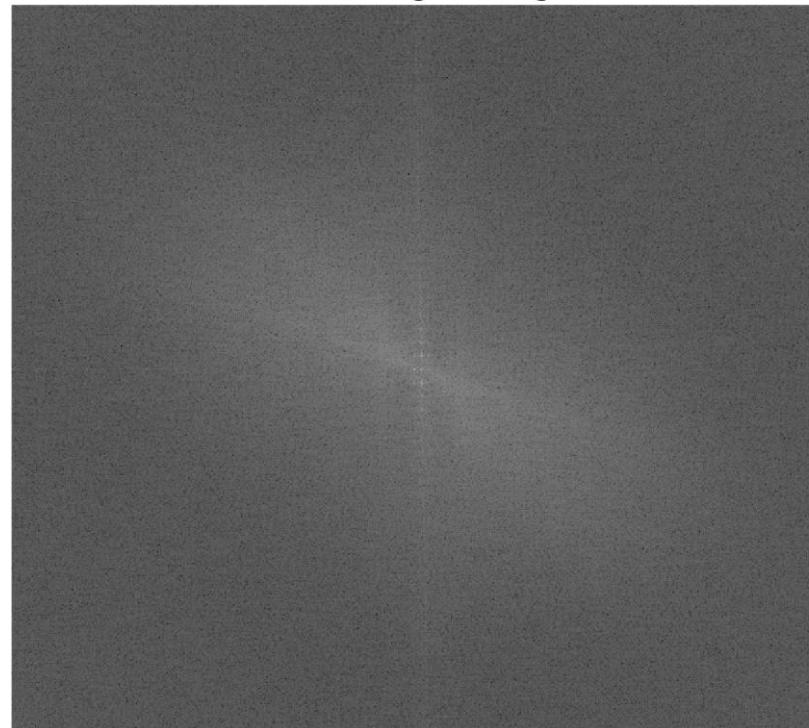
Part 2: Frequency Selective Filtering

Letter_bar.bmp

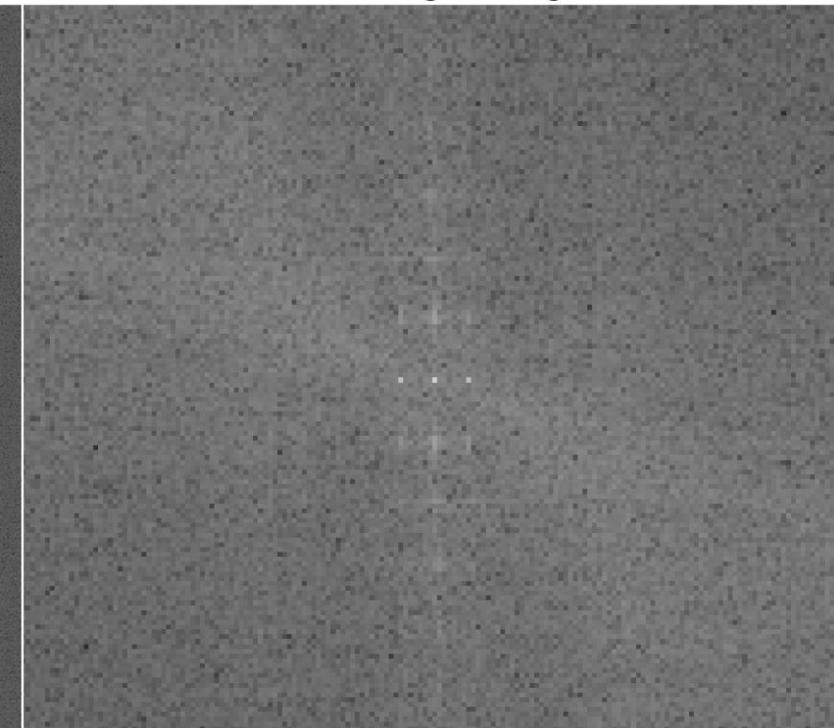
Original Image

Indubtly Six between Stockley
of Knx. And State of Tennessee
Andrew Jackson of the County
that afforsaid of the other part
paid Stockley Donelson for a
of the sum of two thousand
and paid the receipt where
 hath and by these presents
all alien encoff and confirm
Jackson his heirs and a
certain tract or parcels of la
and acres one thousand acre
more or less and his
successors and his

FFT of Original Image



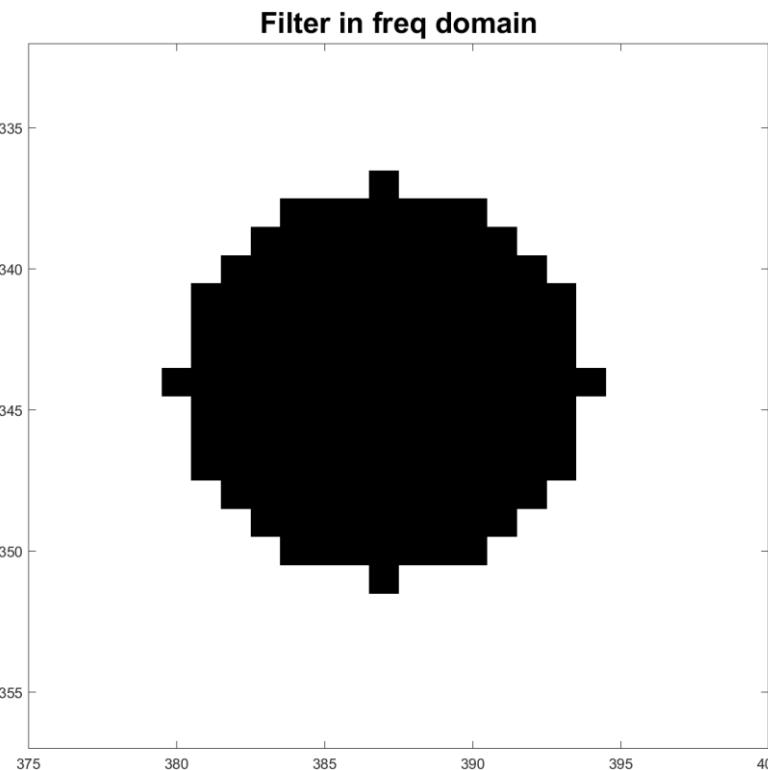
FFT of Original Image



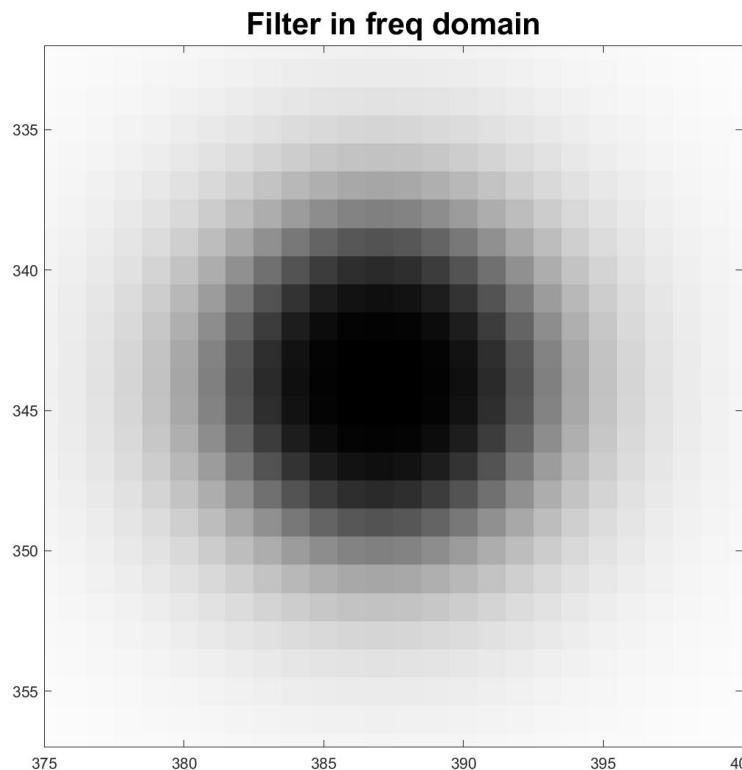
Zoomed in

Letter_bar.bmp - filters

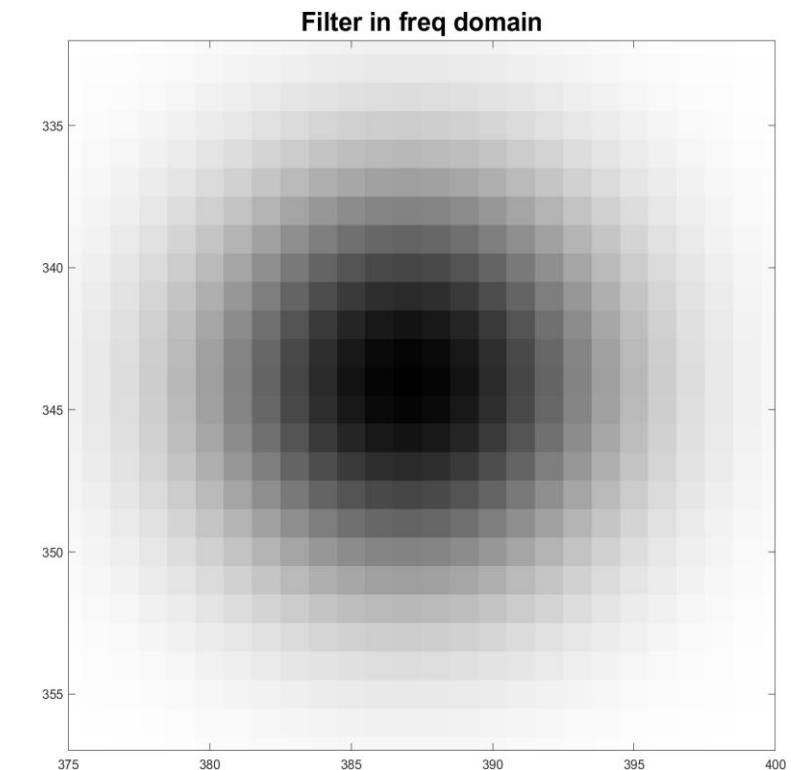
- 1) The bar shaped noises are low frequency components which we want to get rid of, while the texts are high frequency components which we want to preserve. This problem has been addressed in the similar way the fingerprint image was shown to be enhanced in class lecture.
- 2) A high pass filter has been designed, in three different ways – ideal, Butterworth (with order 2) and Gaussian.



Ideal HPF ($D_0=7$)



Butterworth HPF ($D_0=6$)



Gaussian HPF ($D_0=5$)⁹

Letter_bar.bmp – Filtered Images (Ideal HPF)

Filtered Image

Indinty Six between Stockley & Knx and State of Tennessee
Andrew Jackson of the County
that aforesaid of the other part
paid Stockley & Donelson for a
of the sum of two thousand
hand paid the receipt where
 hath and by these presents
self alien enforff and Confir
Jackson his heirs and a
certain traits or parals of La
sand acres 1/4 one thousand payre
and land lies and his
 $D_0 = 5$

None of these are looking good. If some thresholding is applied to binarize the image, then the result with $D_0=7$ might be a little better, but other two will be worse.

