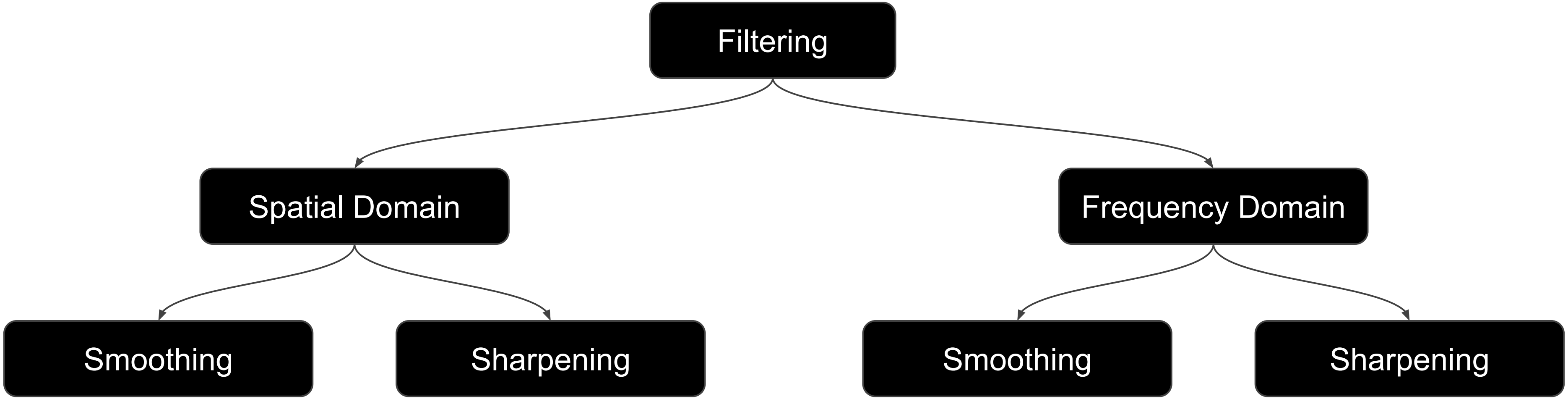
Image Filtering

John Aziz



- Simple Averaging Filter
- Weighted Averaging Filter
- Median Filter

- Laplacian
- Laplacian Variant
- Sobel

Low Pass Filter:

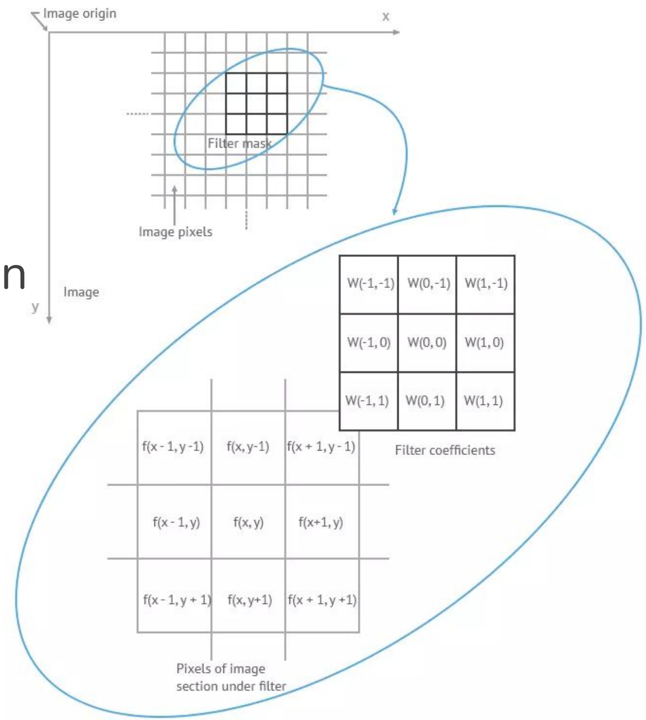
- Butterworth
- Gaussian

High Pass Filter:

- Butterworth
- Gaussian

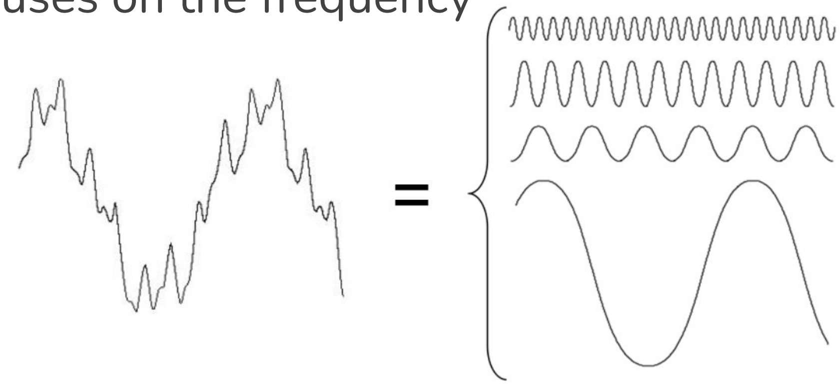
Spatial Domain Filtering

- It is a technique that is used directly on **pixels** of an image.
- **Mask** is usually considered to be added in size so that it has specific center pixel.
- This mask is **moved** on the image such that the **center** of the mask traverses all image pixels.



Frequency Domain Filtering

- They are used for smoothing and sharpening of image by removal of high or low frequency components.
- Sometimes it is possible of removal of very high and very low frequency.
- Frequency domain filters are different from spatial domain filters as it basically focuses on the frequency of the images.



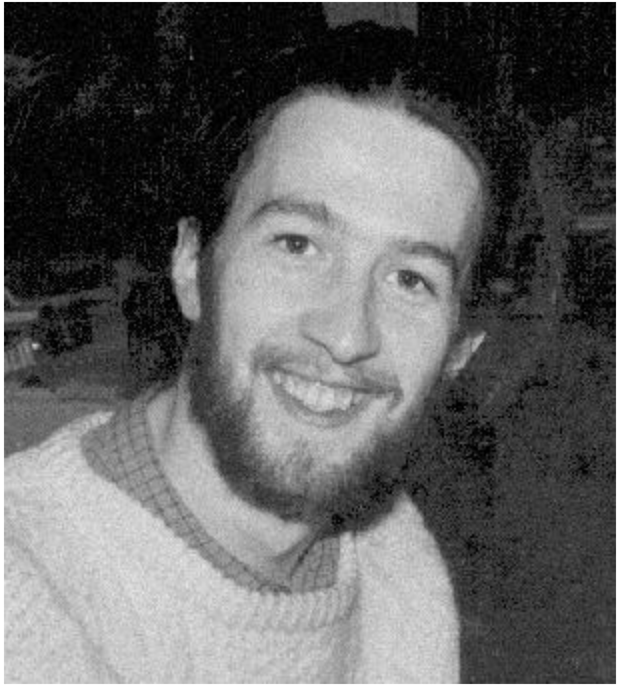
Spatial Domain: Smoothing

Simple Averaging Filter

- Average (or mean) filtering is a method of ‘smoothing’ images by reducing the amount of intensity variation between neighbouring pixels.
- The average filter works by moving through the image pixel by pixel, replacing each value with the average value of neighbouring pixels, including itself.
- Reduces irrelevant details in image.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

Simple Averaging Filter



Gaussian Noise

Mean filtering this with a 3×3 neighborhood

Simple Averaging Filter

Salt & Pepper Noise

Mean filtering this with a 3×3 neighborhood

Weighted Averaging Filter

- In weighted average filter, we gave more weight to the center value, due to which the contribution of center becomes more than the rest of the values. Due to weighted average filtering, we can control the blurring of image.
- More Effective smoothing effect.

