Mahidol University *Wisdom of the Land*

Chapter 11

Image Enhancement in the Frequency Domain

By Dr. Narit Hnoohom

Image Restoration

Noisy image
***f*(x,y)**

Better quality image
***g*(x,y)**

Cause:

a) Noise can be added to images during image acquisition (faulty CCD elements);
b) during image transmission (channel interference).

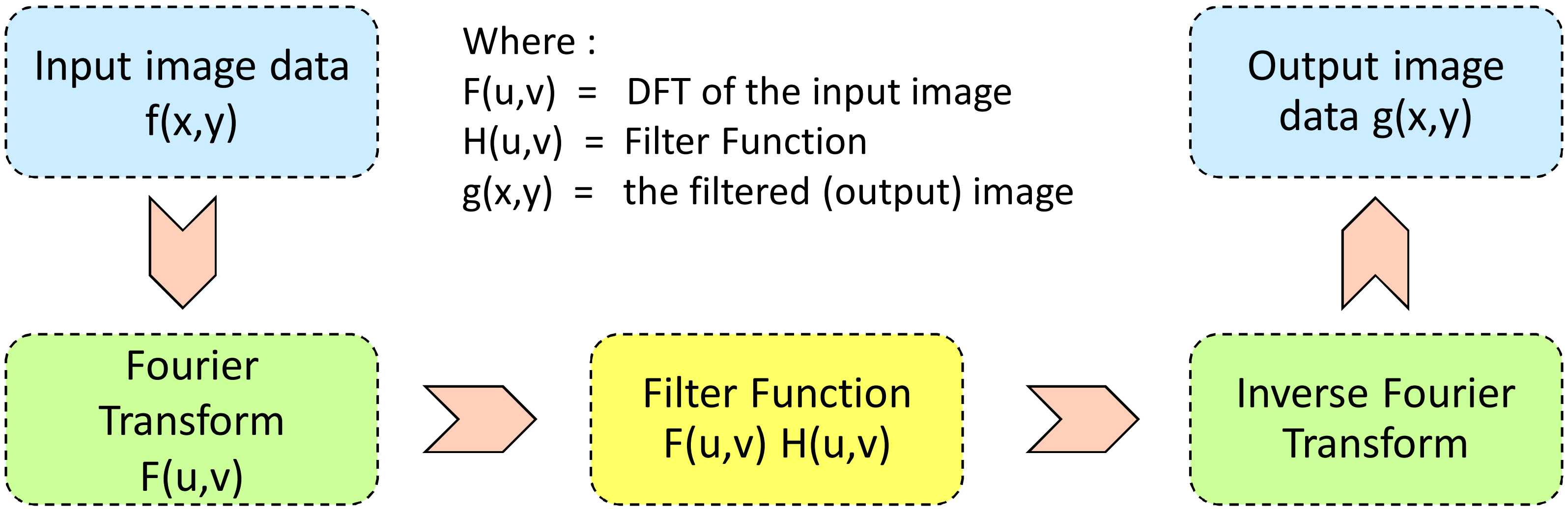
Restoration:

a) Spatial domain
b) Frequency domain

Image Enhancement in the Frequency Domain

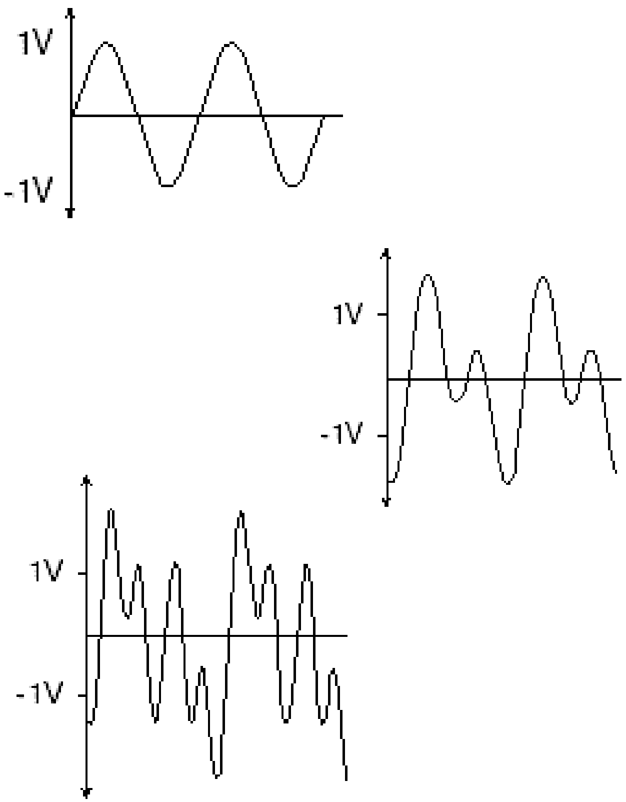
- Image enhancement in the frequency domain use of the same principles of image enhancement in the spatial domain by changing processing in the frequency domain.

- Frequency Domain : Frequency - Base

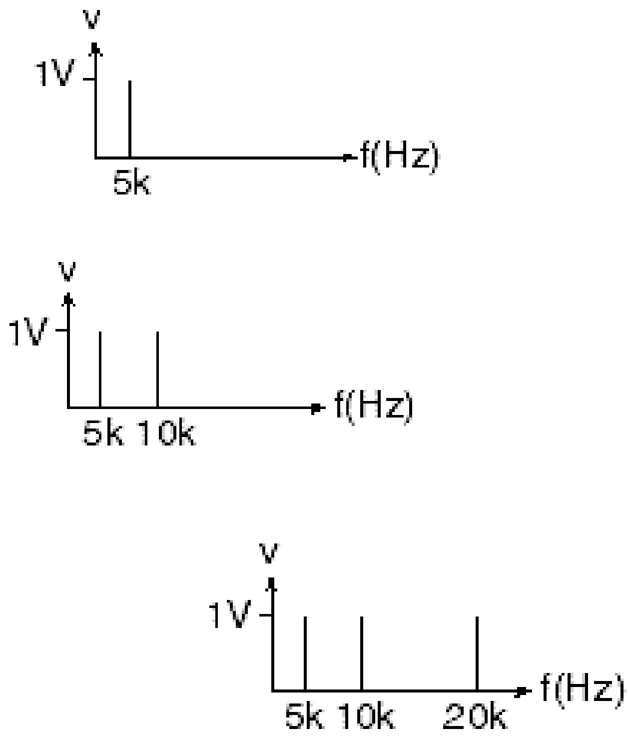


Example

Spatial Domain

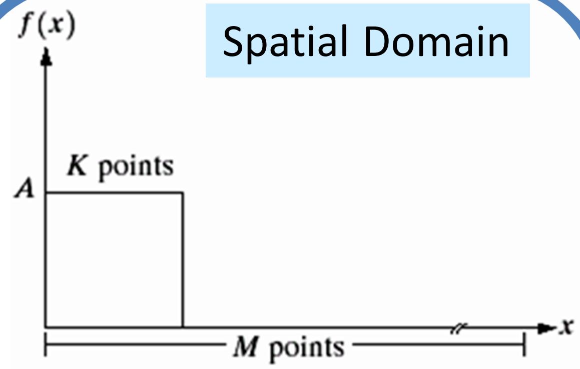


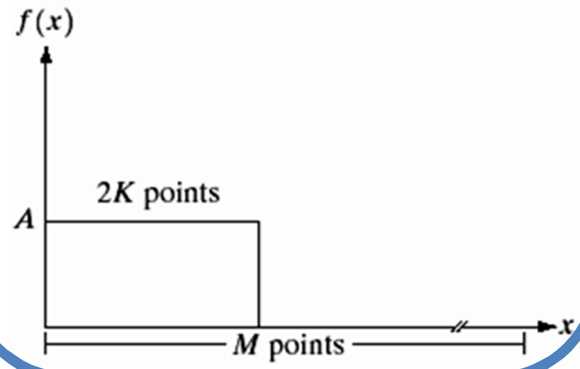
Frequency Domain



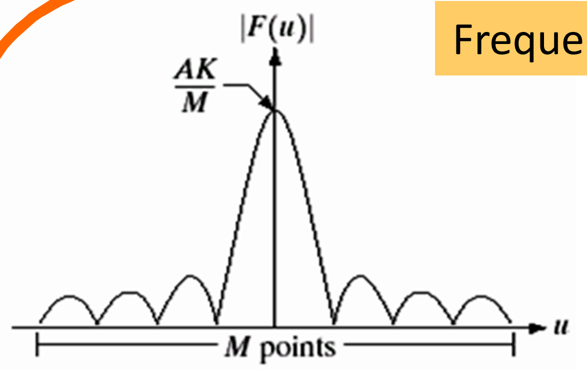
Example

Spatial Domain

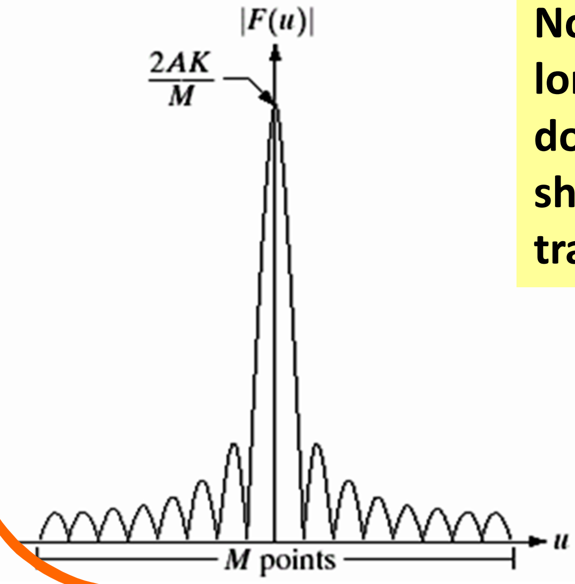




Frequency Domain



Notice that the longer the time domain signal, The shorter its Fourier transform.



Discrete Fourier Transform (DFT)

- The discrete Fourier transform (DFT) is a specific kind of Fourier transform, used in Fourier analysis.
- It transforms one function into another, which is called the frequency domain representation, or simply the DFT, of the original function (which is often a function in the time domain).
- But the DFT requires an input function that is discrete and whose non-zero values have a limited (finite) duration.
- Advantages of DFT
    - Easier image filtering, just attenuate (cut off) the target frequency components.
    - More efficient computation

One-Dimensional DFT

1-D DFT: $$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M}$$ u = 0,…,M-1

1-D IDFT: $$f(x) = \sum_{u=0}^{M-1} F(u)e^{j2\pi ux/M}$$ x = 0,…,M-1

F(*u*) can be written as

$$F(u) = R(u) + jI(u) \quad \text{or} \quad F(u) = |F(u)|e^{-j\phi(u)}$$

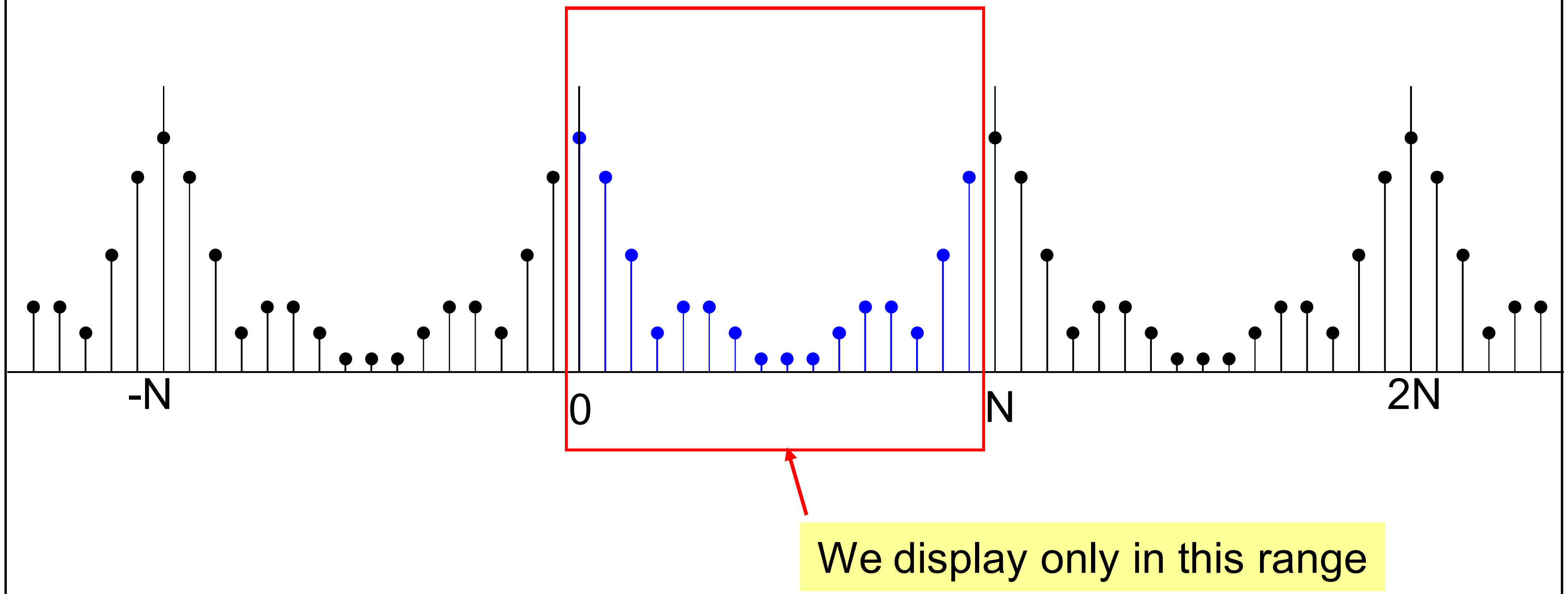
where

$$|F(u)| = \sqrt{R(u)^2 + I(u)^2} \qquad \phi(u) = \tan^{-1}\left(\frac{I(u)}{R(u)}\right)$$

Example : 1-D DFT

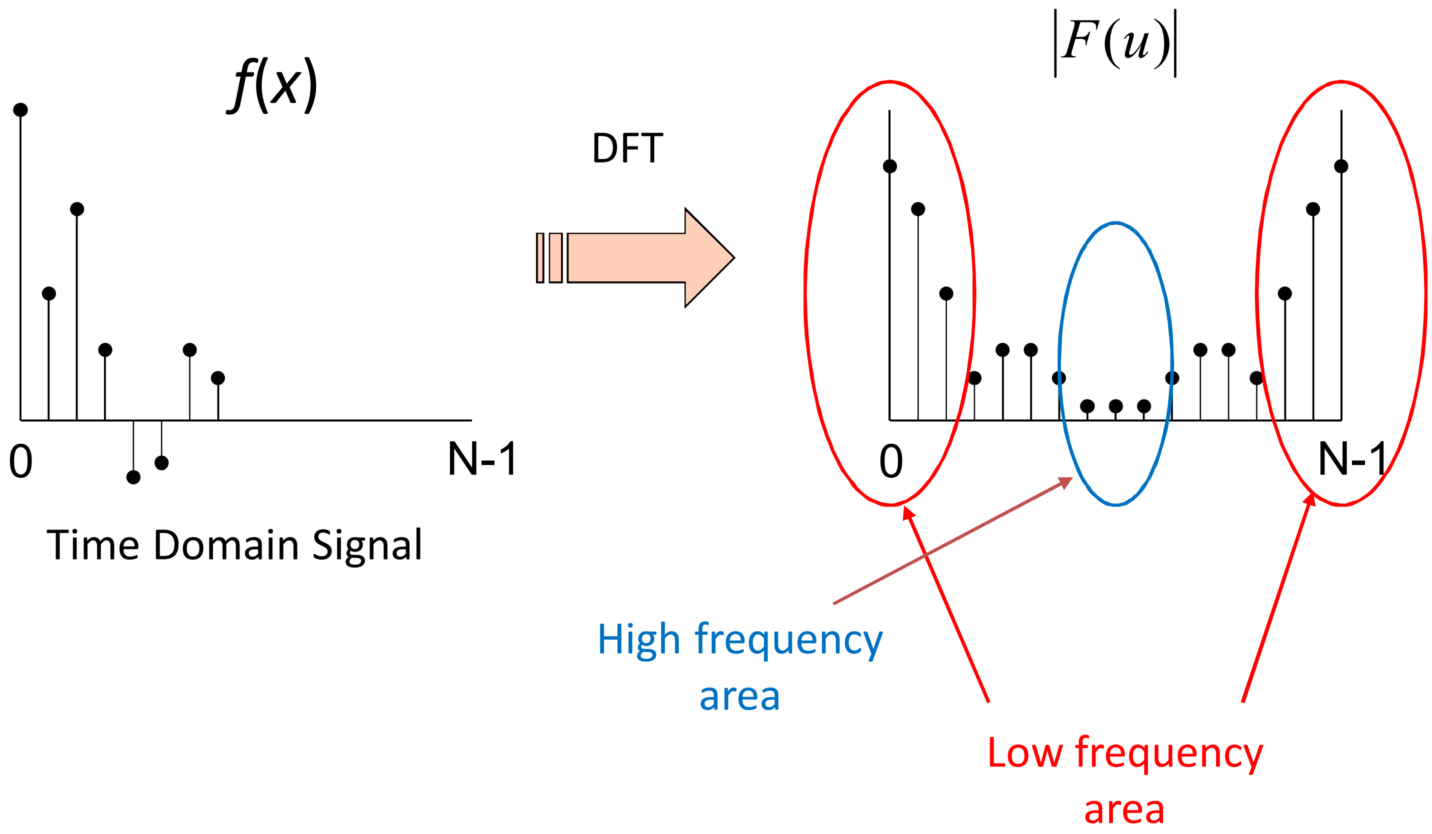
From DFT:

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



DFT repeats itself every N points (Period = N) but we usually display it for $n = 0$,…, N-1

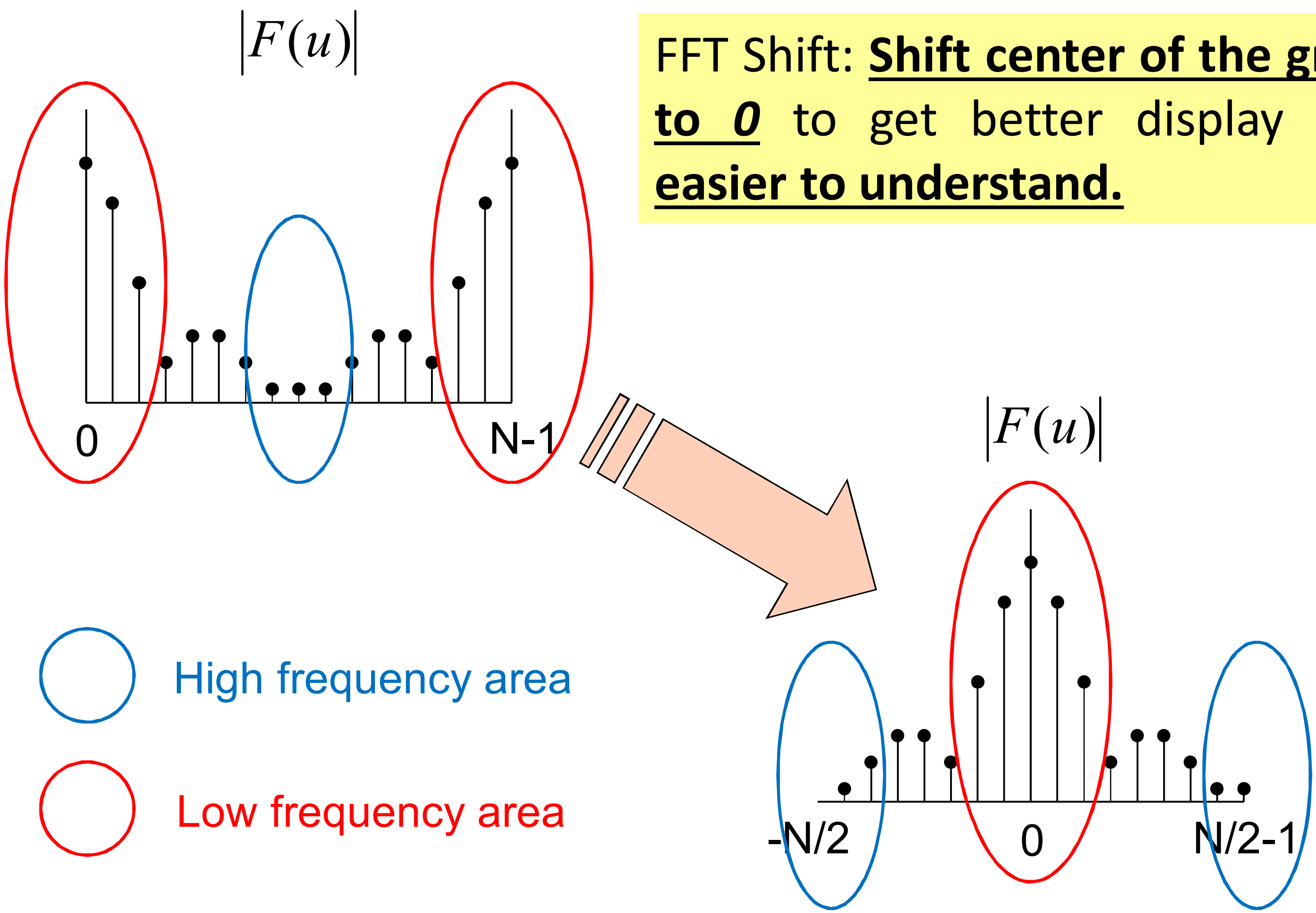
Example : Conventional Display for 1-D DFT



The graph $F(u)$ is not easy to understand !

