Measuring Light

Radiant Measurement
- Flux (W)
- Energy (J)
- Irradiance (W/m^2)
- Emittance (W/m^2)
- Intensity (W/sr)
- Radiance ($W/sr\ m^2$)

These are pure physical quantities.

Luminous Measurement
- Luminous flux (lumen; lm)
- Quantity of light (lm s)
- Illuminance (lux; lx)
- Luminous emittance (lm/m^2)
- Lum. intensity (candela; cd)
- Luminance (cd/m^2)

These relate to visual sensation.

Detectors

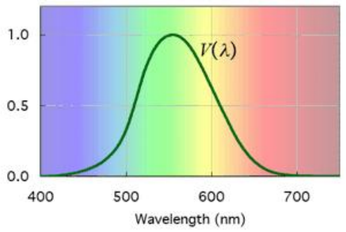
- Point
  - Photomultiplier (PM)
  - Avalanche photodiode (APD)
  - Channel photomultiplier (CPM)
  - …
- Image
  - Eye
  - CCD / CMOS
  - ICCD (i.e. MCP + CCD)
  - EMCCD
  - …

Flux

- Radiant flux (W) measures the energy flowing at a point per unit time.
- Luminous flux (lm) weights the flux for its impact on a visual system.
  - Peak efficiency constant K_m = 683 lm/W
  - Spectral luminous efficiency V(l)

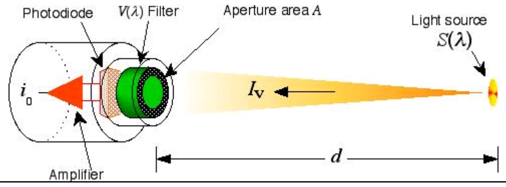
$$\Phi_e = \frac{dQ_e}{dt}$$

$$\Phi_v = K_m \int_{380\text{nm}}^{830\text{nm}} \Phi_e V(\lambda) d\lambda$$



Intensity

- Intensity relates to the flux from a point source.
  - Flux per unit solid angle
  - Definition of the candela
- Intensity is calibrated by a current from a known source with a known response *s*.

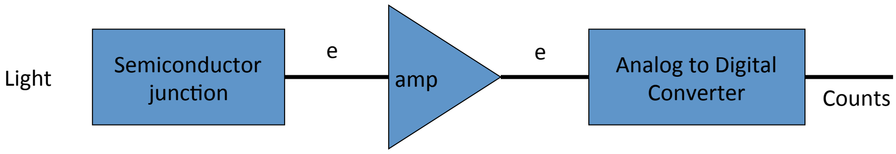


$$I_e = \frac{d\Phi_e}{d\Omega} \qquad I_v = \frac{d\Phi_v}{d\Omega}$$

$$s_v = \frac{A \int S(\lambda) s(\lambda) d\lambda}{K_m \int S(\lambda) V(\lambda) d\lambda}$$

$$I_v = \frac{i_0}{s_v} \frac{d^2}{\Omega_0}$$

Light → Electrons → Counts

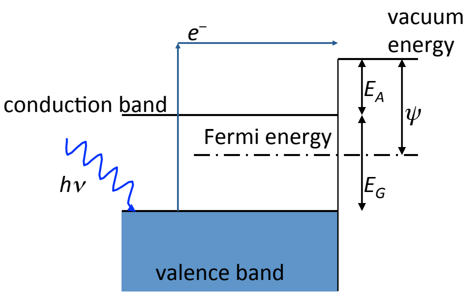


Voltage between
Anode & Cathode

Gain
Offset

2^{Bits} = levels
- 8 = 256
- 12 = 4096
- 14 = 16384
- 16 = 65536

Photoelectron emission



- Counting photons requires conversion to electrons.
- The photoelectric effect can eject electrons from a material into a vacuum.
  - Exceed gap energy E_G and electron affinity energy E_A
  - Compare to work function ψ

Quantum Efficiency

- There is a probability that a photon will produce a free electron.
  – Depends on bulk material properties
  – Depends on atomic properties

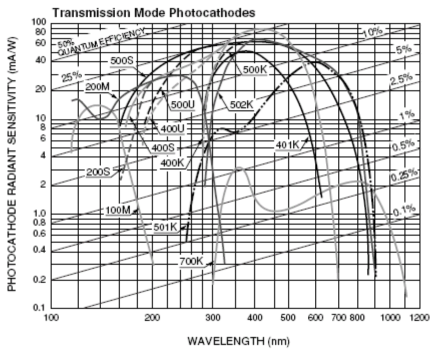
- This is expressed as the quantum efficiency $\eta(\nu)$.

– Reflection coefficient *R*
– Photon absorption *k*
– Mean *e* escape length *L*
– Probability to eject from surface P_s
– Probability to reach vacuum energy P_v

$$\eta(\nu) = (1 - R)\frac{P_s P_v}{k + 1/L}$$

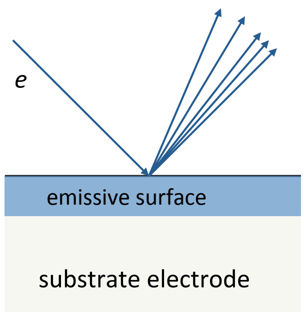
Commercial Photocathodes

- Different photocathodes vary in response to frequency and in quantum efficiency.
  – Alkali for UV detection (Cs-I, Cs-Te)
  – Bialkali for visible light (Sb-Rb-Cs, Sb-K-Cs)
  – Semiconductors for visible to IR (GaAsP, InGaAs)



Hamamatsu.com

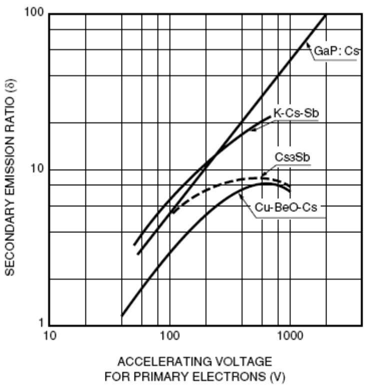
Electron multiplier



- Single photoelectrons would produce little current.
- Electrons can be multiplied by interaction with a surface.
  - Emitter: BeO, GaP
  - Metal substrate: Ni, Fe, Cu
- This electrode is called a dynode.

Multiplication Factor

- Dynodes need good electron multiplication.
  - Emission material
  - accelerating potential for the incident electron
- Dynodes typically operate around 100 V.
  - Factor d of 2 to 6



Hamamatsu.com

Gain

- Dynode gain δ depends on the material and potential E.
  - *k* typically 0.7 to 0.8
- Multiple dynodes are staged to increase gain.
  - Photocathode current I_{d0}
  - Input stage current I_{dn}
- Total gain is a product of stage gain.
  - Collection efficiency α

$$\delta = aE^k$$

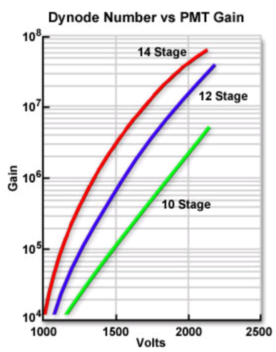
$$I_{dn} = \delta_n I_{d(n-1)}$$

$$I_{out} = I_{d0}\alpha\delta_1\delta_2\ldots\delta_n$$

$$\mu = \frac{I_{out}}{I_{d0}} = \alpha\delta_1\delta_2\ldots\delta_n$$

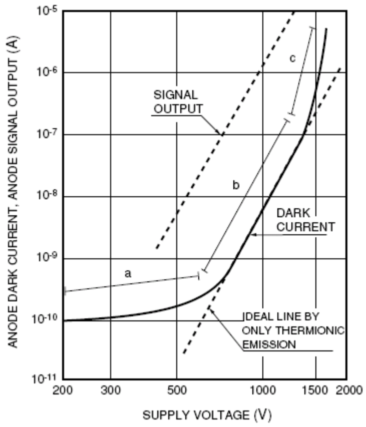
$$\mu \cong \alpha\left(a\left(\frac{V}{n+1}\right)^k\right)^n = AV^{kn}$$

Amplification



- Photomultiplier tubes often have 10 to 14 stages.
  - Gain in excess of 10^7
- A single photon can produce a measurable charge.
  - Single photoelectron
  - $Q_{pe} \sim 10^{-12}$ C
- Fast response in about 1 ns.
  - $I_{pe} \sim 1$ mA

Dark Current



- Phototubes have "dark" current even with no incident light.
  - Thermionic emission
  - Anode leakage
  - Case scintillation
  - Gas ionization
- This increases with applied voltage.