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

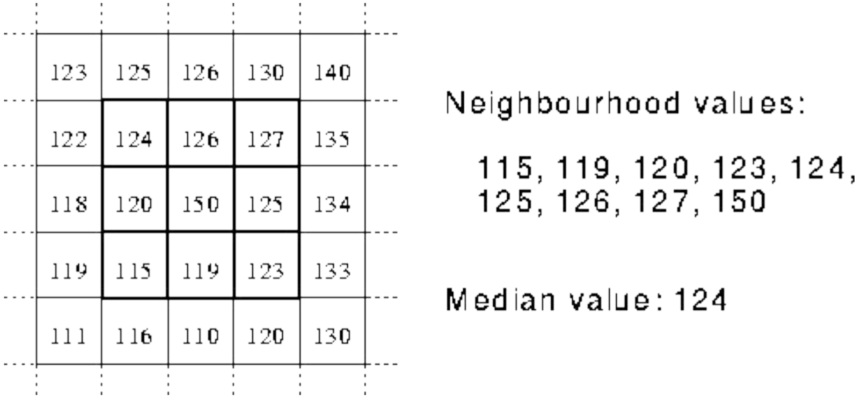
Weighted Averaging Filter

Salt & Pepper Noise

Mean filtering this with a 3×3 neighborhood

Median Filter

- The median filter is normally used to reduce noise in an image, somewhat like the mean filter. However, it often does a better job than the mean filter of preserving useful detail in the image.
- Removes Salt & Pepper Noise
- Will not remove Gaussian Noise
- Details are lost

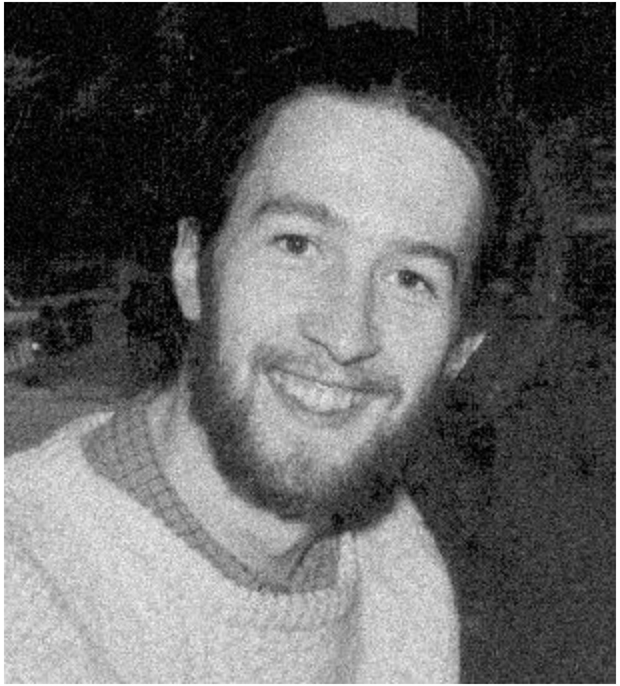


Median Filter

Salt & Pepper Noise

Median filtering this with a 3×3 neighborhood

Median Filter



Gaussian Noise

Median filtering this with
a 3×3 neighborhood

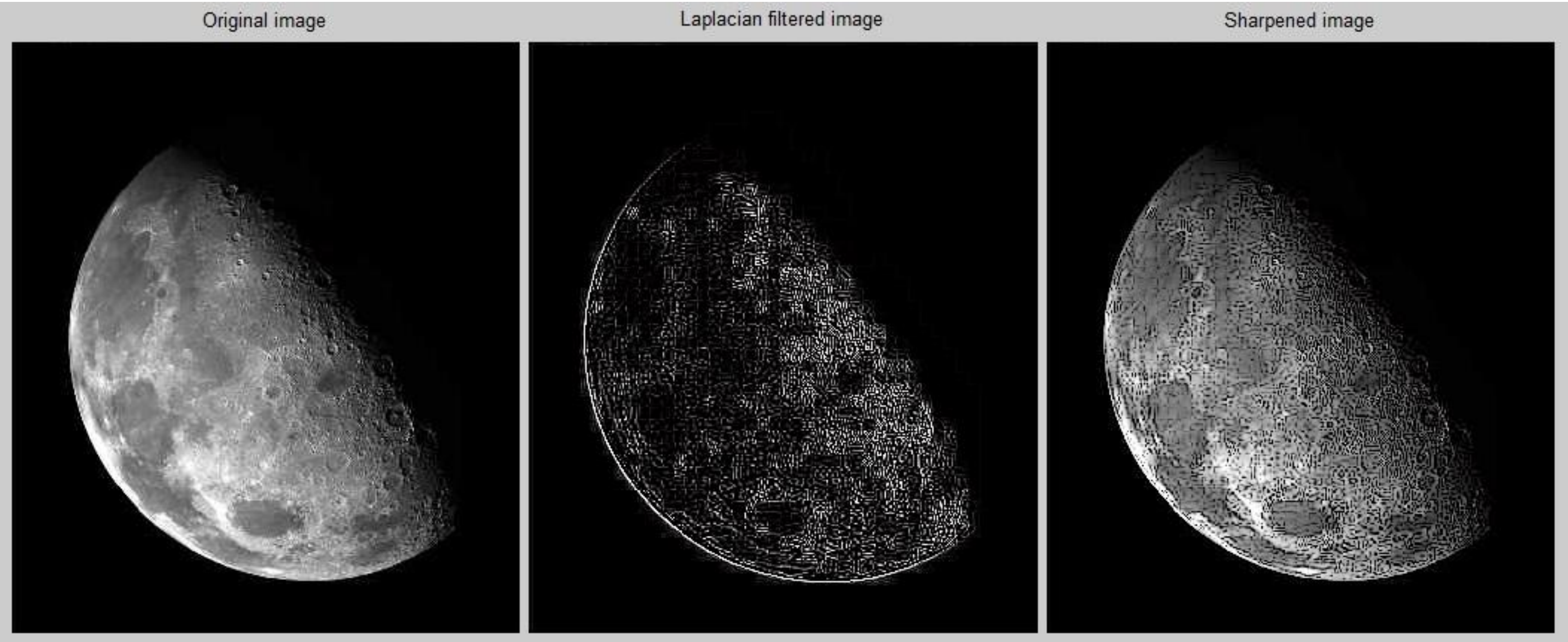
Spatial Domain: Sharpening

Laplacian Filter

- A Laplacian filter is an edge detector used to compute the second derivatives of an image, measuring the rate at which the first derivatives change.
- This determines if a change in adjacent pixel values is from an edge or continuous progression.
- Using the Laplacian filter we detect the edges in the whole image at once.

0	1	0
1	−4	1
0	1	0

Laplacian Filter



Laplacian Variant Filter

- Same as laplacian filter.
- This determines if a change in adjacent pixel values is from an edge or continuous progression.
- Using the Laplacian variant filter we detect the edges in the whole image at once.

1	1	1
1	−8	1
1	1	1

Sobel Filter

- It is a discrete differentiation gradient-based operator.
- It computes the gradient approximation of image intensity function for image edge detection.
- At the pixels of an image, the Sobel operator produces either the normal to a vector or the corresponding gradient vector.
- It uses two 3 x 3 kernels or masks which are convolved with the input image to calculate the vertical and horizontal derivative approximations respectively.

where Gx is for x direction and Gy for y direction.

The sobel masks (3x3):

For x-Direction:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

For Y-direction:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Sobel Filter

