Image & Video Processing

Frequency Domain Filtering

Contents

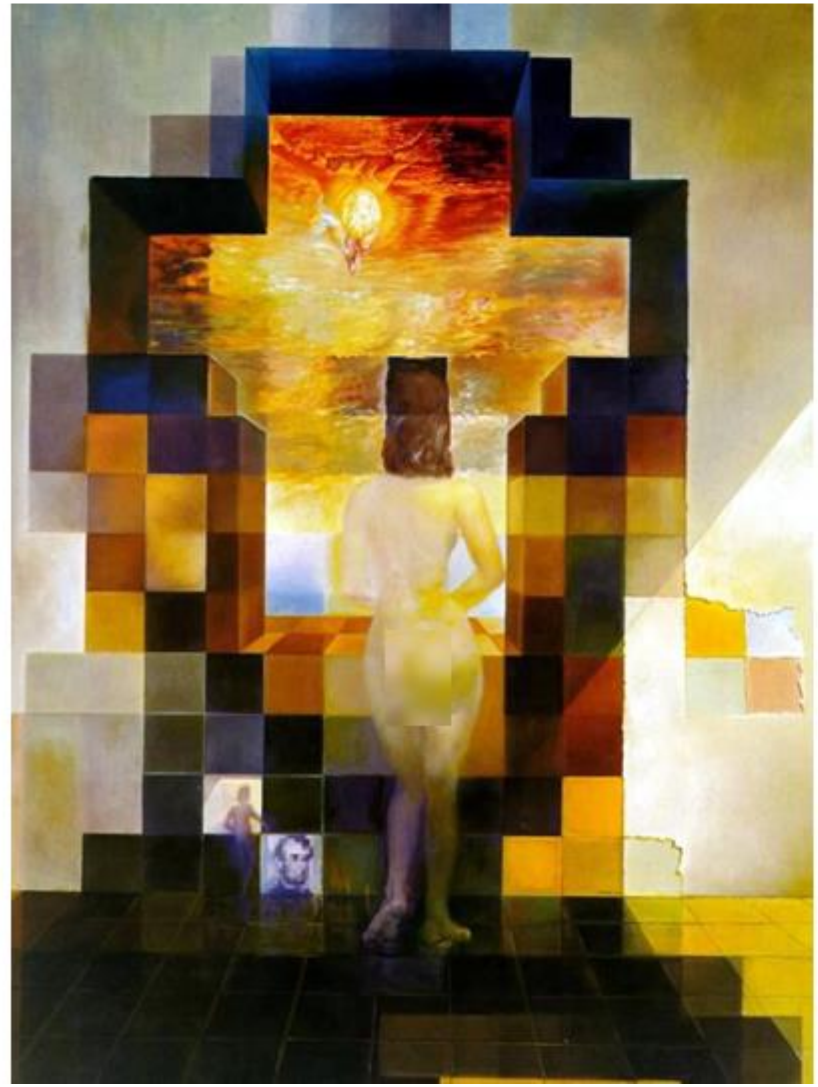
In this lecture we will introduce frequency domain/space and look at mathematical tools to go from time/spatial domain to frequency domain

- What is Frequency domain?
- Frequency in Images
- Why Frequency analysis?
- The Fourier series
- The Fourier transform

What do you see?

Salvador Dali

“Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln”, 1976



What do you see?

Salvador Dali

“Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln”, 1976

Decomposing an image: ***basis sets***

(edit from Wikipedia) A basis B of a vector space V is a linearly independent subset of V that spans V.

Suppose that $B=*\ \boldsymbol{v1}$, … , $\boldsymbol{vn}$ is a finite subset of a vector space V over a field F, then B is a basis if it satisfies the following conditions:

1. Linear independence
2. Spanning property

Decomposing an image: ***basis sets***

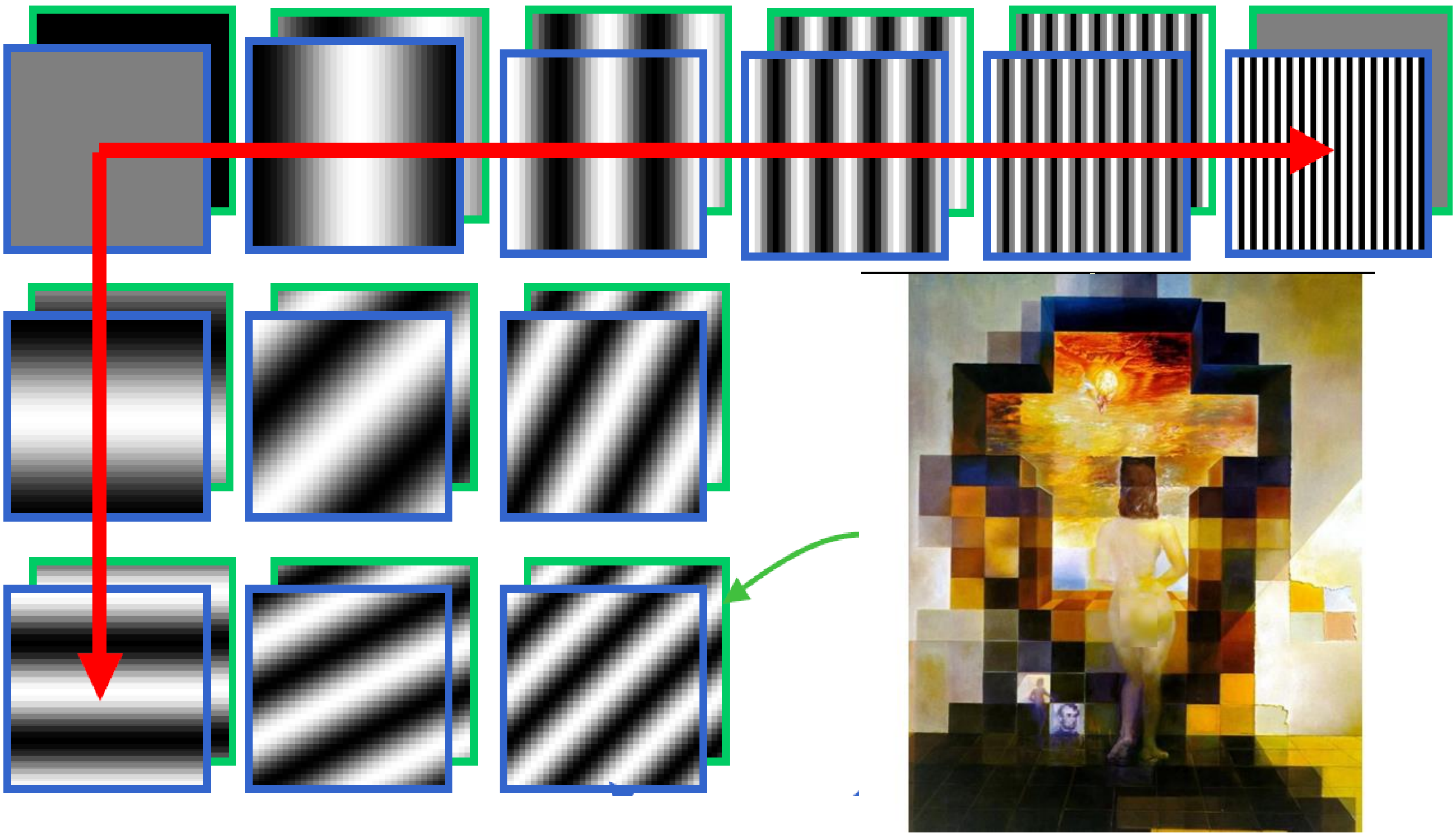
- Consider an image as a point in a NxN size space – can rasterize into a single vector

$$[\, x_{00}\; x_{10}\; x_{20}\; \dots\; x_{(n-1)0}\; x_{10} \dots\; x_{(n-1)(n-1)} \,]^T$$

- The “normal” basis is just the vectors:
  - $[\, 0\; 0\; 0\; 0 \dots 0\; 1\; 0\; 0\; 0\; 0\; 0\; 0\; 0 \dots\; 0 \,]^T$
  - Independent
  - Can create any image
  - Not very helpful…

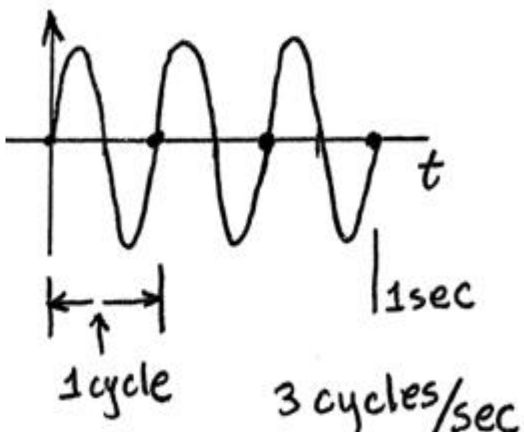
A nice set of basis

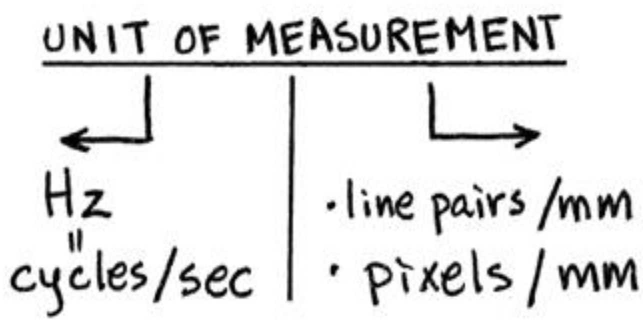
Fast vs. slow changes in the image in different directions

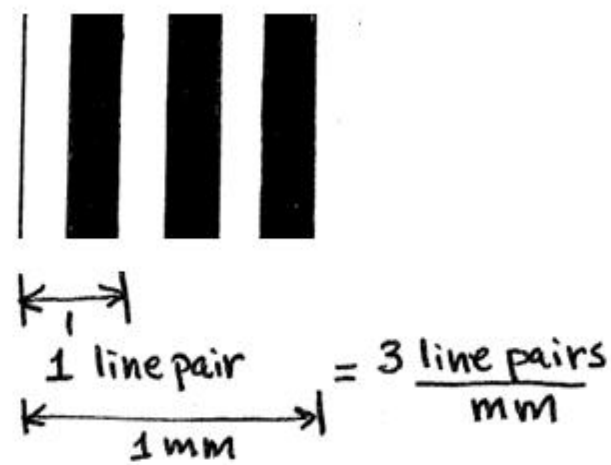


Frequency Domain/Space

- The term frequency comes up a lot in physics, as some variation in time, describing the characteristics of some periodic motion or behaviour.







Frequency in the Time domain (1D)

Frequency in the Spatial domain (2D)

Frequency Domain/Space

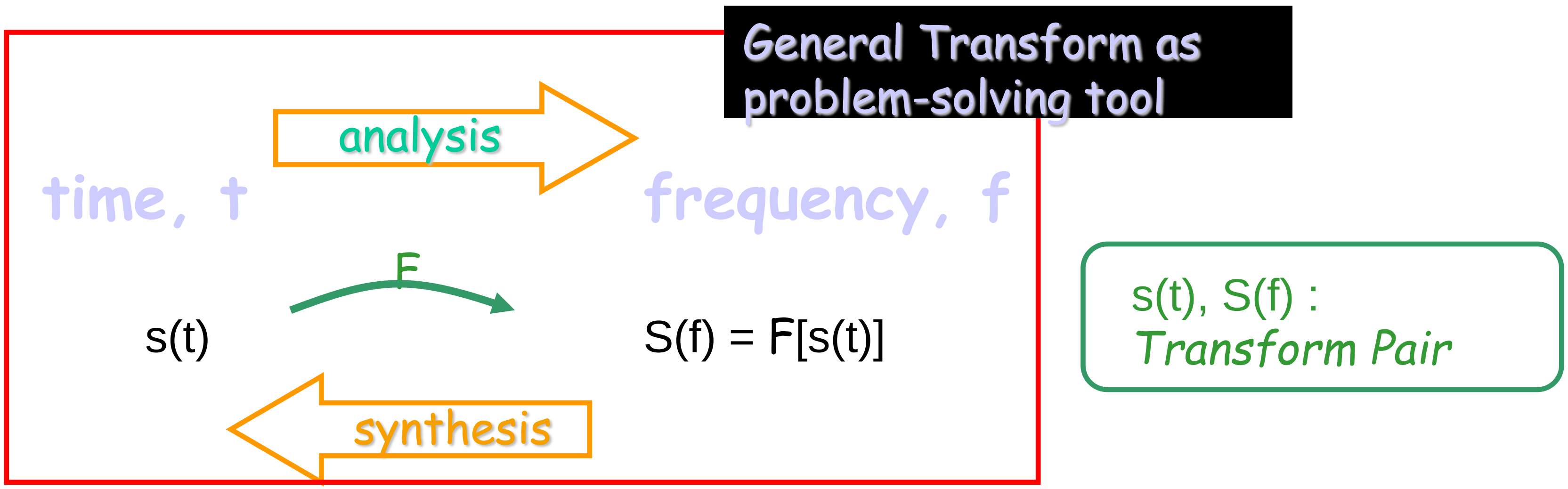
- A signal (wave in time or image in spatial domain) can be decomposed or separated into a sum of sinusoids of different frequencies, amplitude and phase.
- *1D Audio Example:* Consider a complicated sound played on a piano or a guitar. We can describe this sound in two related ways:
  - Temporal Domain: Sample the amplitude of the sound many times a second, which gives an approximation to the sound as a function of time
  - Frequency Domain: Analyse the sound in terms of the pitches of the notes, or the amplitude of each frequency (fundamental plus harmonics) which make up the sound up.

Frequency in Image?

- Image can be represented in two ways:
  - Spatial Domain: Brightness along a line can be recorded as a set of values measured at equally spaced distances apart (represented as 2D array of pixel measurements)
  - Frequency Domain: as a set of spatial frequency values/component (represented as 2D grid of spatial frequencies)
- In short, frequency in image has to do with intensity (brightness or color) variation across the image, i.e. it is a function of spatial coordinates, rather than time.
- Example: If an image represented in frequency space has *high* frequencies then it means that the image has sharp edges or details

Why Frequency analysis?

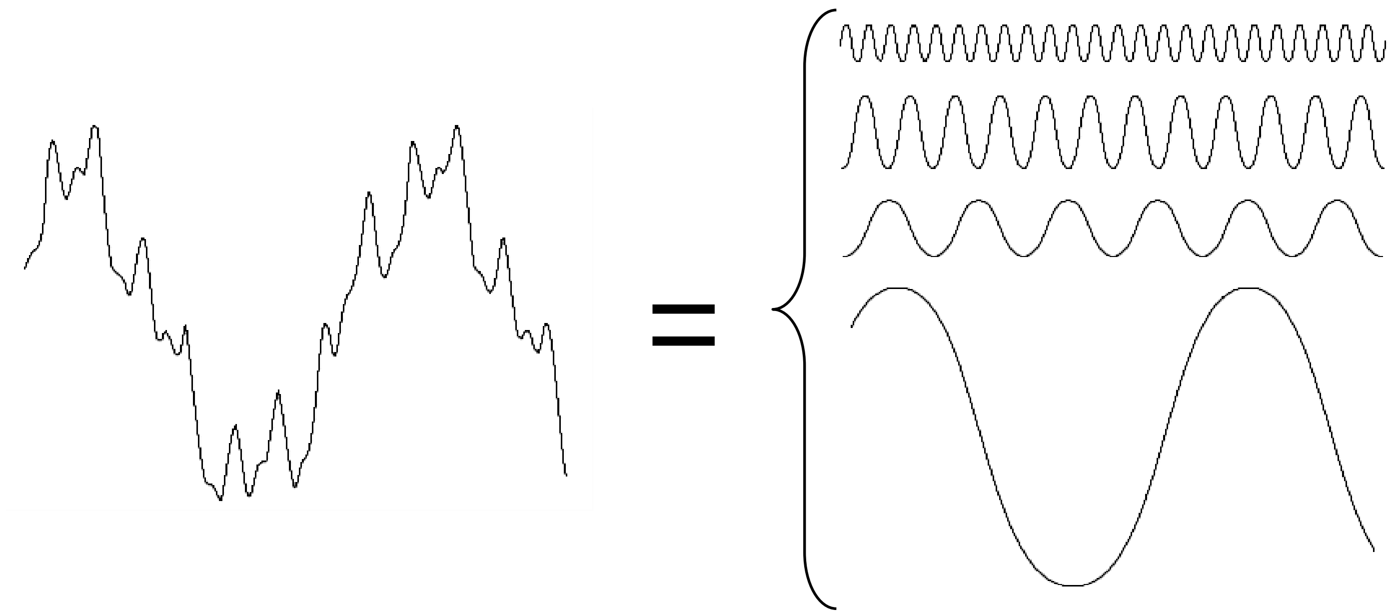
- Makes large filtering operations much faster.
- Powerful & complementary to time/spatial domain analysis techniques.
- Several transforms that makes it easy to go forwards and backwards from the spacial domain to the frequency space: Fourier, Laplace, wavelet, etc.



The Big Idea

Fourier Series and Transforms – given by mathematician **Joseph Fourier**.

