Cvision – 2a
Digital Imaging

João Paulo Silva Cunha
(jcunha@det.ua.pt)

IEETA / Universidade de Aveiro

Outline

- Image sensors
- Fourier, sampling and quantization
- Data structures for digital images
- Histograms

Acknowledgements: Most of this course is based on the excellent courses offered by Prof. Shree Nayar at Columbia University, USA and by Prof. Srinivasa Narasimhan at CMU, USA. This was also based on Prof. Miguel Coimbra's slides. Please acknowledge the original source when reusing these slides for academic purposes.

Topic: Image Sensors

- Image sensors
- Fourier, sampling and quantization
- Data structures for digital images
- Histograms

Image Sensors

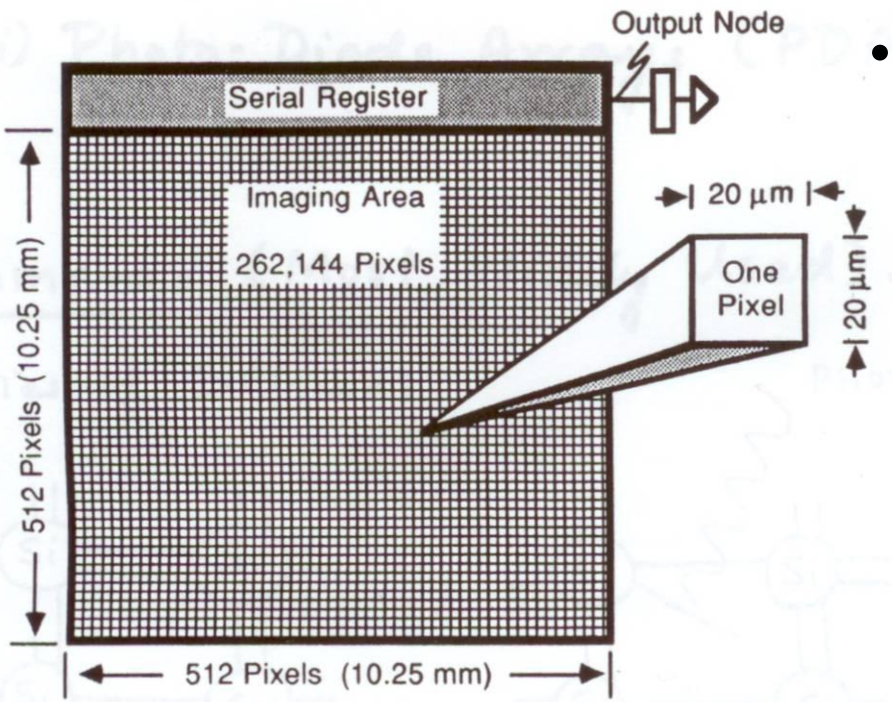
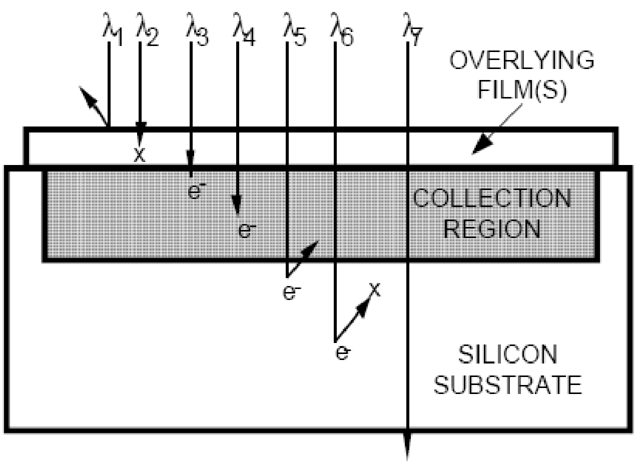


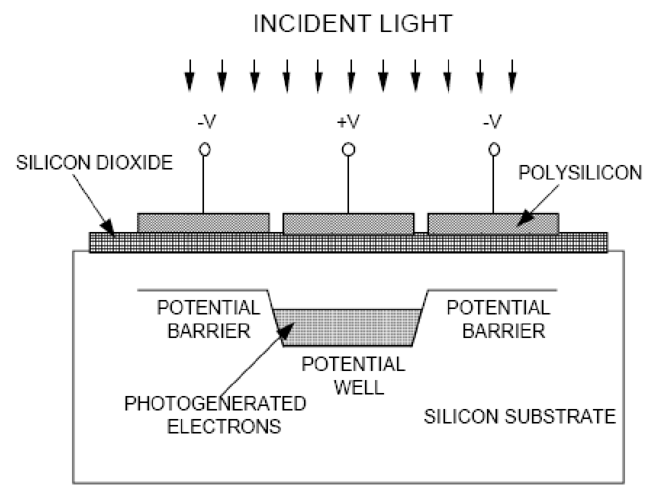
FIG. 4. Typical 512 × 512 CCD.

- Considerations
  - Speed
  - Resolution
  - Signal / Noise Ratio
  - Cost

Image Sensors

- Convert light into an electric charge



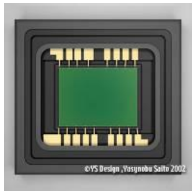


CCD (charge coupled device)

Higher dynamic range

High uniformity

Lower noise

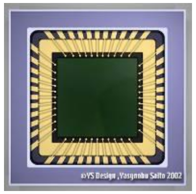


CMOS (complementary metal Oxide semiconductor)

Lower voltage

Higher speed

Lower system complexity

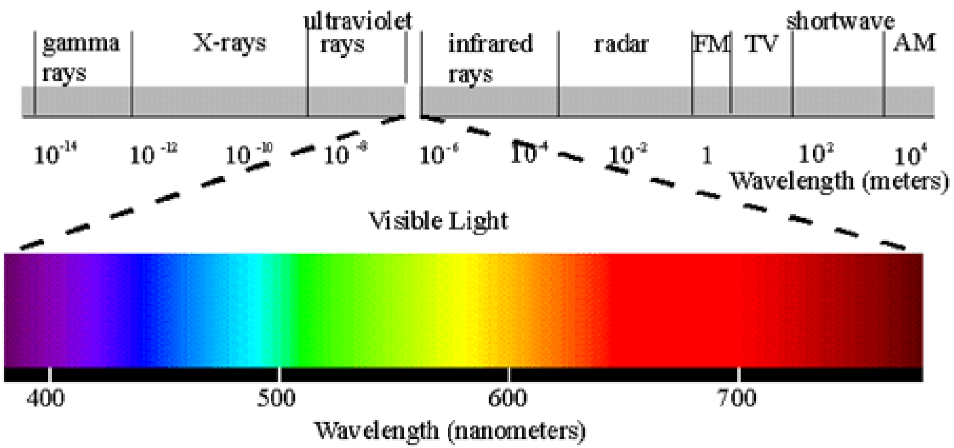


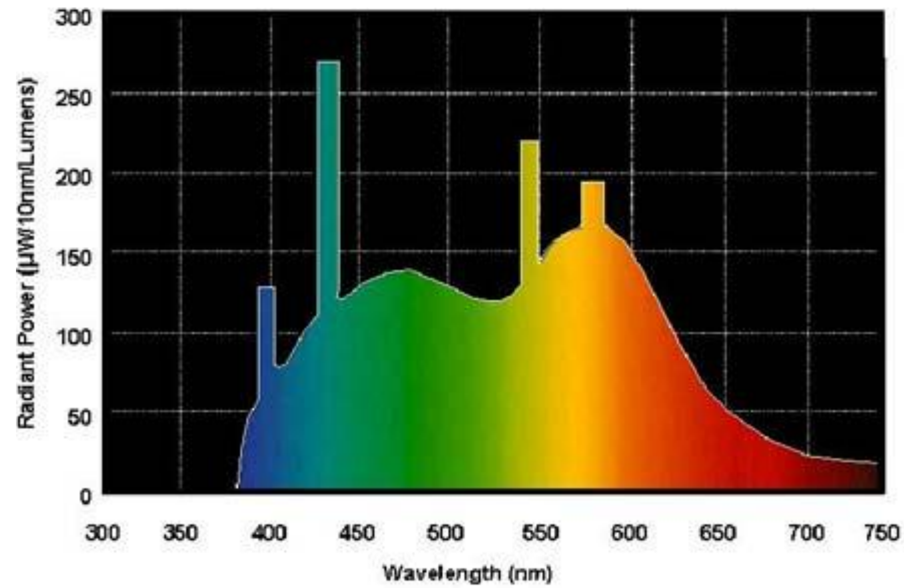
CCD Performance Characteristics

- Linearity Principle: Incoming photon flux vs. Output Signal
  - Sometimes cameras are made non-linear on purpose.
  - Calibration must be done (using reflectance charts)---covered later
- Dark Current Noise: Non-zero output signal when incoming light is zero
- Sensitivity: Minimum detectable signal produced by camera

Sensing Brightness

Incoming light has a **spectral distribution** $p(\lambda)$



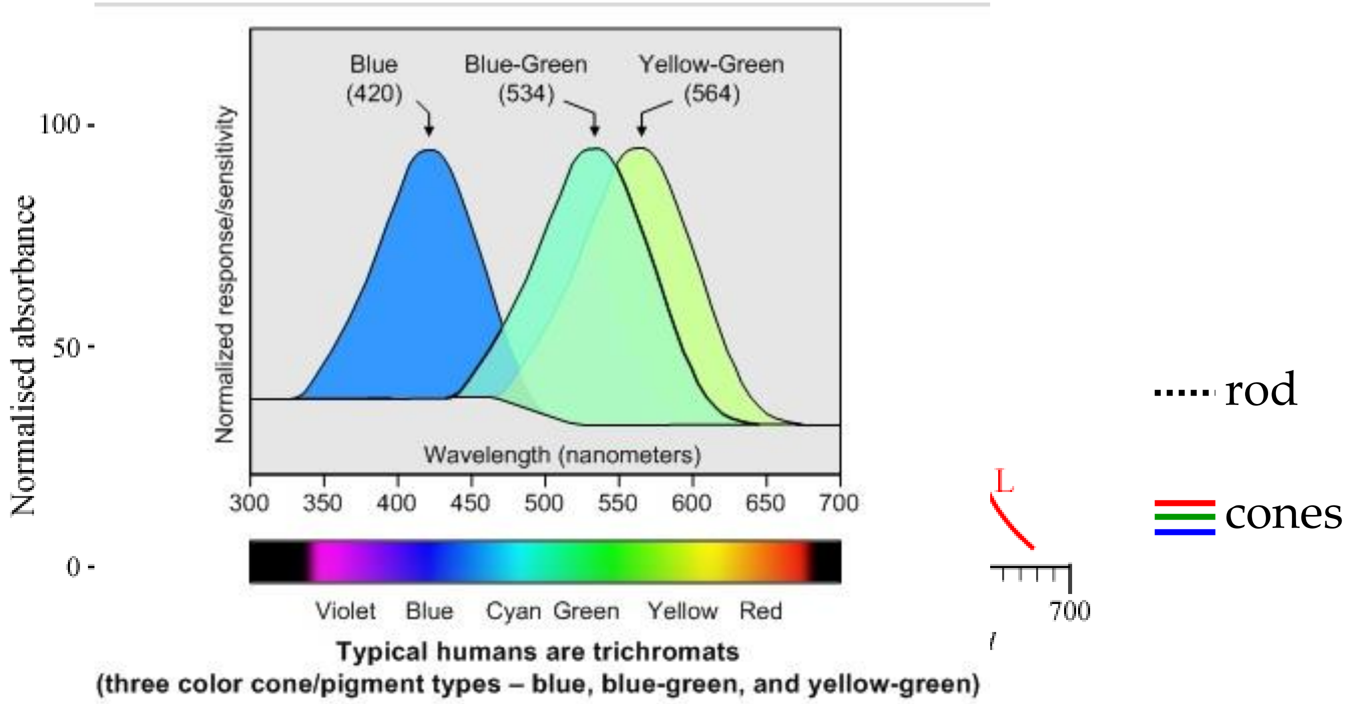


So the pixel intensity becomes

$$I = k \int_{-\infty}^{\infty} q(\lambda)\, p(\lambda)\, d\lambda$$

How do we sense colour?

- Do we have infinite number of filters?

