

**BIO-INSPIRED
MATERIALS**

NATIONAL CENTER OF COMPETENCE
IN RESEARCH

Scientific image processing session 1: Basics

Dimitri Vanhecke

March 2026

**UNI
FR**

UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG



adolphe merkle institute
excellence in pure and applied nanoscience

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE



**UNIVERSITÉ
DE GENÈVE**



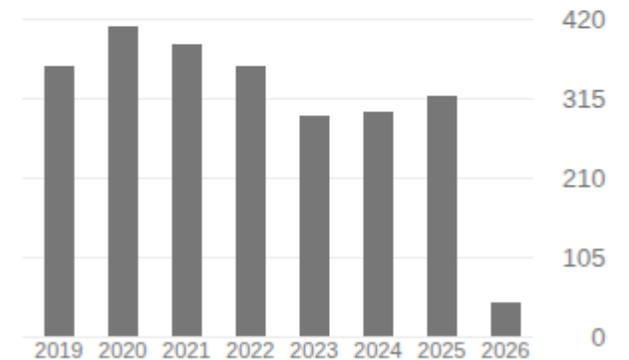
SWISS NATIONAL SCIENCE FOUNDATION

Administration

Who am I?

- PhD in structural biology (University of Bern)
- PostDoc in 3D image quantification
- 2 years at Max Planck Institute of Biochemistry, Munich, Germany
- At AMI since 2012
- Author or co-author on 90+ scientific papers
 - > 4000 citations
 - 📄 h-index: 32
 - 📄 i10-index: 48
- 25 years of experience with scientific images (2D, 3D and 4D: LSM, SEM, TEM, FIB, etc...)
- 20 years of experience with imageJ and Fiji

	All	Since 2021
Citations	4244	1700
h-index	32	25
i10-index	48	39



Administration

How far can
go with FIJI

use key
ImageJ
options
correctly

I want to
know
precisely
what I am
doing

Exploit the
capacity of
ImageJ/FIJI

To improve
own data /
pictures

Determine
the size of
objects and
size
distribution

Distribution
size of my
nano-
particles

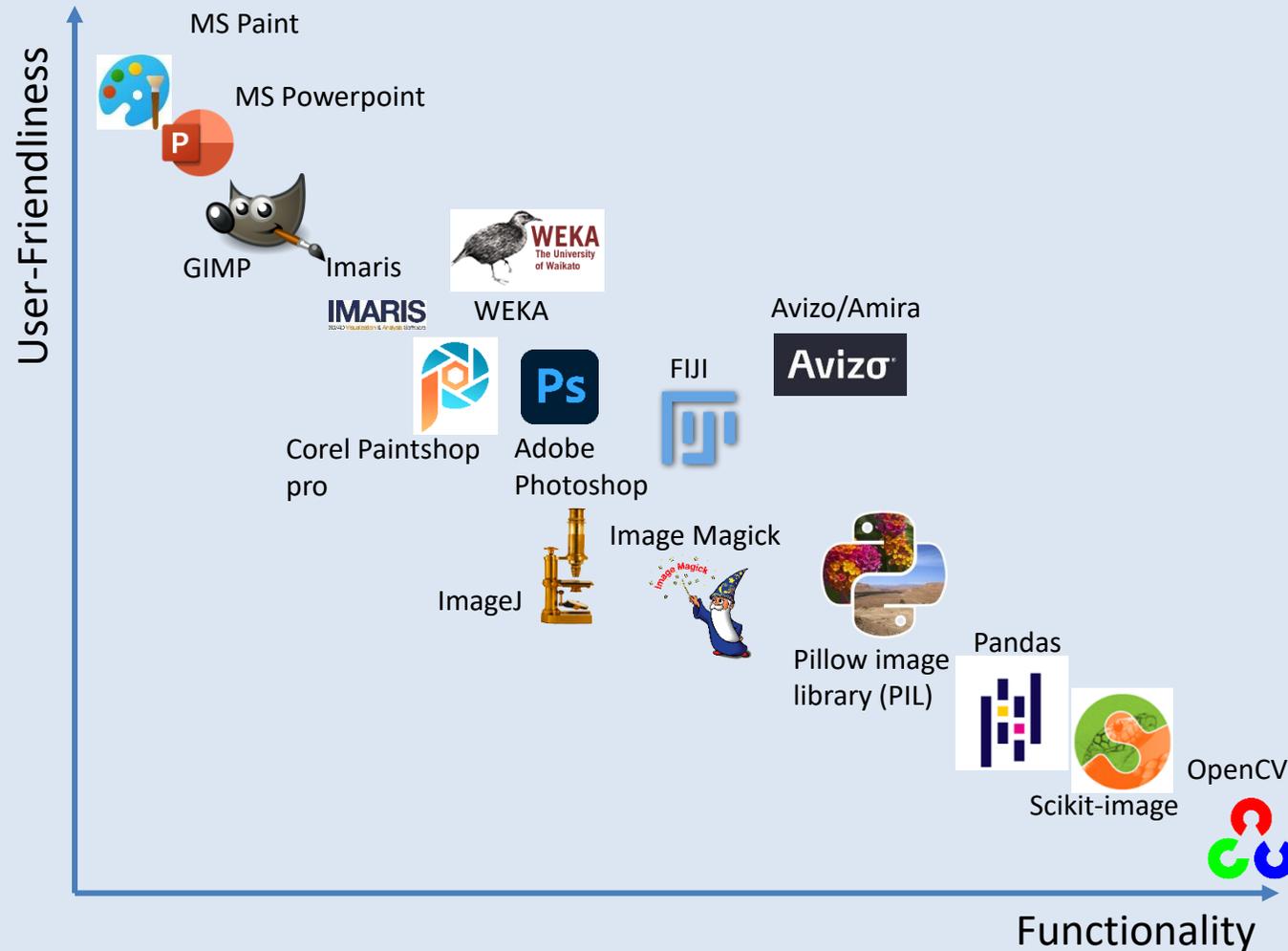
Learning
how
scripting
works

(Nano)-
particle size
distribution
and size
analysis

Count cell
number in
a more
objective
way

Administration

Why ImageJ (in fact: FIJI)?



FIJI (FIJI is just imageJ)

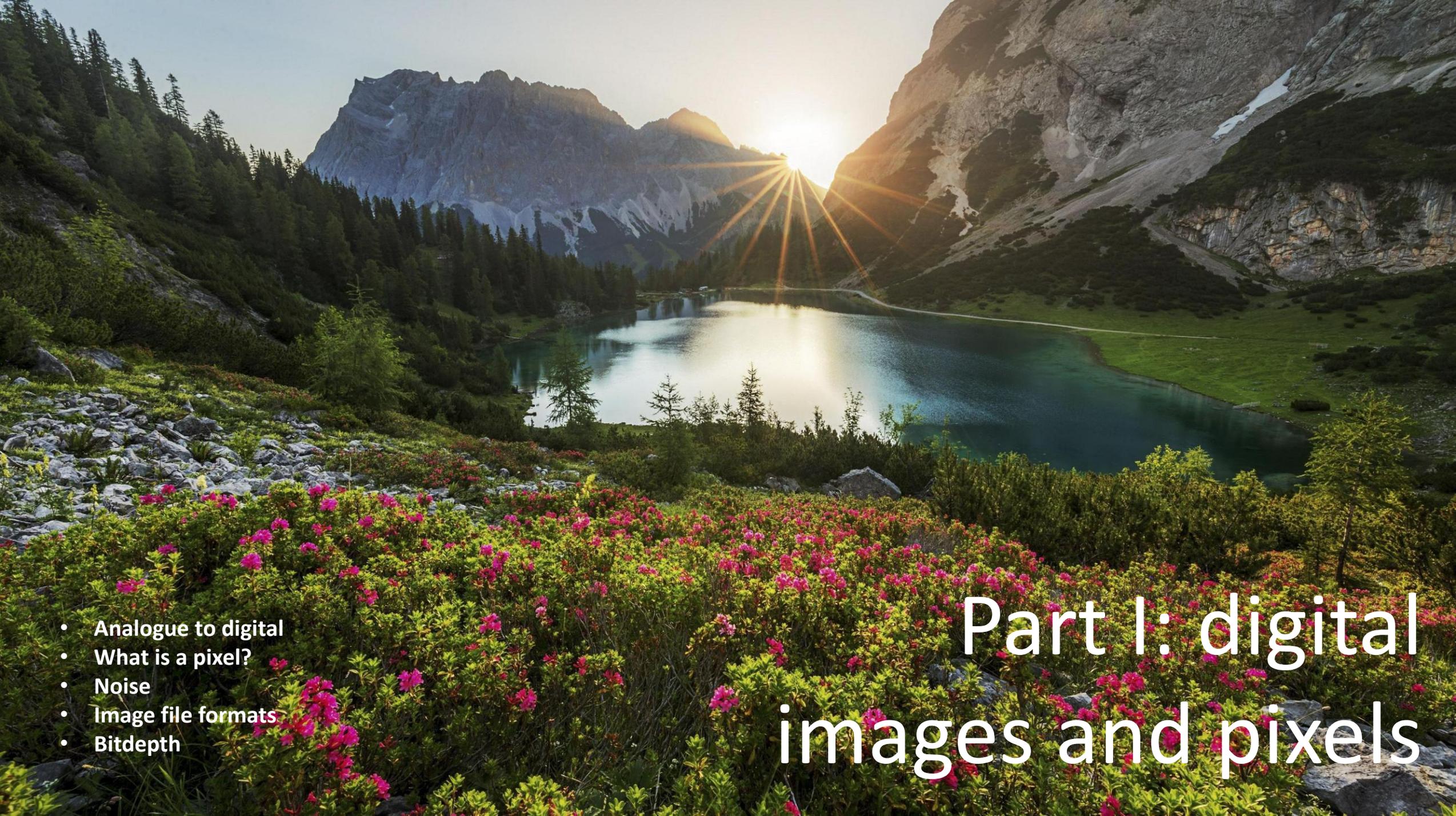
- Graphical user interface
- Extendable with plugins
- Designed for scientific data
- Broad functionalities
- Scripting possibilities

Chapter I (what is a scientific image?)

Chapter II (How to open a scientific image?)

Chapter III (What do you see then?)

Chapter IV (Prepare an image for publication)



- Analogue to digital
- What is a pixel?
- Noise
- Image file formats
- Bitdepth

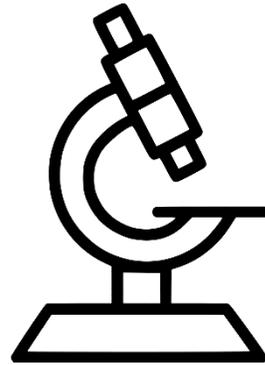
Part I: digital images and pixels

Image processing: GIGO

Image processing starts with recording good images



Sample preparation



Imaging



Image processing

Central idea:

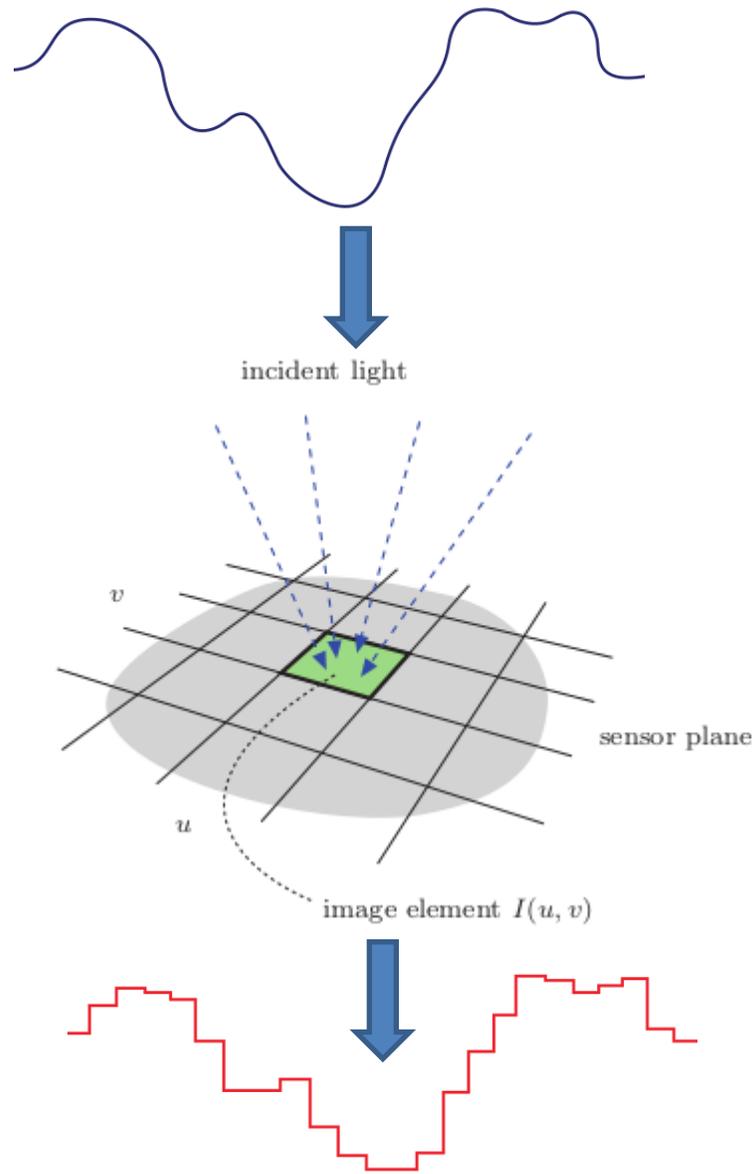
GIGO

Garbage in, garbage out

Good image in → maybe better image out

Bad image in → never a good image out

Going digital: from a continuous to a discrete function

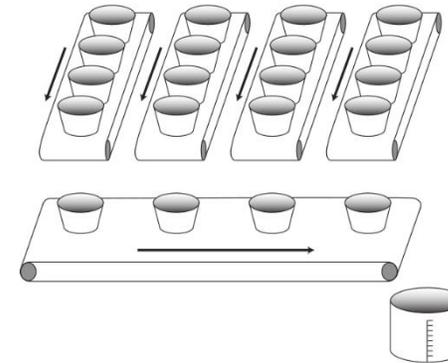
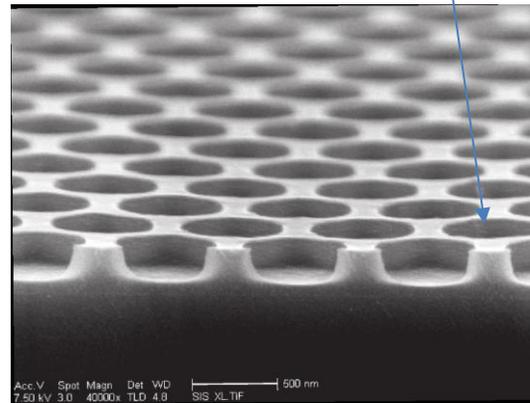
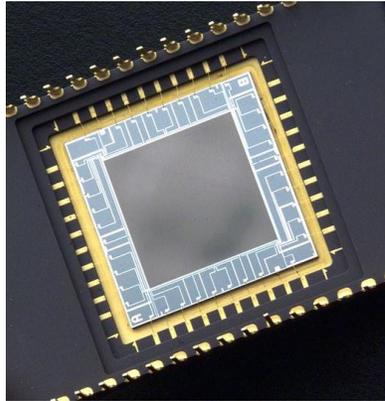


1. Spatial sampling
2. Time sampling
3. Analog to digital conversion

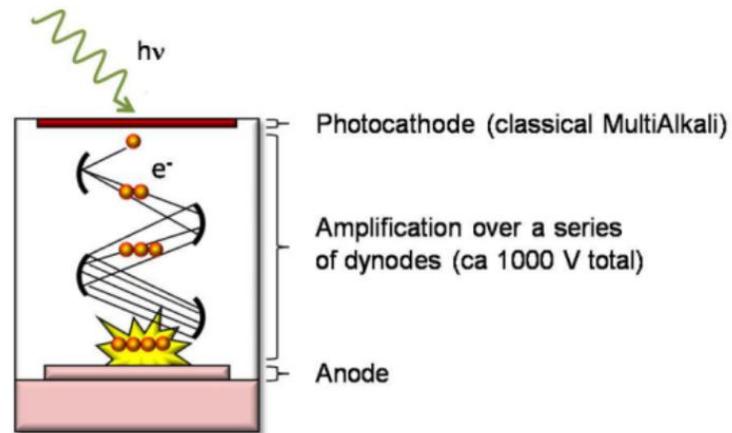
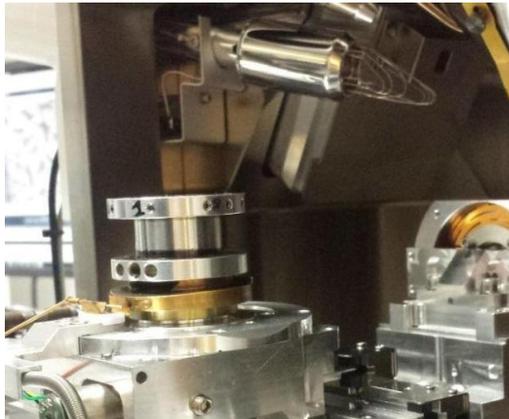
Going digital: Cameras and detectors

Photodiode (or well, or element)

Camera
(e.g LM, TEM)



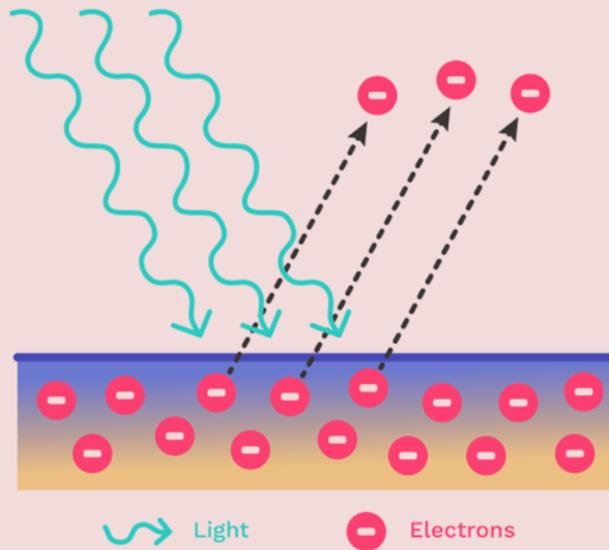
Detector
(e.g LSM, SEM, STEM)



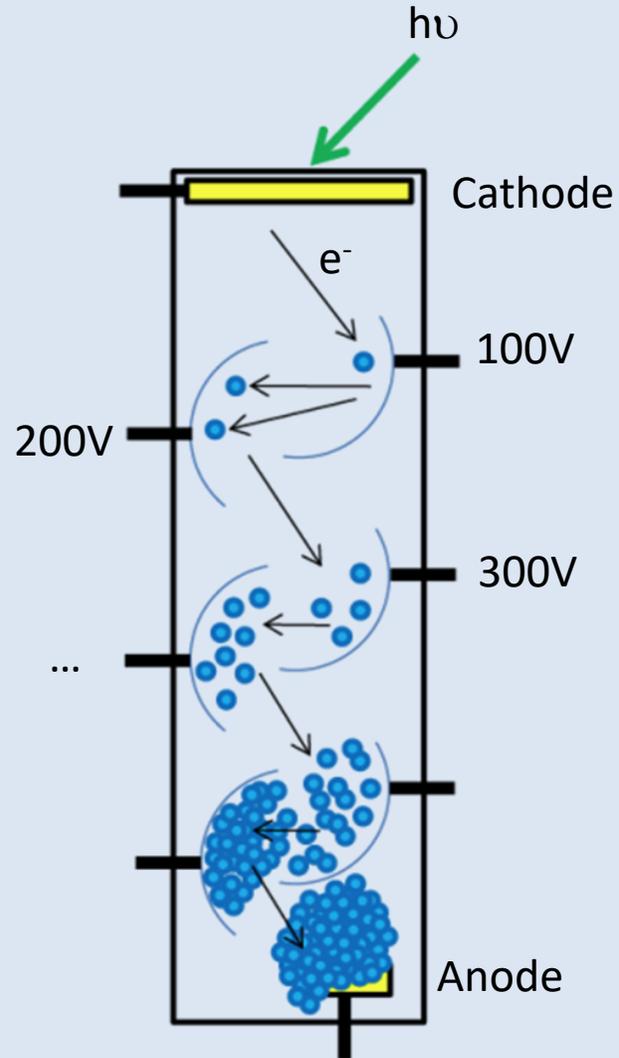
Going digital: from photon to pixel intensity

1. Photon to electron conversion

The **outer photoelectric effect**: release of free electrons when light shines on them.



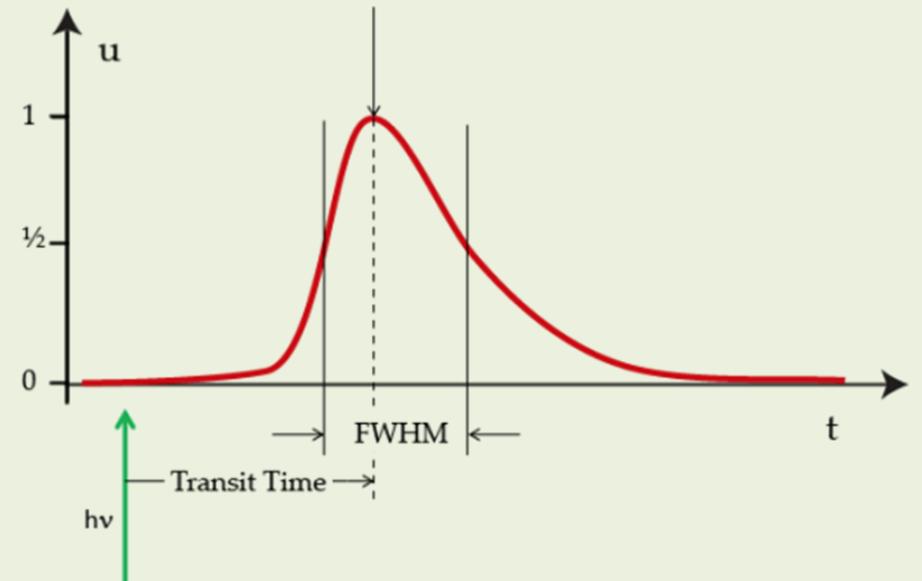
2. Signal multiplier / dynodes



3. Time sampling

The charge is sampled at the anode. Electrons do not all arrive at the same time (Because of the different paths can be taken)

→ **distribution**



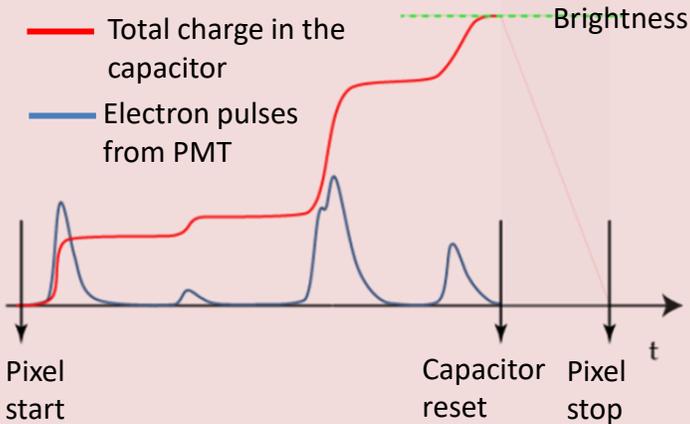
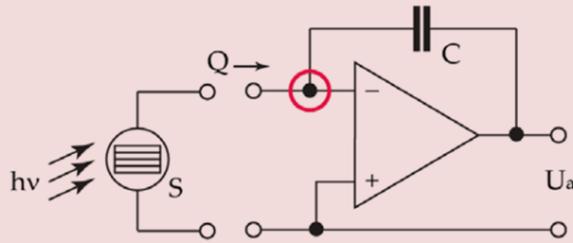
To measure pulse shape:

- full width half maximum. Time const.: 20 ns.
- not symmetrical; the rise time is 2-3X times faster than the decay time.

Going digital: from photon to pixel intensity (PMT)

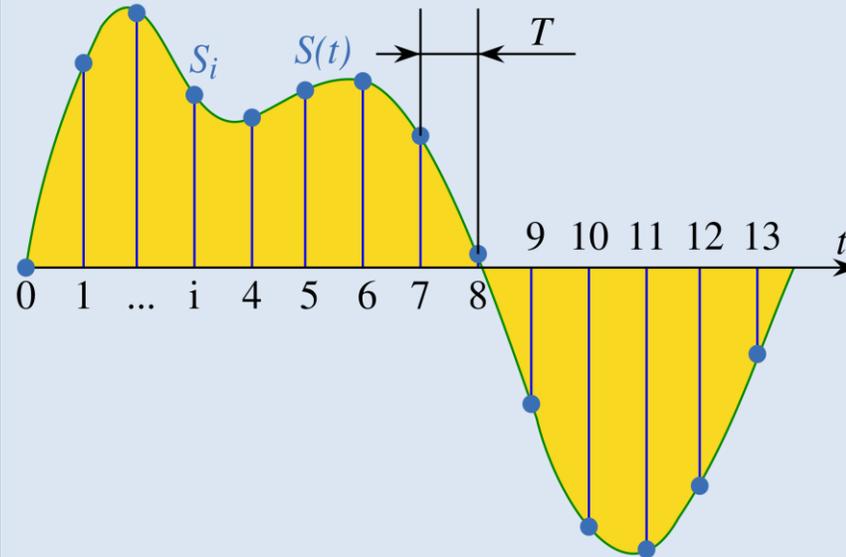
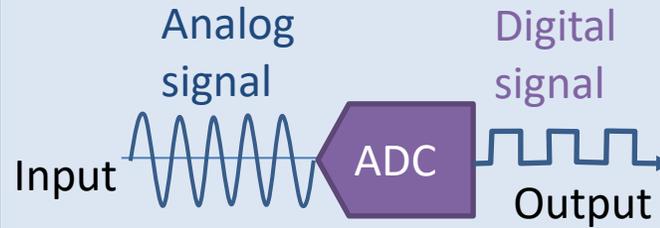
4. The charge amplifier / time sampling

All charges coming from the PMT are stored in a capacitor .