One of the most important mathematical theories in modern engineering



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

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

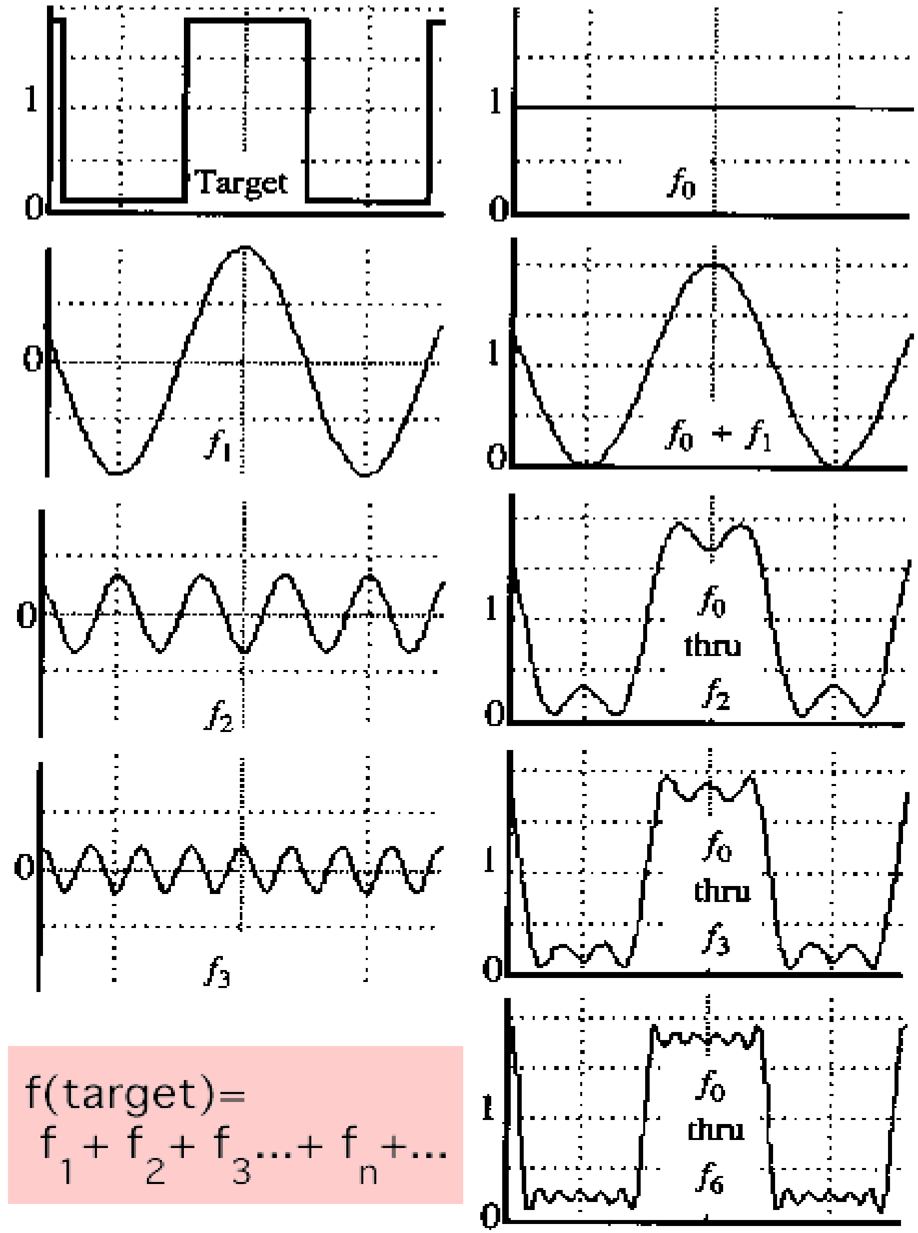
A Sum of Sinusoids

- Our building block:

$$A\sin(\omega x + \phi)$$

- Add enough of them to get any signal *f(x)* you want!

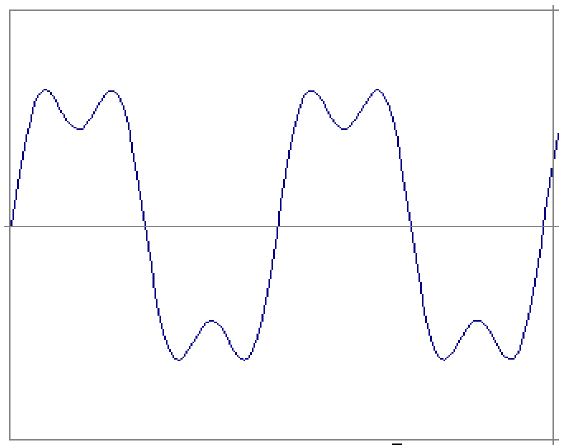
- Which one encodes the coarse vs. fine structure of the signal?



f(target)= $f_1 + f_2 + f_3 ... + f_n + ...$

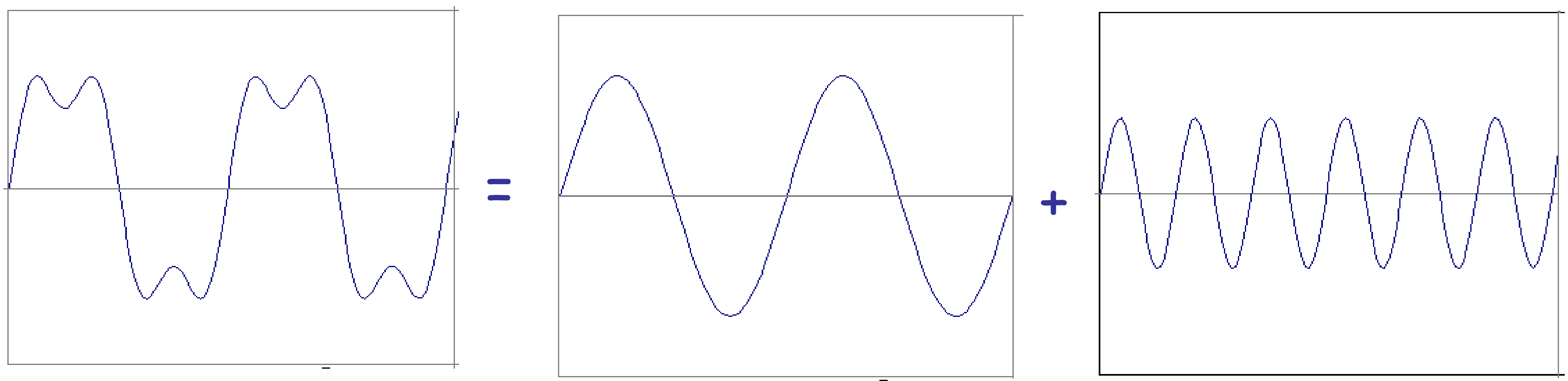
Time and Frequency

- example : $g(t) = \sin(2pi\ f\ t) + (1/3)\sin(2pi\ (3f)\ t)$



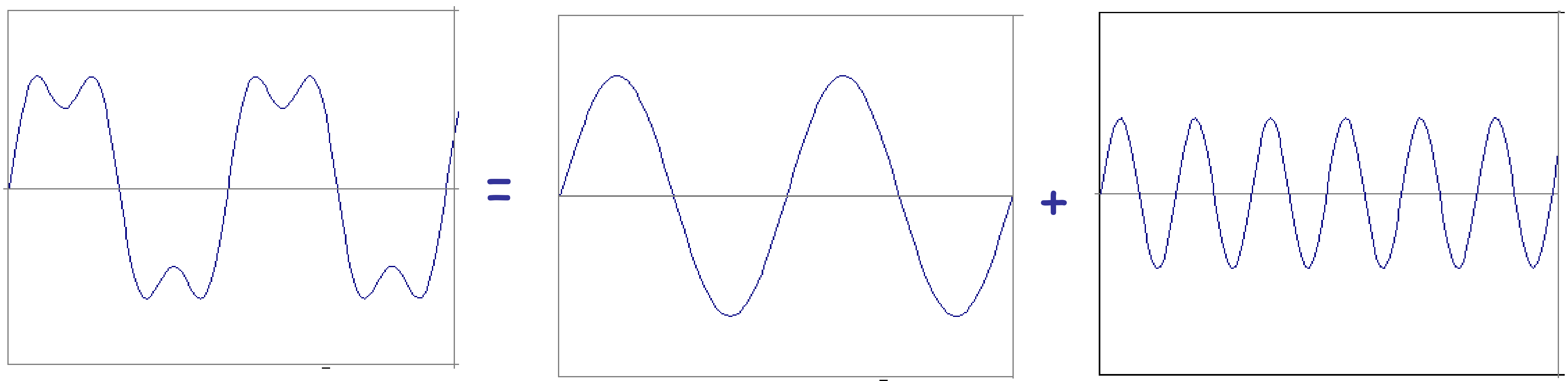
Time and Frequency

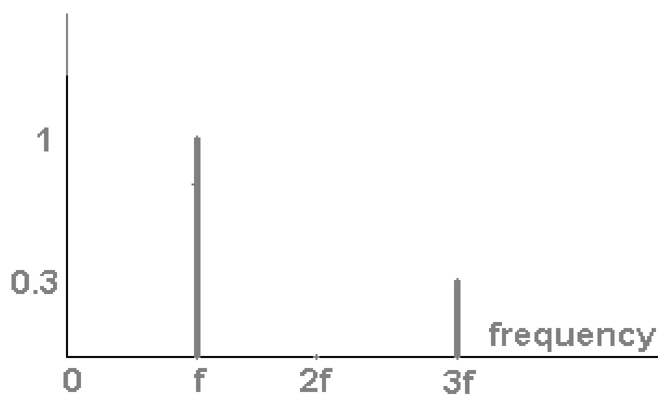
- example : $g(t) = \sin(2pi\ f\ t) + (1/3)\sin(2pi\ (3f)\ t)$



