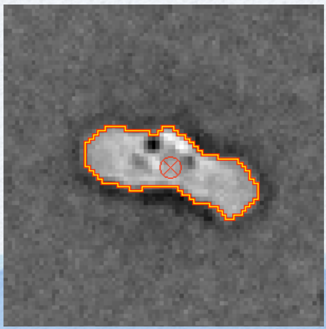
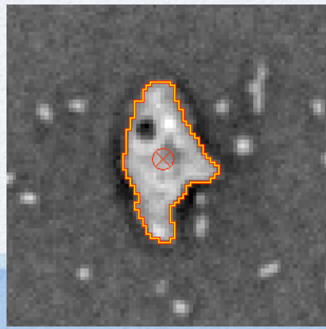


CIMST 2013

September 2



Digital Image Processing and Analysis

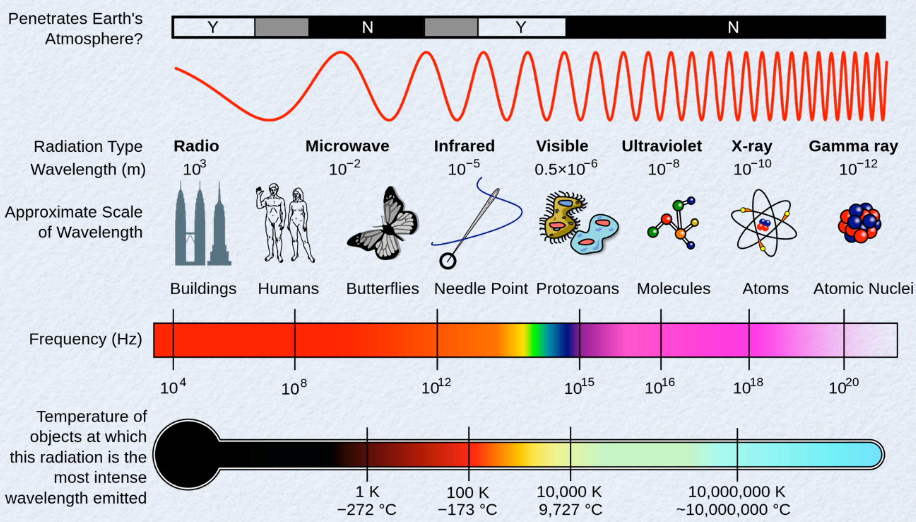
Simon F. Nørrelykke

ETH Zurich

Light Microscopy and Screening Center (LMSC)

Data Analysis Unit (DAU)

Electro-Magnetic Spectrum



We are involved as long as there is a digital image

http://en.wikipedia.org/wiki/File:EM_Spectrum_Properties_edit.svg

OUTLINE

- Introduction
  - History and motivation
  - What is a digital image?
    - Color space
    - What is digital image processing?
- Enhancement and Restoration (processing)
  - Intensity operations
    - power-law and piecewise linear transformations
    - histogram equalization and specification
  - Spatial operations
    - filtering (smoothing and sharpening)
    - affine
    - binary operations (erosion, dilation)
    - Feature extraction
  - Frequency Domain operations
    - filtering
    - deconvolution
- Segmentation (processing)
  - Thresholding
  - Edge based
  - Morphological snakes
- Putting it all together
  - tracking of amoebae (incl. instrument control)

INTRODUCTION

History

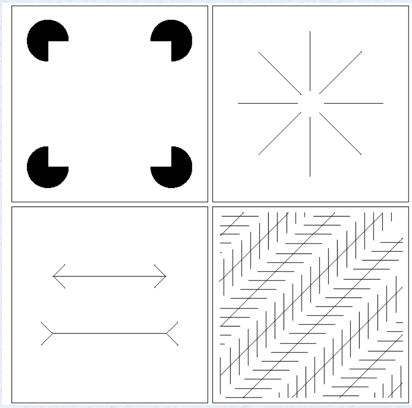
- CCD (2009 nobel prize)
    - invented 1969 in AT&T Bell Labs
    - first commercial devices available in 1974 (100 x 100 pixels = 0.01 MegaPixels)
- GFP (2008 nobel prize)
    - 1992: the cloning and nucleotide sequence of wtGFP reported in *Gene*.
    - since then molecular biology hasn't been the same
- ImageJ (formerly NIH image) is from 1997

Motivation

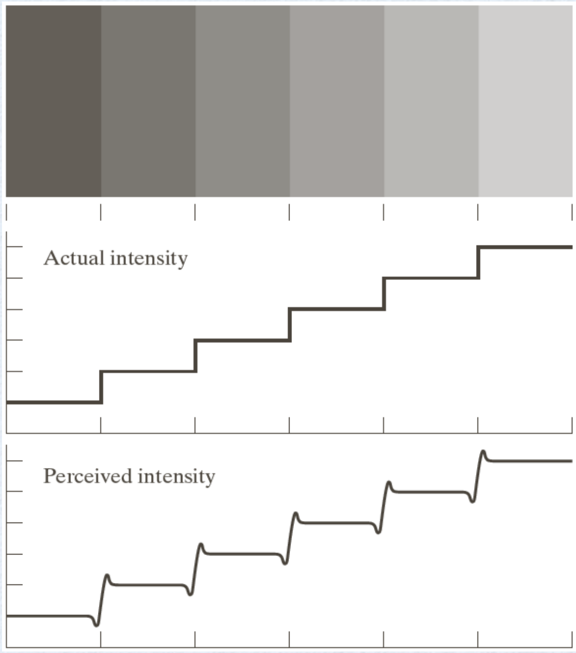
- Make pretty pictures
    - publications, talks, websites, ...
- Automate repetitive tasks
    - contrast adjustment, cropping, counting, ...
- Get numbers out of pictures
    - cell sizes, vessel lengths, GPF expression level, ...
- Make experiment possible
    - whole-genome screen (3 channels): two million images
- Objectivity
    - or at least:
- Reproducibility
    - this is receiving a lot of attention at the moment