Advantages

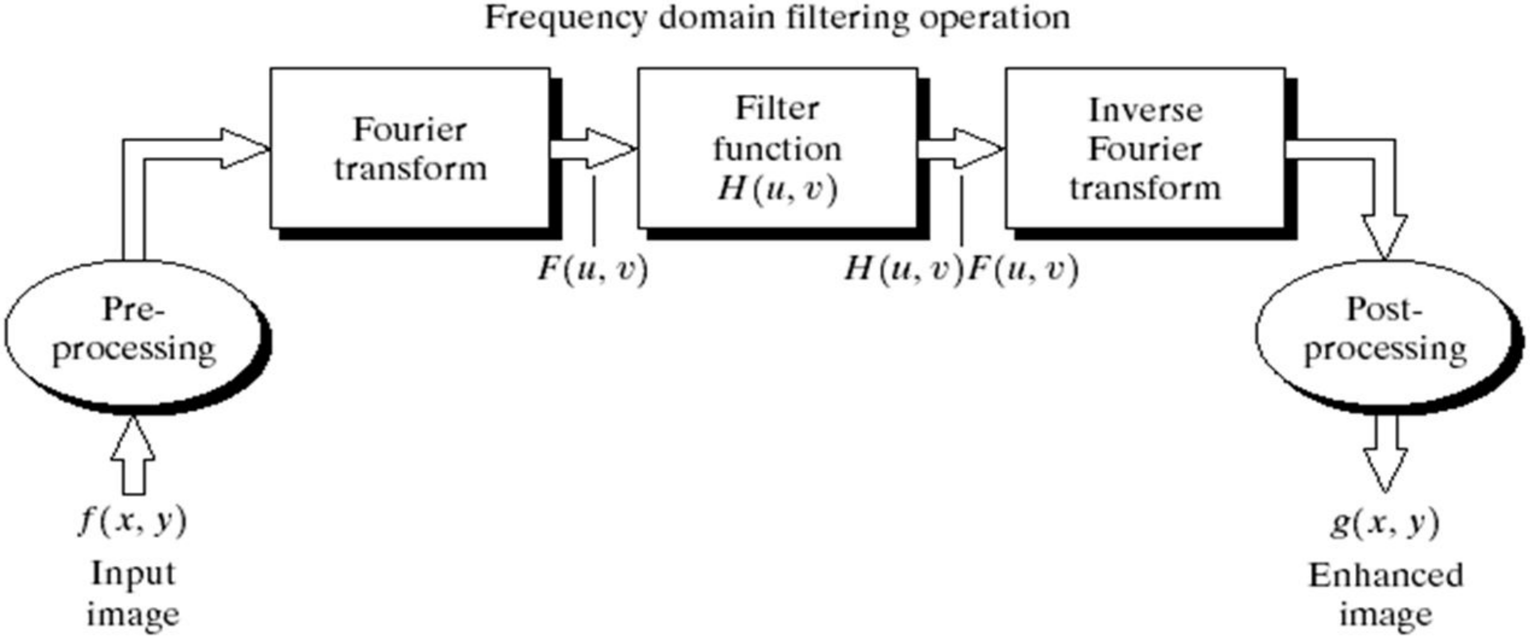
- Simple and time efficient computation
- Very easy at searching for smooth edges

Disadvantages

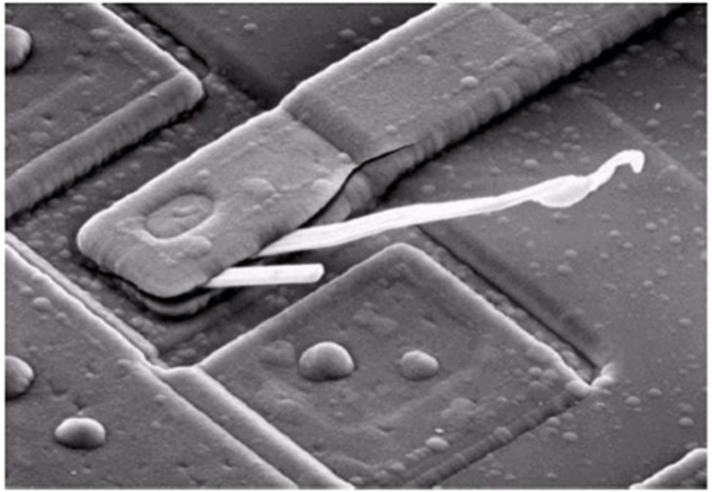
- Diagonal direction points are not preserved always
- Sensitive to noise
- Not very accurate in edge detection
- Detect with thick and rough edges does not give appropriate results