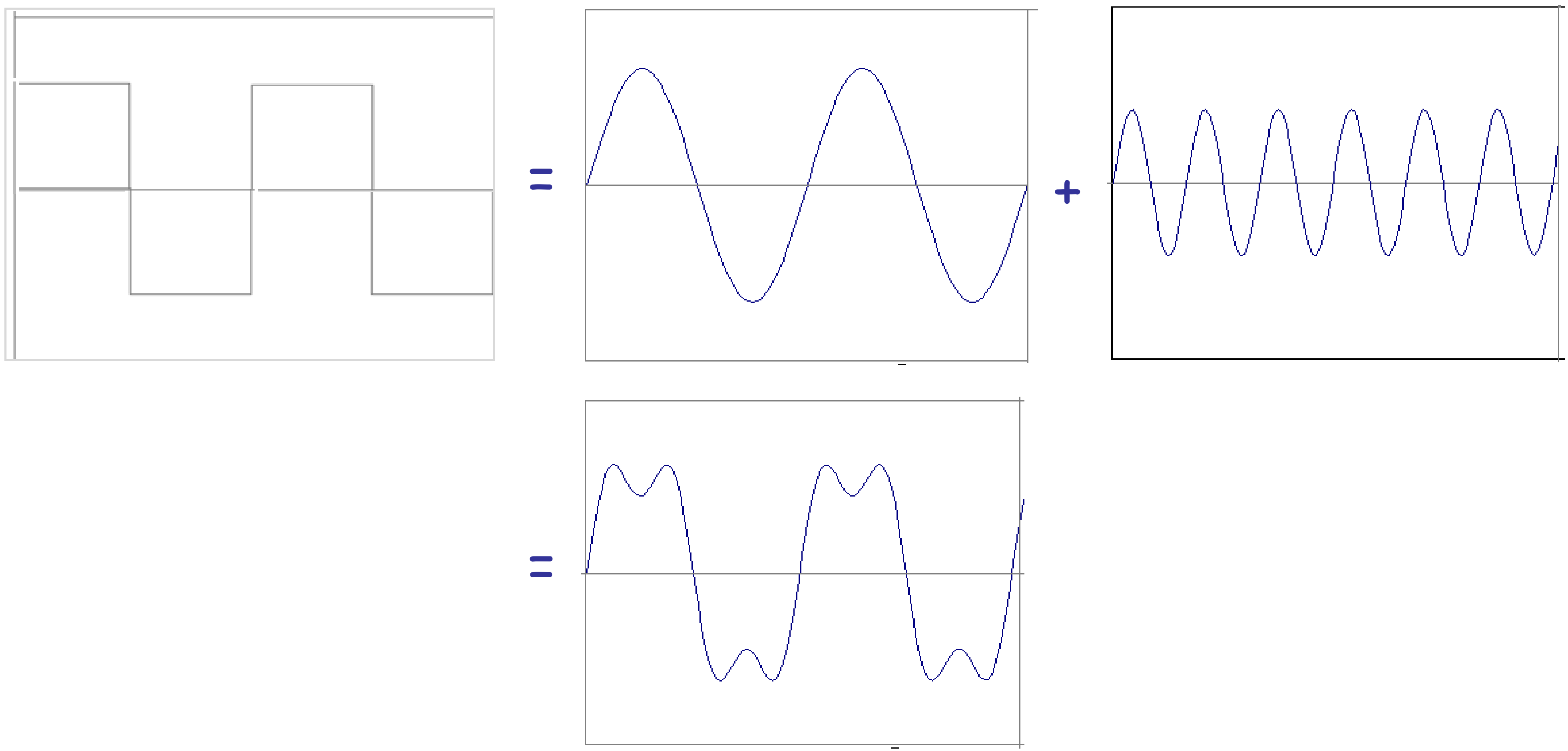
Frequency Spectra

- example : $g(t) = \sin(2pi\ f\ t) + (1/3)\sin(2pi\ (3f)\ t)$

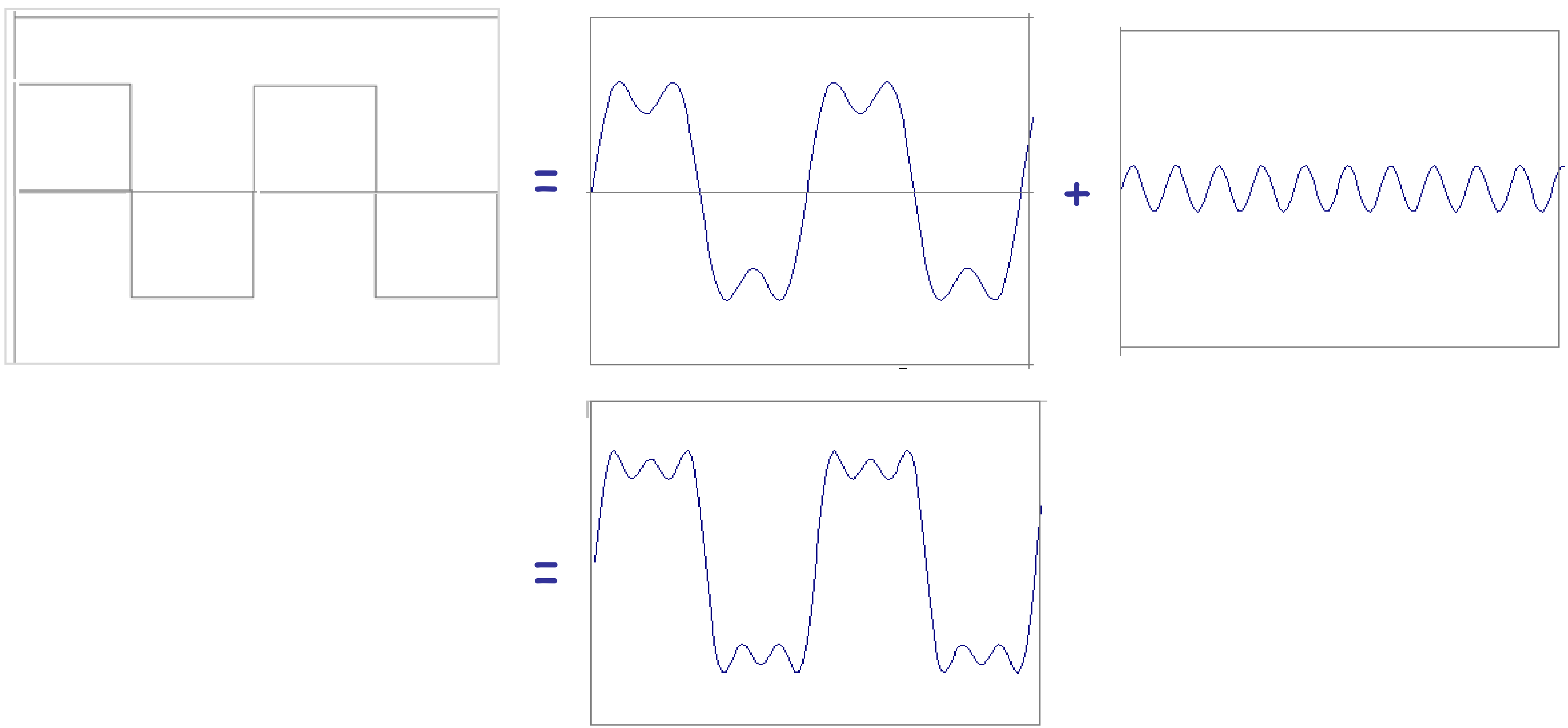




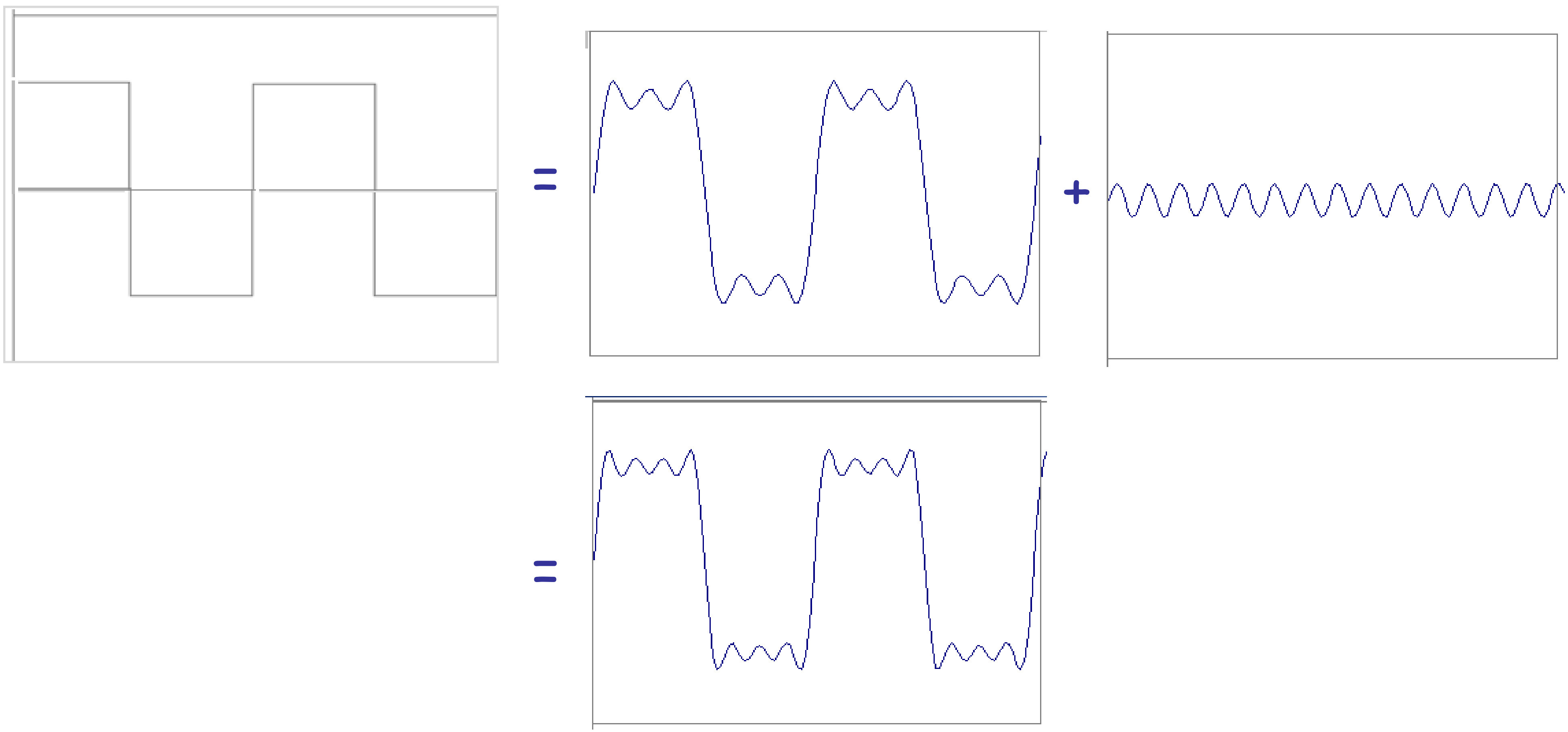
Frequency Spectra



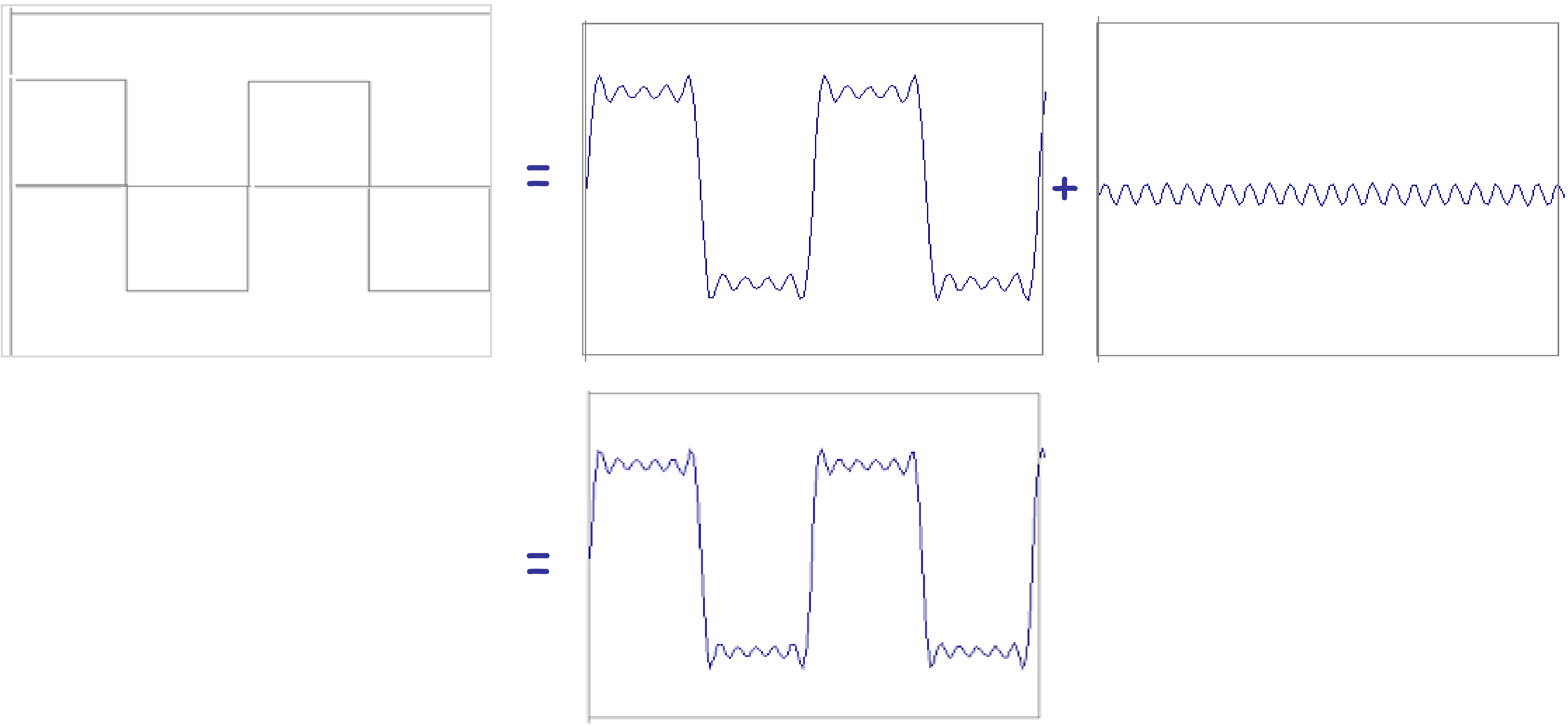
Frequency Spectra



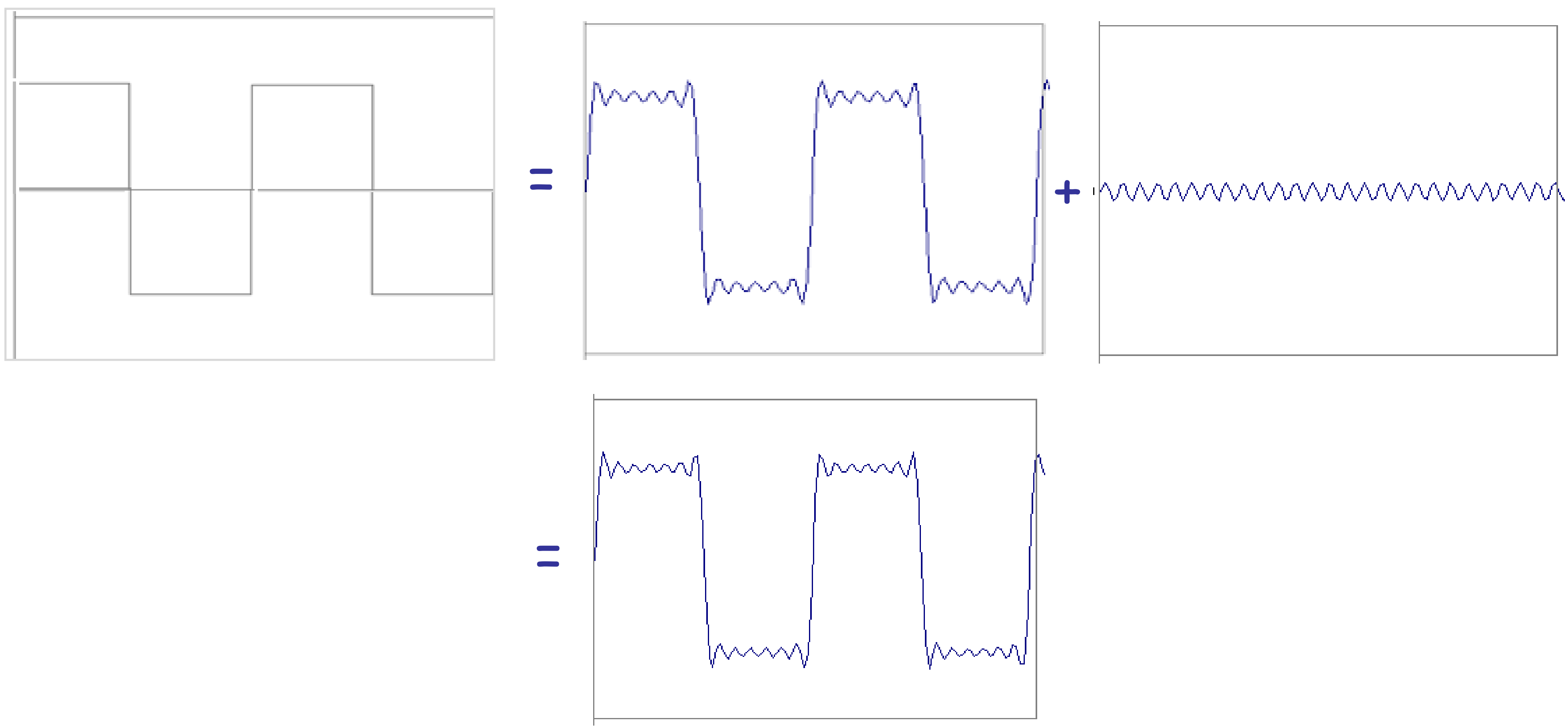
Frequency Spectra



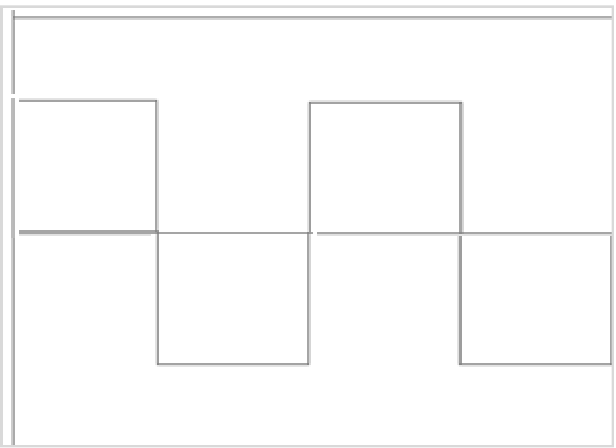
Frequency Spectra



Frequency Spectra

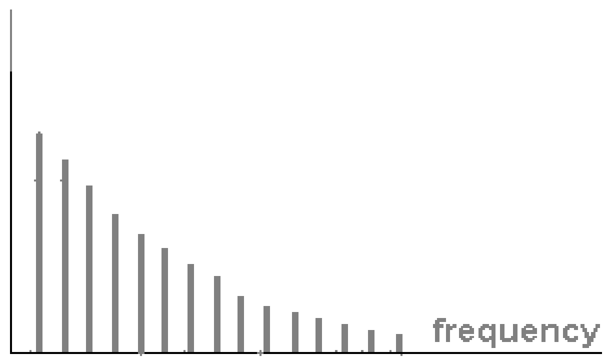


Frequency Spectra



=

$$\boxed{A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kt)}$$



Fourier Series

- A general representation of function *f(t)* that is periodic with period T, by Fourier series as:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\omega_0 t) + \sum_{n=1}^{\infty} b_n \sin(n\omega_0 t)$$

$$a_0 = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)dt$$

$$a_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\cos n\omega_0 t dt \qquad n = 1,2,\cdots$$

$$b_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\sin n\omega_0 t dt \qquad n = 1,2,\cdots$$

Fourier Transform

- We have seen that periodic signals can be represented with the Fourier series
- Can aperiodic signals be analyzed in terms of frequency components?
- Yes, and the Fourier transform provides the tool for this analysis.
- Aperiodic signals can be treated as periodic with period tending to infinity in the limit
- The major difference w.r.t. the discrete line spectra of periodic signals is that the spectra of aperiodic signals are continuous.

Fourier Transform

- We want to understand the frequency ω of our signal. So, let's reparametrize the signal by ω instead of x:

$$f(x) \longrightarrow \boxed{\text{Fourier Transform}} \longrightarrow F(\omega)$$

- For every ω from 0 to inf, ***F(ω)*** holds the amplitude A and phase ϕ of the corresponding sine

$$A\sin(\omega x + \phi)$$

  - How can F hold both? Complex number trick!

$$F(\omega) = R(\omega) + j\, I(\omega)$$

$$A = \pm\sqrt{R(\omega)^2 + I(\omega)^2} \qquad \phi = \tan^{-1}\frac{I(\omega)}{R(\omega)}$$

$$F(\omega) \longrightarrow \boxed{\text{Inverse Fourier Transform}} \longrightarrow f(x)$$

Fourier Transform – more formally

Represent the signal as an infinite weighted sum of an infinite number of sinusoids

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

Note: $e^{j\theta} = \cos\theta + j\sin\theta \qquad j = \sqrt{-1}$

Arbitrary function $\longrightarrow$ Single Analytic Expression

Spatial Domain (*x*) $\longrightarrow$ Frequency Domain (*u*)

(Frequency Spectrum *F(u)*)

Inverse Fourier Transform (IFT)

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

The Discrete Fourier Transform in 2D

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

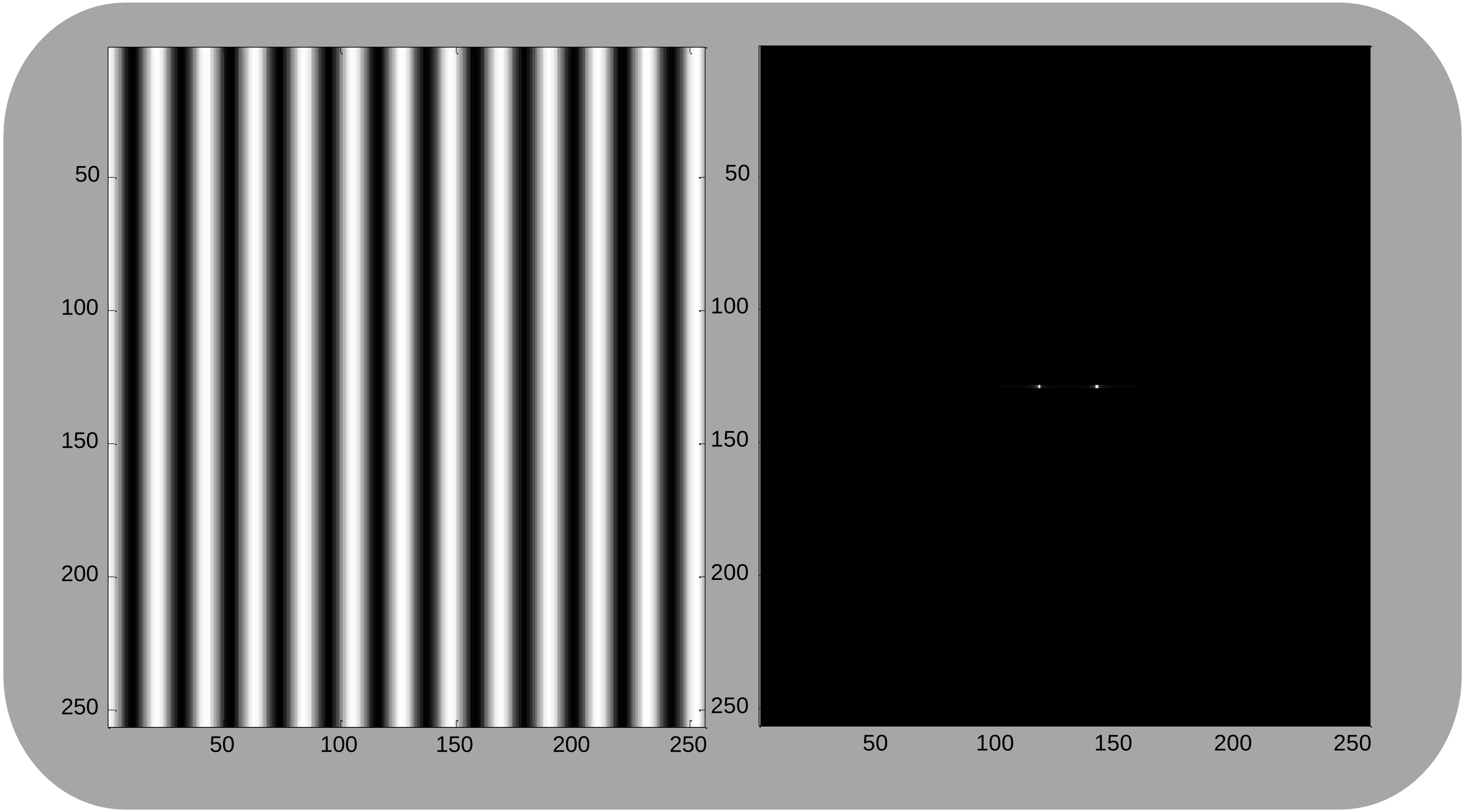
$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$

for u = 0, 1, 2…M-1 and v = 0, 1, 2…N-1.

