

Procesamiento de Imágenes I



Jorge Jara, Dante Castagnini, Nicole C. Huerta

www.scian.cl / www.cimt.cl / <https://bni.cl/biomat.php>

Laboratory for Scientific Image Analysis (SCIAN-Lab)
Centro de Informática Médica y Telemedicina (CIMT)
Biomedical Neuroscience Institute (BNI)
Programa de Biología Integrativa
Instituto de Ciencias Biomédicas (ICBM)
Facultad de Medicina, U. de Chile

Sesiones

Clases

- | | | |
|---------------------------------|-----------------|----|
| • Procesamiento de Imágenes I | Martes 19/10 | JJ |
| • Procesamiento de Imágenes II | Miércoles 20/10 | JJ |
| • Procesamiento de Imágenes III | Jueves 21/10 | NC |

Actividades Prácticas

- | | |
|--|---------|
| 1. Deconvolución & Manejo Básico de Imágenes | NC & JJ |
| 2. Segmentación | JJ & DC |
| 3. Análisis Morfológico | DC & JJ |



<http://phdcomics.com/comics.php?f=1156>

I.

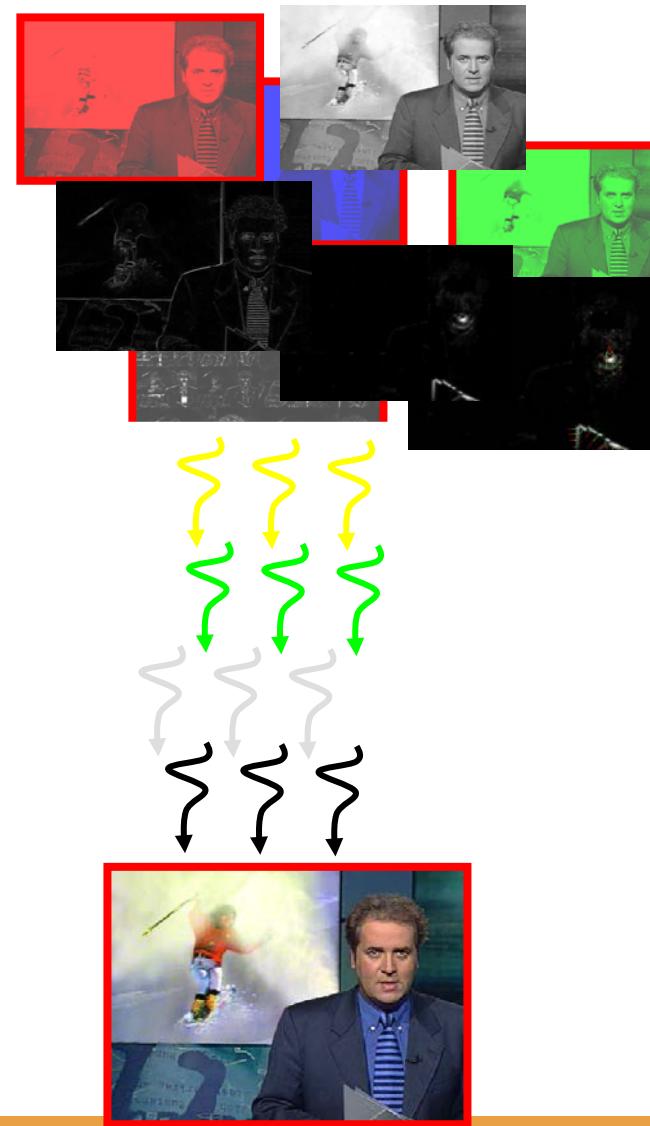
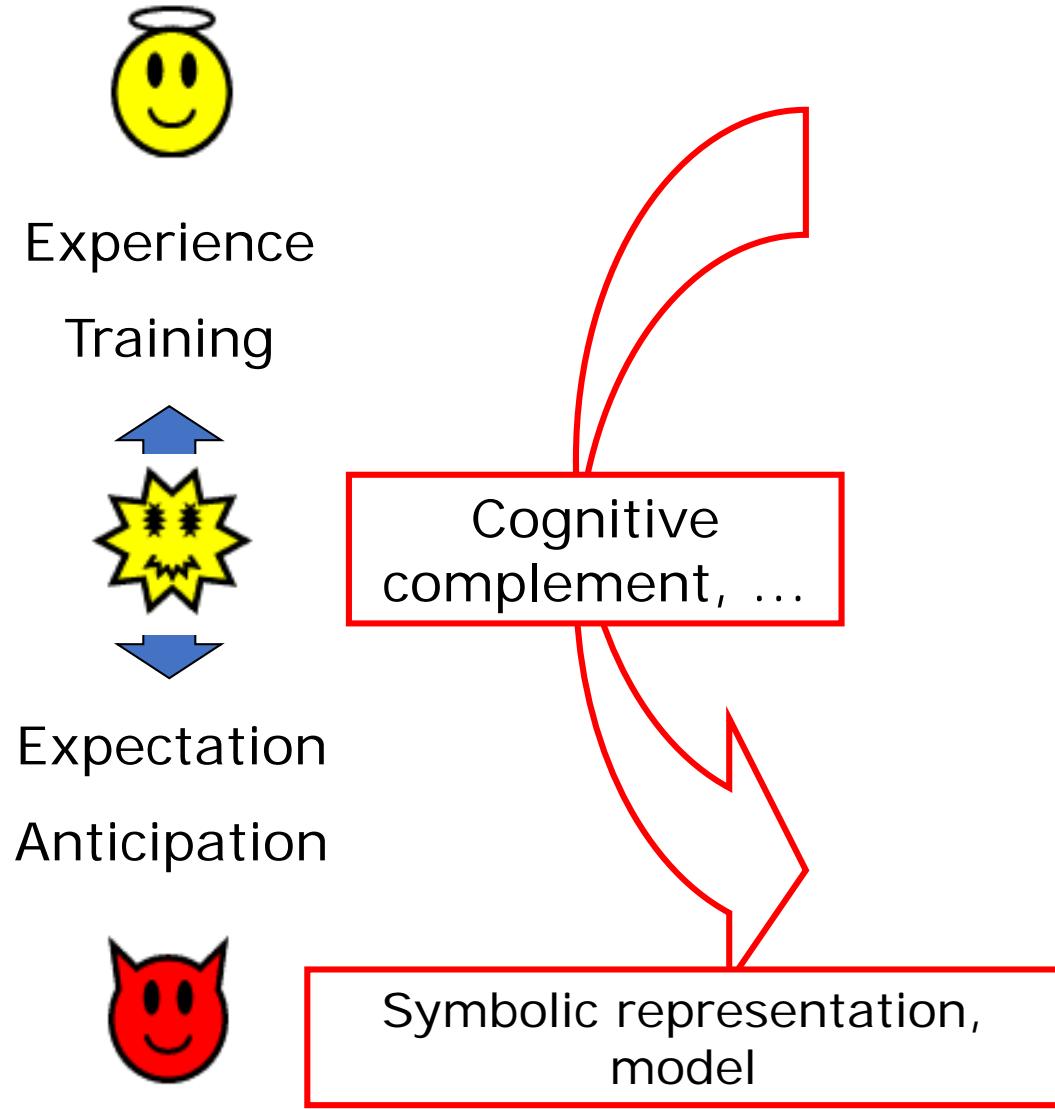
- Digital image processing
- Segmentation basics

II.

- Advanced techniques

0. Concepts & People

- Signal & noise
- Signal amplitude
- Signal frequency/frequencies
- Signal amplitude
 - Fluorescence intensity
- Pixel / Voxel
- Bit
- (Sample) Bit depth
- Sampling frequency/interval
- Filter



- Common criteria

Color similarity
(regions)



Color transitions or
gradients (boundaries)