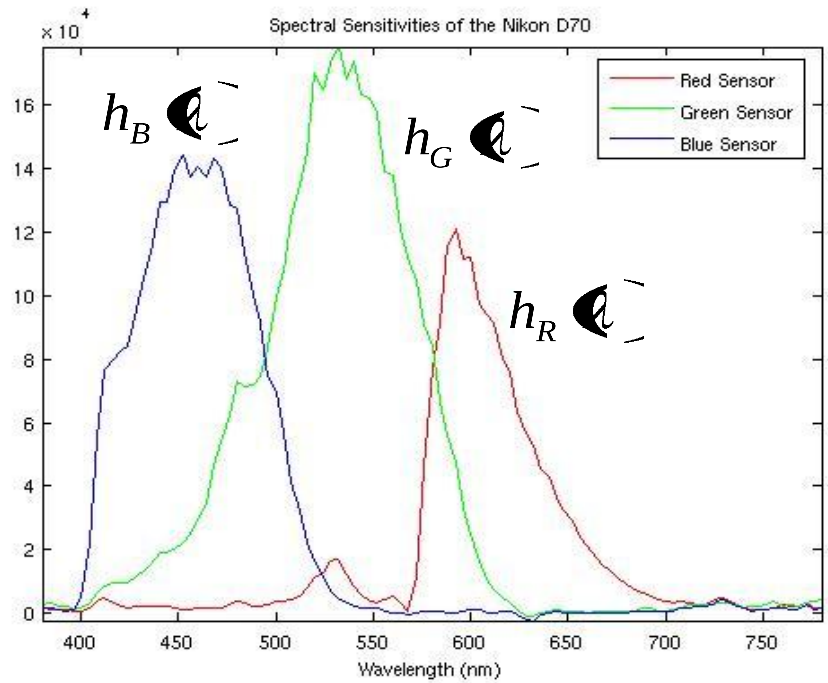


Three filters of different spectral responses

Sensing Colour

- Tristimulus (trichromatic) values (I_R, I_G, I_B)

Camera's spectral response functions: $h_R(\lambda), h_G(\lambda), h_B(\lambda)$

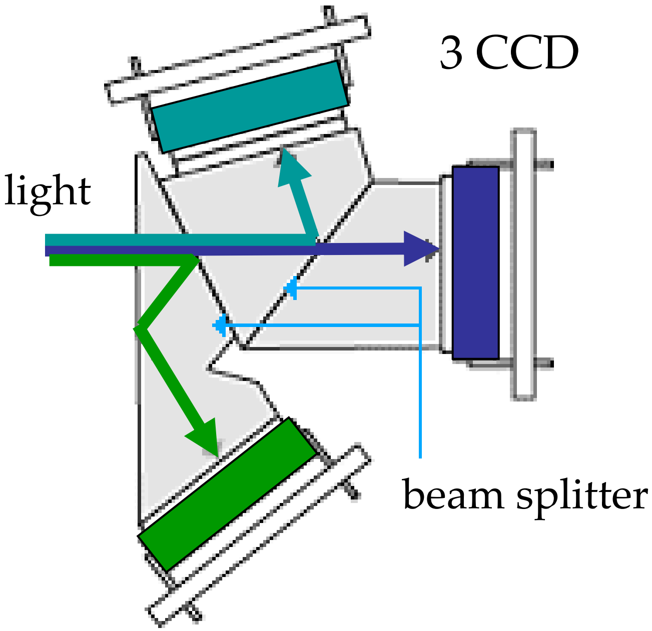


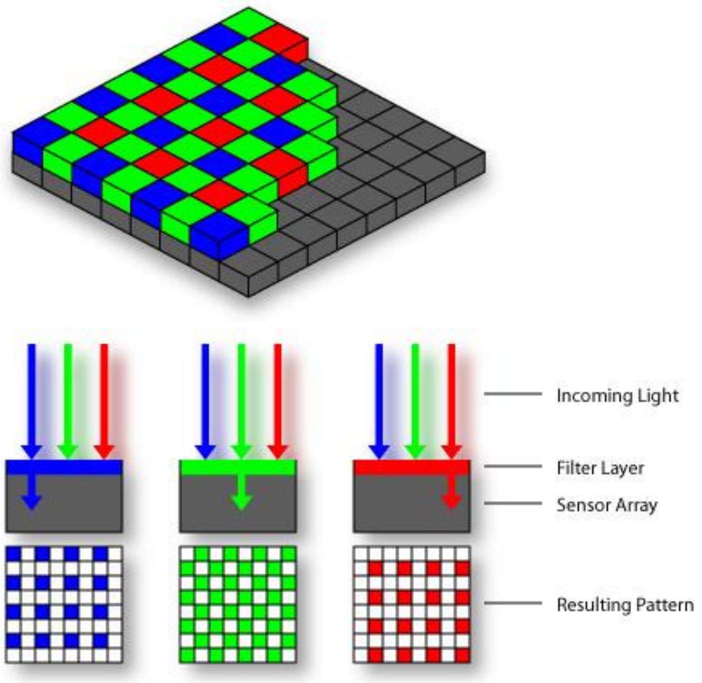
$$I_R = k\int_{-\infty}^{\infty} h_R(\lambda)\,p(\lambda)\,d\lambda$$

$$I_G = k\int_{-\infty}^{\infty} h_G(\lambda)\,p(\lambda)\,d\lambda$$

$$I_B = k\int_{-\infty}^{\infty} h_B(\lambda)\,p(\lambda)\,d\lambda$$

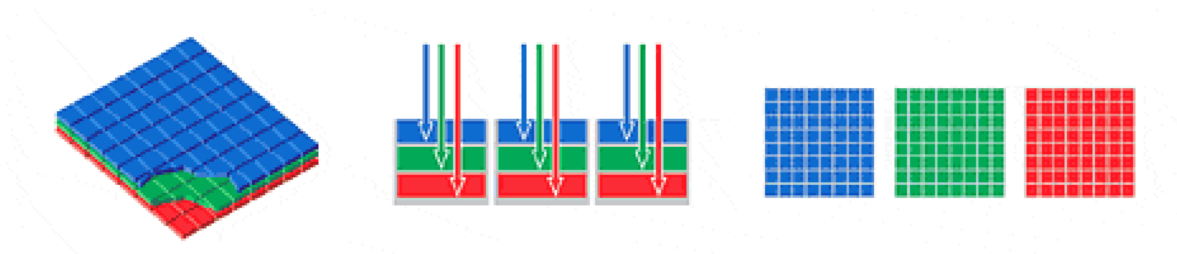
Sensing Colour



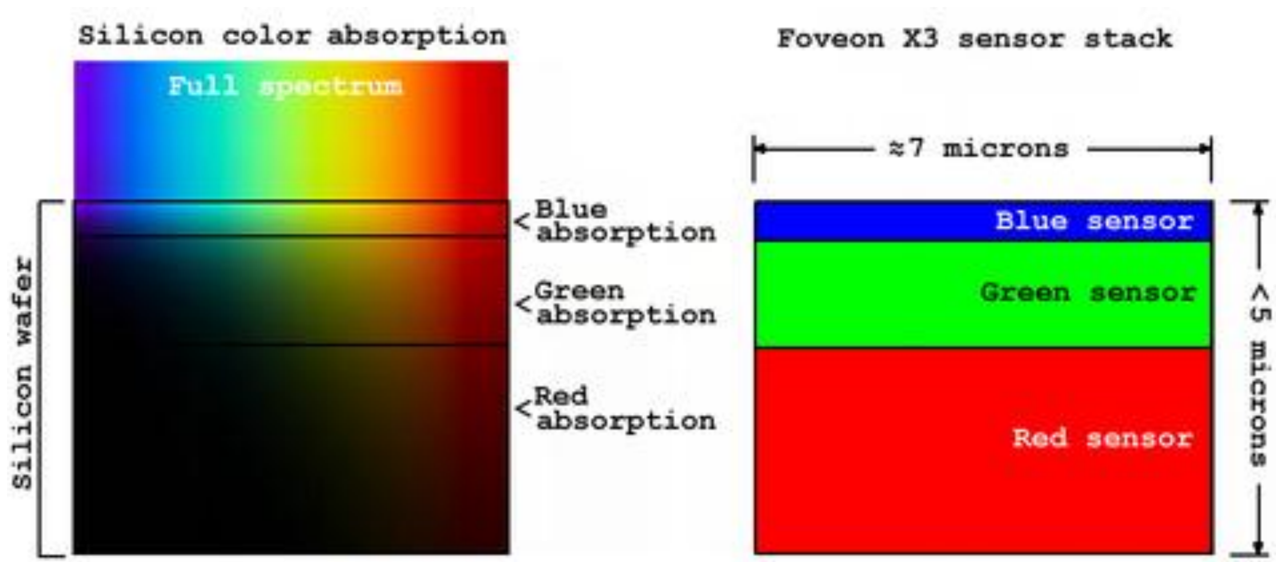


Bayer pattern

Sensing Colour



Foveon X3™



Topic: Fourier analysis

- Image sensors
- Fourier, sampling and quantization
- Data structures for digital images
- Histograms

How to Represent Signals?

- Option 1: Taylor series represents any function using polynomials.

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \ldots.$$

- Polynomials are not the best - unstable and not very physically meaningful.

- Option 2: Easier to talk about "signals" in terms of its "frequencies"

  (how fast/often signals change, etc).

Jean Baptiste Joseph Fourier (1768-1830)

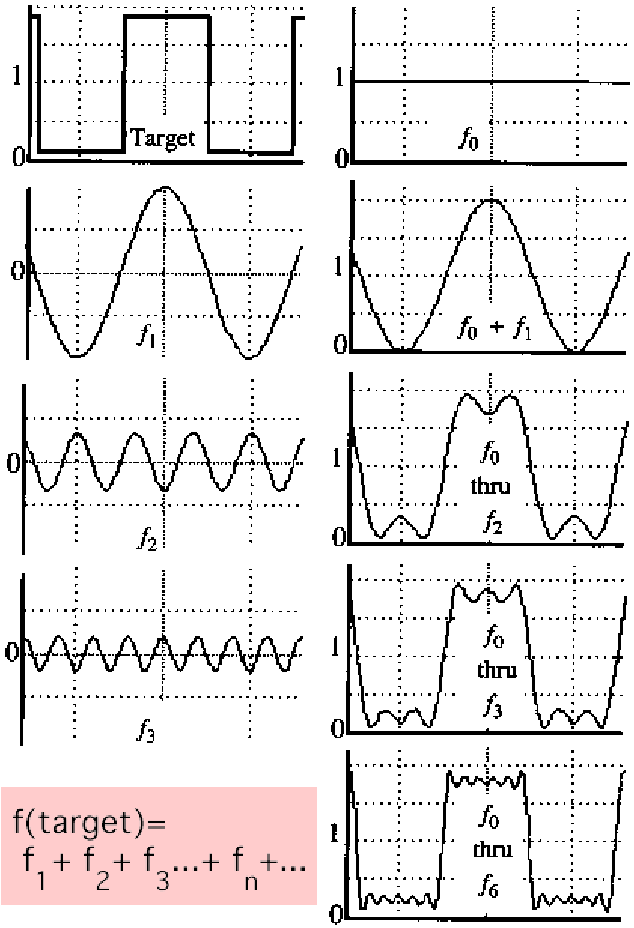
- Had a crazy idea (1807):
- **Any** periodic function can be rewritten as a weighted sum of Sines and Cosines of different frequencies.
- Don’t believe it?
  - Neither did Lagrange, Laplace, Poisson and other big wigs
  - Not translated into English until 1878!
- But it’s true!
  - called Fourier Series
  - Possibly the greatest tool used in Engineering

A Sum of Sinusoids

- Our building block:

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

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



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

Fourier Transform

- We want to understand the frequency ω of our signal. So, let's reparametrize the signal by ω instead of *x*:

$$f(x) \longrightarrow \boxed{\text{Fourier Transform}} \longrightarrow F(\omega)$$

- For every ω from 0 to inf, ***F(ω)*** holds the amplitude *A* and phase ϕ of the corresponding sine
  - How can *F* hold both? Complex number trick!

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

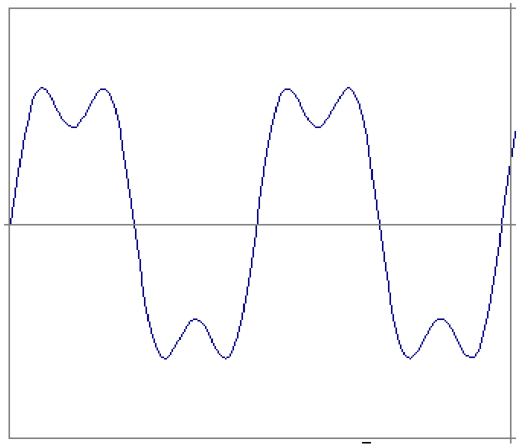
$$F(\omega) = R(\omega) + iI(\omega)$$

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

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

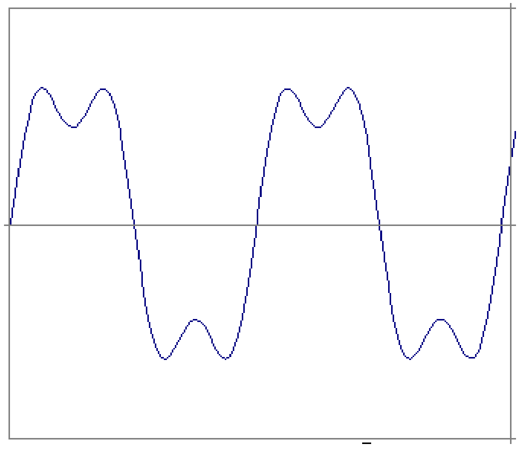
Time and Frequency

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

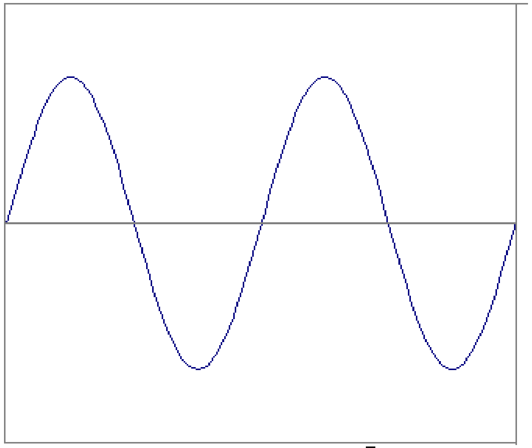


Time and Frequency

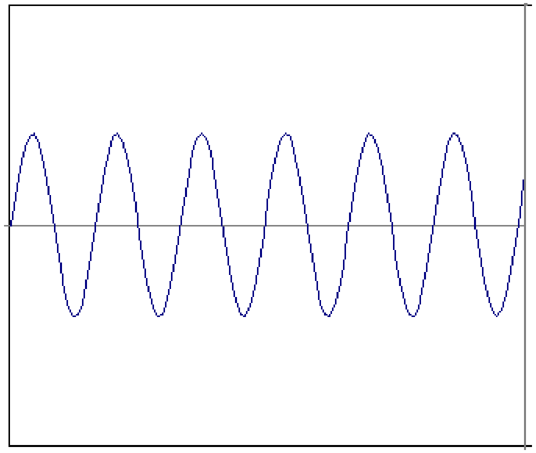
- example : $g(t) = \sin(2\Pi f\, t) + (1/3)\sin(2\Pi (3f)\, t)$