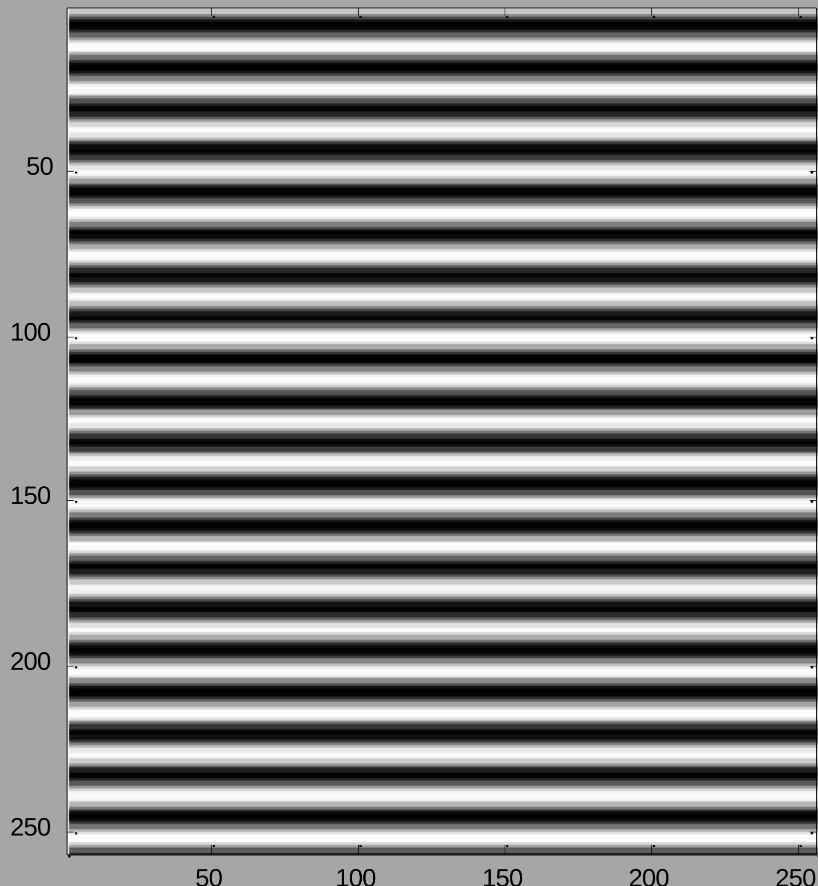
Inverse Discrete Fourier Transform is given by equation:

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

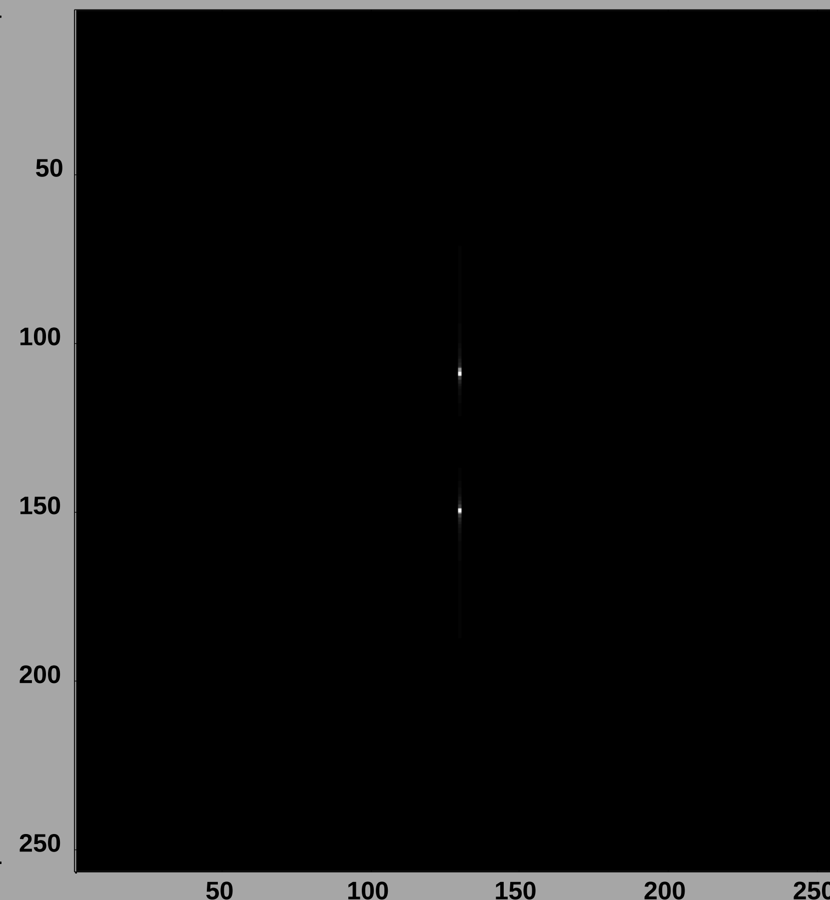
Examples


50
100
150
200
250
50
100
150
200
250
50
100
150
200
250
50
100
150
200
250

Examples

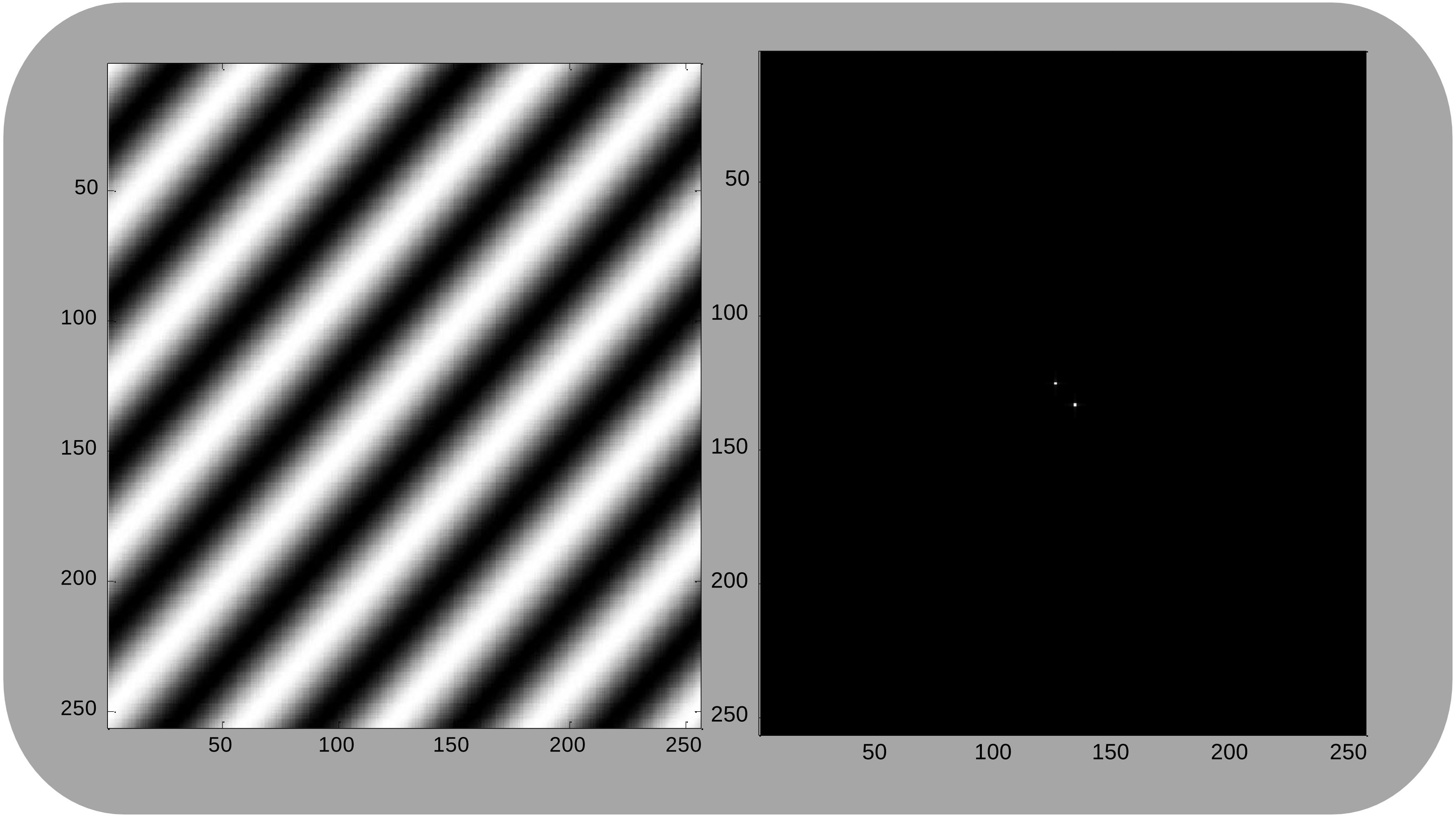


50
100
150
200
250
50
100
150
200
250



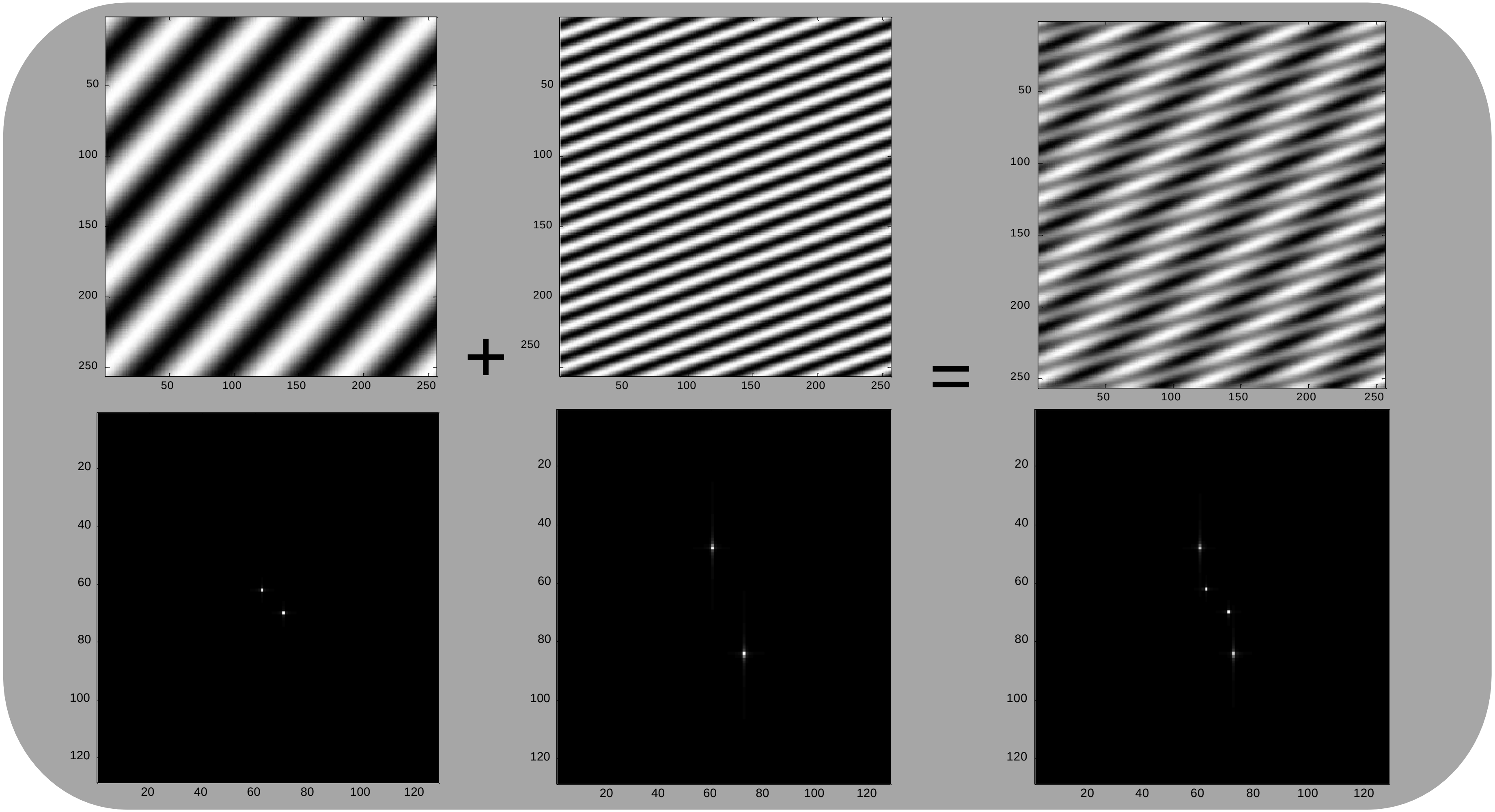
50
100
150
200
250
50
100
150
200
250

Examples



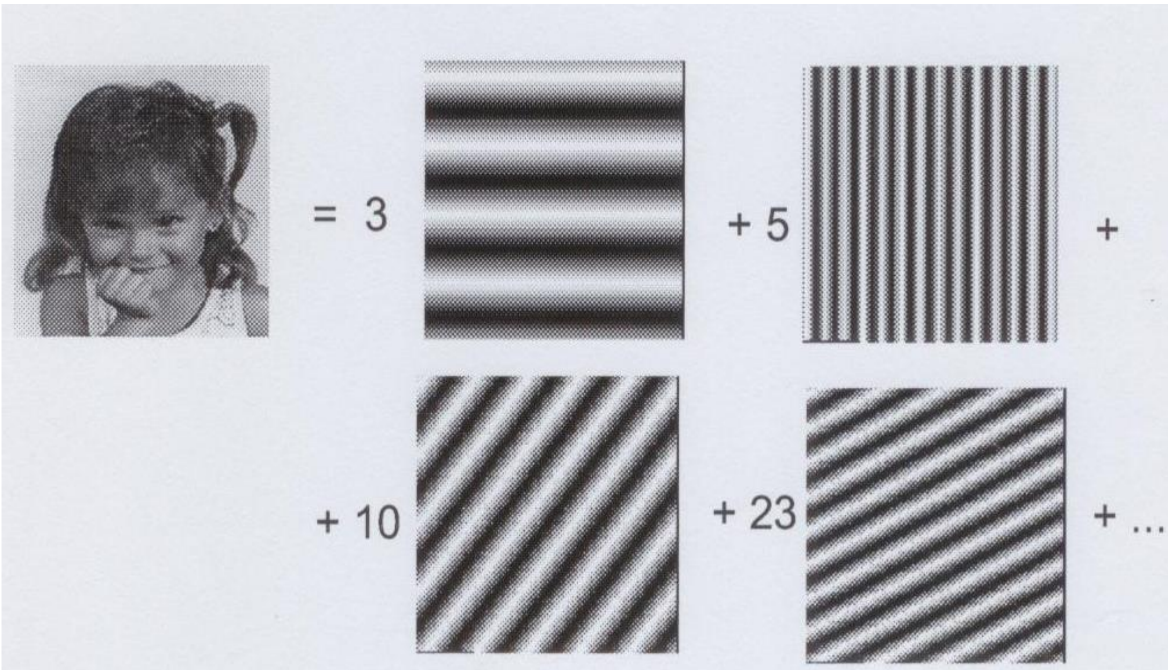
50
100
150
200
250
50
100
150
200
250
50
100
150
200
250
50
100
150
200
250

Linearity of Sum

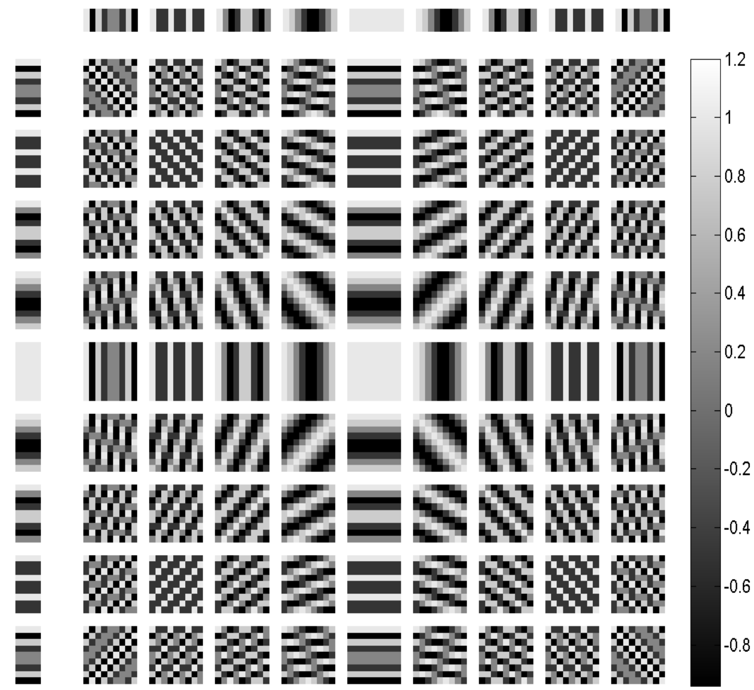


DFT & Images

Interpretation: 2D cos/sin functions

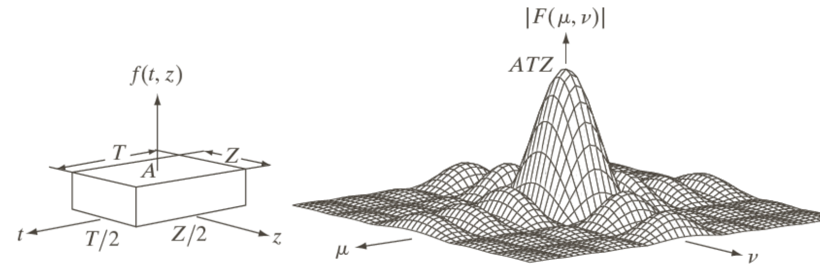


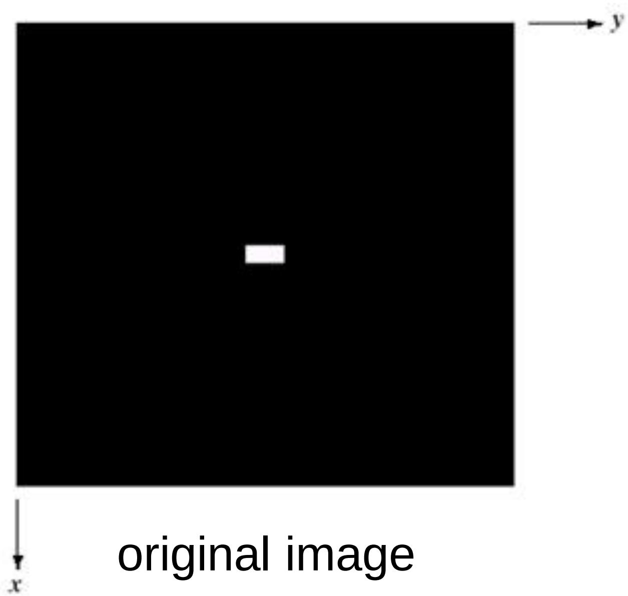
2-D Fourier basis



Visualizing DFT

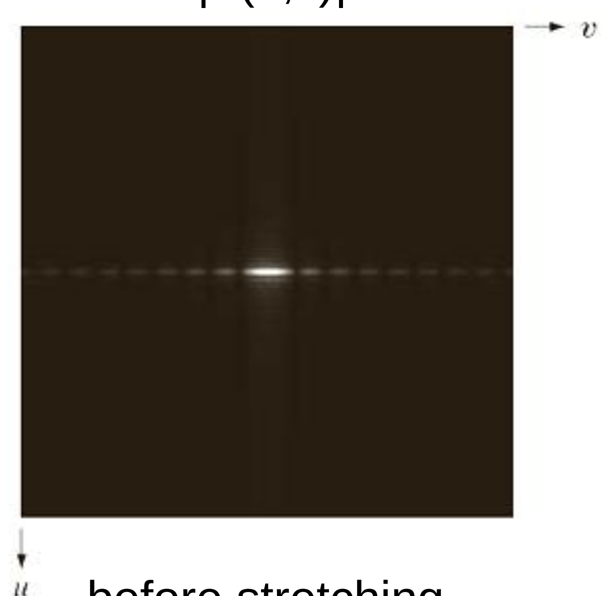
- Typically, we visualize |F(u,v)|
- The dynamic range of |F(u,v)| is typically very large
- Apply log transformation: $D(u,v) = c\ log(1 + |F(u,v)|)$





