

# Image Enhancement: Filtering in the Frequency Domain

## enhancement in the frequency domain

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

# Jean Baptiste Joseph Fourier

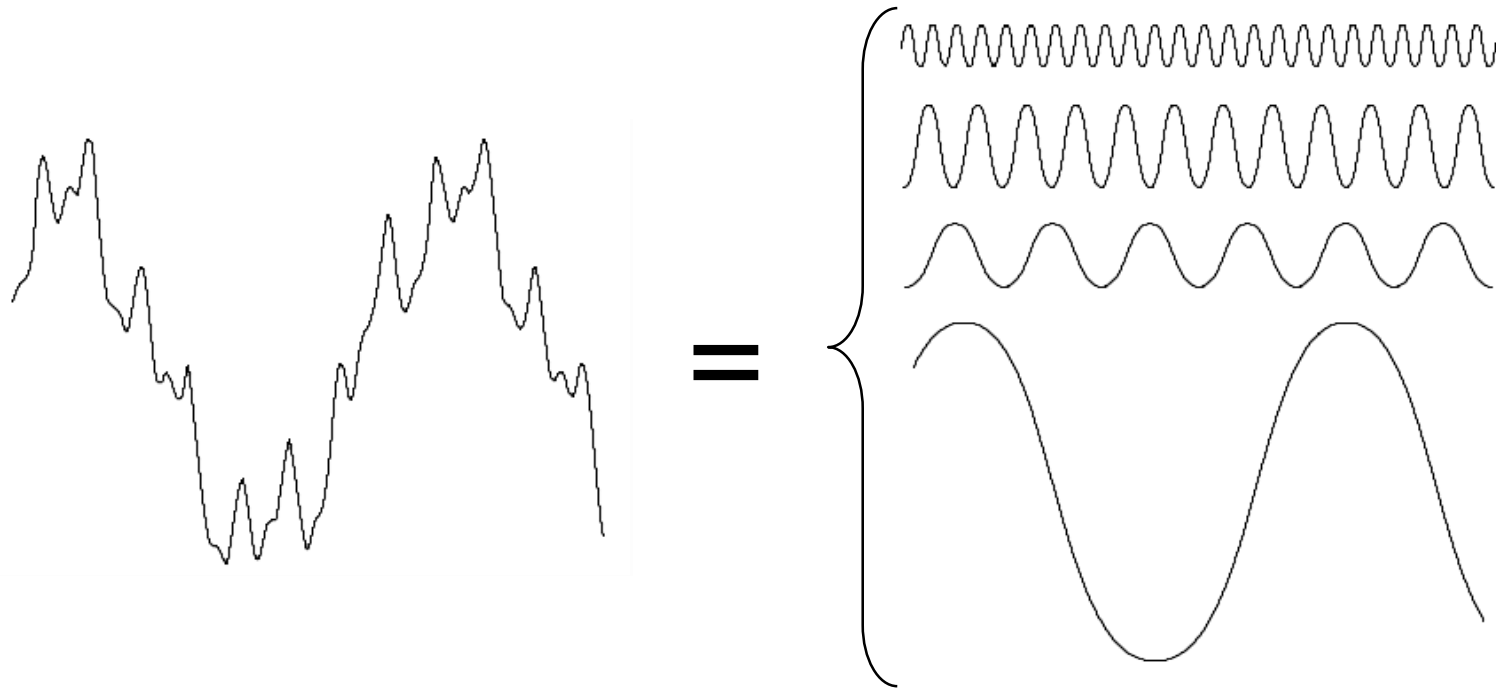


Fourier was born in Auxerre,  
France in 1768

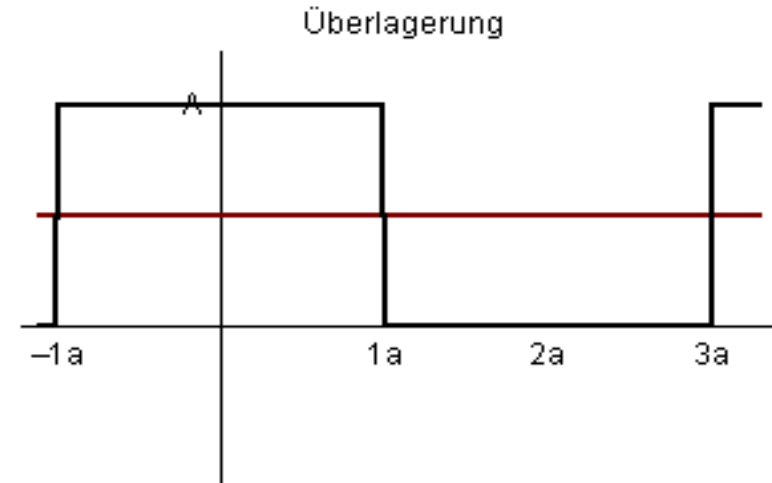
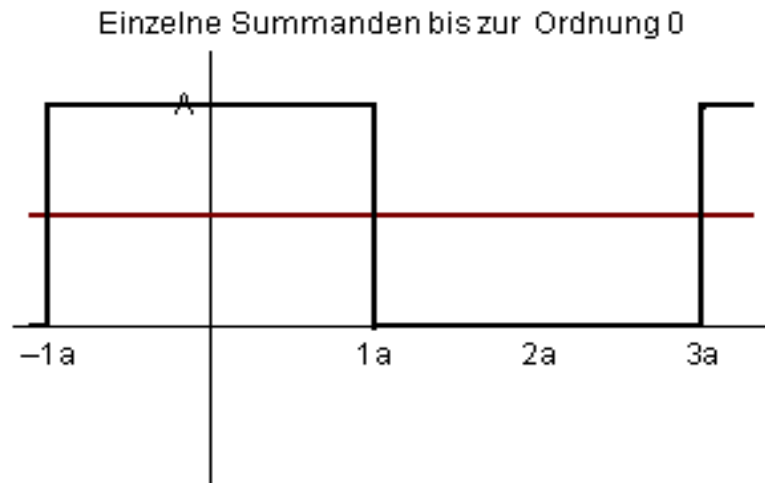
- Most famous for his work “*La Théorie Analytique de la Chaleur*” published in 1822
- Translated into English in 1878: “*The Analytic Theory of Heat*”

Nobody paid much attention when the work was first published

One of the most important mathematical theories in modern engineering



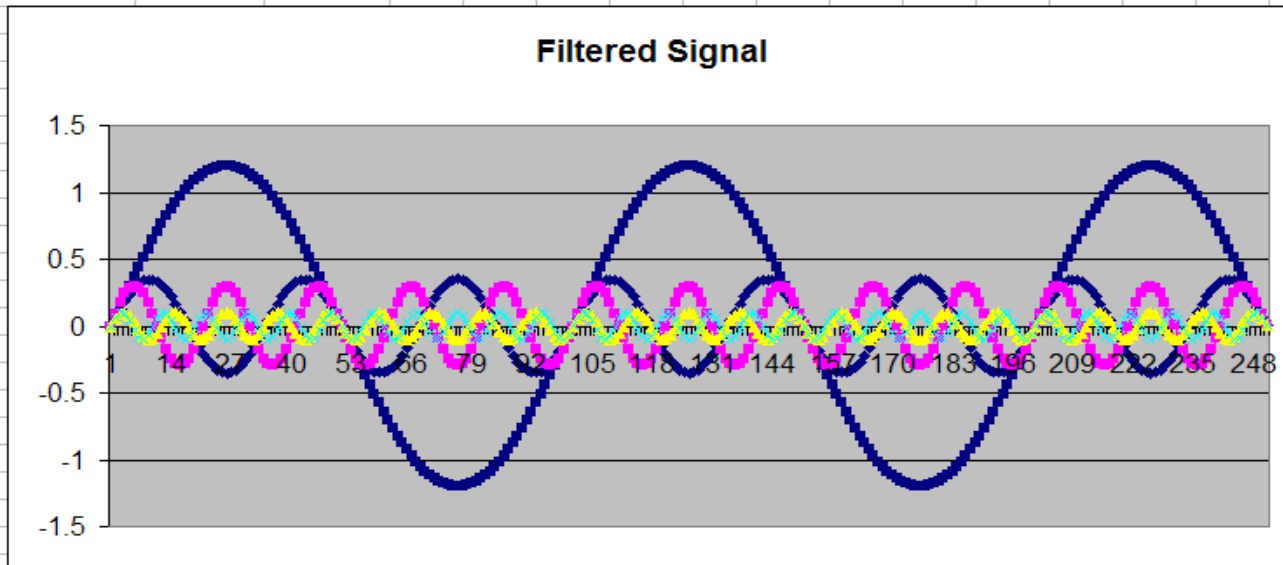
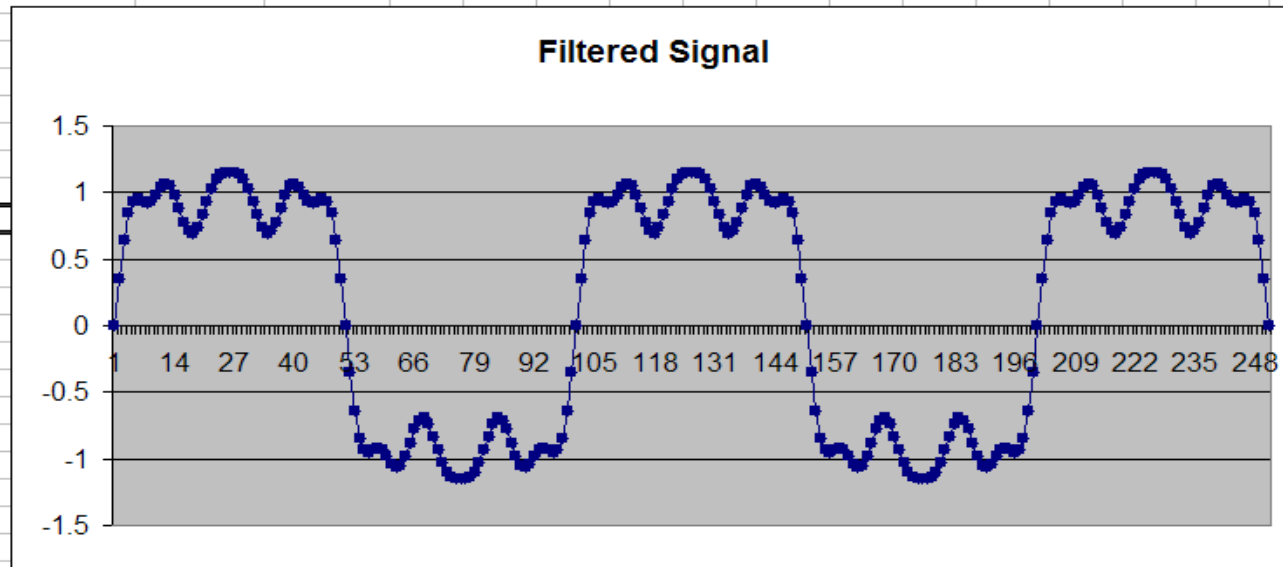
Any function that periodically repeats itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient – a *Fourier series*



Notice how we get closer and closer to the original function as we add more and more frequencies

# The Big Idea (cont...)

Frequency  
domain signal  
processing  
example in Excel



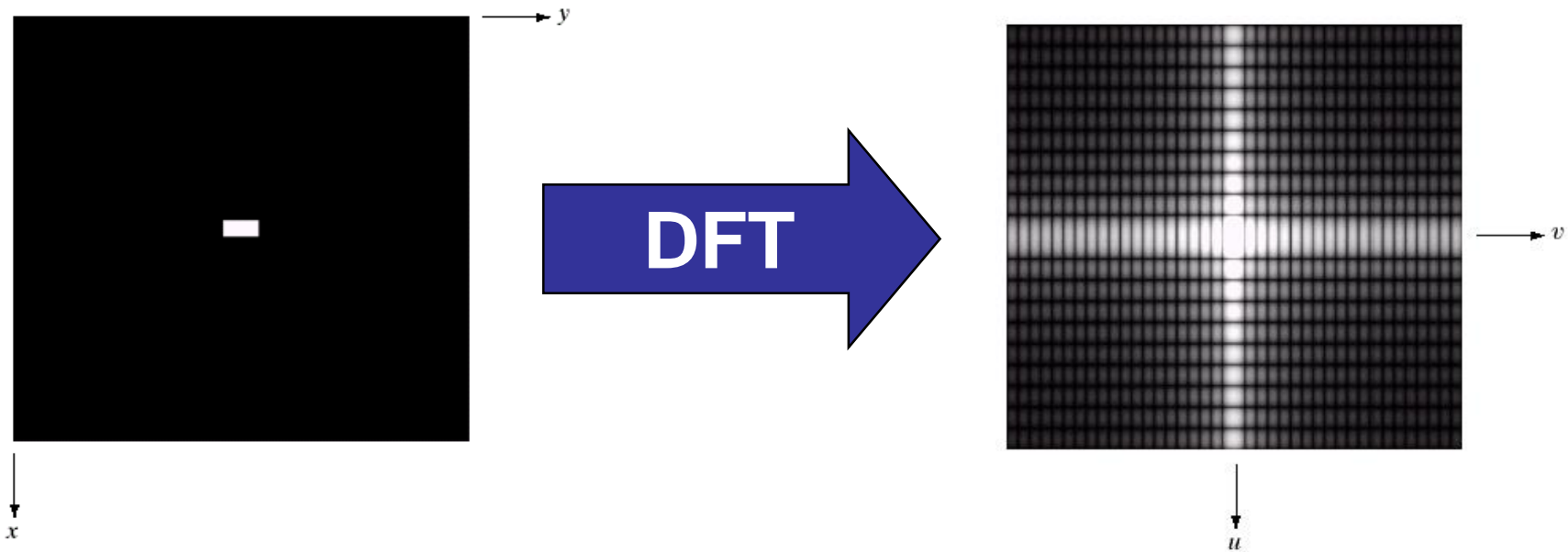
# The Discrete Fourier Transform (DFT)

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

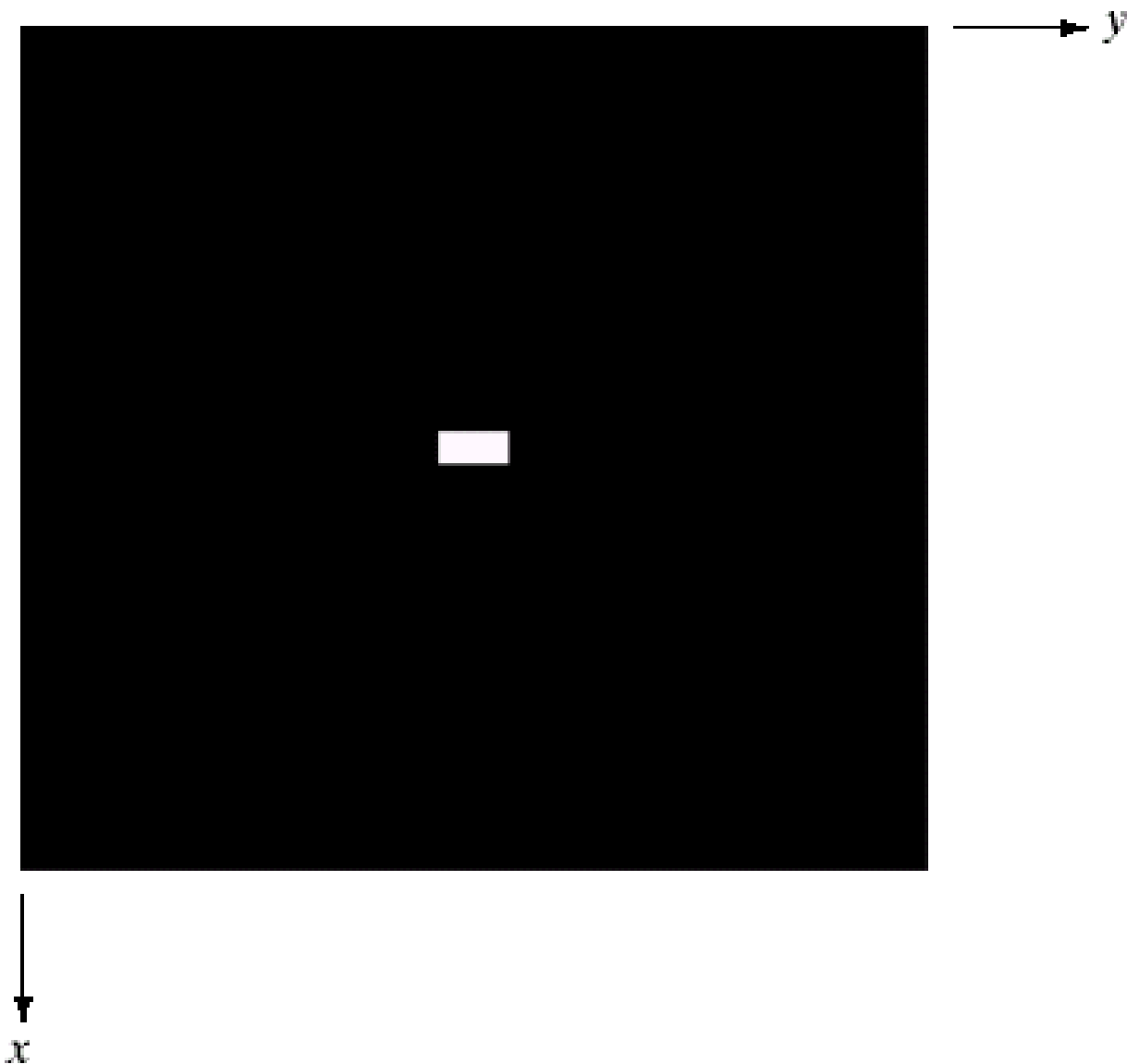
$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

