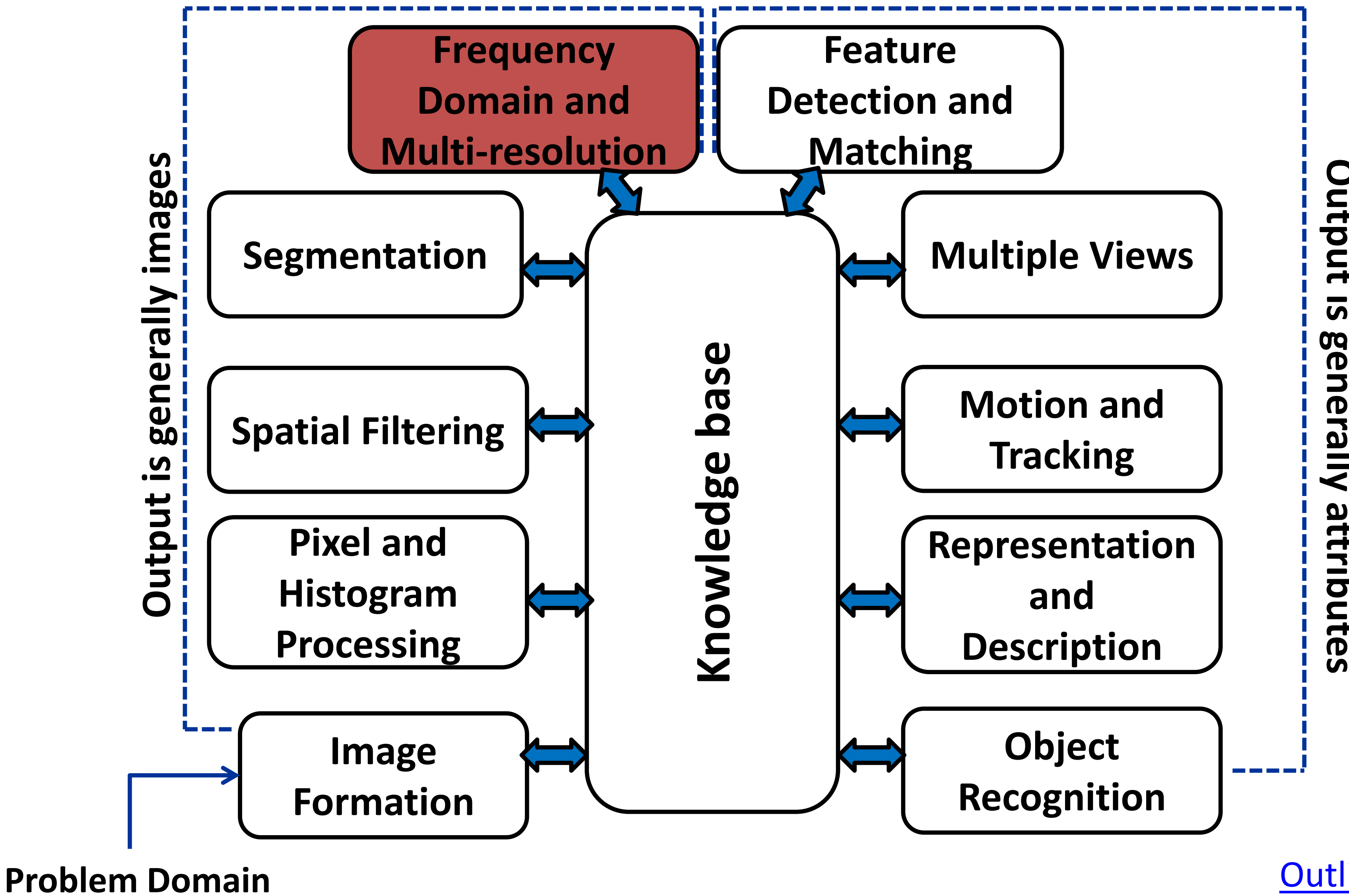
CSC 425 Digital Image Processing

Dr. Maryam Al-Berry
maryam_nabil@cis.asu.edu.eg

Course Outline



Outline

Contents

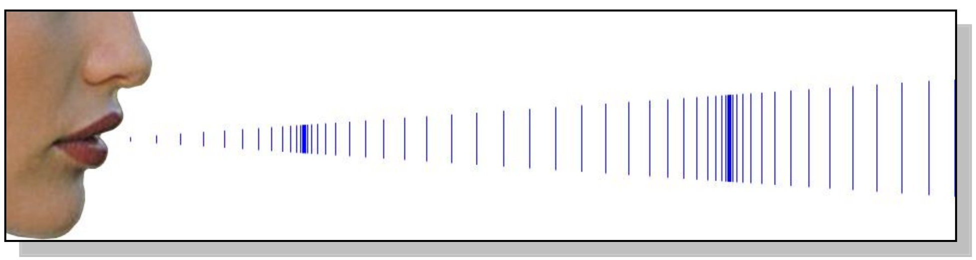
- Signals in Frequency Domain
- Fourier Transform (FT)
- Applications of FT

Signals in Frequency Domain

A Signal

- **A measurable phenomenon that changes over time or throughout space.**

sound



code

01101000101101110110010110001

image

Signals in Frequency Domain – (cont.)

Space-Time vs. Frequency Representation

- **Space/time representation:** a graph of the measurements with respect to a point in time and/or positions in space.
- Signals undulate (otherwise they'd contain no information).
- **Frequency-domain representation:** an exact description of a signal in terms of its undulations.

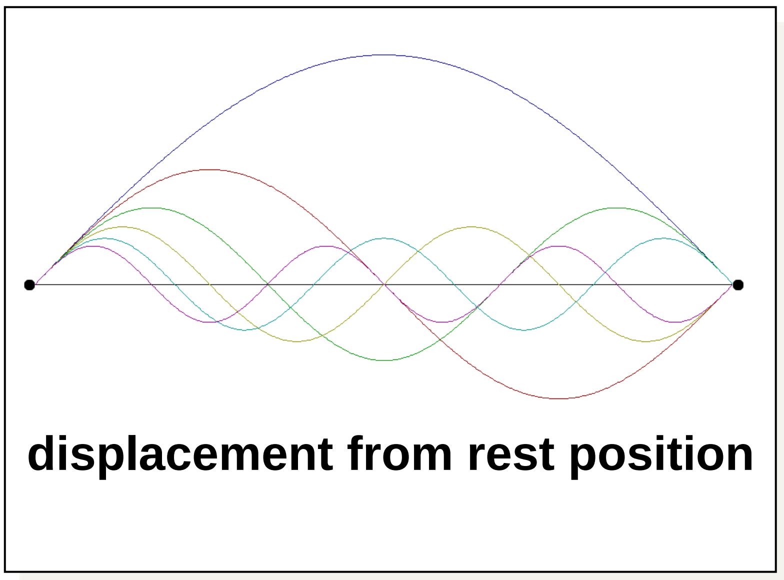
Signals in Frequency Domain – (cont.)

Example: Sound Waves

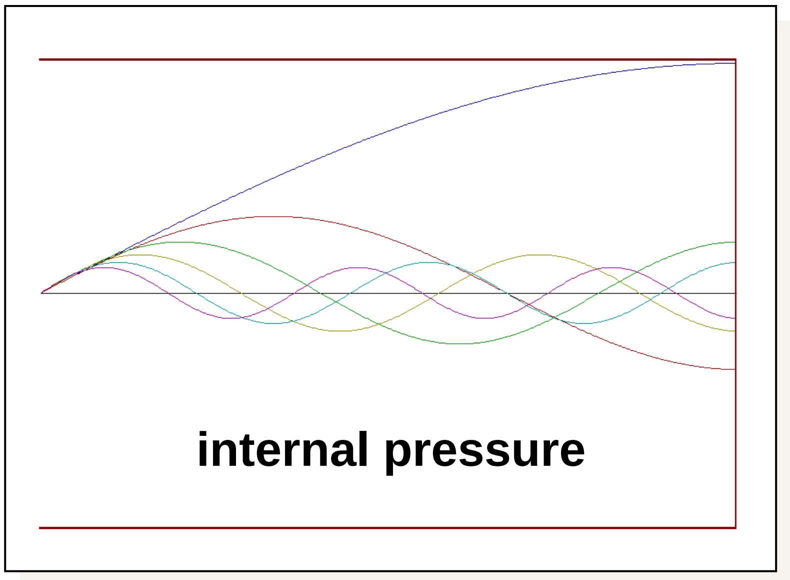
- The mechanical vibrations of an object in an atmosphere.
- The surface of the object undulates causing compressions and rarefactions in the air which propagate through the air away from the surface.
- An object vibrates with different *modes*.
- A mode is a vibratory pattern with a distinctive shape — part of the object surface moves out while another part moves in — a *standing wave*.

Signals in Frequency Domain – (cont.)

Example: Sound Waves



string modes

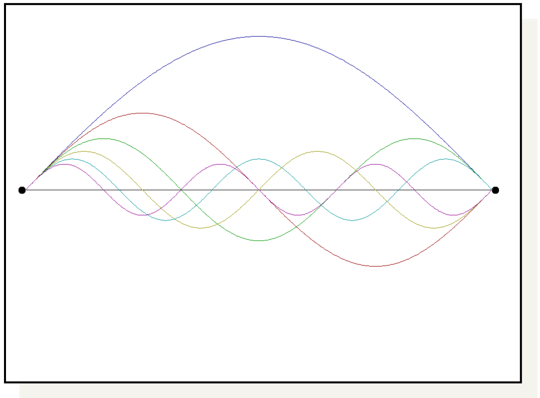


pipe modes

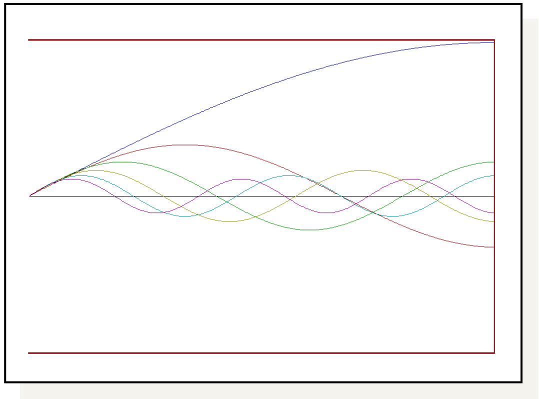
Signals in Frequency Domain – (cont.)

Example: Sound Waves

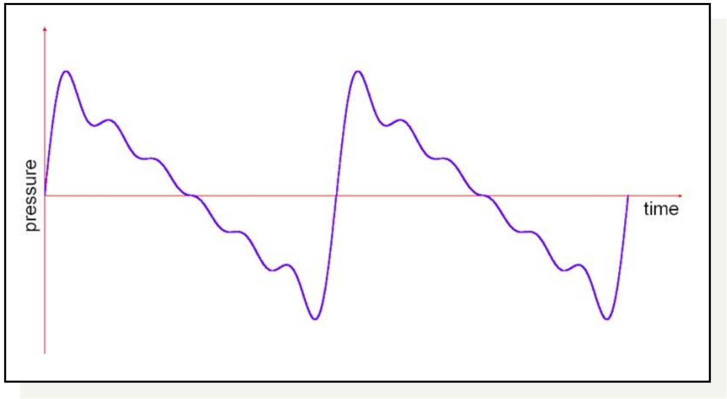
Emerge from the superposition of the modes

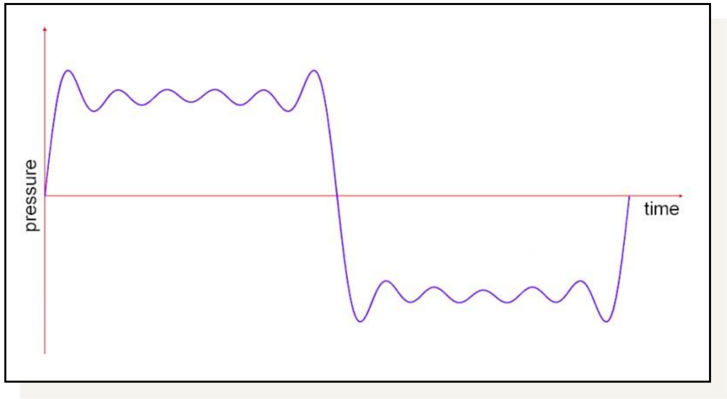


string sound



pipe sound

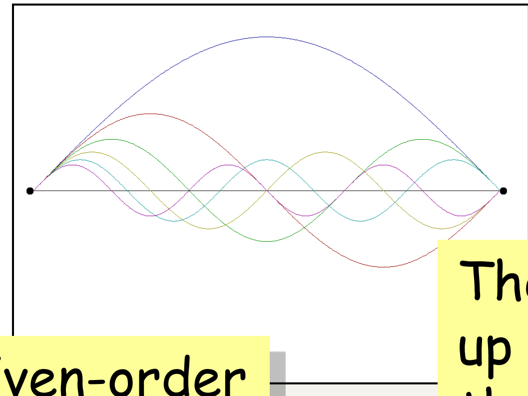




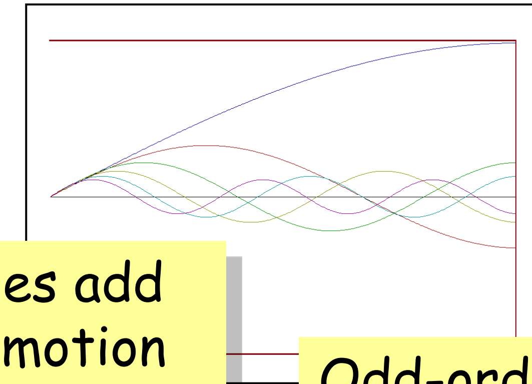
Signals in Frequency Domain – (cont.)

Example: Sound Waves

Emerge from the superposition of the modes



string so

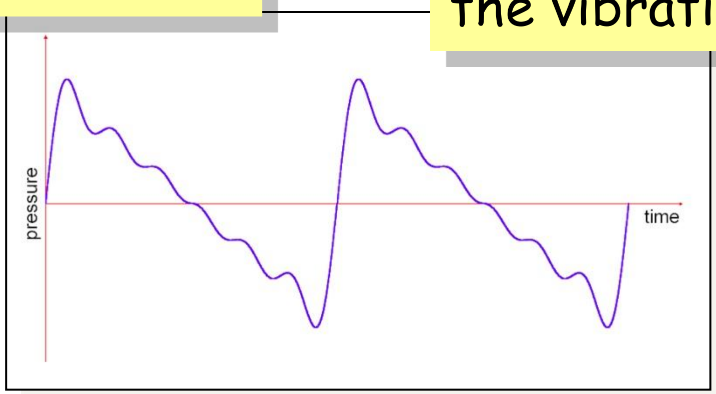


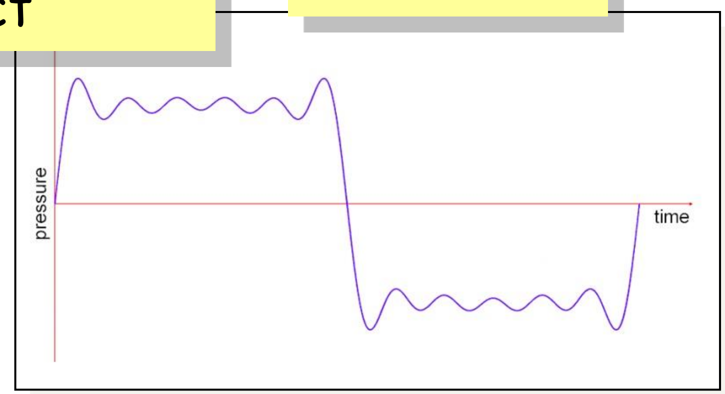
pipe sound

The vibratory modes add up to one complex motion that pushes the air around the vibrating object

Even-order harmonics

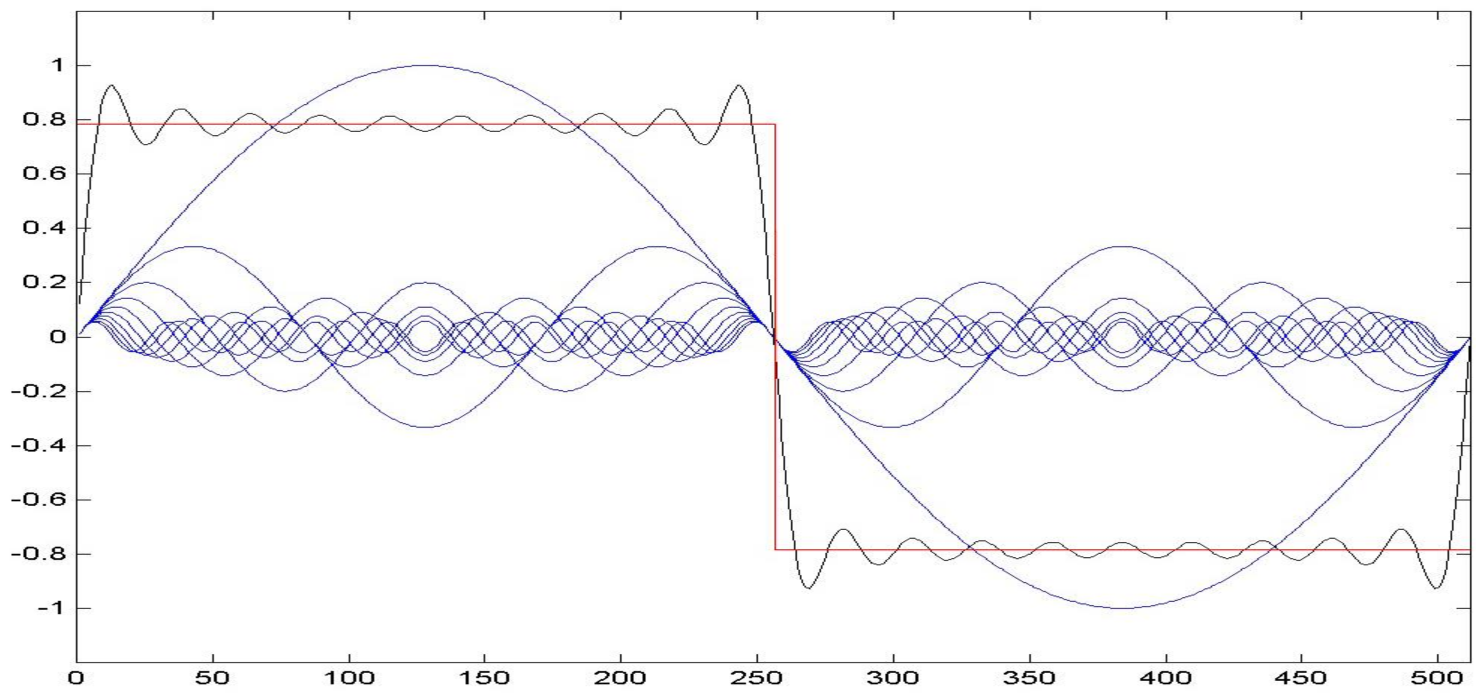
Odd-order harmonics



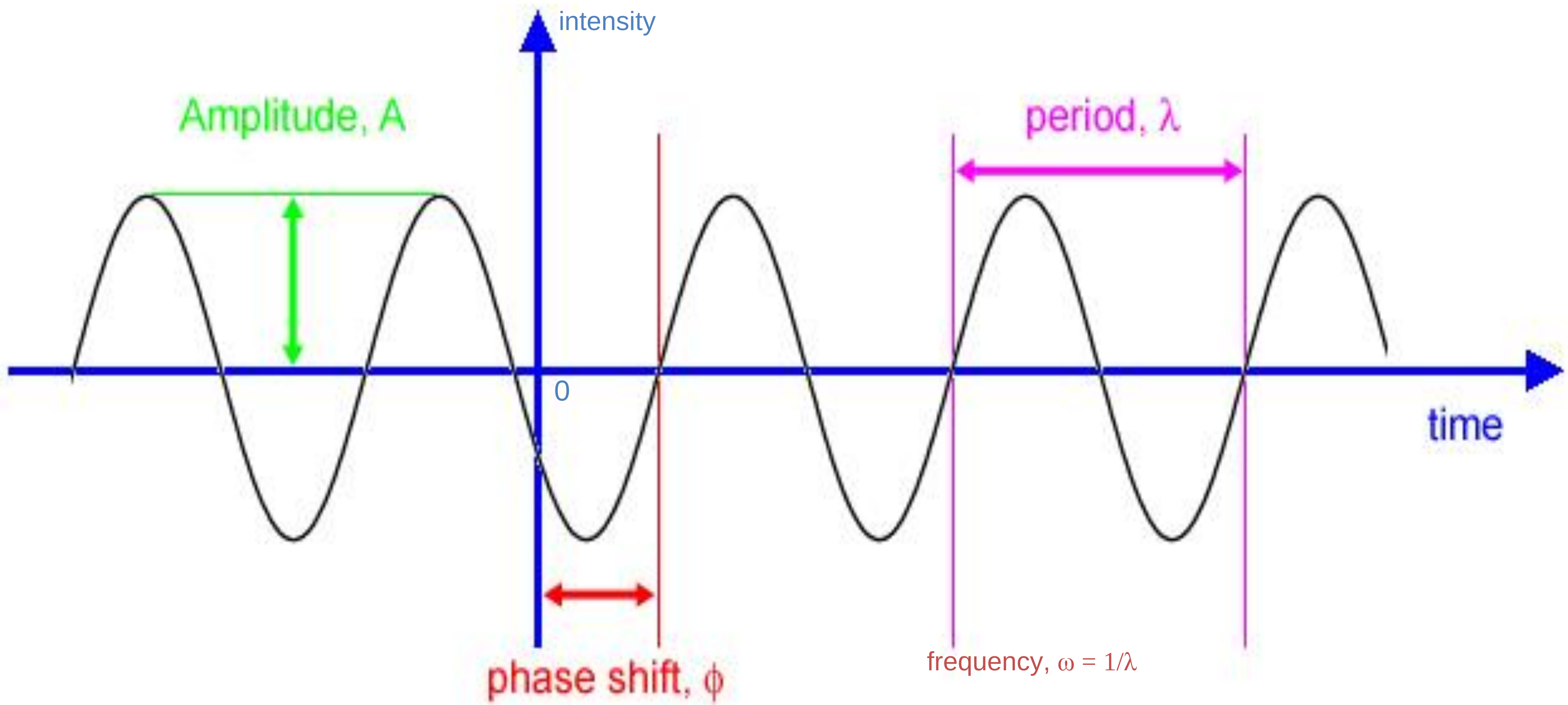


Signals in Frequency Domain – (cont.)

- The (blue) modes sum to the rippling square wave (black).
- As the number of modes in the sum becomes large, it approaches a square wave (red).
- Any real signal has a frequency-domain representation.



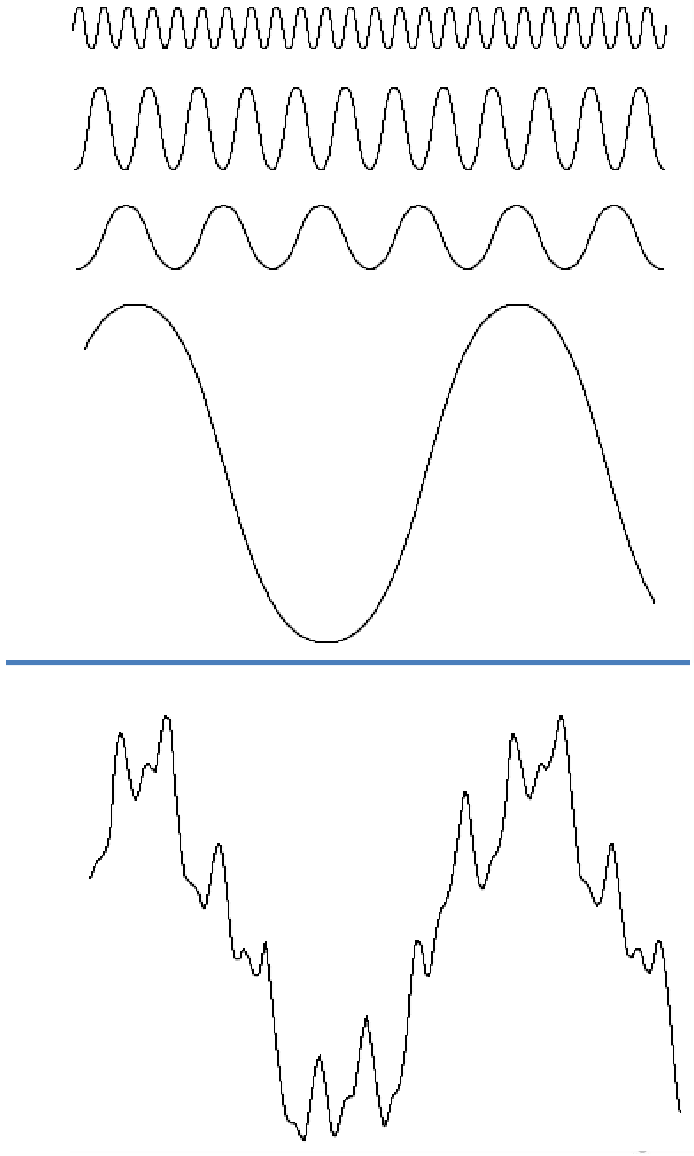
Signals in Frequency Domain – (cont.)



Signals in Frequency Domain – (cont.)

Jean Baptiste Joseph Fourier

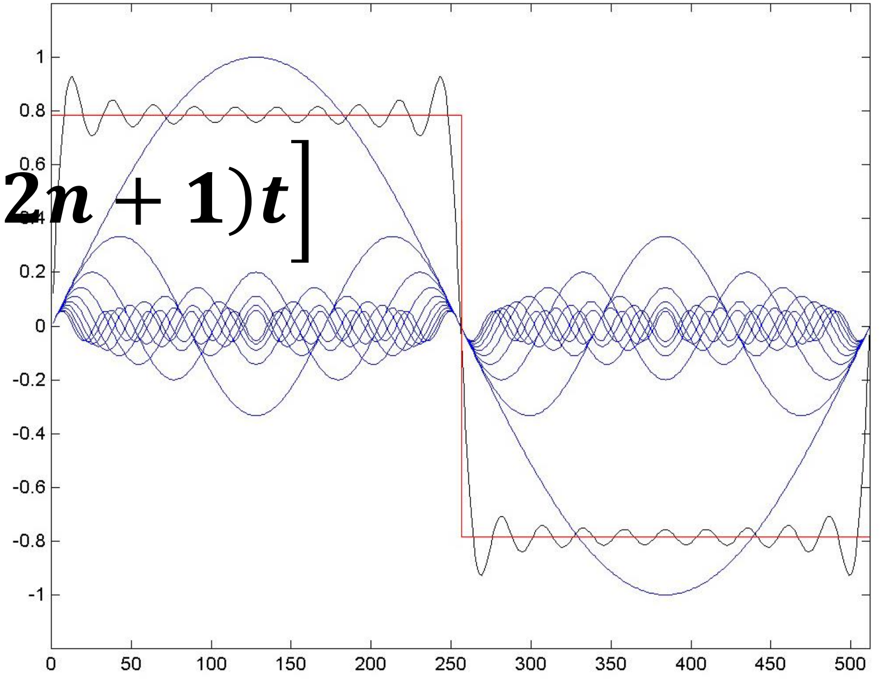
- **Developed one of the most important mathematical theories in modern engineering.**



Signals in Frequency Domain – (cont.)

Frequency Domain Representation

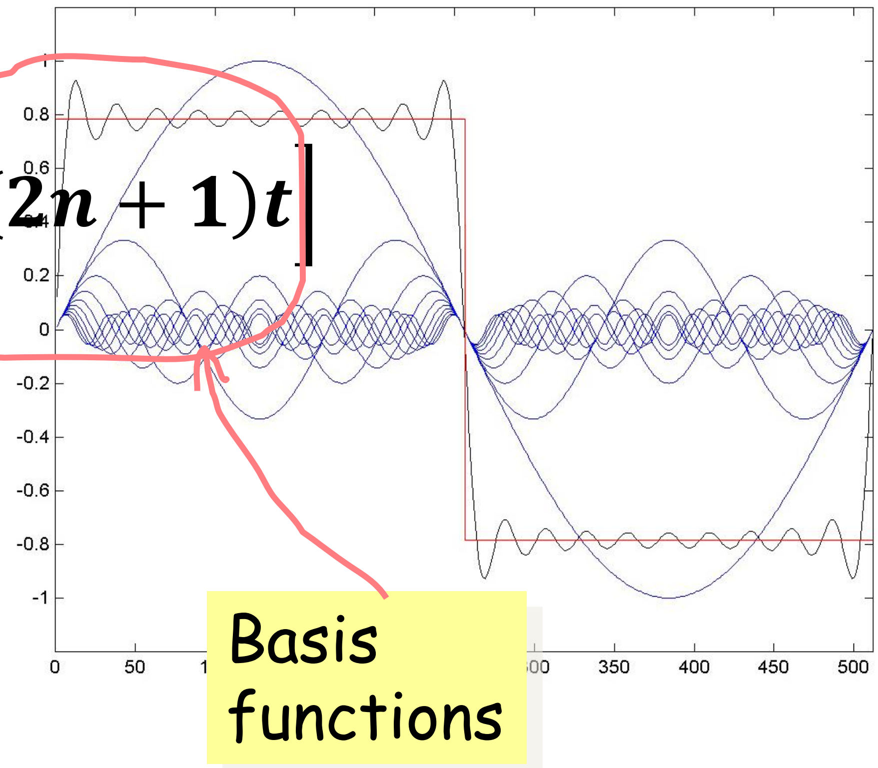
- **Any periodic signal can be described by a sum of sinusoids of different frequencies, each multiplied by a different coefficient.**
- $\boldsymbol{sq(t) = \sum_{n=-\infty}^{\infty} \frac{1}{2n+1} sin\left[\frac{2\pi}{\mu}(2n+1)t\right]}$
- **The sinusoids are called**
  - **"basis functions".**
- **The multipliers are called**
  - **"Fourier coefficients".**



Signals in Frequency Domain – (cont.)

Frequency Domain Representation

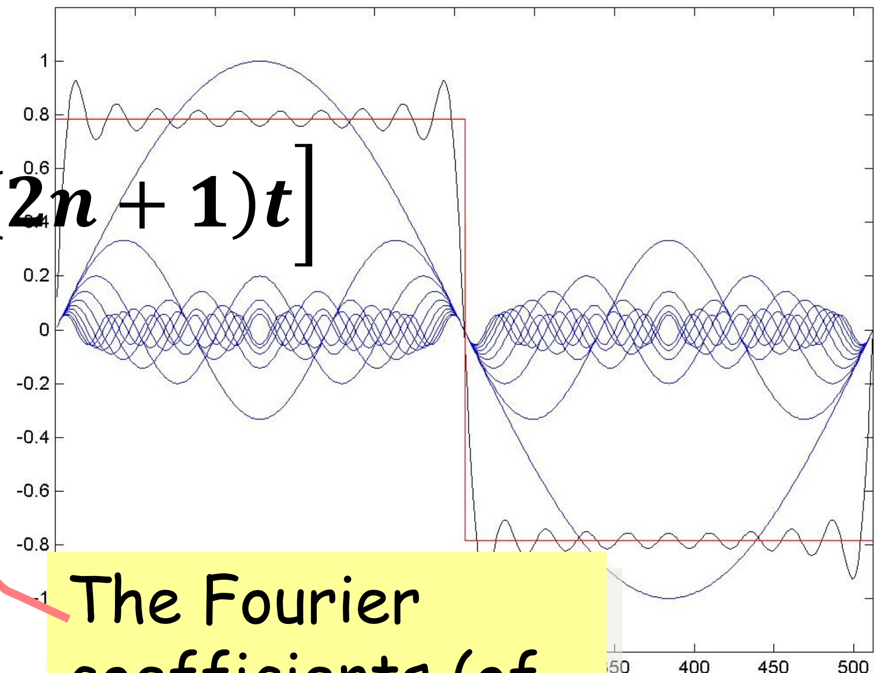
- **Any periodic signal can be described by a sum of sinusoids of different frequencies, each multiplied by a different coefficient.**
- $sq(t) = \sum_{n=-\infty}^{\infty} \frac{1}{2n+1} sin\left[\frac{2\pi}{\mu}(2n+1)t\right]$
- **The sinusoids are called**
  - **"basis functions".**
- **The multipliers are called**
  - **"Fourier coefficients".**



Signals in Frequency Domain – (cont.)

Frequency Domain Representation

- **Any periodic signal can be described by a sum of sinusoids of different frequencies, each multiplied by a different coefficient.**
- $sq(t) = \sum_{n=-\infty}^{\infty} \frac{1}{2n+1} sin\left[\frac{2\pi}{\mu}(2n+1)t\right]$
- **The sinusoids are called**
  - **"basis functions".**
- **The multipliers are called**
  - **"Fourier coefficients".**

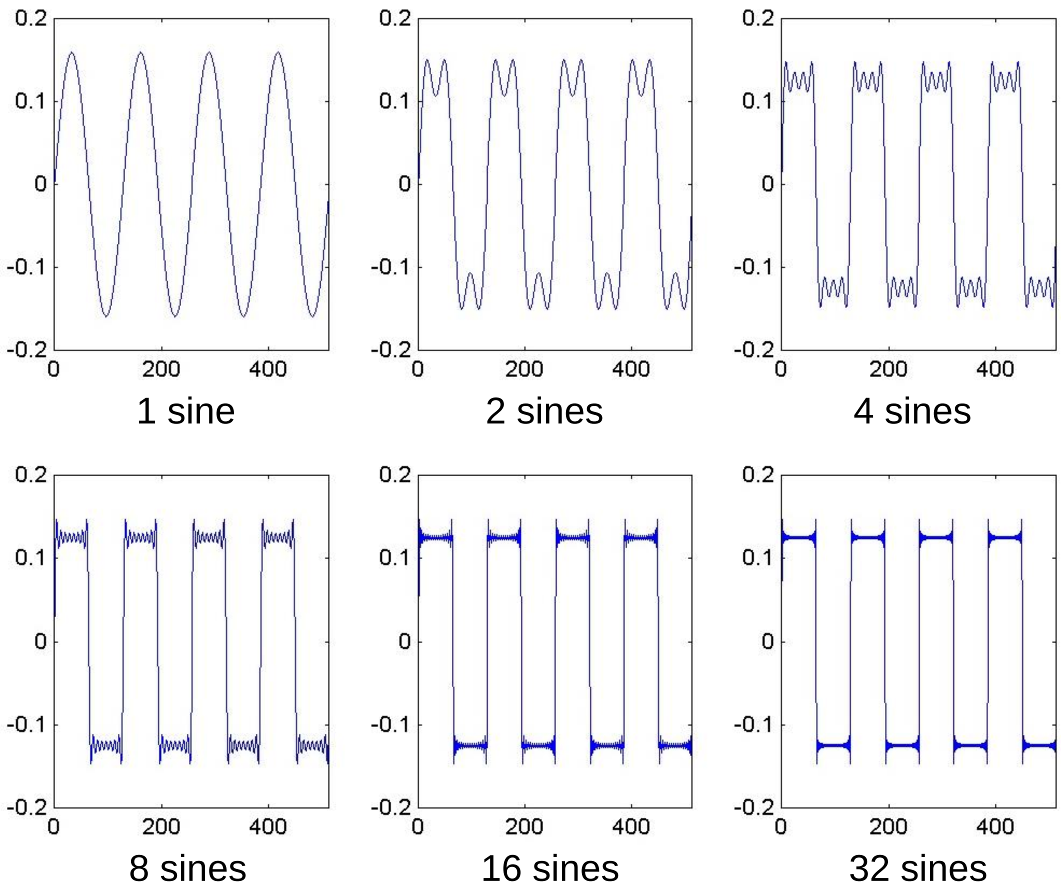


The Fourier coefficients (of a square wave).

Signals in Frequency Domain – (cont.)

Example: Partial Sum of a Square Wave

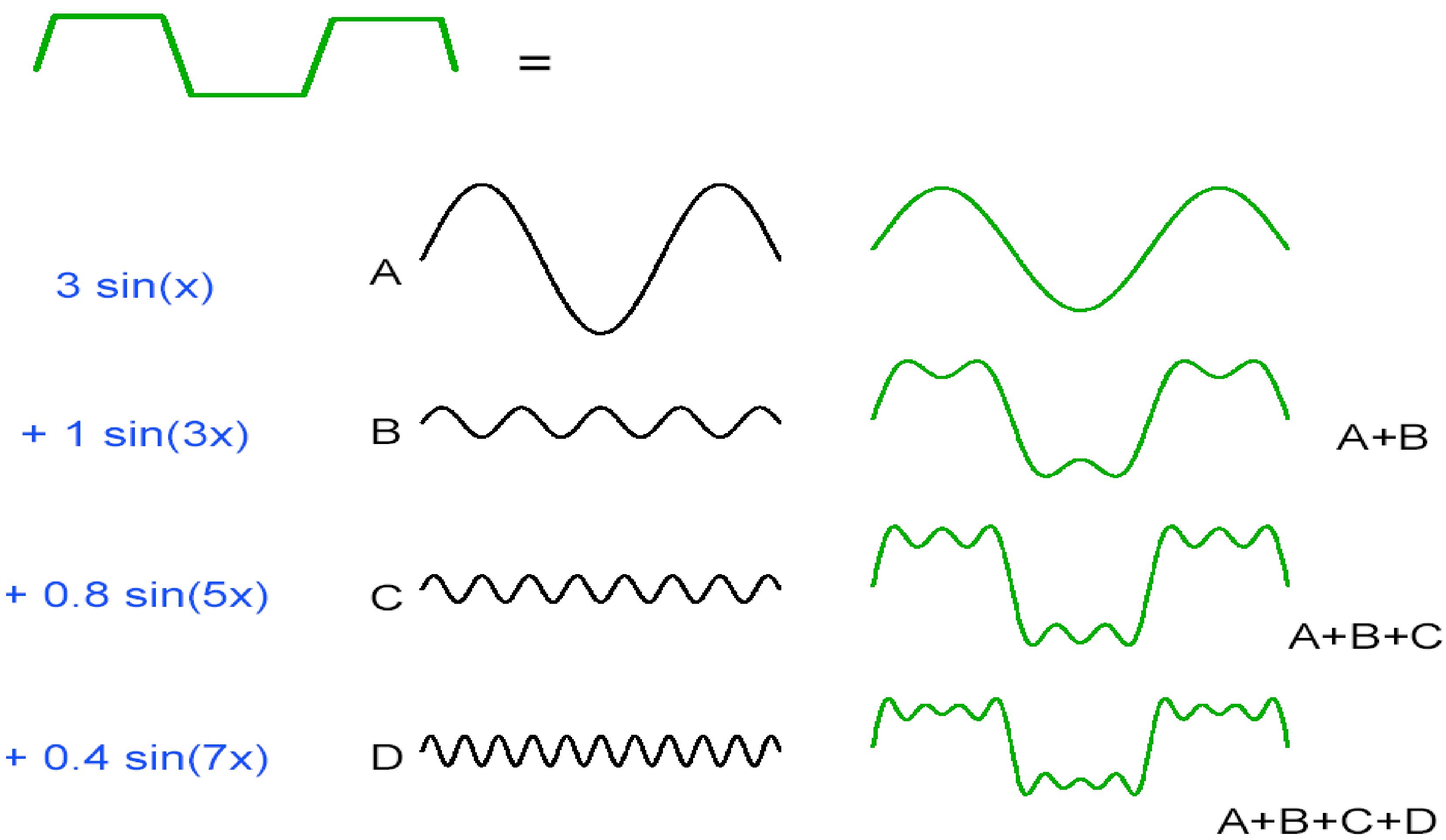
The limit of the given sequence of partial sums[1] is exactly a square wave



[1] the limit as n approaches infinity of the sum of n sines.

Signals in Frequency Domain – (cont.)

Example: Partial Sum of a Square Wave



Signals in Frequency Domain – (cont.)

Fourier Series

- **Is the decomposition of a T-periodic signal into a sum of sinusoids.**

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{2j\pi nt/T}$$

$$c_n = \frac{1}{T} \int_{-\infty}^{\infty} f(t) e^{-j2\pi tn/T}\, dt$$

Recall Euler's $e^{j\theta} = \cos\theta + j\,sin\theta$

The representation of a function by its Fourier Series is the sum of sinusoidal basis functions" multiplied by coefficients.

Fourier coefficients are generated by taking the inner product of the function with the basis.

measure of similarity

The basis functions correspond to modes of vibration.

Signals in Frequency Domain – (cont.)

Fourier Series

- **Is the decomposition of a T-periodic signal into a sum of sinusoids.**

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{2j\pi nt/T}$$

$$c_n = \frac{1}{T}\int_{-\infty}^{\infty} f(t)e^{-j2\pi tn/T}\, dt$$

real number results yield the amplitude of that sinusoid in the function.

Complex number results yields the amplitude and phase of that sinusoid in the function.

Recall Euler's $e^{j\theta} = \cos\theta + j\, sin\theta$

Fourier Transform

Fourier Transform (FT)

- **Even *non-periodic* functions can be decomposed into a continuous sum (integral) of sinusoids multiplied by a weighing function.**

- **The original function can be completely recovered (transformed back to its domain) without loss of data by using an inverse formula.**

Fourier Transform – (cont.)

1D Continuous Fourier Transform

$$\mathscr{F}\{f(t)\} = F(\mu) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t}dt$$

1D Inverse Continuous Fourier Transform

$$f(t) = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu t}d\mu$$

Fourier Transform – (cont.)

2D Continuous Fourier Transform

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}\,dx\,dy$$

2D Inverse Continuous Fourier Transform

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)e^{j2\pi(ux+vy)}\,du\,dv$$

Fourier Transform – (cont.)

1D Discrete Fourier Transform (DFT)

- **A discrete signal $\{f_x | x = 0, 1, \dots, M-1\}$, of finite length M, can be represented as a weighted sum of M sinudsoids $\{e^{-j2\pi ux/M} | u = 0, 1, \dots, M-1\}$, such that**

$$f_x = \sum_{u=0}^{M-1} F_u e^{j2\pi ux/M}$$

$$\boxed{f(t) = \int_{-\infty}^{\infty} F(\mu) e^{j2\pi\mu t} d\mu}$$

- **Where the set $\{F_u | u = 0, 1, \dots M-1\}$ are the Fourier coefficients defined as the projection of the original signal onto sinusoid u, given by**

$$F_u = \frac{1}{M} \sum_{x=0}^{M-1} f_x e^{-j2\pi ux/M}$$

$$\boxed{F(\mu) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi\mu t} dt}$$

Fourier Transform – (cont.)

1D Discrete Fourier Transform (DFT)

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

$u = 0, 1, \dots, M-1$

Original sample

sinusoidal component

$$f(t) = \int_{-\infty}^{\infty} F(\mu) e^{j2\pi\mu t} d\mu$$

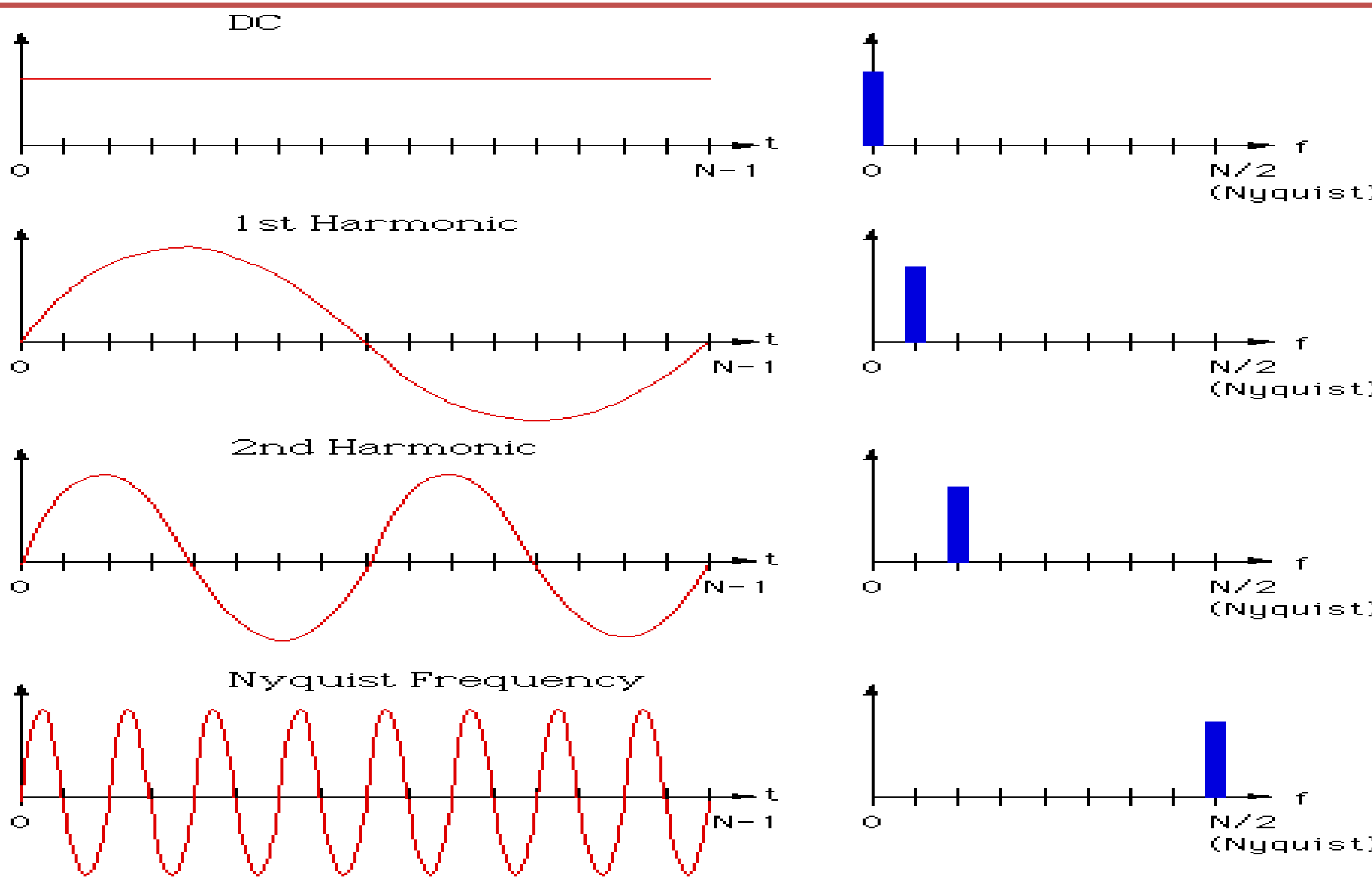
1D Inverse Discrete Fourier Transform (IDFT)

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

$x = 0, 1, \dots, M-1$

$$F(\mu) = \int_{-\infty}^{\infty} f(t) e^{-j2\pi\mu t} dt$$

Fourier Transform – (cont.)



