Introduction to Digital Image Analysis

Part I: Digital Images

Kurt Thorn

NIC

UCSF

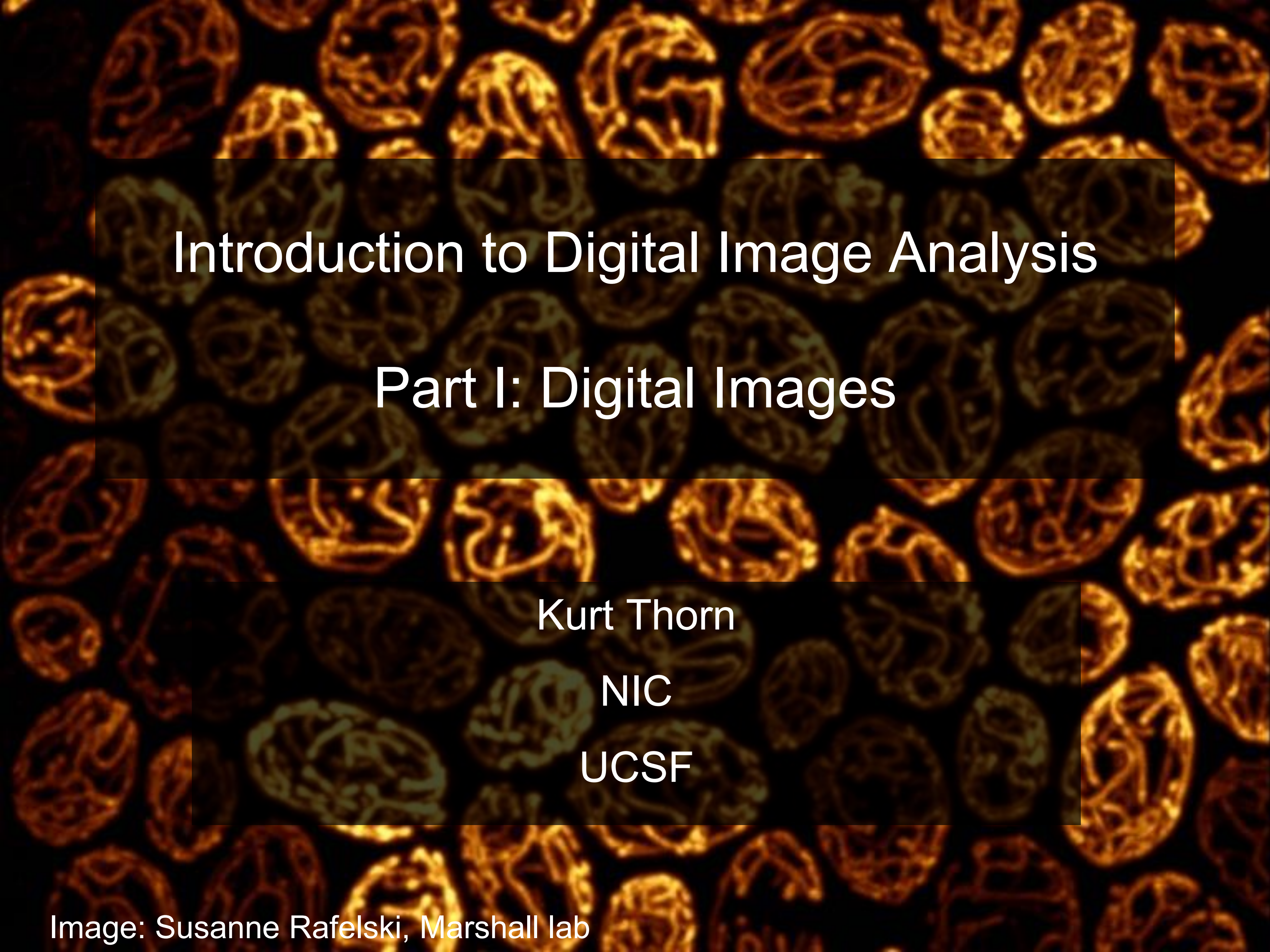
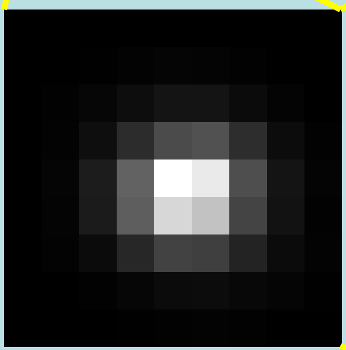


Image: Susanne Rafelski, Marshall lab

What is a digital Image?

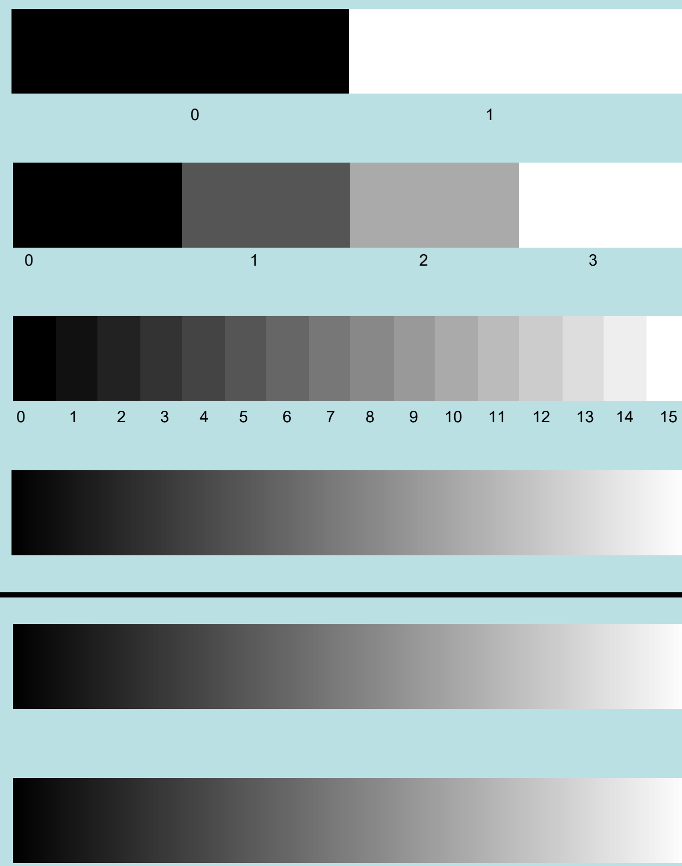
Many measurements of light intensity



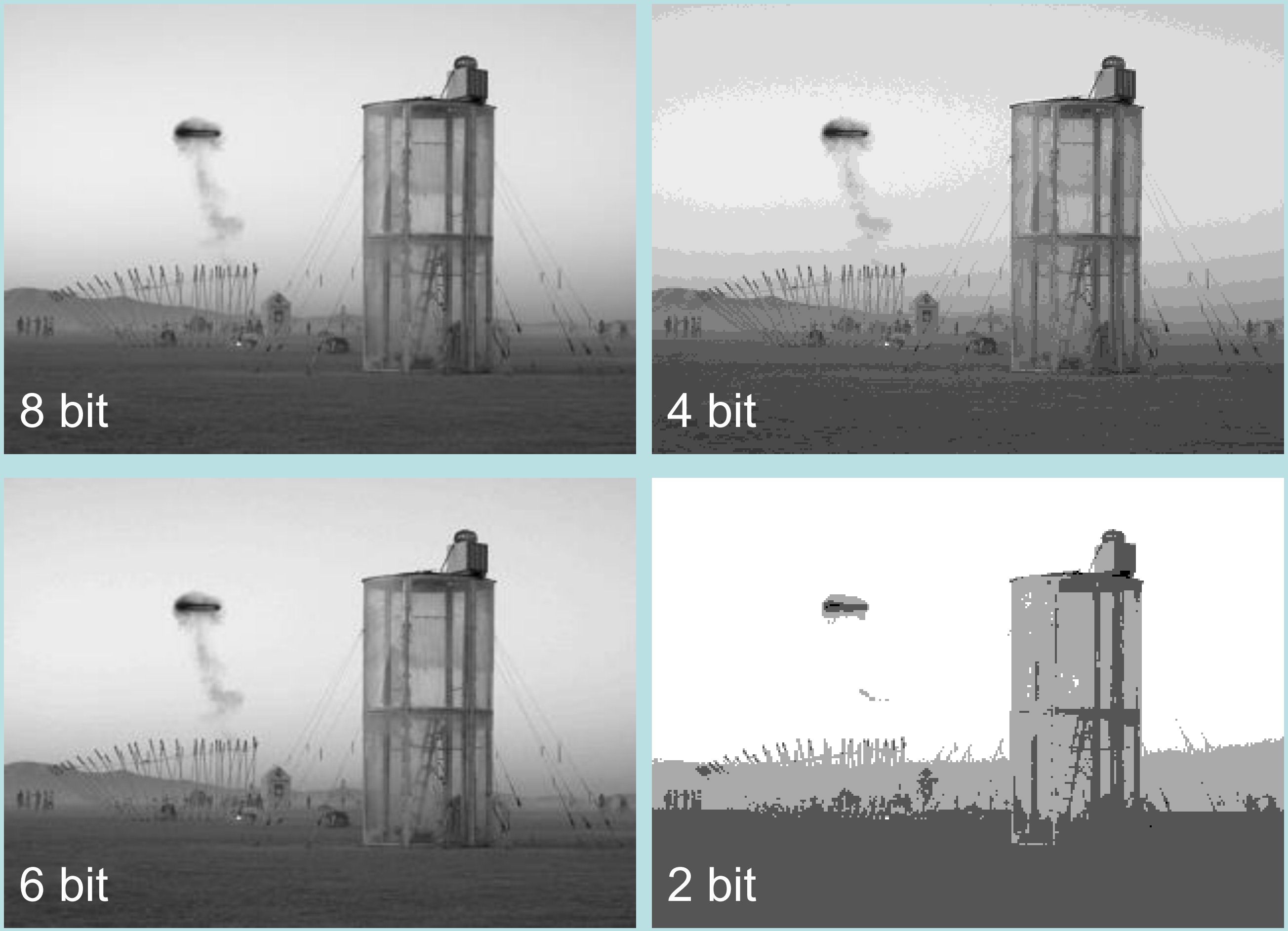
0	0	0	0	0	0	0	0	0
0	0	1	3	5	4	2	0	0
0	2	6	13	20	20	11	4	0
0	3	14	44	75	81	45	12	2
0	5	28	98	255	234	78	20	4
0	4	27	94	215	194	68	18	2
0	3	11	39	66	63	35	11	3
0	0	2	6	11	12	8	5	1
0	0	0	1	2	3	2	0	0