Noise

- Dark current contributes to the noise in a measurement.
  - Equivalent noise input
  - For Δf = 1 Hz, P ~ 10^{-15} W
- Signal to noise depends on the statistical fluctuations, dark current and readout circuit.
  - Dominated by statistics

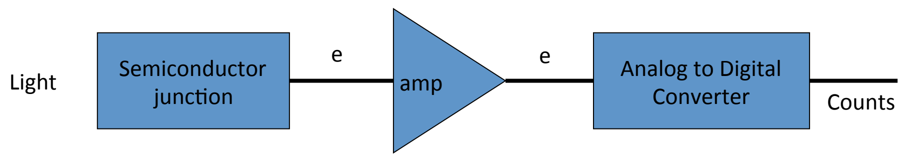
$$P_{ENI} = \frac{(2eI_D \mu \Delta f)^{1/2}}{S}$$

$$S/N = \frac{I_{out}}{i_{out}}$$

$$S/N = \sqrt{\frac{I_0 \alpha}{2e\Delta f} \frac{1}{1 + 1/\delta_1 + \ldots + 1/\delta_1 \ldots \delta_n}}$$

$$S/N \cong I_0 \sqrt{\frac{1}{2e\Delta f (I_0 + 2I_D)} \frac{1}{\delta(\delta - 1)}}$$

Light → Electrons → Counts



Voltage between
Anode & Cathode

Gain
Offset

2^{Bits} = levels
- 8 = 256
- 12 = 4096
- 14 = 16384
- 16 = 65536

Detector

Ideal
- ✓ Detect all photons
- ✓ Perfect reading
- ✓ Give 0 counts if dark
- ✓ Fast

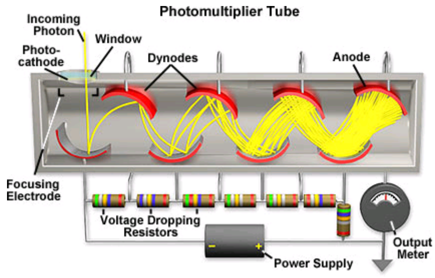
Real
- ✗ Quantum efficiency < 1
- ✗ Readout noise
- ✗ Dark electrons
- ✗ Limited bandwidth

Even in the case of a perfect detector **count noise** is present

$$SNR = \frac{N}{\sqrt{N}} = \sqrt{N}$$

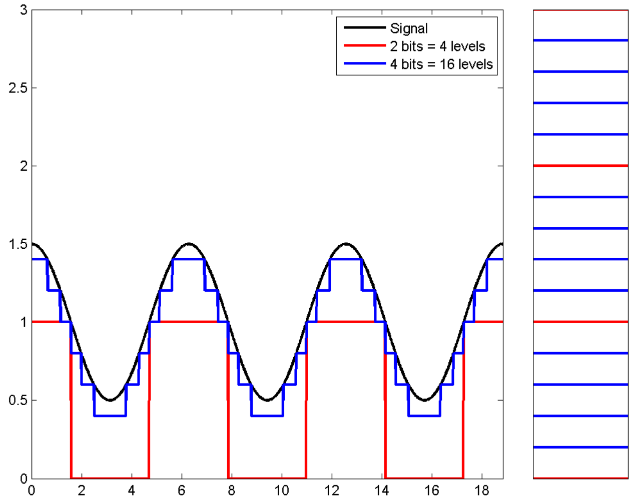
Photomultiplier Tube

- A photomultiplier tube (phototube, PMT) combines a photocathode and series of dynodes.
- The high voltage is divided between the dynodes.
- Output current is measured at the anode.
  - Sometimes at the last dynode

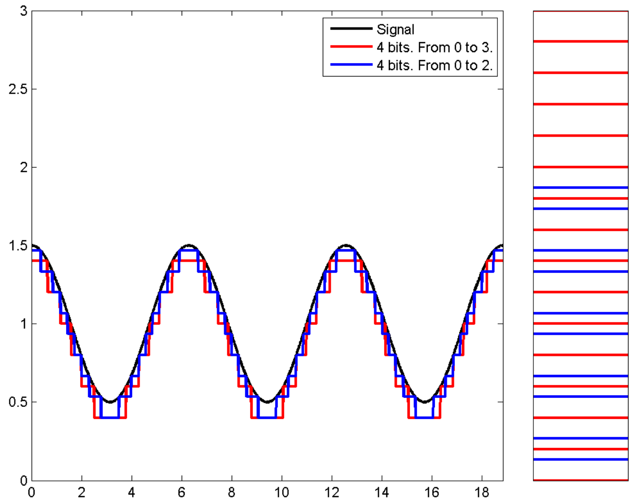


Photomultiplier Tube

Digitalization

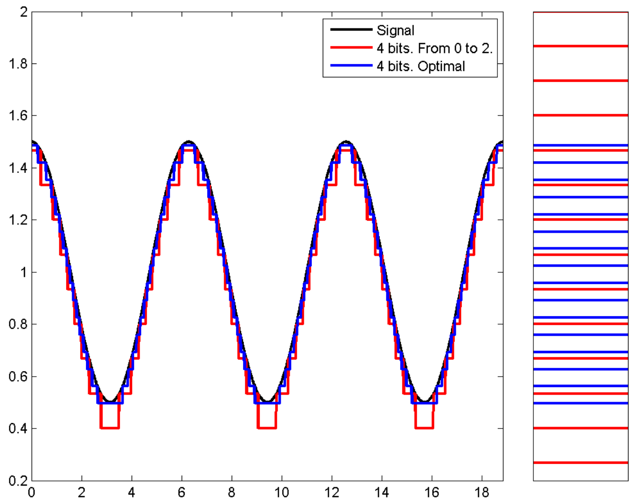


Digitalization



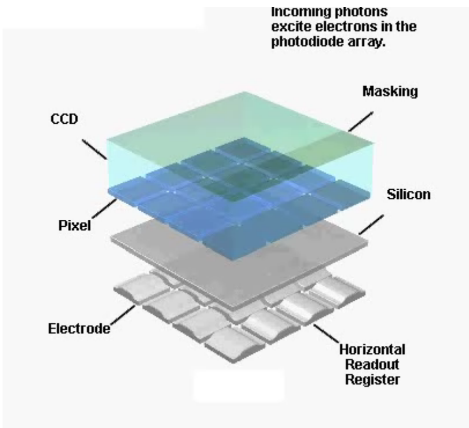
Effect of Adjusting Gain

Digitalization



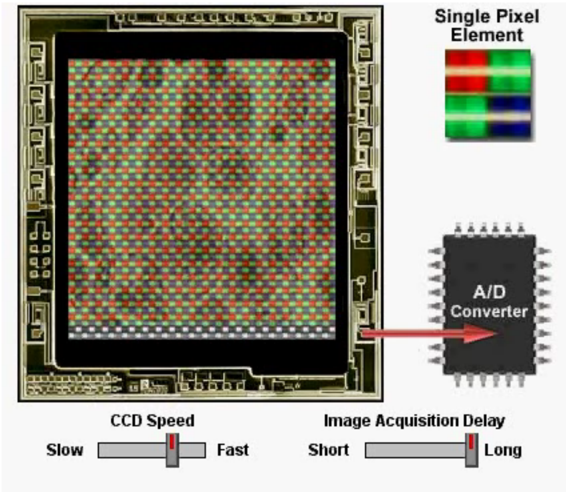
Optimal Gain and Offset

Reading a CCD



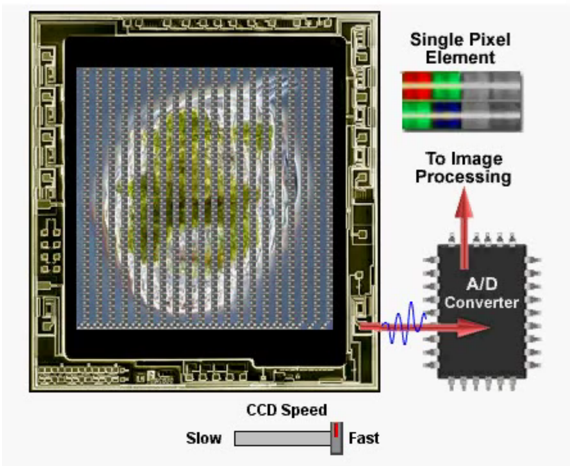
Pixels are read one at a time
Clock ~ 10 MHz
Pixels can be grouped (binning)
Q_e from .1 to .9

