CURSO DE POSTGRADO

**Procesamiento de Imágenes y Bioseñales I & II**

# Image Segmentation I

Basics


Jorge Jara W. [1,2]

[1] Department of Computer Science DCC, U. of Chile

[2] SCIAN-Lab, BNI

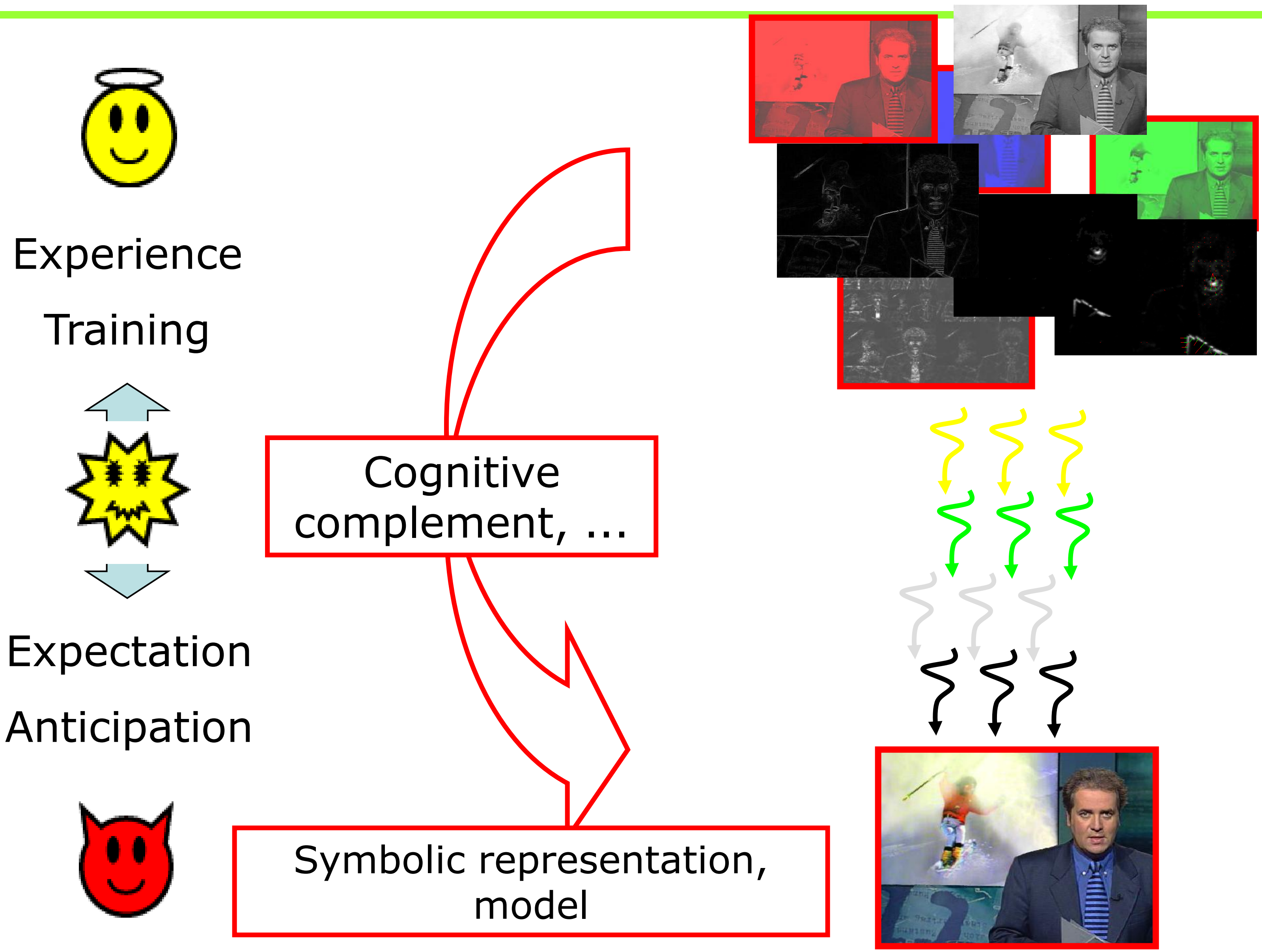
# Outline

1. Segmentation I

    - Digital image processing
    - Segmentation basics

2. Segmentation II
    - Advanced techniques

# Introduction

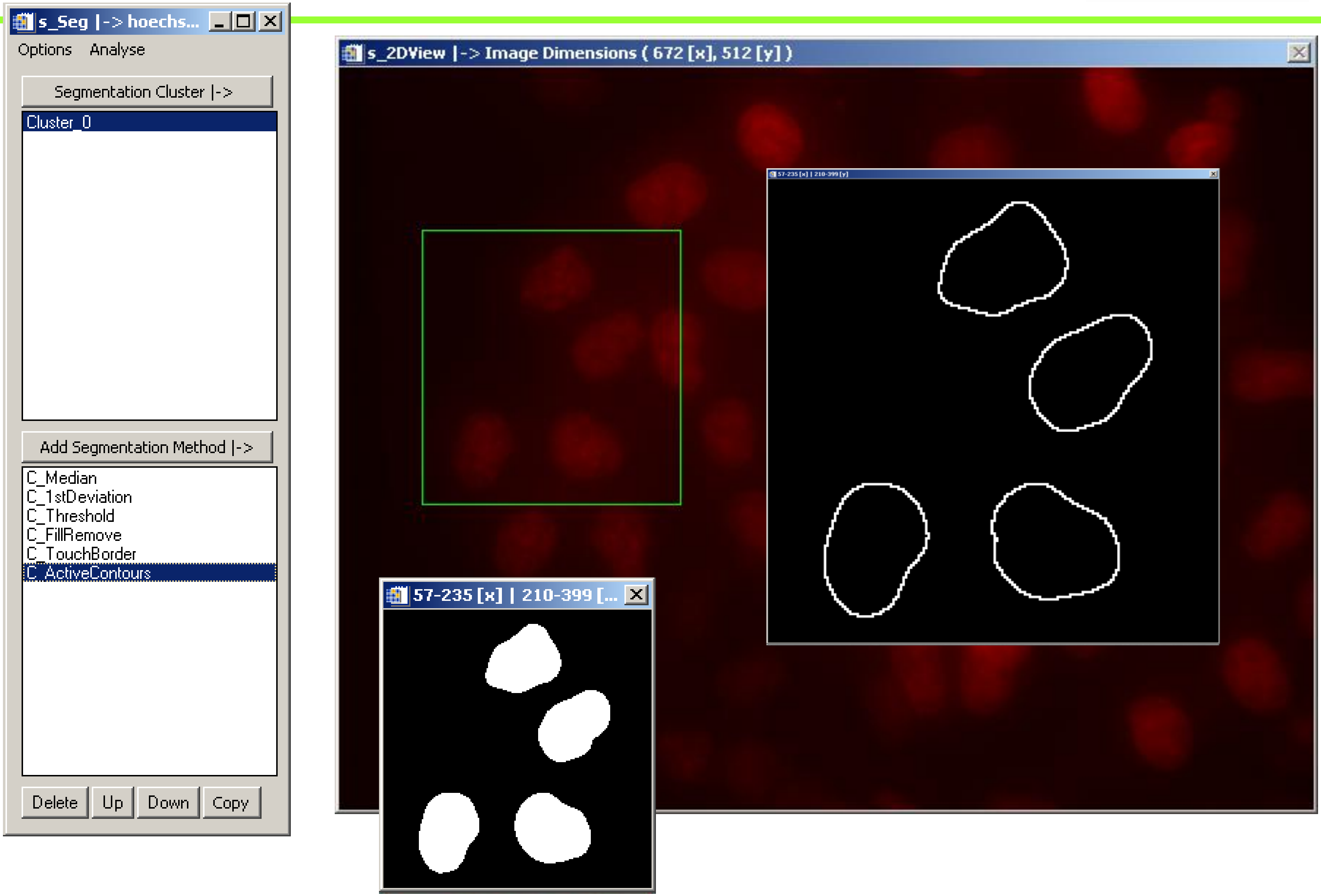


- Common criteria

Color similarity (regions)

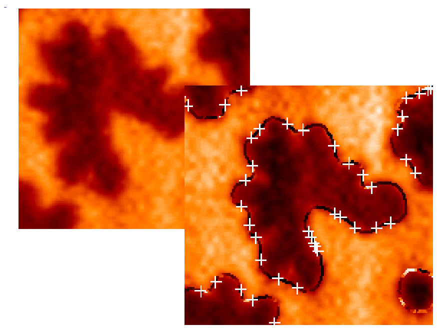
Color transitions or gradients (boundaries)

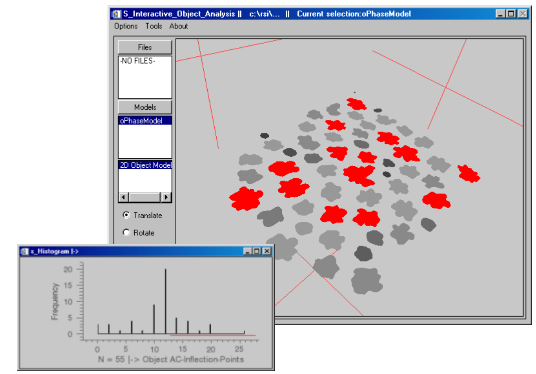
# Introduction



# Introduction

## 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

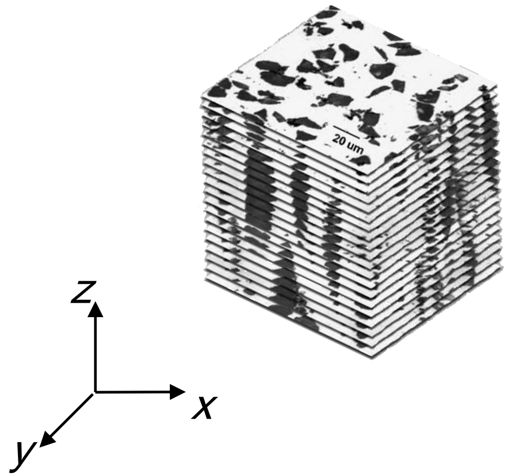
**images** → **images** → **images + descriptions**

# Introduction



Digital image processing

i.e. with math & computers (among others…)



# Introduction

- Digital (computational) image processing
  - Digital… discrete, finite
    - A discrete set is composed by elements that are "isolated" one from another
      Examples:
      - Natural numbers {1, 2, 3, …} (infinite set)
      - Natural numbers from 1 to 10 (finite set)
    - If not discrete? Continuum
      Example: real numbers in the [0,1] interval (infinite set)

# Introduction

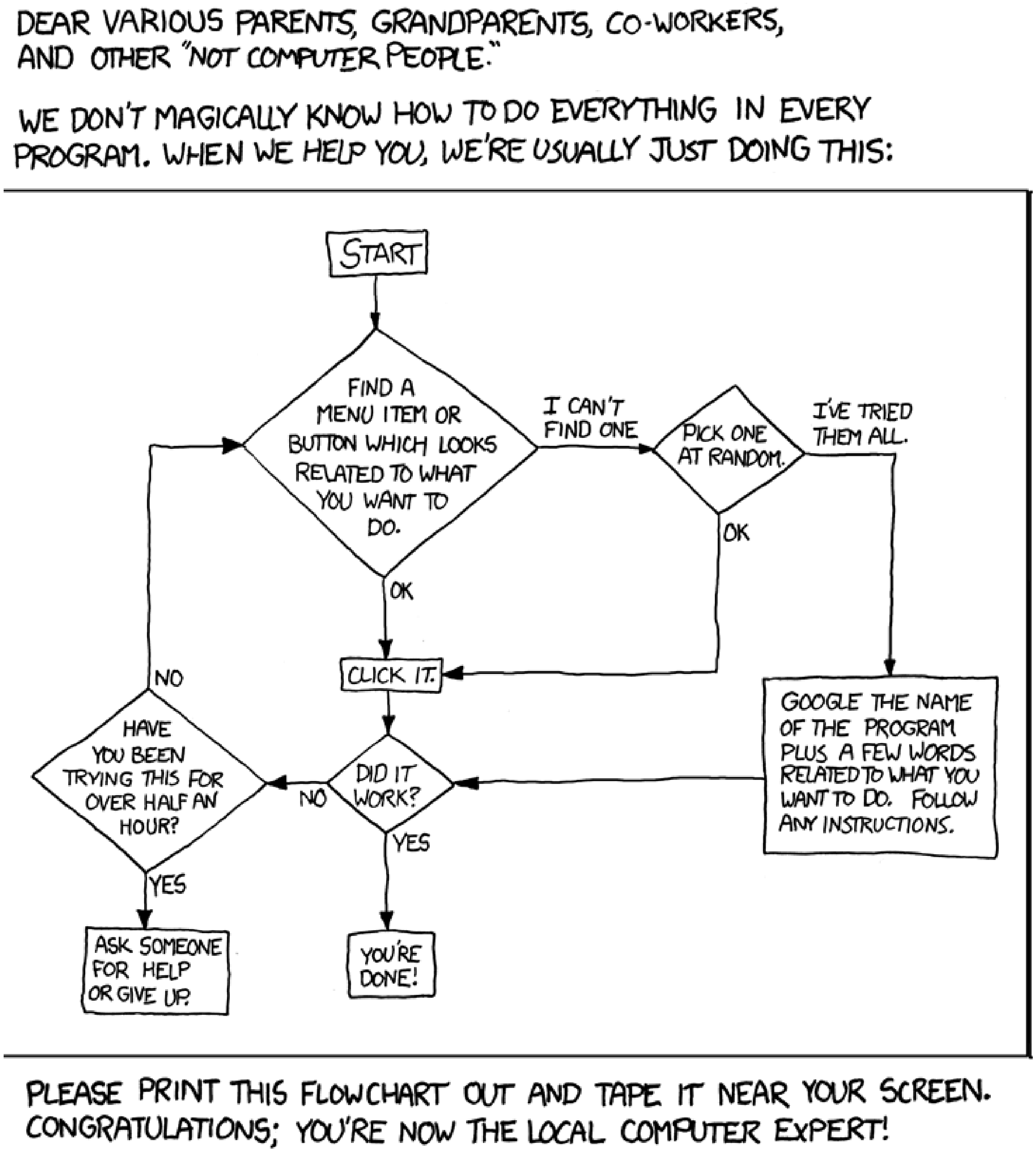
- "Digital" very often implies "computational", referring to formal methods and machines such as algorithms and computers
- An algorithm is a "well defined" procedure, i.e., a non-ambiguous set of:
  - inputs,
  - outputs, y
  - the steps to follow to produce the desired output(s)

**Algorithm** FindLargestNumberOfList

Input: a non-empty list o numbers, namely L.
Output: the largest number in L.

