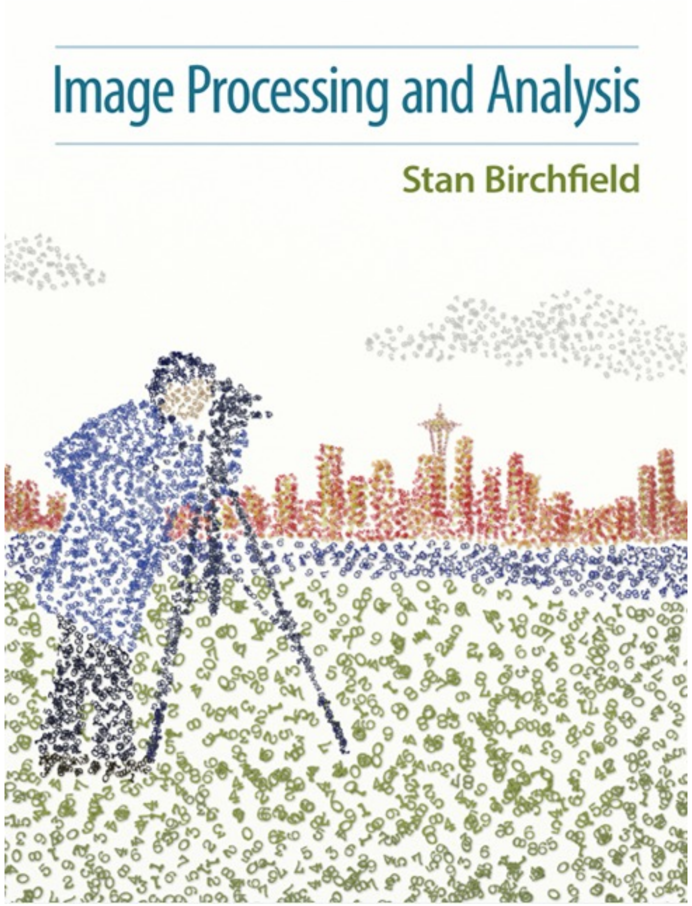
Prof. Kjersti Engan

ELE510 Image processing and computer vision

Frequency Domain processing, Fourier transform (Chap 6.1-6.2 Birchfield) 2023

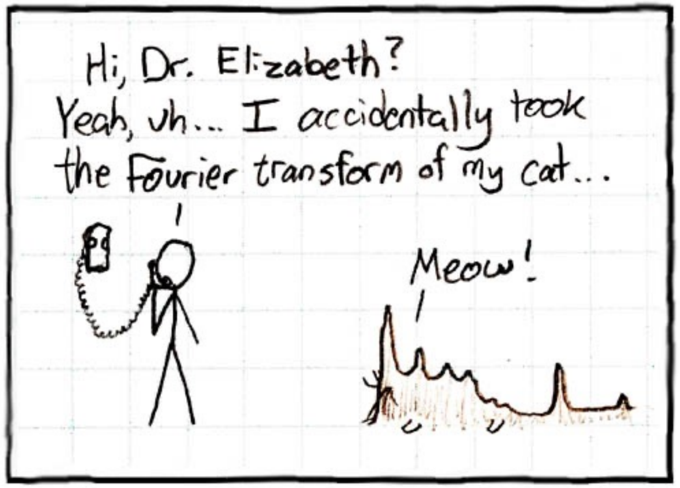


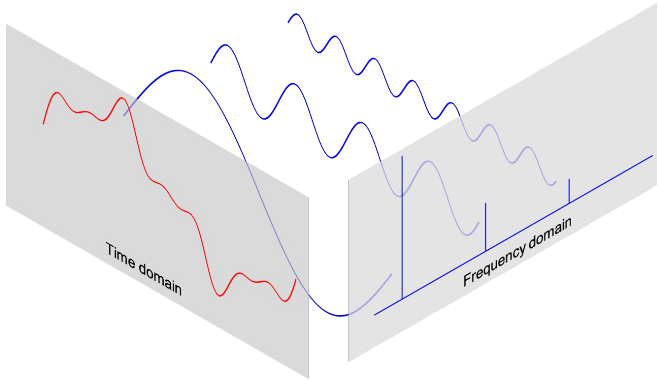
Parts in course presentations is material from Cengage learning. It can be used for teaching, and it can be share it with students on access controlled web-sites (Canvas) for use in THIS course. **But it should not be copied or distributed further in any way** (by you or anybody).

Introduction to Fourier Transform

Three points from the topic:

1. What is a Fourier series and a (discrete) Fourier transform?
2. How can that help us in understanding signals and images?
3. Properties of Fourier Transform?

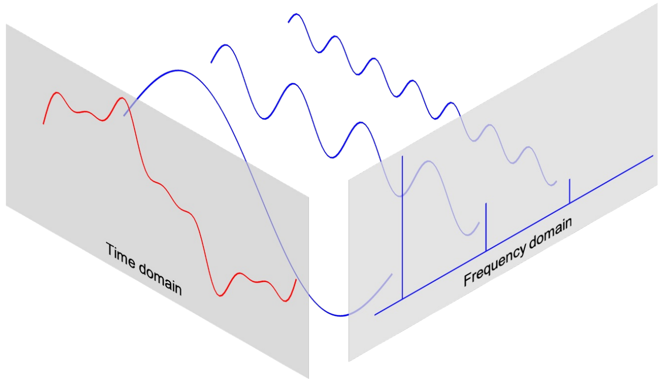




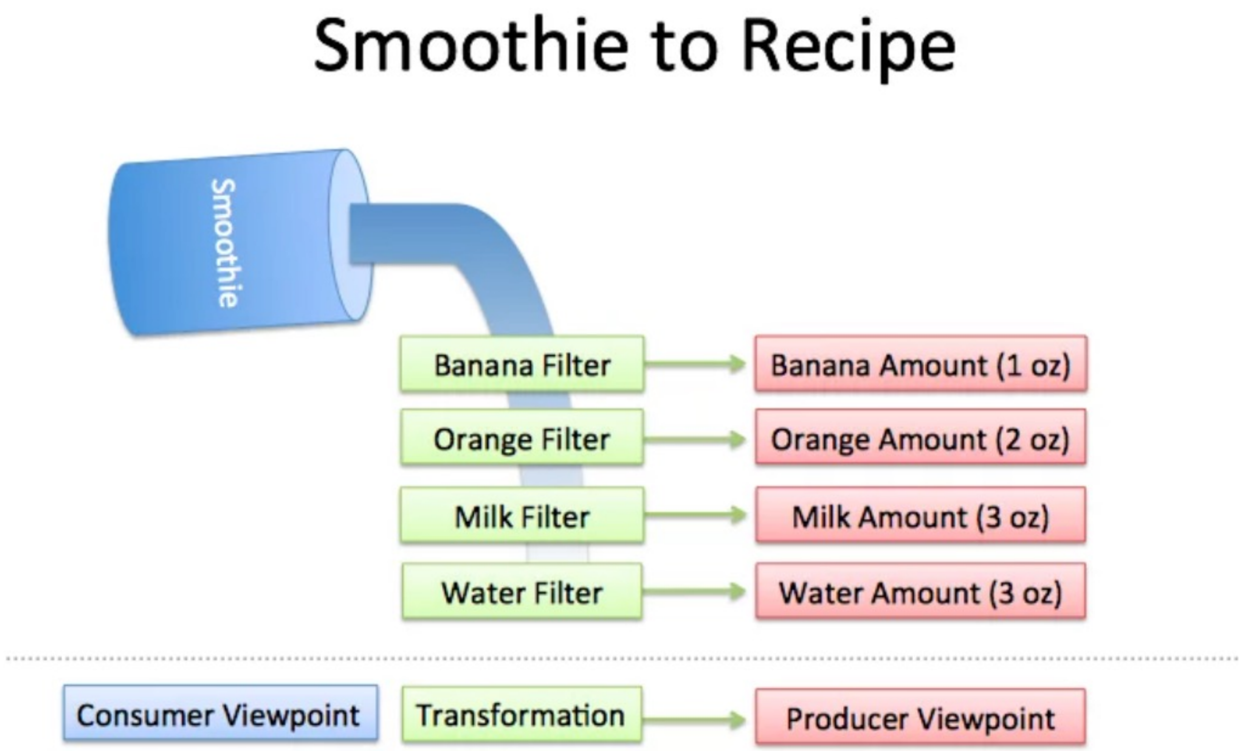
https://www.youtube.com/watch?v=Qm84XIoTy0s

