Filtering Operation in Frequency Domain-2

Design of filters

Recap

- Filtering in Frequency Domain -Basic Observations
- Filtering in Frequency Domain – Requirements
- What About the Padding for Filters in Frequency Domain?
- Steps for Filtering in the Frequency Domain
- Correspondence Between Filtering in Spatial and Frequency Domain
- Constructing Spatial Filters from Frequency Domain Filters
- Constructing Frequency Domain Filters from Spatial Filters

Lecture Objectives

- Image Smoothing Using Lowpass Frequency Domain Filters
- Image Sharpening Using Highpass Frequency Domain Filters
- Laplacian in the Frequency Domain
- Homomorphic Filtering
- Selective Filtering

Key Stages in DIP

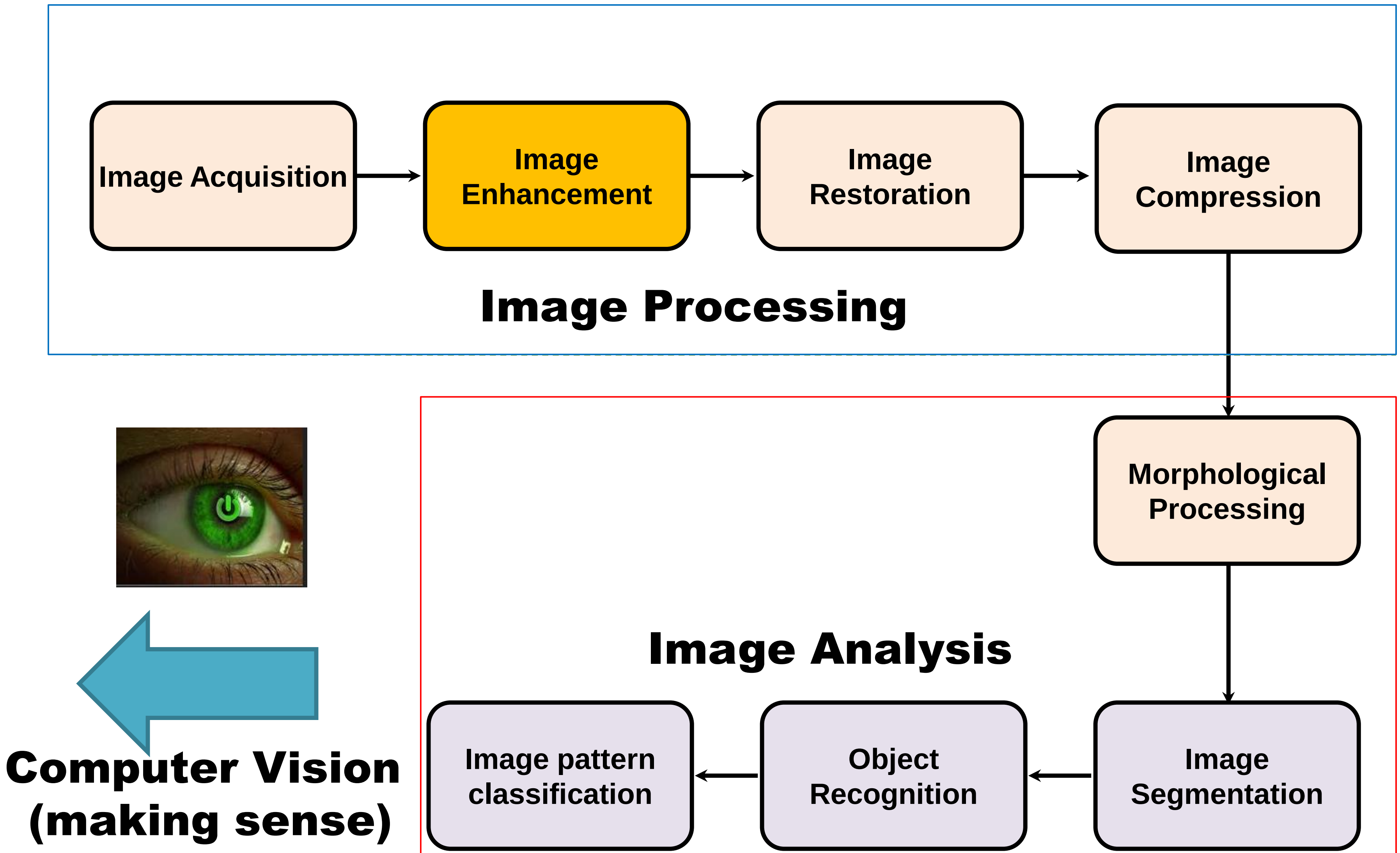


Image Smoothing Using Lowpass Frequency Domain Filters

Smoothing Filters in Frequency Domain

- Low frequencies → uniform regions in image like Walls & Shadows.
- Smoothing → blurring edges and removing regions of abrupt intensity change.
- Smoothing operation → removing high frequency components and allowing lower-frequency components to "pass-through".
- Smoothing filters → low-pass filtering.
- Smoothing filters → zero-phase-shift filters that are radially symmetric.

Image Smoothing Using Lowpass Frequency Domain Filters

- Ideal lowpass filters
- Gaussian lowpass filters
- Butterworth lowpass filters

Ideal Lowpass Filters (ILPF)

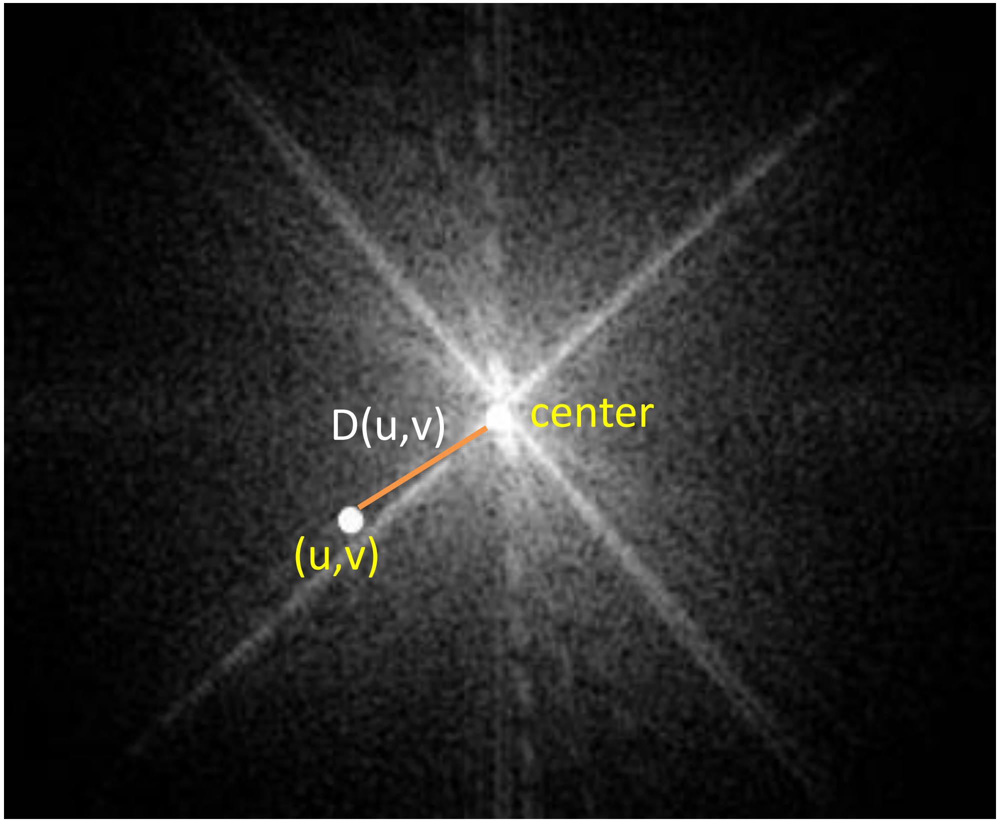
- **Ideal behavior:**
  - Pass frequency values below a threshold frequency.
  - Cut off frequency values above the threshold value.

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

where D_0 is a positive constant, and $D(u,v)$ is the distance between a point (u,v) in the frequency domain and the center of the $P \times Q$ frequency rectangle; that is,

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

Ideal Lowpass Filters (ILPF)

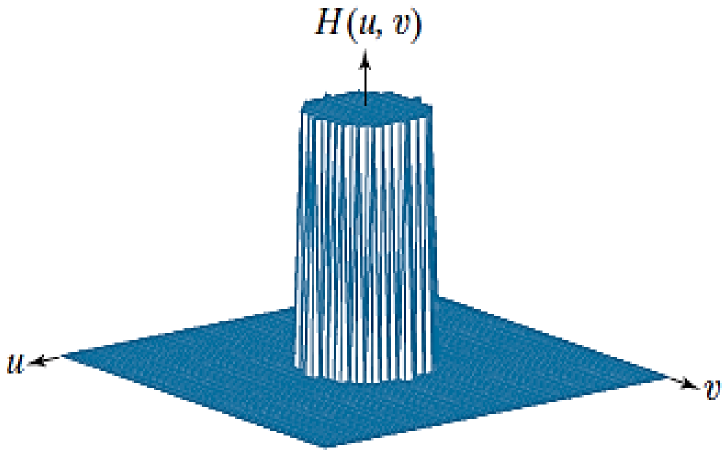


Frequency Rectangle

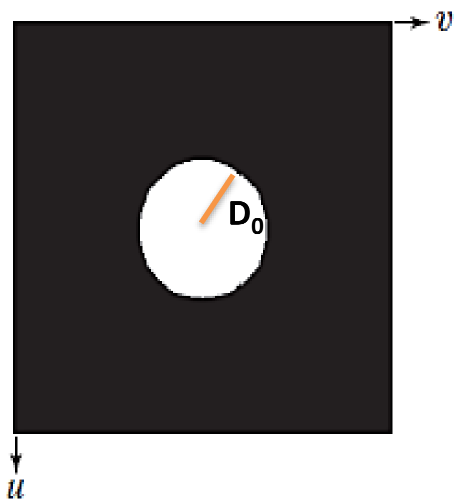
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

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

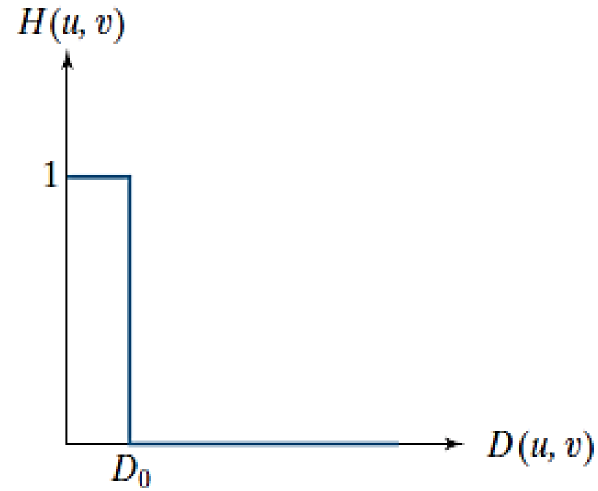
ILPF Representation



Perspective Plot



Filter Image

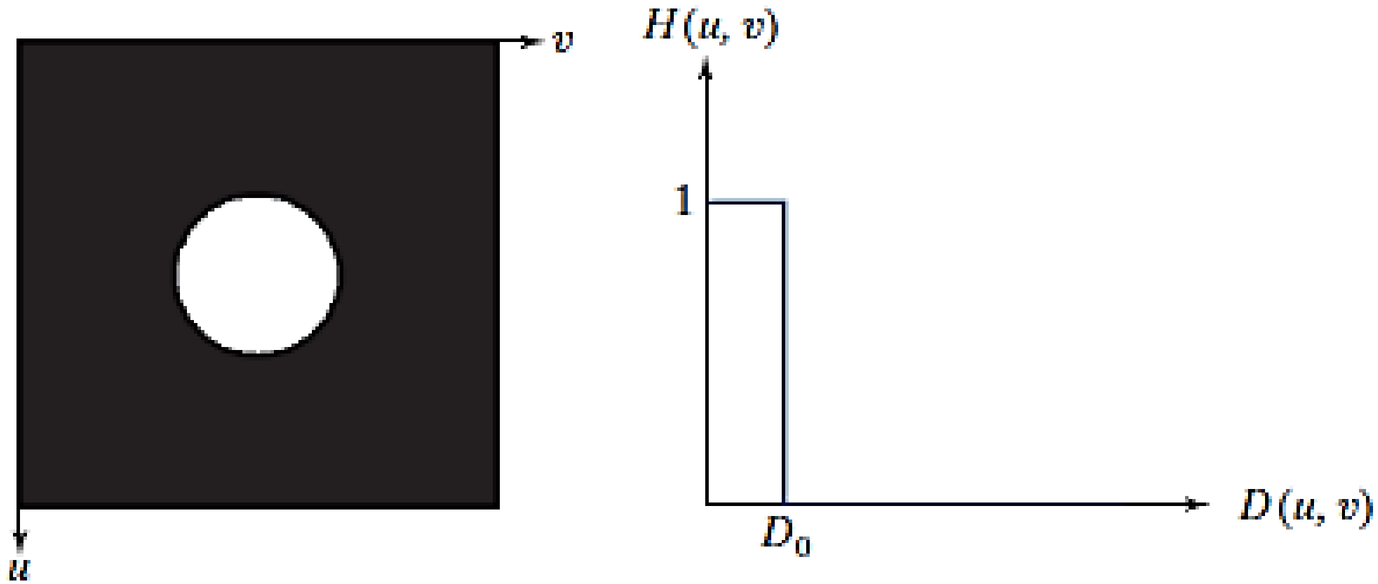


Radial Cross Section

- All frequencies on or inside a **circle of radius** D_0 are passed without attenuation

- All frequencies outside the circle are completely attenuated (filtered out).

Cutoff Frequency



The ***cutoff frequency*** $\mathbf{D_0}$ is a point of transition between the values $H(u,v) = 1$ and $H(u,v) = 0$.

Establishing Cutoff Frequency Circle

- The LPFs are compared based on their ***cutoff frequencies***.

- One way to obtain a certain *cutoff frequency circle* in the H(u,v) plot enclosing $\boldsymbol{\alpha}$ percentage of the power spectrum is by finding the ratio of the **power enclosed in a circular region** of radius **D_0** and the **total image power** **P_T**.

$$P_T = \sum_{u=0}^{P-1} \sum_{v=0}^{Q-1} P(u, v)$$

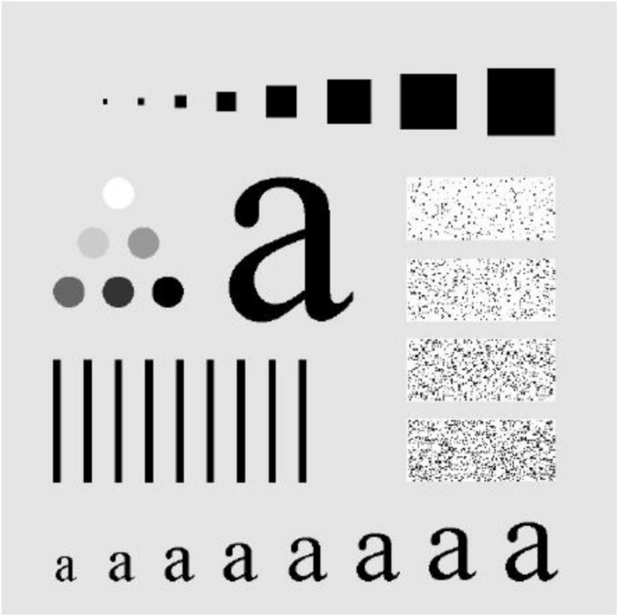
Where,

$$\begin{aligned} P(u,v) &= |F(u,v)|^2 \\ &= R^2(u,v) + I^2(u,v) \end{aligned}$$

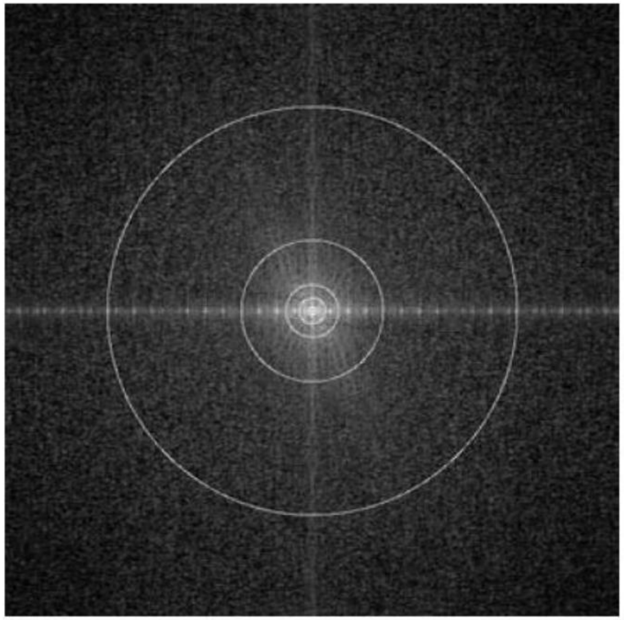
$$\alpha = 100 \left[\sum_u \sum_v P(u,v)/P_T \right]$$

Where the summation is over values of (u,v) that lie inside the circle or on its boundary.

