Thinking in frequencies

Nguyen Thi Oanh

Hanoi University of Science and Technology

oanhnt@soict.hust.edu.vn

SOICT, HUST, 2019

Content

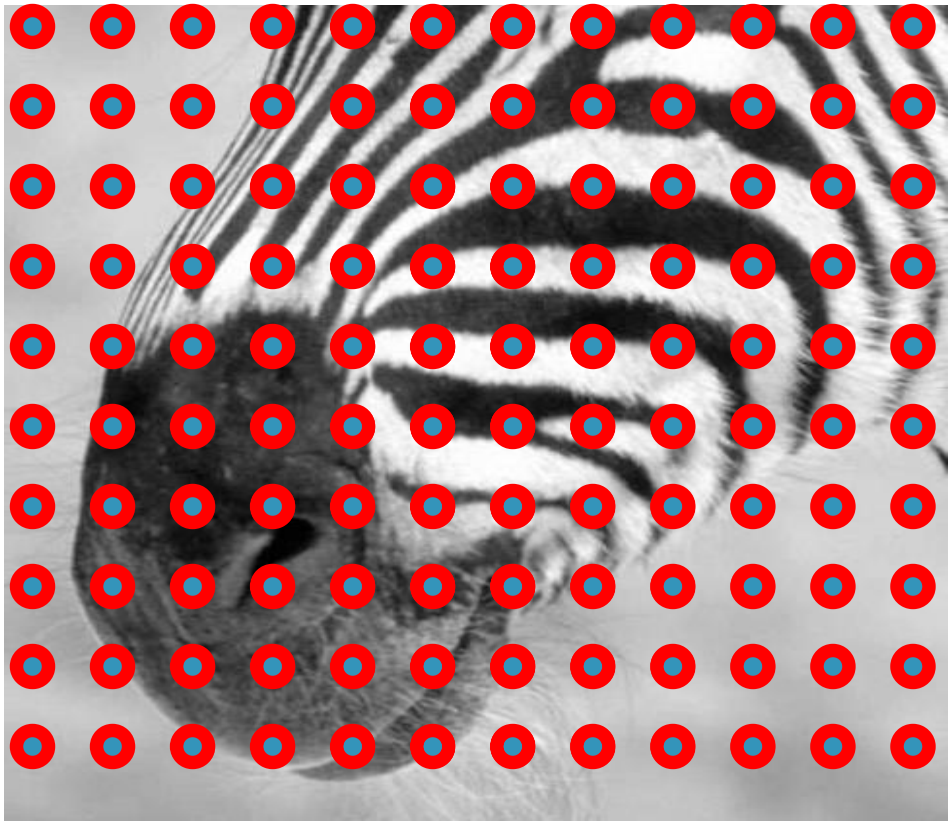
- Fsdfs
- dfsfsdfs

Sampling

- Why does a lower resolution image still make sense to us? What do we lose?

Image: http://www.flickr.com/photos/igorms/136916757/

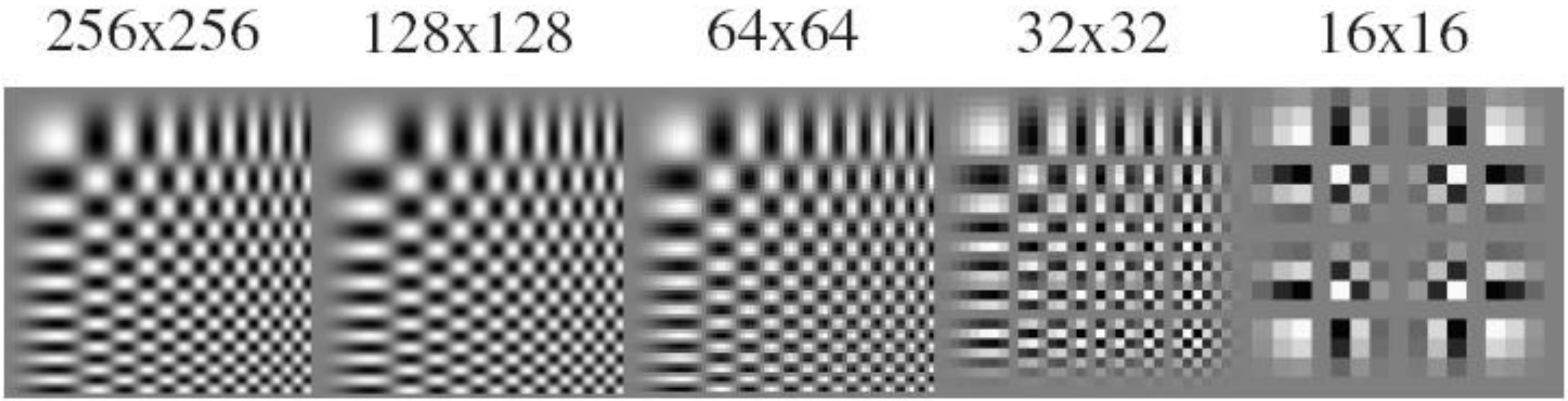
Subsampling by a factor of 2



Throw away every other row and column to create a 1/2 size image

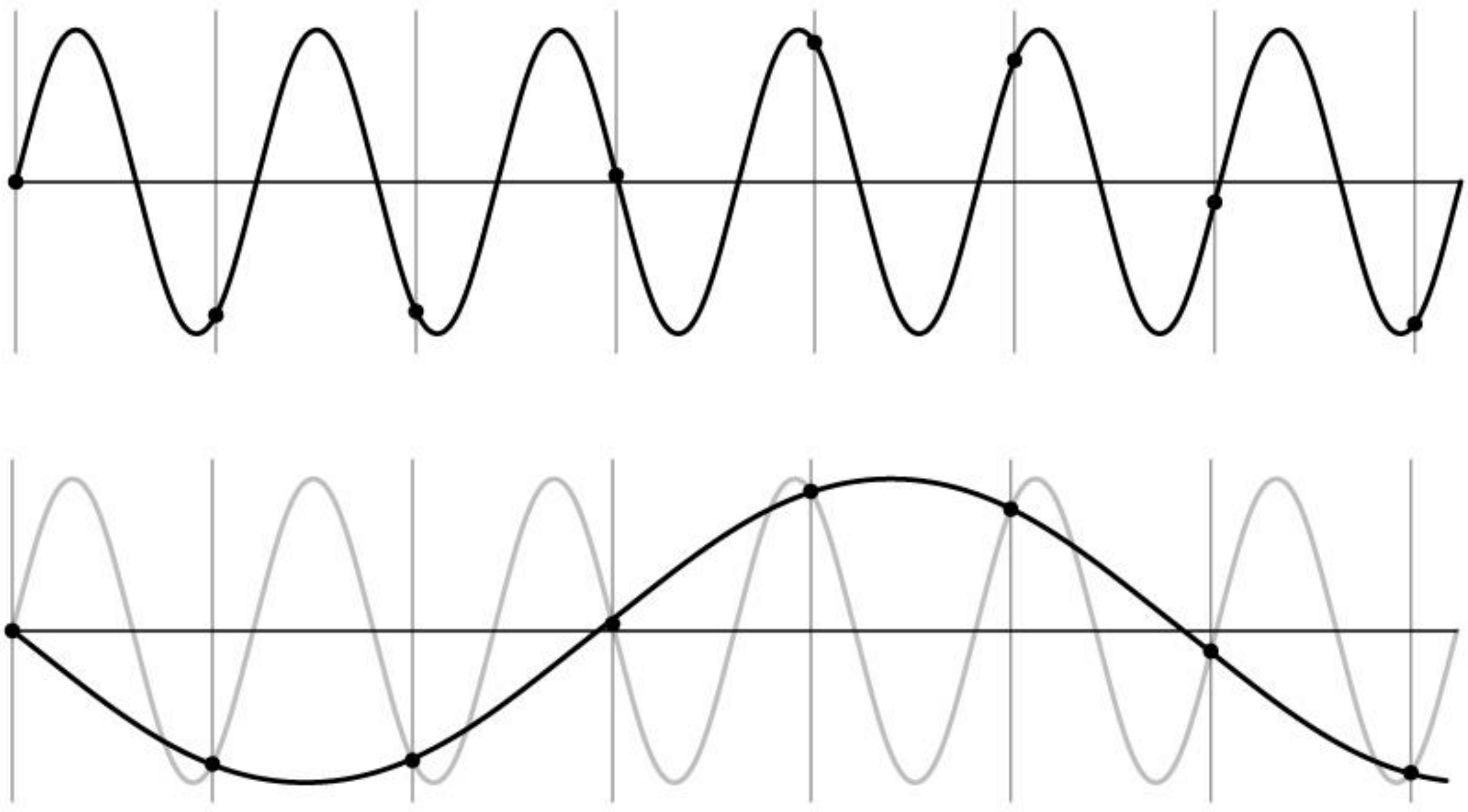
Source: CS131 - Standford

Sampling and aliasing



Aliasing problem

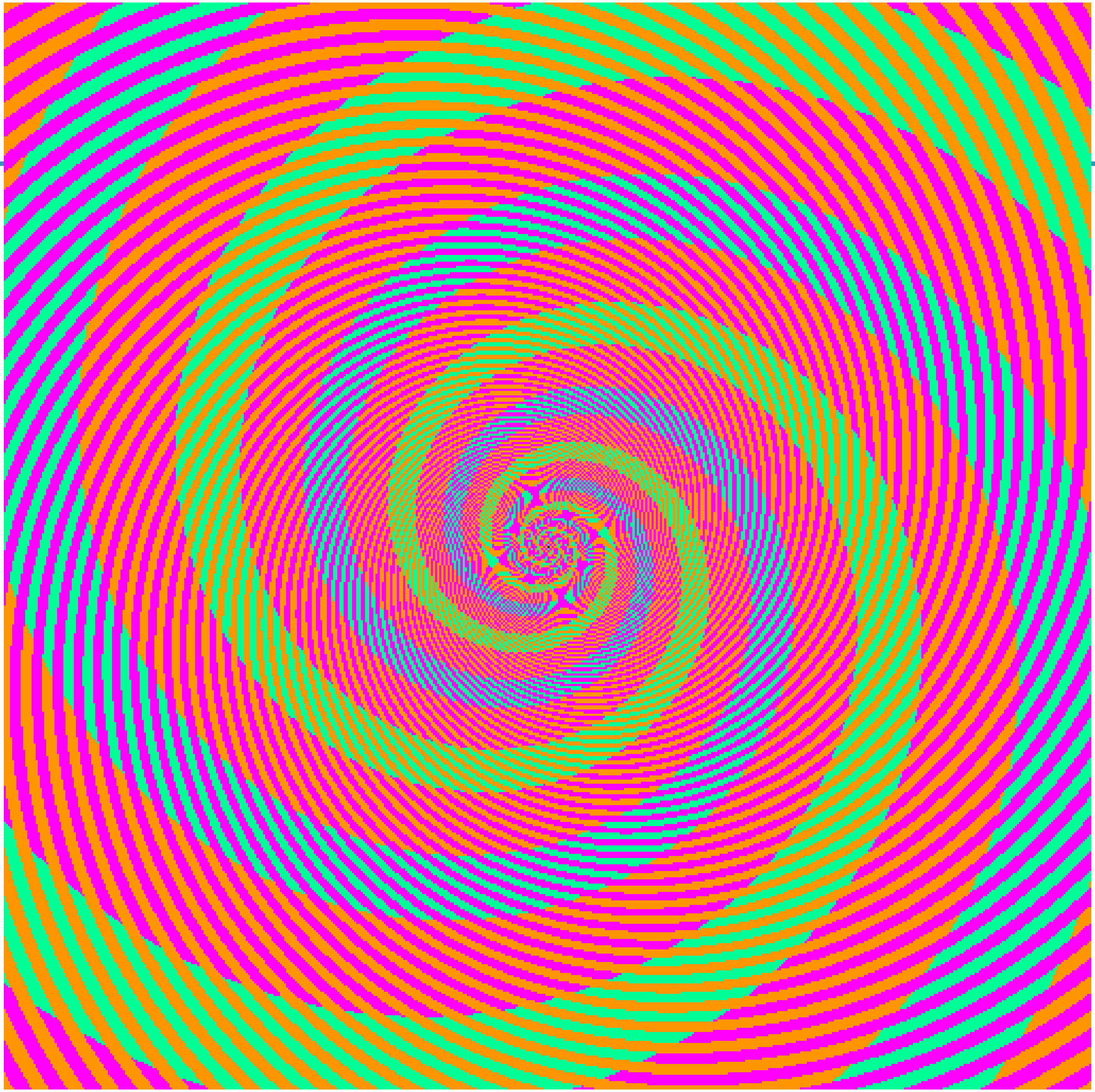
- 1D example (sinewave):

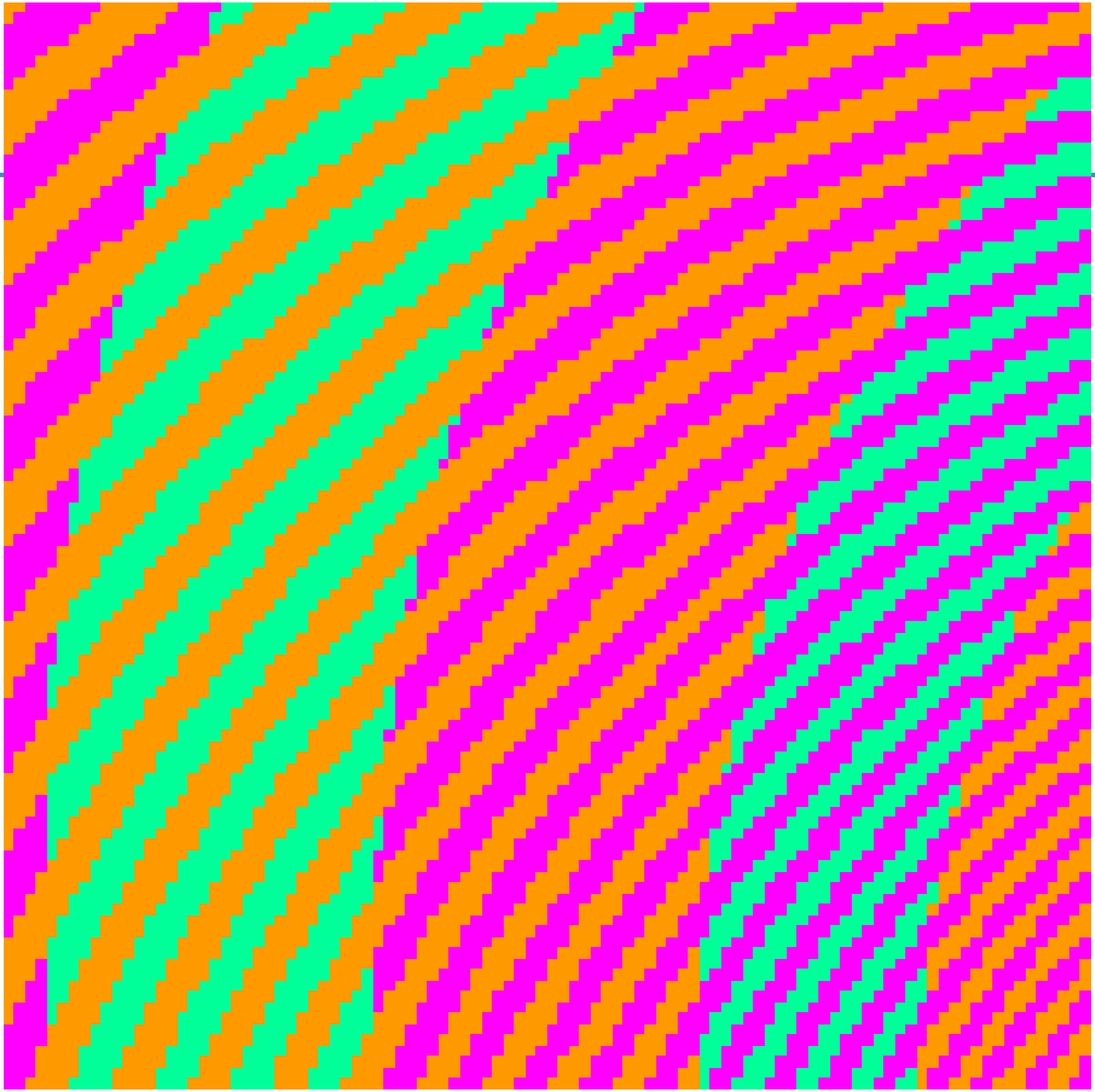


Source: S. Marschner

Aliasing in graphics

Source: A. Efros





The blue and green colors are actually the same
http://blogs.discovermagazine.com/badastronomy/2009/06/24/the-blue-and-the-green/

Alias in videos

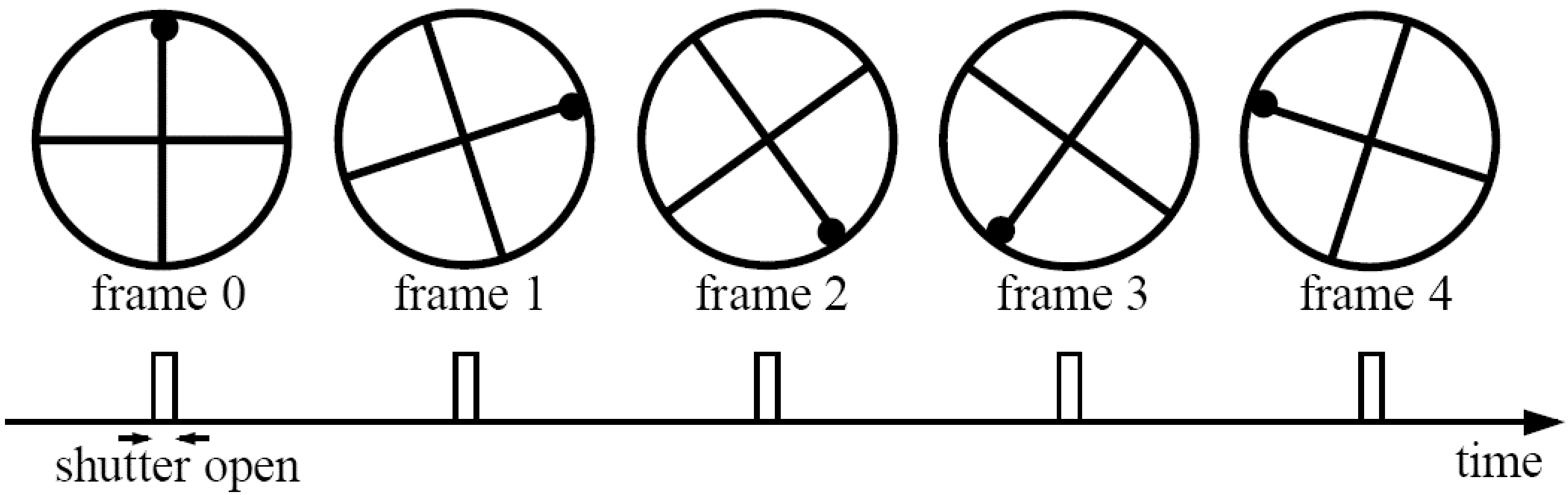
- https://www.youtube.com/watch?v=Y1yHMy0-4TM

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

Aliasing in video

Imagine a spoked wheel moving to the right (rotating clockwise). Mark wheel with dot so we can see what's happening.

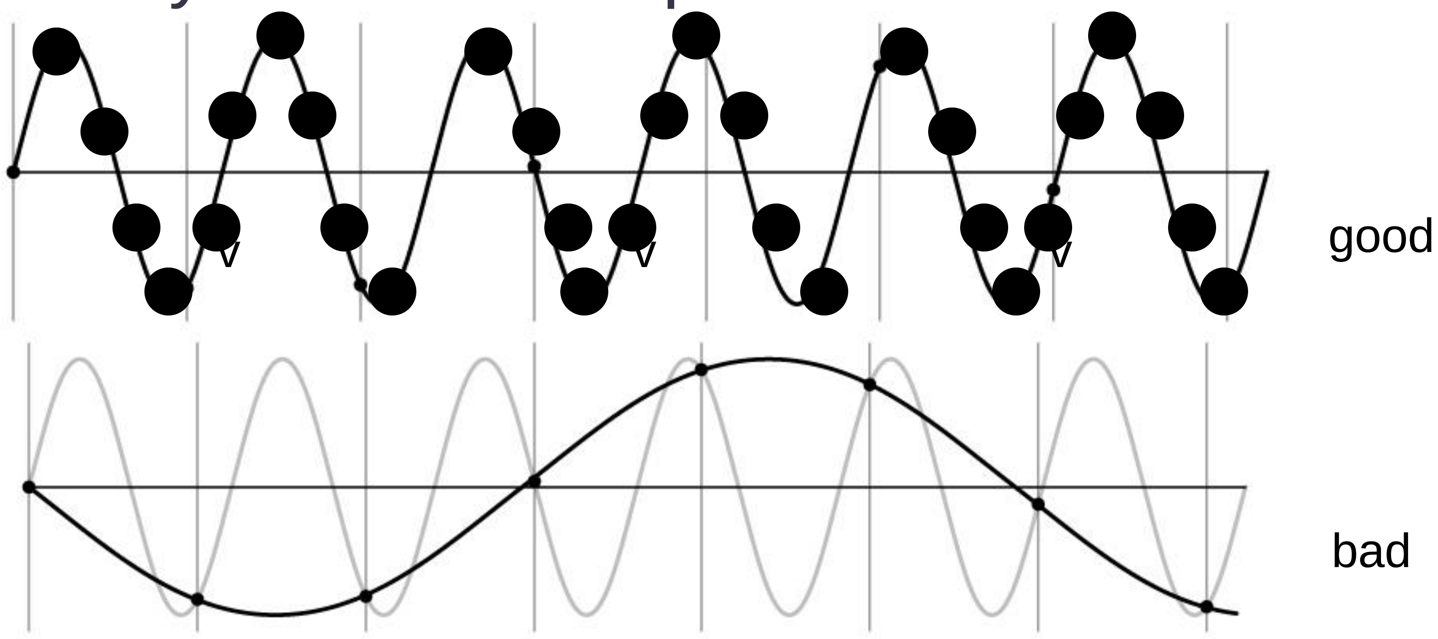
If camera shutter is only open for a fraction of a frame time (frame time = 1/30 sec. for video, 1/24 sec. for film):