Bit depth and dynamic range

# of bits	range	
1	2	
2	4	
4	16	
8	256	1-byte
12	4096	2-bytes
16	65536	



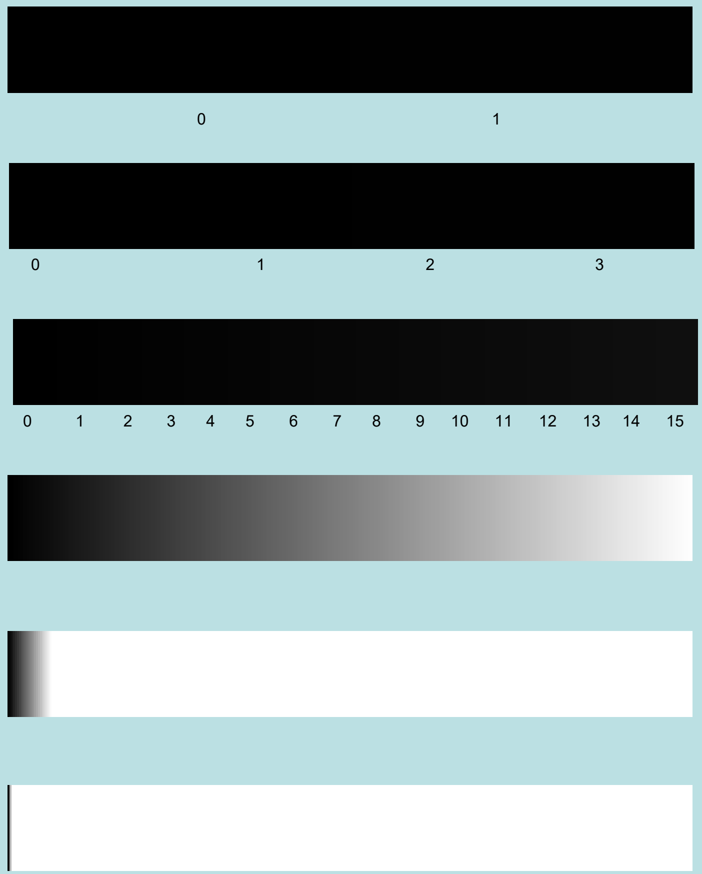
Bit depth and dynamic range



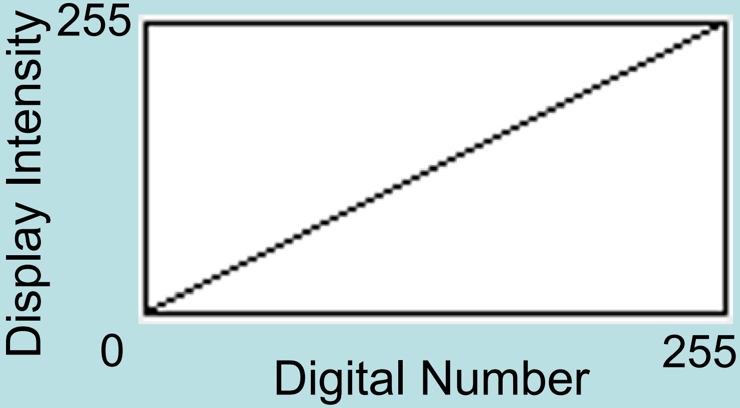
Converting bit-depth

Your monitor is an 8-bit display

# of bits	range
1	2
2	4
4	16
8	256
12	4096
16	65536

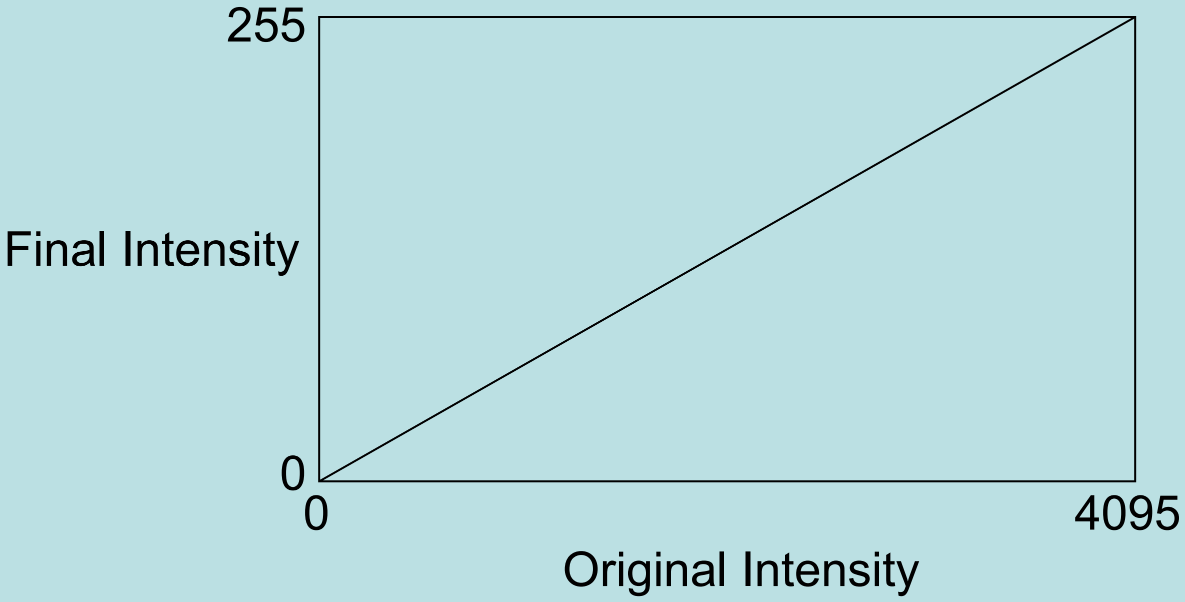


Mapping values onto display



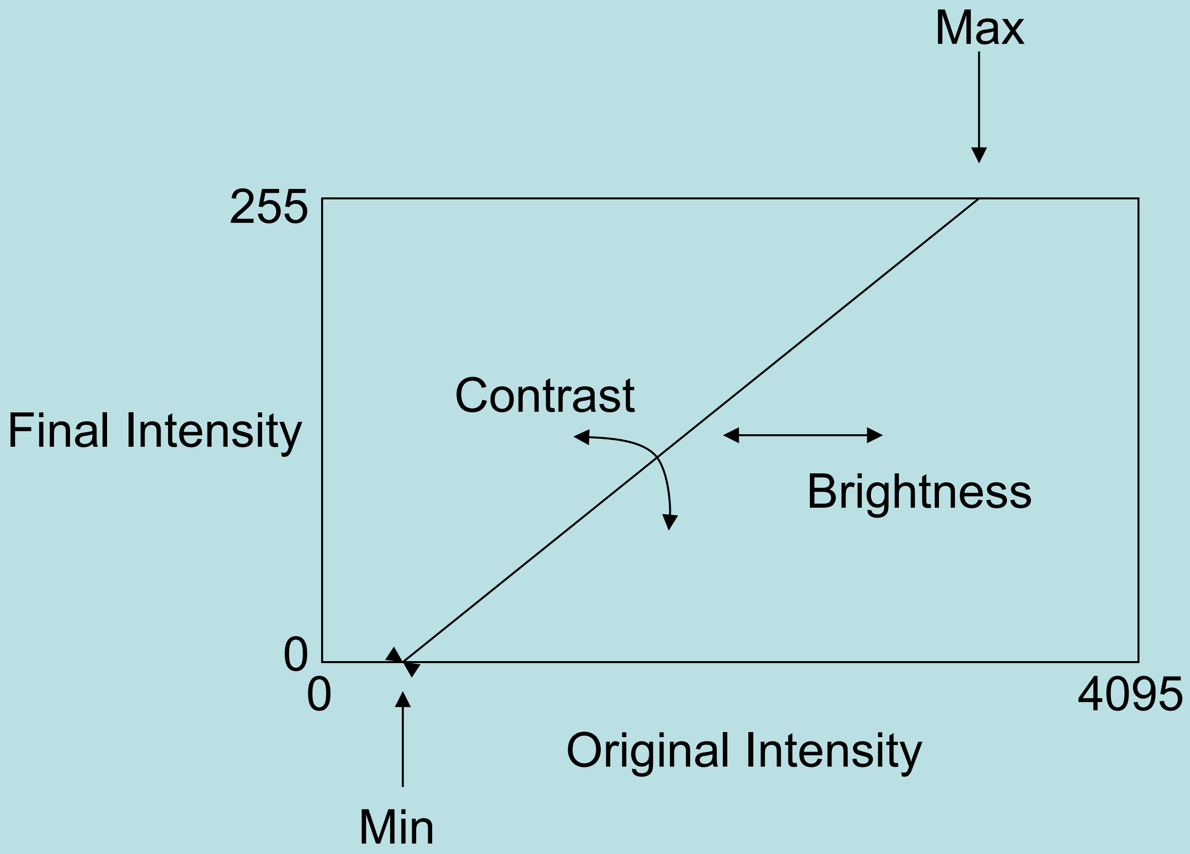
Intensity scaling

- Computer screens are 8-bit
- Publishers also want 8-bit files

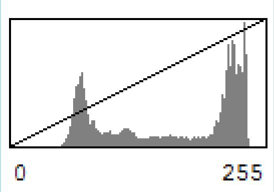


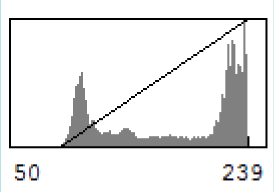
You lose information in this process – values 4080-4095 all end up as 255

