The University of Western Australia

CITS 4402 Computer Vision

Ajmal Mian

Lecture 03 – Discrete Fourier Transform

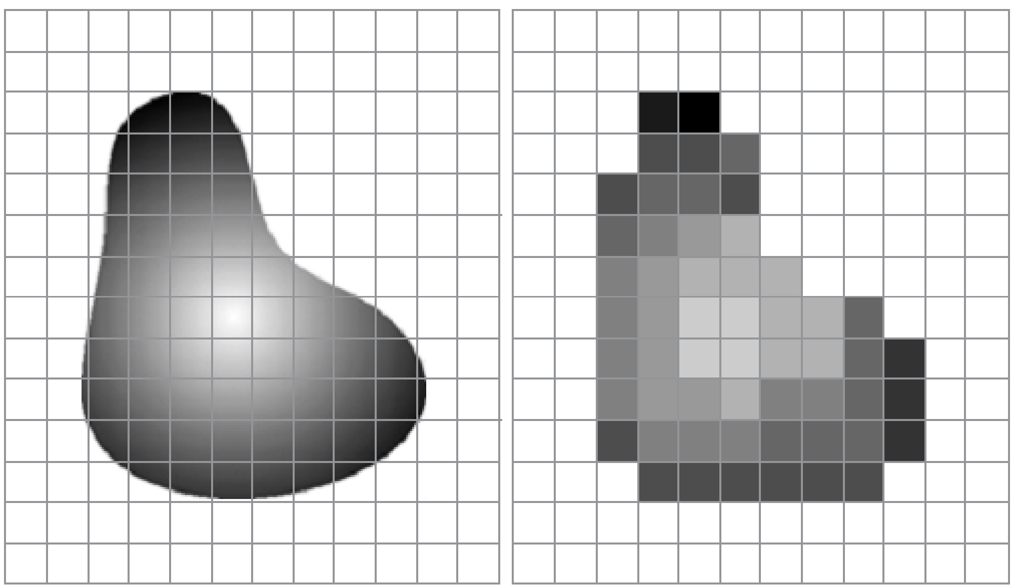
Objectives of this lecture

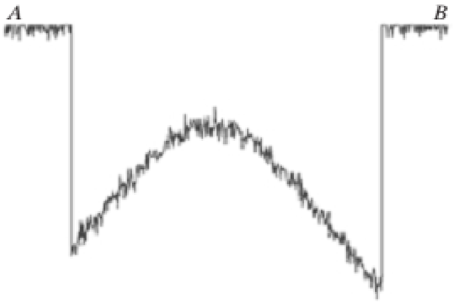
- To learn about linear and non-linear filters
- To learn about convolution
- To learn Fourier Transform and Discrete Fourier Transform (DFT)
- To learn the relationship between frequency and spatial domains
- Frequency domain filtering and Lab week03 (hybrid images)

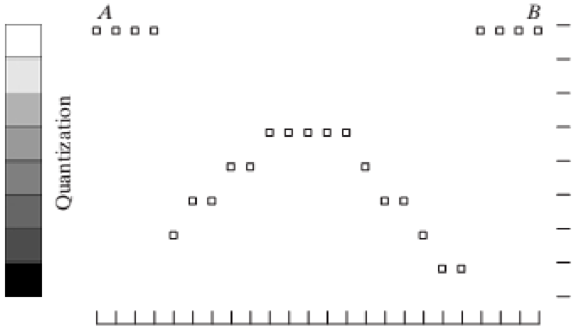
Images as 2-Dimensional Signals

- To understand the processing of greyscale images, it is important to have an understanding of the Fourier Transform
- A greyscale image can be treated as a 2-D *signal.*
- A **signal** is any physical phenomenon that can be modelled as a function of time or position to some real- or vector-valued domain, and is used to carry information.
- A signal is said to be **analog**, when the domain and range are continuous; or **digital**, when both are discrete.

Analog and Digital Signals







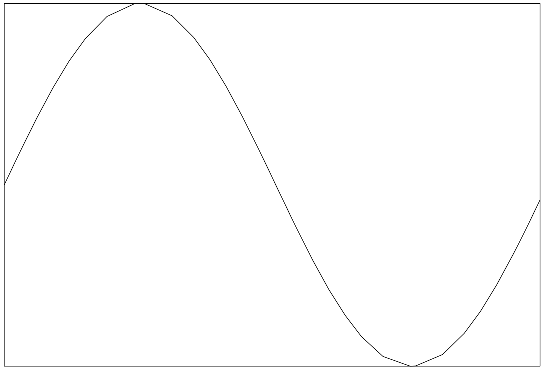
1-D and 2-D Signals

- For simplicity, we will study 1-dimensional signals.

- For 1D, the axis will usually be time, the intuitive way to think of the signal $s(t)$ is an audio signal. For discrete signals, we also write s_j.

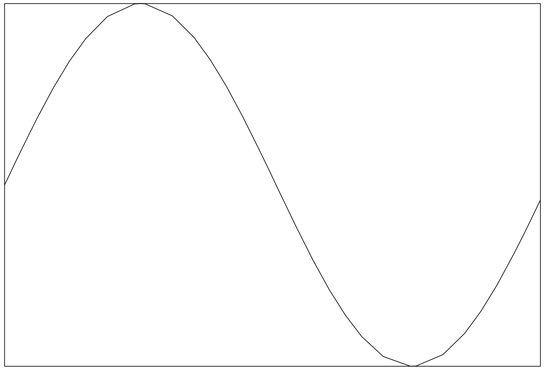
- For 2D, the axes will usually be 2D-*space*, and we will call the signal $s(x, y)$ an *image*. For discrete signals, we also write $s_{i,j}$ or $s[x, y]$.

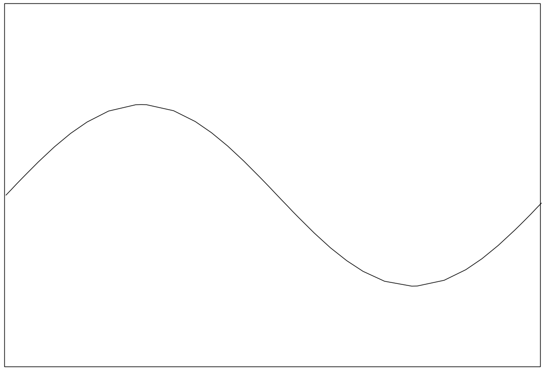
- Despite the different notation, **the principles are the same!**

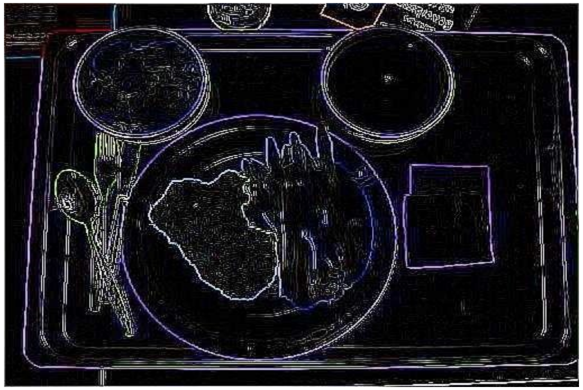


Filters and filtering

- **Filtering** is any operation that transforms one signal into another signal







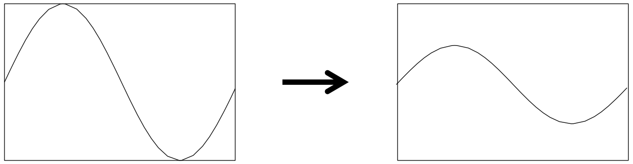
Linear and Non-linear Filters

- A filter T is called **linear** if it acts on the signal linearly, i.e. for all signals s_1 and s_2 and constants α and β the following holds:

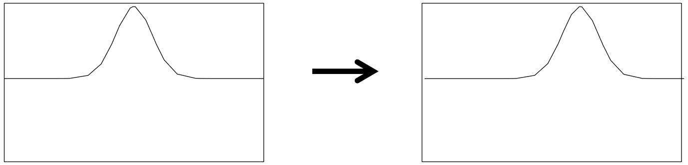
$$T(\alpha s_1 + \beta s_2) = \alpha T(s_1) + \beta T(s_2)$$

Otherwise, the filter is called **non-linear**.

- Scaling of amplitude (volume): T: $s(t) \rightarrow c \cdot s(t)$ is linear



- Shift in time (phase shift): T: $s(t) \rightarrow s(t + c)$ is linear



Linear and Non-linear Image Filters

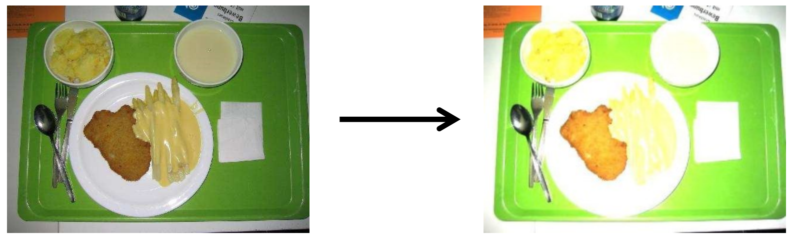
- Scaling of intensity $s(x, y) \rightarrow c \cdot s(x, y)$ is **?**

- Geometric operations (translation, rotation, mirroring) are **?**

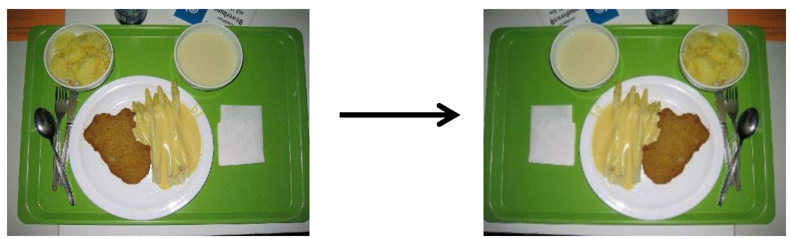
- Gamma correction $s(x, y) \rightarrow c \cdot s(x, y)^{\gamma}$ is **?**

Linear and Non-linear Image Filters

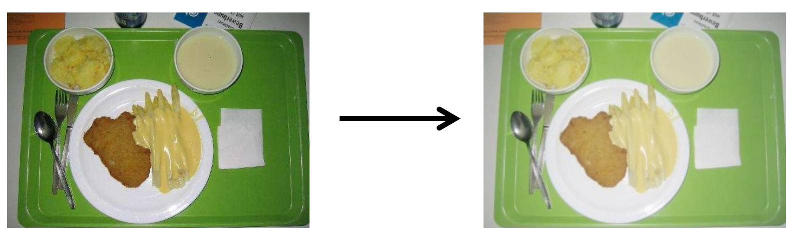
- Scaling of intensity $s(x, y) \rightarrow c \cdot s(x, y)$ is linear