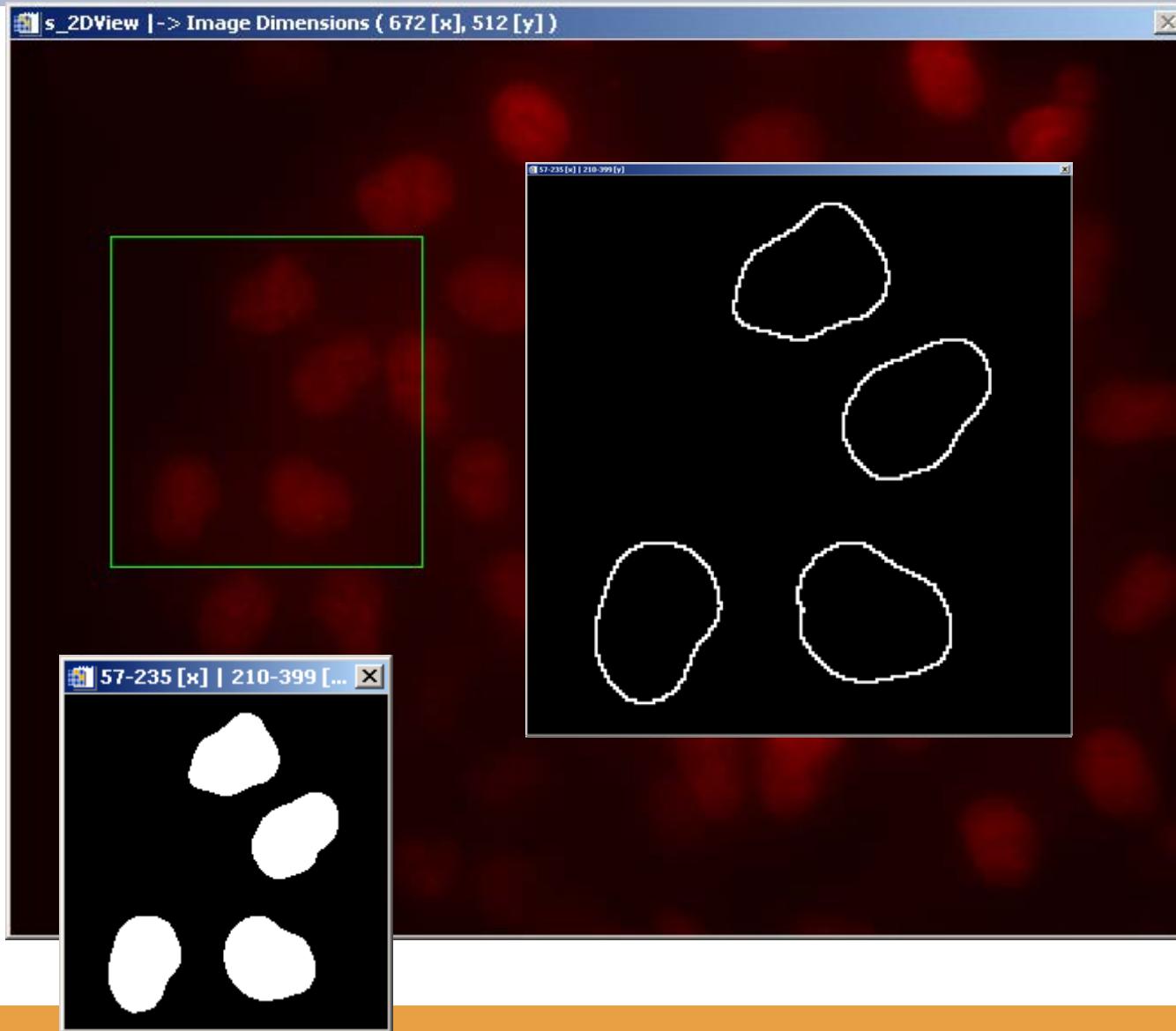
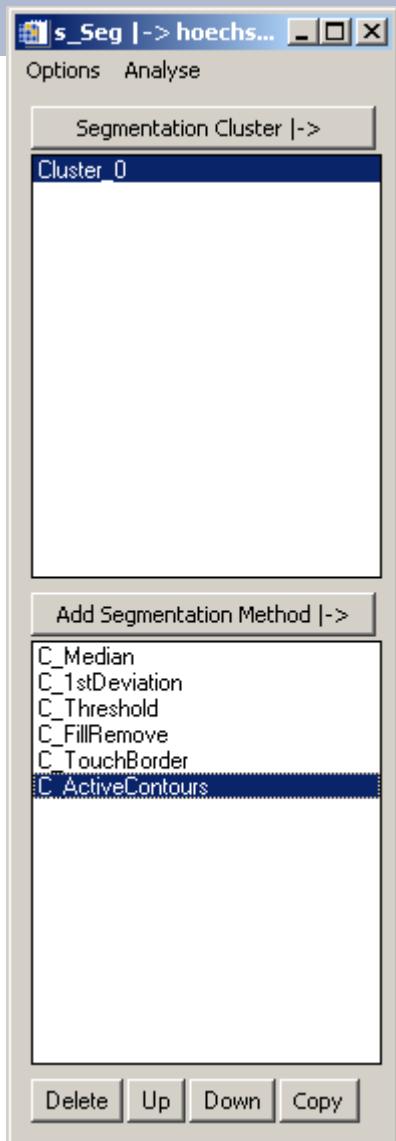
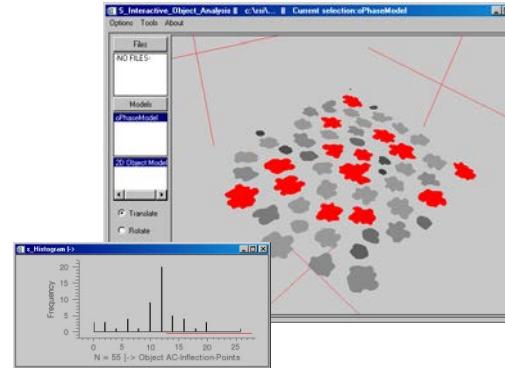
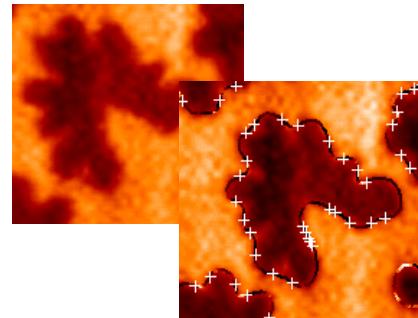
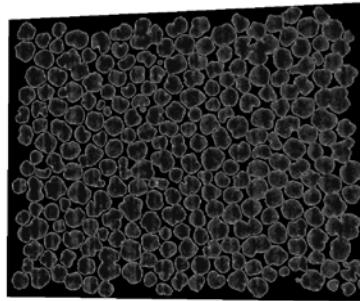


Image Processing



Acquisition
• Microscopes
• Cameras
• ...

(pre)Treatment
• Restoring
(e.g. PSF deconvolution)
• Contrast enhancement
• Artifact removal

Analysis
• ROIs / models
• Measurements

Understanding
• Higher level task
• Hypotheses support

images

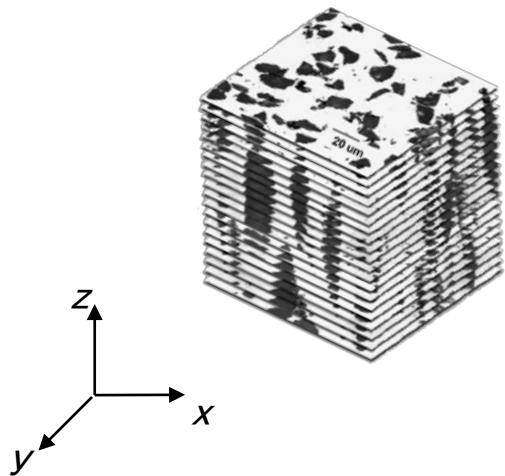
images

images +
descriptions



- Digital image processing means...

- Digital... discrete, finite



- A discrete set is composed by elements which are “isolated” one from another

Examples:

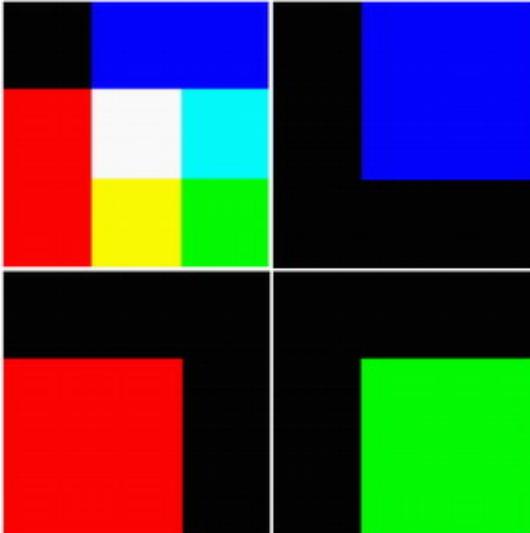
- Natural numbers $\{1, 2, 3, \dots\}$ (infinite set)
 - Natural numbers from 1 to 10 (finite set)

- If not discrete? Continuum

Example: real numbers in the $[0,1]$ interval
(infinite set)

- A **digital image** can be defined as a function over a discrete space

- A typical 2D image model is the **raster image**: array (matrix) of **pixels** in cartesian coordinates (x, y)
- A numeric value for **brightness (intensity)** or **color** is associated to each pixel



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
255	0	255	0	255	0	255	0	255	0	255	0	255	0	255	0
0	0	255	255	0	255	255	0	255	255	0	255	255	0	255	255
255	0	0	255	255	0	255	0	255	0	255	0	255	0	255	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	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	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