Conventional Display for DFT : FFT Shift

FFT Shift: **<u>Shift center of the graph *F*(*u*) to *0*</u>** to get better display which is **<u>easier to understand.</u>**



Two-Dimensional DFT

2-D DFT:

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

u = frequency in *x* direction, *u* = 0 ,…, *M*-1
v = frequency in *y* direction, *v* = 0 ,…, *N*-1

2-D IDFT:

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

x = 0 ,…, *M*-1
y = 0 ,…, *N*-1

Two-Dimensional DFT

F(*u*) can be written as

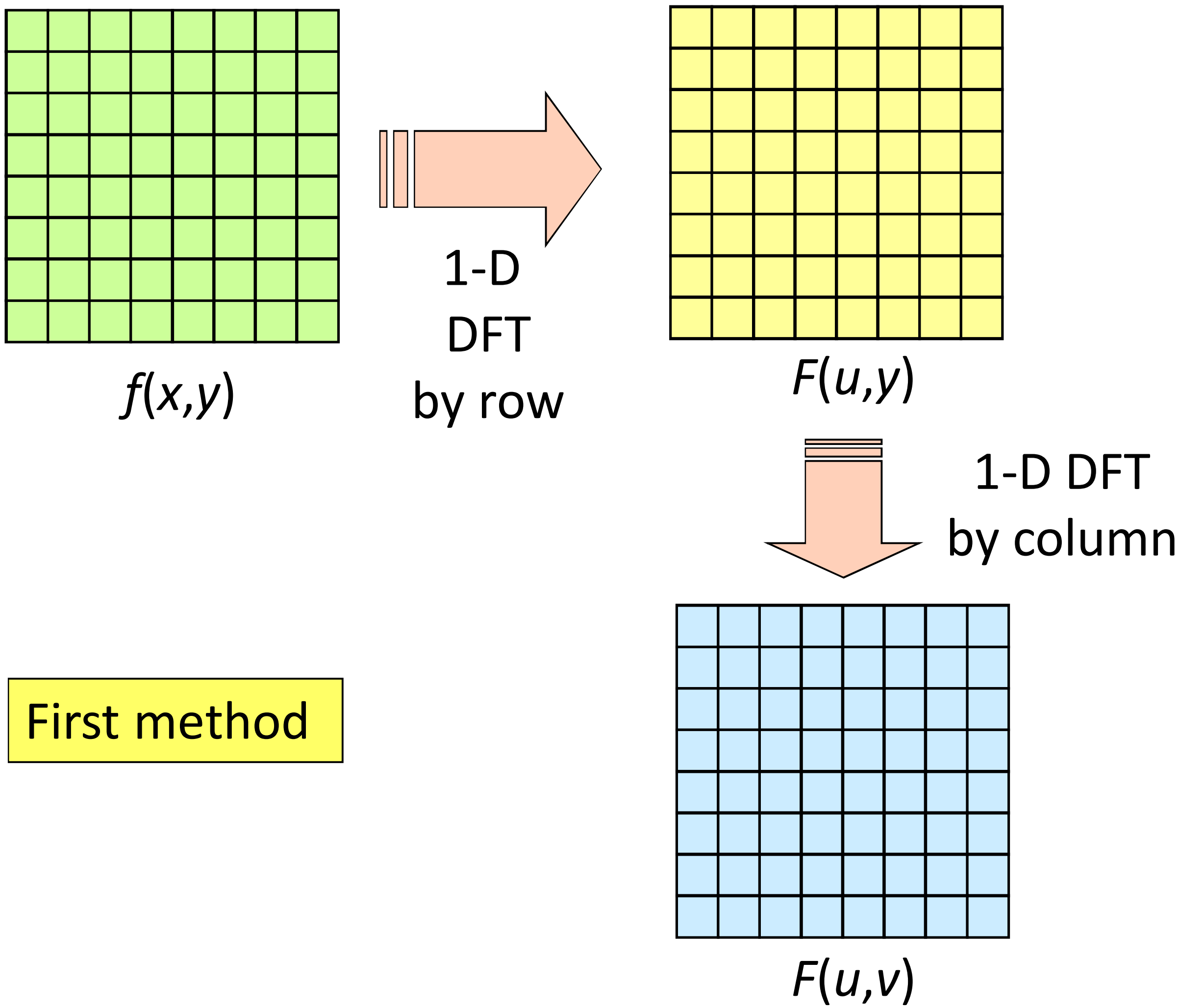
$$F(u) = R(u) + jI(u) \quad \text{or} \quad F(u) = |F(u)| e^{-j\phi(u)}$$

where

$$|F(u)| = \sqrt{R(u)^2 + I(u)^2} \qquad \phi(u) = \tan^{-1}\left(\frac{I(u)}{R(u)}\right)$$

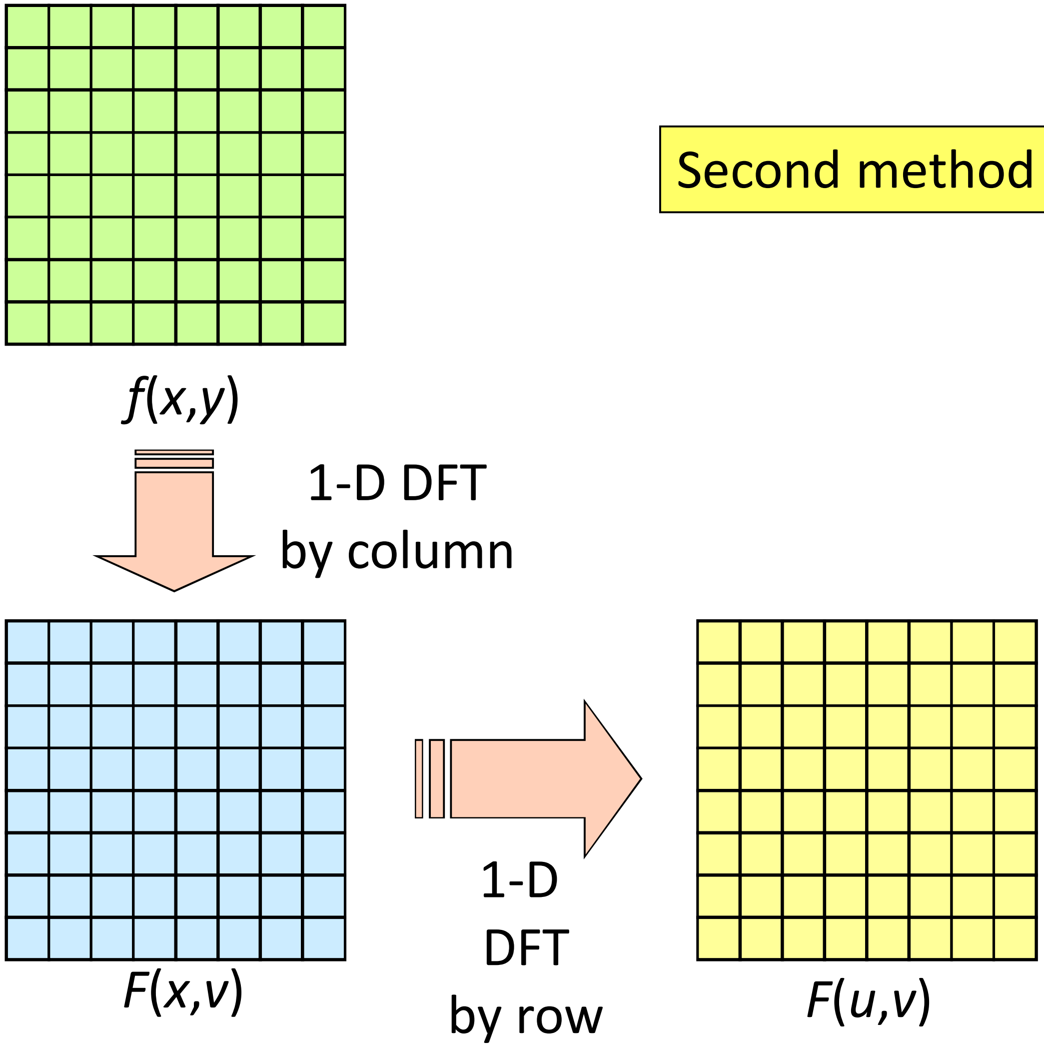
For the purpose of viewing, we usually display only the Magnitude part of *F*(*u*,*v*)

How to Perform 2-D DFT by Using 1-D DFT



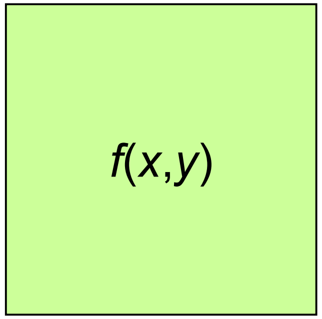
How to Perform 2-D DFT by Using 1-D DFT

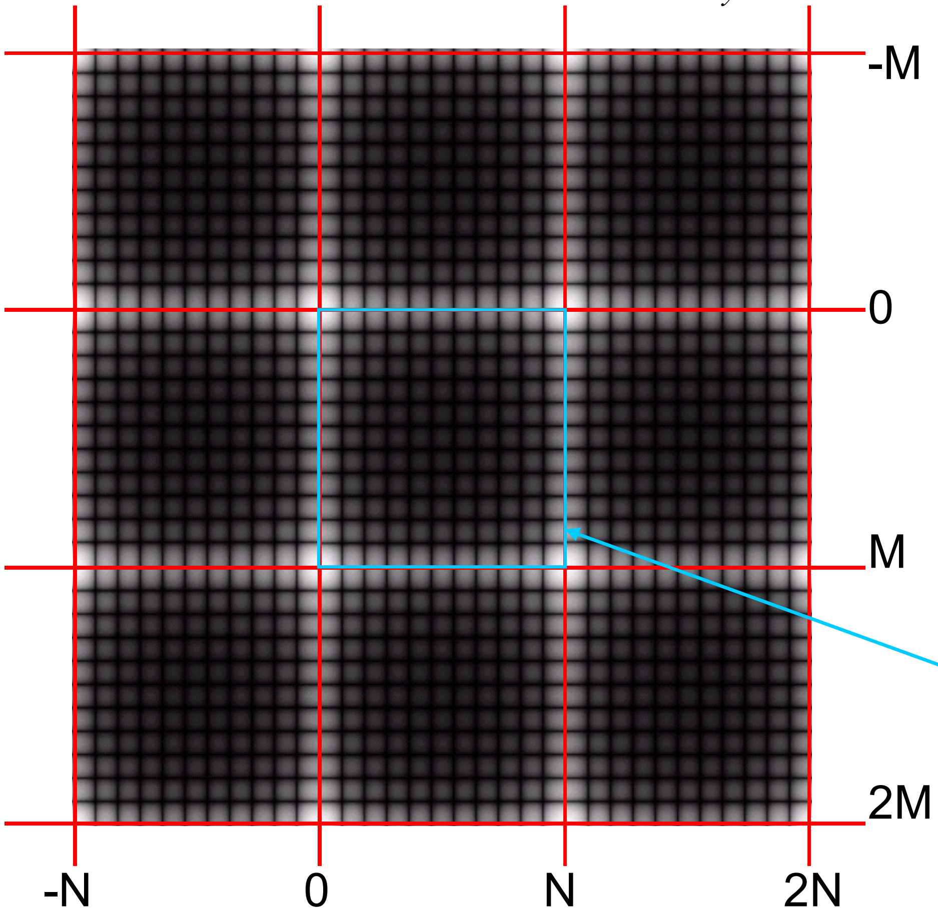
Second method



Example : 2-D DFT

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



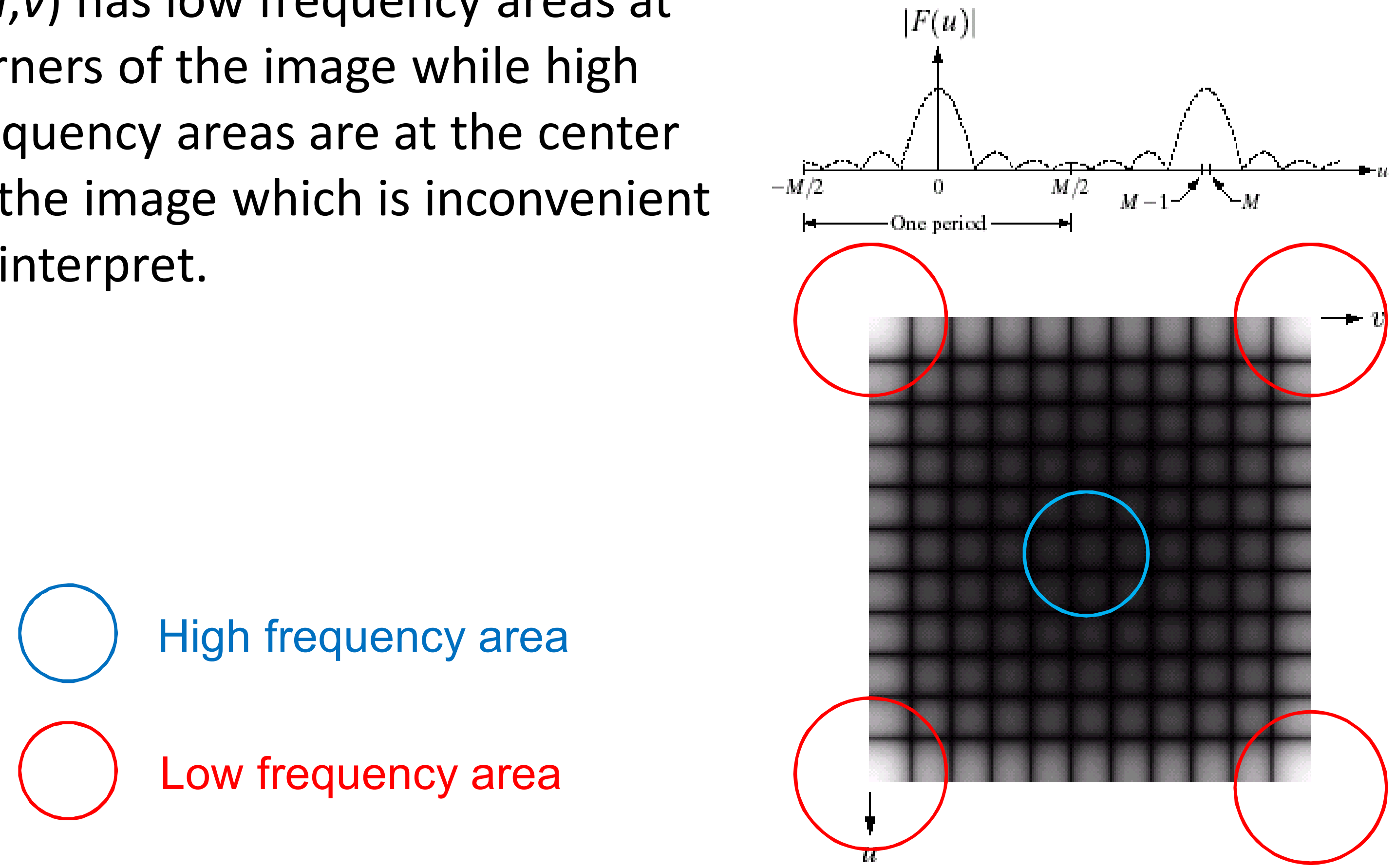


For an image of size *NxM* pixels, its 2-D DFT repeats itself every *N* points in *x*-direction and every *M* points in *y*-direction.

We display only in this range

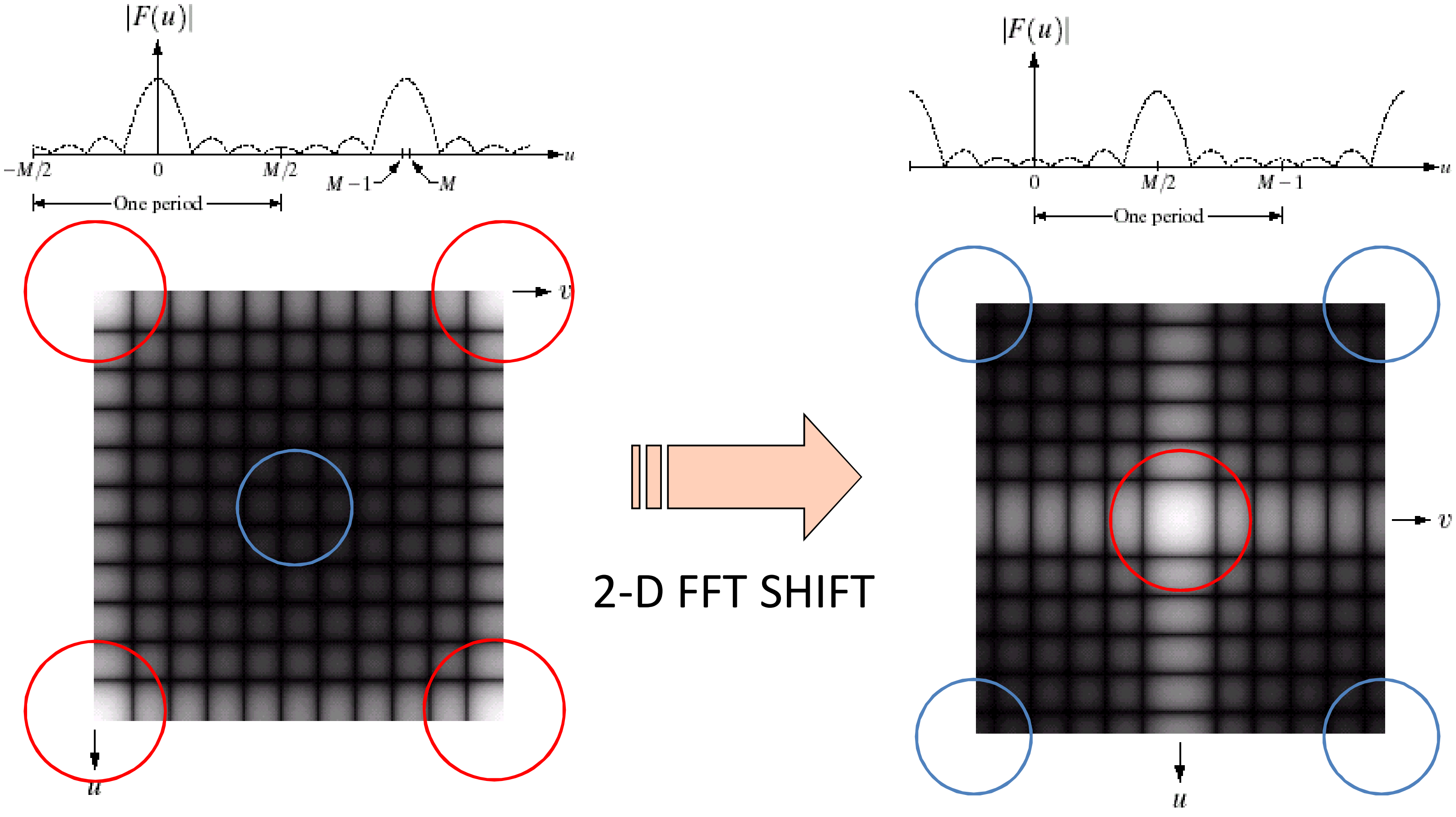
Example : Conventional Display for 2-D DFT

$F(u,v)$ has low frequency areas at corners of the image while high frequency areas are at the center of the image which is inconvenient to interpret.

