Visual Computing:
The Digital Image

Prof. Marc Pollefeys

ETH

inf Informatik Computer Science

Digital cameras are the best sensors <u>ever</u>!

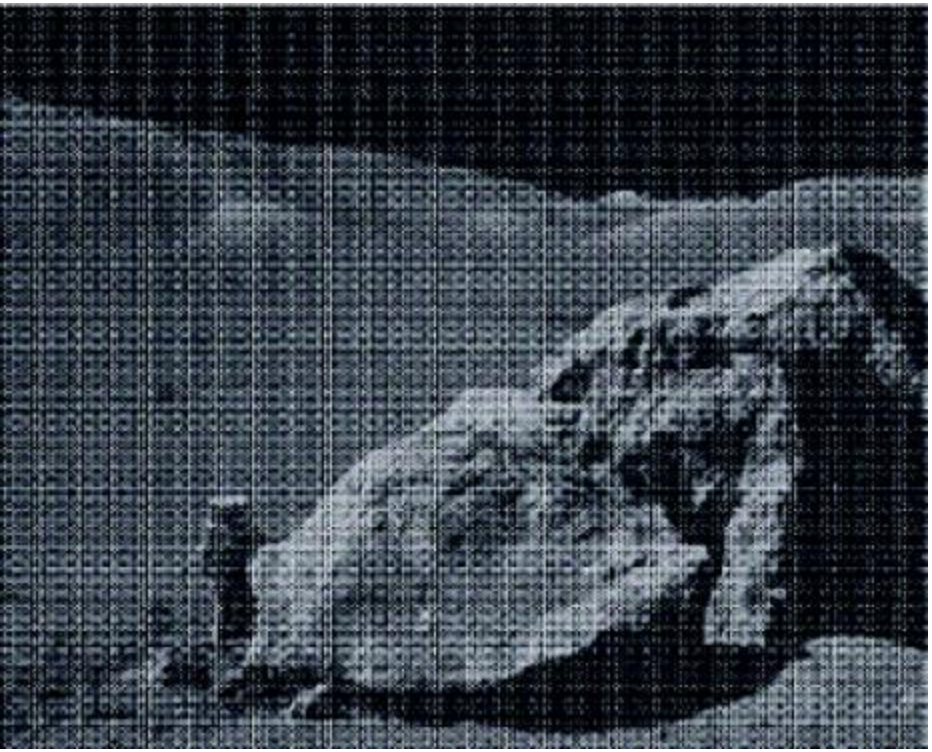
[(Example video)]

With a few problems...

ETH

inf | Informatik
Computer Science

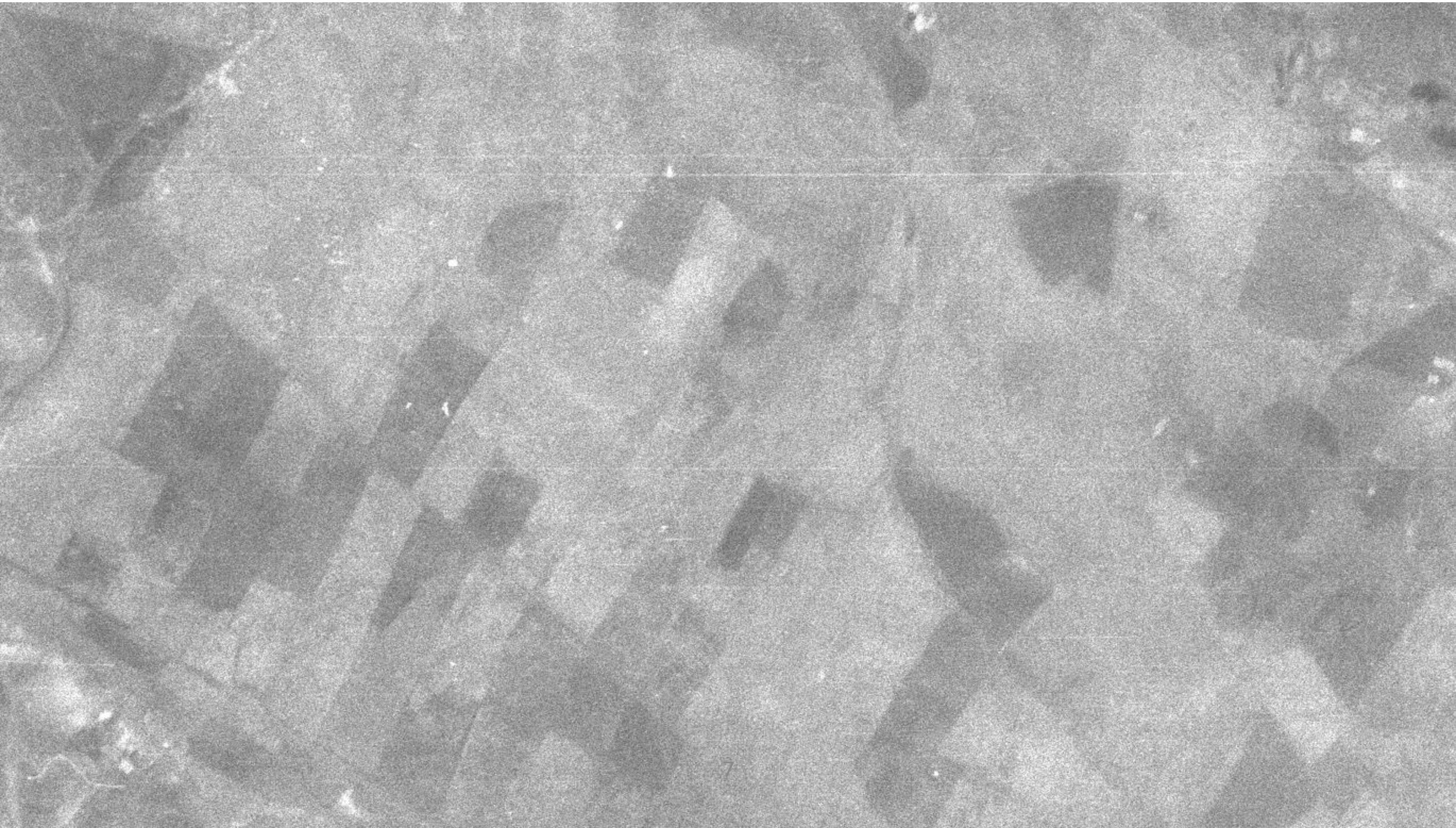
Transmission interference



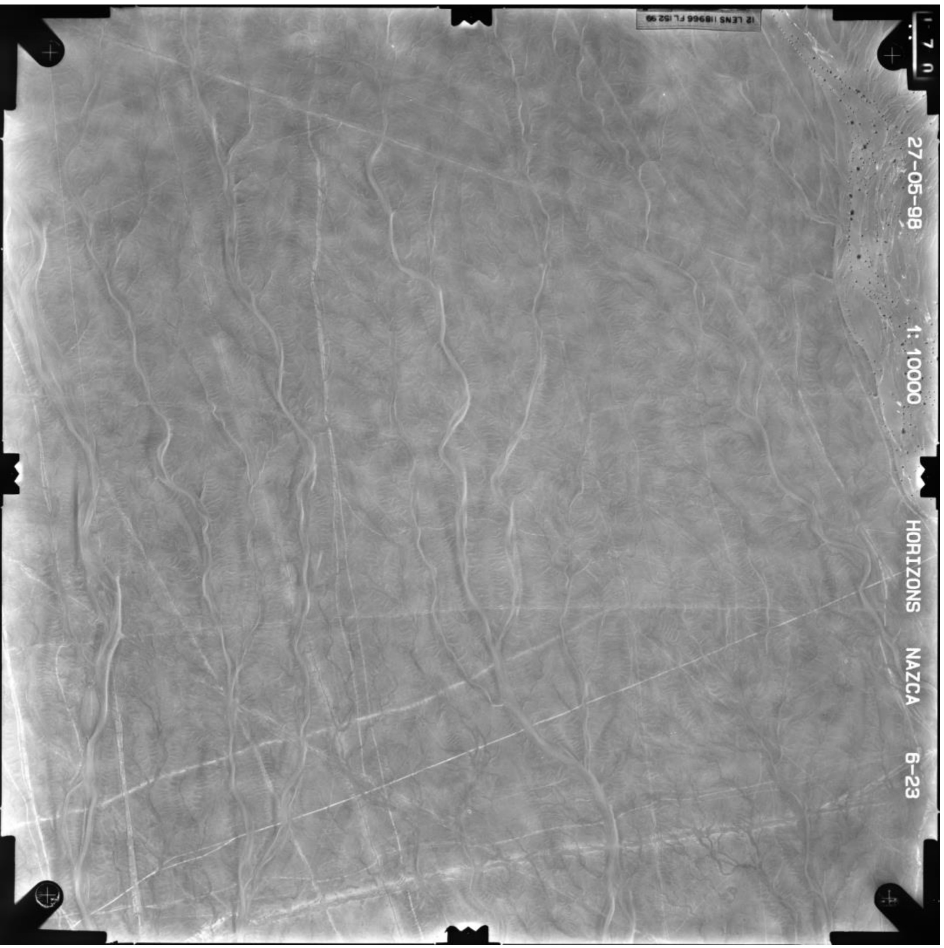
Compression artefacts

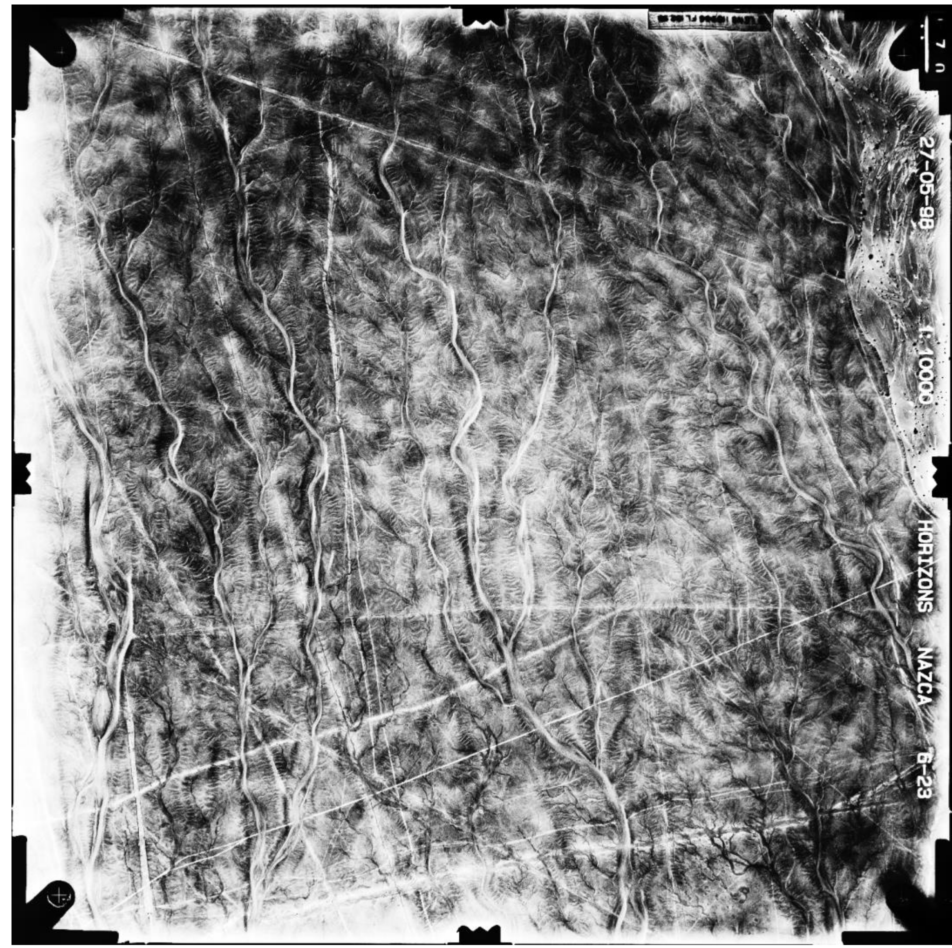
Spilling

Scratches, Sensor noise



Bad contrast

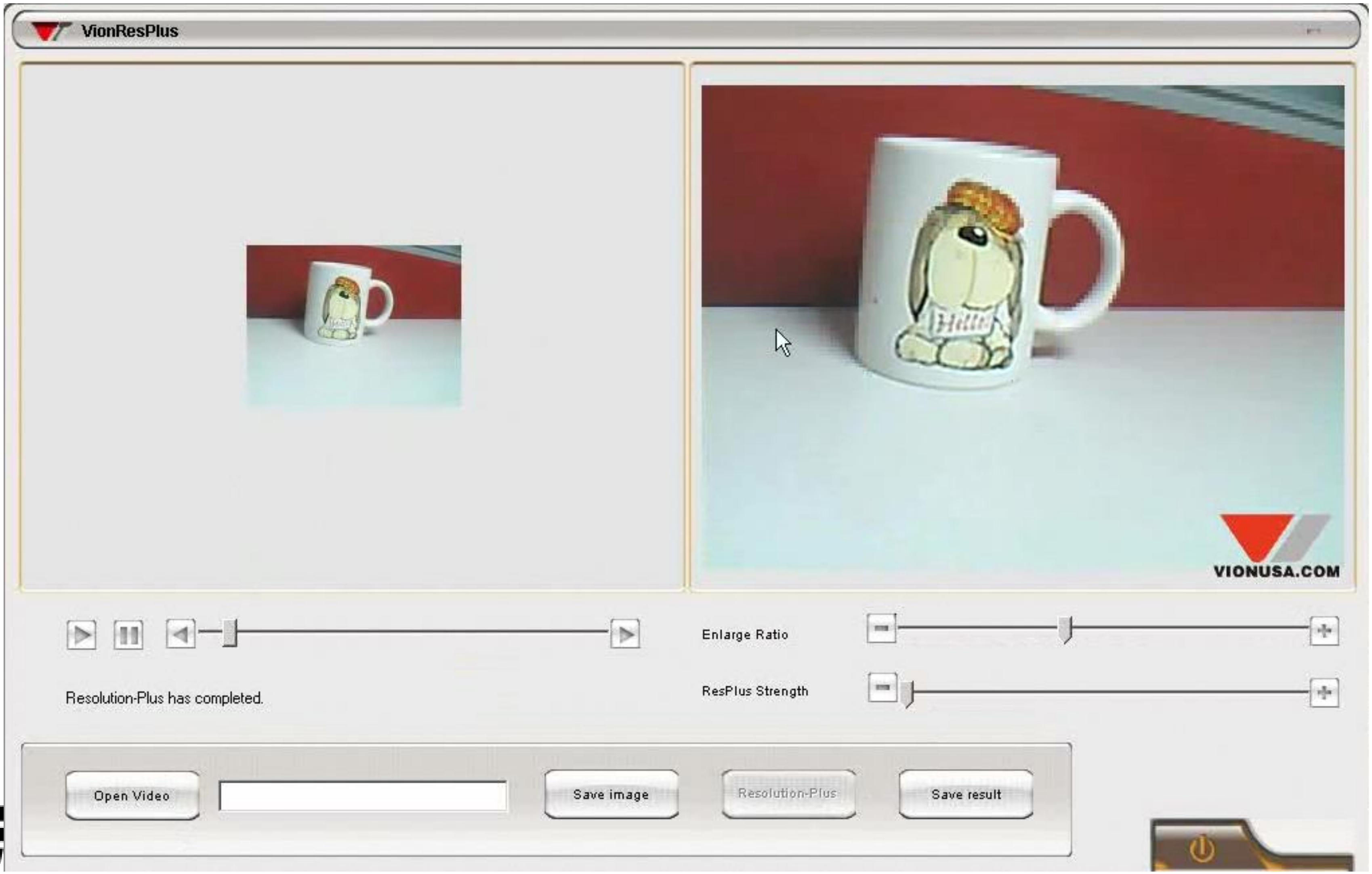




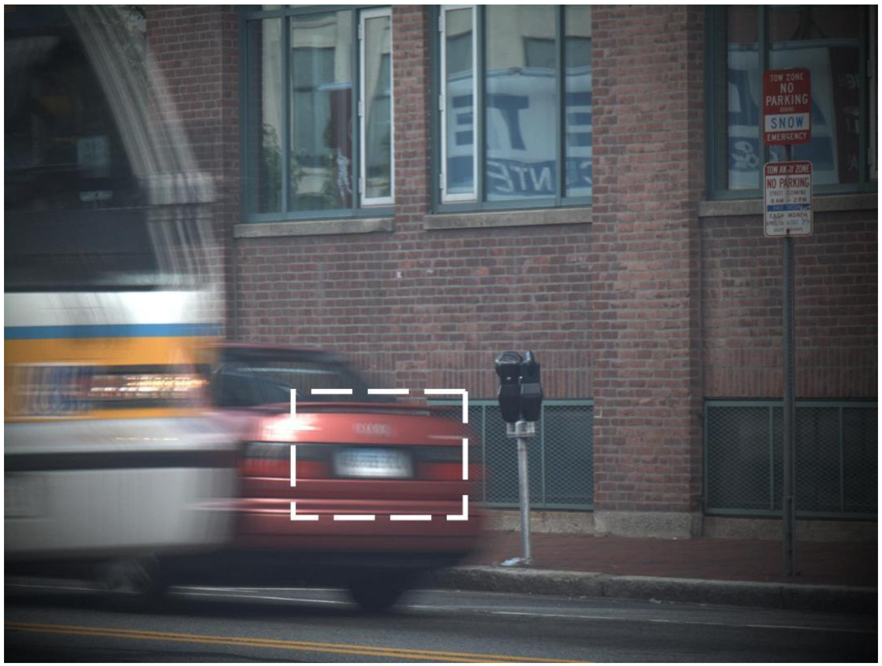
Resolution → Super resolution?