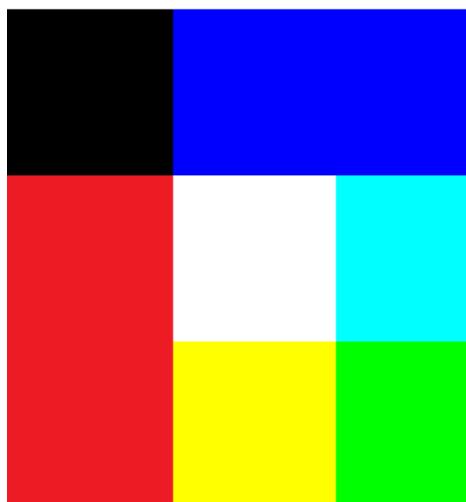
$$I = f(x, y)$$

$$(x, y) \in [0, \dim_x - 1] \times [0, \dim_y - 1]$$

$$I[x_i, y_j] = f[x_i, y_j]$$

- RGB image

- Three channels for respective primary colors: Red, Green, Blue. 8 bits determine [0,255] range



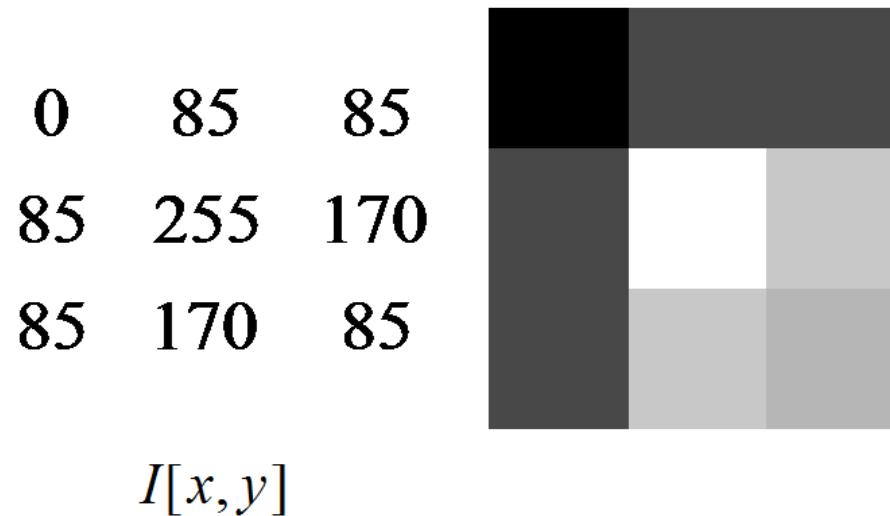
0	0	0
0	0	0
0	255	255
255	255	0
0	255	255
0	255	255
255	255	0
0	255	255
0	0	0

$$r[x, y] \ g[x, y] \ b[x, y]$$

- Analogous case: CMYK (Cyan, Magenta, Yellow, Black)

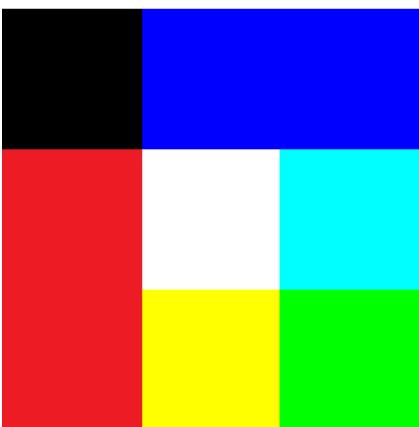
- Greyscale image

- A brightness (intensity) level is defined for each pixel



How to go from an RGB
to a greyscale image?

- From RGB to greyscale



$r[x, y]$ $g[x, y]$ $b[x, y]$

0	0	0
0	0	0
0	255	255
255	255	0
0	255	255
0	255	255
255	255	0
0	255	255
0	0	0

$I[x, y]$

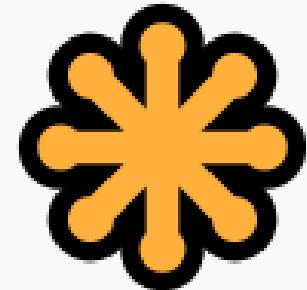
0	85	85
85	255	170
85	170	85



How good is the human eye
resolving colors in R, G or B tones?

Scalable Vector Graphics

- Other than raster images...
 - A **vector image** is defined by using a set of base elements (like shapes or curves), instead of explicitly give the color/intensity for each pixel
 - Example: SVG images; base functions (wavelet, splines, Fourier)
- In order to show a vector image in a common digital screen (pixel matrix) a **rasterization** algorithm is applied

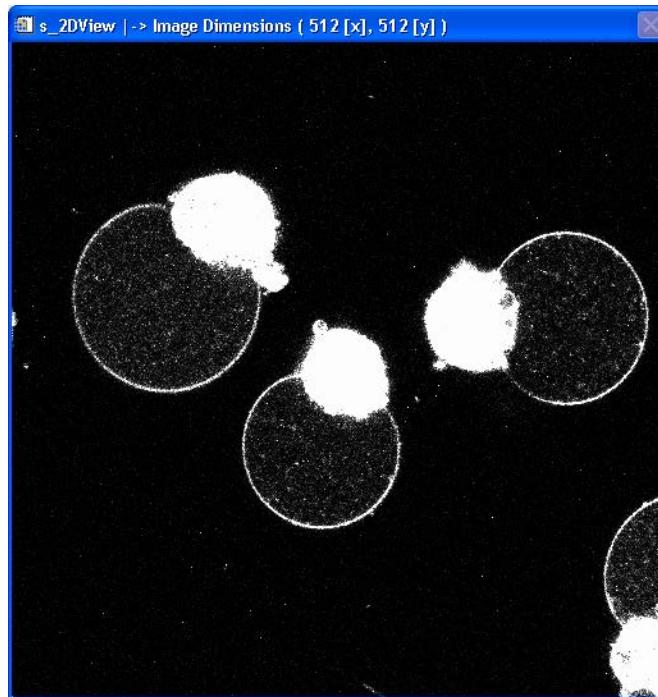


```
<?xml version="1.0" encoding="UTF-8"?>
<svg version="1.0" xmlns="http://www.w3.org/2000/svg">
<defs>
  <linearGradient id="box_gradient" x1="0%" y1="0%" x2="100%" y2="0%">
    <stop stop-color="black" offset="0%"/>
    <stop stop-color="white" offset="100%"/>
  </linearGradient>
  <circle id="circle" cx="100" cy="300" r="100" fill="black" stroke="white" stroke-width="2px" stroke-miterlimit="10" stroke-dasharray="5 5" style="filter: url(<#box_gradient>);>
  <line id="line" x1="100" y1="300" x2="99.7" y2="300" style="stroke: black; stroke-width: 2px; stroke-miterlimit: 10; stroke-dasharray: 5 5; filter: url(<#box_gradient>);>
</defs>
<use xlink:href="#box_gradient" x="0" y="0" width="100%" height="100%"/>
<use xlink:href="#circle" x="100" y="300" width="100%" height="100%"/>
<use xlink:href="#line" x="100" y="300" width="100%" height="100%"/>
<!--add more content-->
<circle cx="90" cy="300" r="100" fill="black" stroke="white" stroke-width="2px" style="filter: url(<#box_gradient>);>
</svg>
```



- ...so, a digital image can be treated as a **function** (in the mathematical sense)...
 - on a discrete domain
 - with numeric values associated to each elements, representing a property (such as color, brightness, depth, etc.)

Binary value	Decimal value
0000 0000	0 (black)
0000 0001	1
0000 0010	2
0000 0011	3
0000 0100	4
0000 0101	5
0000 0110	6
0000 0111	7
0000 1000	8
...	...
1111 1011	251
1111 1100	252
1111 1101	253
1111 1110	254
1111 1111	255 (blanco)