Objectivity

All inner squares have the same intensity. They appear darker as the background becomes lighter



Some well-known optical illusions



The Mach band effect: the perceived intensity is not a simple function of actual intensity

Digital Image

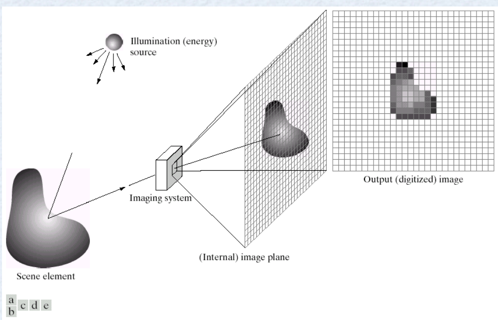
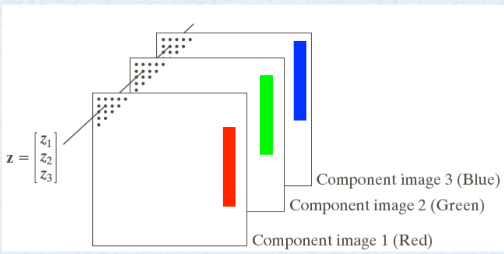


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.



Operations are done on each color channel individually, as if it was a gray-scale image

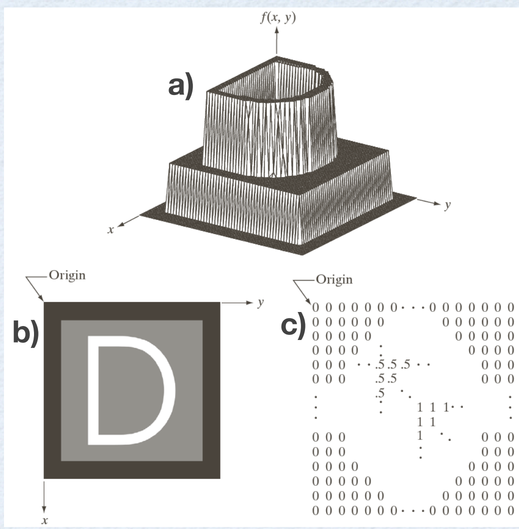
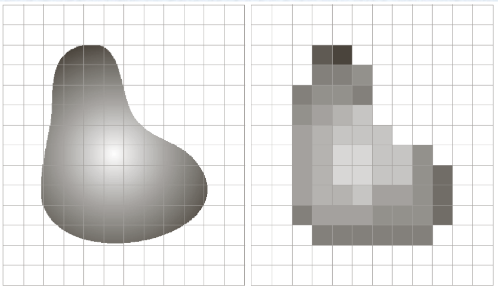
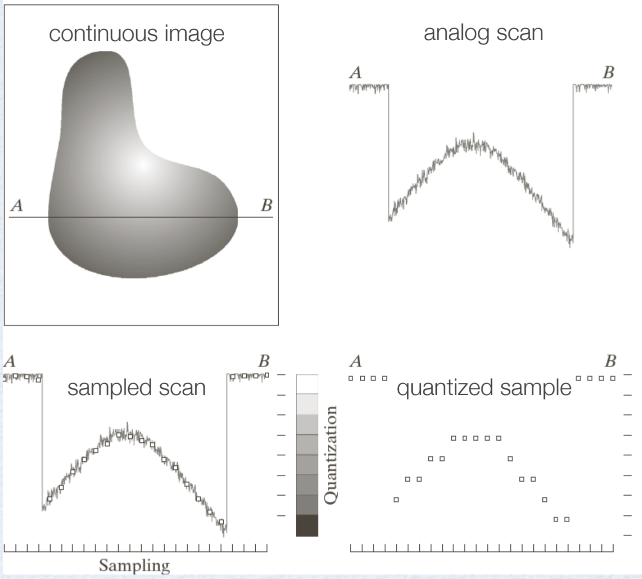


Image shown as: a) surface plot, b) visual intensity array, c) 2D numerical

QUANTIZATION



Continuous image projected onto sensor array.
Result of **spatial sampling** and **intensity quantization** is visible.



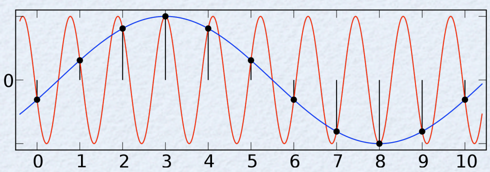
One-dimensional example of the effect of spatial sampling and intensity quantization.
Spatial sampling is influenced by the image-magnification and pixel-density.
The **intensity quantization** is determined by the camera (8bit, 12bit, 16bit, ...), but is influenced by user-adjustments of entire imaging system.

