Tutorial 3
Image Enhancement in the Frequency Domain

COMP 4421: Image Processing

September 22, 2019

Outline

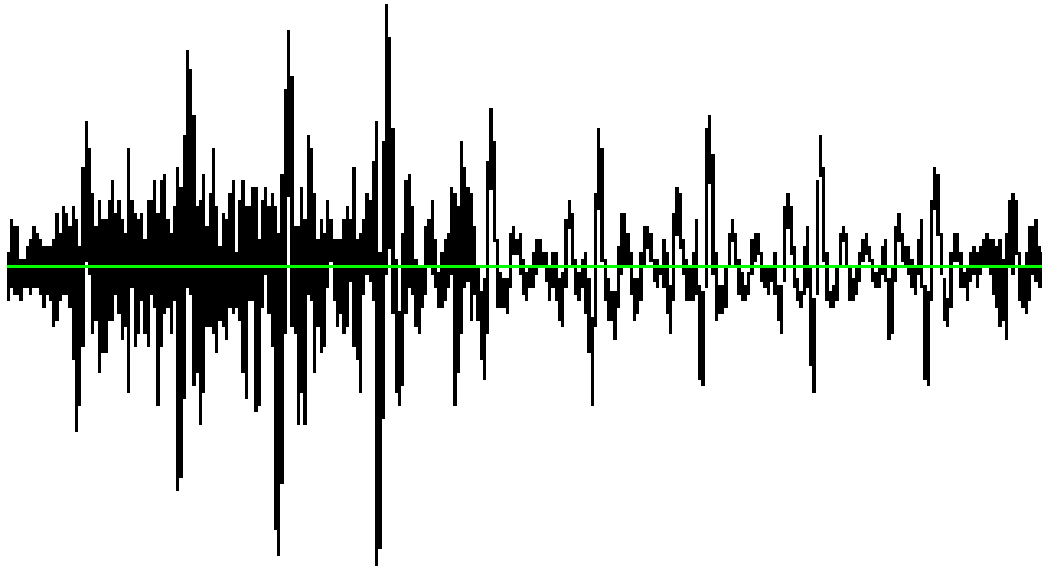
- Fourier Transform
  - Fourier Transform of 1D Signal
  - Fourier Transform of a Synthetic Image
- Low-Pass Filter
  - Ideal Low-pass Filter (ILPF)
  - Butterworth Low-pass Filter (BLPF)
  - Gaussian Low-pass Filter (GLPF)
- High-Pass Filter
  - Ideal High-pass Filter (IHPF)
  - Butterworth High-pass Filter (BHPF)
  - Gaussian High-pass Filter (GHPF)

Outline

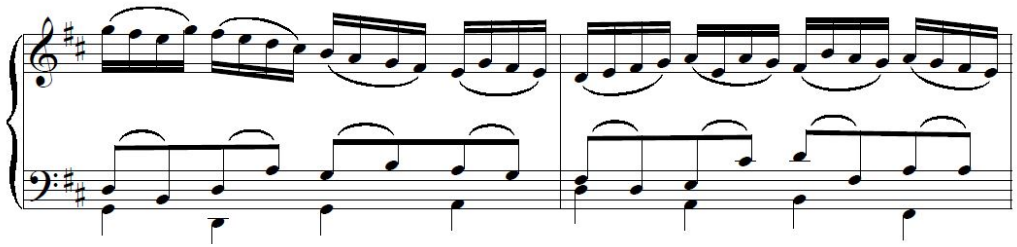
- Fourier Transform
  - Fourier Transform of 1D Signal
  - Fourier Transform of a Synthetic Image
- Low-Pass Filter
- High-Pass Filter

Fourier Transform of 1D Signal

- Time Domain



- Frequency Domain

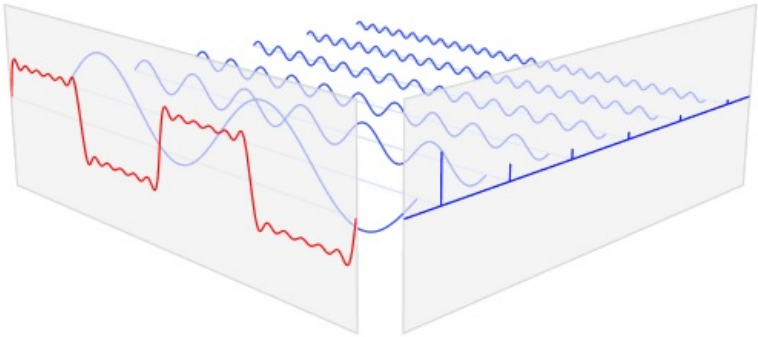


Fourier Transform of 1D Signal

- Any periodic signal can be decomposed to the sum of a set of sine functions with different magnitudes and phases.
- Any music can be decomposed to the combination of a set of keys pressed with various strengths and at different time points.

http://en.wikipedia.org/wiki/Fourier_transform
https://www.mathworks.com/help/signal/ug/discrete-fourier-transform.html

Fourier Transform of 1D Signal



Fourier Transform of 1D Signal

```
M = 1000;
f = zeros(1, M);
l = 10;
f(M/2-l:M/2+l) = 1;
figure,plot(f);
F = fft(f);
Fc = fftshift(F);
rFc = real(Fc);
iFc = imag(Fc);
figure,subplot(2,1,1),plot(abs(F));
subplot(2,1,2),plot(abs(Fc));title
```

Create a simple rectangular 1D signal and examine its Fourier Transform.