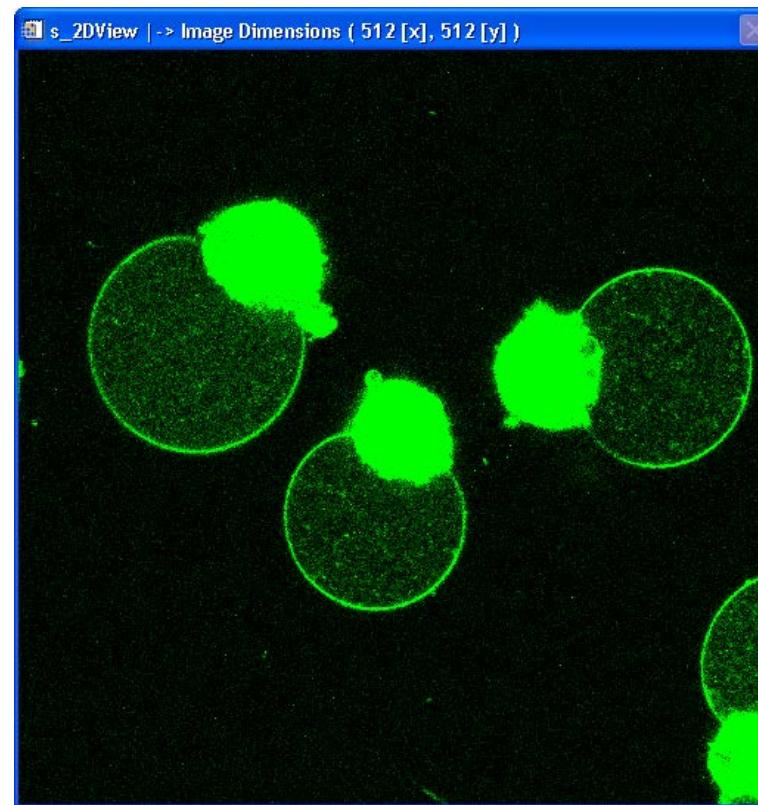
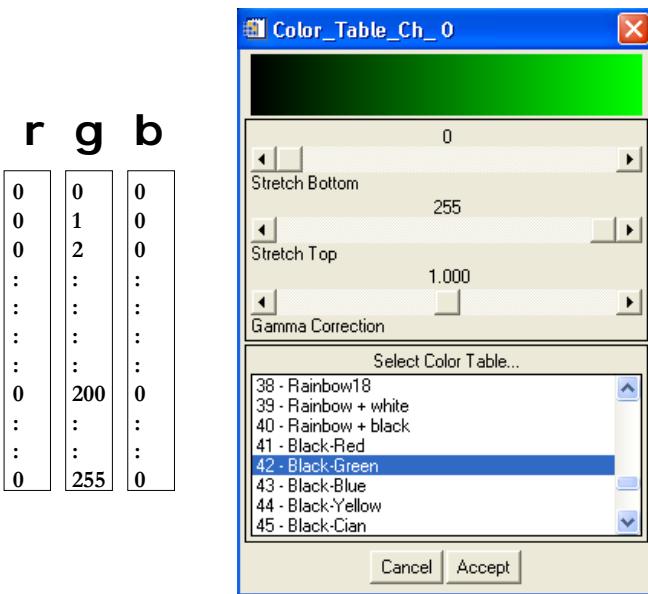


8 bit greyscale image

$$I(290,267) = 220$$

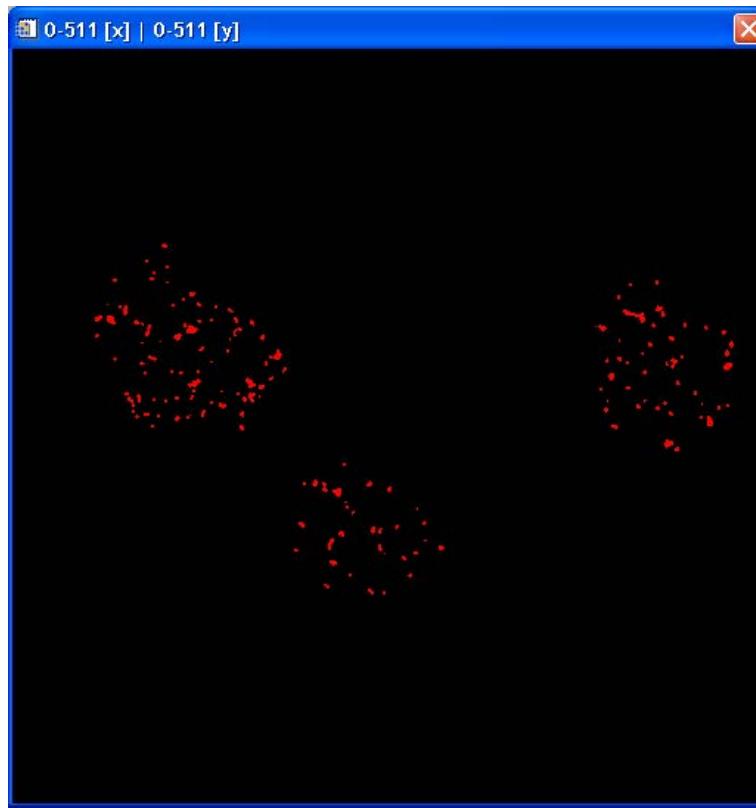
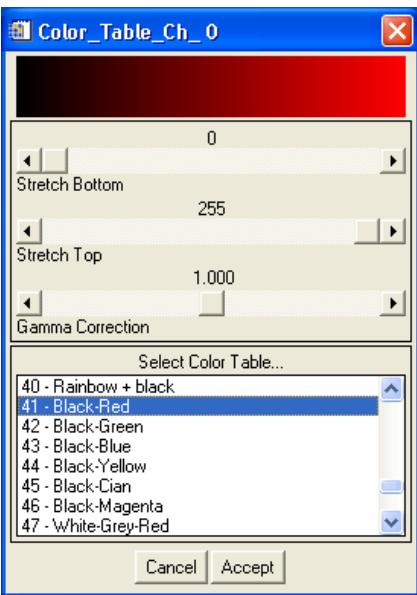
A n-bit greyscale image encodes up to 2^n intensity values

- It is possible to define color *lookup tables* (LUTs) for visualization purposes. For instance, a greyscale image can be displayed using a black-to-green LUT...



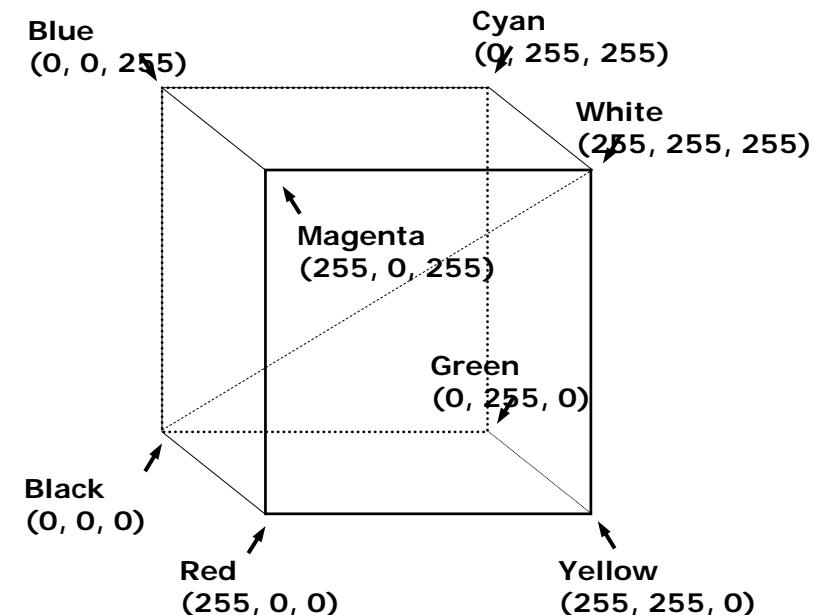
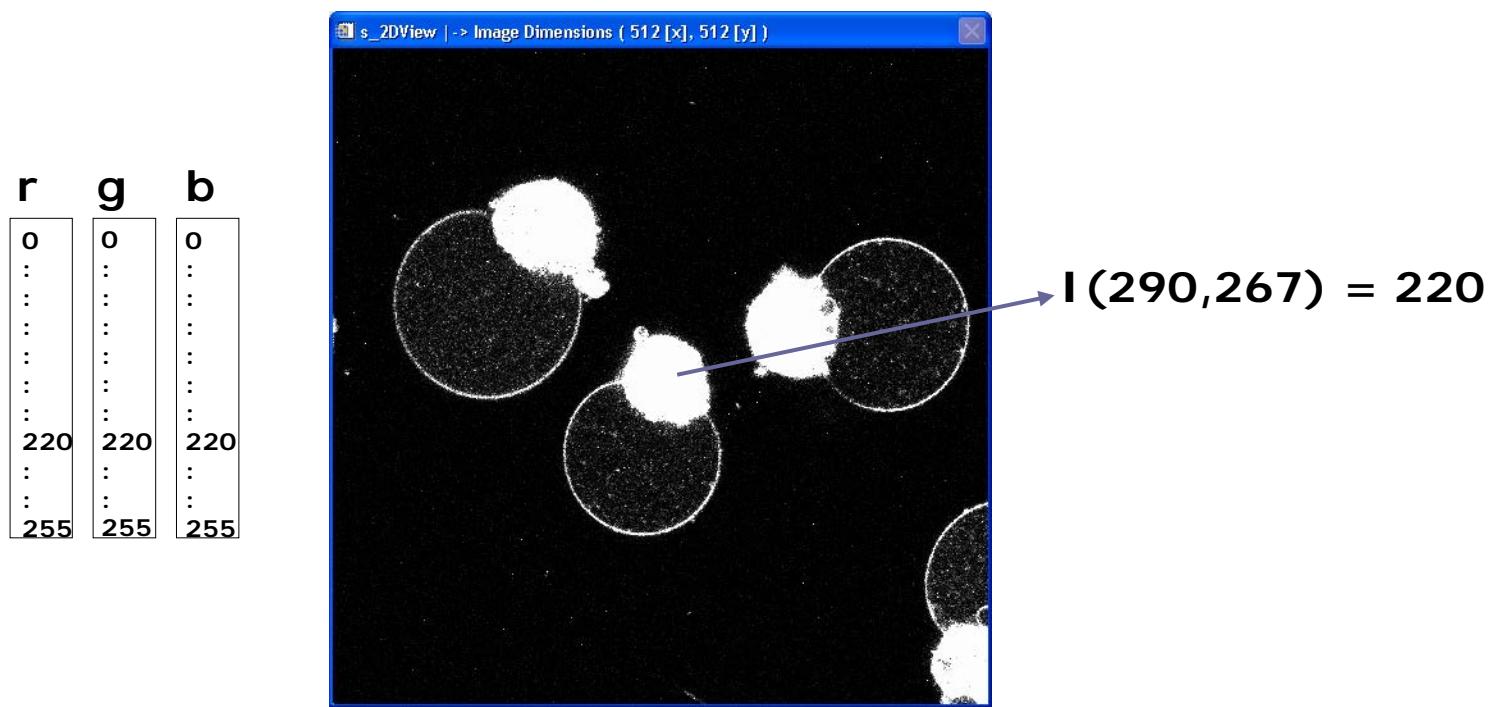
- Now with a black-to-red LUT...