=

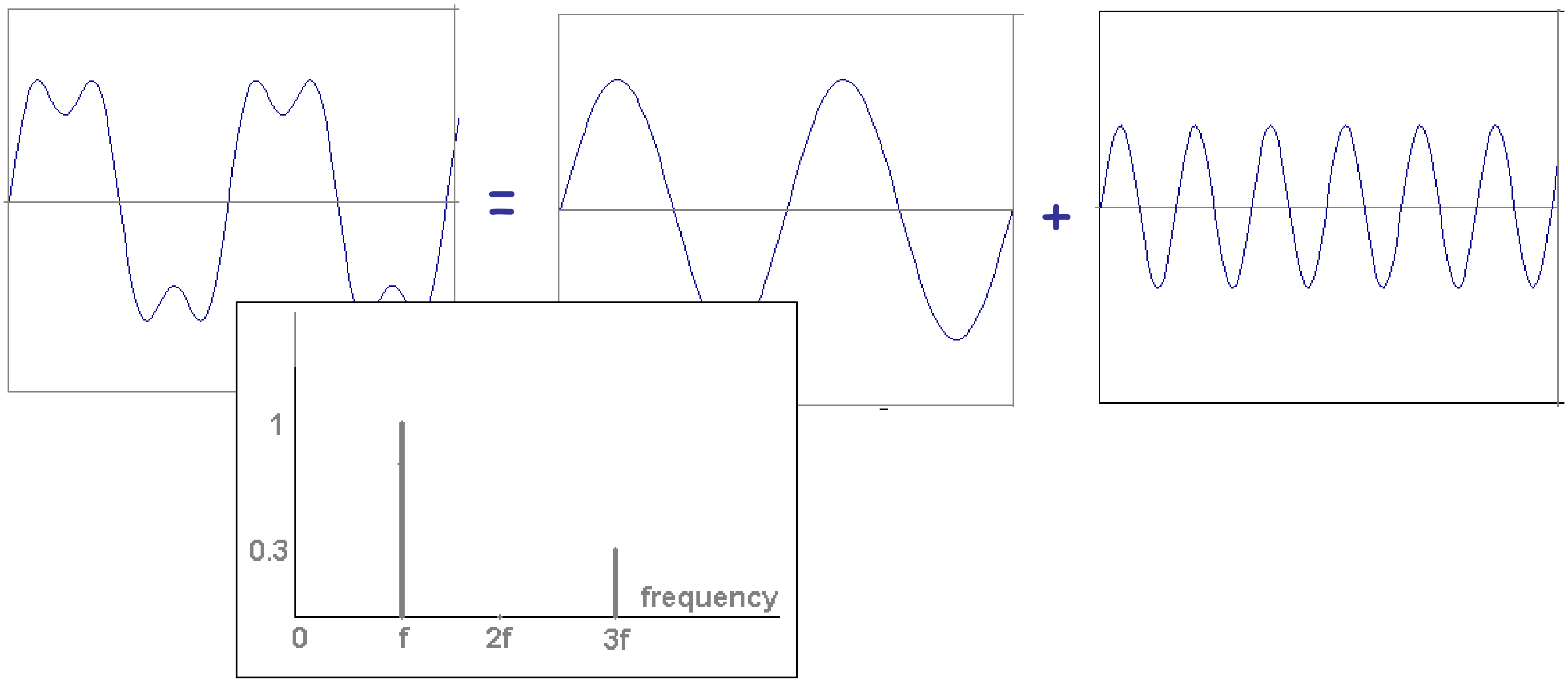


+



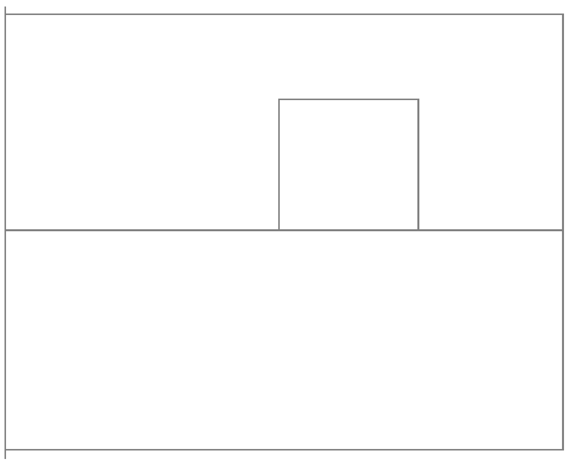
Frequency Spectra

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

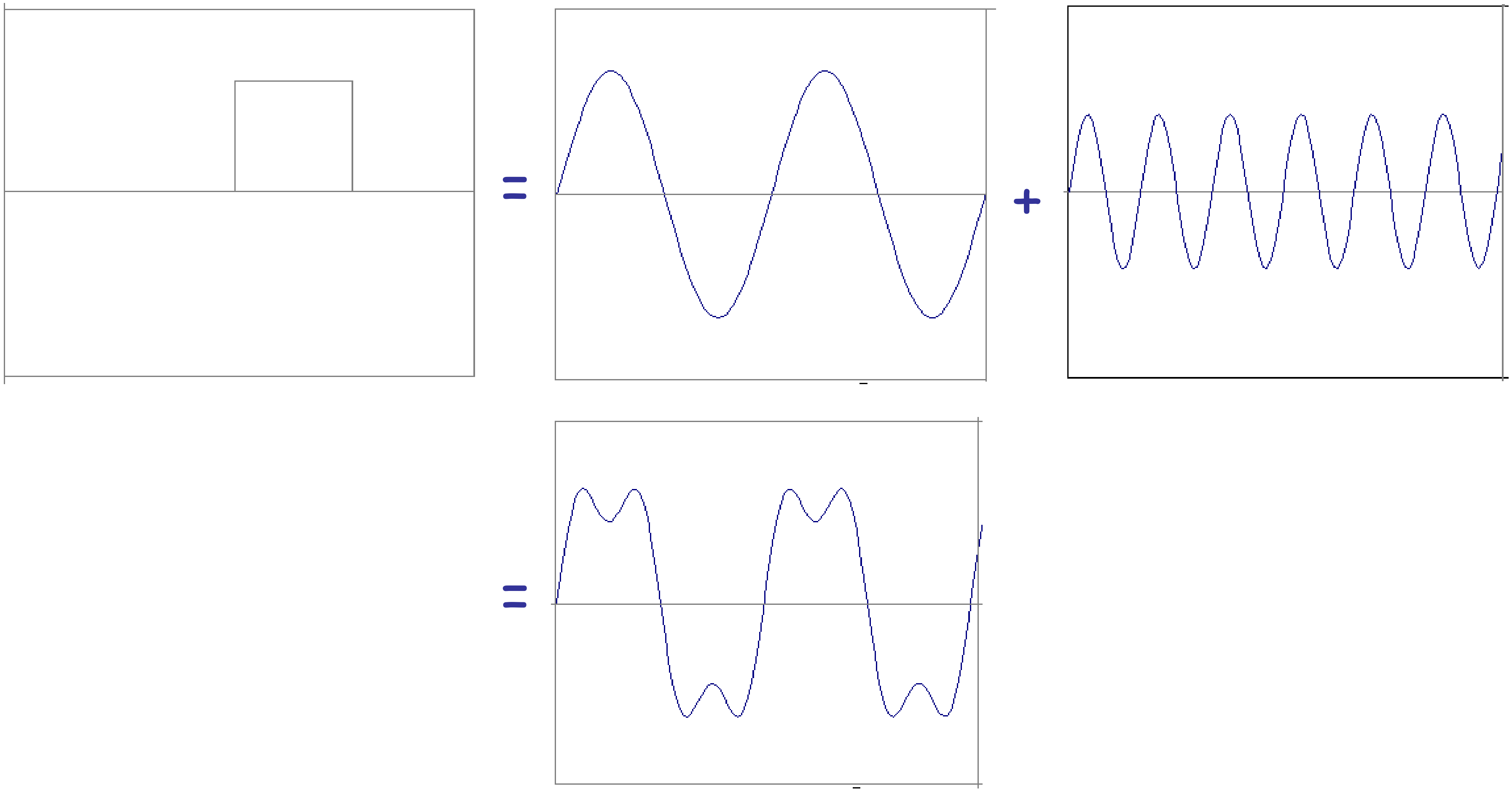


Frequency Spectra

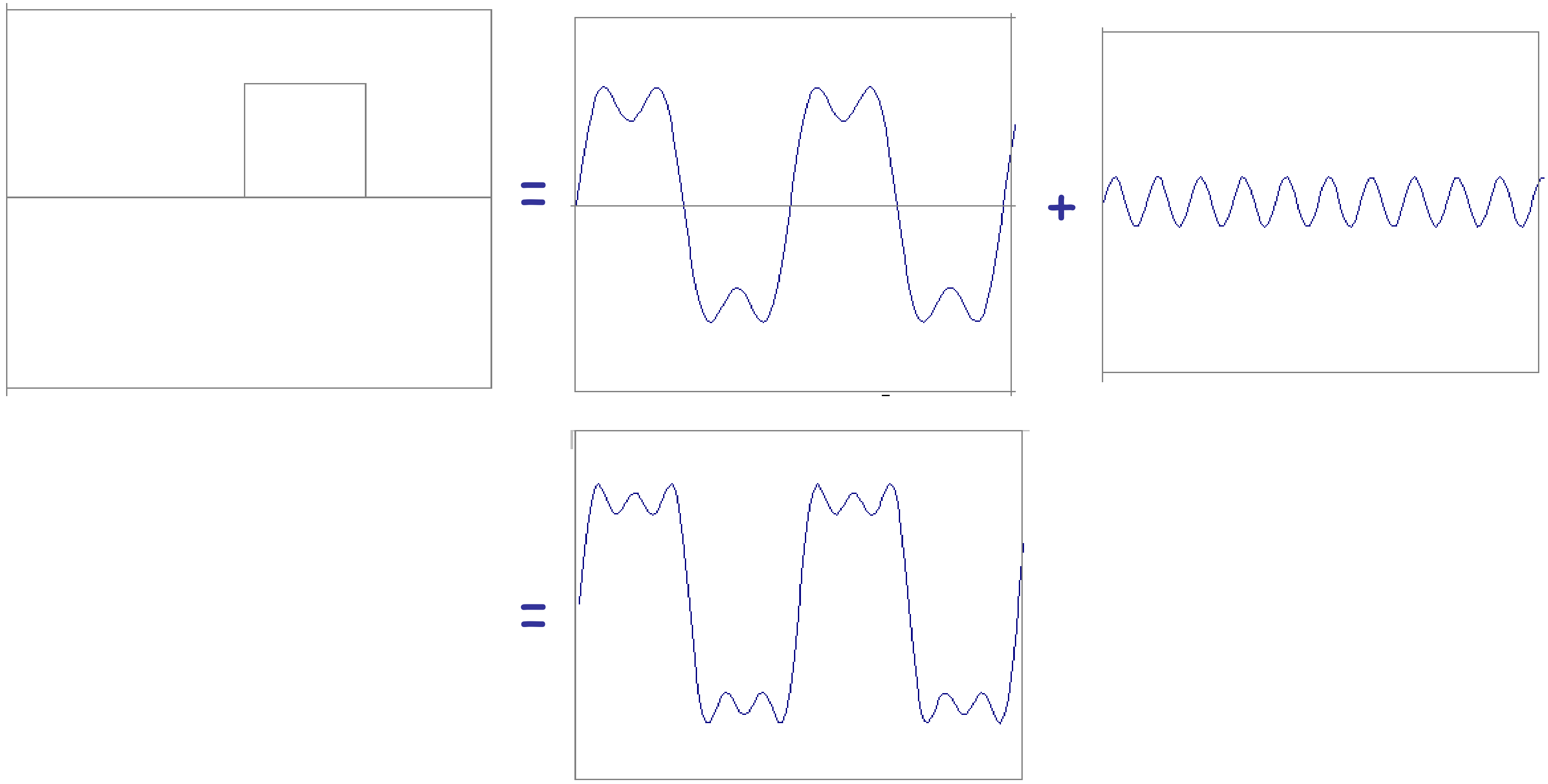
- If I have a more “wideband” signal, I need more sines in the sum to approximate it