Fourier Transform of 1D Signal

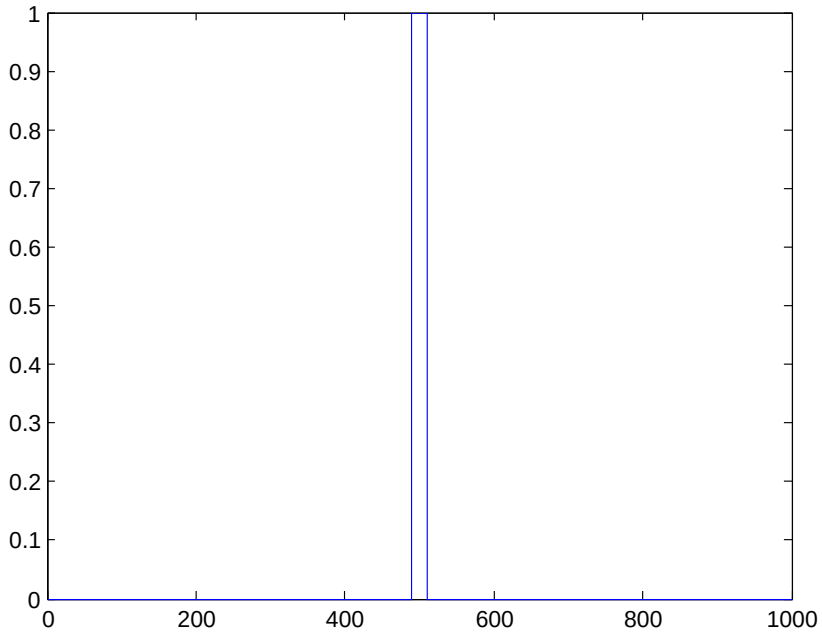


Figure: 1D Signal

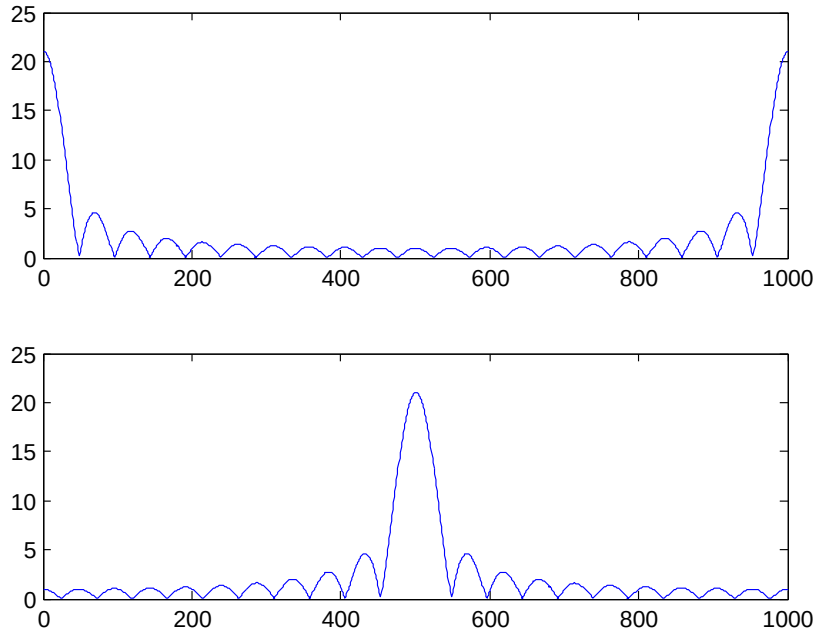
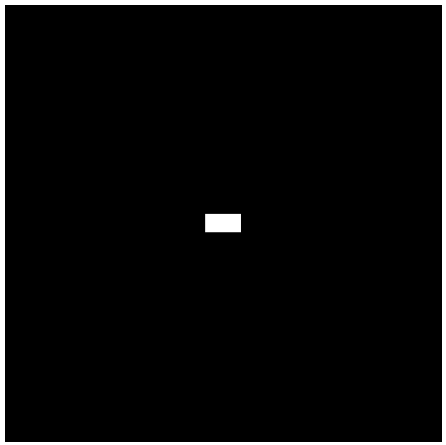


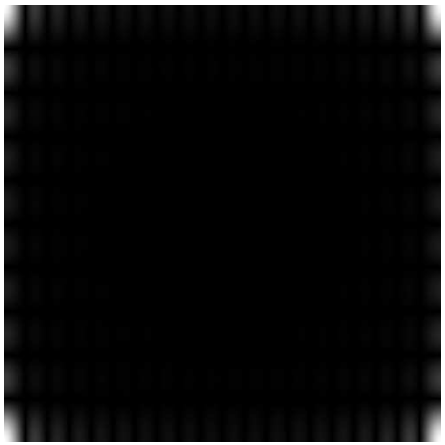
Figure: Spectrum

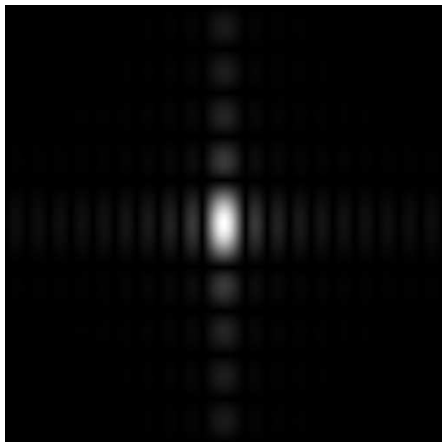
Fourier Transform of a Synthetic Image

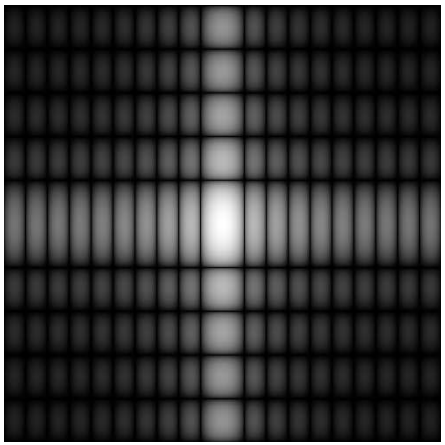
```
f = ones(10,20);
F = fft2(f, 500,500);
f1 = zeros(500,500);
f1(240:260,230:270) = 1;
subplot(2,2,1);imshow(f1,[]);
S = abs(F);
subplot(2,2,2); imshow(S,[]);
Fc = fftshift(F);
S1 = abs(Fc);
subplot(2,2,3); imshow(S1,[]);
S2 = log(1+S1);
subplot(2,2,4);imshow(S2,[]);
```