- Geometric operations (translation, rotation, mirroring) are linear



- Gamma correction $s(x, y) \rightarrow c \cdot s(x, y)^{\gamma}$ is non-linear

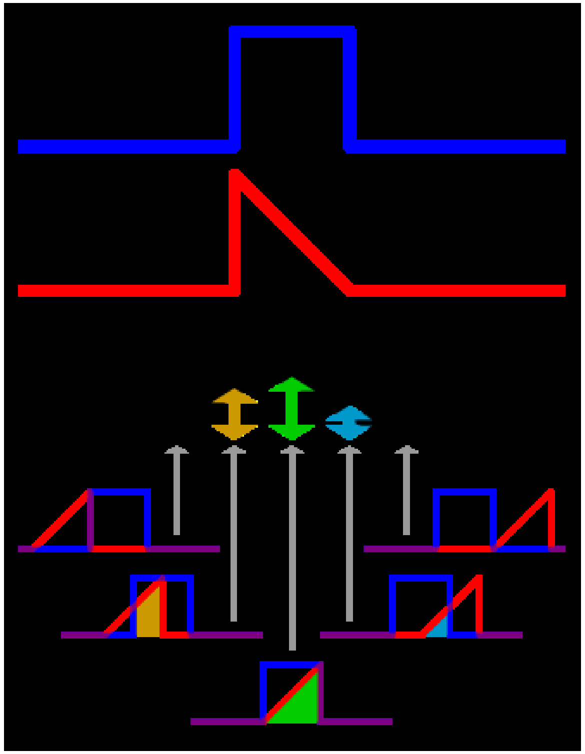


Convolution

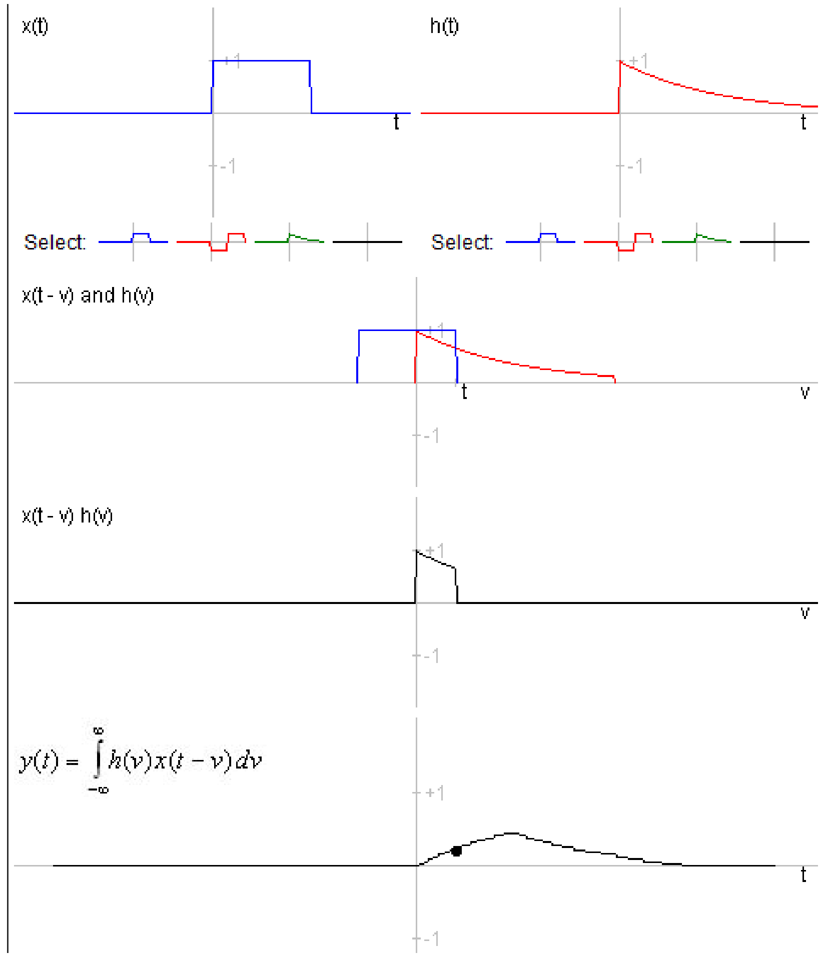
- Convolution ($f * g$) of two signals f and g is defined as the integral/sum of the product of the two functions after one is reversed and shifted

$$(f * g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f[m]\, g[n-m]$$

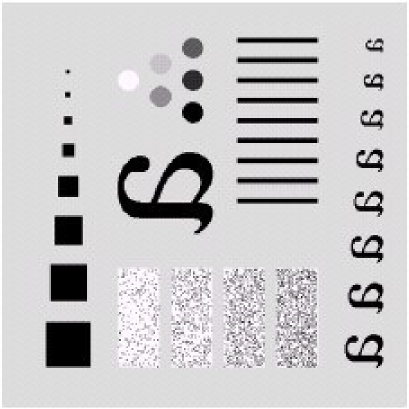
- Convolution gives the area overlap between the two functions as a function of the amount that one of the original functions is translated after reversal



http://www.jhu.edu/~signals/convolve/index.html



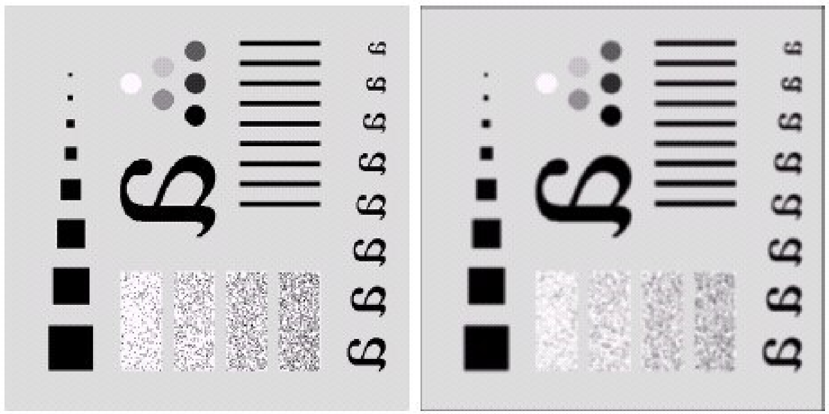
Example of 2-D Convolution



?

Convolution with a box filter of size 5 ?

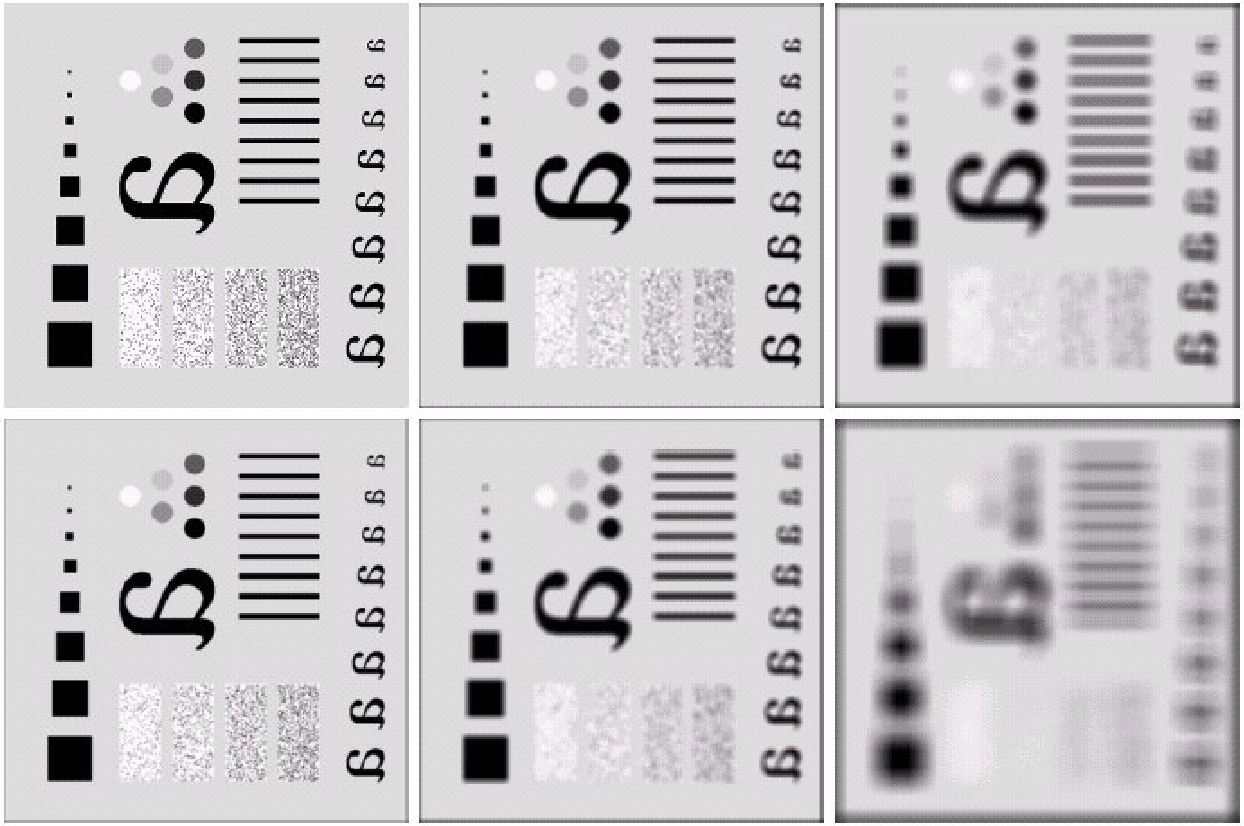
Example of 2-D Convolution



?

Convolution with a box filter of size 15 ?

Example of 2-D Convolution



Convolution with box filters of size 1,5,15,3,9,35 (reading order)

Detecting Signals

- Often, we have two signals: (1) "real-life" that we are measuring, (2) ideal signal we are looking for.
  - How can we check if a signal $s(t)$ contains another signal $\sigma(t)$? And if it does, how much of it?
- First, we have to give a meaning to "how much"

- Let $s(t)$ be a signal on D, then the energy of s is

$$E := \int_D |s(t)|^2 dt$$

- Physically, s is a wave function and E is its energy.
- Mathematically, E is the squared norm of $s \in L^2(D)$.
- Example: a sine wave $\sin(t)$ on $[0, 2\pi]$ has the energy $E = \int_0^{2\pi} \sin^2(t)dt = \pi$

Correlating Signals

- Take a signal σ with energy $E = 1$.
- To calculate how much of σ there is within s, we treat both as random processes and calculate their correlation coefficient:

$$\alpha = \int \sigma(\tau) s(\tau) d\tau$$

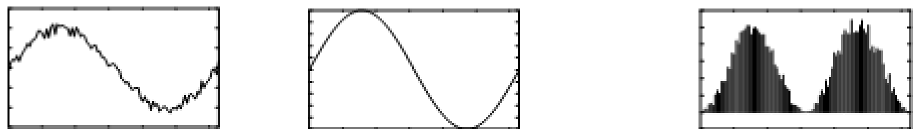
- If s is exactly σ, then $\alpha = E(\sigma) = 1$.
- For signals σ of arbitrary energy, α is often normalized by the energy of σ:

$$\alpha = \frac{1}{E(\sigma)} \int \sigma(\tau) s(\tau) d\tau$$

- That way, $s = \sigma$ again implies $\alpha = 1$.

