Digital Image Processing

Lecture #9

Ming-Sui (Amy) Lee

Announcement

- **Class Information**
  - **The following schedule**

03/23	Lecture 5	05/11	proposal
03/30	Lecture 6 & 7	05/18	Lecture 11
04/06	Lecture 8	05/25	Lecture 12
04/13	RealSense	06/01	Lecture 13
04/20	midterm	06/08	Demo
04/27	RealSense & Lecture 9	06/15	Demo
05/04	Lecture 10	06/22	Final Package Due

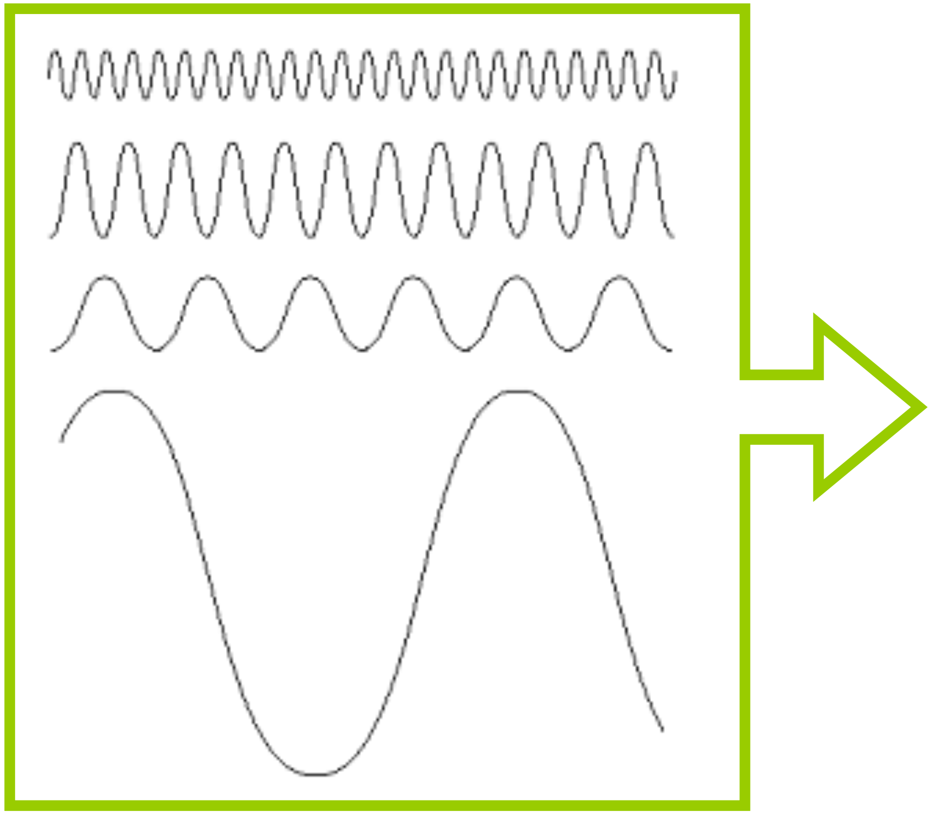
Image Enhancement in Frequency Domain

Background

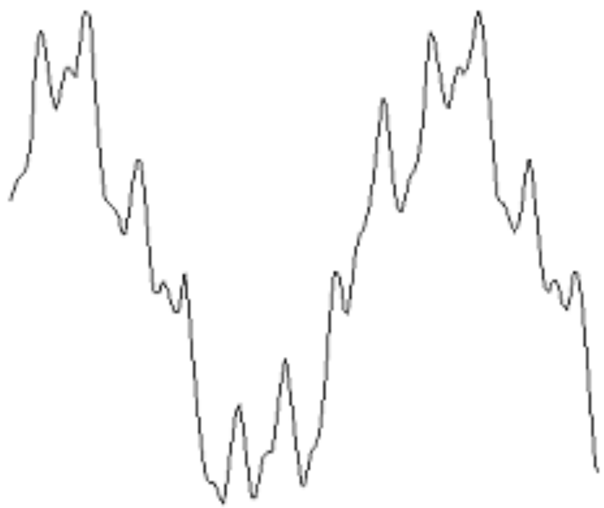
- Jean Baptiste Joseph Fourier (1807)
  - Any periodic function
    - Can be expressed as the sum of sines and/or cosines of different frequencies, each multiplied by a different coefficient → Fourier series
  - Functions that are not periodic
    - Can be expressed as the integral of sines and/or cosines 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**

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

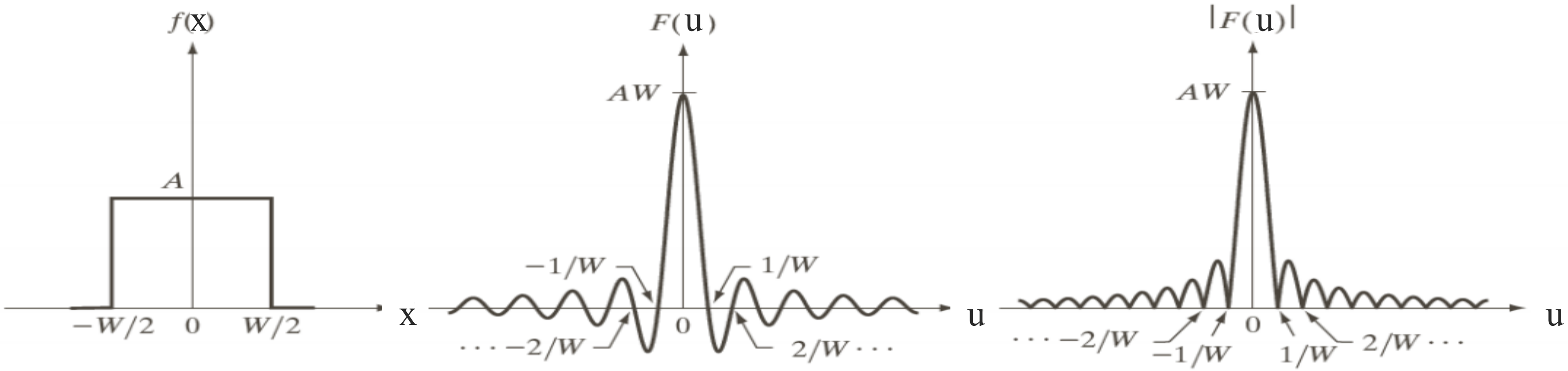
$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) e^{j2\pi(ux+vy)} dudv$$

Fourier Transform

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

$$F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-j2\pi ux} dx = \int_{-W/2}^{W/2} A e^{-j2\pi ux} dx = \frac{-A}{j2\pi u}\left[e^{-j2\pi ux}\right]_{-W/2}^{W/2}$$

$$= \frac{-A}{j2\pi u}\left[e^{-j\pi uW} - e^{j\pi uW}\right] = \frac{A}{j2\pi u}\left[e^{j\pi uW} - e^{-j\pi uW}\right] = AW\frac{\sin(\pi uW)}{\pi uW}$$



Fourier Transform

- **Discrete cases**
  - **Discrete Fourier Transform (DFT)**
    - **One-dimensional**

Let $F(u) \overset{\Delta}{=} F(u\Delta u)$ $\qquad \Delta u = \dfrac{1}{M\Delta j}$

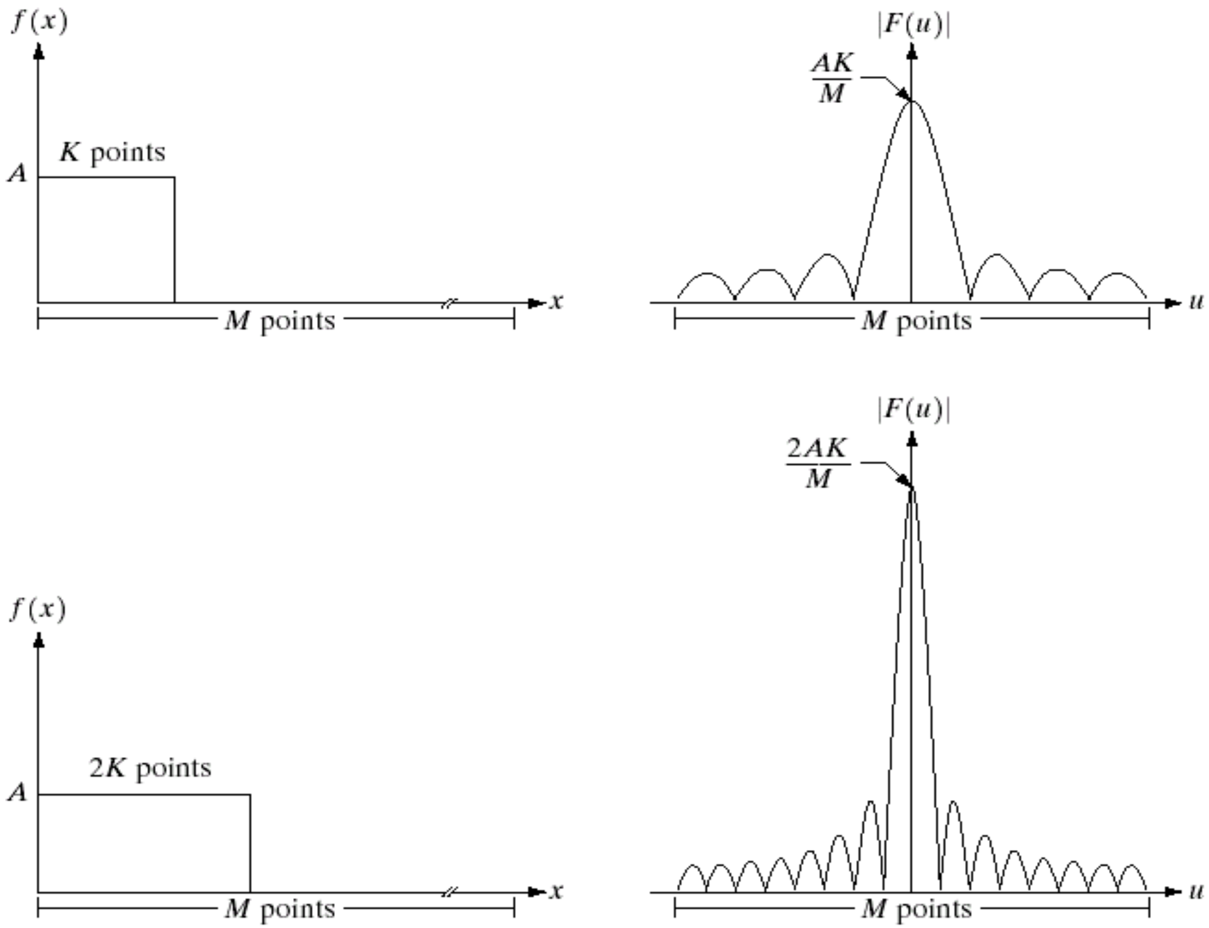
$$f(k) \overset{\Delta}{=} f(j_0 + k\Delta j)$$

$$\Rightarrow \begin{cases} F(u) = \dfrac{1}{M}\displaystyle\sum_{k=0}^{M-1} f(k)e^{-j2\pi uk/M}, \quad u = 0,1,\ldots,M-1 \\ f(k) = \displaystyle\sum_{u=0}^{M-1} F(u)e^{j2\pi uk/M}, \quad k = 0,1,\ldots,M-1 \end{cases}$$

//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

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

frequency component

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

Fourier Transform

- Discrete cases
  - Two-dimensional

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

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

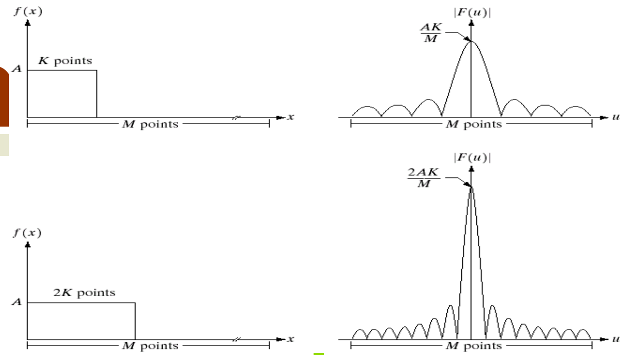
Fourier Spectrum: $$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$$

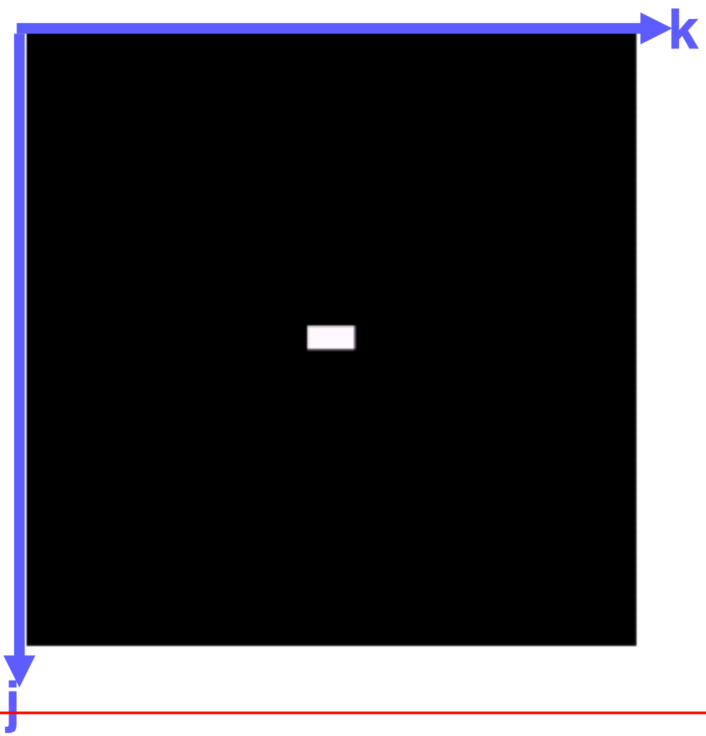
Phase Angle: $$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$$