Fourier Transform of a Synthetic Image





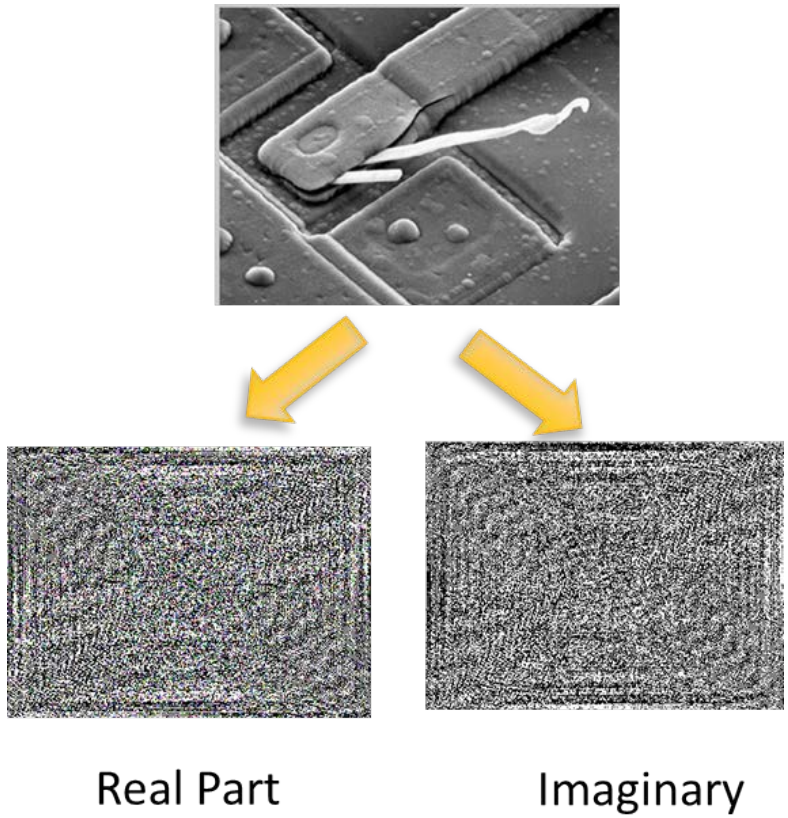




Fourier Transform

Matlab Code

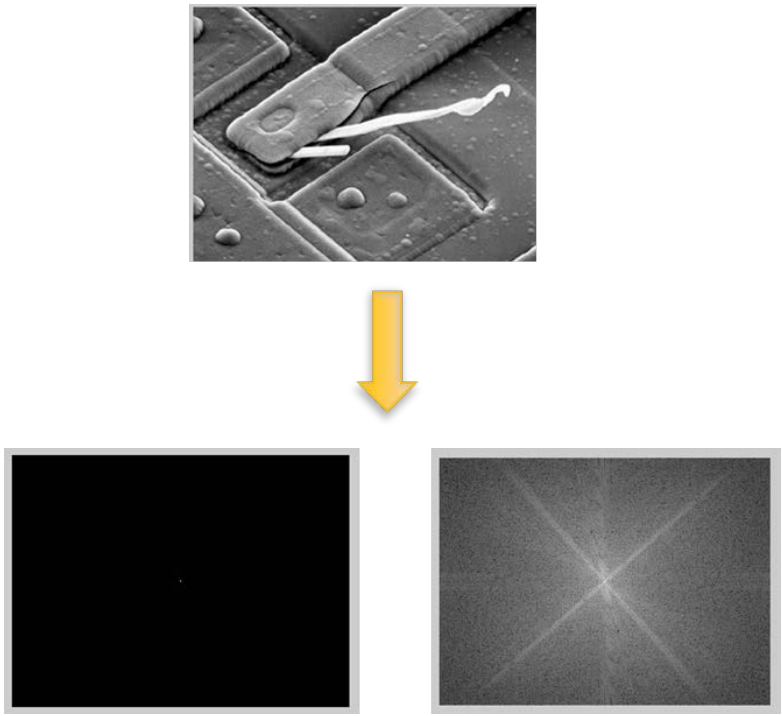
```
Im=imread ('example.bmp');
ft= fft2(Im);
figure,imshow(real(ft));
figure,imshow(imag(ft));
```



Fourier Transform

Matlab Code

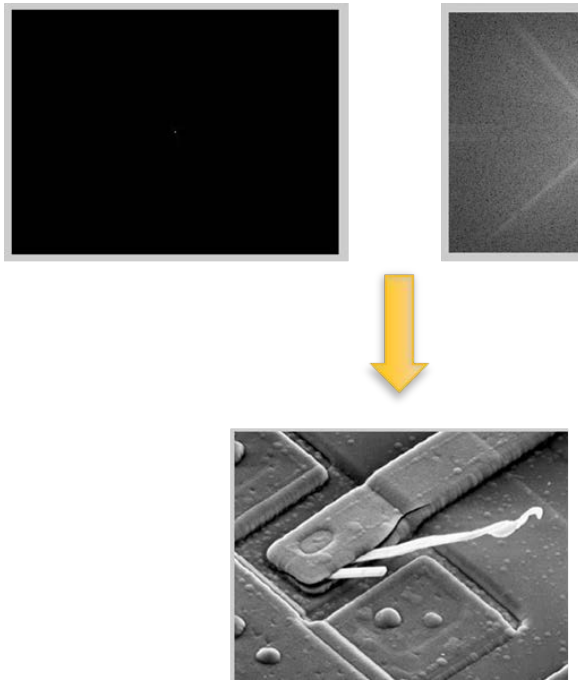
```
Im=imread('example.bmp');
ft= fft2(Im);
fts = fftshift(ft);
figure;imshow(abs(fts),[]);
figure;imshow(log(1+abs(fts)),[]);
```