What is the (Discrete) Fourier Transform?

- Remeber the Singular Value Decomposition video .. “eigenimages” / “buidlingblock-images” that we could use to represent an image.
  - specialized for a particular image/signal
- DFT is a very much used transform that uses general buildingblocks
- Any signal or image can be represented as a linear combination of the basic building-blocks.
- Can use it to DESCRIBE a signal (image)
  - Find features (represent)
  - Analyze
  - Compress (store)



Time (space) vs. Freq. domain – main idea



The Fourier Transform finds the recipe for a signal, like our smoothie process:

Start with a time-based signal
Apply filters to measure each possible "circular ingredient"
Collect the full recipe, listing the amount of each "circular ingredient»

https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/

- We can reverse-engineer the recipe by filtering each ingredient.

The catch?

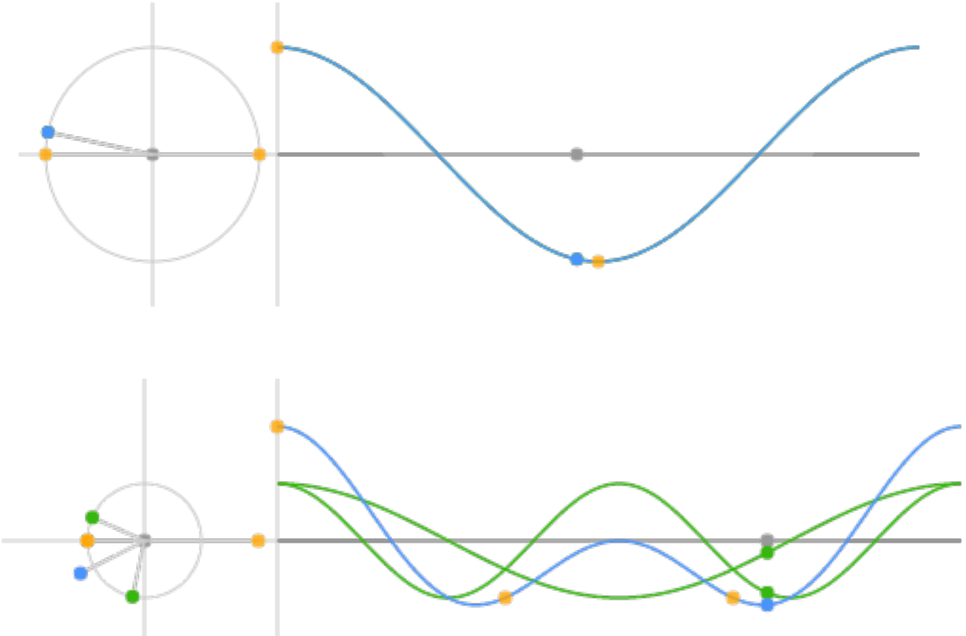
- **Filters must be independent.** The banana filter needs to capture bananas, and nothing else. Adding more oranges should never affect the banana reading.

- **Filters must be complete**. We won't get the real recipe if we leave out a filter ("There were mangoes too!"). Our collection of filters must catch every possible ingredient.

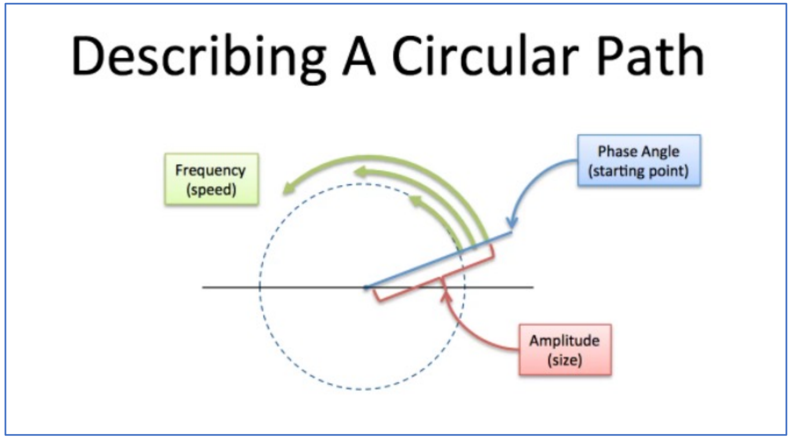
- **Ingredients must be combineable.** Smoothies can be separated and re-combined without issue. The ingredients, when separated and combined in any order, must make the same result.

https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/

- The Fourier Transform takes a **time-based pattern**, measures every possible cycle, and returns the overall "cycle recipe" (the amplitude, offset, & rotation speed for every cycle that was found).



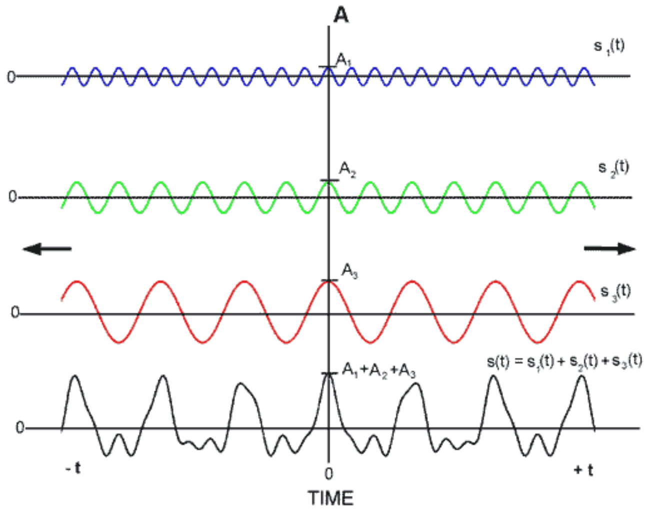
Blue is the sum of the two green (1 Hz and 2Hz, both amplitude 1)