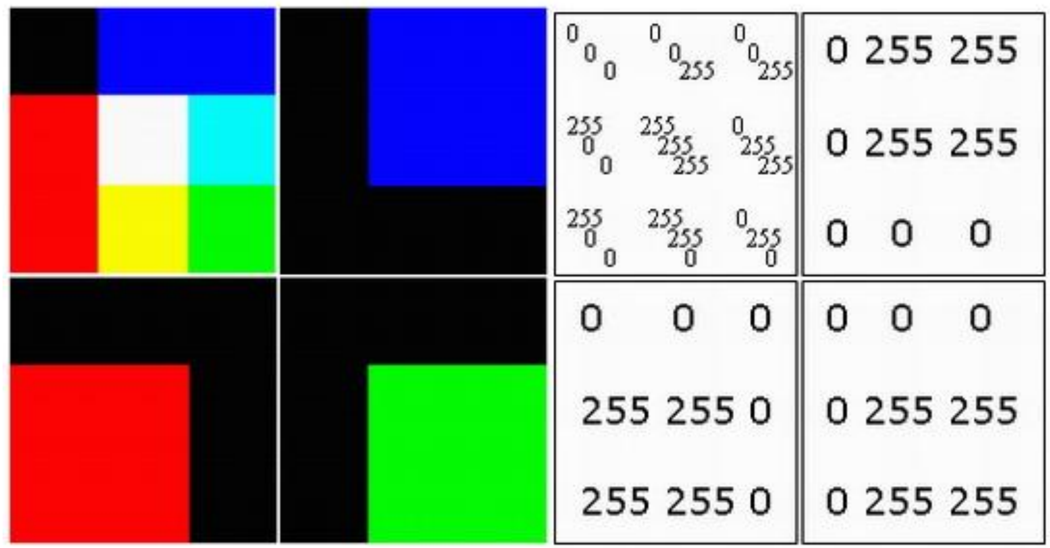
*largest* ← first element of *L*
**if** (numberOfElementsOf *L* ≥ 1) **then**
  **for each** *item* **in** *L*, **do**
    **if** *item* > *largest* **then** *largest* ← *item*
**return** *largest*

An algorithm written in pseudo-code




An algorithm depicted in a flowchart. From http://xkcd.com/627/

- 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 pixeles in cartesian coordinates (*x*, *y*)
  - A numeric value for brightness (intensity) or color is associated to each pixel




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

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

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

# Digital Images

- 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


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

# Digital Images



- …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.)

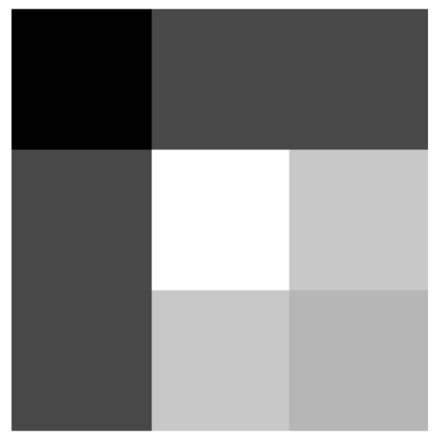
Aubert & Kornprobst (2006)
Mathematical Problems In Image Processing

# Digital Images

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

| | | |
|---|---|---|
| 0 | 85 | 85 |
| 85 | 255 | 170 |
| 85 | 170 | 85 |

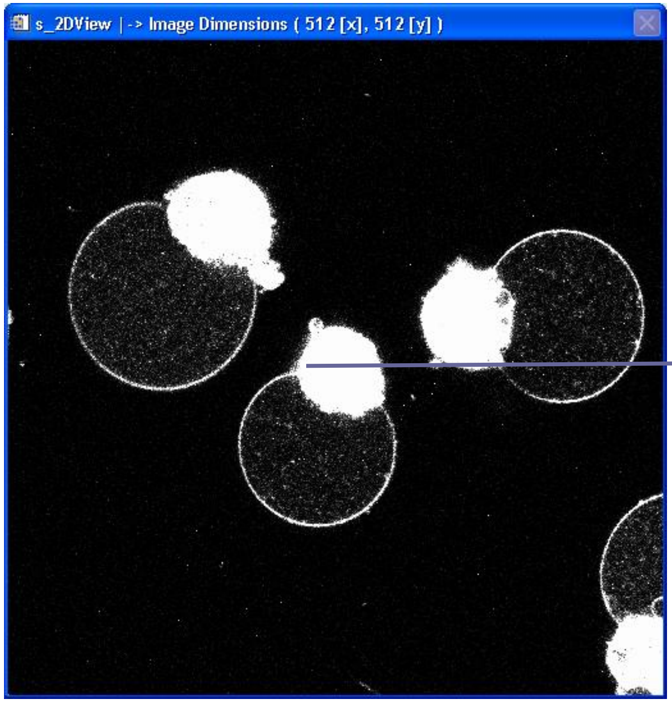
$I[x, y]$



# Digital Images



| 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) |



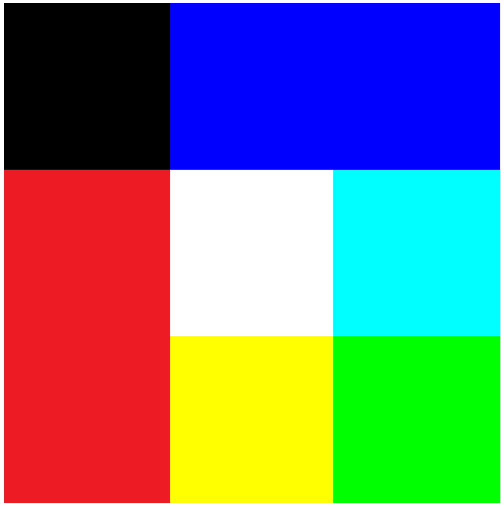

I(290,267) = 220

8 bit greyscale image

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

# Digital Images

- RGB image
  - Three channels for respective primary colors: Red, Green, Blue

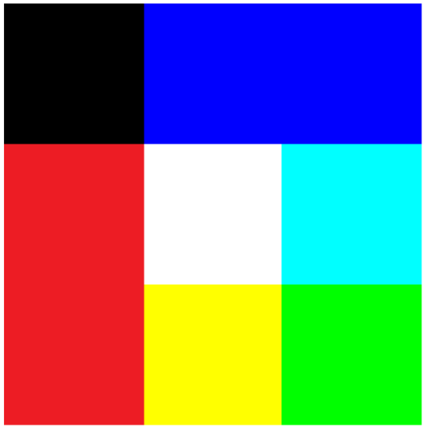


| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 255 | 255 |
| 255 | 255 | 0 |
| 0 | 255 | 255 |
| 0 | 255 | 255 |
| 255 | 255 | 0 |
| 0 | 255 | 255 |
| 0 | 0 | 0 |

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

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

- From RGB to greyscale
  - The conversion might not be an average…

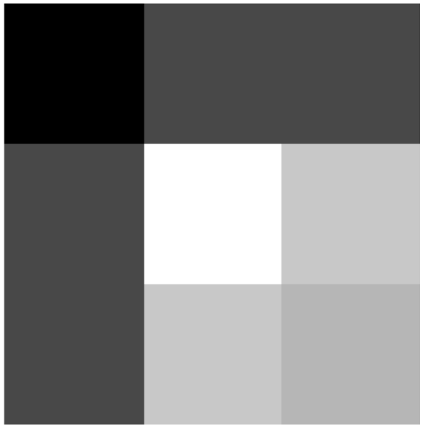


| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 0 |
| 0 | 255 | 255 |
| 255 | 255 | 0 |
| 0 | 255 | 255 |
| 0 | 255 | 255 |
| 255 | 255 | 0 |
| 0 | 255 | 255 |
| 0 | 0 | 0 |

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

| | | |
|---|---|---|
| 0 | 85 | 85 |
| 85 | 255 | 170 |
| 85 | 170 | 85 |

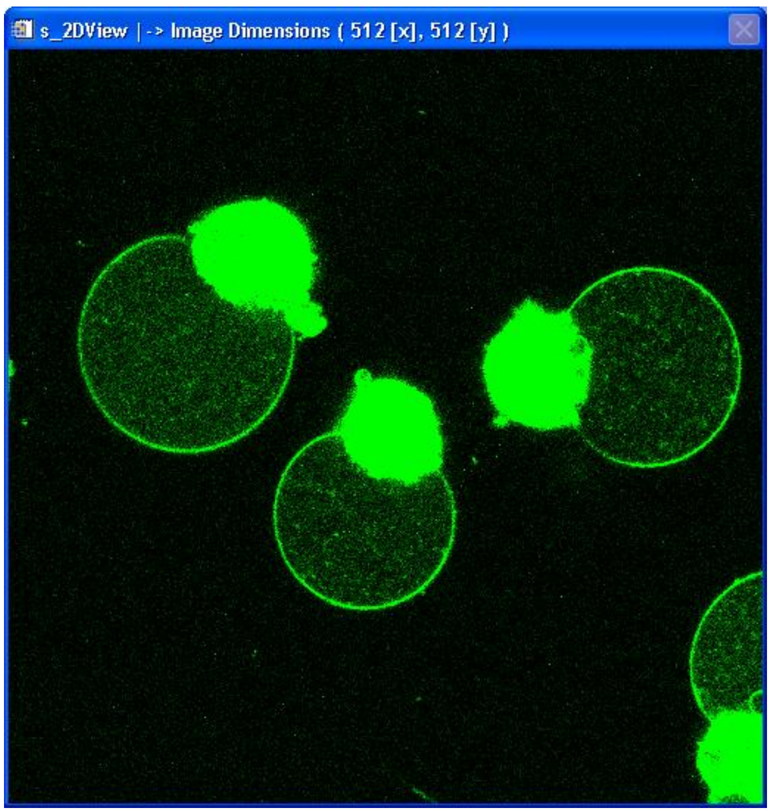
$I[x,y]$

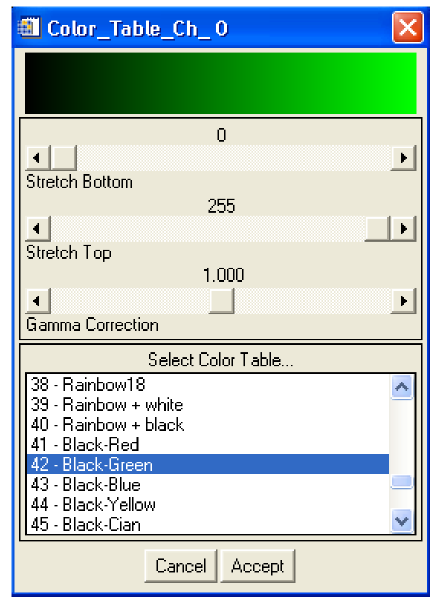


How good is the human eye resolving colors in R, G or B tones?

# Digital Images

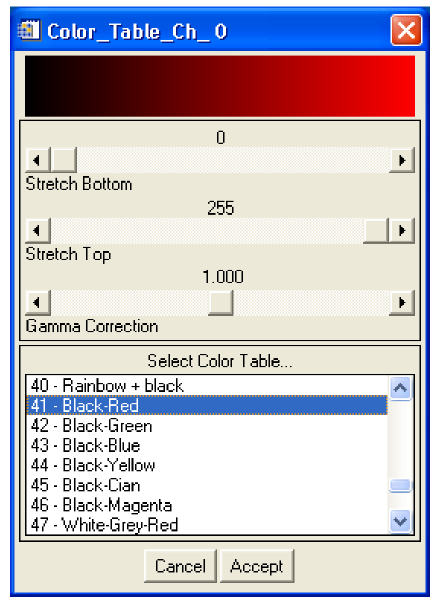
- It is possible to define color tables (or lookup tables, LUTs) for visualization purposes. A grayscale image can be displayed using a green scale.




| r | g | b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 2 | 0 |
| : | : | : |
| : | : | : |
| : | : | : |
| : | : | : |
| 0 | 200 | 0 |
| : | : | : |
| : | : | : |
| 0 | 255 | 0 |

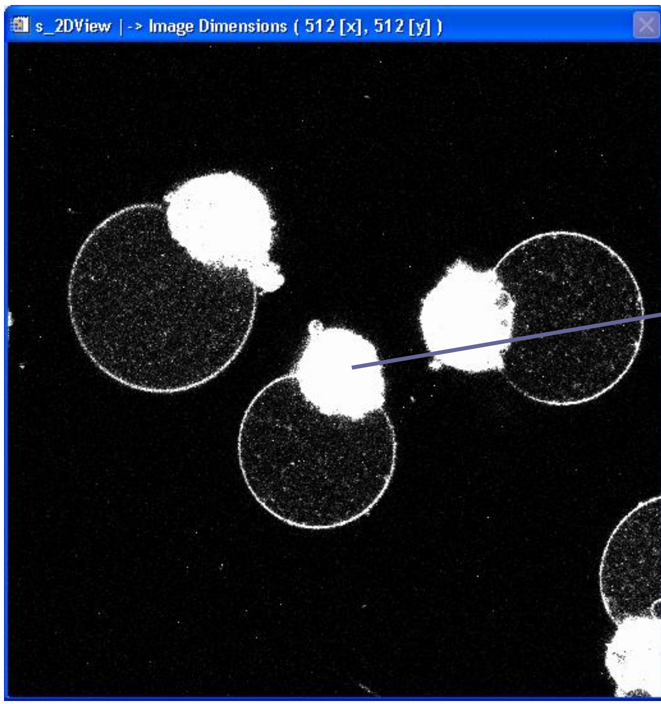
- Red scale…

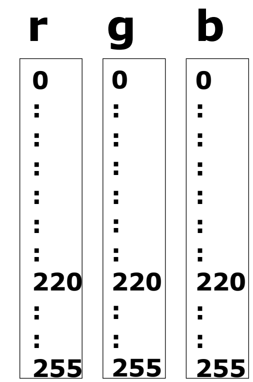

| r | g | b |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 2 | 0 | 0 |
| : | : | : |
| : | : | : |
| : | : | : |
| : | : | : |
| 220 | 0 | 0 |
| : | : | : |
| : | : | : |
| 255 | 0 | 0 |

# Digital Images

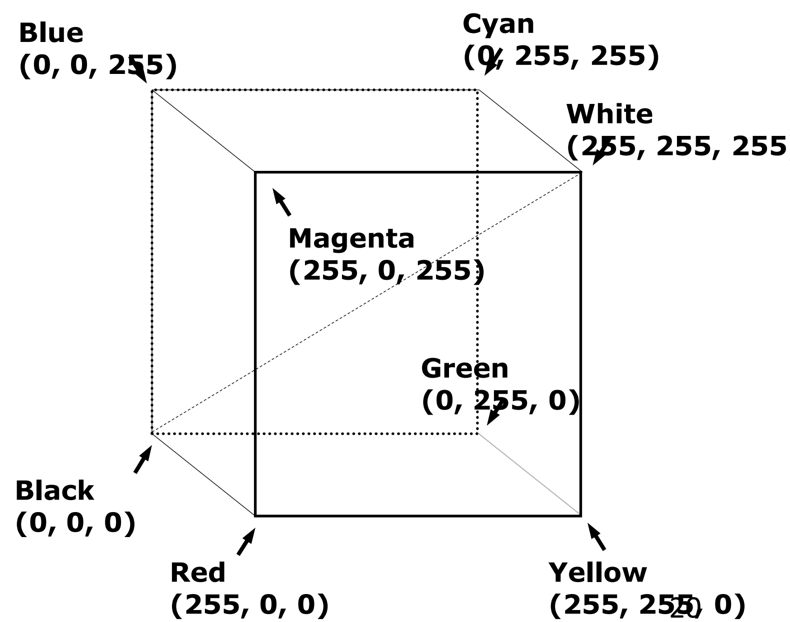
- ...or any custom color table




$I(290,267) = 220$

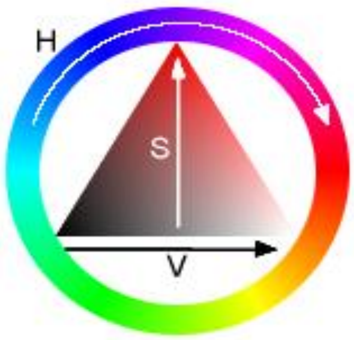



Red = $[r_0, r_1, \ldots, r_{255}]$
Green = $[g_0, g_1, \ldots, g_{255}]$
Blue = $[b_0, b_1, \ldots, b_{255}]$



# HSV (hue, saturation, value) model

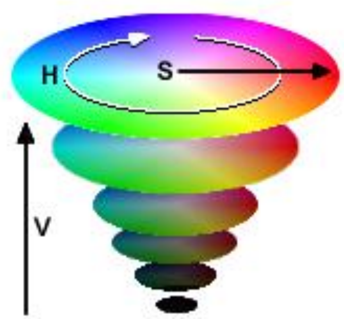
http://en.wikipedia.org/wiki/HSV_color_space




- **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.