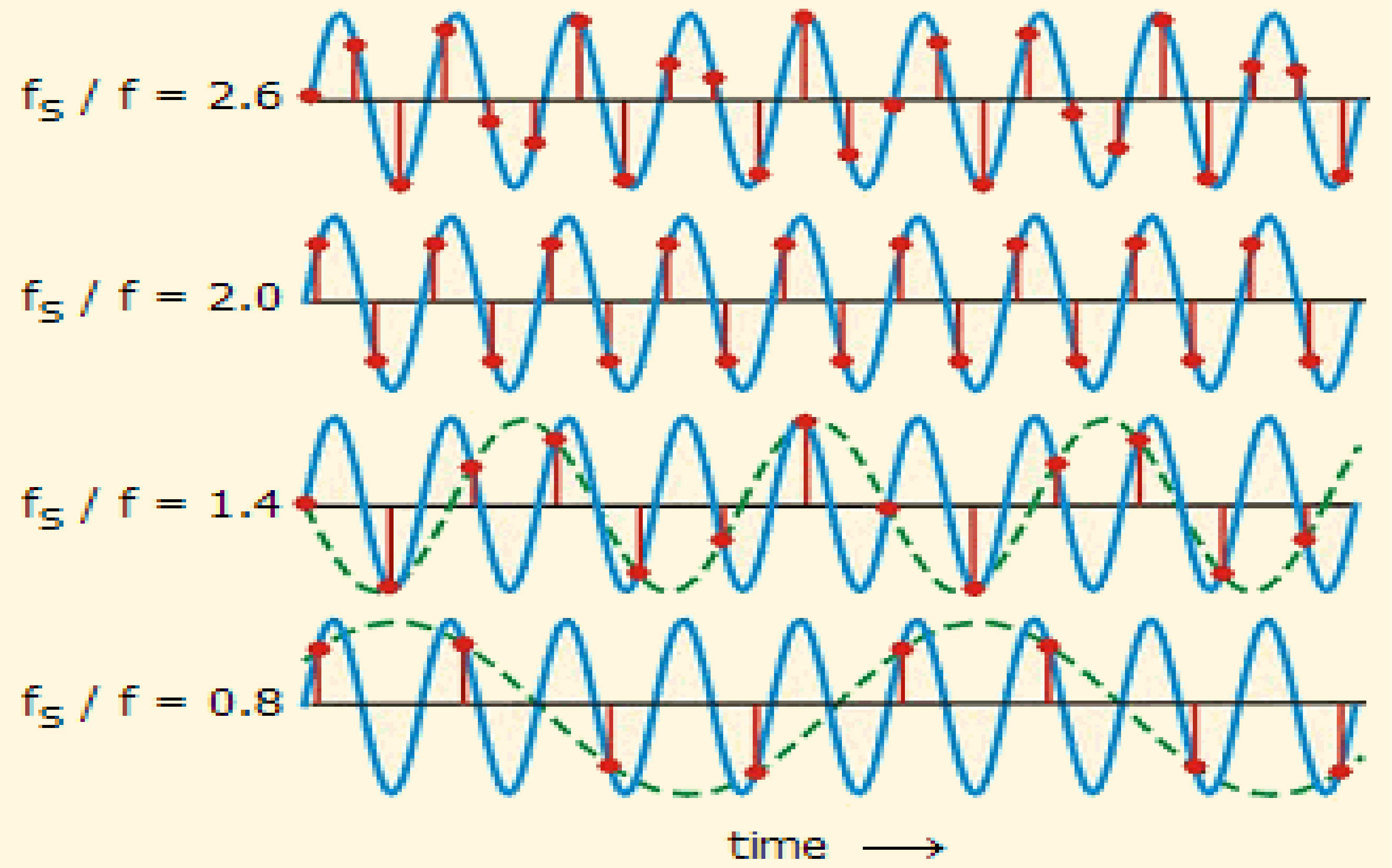
Without dot, wheel appears to be rotating slowly backwards! (counterclockwise)

Nyquist-Shannon Sampling Theorem

- When sampling a signal at discrete intervals, the sampling frequency must be $\geq 2 \times f_{max}$
- f_{max} = max frequency of the input signal
- This will allows to reconstruct the original perfectly from the sampled version



Nyquist-Shannon Sampling Theorem

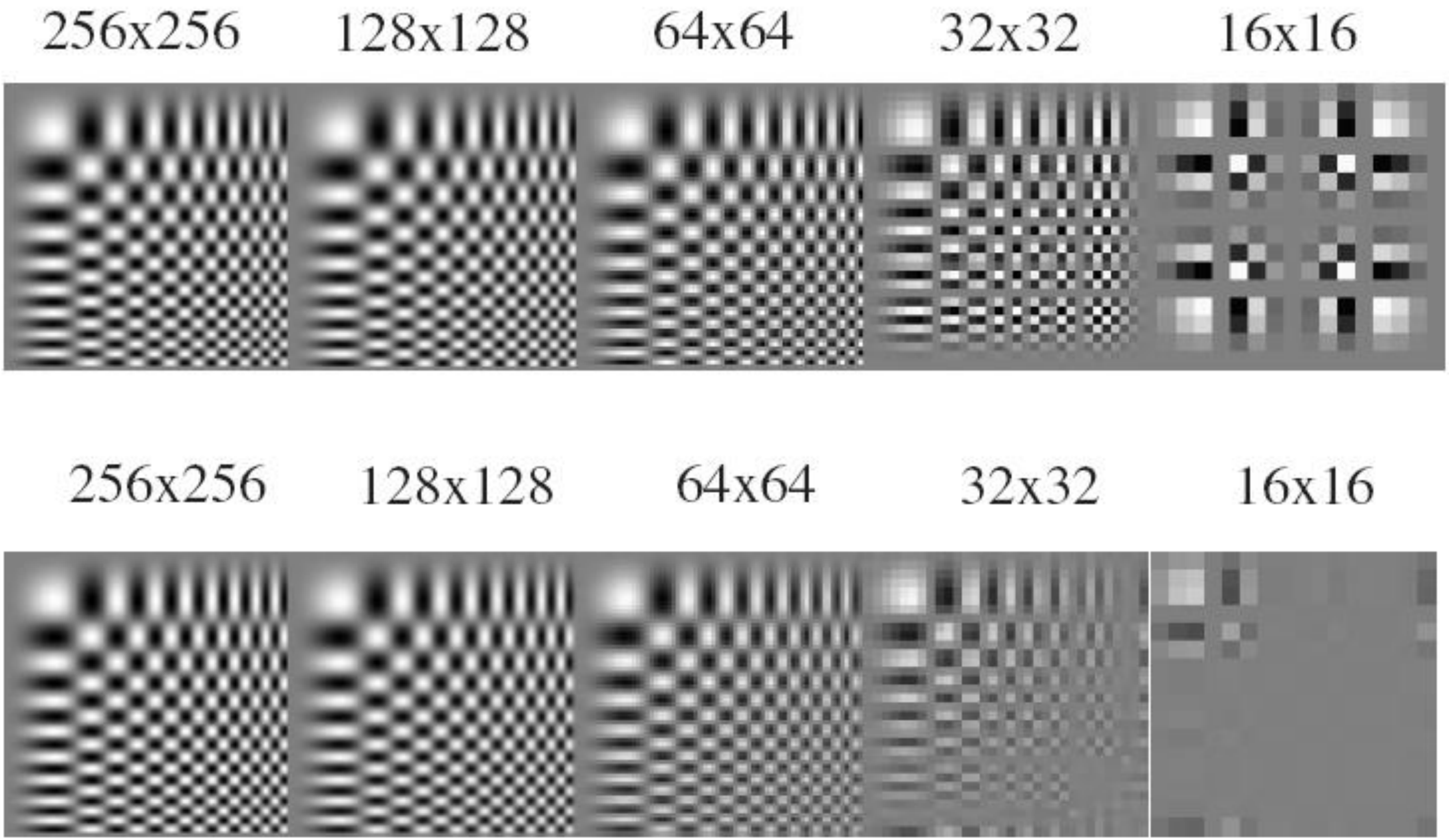


Sampling of a sinusoidal signal of frequency f at different sampling rates fS.
With dashed lines are shown the alias frequencies, occurring when $f_s/f < 2$

How to fix aliasing?

- Better sensors
- Anti-aliasing:
  - Sample more often
  - Get rid of all frequencies that are greater than half the new sampling frequency
    - Will lose information
    - But it’s better than aliasing
    - Apply a **smoothing (*low pass*) filter**

Anti-aliasing

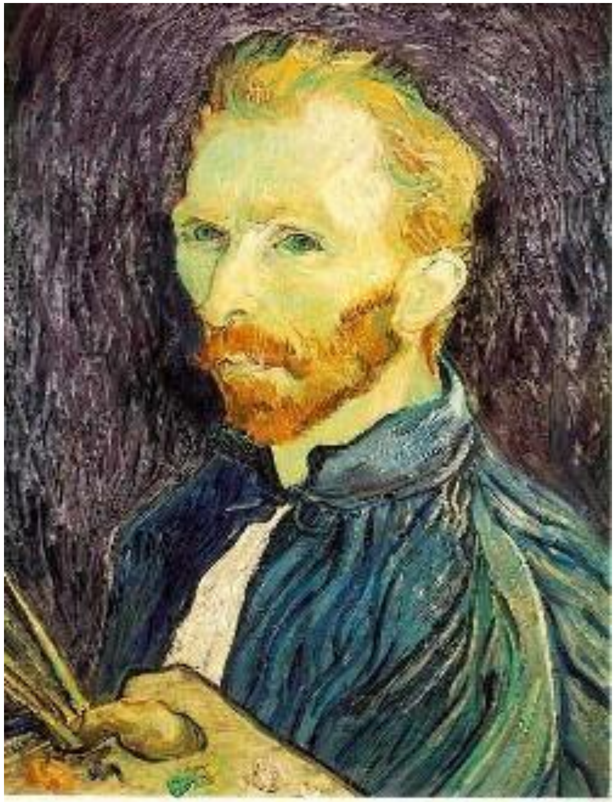


Forsyth and Ponce 2002

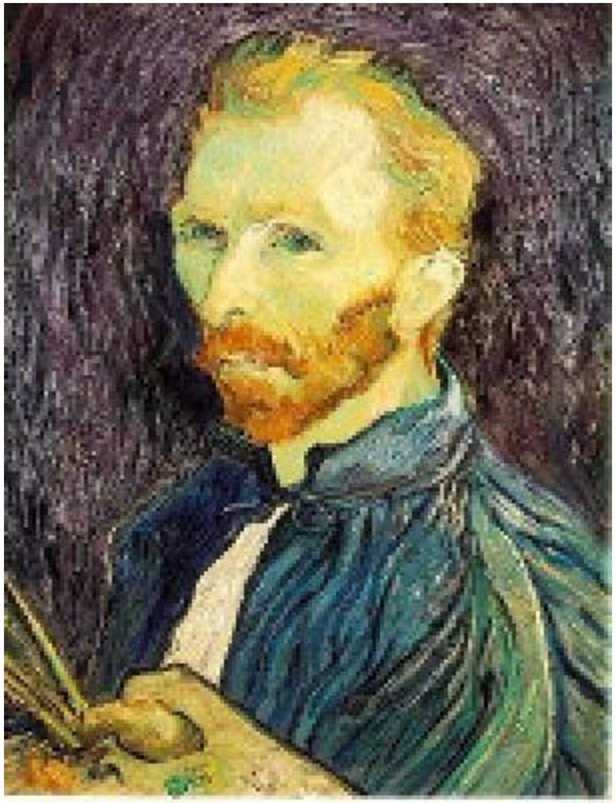
Algorithm for downsampling by factor of 2

1. Start with image(h, w)
2. Apply low-pass filter
   - im_blur = imfilter(image, fspecial('gaussian', 7, 1))
3. Sample every other pixel
   - im_small = im_blur(1:2:end, 1:2:end);

Subsampling without pre-filtering



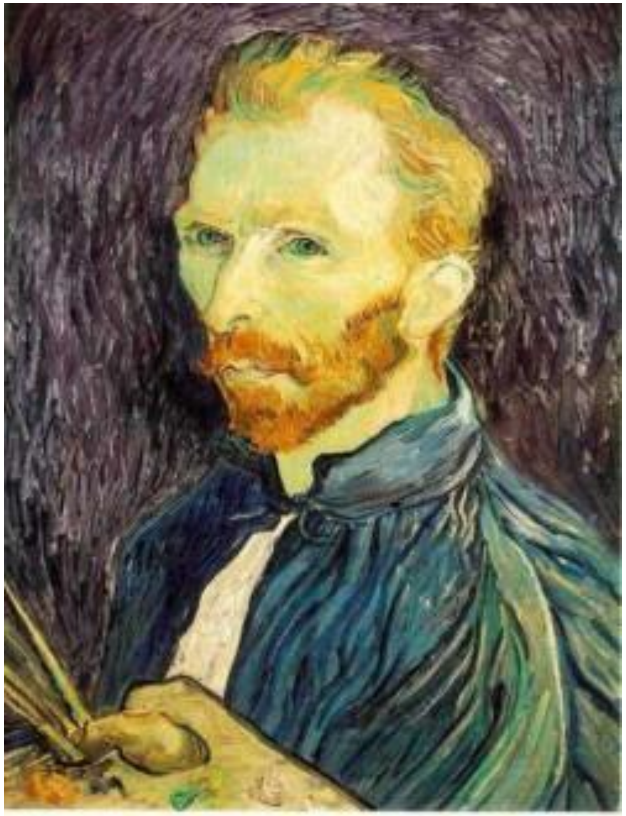
1/2



1/4 (2x zoom)

1/8 (4x zoom)

Subsampling with Gaussian pre-filterin



Gaussian 1/2

G 1/4

G 1/8

Steve Seitz

Another way of thinking about frequency

FOURIER SERIES & FOURIER TRANSFORMS

Fourier series

Jean Baptiste Joseph Fourier (1768-1830)

A bold idea (1807):

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

Fourier series

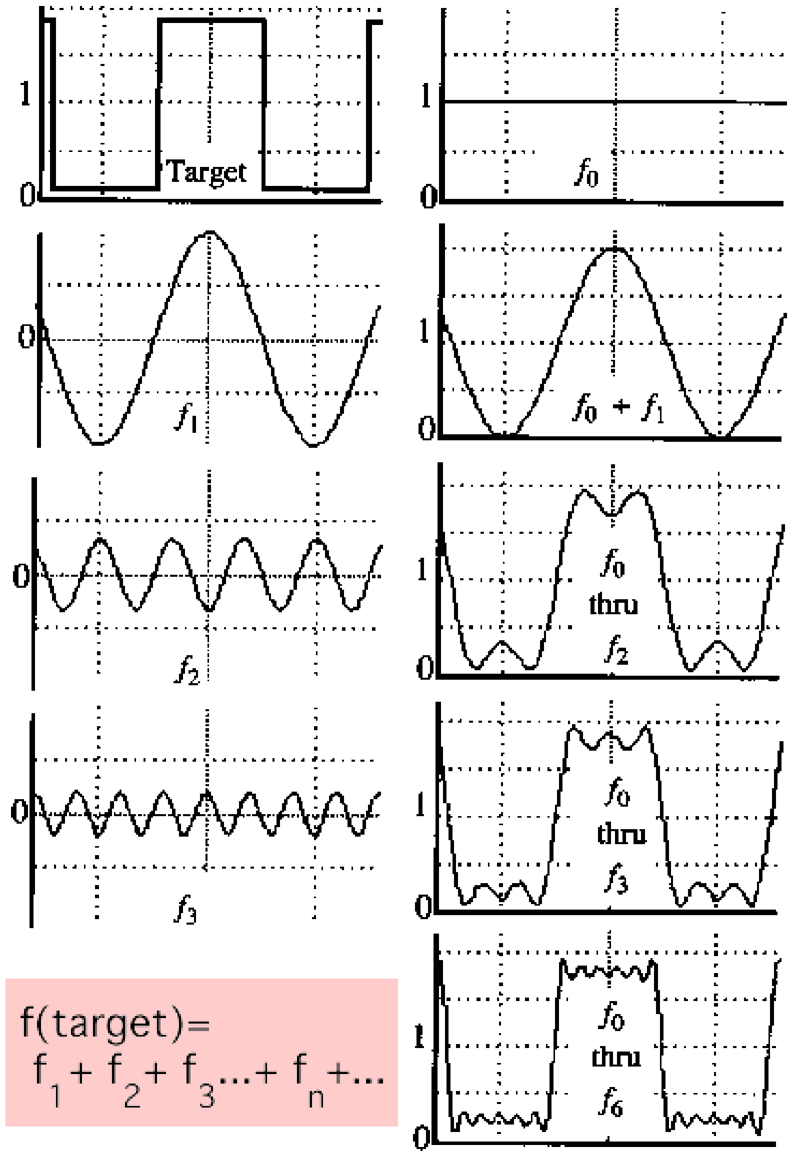
A bold idea (1807):

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

Our building block:

$$A\sin(\omega t) + B\cos(\omega t)$$

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



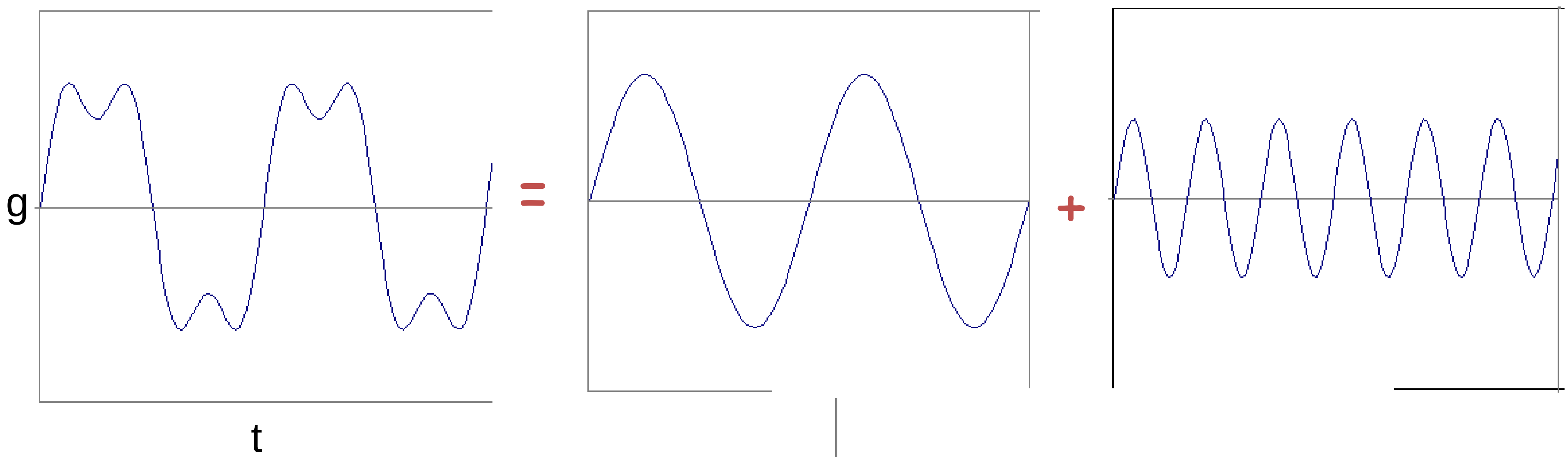
f(target)=
$f_1 + f_2 + f_3 ... + f_n + ...$

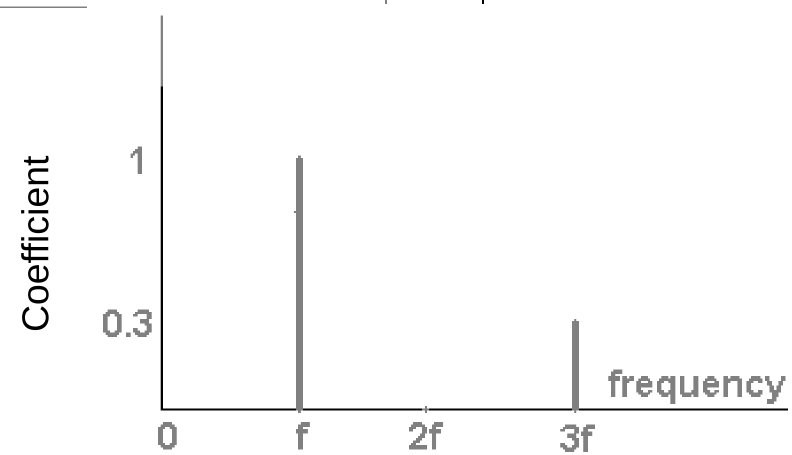
Hays

Example

$t = [0,2], f = 1$

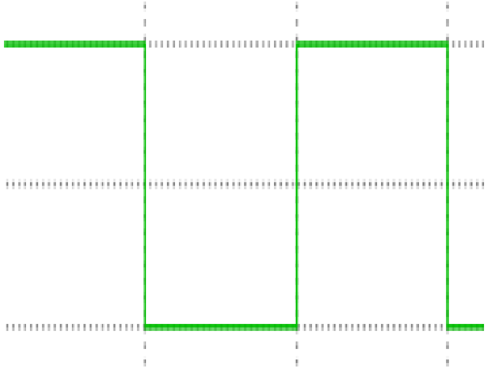
$g(t) = \sin(2\pi f\, t) + (1/3)\sin(2\pi(3f)\, t)$