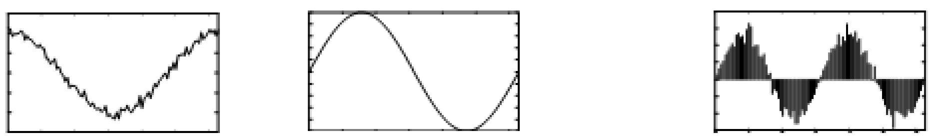
Examples of Signal Correlation

- Let s be a slightly disturbed version of σ:



s σ α

- The contributions to the integral are large. α will be large.
- Warning: s and σ have to be *synchronized*!
- If there is a phase shift, the signal may be missed:



s σ α

- s is the same disturbed version of σ, but shifted in time.
- Positive and negative contributions in the integration cancel each other and α might end up small.

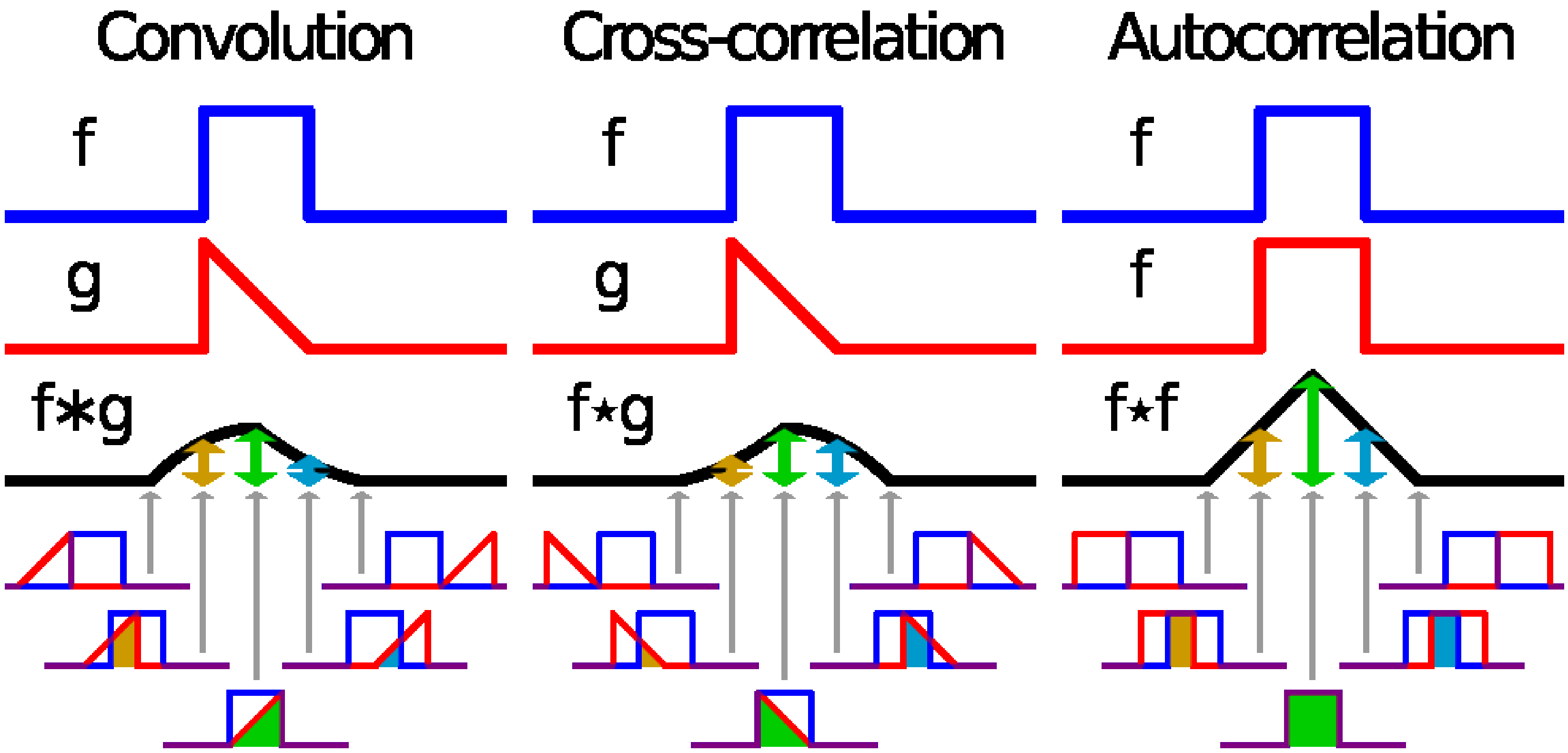
Cross-Correlation

- Usually when we search for a signal, its shape is known, but its position (phase) within the measurement is not!
- To detect phase shifted signals, we introduce a parameter *t* to express the translation in time:

$$\alpha(t) = \int s(\tau)\sigma(t+\tau)d\tau$$

- This integral is called **cross-correlation** between *s* and *σ*.
- As we saw earlier, this is just a linear filter defined by *σ*. This filter is called **matched filter**.
- The filter response is maximal at the 'most probable' position for *σ* to be located within *s*.

Convolution and Cross-Correlation

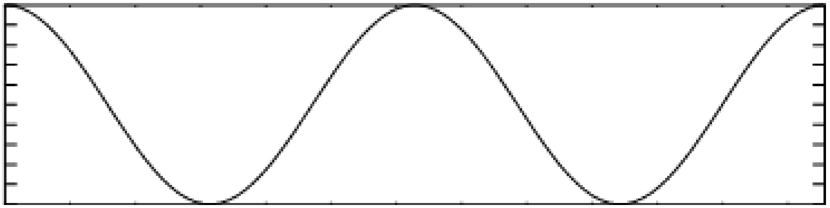


Fourier Theory

- A signal $s(t)$ is called **periodic**, if it forever repeats itself after a certain time, i.e. there is a time p such that

$$s(t+p) = s(t) \qquad \textit{for all } t.$$

- The time p is called the **periodic length** of the signal, its inverse is called the **frequency**.
- Example: the cosine wave $s(t) = \cos(t)$



- Every signal on a finite interval of length l can be made periodic by repeating or mirroring it at the boundaries.

Fourier Theorem

- Any periodic signal is composed of a superposition of pure sine and cosine waves, with suitably chosen amplitudes, whose frequencies are harmonics of the fundamental frequency of the signal.

- In formulas: For any signal *s*(*t*) that is periodic on the interval [0, 2π] there are real-valued constants α_k and β_k such that

$$s(t) = \sum_{k=0}^{\infty} \alpha_k \sin(kt) + \beta_k \cos(kt)$$

- This sum is called the **Fourier series** of *s*. For signals on other intervals and with other periodic lengths, it works essentially the same.

Fourier Series of the Rectangular Wave

The *rectangular* wave with period 2π

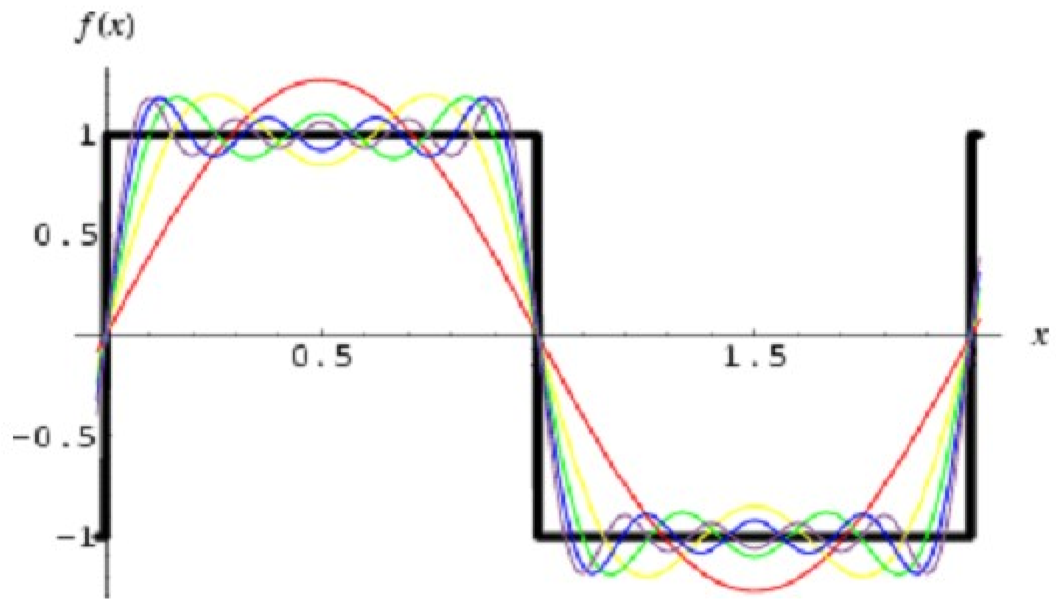
$$s(t) = \begin{cases} 1 & \text{for} \quad 2n\pi \leq x < (2n+1)\pi, \\ -1 & \text{for} \quad (2n+1)\pi \leq x < (2n+2)\pi. \end{cases}$$

has the Fourier series

$$\begin{aligned} s(t) &= \frac{4}{\pi} \sum_{k=1,3,5,\ldots}^{\infty} \frac{1}{k} \sin(kt) \\ &= \frac{4}{\pi} \sin(t) + \frac{4}{3\pi} \sin(3t) + \frac{4}{5\pi} \sin(5t) + \ldots \end{aligned}$$

i.e. $\alpha_k = \begin{cases} \frac{4}{k\pi} & \text{for } k \text{ odd,} \\ 0 & \text{for } k \text{ even,} \end{cases}$ $\qquad \beta_k = 0$ for all k.

Fourier Series of the Rectangular Wave



- Illustration of the first terms in the Fourier series

black:	original wave	red:	k = 1
yellow:	k = 1, 3	green:	k = 1, 3, 5
blue:	k = 1, 3, 5, 7	gray:	k = 1, 3, 5, 7, 9

- Demo: http://www.falstad.com/fourier/

How to find the Fourier coefficients?

$$s(t) = \sum_{k=0}^{\infty} \alpha_k \sin(kt) + \beta_k \cos(kt)$$

- $\sin(kt)$ and $\cos(kt)$ are the signals we want to detect in $s(t)$.
- α_k and β_k indicate how much $\sin(kt)$ and $\cos(kt)$ contribute to the signal $s(t)$.
- We obtain the contribution by integrating the signal against them and dividing by their energy.

$$\alpha_k = \frac{1}{\pi} \int_0^{2\pi} s(t) \sin(kt) \qquad \beta_k = \frac{1}{\pi} \int_0^{2\pi} s(t) \cos(kt)$$

From Discrete to Continuous Frequency Spectrum

- For the Fourier-Series we only need contribution coefficients for sines / cosines with frequencies that are integer multiples of the periodic length.
- We can calculate the integral for any other frequency ω as well.
- We obtain a continuous **frequency spectrum** of the signal:

$$\alpha(\omega) := \frac{1}{\pi}\int_0^{2\pi} s(t)\sin(\omega t)dt$$

$$\beta(\omega) := \frac{1}{\pi}\int_0^{2\pi} s(t)\cos(\omega t)dt$$

- where $\alpha(\omega)$ describes the contribution of a sine-wave with period ω, and $\beta(\omega)$ describes the contribution of a cosine-wave with period ω.