Reading a full frame CCD



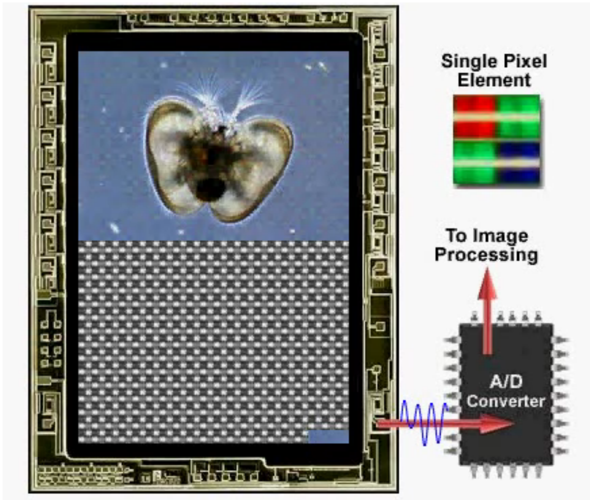
- ✗ Dead time between frames

Reading an interline CCD



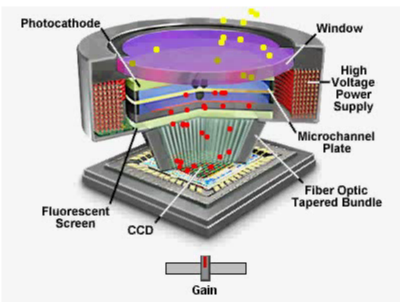
- ✓ No dead times
- ✗ Half resolution

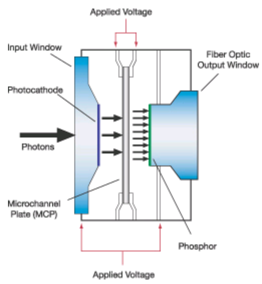
Reading a frame transfer CCD



- ✓ No dead times
- ✗ Expensive

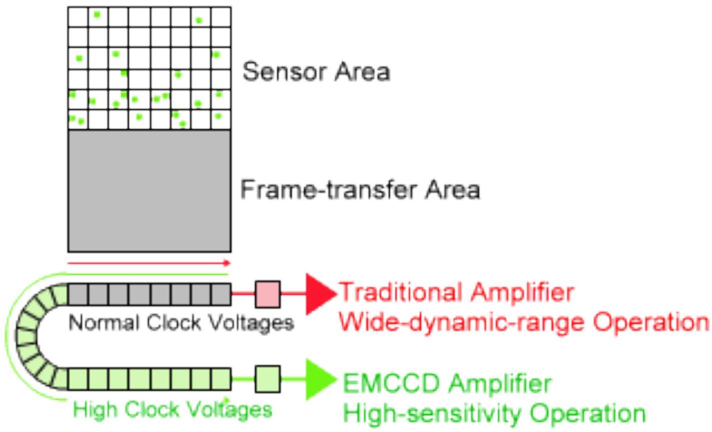
Reading an Intensified CCD





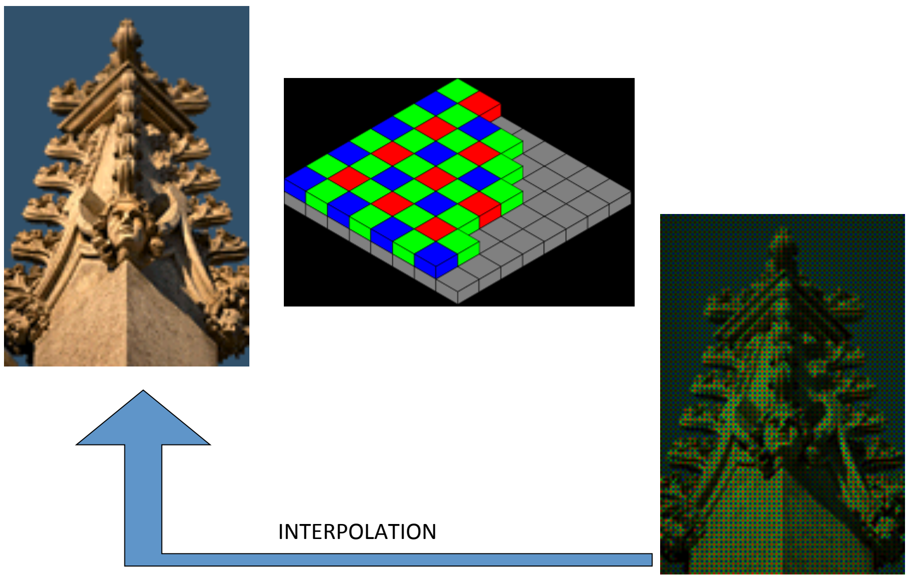
- ✓ MCP can be modulated (FLIM)
- ✗ Less resolution (due to the intensifier)
- ✗ Shape of the MCP appears in the image

Reading an Electron Multiplying CCD



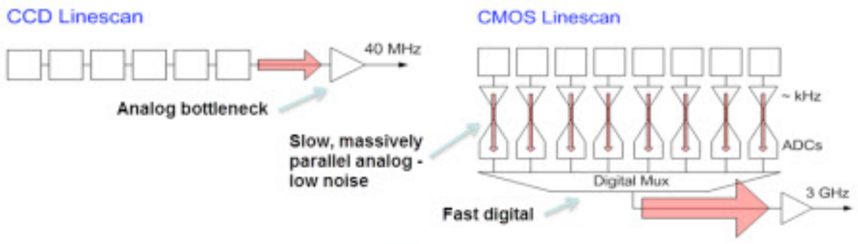
- ✓ Single molecule detection
- ✗ Very expensive