Super resolution



Removing motion blur



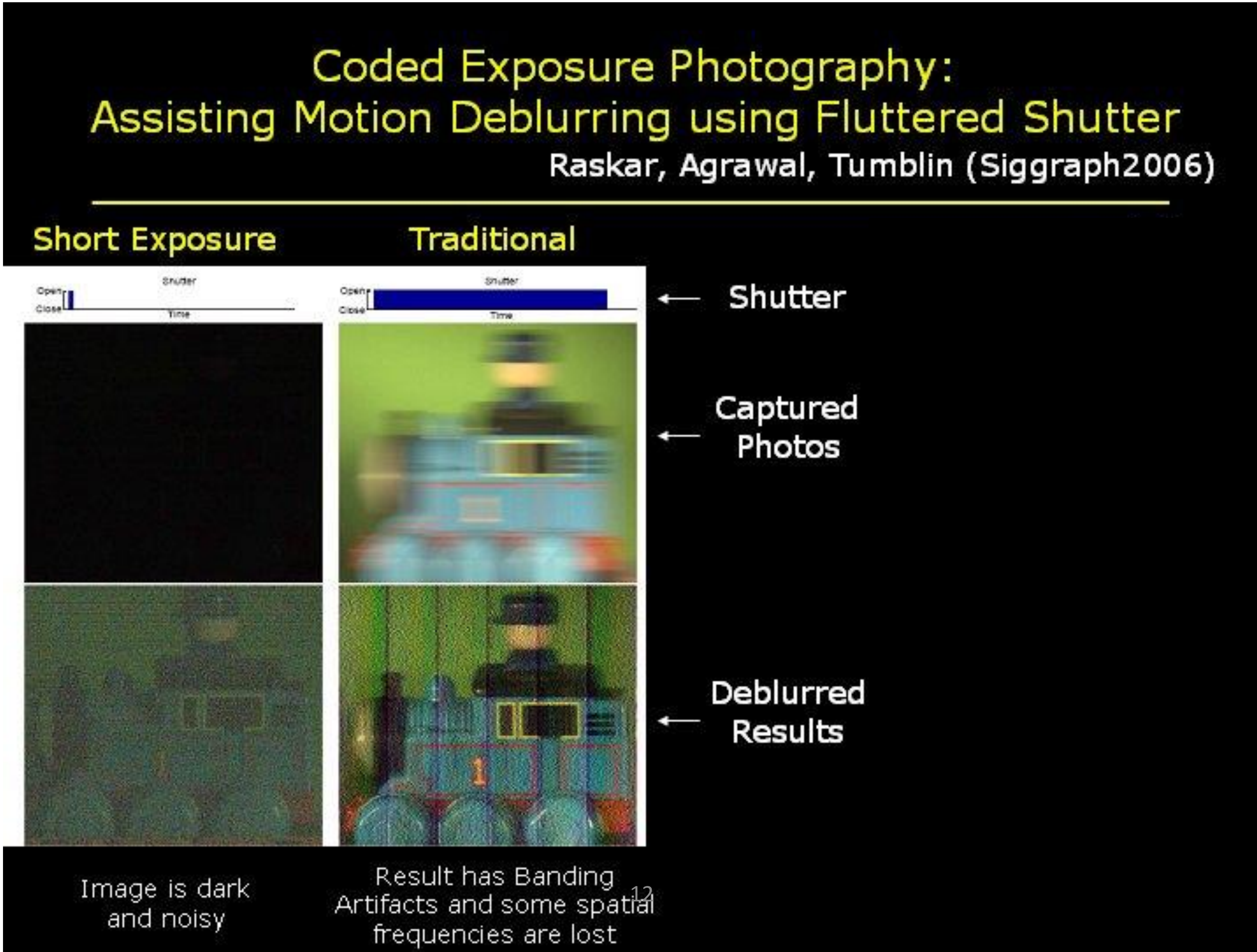
Original image

Cropped subwindow

After motion blur removal

[Images from Amit Agrawal]

Removing motion blur

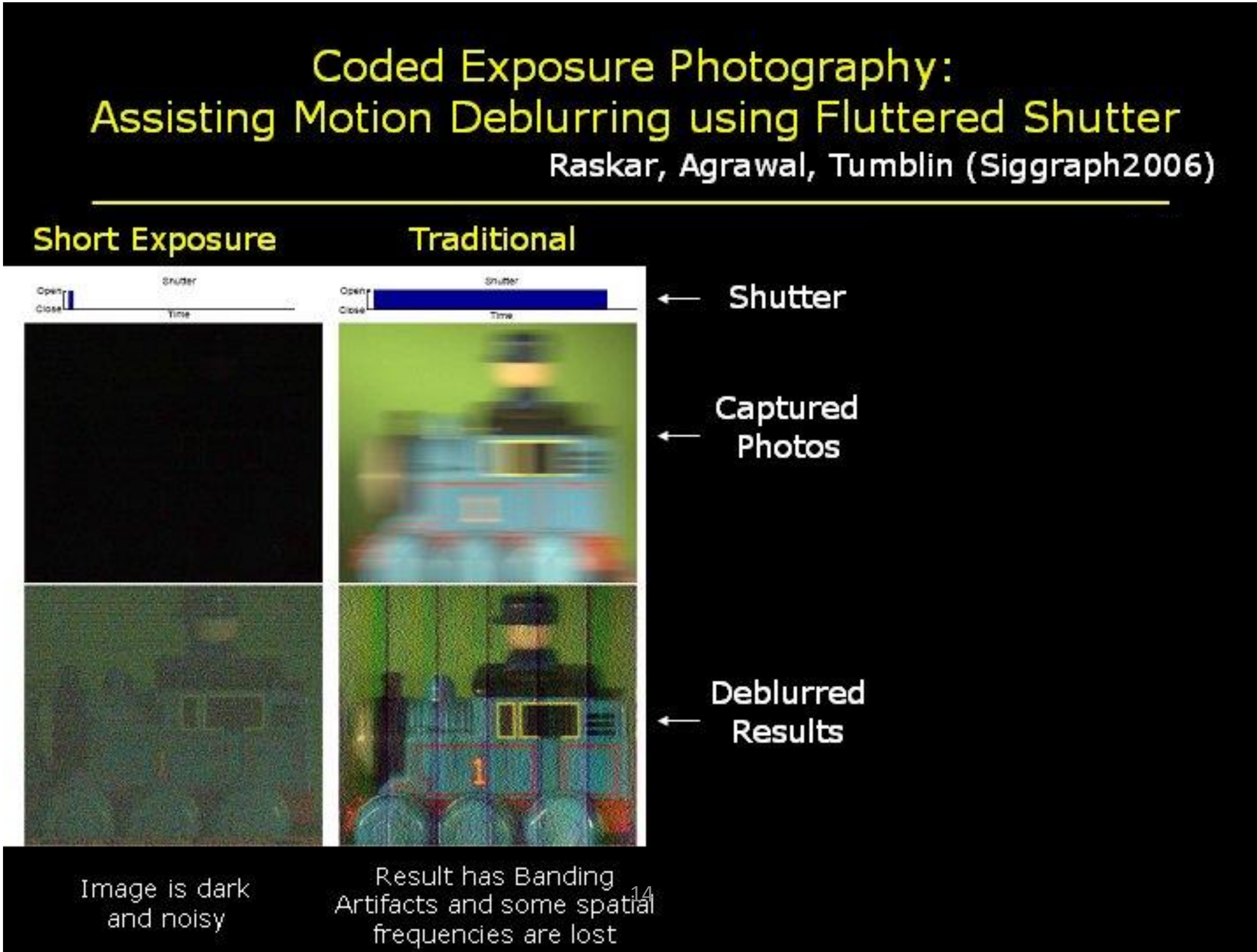


Fluttered Shutter Camera

Raskar, Agrawal, Tumblin Siggraph2006

Ferroelectric shutter in front of the lens is turned opaque or transparent in a rapid binary sequence

Removing motion blur



Python is Your Friend

- Run python:

`$ python` in a terminal or use an online Python notebook (e.g. Microsoft Azure notebook)

- Download any simple image

- Load it into Python:

```
>> import cv2
>> img = cv2.imread('foo.jpg')
```

Unassessed Assignment

- Display the image in Python:

```
>> cv2.imshow('My image', img)
>> cv2.waitKey(0)
```

- Print the image data array:

```
>> img
```

- Print the size of the image array and create a subimage:

```
>> img.shape
>> subimg = img[72:92, 62:82]
```

What is an image?

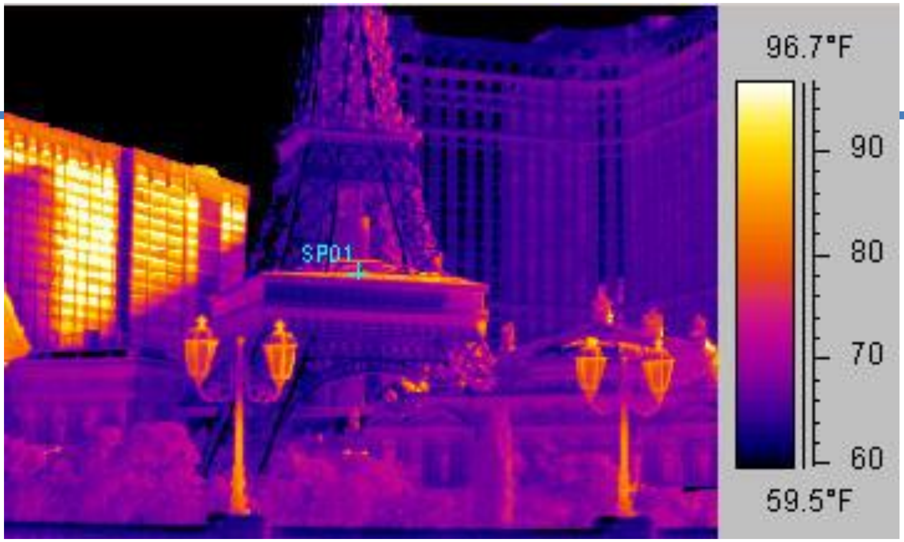
Image as 2D signal

- Signal: function depending on some variable with physical meaning
- Image: continuous function
  - 2 variables: xy - coordinates
  - 3 variables: xy + time (video)
- Brightness is usually the value of the function

- But can be other physical values too: temperature, pressure, depth …

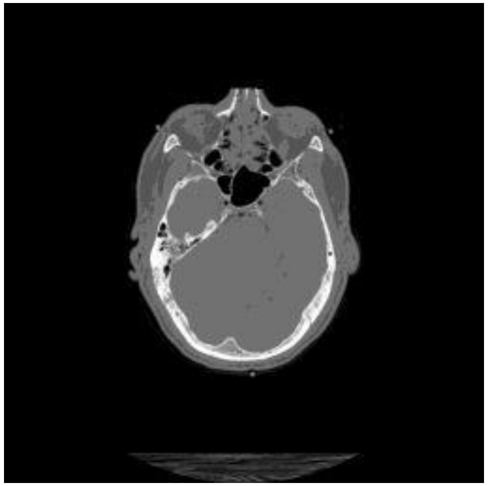
Example 2D Images

ultrasound



temperature (far IR)

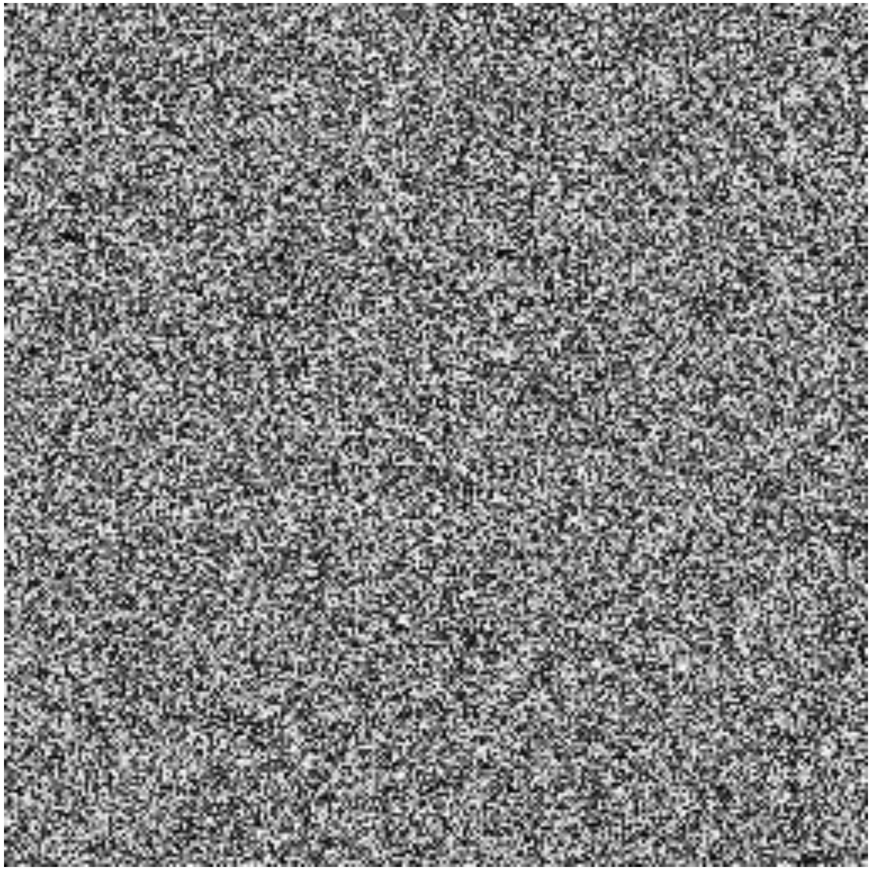
camera image



CT

Random Image

```
>> import numpy as np
>> import cv2
>> t = np.random.rand(64,64)
>> cv2.imshow('Random', t)
>> cv2.waitKey(0)
```



What is an image?

- A picture or pattern of a value varying in space and/or time.
- Representation of a function

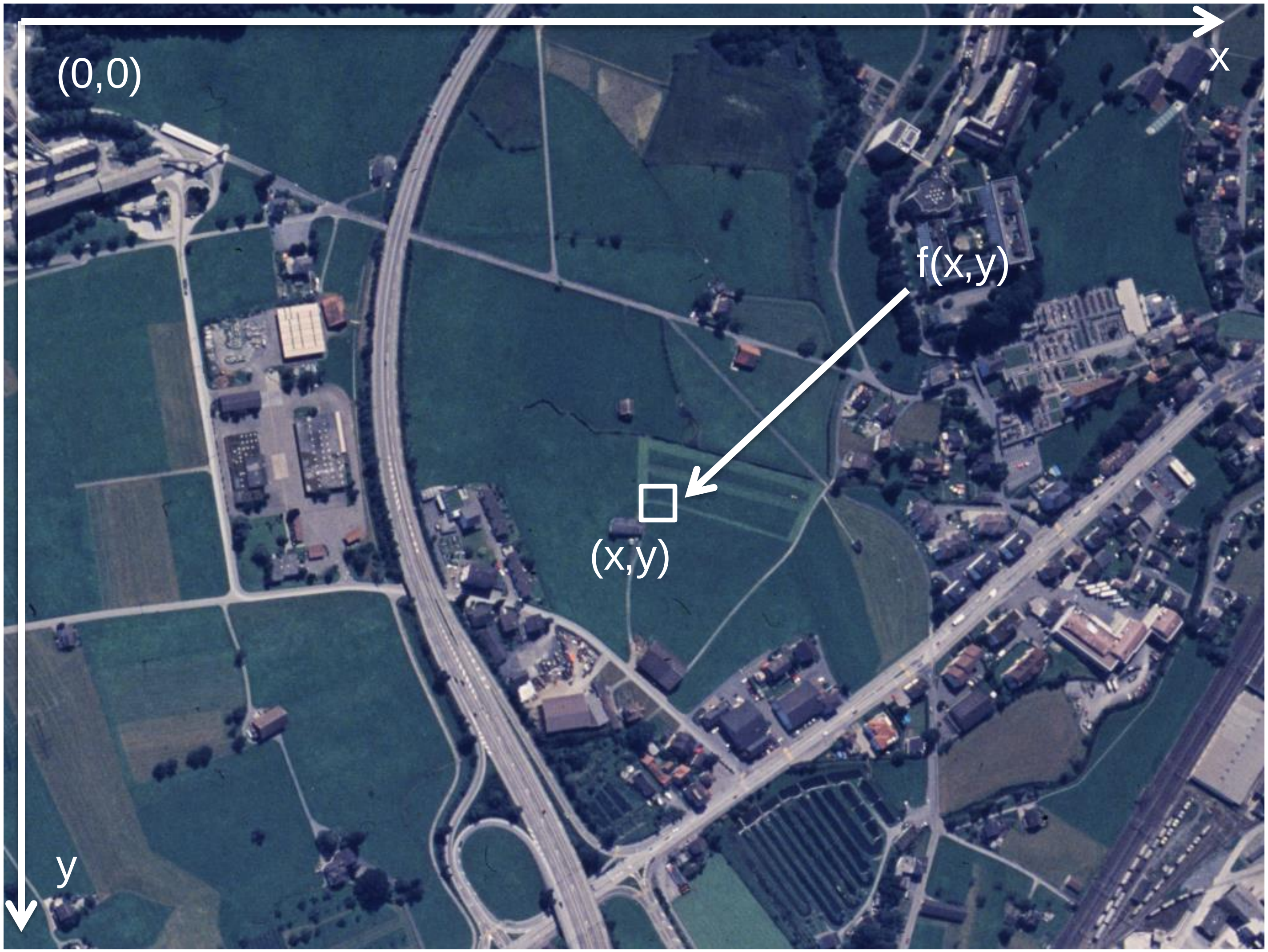
$$f : \Re^n \to S$$

- In digital form, eg:

I:{1, …, X}×{1, …, Y}→S.

- For greyscale images, n = 2, $S = \Re^+$.

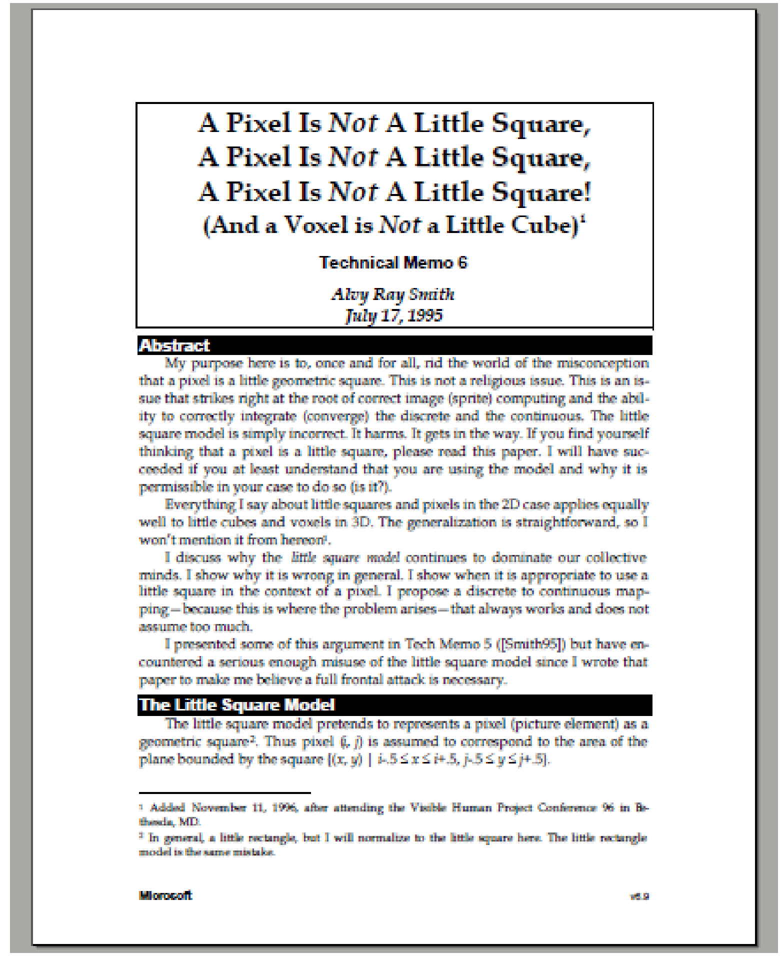
What is a pix-el?



Not a little square!

- ***A Pixel Is* Not *A Little Square, A Pixel Is* Not *A Little Square, A Pixel Is* Not *A Little Square! (And a Voxel is* Not *a Little Cube),***
  - **Alvy Ray Smith,**

  MS Tech Memo 6, Jul 17, 1995



A Pixel Is *Not* A Little Square,
A Pixel Is *Not* A Little Square,
A Pixel Is *Not* A Little Square!
(And a Voxel is *Not* a Little Cube)[1]

Technical Memo 6

Alvy Ray Smith
July 17, 1995

Abstract

My purpose here is to, once and for all, rid the world of the misconception that a pixel is a little geometric square. This is not a religious issue. This is an issue that strikes right at the root of correct image (sprite) computing and the ability to correctly integrate (converge) the discrete and the continuous. The little square model is simply incorrect. It harms. It gets in the way. If you find yourself thinking that a pixel is a little square, please read this paper. I will have succeeded if you at least understand that you are using the model and why it is permissible in your case to do so (is it?).

Everything I say about little squares and pixels in the 2D case applies equally well to little cubes and voxels in 3D. The generalization is straightforward, so I won't mention it from hereon[1].

I discuss why the *little square model* continues to dominate our collective minds. I show why it is wrong in general. I show when it is appropriate to use a little square in the context of a pixel. I propose a discrete to continuous mapping—because this is where the problem arises—that always works and does not assume too much.

I presented some of this argument in Tech Memo 5 ([Smith95]) but have encountered a serious enough misuse of the little square model since I wrote that paper to make me believe a full frontal attack is necessary.

The Little Square Model

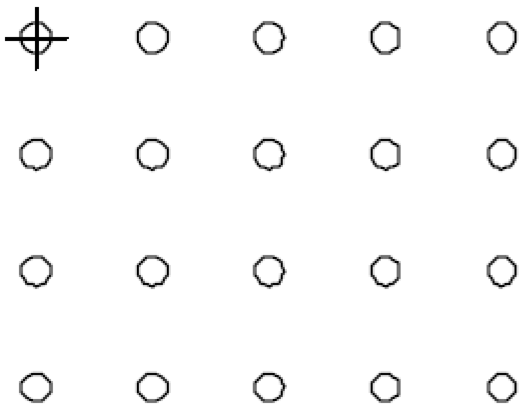
The little square model pretends to represents a pixel (picture element) as a geometric square[2]. Thus pixel (i, j) is assumed to correspond to the area of the plane bounded by the square $\{(x, y) \mid i\text{-}.5 \le x \le i\text{+}.5,\ j\text{-}.5 \le y \le j\text{+}.5\}$.