• The relationship between the harmonics returned by the DFT and the periodic component in the time domain.

Fourier Transform – (cont.)

Properties of 1D Discrete Fourier Transform

Property	Expression
Magnitude (Spectrum)	$\lvert F(u)\rvert = [R^2(u) + I^2(u)]^{1/2}$
Phase Angle	$\lvert \emptyset(u)\rvert = tan^{-1}[\frac{I(u)}{R(u)}]$
Power Spectrum	$P(u) = \lvert F(u)\rvert^2$ $= R^2(u) + I^2(u)$
DC Component	$F(0) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)$
Even symmetry around origin	If signal is real and implies $F(u) = F(-u)$

Fourier Transform – (cont.)

2D Discrete Fourier Transform (DFT)

- For an signal of size $M \times N$

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

$u = 0, 1, \ldots, M-1, v = 0, 1, \ldots, N-1$

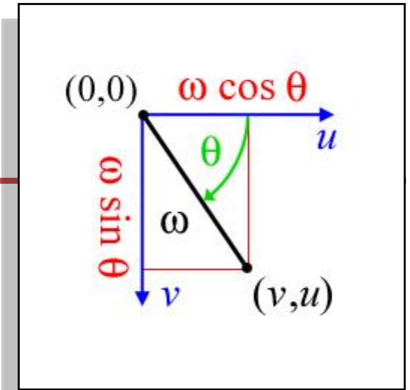
these complex exponentials are 2D sinusoids.

2D Inverse Discrete Fourier Transform (IDFT)

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

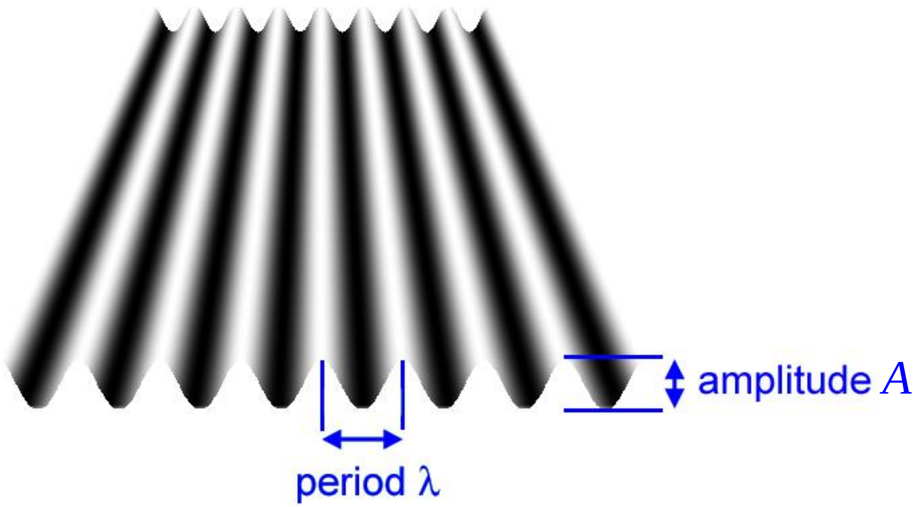
$x = 0, 1, \ldots, M-1, y = 0, 1, \ldots, N-1$

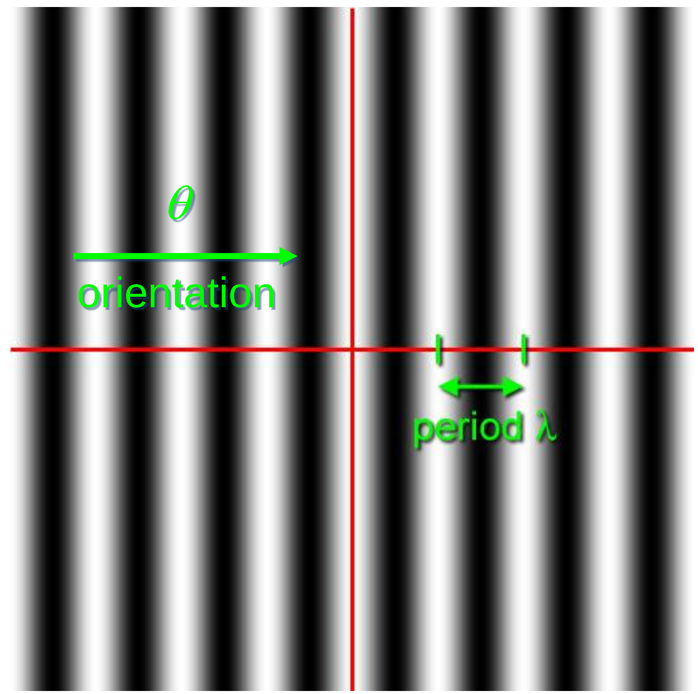
Fourier Transform – (cont.)



2D Sinusoids:

... are plane waves with grayscale amplitudes, periods in terms of lengths, ...



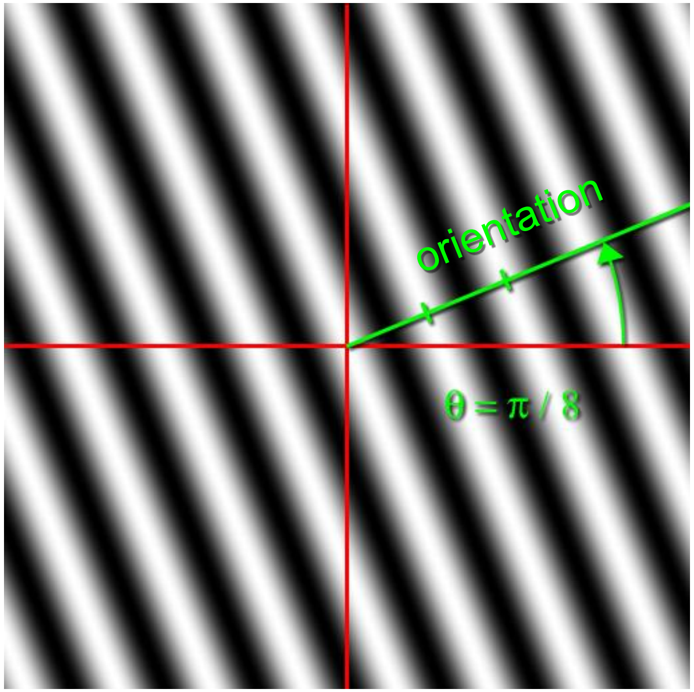


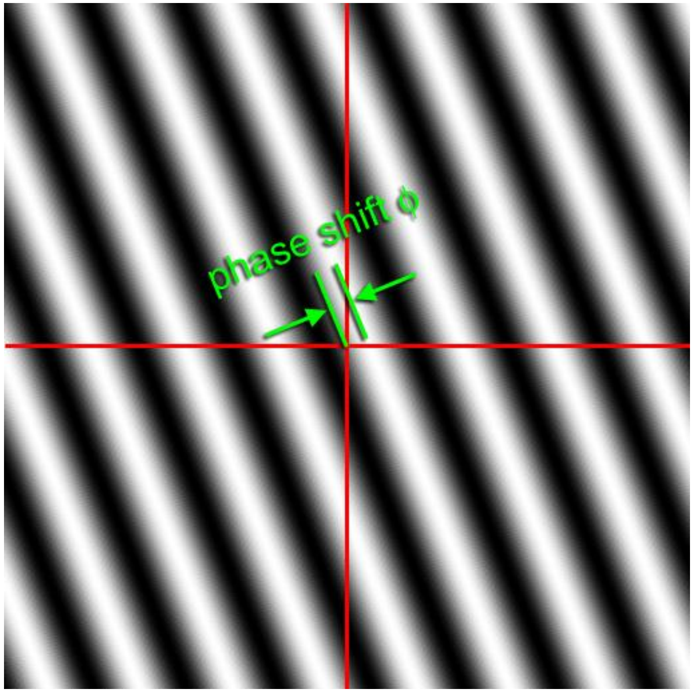
ϕ = phase shift

Fourier Transform – (cont.)

2D Sinusoids:

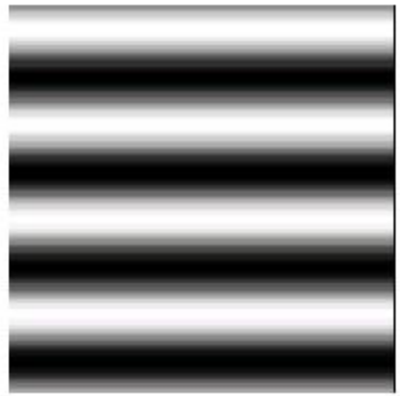
... specific orientations, and phase shifts.



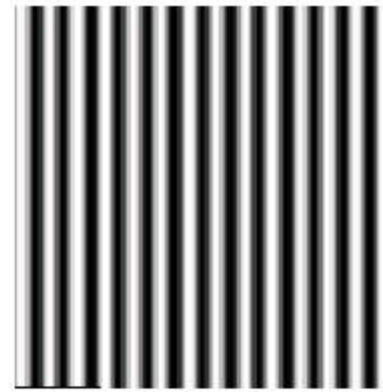


Fourier Transform – (cont.)

= 3

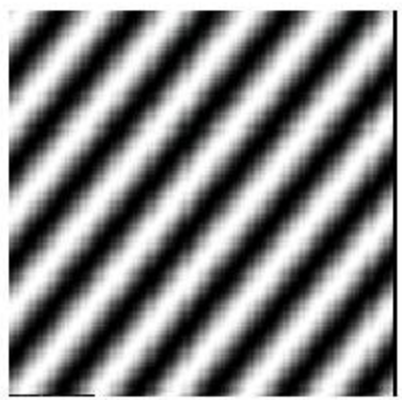


- 23

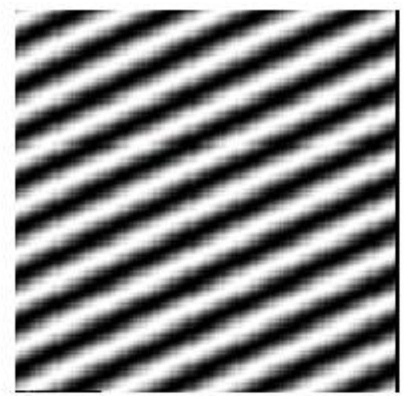


+

- 10



+ 7

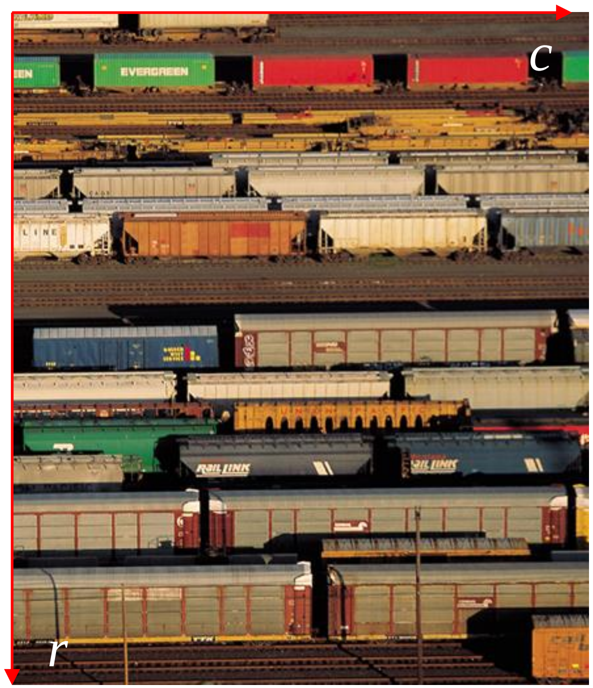


+ ...

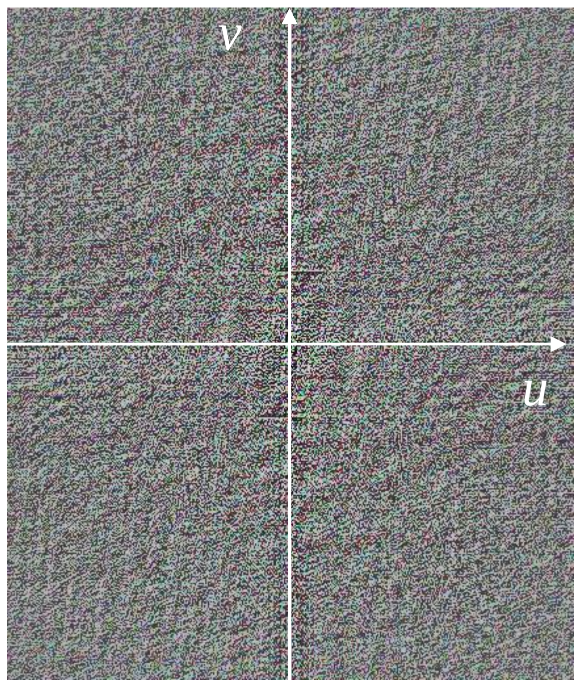
Demo

Fourier Transform – (cont.)

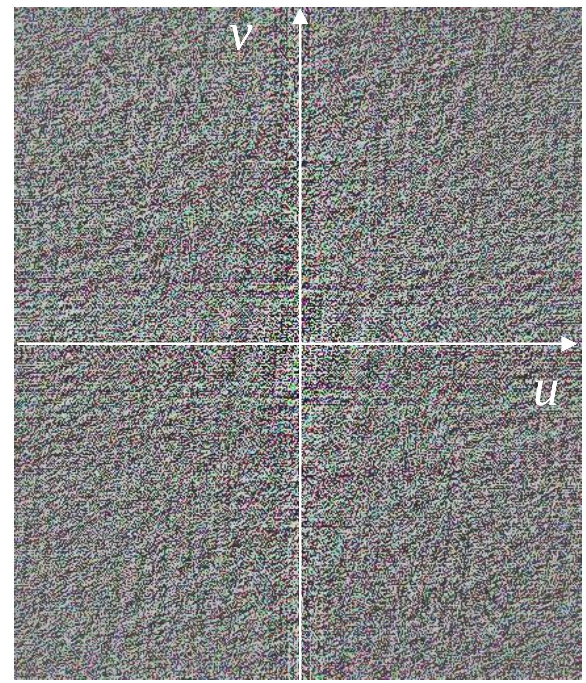
The Fourier Transform of an Image



$\mathbf{I}$



$\mathrm{Re}[\mathfrak{F}\{\mathbf{I}\}]$



$\mathrm{Im}[\mathfrak{F}\{\mathbf{I}\}]$

Fourier Transform – (cont.)

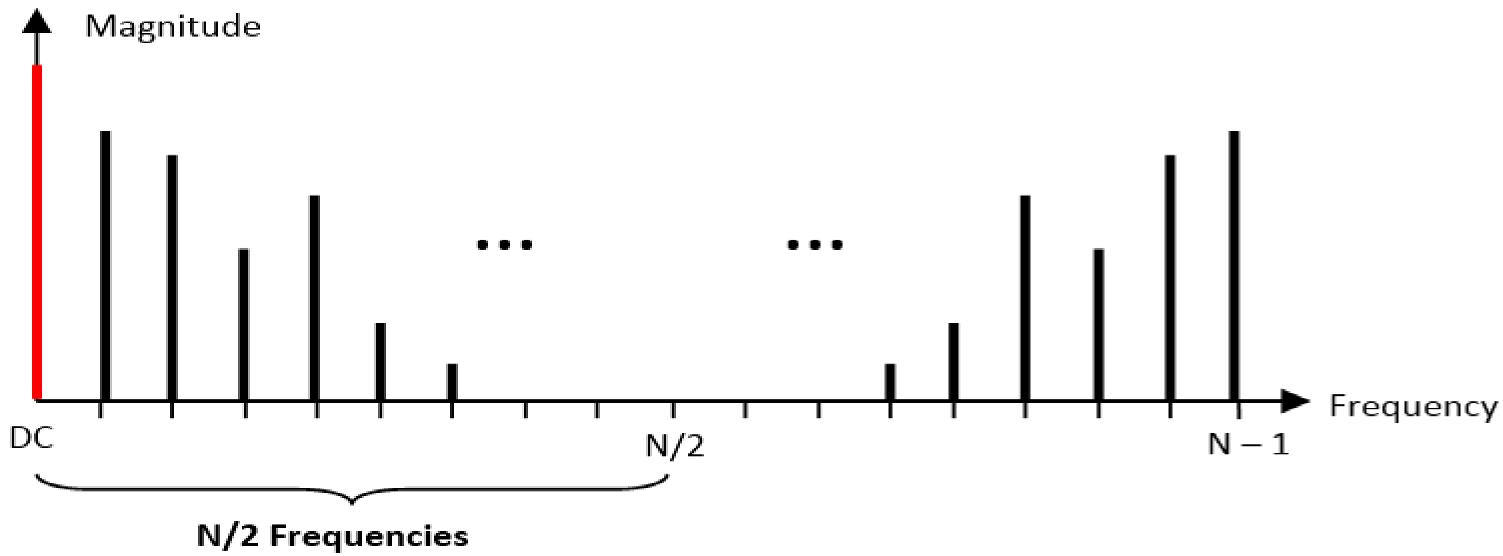
Properties of 2D Discrete Fourier Transform (DFT)

Property	Expression
Magnitude (Spectrum)	$\lvert F(u,v)\rvert = [R^2(u,v) + I^2(u,)]^{1/2}$
Phase Angle	$\lvert \emptyset(u,v)\rvert = tan^{-1}[\frac{I(u,v)}{R(u,v)}]$
Power Spectrum	$P(u,v) = \lvert F(u,v)\rvert^2 = R^2(u,v) + I^2(u,v)$
DC Component	$F(0,0) = \frac{1}{M}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$
Even symmetry around origin	If signal is real and implies $F(u,v) = F(-u,-v)$

Fourier Transform – (cont.)

Properties - Symmetry

- Magnitudes of the FT components are symmetric on $\boldsymbol{N/2}$, while differ in phase shift… (i.e. there are **only $N/2$ different frequencies** that represent the signal with N samples)
- (Try to prove it mathematically by substituting in Fourier equation by $(\boldsymbol{N-u})$ and compare the results with $\boldsymbol{F}(\boldsymbol{u})$).



Fourier Transform – (cont.)

Properties – Translation Invariant

- $\mathscr{F}\left(f(x-x_0, y-y_0)\right) \Leftrightarrow F(u,v)e^{-j2\pi(\frac{ux_0}{M}+\frac{vy_0}{N})}$
- If we take the magnitude of the FT of the translated signal, then it'll be equal to the magnitude of the FT of the original signal.
- Prove it by substituting in FT equation.

Fourier Transform – (cont.)

Complexity

	1D	2D
Original Transform	$\boldsymbol{O}(N^2)$	$\mathbf{O}(N^4)$
Fast Transform	$\mathbf{O}(N \log N)$	$\mathbf{O}(N^2 \log N)$

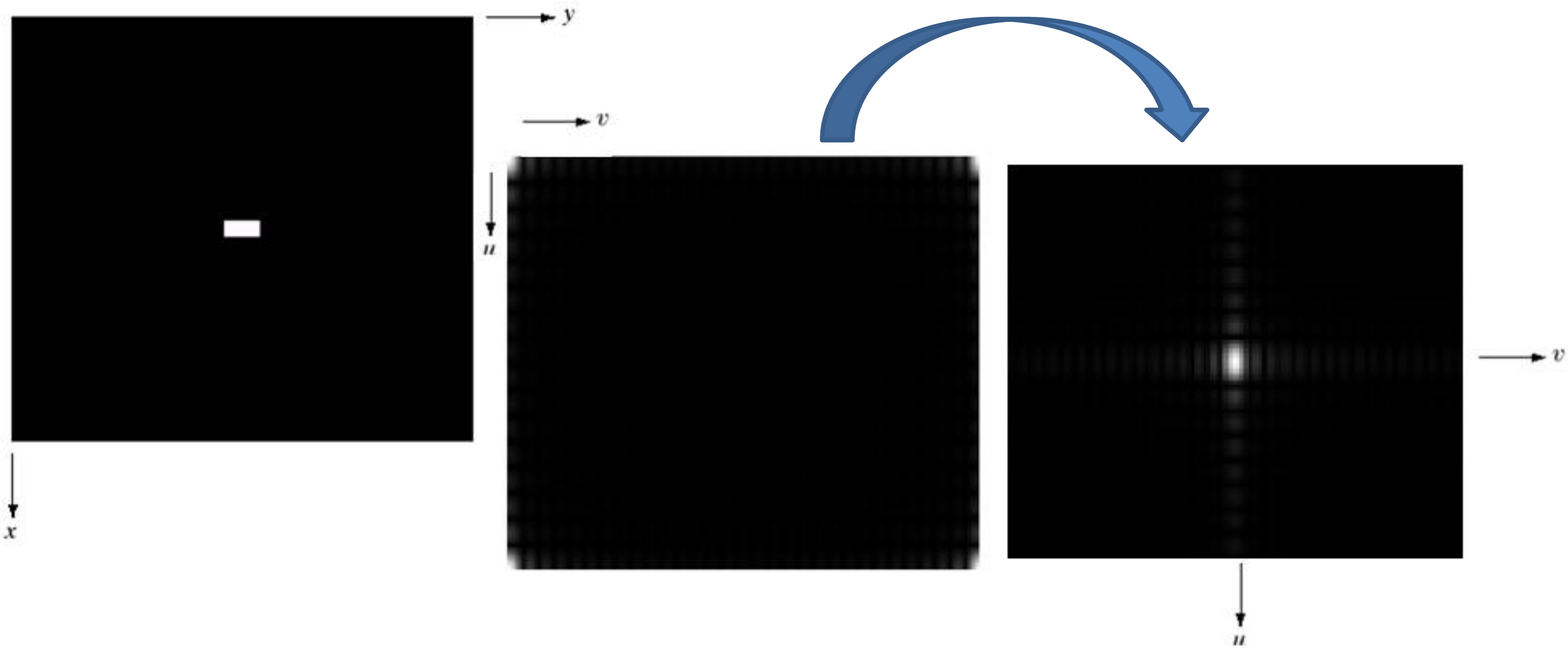
- **Matlab's Fast Fourier Transform Functions**

`fft fft2`

Fourier Transform – (cont.)

Displaying the FT of an Image

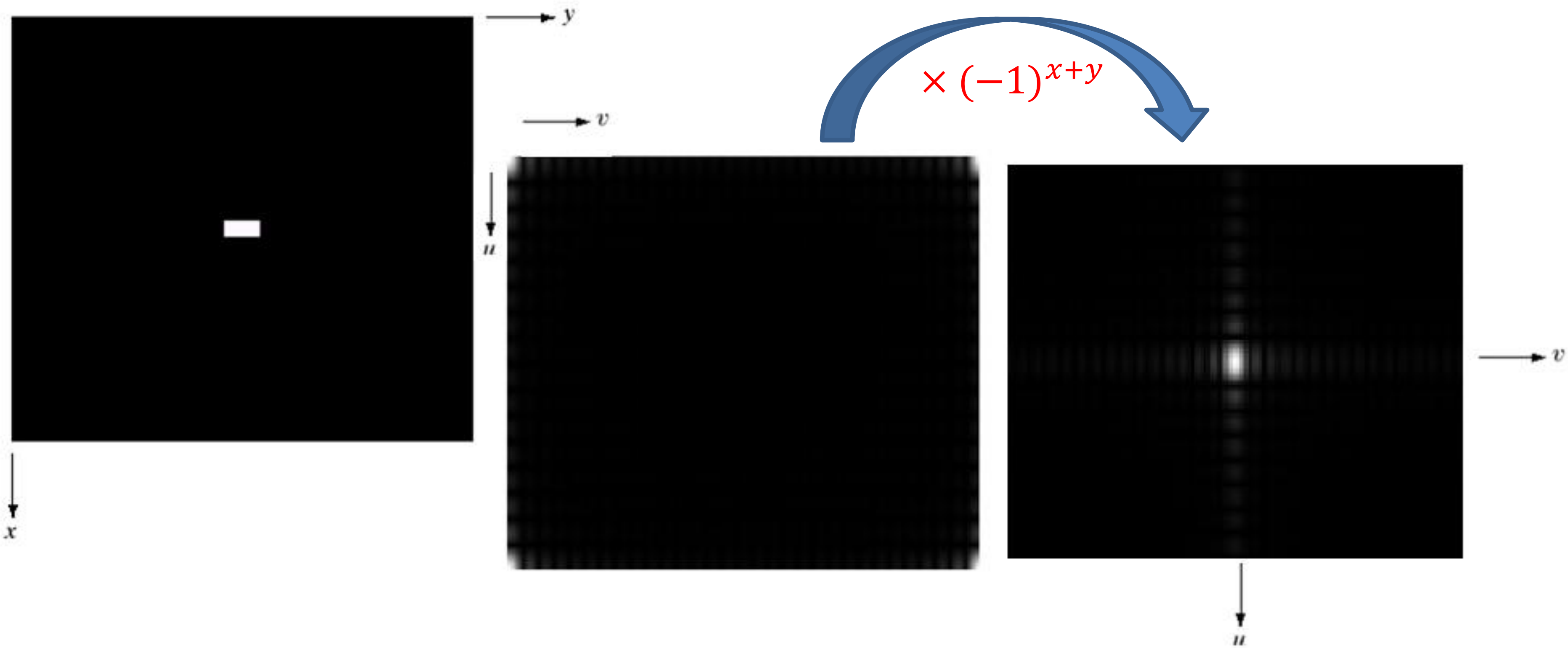
1. We shift the center to place the largest energy in the middle and be able to observe the behavior.



Fourier Transform – (cont.)

Displaying the FT of an Image

1. We shift the center to place the largest energy in the middle and be able to observe the behavior.



Fourier Transform – (cont.)

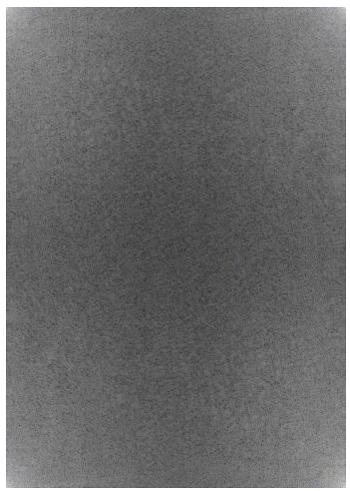
Matlab's `fftshift` and `ifftshift`

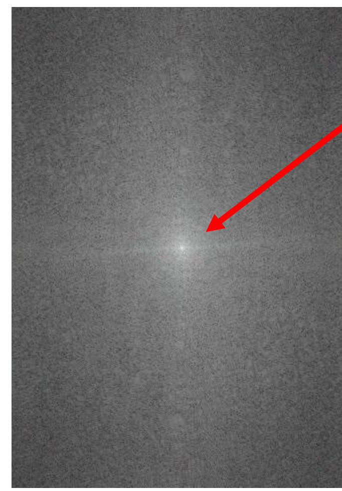
`I = ifftshift(J):`

`J = fftshift(I):`

origin

from FFT2
or ifftshift





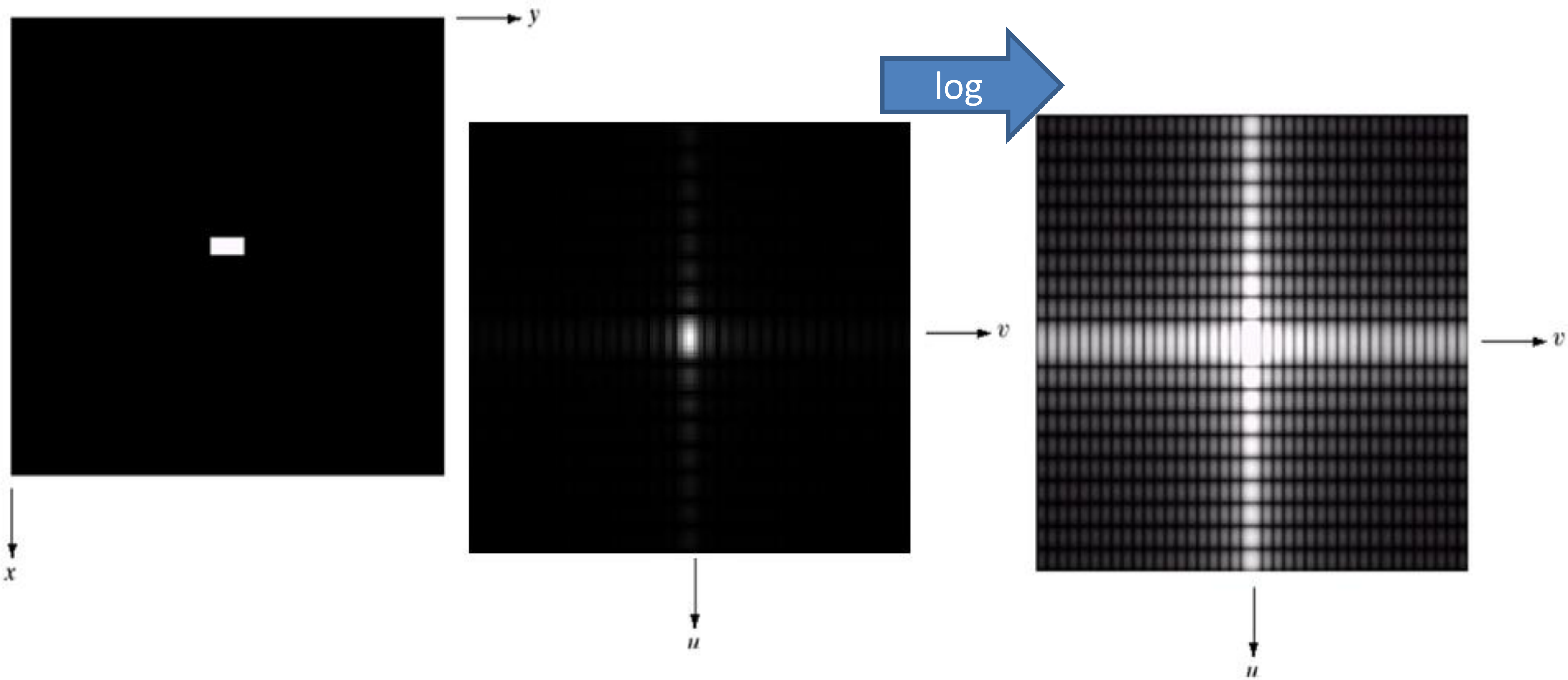
origin

after fftshift

Fourier Transform – (cont.)

Displaying the FT of an Image

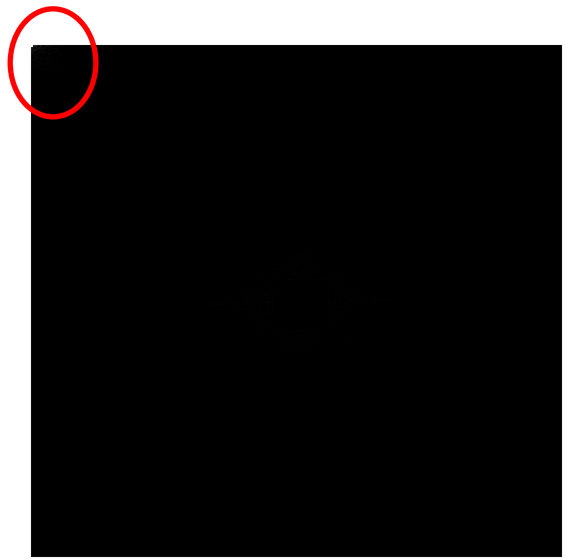
2. It is common practice to scale the FT to be able to display all values in the range.



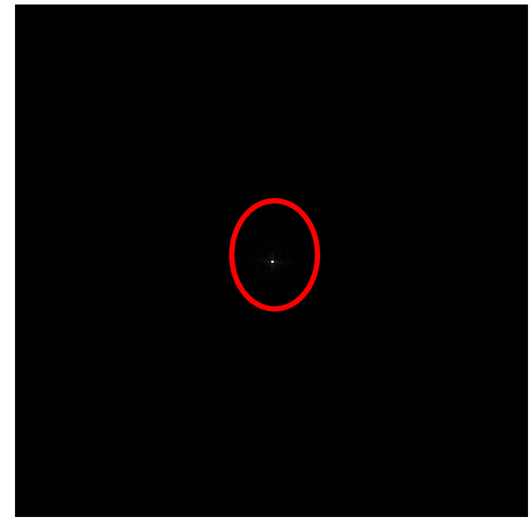
Fourier Transform – (cont.)

Example1: Displaying

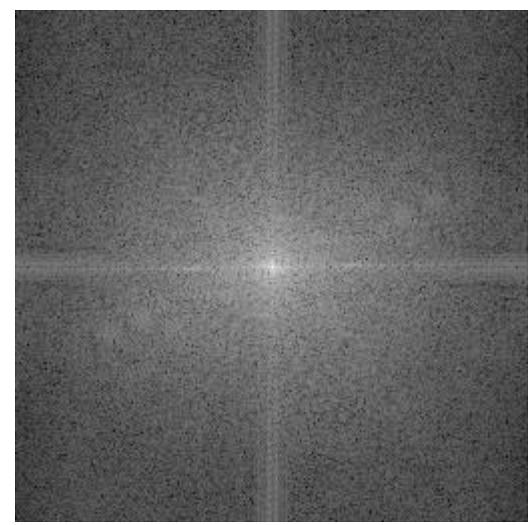
Test image



Spectrum



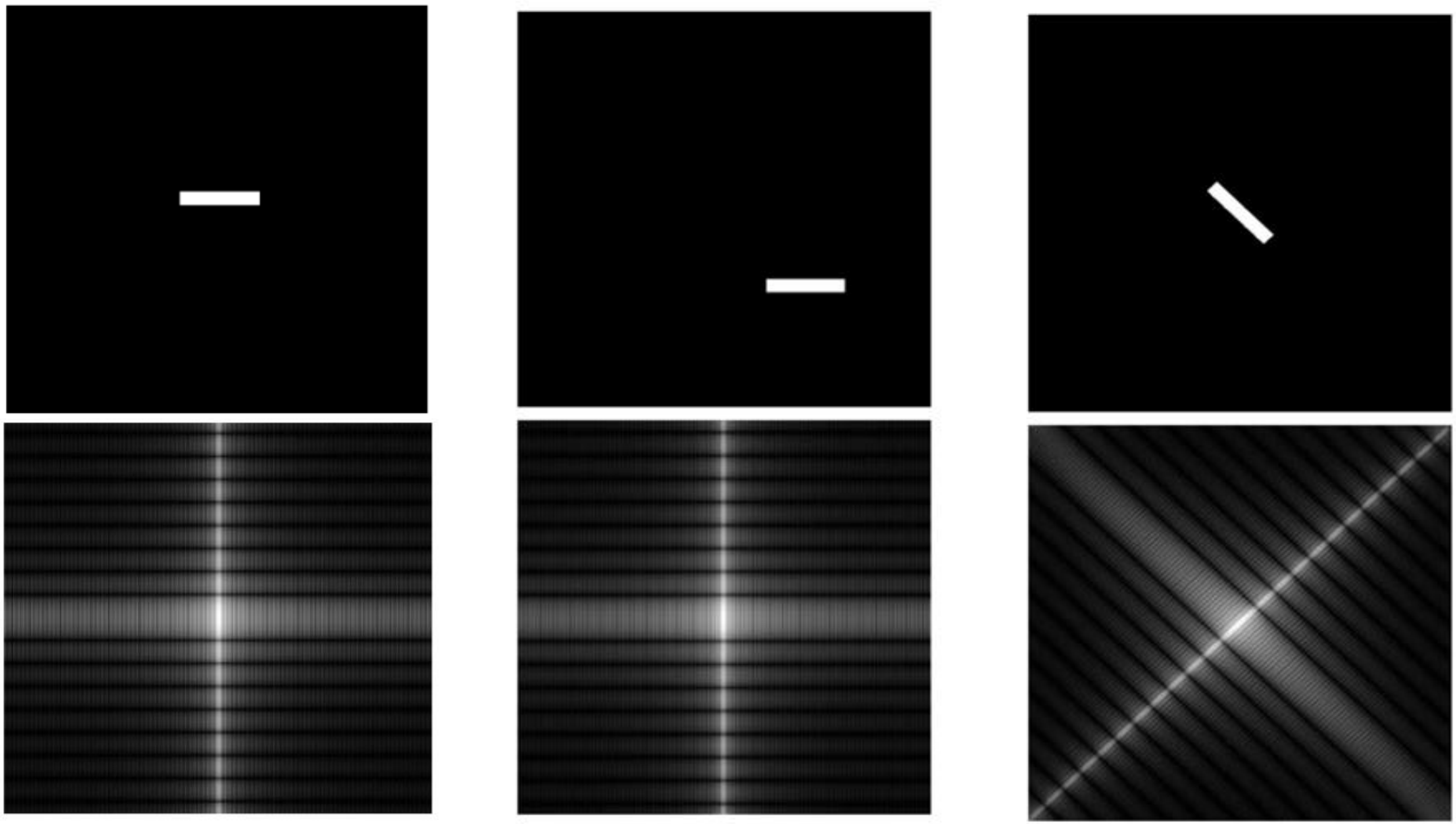
Centered spectrum



Scaled centered spectrum

Fourier Transform – (cont.)

Example2: Translation and Rotation

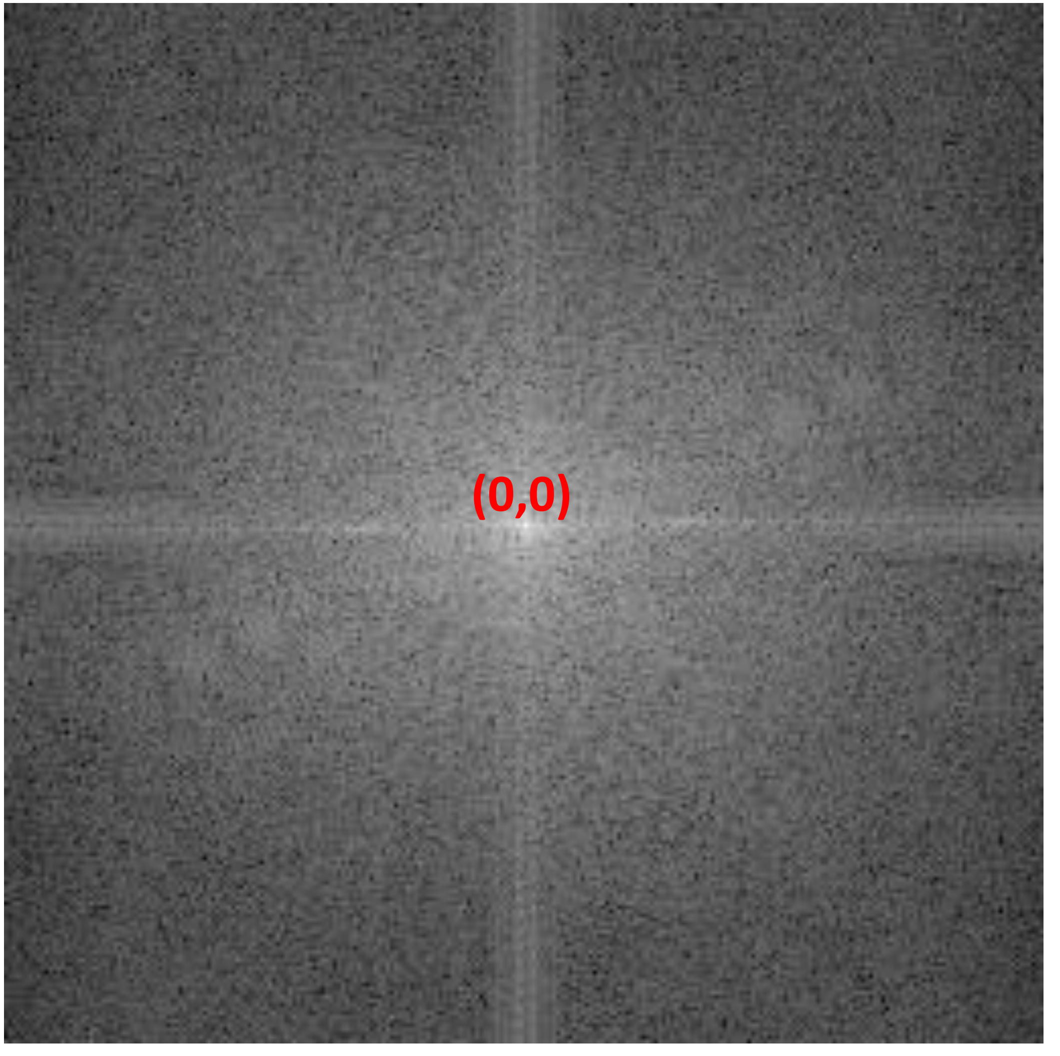


Fourier Transform – (cont.)

Remarks

- The DC component (lowest frequency) has the largest magnitude.

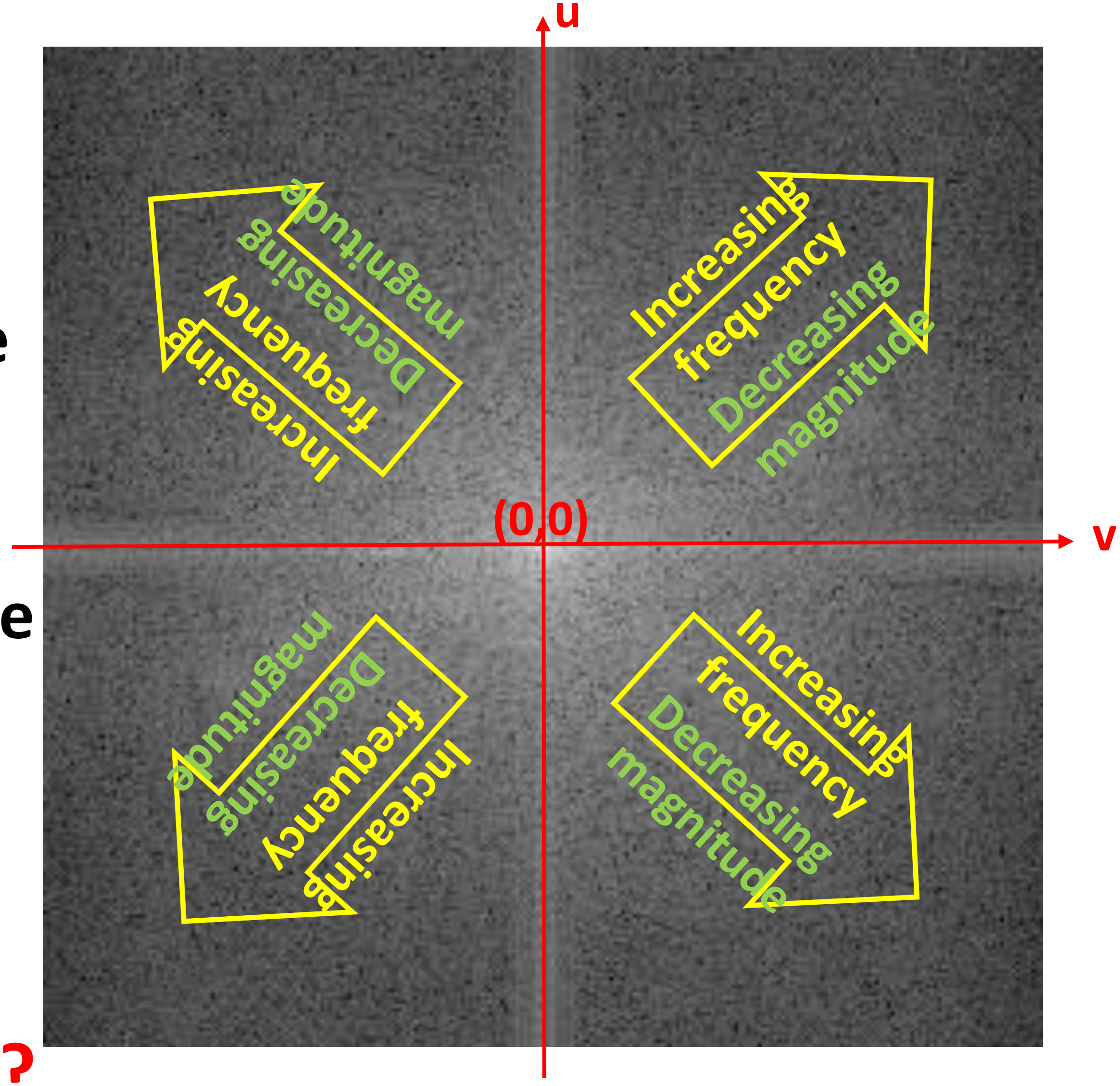
→ average intensity of image.



Fourier Transform – (cont.)

Remarks

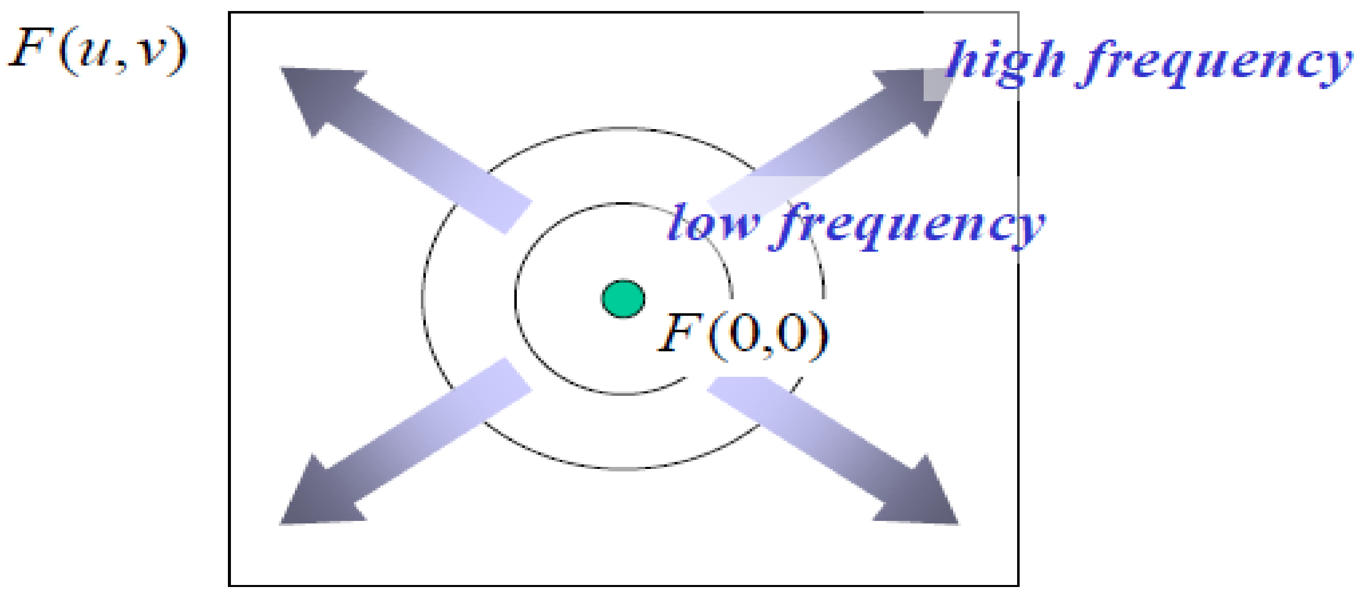
- **Notice the Symmetry of every two quadrants around the origin.**
- **As the frequency increases (towards the corners), the magnitude of the Fourier components decreases.**



WHY shift the center?