Filtered Image

Indinty Six between Stockley & Knx and State of Tennessee
Andrew Jackson of the County
that aforesaid of the other part
paid Stockley & Donelson for a
of the sum of two thousand
hand paid the receipt where
 hath and by these presents
self alien enforff and Confir
Jackson his heirs and a
certain traits or parals of La
sand acres 1/4 one thousand payre
and land lies and his
 $D_0 = 6$

$D_0 = 7$

Filtered Image

Indinty Six between Stockley & Knx and State of Tennessee
Andrew Jackson of the County
that aforesaid of the other part
paid Stockley & Donelson for a
of the sum of two thousand
hand paid the receipt where
 hath and by these presents
self alien enforff and Confir
Jackson his heirs and a
certain traits or parals of La
sand acres 1/4 one thousand payre
and land lies and his
 $D_0 = 7$

Letter_bar.bmp – Ideal LPF with thresholding

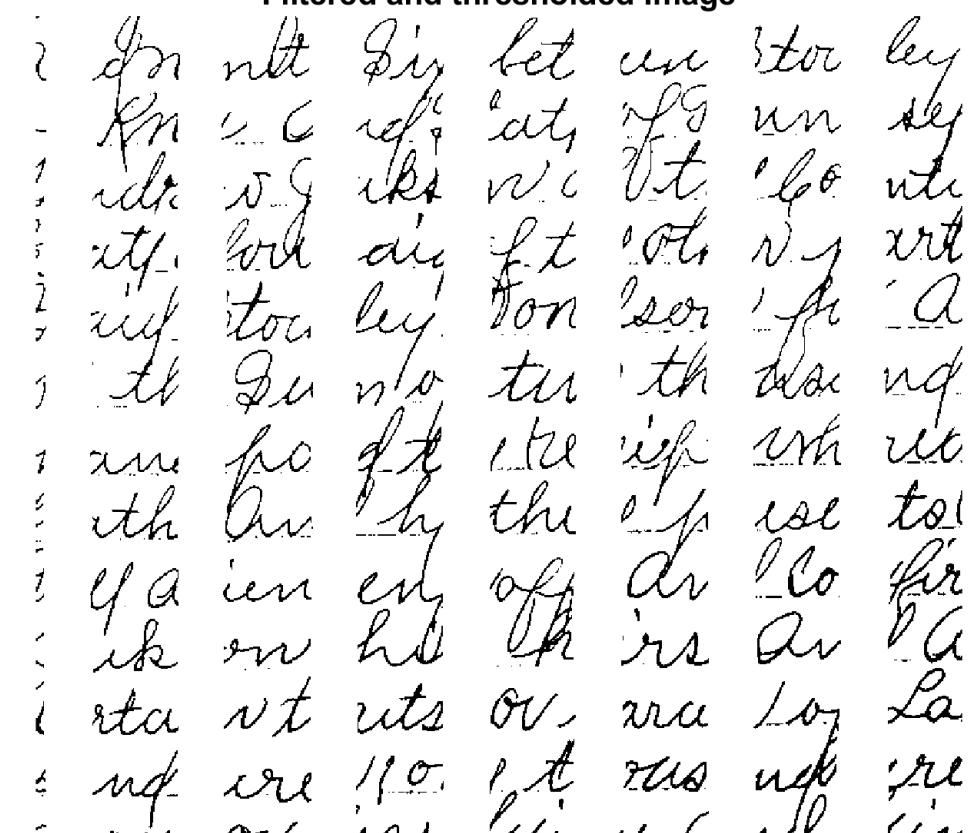
For $D_0=7$, thresholding was applied. The threshold was 0.6 times the mean value of the whole image. The pixel value was set to 1 if its original value is greater than the threshold. The result is better , but still not acceptable because the bars are completely cleaned along with the texts behind them.

Filtered Image



Without thresholding

Filtered and thresholded Image



With thresholding

Letter_bar.bmp - MATLAB code for ideal filtering

```
clc; clear; close all;
fnts=20;I=imread('letter_bar.bmp');[X, Y]=size(I);
JF=fft2(I); JFS=fftshift(JF); FS=JFS;
% For ideal, try different values like 5,6 & 7 but none will give good result.
D0 = 7; H = ones(size(I));
for i=1:X
    for j=1:Y
        D = sqrt((i-(X/2+1))^2+(j-(Y/2+1))^2);
        if(D<=D0) H(i,j) = 0;end
    end
end
JFS=FS.*H;JK=real(ifft2(ifftshift(JFS)));
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
JFF=log(abs(JFS)+1);
Jmin=min(min(JFF)); Jmax=max(max(JFF)); JFF=(JFF-Jmin)/Jmax;
JFT=log(abs(FS)+1);
Jmin=min(min(JFT)); Jmax=max(max(JFT)); JFT=(JFT-Jmin)/Jmax;
figure(1);imshow(I);title('Original Image','FontSize',fnts);
figure(2);imshow(JFT);title('FFT of Original Image','FontSize',fnts);
figure(3);imagesc(H);colormap(gray);title('Filter in freq domain','FontSize',fnts);
figure(4);imshow(JFR);title('Filtered Image before Thresholding','FontSize',fnts);
figure(5);imshow(JFF);title('FFT of filtered Image before thresholding','FontSize',fnts);
%Thresholding
meanval = mean(mean(JFR));threshold = meanval*0.6;
JFRT = zeros([X,Y]);
JFRT(find(JFR>threshold))=1;
JFTH = fftshift(fft2(JFRT));JFTH = log(abs(JFTH)+1);
Jmin=min(min(JFTH));Jmax=max(max(JFTH)); JFTHN=(JFTH-Jmin)/Jmax;
figure(6);imshow(JFRT);title('Filtered and thresholded Image','FontSize',fnts);
figure(7); imshow(JFTHN);title('FFT of filtered Image after thresholding','FontSize',fnts);
```

Letter_bar.bmp – Filtered Images (Butterworth and Gaussian)

Both are doing a lot better job than the ideal one. *But the texts in the background of the bars are not that bright*, hence thresholding should play a crucial role here.

Filtered Image

Indinty Six between Stockley
of Knox And State of Tennessee
Andrew Jackson of the County
that Afore said of the other part
Paid Stockley Donelson for a
of the sum of two thousand
and paid the receipt where it
rath And by these presents
I'll alien enforff And Confir
Jackson his heirs And a
certain traits or parcels of La
sand acres one thousand acre
more or less And his
manor and lands and his

Filtered Image

Indinty Six between Stockley
of Knox And State of Tennessee
Andrew Jackson of the County
that Afore said of the other part
Paid Stockley Donelson for a
of the sum of two thousand
and paid the receipt where it
rath And by these presents
I'll alien enforff And Confir
Jackson his heirs And a
certain traits or parcels of La
sand acres one thousand acre
more or less And his
manor and lands and his

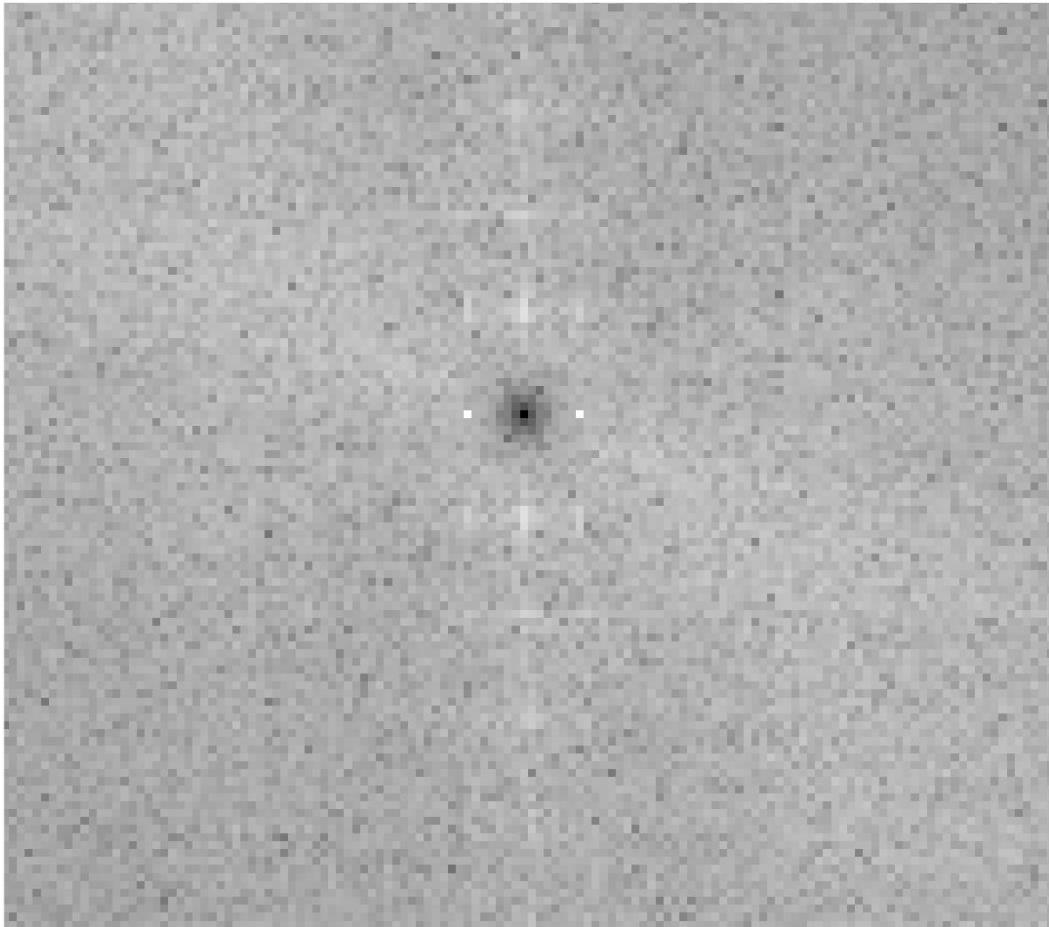
Filtered image : Butterworth ($D_0 = 6$)

Filtered image : Gaussian ($D_0 = 5$)

Letter_bar.bmp – Filtered Images (Butterworth and Gaussian)

