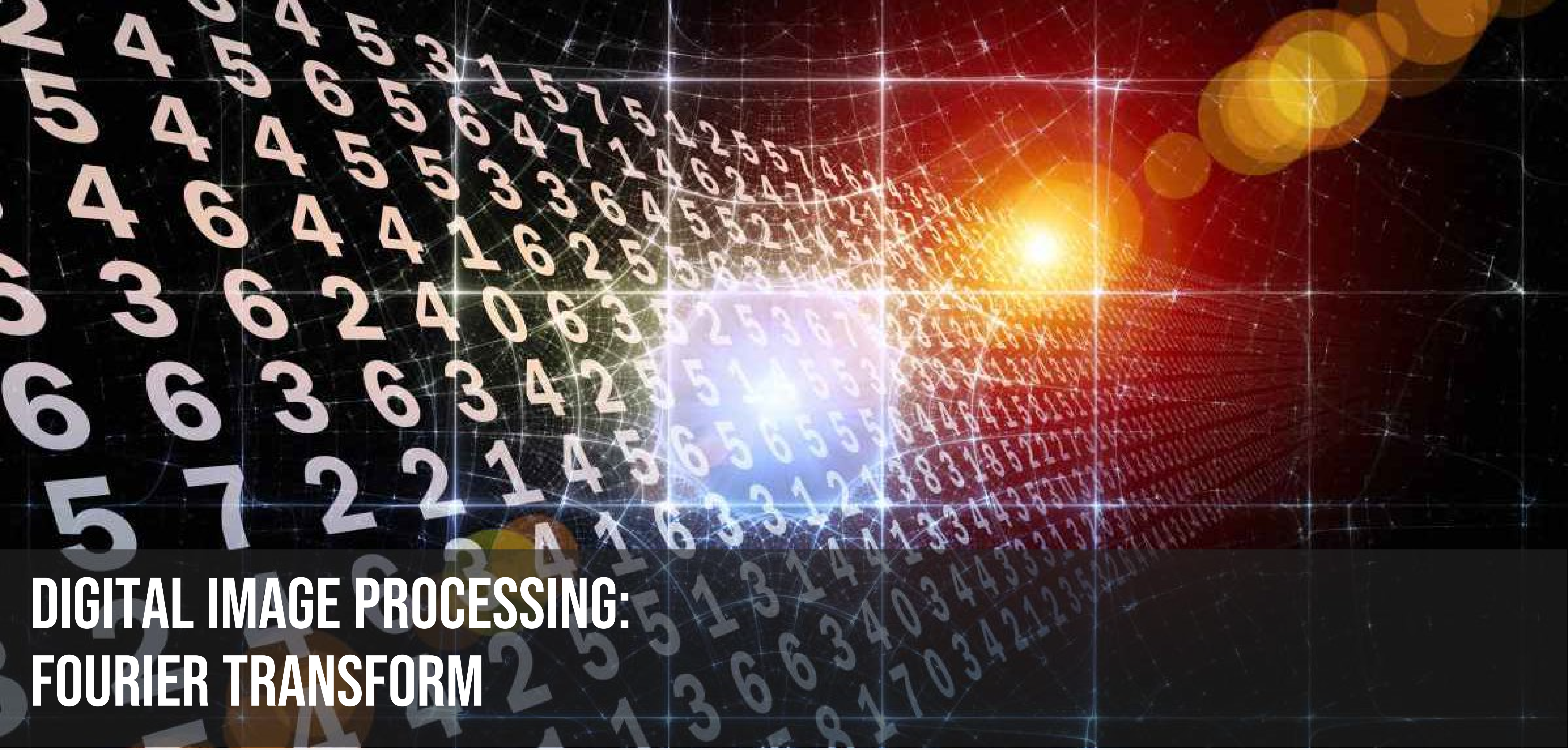
PENGOLAHAN SINYAL DIGITAL

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia



DIGITAL IMAGE PROCESSING: FOURIER TRANSFORM

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

INTRODUCTION

- The Fourier Transform is generally used to decompose a signal into various sinusoidal components.
- For an image, the output of the transformation is the representation of the image in frequency space, while the input image is the real space equivalent.
- In the Fourier space image, each point represents a particular frequency contained in the real domain image.

Jean Baptiste Joseph Fourier

SIGNIFICANCE

- The Fourier Transform allows us to perform tasks which would be impossible to perform any other way; its efficiency allows us to perform other tasks more quickly.

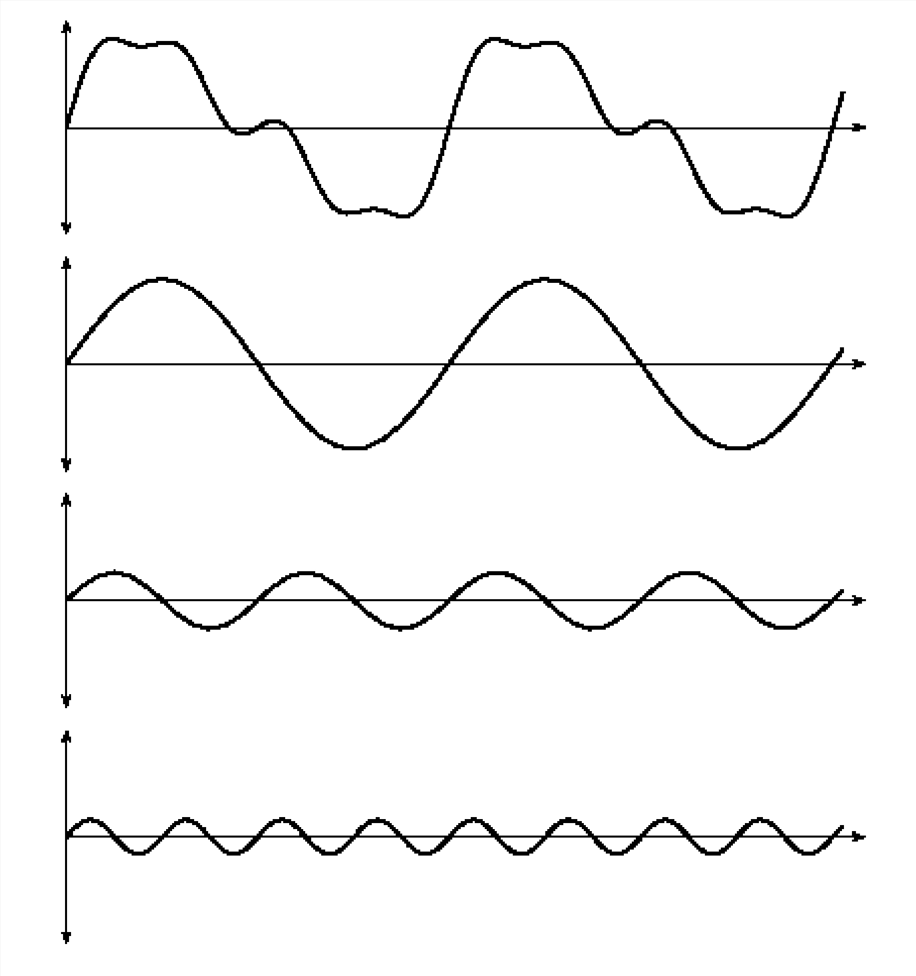
- The Fourier Transform provides a powerful alternative to linear spatial filtering; it is more efficient to use the Fourier transform than a spatial filter for a large filter.

- The Fourier Transform also allows us to isolate and process particular image frequencies, and so perform low-pass and high-pass filtering with a great degree of precision.

SOME INTUITION

- A periodic function may be written as the sum of sines and cosines of varying amplitudes and frequencies.

- Examples =>



$$f(x) = \sin x + \frac{1}{3}\sin 2x + \frac{1}{5}\sin 4x$$

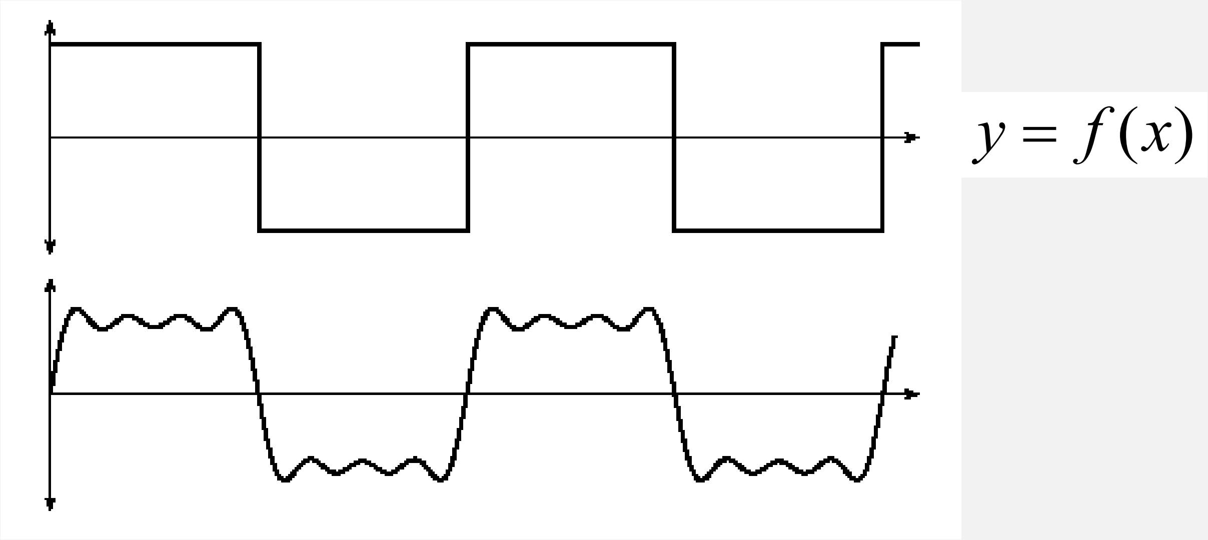
$$y = \sin x$$

$$y = \frac{1}{3}\sin 2x$$

$$y = \frac{1}{5}\sin 4x$$

SOME INTUITION

- Some functions will require only a finite number of functions in their decomposition; others will require an infinite number.



$$f\left(x\right) = \sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x + \frac{1}{7}\sin 7x + \frac{1}{9}\sin 9x + \ldots$$

1-D CONTINUOUS

- $f(x)$ is a linear combination of simple periodic patterns.