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

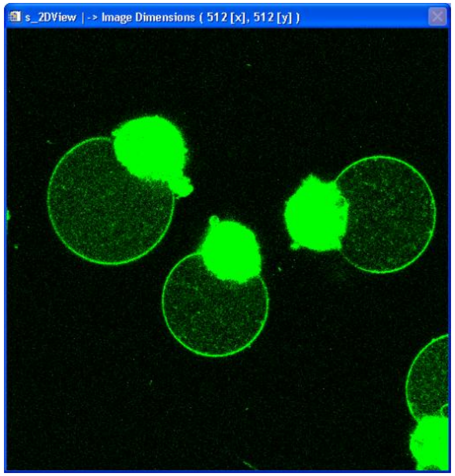
$$S = 1 - 3\frac{\min(R,G,B)}{R+G+B}$$

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

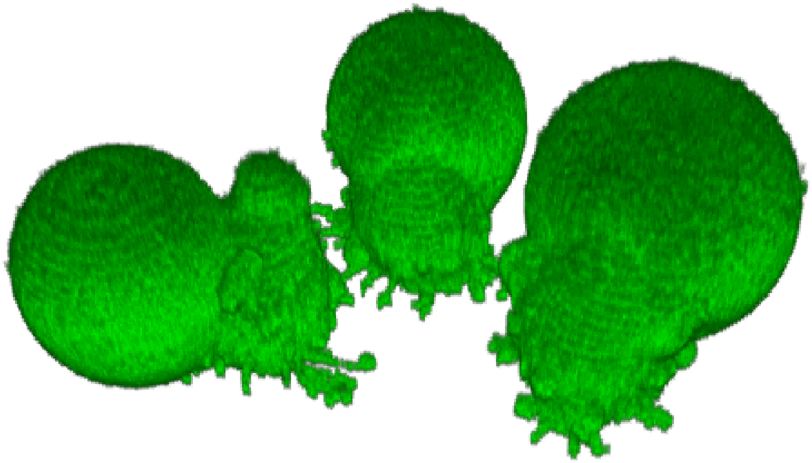
$$\Theta = \arccos\left[\frac{1}{2}\frac{(R-G)(R-B)}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right]$$

# Digital Images





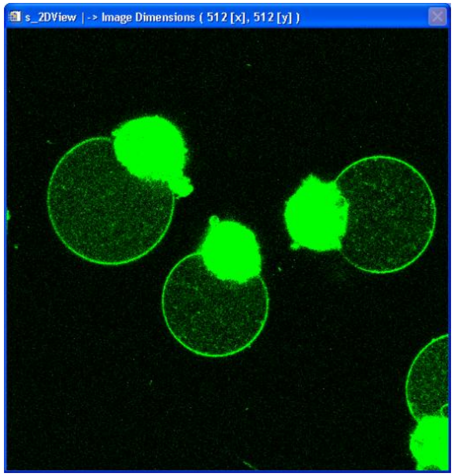




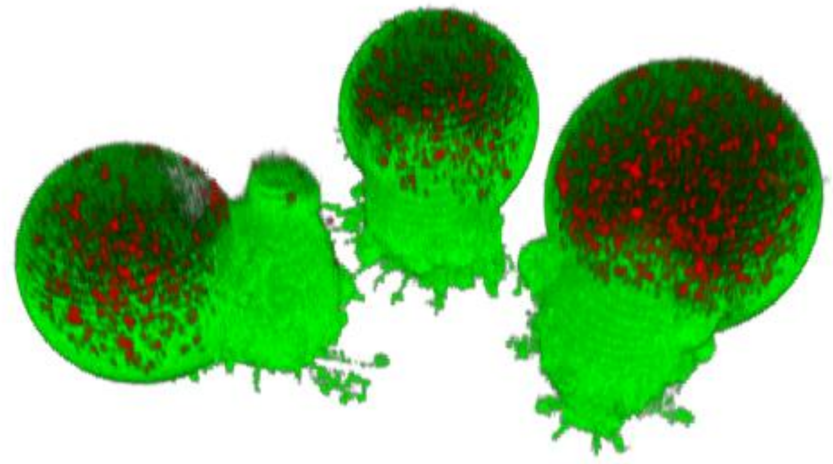
Occlusions may occur in 3D visualization

# Digital Images









[R, G, B, α]

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

# Digital Images

- 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)

- 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

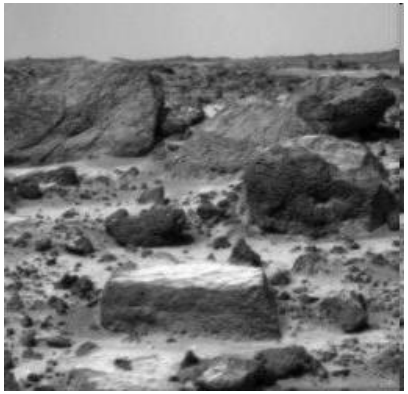
  - Morphology, topology, texture characterization

  - Classification
    - Detection of different populations, anomalies

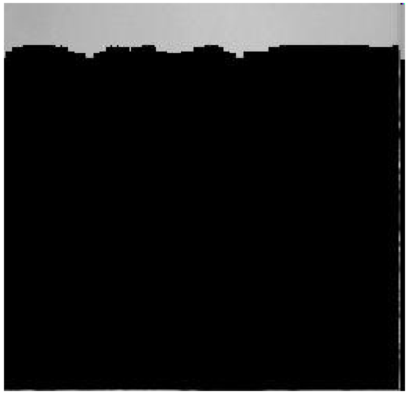
- Segmentation
  - The partitioning of a given image into regions of interest (ROIs) according to given criteria (e.g. color).
  - After segmentation, further characterizations can be performed upon the resulting ROIs.

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

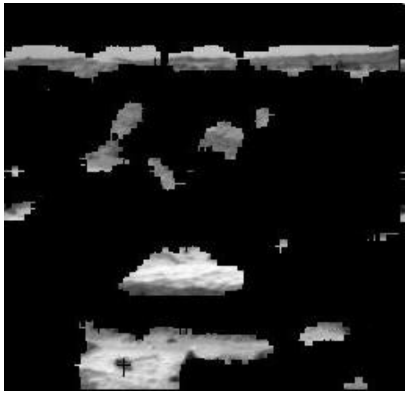
# Segmentation



Sol 3, Mars
Pathfinder Mission

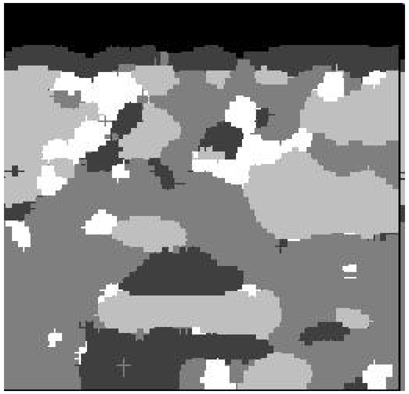


Sky / Flat



Dust / Horizon

...etc...



Final segmentation

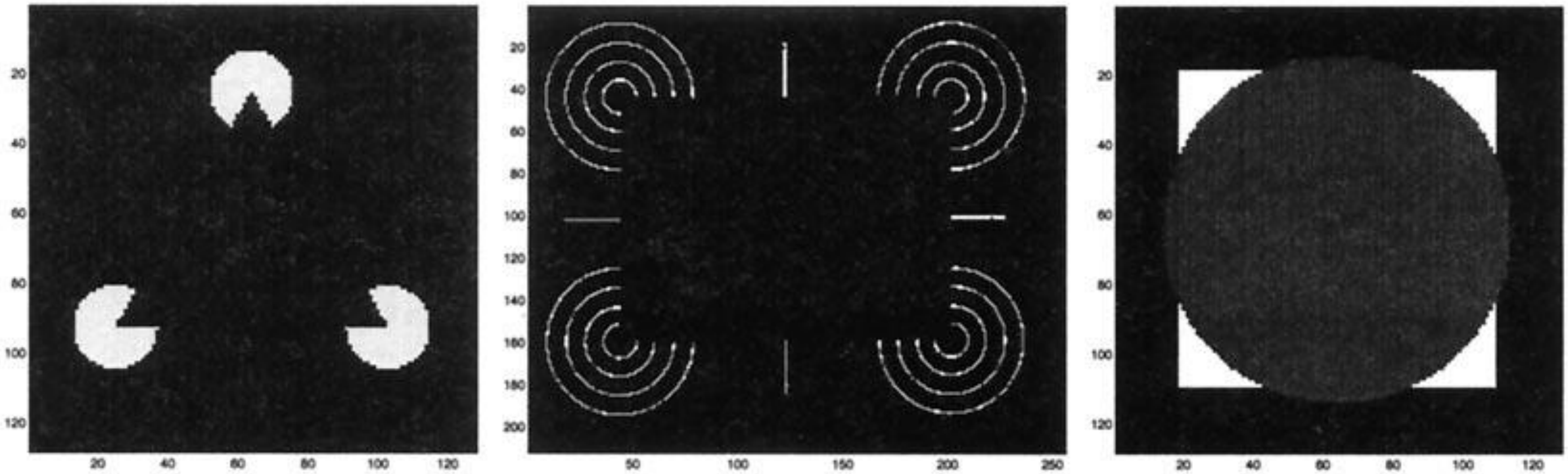
Images from NASA

# Segmentation

- Not only objects as ROIs, but features…


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

- Segmenting… ¿Which features? ¿Which criteria?

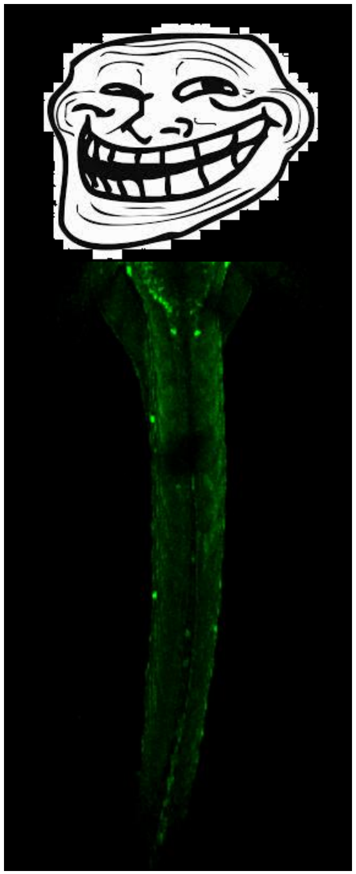



  – …not always (almost never) enough information for a 100% accurate segmentation.

Sarti & Sethian (2000) Subjective surfaces. PNAS

# Segmentation

## Problem?




CG Lemus (LEO, SCIAN-Lab, BNI)

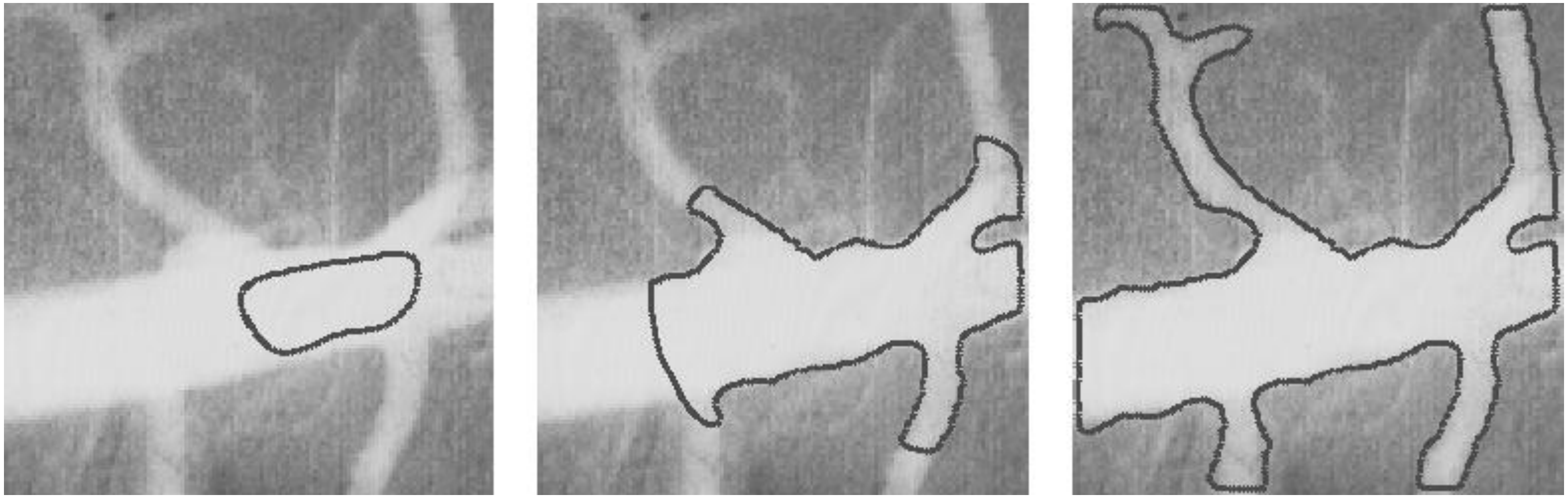
## Problems

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

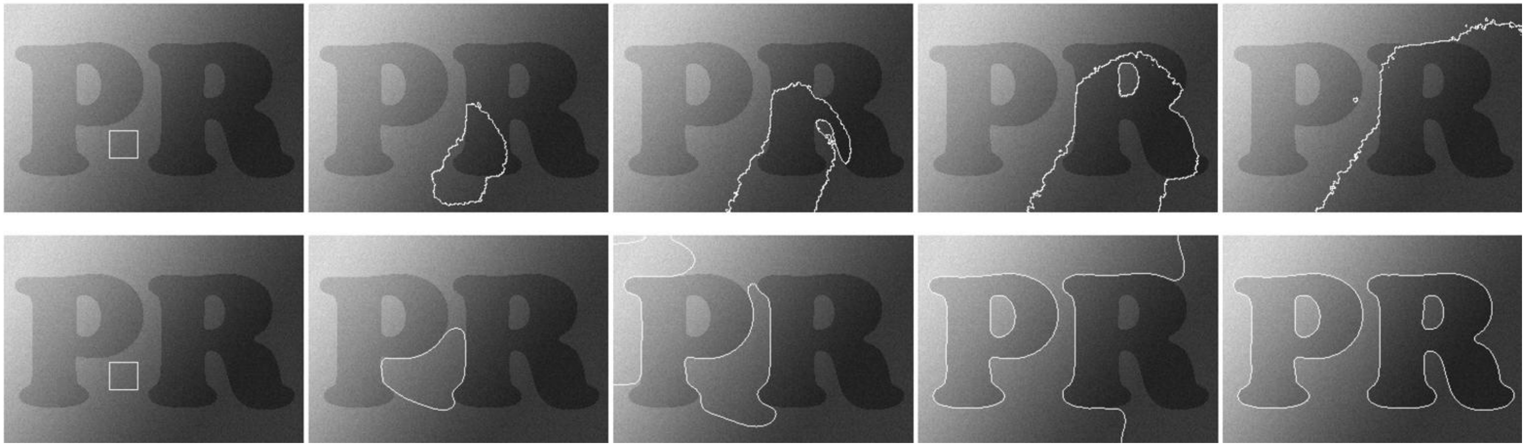
[1] Jason D. Hipp et all. Tryggo: Old norse for truth: The real truth about ground truth. New insights into the challenges of generating ground truth maps for WSI CAD algorithm evaluation. Pathol. Inform 2012, 3:8

[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.