The Fourier-Transform

- Using complex numbers and the Euler identity

$$\cos(\omega t) + i\sin(\omega t) = e^{i\omega t}$$

- This can be combined into one complex-valued equation. This is often renormalized to:

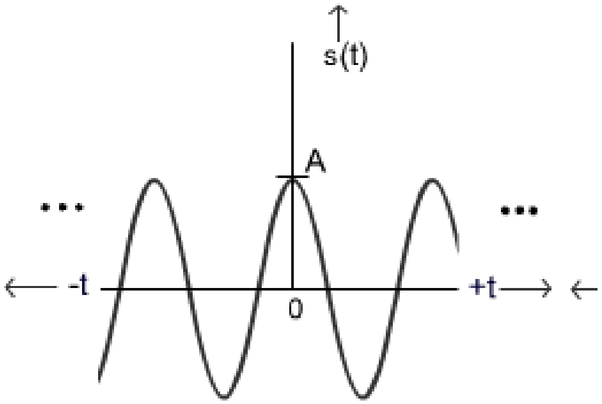
$$\hat{s}(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} s(t)e^{-i\omega t}dt$$

$\hat{s}$ is called the **Fourier-Transform** of s.

Fourier Transform Pairs

TIME DOMAIN

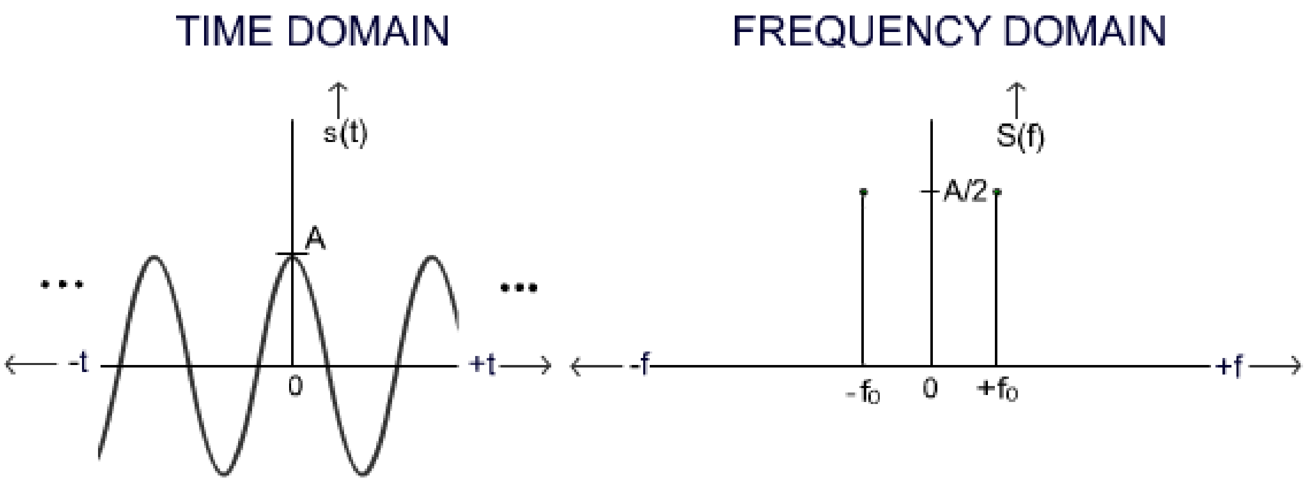
FREQUENCY DOMAIN



?

$s(t) = A\cos(2\pi f_0 t)$

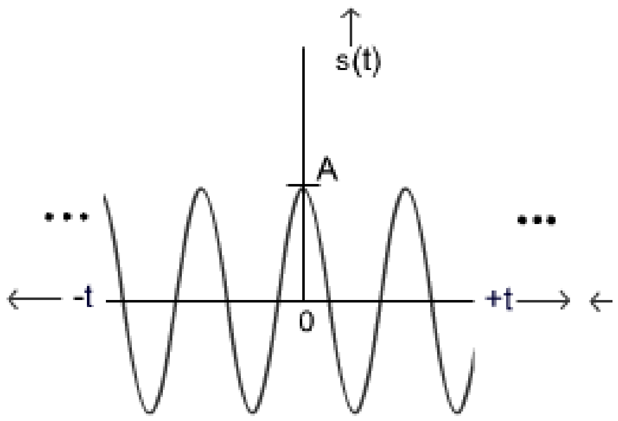
Fourier Transform Pairs



$$s(t) = A\cos(2\pi f_0 t) \quad \longleftrightarrow \quad S(f) = A/2\ \delta(f - f_0) + A/2\ \delta(f + f_0)$$

Fourier Transform Pairs

TIME DOMAIN

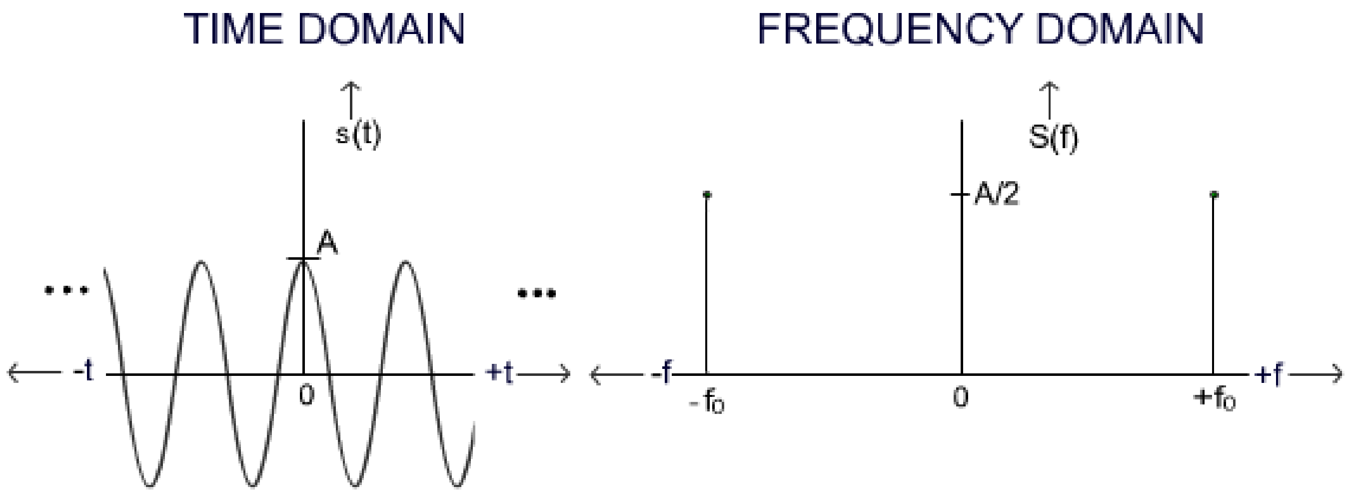


$s(t) = A\cos(2\pi f_0 t)$

FREQUENCY DOMAIN

?

Fourier Transform Pairs



$$s(t) = A\cos(2\pi f_0 t) \longleftrightarrow S(f) = A/2\,\delta(f - f_0) + A/2\,\delta(f + f_0)$$

http://www-rohan.sdsu.edu/~jiracek/DAGSAW/3.4.html

The Inverse Fourier-Transform

- We can reconstruct s(t) from its Fourier transform at integer points.
- If we know all of $\hat{s}$, there is an explicit inversion filter

$$s(t) = \mathcal{F}^{-1}\hat{s}\ (t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{s}(\omega) e^{i\omega t} d\omega$$

Note: sum over all the frequencies

- We can study any signal either in the time domain as $s(t)$, or in the frequency domain as $\hat{s}(\omega)$.
- Since s and $\hat{s}$ both are signals, we can also apply filters in one domain to the other:

$$\begin{array}{c} s \\ T \downarrow \\ Ts \end{array} \quad \text{or} \quad \begin{array}{ccc} s & \xrightarrow{\mathcal{F}} & \hat{s} \\ & & \downarrow T' \\ s' & \xleftarrow{\mathcal{F}^{-1}} & T'\hat{s} \end{array}$$

Why are we interested in the Fourier Transform?

- **Can manipulate an image in the frequency domain**

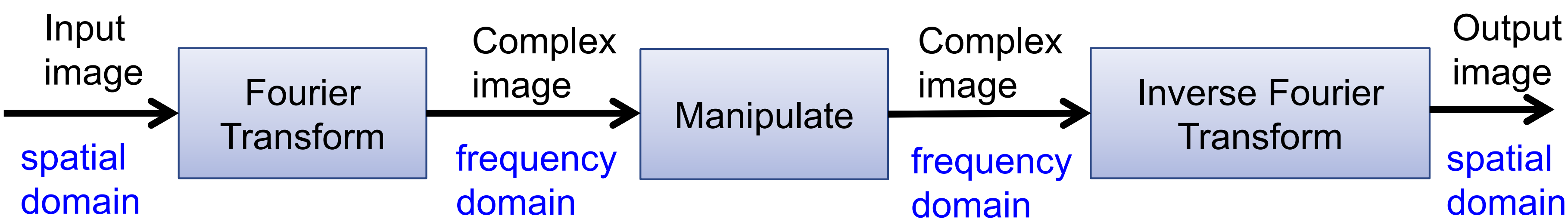
  We can manipulate an image (e.g., sharpening it) in the Fourier domain (or frequency domain) by firstly applying the Fourier Transform to the input image (in the spatial domain)

- **Can represent an Image using its Fourier Components**

  We can also represent an image using its Fourier components – knowing these components allow us to reconstruct the image back

Image Processing Pipeline in the Frequency Domain

- An appropriate manipulation in the frequency domain can lead to
  - the output image being smoothed (noise removal)
  - the output image being sharpened
  - certain features being removed in the output image because they fall inside a specific frequency range

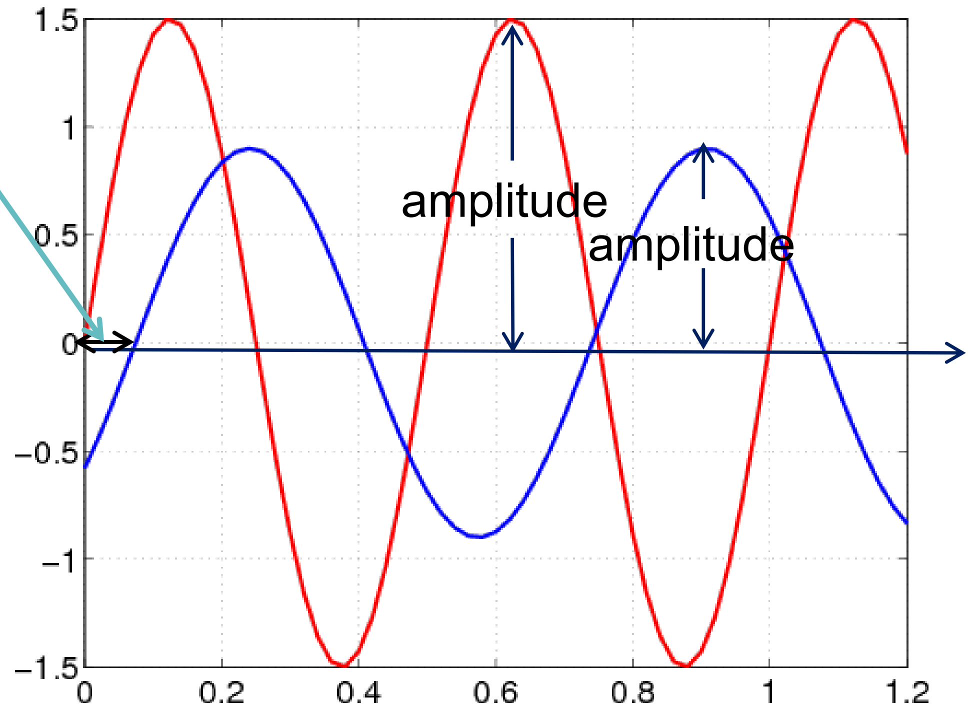


- More on image processing in the frequency domain will come later.