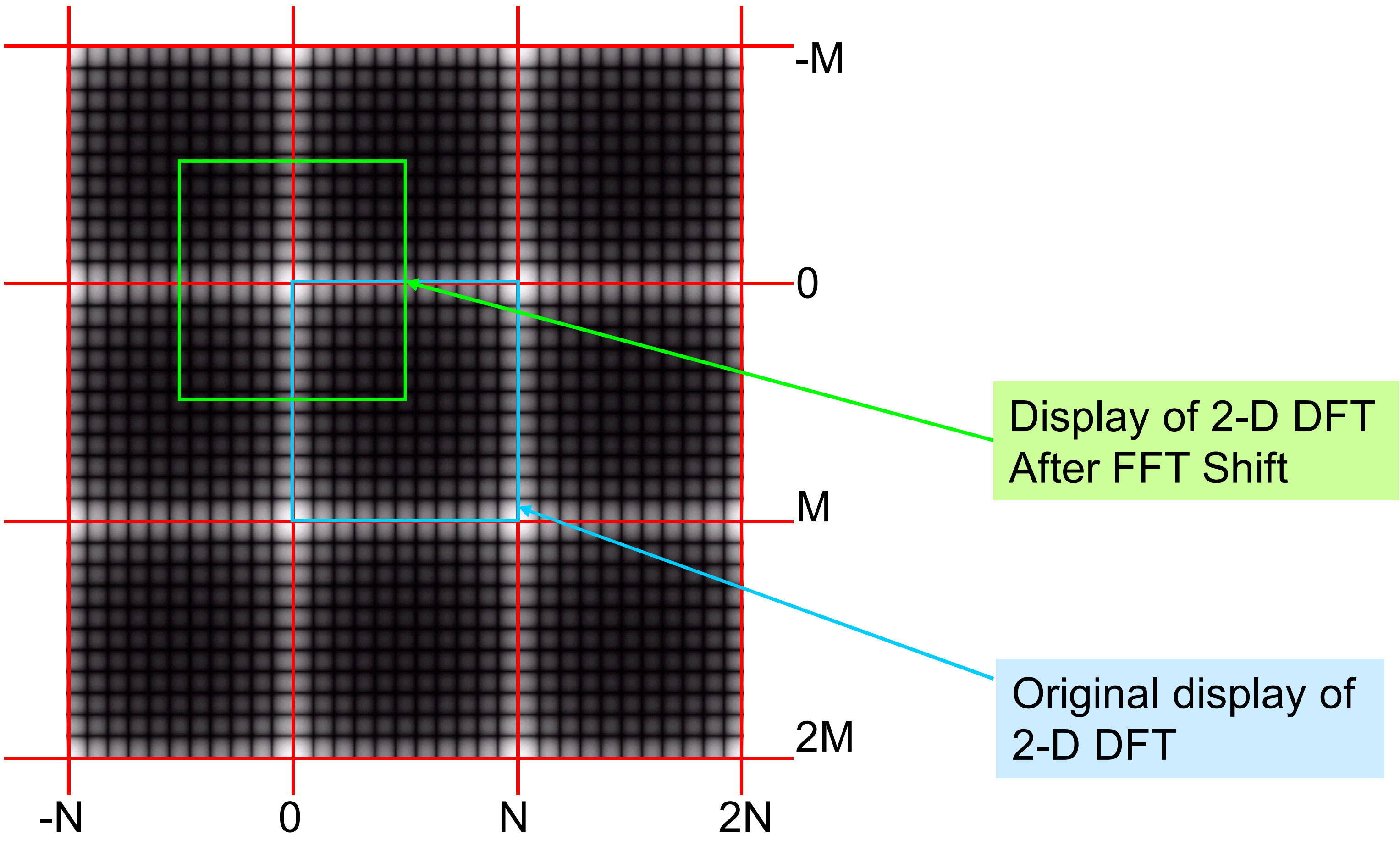


fftshift : Better Display of 2-D DFT

fftshift is a MATLAB function: Shift the zero frequency of *F*(*u*,*v*) to the center of an image.



2-D FFT Shift: How it works



Frequency Shifting

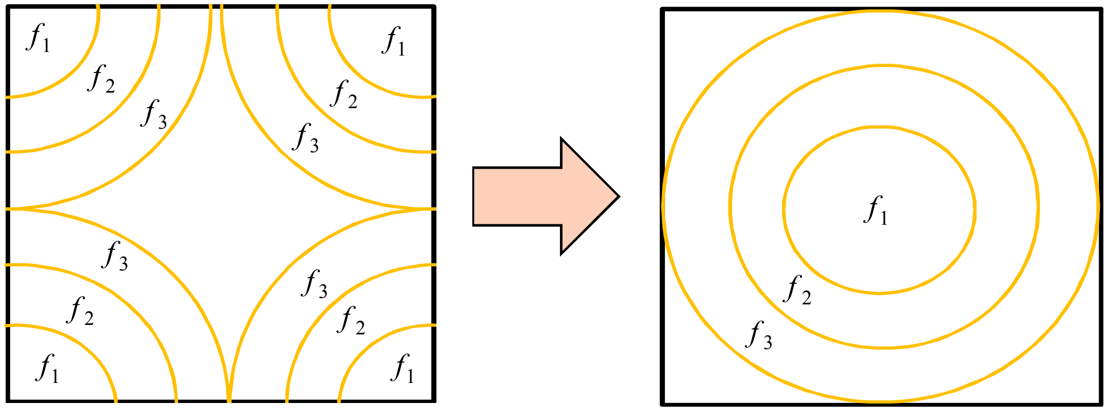
$$f(x,y)e^{-j2\pi\left(\frac{u_0 x}{M}+\frac{v_0 y}{N}\right)} \Leftrightarrow F(u-u_0, v-v_0)$$

$$f(x,y)e^{-j2\pi\left(\frac{M x}{2M}+\frac{N y}{2N}\right)} \Leftrightarrow F(u-M/2, v-N/2)$$

$$f(x,y)e^{-j\pi(x+y)} \Leftrightarrow F(u-M/2, v-N/2)$$

$$f(x,y)(-1)^{(x+y)} \Leftrightarrow F(u-M/2, v-N/2)$$

No Frequency Shifting

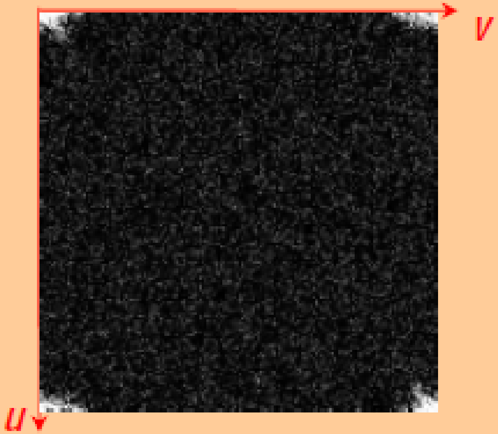


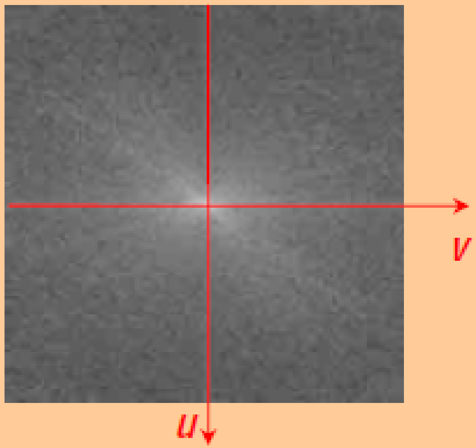
Frequency Shifting

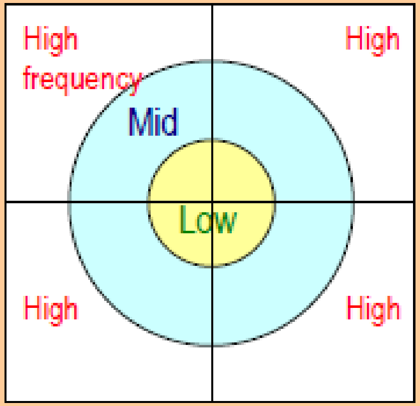
Example : Fourier Transform

MATLAB: DFT of Lena

```
im = imread('lena.bmp');
im1 = fft2(im), imshow(abs(im1), [ ]);
% Shift origin to center and scale the values (range of abs(im1): ~ [5, 8538193])
im1 = fftshift(fft2(im)), imshow(log(1+abs(im1)), [ ]);
```

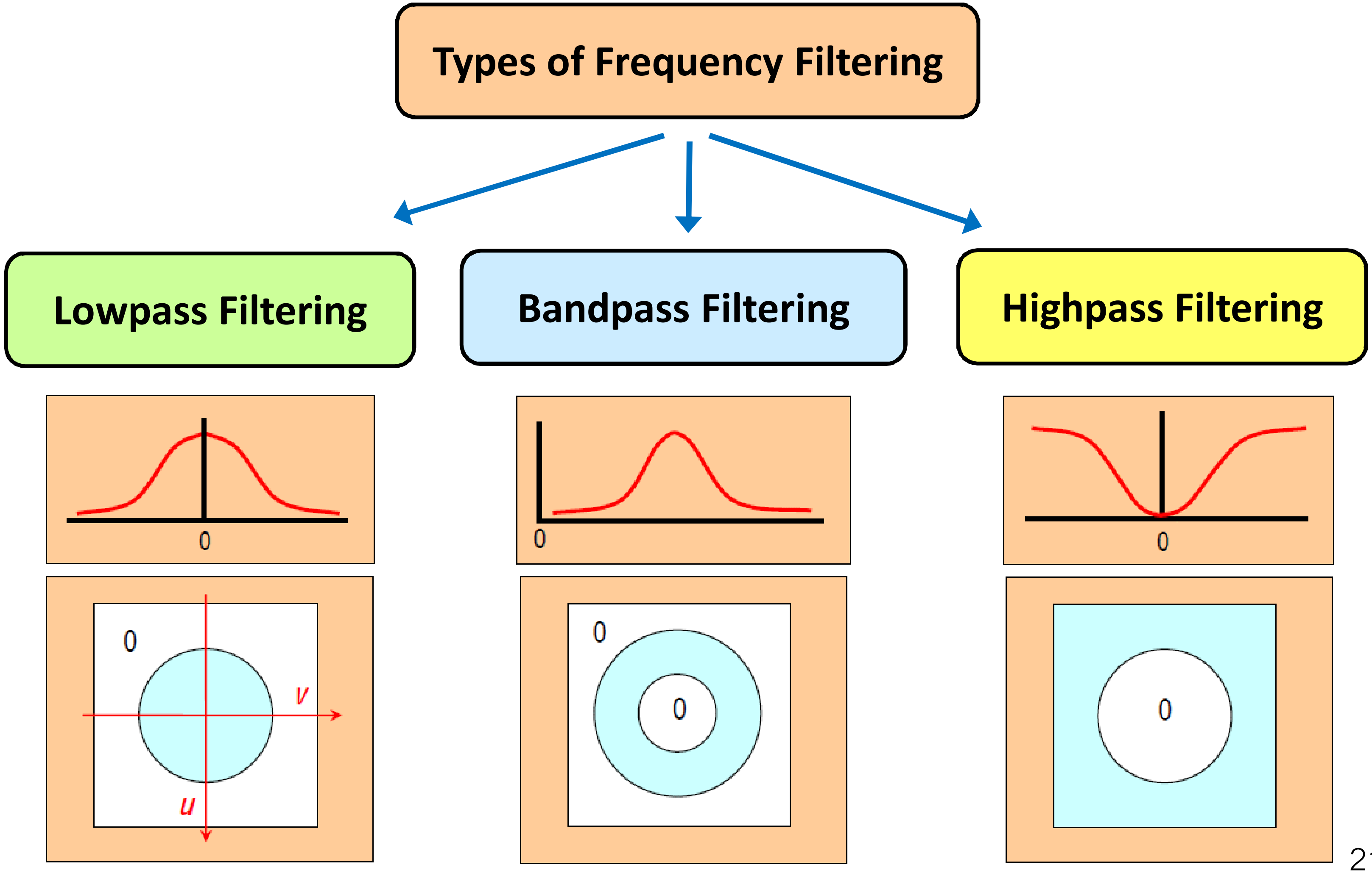




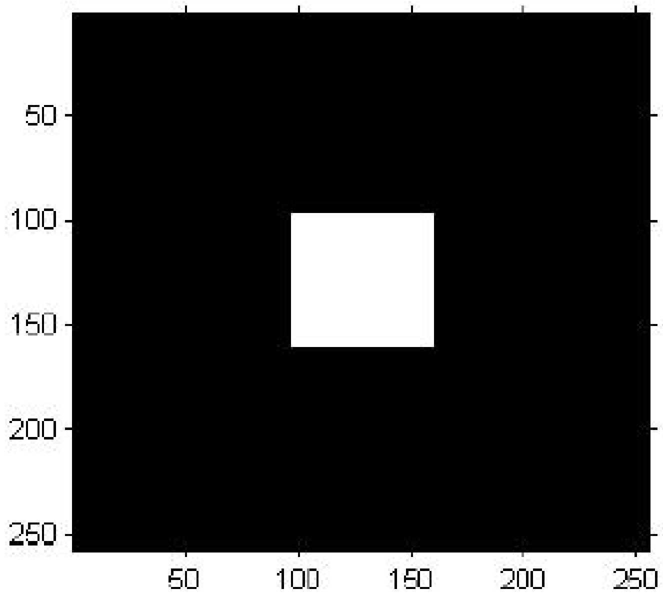


20

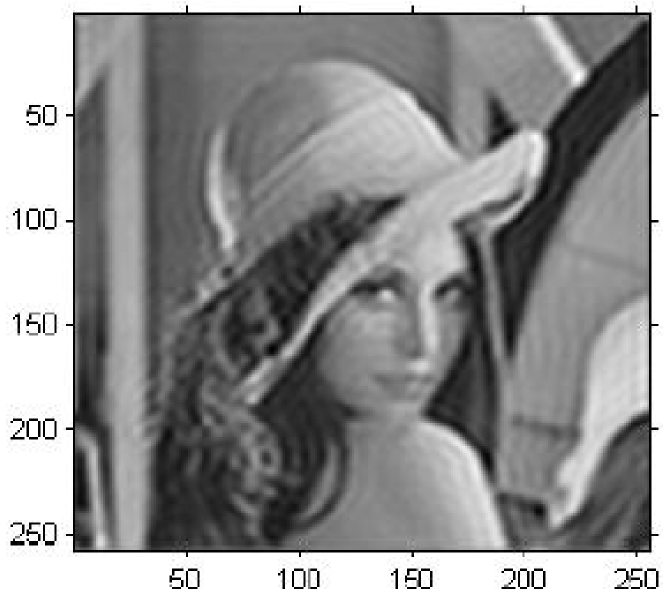
Types of Filtering in the Frequency Domain



Lowpass Filtering in the Frequency Domain

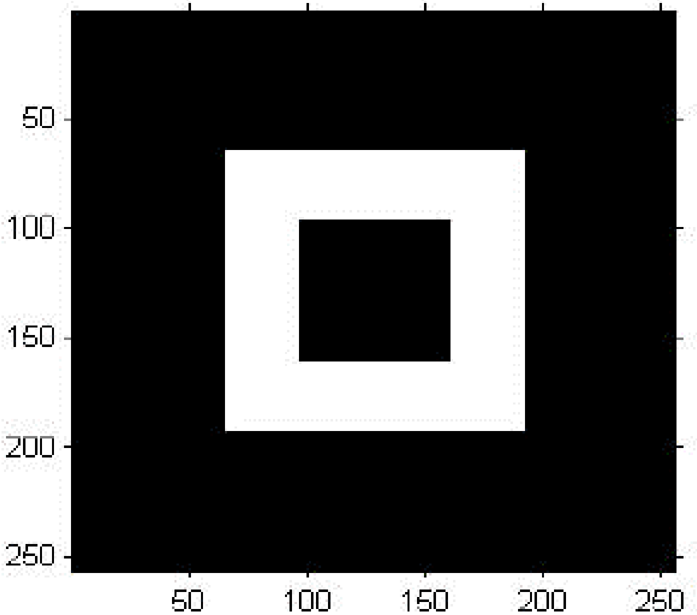


Mask for lowpass filtering

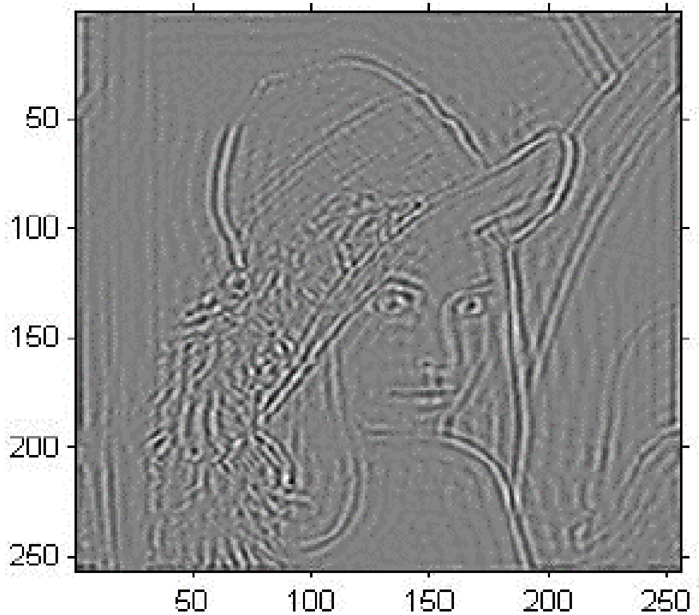


Lowpass filtered image

Bandpass Filtering in the Frequency Domain

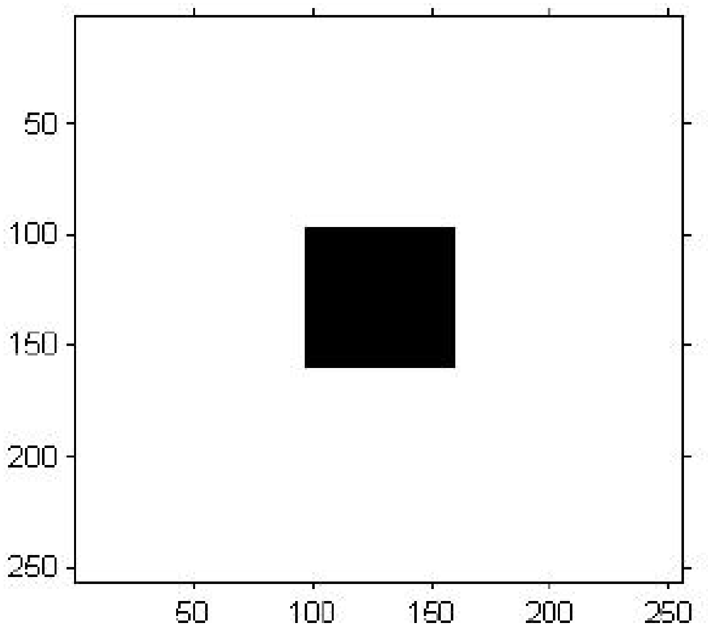


Mask for bandpass filtering

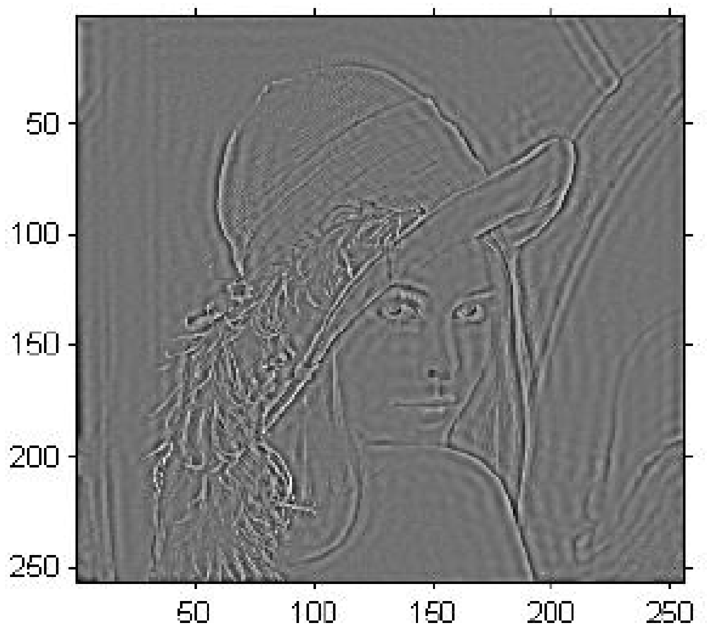


Bandpass filtered image

Highpass Filtering in the Frequency Domain



Mask for highpass filtering



Highpass filtered image

Lowpass Filter VS. Highpass Filter

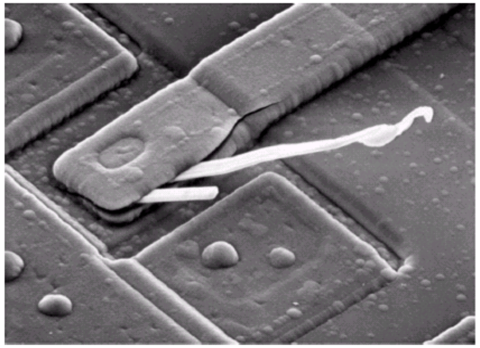
- **Lowpass filter:** A filter that attenuates high frequencies while passing the low frequencies.
  - used for image smoothing

- Low frequencies represent the gray-level appearance of an image over smooth areas.