Reading a color CCD

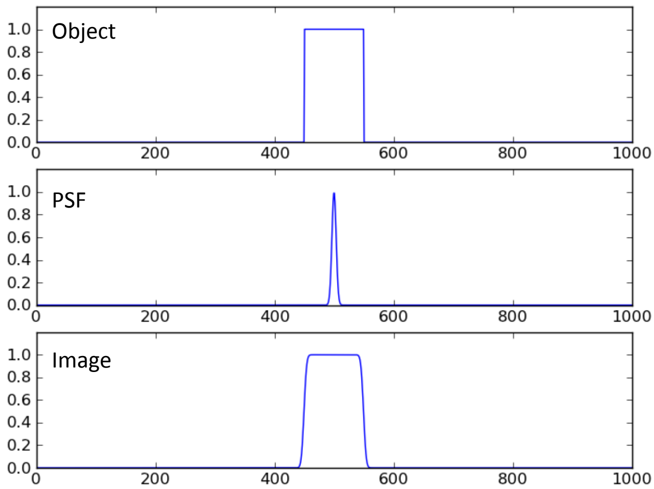


http://www.cambridgeincolour.com/tutorials/sensors.htm

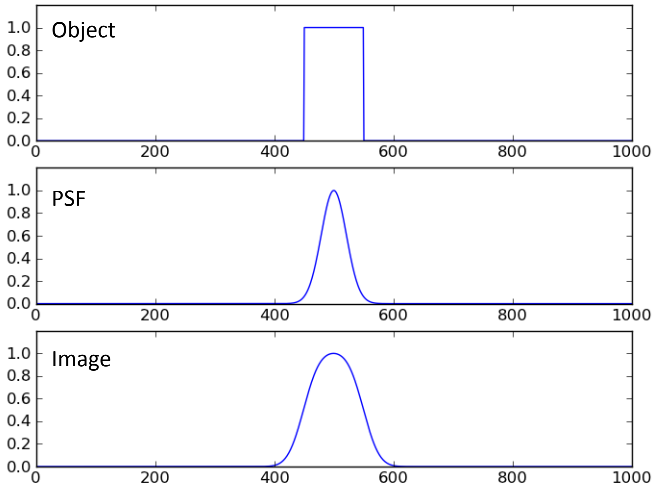
Reading a CMOS



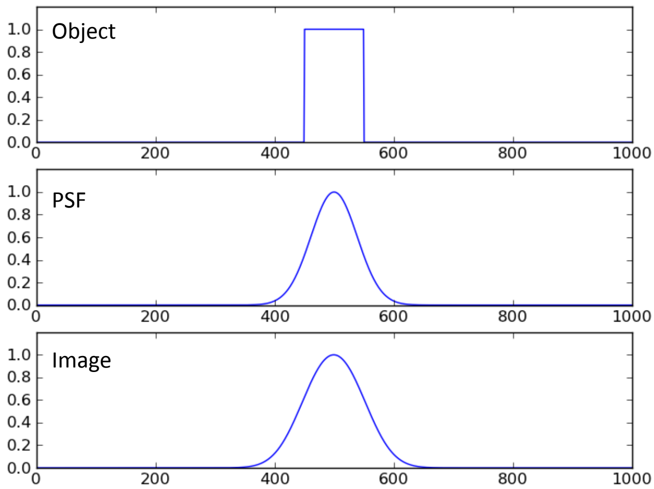
Sampling



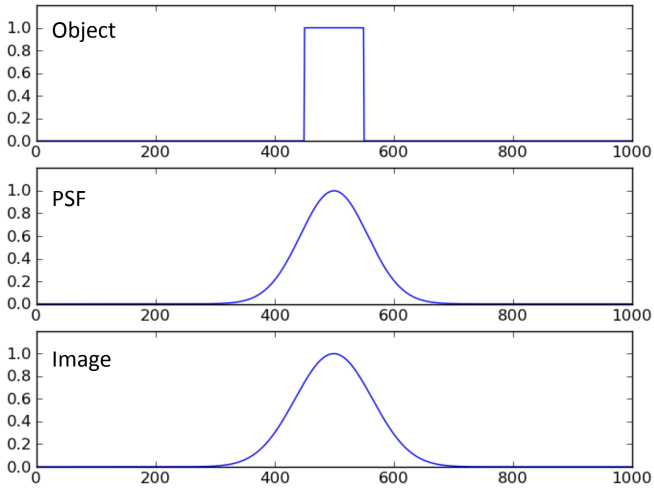
Sampling



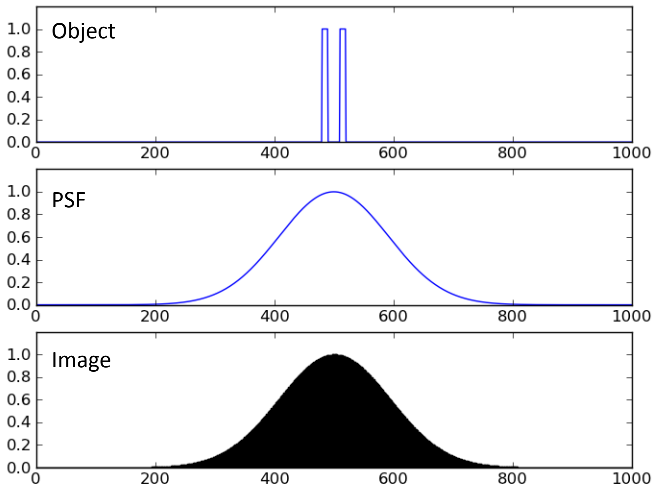
Sampling



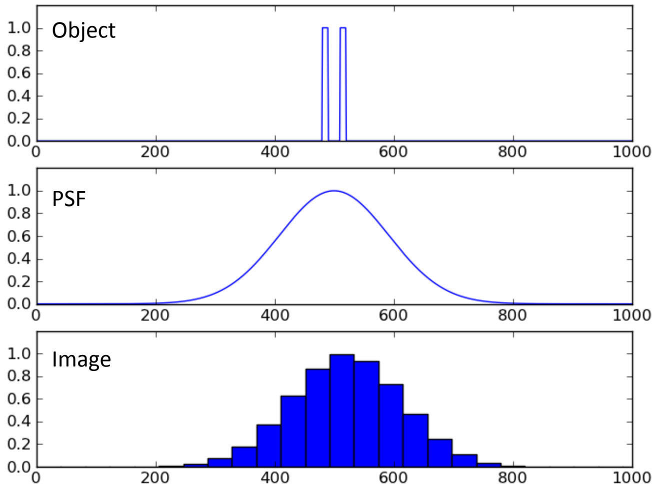
Sampling: PSF Size



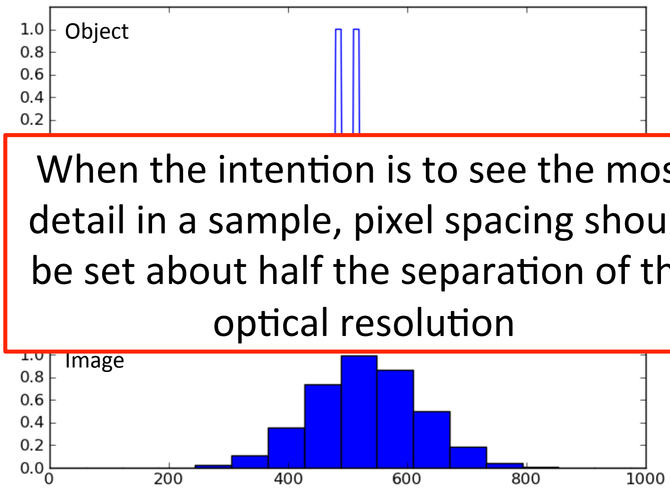
Sampling: Step Size



Sampling: PSF Size



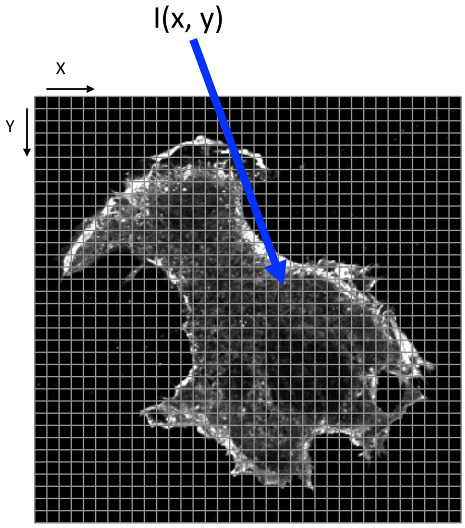
Sampling: Step size



When the intention is to see the most detail in a sample, pixel spacing should be set about half the separation of the optical resolution

Sampling

Basics of Image Processing



Integer values:

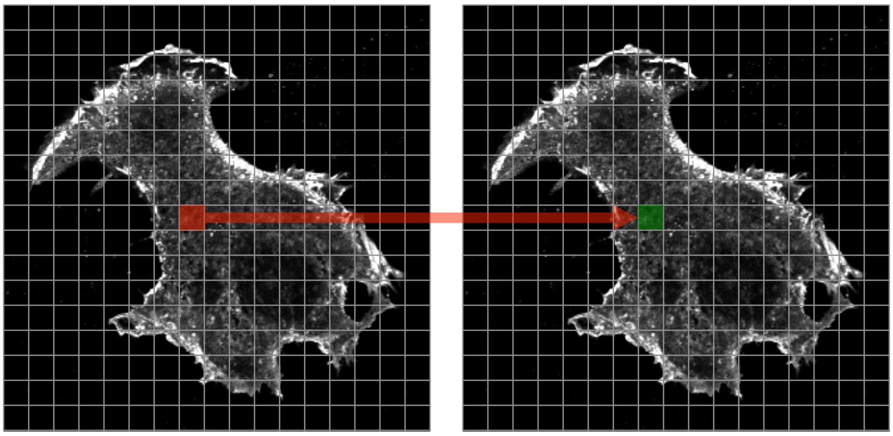
- 8-bit: range: 0 – 255
- 16-bit: range: 0 - 65535
- Signed 16-bit: range: −32768 to 32767

