Filtering in the Frequency Domain

李东晓

lidx@zju.edu.cn

Contents

- Jean Baptiste Joseph Fourier
- The Fourier series & the Fourier transform
- Image filtering in the frequency domain
  - – Image smoothing
  - – Image sharpening
- Fast Fourier Transform

Jean Baptiste Joseph Fourier

Fourier was born in Auxerre, France in 1768 (250+ years ago)

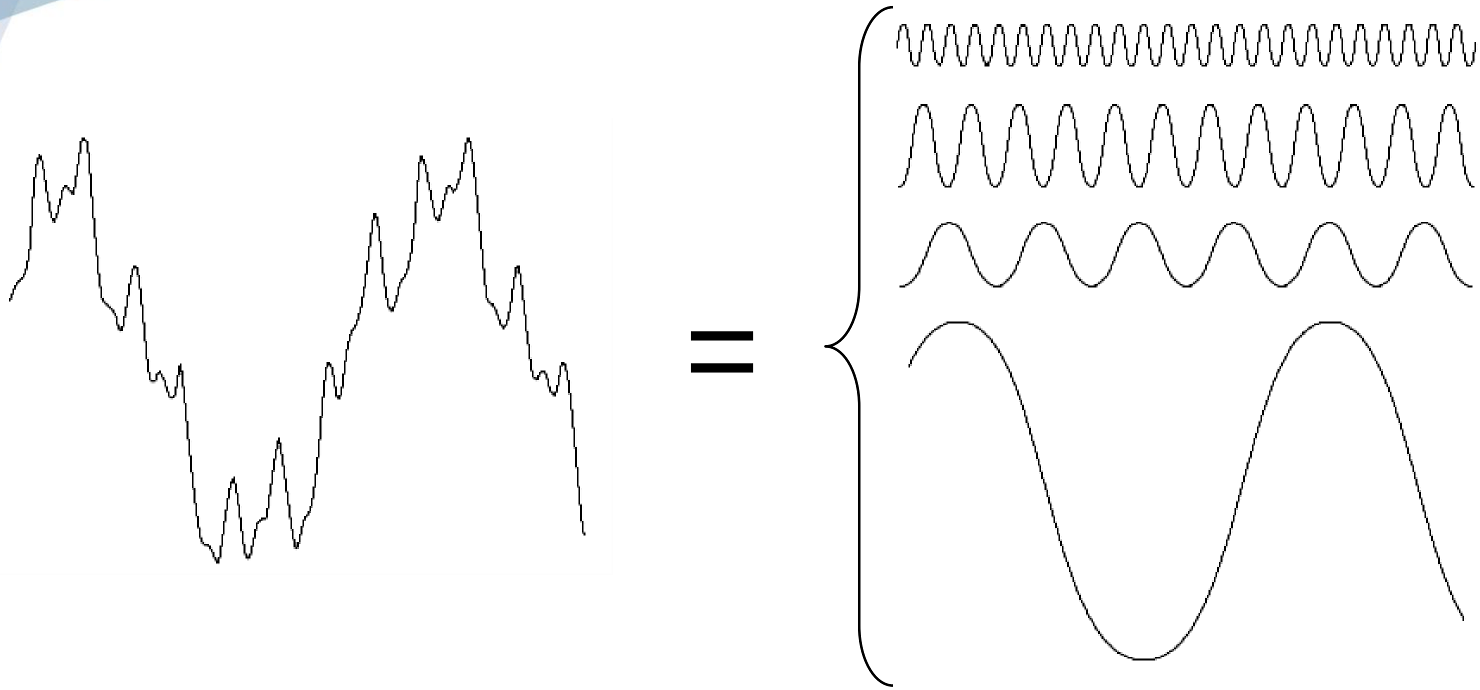
- Most famous for his work "*La Théorie Analitique de la Chaleur*" published in 1822
- Translated into English in 1878: "*The Analytic Theory of Heat*"

  热的解析理论

Nobody paid much attention when the work was first published

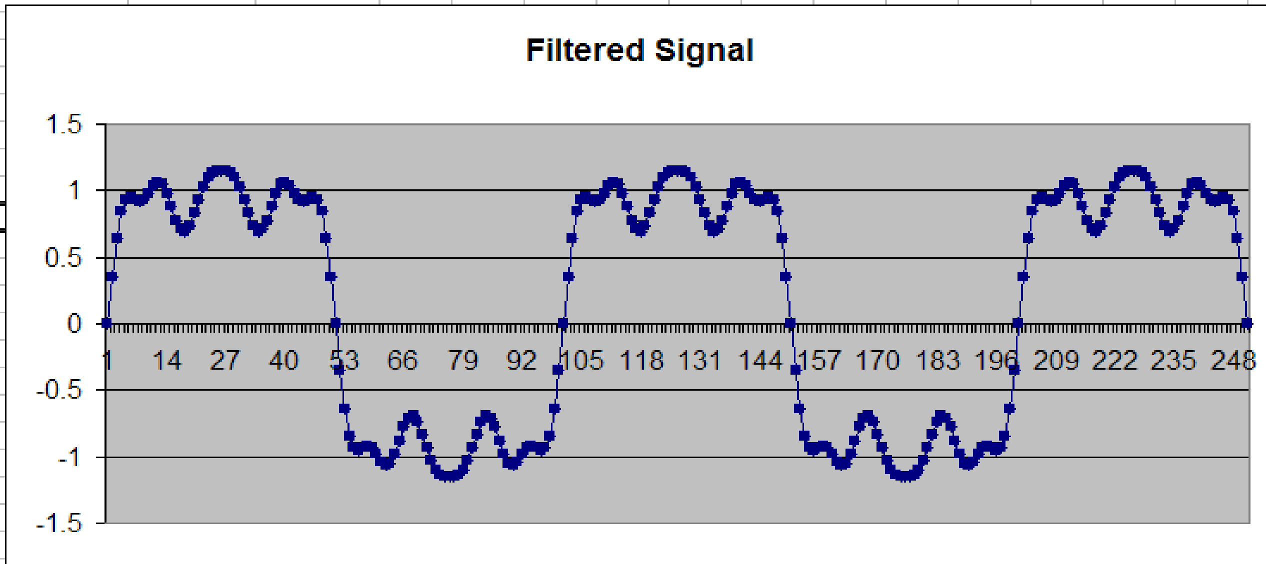
One of the most important mathematical theories in modern engineering

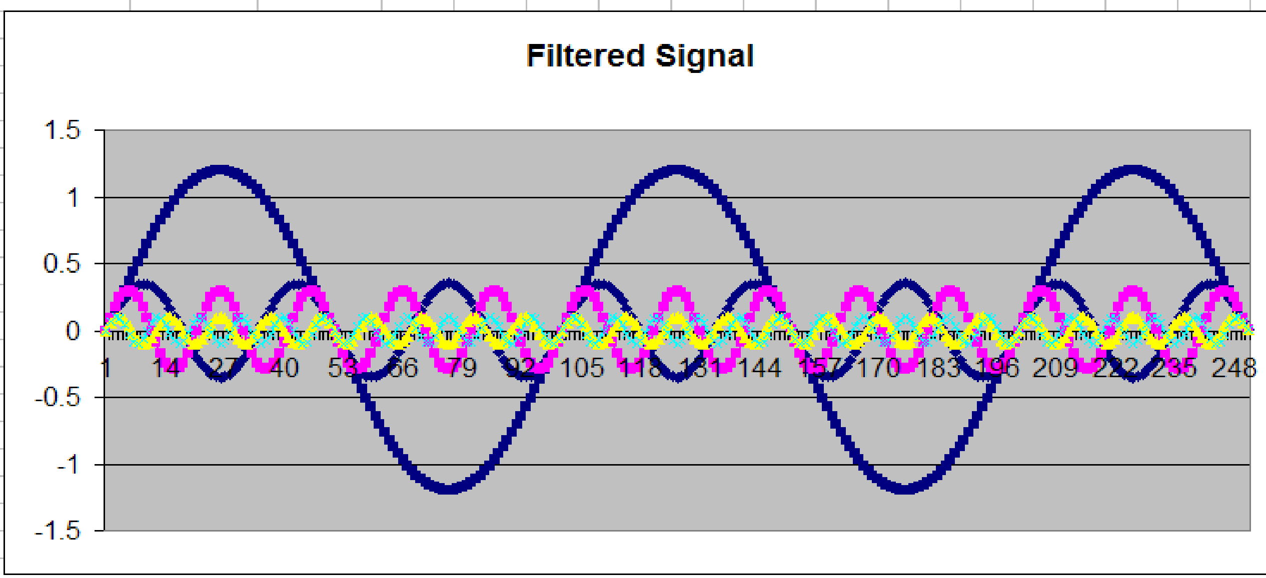
The Big Idea



Any function that periodically repeats itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient – a *Fourier series*

The Big Idea (cont…)





Frequency domain signal processing example in Excel

Discrete Fourier Transform (DFT)

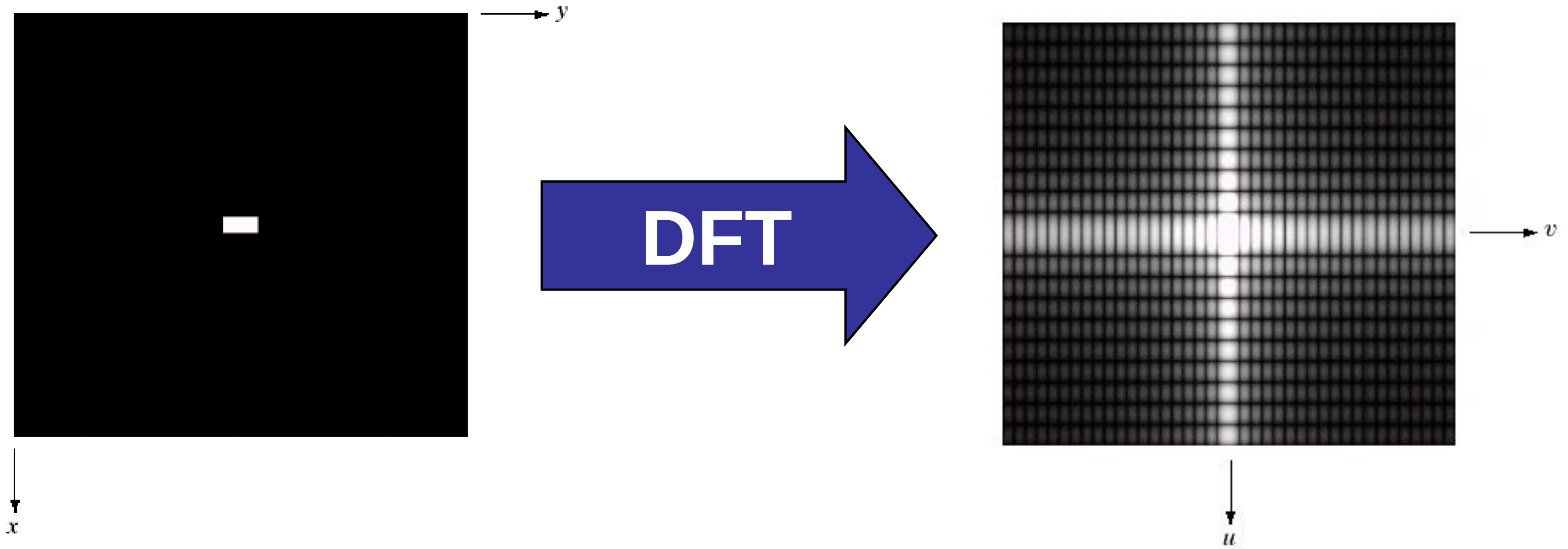
The *Discrete Fourier Transform* of $f(x, y)$, for x = 0, 1, 2…M-1 and y = 0,1,2…N-1, denoted by $F(u, v)$, is given by the equation:

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

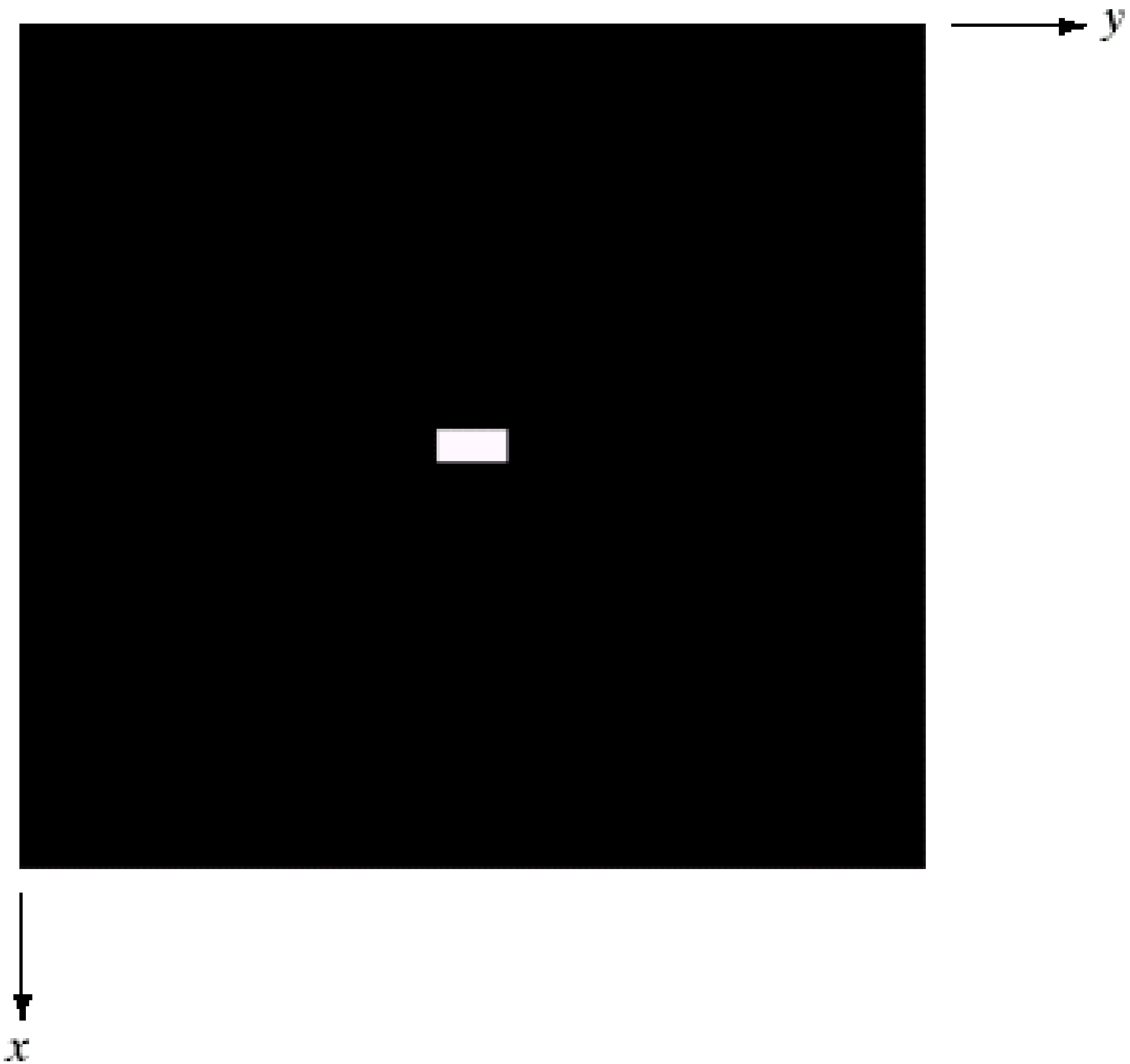
for u = 0, 1, 2…M-1 and v = 0, 1, 2…N-1.

DFT & Images

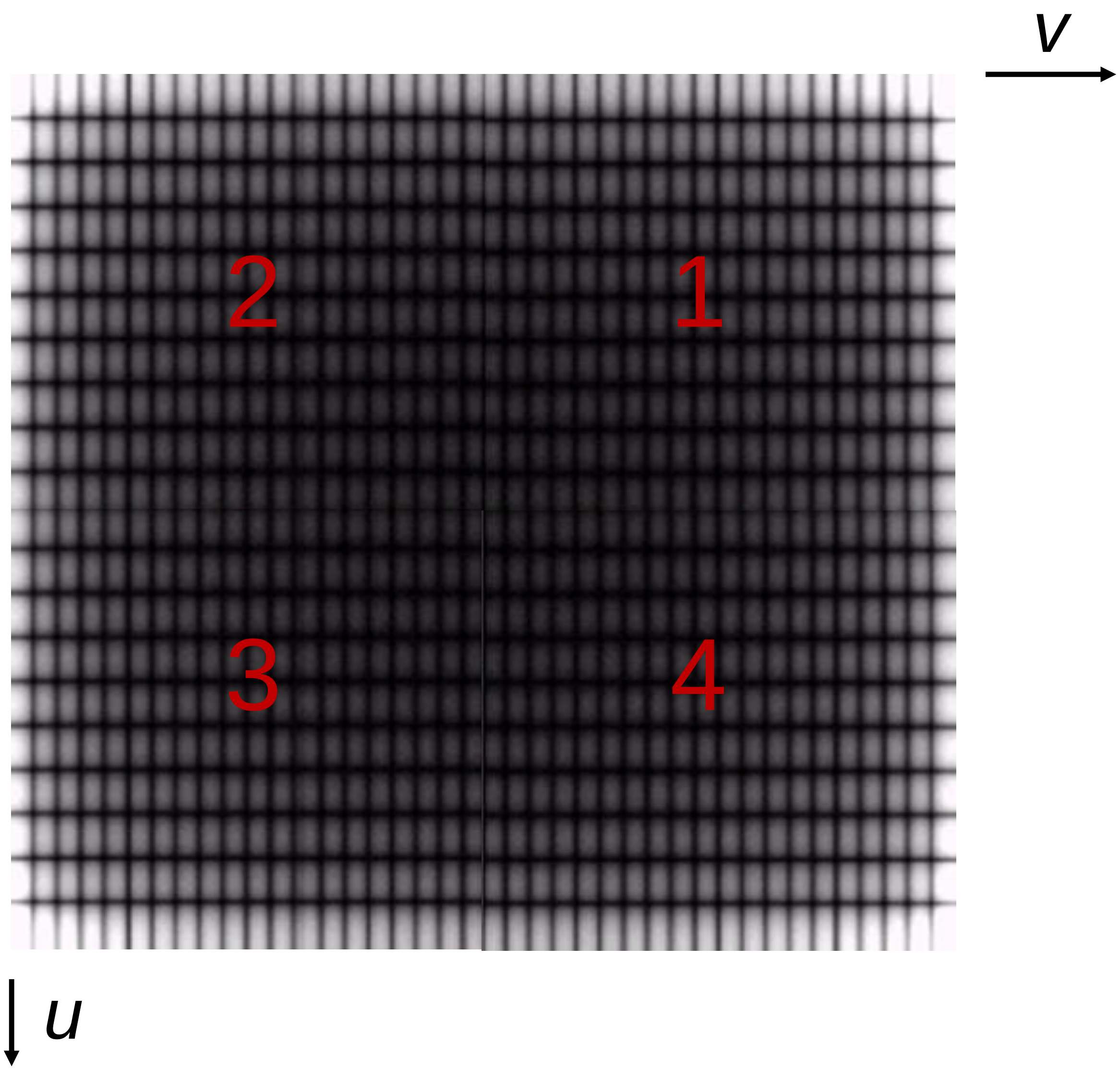
The DFT of a two dimensional image can be visualised by showing the spectrum of the images component frequencies