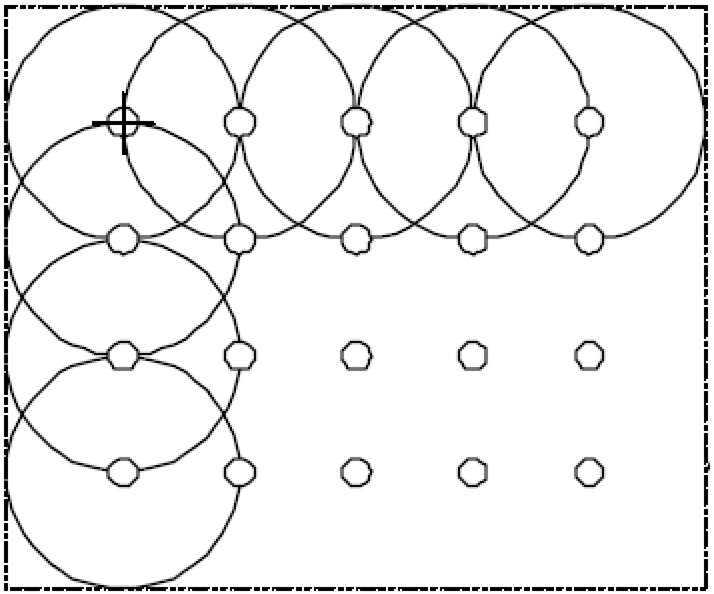
[1] Added November 11, 1996, after attending the Visible Human Project Conference 96 in Bethesda, MD.

[2] In general, a little rectangle, but I will normalize to the little square here. The little rectangle model is the same mistake.

Microsoft v5.9

Not a little square!

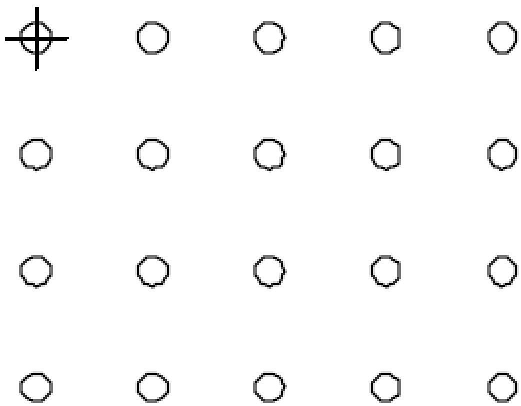


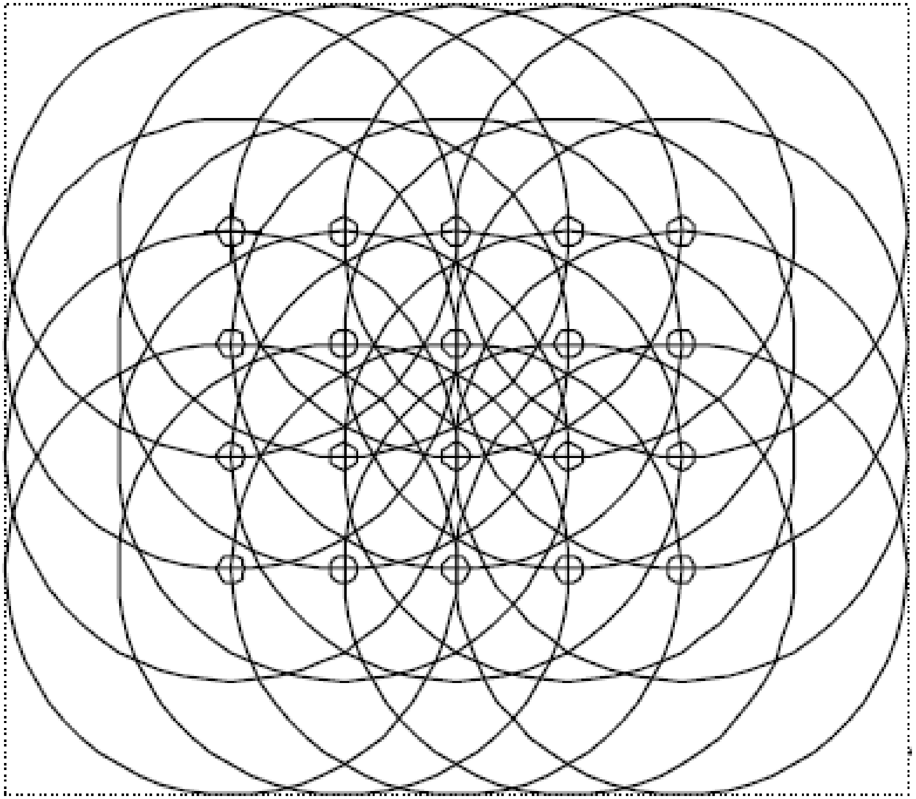


Gaussian reconstruction filter

Illustrations: Smith, MS Tech Memo 6, Jul 17, 1995

Not a little square!





Cubic reconstruction filter

Illustrations: Smith, MS Tech Memo 6, Jul 17, 1995

Not a little square!

Graphics: Dick Lyon, 2006

Where do images come from?

- Digital cameras
- MRI scanners
- Computer graphics packages
- Body scanners
- Laser range finders

- Many more…

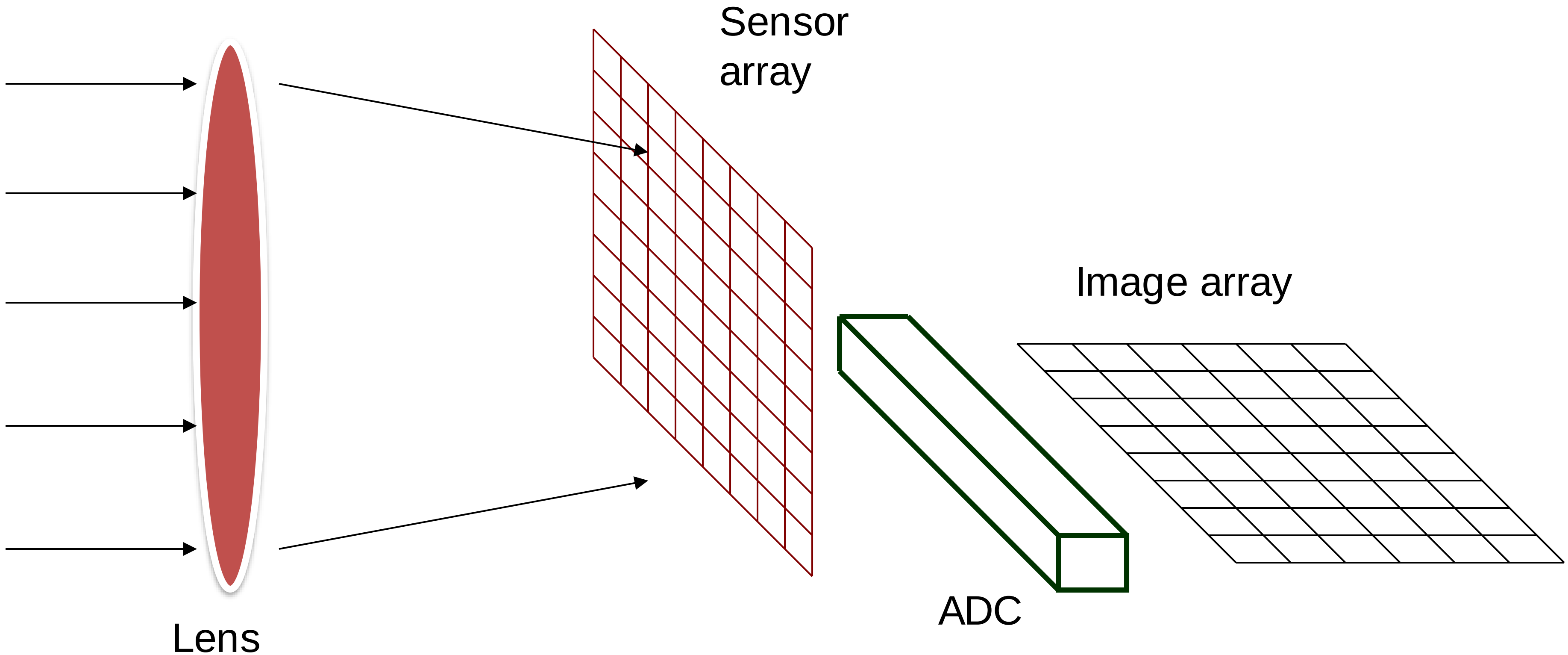
Where do images come from?

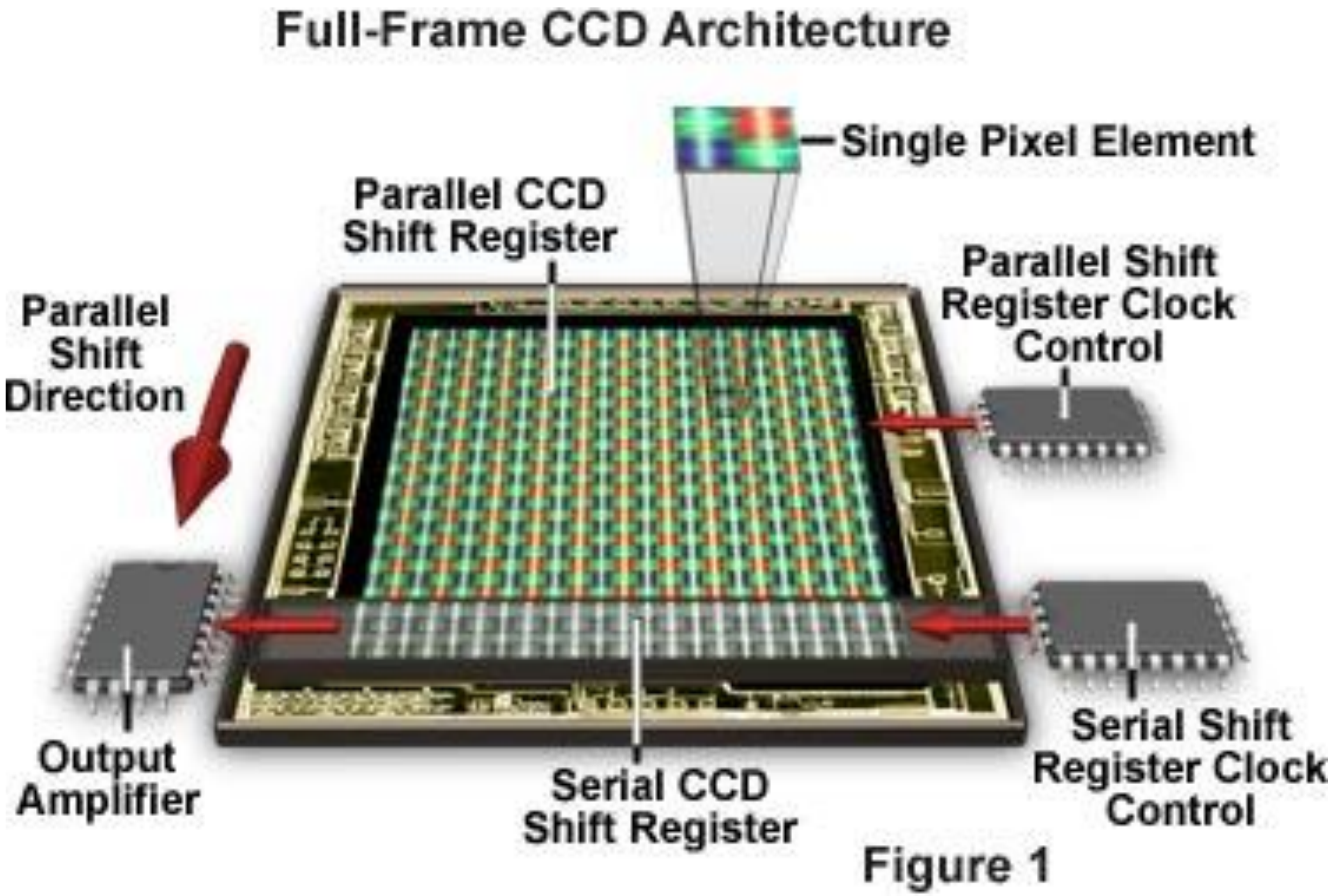
- **Digital cameras**
- MRI scanners
- Computer graphics packages
- Body scanners
- Laser range finders

- Many more…

The digital camera

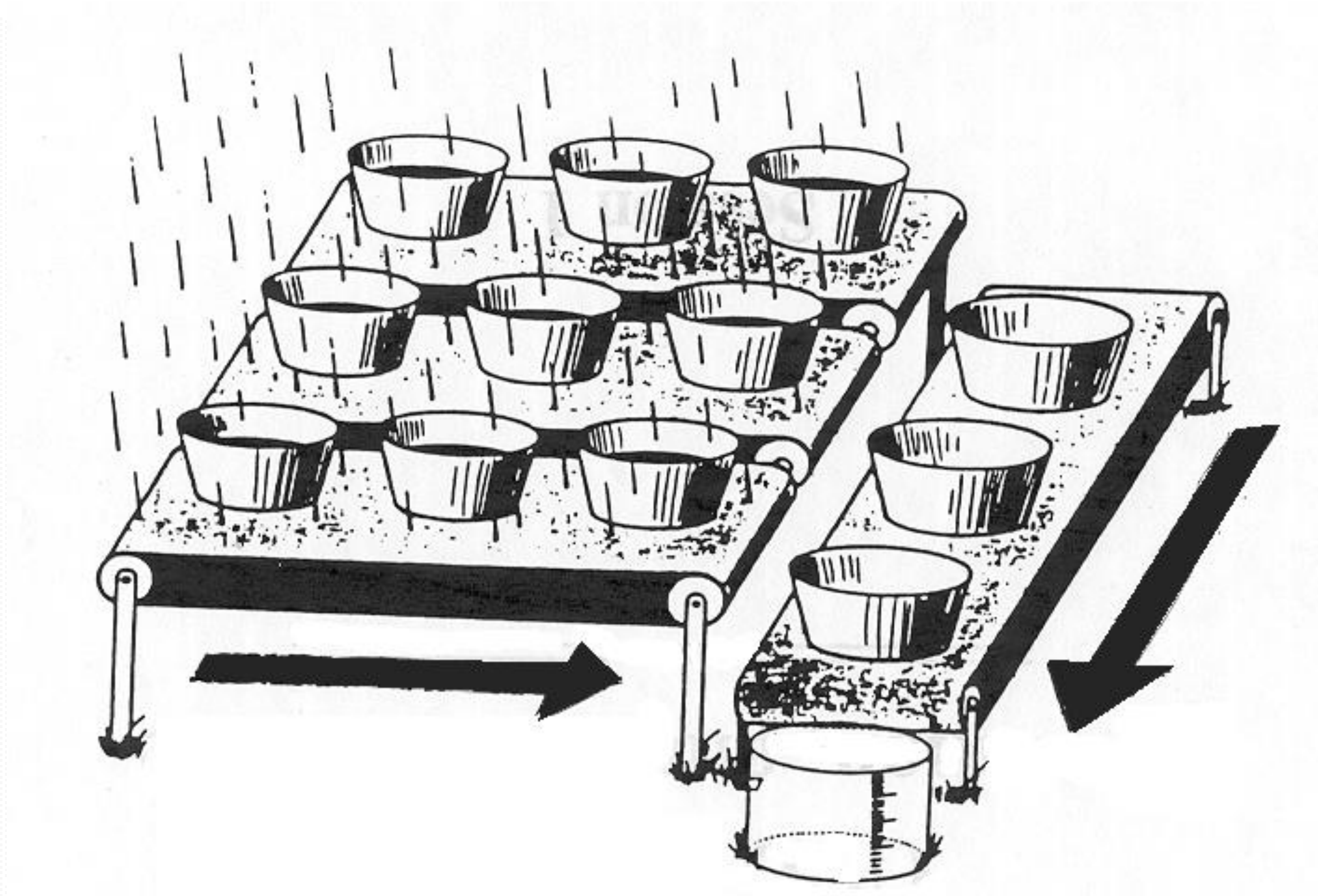
- A *Charge Coupled Device* (CCD).





http://www.astro.virginia.edu/class/oconnell/astr121/im/CCD-fullframearc-FSU.jpg

Capturing photons

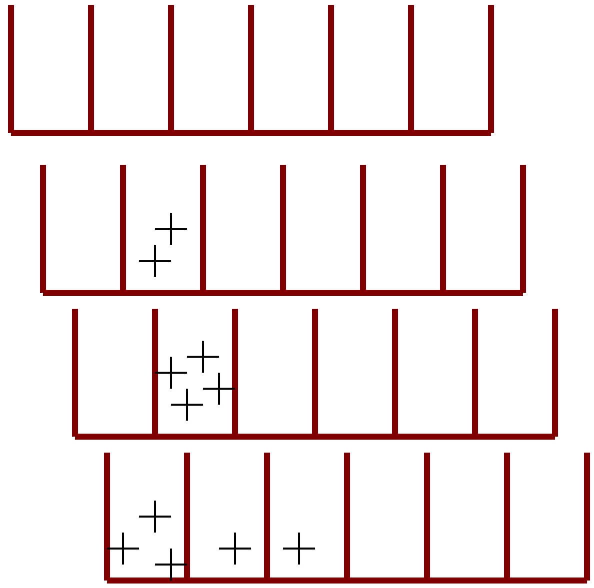


From: Lecture Notes – EAAE and/or Science "Nuggets" 2000

ETH

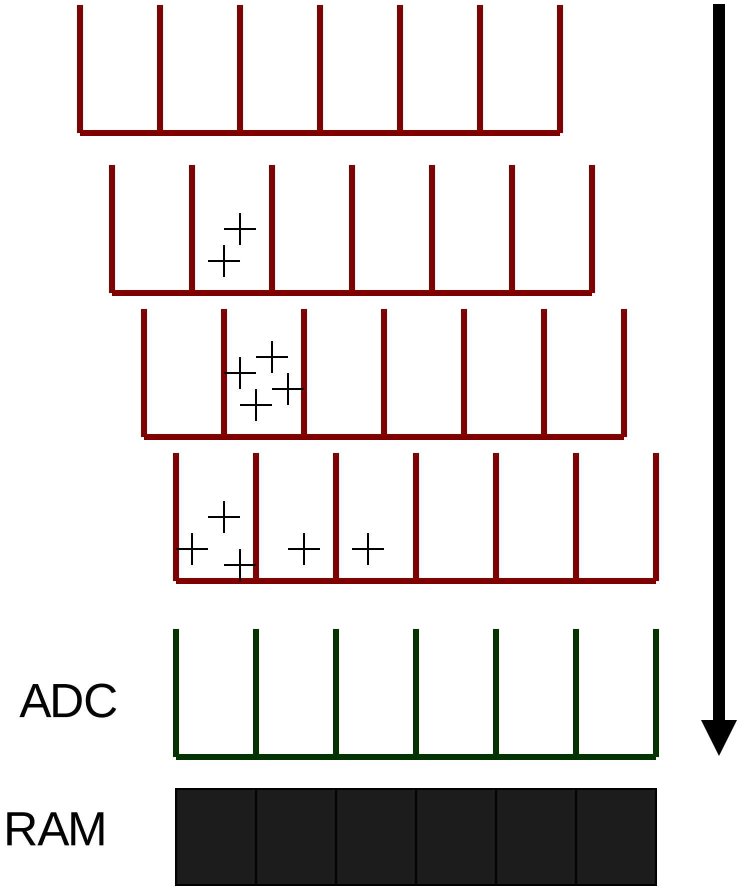
The sensor array

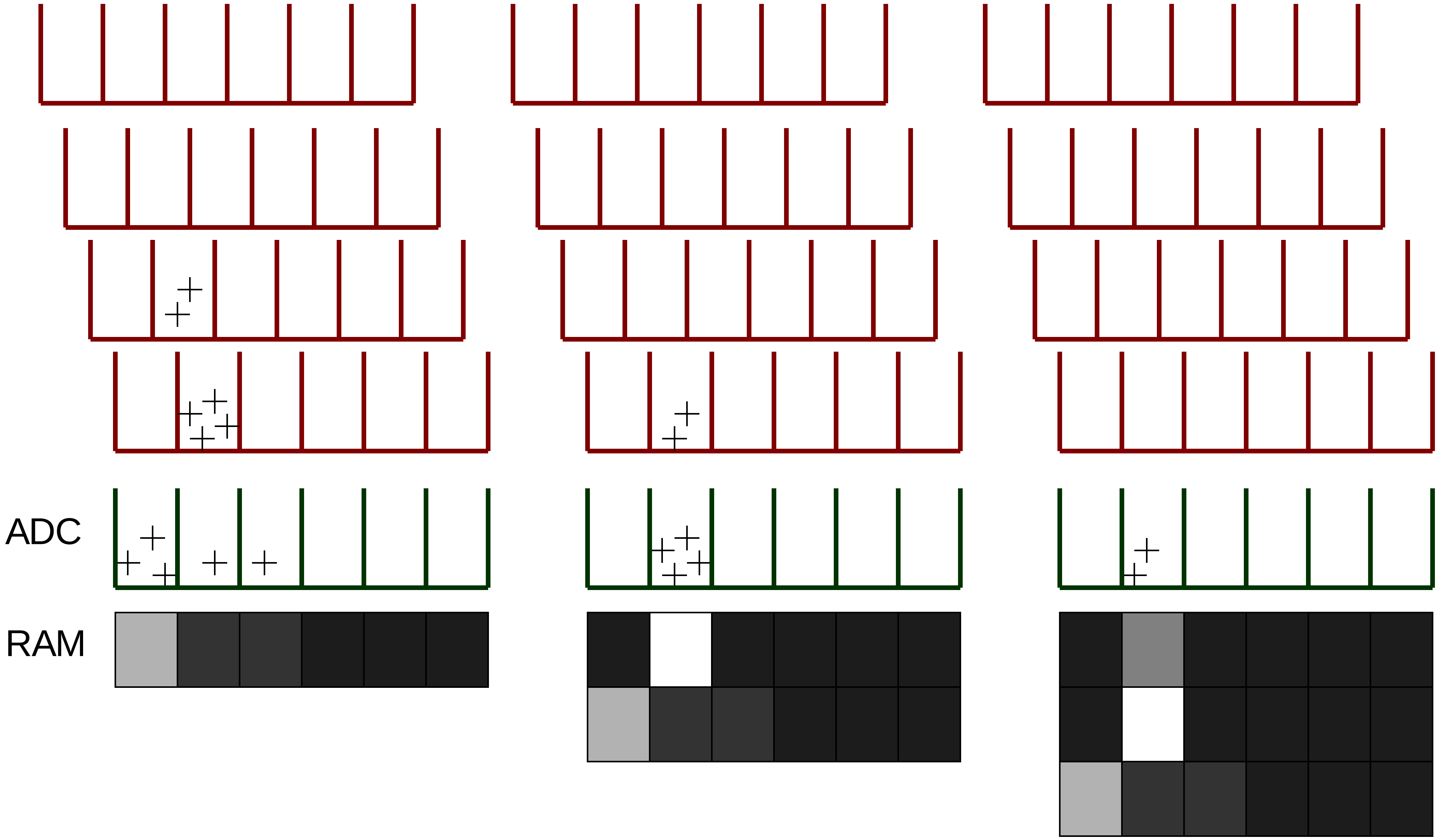
- Can be $< 1cm^2$.
- An array of *photosites*.
- Each photosite is a bucket of electrical charge.
- They contain charge proportional to the incident light intensity during exposure.