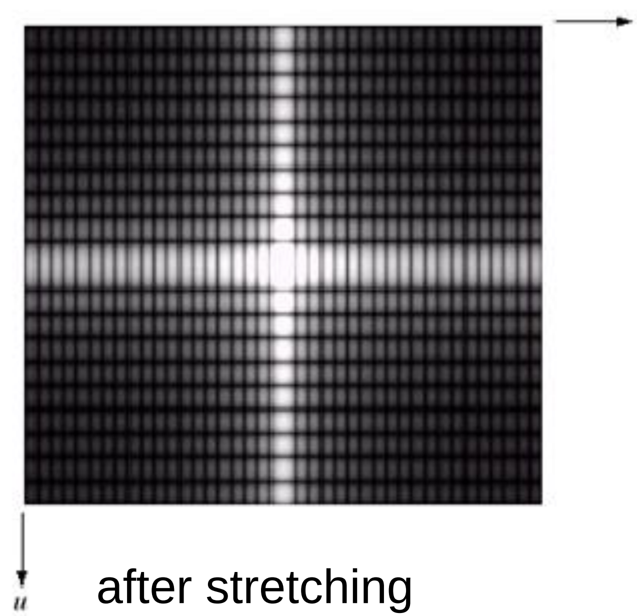
original image

|F(u,v)|



before stretching

|D(u,v)|



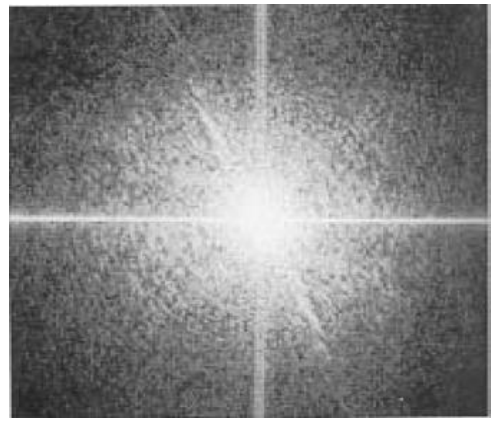
after stretching

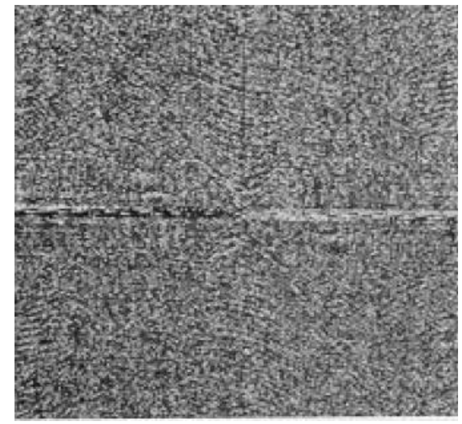
Magnitude and Phase of DFT

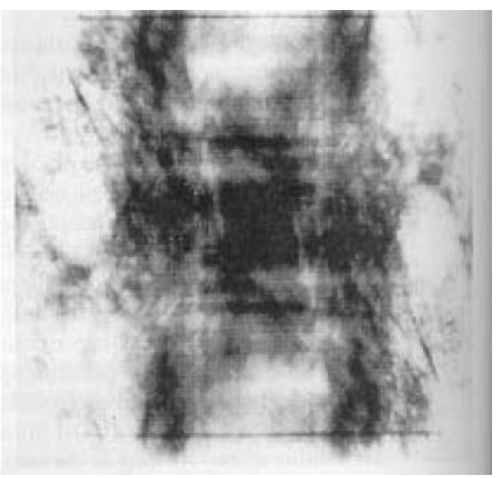
- What is more important?

magnitude

phase







Reconstructed image using magnitude only

Reconstructed image using Phase only

Fast Fourier Transform

- The development of the *Fast Fourier Transform (FFT)* algorithm allows the Fourier transform to be carried out in a reasonable amount of time

- Based on Divide and conquer strategy and exploits property (like Separability) of the 2D transform

- Reduces the amount of time required to perform a Fourier transform by a factor of ***MN log MN*** times

- Discrete, 2-D Fourier & inverse Fast Fourier transforms are implemented in Matlab as `fft2` and `ifft2`

DFT Properties: Multiplication and Convolution

Spatial Domain *(x)*		Frequency Domain *(u)*
$g = f * h$	$\longleftrightarrow$	$G = FH$
$g = fh$	$\longleftrightarrow$	$G = F * H$

So, we can find *g(x)* by Fourier transform

$$\begin{tikzcd} g & = & f \arrow[d, "\text{FT}"] & * & h \arrow[d, "\text{FT}"] \\ G \arrow[u, "\text{IFT}"] & = & F & \times & H \end{tikzcd}$$

Filtering Image in Frequency Domain

To filter an image in the frequency domain:

1. Compute $F(u,v)$ the DFT of the image
2. Multiply $F(u,v)$ by a filter function $H(u,v)$
3. Compute the inverse DFT of the result

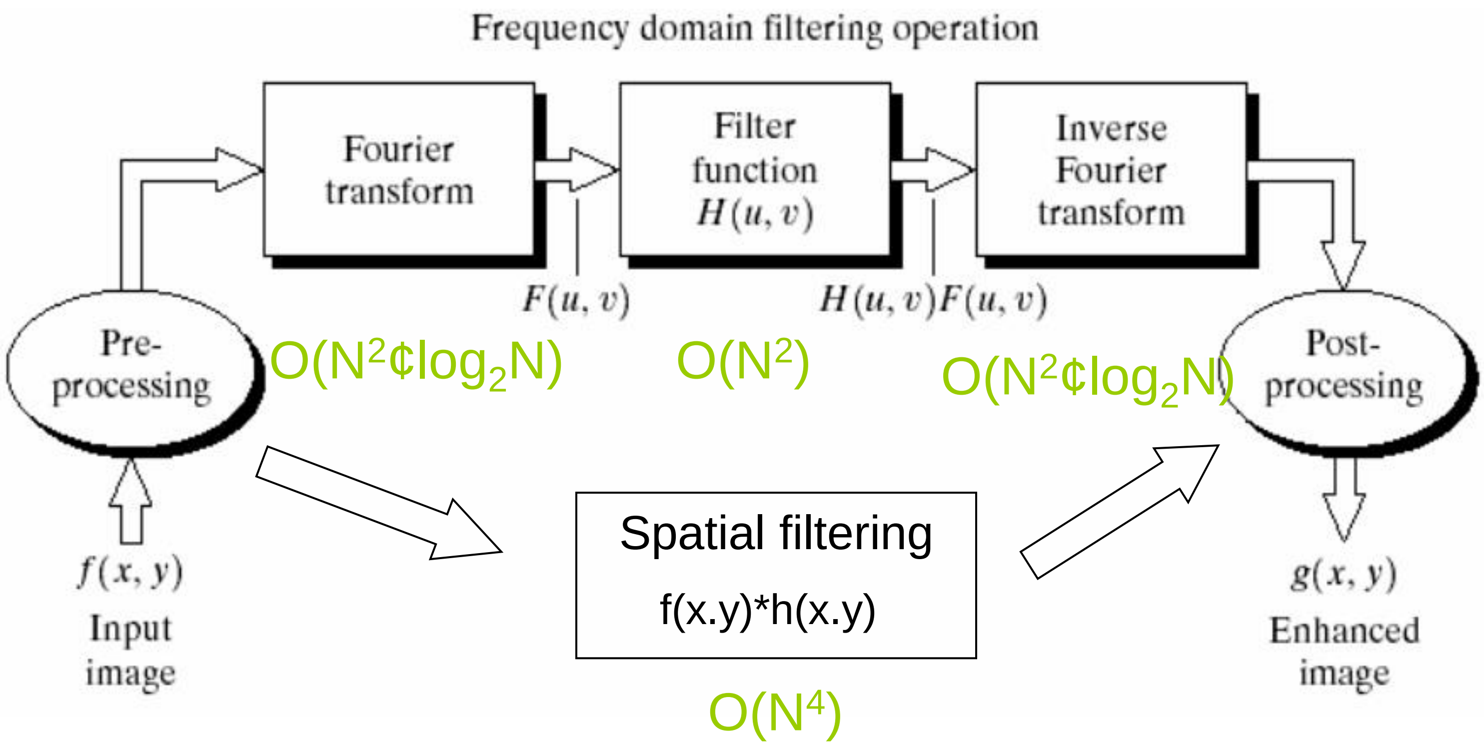


IMAGE ENHANCEMENT IN FREQUENCY DOMAIN

Image Smoothing

Image Sharpening

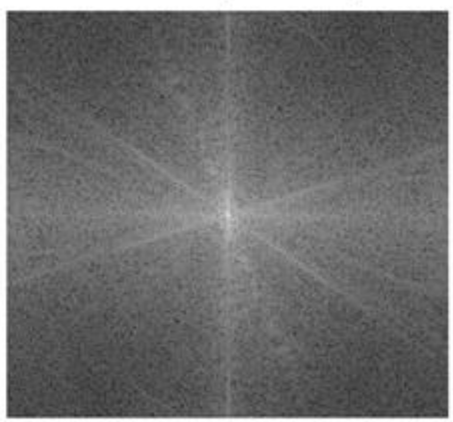
Smoothing: Averaging / Lowpass Filtering

Smoothing/Blurring results from:

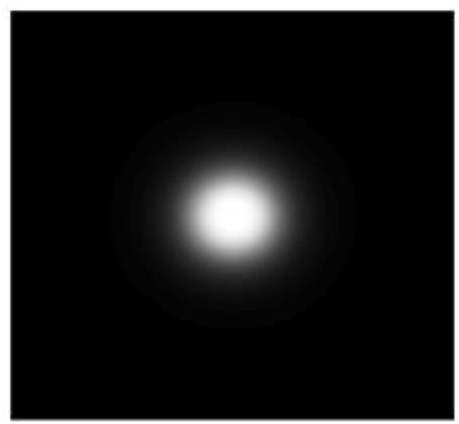
- Pixel averaging in the spatial domain:
  - Each pixel in the output is a weighted average of its neighbors.
  - Is a convolution whose weight matrix sums to 1.
- Lowpass filtering in the frequency domain:
  - High frequencies are diminished or eliminated

Original image

FFT of original image



Low-pass filter

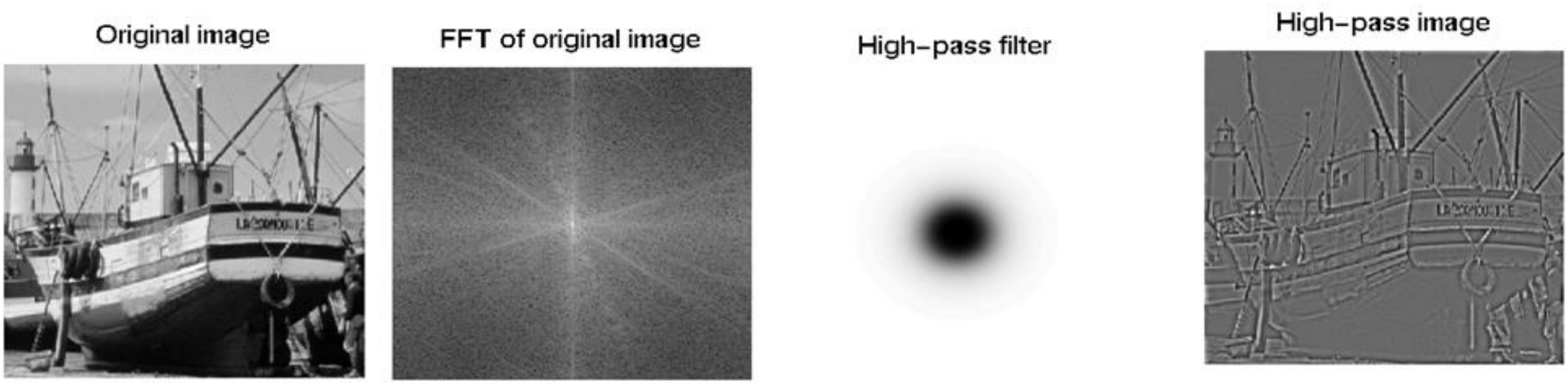


Low-pass image

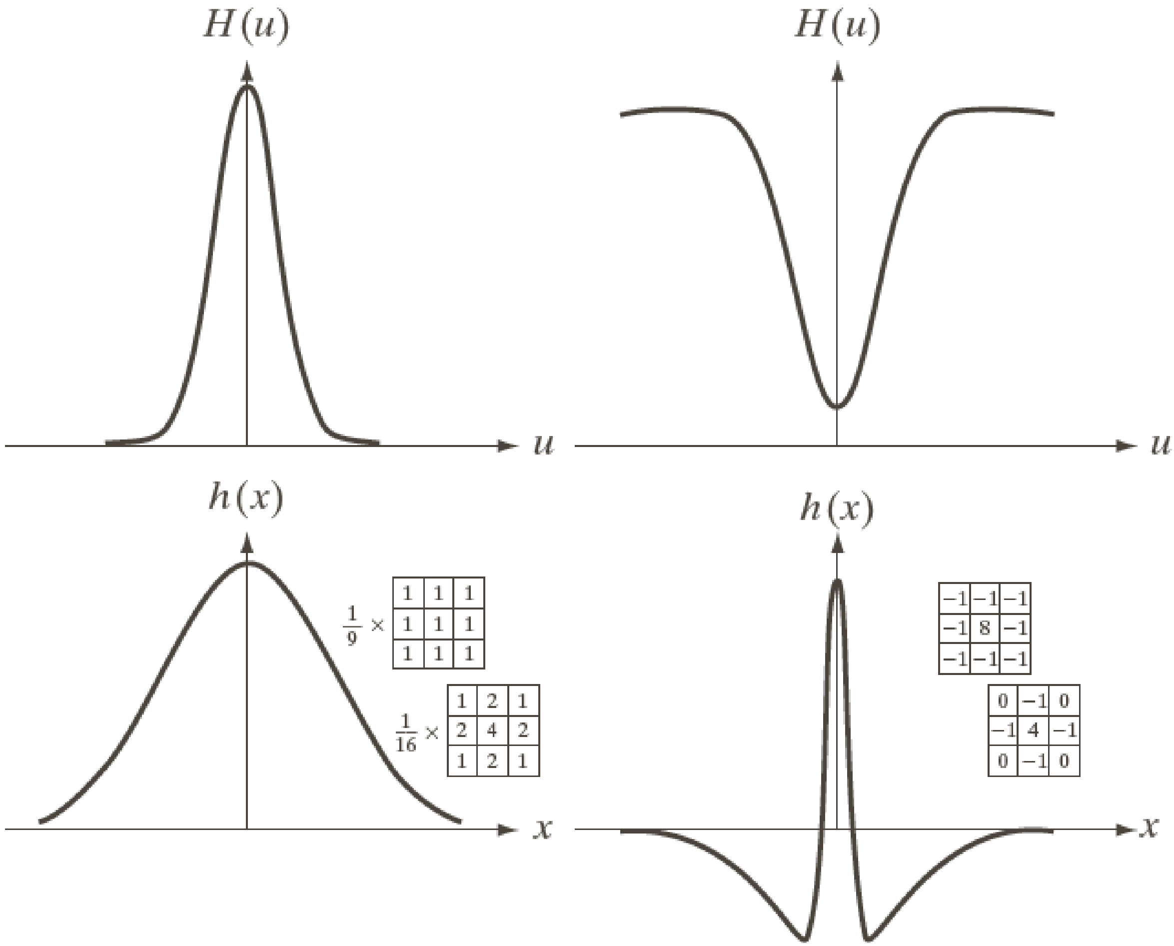
Sharpening: Differencing / Highpass Filtering

Sharpening results from adding to the image, a copy of itself that has been:

- Pixel-differenced in the spatial domain:
  - Each pixel in the output is a difference between itself and a weighted average of its neighbors.
  - Is a convolution whose weight matrix sums to 0.
- Highpass filtered in the frequency domain:
  - High frequencies are enhanced or amplified.