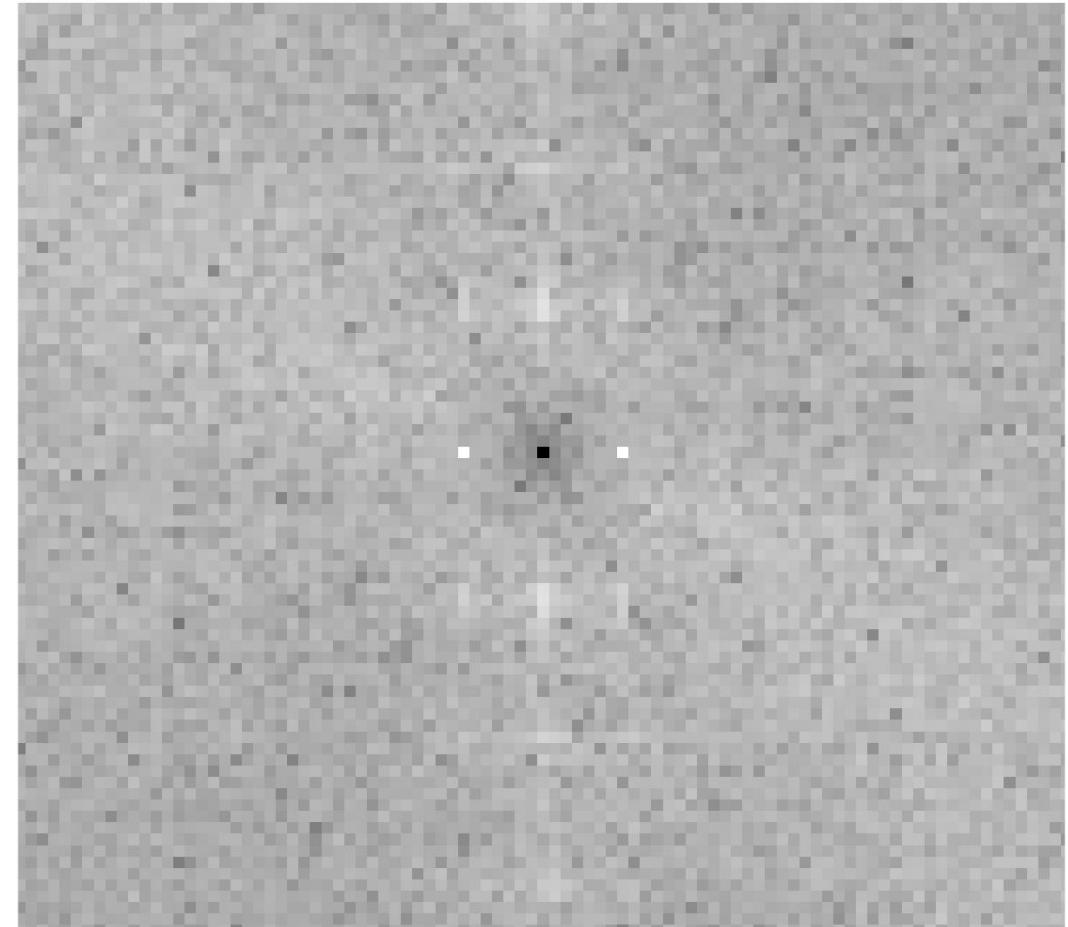
Look at the images in frequency domain after zooming in at the center.

FFT of filtered Image before thresholding



FFT of Filtered image : Butterworth
($D_0 = 6$)

FFT of filtered Image before thresholding



FFT of Filtered image : Gaussian
($D_0 = 5$)

Letter_bar.bmp – Filtered and Thresholded Images (Butterworth and Gaussian)

Thresholding has been performed using the same threshold used for ideal filter. Both of Butterworth and Gaussian filters seem to produce great results after thresholding.

Filtered and thresholded Image

Indinty Six between Stockley
of Knyx and State of Tennessee
Andrew Jackson of the County
of Alor said of the other part
said Stockley Donelson for a
sum of two thousand
hand paid the receipt where
rath and by these presents
I sell alien enfoff and confir
Jackson his heirs and a
certain traits or parcels of la
and acres long thousand payre
and in witness where
I have written this day of
January in the year of our Lord
one thousand eight hundred and
sixty three and signed my name
John Stockley

Filtered image after thresholding :

Butterworth ($D_0 = 6$)

Filtered and thresholded Image

Indinty Six between Stockley
of Knyx and State of Tennessee
Andrew Jackson of the County
of Alor said of the other part
said Stockley Donelson for a
sum of two thousand
hand paid the receipt where
rath and by these presents
I sell alien enfoff and confir
Jackson his heirs and a
certain traits or parcels of la
and acres long thousand payre
and in witness where
I have written this day of
January in the year of our Lord
one thousand eight hundred and
sixty three and signed my name
John Stockley

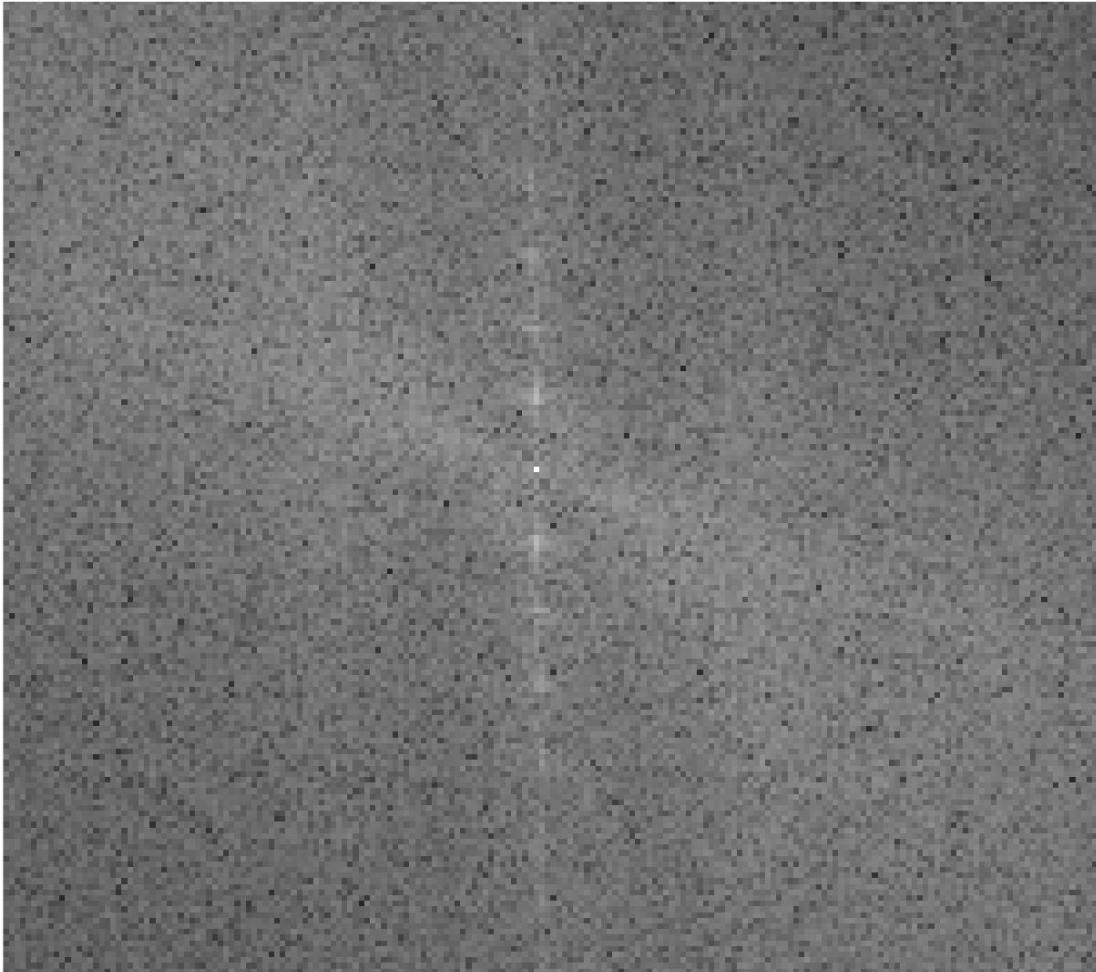
Filtered image after thresholding :

Gaussian ($D_0 = 5$)

Letter_bar.bmp – Filtered and Thresholded Images (Butterworth and Gaussian)

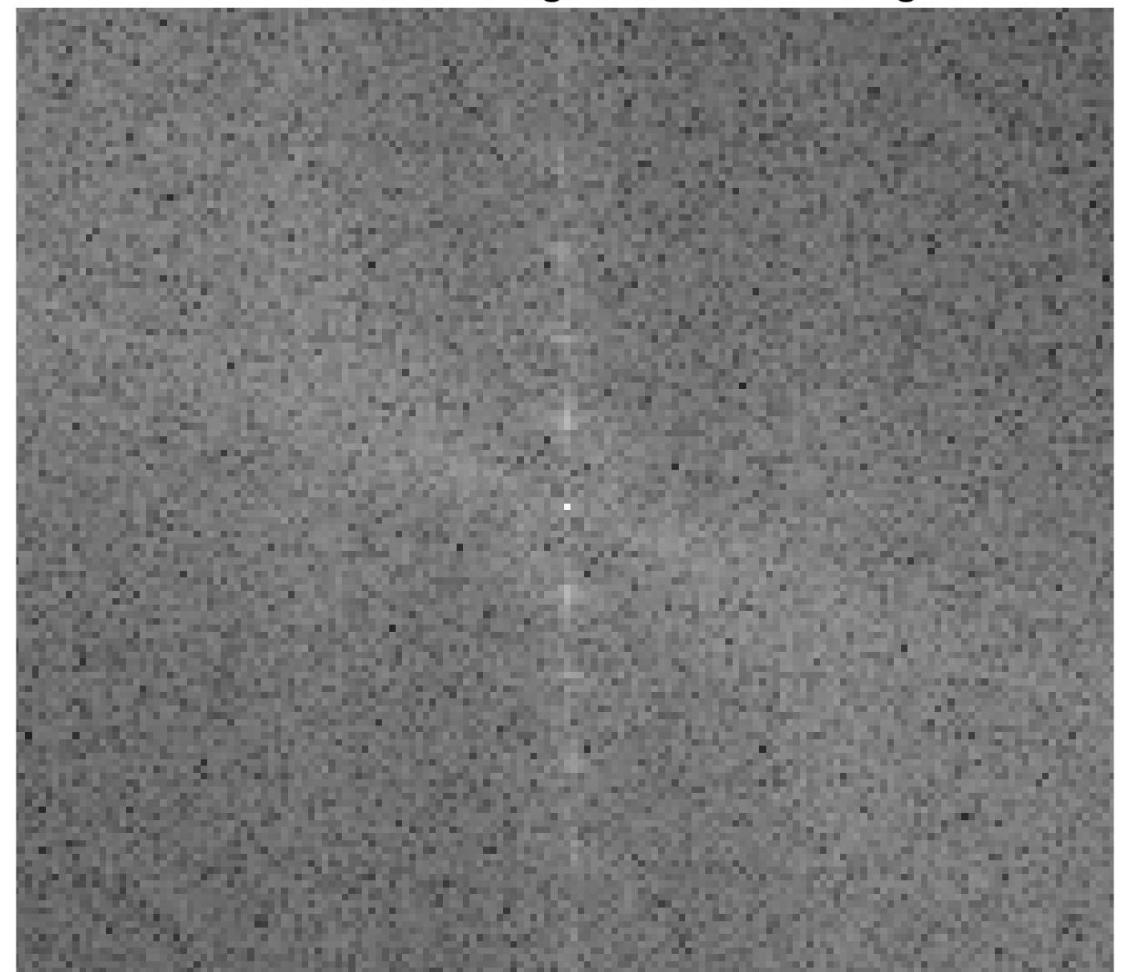
Look at the images in frequency domain after zooming in at the center.