Sobel Filter

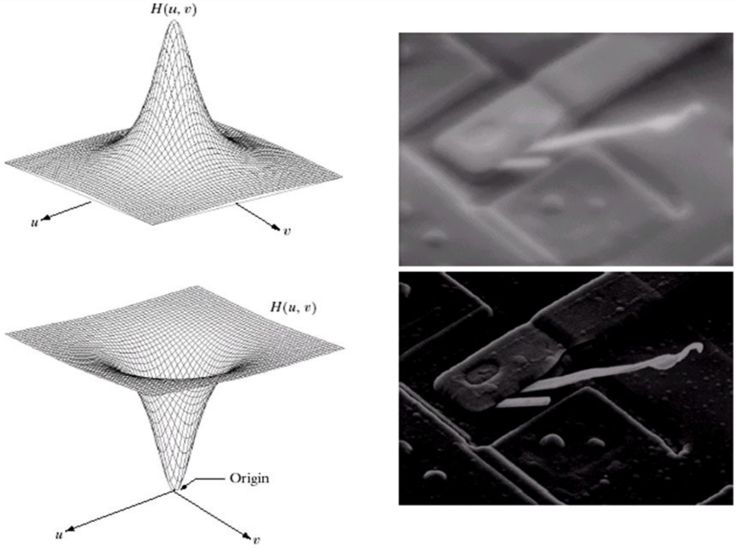
Frequency Domain Filtering



Some Basic Frequency Domain Filters

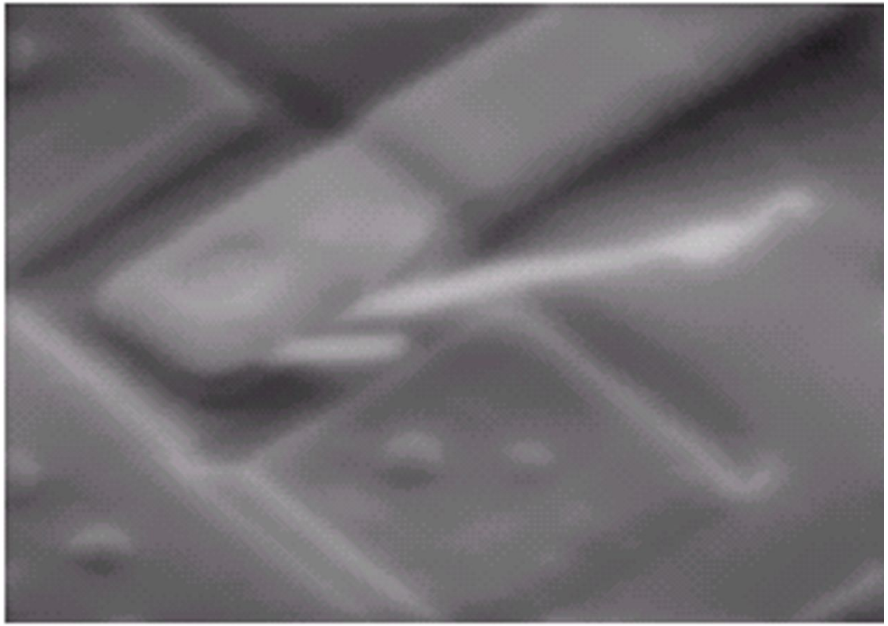


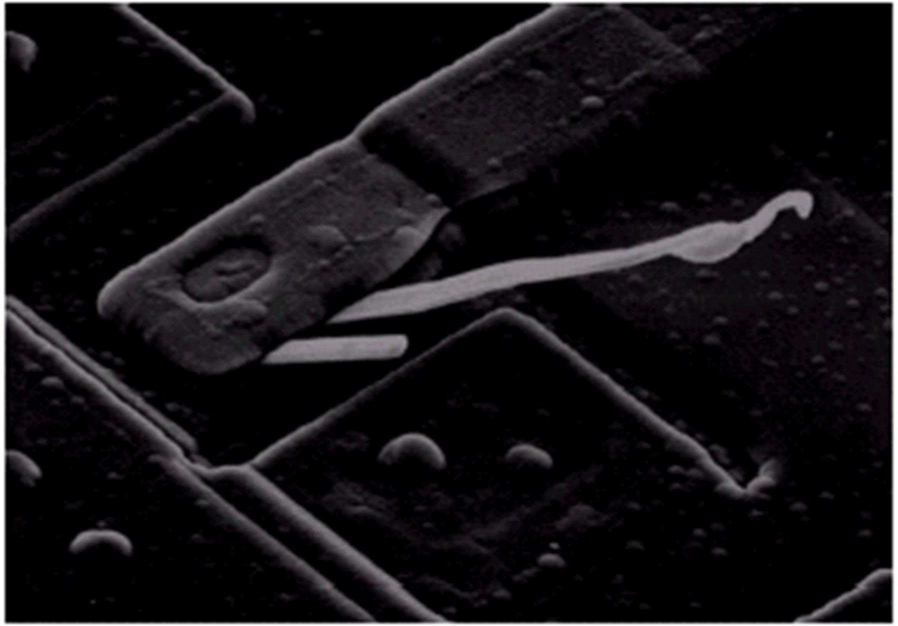
Low Pass Filter



High Pass Filter

Some Basic Frequency Domain Filters



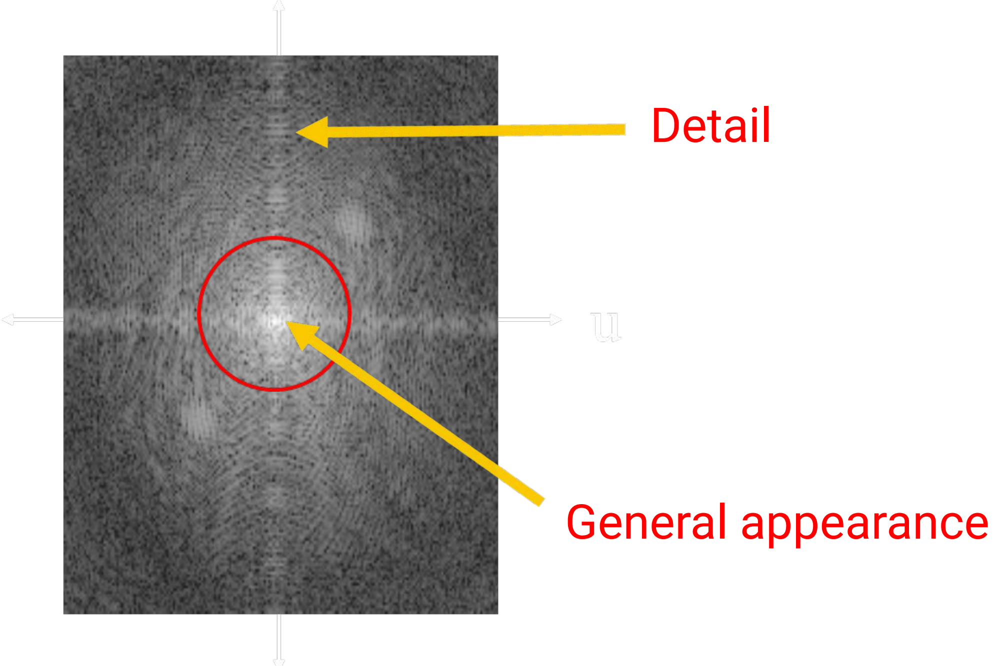


Some Basic Frequency Domain Filters

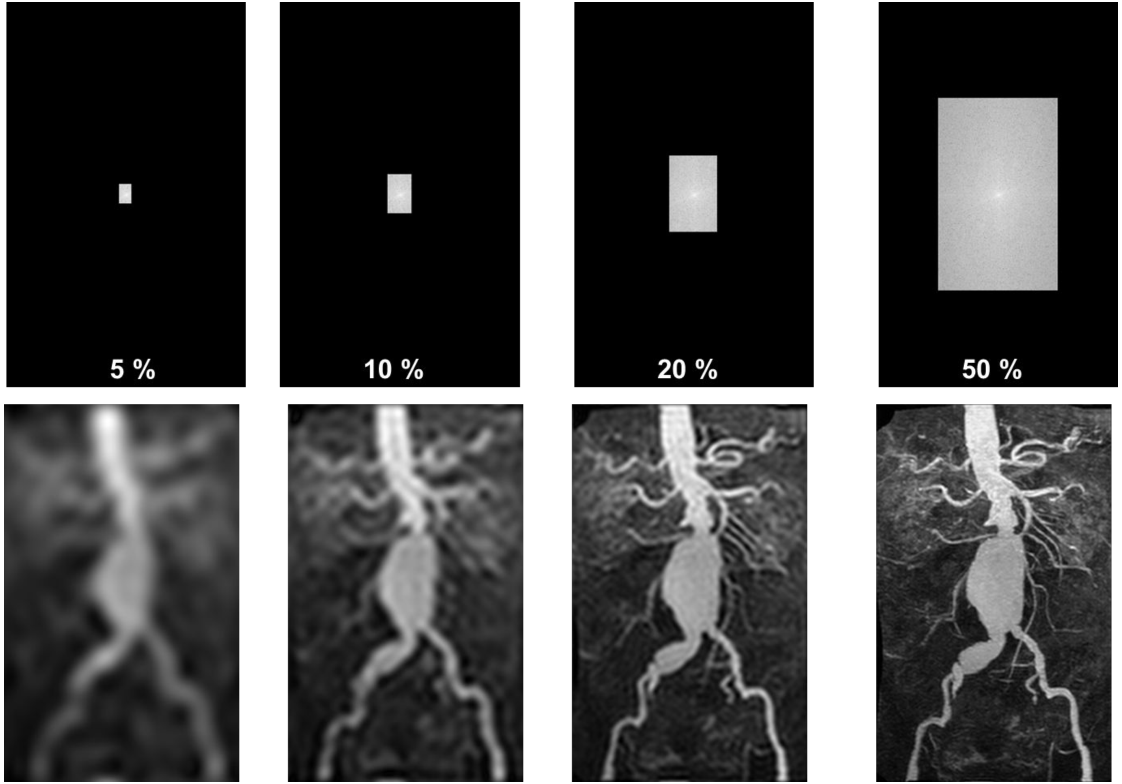
Image

FT

Frequency Domain (log magnitude)



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:

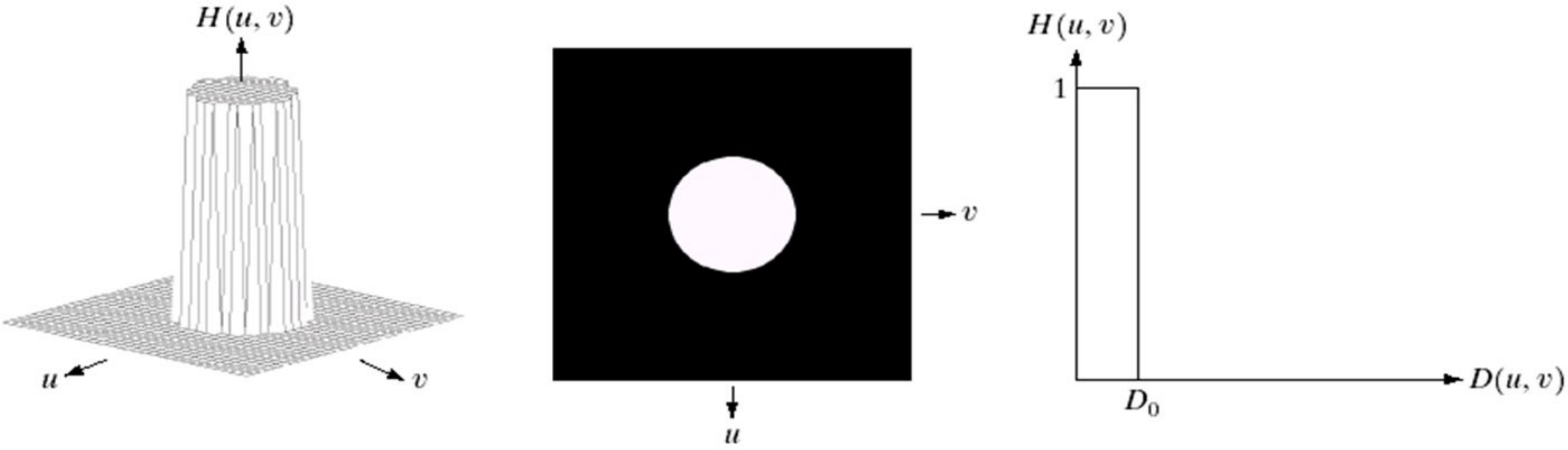
F(u,v) is the Fourier transform of the image being filtered

