# CSC476 Introduction to Computer Vision
# Lecture 5

Feb 20th, 2019

Thinking in Frequency

Bei Xiao

# Course GITHUB

https://github.com/fruittree/CSC476ComputerVision

You can fork, clone, pull request.

Git tutorial:

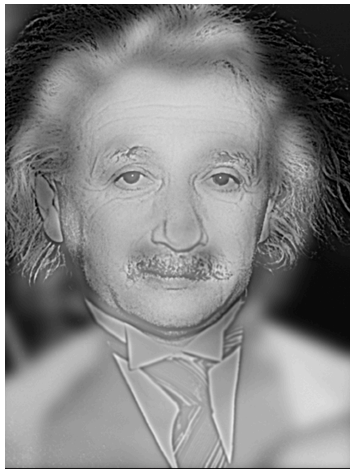
https://guides.github.com/activities/hello-world/

# Today's lecture

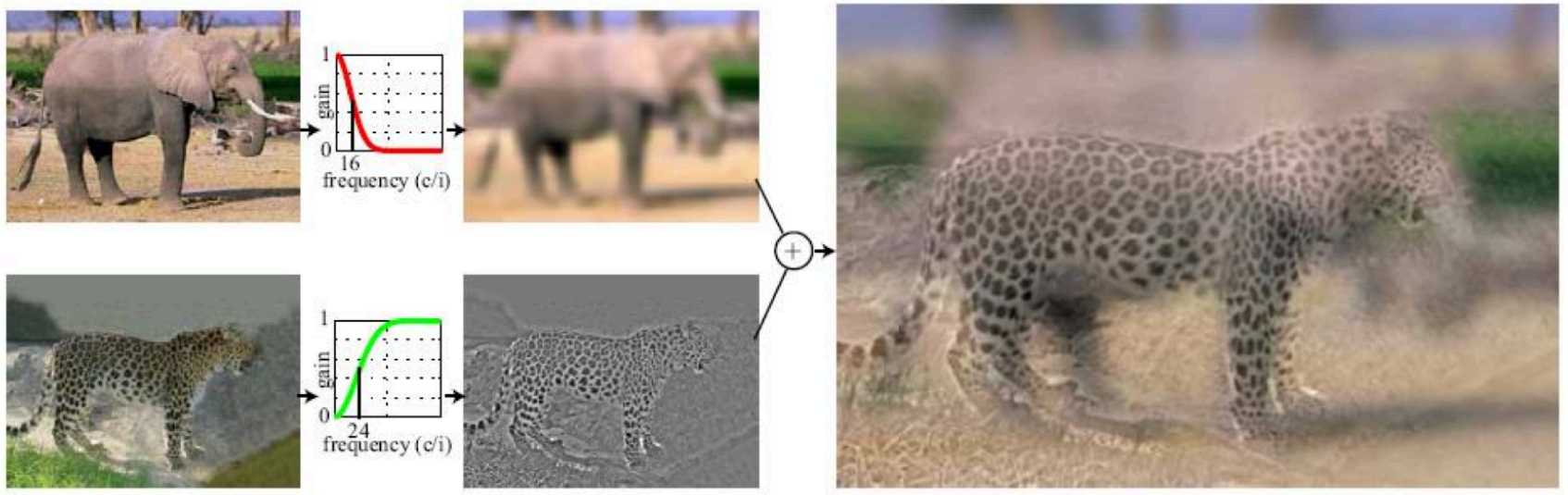
- Fourier transform and frequency domain
  - Fourier transformation in 1d signal
  - Fourier transform of images
  - Fourier transform in Python

- Reminder: Read your text book. Today's lecture covers materials in 3.4. Page 116-126

# Hybrid Image



# Hybrid Image



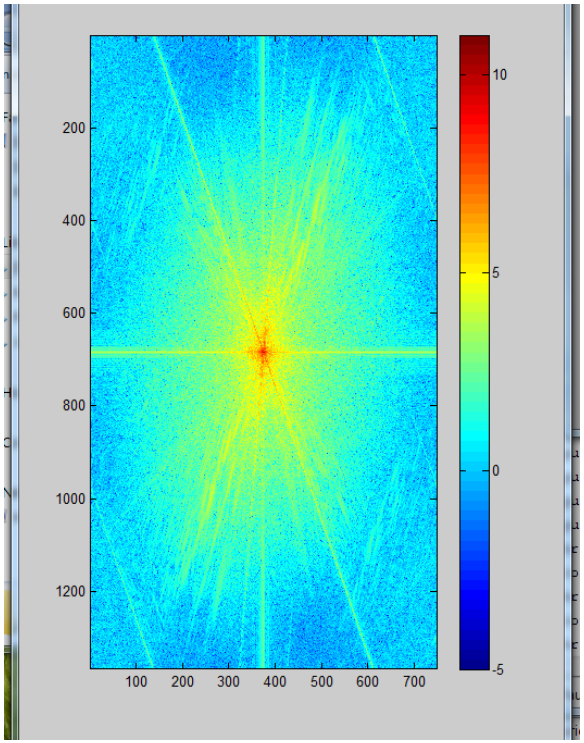

- A. Oliva, A. Torralba, P.G. Schyns, "Hybrid Images," SIGGRAPH 2006

# Why do we get different, distance-dependent interpretations of hybrid images?

?

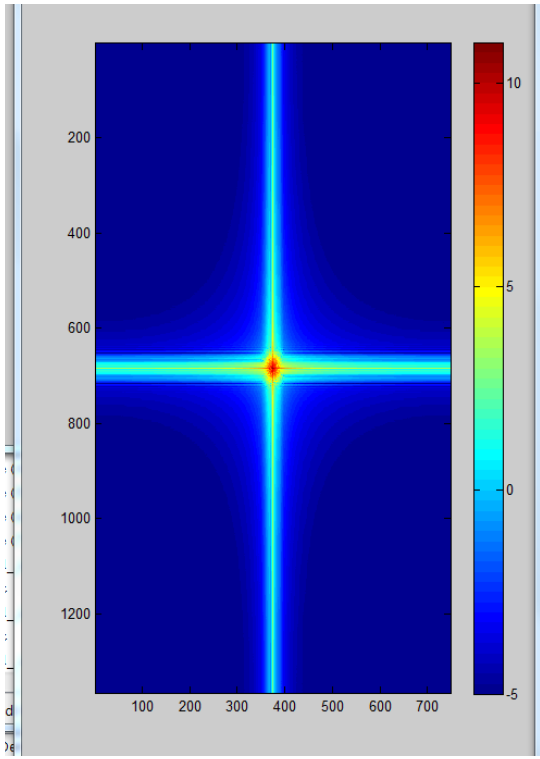
# Hybrid Image in FFT

Hybrid Image



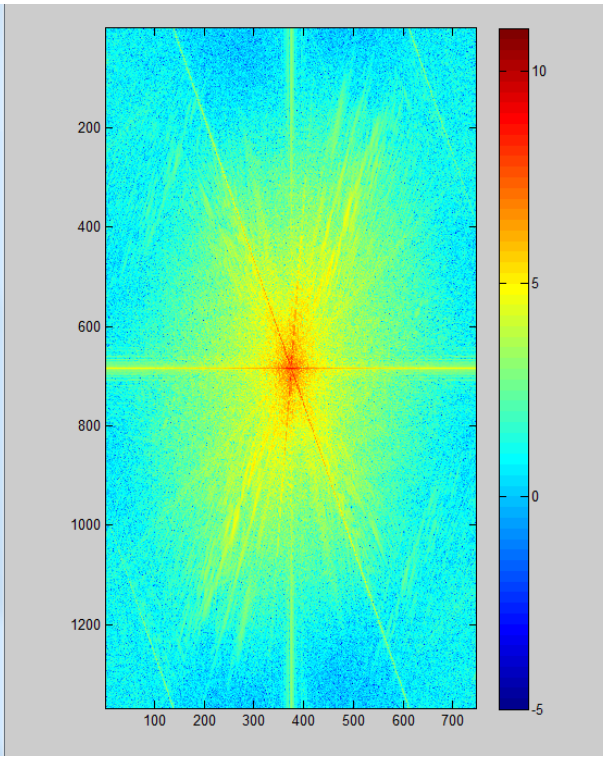

=

Low-passed Image




+

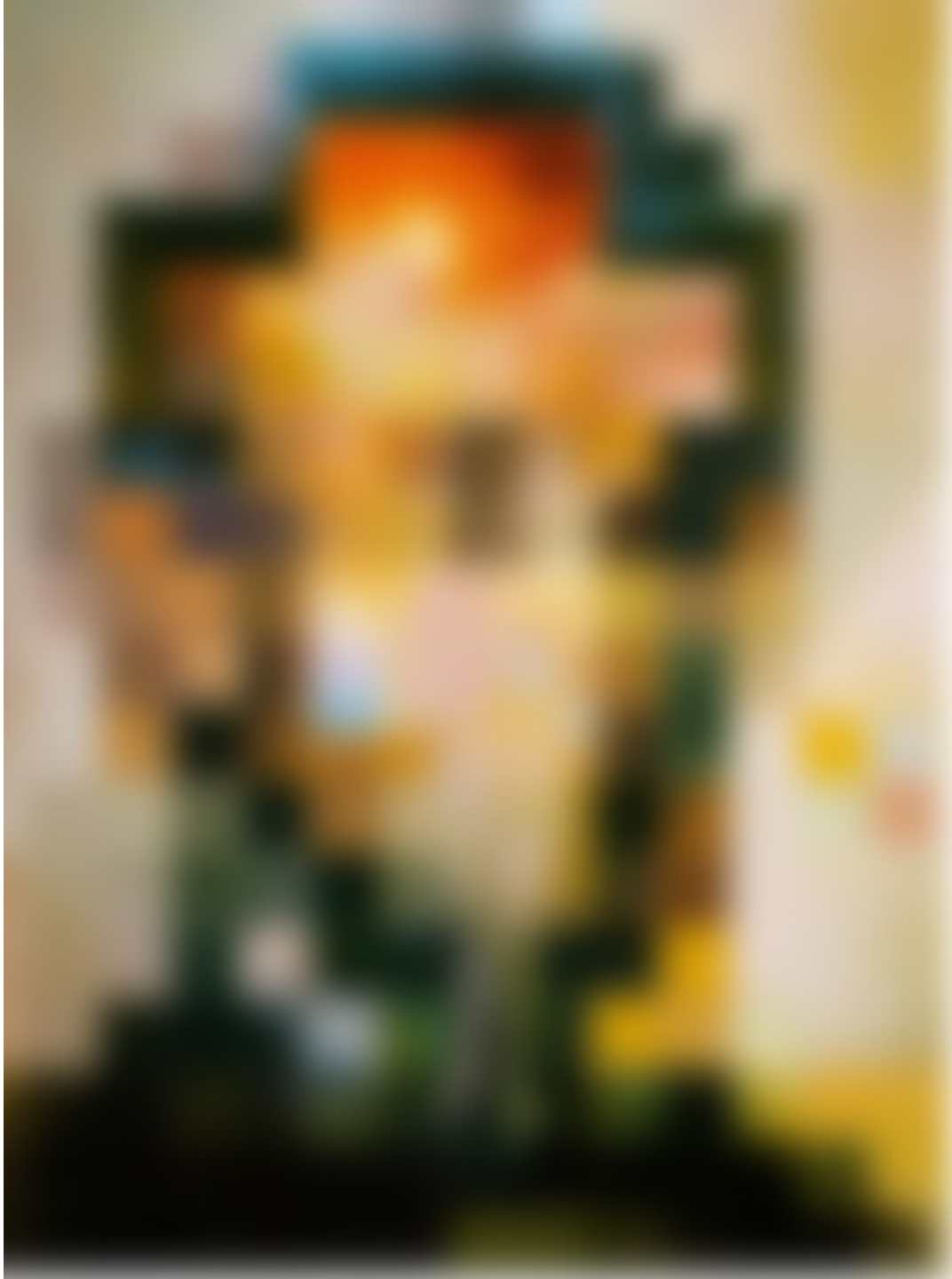
High-passed Image



**Salvador Dali**
*"Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln*", 1976

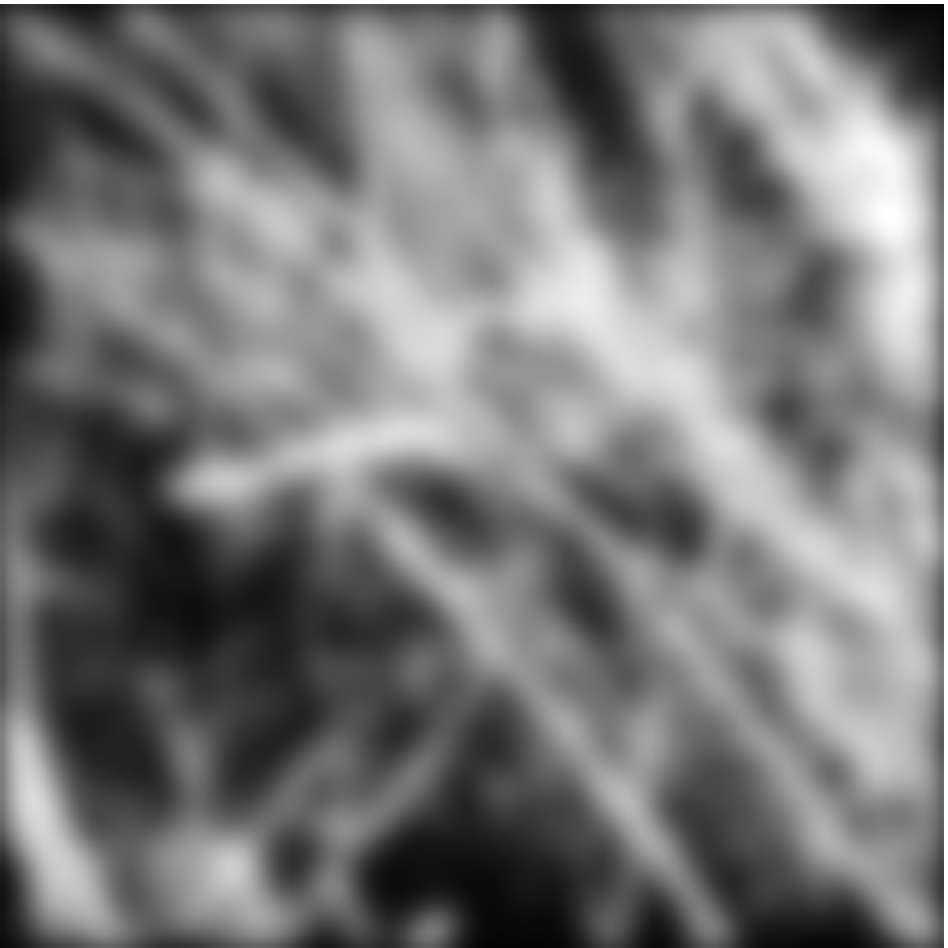
**Salvador Dali invented Hybrid Images?**

**Salvador Dali**
*"Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln*", 1976

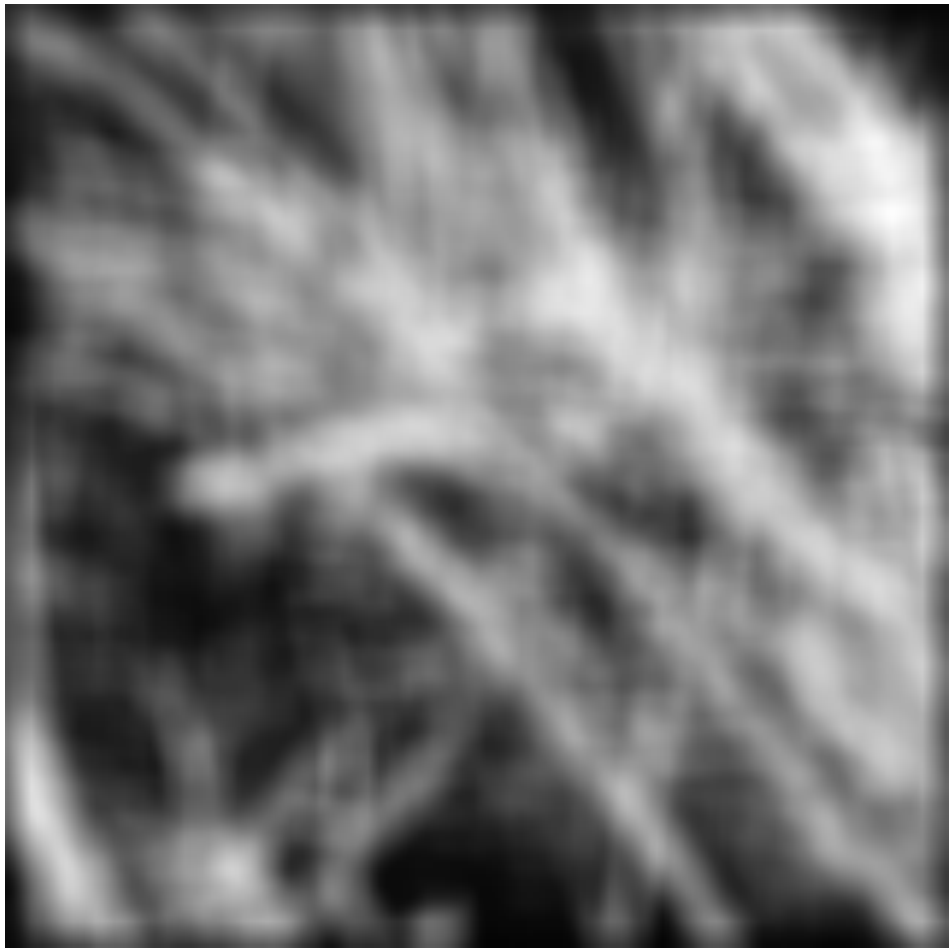


# Why does the Gaussian give a nice smooth image, but the square filter give edgy artifacts?

Gaussian 



Box filter 



How is that a 4MP image can be compressed to a few hundred KB without a noticeable change?

# Jean Baptiste Joseph Fourier (1768-1830)

had crazy idea (1807):

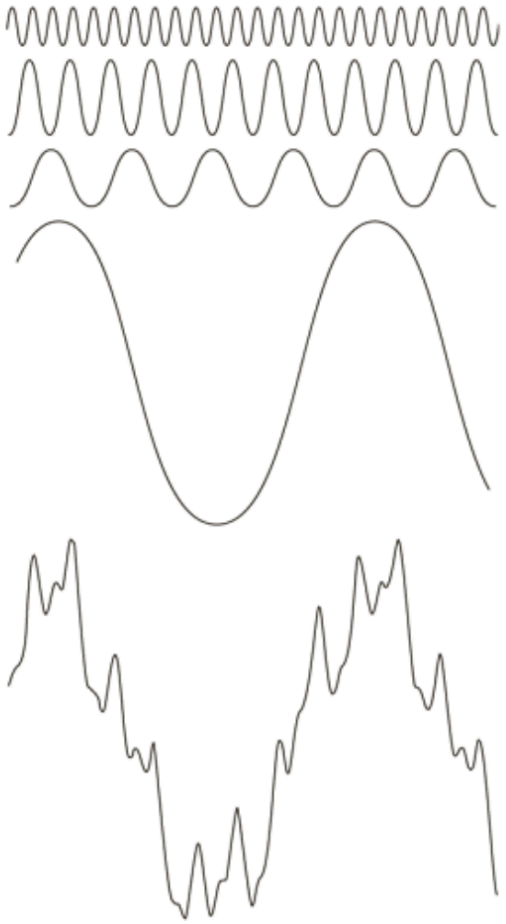
***Any*** *univariate function can be rewritten as a weighted sum of sines and cosines of different frequencies.*

*...the manner in which the author arrives at these equations is not exempt of difficulties and...his analysis to integrate them still leaves something to be desired on the score of generality and even rigour.*

- Don't believe it?
  - Neither did Lagrange, Laplace, Poisson and other big wigs
  - Not translated into English until 1878!
- But it's (mostly) true!
  - called Fourier Series
  - there are some subtle restrictions