Intensity scaling



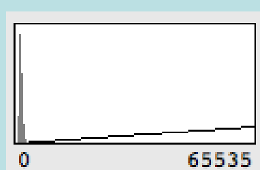
Brightness / Contrast adjustment



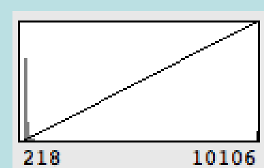


Brightness/contrast

Be aware of how your software scales your image



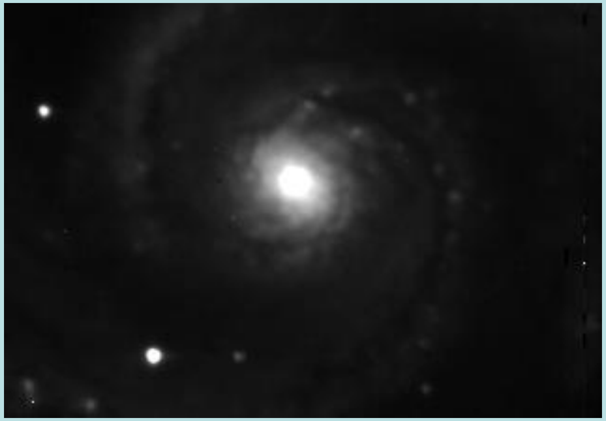
Full Range

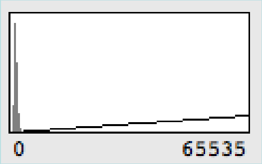


Auto-Scale

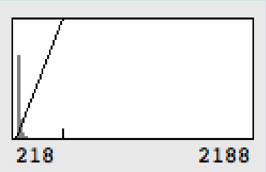
Brightness/contrast

Be aware of how your software scales your image



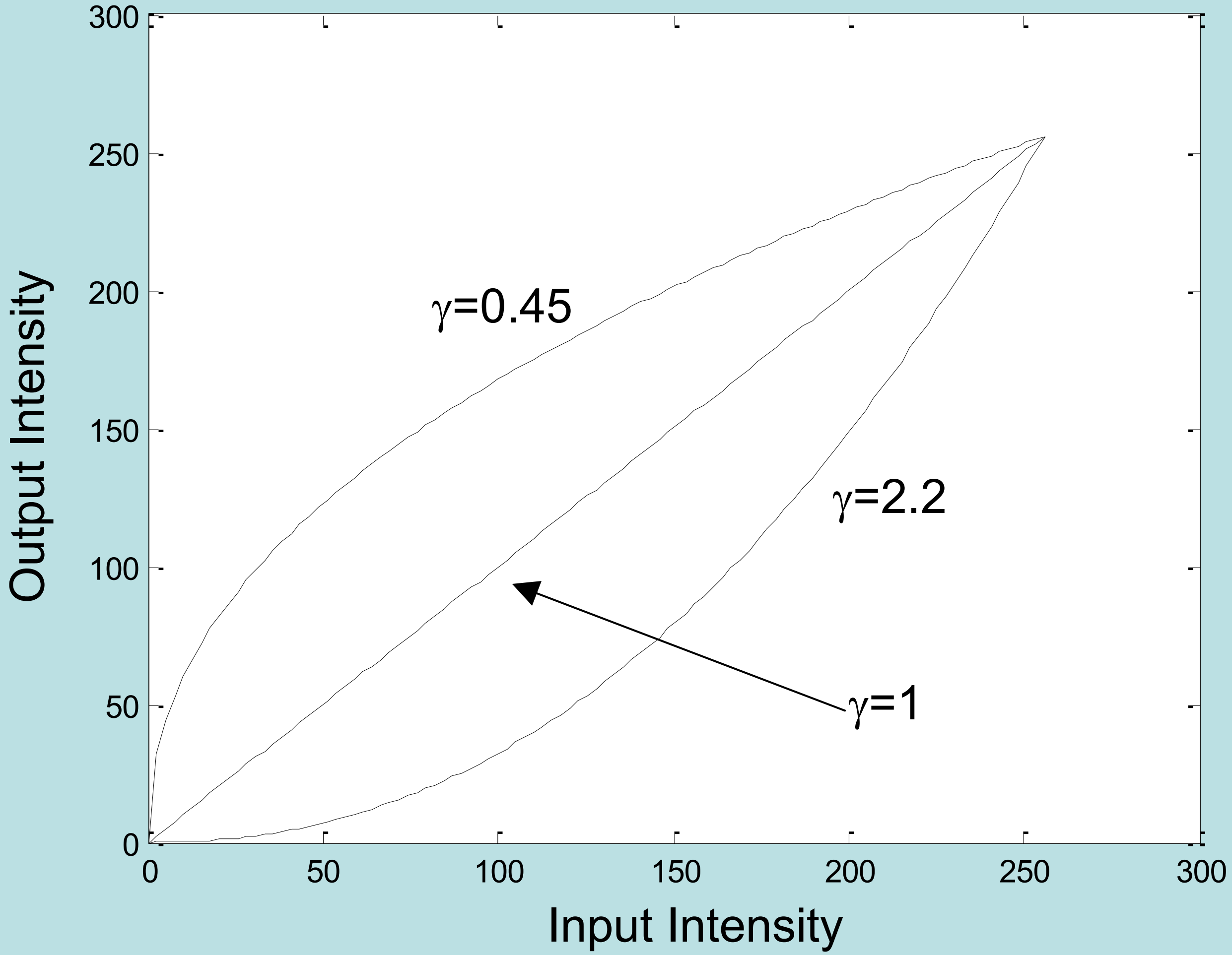


Full Range



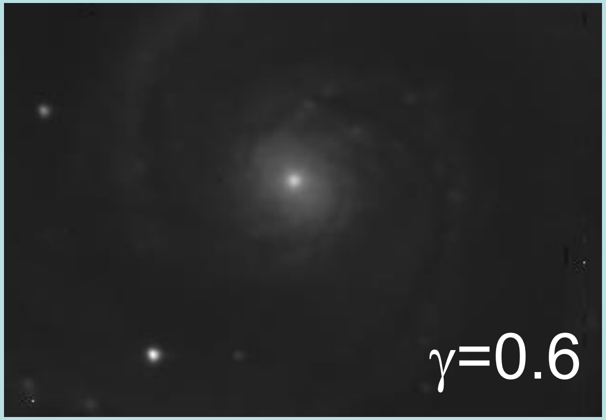
High B/C

Gamma correction



Other contrast stretching transforms….

Gamma adjustment



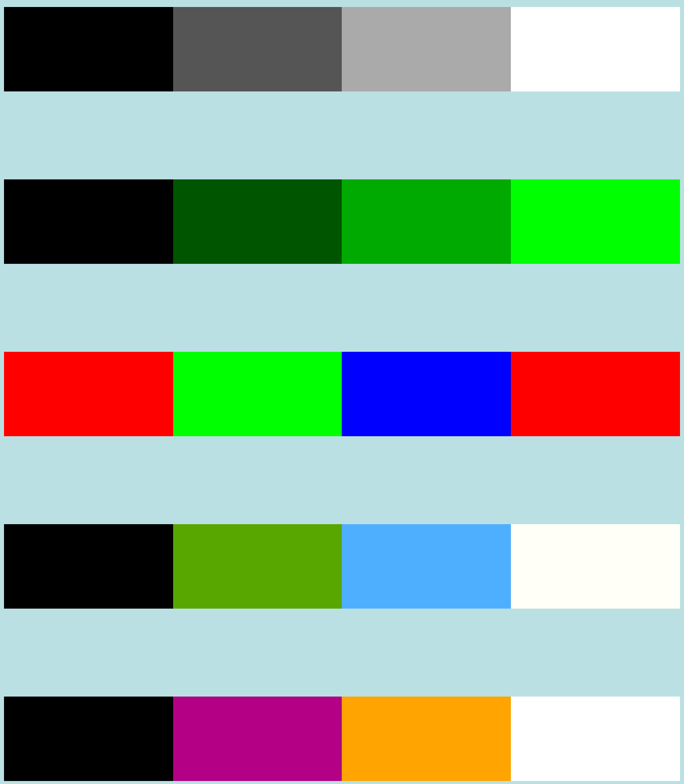
What are acceptable image manipulations?

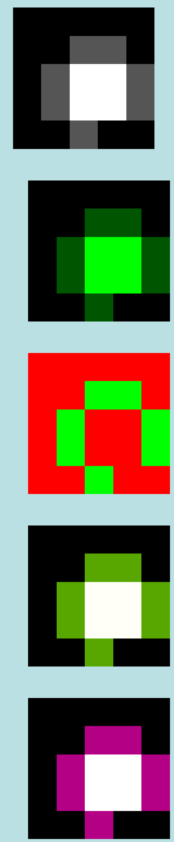
- JCB has the best guidelines
  - http://jcb.rupress.org/misc/ifora.shtml#image_aquisition
  - http://jcb.rupress.org/cgi/content/full/166/1/1
- Brightness and contrast adjustments ok, so long as done over whole image and don't obscure or eliminate background
- Nonlinear adjustments (like gamma) must be disclosed
- No cutting and pasting of regions within an image (e.g. individual cells)
- Control and experiment must be treated identically

Lookup Tables (LUTs)

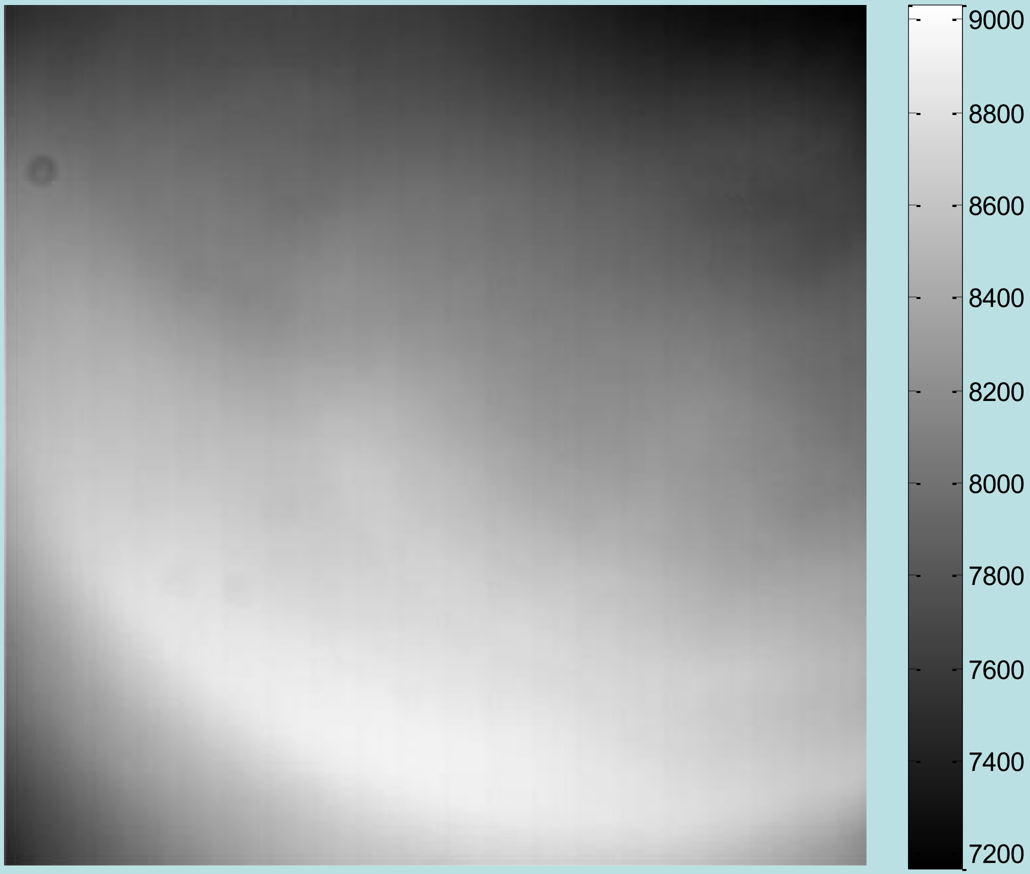
0	0	0	0	0
0	0	1	1	0
0	1	3	3	1
0	1	3	3	1
0	0	1	0	0