ILPF - Example



688 × 688 size image

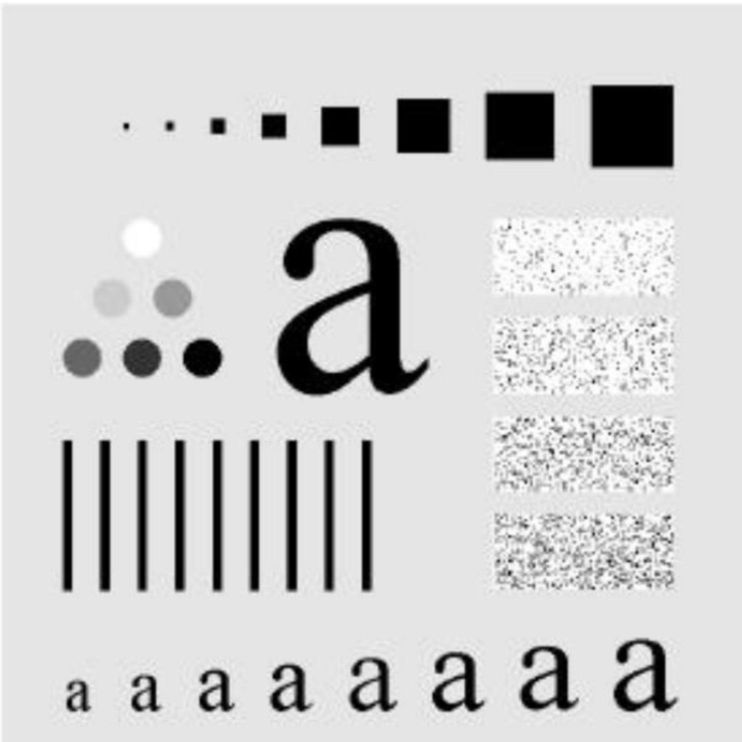


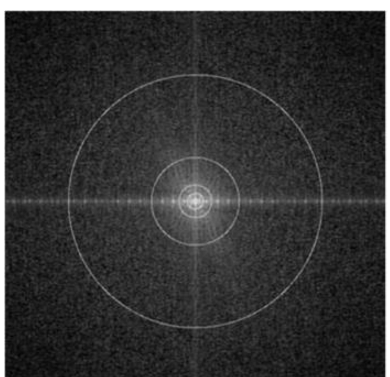
its spectrum

Radii in number of pixels D_0	10	30	60	160	460
Total image power enclosed	86.9	92.8	95.1	97.6	99.4

ILPF – Example

$D_0 = 10$

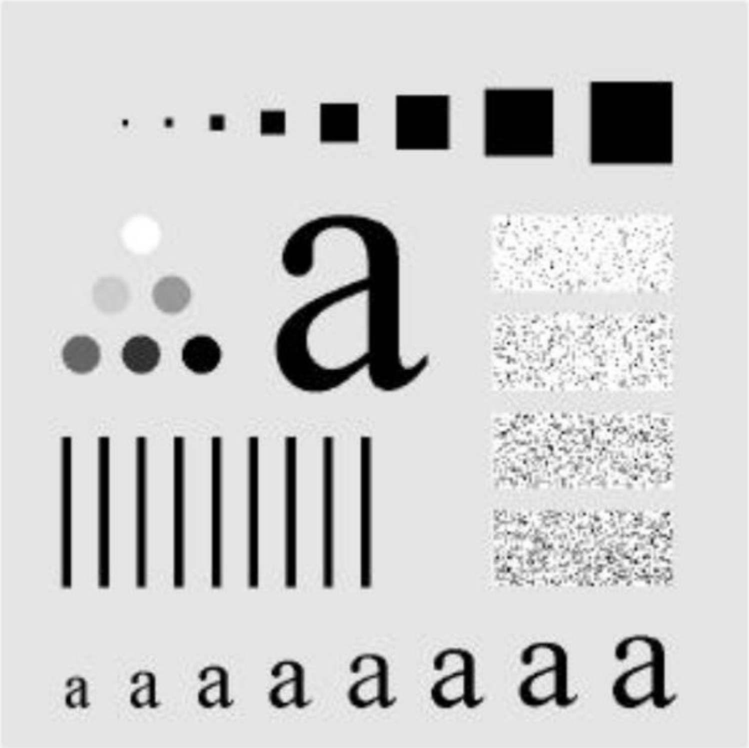


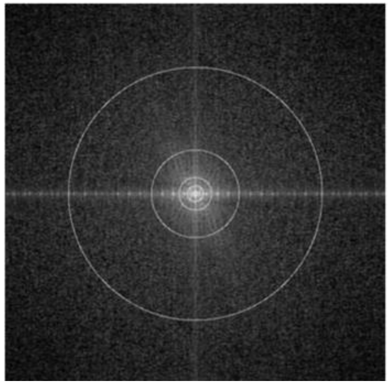


Power Retained	86.9
Power Lost	13.1

ILPF – Example

$D_0 = 30$

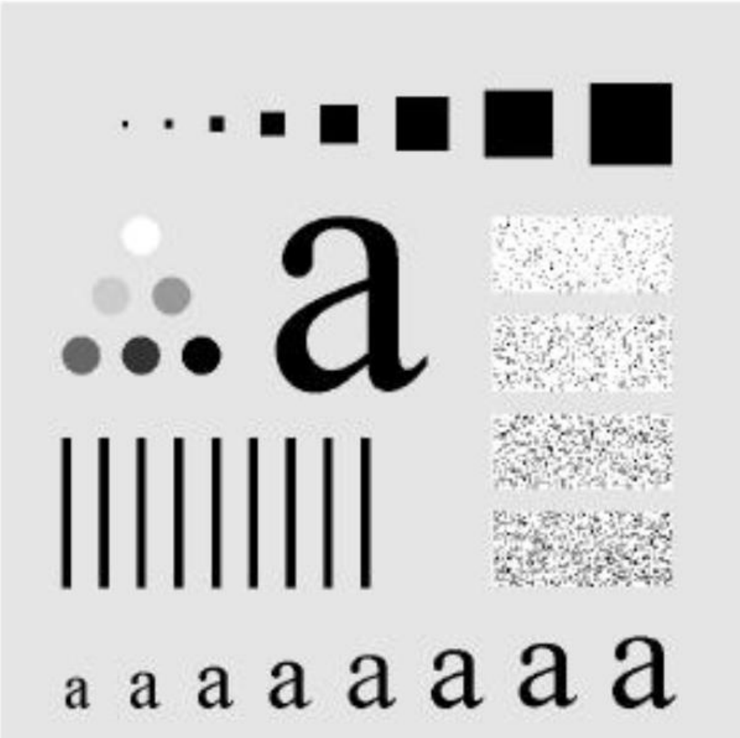


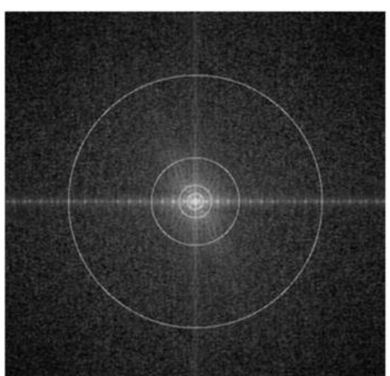


Power Retained	92.8
Power Lost	7.2

ILPF – Example

$D_0 = 60$

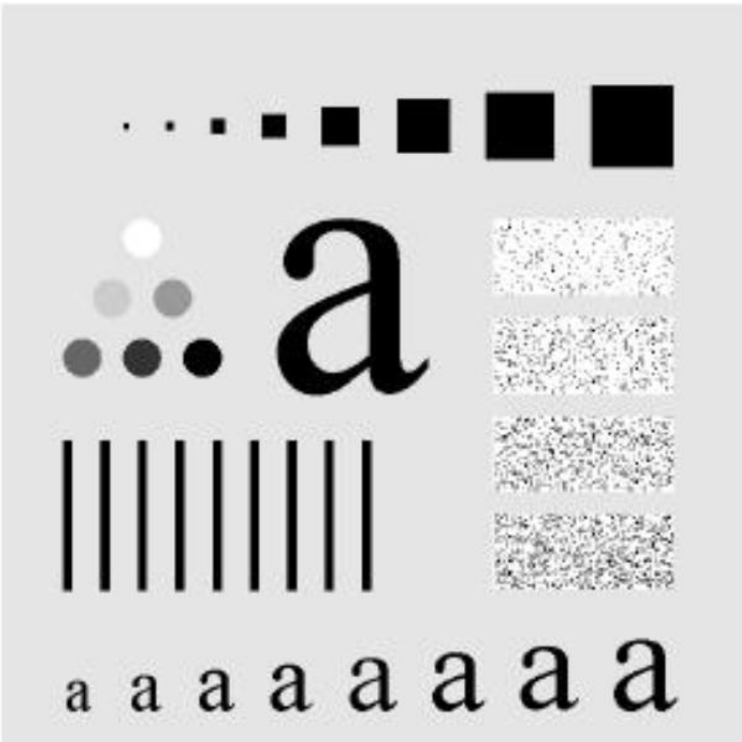


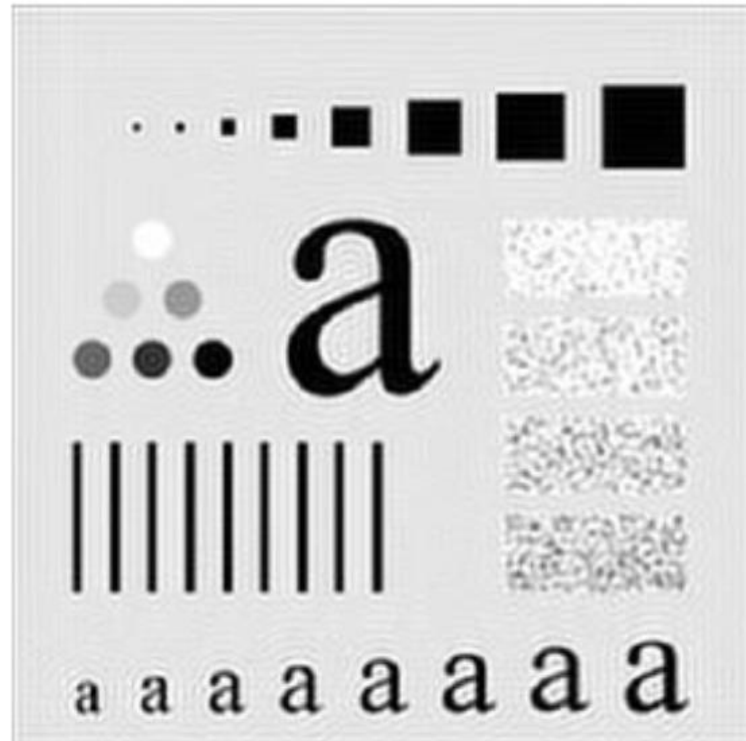


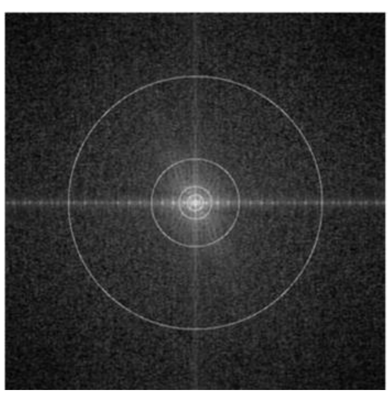
Power Retained	95.1
Power Lost	4.9

ILPF – Example

$D_0 = 160$



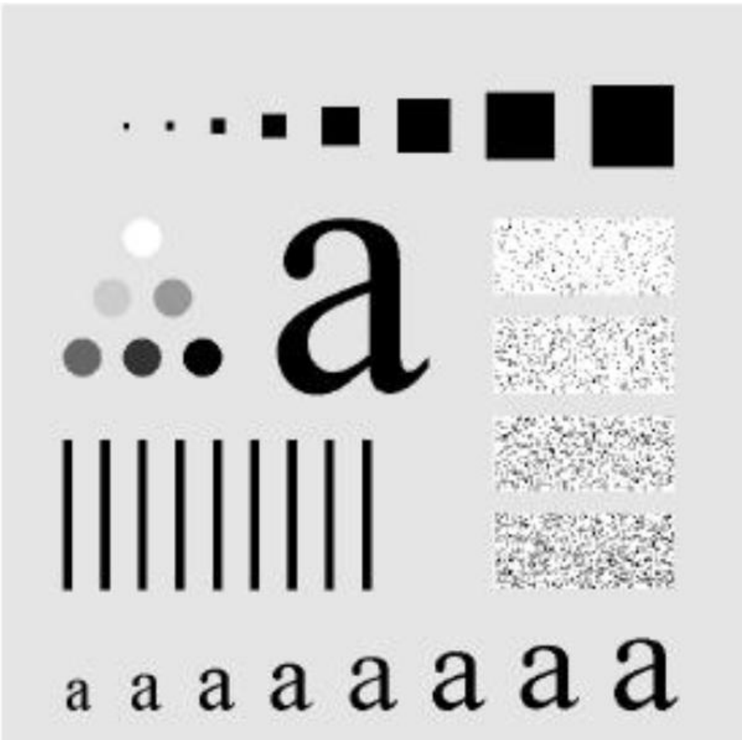


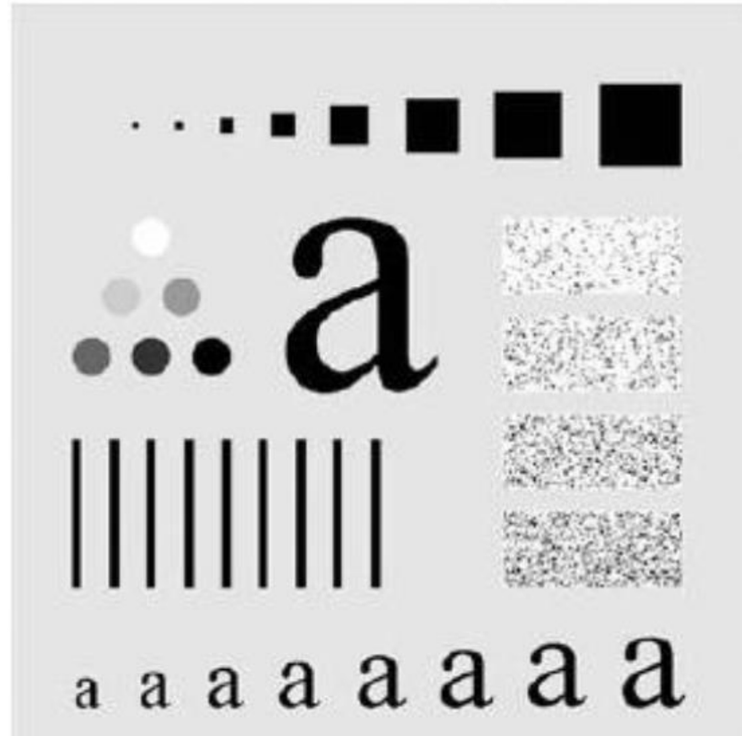


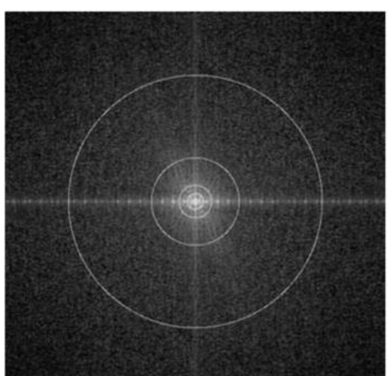
Power Retained	97.6
Power Lost	2.4

ILPF – Example

$D_0 = 460$





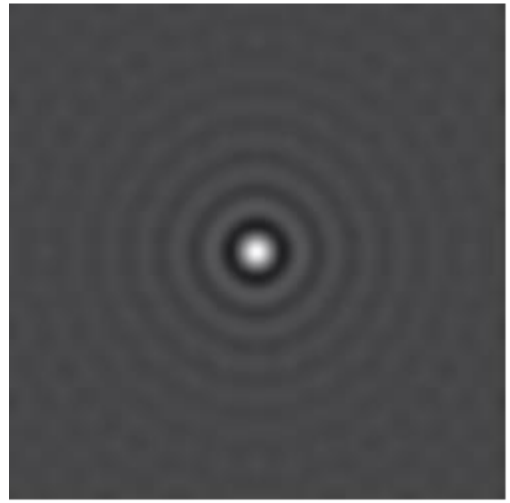


Power Retained	99.4
Power Lost	0.6

ILPF Blurring and Ringing Artifacts

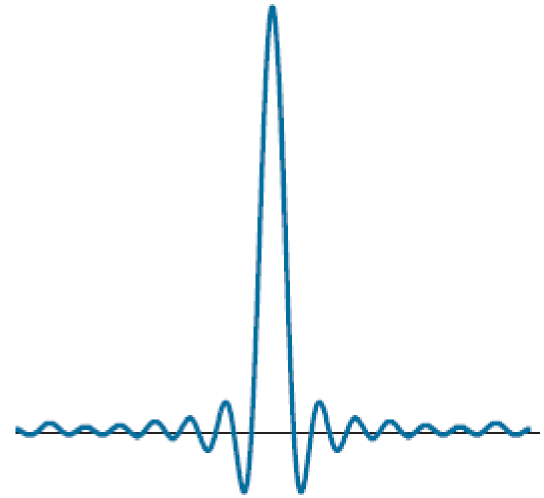
H(u,v)

Image of frequency domain ILPF transfer function of size = 1000 × 1000, radius = 15



h(x,y)

Spatial representation of H(u,v) by taking IDFT of it



Intensity profile

- During filtering in spatial domain, we convolve h(x,y) with the image.

- The center lobe of this spatial function h(x,y) is the principal cause of blurring, while the outer, smaller lobes are mainly responsible for ringing.

How to solve Ringing Artifacts?