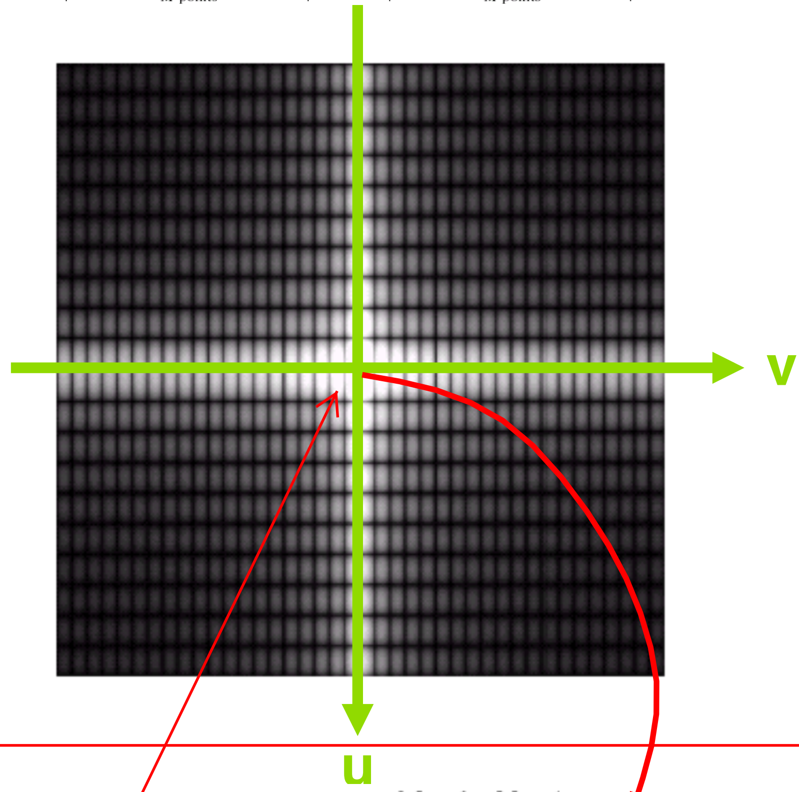
Power Spectrum: $$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v)$$

Fourier Transform

- Example







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

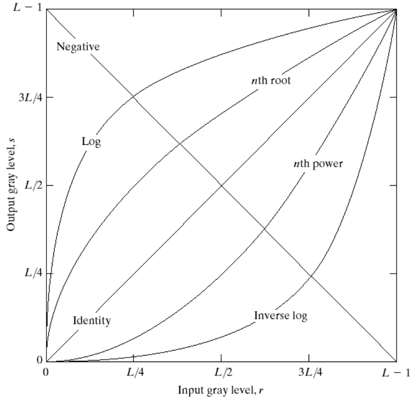
Centering & log transformation

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

Average/dc component (zero frequency)

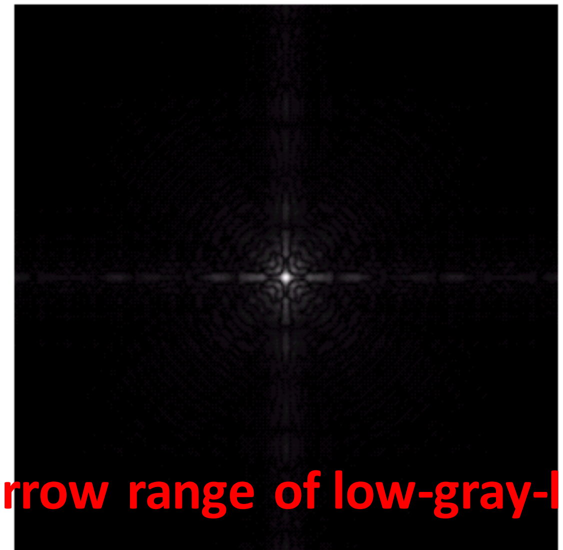
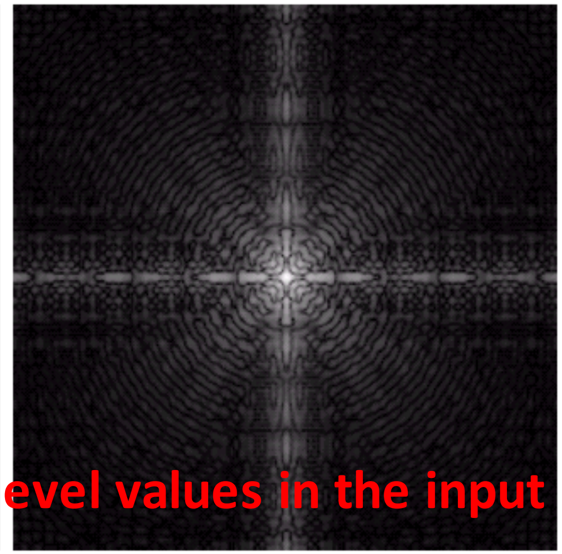
Fourier Transform

- **Log Transformation**



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

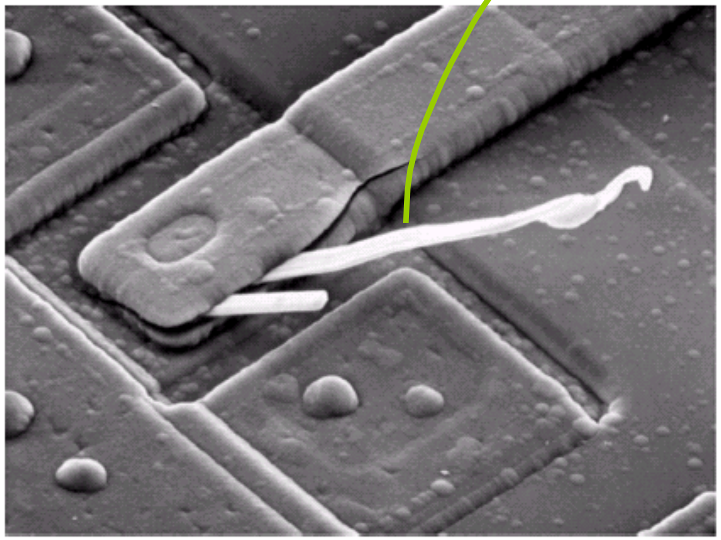
  - Expand the dynamic range of low gray-level values
  - Compress the dynamic range of images with large variations in pixel values

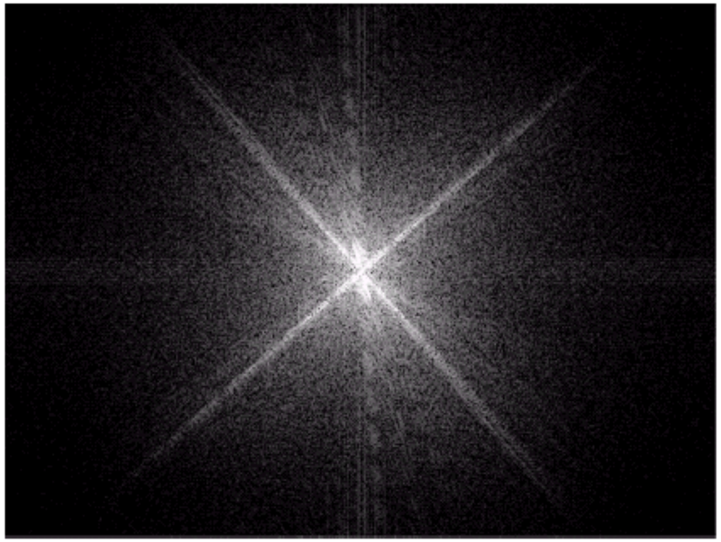



Map a narrow range of low-gray-level values in the input image into a wide range of output levels

Fourier Transform

- **Example**





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

Fourier Transform

- **Properties**
  - **Centering**

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

  - **Conjugate symmetry**

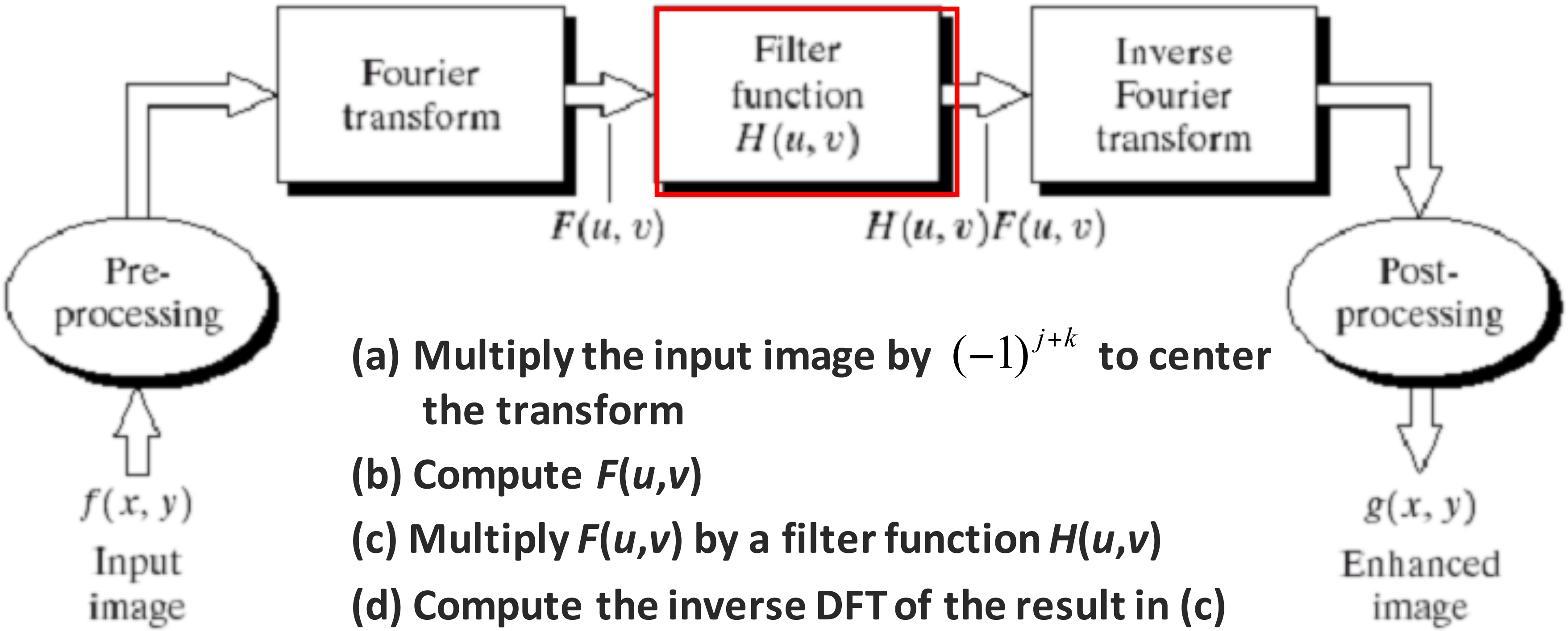
If $f(j,k)$ **is real,**

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

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

Frequency-Domain Filtering

- **Filtering in the frequency domain**



(a) Multiply the input image by $(-1)^{j+k}$ to center the transform

(b) Compute *F*(*u*,*v*)

(c) Multiply *F*(*u*,*v*) by a filter function *H*(*u*,*v*)

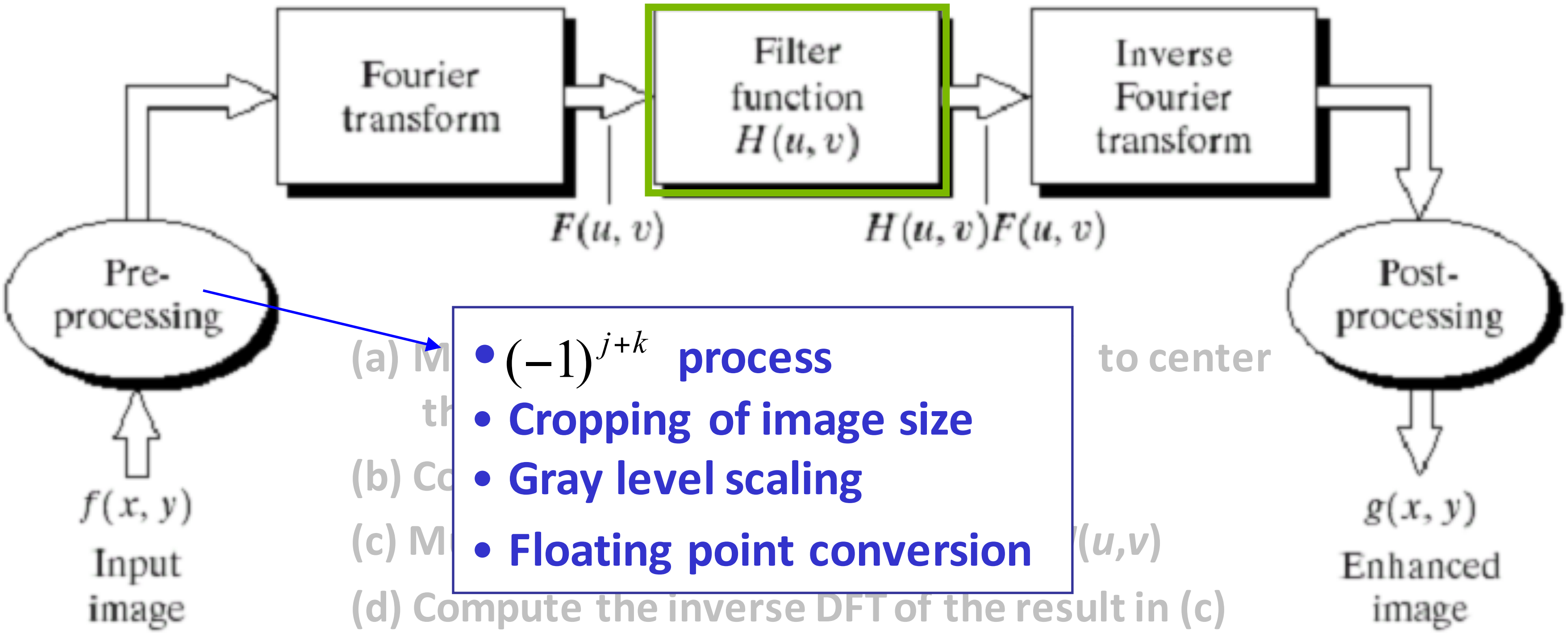
(d) Compute the inverse DFT of the result in (c)

(e) Obtain the real part of (d)

(f) Multiply the result in (e) by $(-1)^{j+k}$

Frequency-Domain Filtering

- Filtering in the frequency domain

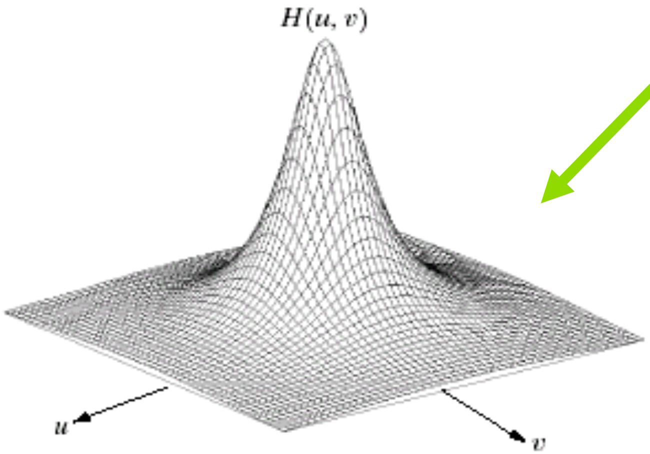


- $(-1)^{j+k}$ process
- Cropping of image size
- Gray level scaling
- Floating point conversion