Spatial frequency (measured in whole cycle per unit of x)

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{i2\pi xu} \, du$$

Inverse Fourier transform

Simple periodic patterns

Weight function for the given frequency

- where

Image co-ordinate

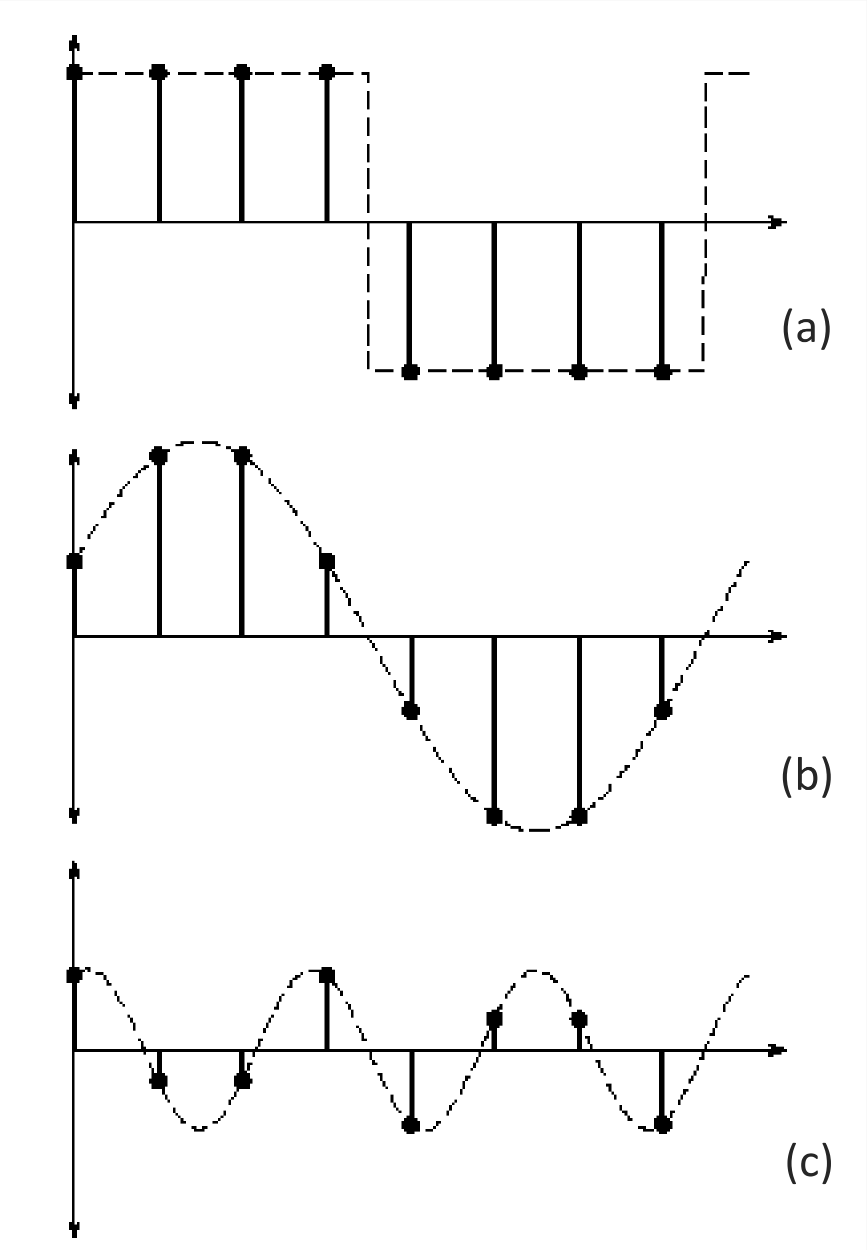
$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi xu} \, dx$$

Fourier transform

$\sqrt{-1}$

1-D DISCRETE (CON'D)

- In image processing, we deal with a discrete function.
- Since we only have to obtain a finite number of values, we only need a finite number of functions to do it.
- For example: 1 1 1 1 -1 -1 -1 -1, which we may take as a discrete pproximation to the square wave of figure (a). This can be expressed as the sum of two sine functions, (b) and (c)



DEFINITION OF 1-D DFT

Suppose $f = [f_0, f_1, f_2, \cdots, f_{N-1}]$

is a sequence of length N. Define its discrete Fourier transform

$$F = [F_0, F_1, F_2, \cdots, F_{N-1}]$$

where $$F_u = \frac{1}{N}\sum_{x=0}^{N-1} \exp[-2\pi i \frac{xu}{N}] \cdot f_x$$

We can express this definition as matrix multiplication $F = \mathbb{F} f$

Where F is an NxN matrix defined by $\mathbb{F}_{m,n} = \frac{1}{N}\exp[-2\pi i \frac{mn}{N}]$

DEFINITION OF 1-D DFT

Given N, we shall define $\omega = \exp[\frac{-2\pi i}{N}]$

So that $\mathbb{F}_{m,n} = \frac{1}{N}\omega^{mn}$ $\mathbb{F}_{m,n} = \frac{1}{N}\exp[-2\pi i\frac{mn}{N}]$

Then we can write $\mathbb{F} = \frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \omega^2 & \omega^3 & \omega^4 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \omega^8 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \omega^{12} & \cdots & \omega^{3(N-1)} \\ 1 & \omega^4 & \omega^8 & \omega^{12} & \omega^{16} & \cdots & \omega^{4(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \omega^{4(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix}$

EXAMPLE

Suppose $f = [1, 2, 3, 4]$ so that N=4. Then

$$\begin{aligned}\omega &= \exp[\frac{-2\pi i}{4}] \\ &= \exp[\frac{-\pi i}{2}] \\ &= \cos(-\frac{\pi}{2}) + i\sin(-\frac{\pi}{2}) \\ &= -i\end{aligned}$$

Then we have

$$\mathbb{F} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & (-i)^2 & (-i)^3 \\ 1 & (-i)^2 & (-i)^4 & (-i)^6 \\ 1 & (-i)^3 & (-i)^6 & (-i)^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}$$

$$\mathrm{F} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 10 \\ -2+2i \\ -2 \\ -2-2i \end{bmatrix}$$

$$F = \mathbb{F}f$$

THE INVERSE DFT

Inverse DFT: $f_x = \sum_{u=0}^{N-1} \exp\left[2\pi i \frac{xu}{N}\right] F_u$

Difference with forward transform:

(1). There is no scaling factor 1/N
(2). The sign inside the exponential function has been changed to positive

Inverse DFT can also be expressed as matrix product

$$f = F^{-1}F \qquad F^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \overline{\omega}^1 & \overline{\omega}^2 & \overline{\omega}^3 & \overline{\omega}^4 & \cdots & \overline{\omega}^{N-1} \\ 1 & \overline{\omega}^2 & \overline{\omega}^4 & \overline{\omega}^6 & \overline{\omega}^8 & \cdots & \overline{\omega}^{2(N-1)} \\ 1 & \overline{\omega}^3 & \overline{\omega}^6 & \overline{\omega}^9 & \overline{\omega}^{12} & \cdots & \overline{\omega}^{3(N-1)} \\ 1 & \overline{\omega}^4 & \overline{\omega}^8 & \overline{\omega}^{12} & \overline{\omega}^{16} & \cdots & \overline{\omega}^{4(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \overline{\omega}^{N-1} & \overline{\omega}^{2(N-1)} & \overline{\omega}^{3(N-1)} & \overline{\omega}^{4(N-1)} & \cdots & \overline{\omega}^{(N-1)^2} \end{bmatrix}$$

where

$$\overline{w} = \frac{1}{w} = \exp\left[\frac{2\pi i}{N}\right]$$

MATLAB FUNCTIONS: *FFT/IFFT*

Example:

```
» a=[1 2 3 4 5 6 7 8 9]
» b=fft(a)
  45.0          -4.5 +12.3i   -4.5 + 5.3i   -4.5 + 2.5i
 -4.5 + 0.7i  -4.5 - 0.7i   -4.5 - 2.5i   -4.5 - 5.3i
 -4.5 -12.3i

» ifft(b)
  1.0 - 0.0i   2.0 - 0.0i    3.0  - 0.0i   4.0  - 0.0i
  5.0- 0.0i    6.0 - 0.0i    7.0  + 0.0i   8.0  + 0.0i
  9.0 + 0.0i
```

PROPERTIES OF 1-D DFT

- Linearity: Suppose **f** and **g** are two vectors of same length, and **p** and **q** are scalars, with **h = pf + qg**. If **F,G** and **H** are the DFT's of **f,g** and **h**, then

$$H = pF + qG$$

- Shifting: Suppose we multiply each element x_n of a vector x by $(-1)^n$ i.e., we change the sign of every second element. Let the resulting vector be denoted x'. Then DFT X' of x' is equal to the DFT X of x with the swapping of the left and right halves.

EXAMPLE

```
» x = [ 2  3  4  5  6  7  8  1]
» x1 = (-1).^[0:7].*x
x1 = 2  -3   4  -5   6  -7   8  -1
```

```
» X=fft(x')

X =
  36.0000
  -9.6569 + 4.0000i
  -4.0000 - 4.0000i
   1.6569 - 4.0000i
   4.0000
   1.6569 + 4.0000i
  -4.0000 + 4.0000i
  -9.6569 - 4.0000i
```

```
» X1=fft(x1')

X1 =
   4.0000
   1.6569 + 4.0000i
  -4.0000 + 4.0000i
  -9.6569 - 4.0000i
  36.0000
  -9.6569 + 4.0000i
  -4.0000 - 4.0000i
   1.6569 - 4.0000i
```

Then the DFT X1 of x1 is equal to the DFT X of x with the swapping of the left and right halves.

PROPERTIES OF 1-D DFT (CON'D)

- Conjugate symmetry: If x is real, and of length N, then its DFT **X** satisfies the condition $X_k = \overline{X_{N-k}}$,where $\overline{X_{N-k}}$ is the complex conjugate of X_{N-k} for all k=1,2,3,…,N-1. (check out previous slide)

- Circular convolution: Suppose x and y are two vectors of the same length N. Then we define their convolution to be the vector, $z = x * y$, where

$$z_k = \frac{1}{n} \sum_{n=0}^{N-1} x_n y_{k-n}$$

EXAMPLE

$$z_0 = \frac{1}{4}(x_0 y_0 + x_1 y_{-1} + x_2 y_{-2} + x_3 y_{-3})$$

$$z_1 = \frac{1}{4}(x_0 y_1 + x_1 y_0 + x_2 y_{-1} + x_3 y_{-2})$$

$$z_2 = \frac{1}{4}(x_0 y_2 + x_1 y_1 + x_2 y_0 + x_3 y_{-1})$$

$$z_3 = \frac{1}{4}(x_0 y_3 + x_1 y_2 + x_2 y_1 + x_3 y_0)$$

$$\bullet\bullet\bullet \; y_0 \; y_1 \; y_2 \; y_3 \; y_0 \; y_1 \; y_2 \; y_3 \; \bullet\bullet\bullet$$

$$= \bullet\bullet\bullet \; y_0 \; y_{-3} \; y_{-2} \; y_{-1} \; y_0 \; y_1 \; y_2 \; y_3 \; \bullet\bullet\bullet$$

