

# Digital Image Processing

Lecture #9  
Ming-Sui (Amy) Lee



# Announcement



## ■ The following schedule

02/17	Lecture 1	04/14	Lecture 8
02/24	Lecture 2	04/21	Proposal
03/03	Lecture 3	04/28	Lecture 9
03/10	Lecture 4	05/05	Lecture 10
03/17	Lecture 5	05/12	Lecture 11
03/24	Lecture 6	05/19	Demo
03/31	Lecture 7	05/26	Demo
04/07	Midterm	06/02	Final Package Due

# **Image Enhancement in Frequency Domain**

# Background

- Jean Baptiste Joseph Fourier (1807)
  - Any periodic function
    - Can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient → Fourier series
  - Functions that are not periodic
    - Can be expressed as the integral of sines and/or cosines multiplied by a weighting function → Fourier transform
  - Important characteristic
    - A function can be reconstructed completely without losing any information
    - Fourier/frequency domain processing

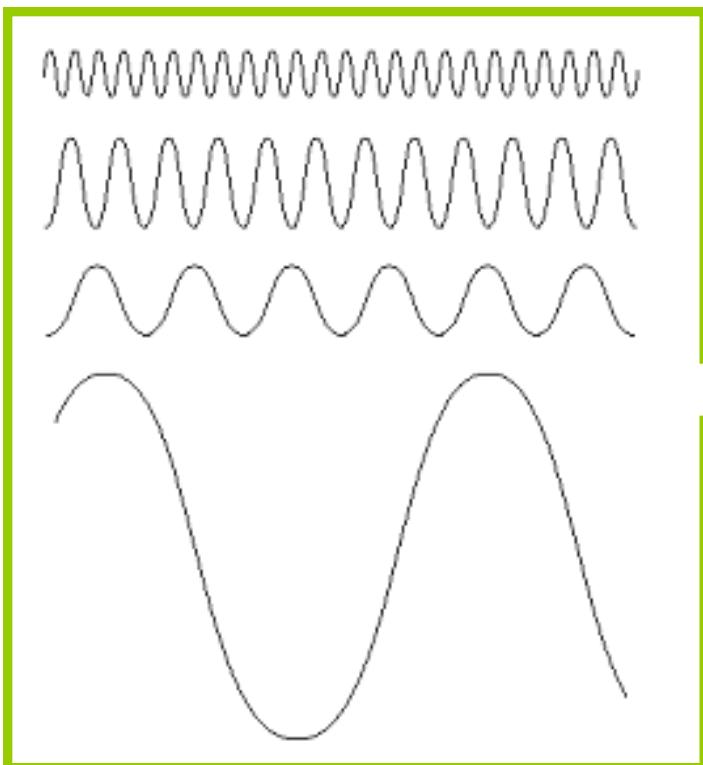


[

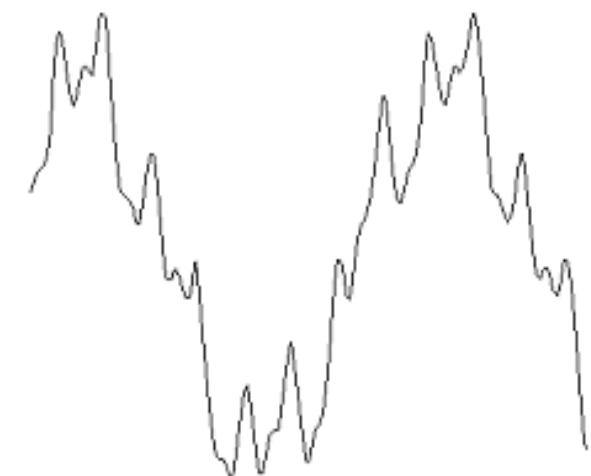
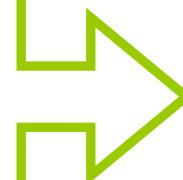
# Fourier Transform

]

## ■ Example



Basis Functions



combination

# Fourier Transform

## Continuous cases

### (Continuous) Fourier Transform (CFT)

#### One-dimensional

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

Euler's formula:

$$e^{j\theta} = \cos \theta + j \sin \theta$$

#### Two-dimensional

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy$$

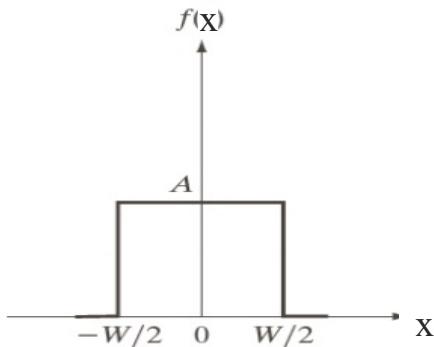
$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv$$

# Fourier Transform

## Continuous Cases

- CFT of a square wave between  $[-\frac{W}{2}, \frac{W}{2}]$  with amplitude A:

$$\begin{aligned} F(u) &= \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx = \int_{-W/2}^{W/2} A e^{-j2\pi ux} dx = \frac{-A}{j2\pi u} [e^{-j2\pi ux}]_{-W/2}^{W/2} \\ &= \frac{-A}{j2\pi u} [e^{-j\pi uW} - e^{j\pi uW}] = \frac{A}{j2\pi u} [e^{j\pi uW} - e^{-j\pi uW}] = AW \frac{\sin(\pi uW)}{\pi uW} \end{aligned}$$



# Fourier Transform

## ■ Discrete cases

### ○ Discrete Fourier Transform (DFT)

#### ■ One-dimensional

Let  $F(u) = \sum_{k=0}^{\Delta} f(k) e^{-j2\pi u k / M}$        $\Delta u = \frac{1}{M \Delta j}$

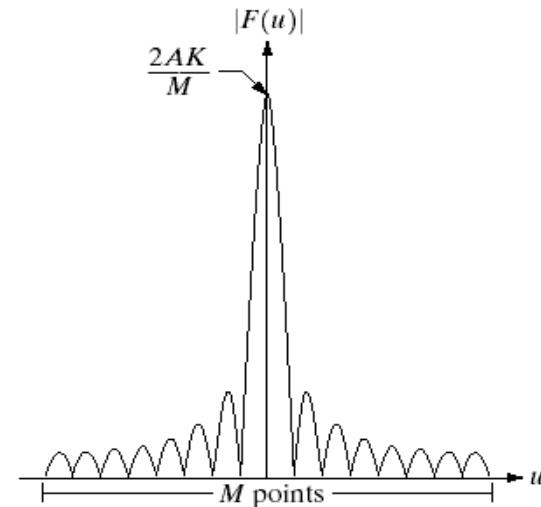
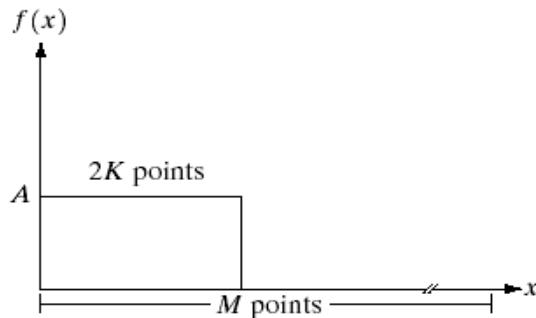
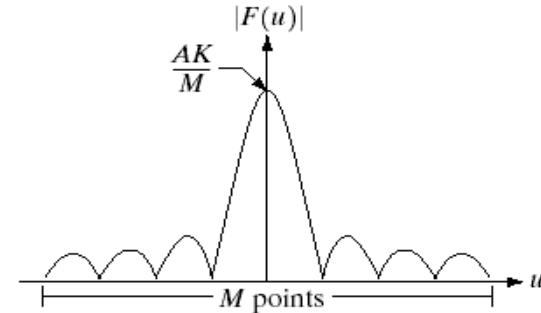
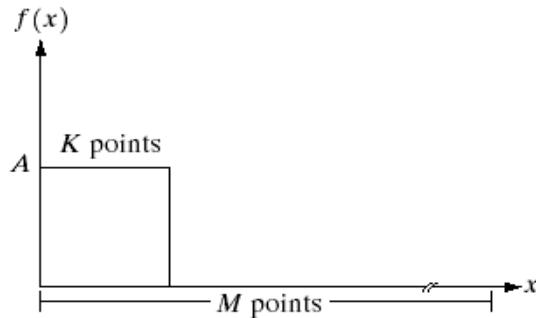
$$f(k) = f(j_0 + k \Delta j)$$

→ 
$$\begin{cases} F(u) = \frac{1}{M} \sum_{k=0}^{M-1} f(k) e^{-j2\pi u k / M}, & u = 0, 1, \dots, M-1 \\ f(k) = \sum_{u=0}^{M-1} F(u) e^{j2\pi u k / M}, & k = 0, 1, \dots, M-1 \end{cases}$$

//Note// DFT and its inverse always exist

# Fourier Transform

## Example



# Fourier Transform

## ■ Discrete cases

### ○ One-dimensional

#### ■ Mathematical prism

- Separate a function into various components

*frequency*

$$F(u) = \frac{1}{M} \sum_{k=0}^{M-1} f(k) e^{-j2\pi uk/M}, \quad u = 0, 1, \dots, M-1$$

*frequency component*

$$= \frac{1}{M} \sum_{k=0}^{M-1} f(k) [\cos(2\pi uk / M) - j \sin(2\pi uk / M)]$$

# Fourier Transform

## ■ Discrete cases

### ○ Two-dimensional

$$F(u, v) = \frac{1}{MN} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} f(j, k) e^{-j2\pi(\frac{uj}{M} + \frac{vk}{N})} \quad \rightarrow \text{Complex}$$

$$f(j, k) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi\left(\frac{uj}{M} + \frac{vk}{N}\right)}$$

**Fourier Spectrum:**  $|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$

**Phase Angle:**  $\phi(u, v) = \tan^{-1} \left[ \frac{I(u, v)}{R(u, v)} \right]$

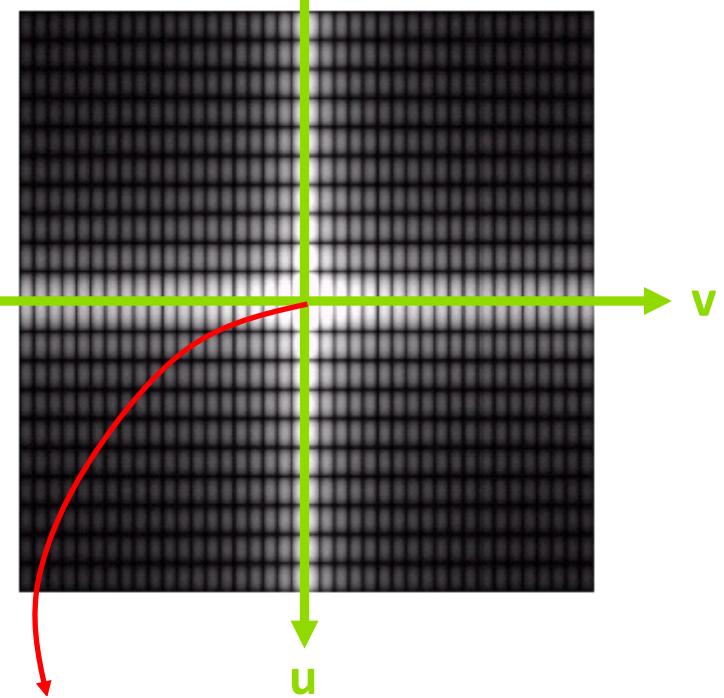
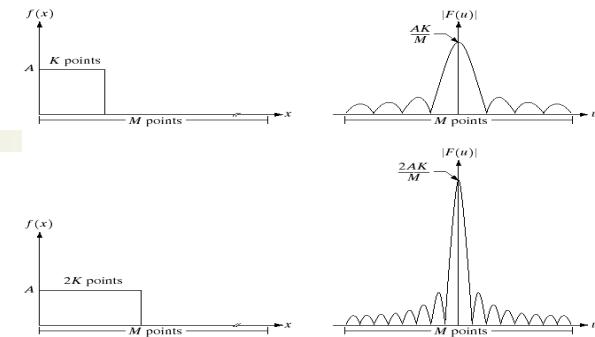
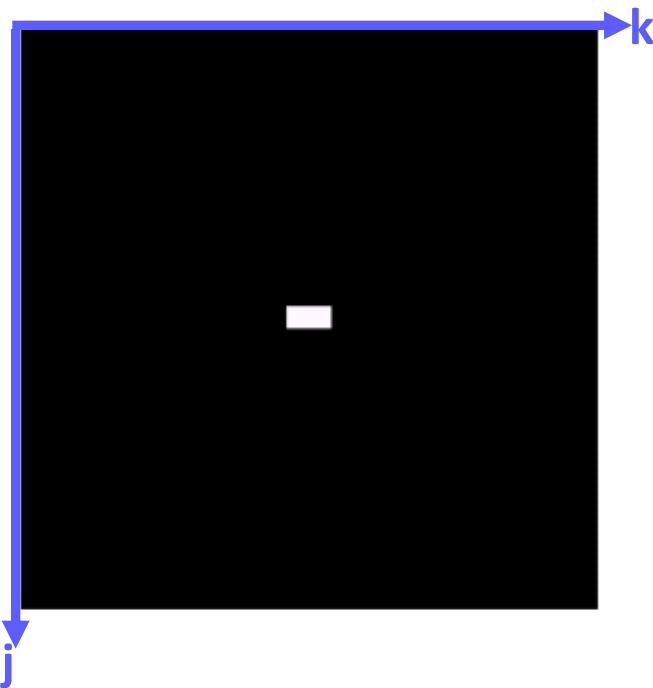
**Power Spectrum:**  $P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$  11

[

# Fourier Transform

]

## Example



$$\mathcal{F}[f(j,k)(-1)^{j+k}] = F(u - M/2, v - N/2)$$

Centering & log transformation

$$F(0, 0) = \frac{1}{MN} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} f(j, k)$$