A fraction of the total time is lost for resetting the charge amplifier

5. Analogue-to-digital converter



Shannon Sampling Theorem

If a function $S(t)$ contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced $1/(2B)$ seconds apart

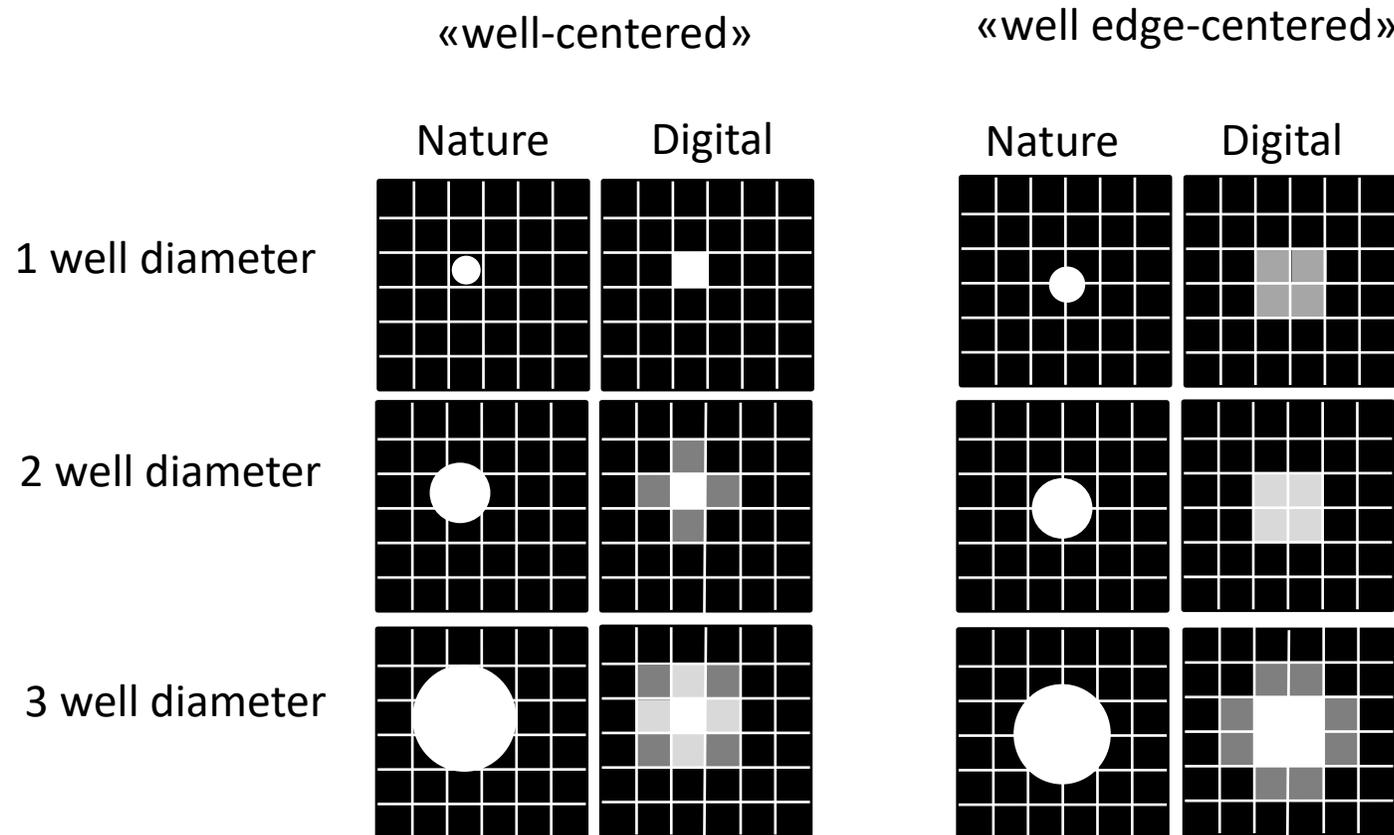
Shannon Sampling Theorem - revisited

If a series of discrete values has been sampled $1/(2B)$ apart, it does not contain information smaller than B hertz.

Shannon Sampling Theorem - applied

If an image has been recorded at a resolution of e.g. 100 nm, it does not contain information smaller than 200 nm.

What is a pixel: Nyquist theorem



Nyquist's theorem ("howto convert analog to digital"):

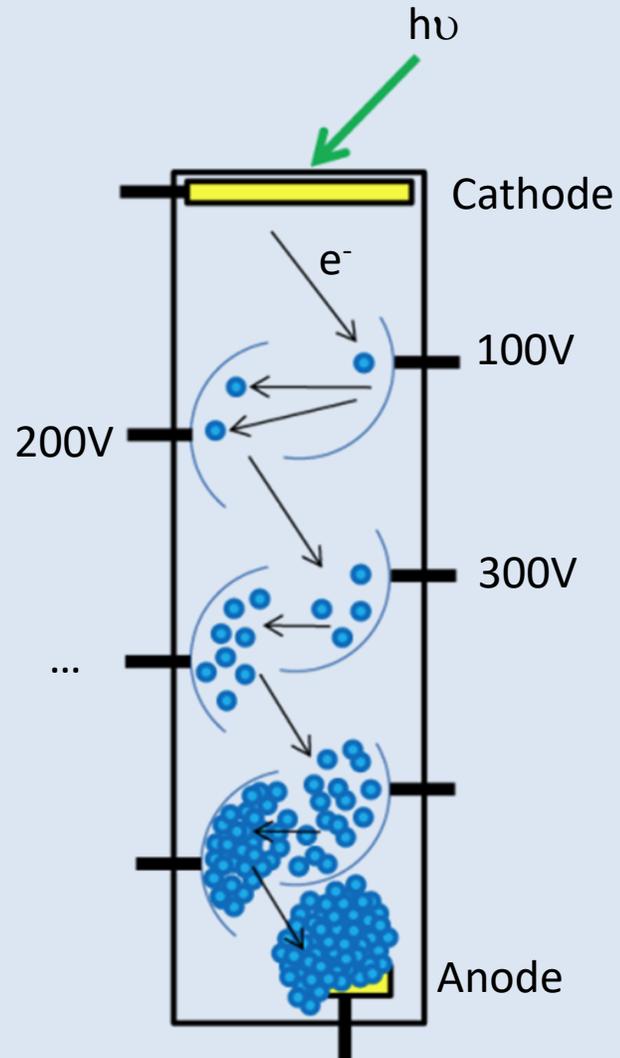
the frequency of the digital sample should be twice that of the analog frequency

For digital images:

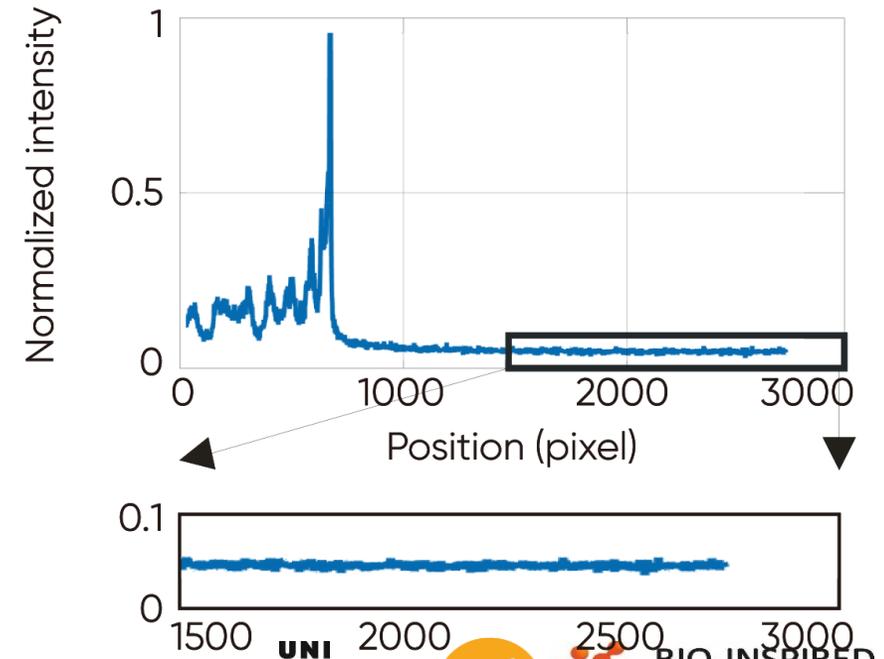
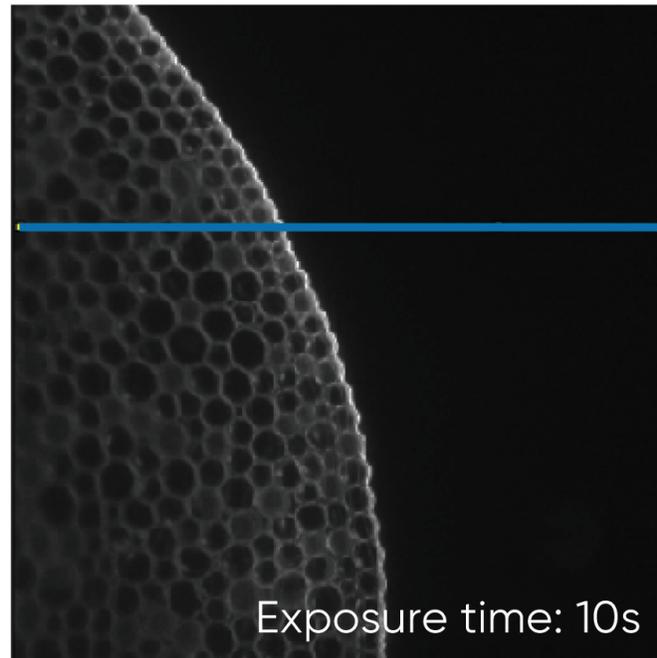
= a "sampling rate" of 2 wells relative to the object image size.

Noise: dark noise (PMT)

Signal multiplier / dynodes



- Even if no photons hit the cathode at all, events will still be measured at the anode
- Triggered by **thermal electrons** emitted by the cathode.
- Amplified by the following stages of dynodes.
- These events are known as 'dark noise'
- For weak signals, they can no longer be differentiated from the background.



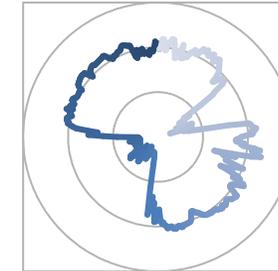
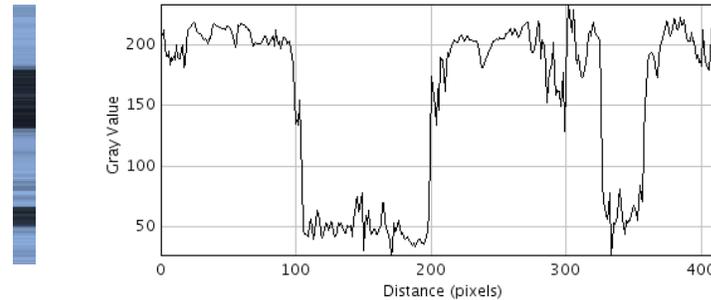
Going digital – what is a digital image?

A digital image is an ordered, rectilinear array (or grid) of **numbers (usually integers: 0,1,2,3...)**. Each element (=number) in the grid is also known as a picture element or 'Pixel'

Spectrum

1 dimensional array

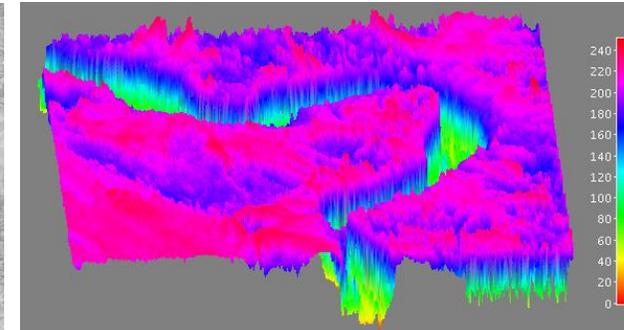
208
209
213
198
191
191
195
184
190
188
191
188
200



Image

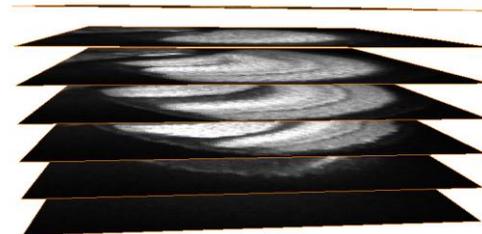
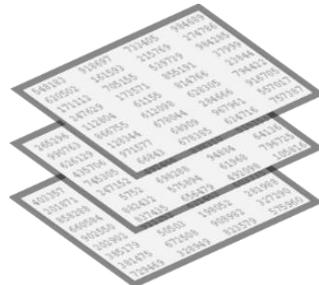
2 dimensional array

82	72	78	86	65	41
157	144	167	188	201	191
185	191	195	188	188	191
193	195	195	191	189	171
173	170	181	192	194	191
210	214	206	202	203	201
237	224	221	230	232	221
183	180	190	188	192	181
178	170	159	187	195	181
167	164	170	186	192	181
159	162	164	184	170	161
180	172	165	172	185	171
193	180	196	195	185	171
167	184	182	183	180	171
195	191	182	189	195	181
183	188	184	183	174	161
161	166	165	170	168	161



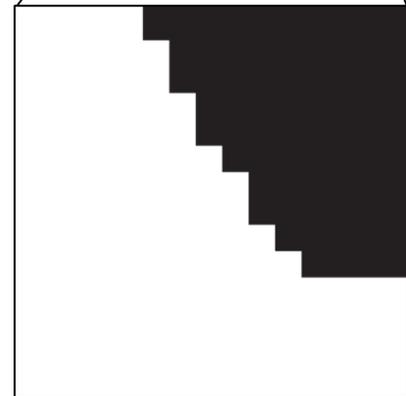
Stacks

3D array
(= volume stack or video)

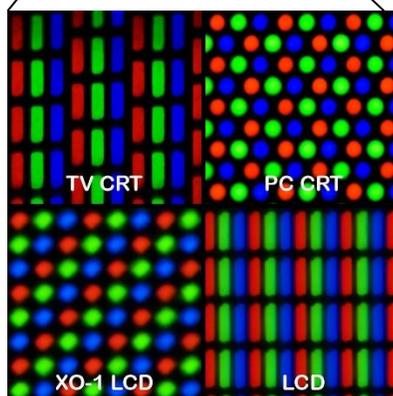
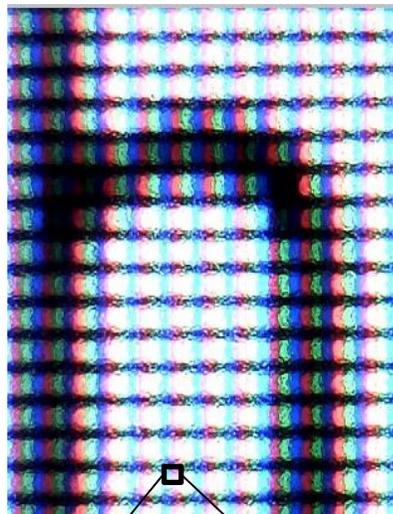


What is a pixel? – it depends on the context

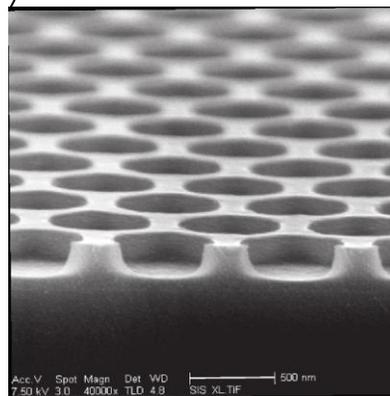
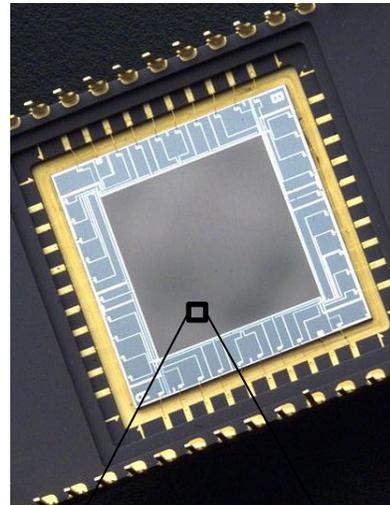
Digital image
Pixels



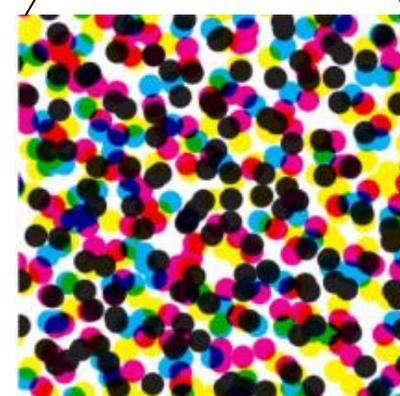
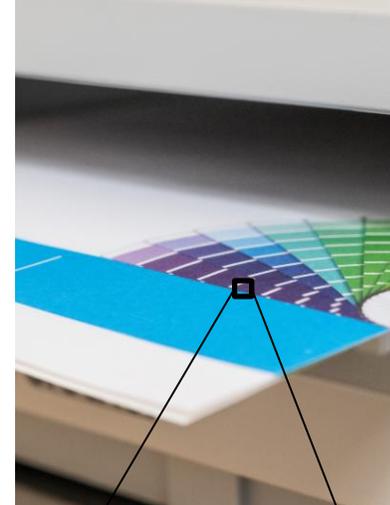
Display
Triades



Camera chip
Wells



Printed output
Dots



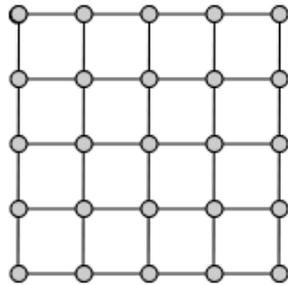
Pixel =
'Pic(ture) +
el(ement)'

What is a pixel – the little square model

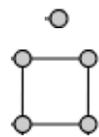
Problem: how to convert a continuous spectrum (nature) to a discrete set of intensities (digital)

Thought experiment

Consider a grid:



What are the pixels?



The corners?

5x5 grid?

The squares?

4x4 grid?

Where is the center of a pixel?

'Integerists'

Pixel (i, j) corresponding to square $(x, y) \rightarrow i - 0.5 \leq x \leq i + 0.5, j - 0.5 \leq y \leq j + 0.5$

A pixel is a point sample
(it exists only at **one point***)

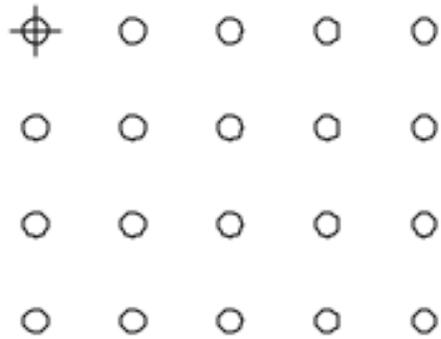
Pixel (i, j) corresponding to square $(x, y) \rightarrow i - 1 \leq x \leq i + 1, j - 1 \leq y \leq j + 1$

*For color pixels, there might be 3 or even 4 values at that point

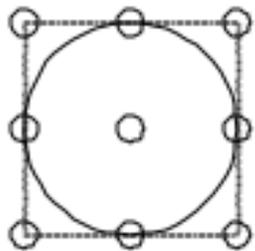
An image = an array on point samples (Shannon sampling theorem!)

What is a pixel – the reconstruction filter

Problem: how to convert a continuous spectrum (nature) to a discrete set of intensities (digital)



A 5x4 image



The footprint of a
"reconstruction filter"
(e.g. beam)

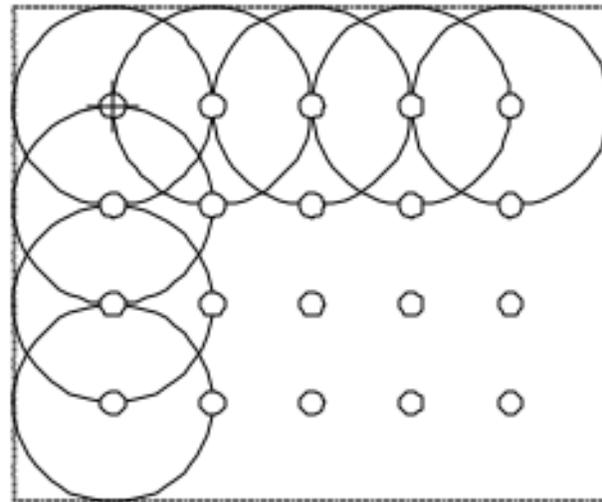
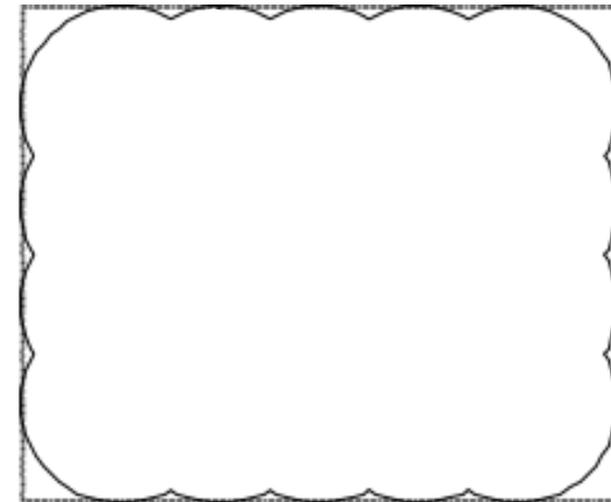


Image under construction

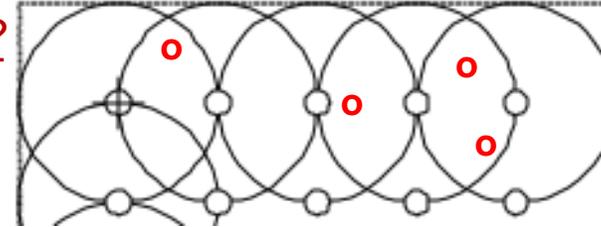


Recorded image

At no point, a
square is involved!

What is a pixel – the point sample model

Q: What about the information (O) below the Shannon limit?



A: we simply do not know!
But you could try to guess. This is called **Reconstruction**
or **interpolation**

Nature is continuous (has a value at every position), a digital image is a finite set of measurements: information will be lost.

Using the point sample model, we include the notion that information has been lost, and can talk about the best ways to measure that loss

***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

What is a resolution? And what is magnification?

	Magnification	Resolution	Resolving power
Low			
High			
Assessment	Easy	Very difficult	

Noise

Additive noise

Gaussian noise

Sum of all natural sources forming a normal distribution (=Gaussian distribution).

- **Johnson–Nyquist noise** (thermal vibrations of atoms in conductors)
- **Black-body radiation** (radiation from the earth or other objects)
- **Sensor noise** (when not recording at 0 K)
- **Electronic circuit noise** (impedance in electronic cables)
- **kTC noise** (Effects of capacitors)



FFT filtering, Gaussian smoothing.

Poisson (shot) noise

Caused by quantum effects due to the movement of discrete, quantized, packets

- In the source (light)
- In the electronics (electric current).
- **Dark shot** noise: shot noise from the dark leakage current in the image sensor



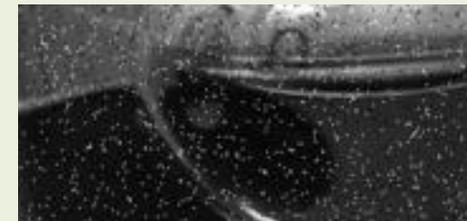
high intensity, exposure times
Correct with bias images

Multiplicative noise

Salt and pepper noise

Caused by errors. Typically B/W distribution

- During data transmission
- Failure in a memory cell
- Analog-to-digital converter errors
- Occasionally in TEM (X-rays)
- In camera-based systems: dead pixels



Median filtering

Repetitive noise

caused by electrical interference during the image capturing process.



Signal deconvolution in reciprocal space

Intermezzo - how does a computer work?