Adapted from Figs. 2.16 and 2.17 in "Digital Image Processing, 3rd Edition"

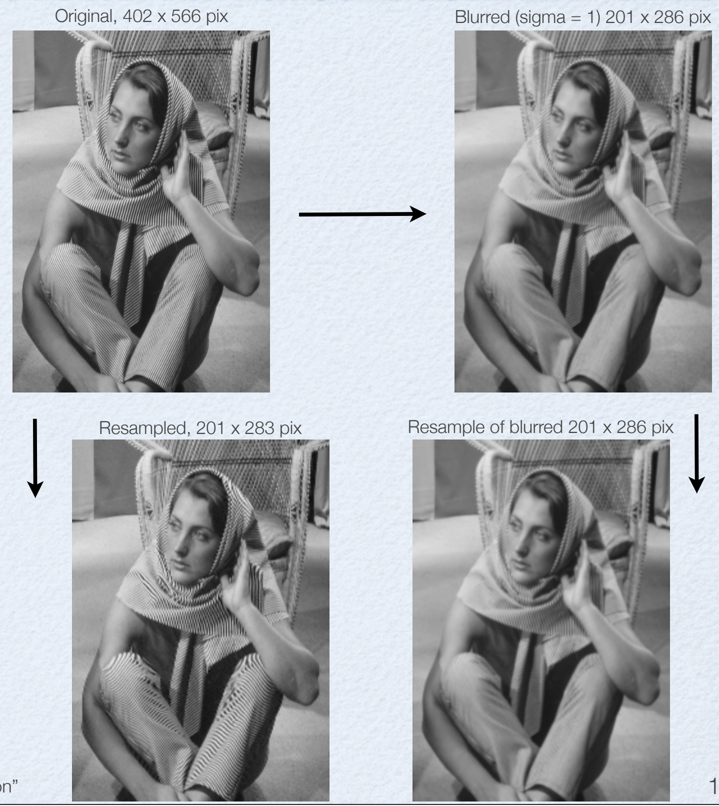
ALIASING (SPATIAL)

- **Sample densely enough** that smallest feature of interest can be resolved.
- Be careful when resampling/ scaling an image

- **Aliasing** refers to an effect that causes different signals to become indistinguishable (or *aliases* of one another) when sampled.
- It also refers to the distortion or artifact that results when the signal reconstructed from samples is different from the original continuous signal.
- http://en.wikipedia.org/wiki/Aliasing

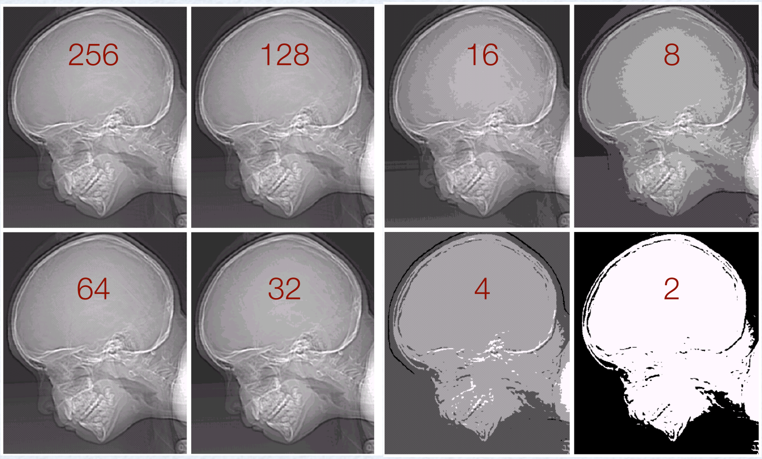


A graph showing aliasing of an *f=0.9* sine wave by an *f=0.1* sine wave by sampling at a period of *T=1.0*



Adapted from Fig. 4.17 in "Digital Image Processing, 3rd Edition"

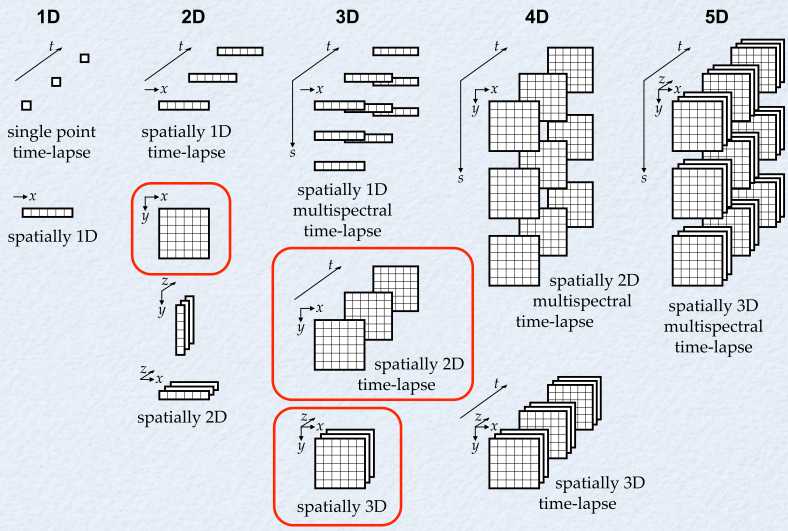
Quantization (Intensity)



Number of gray-scale values and the effect.
256 = 2^8 values: an 8-bit image.
False contours visible from ~32.
Two levels correspond to a thresholded image

Adapted from Fig. 2.21 in “Digital Image Processing, 3rd Edition”

Image Dimensions



- We will focus on the spatial 2D case
  - it is the most common
  - 3D is usually a *z*-stack of 2D
  - movies are 2D time-lapse

Meijering and van Cappellen. Biological Image Analysis Primer. (2006)

COLOR SPACE

- Most color cameras operate with RGB
  - RGB is inspired by the human eye
  - RGB: cartesian coordinates
  - HSV (hue, saturation, lightness): cylindrical coordinates
- Other color spaces exist
  - CMYG: cyan, magenta, yellow, green; for printers
  - L*a*b
  - sRGB
  - CIEXYZ
  - ...
- Color images can be decomposed into a number of gray-scale images
- **We will proceed in the gray-scale**