Fourier Transform

Matlab Code

```
orift=ifftshift(fts);
oriIm=ifft2(orift);
figure;imshow(oriIm,[]);
```



Review of frequently-used funcions

- 2D Fourier transform: F = fft2(f);
- Spectrum shift: Fs=fftshift(F);
  Shift zero-frequency component to center of spectrum.
- Absolute value: Fm=abs(F);
  Return spectrum of F if it is complex
- Real or imaginary part of complex signal: real(F); imag(F);
- Demonstrating 2D signal(matrix): imshow(Fm, [])

Outline

- Fourier Transform
- Low-Pass Filter
  - Ideal Low-pass Filter (ILPF)
  - Butterworth Low-pass Filter (BLPF)
  - Gaussian Low-pass Filter (GLPF)
- High-Pass Filter
  - Ideal High-pass Filter (IHPF)
  - Butterworth High-pass Filter (BHPF)
  - Gaussian High-pass Filter (GHPF)

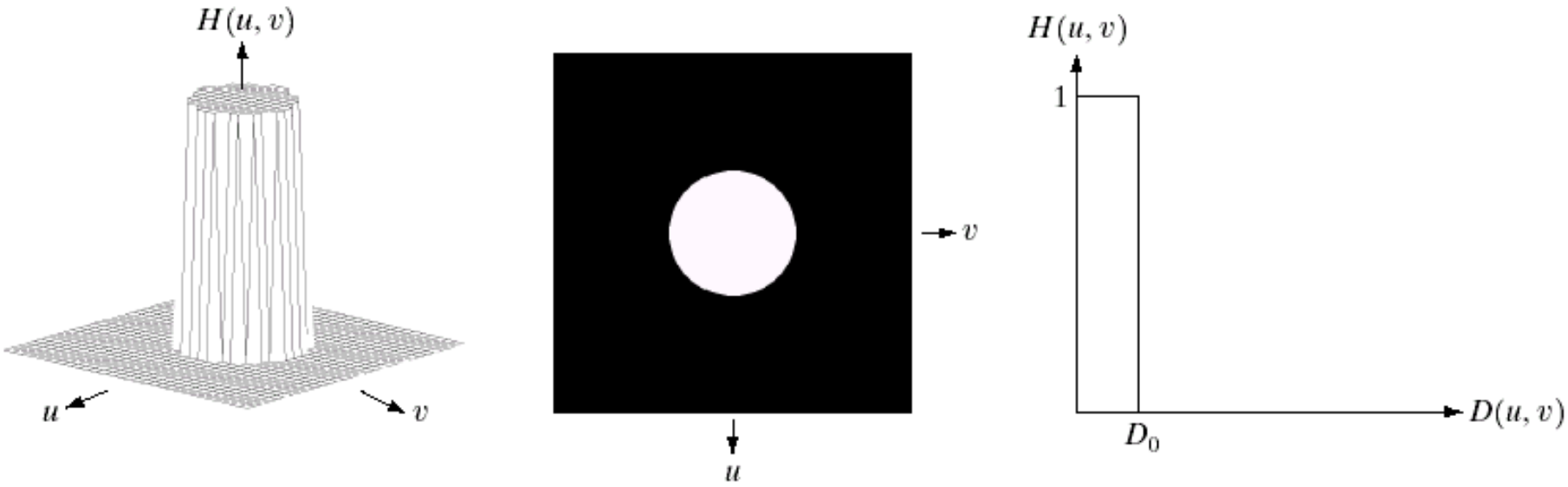
Ideal Low-pass Filter (ILPF)

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

- Where $D(u,v)$ is the distance from point (u,v) to the origin of the frequency plane, $D(u,v) = \sqrt{u^2 + v^2}$.
- And D_0 is a nonnegative quantity, cutoff frequency.

Ideal Low-pass Filter (ILPF)

>> idealfilter.m



a b c

FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Ideal Low-pass Filter (ILPF)

Original Image and results of filtering with ILPF, with $D_0 = 5, 15, 30, 80, 230$ respectively.

Outline

- Fourier Transform
- Low-Pass Filter
  - Ideal Low-pass Filter (ILPF)
  - Butterworth Low-pass Filter (BLPF)
  - Gaussian Low-pass Filter (GLPF)
- High-Pass Filter
  - Ideal High-pass Filter (IHPF)
  - Butterworth High-pass Filter (BHPF)
  - Gaussian High-pass Filter (GHPF)

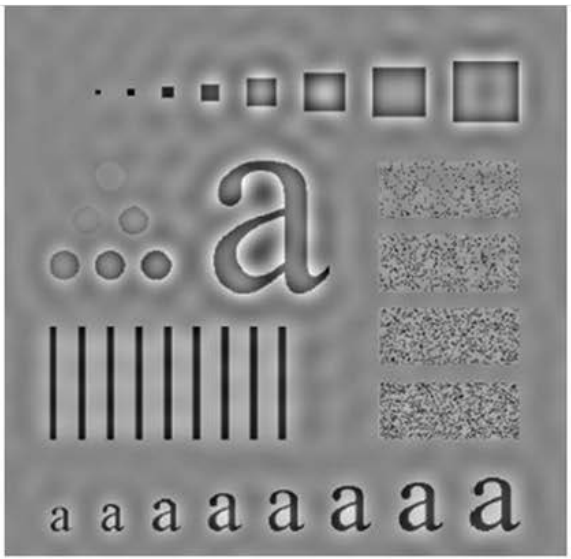
Ideal High-pass Filter (IHPF)