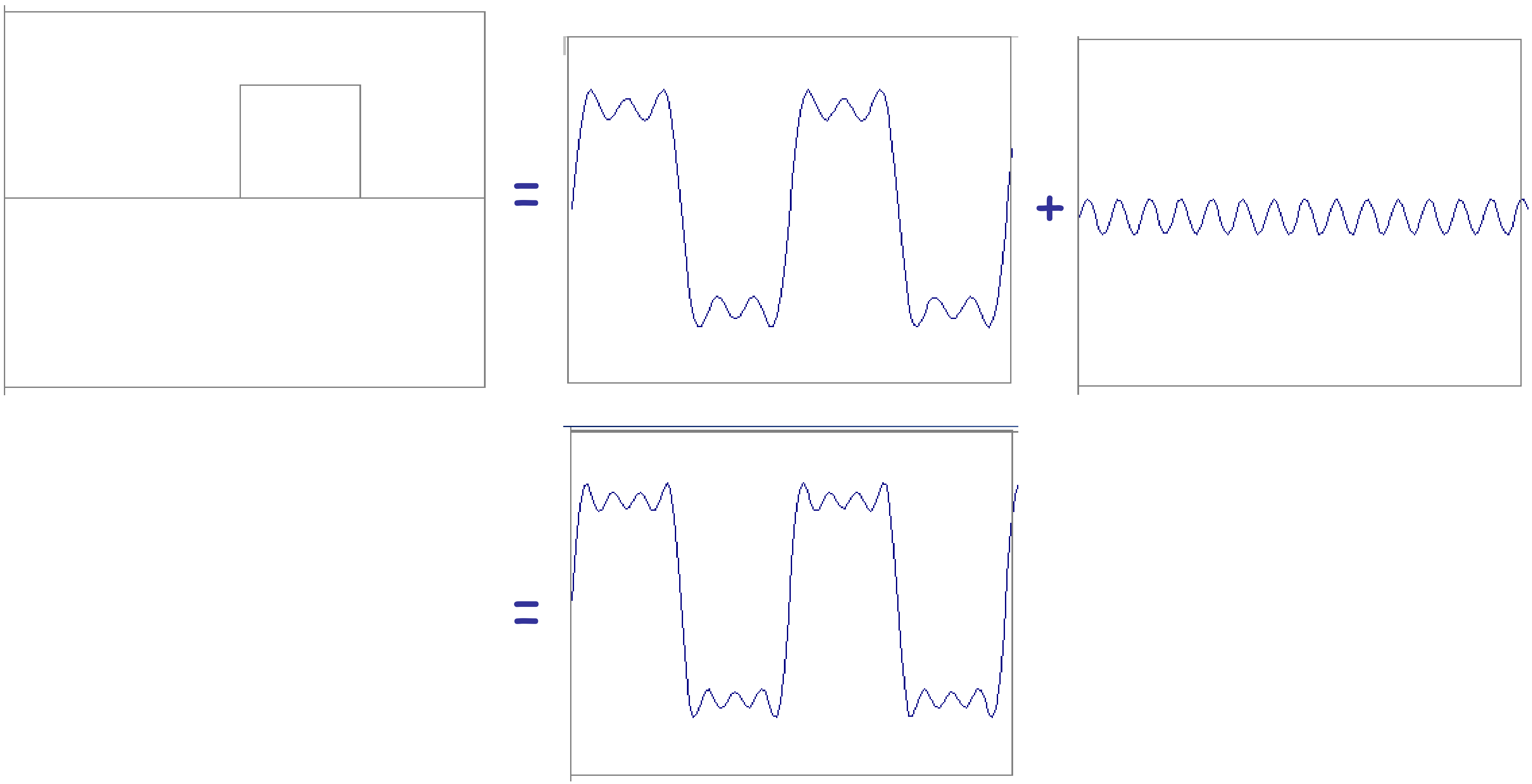
Frequency Spectra



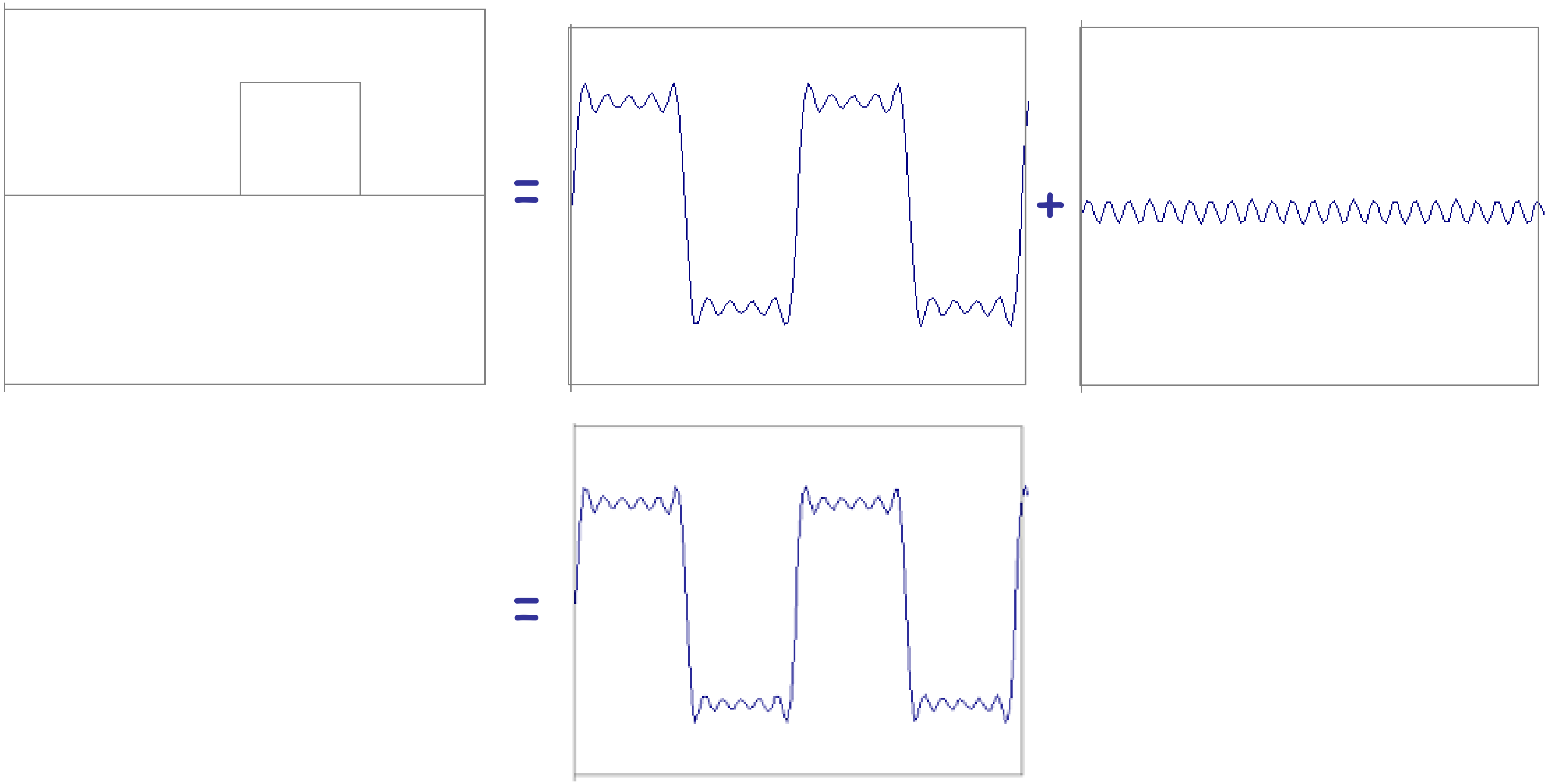
Frequency Spectra



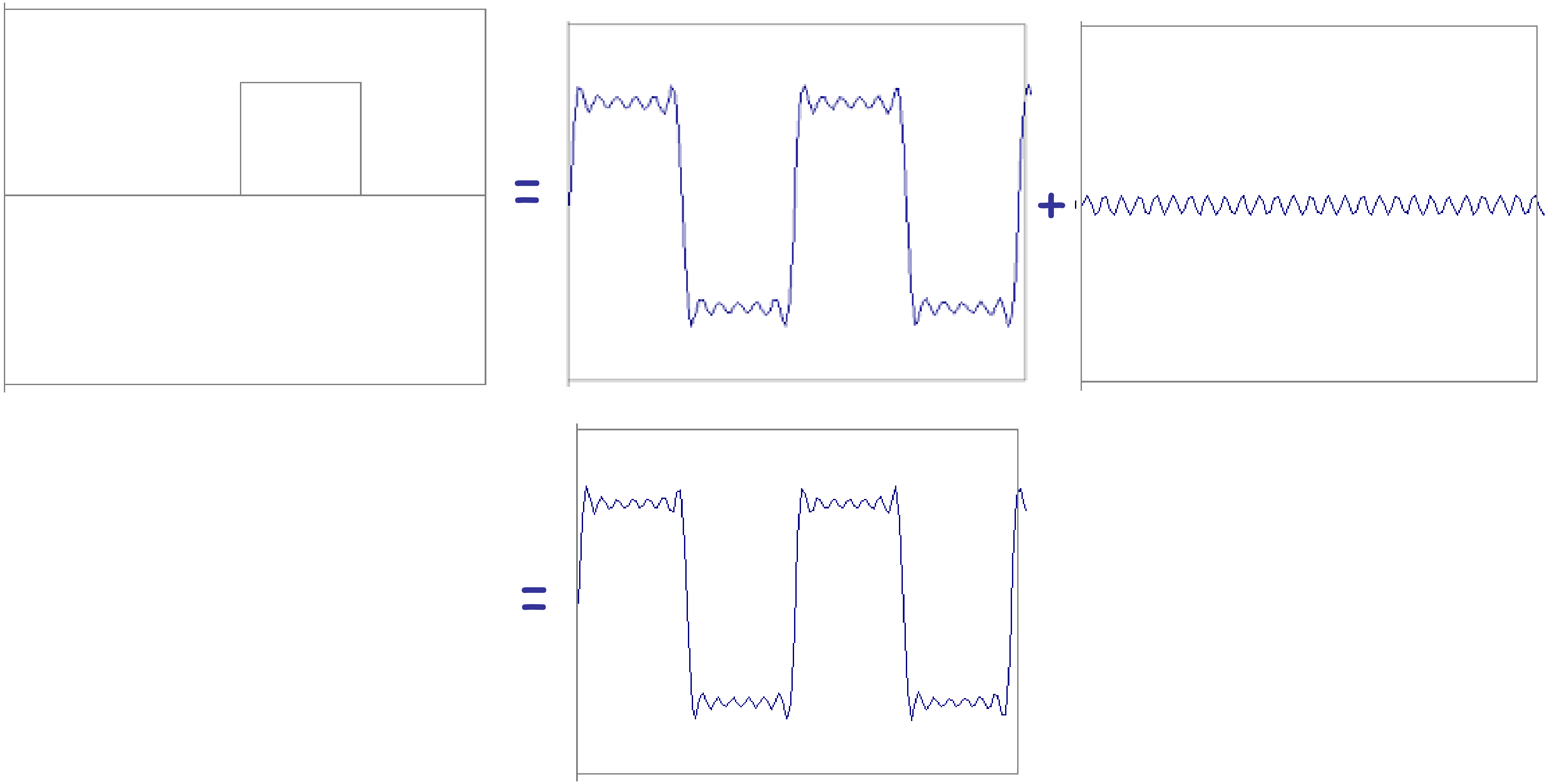
Frequency Spectra



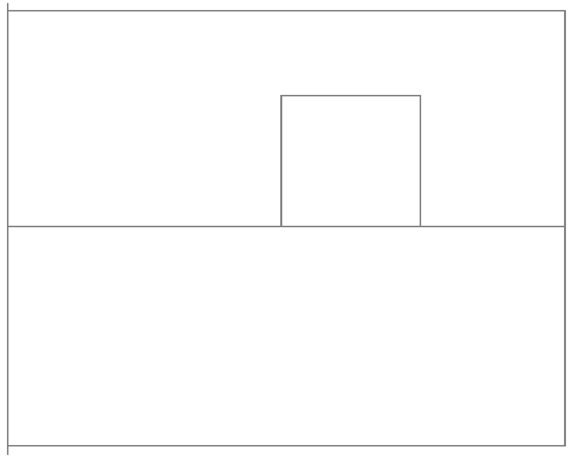
Frequency Spectra



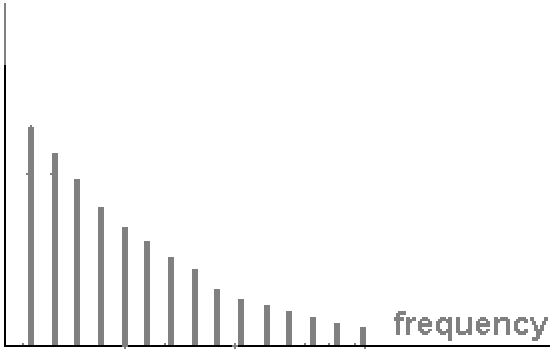
Frequency Spectra



Frequency Spectra



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



Fourier Transform – more formally

Represent the signal as an infinite weighted sum of an infinite number of sinusoids

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

Note: $e^{ik} = \cos k + i \sin k \qquad i = \sqrt{-1}$

Spatial Domain (x) $\longrightarrow$ Frequency Domain (u)

(Frequency Spectrum $F(u)$)

Inverse Fourier Transform (IFT) $$f(x) = \int_{-\infty}^{\infty} F(u)\, e^{i2\pi ux}\, dx$$

Properties of Fourier Transform

	Spatial Domain	Frequency Domain
Linearity	$c_1 f(x) + c_2 g(x)$	$c_1 F(u) + c_2 G(u)$
Scaling	$f(ax)$	$\frac{1}{\lvert a \rvert} F\left(\frac{u}{a}\right)$
Shifting	$f(x - x_0)$	$e^{-i2\pi u x_0} F(u)$
Symmetry	$f(x)$	$F(-u)$
Conjugation	$f^*(x)$	$F^*(-u)$
Convolution	$f(x) * g(x)$	$F(u)G(u)$
Differentiation	$\frac{d^n f(x)}{dx^n}$	$(2\pi u)^n F(u)$