Examples of decimal (=10 digits) counting

- begins with 1 digit (rightmost digit or first digit)
- When all available symbols are exhausted:
 - the least significant digit is reset to 0,
 - the next digit (one position to the left) is incremented (=overflow)

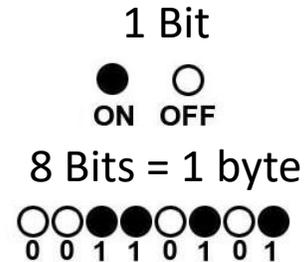
0 1 2 3 4 5 6 7 8 9
 10 11 12 13 14 15 16 17 18 19

Examples of binary (=2 digits) counting

- begins with 1 digit (rightmost digit or first digit)
- When all available symbols are exhausted:
 - the least significant digit is reset to 0,
 - the next digit (one position to the left) is incremented (=overflow)

0 1 10 11 100 101 110 111
 (0) (1) (2) (3) (4) (5) (6) (7)

Computer = on/off
 = binary digits ('bit')



1024 bytes = 1 KB

KB = Kilobyte

1024 KB = 1 MB

MB = Megabyte

1024 MB = 1 GB

GB = Gigabyte

1024 GB = 1 TB

TB = Terabyte

1024 TB = 1 PB

PB = Petabyte

From numbers to text

- ASCII: American Standard Code for Information Interchange
- List of numbers with defined characters

Dec	Char	Dec	Char	Dec	Char
32	SPACE	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(72	H	104	h
41)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3				
52	4				
53	5				
54	6				
55	7				
56	8				
57	9				
58	:				
59	;				
60	<	91	[120	\
61	=	92	\	124	
62	>	93]	125	}
63	?	94	^	126	~
		95	_	127	

'AMI'

(dec): 65 77 73

(bin): 1000001 1001101 1001001

File formats

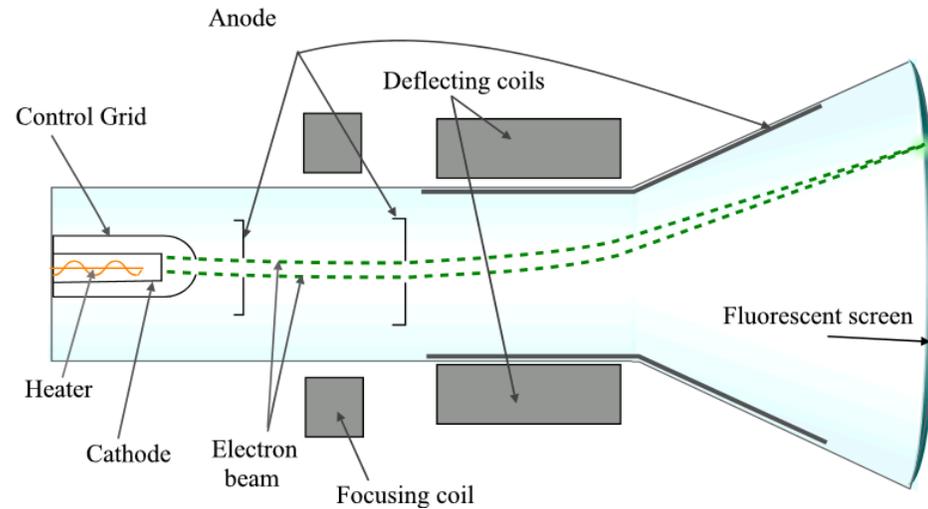
From concept: 3 different approaches:

1. Vector graphics formats
2. Raster graphics formats
3. Hierarchical file formats

File formats: Raster graphics

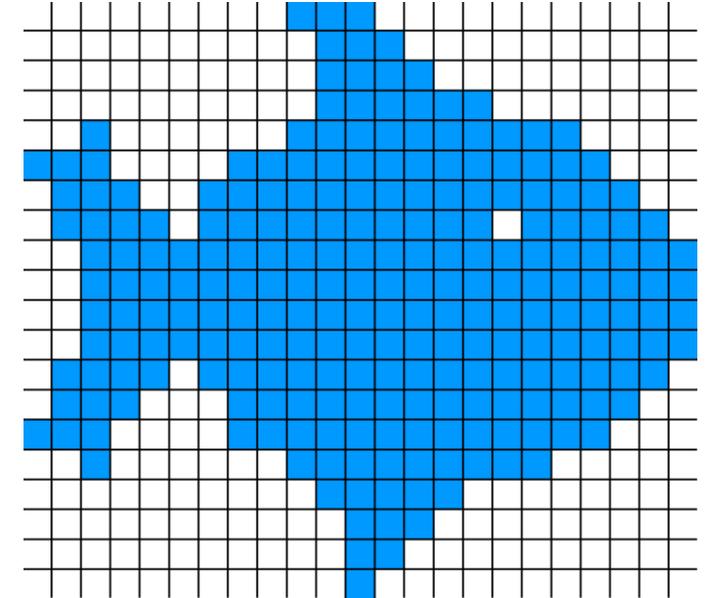
Formats
Origin

tif, jpg, png, bmp, webP, psb, ...
raster scan of cathode ray tubes (i.e. early TV screens)
'Raster' comes from rastrum (Lat.) = to scrape



Concept

Tessellation (tiling) of a 2D plane with a value for every cell ('pixel')



File formats: Raster graphics – Headers

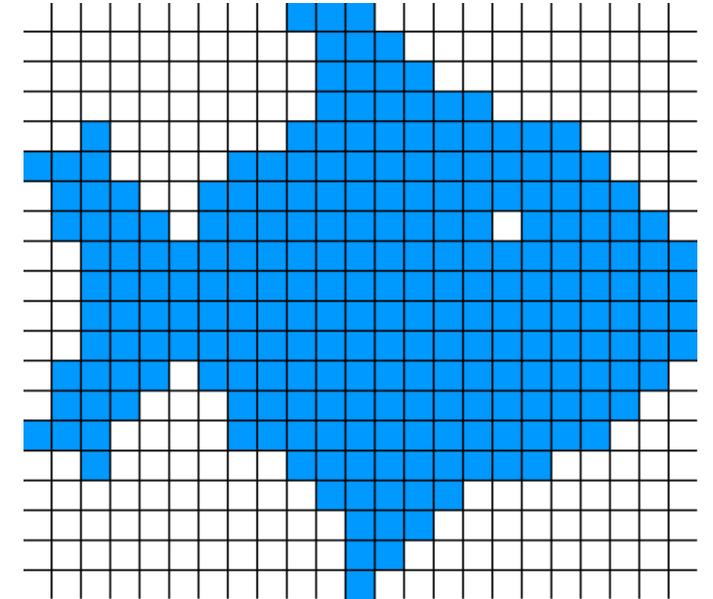
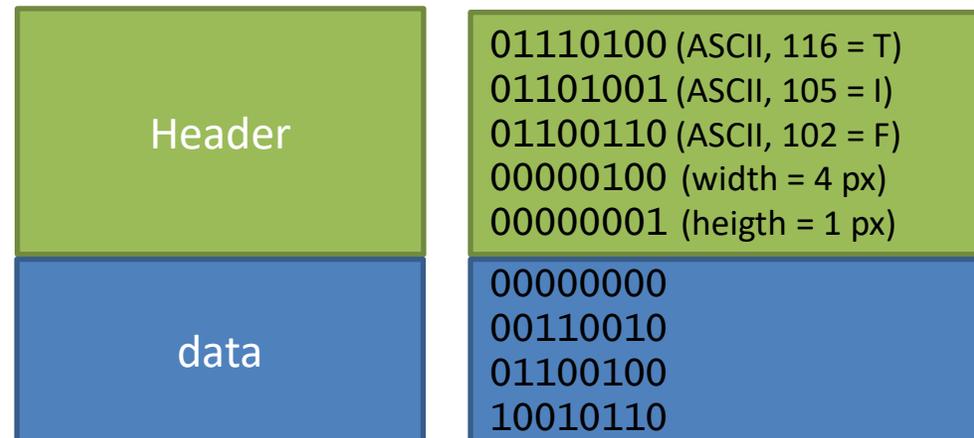
Concept Problem

Tessellation (tiling) of a 2D plane with a value for every cell ('pixel')
Metadata: How large is the image? When was it made? But also:
e.g. medical data (patient information)
e.g. scientific data (experiment information)
e.g. consumer data (GPS coordinates, time and date)

Solution

Header
contains height and width of the raster (=image) +lots of other data

Example



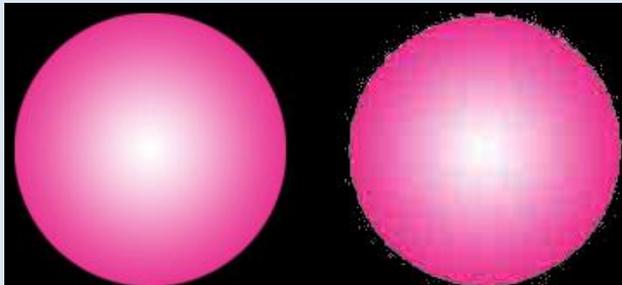
File formats: Raster graphics – Compression

Concept
Problem
Solution

Tessellation (tiling) of a 2D plane with a value for every cell ('pixel')
Data gets really big
Compression. There are 2 compression concepts:

Lossy compression

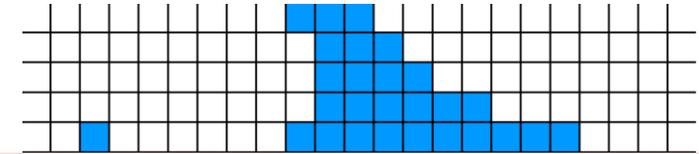
The original data is approximated, but not restored
Example: JPG
Note: compression artefacts!



Lossless compression

the original data can be restored
Example: TIFF, PNG
Note: No artefacts!

Algorithms:
Run length encoding (RLE)
Lempel Ziv (LZx, uses Lookups)



Original data 32 bits

```
00000000
00110010
01100100
10010110
```

RLE compressed 29 bits

```
8x0
2x0 2x1 2x0 10
0 2x1 2x0 1 2x0
1 2x0 10 2x1 0
```

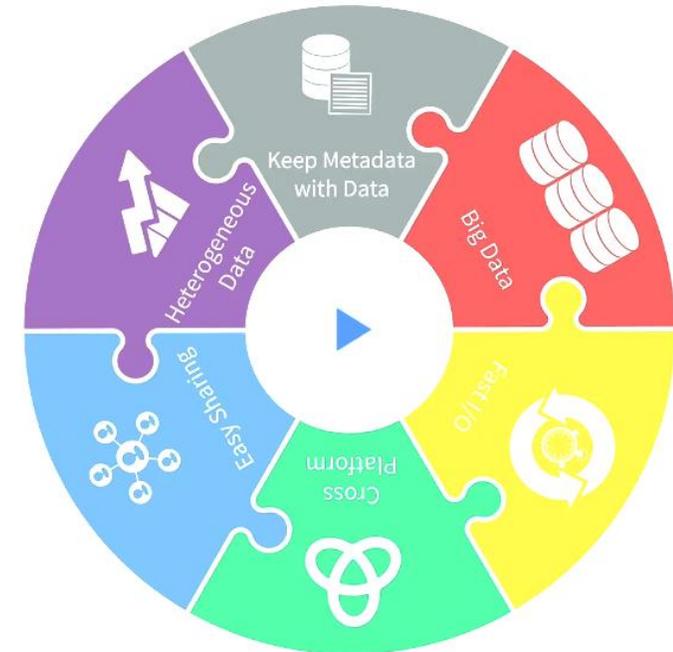
LZ compressed 38 bits

```
A = 00
B = 11
C = 10
4XA
A B A C
0 B A 1 A
1 A C B 0
```

```
Repeated to get 1024x1024:
Original 8 Mb
RLE 7.25 Mb (-10%)
LZ 7 bytes (!)
(-99.99999%)
```

File formats: Hierarchical data formats (HDF)

Formats	h5
Origin	1987 by the Graphics Foundations Task Force. Around 1992, NASA investigated 15 different file formats for use in the Earth Observing System (EOS) project and settled for HDF.
Concept	Container of multiple, heterogenous data (including really big datasets), with metadata
Some advantages	<ul style="list-style-type: none">- really really big datasets possible (e.g. tomography data)- hierarchical data (e.g. google maps)- heterogenous data (e.g. EDX spectra with images)- slices: only part of an image can be read (e.g. google maps)- embedded coding (allows advanced compression techniques)



File formats: raster graphics: bit depth

The range of the values a pixel can take is called the bit depth

$$2^n = \text{number of shades}$$
$$n = \text{bitdepth}$$

1 bit (=black and white, =binary image)

2 shades



Binary (e.g. Thresholding!)

8 bit (=common consumer cameras)

256 shades



~9 bit (=human eye)

~500 shades



12 bit (=middle range scientific cameras)

4096 shades



16 bit (=high range scientific cameras)

65536 shades



32 bit (=only computational)

4 294 967 296 shades



Grayscale

Signed (e.g. Signed 16bit). -32768 → +32768

Special

File formats: raster graphics - bit depth & colour

The range of the values a Pixel can take is called the bit depth

$2^n = \text{number of shades}$
 $n = \text{bitdepth}$

RGB (=3 channels of each 8 bit = 24 bit) 256 x 256 x 256 colours



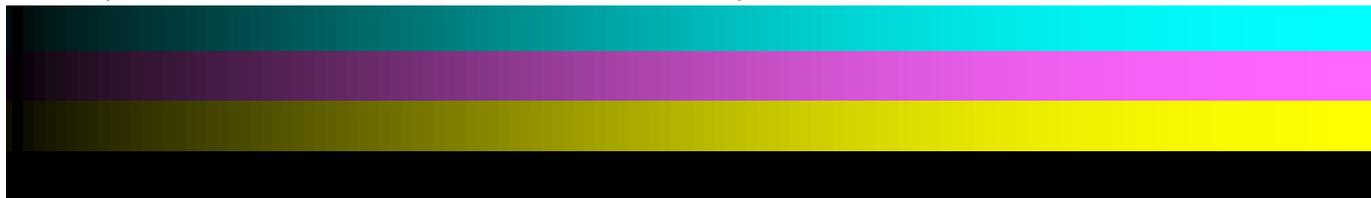
RGBA (=4 channels of each 8 bit = 32 bit) 256 x 256 x 256 colours x 256 Alpha



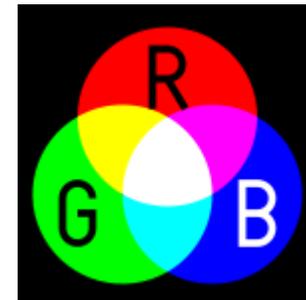
48 bit RGB (= 3 channels of each 16 bit = 48 bit)



CMYK (=4 channels of each 8 bit = 32 bit) 256 x 256x 256x 256 colours

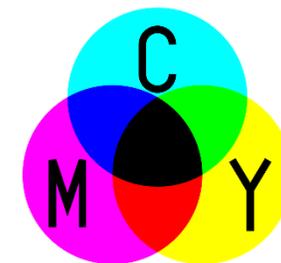


Additive color mixing (Displays)



Computational

Subtractive color mixing (pigments, Printers)



Note: light sources and dyes/pigments

Additive color mixing

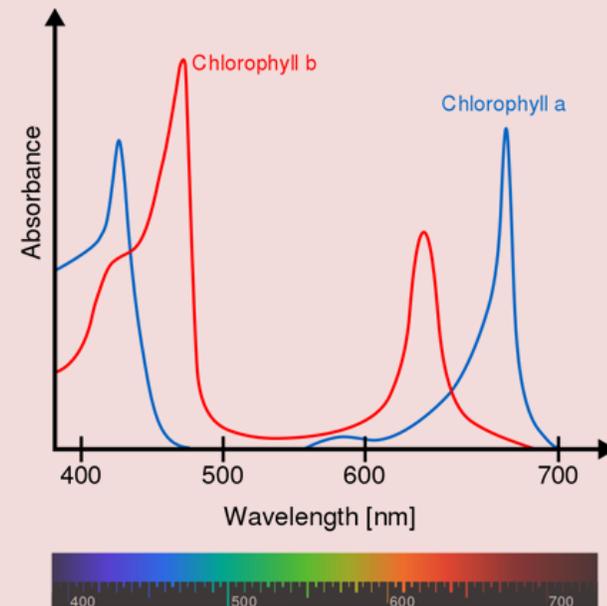
Light sources



A green laser is green because it emits green photons

Subtractive color mixing

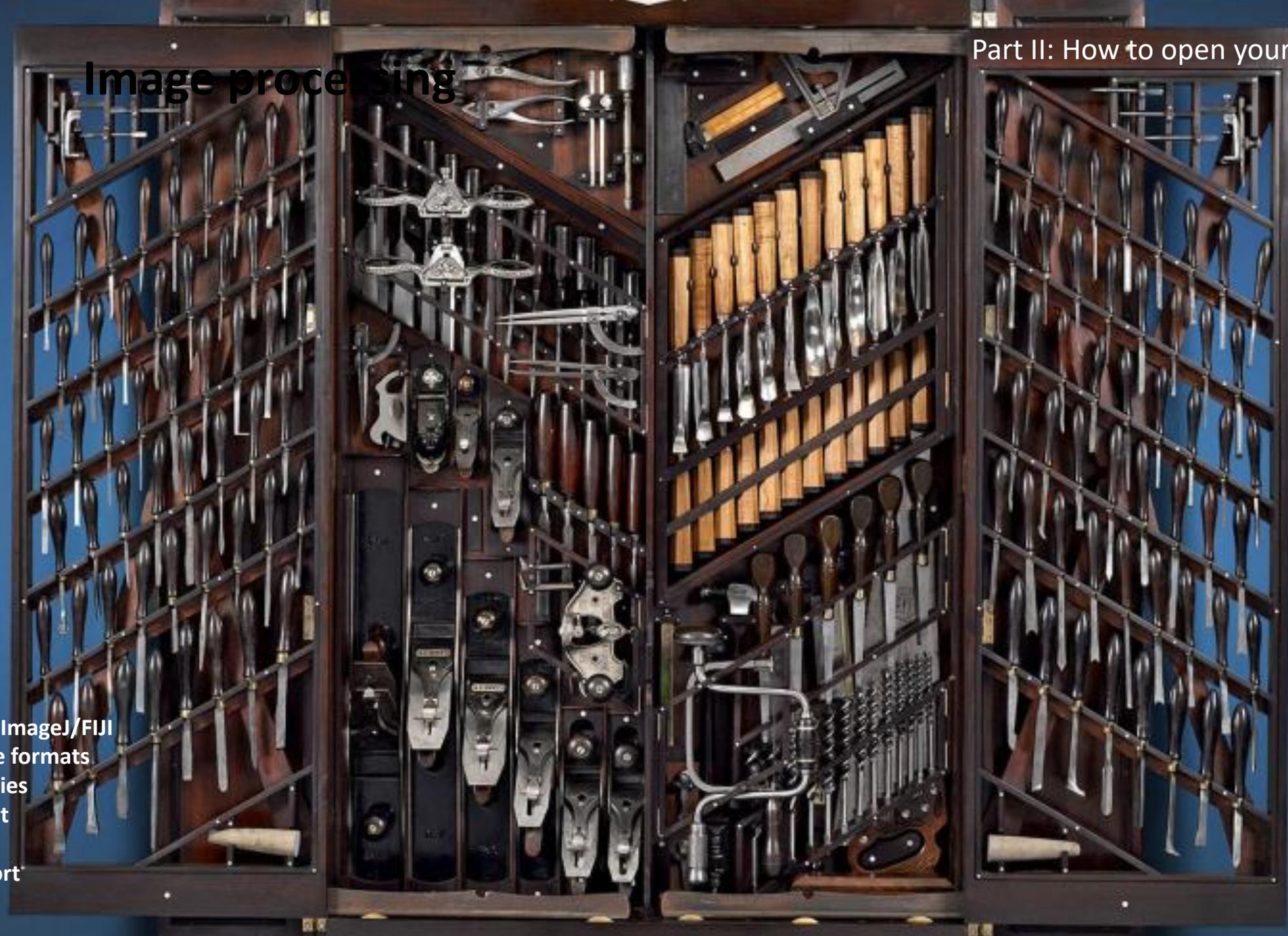
Dyes and pigments



A leaf is green because it absorbs all white light except the green photons (which it reflects)

Image processing

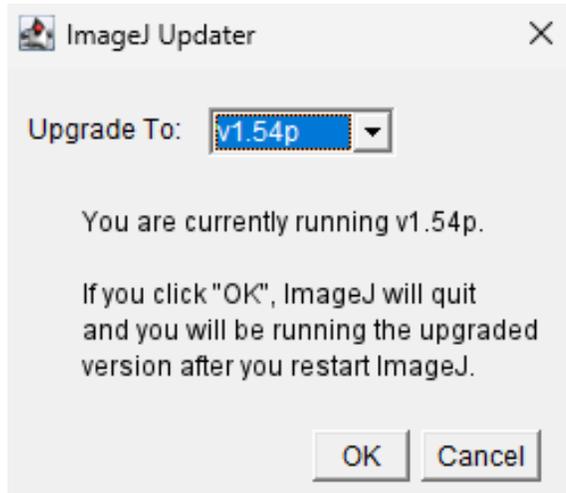
Part II: How to open your data?



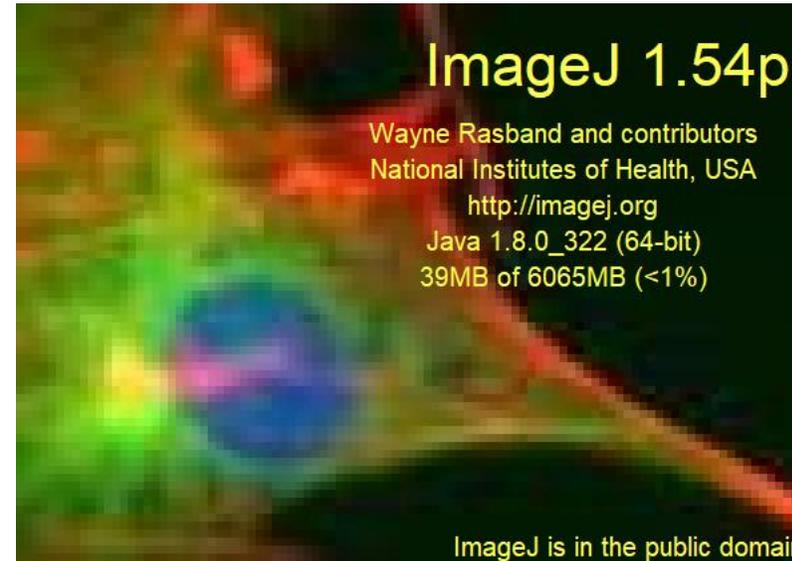
- Updating ImageJ/FIJI
- Native file formats
- Repositories
- FIJI import
- Plugins
- Raw import

0. Update FIJI

Help > update ImageJ...



Start ImageJ, then Help > About ImageJ...



1. How to open your data in FIJI?

5 Possibilities:

- Native file formats
- Use a repository
- Use Import (Fiji only!)
- Use a plugin
- Through RAW import (raster graphics only)

1. Native file formats

5 Possibilities:

- Native file formats
- Use Import (Fiji only!)
- Use a repository
- Use a plugin
- Through RAW import (raster graphics only)

Format	1 bit	8 bit	16 bit	32 bit	RGB	stacks	Hyper-stacks	Compression
TIFF	✓	✓	✓	✓	✓	✓	✓	Lossless
GIF	✓	✓			✓	✓		Lossless
JPEG		✓			✓			lossy
PNG	✓	✓	✓					Lossless
DICOM	✓	✓	✓	✓	✓	✓	✓	Lossless
BMP	✓	✓			✓			Lossless
PGM	✓	✓	✓		✓			Lossless
FITS	✓	✓	✓		✓			Lossless
AVI	✓	✓			✓	✓		Lossless

TIFF is the 'default' format of ImageJ (OME-TIFF)

DICOM is a standard popular in the medical imaging community

FITS Flexible Image Transport System: adopted by the astronomical community

PGM: Portable GrayMap

AVI container format, only uncompressed AVIs supported

Red: no header information

Native and non-native file formats

EXERCISE 1

Try to open example 1 A, B, C or D

File > open...

Or drag and drop the icon from the folder onto ImageJ/FIJI

Does it work?

Native and non-native file formats

EXERCISE 1

Try to open example 1 A, B, C or D

File > open...

Or drag and drop the icon from

```
Console
Edit
Console Log
[INFO] Reading available sites from https://imagej.net/
Plate :/sample/0/plate/
Well :/sample/0/plate/PLATE_00/experiment/
Site :/sample/0/plate/PLATE_00/experiment/WELL_00/position/
[Fatal Error] Example%201C%20-%20non-native%20file%20formats%20-%20Si04.ser:1:1
```

```
Log
File Edit Font
at ij.io.Opener.openImage(Opener.java:241)
at ij.io.Opener.open(Opener.java:104)
at ij.io.Opener.openAndAddToRecent(Opener.java:
at ij.plugin.DragAndDrop.openFile(DragAndDrop.java:
at ij.plugin.DragAndDrop.run(DragAndDrop.java:1
at java.lang.Thread.run(Thread.java:750)
```

Does it work?

No... Examples 1D does not open
(maybe even 1C and 1B also not)

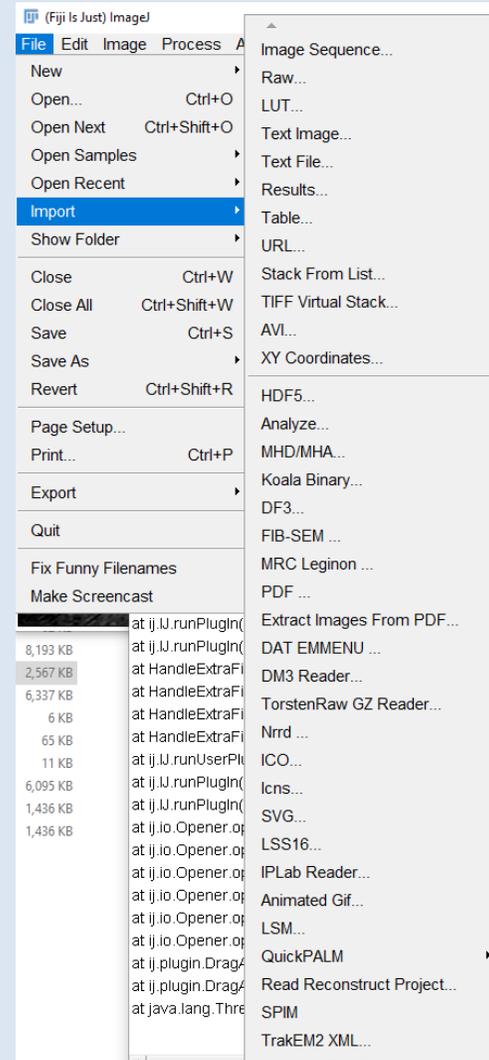
```
Error
Encountered an Error while translating
OK
```

2. Using a Fiji resident plugin (Import)

5 Possibilities:

- Native file formats
- Use Import (Fiji only!)
- Use a repository
- Use a plugin
- Through RAW import (raster graphics only)

File > Import >



2. Using a Fiji resident plugin (Import)

EXERCISE 1

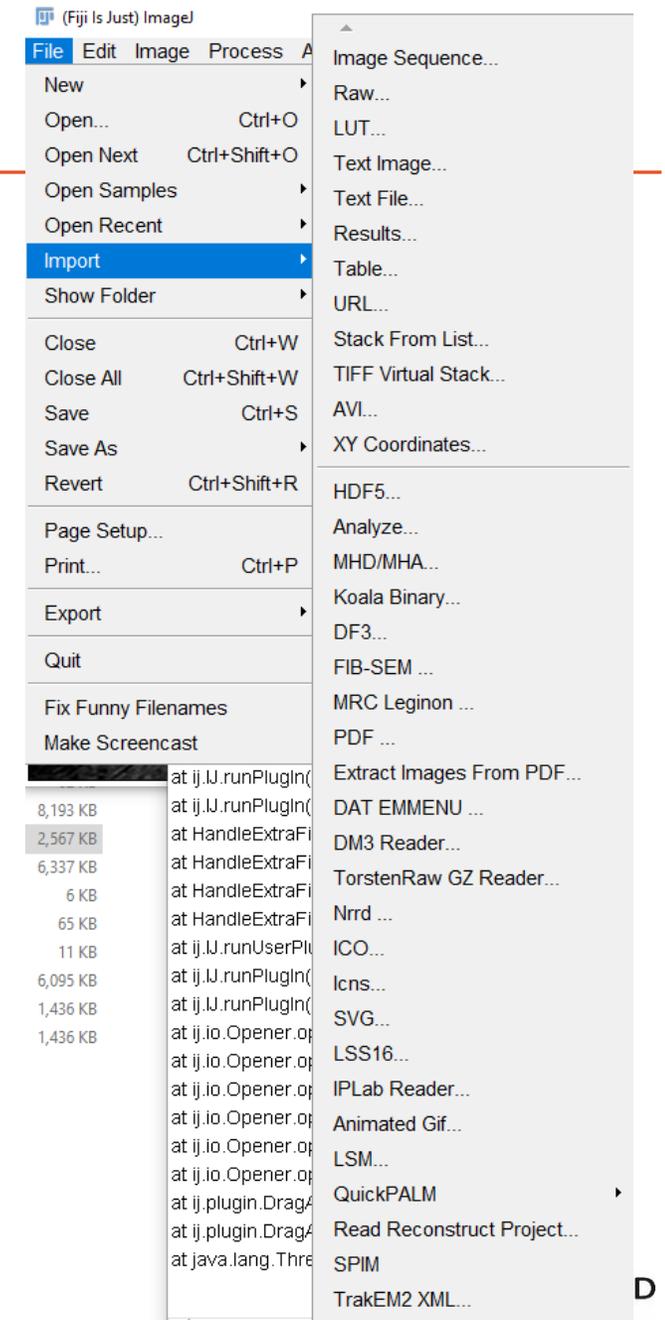
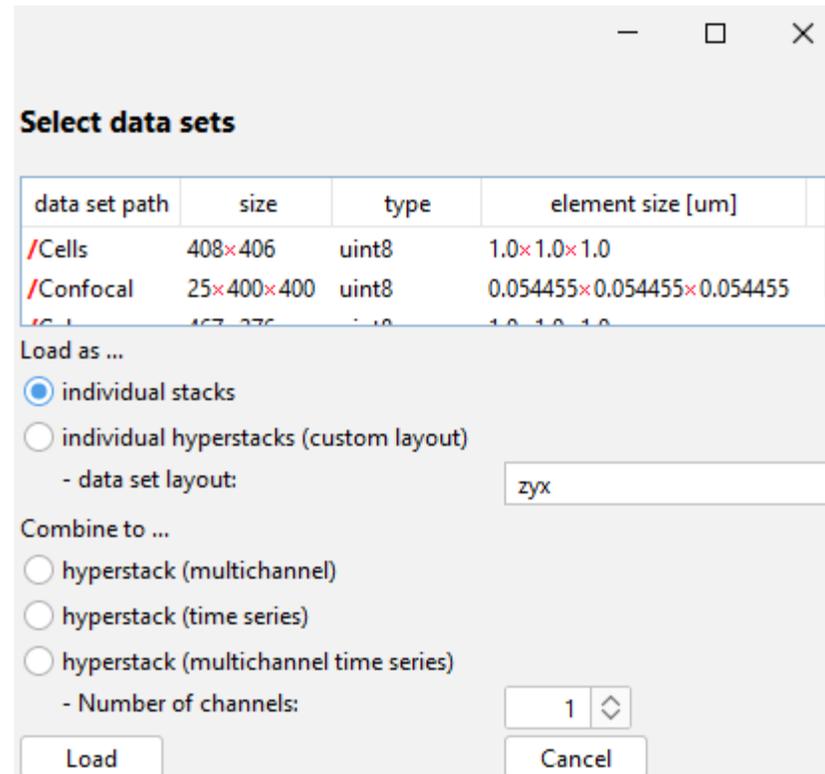
Try to open the example HDF file (1C)

File > Import > HDF5

Point to the file example 1C

The file contains 3 datasets (!)