Floating point values:

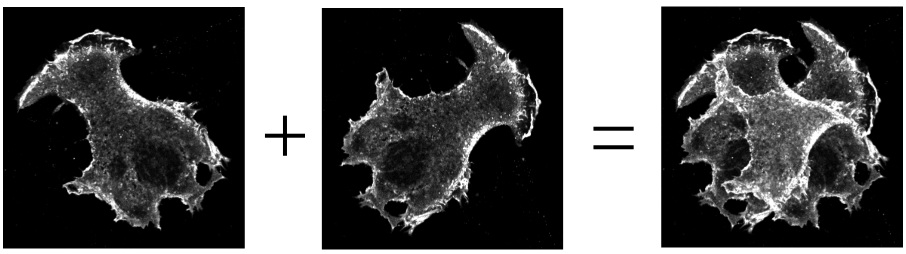
- 32-bit:
  range ≈ $-10^{37} - 10^{37}$, precision ≈ 10^{-7}
- 64-bit:
  range: $-10^{308} - 10^{308}$, precision ≈ 10^{-16}
- Special values:
  - NaN (Not a Number, 0/0 = NaN)
  - -Inf, +Inf (+/- infinity)

Image Processing Operations

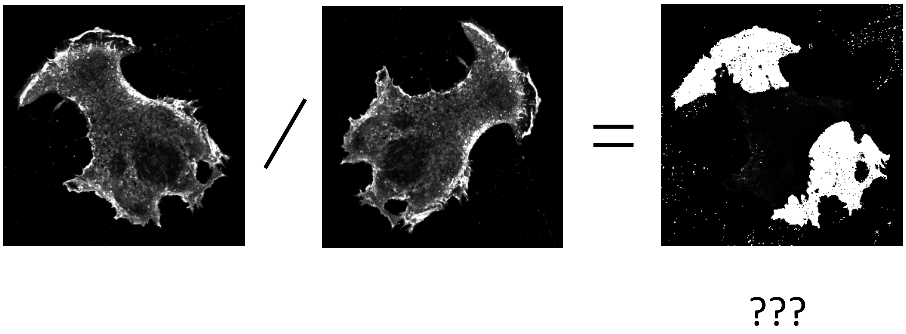
Point operations: each result pixel depends on one input pixel



Point operations: Image arithmetic



Point operations: Image arithmetic



Point operations: Masking

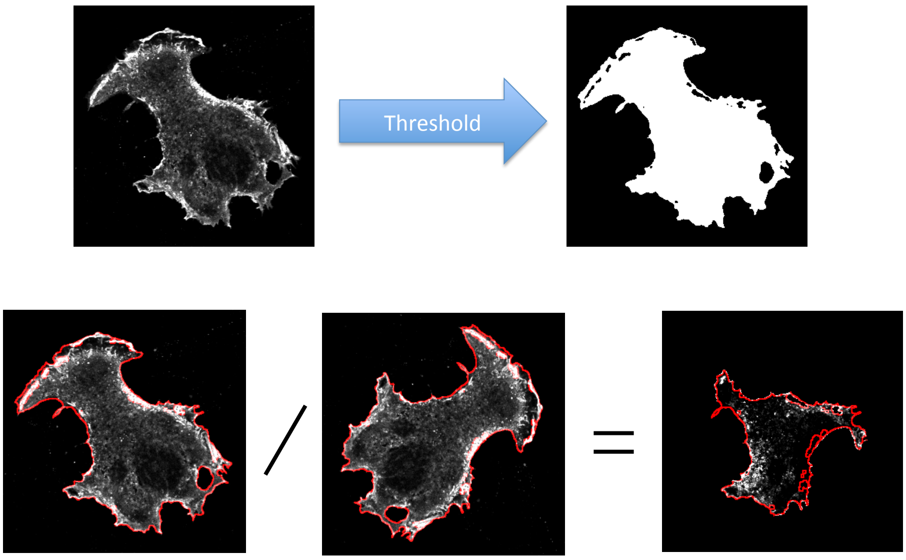


Image processing operations

Local neighborhood operations: each result pixel depends on several input pixels

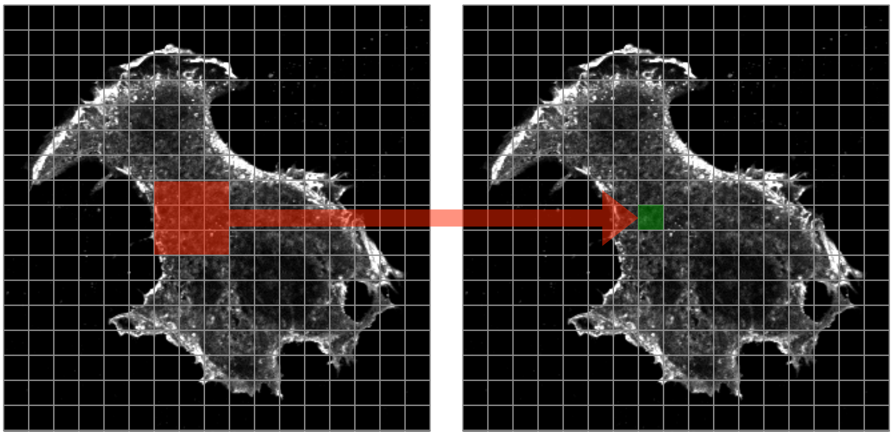


Image processing operations

Local neighborhood operations: mean in a 3 × 3 neighborhood

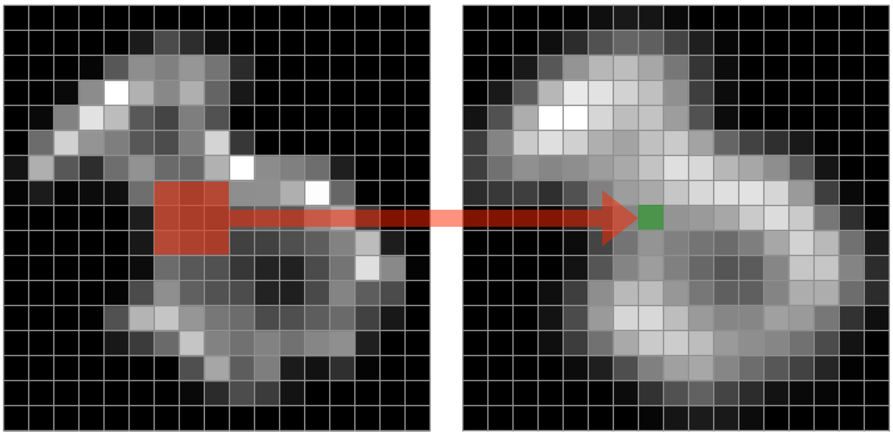
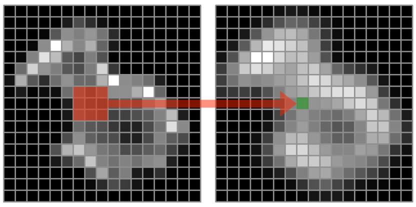