r	g	b
0	0	0
1	0	0
2	0	0
:	:	:
:	:	:
:	:	:
220	0	0
:	:	:
255	0	0



In ImageJ/FIJI: Go to
the "Image"->"Lookup
Tables" menu

- ...or any custom color table



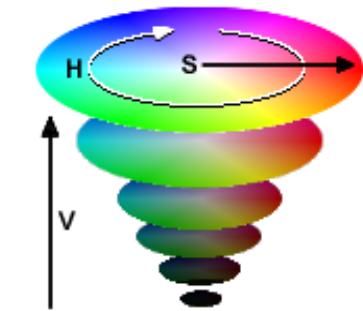
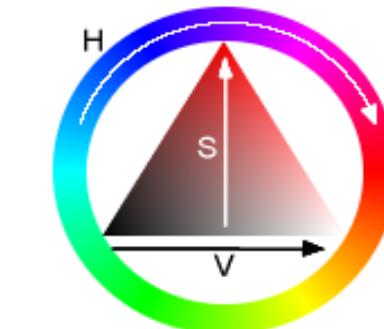
$$\begin{aligned}
 \text{Red} &= [r_0, r_1, \dots, r_{255}] \\
 \text{Green} &= [g_0, g_1, \dots, g_{255}] \\
 \text{Blue} &= [b_0, b_1, \dots, b_{255}]
 \end{aligned}$$

HSV (hue, saturation, value) model

http://en.wikipedia.org/wiki/HSV_color_space

- **Hue**
color „type“, range 0-360° (0° red, 120° green, 240° blue)
- **Saturation**
color „intensity“, range 0-100%.
- **Value**
brightness, range 0-100%.

HSV is a **non linear** transformation from the RGB color space.

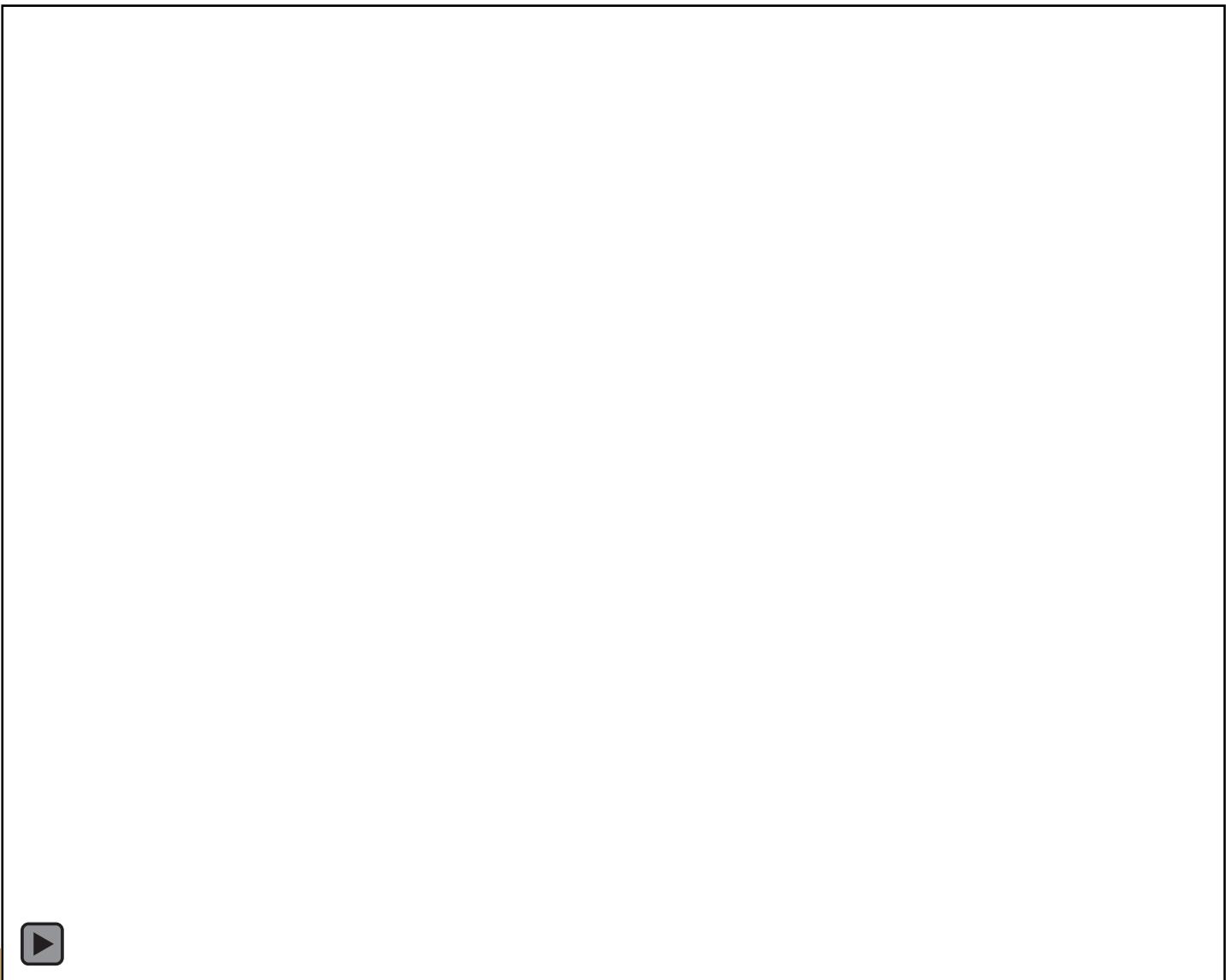
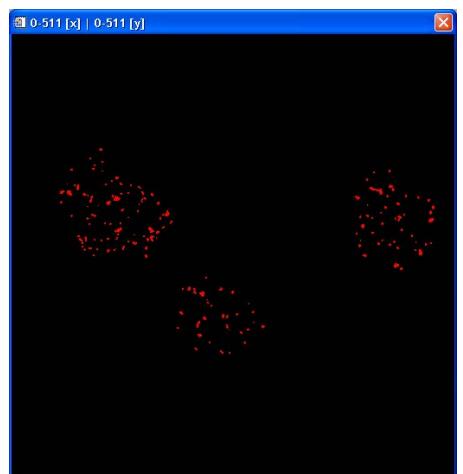
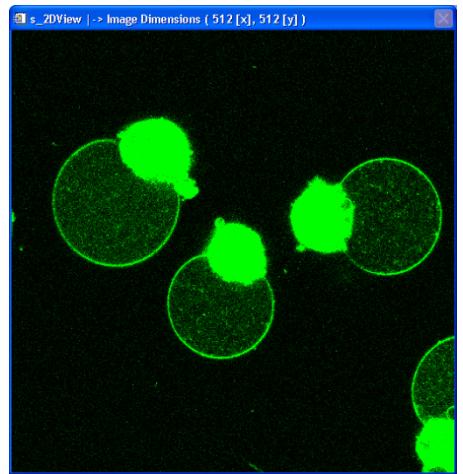


$$H = \begin{cases} \Theta & G \geq B \\ 2\pi - \Theta & G \leq B \end{cases}$$

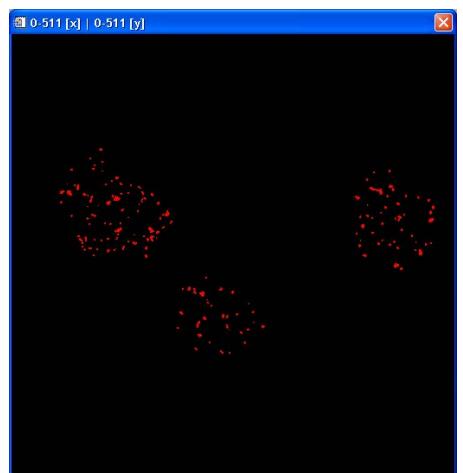
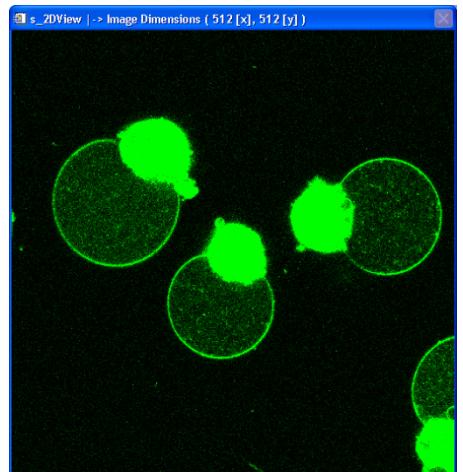
$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B}$$

$$I = \frac{R + G + B}{3}$$

$$\Theta = \arccos \left[\frac{1}{2} \sqrt{\frac{(R-G)(R-B)}{(R-G)^2 + (R-B)(G-B)}} \right]$$