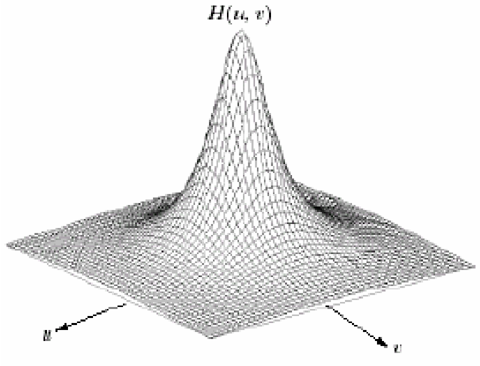
- **Highpass filter:** A filter that attenuates low frequencies while passing the high frequencies.
  - used for image sharpening

- High frequencies represents the details such as edges and noise.

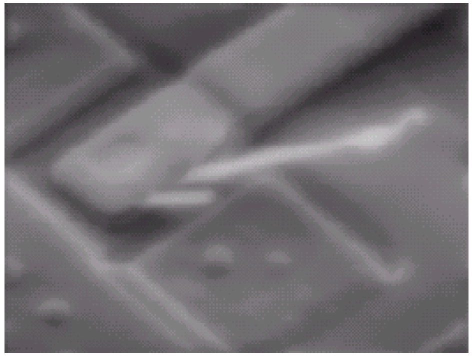
Lowpass Filter VS. Highpass Filter



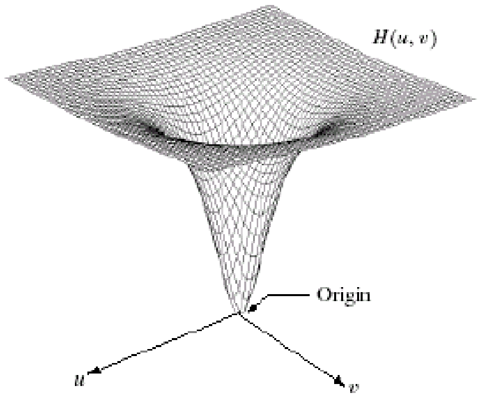
Original image



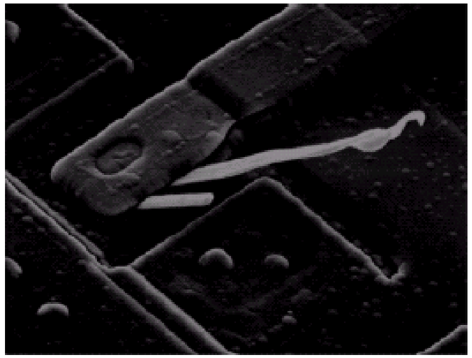
2D lowpass filter



Blurred image



2D highpass filter



Sharped image

Spatial and Frequency Domain Correspondence

- The link between filtering in the spatial and frequency domains is the convolution theorem.

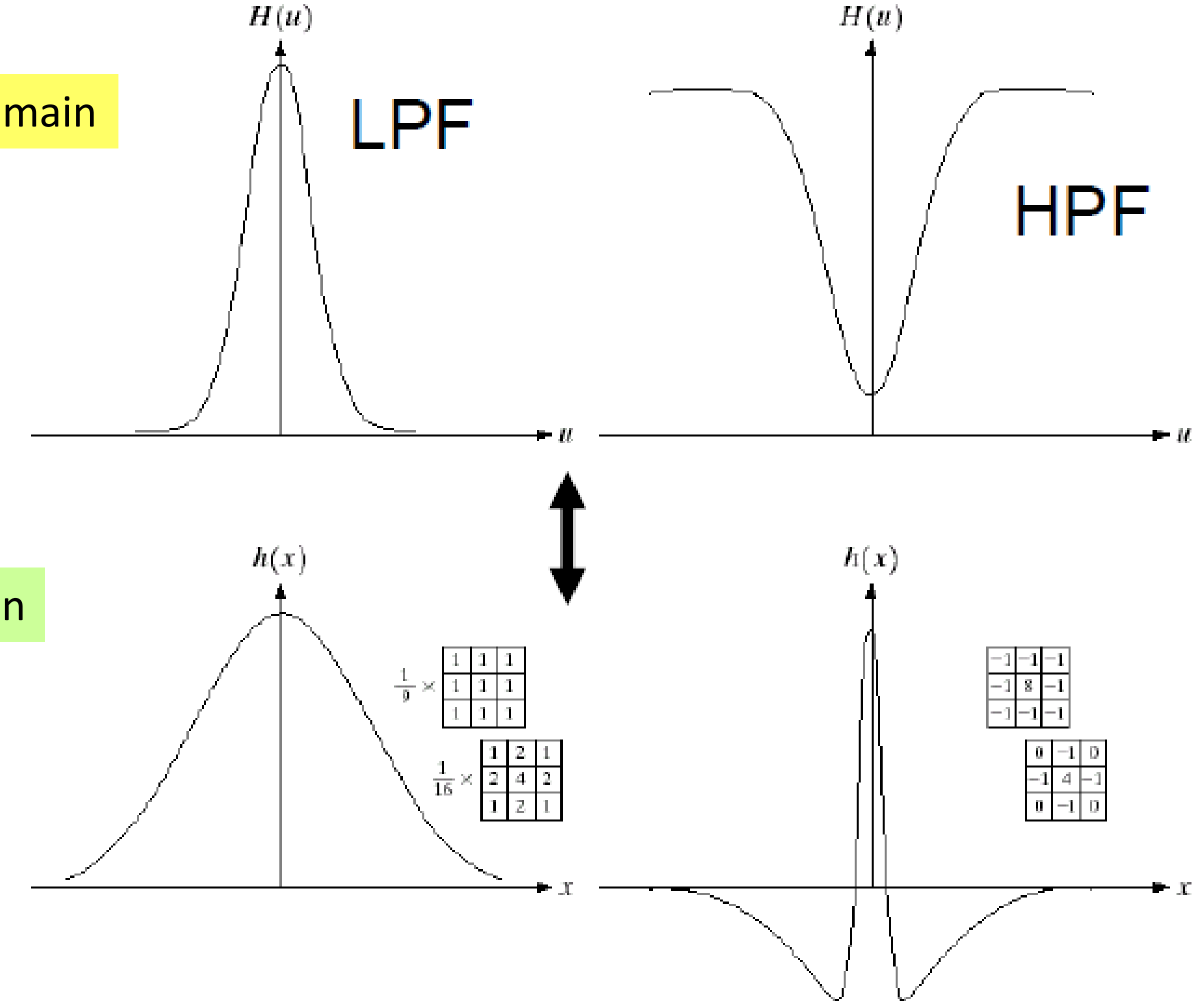
- In spatial domain, filtering is defined as the spatial convolution between an image and a filter mask.

- While in frequency domain, filtering is the multiplication of a filter function and the Fourier transform of the input image.

Spatial and Frequency Domain Correspondence

Frequency Domain

Spatial Domain



Gaussian Filtered Images in Two Domains

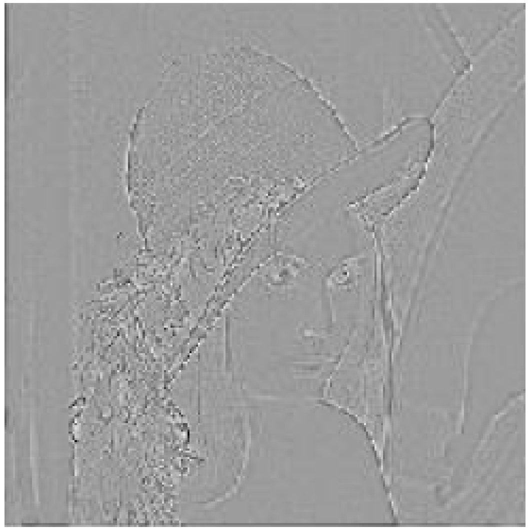
Filtered in the spatial domain

Filtered in the frequency domain

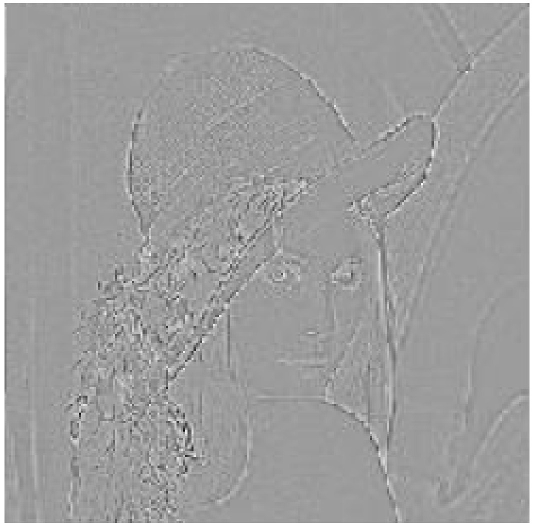
These images are identical !

Laplacian Filtered Images in Two Domains

Filtered in the spatial domain



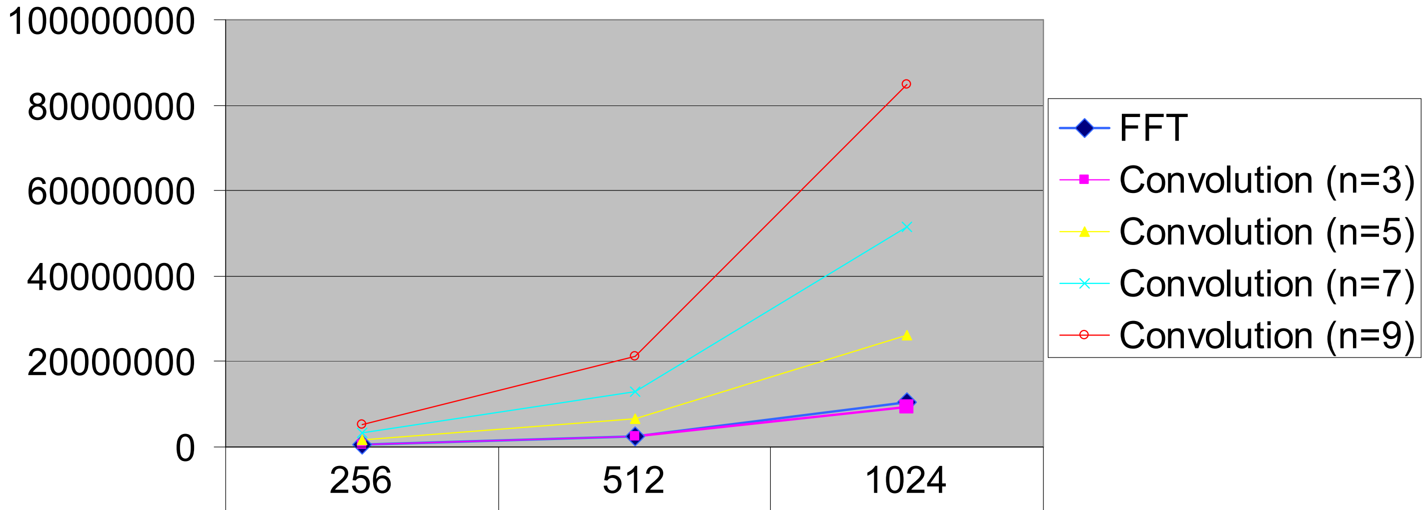
Filtered in the frequency domain



These images are again identical!

Comparison between filtering in spatial and frequency domain

- Number of Multiplication (FFT vs Convolution)



	256	512	1024
FFT	524288	2359296	10485760
Convolution (n=3)	589824	2359296	9437184
Convolution (n=5)	1638400	6553600	26214400
Convolution (n=7)	3211264	12845056	51380224
Convolution (n=9)	5308416	21233664	84934656

Comparison between filtering in spatial and frequency domain

- Filtering in the spatial domain
  - We often specify small spatial mask that attempt to capture the essence of the full filter function

- Filtering in the frequency domain
  - Enhancement task would become trivial to formulate
  - More computational efficiency for a large window size or input image size

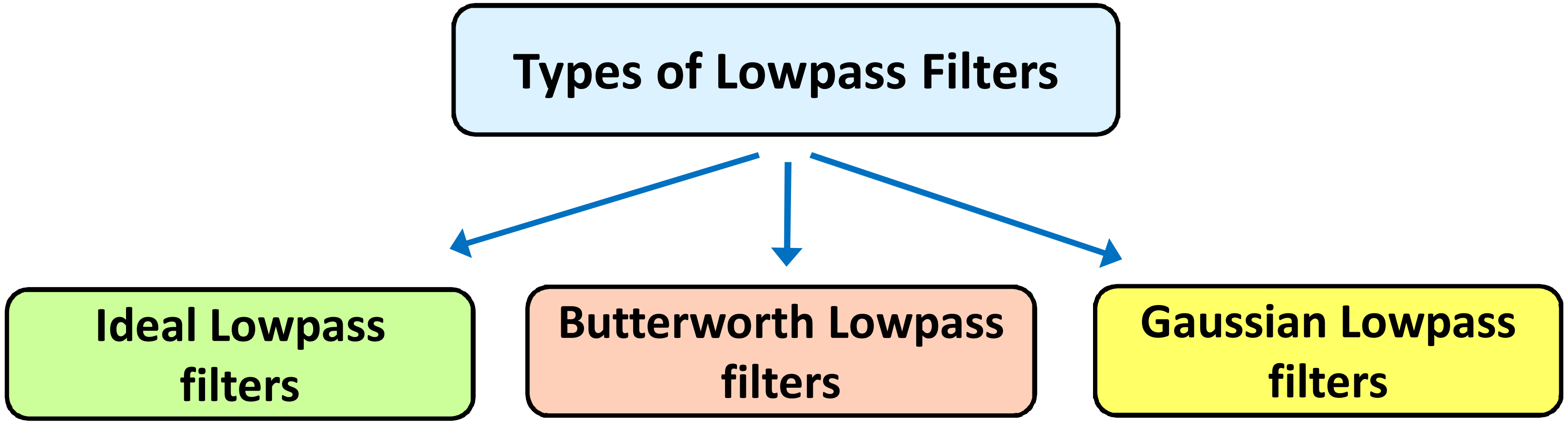
Smoothing Frequency-Domain Filters - Lowpass Filters

- Smoothing or Blurring is achieved in the frequency domain by dropping out the high frequency components.
- The high-frequency components are : edges and sharp transitions such as noise.
- Given the Fourier transformed image F(u) , the filtered image G(u) can be obtained by :

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

- Where :
  - F(u,v) is the Fourier transform of the image being filtered.
  - H(u,v) is the filter transform function.

Types of Lowpass Filters



Smoothing Frequency-Domain Filters - Lowpass Filters

- Smoothing can be achieved by lowpass filters. We will consider only 3 types of lowpass filters :
  - Ideal Lowpass filters (ILPF)
  - Butterworth Lowpass filters (BLPF)
  - Gaussian Lowpass filters (GLPF)
- These three filters cover the range from very sharp (ideal) to very smooth (Gaussian) filter functions.

Ideal Lowpass filters (ILPF)

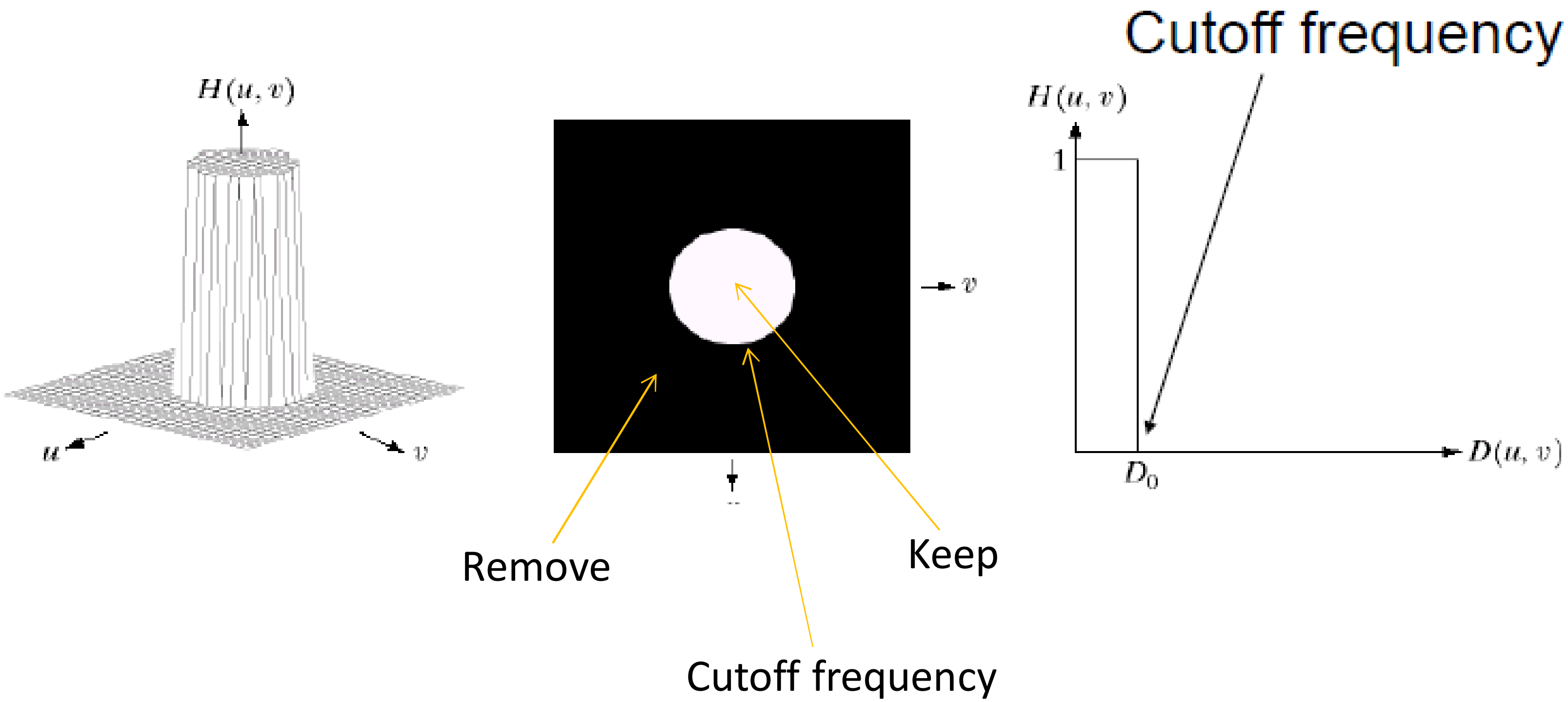
- An ideal lowpass filter keeps low frequencies in an image and discards the high frequencies.
- Simply cuts off all the high frequencies higher than the specified cutoff frequency.
- The filter transfer function :

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

- Where :
  - D(u,v) is the distance from the origin
  - D_0 is the cutoff frequency
- The center of the frequency rectangle is at (u,v) = (M/2,N/2)

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

Visualization of Ideal Lowpass filters



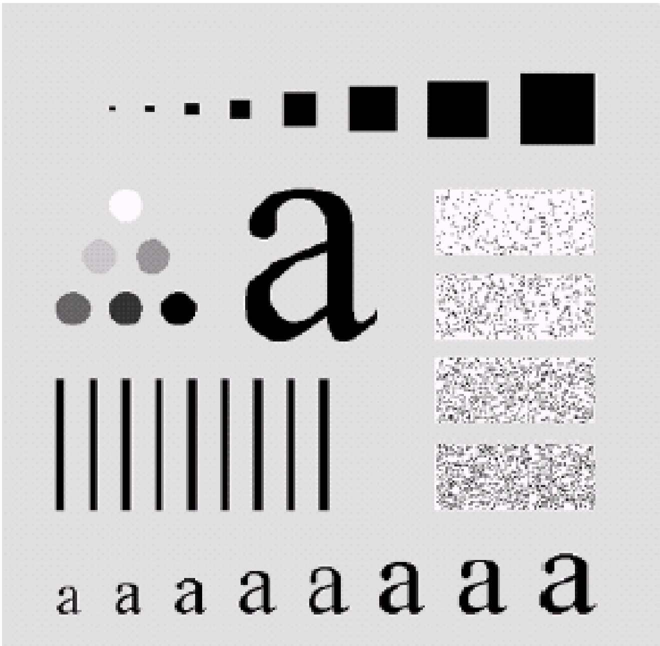
- a) Perspective plot of an ideal lowpass filter transfer function.
- b) Filter displayed as an image.
- c) Filter radial cross section.