Fourier Transform – (cont.)

- **Low frequency**: slowly varying components of image such as backgrounds and steady areas.
- **High frequency**: faster gray level changes such as noise and edges of objects.



Demo

Fourier Transform – (cont.)

Power Spectrum

For display, the $\log$ of the power spectrum is often used.

The power spectrum of a signal is the square of the magnitude of its Fourier Transform.

$$\begin{aligned}\left|\mathfrak{I}(u,v)\right|^2 &= \mathfrak{I}(u,v)\mathfrak{I}^*(u,v)\\ &= [\mathrm{Re}\,\mathfrak{I}(u,v)+ i\,\mathrm{Im}\,\mathfrak{I}(u,v)][\mathrm{Re}\,\mathfrak{I}(u,v)- i\,\mathrm{Im}\,\mathfrak{I}(u,v)]\\ &= [\mathrm{Re}\,\mathfrak{I}(u,v)]^2 + [\mathrm{Im}\,\mathfrak{I}(u,v)]^2 .\end{aligned}$$

At each location (*u,v*) it indicates the squared intensity of the frequency component.

```
For display in Matlab:
PS = fftshift(2*log(abs(fft2(I))+1));
```

Fourier Transform – (cont.)

Power Spectrum

The power spectrum (PS) is defined by $\mathrm{PS}(\mathrm{I}) = \left|\mathfrak{F}\{\mathrm{I}(u,v)\}\right|^2$ **.**

We take the base-e logarithm of the PS in order to view it. Otherwise its dynamic range could be too large to see everything at once. We add 1 to it first so that the minimum value of the result is 0 rather than –infinity, which it would be if there were any zeros in the PS. Recall that log(f^2 **) = 2log(** f **).**

Multiplying by 2 is not necessary if you are generating a PS for viewing, since you'll probably have to scale it into the range 0-255 anyway. It is much easier to see the structures in a Fourier plane if the origin is in the center. Therefore we usually perform an fftshift on the PS before it is displayed.

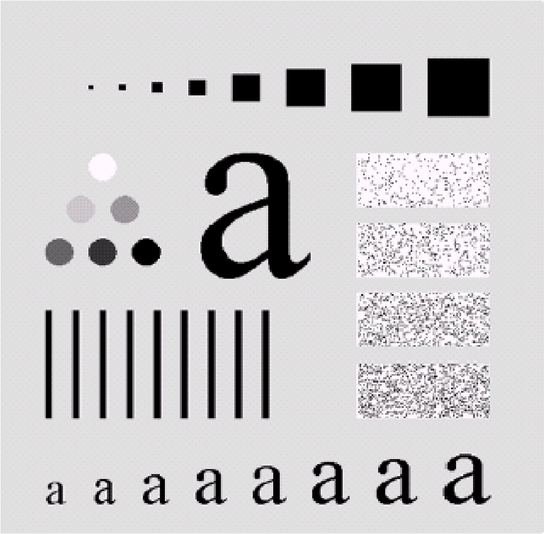
```
>> PS = fftshift(log(abs(fft2(I))+1));
>> M  = max(PS(:));
>> image(uint8(255*(PS/M)));
```

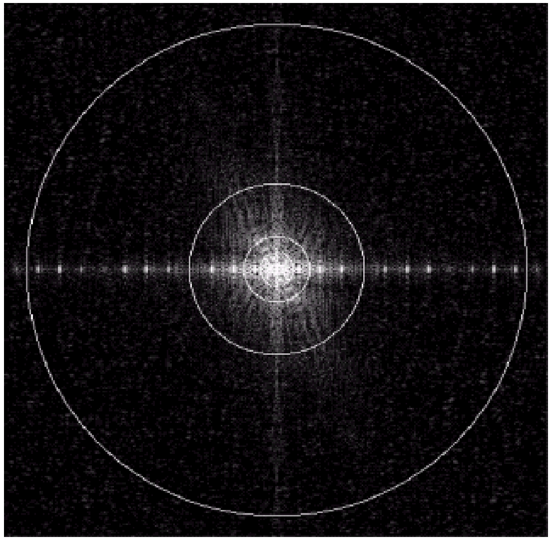
If the PS is being calculated for later computational use -- for example the autocorrelation of a function is the inverse FT of the PS of the function -- it should be calculated by `>> PS = abs(fft2(I)).^2;`

Fourier Transform – (cont.)

Power Spectrum

- By setting some standard *cutoff frequencies* loci, and computing circles that enclose specified amount of total image power P_T, we can compare the contribution of different ranges of frequencies to the reconstruction of the image, which gives more knowledge and control in the design of filters.
- $P_T = \sum_{u=0}^{P-1} \sum_{v=0}^{P-1} P(u, v)$





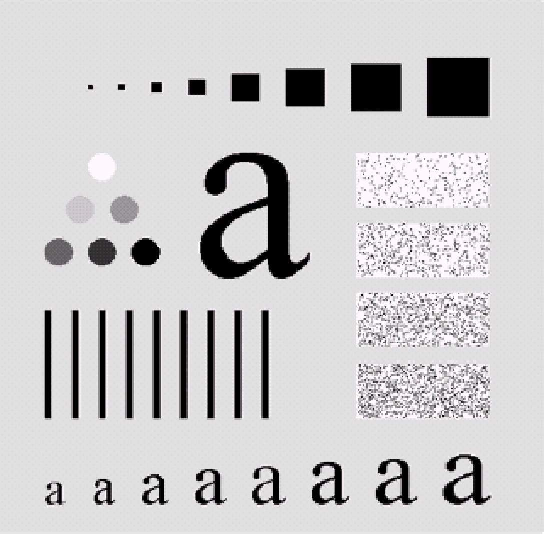
Fourier Transform – (cont.)

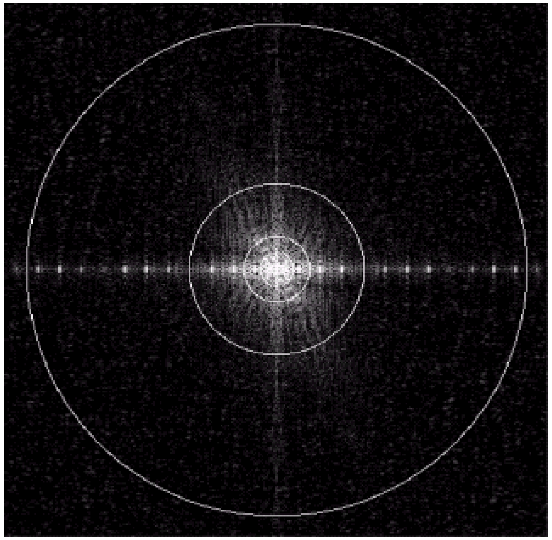
Power Spectrum

- If the FT is centered, a circle of radius D_o with the origin at the center of the frequency rectangle encloses α percent of the power, where

$$\alpha = 100\left[\sum_u \sum_v \frac{P(u,v)}{P_T}\right].$$

Circle radius	% of Power
10	87.0
30	93.1
60	95.7
160	97.8
460	99.2





Applications of FT

1. Sampling
2. Filtering
3. Shape Description.

Frequency Filtering

Why?

- **It makes more sense to filter in the spatial domain using small filter masks, but it is more computationally efficient to do the filtering in the frequency domain.**
- **Filtering is more intuitive and controllable in the frequency domain.**

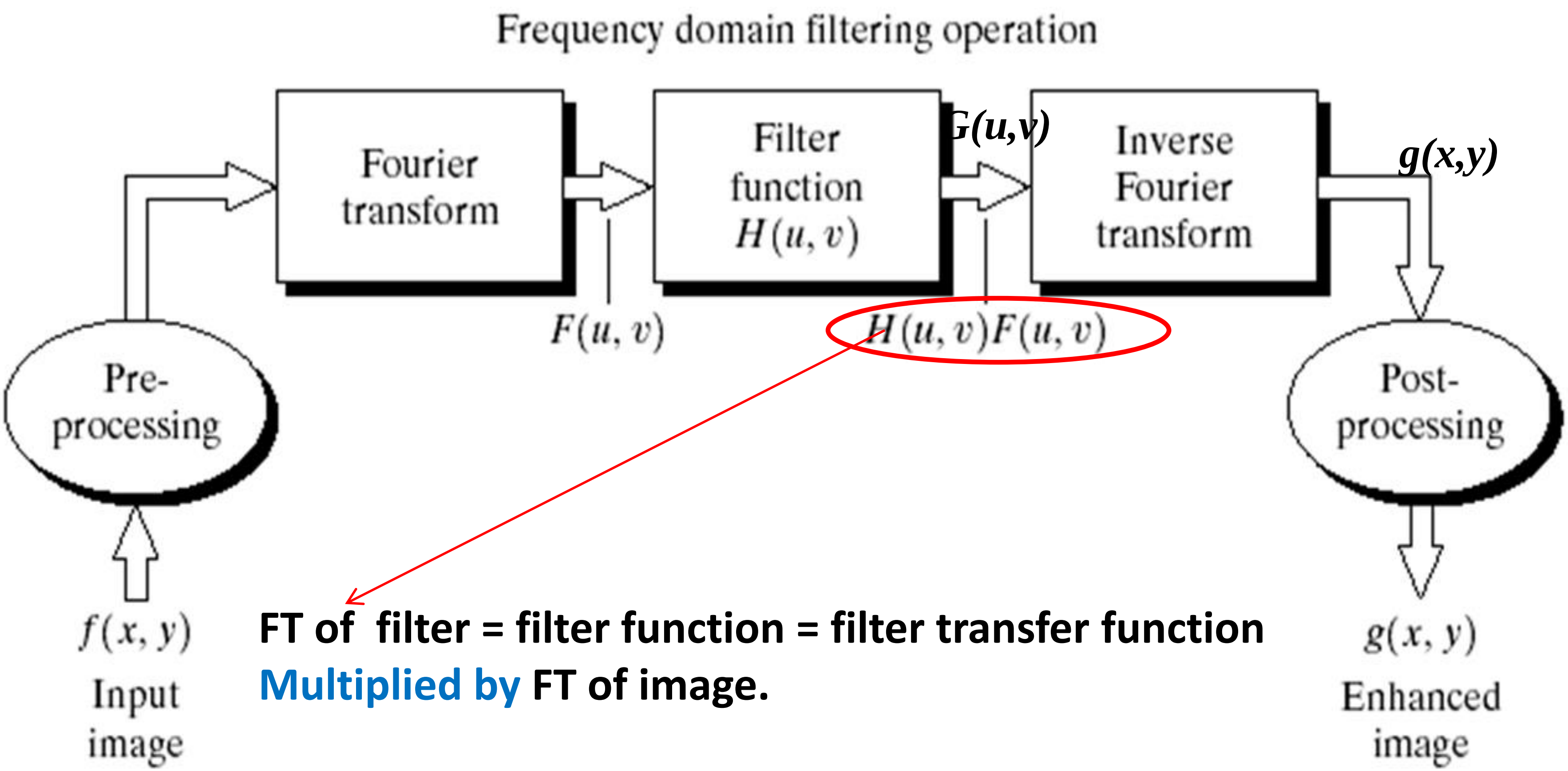
Frequency Filtering – (cont.)

Convolution Theorem

- **Convolution in the spatial domain corresponds to multiplication in the frequency domain, and vice versa.**
- $\boldsymbol{f(x,y) \star h(x,y) \leftrightarrow F(u,v)H(u,v)}$
- $\boldsymbol{f(x,y)h(x,y) \leftrightarrow F(u,v) \star H(u,v)}$
- **This theorem forms the basic step of filtering the in frequency domain.**

Frequency Filtering – (cont.)

Basic Steps

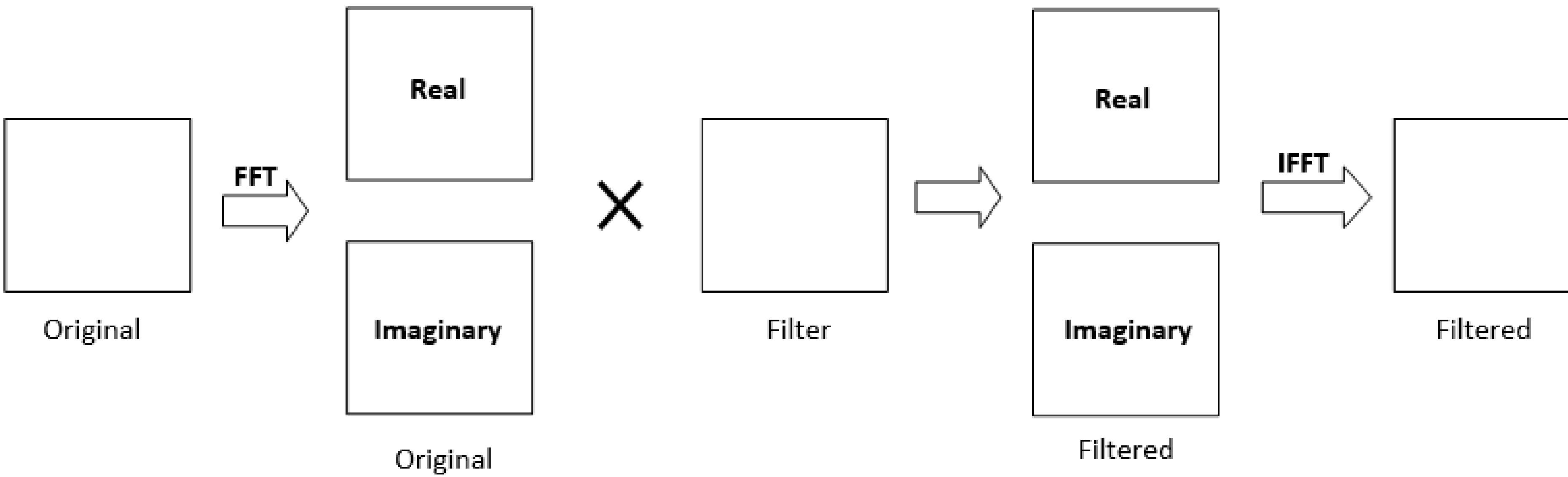


FT of filter = filter function = filter transfer function
Multiplied by FT of image.

This is NOT matrix multiplication.
It is array multiplication, or pixel-wise product.

Frequency Filtering – (cont.)

Basic Steps



FT of filter = filter function = filter transfer function
Multiplied by FT of image.

This is NOT matrix multiplication.
It is array multiplication, or pixel-wise product.

Frequency Filtering – (cont.)

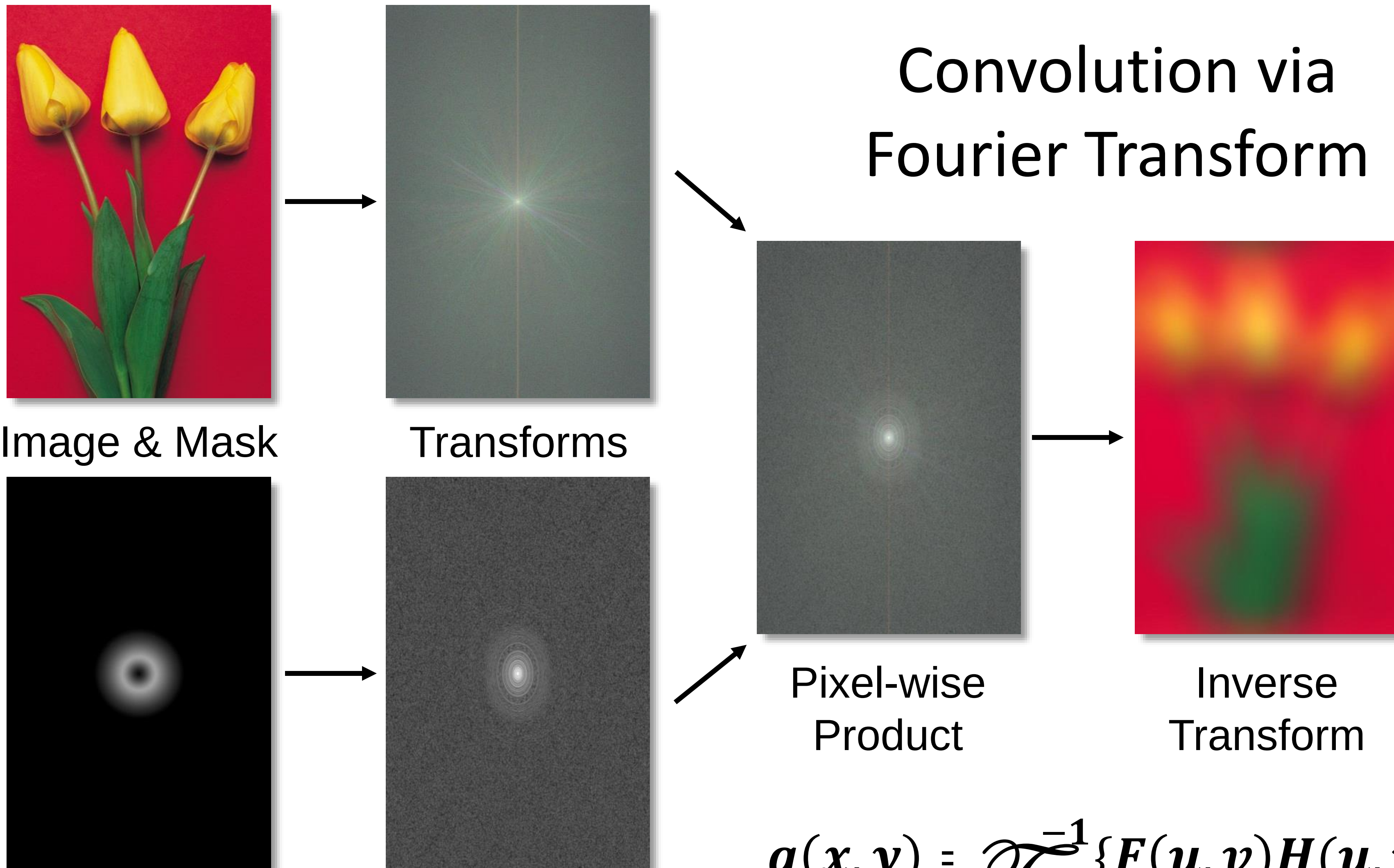
Correspondence Between Filtering in the Spatial and Frequency Domains

- **Convolution between image and mask in spatial domain ←→ multiplication between FT of image and FT of mask.**
- **Not that straightforward, both must be padded before the transform.**

Domain	Filtering Process	Complexity
Spatial	convolution	$\mathbf{O}(N^2K^2)$
Frequency	DFT, multiplication, IDFT	$\mathbf{O}(N^2 \log N)$

Frequency Filtering – (cont.)

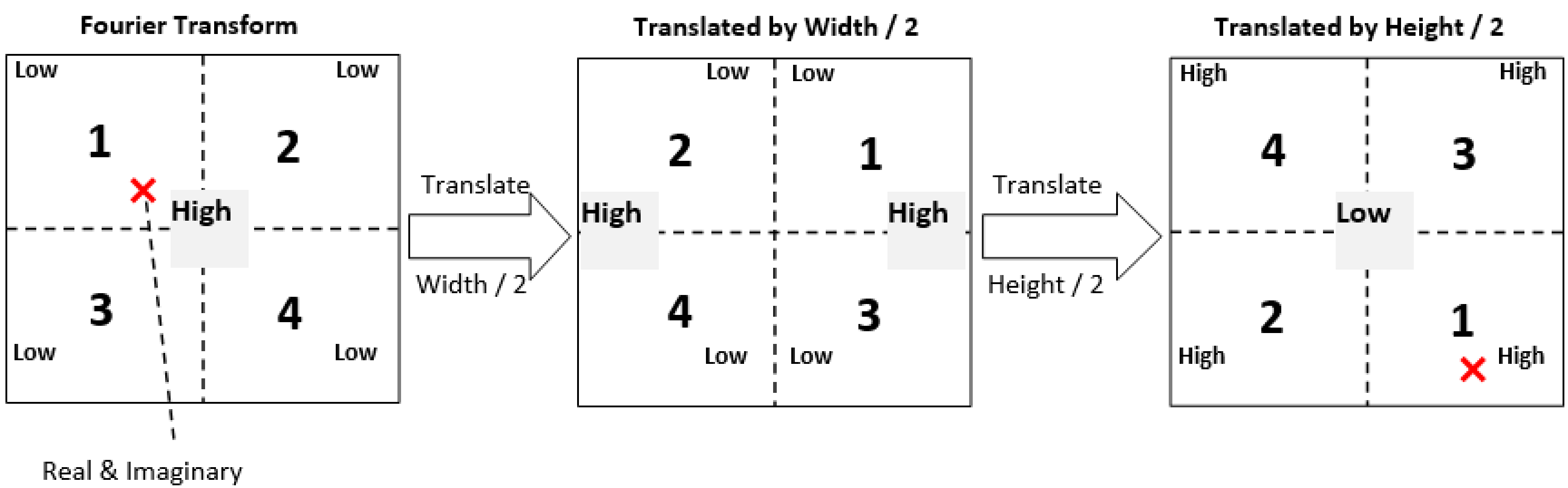
Convolution via Fourier Transform



$$g(x,y) = \mathcal{F}^{-1}\{F(u,v)H(u,v)\}$$

Frequency Filtering – (cont.)

- **Low frequency: slowly varying components of image such as backgrounds and steady areas.**
- **High frequency: faster gray level changes such as noise and edges of objects.**



Frequency Filtering – (cont.)

- Edges and sharp transitions in an image contribute significantly to high frequencies.

- Blurring (Lowpass)

- Hence, smoothing is achieved by attenuating high frequency components and passing low.

- Sharpening (Highpass)

- Hence, sharpening is achieved by attenuating low frequency components and amplifying high.

Frequency Filtering – (cont.)

Blur/Sharpen

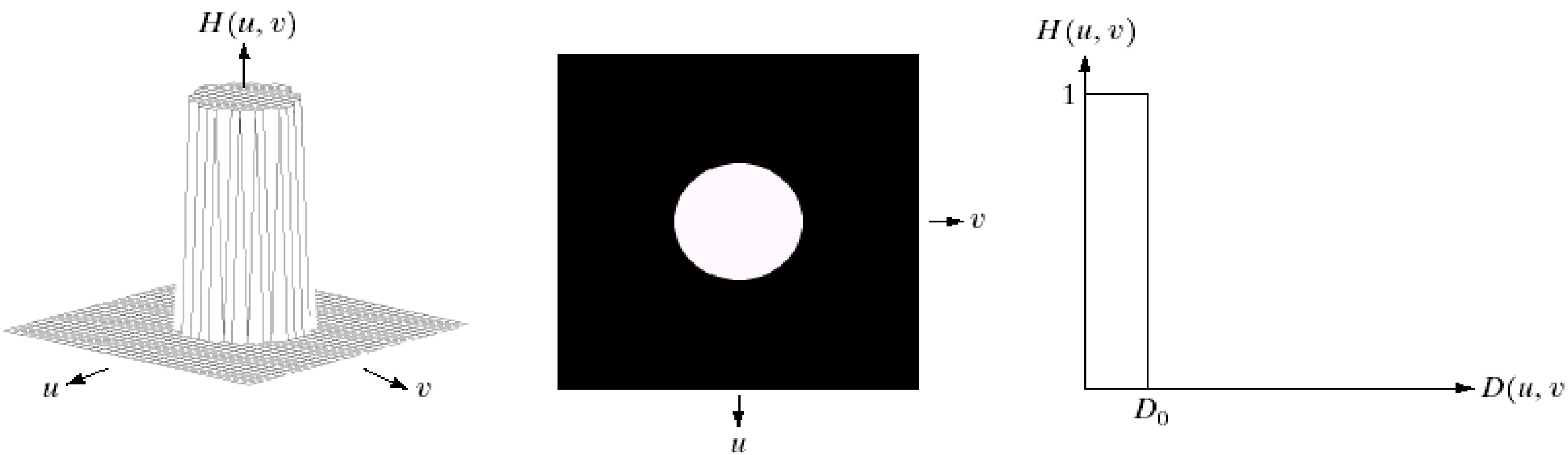
- Ideal
- Butterworth
- Gaussian

Noise Removal

- Band bass/reject
- Notch

Ideal Lowpass Filter (ILPF)

- Cuts off all high frequency components that are a specified distance D_0 from the origin of the transform.



Ideal Lowpass Filter (ILPF) – (cont.)

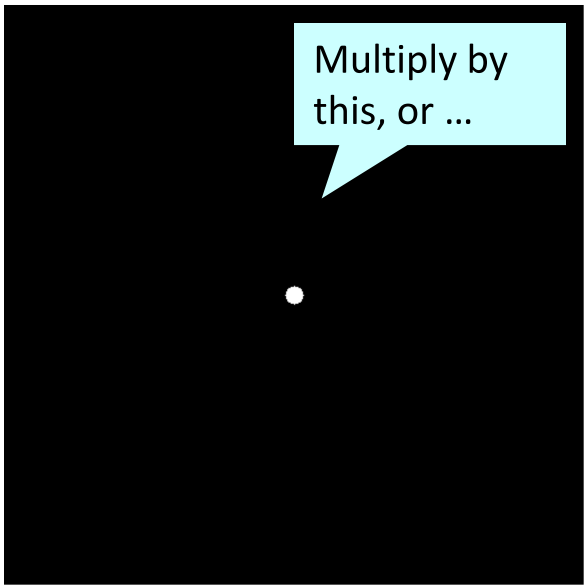
- **The transfer function is given by:**

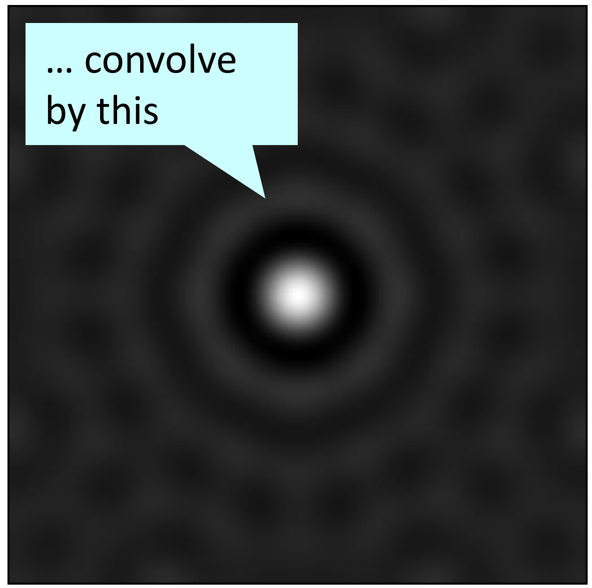
$$H(u,v) = \begin{cases} 1 \;\text{ if }\; D(u,v) \le D_0 \\ 0 \;\text{ if }\; D(u,v) > D_0 \end{cases}$$

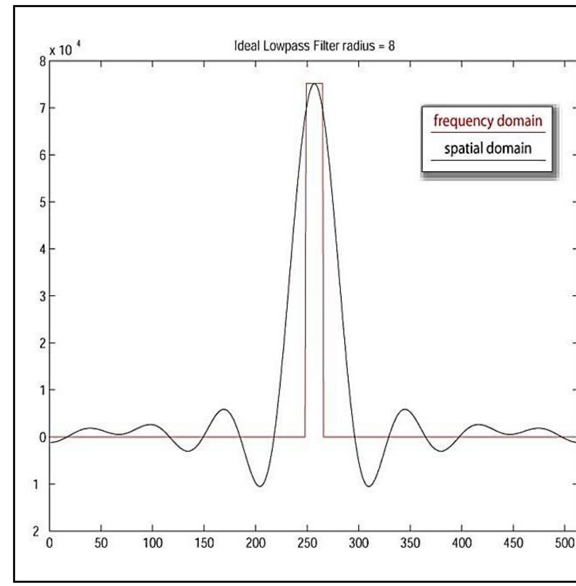
- **Changing the distance changes the effect of the filter.**
- **Point of transition between $H(u,v) = 1$ and $H(u,v) = 0$ is called *cutoff frequency*.**

Ideal Lowpass Filter (ILPF) – (cont.)

Image size: 512x512
FD filter radius: 8



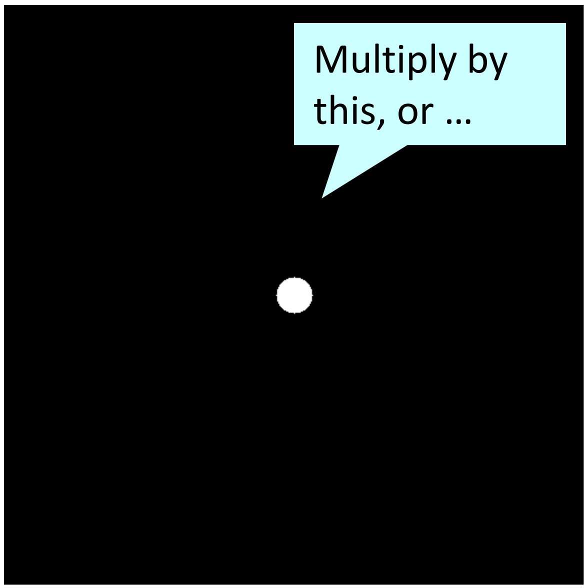




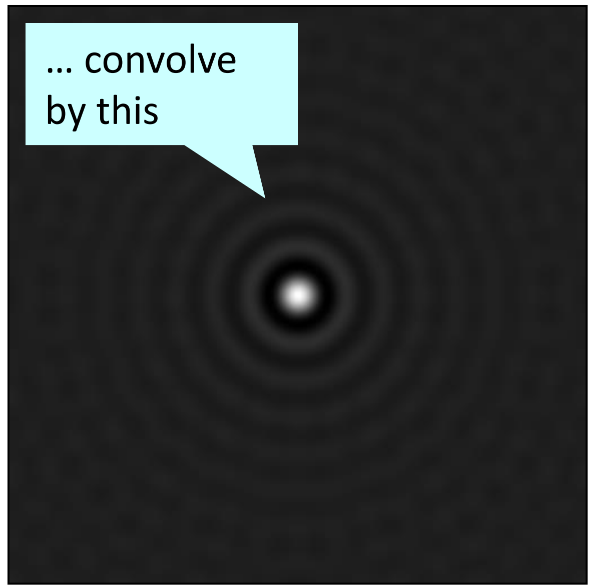
Fourier Domain Rep. | Spatial Representation | Central Profile

Ideal Lowpass Filter (ILPF) – (cont.)

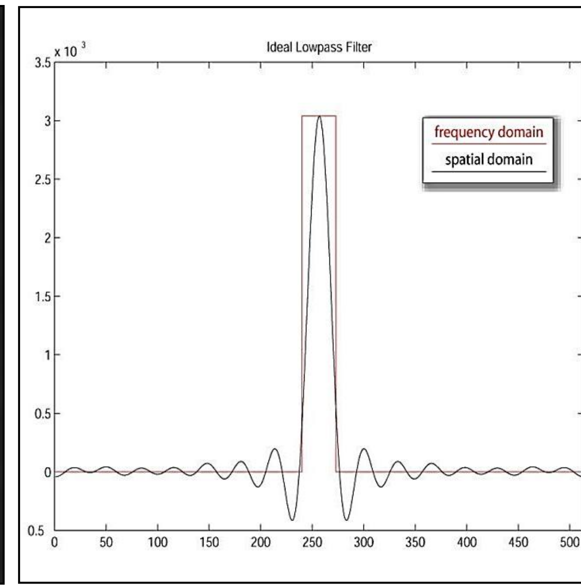
Image size: 512x512
FD filter radius: 16



Fourier Domain Rep.



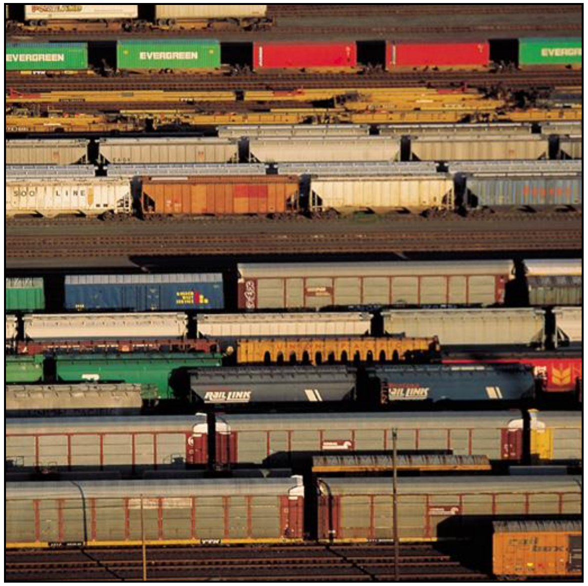
Spatial Representation

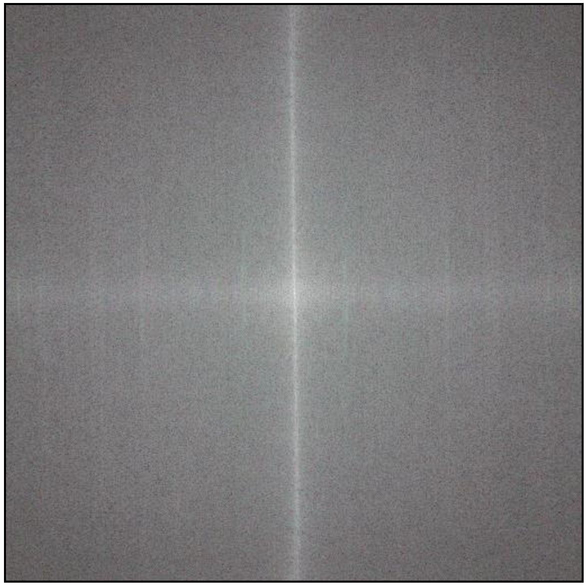


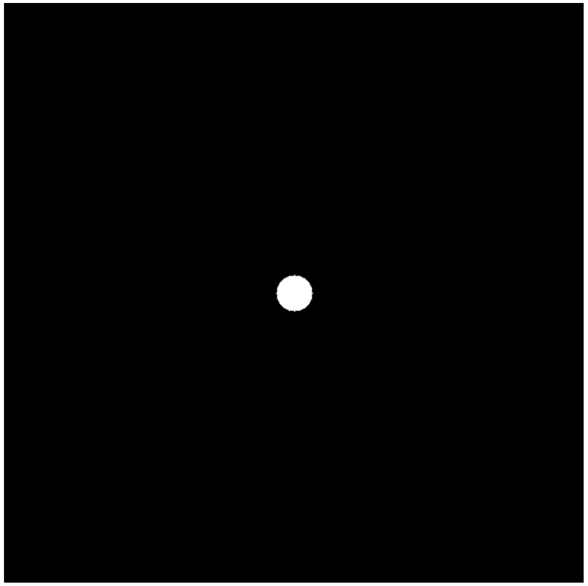
Central Profile

Ideal Lowpass Filter (ILPF) – (cont.)

Image size: 512x512
FD filter radius: 16







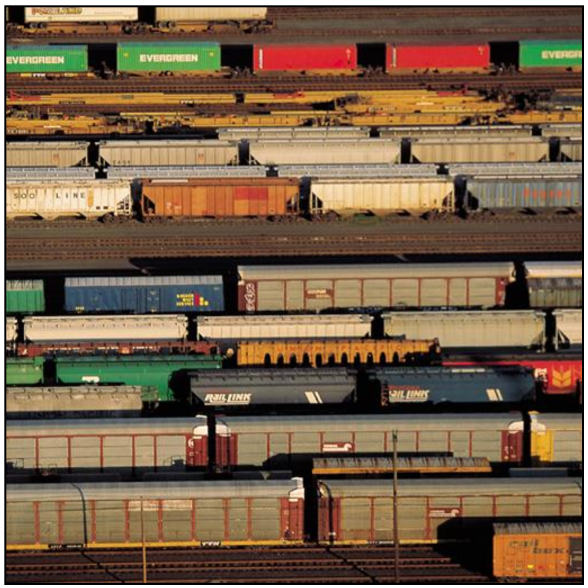
Original Image

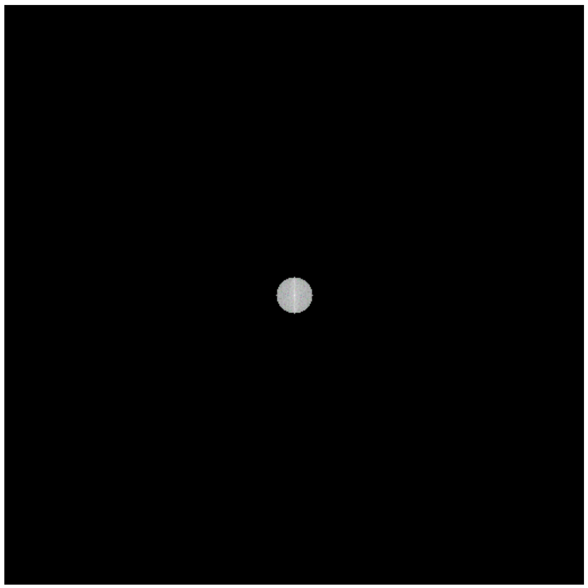
Power Spectrum

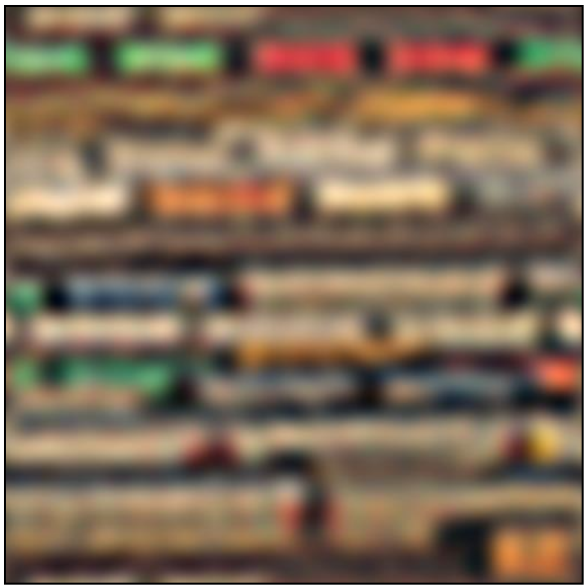
Ideal LPF in FD

Ideal Lowpass Filter (ILPF) – (cont.)

Image size: 512x512
FD filter radius: 16







Original Image

Filtered Power Spectrum

Filtered Image

Ideal Lowpass Filter (ILPF) – (cont.)

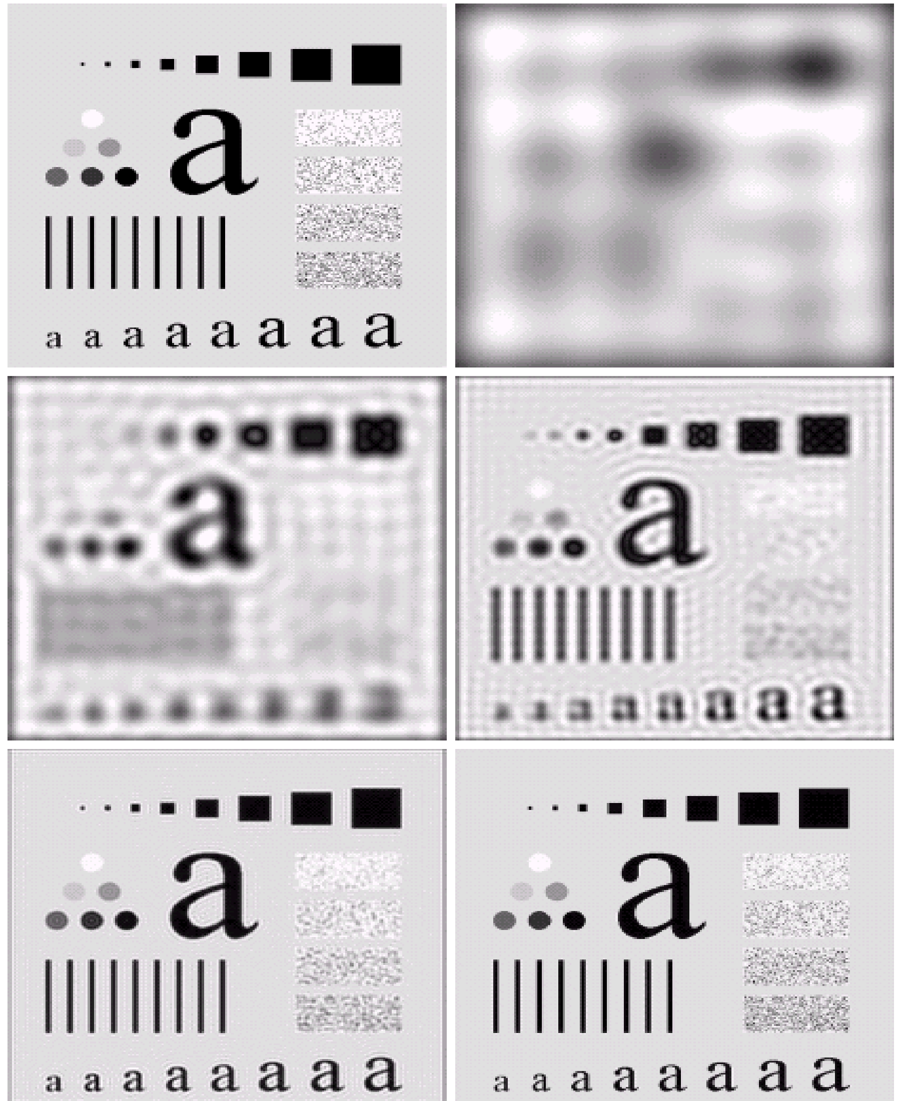
Original image

$D_0 = 10$

$D_0 = 30$

-Ringing effect

$D_0 = 60$

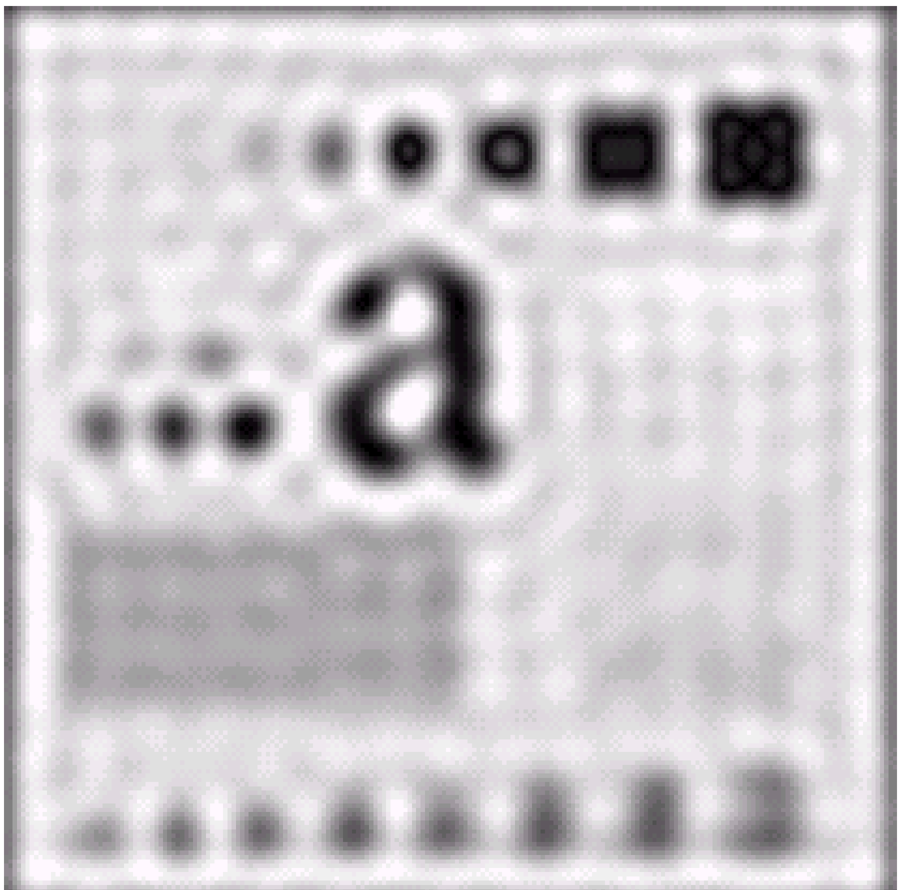


$D_0 = 160$

$D_0 = 460$

Ideal Lowpass Filter (ILPF) – (cont.)

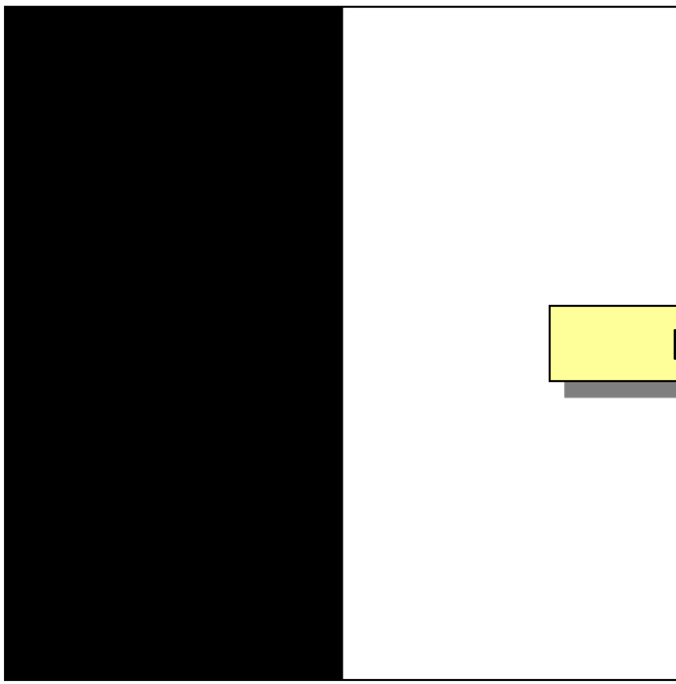
-Ringing effect

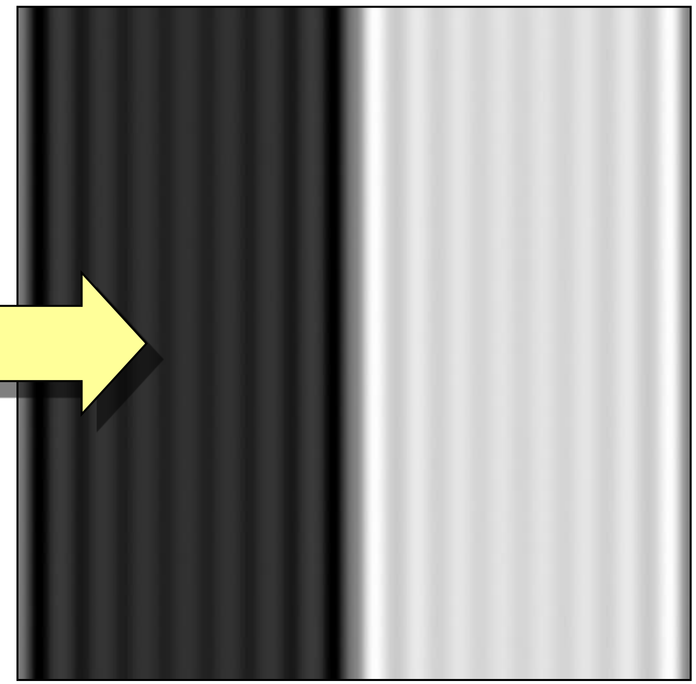


Result of filtering with ideal low pass filter with cutoff 30

Ideal Lowpass Filter (ILPF) – (cont.)