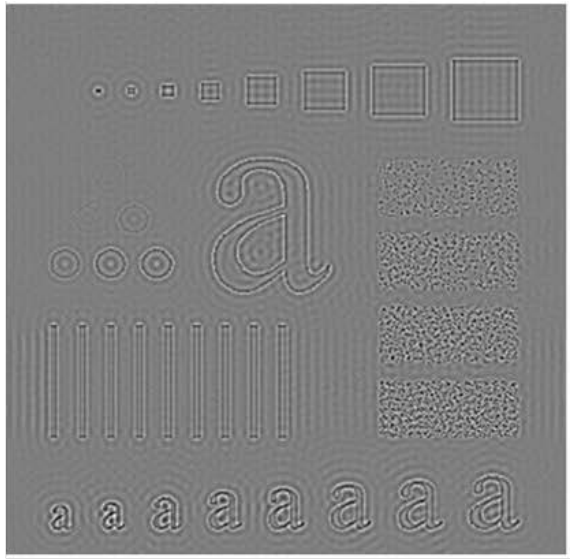
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

- Where $D(u,v)$ is the distance from point (u,v) to the origin of the frequency plane, $D(u,v) = \sqrt{u^2 + v^2}$.
- And D_0 is a nonnegative quantity, cutoff frequency.

- Relation between low-pass and high-pass filters
  $H_{hp}(u,v) = 1 - H_{lp}(u,v)$.

Ideal High-pass Filter (IHPF)





Results of filtering with IHPF , with $D_0 = 15, 30, 60$ respectively.

Outline

- Fourier Transform
- Low-Pass Filter
  - Ideal Low-pass Filter (ILPF)
  - Butterworth Low-pass Filter (BLPF)
  - Gaussian Low-pass Filter (GLPF)
- High-Pass Filter
  - Ideal High-pass Filter (IHPF)
  - Butterworth High-pass Filter (BHPF)
  - Gaussian High-pass Filter (GHPF)

Butterworth Low-pass Filter (BLPF)

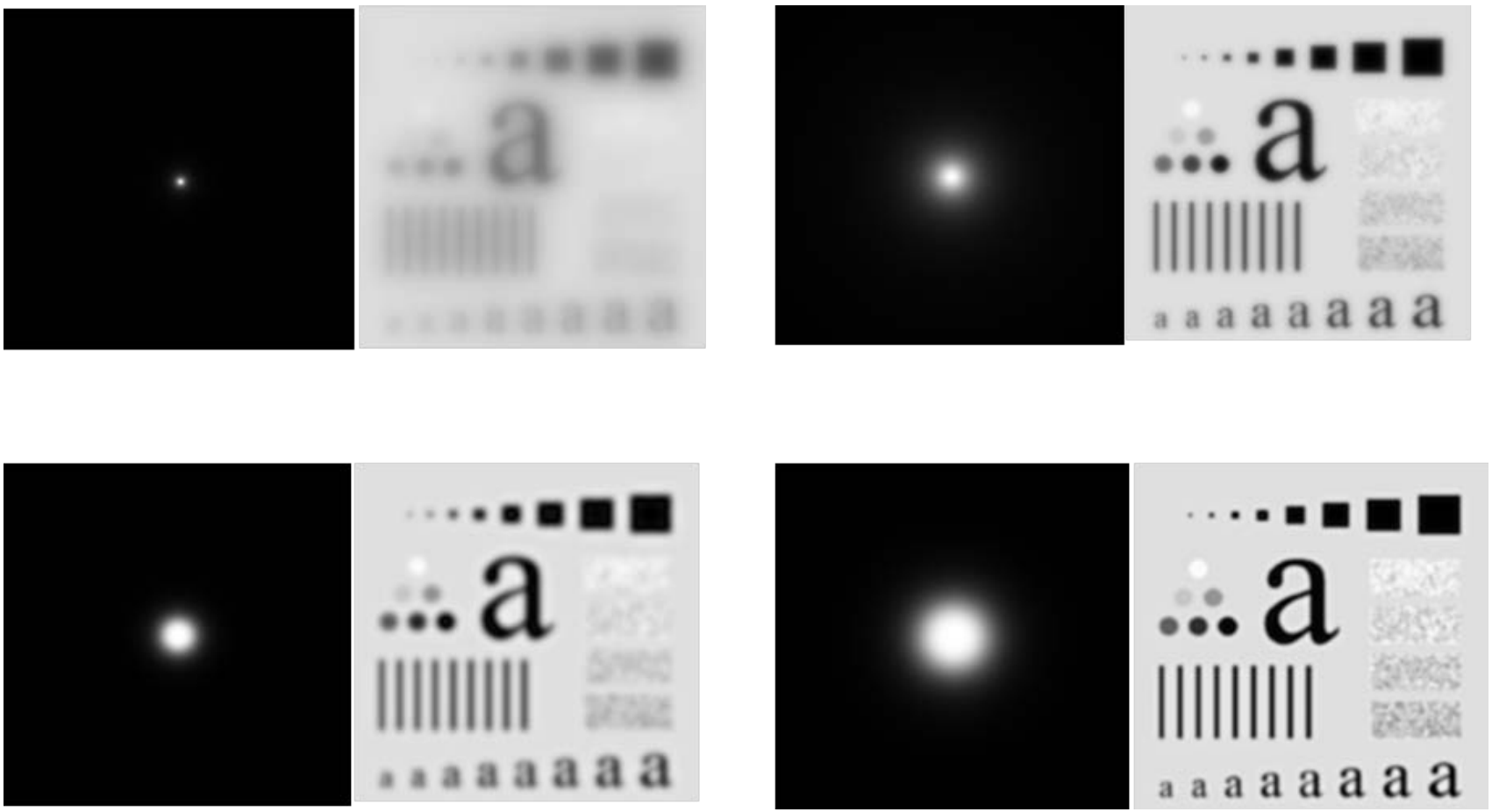
```
f = imread('A.bmp');
f = rgb2gray(f);
f = double(f);
F = fftshift(fft2(f));
[m, n] = size(f);
H=ButterWorth([m, n], 0.05, 2);
G = H.*F;
figure,imshow(H);
figure,mesh(H);
g = abs(ifft2(G));
g = uint8(g);
figure,imshow(g);
```