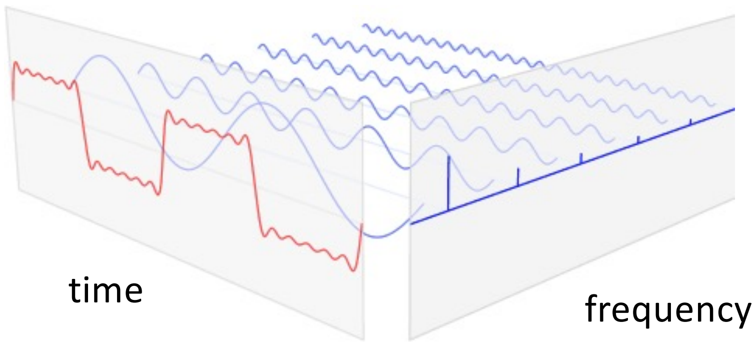


https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/

Time (space) vs. Freq. domain – main idea

(6.1) Fourier transform – frequency content





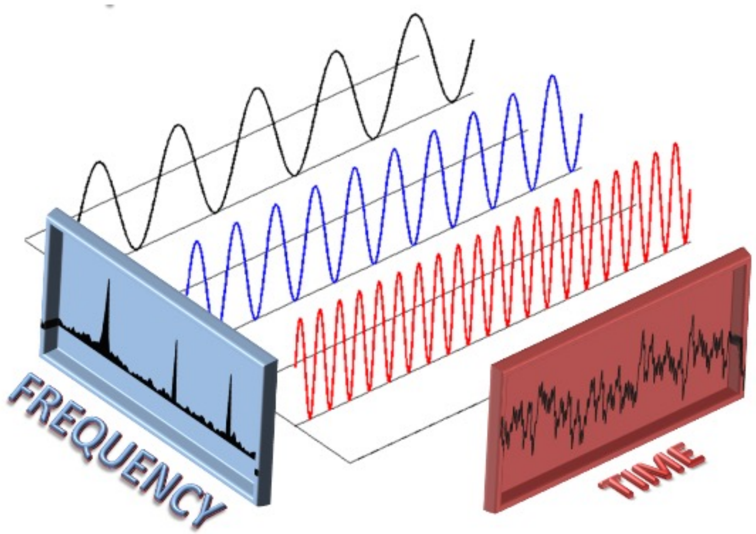
A signal can be regarded as a weighted sum of different frequency components.

How can Fourier Transform be useful?

- If sound waves can be separated into ingredients (bass and treble frequencies), we can boost the parts we care about, and hide the ones we don't.

- If data can be represented with oscillating patterns, perhaps the least-important ones can be ignored. This *lossy compression* can drastically shrink file sizes (and is why JPEG and MP3 files are much smaller than raw .bmp or .wav files).

- Many more examples…



Fourier transform (4 types)

1. Fourier series: any periodic function can be represented as a weighted sum of sines and cosines.
2. Fourier transfrom: Extend to aperiodic signals
3. Discrete time Fourier transform : Discrete signals
4. Discrete Fourier Transform: Discrete in both domains - can be used in computer.

In general we look at complex sinusoids as our building blocks, not sines and cosines

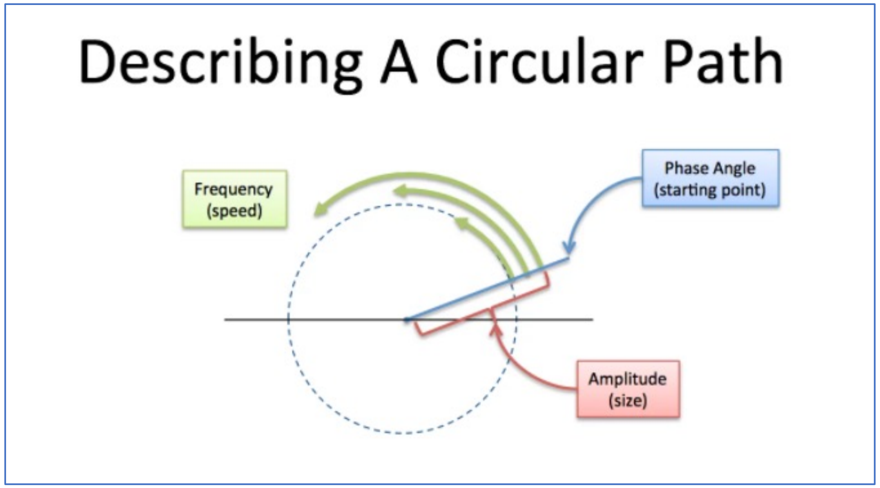
Can we extend to 2D (images)? -> Yes – straight forward and separable!

Fourier transform – building blocks

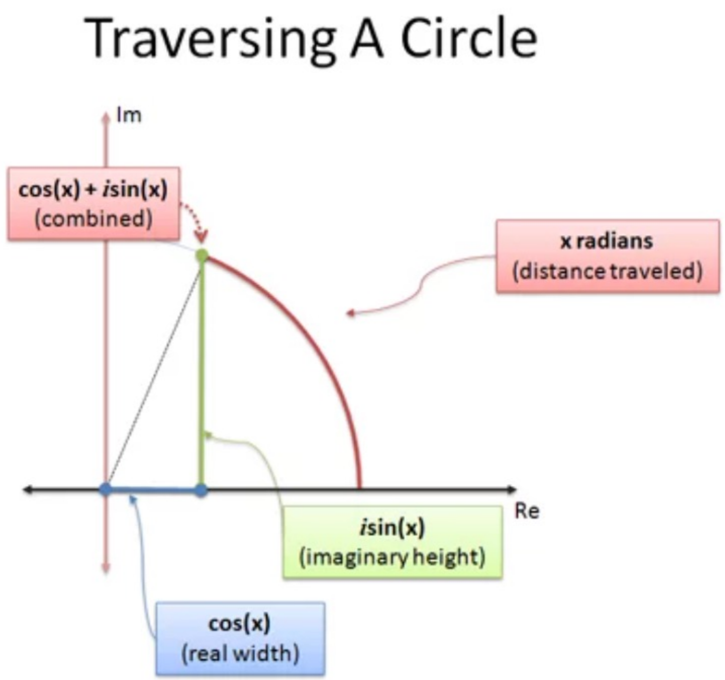
We look at complex sinusoids as our building blocks:

Eulers formula: $e^{jw} = cosw + j \cdot sinw$

Omega (ω) is angular frequency (rad/s). $\omega = 2\pi f$
where f is the frequency in Hz (=1/s)



Understanding Eulers formula



cos(x) is the x-coordinate (horizontal distance)
sin(x) is the y-coordinate (vertical distance)

cos(x) + *j* sin(x)

is a clever way to combine the x and y coordinates into a single number.