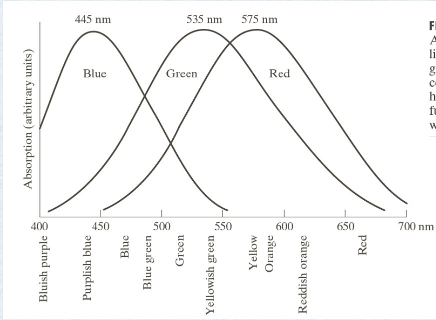
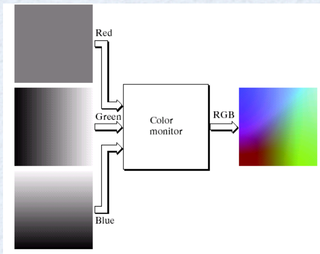


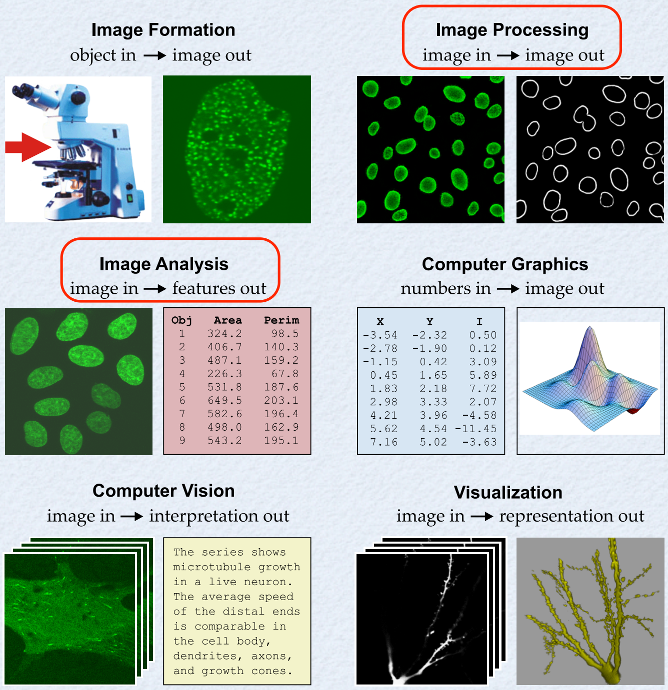
FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.



IMAGES ARE JUST NUMBERS

- So **image analysis is just math**
  - linear algebra, statistics, topology, calculus, partial differential equations, functional analysis, ...
  - and art, or at least a sense of beauty
- We can add, subtract, multiply, and divide images with each other
- We can perform **logical** operations on images
  - e.g. find all pixel values less than 17 and change them to zero; set the rest to 255 (thresholding)
- We can perform local and global **mathematical** operations on an image
  - e.g. replace each pixel by the average of itself and its neighbors (smoothing, linear operation)
  - or, replace each pixel by the median of itself and its neighbors (de-noising, non-linear operator)
  - or, replace each pixel by the value of the local spatial derivative (edge detection)
- We can **transform** images to another space, operate there, then transform back
  - e.g. Fourier Transform two images to *k*-space, multiply, inverse transform (gives the correlation/convolution). Can speed up processing orders of magnitude. See the convolution theorem http://en.wikipedia.org/wiki/Convolution_theorem.
  - Think of this as flying: You enter the airplane (transform), translate at ~1000 km/h (operate), then exit the airplane at/near destination (inverse transform). Could have been done on foot or by car, but if distance is great, then flying is faster.

Processing vs. Analysis

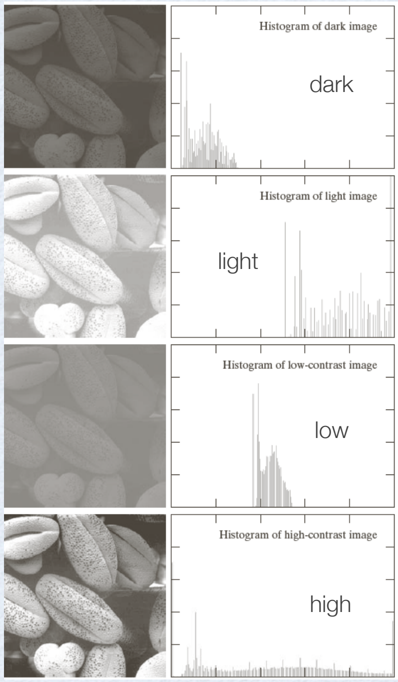


Meijering and van Cappellen. Biological Image Analysis Primer. (2006)

Enhance and Restore

Intensity operations

- **Enhance contrast** by
  - **Proper acquisition** (junk in => junk out)
    - Use the entire dynamic range of the camera
    - Inspect histogram of test-images
    - Adjust microscope (filters, iris, illumination strength, shutter time, ...)
    - Adjust camera (exposure time)
  - specification of look-up table (**LUT**)
    - after acquisition is done
    - typically a piecewise linear LUT is specified
  - when choice of LUT is based on histogram of pixel values:
    - **histogram equalization**, if flat histogram output
    - **histogram specification**, any shape histogram



Adapted from "Digital Image Processing, 3rd Edition"