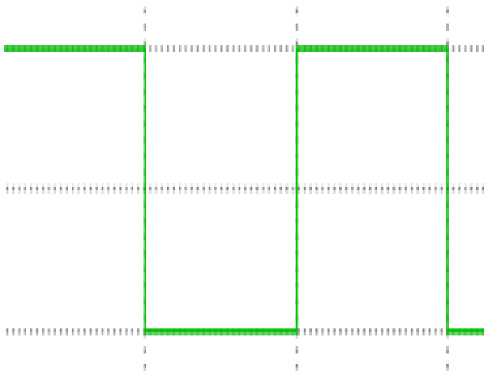


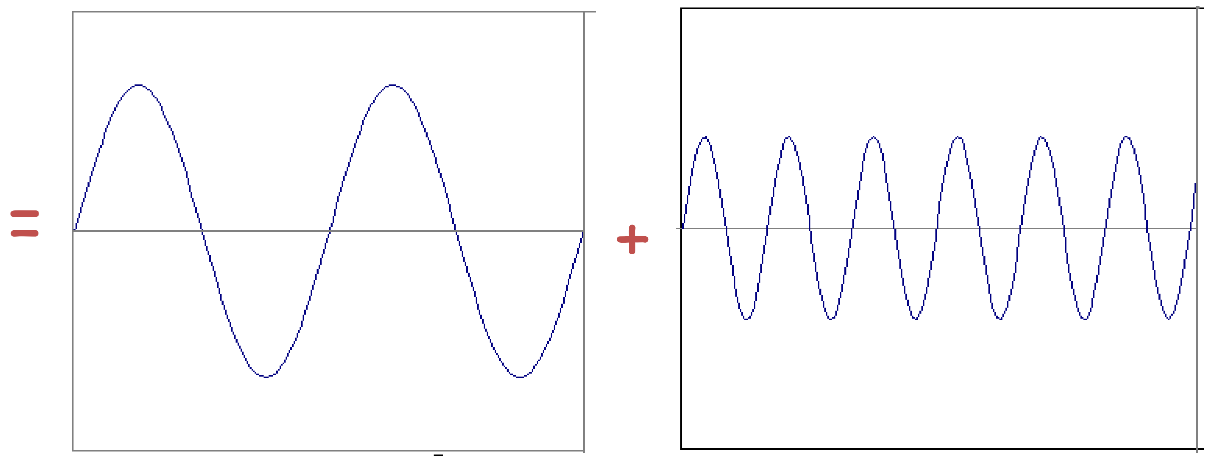
Slides: Efros

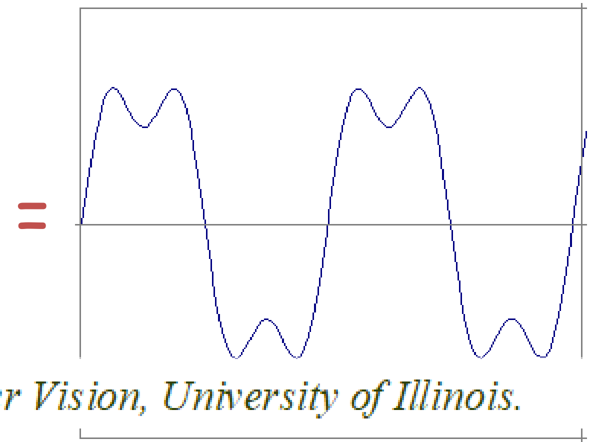
Square wave spectra



Square wave spectra

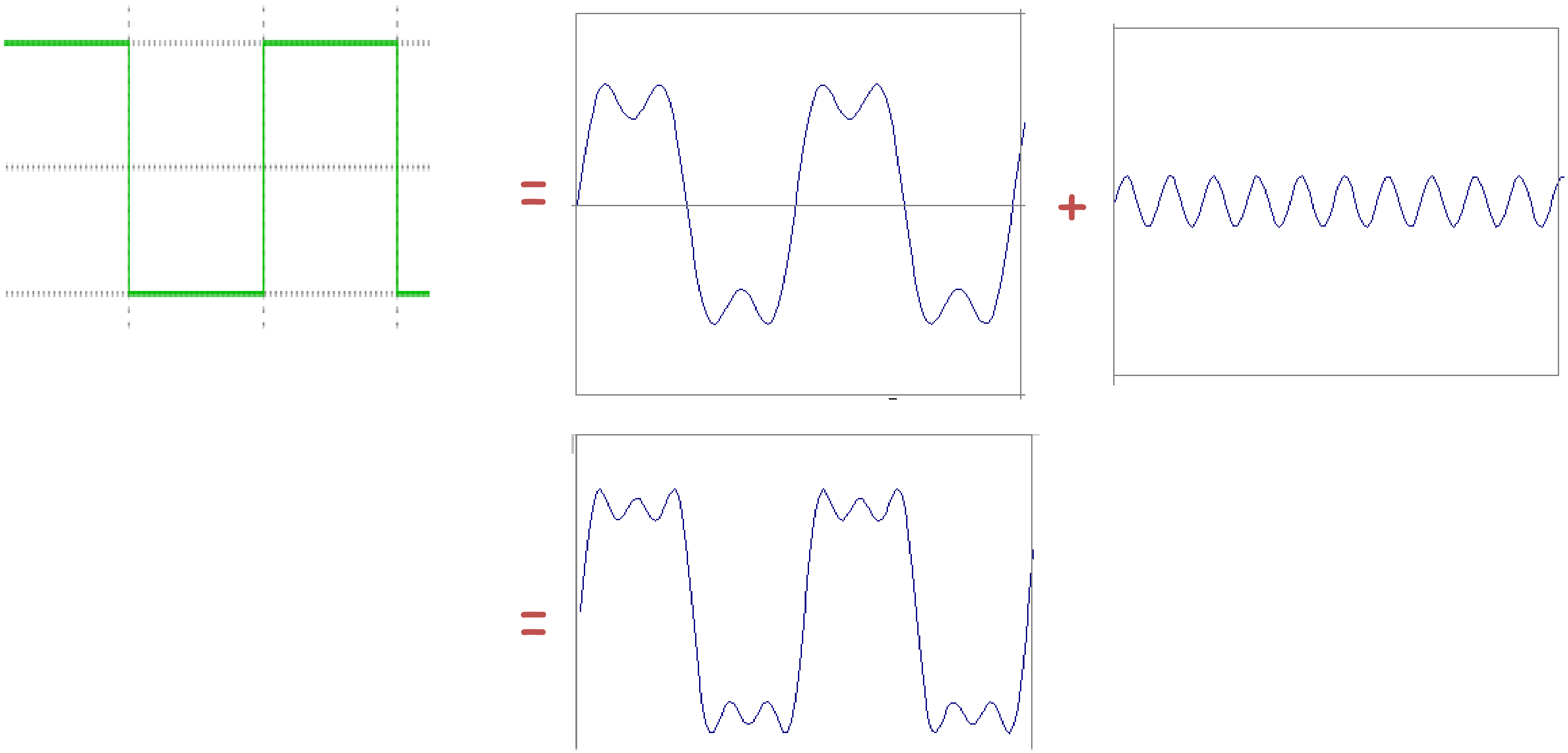






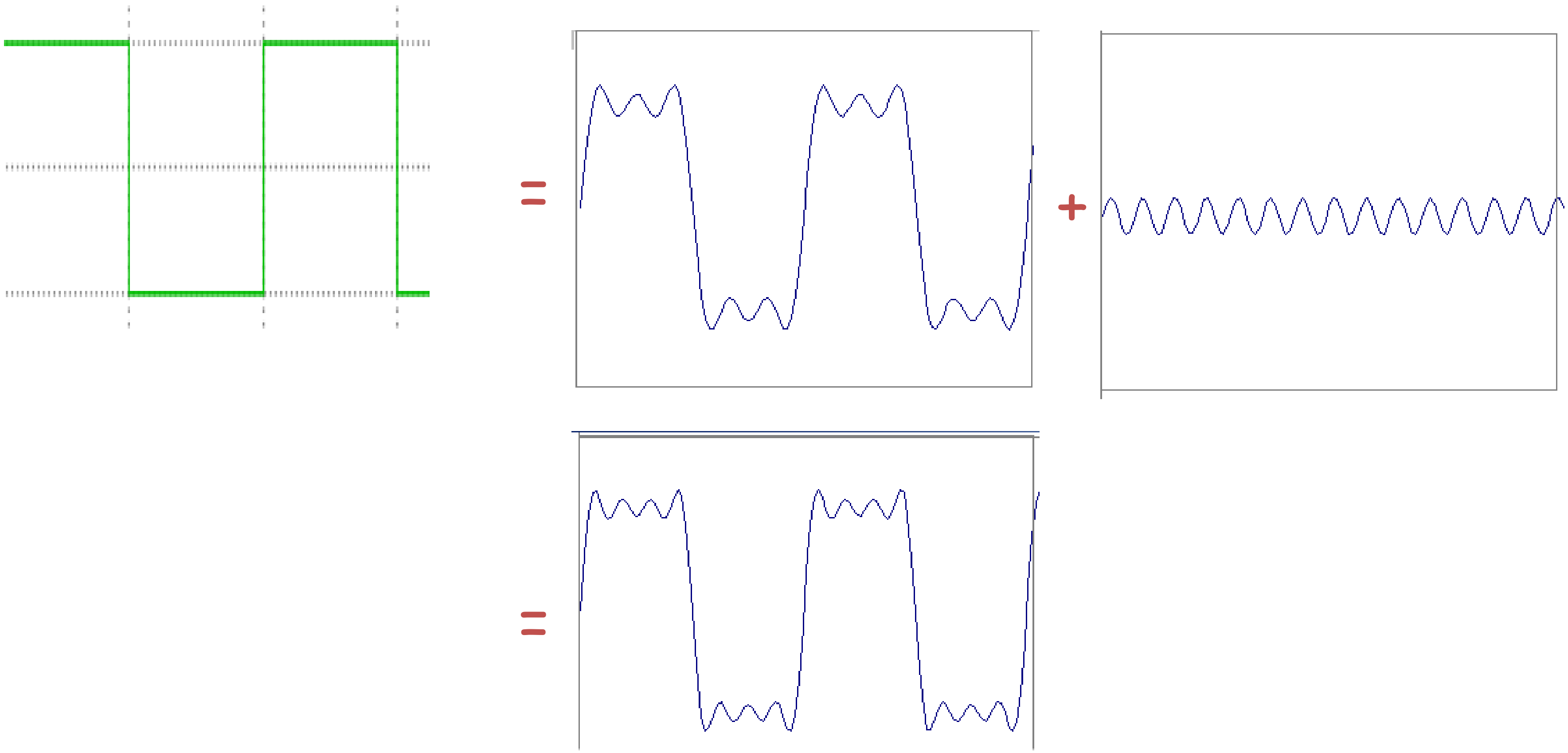
Source : Derek Hoiem. Computer Vision, University of Illinois.

Square wave spectra



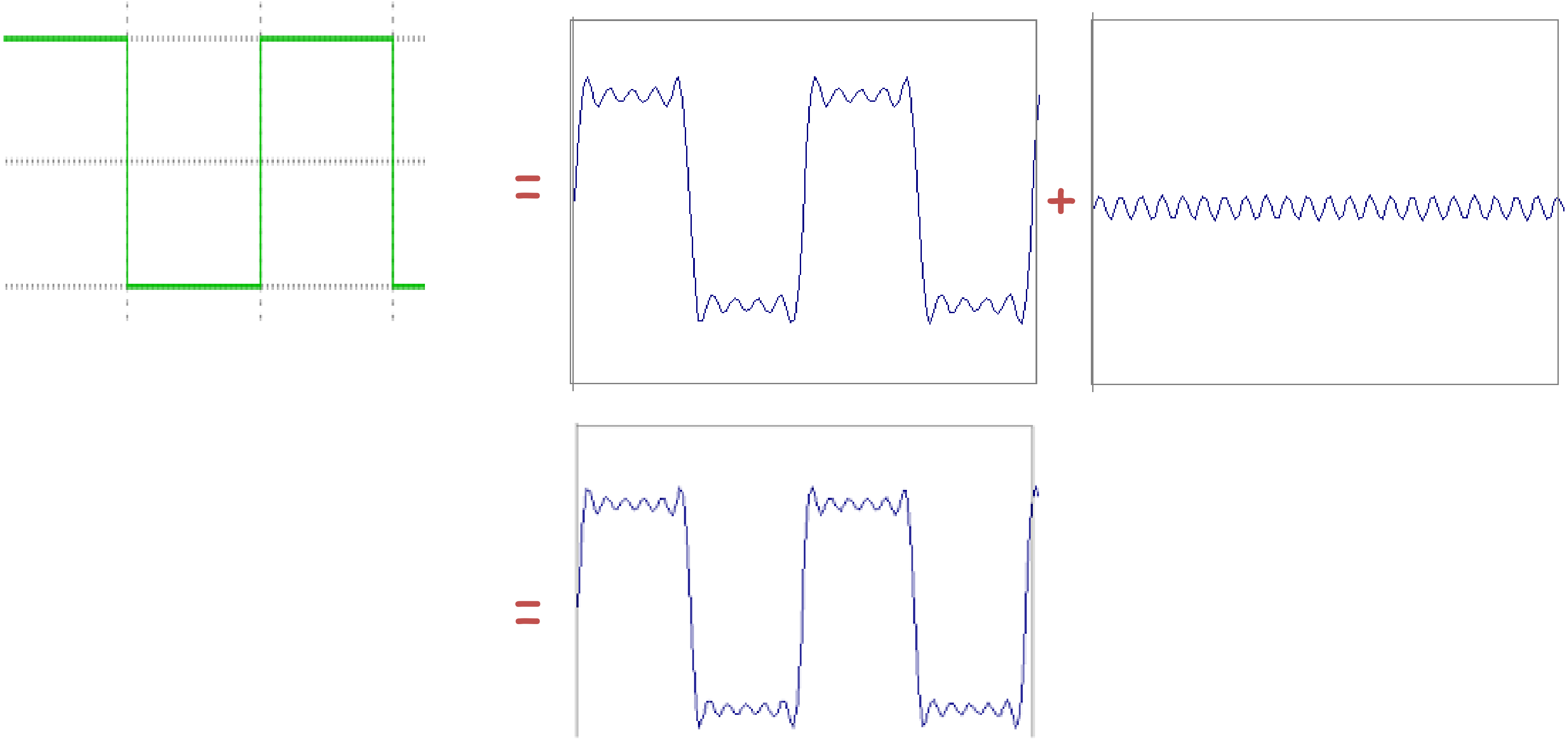
Source : Derek Hoiem. Computer Vision, University of Illinois.

Square wave spectra



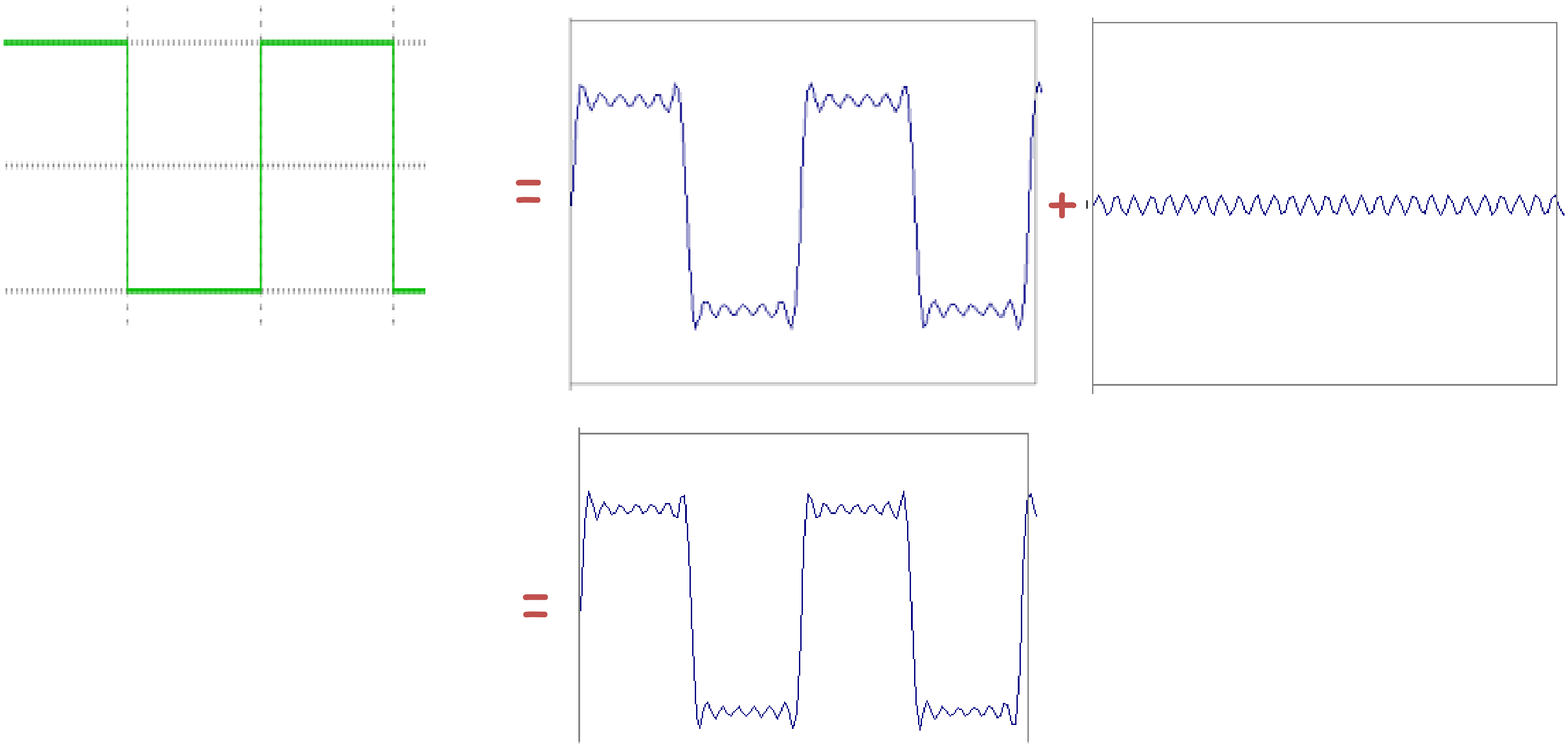
Source : Derek Hoiem. Computer Vision, University of Illinois.

Square wave spectra



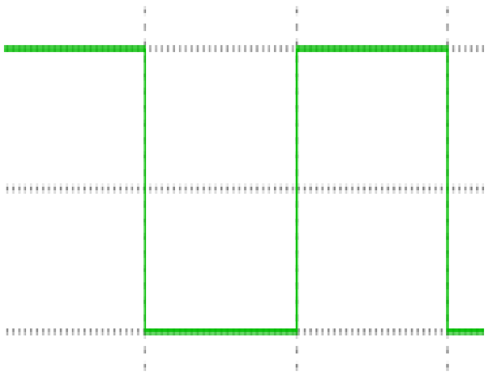
Source : Derek Hoiem. Computer Vision, University of Illinois.

Square wave spectra

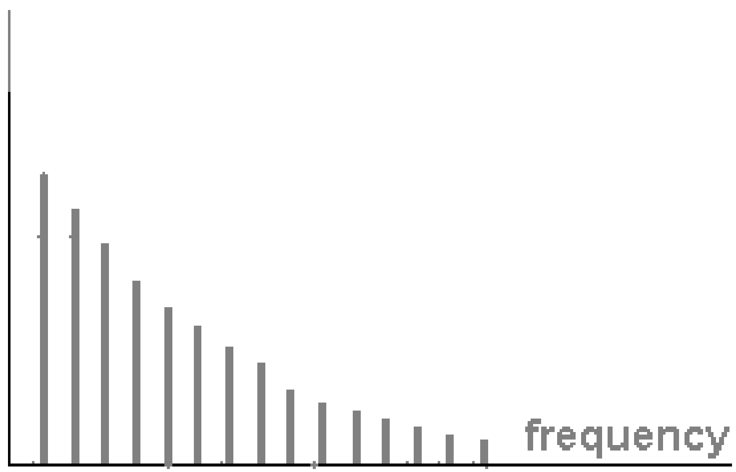


Source : Derek Hoiem. Computer Vision, University of Illinois.

Square wave spectra

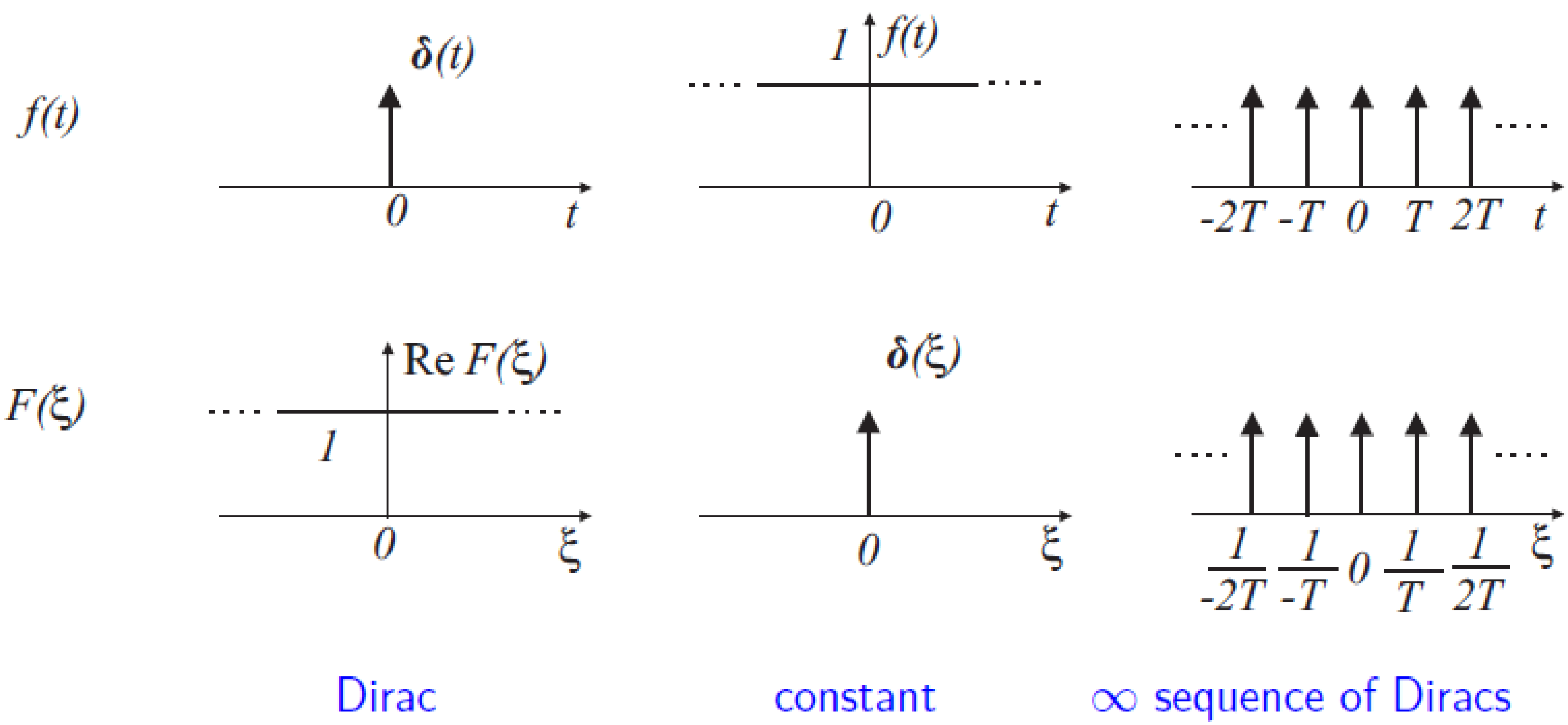


$$= A\sum_{f=1}^{\infty}\frac{1}{f}\sin(2\pi f t)$$



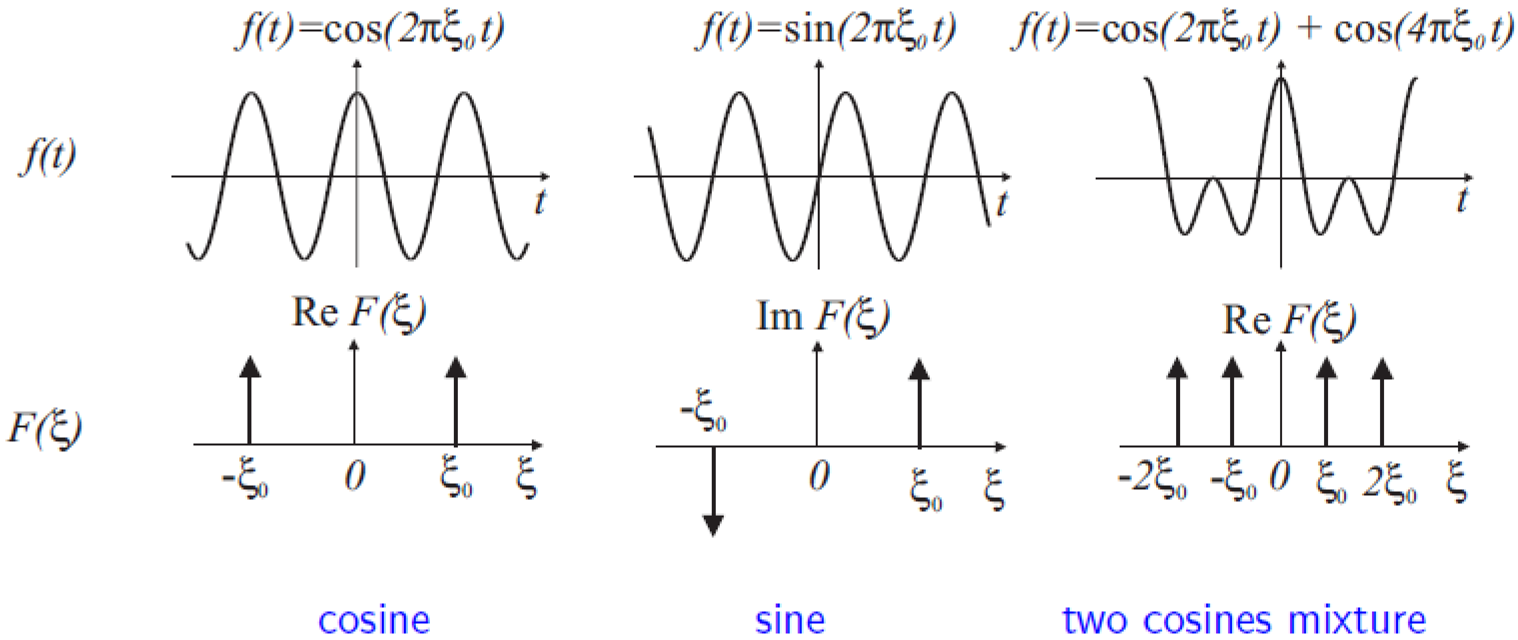
Source : Derek Hoiem. Computer Vision, University of Illinois.