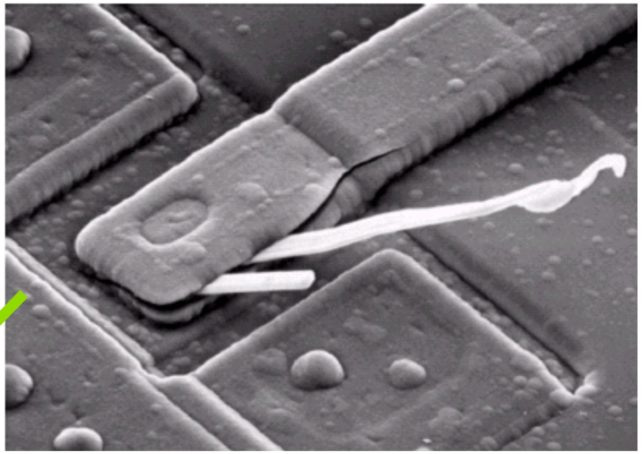
(a) M… to center
th…
(b) Co…
(c) M… (u,v)
(d) Compute the inverse DFT of the result in (c)
(e) Obtain the real part of (d)
(f) Multiply the result in (e) by $(-1)^{j+k}$

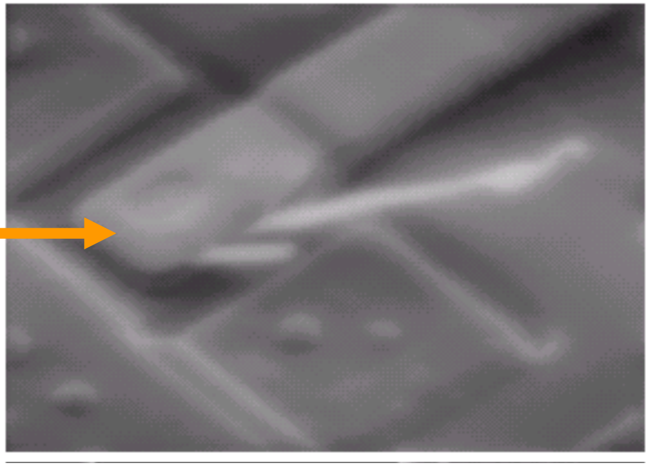
Frequency-Domain Filtering

- Basic filters and their properties
  - Low-pass filter
  - High-pass filter
  - Notch filter



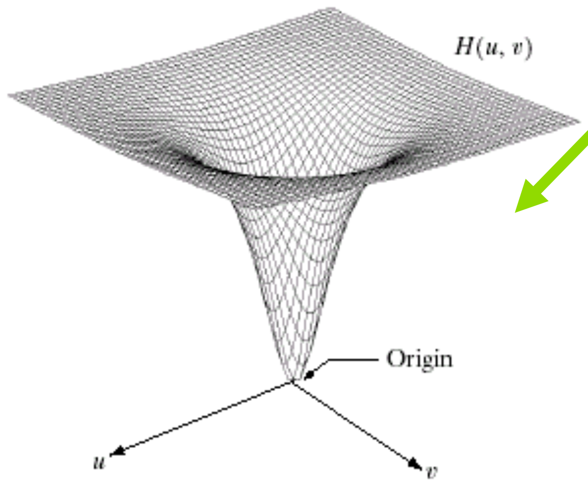
- Attenuates high frequencies
- Passes low frequencies



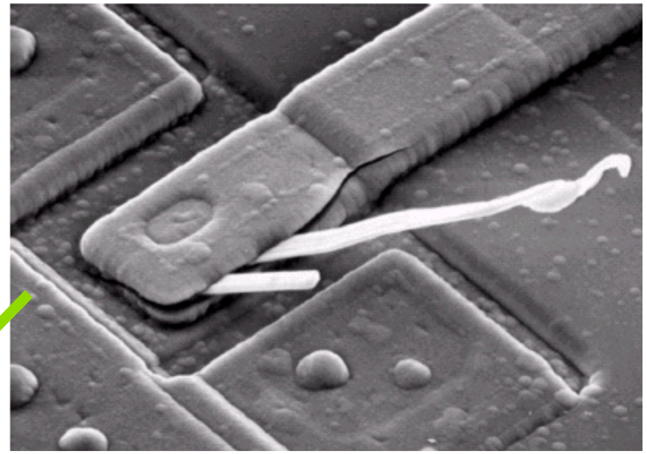


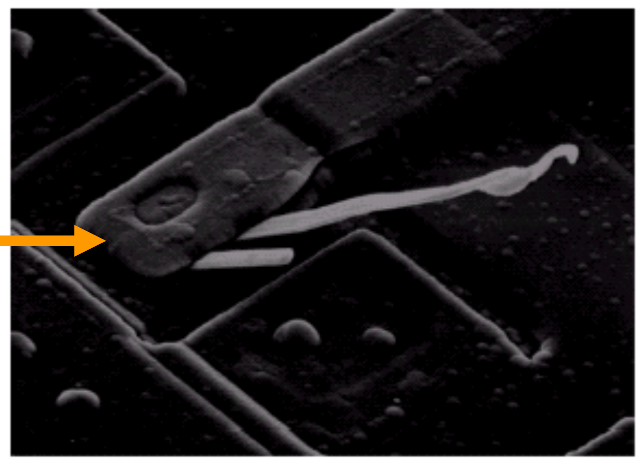
Frequency-Domain Filtering

- **Basic filters and their properties**
  - Low-pass filter
  - **High-pass filter**
  - Notch filter



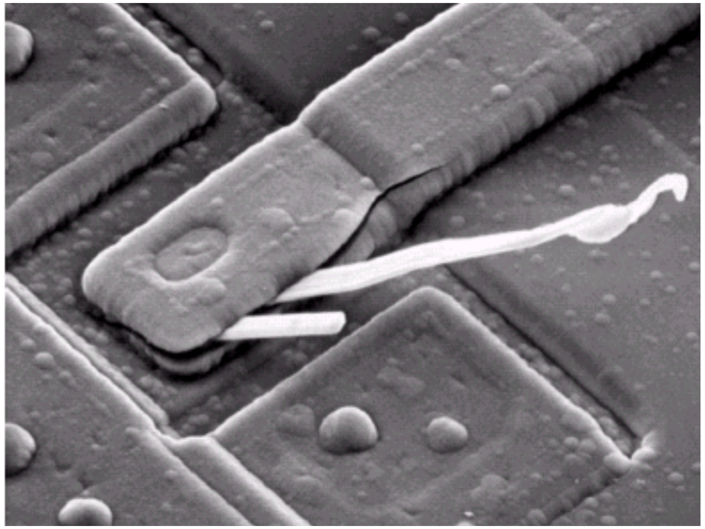
- Attenuates low frequencies
- Passes high frequencies



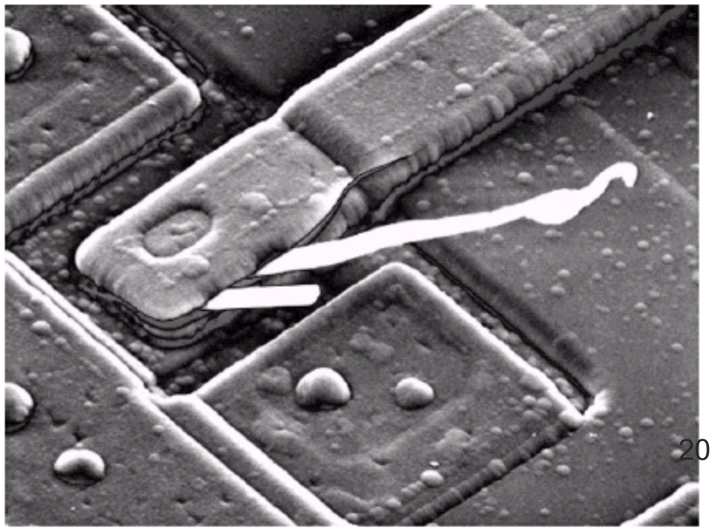


Frequency-Domain Filtering

- 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

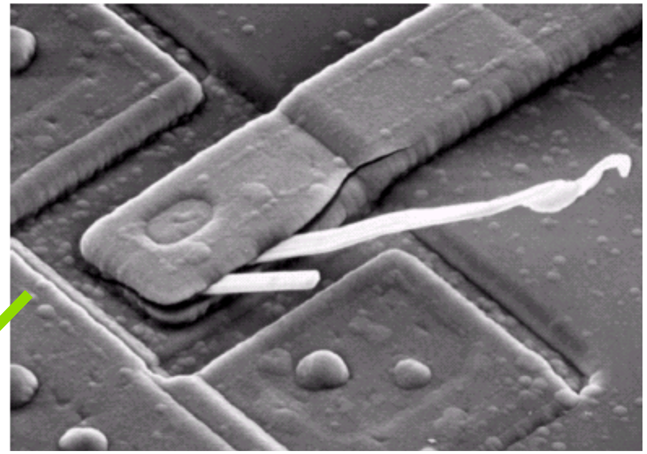


Frequency-Domain Filtering

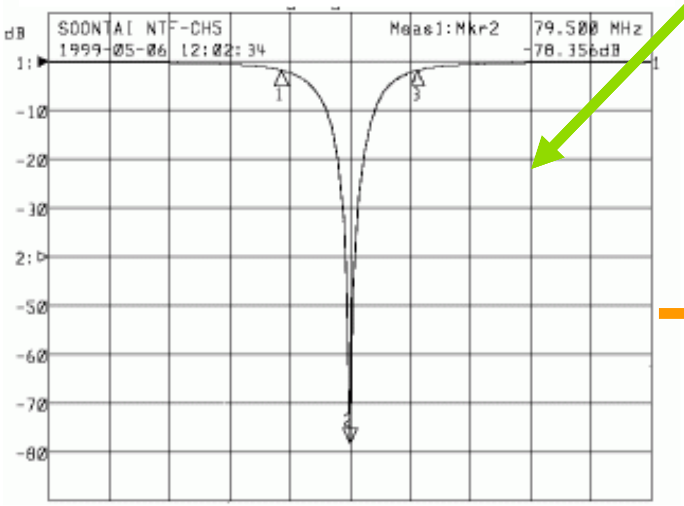
- **Basic filters and their properties**
  - **Notch filter**

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

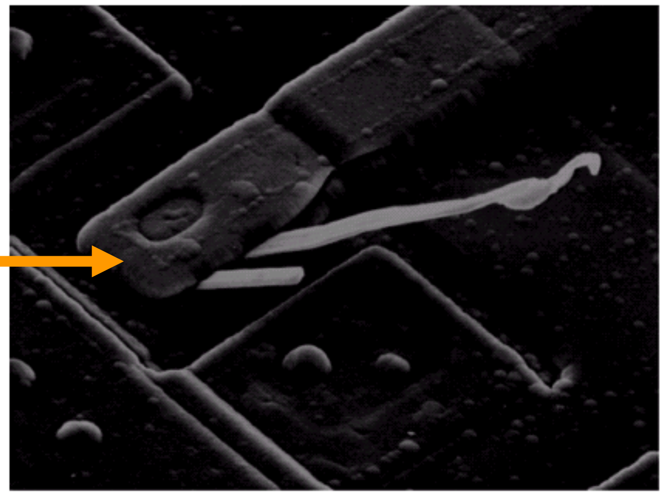
→ A constant function with a hole at the origin



1D notch filter



http://www.soontai.com/da_filterprofile.html



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**

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

$$f(j,k)\underbrace{h(j,k)}_{\text{Impulse function}}\Leftrightarrow F(u,v)*\underbrace{H(u,v)}_{\text{Impulse response}}$$

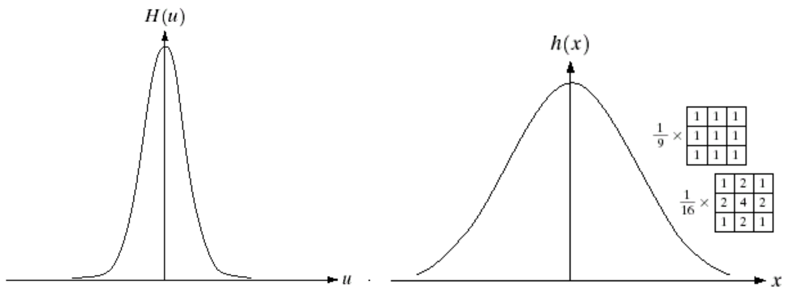
Convolution Theorem

Frequency-Domain Filtering

- Gaussian filter
  - One-dimensional

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

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

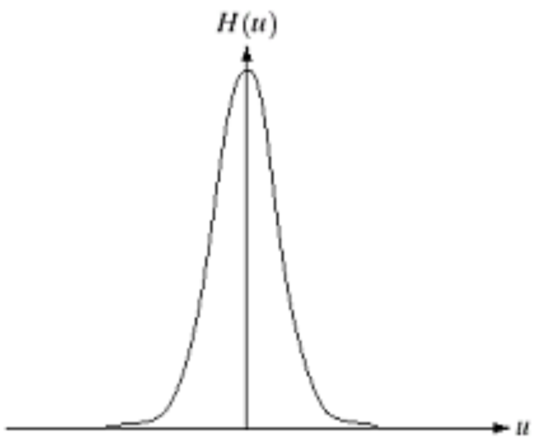


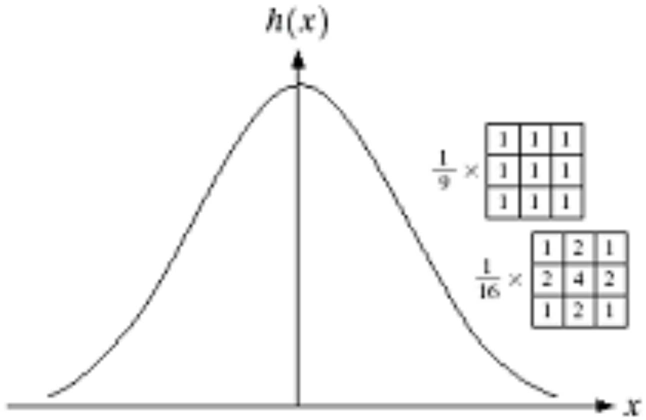
  - Important characteristics
    - Easy to specify and manipulate
    - Fourier transform of a Gaussian function is a 'real' Gaussian function
    - Two functions behave reciprocally w.r.t one another

Frequency-Domain Filtering

- **Gaussian filter**
  - **Low-pass filter**

$$H(u) = Ae^{-u^2/2\sigma^2} \qquad h(k) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2 k^2}$$