- Select `/Cells` in data set path
- load as 'Individual stacks'
- Click 'Load'



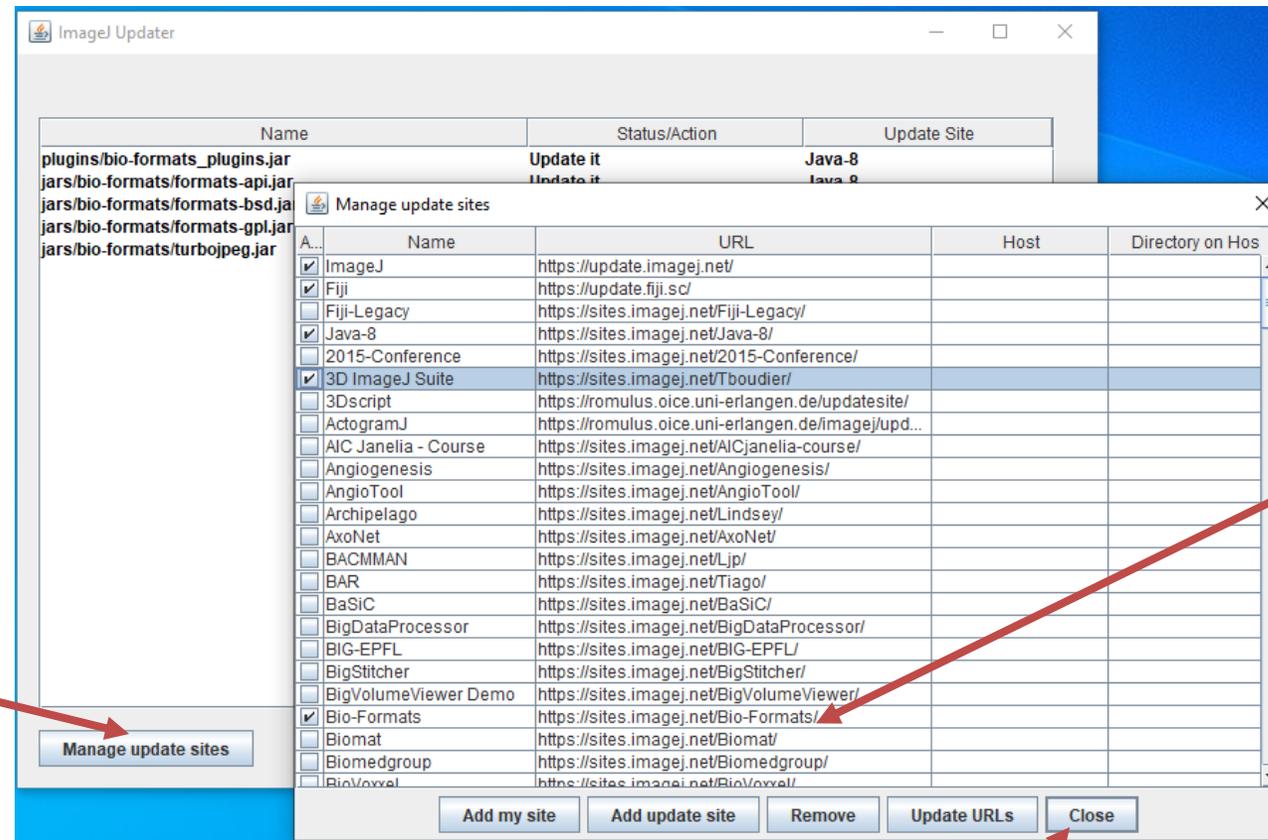
3. Using a plugin from the repository

5 Possibilities:

- Native file formats
- Use Import (Fiji only!)
- Use a repository
- Use a plugin
- Through RAW import (raster graphics only)

From the repositories

- Help > Update...



2. Select the repositories you wish to add. In this case: **Bio-Formats**

1 Choose Manage update sites

3. Close and
4. Apply changes

3. Using a plugin from the repository

5 Possibilities:

- Native file formats
- Use Import (Fiji only!)
- Use a repository
- Use a plugin
- Through RAW import (raster graphics only)

Help > update...

Name	Status/Action	Update Site
plugins/Sholl_Analysis.jar	Update it	Java-8
plugins/Simple_Neurite_Tracer.jar	Update it	Java-8
plugins/bigdataviewer_fiji.jar	Update it	Java-8
plugins/bigwarp_fiji.jar	Update it	Java-8
plugins/bio-formats_p		
plugins/n5-viewer_fiji.jar		
jars/FilamentDetector.jar		
jars/bigdataviewer-coi		
jars/bio-formats/form		
jars/bio-formats/form		
jars/bio-formats/form		
jars/bio-formats/ome-		
jars/bio-formats/ome-		
jars/bio-formats/specification.jar	Update it	Java-8
jars/bio-formats/turbojpeg.jar	Update it	Java-8
jars/imglib2-cache.jar	Update it	Java-8
jars/n5.jar	Update it	Java-8
jars/n5-ij.jar	Update it	Java-8

<input checked="" type="checkbox"/> 3D ImageJ Suite	https://sites.imagej.net/Tboudier/
<input checked="" type="checkbox"/> Bio-Formats	https://sites.imagej.net/Bio-Formats/
<input checked="" type="checkbox"/> CSBDeep	https://sites.imagej.net/CSBDeep/
<input checked="" type="checkbox"/> DeepImageJ	https://sites.imagej.net/DeepImageJ/
<input type="checkbox"/> HDF5	https://sites.imagej.net/Ronneber/
<input checked="" type="checkbox"/> Labkit	https://sites.imagej.net/Labkit/

Information: Please restart ImageJ and call Help>Update to continue with the update

File > import > Bio-Formats

or

Plugins > Import > Bio-Formats > Bio-Formats Importer

EXERCISE 1

Try to open example 1B with the Bio-Formats plugin

Stack viewing: View stack with: Hyperstack, Stack order: XYZCT

Metadata viewing: Display metadata, Display OME-XML metadata, Display ROIs, ROIs Import Mode: ROI manager

Dataset organization: Group files with similar names, Open files individually, Swap dimensions, Open all series, Concatenate series when compatible

Memory management: Use virtual stack, Specify range for each series, Crop on import

Color options: Color mode: Default, Autoscale

Information: View stack with - The type of image viewer to use when displaying the dataset. Possible choices are:

- Metadata only - Display no pixels, only metadata.
- Standard ImageJ - This option is deprecated (i.e. intended for use by old macros only). Please use Hyperstack instead.
- Hyperstack - Display the pixels in ImageJ's built-in 5D viewer.
- Data Browser - Display the pixels in the multidimensional Data Browser viewer. The Data Browser has some additional features on top of the normal ImageJ hyperstack.
- Image5D - Display the pixels in Joachim Walter's Image5D viewer. Requires the Image5D plugin.
- View5D - Display the pixels in Rainer Heintzmann's View5D viewer. Requires the View5D plugin.

3. Using a plugin from the repository

5 Possibilities:

- Native file formats
- Use Import (Fiji only!)
- Use a repository
- Use a plugin
- Through RAW import (raster graphics only)

Update lists / repositories

ImageJ
Fiji
3D ImageJ Suite
3Dscript
ActogramJ
AIC Janelia
Angiogenesis
AngioTool
Archipelago
AxoNet
BACMMAN
BAR
BaSiC
BigDataProcessor
BIG-EPFL
BigStitcher
BigVolumeViewer Demo
Bio-Formats
Biomat
Biomedgroup
BioVoxel
Blind Analysis Tools
BoneJ
CALM
CAMDU
CATS
CellTrackingChallenge
CIP
CircleSkinner
clij
clij2
clijx-assistant
clijx-assistant-extensions
ClearVolume
CMCI-EMBL
CMP-BIA tools
CMTK Registration
Colocalization by Cross Correlation
Colour Deconvolution2

Bio-formats repository

3i SlideBook
AIM
AVI (Audio Video Interleave)
Adobe Photoshop PSD
Alicona 3D
Amersham Biosciences Gel
Amira Mesh
Amnis FlowSight
Analyze 7.5
Andor Bio-Imaging Division (ABD) TIFF
Animated PNG
Aperio AFI
Aperio SVS TIFF
Applied Precision CellWorX
Axon Raw Format
BD Pathway
Becker & Hickl SPCImage
Bio-Rad Gel
Bio-Rad PIC
Bio-Rad SCN
Bitplane Imaris
Bruker MRI
Burleigh
Canon DNG
CellH5
CellVoyager
Cellomics
DICOM
DeltaVision
ECAT7
EPS (Encapsulated PostScript)
Evotec/PerkinElmer Opera Flex
FEI
FEI TIFF
Gatan Digital Micrograph
Gatan Digital Micrograph 2
Hamamatsu Aquacosmos NAF
Hamamatsu HIS
Hamamatsu VMS

4. Using a plugin from the internet

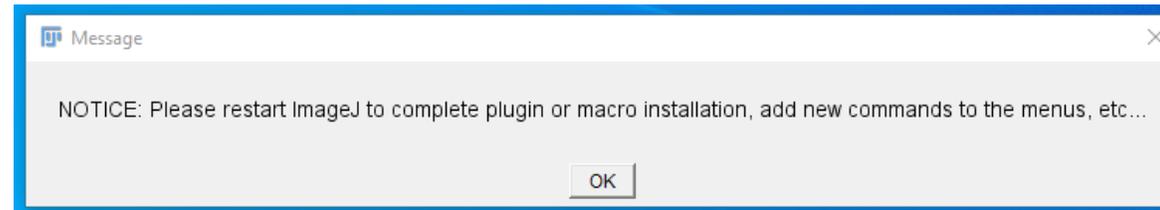
5 Possibilities:

- Native file formats
- Use a repository
- Use Import (Fiji only!)
- Use a plugin
- Through RAW import (raster graphics only)

Installing a plugin from a downloaded .jar file
(<https://imagej.nih.gov/ij//plugins/tia-reader.html>)

3 ways of installing the jar file (it is also provided):

1. Drag and drop the file onto Imagej (and save in the plugin folder)
2. Plugins > install plugin... and point to the .jar file
3. Copy the .jar file into the plugins folder of your FIJI folder

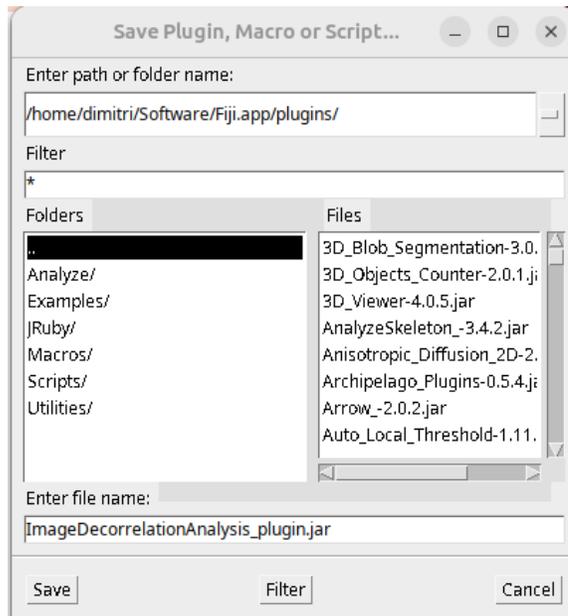


In any case: **Restart FIJI** (close it and start it again)

Find the new functionality under
Plugins > Input / output > Tia Reader

Provided on the ImageJ website

ASCII header format
 SPE Images
 PICT, Targa using Jimi
 Biorad Z-Series
 Leica SP multi-channel stacks
 QuickTime
 Jimi Stack Writer
 AVI Writer
 JMF Reader
 Animated Gif
 PDS Images
 AVI Reader
 LSM Reader (Zeiss LSM confocal microscopes)
 Qness RAW
 IPLab Reader
 Excel Writer
 Multi FDF
 VFF Opener
 OpenSIF (opens Andor SIF files)
 EXIF Reader
 Bruker NMR
 Zeiss ZVI Reader
 ISAC images (e.g., Fuji BAS scanners)
 Gatan DM3 Reader
 Deltavision Opener
 Nanoscope AFM files
 NIFTI Input/Output
 UNC format images
 PDF Writer
 Leica SP2 TIFF Sequence
 EPS (Encapsulated PostScript) Writer
 PerkinElmer Reader
 Nikon ND2 Reader (Windows only)
 TIA Reader (FEI/Emispec .ser files)
 ...



4. Using a plugin from the internet

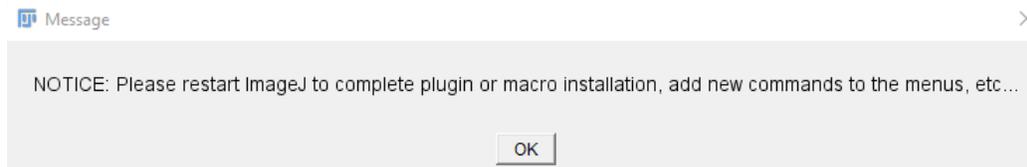
EXERCISE 1

Install the TIA reader plugin (TIA_Reader.jar) if you have not done so

Then try to open Example 1D, the .ser file, using plugins > input/output > TIA reader

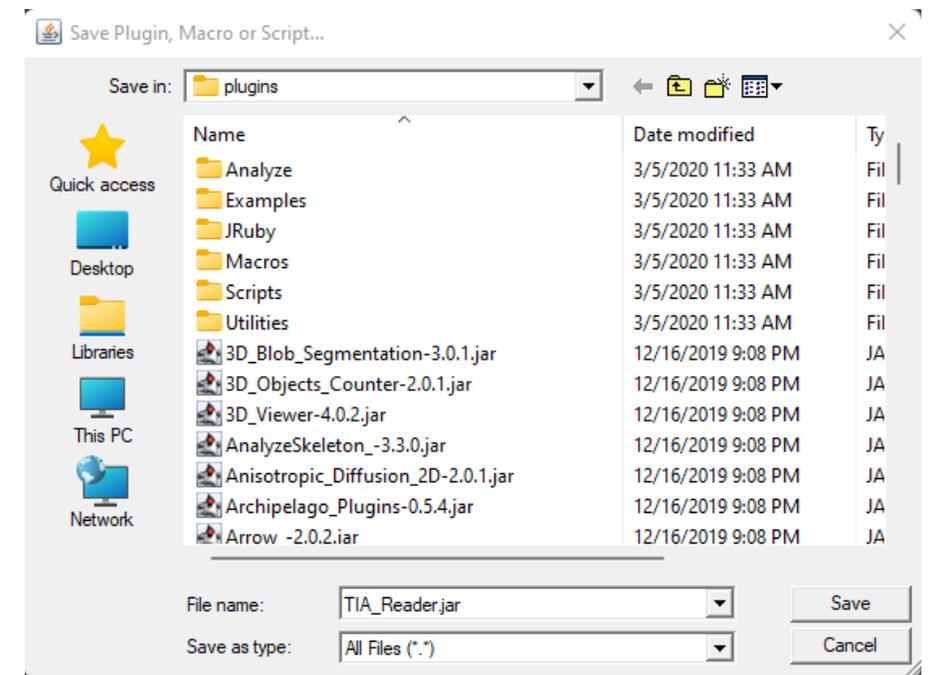
Installing a plugin from a downloaded .jar file

- drag and drop the .jar file onto Imagej or click Plugins > install plugin...
- Save it in the plugins folder of your FIJI folder



- Restart FIJI
- Then find the new functionality under Plugins > **Input / output** > Tia Reader
- Plugins > Input / output > Tia Reader
- Point to the .ser file
- Click 'Open'

Does it work?

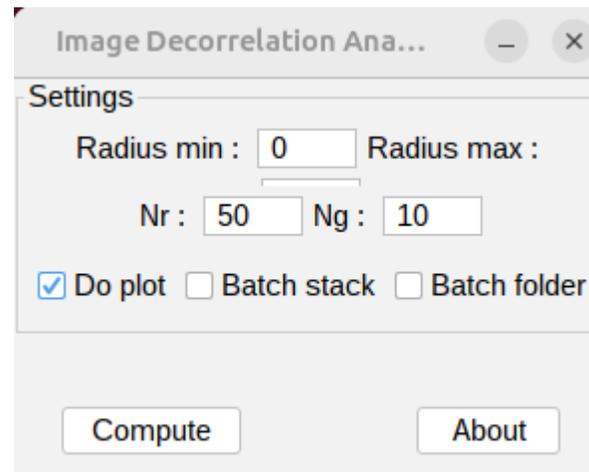


4. Using a plugin from the internet (short resolution intermezzo)

EXERCISE 1

Install the image decorrelation plugin.

- First install the dependencies: JTransforms-3.1-with-dependencies.jar, then the decorrelation plugin:
- ImageDecorrelationAnalysis_plugin.jar (it is in the Image decorrelation plugin folder)
- Open Example 1B – Header – Cells.Ism with the Bio-Formats plugin (Plugins > Bio-Formats > Bio-Formats Importer)
- Run the Image Decorrelation plugin on the blue channel using the default settings (Plugins > Image Decorrelation Analysis)

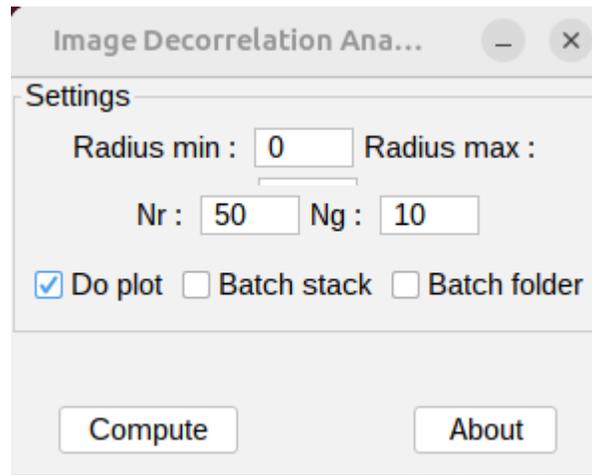


4. Using a plugin from the internet (short resolution intermezzo)

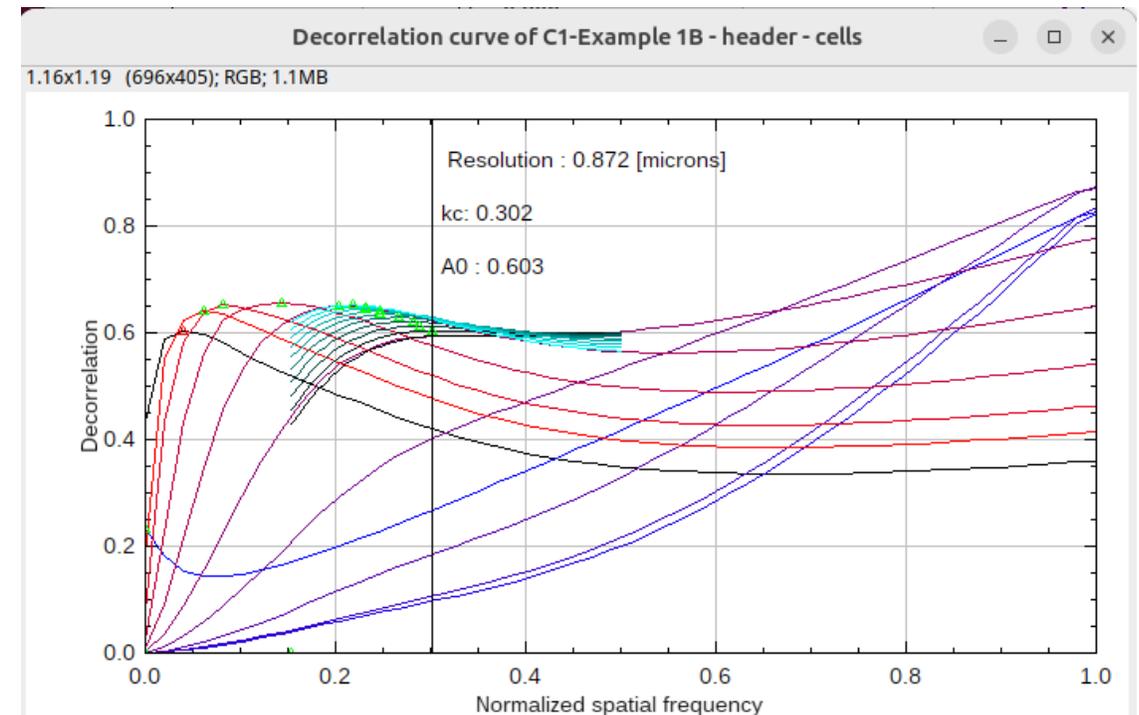
EXERCISE 1

Install the image decorrelation plugin.

- Run the Image Decorrelation plugin on the blue channel (Plugins > Image Decorrelation Analysis)



Label	Res.	units
C1-Example 1B - header - cells	0.872	[microns]



5. RAW import

5 Possibilities:

- Native file formats
- Use a repository
- Use Import (Fiji only!)
- Use a plugin
- Through RAW import (raster graphics only)



```
0000 49 49 2a 00 10 00 00 00 43 52 02 00 14 b7 00 00 II*.....CR..... TIFF header [CR2 header]
0010 11 00 .. IFD0 entries
0012 00 01 03 00 01 00 00 00 40 14 00 00 .....@... IFD0-00 ImageWidth
001e 01 01 03 00 01 00 00 00 80 0d 00 00 ..... IFD0-01 ImageHeight
002a 02 01 03 00 03 00 00 00 e2 00 00 00 ..... IFD0-02 BitsPerSample
0036 03 01 03 00 01 00 00 00 06 00 00 00 ..... IFD0-03 Compression
0042 0f 01 02 00 06 00 00 00 e8 00 00 00 ..... IFD0-04 Make
004e 10 01 02 00 0e 00 00 00 ee 00 00 00 ..... IFD0-05 Model
005a 11 01 04 00 01 00 00 00 64 1b 01 00 .....d... IFD0-06 PreviewImageStart
0066 12 01 03 00 01 00 00 00 01 00 00 00 ..... IFD0-07 Orientation
0072 17 01 04 00 01 00 00 00 b2 83 21 00 .....!... IFD0-08 PreviewImageLength
007e 1a 01 05 00 01 00 00 00 0e 01 00 00 ..... IFD0-09 XResolution
008a 1b 01 05 00 01 00 00 00 16 01 00 00 ..... IFD0-10 YResolution
0096 28 01 03 00 01 00 00 00 02 00 00 00 (..... IFD0-11 ResolutionUnit
00a2 32 01 02 00 14 00 00 00 1e 01 00 00 2..... IFD0-12 ModifyDate
00ae 3b 01 02 00 01 00 00 00 00 00 00 00 ;..... IFD0-13 Artist
00ba bc 02 01 00 00 20 00 00 74 b7 00 00 .....t... IFD0-14 ApplicationNotes
00c6 98 82 02 00 01 00 00 00 00 00 00 00 ..... IFD0-15 Copyright
00d2 69 87 04 00 01 00 00 00 b2 01 00 00 i..... IFD0-16 ExifOffset
00de 3e b6 >.. Next IFD
```

Total file size (in bytes) = Header size + Image data size

5. RAW import

3 Possibilities:

- Native file formats
- Use a plugin
- **Through RAW import (raster graphics only)**



```
01110100 (ASCII, 116 = T)
01101001 (ASCII, 105 = I)
01100110 (ASCII, 102 = F)
00000100 (width = 4 px)
00000001 (height = 1 px)
```

```
00000000
00110010
01100100
10010110
```

Header: 5 bytes (TIF, 4x1 pixel image)

Data: 4 bytes



Total file size: 9 bytes

Total file size (in bytes) = Header size + Image data size

5. RAW import

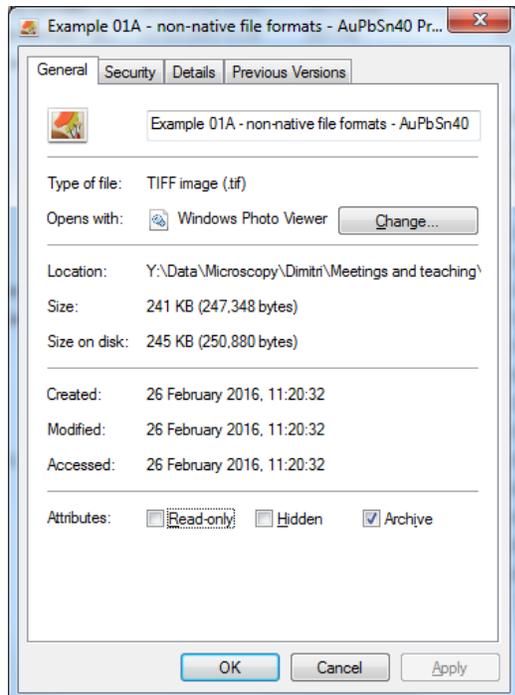
EXERCISE 1

Import through RAW (file > import > Raw...) of example 1D, the .SER file

Idea: we will only read the DATA – and jump over the HEADER

Example 1A

1. A priori information: Camera size = 600x412 px x 8 bit depth
2. Use your operating system to find the file size (in bytes) of Example 1A



Size: 241 KB (247,348 bytes)

Size on disk: 245 KB (250,880 bytes)

Example 1D

Camera size: Veleta at AMI (2048 x 2048, 16 bit)

Use your operating system to find the file size

5. RAW import

EXERCISE 1

Import through RAW (file > import > Raw...)