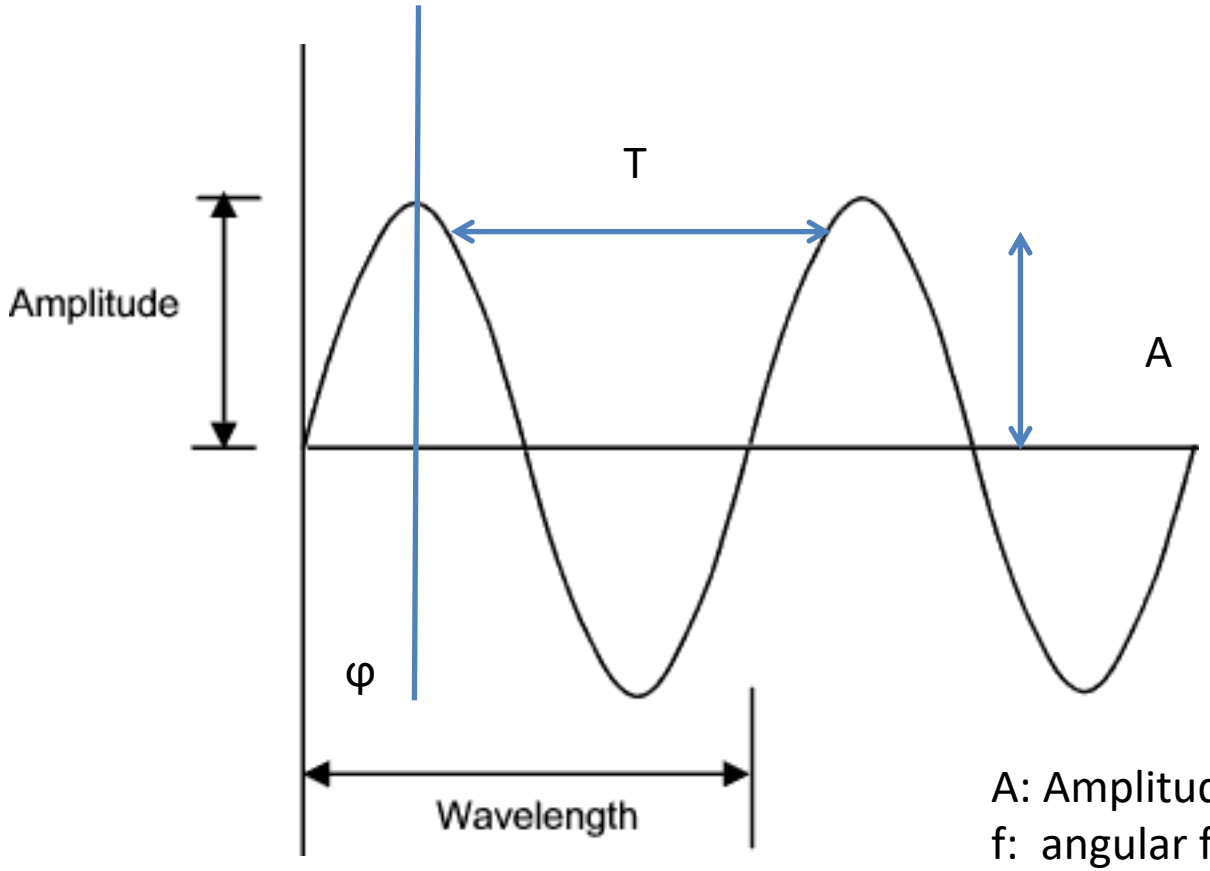
# Overview of Fourier Transform

- In 1807 Fourier discovered that any periodic function can be written as a sum of sin and cos functions
- Function on right is sum of the four functions above it
- This idea is cornerstone of linear systems theory and frequency domain filtering



http://www.csce.uark.edu/~jgauch/5683/notes/ch04a.pdf

# Sine Wave




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

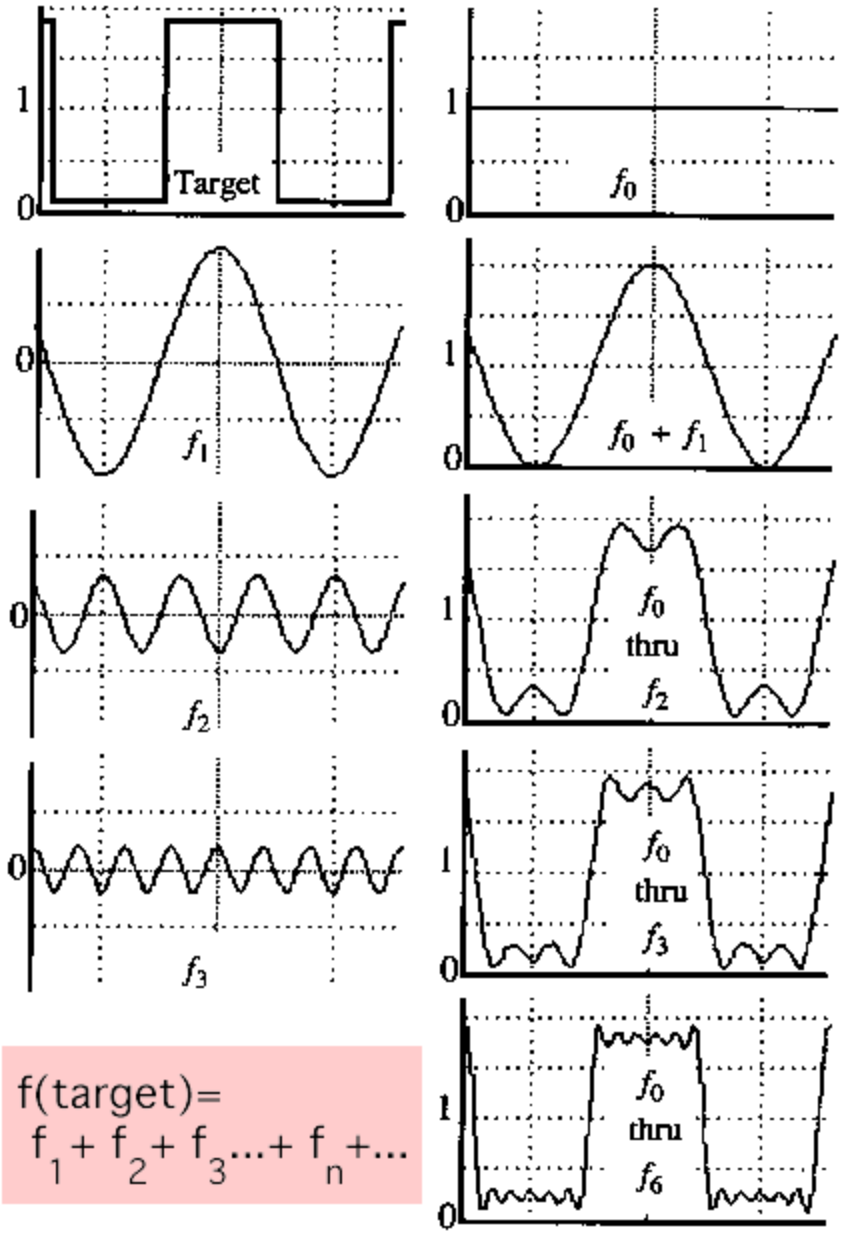
A: Amplitude
f: angular frequency, ω/2π (HZ)
Number of oscillations that occur each second of time
φ: phase

# A sum of sines

Our building block:

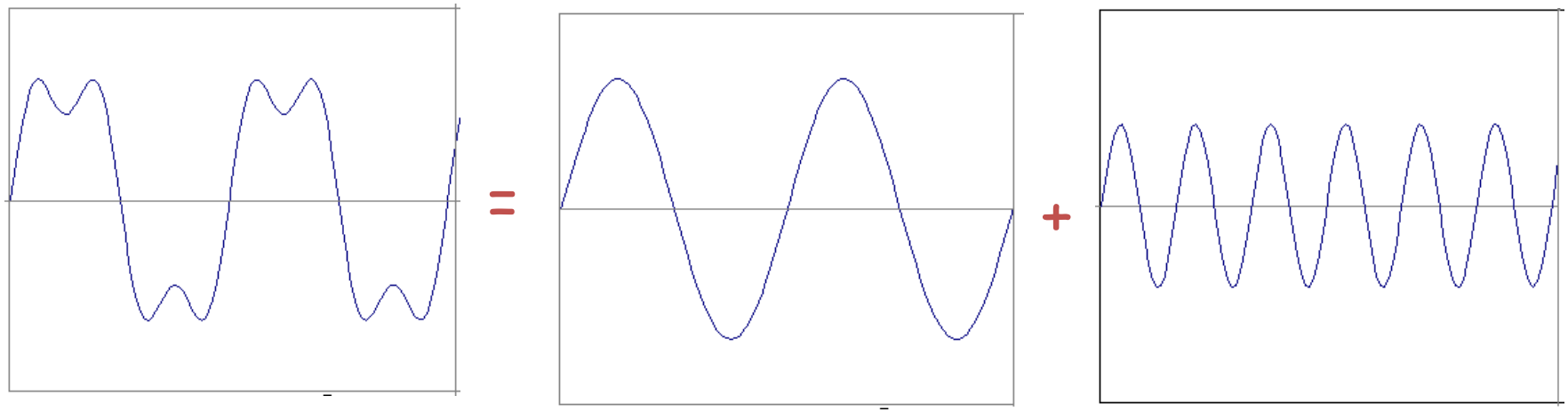
$$A\sin(\omega x + \phi)$$

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

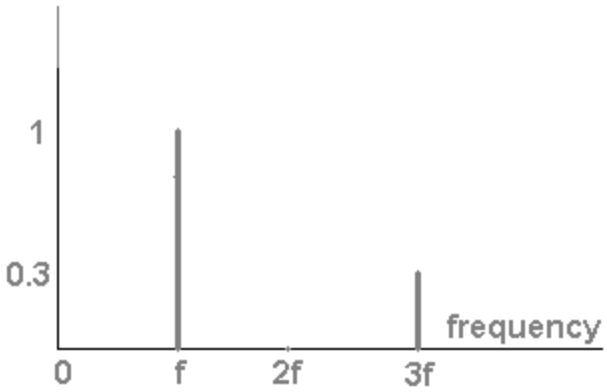


# Frequency Spectra

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



amplitude




Slides: Efros

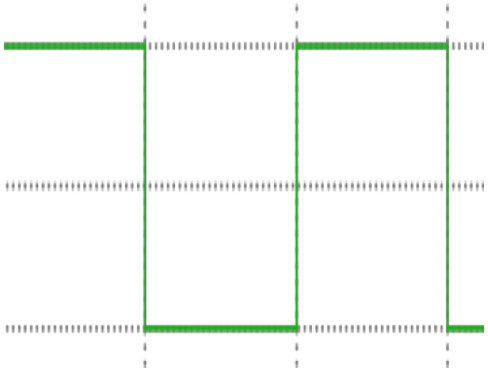
# Why decompose signals into sine waves?

- Why don’t we represent the system with some other periodic signals?
- Sine wave is the only waveform that doesn’t change shape when subject to a linear-time-invariant (LTI) system. Input sinusoids, output will be sinusoids.
- Convolution, image domain filtering,  is LTI.

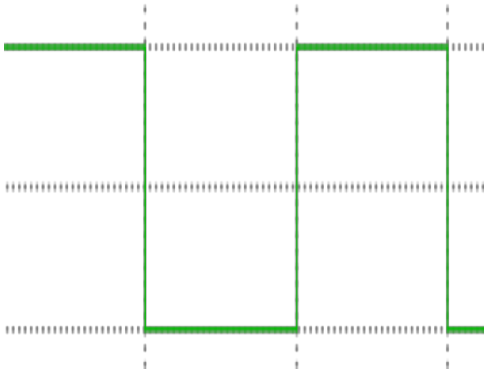
# Fourier transform

# Frequency Spectra

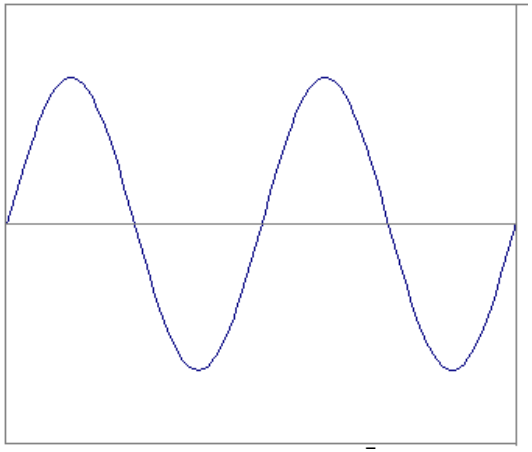
- Consider a square wave f(x)