Basic Fourier Transform pairs



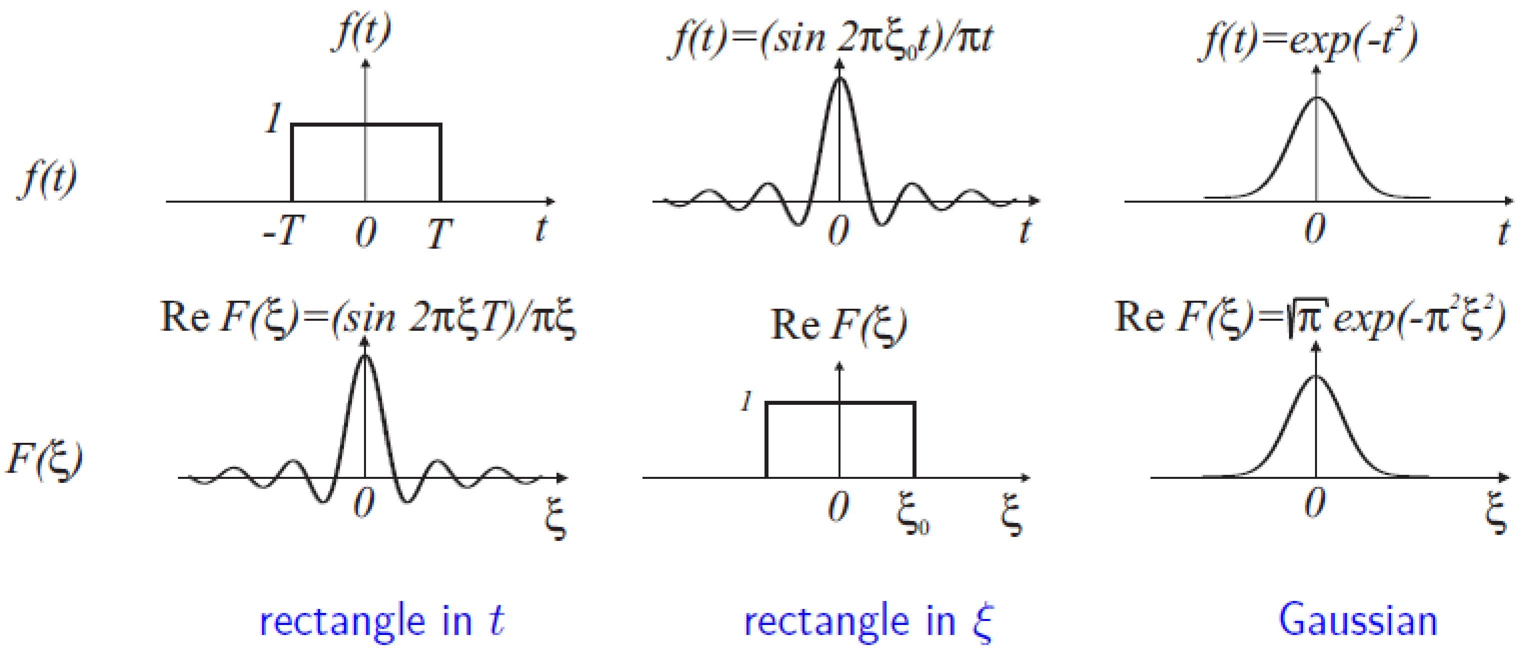
Source: *Václav Hlavác* - ***Fourier transform, in 1D and in 2D***

Basic Fourier Transform pairs



Source: Václav Hlavác - ***Fourier transform, in 1D and in 2D***

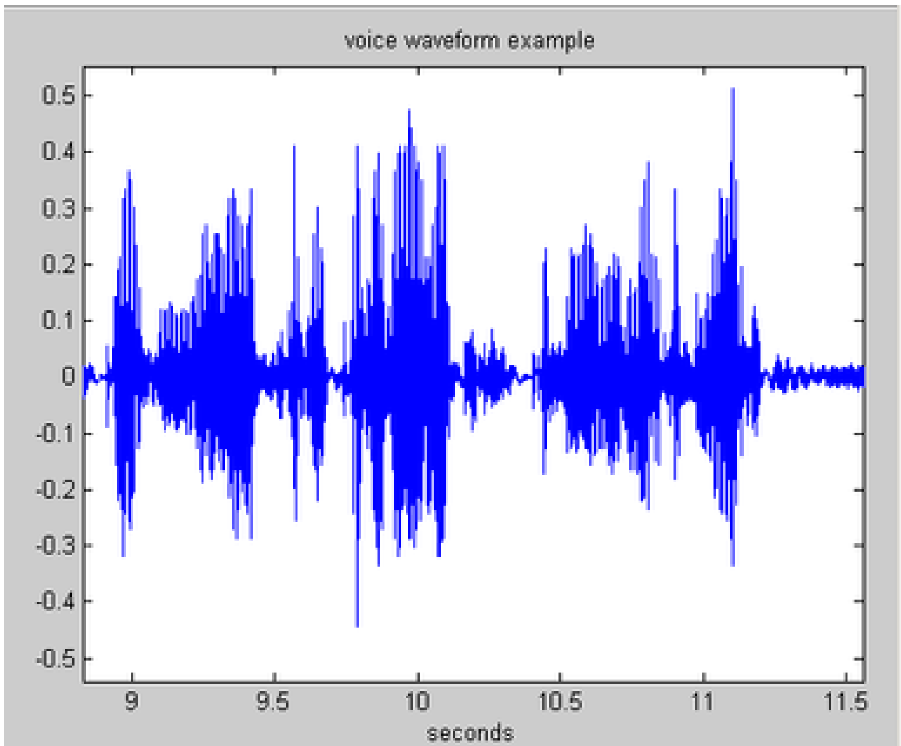
Basic Fourier Transform pairs

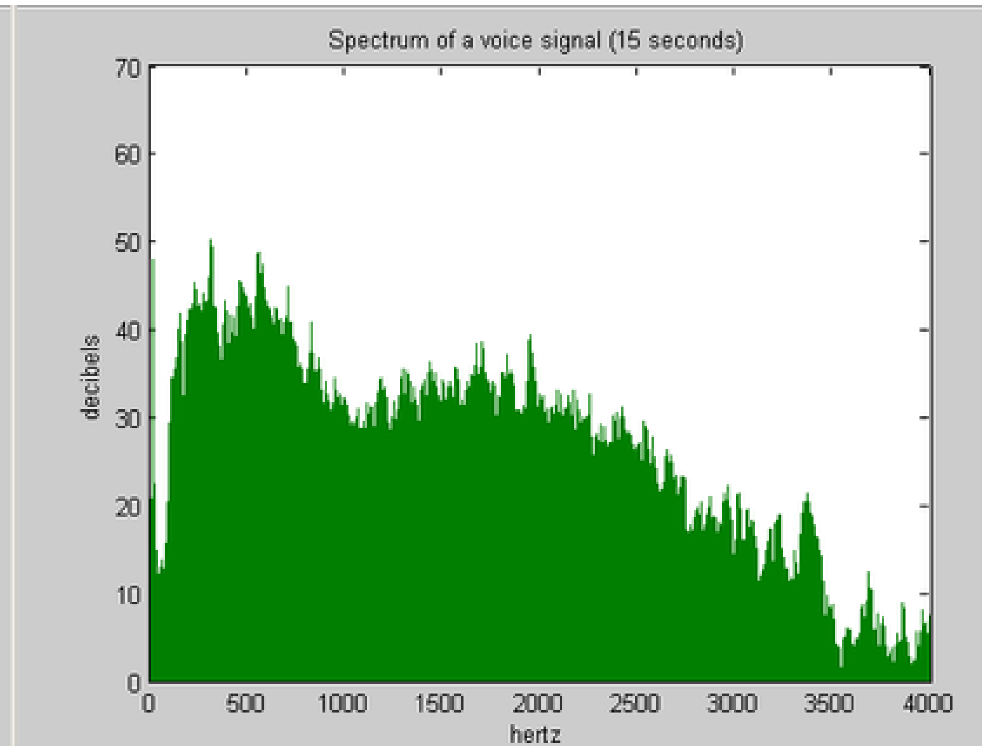


Source: Václav Hlavác - ***Fourier transform, in 1D and in 2D***

Example: Music

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





Slide: Hoiem

Computing Fourier transform

The Fourier transform is defined as:

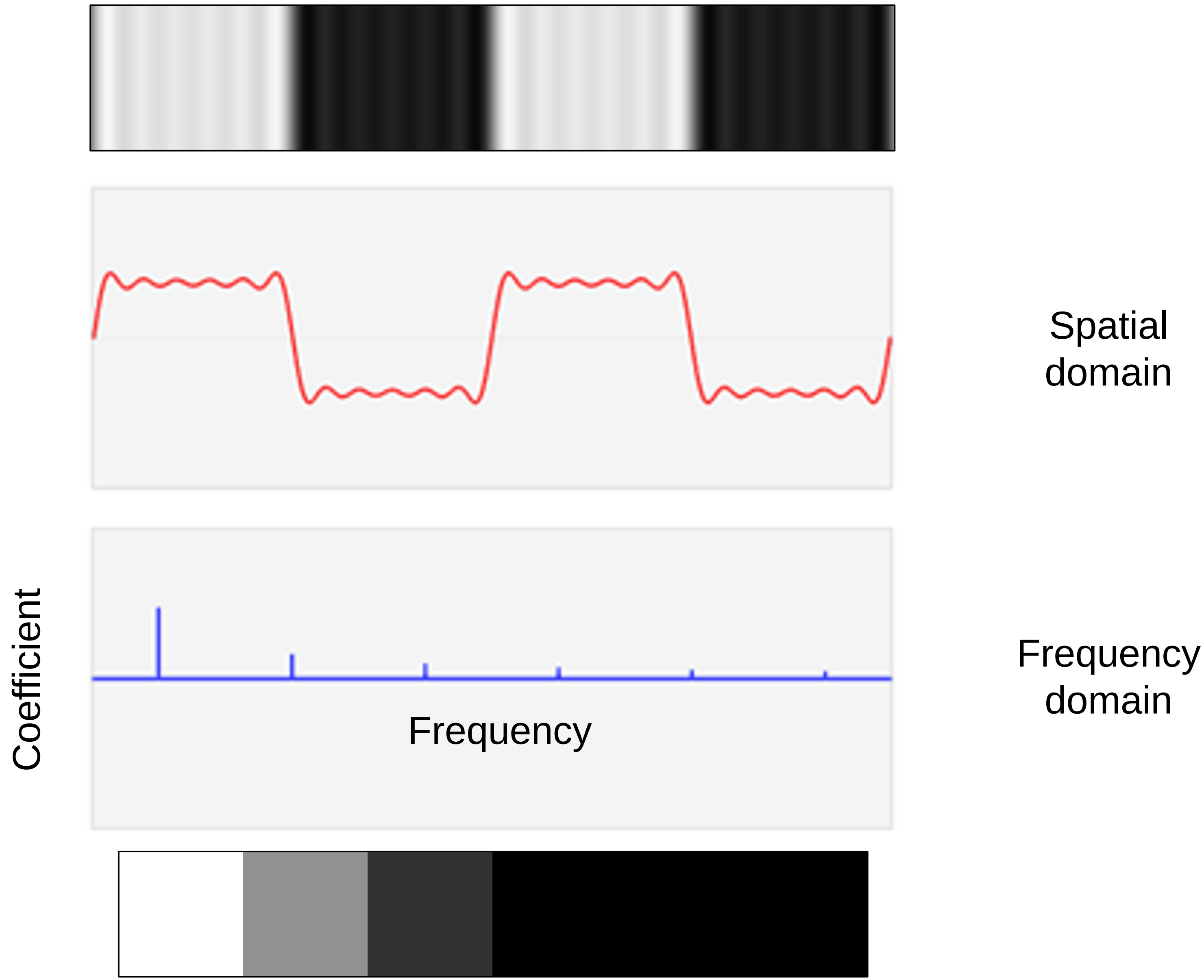
$$H(f) = \int_{-\infty}^{\infty} h(t) e^{2\pi i f t} dt$$
$$h(t) = \int_{-\infty}^{\infty} H(f) e^{-2\pi i f t} df$$

where $h(t)$ is the signal, and $H(f)$ is its Fourier transform; if t is measured in seconds, then f is measured in hertz (Hz).

The discrete Fourier transform is defined as:

$$H_{f_j} = \frac{1}{N} \sum_k h_{t_k} e^{2\pi i f_j t_k}$$
$$h_{t_j} = \frac{1}{N} \sum_k H_{f_k} e^{-2\pi i f_k t_j}$$

where the t_k are the time corresponding to my signal in the time domain h_{t_k}, f_k are the corresponding frequency to my signal in the frequency domain, and N is the number of points of the signal data.



Wikipedia – Fourier transform

2D FFT

- Continuous FFT:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-2\pi i(xu+yv)}\, \mathrm{d}x\, \mathrm{d}y$$

- Inverse FFT:

$$f(x,y) = \int^{\infty}\int^{\infty} F(u,v)\, e^{2\pi i(xu+yv)}\, \mathrm{d}u\, \mathrm{d}v$$

2D FFT

Direct transform

$$F(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \exp\left[-2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)\right] ,$$

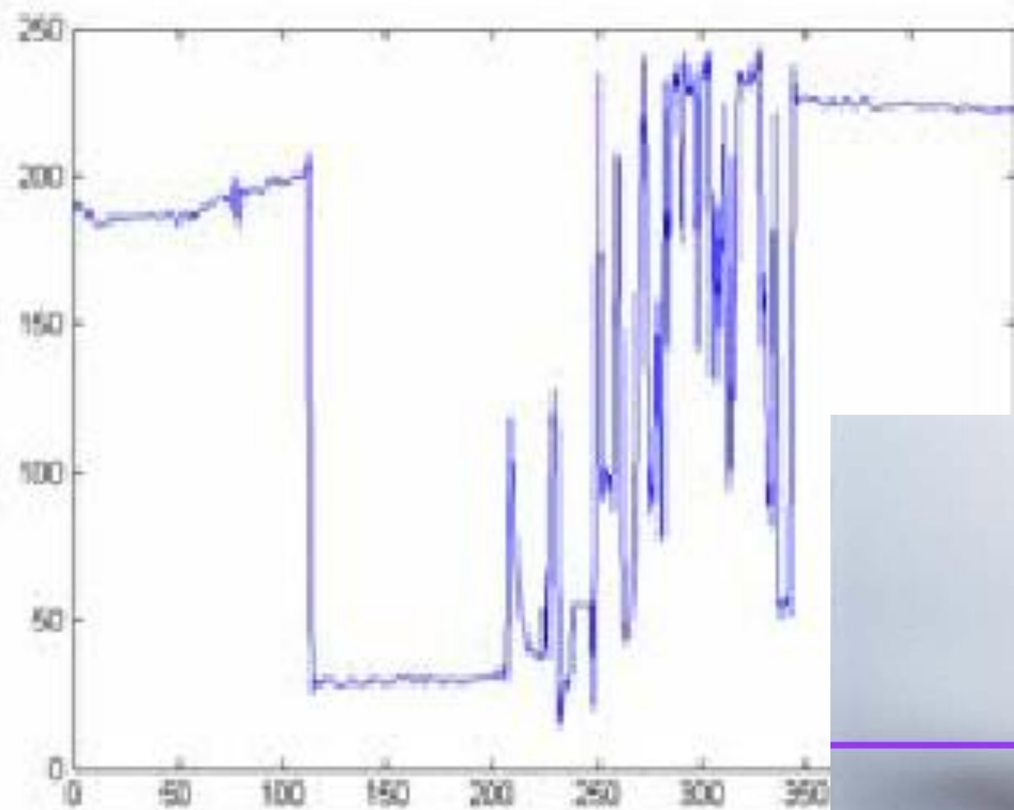
$$u = 0, 1, \ldots, M-1 , \qquad v = 0, 1, \ldots, N-1 ,$$

Inverse transform

$$f(m,n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp\left[2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)\right] ,$$

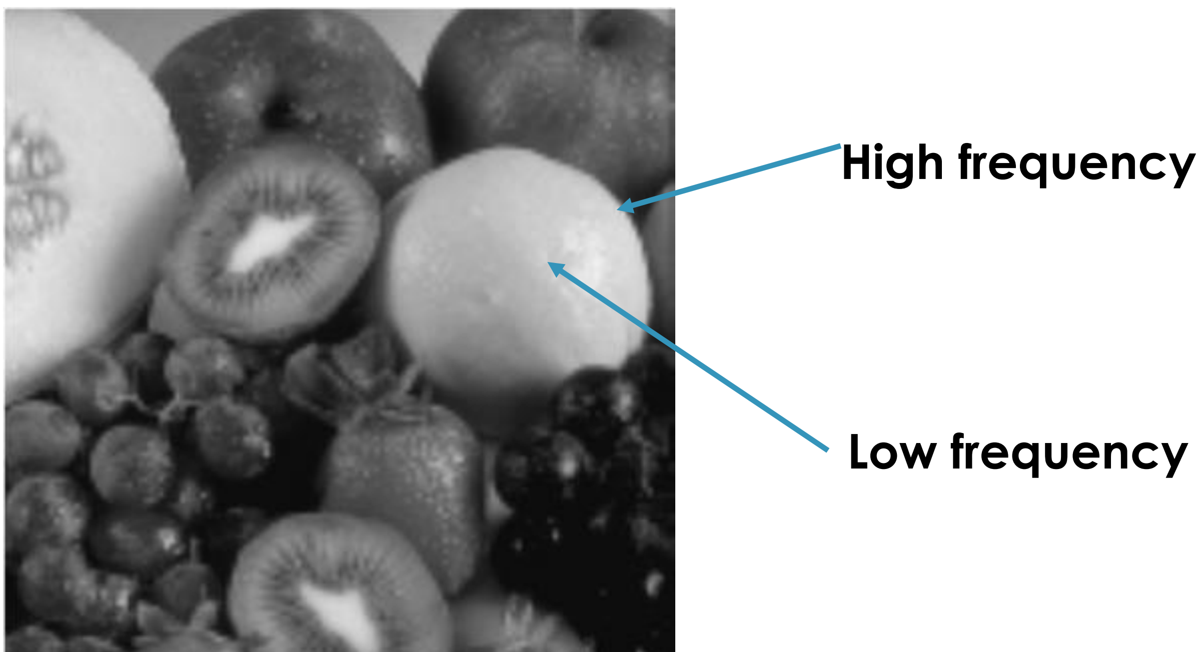
$$m = 0, 1, \ldots, M-1 , \qquad n = 0, 1, \ldots, N-1 .$$

Frequencies in image



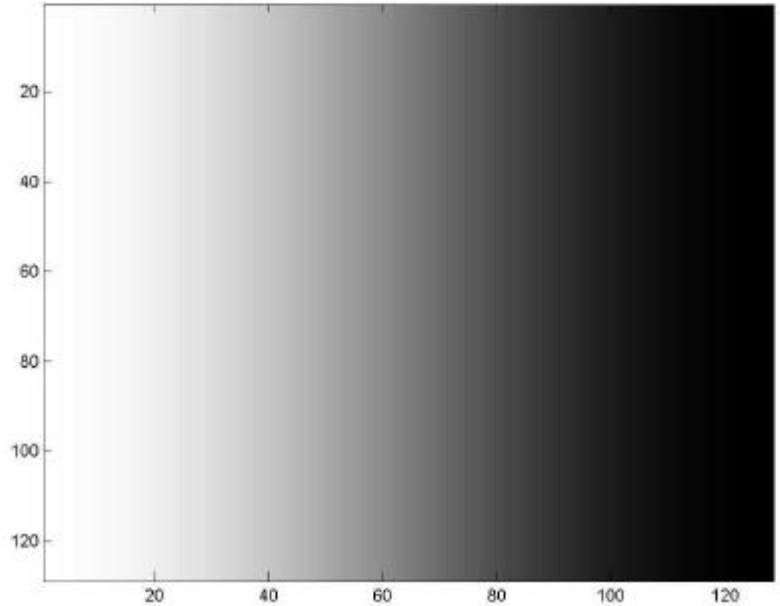
Frequencies in image

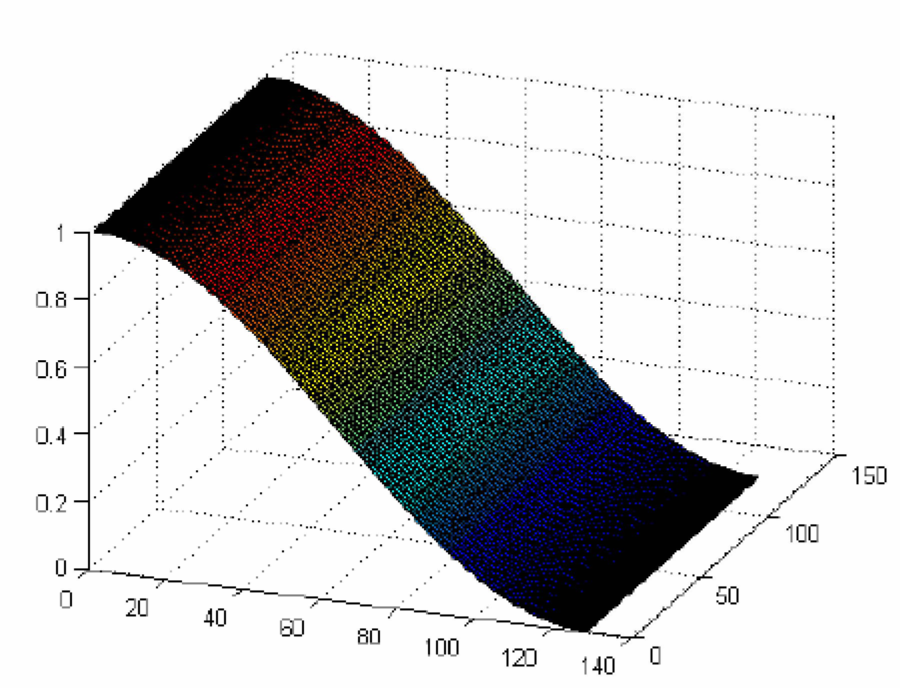
- What are the (low/high) frequencies in an image?
  - Frequency **= intensity change**
  - Slow changes (homogeneous /blur regions): low frequency
  - fast/abrupt changes (egde, contour, noise): high frequency



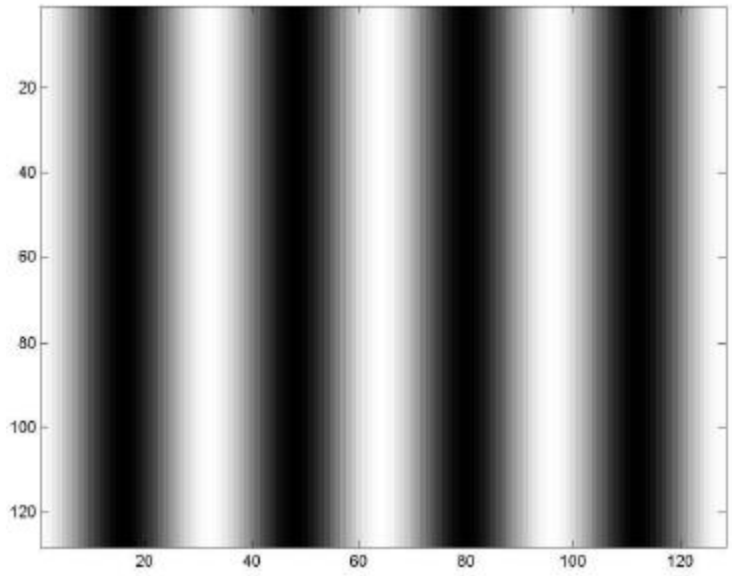
Most of energy concentrated in low frequencies