# Segmentation models regarded as “good” for a given scenario can be “bad” for other



J A Sethian – Fast marching and level set methods
http://math.berkeley.edu/~sethian/2006/Applications/Medical_Imaging/artery.html



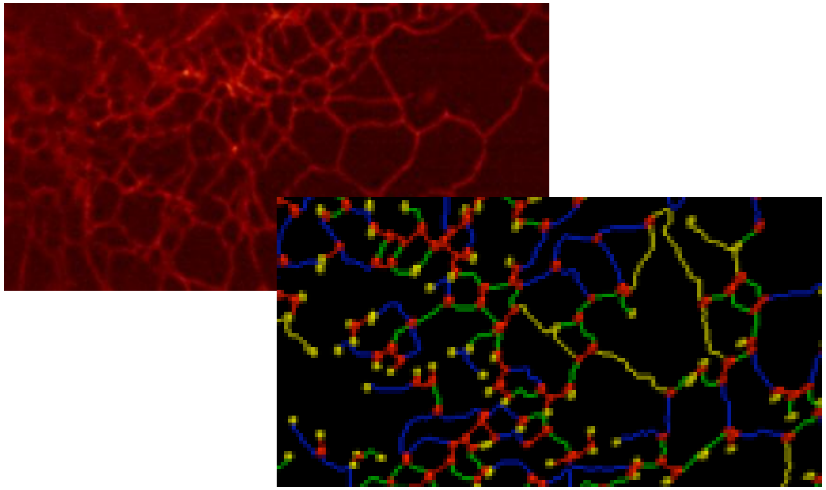
X Xie (2010) Magnetostatic Active Contours

# Segmentation

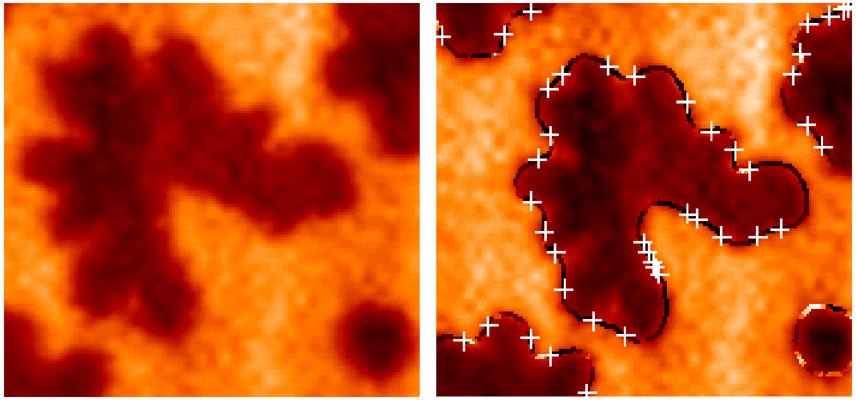
- Segmentation is the first step toward further quantifications
  - In addition to images, ROI models and data structures can suit for different types of descriptions

Parameter estimation…

- Size: area, perimeter
- Boundary: inflections, shape
- Topology: connectivity, endpoints



Endoplasmic reticulum in a COS-7 cell
O Ramírez, L Alcayaga (2012)



Lipid monolayers
J Jara (2006), Fanani et al (2010)

# Segmentation



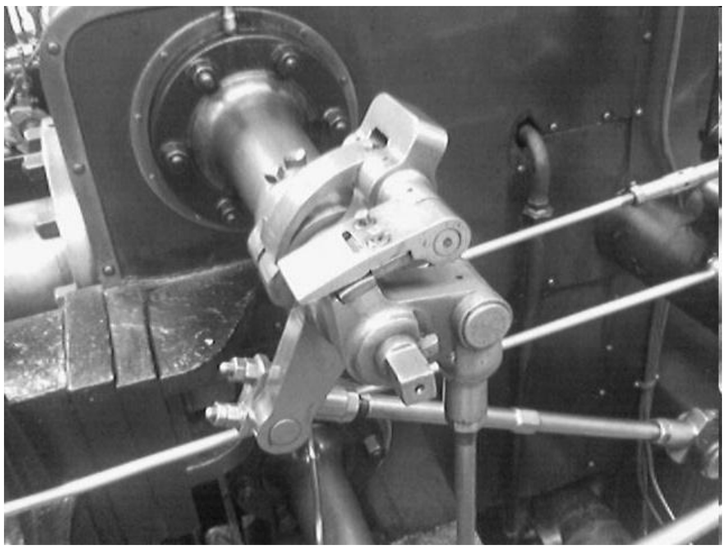
1. Classic approaches (filters)
   - Thresholding
   - Matrix convolution filters
   - Mathematical morphology
   - Fourier

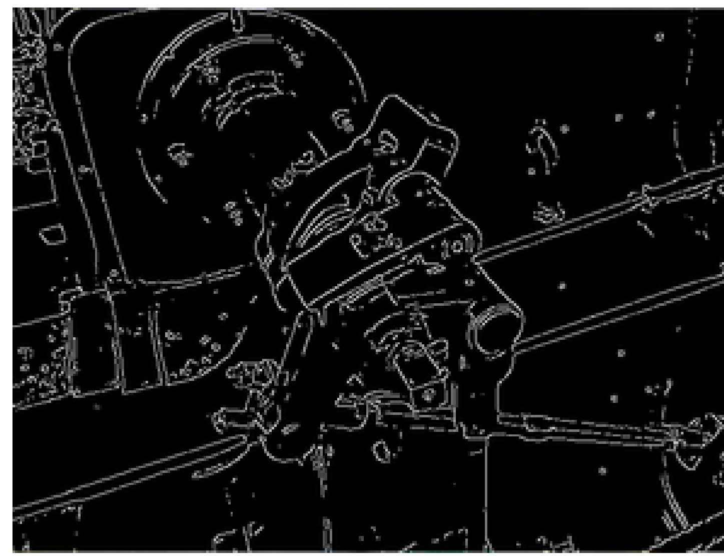
2. Advanced approaches
   - Shape priors (*pattern matching*)
   - Clustering methods (graph cuts, entropy)
   - Deformable models (active contours)
     - parametric
     - implicit

# Segmentation – basics

- Filter

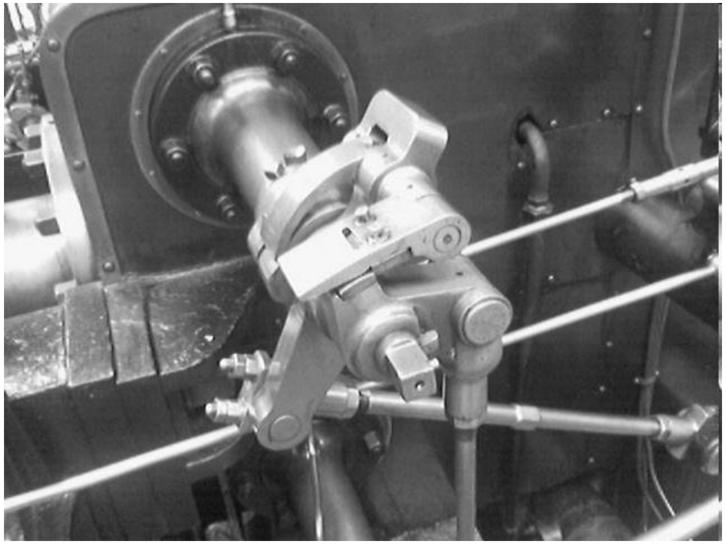
image → filter → filtered image

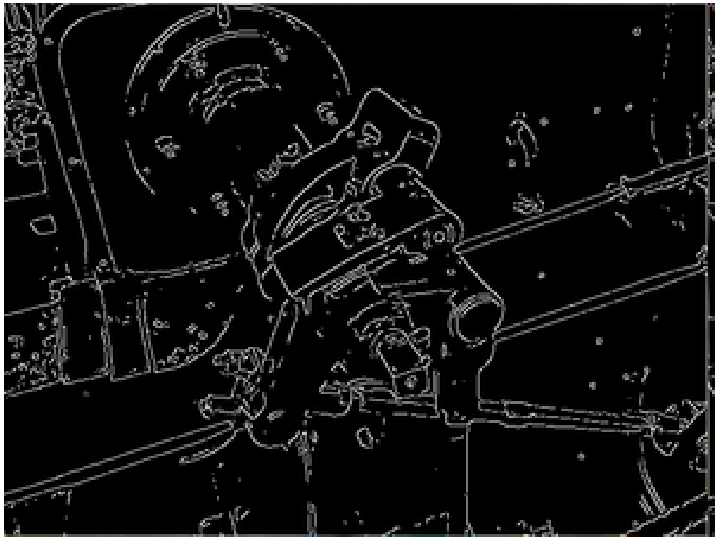


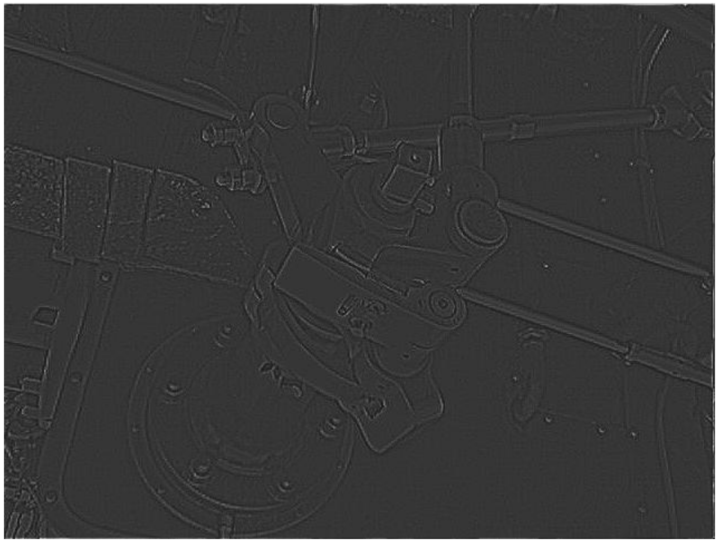


Imágenes from Wikipedia

- A sample filter...



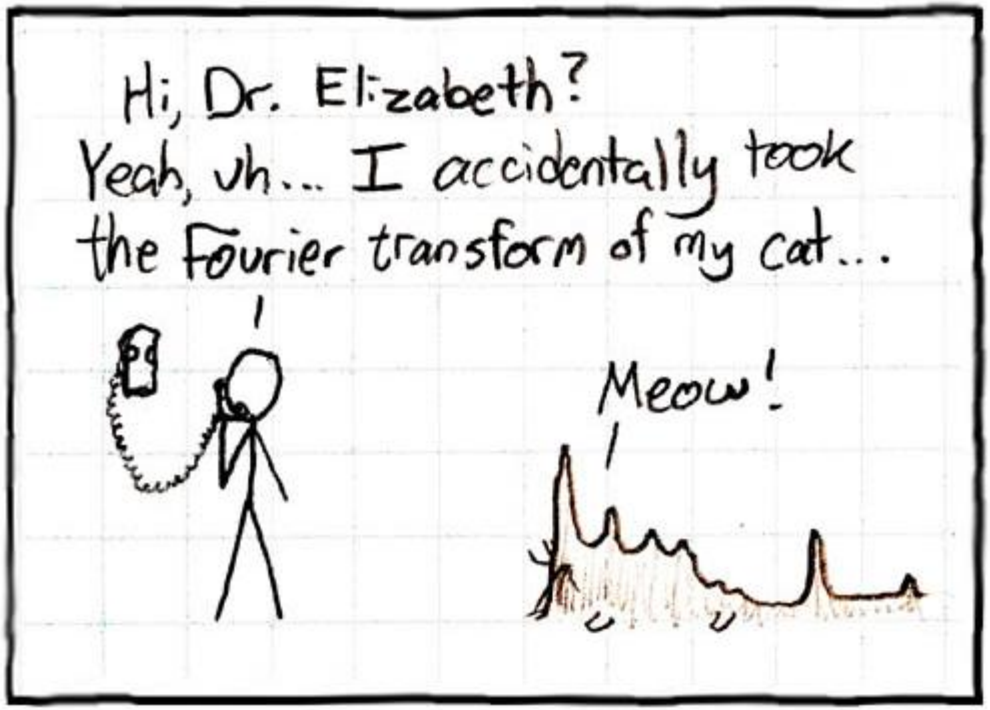




# Domain transformation… frequency (Fourier), Wavelets, …

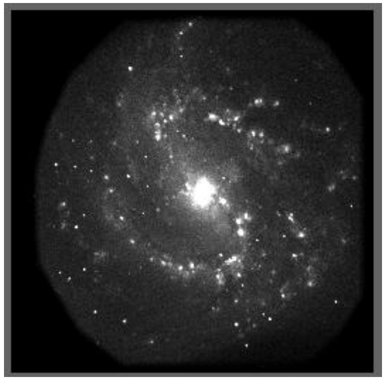
## Jean-Baptiste Joseph Fourier

**Jean-Baptiste-Joseph Fourier** (21 de marzo 1768 en Auxerre - 16 de mayo 1830 en París), matemático y físico francés conocido por sus trabajos sobre la descomposición de funciones periódicas en series trigonométricas convergentes llamadas Series de Fourier, método con el cual consiguió resolver la ecuación del calor. La transformada de Fourier recibe su nombre en su honor. Fue el primero en dar una explicación científica al efecto invernadero en un tratado. Se le dedicó un asteroide que lleva su nombre y que fue descubierto en 1992.