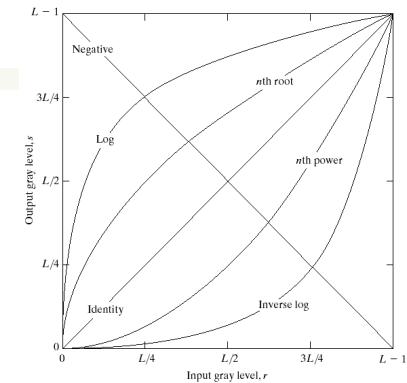
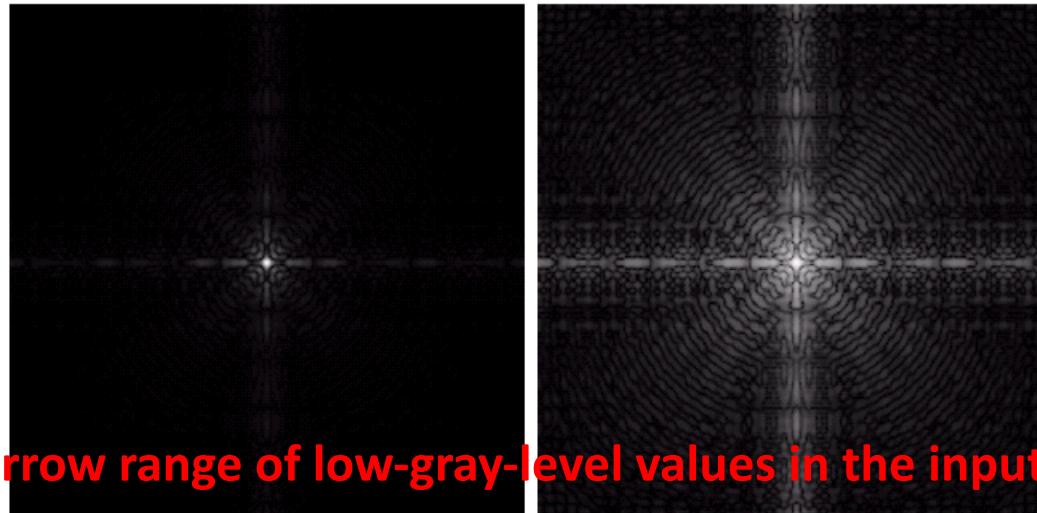
Average/dc component (zero frequency)

# Fourier Transform

## ■ Log Transformation

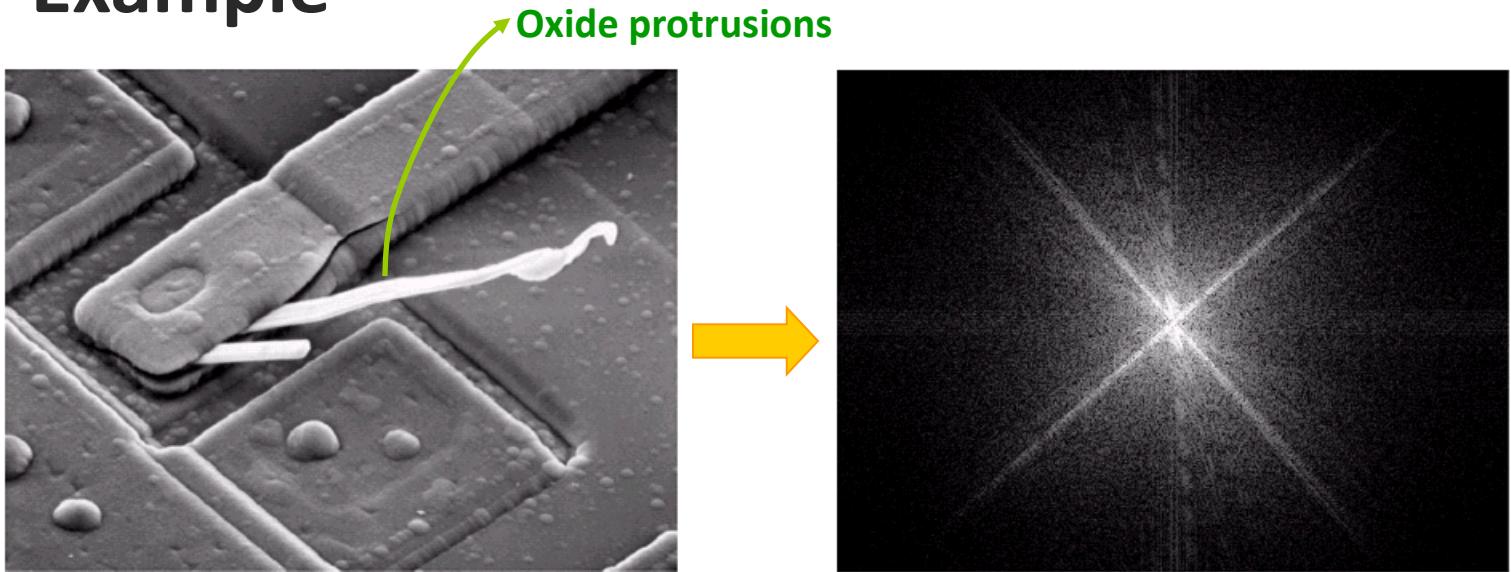
$$s = c \log(1 + r), \quad r \geq 0$$

- Expand the dynamic range of low gray-level values
- Compress the dynamic range of images with large variations in pixel values



# Fourier Transform

## Example



- Strong edges  $\rightarrow \pm 45^\circ$  directions
- Oxide protrusions  $\rightarrow$  vertical component slightly slant to the left
- Zeros in the vertical frequency component  $\rightarrow$  narrow vertical span of the oxide protrusions

# [ Fourier Transform ]

## ■ Properties

### ○ Centering

$$\mathcal{F}[f(j,k)(-1)^{j+k}] = F(u - M/2, v - N/2)$$

### ○ Conjugate symmetry

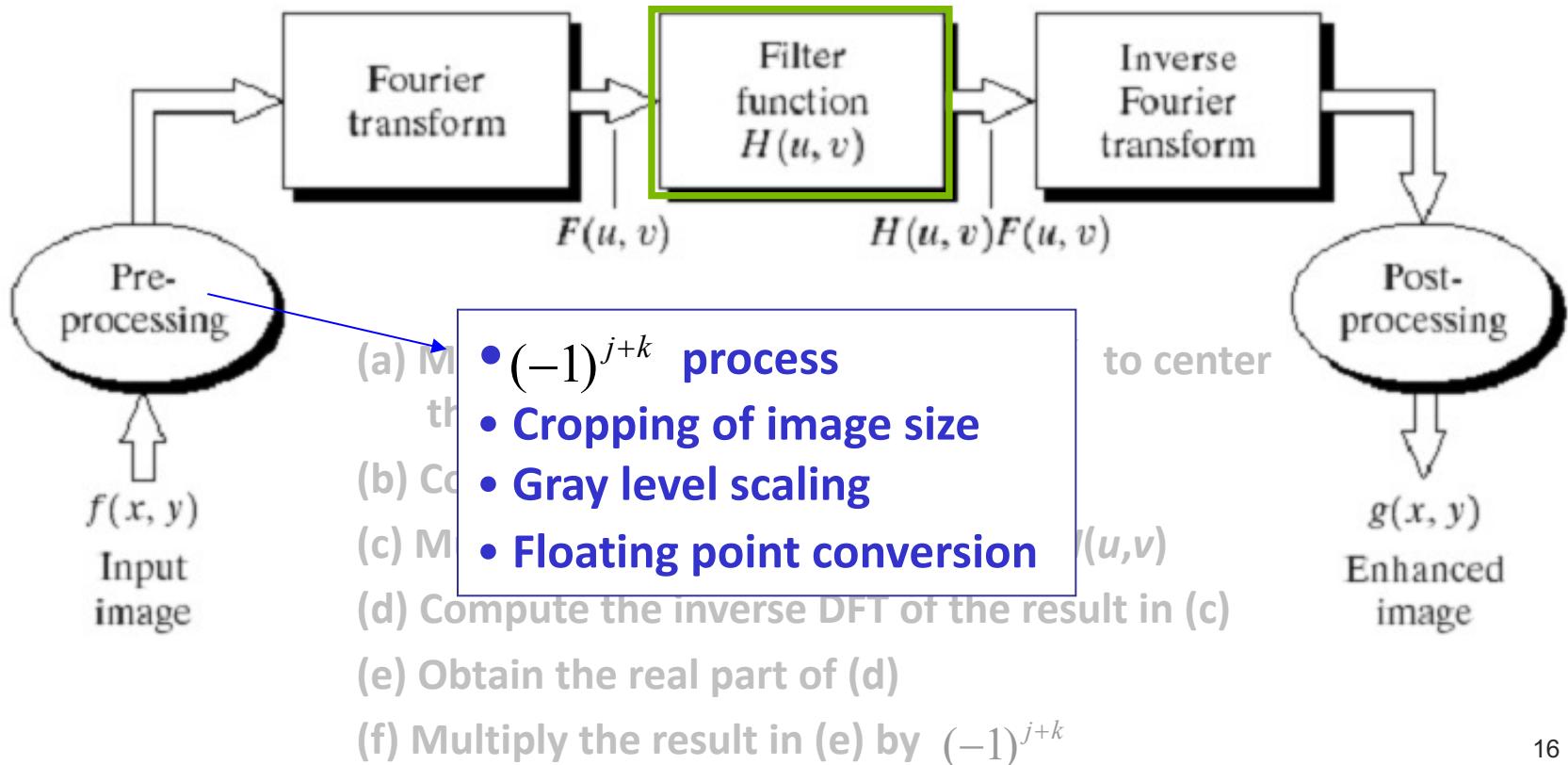
If  $f(j,k)$  is real,

$$F(u, v) = F^*(-u, -v)$$

$$|F(u, v)| = |F(-u, -v)|$$

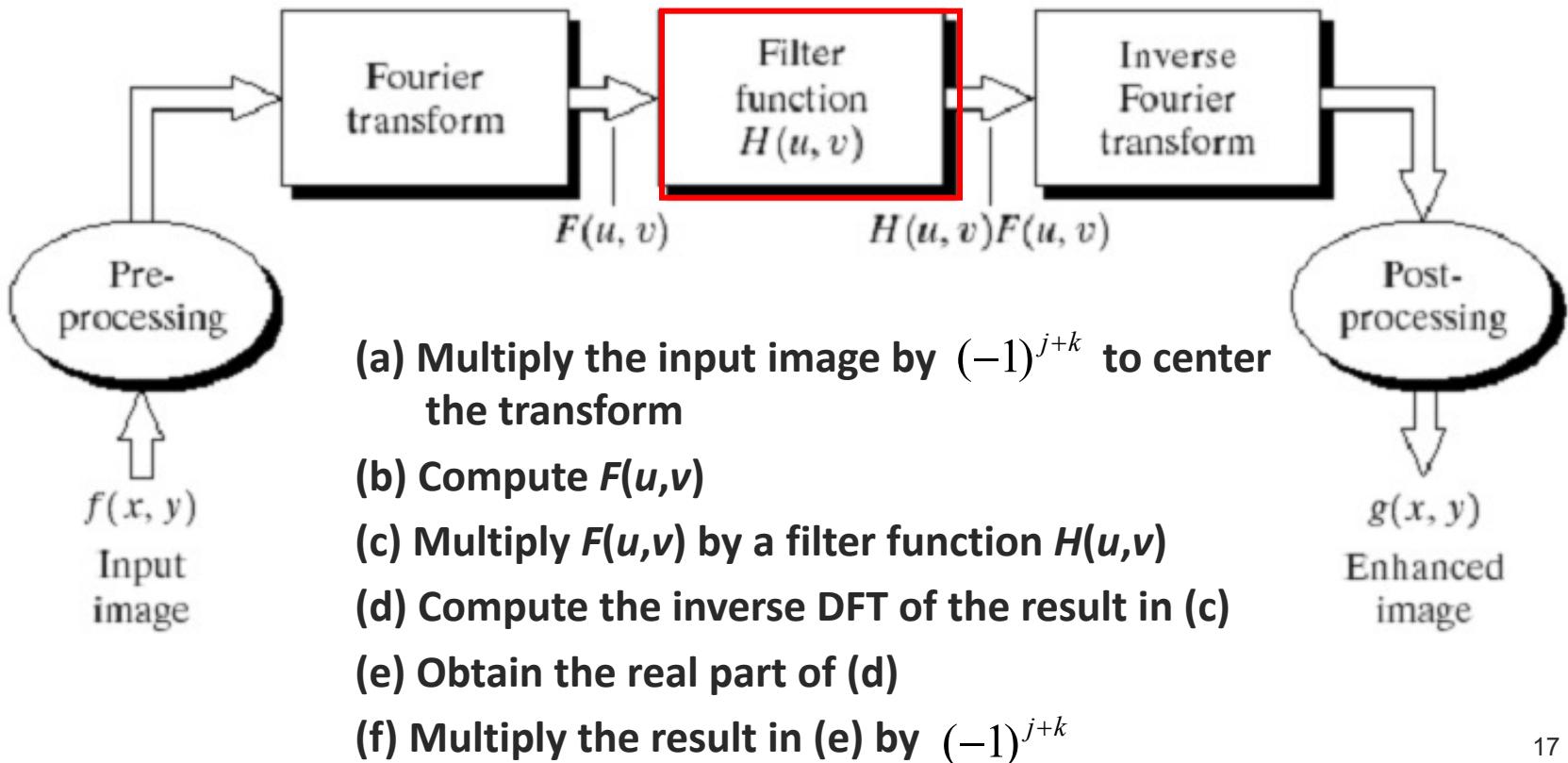
# Frequency-Domain Filtering

## Filtering in the frequency domain



# Frequency-Domain Filtering

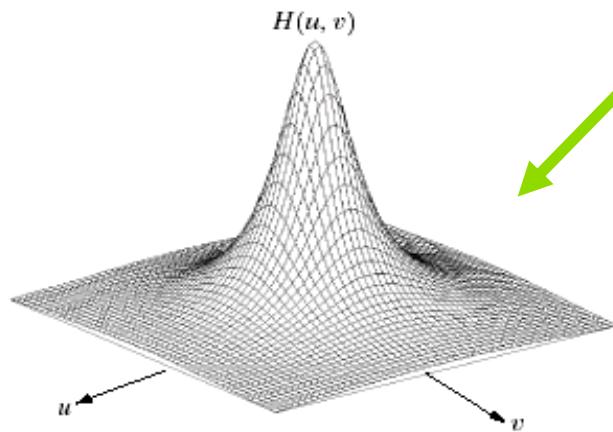
## Filtering in the frequency domain



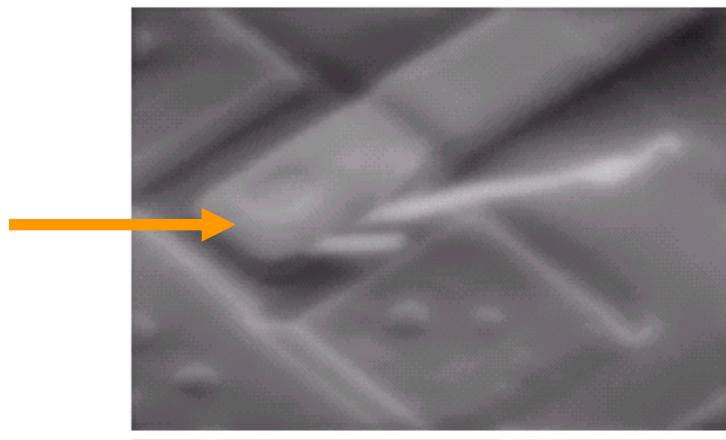
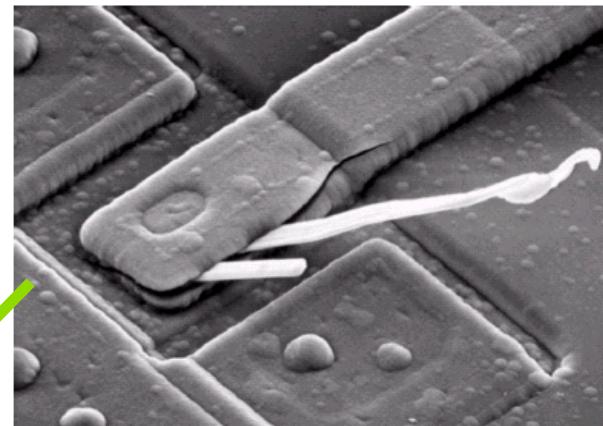
# [ Frequency-Domain Filtering ]