- Gaussian Low Pass Filters
- Butterworth Lowpass Filters

2D Gaussian Lowpass Filter (GLPF)

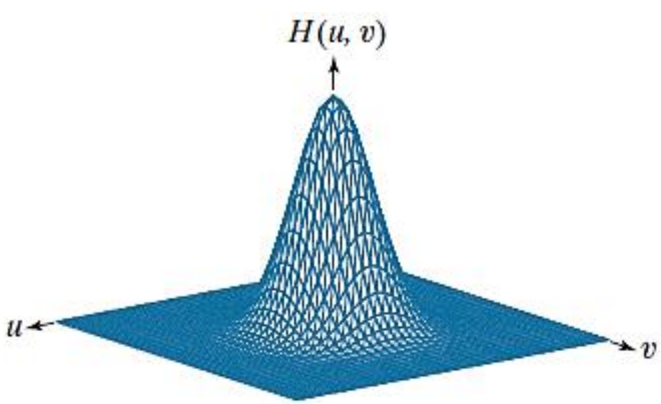
GLPF transfer function has the form:

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

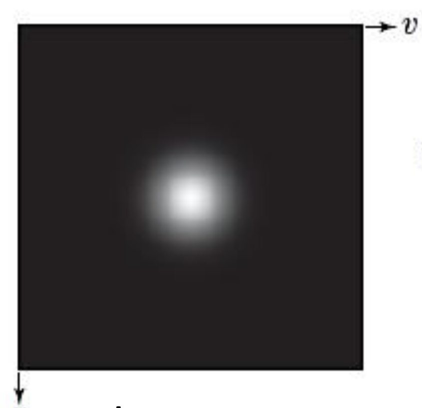
Since both σ and D_0 give the measure of spread about the center, by letting $\sigma = D_0$ we get:

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

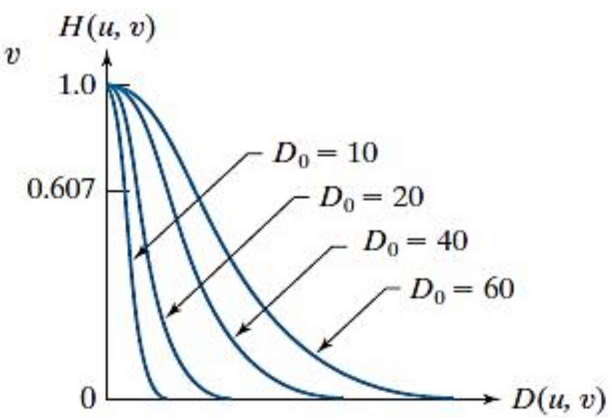
- D(u,v) is the distance from the center of the P×Q frequency rectangle to any point (u,v) contained by this rectangle.
- **D_0** is the cutoff frequency.



Perspective Plot



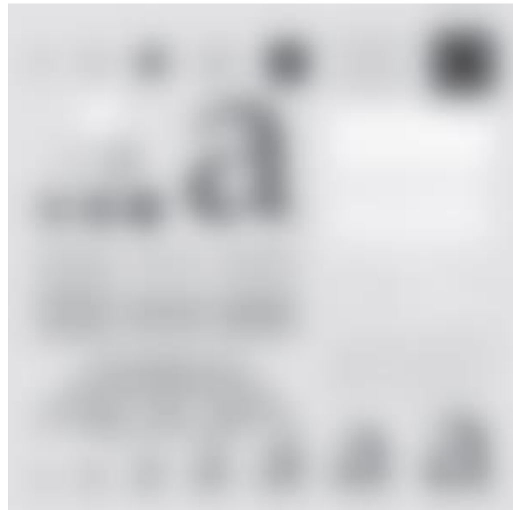
Filter Image



Radial Cross Section

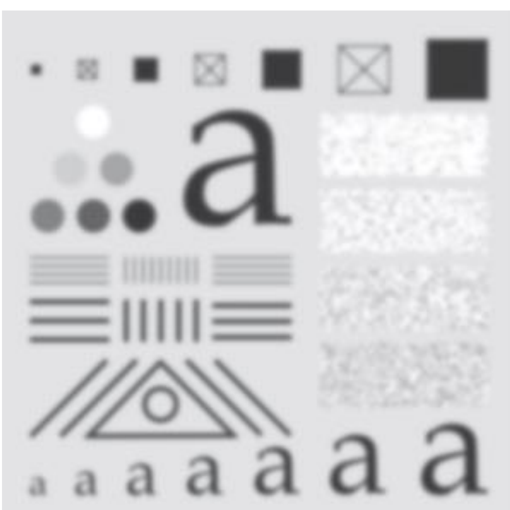
2D Gaussian Lowpass Filter (GLPF)

Original image



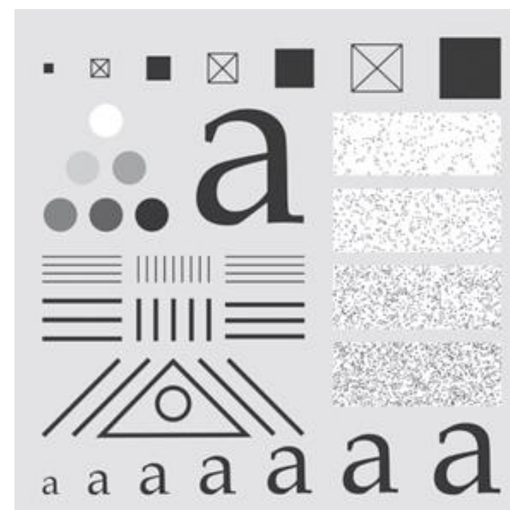
$D_0 = 10$

$D_0 = 30$



$D_0 = 60$

$D_0 = 160$

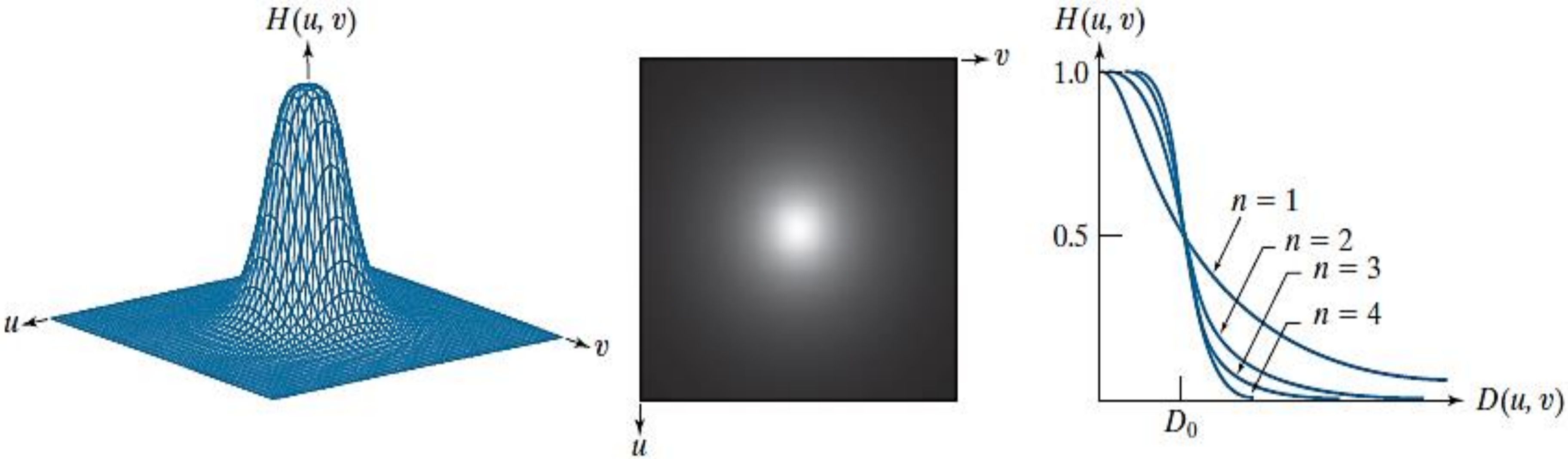


$D_0 = 460$

Butterworth Lowpass Filters (BLPF)

- Butterworth lowpass filter of *order* **n**, with *cutoff frequency* at a distance **D_0** from the center of the frequency rectangle is defines as:

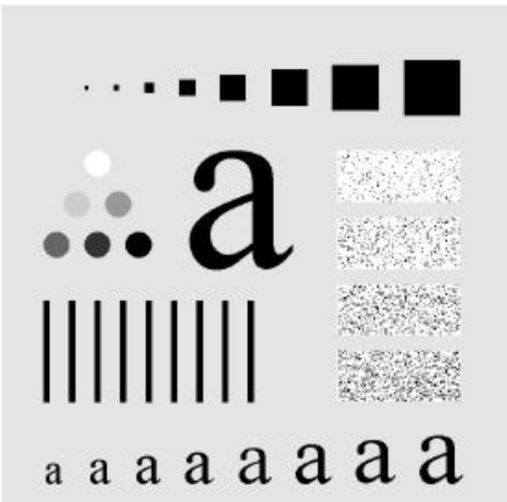
$$H(u,v) = \frac{1}{1 + \left[D(u,v)/D_0 \right]^{2n}}$$



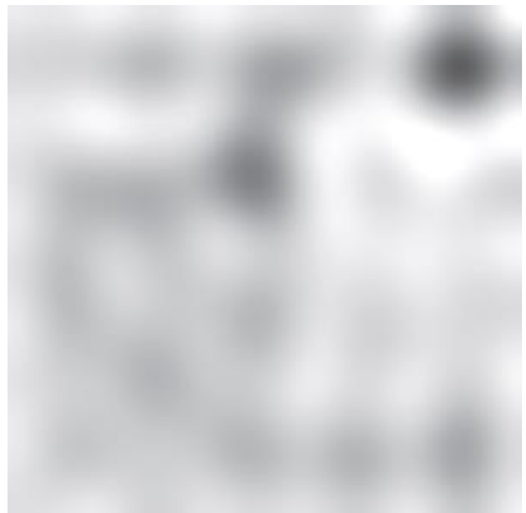
- The BLPF function can be controlled to approach the characteristics of the **ILPF** and **GLPF** using **higher values of *n*** and **lower values of *n*** respectively.
- The BLPF provide a smooth transition from low to high frequencies. Thus, having considerably **less ringing**.

Butterworth Lowpass Filters (BLPF)

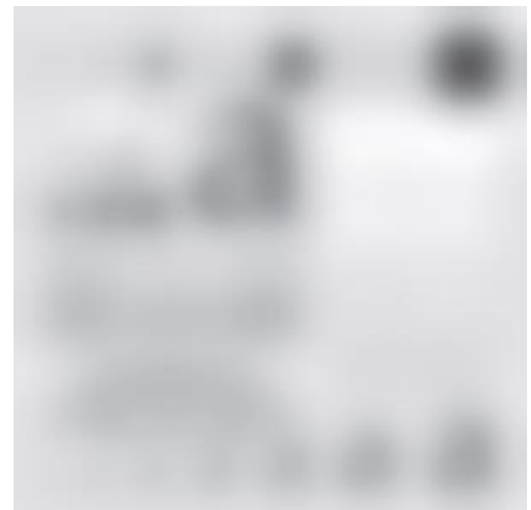
$D_0 = 10, n = 2.25$



Original image



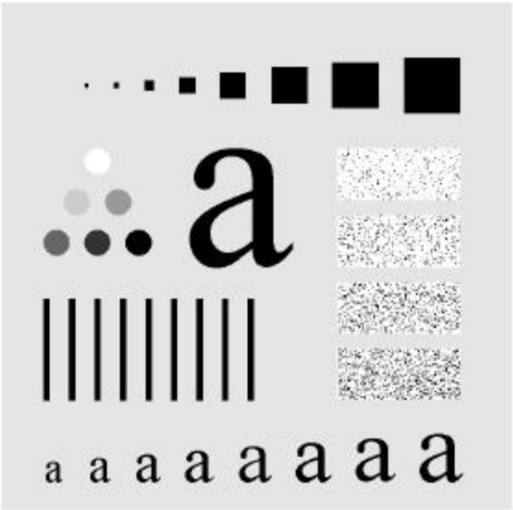
ILPF



BLPF

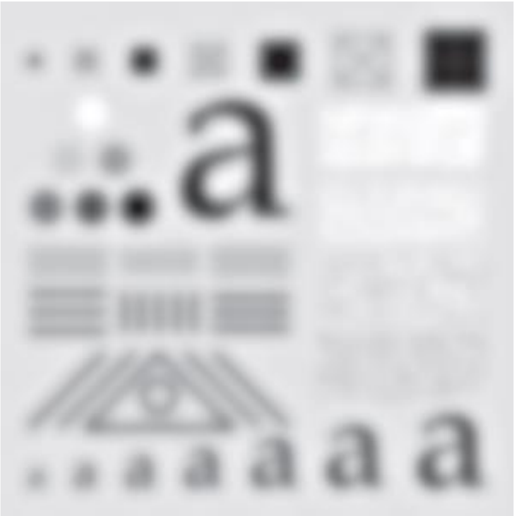
Butterworth Lowpass Filters (BLPF)

$D_0 = 30$, $n = 2.25$



Original image

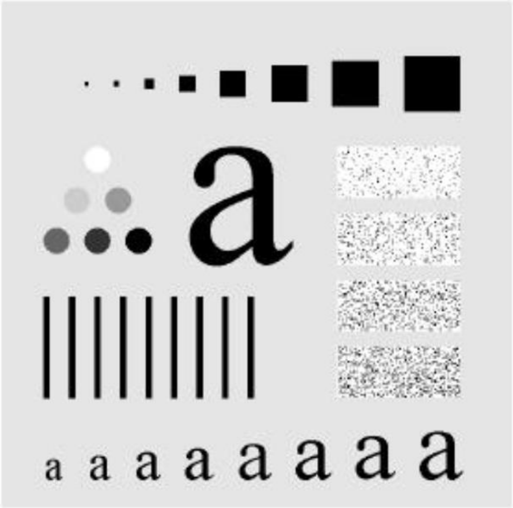
ILPF



BLPF

Butterworth Lowpass Filters (BLPF)

$D_0 = 60, n = 2.25$



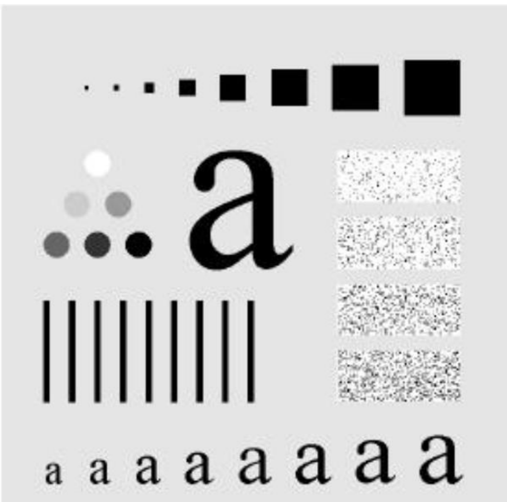
Original image

ILPF

BLPF

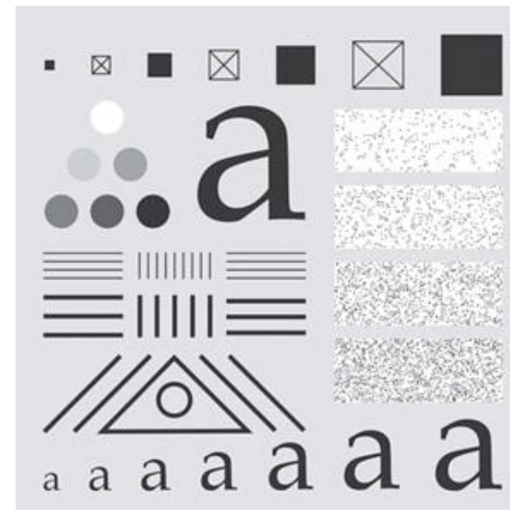
Butterworth Lowpass Filters (BLPF)

D_0 = 160, n = 2.25



Original image

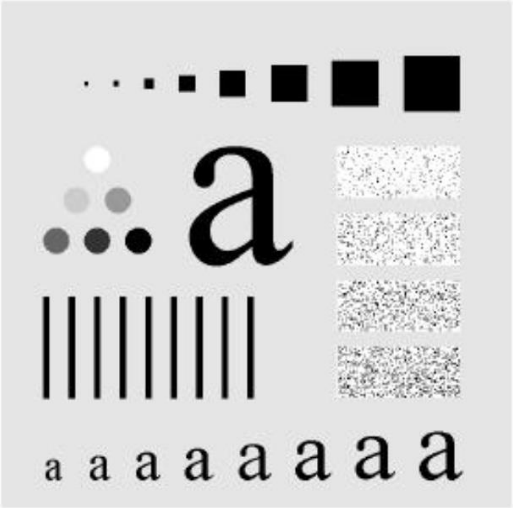
ILPF



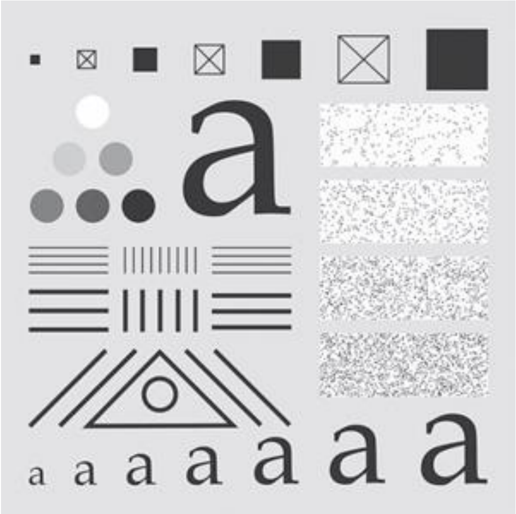
BLPF

Butterworth Lowpass Filters (BLPF)

$D_0 = 460$, $n = 2.25$



Original image



ILPF

BLPF

Facts about BLPF

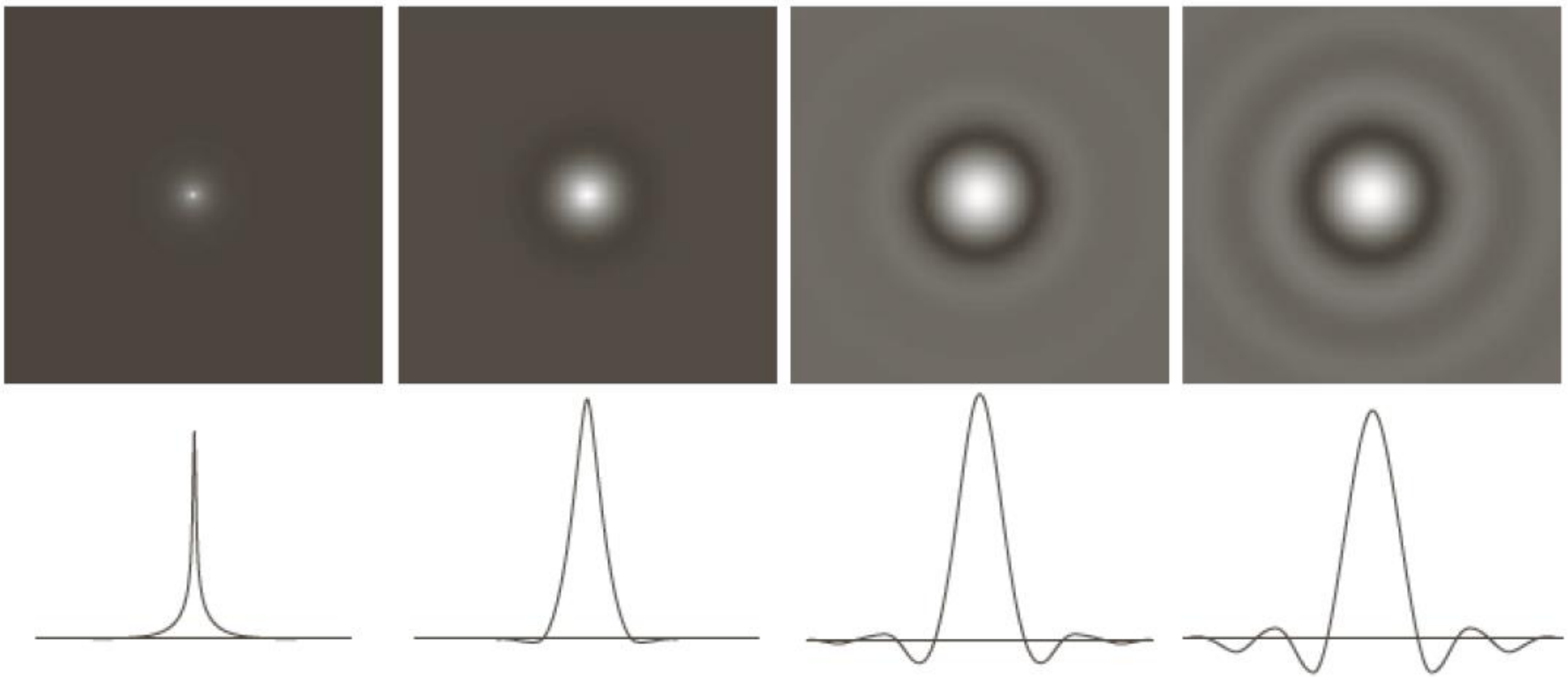
- BLPF performs better than the ILPF in all cases.