Image processing operations

Local neighborhood operations: *convolution*



1. Multiply the input pixels from a small area around the input pixel, with weights given by the *kernel*:

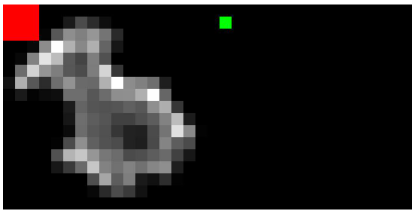
50	52	55
44	56	49
37	56	48

×

1	2	1
2	4	2
1	2	1

=

50	104	55
88	224	98
37	112	48



2. Add the weighted input pixels and normalize with the sum of the kernel values:

$\sum$

50	104	55
88	224	98
37	112	48

/ 16 = 51

3. Repeat by sliding the kernel over all input pixels.

Image processing operations

$K_1 =$

1	2	1
2	4	2
1	2	1

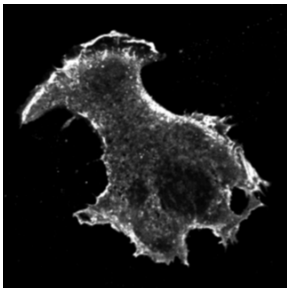
$K_2 =$

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

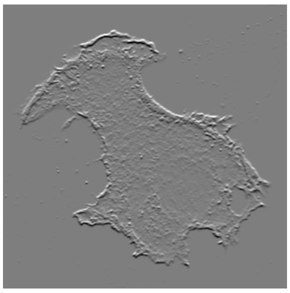
$K_3 =$

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

$I * K_1$



$I * K_2$



$\sqrt{[I * K_2]^2 + [I * K_3]}$

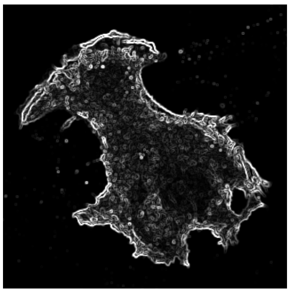
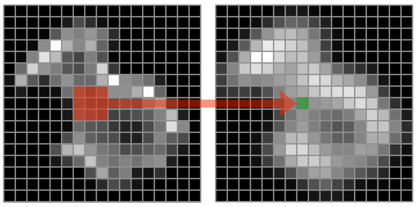
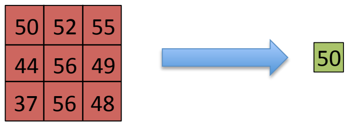


Image processing operations

Non-linear local neighborhood operations: *median filter*



Replace each pixel with the median value of the pixels in a small region around it



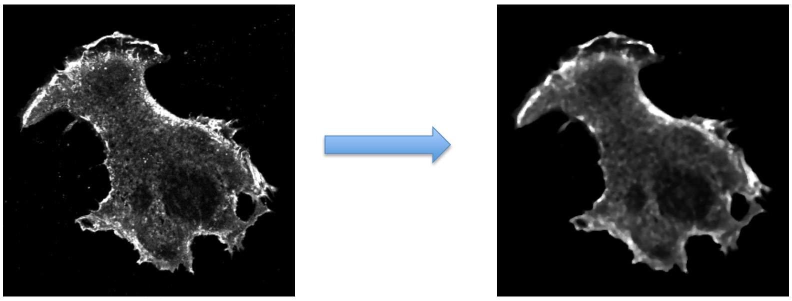


Image processing operations

Global operations: each result pixel depends on all input pixels

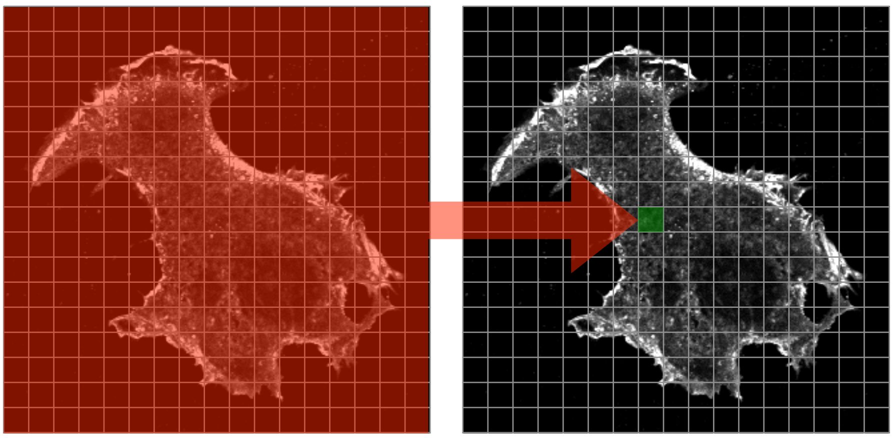
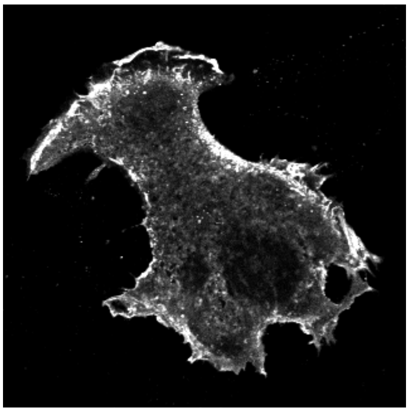
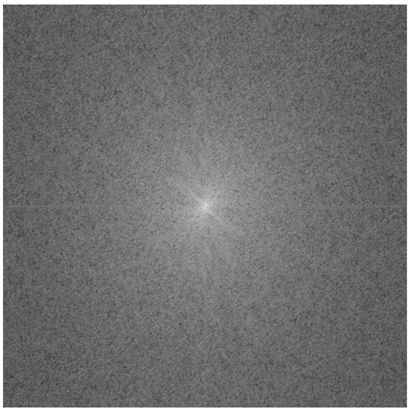


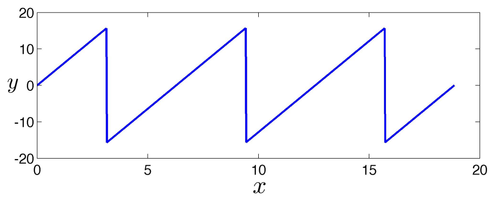
Image processing operations

Global operations: *Fourier Transform*



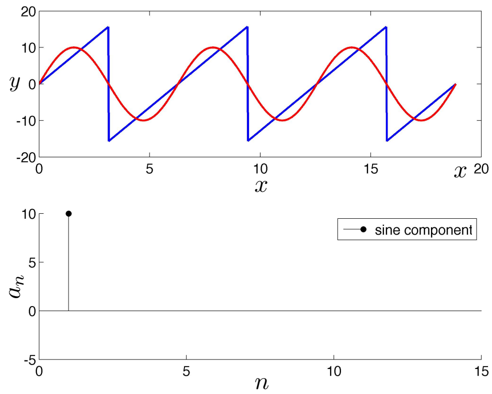


The Fourier Transform



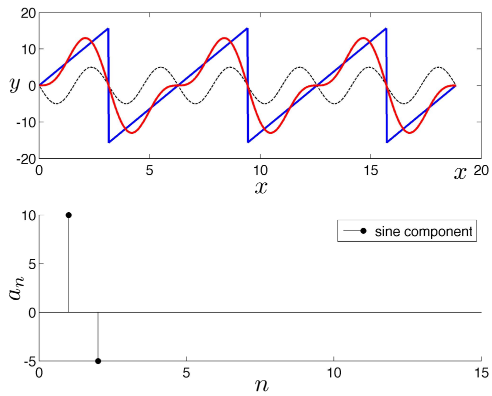
Can we approximate this function by a sum of simpler functions?