The analogy "complex numbers are 2-dimensional" helps us interpret a single complex number as a position on a circle.

https://betterexplained.com/articles/intuitive-understanding-of-eulers-formula/

Fourier transform (continous)

- **Fourier transform *G(f)*:** the integration of the continuous signal after first multiplying by a certain complex exponential:

$$\boxed{G(f) \equiv \mathcal{F}\{g\} \equiv \int_{-\infty}^{\infty} g(t) e^{-j2\pi ft}\, dt}$$

$$\boxed{g(t) = \mathcal{F}^{-1}\{G\} \equiv \int_{-\infty}^{\infty} G(f) e^{j2\pi ft}\, df}$$

- If t is measured in seconds, then f is measured in inverse seconds, also known as hertz.
- By applying **Euler's formula:**

$$G(f) = \underbrace{\int_{-\infty}^{\infty} g(t) \cos 2\pi ft\, dt}_{G_{even}} + j \underbrace{\int_{-\infty}^{\infty} - g(t) \sin 2\pi ft\, dt}_{G_{odd}}$$

Captures freq in signals with even symmetry

Captures freq in signals with odd symmetry

Discrete time Fourier transform (1D)

Notation common from signal processing.
Omega (ω) represents the continous angular frequency.
x(n) a signal sampled a time points n

$$X(e^{jw}) = \mathcal{F}(x(n)) = \sum_{n=-\infty}^{\infty} x(n)e^{-jwn}$$

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{jw})e^{jwn}\,dw$$

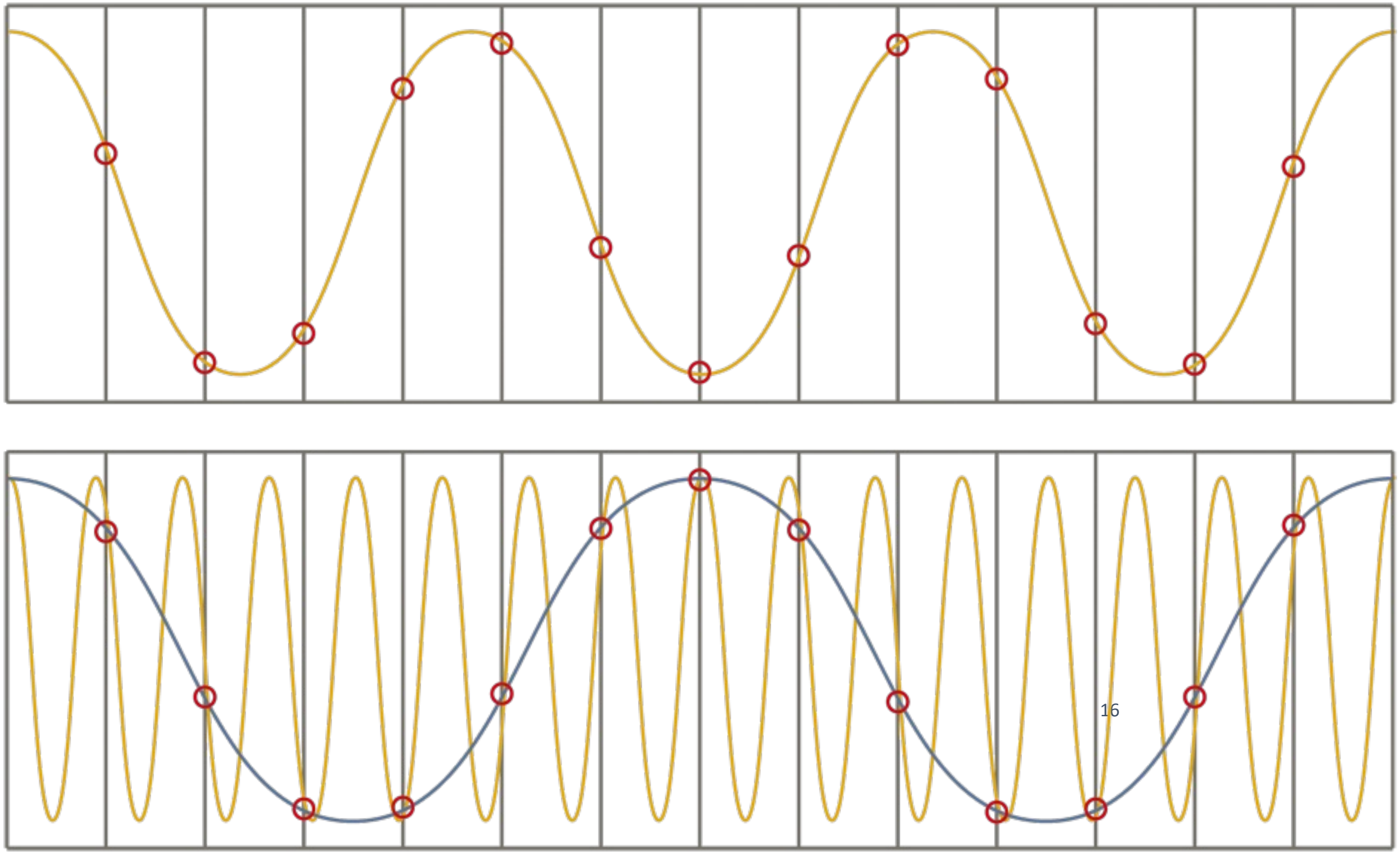
Eulers formula:

$$e^{jx} = \cos x + j\sin x$$

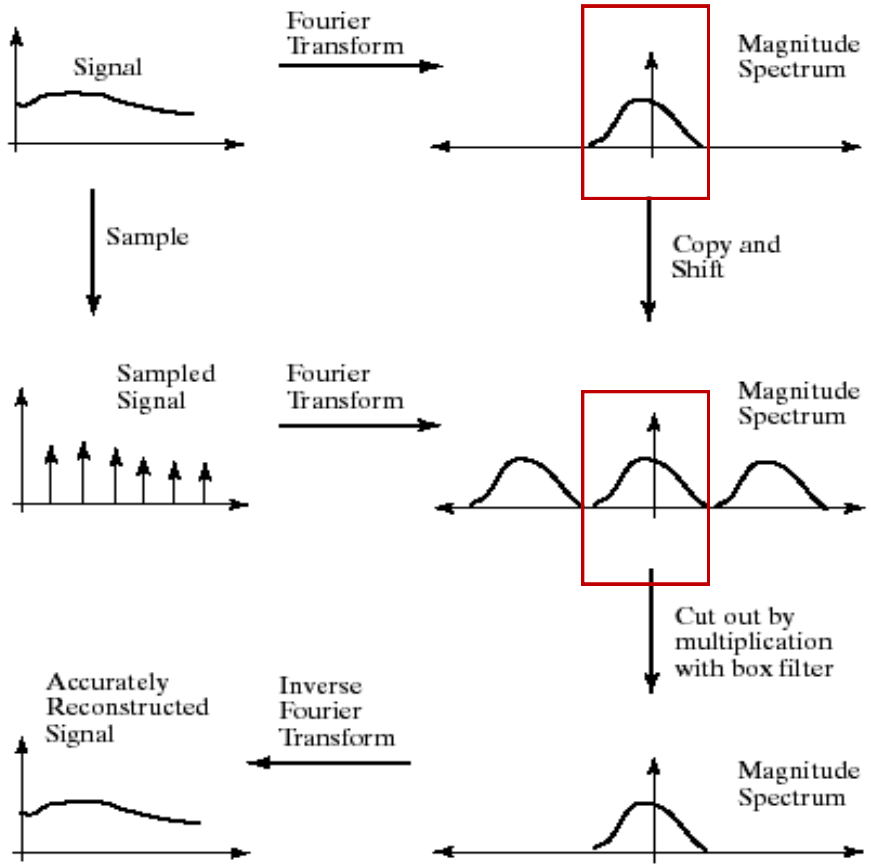
Dealing with a discrete time signals (or image) we can only represent frequencies between [-Fs/2 , Fs/2]
Where Fs is the sampling-frequency.