Butterworth Low-pass Filter (BLPF)

```
function f = ButterWorth(sze, cutoff, n)

if (length(sze) == 1)    rows = sze; cols = sze;
else    rows = sze(1); cols = sze(2);
end
if (mod(cols,2))    xrange = [-(cols-1)/2:(cols-1)/2]/(cols-1);
else    xrange = [-cols/2:(cols/2-1)]/cols;
end
if (mod(rows,2))    yrange = [-(rows-1)/2:(rows-1)/2]/(rows-1);
else    yrange = [-rows/2:(rows/2-1)]/rows;
end
[x, y] = meshgrid(xrange, yrange);
radius = sqrt(x.^2 + y.^2);
f = 1.0 ./ (1.0 + (radius ./ cutoff).^(2*n));
```

Butterworth Low-pass Filter (BLPF)



Results of filtering with BLPF, with $n = 1, D_0 = 0.01; n = 1, D_0 = 0.05;$ $n = 2, D_0 = 0.05; n = 2, D_0 = 0.1$ respectively.

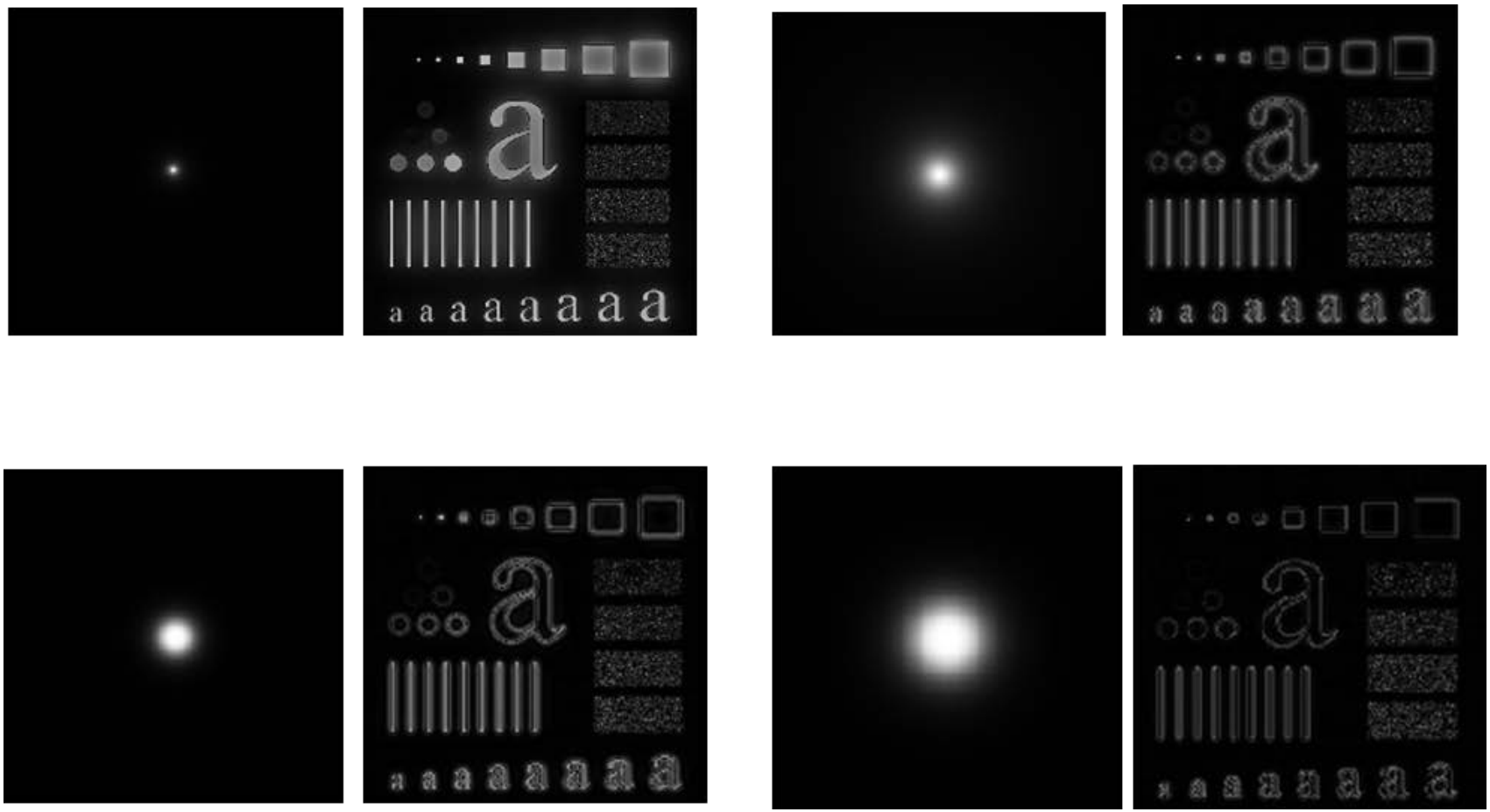
Outline

- Fourier Transform
- Low-Pass Filter
  - Ideal Low-pass Filter (ILPF)
  - Butterworth Low-pass Filter (BLPF)
  - Gaussian Low-pass Filter (GLPF)
- High-Pass Filter
  - Ideal High-pass Filter (IHPF)
  - Butterworth High-pass Filter (BHPF)
  - Gaussian High-pass Filter (GHPF)