Ideal Filters Do Not Produce Ideal Results



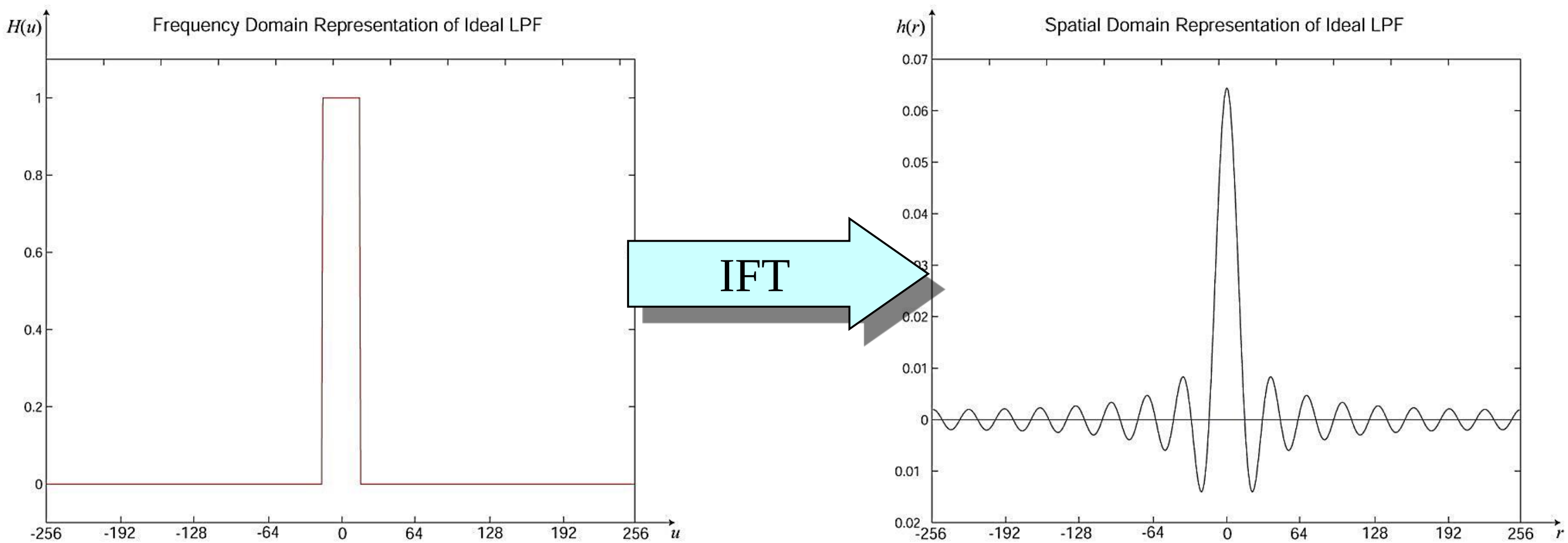


Ideal LPF

Blurring the image above w/ an ideal lowpass filter…

…distorts the results with ringing or ghosting.

Ideal Lowpass Filter (ILPF) – (cont.)



A sharp cutoff in the frequency domain…

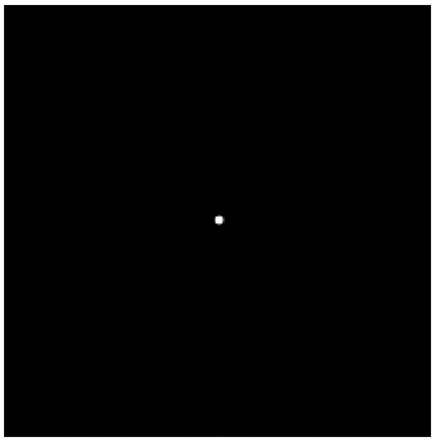
…causes ringing in the spatial domain.

Ideal Lowpass Filter (ILPF) – (cont.)

- **Ringing Effect** $\delta(x,y) * h(x,y) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\delta(m,n)h(x-m,y-n) = \frac{1}{MN}h(x,y)$

ILPF

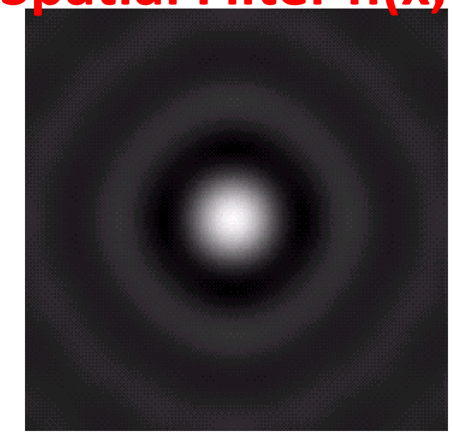
(a)

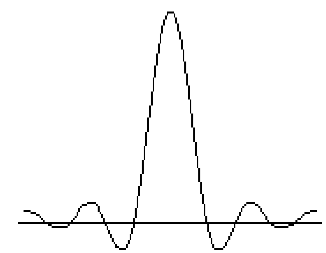


IDFT

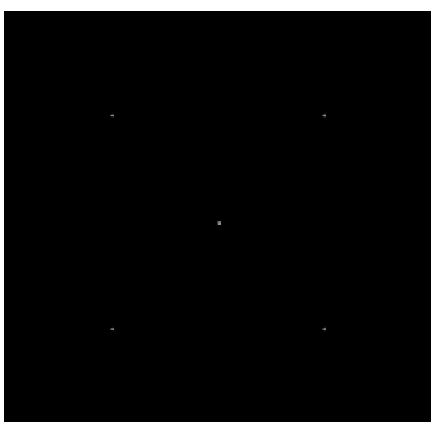
Spatial Filter h(x,y)

(b)



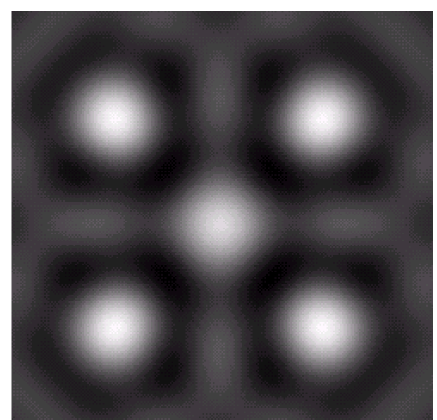


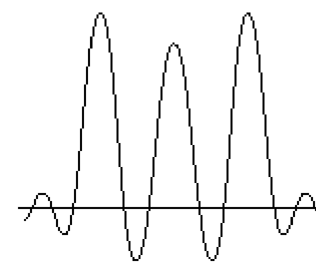
(c)



(c) ★(b)

(d)





Five spatial impulses δ(x,y)

δ(x,y) ★h(x,y)

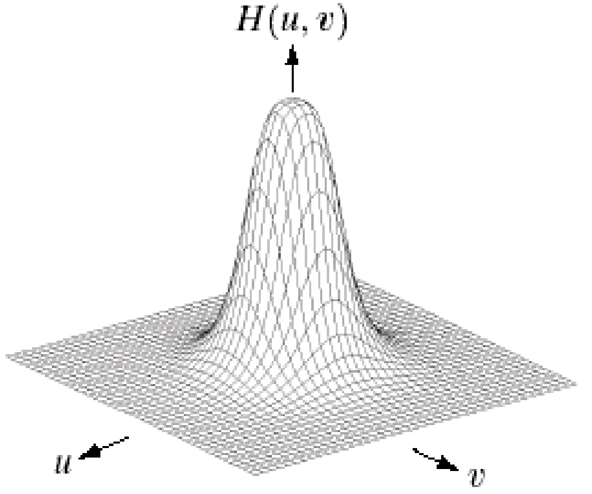
Ideal Lowpass Filter (ILPF) – (cont.)

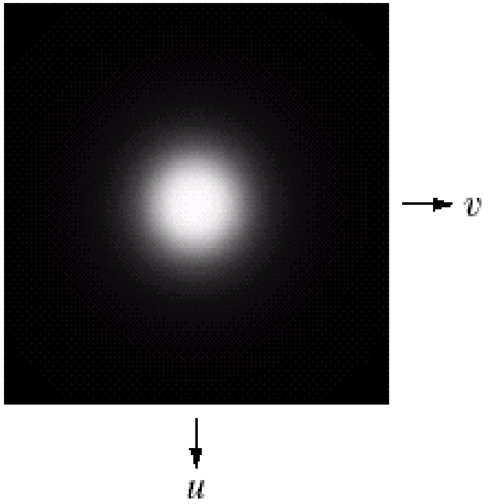
Remarks

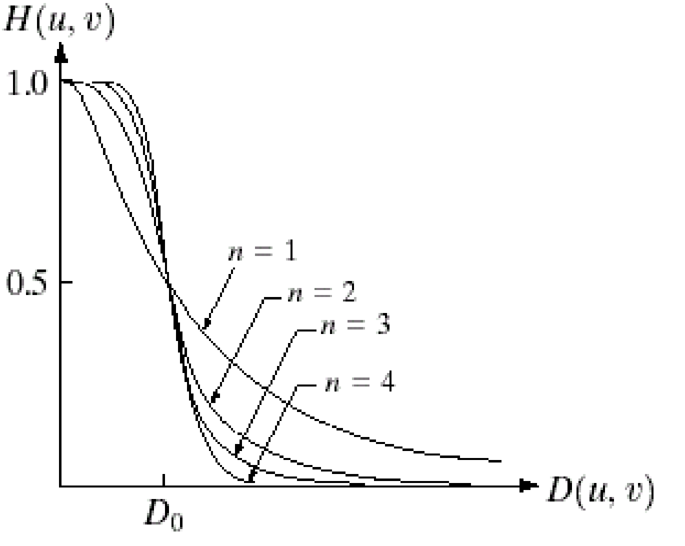
- **Ringing is a characteristic of the ILPF.**
- **Increasing radius → less power removed.**
- **Most sharp details are contained in the 13% power removed.**
- **Smaller D_0 == large spread of sinc == more blurring.**
- **Larger D_0 == narrower sinc (nearly an impulse) == no blurring.**

Butterworth Lowpass Filter (BLPF)

- Does not have the sharp discontinuity of the ideal filter.
- Represents the transition between two extremes.



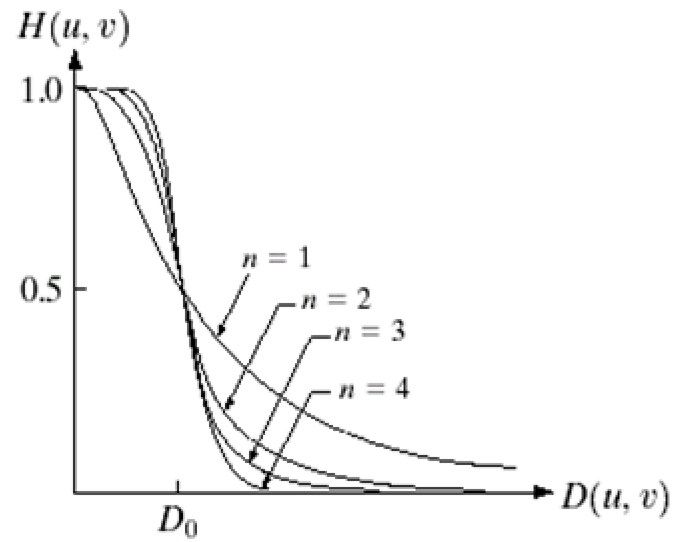




Butterworth Lowpass Filter (BLPF)

- The transfer function is given by:

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



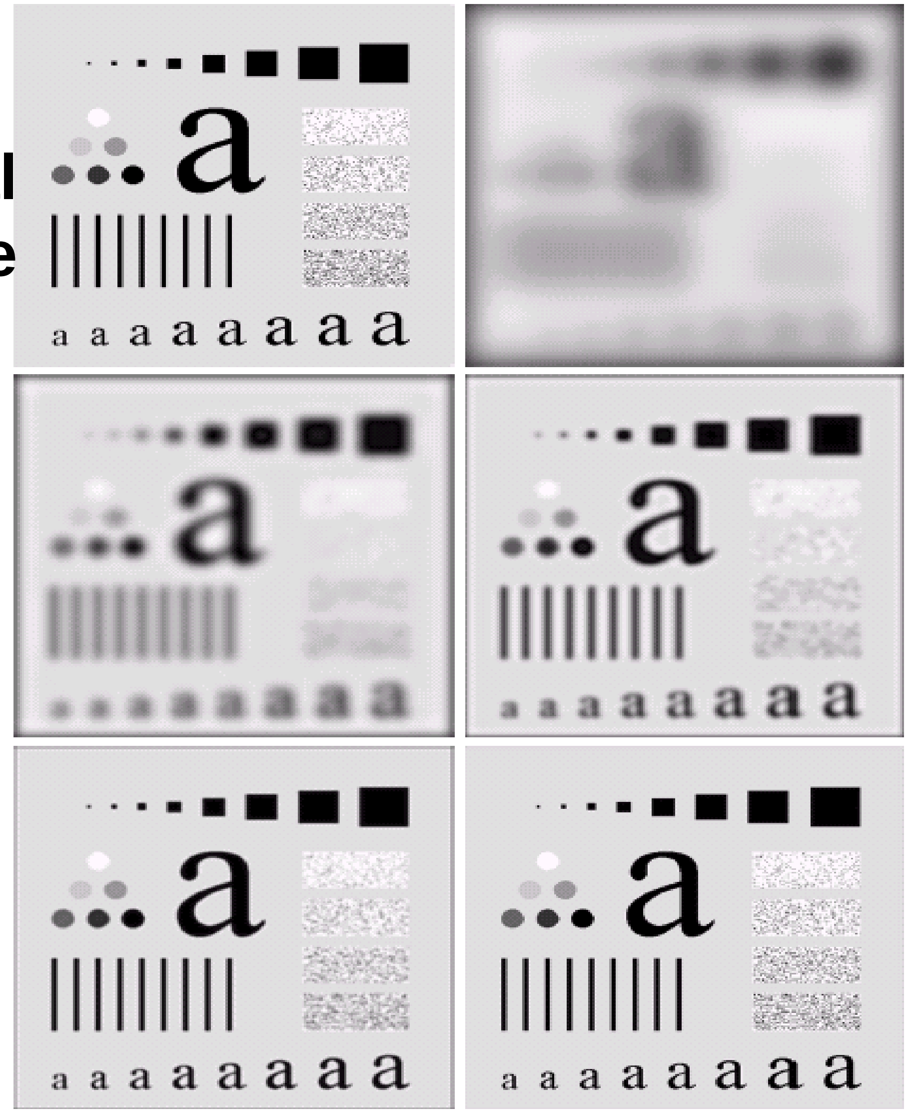
- The order n of the filter controls the "degree" of smoothness.
  - Higher order → sharper cutoff ~~ ideal.
  - Lower order → smoother cutoff ~~ Gaussian.

Butterworth Lowpass Filter (BLPF)

- n = 2

- Smoother transitions

- less ringing effect than ideal filter.



Original image

$D_0 = 10$

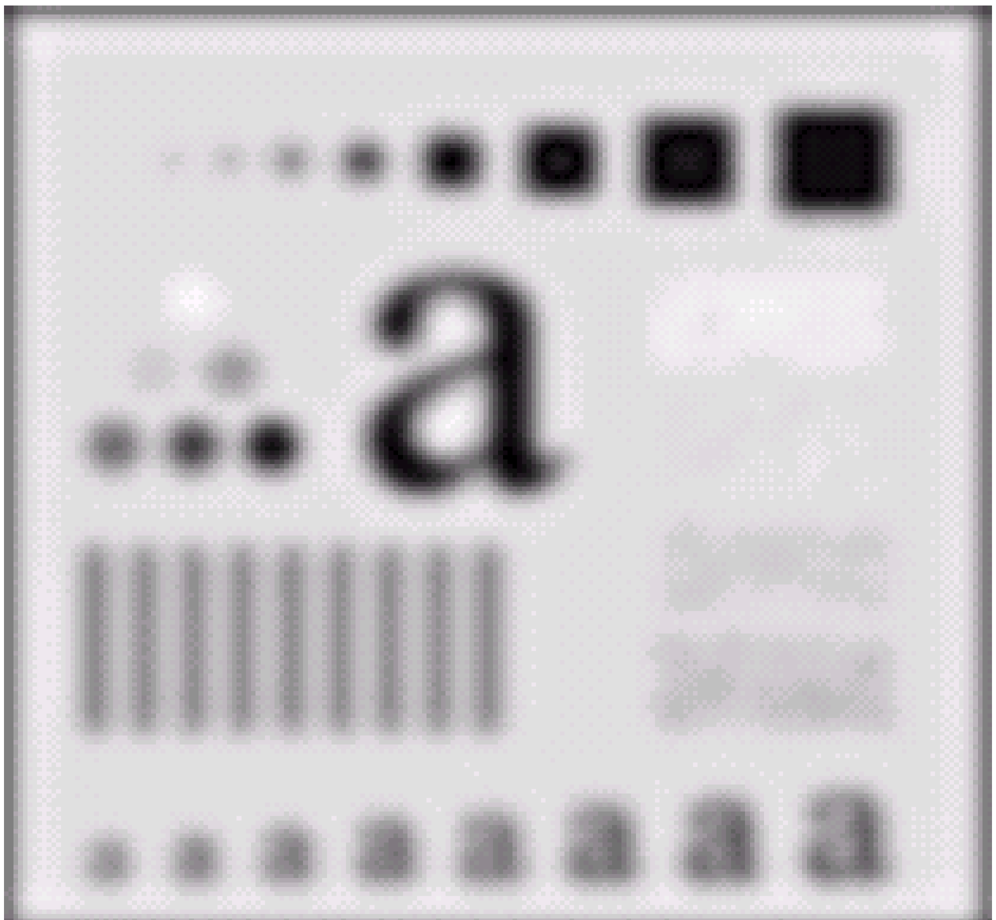
$D_0 = 30$

$D_0 = 60$

$D_0 = 160$

$D_0 = 460$

Butterworth Lowpass Filter (BLPF)



Result of filtering with Butterworth low pass filter with cutoff 30

Butterworth Lowpass Filter (BLPF)

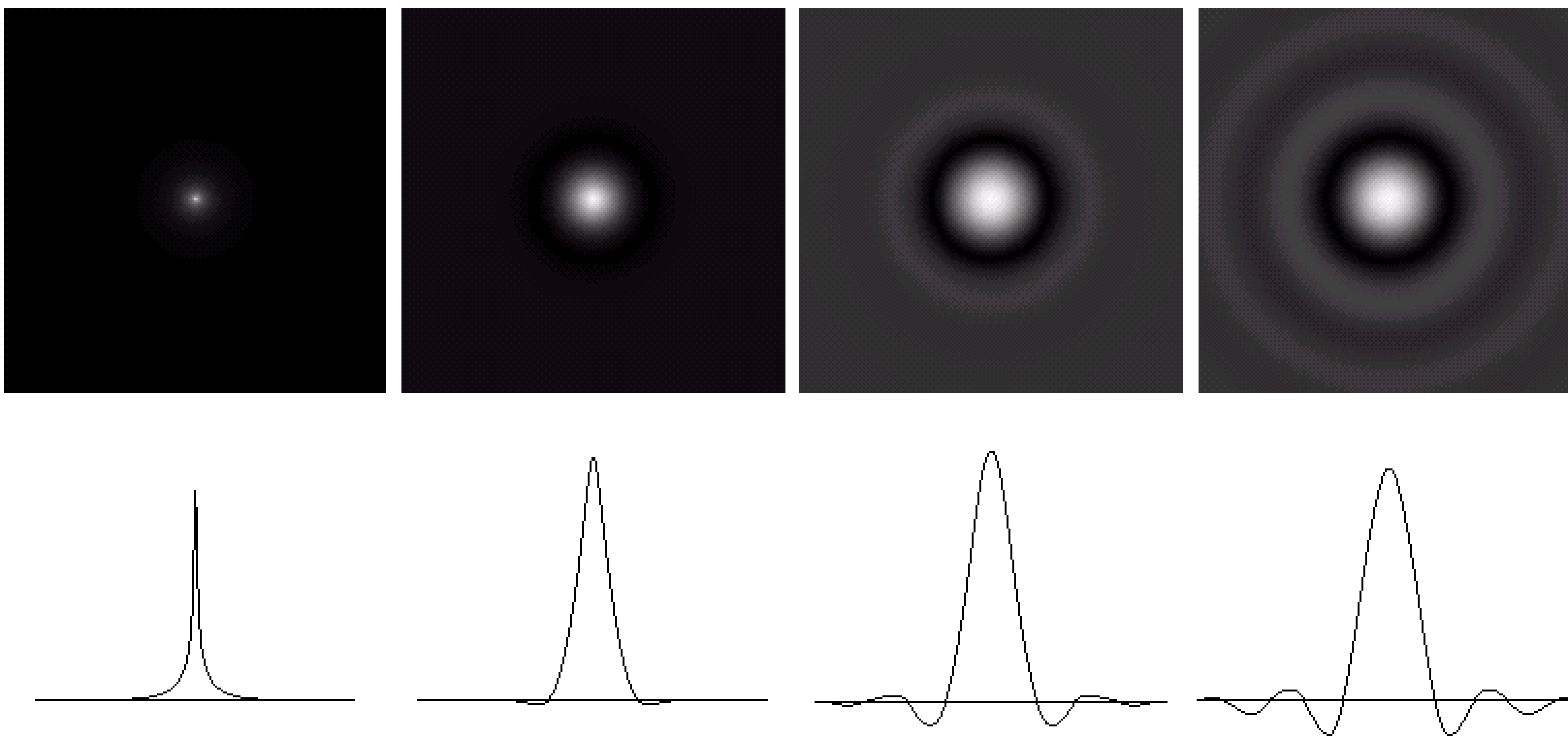
Ringing Effect

- $D_0 = 5$

Ringing effect ↑

Ideal Lowpass Filter

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



Butterworth Lowpass Filter (BLPF)

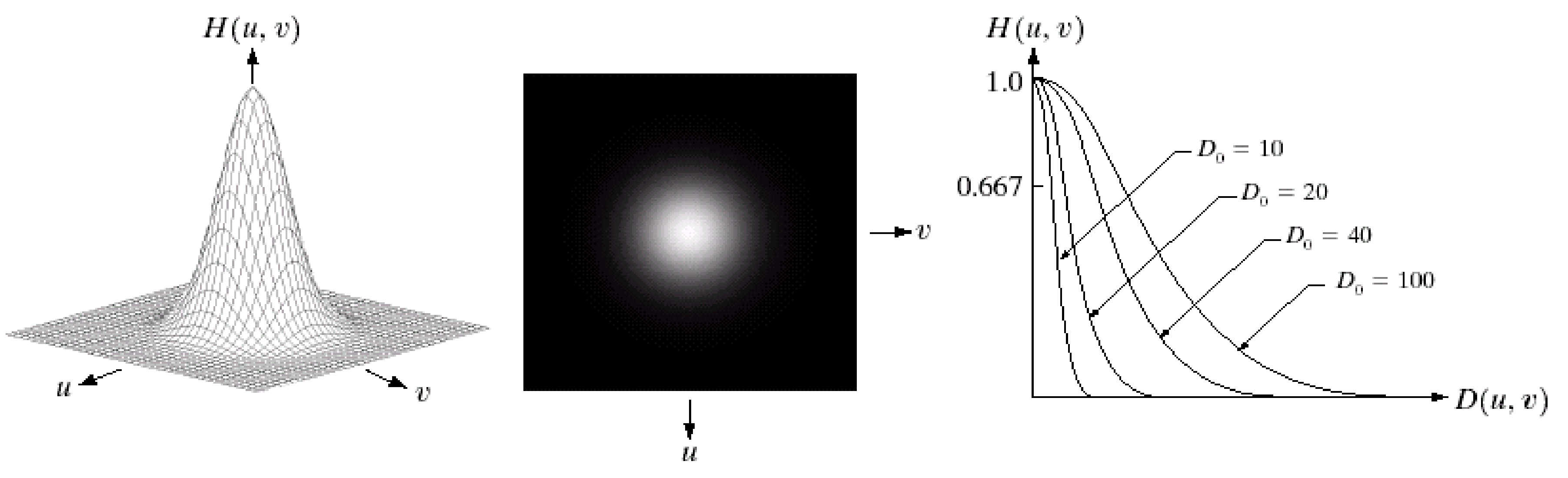
Remarks

- **No ringing effect for small order.**
- **Smoother transitions in blurring with respect to increasing cutoff.**

Gaussian Lowpass Filter (GLPF)

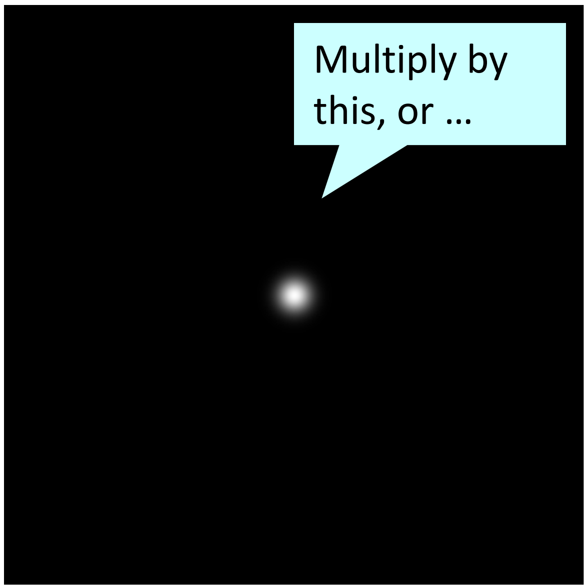
- **The transfer function is given by:**

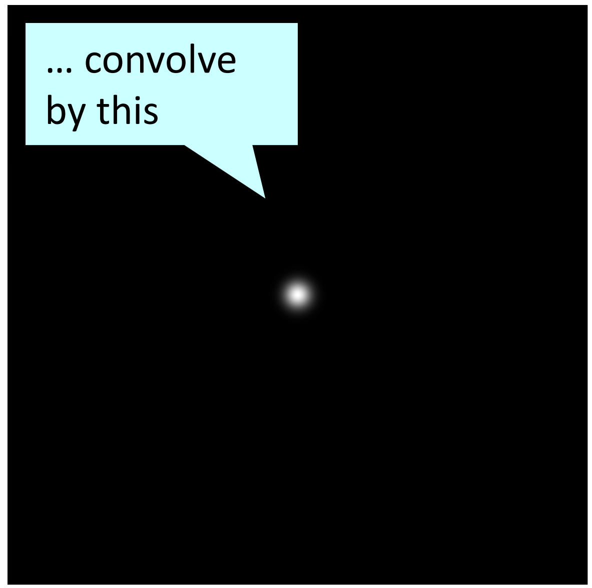
$$H(u,v) = e^{-D^2(u,v)/2D_0^{\,2}}$$

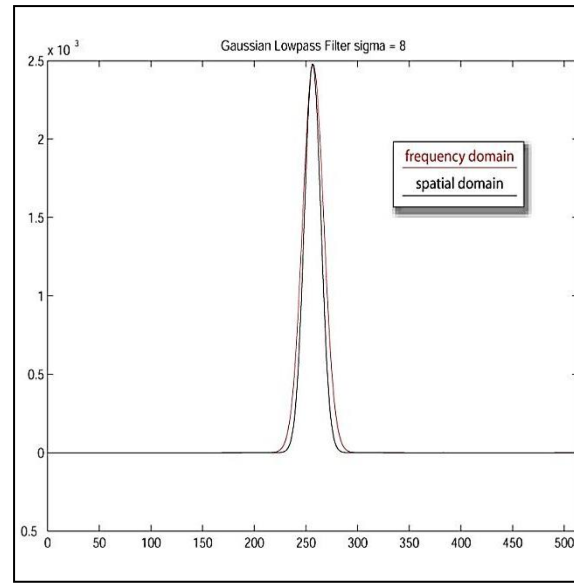


Gaussian Lowpass Filter (GLPF)

Image size: 512x512
SD filter sigma = 8







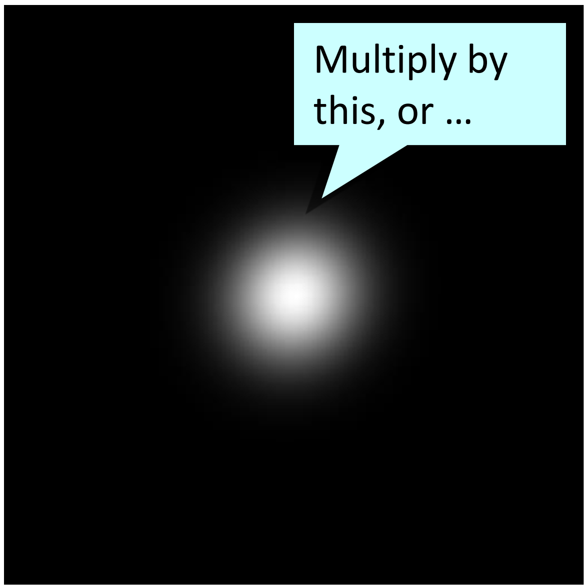
Fourier Domain Rep.

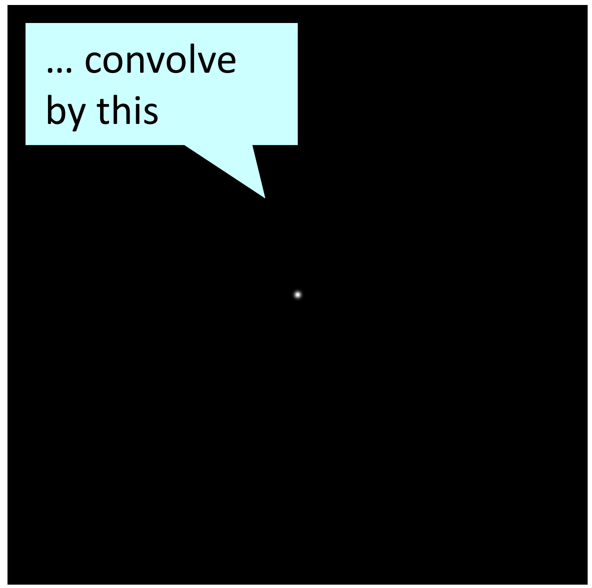
Spatial Representation

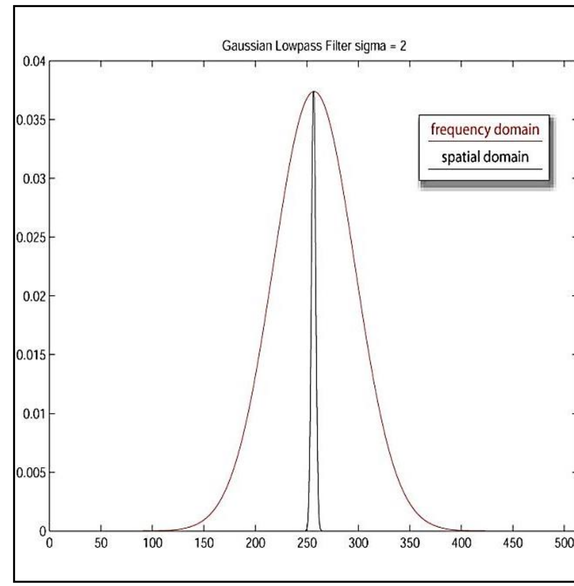
Central Profile

Gaussian Lowpass Filter (GLPF)

Image size: 512x512
SD filter sigma = 2







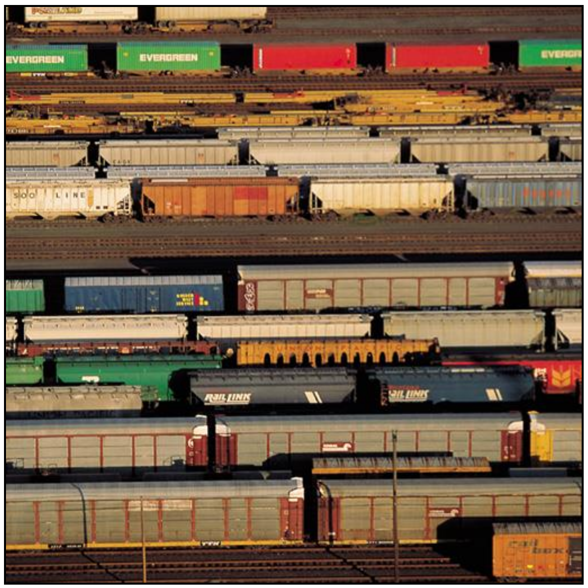
Fourier Domain Rep.

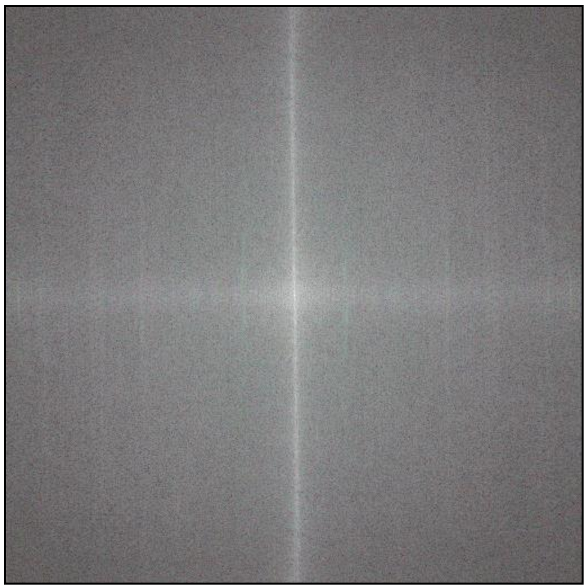
Spatial Representation

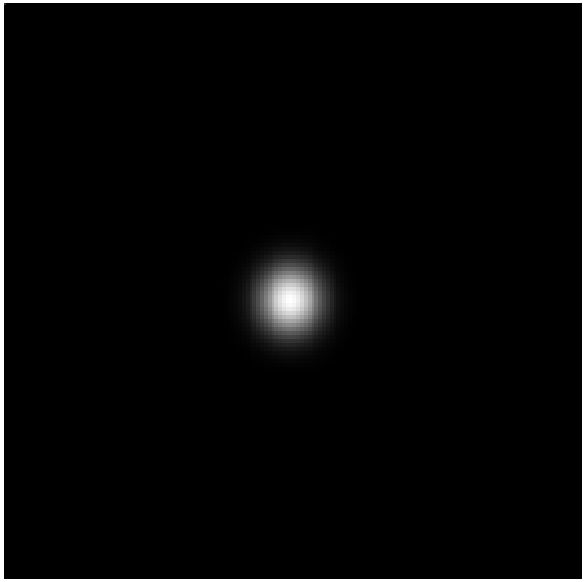
Central Profile

Gaussian Lowpass Filter (GLPF)

Image size: 512x512
SD filter sigma = 8







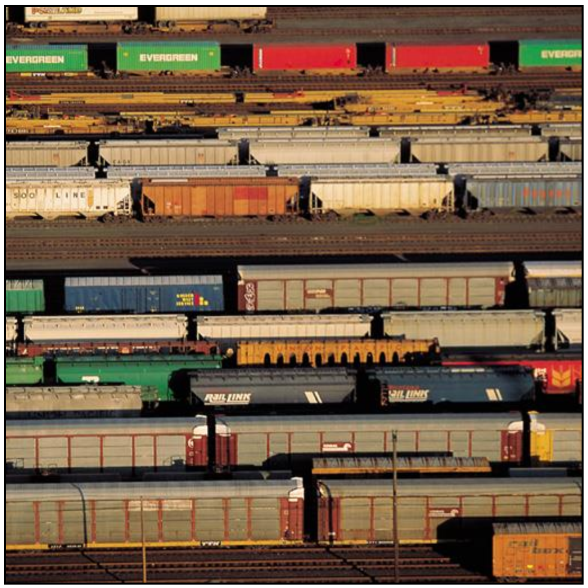
Original Image

Power Spectrum

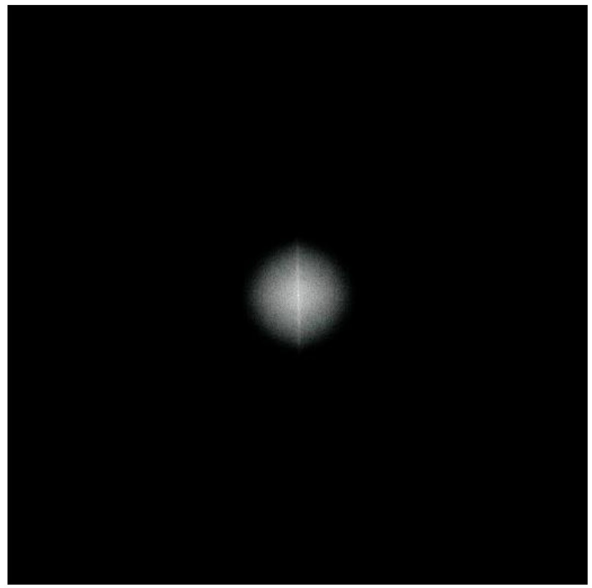
Gaussian LPF in FD

Gaussian Lowpass Filter (GLPF)

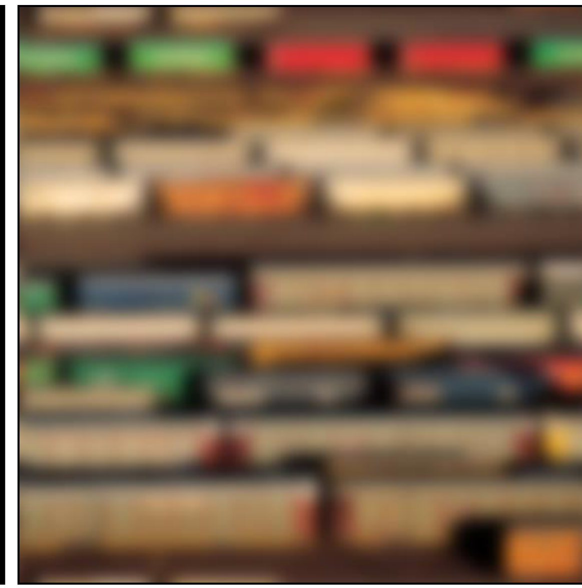
Image size: 512x512
SD filter sigma = 8



Original Image



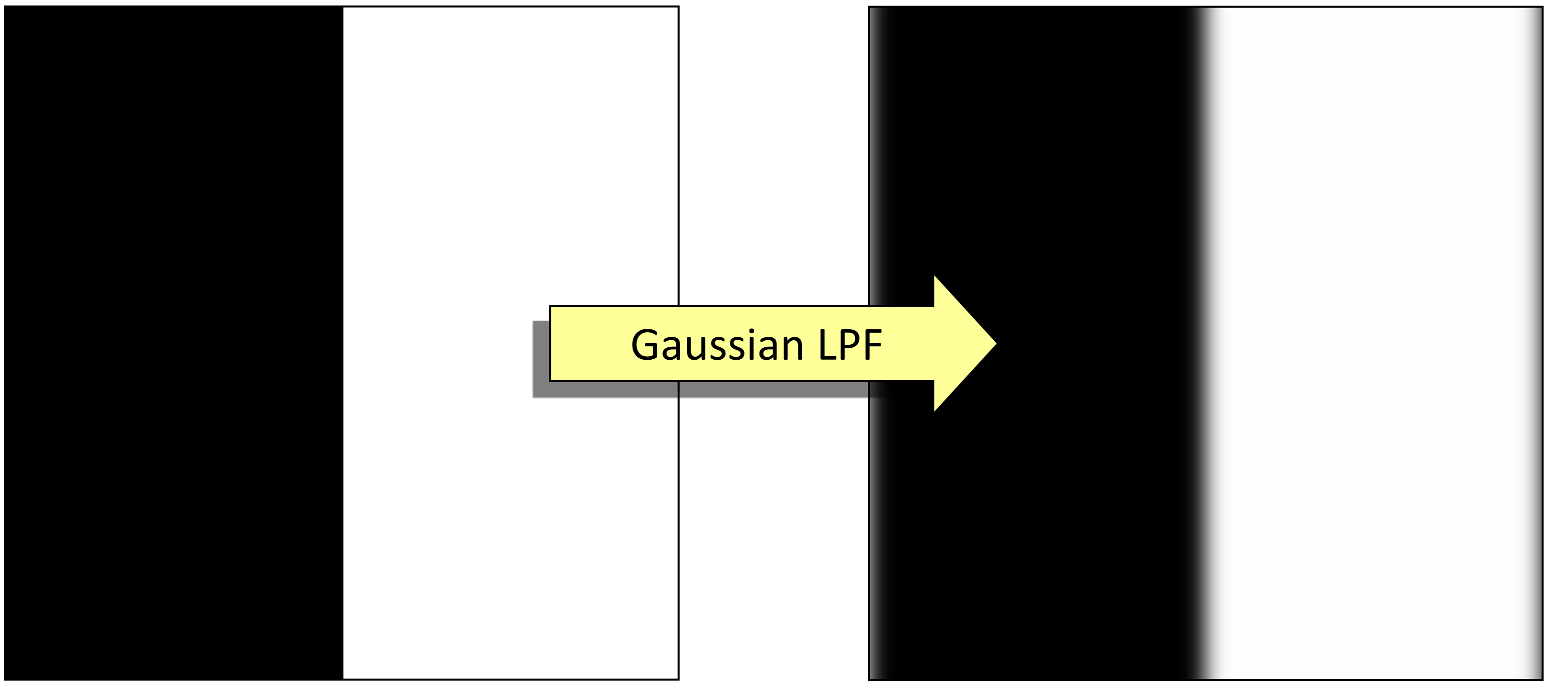
Filtered Power Spectrum



Filtered Image

Gaussian Lowpass Filter (GLPF)

Optimal Filter: The Gaussian



With a gaussian lowpass filter, the image above …

… is blurred without ringing or ghosting.

Gaussian Lowpass Filter (GLPF)

Original image

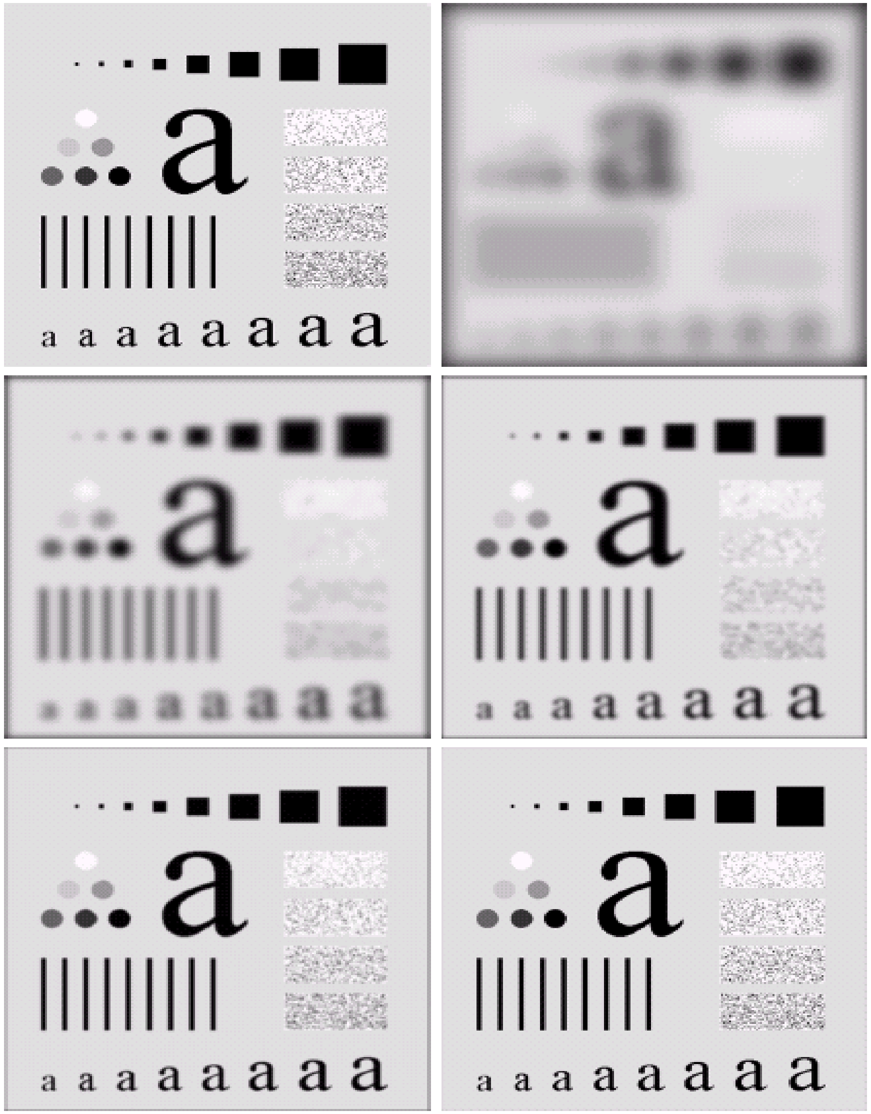
$D_0 = 10$

$D_0 = 30$

$D_0 = 60$

$D_0 = 160$

$D_0 = 460$

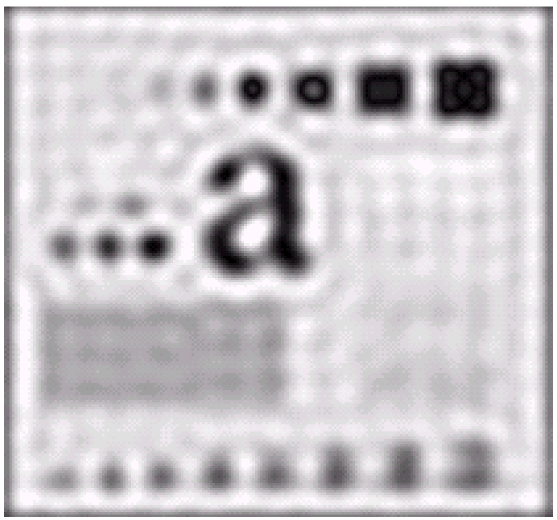


- Smoother transitions
- No ringing effect

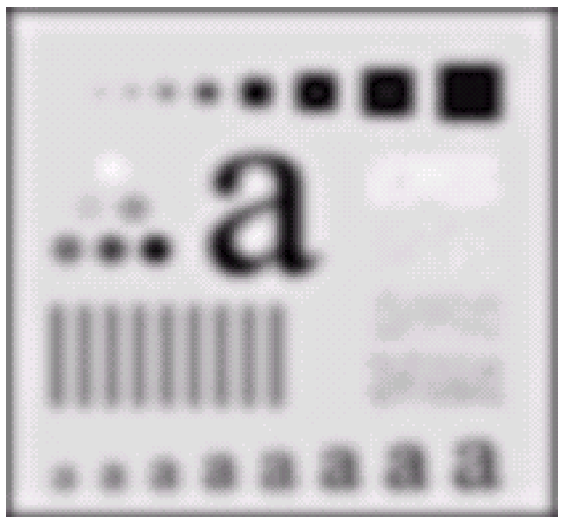
Lowpass Filters Effects Compared

- **$D_0 = 30$**

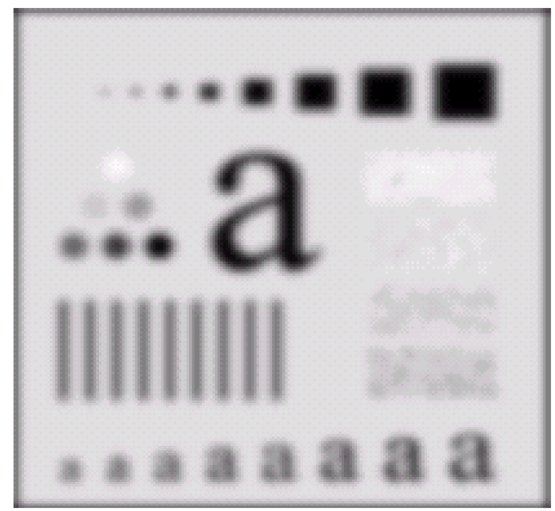
ILPF



BLPF



GLPF



Demo

Highpass Filters

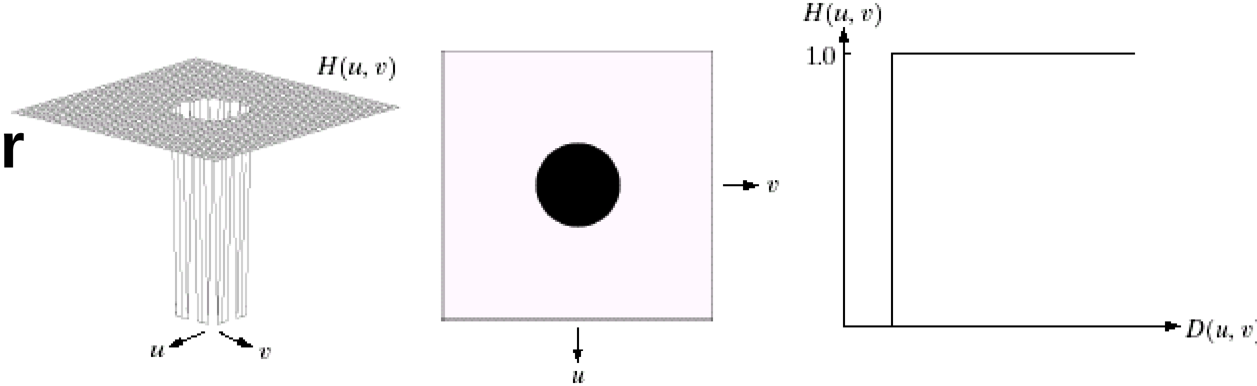
- The filters cut off (set to zero) all low frequency components that are a specified distance D_0 from the origin of the transform.