Occlusions may
occur in 3D
visualization



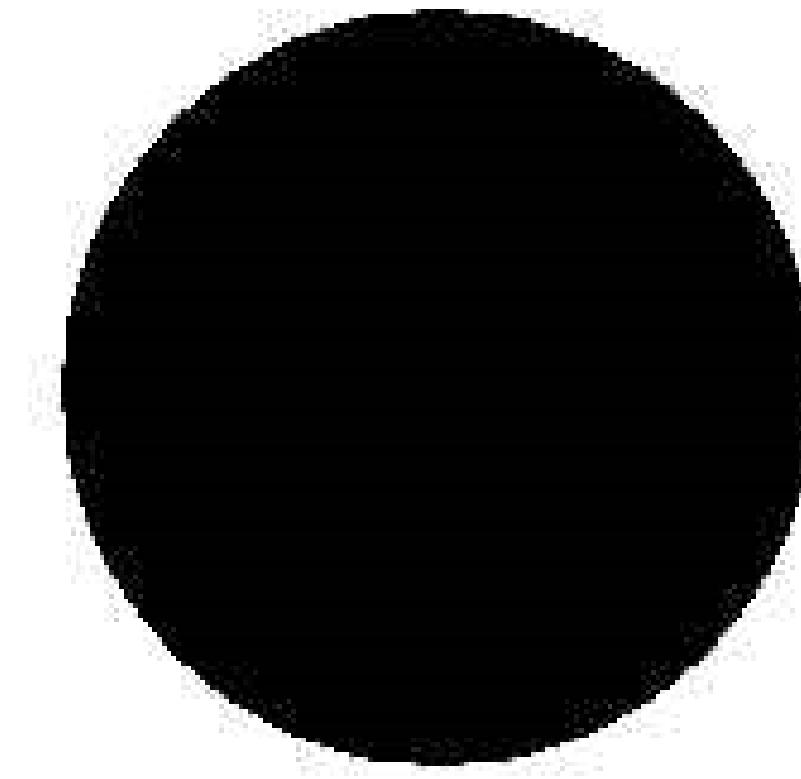
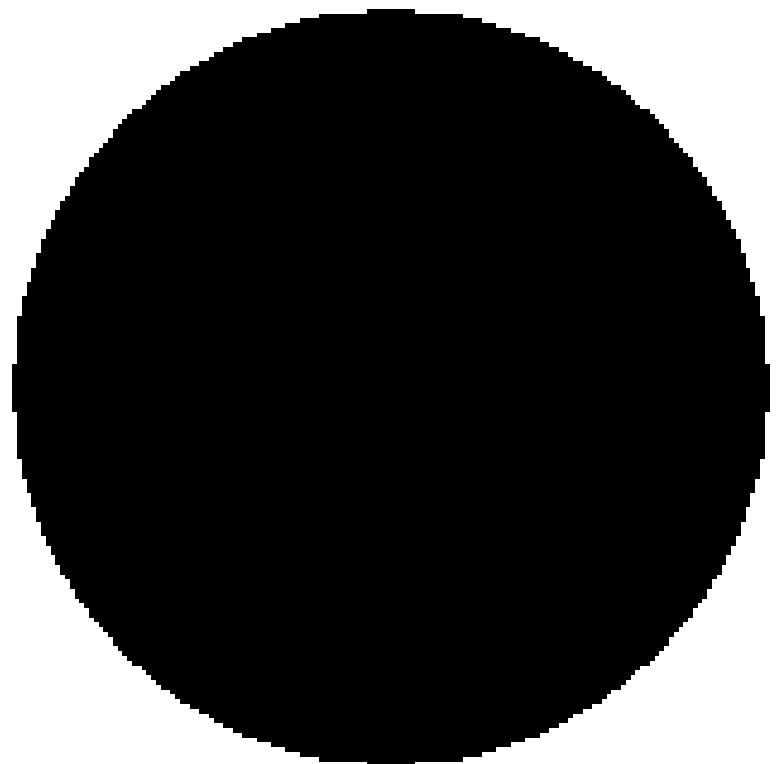
[R, G, B, α]

opacity values can
be associated to
pixels (or voxels in
3D) for visualization
purposes



- Function representation
 - Raster, SVG, base functions
- Color mode
 - grayscale
 - color (RGB, CMYK, HSV, Lab, etc.)
- Color depth (bit depth)
 - How many bits for how many values (e.g. 8 bits, 32 bits)
 - Number format
 - Integer (typically unsigned, e.g. TIFF)
 - Decimal (can be signed, e.g. ICS)
- Also important: storage mode
 - “Raw”: each pixel value is stored (lots of space)
 - Compressed, with or without information loss (e.g. JPEG *lossy* format, TIFF can be compressed or uncompressed)

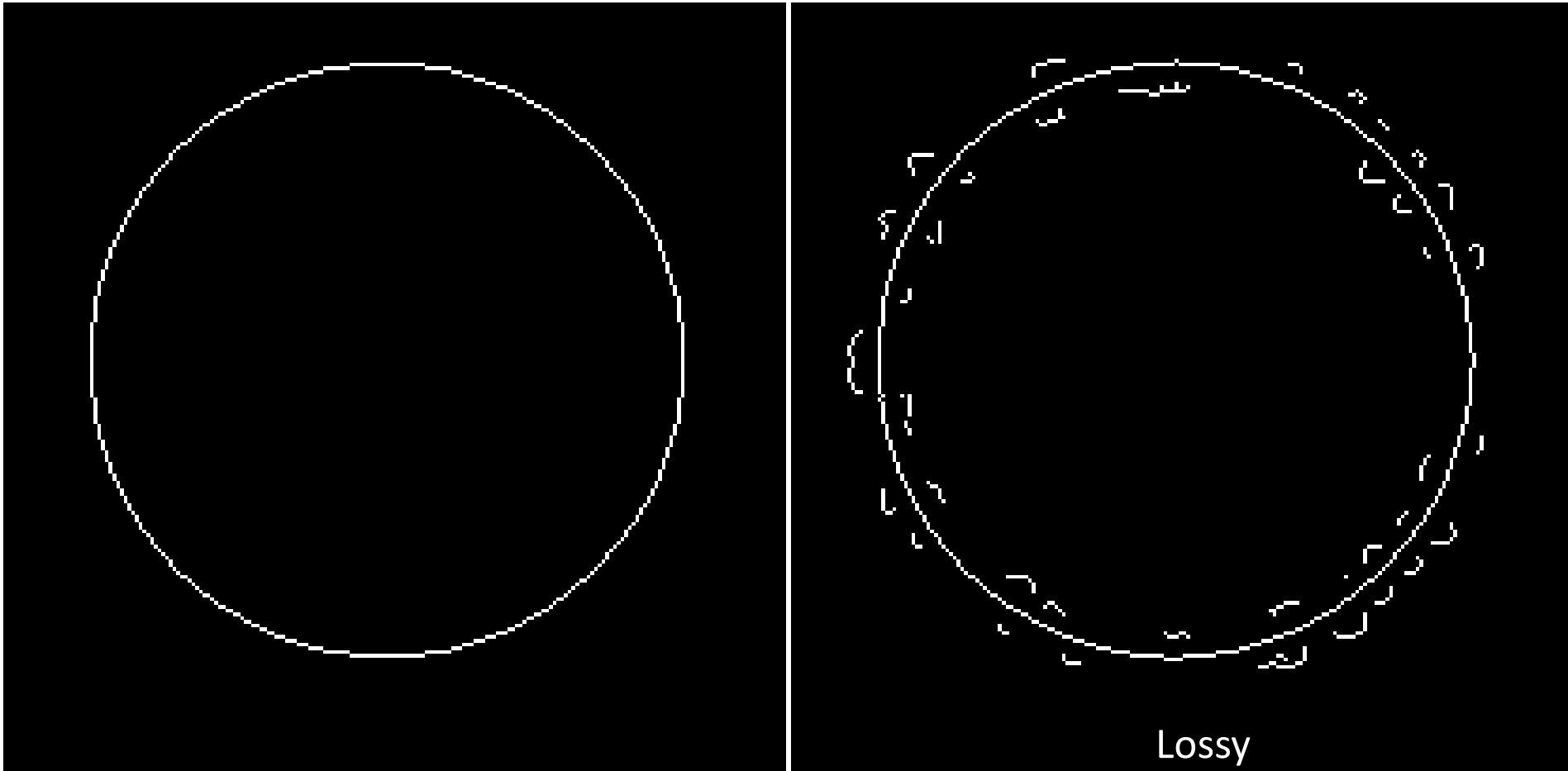
- Lossless (left) vs. Lossy (right)