- The spatial domain kernel obtainable from a BLPF of order 1 has no ringing.

- Ringing artifacts are not “visible” for BLPF of order 2 or 3.

- BLPFs of orders 2 to 3 are a good compromise between effective lowpass filtering and acceptable spatial-domain ringing.

Comparisons of BLPF based on their Order



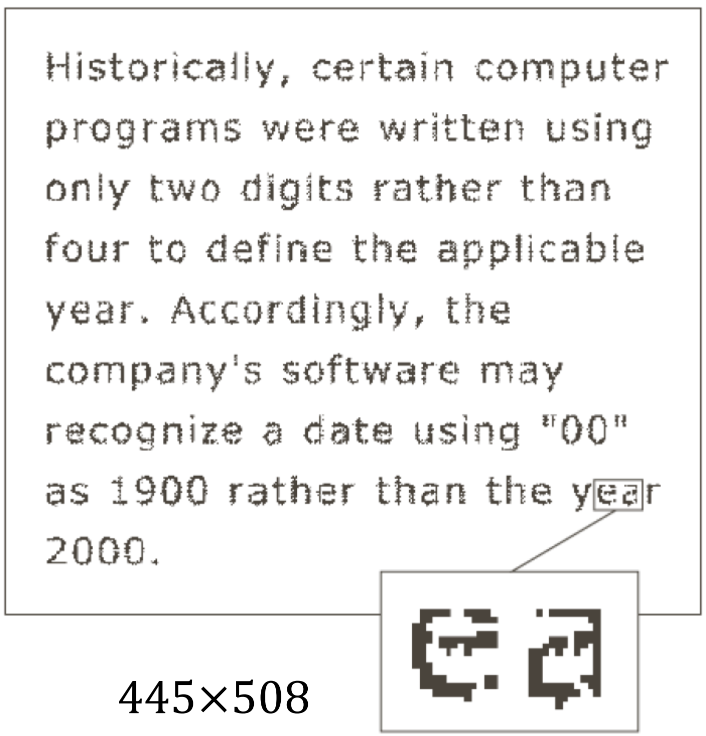
n=1 n=2 n=5 n=20

Spatial kernels corresponding to BFPF transfer functions of size = 1000 × 1000, cutoff frequency D_0= 5

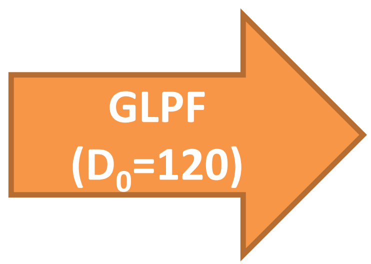
Lowpass Filtering Example using GLPF

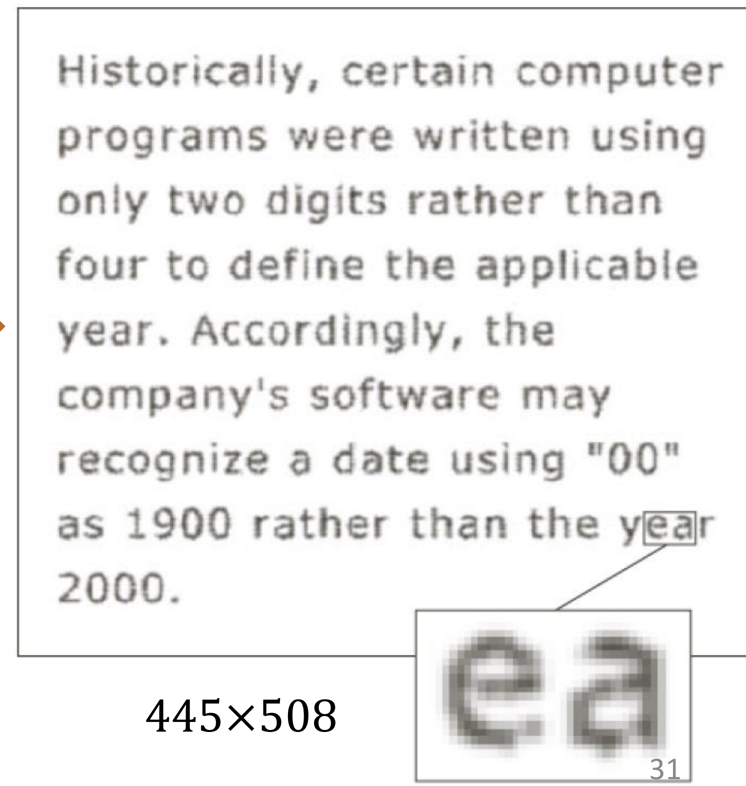
Character Recognition

Conversion of scanned or photographed images of typewritten or printed text into machine-encoded/computer-readable text.


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.

445×508




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.

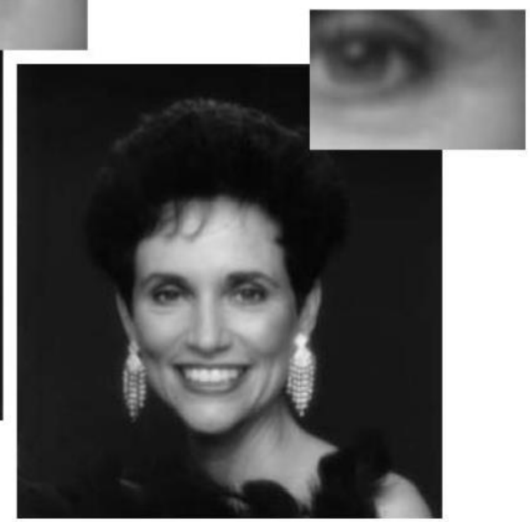
445×508

Lowpass Filtering Example using GLPF

Cosmetic Processing

784×732

GLPF ($D_0 = 150$)



GLPF ($D_0 = 130$)

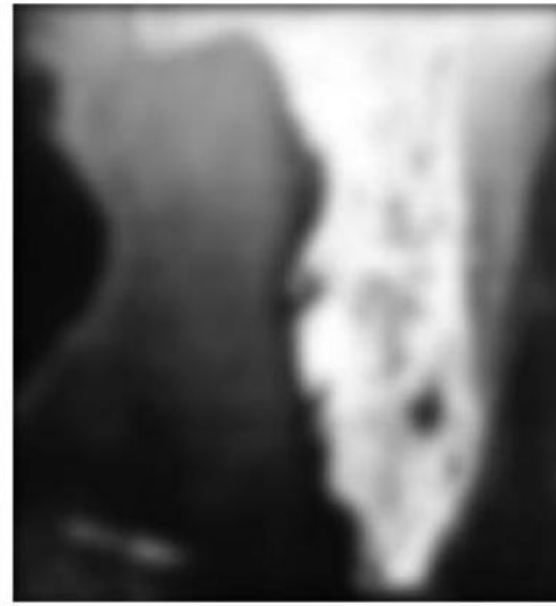
Smoothing provides a “softer”, pleasing photograph by removing skin blemishes and wrinkles

Lowpass Filtering Example using GLPF

Filtering in Aerial Images

808×754

GLPF (D_0=50)



GLPF (D_0=20)

- **D_0=50** : reduced scan lines simplifies the detection of features such as the interface boundaries between Ocean currents.
- **D_0=20** : blurs out as much detail as possible while leaving large features recognizable such as Lake region (nearly round dark region at the bottom right).

Image Sharpening Using Highpass Frequency Domain Filters

Sharpening Filters in Frequency Domain

- High frequencies → edges in the image.
- Sharpening → enhancing edges and removing regions of gradual change.
- Sharpening operation → removing low frequency components and allowing higher-frequency components to “pass-through”.
- Sharpening filters → high-pass filtering.
- Sharpening filters → zero-phase-shift filters that are radially symmetric.

Image Smoothing Using Lowpass Frequency Domain Filters

- Ideal highpass filters
- Gaussian highpass filters
- Butterworth highpass filters

Highpass Filter

A $P \times Q$ highpass filter is obtained from a given lowpass filter using:

$$H_{HP}(u,v) = 1 - H_{LP}(u,v)$$

where $H_{LP}(u,v)$ is the transfer function of the lowpass filter.

$H_{LP}(u,v)$ can be any **LPF** like ideal, butterworth or Gaussian.

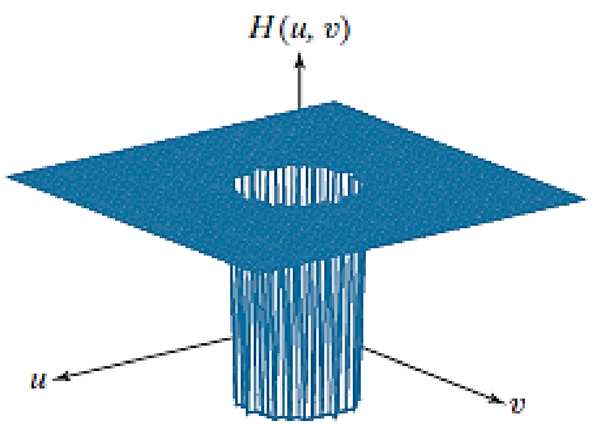
Ideal Highpass Filter (IHPF)

A 2D ideal highpass filter (IHPF) is defined as:

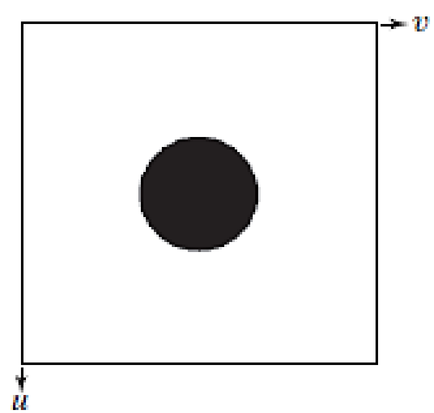
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0, \end{cases}$$

where D_0 is a positive constant, and $D(u,v)$ is the distance between a point (u,v) in the frequency domain and the center of the $P \times Q$ frequency rectangle; that is,

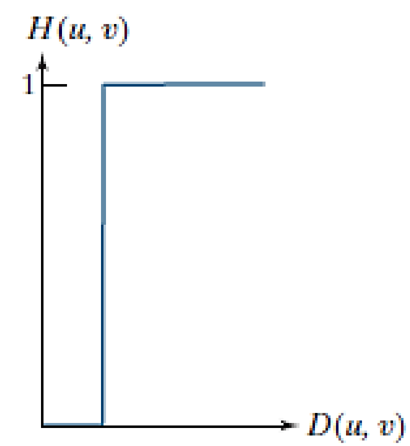
$$D(u,v) = \left[(u - P/2)^2 + (v - Q/2)^2 \right]^{1/2}$$



Perspective Plot



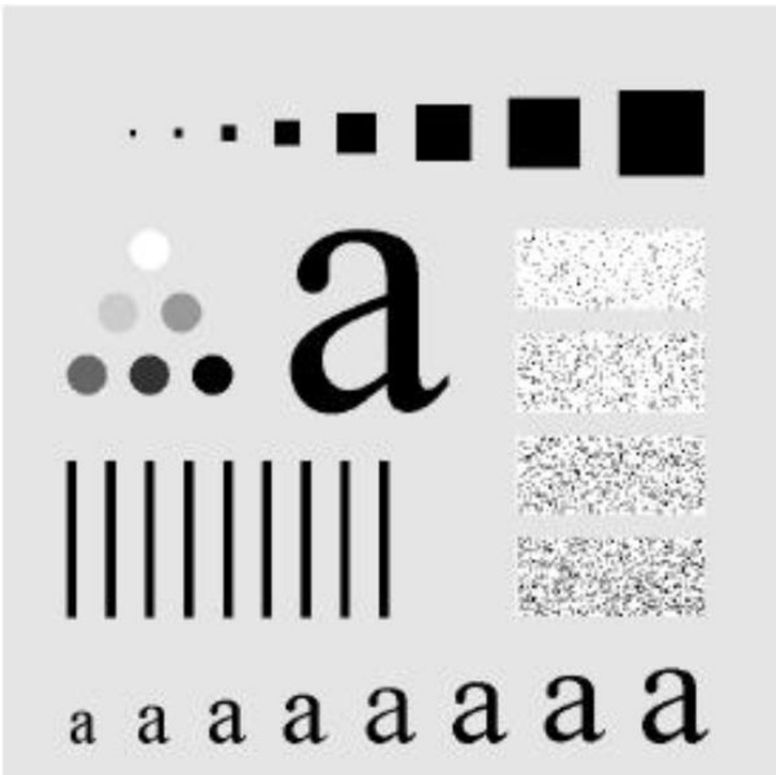
Filter Image

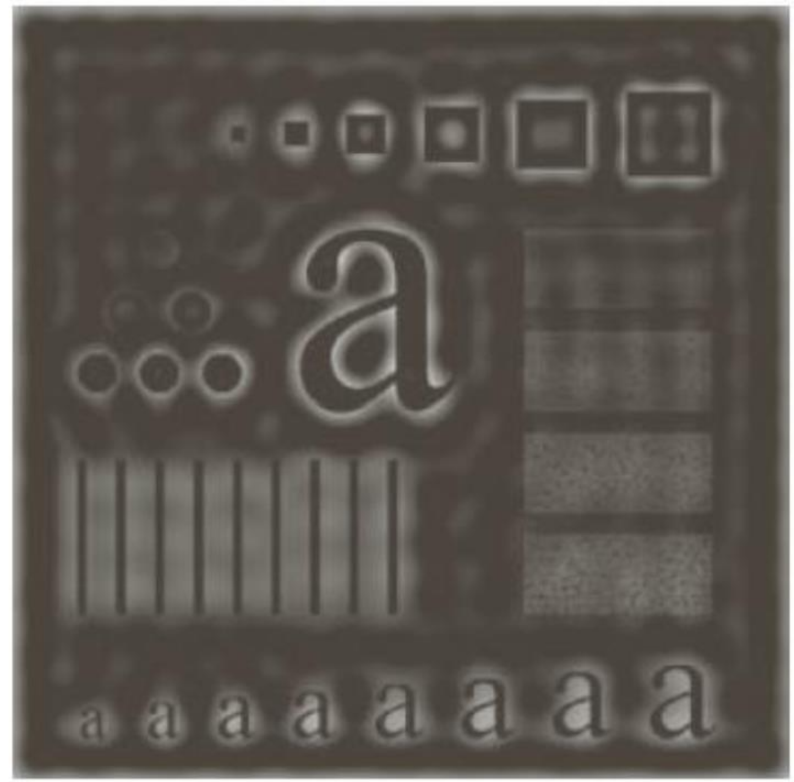


Radial Cross Section

Ideal Highpass Filter (IHPF)

