COS791: Fourier Transform

Dr Mardé Helbig

Image Formation

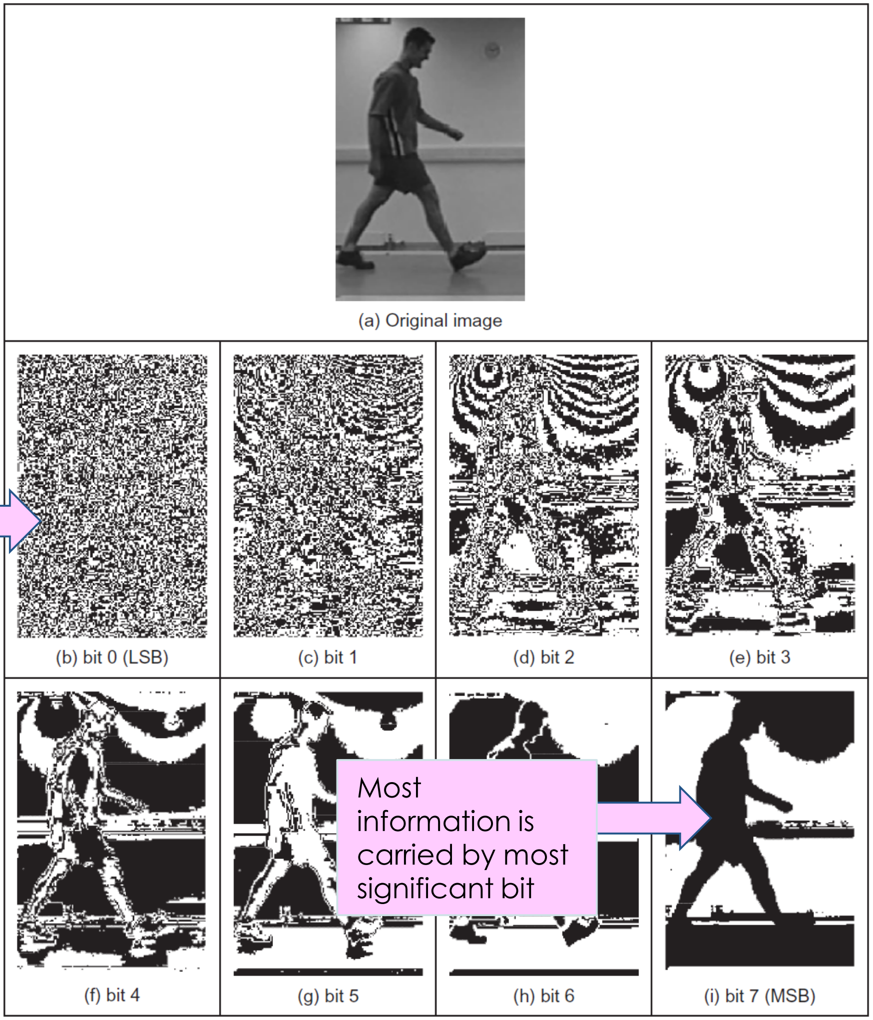
- A digital image is a matrix (2D array) of pixels
- Value of each pixel is proportional to the **brightness** of the corresponding point in the scene
- Value is derived from the output of an A/D converter
- Matrix of pixels is usually square
- Describes an image as ***NxN*** *m*-bit pixels where:
  - *N* is the number of points, and
  - *m* controls the number of brightness values
- Using *m* bits gives a range of 2^m values, ranging from 0 to $2^m - 1$
- If **m = 8** => brightness levels range between **0 and 255**
- Usually displayed as **black and white**, with **shades of gray** in-between

- Smaller values of *m* give fewer available levels reducing the contrast in an image
- The ideal value of *m* is actually related to the signal-to-noise ratio (dynamic range) of the camera
- Choosing 8-bit is convenient – can store it in a byte

As the order of bits increases, change less rapidly and carry more information

Carries the least information and changes rapidly

The fact that there is a walker in the original image can be recognized much better from the high order bits



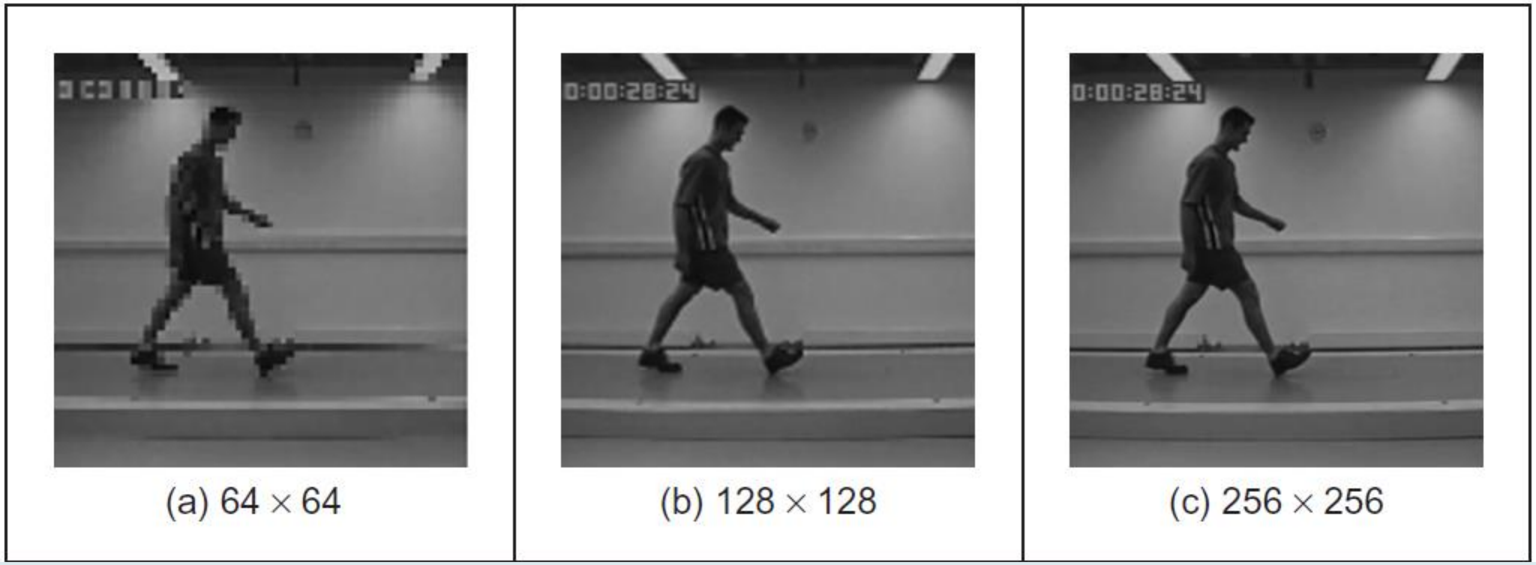
(a) Original image

(b) bit 0 (LSB) (c) bit 1 (d) bit 2 (e) bit 3

(f) bit 4 (g) bit 5 (h) bit 6 (i) bit 7 (MSB)

Image Formation

- Colour images have multiple intensity components:
  - RGB – 3 components => red, green blue
  - Use 8 bits per colour => 24 bits, 24-bit true colour
- Choosing *N* is difficult:
  - Too low values lead to blocky lines and lost detail
  - Large values provide more detail, but require more space and take longer to process



(a) 64 × 64 (b) 128 × 128 (c) 256 × 256

Effects of different image resolution

Image Formation

is there any rule to choose N?

- Yes, the sampling frequency is dictated by the sampling criterion
- For this, we need to understand how signals are interpreted in the frequency domain
- Need to understand the **Fourier transform**

Fourier Transform

- A way of mapping a signal into its component frequencies
- Frequency is measured in Hertz (Hz)
- The rate of repetition with time is measured in seconds (s)
- Time is the reciprocal of frequency and vice versa (Hertz=1/s; s=1/Hz)
- By knowing the frequencies, you can use them

Fourier Transform

- Imagine that you have a smoothie
- Given the smoothie, the Fourier Transform will find the recipe for the smoothie
- **How?** By running the smoothie through filters to extract each ingredient
- **Why?** Recipes are easier to analyse, compare and modify
- **Can we get the smoothie back?** Yes, by blending the ingredients

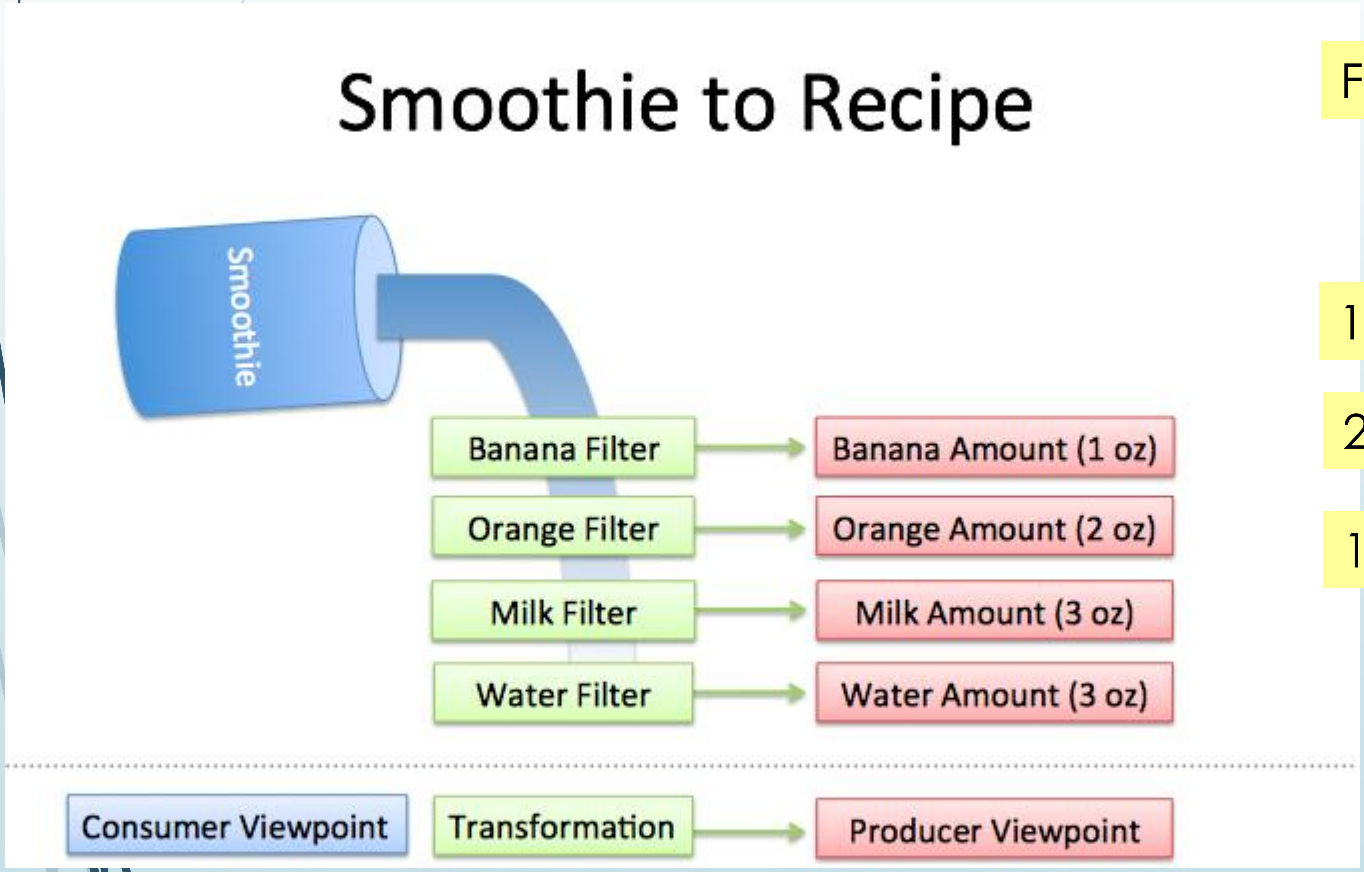
Fourier Transform

- Imagine that you have a smoothie time-based pattern
- Given the smoothie, the Fourier Transform will find the recipe for the smoothie overall cycle recipe
- **How?** By running the smoothie through filters to extract each ingredient amplitude, offset and rotation speed
- **Why?** Recipes are easier to analyse, compare and modify
- **Can we get the smoothie back?** Yes, by blending the ingredients

Fourier Transform

- Changes our perspective from consumer (what do I have?) to producer (what is it made of?)
- So given a smoothie, what is the recipe?
- But, how can we find the recipe?