Lossy

- Lossy vs. Lossless

Canny edge detector



Standard formats for 2D images

- PNG
- TIFF (2D/3D multi-channel, LUT support, some variants allow for compression)

Developing formats for large n-dimensional array containers

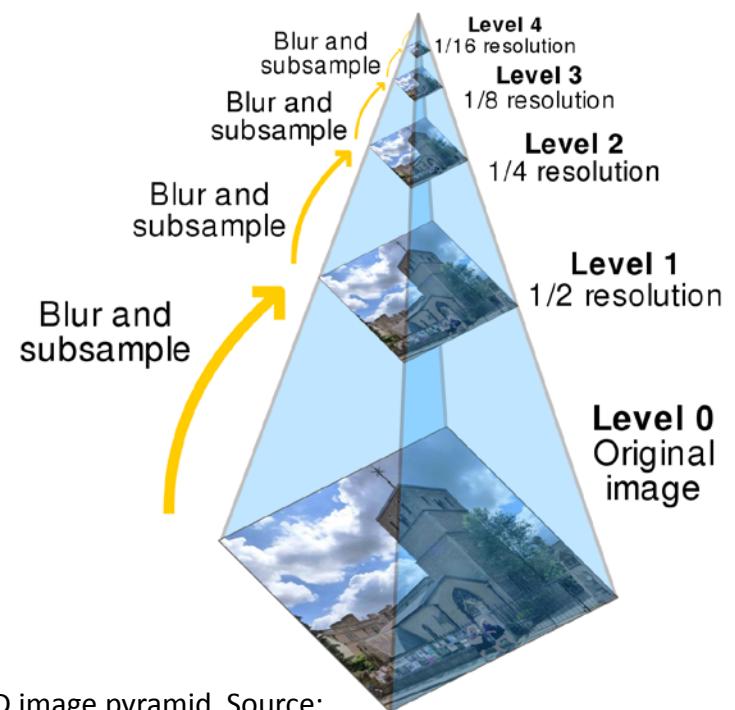
- N5
- Zarr

<https://github.com/saalfeldlab/n5>

<https://imagej.net/N5>

https://www.youtube.com/watch?v=lnqZ_NsgB4g

<https://zarr.readthedocs.io/en/stable/>



A 2D image pyramid. Source:
[https://en.wikipedia.org/wiki/Pyramid_\(image_processing\)](https://en.wikipedia.org/wiki/Pyramid_(image_processing))

- Image Analysis

The extraction of meaningful descriptions of features of interest from images

Adapted from
Young I, Gerbrands J, van Vliet L (1995)
Fundamentals of Image Processing. Delft: PH

- Some examples of analysis tasks
 - **Objects/regions identification (segmentation)**
 - Cells and/or their organelles
 - **Registration: image, region and/or feature “matching”**
 - Drift correction of the sample (from acquisition)
 - Relative speeds/displacements inside a given cell or reference system
 - Correspondence finding between images, objects or sections of these
 - **Motion estimation, object tracking**
 - Individual & collective migration
 - **Morphology, topology, texture characterization**
 - **Classification**
 - Detection of different populations, anomalies

Now: segmentation

Coming soon: motion
estimation & tracking,
morphology... characterization

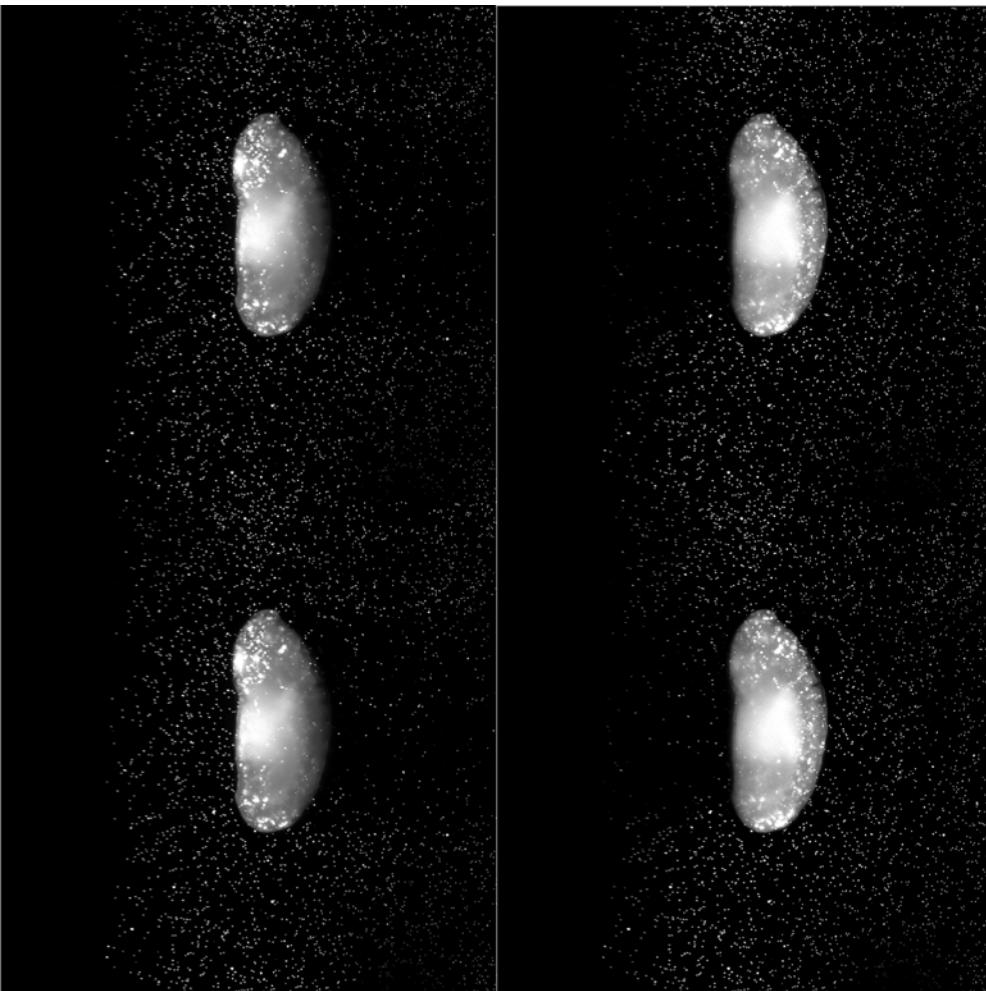
Next year(?): classification

- Segmentation

- The partitioning of a given image into regions of interest (ROIs) according to given criteria (e.g. color).
- After segmentation, further characterizations can be performed upon the resulting ROIs.

Shapiro LG and Stockman GC (2001):
“Computer Vision”, pp 279-325
New Jersey, Prentice-Hall

- Registration



In FIJI: Plugins->Registration

Additional FIJI plugins: StackReg, TurboReg

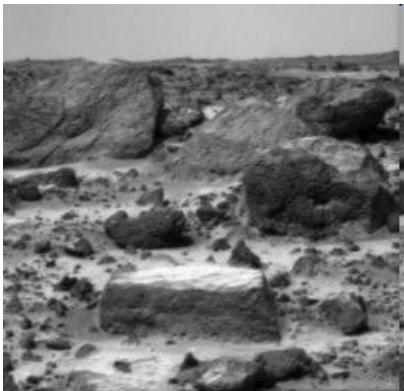
<http://bigwww.epfl.ch/algorithms.html>

Applied example: 4-view
LSFM image merging



C . Parada & C. Bolatto
(UdelaR, Uruguay), L. Alé
(LEO/SCIAN & BNI)

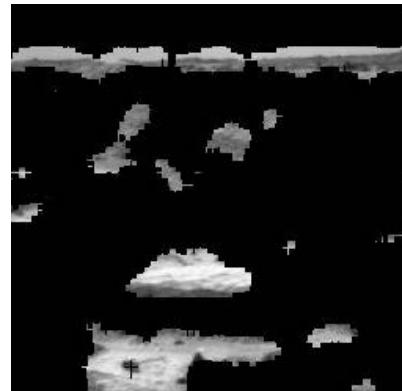
- Segmentation



Sol 3, Mars
Pathfinder Mission

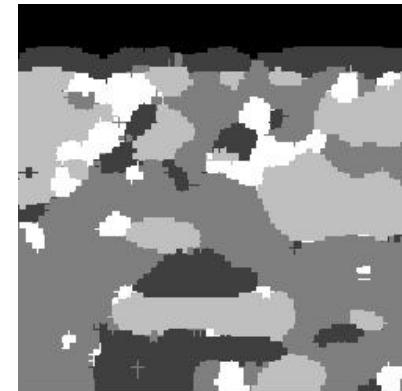


Sky / Flat



Dust / Horizon

...etc...