- Highpass filter can be obtained from lowpass filter by:

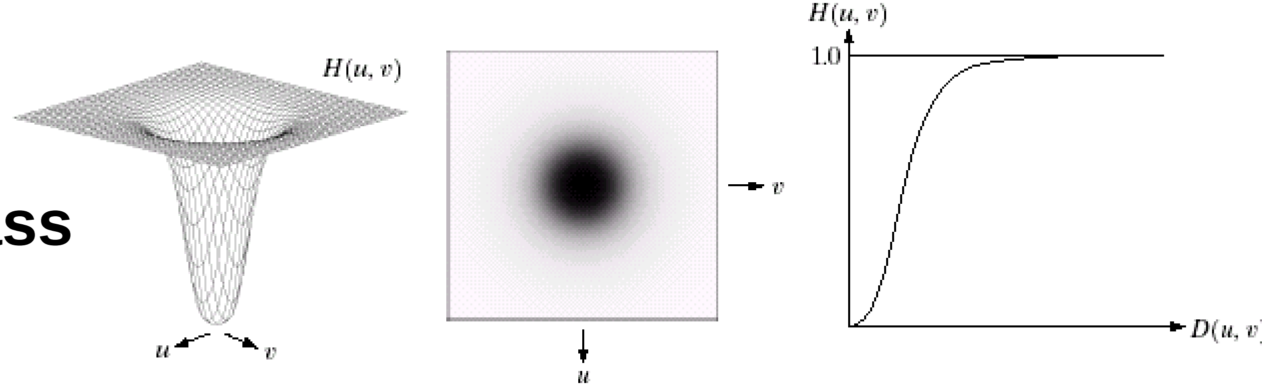
$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

Highpass Filters – (cont.)

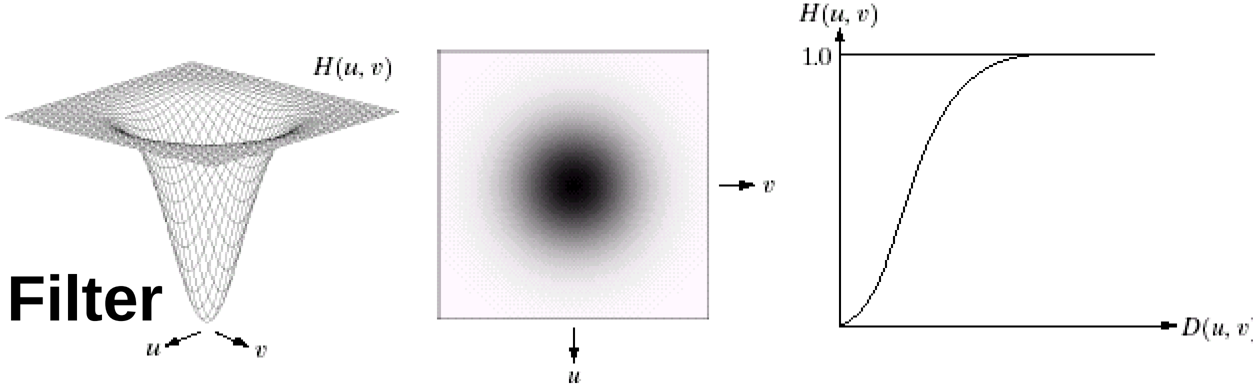
Ideal Highpass Filter



Butterworth Highpass Filter



Gaussian Highpass Filter

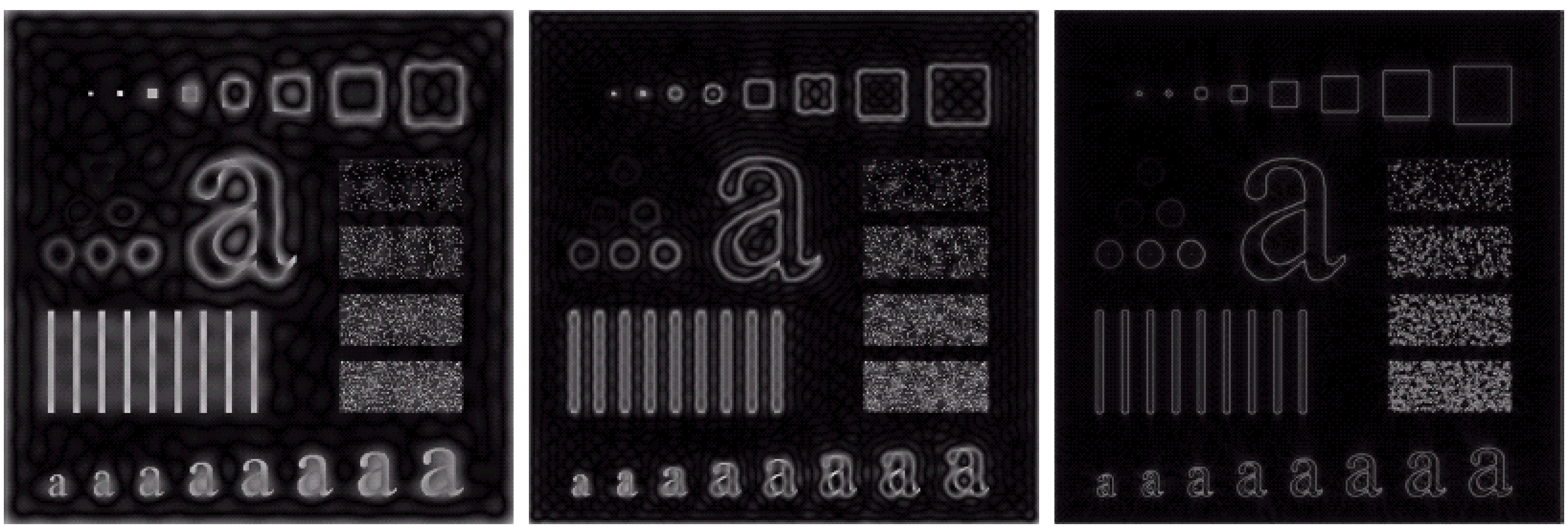


Ideal Highpass Filter (IHPF)

- **Transfer function is given by:**

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

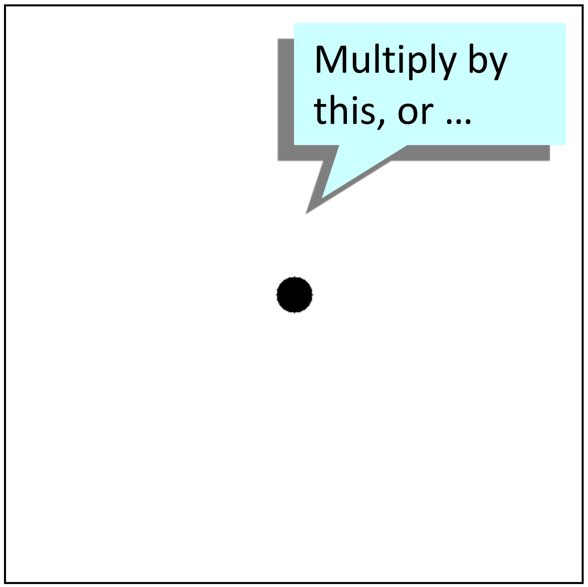
D_0 =30 **D_0 =60** **D_0 =160**

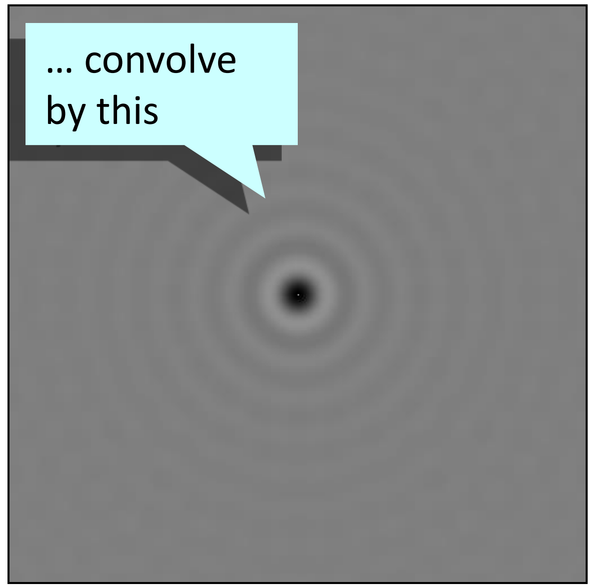


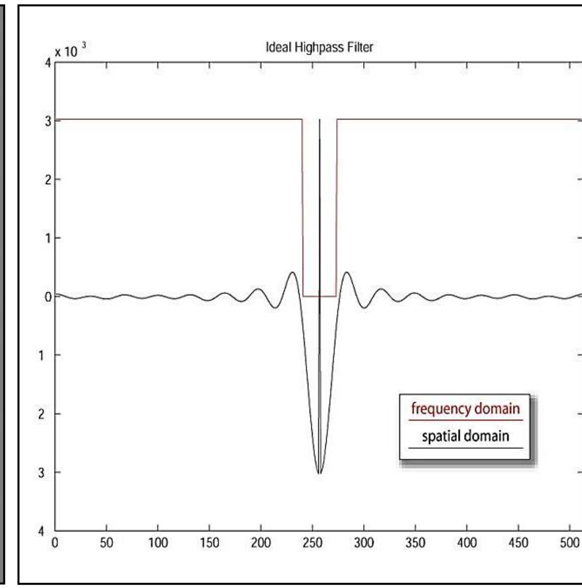
Ringing effect

Ideal Highpass Filter (IHPF)

Image size: 512x512
FD notch radius: 16







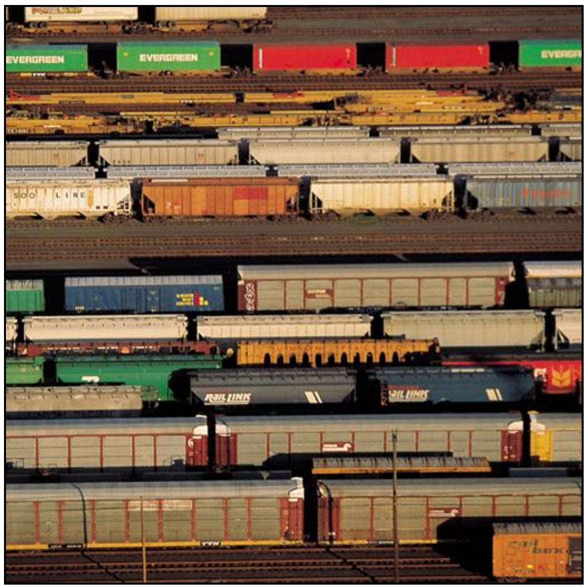
Fourier Domain Rep.

Spatial Representation

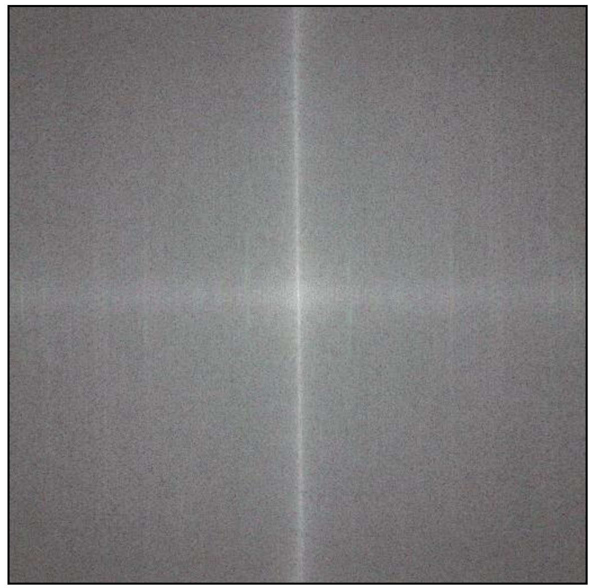
Central Profile

Ideal Highpass Filter (IHPF)

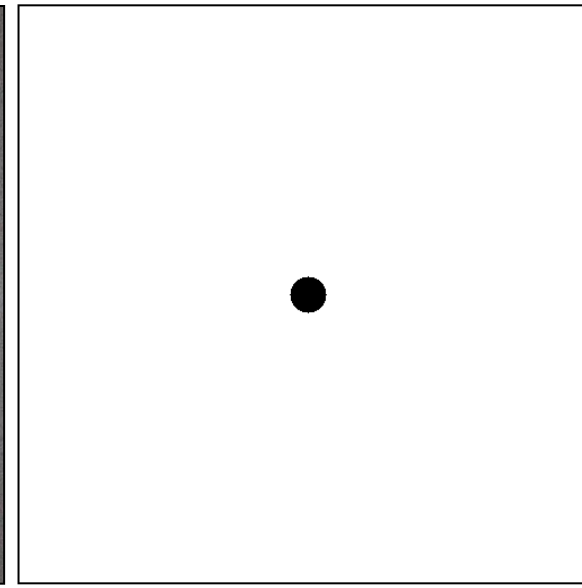
Image size: 512x512
FD notch radius: 16



Original Image



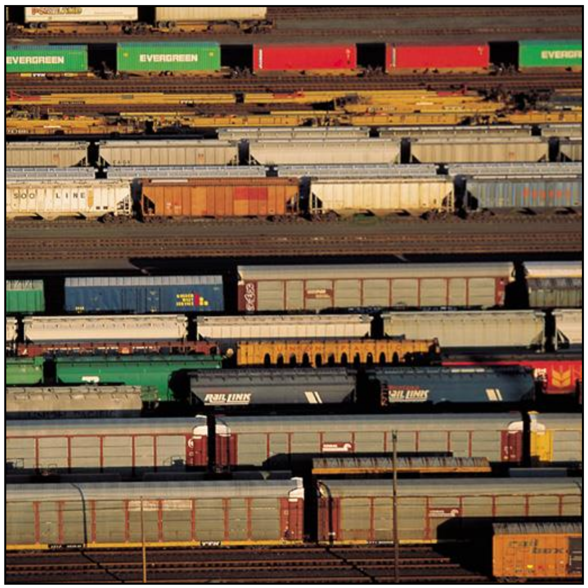
Power Spectrum



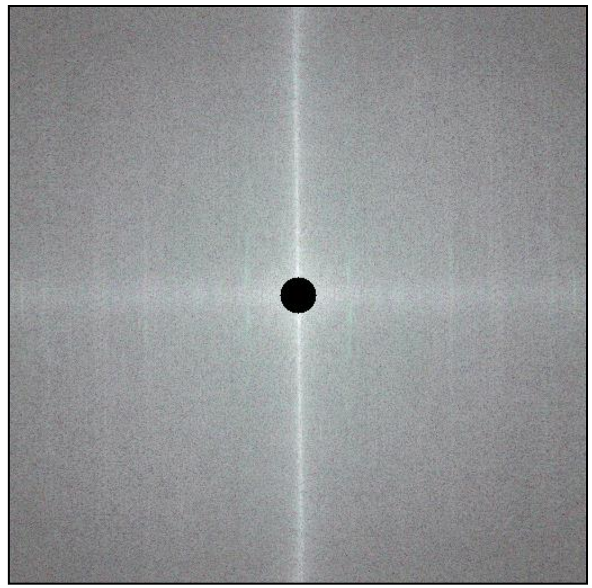
Ideal HPF in FD

Ideal Highpass Filter (IHPF)

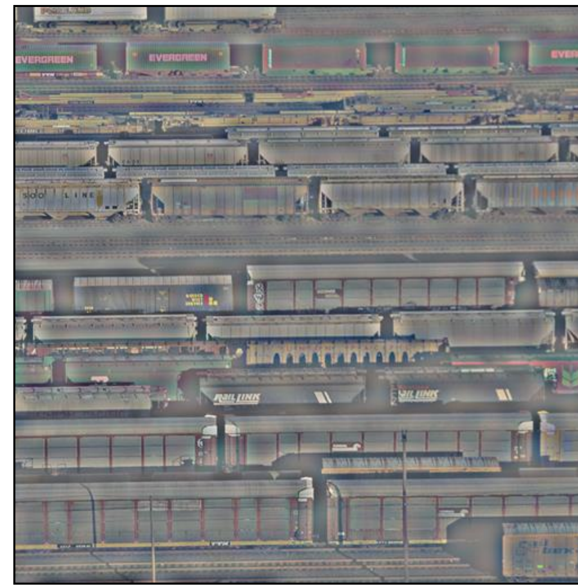
Image size: 512x512
FD notch radius: 16



Original Image



Filtered Power Spectrum



Filtered Image*

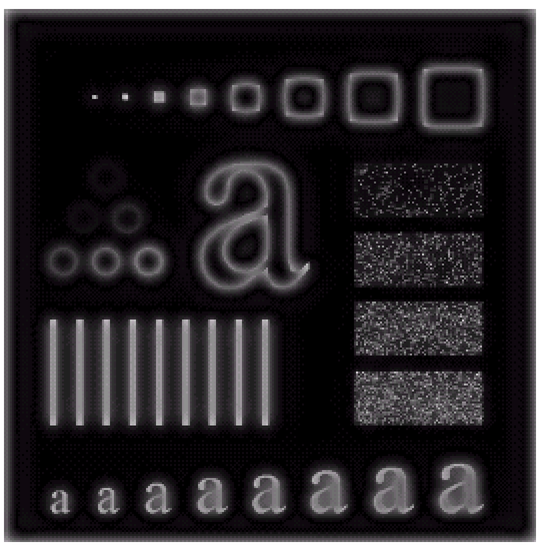
*signed image:
0 mapped to 128

Butterworth Highpass Filter (BHPF)

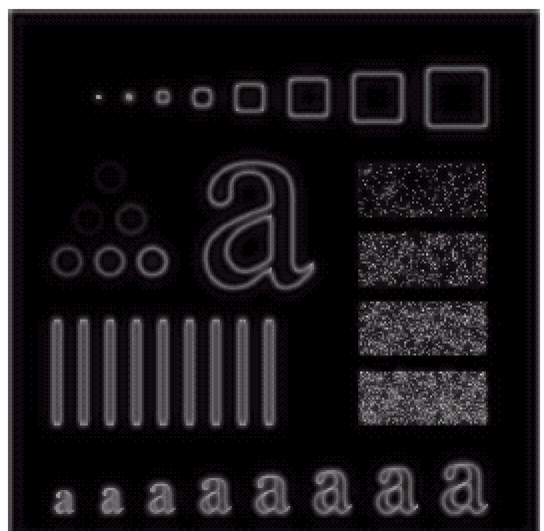
- **Transfer function is given by:**

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

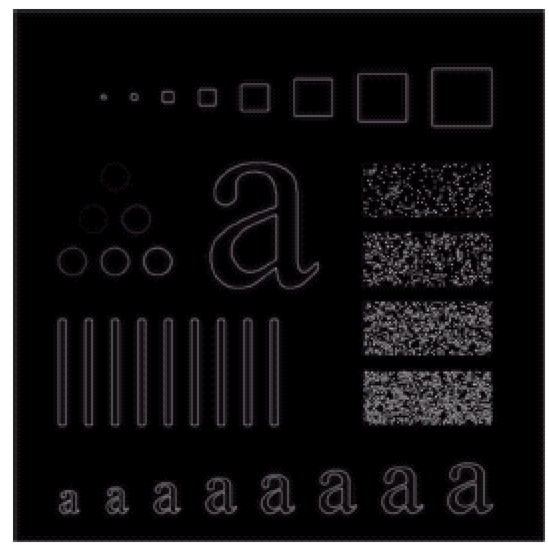
D_0 =30



D_0 =60



D_0 =160



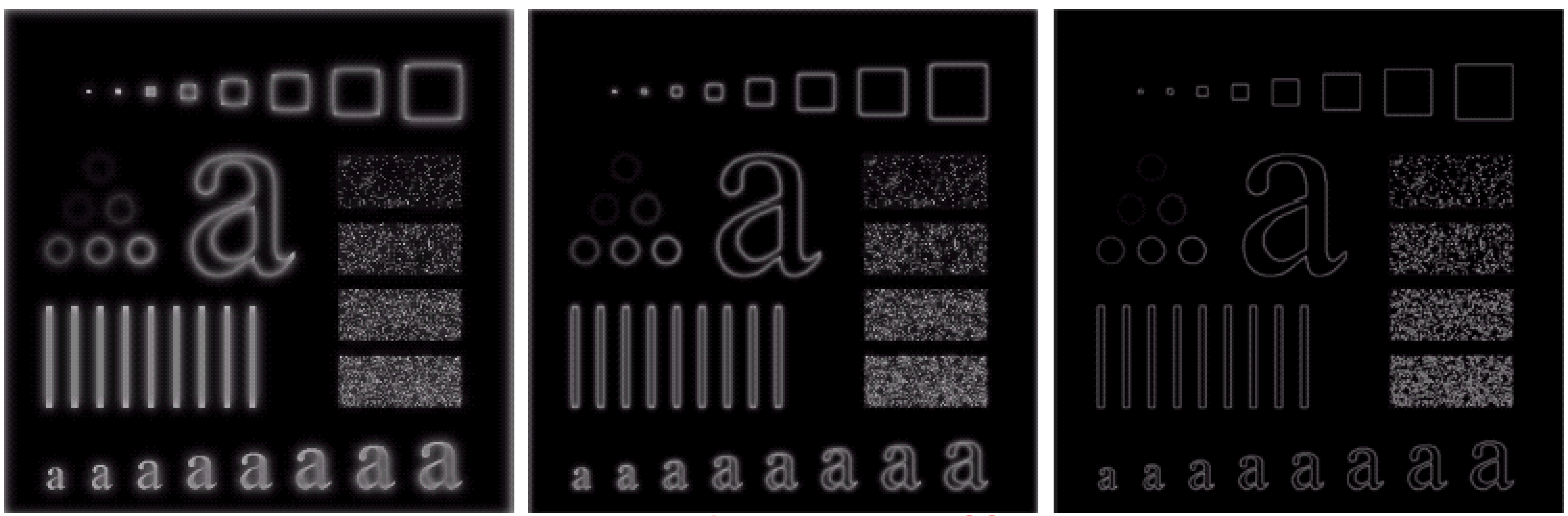
Smoother cutoff

Gaussian Highpass Filter (GHPF)

- **Transfer function is given by:**

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

D_0 =30 **D_0 =60** **D_0 =160**

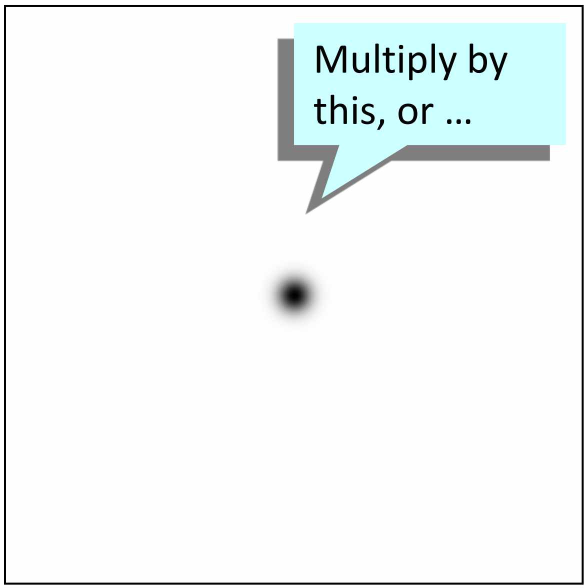


Smoother cutoff

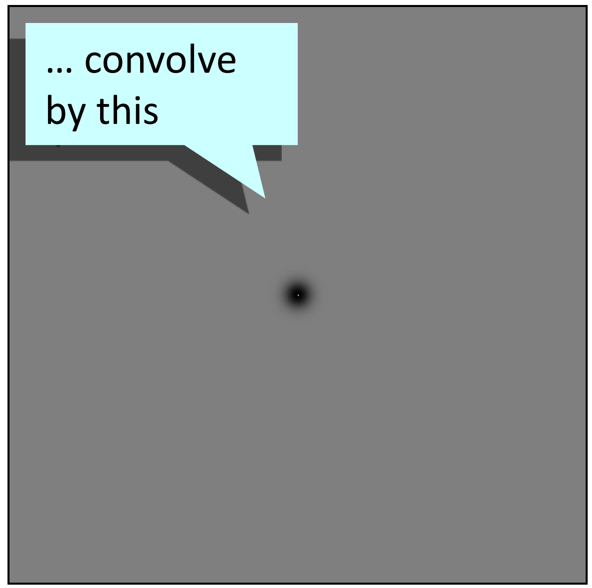
Gaussian Highpass Filter (GHPF)

Gaussian Highpass Filter

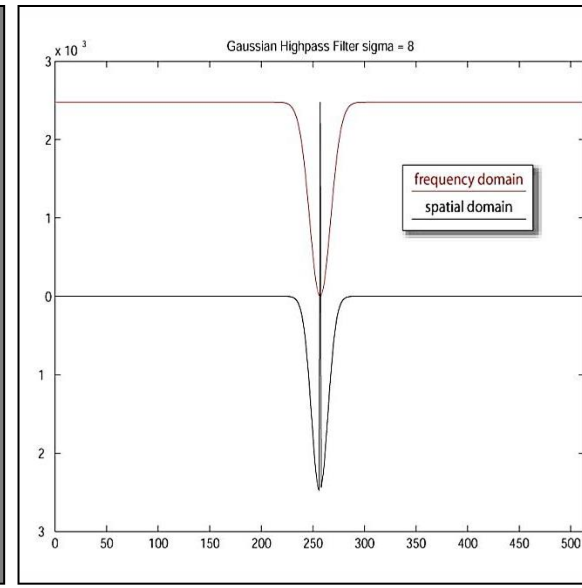
Image size: 512x512
FD notch sigma = 8



Fourier Domain Rep.



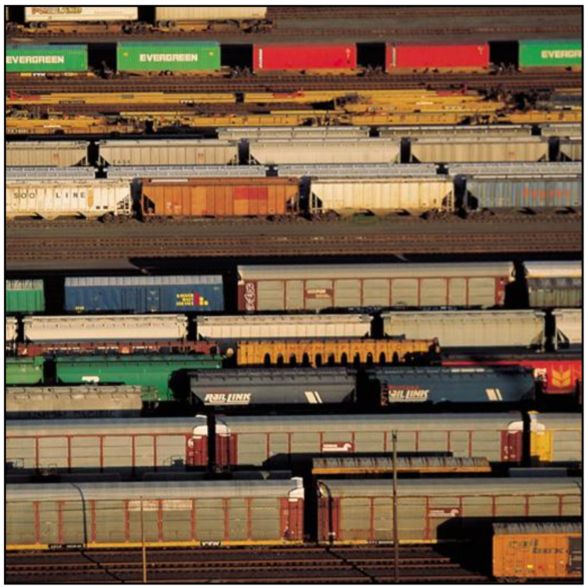
Spatial Representation

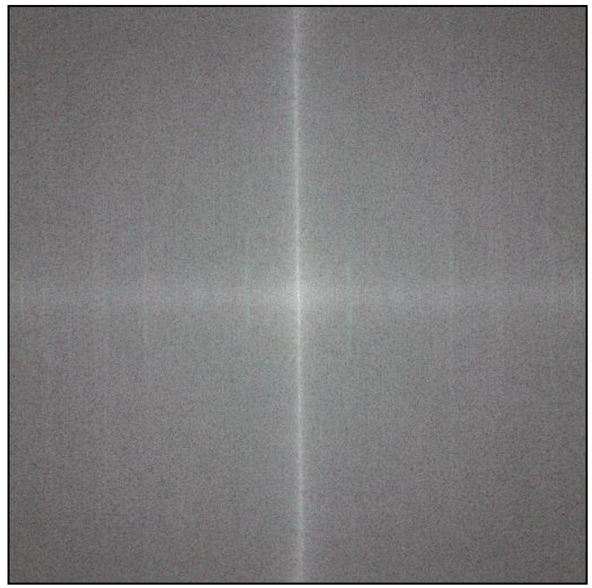


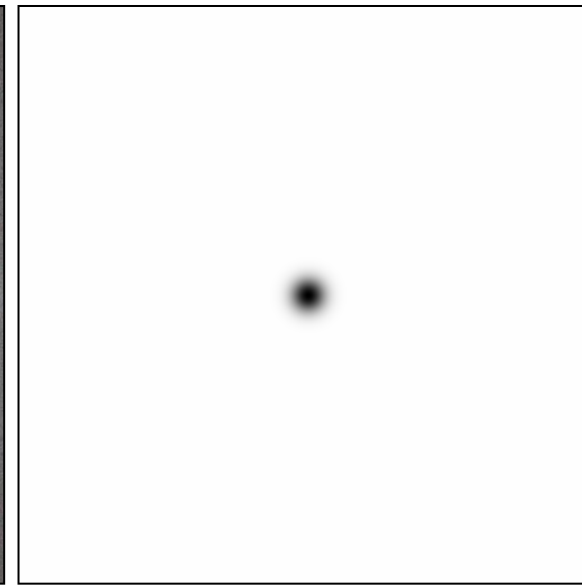
Central Profile

Gaussian Highpass Filter (GHPF)

Image size: 512x512
FD notch sigma = 8







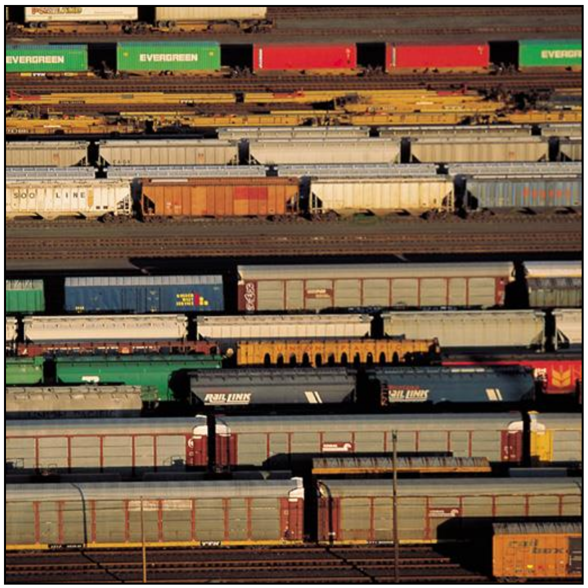
Original Image

Power Spectrum

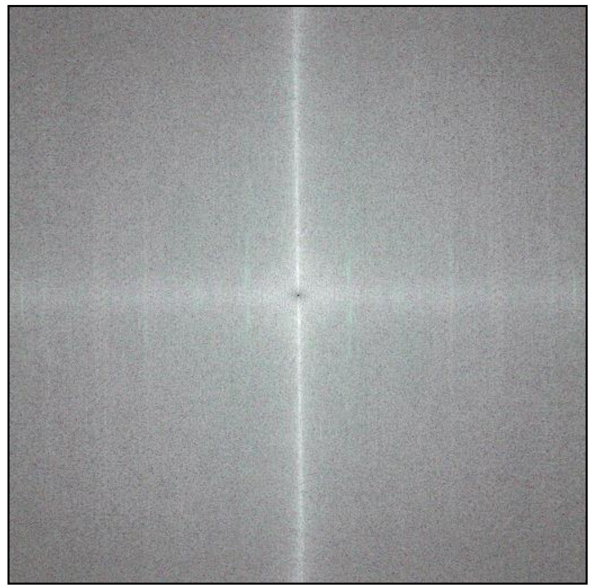
Gaussian HPF in FD

Gaussian Highpass Filter (GHPF)

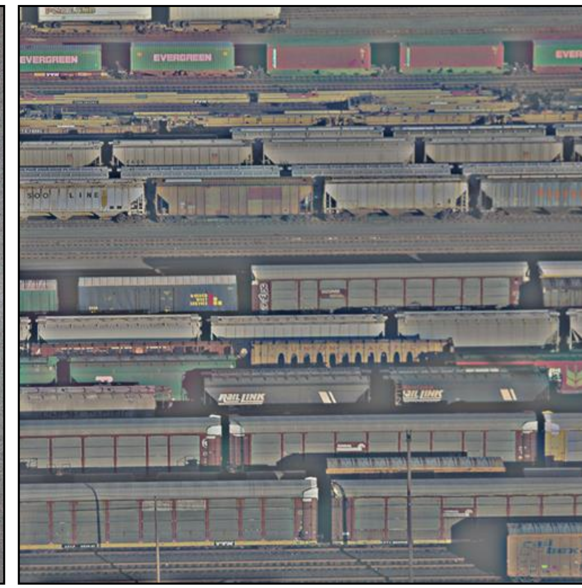
Image size: 512x512
FD notch sigma = 8



Original Image



Filtered Power Spectrum

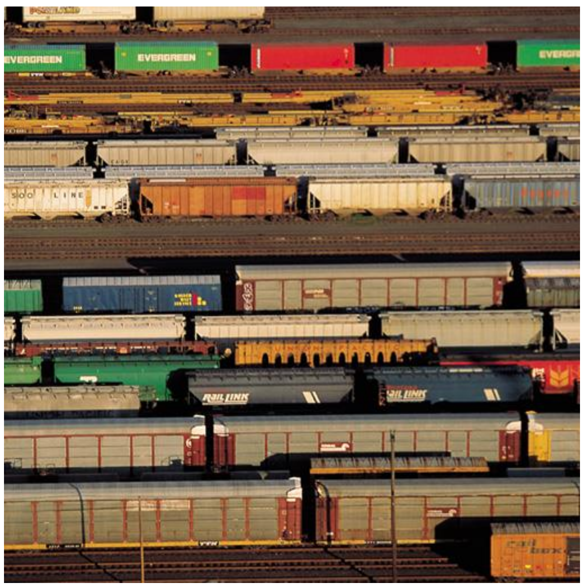
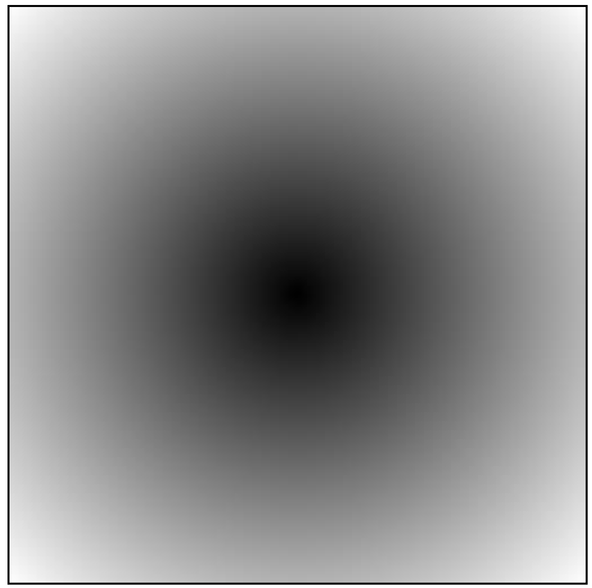
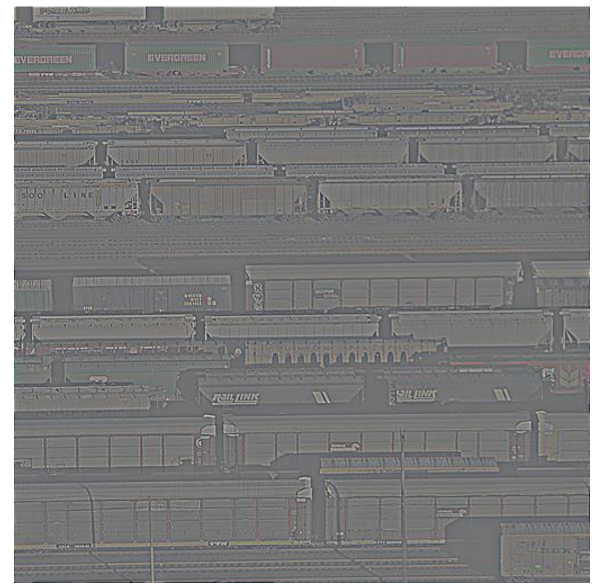


Filtered Image*

*signed image:
0 mapped to 128

Gaussian Highpass Filter (GHPF)

Another Gaussian Highpass Filter

original image

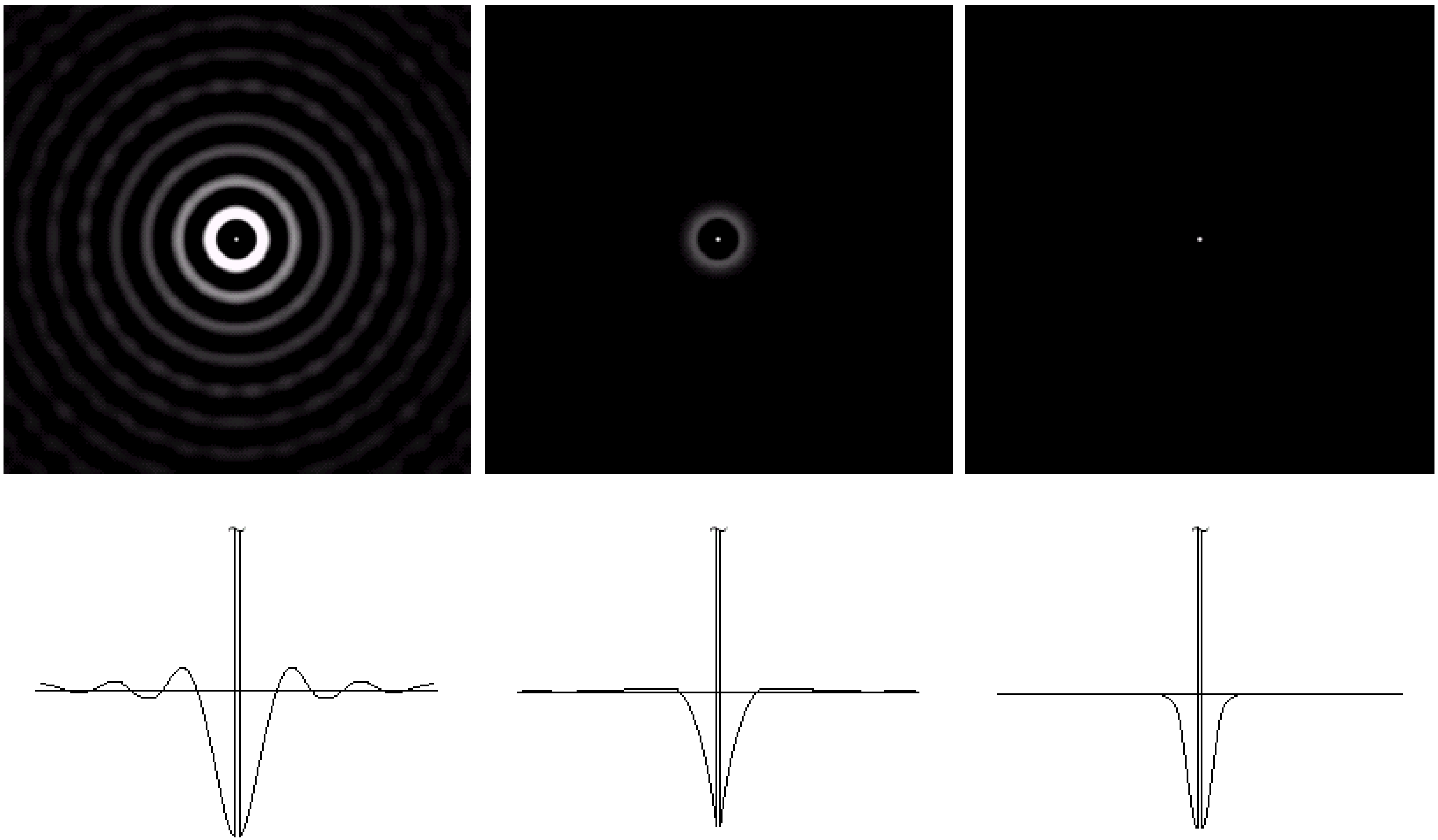
filter power spectrum

filtered image*

*signed image:
0 mapped to 128

Highpass Filters Effects Compared

Ringing Effect

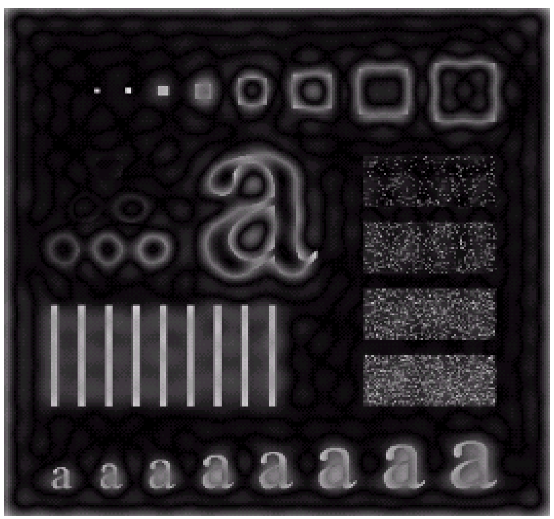


Ideal **Butterworth** **Gaussian**

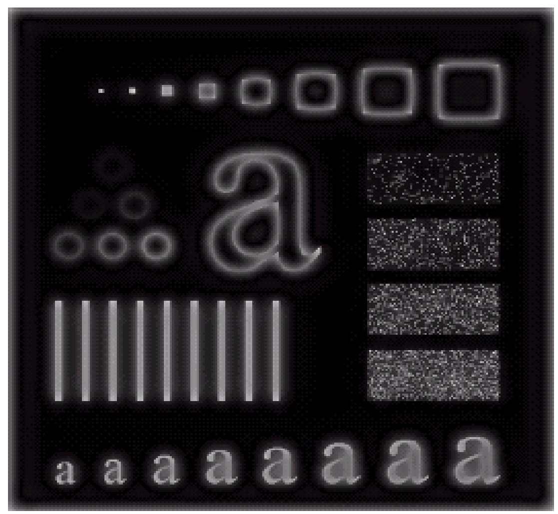
Highpass Filters Effects Compared

- $D_0 = 30$

IHPF



BHPF



GHPF

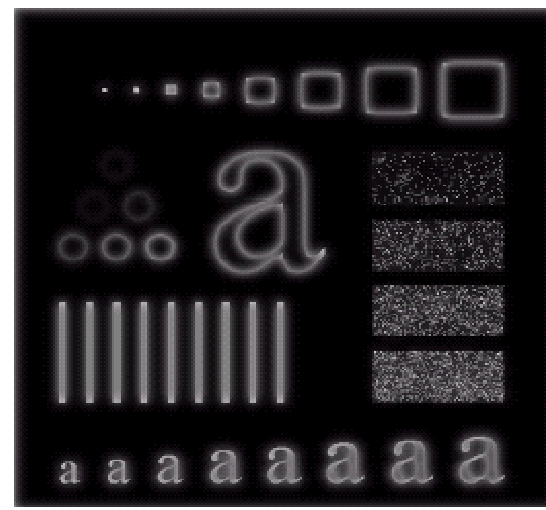
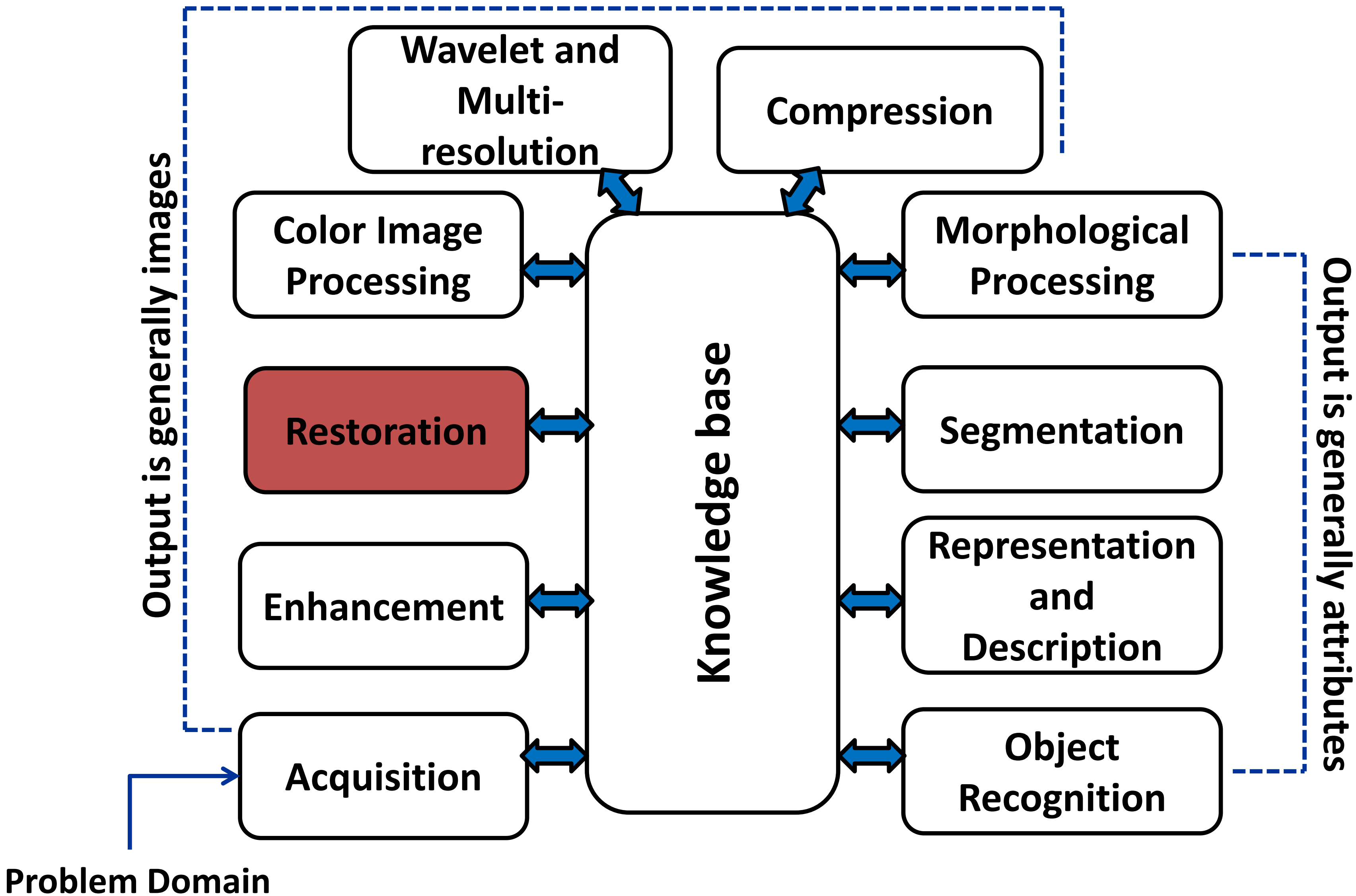