Fourier Transform



Filters must be…

1. Independent
2. Complete
1. Combine-able

Fourier Transform

- Joseph Fourier said:

  What if any signal could be filtered into a bunch of circular paths?

Fourier transform:

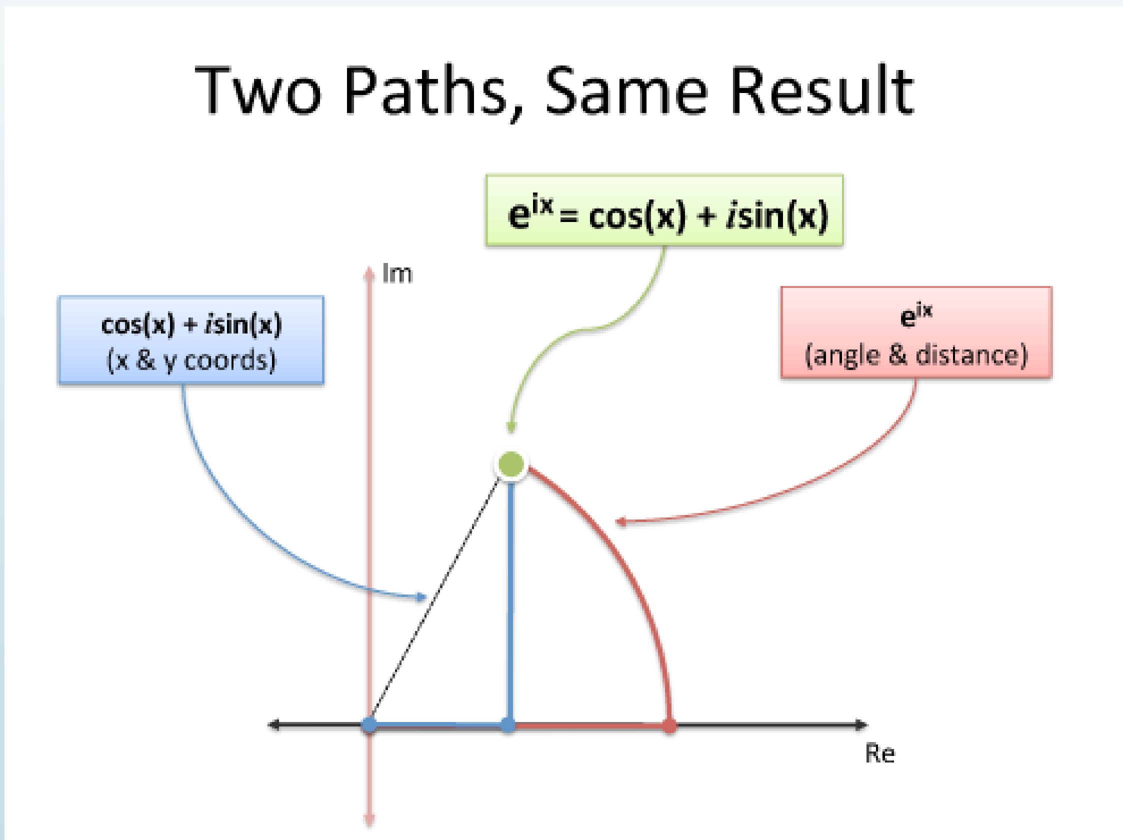
- Starts with a time-based signal
- Applies filters to measure each possible "circular ingredient"
- Collects the full recipe, listing the amount of each "circular ingredient“

Examples:

- Earthquake vibrations
- Sound waves
- Computer data
- Radio waves

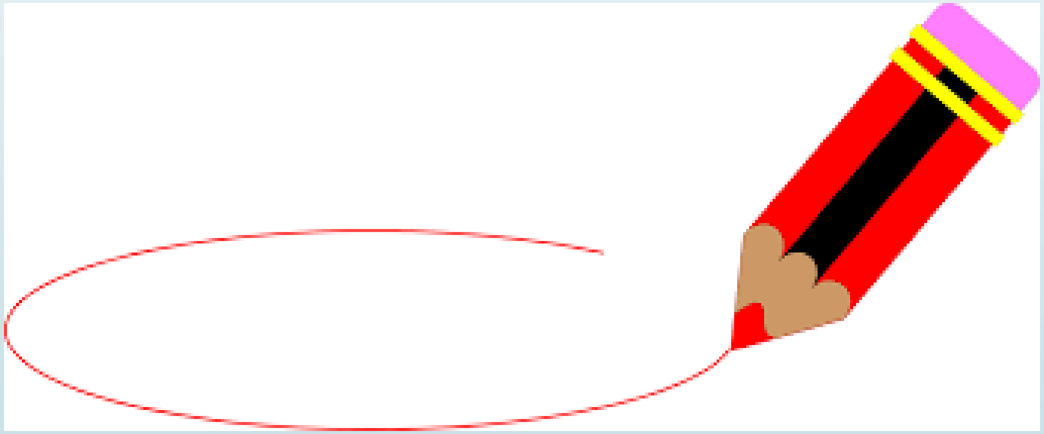
Fourier Transform

- Fourier transform is about **circular paths**
- Can use **Euler's formula**:



Fourier Transform

- Imaging we are speaking over the phone
- I want us to draw a circle simultaneously
- What do we need to define?



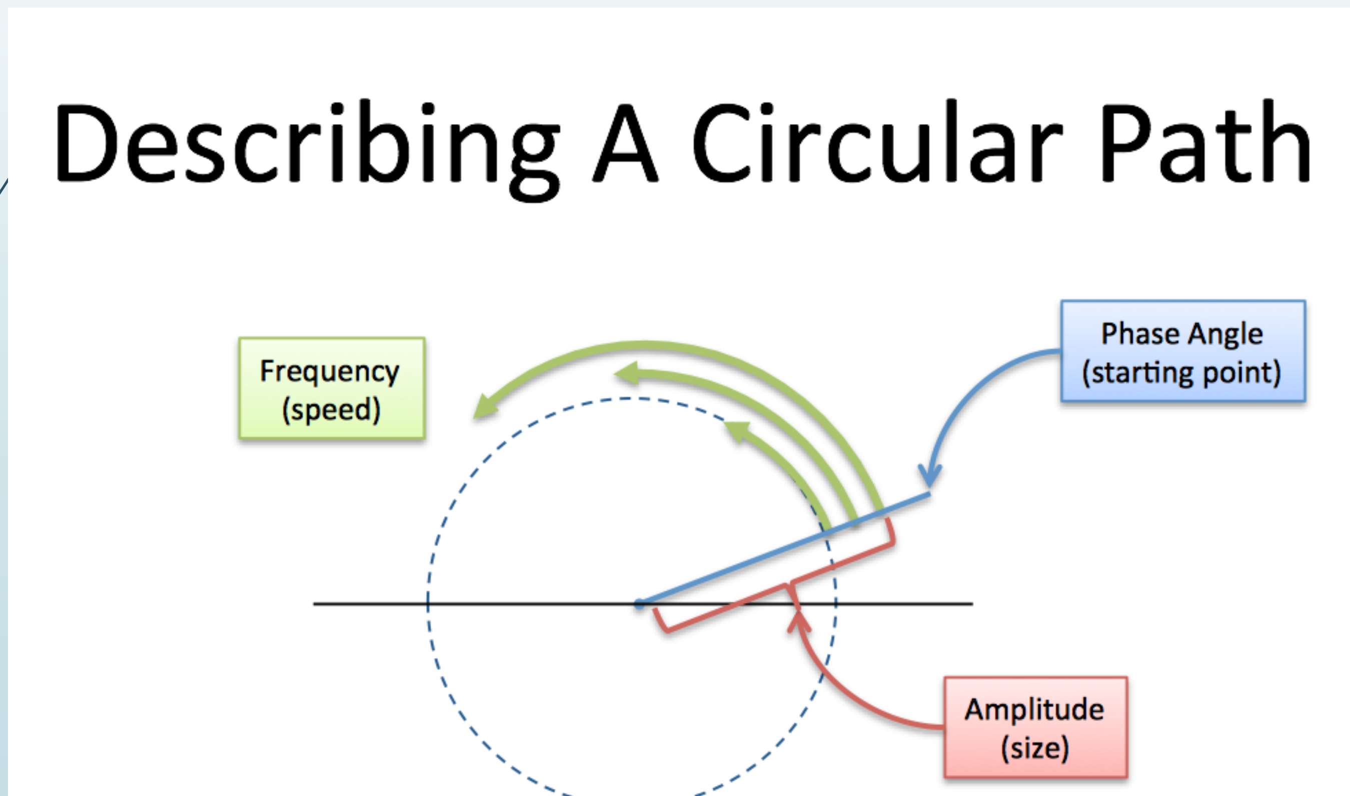
Fourier Transform

- Imaging we are speaking over the phone
- I want us to draw a circle simultaneously
- What do we need to define?

1. How big is the circle? (**amplitude**, i.e. size of radius)
2. How fast should we draw it? (**frequency**, where 1 circle/s is a frequency of 1 Herz (Hz)
3. Where do we start? (**phase angle**, where 0 degrees is the x-axis)

Fourier Transform

- Imaging we are speaking over the phone
- I want us to draw a circle simultaneously
- What do we need to define?