- Thus

$$y_{-1} = y_3 \;, y_{-2} = y_2 \;, \quad y_{-3} = y_1$$

PROPERTIES OF 1-D DFT (CON'D)

- Circular convolution: can be defined in terms of polynomial products.
  - Suppose p(u) the polynomial in u whose coefficients are elements of x. Let q(u) be the polynomial whose coefficients are elements of y. From the product $p(u)q(u)(1+u^N)$, and extract the coefficients of u^N to u^{2N-1}, these will be the required circular convolution
- Example: $x=[1,2,3,4], \quad y=[5,6,7,8]$
  - We have $p(u)=1+2u+3u^2+4u^3$ and $q(u)=5+6u+7u^2+8u^3$
  - Then we expand

$$p(u)q(u)(1+u^4)=5+16u+34u^2+60u^3+66u^4+68u^5+66u^6+60u^7+61u^8+52u^9+32u^{10}$$

- Extracting the coefficients of u4,u5,…., u7 we obtain

$$x*y=[66,68,66,60]$$

IMPORTANCE OF CONVOLUTION

- Suppose x and y are vectors of equal length. Then the DFT of their circular convolution is equal to the element-by-element product of the DFT's of x and y.
- If Z,X,Y are the DFT's of z=x*y, x and y respectively, then Z=X.Y
- Example:

```
» fft(cconv(a,b)')
ans =

  1.0e+002 *

   2.6000
  -0.0000 - 0.0800i
   0.0400
  -0.0000 + 0.0800i
```

```
» fft(a').*fft(b')
ans =
  1.0e+002 *

   2.6000
  -0.0000 - 0.0800i
   0.0400
  -0.0000 + 0.0800i
```

MORE ON DFT

- In general, the transform into the frequency domain will be a complex valued function, that is, with magnitude and phase.

$$\text{magnitude} \;=\; \left\|F_u\right\| = \sqrt{F_{real} * F_{real} + F_{imag} * F_{imag}}$$

$$\text{phase} \;=\; \tan^{-1}\left(\frac{F_{imag}}{F_{real}}\right)$$

$$F_u = \left\|F_u\right\| \exp\left(i\theta\right)$$

- The DC coefficient: The value $F(0)$ average of the input series.

$$F(0) = \frac{1}{N}\sum_{x=0}^{N-1} f_x \exp(0) = \frac{1}{N}\sum_{x=0}^{N-1} f_x$$

SOME PROPERTIES OF TRANSFORM PAIR

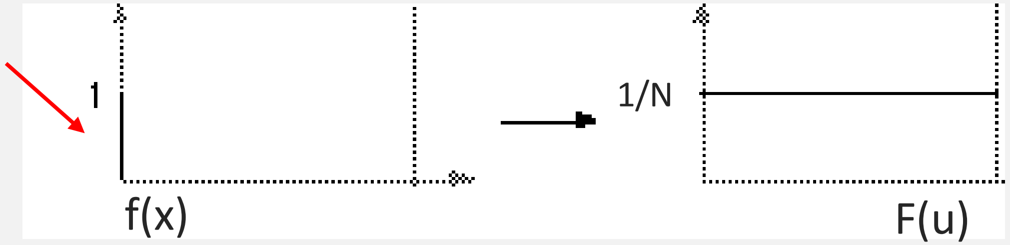
- Scaling relationship: $f\left(\frac{x}{a}\right) \rightarrow aF(au); \quad f(ax) \rightarrow \frac{1}{a} F\left(\frac{u}{a}\right)$

- Time Shift / Frequency Modulation:

$$f(x+a) \rightarrow F(u)e^{i2\pi au/M}; \quad f(x)e^{i2\pi ax/M} \rightarrow F(u-a)$$

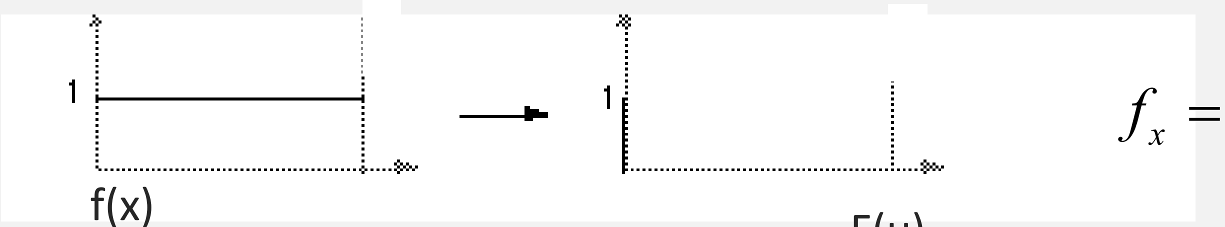
- The transform of a delta function at the origin is a constant

Unit impulse

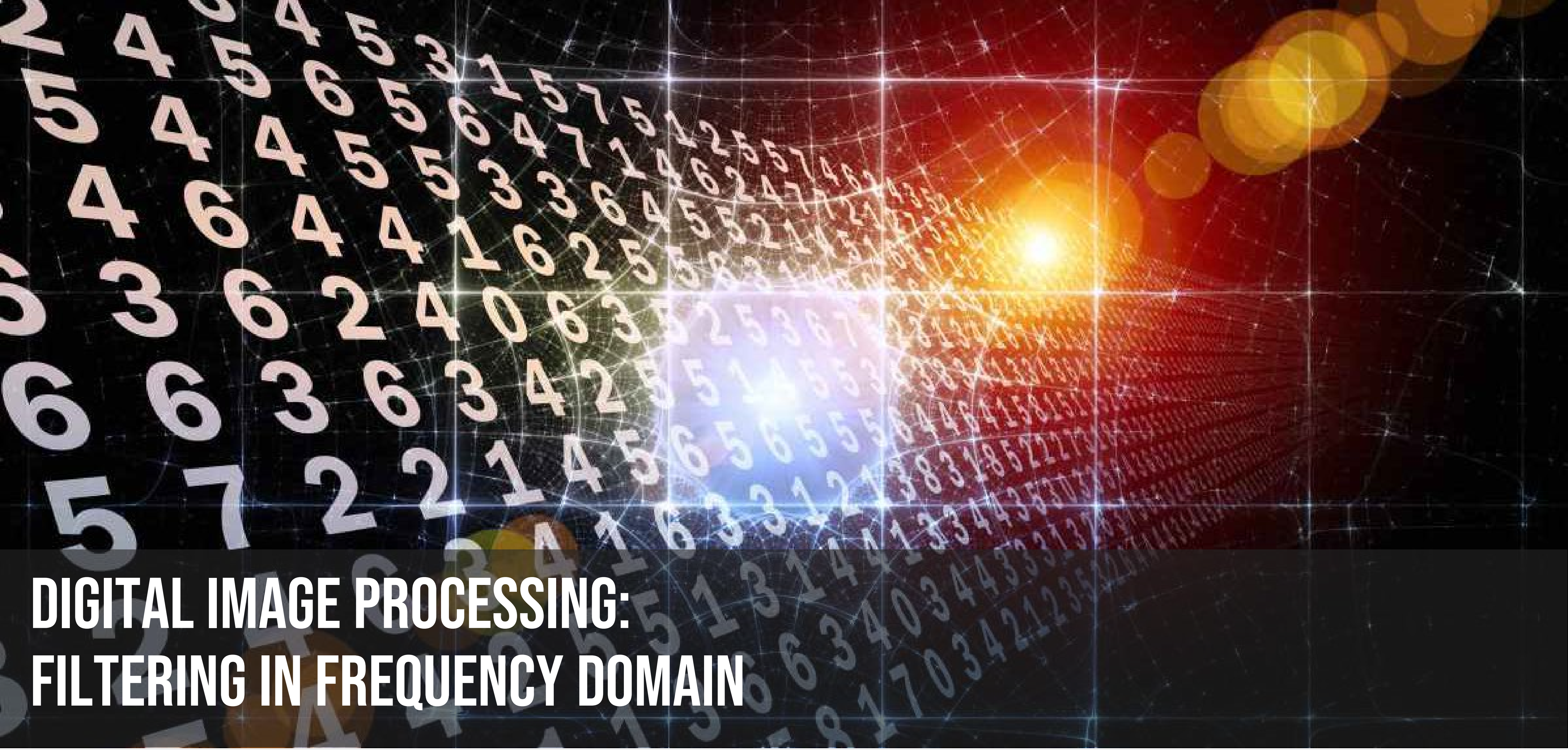


$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} \delta(x) e^{-i2\pi(ux/N)} = \frac{1}{N}$$

- The transform of a constant function is a DC value only.



$$f_x = \sum_{u=0}^{N-1} \exp\left[2\pi i \frac{xu}{N}\right] F_u$$



DIGITAL IMAGE PROCESSING: FILTERING IN FREQUENCY DOMAIN

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

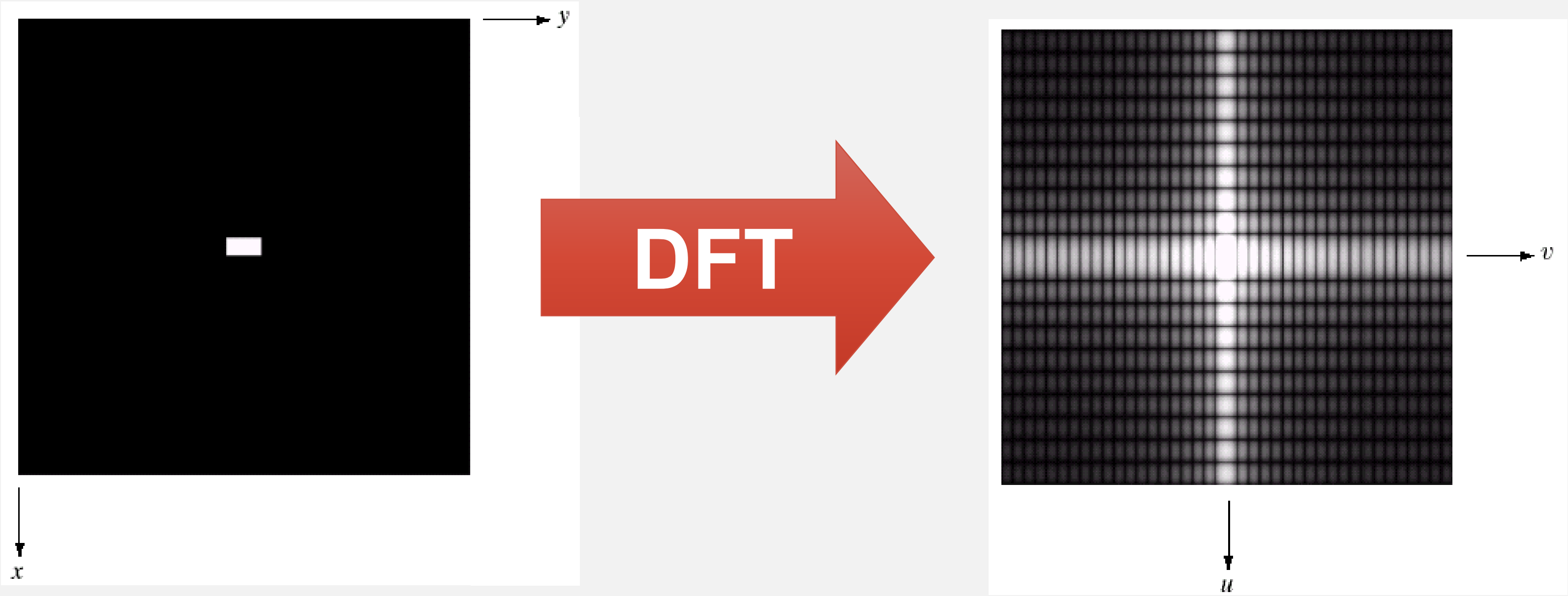
FILTERING IN THE FREQUENCY DOMAIN

- Filter: A device or material for suppressing or minimizing waves or oscillations of certain frequencies.