Effects for Ideal Lowpass filters

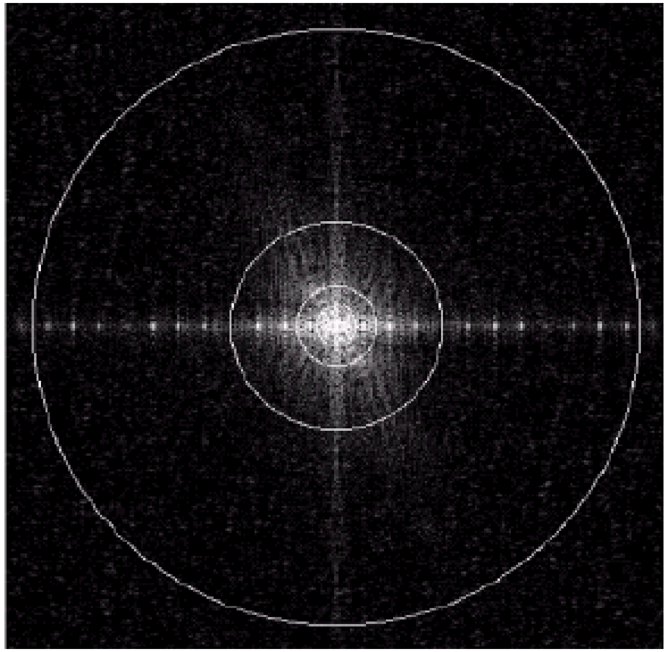
- Ideal lowpass filtering is not very practical but they can be implemented on a computer to study their behavior.
- Ideal Lowpass filters causes severe ringing effects.

Example of Ideal Lowpass filters

Fourier Spectrum with circles



Original Image

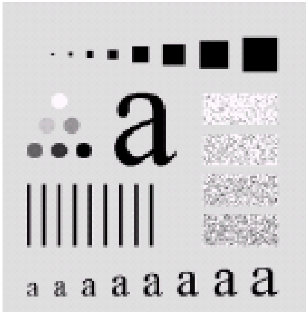


Fourier Spectrum

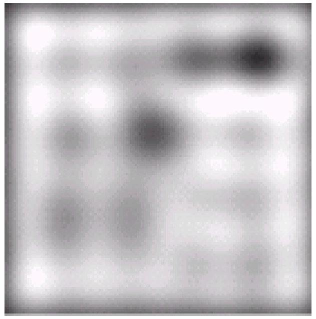
The smaller D_0, the more high frequency components are removed

Results of Ideal Lowpass filters

Original image

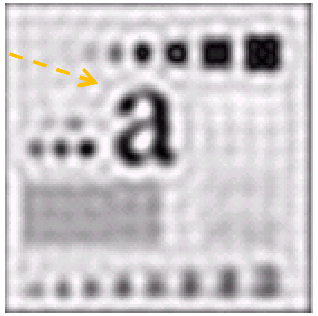


Result of filtering with ideal low pass filter of radius 5



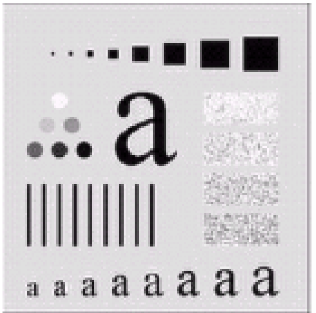
Ringing is a characteristics of ideal Filter

Result of filtering with ideal low pass filter of radius 15

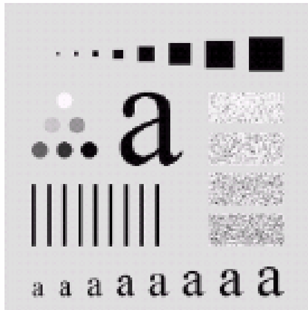


Result of filtering with ideal low pass filter of radius 30

Result of filtering with ideal low pass filter of radius 80



Result of filtering with ideal low pass filter of radius 230



Ringing effect can be obviously seen!

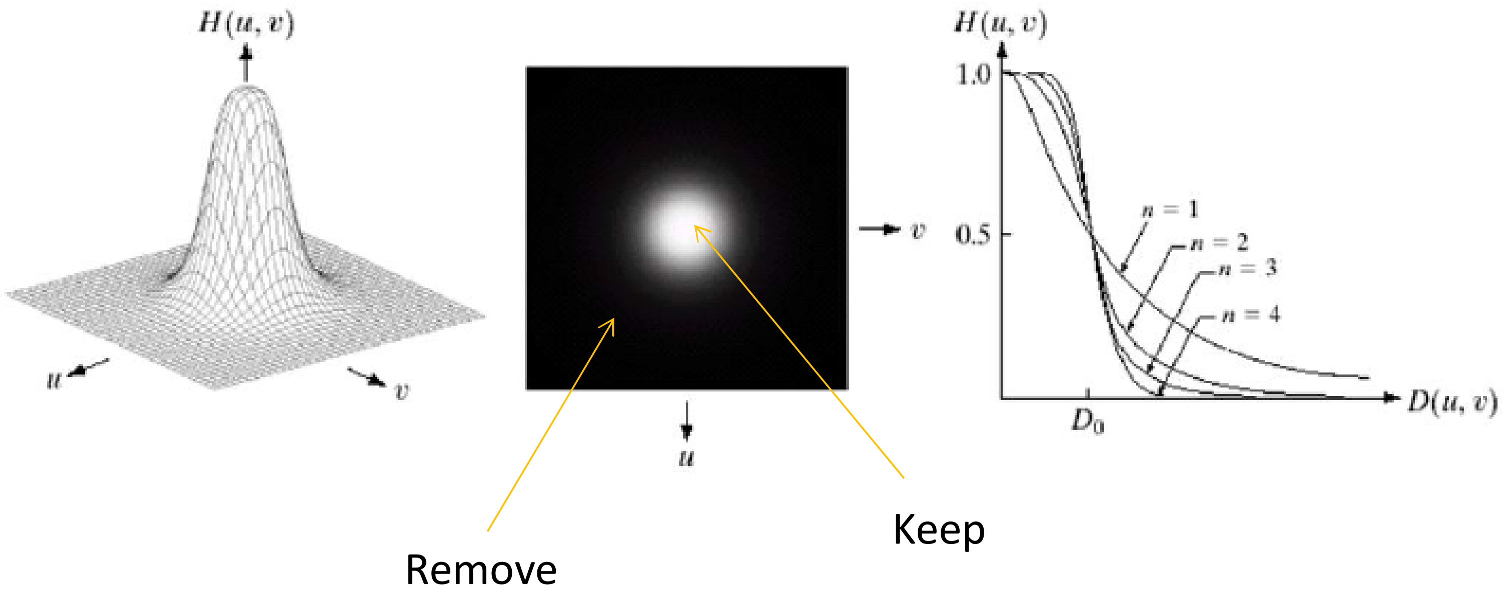
Butterworth Lowpass filters (BLPF)

- A Butterworth lowpass keeps frequencies inside radius D_0 and discards values outside.
- The transfer function of BLPF of order n and with a specified cutoff frequency is denoted by the following filter transfer function :

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

- Where :
  - D(u,v) is the distance from the origin
  - D_0 is the cutoff frequency
  - n is the order of the filter

Visualization of Butterworth Lowpass filters

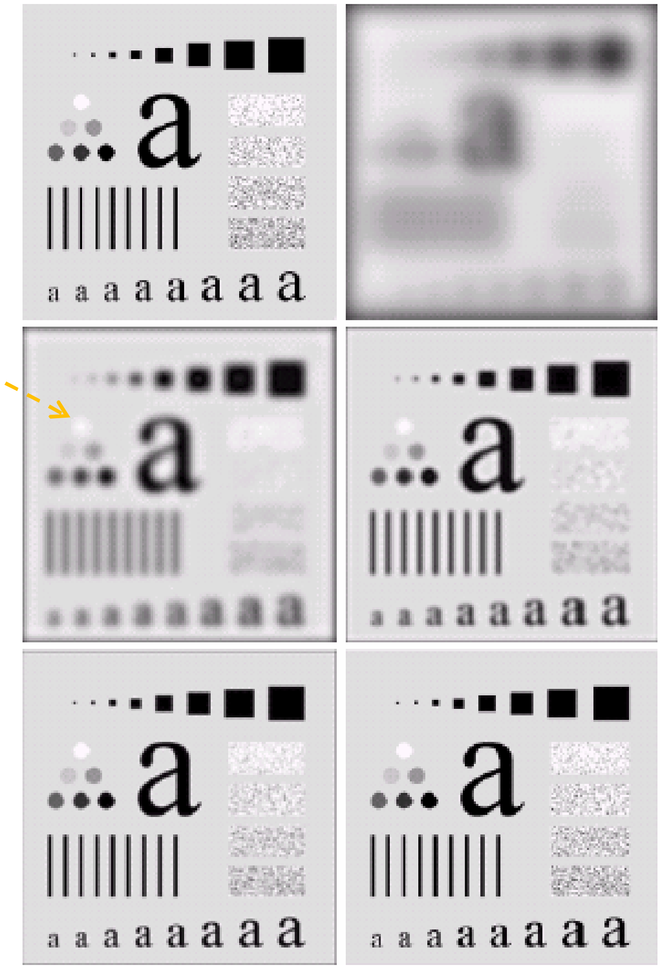


- a) Perspective plot of a Butterworth Lowpass filters transfer function
- b) Filter displayed as an image
- c) Filter radial cross section.

Result of Butterworth Lowpass filters

Original image

Ringing is not visible in any of these images.



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

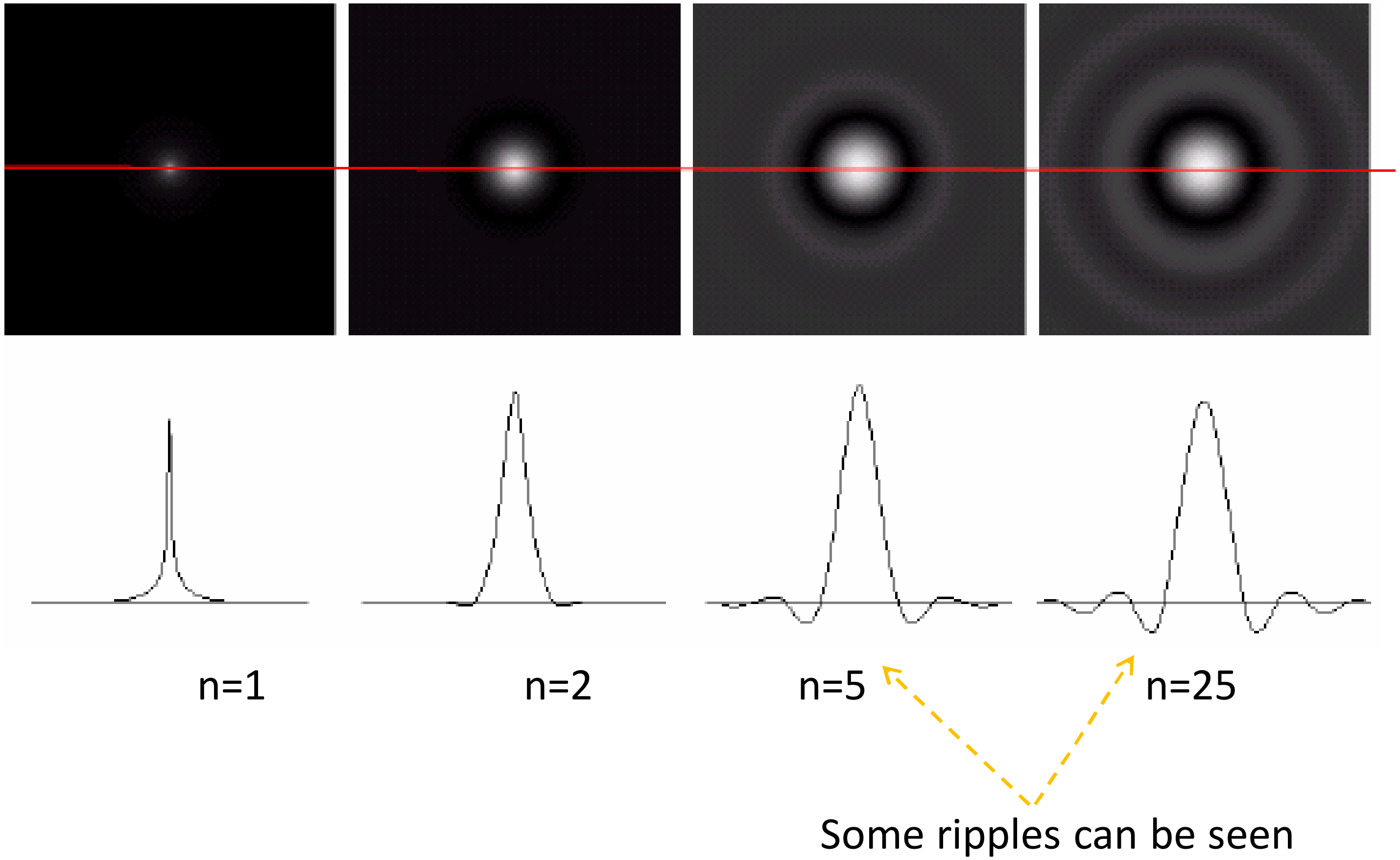
Result of filtering with Butterworth filter of order 2 and cutoff radius 30

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

There is less ringing effect compared to those of ideal lowpass filters!

Result of Butterworth Lowpass filters



Ringing artifact increases as n increases

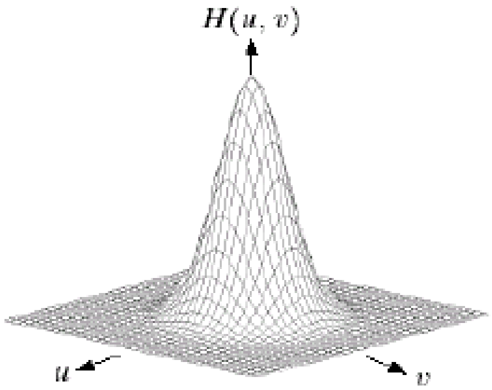
Gaussian Lowpass filters (GLPF)

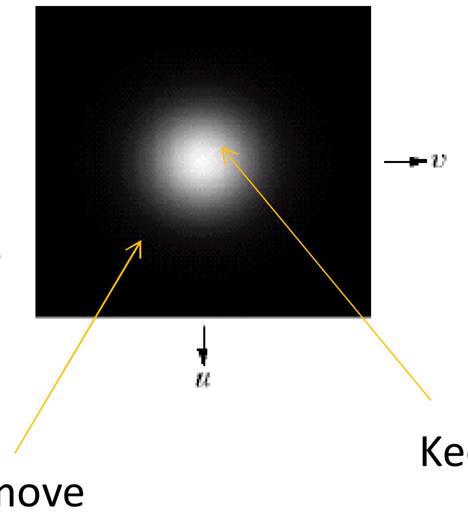
- A Gaussian lowpass filter has the most natural and well behaved filter shape.
- The transfer function of GLPF is given as follows :

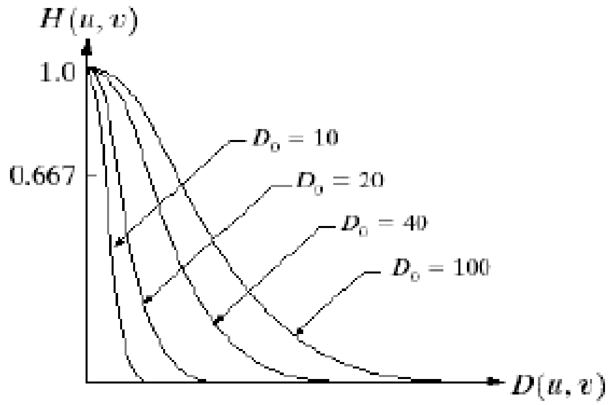
$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

- Where :
  - D(u,v) is the distance from the origin
  - D_0 is the cutoff frequency

Visualization of Gaussian Lowpass filters







- a) Perspective plot of a Gaussian Lowpass filters transfer function
- b) Filter displayed as an image
- c) Filter radial cross sections for various of D_0

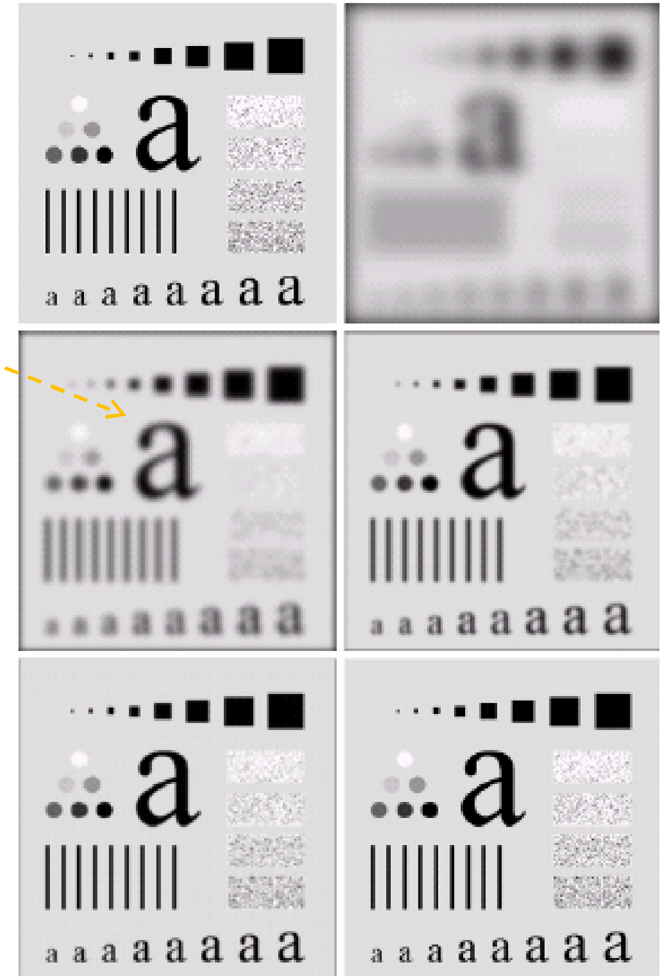
Effects for Gaussian Lowpass filters

- The inverse Fourier transform of the Gaussian Lowpass filter is also Gaussian in the Spatial domain.
- Therefore there is no ringing effect of the GLPF. Ringing artifacts are not acceptable in fields like medical imaging. Hence use Gaussian instead of the ILPF/BLPF.
- Transfer function is smooth, like Butterworth filter.
- Advantage : No ringing artifacts.

Result of Gaussian Lowpass filters

Original image

No Ringing.