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

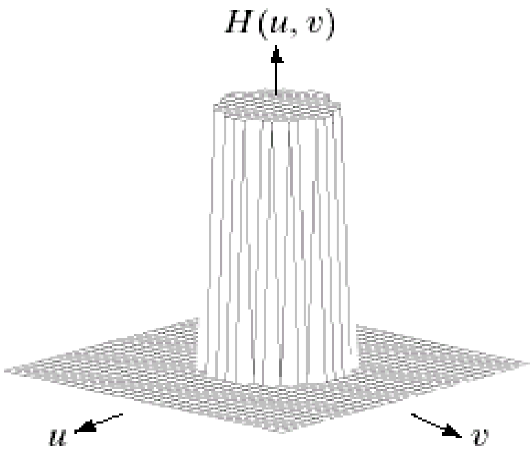


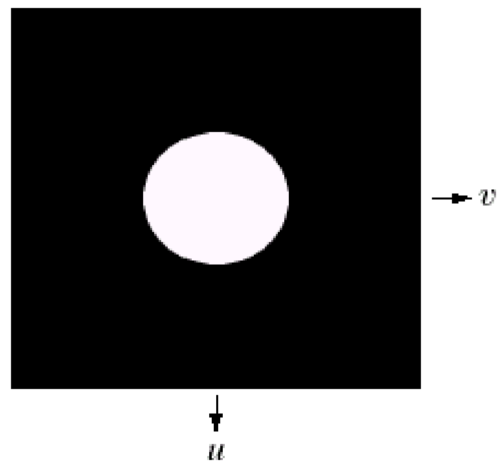
a c
b d

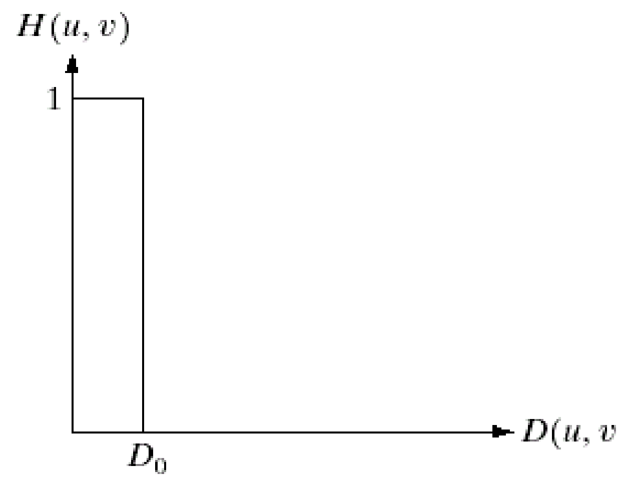
FIGURE 4.37 (a) A 1-D Gaussian lowpass filter in the frequency domain. (b) Spatial lowpass filter corresponding to (a). (c) Gaussian highpass filter in the frequency domain. (d) Spatial highpass filter corresponding to (c). The small 2-D masks shown are spatial filters we used in Chapter 3.

Ideal Low Pass Filter

Simply cut off all high frequency components that are a specified distance D_0 from the origin of the transform







changing the distance changes the behaviour of the filter

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Ideal Low Pass Filter (cont…)

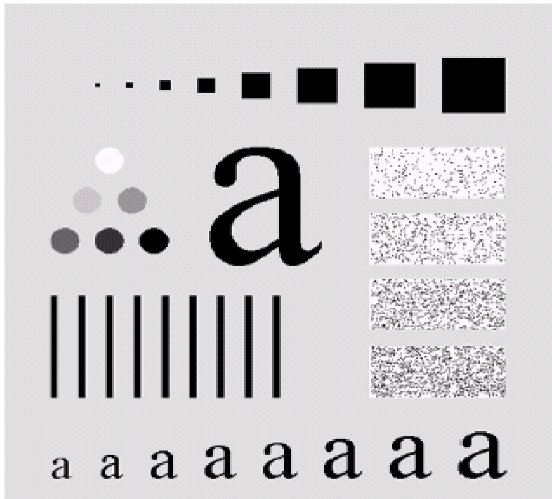
The transfer function for the ideal low pass filter can be given as:

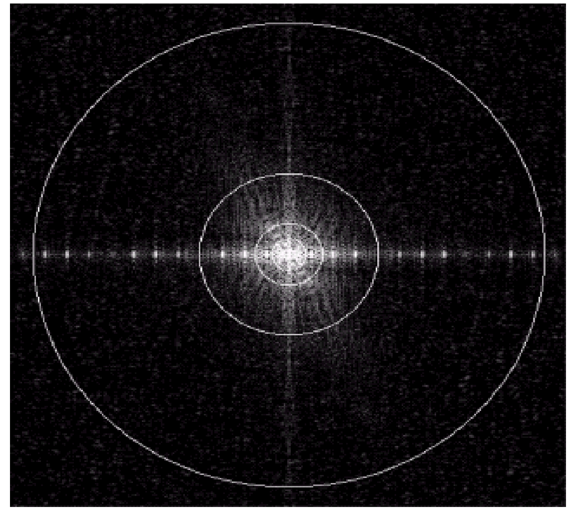
$$H(u,v) = \begin{cases} 1 \text{ if } D(u,v) \le D_0 \\ 0 \text{ if } D(u,v) > D_0 \end{cases}$$

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

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

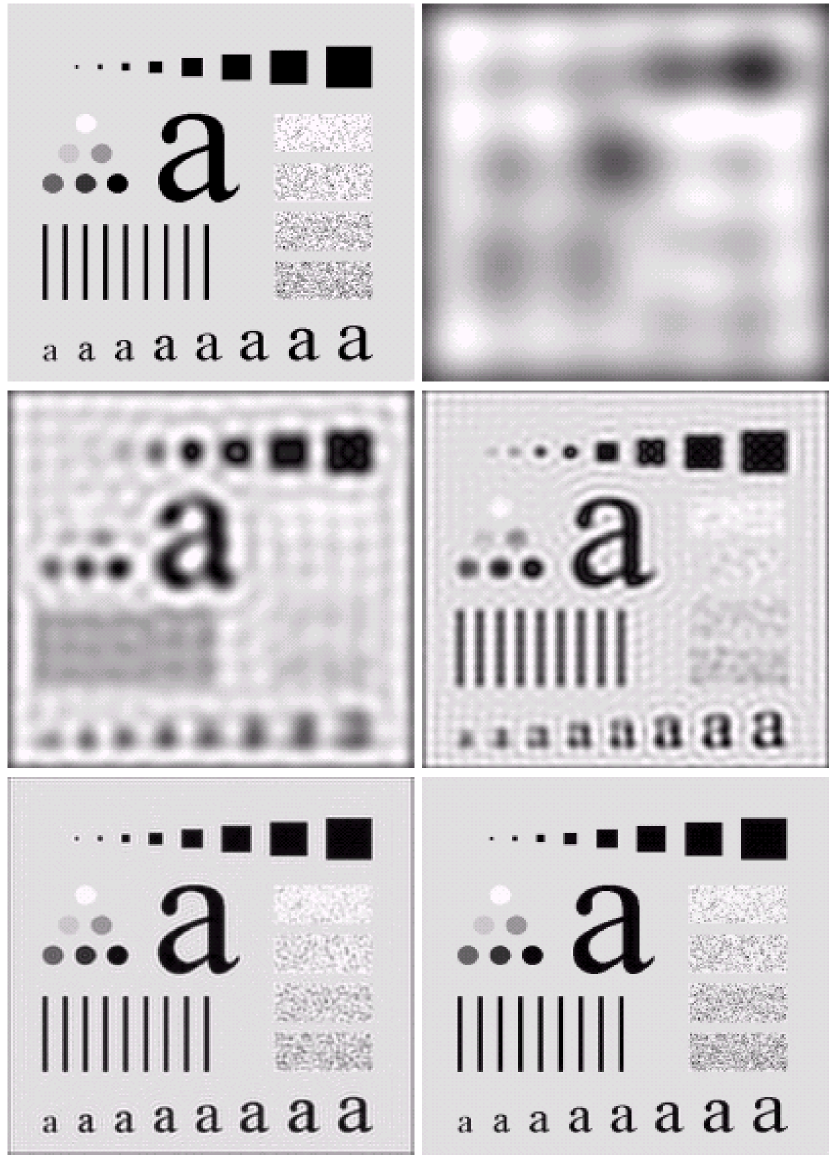
Ideal Low Pass Filter (cont…)





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

Ideal Low Pass Filter (cont…)



Original image

Result of filtering with ideal low pass filter of radius 5

Result of filtering with ideal low pass filter of radius 15

Result of filtering with ideal low pass filter of radius 30

Result of filtering with ideal low pass filter of radius 80

Result of filtering with ideal low pass filter of radius 230

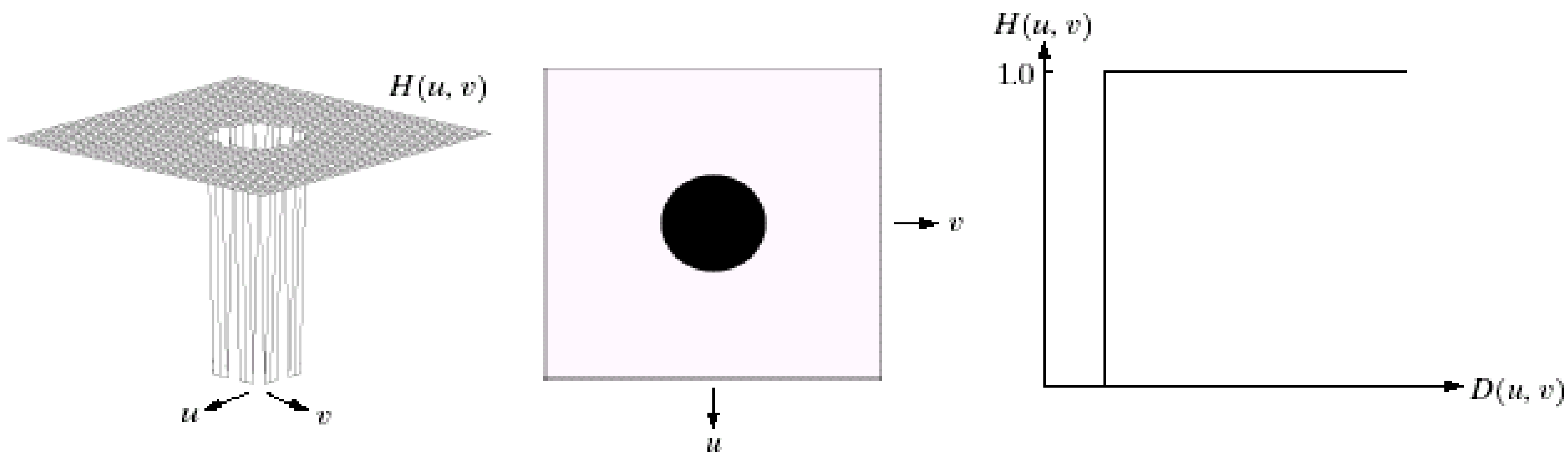
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Ideal High Pass Filters

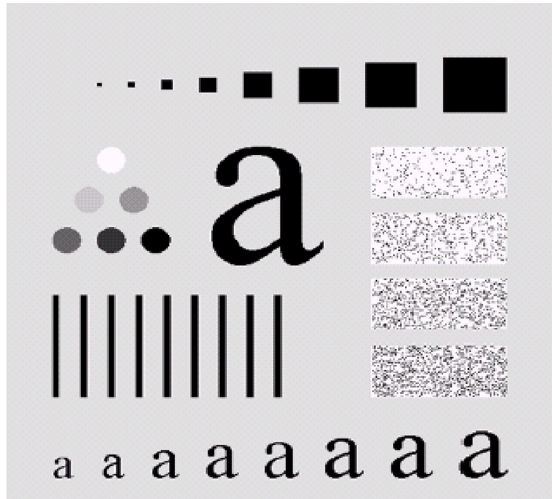
The ideal high pass filter is given as:

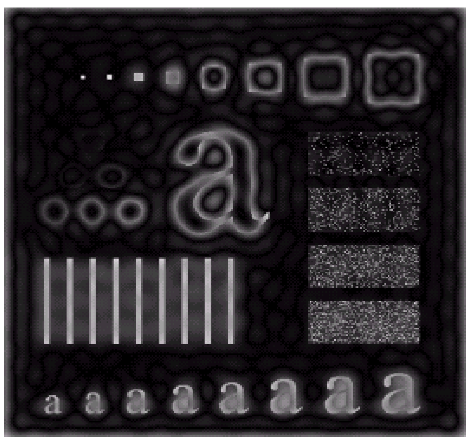
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

where D_0 is the cut off distance as before

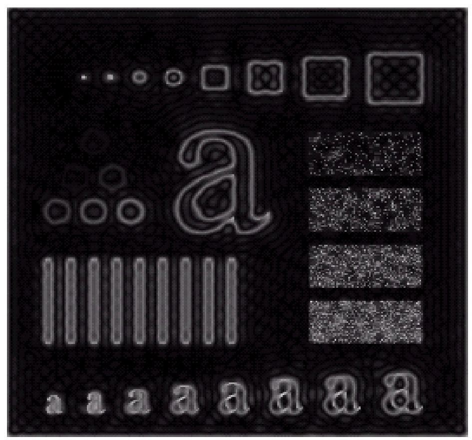


Ideal High Pass Filters (cont…)

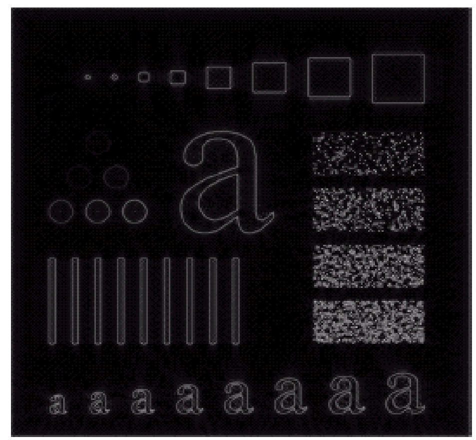




Results of ideal high pass filtering with $D_0 = 15$



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

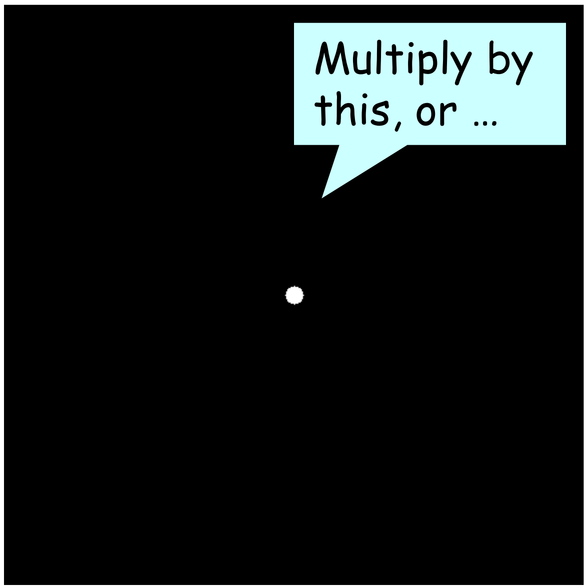


Results of ideal high pass filtering with $D_0 = 80$

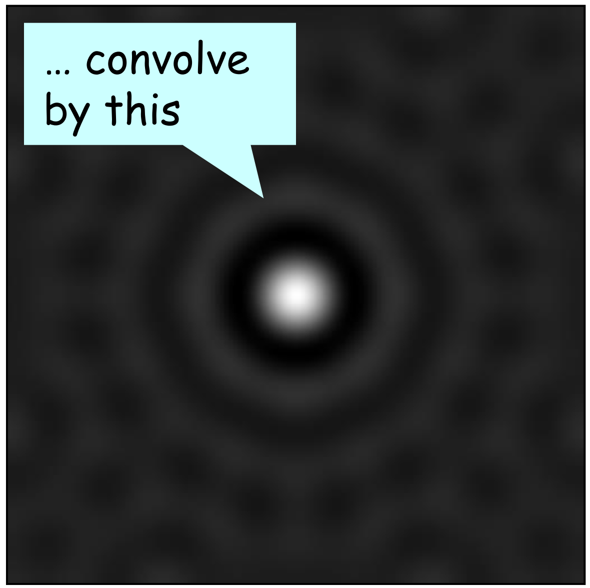
The Ringing problem

Ideal Filters Do Not Produce Ideal Results

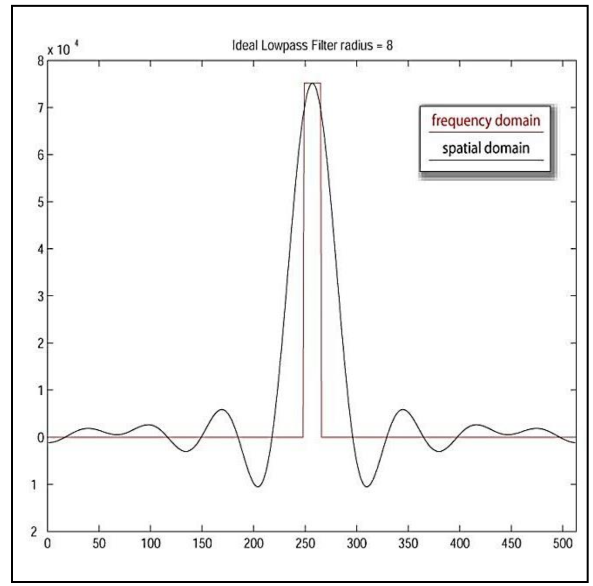
Image size: 512x512
FD filter radius: 8



Fourier Domain Rep.