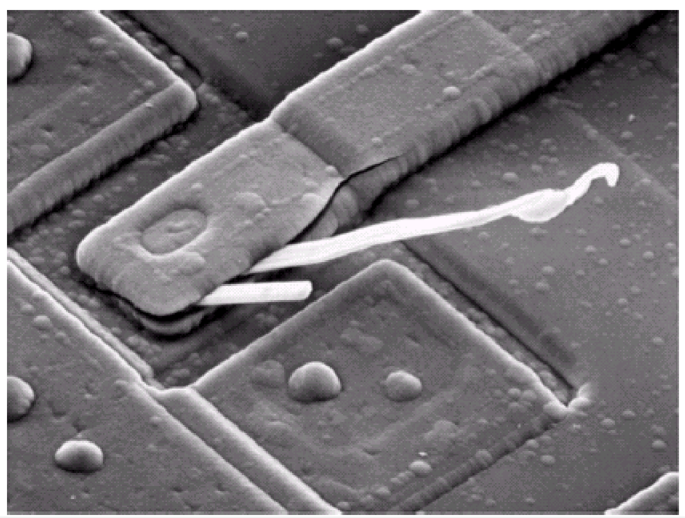
- Frequency: The number of times that a periodic function repeats the same sequence of values during a unit variation of the independent variable.

DFT & IMAGES

- The DFT of a two-dimensional image can be visualised by showing the spectrum of the image component frequencies

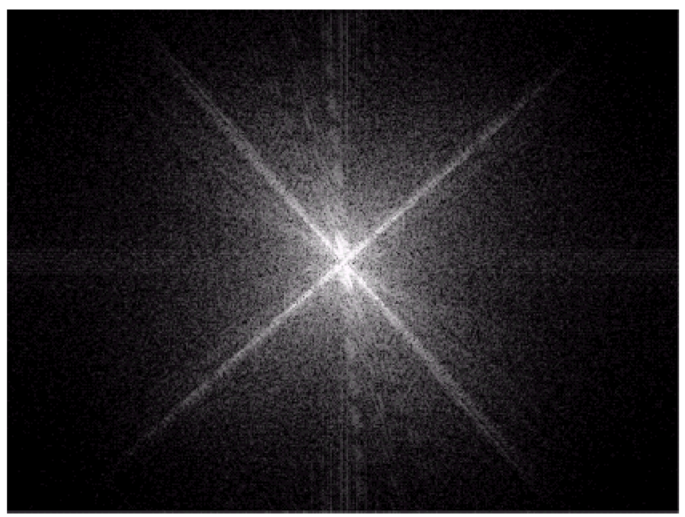


DFT & IMAGES



Scanning electron microscope image of an integrated circuit magnified ~2500 times

DFT



Fourier spectrum of the image

2-D DISCRETE

- For an MxN matrix $$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\exp\left[-2\pi i\left(\frac{xu}{M}+\frac{yv}{N}\right)\right]$$

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\exp\left[2\pi i\left(\frac{xu}{M}+\frac{yv}{N}\right)\right]$$

- All 1-D properties transfer into 2-D
- Some more properties useful for image processing.

SEPARABILITY

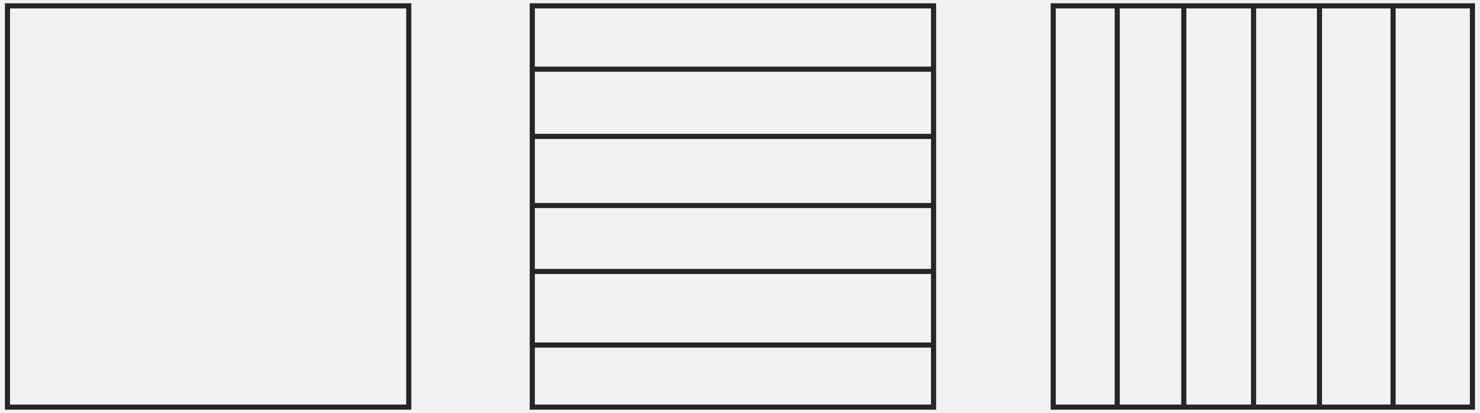
$$\exp\left[2\pi i\left(\frac{xu}{M}+\frac{yv}{N}\right)\right]=\underbrace{\exp\left[2\pi i\frac{xu}{M}\right]}_{\text{1st term}}\underbrace{\exp\left[2\pi i\frac{yv}{N}\right]}_{\text{2nd term}}$$

- The 1st term depends only on x and u, and the 2nd term depends on y and v.
- Advantage: computing DFT of all the columns first, then computing the DFT of all the rows of the result.

(CON'D)

- 1D Fourier pair:

$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1}f(x)\exp\left[-2\pi i\frac{xu}{M}\right];\quad f(x)=\sum_{u=0}^{M-1}F(u)\exp\left[2\pi i\frac{xu}{M}\right]$$



(a) Original image

(b) DFT of each row of (a), using x & u

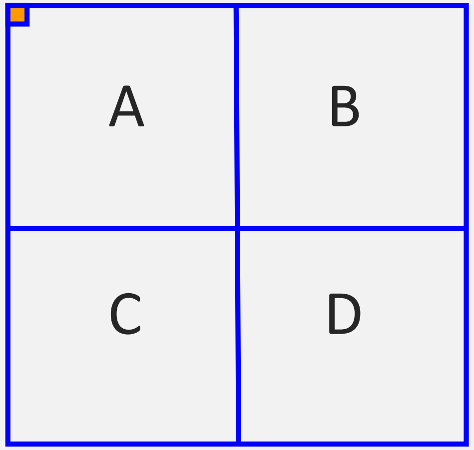
(c) DFT of each column of (b), using y & v

Convolution Theory

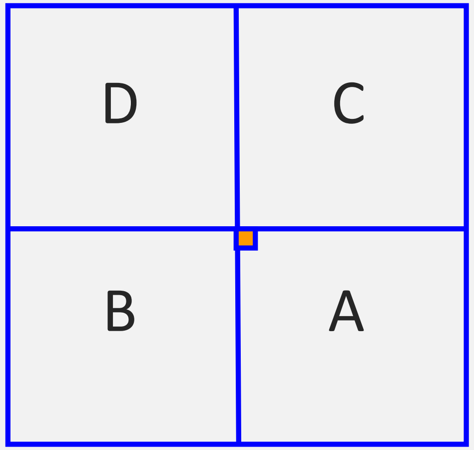
- Review:
  - How to convolve an image M with a spatial filter S in spatial domain?
  - If Z,X,Y are the DFT's of z=x*y, x and y respectively, then Z=X.Y (convolution theory)
- We can perform the same operation (convolution) in frequency domain
  - Pad S with 0, so same size as M; denote the new matrix S'
  - Perform DFT on M and S' to obtain F(M) and F(S')
  - Perform inverse DFT on F(M).F(S') to get F-1
  - M*S is F-1
- Great saving for a large filter.

SHIFTING

- Review: In 1-D, if multiplying each element x_n of vector x by $(-1)^n$ we swap the left and right halves of the Fourier transform.
- In 2-D, the same principle applies if multiplying all elements $x_{m,n}$ by $(-1)^{m+n}$ before the transform.