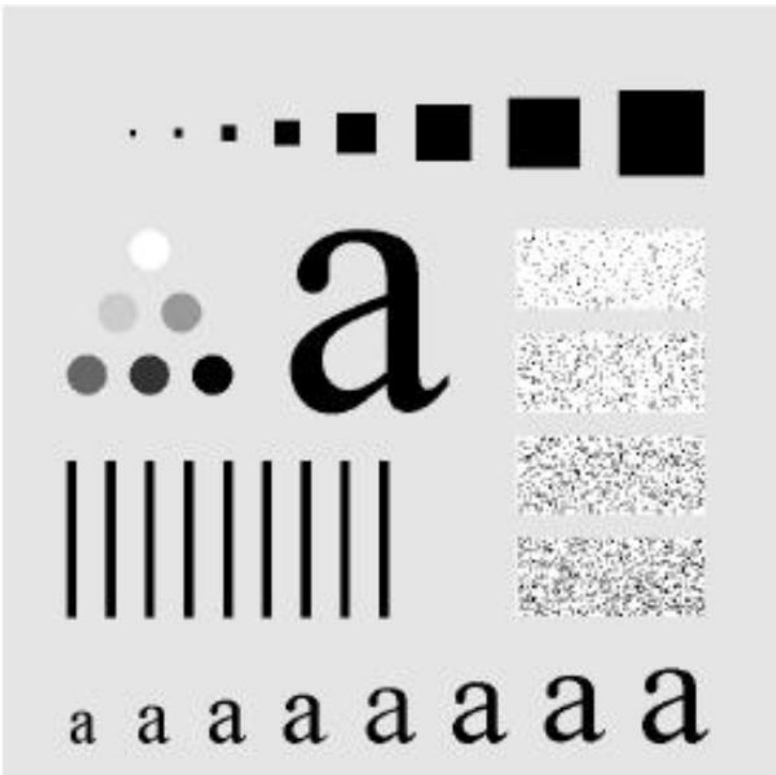
$D_0 = 30$





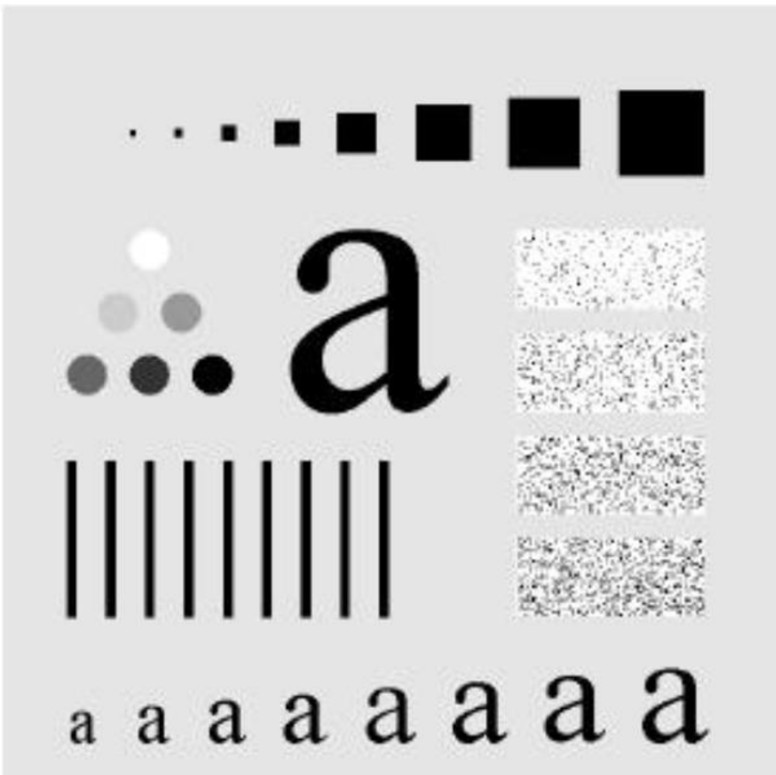
Ideal Highpass Filter (IHPF)

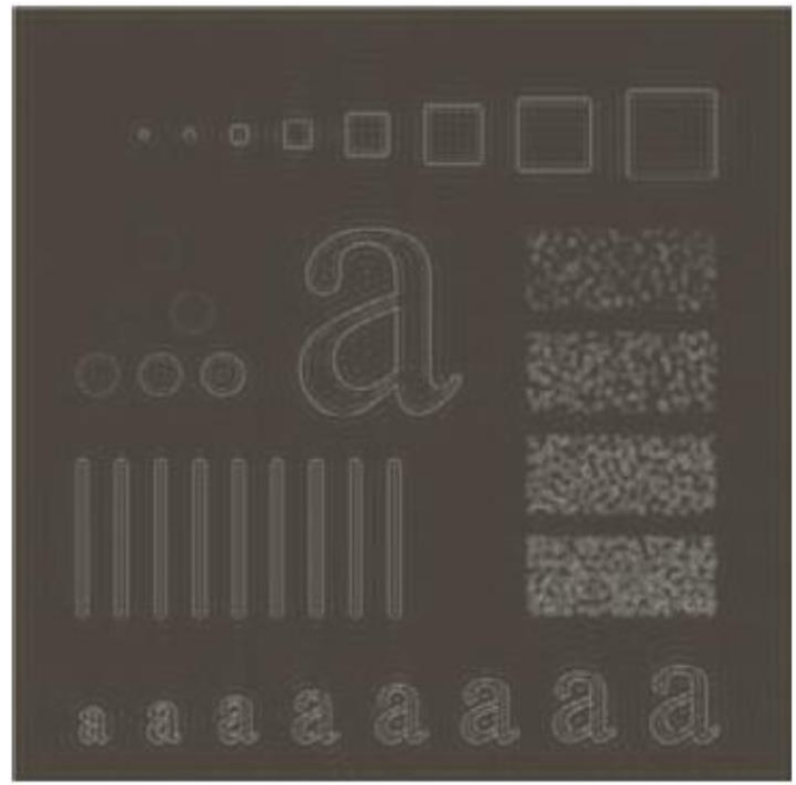
$D_0 = 60$



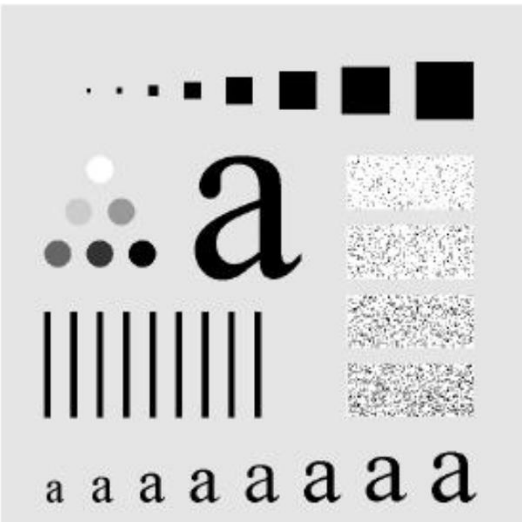
Ideal Highpass Filter (IHPF)

$D_0 = 160$



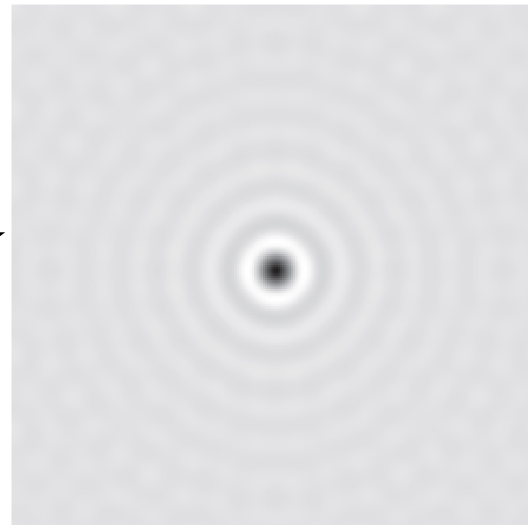


Ideal Highpass Filter (IHPF)



Input image

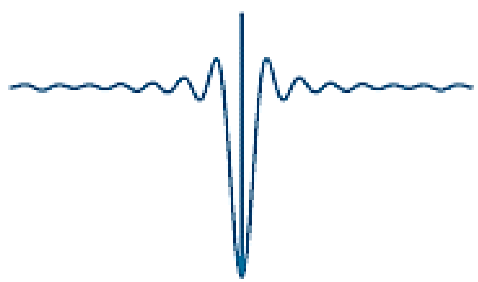
★



Spatial Kernel h(x,y) obtained from a IHPF in frequency domain

=

Filtered image

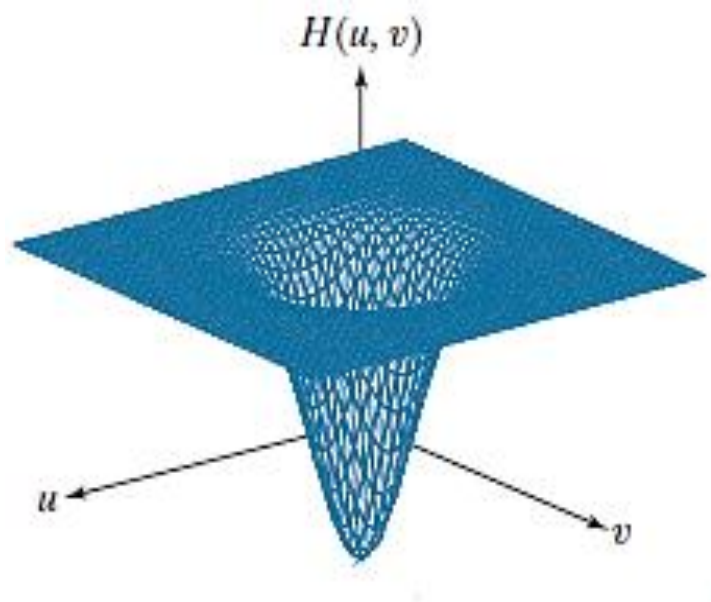


IHPF kernel has the same **ringing property** as that of its parent lowpass kernel

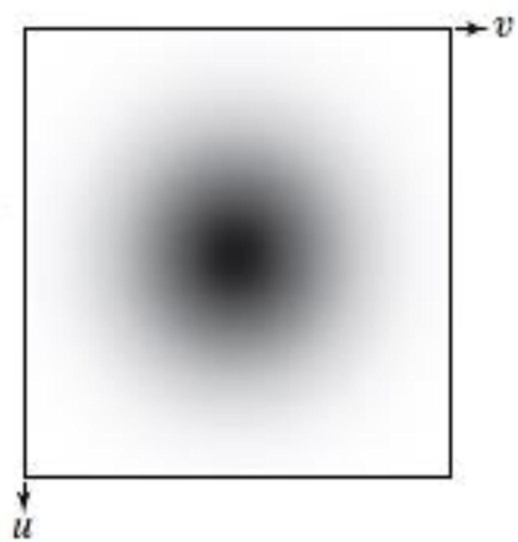
Gaussian Highpass Filter (GHPF)

A 2D *Gaussian highpass filter* with cutoff frequency D_0 is defined as :

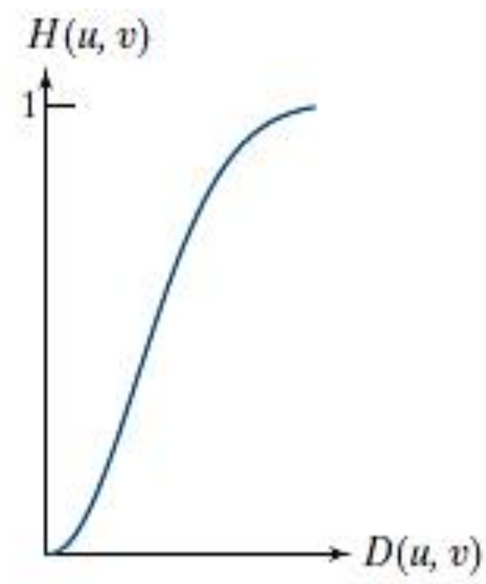
$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$



Perspective Plot



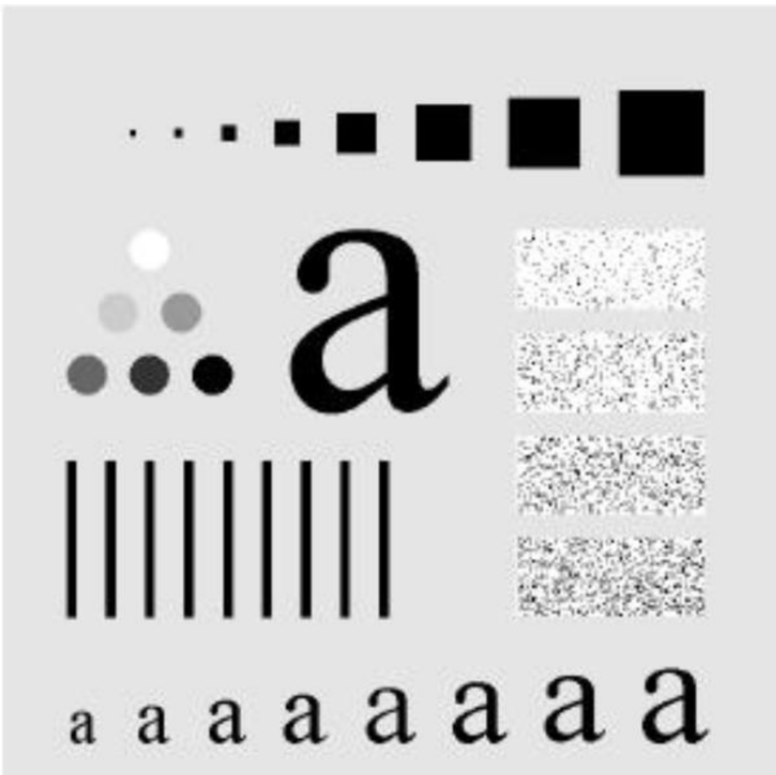
Filter Image



Radial Cross Section

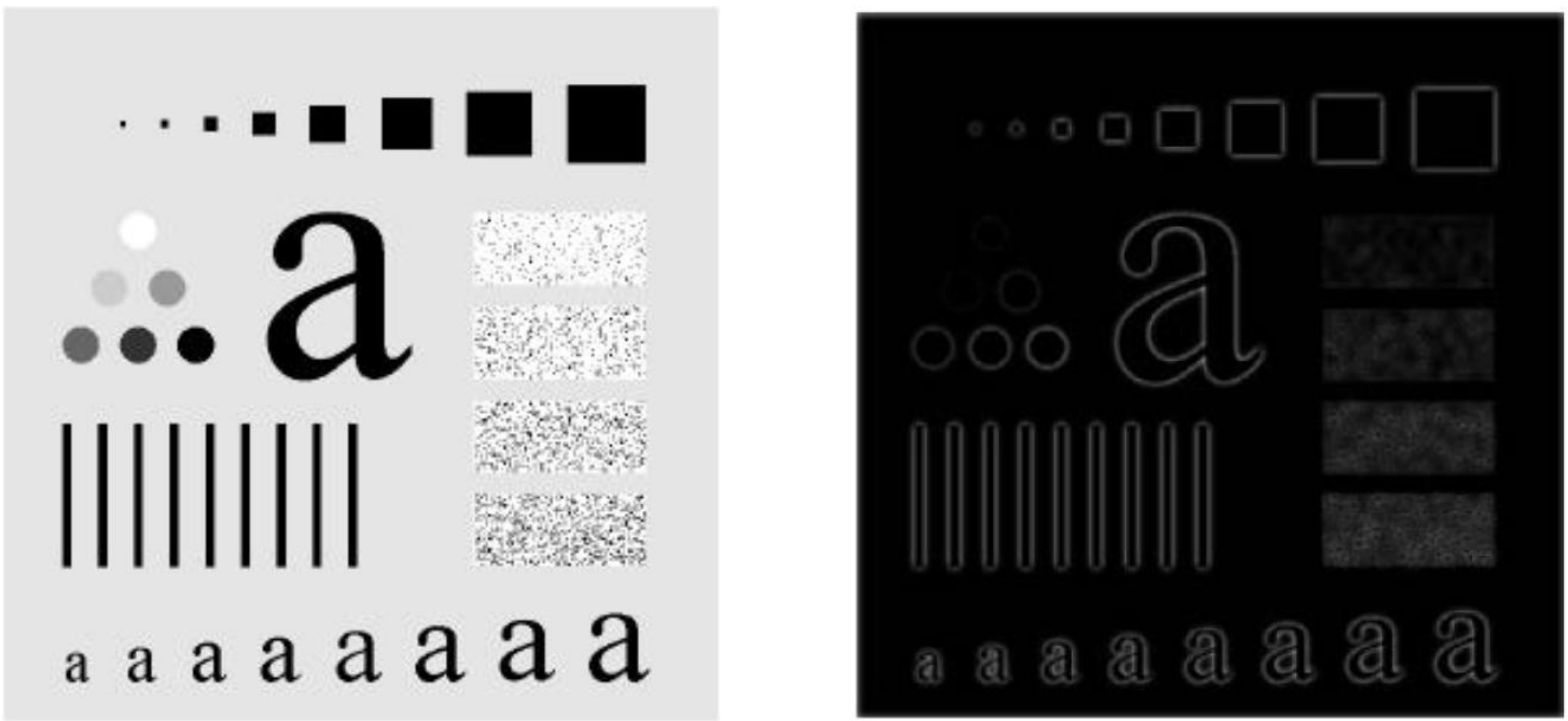
Gaussian Highpass Filter (GHPF)

$D_0 = 30$



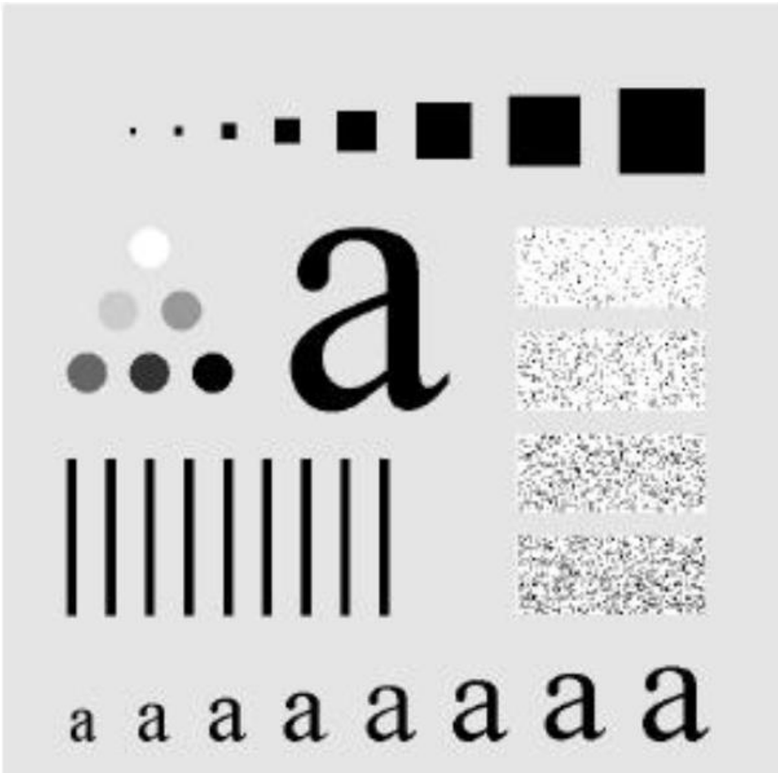
Gaussian Highpass Filter (GHPF)