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

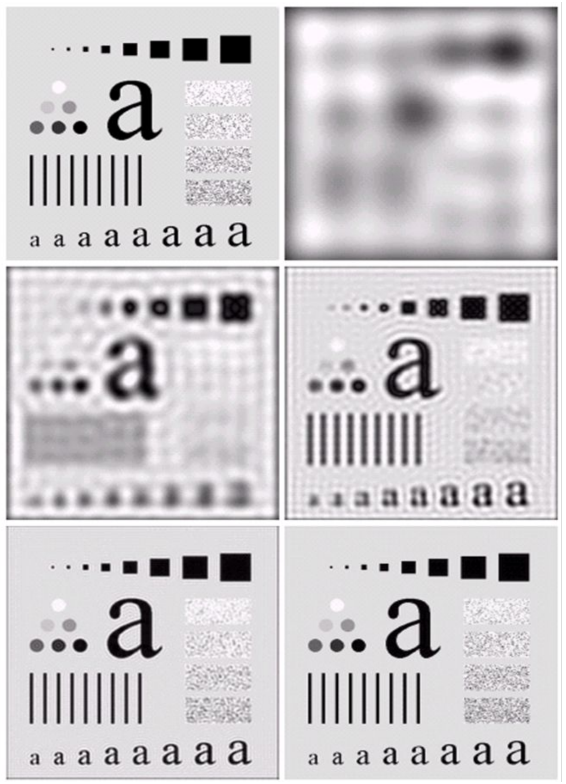


changing the distance changes the behaviour of the filter

Ideal Low Pass Filter (continue…)

Original image

Result of filtering with ideal low pass filter of radius 5



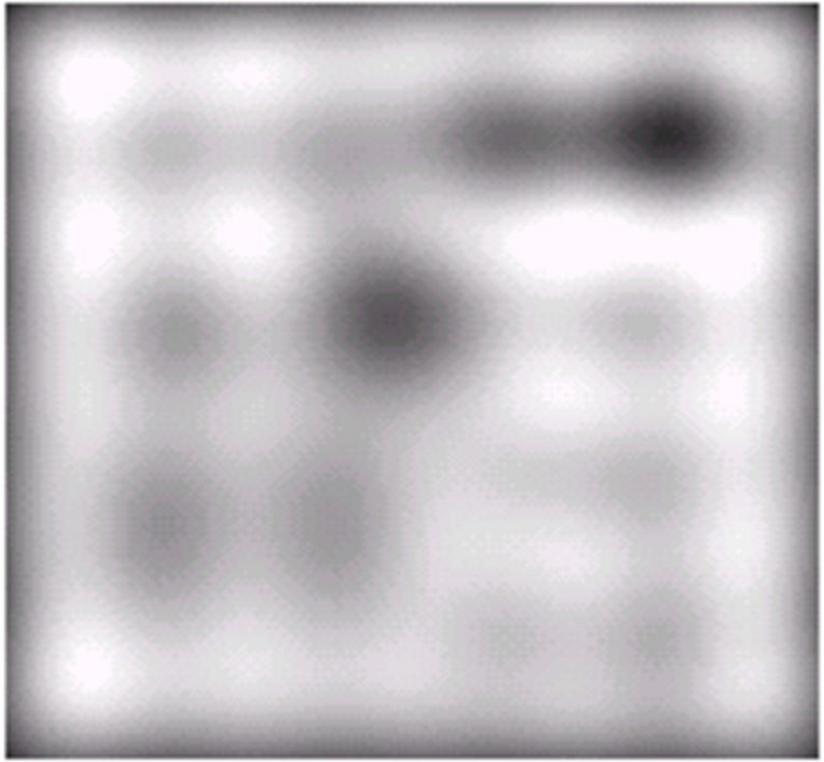
Result of filtering with ideal low pass filter of radius 15

Result of filtering with ideal low pass filter of radius 30

Result of filtering with ideal low pass filter of radius 80

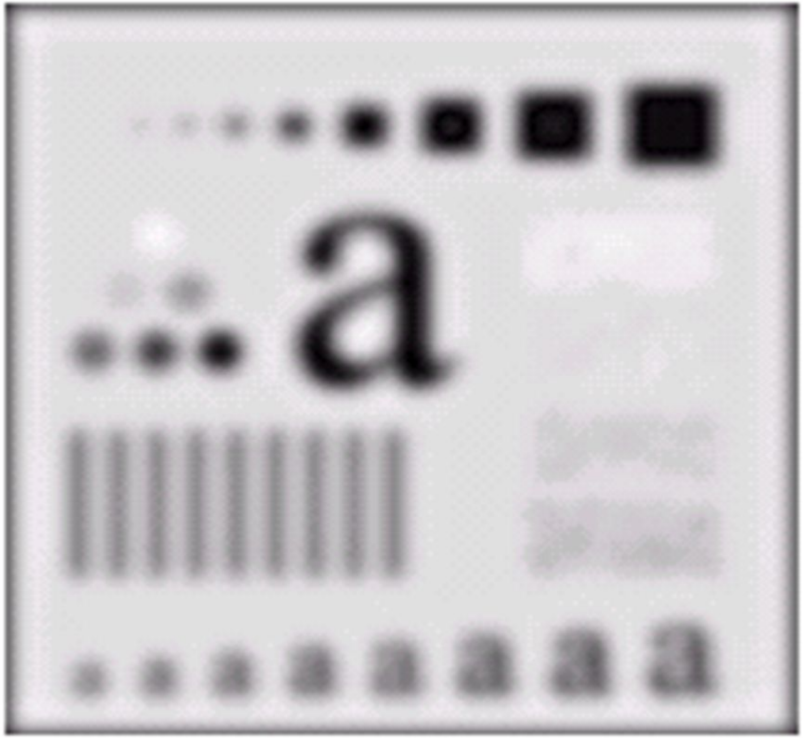
Result of filtering with ideal low pass filter of radius 230

Ideal Low Pass Filter (continue…)



Result of filtering with ideal low pass filter of radius 5

Ideal Low Pass Filter (continue…)

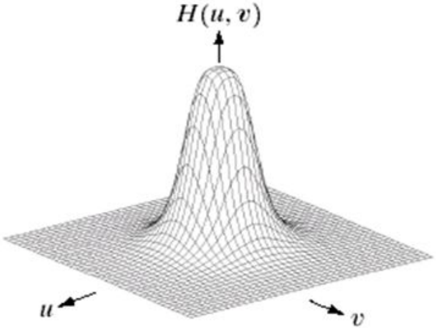


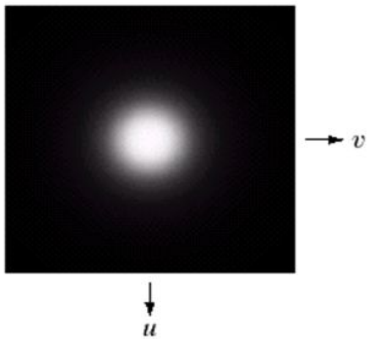
Result of filtering with Butterworth filter of order 2 and cutoff radius 15

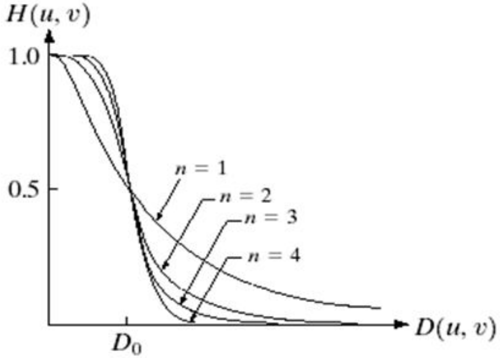
Butterworth Low-Pass Filters

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

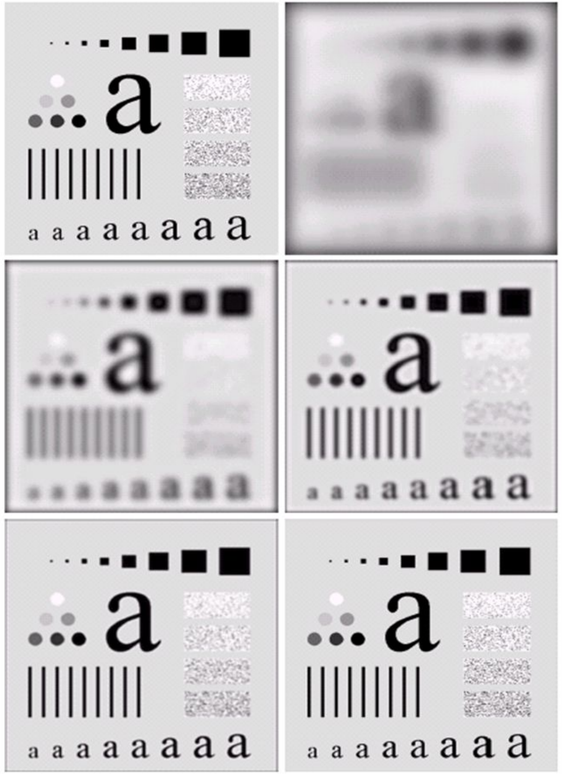
$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$