Analog to Digital Conversion

- The ADC measures the charge and digitizes the result.
- Conversion happens line by line.
- The charges in each photosite move down through the sensor array.

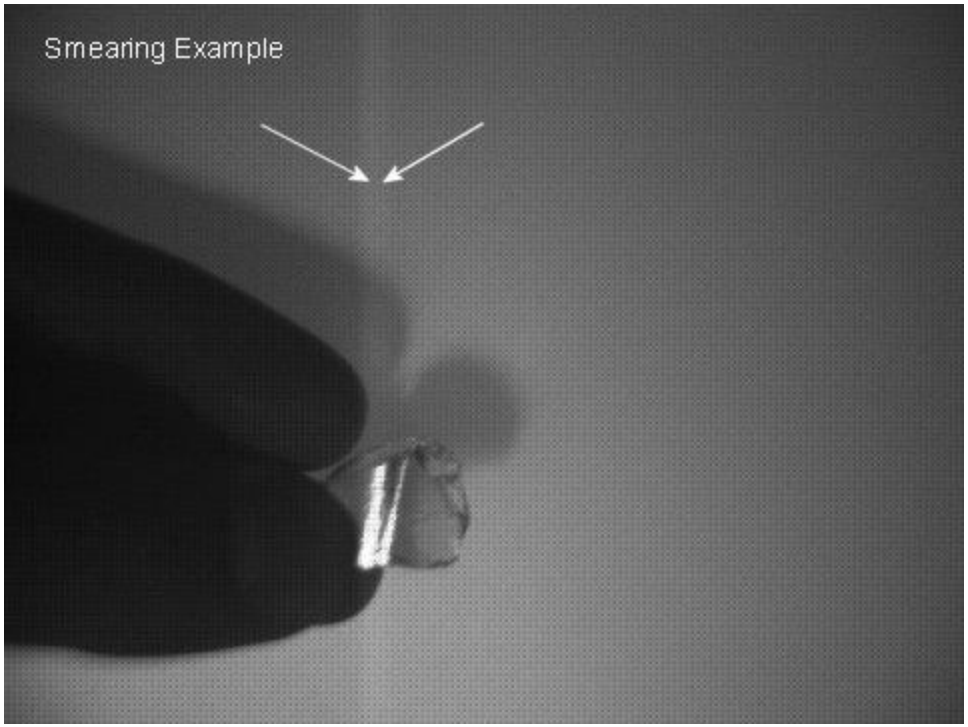


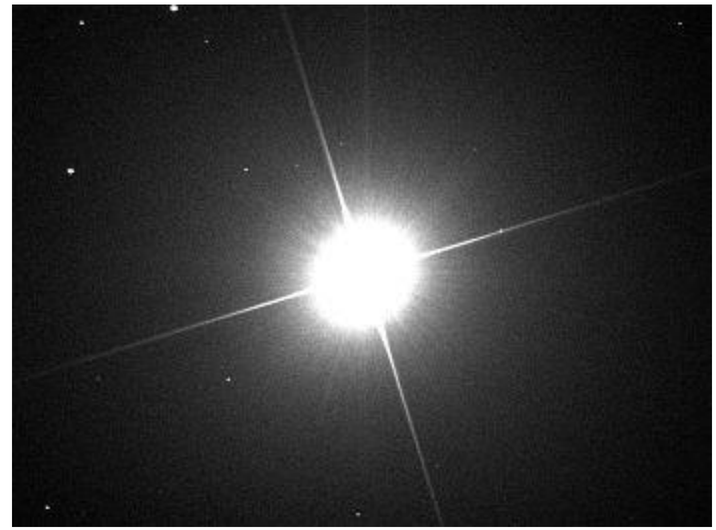

ADC
RAM

Blooming

- The buckets have finite capacity
- Photosite saturation causes blooming

Bleeding or smearing

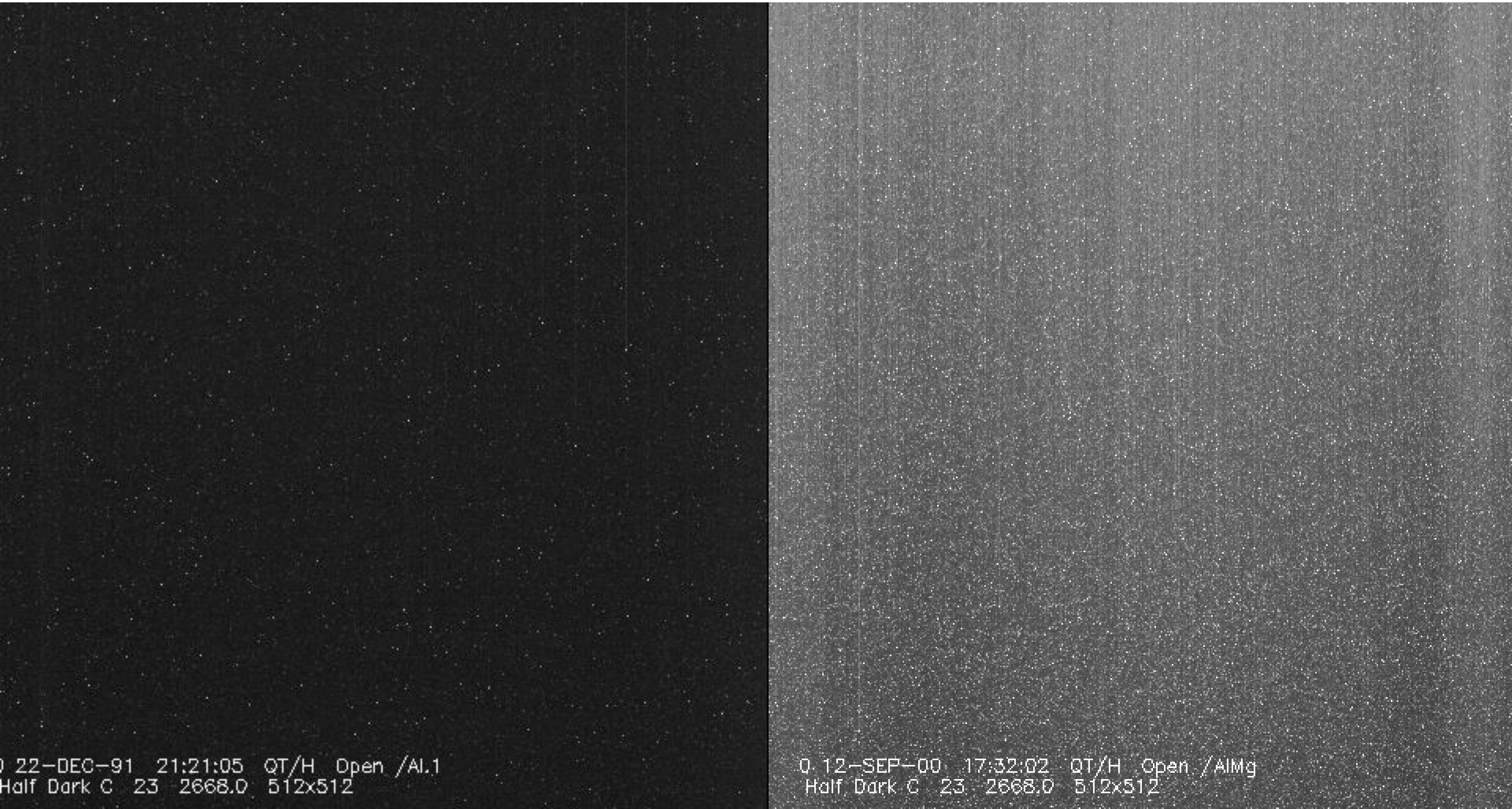




During transit buckets still accumulate some charges
Influenced by time ‘in transit’ versus integration time
Effect is worse for short shutter times (only problem with electronic shutter)

Dark Current

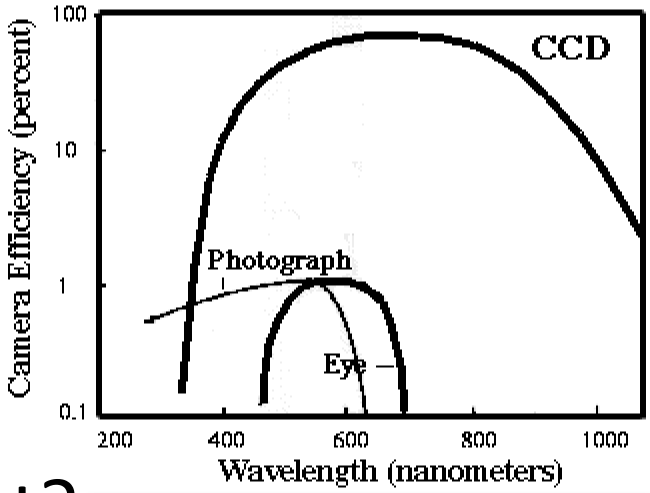
Yohkoh satellite, 9 years apart [..]()



Dark Current

- CCDs produce thermally-generated charge.
- They give non-zero output even in darkness.
- Partly, this is the *dark current*.
- Fluctuates randomly.

- How can we reduce dark current?



From: Lecture Notes - EAAE

CMOS

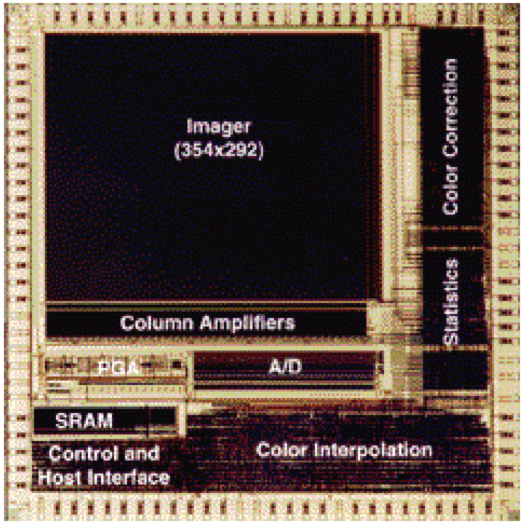
Same sensor elements as CCD

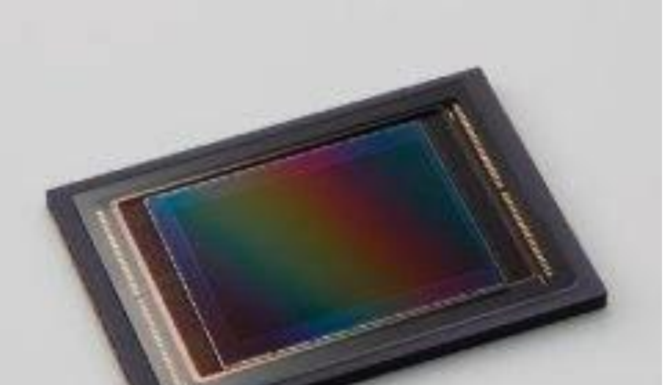
Each photo sensor has its own amplifier

- More noise (reduced by subtracting 'black' image)
- Lower sensitivity (lower fill rate)

Uses standard CMOS technology

- Allows to put other components on chip
- 'Smart' pixels



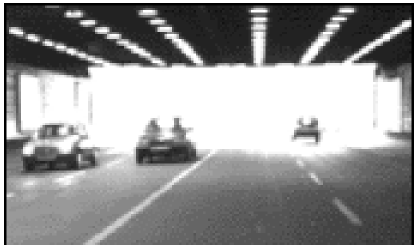


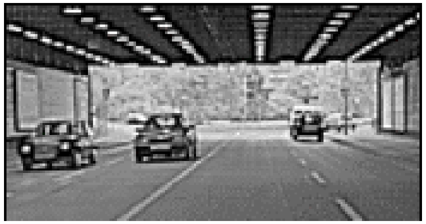
Canon sensor
120Mpix@9.5fps burst

CCD vs. CMOS

- Mature technology
- Specific technology
- High production cost
- High power consumption
- Higher fill rate
- Blooming
- Sequential readout

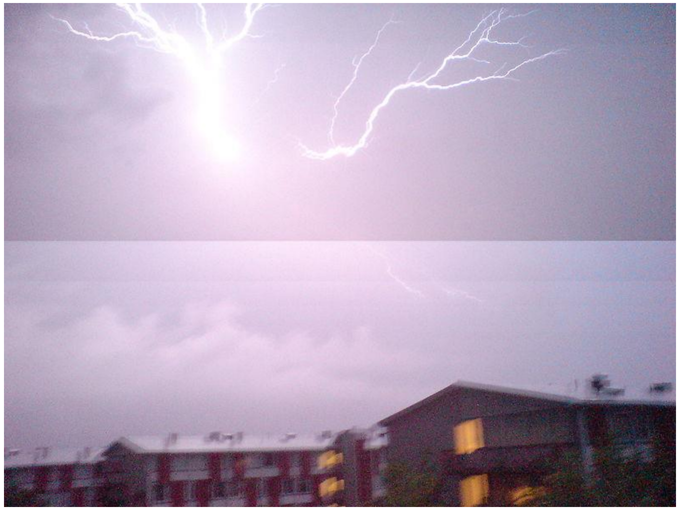
- More recent technology
- Standard IC technology
- Cheap
- Low power
- Less sensitive
- Per pixel amplification
- Random pixel access
- Smart pixels
- On chip integration with other components





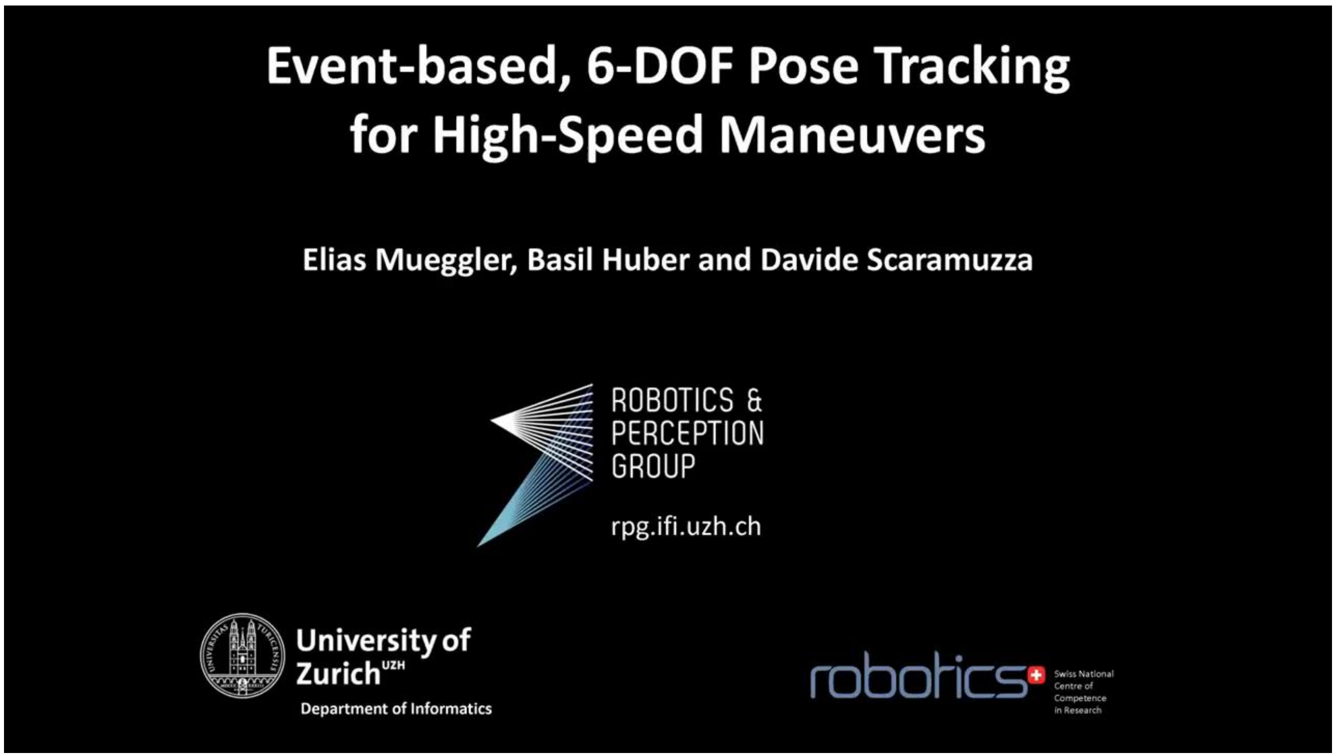
CMOS video sensor issues

- Rolling shutter
  - Sequential read-out of lines



[Video]

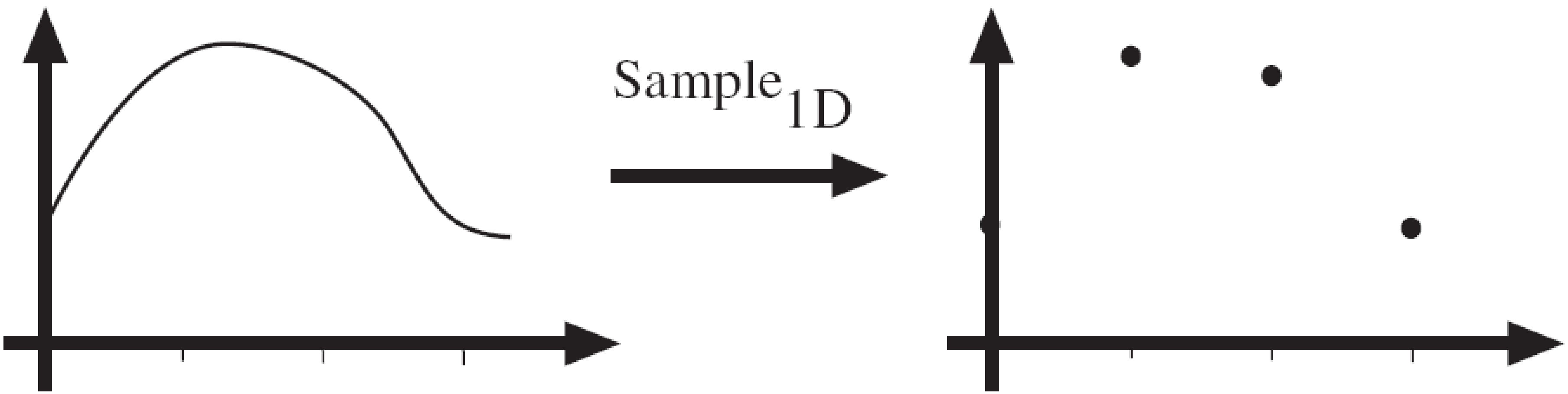
DVS camera



DVS event camera from INI labs (spin-off UNIZ/ETHZ inst. neuro-inf.)