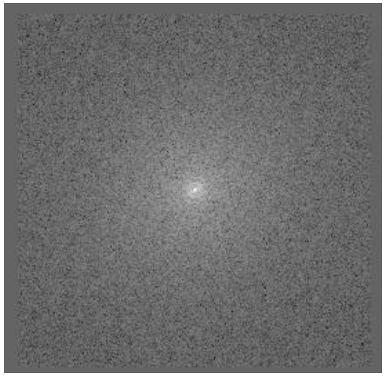


## Jean-Baptiste Joseph Fourier

**Jean-Baptiste-Joseph Fourier** (21 de marzo 1768 en Auxerre - 16 de mayo 1830 en París), matemático y físico francés conocido por sus trabajos sobre la descomposición de funciones periódicas en series trigonométricas convergentes llamadas Series de Fourier, método con el cual consiguió resolver la ecuación del calor. La transformada de Fourier recibe su nombre en su honor. Fue el primero en dar una explicación científica al efecto invernadero en un tratado. Se le dedicó un asteroide que lleva su nombre y que fue descubierto en 1992.



$$F(k,l) = \frac{1}{N^2} \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} f(a,b)\, e^{-\iota 2\pi\left(\frac{ka}{N} + \frac{lb}{N}\right)}$$



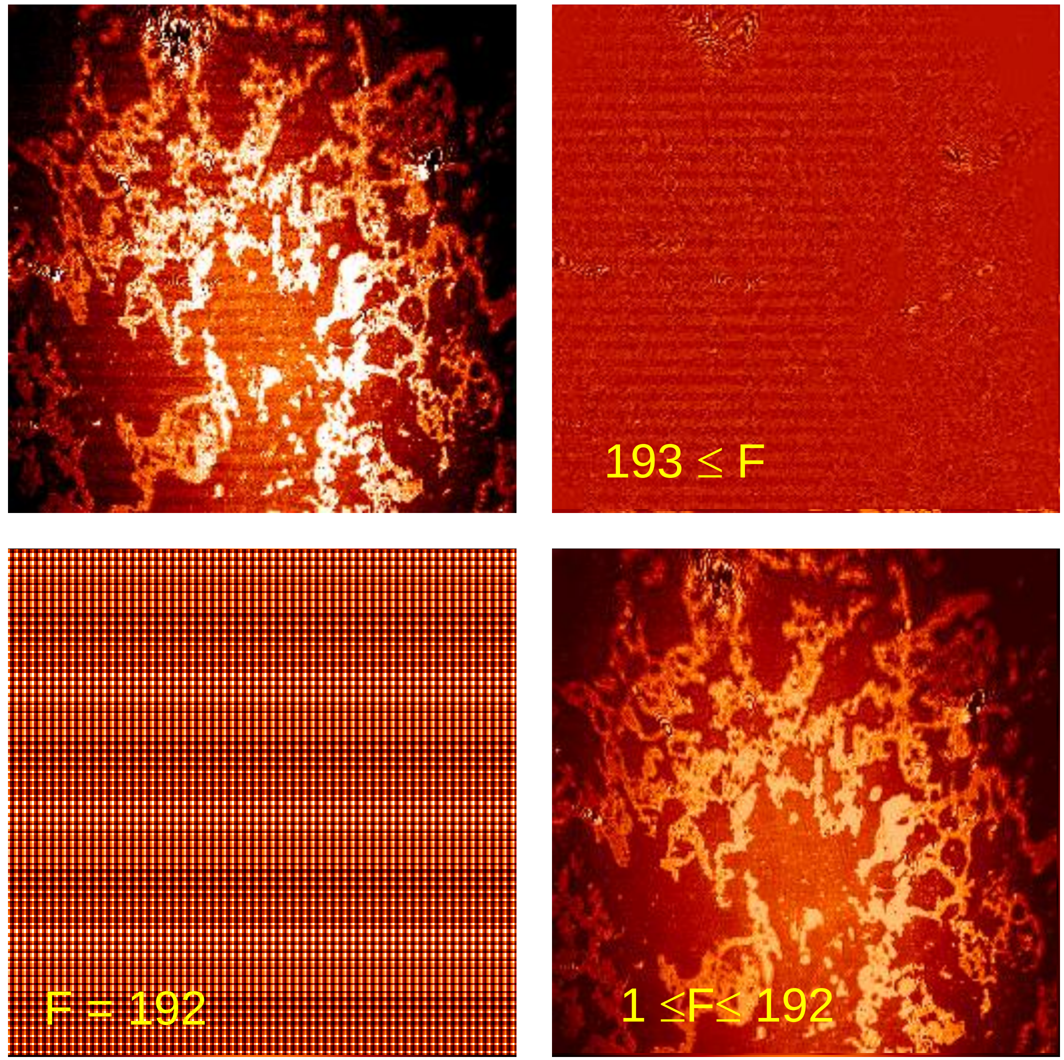
$$f(a,b) = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l)\, e^{\iota 2\pi\left(\frac{ka}{N} + \frac{lb}{N}\right)}$$

# Segmentation – basics

*Local filters:*
- Linear
- Non linear

*Global filters:*
-Fourier
-Wavelets
- ...

*Others:*
- Adaptive analysis
- ...



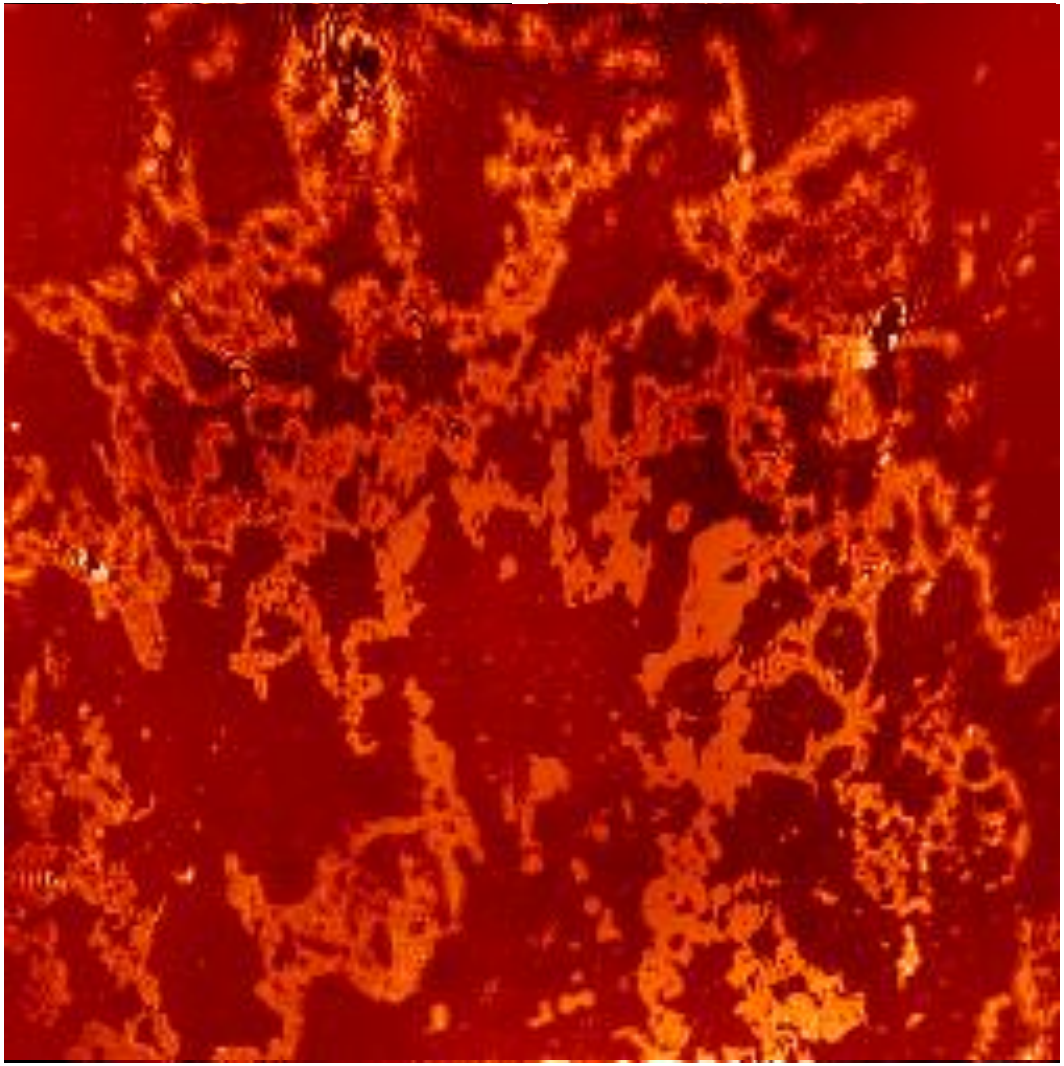
# Segmentation – basics



*Local filters:*
- Linear
- Non linear

*Global filters:*
-Fourier
-Wavelets
- ...

*Others:*
- Adaptive analysis
- ...



# Segmentation – basics



Thresholding
  example: Otsu

Convolution based
  Convolution operation
  Examples: gradient, Laplace, Sobel, Gaussian

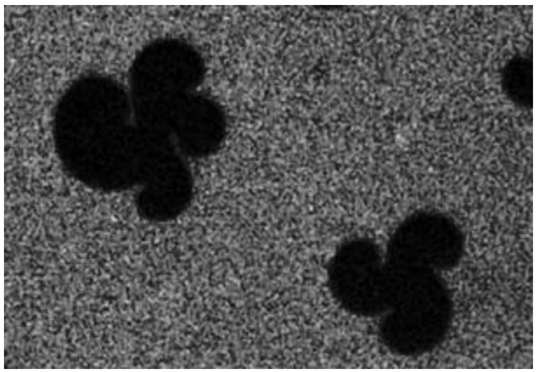
Morphological
  Morphological operators
  Size
  Thinning / skeletonization

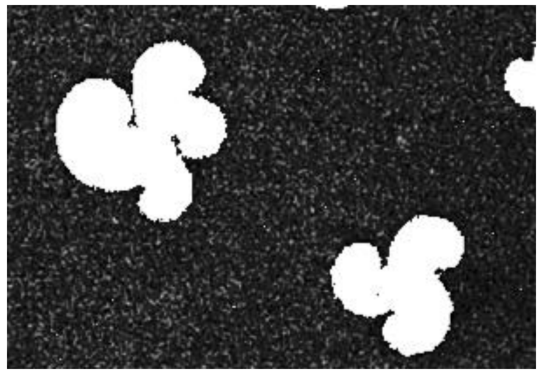
Arithmetic-logic
  AND, OR, XOR

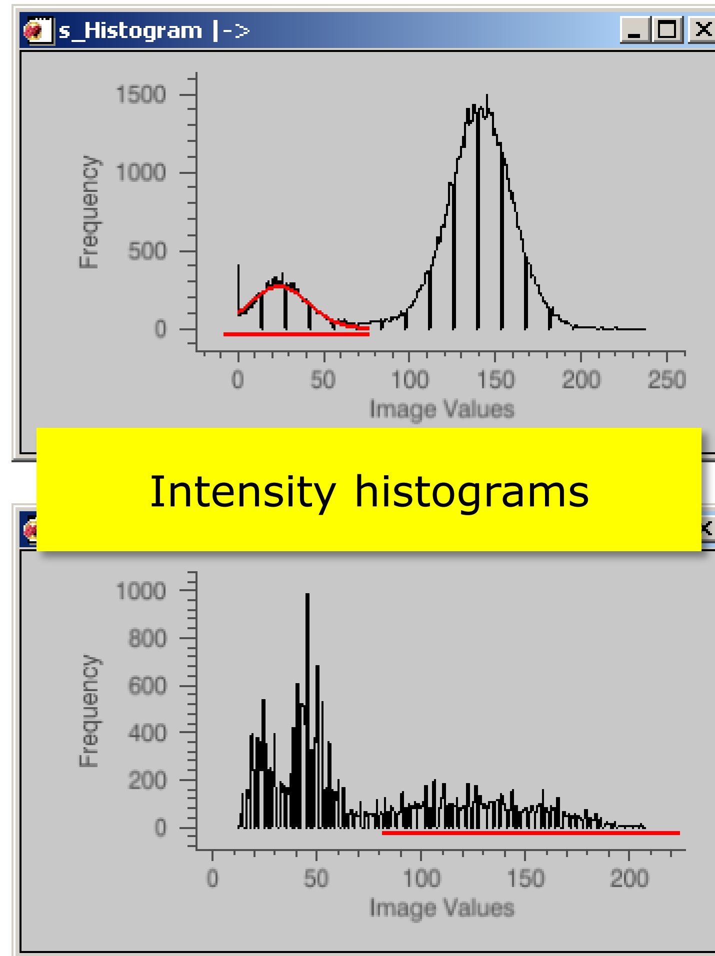
And a long etc.

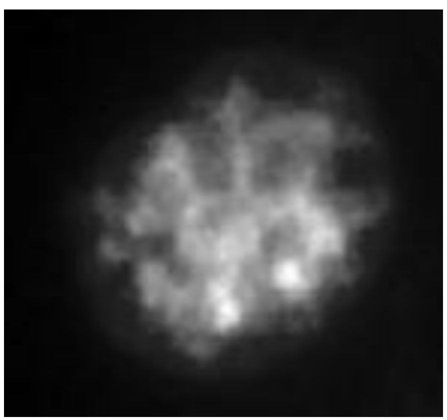
# Segmentation – basics

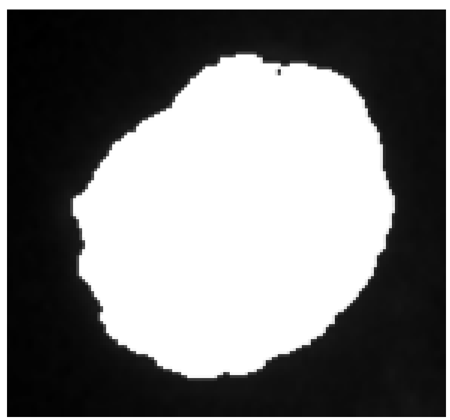
- Threshold fillter
  segmentation: ROIs (white) / background (black)