Result of filtering with Gaussian filter with cutoff radius 5

Result of filtering with Gaussian filter with cutoff radius 15

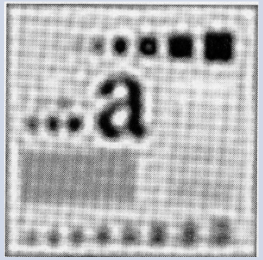
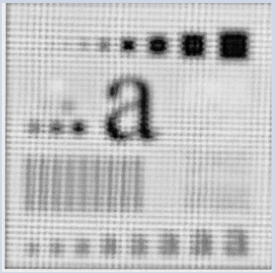
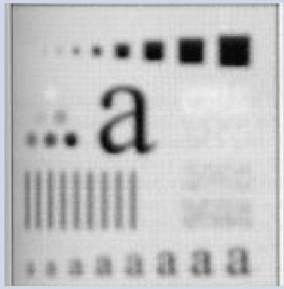
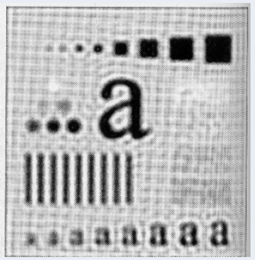
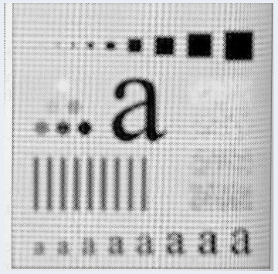
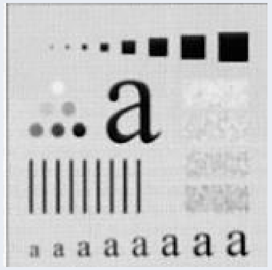
Result of filtering with Gaussian filter with cutoff radius 30

Result of filtering with Gaussian filter with cutoff radius 85

Result of filtering with Gaussian filter with cutoff radius 230

No ringing effect!

Lowpass Filters Compared

Ideal LPF	Butterworth LPF (N=2)	Gaussian LPF
 $D_0 = 15$	 $D_0 = 15$	 $D_0 = 15$
 $D_0 = 30$	 $D_0 = 30$	 $D_0 = 30$

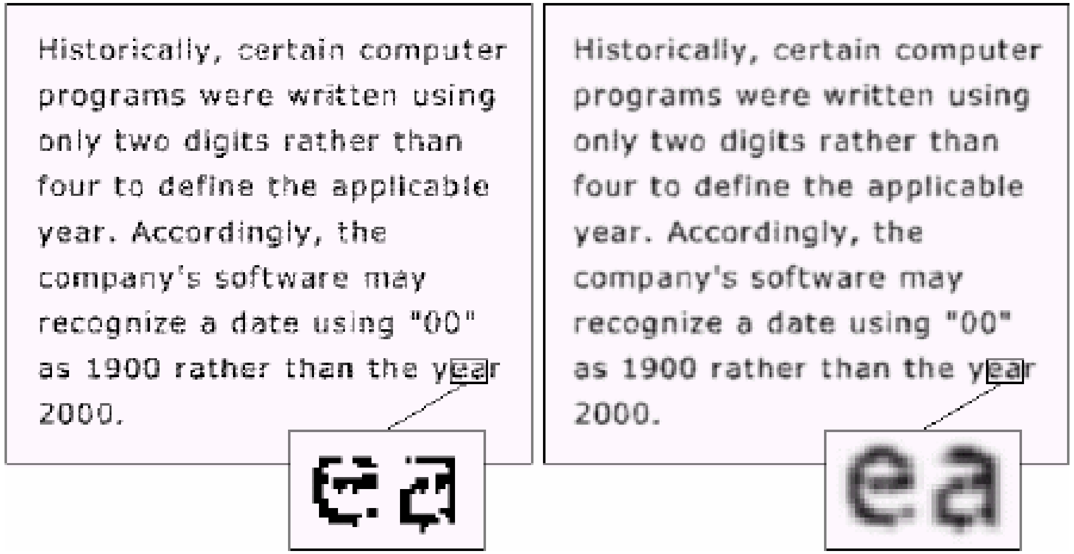
- These three filters cover the range from very sharp (ideal) to very smooth (Gaussian) filter functions.

Additional Examples of Lowpass Filtering

- Machine recognition systems (OCR)
  - Bridge small gaps in alphabets of text
- Printing/publishing/advertising
  - "Photoshopping" to remove blemishes/lines and obtain a smoother, softer result
- Aerial/Satellite imagery
  - Removing unwanted scan lines in images

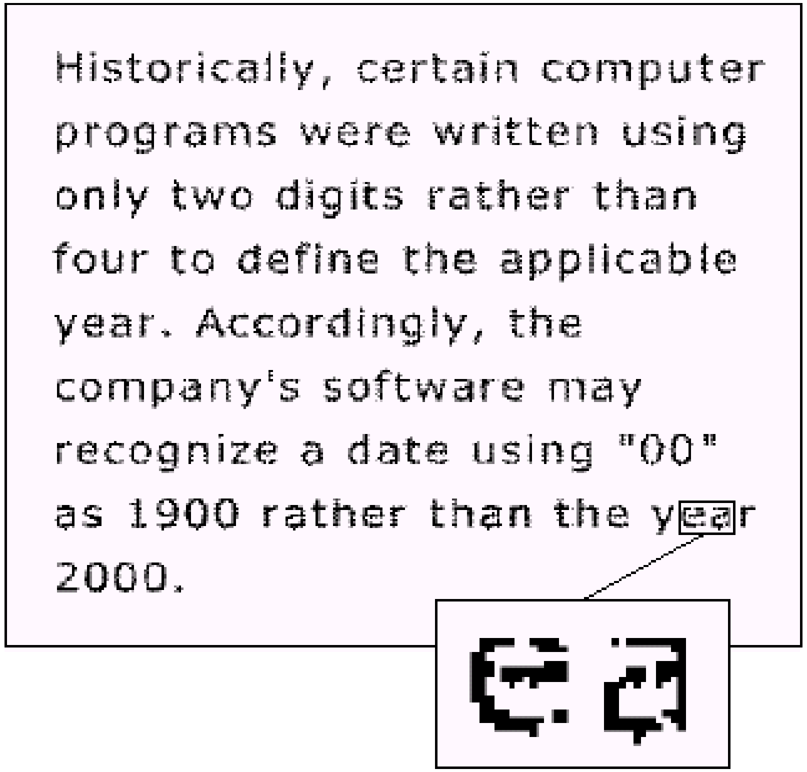
Example : Lowpass Filtering

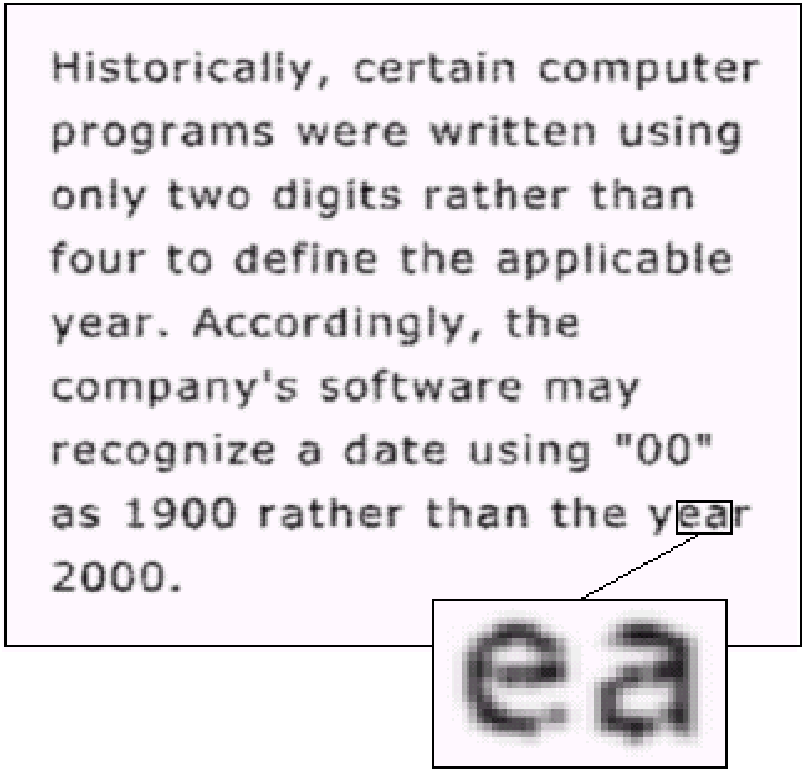
Character recognition



- a) Sample text of poor resolution (note broken characters in magnified view)
- b) Result of filtering with a GLPF (broken character segment were joined)

Example : Lowpass Filtering





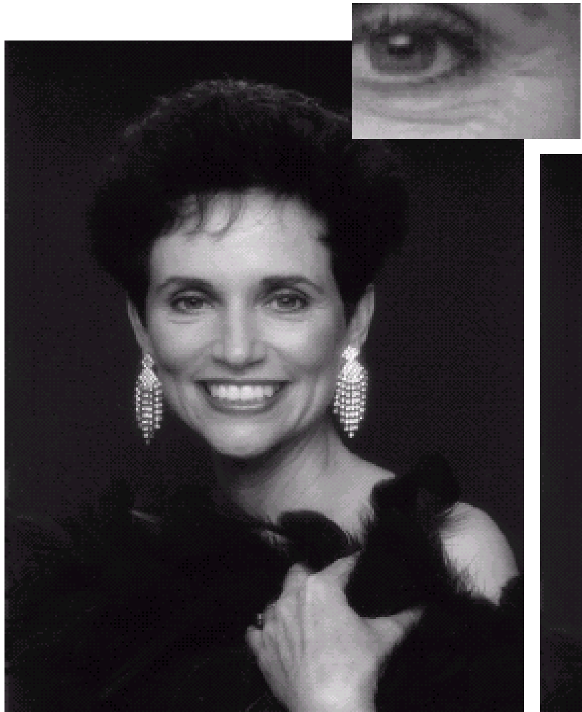
Example : Lowpass Filtering

Picture Studio Decoration

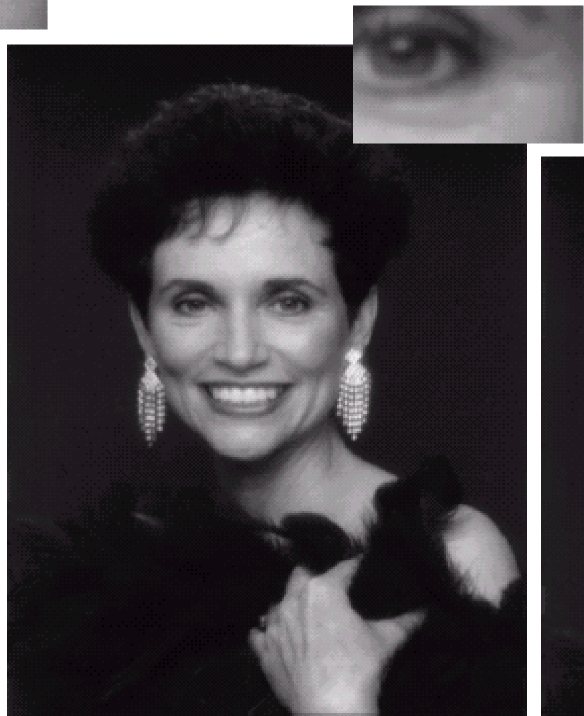
- a) Original image (1028 x 732 pixels)
- b) Result of filtering with a GLPF with D_0 = 100
- c) Result of filtering with a GLPF with D_0 = 80

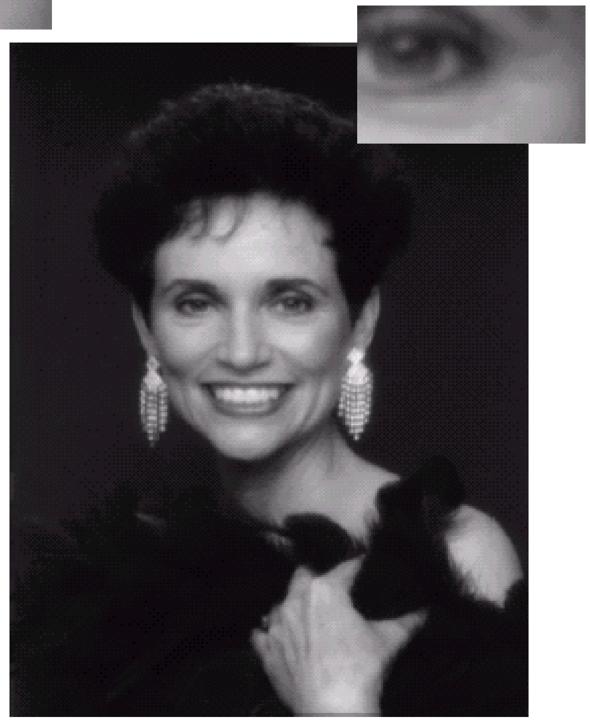
Note : reduction in skin fine lines in the magnified sections of b) and c)

Example : Lowpass Filtering



Remove blemishes





Softer-Looking

Example : Lowpass Filtering

Prominent scan lines

- a) Image showing prominent scan lines
- b) Result of using a GLPF with D_0 = 30
- c) Result of using a GLPF D_0 = 10

Example : Lowpass Filtering

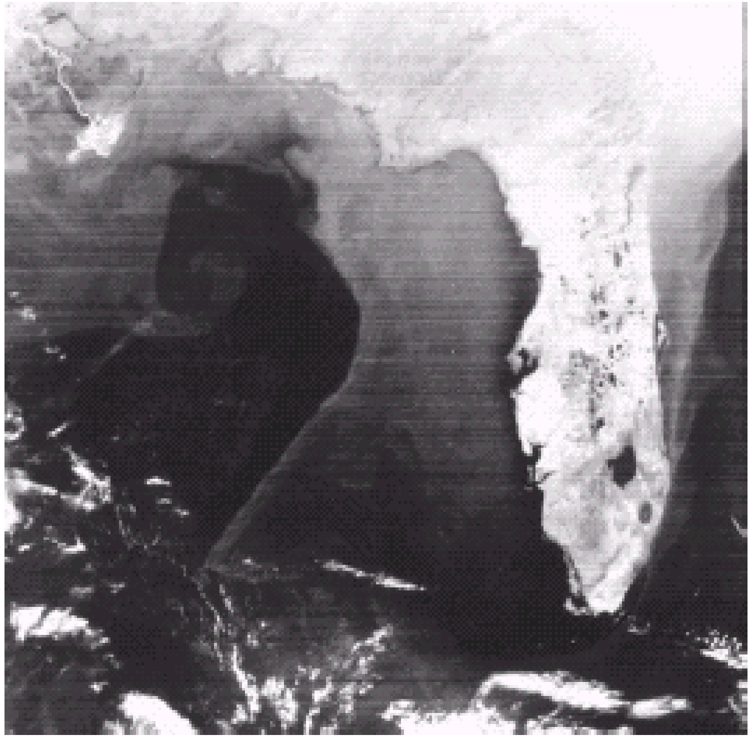


Image showing prominent scan lines.

Result of using a GLPF D_0 = 10

Sharpening Frequency Domain Filters - Highpass Filters

- Edges and other abrupt changes in gray levels are associated with high frequency components, image sharpening can be achieved in the frequency domain by a highpass filtering process, which attenuates the low frequency component's without disturbing high frequency information in the Fourier transform.
- High pass frequencies are precisely the reverse of low pass filters.

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

- where :
  - H_{lp}(u,v) is the transfer function of the corresponding lowpass filter.

Sharpening Frequency Domain Filters - Highpass Filters

- Sharpening can be achieved by highpass filters. We will consider only 3 types of sharpening highpass filters :
  - Ideal Highpass Filters (IHPF)
  - Butterworth Highpass Filters (BHPF)
  - Gaussian Highpass Filters (GHPF)

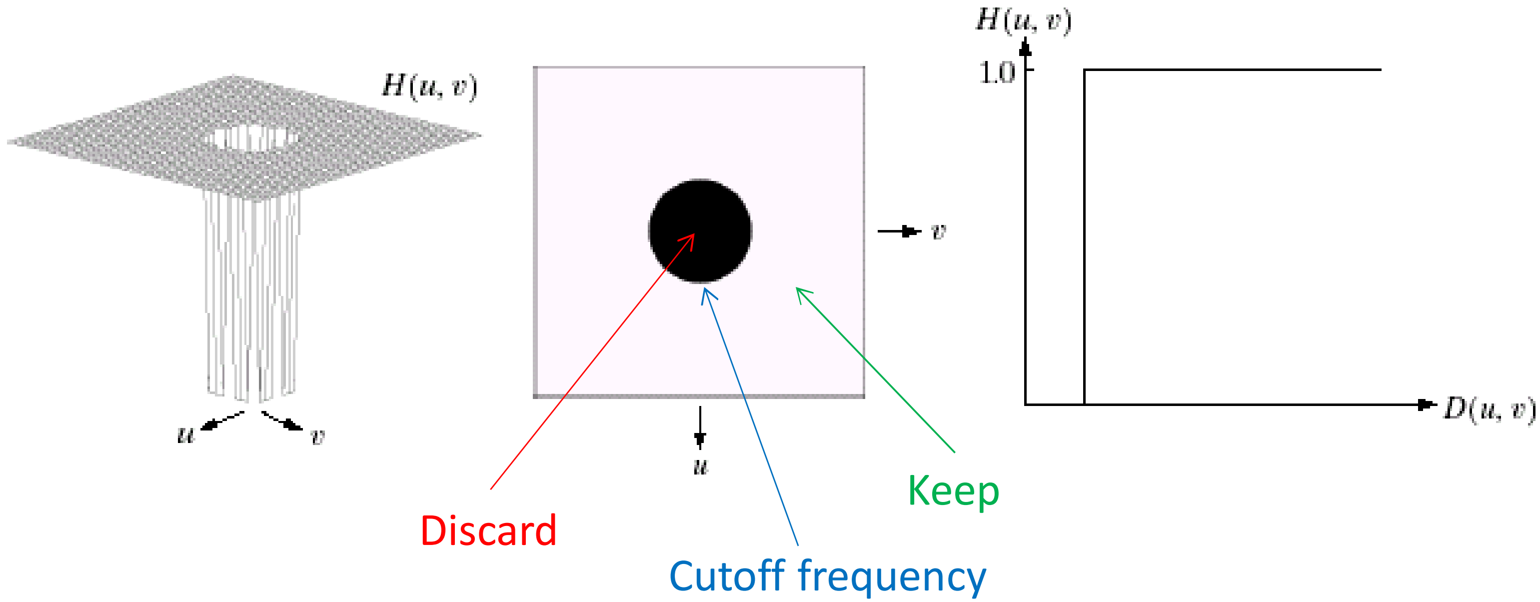
Ideal Highpass Filters (IHPF)

- An ideal highpass filter keeps the high frequencies and discards the low frequencies.
- Simply cuts off all the low frequencies lower than the specified cutoff frequency.
- The filter transfer function :

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

- where :
  - D(u,v) is the distance from the origin
  - D_0 is the cutoff frequency

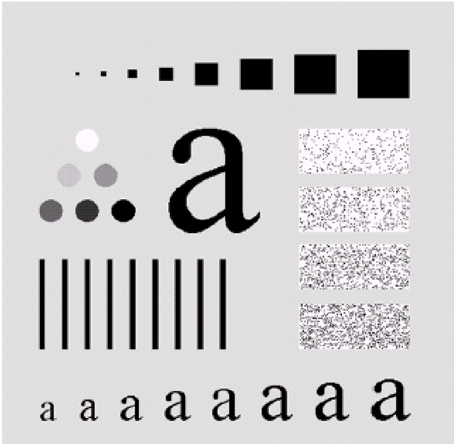
Visualization of Ideal Highpass Filters

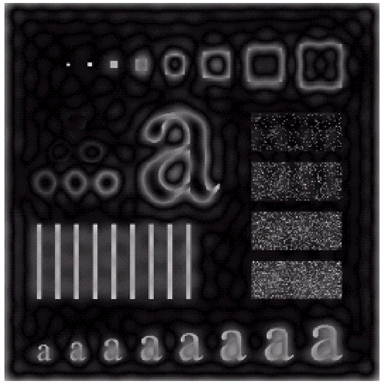


- a) Perspective plot of IHPF
- b) Image representation of IHPF
- c) Cross section of IHPF