Amplitude and Phase

- At a frequency value ω_0, the term $F(\omega_0)$ will be a complex number of the form $a + i\,b$. The **amplitude** at that frequency ω_0 is $\sqrt{a^2 + b^2}$ and the **phase angle** is $\tan^{-1}\left(\frac{b}{a}\right)$

The blue curve has a negative phase angle. The red curve has a 0 phase angle.



Amplitude and Phase (cont.)

- In many applications only the amplitude information is used.
  However, in images (and in sound signals) phase carries most of the information

- The classic demonstration of this was devised by Oppenheim & Lim in 1981: *if you construct a synthetic image composed of the amplitude information from image A and the phase information from image B, it is image B that you see in the result.*

Reference: Oppenheim, A. V. and Lim, J. S. "The Importance of Phase in Signals", Proceedings of the IEEE, vol. 69, no. 5, pp. 529-541, May 1981.

Amplitude and Phase (cont.)

image A

image B

Reconstructed image using the phase information from image B and amplitude information from image A

Oppenheim & Lim's demonstration

Important Properties of the Fourier Transform

- If $F(\omega)$ is the Fourier Transform of $f(x)$,
  $G(\omega)$ is the Fourier Transform of $g(x)$, and
  a and b are constants
  - **The Fourier Transform is linear**

$$\mathscr{F}\big(a\,f(x) + b\,g(x)\big) = aF(\omega) + bG(\omega)$$

  - Changing the spatial scale inversely affects frequency and amplitude

$$\mathscr{F}\big(f(ax)\big) = \frac{1}{a}F\left(\frac{\omega}{a}\right)$$

  - Shifting the function only changes the phase of the spectrum

$$\mathscr{F}\big(f(x-a)\big) = e^{-i2\pi\omega a}F(\omega)$$

    Note that the amplitude spectrum is invariant to spatial shift.

The Convolution Theorem

- **The convolution Theorem** states that:

  Convolution in the spatial domain corresponds to **multiplication** in the frequency domain and vice versa. That is,

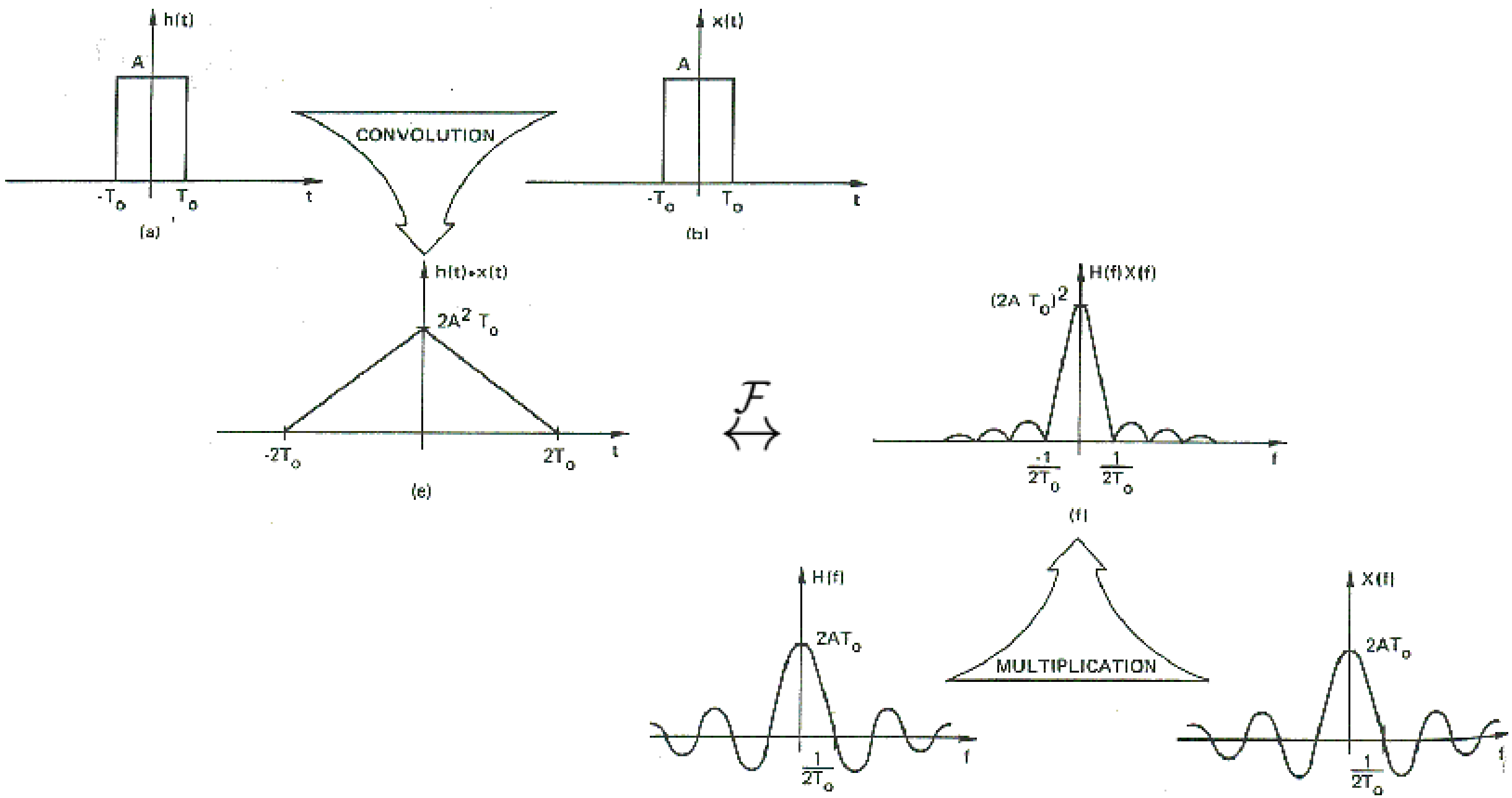
  $$f(x) \otimes g(x) \leftrightarrow F(\omega)G(\omega)$$

  $$f(x)\, g(x) \leftrightarrow F(\omega) \otimes G(\omega)$$

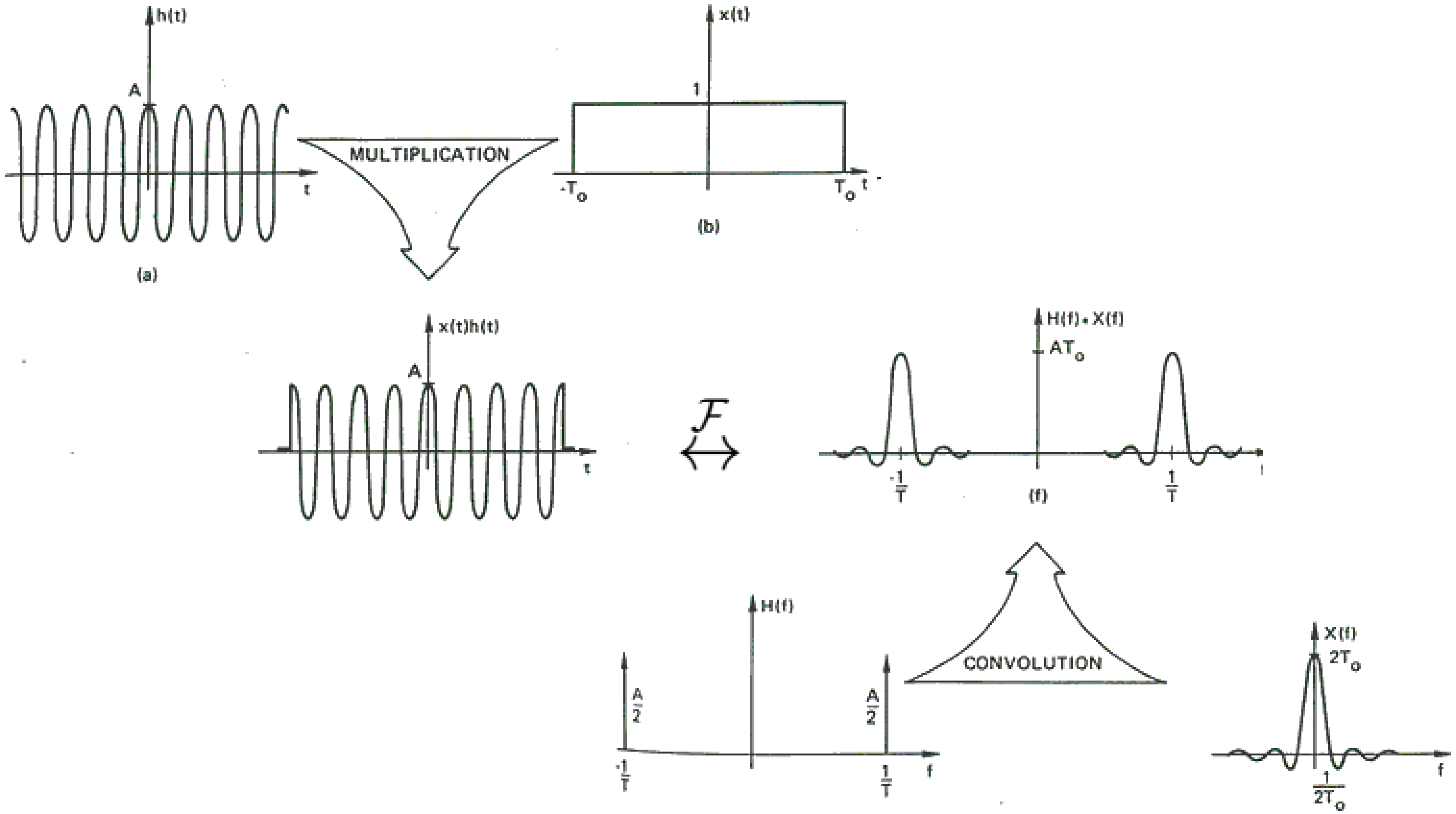
- For efficiency purpose, convolution in one domain is often implemented as multiplication in the other domain.