→All the values are positive in both domains

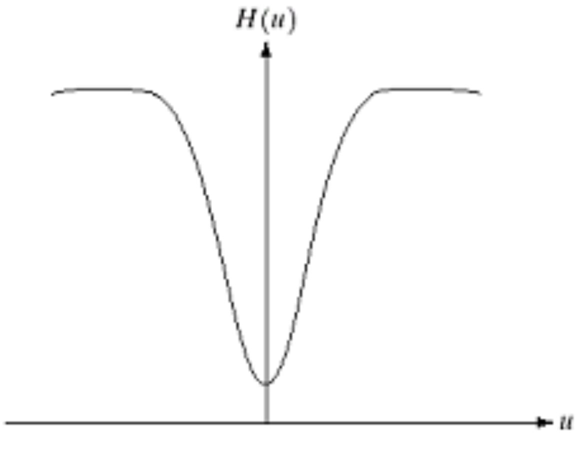
→The narrower the frequency domain filter, the wider the spatial domain filter i.e. more severe blurring effect

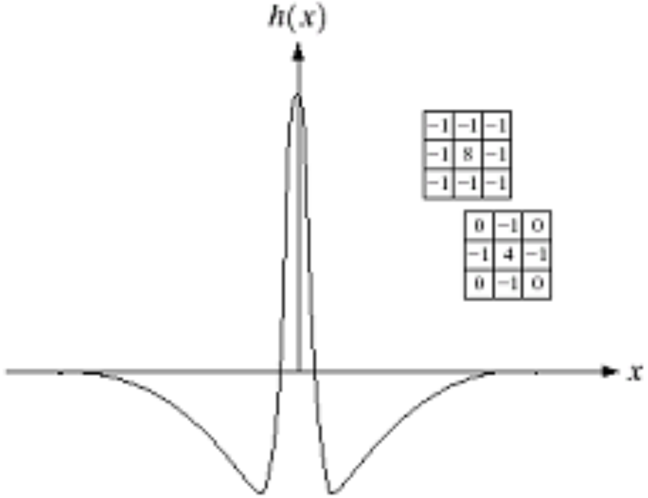
Frequency-Domain Filtering

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

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

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





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

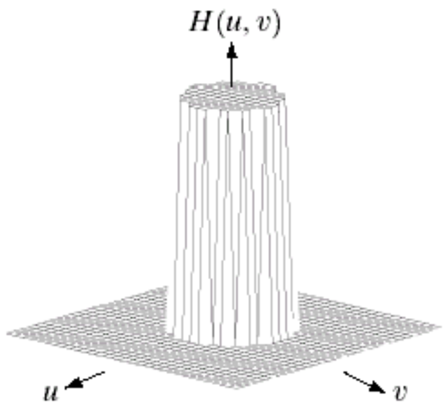
- **Ideal low-pass filters**

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

Radius:
Non-negative

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

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



Perspective plot

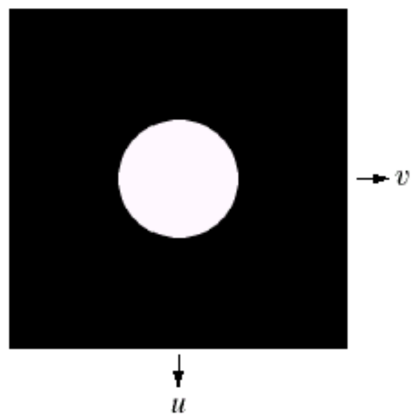
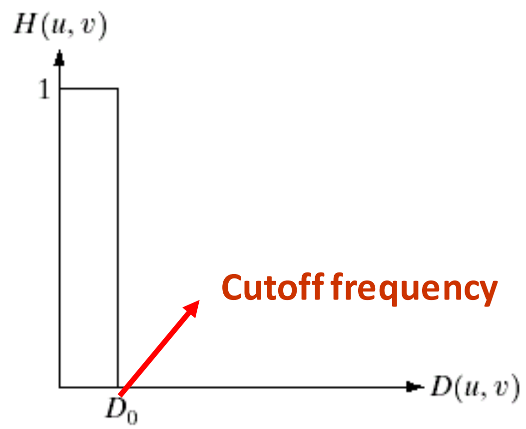


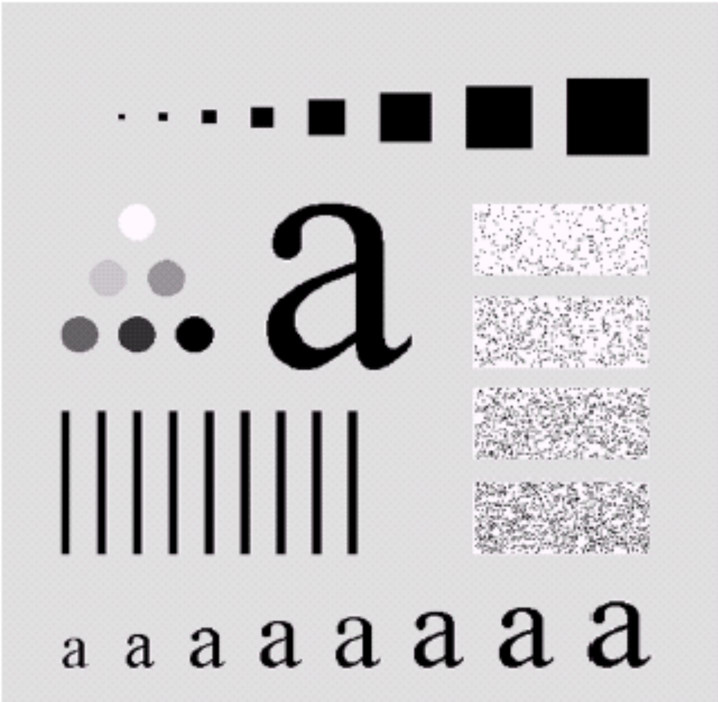
Image display



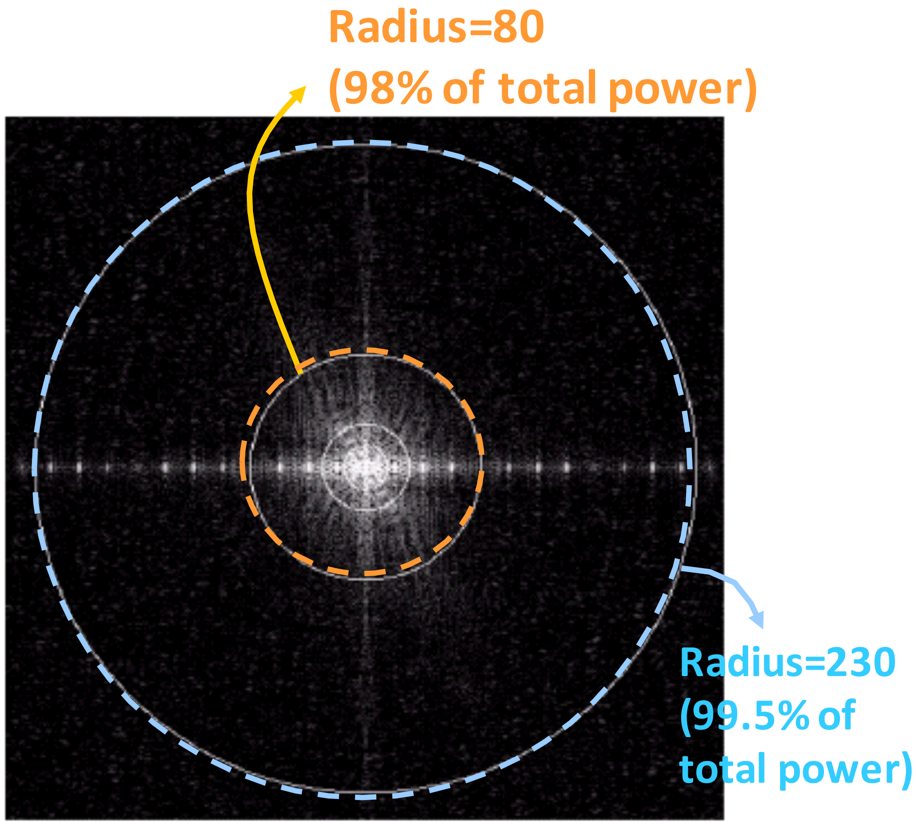
Radial cross section

Frequency-Domain Filtering

- Ideal low-pass filters
  - Example



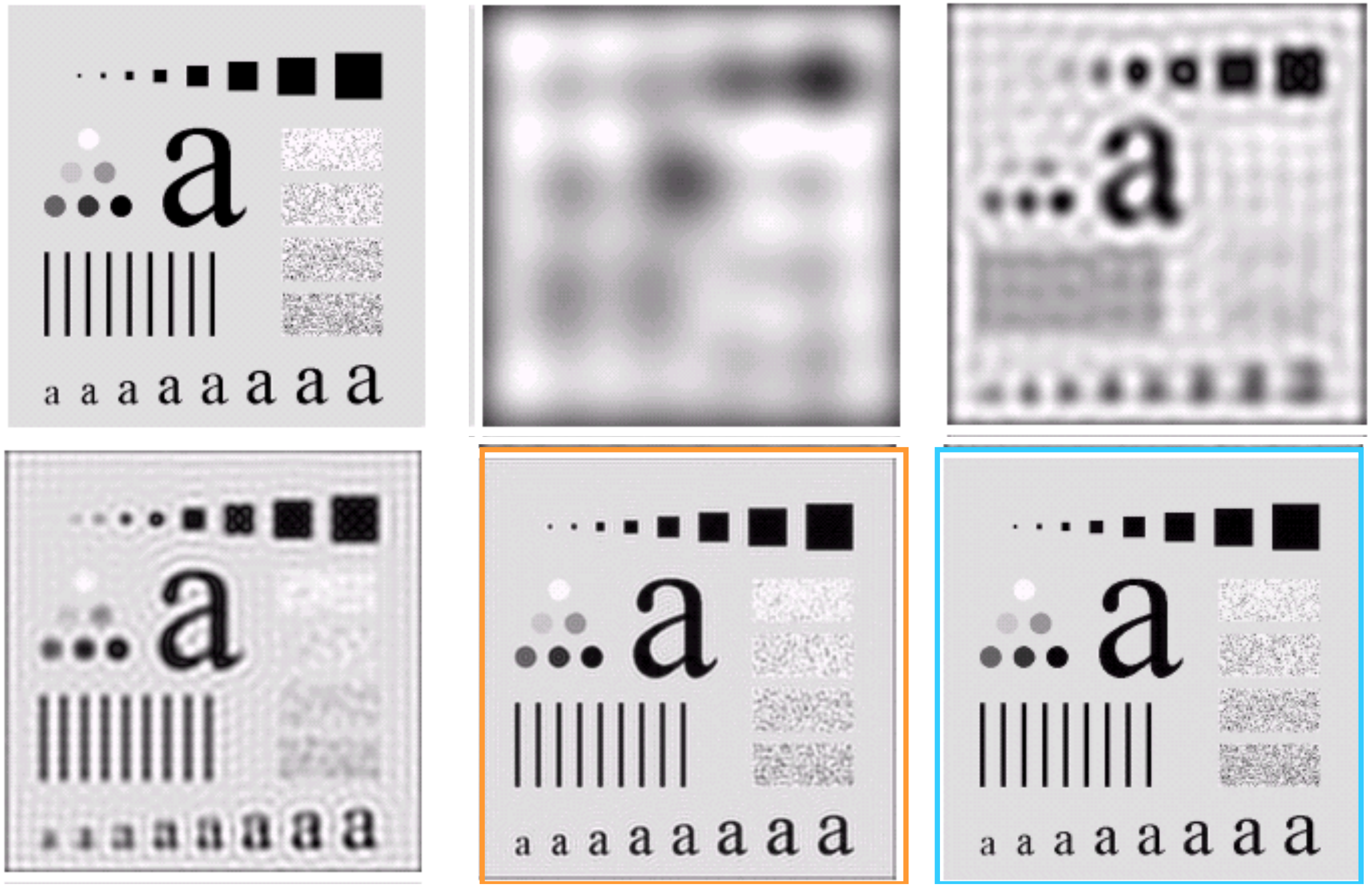
Original image
(in spatial domain)



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

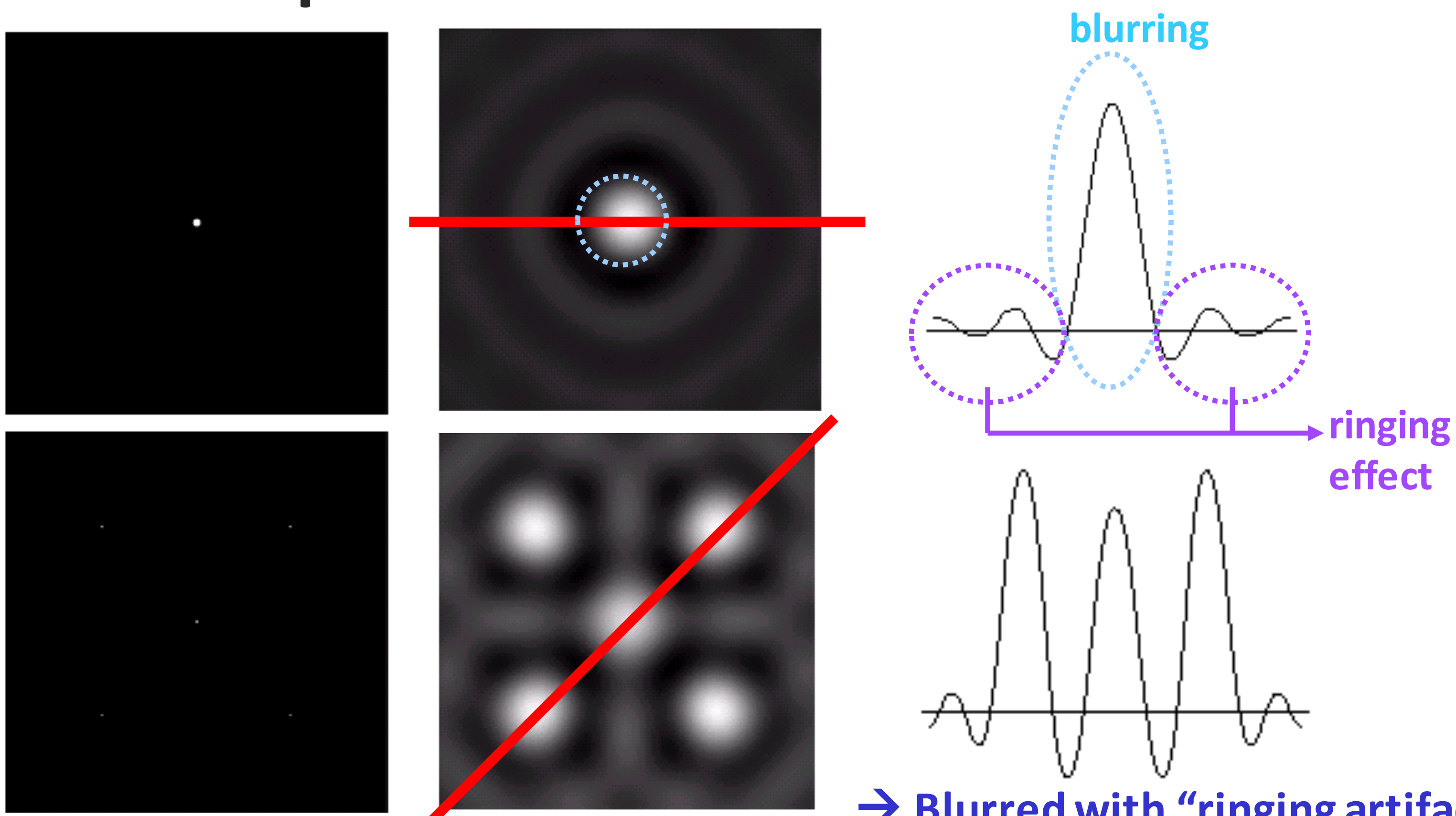
Frequency-Domain Filtering

- Ideal low-pass filters
  - Example (with radius = 5, 15, 30, 80 & 230)