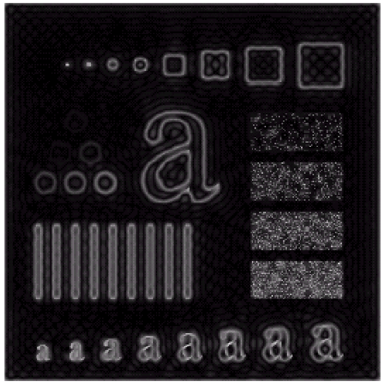
Result of Ideal Highpass Filters

Original image

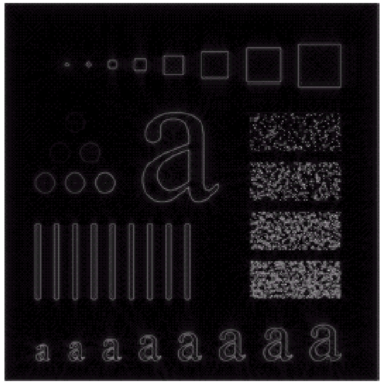




Results of ideal high pass filtering with $D_0 = 15$



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



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

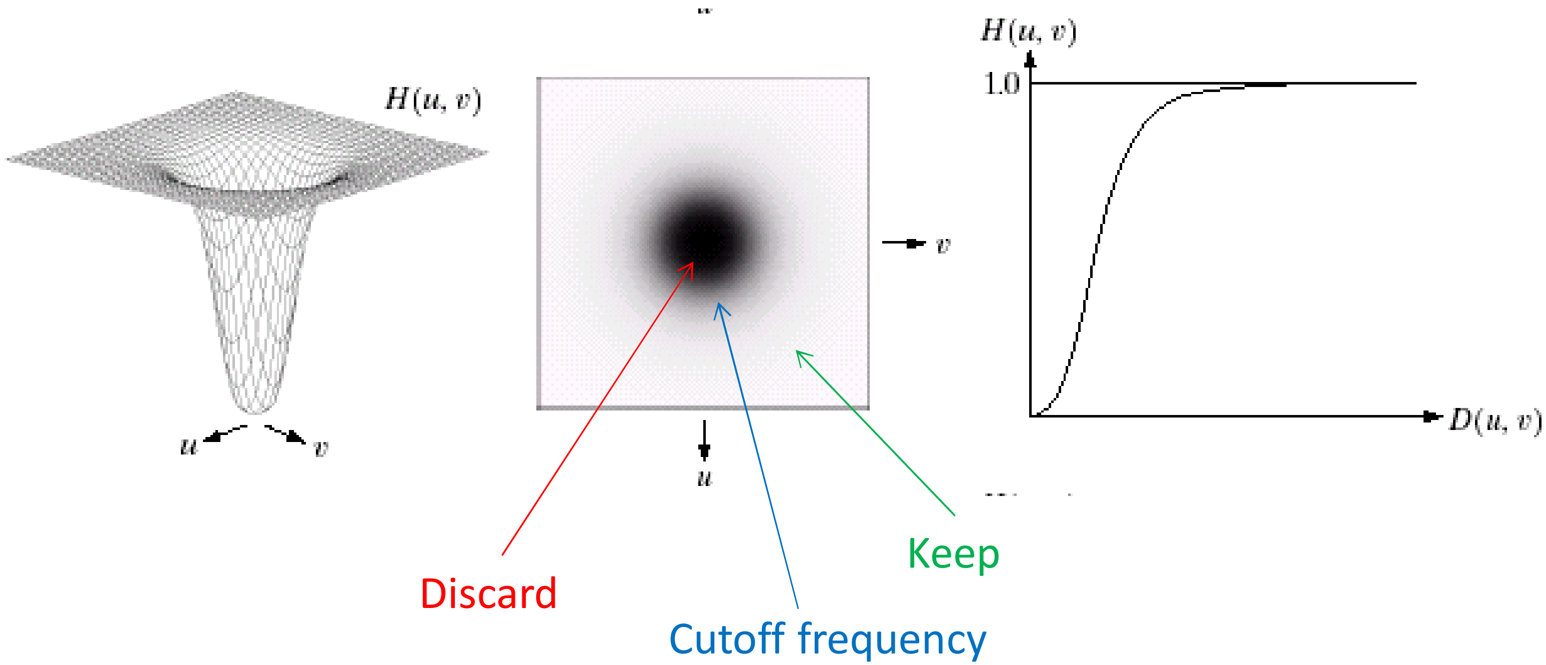
Butterworth Highpass Filters (BHPF)

- A Butterworth highpass keeps frequencies outside radius D_0 and discards values inside.
- The transfer function of BHPF of order n and with a specified cutoff frequency is given by :

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

- where :
- D(u,v) is the distance from the origin
  - D_0 is the cutoff frequency
  - n is the order of the filter

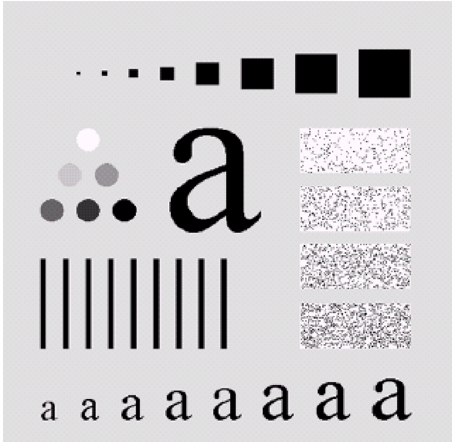
Visualization of Butterworth Highpass Filters



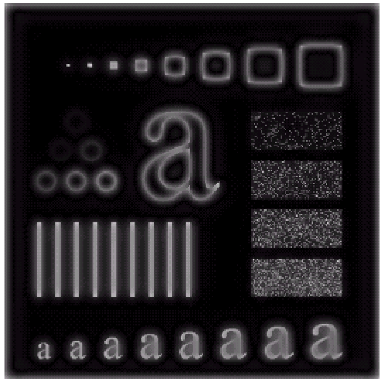
- a) Perspective plot of BHPF
- b) Image representation of BHPF
- c) Cross section of BHPF

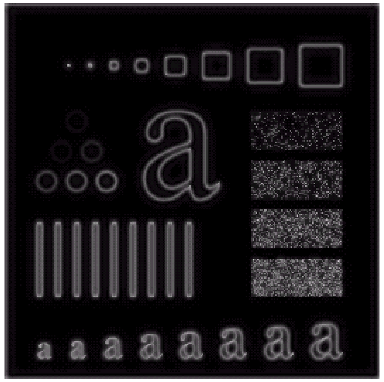
Result of Butterworth Highpass Filters

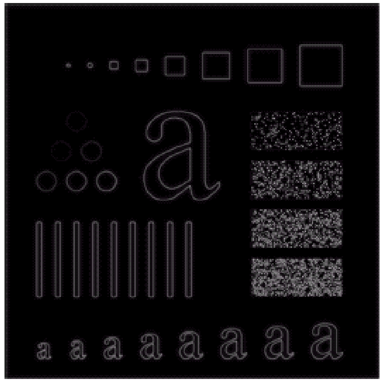
Original image



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 = 80

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

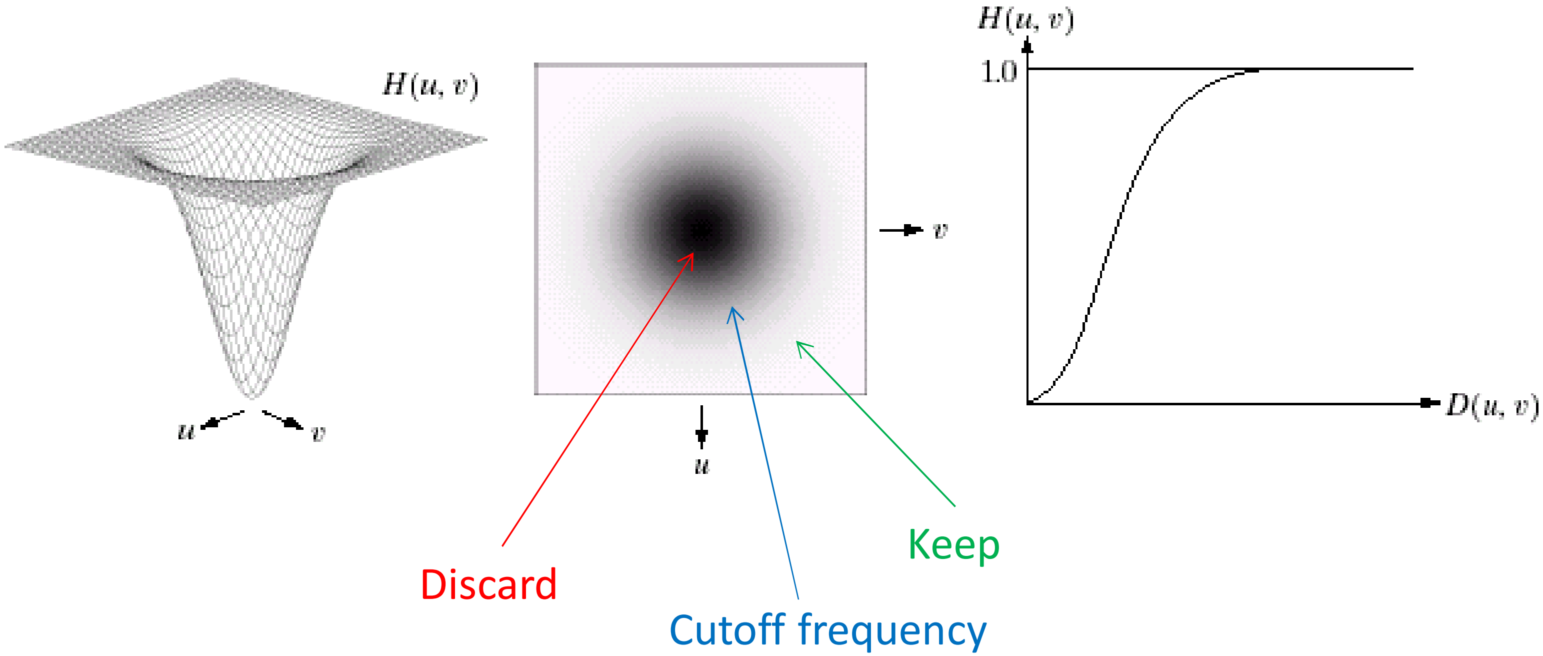
Gaussian Highpass filters (GHPF)

- The Gaussian highpass filter discards low frequencies and keeps high frequencies.
- The transfer function of GHPF is given by :

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

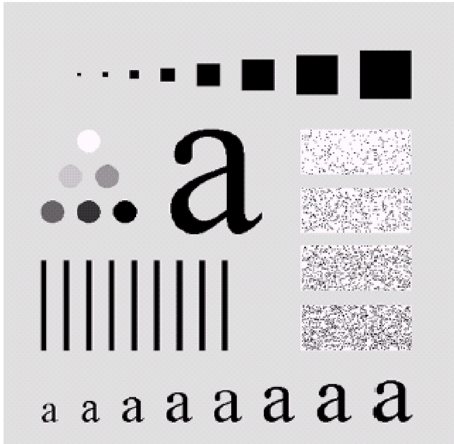
- where :
  - D(u,v) is the distance from the origin
  - D_0 is the cutoff frequency

Visualization of Gaussian Highpass filters

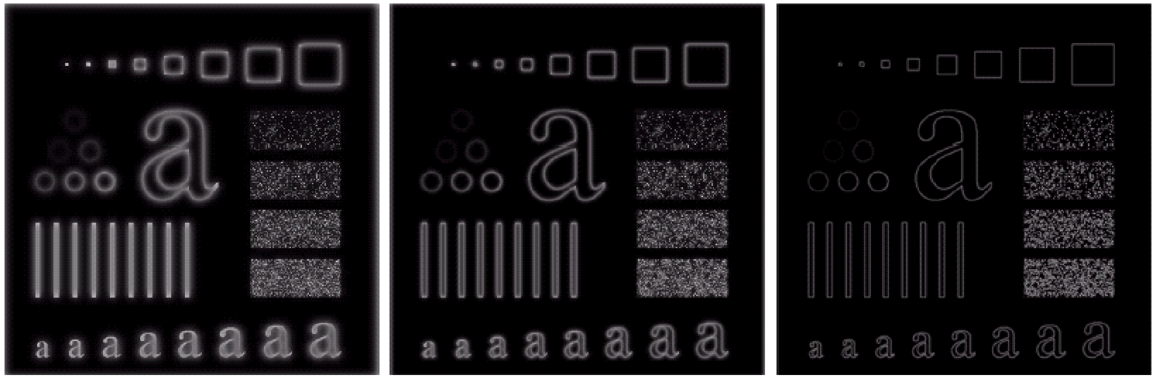


- a) Perspective plot of GHPF
- b) Image representation of GHPF
- c) Cross section of GHPF

Result of Gaussian Highpass filters



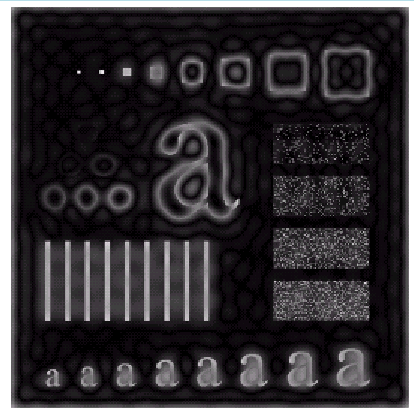
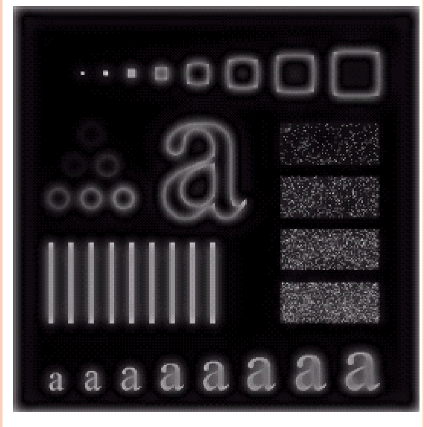
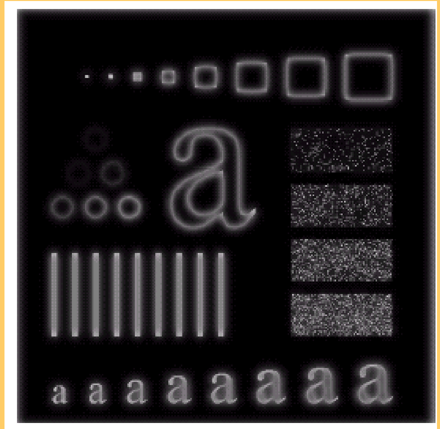
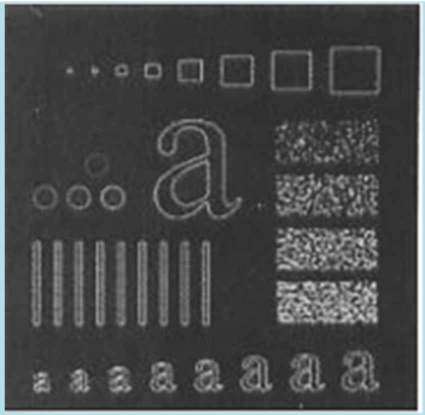
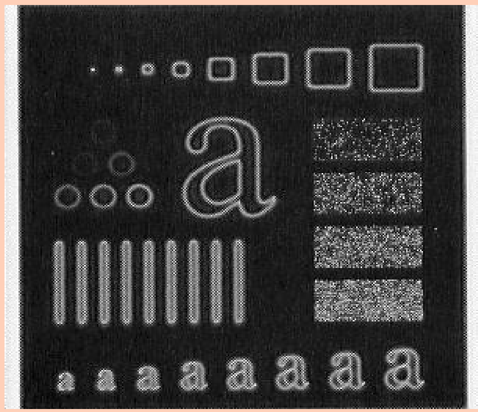
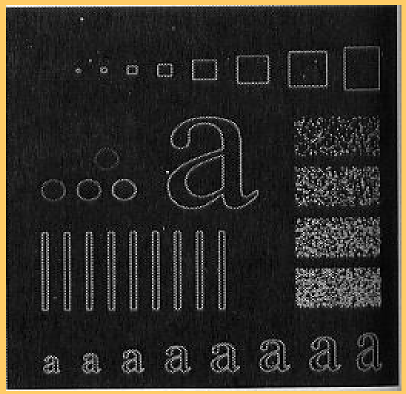
Results of Gaussian high pass filtering with $D_0 = 15$



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

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

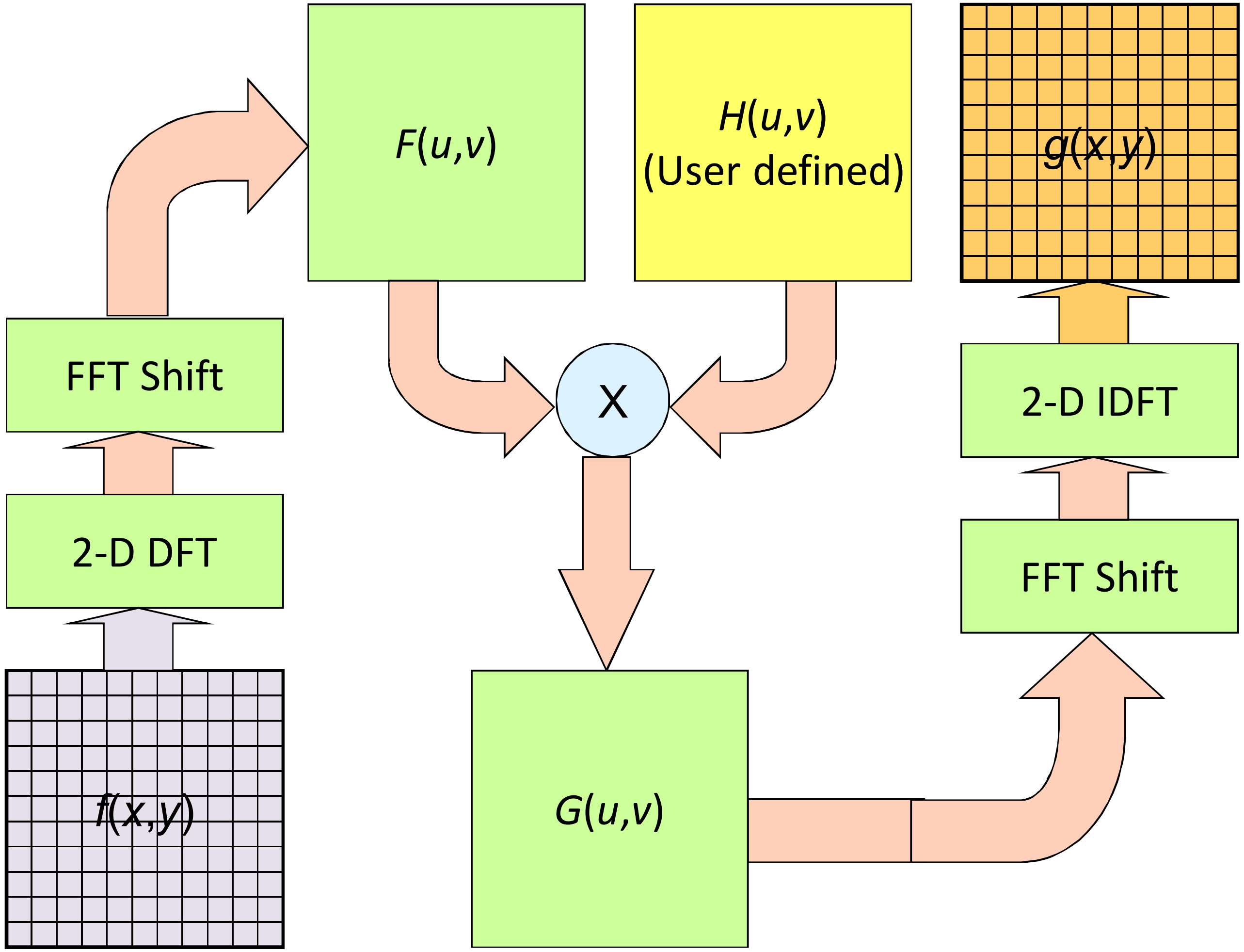
Highpass Filters Compared

	Ideal HPF	**Butterworth HPF**	**Gaussian HPF**
$D_0 = 15$			
$D_0 = 30$			

Observations

- As with ideal low-pass filter, ideal high-pass filter shows significant ringing artifacts.
- Second-order Butterworth high-pass filter shows sharp edges with minor ringing artifacts.
- Gaussian high-pass filter shows good sharpness in edges with no ringing artifacts.