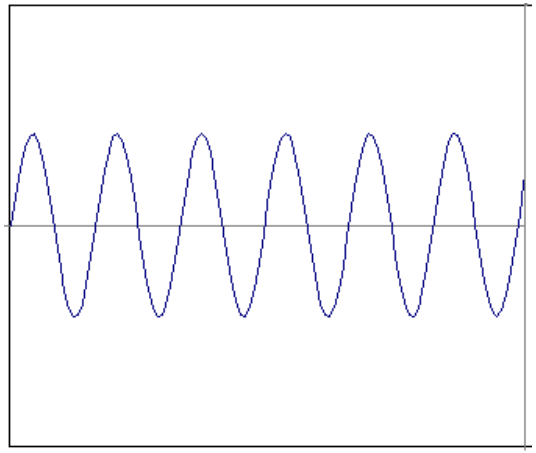
# Frequency Spectra



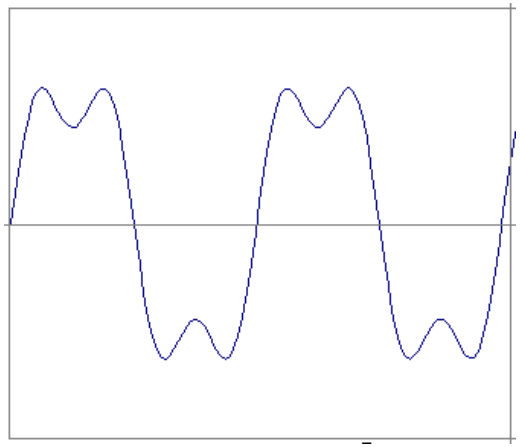
=



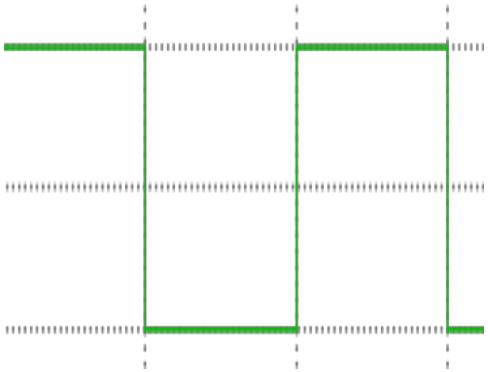
+



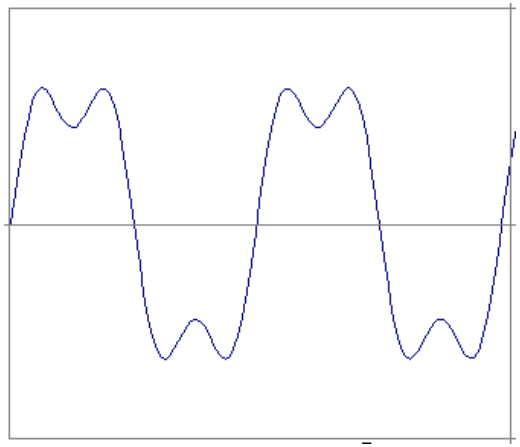
=



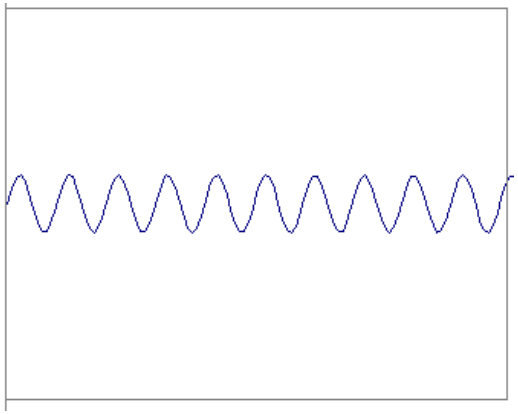
# Frequency Spectra



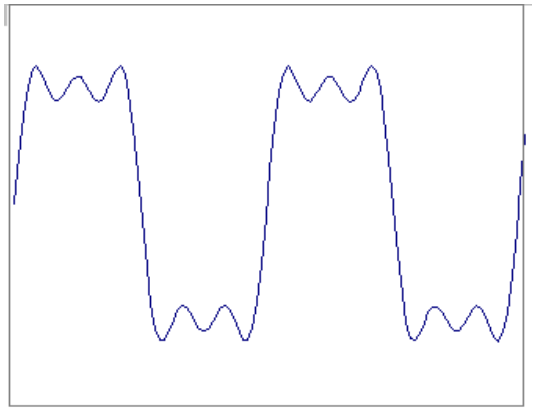
=



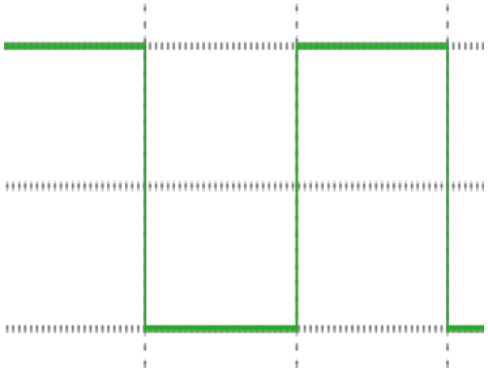
+



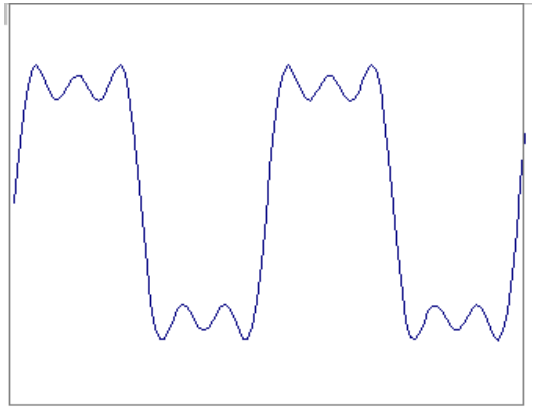
=



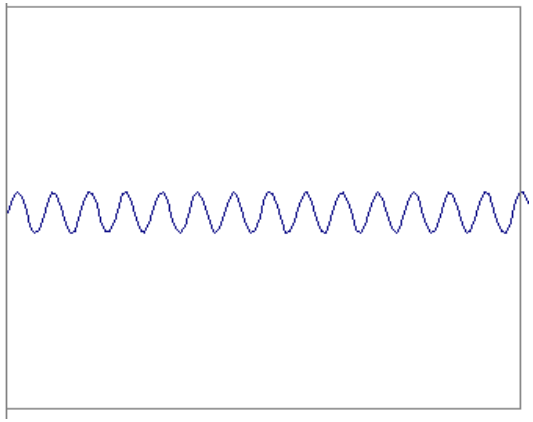
# Frequency Spectra



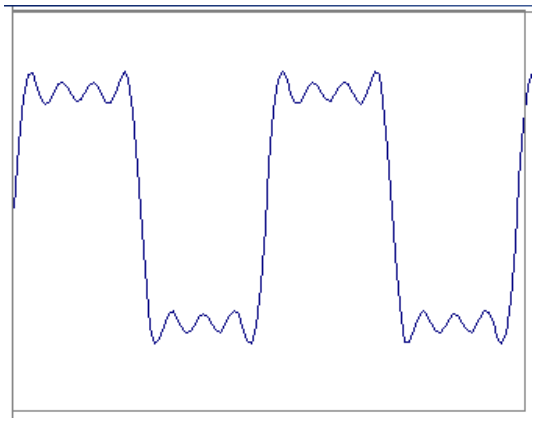
=



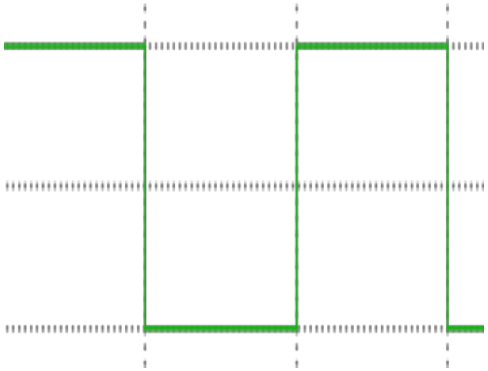
+



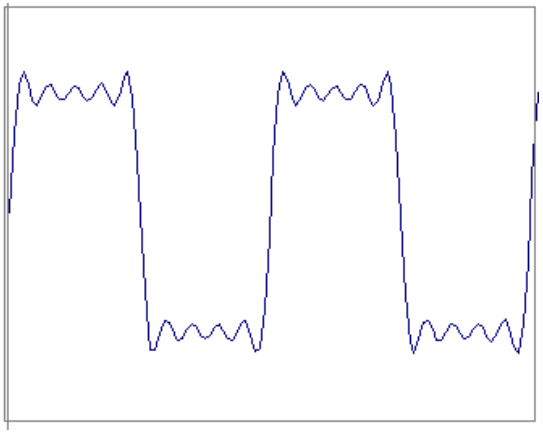
=



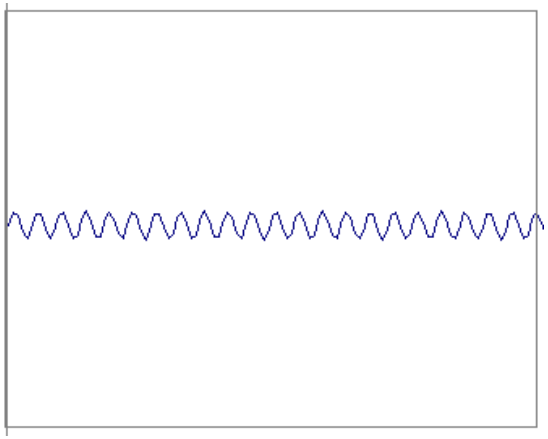
# Frequency Spectra



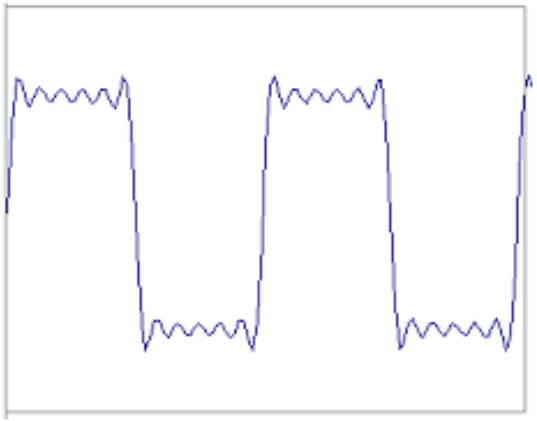
=



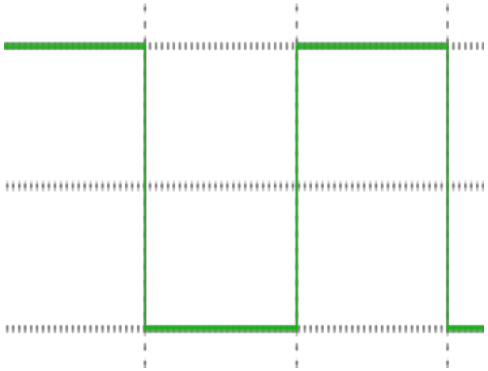
+

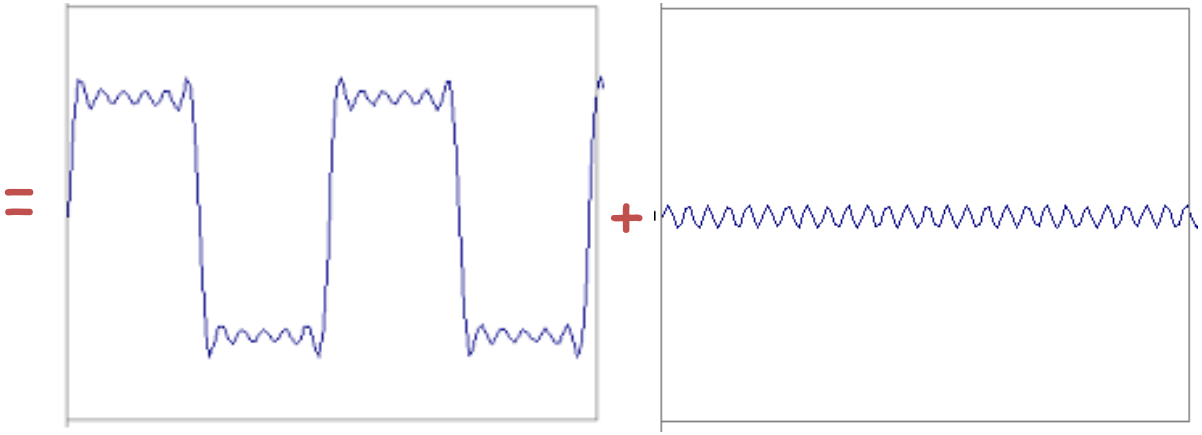


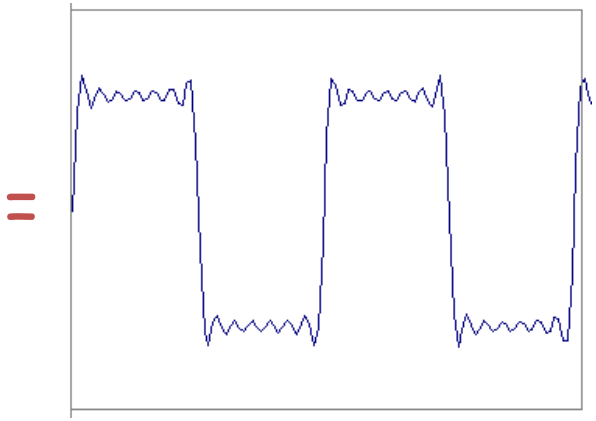
=



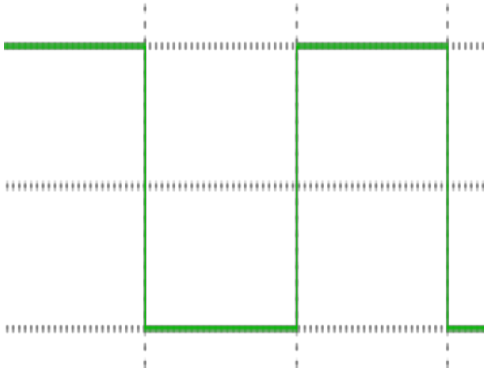
# Frequency Spectra



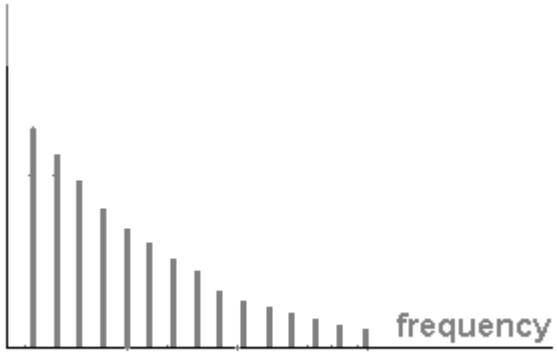




# Frequency Spectra



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



# Fourier Transform

- Fourier Transform

Inverse Fourier transform

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \qquad f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

Time f(t)  Frequency f(w)

What is transformation? It is a mapping between two domains

# Fourier Transform

- Let f(x) be a continuous function, then the FT is the function F(u) given by:

$$\mathrm{FT}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\Pi ux}\, dx$$

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

$$F(u).re = \int_{-\infty}^{\infty} f(x)\cos(-2\Pi ux)dx$$

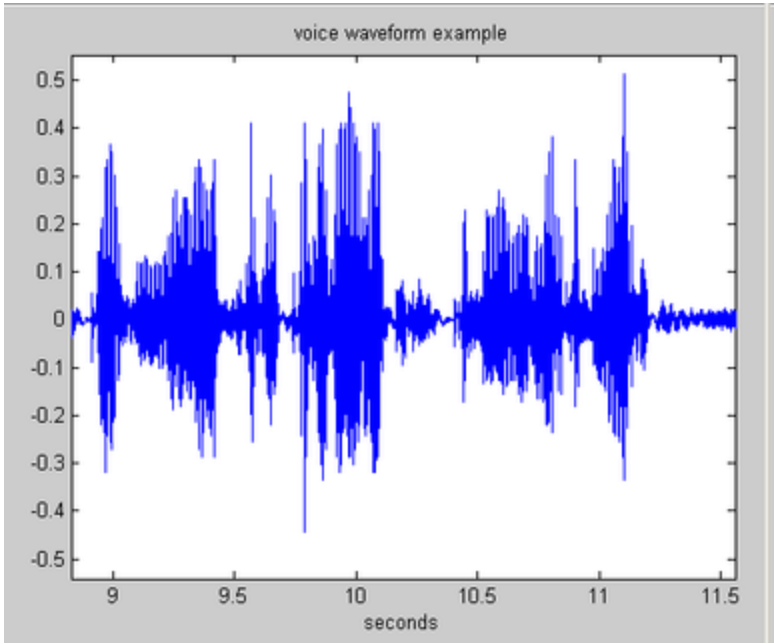
Real part

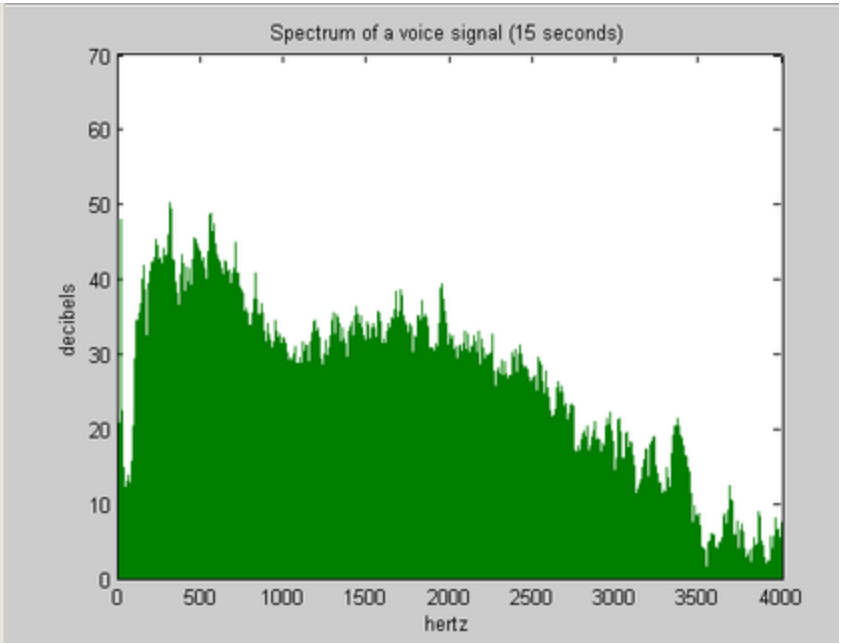
$$F(u).im = \int_{-\infty}^{\infty} f(x)\sin(-2\Pi ux)dx$$

Imaginary part

# Example: Music

- We think of music in terms of frequencies at different magnitudes

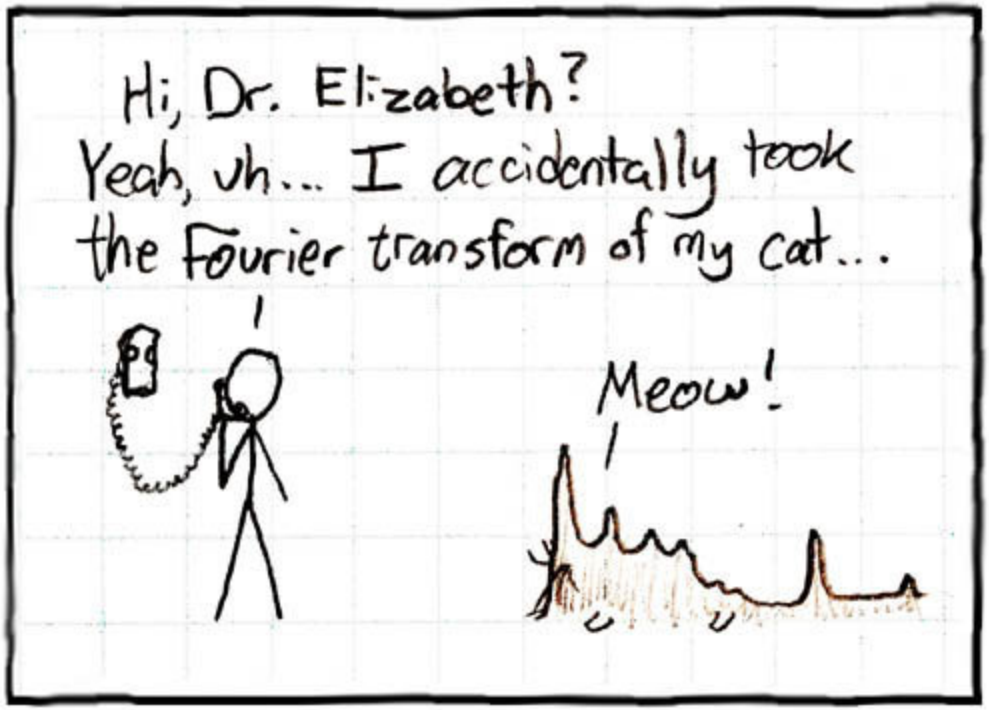



Slide: Hoiem

# Other signals

- We can also think of all kinds of other signals the same way




xkcd.com

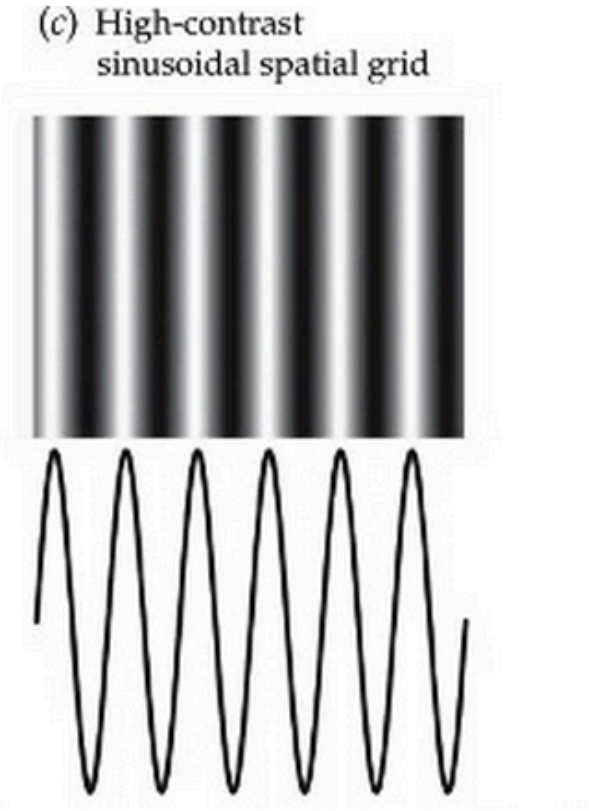
# Fourier Analysis of Images

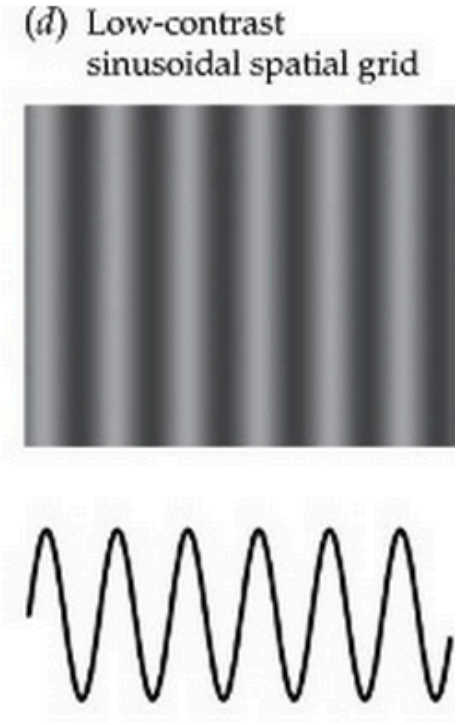
- Let’s watch a funny video:

- https://www.youtube.com/watch?v=mEN7DTdHbAU

# Spatial Frequency in Image

- Any structure periodic in images

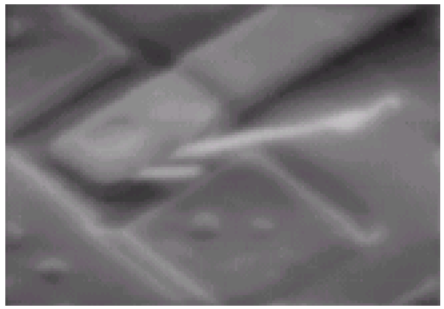


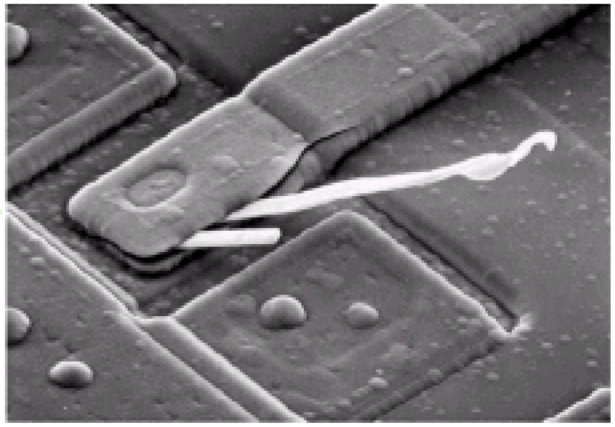
# Image composition in Fourier Space

- An image can be represented as two components: high frequencies and low frequencies
  - Low frequencies make up the bulk of the information (areas of low variation in intensity)
  - High frequencies make up the edges and fine detail (areas of high variation in intensity)

Low Frequencies only:



High Frequencies only:



THE ARC.

# Fourier Transform

- Let f(x) be a continuous function, then the FT is the function F(u) given by:

$$\mathrm{FT}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\Pi ux}\, dx$$

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

$$F(u).re = \int_{-\infty}^{\infty} f(x)\cos(-2\Pi ux)dx$$

$$F(u).im = \int_{-\infty}^{\infty} f(x)\sin(-2\Pi ux)dx$$

# Fourier Transform

- Let f(x,y) be a continuous function, then the FT is the function F(u,v) given by:

$$\mathrm{FT}\{f(x,y)\} = F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-i2\Pi(ux+vy)}\, dxdy$$

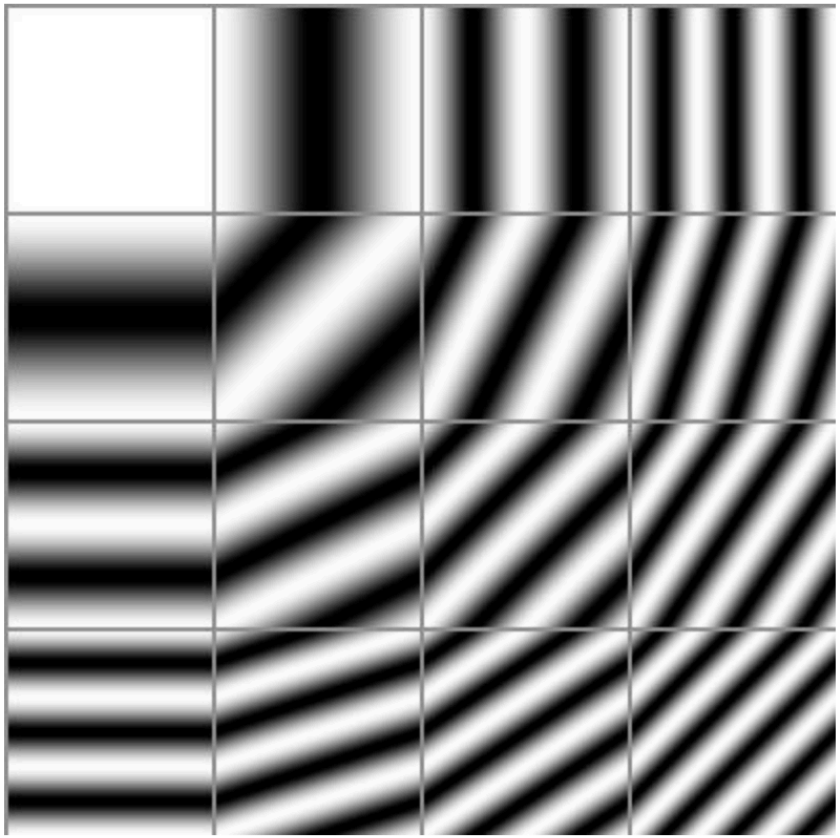
$$e^{i\theta} = \cos(\theta) + i\sin(\theta)$$

$$F(u,v).re = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\cos(-2\Pi(ux+vy))dxdy$$

$$F(u,v).im = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\sin(-2\Pi(ux+vy))dxdy$$

# 2D Fourier Transform Basis

- The 2D FT basis functions look like waves with different frequencies and orientations
- Orientation and frequency of wave given by (u,v) value



http://www.csce.uark.edu/~jgauch/5683/notes/ch04a.pdf

# 2D discrete Fourier transform

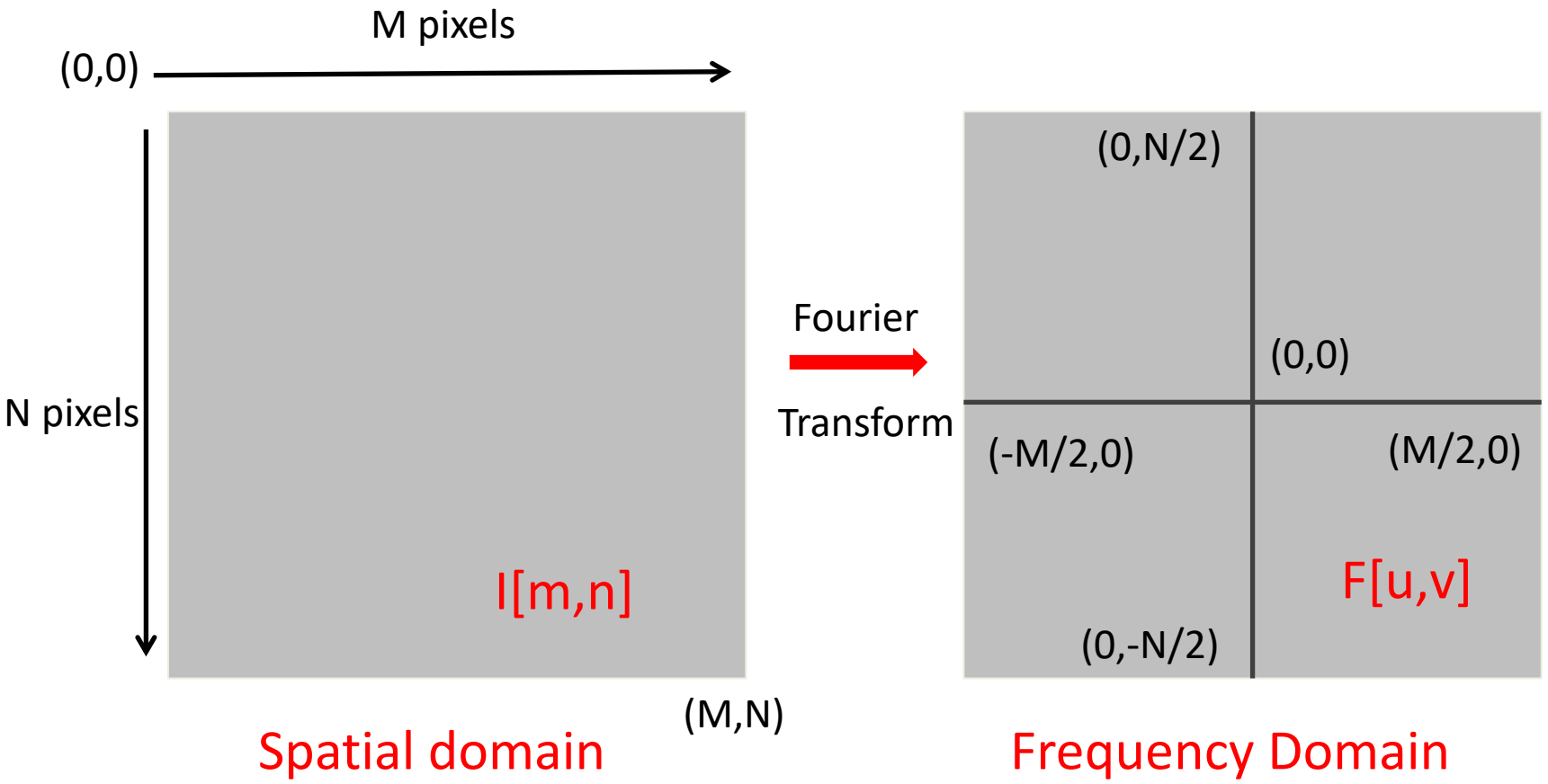
- The discrete FT for a discrete signal f(x) with N values is given by:

$$F(u) = \frac{1}{N} \sum_{x=0..N-1} f(x) e^{-i2\Pi ux/N}$$

$$F(u).re = \frac{1}{N} \sum_{x=0..N-1} f(x) \cos(-2\Pi ux/N)$$

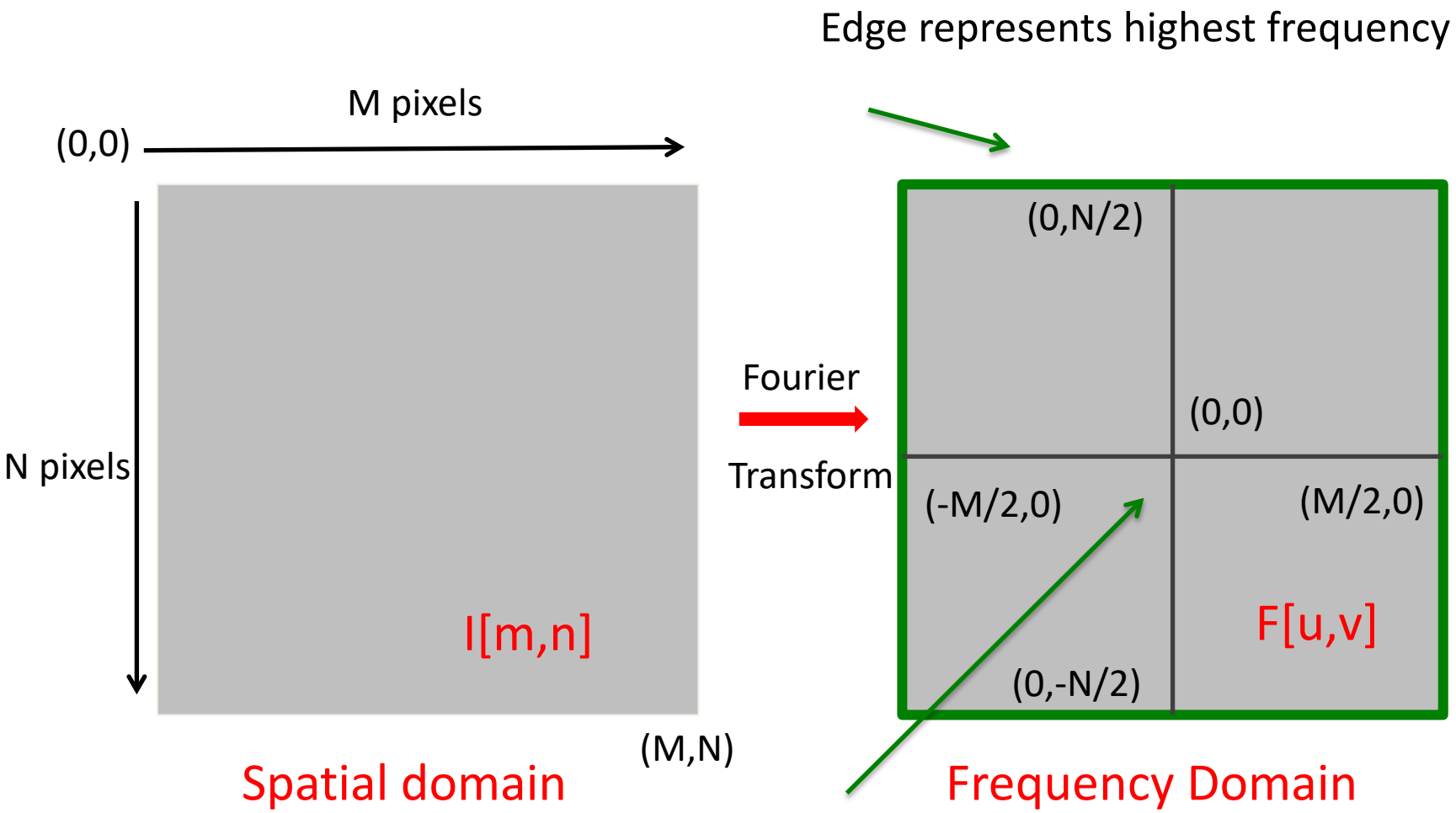
$$F(u).im = \frac{1}{N} \sum_{x=0..N-1} f(x) \sin(-2\Pi ux/N)$$

# Fourier Transform of Images

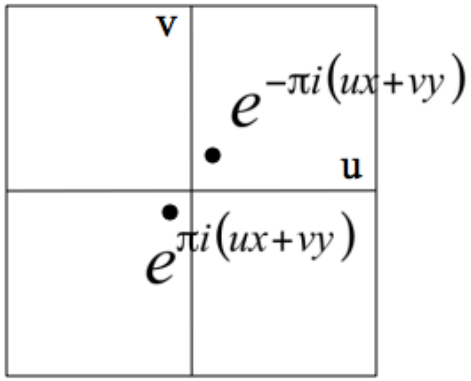


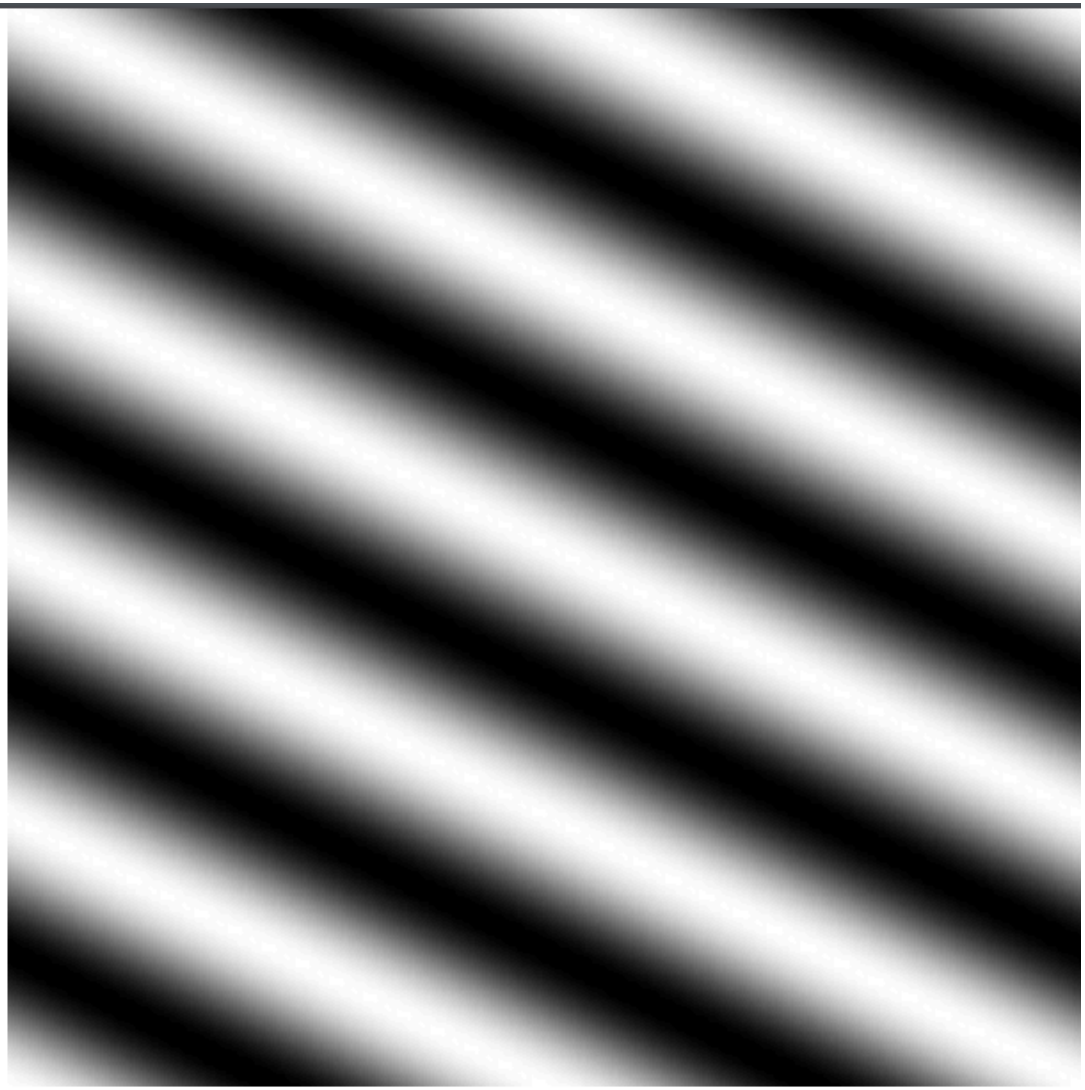

2D FFT can be composed as two discrete Fourier Transforms in 1 dimension

# Fourier Transform of Images



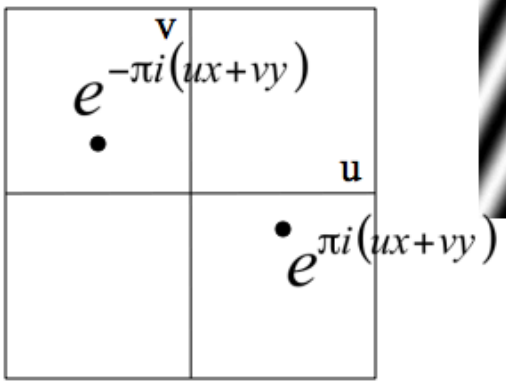
To get some sense of what basis elements look like, we plot a basis element --- or rather, its real part --- as a function of x,y for some fixed u, v. We get a function that is constant when (ux+vy) is constant. The magnitude of the vector (u, v) gives a frequency, and its direction gives an orientation. The function is a sinusoid with this frequency along the direction, and constant perpendicular to the direction.



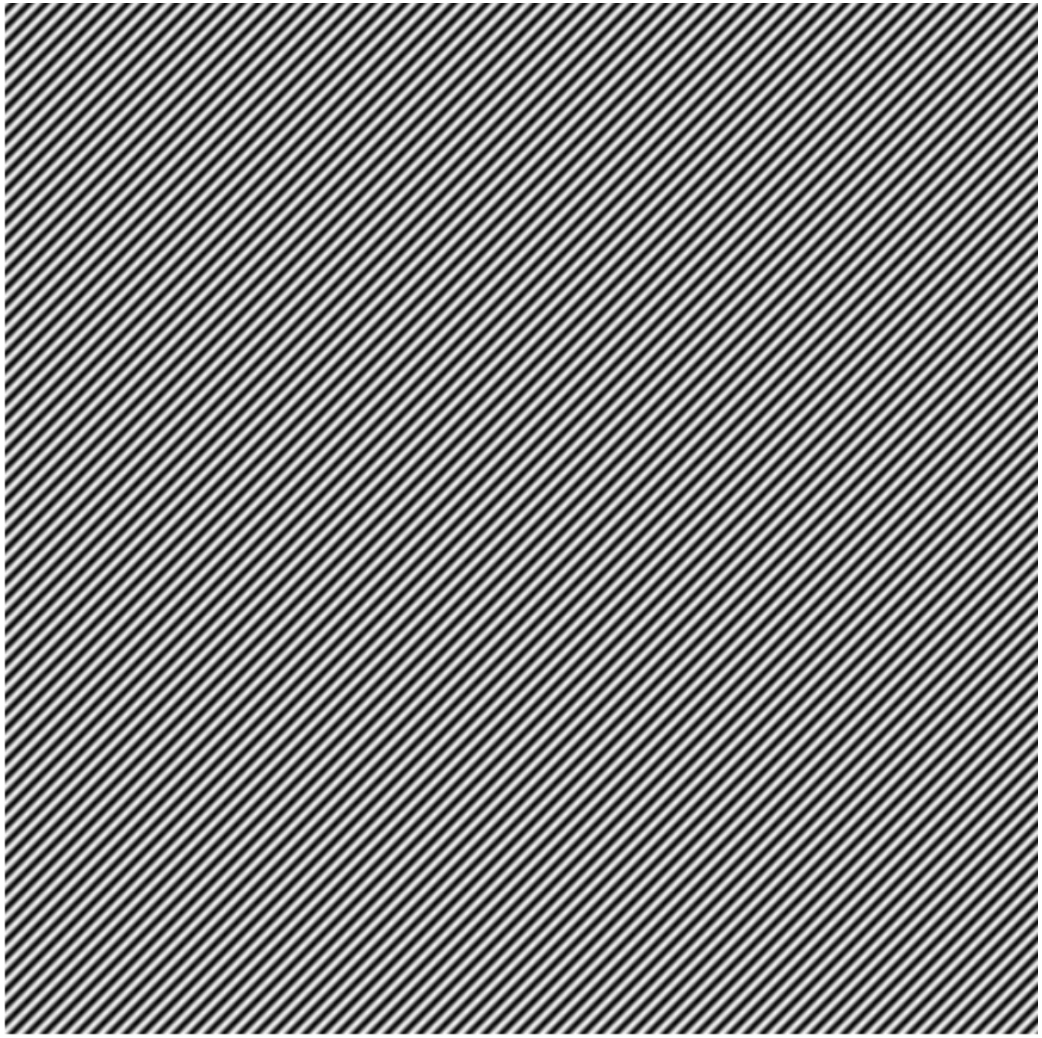



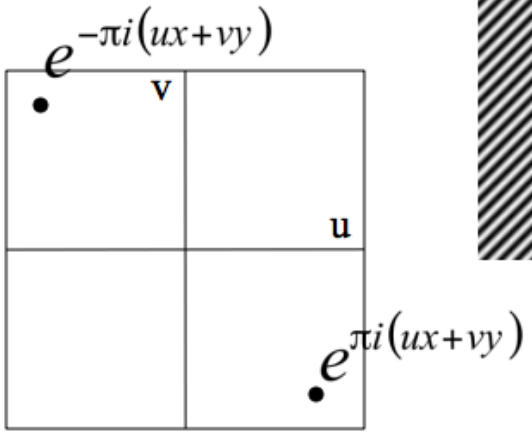
x,y plane with u,v given. Orientation tan a = v/u
Frequency of the sinusoid sqrt(u^2 + v^2).
The real component only here as intensity image.

Here u and v are larger than in the previous slide.

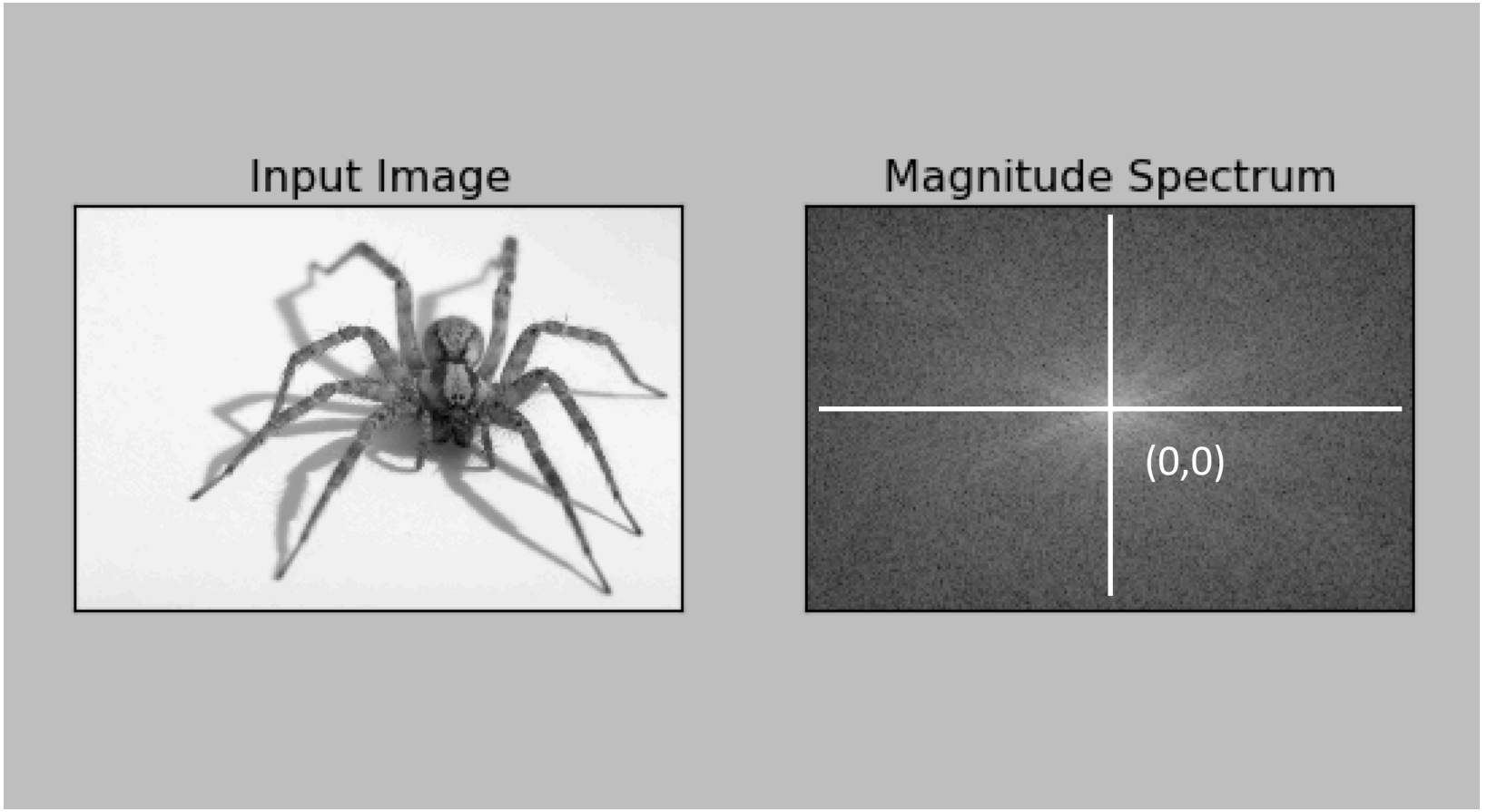


And larger still...





# Fourier Transform of Images

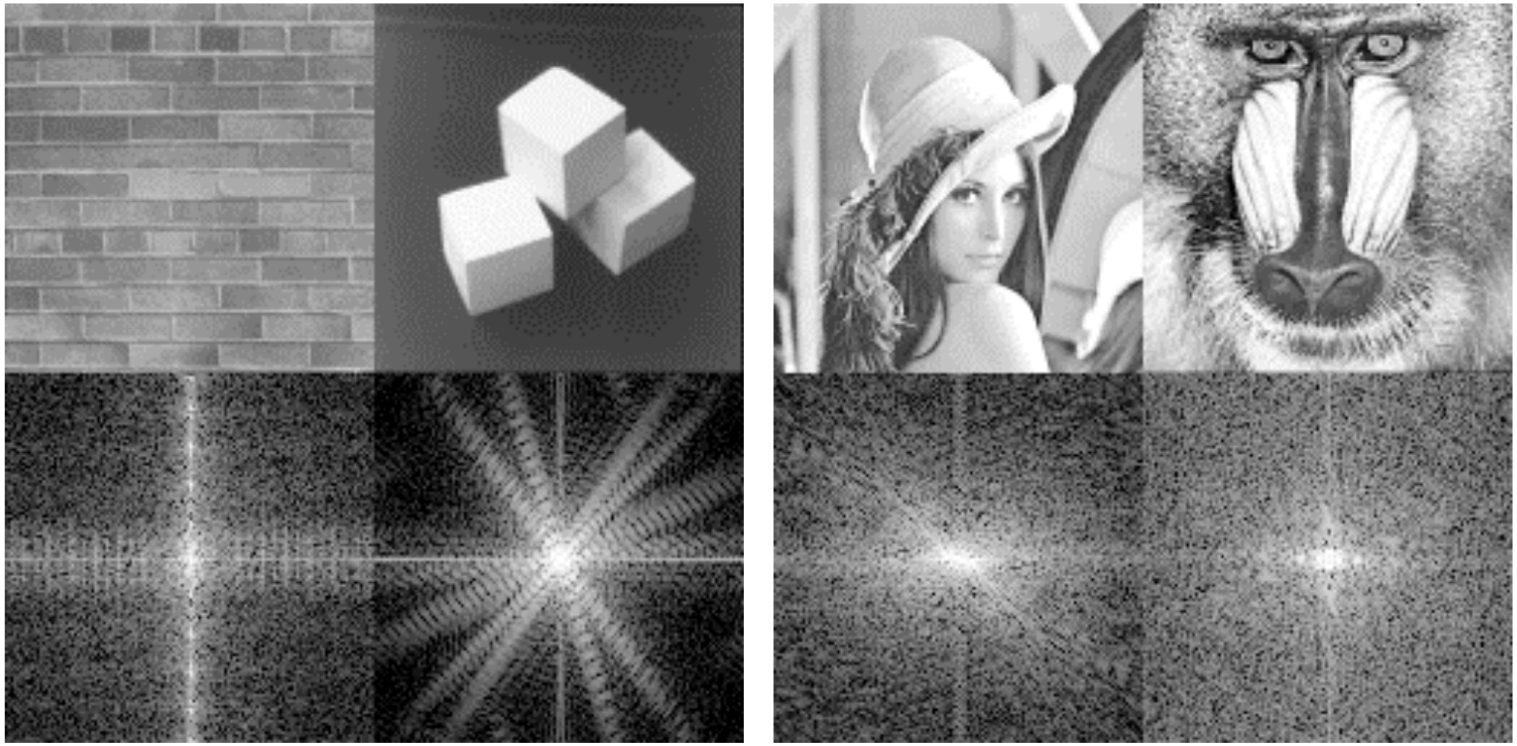



Spatial domain

Frequency Domain

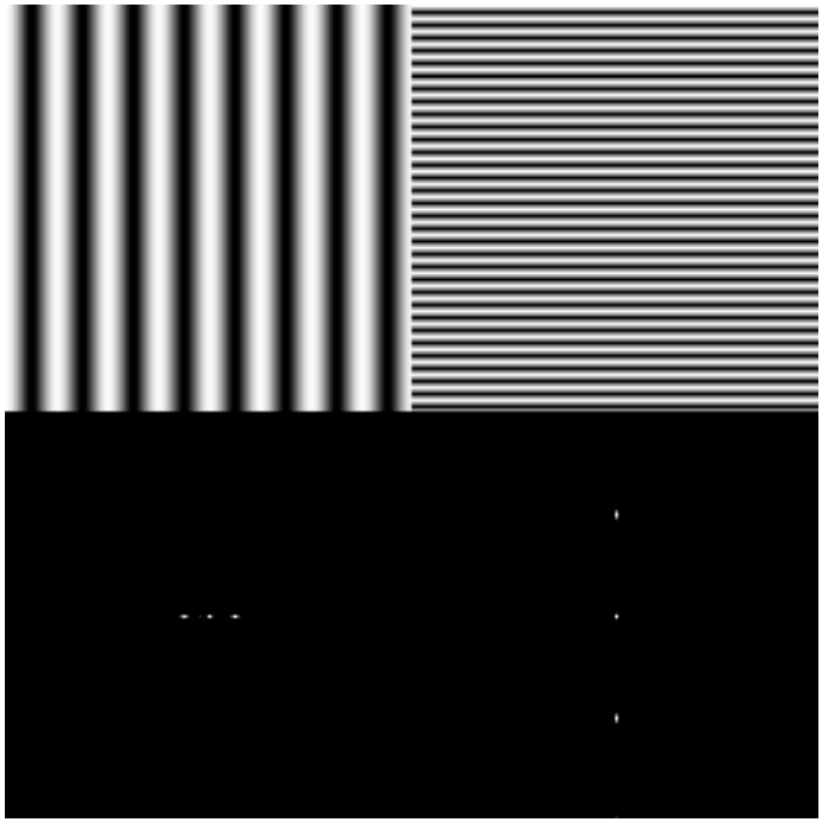
# FT of natural images

- FT of natural images



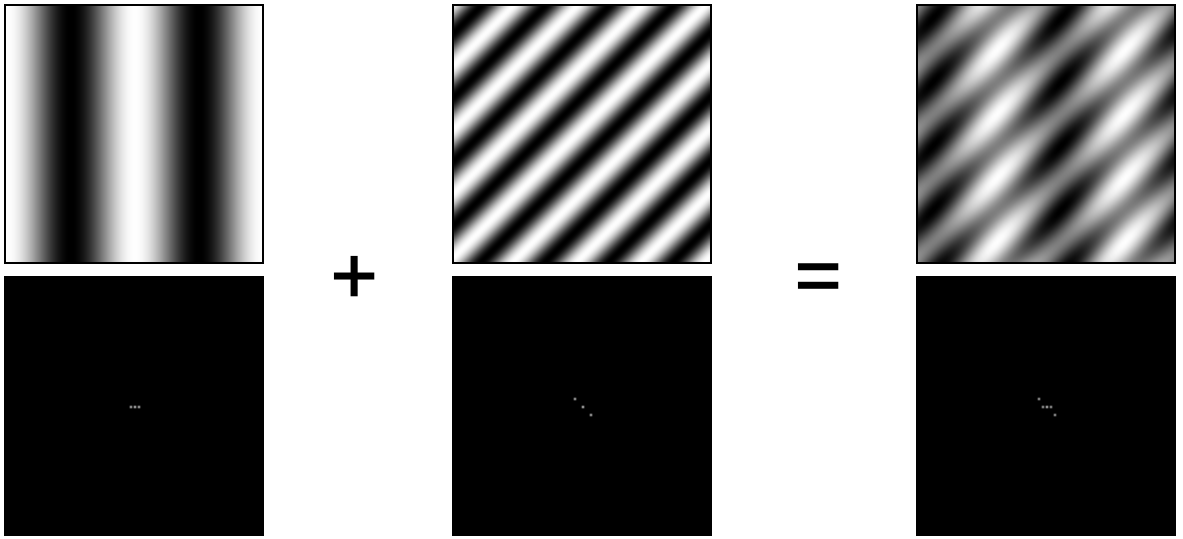
# Fourier analysis in images

- For display purposes, we often show |F(u,v)| of an image on a log scale with the origin in the center of the image
- Here we have FT of two wave images