Fourier Transform

- We can also **combine paths**
- Imagine small cars driving in circles at different speeds
- The combined position of all paths or cycles are the signal (just like the combined flavours are our smoothie)

Simulation of a basic circular path:
https://betterexplained.com/examples/fourier/?cycles=0,1

Adding another cycle:
https://betterexplained.com/examples/fourier/?cycles=0,1,1

Changing the phase:
https://betterexplained.com/examples/fourier/?cycles=0,1:45

Fourier Transform

- The Fourier Transform finds the set of cycle speeds, amplitudes and phases to match any time signal

Simulation of a basic circular path:
https://betterexplained.com/examples/fourier/?cycles=0,1

Adding another cycle:
https://betterexplained.com/examples/fourier/?cycles=0,1,1

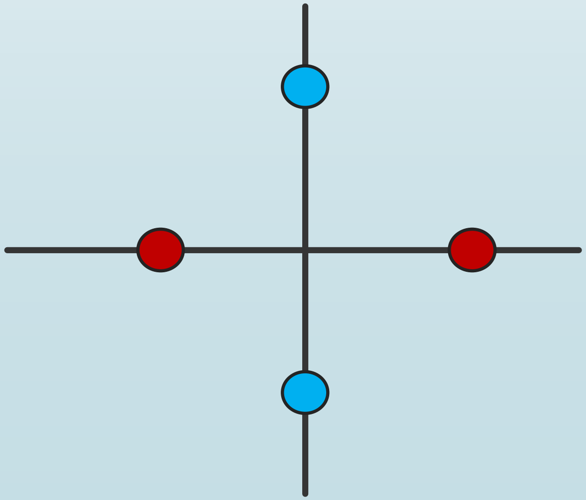
Changing the phase:
https://betterexplained.com/examples/fourier/?cycles=0,1:45

Fourier Transform

Can we make a spike in time, for example a spike at the timepoint (4 0 0 0) using cycles?

Yes, if:

- At time t=0, each cycle ingredient reaches its max
- Can be made from 4 cycles (0Hz 1Hz 2Hz 3Hz), each with a *magnitude of 1* and a *phase of 0*
- And at t=1, 2, 3 the cycles must cancel out to give 0

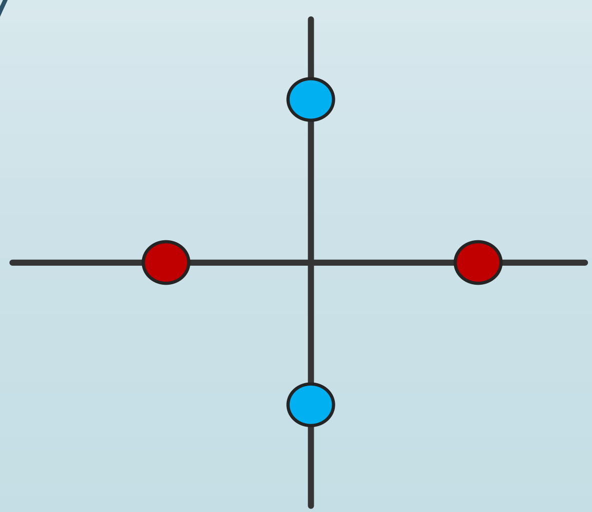


Fourier Transform

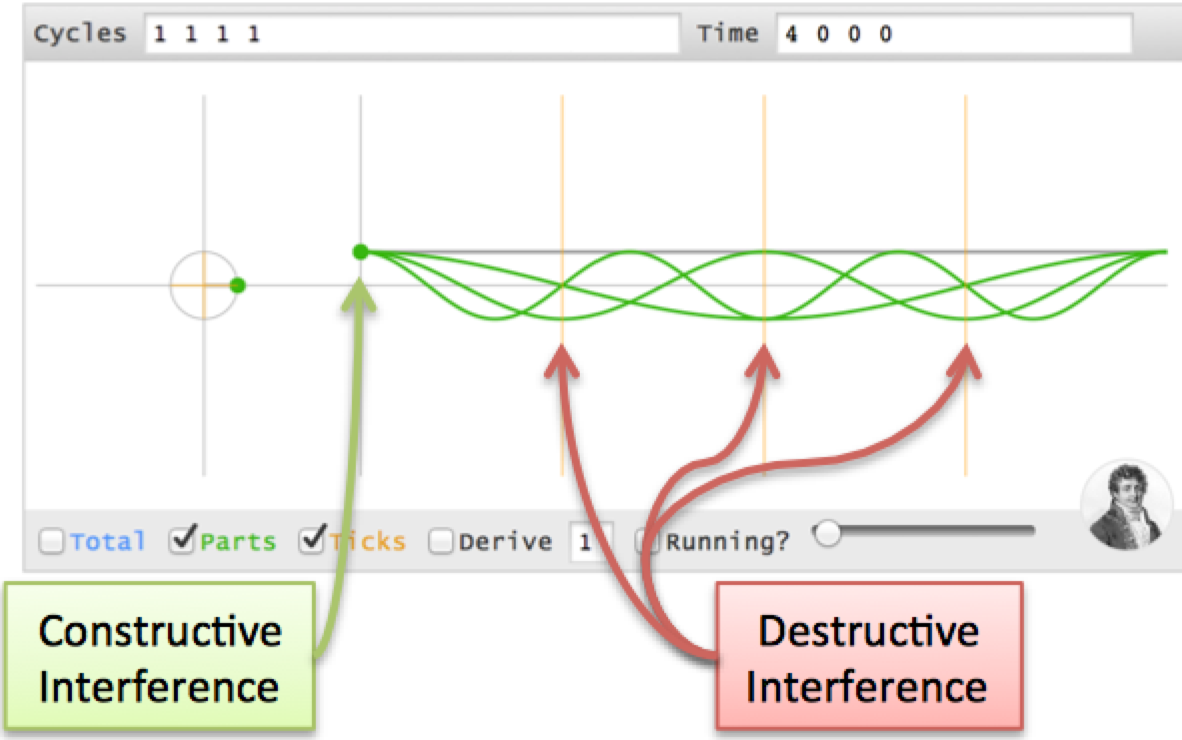
Can we make a spike in time, for example a spike at the timepoint (4 0 0 0) using cycles?

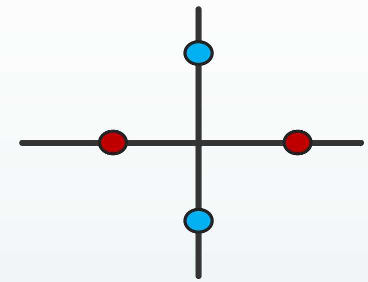
Yes, if:

- At time t=0, each cycle ingredient reaches its max
- Can be made from 4 cycles (0Hz 1Hz 2Hz 3Hz), each with a *magnitude of 1* and a *phase of 0*
- And at t=1, 2, 3 the cycles must cancel out to give 0



```
Time 0 1 2 3
------------
0Hz: 0 0 0 0
1Hz: 0 1 2 3
2Hz: 0 2 0 2
3Hz: 0 3 2 1
```



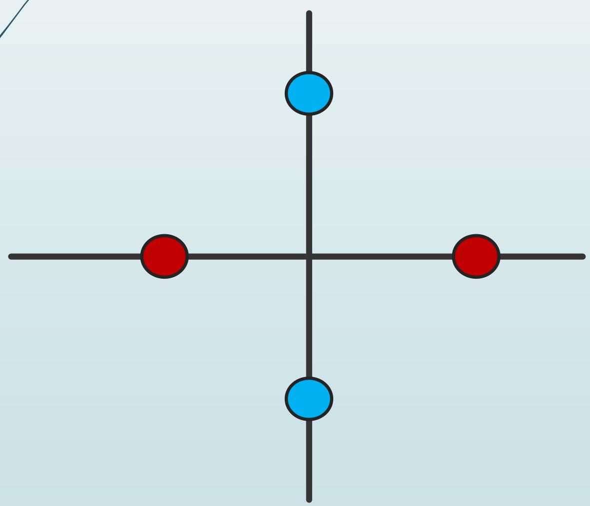


```
Time 0 1 2 3
------------
0Hz: 0 0 0 0
1Hz: 0 1 2 3
2Hz: 0 2 0 2
3Hz: 0 3 2 1
```

Fourier Transform

Now what if we move the spike to (0 4 0 0)?

How will you approach this?



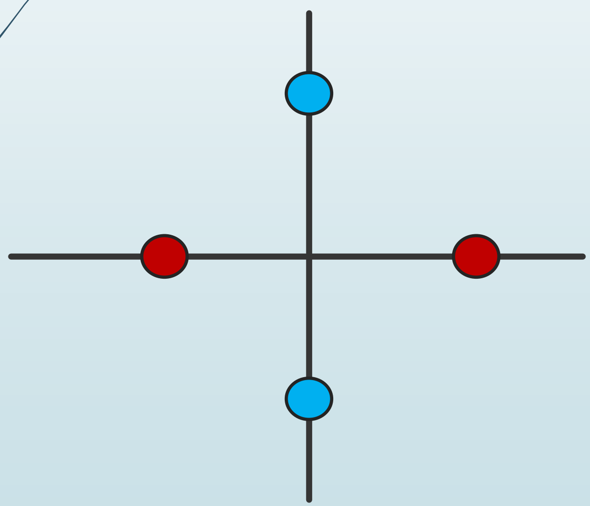
The (4 0 0 0) is a race where all runners line up at starting point…

But what happens with (0 4 0 0)? Why is this different?

Fourier Transform

Now what if we move the spike to (0 4 0 0)?

How will you approach this?



Imagine that you want to line up all the runners to finish at the same time…

The runners are: a granny, you and Usain Bolt

Fourier Transform

Now what if we move the spike to (0 4 0 0)?

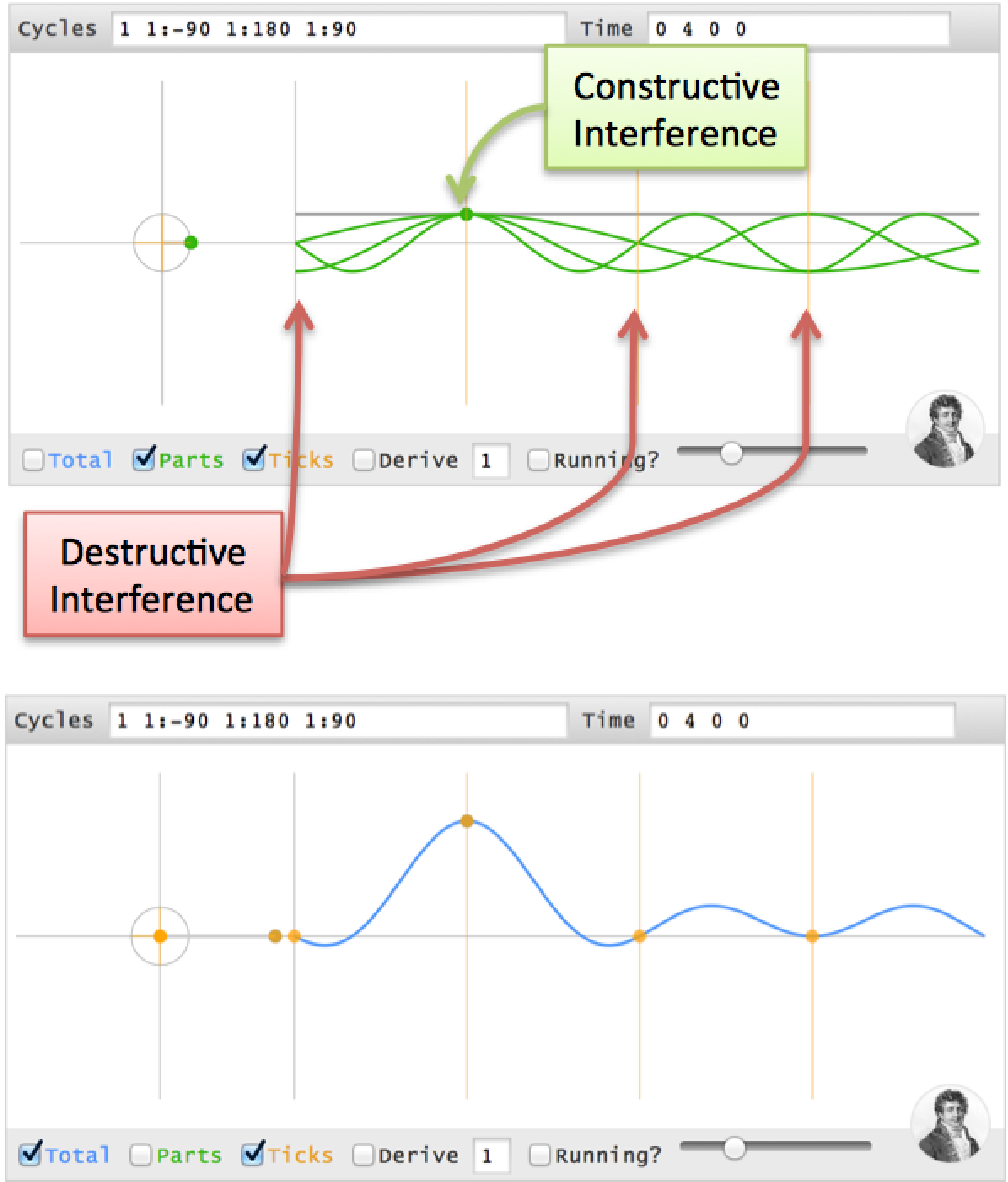
How will you approach this?