Multiplication in Frequency Domain



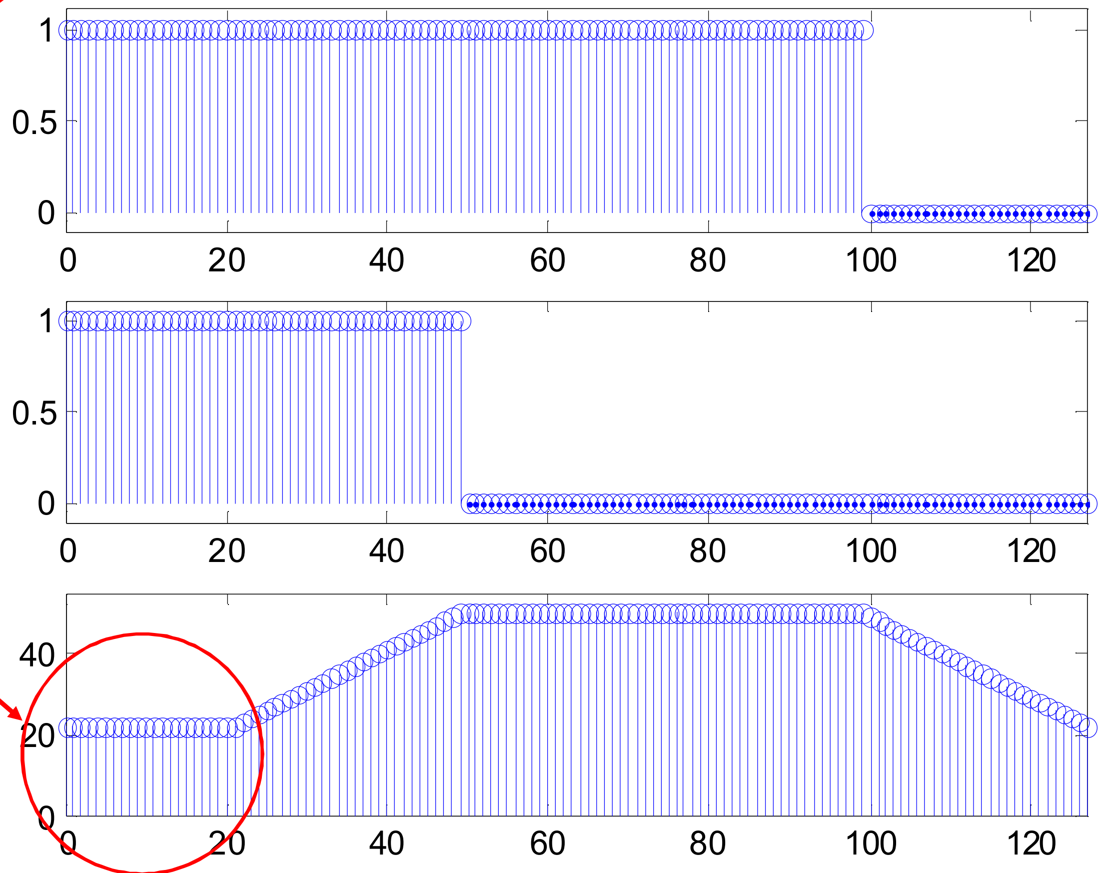
Multiplication in Frequency Domain

$f(x)$ → DFT → $F(u)$ → $G(u) = F(u)H(u)$ → IDFT → $g(x)$

$H(u)$ → $G(u) = F(u)H(u)$

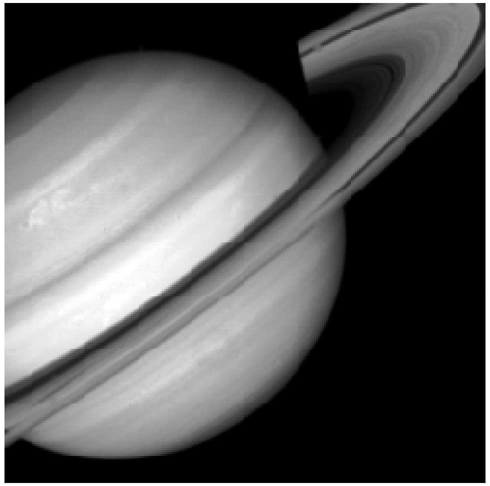
Multiplication of DFTs of 2 signals is equivalent to perform **convolution** in the spatial domain.

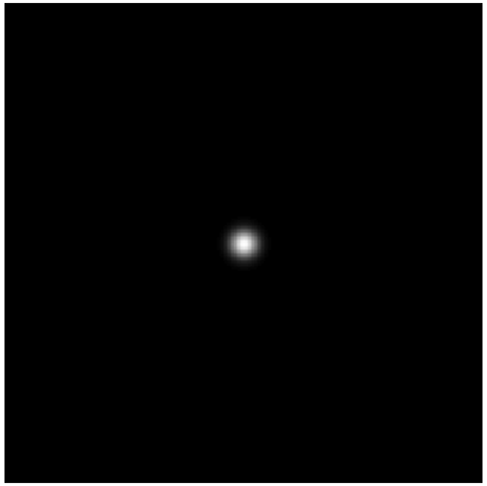
“Wrap around” effect



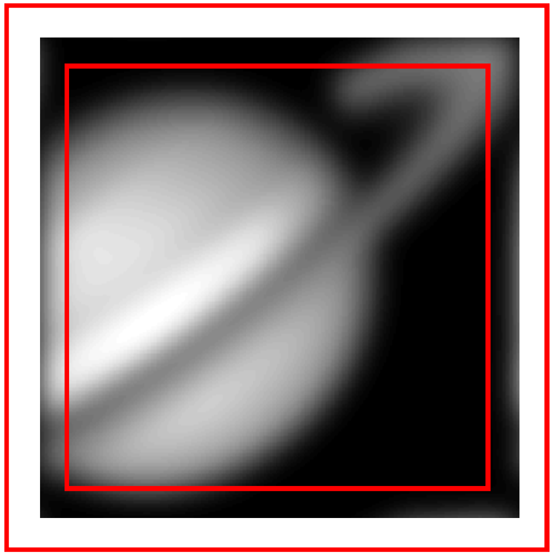
Multiplication in Frequency Domain

Original image





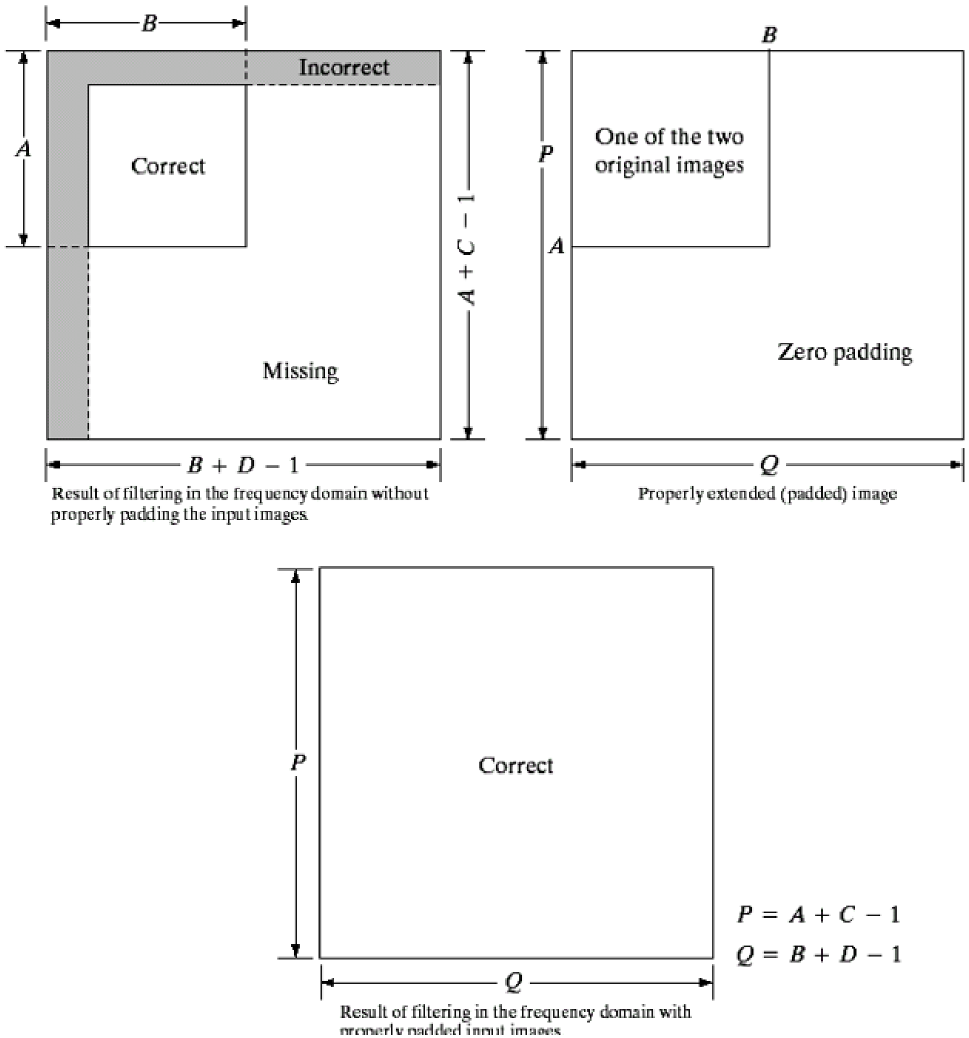
$H(u,v)$ Gaussian Lowpass Filter



Filtered image (obtained using convolution)

Incorrect areas at image rims (Wrap around effect)

Multiplication in Frequency Domain using zero Padding



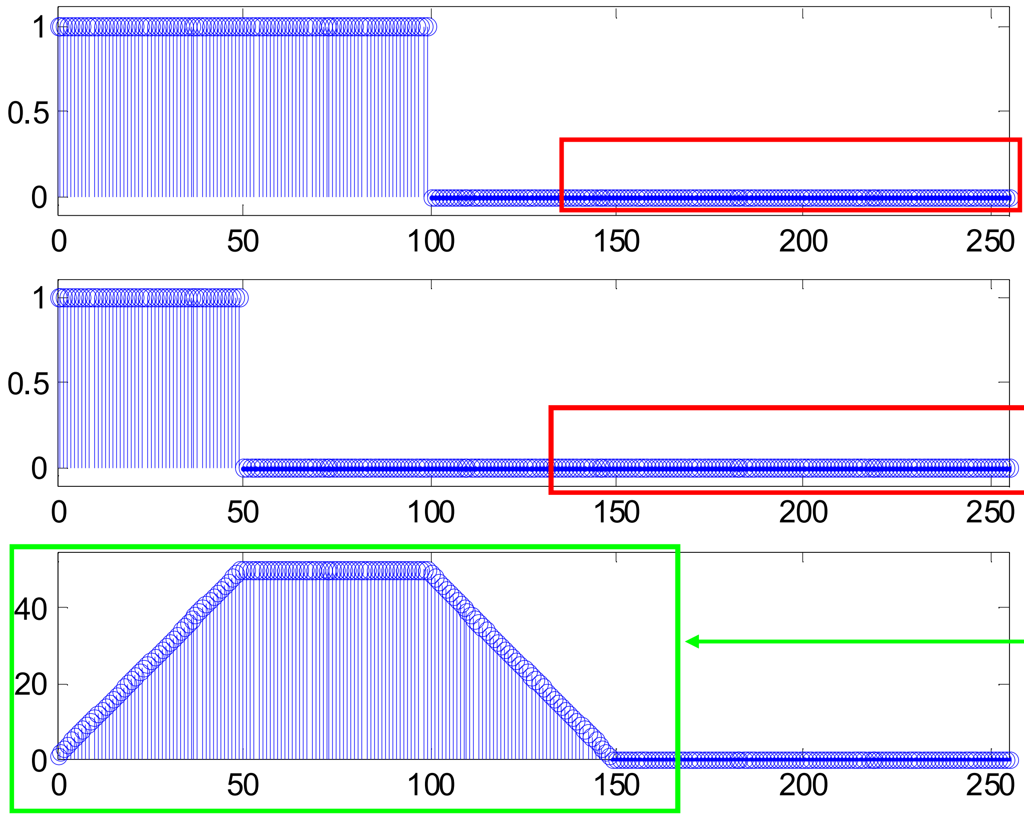
$P = A + C - 1$

$Q = B + D - 1$

Multiplication in Frequency Domain using zero Padding

f(*x*) → Zero padding → DFT → *F*(*u*) → *G*(*u*) = *F*(*u*)*H*(*u*) → IDFT → *g*(*x*)

H(*u*) → *G*(*u*) = *F*(*u*)*H*(*u*)

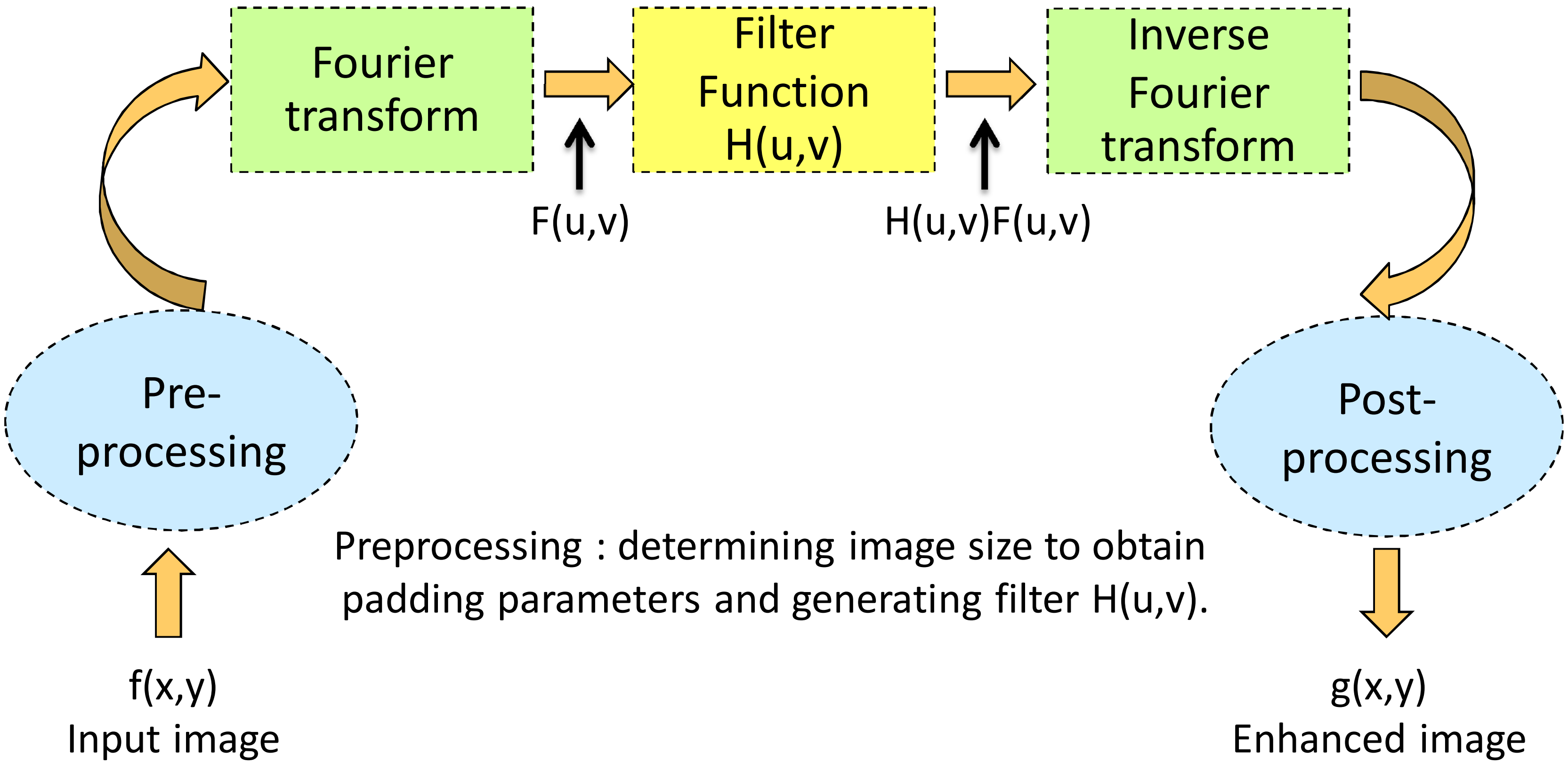


Padding zeros
Before DFT

Keep only this part

Basic steps for filtering in the frequency domain

Frequency domain filtering operation

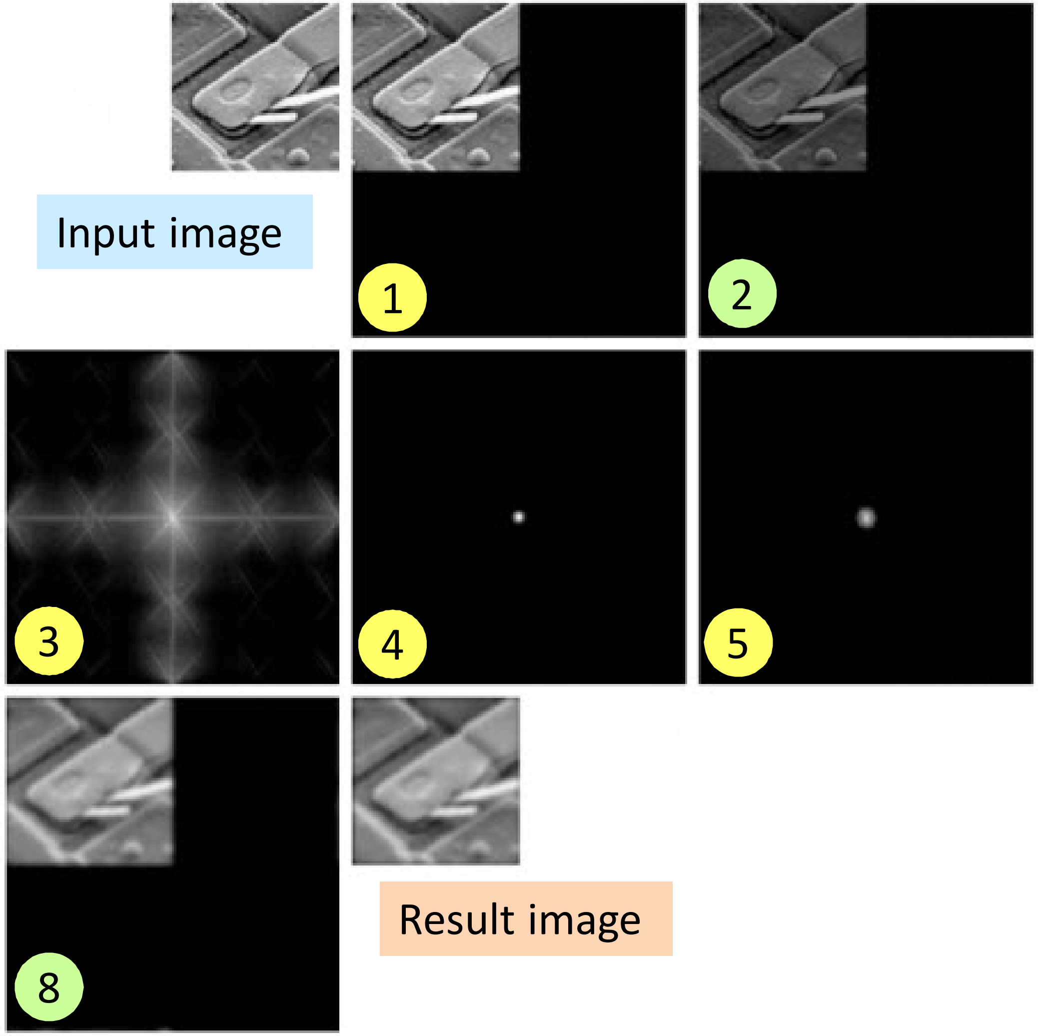


Preprocessing : determining image size to obtain padding parameters and generating filter H(u,v).

Frequency Domain Filter using zero-padding

1. For the input image *f(x,y)* of size *MxN*, form the zero-padded image *fp(x,y)* of size *PxQ* (typically $P = 2M$, $Q = 2N$), where $P \geq 2M - 1$; $Q \geq 2N - 1$
2. Obtain *fc(x,y)* multiplying *fp(x,y)* by -1^{x+y} to center its transform
3. Compute the DFT of the *fc(x,y)*
4. Generate a filter function H(u,v) of size *PxQ* with center at coordinates ($P/2$, $Q/2$)
5. Form the product G(u,v) = H(u,v) F(u,v) via array multiplication
6. Compute the inverse DFT of the G(u,v)
7. Obtain *gc(x,y)* selecting only the real part of the result in (6)
8. Obtain *gp(x,y)* multiplying *gc(x,y)* by -1^{x+y}
9. Finally, extract *g(x,y)* – the *MxN* region from the top left quadrant of *gp(x,y)*

Frequency Domain Filter using zero-padding



Thanks for your attention