Final segmentation

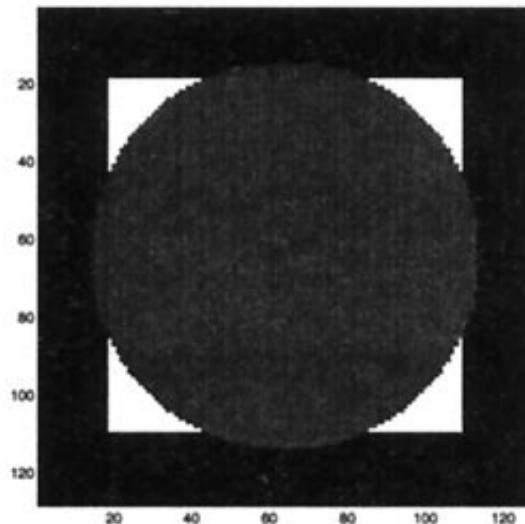
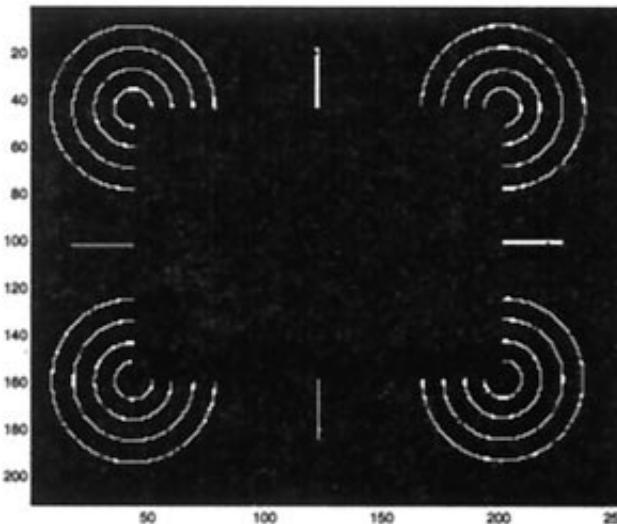
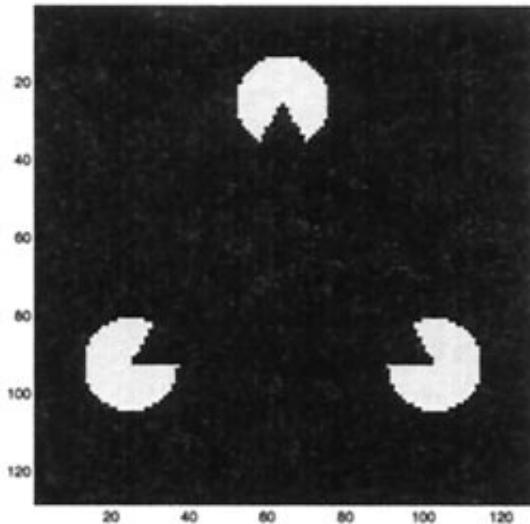
Source: NASA

- Identification (features, objects, scenes)



Scale Invariant **Feature** Transformation (SIFT), D Lowe (2004). Image from J Clemons (2009)

- Segmentation... ¿Which features? ¿Which criteria?
 - The segmentation problem has a subjective component

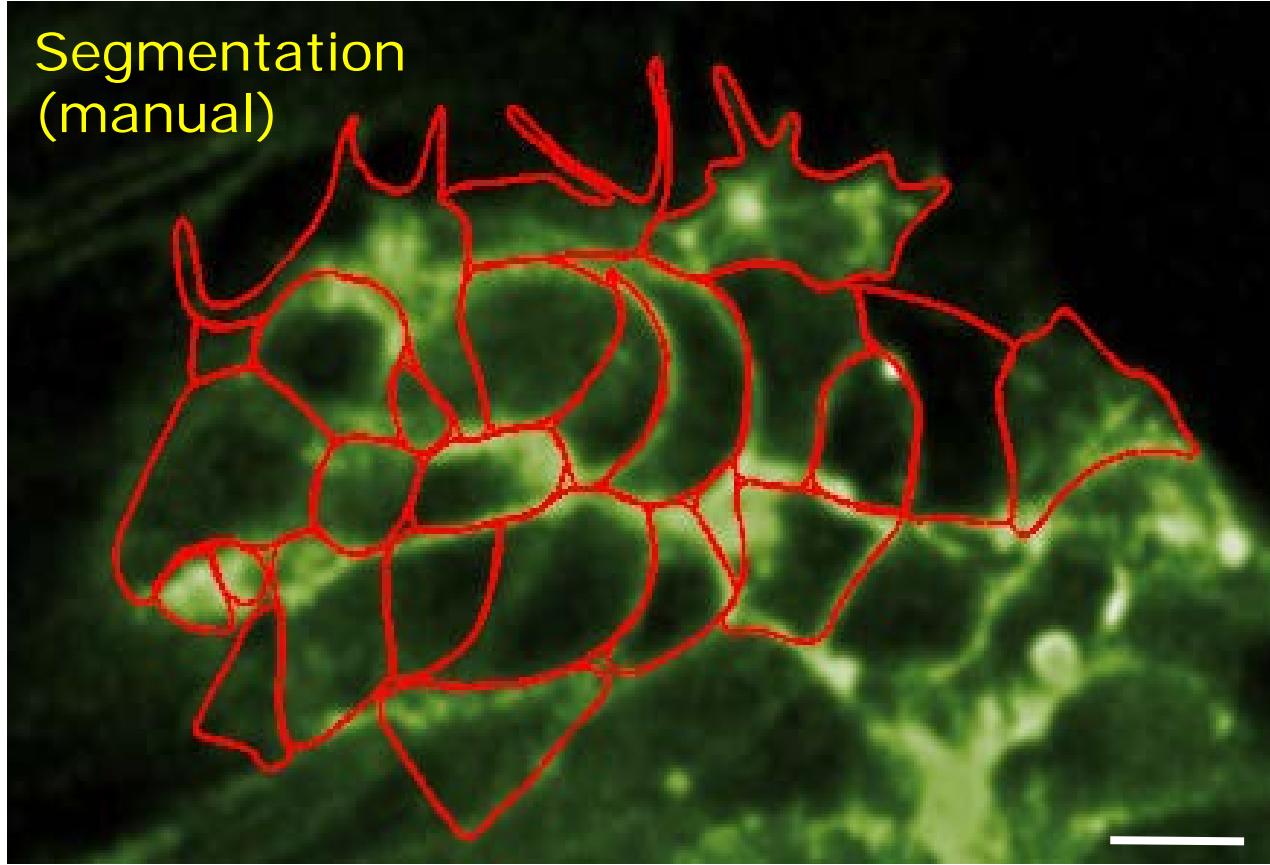


- ...not always (almost never) enough information for a 100% accurate segmentation.

Marr (1982) Vision

Sarti & Sethian (2000) Subjective surfaces. PNAS

Problem?



Problems

- Lack of absolute criteria or standards (Ground Truth, Gold Standard [1,2])
- Missing or erroneous information (e.g. non-specific markers in samples)
- What to do?
A “good” (i.e. carefully performed and controlled) acquisition to ease this task

[1] Jason D. Hipp et all. Tryggo: Old nurse for truth: The real truth about ground truth. New insights into the challenges of generating ground truth maps for WSI CAD algorithm evaluation. *Pathol. Inform* 2012, 3:8

[2] Luc Bidaut, Pierre Jannin. Biomedical multimodality imaging for clinical and research applications: principles, techniques and validation. In *Molecular Imaging:Computer Reconstruction and Practice (NATO Science for Peace and Security Series B: Physics and Biophysics)*, Springer, 2008, ISBN-13: 978-1402087516.

- David Marr. Vision
MIT press, 1982
- John Russ. The image processing handbook, 4th ed.
CRC Press, 2002
- Nixon, Aguado. Feature extraction & image processing, 1st | 2nd | 3rd ed.
Academic Press, 2002 | 2008 | 2012
- Aubert & Kornprobst. Mathematical Problems In Image Processing
Springer, 2006

Some free and open source software tools

- Java based (Java runtime required)
 - ImageJ (<http://rsbweb.nih.gov/ij/>, public domain)
 - FIJI (<http://fiji.sc>; GPL license)
 - Icy (<http://icy.bioimageanalysis.org>; GPLv3 license)
- Others
 - CellProfiler (<http://cellprofiler.org>; GPL, BSD licenses)
 - Slicer (www.slicer.org; BSD license)
 - IPOL (Image Processing Online): open access electronic journal with peer reviewed articles + code (languages: C/Python) + examples (www.ipol.im; BSD / GPL / LGPL licenses or similar)