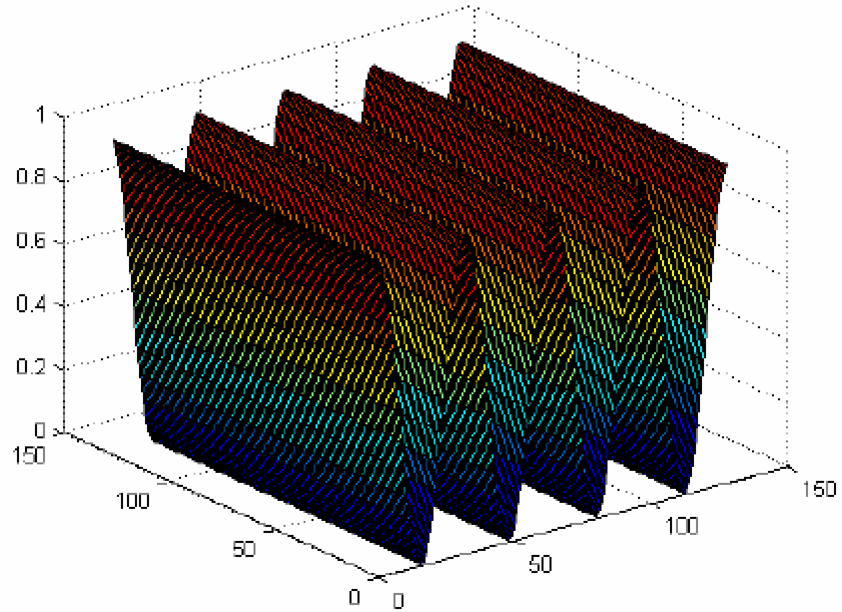
Low frequencies





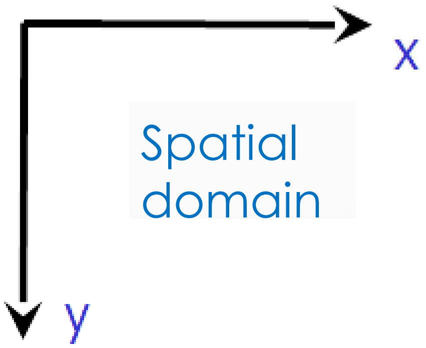
High frequencies



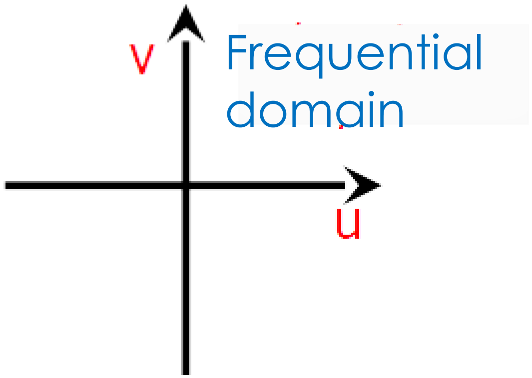


Fourrier transform

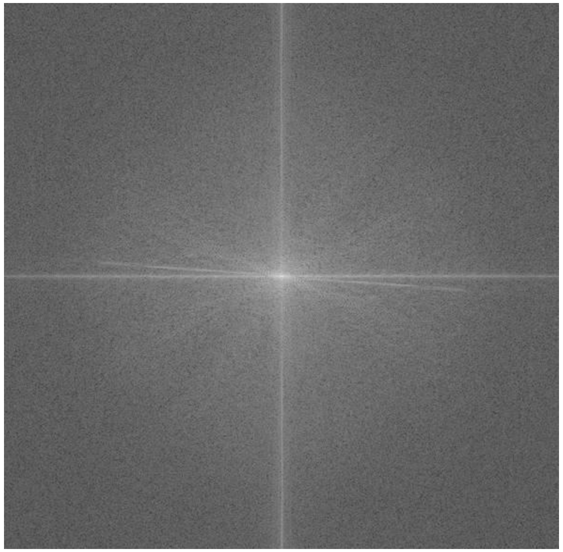
Original image



Spectra | F(u,v) |

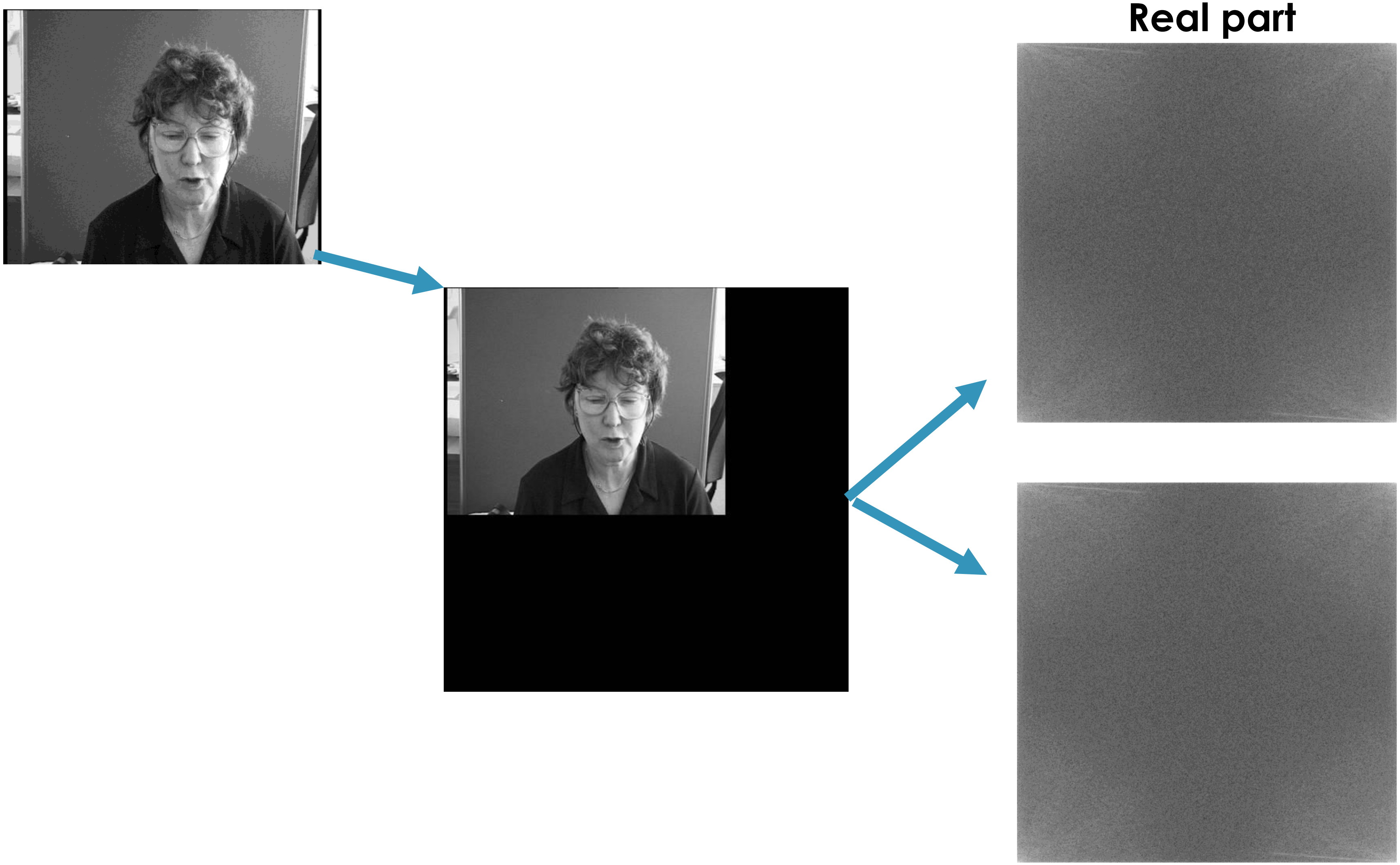


Enhanced Spectra
log (1+ | F(u,v) |)



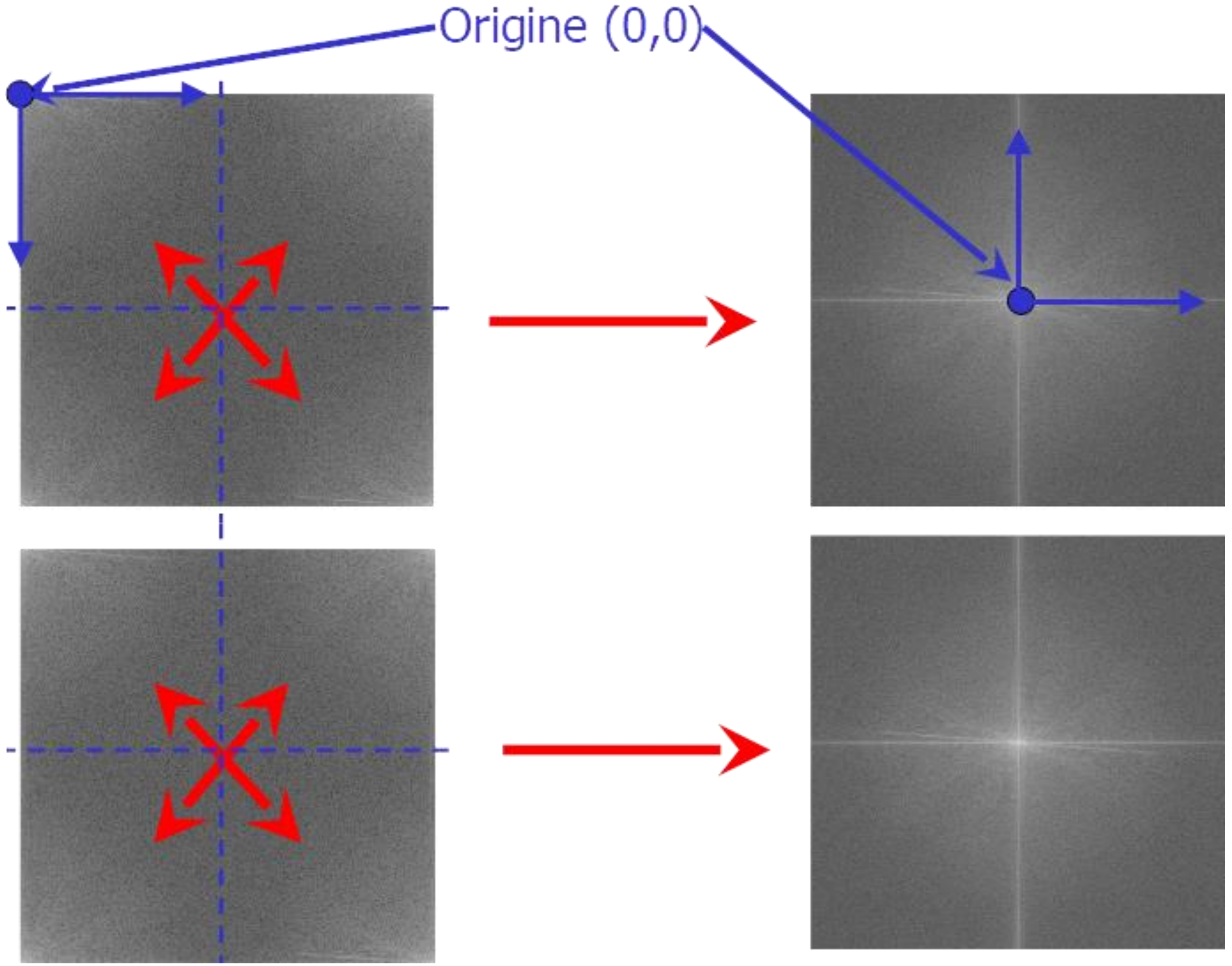
Fourrier transform

Real part

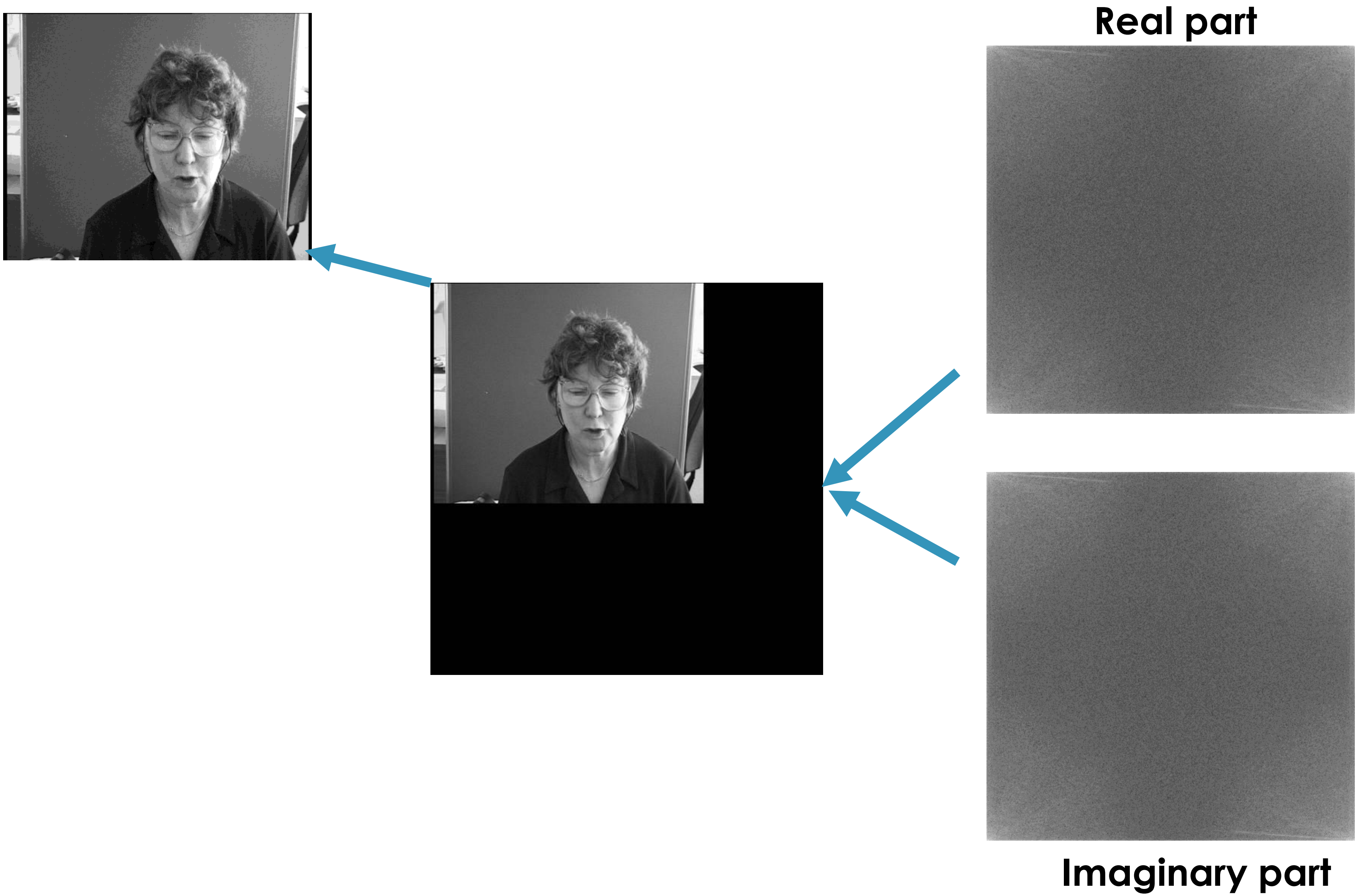


Imaginary part

Fourrier transform

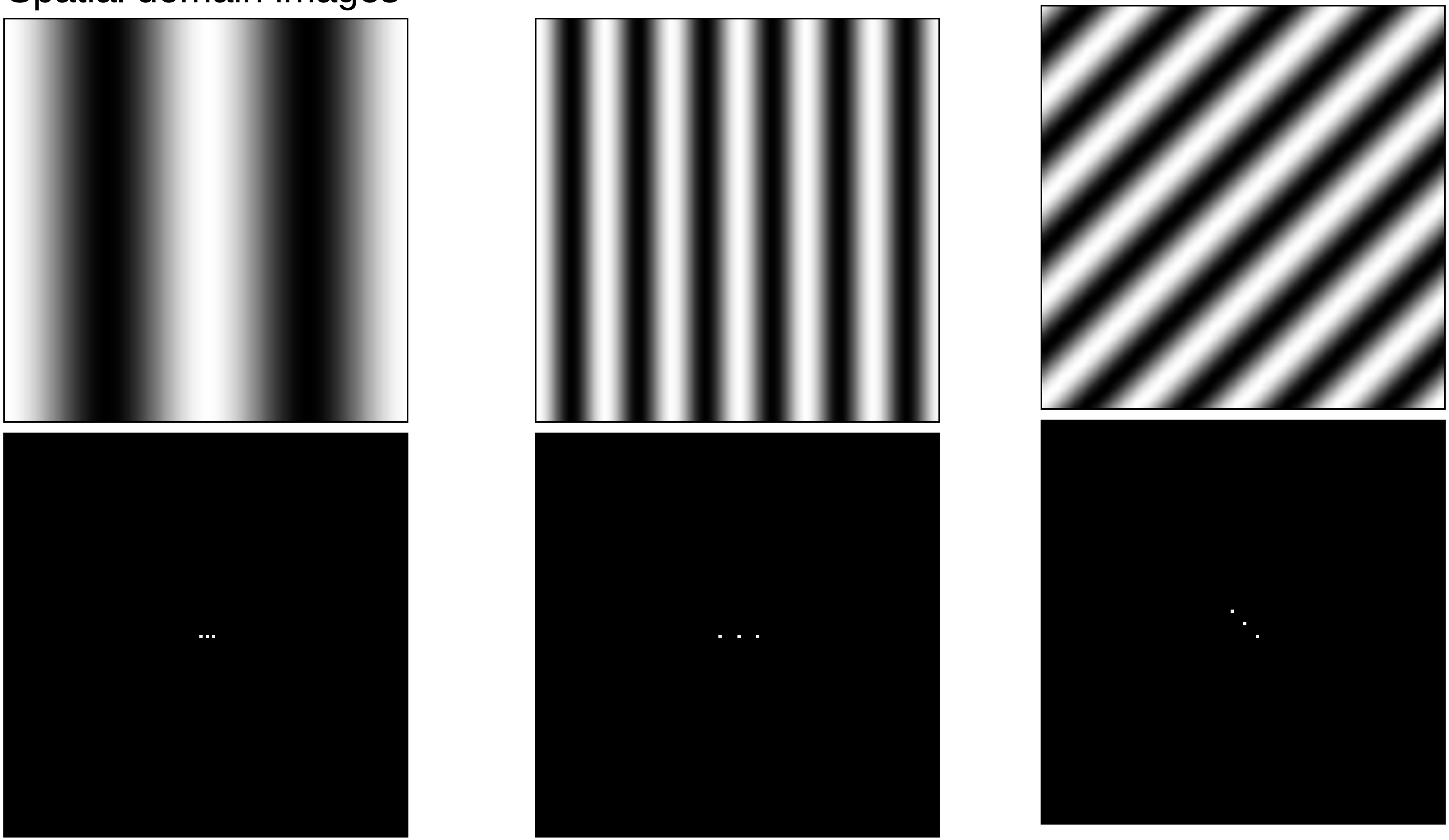


Inverse Fourrier transform



Fourier analysis in images

Spatial domain images

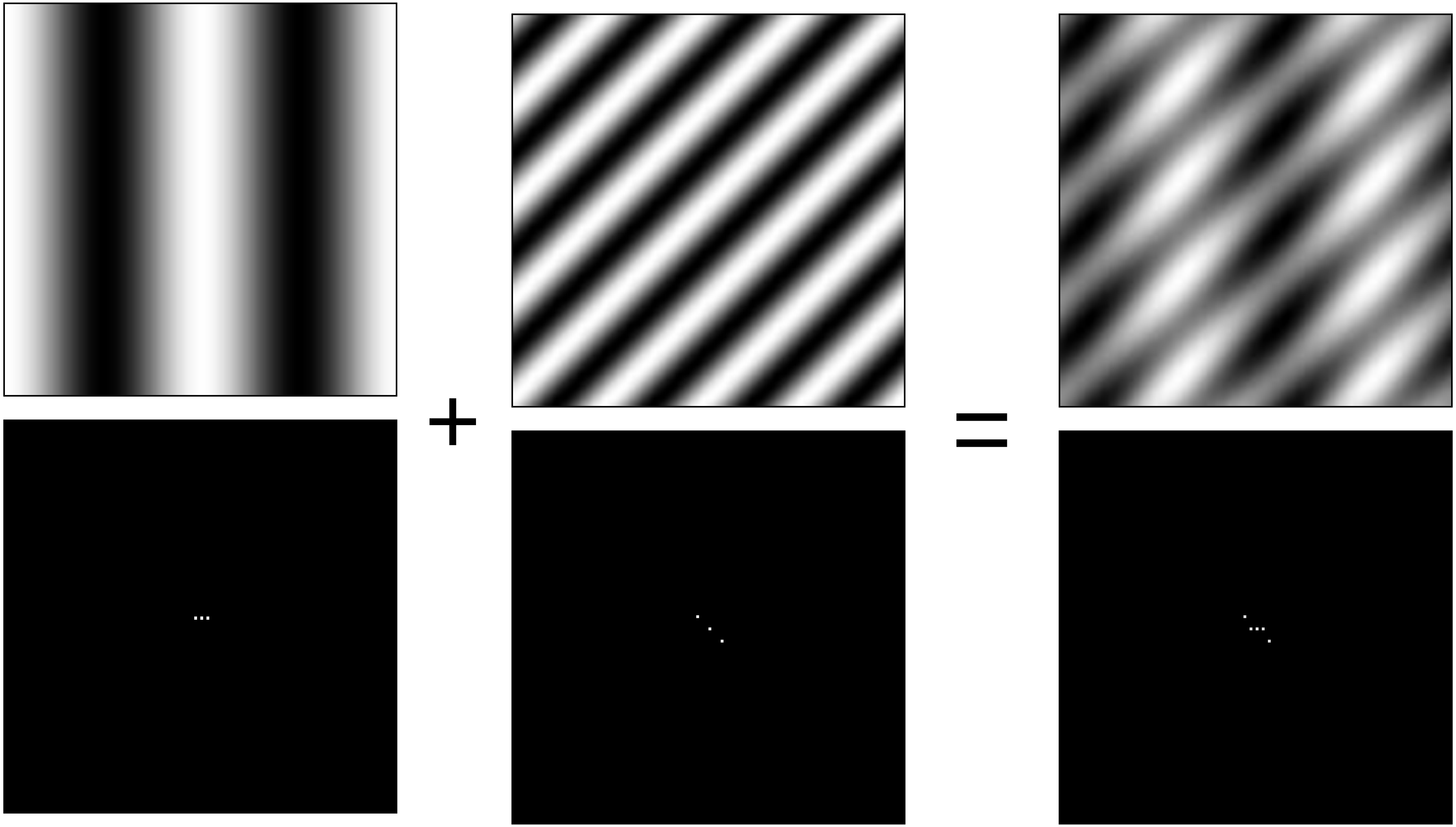


Fourier decomposition frequency amplitude images

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

Signals can be composed

Spatial domain images



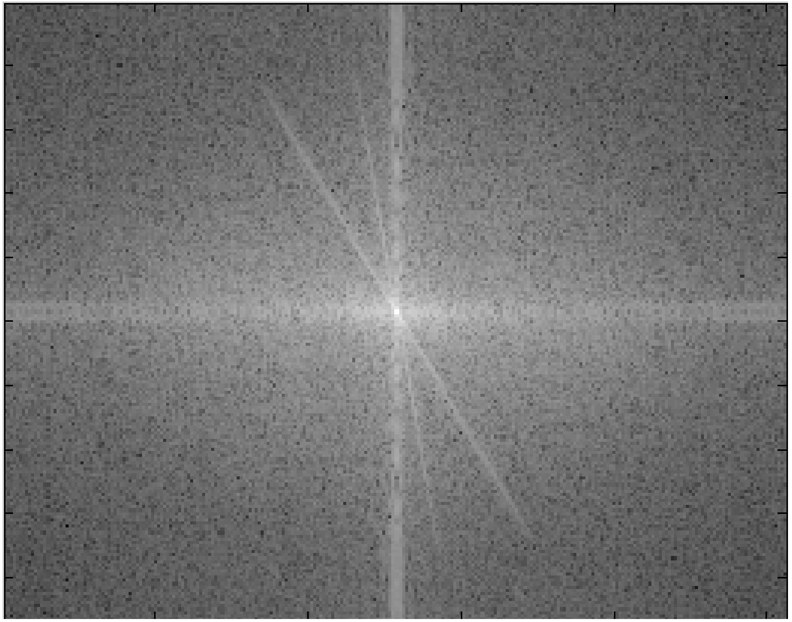
Fourier decomposition frequency amplitude images

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

Natural image

Natural image

Fourier decomposition
Frequency coefficients (amplitude)

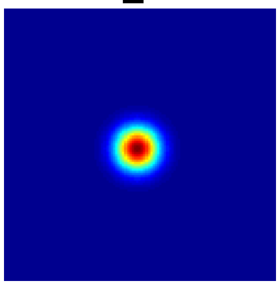


What does it mean to be at pixel x,y?
What does it mean to be more or less bright in the Fourier decomposition image?