Image Restoration

Fundamental Steps of DIP



Contents

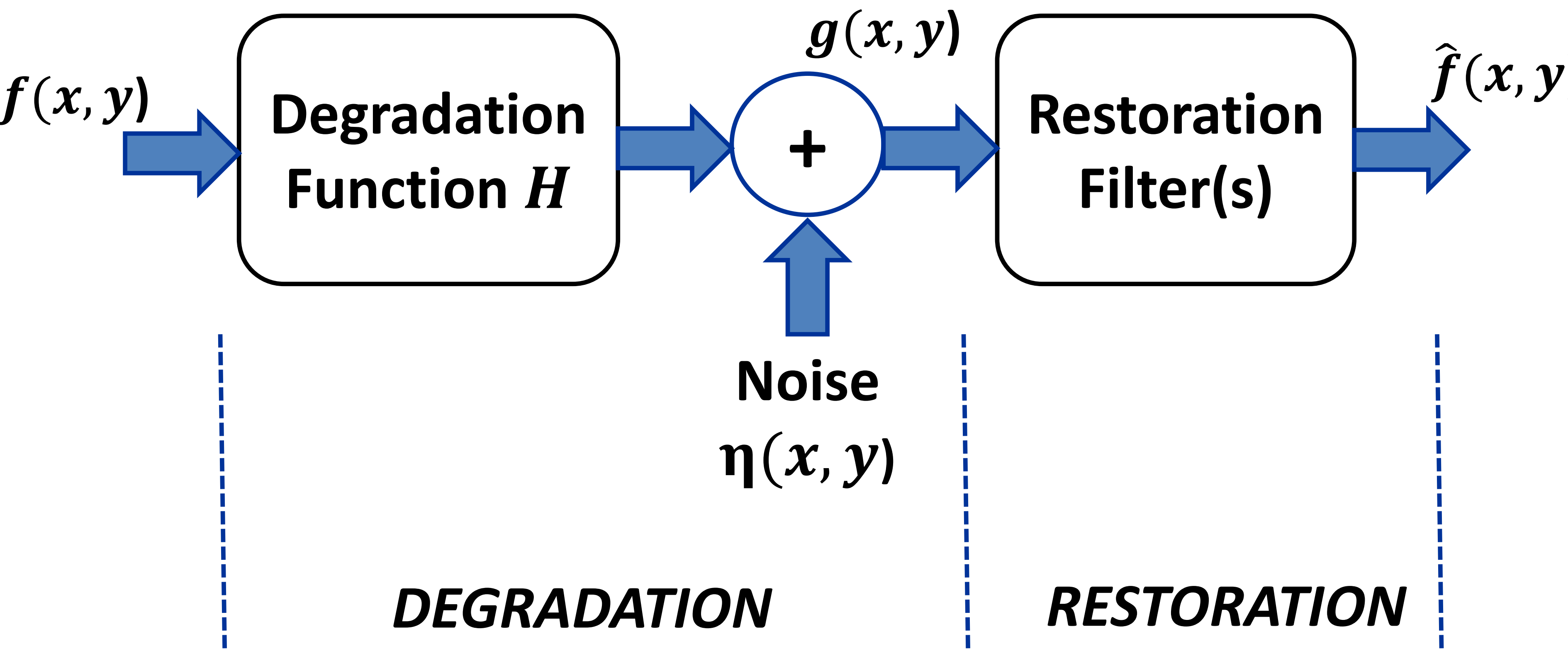
Degradation Model

Restoration

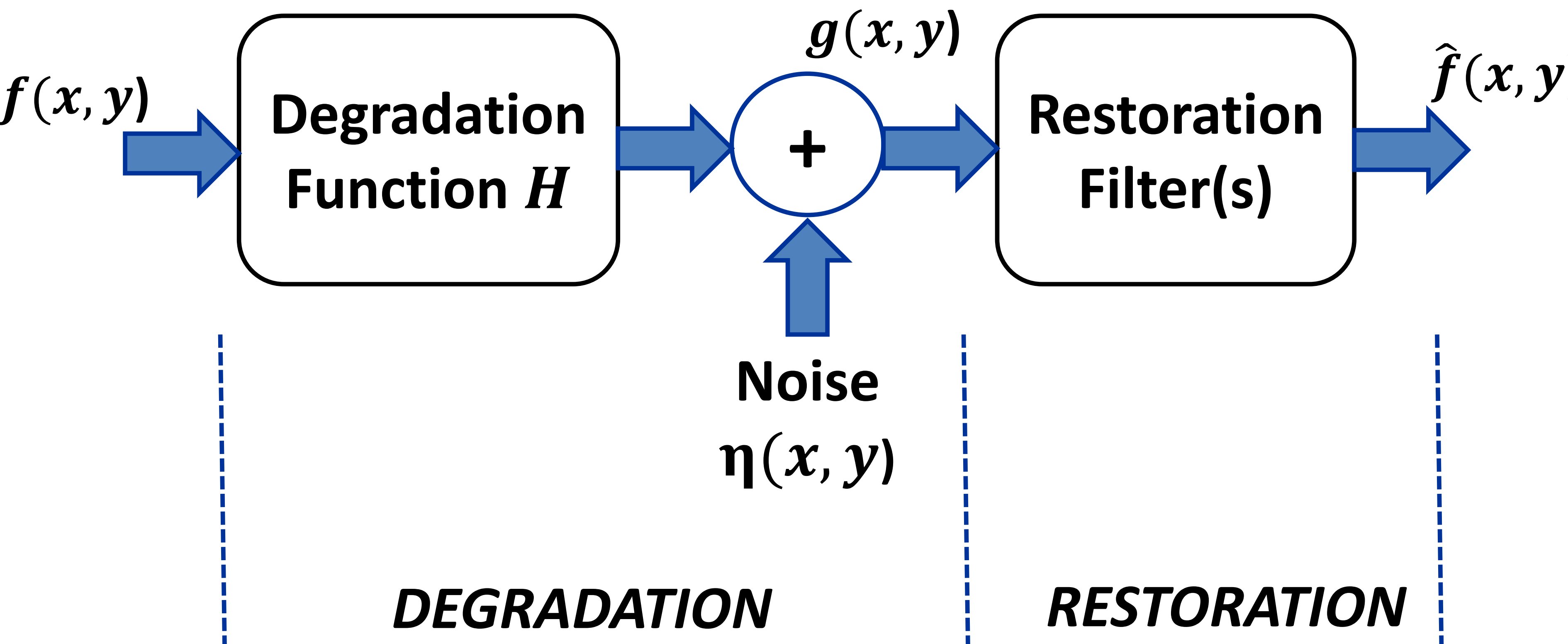
Basics

- **Recovering an image that has been degraded by using a priori knowledge of the degradation process.**
- **By modelling the degradation and applying the inverse process.**
- **Involves criteria of “goodness”.**
- **Restoration vs. enhancement.**

Degradation/Restoration Model



Degradation/Restoration Model – (cont.)



if H is Linear, position invariant process, then

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$
$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

Degradation/Restoration Model – (cont.)

degraded image | pointspread function | undegraded image | Additive noise

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$
$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$

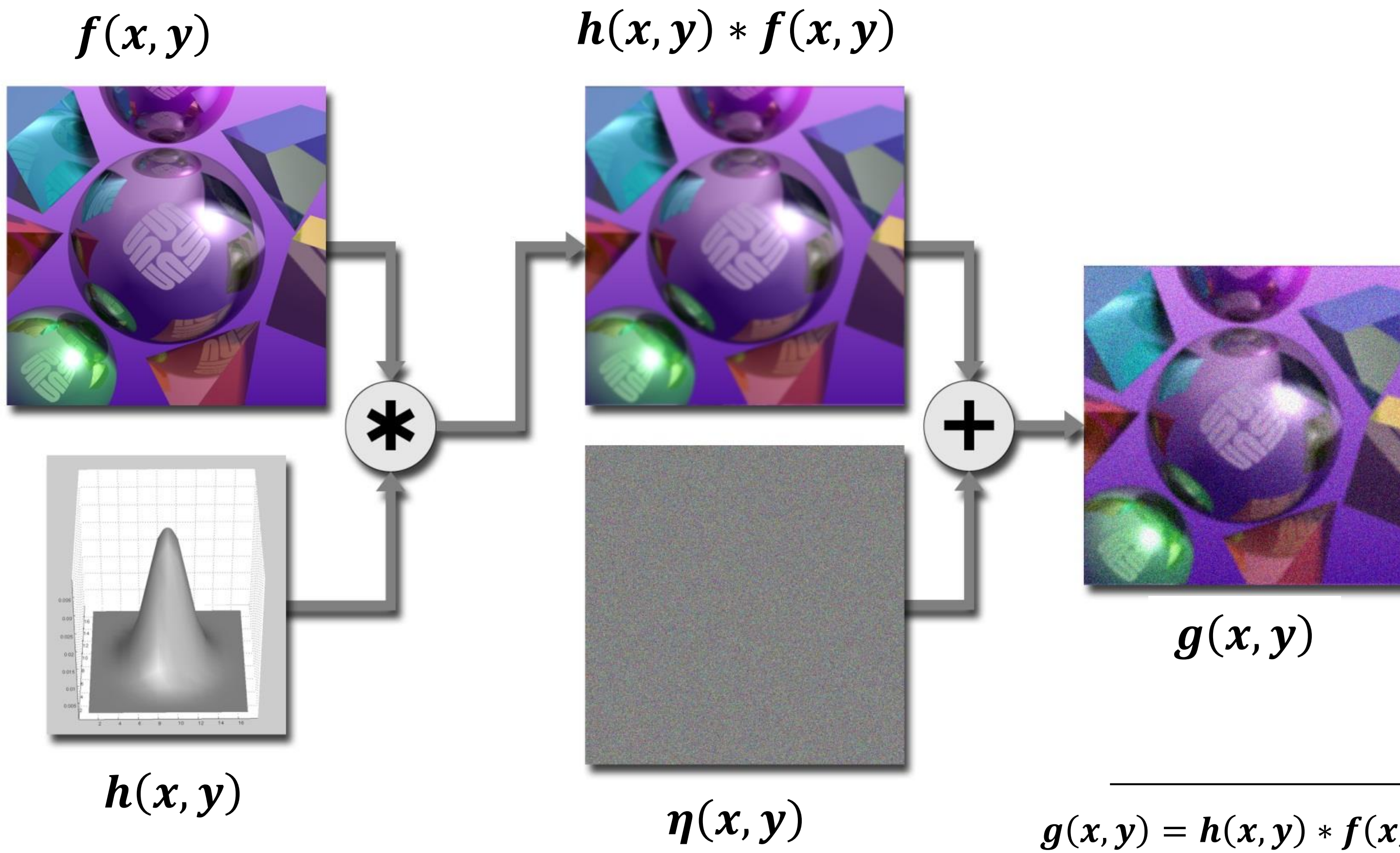
- **A general model of image degradation includes convolution with a point spread function, h, as well as additive noise.**

Point spread function is the system's response to a point source of light, or in signal and systems terminology, it is the impulse response (h) of the system, since the point source of light can be described with a delta function (pulse).

Optical transfer function is the transfer function of the acquisition system, and is defined as the (H), the Fourier transform of the point spread function.

Degradation/Restoration Model – (cont.)

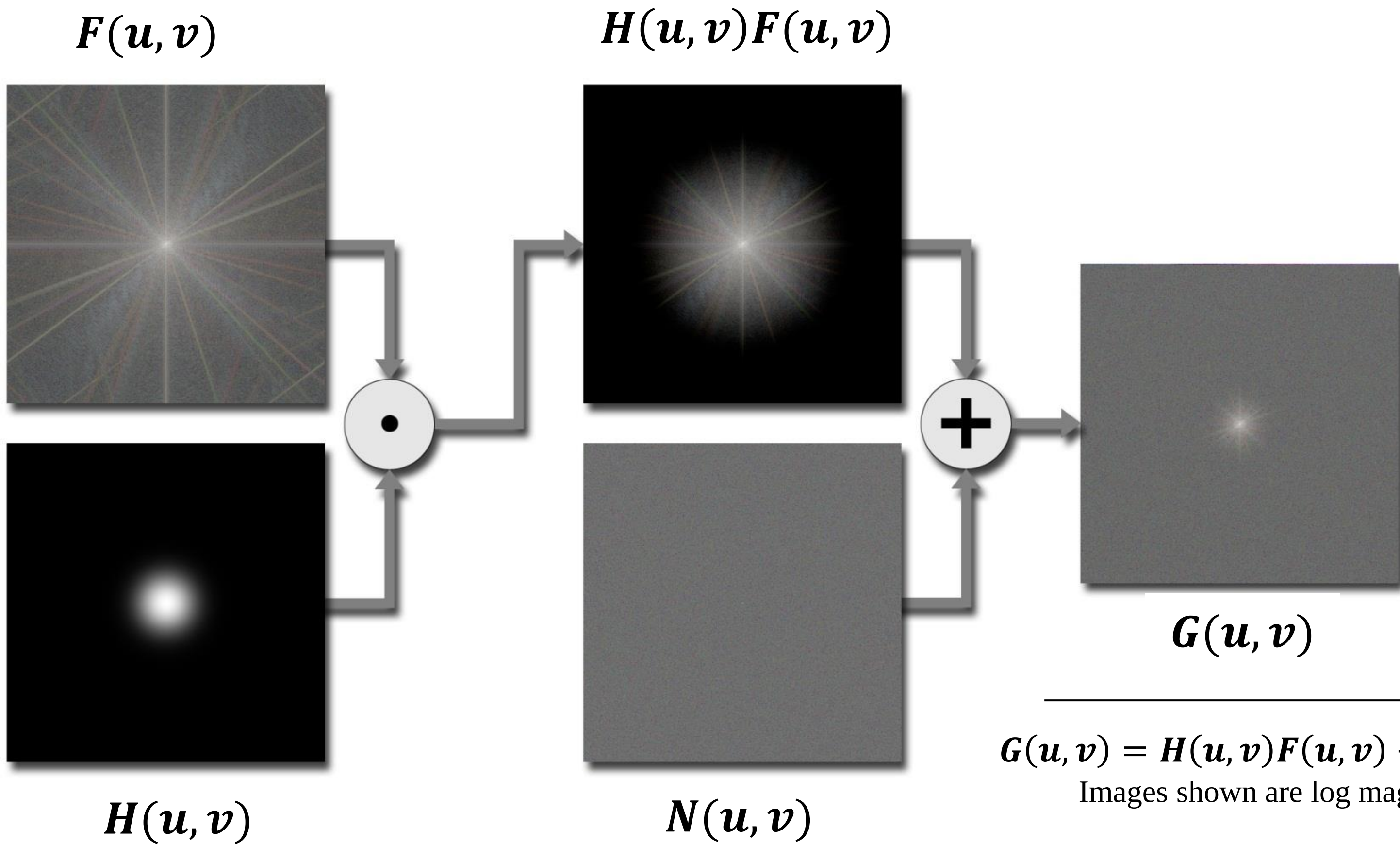
Image Degradation Model



$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

Degradation/Restoration Model – (cont.)

Image Degradation Model (Frequency Domain)



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

Images shown are log magnitude.

Degradation/Restoration Model – (cont.)

- **Assuming the only degradation present is noise and $(H = 1)$, the model becomes**

$$g(x,y) = f(x,y) + \eta(x,y)$$
$$G(u,v) = F(u,v) + N(u,v)$$

- **If we can estimate the model of the noise $\eta(x,y)$ in an image, then this will help us figure out how to denoise the image.**

Degradation/Restoration Model – (cont.)

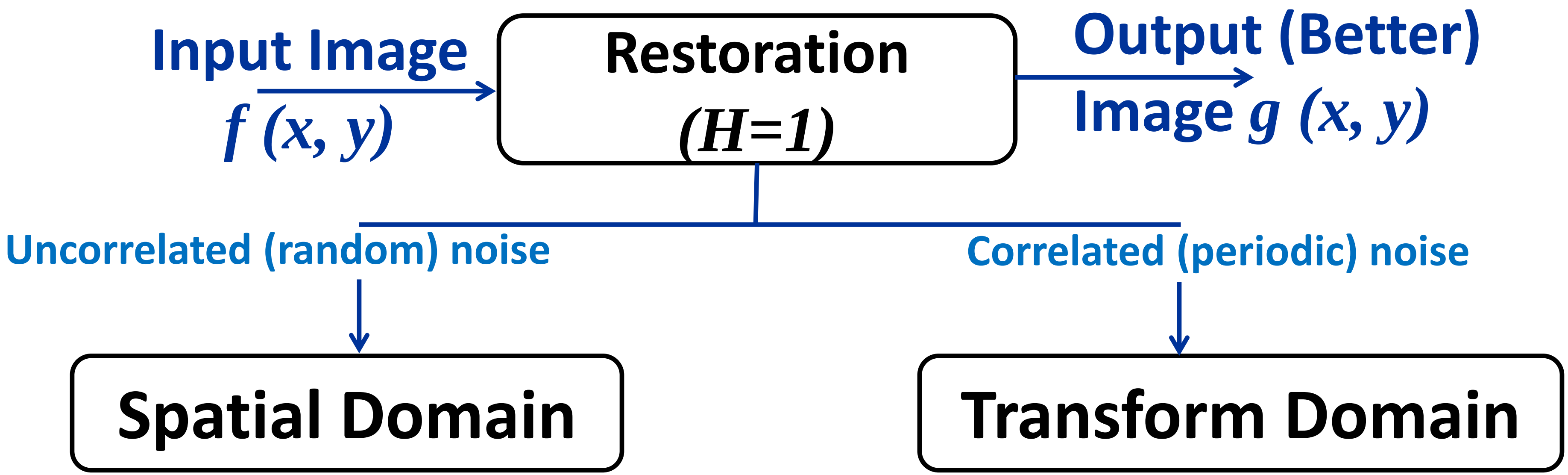
Sources/Types of Noise

- **The type of noise determines the domain and method of denoising the image.**
- **Sources of noise in digital images arise during image acquisition and transmission**
  - **Imaging sensors can be affected by ambient conditions → random noise.**
  - **Interference can be added to an image during transmission → periodic noise.**

Degradation/Restoration Model – (cont.)

Denoising Approach

1. Estimate noise model and parameters
2. Choose suitable domain
3. Design filter and apply it for denoising.



$$g(x, y) = f(x, y) + \eta(x, y)$$

$$G(u, v) = F(u, v) + N(u, v)$$

Denoising in Frequency Domain

Denoising in Frequency Domain

Sources/Types of Noise

- **The type of noise determines the domain and method of denoising the image.**
- **Sources of noise in digital images arise during image acquisition and transmission**
  - **Imaging sensors can be affected by ambient conditions → random noise.**
  - **Interference can be added to an image during transmission → periodic noise.**

Denoising in Frequency Domain – (cont.)

Assuming the only degradation is noise

$$g(x, y) = f(x, y) + \eta(x, y)$$
$$G(u, v) = F(u, v) + N(u, v)$$

- For *periodic noise*, N can be estimated from the spectrum, and hence G is computed.

Denoising in Frequency Domain – (cont.)

(7) Periodic Noise

- From electrical or electromechanical interference during acquisition.
- Spatially dependent.
- Can be reduced significantly via frequency domain filtering.

Denoising in Frequency Domain

Still, the only degradation is noise

$$g(x, y) = f(x, y) + \eta(x, y)$$
$$G(u, v) = F(u, v) + N(u, v)$$

- **For *periodic noise*, N can be estimated from the spectrum, and hence G is computed.**

Denoising in Frequency Domain – (cont.)

- **Periodic noise appears as concentrated bursts (spikes) of energy in the Fourier Transform. So we use *selective filters* to isolate the noise.**
- **Can be observed even by visual analysis of the PS.**

Parameter estimation

- **From the PS of the noisy image.**

Denoising in Frequency Domain – (cont.)

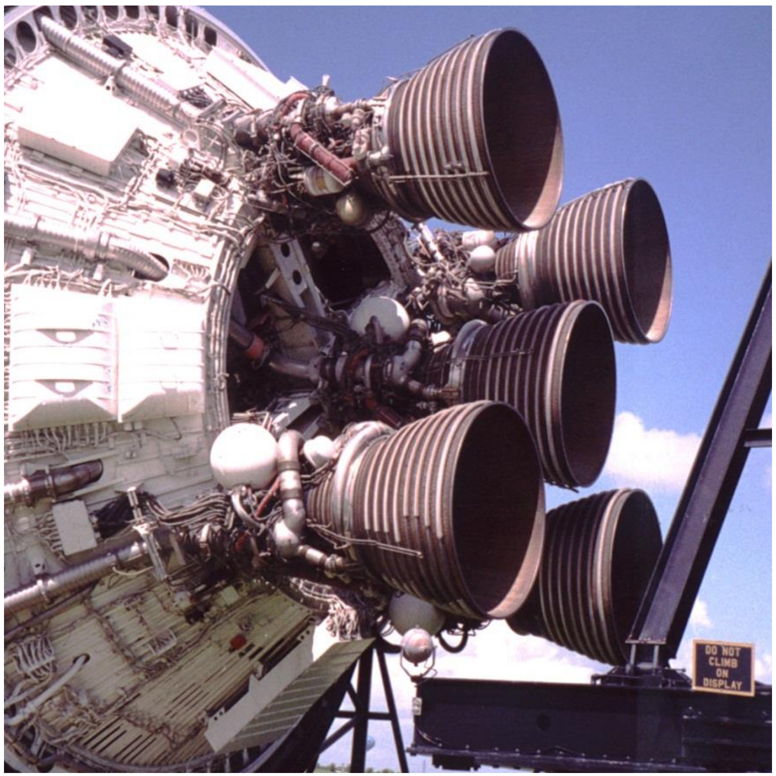
- Automatic analysis only in case the spikes are exceptionally pronounced, or prior knowledge is available about the general location of the frequency component of the interference.

Selective Filters

- Bandreject Filters
- Bandpass Filters
- Notch Filters

Denoising in Frequency Domain – (cont.)

Periodic Noise



original image

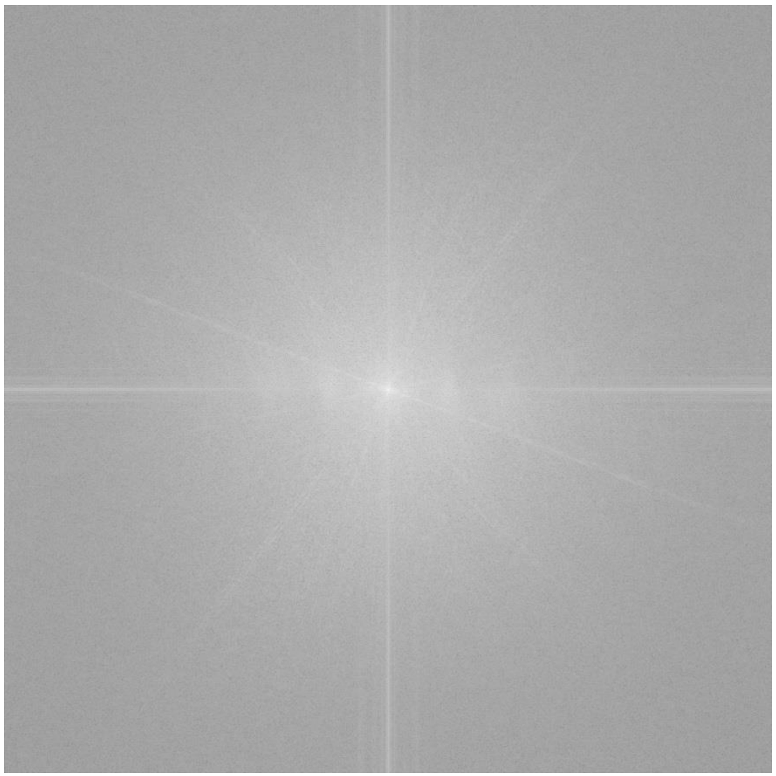
image + noise

Pratt & Whitney Rocketdyne J-2 rocket engines on Apollo 18's Saturn V second stage.

119

Denoising in Frequency Domain – (cont.)

Power Spectrum of Image with Periodic Noise



original image

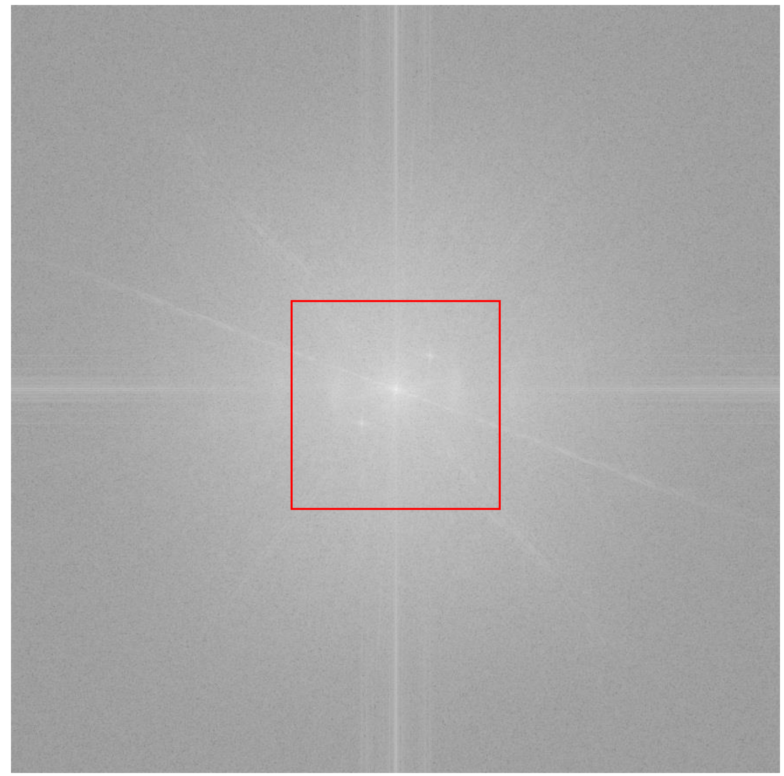


image + noise

Selective Filters

1. Bandreject

- If the general location of the noise components in the frequency domain is approximately known.

$D(u,v)$: **Distance from center of frequency rectangle.**

D_o: **Radial center of the band.**

W: **Width of the band.**

- **Ideal**

$$H_{BRI}(u,v) = \begin{cases} 1 & D_o - \frac{W}{2} \leq D \leq D_o + \frac{W}{2} \\ 0 & otherwise \end{cases}$$

- **Butterworth**

$$H_{BRB}(u,v) = \frac{1}{1 + \left(\frac{DW}{D^2 - D_o^2}\right)^{2n}}$$

- **Gaussian**

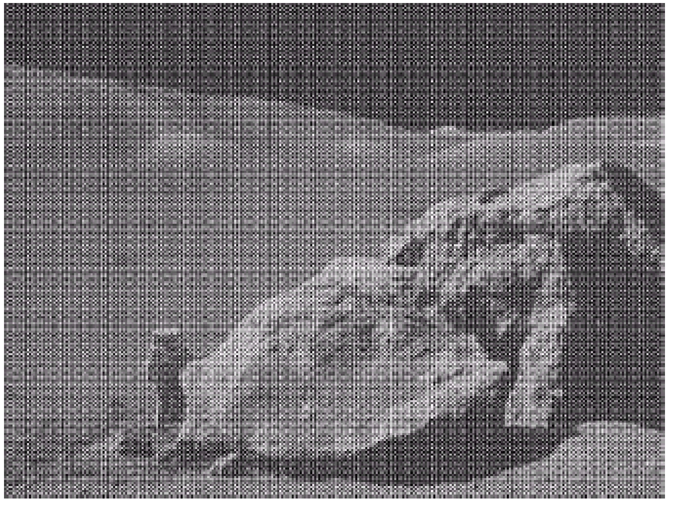
$$H_{BRG}(u,v) = 1 - e^{\left(\frac{D^2 - D_o^2}{DW}\right)^2}$$

Selective Filters – (cont.)

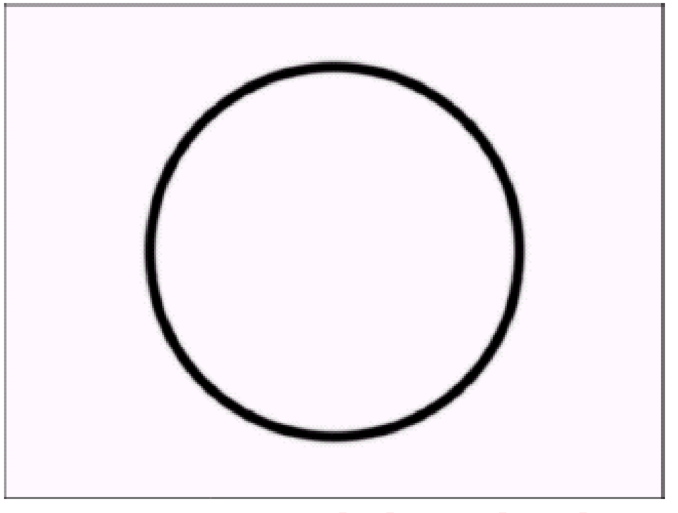
1. Bandreject

- **Example**

Image corrupted by sinusoidal noise



Fourier spectrum of corrupted image



Butterworth band reject

Filtered image

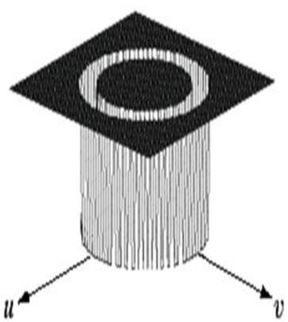
Selective Filters – (cont.)

2. Bandpass **- Not common in practice as it removes too much image detail.** **- Useful in isolating the noise on an image:** ***(1) Perform BP filtering to pass noise.*** ***(2) Take IDFT of the filtered transform.***	- **Ideal** $H_{BPI}(u,v) = 1 - H_{BRI}(u,v)$ - **Butterworth** $H_{BPB}(u,v) = 1 - H_{BRB}(u,v)$ - **Gaussian** $H_{BPG}(u,v) = 1 - H_{BRG}(u,v)$

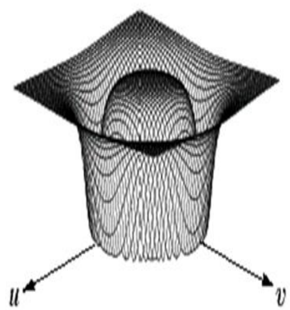
Selective Filters – (cont.)

2. Bandpass

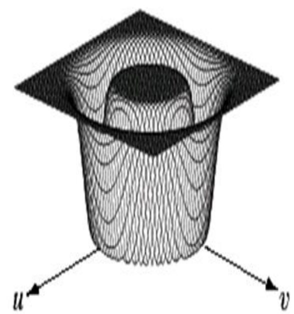
- **Example**



Ideal

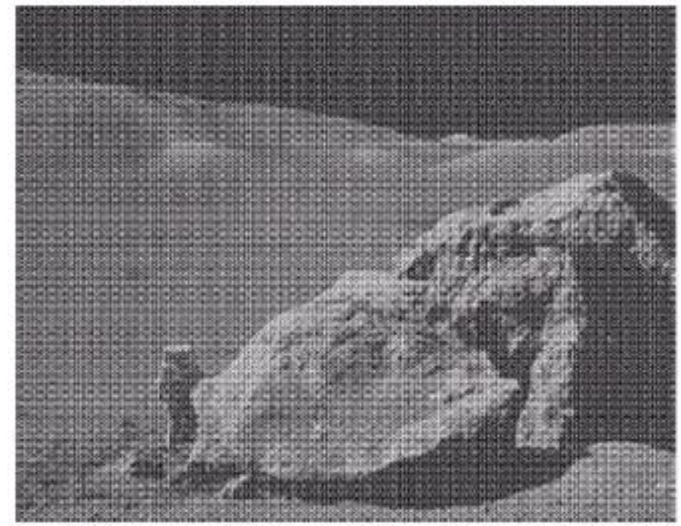


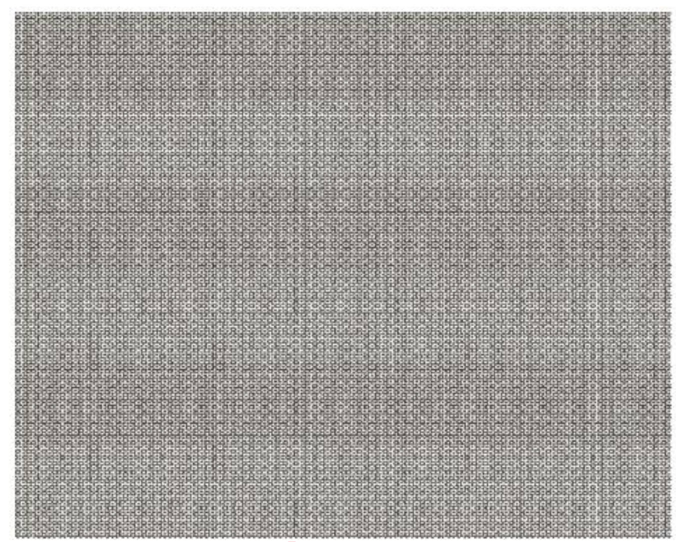
Butterworth (of order 1)



Gaussian

Image corrupted by sinusoidal noise



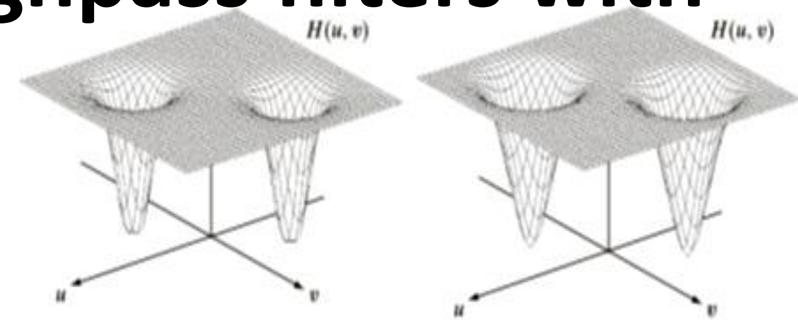


Noise Pattern

Selective Filters – (cont.)

3. Notch Filter

- To process a small range of frequencies in a predefined neighborhood.
- Appears in symmetric pairs.
- Ideal, Butterworth, Gaussian.
- Usually the product of highpass filters with shifted centers.

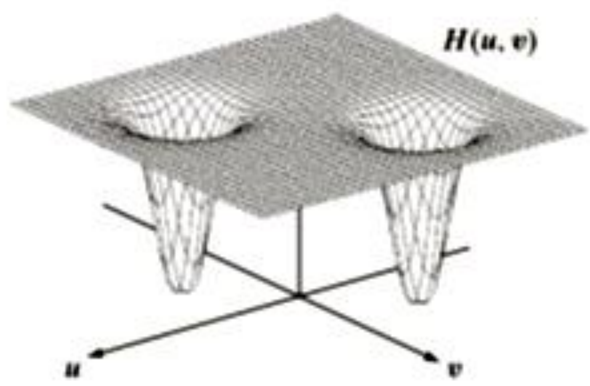


Selective Filters – (cont.)

3. Notch Filter

- **Shapes**

Ideal



Butterworth

Gaussian

Selective Filters – (cont.)

3. Notch Filter

- Appear in symmetric pairs.

- Number and shape of pairs is arbitrary.

- Notch reject:

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

$$H_{-k}(u,v) = H(-u,-v)$$

Q**: Number of pairs.**

- Notch Pass:

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

- Ideal

$$H_{INR}(u,v) = \prod_{k=1}^{Q} H_{IHP}(u,v) H_{-IHP}(u,v)$$

$$H_{IHP}(u,v) = \begin{cases} 1 & D \leq D_o \\ 0 & otherwise \end{cases}$$

- Butterworth

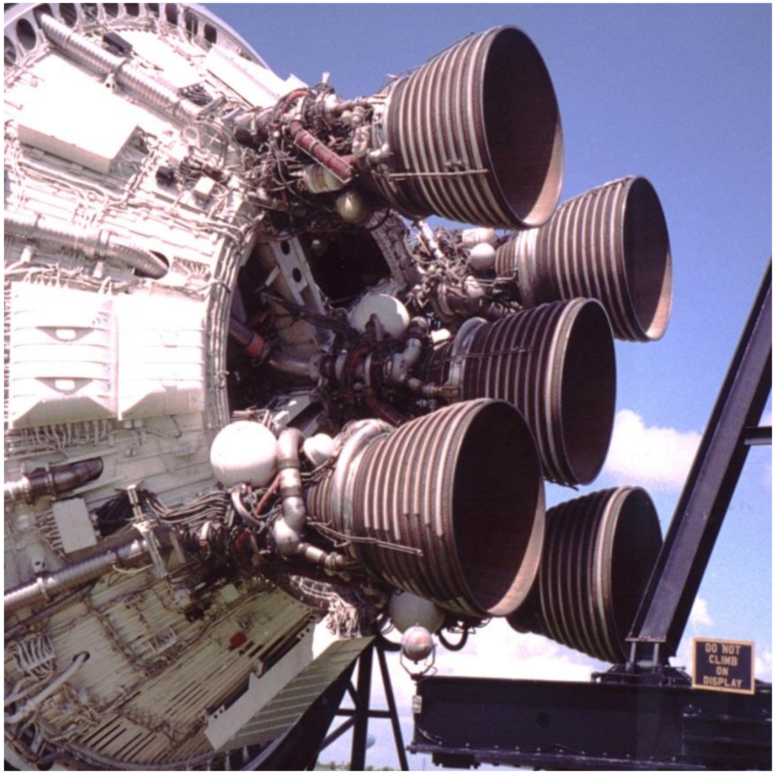
$$H_{BNR}(u,v) = \prod_{k=1}^{Q} \left[\frac{1}{1 + \left(\frac{D_o}{D_k(u,v)} \right)^{2n}} \right] \left[\frac{1}{1 + \left(\frac{D_o}{D_k(-u,-v)} \right)^{2n}} \right]$$

- Gaussian

$$H_{GNR}(u,v) = \prod_{k=1}^{Q} \left[1 - e^{-\left(\frac{D_k^2(u,v)}{2D_o} \right)} \right] \left[1 - e^{-\left(\frac{D_k^2(-u,-v)}{2D_o} \right)} \right]$$

Selective Filters – (cont.)

Periodic Noise



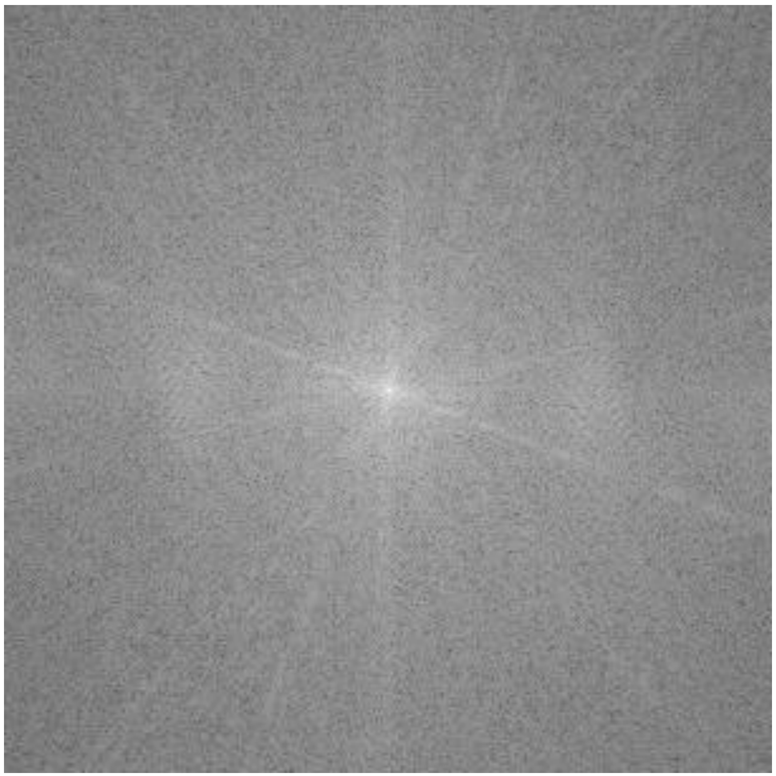
original image

image + noise

Pratt & Whitney Rocketdyne J-2 rocket engines on Apollo 18's Saturn V second stage.

Selective Filters – (cont.)

Low Frequency Region



original image

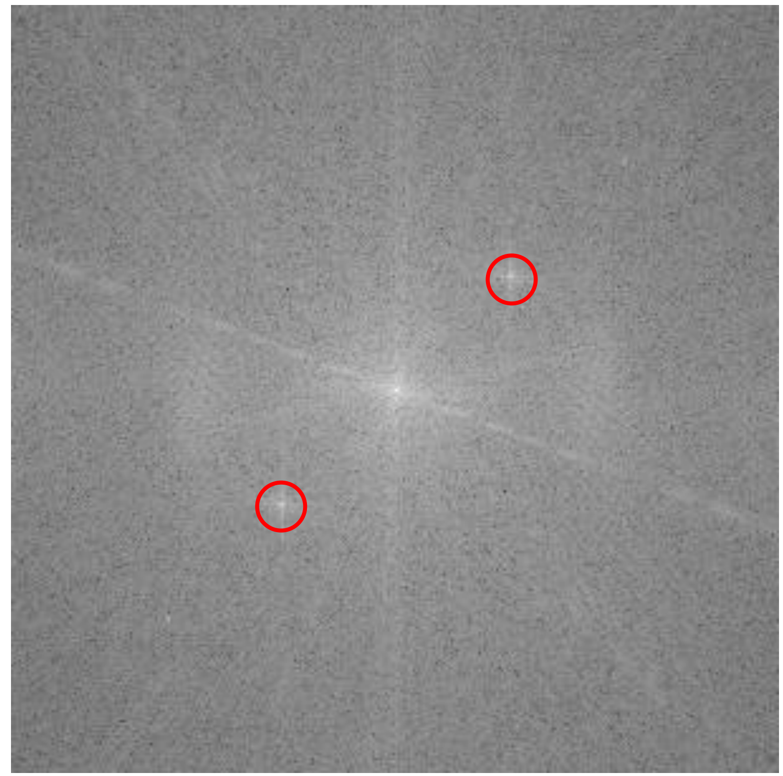
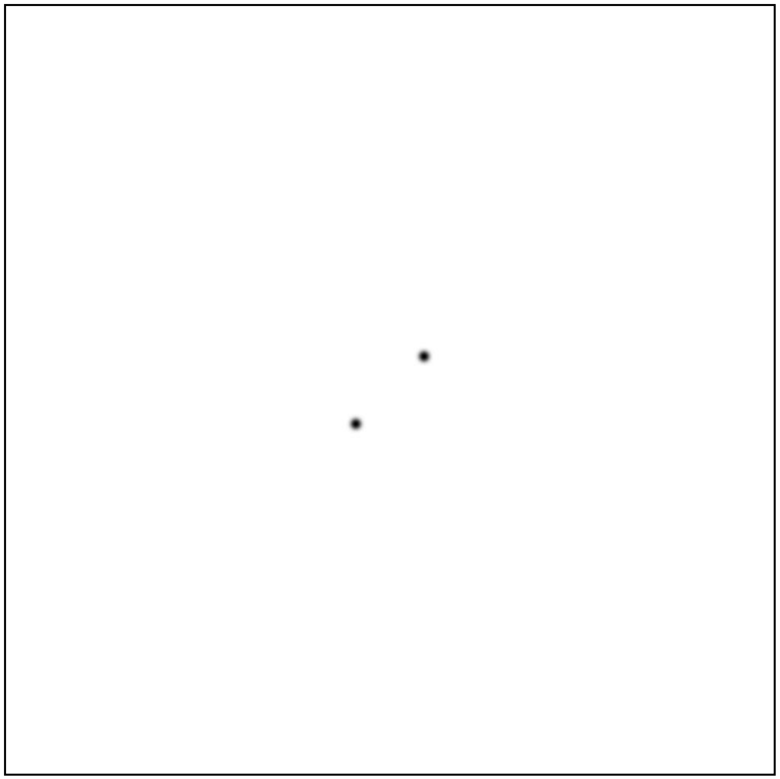


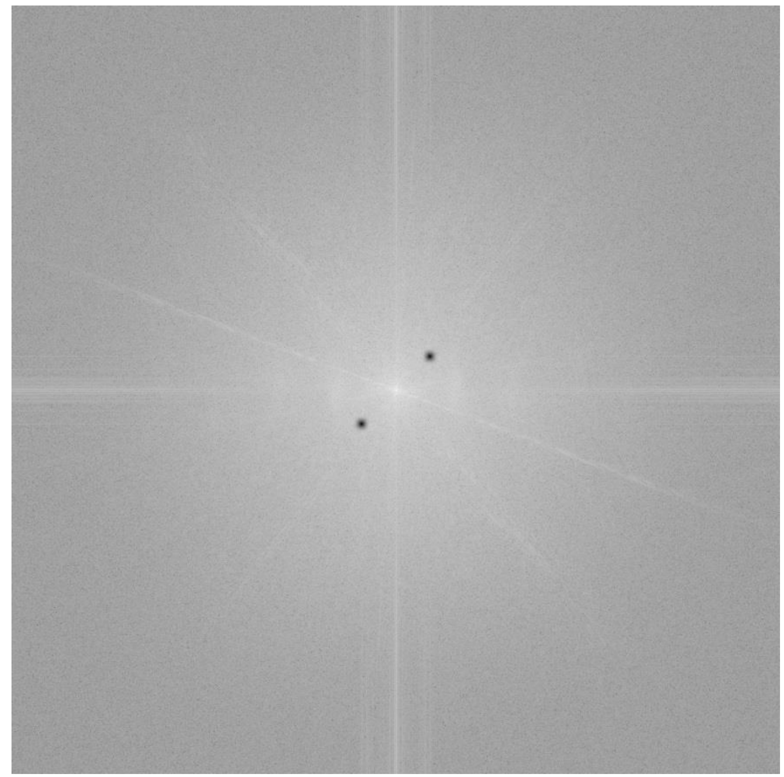
image + noise

Selective Filters – (cont.)