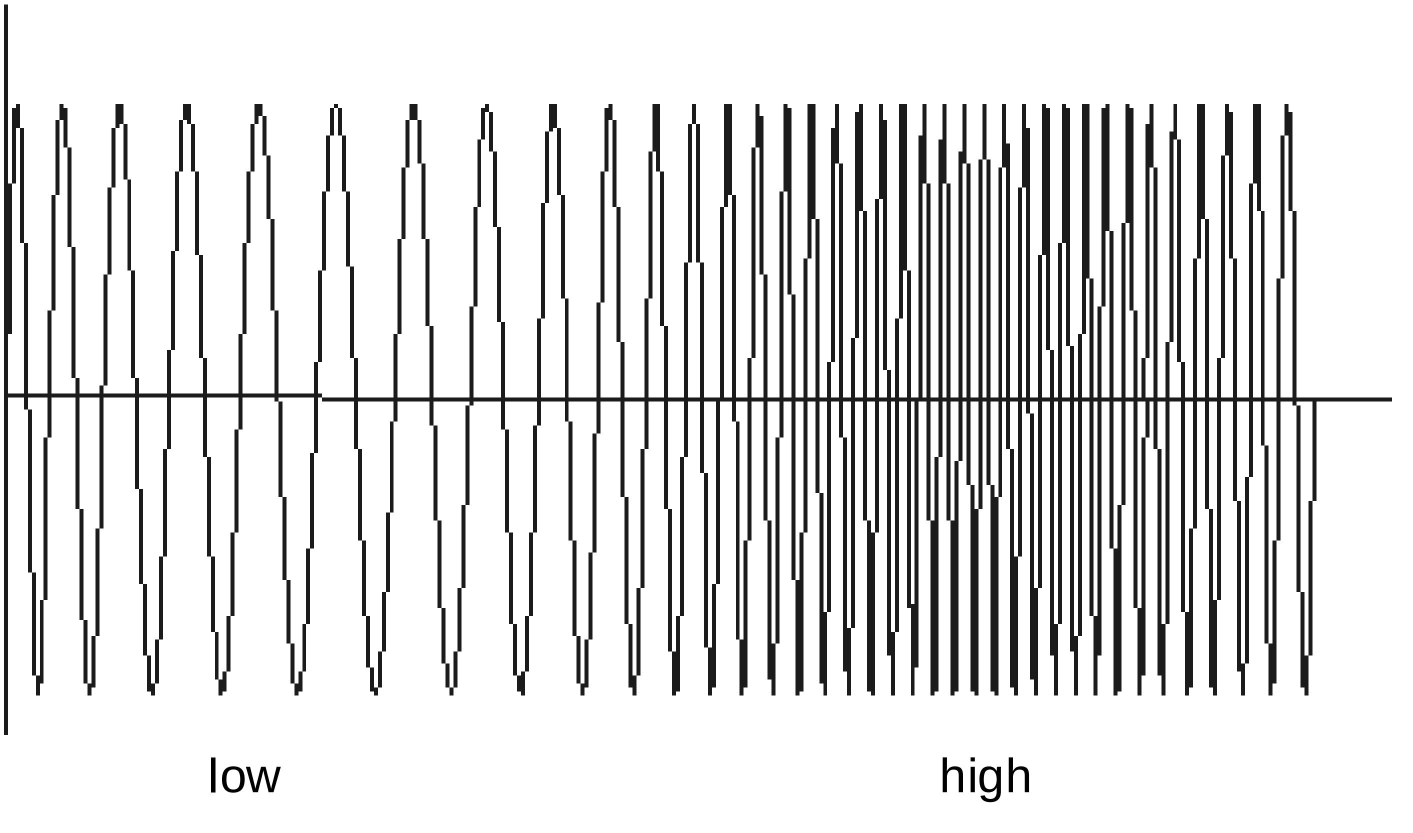
Camera inspired by human visual system

Sampling 1D



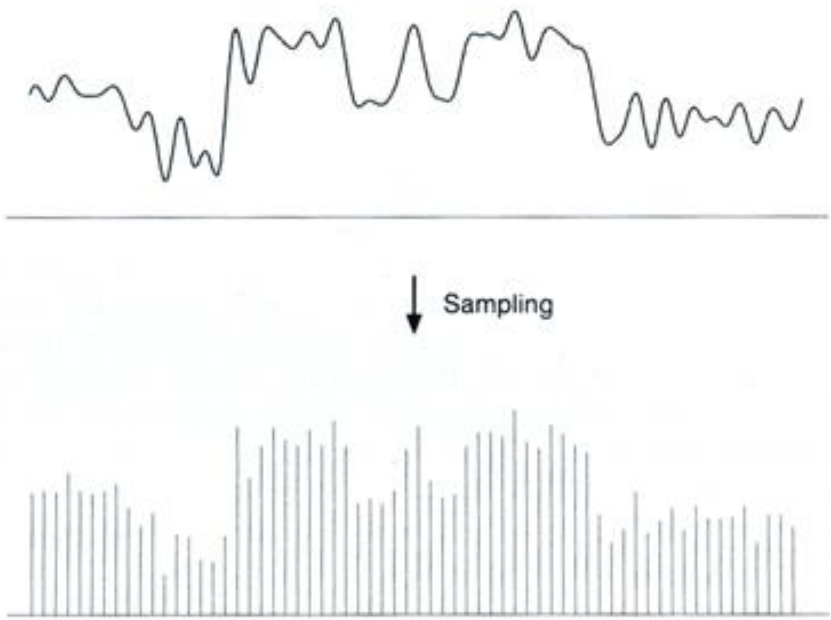
Sampling in 1D takes a function, and returns a vector whose elements are values of that function at the sample points

1D Example: Audio



Sampled representations

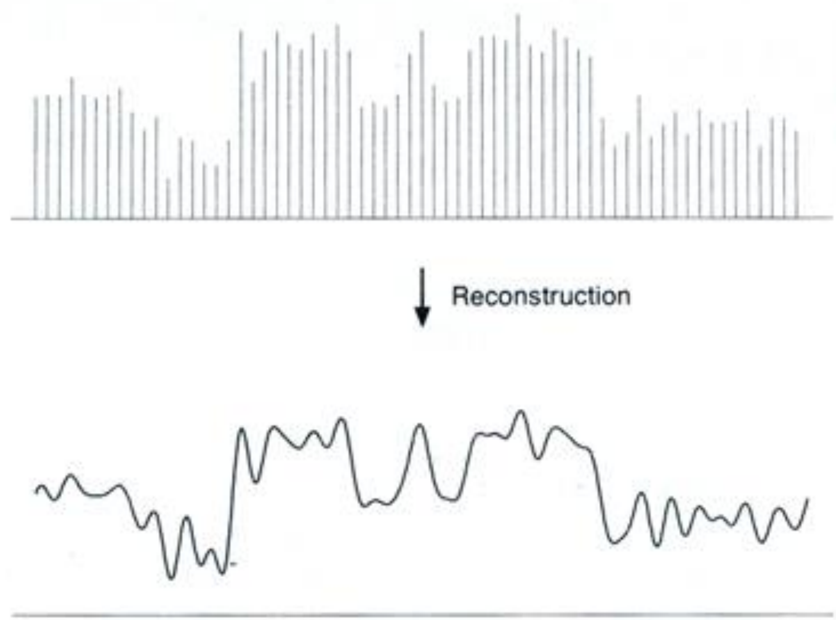
- How to store and compute with continuous functions?
- Common scheme for representation: samples
  - write down the function's values at many points



[FvDFH fig.14.14b / Wolberg]

Reconstruction

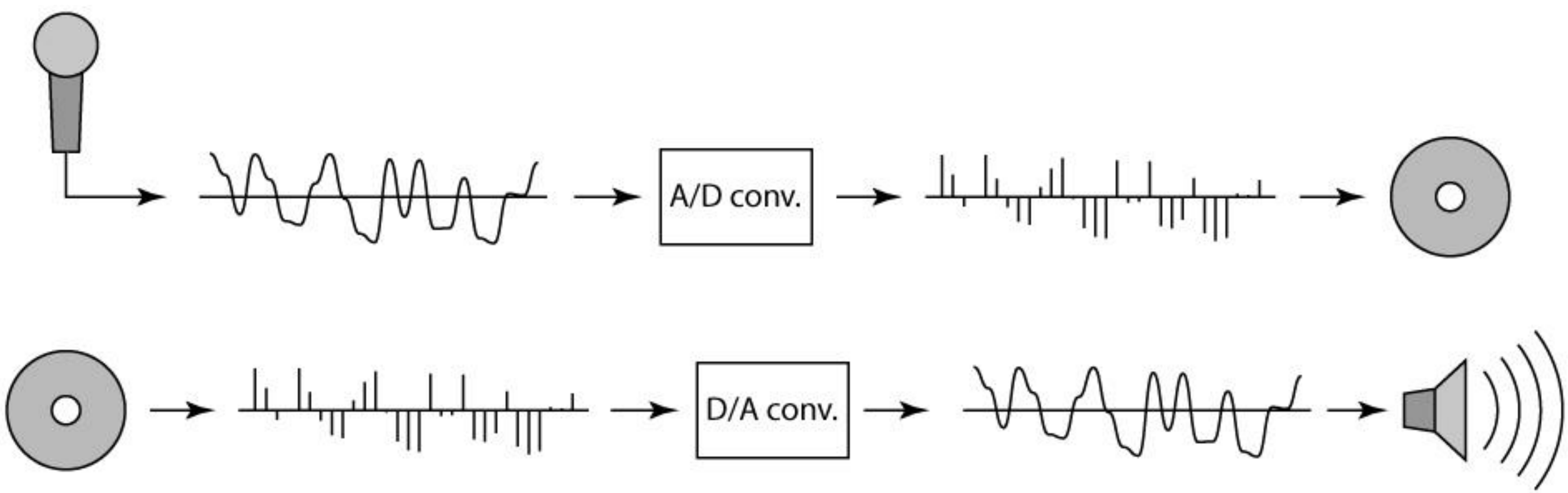
- Making samples back into a continuous function
  - for output (need realizable method)
  - for analysis or processing (need mathematical method)
  - amounts to “guessing” what the function did in between



[FvDFH fig.14.14b / Wolberg]

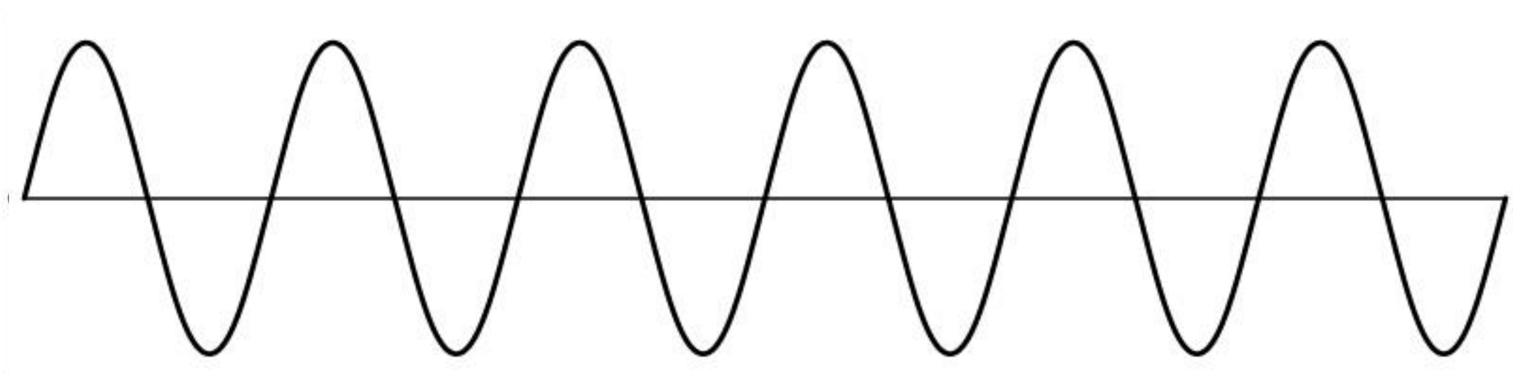
Sampling in digital audio

- Recording: sound to analog to samples to disc
- Playback: disc to samples to analog to sound again
  – how can we be sure we are filling in the gaps correctly?



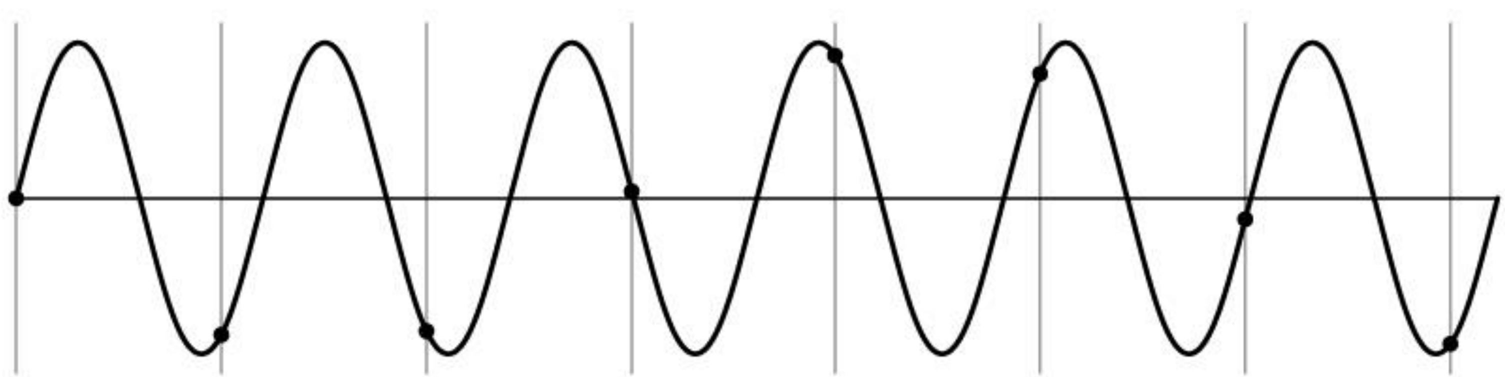
Sampling and Reconstruction

- Simple example: a sine wave



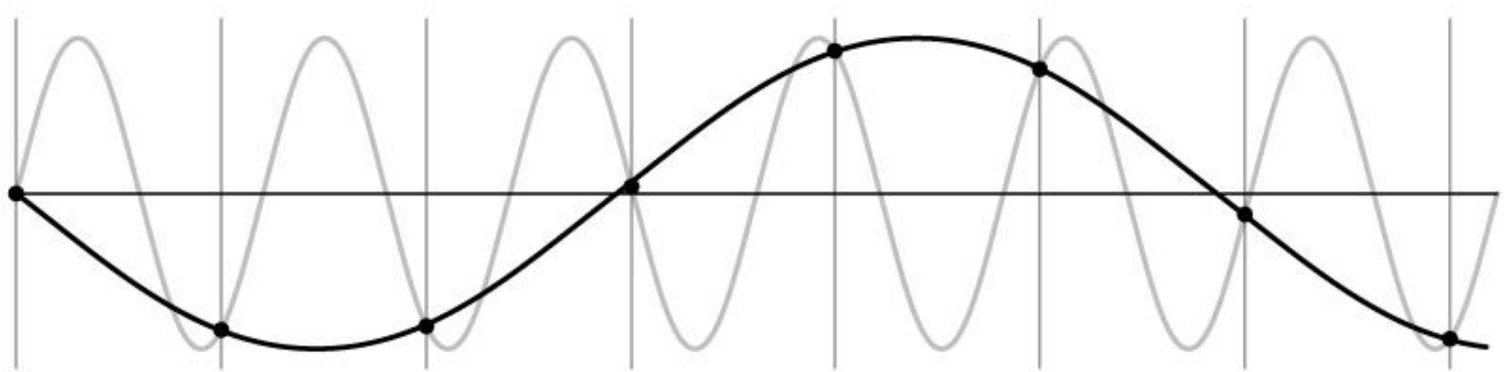
Undersampling

- What if we "missed" things between the samples?
- Simple example: undersampling a sine wave
  - unsurprising result: information is lost



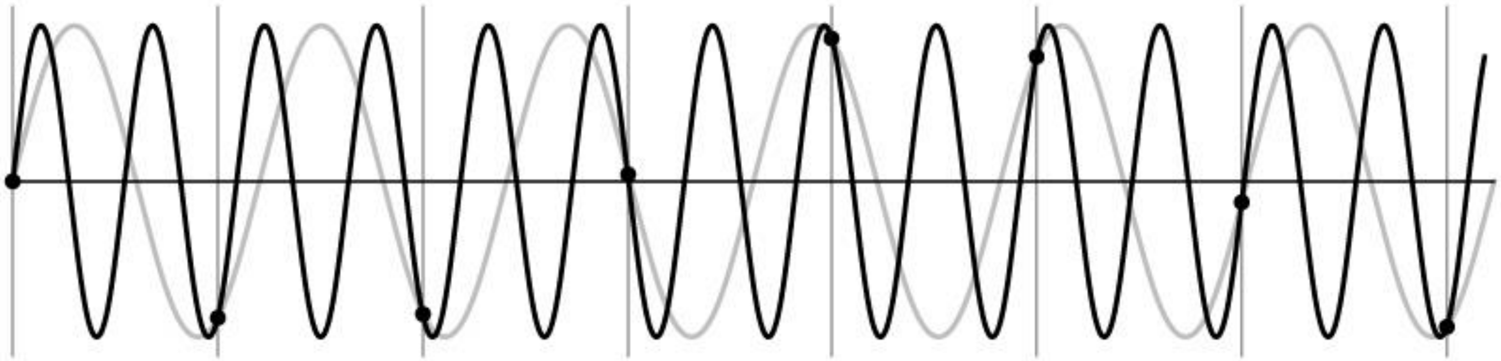
Undersampling

- What if we “missed” things between the samples?
- Simple example: undersampling a sine wave
  – unsurprising result: information is lost
  – surprising result: indistinguishable from lower frequency



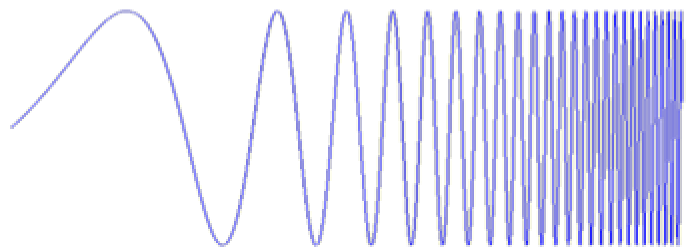
Undersampling

- What if we "missed" things between the samples?
- Simple example: undersampling a sine wave
  - unsurprising result: information is lost
  - surprising result: indistinguishable from lower frequency
  - also was always indistinguishable from higher frequencies
  - *aliasing*: signals "traveling in disguise" as other frequencies

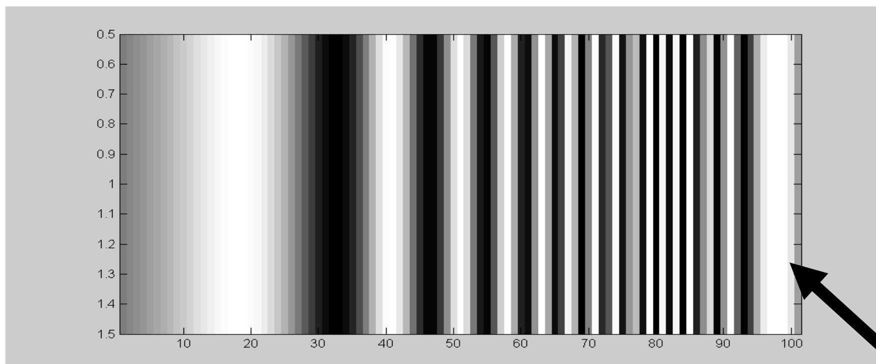


What’s happening?