Frequency-Domain Filtering

- Ideal low-pass filters
  - Example

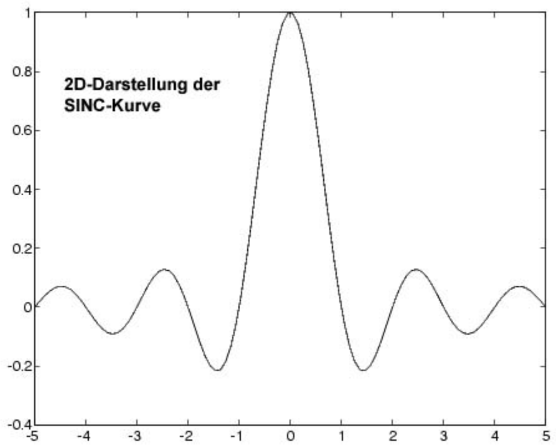


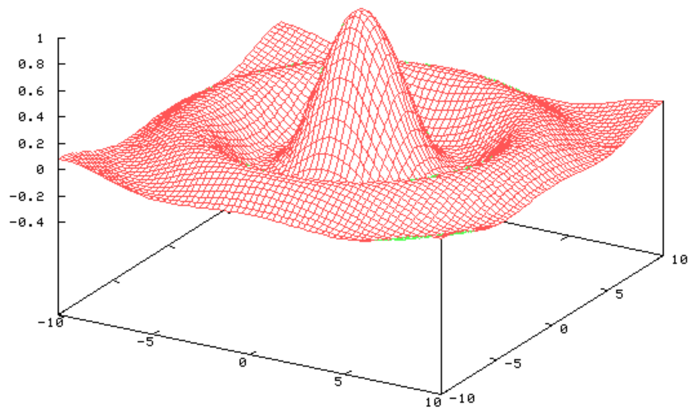
Frequency domain spatial domain

→ Blurred with "ringing artifacts"

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

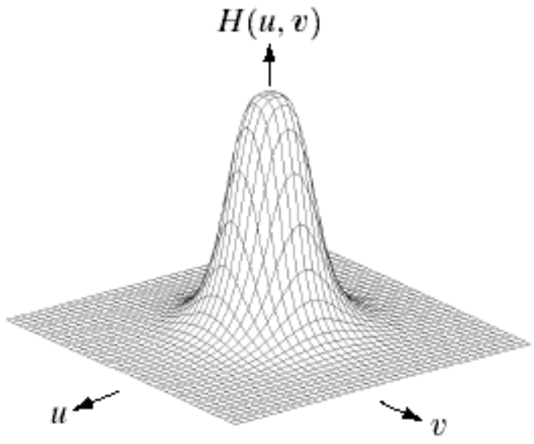
- **Butterworth low-pass filters**

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

n: order (must be an integer)

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

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



Perspective plot

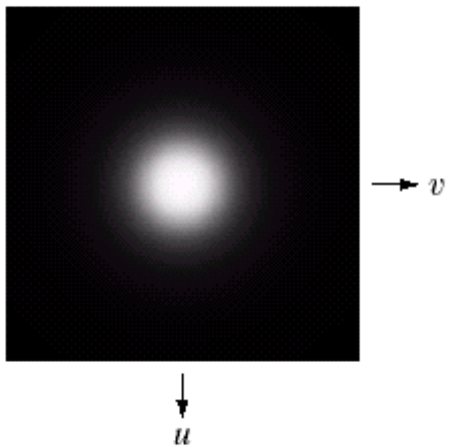
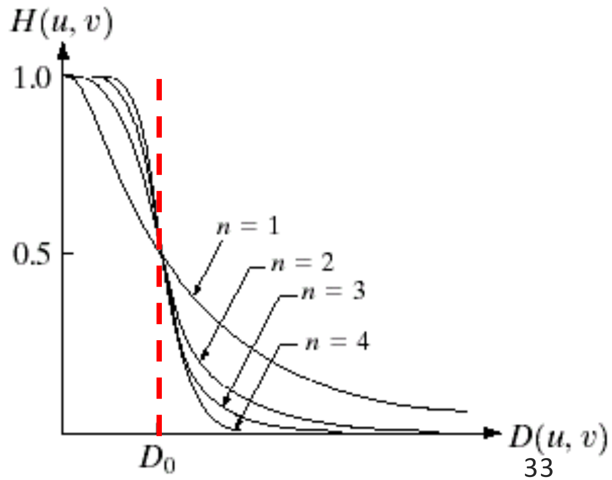


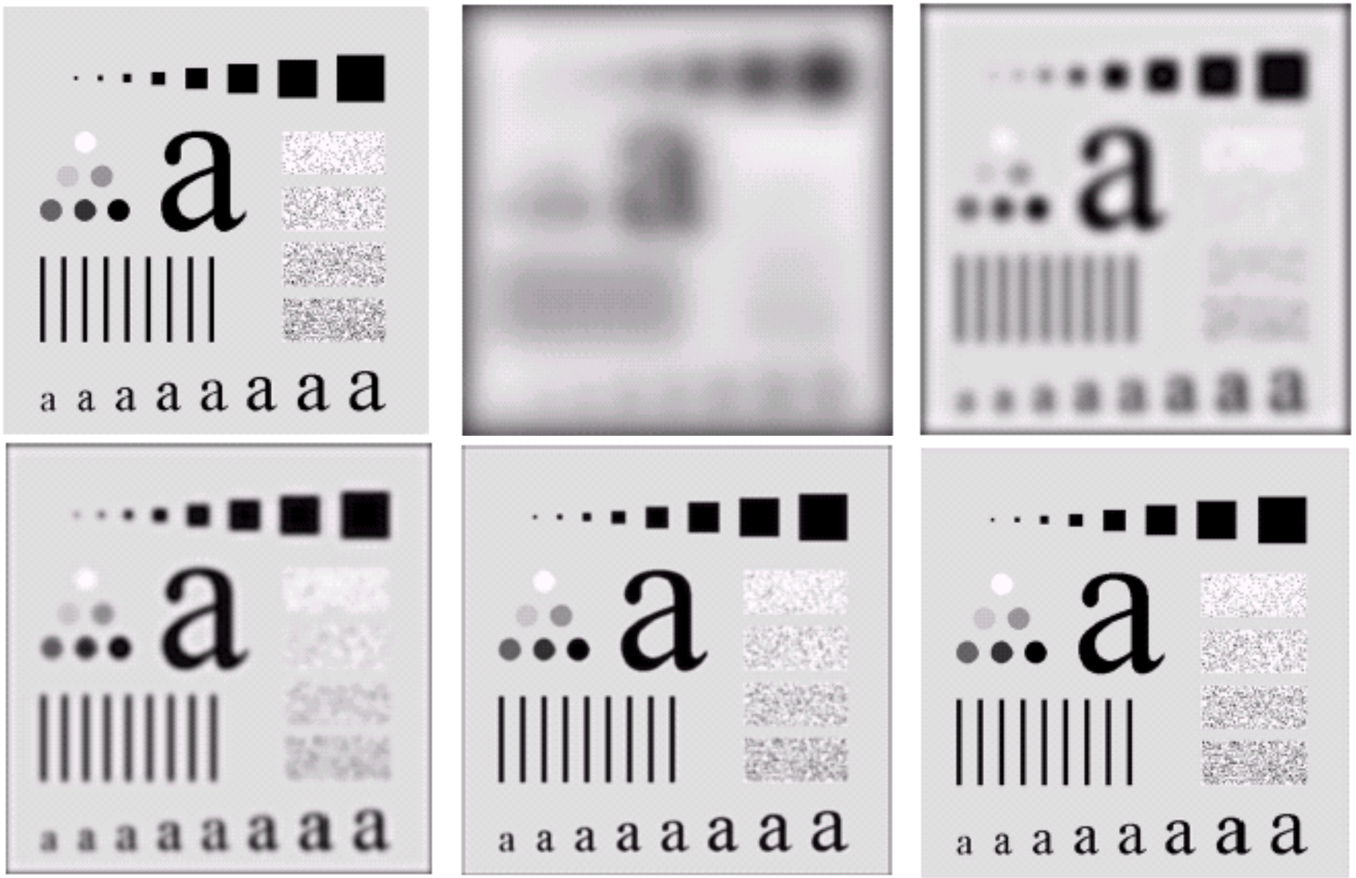
Image display



Radial cross section

Frequency-Domain Filtering

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

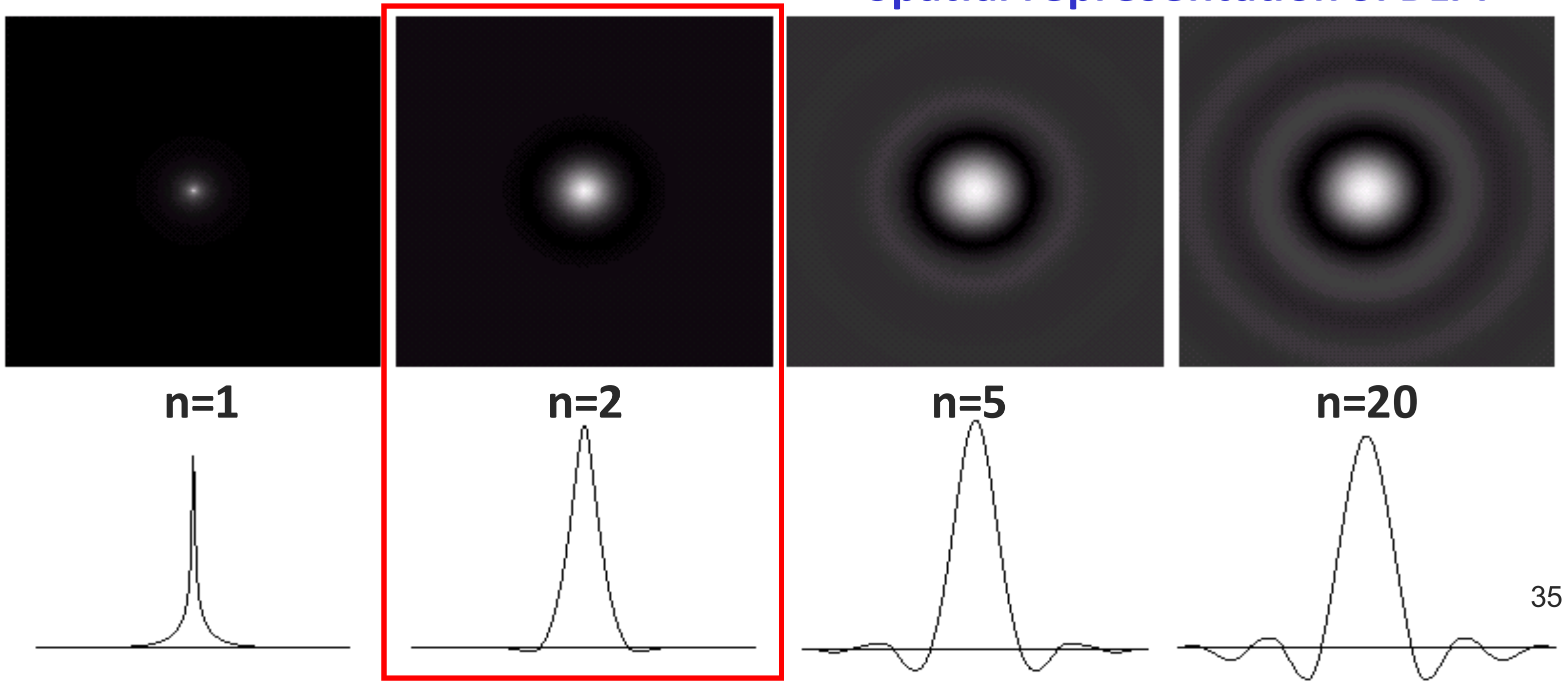


Frequency-Domain Filtering

- **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

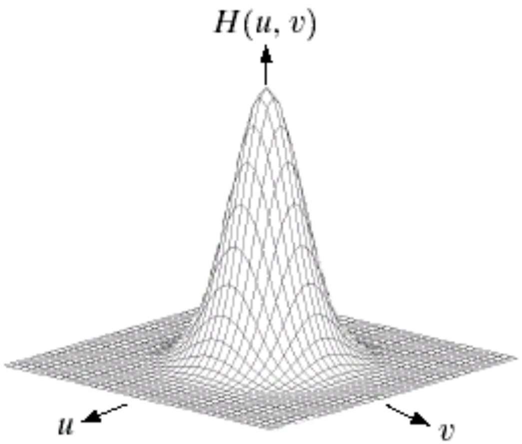


Frequency-Domain Filtering

- **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

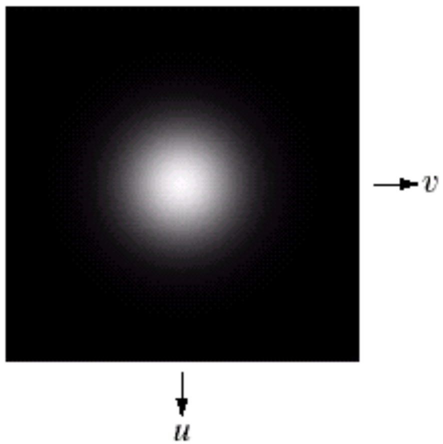
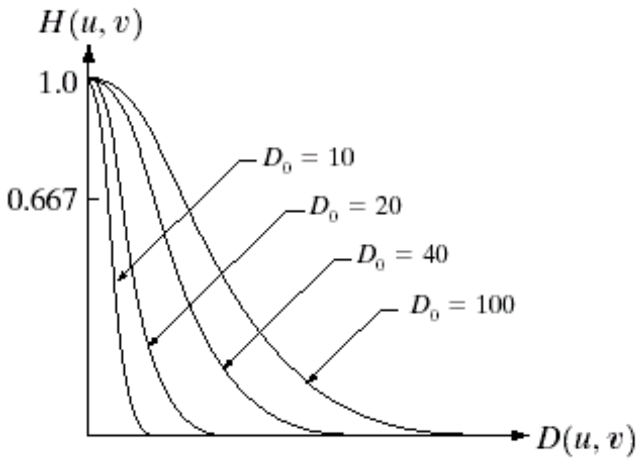