http://www.csce.uark.edu/~jgauch/5683/notes/ch04a.pdf

# Signals can be composed

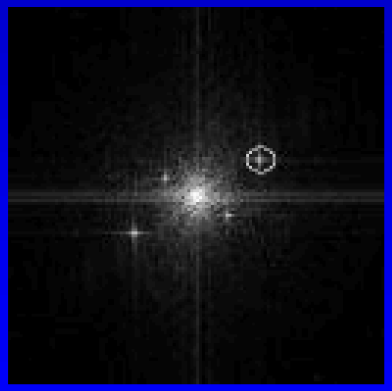


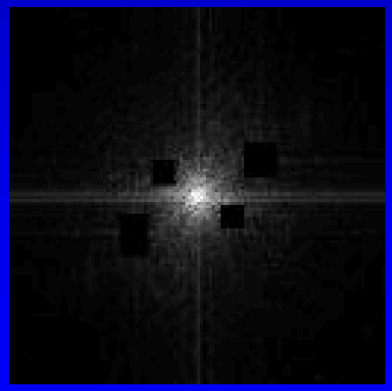
http://sharp.bu.edu/~slehar/fourier/fourier.html#filtering
More: http://www.cs.unm.edu/~brayer/vision/fourier.html

# Noise Removal

FFT

Inverse FFT





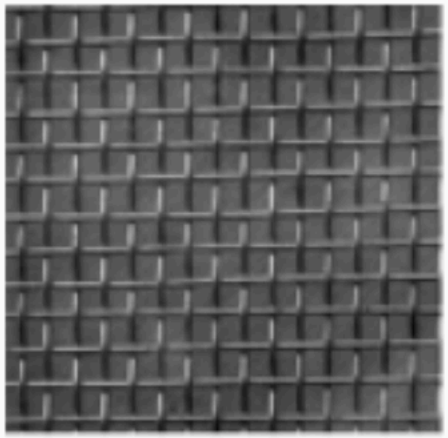
Noise Pattern Stands Out as Four Spikes

Four Noise Spikes Removed

Edit FFT

Source: www.mediacy.com/apps/fft.htm, Image Pro Plus FFT Example. Last seen online in 2004.

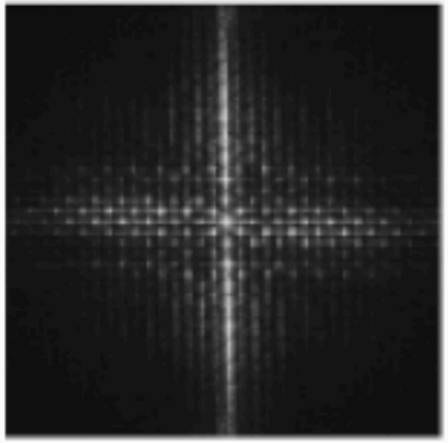
# Pattern/Texture Recognition



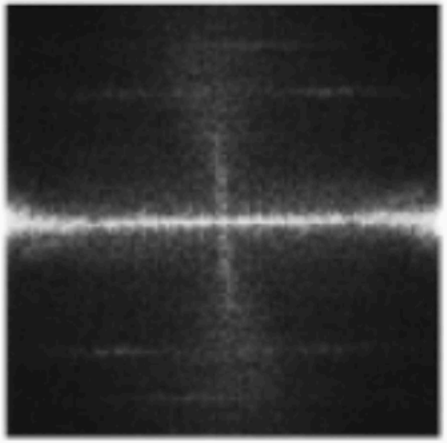
(a) A periodic texture (D1).

(b) A directional texture (D106).

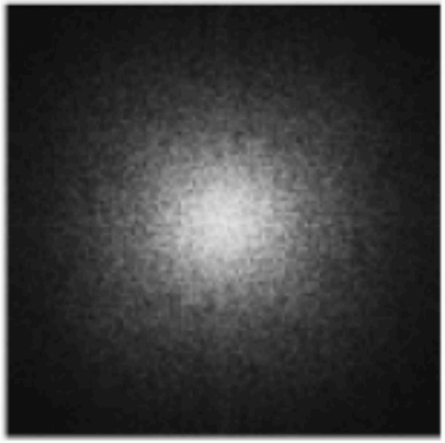
(c) A random texture (D106).



(d) The Fourier spectrum of (a).



(e) The Fourier spectrum of (b).



(f) The Fourier spectrum of (c).

Source: Lee and Chen, A New Method for Coarse Classification of Textures and Class Weight Estimation for Texture Retrieval, *Pattern Recognition and Image Analysis*, Vol. 12, No. 4, 2002, pp. 400–410.

# Exercise 1

- Open an image
- Add both Salt and Pepper, Gaussian noise.
- Display FFT images of original and noisy images.
- img = misc.imread('peppers256.png',flatten=1)
- f = np.fft.fft2(img)
- fshift = np.fft.fftshift(f)
- magnitude_spectrum = 20*np.log(np.abs(fshift))

# Exercise 2

- Apply the Box or Gaussian filter to an image and compare the image with the Fourier transformed image.

# Fourier Transform

- Fourier transform stores the magnitude and phase at each frequency
  - Magnitude encodes how much signal there is at a particular frequency
  - Phase encodes spatial information (indirectly)
  - For mathematical convenience, this is often notated in terms of real and complex numbers

Amplitude: $A = \pm\sqrt{R(\omega)^2 + I(\omega)^2}$

Phase: $\phi = \tan^{-1}\frac{I(\omega)}{R(\omega)}$

# The Convolution Theorem

- The Fourier transform of the convolution of two functions is the product of their Fourier transforms

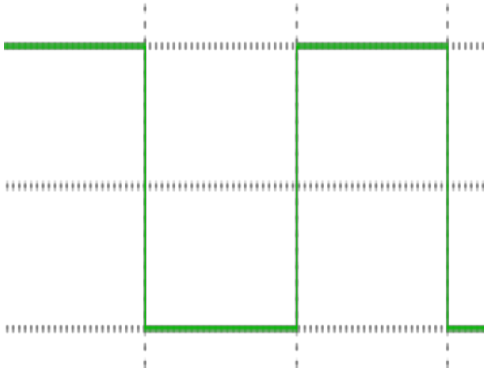
$$\mathrm{F}[g * h] = \mathrm{F}[g]\mathrm{F}[h]$$

- **Convolution** in spatial domain is equivalent to **multiplication** in frequency domain!

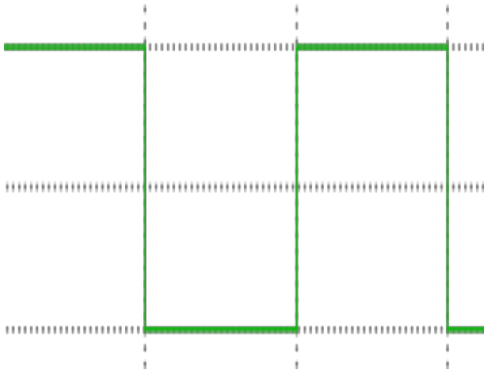
$$g * h = \mathrm{F}^{-1}[\mathrm{F}[g]\mathrm{F}[h]]$$

# Frequency Spectra

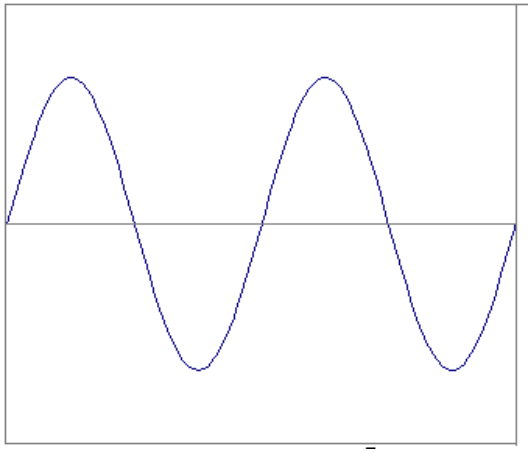
- Consider a square wave f(x)