- **Division** in the frequency domain corresponds to **deconvolution** in the spatial domain. This can be the basis by which blurred images can be restored.

The Convolution Theorem



The Convolution Theorem



Discrete Fourier Transform (DFT)

- In the continuous domain we have an infinite number of basis functions.
- In an image we have a discrete number of points. The Discrete Fourier Transform provides information over a discrete number of frequencies.

  What frequencies are these?

  - To answer this question, we need to understand the sampling of data

Some terminology:

- In the time domain:
  - Sampling rate (or sampling frequency): #samples/second (Hz)
  - Sampling interval = 1/(sample rate), e.g., samples are 0.3 seconds apart
- In the spatial domain:
  - Sampling rate (or sampling frequency): #samples/mm
  - Sampling interval = 1/(sample rate), e.g., samples are 0.07mm apart

The Nyquist Critical Frequency

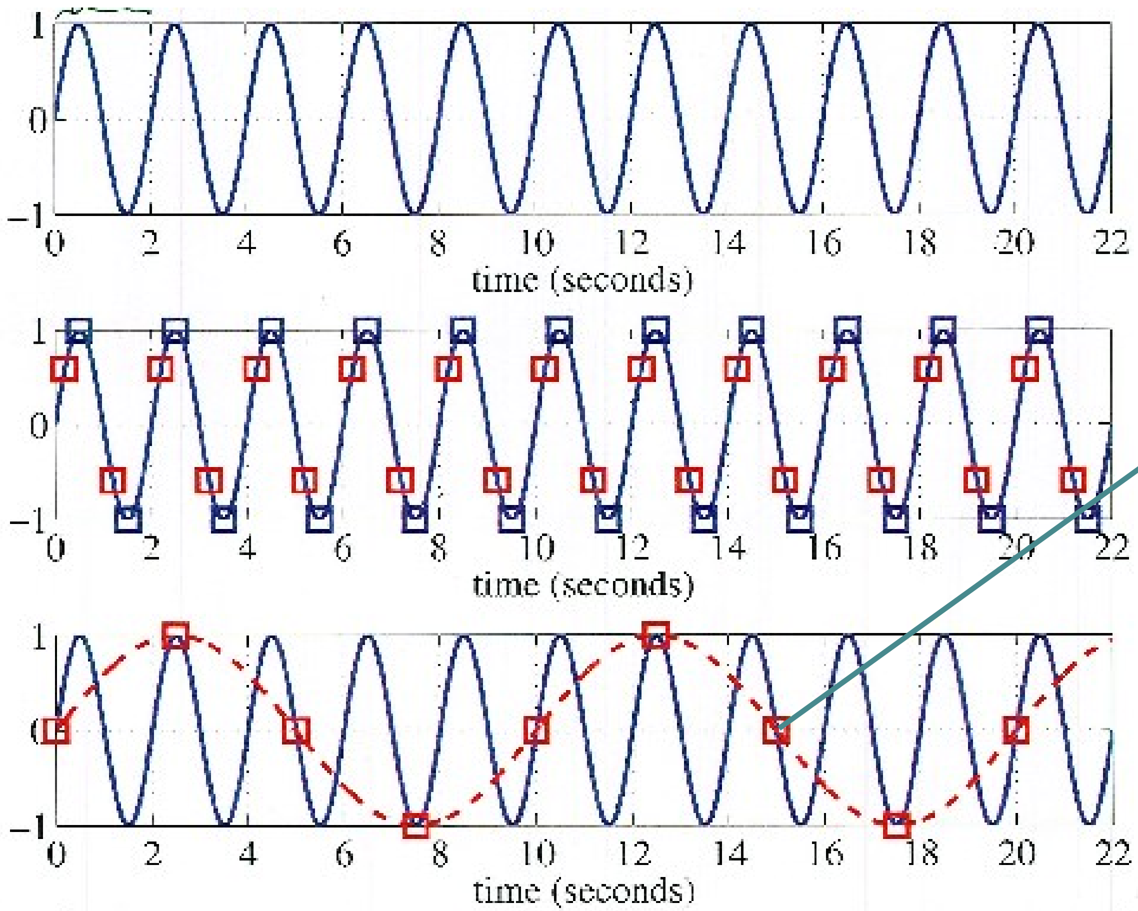
- For any sampling frequency ω_s, there is a special frequency called the **Nyquist critical frequency**

$$\omega_c = \frac{\omega_s}{2}$$

It is the highest frequency that can be represented by something sampled at that sampling frequency.

Conversely, if we know the Nyquist critical frequency ω_c then we can work out the <u>minimum</u> sampling frequency ω_s, which should be double of ω_c.

Sampling and Reconstruction



Aliasing: high spatial frequency components appear as low spatial frequency components in the sampled signal

DFT and Inverse DFT

- **Discrete Fourier Transform (1D):**

  Let $f(x)$ be a 1D discrete signal of N samples. The DFT of $f(x)$ is given by

$$F(\omega) = \sum_{x=0}^{N-1} f(x)\, e^{-i2\pi\omega x/N}$$

note the minus sign

sum over x

- **Inverse Discrete Fourier Transform (1D):**

  Let $F(\omega)$ be the Discrete Fourier Transform of a 1D signal having N samples. The inverse DFT is given by

$$\hat{f}(x) = \frac{1}{N} \sum_{\omega=0}^{N-1} F(\omega)\, e^{i2\pi\omega x/N}$$

no minus sign

note the scale factor $1/N$

sum over frequency ω

DFT and Inverse DFT (cont.)

- **Discrete Fourier Transform (2D):**

  Let $f(x,y)$ be an $N \times M$ image. The DFT of $f(x,y)$ is given by

$$F(\omega,\nu) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y)\, e^{-i2\pi\left(\frac{\omega x}{N}+\frac{\nu y}{M}\right)}$$

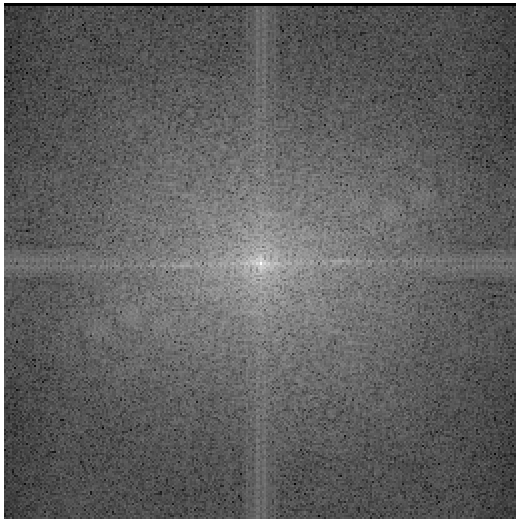
- **Inverse Discrete Fourier Transform (2D):**

  Let $F(\omega,\nu)$ be the Discrete Fourier Transform of an $N \times M$ image. The inverse DFT is given by

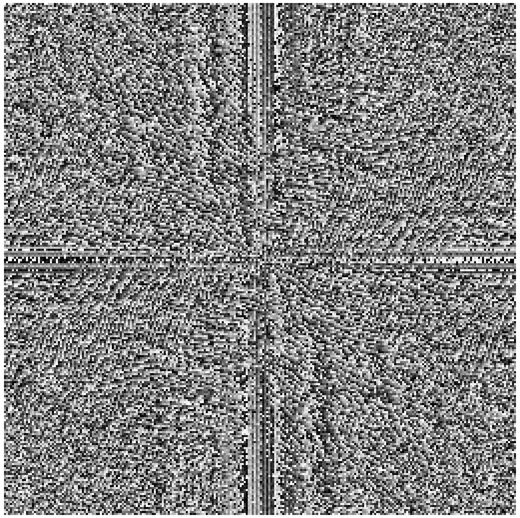
$$\hat{f}(x,y) = \frac{1}{NM} \sum_{\omega=0}^{N-1} \sum_{\nu=0}^{M-1} F(\omega,\nu)\;\, e^{i2\pi\left(\frac{\omega x}{N}+\frac{\nu y}{M}\right)}$$