Example 1A, the .tif file

Example 1D, the .ser file

Idea: we will only read the DATA – and jump over the HEADER

Example 1A

1. Camera size: 600x412 px x 8 bit
2. 247 348 bytes

Example 1D

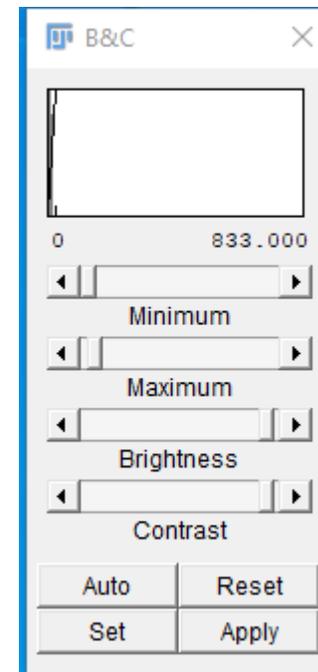
1. Camera size: Veleta at AMI (2048 x 2048, 16 bit)
2. 8 388 754 bytes

File > import > RAW...

If your image is black, update brightness contrast:

Image > adjust > Brightness / contrast...

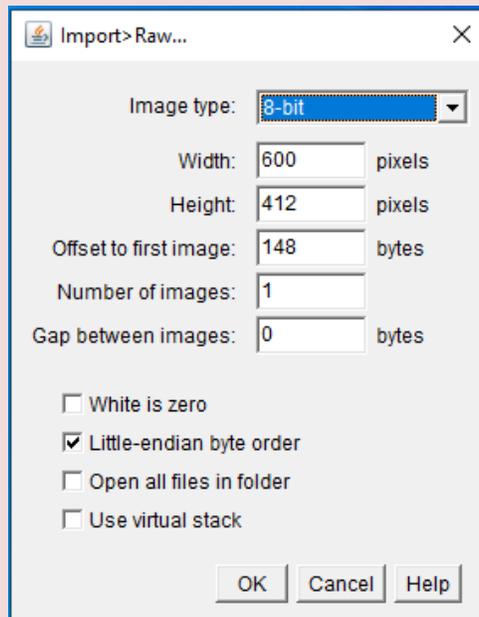
And click auto, or set (between 100 and 450)



5. RAW import

1. Data size: $600 \times 412 = 247\,200$ bytes
2. File size : 247 348 bytes

Therefore: header is $247\,348 - 247\,200 = 148$ bytes

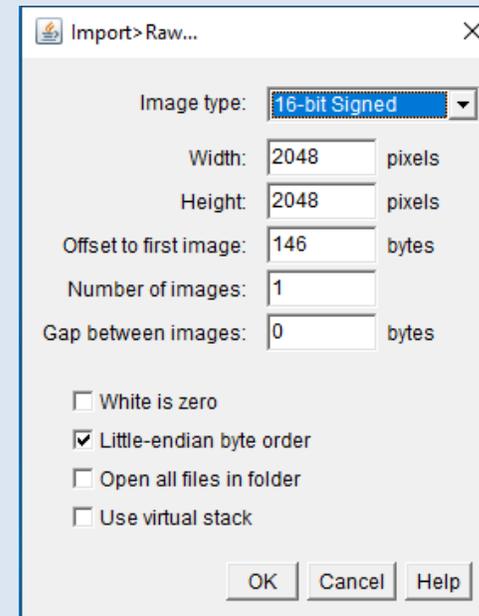


Endian: big-endian is often an “Apple thing”

1. Data size: $2048 \times 2048 \times 2 = 8\,388\,608$ bytes
(**why x 2??**)

2. File size : 8 388 754 bytes

Therefore: header is $8\,388\,754 - 8\,388\,608 = 146$ bytes



Header

Files may (must) contain meta data (additional, useful information about the image)
F.A.I.R. Principles (<https://www.go-fair.org/>), open source data



Open Example 1D using File > **import** > **RAW...**
Select Image > Show info...

Title: Example 1D - non-native file formats - SiO4.ser
Width: 2048 pixels
Height: 2048 pixels
Size: 8MB
Pixel size: 1x1 pixel²
ID: -77
Bits per pixel: 16 (signed)
Display range: 100 - 450
No threshold

Open Example 1D **using the TIA Plugin** >
Select Image > Show info...

Title: /home/.../Example 1D - non-native file formats.ser
Width: **8.3029 microns** (2048)
Height: **8.3029 microns** (2048)
Size: 8MB
Resolution: 246.6609 pixels per microns
Pixel size: **0.0041x0.0041 microns²**
ID: -63
Bits per pixel: 16 (unsigned, grayscale LUT)
Display range: 0-922
Pixel value range: 0-922
Image: 1/1 (1) No threshold
...

Summary: Howto open your data in FIJI

Native file formats

Use TIF whenever possible
Forget JPEG (has no header)
Metadata properly imported

FIJI Import

Using the repositories or from the internet

Repositories

500+ scientific file formats available through Repositories.
Metadata is (very very often) imported
E.g. Zeiss, Leica, Olympus, Nikon, FEI, ...

Install a plugin

Using the repositories or from the internet

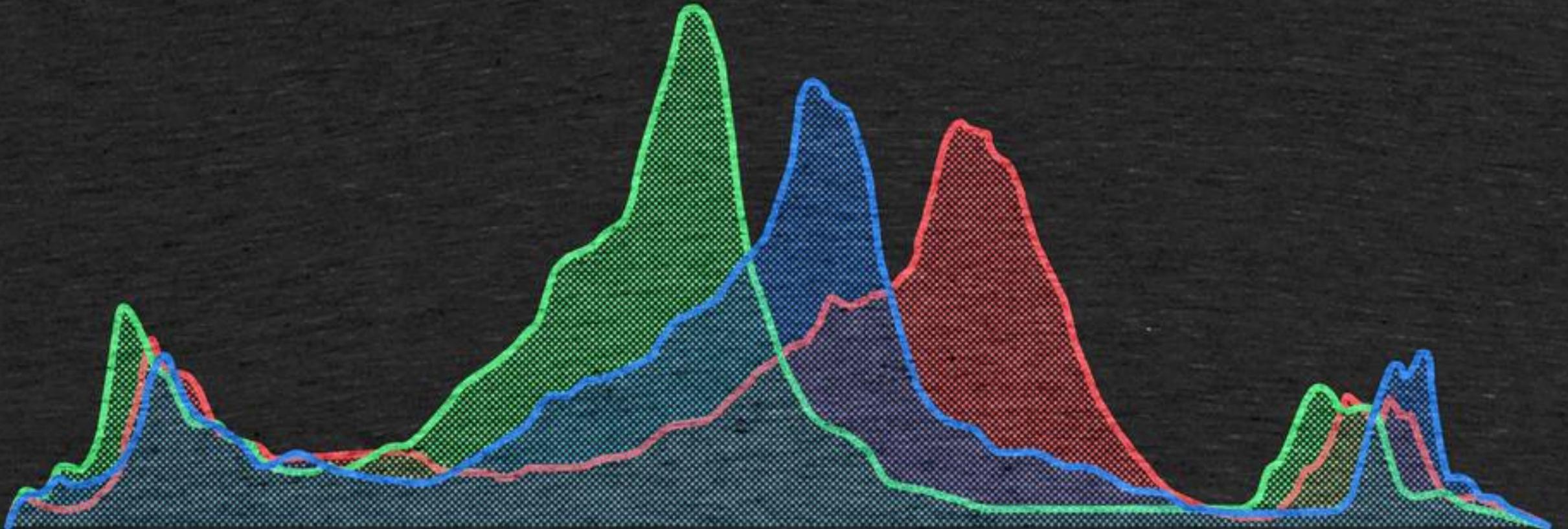
Raw import

Last resort, if everything else fails
Typically: *a priori* information about the file needed (dimension, bit depth)
Opens only the image data. No metadata.

Metadata is a love note to the future

Part III: Histograms

- Histograms of grayscale image
- Color images
- Histogram normalization and histogram equalization



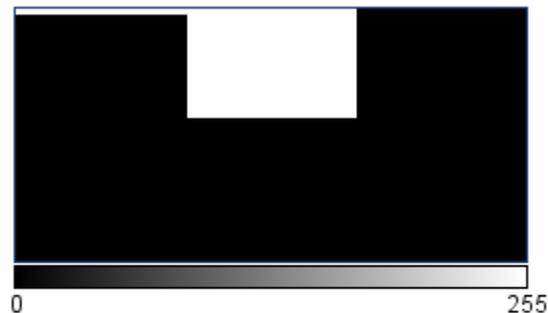
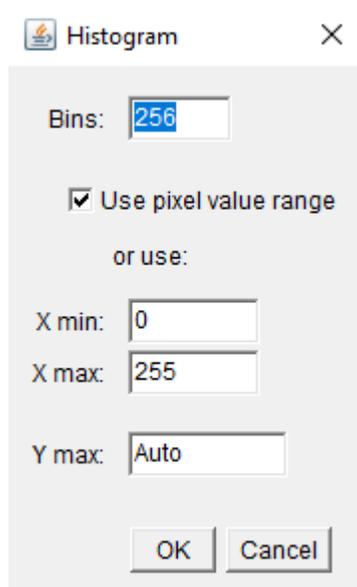
Histograms

EXERCISE 2

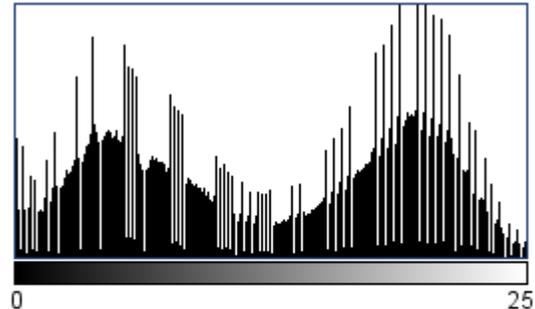
Open Example 1A and produce a histogram

Analyze > histogram (or CTRL+H)

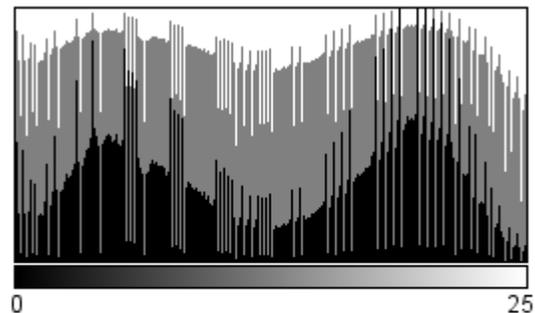
- What can you deduce from the histogram?
- Try: List, Copy, Log, Live
- Save the histogram itself to a TIFF?
- Also try CTRL+ALT+H



N: 247200
Mean: 125.901
StdDev: 73.247
Bins: 3
Value: 85
Min: 0
Max: 255
Mode: 170 (97279)
Bin Width: 85.333
Count: 55188



N: 247200
Mean: 125.901
StdDev: 73.247
Value: 101
Min: 0
Max: 255
Mode: 201 (3164)
Count: 129



N: 247200
Mean: 125.901
StdDev: 73.247
Value: 80
Min: 0
Max: 255
Mode: 201 (3164)
Count: 211

Histogram

a representation of the distribution of numerical data.

Pearson, K. (1895)

The intensity distribution of the image (= it plots the number of pixel for each intensity or tonal value)

Histograms

EXERCISE 2

Open Example 2A and 2B and look at the image

- Do you see a difference between the images?
- Check the histograms
- Do you see a difference between the histograms?

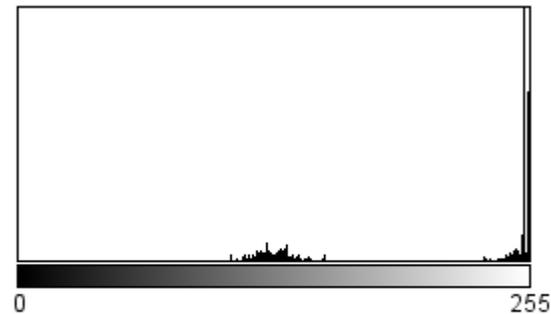
Histograms

EXERCISE 2

Open Example 2A and 2B and look at the image



N: 65536
Mean: 247.269
StdDev: 30.036
Value: 23
Min: 128
Max: 255
Mode: 255 (61272)
Count: 0



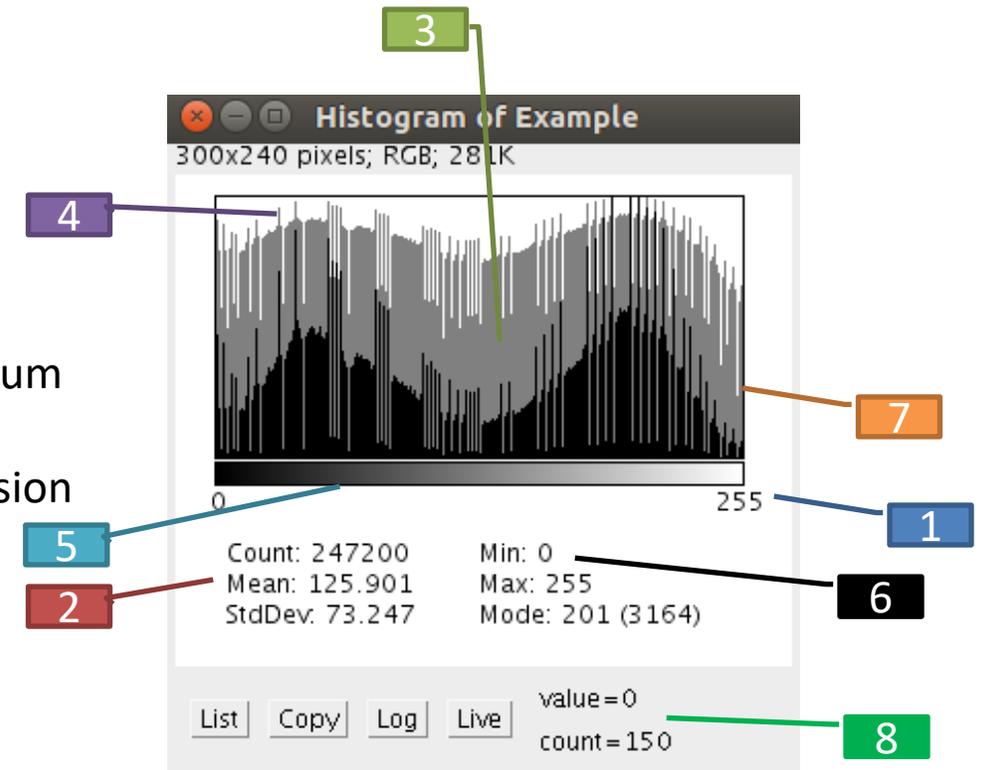
N: 65536
Mean: 245.270
StdDev: 29.543
Value: 134
Min: 88
Max: 255
Mode: 253 (57340)
Count: 219



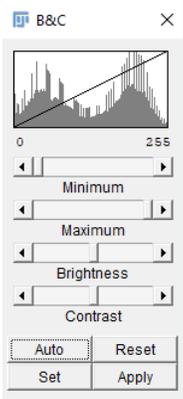
Histograms

What can you deduce from the histogram?

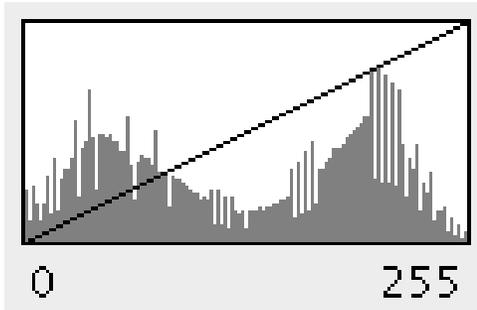
- 1. Bitdepth: 8 bit image (ranging from 0 to 255)
- 2. Pixel value distribution:
Mean pixel value, variance of the intensity,
Min, Max & modal
- 3. Type of distribution: e.g. bimodal, exponential, ...
- 4. Spikes: image normalization or equalization occurred (see later)
- 5. Contrast and Lookup table (see later): the range between maximum and minimum (in this case: 255)
- 6. Dynamic range: the number of distinct pixel values. Eg. Compression will affect the dynamic range
- 7. Overillumination effects
- 8. The intensity at each grayscale value



Histograms → Linear transfer function



Find the contrast brightness window (Image > adjust > brightness/contrast)



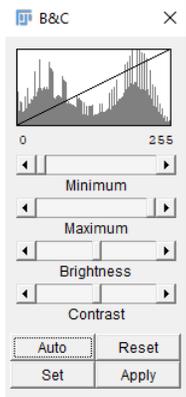
Histogram: Each image intensity value is mapped to its corresponding screen value
The linear transfer function is running from bottom left to top right

The linear transfer function

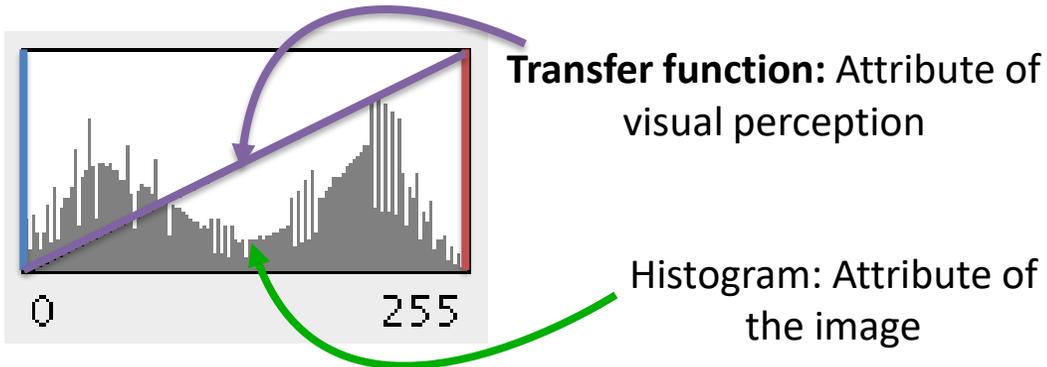
A linear transfer function is a tool applied over an image's histogram to manipulate contrast and brightness by mapping input pixel intensity values (X axis) to new output values (Y axis) using a linear relationship

The endpoints of the function determine what value is white, and what black

Histograms → Linear transfer function



Find the contrast brightness window (Image > adjust > brightness/contrast)

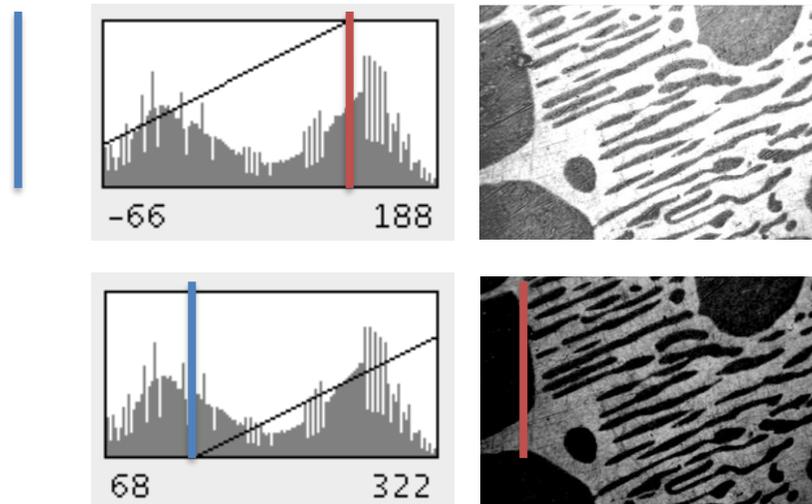


Pixels to the left of TF_{min} = black
Pixels to the right of TF_{max} = white
Pixels $> TF_{min}$ and $< TF_{max}$ = gray (linearly)

KEY: Changing the transfer function does not alter your data, only the way it is depicted

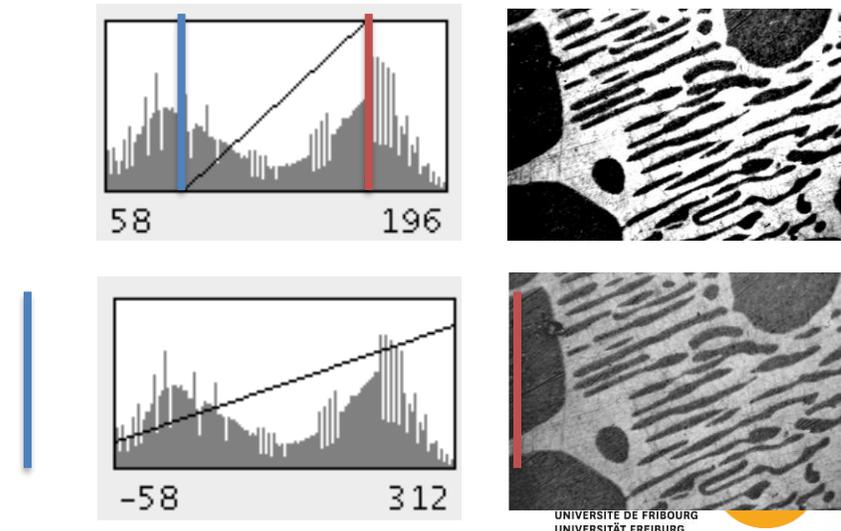
Brightness

The transfer function is shifted



Contrast

The transfer function is tilted



Histograms

EXERCISE 3

Load Example 3 – GrayscaleLUT, and read out the grayscale values in the ImageJ statusbar.

- Use (+ to zoom in, - to zoom out)
- Hoover with the mouse over the image and check the status bar in ImageJ
- Image > adjust > Brightness/Contrast (CTRL+Shift+C)
- Make a histogram of this image. How do you interpret the histogram?
- Play around with the Minimum and Maximum in the transfer function (Image > Adjust> Brightness/Contrast). Check the effect on the values.
- Do you delete information?