Discrete Fourier Transform

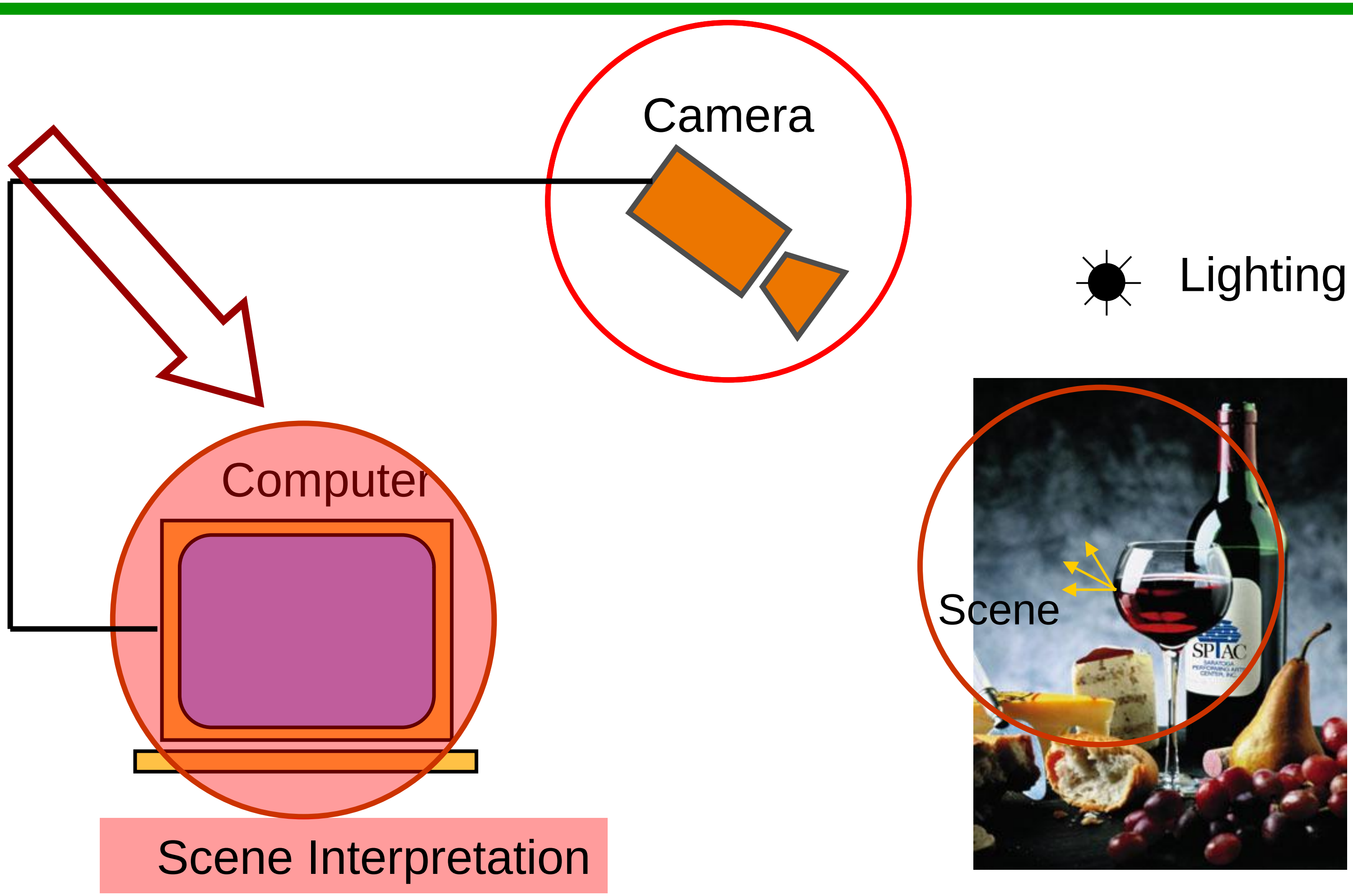
- If we have a sampled signal f(i), where i=0…N-1, the FT will become the DFT

$$F(u) = \sum_{i} f(i) e^{-i 2\pi \frac{u}{N} i}$$

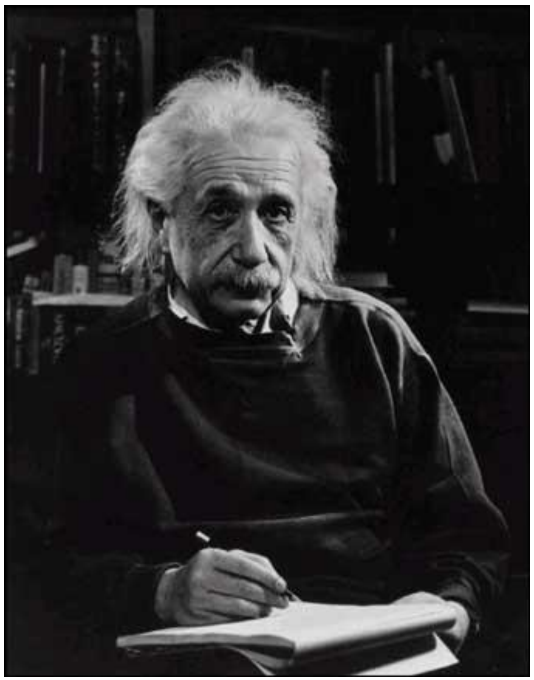
Topic: Sampling and quantization

- Image sensors
- Fourier, sampling and quantization
- Data structures for digital images
- Histograms

Components of a Computer Vision System



Digital Images



What we see

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

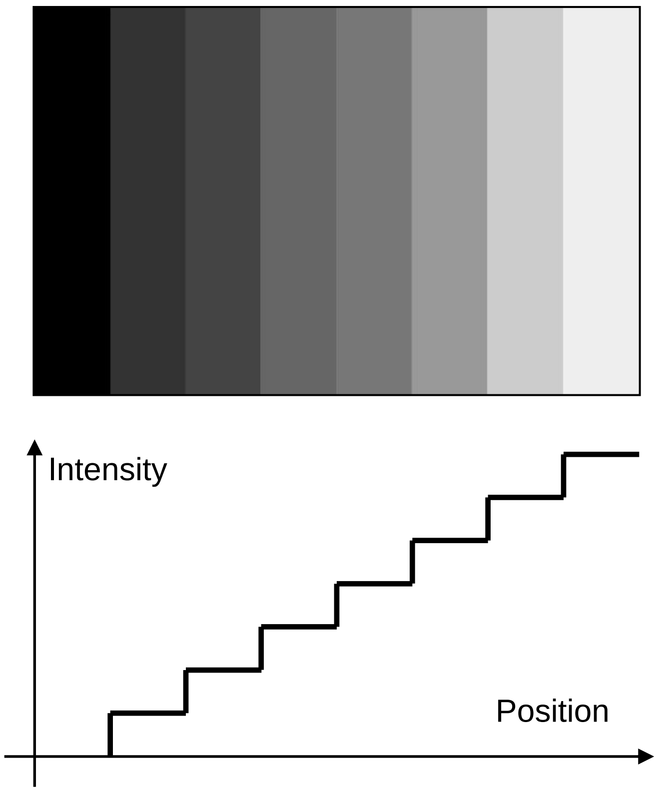
Simple Image Model

- Image as a 2D light-intensity function

$$f(x, y)$$

- Continuous
- Non-zero, finite value

$$0 < f(x, y) < \infty$$

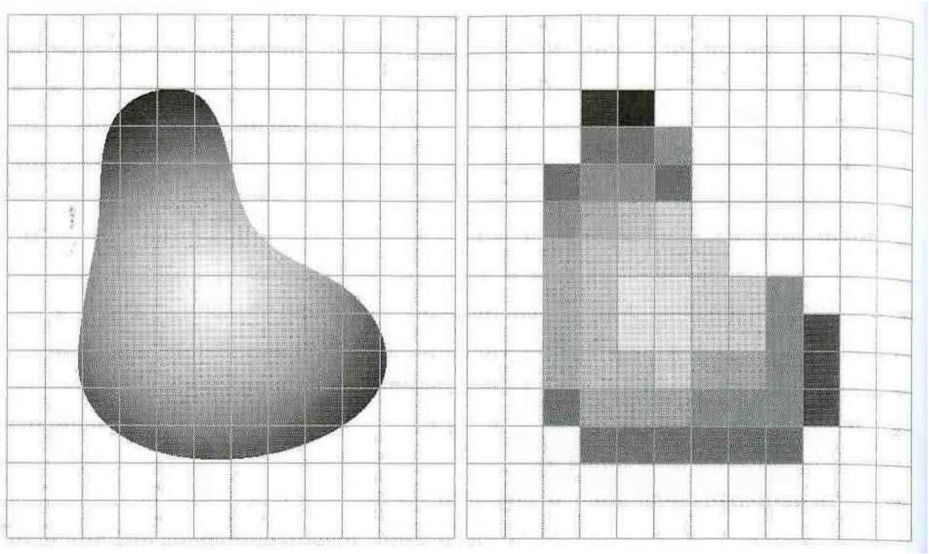


[Gonzalez & Woods]

Analog to Digital

The scene is:

- **projected** on a 2D plane,
- **sampled** on a regular grid, and each sample is
- **quantized** (rounded to the nearest integer)

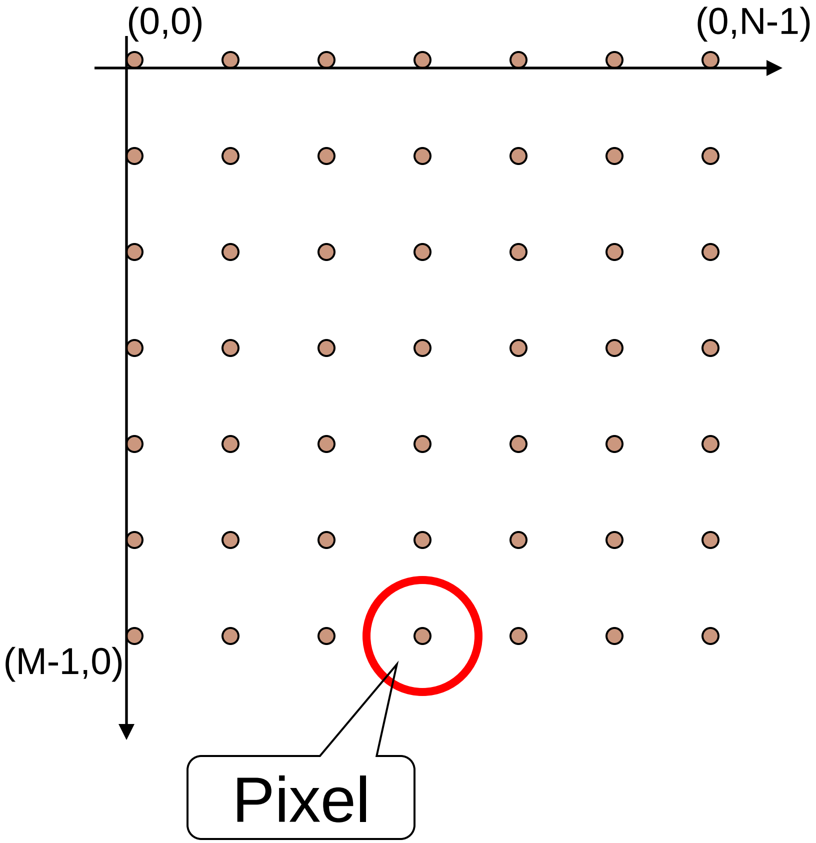


$$f(i, j) = \text{Quantize}\{f(i\Delta, j\Delta)\}$$

Images as Matrices

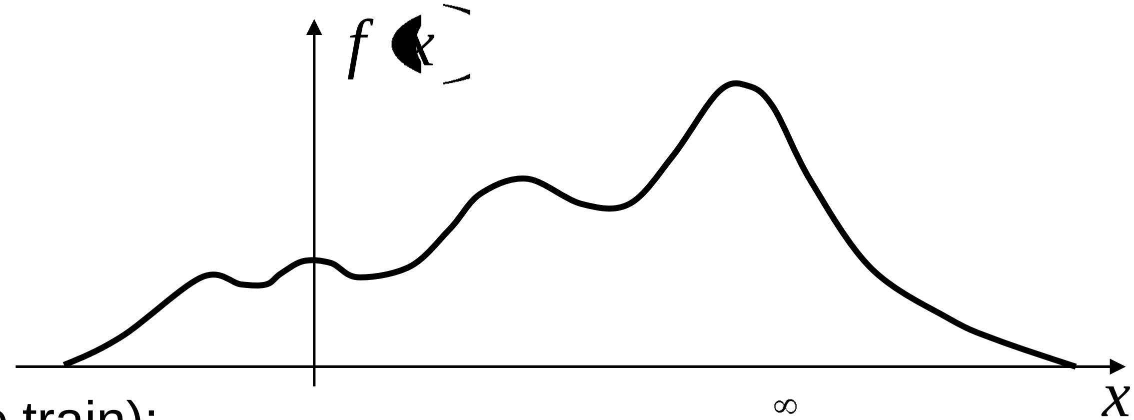
- Each point is a pixel with amplitude:
  - f(x,y)
- An image is a matrix with size N x M

M = [(0,0) (0,1) …
[(1,0) (1,1) …
…



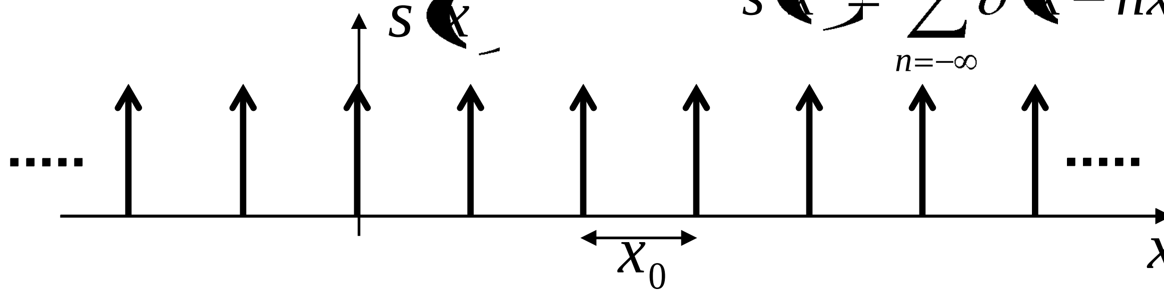
Sampling Theorem

Continuous signal:



Shah function (Impulse train):

$$s(x) = \sum_{n=-\infty}^{\infty} \delta(x - nx_0)$$



Sampled function:

$$f_s(x) = f(x)s(x) = f(x)\sum_{n=-\infty}^{\infty} \delta(x - nx_0)$$

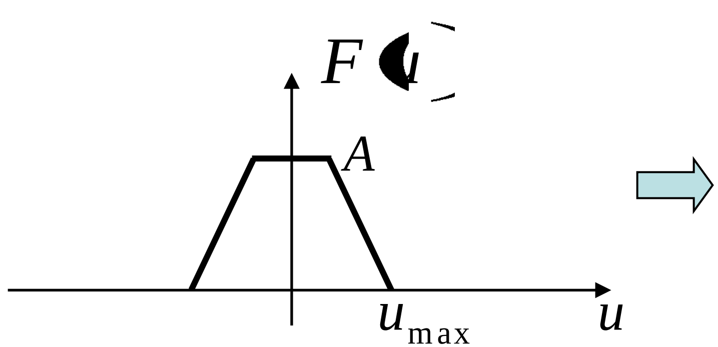
Sampling Theorem

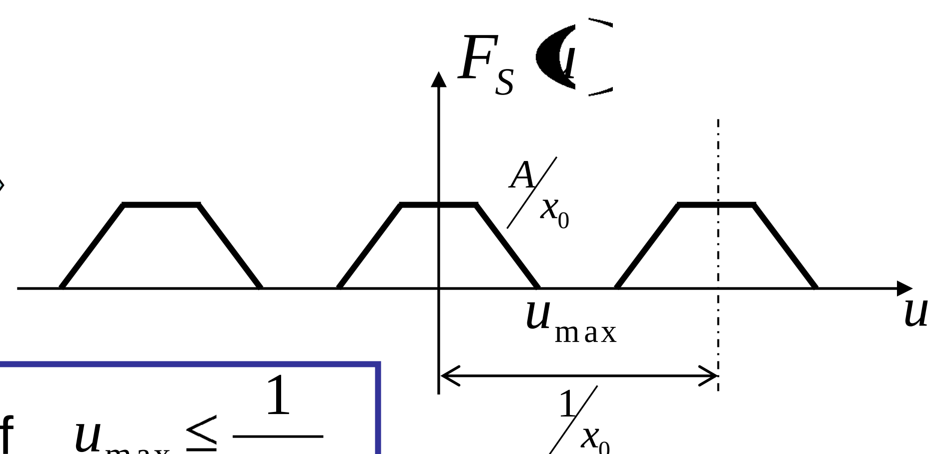
Sampled function: $f_s(x) = f(x)s(x) = f(x)\sum_{n=-\infty}^{\infty}\delta(x - nx_0)$

Sampling frequency $\frac{1}{x_0}$

Freq Domain

$$F_S(u) = F(u) * S(u) = F(u) * \frac{1}{x_0}\sum_{n=-\infty}^{\infty}\delta\left(u - \frac{n}{x_0}\right)$$

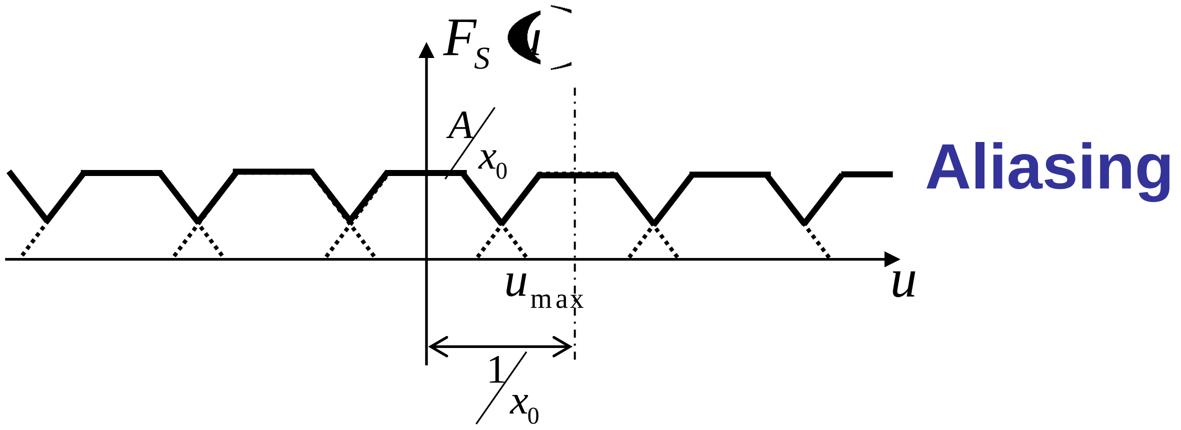




Only if $u_{max} \leq \frac{1}{2x_0}$

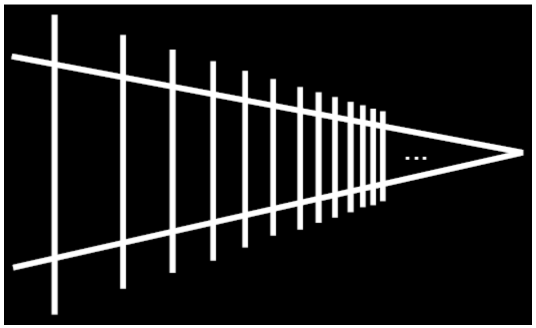
Nyquist Theorem

If $u_{max} > \dfrac{1}{2x_0}$



$\boxed{\text{Sampling frequency must be greater than } 2u_{max}}$

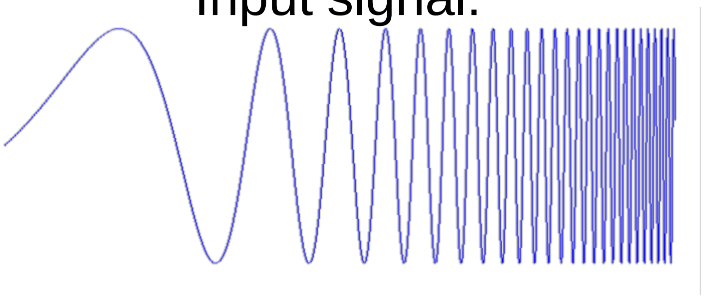
Aliasing



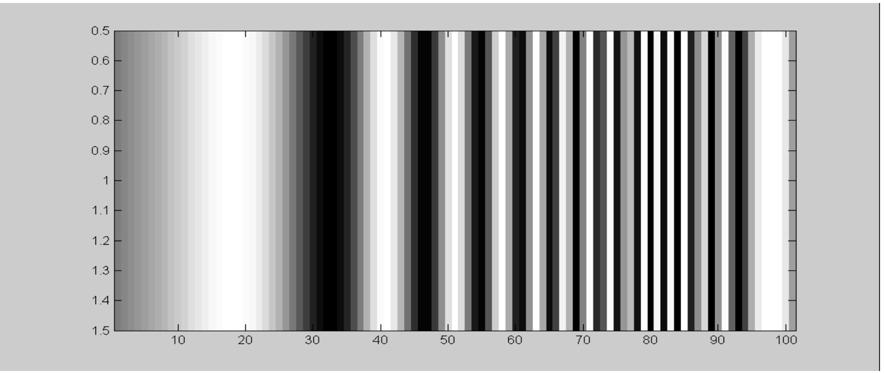
Picket fence receding into the distance will produce aliasing…

WHY?

Input signal:



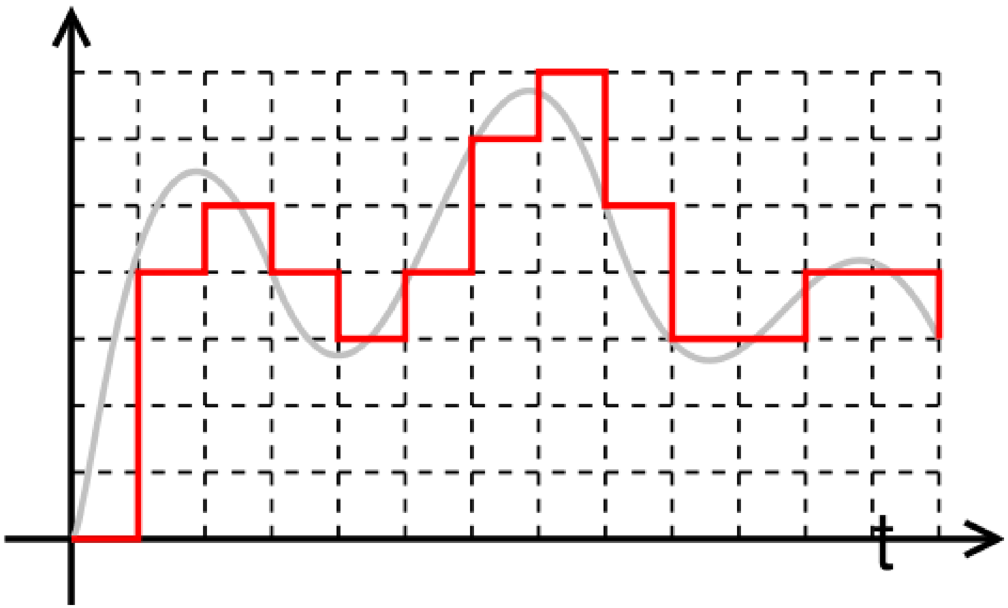
Matlab output:



x = 0:.05:5; imagesc(sin((2.^x).*x))

Quantization

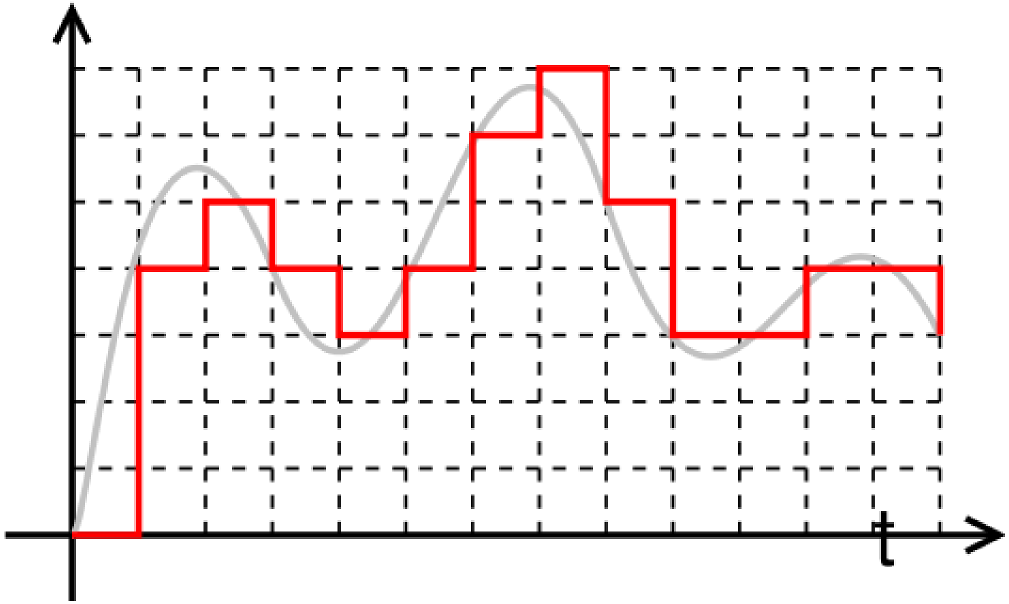
- Analog: $0 < f(x, y) < \infty$
- Digital: Infinite storage space per pixel!
- Quantization



Quantization Levels

- G - number of levels
- m – storage bits
- Round each value to its nearest level

$$G = 2^m$$



Effect of quantization

Effect of quantization

Image Size

- Storage space
  - Spatial resolution: N x M
  - Quantization: m bits per pixel
  - Required bits b:

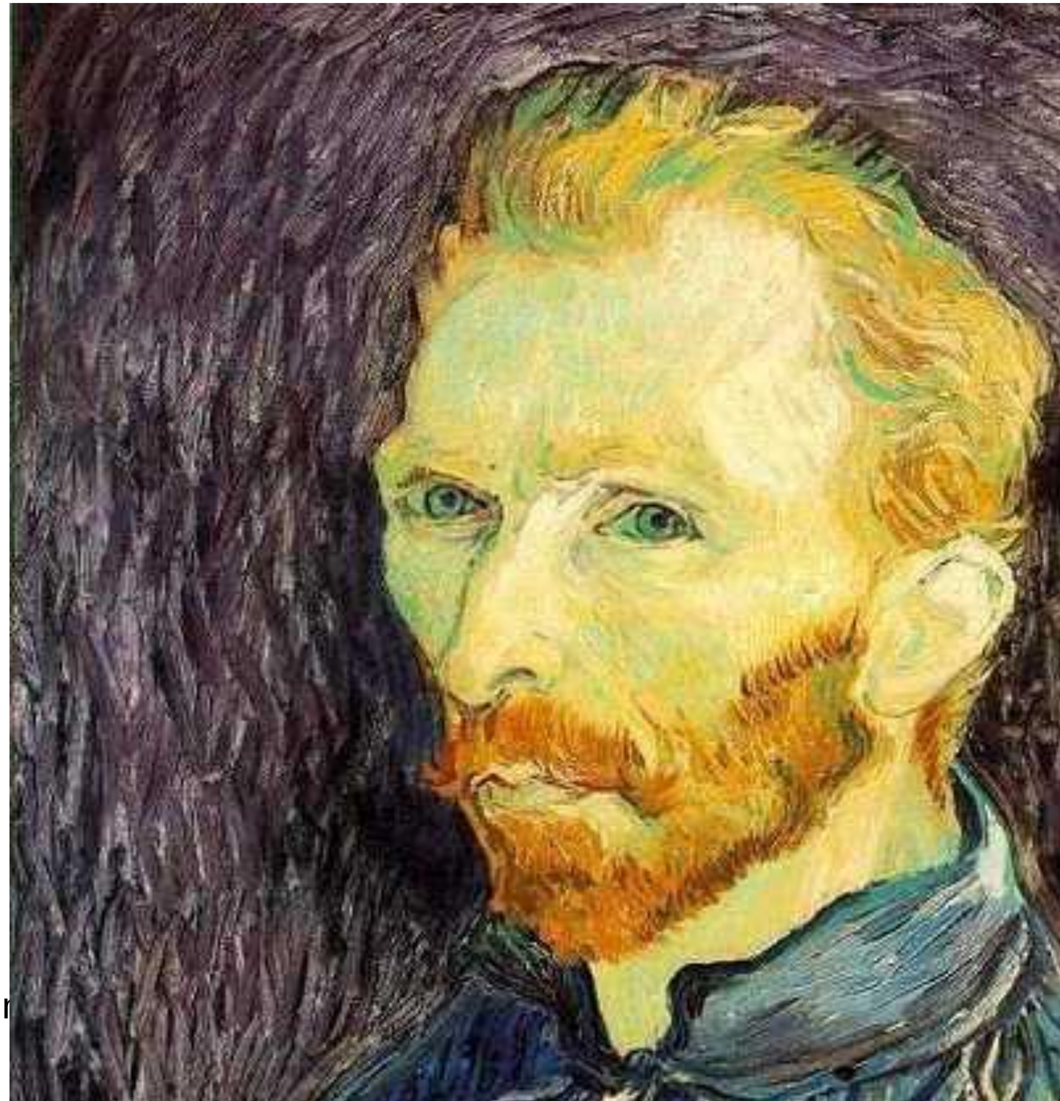
$$b = N \times M \times m$$

- Rule of thumb:
  - More storage space means more image quality

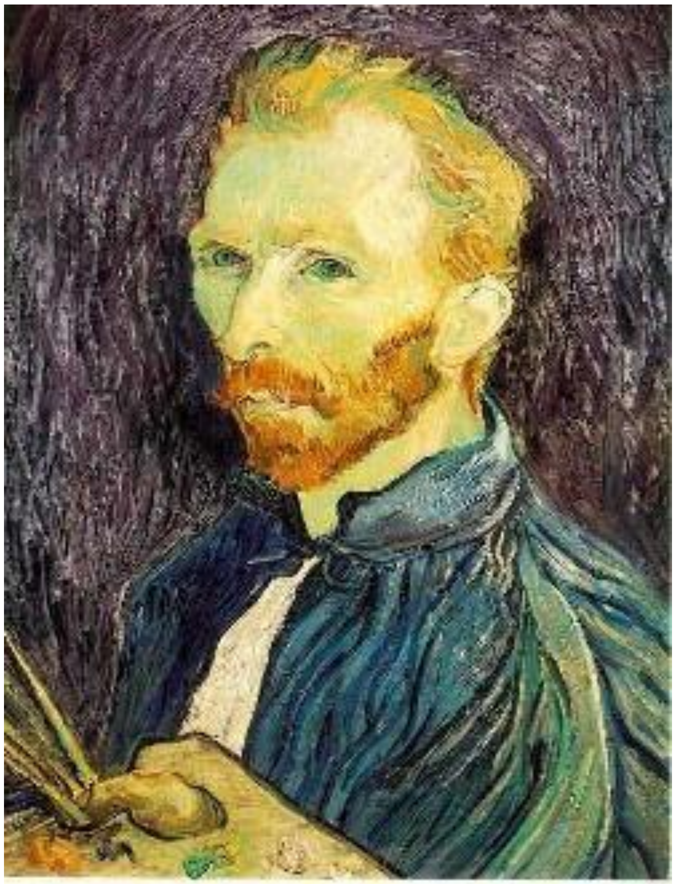
Image Scaling

This image is too big to fit on the screen. How can we reduce it?

How to generate a half-sized version?



Sub-sampling

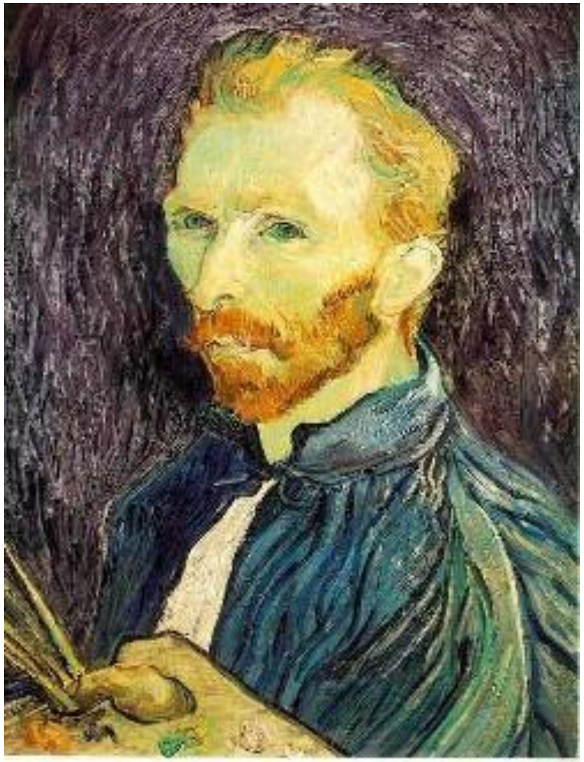


1/4

1/8

Throw away every other row and column to create a 1/2 size image
- called *image sub-sampling*

Sub-sampling

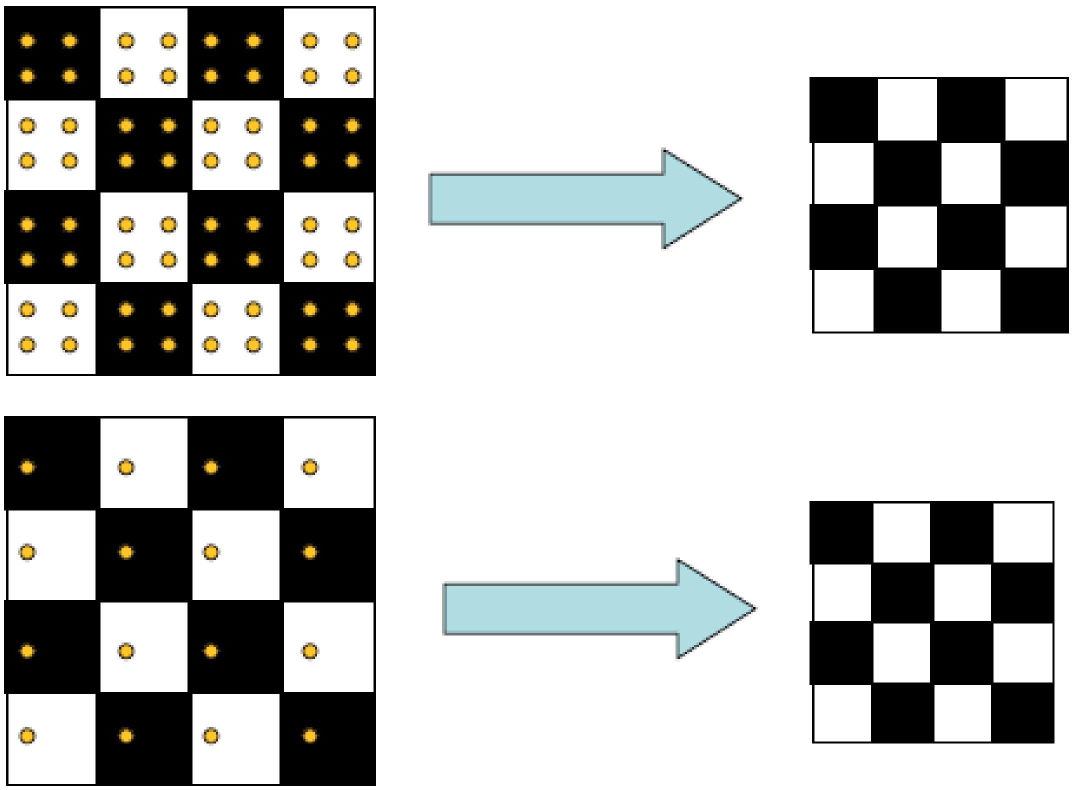


1/2

1/4 (2x zoom)

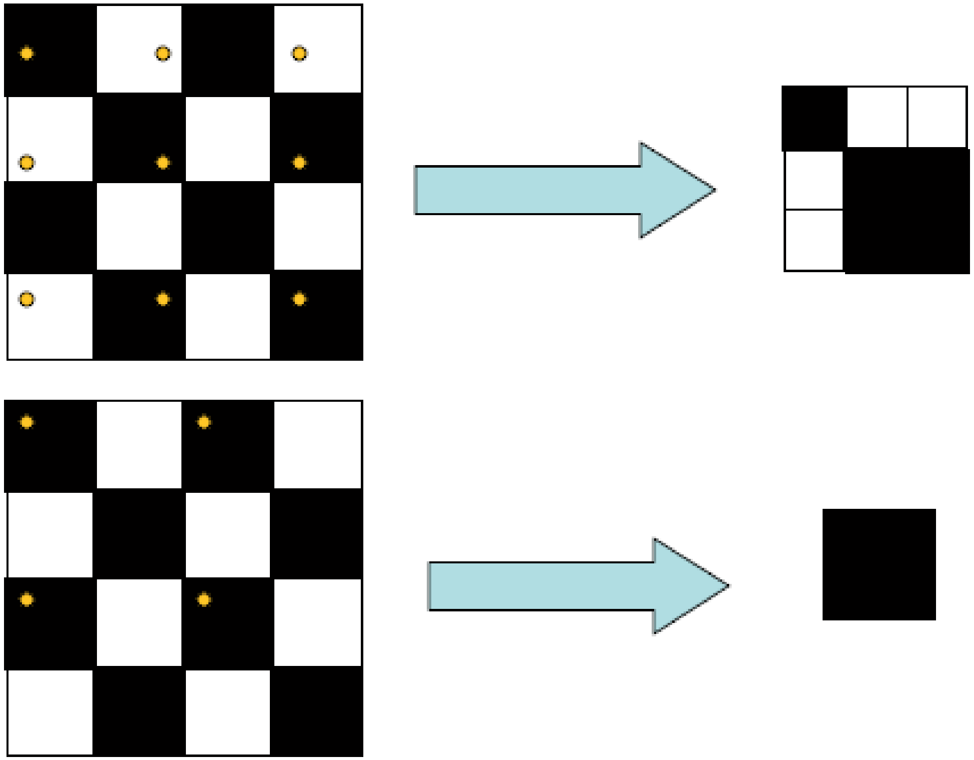
1/8 (4x zoom)

Sampling an Image



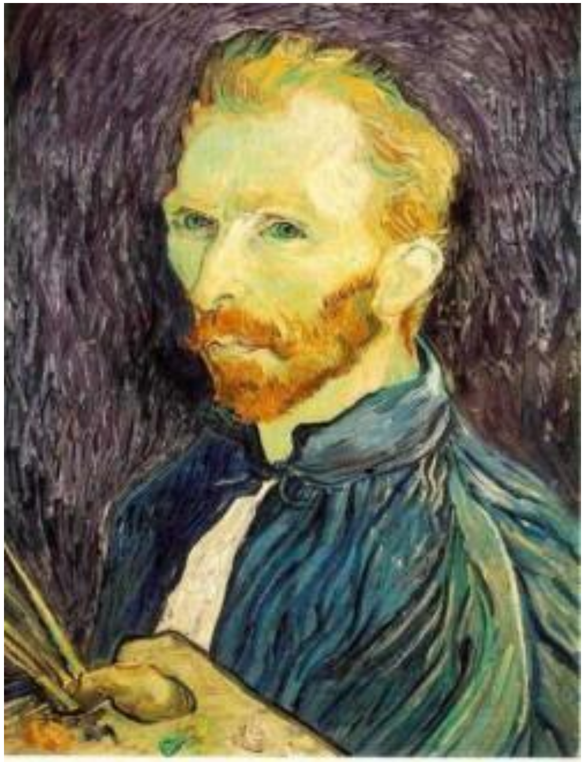
GOOD sampling

Sampling an Image



BAD sampling -> Aliasing

Sub-Sampling with Gaussian Pre-Filtering

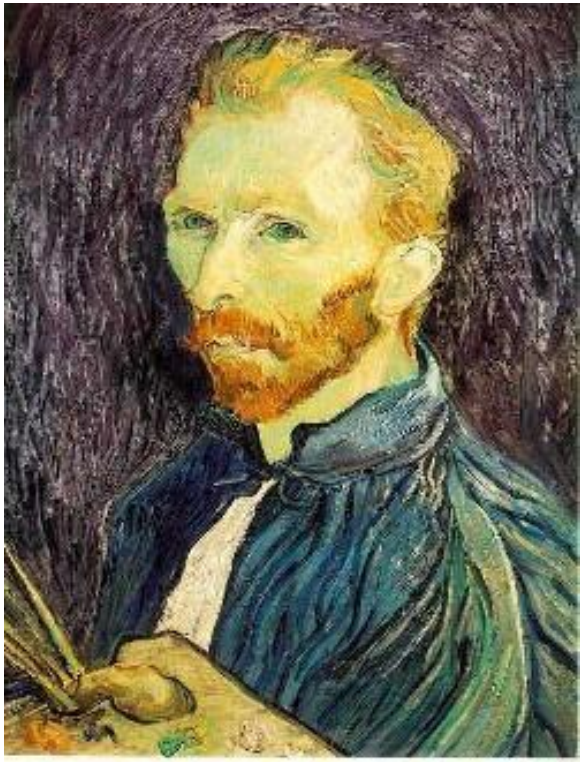


Gaussian 1/2

G 1/4

G 1/8

Compare with...