Butterworth Low-Pass Filter (continue…)



Original image

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

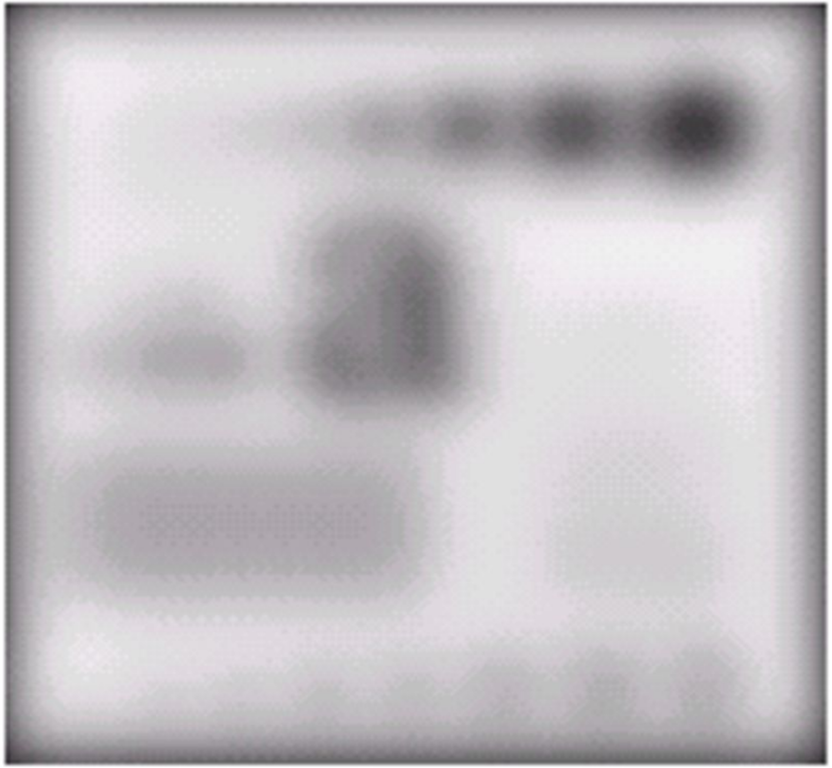
Result of filtering with Butterworth filter of order 2 and cutoff radius 15

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

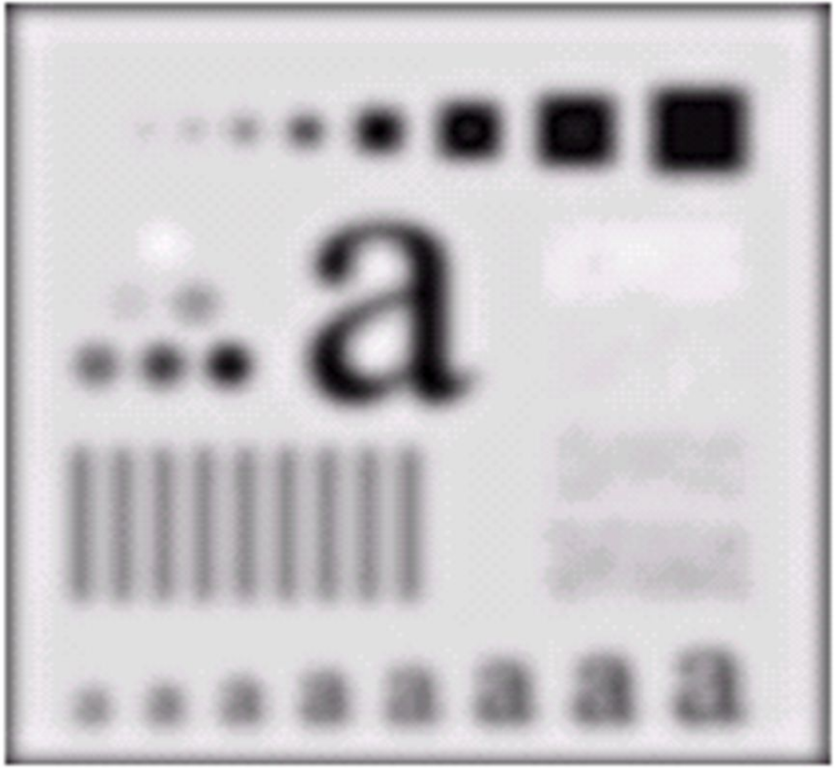
Result of filtering with Butterworth filter of order 2 and cutoff radius 230

Butterworth Low-Pass Filter (continue…)



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

Butterworth Low-Pass Filter (continue…)

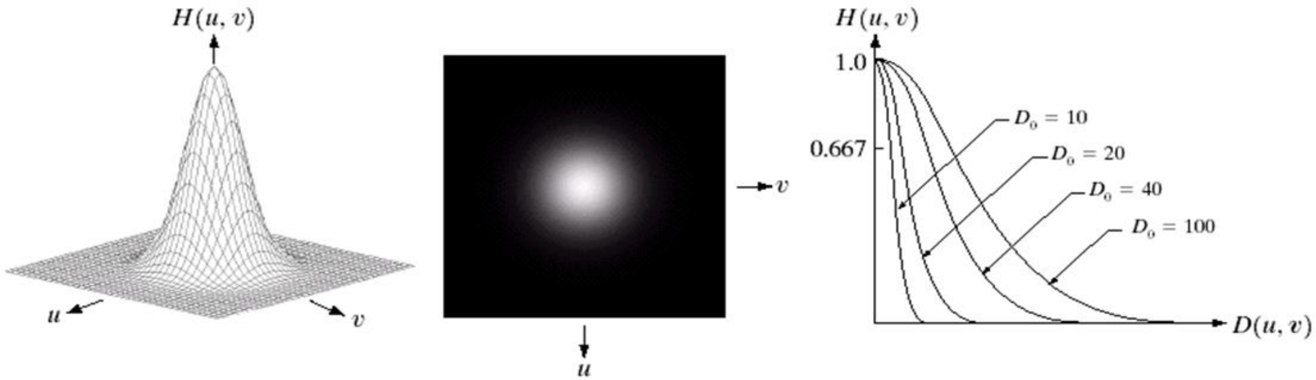


Result of filtering with Butterworth filter of order 2 and cutoff radius 15

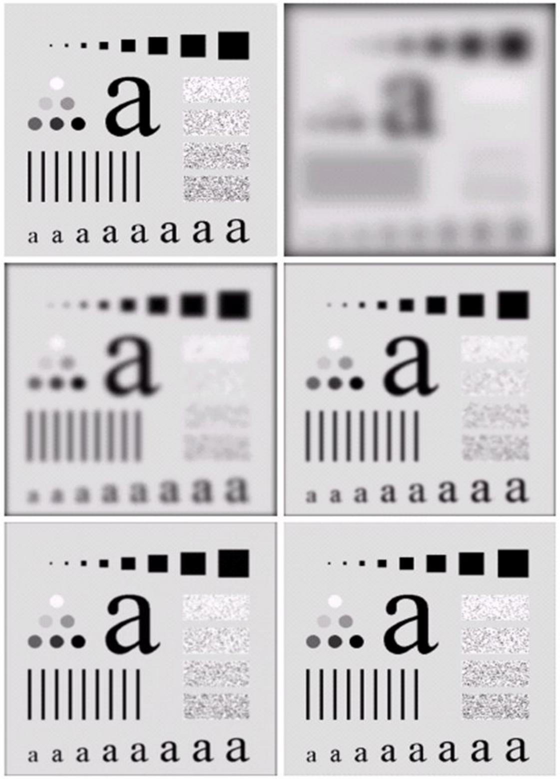
Gaussian Low-Pass Filters

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

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



Gaussian Low-Pass Filters (continue…)



Original image

Result of filtering with Gaussian filter with cutoff radius 5

Result of filtering with Gaussian filter with cutoff radius 15

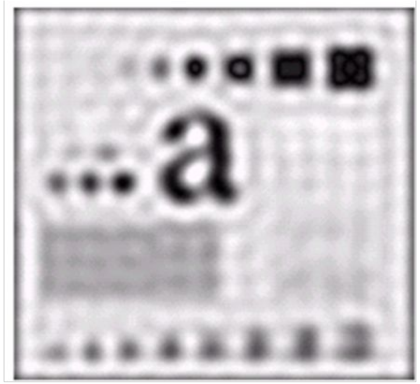
Result of filtering with Gaussian filter with cutoff radius 30