FFT of filtered Image after thresholding



FFT of Filtered image after
thresholding : Butterworth ($D_0 = 6$)

FFT of filtered Image after thresholding



FFT of Filtered image after
thresholding : Gaussian ($D_0 = 5$)

Letter_bar.bmp - MATLAB code for Butterworth filtering

```
clc; clear; close all;
fnts=20; I=imread('letter_bar.bmp');[X, Y]=size(I);
JF=fft2(I); JFS=fftshift(JF); FS=JFS;
D0 = 6; %D0 is filter width. For best result, use 6 for butterworth
n = 2; %Order of the Butterworth filter
H = ones(size(I));
for i=1:X
    for j=1:Y
        D = sqrt((i-(X/2+1))^2+(j-(Y/2+1))^2);
        H(i,j) = 1/(1+(D0/D).^(2*n));
    end
end
JFS=FS.*H;JK=real(ifft2(ifftshift(JFS)));
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
JFF=log(abs(JFS)+1);
Jmin=min(min(JFF)); Jmax=max(max(JFF)); JFF=(JFF-Jmin)/Jmax;
JFT=log(abs(FS)+1);
Jmin=min(min(JFT)); Jmax=max(max(JFT)); JFT=(JFT-Jmin)/Jmax;
figure(1); imshow(I);title('Original Image','FontSize',fnts);
figure(2); imshow(JFT);title('FFT of Original Image','FontSize',fnts);
figure(3);imagesc(H);colormap(gray);title('Filter in freq domain','FontSize',fnts);
figure(4);imshow(JFR);title('Filtered Image before Thresholding','FontSize',fnts);
figure(5); imshow(JFF);title('FFT of filtered Image before thresholding','FontSize',fnts);
%Thresholding
meanval = mean(mean(JFR));threshold = meanval*0.6;
JFRT = zeros([X,Y]);
JFRT(find(JFR>threshold))=1;
JFTH = fftshift(fft2(JFRT));JFTH = log(abs(JFTH)+1);
Jmin=min(min(JFTH)); Jmax=max(max(JFTH)); JFTHN=(JFTH-Jmin)/Jmax;
figure(6);imshow(JFRT);title('Filtered and thresholded Image','FontSize',fnts);
figure(7); imshow(JFTHN);title('FFT of filtered Image after thresholding','FontSize',fnts);
```

Letter_bar.bmp - MATLAB code for Gaussian filtering

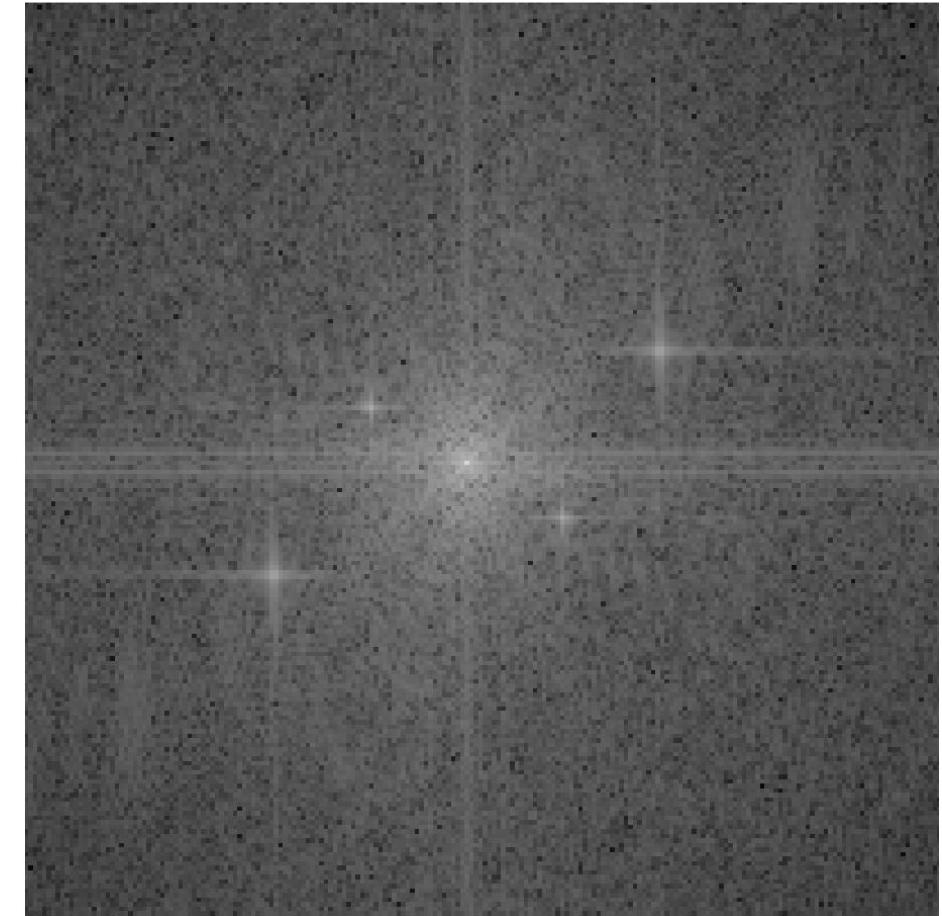
```
clc; clear; close all;
fnts=20;I=imread('letter_bar.bmp');[X, Y]=size(I);
JF=fft2(I); JFS=fftshift(JF); FS=JFS;
D0 = 5; %D0 is filter width. For best result, use 5 for Gaussian.
H = ones(size(I));
for i=1:X
    for j=1:Y
        D = sqrt((i-(X/2+1))^2+(j-(Y/2+1))^2);
        H(i,j) =1-exp(-0.5*((D^2)/(D0*D0)));
    end
end
JFS=FS.*H;JK=real(ifft2(ifftshift(JFS)));
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
JFF=log(abs(JFS)+1);
Jmin=min(min(JFF)); Jmax=max(max(JFF)); JFF=(JFF-Jmin)/Jmax;
JFT=log(abs(FS)+1);
Jmin=min(min(JFT)); Jmax=max(max(JFT)); JFT=(JFT-Jmin)/Jmax;
figure(1); imshow(I);title('Original Image','FontSize',fnts);
figure(2); imshow(JFT);title('FFT of Original Image','FontSize',fnts);
figure(3);imagesc(H);colormap(gray);title('Filter in freq domain','FontSize',fnts);
figure(4);imshow(JFR);title('Filtered Image before Thresholding','FontSize',fnts);
figure(5); imshow(JFF);title('FFT of filtered Image before thresholding','FontSize',fnts);
%Thresholding
meanval = mean(mean(JFR));threshold = meanval*0.6;
JFRT = zeros([X,Y]);
JFRT(find(JFR>threshold))=1;
JFTH = fftshift(fft2(JFRT));JFTH = log(abs(JFTH)+1);
Jmin=min(min(JFTH)); Jmax=max(max(JFTH)); JFTHN=(JFTH-Jmin)/Jmax;
figure(6);imshow(JFRT);title('Filtered and thresholded Image','FontSize',fnts);
figure(7); imshow(JFTHN);title('FFT of filtered Image after thresholding','FontSize',fnts);
```

Clowngray.bmp

Original Image



FFT of Original Image

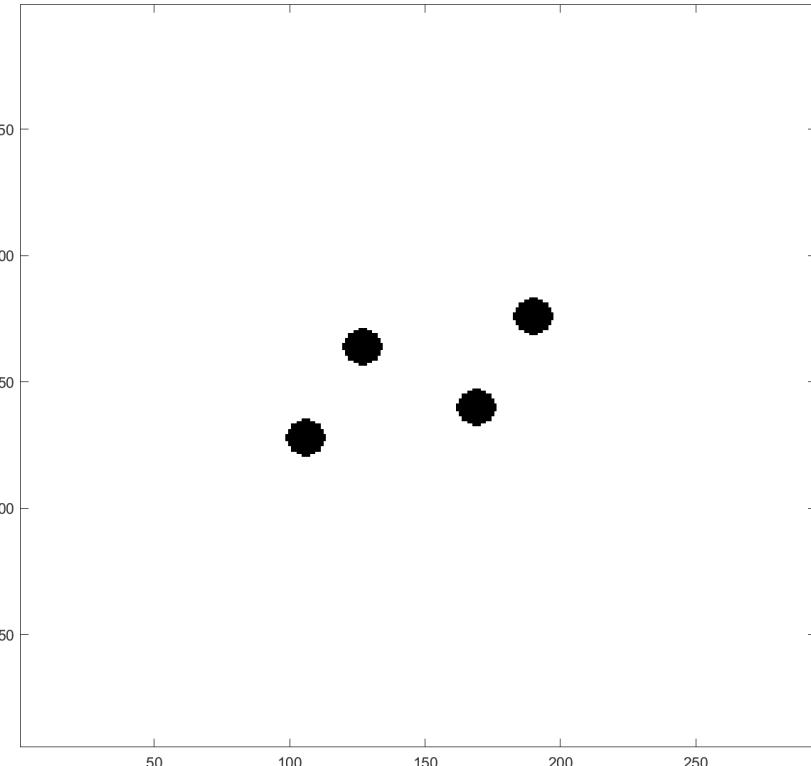


Slightly zoomed in

There are four bright peaks which are responsible for noise. Notch filter has to be used to suppress these.