0 1 2 3

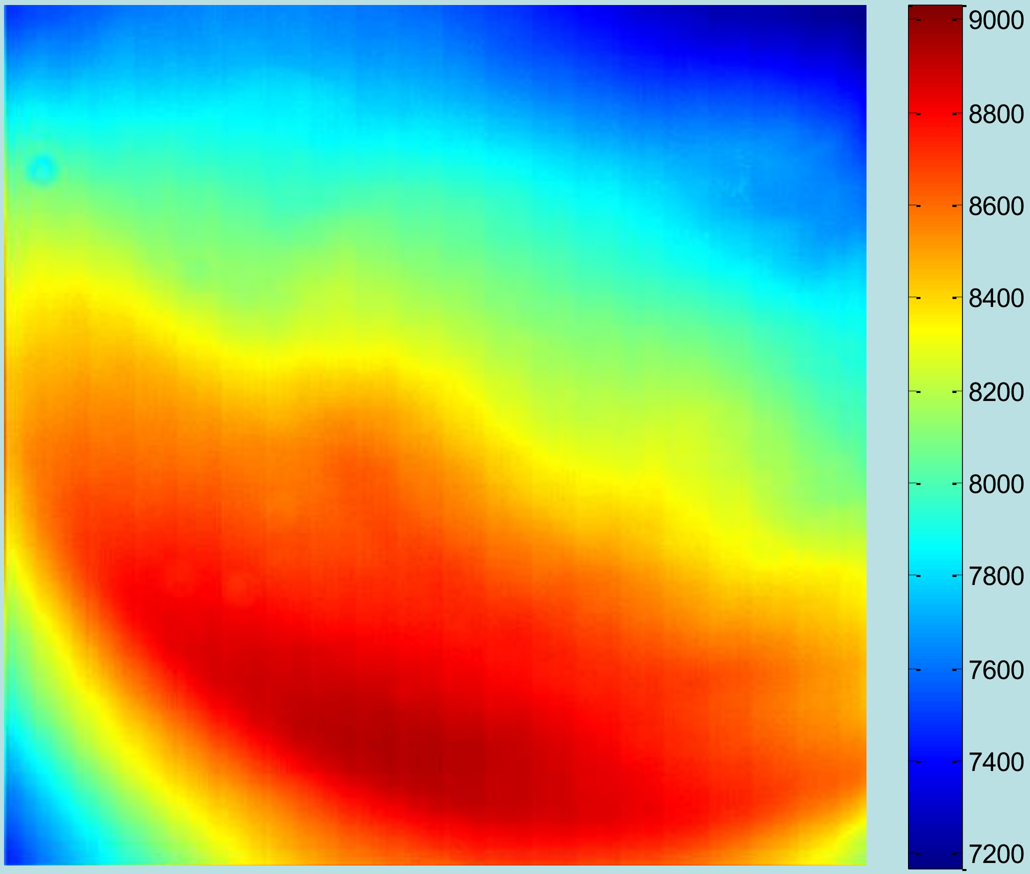




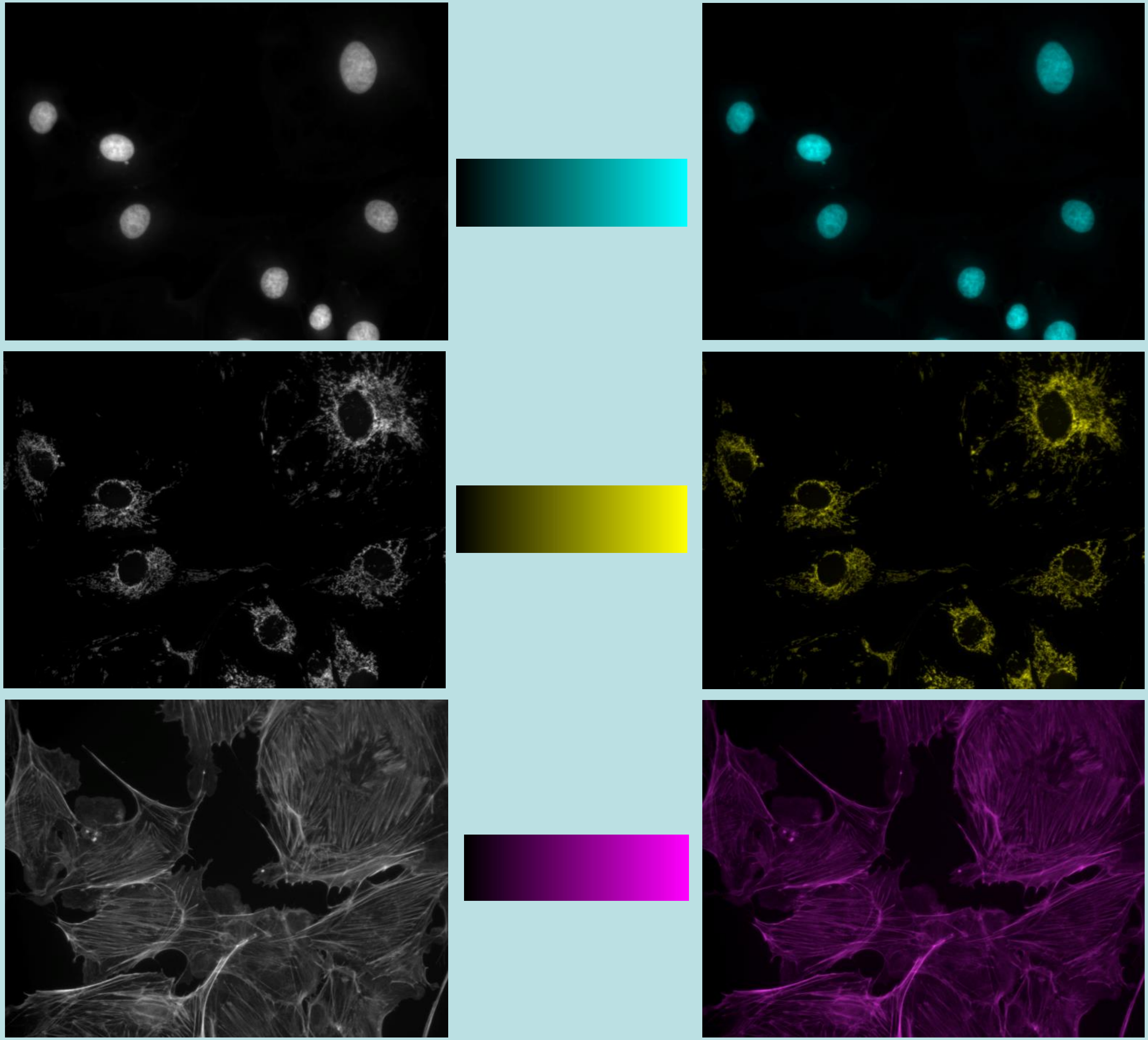
False coloring to bring out detail



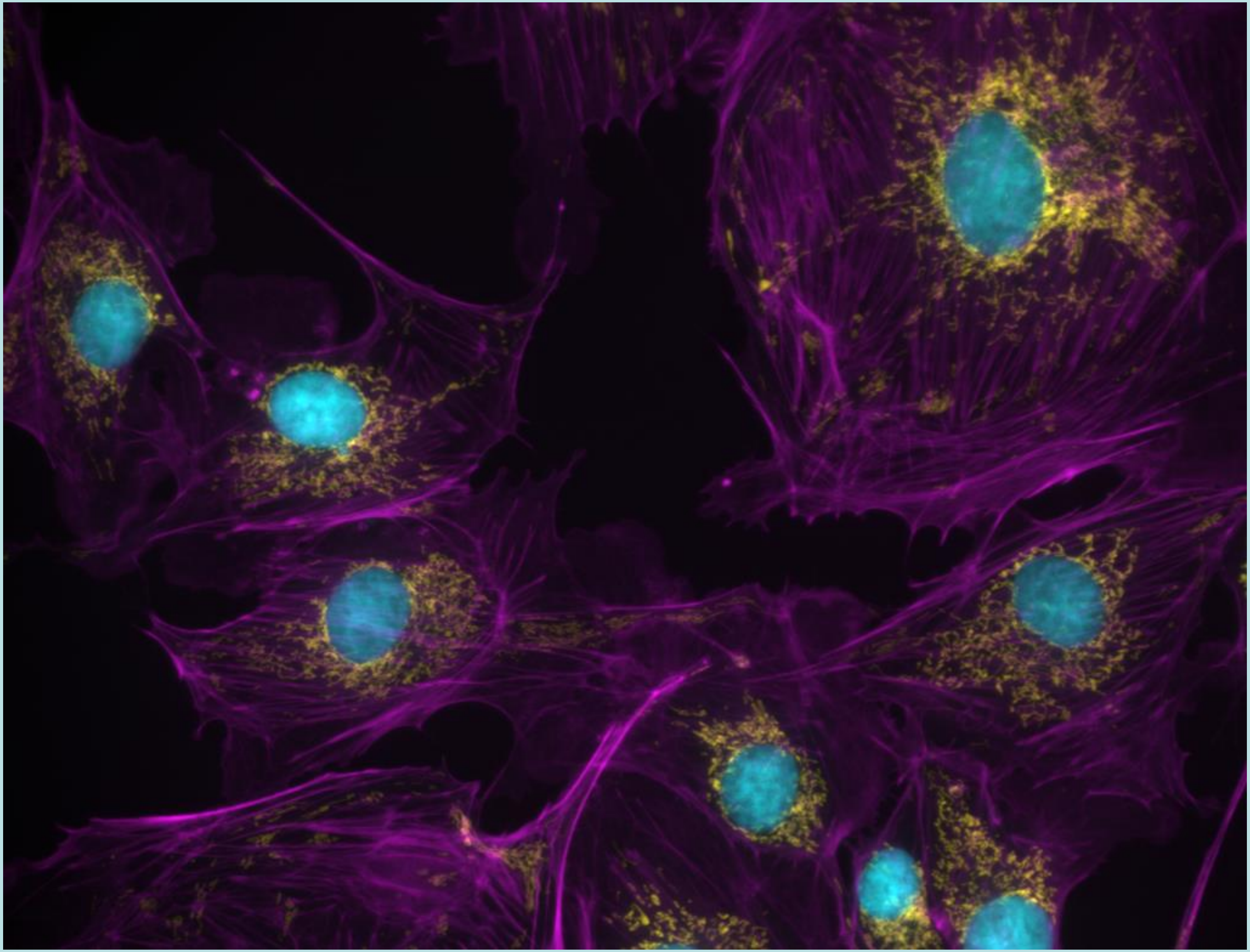
False coloring to bring out detail



Lookup Tables (LUTs)