Butterworth High-pass Filter (BHPF)

```
f = imread('A.bmp');
f = rgb2gray(f);
f = double(f);
F = fftshift(fft2(f));
[m, n] = size(f);
H=ButterWorth([m, n], 0.1, 2);
figure,imshow(H);
I = ones(m,n);
G = (I-H).*F;
g = abs(ifft2(G));
g = uint8(g);
figure,imshow(g);
```

Butterworth High-pass Filter (BHPF)



Results of filtering with BHPF, with $n = 1, D_0 = 0.01; n = 1, D_0 = 0.05;$ $n = 2, D_0 = 0.05; n = 2, D_0 = 0.1$ respectively.

Outline

- Fourier Transform
- Low-Pass Filter
  - Ideal Low-pass Filter (ILPF)
  - Butterworth Low-pass Filter (BLPF)
  - Gaussian Low-pass Filter (GLPF)
- High-Pass Filter
  - Ideal High-pass Filter (IHPF)
  - Butterworth High-pass Filter (BHPF)
  - Gaussian High-pass Filter (GHPF)

Gaussian Low-pass Filter (GLPF)

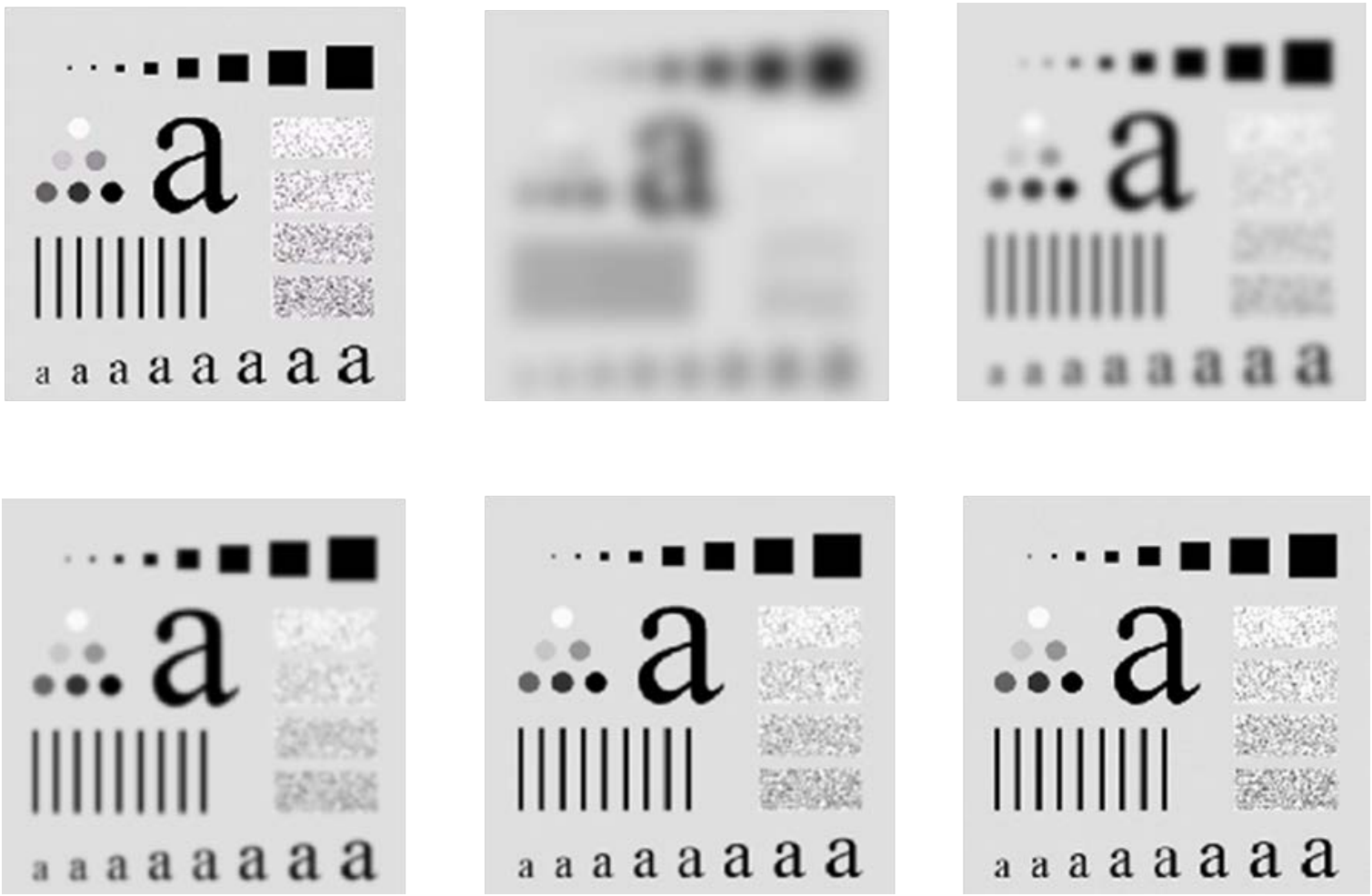
```
f = imread('A.bmp');
f = rgb2gray(f);
f = double(f);
F = fftshift(fft2(f));
[m, n] = size(f);
sig = 50;
H = Gaussian(m, n, sig);
G = H.*F;
g = abs(ifft2(G));
g = uint8(g);
figure,imshow(g);
```