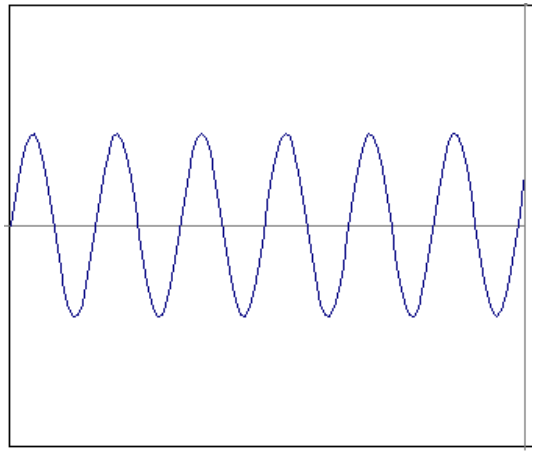
# Frequency Spectra



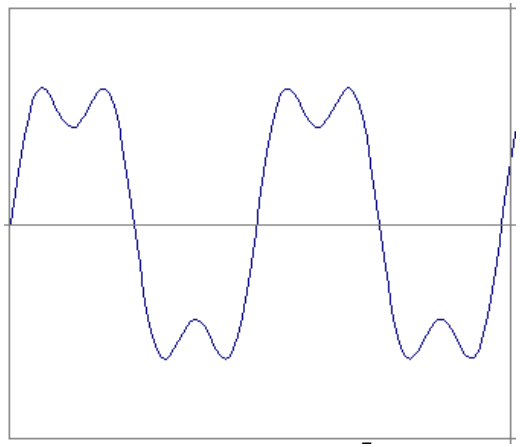
=



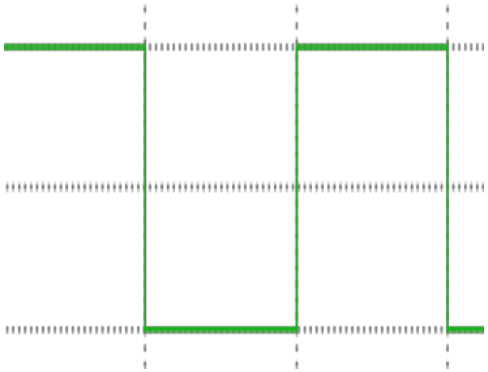
+



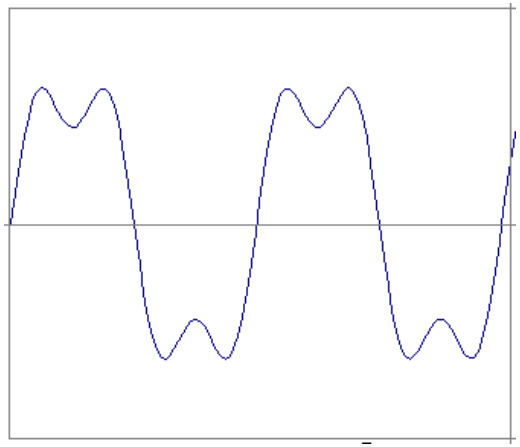
=



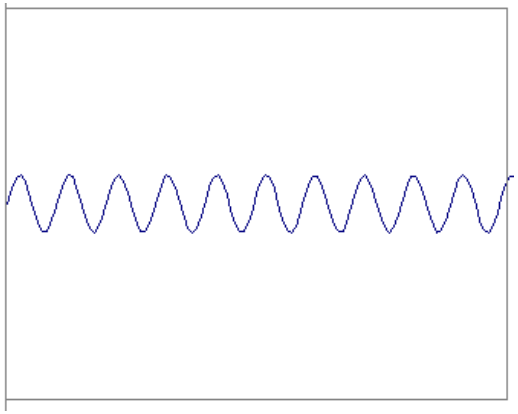
# Frequency Spectra



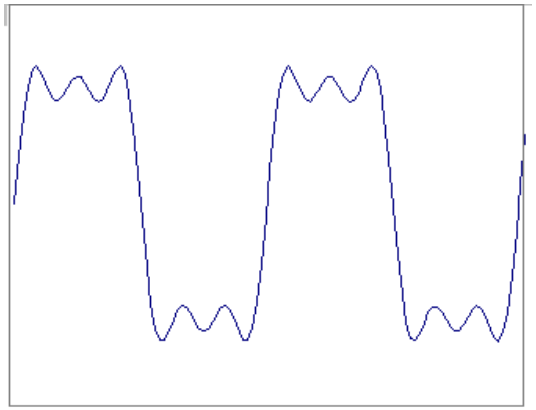
=



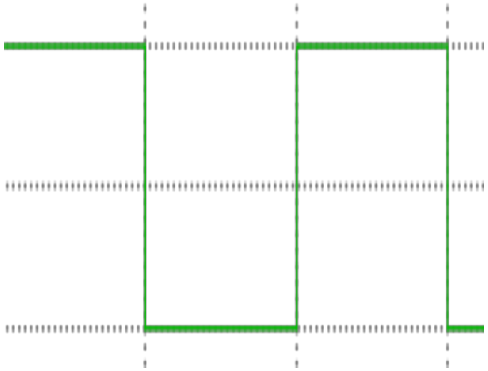
+



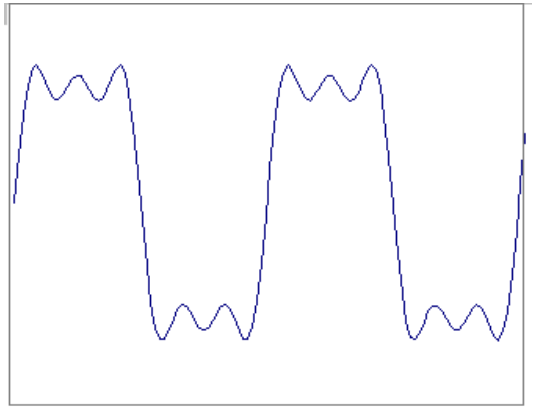
=



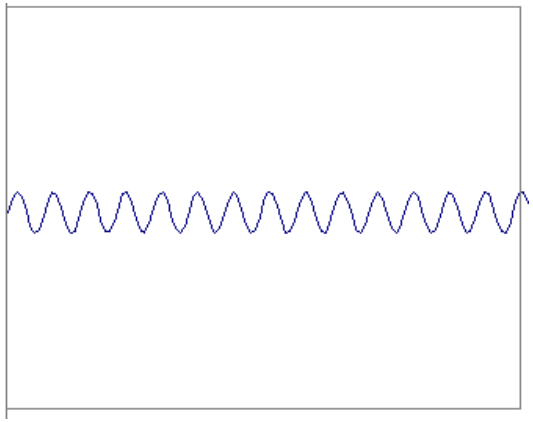
# Frequency Spectra



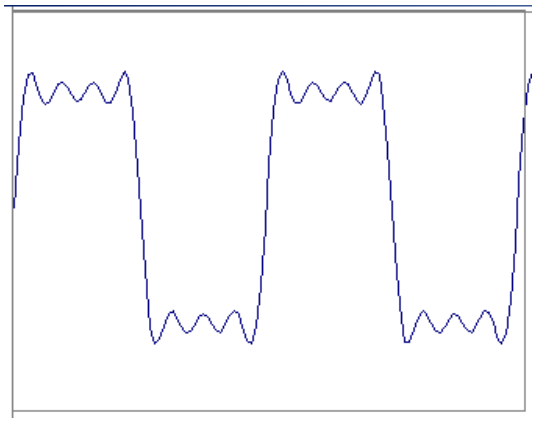
=



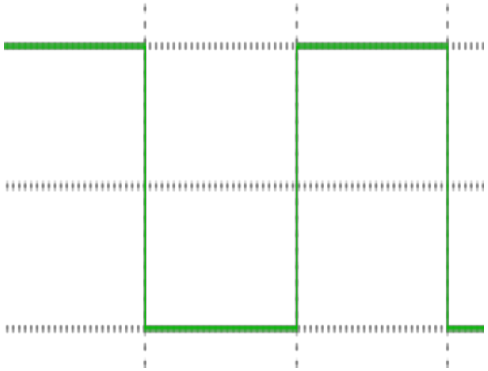
+



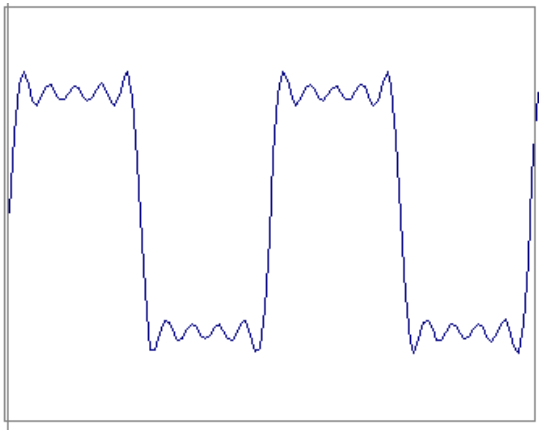
=



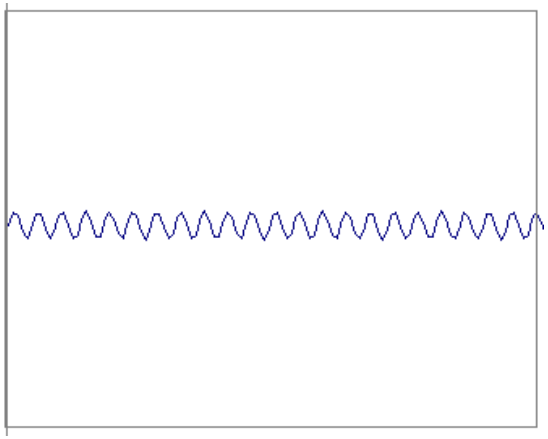
# Frequency Spectra



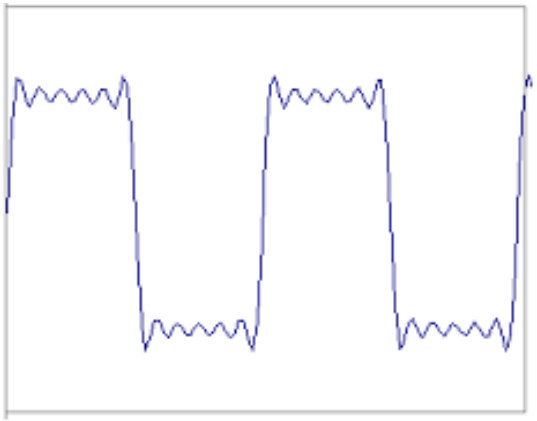
=



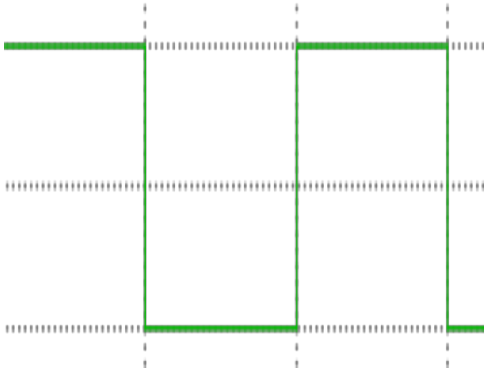
+

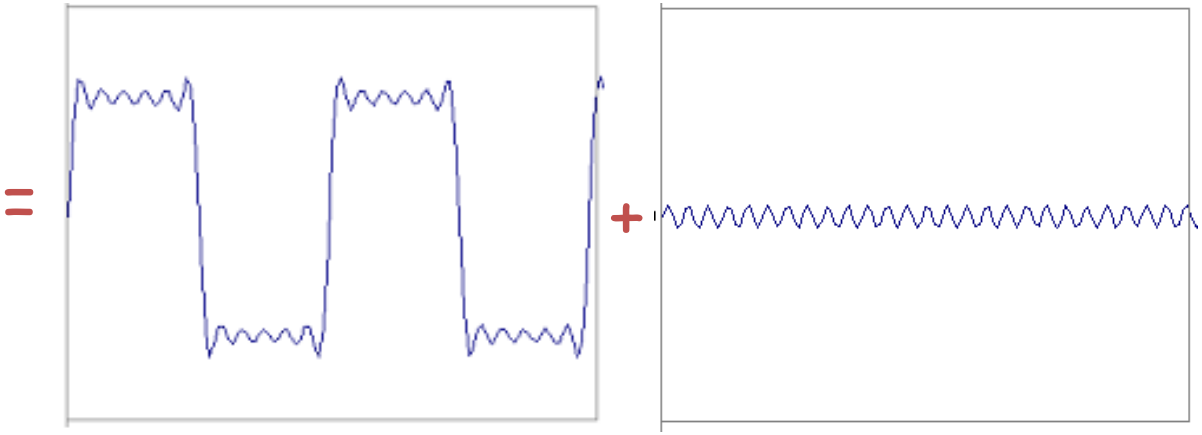


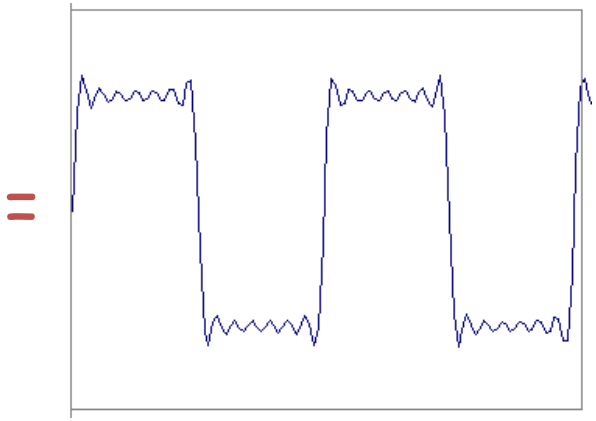
=



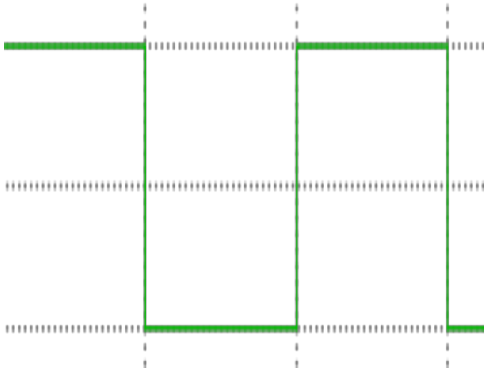
# Frequency Spectra



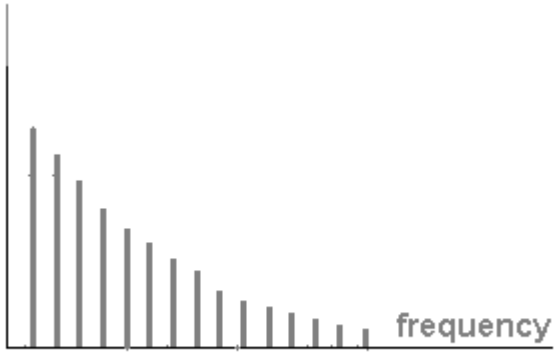




# Frequency Spectra



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



# Fourier Transform

- Fourier Transform

Inverse Fourier transform

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \qquad f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

Time f(t)  Frequency f(w)

What is transformation? It is a mapping between two domains

# Fourier Transform

- Let f(x) be a continuous function, then the FT is the function F(u) given by:

$$\mathrm{FT}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\Pi ux}\, dx$$

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

$$F(u).re = \int_{-\infty}^{\infty} f(x)\cos(-2\Pi ux)dx$$

Real part

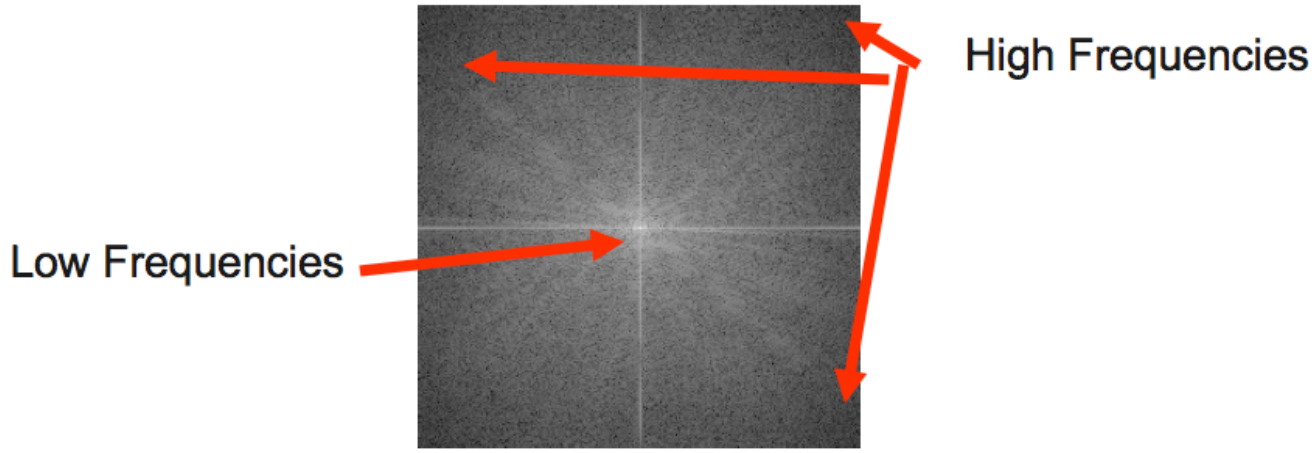
$$F(u).im = \int_{-\infty}^{\infty} f(x)\sin(-2\Pi ux)dx$$

Imaginary part

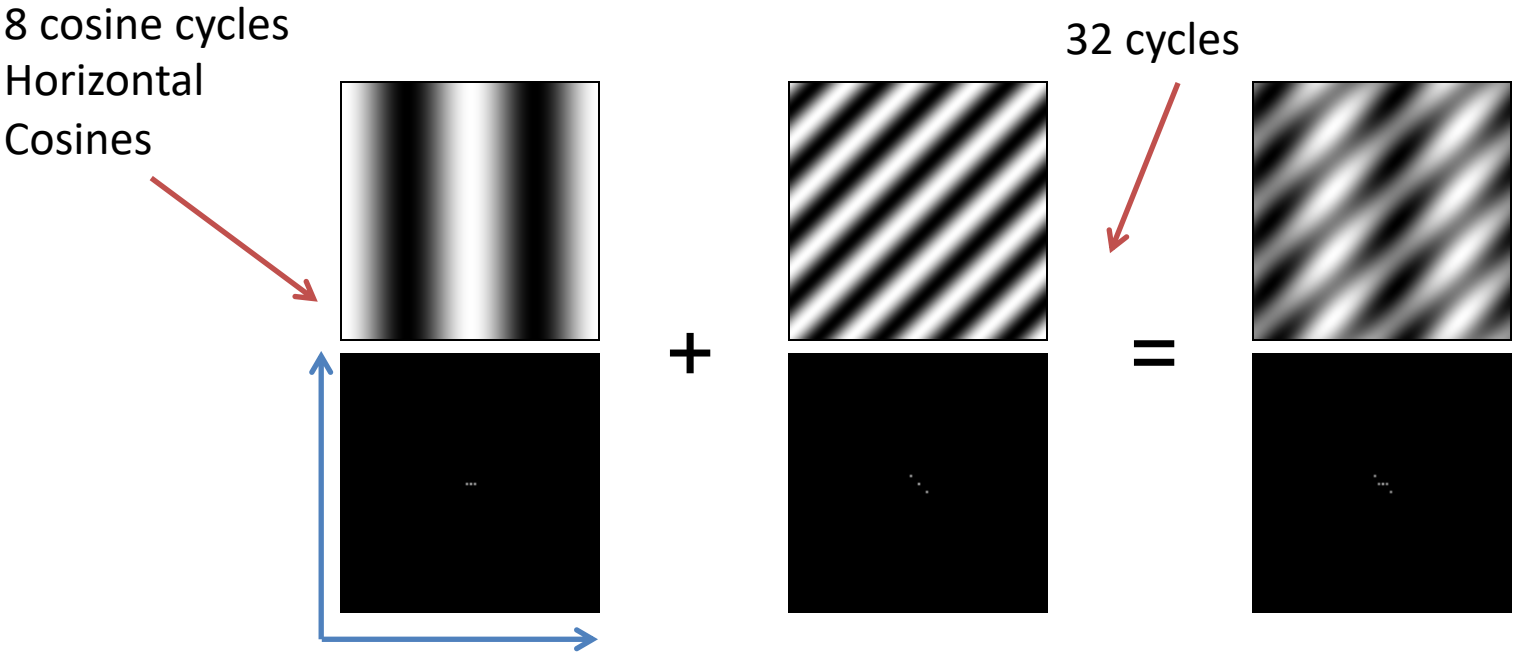
# Fourier Transform for image processing

# Fourier Transform

- Essentially, we are figuring out how to build the image out of these sinusoids of different frequency
- Each pixel in the transformed image corresponds to the amount of a sinusoid of a particular sinusoid that is needed



# Signals can be composed

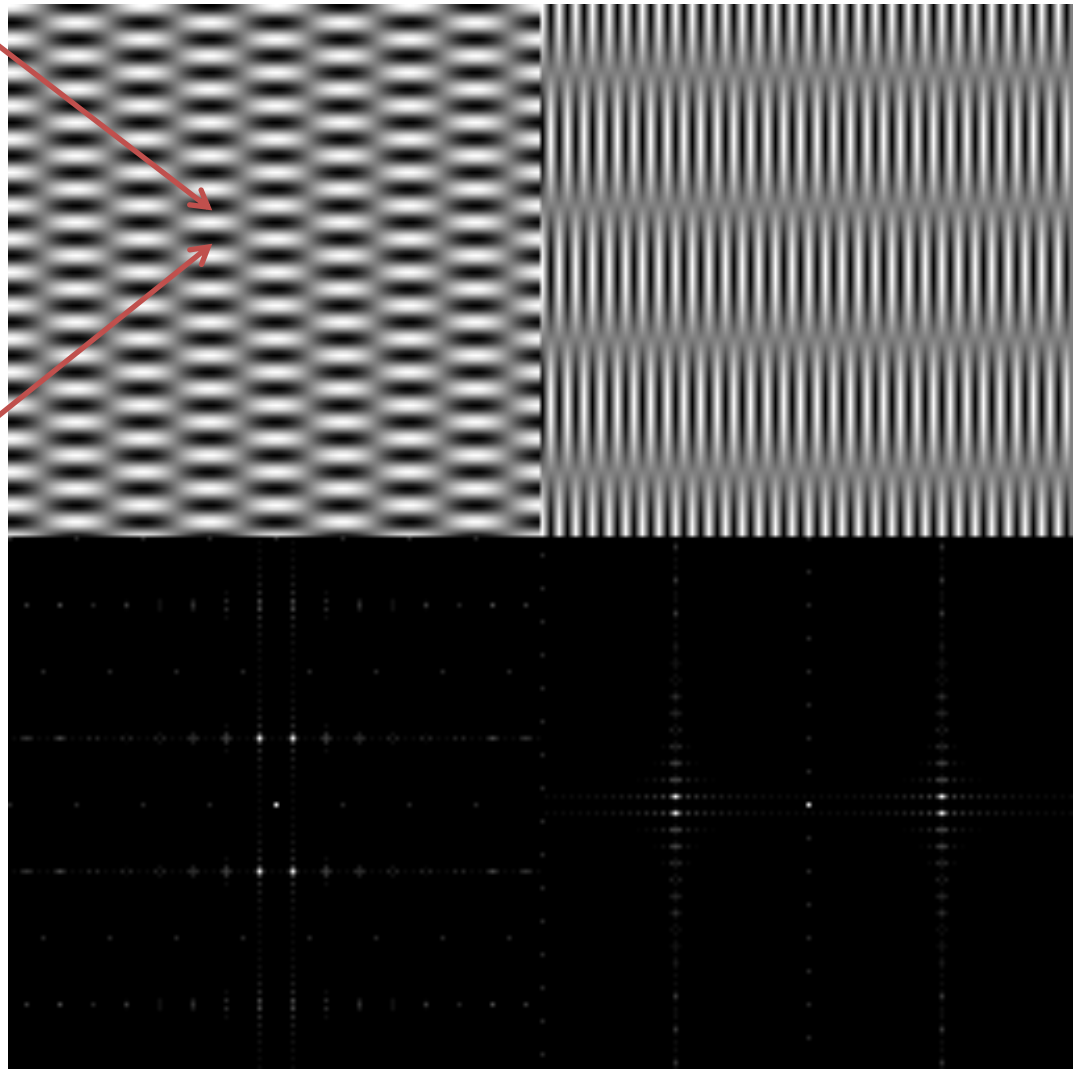



Dot at the center: zero frequency

More: http://www.cs.unm.edu/~brayer/vision/fourier.html

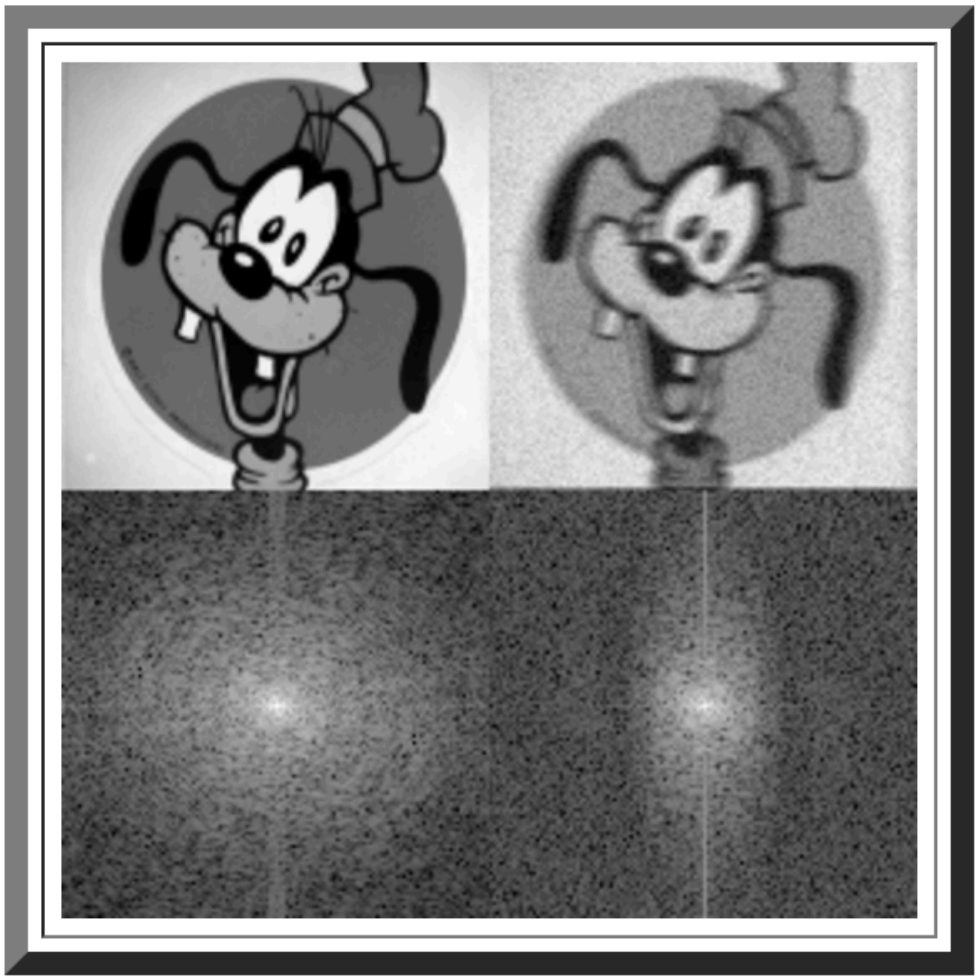
4 cycles
Horizontal

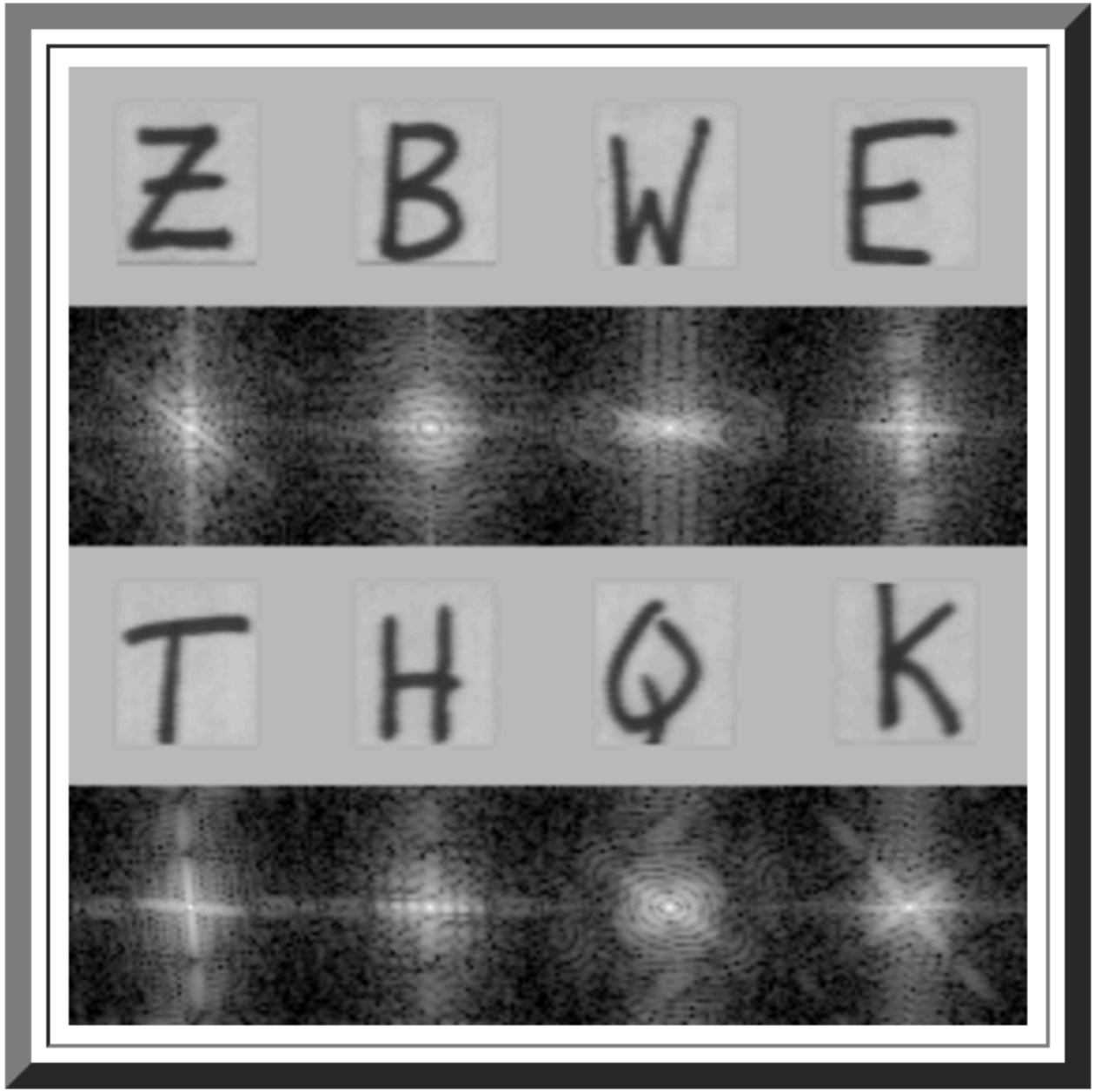
16 cycles
Vertical



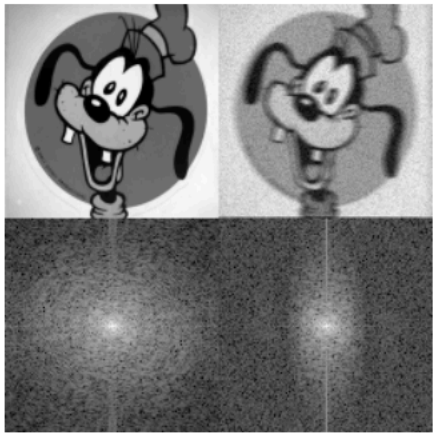
original

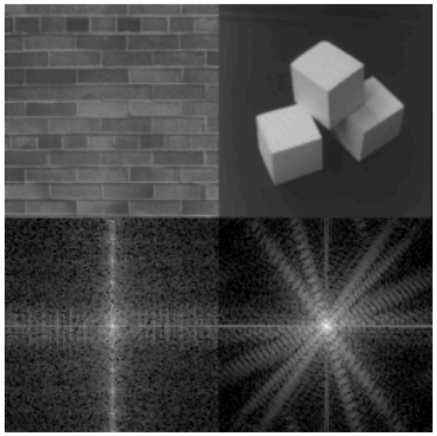
degraded



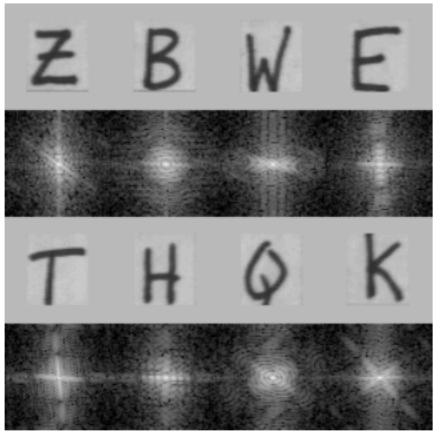

Z B W E
T H Q K

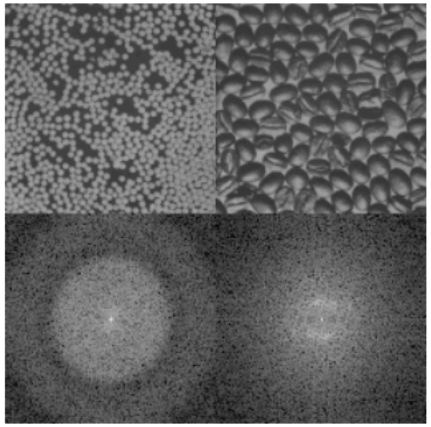
# FFT magnitude spectrum of images

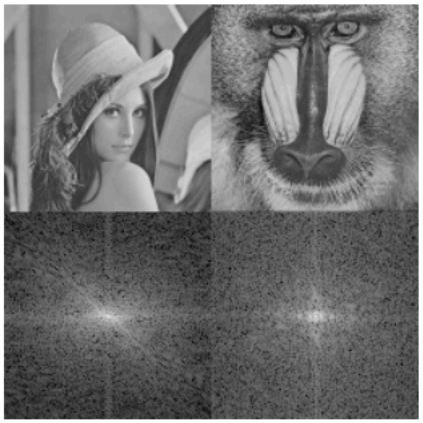




- notice a bright band going to high frequencies perpendicular to the strong edges in the image
- Anytime an image has a strong-contrast, sharp edge the gray values must change very rapidly. It takes lots of high frequency power to follow such an edge so there is usually such a line in its magnitude spectrum.