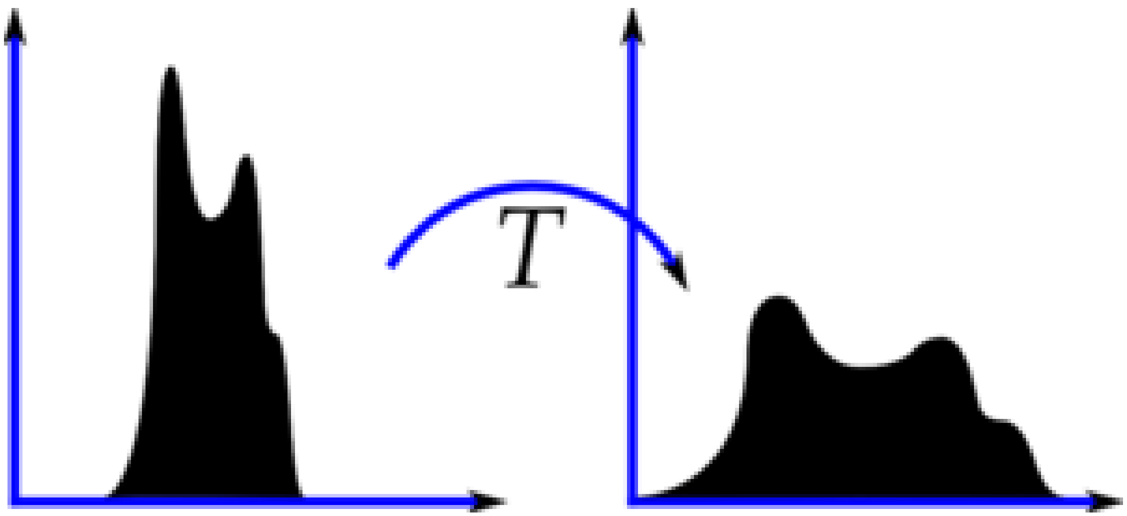




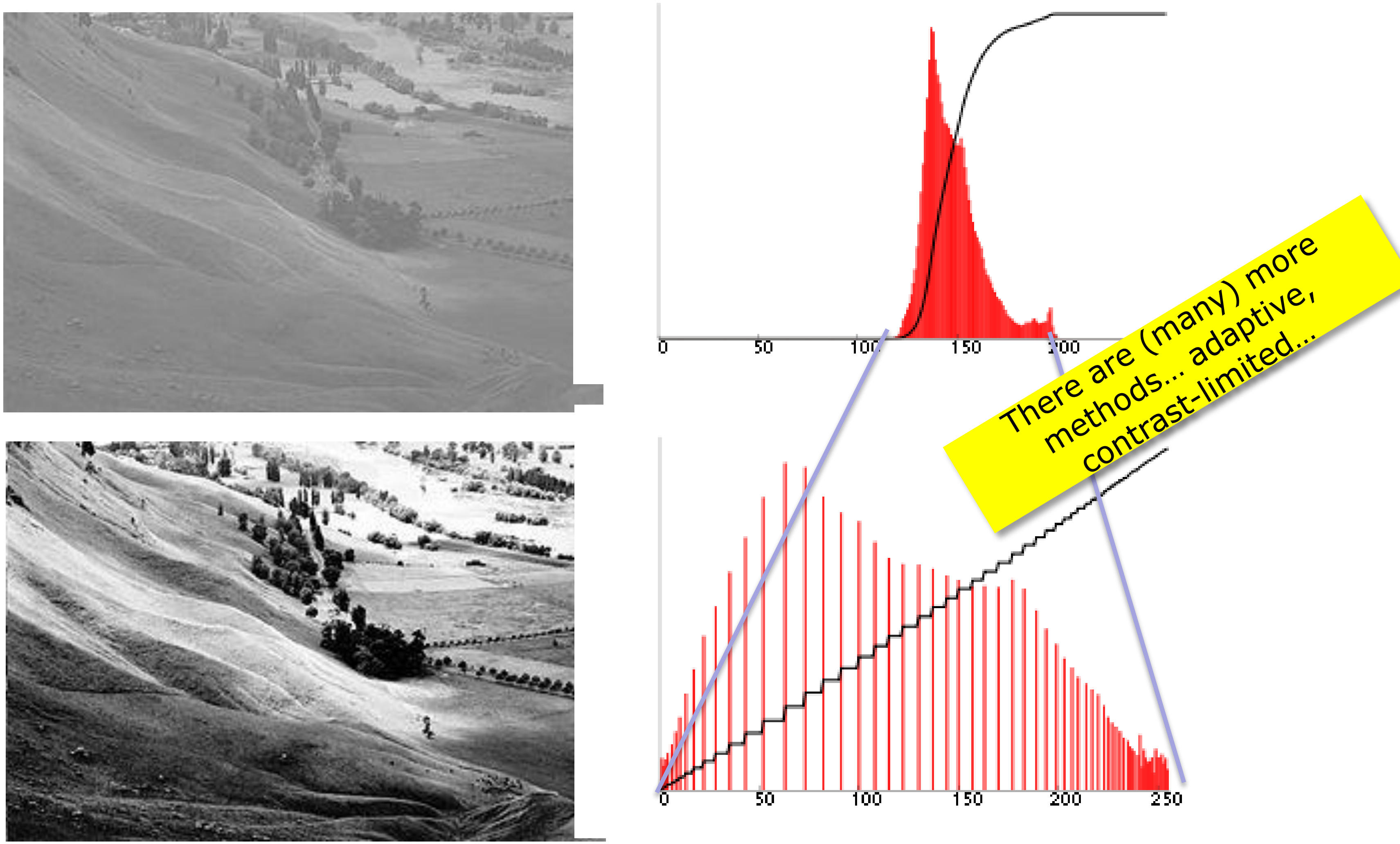


- Histogram equalization




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

- Histogram equalization




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

- Otsu threshold
  - Idea: to separate the image pixel in two classes (sets), minimizing the sum of variances from both classes

$$\min \sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

$$t: \text{threshold}, \quad \omega_i: \text{probability of class } i$$

$$\omega_1 = \sum_0^t p(i) \qquad \omega_2 = \sum_{t+1}^{top} p(i)$$

**Algorithm**

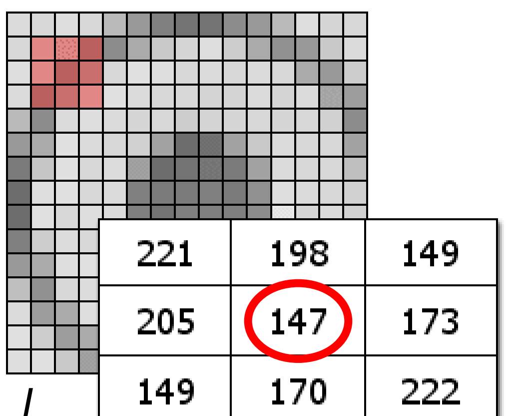
1. Compute histogram and probabilities of each intensity level
2. Set up initial $\omega_i(0)$ and $\mu_i(0)$
3. Step through all possible thresholds $t = 1 \ldots$ maximum intensity
   1. Update $\omega_i$ and $\mu_i$
   2. Compute $\sigma_b^2(t)$
4. Desired threshold corresponds to the maximum $\sigma_b^2(t)$
5. You can compute two maximums (and two corresponding thresholds). $\sigma_{b1}^2(t)$ is the greater max and $\sigma_{b2}^2(t)$ is the greater or equal maximum
6. Desired threshold = $\frac{threshold_1 + threshold_2}{2}$

N. Otsu (1979).

- Convolution
  - Lots of filters based on this principle
    http://en.wikipedia.org/wiki/Convolution
- Matrix convolution, in our case, is an operation between two matrices, namely…
  - the input image, *I*
  - a *kernel*, *K*

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

*K*




*I*

$$(K \otimes I)_{i,j} = (-1 \cdot 221) + (0 \cdot 198) + (1 \cdot 149) + (-2 \cdot 205) + (0 \cdot \mathbf{147}) + (2 \cdot 173) + (-1 \cdot 149) + (0 \cdot 170) + (1 \cdot 222) = -63$$

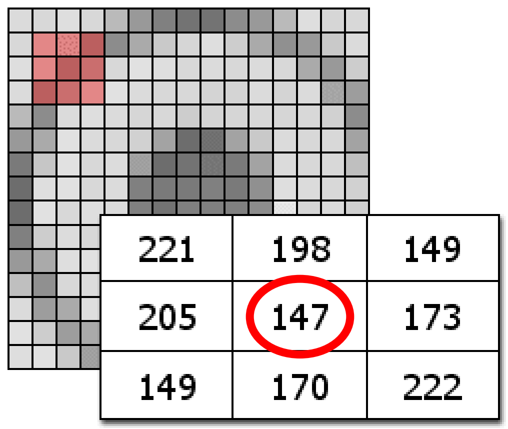
Adapted from James Matthews, 2002
http://www.generation5.org/content/2002/convolution.asp

# Segmentation – basics

*K*

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

*I*



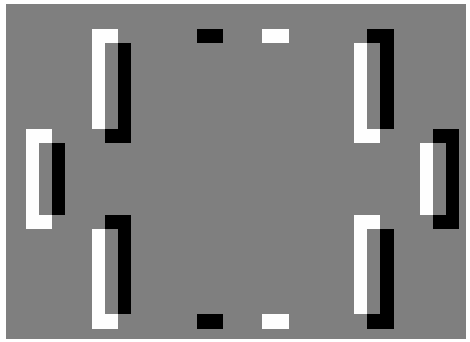

$(K \otimes I)_{i,j}$ = (-1 * 222)
+ ( 0 * 170)
+ ( 1 * 149)
+ (-2 * 173)
+ ( 0 * **147**)
+ ( 2 * 205)
+ (-1 * 149)
+ ( 0 * 198)
+ ( 1 * 221) = +63

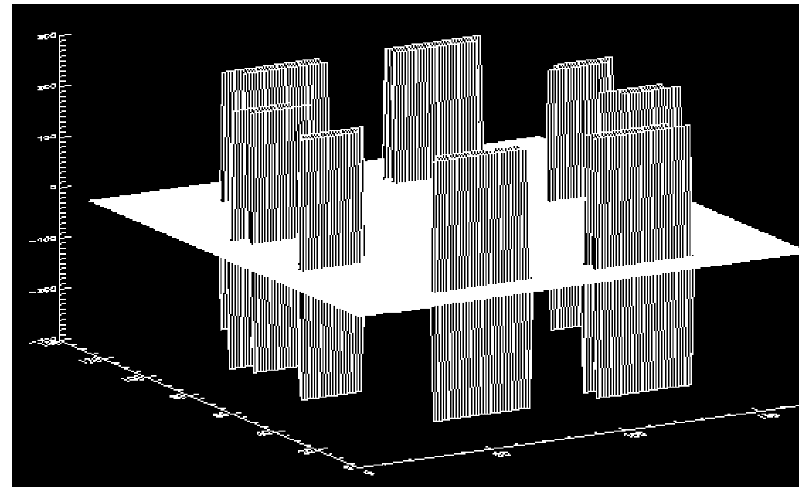
Matrix convolution can be implemented in different ways… beware of the algorithm!

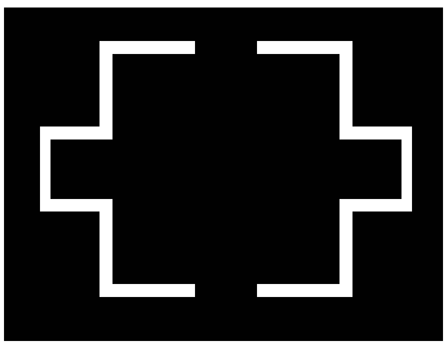
James Matthews, 2002
http://www.generation5.org/content/2002/convolution.asp

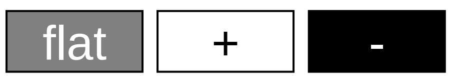
- Intensity gradients (discrete approximation)