The Fourier Transform



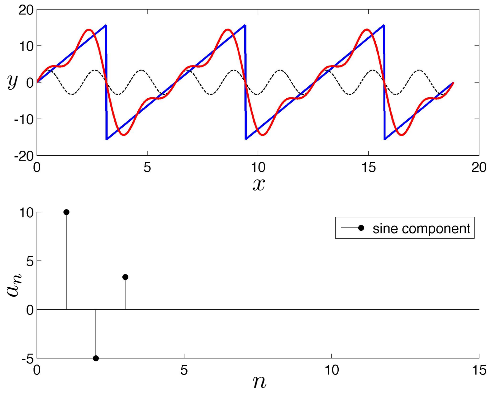
$$y = a_1 \sin(x)$$

The Fourier Transform



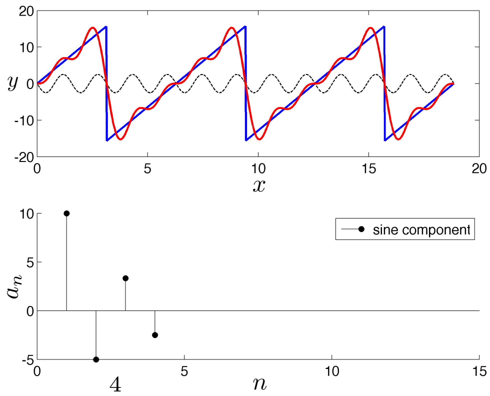
$$y = a_1 \sin(x) + a_2 \sin(2x)$$

The Fourier Transform



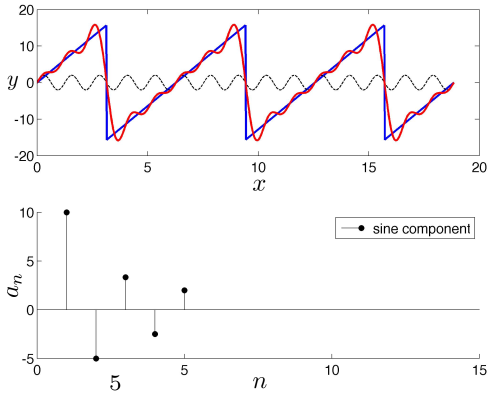
$$y = a_1 \sin(x) + a_2 \sin(2x) + a_3 \sin(3x)$$

The Fourier Transform



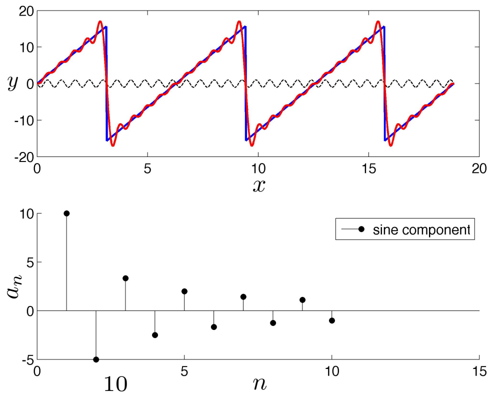
$$y = \sum_{n=1}^{4} a_n \sin(nx)$$

The Fourier Transform



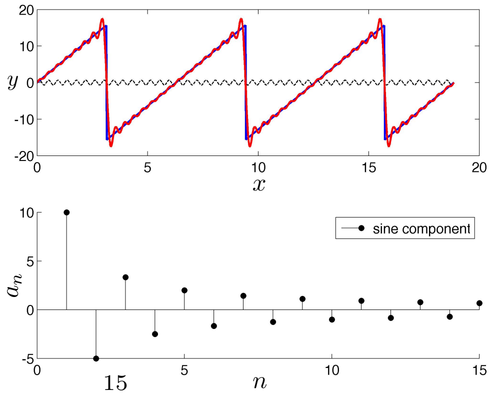
$$y = \sum_{n=1}^{5} a_n \sin(nx)$$

The Fourier Transform



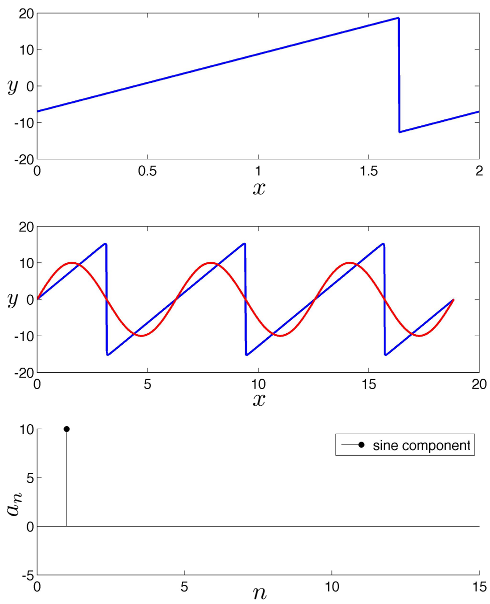
$$y = \sum_{n=1}^{10} a_n \sin(nx)$$

The Fourier Transform



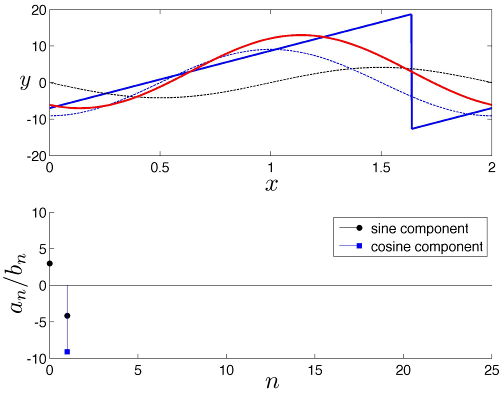
$$y = \sum_{n=1}^{15} a_n \sin(nx)$$

The Fourier Transform



In general, simple sine functions are not enough...

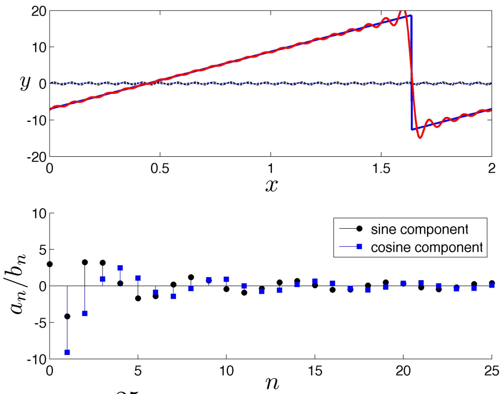
The Fourier Transform



You need a sum of scaled sine and cosine functions and a constant offset!

$$y = a_0 + a_1 \cos\left(\frac{2\pi}{T}x\right) + b_1 \sin\left(\frac{2\pi}{T}x\right)$$

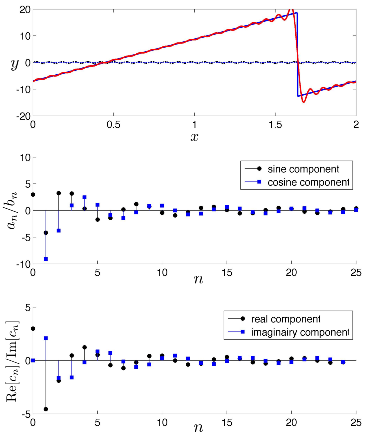
The Fourier Transform



$$y = \sum_{n=0}^{25} a_n \cos\left(\frac{2\pi n x}{T}\right) + b_n \sin\left(\frac{2\pi n x}{T}\right)$$

The Fourier Transform

$$y = \sum_{n=0}^{\infty} a_n \cos\left(\frac{2\pi n x}{T}\right) + b_n \sin\left(\frac{2\pi n x}{T}\right)$$



Usually this is written in a more compact form using complex numbers:

$$y = \sum_{n=-\infty}^{\infty} c_n\left[\cos\left(\frac{2\pi n x}{T}\right) + i \sin\left(\frac{2\pi n x}{T}\right)\right]$$

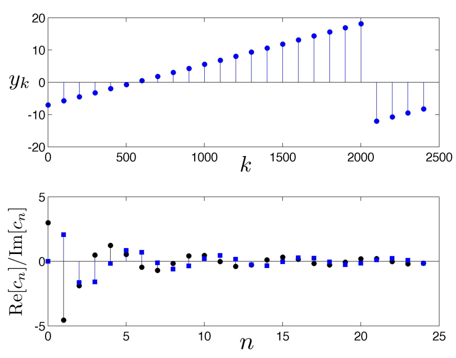
$$= \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi n x}{T}}$$

$$c_n = \begin{cases} \frac{1}{2}(a_n - i b_n) & n > 0 \\ a_0 & n = 0 \\ \frac{1}{2}(a_{-n} + i b_{-n}) & n < 0 \end{cases}$$