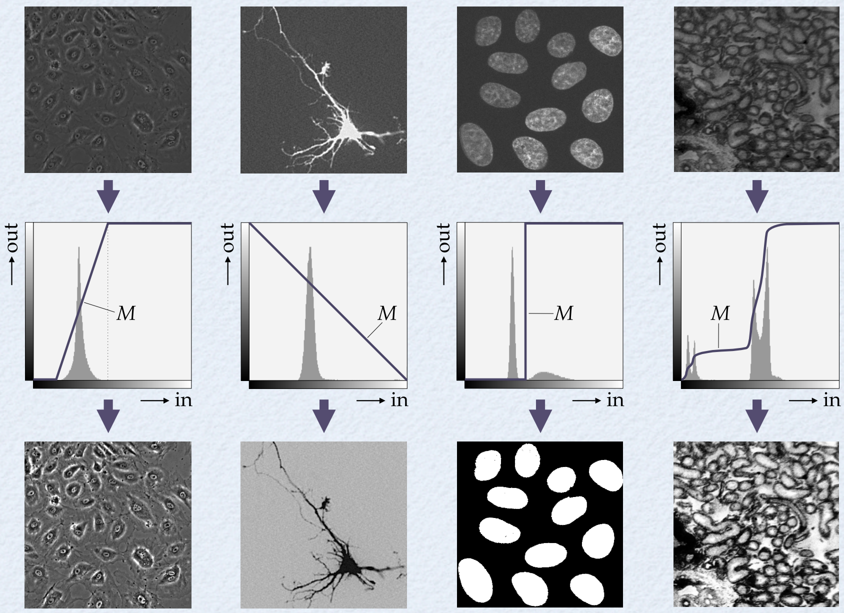
Intensity transformation

	Contrast Stretching	Intensity Inversion	Intensity Thresholding	Histogram Equalization

Originals

LUTs

Results



Meijering and van Cappellen. Biological Image Analysis Primer. (2006)

Histogram Equalization

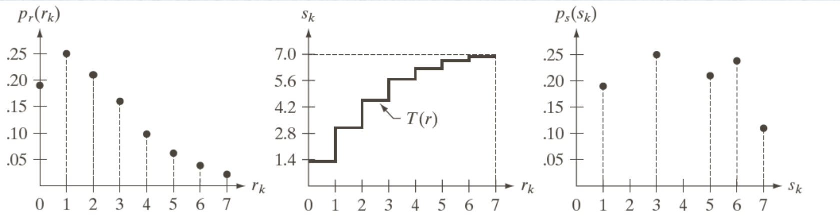
From probability theory we know a change of variable formula, used for finding one probability-density-function from another:

$$p_s(s) = p_r(r)\ |dr/ds|$$

r: input image, $p_r(r) = n_r/N$
s: output image, $p_s(s) = n_s/N$

This is what we use in histogram equalization (specification is similar):

$$s_k = (L-1)/N \sum_{j=0}^{k} n_j$$



a b c

FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

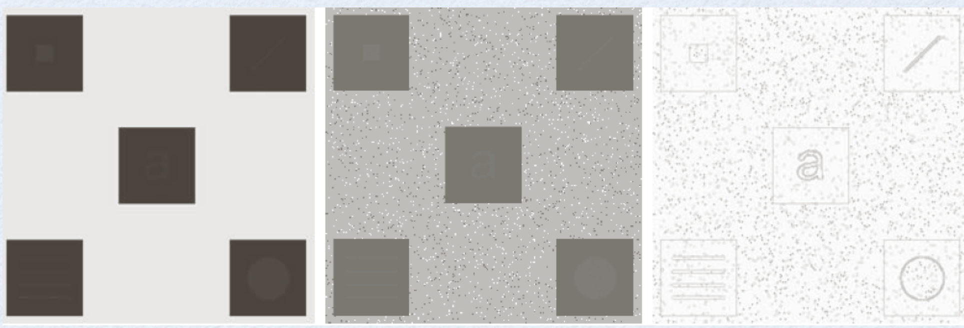
Adapted from Fig. 3.19 in "Digital Image Processing, 3rd Edition"

Local Intensity Operations

Original

Global histogram equalization

Local histogram equalization in 3 x 3 neighborhood



Left: features almost completely hidden inside squares
Middle: *global* histogram equalization reveals noise and hint of shapes.
Right: *local* histogram equalization reveals shapes inside squares

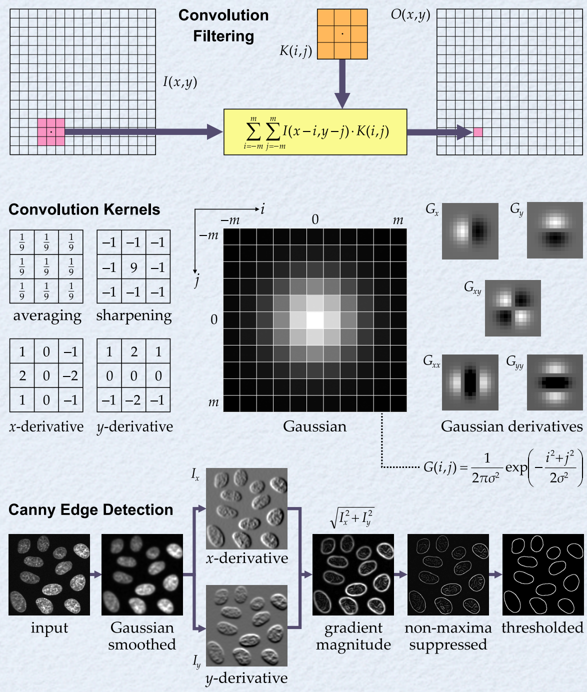
Adapted from Fig. 3.26 in "Digital Image Processing, 3rd Edition"

CONVOLUTION

Convolution is achieved by replacing each pixel by a **weighted average** of the local neighborhood.

Depending on the weights, the resulting image will be **smoothed**, **sharpened**, **show edges**, ...