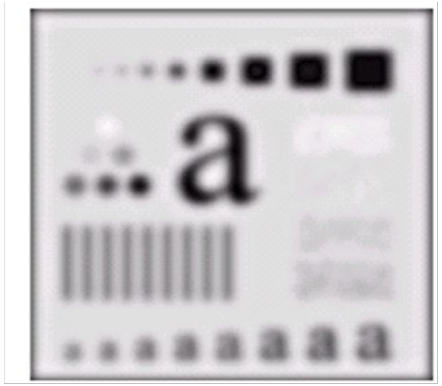
Result of filtering with Gaussian filter with cutoff radius 85

Result of filtering with Gaussian filter with cutoff radius 230

Low-Pass Filters Compared

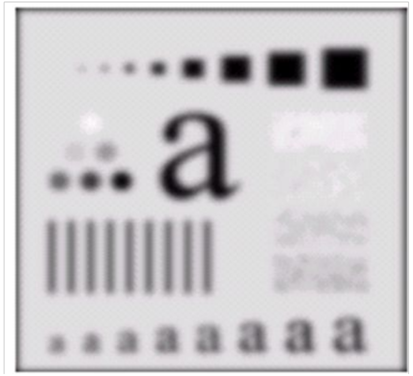
Result of filtering with ideal low pass filter of radius 15





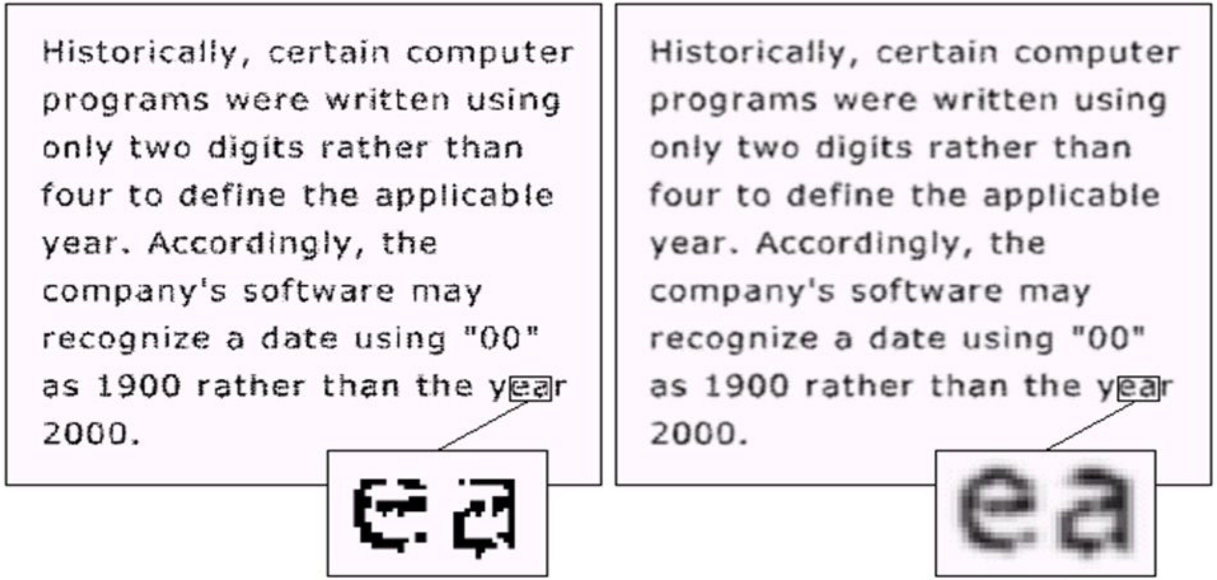
Result of filtering with Butterworth filter of order 2 and cutoff radius 15

Result of filtering with Gaussian filter with cutoff radius 15



Low-Pass Filtering Examples

A low pass Gaussian filter is used to connect broken text



Low-Pass Filtering Examples

Different low-Pass Gaussian filters used to remove blemishes in a photograph

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:

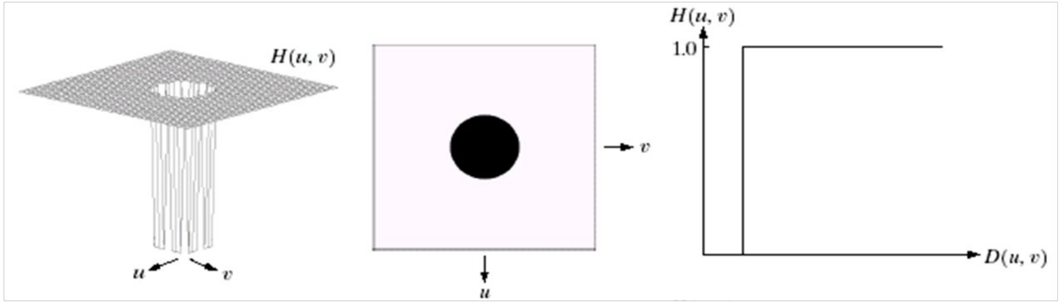
$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

Ideal High Pass Filters

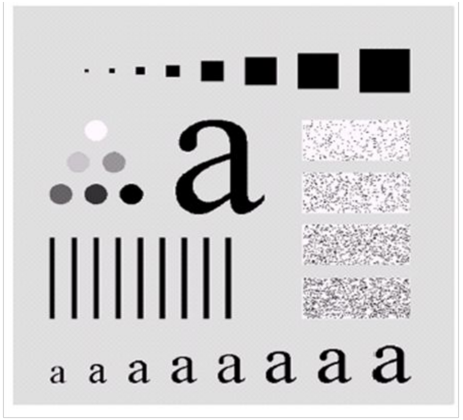
The ideal high pass filter is given as:

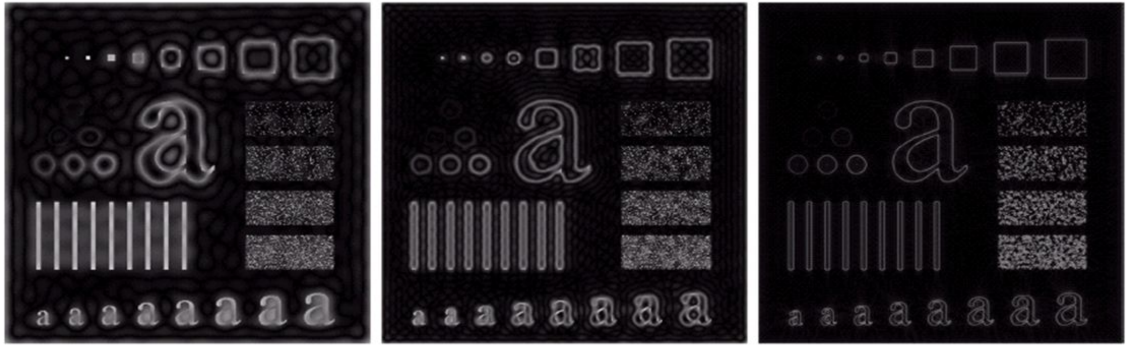
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

where D_0 is the cut off distance as before



Ideal High Pass Filters (continue…)



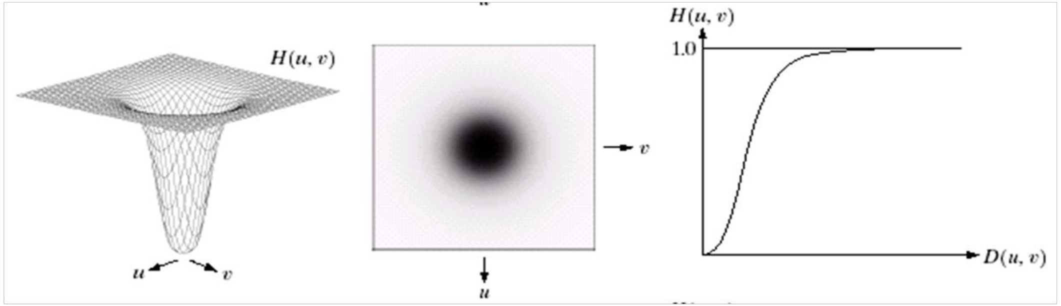


Butterworth High Pass Filters

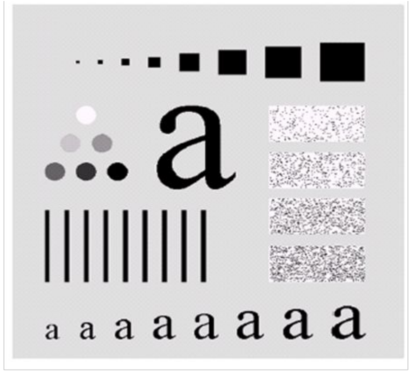
The Butterworth high pass filter is given as:

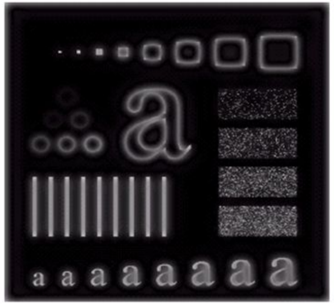
$$H(u,v) = \frac{1}{1 + [D_0 / D(u,v)]^{2n}}$$

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

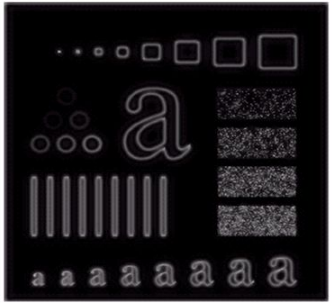


Butterworth High Pass Filters (continue…)

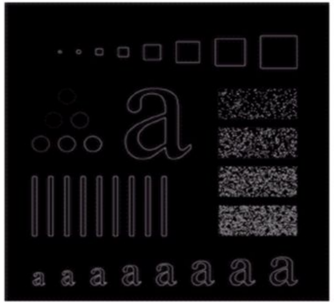




Results of Butterworth high pass filtering of order 2 with D_0 = 15



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



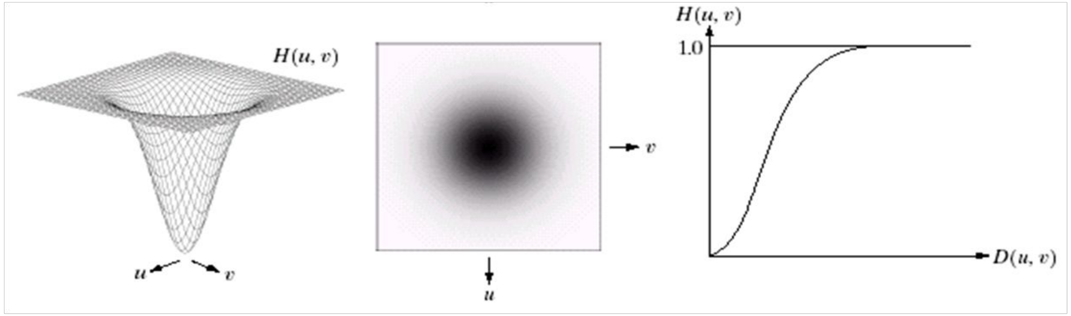
Results of Butterworth high pass filtering of order 2 with D_0 = 80

Gaussian High Pass Filters

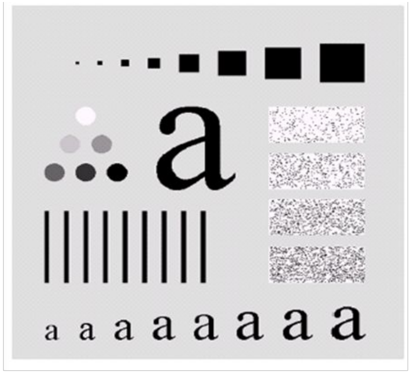
The Gaussian high pass filter is given as:

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

where D_0 is the cut off distance as before



Gaussian High Pass Filters (continue…)

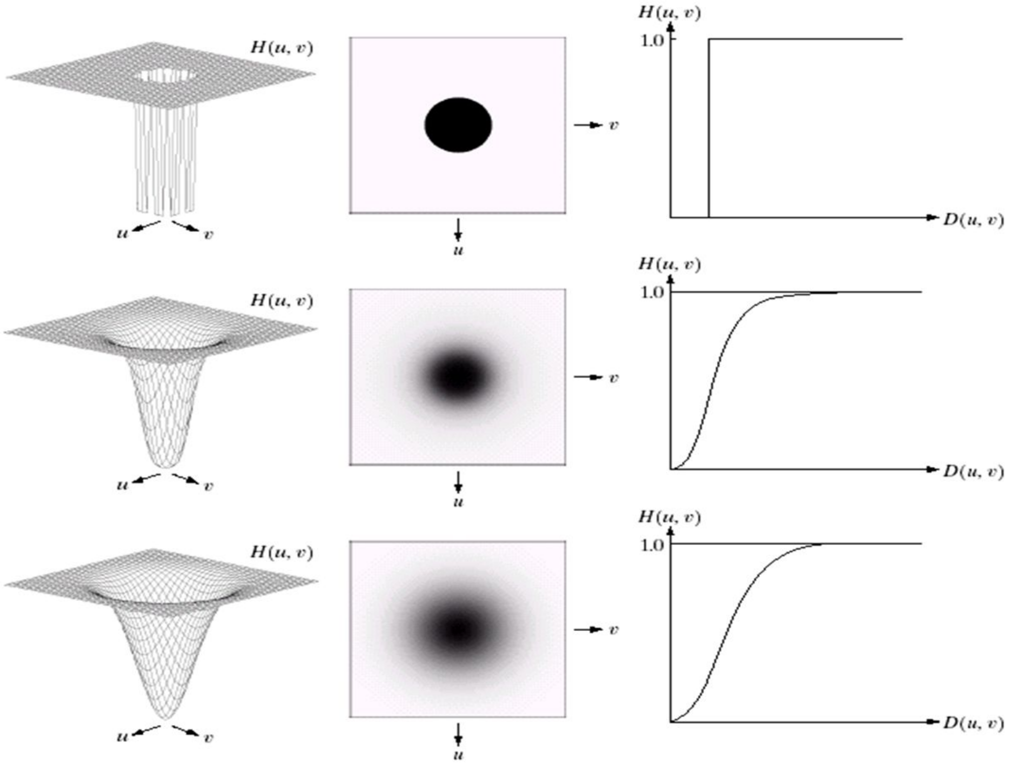


Results of Gaussian high pass filtering with D_0 = 15

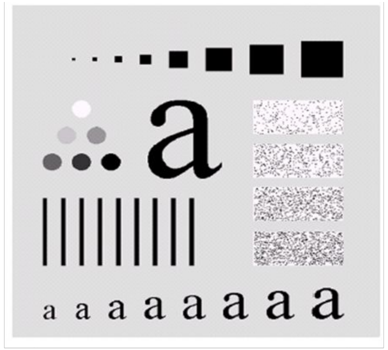
Results of Gaussian high pass filtering with D_0 = 30

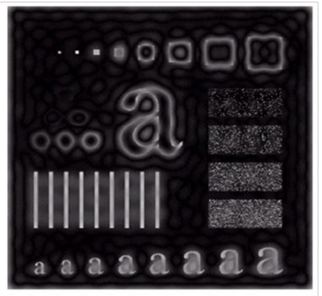
Results of Gaussian high pass filtering with D_0 = 80

High-Pass Filter Comparison

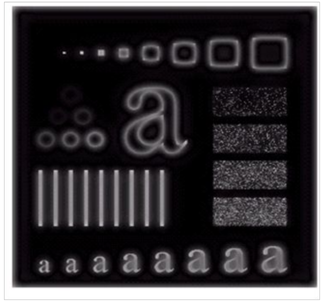


High-Pass Filter Comparison

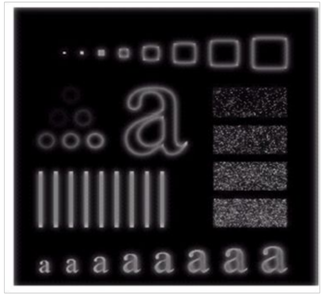




Results of ideal high pass filtering with D_0 = 15



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



Results of Gaussian high pass filtering with D_0 = 15

Fast Fourier Transform

- 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!**

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**.

Thank You