## ■ Basic filters and their properties

- **Low-pass filter**
- **High-pass filter**
- **Notch filter**



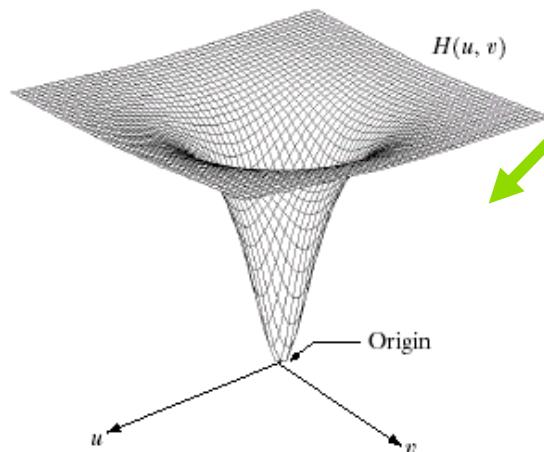
- Attenuates high frequencies
- Passes low frequencies



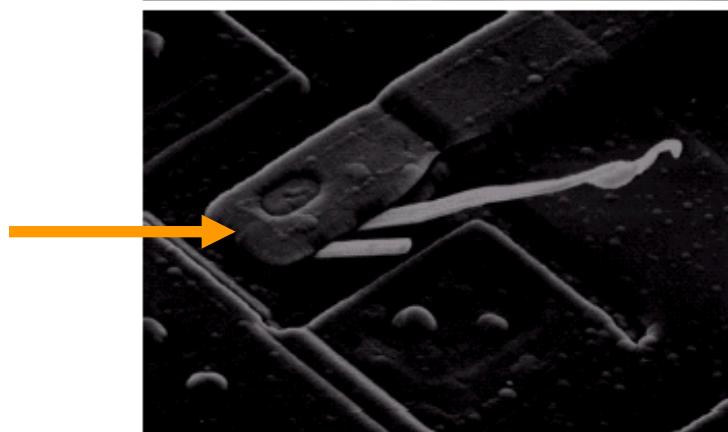
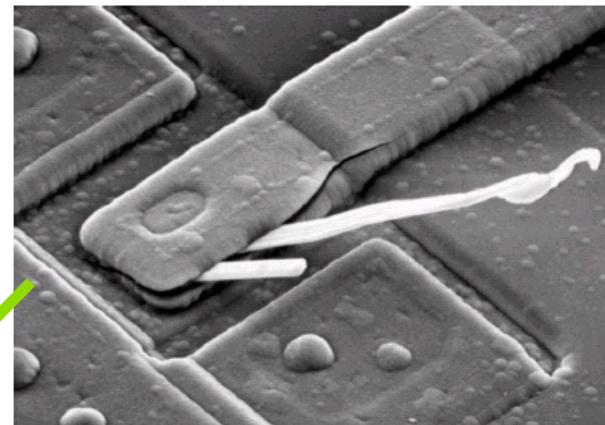
# [ Frequency-Domain Filtering ]

## ■ Basic filters and their properties

- Low-pass filter
- **High-pass filter**
- Notch filter



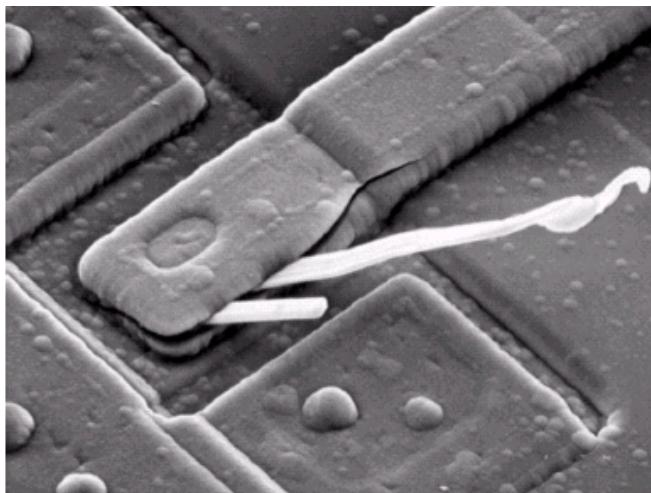
- Attenuates low frequencies  
- Passes high frequencies



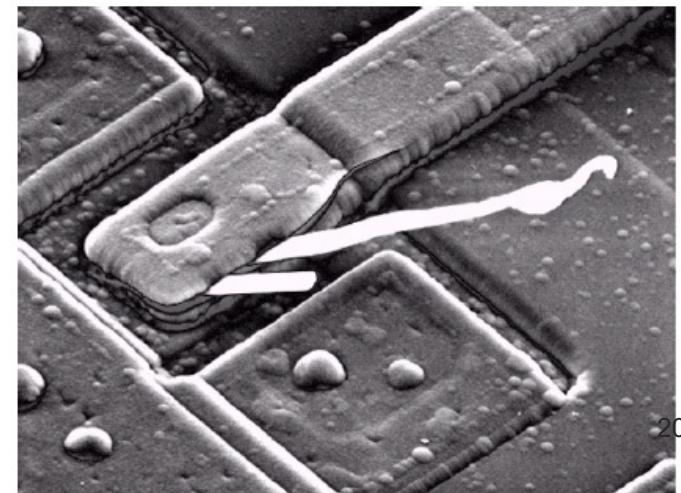
# Frequency-Domain Filtering

## Basic filters and their properties

- Low-pass filter
- **Modified high-pass filter**
  - Adding a constant of one-half the filter height to the filter function to avoid complete elimination
- Notch filter



Modified high-pass filter



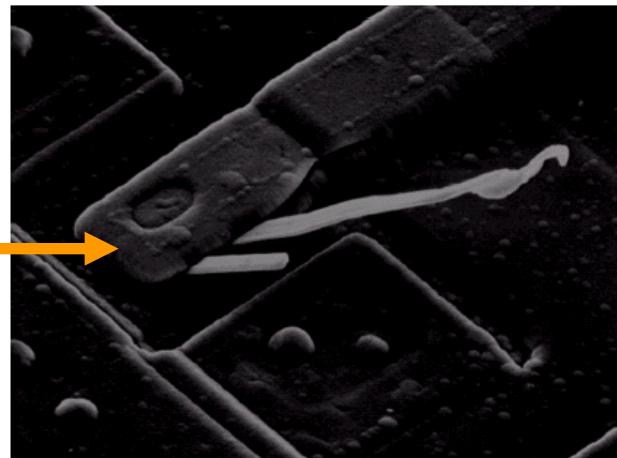
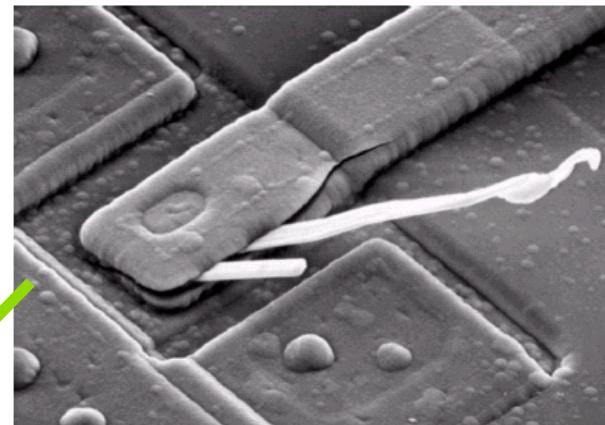
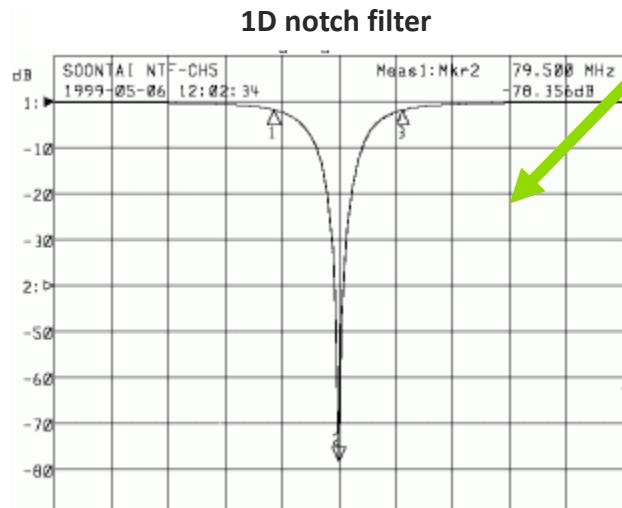
# [ Frequency-Domain Filtering ]

## ■ Basic filters and their properties

### ○ Notch filter

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

→ A constant function with a hole at the origin



# Frequency-Domain Filtering

## Convolution Theorem

- Discrete convolution

$$f(j, k) * h(j, k) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(j-m, k-n)$$

- Flip → shift → sum of products

- Fourier transform pairs

$$f(j, k) * h(j, k) \Leftrightarrow F(u, v)H(u, v)$$

$$f(j, k) \underline{h(j, k)} \Leftrightarrow F(u, v) * \underline{H(u, v)}$$

Impulse  
function

Impulse  
response

Convolution  
Theorem

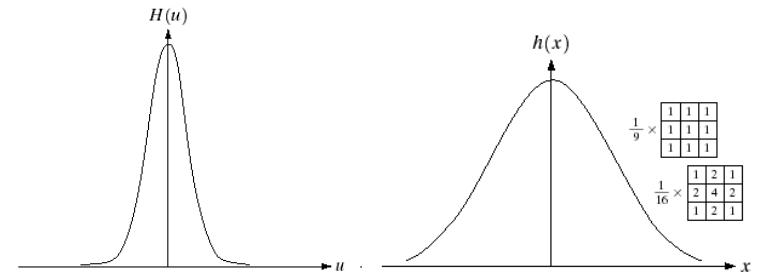
# Frequency-Domain Filtering

## Gaussian filter

### One-dimensional

$$H(u) = Ae^{-u^2/2\sigma^2}$$

$$h(k) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2 k^2}$$



### Important characteristics

- Easy to specify and manipulate
- Fourier transform of a Gaussian function is a ‘real’ Gaussian function
- Two functions behave reciprocally w.r.t one another



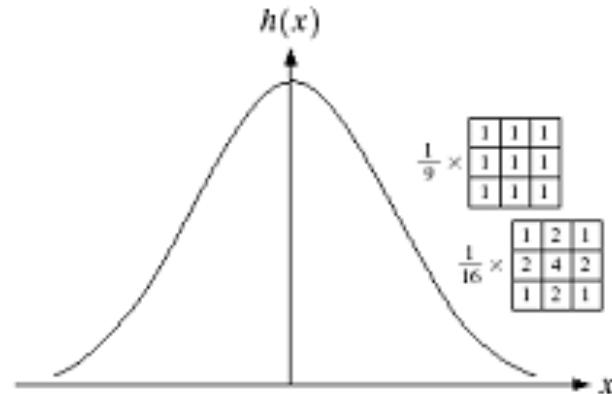
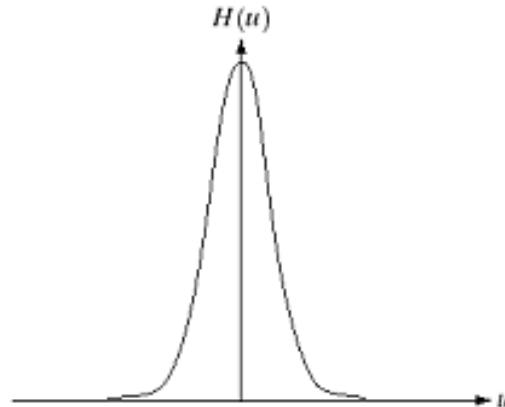
# Frequency-Domain Filtering

## Gaussian filter

### Low-pass filter

$$H(u) = Ae^{-u^2/2\sigma^2}$$

$$h(k) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2 k^2}$$



→ All the values are positive in both domains

→ The narrower the frequency domain filter, the wider the spatial domain filter  
i.e. more severe blurring effect