Gaussian Low-pass Filter (GLPF)

```
function f = Gaussian(M, N, sig)

if (mod(M, 2) == 0)    cM = floor(M/2) + 0.5;
else    cM = floor(M/2) + 1;
end
if (mod(N, 2) == 0)    cN = floor(N/2) + 0.5;
else    cN = floor(N/2) + 1;
end
a = [1:M];
b = [1:N];
A = repmat(a',1,N);
B = repmat(b,M,1);
A = (A-cM).^2;
B = (B-cN).^2;
f = exp(-(A+B)./(2*sig^2));
```

Gaussian Low-pass Filter (GLPF)



Original Image and results of filtering with GLPF of order 2, with $D_0 = 5, 15, 30, 80, 230$ respectively.

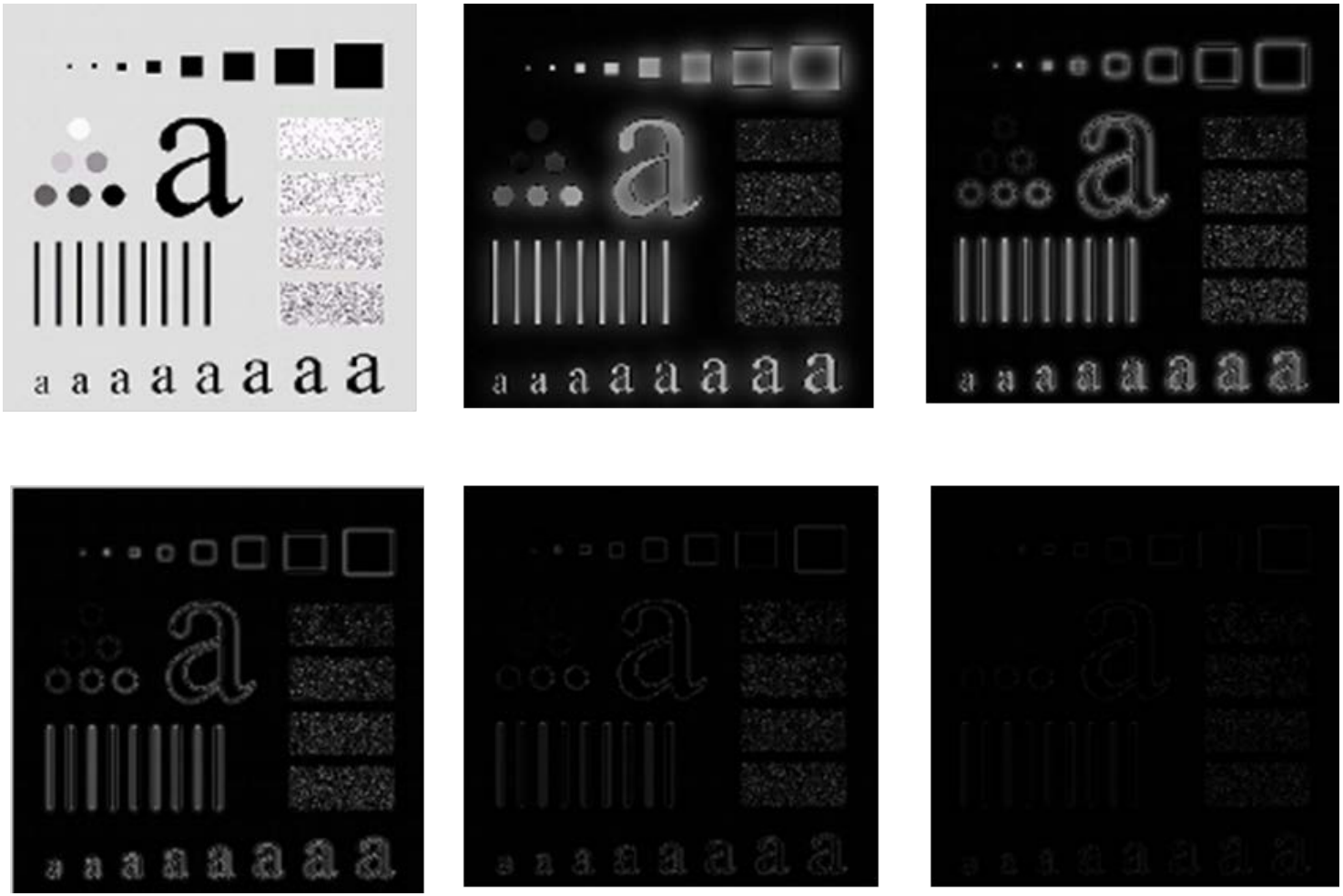
Outline

- Fourier Transform
- Low-Pass Filter
  - Ideal Low-pass Filter (ILPF)
  - Butterworth Low-pass Filter (BLPF)
  - Gaussian Low-pass Filter (GLPF)
- High-Pass Filter
  - Ideal High-pass Filter (IHPF)
  - Butterworth High-pass Filter (BHPF)
  - Gaussian High-pass Filter (GHPF)

Gaussian High-pass Filter (GHPF)

```
f = imread('A.bmp');
f = rgb2gray(f);
f = double(f);
F = fftshift(fft2(f));
[m, n] = size(f);
sig = 50;
H = Gaussian(m, n, sig);
I = ones(m,n);
G = (I-H).*F;
g = abs(ifft2(G));
g = uint8(g);
figure,imshow(g);
```

Gaussian High-pass Filter (GHPF)



Original Image and results of filtering with GHPF of order 2, with $D_0 = 5, 15, 30, 80, 150$ respectively.

Thank you!