An FFT

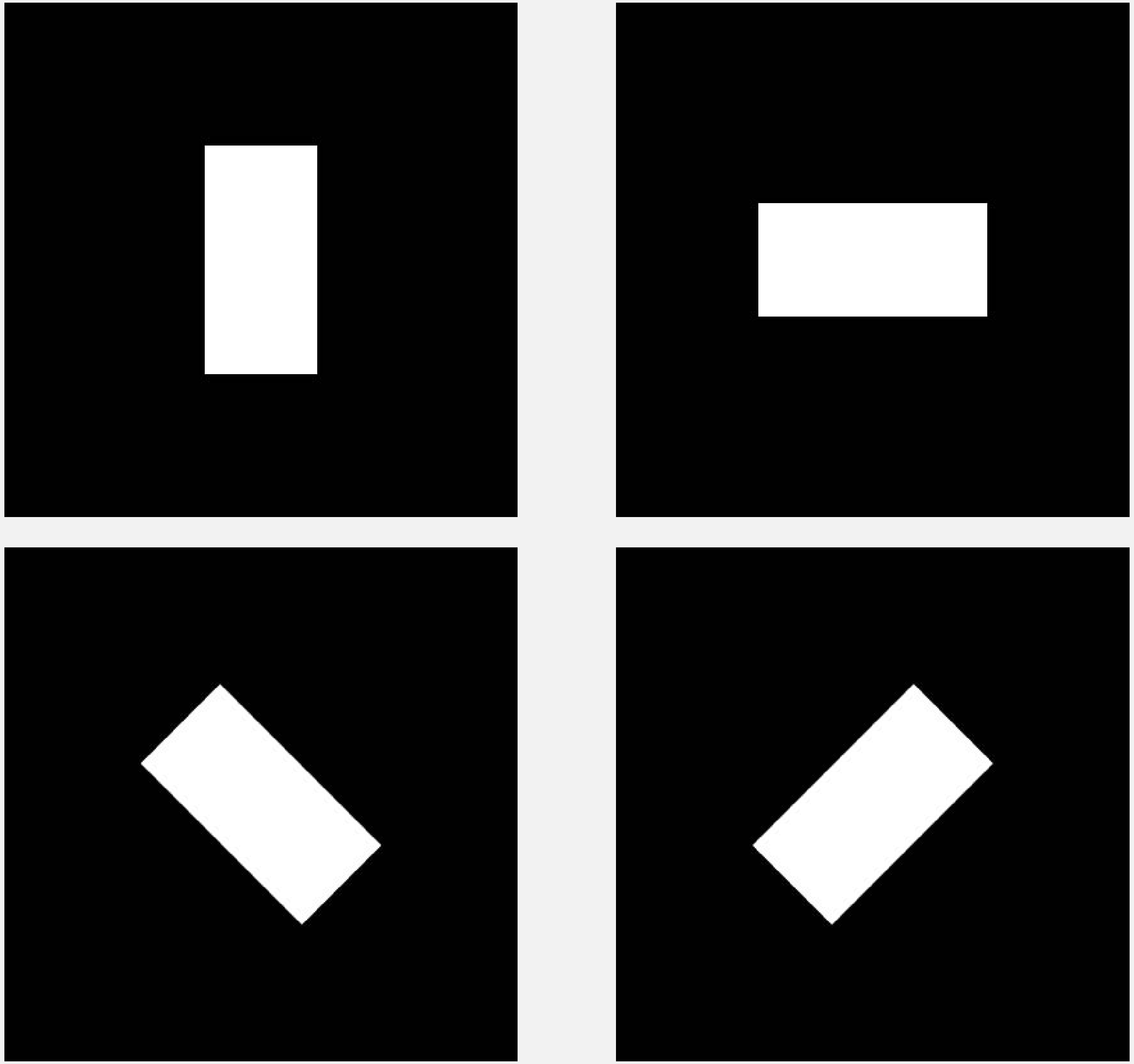


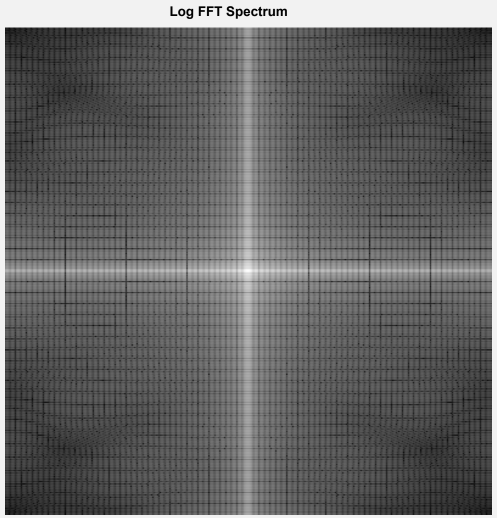
After shifting

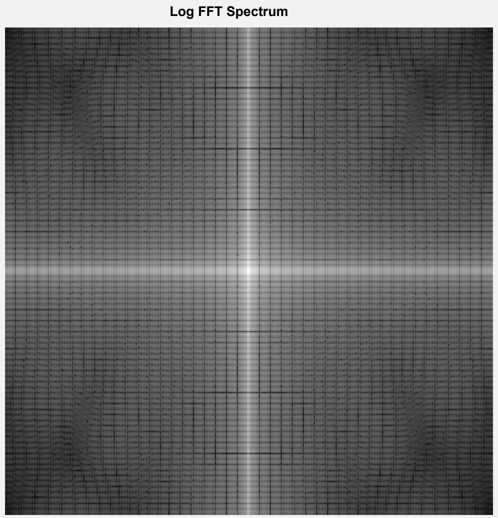
EXAMPLE

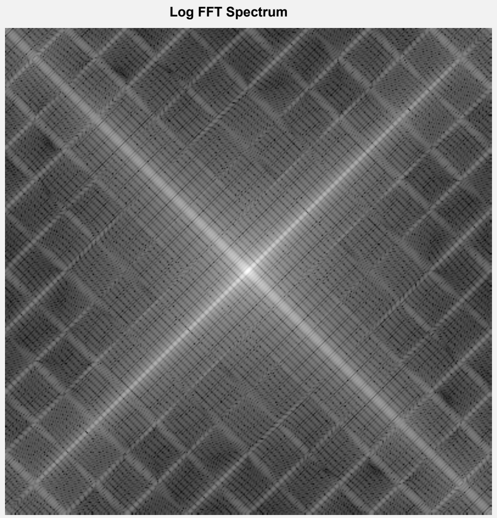
```
clc; clear; close all;
img     = imread('dsp_blackSquare.jpg');          % should be graylevel image
imgGray = rgb2gray(img);
figure; imshow(imgGray); title('Original Image');
% Fourier transform
F       = fft2(imgGray);
% Display the spectrum
S       = abs(F);
figure; imshow(S, []); title('FFT  Spectrum');
% move the origin of the transform to the center of the frequency rectangle
Fc      = fftshift(F);
figure; imshow(abs(Fc), []); title('Centered FFT Spectrum');
% a log transformation
S2 = log (1+abs(Fc));
figure; imshow(abs(S2), []); title('Log FFT Spectrum');
phi = atan2(imag(F), real(F));
figure; imshow(phi, []); title('Angle FFT Spectrum');
```

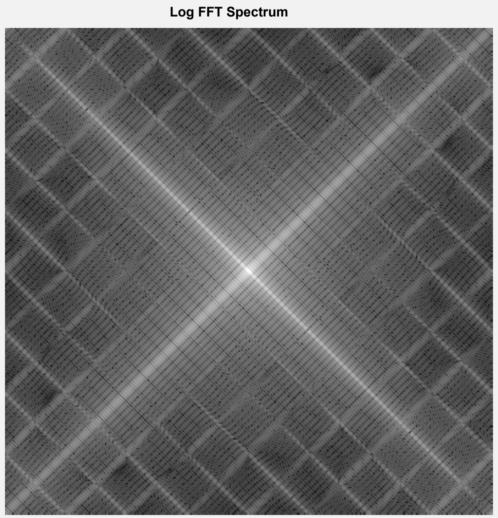
ANGLE VARIATION




Log FFT Spectrum

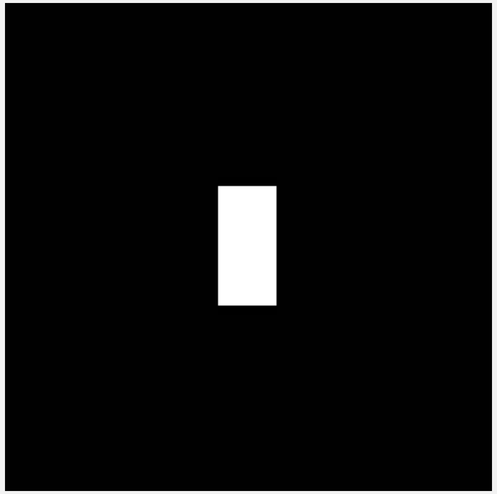

Log FFT Spectrum


Log FFT Spectrum

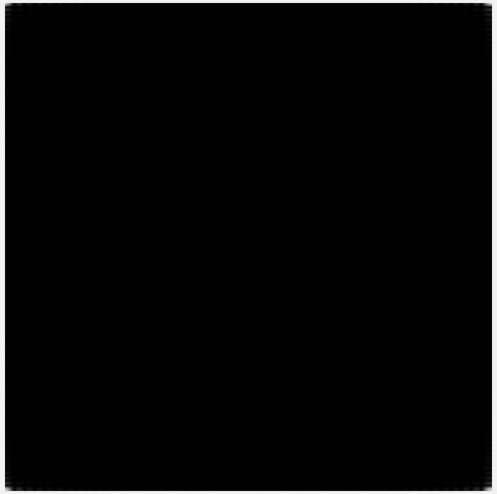

Log FFT Spectrum

EXAMPLE

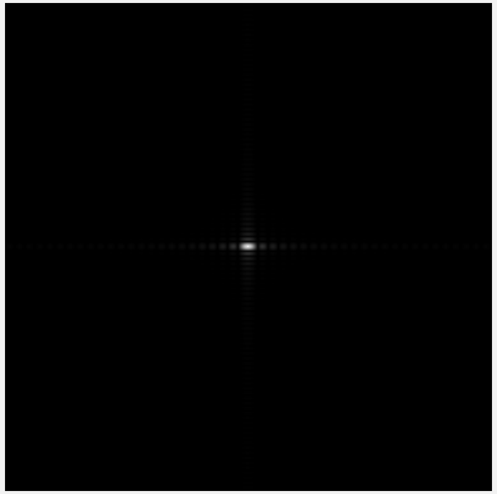
Original Image



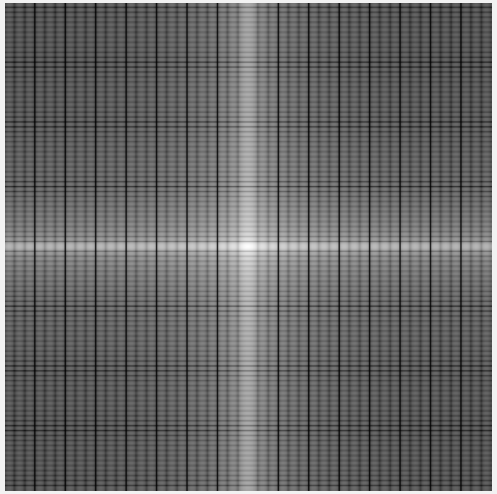
FFT Spectrum



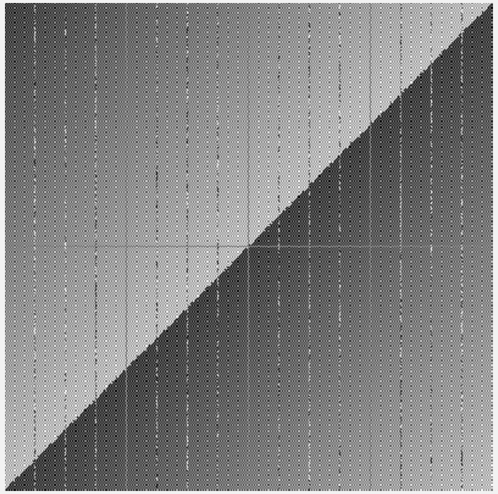
Centered FFT Spectrum



Log FFT Spectrum



Angle FFT Spectrum



FILTERING

- Our basic model for filtering in the frequency domain is

$$\underbrace{G(u,v)}_{\text{result}} = \underbrace{H(u,v)}_{\text{filter}} \underbrace{F(u,v)}_{\substack{\text{Fourier transform of} \\ \text{image to be smoothed}}}$$

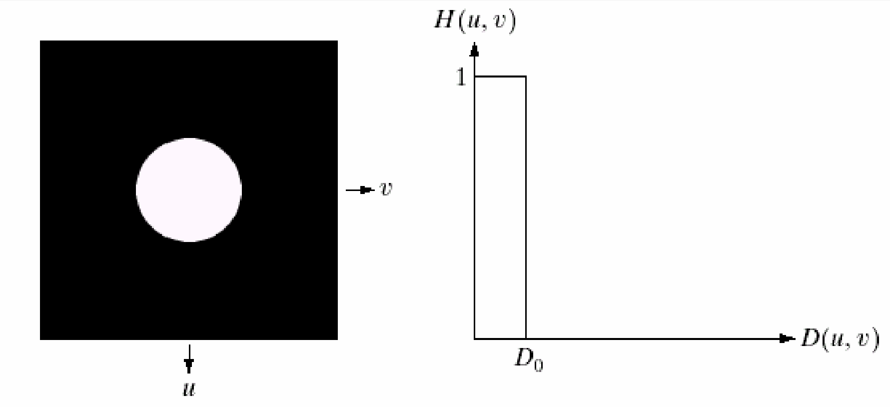
- We'll briefly discuss 3 types of filters in the order of increasing smoothness:
  - Ideal
  - Butterworth
  - Gaussian
- Preprocessing: F is shifted so that the DC coefficient is in the center.