Imagine that you want to line up all the runners to finish at the same time…

The runners are: a granny, you and Usain Bolt

Phase shifts (starting angles) are **time delays** in the cycle universe…

So how will you delay each cycle with 1s by adjusting the starting point? I.e. each cycle must reach phase 0 at t=1 to create (0 4 0 0)



Cycles 1 1:-90 1:180 1:90
Time 0 4 0 0
Constructive Interference
Total
Parts
Ticks
Derive 1
Running?
Destructive Interference
Cycles 1 1:-90 1:180 1:90
Time 0 4 0 0
Total
Parts
Ticks
Derive 1
Running?

Fourier Transform

So, can you now make (0 0 4 0), i.e. a 2 second delay?

Fourier Transform

- The Fourier transform builds the recipe frequency by frequency
- Separate full signal into different time spikes for the Discrete Fourier Transform: (a 0 0 0) (0 b 0 0) (0 0 c 0) (0 0 0 d)
- Then loop through each frequency to find full Fourier Transform

$$\underbrace{X_k}_{\text{Frequency recipe =}} = \overbrace{\sum_{n=0}^{N-1}}^{\text{Add up}} \overbrace{x_n \cdot e^{-i2\pi kn/N}}^{\text{Contributions to this frequency, from each time spike}}$$

$$\underbrace{x_n}_{\text{Time point =}} = \frac{1}{N}\underbrace{\sum_{k=0}^{N-1}}_{\text{Add up}} \underbrace{X_k \cdot e^{i2\pi kn/N}}_{\text{Contributions to this time point, from each frequency}}$$

N: # time samples
n: time we are currently considering
x_n: value of signal at time *n*
k: current frequency (0 to N-1 Hz)
X_K: amount of frequency *k* in signal (amplitude and phase – complex nr)
2πk: speed in radians

Fourier Transform

- Tells us which frequencies make up a time-domain signal
- The **magnitude** of the transform at a particular frequency is the **amount of that frequency** in the original signal
- If we **add together** sinusoidal signals in amounts specified by the Fourier transform, we should obtain the **originally transformed signal**

Fourier Transform

Notation in textbook

Play video

$$Fp(\omega) = \Im(p(t)) = \int_{-\infty}^{\infty} p(t)\,e^{-j\omega t}\,dt \qquad (2.1)$$

where:

$Fp(\omega)$ is the Fourier transform, and $\Im$ denotes the Fourier transform process;
ω is the **angular** frequency, $\omega = 2\pi f$, measured in **radians/s** (where the frequency f is the reciprocal of time t, $f = 1/t$);
j is the complex variable, $j = \sqrt{-1}$
$p(t)$ is a **continuous** signal (varying continuously with time); and
$e^{-j\omega t} = \cos(\omega t) - j\sin(\omega t)$ gives the frequency components in $p(t)$.

Fourier Transform – simple example

A pulse *p* with amplitude *A* between t= -T/2 and t=T/2, and zero everywhere else:

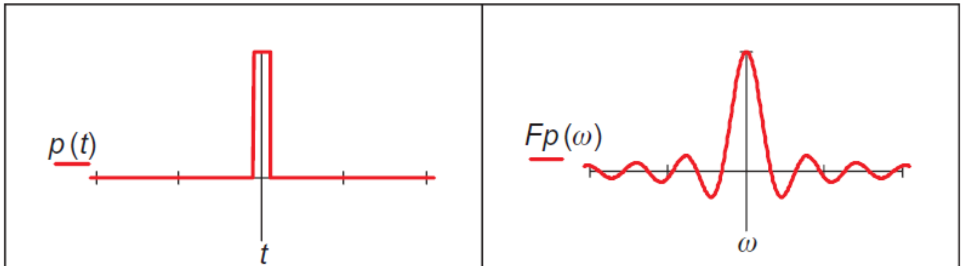
$$p(t) = \begin{vmatrix} A & \text{if} & -T/2 \le t \le T/2 \\ 0 & \text{otherwise} \end{vmatrix}$$

Therefore, the Fourier transform of the pulse is:

$$Fp(\omega) = \int_{-T/2}^{T/2} A\, e^{-j\omega t}\, dt \quad = -\frac{A\, e^{-j\omega T/2} - A\, e^{j\omega T/2}}{j\omega}$$

using the relation $\sin(\theta) = (e^{j\theta} - e^{-j\theta})/2j$:

$$Fp(\omega) = \begin{vmatrix} \frac{2A}{\omega}\sin\left(\frac{\omega T}{2}\right) & \text{if} & \omega \neq 0 \\ AT & \text{if} & \omega = 0 \end{vmatrix}$$



(a) Pulse of amplitude *A* = 1

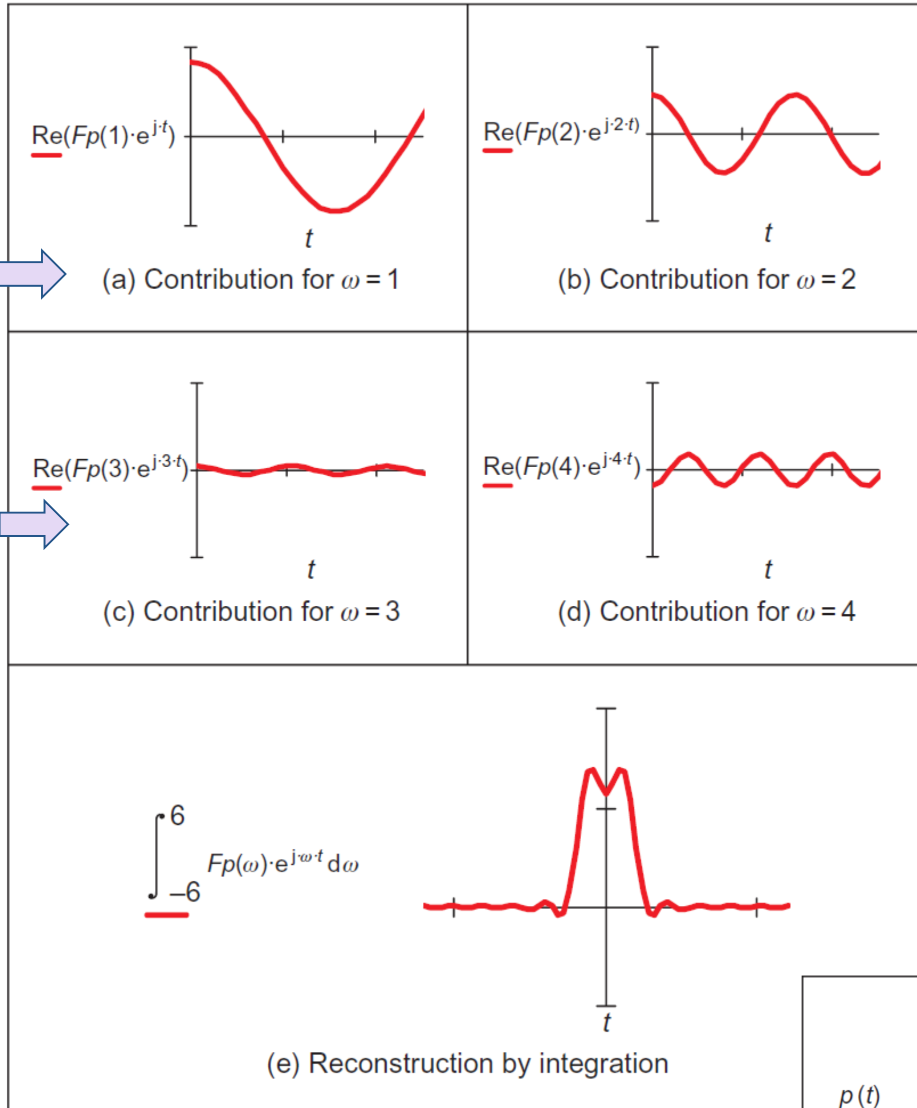
(b) Fourier transform

Only plot real part, not complex part

Contributes large part of signal

Contributes small part of signal

Why?