$D_0 = 60$

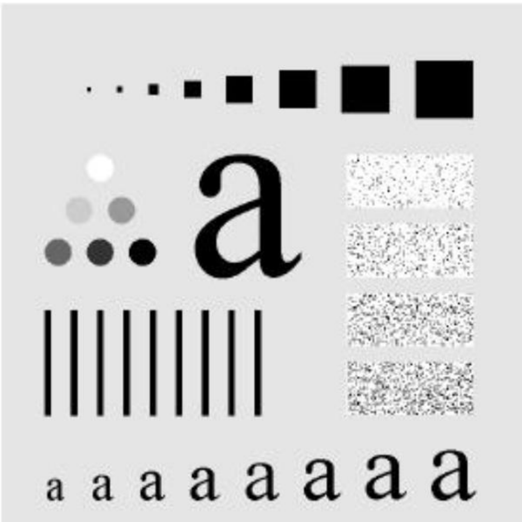


Gaussian Highpass Filter (GHPF)

$D_0 = 160$

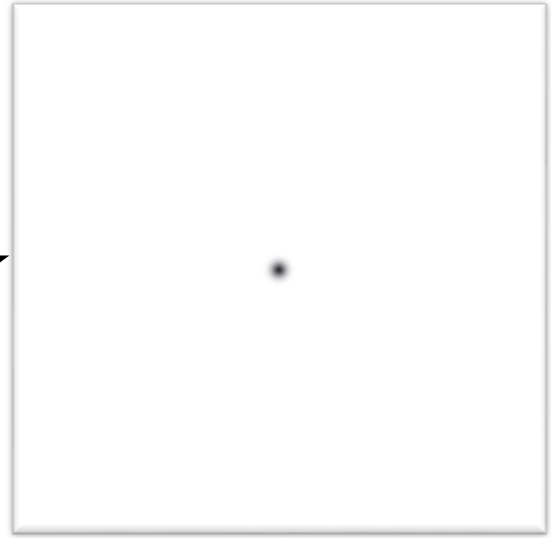


Gaussian Highpass Filter (GHPF)



Input image

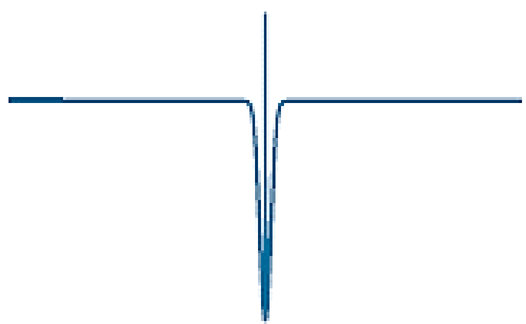
★



Spatial Kernel h(x,y) obtained from a GHPF in frequency domain

=

Filtered image

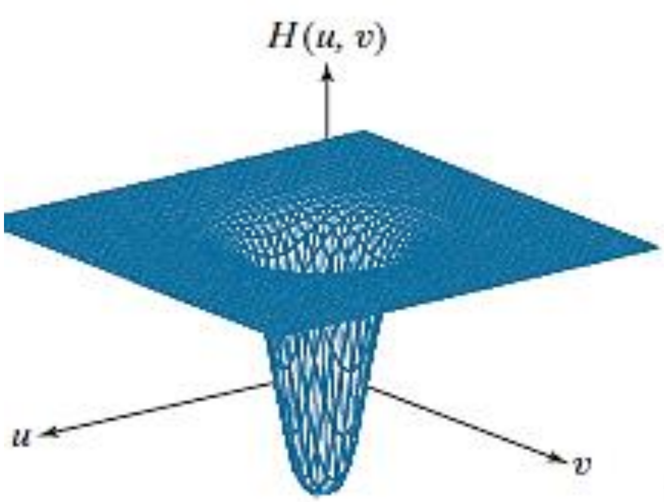


GHPF kernel has no **ringing property**

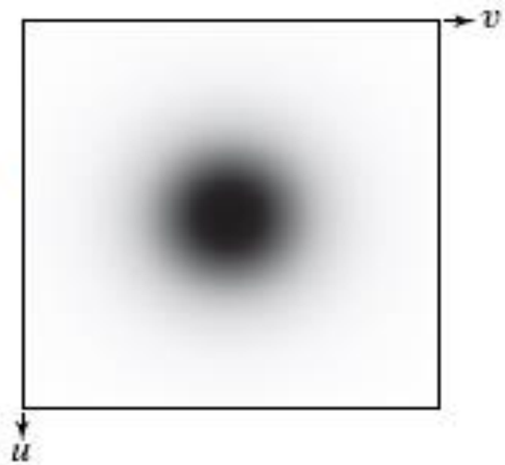
Butterworth Highpass Filter (BHPF)

A 2D *Butterworth highpass filter* of order n and cutoff frequency D_0 is defined as :

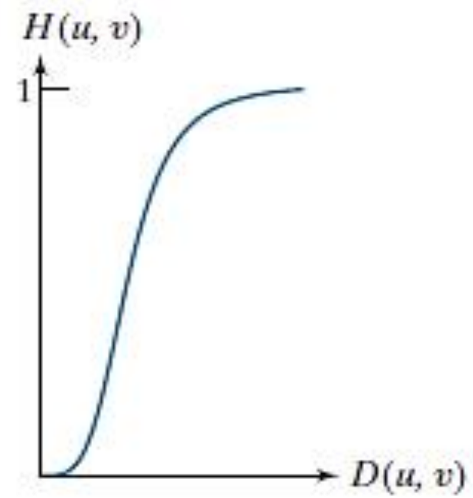
$$H(u,v) = \frac{1}{1 + \left[D_0 / D(u,v)\right]^{2n}}$$



Perspective Plot



Filter Image

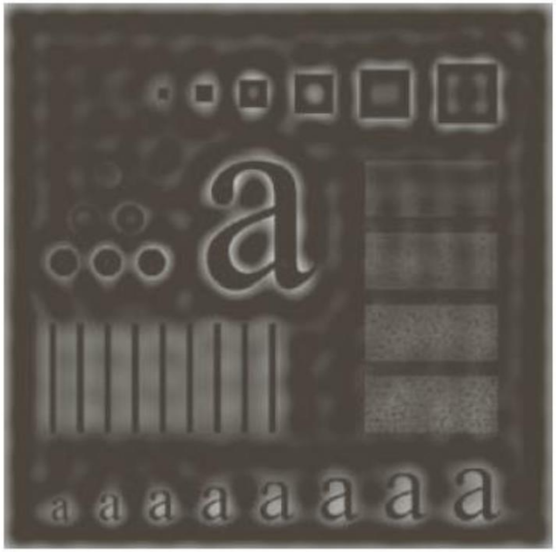


Radial Cross Section

Butterworth Highpass Filter (BHPF)

D_0 = 30, n=2

Input image

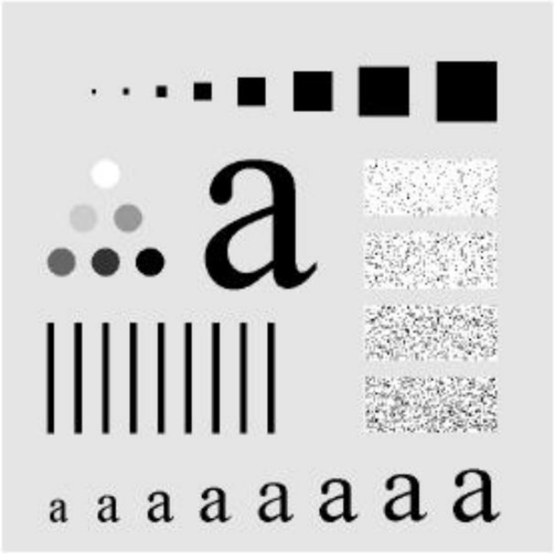


ILPF

BLPF

Butterworth Highpass Filter (BHPF)

D_0 = 60, n=2

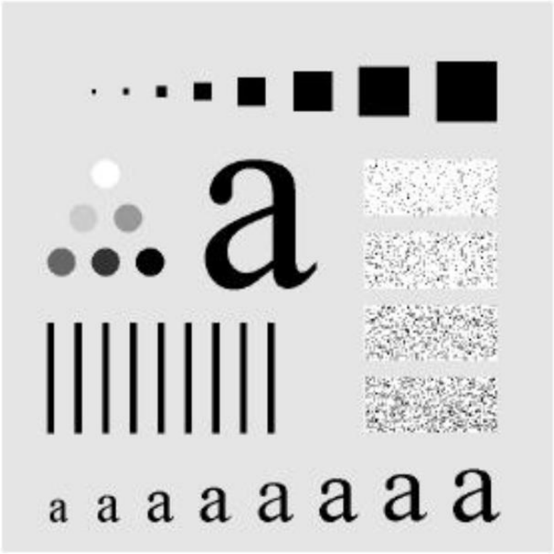


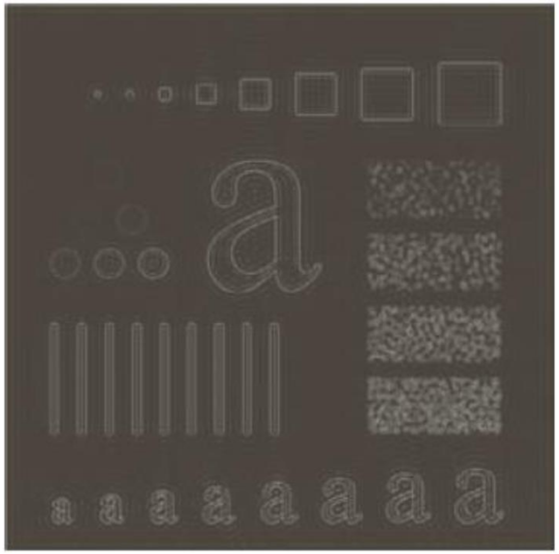
ILPF

BLPF

Butterworth Highpass Filter (BHPF)

D_0 = 160, n=2

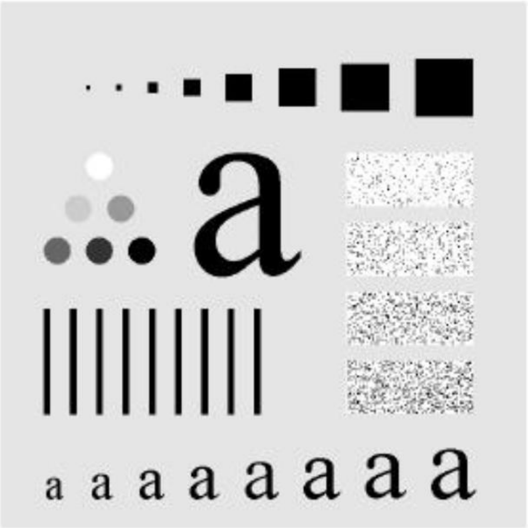




ILPF

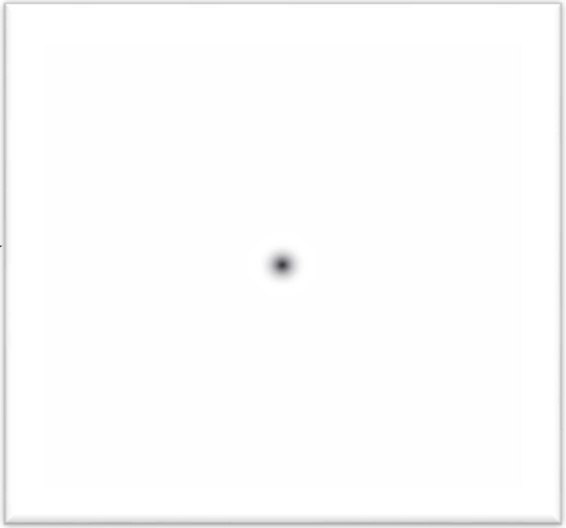
BLPF

Butterworth Highpass Filter (BHPF)



Input image

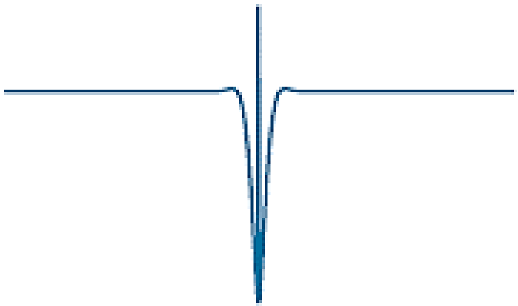
$\star$



Spatial Kernel h(x,y) obtained from a BHPF in frequency domain

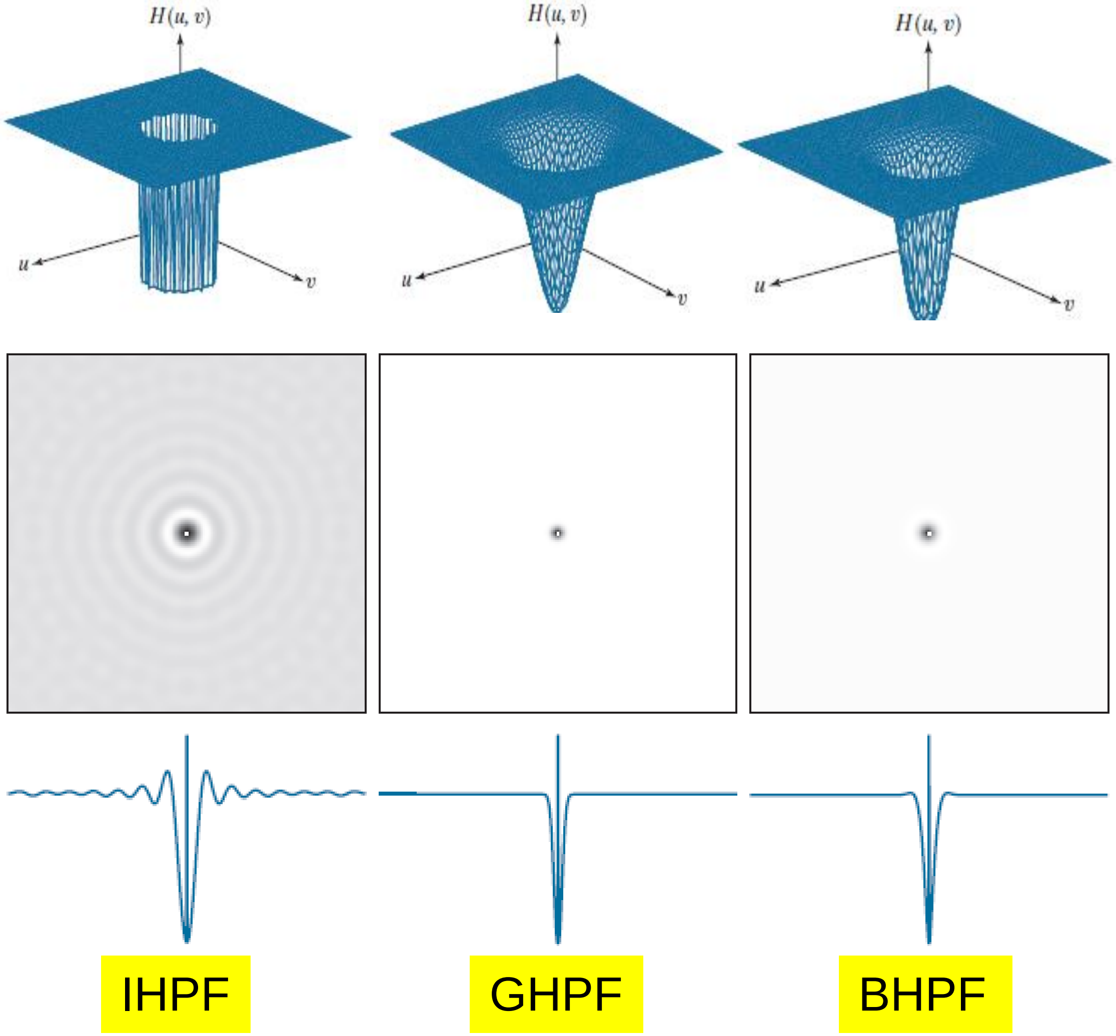
=

Filtered image



BHPF kernel has no **ringing property**

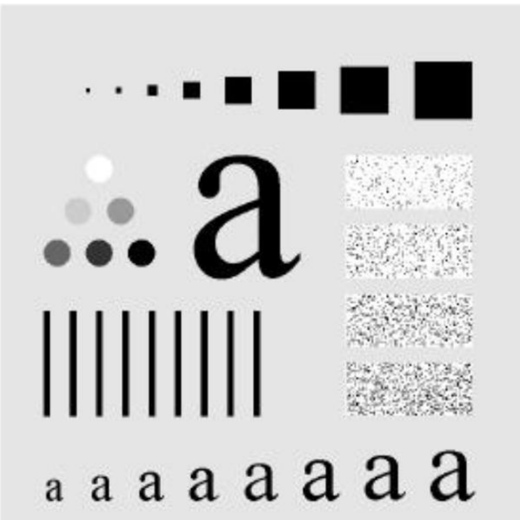
Ringing Artifacts for HighPass Filters



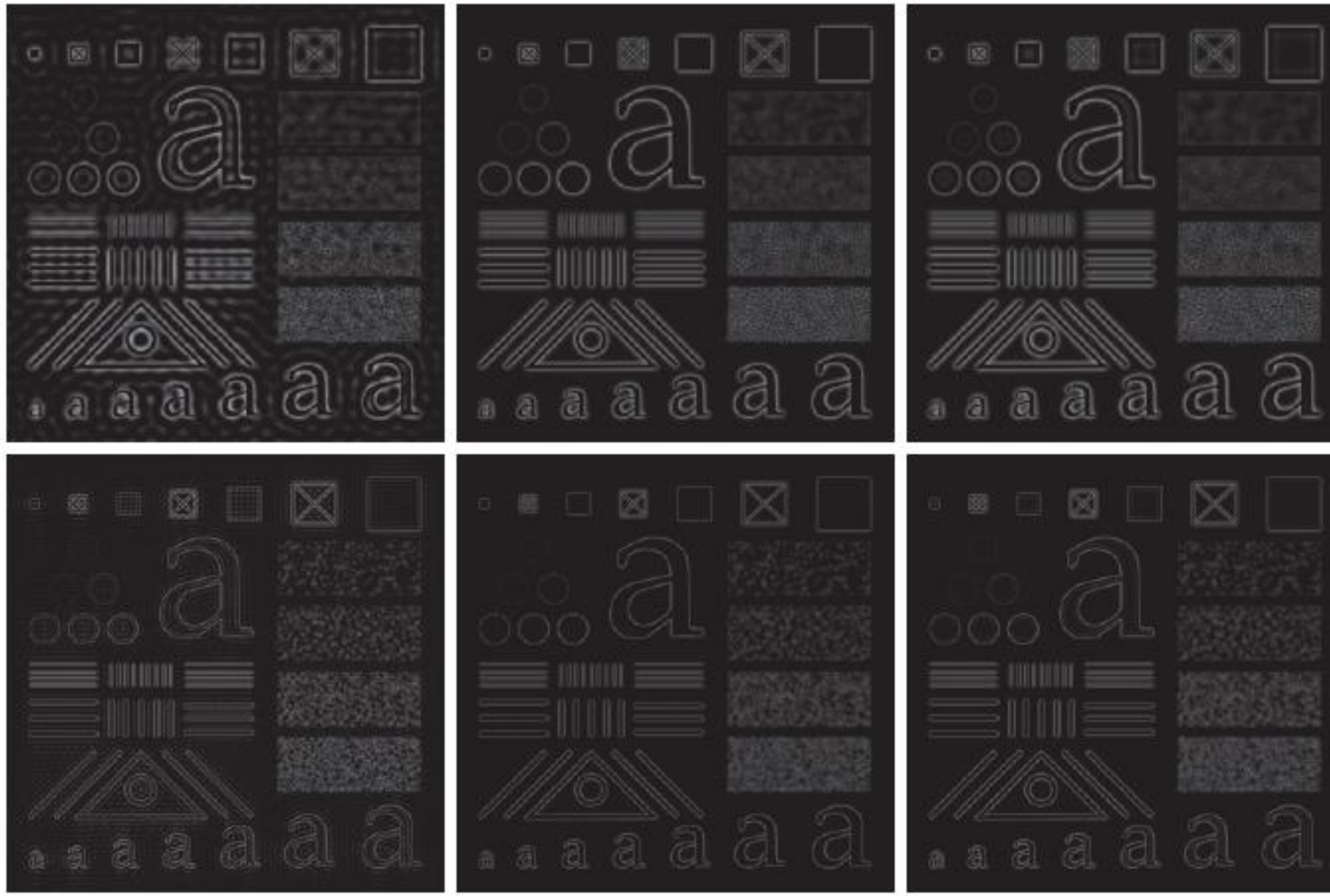
Ideal, Gaussian, and Butterworth highpass spatial kernels obtained from IHPF, GHPF, and BHPF frequency-domain transfer functions.