Spatial Frequencies and DFT

Image

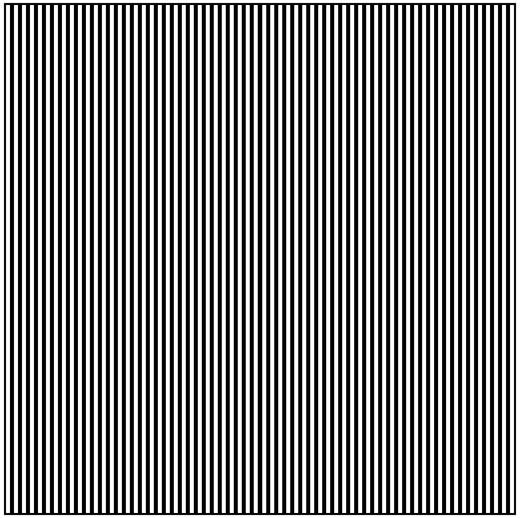


DFT Magnitude

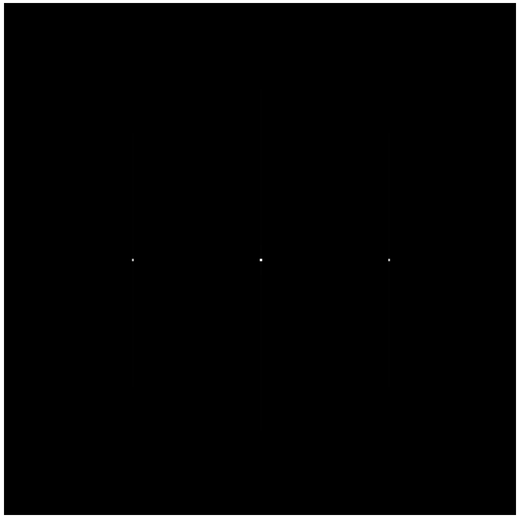


DFT Phase

Spatial Frequencies and DFT

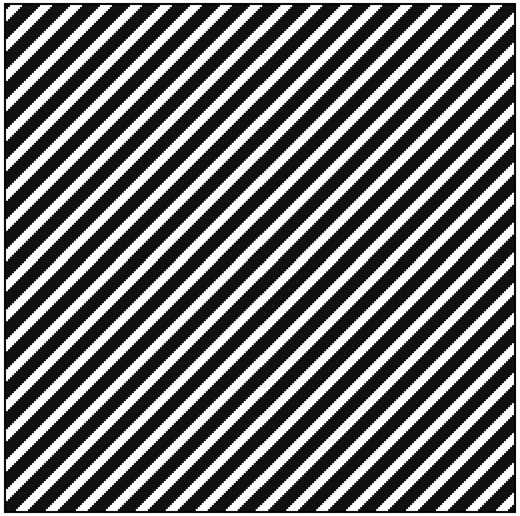


Image

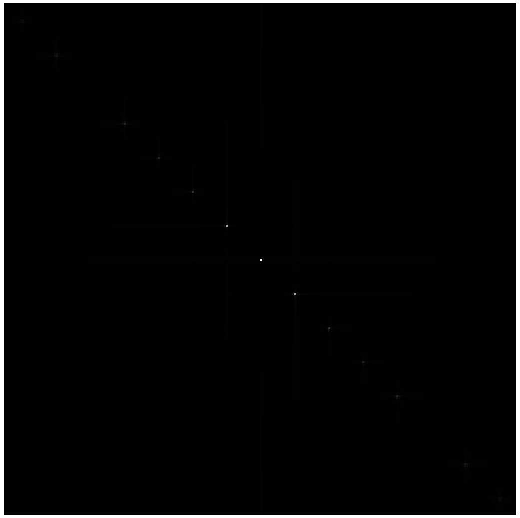


DFT Magnitude

Spatial Frequencies and DFT

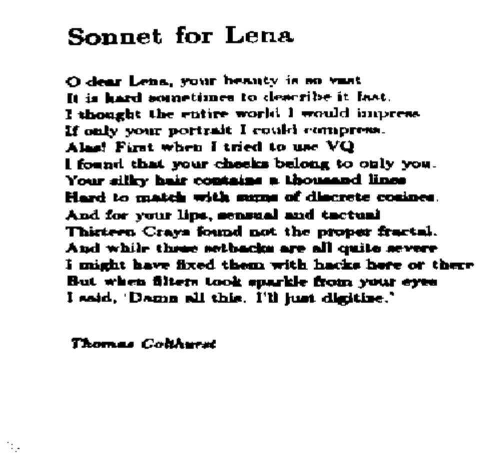


Image



DFT Magnitude

Spatial Frequencies and DFT

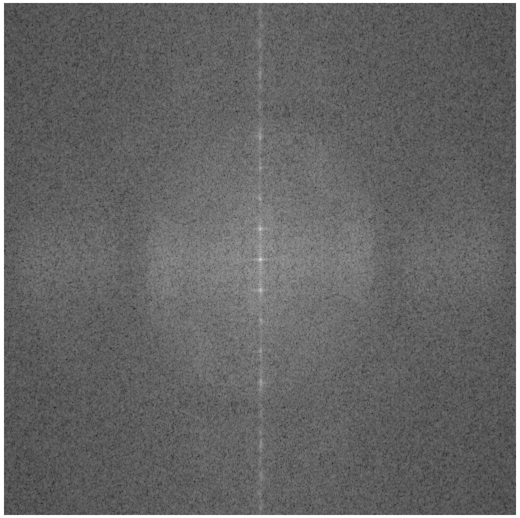


Sonnet for Lena

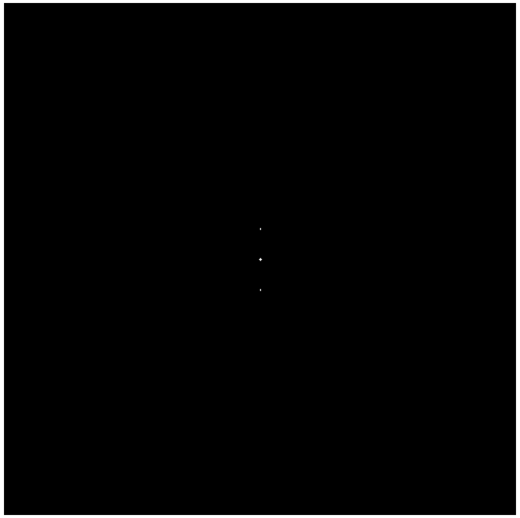
O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Image

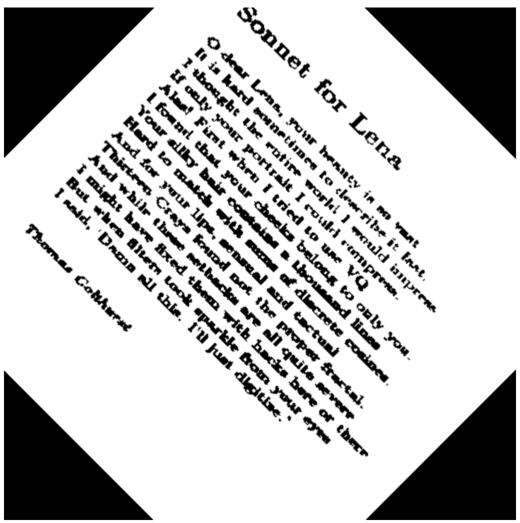


DFT Magnitude

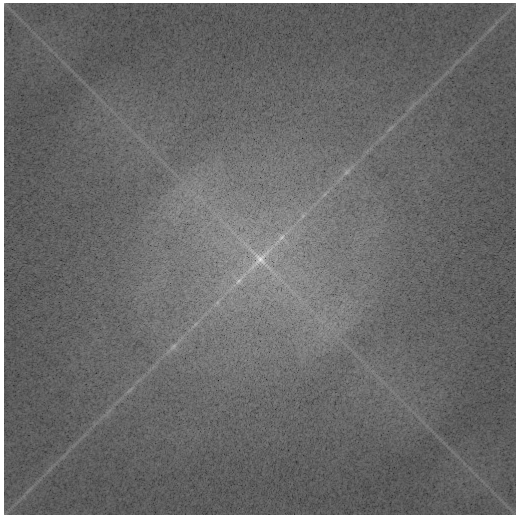


DFT Phase

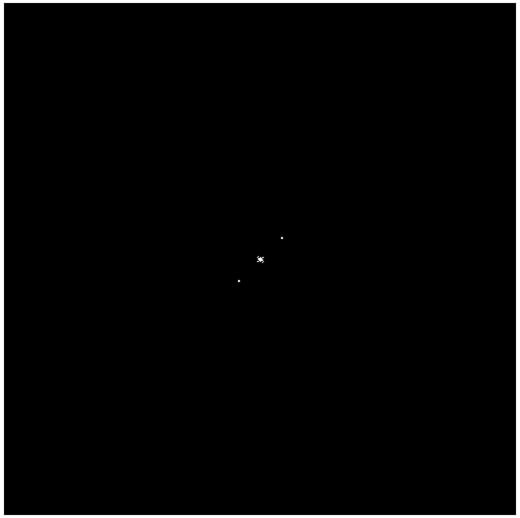
Spatial Frequencies and DFT



Image

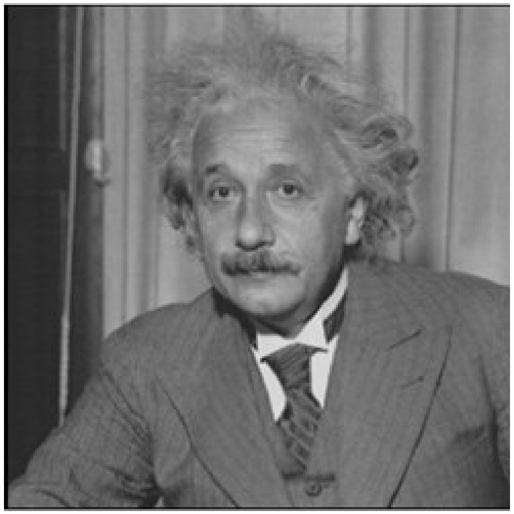


DFT Magnitude

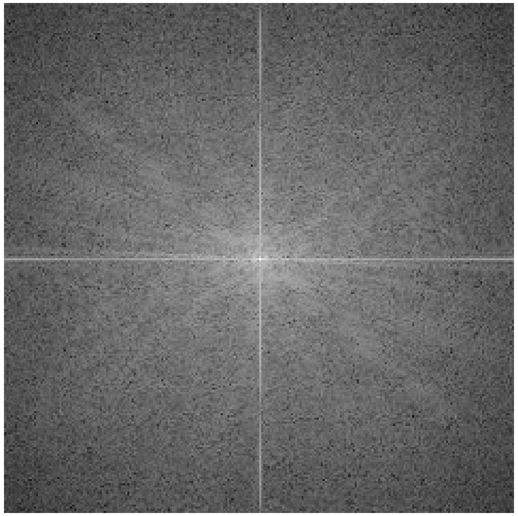


DFT Phase

DFT of Natural Images



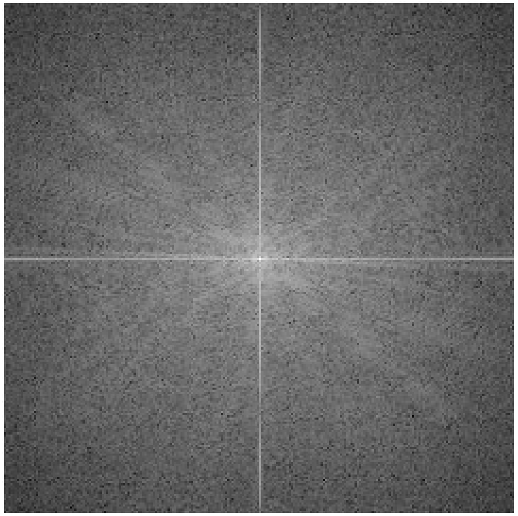
Image



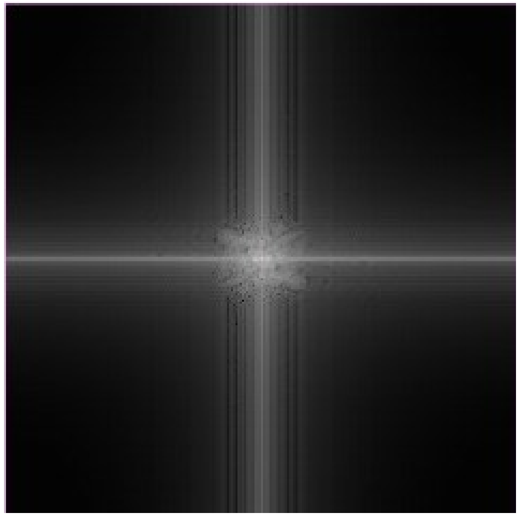
DFT Magnitude

Low-Pass Filtering

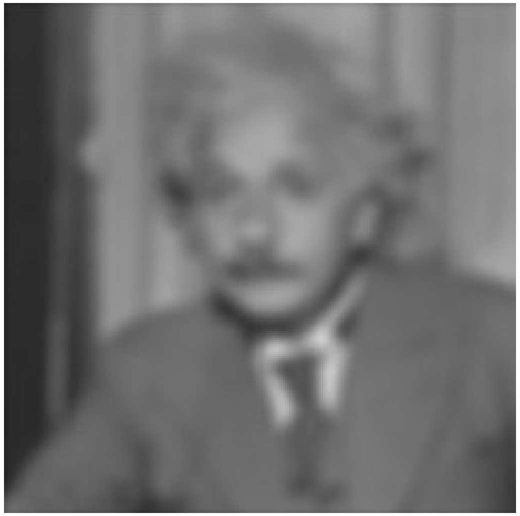
- Removing all *high* spatial frequencies from a signal to retain only *low* spatial frequencies is called **low-pass filtering**.