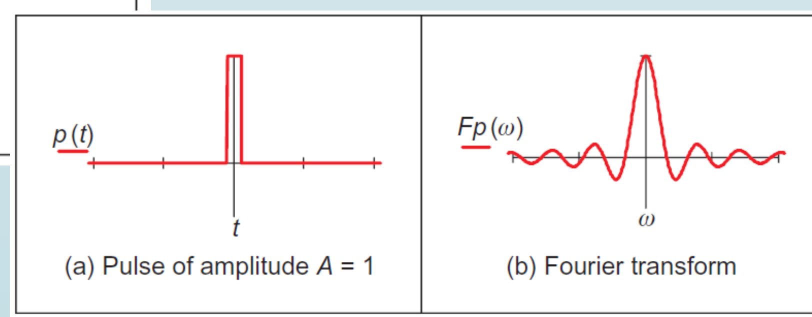
(a) Contribution for $\omega = 1$

(b) Contribution for $\omega = 2$

(c) Contribution for $\omega = 3$

(d) Contribution for $\omega = 4$

(e) Reconstruction by integration



(a) Pulse of amplitude $A = 1$

(b) Fourier transform

Fourier Transform

- Result of the Fourier transform is a complex number
- It is usually represented in terms of its **magnitude** (or size or modulus) and **phase** (or argument)
- The transform can be represented as:

$$\int_{-\infty}^{\infty} p(t)\,\mathrm{e}^{-\mathrm{j}\omega t}\,\mathrm{d}t = \underbrace{\mathrm{Re}[Fp(\omega)]}_{\text{Real part}} + \mathrm{j}\,\underbrace{\mathrm{Im}[Fp(\omega)]}_{\text{Imaginary part}}$$

- The magnitude of the transform:

$$\left|\int_{-\infty}^{\infty} p(t)\,\mathrm{e}^{-\mathrm{j}\omega t}\,\mathrm{d}t\right| = \sqrt{\mathrm{Re}[Fp(\omega)]^2 + \mathrm{Im}[Fp(\omega)]^2}$$

Amount of each frequency component

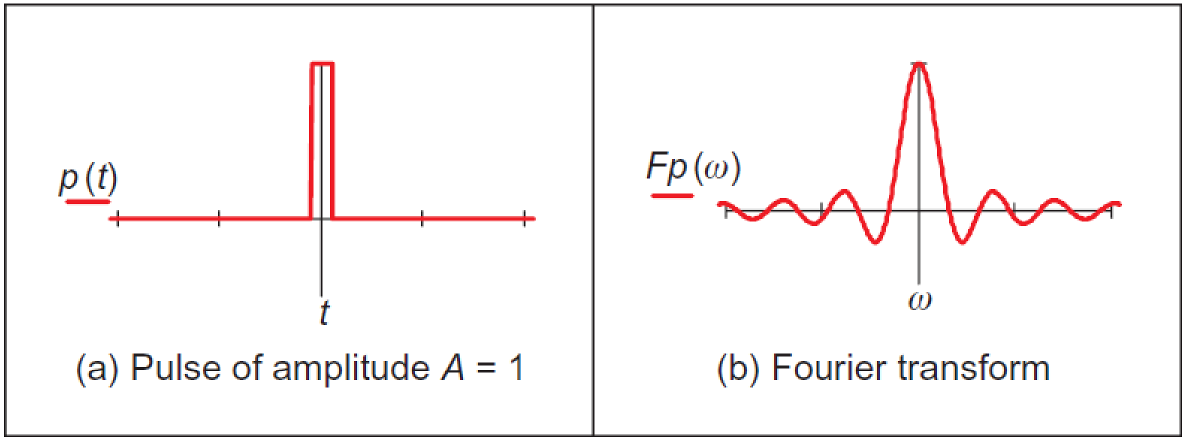
- The phase of the transform:

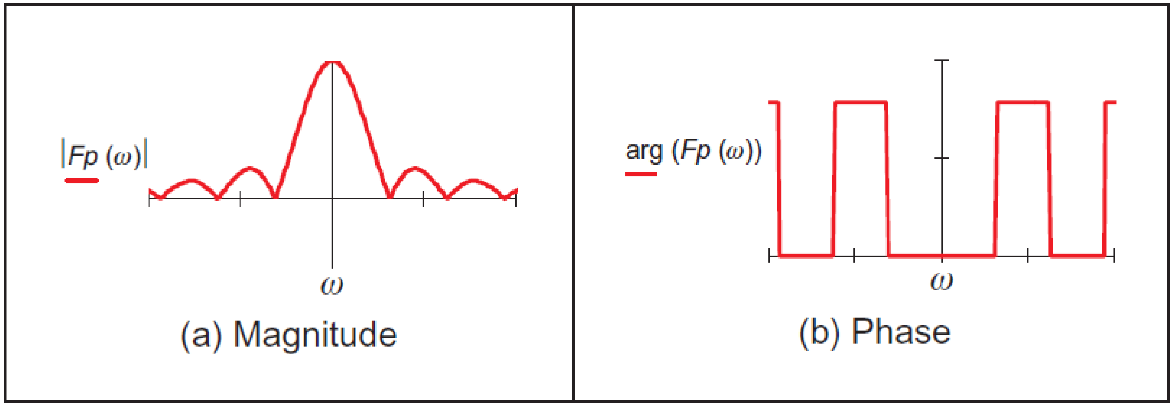
angle

$$\left\langle \int_{-\infty}^{\infty} p(t)\,\mathrm{e}^{-\mathrm{j}\omega t}\,\mathrm{d}t = \tan^{-1}\frac{\mathrm{Im}[Fp(\omega)]}{\mathrm{Re}[Fp(\omega)]}\right.$$

Timing when frequency components occur

Fourier Transform





Fourier Transform - Convolution

- The convolution of one signal $p_1(t)$ with another signal $p_2(t)$:

Memory

$$p_1(t) * p_2(t) = \int_{-\infty}^{\infty} p_1(\tau)\, p_2(t-\tau)\, \mathrm{d}\tau$$

- Basis of systems theory where the output of a system is the convolution of:
  - a stimulus, p_1, and
  - a system's response, p_2
- By inverting the time axis of the system response, to give $p_2(t\text{-}\tau)$, we obtain a memory function
- The convolution process then sums the effect of a stimulus multiplied by the memory function
- The output of the system is the cumulative response to a stimulus

Fourier Transform - Convolution

- Taking the Fourier transform:

$$\Im[p_1(t) * p_2(t)] = \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} p_1(\tau) p_2(t-\tau) \mathrm{d}\tau \right\} \mathrm{e}^{-\mathrm{j}\omega t} \, \mathrm{d}t$$
$$= \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} p_2(t-\tau) \mathrm{e}^{-\mathrm{j}\omega t} \, \mathrm{d}t \right\} p_1(\tau) \mathrm{d}\tau$$

But:

$$\Im[p_2(t-\tau)] = \mathrm{e}^{-\mathrm{j}\omega\tau} \, Fp_2(\omega)$$

Therefore:

$$\Im[p_1(t) * p_2(t)] = \int_{-\infty}^{\infty} Fp_2(\omega) p_1(\tau) \mathrm{e}^{-j\omega\tau} \, \mathrm{d}\tau$$
$$= Fp_2(\omega) \int_{-\infty}^{\infty} p_1(\tau) \mathrm{e}^{-j\omega\tau} \, \mathrm{d}\tau$$
$$= Fp_2(\omega) \times Fp_1(\omega)$$

Frequency domain is **multiplication**

Fourier Transform - Correlation

- Correlation gives a measure of the match between 2 signals:

$$p_1(t) \otimes p_2(t) = \int_{-\infty}^{\infty} p_1(\tau)\, p_2(t+\tau)\mathrm{d}\tau$$

- When $p_2(w) = p_1(w)$, correlating a signal with itself => *autocorrelation*
- Will use correlation later to find “things” in images

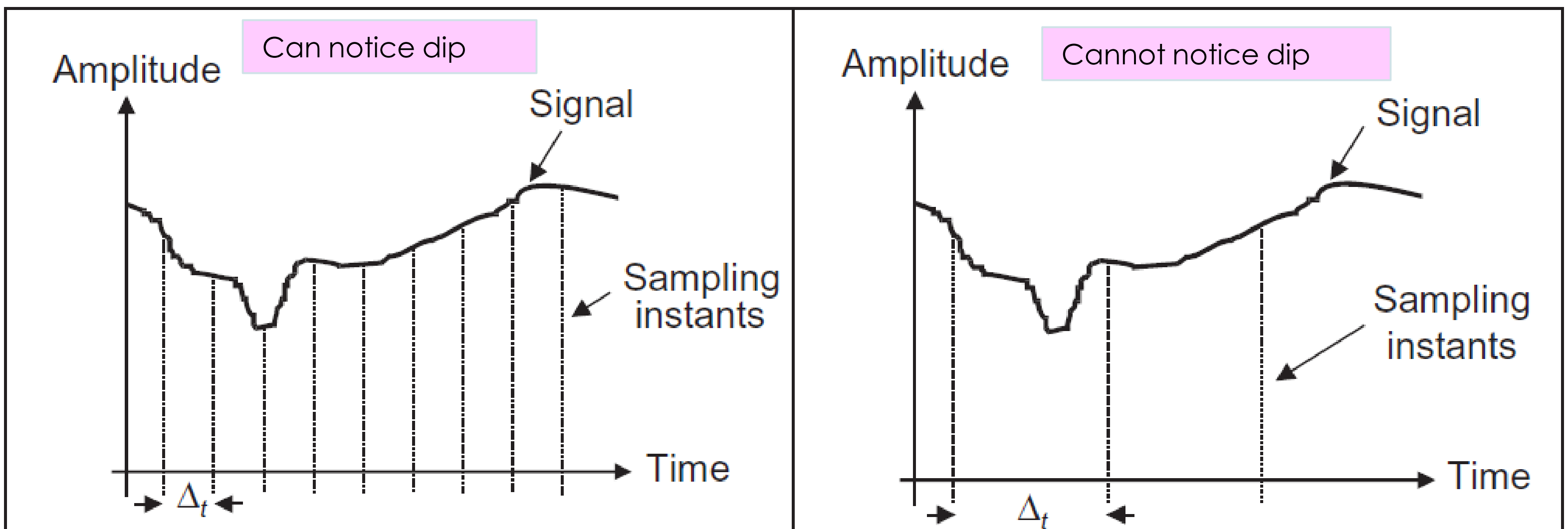
Fourier Transform – delta function

Delta is a function that occurs within a specific time interval:

$$\text{delta}(t-\tau) = \begin{cases} 1 & \text{if} \quad t = \tau \\ 0 & \text{otherwise} \end{cases}$$

Sampling Criterion

- Condition for correct choice of sampling frequency
- Sampling concerns taking instantaneous values of a continuous signal



(a) Sampling at high frequency

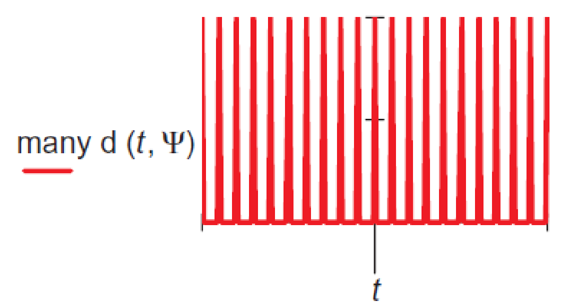
(b) Sampling at low frequency

Spacing between samples is low

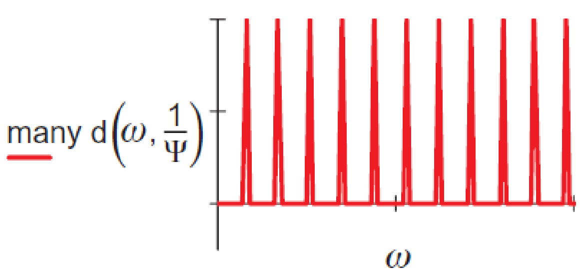
Spacing between samples is high

Sampling Criterion

- Working with transform pair:
- Sampling signal is the result of multiplying the time-variant signal by the sequence of spikes
- The frequency domain analog of this sampling process is to **convolve** the spectrum of the time-variant signal with the spectrum of the sampling function
- Take the spectrum of one, flip it along the horizontal axis and then slide it across the other