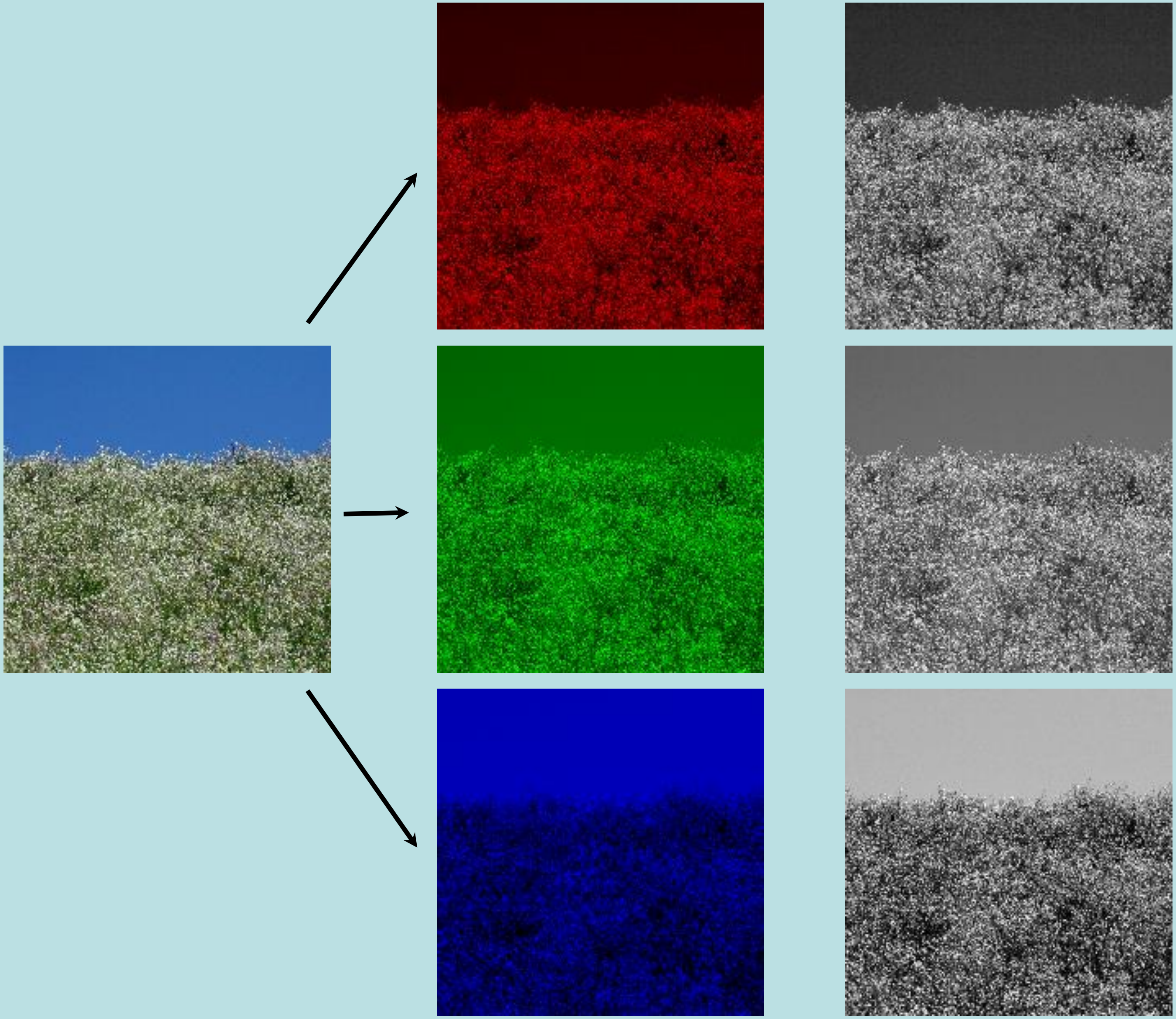
Lookup Tables (LUTs)



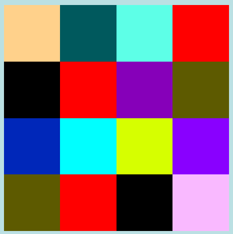
Color Images

- Color images are made up of three gray scale images, one for each of red, green, and blue
- Can be 8 or 16 bits per channel

Color Images



Color Images



=

255 209 139	0 89 93	93 255 231	255 0 0
0 0 0	255 0 0	134 0 185	93 90 0
0 39 185	0 255 255	214 255 0	137 0 255
93 90 0	255 0 0	0 0 0	249 185 255

Or

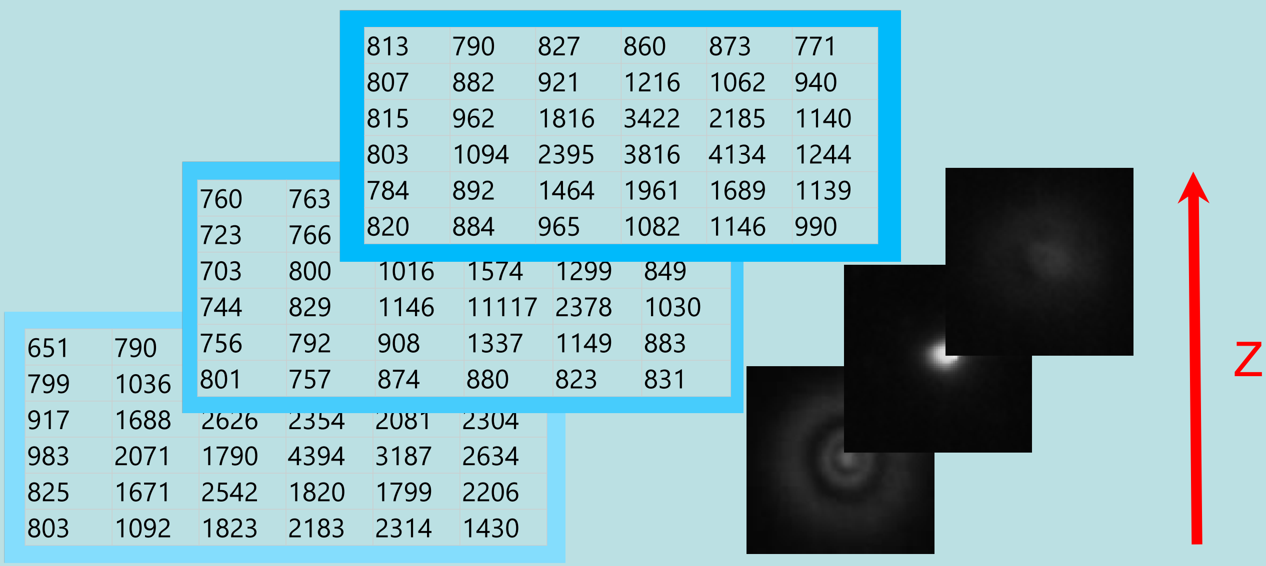
255	0	93	255
0	255	134	93
0	0	214	137
93	255	0	249

209	89	255	0
0	0	0	90
39	255	255	0
90	0	0	185

139	93	231	0
0	0	185	0
185	255	0	255
0	0	0	255

Stacks:
Sequences of images

Can represent time series(movies), z-positions, or other variables



File Formats

Data sets can be big:
$1392 \times 1040 \times 2 = 2.8$MB

3-channels, 15 image z-stack, 200 time points:
$2.8 \times 3 \times 15 \times 200 = 25.2$GB

Compression!

Compression

Lossless vs. Lossy

- Lossless: preserves all information in original image
  - Original image can be restored
- Lossy: discards information not necessary for visual appearance
  - Original image cannot be restored

- See:
  http://dvd-hq.info/data_compression.php

File Formats

- Most portable: TIFF
  - 8 or 16-bit, lossless, supports grayscale or RGB
- Good: JPEG2000, custom formats (nd2, ids, zvi, lsm, etc.)
  - Lossless, supports full bitdepth
  - Custom formats often support multidimensional images
  - Not so portable
- Generally Bad: Jpeg, GIF, BMP, etc.
  - Lossy and / or 8-bit

Software Tools

Acquisition +
Analysis

- NIS-Elements
- AxioVision
- MetaMorph
- Zen
- Slidebook
- Micro-Manager
  http://micro-manager.org/

Presentation

- Photoshop
- Gimp

Analysis

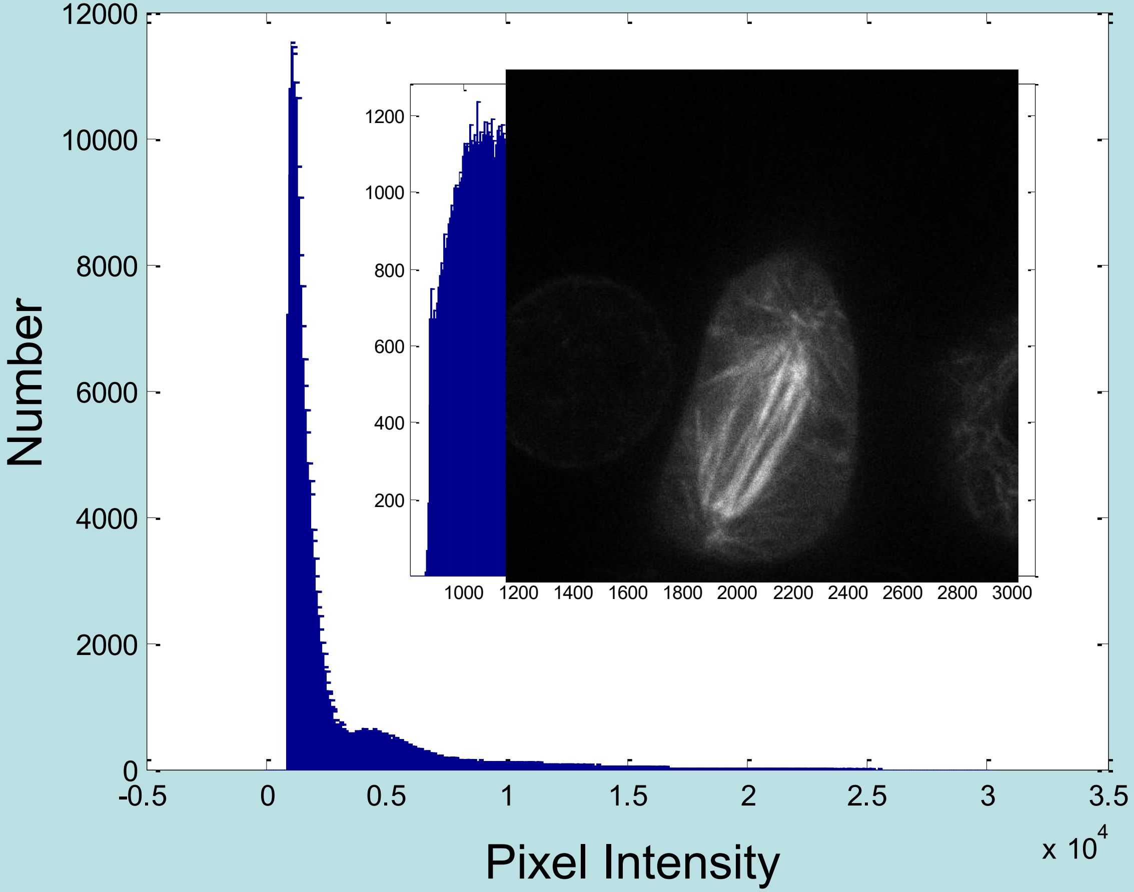
- Matlab
- ImageJ
  http://rsb.info.nih.gov/ij/
- Imaris (3D visualization)
- CellProfiler
  http://cellprofiler.org

Basic Image Analysis

Background correction

- Cameras have a non-zero offset
- There can also be background fluorescence due to media autofluorescence, etc.
- Want to correct for this by background subtraction
  - Camera dark image
  - Estimate background from image