1/2

1/4 (2x zoom)

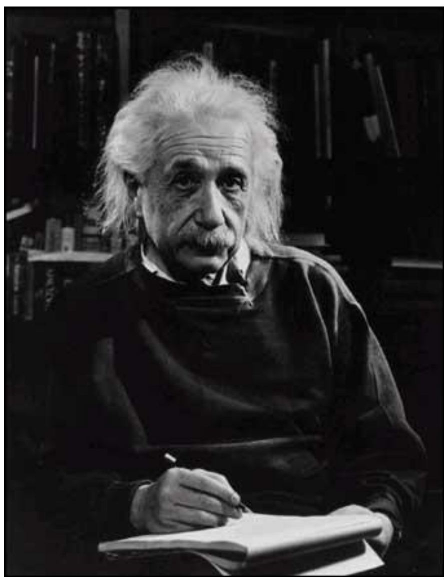
1/8 (4x zoom)

Topic: Data structures for digital images

- Image sensors
- Sampling and quantization
- Data structures for digital images
- Histograms

Data Structures for Digital Images

- Are there other ways to represent digital images?



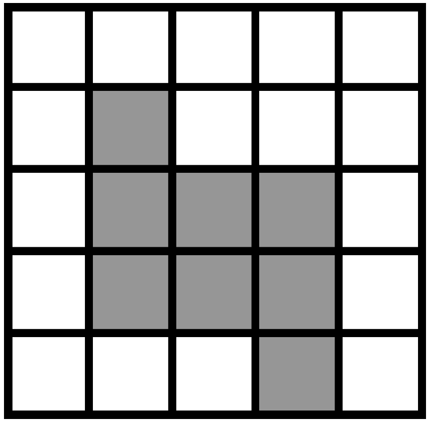
What we see

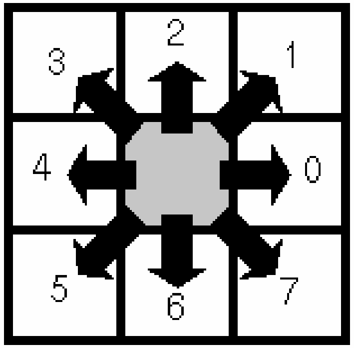
0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

Chain codes

- Chains represent the borders of objects.
- Coding with *chain codes*.
  - Relative.
  - Assume an initial starting point for each object.
- Needs segmentation!



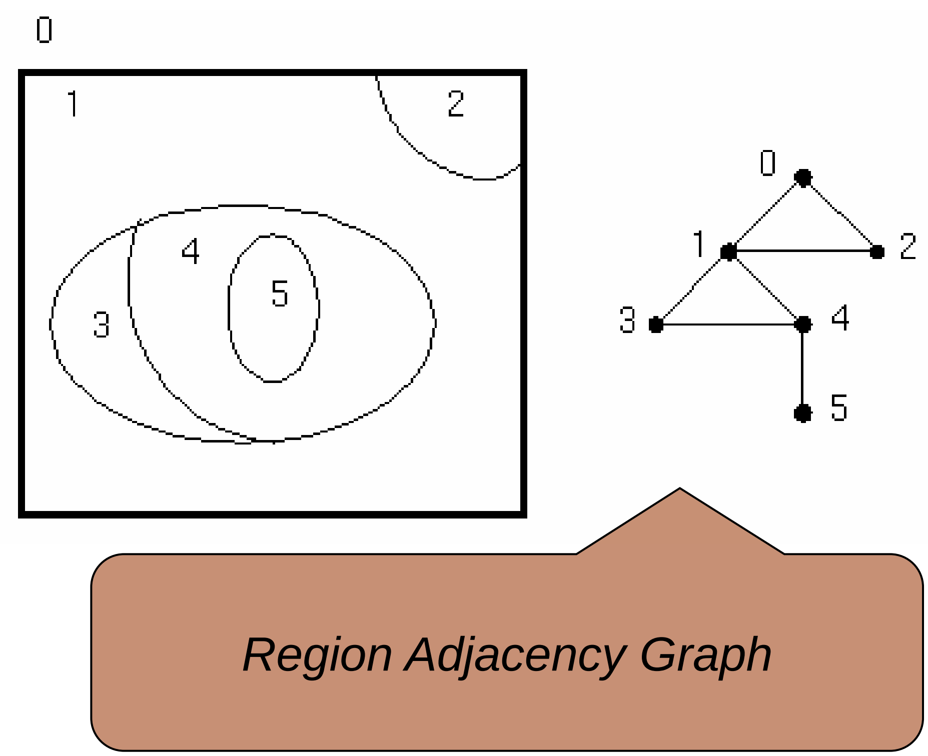


Freeman Chain Code

Using a Freeman Chain Code and considering the top-left pixel as the starting point:
70663422

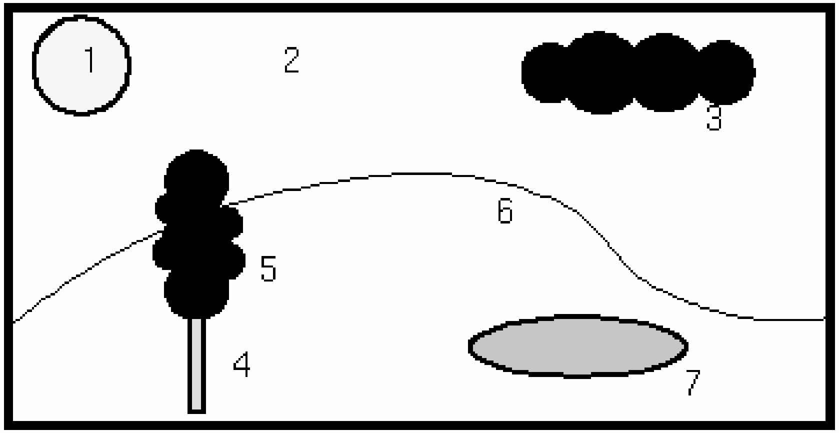
Topological Data Structures

- *Region Adjacency Graph*
  - **Nodes** - Regions
  - **Arcs** – Relationships
- Describes the elements of an image and their spatial relationships.
- Needs segmentation!



Relational Structures

- Stores relations between objects.
- Important semantic information of an image.
- Needs segmentation and an image description (features)!



No.	Object name	Colour	Min. row	Min. col.	Inside
1	sun	white	5	40	2
2	sky	blue	0	0	-
3	cloud	grey	20	180	2
4	tree trunk	brown	95	75	6
5	tree crown	green	53	63	-
6	hill	light green	97	0	-
7	pond	blue	100	160	6

Relational Table

Topic: Histograms

- Image sensors
- Sampling and quantization
- Data structures for digital images
- Histograms

Histograms

- In statistics, a histogram is a graphical display of tabulated frequencies.

Typically represented as a bar chart:

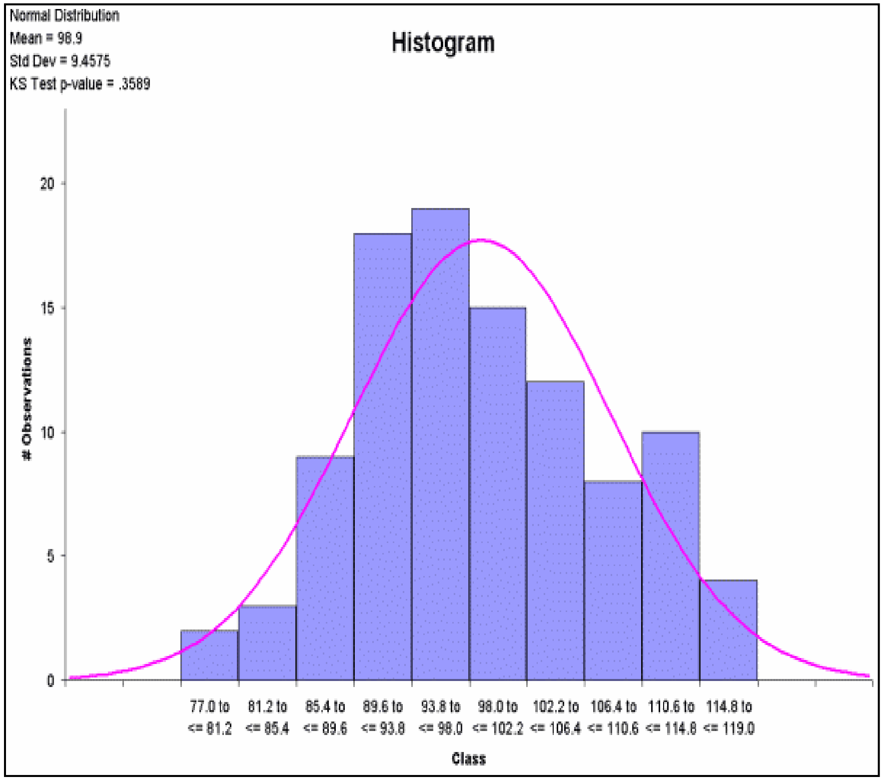
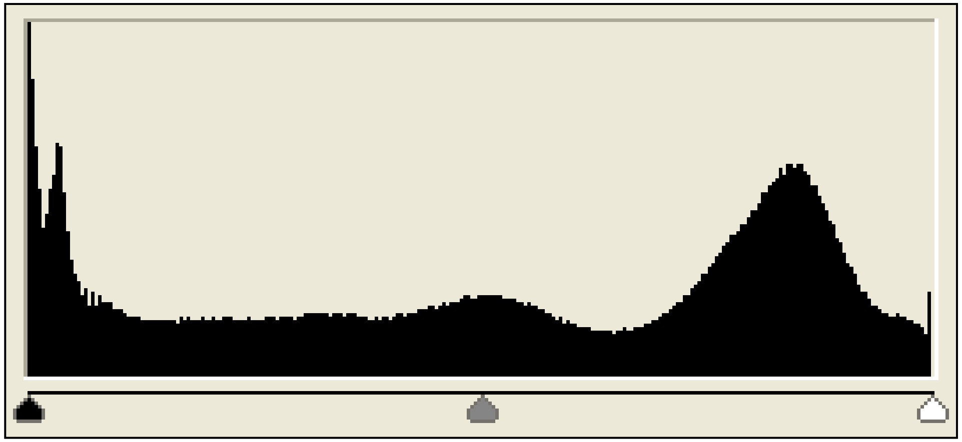


Image Histograms

- Colour or Intensity distribution.
- Typically:
  - Reduced number of *bins*.
  - Normalization.
- Compressed representation of an image.
  - No spatial information whatsoever!



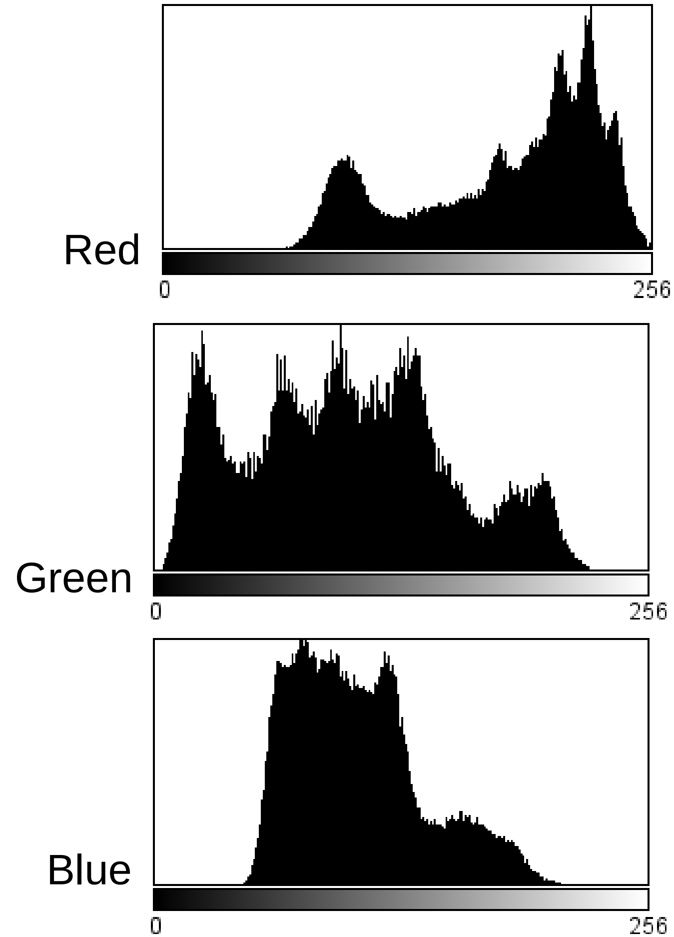
Colour Histogram

- As many histograms as axis of the colour space.

  Ex: RGB Colour space

  - Red Histogram
  - Green Histogram
  - Blue Histogram

- Combined histogram.



Resources

- J.C. Russ – Chapters 2
- R. Gonzalez, and R. Woods – Chapter 2