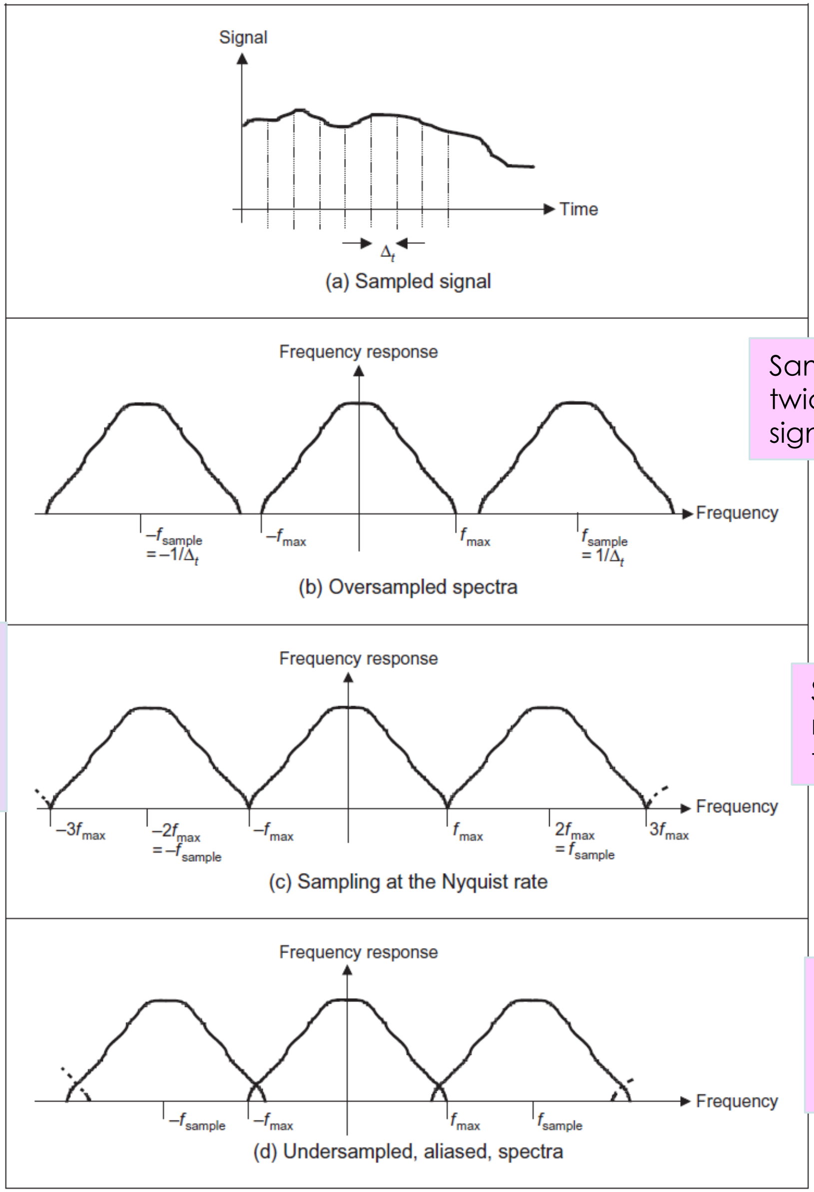


(g) Sampling function in time domain



(h) Transform of sampling function

Original signal is repeated every 1/delta_t Hz



Sampling frequency is more than twice the maximum frequency of signal - spaces

Nyquist sampling criterion:
In order to reconstruct a signal from its samples, the sampling frequency must be at least twice the highest frequency of the sampled signal

Sampling frequency is twice the maximum frequency of signal - touch

Inverse Fourier will produce wrong results

Results in aliasing effects

Ruin information

Sample spacing is large, time-variant signal's spectrum is replicated close together and spectra overlap (interfere)

Discrete Fourier Transform – 2D

- We need to generate Fourier transforms of images => need a 2D DFT
- This is a transform of pixels (sampled picture points) with a 2D spatial location indexed by coordinates x and *y*
- Two dimensions of frequency, *u* and *v*, which are the horizontal and vertical spatial frequencies, respectively

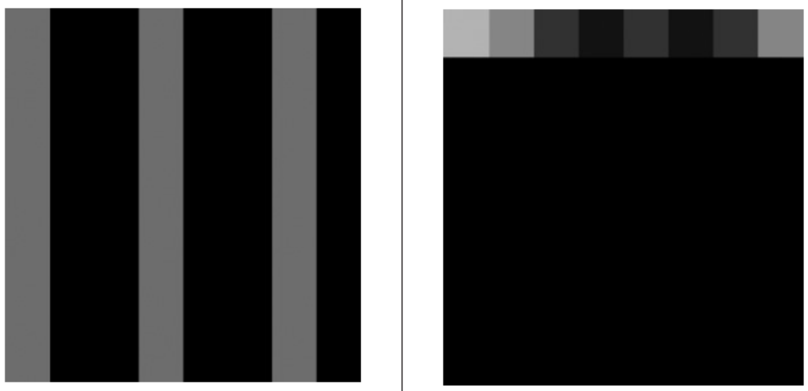
Discrete Fourier Transform – 2D

- 2D DFT:

$$\mathbf{FP}_{u,v} = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\mathbf{P}_{x,y}\, e^{-j\left(\frac{2\pi}{N}\right)(ux+vy)}$$

2D inverse DFT:

$$\mathbf{P}_{x,y} = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\mathbf{FP}_{u,v}\, e^{j\left(\frac{2\pi}{N}\right)(ux+vy)}$$



(a) Image of vertical bars

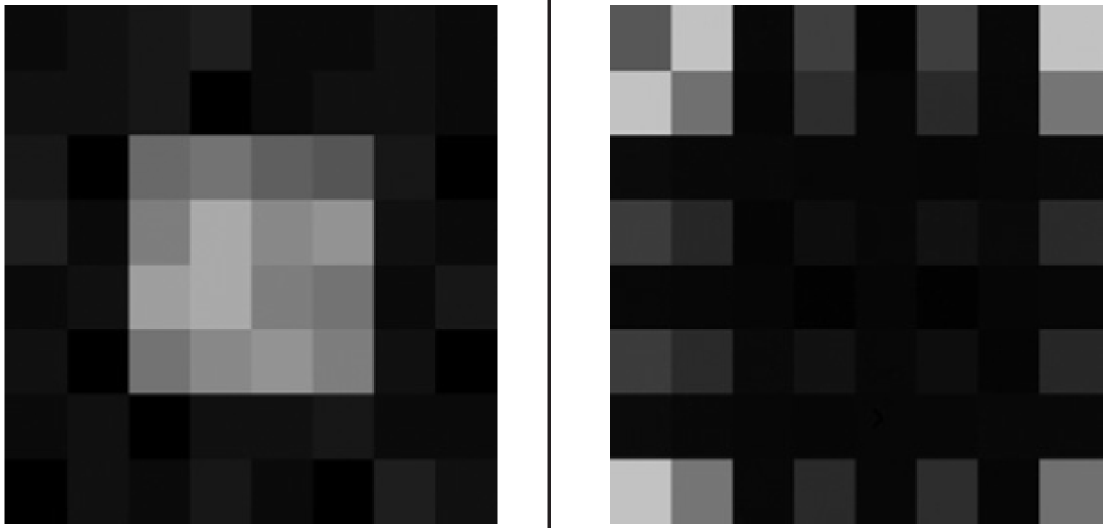
(b) Fourier transform of bars

Only horizontal spatial frequencies

Discrete Fourier Transform – 2D

- One difficulty is that the nature of the Fourier transform produces an image which is **difficult to interpret**
- The Fourier transform of an image gives the **frequency components**
- The position of each component reflects its frequency:
  - **low-frequency** components are **near the origin** and
  - **high frequency** components **are further away**
- The **lowest frequency component for zero frequency**, the d.c. component, **represents the average value** of the samples
- Unfortunately, the arrangement of the 2D Fourier transform **places the low-frequency components at the corners** of the transform