The frequency is normalized to $[-\pi, \pi]$ rad/s. Outside this frequency area, the specter repeats itself, i.e. the Discrete time fourier transform is periodic. WHY? - aliasing

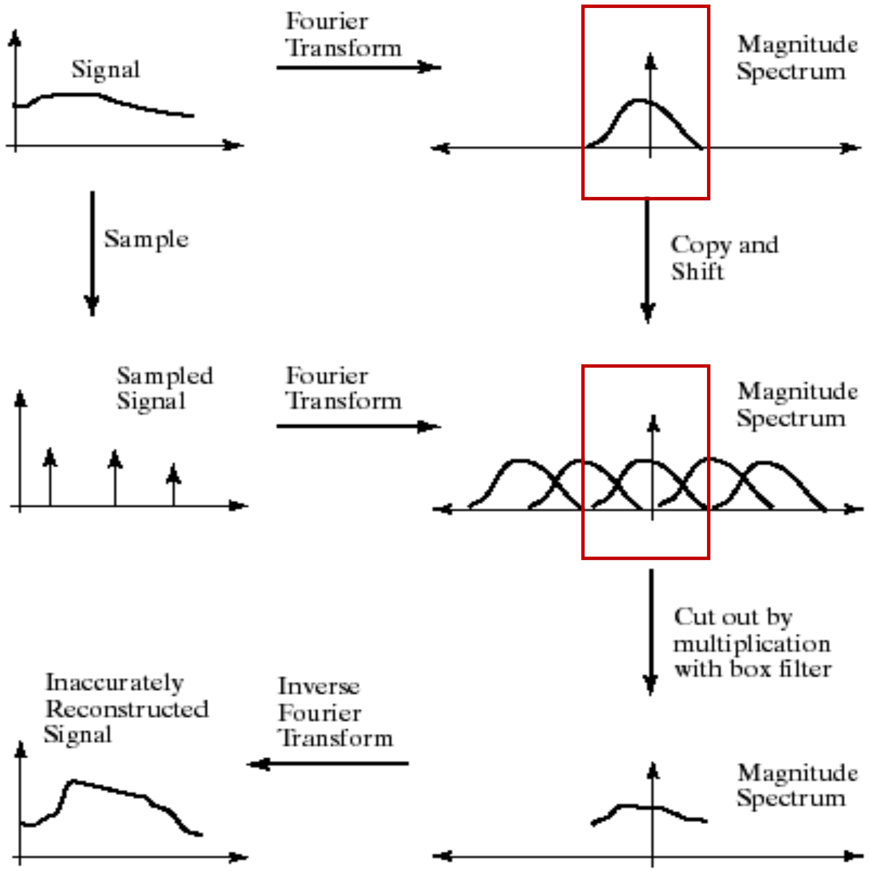
Aliasing – for discrete signals, multiple frequencies look the same.



Sampling without aliasing



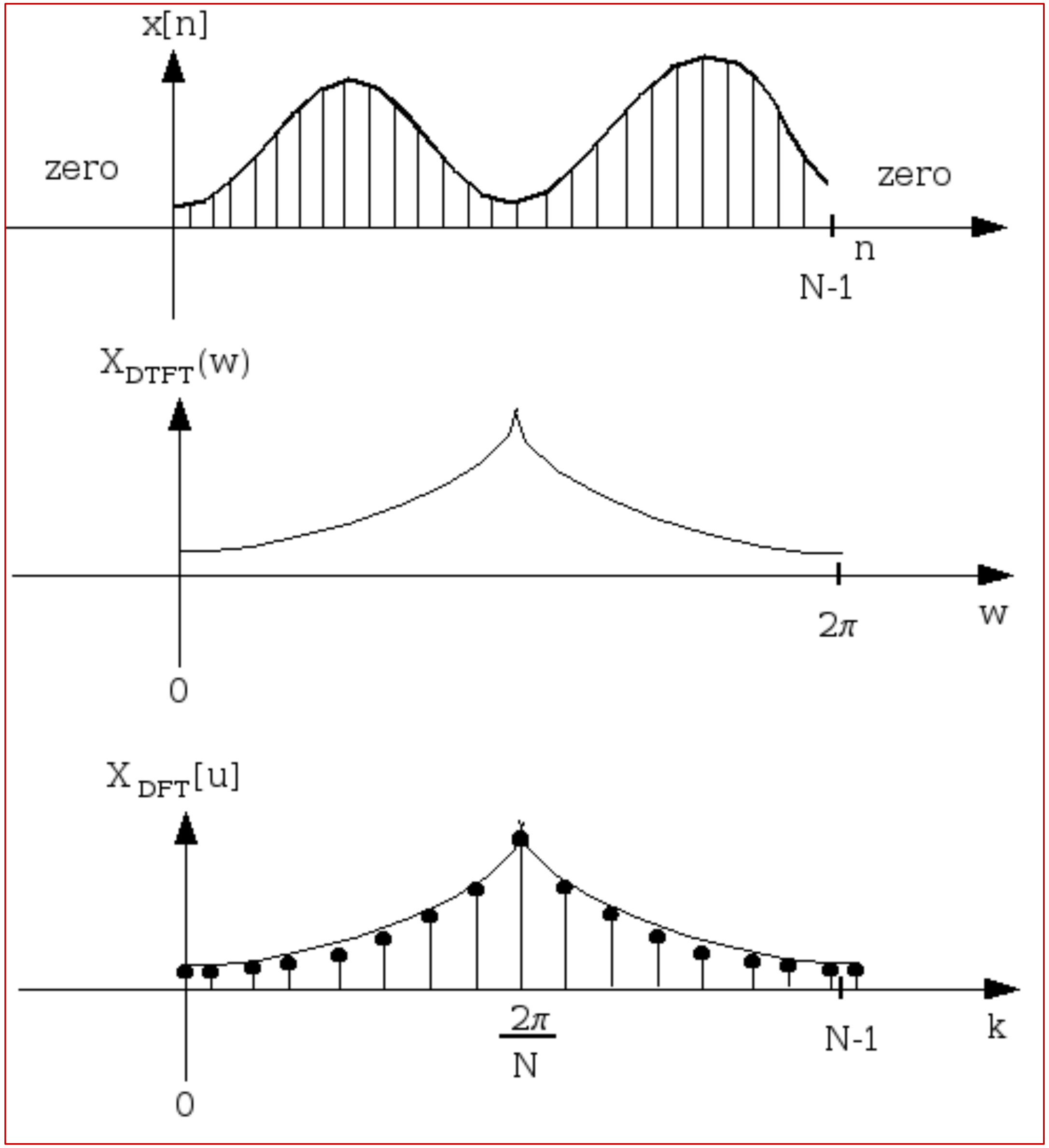
Sampling with aliasing



We can not represent the continous frequencies of DTFT in the computer.

What if we sample the DTFT?