$$\frac{\partial I}{\partial x} \approx$$

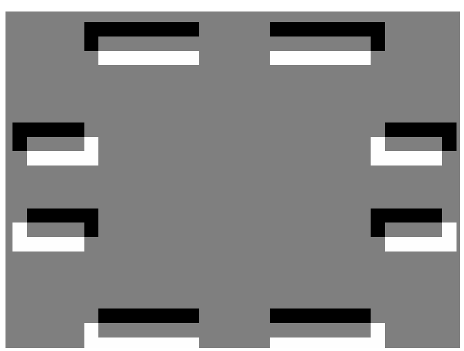


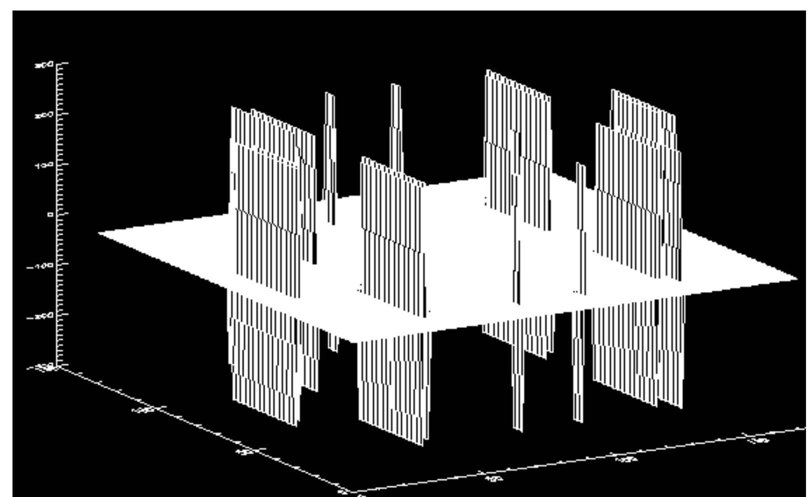


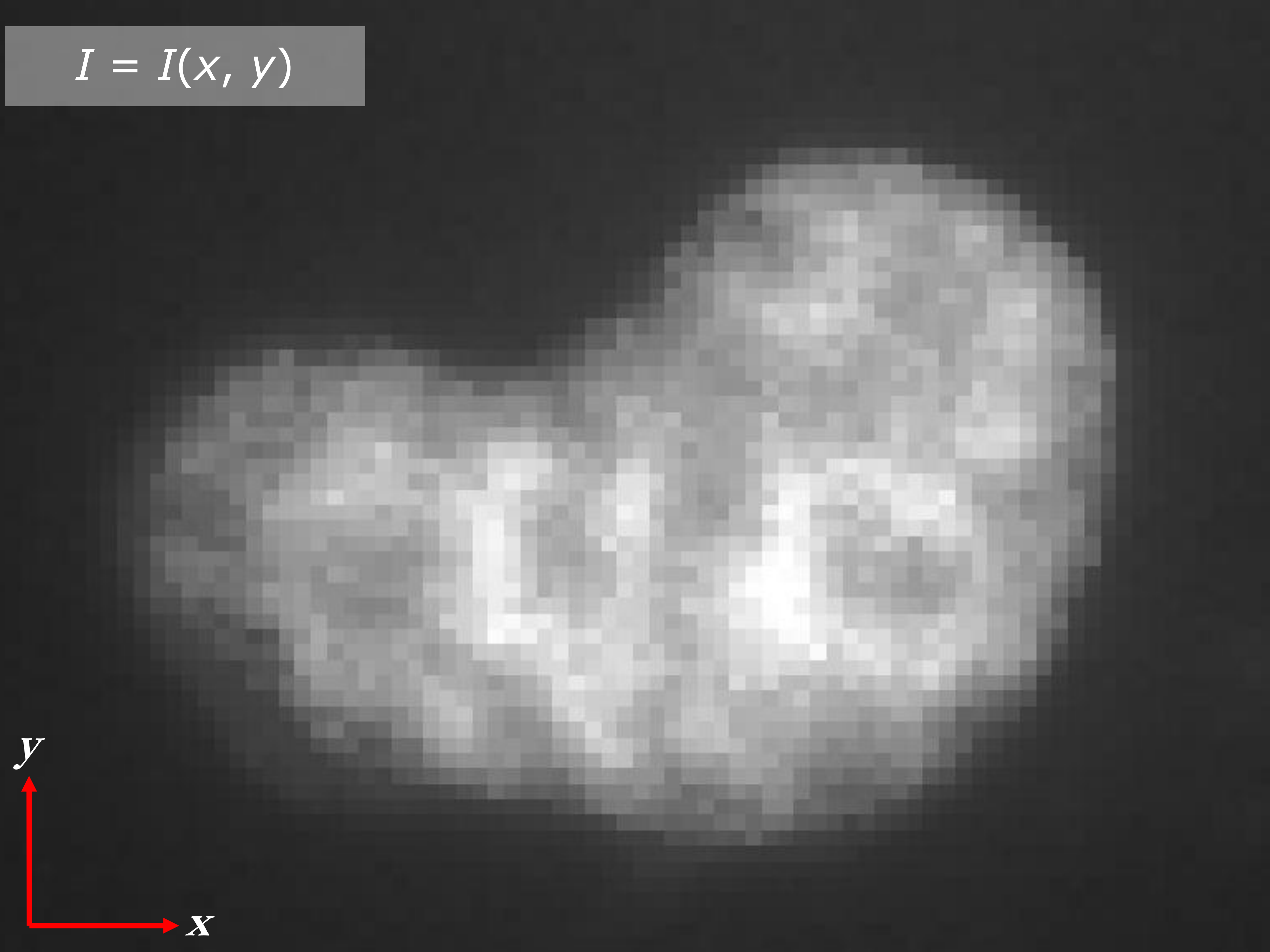


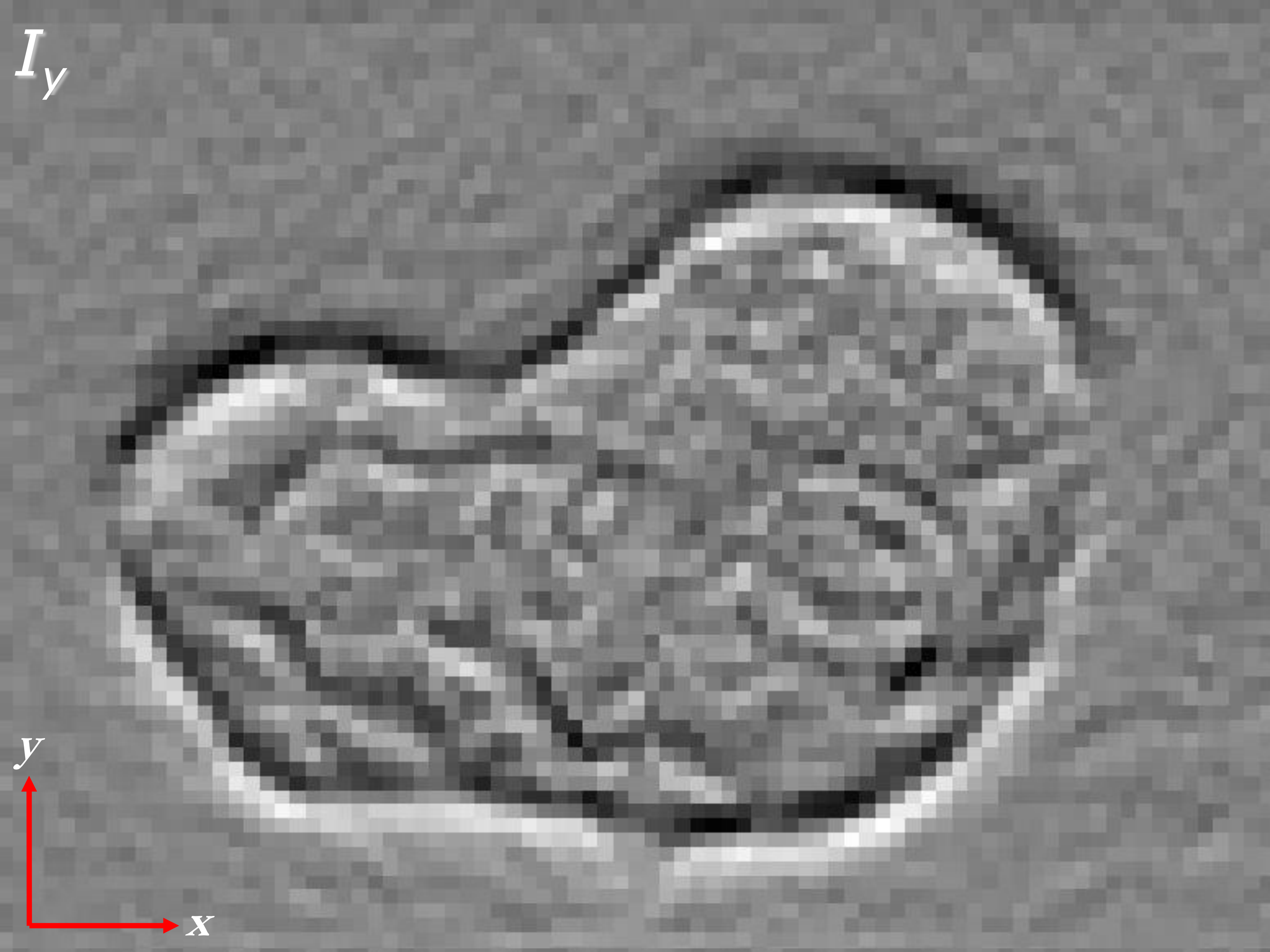


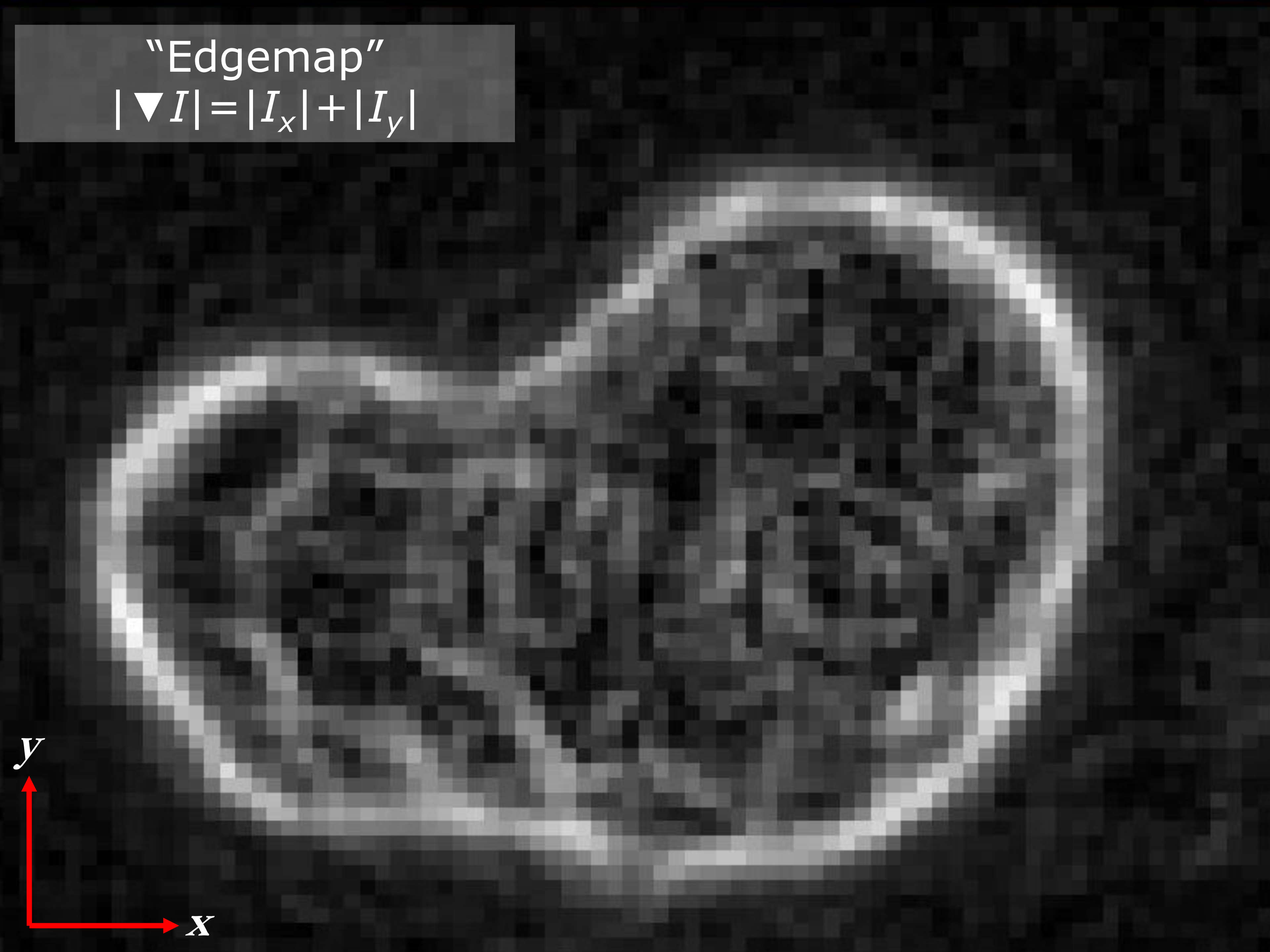

$$\frac{\partial I}{\partial y} \approx$$





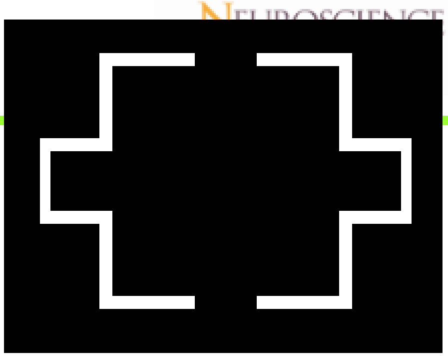

I = I(x, y)
y
x


$I_y$
y
x


"Edgemap"
$|\nabla I| = |I_x| + |I_y|$
y
x

# Segmentation – basics

- Intensity gradients (discrete approximation)

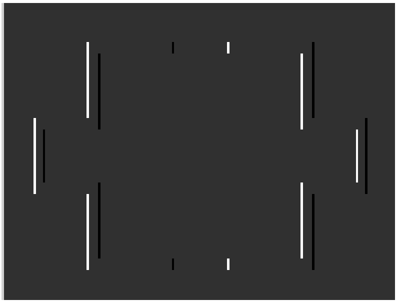


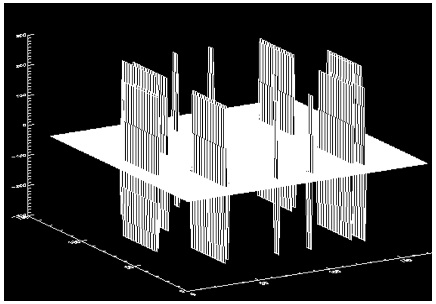
$$I = I(x, y)$$

$$\frac{\partial I}{\partial x} \approx \frac{I(x+\Delta x, y) - I(x, y)}{\Delta x} = Kx \otimes I$$

$$\Delta x = 1 \text{ pixel}$$

$$Kx = \begin{Bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{Bmatrix}$$



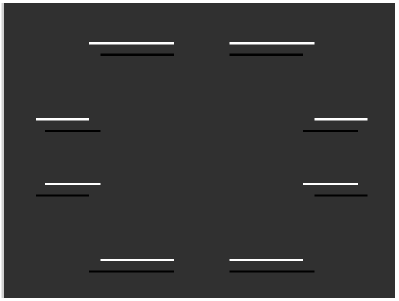


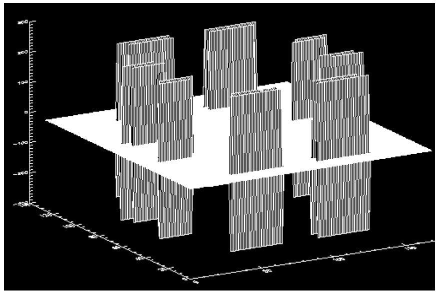
flat + -

$$\frac{\partial I}{\partial y} \approx \frac{I(x, y+\Delta y) - I(x, y)}{\Delta y} = Ky \otimes I$$

$$\Delta y = 1 \text{ pixel}$$

$$Ky = \begin{Bmatrix} 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{Bmatrix}$$

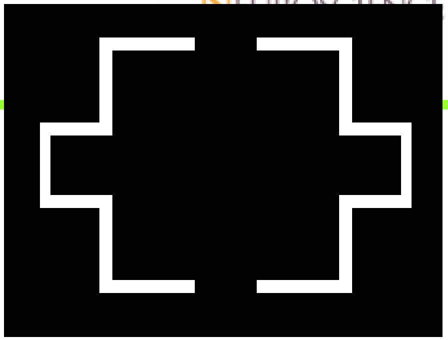




# Segmentation – basics



- Kernels...



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

Laplace

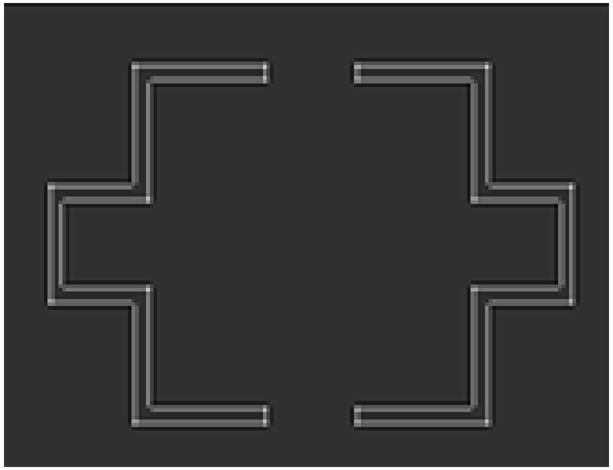
$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

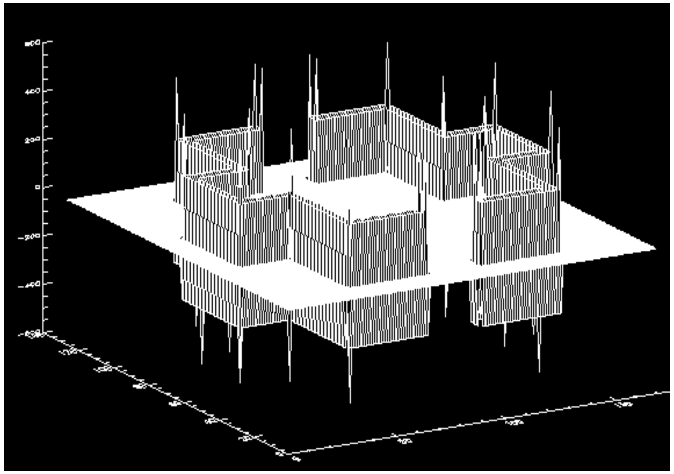
$$\nabla^2 I \approx \frac{f(x+\Delta x, y) - 2f(x,y) + f(x-\Delta x, y)}{(\Delta x)^2} + \frac{f(x, y+\Delta y) - 2f(x,y) + f(x, y-\Delta y)}{(\Delta y)^2}$$

$$\nabla^2 I \approx \frac{f(x+\Delta x, y) + f(x, y+\Delta y) - 4f(x,y) + f(x-\Delta x, y) + f(x, y-\Delta y)}{(\Delta x)^2} = K_L \otimes I$$

$$\Delta x = \Delta y = 1\,\text{pixel}$$

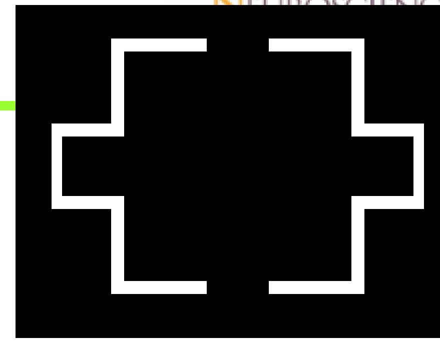
$$K_L = \begin{Bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{Bmatrix}$$