Spatial Representation

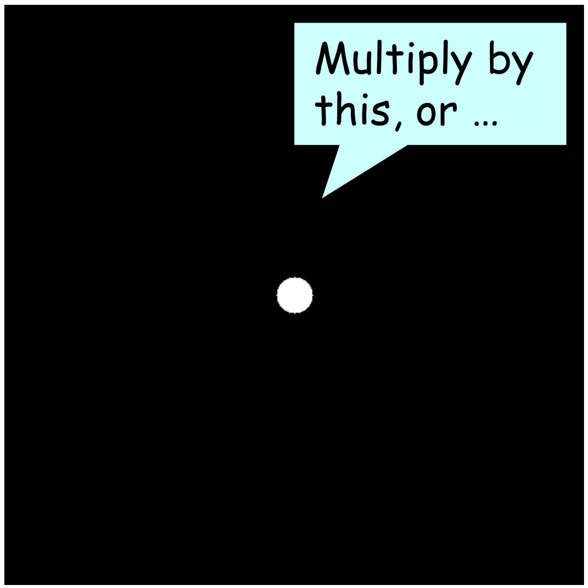


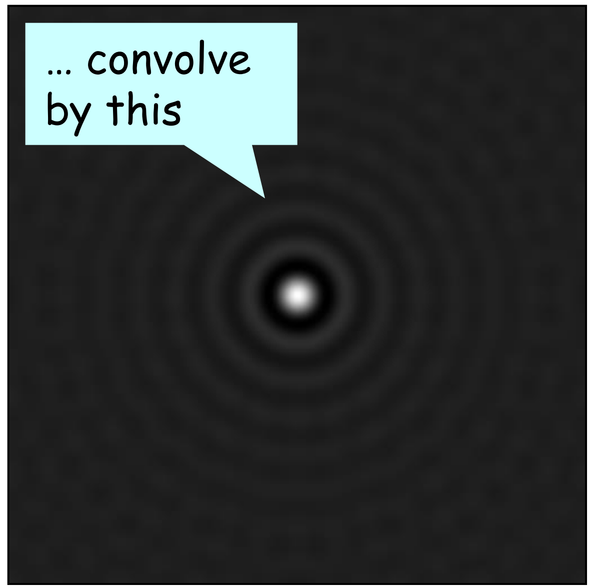
Central Profile

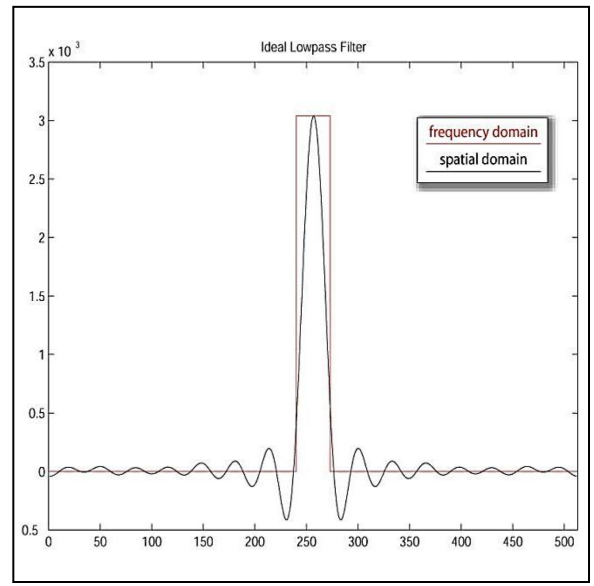
The Ringing problem

Ideal Filters Do Not Produce Ideal Results

Image size: 512x512
FD filter radius: 16







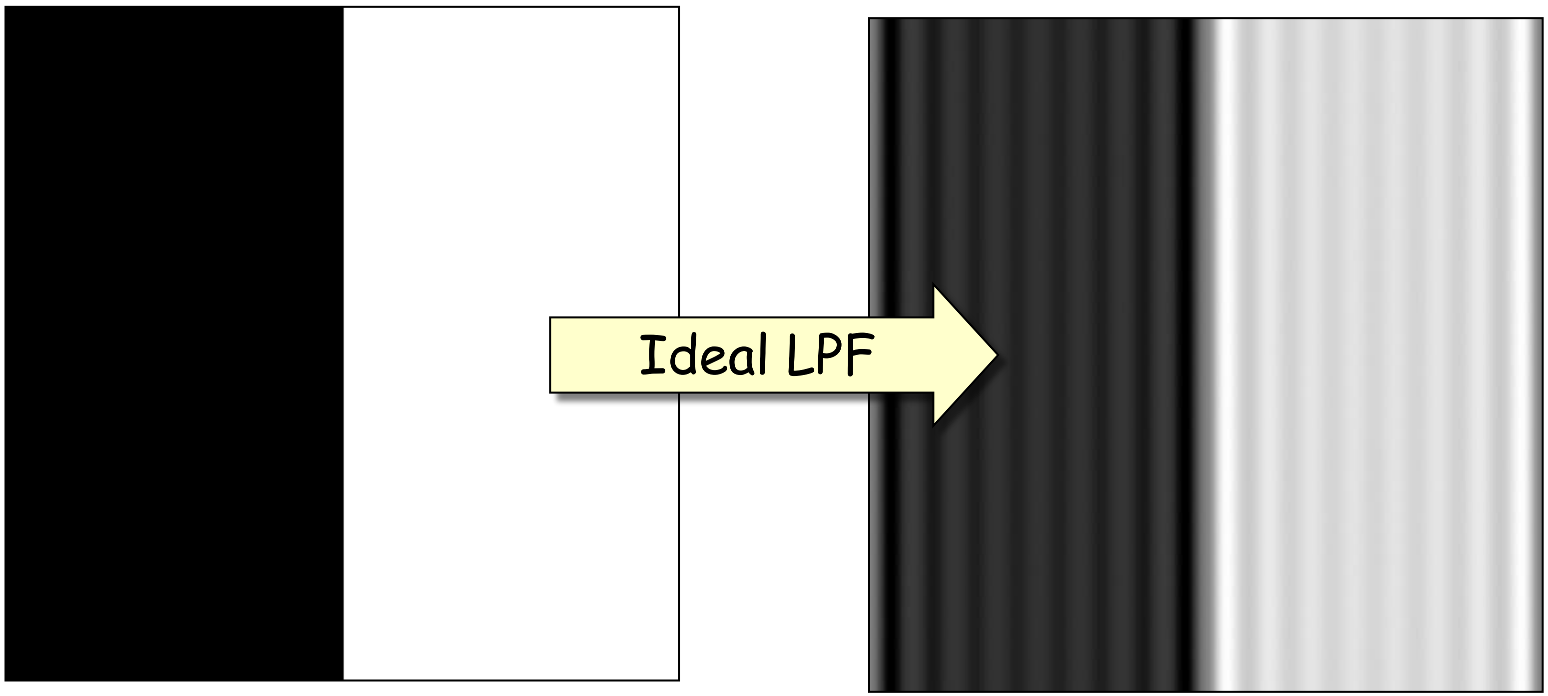
Fourier Domain Rep.

Spatial Representation

Central Profile

$\uparrow D_0 \longrightarrow \downarrow$ Ringing radius + $\downarrow$ blur

The Ringing problem

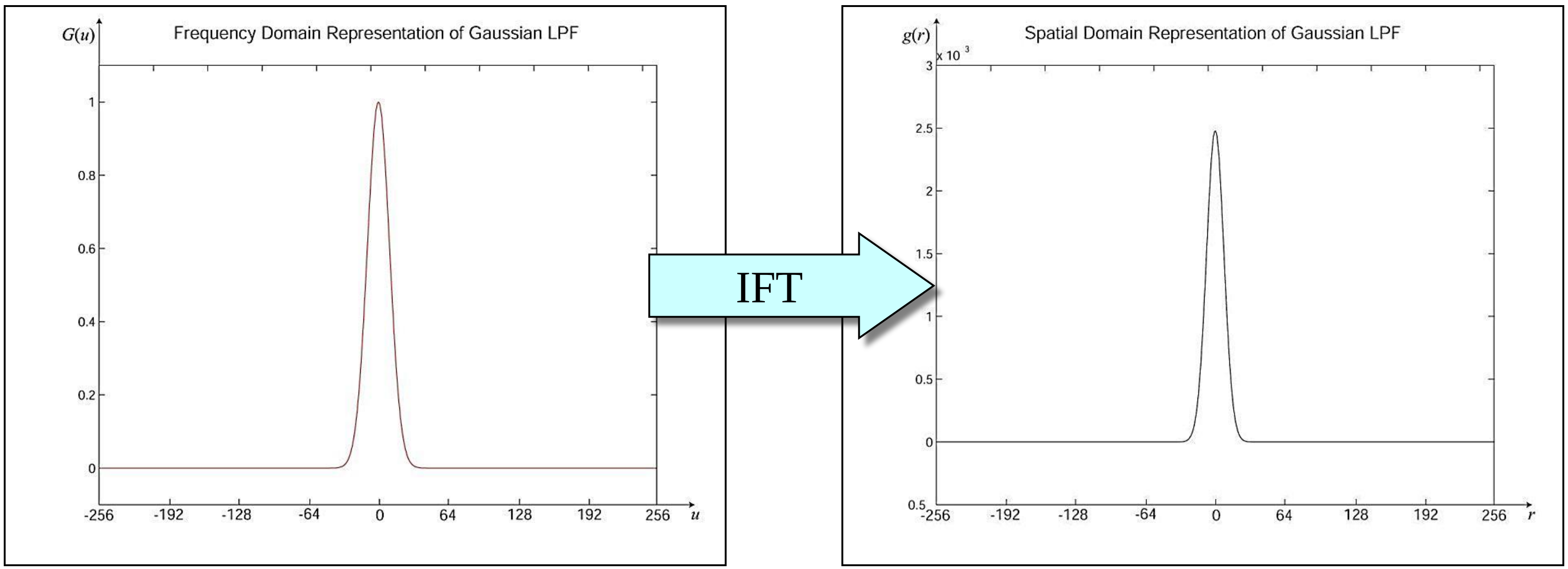


Blurring the image above w/ an ideal lowpass filter…

…distorts the results with ringing or ghosting.

Optimal Filter

The Gaussian

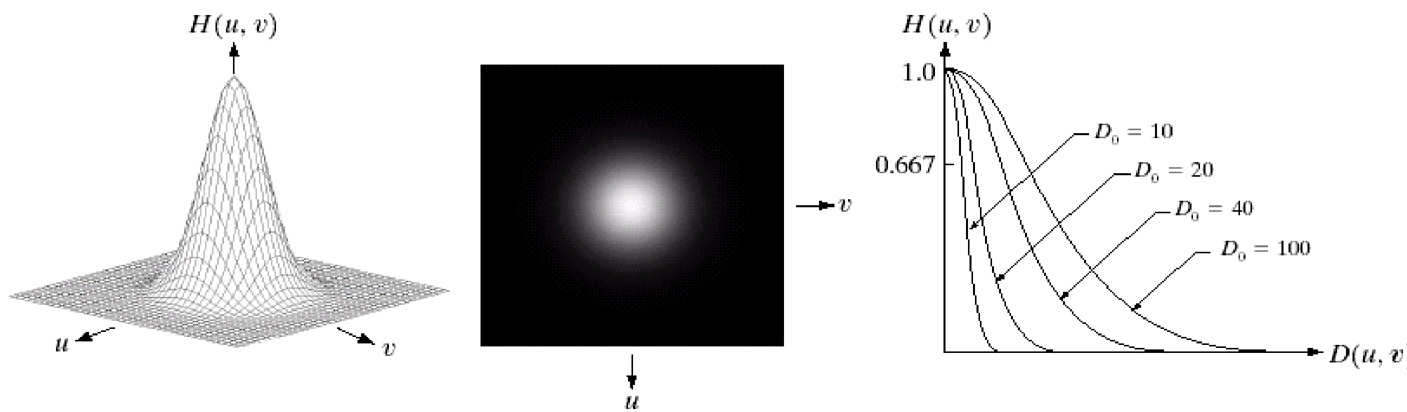


The Gaussian filter optimizes it by providing the sharpest cutoff possible without ringing.

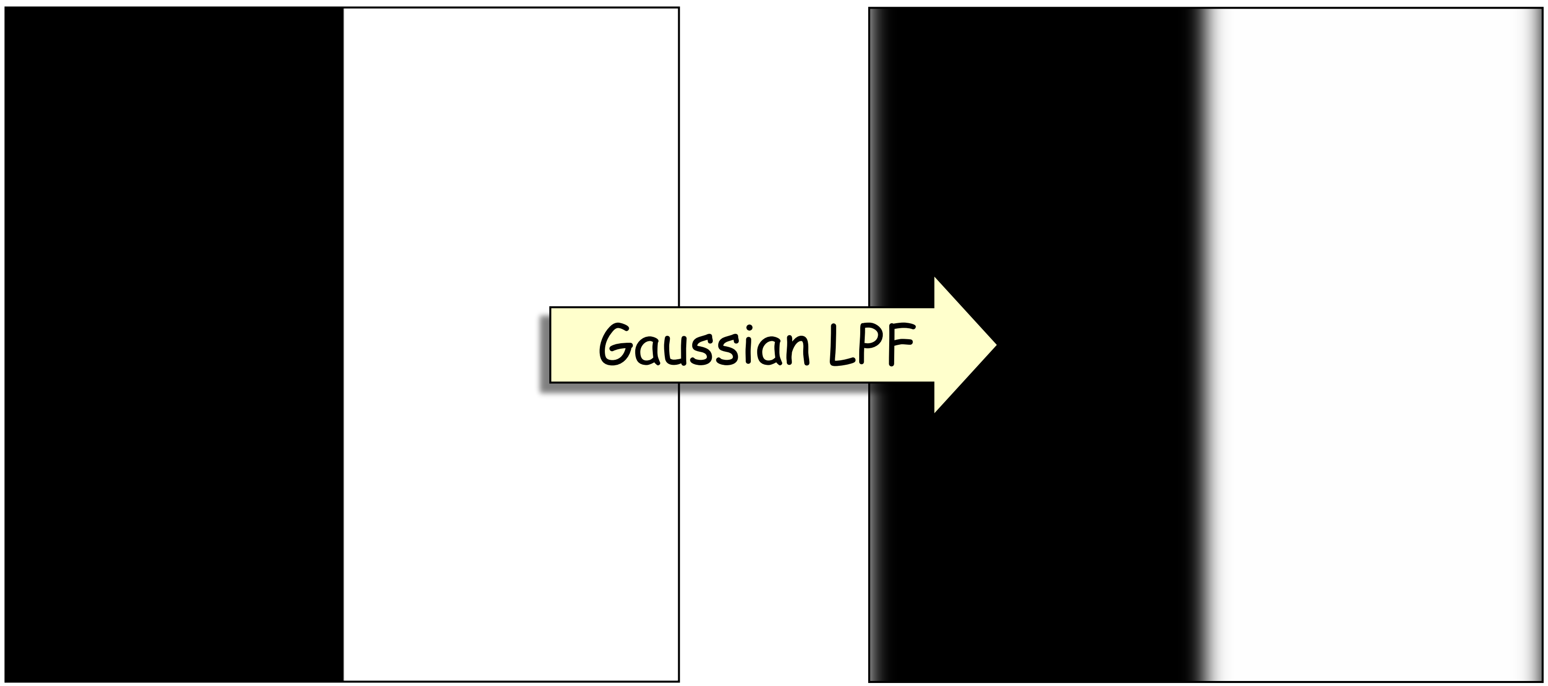
Gaussian Lowpass Filters

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

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



Optimal Filter: The Gaussian

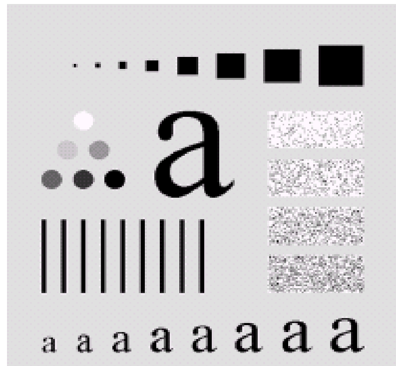


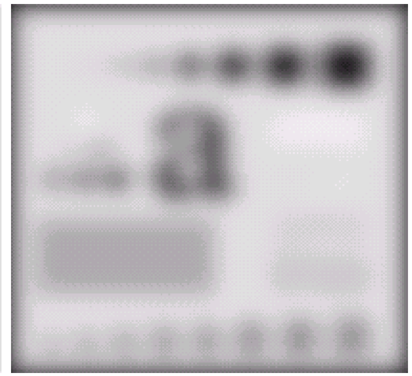
With a gaussian lowpass filter, the image above …

… is blurred without ringing or ghosting.