IDEAL FILTERING:LOW-PASS (ILPF)

- The low-frequency components are toward the center.
- Multiplying the transform by a matrix to remove or minimize the values away from the center.
- Ideal low-pass matrix H

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

- The inverse DFT of H.F is the smoothed image.



D(u,v) is distance from the origin of the Fourier transform, shifted to the center

COMPUTING THE 2-D DFT IN MATLAB

- The FFT of an image array f is obtained in MATLAB using function fft2

```
F = fft2(f)
```

  - returns a Fourier transform of size M X N
- It is necessary to pad the input image with zeros when the Fourier transform is used for filtering

```
F = fft2(f,P,Q)
```

- The Fourier spectrum is obtained by using function abs

```
S = abs(F)
```

BASIC STEPS IN DFT FILTERING

- Convert the input image to floating point using function tofloat

```
[f,revertclass] = tofloat(f);
```

- Obtain the padding parameters using function paddedsize

```
PQ = paddedzsize(size(f));
```

- Obtain the Fourier transform with padding

```
F = fft2(f,PQ(1),PQ(2));
```

- Generate a filter function, H, of size PQ (1) x PQ (2)

- Multiply the transform by the filter

```
G = H.*F;
```

BASIC STEPS IN DFT FILTERING (CONT)

- Obtain the inverse FFT of G

```
g = ifft2(G);
```

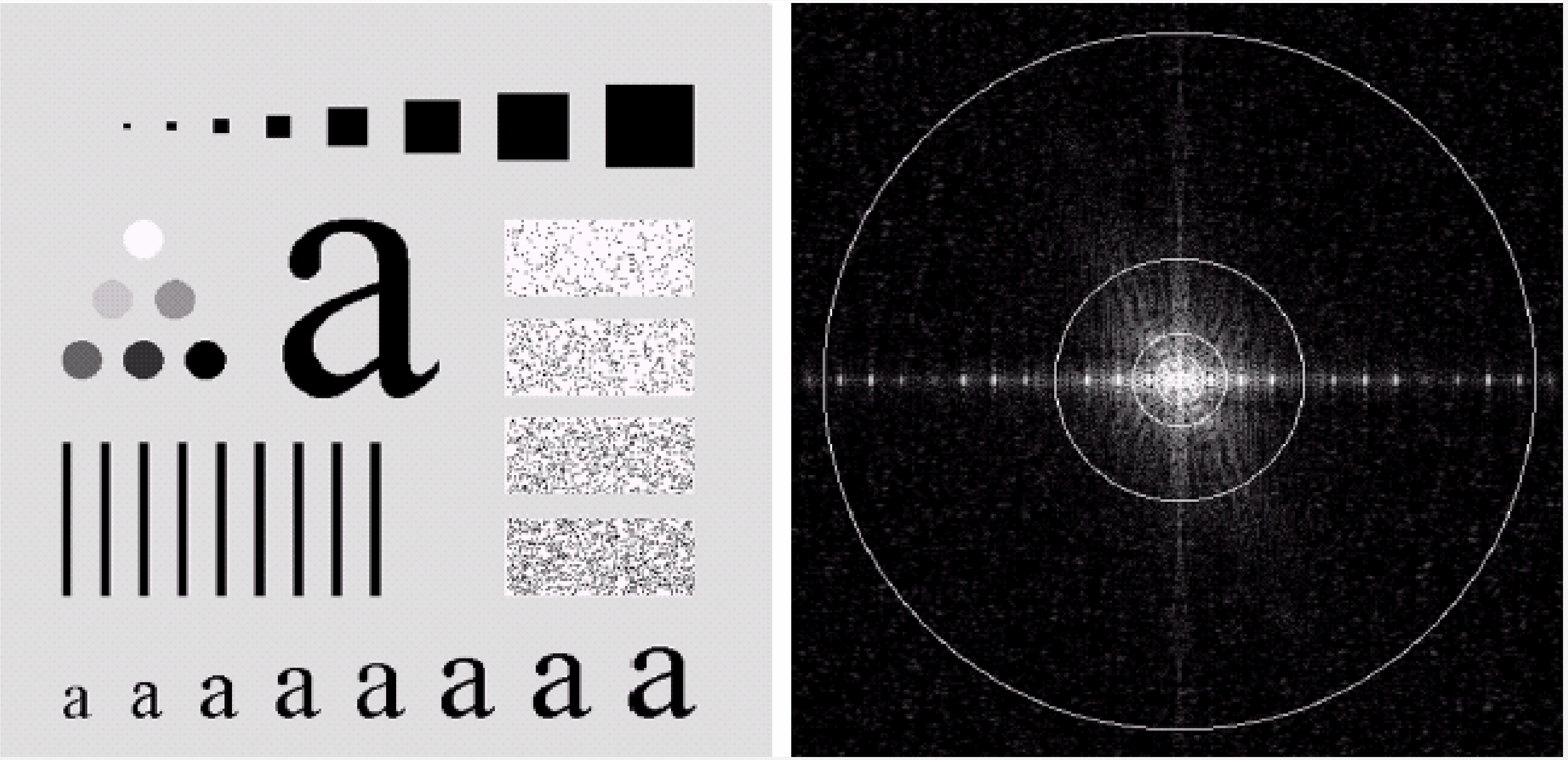
- Crop the top, left rectangle to the original size

```
g = g(1:size(f,1),1:size(f,2));
```

- Convert the filtered image to the class of the input image, if so desired

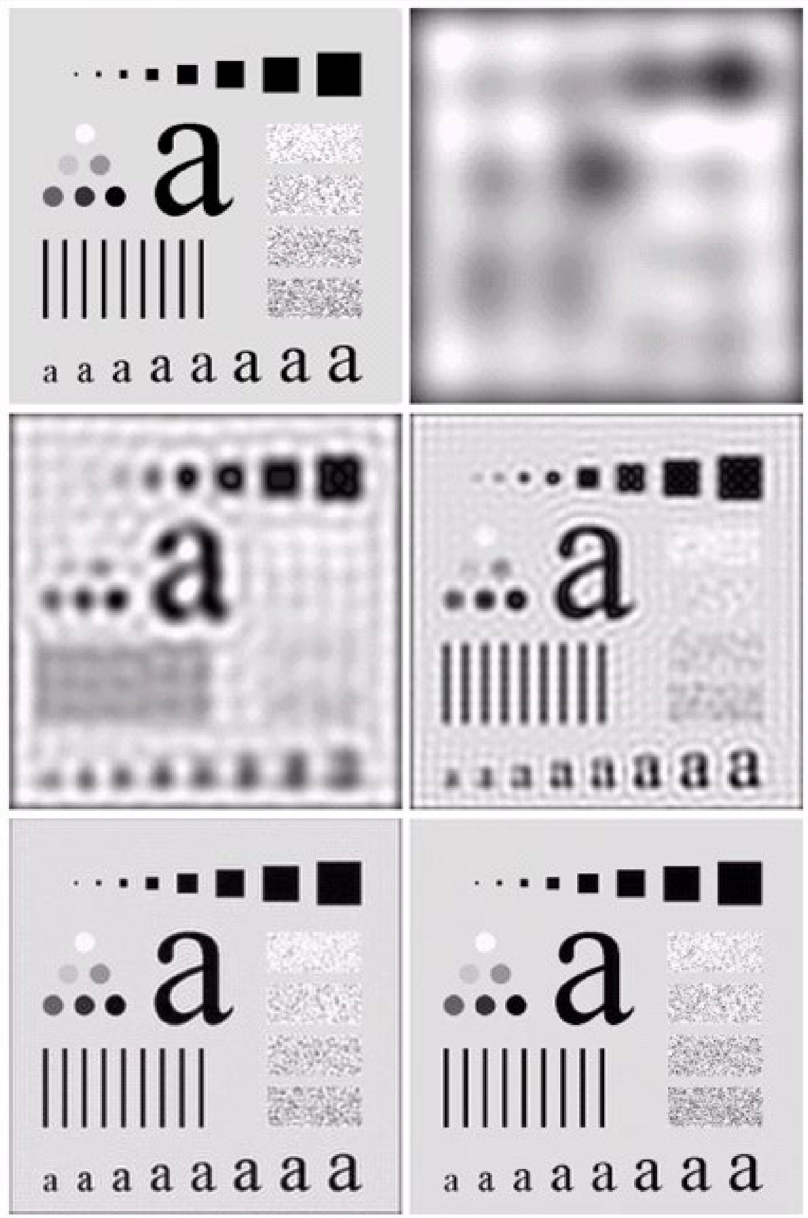
```
g = revertclass(g);
```

DEMO



An image with its Fourier spectrum. The superimposed circles have radii of 5, 15, 30, 80, and 230