In real life the convolution is done in the Fourier domain to speed up the process (often dramatically)



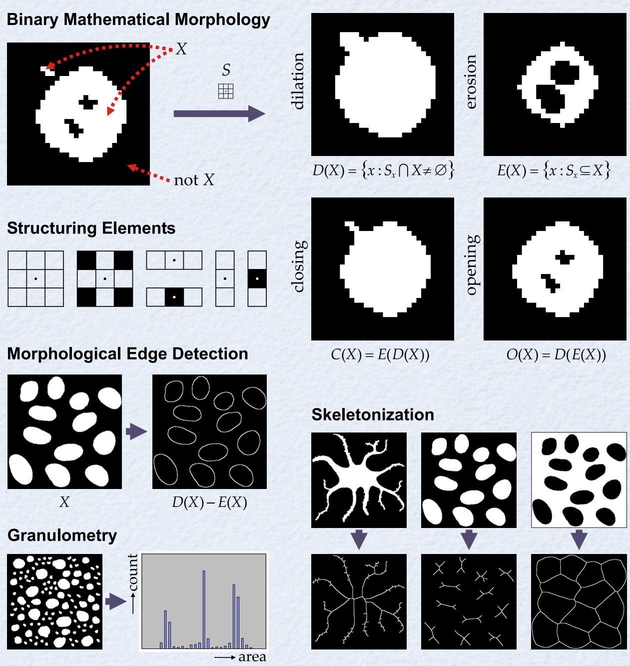
BINARY IMAGE OPERATIONS

After thresholding, operate on the binary image, ***X,*** by **dilation**, **erosion**, etc.

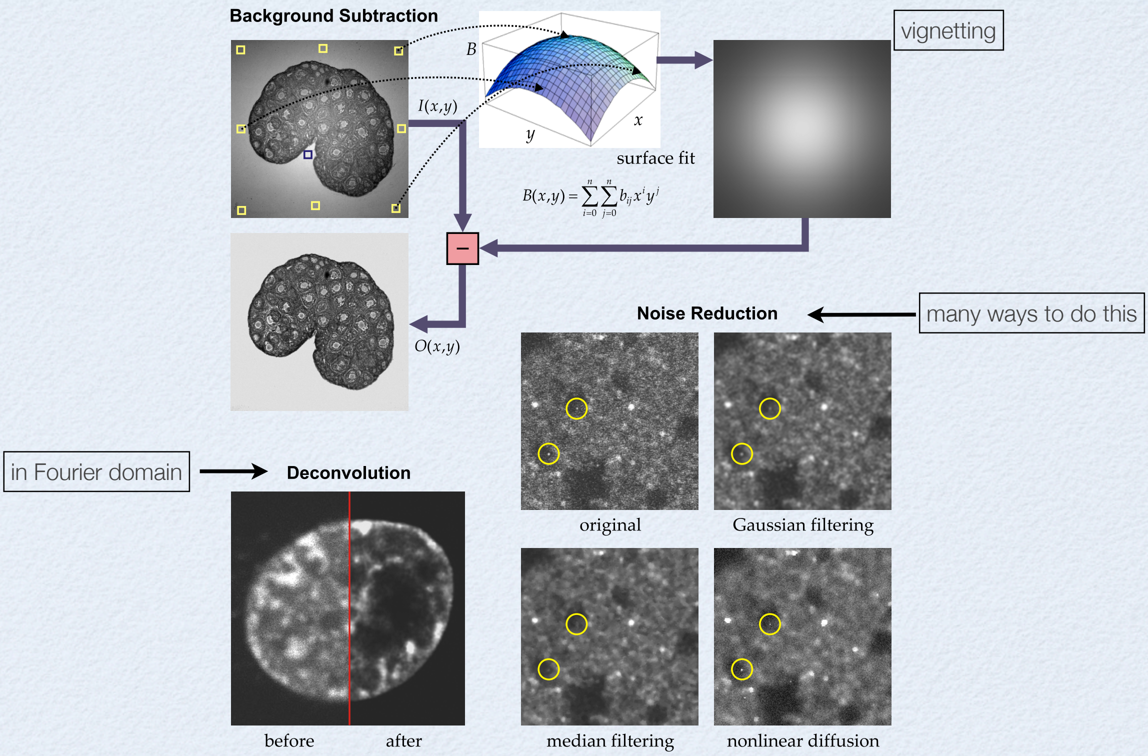
Done by centering a **structuring element, *S***, on each pixel and performing a logical query.

Opening = erosion, then dilation
Closing = dilation, then erosion

Many of these operations can also be done on gray-scale images.

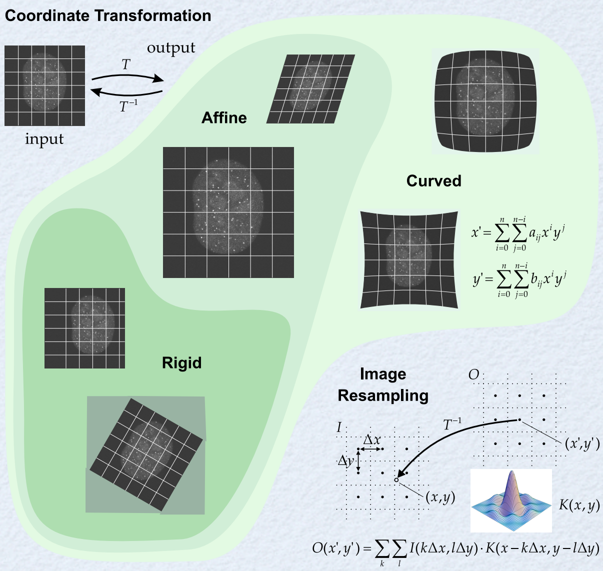


Restoration



Geometrical Transformation

- Coordinate **transformations**, *T*, are
  - rigid (translation and rotation)
  - affine (rigid + scaling and skewing)
  - curved (affine + nonlinear deformation)
- obtained by polynomial functions
- Image **resampling** involves
  - computation of output image, *O,* pixel values
  - by interpolation of input image, *I,* pixel values
  - via inverse transform T^{-1} from pixel positions in *O*
- Basically, for each pixel in the output:
  - ask where that position came from (in the original) and
  - what the value was there (find by interpolation)

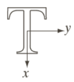
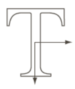
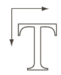
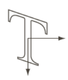


Affine Transformations

Affine transformations are simple matrix multiplications

$[x\ y\ 1] = [v\ w\ 1]\ \mathbf{T}$

TABLE 2.2
Affine transformations based on Eq. (2.6.–23).