Old Spectrum



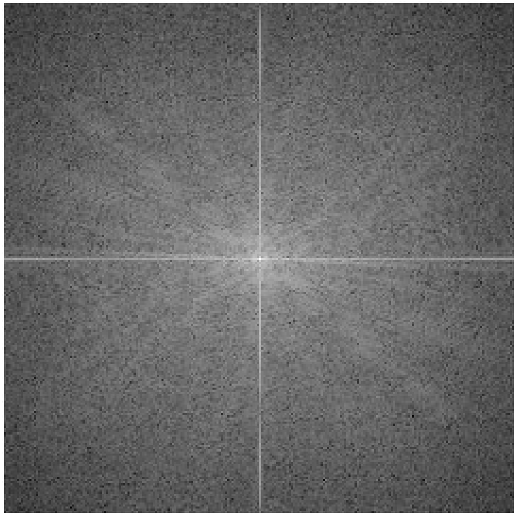
New Spectrum



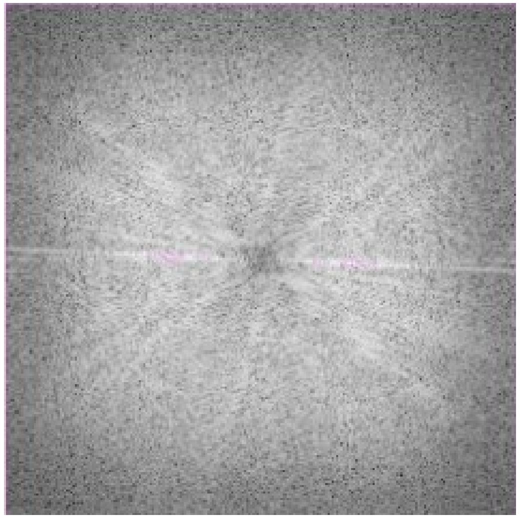
Low-Pass Filtered Image

High-Pass Filtering

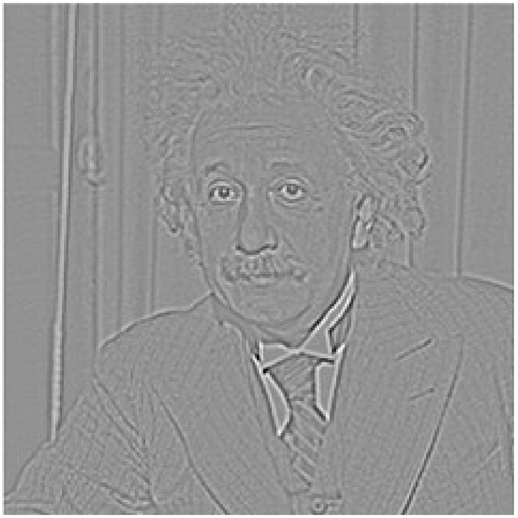
- Removing all *low* spatial frequencies from a signal to retain only *high* spatial frequencies is called **high-pass filtering**.



Old Spectrum



New Spectrum



High-Pass Filtered Image

Fast Fourier Transform (FFT)

- A simple implementation of the Discrete Fourier Transform of an N-sampled signal requires **O(N^2)** operations.
- The **Fast Fourier Transform (FFT)** is an ingenious algorithm which exploits properties of the Fourier Transform to enable the transformation to be done in **O($N \log_2 N$)** operations.

  However, ideally the size of the data should be a power of 2.

  If this is not the case, the data is either truncated or padded out with zeros.

- The FFT has made Fourier analysis a practical reality.

  e.g., for a 256 × 256 image (total is: $N = 2^{16}$ pixels)

  the 'slow' Fourier Transform needs 2^{32} complex multiplications

  FFT needs $N \log_2 N = 2^{16} \times 16$

  $= 2^{20}$ complex multiplications

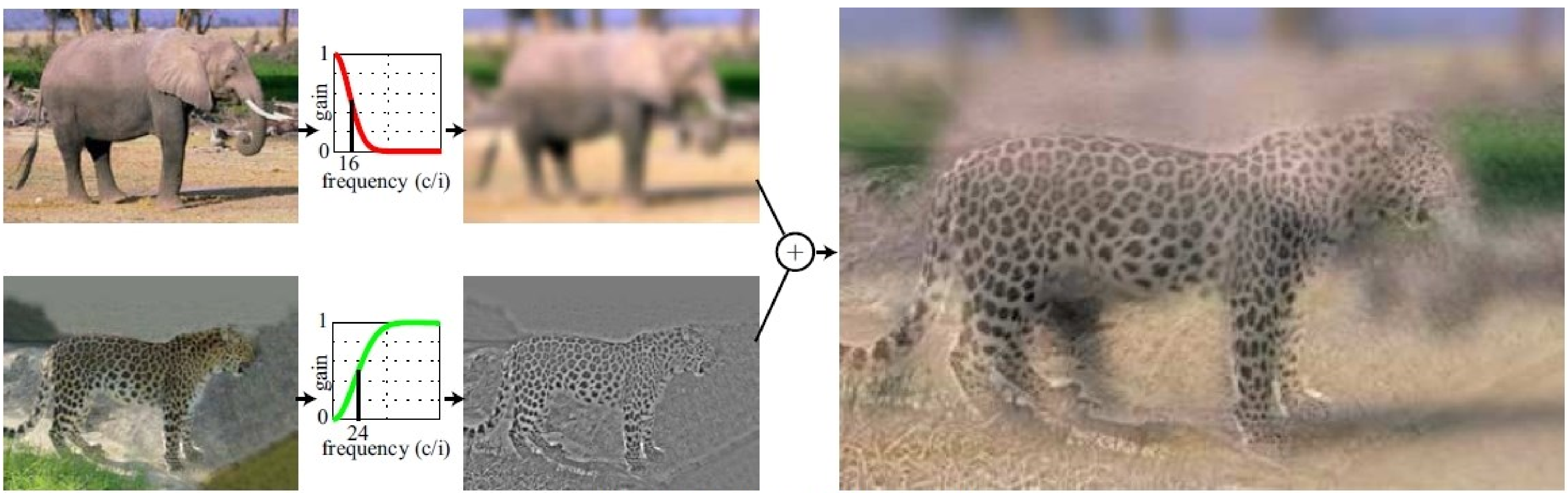
  This is 2^{12} times faster !!!

Doing Fourier Transform in Matlab

- **Matlab functions:**
  - `fft` – for computing 1D Fourier Transform
  - `ifft` – for computing the inverse FT
  - `fftshift` – for shifting quadrants (for display purpose). Matlab returns the Fourier components for the 0 frequency (the DC term) as the first element of the array, whereas we want the DC term to be roughly in the middle of the array (see also `ifftshift`)
  - `fft2` – for 2D FT
  - `ifft2` – for 2D inverse FT
  - `abs` – use this function to get the amplitude
  - `angle` – and this one to get the phase angle
  - `log`, `sqrt` – the range of Fourier amplitudes for any given image is usually very large. Use these functions appropriately to rescale the range
  - `imagesc` – use this to display an image

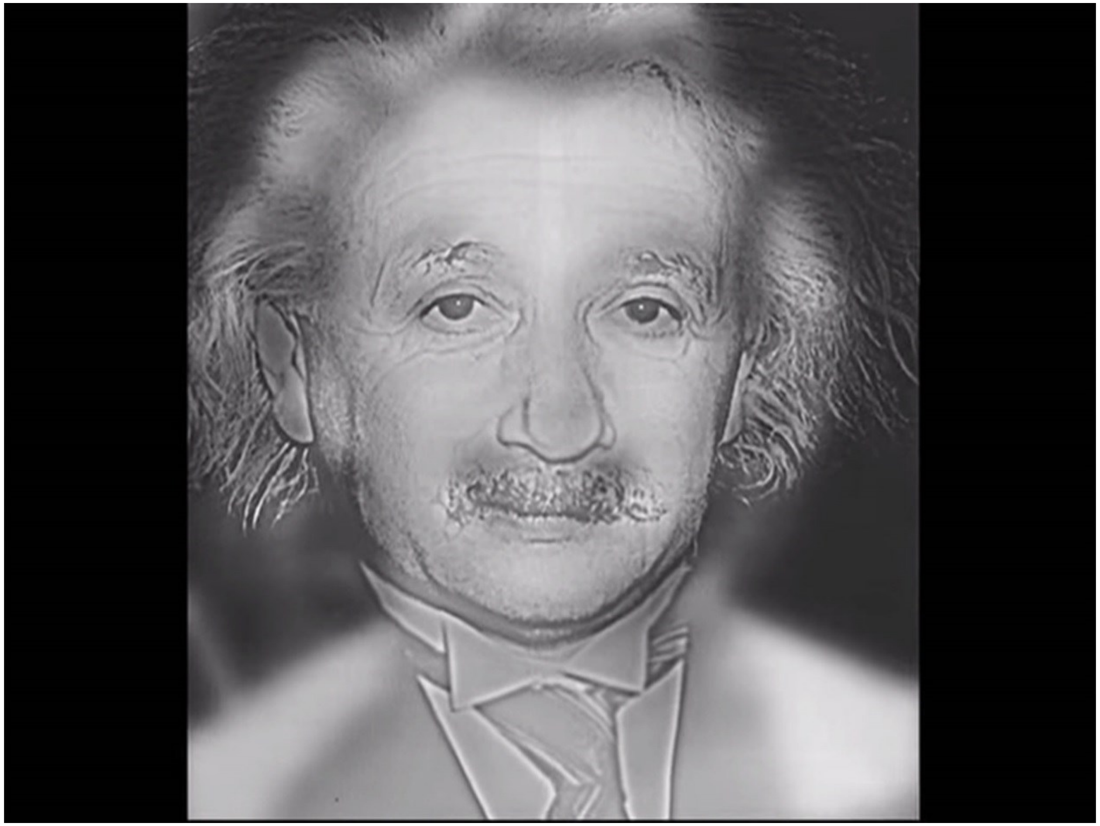
Week 03 Lab : Hybrid Images

- A. Oliva, A. Torralba, P.G. Schyns, "Hybrid Images“, ACM Transactions on Graphics, ACM Siggraph, 2006. (presentation, paper)



(Extracted from Figure 2 of Oliva et al [1])

Week 03 Lab (a very popular example)



Week 03 Lab submission

- **Your tasks...**

Write a Matlab script and save it to a file named **test1.m**. Demonstrate in the script file how you generate a hybrid image using two input images. Inspect the hybrid image at different distances. Adjust the values *c* and *n* of your low-pass and high-pass filters until the hybrid image reveals either input image when viewed at different distances.Your Matlab script should read in two input images, call the **lowpassfilter.m** and **highpassfilter.m** functions and construct the resultant image. Your code should display the resultant image. When we mark your lab exercises, we will run your Matlab script and inspect the display. Apart from the resultant hybrid image, display intermediate images as well such as those shown in the example above to demonstrate your understanding of the hybrid image construction process.

Repeat the same process above for another pair of input images. Save the Matlab script to **test2.m**.

- **Submission Requirements:** Submit to cssubmit the following as a single zip file:

1. your **test1.m** script file.
2. your **test2.m** script file.
3. your own image pairs (other than the sample ones supplied) that you use in test1.m and test2.m.

Ensure that your submission is complete and also includes any other functions that you have used such as the **lowpassfilter.m** and **highpassfilter.m** from Prof. Kovesi's website

Summary

- Images as 2-D signals
- Linear and non-linear filters
- Convolution
- Fourier Series and Fourier Transform
- Discrete Fourier Transform