Image display



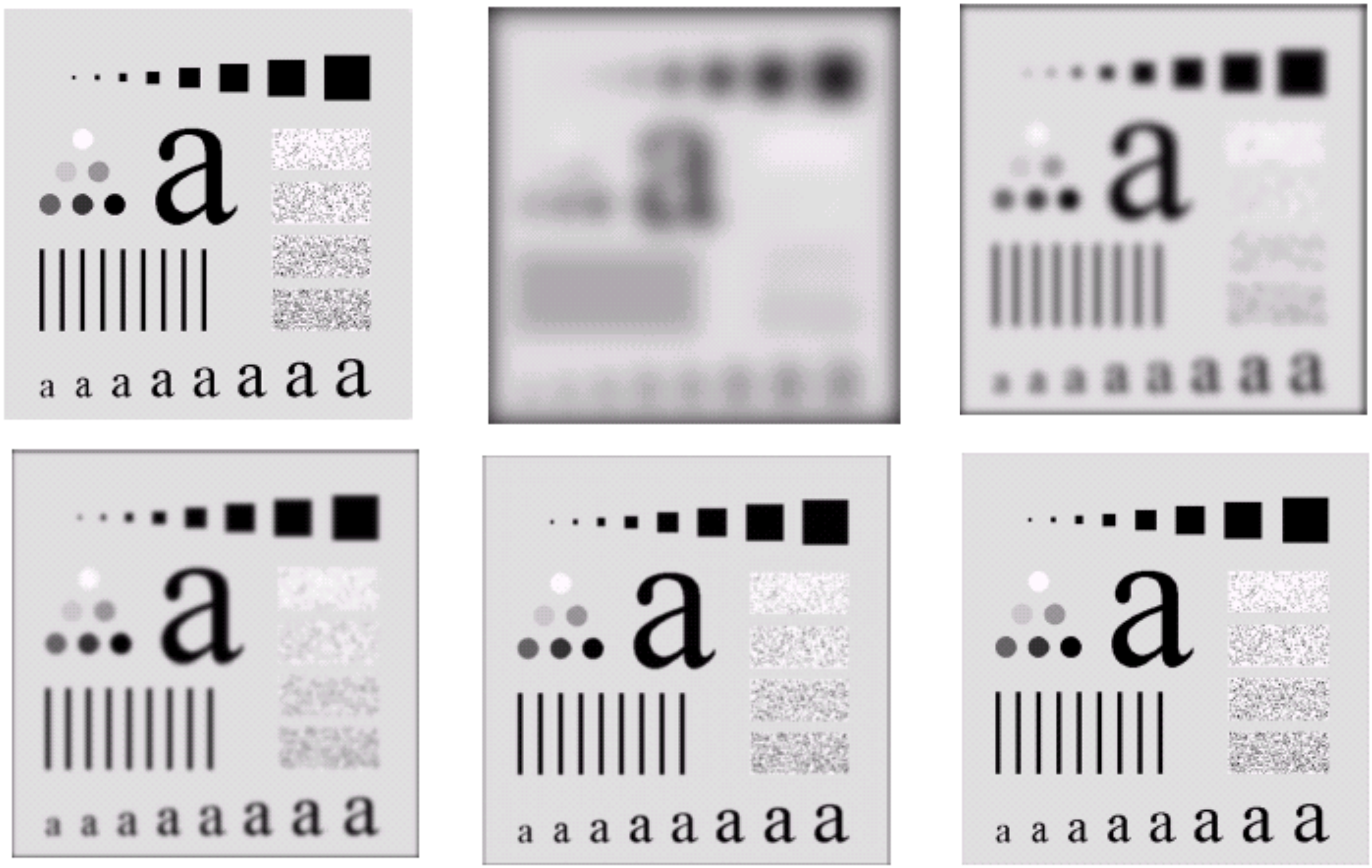
Radial cross section

→Gaussian in frequency domain → Gaussian in spatial domain

→ No ringing artifacts

Frequency-Domain Filtering

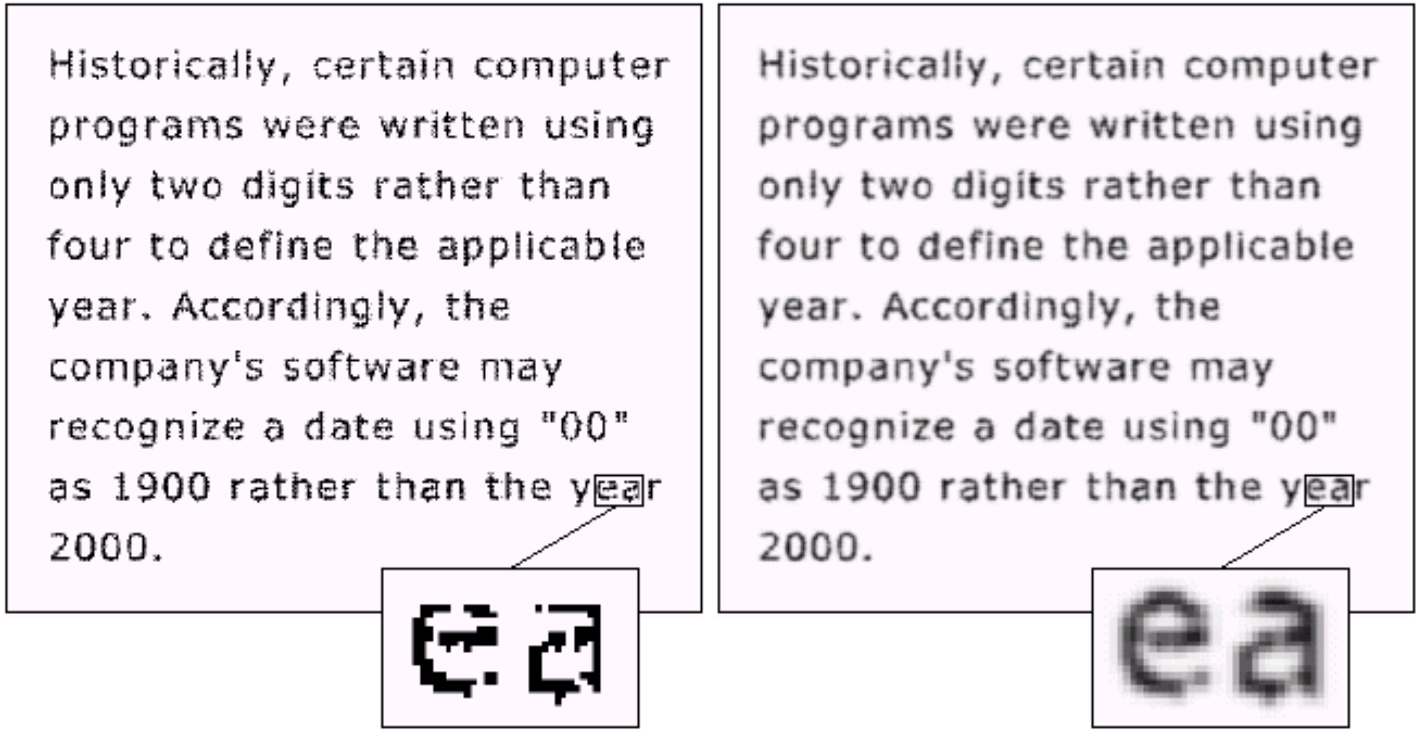
- **Gaussian low-pass filters**



“ No ringing artifacts “ “ achieve less smoothing than BLPF “

Frequency-Domain Filtering

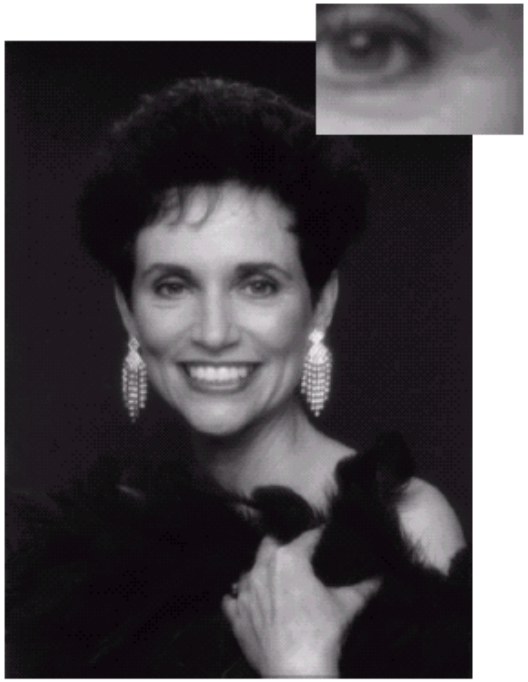
- Examples
  - Gaussian low-pass filters
    - Connect broken characters



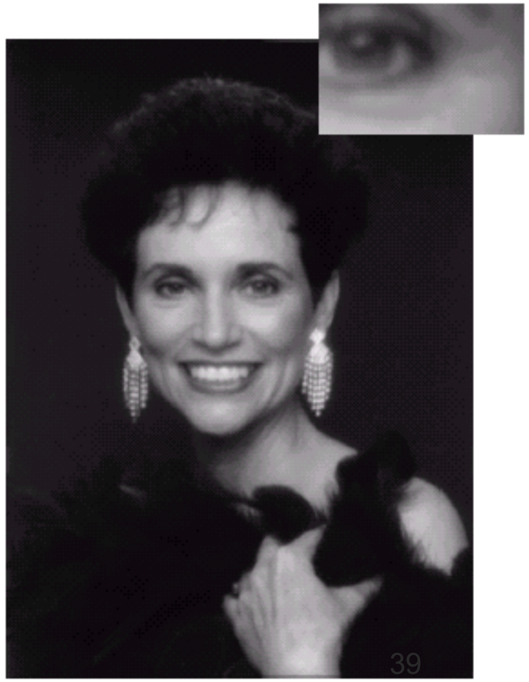
Frequency-Domain Filtering

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

Original



Cutoff=100



Cutoff=80

Frequency-Domain Filtering

- **Sharpening frequency-domain filters**

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

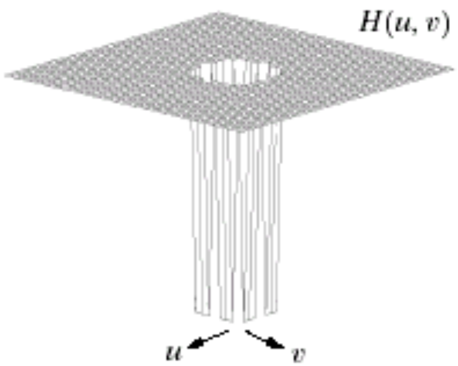
  - **Ideal high-pass filters**
  - **Butterworth high-pass filters**
  - **Gaussian high-pass filters**
  - **The Laplacian in the frequency domain**

Frequency-Domain Filtering

- **Ideal high-pass filters**

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

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



Perspective plot

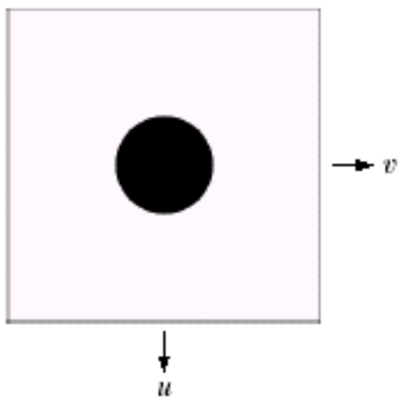
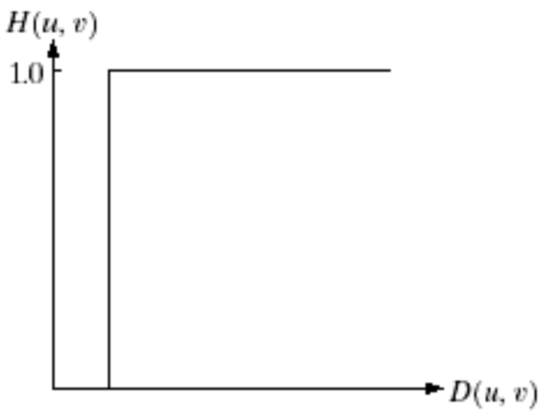
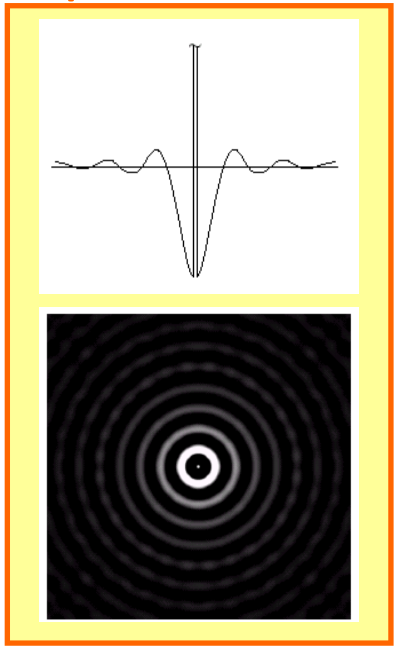