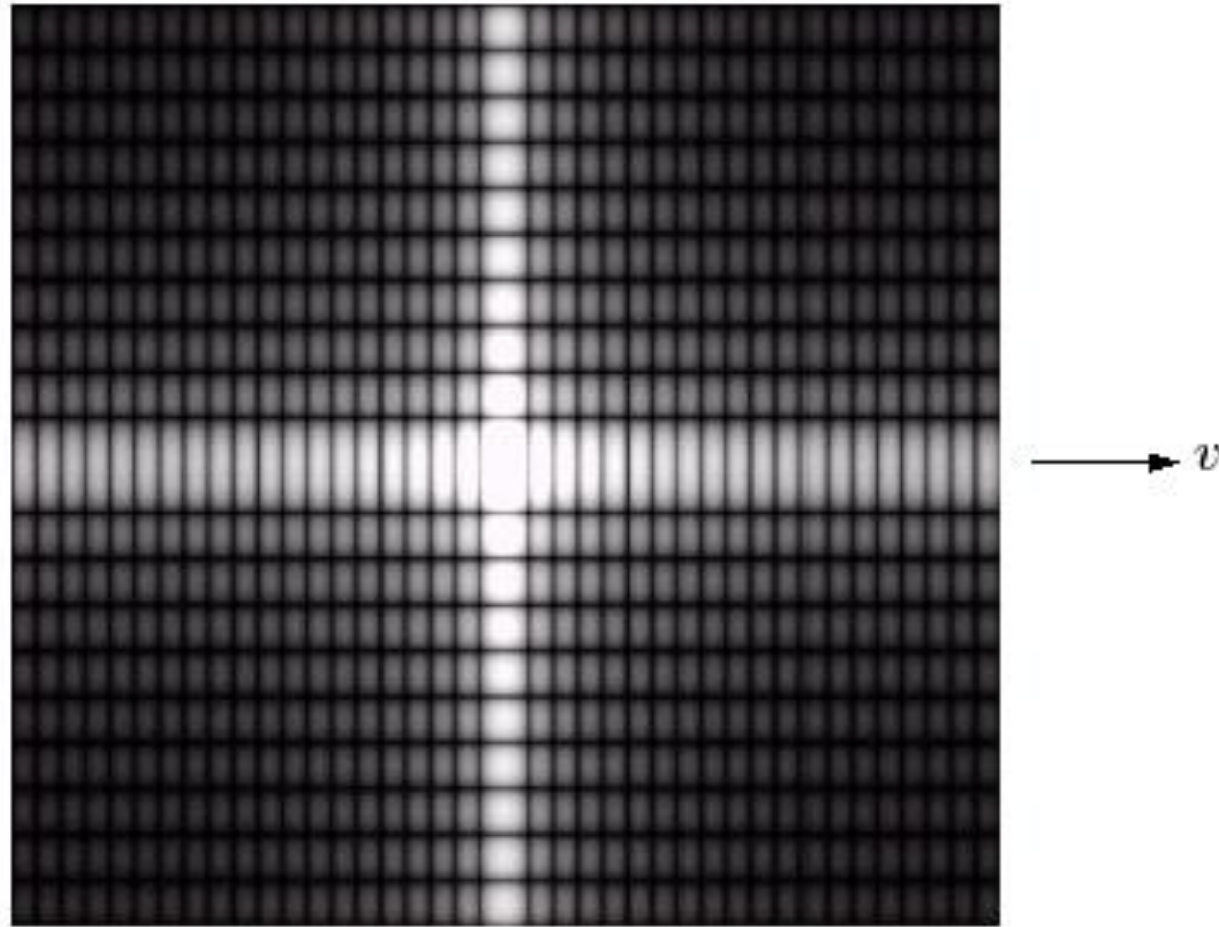
for  $u = 0, 1, 2 \dots M-1$  and  $v = 0, 1, 2 \dots N-1$ .

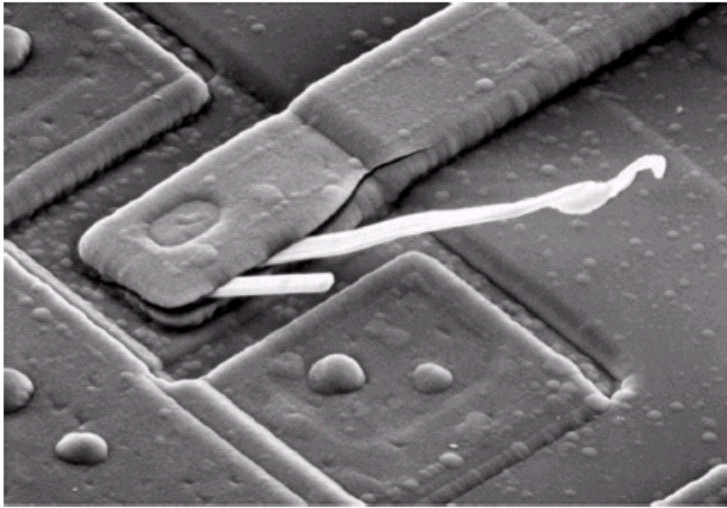
The DFT of a two dimensional image can be visualised by showing the spectrum of the images component frequencies



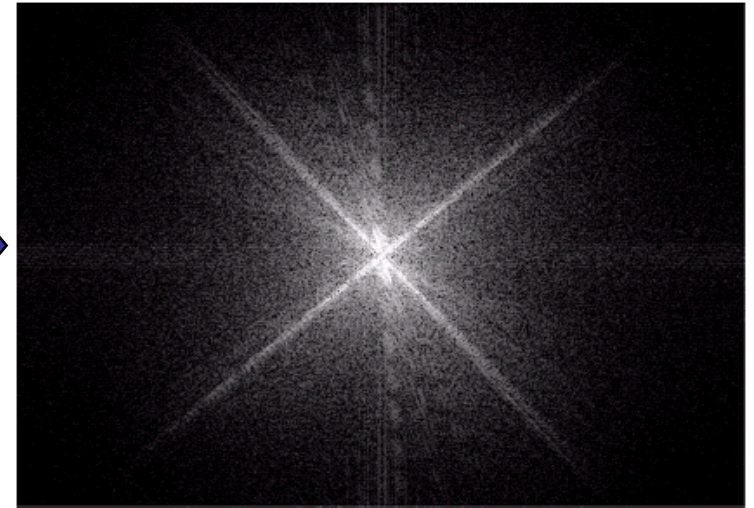




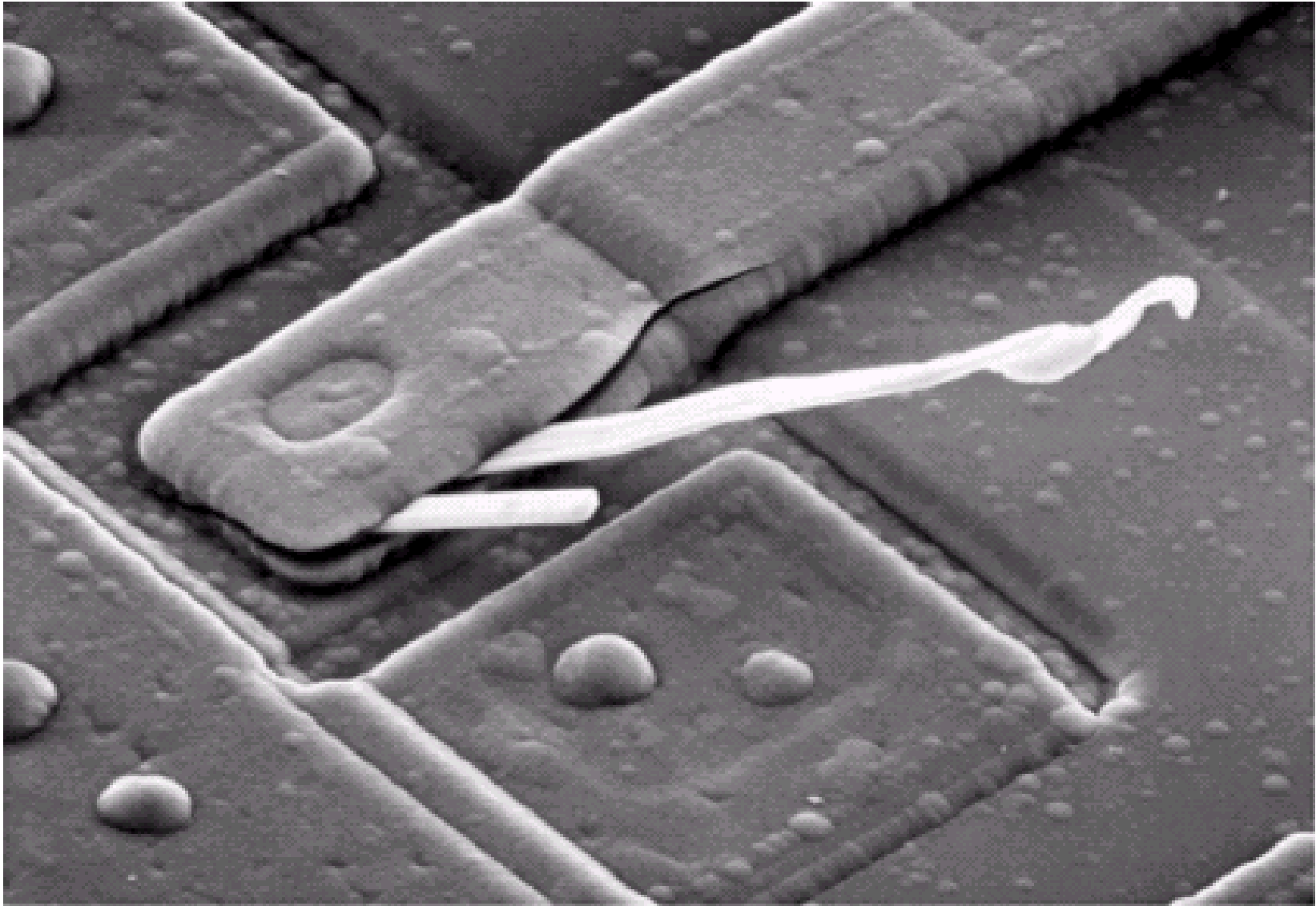


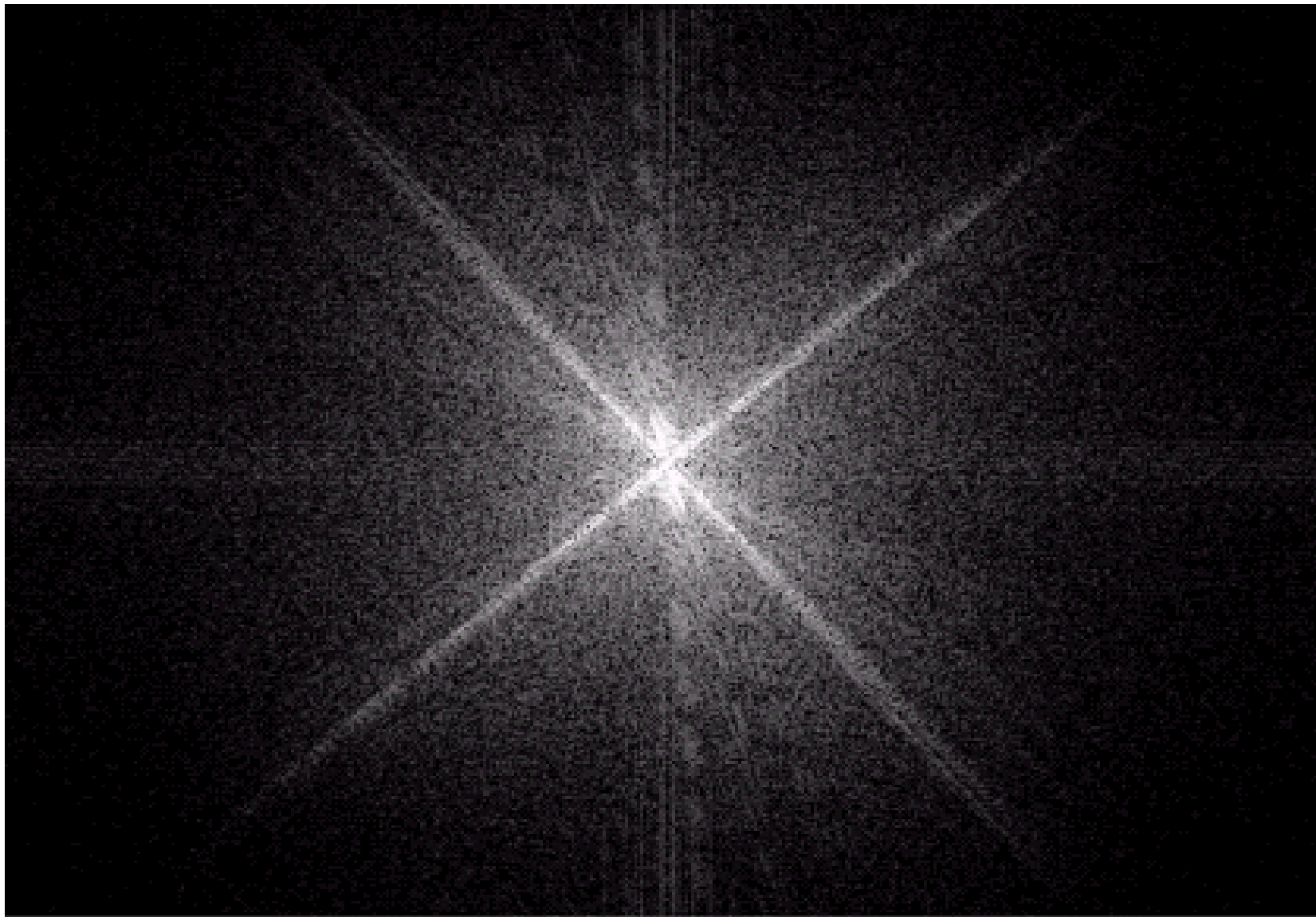


Scanning electron microscope image of an integrated circuit magnified ~2500 times



Fourier spectrum of the image





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

The inverse DFT is given by:

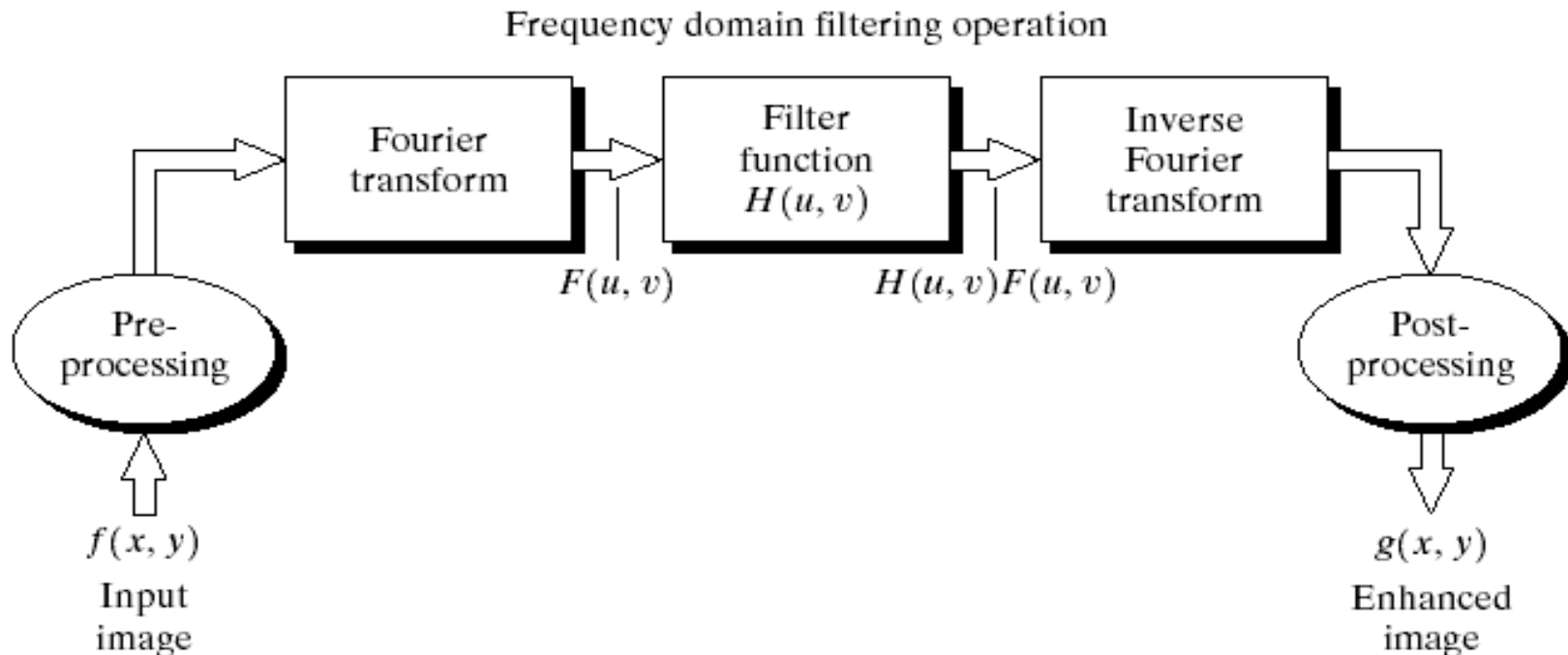
$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

for  $x = 0, 1, 2 \dots M-1$  and  $y = 0, 1, 2 \dots N-1$

# The DFT and Image Processing

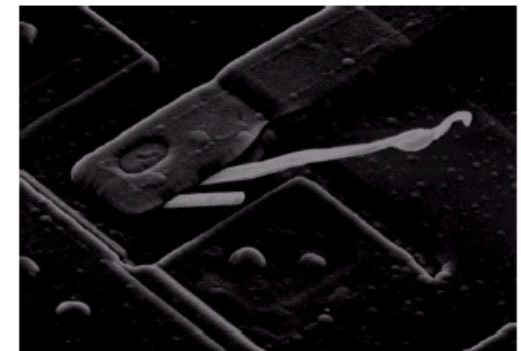
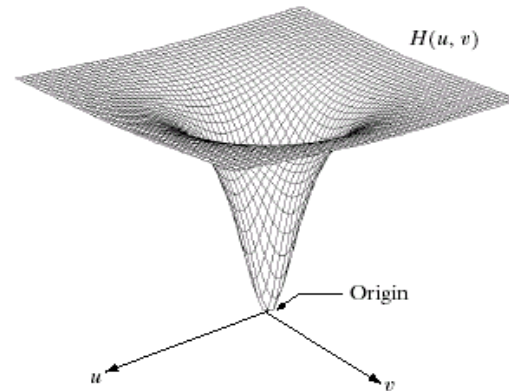
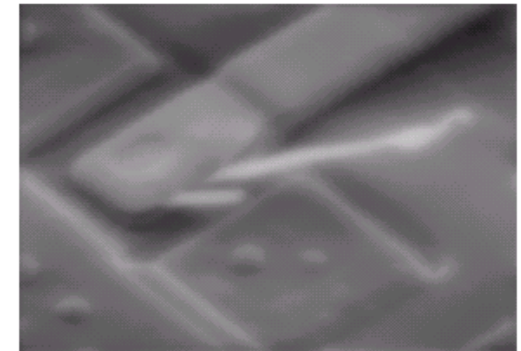
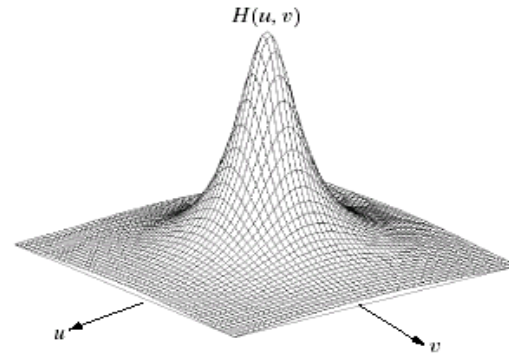
To filter an image in the frequency domain:

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

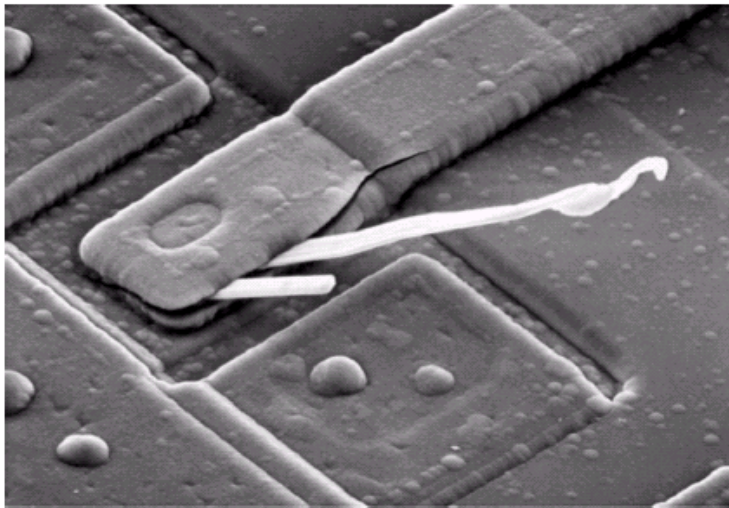


# Some Basic Frequency Domain Filters

## Low Pass Filter

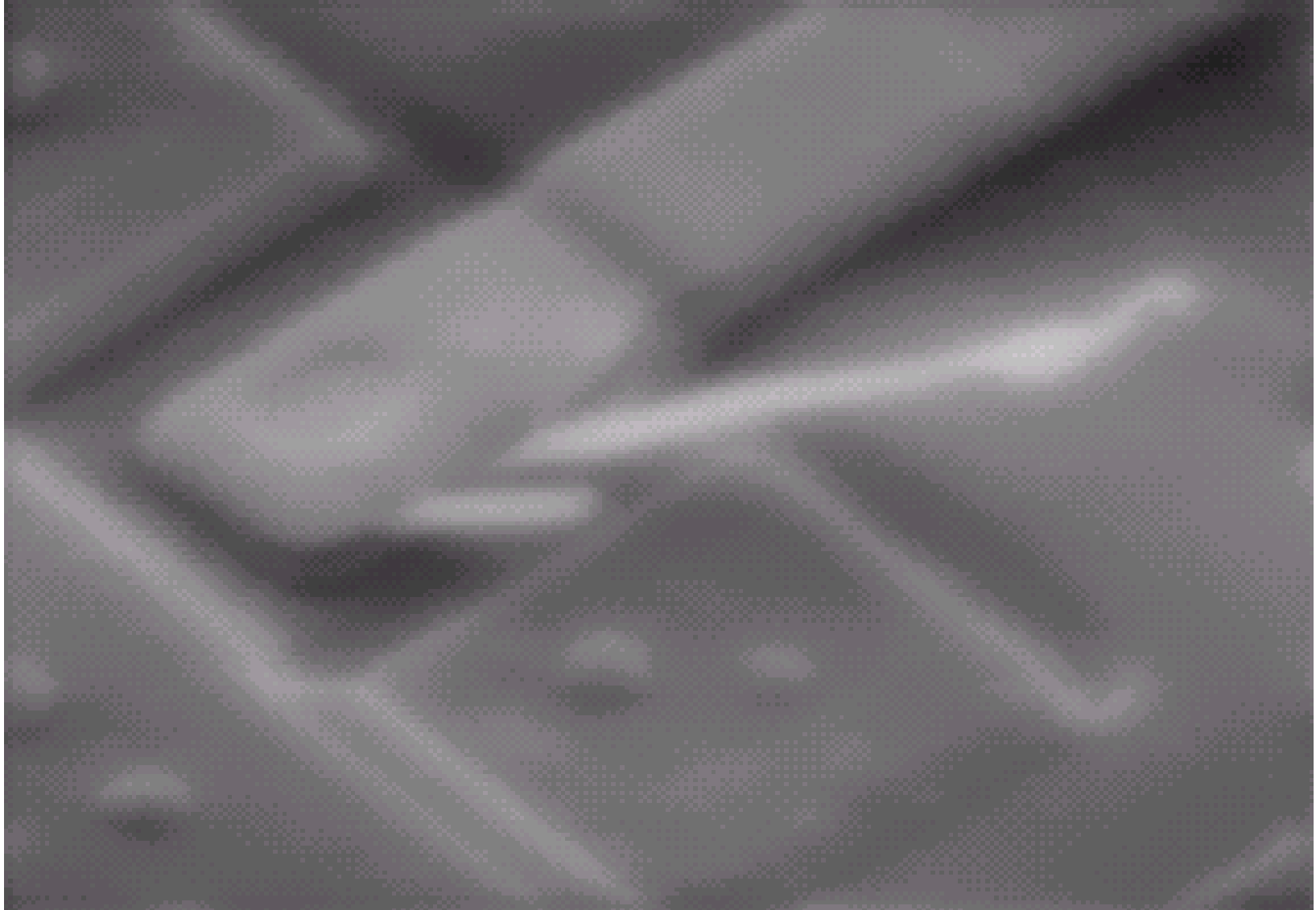


## High Pass Filter

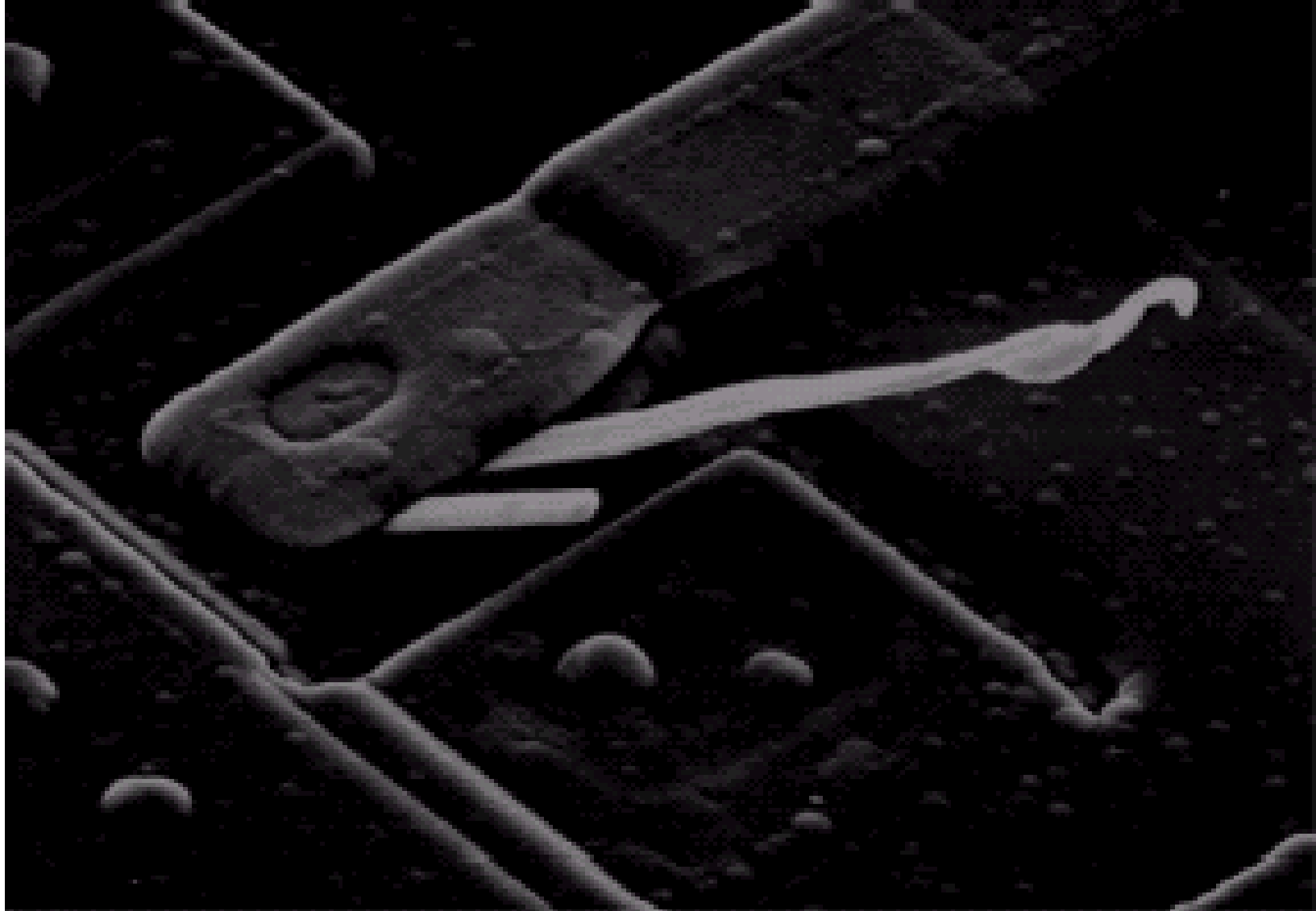




# Some Basic Frequency Domain Filters



# Some Basic Frequency Domain Filters



# Smoothing Frequency Domain Filters

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

The basic model for filtering is:

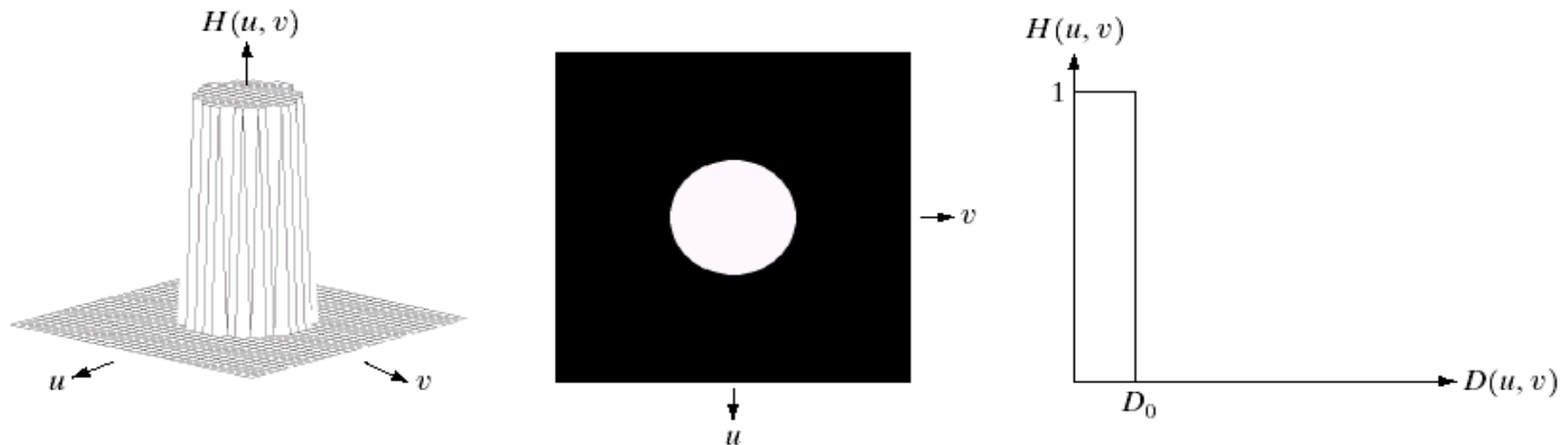
$$G(u, v) = H(u, v)F(u, v)$$

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

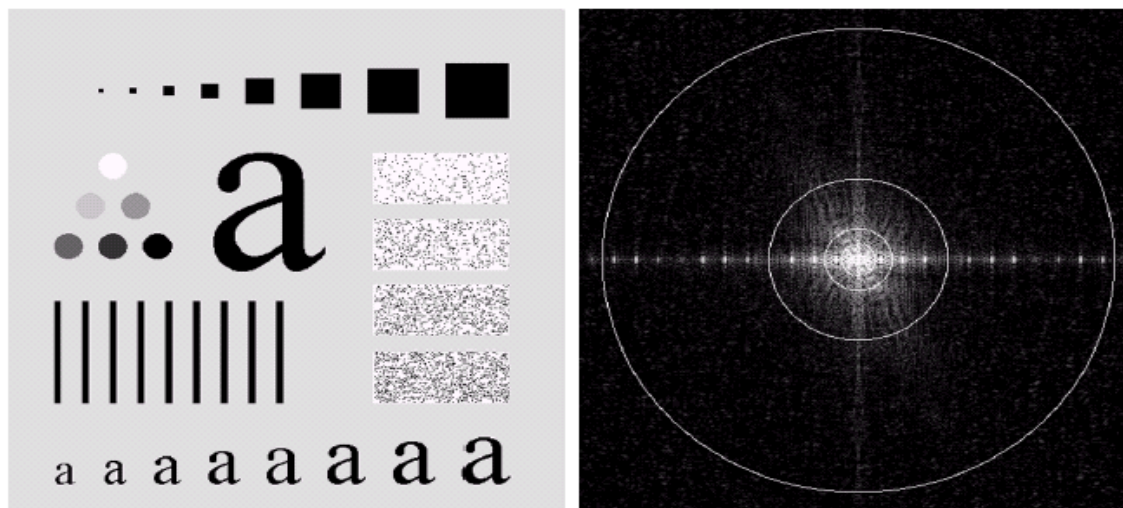
The transfer function for the ideal low pass filter can be given as:

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

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

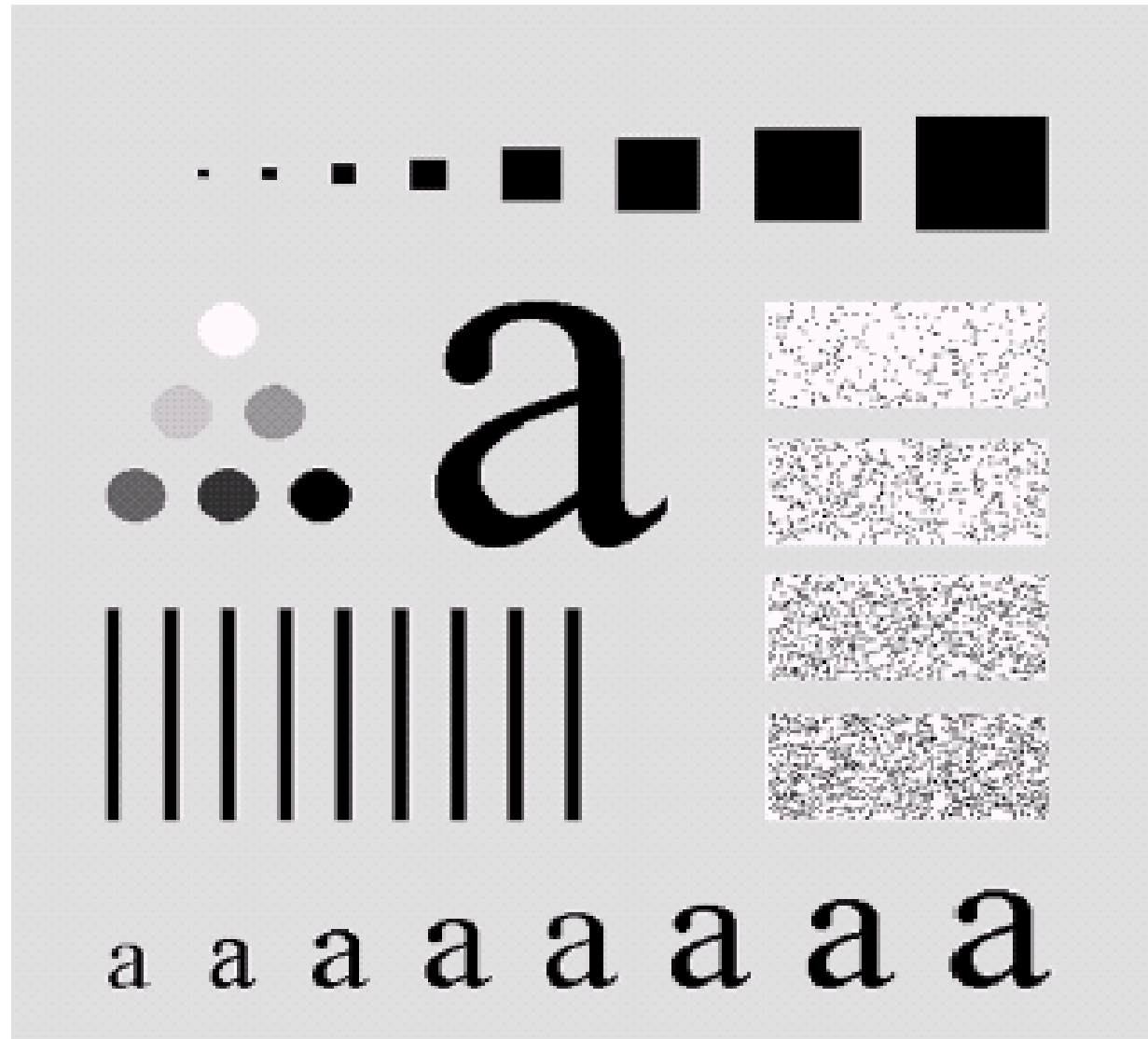
$$D(u, v) = [(u - M / 2)^2 + (v - N / 2)^2]^{1/2}$$

# Ideal Low Pass Filter (cont...)

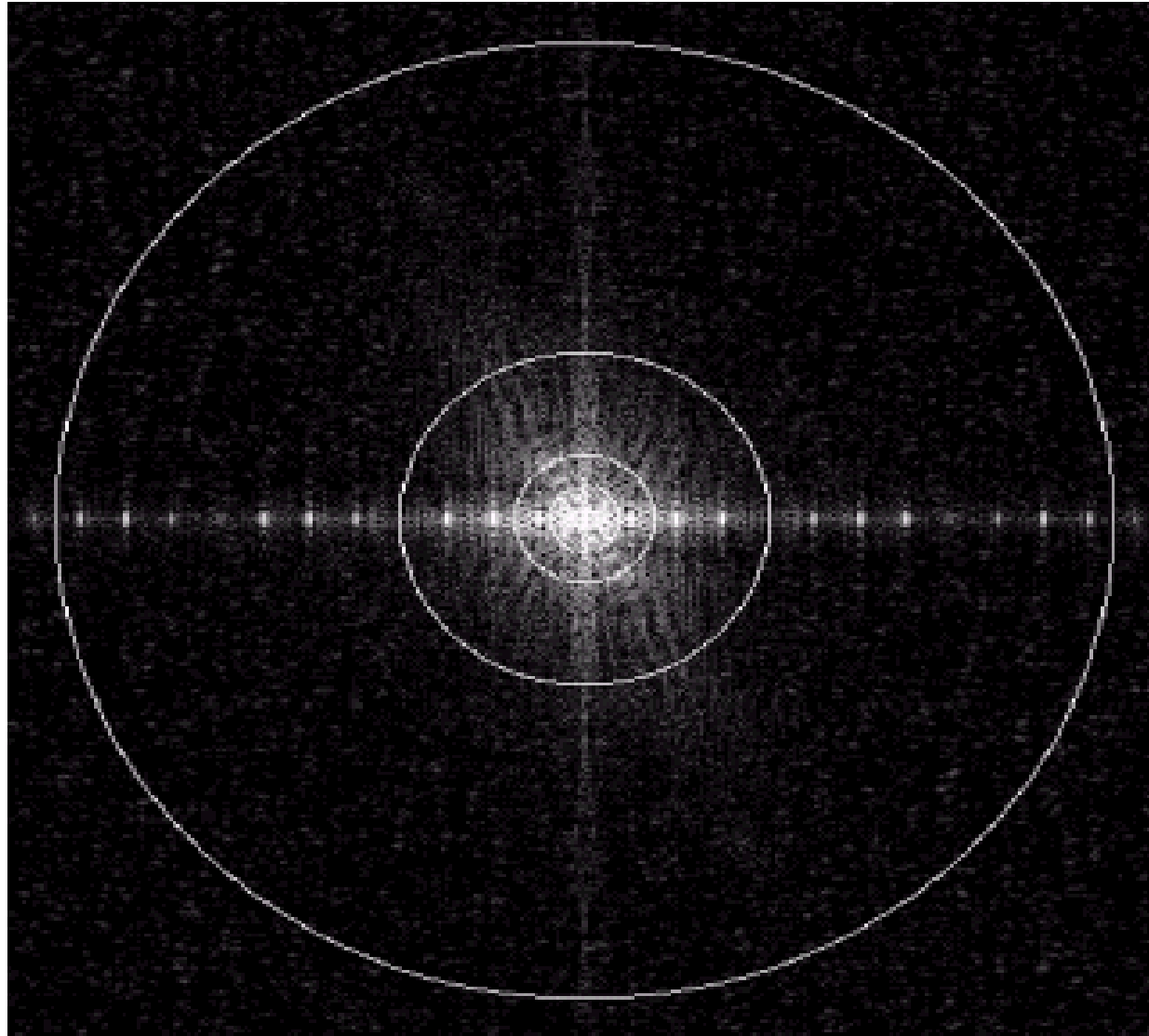


Above we show an image, it's Fourier spectrum and a series of ideal low pass filters of radius 5, 15, 30, 80 and 230 superimposed on top of it

# Ideal Low Pass Filter (cont...)



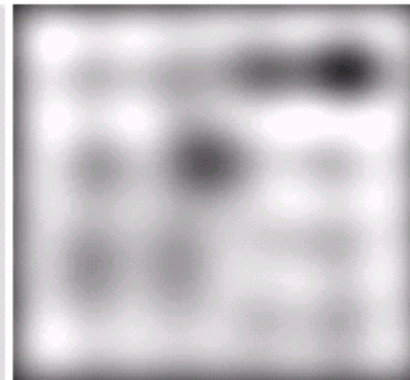
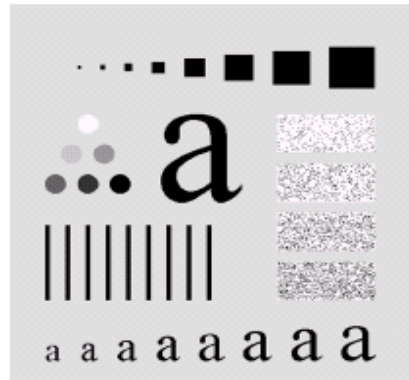
# Ideal Low Pass Filter (cont...)





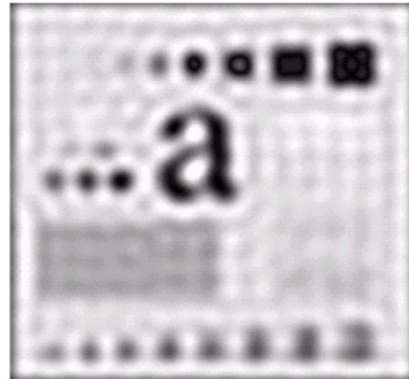
# Ideal Low Pass Filter (cont...)

Original image



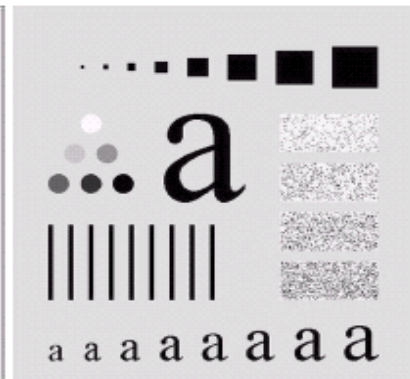
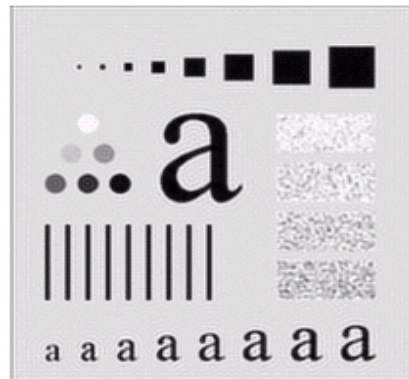
Result of filtering with ideal low pass filter of radius 5

Result of filtering with ideal low pass filter of radius 15



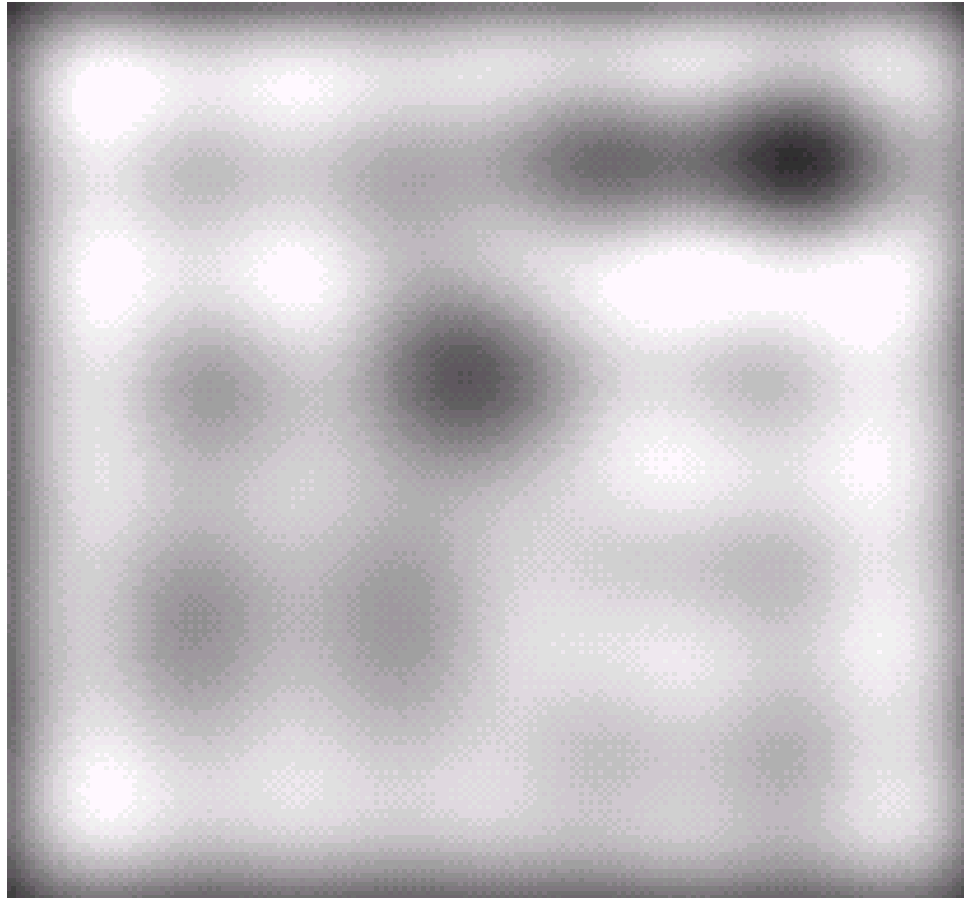
Result of filtering with ideal low pass filter of radius 30

Result of filtering with ideal low pass filter of radius 80



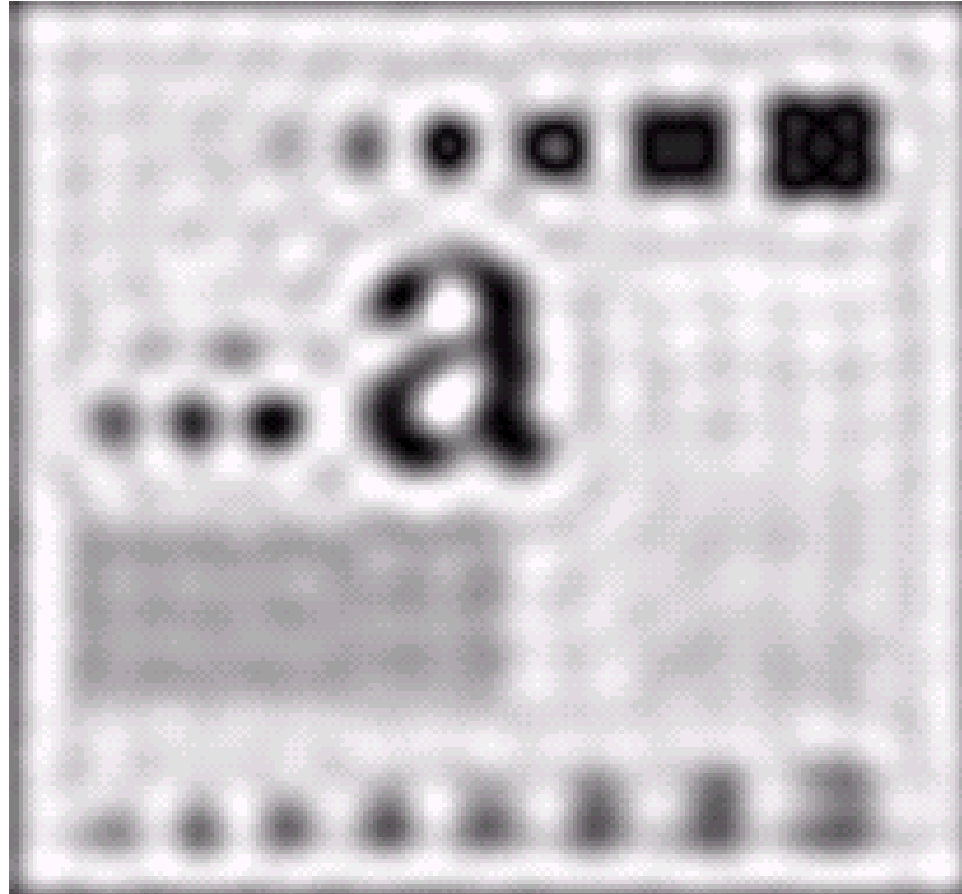
Result of filtering with ideal low pass filter of radius 230

# Ideal Low Pass Filter (cont...)



Result of filtering  
with ideal low pass  
filter of radius 5

# Ideal Low Pass Filter (cont...)

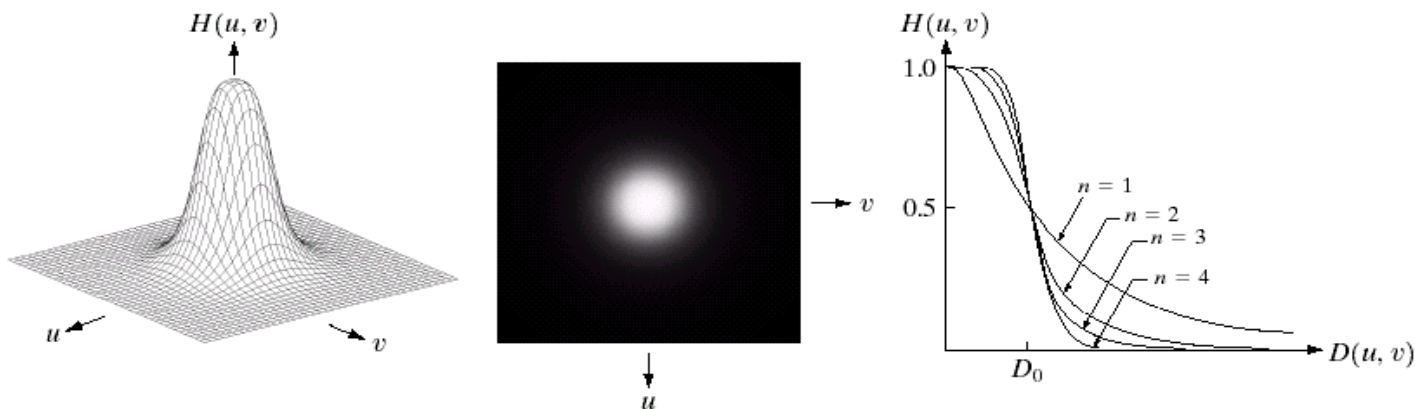


Result of filtering  
with ideal low pass  
filter of radius 15

# Butterworth Lowpass Filters

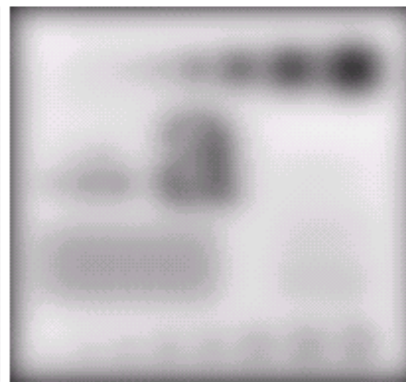
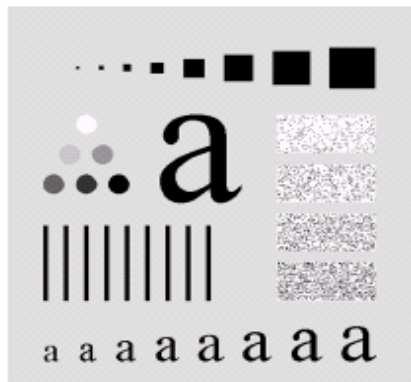
The transfer function of a Butterworth lowpass filter of order  $n$  with cutoff frequency at distance  $D_0$  from the origin is defined as:

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



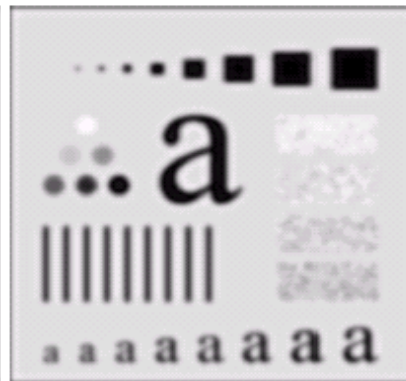
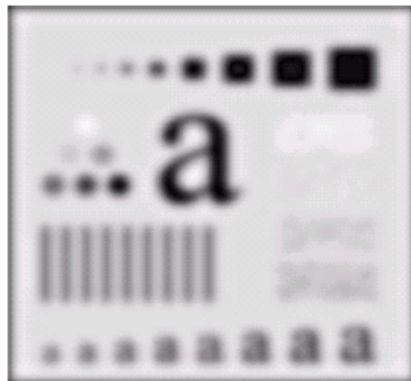
# Butterworth Lowpass Filter (cont...)