DFT & Images



DFT & Images



Shift the zero-frequency to the centre

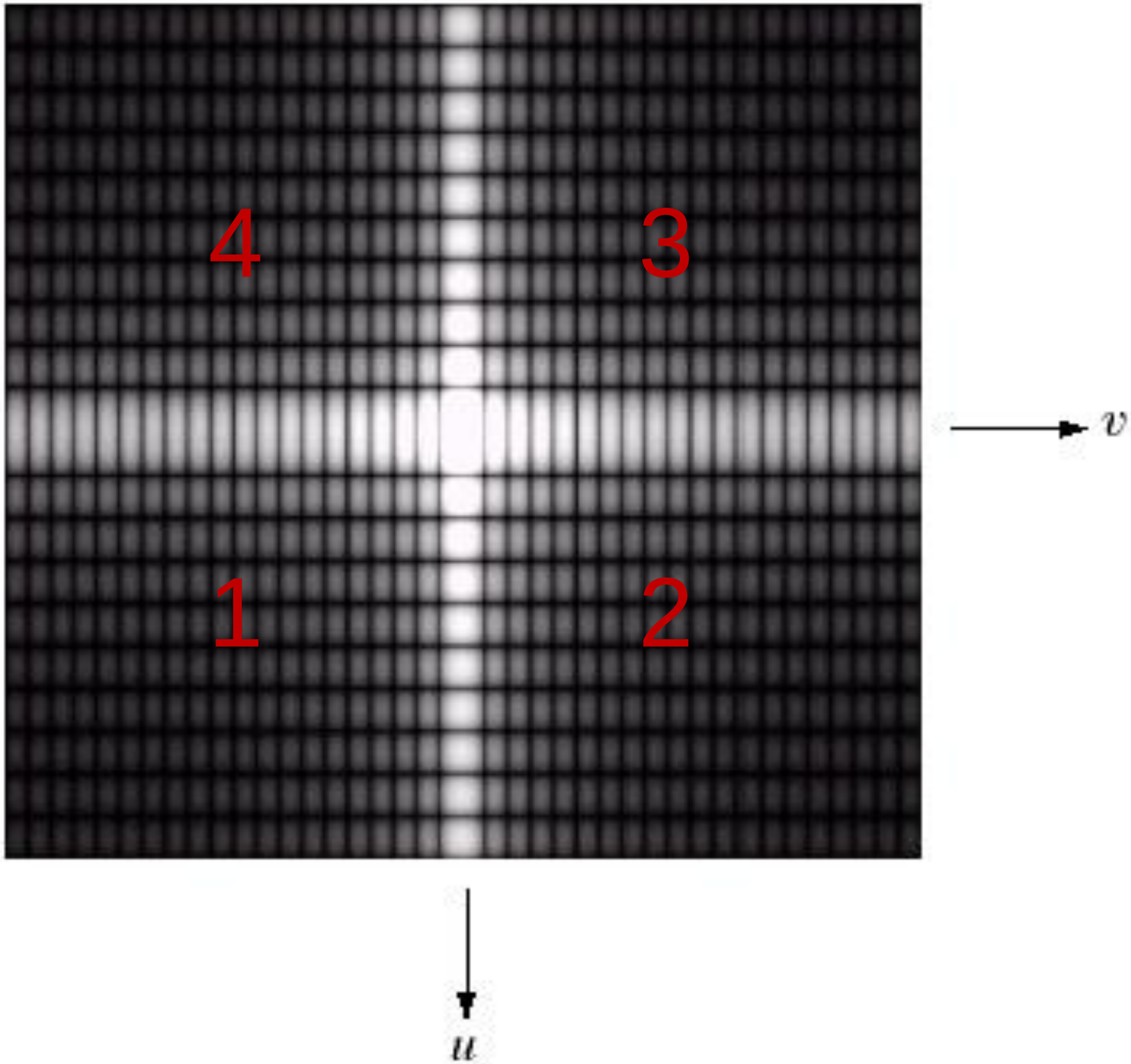
Y = fftshift(X) 通过将零频分量移动到数组中心，重新排列傅里叶变换 X

- 如果 X 是向量，则 fftshift 会将 X 的左右两半部分进行交换
- 如果 X 是矩阵，则 fftshift 会将 X 的第一象限与第三象限交换，将第二象限与第四象限交换
- 如果 X 是多维数组，则 fftshift 会沿每个维度交换 X 的半空间

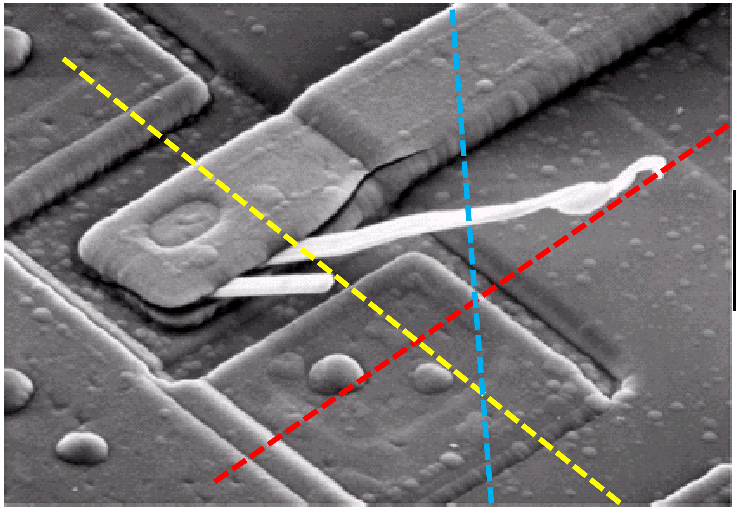
Y = fftshift(X,dim) 沿 X 的维度 dim 执行运算

例如，如果 X 是矩阵，其行表示多个一维变换，则 fftshift(X,2) 会将 X 的每一行的左右两半部分进行交换。

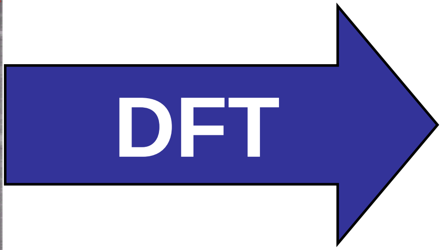
DFT & Images

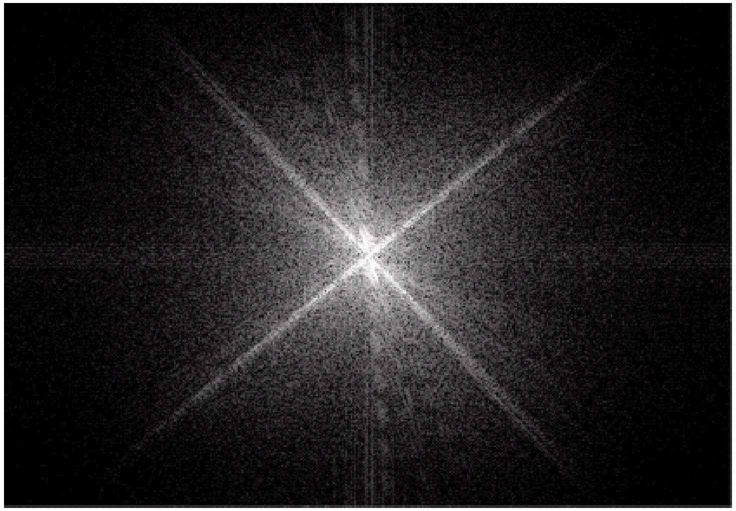


DFT & Images (cont…)



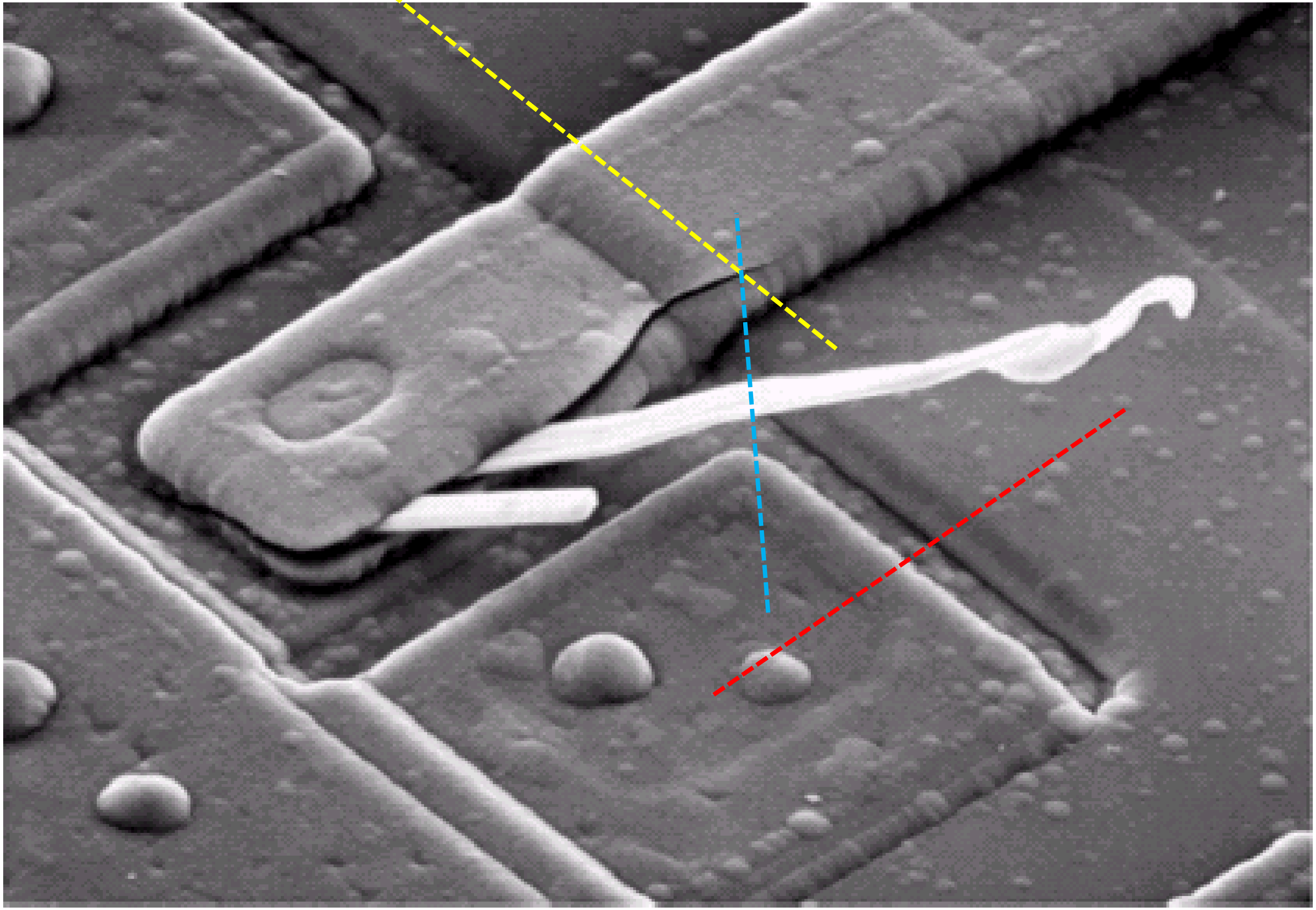
Scanning electron microscope image of an integrated circuit magnified ~2500 times



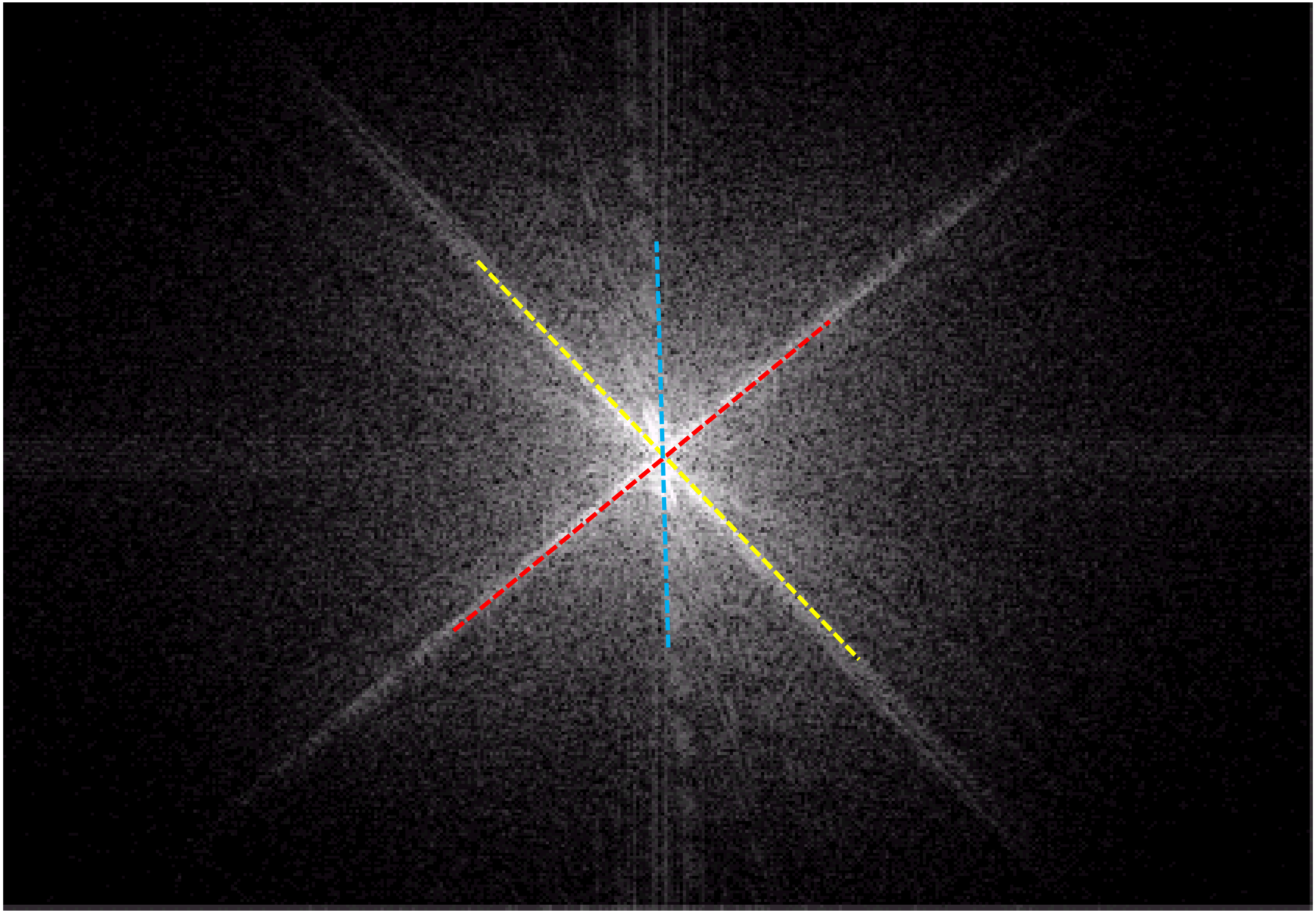


Fourier spectrum of the image

DFT & Images (cont…)



DFT & Images (cont…)



The Inverse DFT

It is really important to note that the Fourier transform is completely **reversible**

The inverse DFT is given by:

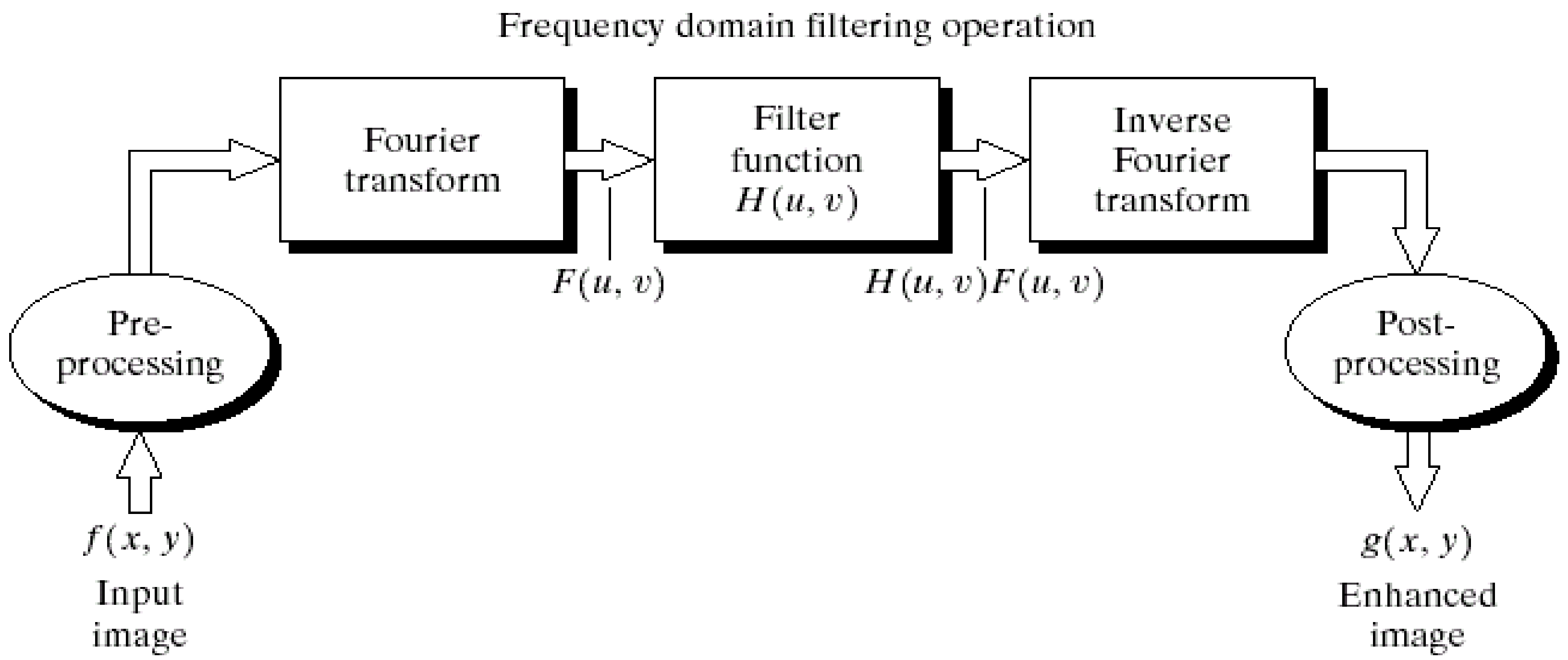
$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M + vy/N)}$$

for x = 0, 1, 2…M-1 and y = 0, 1, 2…N-1

The DFT and Image Processing

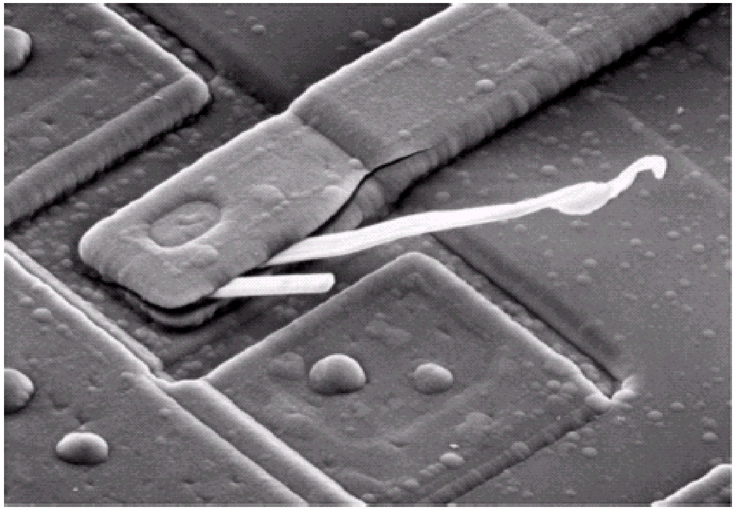
To filter an image in the frequency domain:

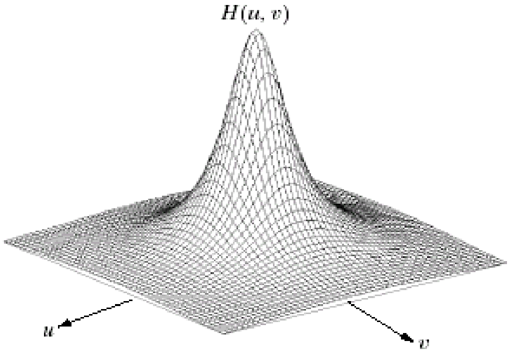
1. Compute *F(u,v)* the DFT of the image
2. Multiply *F(u,v)* by a filter function *H(u,v)*
3. Compute the inverse DFT of the result