Input signal:

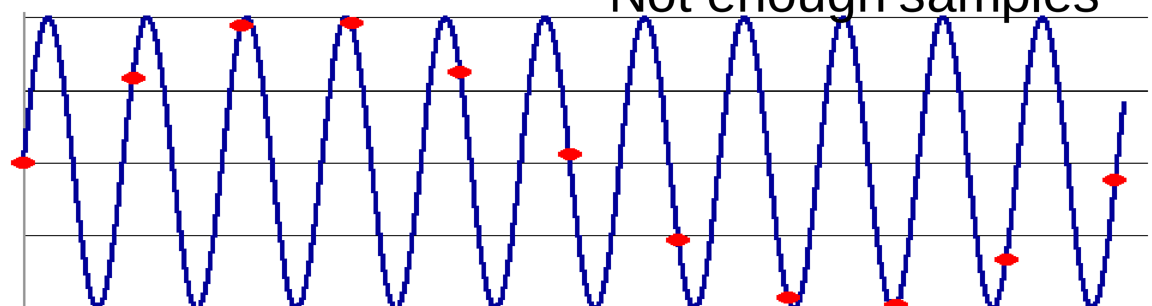


Plot as image:

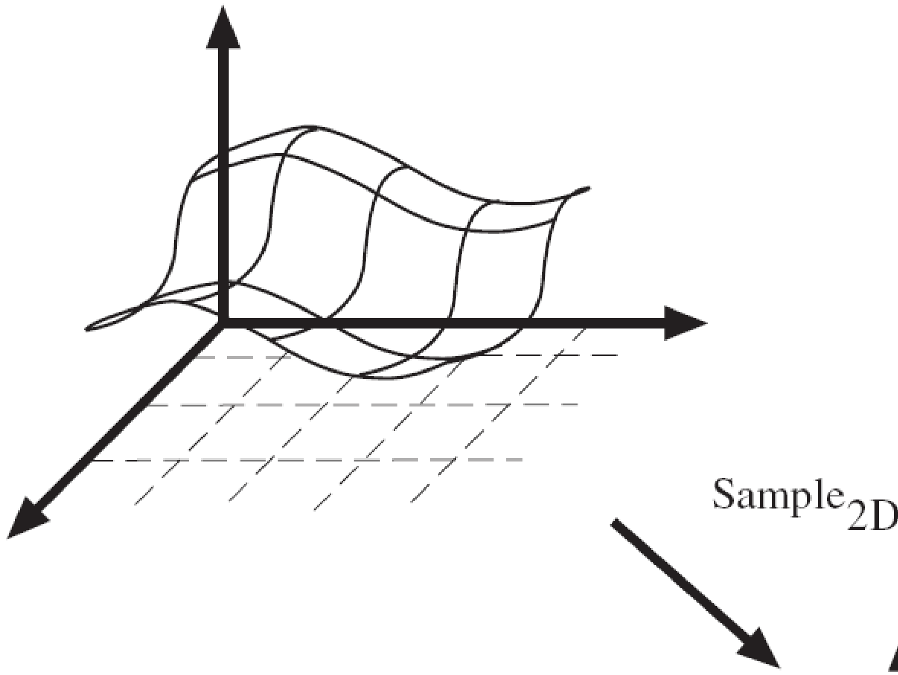


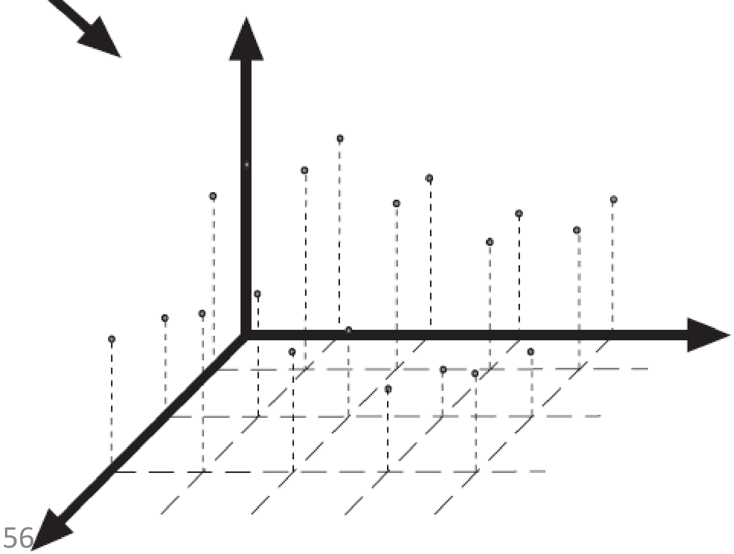
x = 0:.05:5; imagesc(sin((2.^x).*x))

Alias!
Not enough samples



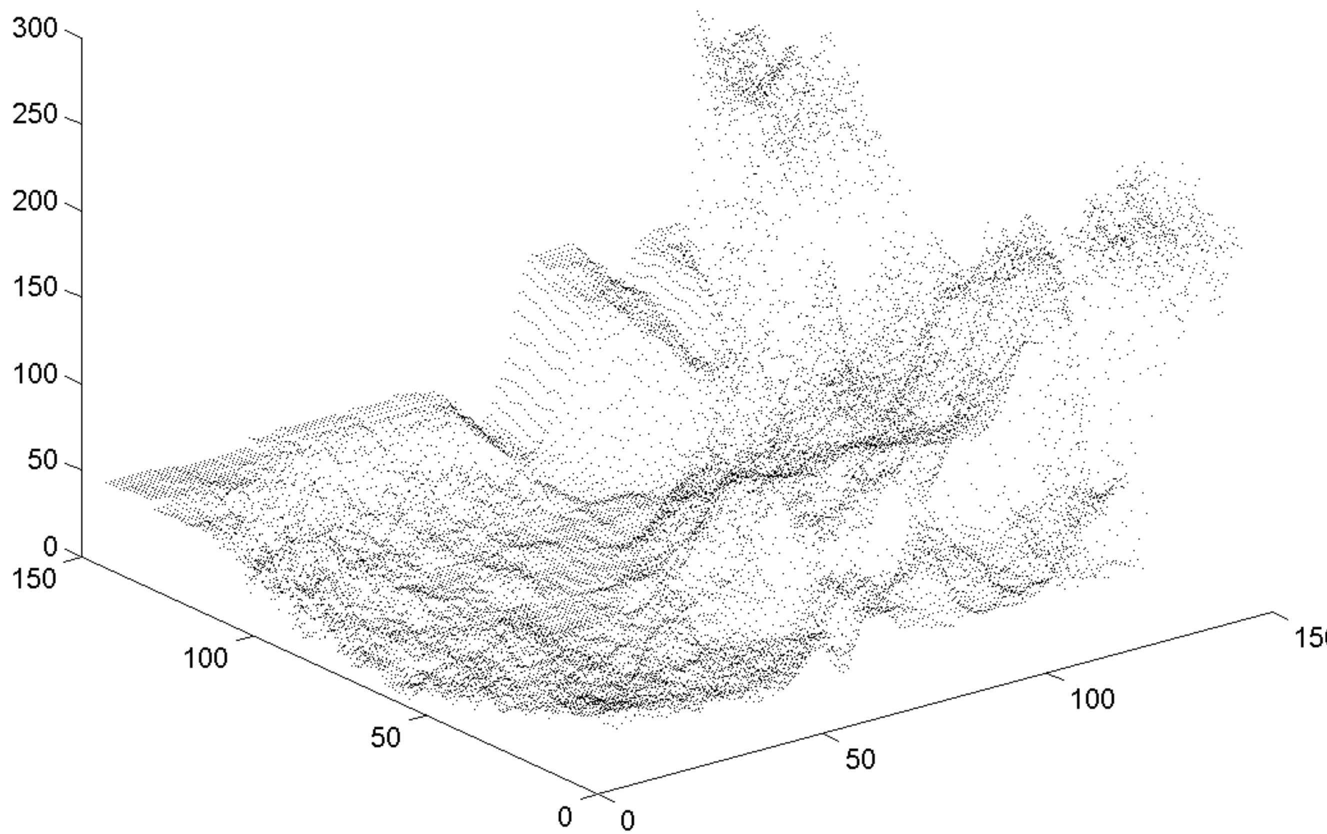
Sampling 2D





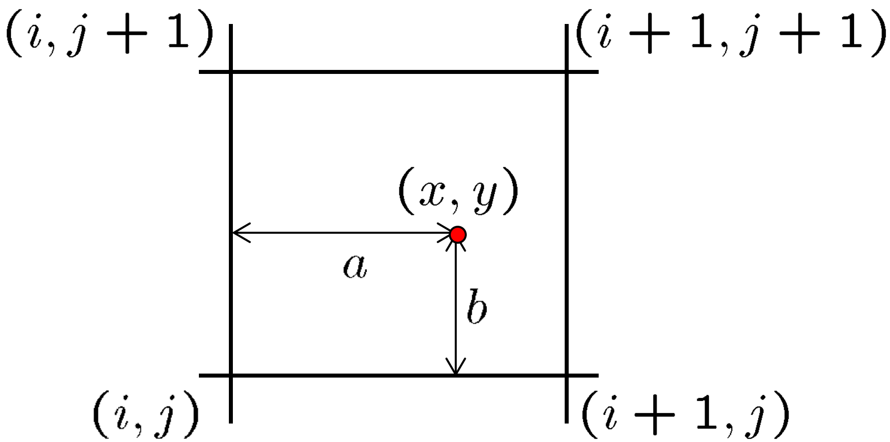
Sampling in 2D takes a function and returns an array; we allow the array to be infinite dimensional and to have negative as well as positive indices.

Greyscale digital image



Reconstructing continuous signal

- e.g. Bilinear interpolation



$$
\begin{aligned}
f(x,y) = \quad & (1-a)(1-b) \quad && f[i,j] \\
& +a(1-b) && f[i+1,j] \\
& +ab && f[i+1,j+1] \\
& +(1-a)b && f[i,j+1]
\end{aligned}
$$

Nyquist Frequency

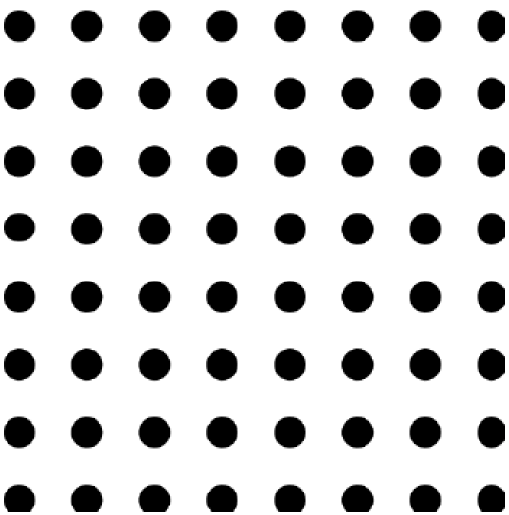
(a.k.a. Nyquist–Shannon sampling theorem)

- Half the sampling frequency of a discrete signal processing system

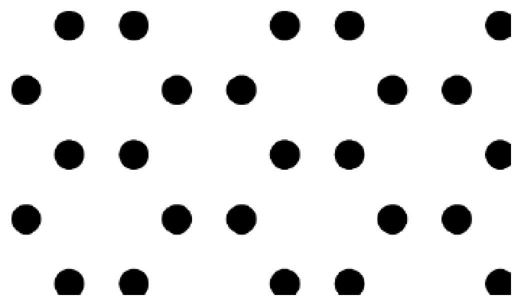
- Signal's max frequency (bandwidth) must be **smaller*** than this

*In later lectures: coping when it's >=.