Original  
image



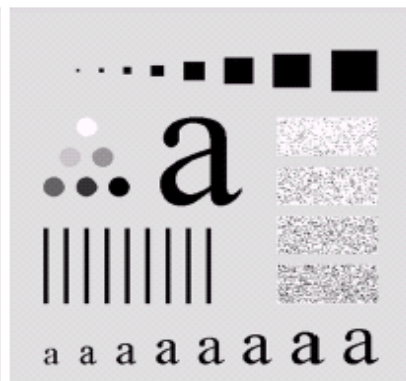
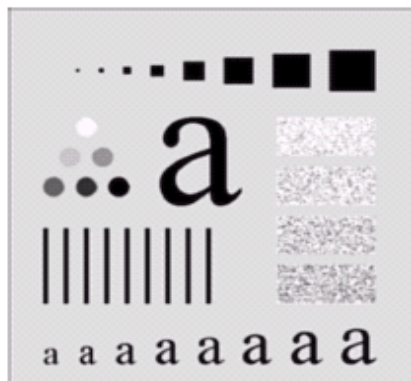
Result of filtering  
with Butterworth filter  
of order 2 and cutoff  
radius 5

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



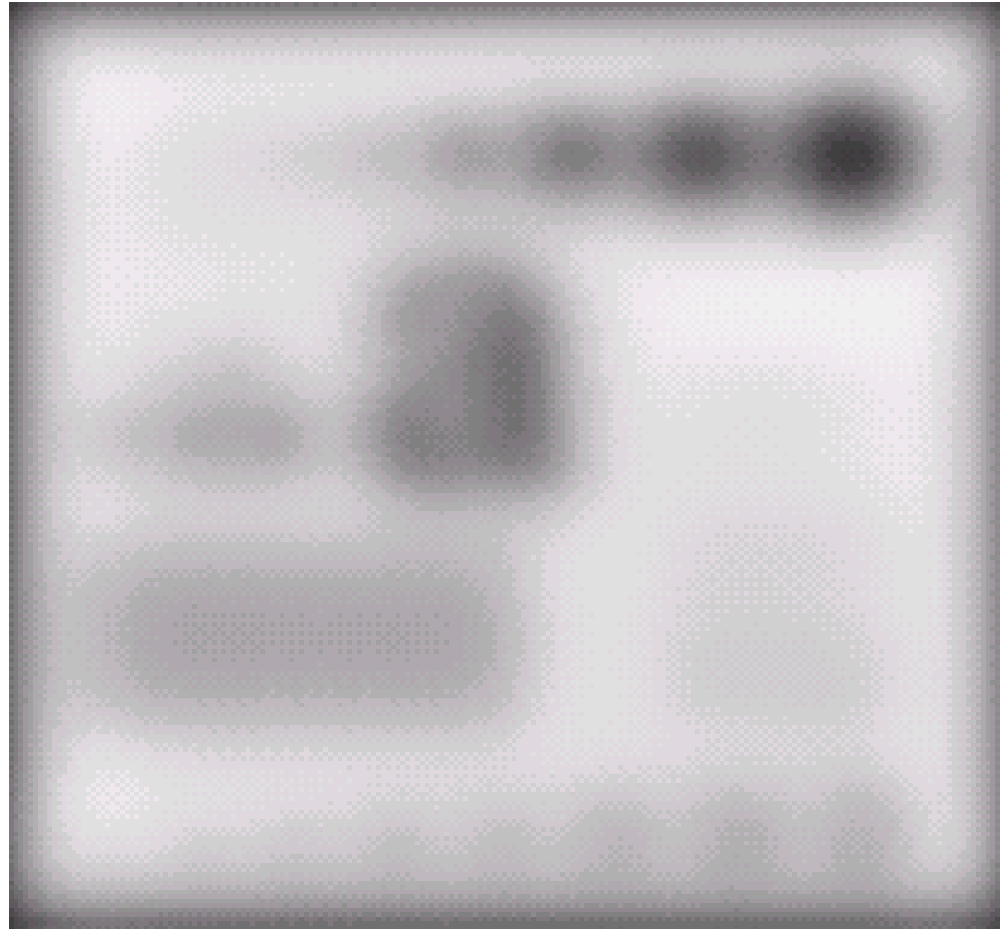
Result of filtering  
with Butterworth  
filter of order 2 and  
cutoff radius 30

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

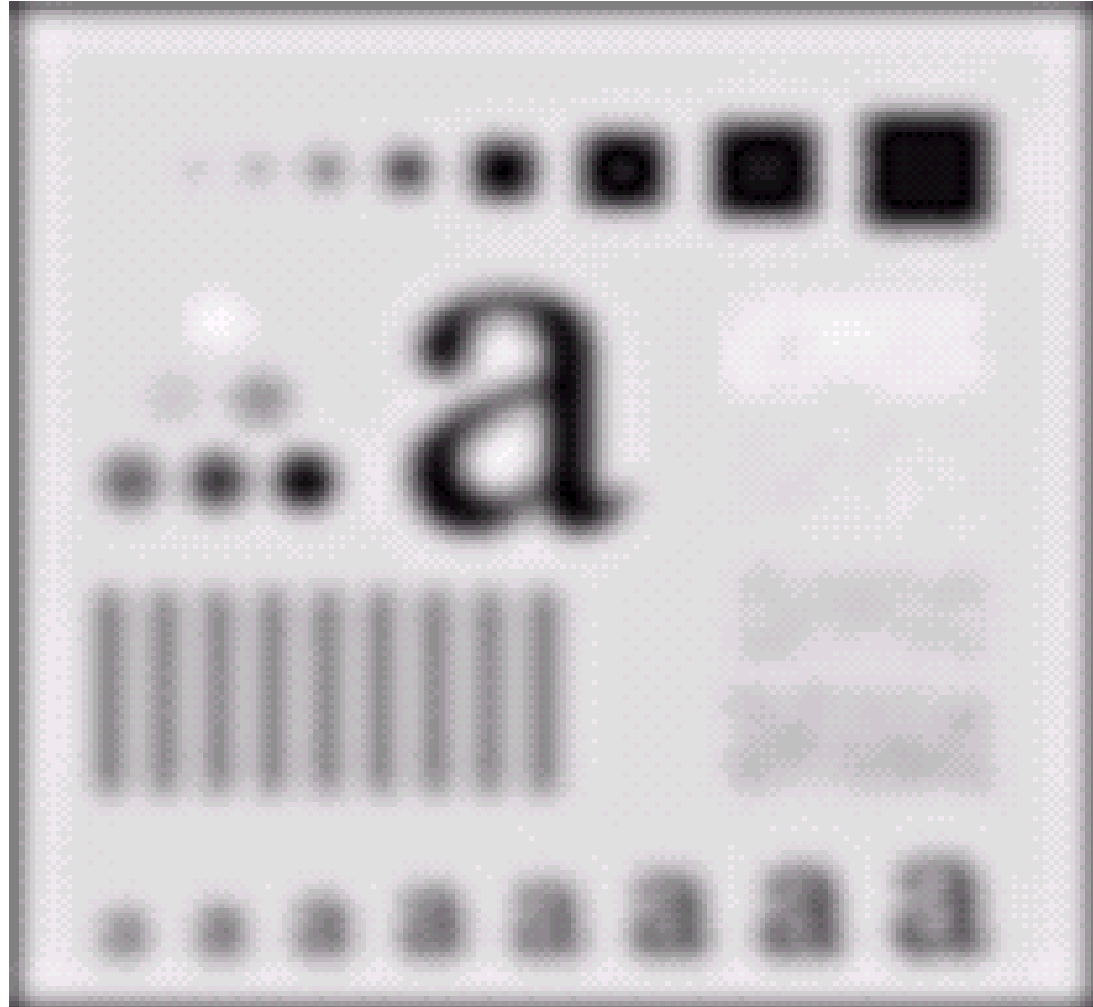
# Butterworth Lowpass Filter (cont...)



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

# Butterworth Lowpass Filter (cont...)

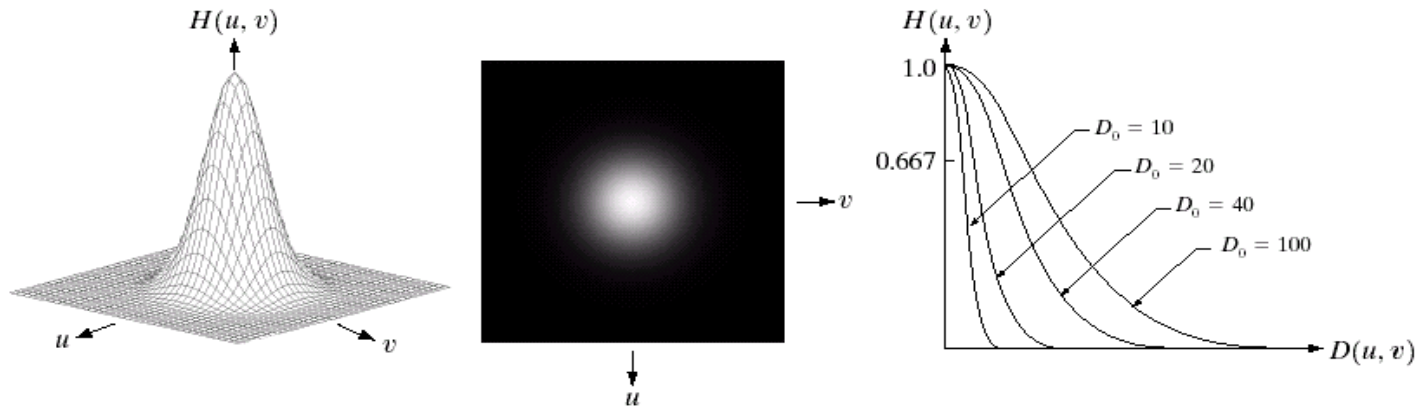
Result of filtering with  
Butterworth filter of  
order 2 and cutoff  
radius 15



# Gaussian Lowpass Filters

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

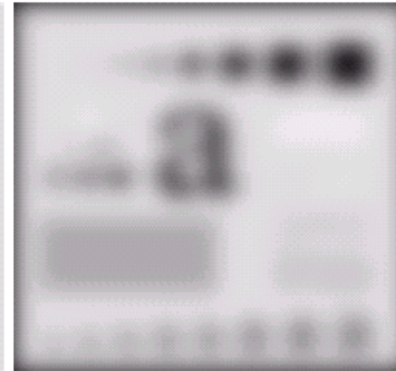
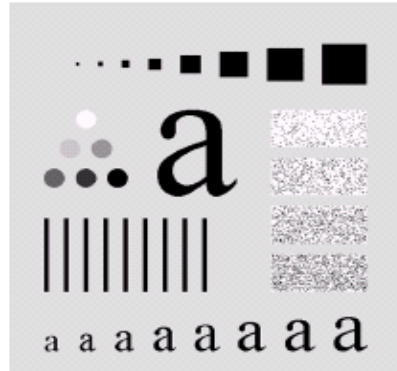
$$H(u, v) = e^{-D^2(u, v) / 2D_0^2}$$





# Gaussian Lowpass Filters (cont...)

Original  
image



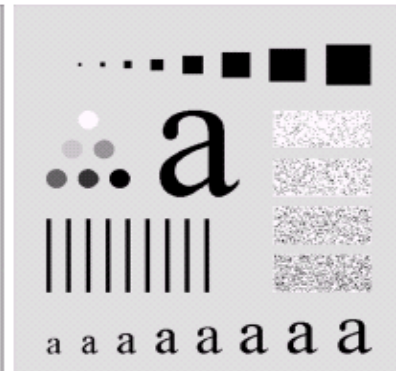
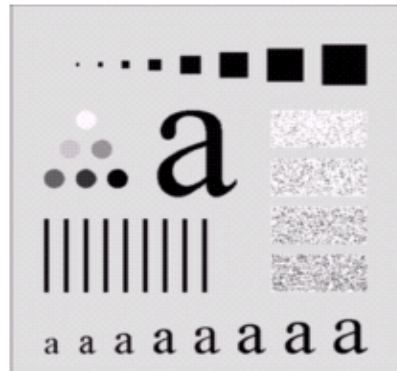
Result of filtering  
with Gaussian filter  
with cutoff radius 5

Result of filtering  
with Gaussian  
filter with cutoff  
radius 15



Result of filtering  
with Gaussian filter  
with cutoff radius 30

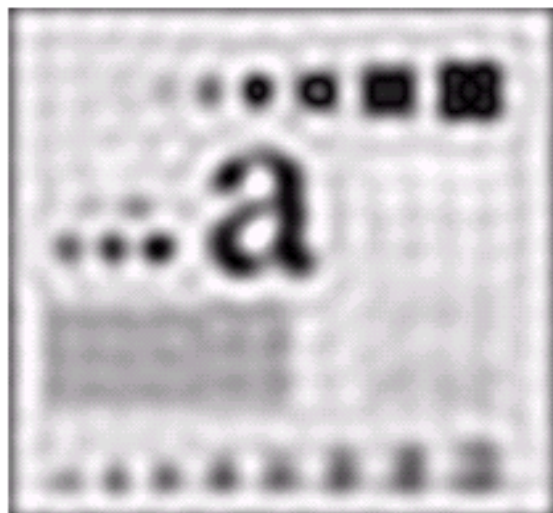
Result of filtering  
with Gaussian  
filter with cutoff  
radius 85



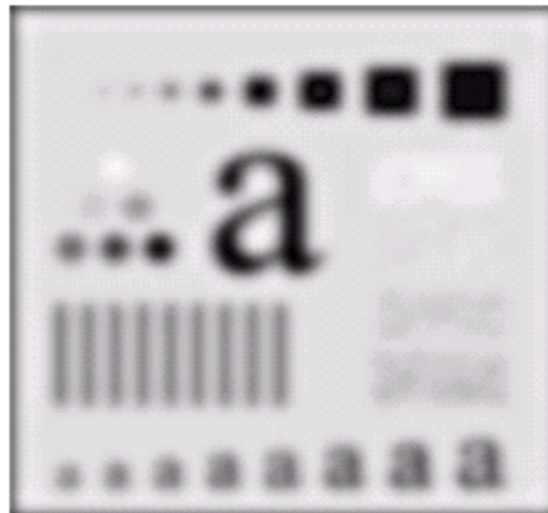
Result of filtering  
with Gaussian filter  
with cutoff radius  
230

# Lowpass Filters Compared

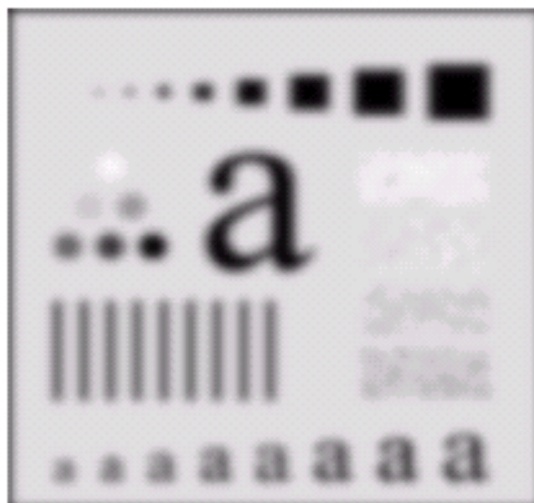
Result of filtering  
with ideal low pass  
filter of radius 15