# Frequency-Domain Filtering

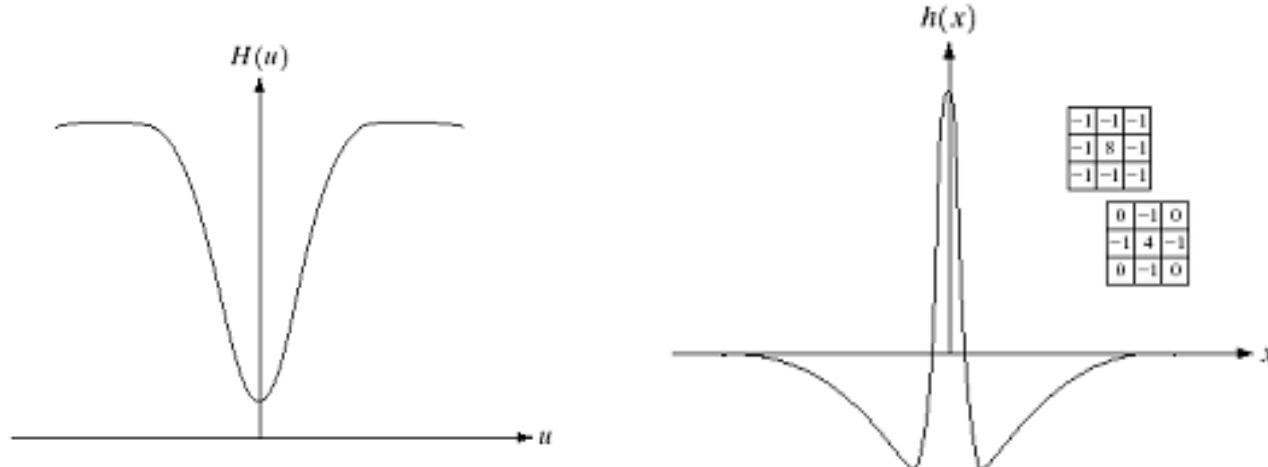
## Gaussian filter

### High-pass filter

- Construct a high-pass filter as a difference of Gaussians

$$H(u) = Ae^{-u^2/2\sigma_1^2} - Be^{-u^2/2\sigma_2^2}$$

$$h(k) = \sqrt{2\pi} \left( \sigma_1 A e^{-2\pi^2 \sigma_1^2 k^2} - \sigma_2 B e^{-2\pi^2 \sigma_2^2 k^2} \right)$$



# Comparison

- Comparison of spatial-domain and frequency-domain filtering
  - Filtering in spatial domain
    - Specific masks are needed
  - Filtering in frequency domain
    - $f(j,k) * h(j,k) \Leftrightarrow F(u,v)H(u,v)$
    - $f(j,k)h(j,k) \Leftrightarrow F(u,v) * H(u,v)$
    - Easy to implement
      - Fast Fourier Transform (FFT)
    - Save computation complexity for larger signal size

# Frequency-Domain Filtering

## ■ Smoothing frequency-domain filters

$$G(u, v) = H(u, v)F(u, v)$$

- Ideal low-pass filters (ILPF)
- Butterworth low-pass filters (BLPF)
- Gaussian low-pass filters (GLPF)

# Frequency-Domain Filtering

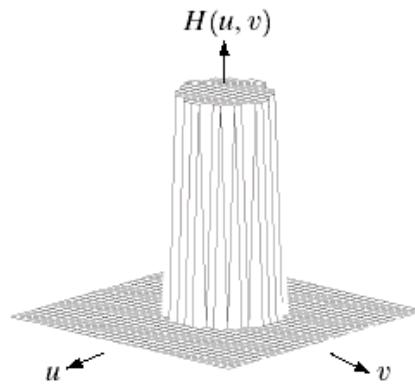
## Ideal low-pass filters

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

Radius: Non-negative

$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

< Distance from point  $(u,v)$  to the center of the frequency rectangle >



Perspective plot

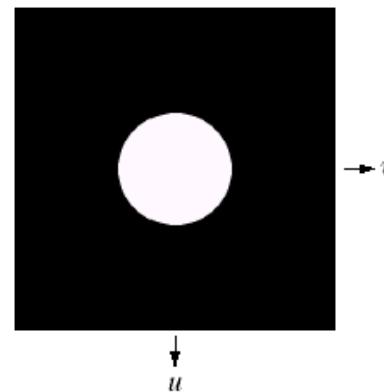
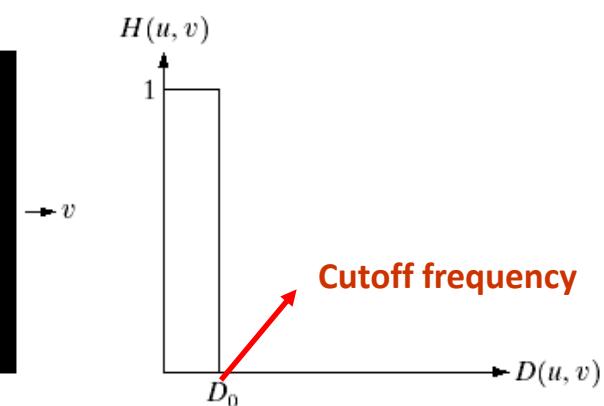


Image display

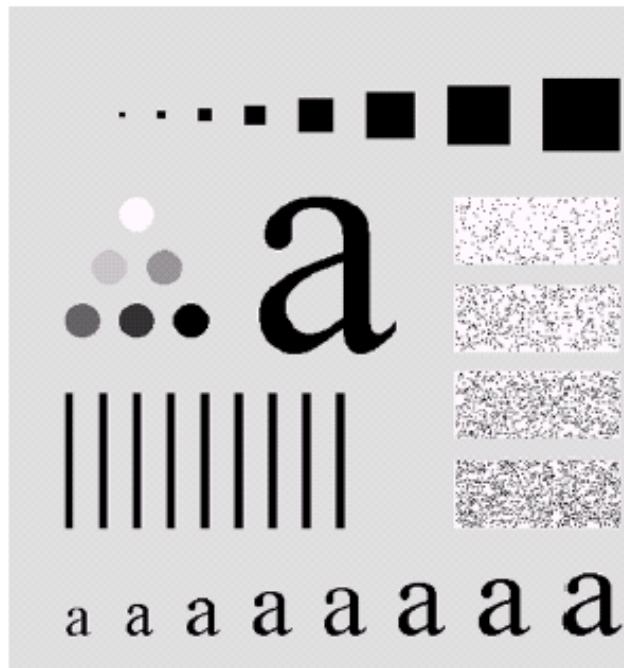


Radial cross section

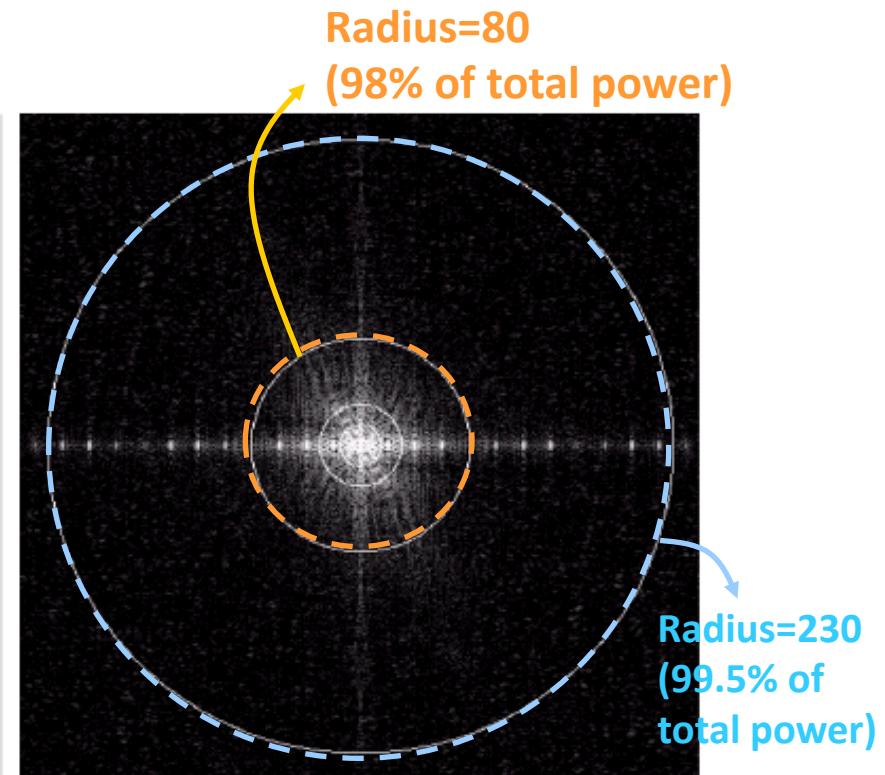
# Frequency-Domain Filtering

## Ideal low-pass filters

### Example



Original image  
(in spatial domain)

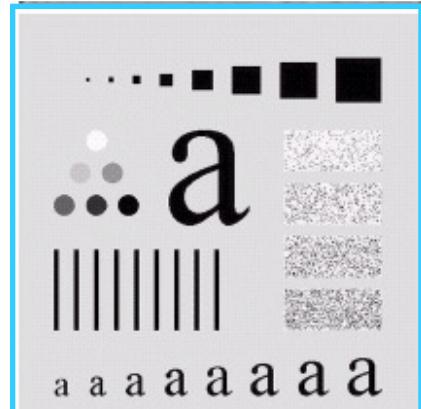
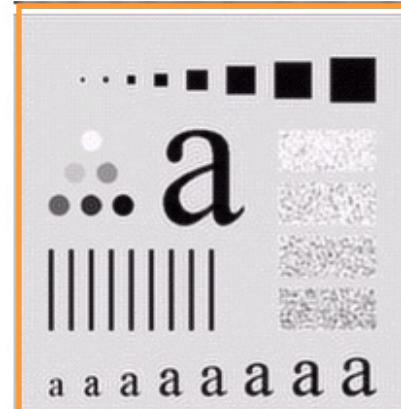
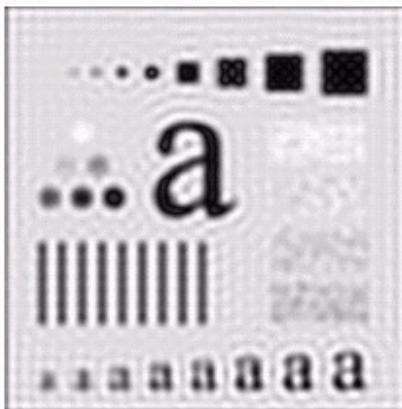
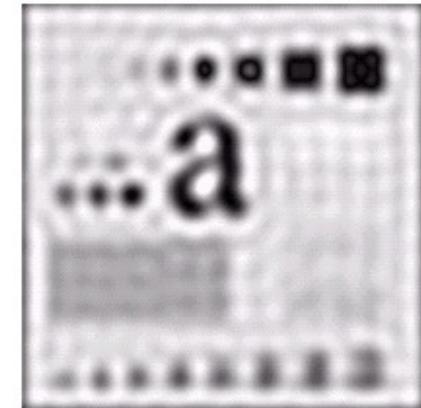
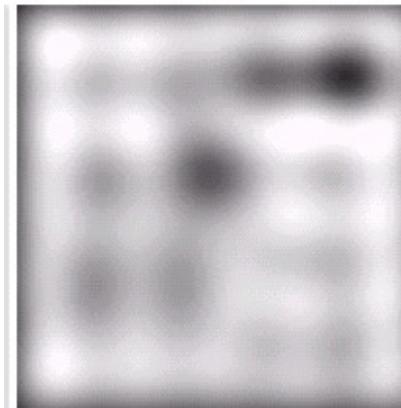
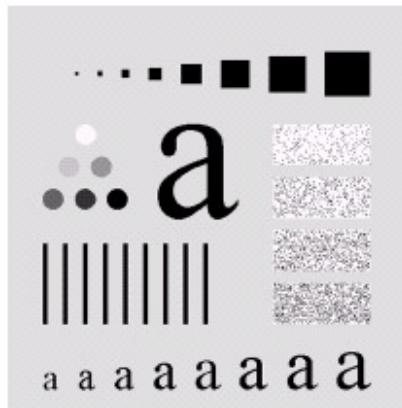


Radius = 5, 15, 30, 80 & 230  
(in frequency domain)

# [ Frequency-Domain Filtering ]

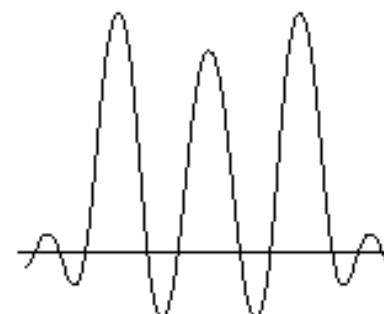
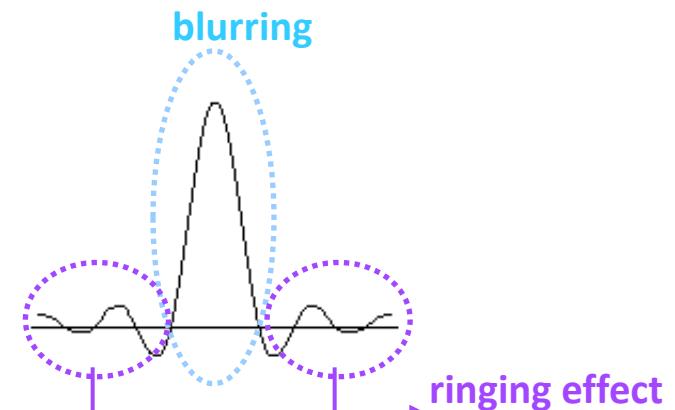
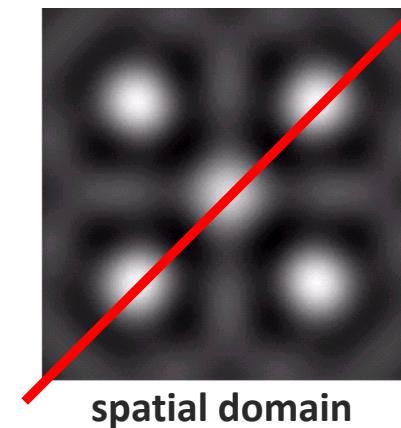
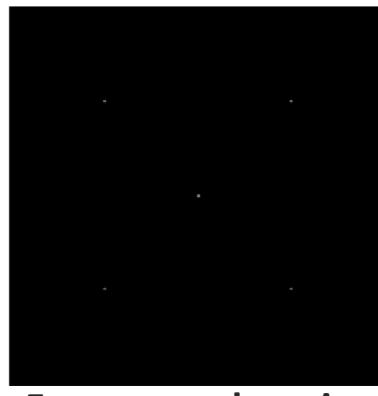
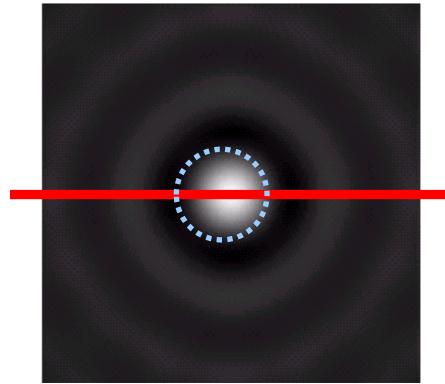
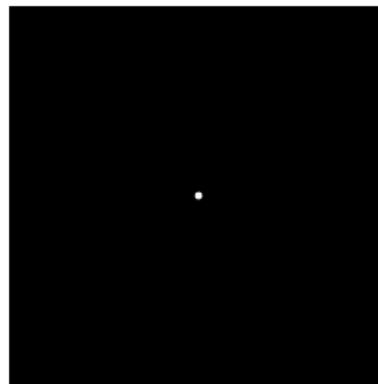
## ■ Ideal low-pass filters