Image display



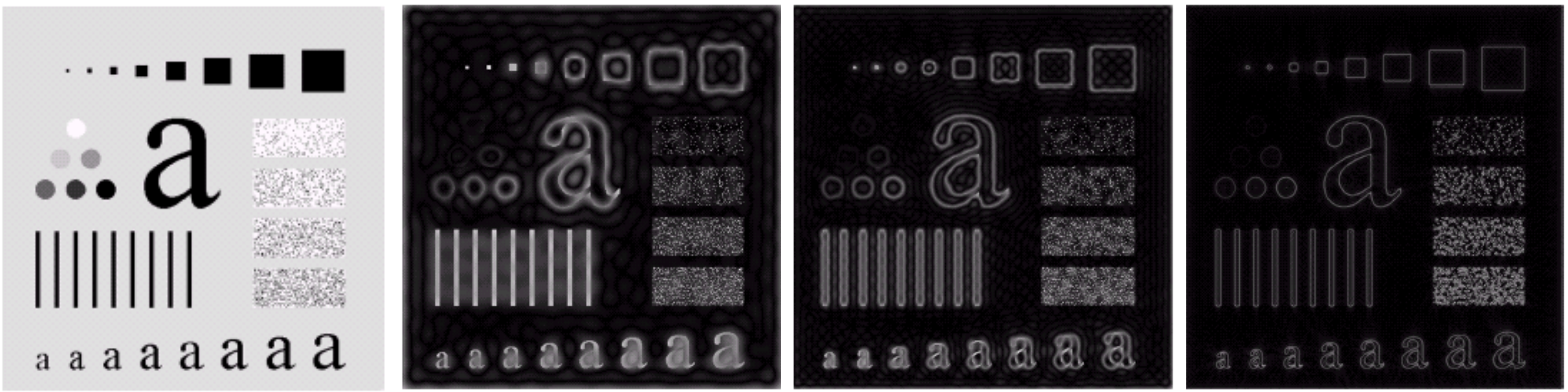
Radial cross section

Spatial domain



Frequency-Domain Filtering

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



Original | Cutoff=15 | Cutoff=30 | Cutoff=80

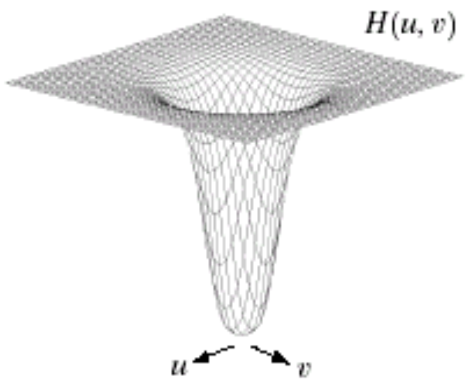
Frequency-Domain Filtering

- **Butterworth high-pass filters**

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

n: order (must be an integer)

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



Perspective plot

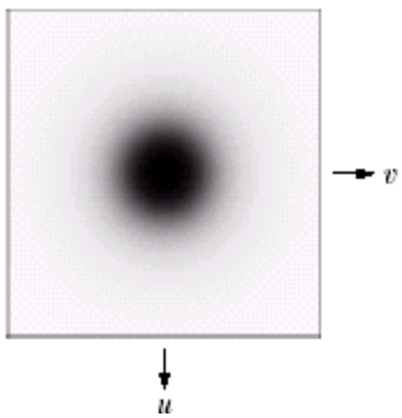
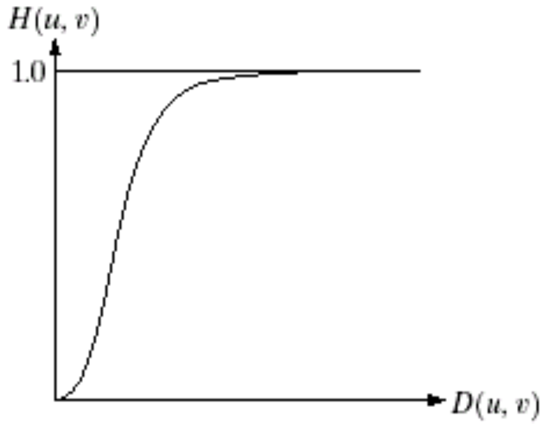
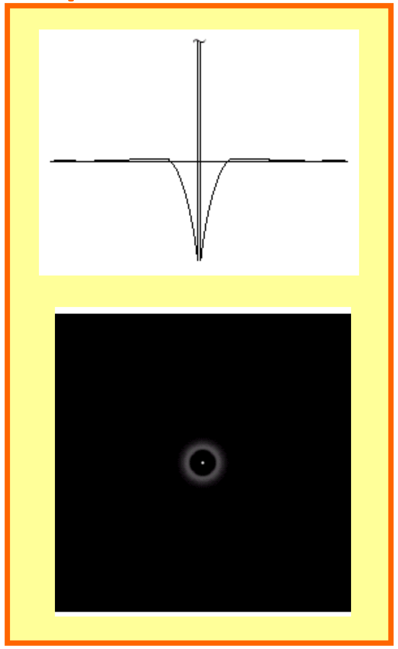


Image display



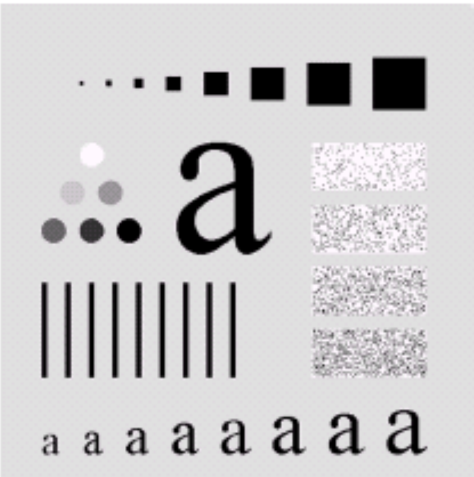
Radial cross section

Spatial domain

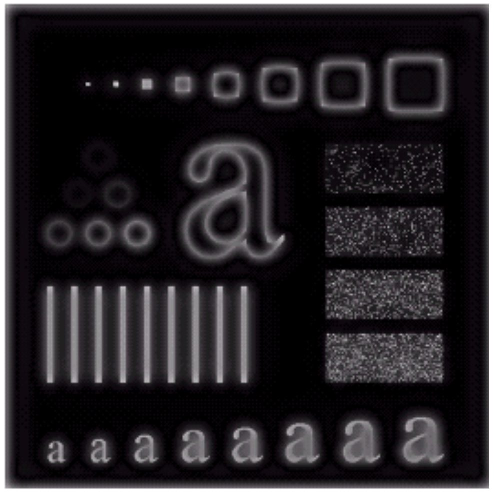


Frequency-Domain Filtering

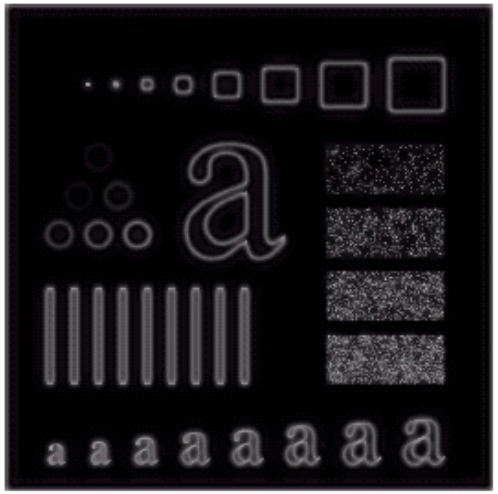
- Butterworth high-pass filters (n=2)
  - Example (with radius = 15, 30 & 80)



Original



Cutoff=15



Cutoff=30

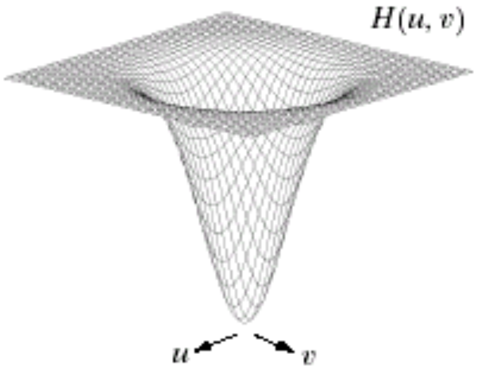
Cutoff=80

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

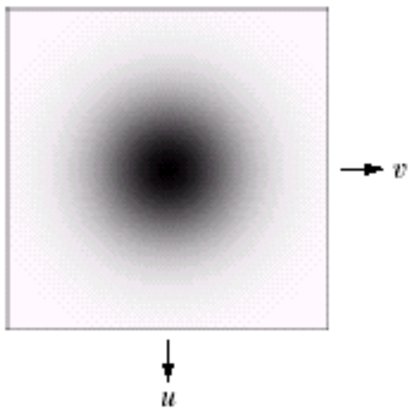
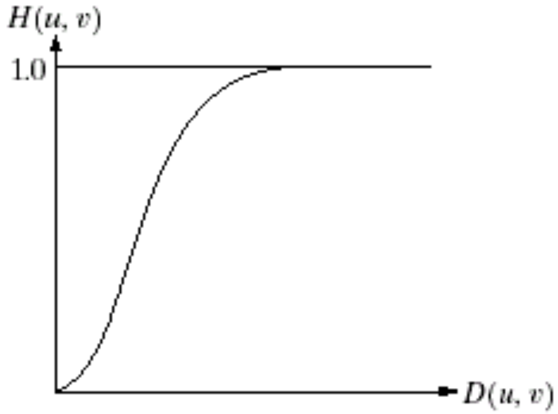
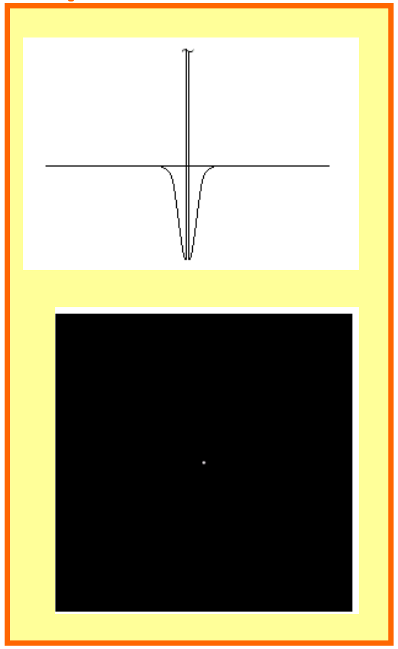


Image display



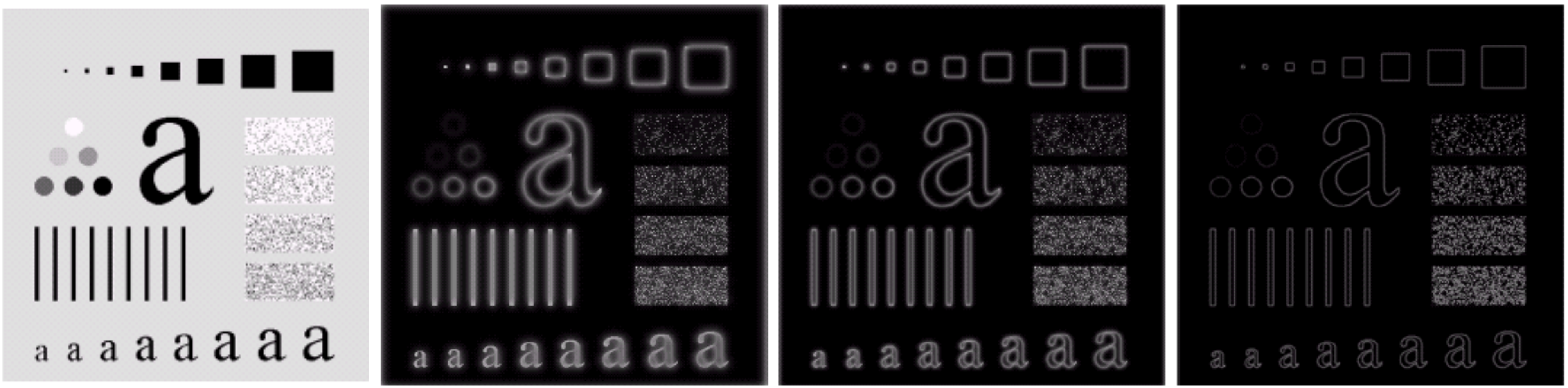
Radial cross section

Spatial domain



Frequency-Domain Filtering

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



Original | Cutoff=15 | Cutoff=30 | Cutoff=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**

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

$$\Im\left[\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}\right] = (ju)^2 F(u,v) + (jv)^2 F(u,v)$$

$$= -(u^2 + v^2) F(u,v)$$

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

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

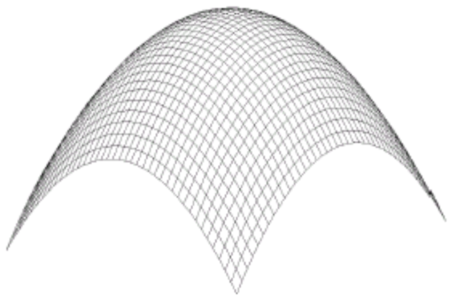
Center of the filter needs to be shifted

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

Frequency-Domain Filtering

- **The Laplacian in the Frequency domain**

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



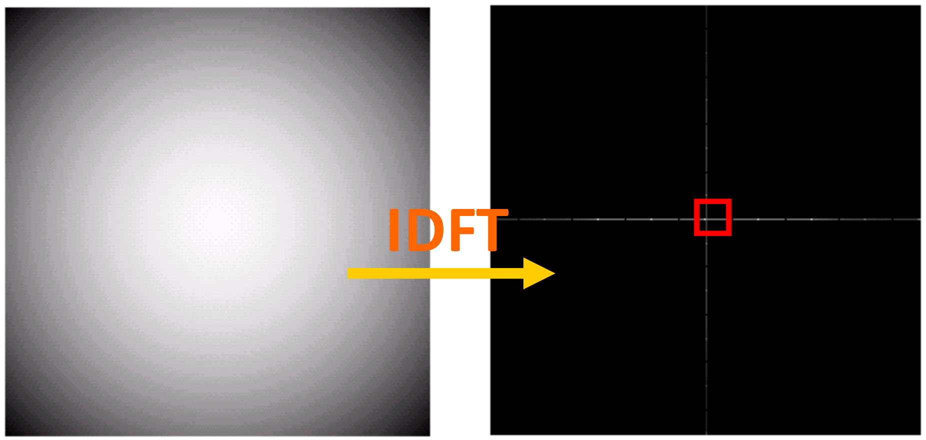
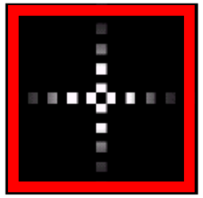