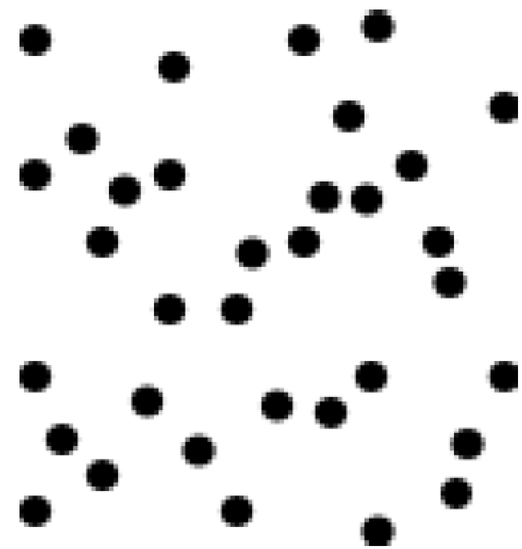
Sampling grids



Cartesian sampling

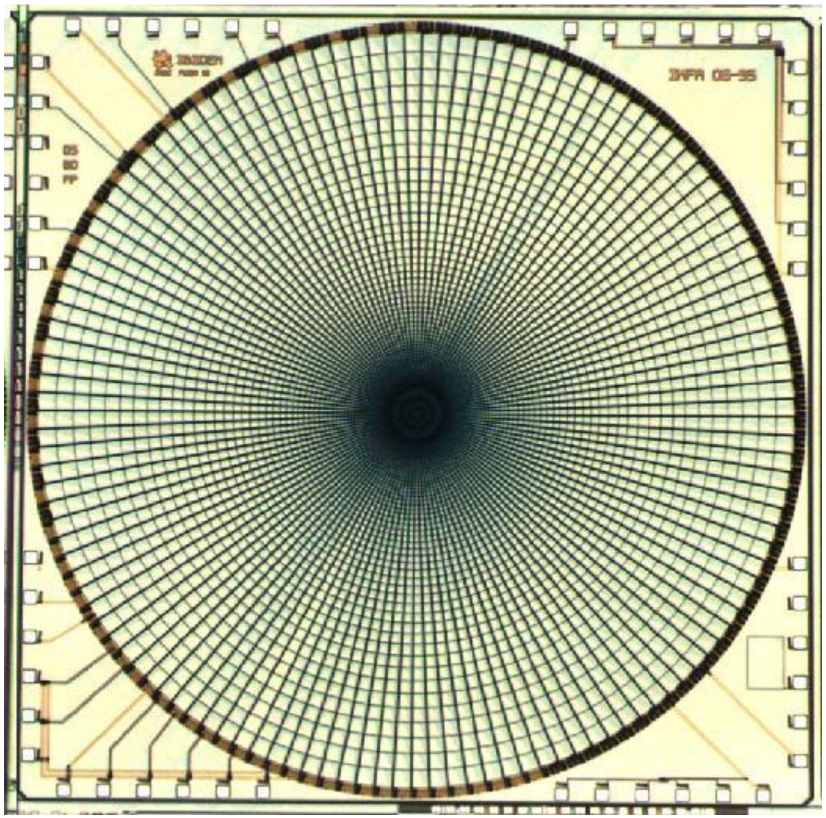


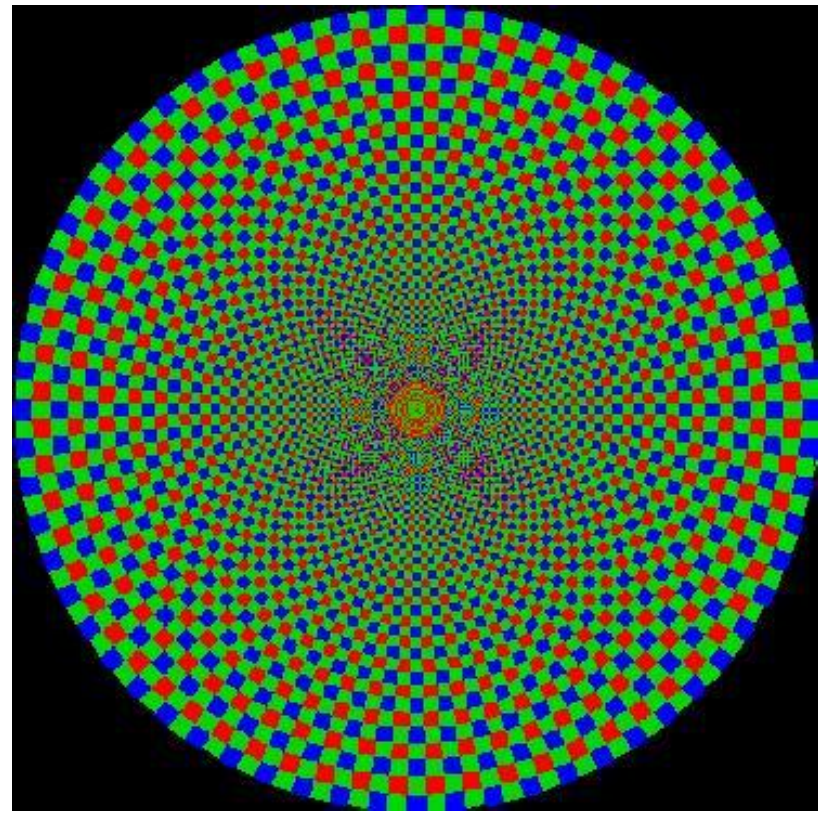
Hexagonal sampling



Non-uniform sampling

Retina-like sensors



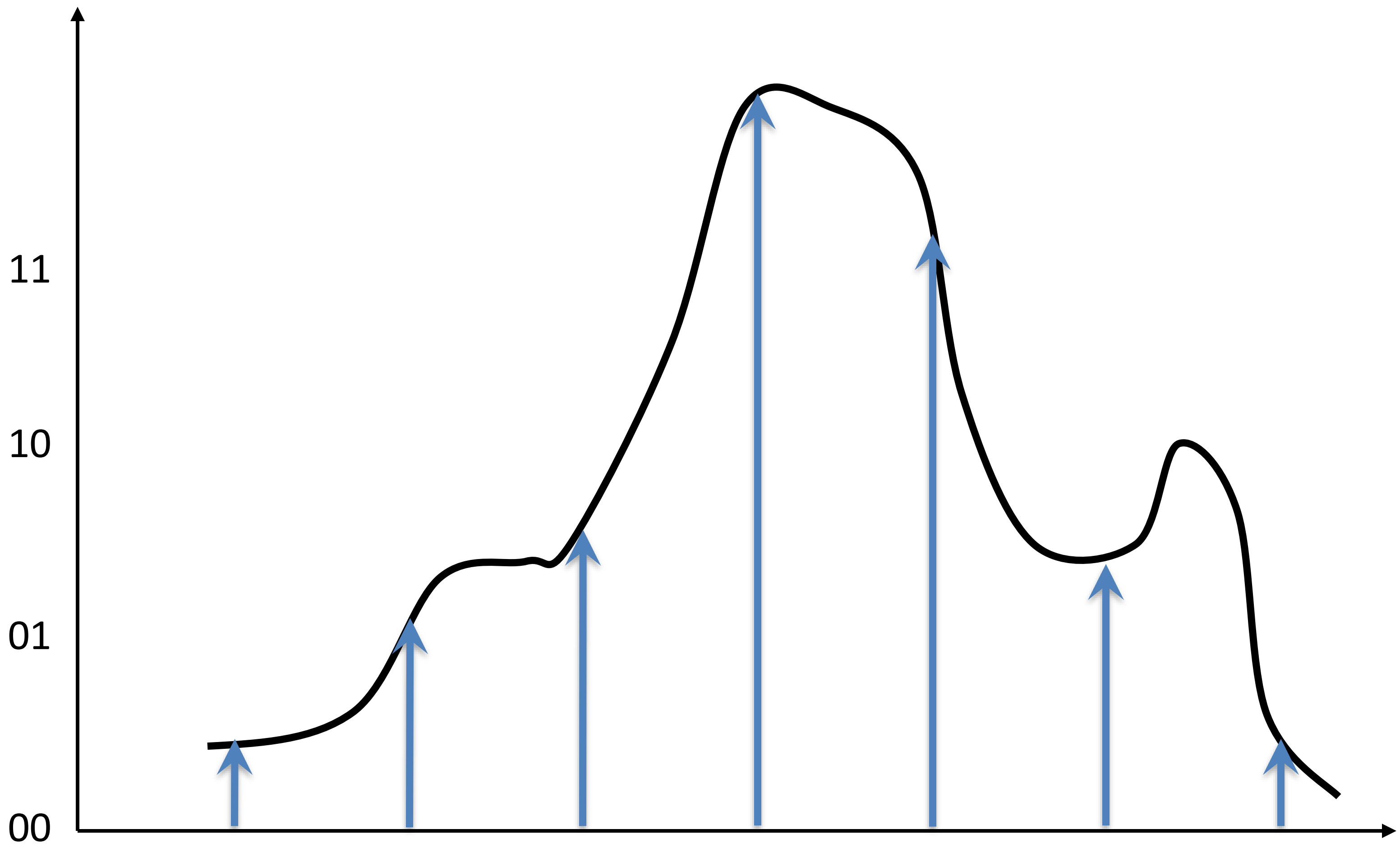


Quantization

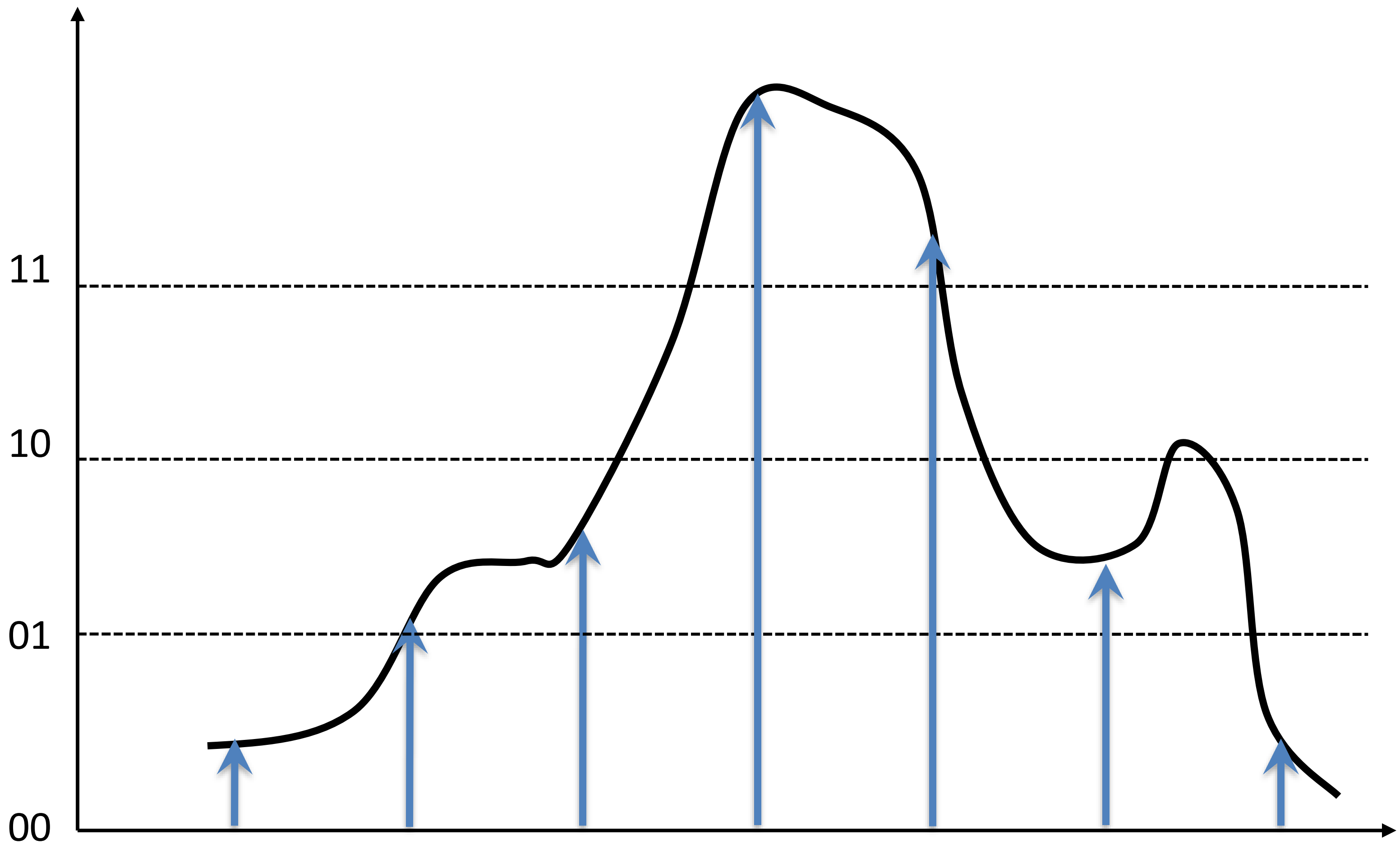
- Real valued function will get digital values – integer values
- Quantization is lossy!!
  - After quantization, the original signal cannot be reconstructed anymore
- This is in contrast to sampling, as a sampled but not quantized signal **can** be reconstructed.
- Simple quantization uses equally spaced levels with k intervals

$$\boxed{k = 2^b}$$

Quantization



Quantization



ETH

Quantization

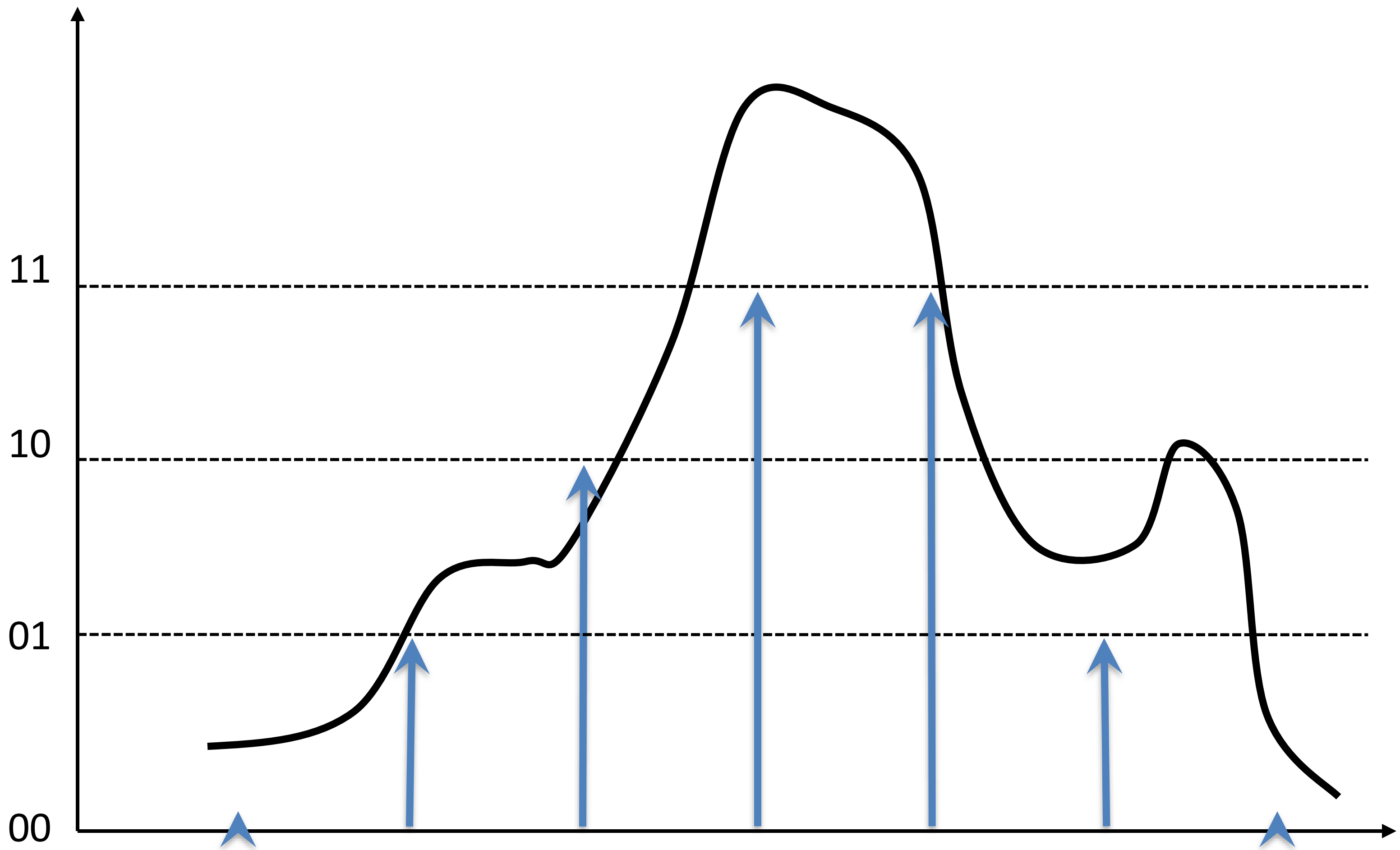


Image Properties

- Image resolution
- Geometric resolution: How many pixels per area
- Radiometric resolution: How many bits per pixel

Image resolution

1024x1024

512x1024

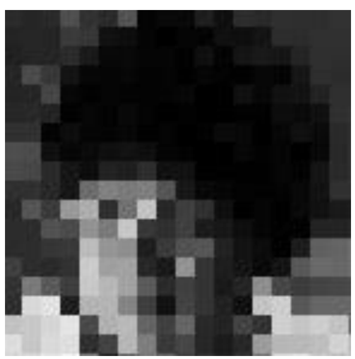
512x512

Geometric resolution

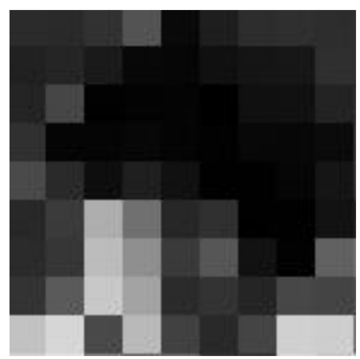
144x144

72x72

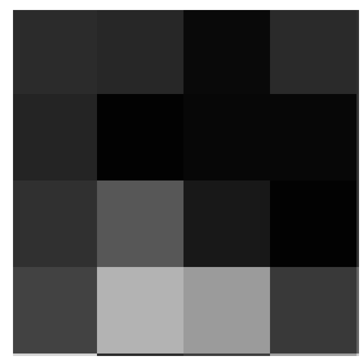
36x36



18x18



9x9



4x4

Radiometric resolution

256

128

64

32

16

8

4

2

Aliasing and SNR

- What is the disadvantage of low sampling resolution?
- What is the disadvantage of high sampling resolution?
- Lossless vs. Lossy
  - Name some formats?

Unassessed Assignment

Use python to change the geometric and radiometric quantization resolution in one of your images. For each level of sampling and quantization, plot the image function, as in slides 71 & 72, and compare the approximations to the true intensity function that you get at each level.

Usual quantization intervals

- Grayscale image
  - 8 bit = 2^8 = 256 grayvalues
- Color image RGB (3 channels)
  - 8 bit/channel = 2^24 = 16.7M colors
- 12bit or 16bit from some sensors
- Nonlinear, for example log-scale

Photo: Paulo Barcellos Jr.

ETH

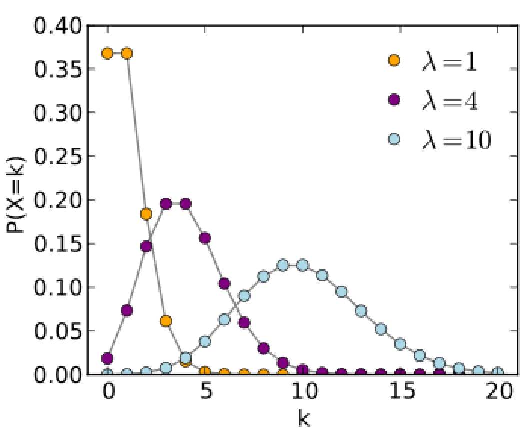
Image Noise

- A common model is *additive Gaussian* noise:

$$I(x,y) = f(x,y) + c$$

where $c \sim N(0,\sigma^2)$. So that $p(c) = (2\pi\sigma^2)^{-1} e^{-c^2/2\sigma^2}$

- Poisson noise:
(shot noise) $p(k) = \dfrac{\lambda^k e^{-\lambda}}{k!}$



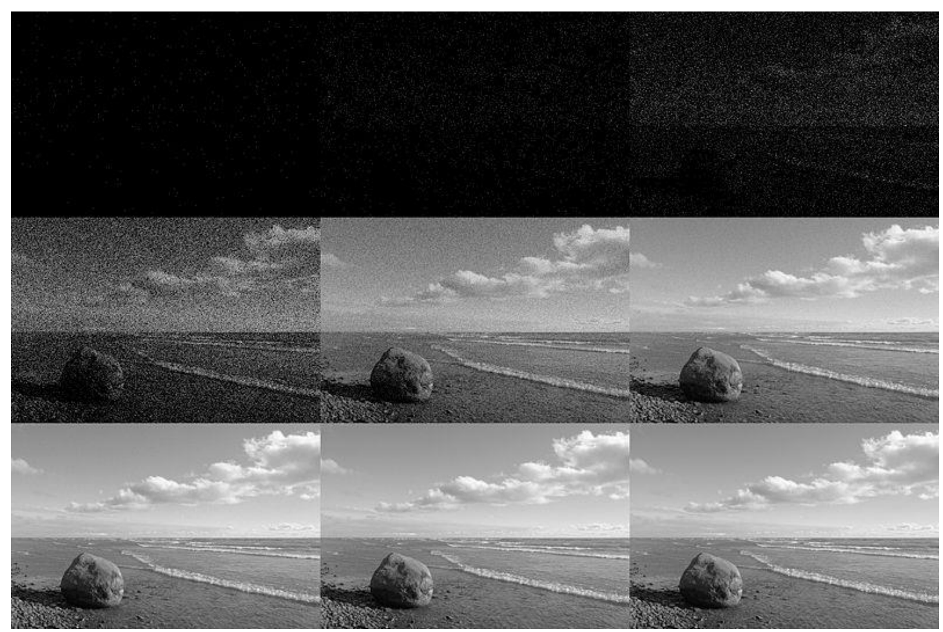
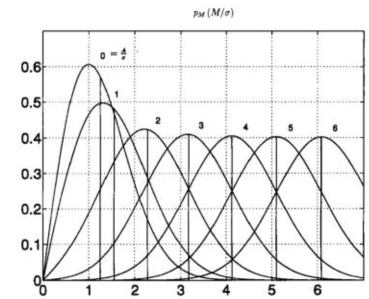


Image Noise



- Rician noise:
  (appears in MRI)

$$p(I) = \frac{I}{\sigma^2} \exp\left(\frac{-(I^2 + f^2)}{2\sigma^2} \right) I_0\left(\frac{If}{\sigma^2} \right)$$

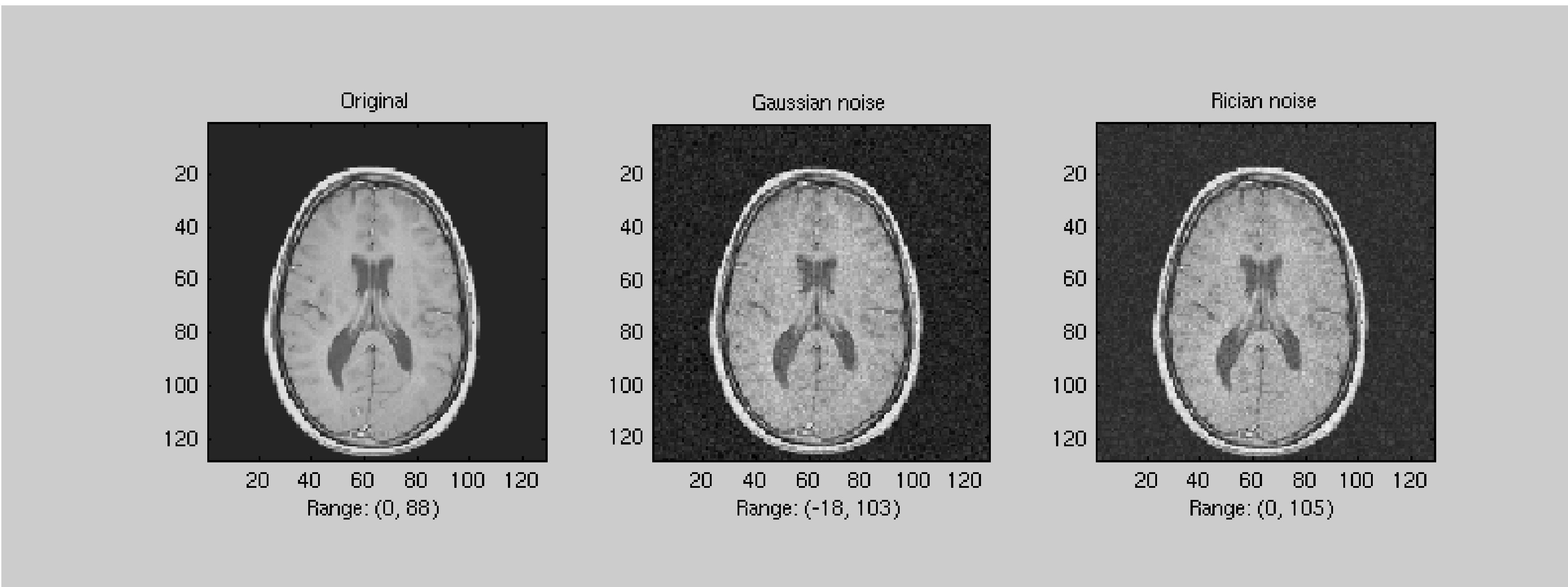
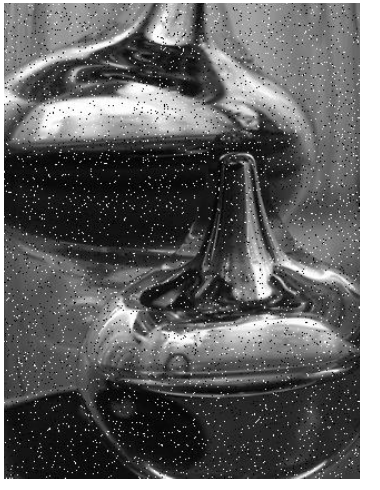


Image Noise

- Multiplicative noise:

$$I = f + fc$$

- Quantization errors
- Impulse "salt-and-pepper" noise



- The *signal to noise ratio (SNR) s* is an index of image quality

$$s = \frac{F}{\sigma}, \text{ where } F = \frac{1}{XY}\sum_{x=1}^{X}\sum_{y=1}^{Y} f(x,y)$$

Often used instead: *Peak Signal to Noise Ratio* (PSNR) $s_{peak} = \frac{F_{\max}}{\sigma}$

Colour Images

R

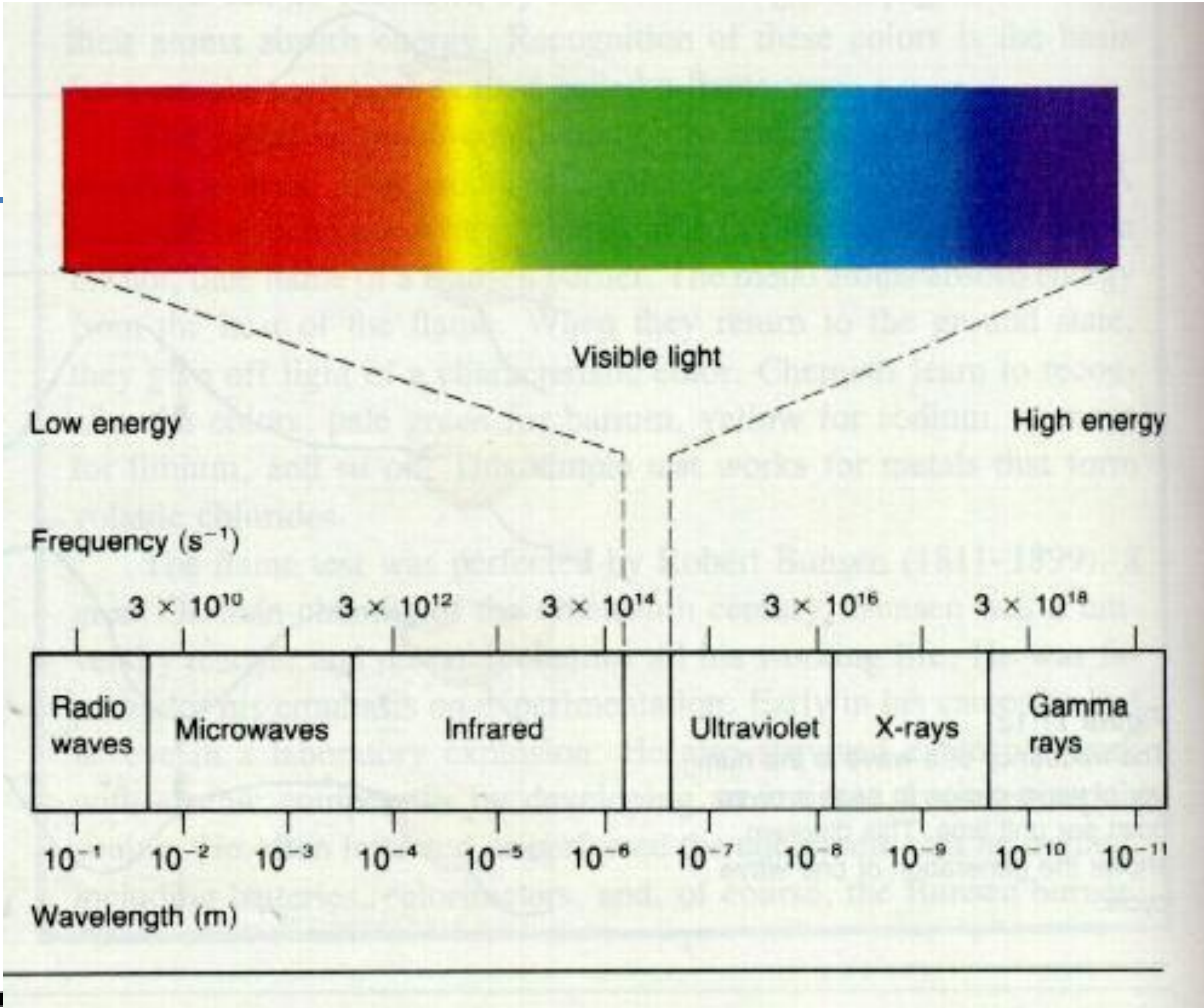
\+

G

\+

B

=


Visible light
Low energy
High energy
Frequency (s⁻¹)
3 × 10^10
3 × 10^12
3 × 10^14
3 × 10^16
3 × 10^18
Radio waves
Microwaves
Infrared
Ultraviolet
X-rays
Gamma rays
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-9
10^-10
10^-11
Wavelength (m)