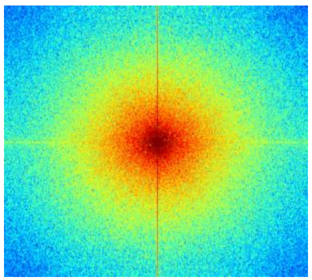
Think-Pair-Share

Match the spatial domain image to the Fourier magnitude image

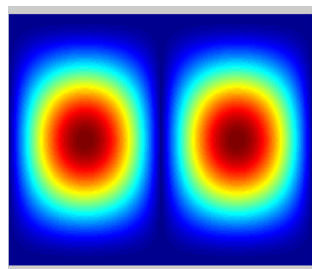
1



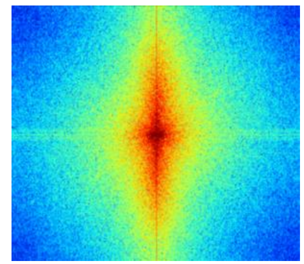
2



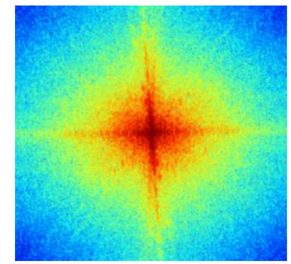
3



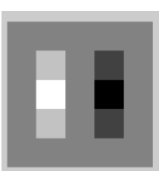
4



5

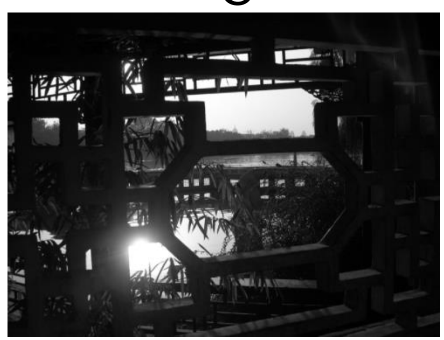


A

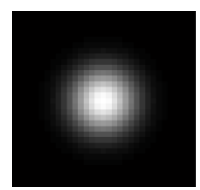


B

C



D

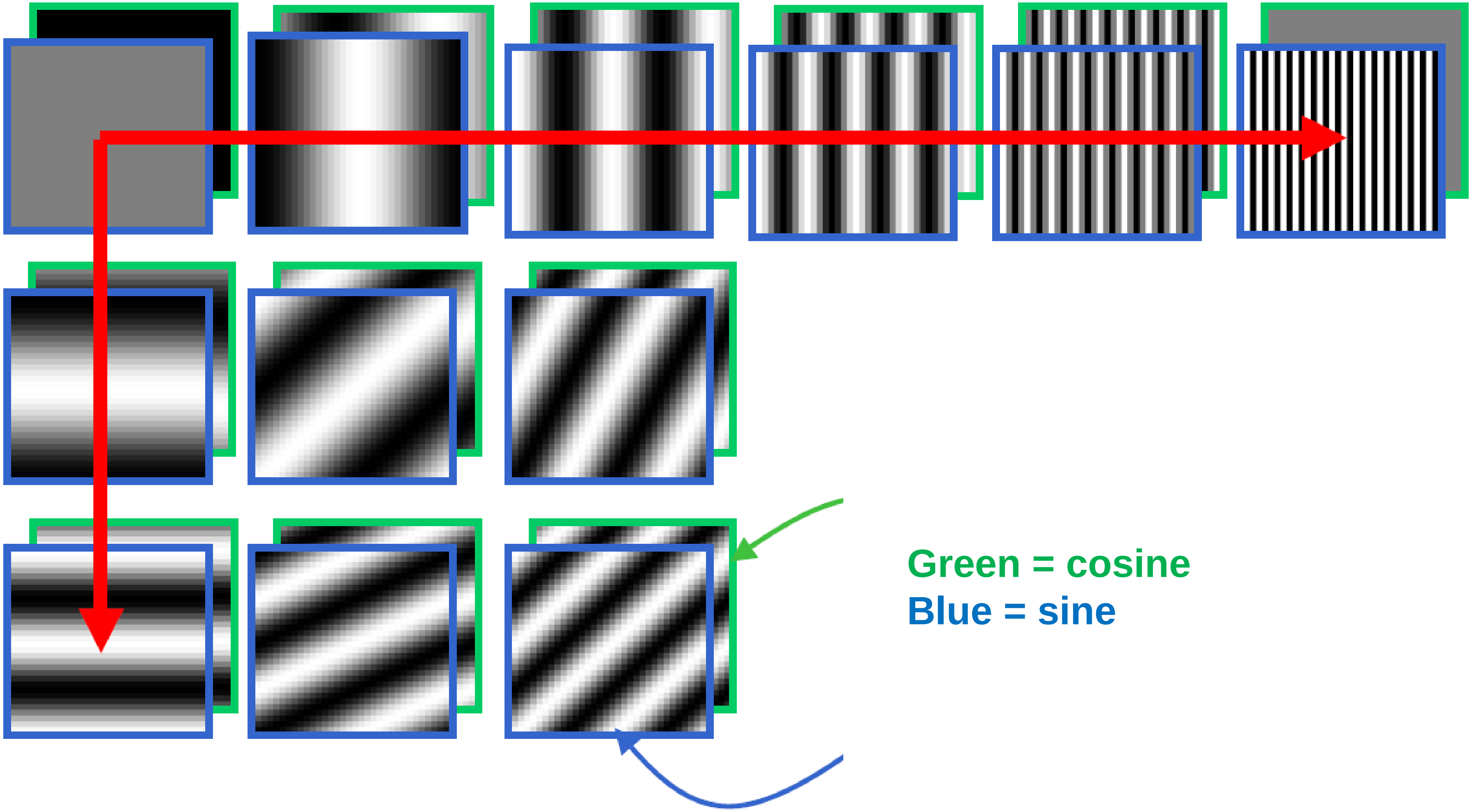


E

Hoiem

Fourier Bases

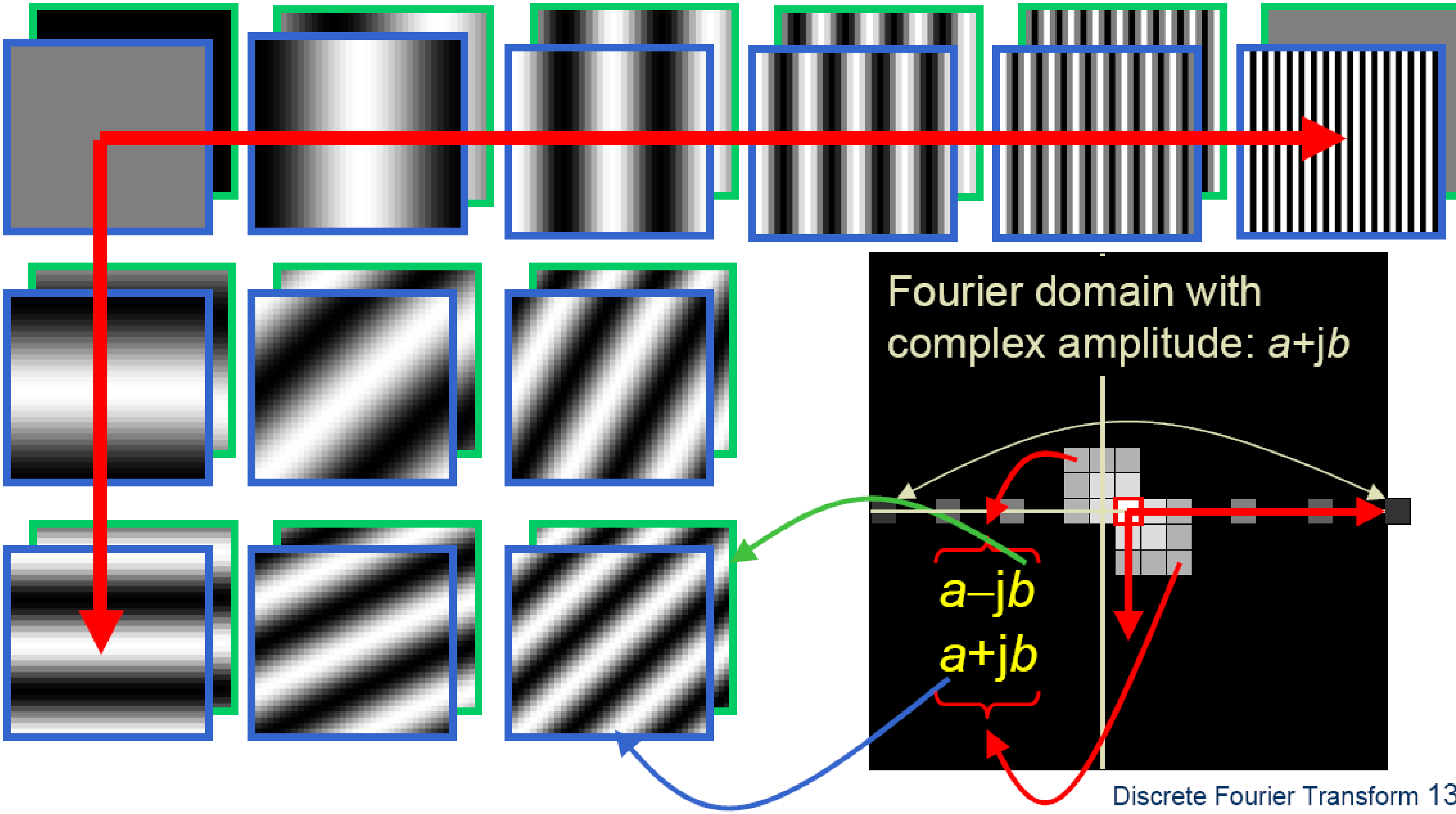
Teases away ‘fast vs. slow’ changes in the image.



This change of basis is the Fourier Transform

Hays

Fourier Bases



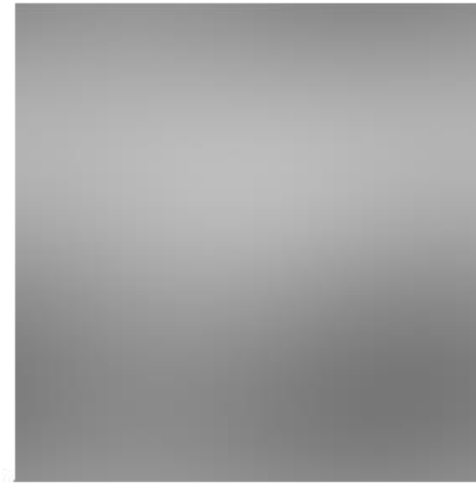
Discrete Fourier Transform 13

Hays

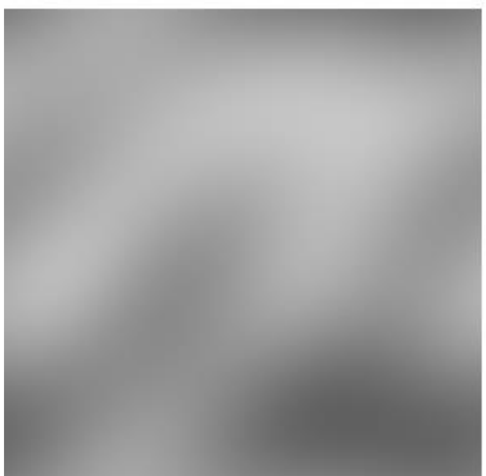
Basis reconstruction

Full image

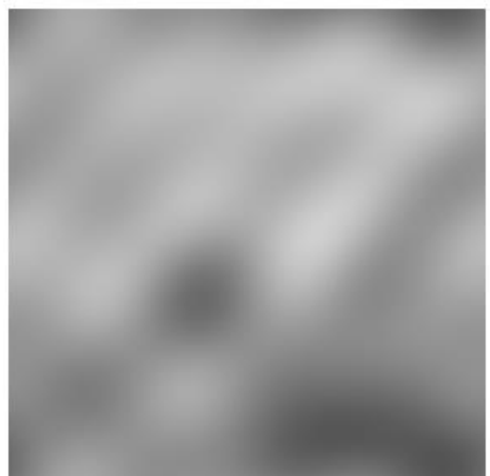
First 1 basis fn



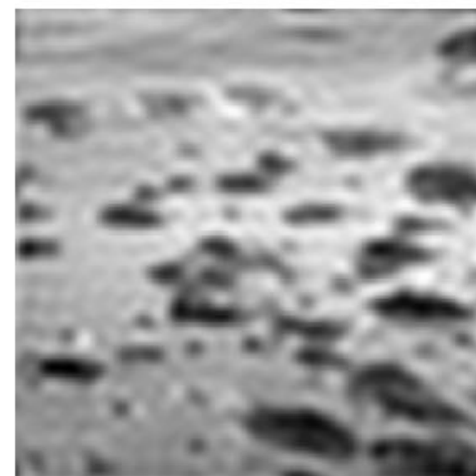
First 4 basis fns



First 9 basis fns



First 16 basis fns



First 400 basis fns

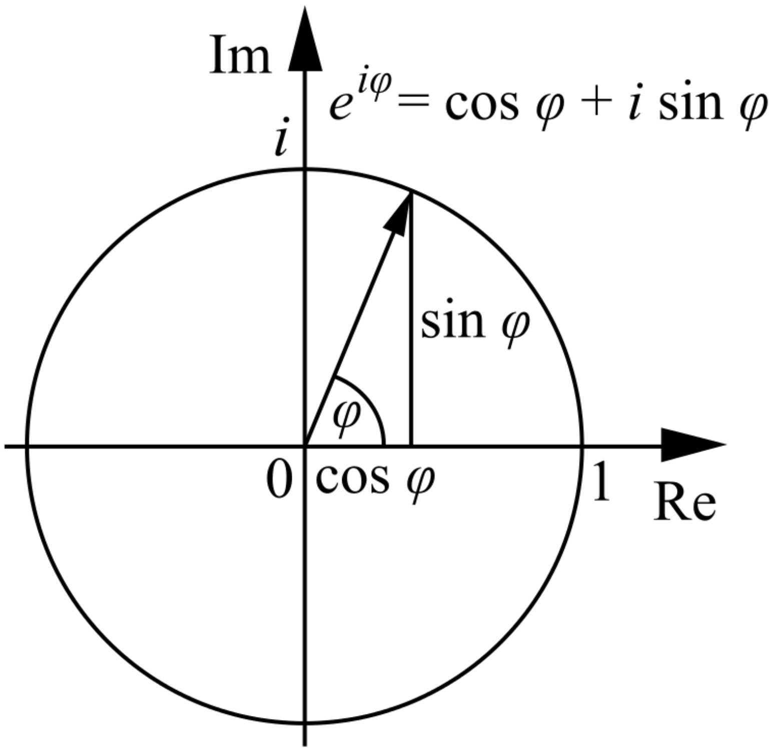
Danny Alexander

Fourier Transform

- Stores the amplitude and phase at each frequency:
  - For mathematical convenience, this is often notated in terms of real and complex numbers
  - Related by Euler's formula

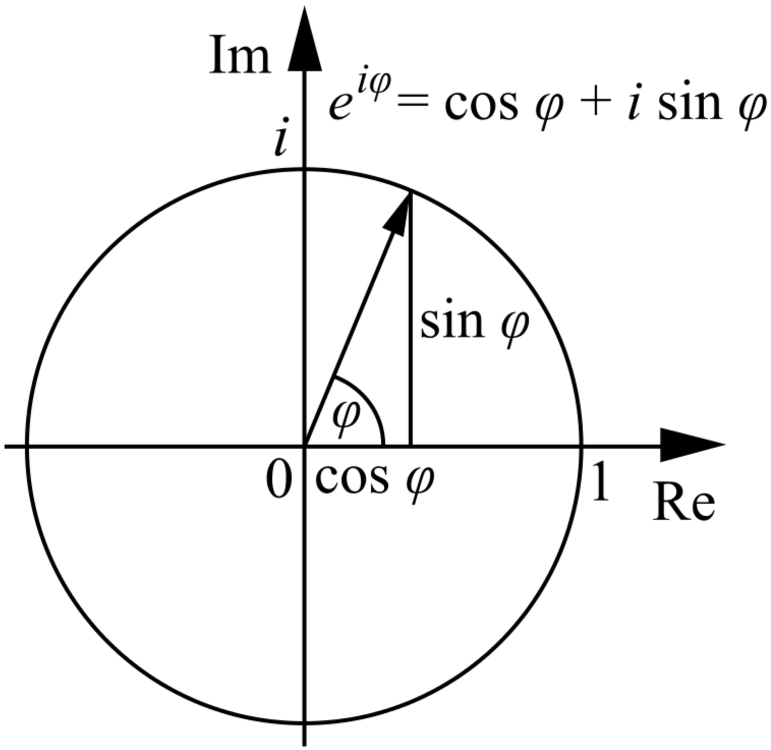
Fourier Transform

- Stores the amplitude and phase at each frequency:
  - For mathematical convenience, this is often notated in terms of real and complex numbers
  - Related by Euler's formula



Fourier Transform

- Stores the amplitude and phase at each frequency:
  - For mathematical convenience, this is often notated in terms of real and complex numbers
  - Related by Euler's formula



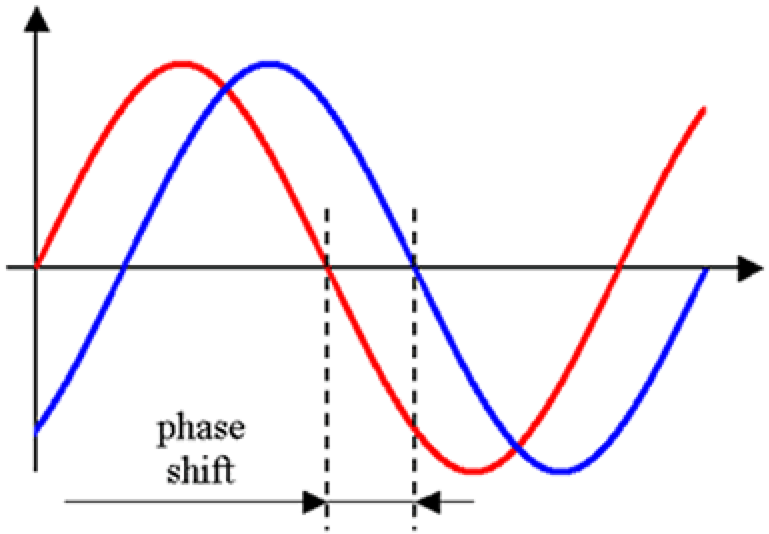
Amplitude encodes how much signal there is at a particular frequency:

$$A = \pm\sqrt{\mathrm{Re}(\varphi)^2 + \mathrm{Im}(\varphi)^2}$$

Phase encodes spatial information (indirectly):

$$\phi = \tan^{-1}\frac{\mathrm{Im}(\varphi)}{\mathrm{Re}(\varphi)}$$

Amplitude / Phase



- Amplitude tells you “how much”
- Phase tells you “where”

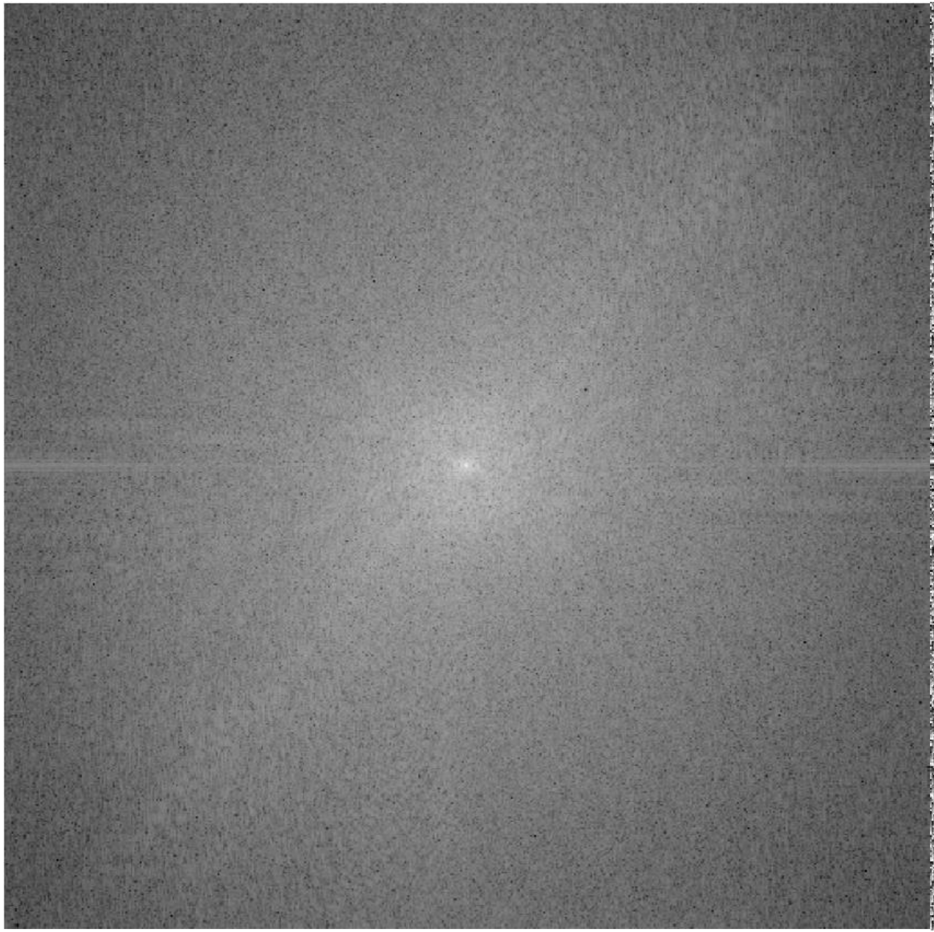
- Translate the image?
  - Amplitude unchanged
  - Adds a constant to the phase.

What about phase?

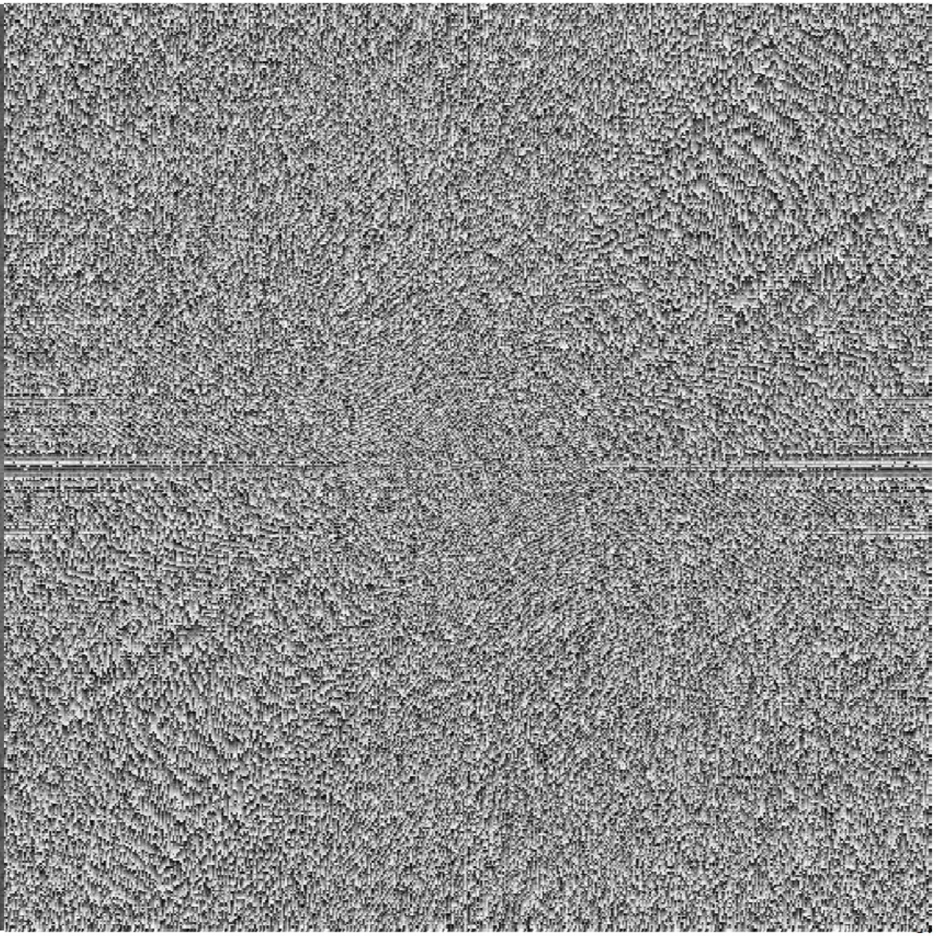
cheetah

What about phase?