RESULTS



D=5

D=15

D=30

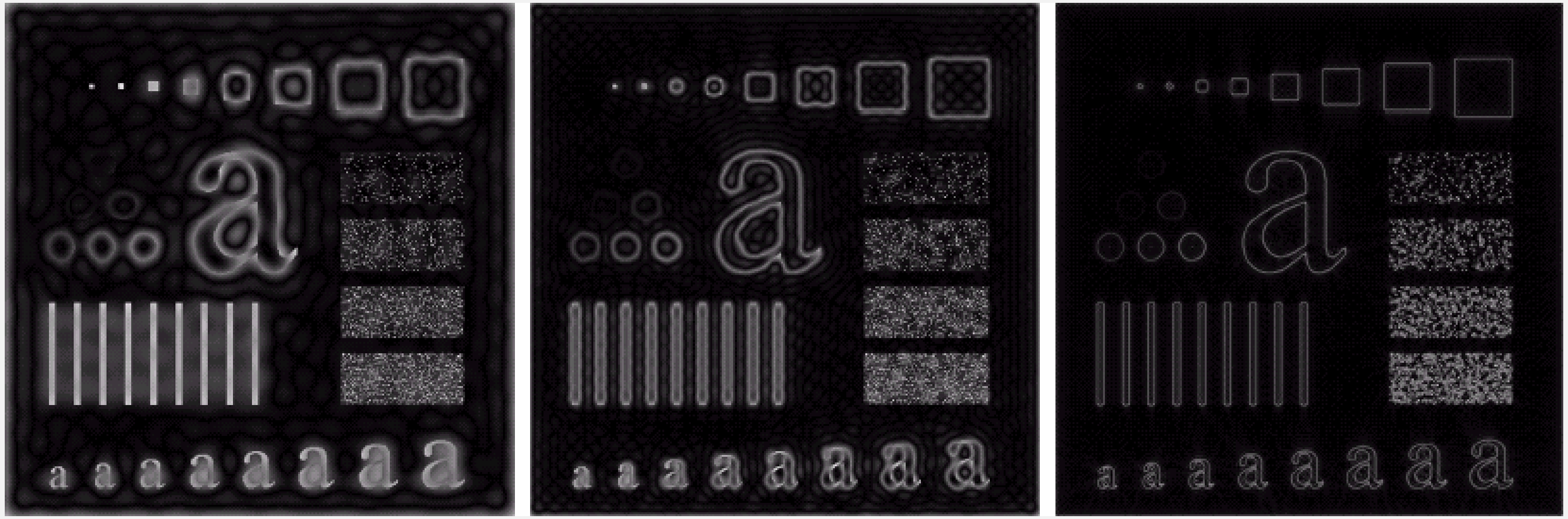
D=80

D=230

IDEAL FILTERING:HIGH-PASS (IHPF)

- Opposite to low-pass filtering: eliminating center and keeping the others.

$$H(u,v) = \begin{cases} 0 & \text{if } \mathrm{D}(u,v) \leq \mathrm{D}_0 \\ 1 & \text{otherwise} \end{cases}$$



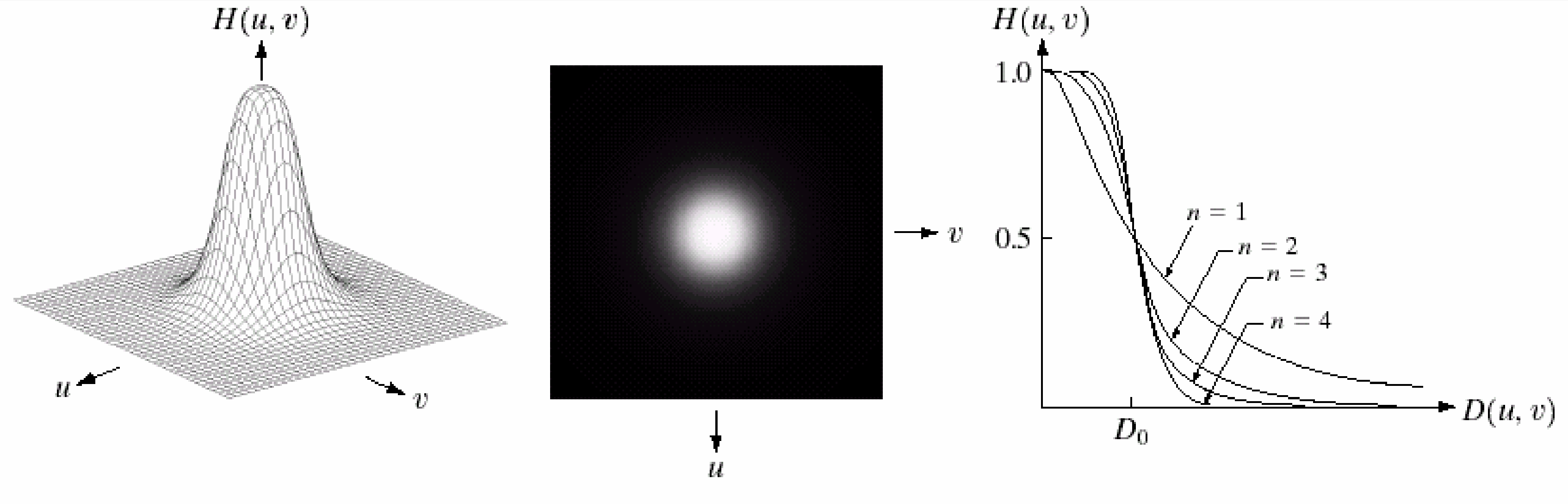
D_0=15 D_0=30 D_0=80

BUTTERWORTH FILTERING: LOW-PASS (BLPF)

- Unlike ILPF, BLPF does not have a clear cutoff between passed and filtered frequencies.

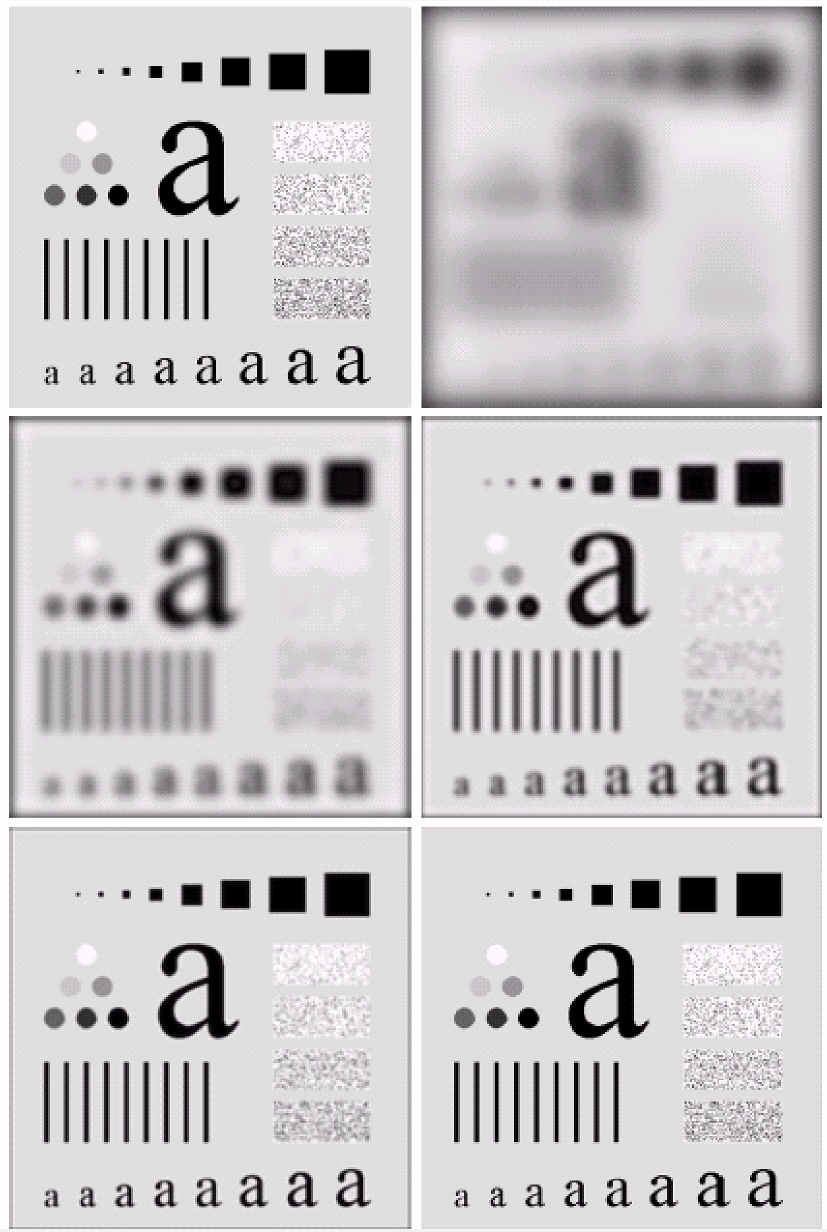
$$H(u,v) = \frac{1}{1+\left[D(u,v)/D_0\right]^{2n}}$$

n is the order of the filter



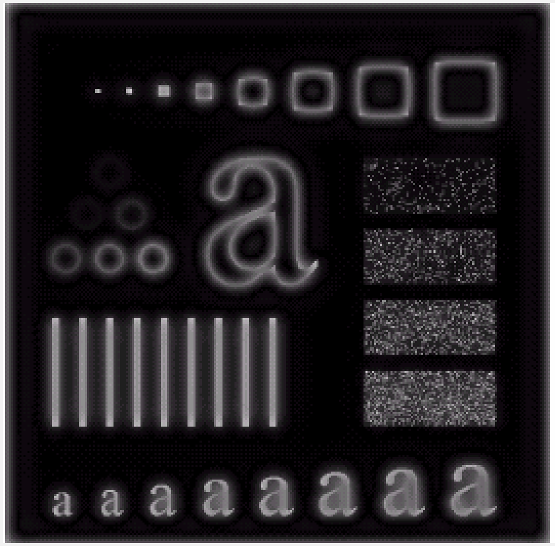
DEMO

- n=2 and D_0 equal the 5 radii

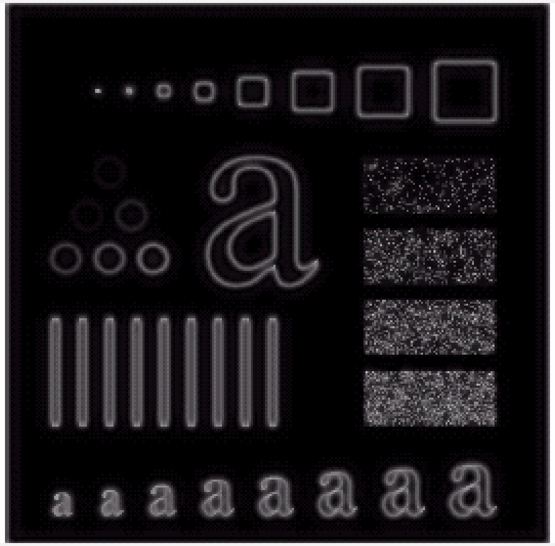


BUTTERWORTH FILTERING: HIGH-PASS (BHPF)