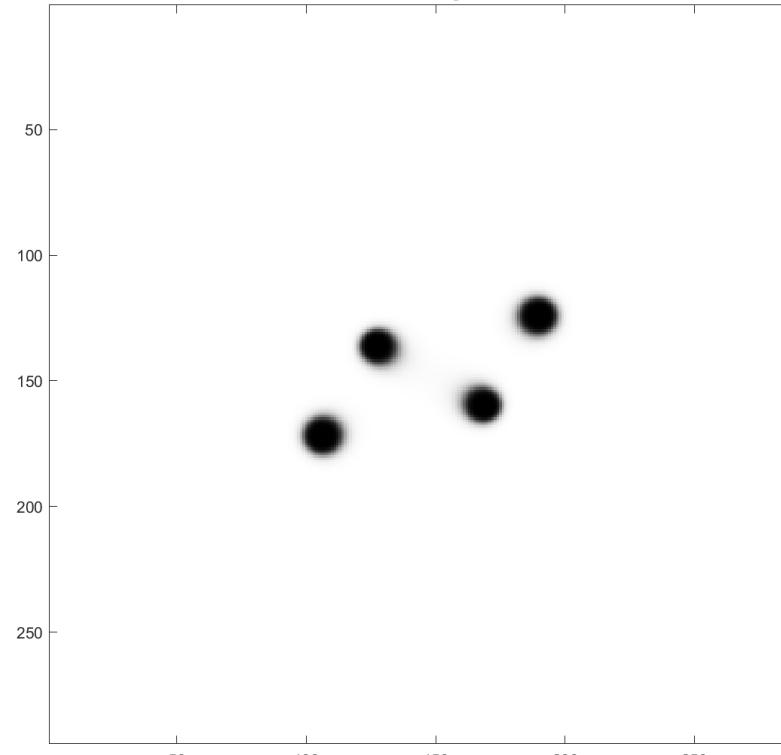
Clowngray.bmp – designed notch filters

Filter in freq domain



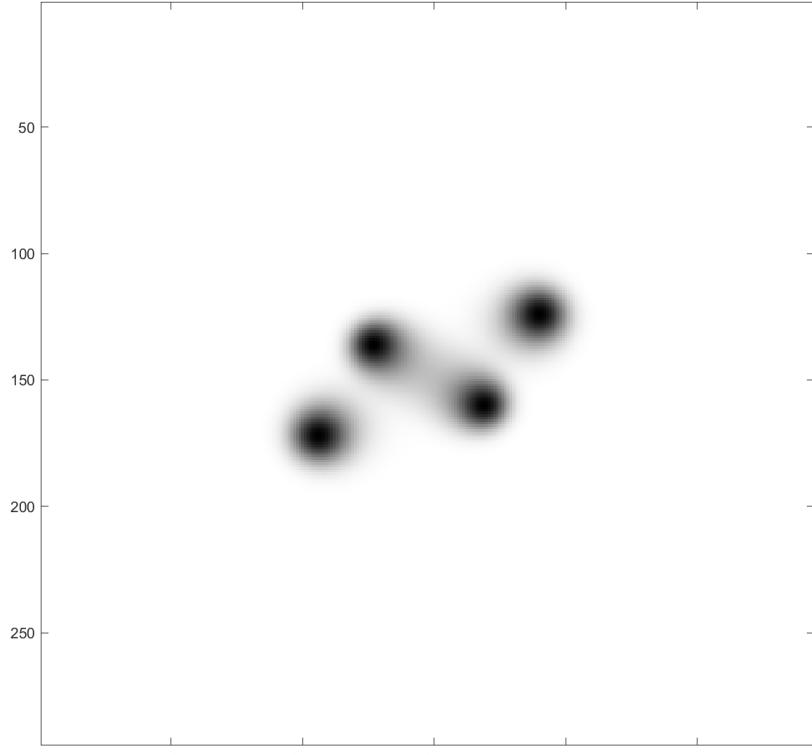
Ideal notch filter

Filter in freq domain



Butterworth notch filter (order=2)

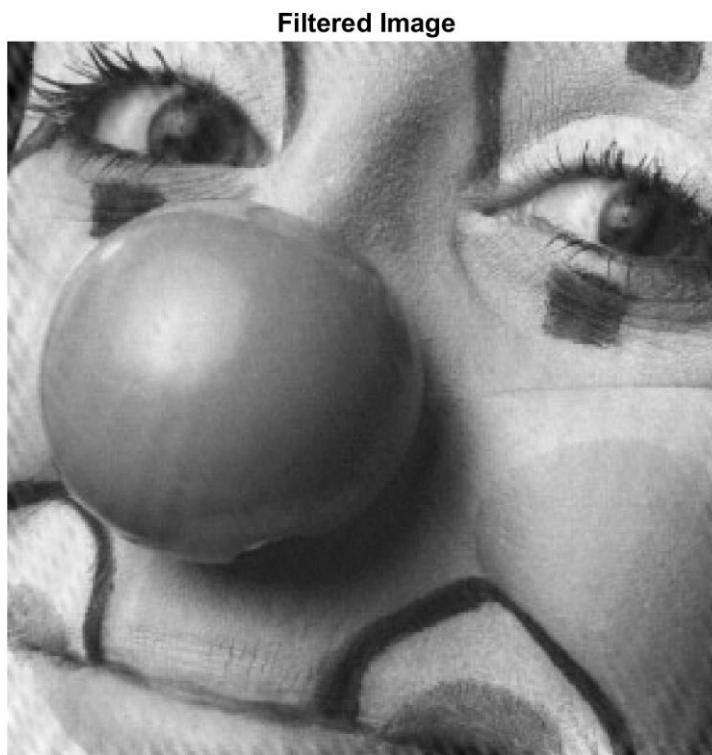
Filter in freq domain



Gaussian notch filter

In case of Gaussian filter, the center values (average) are being affected by the notches closer to the center. Hence a windowing is performed to preserve the original low frequency values (not shown in above images, it has been done on the FFT of filtered image). Butterworth one is affecting the center less, but still the same windowing was performed to stay safe. For ideal one, windowing is not performed.

Clowngray.bmp – Filtered images



Filtered by Ideal notch filter



Filtered by Butterworth notch filter,
followed by windowing at the center

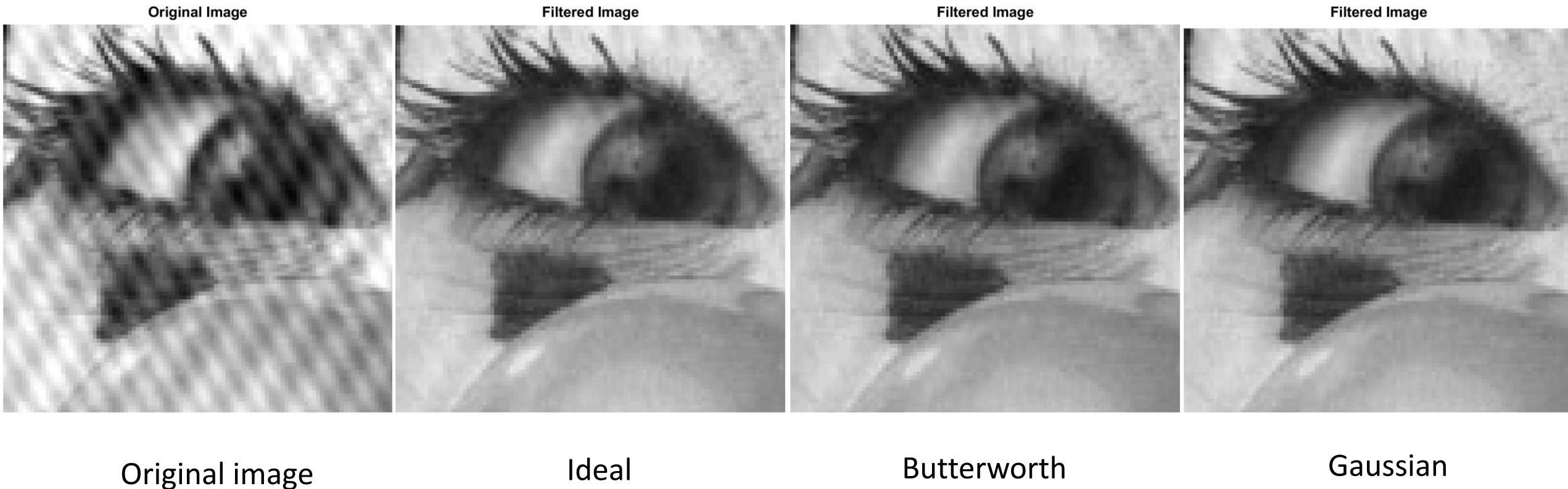


Filtered by Gaussian notch filter,
followed by windowing at the center

Some noise or artifact remained at the border of each images, because of circular convolution without zero padding. Zero padding would solve or at least minimize this problem.

Clowngray.bmp – Filtered images

Compare by zooming in. The image that was filtered by ideal filter has some noise or artifact. Butterworth and Gaussian filter are giving the best results.