Amplitude



Phase



Efros

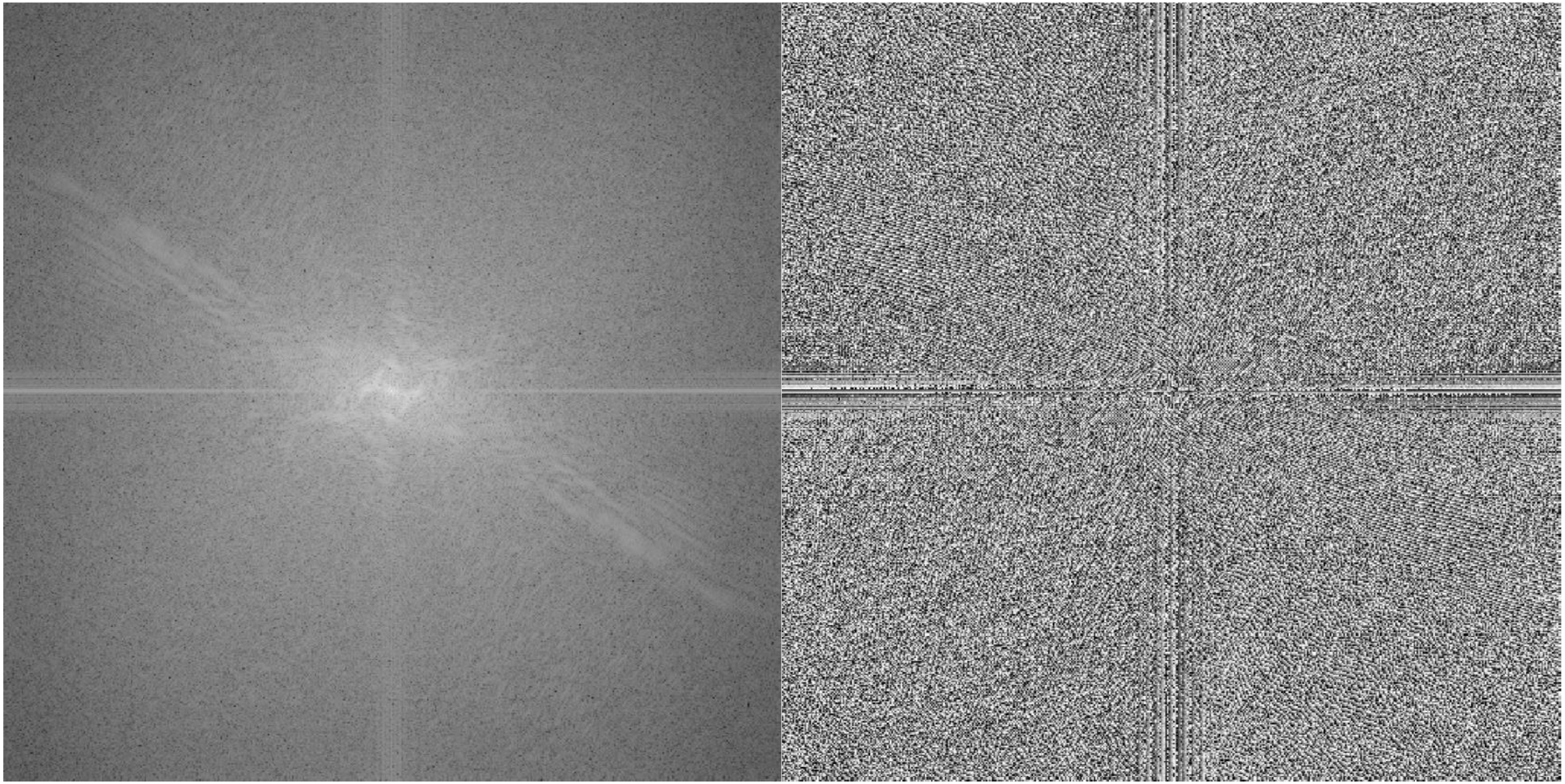
What about phase?

Zebra

Efros

What about phase?

Amplitude Phase



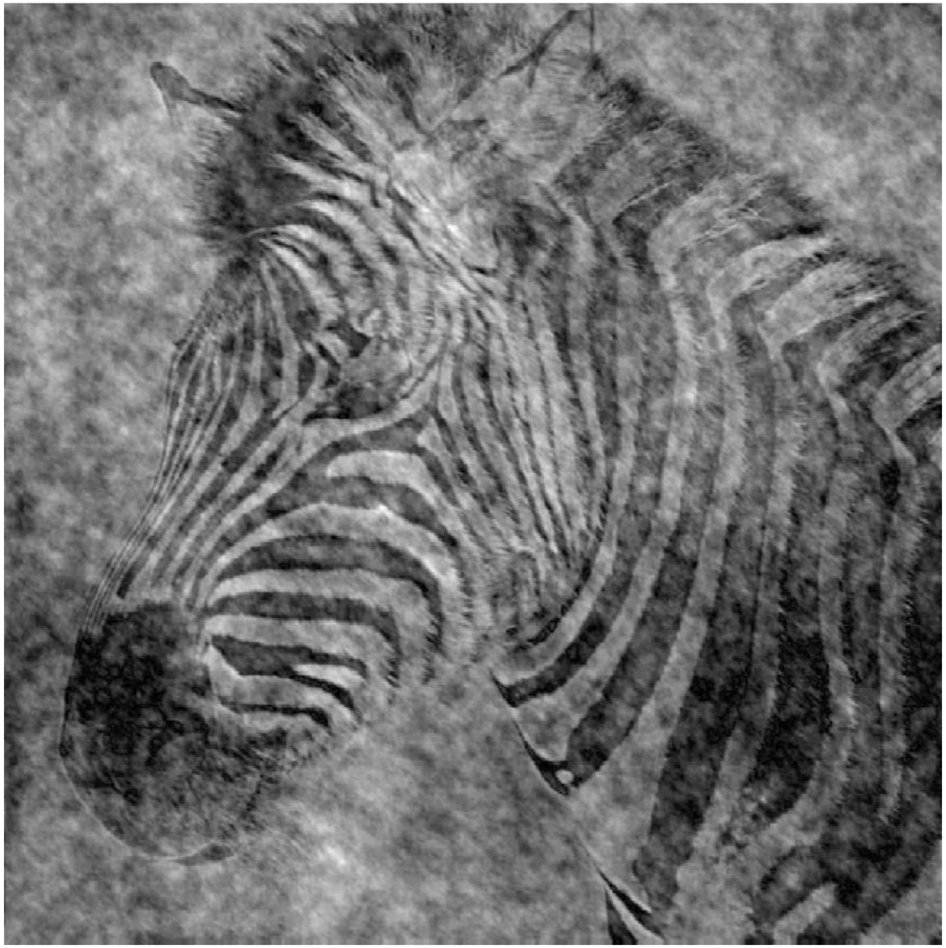
Efros

Think-Pair-Share

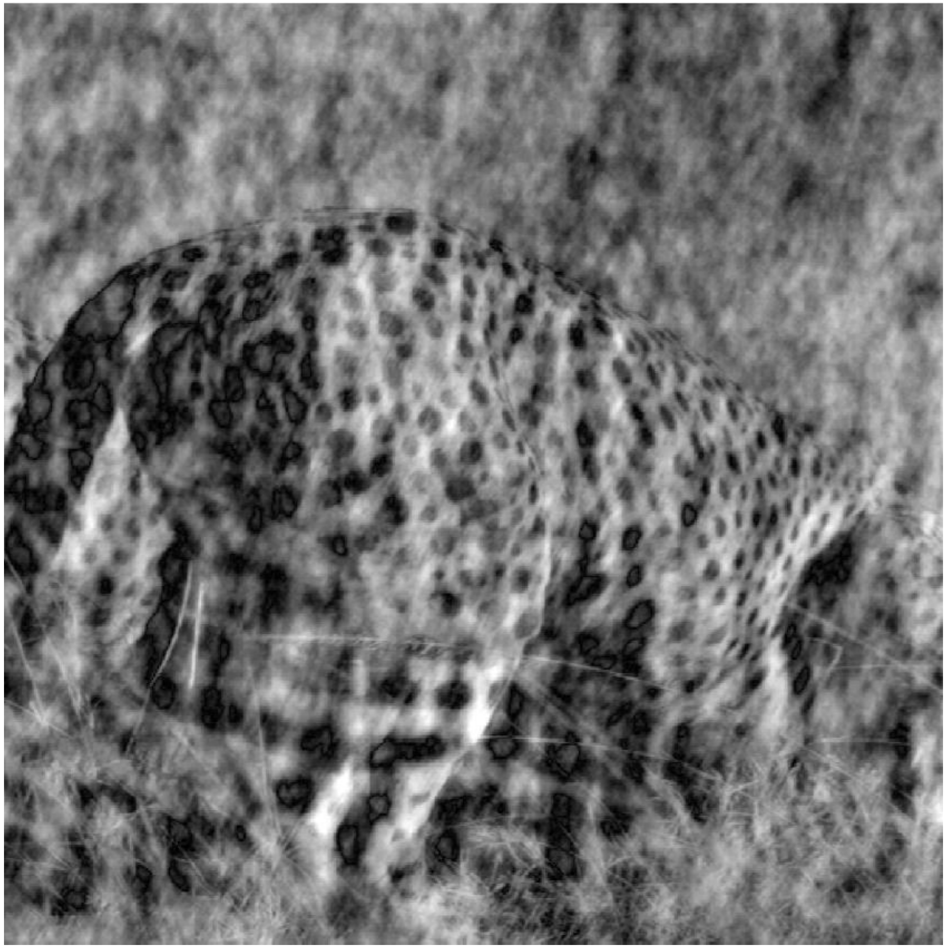
- In Fourier space, where is more of the information that we see in the visual world?
  - Amplitude
  - Phase

Cheebra

Zebra phase, cheetah amplitude



Cheetah phase, zebra amplitude



Efros

- The frequency amplitude of natural images are quite similar
  - Heavy in low frequencies, falling off in high frequencies
  - Will *any* image be like that, or is it a property of the world we live in?
- Most information in the image is carried in the phase, not the amplitude
  - Not quite clear why

Efros

Properties of Fourier Transforms

- Linearity

$$\mathrm{F}[ax(t)+by(t)] = a\,\mathrm{F}[x(t)]+b\,\mathrm{F}[y(t)]$$

- Fourier transform of **a real signal** is symmetric about the origin

$$F(u,v) = F(-u,-v)$$

- The energy of the signal is the same as the energy of its Fourier transform

See Szeliski Book (3.4)

The Convolution Theorem

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

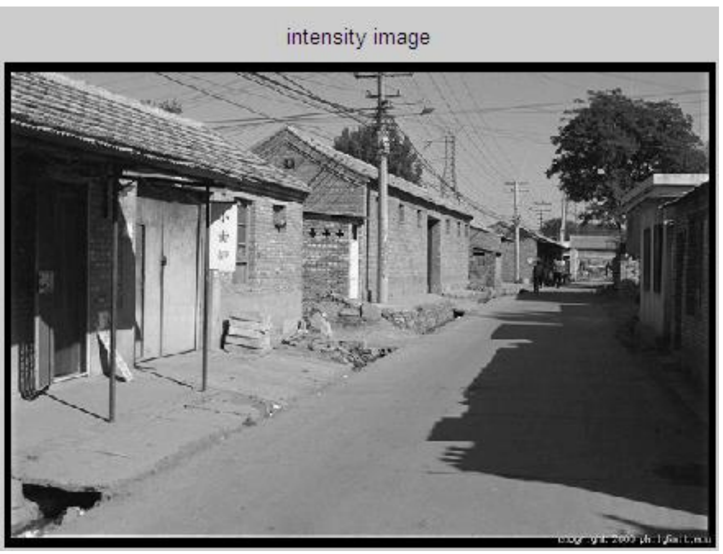
$$\mathrm{F}[g * h] = \mathrm{F}[g]\,\mathrm{F}[h]$$

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

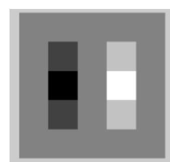
$$g * h = \mathrm{F}^{-1}[\mathrm{F}[g]\,\mathrm{F}[h]]$$

Filtering in spatial domain

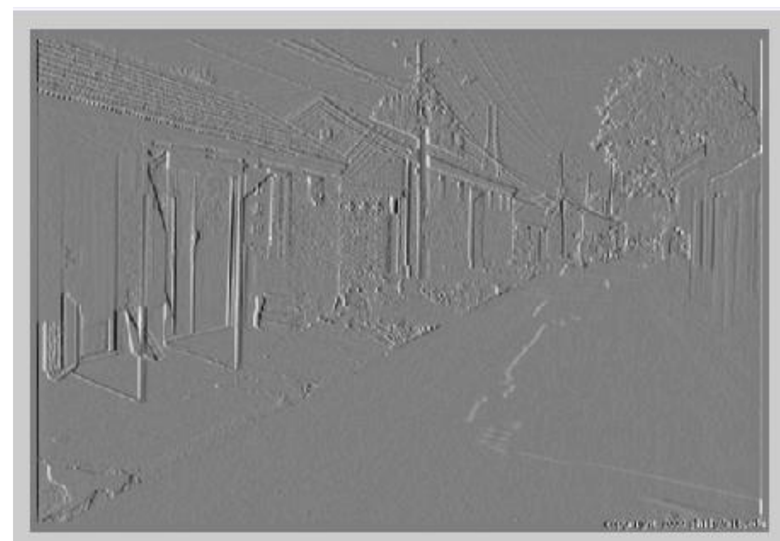
1	0	-1
2	0	-2
1	0	-1



*



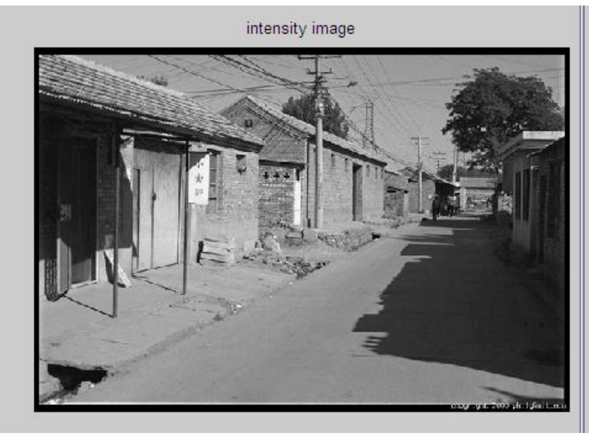
=



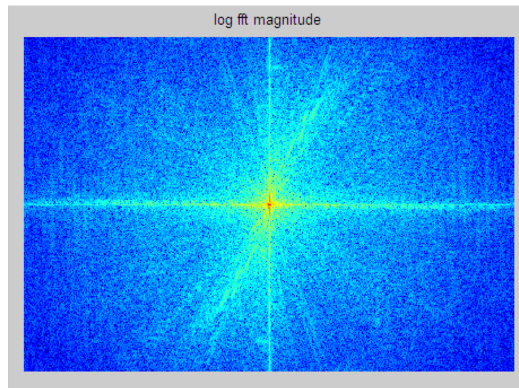
Hays

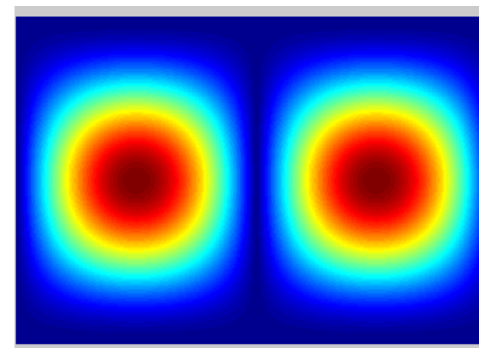
Filtering in frequency domain

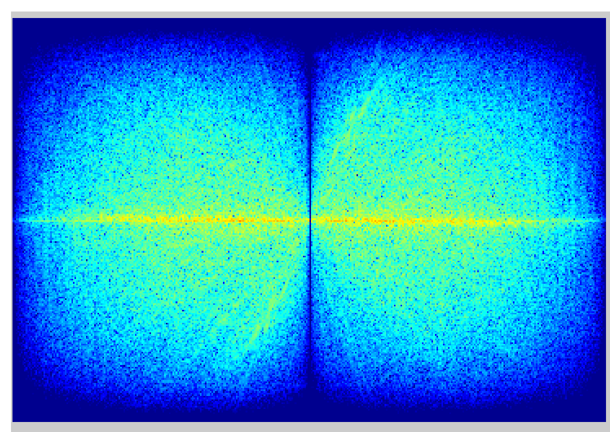
FFT



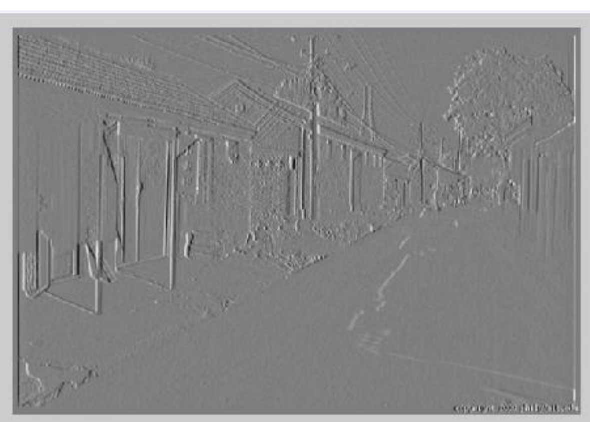
FFT







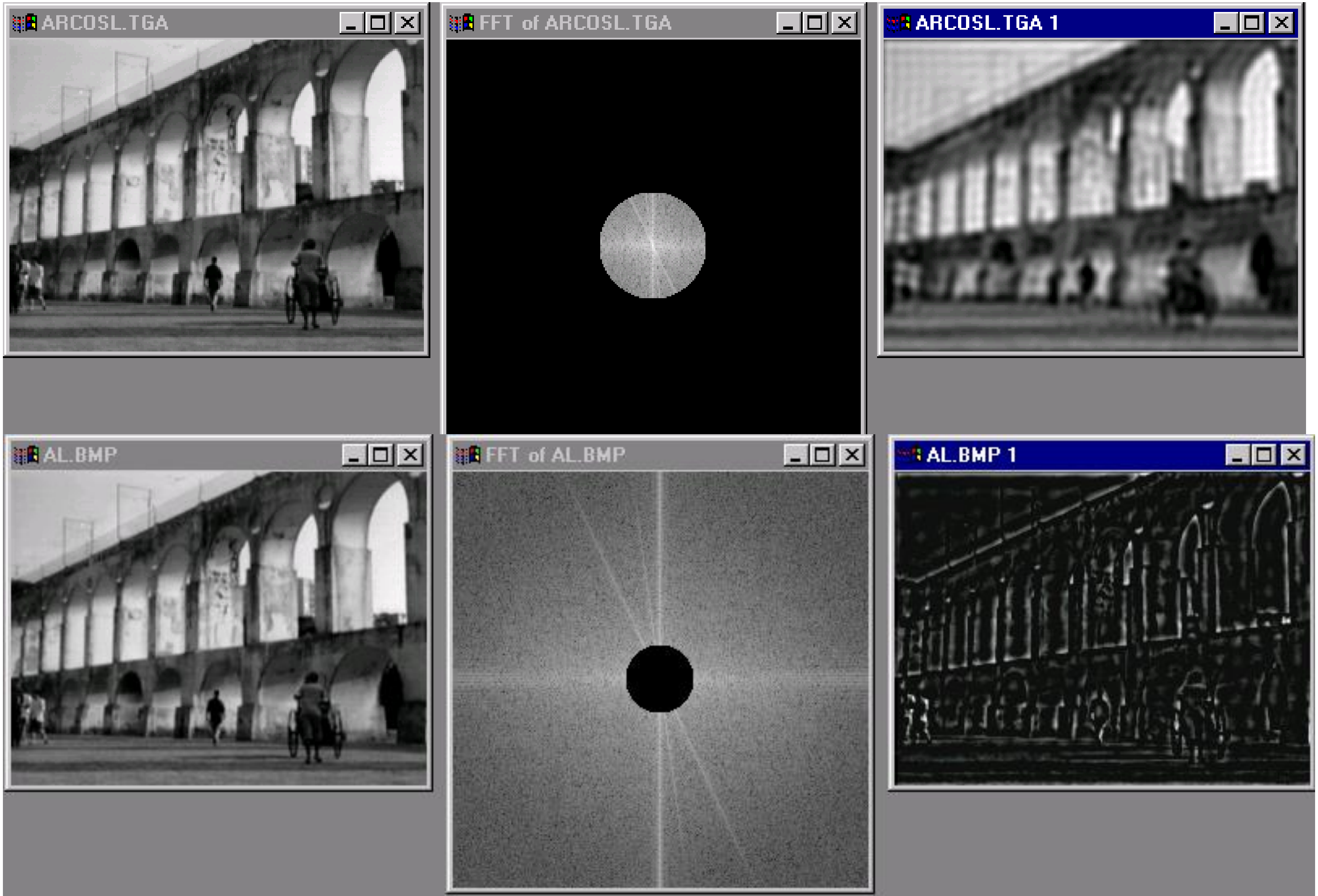
Inverse FFT



Slide: Hoiem

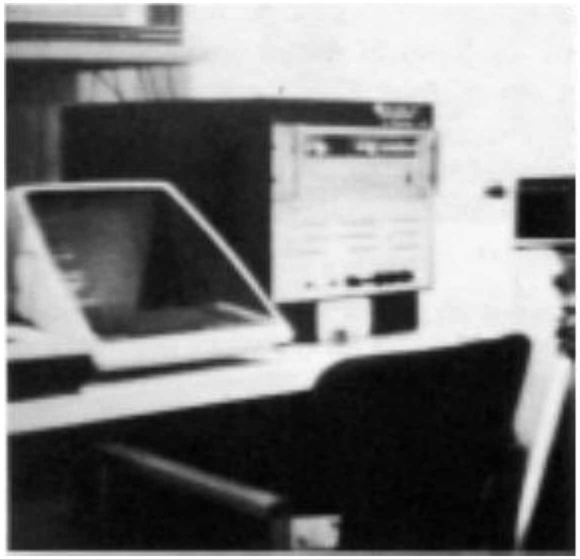
Now we can edit frequencies!