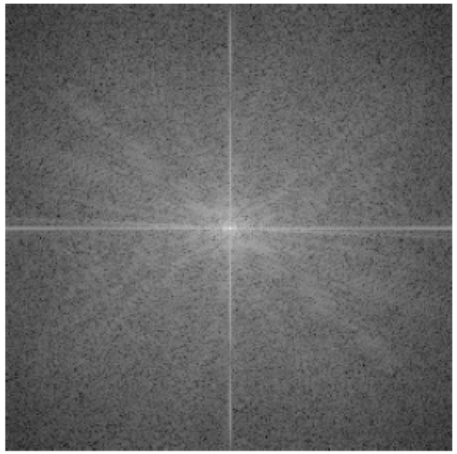


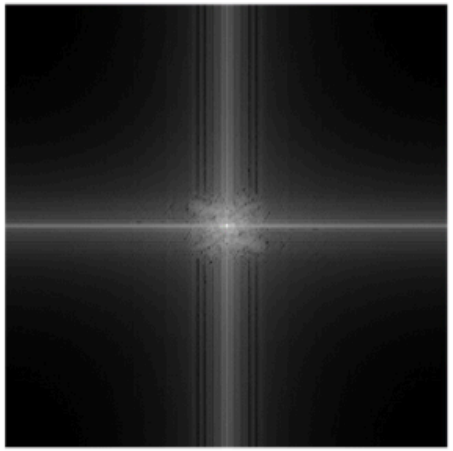


# What are the high frequencies?

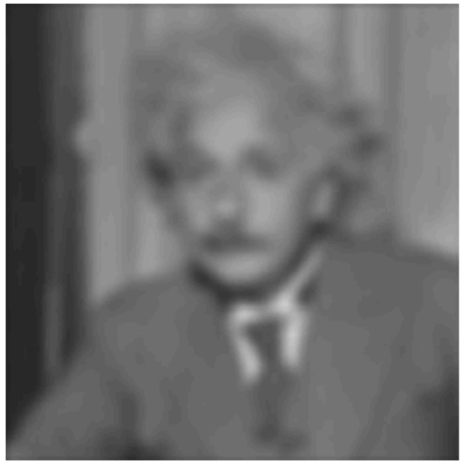
What if we remove the high frequencies?



Old Spectrum



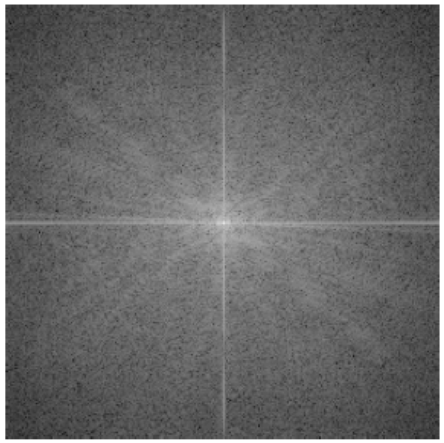
New Spectrum



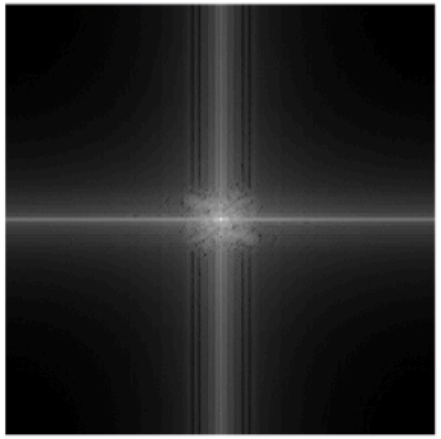
How will the new image look?

# What are the high frequencies?

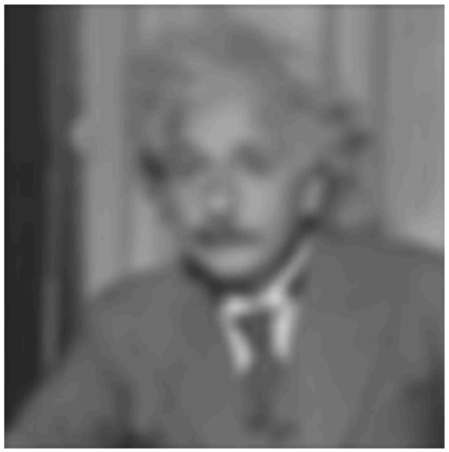
Removing the high frequencies makes the image look blurry



Old Spectrum



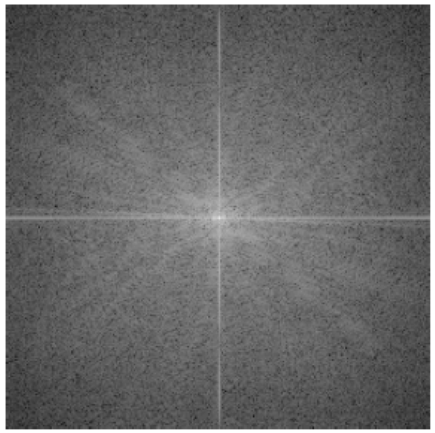
New Spectrum



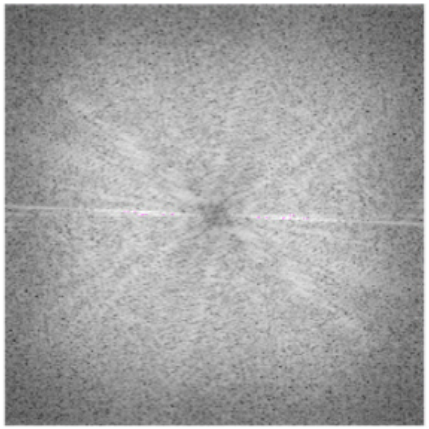
Try building a sharp edge out of low-frequency sinusoids

# What are the low frequencies?

What if we remove the low frequencies?



Old Spectrum

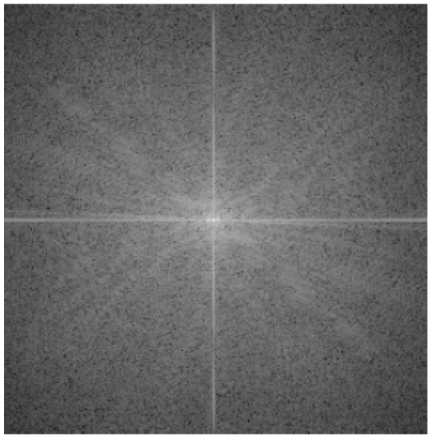


New Spectrum

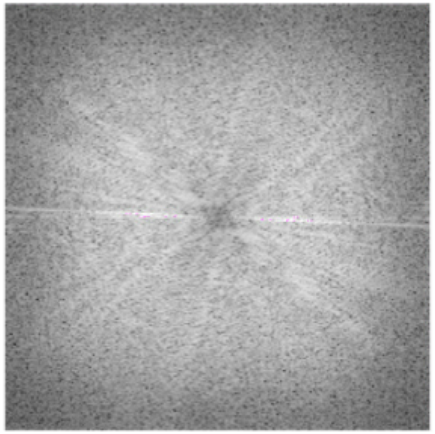
How will the new image look?

# What are the low frequencies?

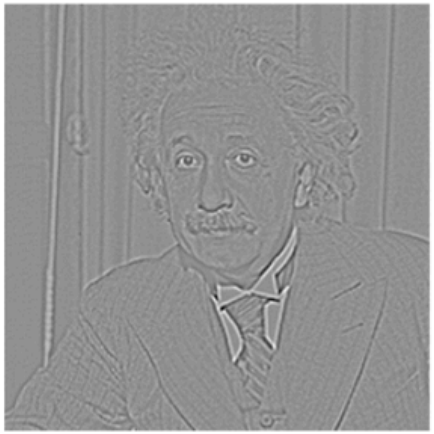
What if we remove the low frequencies?



Old Spectrum



New Spectrum



How will the new image look?

# 2D discrete Fourier transform

- The discrete FT for a discrete signal f(x) with N values is given by:

$$F(u) = \frac{1}{N} \sum_{x=0..N-1} f(x) e^{-i 2\Pi ux / N}$$

$$F(u).re = \frac{1}{N} \sum_{x=0..N-1} f(x) \cos(-2\Pi ux / N)$$

$$F(u).im = \frac{1}{N} \sum_{x=0..N-1} f(x) \sin(-2\Pi ux / N)$$

# Working with DFT (Discrete Fourier Transform)

- Is the complex part bothering you yet?
- Let's look at a different representation
- Every complex number can also be represented as: $Z = x+jy = re^{j\theta}$

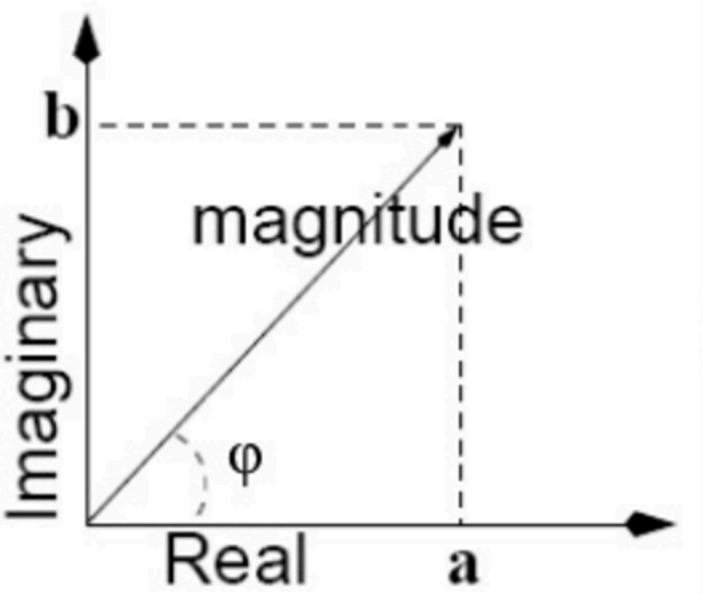
- r- magnitude (real number), r = abs(Z) or $\sqrt{x^2 + y^2}$

- θ-phase

# Phase and Magnitude

**MAGNITUDE** tells "how much" of a certain frequency component is present and the

**PHASE** tells "where" the frequency component is in the image. To illustrate this consider the following.




Magnitude: $\sqrt{x^2 + y^2}$

Phase: φ

# Phase and Magnitude

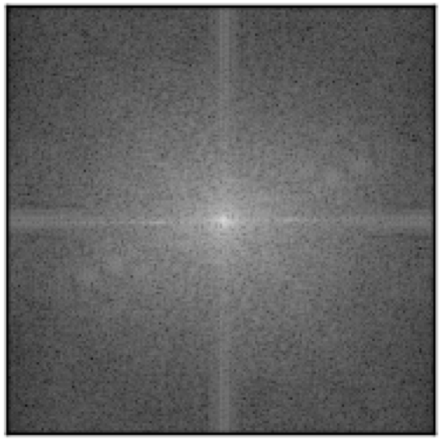
- Fourier transform of a real function is complex
  - Difficult to plot, visualize, instead we can think of phase and magnitude of the transform
- Phase is the phase of the complex transform
- Magnitude is the magnitude of the complex transform

- Curious fact
- All natural images have about the same magnitude transform Demonstration
- Take two pictures, swap the phase transform, compute the inverse, what does the result look like?

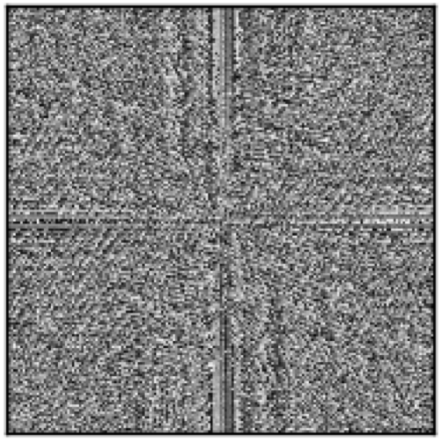
# Phase and Magnitude

Input Image

Log Magnitude Spectrum

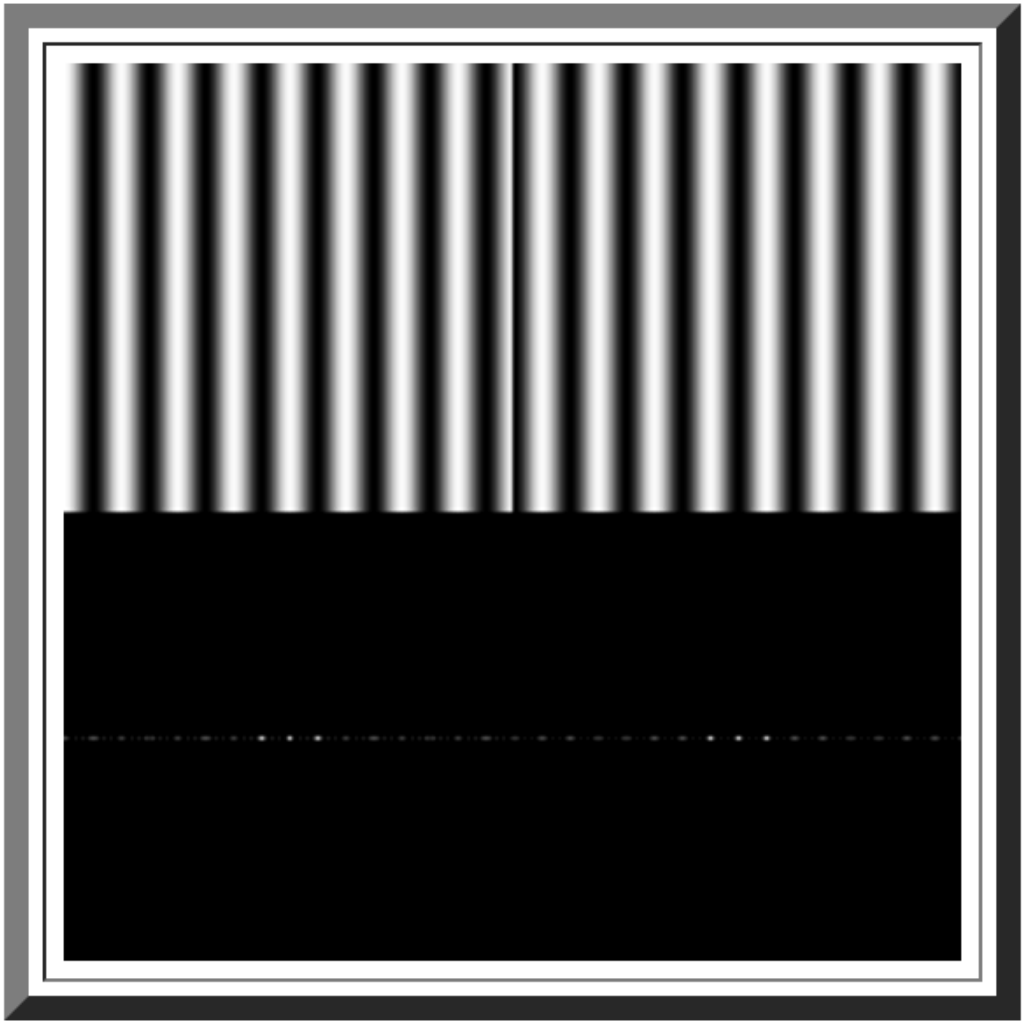


Phase



Not much new image, but

# Phase shifted images!

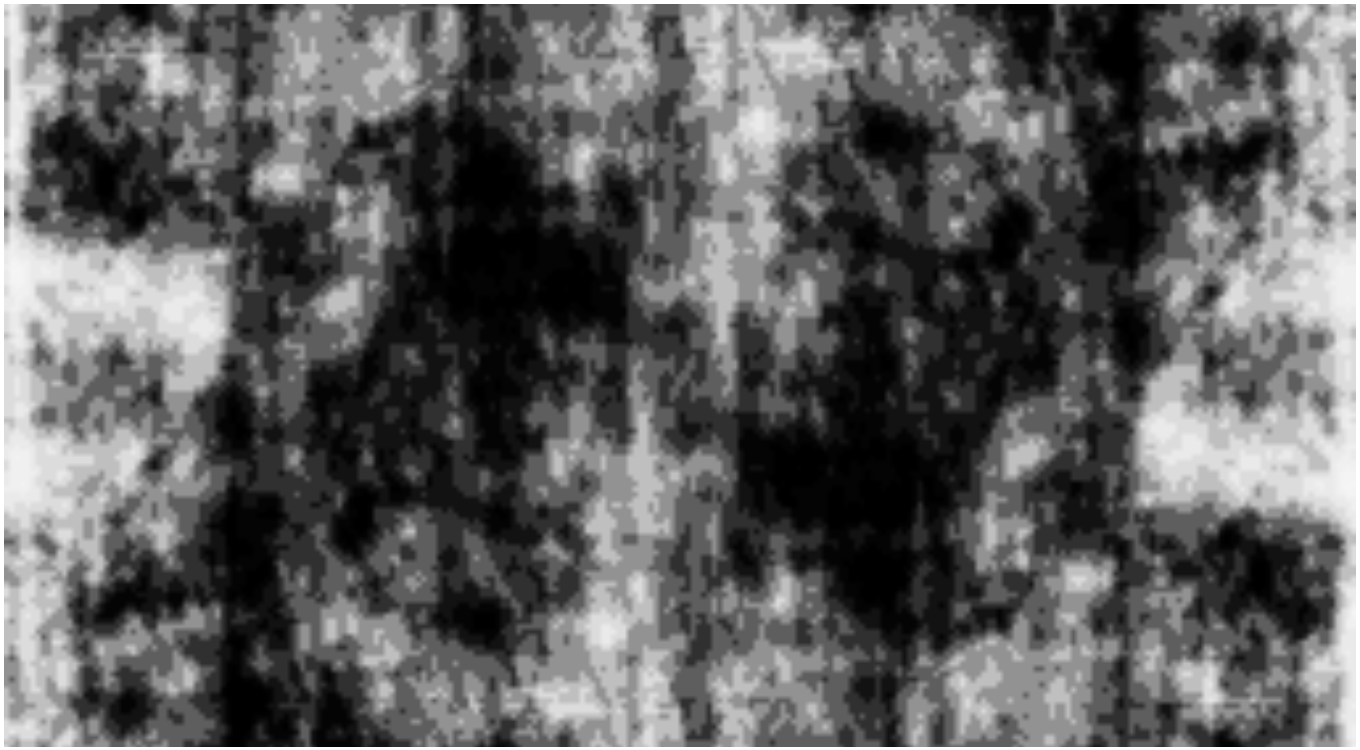


# What happens if you reverse Fourier without the phase?

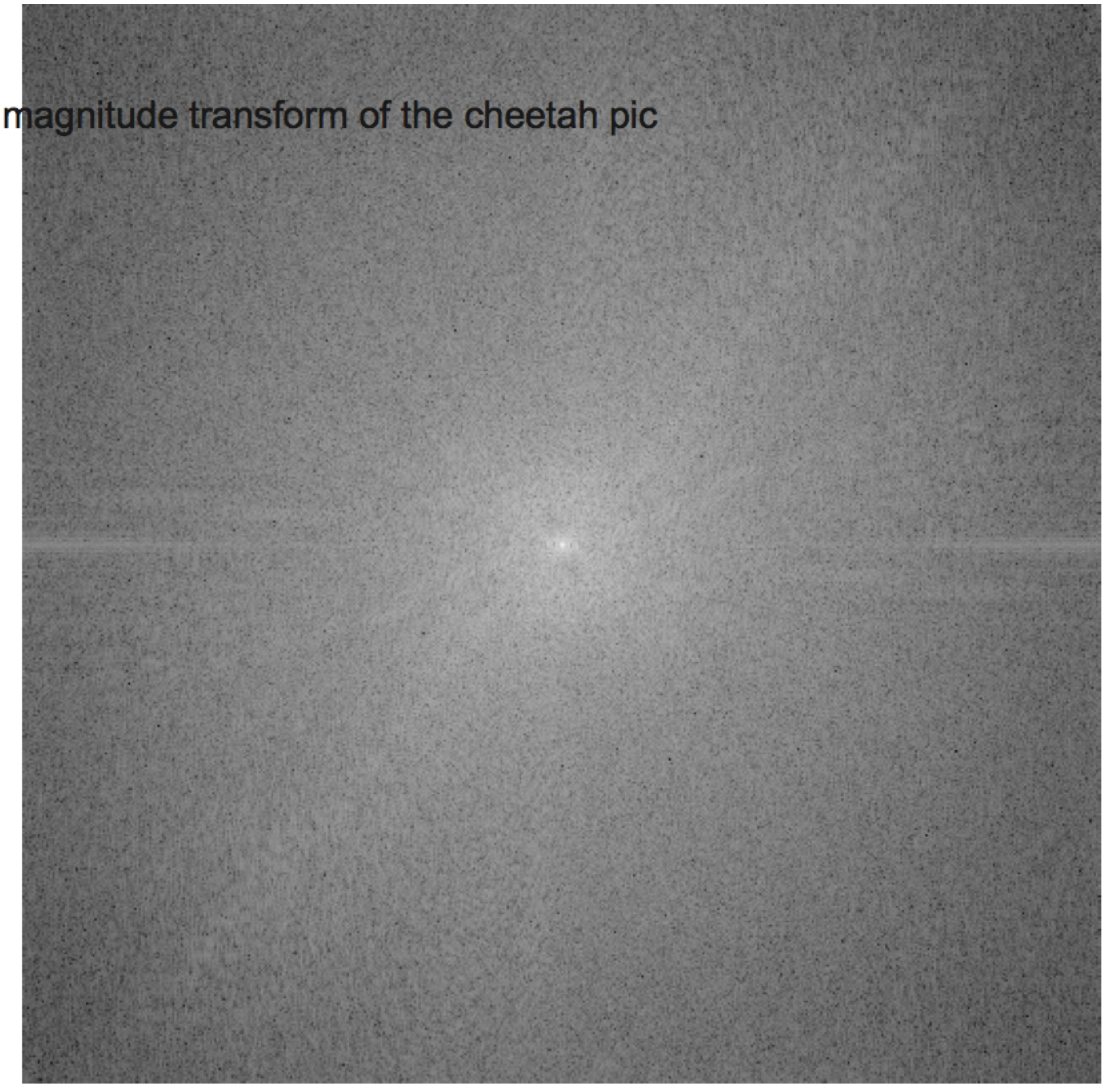
- Open an image
- Compute magnitude and phase of the image
- Reverse the ONLY magnitude back
- Image histogram equalize the reversed FFT image
- Display the image
- What happens?