Image display

Spatial domain



0	1	0
1	-4	1
0	1	0

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

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

$$\nabla^2 f = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

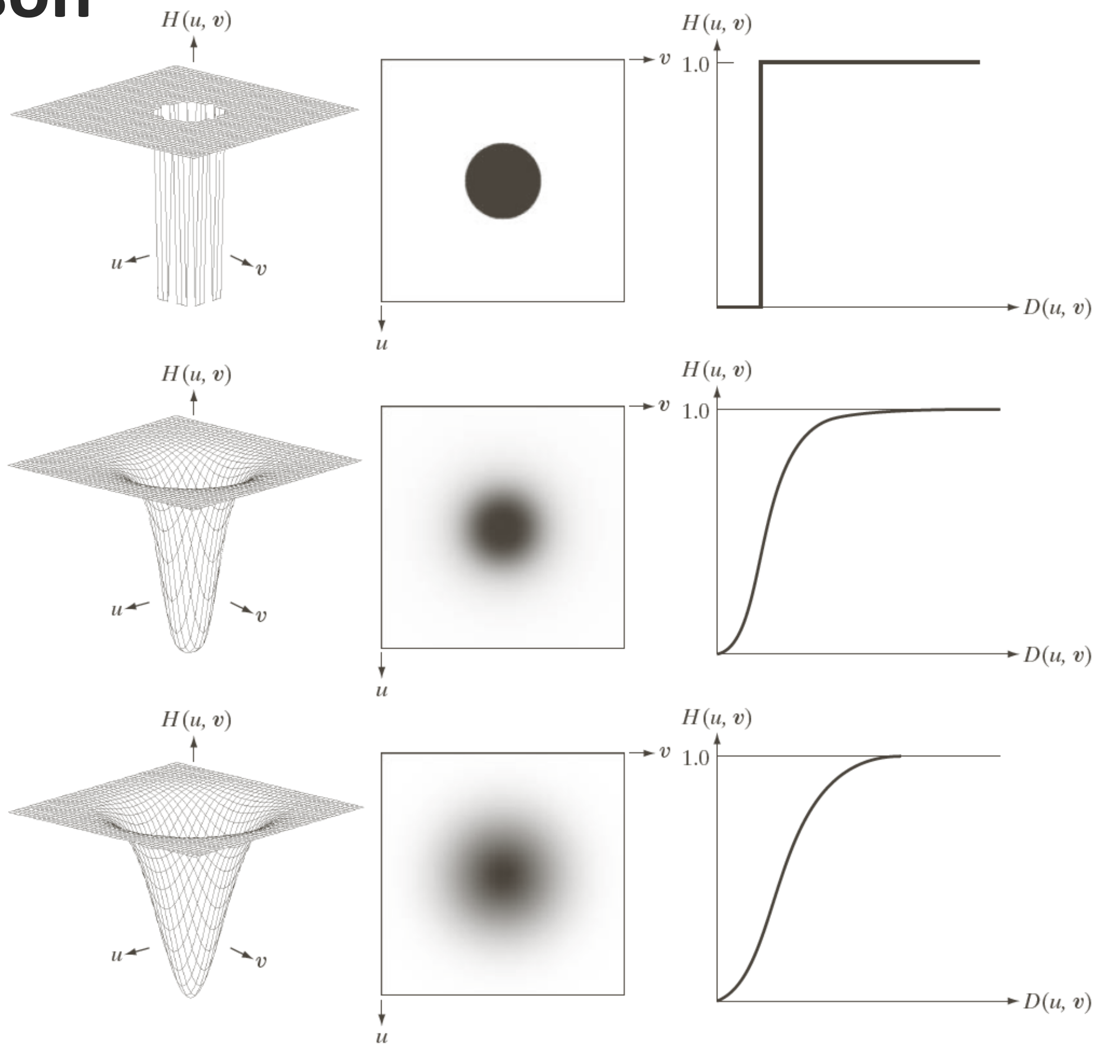
Frequency-Domain Filtering

- **Comparison**

Ideal

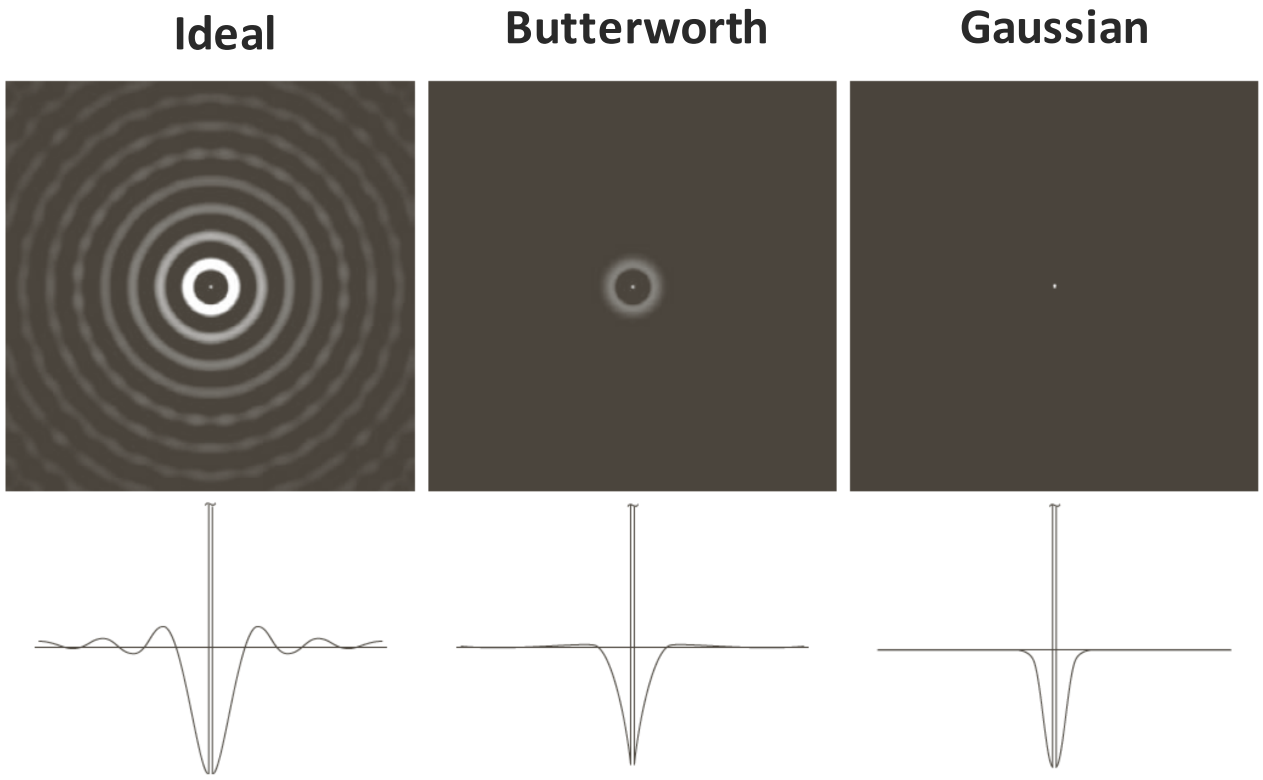
Butterworth

Gaussian



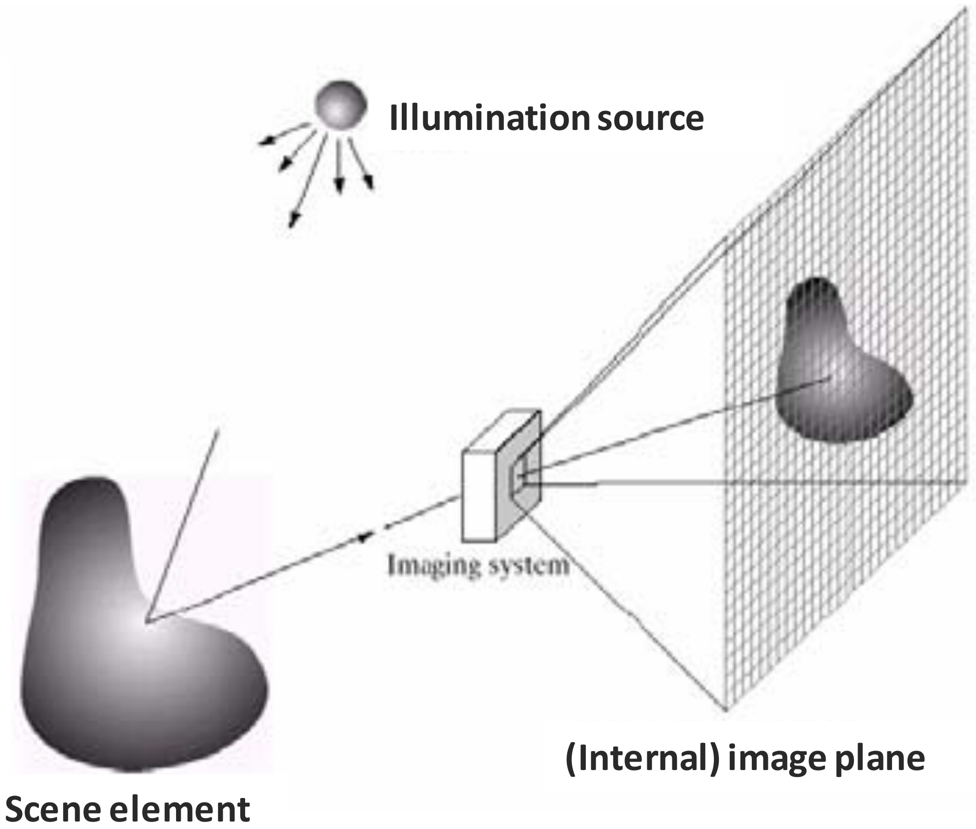
Frequency-Domain Filtering

- **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**

$$\boxed{\Im\{f(x,y)\} \neq \Im\{i(x,y)\}\Im\{r(x,y)\}}$$

  - **Define**

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

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

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

Frequency-Domain Filtering

- **Homomorphic filtering**

$$
\begin{aligned}
S(u,v) &= H(u,v)Z(u,v) \\
&= H(u,v)F_i(u,v) + H(u,v)F_r(u,v) \\
\Rightarrow s(x,y) &= \Im^{-1}\{S(u,v)\} \\
&= \Im^{-1}\{H(u,v)F_i(u,v)\} + \Im^{-1}\{H(u,v)F_r(u,v)\}
\end{aligned}
$$

s(x,y) spatial domain

- let

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

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

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

Frequency-Domain Filtering

- **Homomorphic filtering**

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

$$g(x,y) = \boxed{e^{s(x,y)}}$$

$$= e^{i'(x,y)} \cdot e^{r'(x,y)} = \boxed{i_0(x,y) r_0(x,y)}$$

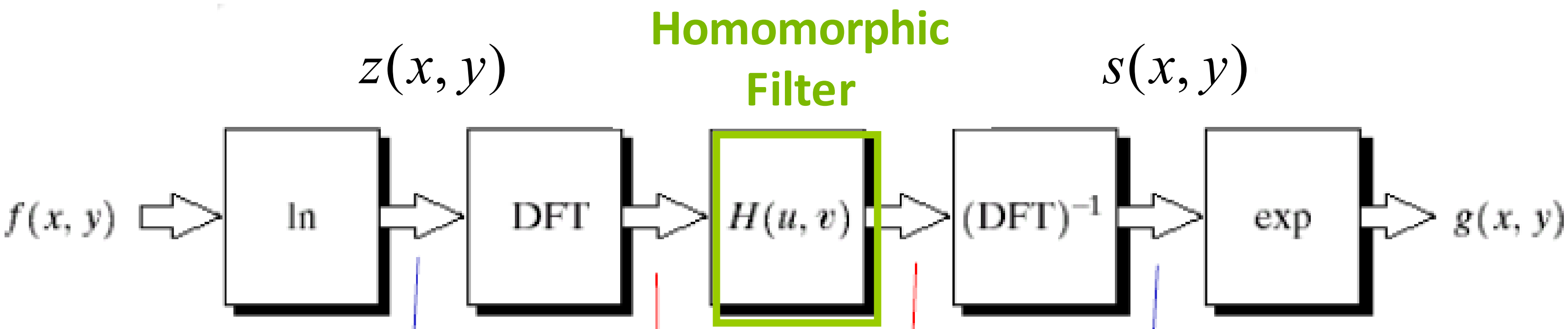
where

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

are the illumination and reflectance components of the output image

Frequency-Domain Filtering

- **Homomorphic filtering**



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

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

$$\boxed{Z(u, v) = F_i(u, v) + F_r(u, v)}$$

$$\boxed{H(u, v)F_i(u, v) + H(u, v)F_r(u, v)}$$

filtered image in frequency domain.

$$\boxed{s(x, y) = i'(x, y) + r'(x, y)}$$

$$g(x, y) = i_0(x, y)r_0(x, y)$$

$$i_0(x, y) = e^{i'(x, y)}$$

$$r_0(x, y) = e^{r'(x, y)}$$

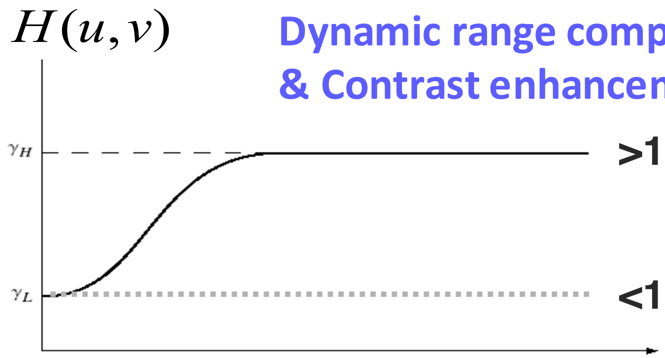
Frequency-Domain Filtering

- 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



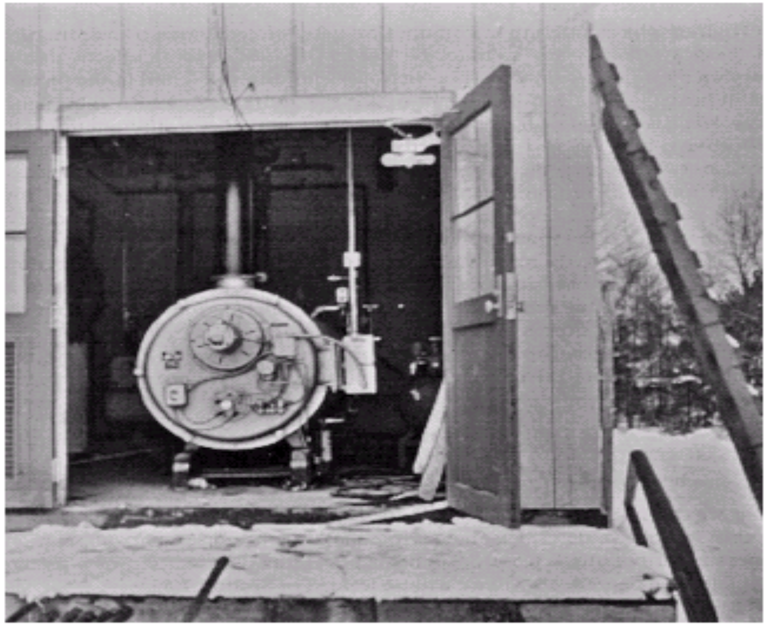
Dynamic range compression & Contrast enhancement

rotate around vertical axis to form a 3D case. This filter is like a bowl.
This filter enhances high frequency(reflectance) and lowers the low frequency(illumination)

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

The distance from point (u,v) to the origin of the centered transform

If the illumination is too high, the dynamic range is also very high, but the sensor has no such high dynamic range capability, so the darker area gets darkest.



Frequency-Domain Filtering

- Another example of homomorphic filtering
  - PET image

edge is more clear.
high frequency is enhanced.
dynamic range is compressed.
more details are preserved.