Histograms: other lookup tables (LUTs)

82	72	78	86	65	41
157	144	167	188	201	191
185	191	195	188	188	191
193	195	195	191	189	171
173	170	181	192	194	191
210	214	206	202	203	201
237	224	221	230	232	221
183	180	190	188	192	181
178	170	159	187	195	181
167	164	170	186	192	181
159	162				
180	172				
193	180				
167	184				
195	191				
183	188				
101	106				

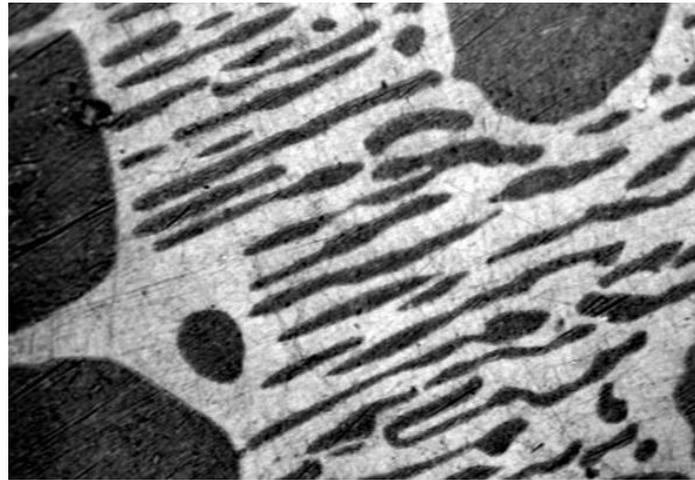
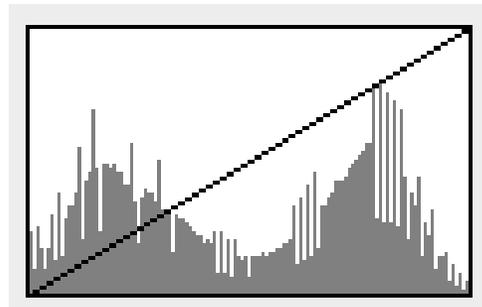
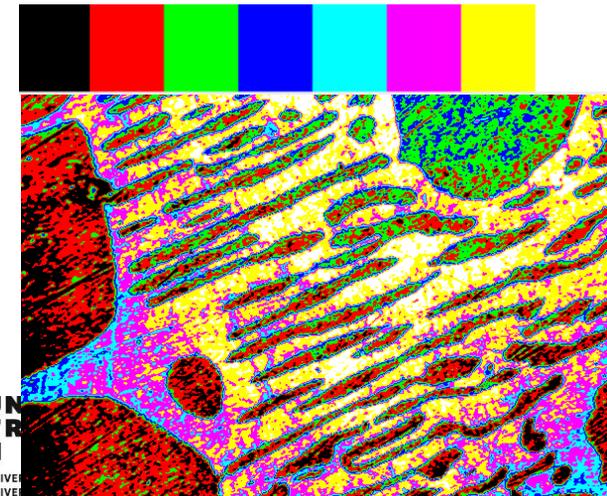
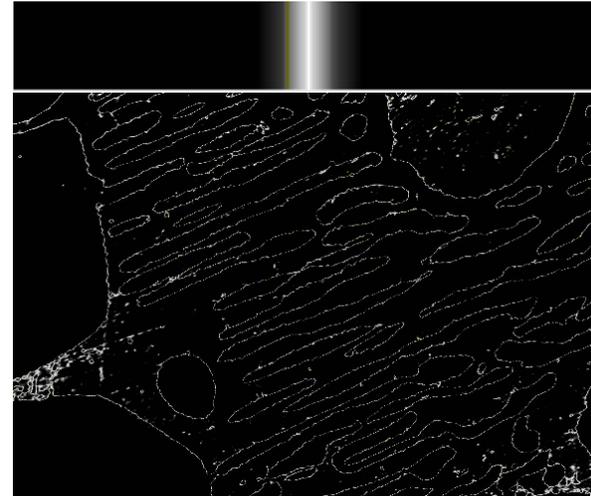
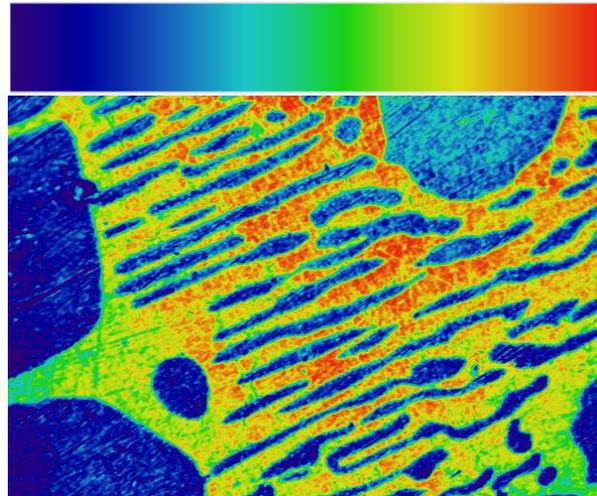
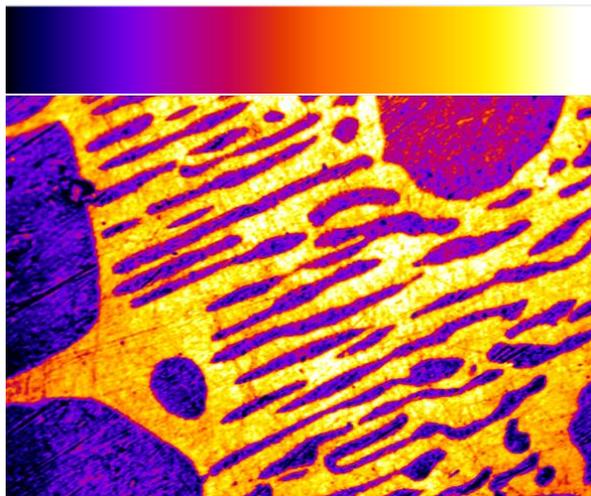
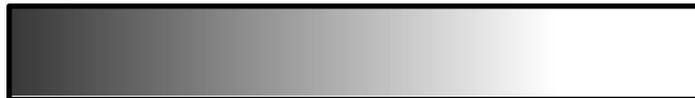


Image = 2D array of numbers

Grayscale = giving a graphical meaning to these Look-Up Tables.

But Grayscale is just one of these Look-Up tables!



Histograms: other lookup tables (LUTs)

EXERCISE 3

Load Example 3 or Example 1A and try different LUTs

- Image > Lookup Tables > ...
- Image > Color > Show LUT
- Change brightness and contrast with some exotic LUTs

- You can make your own LUT using Image > color > Edit LUT

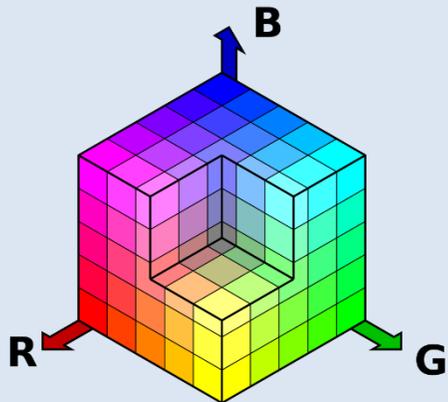
Color: the color model

A color model

is a method of describing a color. Color models can be represented as tuples of numbers, typically as three or four values or color components.

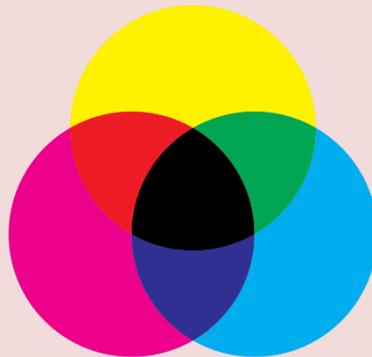
RGB

- 3 values: Red, green & blue
- Additive



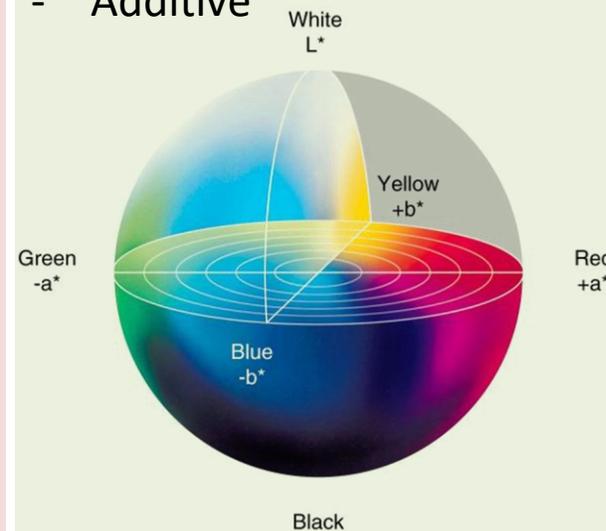
CMYK

- 4 values: Cyan, Magenta, Yellow and black
- Subtractive



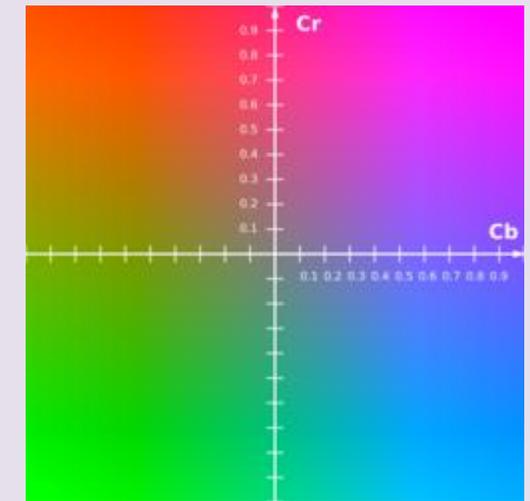
L*a*b* (CIELAB)

- 3 values: Lightness, a (green to red) and b (blue to yellow). Human vision based.
- Additive



YCbCr

- Y (Luminance/Brightness)
- Cb (Blue-difference chroma)
- Cr (Red-difference chroma)

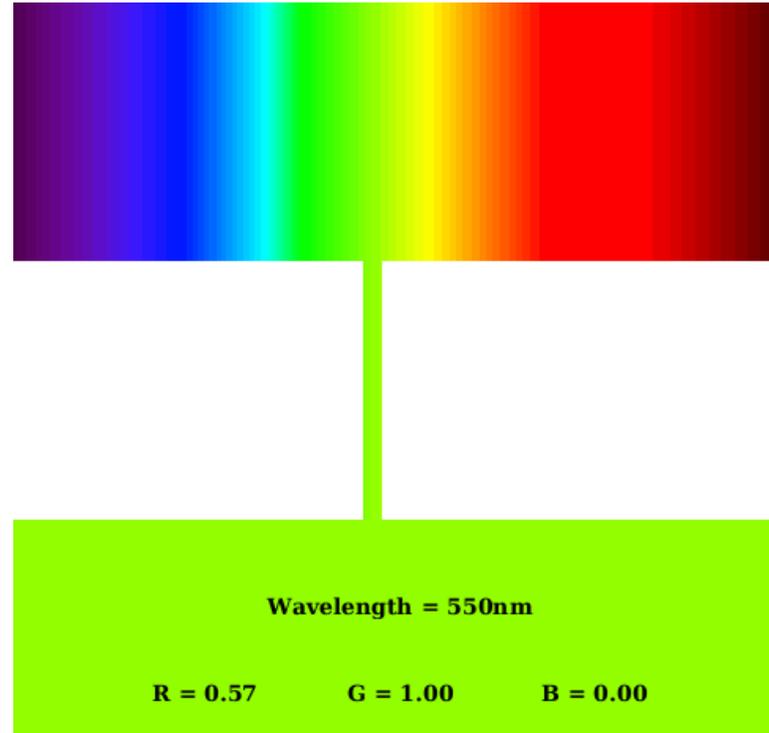
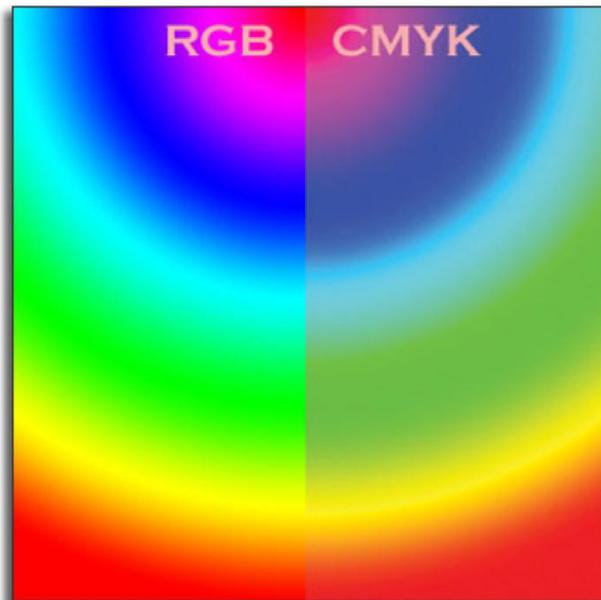


And many more: HSL, HSV, ...

Color: the color space

A color space

is a way of mapping real colors to the color model's particular values. the goal having reproducible, unambiguous representations of color – whether such representation entails an analog or a digital representation.



Based on human perception

CIE 1931 XY

CIEUVW

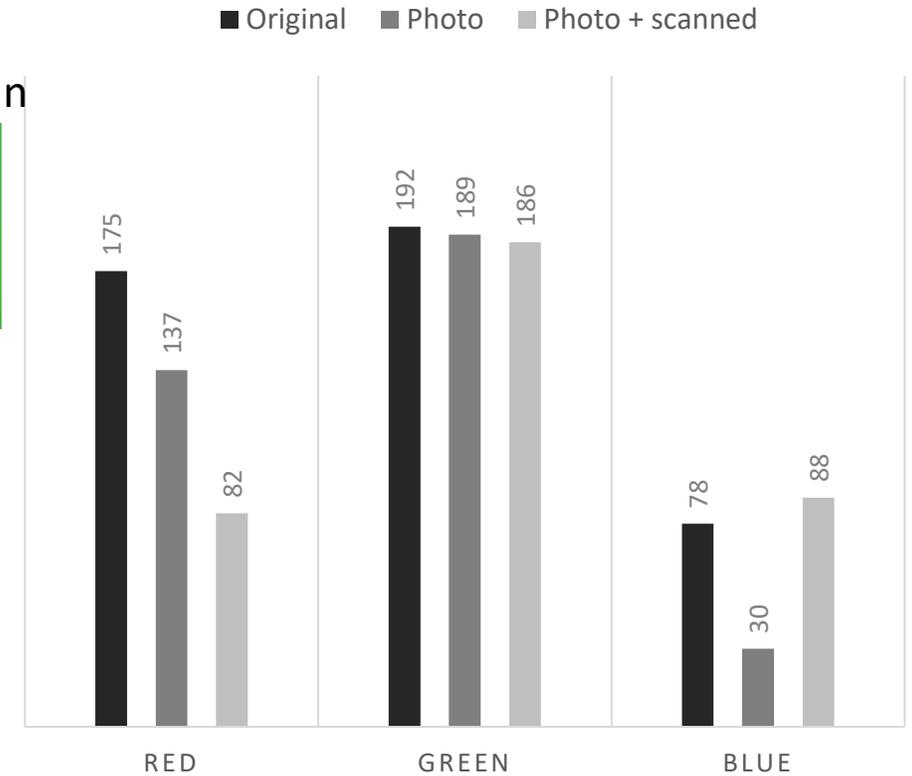
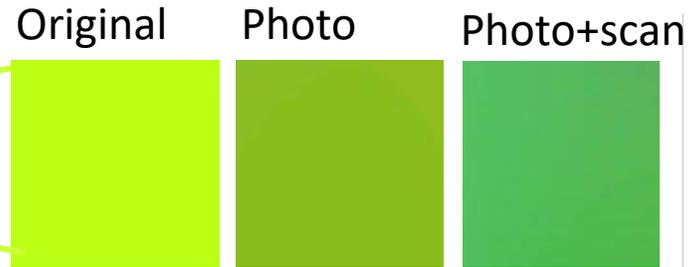
Uniform color spaces CIELUV

CIELAB

HSLuv

Color: the color space

Original



Photographed



Photographed + Scanned



Watch out with posters!

Color: the color gamut

The color gamut

The entire range of colors and tones achievable by an imaging system (eyes, printer, display)

Inside the diagram:

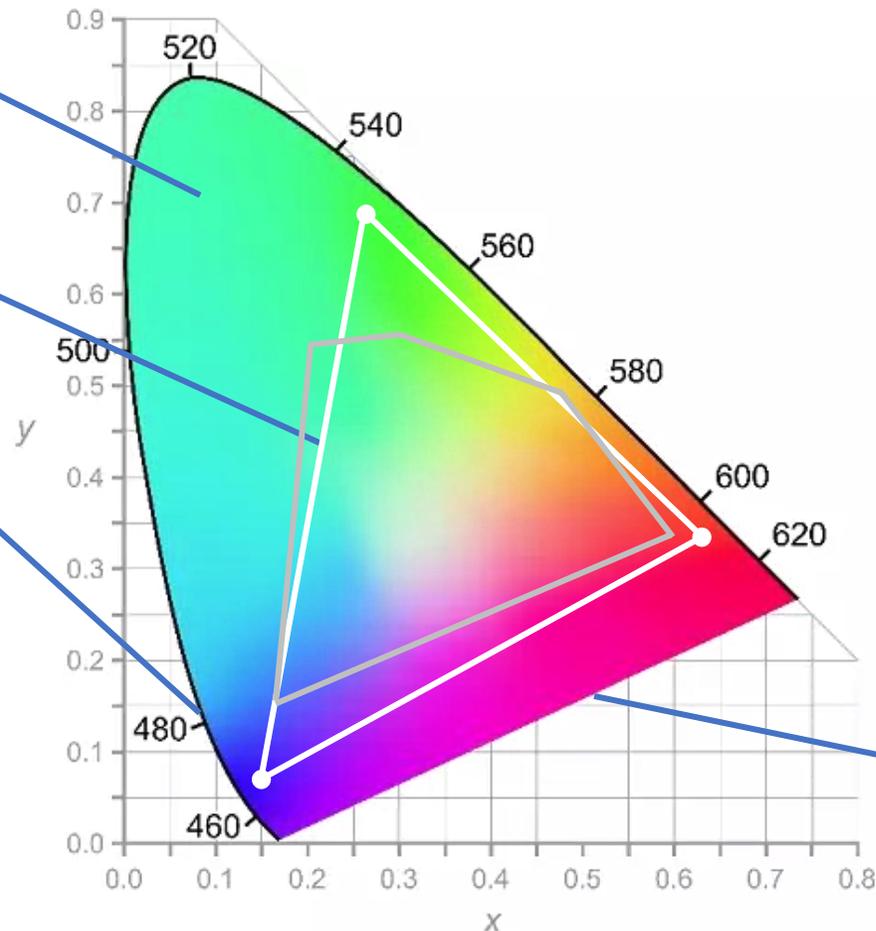
All colors visible to the average human eye (= CIELAB color space)

Color mixing:

The colors along any line between two points can be made by mixing the colors at the end points

Spectral locus:

The edge of the diagram represents pure monochromatic light (single wavelength, in nm)



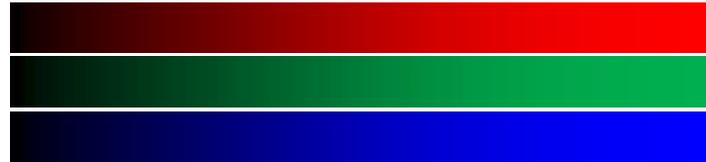
White: sRGB color space
Gray: color space defined by CMYK

Line of purples:

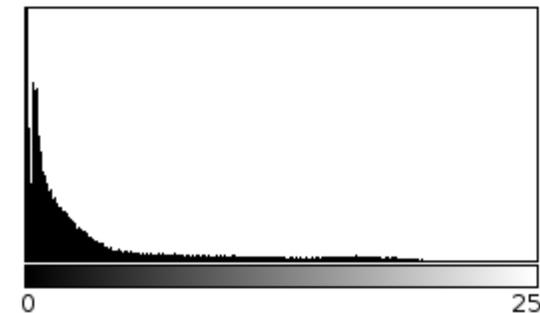
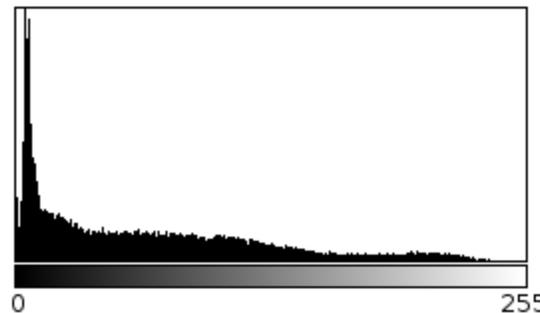
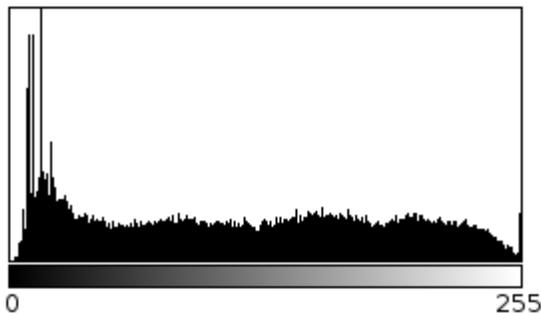
Non-monochromatic saturated colors (cannot be represented by a single wavelength)

Color images

- Color image = 3 grayscale images combined (=composite).
- with a red LUT
- With a green LUT
- With a blue LUT



They are also called RGB images, or 24 bit images (=3x8 bit)



Color images

- Color image = 3 grayscale images combined (composite).
- with a red LUT 
- With a green LUT 
- With a blue LUT 

They are also called RGB images, or 24 bit images (=3x8 bit)



Header

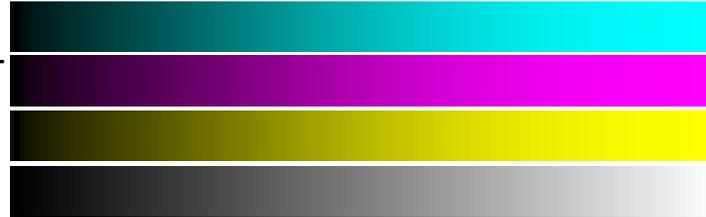
Image –
red data

Image –
green data

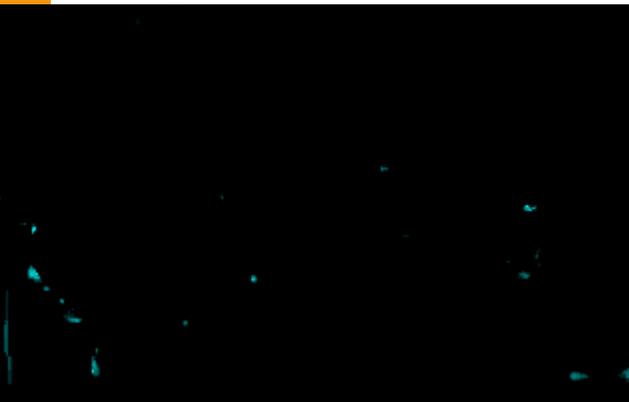
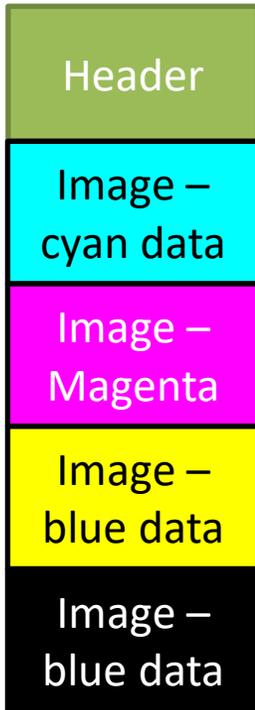
Image –
blue data

Color images

- Color image = 4 grayscale images combined (composite).
- with a cyan LUT
- With a magenta LUT
- With a yellow LUT
- With a black LUT

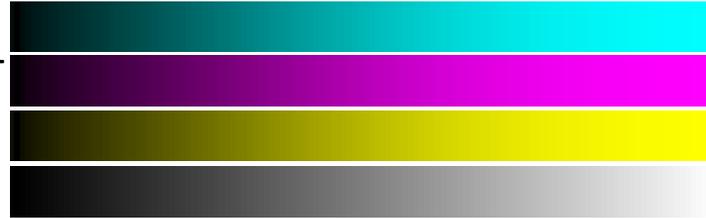


They are also called CMYK images, or 32 bit color images (=4x8 bit)

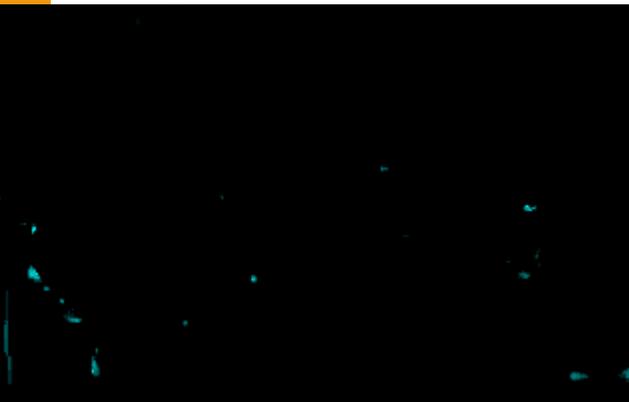
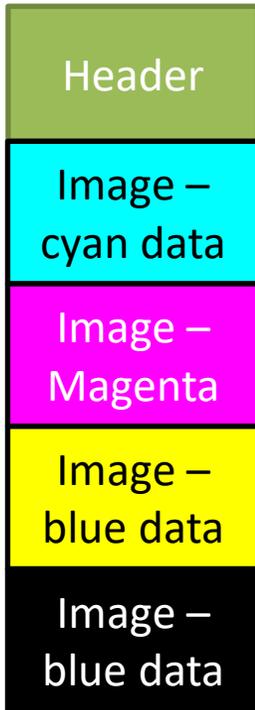


Color images

- Color image = 4 grayscale images combined (composite).
- with a cyan LUT
- With a magenta LUT
- With a yellow LUT
- With a black LUT



They are also called CMYK images, or 32 bit color images (=4x8 bit)



Measuring color

Natural: by eyes
(generally a very bad idea)



“green”



“blue”



“red”



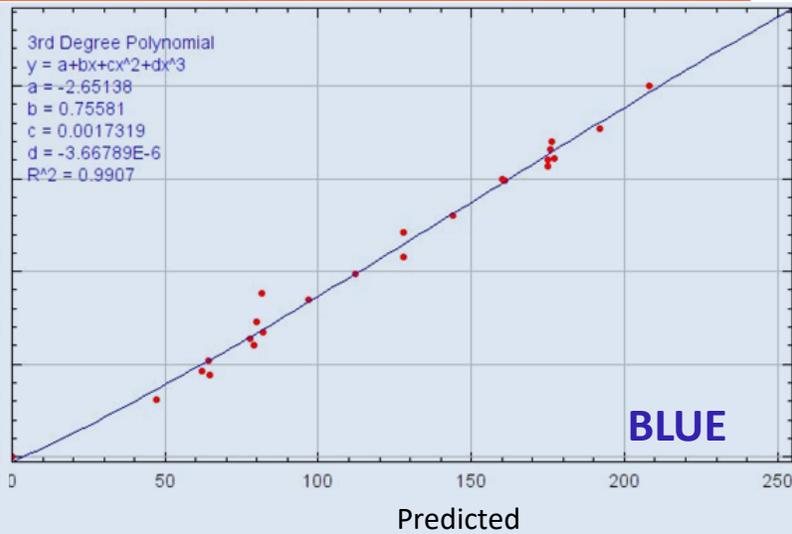
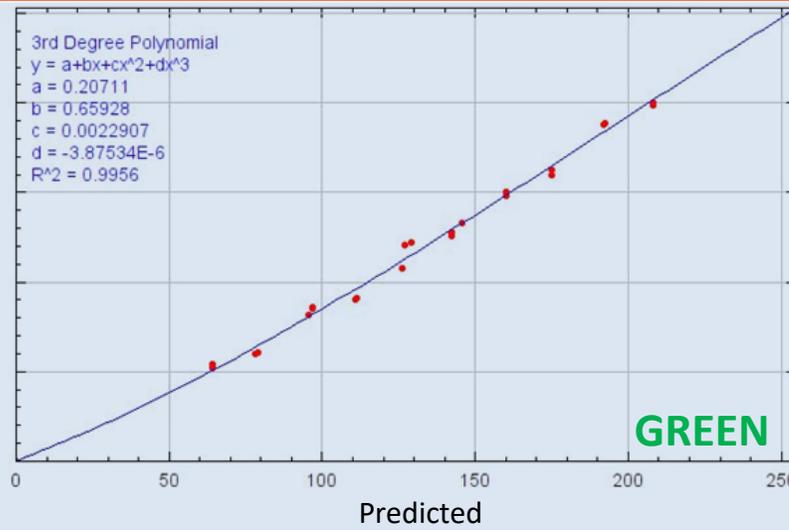
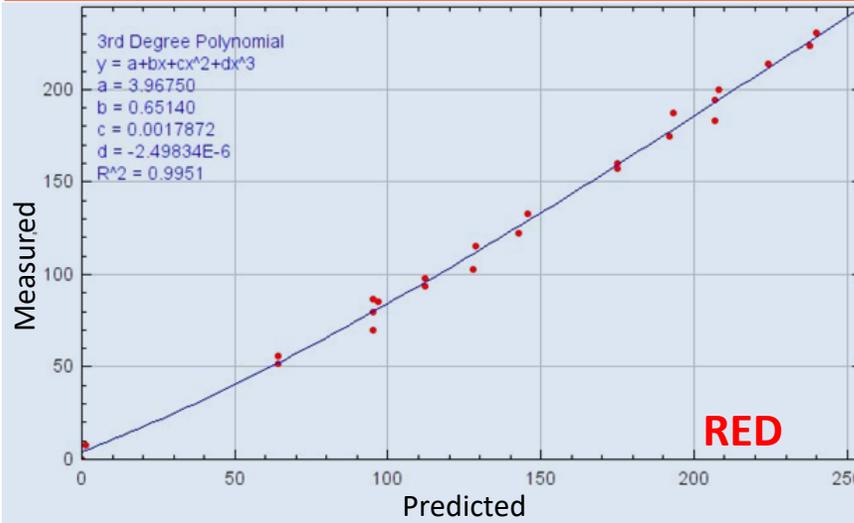
Still very tricky: using a commercial camera