- Example (with radius = 5, 15, 30, 80 & 230)



# [ Frequency-Domain Filtering ]

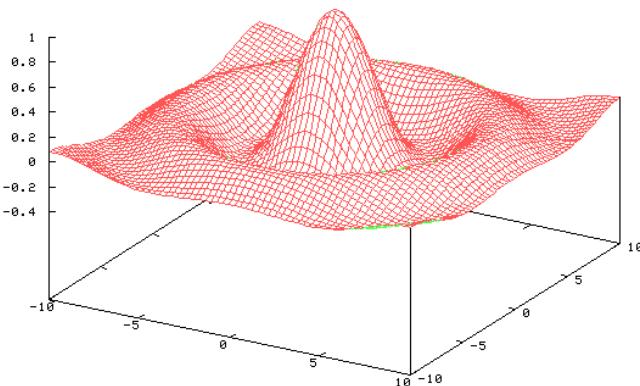
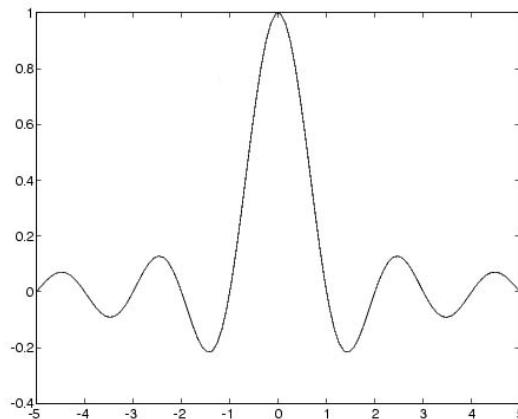
- Ideal low-pass filters
  - Example



# Frequency-Domain Filtering

## Ideal low-pass filters

- Ideal LPF presents a Sinc function in the spatial domain
- Radius of the main lobe is inversely proportional to the cutoff frequency



# Frequency-Domain Filtering

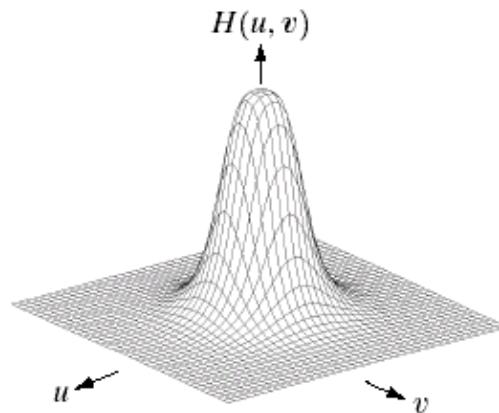
## ■ Butterworth low-pass filters

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

n: order (must be an integer)

$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

< Distance from point  $(u, v)$  to the center of the frequency rectangle >



Perspective plot

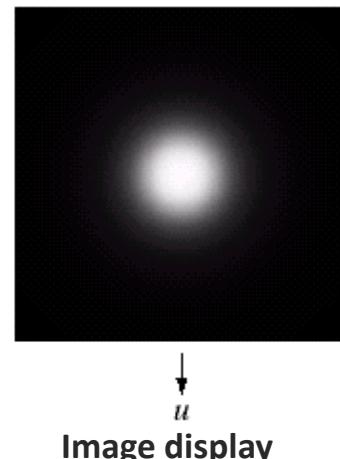
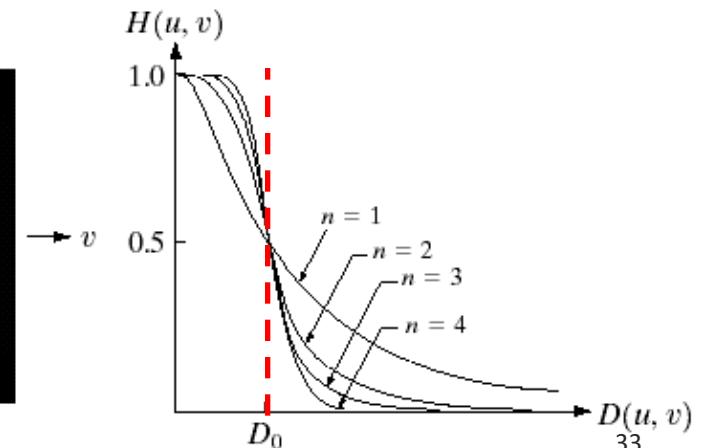


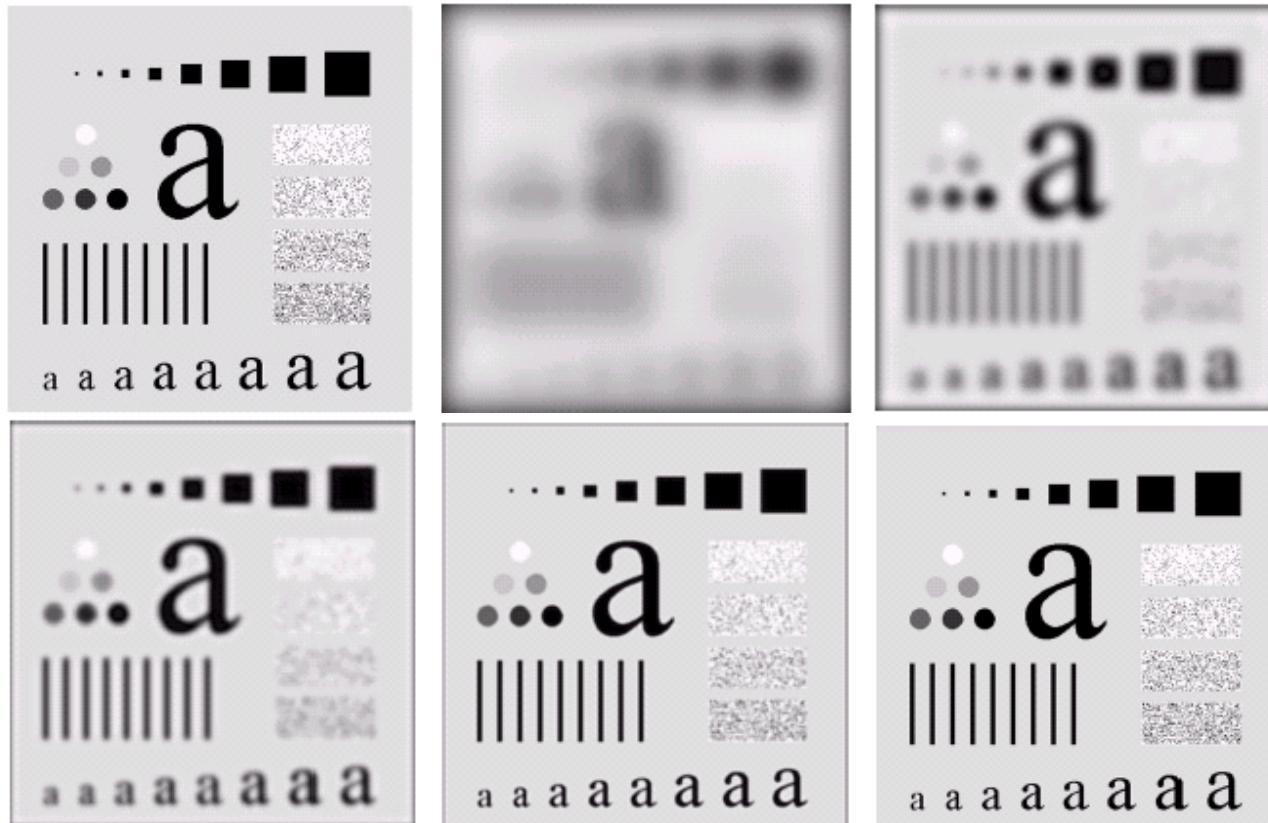
Image display



Radial cross section

# [ Frequency-Domain Filtering ]

- Butterworth low-pass filters ( $n=2$ )
  - Example (with radius = 5, 15, 30, 80 & 230)



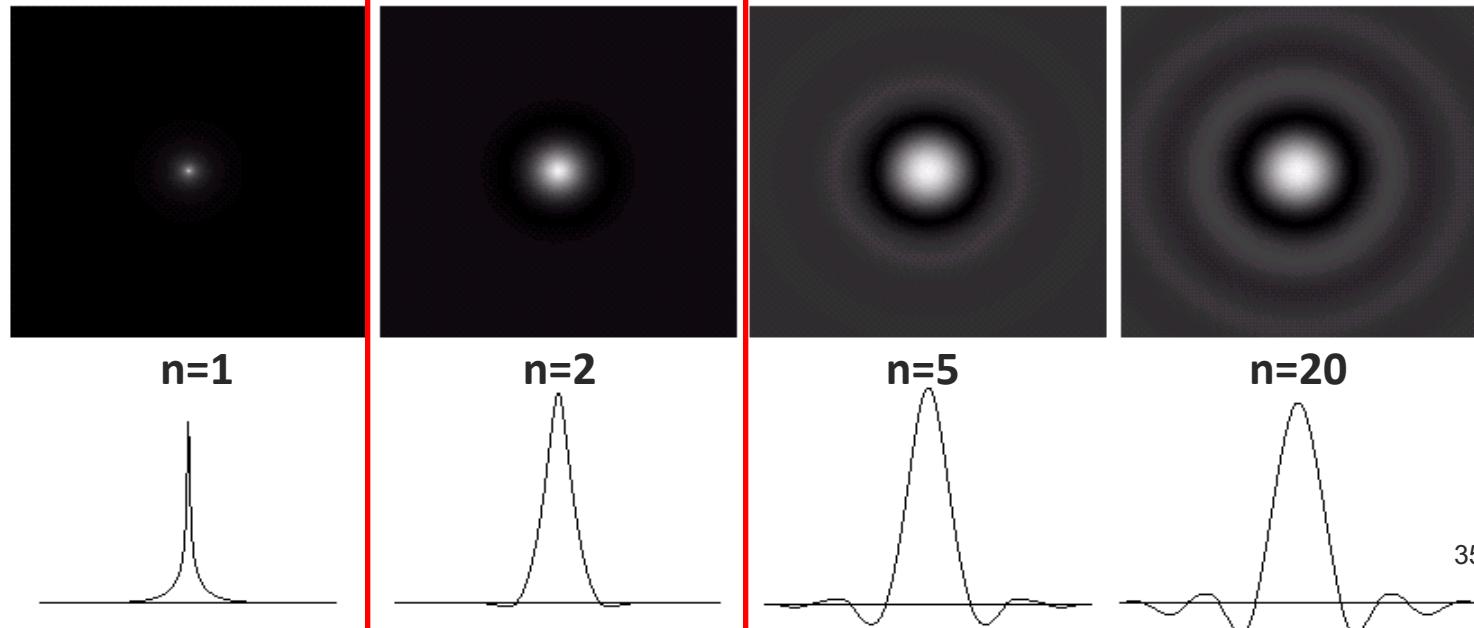
# Frequency-Domain Filtering

## Butterworth low-pass filters

- Order  $n = 1, 2, 5$ , and  $20$  with cutoff=5 pixels

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

Spatial representation of BLPF

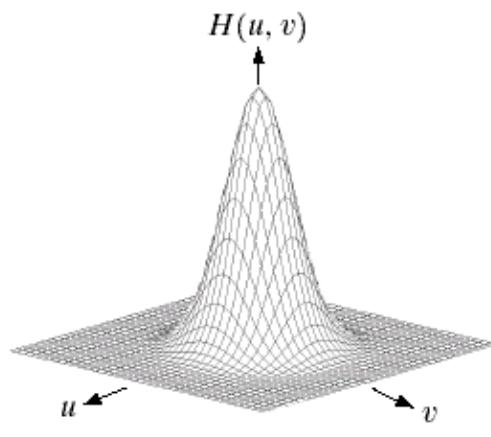


# Frequency-Domain Filtering

## Gaussian low-pass filters

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

$D_0$  is the cutoff frequency



Perspective plot

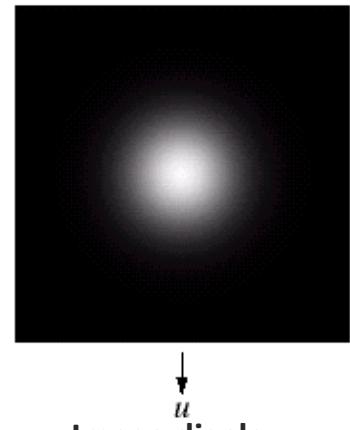
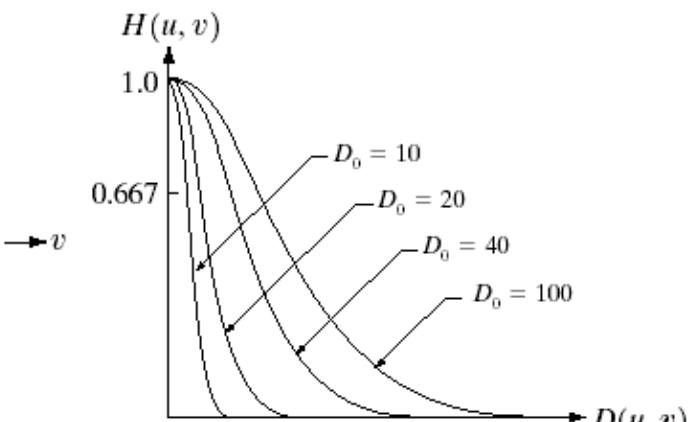


