Image and Video Processing

Programming Assignment 7

Frequency Domain Filters

Submitted by
Madhu Krishnan A P
(Student ID: 24100488)
M.Tech VLSI and Embedded Systems
Cochin University of Science and Technology
Cochin - 22

Question 1

Implement a 2D Ideal LPF. Plot the perspective plot, and filter transfer function as an image. Apply the filter to Figure Fig 0442(a) for cut-off frequencies 10, 30, 60, 160 and 460.

Load input image

```
img = imread('Fig0442(a)(characters_test_pattern).tif');
[M, N] = size(img);
```

Define cutoff frequencies

```
D0_values = [10, 30, 60, 160, 460];
```

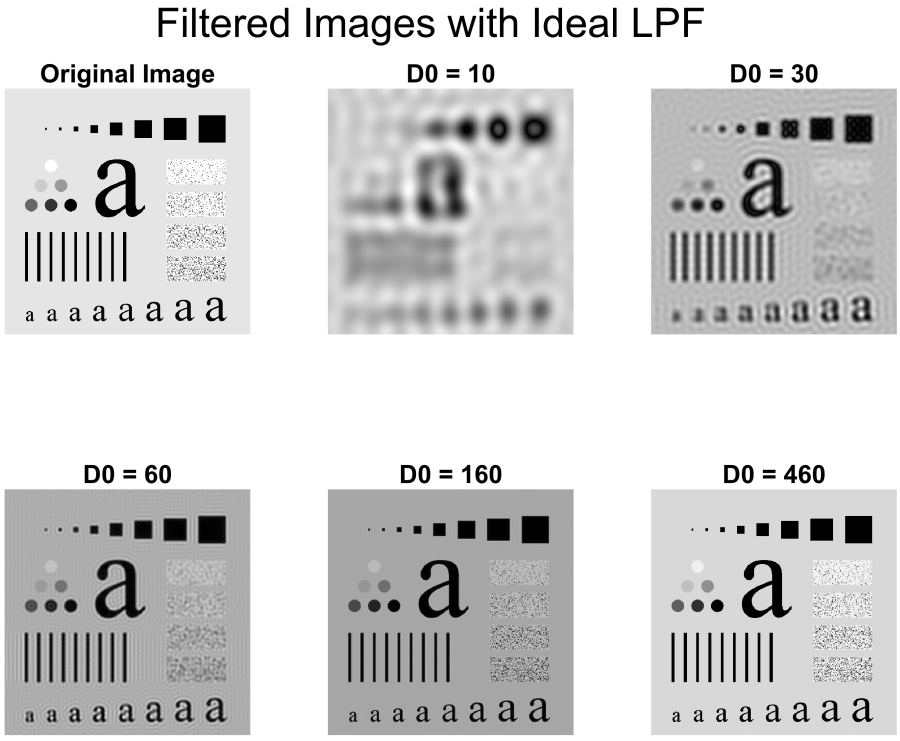
Create frequency coordinates

```
[U, V] = meshgrid(-N/2:N/2-1, -M/2:M/2-1);
D = sqrt(U.^2 + V.^2);
```

Compute FFT of the image

```
img_fft = fftshift(fft2(img));
```

Apply ideal Low Pass Filter (LPF) and display the original image and results



```
figure;
subplot(2, ceil((length(D0_values) + 1)/2), 1);
imshow(img, []);
title('Original Image');

for i = 1:length(D0_values)
    subplot_index = i + 1;
    D0 = D0_values(i);

    H = double(D <= D0);

    img_fft_filtered = H .* img_fft;
    img_filtered = abs(ifft2(ifftshift(img_fft_filtered)));

    subplot(2, ceil((length(D0_values) + 1)/2), subplot_index);
    imshow(img_filtered, []);
    title(['D0 = ', num2str(D0)]);
end

sgtitle('Filtered Images with Ideal LPF');
```

Question 2

Implement a 2D Butterworth LPF. Plot the perspective plot, filter transfer function as an image. Apply the filter to Figure Fig 0445(a) for order 2 and cut off frequencies as 10, 30, 60, 160 and 460.