Noise Reduction through Notch Filtering



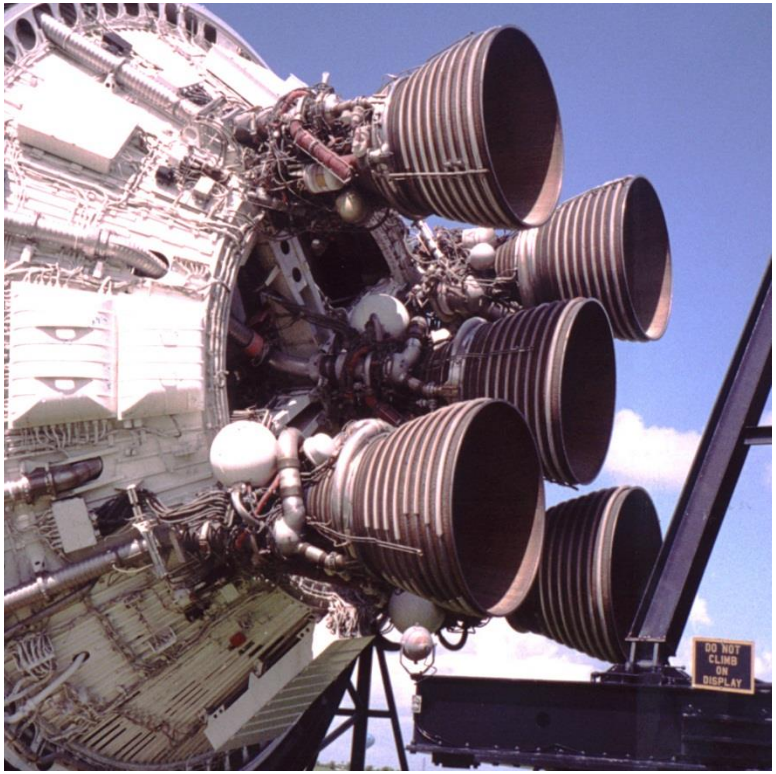
noise mask



masked power spectrum

Selective Filters – (cont.)

Inverse of Masked Fourier Transform



original image

noise reduced image

Pratt & Whitney Rocketdyne J-2 rocket engines on Apollo 18's Saturn V second stage.

Uncorrelated Noise Models

Uncorrelated (Random) Noise

Assumptions

- **Noise is independent of spatial coordinates.**
- **Noise is uncorrelated to image itself.**

Spatial Descriptor

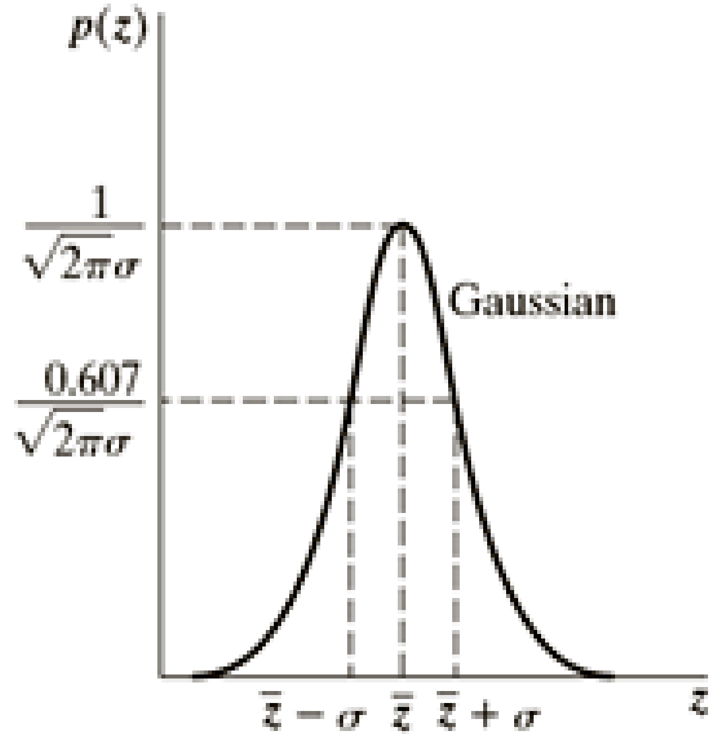
- **Statistical behavior of intensity values of noise component ← Probability Density Function of the noise.**

Uncorrelated (Random) Noise – (cont.)

Most Common Noise PDF

(1) Gaussian Noise

- Also called normal noise, and is mathematically tractable in spatial and frequency domain.
- Arises due to electronic circuit noise and sensor noise due to poor illumination and/or high temperature.
- $p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$

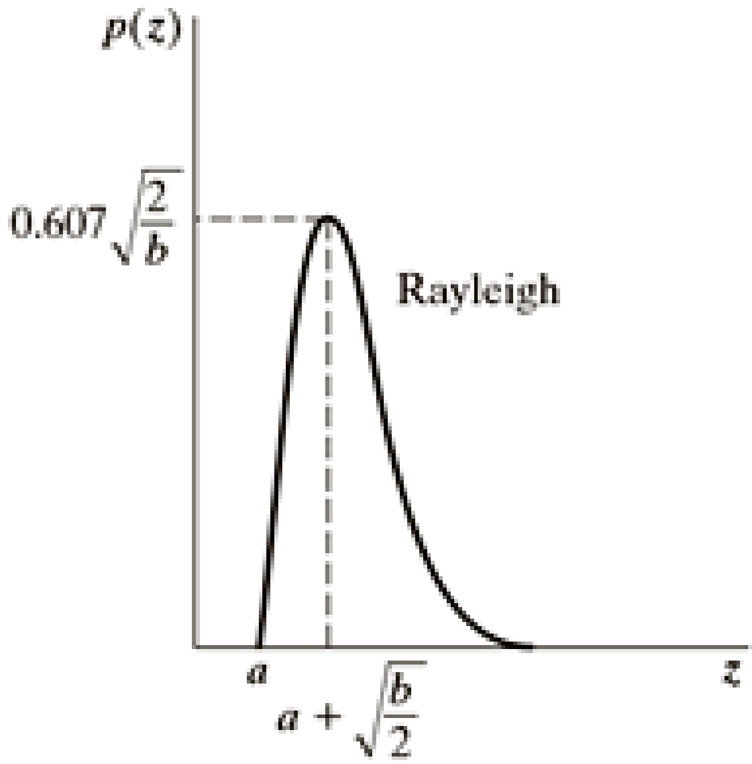


Uncorrelated (Random) Noise – (cont.)

Most Common Noise PDF

<u>(2) Rayleigh Noise</u>

- Helpful in characterizing noise in range imaging.
- Useful for approximating skewed histograms.



- $p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}} & z \geq a \\ 0 & z < a \end{cases}$

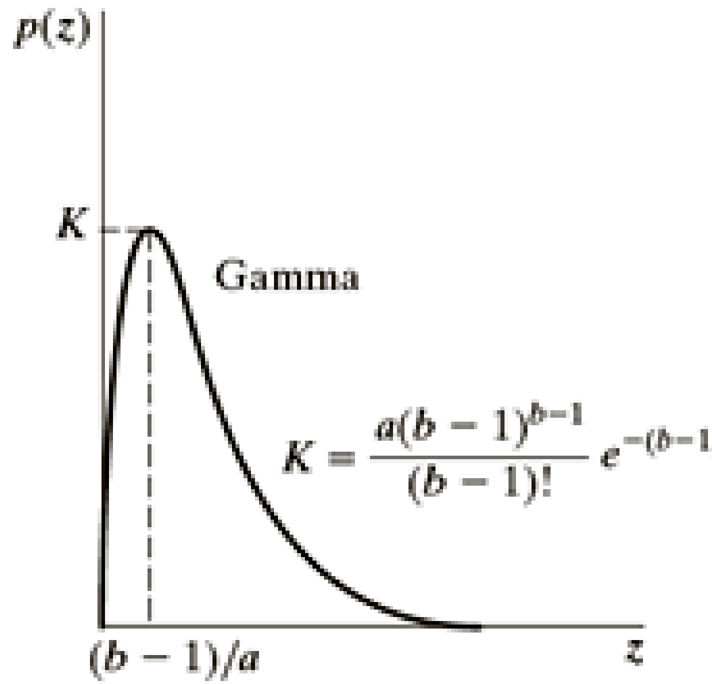
- $\bar{z} = a + \sqrt{\frac{\pi b}{4}}, \sigma^2 = \frac{b(4-\pi)}{4}.$

Uncorrelated (Random) Noise – (cont.)

Most Common Noise PDF

(3) Erlang (Gamme) Noise

- Called Gamma if the denominator is $\Gamma(b)$.
- Apps in laser imaging.
- $p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$
- $\bar{z} = \frac{b}{a}, \sigma^2 = \frac{b}{a^2}$

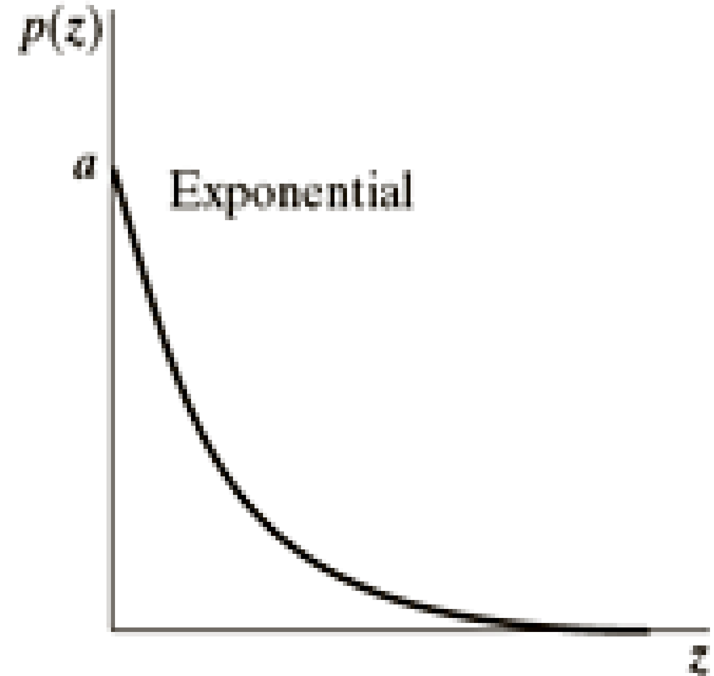


Uncorrelated (Random) Noise – (cont.)

Most Common Noise PDF

(4) Exponential Noise

- Special case of Erlang ($b = 1$).
- Apps in laser imaging.
- $p(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$
- $\bar{z} = \frac{1}{a}, \sigma^2 = \frac{1}{a^2}$



Uncorrelated (Random) Noise – (cont.)

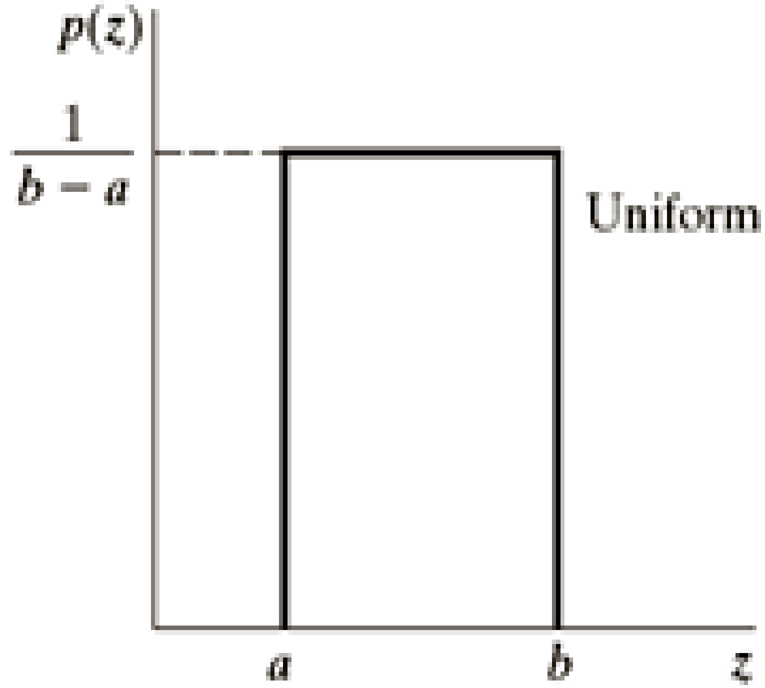
Most Common Noise PDF

(5) Uniform Noise

- Least used practically.

- $p(z) = \begin{cases} \frac{1}{(b-a)} & a \leq z \leq b \\ 0 & otherwise \end{cases}$

- $\bar{z} = \frac{a+b}{2}, \sigma^2 = \frac{(b-a)^2}{12}$

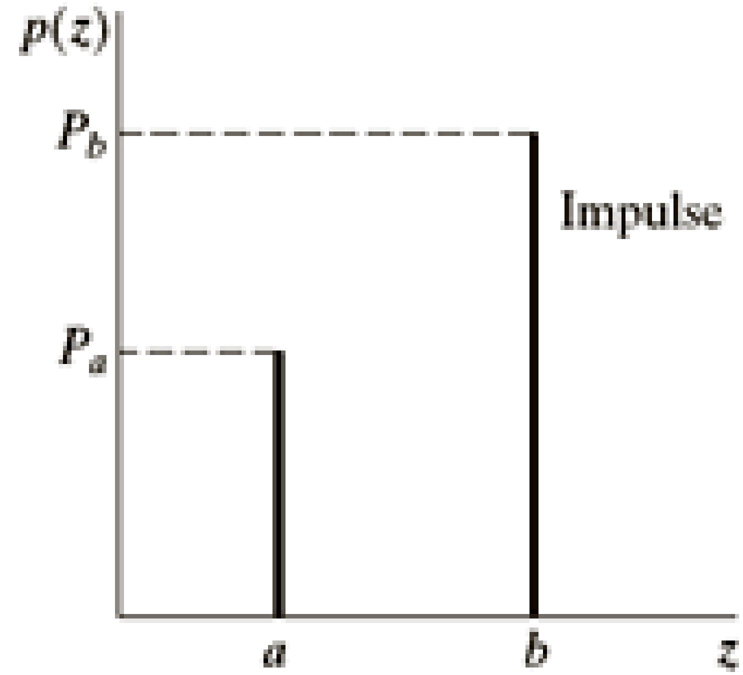


Uncorrelated (Random) Noise – (cont.)

Most Common Noise PDF

(6) Impulse Noise

- Unipolar or bipolar.

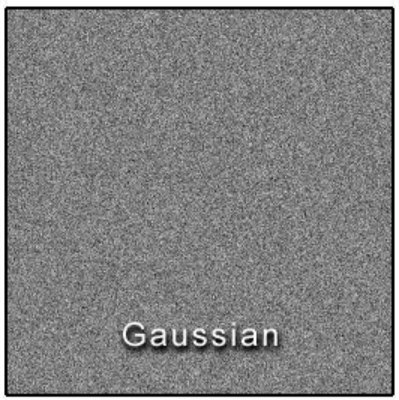


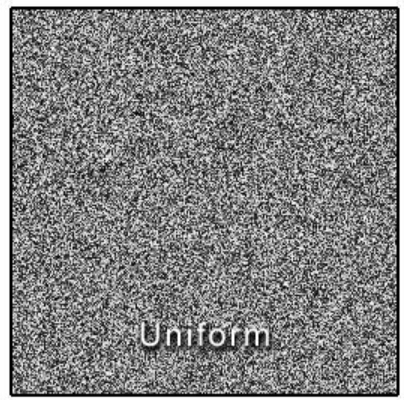
- $$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & otherwise \end{cases}$$

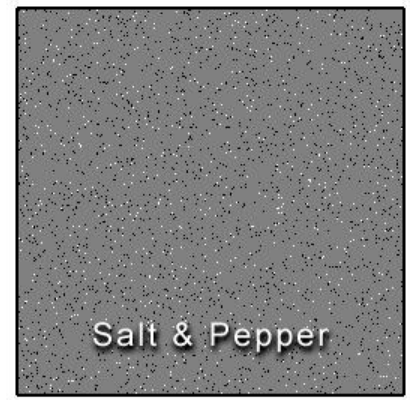
- Salt-and-pepper, impulse, data-drop-out, spike.
- Usually, a and b are digitized as extremes (B&W).

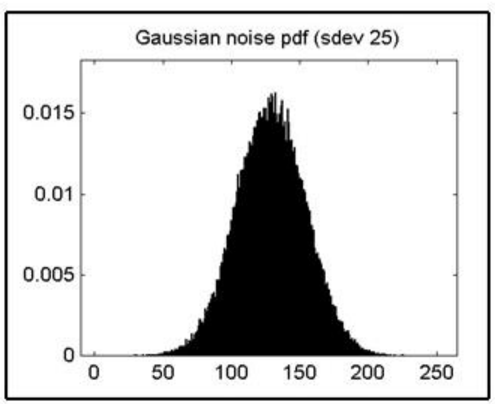
Uncorrelated (Random) Noise – (cont.)

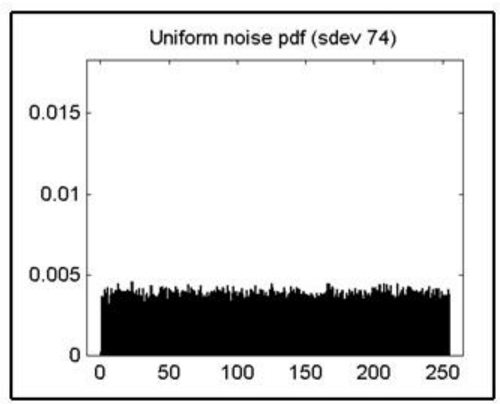
Intensity distributions – normalized histograms

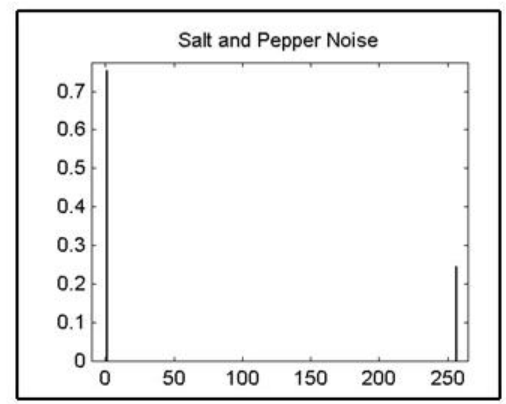












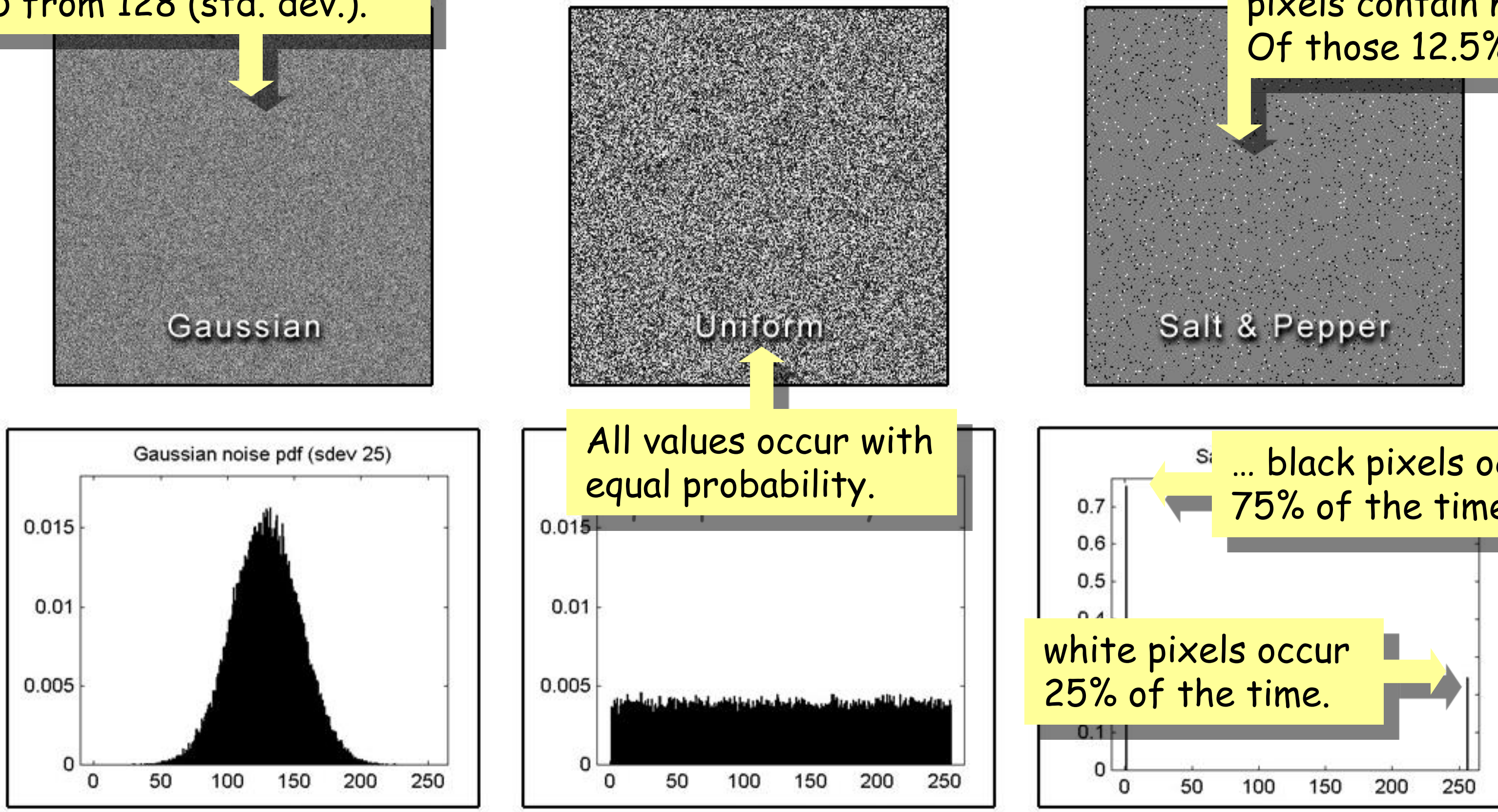
Each pixel's value has probability of occurrence given by the associated distribution.

Uncorrelated (Random) Noise – (cont.)

Intensity distributions – normalized histograms

The most likely value is 128 with an average difference of 25 from 128 (std. dev.).

This is sparse noise: Only 12.5% of the pixels contain noise. Of those 12.5% …



All values occur with equal probability.

… black pixels occur 75% of the time and

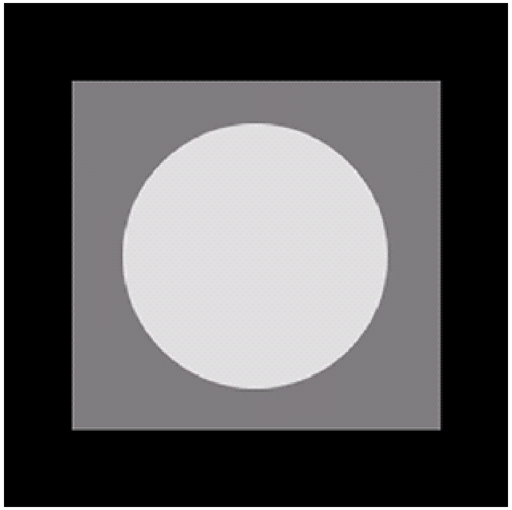
white pixels occur 25% of the time.

Each pixel’s value has probability of occurrence given by the associated distribution.

Uncorrelated (Random) Noise – (cont.)

Example

Can we determine the noise type from the histogram of the image?

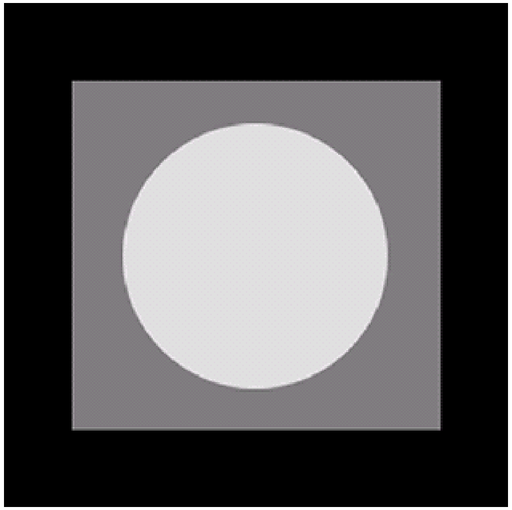


Image

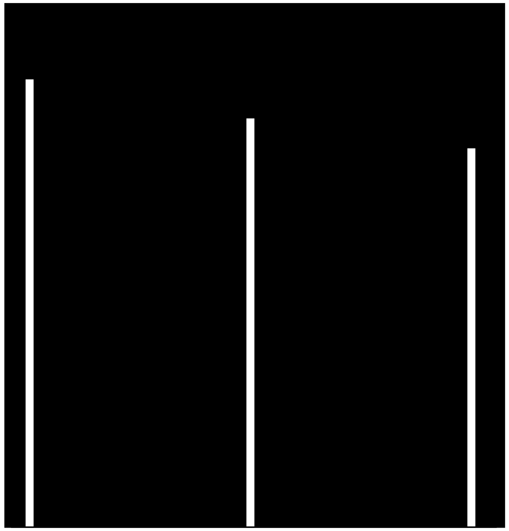
Uncorrelated (Random) Noise – (cont.)

Example

Can we determine the noise type from the histogram of the image?

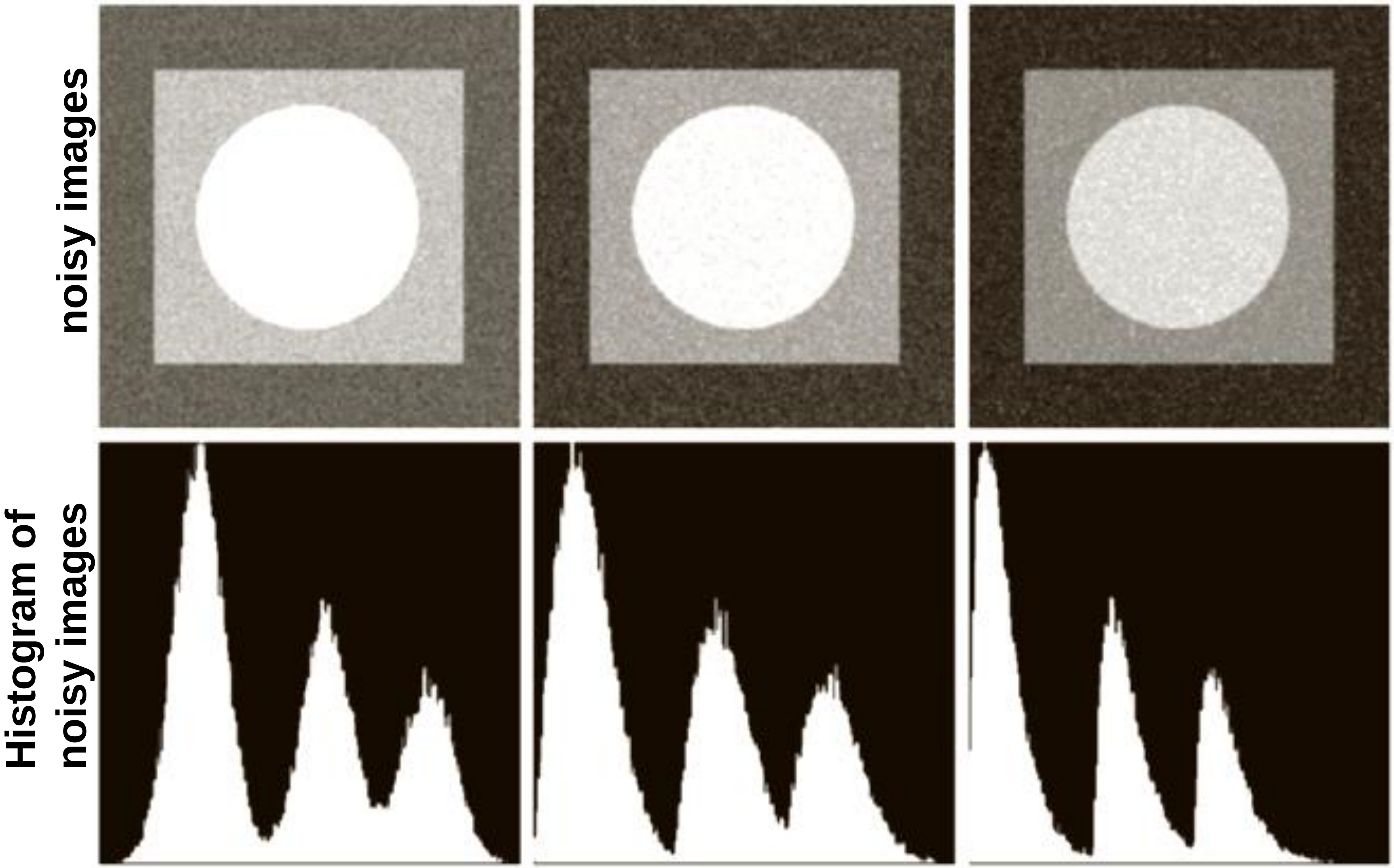


Image

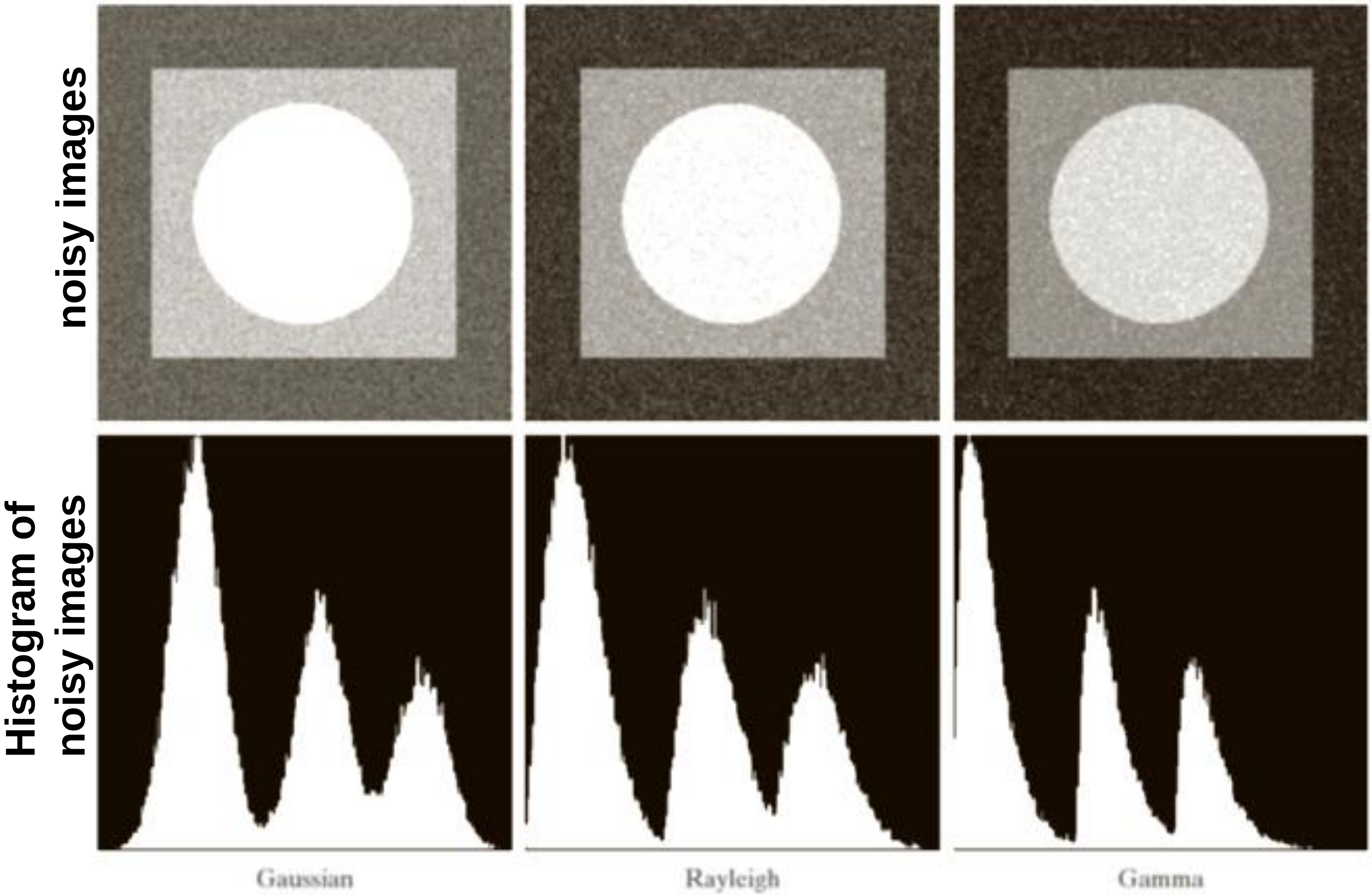


Histogram

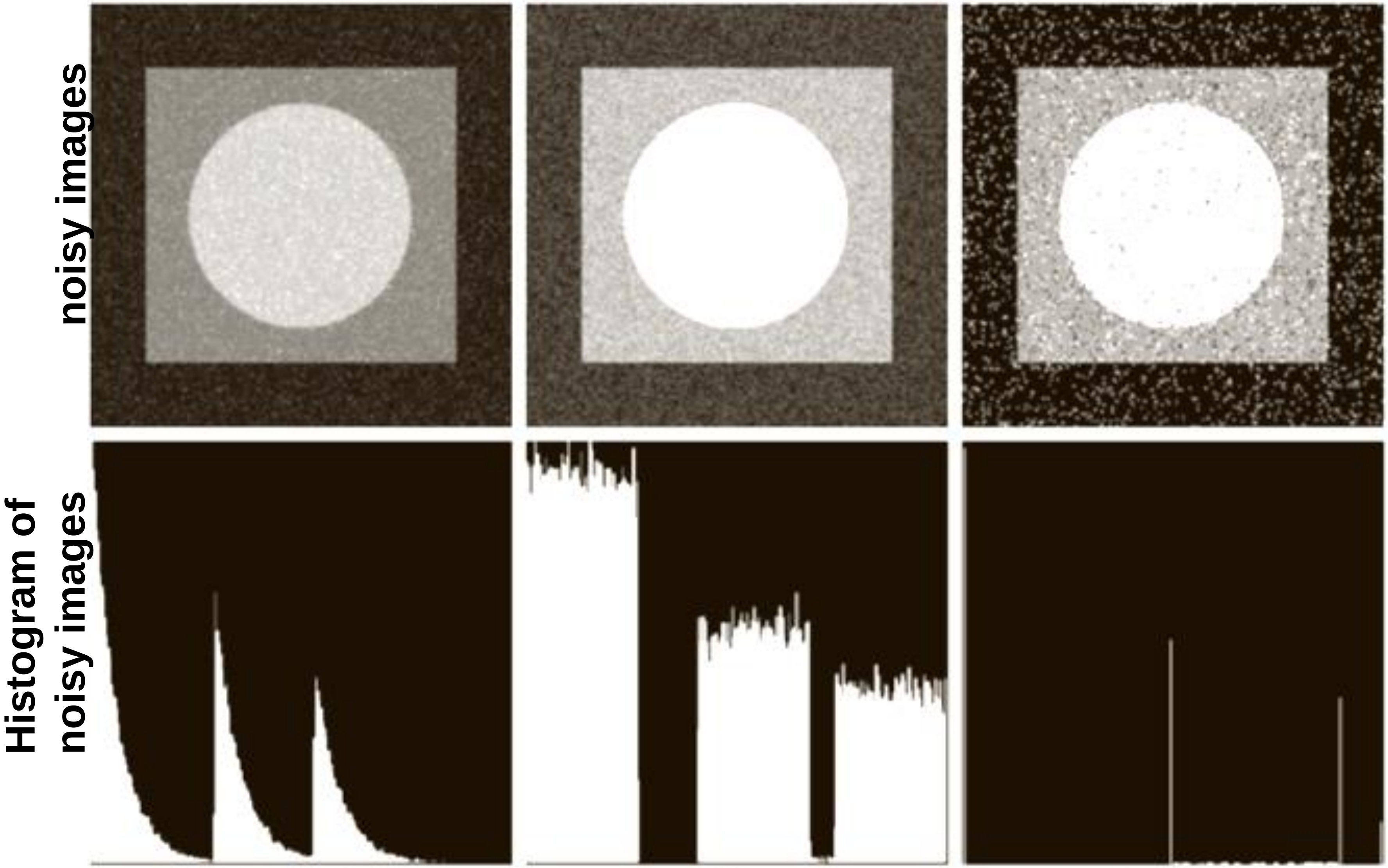
Uncorrelated (Random) Noise – (cont.)



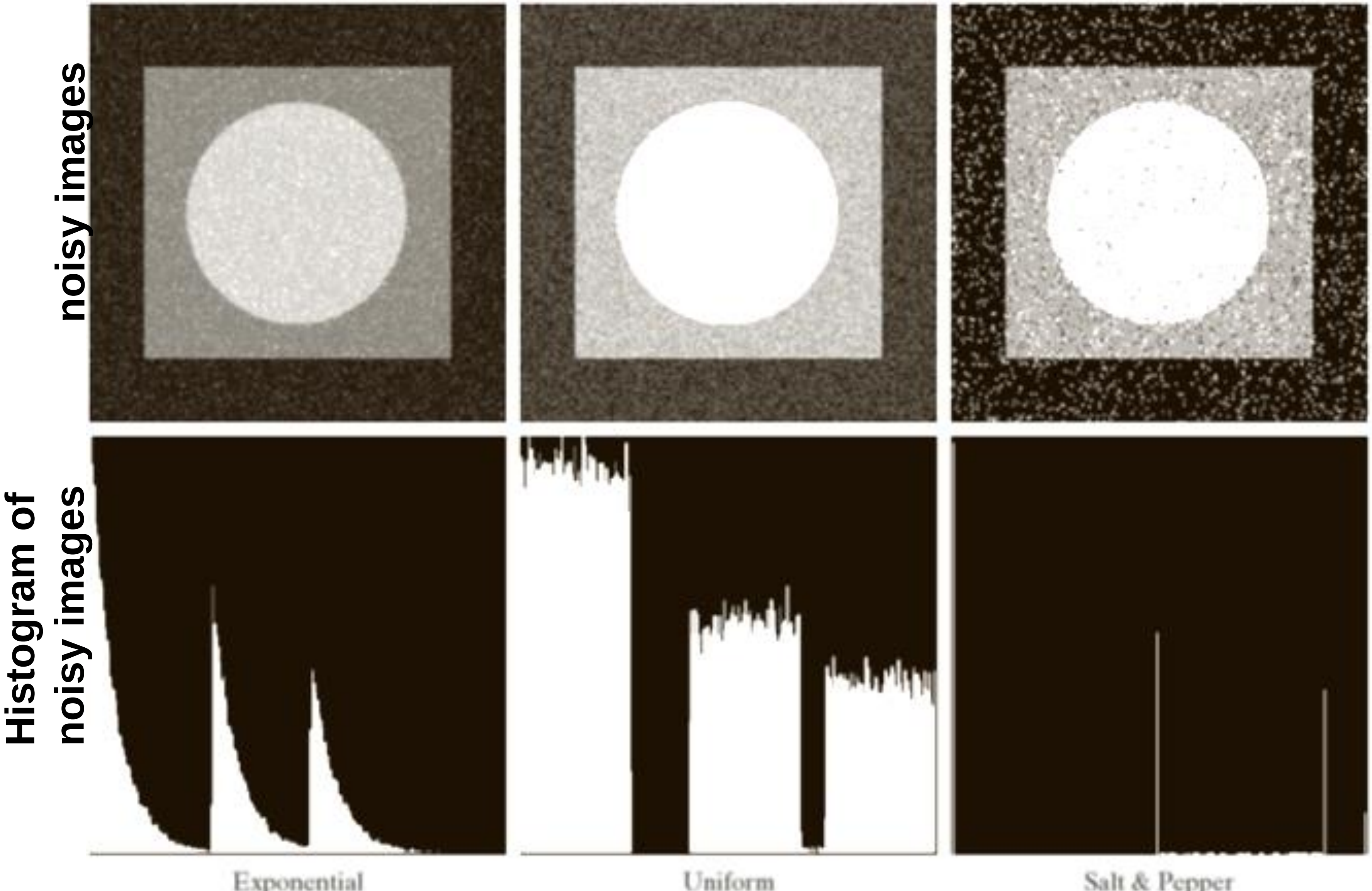
Uncorrelated (Random) Noise – (cont.)



Uncorrelated (Random) Noise – (cont.)



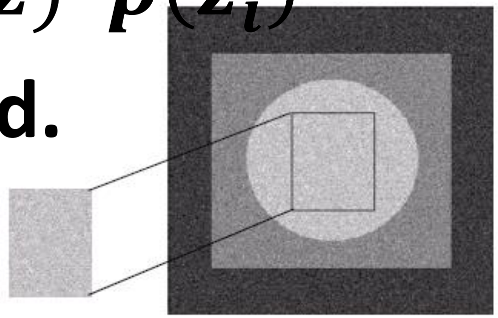
Uncorrelated (Random) Noise – (cont.)



Uncorrelated (Random) Noise – (cont.)

Estimating Noise Parameters

- Partially from sensor specs.
- Available imaging device: shoot flat uniform areas.
- Available images: crop patch of constant intensity.
- From histogram or from the patch estimate:
  - $p(z_i), i = 0, 1, \dots, L-1$ → mean and variance.
  - $\bar{z} = \sum_{i=0}^{L-1} z_i p(z_i), \sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z})^2 p(z_i)$
  - Estimate other parameters if needed.



Denoising in Spatial Domain

Denoising in Spatial Domain

Remember the only degradation is noise

$$g(x, y) = f(x, y) + \eta(x, y)$$

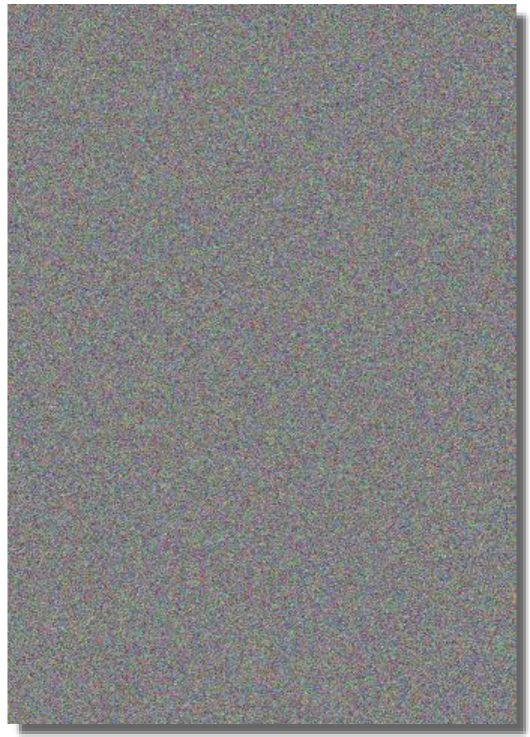
- **Spatial filtering is the best method of choice when only additive random noise of a possibly modelled distribution is present.**
- **Why?**

Denoising in Spatial Domain – (cont.)

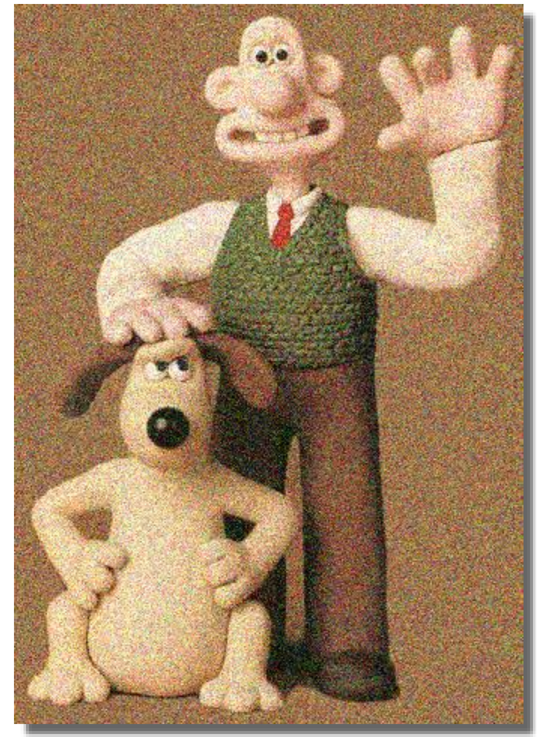
Additive Noise: Example

What if we try denoising these types of noise in the frequency domain?

original image



noise image



image+noise

Denoising in Spatial Domain – (cont.)