Image display



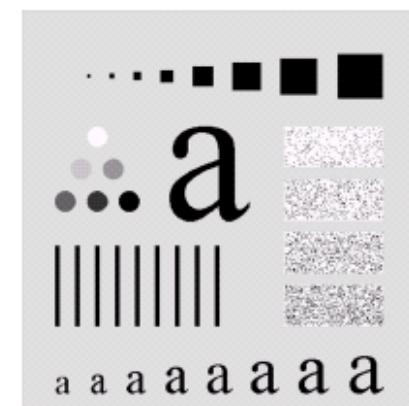
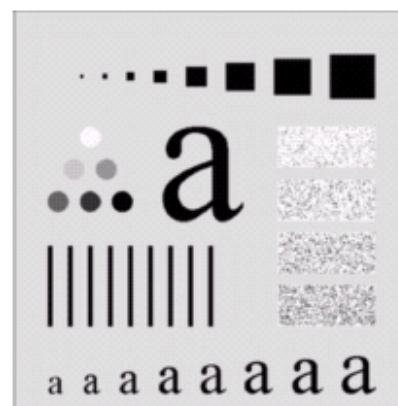
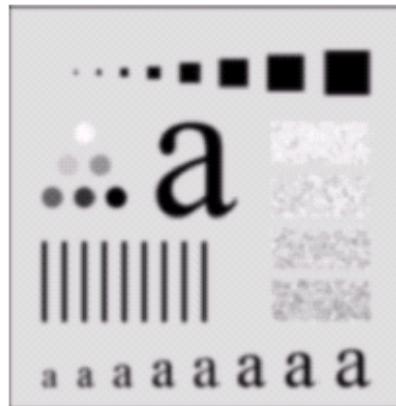
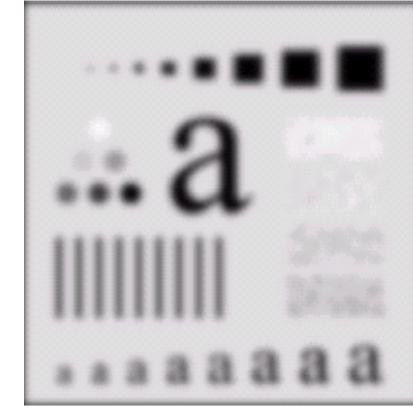
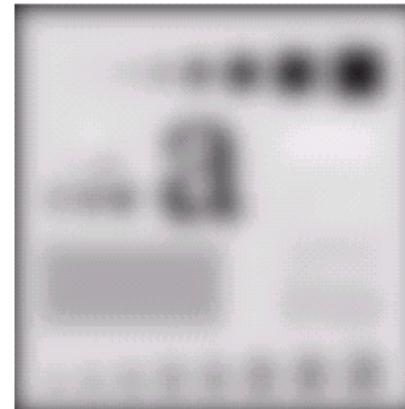
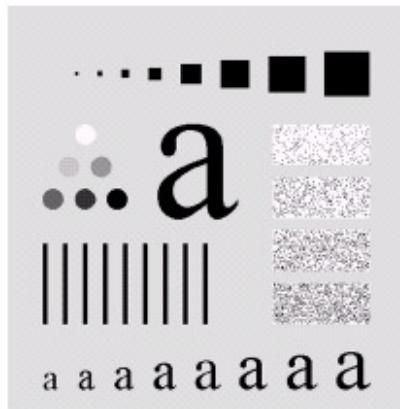
Radial cross section

→ Gaussian in frequency domain → Gaussian in spatial domain

→ No ringing artifacts

# [ Frequency-Domain Filtering ]

## ■ Gaussian low-pass filters



“ No ringing artifacts ”

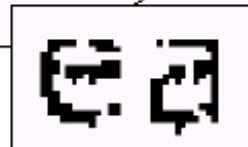
“ achieve less smoothing than BLPF ”

# [Frequency-Domain Filtering]

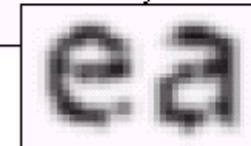
## ■ Examples

- Gaussian low-pass filters
  - Connect broken characters

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



# Frequency-Domain Filtering

## ■ Examples

- Gaussian low-pass filters
  - Reduction in skin fine lines



Original



Cutoff=100



Cutoff=80

# Frequency-Domain Filtering

## ■ Sharpening frequency-domain filters

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

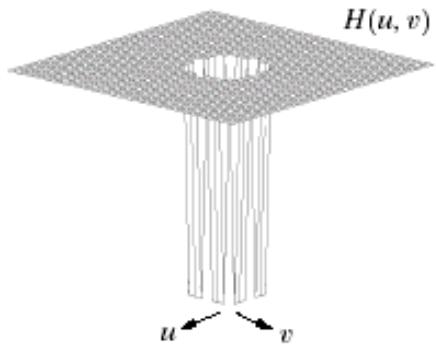
- Ideal high-pass filters
- Butterworth high-pass filters
- Gaussian high-pass filters
- The Laplacian in the frequency domain

# Frequency-Domain Filtering

## Ideal high-pass filters

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

$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$



Perspective plot

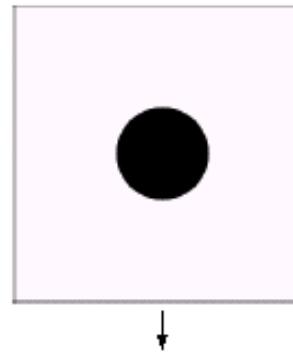
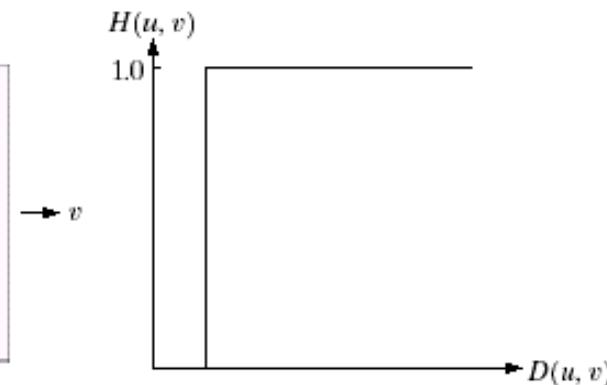
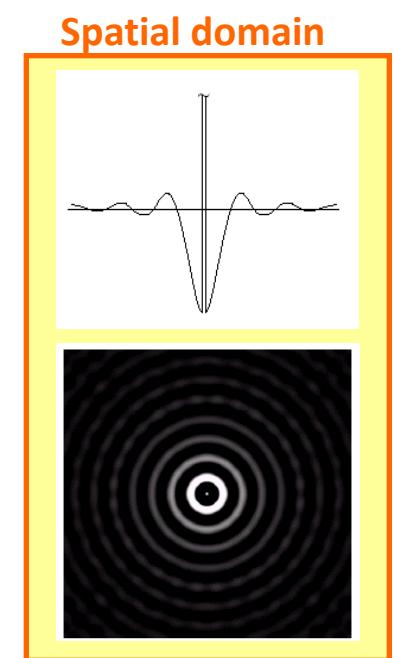


Image display



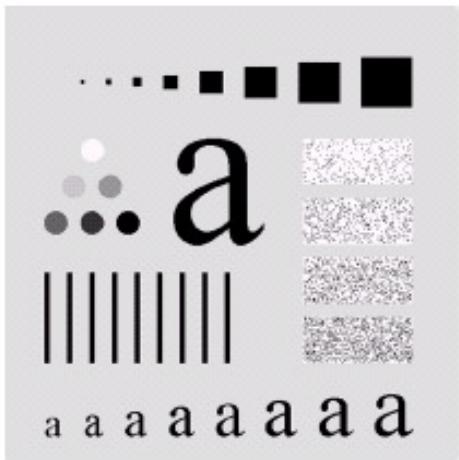
Radial cross section



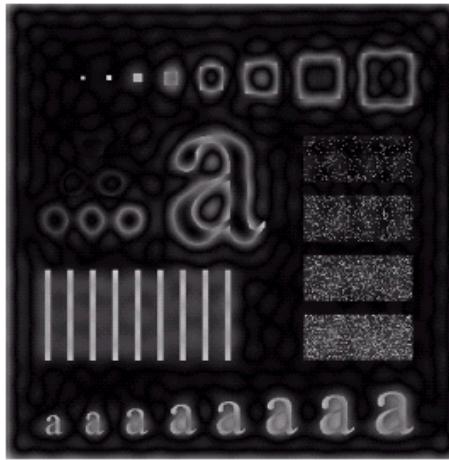
Spatial domain

# [ Frequency-Domain Filtering ]

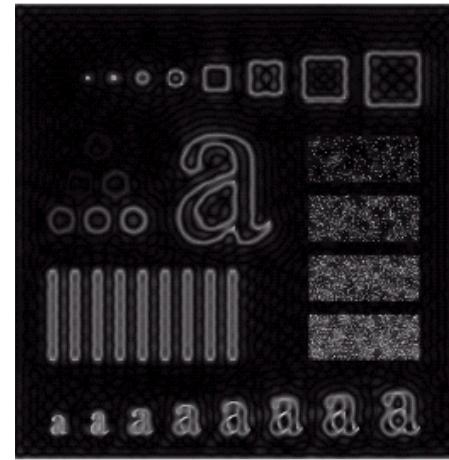
- Ideal high-pass filters
  - Example (with radius = 15, 30 & 80)



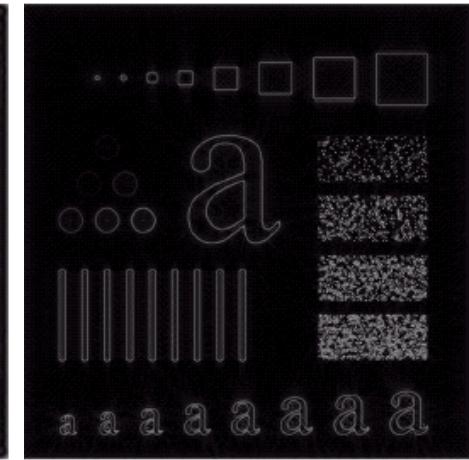
Original



Cutoff=15



Cutoff=30



Cutoff=80

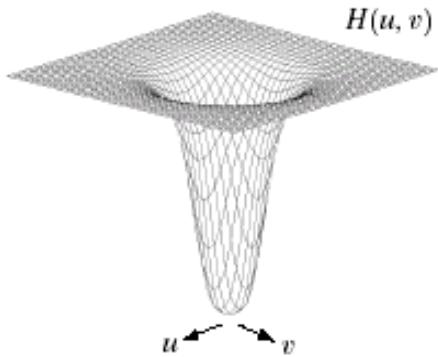
# Frequency-Domain Filtering

## ■ Butterworth high-pass filters

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

n: order (must be an integer)

$$D(u, v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$



Perspective plot

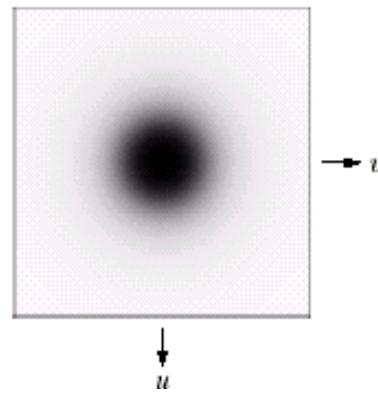
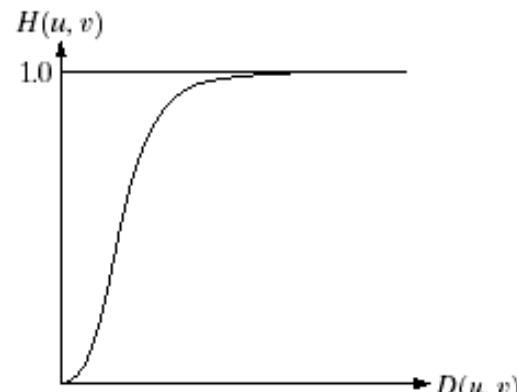
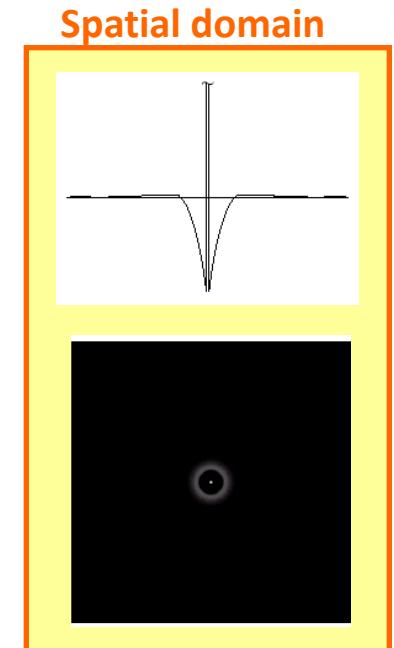


Image display



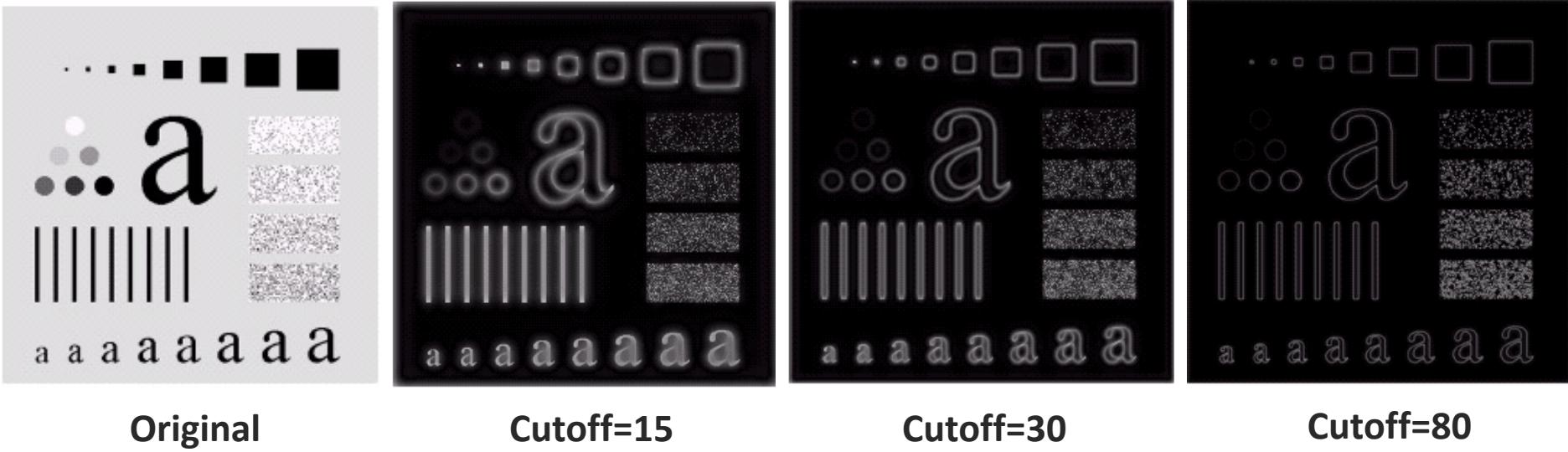
Radial cross section



Spatial domain

# Frequency-Domain Filtering

- Butterworth high-pass filters ( $n=2$ )
  - Example (with radius = 15, 30 & 80)