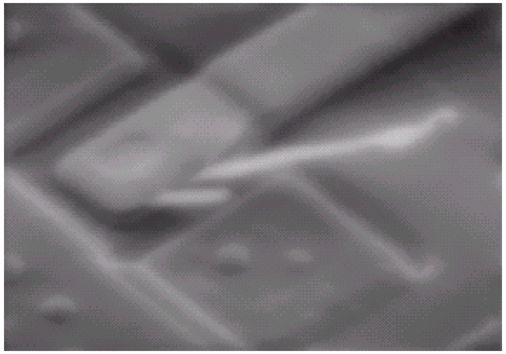


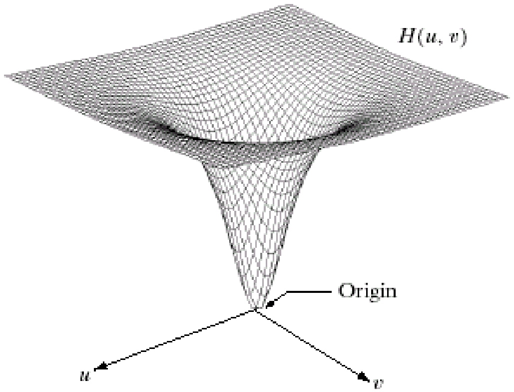
Some Basic Frequency Domain Filters

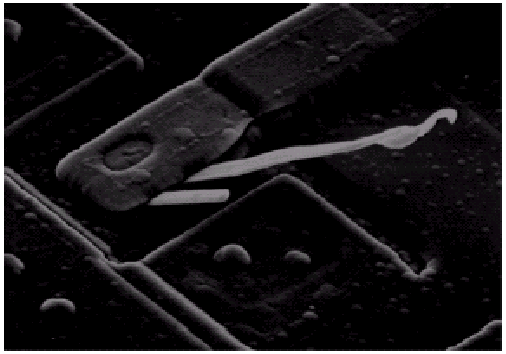
Low Pass Filter





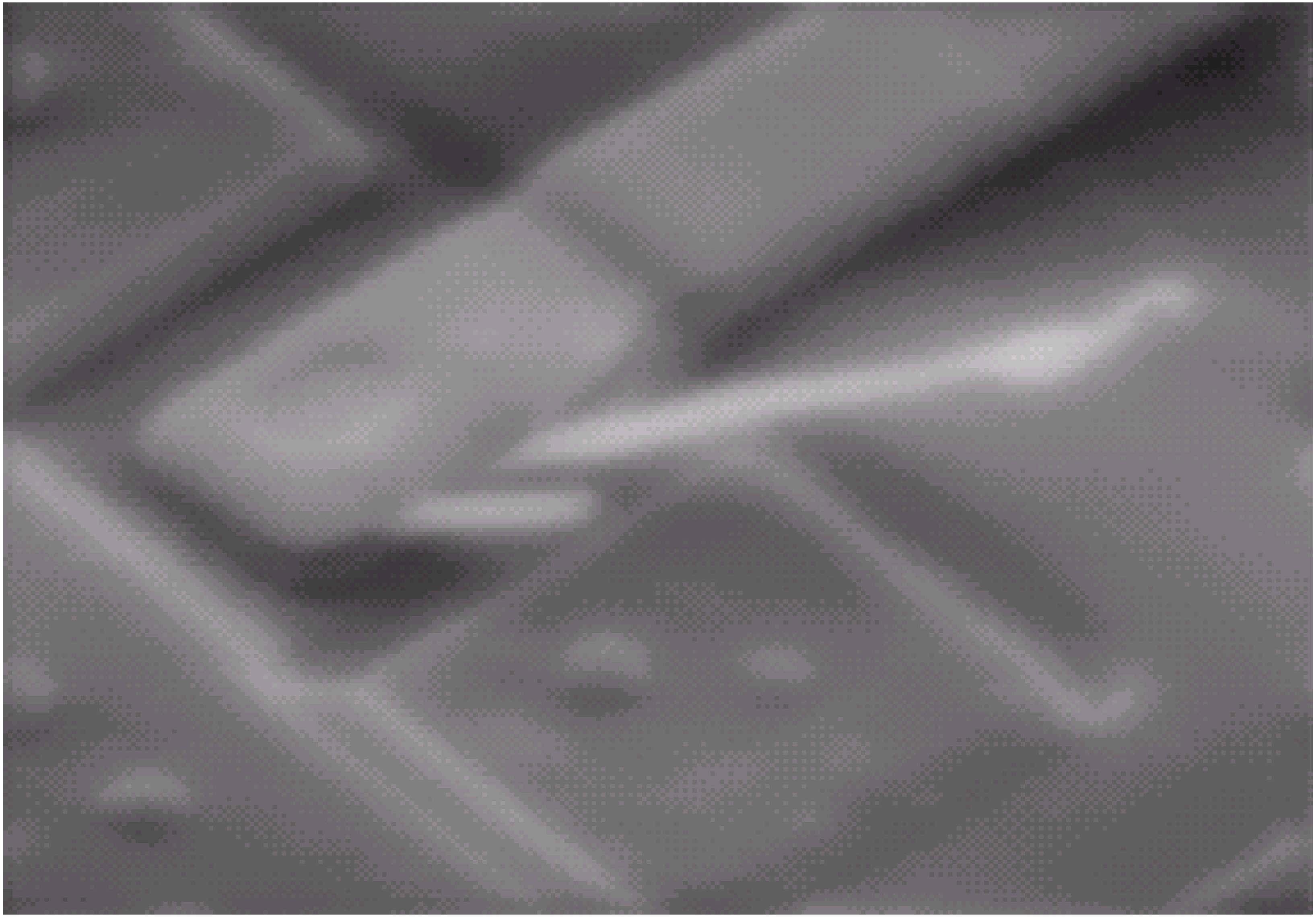




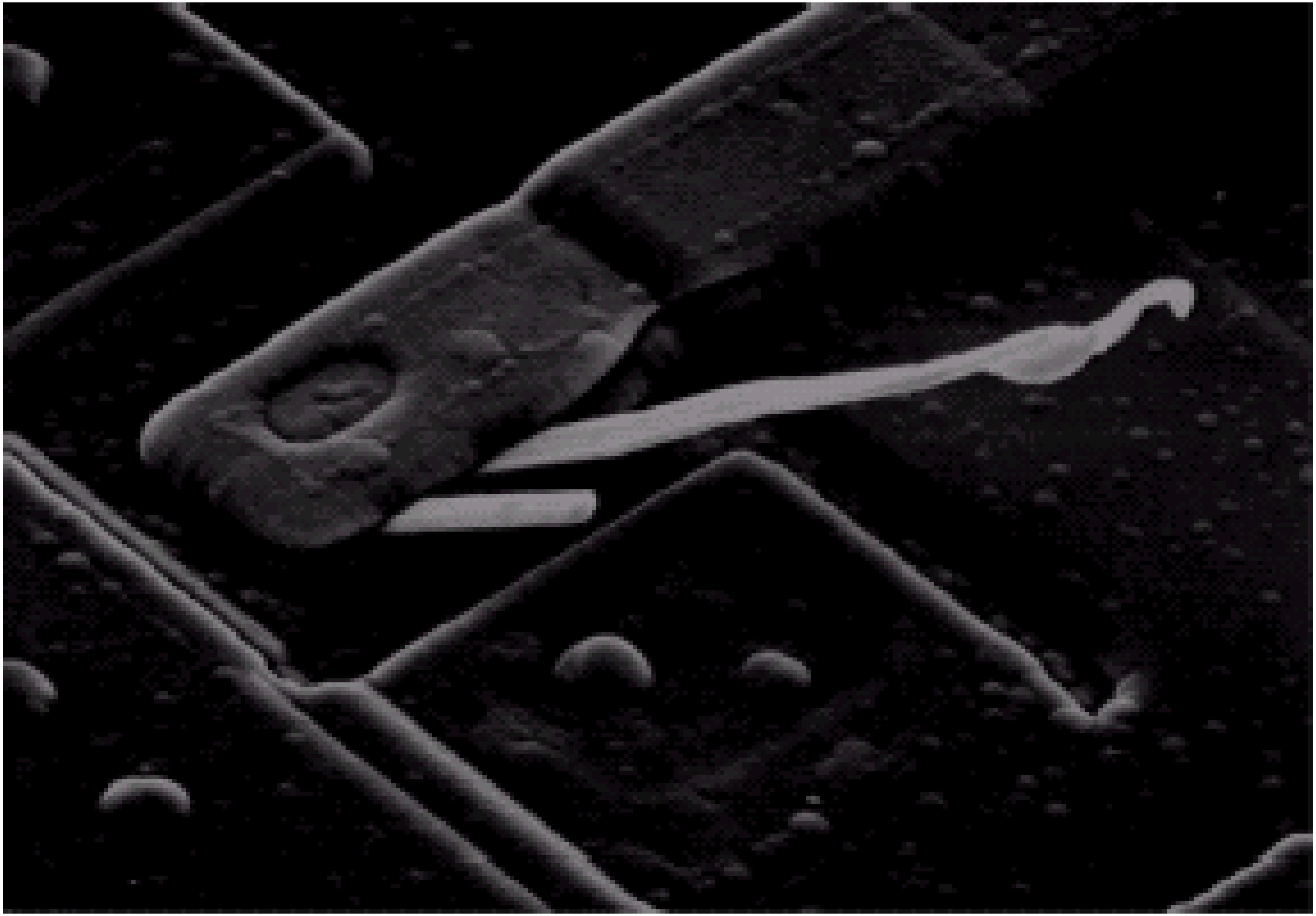


High Pass Filter

Low Pass Filtered



High Pass Filtered



Smoothing Frequency Domain Filters

Smoothing is achieved in the frequency domain by dropping out the high frequency components

The basic model for filtering is:

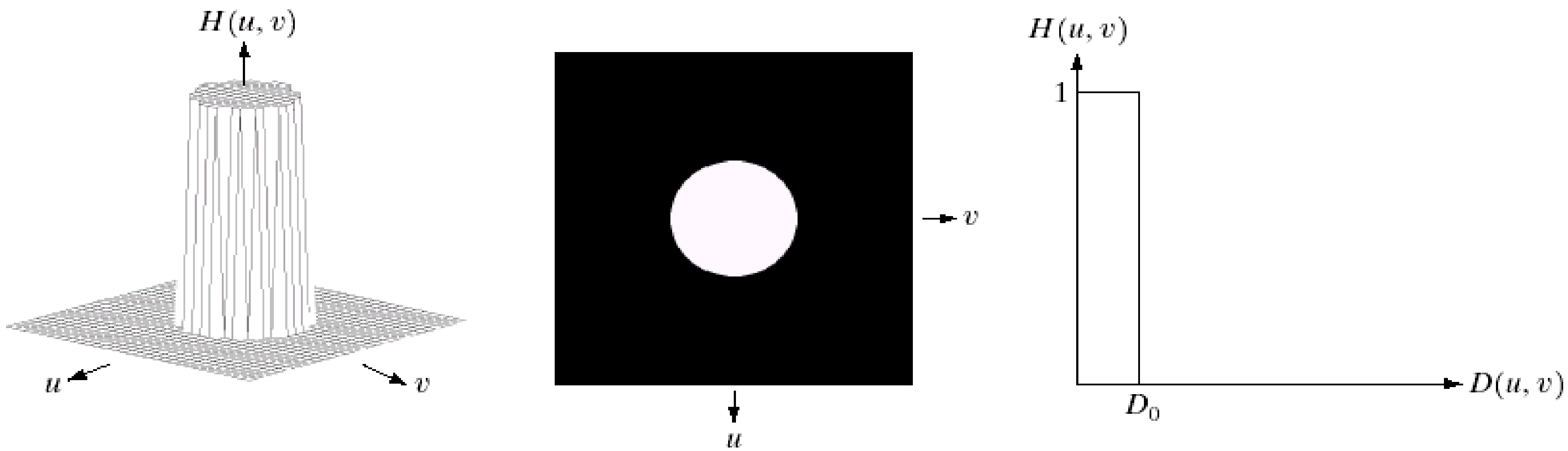
$$G(u,v) = H(u,v)F(u,v)$$

where $F(u,v)$ is the Fourier transform of the image being filtered and $H(u,v)$ is the filter transform function

Low pass filters – only pass the low frequencies, drop the high ones

Ideal Low Pass Filter

Simply cut off all high frequency components that are a specified distance D_0 from the origin of the transform



changing the distance changes the behaviour of the filter

Ideal Low Pass Filter (cont…)

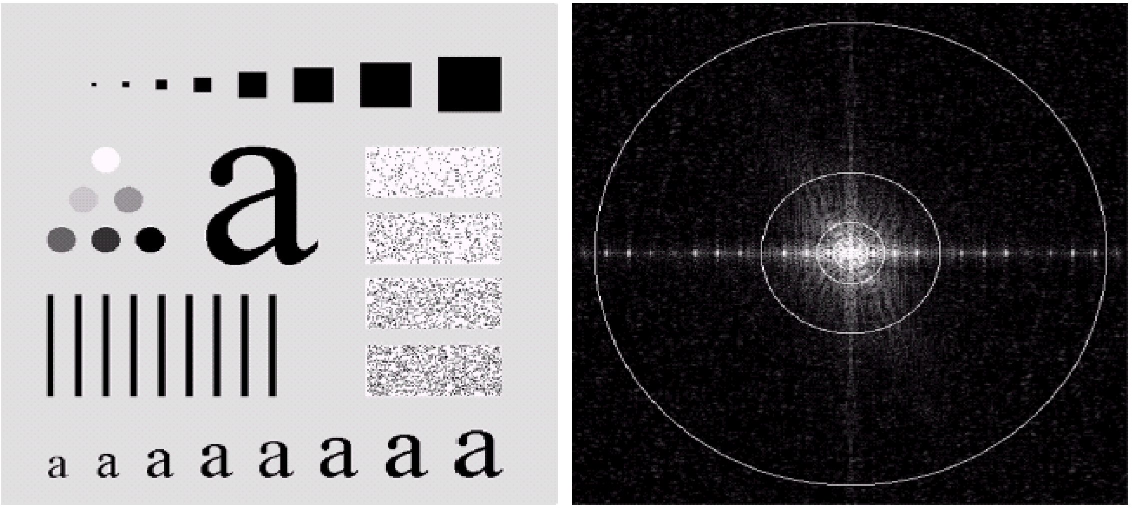
The transfer function for the ideal low pass filter can be given as:

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

where $D(u,v)$ is given as:

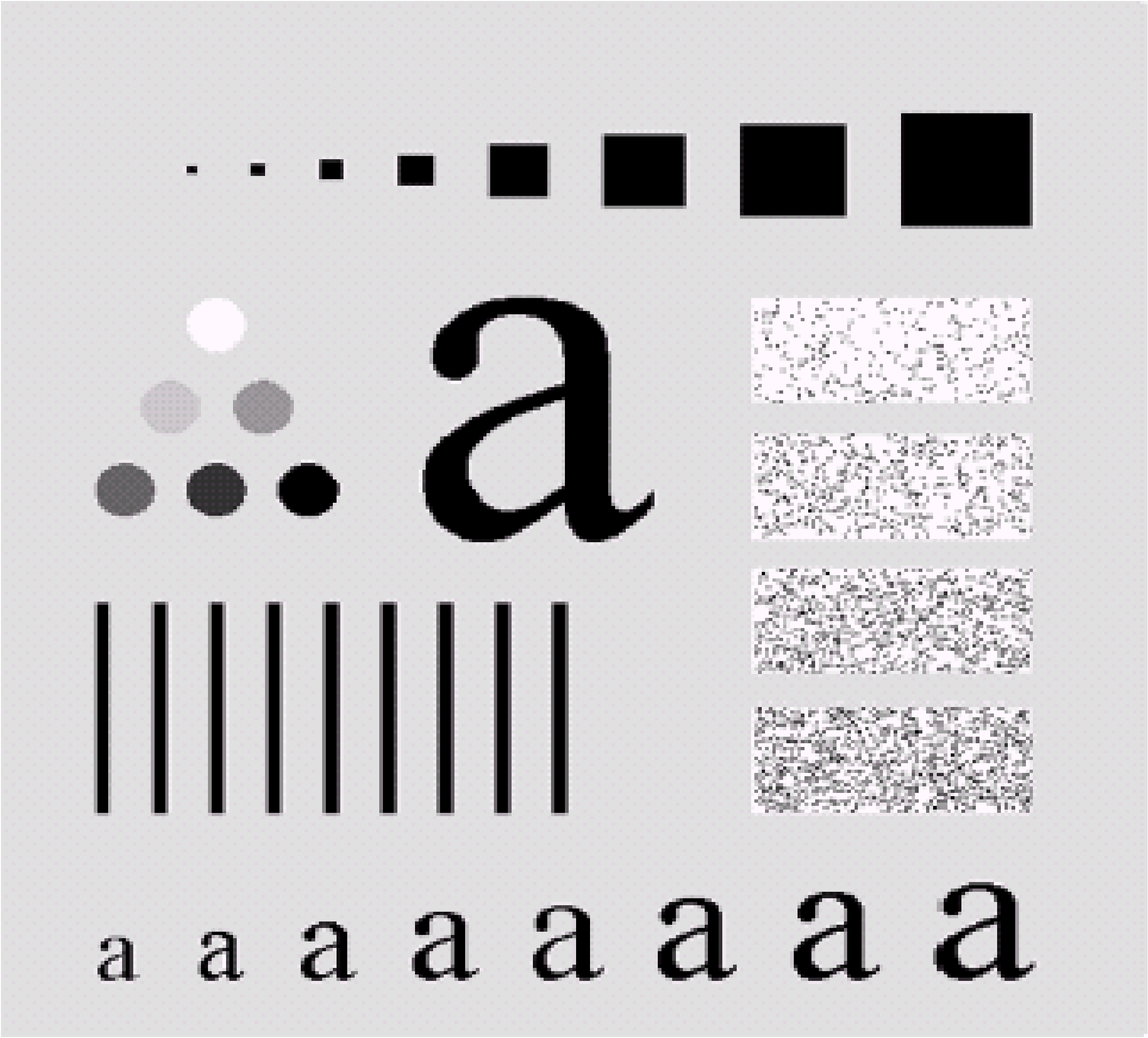
$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

Ideal Low Pass Filter (cont…)

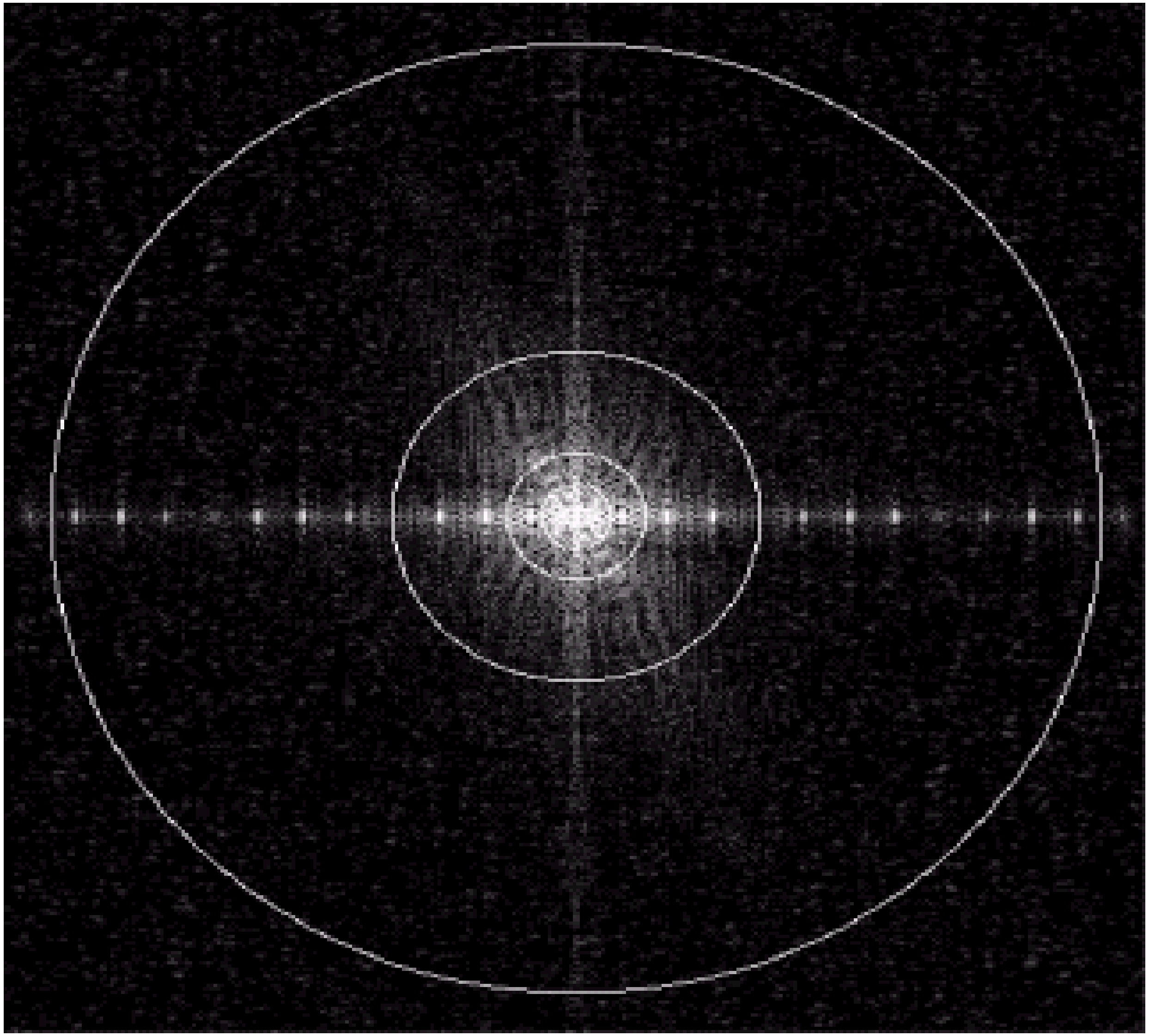


Above we show an image, it's Fourier spectrum and a series of ideal low pass filters of radius 5, 15, 30, 80 and 230 superimposed on top of it