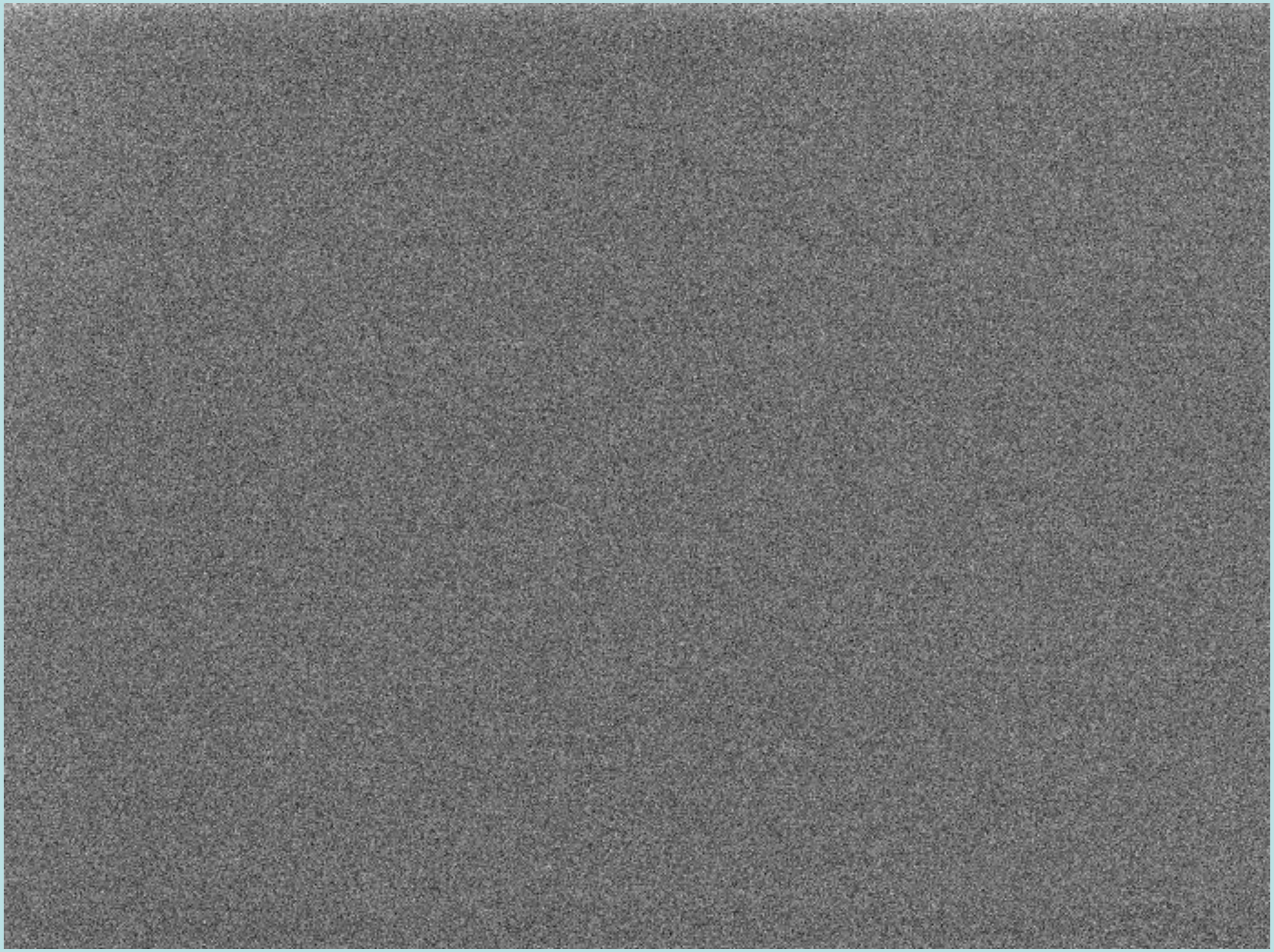
Estimating background from image



Dark image

- Acquired with no light going to the camera
  - Allows you to measure instrument background
  - Can detect what's real background autofluorescence

Dark image



Shading correction

- Measure and correct for nonuniformity in illumination and detection
- Image a uniform fluorescent sample

Shading correction

Correction procedure

$I_{meas} = I_{true} * \text{Shading} + \text{Dark}$

$I_{true} = (I_{meas} - \text{Dark}) / \text{Shading}$

Good to do on all images

Digital Image Filters

1	1	1
1	1	1
1	1	1

Averaging / Smoothing

0	1	2	1	0
1	6	10	6	1
2	10	16	10	2
1	6	10	6	1
0	1	2	1	0

Gaussian smoothing

How this works

1	1	1
1	1	1
1	1	1

(10+11+22+13+8+10+20+20+15)/9 = 14

Multiply corresponding pixels and sum

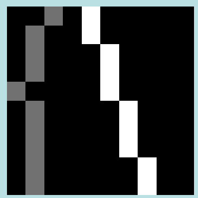
10	11	22	5	7
13	14	10	5	24
20	20	15	23	14
0	3	17	15	8
7	11	6	15	12

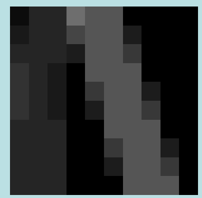
Linear Filters

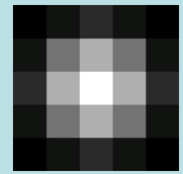
Kernel

1	1	1
1	1	1
1	1	1

Simple Smoothing







0	1	2	1	0
1	6	10	6	1
2	10	16	10	2
1	6	10	6	1
0	1	2	1	0

Gaussian Smoothing

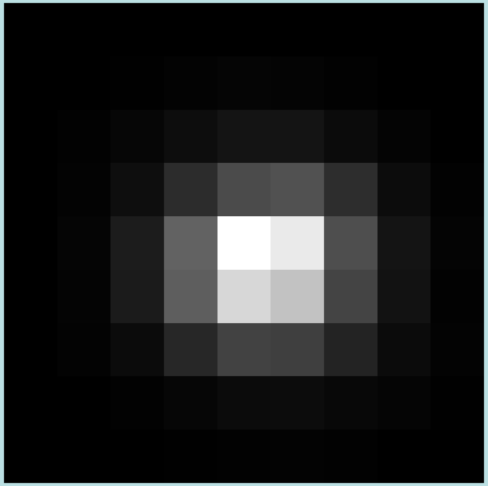
0	0	113	0	255	0	0	0	0	0
0	113	0	0	255	0	0	0	0	0
0	113	0	0	0	255	0	0	0	0
0	113	0	0	0	255	0	0	0	0
113	0	0	0	0	255	0	0	0	0
0	113	0	0	0	0	255	0	0	0
0	113	0	0	0	0	255	0	0	0
0	113	0	0	0	0	255	0	0	0
0	113	0	0	0	0	0	255	0	0
0	113	0	0	0	0	0	255	0	0

12	37	37	110	85	85	0	0	0	0
25	37	37	69	85	85	28	0	0	0
37	37	37	28	85	85	56	0	0	0
50	37	25	0	85	85	85	0	0	0
50	37	25	0	56	85	85	28	0	0
50	37	25	0	28	85	85	56	0	0
37	37	37	0	0	85	85	85	0	0
37	37	37	0	0	56	85	85	28	0
37	37	37	0	0	28	85	85	56	0
37	37	37	0	0	0	85	85	85	0

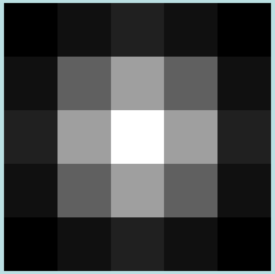
Why smooth?

- If your image is sampled appropriately (at Nyquist) the point spread function will be spread out over multiple pixels
- Properly exploiting this redundancy requires deconvolution
- But smoothing helps
- Also reduces single pixel noise artifacts that can’t be real

Actual PSF and Gaussian Filter



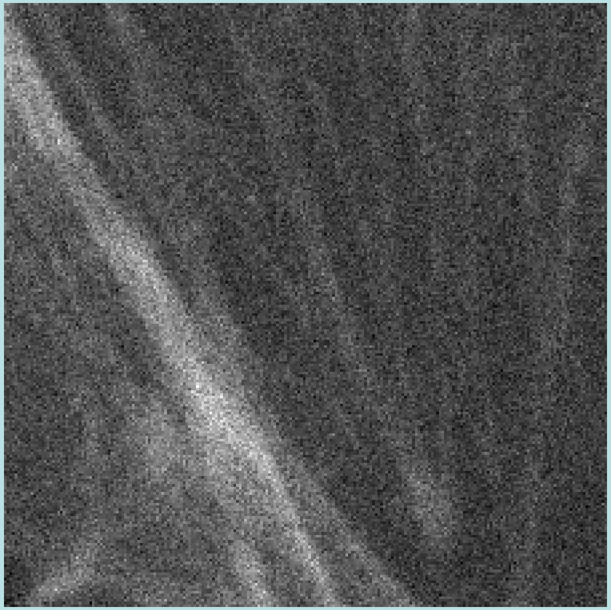
PSF



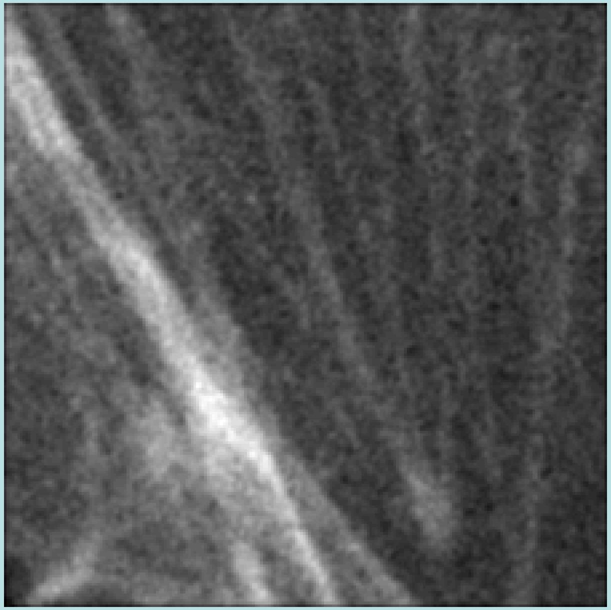
Gaussian

Why smooth?