Load the input image

```
img = imread('Fig0445(a)(characters_test_pattern).tif');
[M, N] = size(img);
```

Define cutoff frequencies & order

```
D0_values = [10, 30, 60, 160, 460];
n = 2;
```

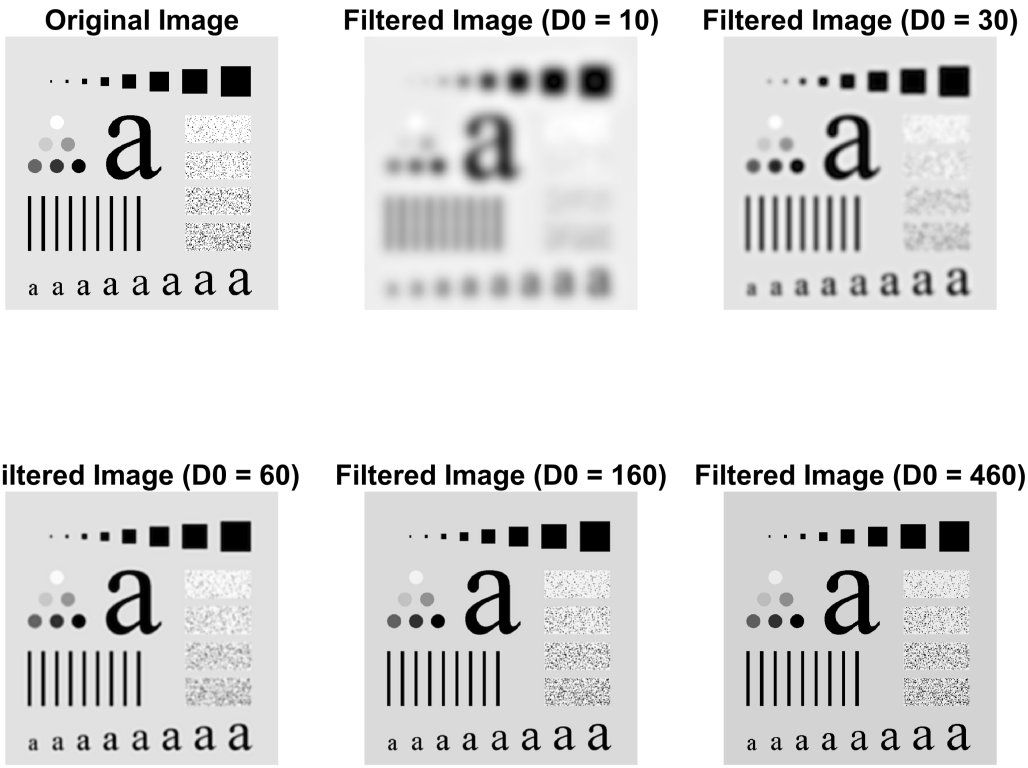
Create frequency coordinates

```
[U, V] = meshgrid(-N/2:N/2-1, -M/2:M/2-1);
D = sqrt(U.^2 + V.^2);
```

Apply Butterworth Low Pass Filter and display the original image and results

```
figure;

subplot(2, 3, 1);
imshow(img, []);
title('Original Image');

for i = 1:length(D0_values)
    D0 = D0_values(i);

    H = 1 ./ (1 + (D / D0).^(2 * n));
```

```
    img_fft = fftshift(fft2(img));
    img_fft_filtered = H .* img_fft;
    img_filtered = abs(ifft2(ifftshift(img_fft_filtered)));

    subplot(2, 3, i+1);
    imshow(img_filtered, []);
    title(['Filtered Image (D0 = ', num2str(D0), ')']);
end
```



Question 3

Implement a 2D Gaussian LPF and apply it to an input image Plot the perspective plot, and filter transfer function as an image. Apply the filter to Figure Fig 0445(a) for order 2 and cut off frequencies as 10, 30, 60, 160 and 460.

Load the input image

```
img = imread('Fig0445(a)(characters_test_pattern).tif');
[M, N] = size(img);
```

Define cutoff frequencies

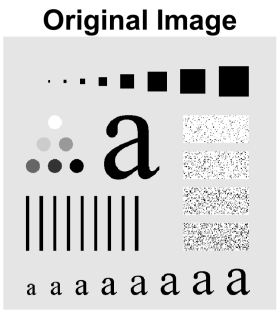
```
D0_values = [10, 30, 60, 160, 460];
```

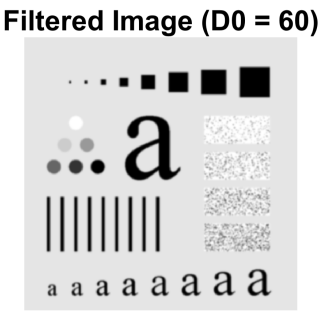
Create frequency coordinates

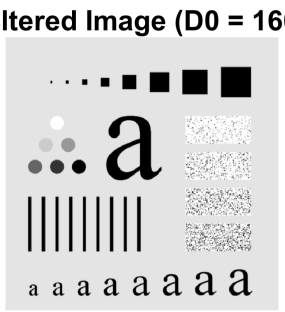
```
[U, V] = meshgrid(-N/2:N/2-1, -M/2:M/2-1);
D = sqrt(U.^2 + V.^2);
```