Low and High Pass filtering

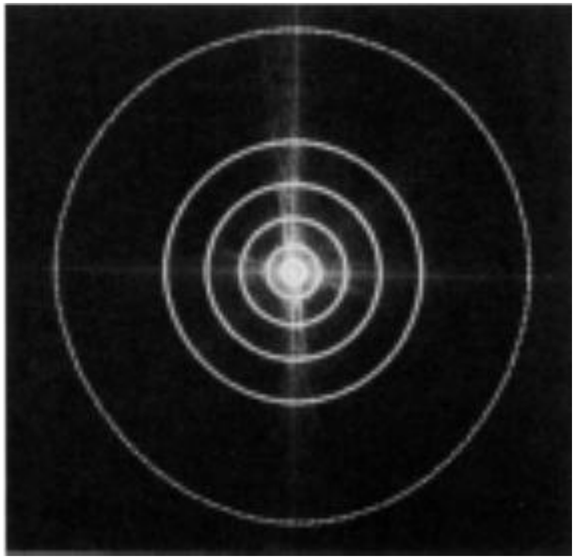


Frequence band

Original image



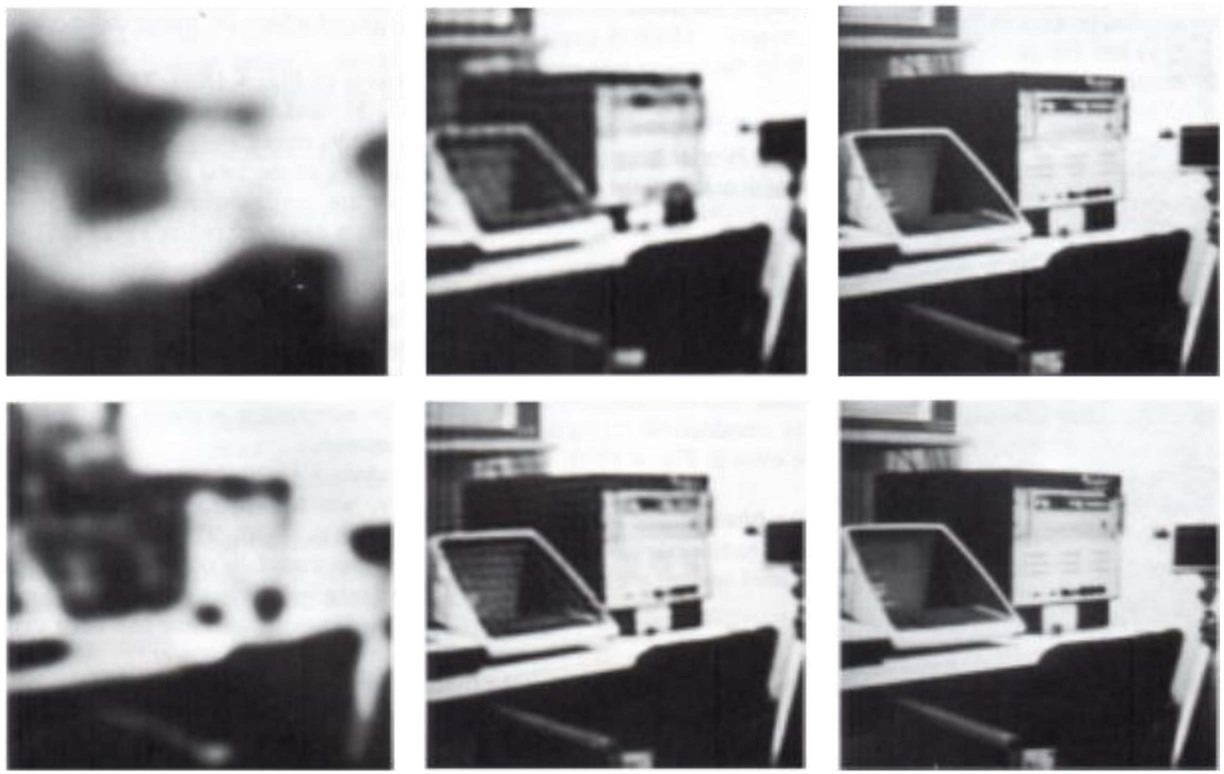
Spectre with filters



90%, 95%, 98%, 99%, 99.5%, 99.9%

Low pass filtering

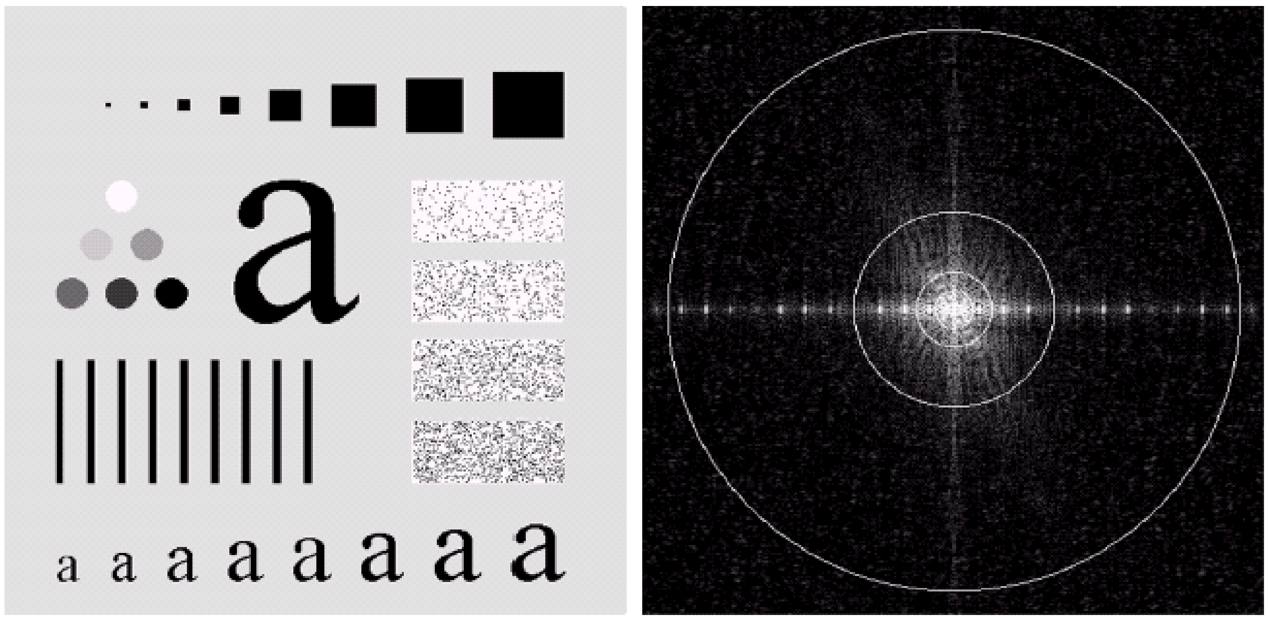
90% 98% 99.5%



95% 99% 99.9%

High pass filtering

Original image | Spectre with filters

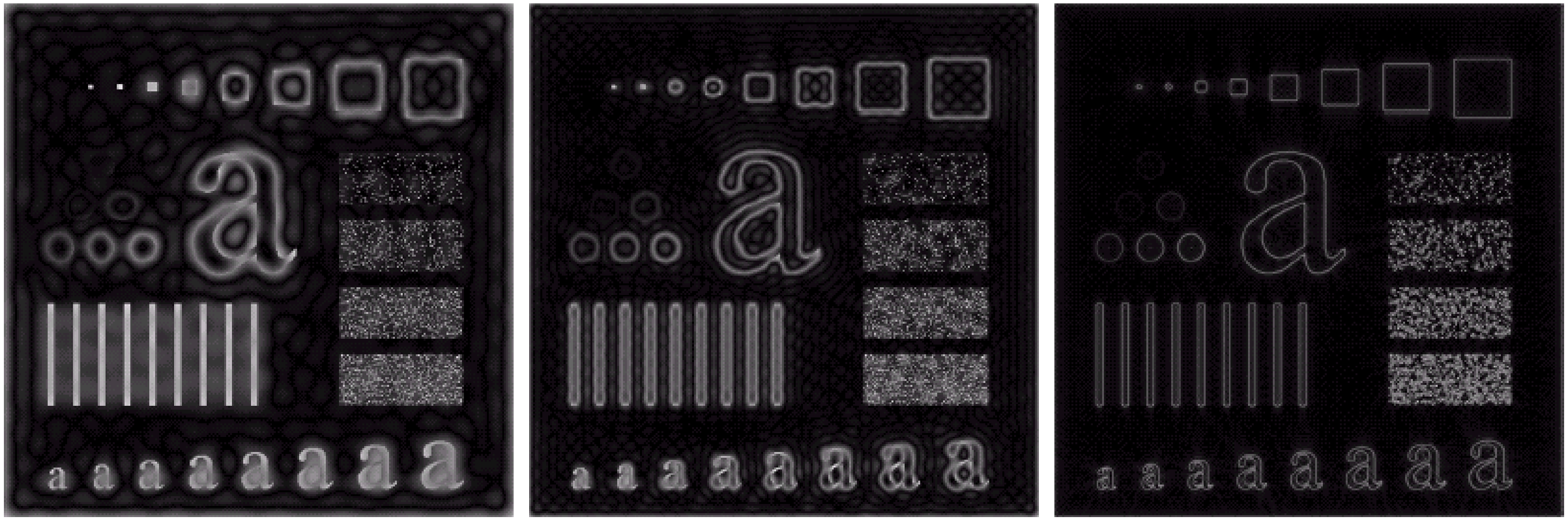


a b

FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

High pass filtering

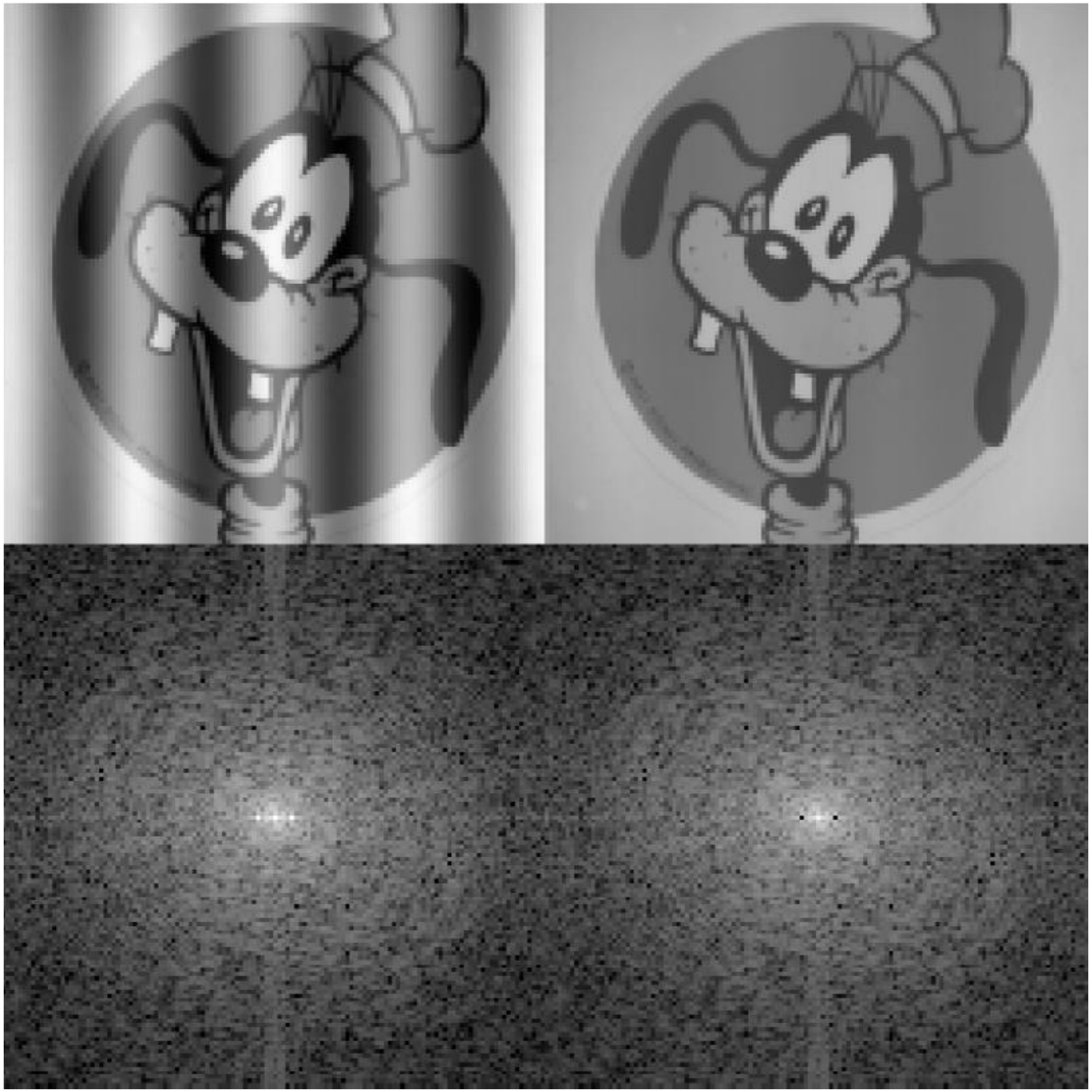


a b c

FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

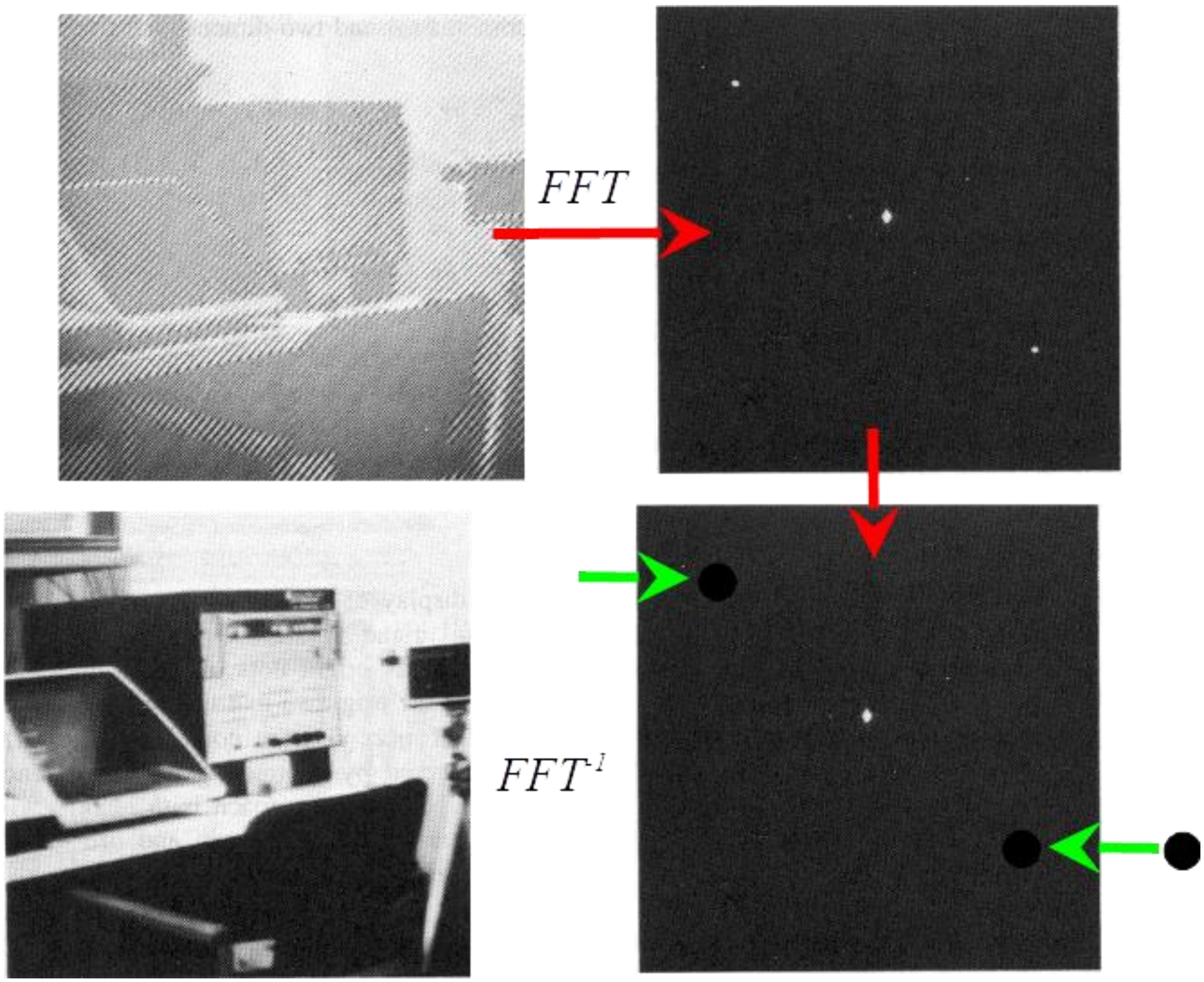
Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

Removing frequency bands



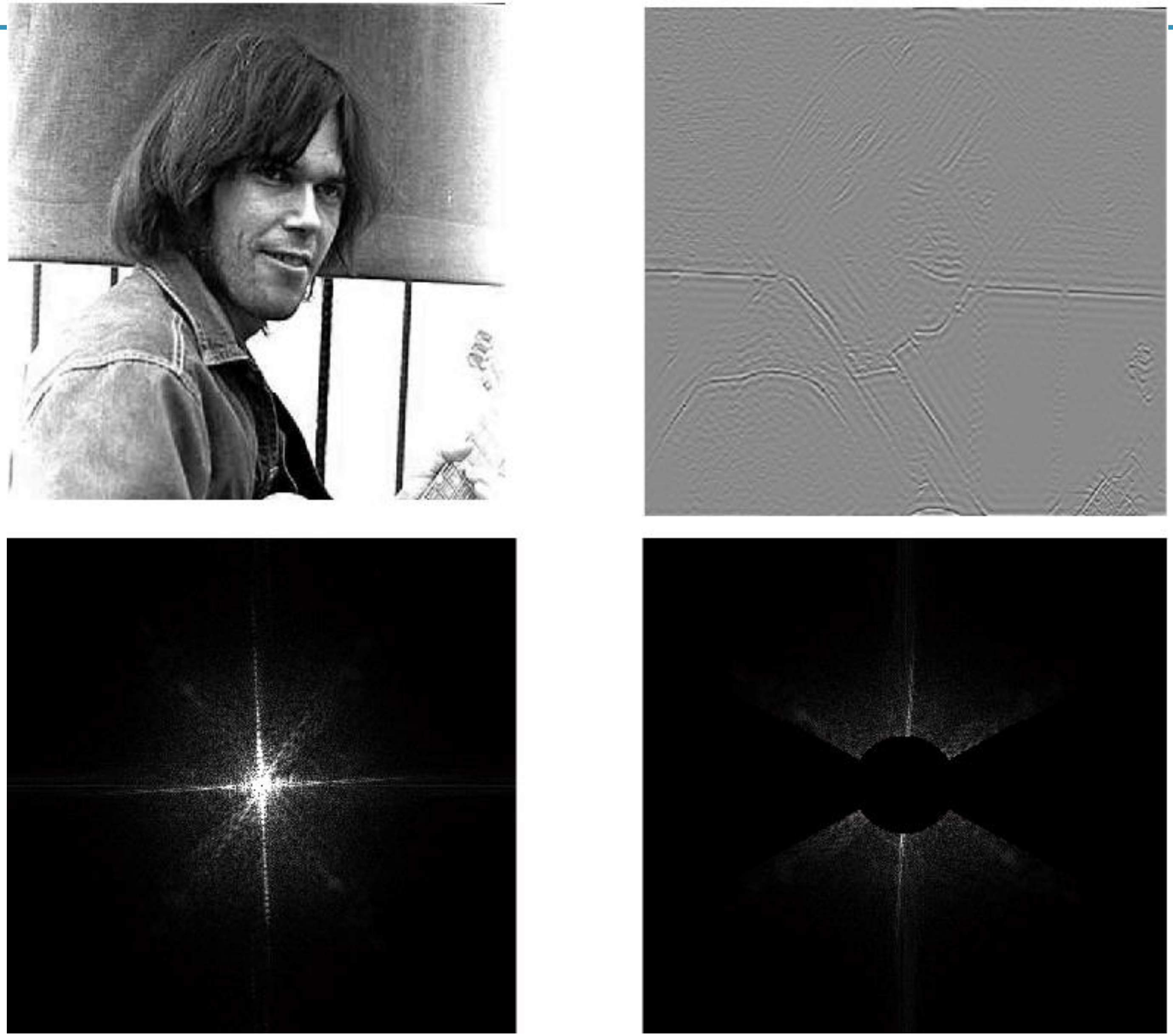
Brayer

Removing frequency bands



Brayer

High pass filtering + orientation



Application: Hybrid Images

When we see an image from far away, we are effectively subsampling it!

A. Oliva, A. Torralba, P.G. Schyns, SIGGRAPH 2006

Application: Hybrid Images

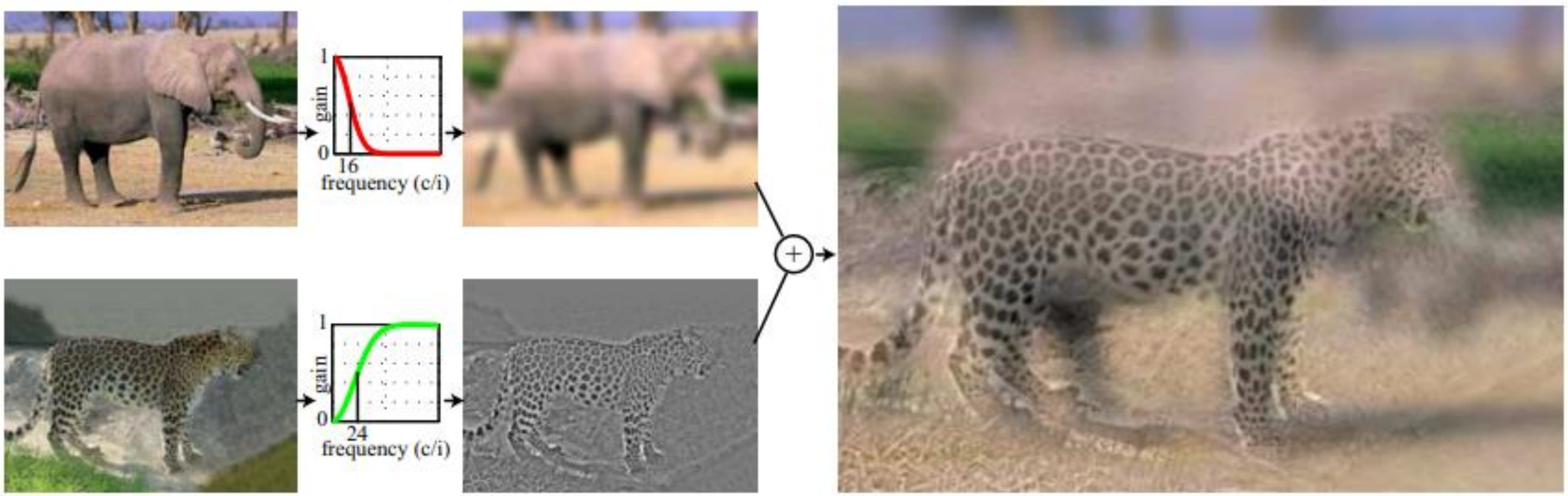


Figure 2: hybrid images are generated by superimposing two images at two different spatial scales: the low-spatial scale is obtained by filtering one image with a low-pass filter, and the high spatial scale is obtained by filtering a second image with a high-pass filter. The final hybrid image is composed by adding these two filtered images.

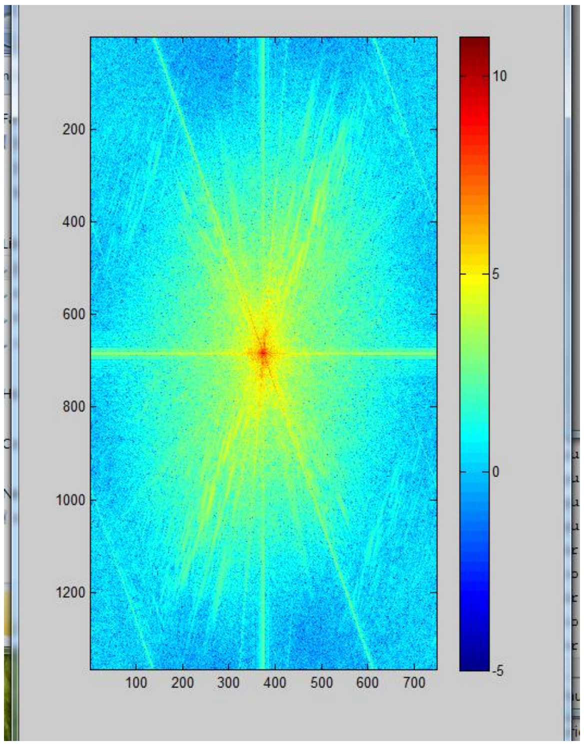
A. Oliva, A. Torralba, P.G. Schyns, SIGGRAPH 2006

Application: Hybrid Images

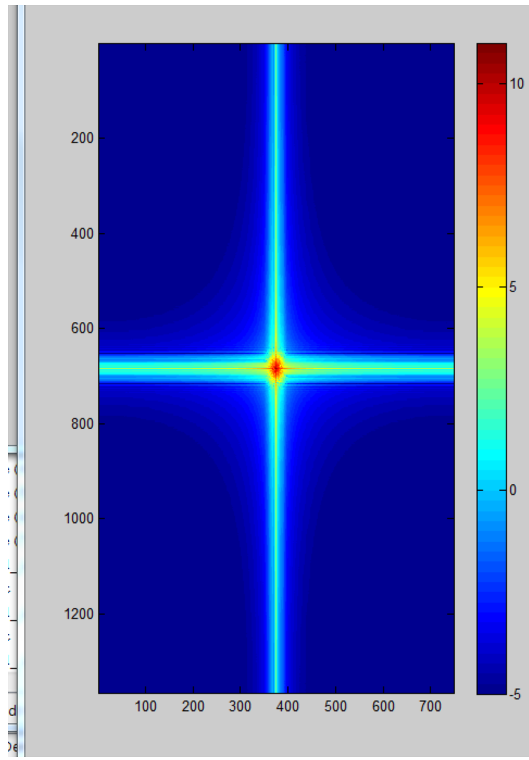
A. Oliva, A. Torralba, P.G. Schyns, SIGGRAPH 2006

Hybrid Image in FFT

Hybrid Image



Low-passed Image



High-passed Image