- Averages redundancy and suppresses noise

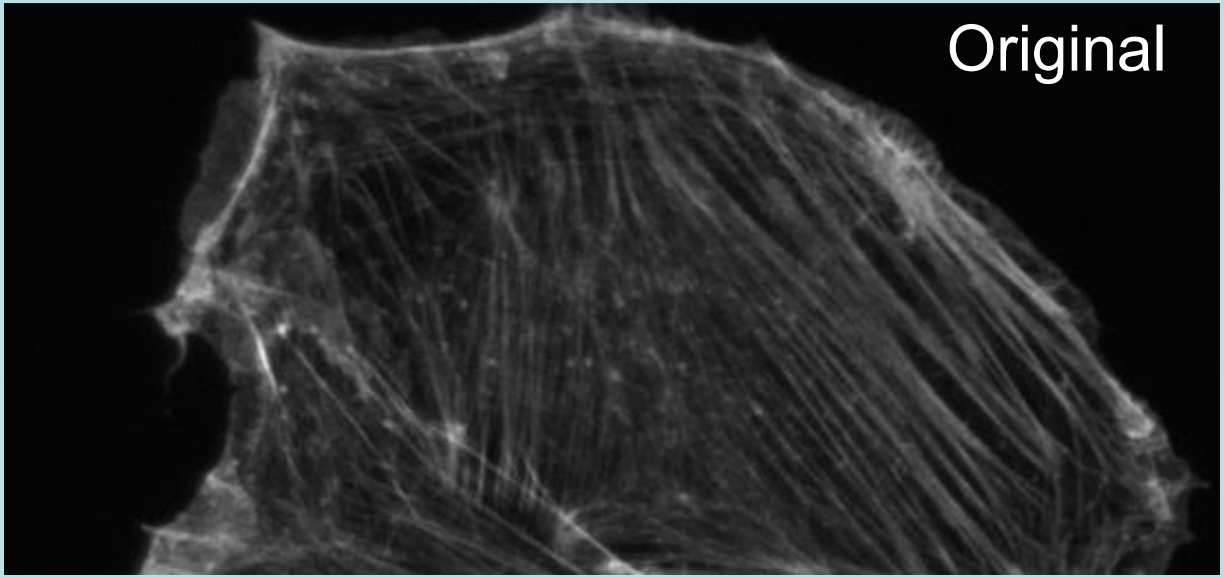


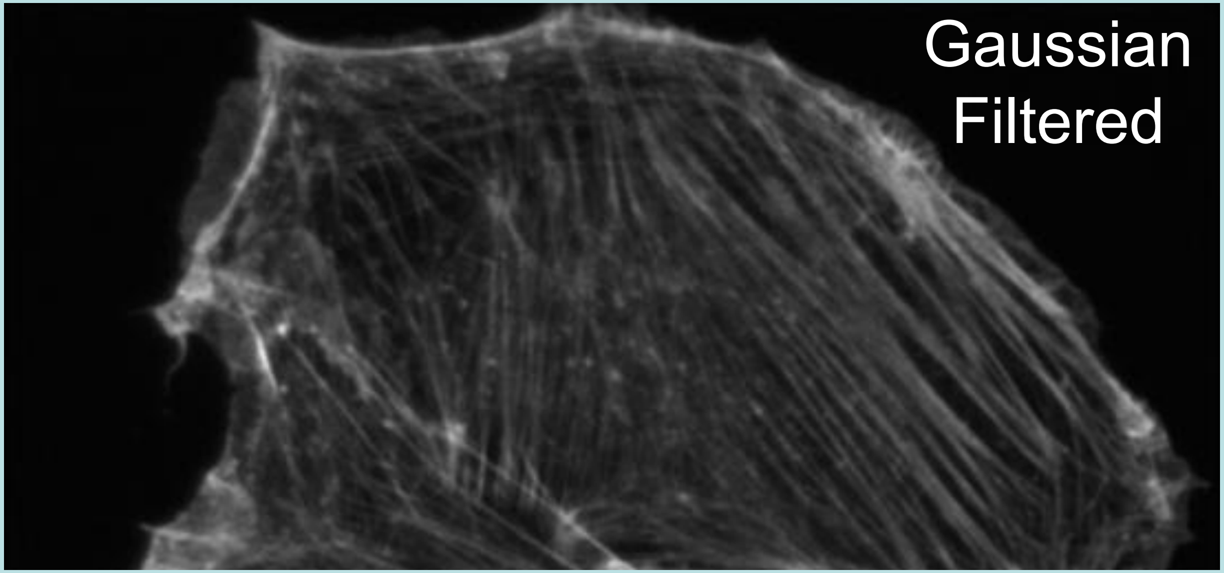
Noisy image



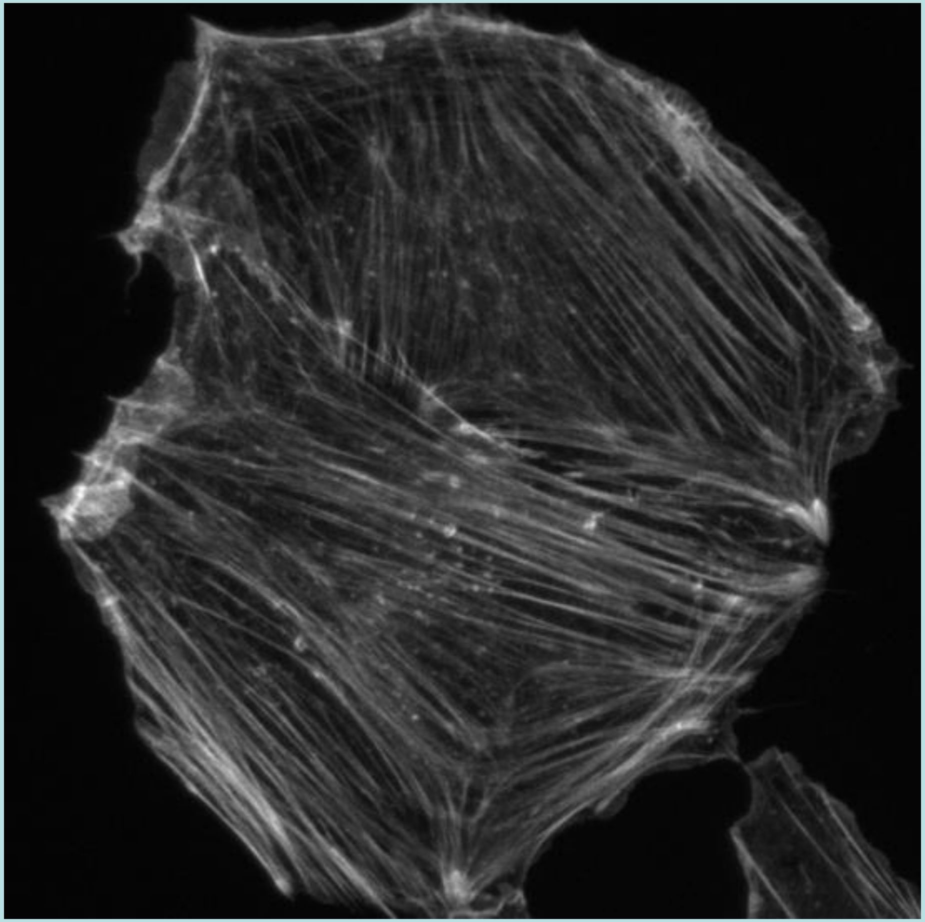
Gaussian smoothing filter, σ = 1 pixel

Smoothing





Edge Detection

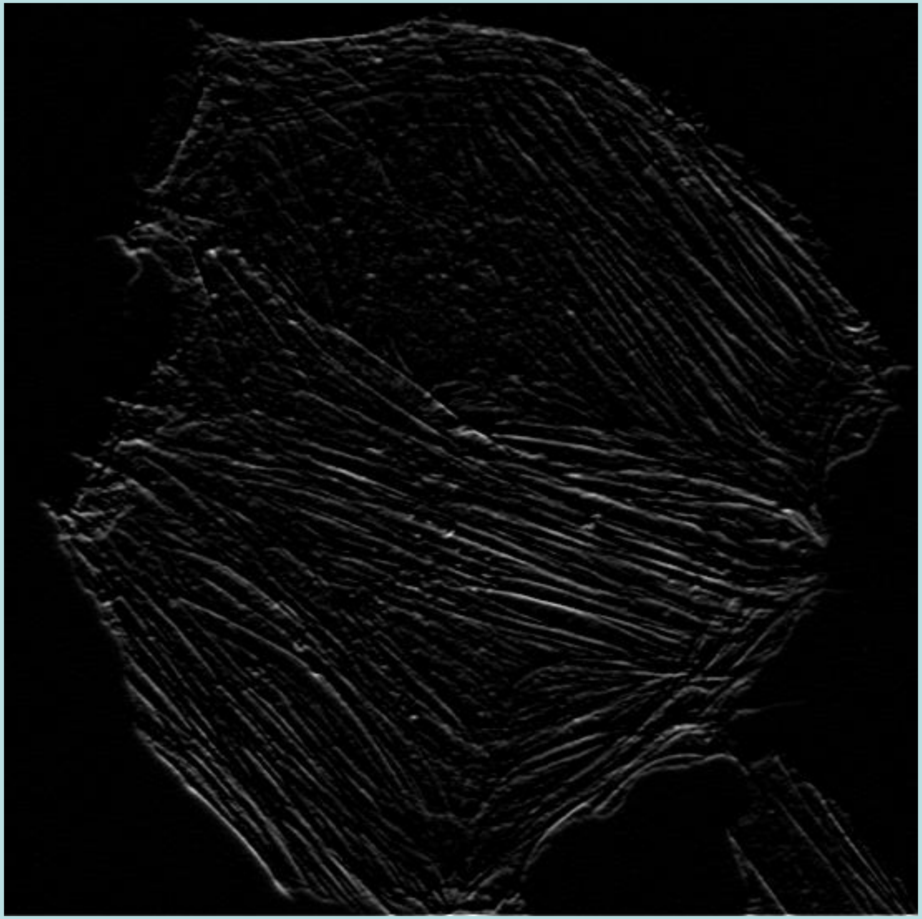


Original

1	1	1
0	0	0
-1	-1	-1

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

Edge Detection



Horizontal edge detection

1	1	1
0	0	0
-1	-1	-1

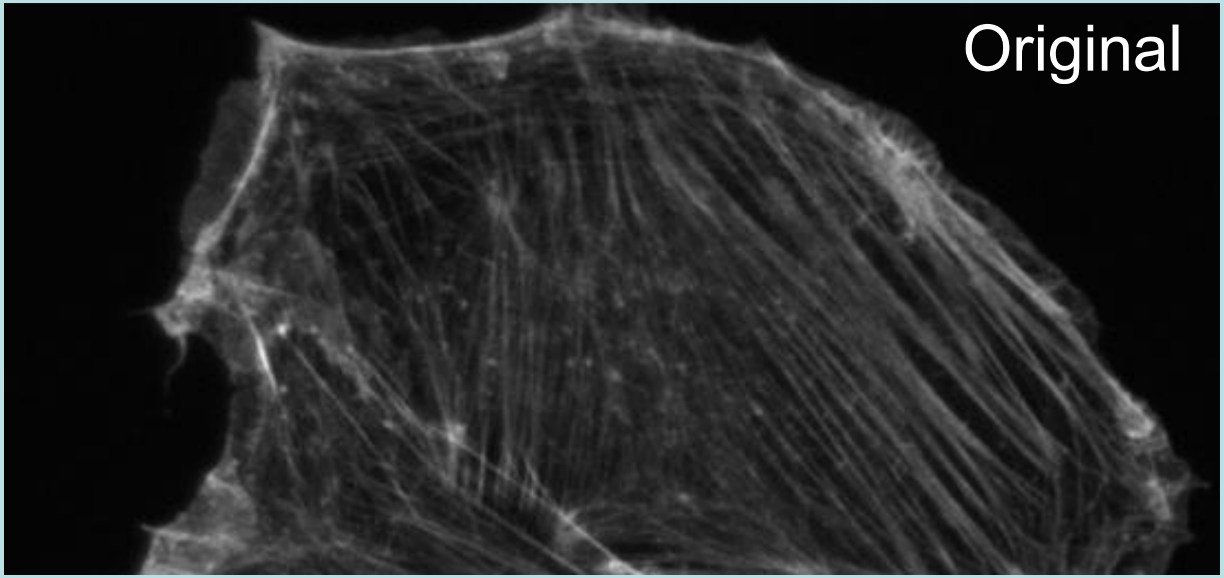
1	2	1
0	0	0
-1	-2	-1

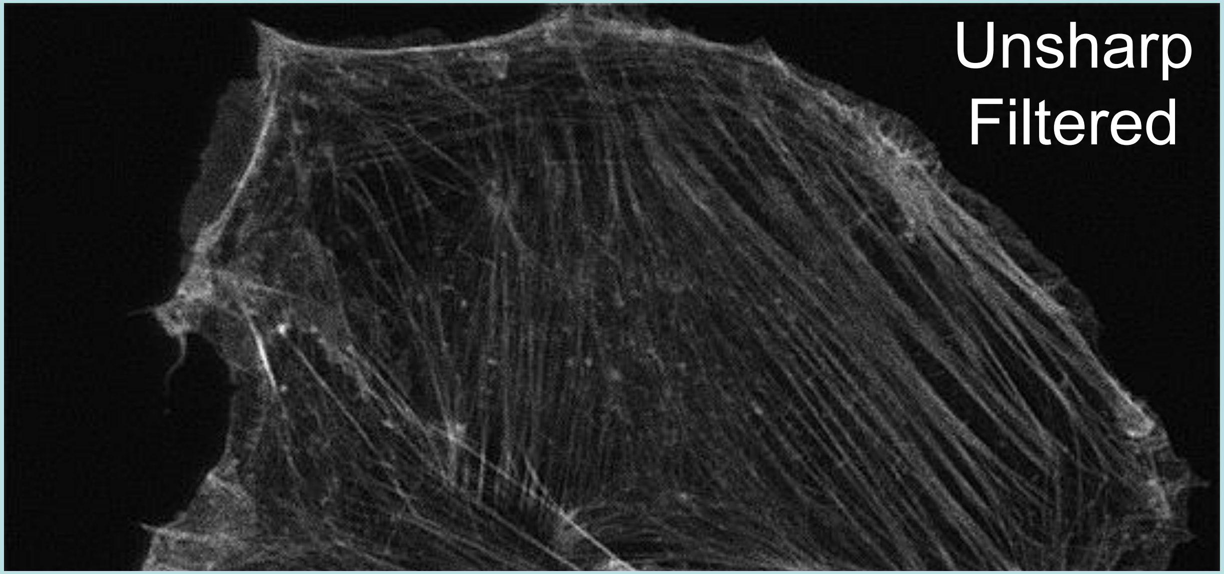
Contrast enhancement filters

- Unsharp
- Laplacian
- Laplacian of Gaussian

-1	-4	-1
-4	26	-4
-1	-4	-1

Contast enhancement

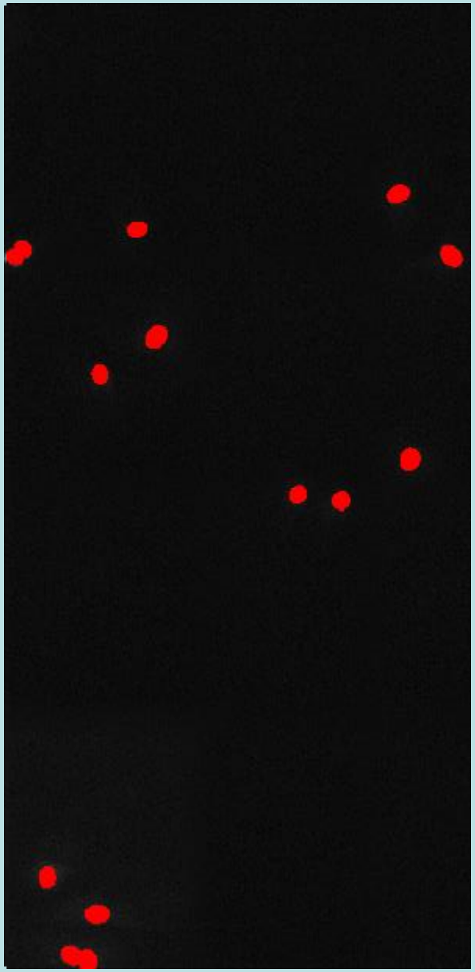


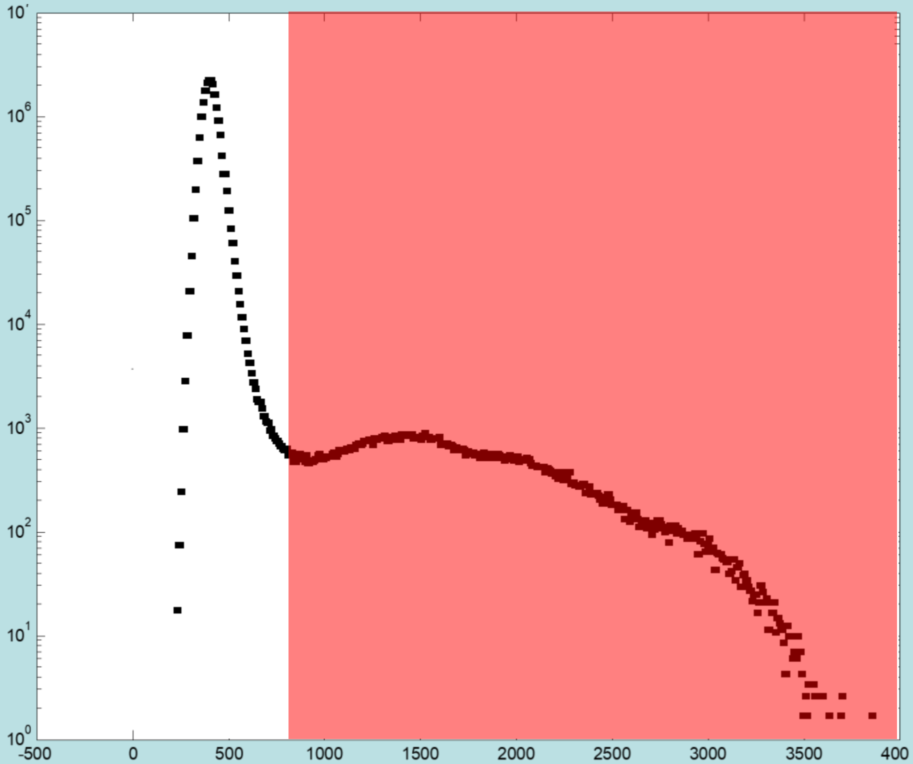


Nonlinear filters

- Can do things like median filtering – replace center pixel by median value within box
- Good for smoothing while maintaining edges

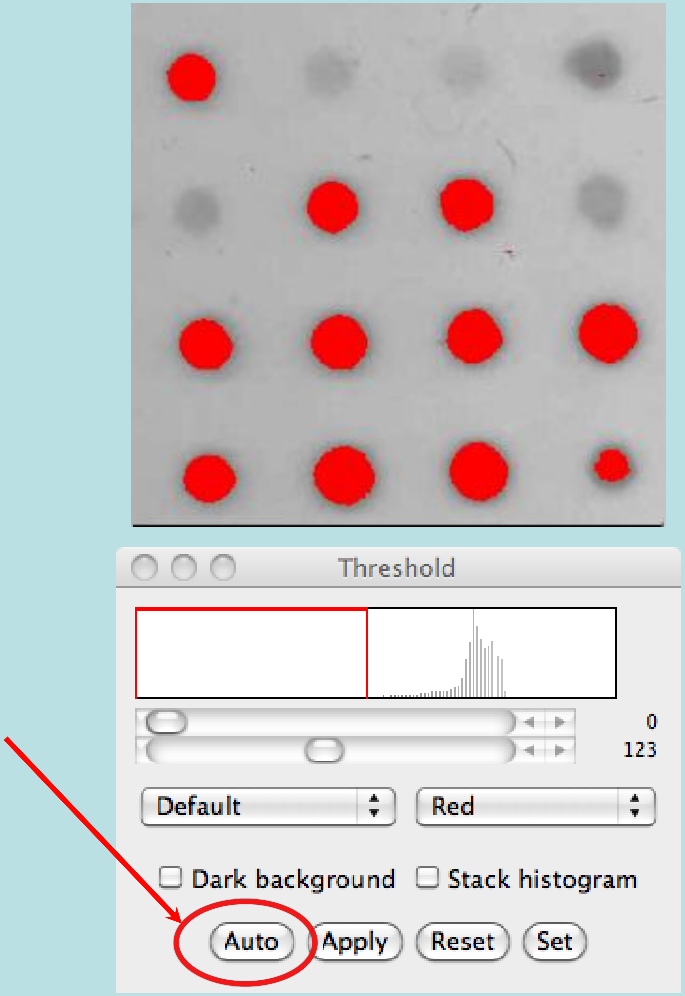
Thresholding



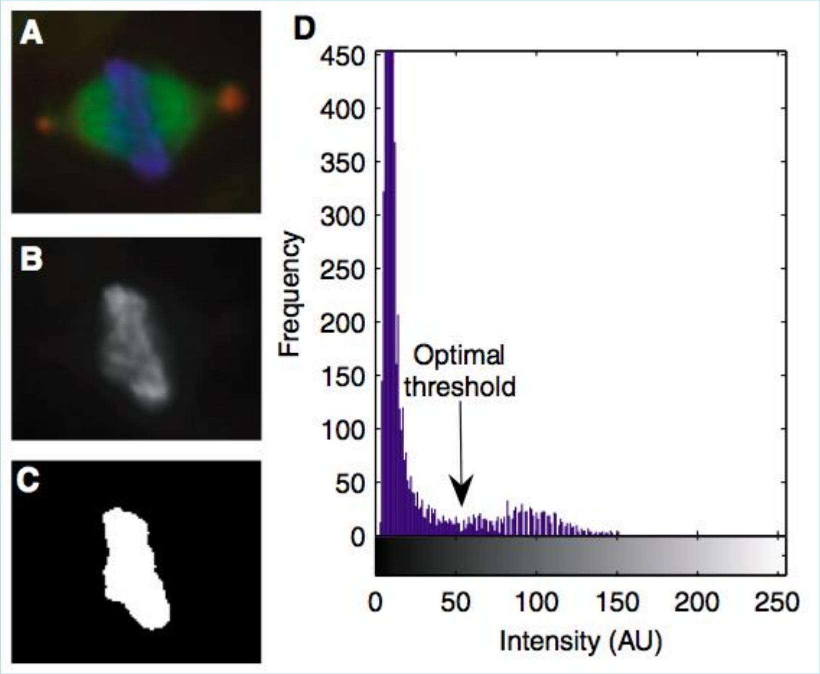


Pixel Intensity

Thresholding, where to set the cutoff?



Thresholding, where to set the cutoff?



Automatic segmentation using Otsu's method