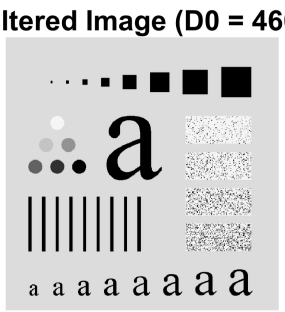
Apply Gaussian Low Pass Filter and display the original image and results

```
figure;
subplot(2, 3, 1);
imshow(img, []);
title('Original Image');

for i = 1:length(D0_values)
    D0 = D0_values(i);

    H = exp(-(D.^2) / (2 * (D0^2)));

    img_fft = fftshift(fft2(img));
    img_fft_filtered = H .* img_fft;
    img_filtered = abs(ifft2(ifftshift(img_fft_filtered)));

    subplot(2, 3, i+1);
    imshow(img_filtered, []);
    title(['Filtered Image (D0 = ', num2str(D0), ')']);
end
```









Question 4

Implement a 2D Ideal HPF. Plot the perspective plot, filter transfer function as an image. Apply the filter to Figure Fig 0441(a) for cut off frequencies D0 = 30, 60 and 160

Load the input image

```
img = imread('Fig0441(a)(characters_test_pattern).tif');
[M, N] = size(img);
```

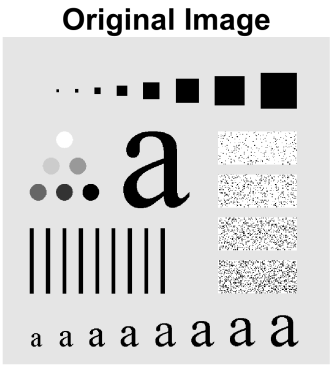
Define cutoff frequencies

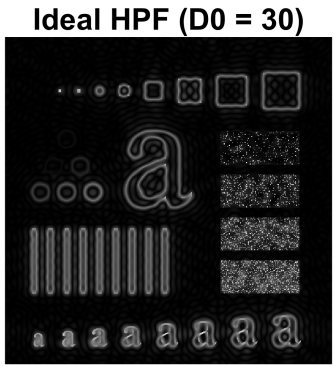
```
D0_values = [30, 60, 160];
```

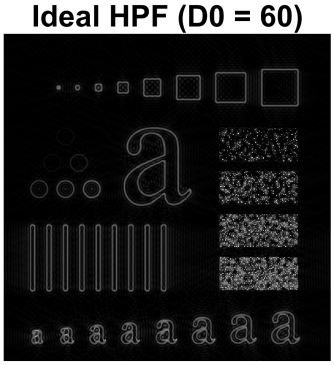
Create frequency coordinates

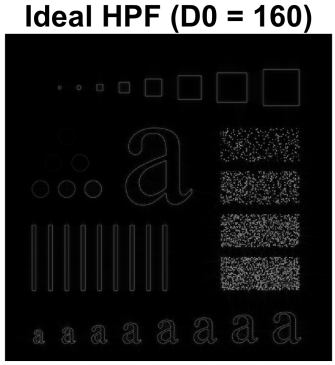
```
[U, V] = meshgrid(-N/2:N/2-1, -M/2:M/2-1);
D = sqrt(U.^2 + V.^2);
```

Apply ideal High Pass Filter and display the original image and results










```
figure;

subplot(2, 2, 1);
imshow(img, []);
title('Original Image');

for i = 1:length(D0_values)
    D0 = D0_values(i);

    H = double(D > D0);
```

```
    img_fft = fftshift(fft2(img));
    img_fft_filtered = H .* img_fft;
    img_filtered = abs(ifft2(ifftshift(img_fft_filtered)));

    subplot(2, 2, i+1);
    imshow(img_filtered, []);
    title(['Ideal HPF (D0 = ', num2str(D0), ')']);
end
```

Question 5

Implement a 2D Butterworth HPF and apply it to an input image Plot the perspective plot, filter transfer function as an image. Apply the filter to Figure Fig 0441(a) for order 2 and cut off frequencies D0 = 30, 60 and 160

Load image

```
img = imread('Fig0441(a)(characters_test_pattern).tif');
[M, N] = size(img);
```

Define cutoff frequencies

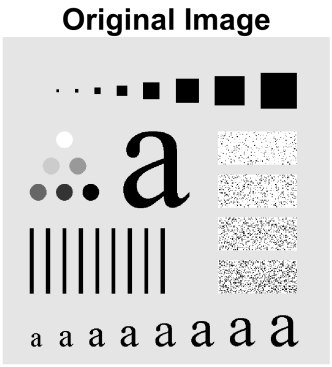
```
D0_values = [30, 60, 160];
n = 2;
```