→ Need for calibration using a color calibration chart



Note: non-linear shifts in Red, Green and Blue!
Output: RGB color space

Measuring color



Mismatch:
 RED: 15.5%
 GREEN: 12.3%
 BLUE: 16.1%

R=129 G=95 B=83	R=207 G=160 B=144	R=112 G=129 B=175	R=95 G=126 B=82	R=146 G=142 B=192	R=128 G=192 B=176
R=225 G=142 B=66	R=95 G=112 B=177	R=207 G=111 B=112	R=112 G=78 B=128	R=175 G=192 B=78	R=242 G=170 B=68
R=64 G=79 B=160	R=95 G=160 B=80	R=192 G=64 B=79	R=240 G=208 B=47	R=193 G=97 B=161	R=1 G=146 B=177
R=255 G=255 B=255	R=208 G=208 B=208	R=175 G=175 B=175	R=144 G=128 B=129	R=97 G=97 B=97	R=64 G=64 B=64

Color Checker Chart					
Dark Skin R=115 G= 82 B= 68	Light Skin R=194 G=150 B=130	Blue Sky R= 98 G=122 B=157	Foliage R= 87 G=108 B= 67	Blue Flower R=133 G=128 B=177	Bluish Green R=103 G=189 B=170
Orange R=214 G=126 B= 44	Purple Red R= 80 G= 91 B=166	Moderate Red R=193 G= 90 B= 99	Purple R= 94 G= 60 B=108	Yellow Green R=157 G=188 B= 64	Orange Yellow R=224 G=163 B= 46
Blue R= 56 G= 61 B=150	Green R= 70 G=148 B= 73	Red R=175 G= 54 B= 60	Yellow R=231 G=199 B= 31	Magenta R=187 G= 86 B=149	Cyan R= 8 G=133 B=161
White R=243 G=243 B=242	Neutral 8 R=200 G=200 B=200	Neutral 65 R=160 G=160 B=160	Neutral 5 R=122 G=122 B=121	Neutral 35 R= 85 G= 85 B= 85	Black R= 52 G= 52 B= 52

Measuring color

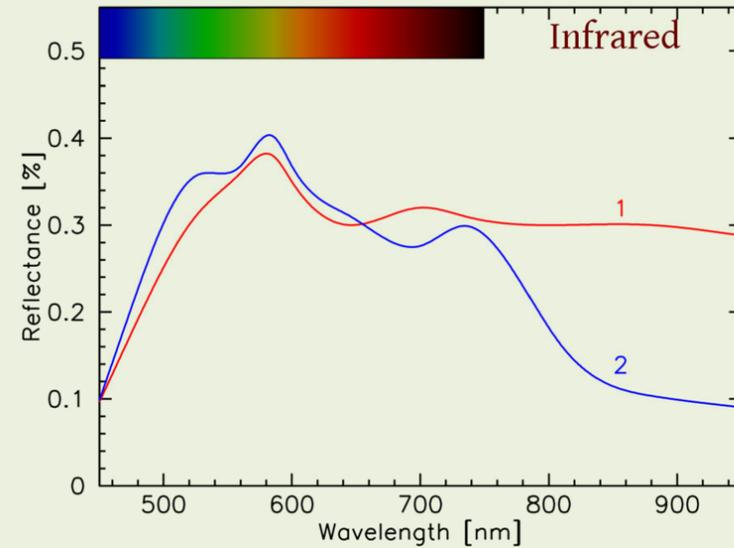
Standard: using a spectrophotometer



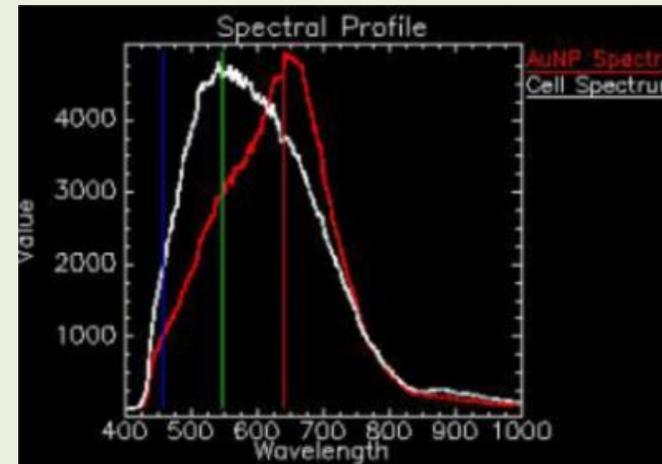
OUTPUT: L*a*b

Cytoviva: measuring reflectance spectra

Reflectance photospectrometer output



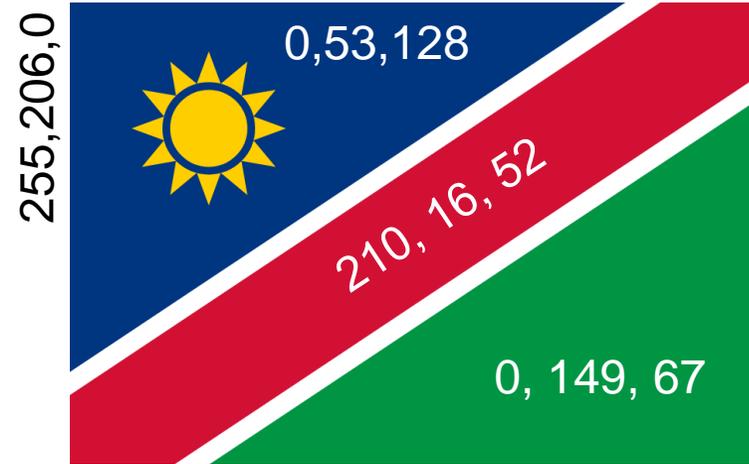
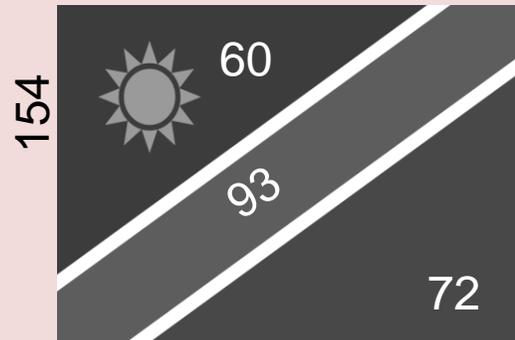
Cytoviva darkfield microscope output



Converting color to grayscale

Direct conversion

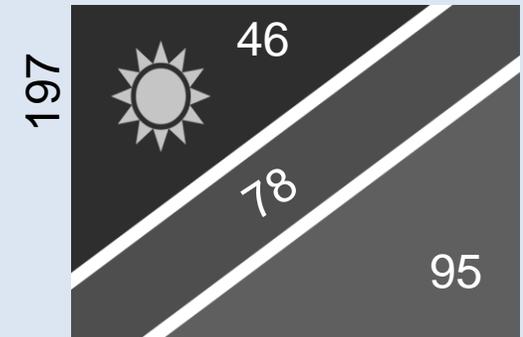
$$8bit = \frac{Red + Green + Blue}{3}$$



Edit > Options > Conversions...

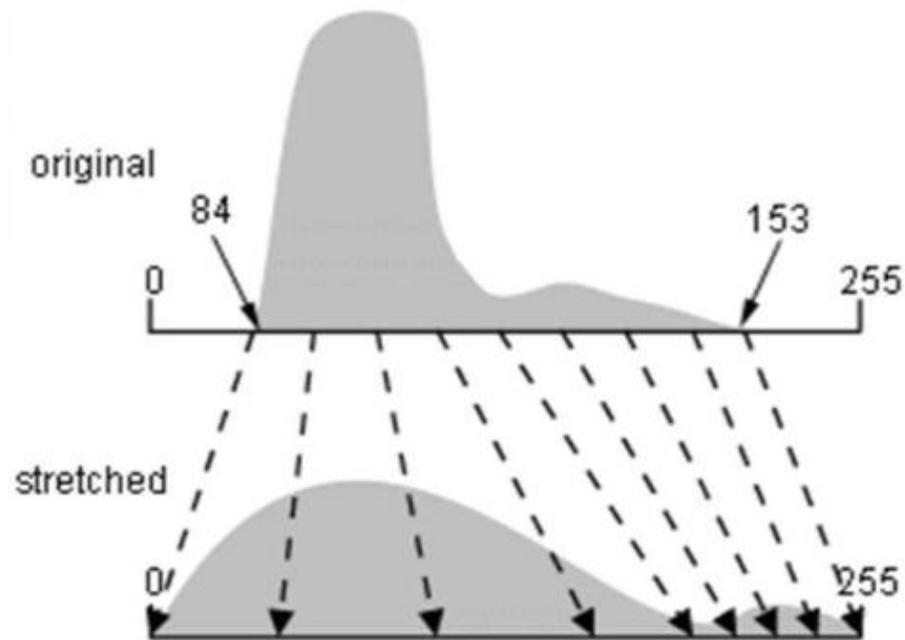
Weighted conversion

$$8bit = 29.9\%Red + 58.7\%Green + 11.4\%Blue$$



Histogram normalization, histogram equalization

Goal: to use the entire range of intensities in the histogram



Solutions:

- Histogram normalization (histogram stretching)
- Histogram equalization

WARNING:

1. GIGO
2. We will now **change the actual data**, not longer only the transfer function.

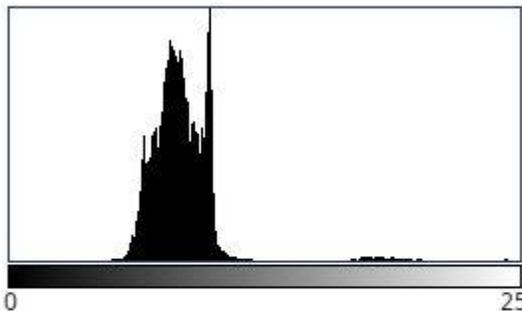
Histogram normalization



Contrast: the range between the maximum and minimum intensity in the image

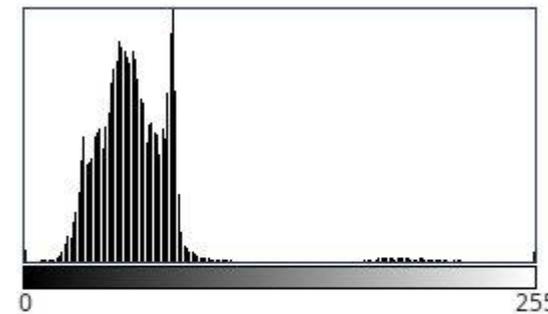
$$g(x, y) = \frac{f(x, y) - I_{min}}{I_{max} - I_{min}} \cdot 2^n$$

$$g(x, y) = \frac{f(x, y) - 7}{249 - 7} \cdot 256$$



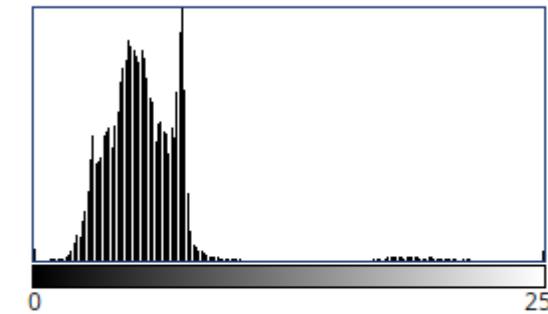
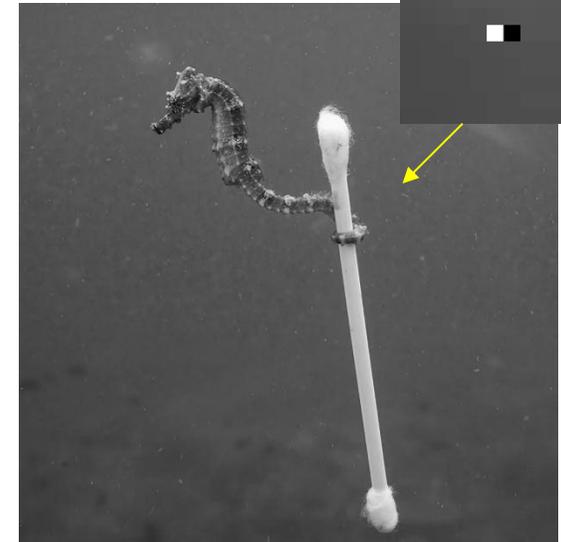
N: 1830560
Mean: 87.467
StdDev: 21.889
Value: 227

Min: 7
Max: 249
Mode: 100 (72268)
Count: 206



N: 1830560
Mean: 57.078
StdDev: 29.453
Value: 216

Min: 0
Max: 255
Mode: 74 (72268)
Count: 0



N: 1830560
Mean: 57.078
StdDev: 29.453

Min: 0
Max: 255
Mode: 74 (72268)

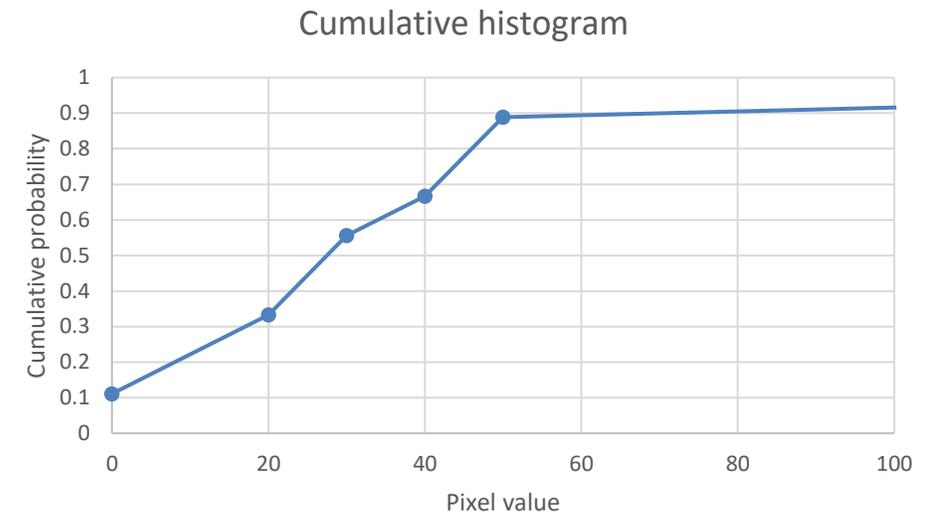
Histogram normalization is very prone to noise!

Histogram equalization

Idea: Values > Probability mass function > Cumulative distribution function > normalize

5	20	40
20	30	30
50	100	50

Values	Occurrence	PMF*	CDF*
5	1	0.111	0.111
20	2	0.222	0.333
30	2	0.222	0.555
40	1	0.111	0.666
50	2	0.222	0.888
100	1	0.111	1
	Total = 9	Total = 1	



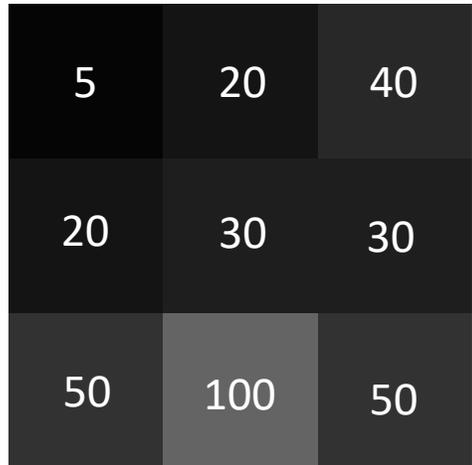
= the normalization of the cumulative histogram of the image

*PMF = Probability mass function

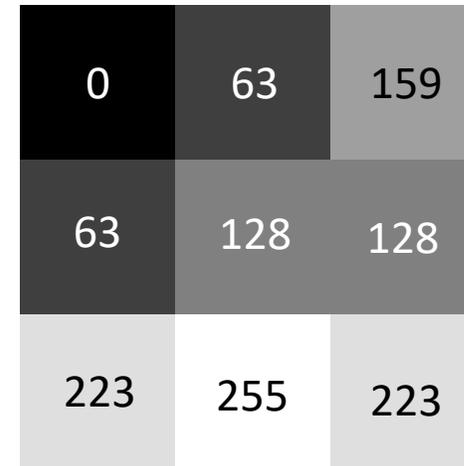
*CDF = Cumulative distribution function = cumulative histogram

Histogram equalization

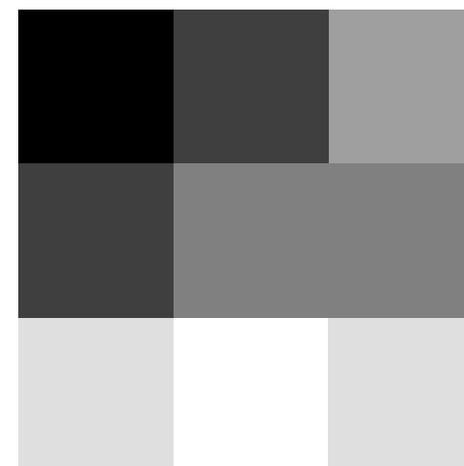
Note to the not-so-mathematically-inclined:
This is exactly the same formula as 3 slides back, but now we use the newly created cumulative histogram instead of the histogram



Values	CDF*	CDF norm	8 bit
5	0.111	0	0
20	0.333	0.25	63
30	0.555	0.5	128
40	0.666	0.625	159
50	0.888	0.875	223
100	1	1	255



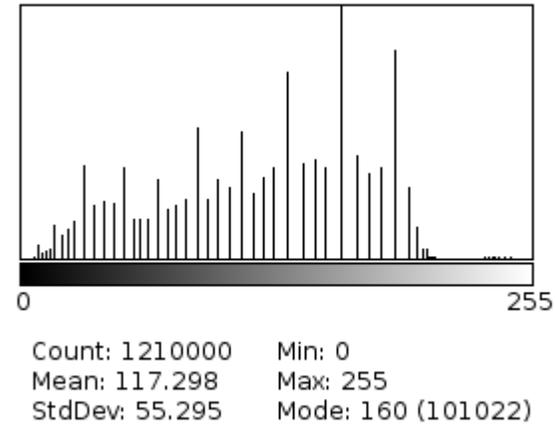
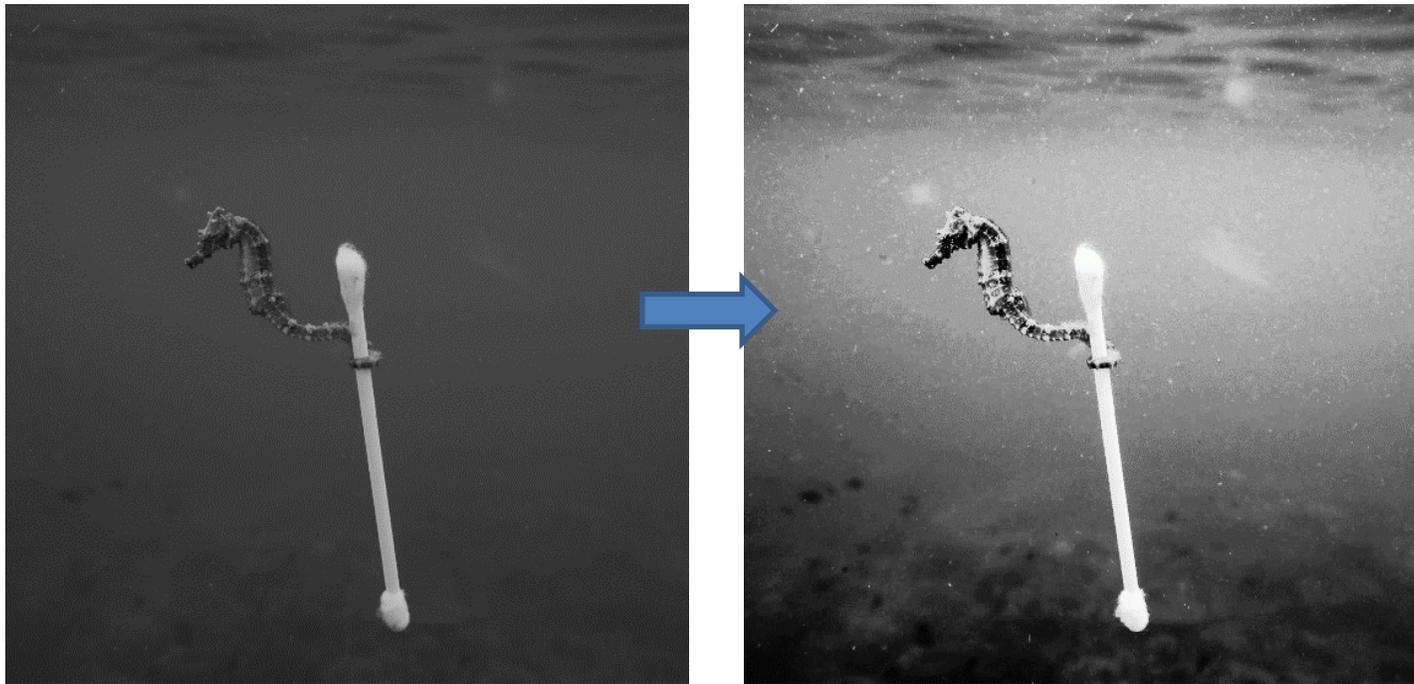
= the normalization of the cumulative histogram of the image



*PMF = Probability mass function

*CDF = Cumulative distribution function = cumulative histogram

Histogram equalization: attention!



Watch out with Histogram equalization!

- Unrealistic artefacts in 8 bit (grayscale images)
- Image gradients in images with low depth (it will further reduce dynamic range)
- Undesired effects when histogram is not continuous

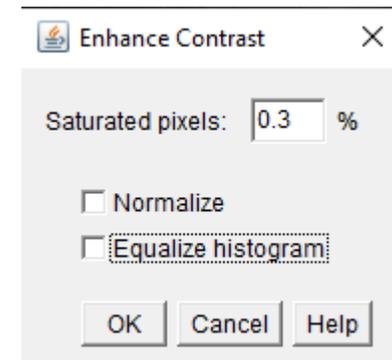
Histogram equalization: attention!

EXERCISE 5

Open Example 5B.tif and try the histogram normalization and histogram equalization

Process > Enhance contrast...

Amount of Saturated pixels: default = 0.3% (but play with it)
(usually, you want this very low)

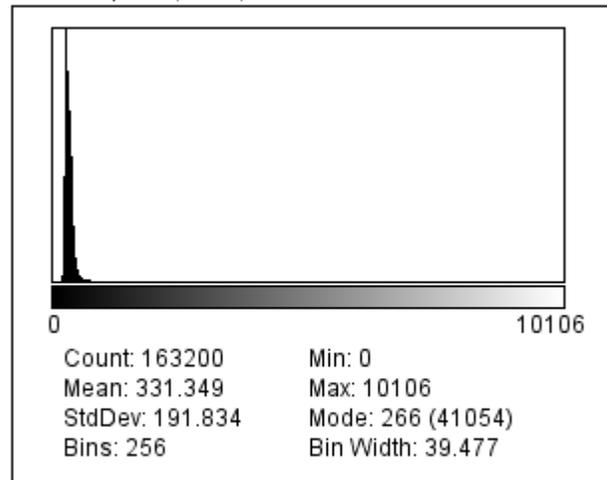


Histogram equalization: attention!