The Discrete Fourier Transform

$$y_k = \frac{1}{N} \sum_{n=0}^{N-1} c_n e^{i\frac{2\pi n k}{N}}$$

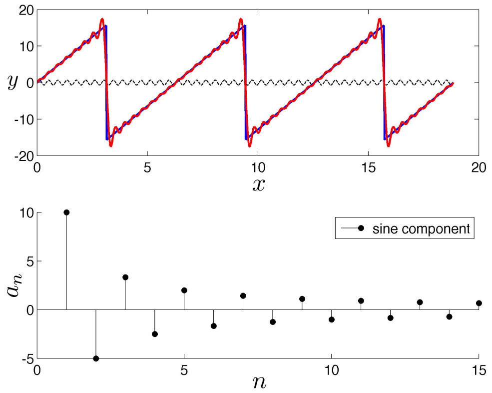
Inverse DFT



DFT

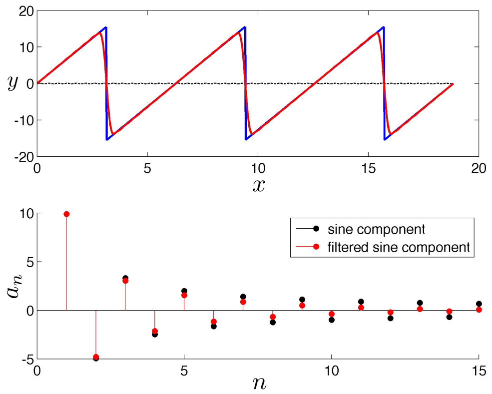
$$c_n = \sum_{k=0}^{N-1} y_k e^{-i\frac{2\pi n k}{N}}$$

What is this good for?



$$y = \sum_{n=1}^{15} a_n \sin(nx)$$

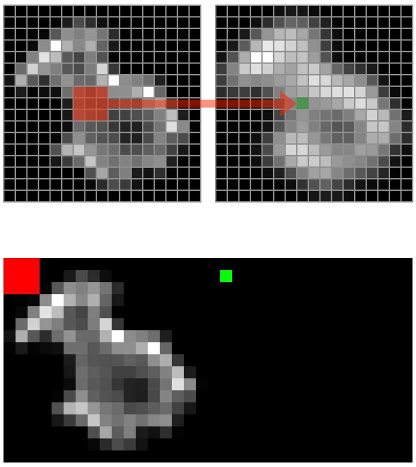
Fourier domain filtering



$$y = \sum_{n=1}^{15} a_n \sin(nx)$$

Remember convolution?

Local neighborhood operations: *convolution*



1. Multiply the input pixels from a small area around the input pixel, with weights given by the *kernel*:

50	52	55
44	56	49
37	56	48

×

1	2	1
2	4	2
1	2	1

=

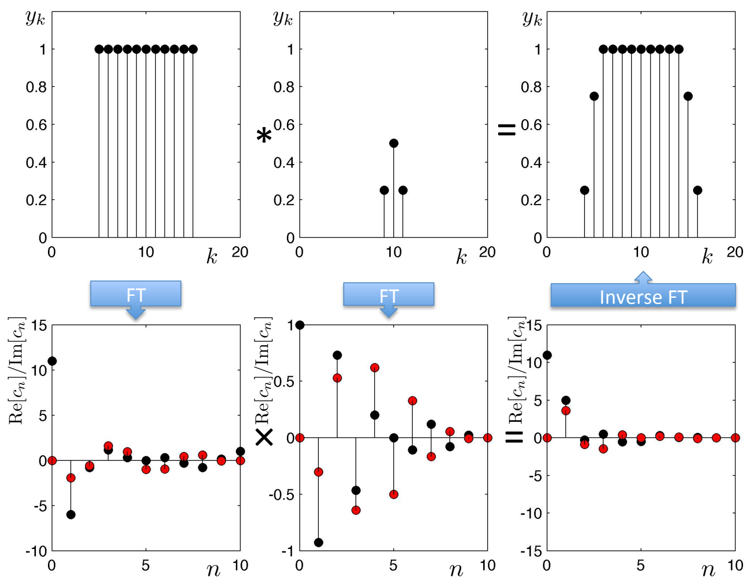
50	104	55
88	224	98
37	112	48

2. Add the weighted input pixels and normalize with the sum of the kernel values:

$$\sum \begin{bmatrix} 50 & 104 & 55 \\ 88 & 224 & 98 \\ 37 & 112 & 48 \end{bmatrix} \Big/ 16 = 51$$

3. Repeat by sliding the kernel over all input pixels.

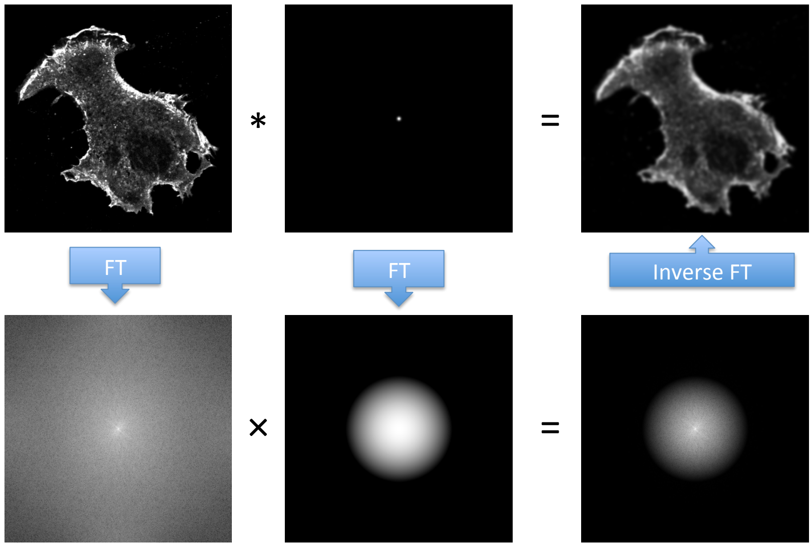
Convolution in the Fourier domain



Convolution in the spatial domain is equivalent to multiplication in the Fourier domain

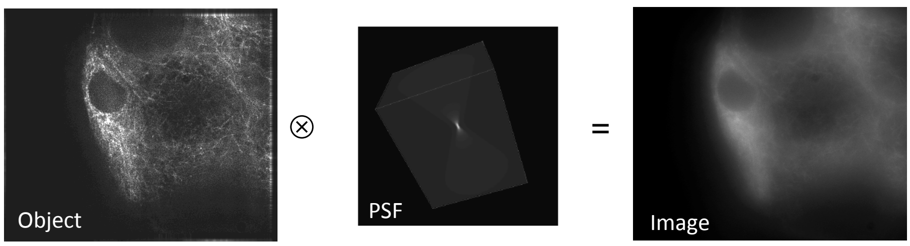
Image processing operations

Smoothing in the Fourier domain

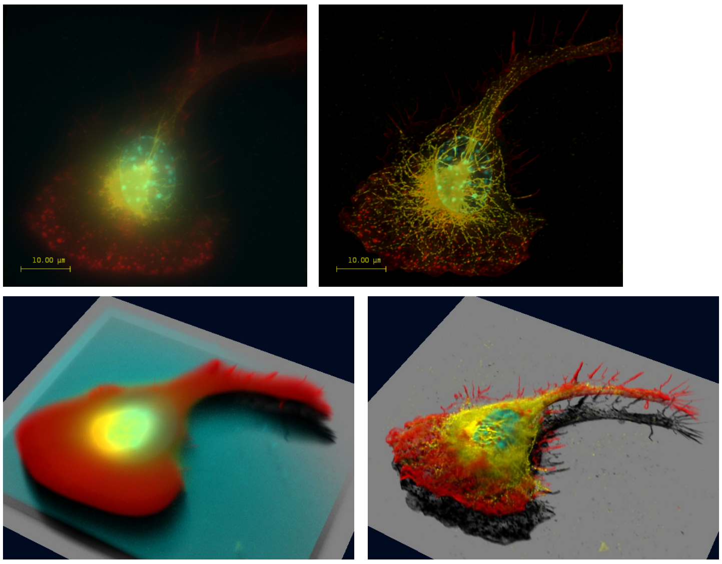


Deconvolution

The imaging system replaces each point in the original 3D object by an Image of the PSF (*convolution*), blurring the resulting image:



Deconvolution mathematically reverts that process:



Macrophage fluorescently stained for tubulin (yellow/green), actin (red) and the nucleus (DAPI, blue)

http://www.svi.nl/EvansMacrophage