Original image

Ideal

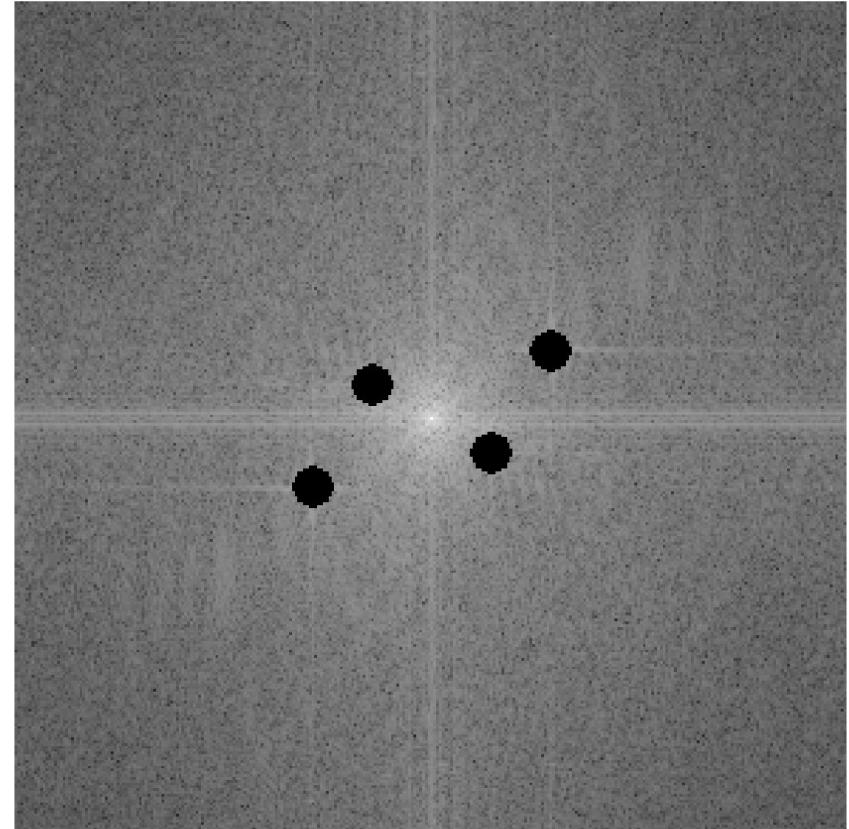
Butterworth

Gaussian

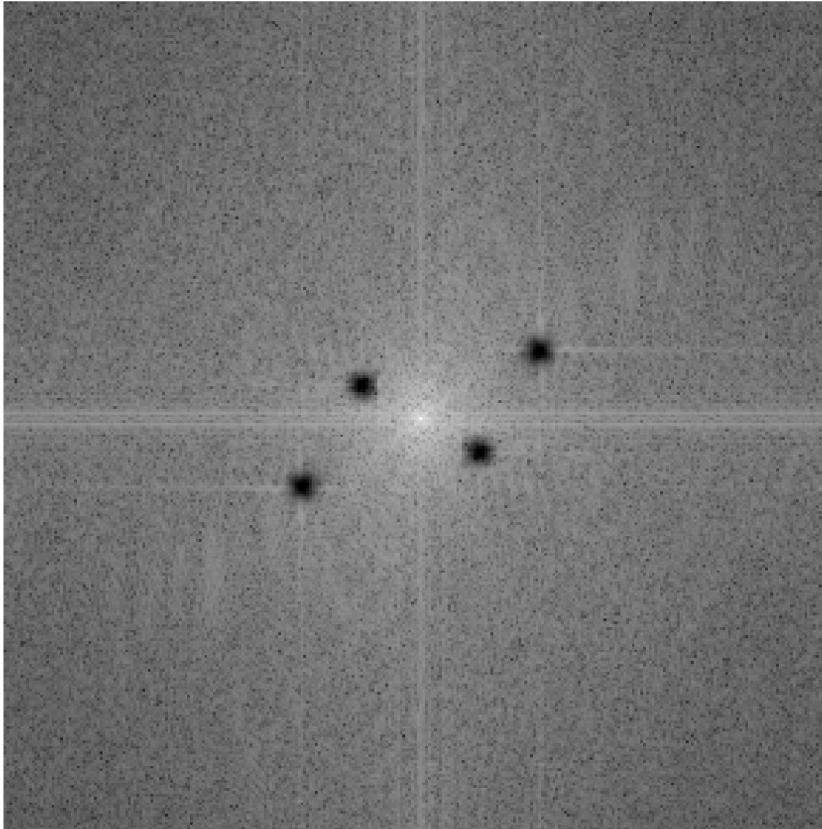
Clowngray.bmp – Filtered images

Look at the images in frequency domain.

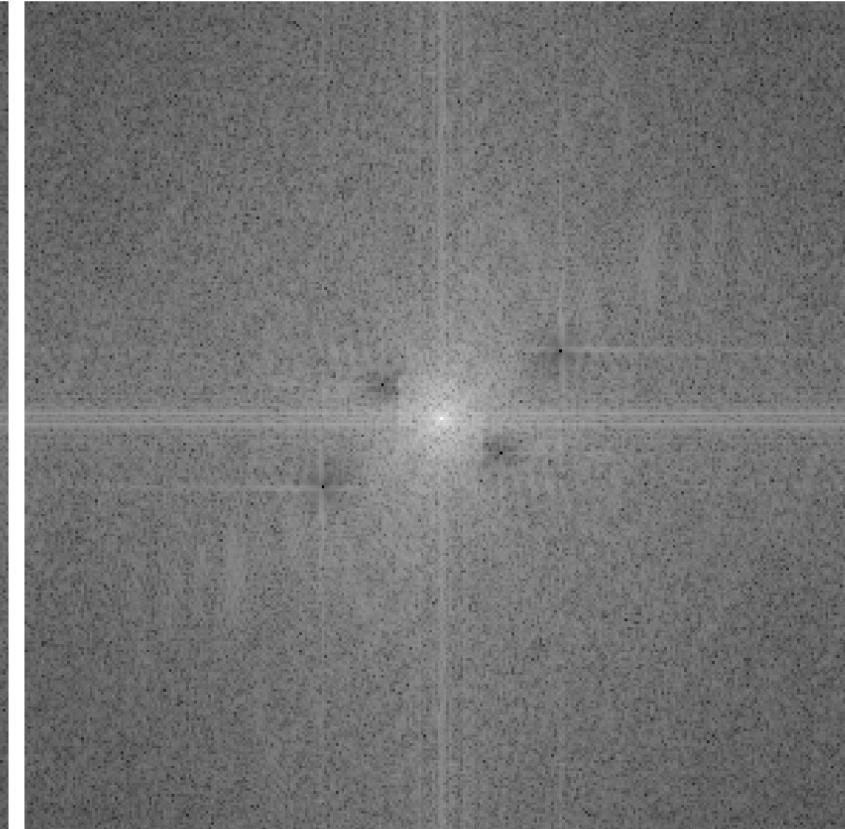
FFT of filtered Image



FFT of filtered Image



FFT of filtered Image



FFT of the image filtered by
Ideal notch filter

FFT of the image filtered by
Butterworth notch filter, followed by
windowing at the center

FFT of the image filtered by
Gaussian notch filter, followed by
windowing at the center

clowngray.bmp - MATLAB code for ideal filtering

```
clc; clear; close all;
fnts=20;
I=imread('clowngray.bmp');[X, Y]=size(I);
JF=fft2(I); JFS=fftshift(JF); FS=JFS;
u0=124; v0=190; u1=172; v1=106;
u3=136; v3=127; u4=160; v4=169;
D0=50;
H = ones(size(I));
for i=1:X
    for j=1:Y
        D1=(i-u0)^2+(j-v0)^2;
        D2=(i-u1)^2+(j-v1)^2;
        D3=(i-u3)^2+(j-v3)^2;
        D4=(i-u4)^2+(j-v4)^2;
        if(D1<=D0| |D2<=D0| |D3<=D0| |D4<=D0) H(i,j) = 0;end
    end
end
JFS=FS.*H;JK=real(ifft2(ifftshift(JFS)));
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
JFF=log(abs(JFS)+1);
Jmin=min(min(JFF)); Jmax=max(max(JFF)); JFF=(JFF-Jmin)/Jmax;
JFT=log(abs(FS)+1);
Jmin=min(min(JFT)); Jmax=max(max(JFT)); JFT=(JFT-Jmin)/Jmax;
figure(1); imshow(I);title('Original Image','FontSize',fnts);
figure(2); imshow(JFT);title('FFT of Original Image','FontSize',fnts);
figure(3);imagesc(H);colormap(gray);
title('Filter in freq domain','FontSize',fnts);
figure(4);imshow(JFR);title('Filtered Image','FontSize',fnts);
figure(5); imshow(JFF);title('FFT of filtered Image','FontSize',fnts);
```

clowngray.bmp - MATLAB code for Butterworth filtering

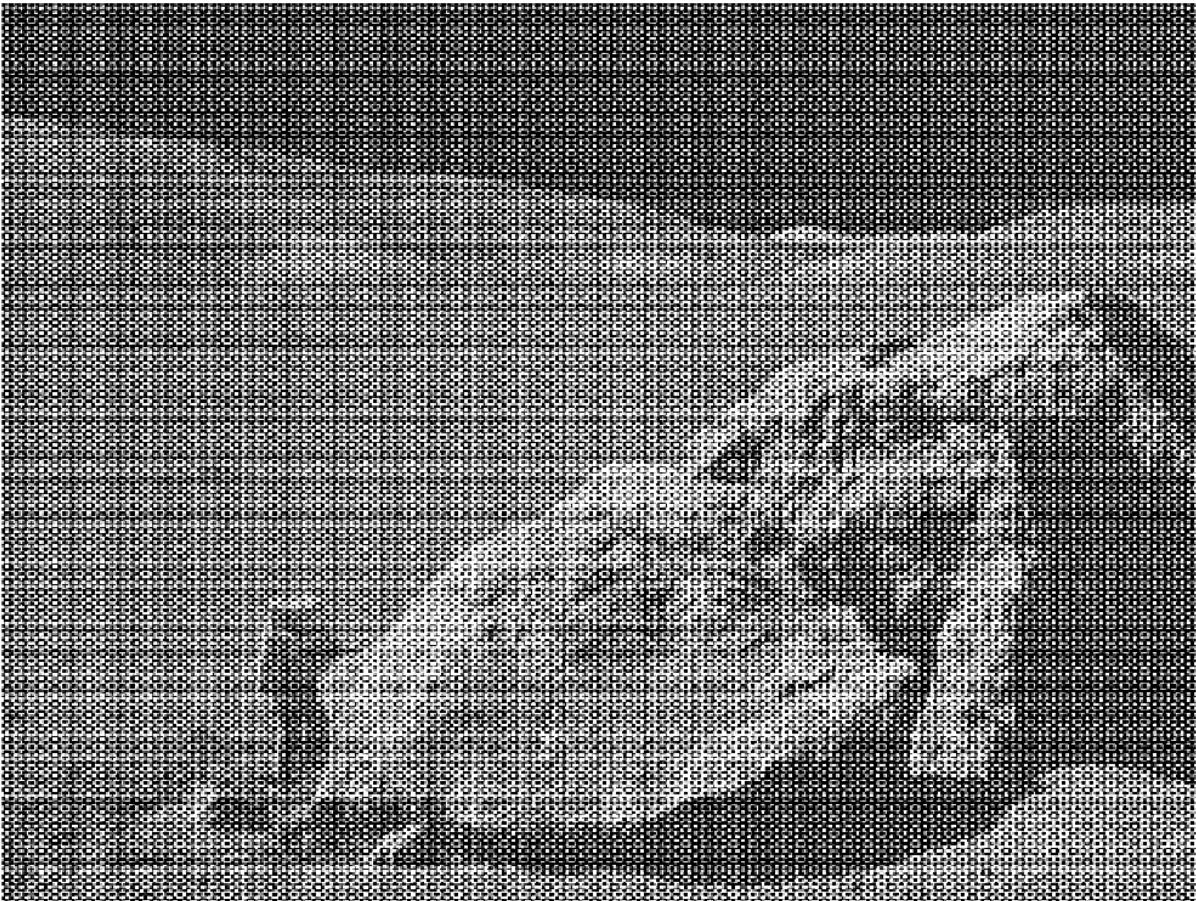
```
clc; clear; close all;
fnts=20;
I=imread('clowngray.bmp');[X, Y]=size(I);
JF=fft2(I); JFS=fftshift(JF); FS=JFS;JFO = JFS;
u0=124; v0=190; u1=172; v1=106;
u3=136; v3=127; u4=160; v4=169;D01=750;D02=350;
n = 2; %Order of the Butterworth filter
H = ones(size(I));
for i=1:X
    for j=1:Y
        D1=(i-u0)^2+(j-v0)^2; D2=(i-u1)^2+(j-v1)^2;
        D3=(i-u3)^2+(j-v3)^2; D4=(i-u4)^2+(j-v4)^2;
        H1(i,j) =1/(1+(D01^2/(D1*D2))^(2*n));
        H2(i,j) =1/(1+(D02^2/(D3*D4))^(2*n));
        H(i,j) = H1(i,j)*H2(i,j);
    end
end
JFS=FS.*H;
W=14; JFS(X/2-W:X/2+1+W,Y/2-W:Y/2+W+1)=JFO(X/2-W:X/2+1+W,Y/2-W:Y/2+W+1);
JK=real(ifft2(ifftshift(JFS)));
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
JFF=log(abs(JFS)+1);
Jmin=min(min(JFF)); Jmax=max(max(JFF)); JFF=(JFF-Jmin)/Jmax;
JFT=log(abs(FS)+1);
Jmin=min(min(JFT)); Jmax=max(max(JFT)); JFT=(JFT-Jmin)/Jmax;
figure(1); imshow(I);title('Original Image','FontSize',fnts);
figure(2); imshow(JFT);title('FFT of Original Image','FontSize',fnts);
figure(3);imagesc(H);colormap(gray);
title('Filter in freq domain','FontSize',fnts);
figure(4);imshow(JFR);title('Filtered Image','FontSize',fnts);
figure(5); imshow(JFF);title('FFT of filtered Image','FontSize',fnts);
```