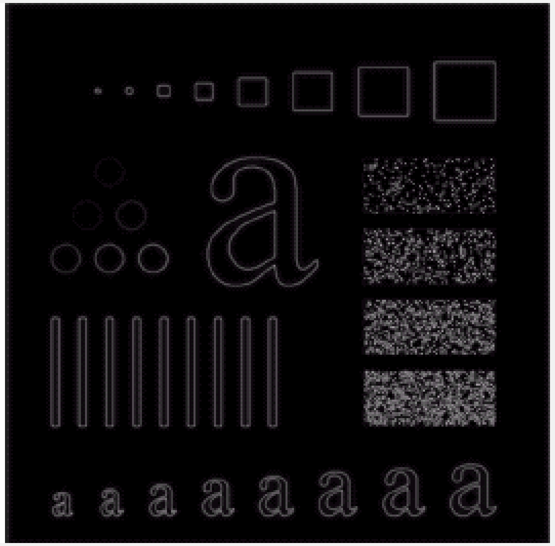
$$H(u,v) = \frac{1}{1+\left[D_0/D(u,v)\right]^{2n}}$$



D_0=15



D_0=30



D_0=80

GAUSSIAN FILTERING: LOW-PASS (GLPF)

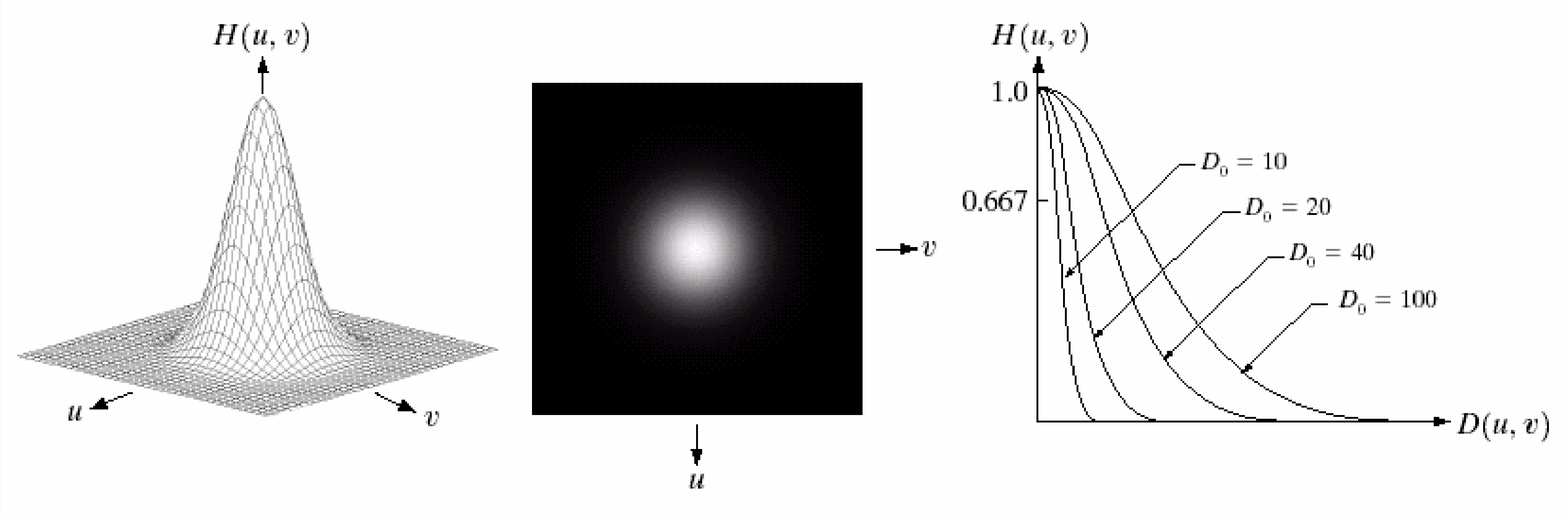
- We mentioned Gaussian once in the section for spatial filtering.
- Will discuss more in detail in image restoration.
- Gaussian filter in frequency domain:

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}$$

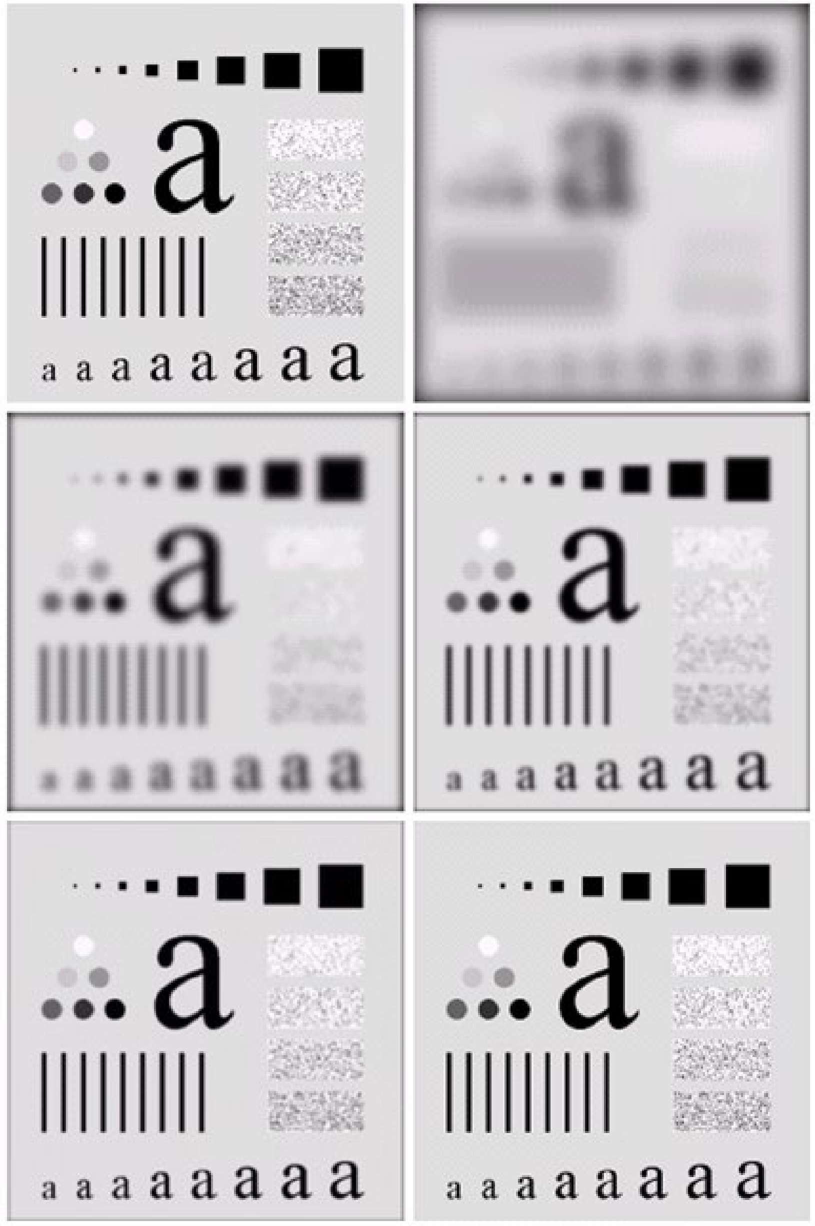
- The inverse is also a Gaussian.
- We may replace σ by D_0, which is the cutoff frequency.

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

GLPF (CON'D)



DEMO



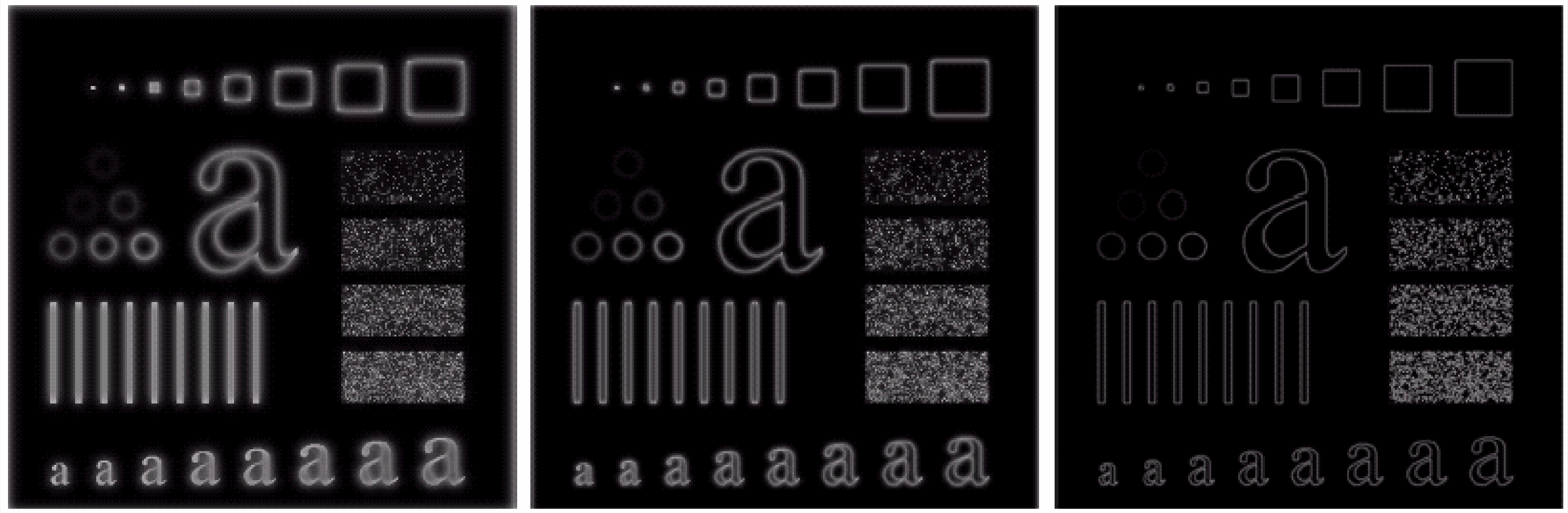
$D_0=5$

$D_0=15$ $D_0=30$

$D_0=80$ $D_0=230$

GAUSSIAN FILTERING: HIGH-PASS (GHPF)

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$



D_0=15 D_0=30 D_0=80

IN-CLASS EXERCISE

- Q. What is the Fourier transform of the average filtering using the 4 immediate neighbors of point (x,y), but excluding itself?

$$g(x,y)=\frac{1}{4}\left[f(x,y+1)+f(x+1,y)+f(x-1,y)+f(x,y-1)\right]$$

SUMMARY

- To perform filtering in frequency domain, do the following steps:
  - The Fourier transform of the image is shifted, so that the DC coefficient is in the center. (multiply the image by (-1)x+y)
  - Create the filter
  - Multiply it by the image transform
  - Invert the result
  - Multiplying the result by (-1)x+y.
- The relationship between the corresponding high- and low-pass filters:

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

TERIMA KASIH

Adhi Harmoko Saputro

UNIVERSITAS INDONESIA

Veritas, Probitas, Iustitia