Color cameras

We consider 3 concepts:

1. Prism (with 3 sensors)
2. Filter mosaic
3. Filter wheel

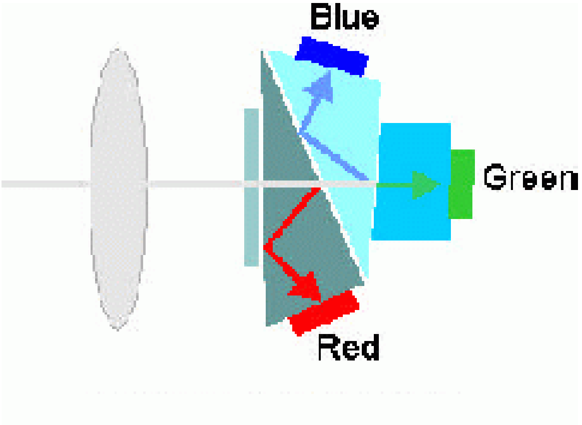
… and X3

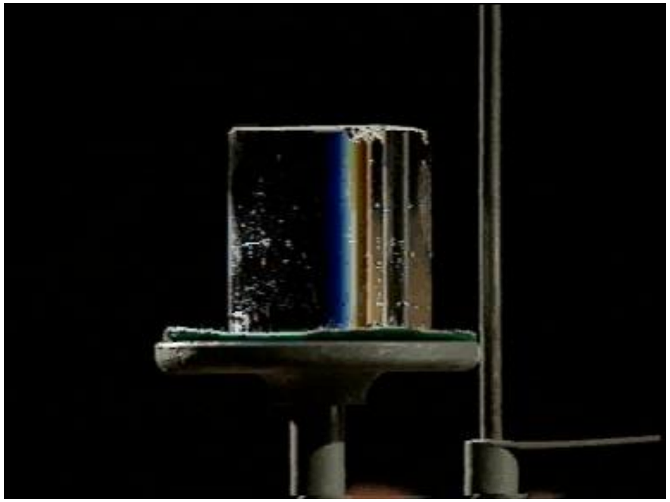
Prism color camera

Separate light in 3 beams using dichroic prism

Requires 3 sensors & precise alignment

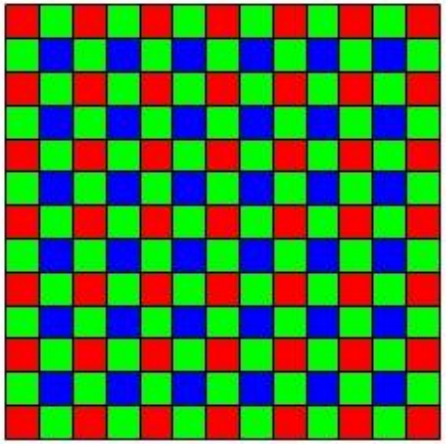
Good color separation



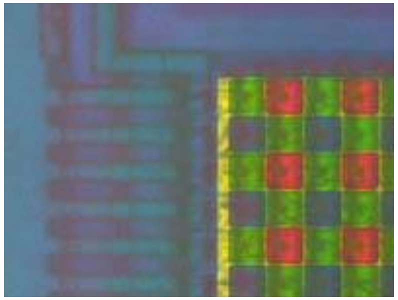


Filter mosaic

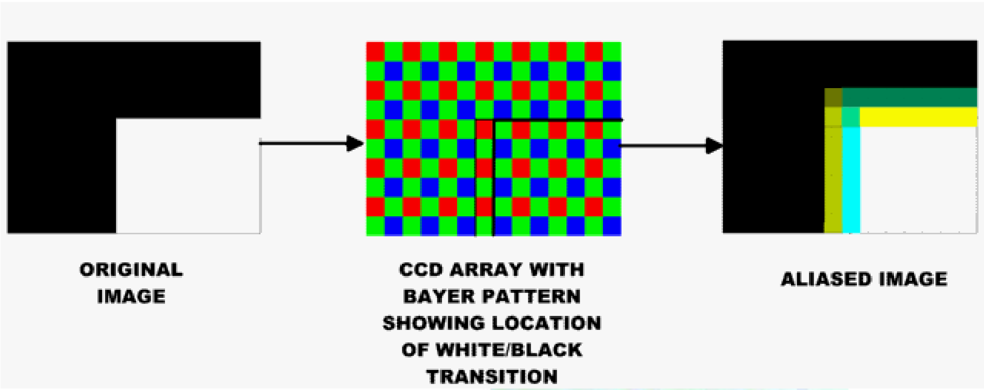
Coat filter directly on sensor



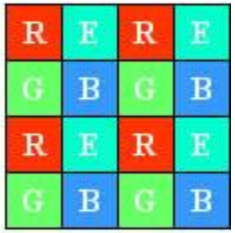
Bayer filter



Demosaicing (obtain full colour & full resolution image)



More colors:



Filter wheel

Rotate multiple filters in front of lens

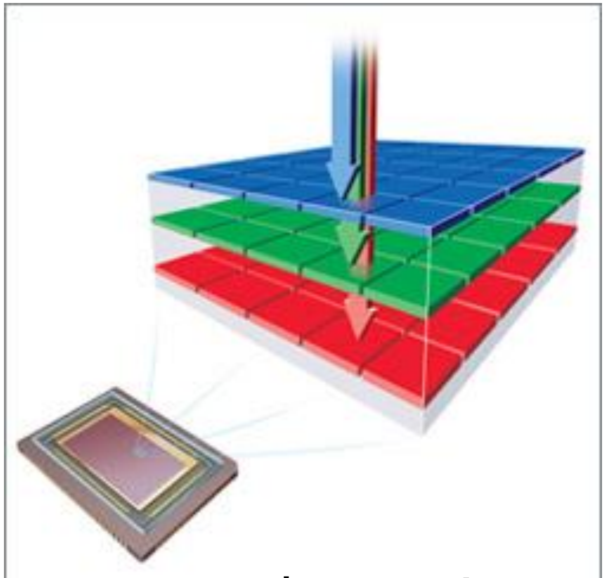
Allows more than 3 colour bands

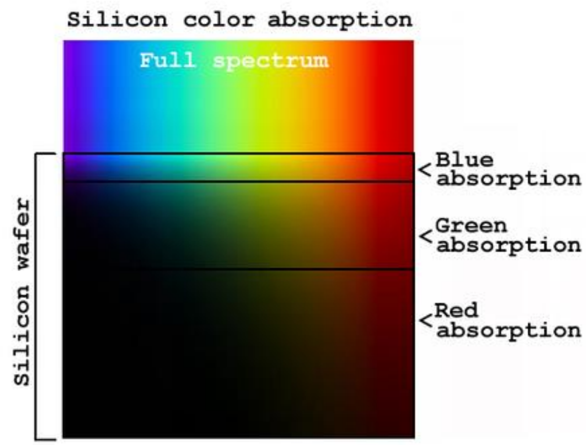
Only suitable for static scenes

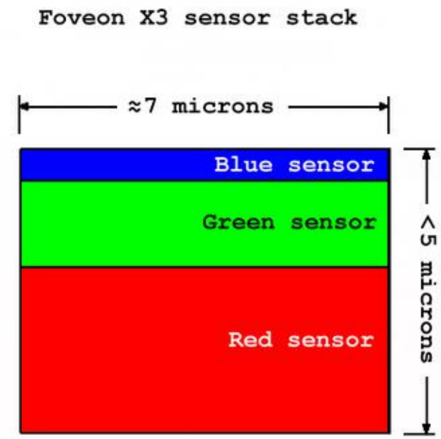
Prism vs. mosaic vs. wheel

approach	Prism	Mosaic	Wheel
# sensors	3	1	1
Separation	High	Average	Good
Cost	High	Low	Average
Framerate	High	High	Low
Artefacts	Low	Aliasing	Motion
Bands	3	3	3 or more
	High-end cameras	Low-end cameras	Scientific applications

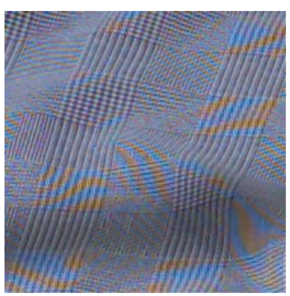
color CMOS sensor
Foveon's X3

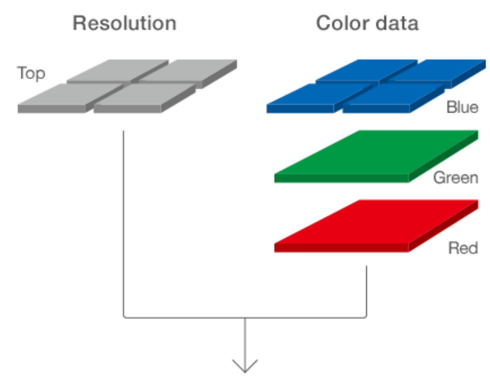






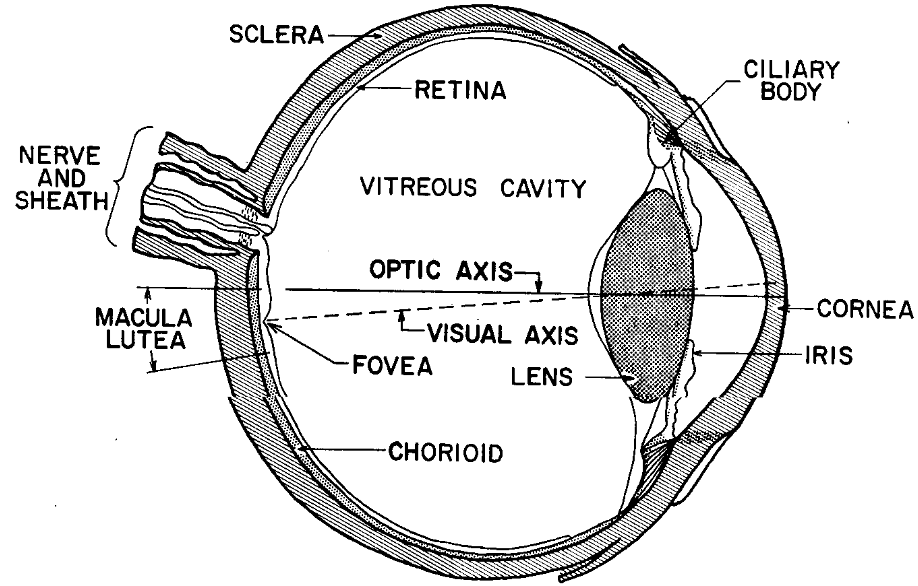
better image quality

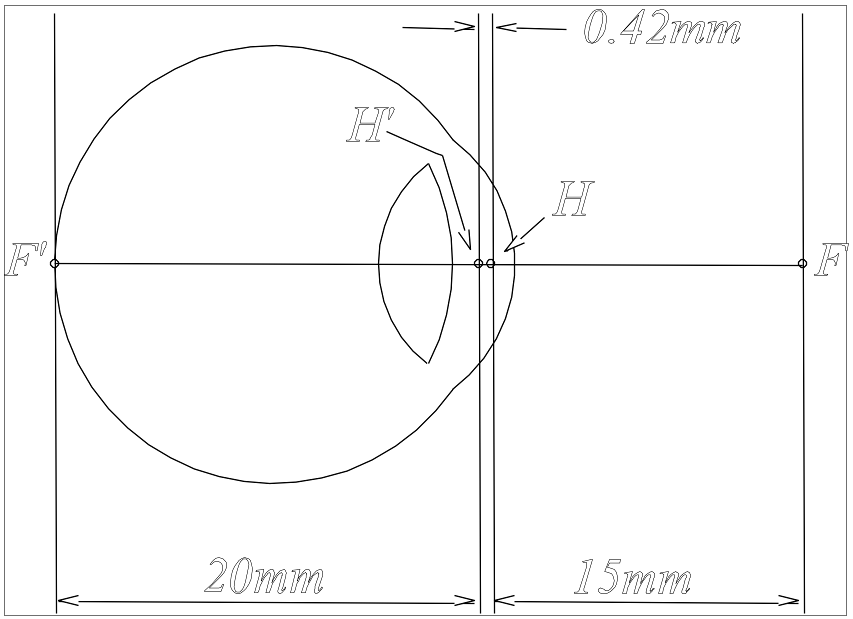




Reproduced by permission, the American Society of Photogrammetry and Remote Sensing. A.L. Nowicki, "Stereoscopy." Manual of Photogrammetry, Thompson, Radlinski, and Speert (eds.), third edition, 1966.

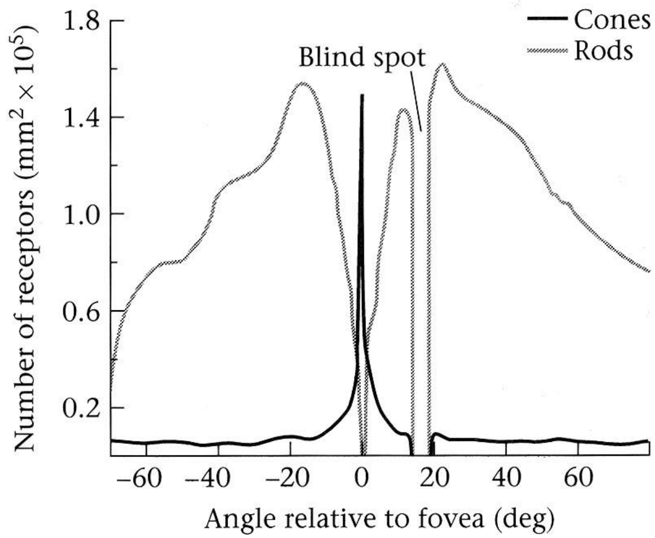
The Human Eye



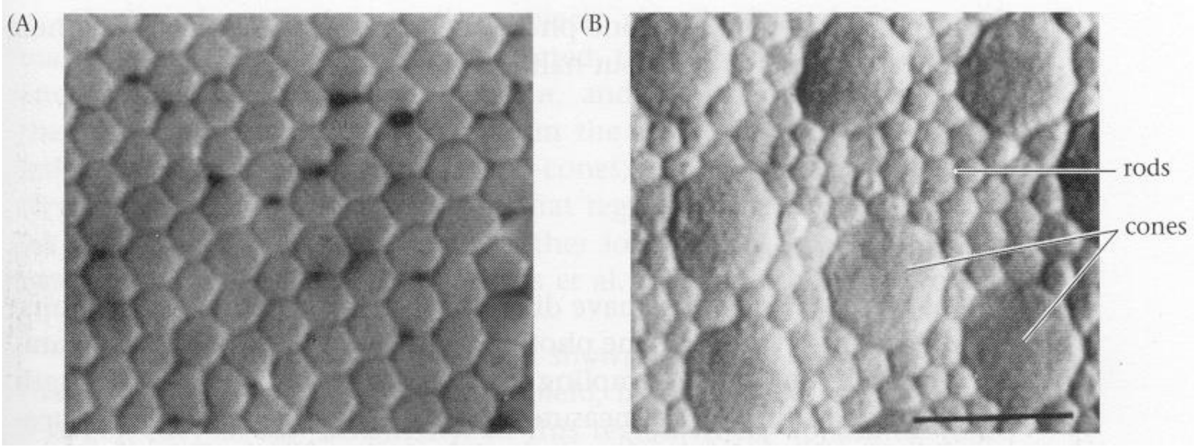


Helmoltz's Schematic Eye

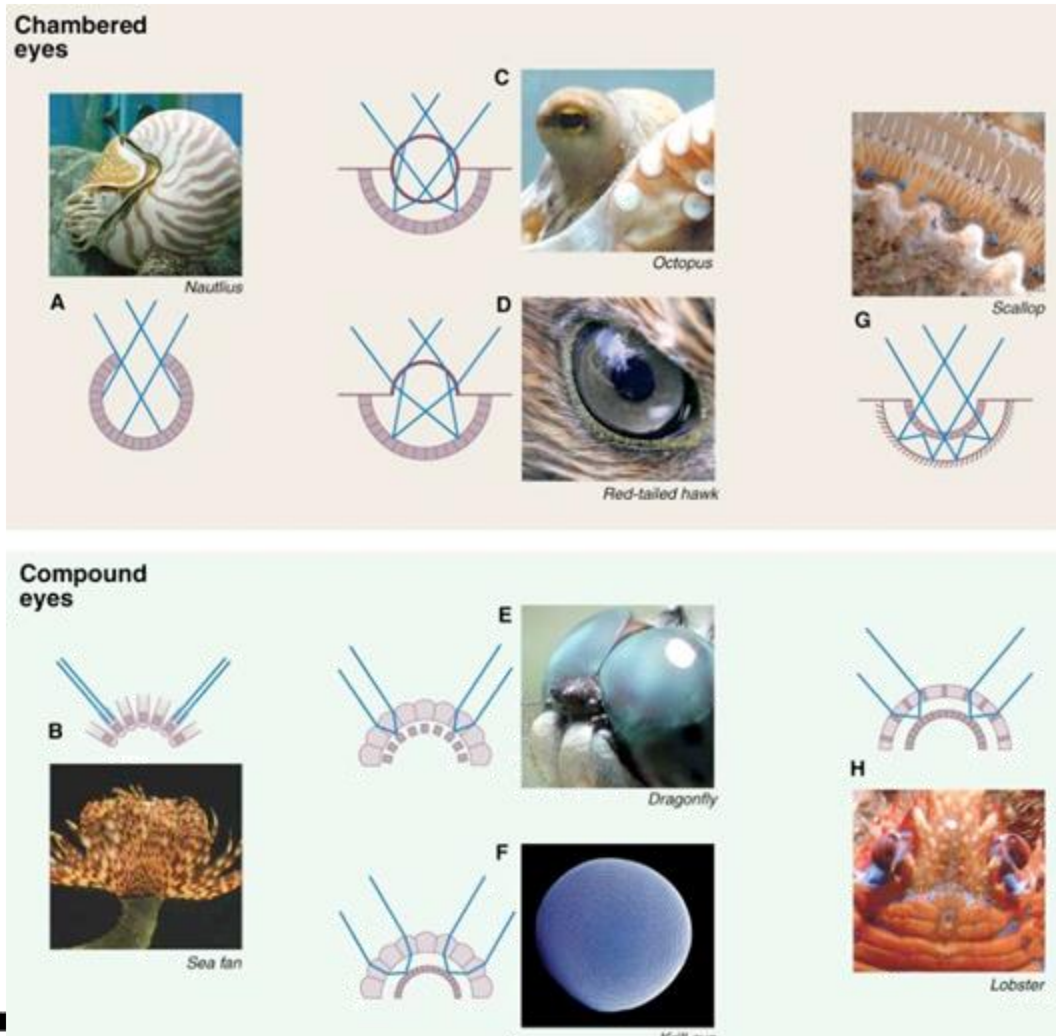
The distribution of rods and cones across the retina



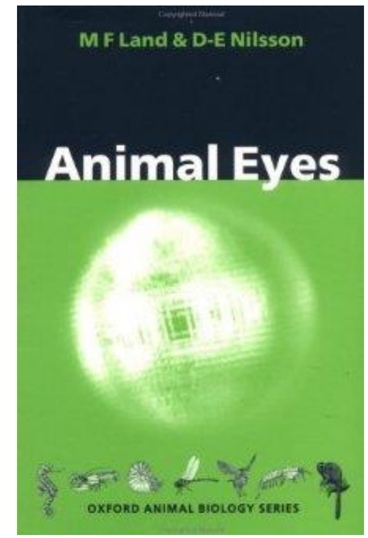
Cones in the fovea



More eyes in nature…



More info:



Fernald, R. D. 2006. Casting a Genetic Light on the Evolution of Eyes. Science 313, 1914-1918

Next week:

Image Segmentation