clowngray.bmp - MATLAB code for Gaussian filtering

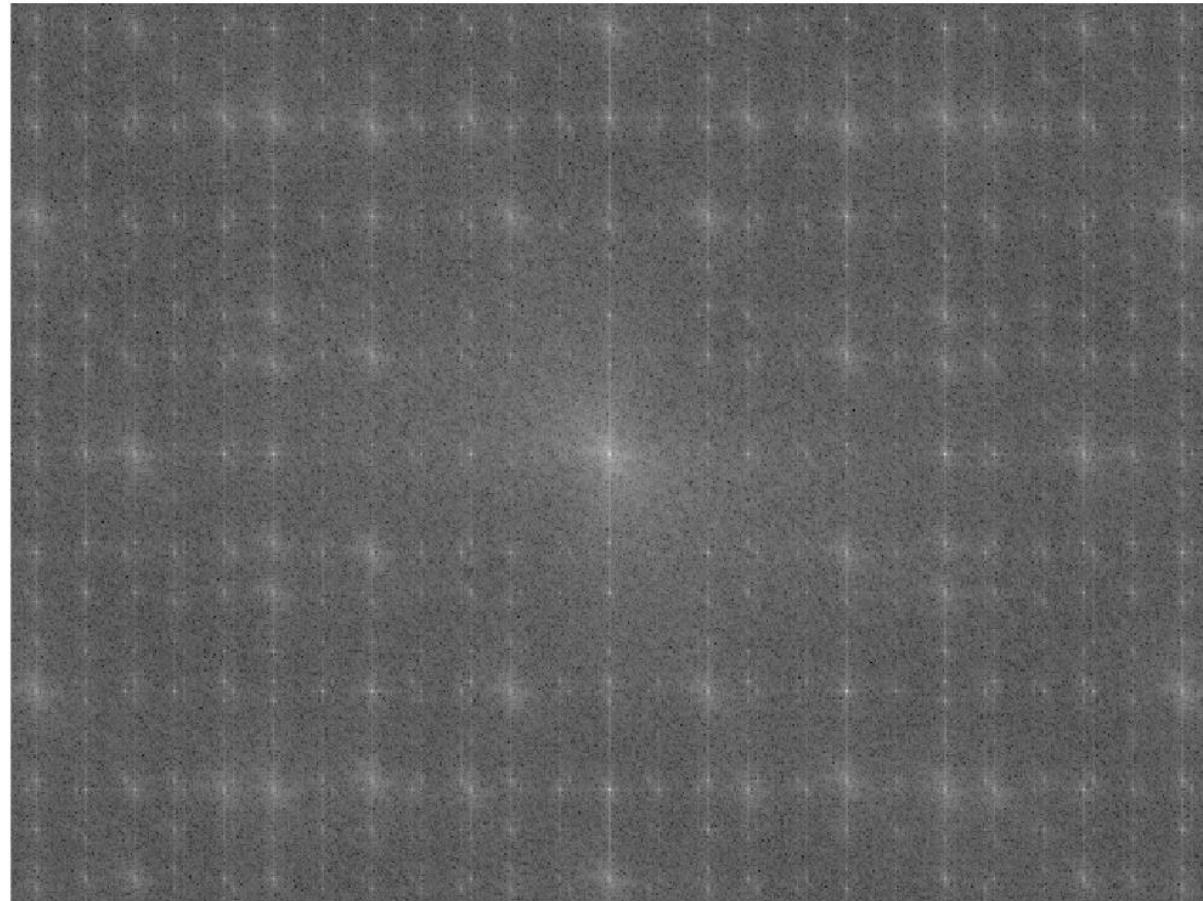
```
clc; clear; close all;
fnts=20;I=imread('clowngray.bmp');[X, Y]=size(I);
JF=fft2(I); JFS=fftshift(JF); FS=JFS;JFO = JFS;
u0=124; v0=190; u1=172; v1=106;
u3=136; v3=127; u4=160; v4=169;
D01=750;D02=350;
for i=1:X
    for j=1:Y
        D1=(i-u0)^2+(j-v0)^2;
        D2=(i-u1)^2+(j-v1)^2;
        D3=(i-u3)^2+(j-v3)^2;
        D4=(i-u4)^2+(j-v4)^2;
        H1(i,j) =1-exp(-0.5*((D1*D2)/(D01^2)));
        H2(i,j) =1-exp(-0.5*((D3*D4)/(D02^2)));
        H(i,j) = H1(i,j)*H2(i,j);
    end
end
JFS=FS.*H;
W=14; JFS(X/2-W:X/2+1+W,Y/2-W:Y/2+W+1)=JFO(X/2-W:X/2+1+W,Y/2-W:Y/2+W+1);
JK=real(ifft2(ifftshift(JFS)));
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
JFF=log(abs(JFS)+1);
Jmin=min(min(JFF)); Jmax=max(max(JFF)); JFF=(JFF-Jmin)/Jmax;
JFT=log(abs(FS)+1);
Jmin=min(min(JFT)); Jmax=max(max(JFT)); JFT=(JFT-Jmin)/Jmax;
figure(1); imshow(I);title('Original Image','FontSize',fnts);
figure(2); imshow(JFT);title('FFT of Original Image','FontSize',fnts);
figure(3);imagesc(H);colormap(gray);
title('Filter in freq domain','FontSize',fnts);
figure(4);imshow(JFR);title('Filtered Image','FontSize',fnts);
figure(5); imshow(JFF);title('FFT of filtered Image','FontSize',fnts);
```

MoonLanding.bmp

Original Image

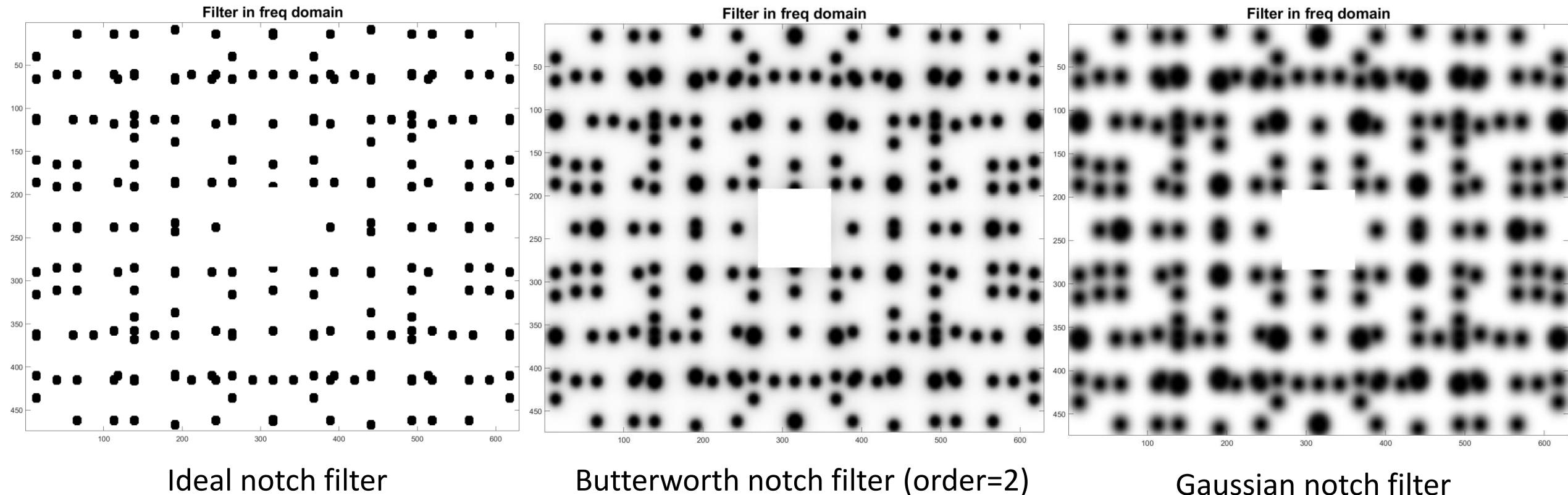


FFT of Original Image



Here, the relevant content has mostly low frequency hence that part should be preserved. The periodic noise has high frequency which should be suppressed by notch filter.

MoonLanding.bmp – designed notch filters



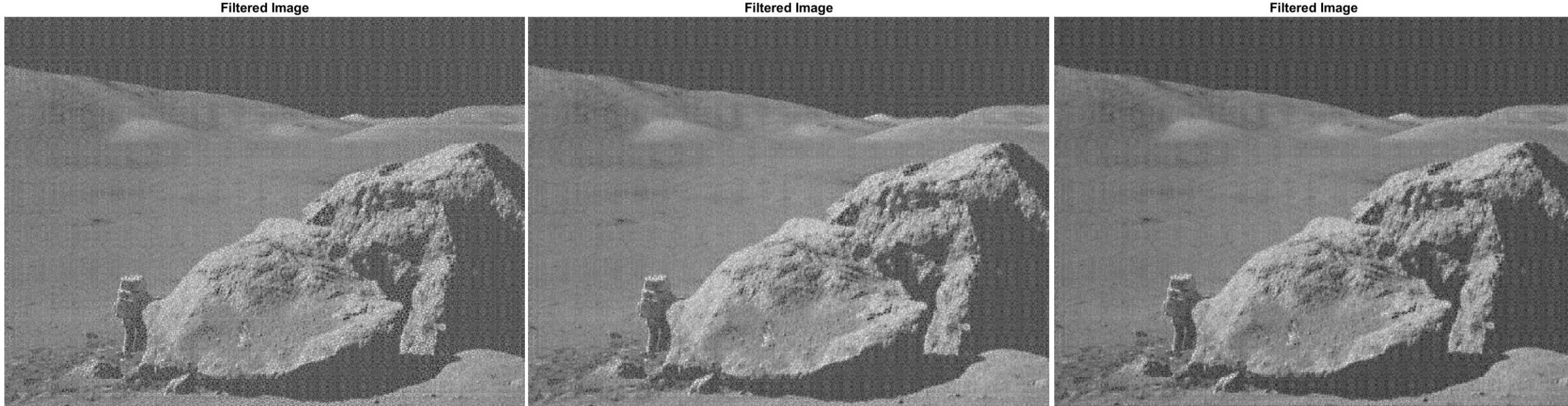
Ideal notch filter

Butterworth notch filter (order=2)

Gaussian notch filter

Widowing has been performed at the center to preserve the low frequency average value