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

Adapted from Table 2.2 in "Digital Image Processing, 3rd Edition"

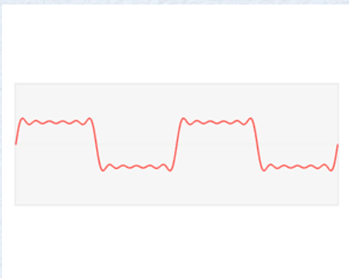
Frequency Domain

Fourier Transformations

- **Why** do it?
  - Makes things **easier** or **faster**

$$\begin{tikzcd}
\text{(problem)} \arrow[r, "\text{Fourier transform}"] \arrow[d, dashed, "\text{solve}"', "\text{(hard)}"] & \text{(transformed problem)} \arrow[d, "\text{solve}"', "\text{(easy)}"] \\
\text{(solution)} & \text{(transformed solution)} \arrow[l, "\text{inverse Fourier transform}"']
\end{tikzcd}$$

- **What** is it and **how** is it done?
  - Take a course, or
  - study the wiki-page for 1 month to get a solid grasp, or
  - accept that it is mathematically sound although it looks like magic/nonsense



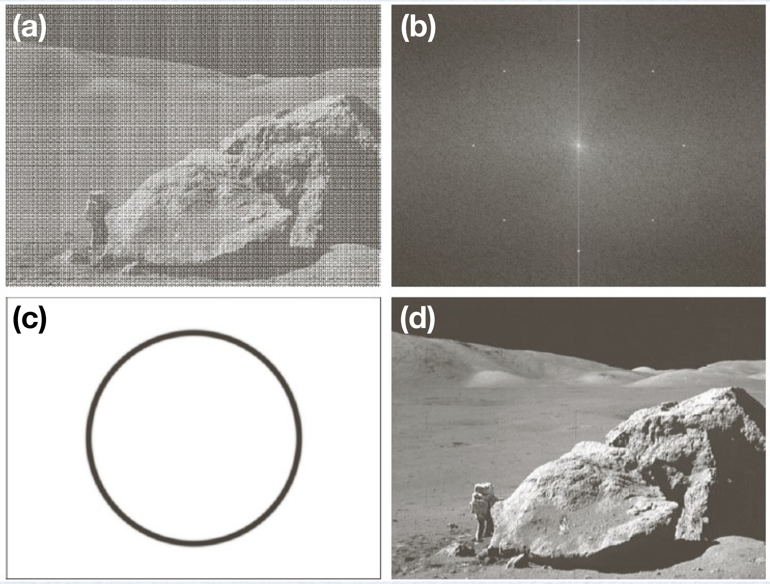
1-dim FT $\quad X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N}.$

1-dim inverse FT $\quad x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{i2\pi kn/N},$

d-dim FT $\quad X_{k_1,k_2,\dots,k_d} = \sum_{n_1=0}^{N_1-1} \left(\omega_{N_1}^{k_1 n_1} \sum_{n_2=0}^{N_2-1} \left(\omega_{N_2}^{k_2 n_2} \cdots \sum_{n_d=0}^{N_d-1} \omega_{N_d}^{k_d n_d} \cdot x_{n_1,n_2,\dots,n_d} \right) \right),$

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

FREQUENCY-FILTERING



a b
c d

FIGURE 2.40
(a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy of NASA.)

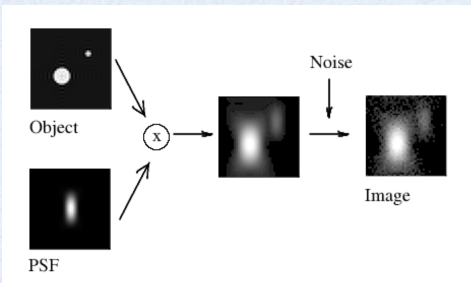
Example of image restoration by **filtering in frequency-space** (Fourier domain). Butterworth band-reject filter used in (c).
In this case, operating in frequency space was both faster and better than filtering in real-space.

Adapted from Fig. 2.40 in “Digital Image Processing, 3rd Edition”

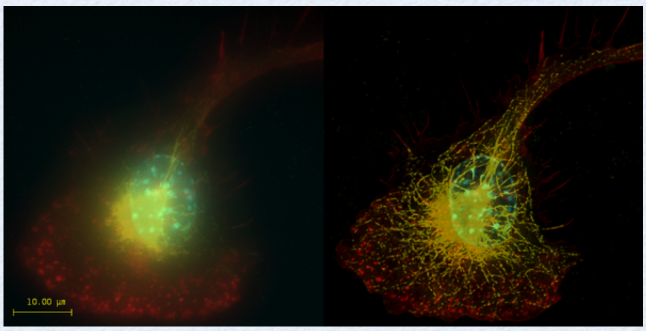
DECONVOLUTION

Any image is the **convolution** of the **point-spread-function** (PSF) of the entire imaging system with the **object**, plus **noise** with camera-dependent statistics.