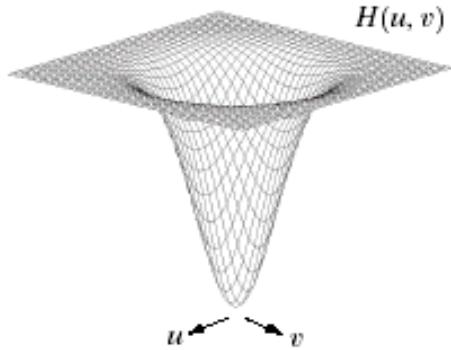


# [ Frequency-Domain Filtering ]

## ■ Gaussian high-pass filters

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

$D_0$  is the cutoff frequency



Perspective plot

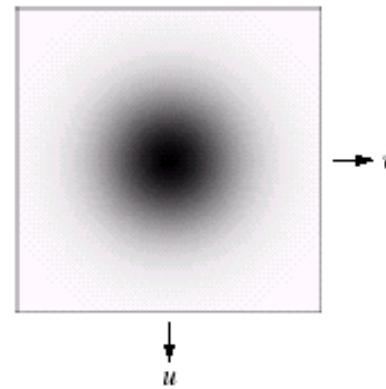
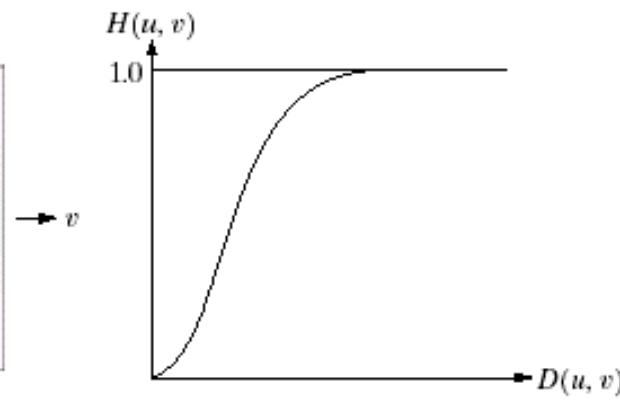
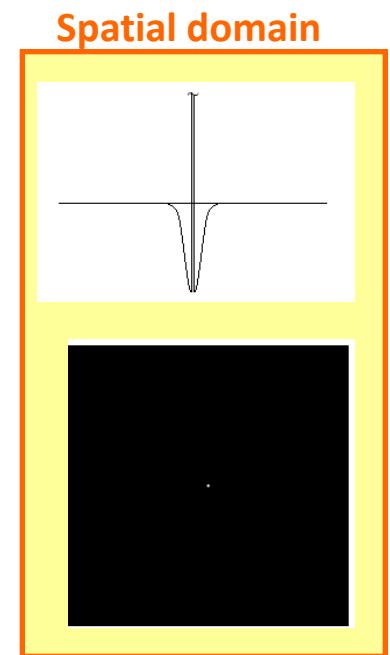


Image display

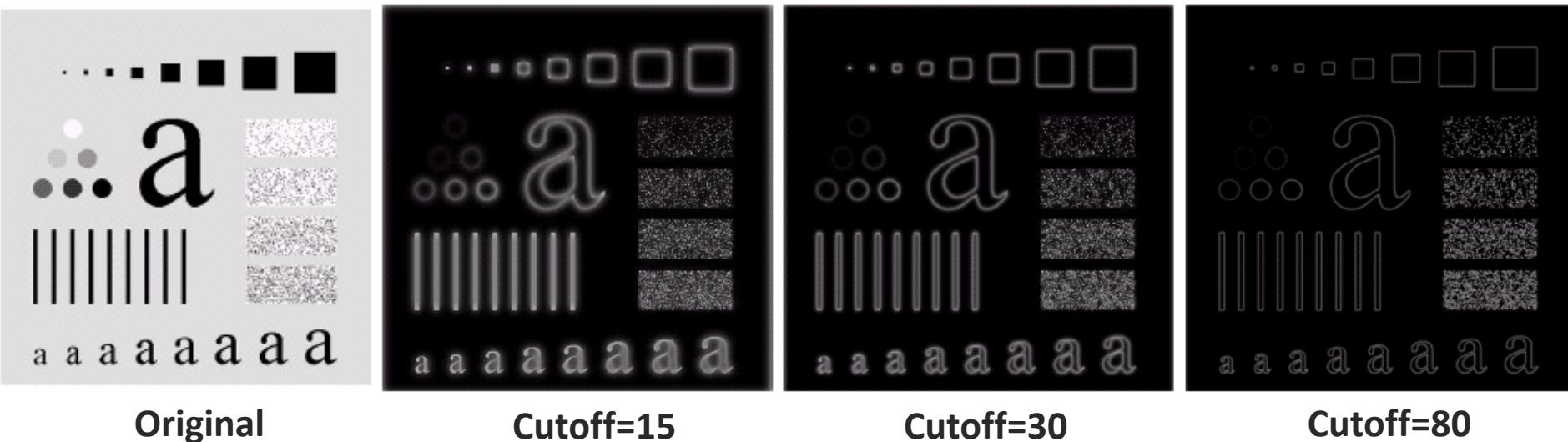


Radial cross section



# Frequency-Domain Filtering

- Gaussian high-pass filters
  - Example (with radius = 15, 30 & 80)



//Note// High-pass filters can be constructed by the difference of Gaussian low-pass filters  
→ more parameters → more control over the filter shape

# Frequency-Domain Filtering

## The Laplacian in the Frequency domain

$$\mathfrak{F} \left[ \frac{d^n f(x)}{dx^n} \right] = (ju)^n F(u)$$

$$\begin{aligned} \mathfrak{F} \left[ \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \right] &= (ju)^2 F(u, v) + (jv)^2 F(u, v) \\ &= -(u^2 + v^2) F(u, v) \end{aligned}$$

$$\mathfrak{F} [\nabla^2 f(x, y)] = -(u^2 + v^2) F(u, v)$$

$$G(u, v) = H(u, v) F(u, v)$$

$$\Rightarrow H(u, v) = -(u^2 + v^2)$$

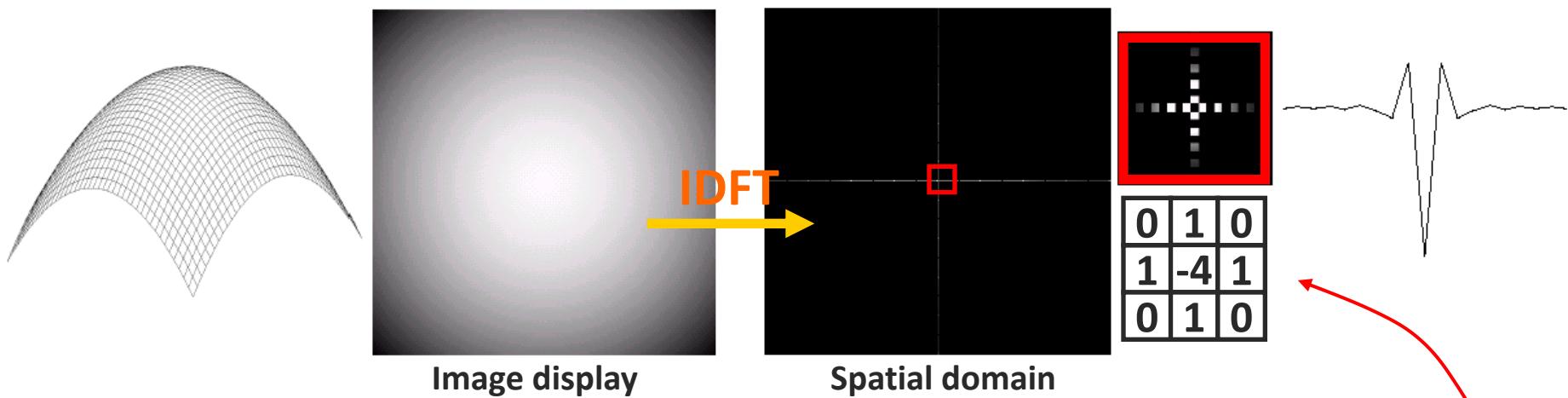
Center of the filter needs to be shifted

$$\Rightarrow H(u, v) = - \left[ \left( u - \frac{M}{2} \right)^2 + \left( v - \frac{N}{2} \right)^2 \right]$$

# [ Frequency-Domain Filtering ]

## ■ The Laplacian in the Frequency domain

$$\Im[\nabla^2 f(x, y)] = -((u - M/2)^2 + (v - N/2)^2) F(u, v)$$



$$H(u, v) = -\left[ \left( u - \frac{M}{2} \right)^2 + \left( v - \frac{N}{2} \right)^2 \right]$$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