Example (Galaxy M51)

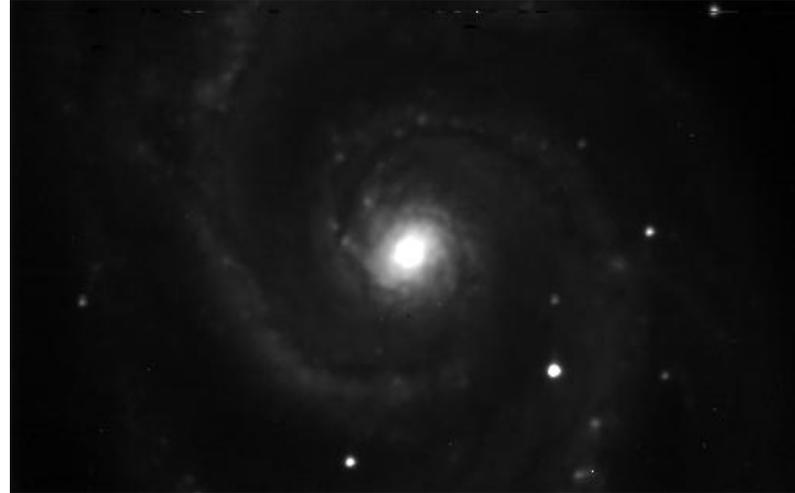


300x240 pixels; RGB; 281K

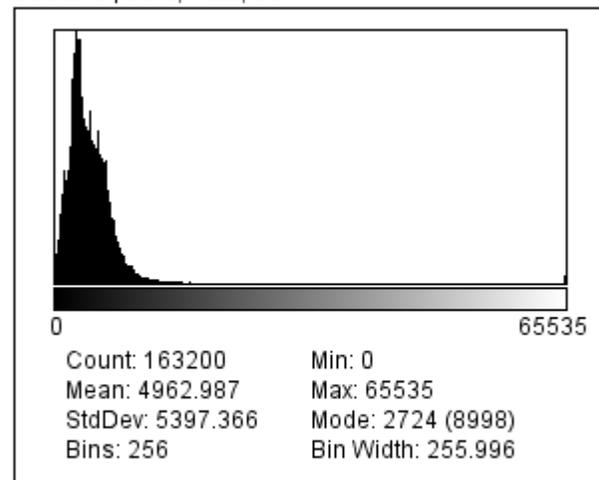


[List](#) [Copy](#) [Log](#) [Live](#) value=4342.422
count=0

Histogram normalization

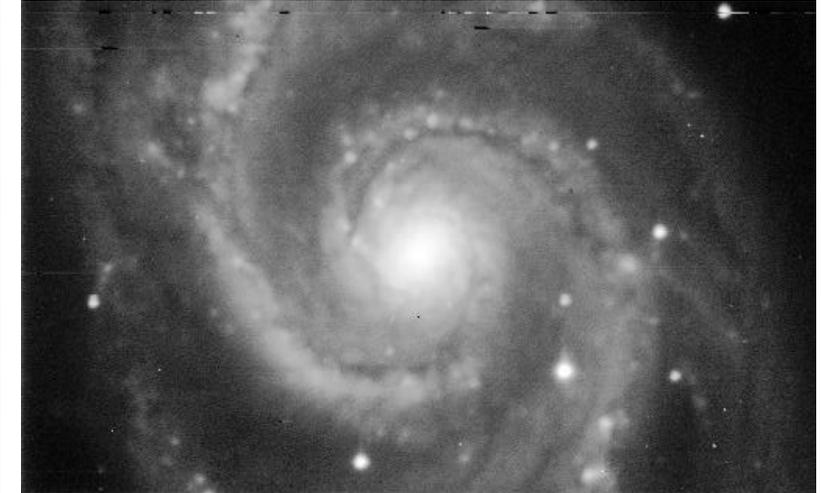


300x240 pixels; RGB; 281K

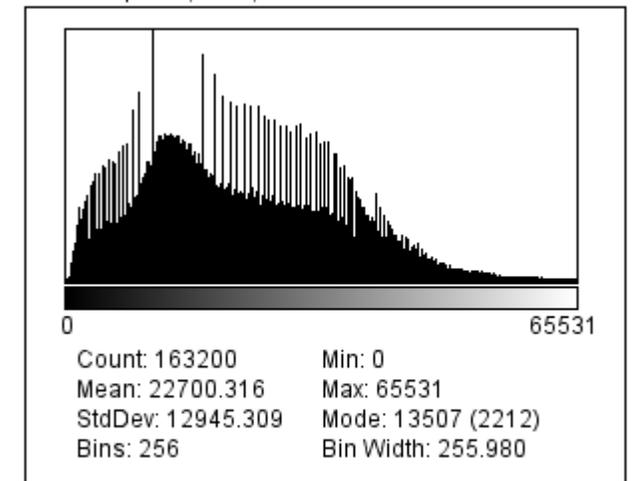


[List](#) [Copy](#) [Log](#) [Live](#)

Histogram equalization



300x240 pixels; RGB; 281K



[List](#) [Copy](#) [Log](#) [Live](#) value=6655.492
count=1147

Part IV: Overlays and preparation for publication

- Overlays
- Annotations and scale bars
- Preparing figures for publications
- Inkscape
- FigureJ

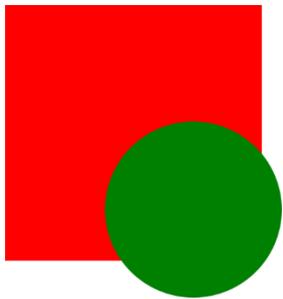


Overlays

Overlays are **vector graphics**: non-active, non-destructive selections displayed 'over' the rastered graphics data.

Overlay selections are

- Mathematically-defined **paths** (=not rastered), not affected by scaling, i.e., do not become pixelated.
- Overlays are **saved in the header** (e.g. of tif images), and do not need to be saved externally.
- Examples: Scalebars, annotations, ...



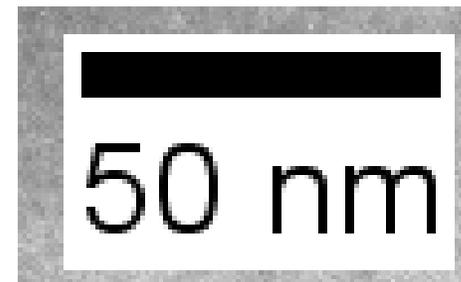
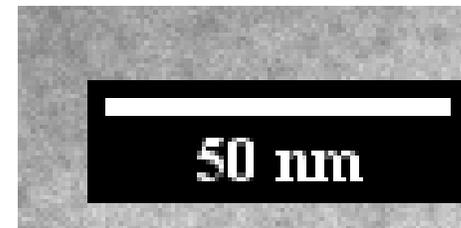
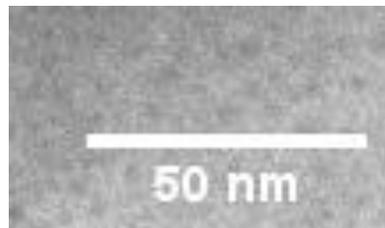
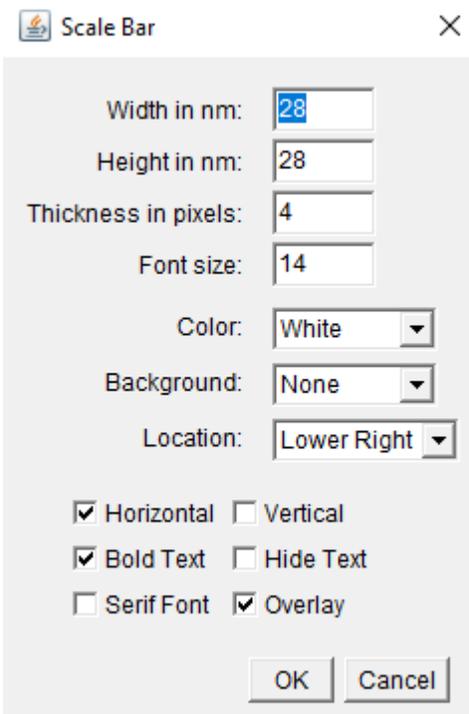
Scale bars

EXERCISE 6

Add a scale bar to Example 1D (the .ser file)

- Open Example 1D (Plugins > Input-Output > TIA reader)
- Check the info in the header: CTRL + i (or Image > Show info ...)
- Analyze > Tools > Scale bar...

Width: 8.3029 microns (2048)
Height: 8.3029 microns (2048)
Size: 8MB
Resolution: 246.6609 pixels per microns
Pixel size: 0.0041x0.0041 microns²



Scale bars

EXERCISE 6

Try to repeat for Example 4 – Manual scale. Retrieve the scale and add a scale bar to a cropped version of Example 4

- Open Example 4
- Image > show info...

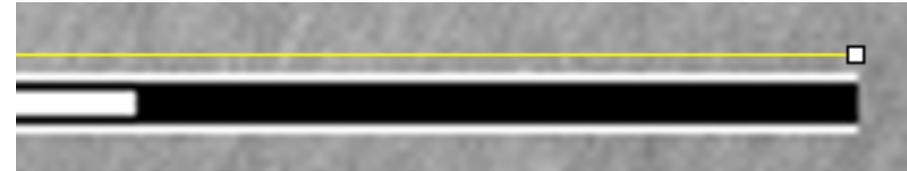
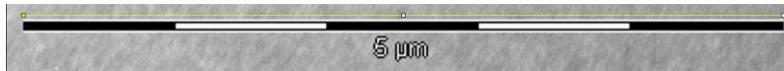
Scale bars

EXERCISE 6

Try to repeat for Example 4 – Manual scale. Retrieve the scale and add a scale bar to a cropped version of Example 4

- Open Example 4
- Image > show info... The pixel size is NOT calibrated →
- But! There is a scale bar burnt into the rastered image.
- Draw a line exactly the length of the burnt-in scalebar

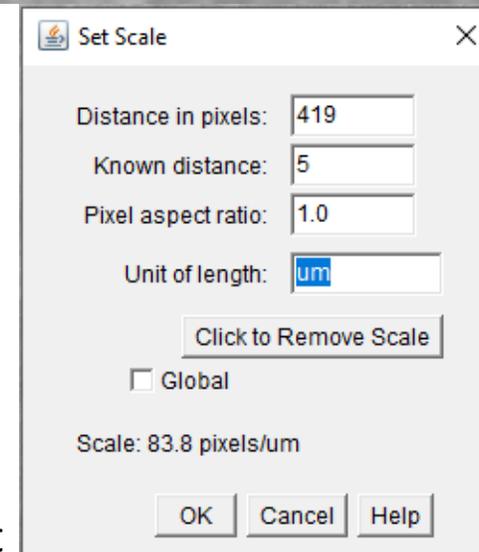
Width: 2970 pixels
Height: 2100 pixels
Size: 5.9MB
Pixel size: 1x1 pixel²



- Use Analyze > Set Scale
- The length of the line (in pixels) will be automatically filled in
- Enter the known distance (1) and the unit (um, there is no μ)
- (Global will set this scale to all open images)
- Click OK. Your image is now calibrated (Check the info!)

Now add the overlay scalebar

- Analyze > Tools > Scale bar...
- Note: the burnt-in scale cannot be removed, only cropped out



Annotations

EXERCISE 7

Open Example 4 – Manual scale and annotate the image

Arrows, lines, text

- Image > annotate > Arrow..

Accepting the annotation

- Press CTRL+B to make the annotation in overlay (recommended)
- Press CTRL+D to draw the annotation in the image (burn in: convert vector to pixels)

Note: ImageJ/FIJI is not great for annotations

Illustrator, Affinity designer, Inkscape, CorelDraw, ... are superior tools for vector graphics design.

Annotations

EXERCISE 7

Open Example 4- Manual scale and annotate the image

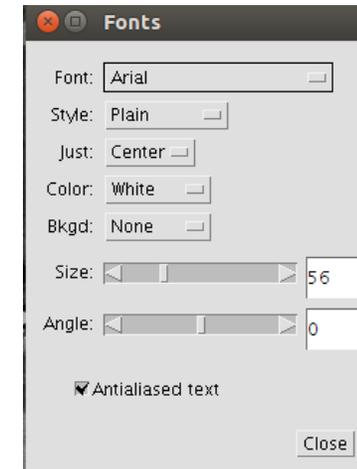
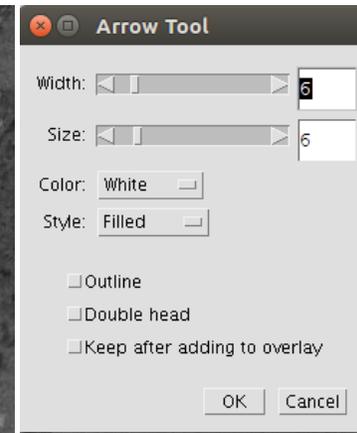
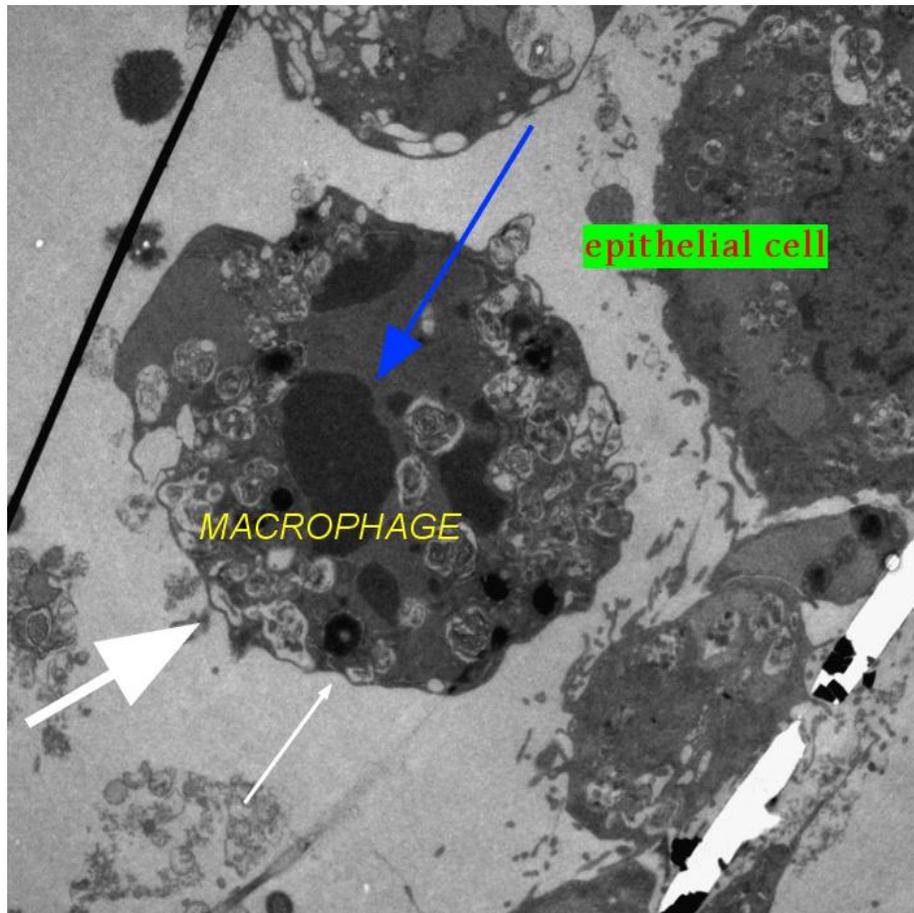
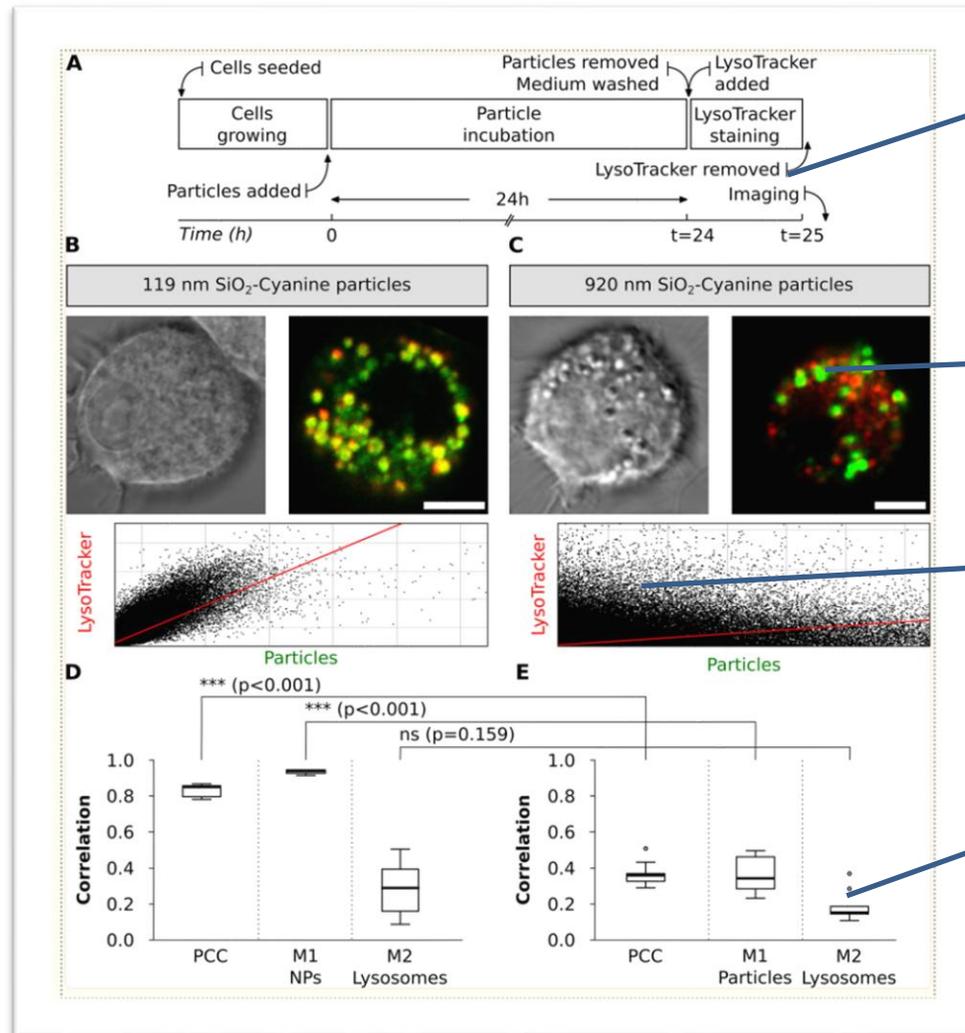


Image > Overlay > List elements

Type	X	Y	Width	Height	Color	Fill	LWidth
Straight Line	396	1070	139	189	white	none	6
Straight Line	58	977	273	164	white	none	16
Straight Line	602	204	245	400	#0036ff	none	8
Text	320	795	422	71	yellow	none	0
Text	923	360	378	68	red	green	0

Preparing for publication - semantics

Some semantics:



Line art

Image

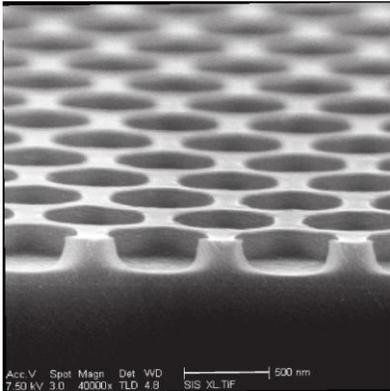
Graph or plot

Line art

Figure

Preparing for publication - DPI

Images: from camera to printer: PPI and DPI



Dots per inch (DPI)

is a measure of spatial dot density in printing, in particular the number of individual dots that can be placed in a line within the span of 1 inch (2.54 cm).

Points per inch (PPI)

is the same but with concerning electronic displays. Not relevant here.

Rule of thumb: what resolution is required to publish my data?

Images: at least 300 DPI, but 600 DPI has been requested

Line art: higher: 1200 DPI, but I have seen 2400 DPI requested

Graphs: 600 DPI and up

Low resolution

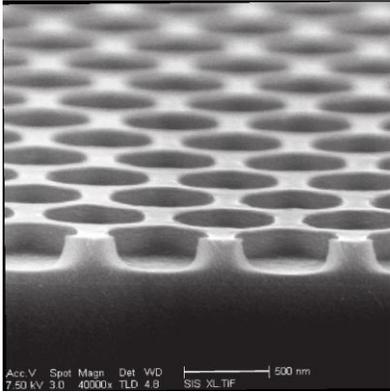


High resolution



Preparing for publication - Example

Images: from camera to printer: PPI and DPI



Info:

1 inch = 2.54 cm

Expected Image width in print: 10 cm

Expected printed resolution: 300 DPI

Question: how many pixels wide does my digital picture have to be?

Calculation:

$$\text{Width (printer)} = \frac{10 \text{ cm}}{2.54 \frac{\text{cm}}{\text{inch}}} = 3.937 \text{ inch}$$

$$\text{at 300 dots per inch} = 3.9737 \text{ inch} \times 300 \frac{\text{dots}}{\text{inch}} = 1181 \text{ dots}$$

Result:

The image must be 1181 pixels wide to get a 10 cm wide figure at 300 DPI

Dots per inch (DPI)

is a measure of spatial dot density in printing, in particular the number of individual dots that can be placed in a line within the span of 1 inch (2.54 cm).

Resizing and cropping

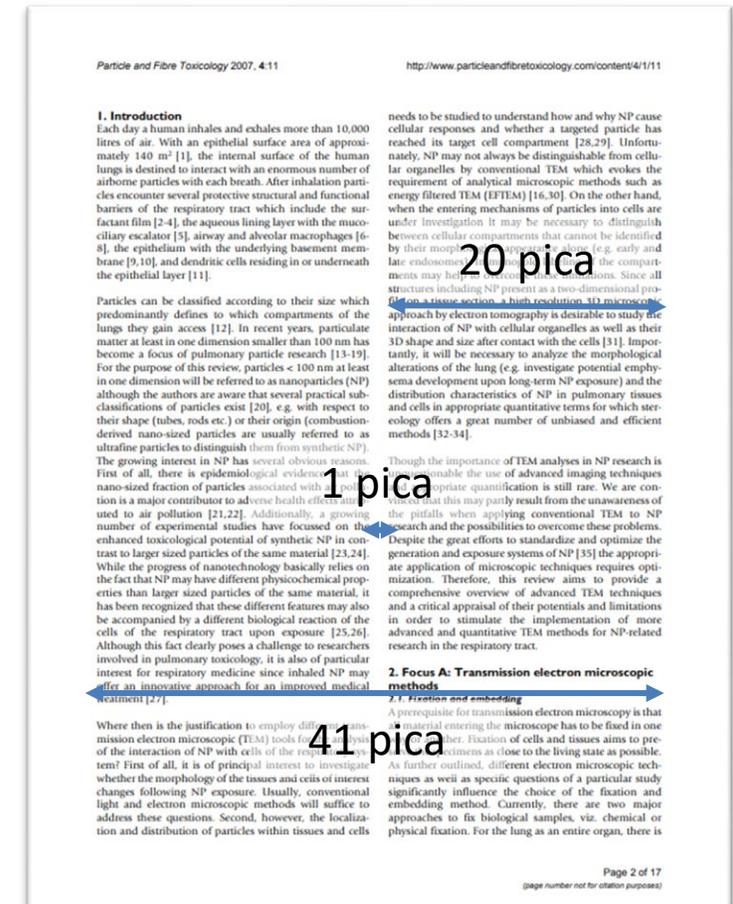
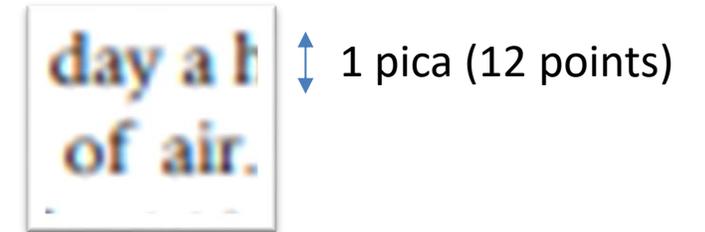
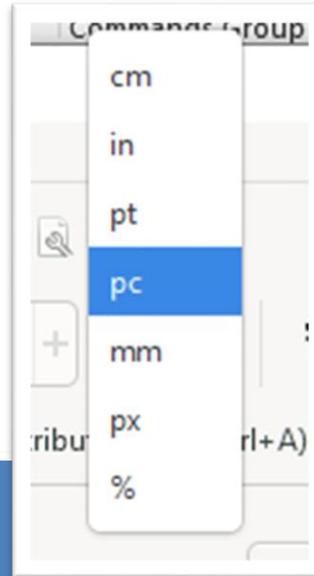
What happens with my image during resizing? Am I allowed to crop an image?

==> See lecture 2 Advanced Image processing

Preparing for publication - pica

Publishers and printers work often with PICA (pc) and POINTS (pt).

- 12 points (pt) 1 pica (pc)
- 1 pc 1/6 of an inch
- 1 pc 4.23333 mm
- 1 pt 0.35277 mm
- Text height 1 pica (12 points)
- Text width (page) 41 pica
- Column width (2 / page) 20 pica
- Central space 1 pica



Info

Expected Image width in print: 20 pc

Expected printed resolution: 300 DPI

Question: how many pixels wide does my digital picture have to be?

Calculation:

$20 \text{ pc} = 20/6 \text{ inch} = 3.333 \text{ inch} (= 8.333 \text{ cm})$

Width (printer) = 3.333 inch

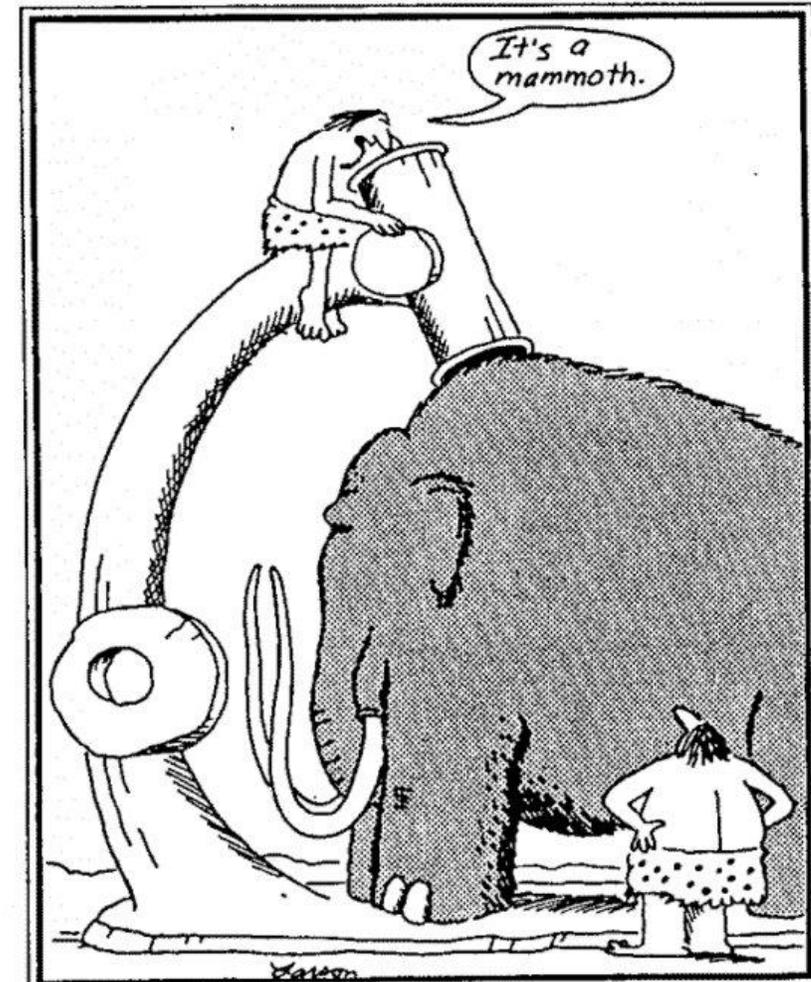
at 300 dots per inch = $3.333 \text{ inch} \times 300 \frac{\text{dots}}{\text{inch}} = 1000 \text{ dots}$

Result:

The image must be 1000 pixels wide to get a columnwide figure at 300 DPI

Finished!

✓ Congratulations,
You finished Part I, Basics!



Early microscope