Discrete Fourier Transform – 2D

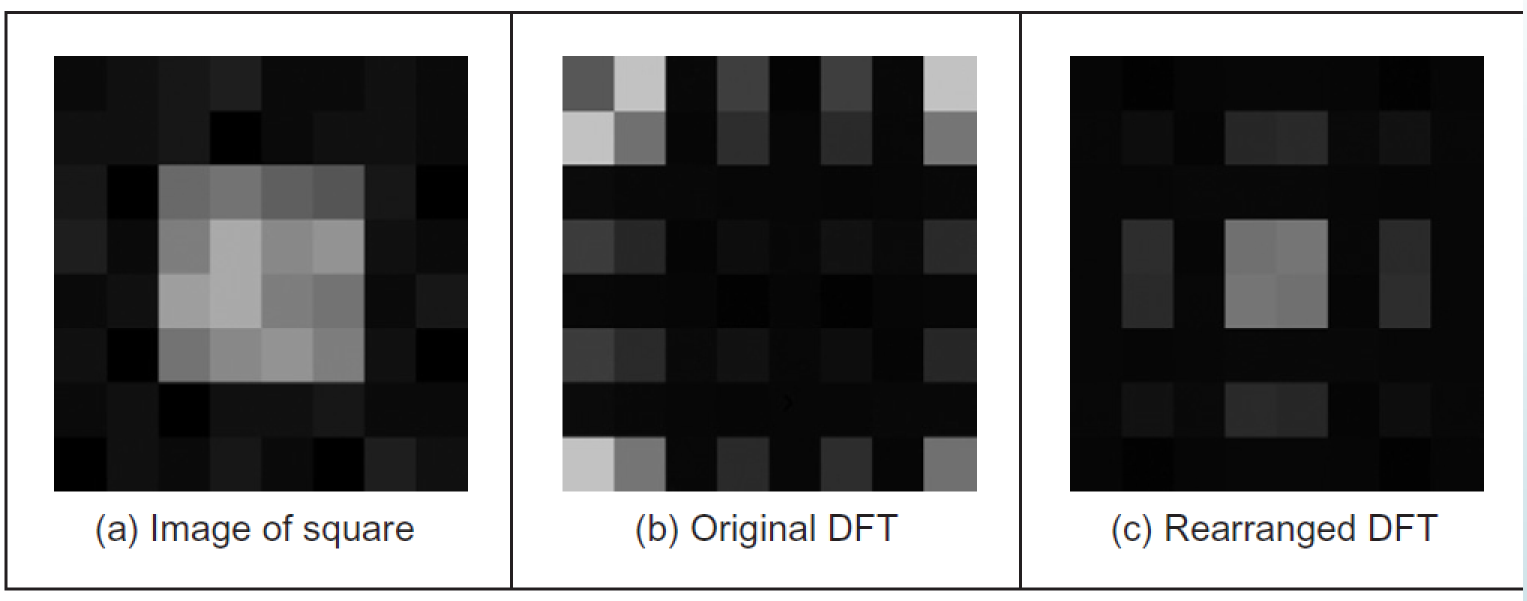


(a) Image of square (b) Original DFT

Discrete Fourier Transform – 2D

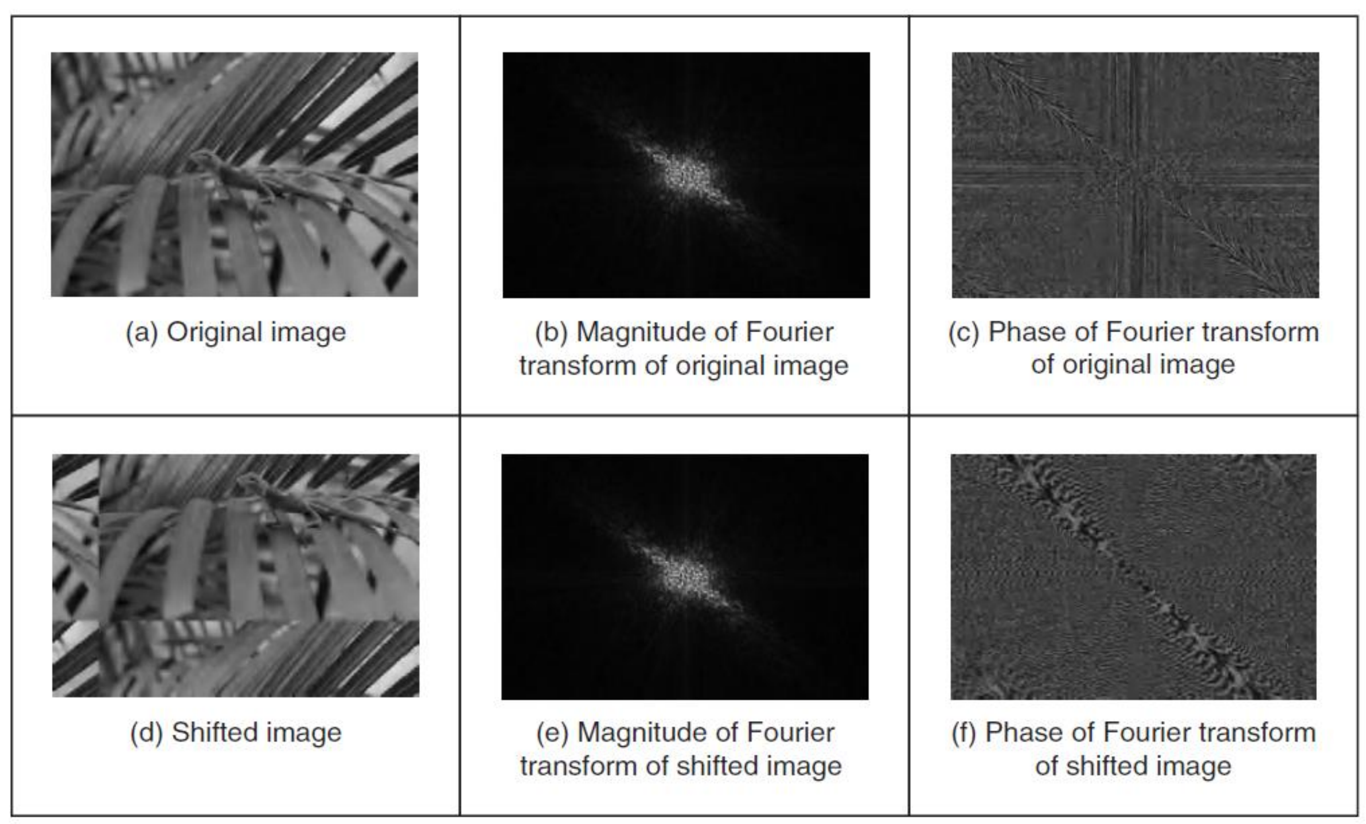
- A spatial transform is **easier to visualize if the d.c.** (zero frequency) component **is in the center**
- With **frequency increasing toward the edge** of the image
- **Reorder** the original image to give a transform which shifts the transform to the center
- This **improves visualization** and does not change any of the frequency domain information => only the way it is displayed

Discrete Fourier Transform – 2D



DFT - Properties

- **Shift invariance**: Shifting an image with a certain amount will not affect the magnitude



(a) Original image

(b) Magnitude of Fourier transform of original image

(c) Phase of Fourier transform of original image

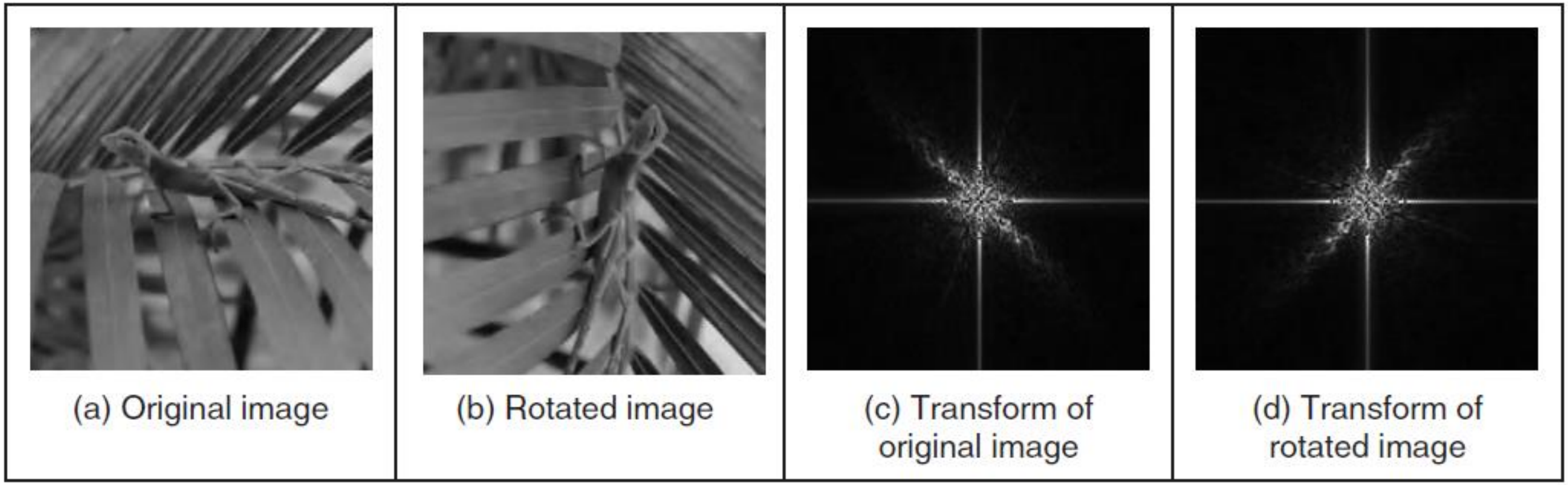
(d) Shifted image

(e) Magnitude of Fourier transform of shifted image

(f) Phase of Fourier transform of shifted image

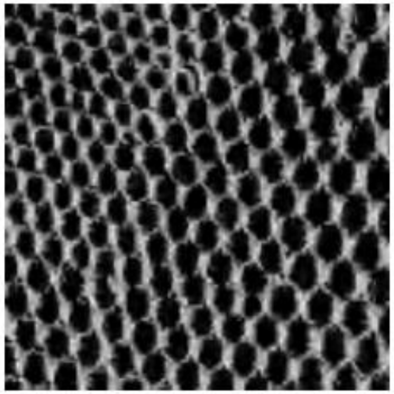
DFT - Properties

- **Rotation:** If an image rotates, the Fourier transform rotates

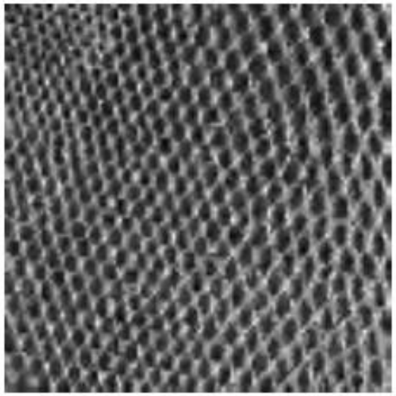


DFT - Properties

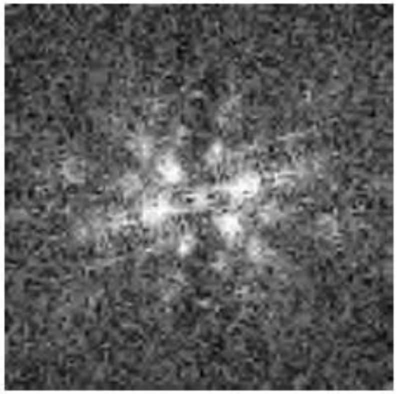
- **Scaling**: If we scale an image, the frequency components will be altered