Gaussian Lowpass Filters (cont…)

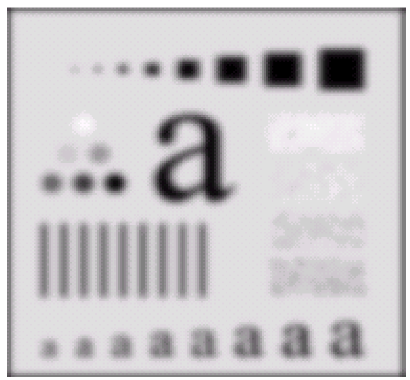
Original image

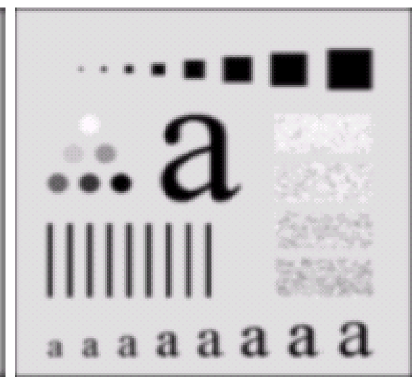




Result of filtering with Gaussian filter with cutoff radius 5

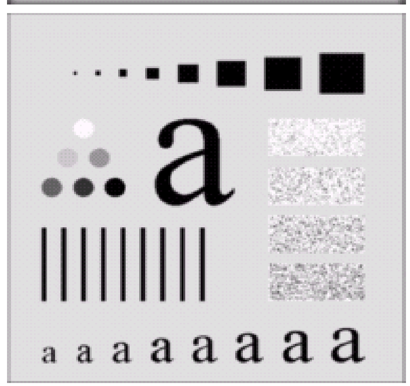
Result of filtering with Gaussian filter with cutoff radius 15

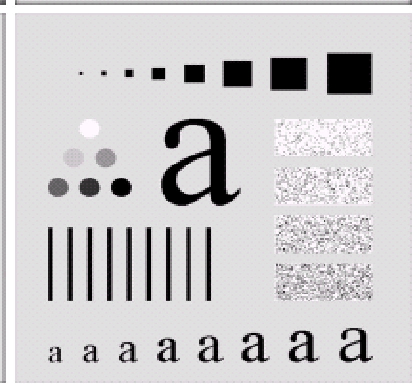




Result of filtering with Gaussian filter with cutoff radius 30

Result of filtering with Gaussian filter with cutoff radius 85





Result of filtering with Gaussian filter with cutoff radius 230

Resolution Sequence

Original Image

Resolution Sequence

Gaussian
LPF

$\sigma_1 = 64$

Resolution Sequence

Gaussian
LPF
$\sigma_2 = 32$

Resolution Sequence

Gaussian
LPF
$\sigma_3 = 16$

Resolution Sequence

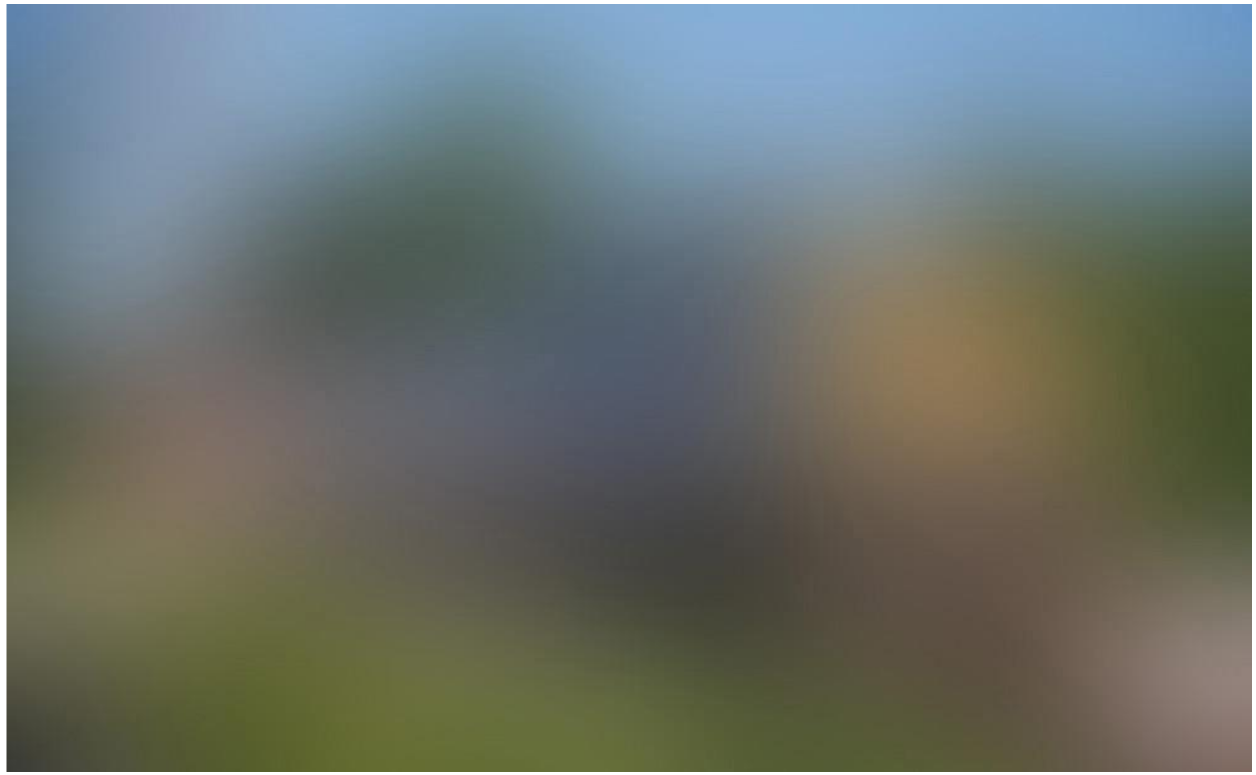
Gaussian LPF

$\sigma_4 = 8$

Resolution Sequence

Gaussian
LPF
$\sigma_5 = 4$

Resolution Sequence



Gaussian LPF

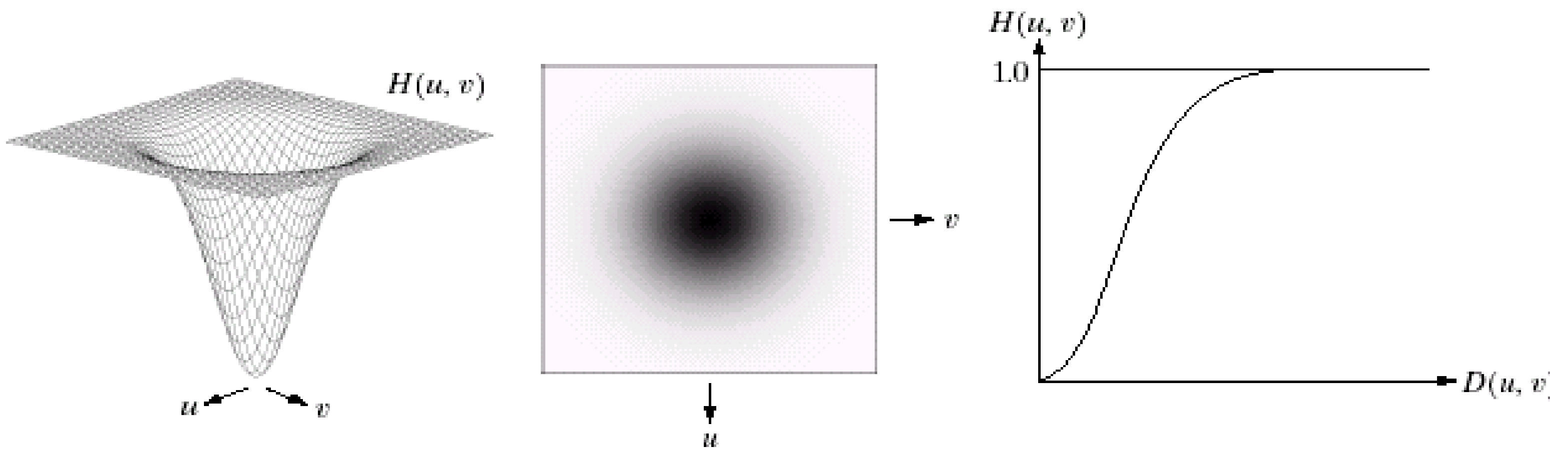
$\sigma_6 = 2$

Gaussian High Pass Filters

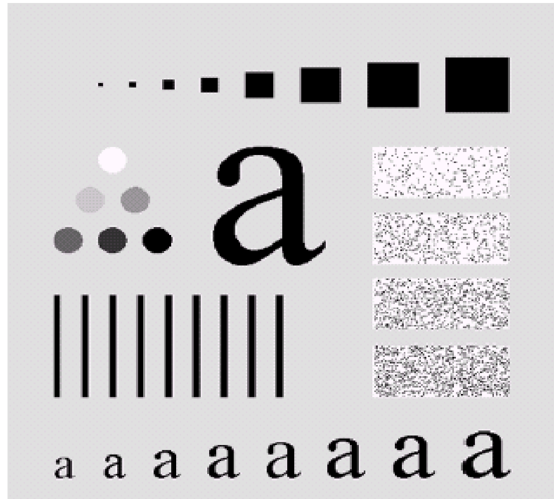
The Gaussian high pass filter is given as:

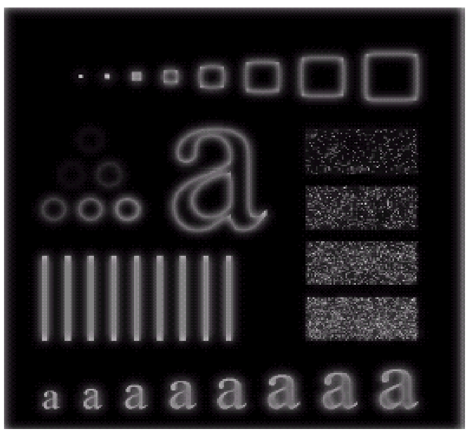
$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

where D_0 is the cut off distance as before

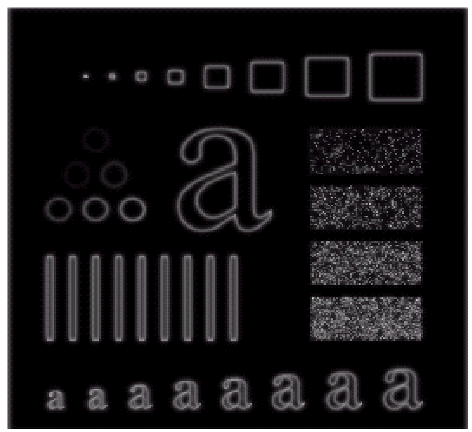


Gaussian High Pass Filters (cont…)





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



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

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

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_6 = 2$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_5 = 4$.

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_4 = 8$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_4\right)\right].$$

Highpass Sequence

Difference between original image and Gaussian LPF image at $\sigma_3 = 32$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_3\right) \right].$$

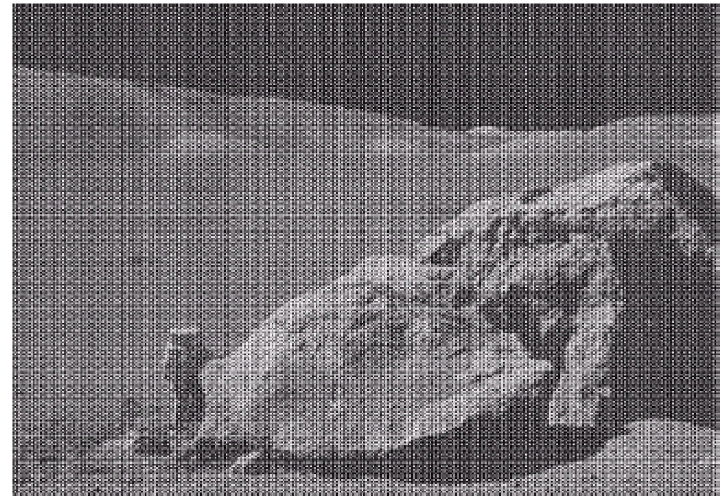
Highpass Sequence

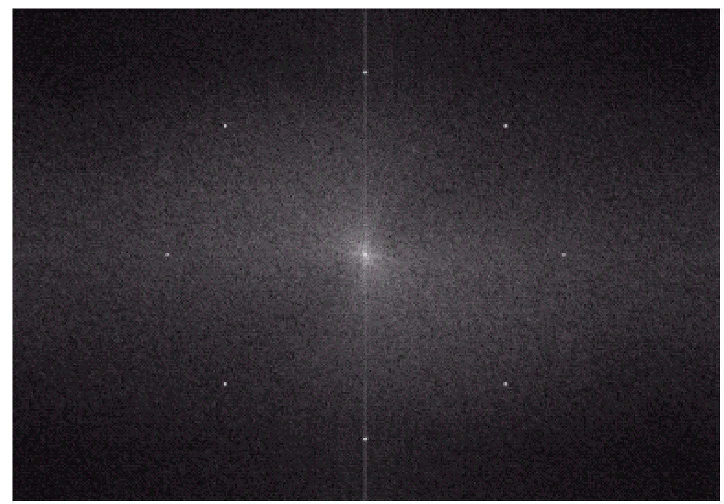
Difference between original image and Gaussian LPF image at $\sigma_2 = 64$.

$$\mathbf{J} = \mathbf{I} - \left[\mathbf{I} * g\left(\sigma_2\right)\right].$$

Periodic Noise removal

- Typically arises due to electrical or electromagnetic interference
- Gives rise to regular noise patterns in an image
- Frequency domain techniques in the Fourier domain are most effective at removing periodic noise





Band Reject Filters

Removing periodic noise from an image involves removing a particular range of frequencies from that image

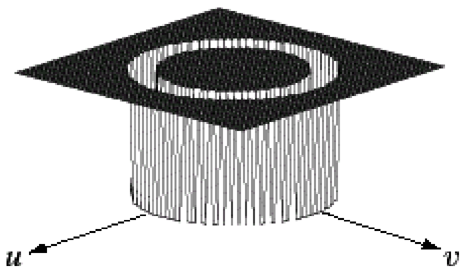
Band reject filters can be used for this purpose

An ideal band reject filter is given as follows:

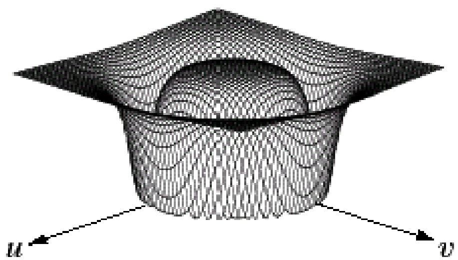
$$H(u,v) = \begin{cases} 1 & if \ D(u,v) < D_0 - \dfrac{W}{2} \\ 0 & if \ D_0 - \dfrac{W}{2} \le D(u,v) \le D_0 + \dfrac{W}{2} \\ 1 & if \ D(u,v) > D_0 + \dfrac{W}{2} \end{cases}$$

Band Reject Filters (cont…)

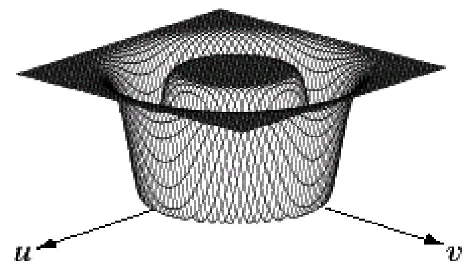
The ideal band reject filter is shown below, along with Butterworth and Gaussian versions of the filter



Ideal Band Reject Filter



Butterworth Band Reject Filter (of order 1)

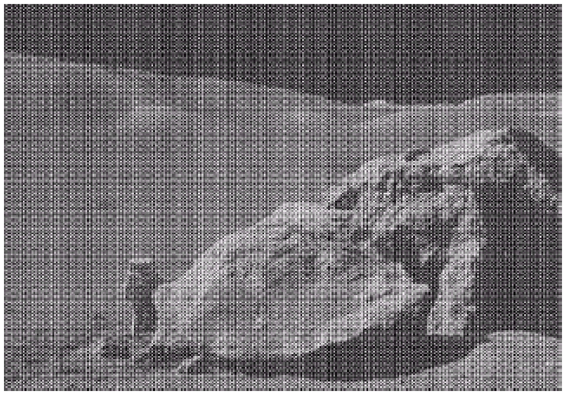


Gaussian Band Reject Filter

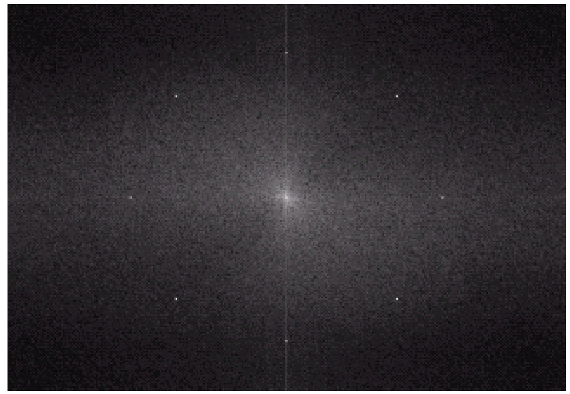
Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

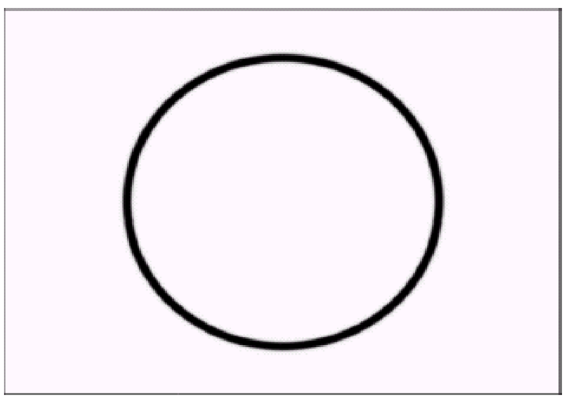
Band Reject Filter Example

Image corrupted by sinusoidal noise



Fourier spectrum of corrupted image





Butterworth band reject filter

Filtered image