Clowngray.bmp – Filtered images



Filtered by Ideal notch filter

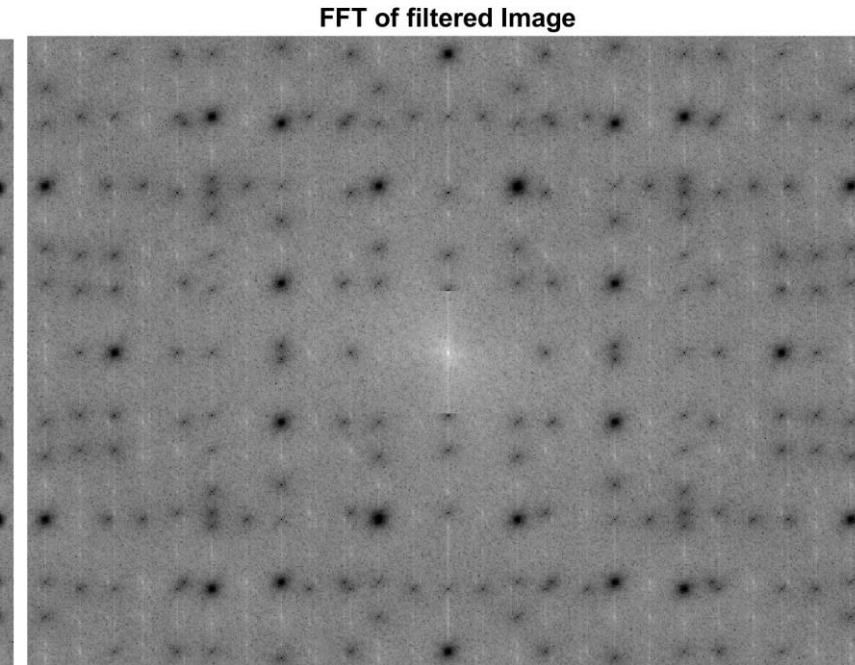
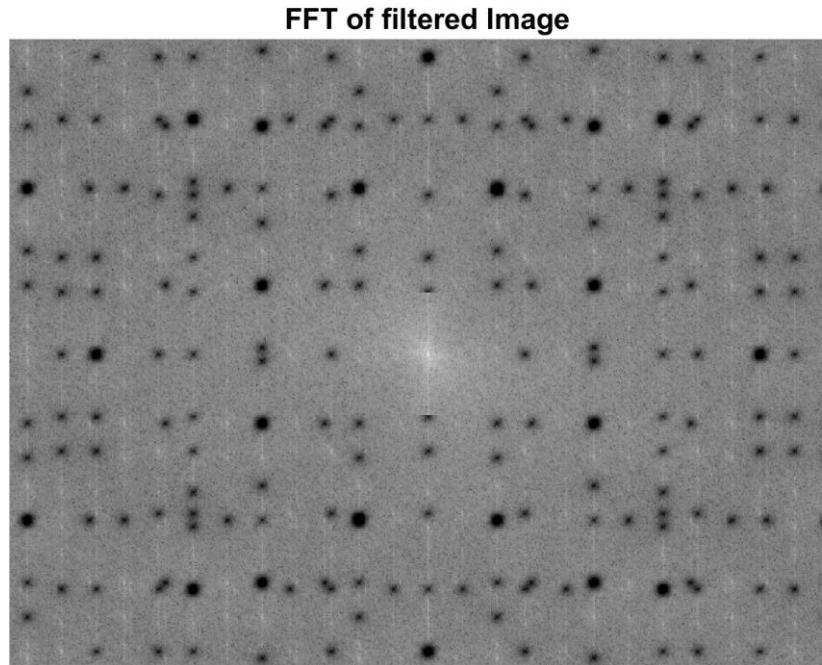
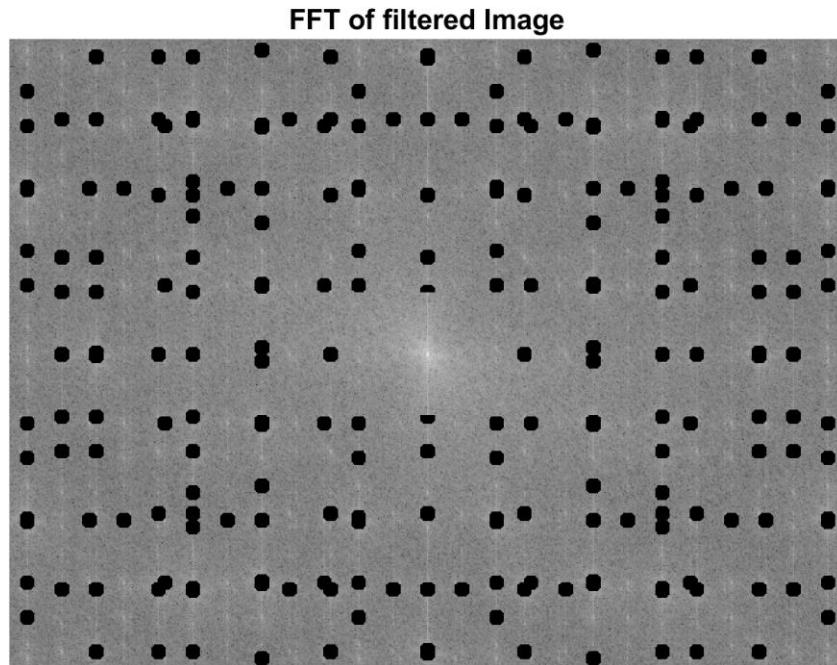
Filtered by Butterworth notch filter,
followed by windowing at the center

Filtered by Gaussian notch filter,
followed by windowing at the center

Gaussian and Butterworth is doing a better job, giving clearer images.

MoonLanding.bmp – Filtered images

Look at the images in frequency domain.



FFT of the image filtered by
Ideal notch filter

FFT of the image filtered by
Butterworth notch filter, followed by
windowing at the center

FFT of the image filtered by
Gaussian notch filter, followed by
windowing at the center

MoonLanding.bmp - MATLAB code for ideal filtering

```
clc; clear; close all;
fnts=20;I=imread('MoonLanding.bmp');[X, Y]=size(I);
JF=fft2(I); JFS=fftshift(JF); FS=JFS;JFO = JFS;
JFT=log(abs(FS)+1);
Jmin=min(min(JFT)); Jmax=max(max(JFT)); JFT=(JFT-Jmin)/Jmax;
th=mean(mean(JFT));
[un, vn] = find(JFT>th*1.5);
D0 = 6; H = ones(size(I));
for i=1:X
    for j=1:Y
        for k = 1:length(un)
            D(k) = sqrt((i-un(k))^2+(j-vn(k))^2);
        end
        if(sum(D<D0)>=1)
            H(i,j)=0;
        end
    end
end
W=45; H(X/2-W:X/2+1+W,Y/2-W:Y/2+W+1)=1;
JFS=FS.*H;JK=real(ifft2(ifftshift(JFS)));
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
JFF=log(abs(JFS)+1);
Jmin=min(min(JFF)); Jmax=max(max(JFF)); JFF=(JFF-Jmin)/Jmax;
figure(1); imshow(I);title('Original Image','FontSize',fnts);
figure(2); imshow(JFT);title('FFT of Original Image','FontSize',fnts);
figure(3);imagesc(H);colormap(gray);
title('Filter in freq domain','FontSize',fnts);
figure(4);imshow(JFR);title('Filtered Image','FontSize',fnts);
figure(5); imshow(JFF);title('FFT of filtered Image','FontSize',fnts);
```

MoonLanding.bmp - MATLAB code for Butterworth filtering

```
clc; clear; close all;
fnts=20;n=2;I=imread('MoonLanding.bmp');[X, Y]=size(I);
JF=fft2(I); JFS=fftshift(JF); FS=JFS;JFO = JFS;
JFT=log(abs(FS)+1);
Jmin=min(min(JFT)); Jmax=max(max(JFT)); JFT=(JFT-Jmin)/Jmax;
th=mean(mean(JFT));
[un, vn] = find(JFT>th*1.5);
D0 = 8; H = ones(size(I));
for i=1:X
    for j=1:Y
        for k = 1:length(un)
            D = sqrt((i-un(k))^2+(j-vn(k))^2);
            H_temp(i,j) =1/(1+(D0/D)^(2*n));
            H(i,j) = H(i,j)*H_temp(i,j);
        end
    end
end

W=45; H(X/2-W:X/2+1+W,Y/2-W:Y/2+W+1)=1;
JFS=FS.*H;JK=real(ifft2(ifftshift(JFS)));
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
JFF=log(abs(JFS)+1);
Jmin=min(min(JFF)); Jmax=max(max(JFF)); JFF=(JFF-Jmin)/Jmax;
figure(1); imshow(I);title('Original Image','FontSize',fnts);
figure(2); imshow(JFT);title('FFT of Original Image','FontSize',fnts);
figure(3);imagesc(H);colormap(gray);
title('Filter in freq domain','FontSize',fnts);
figure(4);imshow(JFR);title('Filtered Image','FontSize',fnts);
figure(5); imshow(JFF);title('FFT of filtered Image','FontSize',fnts);
```

MoonLanding.bmp - MATLAB code for Gaussian filtering

```
clc; clear; close all;
f=20;I=imread('MoonLanding.bmp');[X, Y]=size(I);
JF=fft2(I); JFS=fftshift(JF); FS=JFS;JFO = JFS;
JFT=log(abs(FS)+1);
Jmin=min(min(JFT)); Jmax=max(max(JFT)); JFT=(JFT-Jmin)/Jmax;
th=mean(mean(JFT));
[un, vn] = find(JFT>th*1.5);
D0 = 8; H = ones(size(I));
for i=1:X
    for j=1:Y
        for k = 1:length(un)
            D = sqrt((i-un(k))^2+(j-vn(k))^2);
            H_temp(i,j) =1-exp(-0.5*((D^2)/(D0^2)));
            H(i,j) = H(i,j)*H_temp(i,j);
        end
    end
end
W=45; H(X/2-W:X/2+1+W,Y/2-W:Y/2+W+1)=1;
JFS=FS.*H;JK=real(ifft2(ifftshift(JFS)));
Jmin=min(min(JK)); Jmax=max(max(JK)); JFR=(JK-Jmin)/Jmax;
JFF=log(abs(JFS)+1);
Jmin=min(min(JFF)); Jmax=max(max(JFF)); JFF=(JFF-Jmin)/Jmax;
figure(1); imshow(I);title('Original Image','FontSize',f);
figure(2); imshow(JFT);title('FFT of Original Image','FontSize',f);
figure(3);imagesc(H);colormap(gray);
title('Filter in freq domain','FontSize',f);
figure(4);imshow(JFR);title('Filtered Image','FontSize',f);
figure(5); imshow(JFF);title('FFT of filtered Image','FontSize',f);
```

Thanks