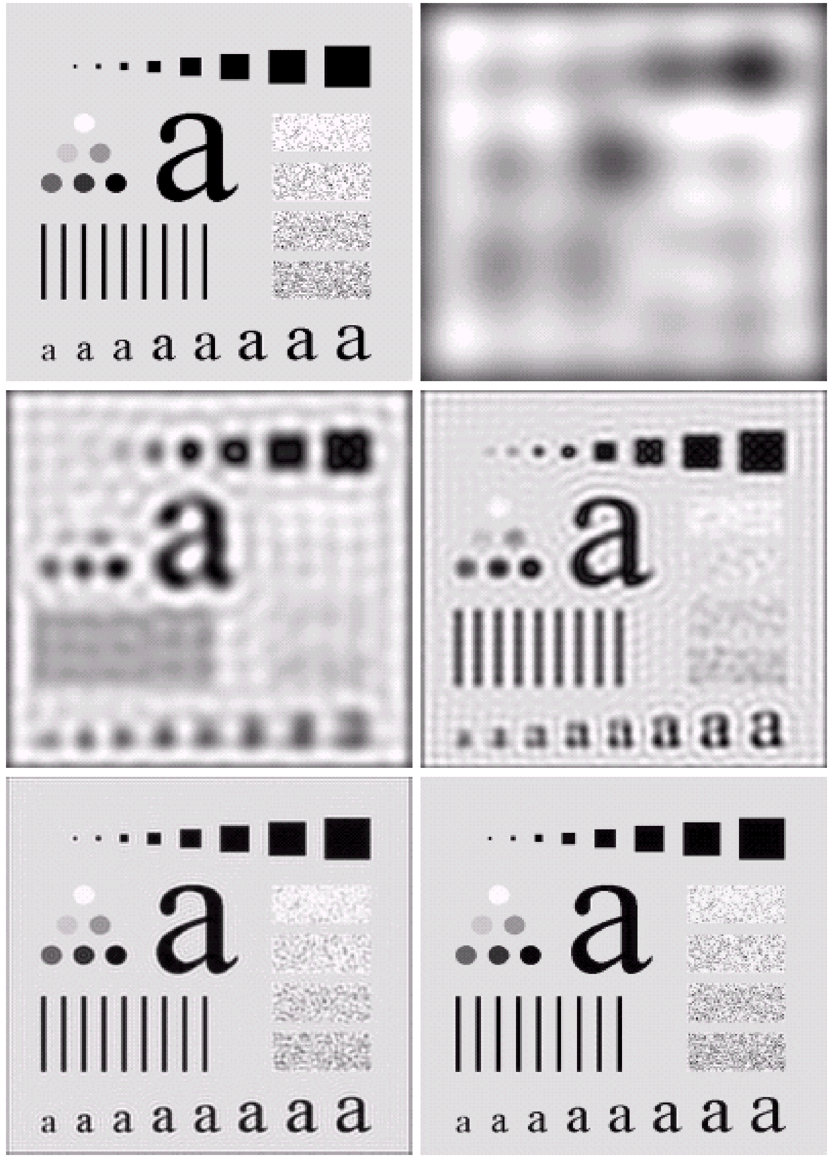
Ideal Low Pass Filter (cont…)



Ideal Low Pass Filter (cont…)



Ideal Low Pass Filter (cont…)



Original image

Result of filtering with ideal low pass filter of radius 5

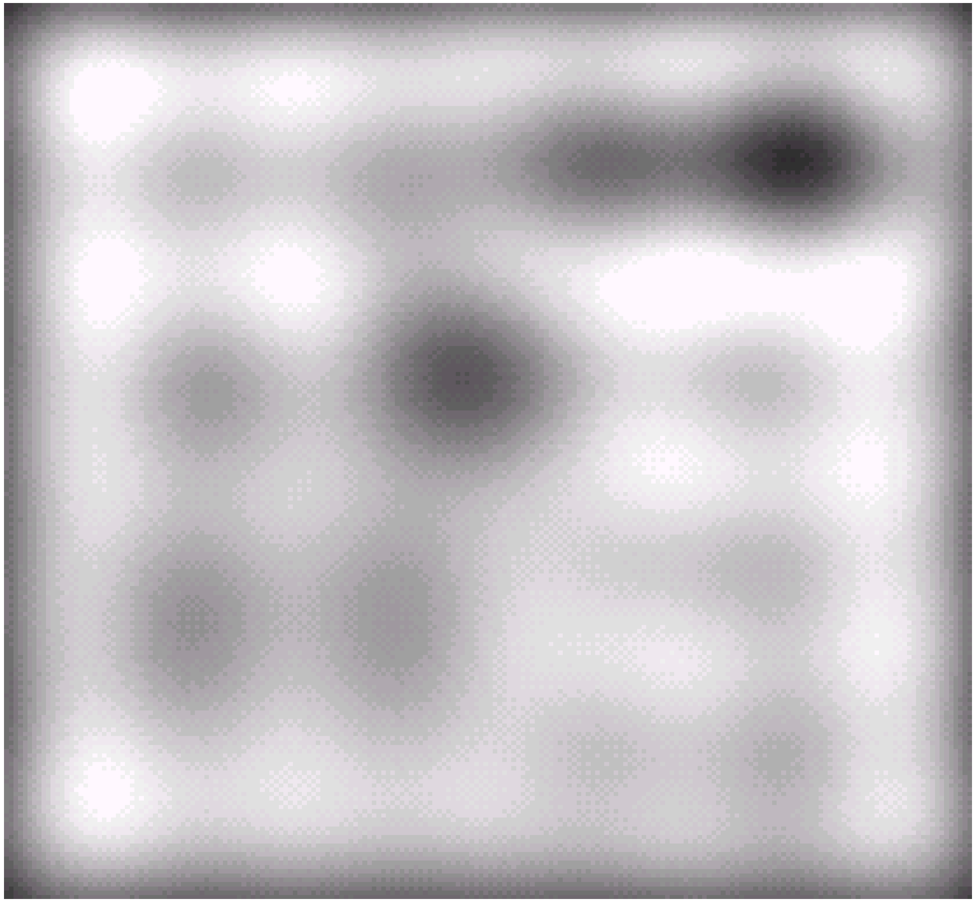
Result of filtering with ideal low pass filter of radius 15

Result of filtering with ideal low pass filter of radius 30

Result of filtering with ideal low pass filter of radius 80

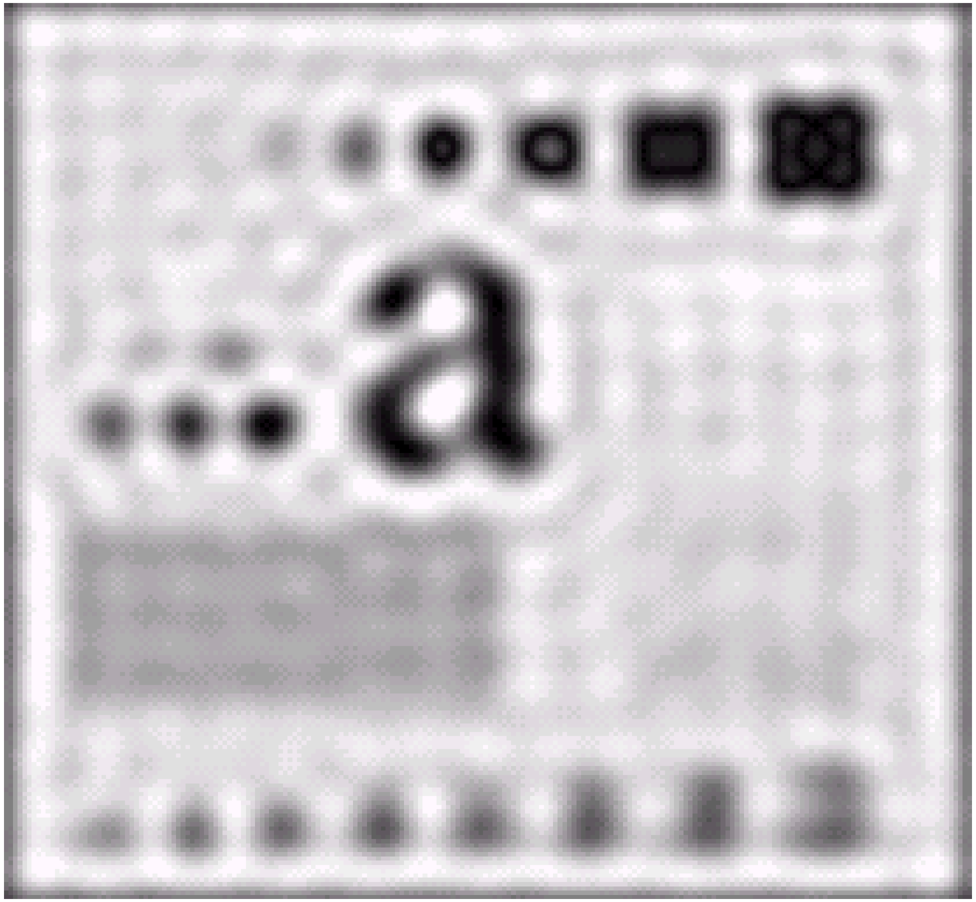
Result of filtering with ideal low pass filter of radius 230

Ideal Low Pass Filter (cont…)



Result of filtering with ideal low pass filter of radius 5

Ideal Low Pass Filter (cont…)

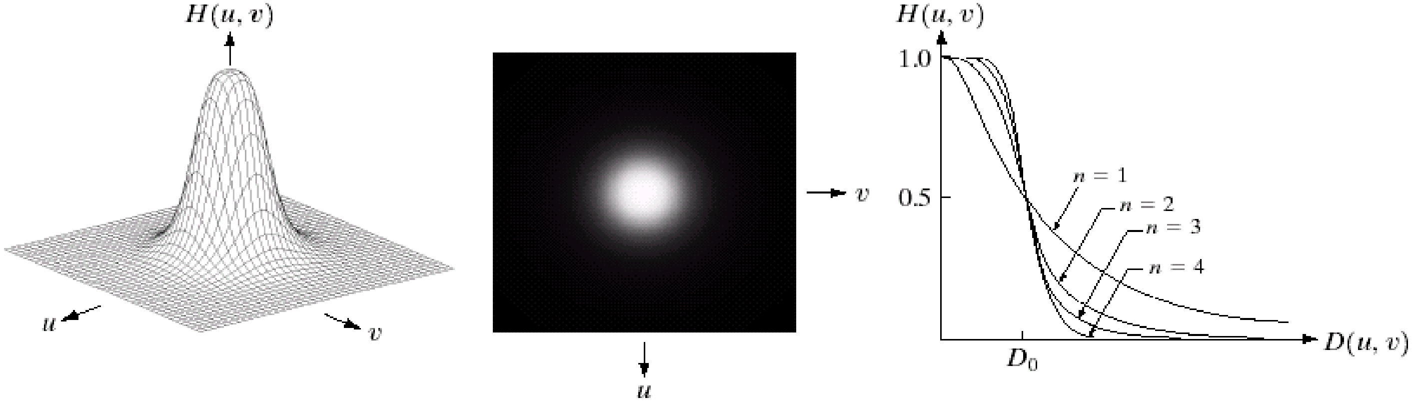


Result of filtering with ideal low pass filter of radius 15

Butterworth Lowpass Filters

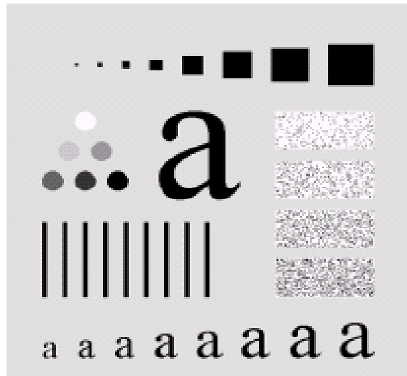
The transfer function of a Butterworth lowpass filter of order n with cutoff frequency at distance D_0 from the origin is defined as:

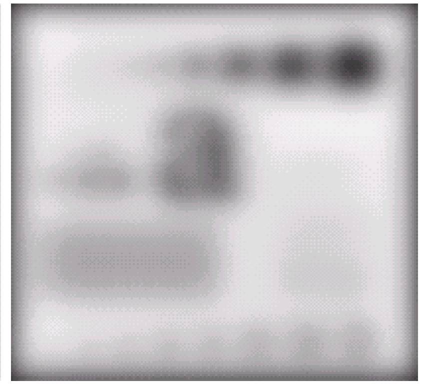
$$H(u,v)=\frac{1}{1+[D(u,v)/D_0]^{2n}}$$



Butterworth Lowpass Filter (cont…)

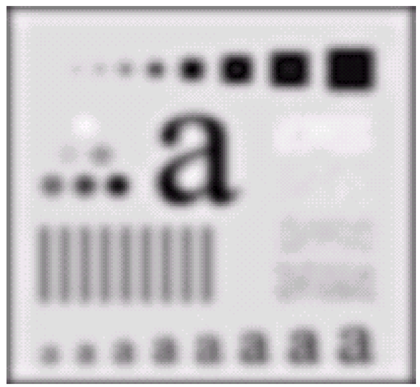
Original image

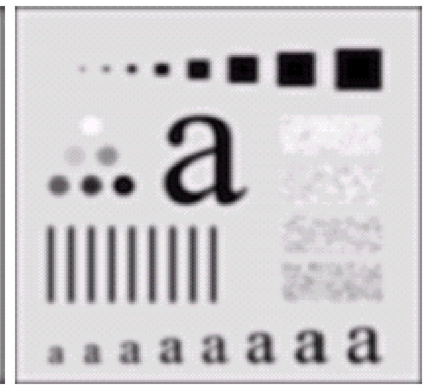




Result of filtering with Butterworth filter of order 2 and cutoff radius 5

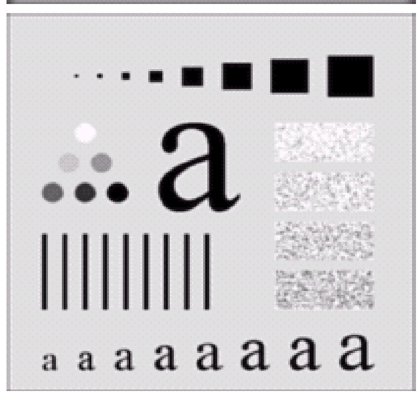
Result of filtering with Butterworth filter of order 2 and cutoff radius 15

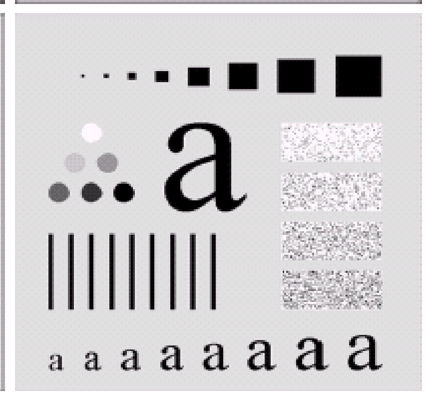




Result of filtering with Butterworth filter of order 2 and cutoff radius 30

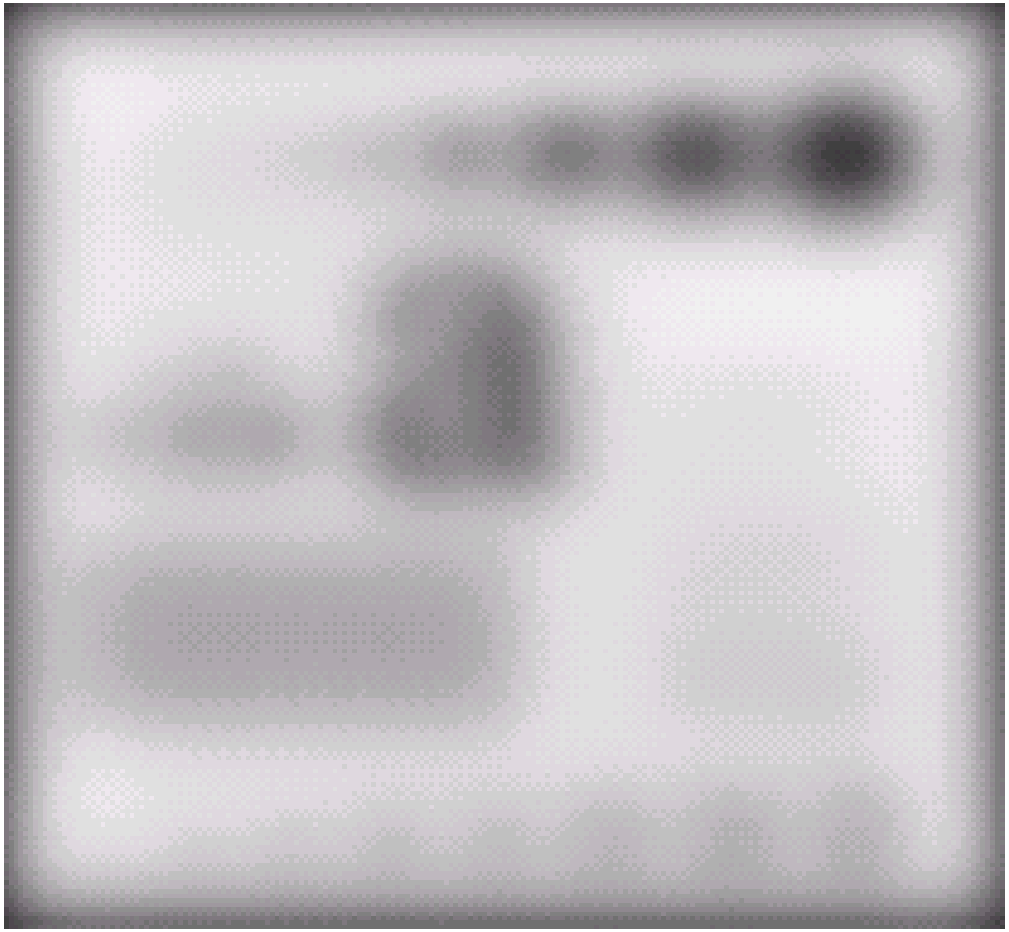
Result of filtering with Butterworth filter of order 2 and cutoff radius 80





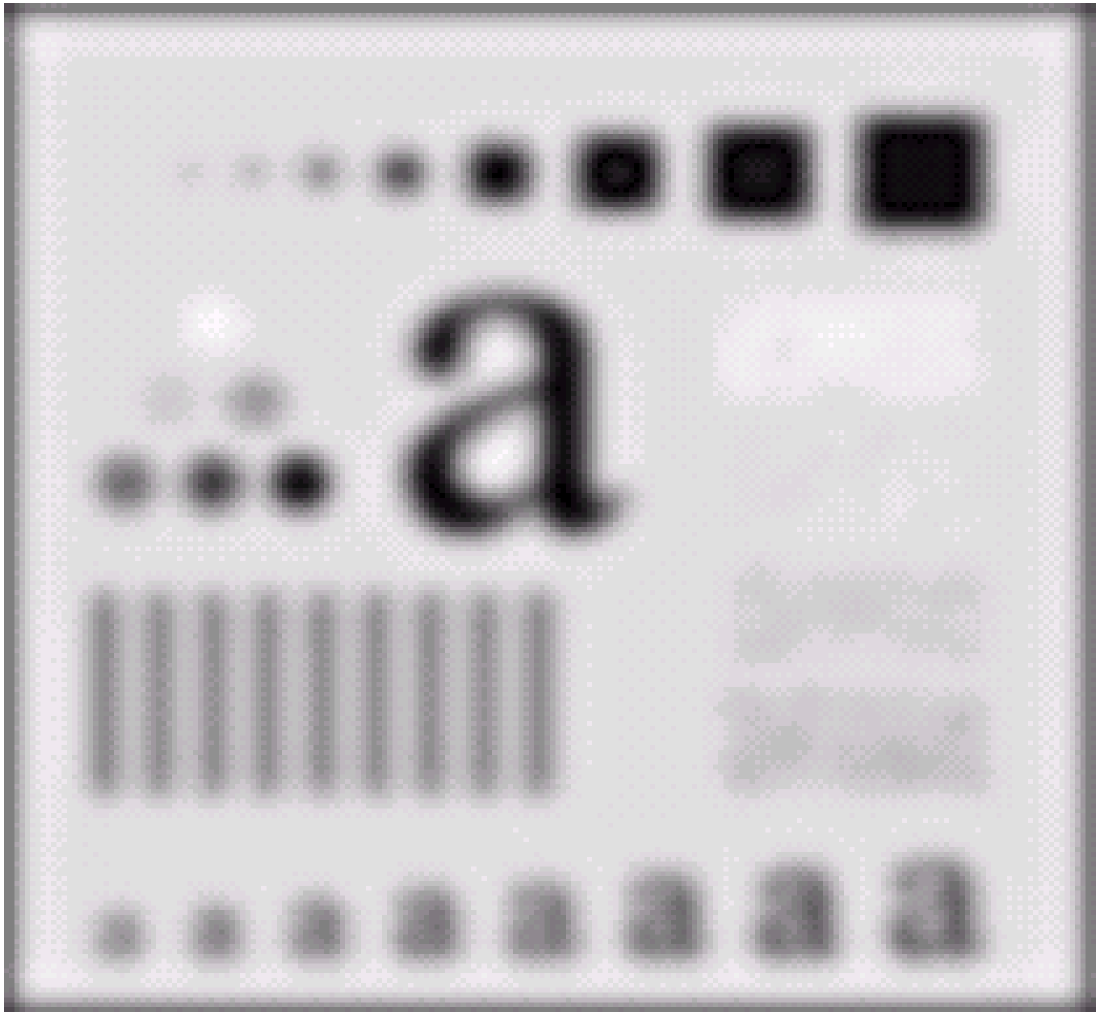
Result of filtering with Butterworth filter of order 2 and cutoff radius 230