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

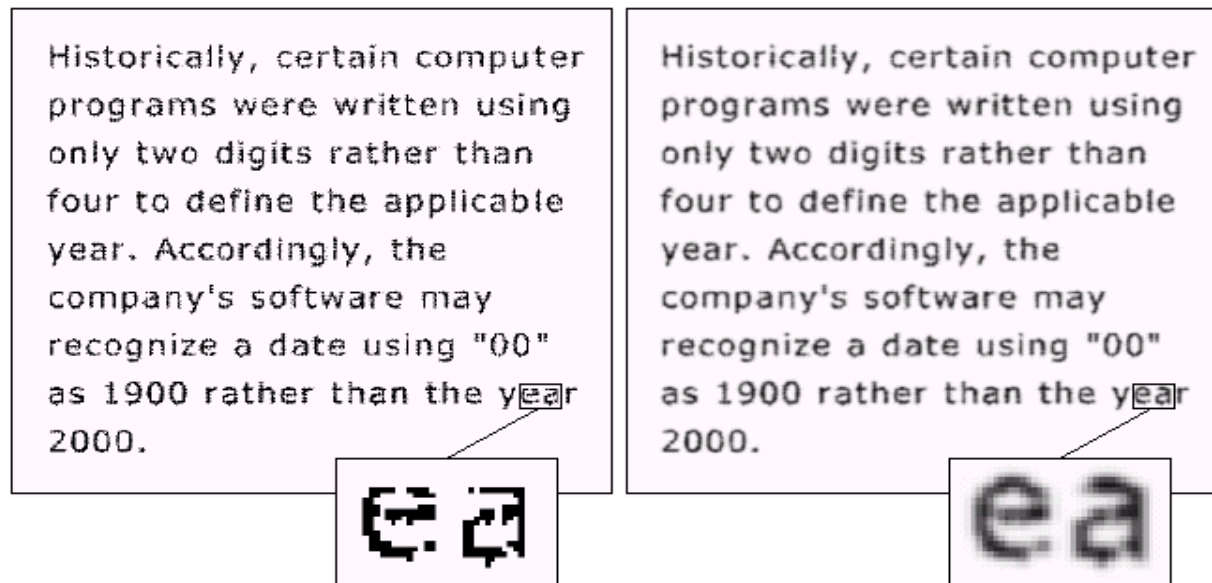


Result of filtering  
with Gaussian  
filter with cutoff  
radius 15



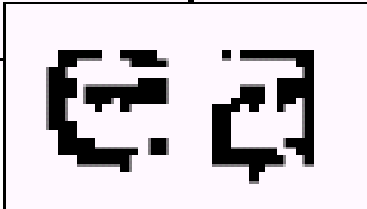
# Lowpass Filtering Examples

A low pass Gaussian filter is used to connect broken text



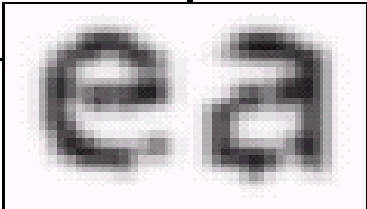
# Lowpass Filtering Examples

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.



ea

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.



ea

# Lowpass Filtering Examples (cont...)

Different lowpass Gaussian filters used to remove blemishes in a photograph

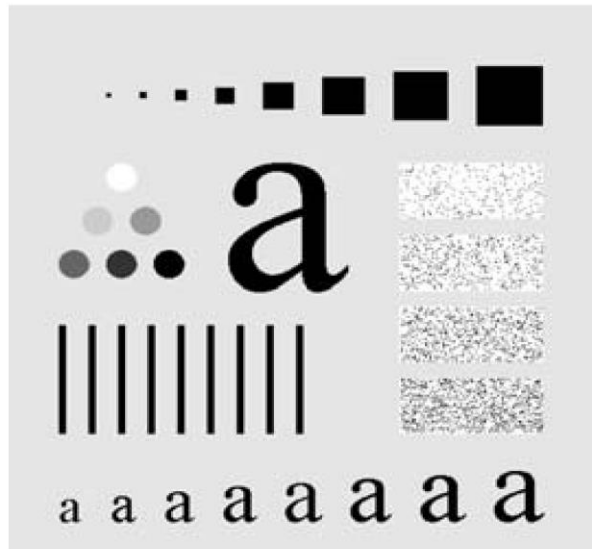


# Lowpass Filtering Examples (cont...)

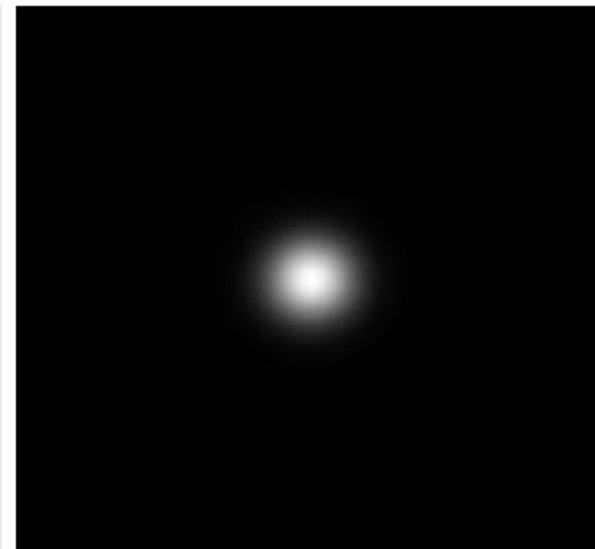


# Lowpass Filtering Examples (cont...)

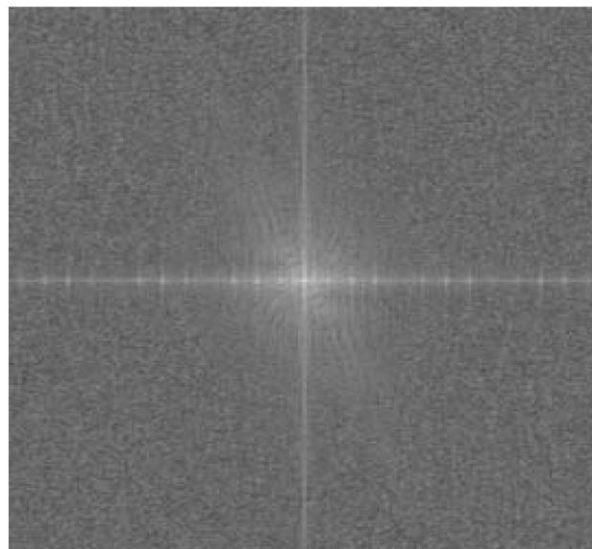
Original  
image



Gaussian lowpass  
filter



Spectrum of  
original image



Processed  
image



# 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:

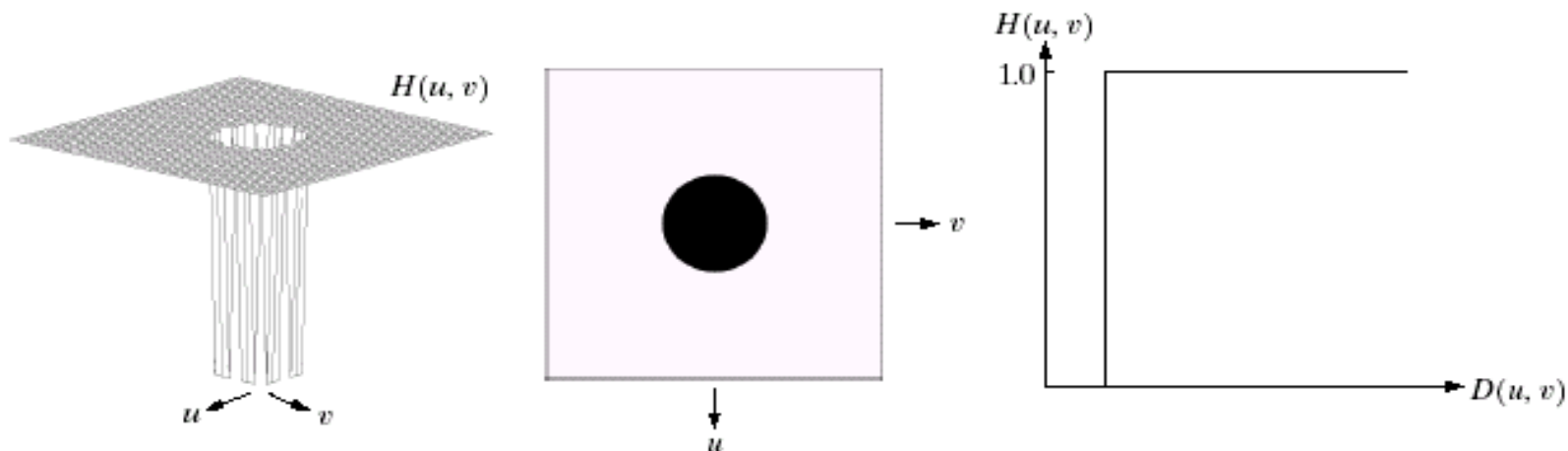
$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$



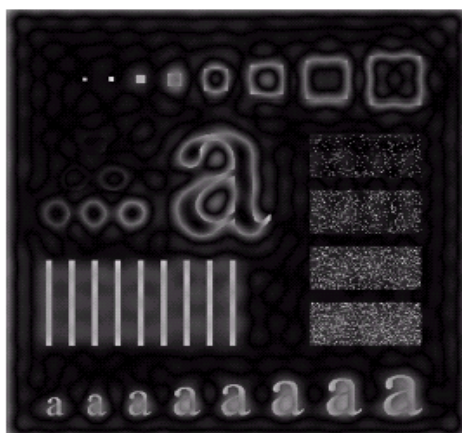
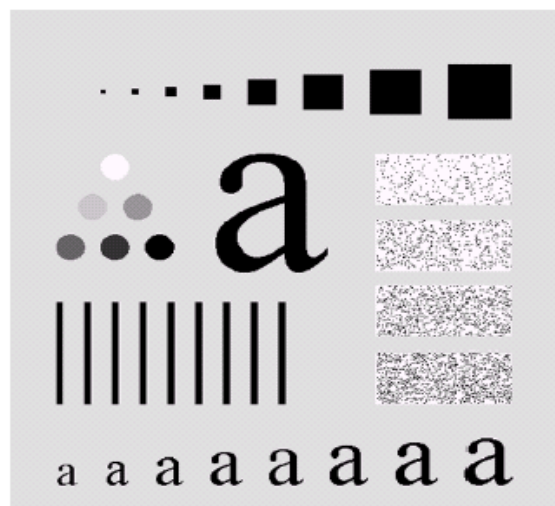
The ideal high pass filter is given as:

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

where  $D_0$  is the cut off distance as before



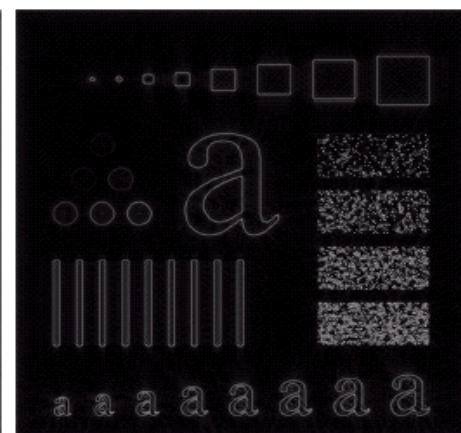
## Ideal High Pass Filters (cont...)



Results of ideal high pass filtering with  $D_0 = 15$



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



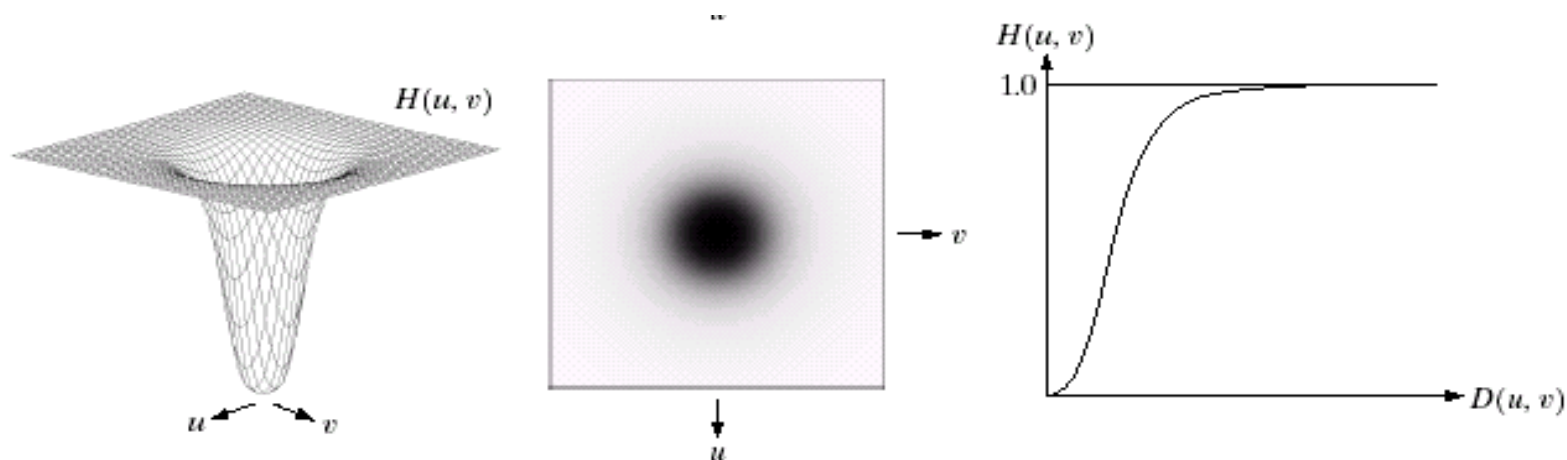
Results of ideal high pass filtering with  $D_0 = 80$

# Butterworth High Pass Filters

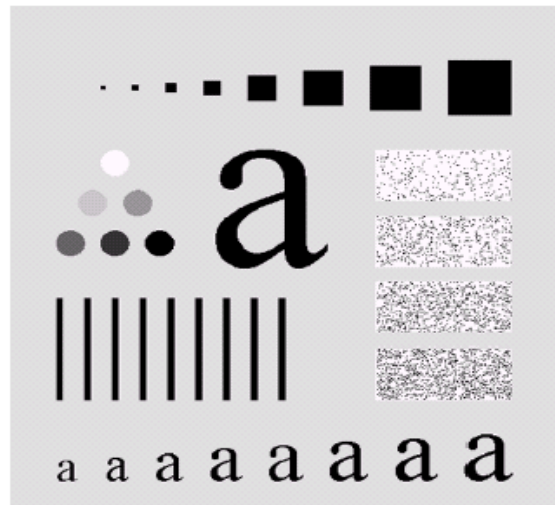
The Butterworth high pass filter is given as:

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

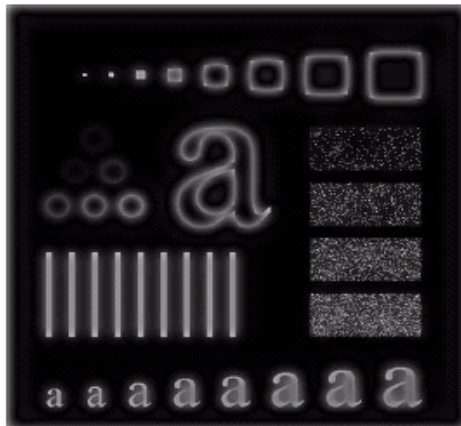
where  $n$  is the order and  $D_0$  is the cut off distance as before



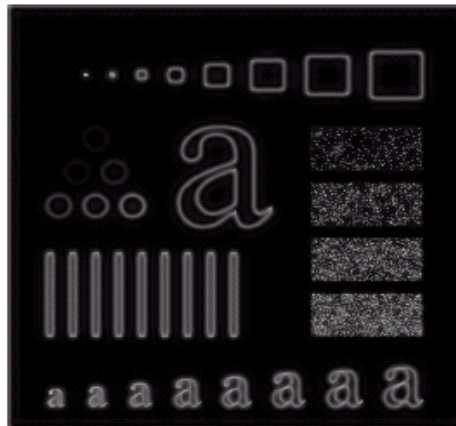
# Butterworth High Pass Filters (cont...)