This gives the Discrete Fourier Transform DFT



(6.2) Discrete Fourier Transform

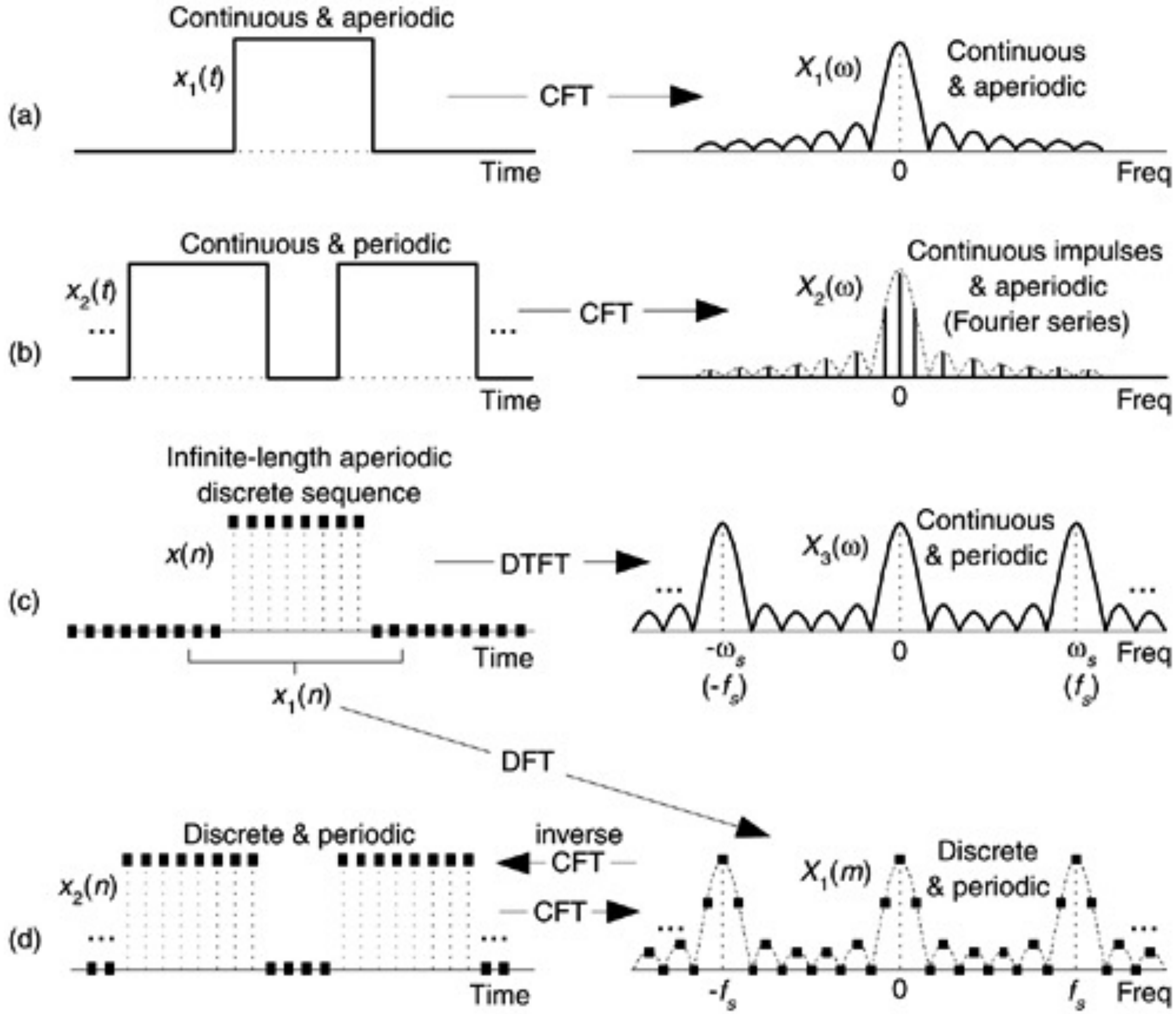
Not ω (omega)

Notation from the book. Let *g*(*x*) be a 1D discrete signal with *w* number of samples. The DFT and inverse DFT of *g* is defined (*x* and *k* are integers):

DFT: $$\boxed{G(k) = \mathcal{F}\{g(x)\} = \sum_{x=0}^{w-1} g(x) e^{-j2\pi kx/w}}$$

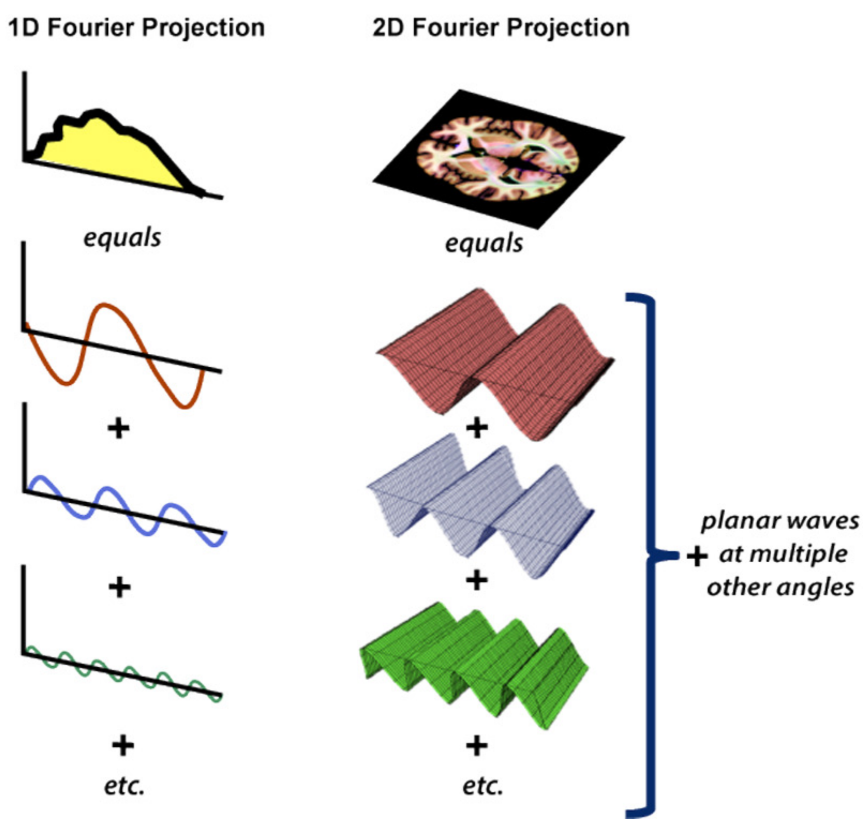
IDFT: $$\boxed{g(x) = \mathcal{F}^{-1}\{G(k)\} = \frac{1}{w}\sum_{k=0}^{w-1} G(k) e^{j2\pi kx/w}}$$

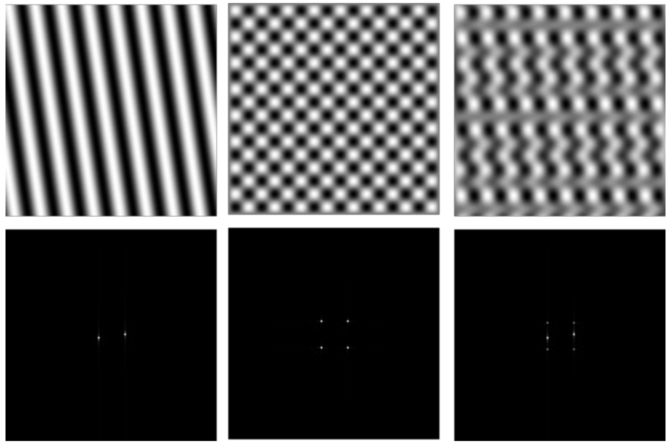
All modern implementations of the DFT use some variation of the FFT algorithm FFT – Fast Fourier Transform.