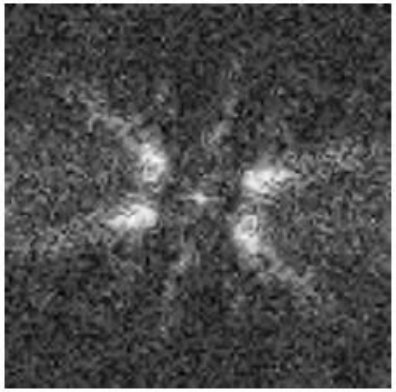
(a) Texture image



(b) Scaled texture image



(c) Transform of original texture

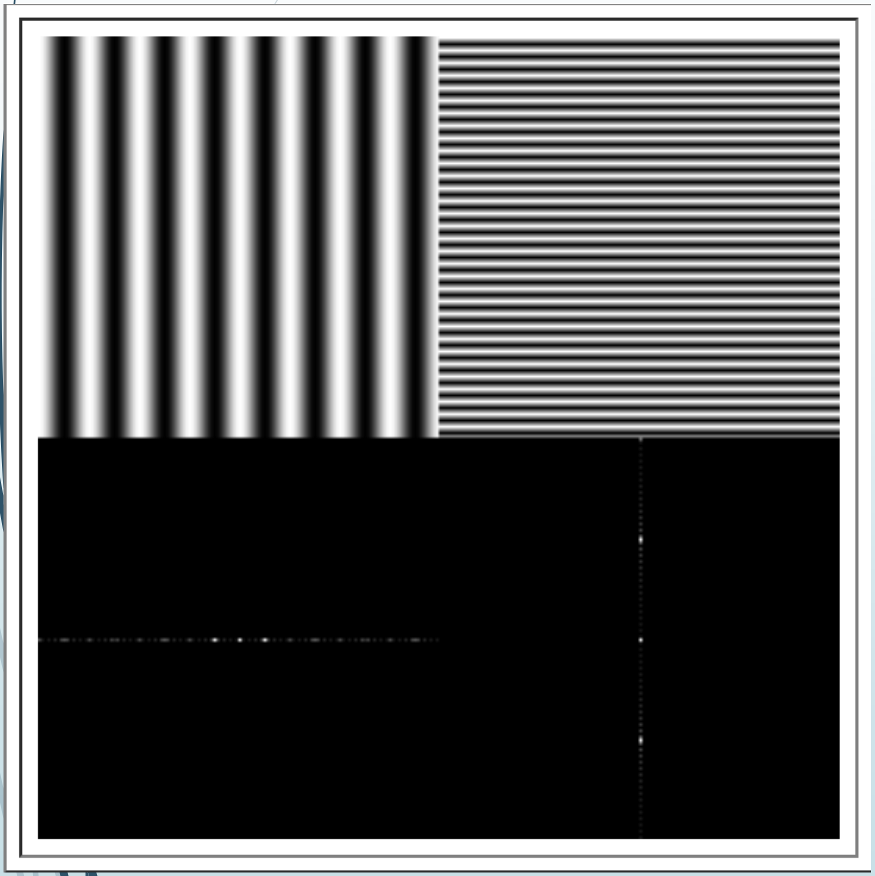


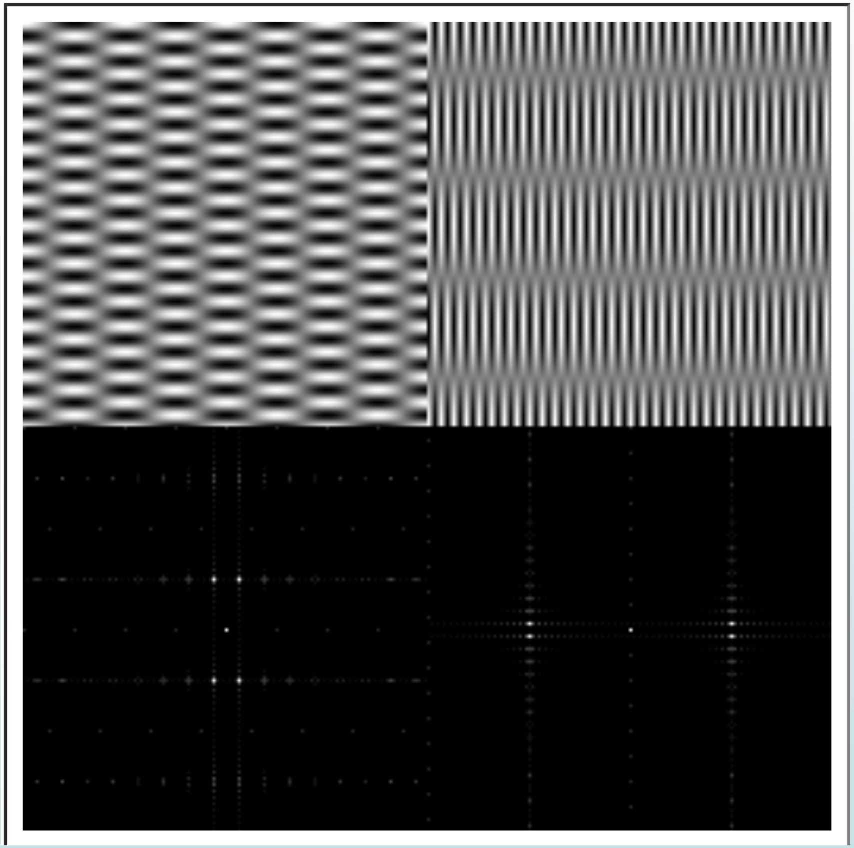
(d) Transform of scaled texture

DFT - Properties

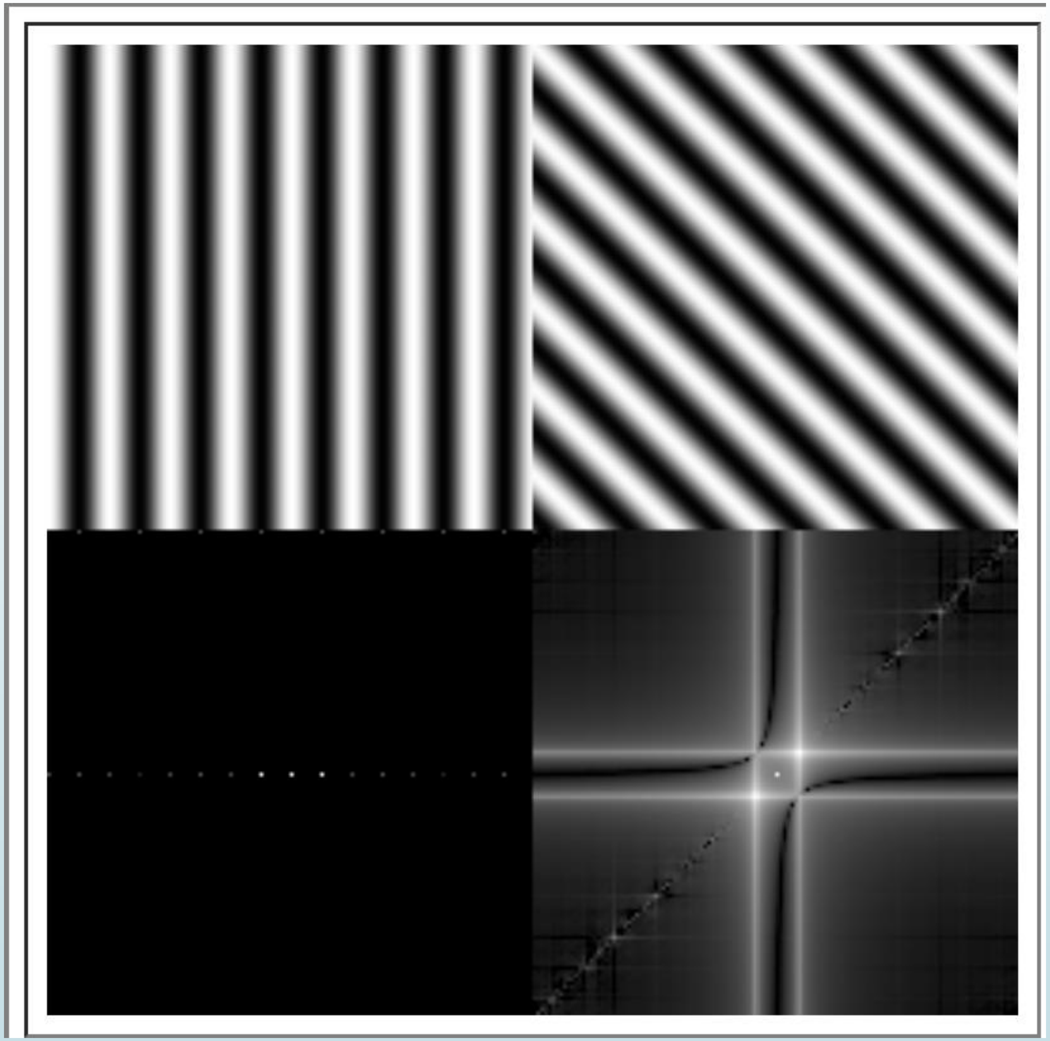
- **Superposition (linearity)**: a system is linear if its response to two combined signals equals the sum of the responses to the individual signals
- Fourier Transform is a linear operation

Fourier Transform Examples

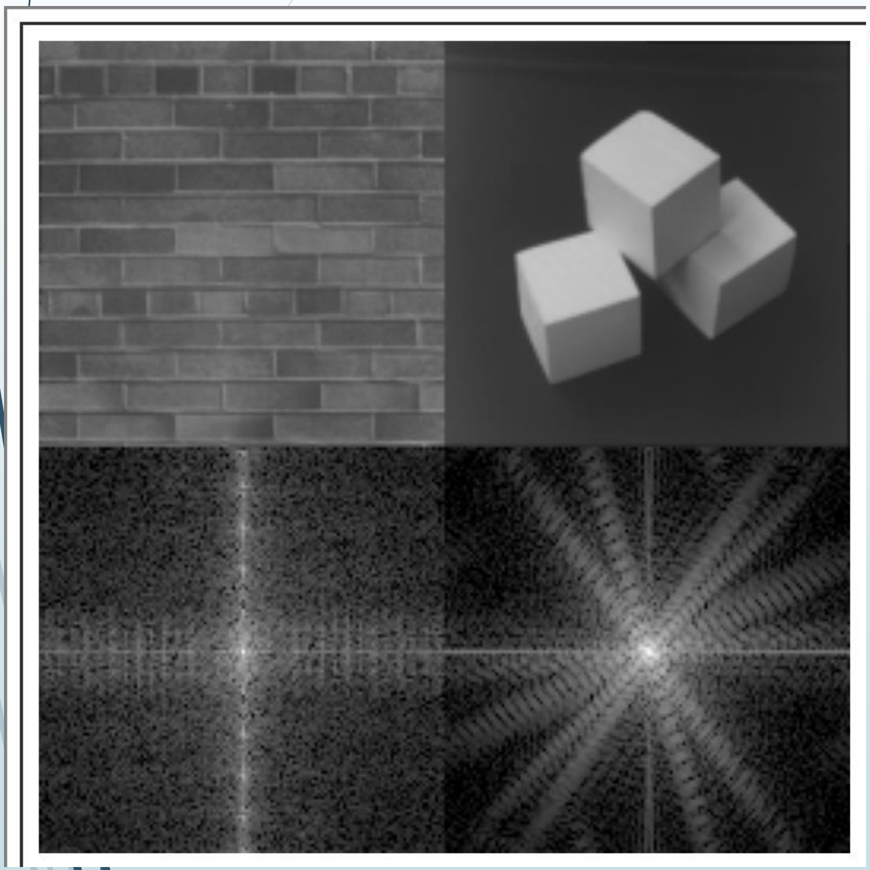


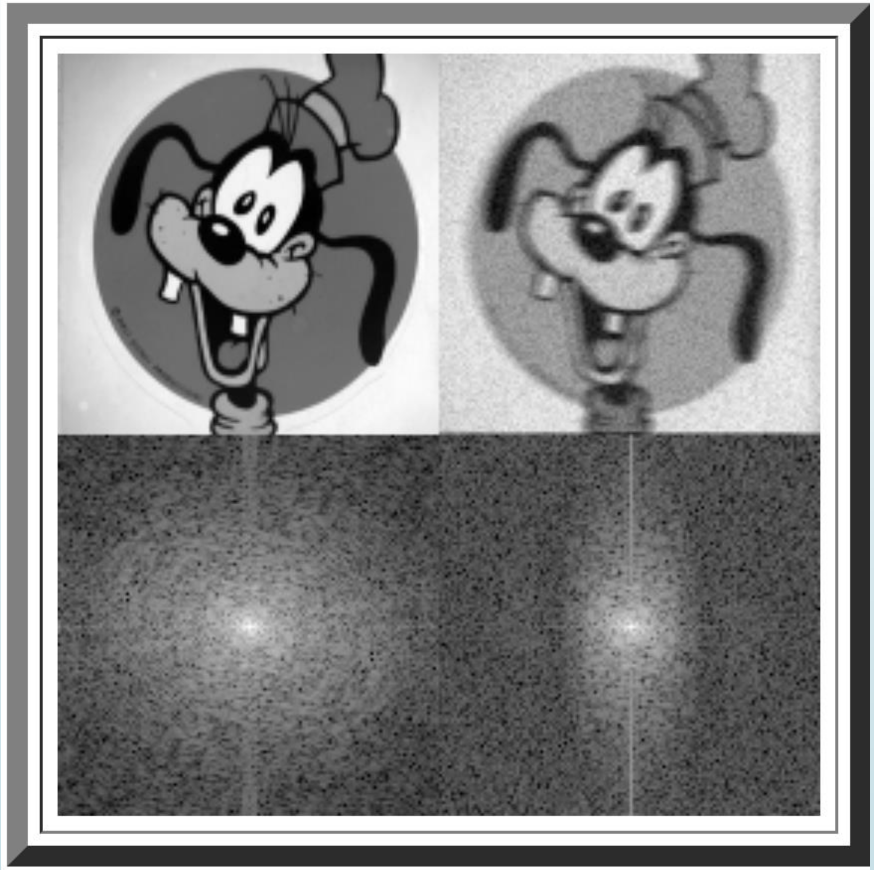


Fourier Transform Examples



Fourier Transform Examples





THE END