Results of Butterworth high pass filtering of order 2 with  $D_0 = 15$



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



Results of Butterworth high pass filtering of order 2 with  $D_0 = 80$

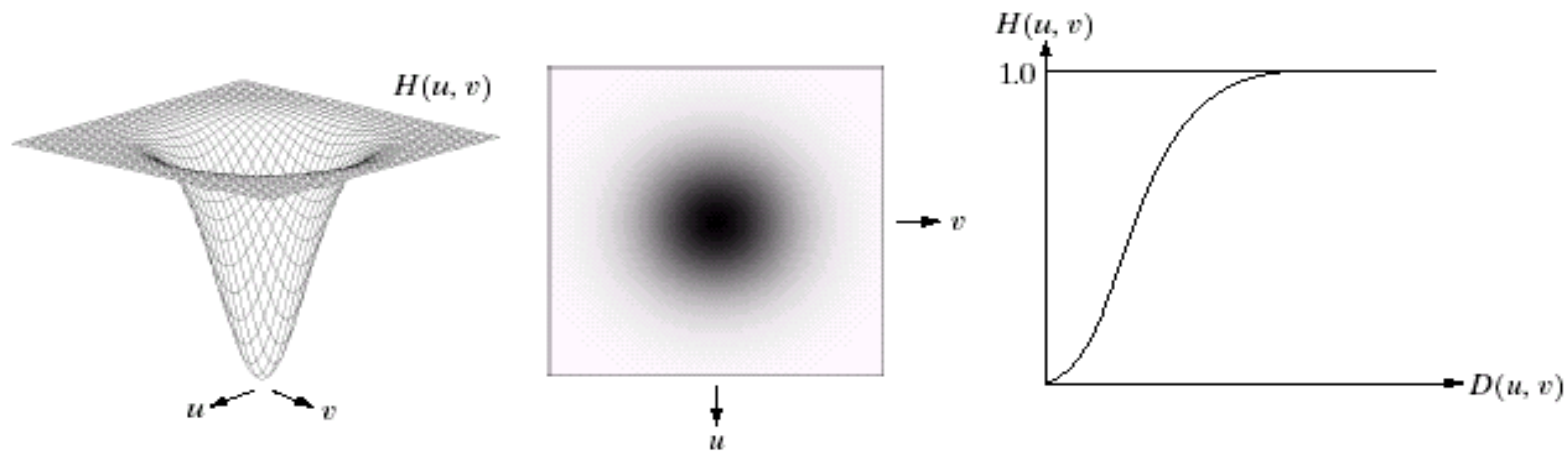


# Gaussian High Pass Filters

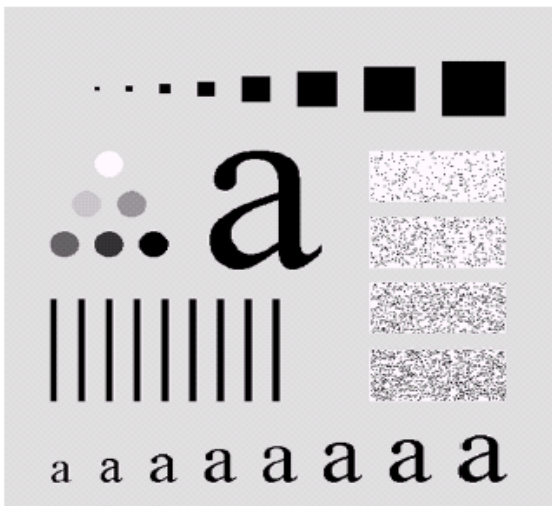
The Gaussian high pass filter is given as:

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

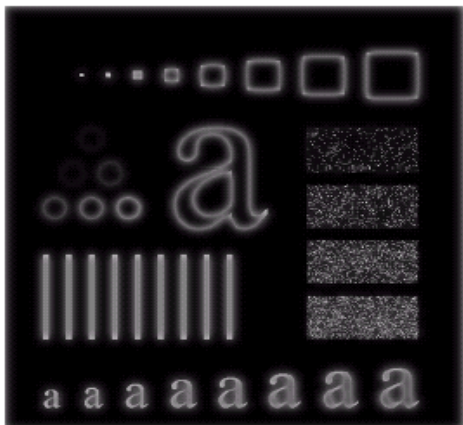
where  $D_0$  is the cut off distance as before



# Gaussian High Pass Filters (cont...)



Results of Gaussian high pass filtering with  $D_0 = 15$



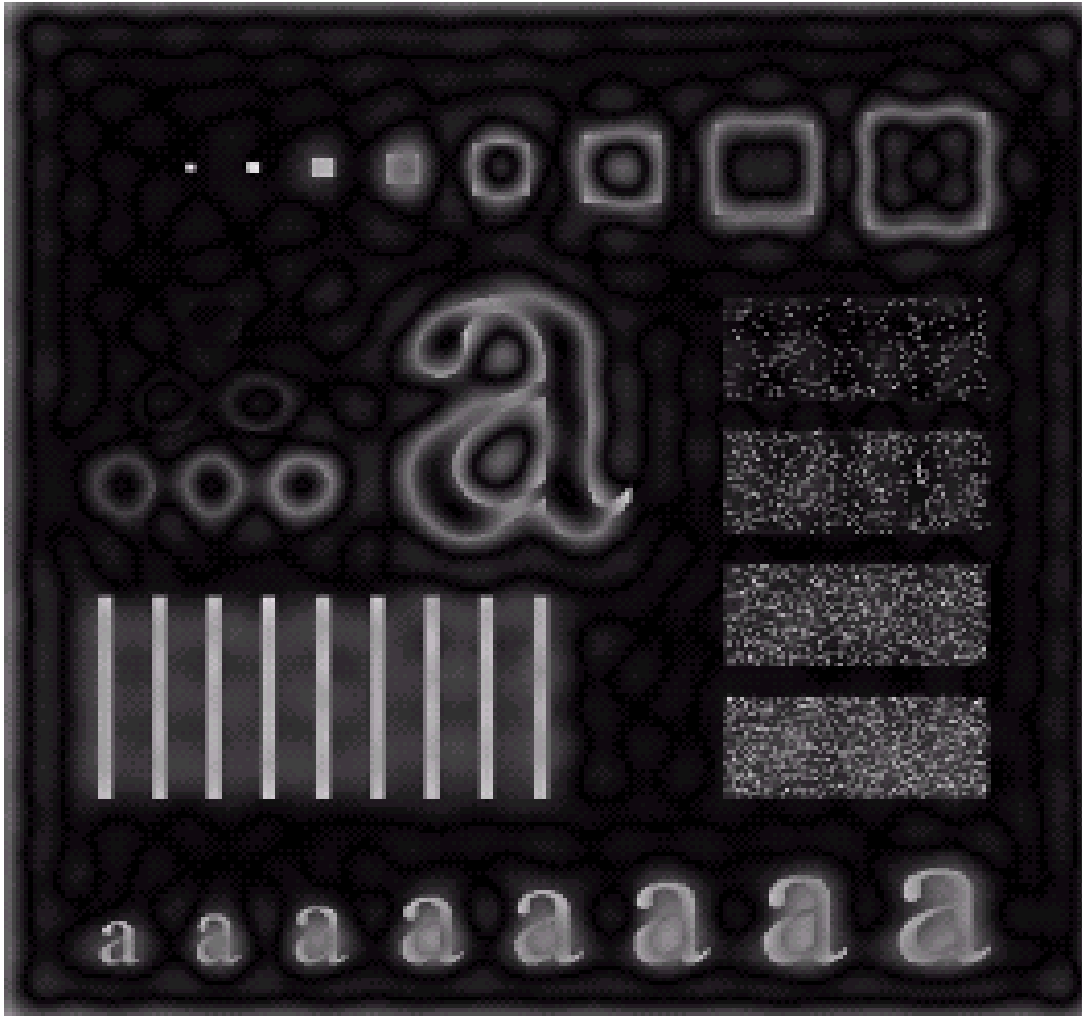
Results of Gaussian high pass filtering with  $D_0 = 30$



Results of Gaussian high pass filtering with  $D_0 = 80$

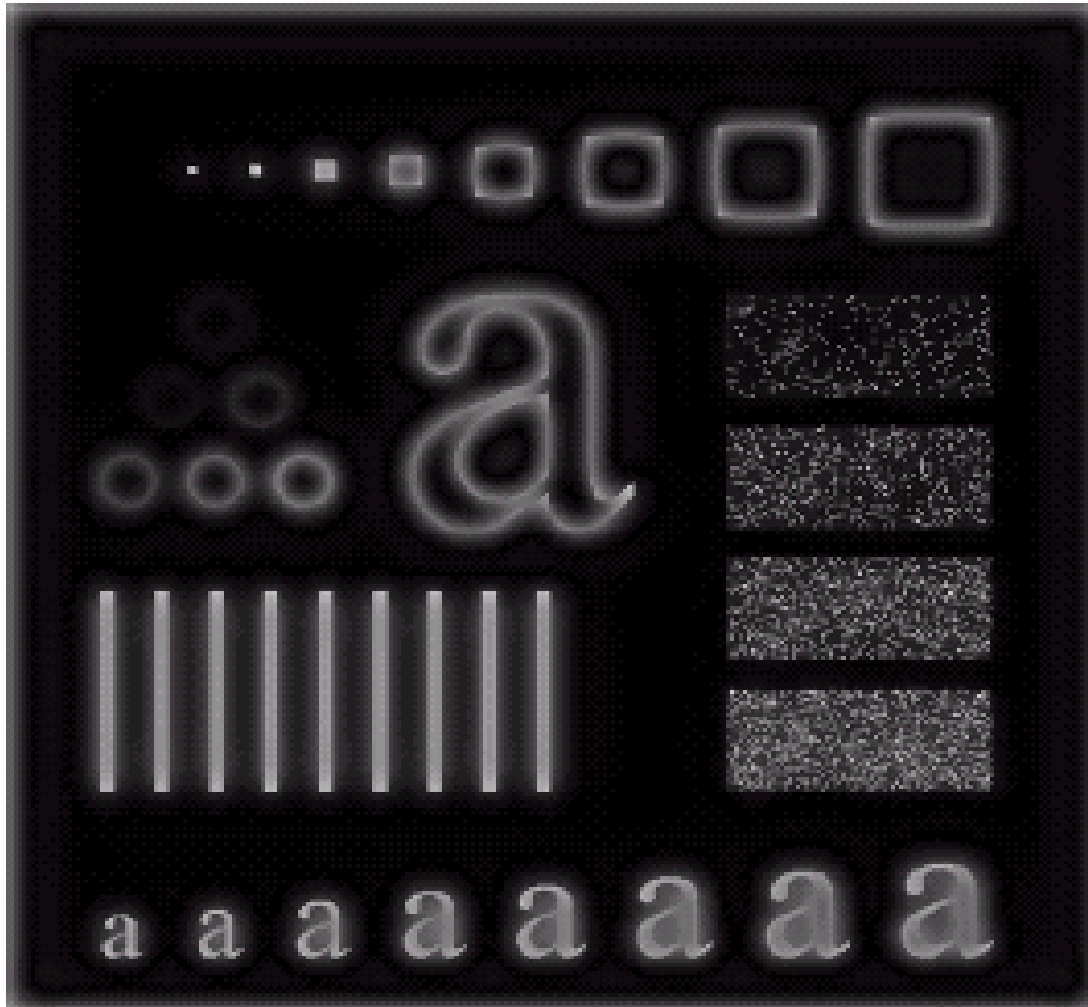


# Highpass Filter Comparison



Results of ideal  
high pass filtering  
with  $D_0 = 15$

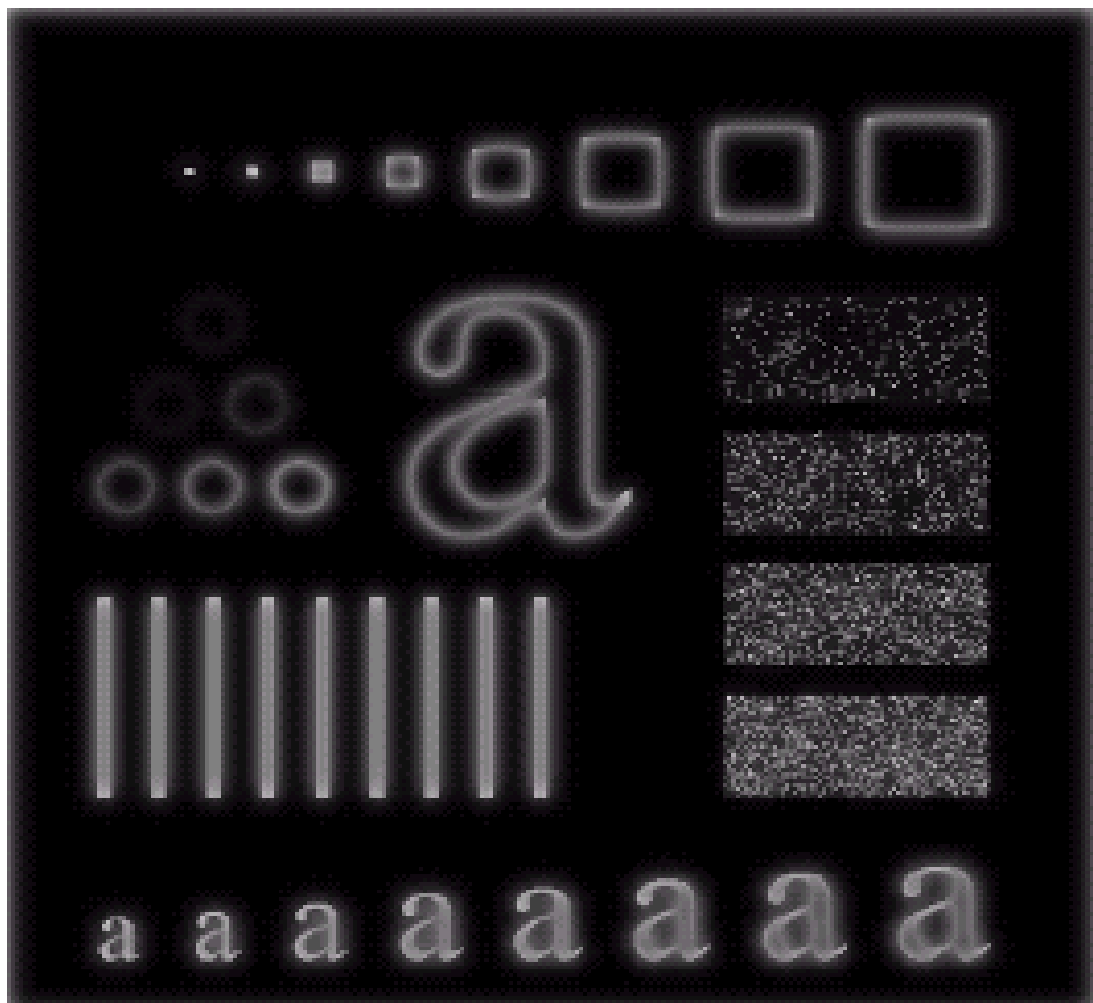
# Highpass Filter Comparison



Results of Butterworth  
high pass filtering of order  
2 with  $D_0 = 15$

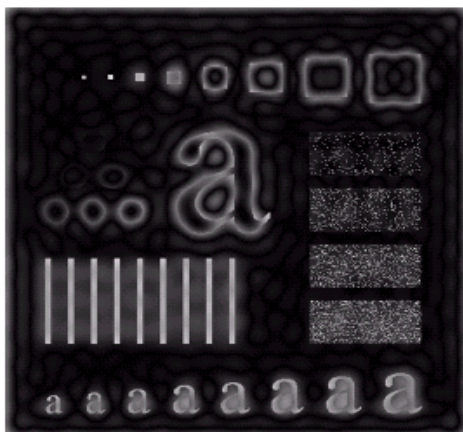
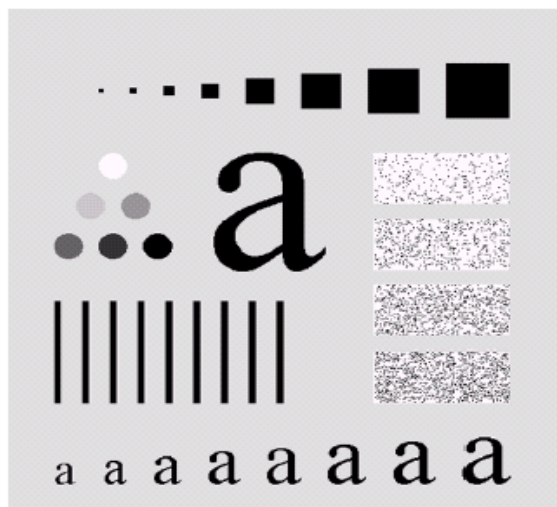