Additive Noise: Example

displayed:

$$\log\left\{\left|\mathscr{F}(\mathbf{I})\right|^2 + 1\right\}$$

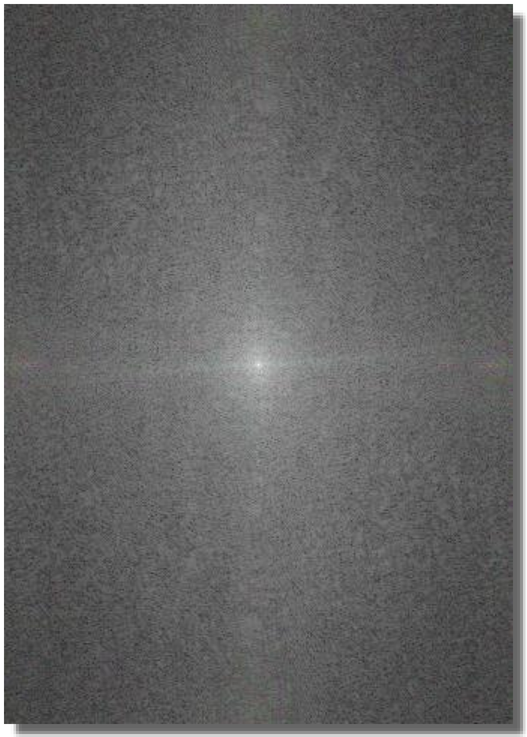
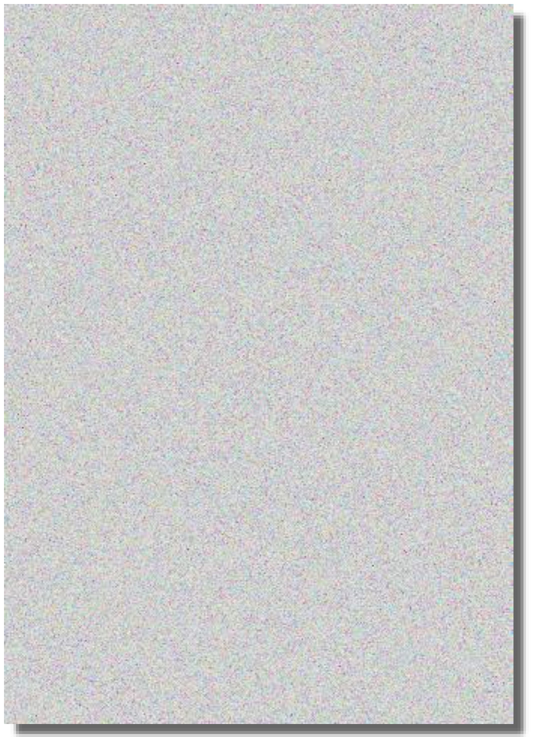
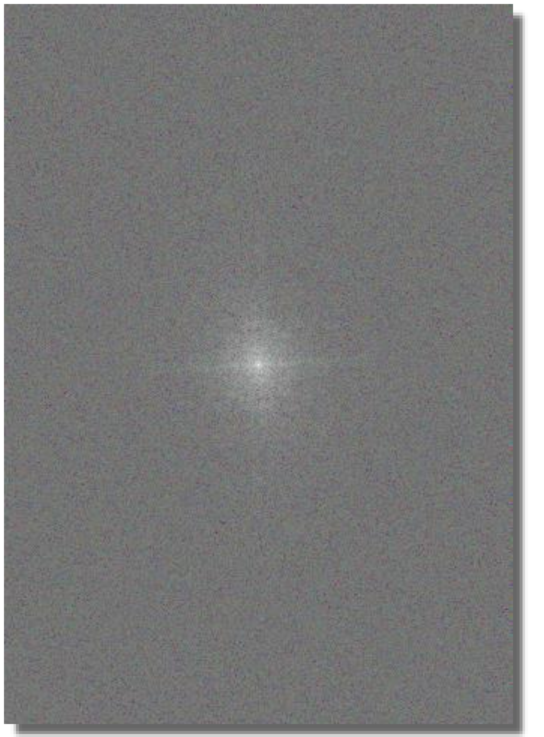


image PS



noise PS



image+noise PS

Denoising in Spatial Domain – (cont.)

Additive Noise: Example

displayed:

$$\log\left\{\left|\mathscr{F}(\mathbf{I})\right|^2 + 1\right\}$$

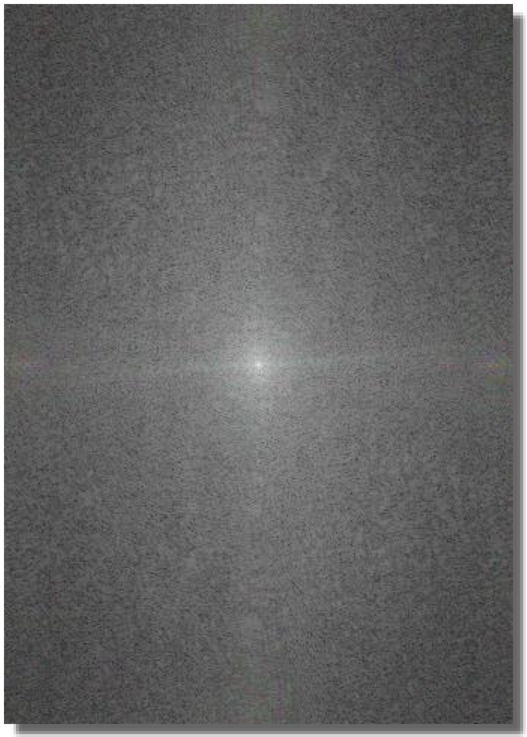
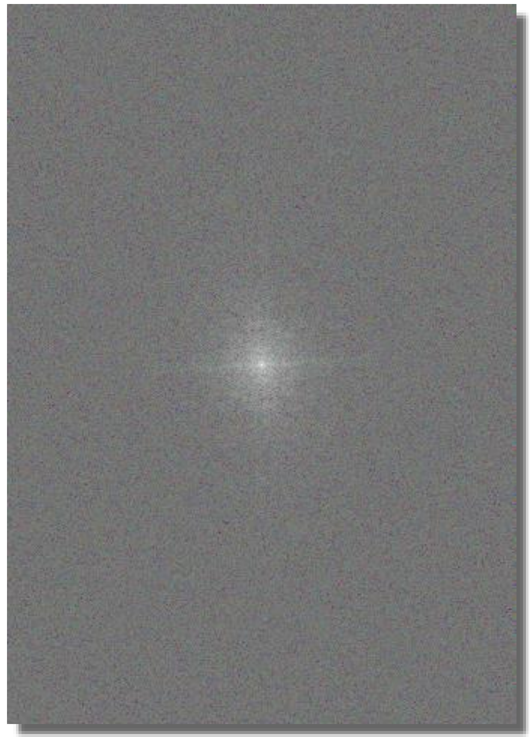


image PS



image+noise PS

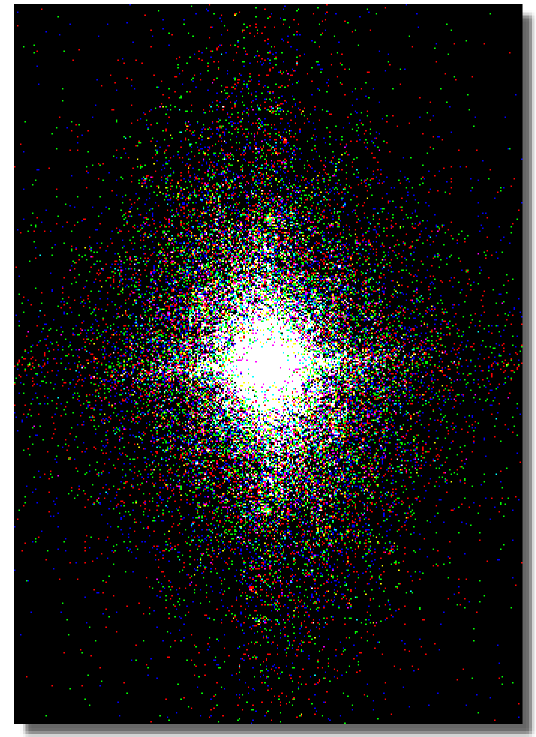
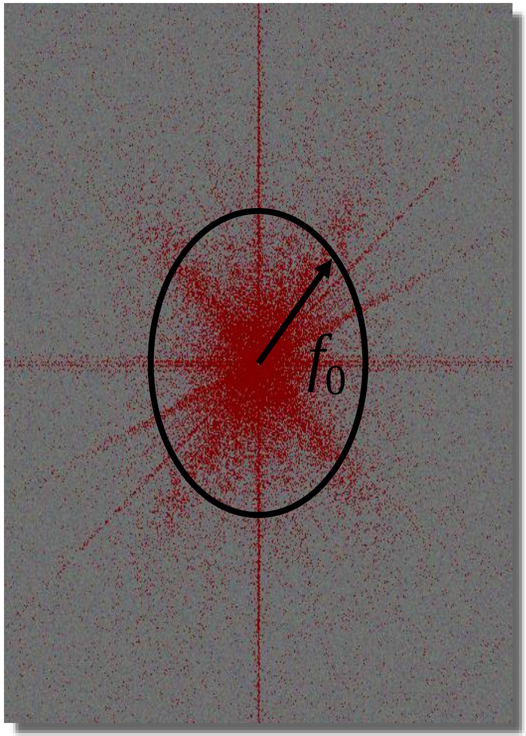


image PS > noise PS

Denoising in Spatial Domain – (cont.)

Additive Noise: Reduce with Blurring?



red indicates image > noise

At some frequency, f_0, there are more components where the noise power is greater than the image power.

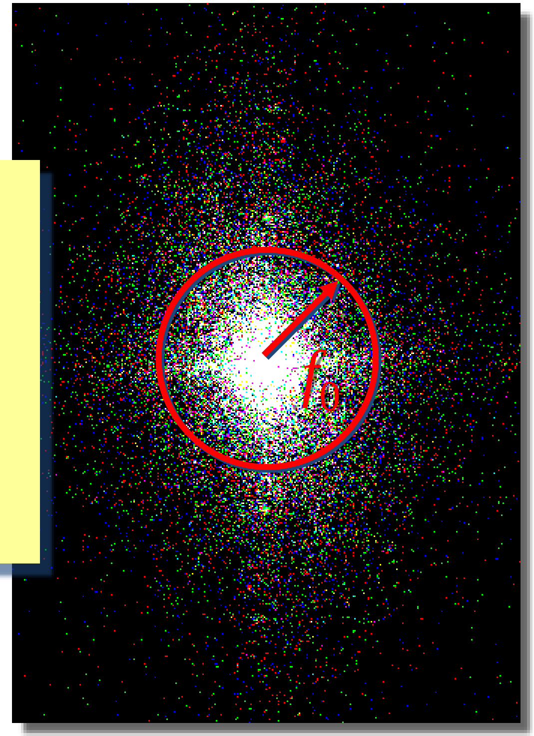
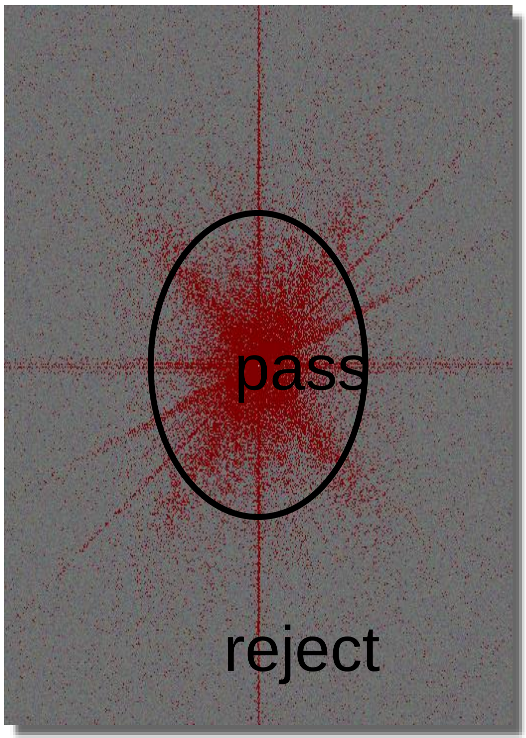


image PS > noise PS

Denoising in Spatial Domain – (cont.)

Additive Noise: Reduce with Blurring?



red indicates image > noise

Thus, it makes sense to apply a LPF with cutoff f_0, (a blurring filter) to the images and see what happens.

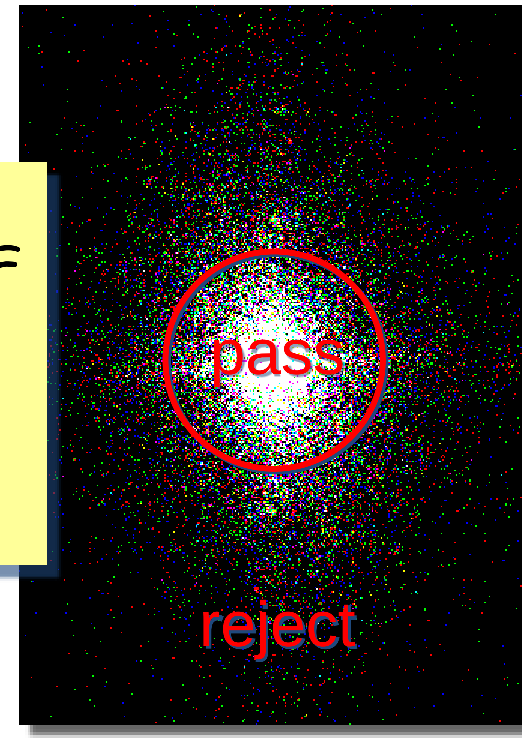


image PS > noise PS

Denoising in Spatial Domain – (cont.)

Additive Noise: Reduce with Blurring?

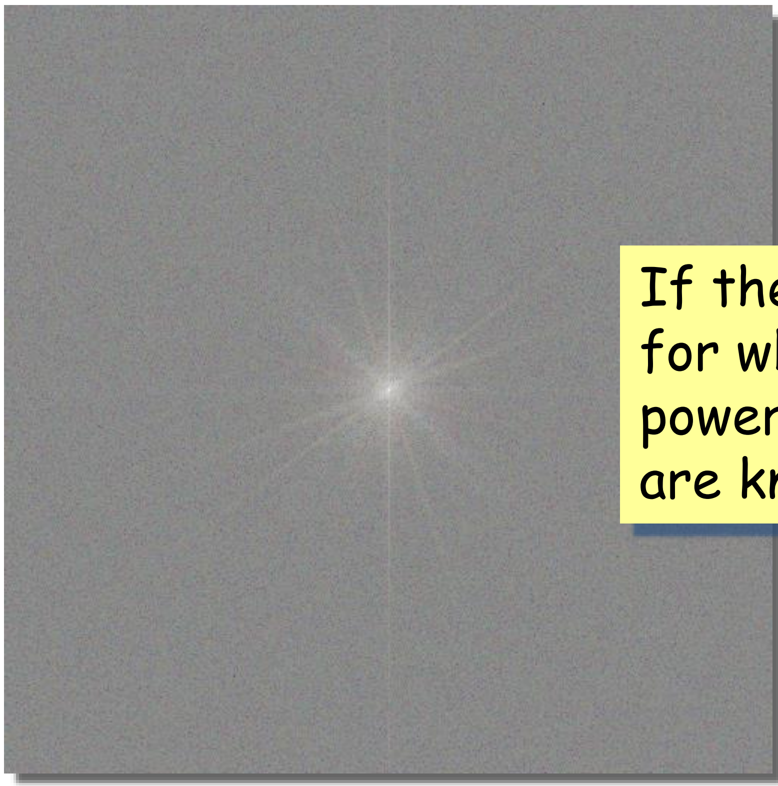
The result is actually no better. There's less noise but the blurring looks worse.

PS of Gaussian blurred image

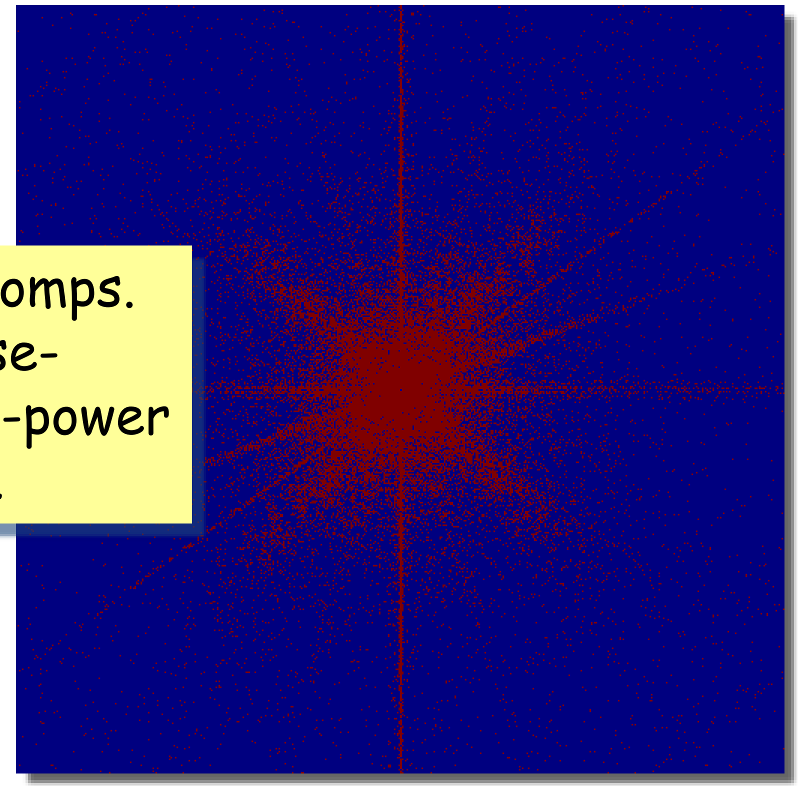
Gaussian Blurred Image

Denoising in Spatial Domain – (cont.)

Additive Noise: Reduce with Noise Masking?



power spec. of noisy image



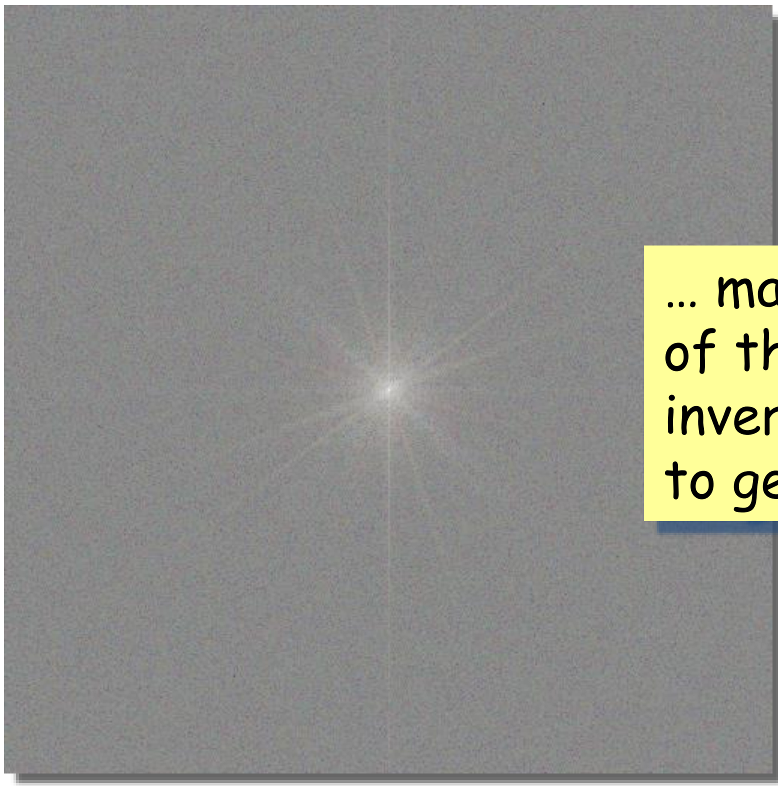
red: image > noise
blue: image < noise

If the freq. comps. for which noise-power > image-power are known(1)…

(1) of course they almost never are.

Denoising in Spatial Domain – (cont.)

Additive Noise: Reduce with Noise Masking?



power spec. of noisy image

… mask those out of the FT and invert the result to get…

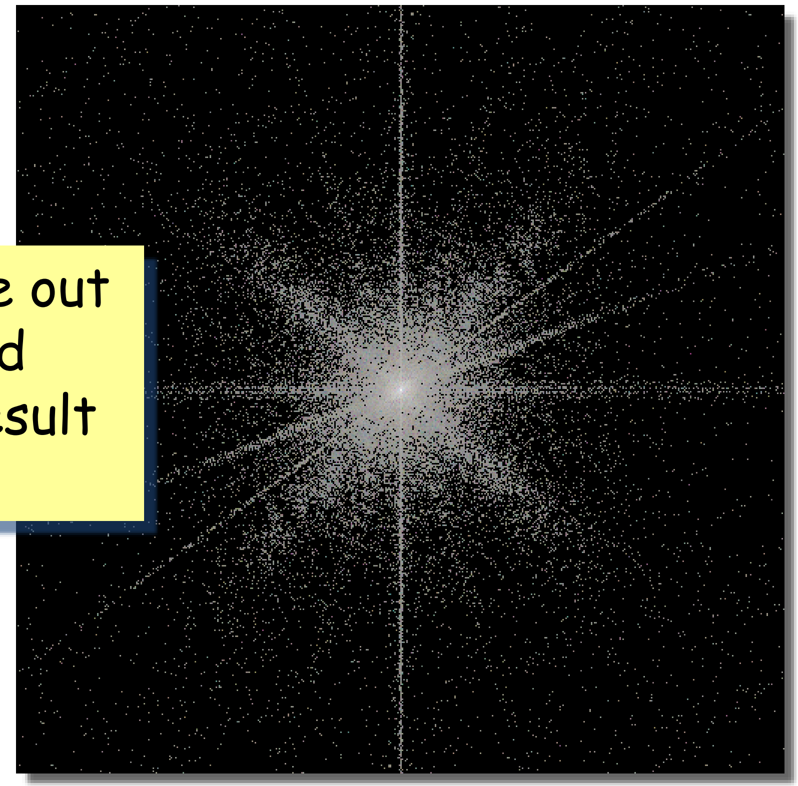
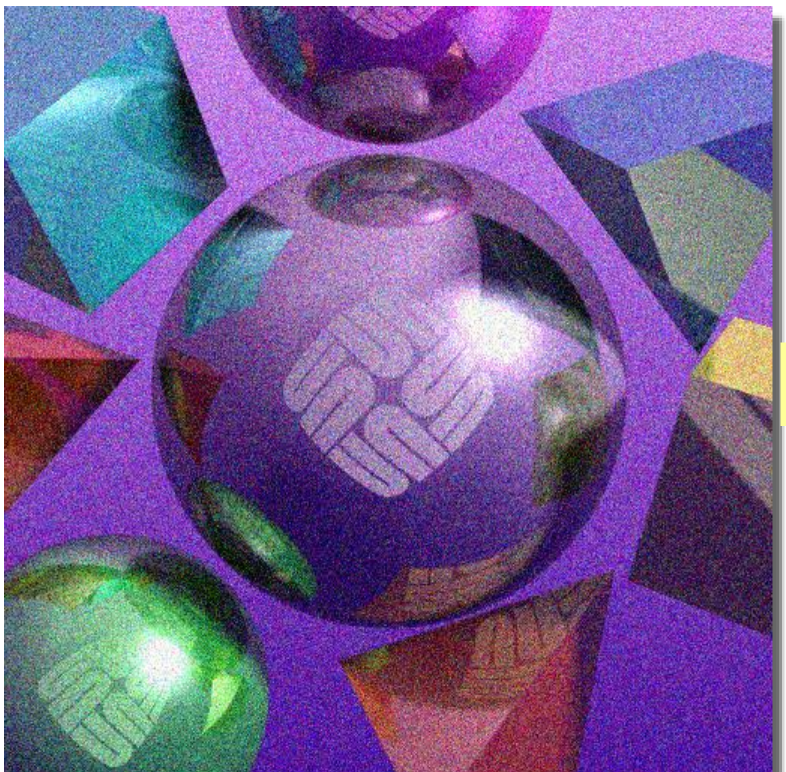


image < noise masked out

Denoising in Spatial Domain – (cont.)

Additive Noise: Reduce with Noise Masking?



noisy image

… this:

noise-masked mage

Denoising in Spatial Domain – (cont.)

Additive Noise: Reduce with Noise Masking?

Although the noise-masked image looks better than the blurred one, it is still noisy. Moreover, this example is unrealistic because we know the exact noise power spectrum. In any real case we will at most know its statistics.

blurred noisy image

noise-masked image

Denoising in Spatial Domain – (cont.)

Remember the only degradation is noise

$$g(x, y) = f(x, y) + \eta(x, y)$$

- **Spatial filtering is the best method of choice when only additive random noise of a possibly modelled distribution is present.**
- **Why? Now we know why.**
- **We will take a look at some example filters.**

Denoising in Spatial Domain – (cont.)

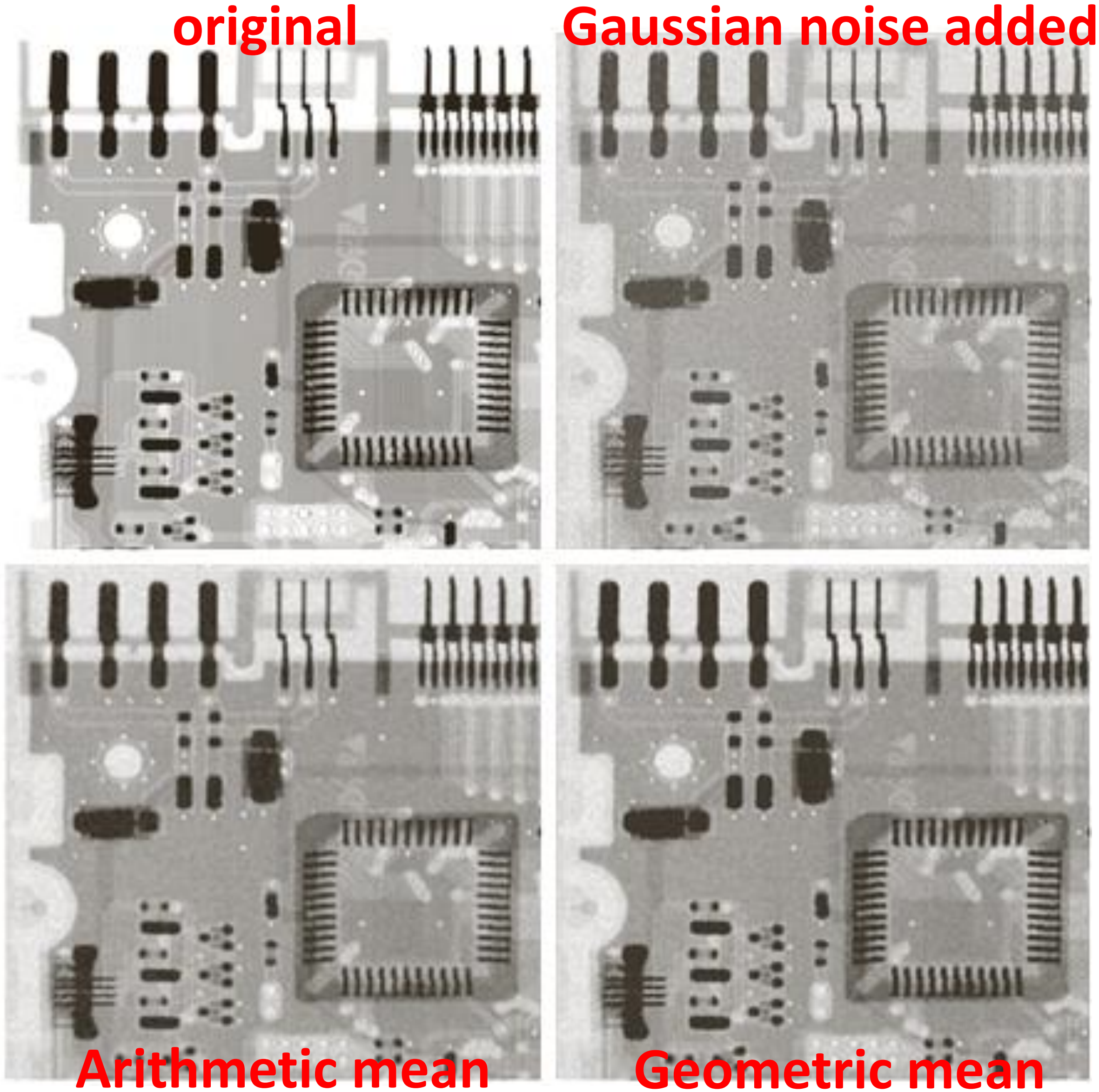
1. Mean Filter

Filter	Restored Image	Effect
Arithmetic	$\hat{f}(x,y) = \frac{1}{mn}\sum_{(s,t)\in S_{xy}} g(s,t)$	Smoothing → reduces several types of noise.
Geometric	$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t)\right]^{\frac{1}{mn}}$	Reduces Gaussian and Uniform. Smoothing with less details lost.
Harmonic	$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$	Reduces Gaussian and others. Works well for salt, but fails for pepper.
Contraharmonic (of order Q)	$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$	$Q > 0$ → eliminates pepper. $Q < 0$ → eliminates salt. $Q = 0$ → Arithmetic. $Q = -1$ → Harmonic.

Denoising in Spatial Domain – (cont.)

1. Mean Filter

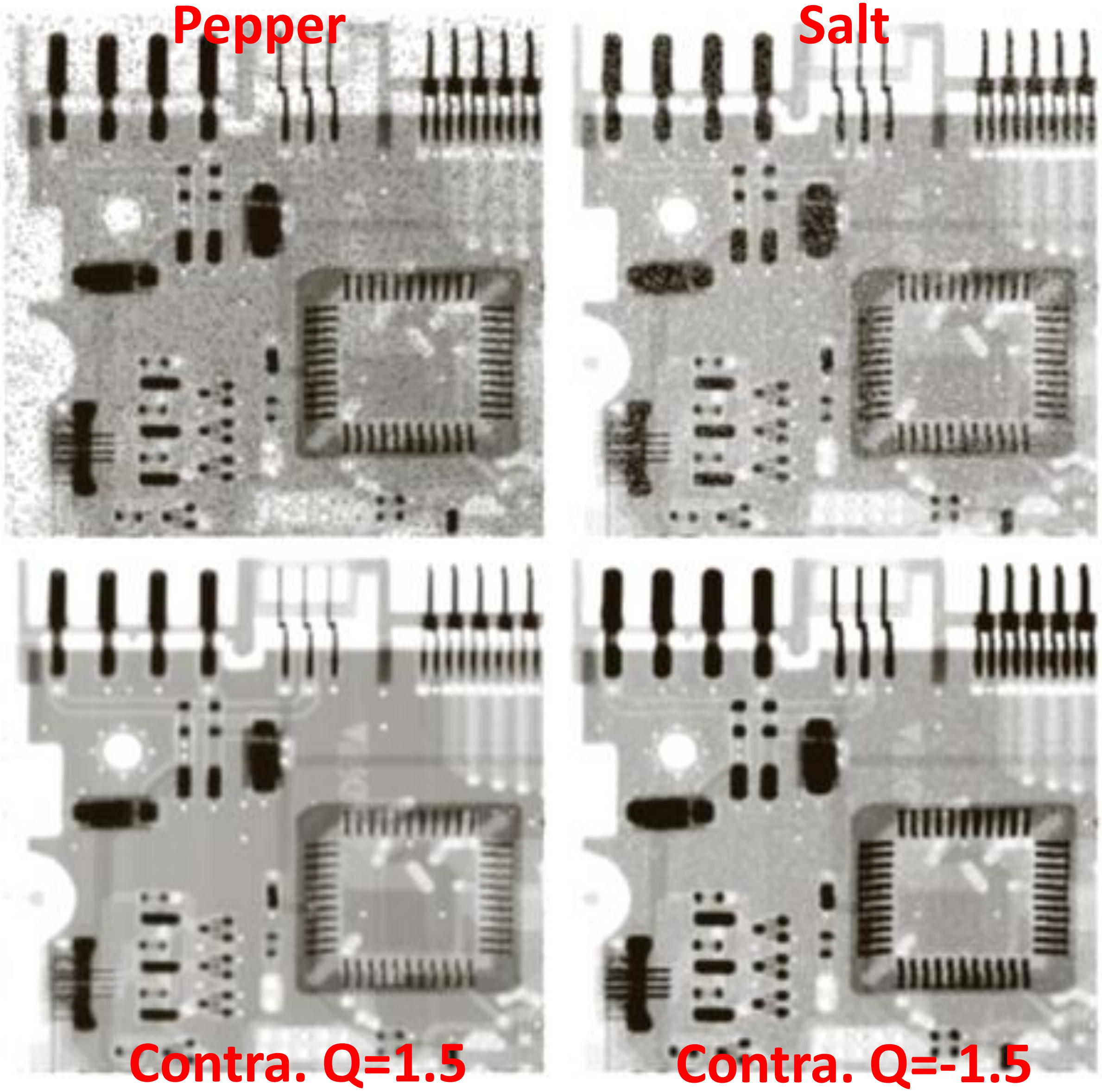
Example



Denoising in Spatial Domain – (cont.)

1. Mean Filter

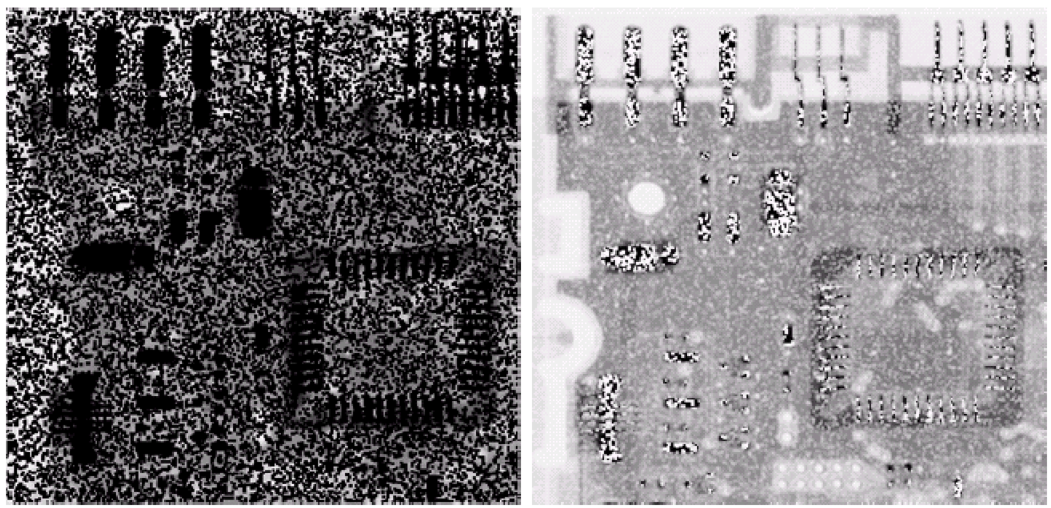
Example



Denoising in Spatial Domain – (cont.)

1. Mean Filter

Example



Choosing the wrong polarity for the contraharmonic filter can have very bad results.

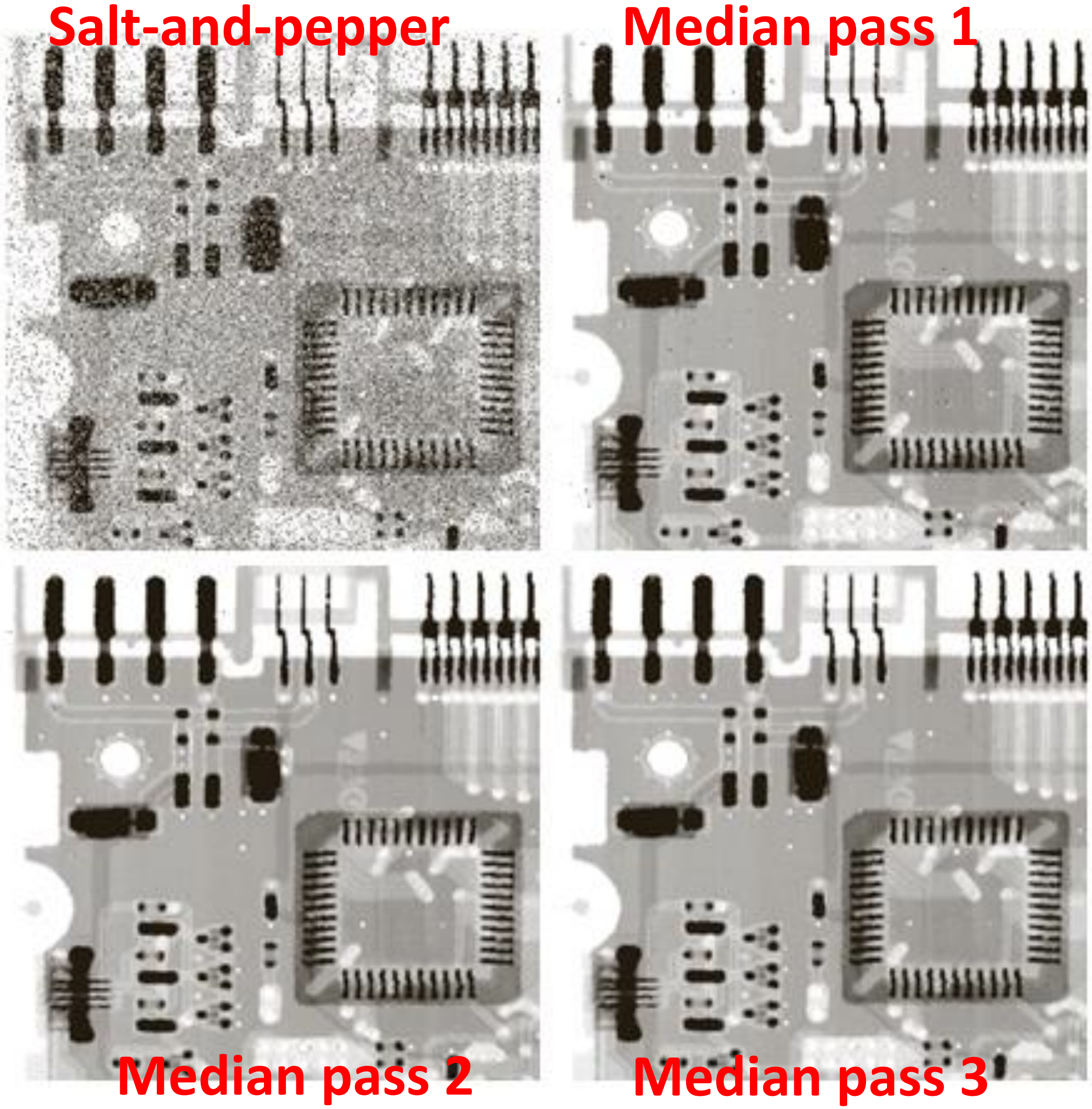
Denoising in Spatial Domain – (cont.)

2. Order-Statistic Filters (nonlinear)

Filter	Restored Image	Effect
Median	$f(x,y) = median_{(s,t)\in S_{xy}}\{g(s,t)\}$	Excellent noise removal with less blurring. Very effective for salt-and-pepper.
Max	$f(x,y) = \ max_{(s,t)\in S_{xy}}\{g(s,t)\}$	Reduces pepper.
Min	$f(x,y) = \ min_{(s,t)\in S_{xy}}\{g(s,t)\}$	Reduces salt.
Midpoint	$f(x,y) = \frac{1}{2}\left\{max_{(s,t)\in S_{xy}}\{g(s,t)\} + min_{(s,t)\in S_{xy}}\{g(s,t)\}\right\}$	Combines ranking and averaging Reduces Gaussian and Uniform.
Alpha-trimmed Mean (trim d/2 from both ends)	$\hat{f}(x,y) = \frac{1}{mn-d}\sum_{(s,t)\in S_{xy}} g_r(s,t)$	$d = 0$ → arithmetic mean. $d = mn - 1$ → median. Other d → good for combination of salt-and-pepper, and Gaussian.

Denoising in Spatial Domain – (cont.)

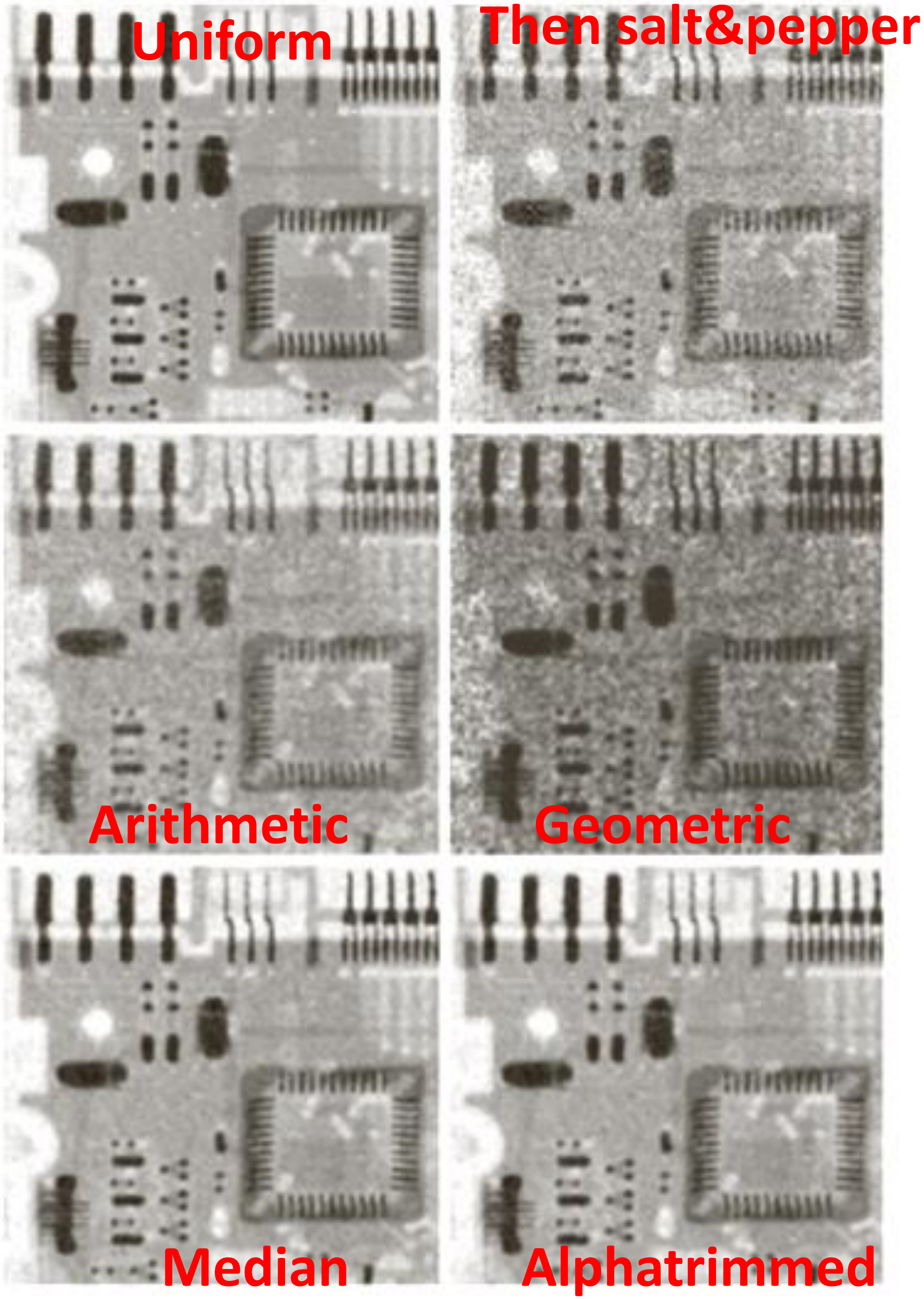
2. Order-Statistic Example



Denoising in Spati

2. Order-Statistic

Example



Useful "Goodness" Measures

- Signal-to-Noise Ratio

$$SNR = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|N(u,v)|}$$

- Mean Square Error

$$MSE = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)-\hat{f}(x,y)\right]^2$$

- Signal-to-Noise Ratio in the Spatial Domain

$$SNR = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\hat{f}^2(x,y)}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)-\hat{f}(x,y)\right]^2}$$

Important Notes

- Quantitative metrics do NOT necessarily relate well to the perceived image quality.
- Optimality criteria, while satisfying from a theoretical point of view, are NOT related to the dynamics of visual perception.
- Hence, the choice of an algorithm over the other will almost always be determined (at least partially) by the perceived visual quality of the resulting image.
- Automatically determined restoration filters yield inferior results to manual adjustment of filter parameters.

Summary

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$$
$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

Input Image *f (x, y)* → **Restoration (H=1)** → **Output (Better) Image** *g (x, y)*

Uncorrelated (random) noise

Spatial Domain

$$g(x,y) = f(x,y) + \eta(x,y)$$

Additive/impulse	Filters
Gaussian	Arithmetic
Rayleigh	Geometric
Gamma	Harmonic
Exponential	Contra-harmonic
Uniform	Adaptive
Salt & pepper	Median
	Min
	Max
	Mid-Point
	Alpha-Trim
	Adaptive

Correlated (periodic) noise

Transform Domain

$$G(u,v) = F(u,v) + N(u,v)$$

Periodic	Selective Filters
Concentrated bursts of energy in the FT	Band reject Notch

Goodness measures

$$SNR = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|}{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|N(u,v)|}$$

$$MSE = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y) - \hat{f}(x,y)\right]^2$$

$$SNR = \frac{\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\hat{f}^2(x,y)}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y) - \hat{f}(x,y)\right]^2}$$

Summary

- Discrete Fourier Transform forward/inverse.
- Properties of FT.
- Applications of DFT in IP.
- Highpass vs Lowpass filters.
- Noise removal

Covered

Szeliski: Ch. 3.4

Gonzalez: Ch. 4, 5.4

7, 8, 9, 10

Next lecture Reading

Feature Detection and Matching

Szeliski: Ch. 4

Gonzalez: Ch. 10, 11

(extra) F&P: Ch. 9

References

- Gonzalez and Woods, *Digital Image Processing*, 2008.

- Peters, Richard Alan, II, "Fourier Transform" and "Frequency Filtering", Lectures on Image Processing, Vanderbilt University, Nashville, TN, April 2008, Available on the web at the Internet Archive, http://www.archive.org/details/Lectures_on_Image_Processing.

- S. W. Smith, *Digital Signal Processing*, California Technical Publishing, 1999.

-http://www.cs.uregina.ca/Links/class-info/425-nova/Lab5/

- S. K. Mitra, *Digital Signal Processing*, McGrow-Hill, ????.

- J. G. Proakis, and D. G. Monalakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, Prentice Hall, ????.

- God bless Google Scholar!