# What happens if you reverse Fourier without the phase?

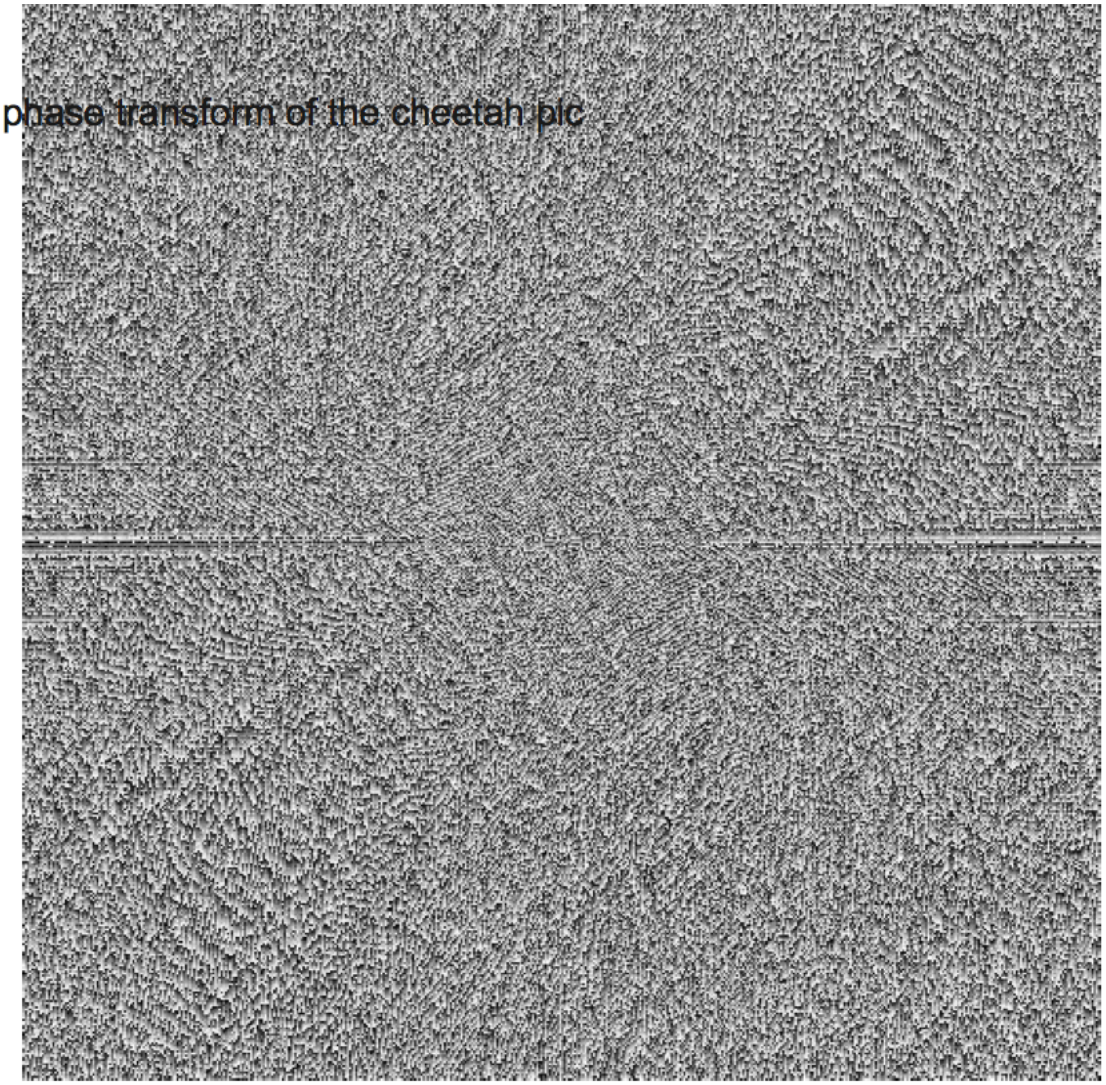
Same frequency, abstract nonsense



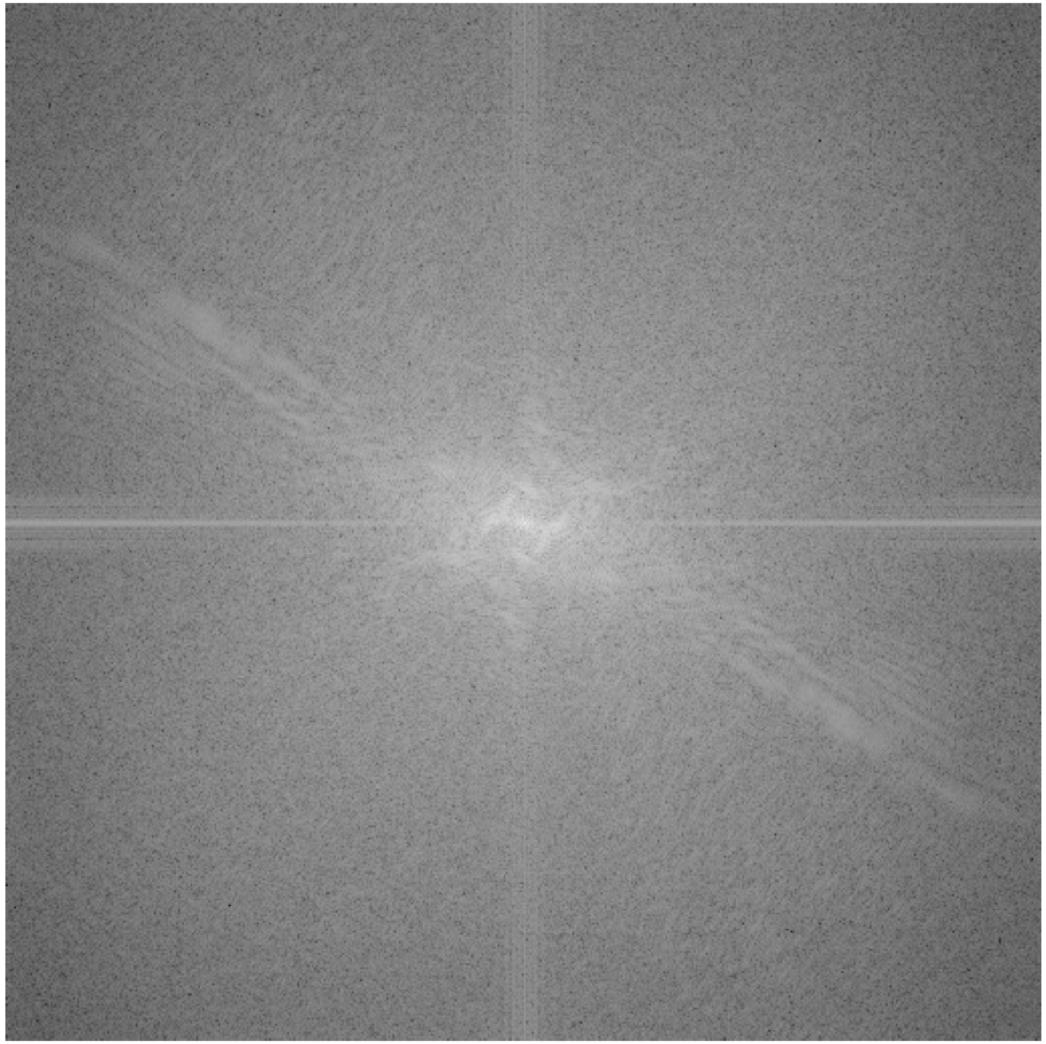
This is the magnitude transform of the cheetah pic



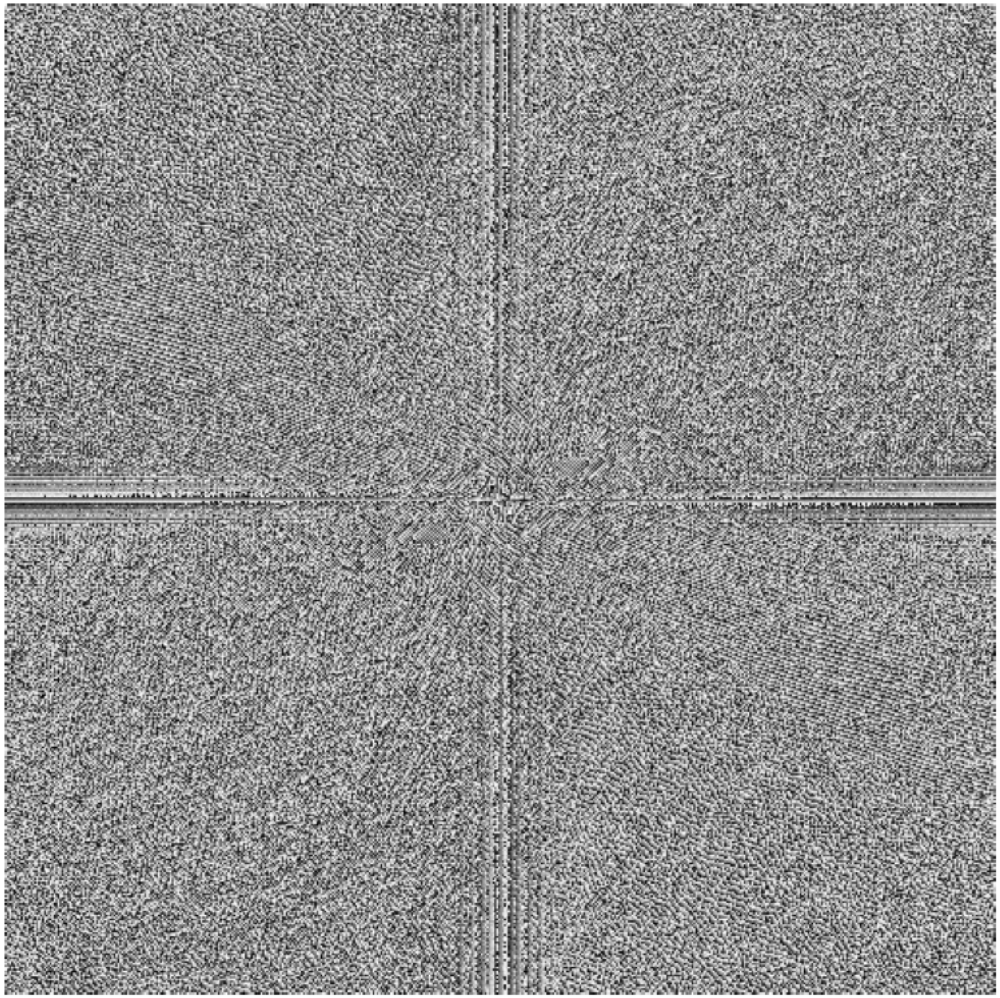
This is the phase transform of the cheetah pic



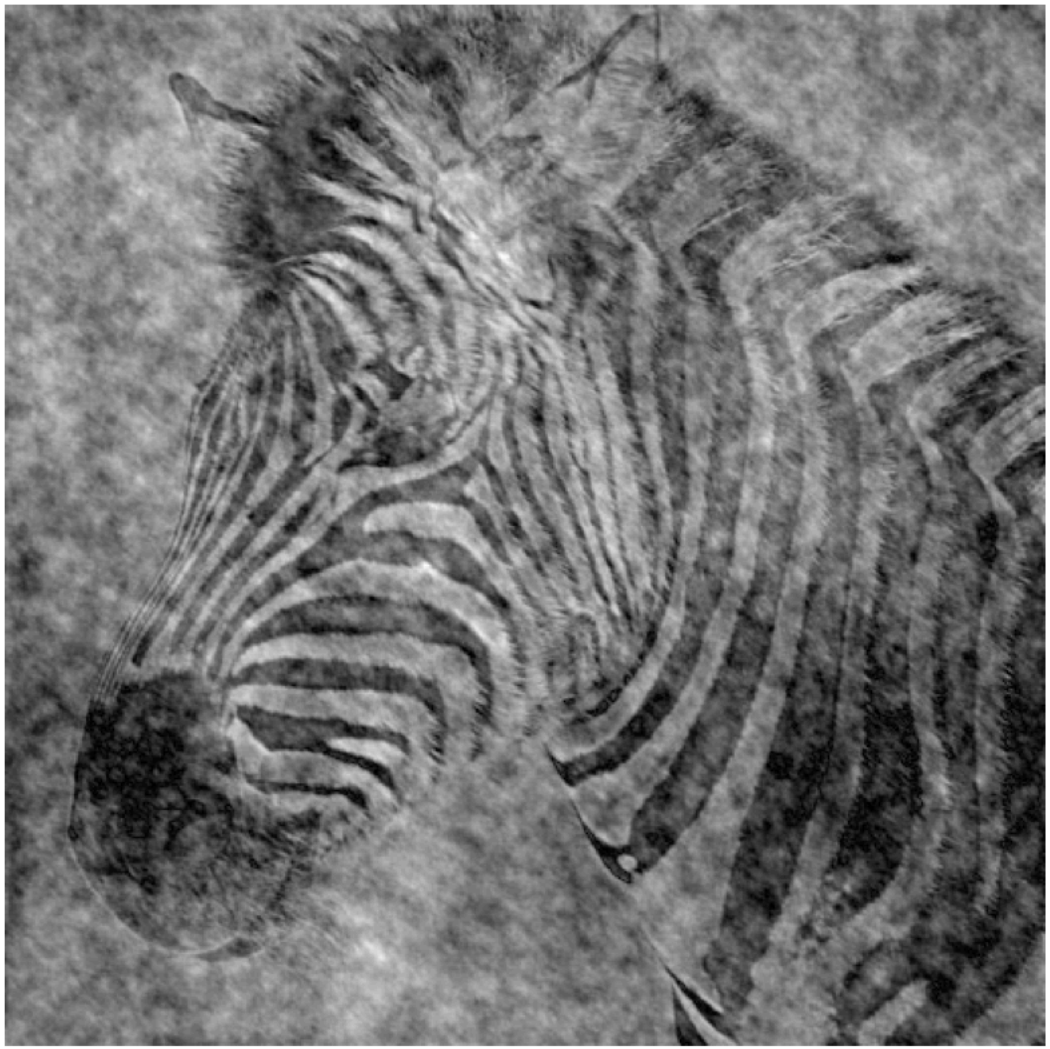
This is the magnitude transform of the zebra pic



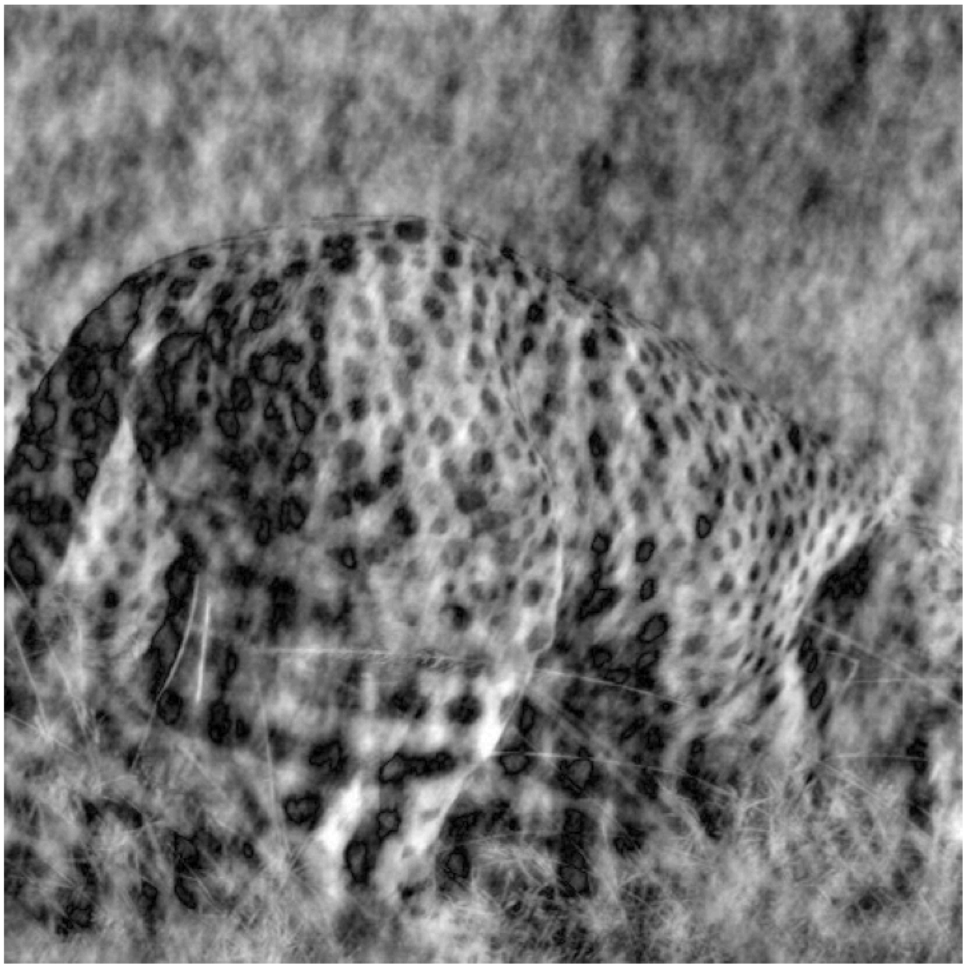
This is the phase transform of the zebra pic



Reconstruction with zebra phase, cheetah magnitude



Reconstruction with cheetah phase, zebra magnitude



# FFT in Python

FFT

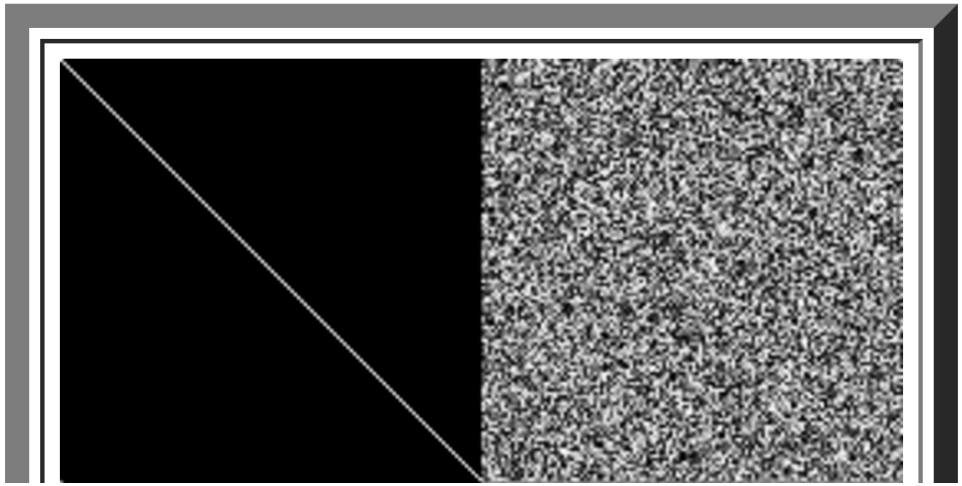
```
f = np.fft.fft2(img)
fshift = np.fft.fftshift(f)
magnitude_spectrum = 30*np.log(np.abs(fshift))
phase = np.angle(fshift)
```

Inverse FFT

```
rows, cols = img.shape
crow, ccol = rows/2 , cols/2 # center of the image
fshift[crow-5:crow+5, ccol-5:ccol+5] = 0
f_ishift = np.fft.ifftshift(fshift)
img_back = np.fft.ifft2(f_ishift)
```

# Exercise 0

- What is the FFT of the following images? Using Numpy to find out:



```
Z =
np.random.random((25
6,256))

plt.imshow(Z,
cmap='gray',
interpolation='nearest')
plt.show()
```