# Highpass Filter Comparison

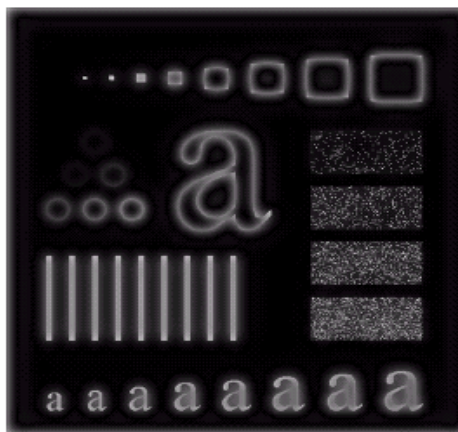


Results of Gaussian  
high pass filtering with  
 $D_0 = 15$

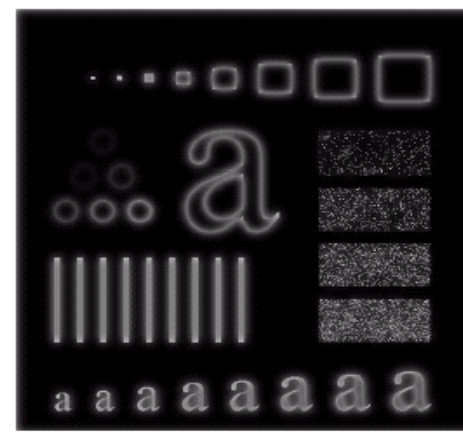
# Highpass Filter Comparison



Results of ideal  
high pass filtering  
with  $D_0 = 15$

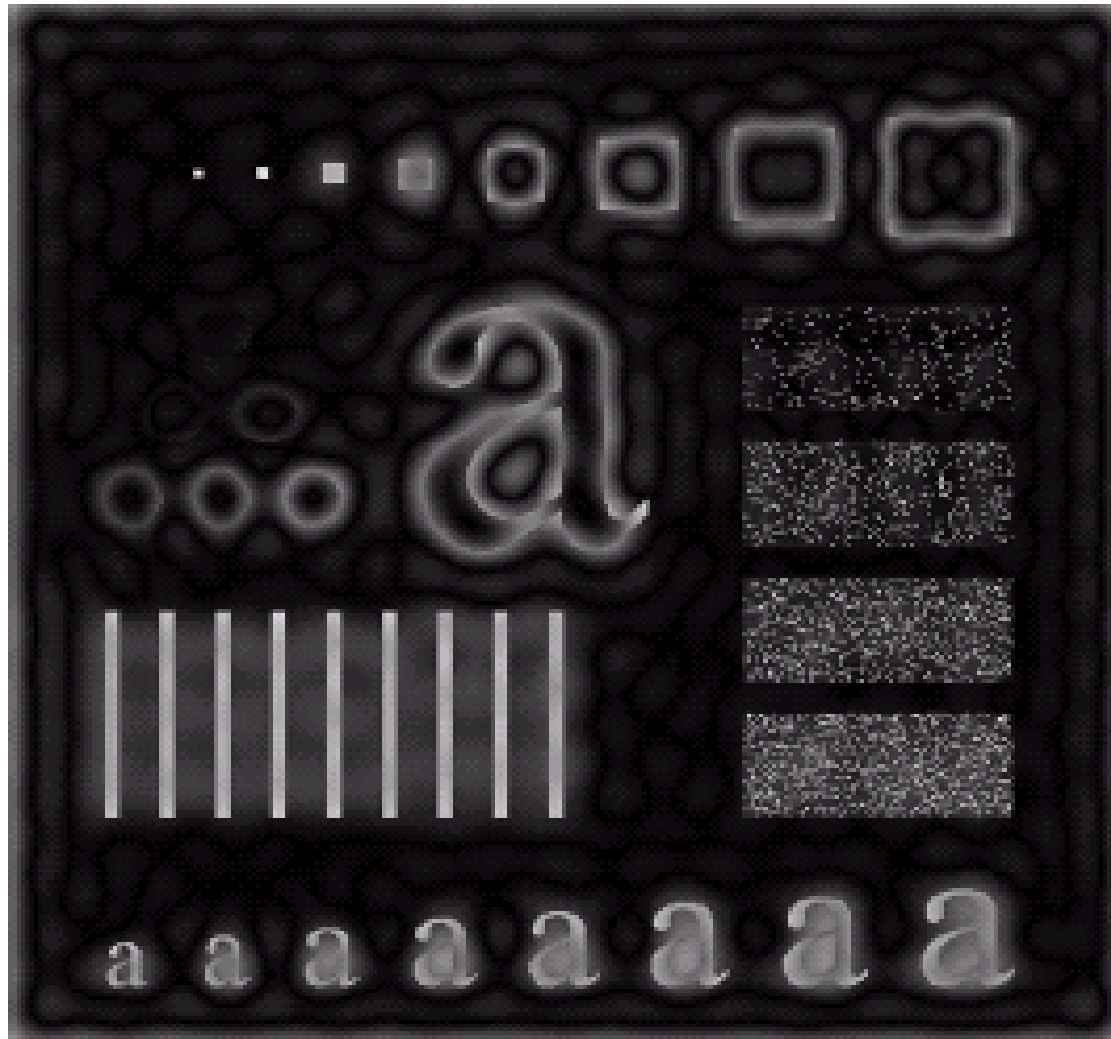


Results of Butterworth  
high pass filtering of order  
2 with  $D_0 = 15$



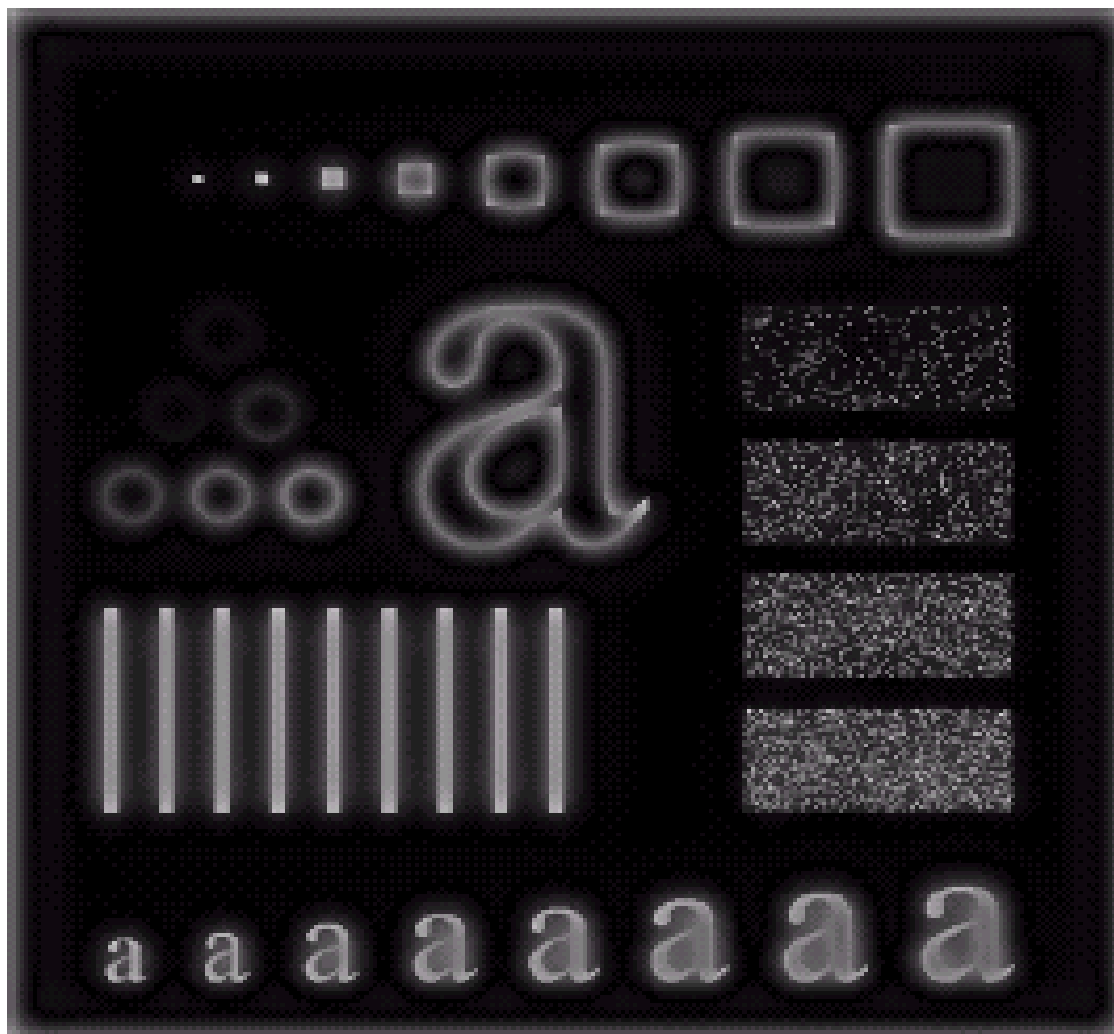
Results of Gaussian  
high pass filtering with  
 $D_0 = 15$

# Highpass Filter Comparison



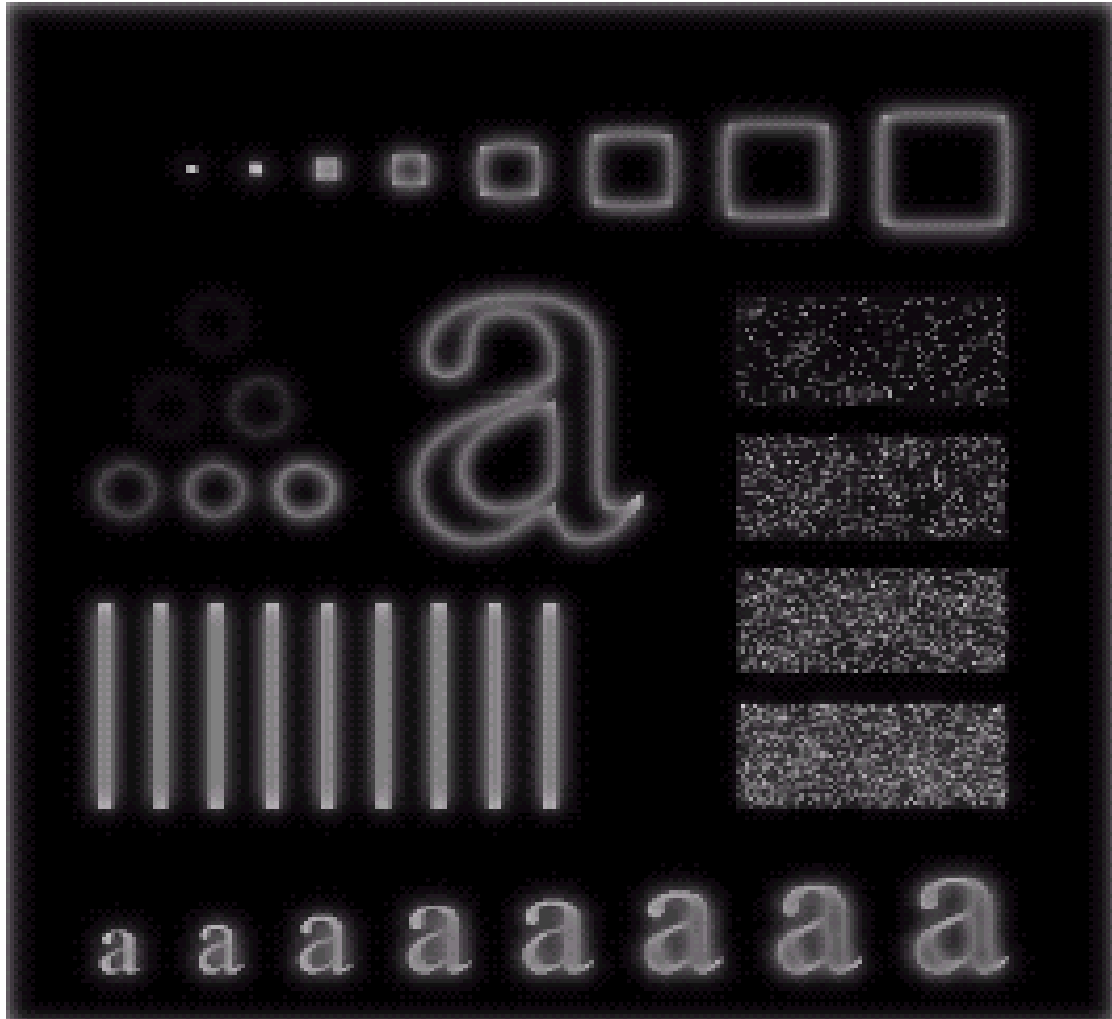
Results of ideal  
high pass filtering  
with  $D_0 = 15$

# Highpass Filter Comparison



Results of Butterworth  
high pass filtering of order  
2 with  $D_0 = 15$

# Highpass Filter Comparison



Results of Gaussian  
high pass filtering with  
 $D_0 = 15$

# Highpass Filtering Example

Original image



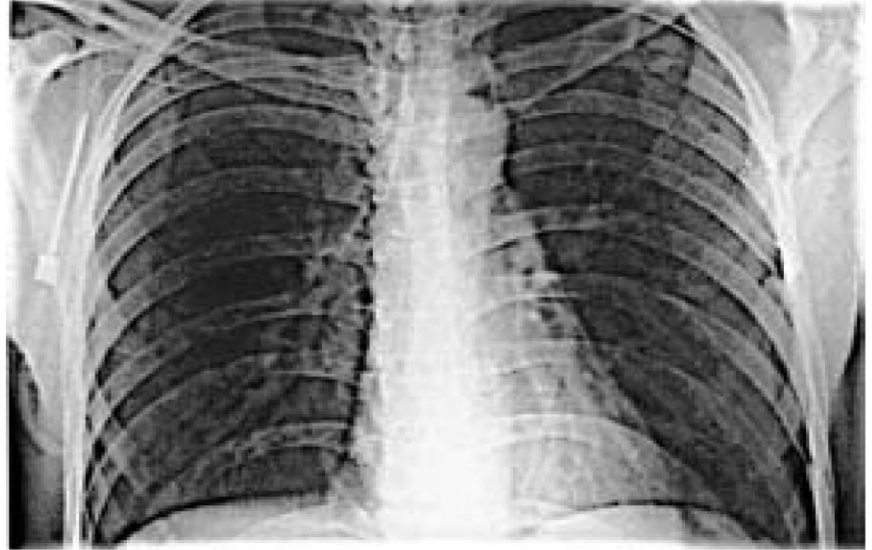
Highpass filtering result



High frequency  
emphasis result



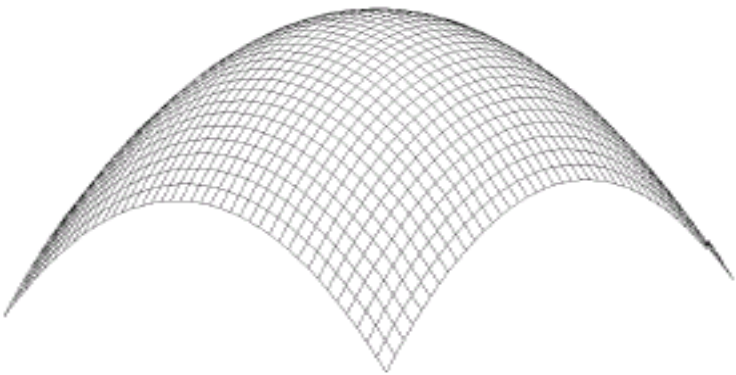
After histogram  
equalisation



# Laplacian In The Frequency Domain

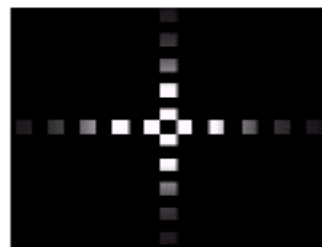
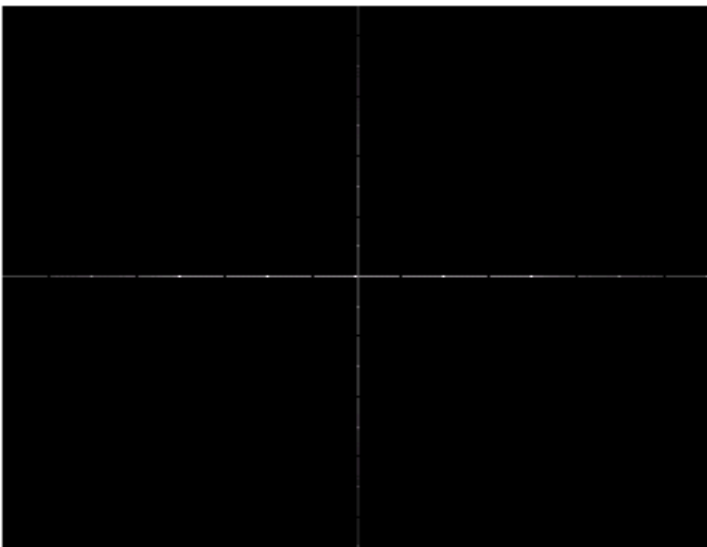
55  
of  
54

Laplacian in the  
frequency domain

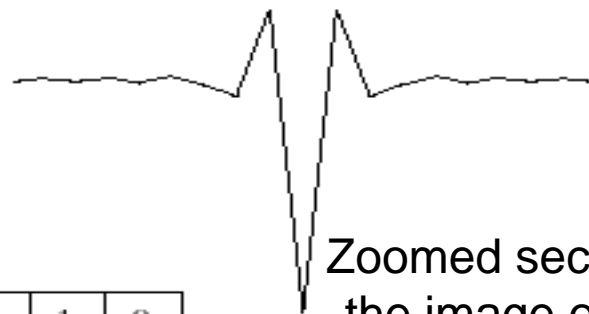


2-D image of Laplacian  
in the frequency  
domain

Inverse DFT of  
Laplacian in the  
frequency domain



0	1	0
1	-4	1
0	1	0



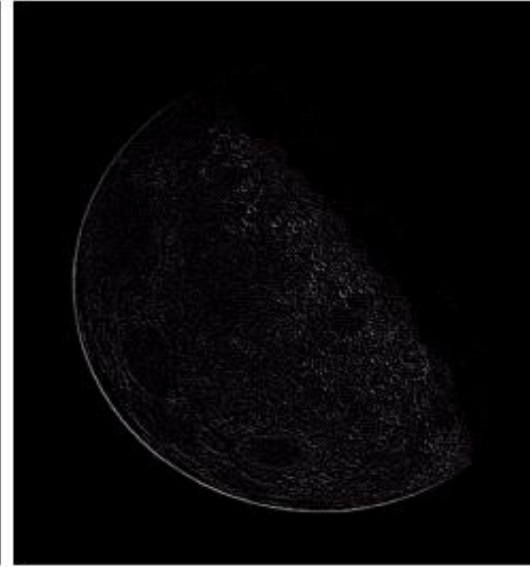
Zoomed section of  
the image on the  
left compared to  
spatial filter

# Frequency Domain Laplacian Example

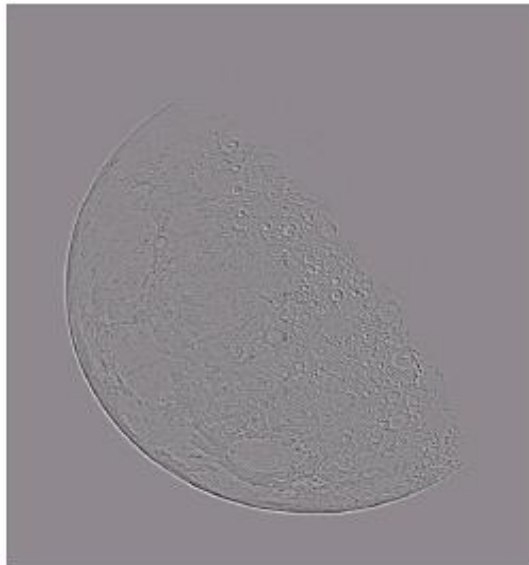
Original  
image



Laplacian  
filtered  
image



Laplacian  
image scaled



Enhanced  
image





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

Allows the Fourier transform to be carried out in a reasonable amount of time

Reduces the amount of time required to perform a Fourier transform by a factor of 100 – 600 times!

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