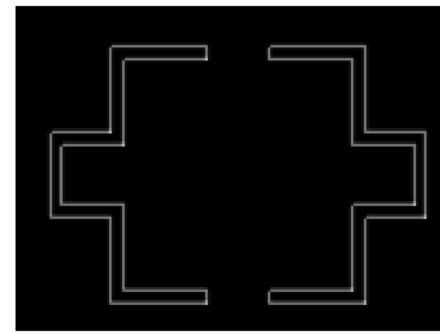
# Segmentation – basics

- Edgemaps



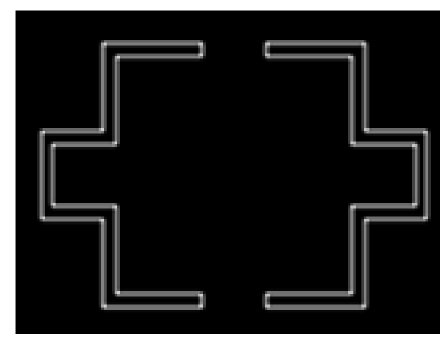
$$I = I(x, y)$$

An ***edgemap*** filter takes intensity changes as ROI boundaries or "edges"



$$f = \sqrt{(Kx \otimes I)^2 + (Ky \otimes I)^2}$$

**Example:Sobel filter**
(notice the thick ROI edges)



$$Sx = \begin{Bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{Bmatrix} \qquad Sy = \begin{Bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{Bmatrix}$$

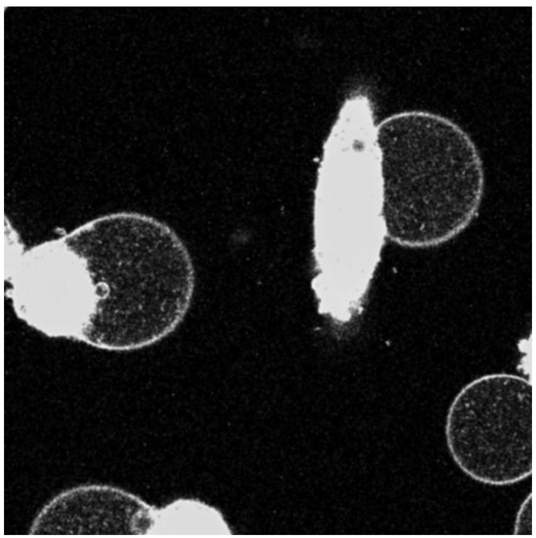
$Sx \otimes I$ ?

$Sy \otimes I$ ?

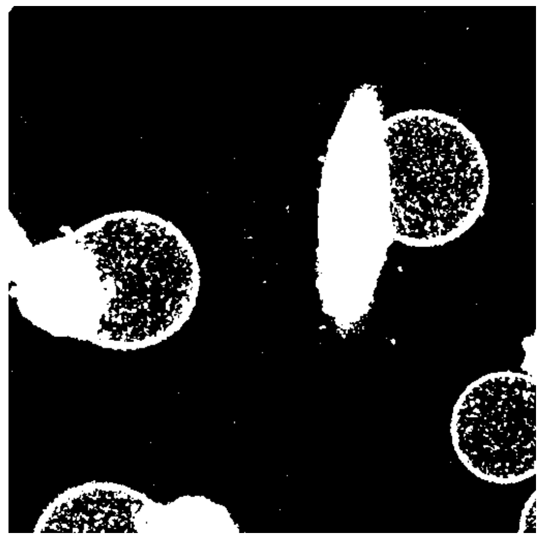
$$f_{Sobel} = \sqrt{(Sx \otimes I)^2 + (Sy \otimes I)^2}$$

# Segmentation – basics

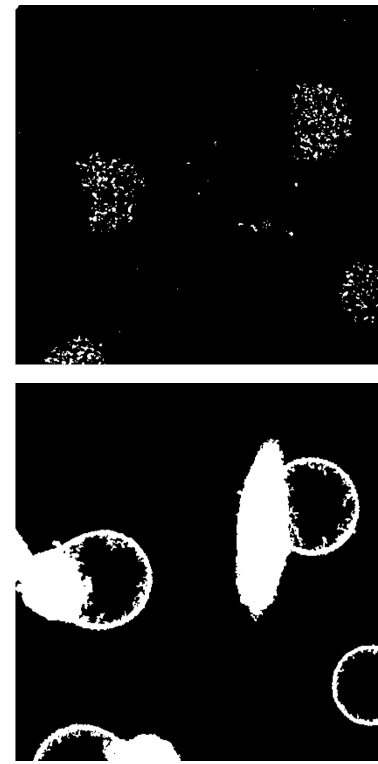
- Morphology based filters
  - Example: size selection



Input greyscale image



After thresholding…



Size selection

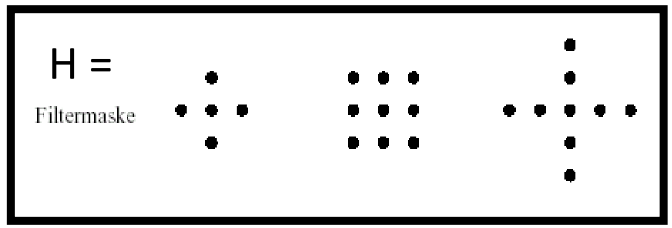
How to define a size-select algorithm?

# Segmentation – basics

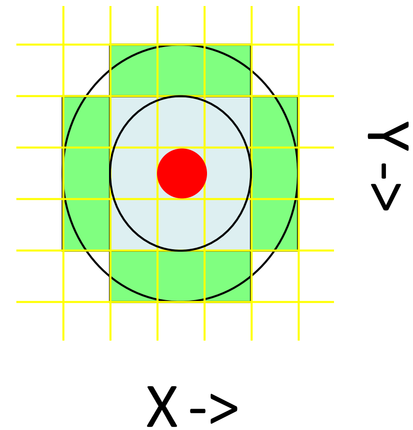
Filtering with morphological operators:

- Structuring element, template or mask H ...

- Additional rules...




H $(x_i,y_j,r)$:




Polynomial filters $$y(m,n) = \bar{h}_1[x(m,n)] + \bar{h}_2[x(m,n)],$$

$$\bar{h}_1[x(m,n)] = \sum_{\substack{p=0 \\ (p,q)\neq(0,0)}}^{P-1} \sum_{q=0}^{Q-1} a(p,q) \cdot x(m-p,n-q)$$

$$\bar{h}_2[x(m,n)] = \sum_{\substack{p=0 \\ (p,q)\neq(0,0)}}^{P-1} \sum_{q=0}^{Q-1} \sum_{\substack{k=0 \\ (k,l)\neq(0,0)}}^{P-1} \sum_{l=0}^{Q-1} b(p,q,k,l) \cdot x(m-p,n-q) \cdot x(m-k,n-l)$$

# Segmentation – basics

- Mathematical morphology

## Minkowski Operations

Addition... dilation: $A \oplus S = \{(m,n) \mid [S+(m,n)] \cap A \neq 0\}.$

Substraction ... erosion: $A \ominus S = \{(m,n) \mid [S+(m,n)] \subseteq A \neq 0\}$

*...opening* : $A \circ S = (A \ominus S) \oplus S,$

*...closing* : $A \bullet S = (A \oplus S) \ominus S,$

- A: image
- S: structuring element

- Mathematical morphology

$$A \oplus S = \{(m,n) \,|\, [S + (m,n)] \cap A \neq 0\}.$$

$$A \ominus S = \{(m,n) \,|\, [S + (m,n)] \subseteq A \neq 0\}$$

$$A \circ S = (A \ominus S) \oplus S,$$

$$A \bullet S = (A \oplus S) \ominus S,$$

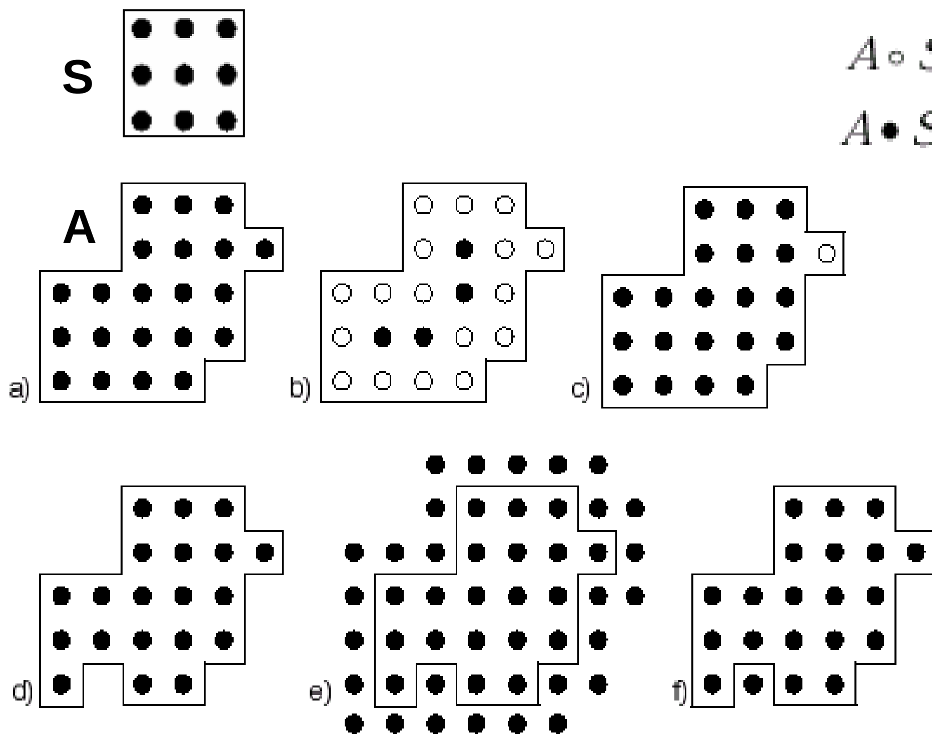
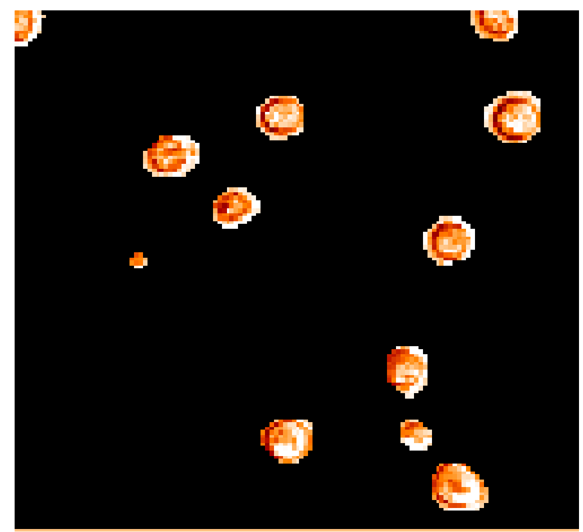
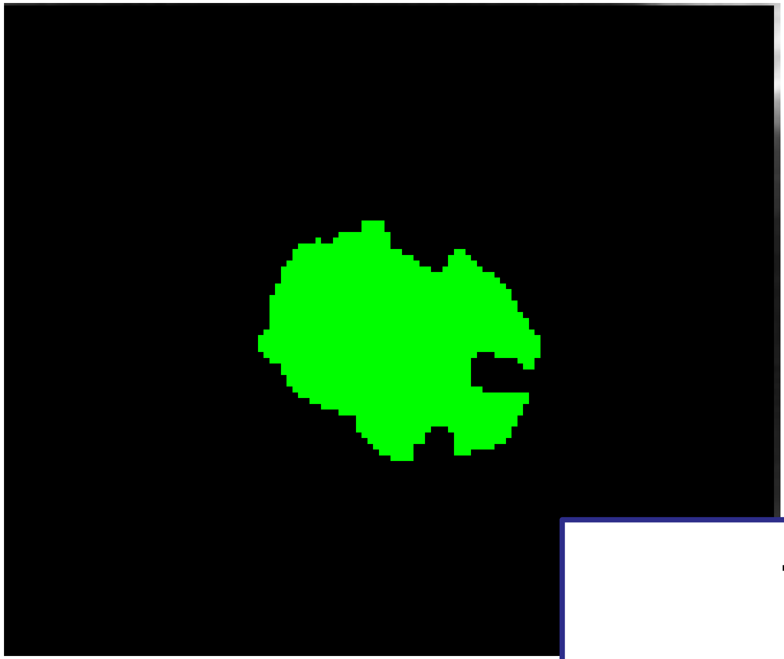


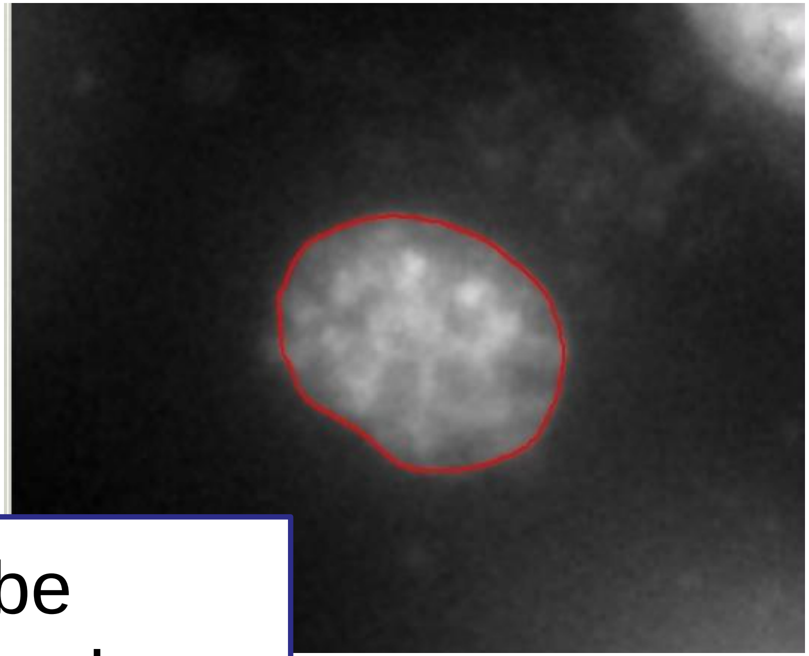

**Abb. 2.5.** **a** Originalform **b** erodiert **c** opening (Dilatation von b)
**d** Originalform **e** dilatiert **f** closing (Erosion von e)



# Segmentation – basics

- “Some” times more information is needed in order to achieve a good segmentation





To be continued…

# References


- David Marr. Vision
  MIT press, 1982

- John Russ. The image processing handbook, $4^{th}$ ed.
  CRC Press, 2002

- Nixon, Aguado. Feature extraction & image processing, $1^{st}$|$2^{nd}$|$3^{rd}$ ed.
  Academic Press, 2002|2008|2012

- Aubert & Kornprobst. Mathematical Problems In Image Processing
  Springer, 2006

- From SCIAN-Lab (see publications section on the website)
  - Jara, 2006. 2D/3D active contours
  - Olmos, 2009. 2D topology adaptive active contours (t-snakes)

# References

Some free and open source software tools

- Java based (Java runtime requried)
  - 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 (in C language) + examples (www.ipol.im; BSD / GPL / LGPL licenses or similar)