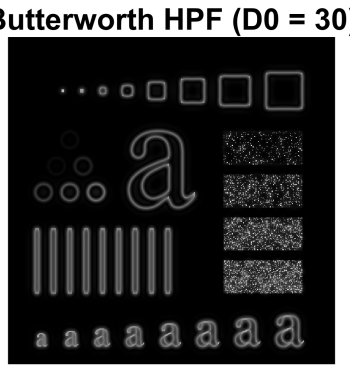
Create frequency coordinates

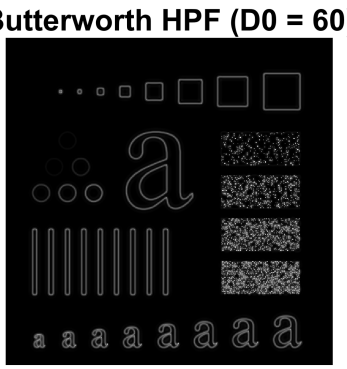
```
[U, V] = meshgrid(-N/2:N/2-1, -M/2:M/2-1);
D = sqrt(U.^2 + V.^2);
```

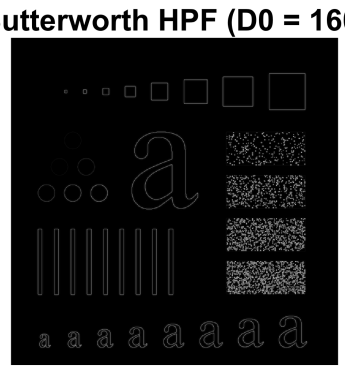
Apply Butterworth High Pass Filter and display the original image and results

```
figure;

subplot(2, 2, 1);
imshow(img, []);
title('Original Image');

for i = 1:length(D0_values)
    D0 = D0_values(i);

    H = 1 ./ (1 + (D0 ./ D).^(2 * n));

    img_fft = fftshift(fft2(img));
    img_fft_filtered = H .* img_fft;
    img_filtered = abs(ifft2(ifftshift(img_fft_filtered)));

    subplot(2, 2, i+1);
    imshow(img_filtered, []);
    title(['Butterworth HPF (D0 = ', num2str(D0), ')']);
end
```









Question 6

Implement a 2D Gaussian HPF and apply it to an input image Plot the perspective plot, and filter transfer function as an image. Apply the filter to Figure Fig 0441(a) for order 2 and cut off frequencies D0 = 30, 60 and 160

Load image

```
img = imread('Fig0441(a)(characters_test_pattern).tif');
[M, N] = size(img);
```

Define cutoff frequencies

```
D0_values = [30, 60, 160];
```

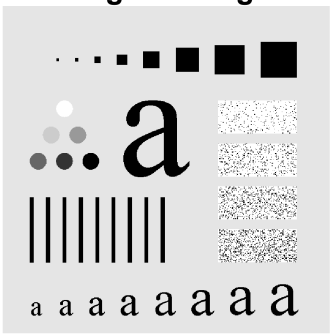
Create frequency coordinates

```
[U, V] = meshgrid(-N/2:N/2-1, -M/2:M/2-1);
D = sqrt(U.^2 + V.^2);
```

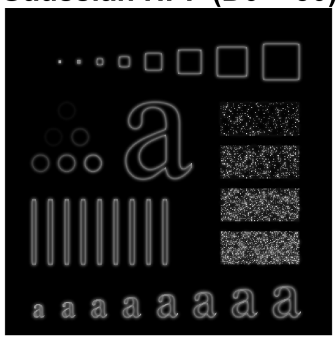
Apply Gaussian High Pass Filter and display the original image and results

```
figure;

subplot(2, 2, 1);
imshow(img, []);
title('Original Image');

for i = 1:length(D0_values)
    D0 = D0_values(i);

    H = 1 - exp(-(D.^2) / (2 * (D0^2)));

    img_fft = fftshift(fft2(img));
    img_fft_filtered = H .* img_fft;
    img_filtered = abs(ifft2(ifftshift(img_fft_filtered)));

    subplot(2, 2, i+1);
    imshow(img_filtered, []);
    title(['Gaussian HPF (D0 = ', num2str(D0), ')']);
end
```

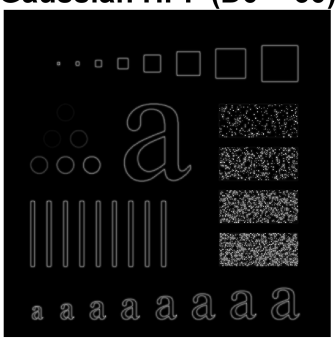
Original Image



Gaussian HPF (D0 = 30)



Gaussian HPF (D0 = 60)



Gaussian HPF (D0 = 160)