Butterworth Lowpass Filter (cont…)



Result of filtering with Butterworth filter of order 2 and cutoff radius 5

Butterworth Lowpass Filter (cont…)

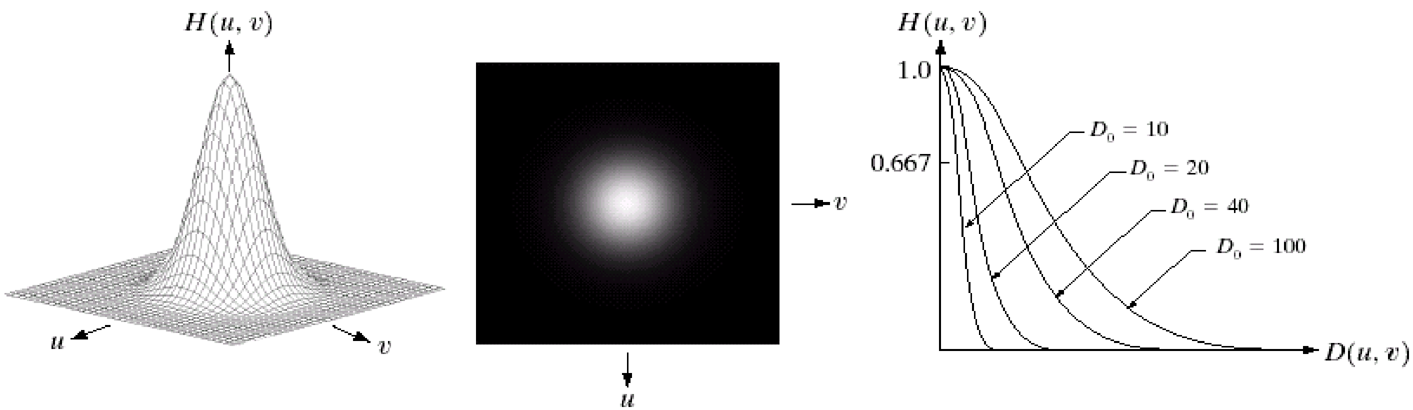


Result of filtering with Butterworth filter of order 2 and cutoff radius 15

Gaussian Lowpass Filters

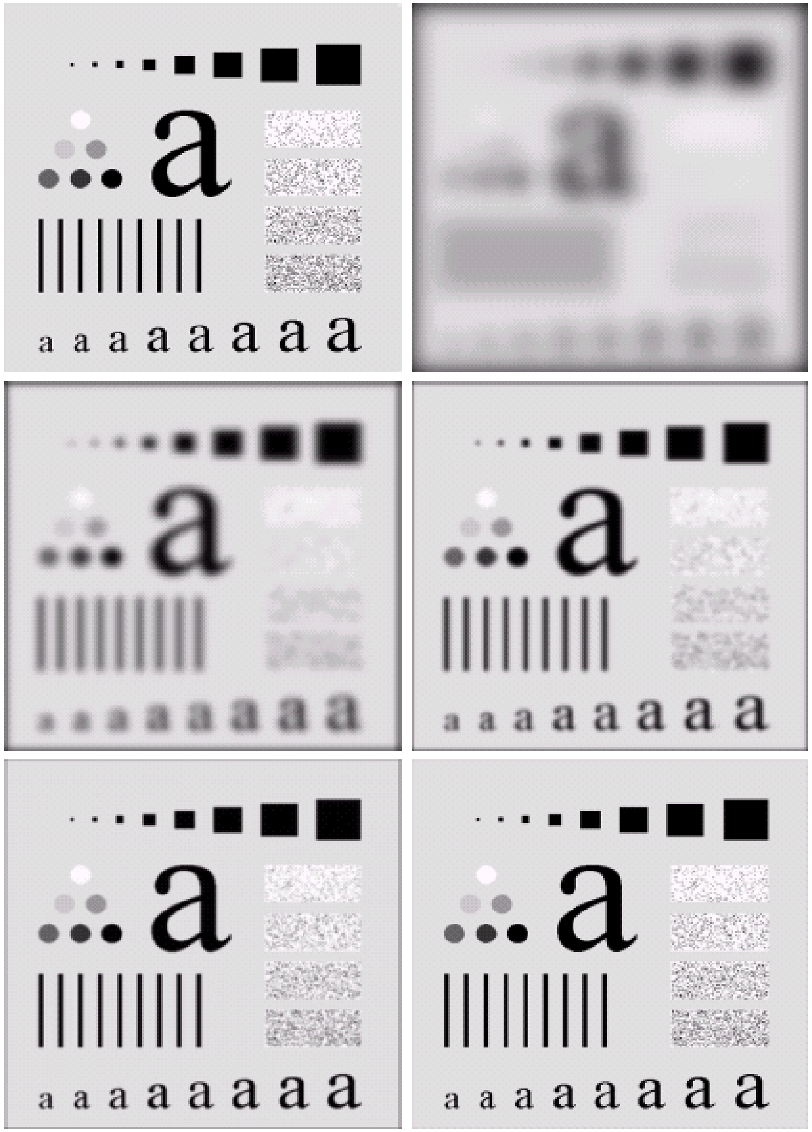
The transfer function of a Gaussian lowpass filter is defined as:

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



Gaussian Lowpass Filters (cont…)

Original image



Result of filtering with Gaussian filter with cutoff radius 5

Result of filtering with Gaussian filter with cutoff radius 15

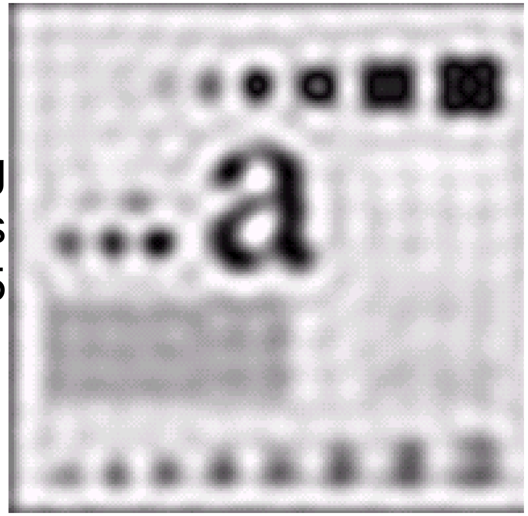
Result of filtering with Gaussian filter with cutoff radius 30

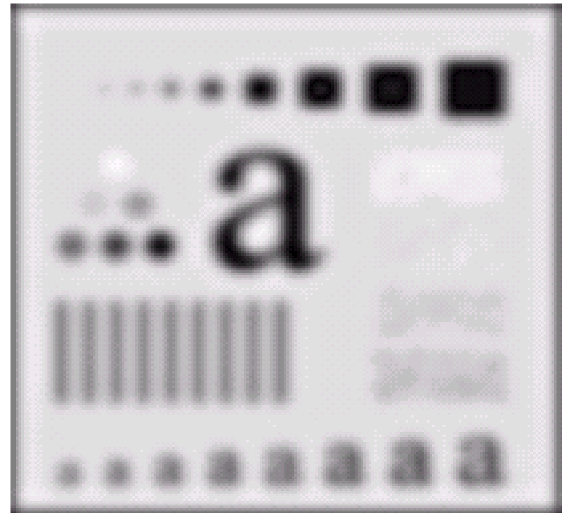
Result of filtering with Gaussian filter with cutoff radius 85

Result of filtering with Gaussian filter with cutoff radius 230

Lowpass Filters Compared

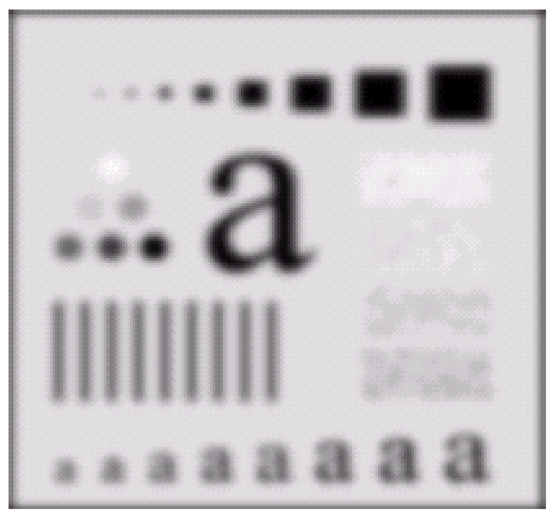
Result of filtering with ideal low pass filter of radius 15





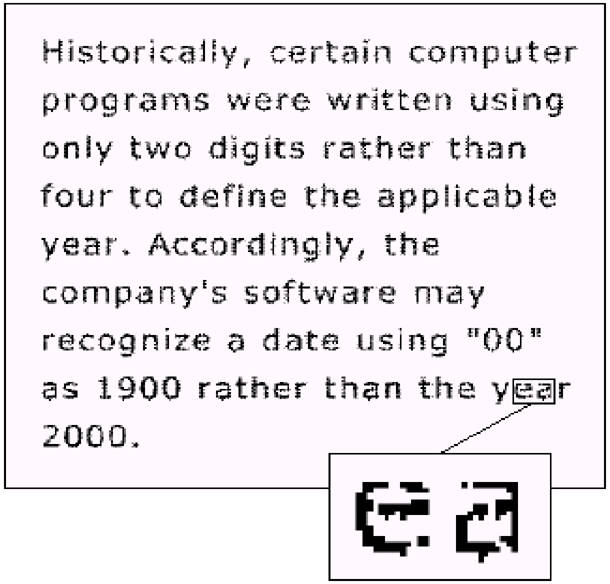
Result of filtering with Butterworth filter of order 2 and cutoff radius 15

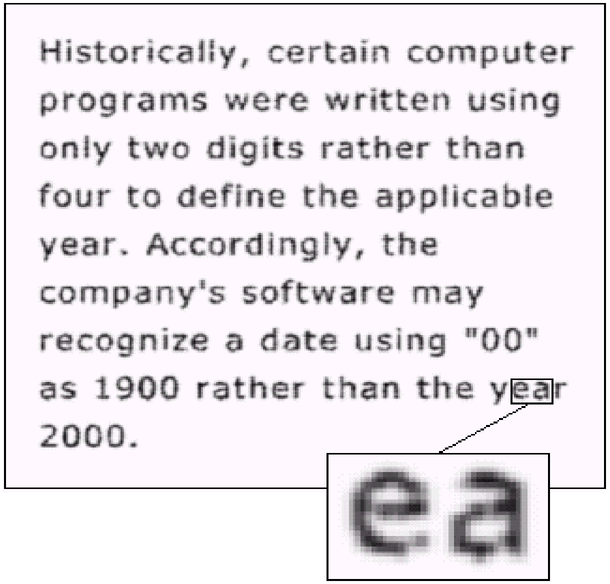
Result of filtering with Gaussian filter with cutoff radius 15



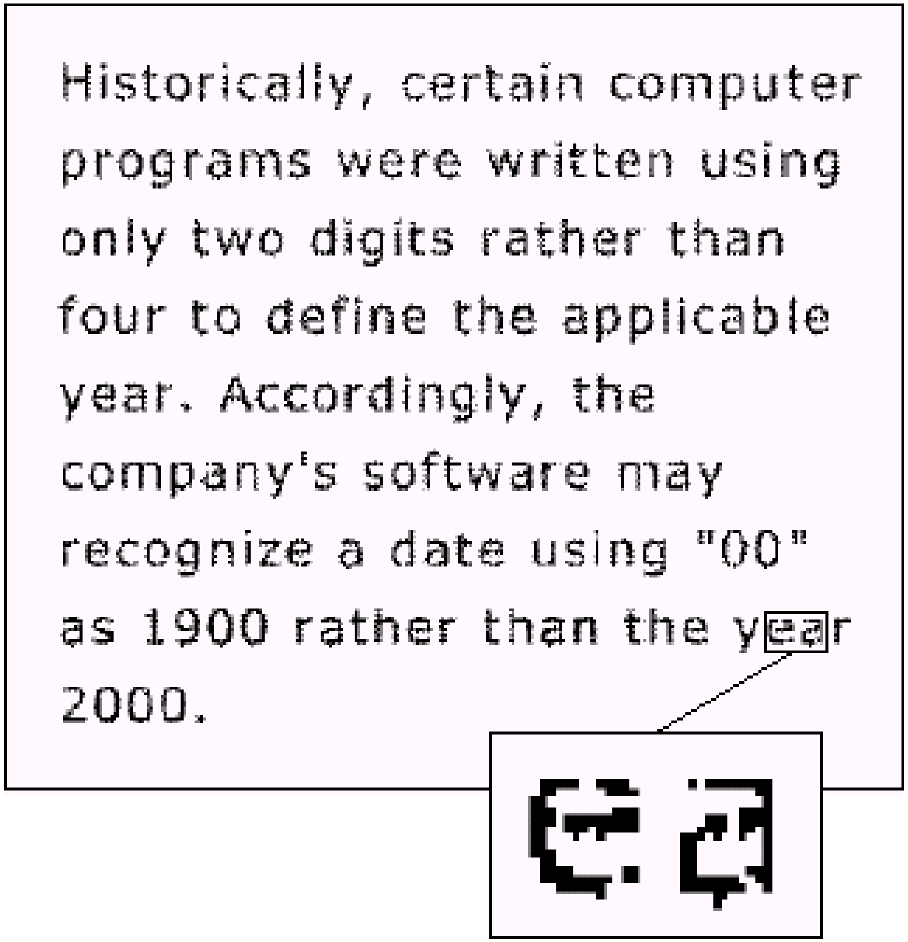
Lowpass Filtering Examples

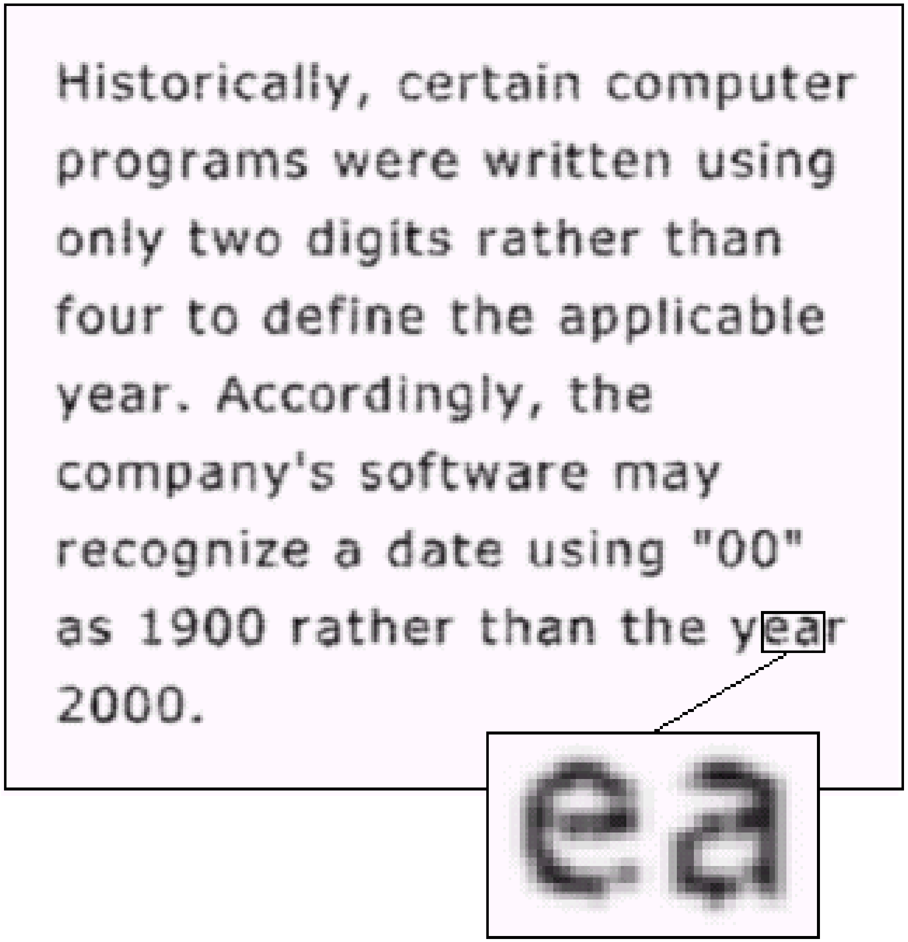
A low pass Gaussian filter is used to connect broken text





Lowpass Filtering Examples





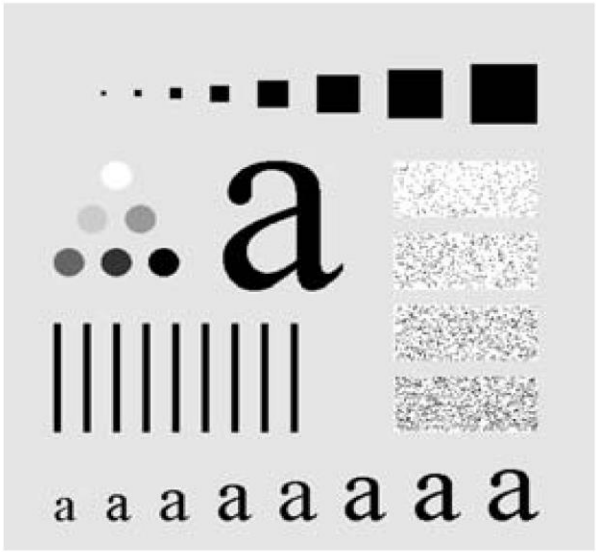
Lowpass Filtering Examples (cont…)

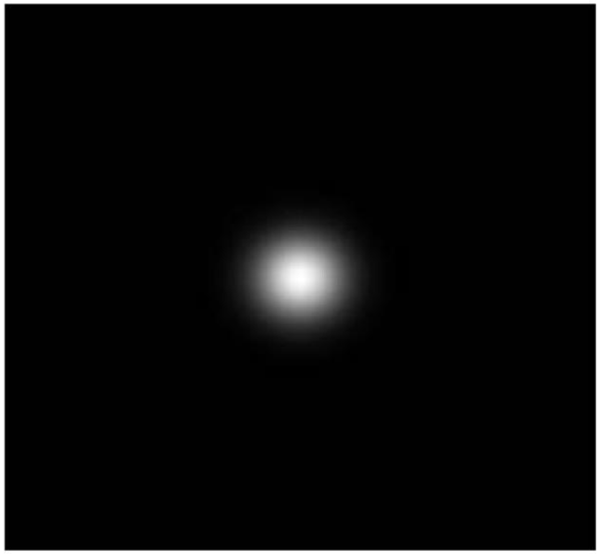
Different lowpass Gaussian filters used to remove blemishes in a photograph

Lowpass Filtering Examples (cont…)

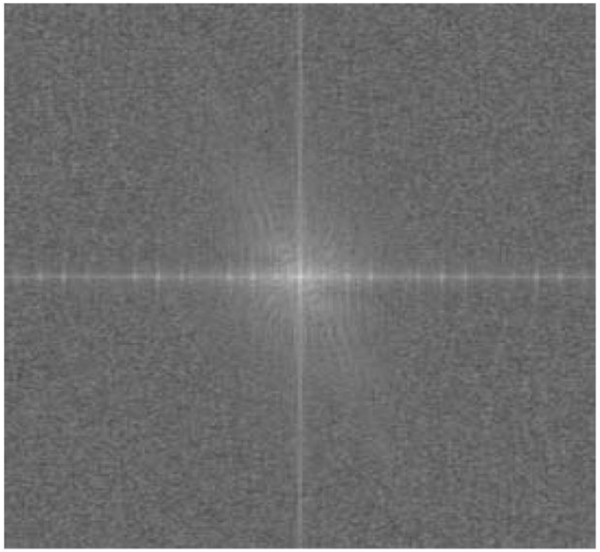
Lowpass Filtering Examples (cont…)

Original image





Gaussian lowpass filter



Processed image

Lowpass Filtering Examples (cont…)

Original image

Gaussian lowpass filter with $D_0 = 50$

Gaussian lowpass filter with $D_0 = 20$

Sharpening in the Frequency Domain

Edges and fine detail in images are associated with high frequency components

High pass filters – only pass the high frequencies, drop the low ones

High pass frequencies are precisely the reverse of low pass filters, so:

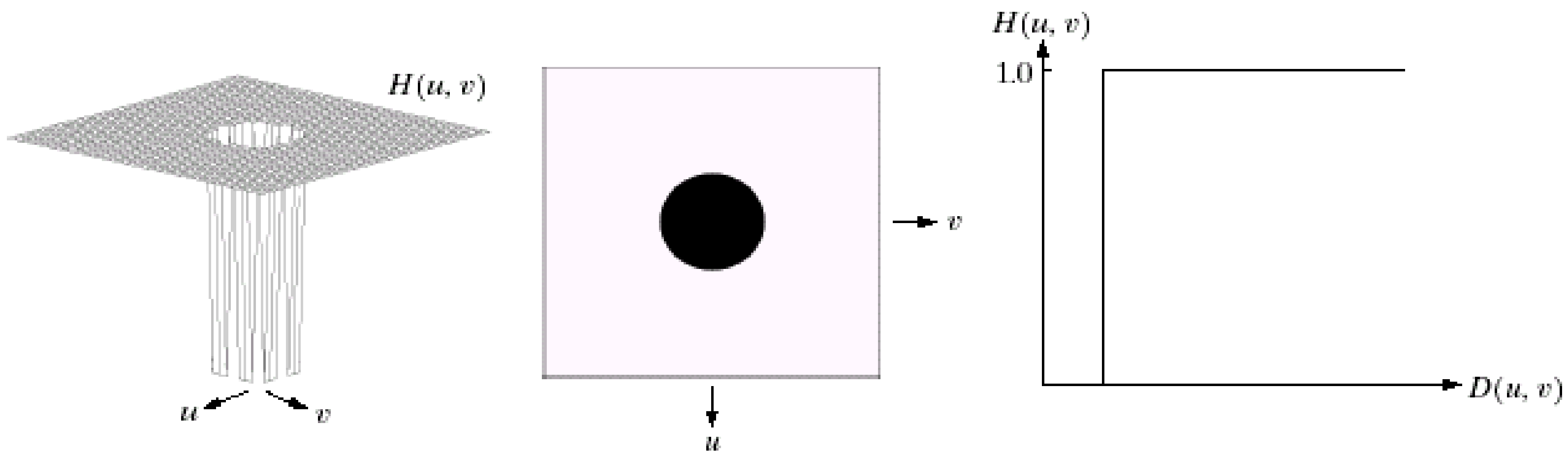
$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

Ideal High Pass Filters

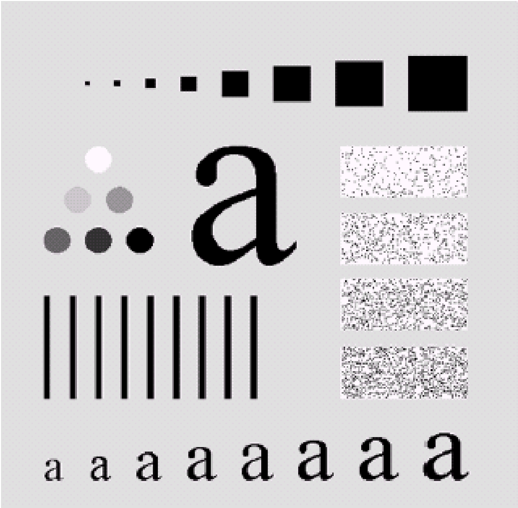
The ideal high pass filter is given as:

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

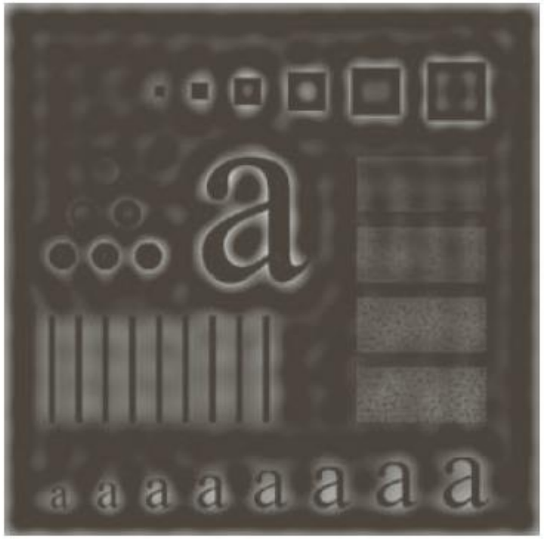
where D_0 is the cut off frequency as before



Ideal High Pass Filters (cont…)

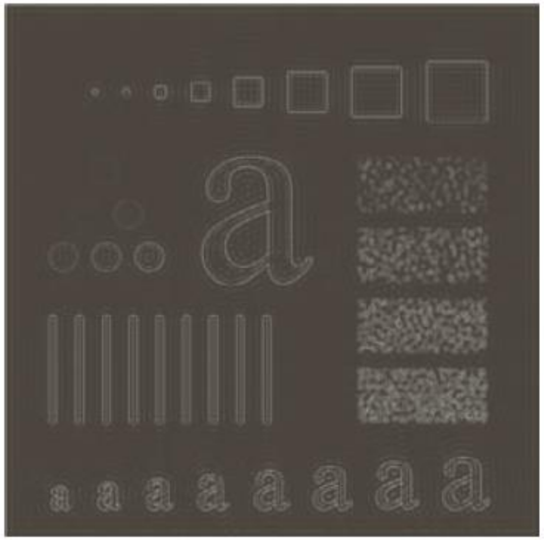


Results of ideal high pass filtering



$D_0 = 30$

$D_0 = 60$



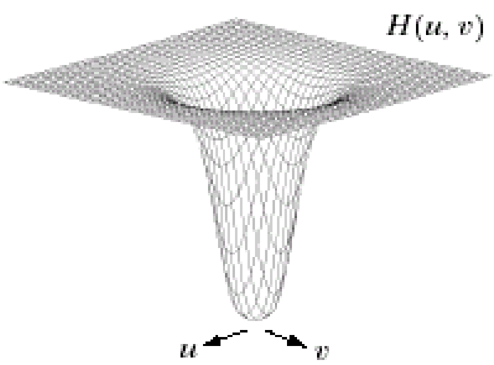
$D_0 = 160$

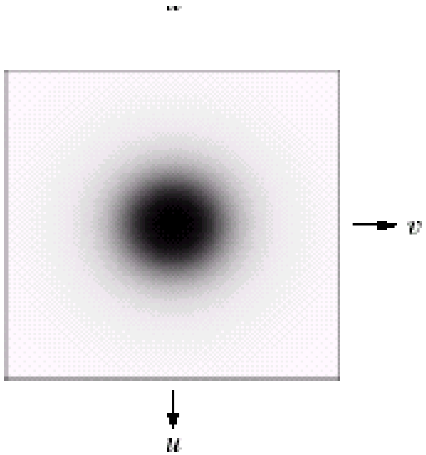
Butterworth High Pass Filters

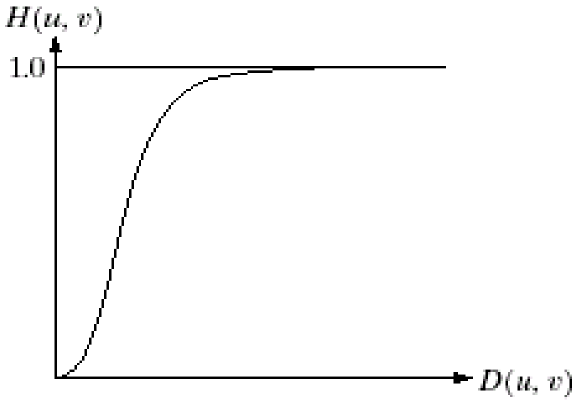
The Butterworth high pass filter is given as:

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

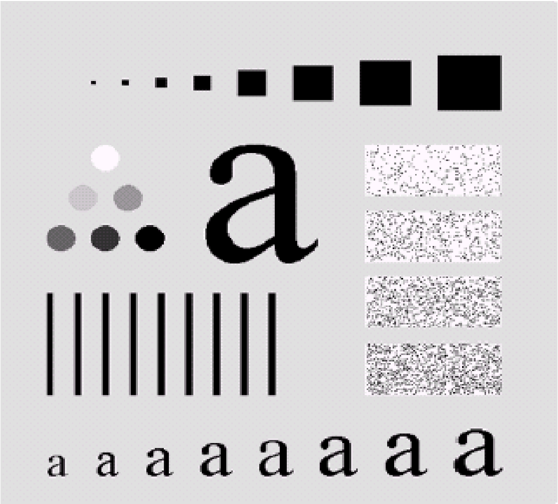
where n is the order and D_0 is the cut off frequency as before







Butterworth High Pass Filters (cont…)



Results of Butterworth high pass filtering of order 2

D_0 = 30

D_0 = 60

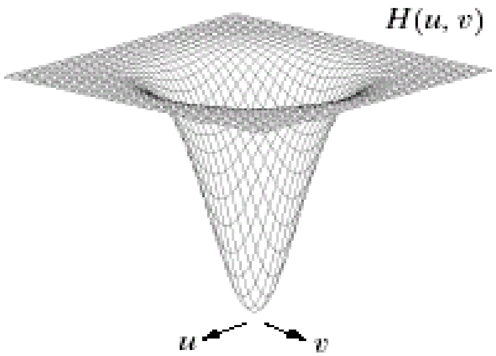
D_0 = 160

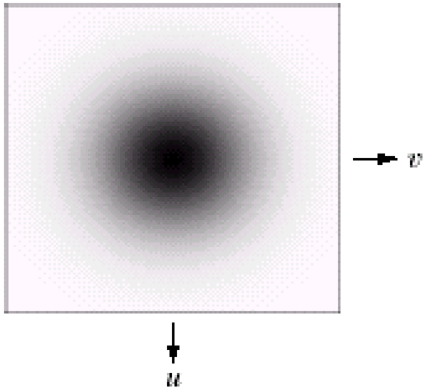
Gaussian High Pass Filters

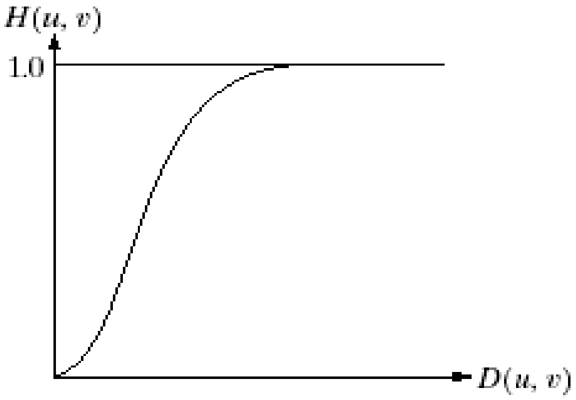
The Gaussian high pass filter is given as:

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

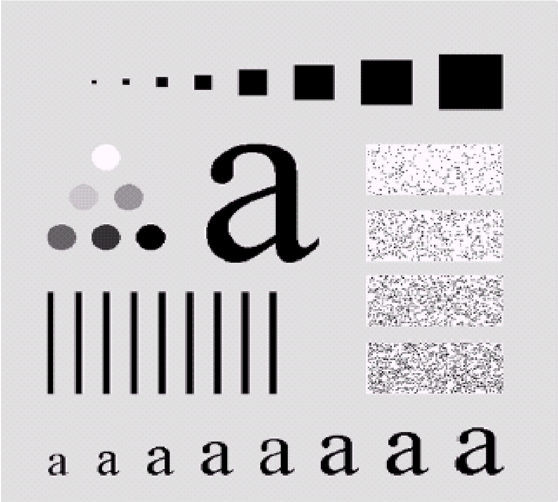
where D_0 is the cut off distance as before







Gaussian High Pass Filters (cont…)



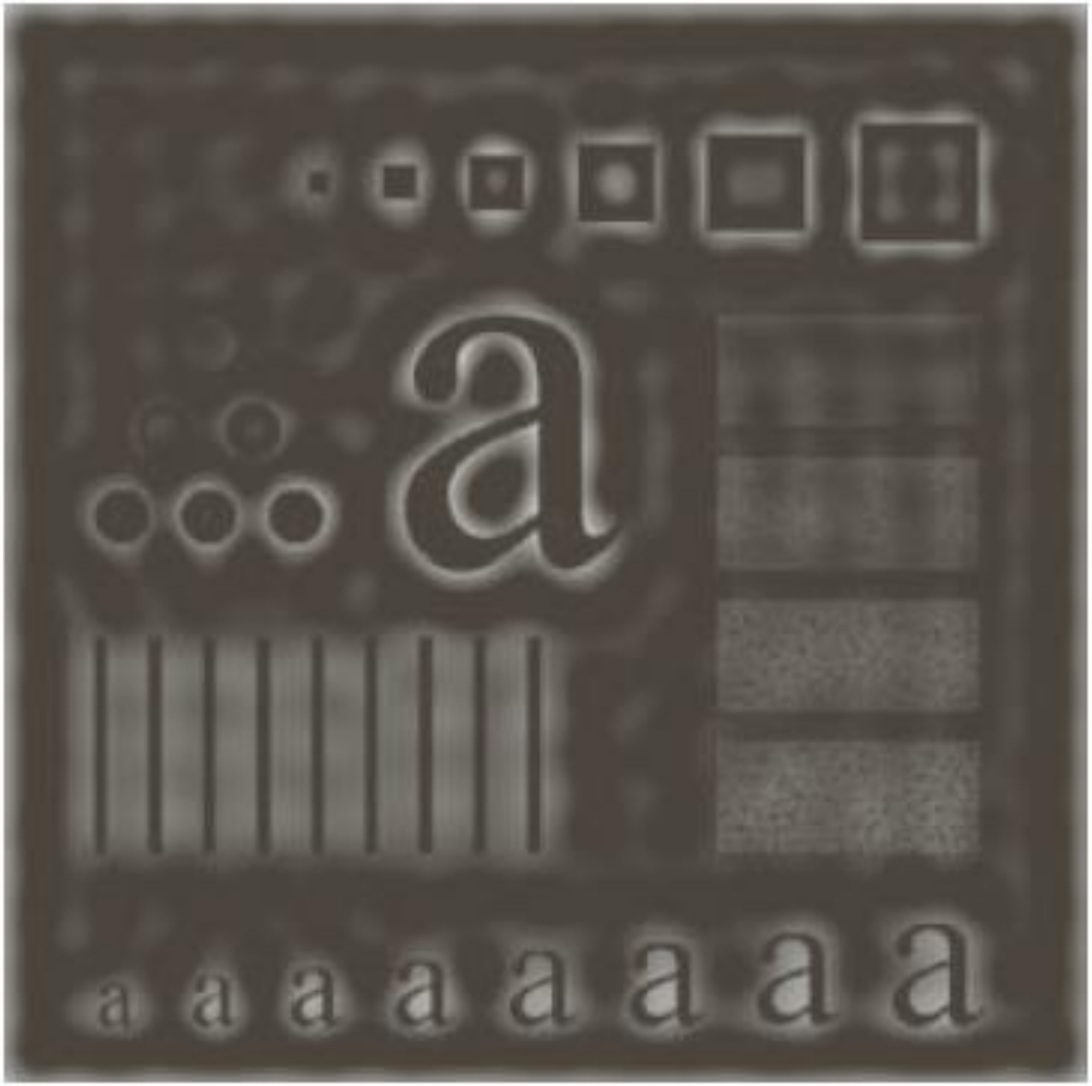
Results of Gaussian high pass filtering

D_0 = 30

D_0 = 60

D_0 = 160

Highpass Filter Comparison



Results of ideal high pass filtering with D_0 = 30

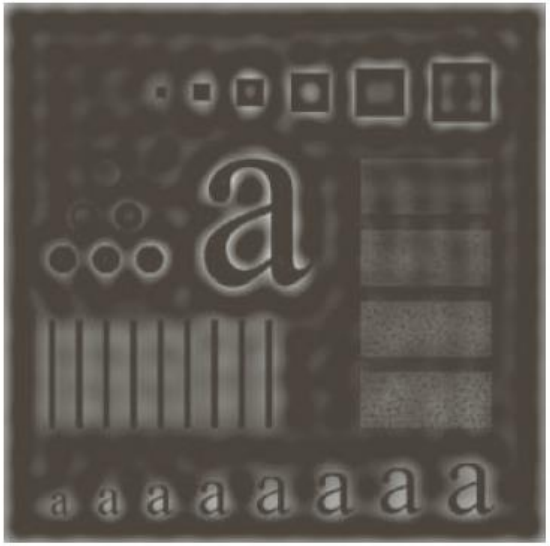
Highpass Filter Comparison

Results of Butterworth high pass filtering of order 2 with D_0 = 30

Highpass Filter Comparison

Results of Gaussian high pass filtering with D_0 = 30

Highpass Filter Comparison



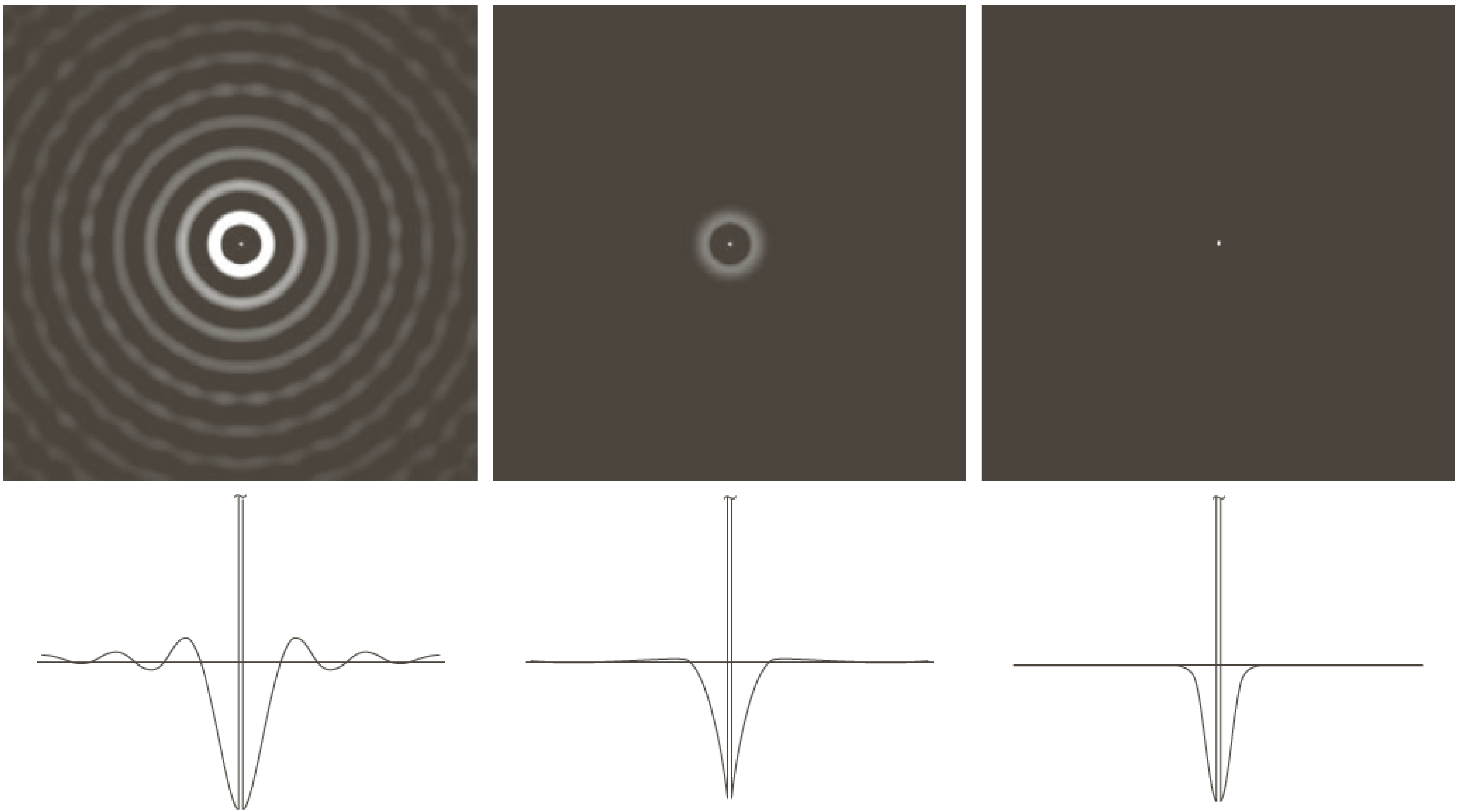
Results of ideal high pass filtering with $D_0 = 30$

Results of Butterworth high pass filtering of order 2 with $D_0 = 30$

Results of Gaussian high pass filtering with $D_0 = 30$

Highpass Filter Comparison

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D_0/D(u,v)]^{2n}}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$



Application Example of HPF

Enhancement of thumb print ridges and reduction of smudges:

Butterworth HPF + Thresholding

Laplacian In The Frequency Domain

- Assignment 4.52

$$H(u,v) = -4\pi^2(u^2 + v^2)$$

- Laplacian image

$$\nabla^2 f(x, y) = \Im^{-1}\{H(u, v)F(u, v)\}$$

- Image sharpening

$$g(x, y) = f(x, y) + c\nabla^2 f(x, y)$$

$c = -1$ because $H(u, v)$ is negative.