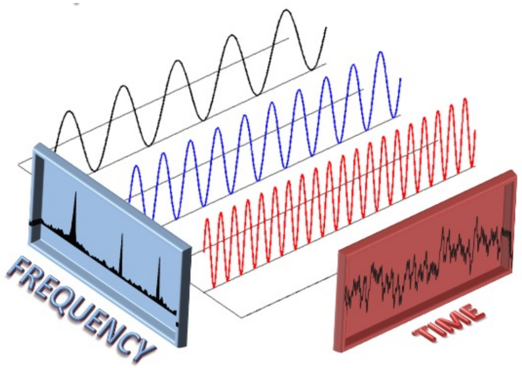

Continuous & aperiodic
$x_1(t)$
CFT
$X_1(\omega)$
Continuous & aperiodic
(a)
Time
0
Freq
Continuous & periodic
$x_2(t)$
CFT
$X_2(\omega)$
Continuous impulses & aperiodic (Fourier series)
(b)
Time
0
Freq
Infinite-length aperiodic discrete sequence
$x(n)$
DTFT
$X_3(\omega)$
Continuous & periodic
(c)
Time
$-\omega_s$ $(-f_s)$
0
ω_s (f_s)
Freq
$x_1(n)$
DFT
Discrete & periodic
inverse CFT
CFT
$x_2(n)$
$X_1(m)$
Discrete & periodic
(d)
Time
$-f_s$
0
f_s
Freq