One problem with this approach…

- It’s biased towards brighter objects
- Ideally would use second channel to independently define objects to measure

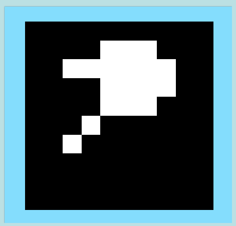
Binary images

- Thresholding gives you a binary image; 1 inside an object, 0 elsewhere
- This can be used to identify objects
- It can also be manipulated

Binary Operations: Erosion/Dilation

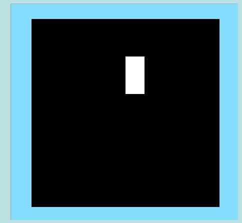
Structuring Element:

1	1	1
1	1	1
1	1	1



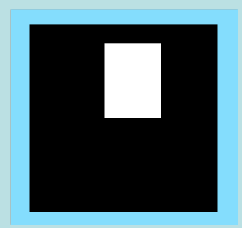
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	1	1	1	1	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Erosion



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Dilation



0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Other binary operations

- Sequential erosion and dilation – tends to smooth objects
- Hole filling
- Removing objects at borders

Acknowledgements & Further Reading

- Nico Stuurman

- Gonzalez, Woods, and Eddins: Digital Image Processing (Using Matlab)
- Burger and Burge, Digital Image Processing, An Algorithmic Introduction using Java (ImageJ)
- Pawley: Handbook of Biological Confocal Microscopy