Laplacian In The Frequency Domain

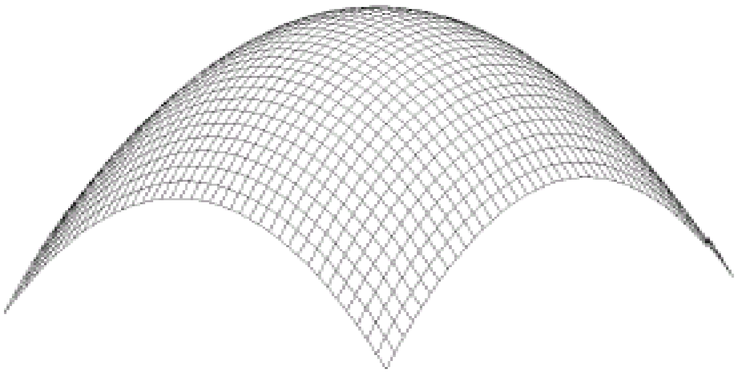
- Image sharpening

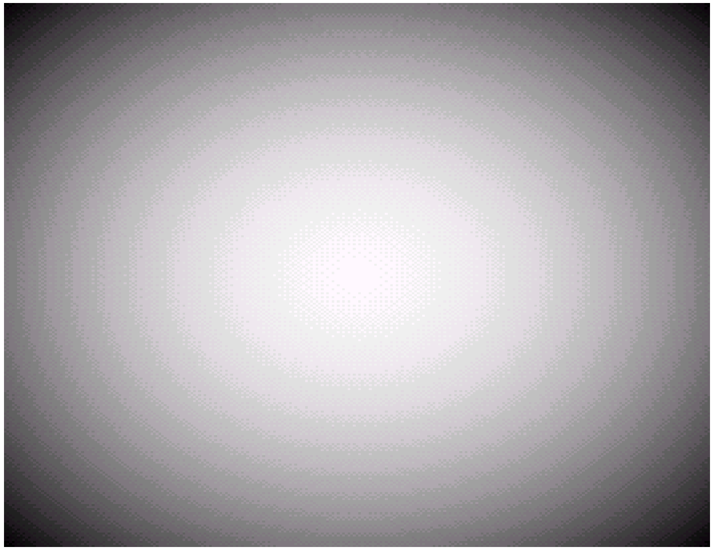
$$g(x, y) = f(x, y) + c\nabla^2 f(x, y)$$

$$\begin{aligned} g(x, y) &= \Im^{-1}\left\{F(u, v) - H(u, v)F(u, v)\right\} \\ &= \Im^{-1}\left\{\left[1 - H(u, v)\right]F(u, v)\right\} \\ &= \Im^{-1}\left\{\left[1 + 4\pi^2 D^2(u, v)\right]F(u, v)\right\} \end{aligned}$$

Laplacian In The Frequency Domain

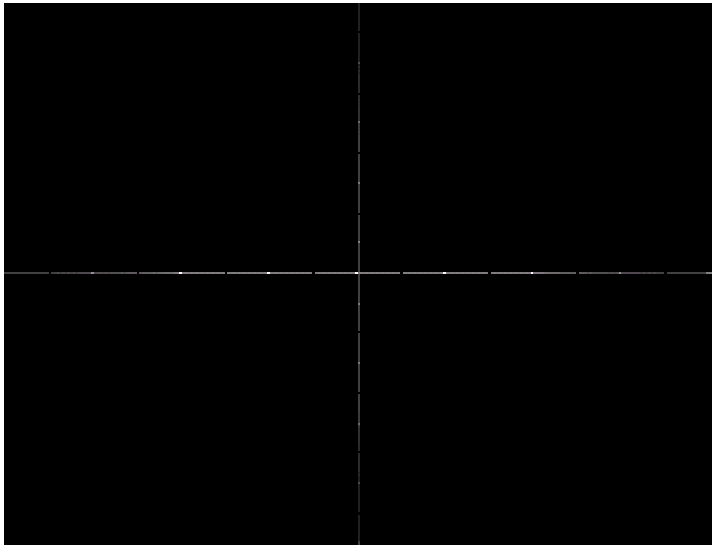
Laplacian in the frequency domain

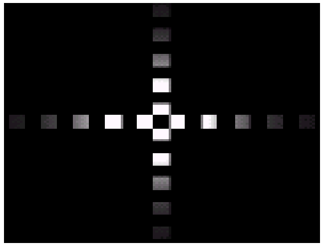


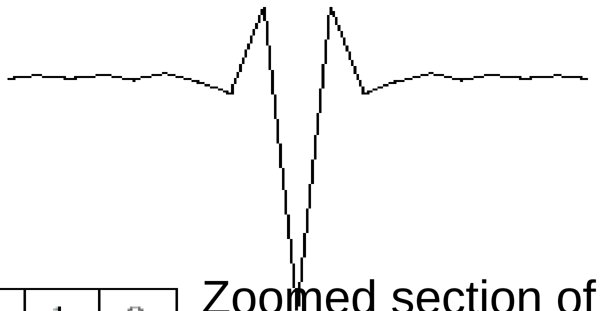


2-D image of Laplacian in the frequency domain

Inverse DFT of Laplacian in the image domain







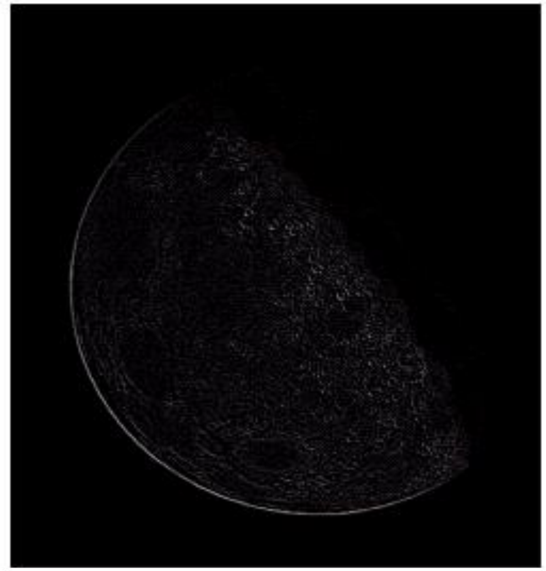
0	1	0
1	−4	1
0	1	0

Zoomed section of the image on the left compared to spatial filter

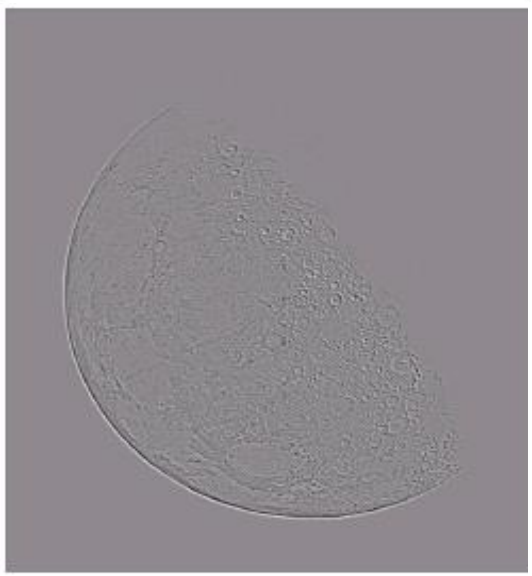
Frequency Domain Laplacian Example

Original image

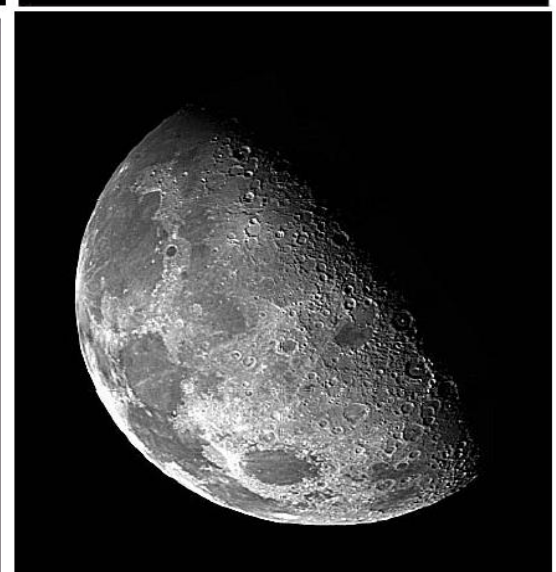
Laplacian filtered image



Laplacian image scaled



Enhanced image



High-frequency Emphasis Filter

$$g_{\mathrm{mask}}(x, y) = f(x, y) - f_{\mathrm{LP}}(x, y)$$

$$f_{\mathrm{LP}}(x, y) = \Im^{-1}\left[H_{\mathrm{LP}}(u, v)F(u, v)\right]$$

$$g(x, y) = f(x, y) + k * g_{\mathrm{mask}}(x, y)$$

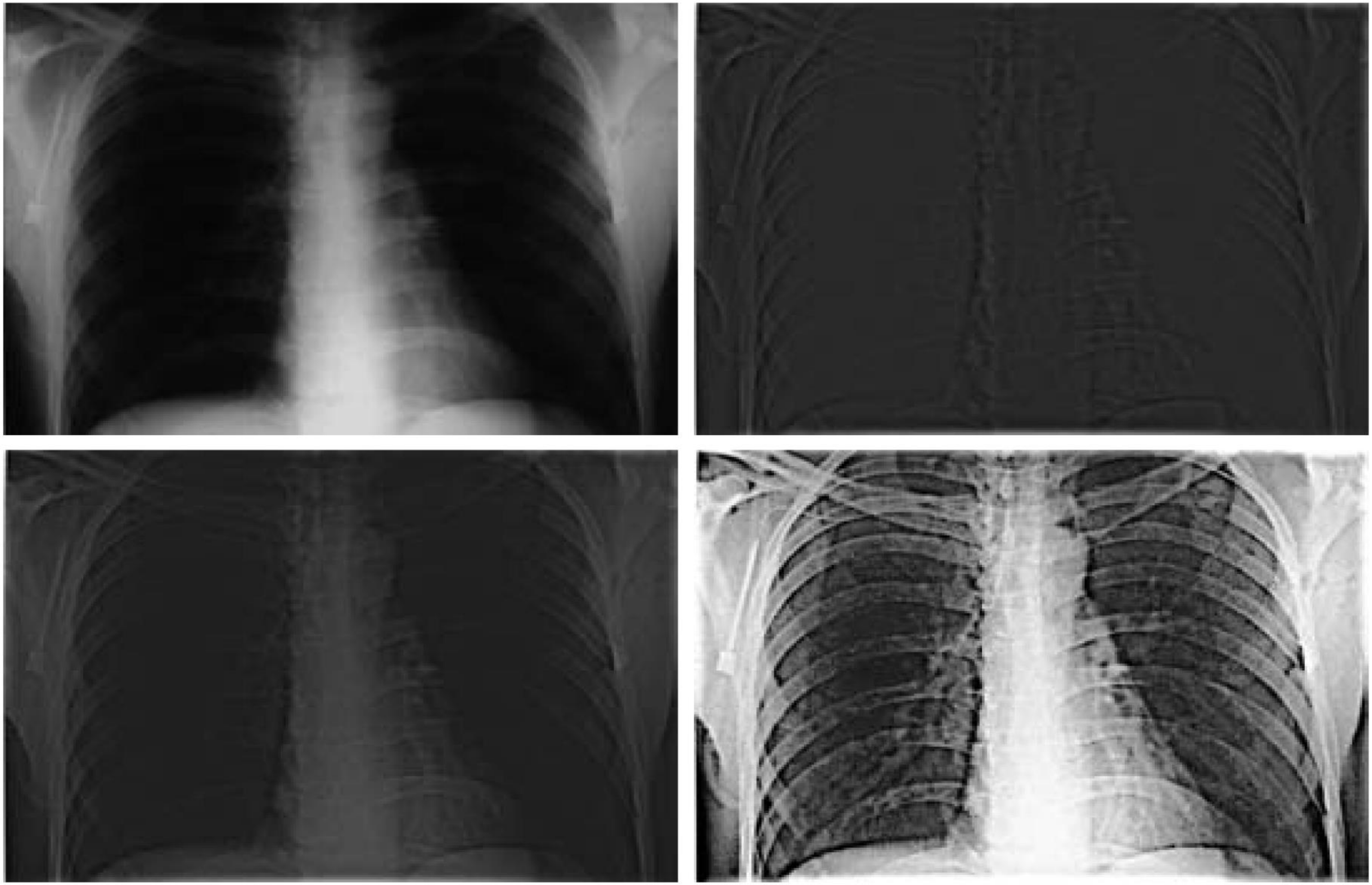
$$g(x, y) = \Im^{-1}\left\{\left[1 + k * [1 - H_{\mathrm{LP}}(u, v)]\right]F(u, v)\right\}$$

$$g(x, y) = \Im^{-1}\left\{[1 + k * H_{\mathrm{HP}}(u, v)]F(u, v)\right\}$$

- General formulation

$$g(x, y) = \Im^{-1}\left\{[k_1 + k_2 * H_{\mathrm{HP}}(u, v)]F(u, v)\right\}$$

High-frequency Emphasis Filtering



Original image

After Gaussian HPF

High freq. emphasis result

After histogram equalisation

Homomorphic Filtering

- Illumination-reflectance model

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

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

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

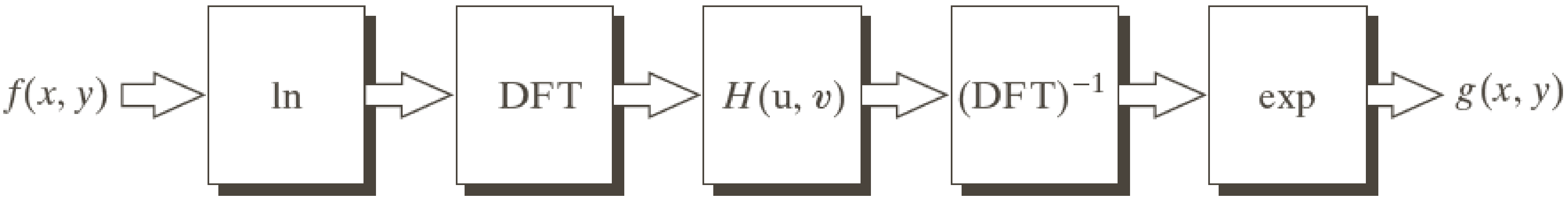
Homomorphic Filtering (cont.)

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

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

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

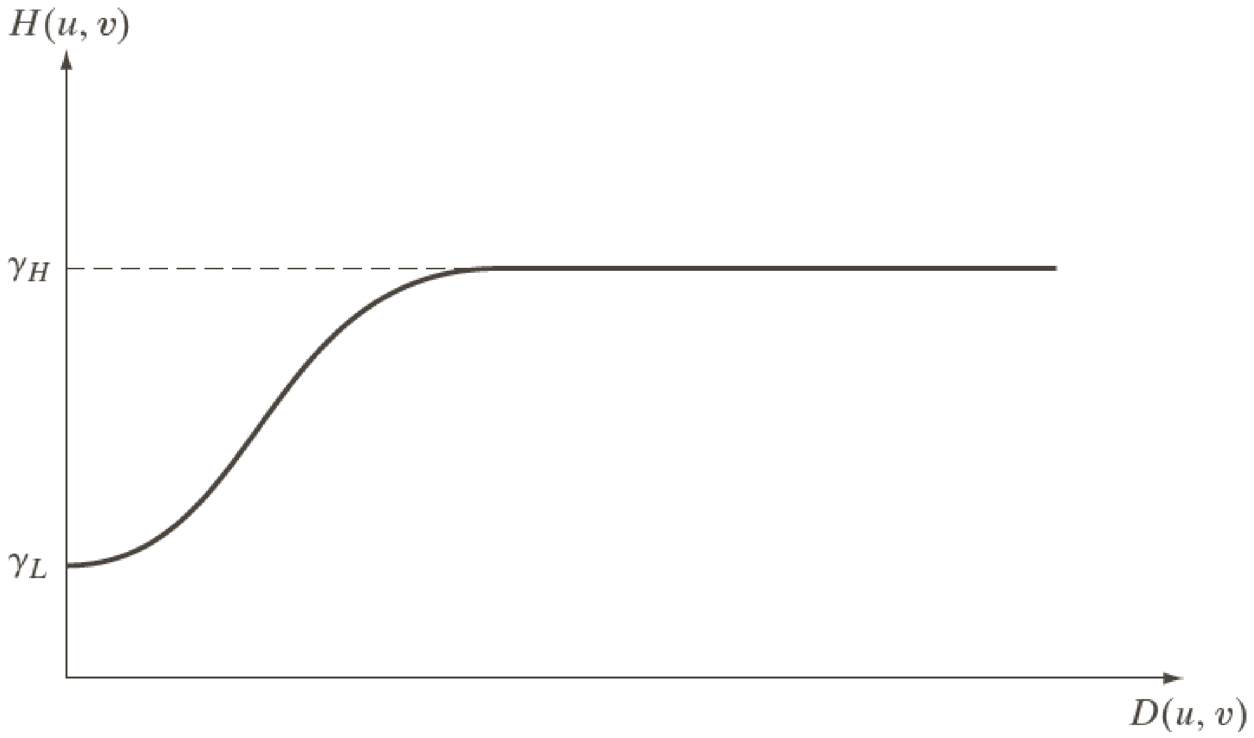
$$= \Im^{-1}\{H(u,v)F_i(u,v)\} + \Im^{-1}\{H(u,v)F_r(u,v)\}$$



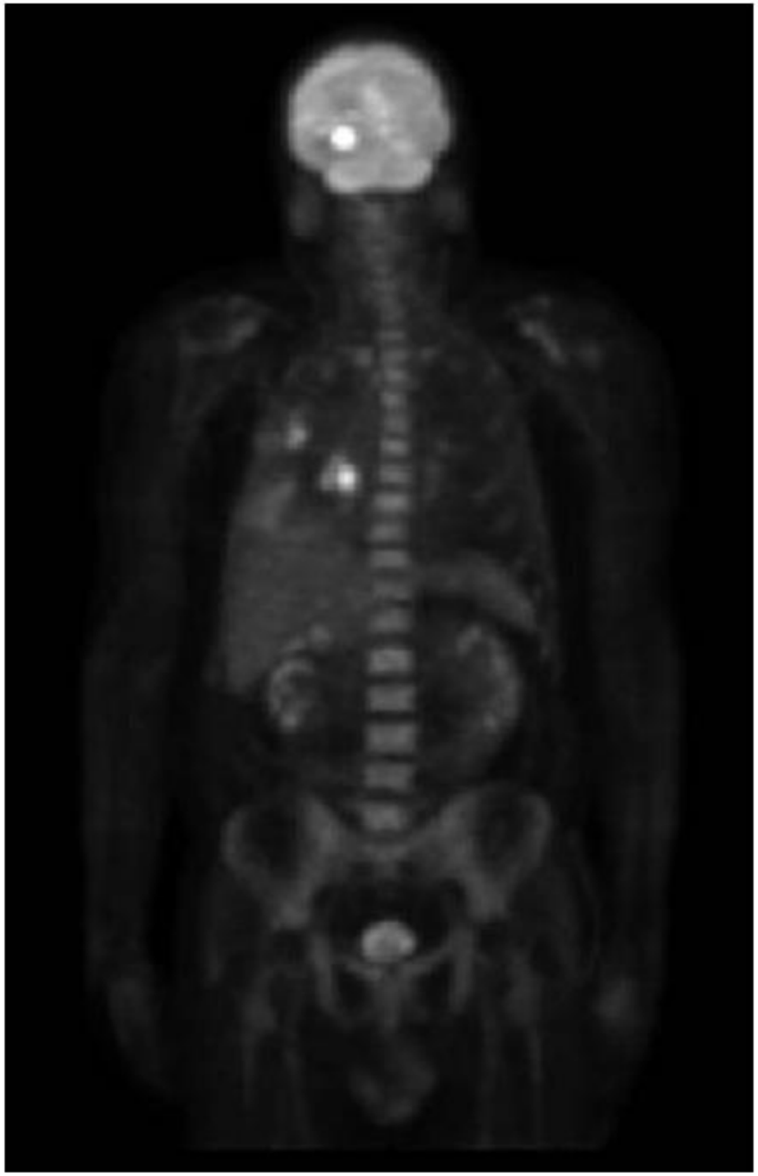
Homomorphic Filtering (cont.)

- Simultaneous dynamic range compression & contrast enhancement

$$H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c[D^2(u,v)/D_0^2]}\right] + \gamma_L$$



Homomorphic Filtering (PET)



Selective Filtering

- Bandreject filters

TABLE 4.6
Bandreject filters. W is the width of the band, D is the distance $D(u, v)$ from the center of the filter, D_0 is the cutoff frequency, and n is the order of the Butterworth filter. We show D instead of $D(u, v)$ to simplify the notation in the table.