Fourier series and transform - summary

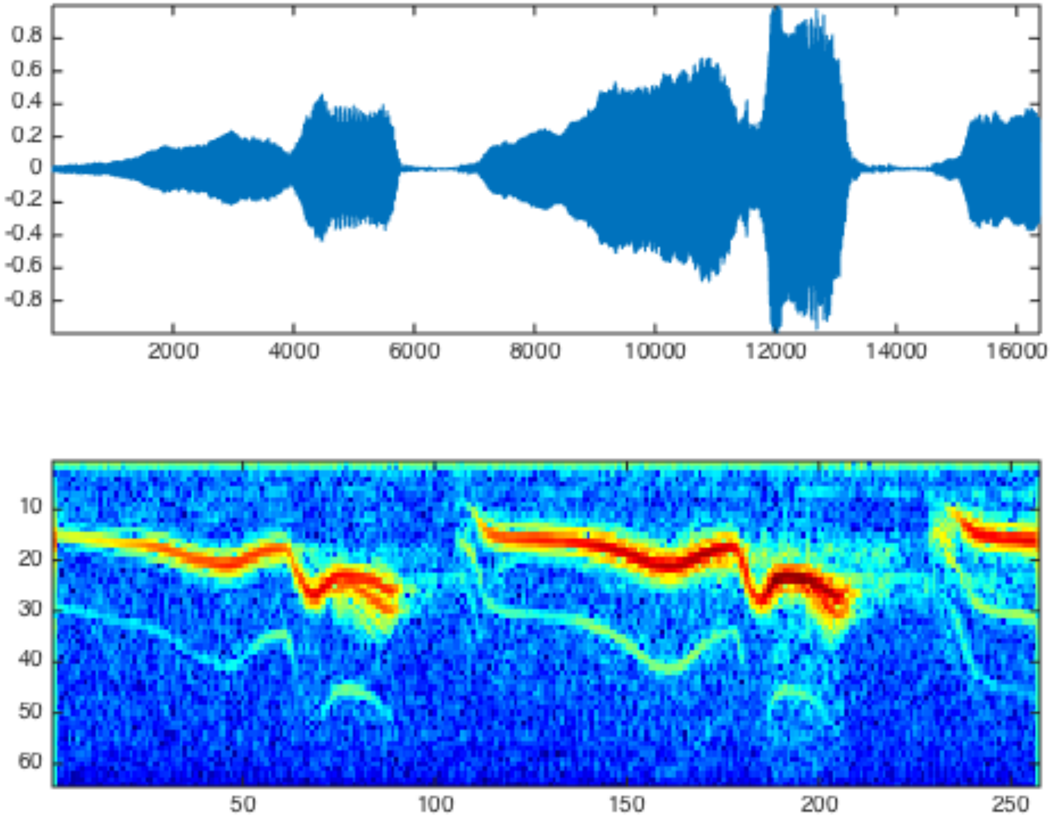
Fourier transform – visual examples





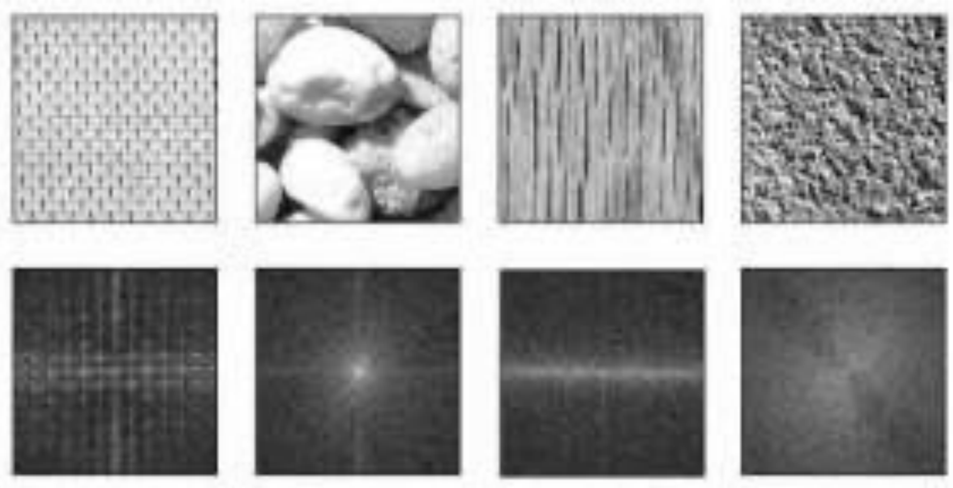


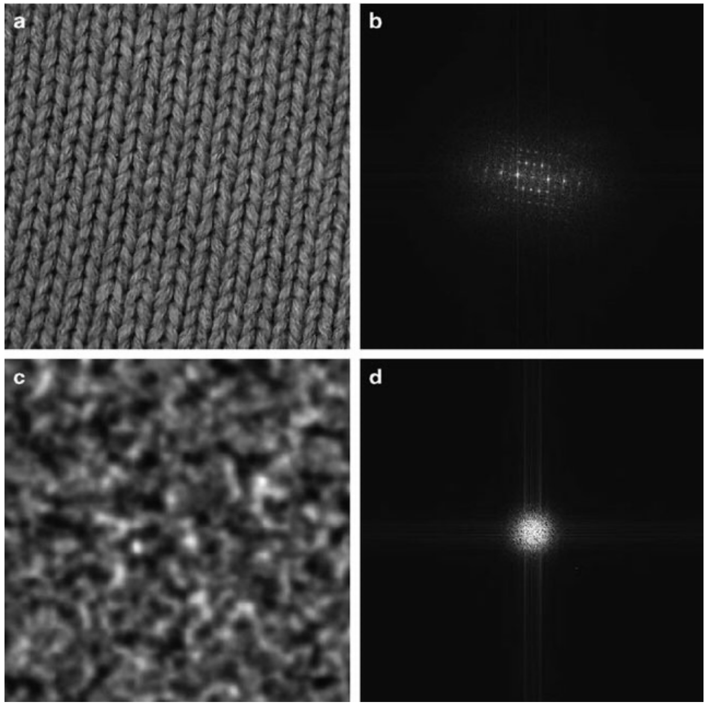
Short Time fourier transform



http://www.numerical-tours.com/matlab/audio_1_processing/

Textures in frequency (Fourier) domain





Display of DFT values

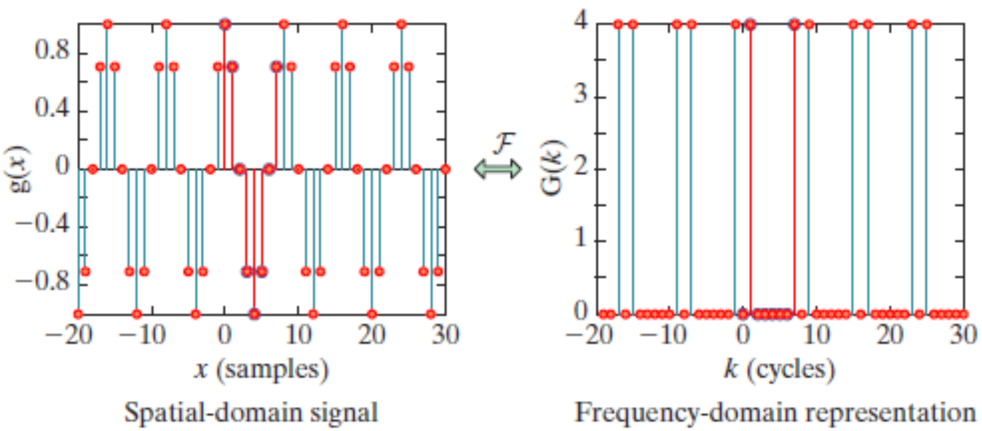
Some properties of the DFT

- The DFT of a real-valued, even-symmetric signal is also real-valued and even-symmetric.

- The DFT is **linear**. $\mathcal{F}\{ag(x) + bh(x)\} = a\mathcal{F}\{g(x)\} + b\mathcal{F}\{h(x)\}$

- The DFT is **periodic**. $g(x + nw) = g(x) \overset{DFT}{\Longleftrightarrow} G(k) = G(k + nw), \quad x, k, n, w \in \mathbb{Z}$

Figure 6.3 Periodicity of the DFT. The discrete signal consisting of eight samples $x = 0, \ldots, 7$ (red, left) gives rise to the DFT consisting of eight samples $k = 0, \ldots, 7$ (red, right). If the DFT is evaluated for other values of k, or if the inverse DFT of the DFT is evaluated for other values of x, the signal repeats with period $w = 8$.



- **Shift theorem**: computing the DFT of a shifted signal is the same as multiplying the DFT of the original, unshifted signal by an appropriate complex exponential.

$$g(x) \overset{DFT}{\Longleftrightarrow} G(k)$$
$$g(x - x_0) \overset{DFT}{\Longleftrightarrow} G(k)e^{-j2\pi k x_0/w}$$

- **Modulation**: states that multiplying a signal by a complex exponential causes a shift in the frequency domain:

$$g(x)e^{j2\pi k_0 x/w} \overset{DFT}{\Longleftrightarrow} G(k - k_0)$$
$$g(x)(-1)^x \overset{DFT}{\Longleftrightarrow} G\left(k - \frac{w}{2}\right)$$

- The **scaling property** says that if the signal is stretched in the spatial domain, then the Fourier transform is compressed in the frequency domain, and vice versa:

$$g(x) \overset{\mathcal{F}}{\Longleftrightarrow} G(k)$$
$$g(ax) \overset{\mathcal{F}}{\Longleftrightarrow} \frac{1}{a}G\left(\frac{k}{a}\right)$$

- **Parseval's theorem:** the energy is preserved in the frequency domain, where the energy is defined as the sum of the squares of the magnitudes of the elements:

$$\sum_{x=0}^{w-1} |g(x)|^2 = \sum_{k=0}^{w-1} |G(k)|^2$$

More DFT properties

- The **DC component** of the signal is captured by $G(0)$, which is the sum of the values in *g(x).*

- Circular convolution in the time (or spatial) domain is equivalent to multiplication in the frequency domain, and vice versa. If standard convolution is desired, the signals must be zero padded:

$$g_1(x) \circledast g_2(x) \quad \stackrel{DFT}{\Longleftrightarrow} \quad G_1(k)G_2(k)$$

$$g_1(x)g_2(x) \quad \stackrel{DFT}{\Longleftrightarrow} \quad \frac{1}{w}G_1(k) \circledast G_2(k)$$

- It is often convenient to convert the real and imaginary components of the Fourier transform into **polar coordinates**:

$$G(k) = G_{even}(k) + jG_{odd}(k) = |G(k)|e^{j\angle G(k)}$$

$$|G(k)| = \sqrt{G_{even}^2(k) + G_{odd}^2(k)}$$

$$\angle G(k) = \tan^{-1}\left(\frac{G_{odd}(k)}{G_{even}(k)}\right)$$

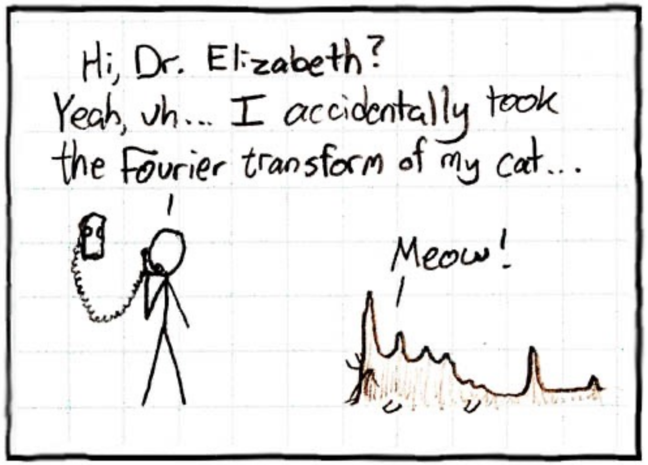
Introduction to Fourier Transform

Three points from the topic:

1. What is a Fourier series and a (discrete) Fourier transform?
   - ✓ Building blocks that can completely describe a signal (recipie) by complex sinusoids (cosx+jsinx)

2. How can that help us in understanding signals and images?
   - ✓ Effective representation, compression, analysis, fetaure extraction

3. Properties of Fourier Transform ?
   - ✓ Linear, periodic (DFT), energy is preserved …



- https://programmathically.com/the-fourier-transform-and-its-math-explained-from-scratch/
- https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/
- https://www.youtube.com/watch?v=Qm84XIoTy0s
- https://www.youtube.com/watch?v=iN0VG9N2q0U&t=19s