Highpass Filtering

Comparison



Original Image



First row : Original image filtered with IHPF, GHPF, and BHPF transfer functions using $D0 = 60$ in all cases ($n = 2$ for the BHPF).

Second row : Same sequence, but using $D0 = 160$.

Highpass Filtering Example

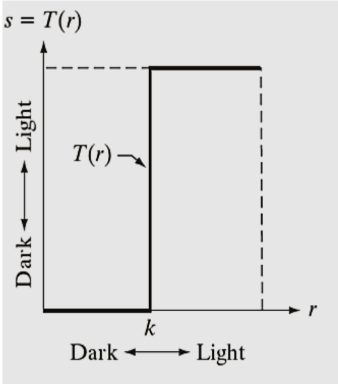
Thumb Print Recognition

Smudged thumbprint

BHPF

(D_0=50, n=4)

BHPF



Result of thresholding

Laplacian in the Frequency Domain

Recall: Laplacian in the Frequency Domain

For a function (image) ***f* (*x*, *y*)** of two variables, it is defined as:

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

Recall that in 1D:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

So, in the x-direction, we have:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

and, in the x-direction, we have:

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

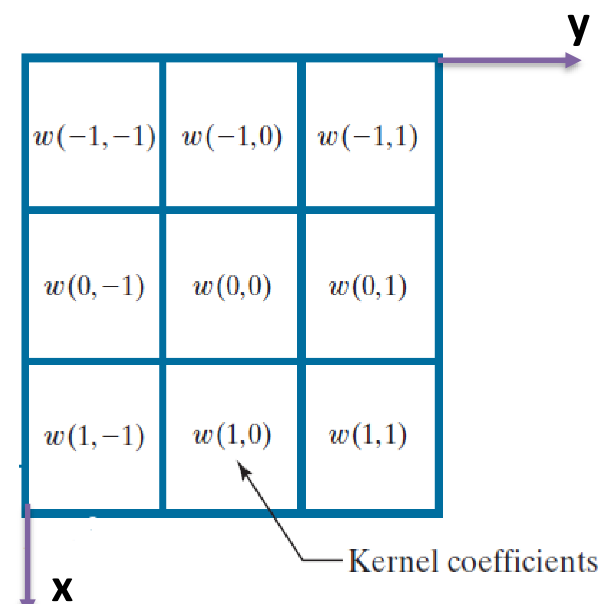
Recall: Laplacian in the Frequency Domain

So, from the preceding two equations, the discrete Laplacian for a function (image) ***f* (*x*, *y*)** of two variables, it is defined as:

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

$$w(x,y) \star f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$w(x,y) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



Recall: Laplacian in the Frequency Domain

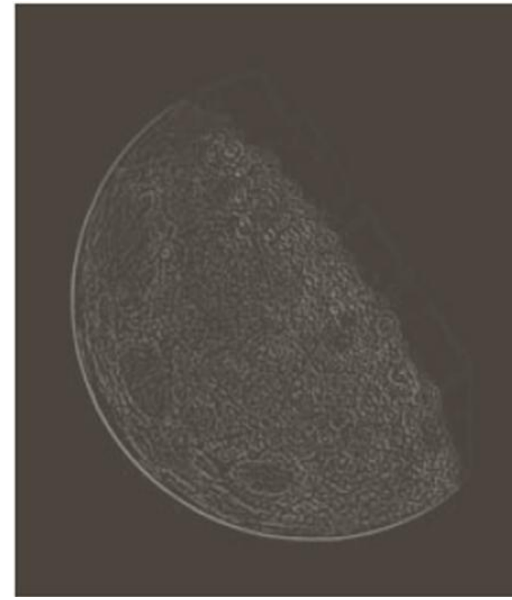
f(x,y)
North pole of the moon

$\star$

0	1	0
1	-4	1
0	1	0

Laplacian kernel

$=$



$\nabla^2 f(x,y)$

$$g(x,y) = f(x,y) - c\left[\nabla^2 f(x,y)\right]$$

Laplacian in the Frequency Domain

- The Laplacian can be implemented in the frequency domain using the filter transfer function :

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

- With respect to the center of the frequency rectangle, it is given as:

$$\begin{aligned} H(u,v) &= -4\pi^2\left[(u - P/2)^2 + (v - Q/2)^2\right] \\ &= -4\pi^2 D^2(u,v) \end{aligned}$$

- The Laplacian image can be obtained by:

$$\nabla^2 f(x,y) = \Im^{-1}\left[H(u,v)F(u,v)\right]$$, where F(u,v) is the DFT of f(x,y)

- Enhancement can then be achieved by:

$$g(x,y) = f(x,y) - c\nabla^2 f(x,y)$$ Here, ***c* = −1** because *H*(u,v) is negative.

Laplacian in the Frequency Domain

- In the frequency domain we can write **Laplacian Equation** in **single line** as:

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

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

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

Laplacian in the Frequency Domain

f(x,y)

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

Laplacian in the Frequency Domain

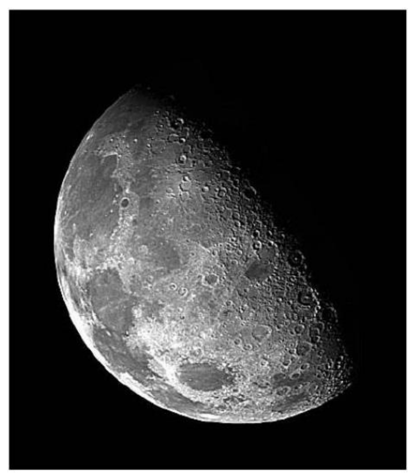
0	1	0
1	-4	1
0	1	0

Image enhanced using the Laplacian in spatial domain

1	1	1
1	-8	1
1	1	1

Image enhanced using the Laplacian in spatial domain

f(x,y)



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

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

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

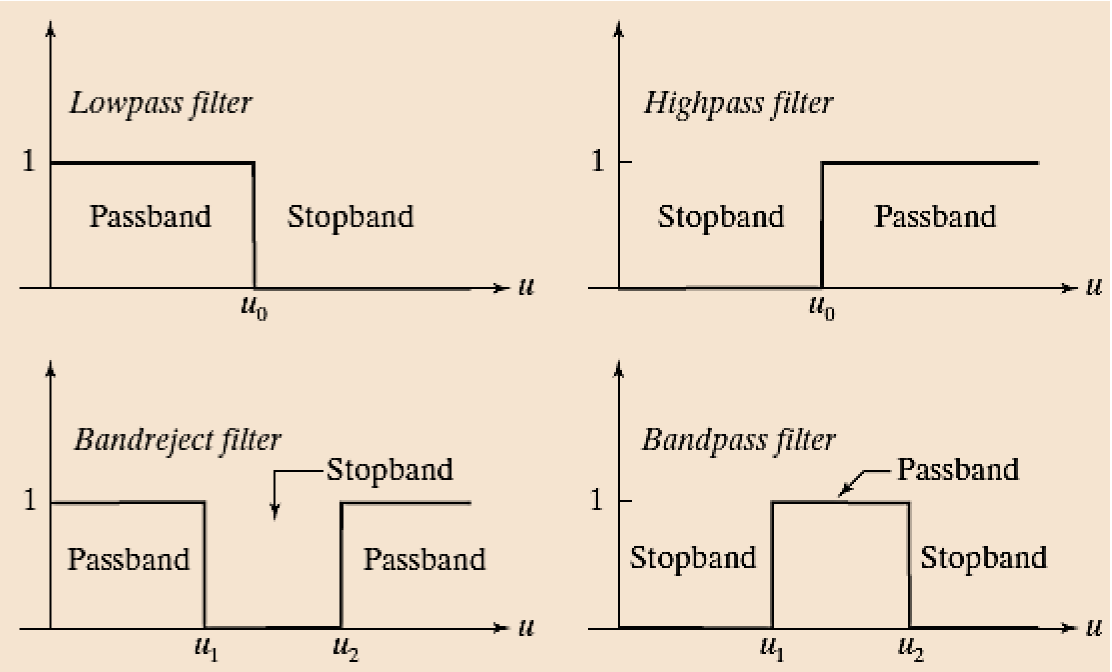
Image enhanced using the Laplacian in frequency domain

Selective Filtering

Selective Filtering

1. Used to process *specific bands* of frequencies (**Band filters**).
   - Bandreject filter
   - Bandpass filter

2. Used to process *small regions* of the frequency rectangle (**Notch filters**).
   - Notch reject filter
   - Notch pass filter

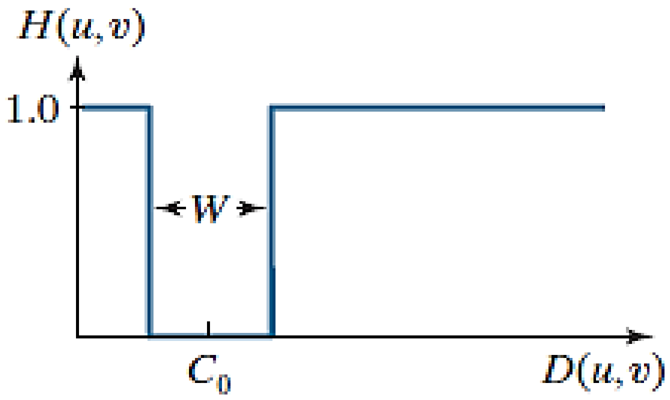


Bandreject and Bandpass filters

- Bandpass and bandreject filter transfer functions in the frequency domain can be constructed by **combining** *lowpass* and *highpass filter* transfer functions.

- *Lowpass filter* transfer functions are the basis for forming *highpass*, *bandreject*, and *bandpass* filter functions.

Ideal Bandreject Filter (IBRF)



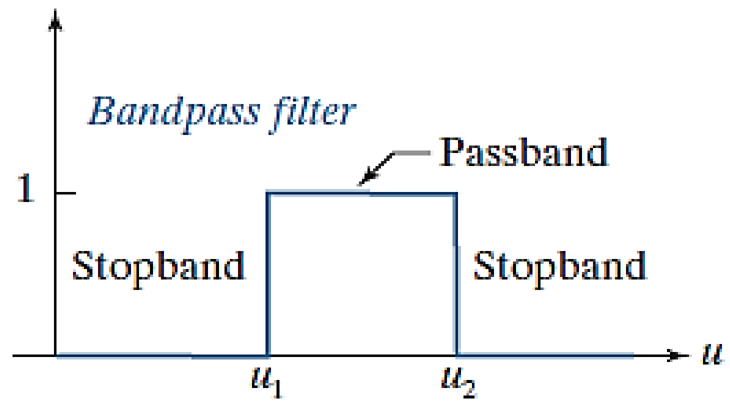
$$H(u,v)=\begin{cases}0 & \text{if } C_0-\dfrac{W}{2}\le D(u,v)\le C_0+\dfrac{W}{2}\\ 1 & \text{otherwise}\end{cases}$$

An **IBRF** transfer function consisting of an ILPF and an IHPF function with different cutoff frequencies.

- ***W*** is the width of band
- $\boldsymbol{C_0}$ is the center of band

Ideal Bandpass Filter (IBPF)

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



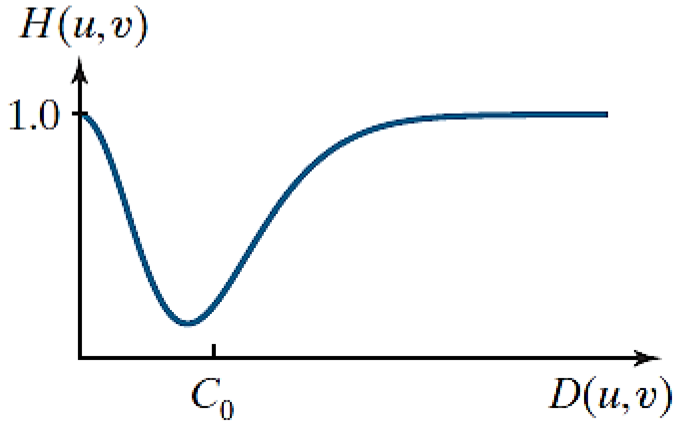
The key requirements of a bandpass transfer function are:

1) the values of the function must be in the range **[0, 1]**.

2) the value of the function must be zero at a distance D_0 from the origin (center) of the function.

3) we must be able to specify a value for W. Clearly, the IBRF function just developed satisfies these requirements.

Gaussian Bandreject Filter (GBRF)



The **two problems** in GBRF are:

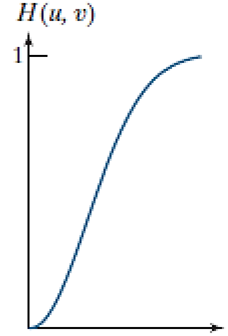
1. We have no direct control over ***W***.
2. The value of ***H*(u,v)** is not **0** at C_0.

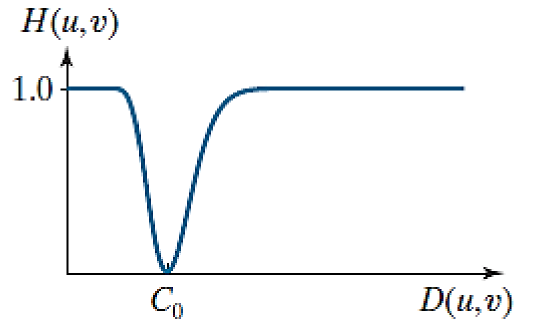
A bandpass function formed as the sum of lowpass and highpass Gaussian functions with different cutoff points

Solution… :

- Modify the expression for the Gaussian highpass transfer function by changing the point at which ***H*(u,v) = 0** from ***D*(u,v) = 0** to ***D*(u,v) = C_0**

$$H(u,v) = 1 - e^{-\left[\frac{D^2(u,v) - C_0^2}{D(u,v)W}\right]^2}$$





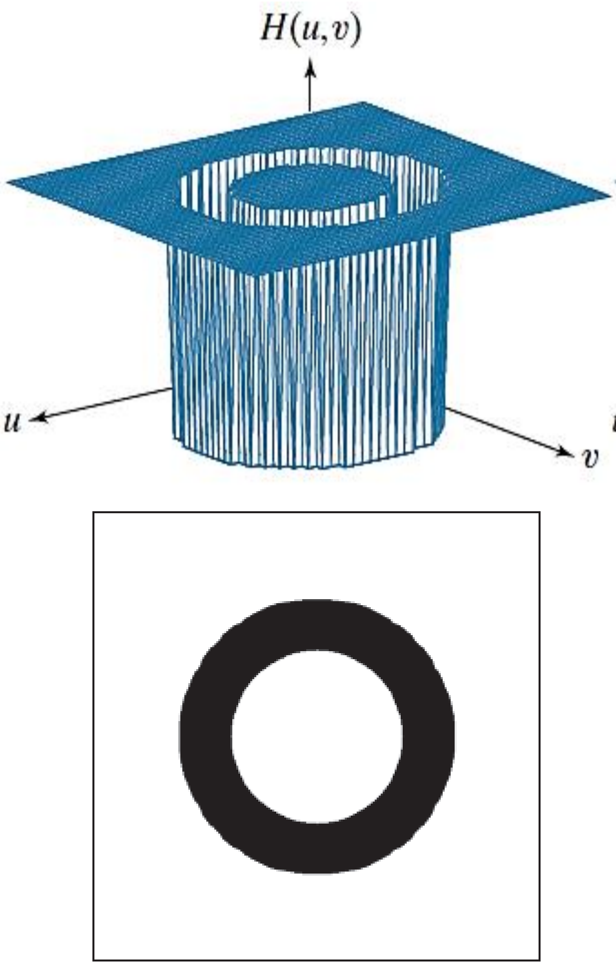
Butterworth Bandreject Filter (BBRF)

Same analysis as in for GBRF will lead us to the following BBRF transfer function:

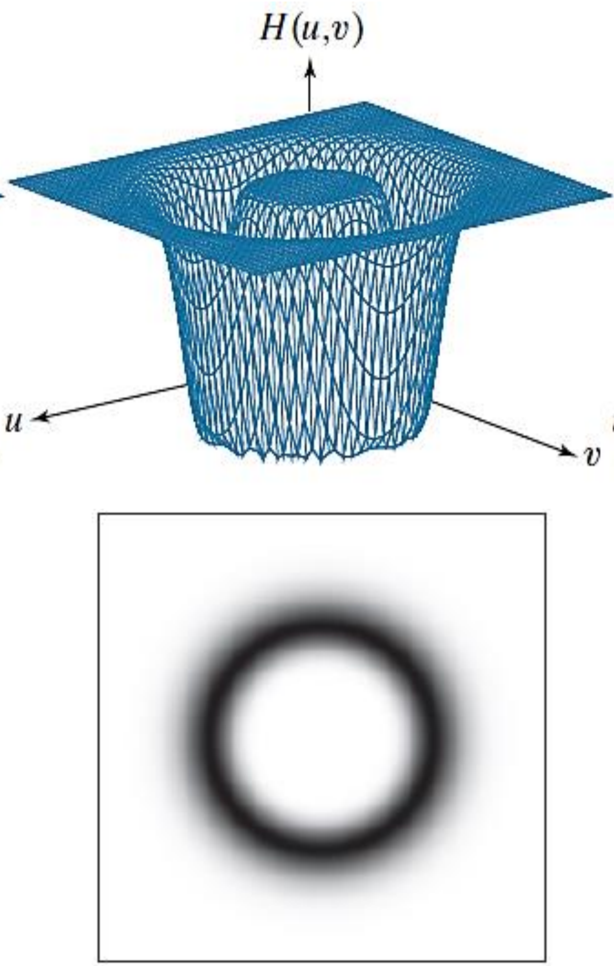
$$H(u,v) = \frac{1}{1 + \left[\dfrac{D(u,v)W}{D^2(u,v) - C_0^2} \right]^{2n}}$$

n – Order of the Butterworth filter

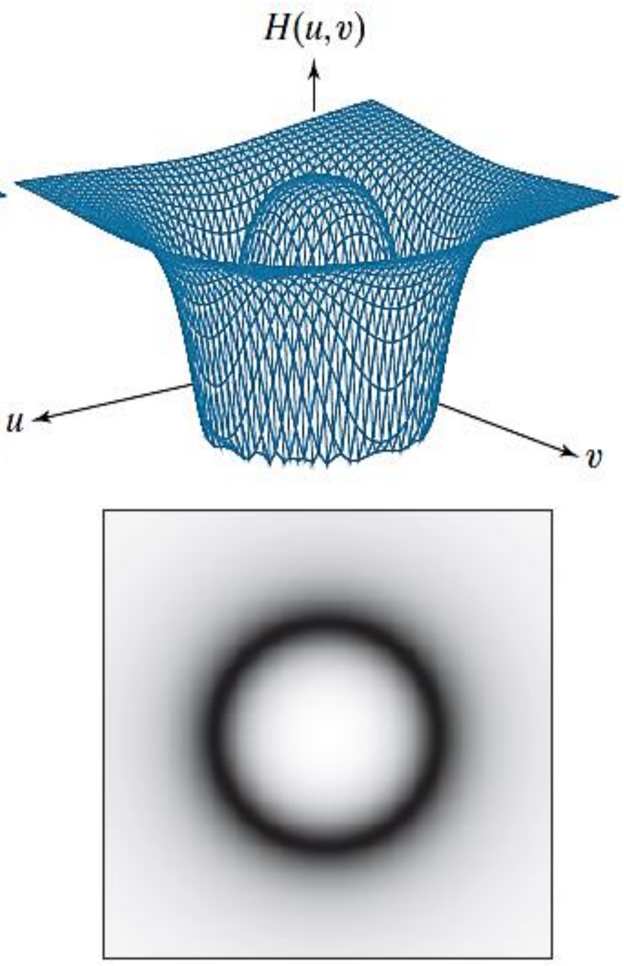
Perspective Plots of Bandreject Transfer Functions



IBRF



Modified GBRF



Modified BBRF

Notch Bandreject Filter (NBRF)

- Notch filters are the most useful of the band reject filters.

- A Notch filter reject/pass signals in a **specific frequency band** called the stop band frequency range and *pass the signals above and below this band*.

- These are also **Zero-phase-shift filters** which are symmetric about their origin.

$$h(u_0, v_0) = h(-u_0, -v_0)$$

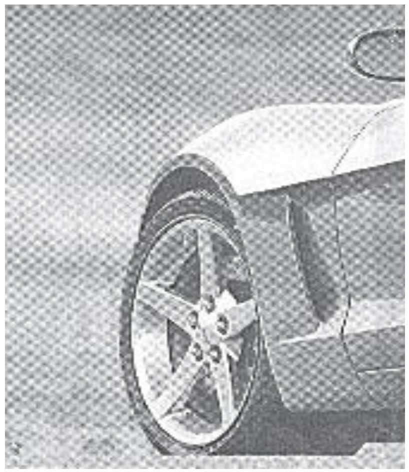
- *Notch reject* filter transfer functions are constructed as **products of highpass filter transfer functions** *whose centers have been translated to the centers of the notches*.

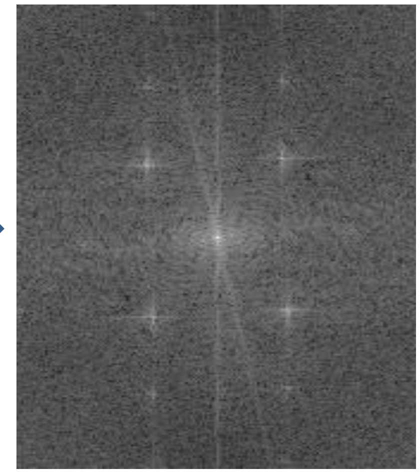
- **A Notch filter contain notch pairs.**

Notch Filtering Example

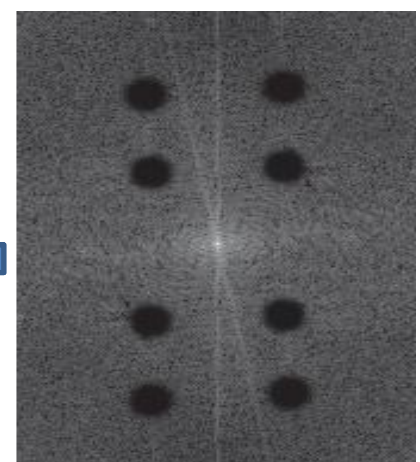
removing moiré patterns from digitized printed media images

Newspaper image showing a moiré pattern
f(x,y)

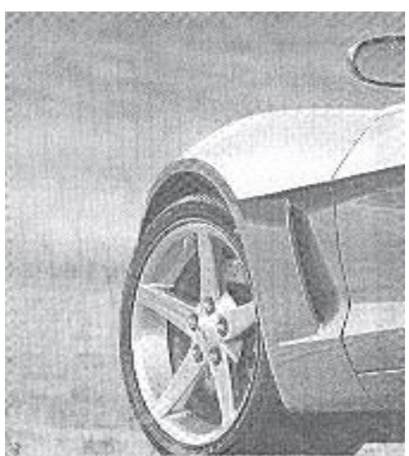




F(u,v)



F(u,v) multiplied by a pair of Butterworth notch reject filter transfer function

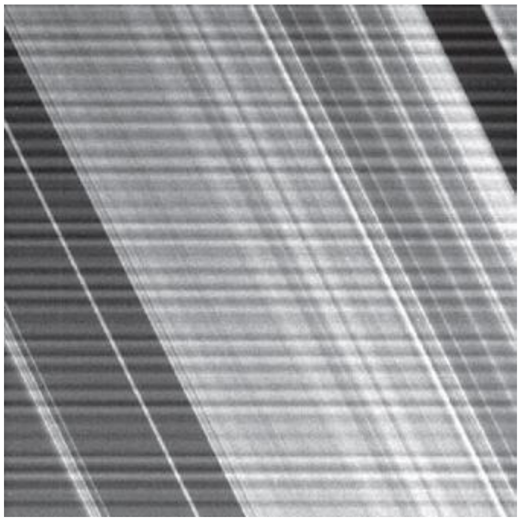


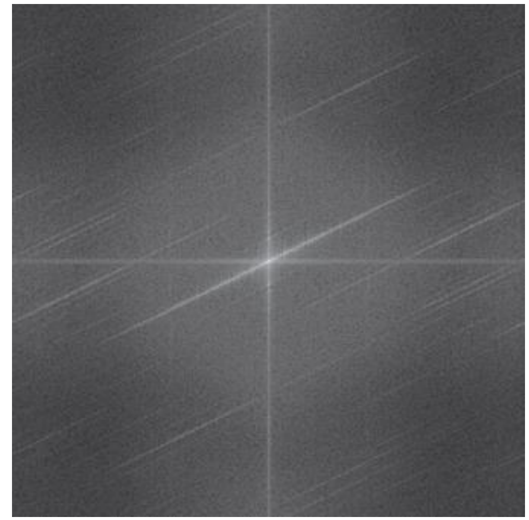
Filtered image

Notch Filtering Example

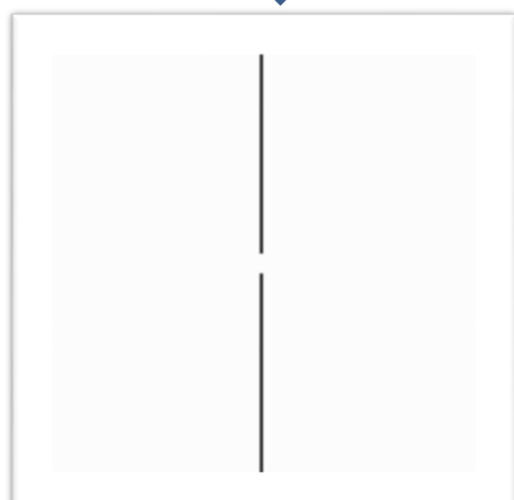
removing periodic interference

Image of Saturn rings showing periodic interference **f(x,y)**

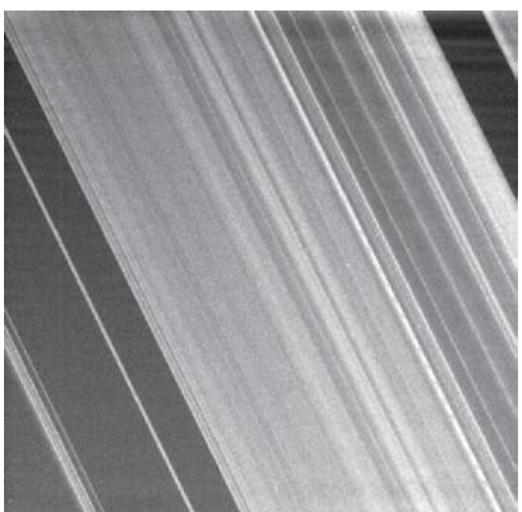




F(u,v)



A vertical notch reject filter transfer function

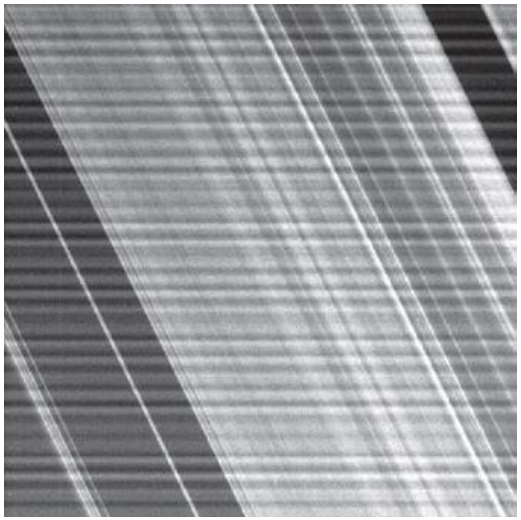


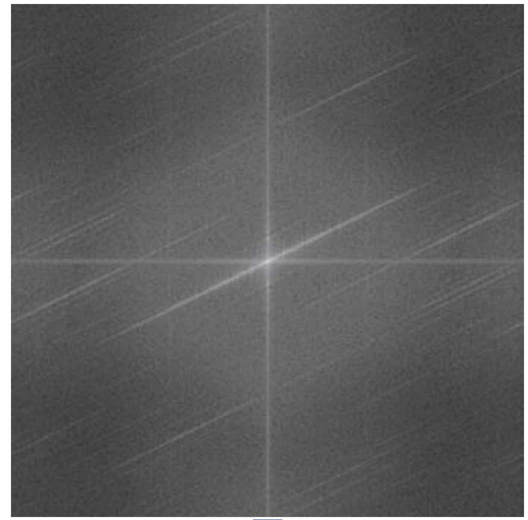
Filtered image

Notch Filtering Example

obtaining image of interference pattern

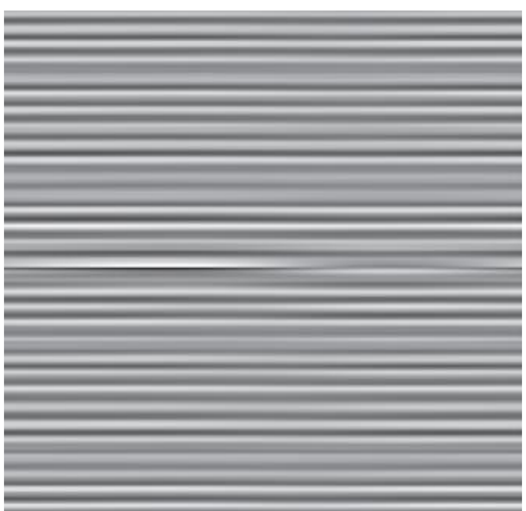
Image of Saturn rings showing periodic interference **f(x,y)**





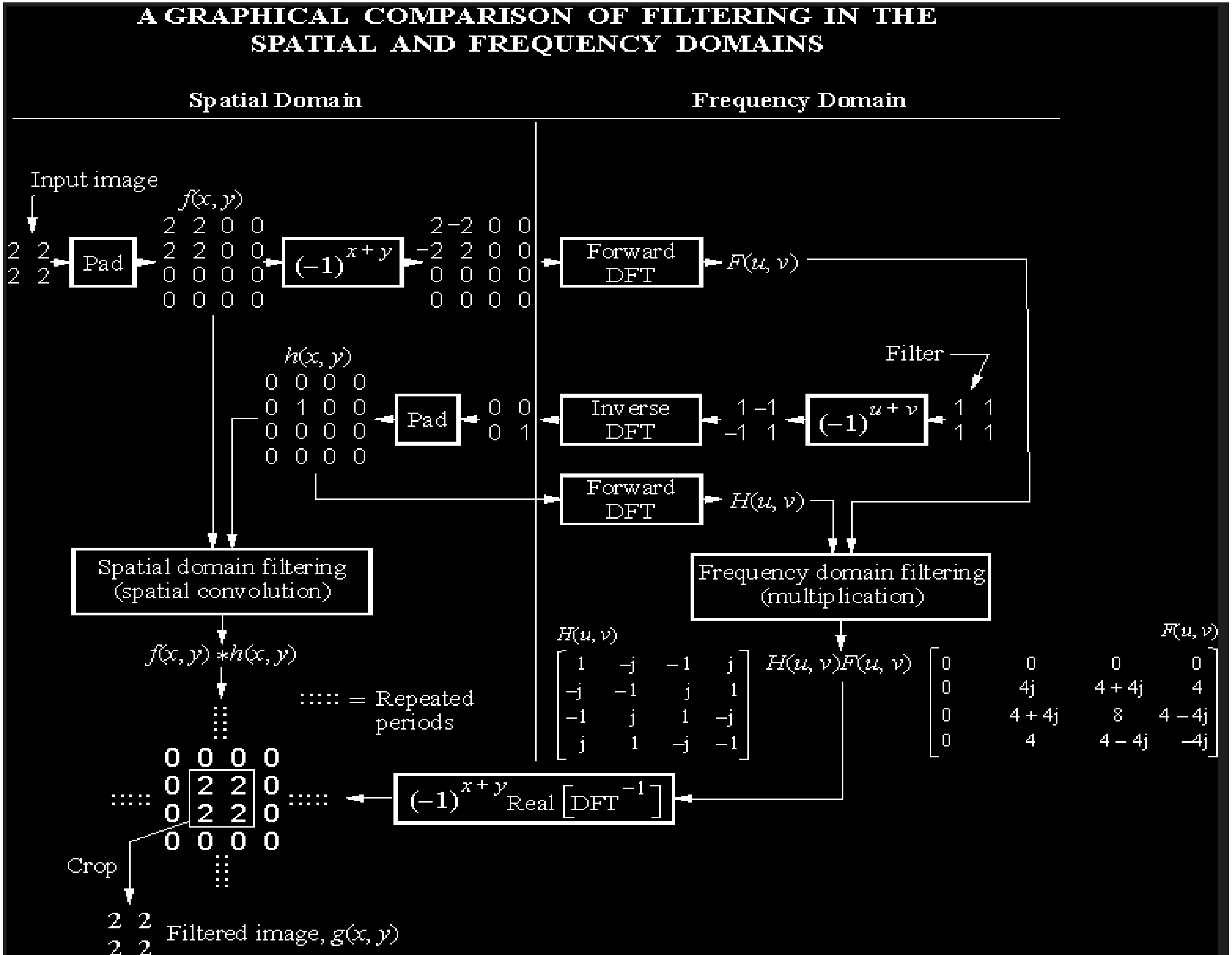
F(u,v)

A vertical notch pass filter transfer function



Interference pattern obtained by computing IDFT of filter result

Graphical illustration of filtering in the spatial and frequency domains



Next Lecture

- The 2-D DFT - Some Observations
- Separability of Fourier Transform
- IDFT in terms of DFT
- Fast Fourier Transform (FFT)
  - FFT Process in 1-D
  - Special Properties of W_M
  - FFT even-odd approach
  - FFT "Butterfly" Method
  - FFT – time complexity
  - Can we speed it up??
  - FFT Algorithm