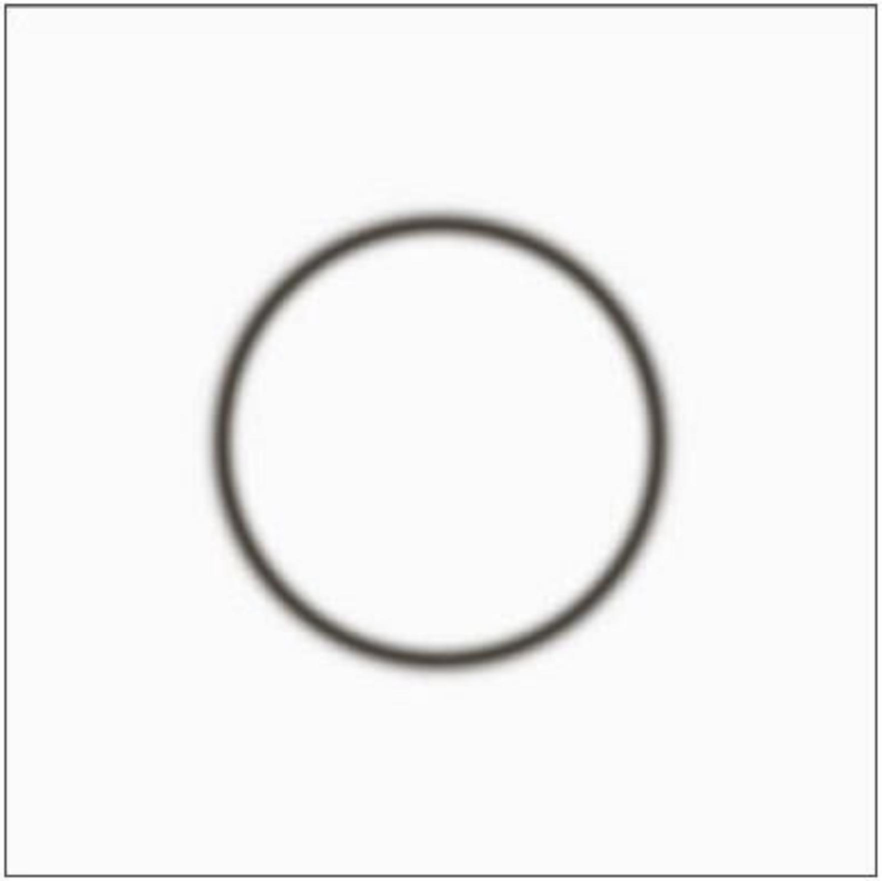
Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

- Bandpass filters

$$H_{\mathrm{BP}}(u,v) = 1 - H_{\mathrm{BR}}(u,v)$$

Selective Filtering

- Bandreject and bandpass Gaussian filter



Selective Filtering

A Notch filter rejects (or passes) frequencies in a predefined neighborhood about the center of the frequency rectangle

$$H_{\mathrm{NR}}(u,v) = \prod_{k=1}^{Q} H_k(u,v)H_{-k}(u,v)$$

$$H_{\mathrm{NP}}(u,v) = 1 - H_{\mathrm{NR}}(u,v)$$

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

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

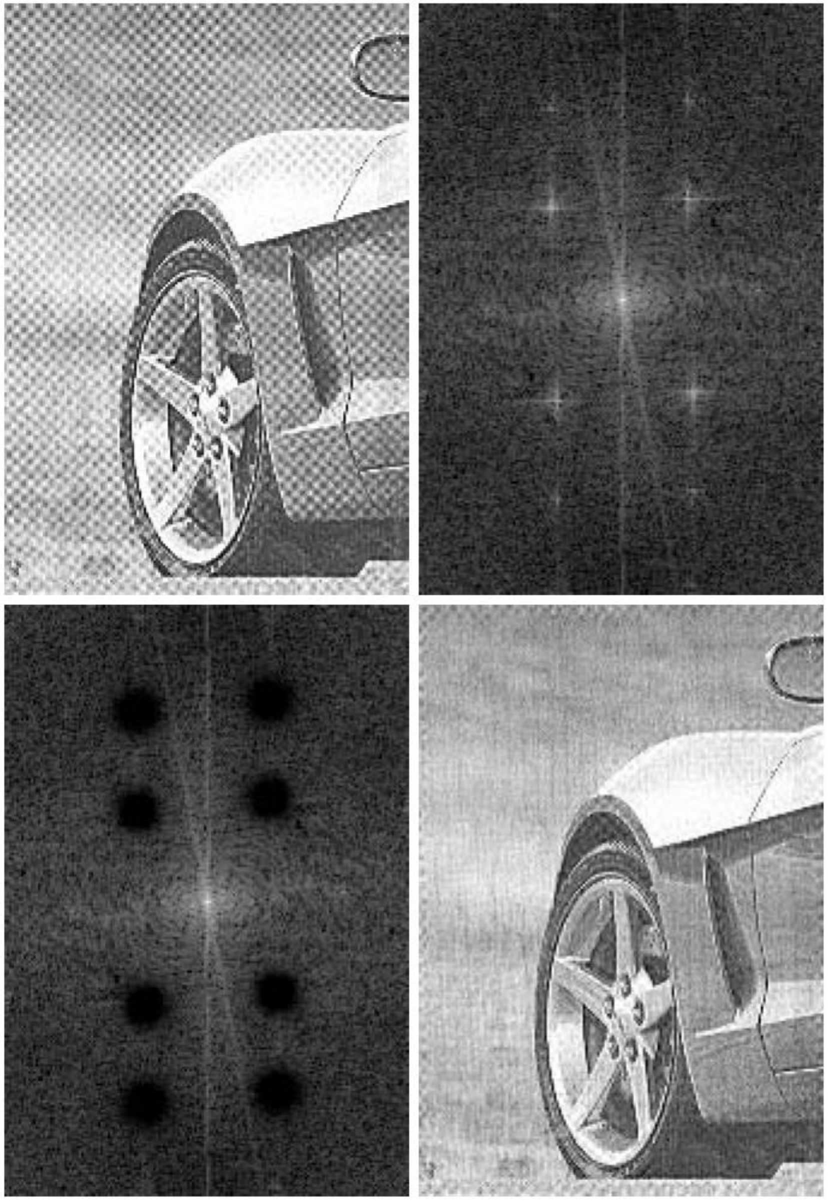
Why symmetric?

- Zero-phase-shift filters must be symmetric about the origin.
- Hermitian symmetry for real signals

$$H(u,v)=\frac{1}{WH}\sum_{y=0}^{H-1}\sum_{x=0}^{W-1}\left(h(x,y)e^{-(\frac{j2\pi ux}{W}+\frac{j2\pi vy}{H})}\right)$$

$$H^{*}(u,v)=\frac{1}{WH}\sum_{y=0}^{H-1}\sum_{x=0}^{W-1}\left(h(x,y)e^{(\frac{j2\pi ux}{W}+\frac{j2\pi vy}{H})}\right)=H(-u,-v)$$

Notch Filtering Example



a	b
c	d

FIGURE 4.64
(a) Sampled newspaper image showing a moiré pattern.
(b) Spectrum.
(c) Butterworth notch reject filter multiplied by the Fourier transform.
(d) Filtered image.

Notch Filtering Example

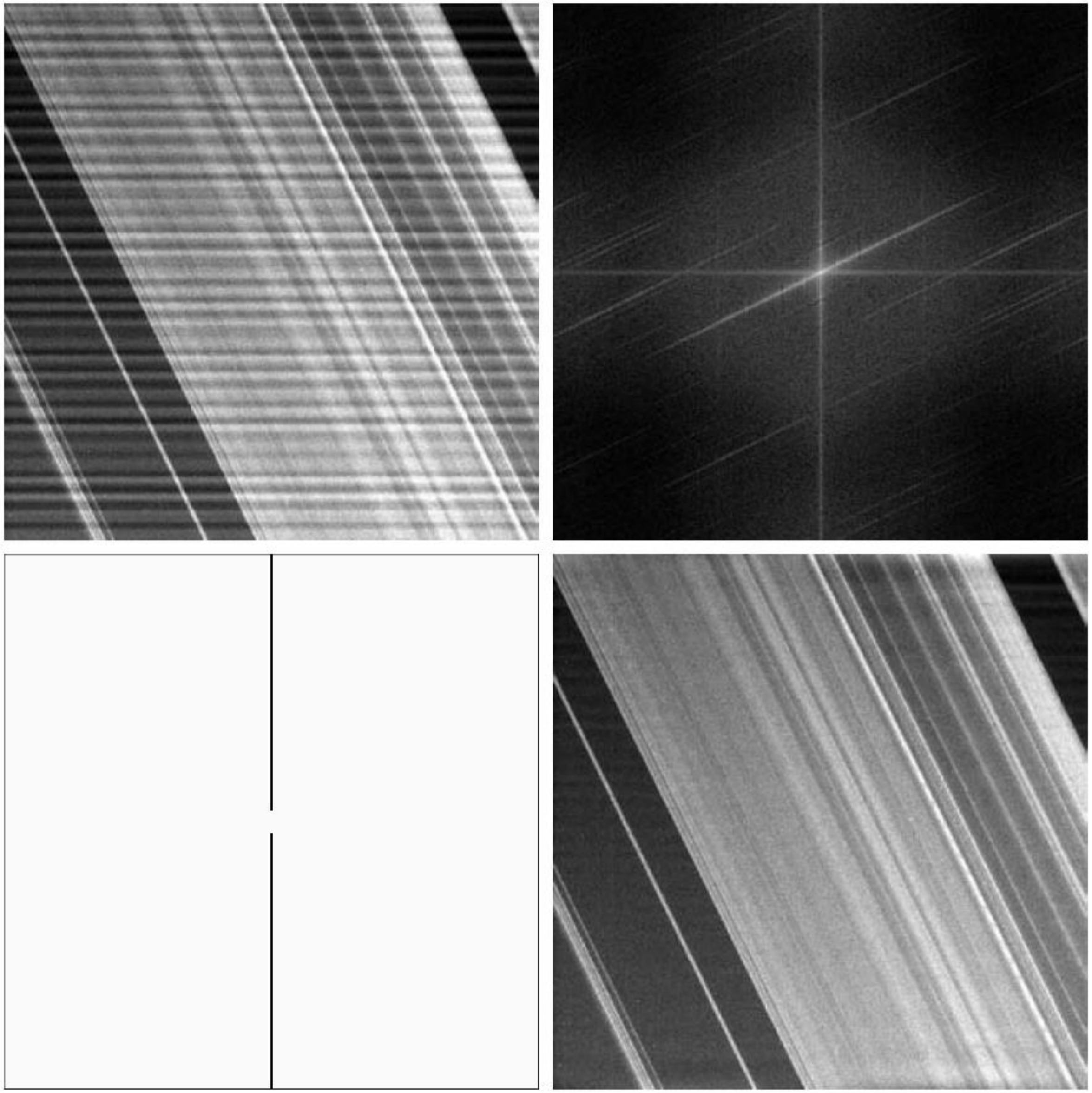
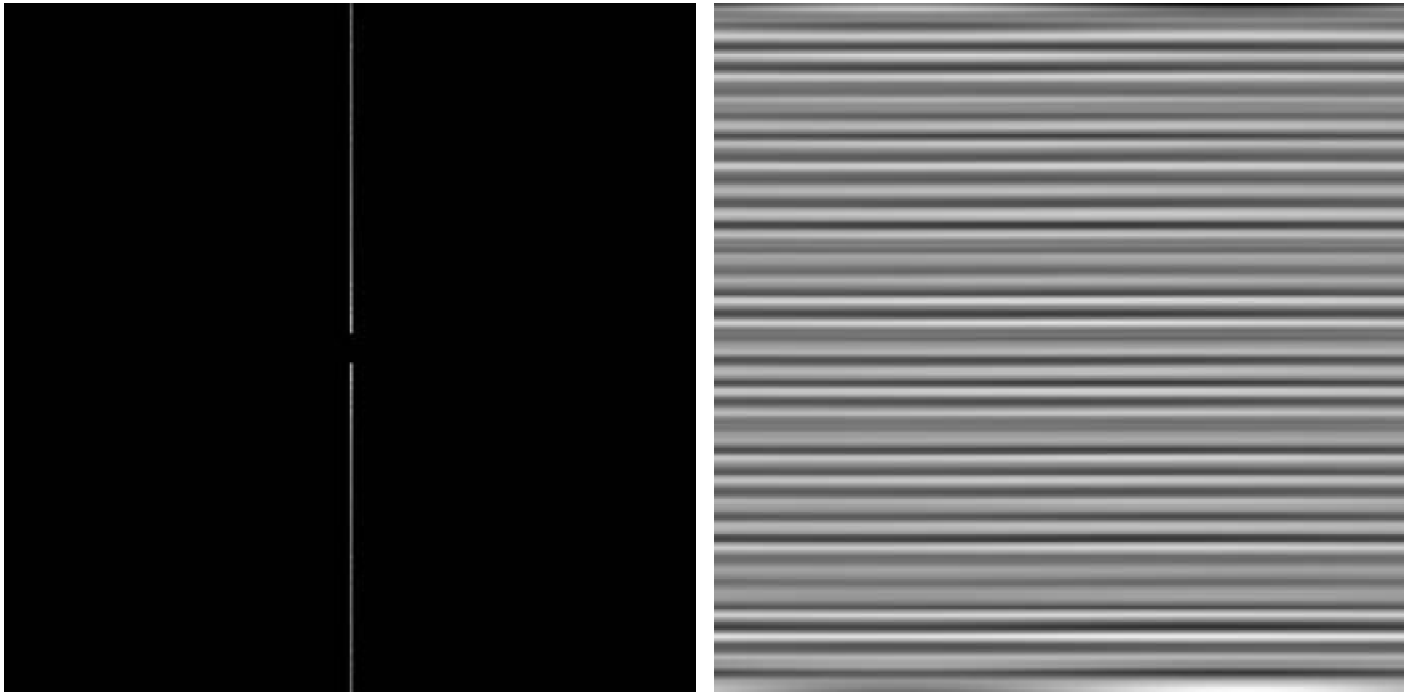


FIGURE 4.65 (a) 674 × 674 image of the Saturn rings showing nearly periodic interference. (b) Spectrum: The bursts of energy in the vertical axis near the origin correspond to the interference pattern. (c) A vertical notch reject filter. (d) Result of filtering. The thin black border in (c) was added for clarity; it is not part of the data. (Original image courtesy of Dr. Robert A. West, NASA/JPL.)

Notch Filtering Example



a b

FIGURE 4.66
(a) Result (spectrum) of applying a notch pass filter to the DFT of Fig. 4.65(a). (b) Spatial pattern obtained by computing the IDFT of (a).

Implementation

- Separability of the 2-D DFT

$$F(u,v) = \sum_{x=0}^{M-1} e^{-j2\pi ux/M} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi vy/N}$$

$$= \sum_{x=0}^{M-1} F(x,v)\, e^{-j2\pi ux/M}$$

where

$$F(x,v) = \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi vy/N}$$

Implementation

- DFT

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

- IDFT

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$$

- Computing the IDFT using DFT

$$MNf^{*}(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F^{*}(u,v)e^{-j2\pi(ux/M+vy/N)}$$

Fast Fourier Transform

The reason that Fourier based techniques have become so popular is the development of the *Fast Fourier Transform (FFT)* algorithm

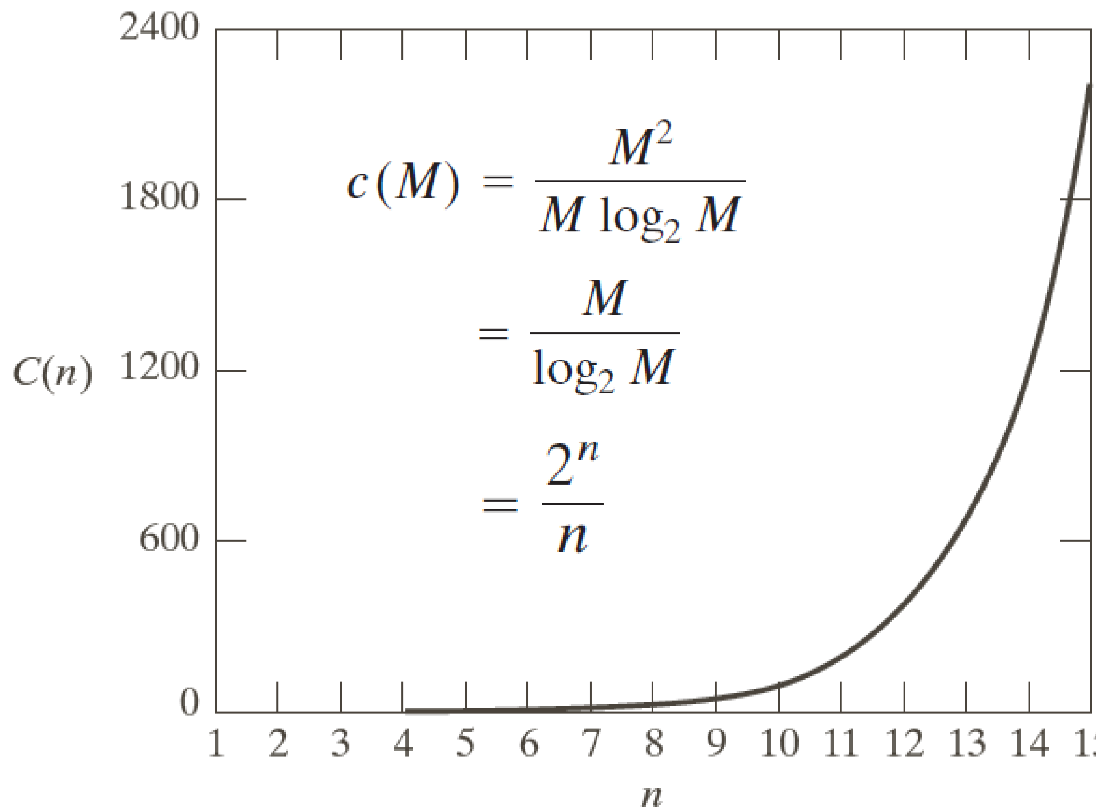


FIGURE 4.67 Computational advantage of the FFT over a direct implementation of the 1-D DFT. Note that the advantage increases rapidly as a function of n.

Frequency Domain Filtering & Spatial Domain Filtering

Similar jobs can be done in the spatial and frequency domains

Filtering in the spatial domain can be easier to understand

Filtering in the frequency domain can be much faster – especially for large images

2-D Convolution Theorem

$$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n)$$

$$f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$$

$$f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$$

Avoid wraparound error by padding with 0

$$f_p(x, y) = \begin{cases} f(x, y) & 0 \le x \le A-1 \quad \text{and} \quad 0 \le y \le B-1 \\ 0 & A \le x \le P \quad \text{or} \quad B \le y \le Q \end{cases}$$

and

$$h_p(x, y) = \begin{cases} h(x, y) & 0 \le x \le C-1 \quad \text{and} \quad 0 \le y \le D-1 \\ 0 & C \le x \le P \quad \text{or} \quad D \le y \le Q \end{cases}$$

with

$$P \ge A + C - 1$$

and

$$Q \ge B + D - 1$$

Frequency Domain Filtering & Spatial Domain Filtering

$$T = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Spatial filter

$$\begin{aligned} g(x,y) = 9f(x,y) - f(x-1,y-1) - f(x-1,y) - f(x-1,y+1) - f(x,y-1) \\ - f(x,y+1) - f(x+1,y-1) - f(x+1,y) - f(x+1,y+1) \end{aligned}$$

Impulse response function:

$$\begin{aligned} h(x,y) = 9\delta(x,y) - \delta(x-1,y-1) - \delta(x-1,y) - \delta(x-1,y+1) - \delta(x,y-1) \\ - \delta(x,y+1) - \delta(x+1,y-1) - \delta(x+1,y) - \delta(x+1,y+1) \end{aligned}$$

Frequency Domain Filtering & Spatial Domain Filtering

$$T = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Filter in the frequency domain:

$$H(u,v) = \frac{1}{MN}(9 - e^{-j\frac{2\pi v}{N}} - e^{j\frac{2\pi v}{N}} - e^{-j\frac{2\pi u}{M}} - e^{j\frac{2\pi u}{M}} - (e^{-j\frac{2\pi u}{M}} + e^{j\frac{2\pi u}{M}})(e^{-j\frac{2\pi v}{N}} + e^{j\frac{2\pi v}{N}}))$$

$$= \frac{1}{MN}(9 - 2\cos(\frac{2\pi v}{N}) - 2\cos(\frac{2\pi u}{M}) - 4\cos(\frac{2\pi u}{M})\cos(\frac{2\pi v}{N}))$$

$$= \frac{1}{MN}(10 - (1 + 2\cos(\frac{2\pi v}{N}))(1 + 2\cos(\frac{2\pi u}{M}))$$

High-boost filtering → Sharpening, Edge enhancement

Summary

In this lecture we examined image enhancement in the frequency domain

- The Fourier series & the Fourier transform
- Image Processing in the frequency domain
  - Image smoothing
  - Image sharpening
- Fast Fourier Transform

Next time we will begin to examine image restoration using the spatial and frequency based techniques we have been looking at

Assignments

- 4.10，4.26，4.41，4.42，4.43，4.44，4.47