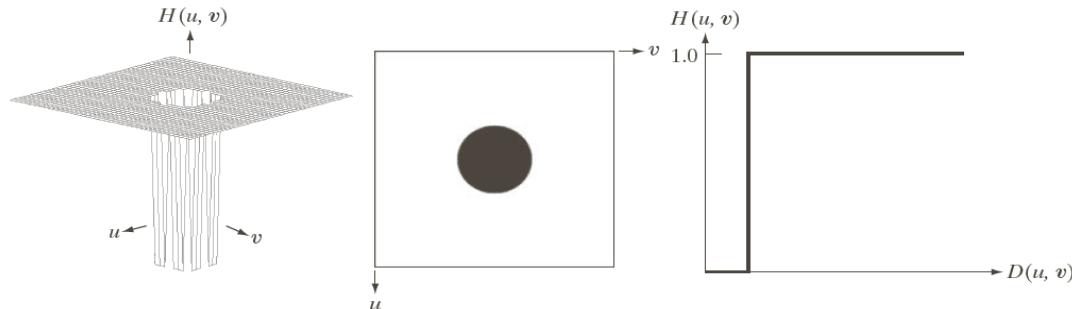
$$\nabla^2 f = f(x+1, y) + f(x-1, y)$$

$$+ f(x, y+1) + f(x, y-1) - 4f(x, y)$$

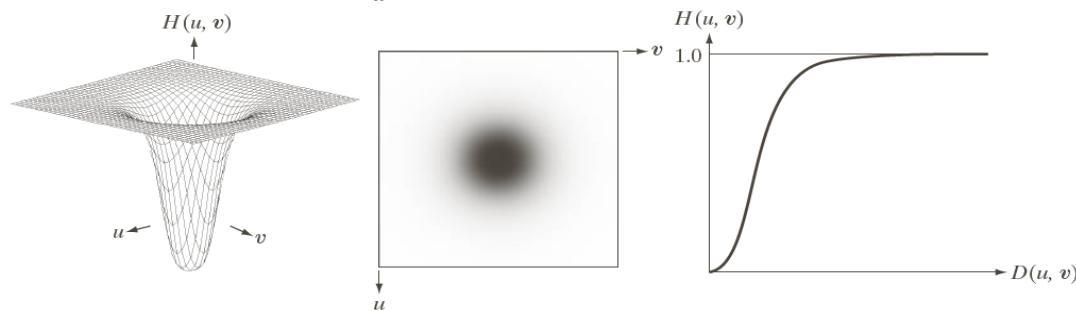
# Frequency-Domain Filtering

## Comparison – frequency domain

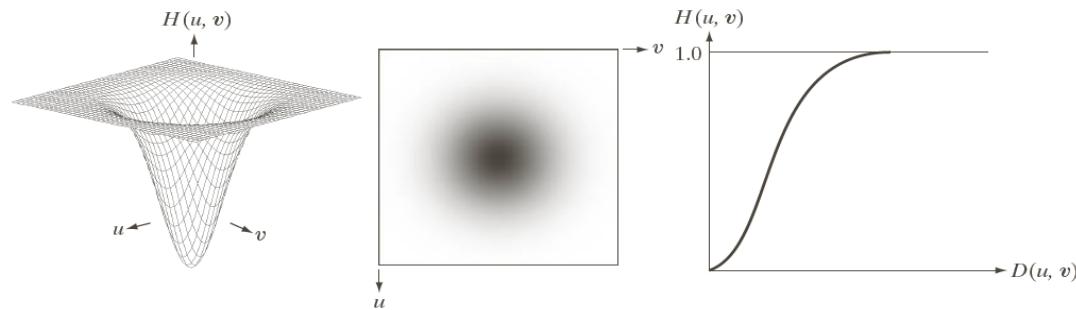
Ideal



Butterworth

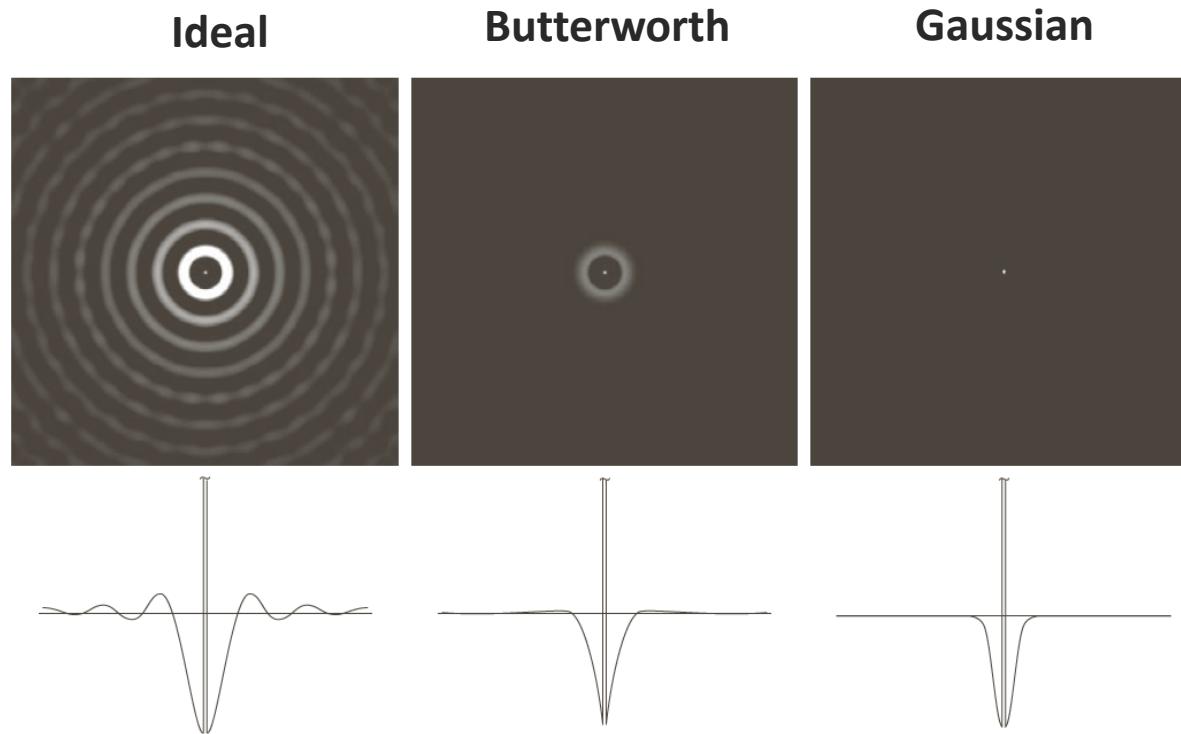


Gaussian



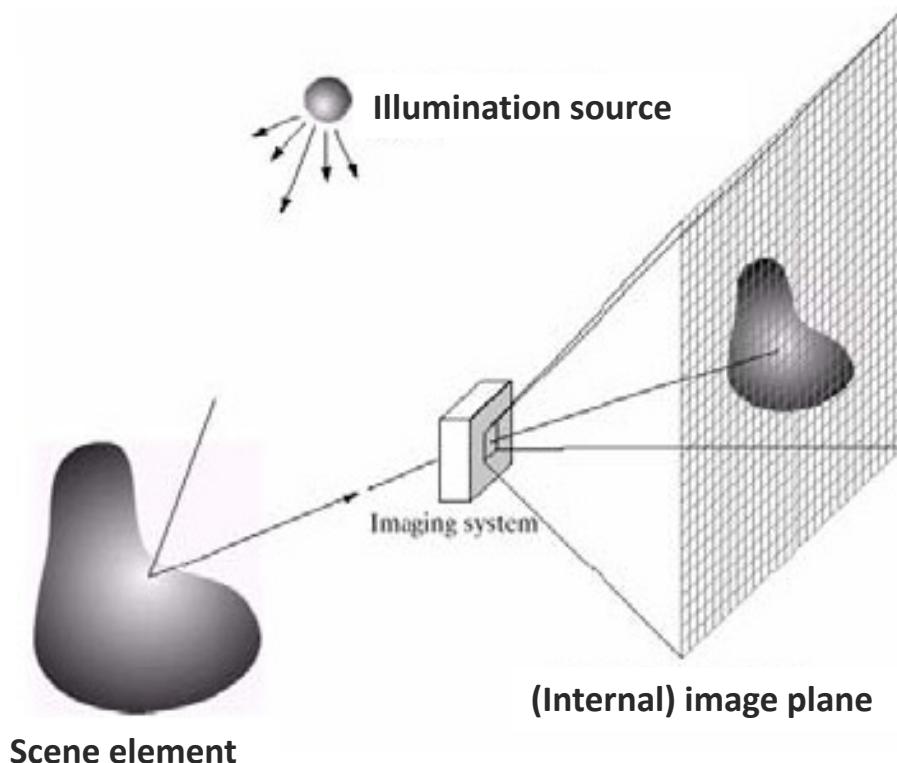
# [ Frequency-Domain Filtering ]

## ■ Comparison – spatial representation



# [ Frequency-Domain Filtering ]

- Homomorphic filtering
  - Recall: (Lecture #2)



$$f(x, y) = i(x, y)r(x, y)$$

- Illumination

$$0 < i(x, y) < \infty$$

- Reflectance

$$0 < r(x, y) < 1$$

# Frequency-Domain Filtering

## Homomorphic filtering

$$f(x, y) = i(x, y)r(x, y)$$

- The Fourier transform of the product of two functions is NOT separable

$$\mathfrak{F}\{f(x, y)\} \neq \mathfrak{F}\{i(x, y)\} \mathfrak{F}\{r(x, y)\}$$

- Define

$$z(x, y) = \ln f(x, y) = \ln i(x, y) + \ln r(x, y)$$

$$\begin{aligned} \rightarrow \mathfrak{F}\{z(x, y)\} &= \mathfrak{F}\{\ln f(x, y)\} \\ &= \mathfrak{F}\{\ln i(x, y)\} + \mathfrak{F}\{\ln r(x, y)\} \end{aligned}$$

$$or \quad Z(u, v) = F_i(u, v) + F_r(u, v)$$

# Frequency-Domain Filtering

## Homomorphic filtering

$$S(u, v) = H(u, v)Z(u, v)$$

$$= H(u, v)F_i(u, v) + H(u, v)F_r(u, v)$$

$$\Rightarrow s(x, y) = \mathcal{F}^{-1}\{S(u, v)\}$$

$$= \mathcal{F}^{-1}\{H(u, v)F_i(u, v)\} + \mathcal{F}^{-1}\{H(u, v)F_r(u, v)\}$$

○ let

$$i'(x, y) = \mathcal{F}^{-1}\{H(u, v)F_i(u, v)\}$$

$$r'(x, y) = \mathcal{F}^{-1}\{H(u, v)F_r(u, v)\}$$

$$\Rightarrow s(x, y) = i'(x, y) + r'(x, y)$$

# Frequency-Domain Filtering

## ■ Homomorphic filtering

$$s(x, y) = i'(x, y) + r'(x, y)$$

$$\begin{aligned} g(x, y) &= e^{s(x, y)} \\ &= e^{i'(x, y)} \cdot e^{r'(x, y)} = i_0(x, y)r_0(x, y) \end{aligned}$$

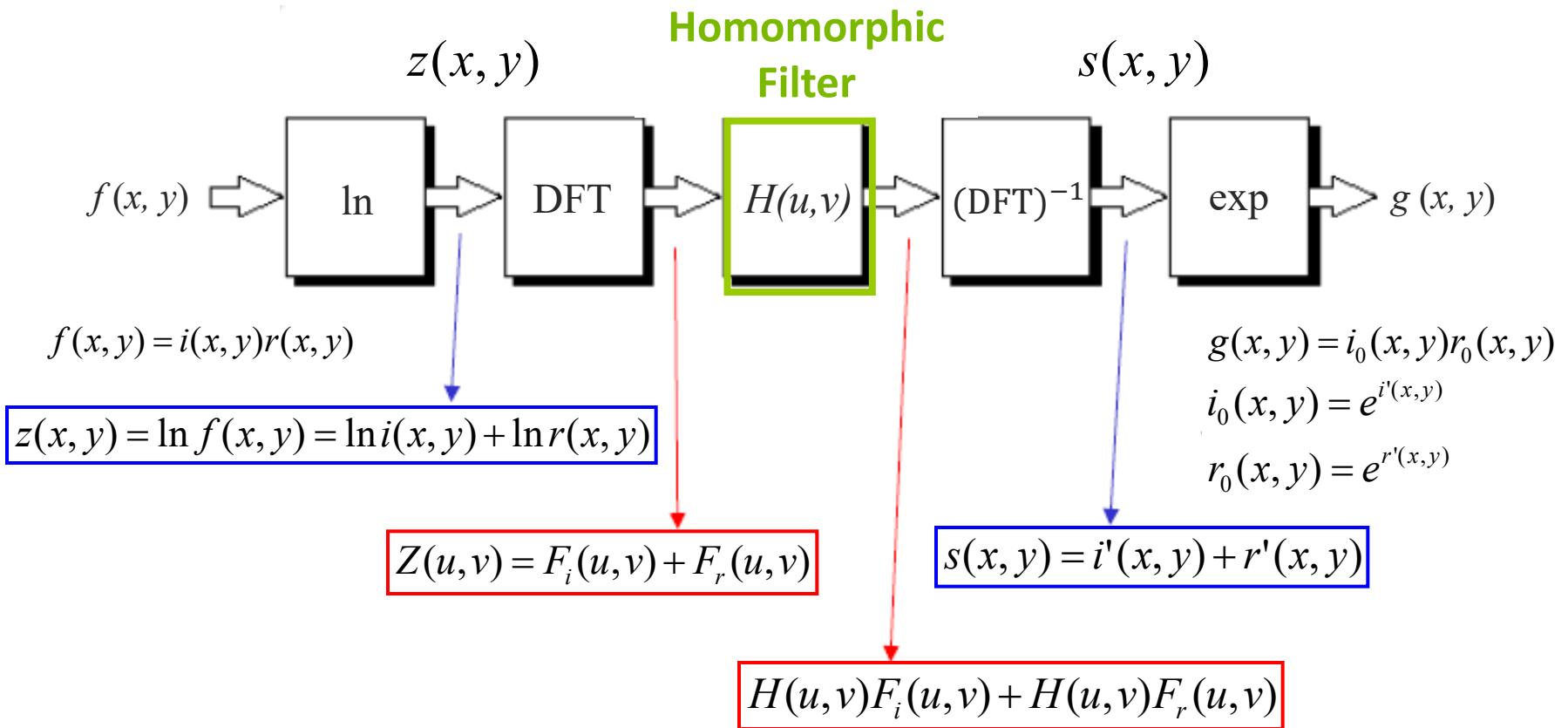
where

$$i_0(x, y) = e^{i'(x, y)} \quad \text{and} \quad r_0(x, y) = e^{r'(x, y)}$$

are the illumination and reflectance components of the output image

# Frequency-Domain Filtering

## Homomorphic filtering



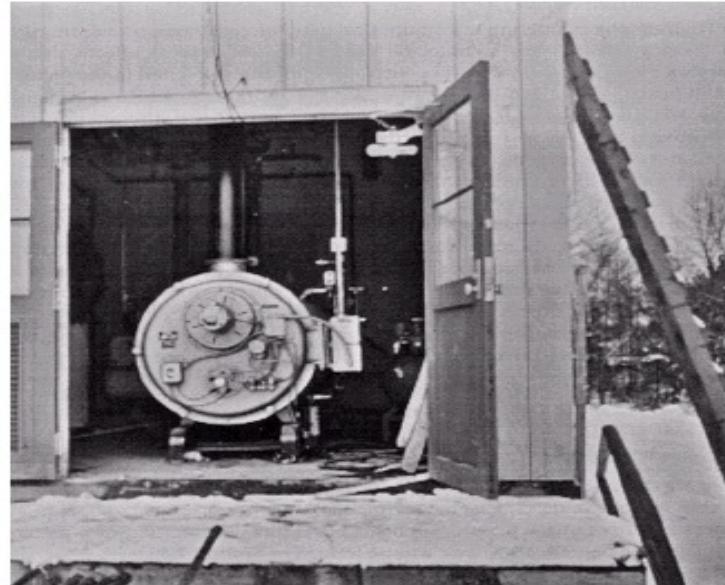
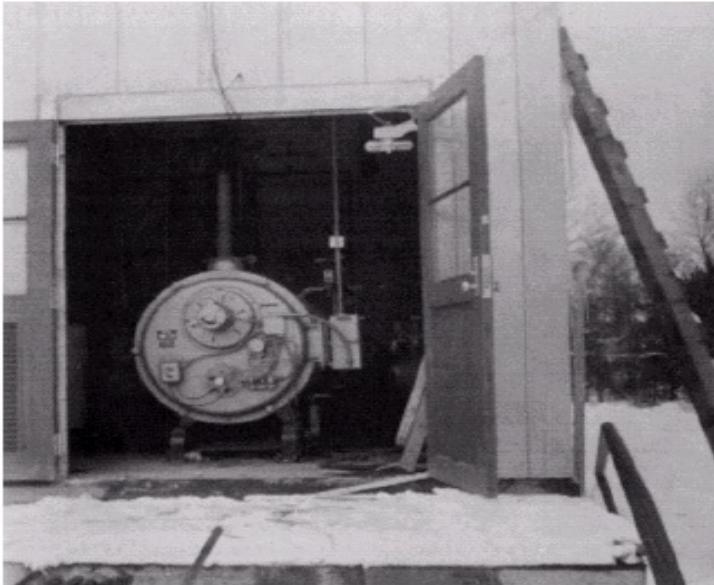
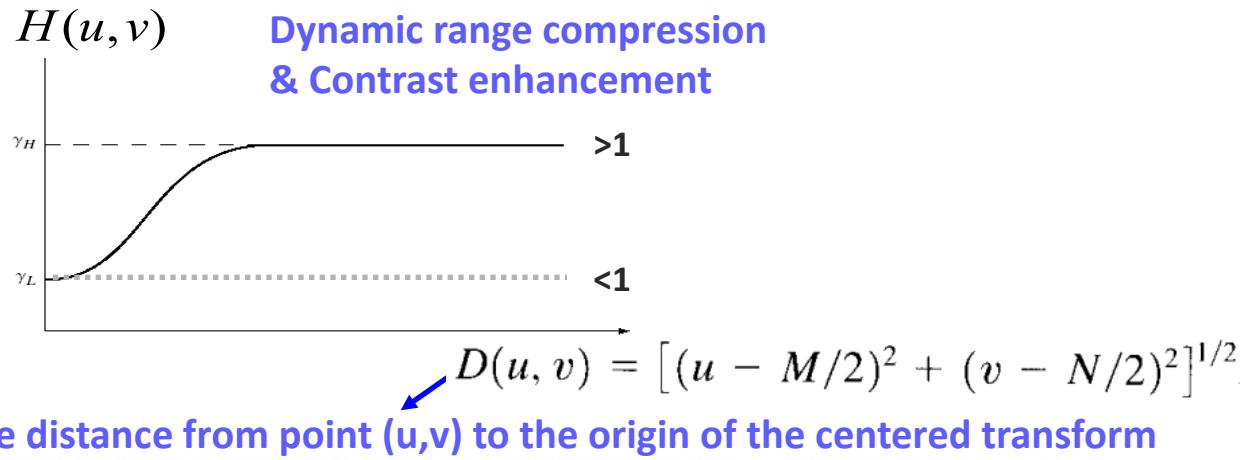
# [Frequency-Domain Filtering]

- Two components
  - Illumination component
    - Vary slowly (slow spatial variations)
    - Associate low frequencies
  - Reflectance component
    - Vary abruptly
    - Associate high frequencies
- Homomorphic filter
  - Separation of illumination & reflectance components
  - Affect the low- and high-frequency components of the Fourier transform in different ways

# Frequency-Domain Filtering

## ■ Example

“circularly  
symmetric



# [Frequency-Domain Filtering]

- Another example of homomorphic filtering
  - PET image