Convolution is a mathematically invertible operation = **deconvolution.**
Addition of noise makes the problem much harder.



$$g = f * h = \int\int\int_{-\infty}^{\infty} f(\vec{x})\, h(\vec{x} - \vec{x}')\, d^3\vec{x}'$$



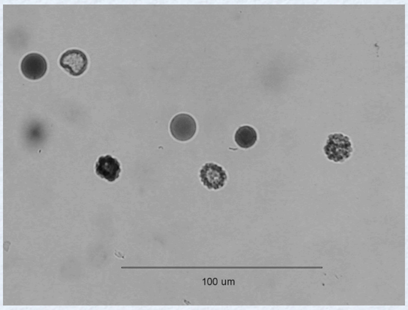
Example of deconvolution (right) of a macrophage

http://www.svi.nl/HuygensDeconvolution

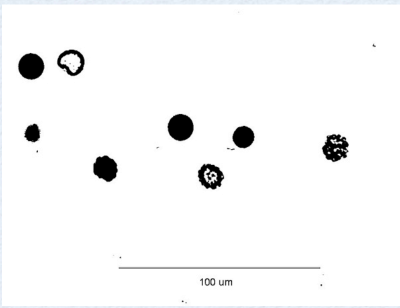
Segmentation

- Segmentation is the **division** of an image into **discrete regions**.
- It can be achieved by an ever-expanding list of methods.
  - Thresholding
  - Edge-based
  - Active contours
  - Water-shed
  - k-means clustering
  - ...

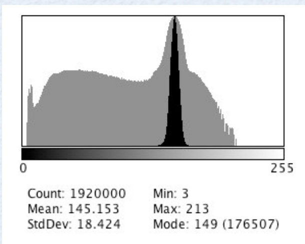
Thresholding

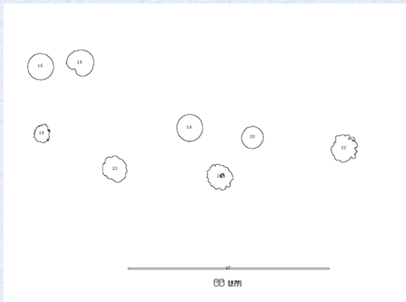


Original, 8bit grayscale
Embryos: Fiji example



Thresholded @ 104 (Otsu)



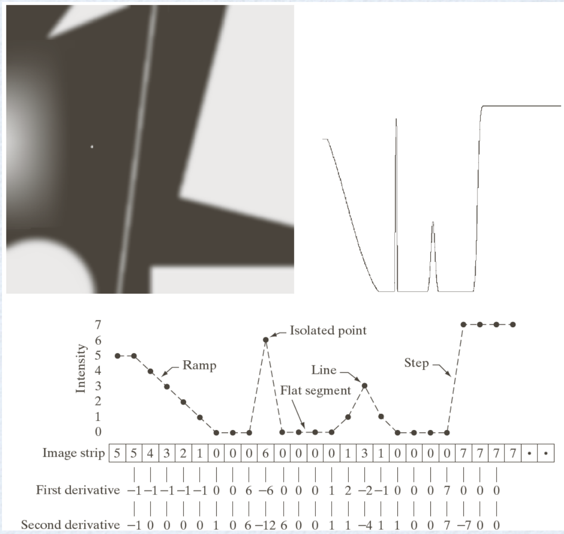


Segmented

Segmentation here consisted in keeping (connected) black objects larger than 50 pixels^2.
Image and operations from Fiji/ImageJ.

Edge Detection

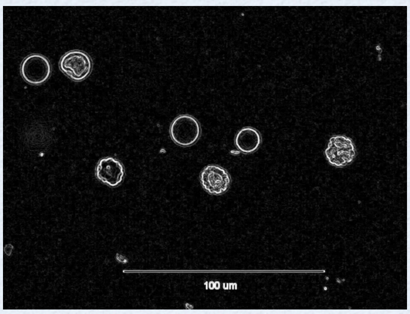
- Edges are found by calculating the **derivatives** of the image
  - first order: detects slopes
  - second order: detects changes in slopes
- Differentiation enhances noise
- Smooth the image first
- Sobel: 1st order
- Canny: 1st order, smooth & detect



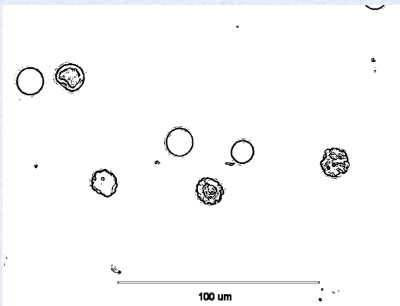
Sobel

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

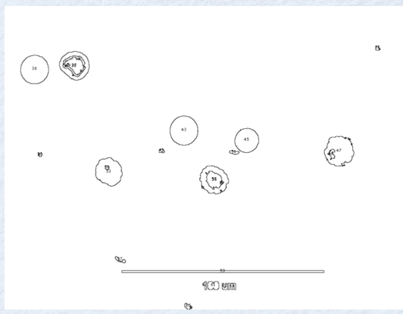
Edge Detection



Edges from original

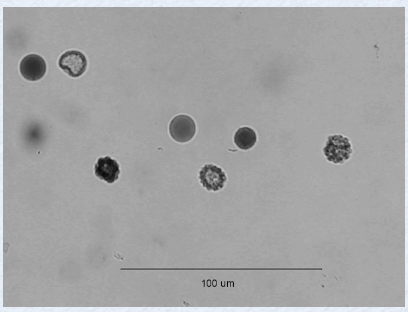


Thresholded edges (Otsu)



Segmented binary

Original



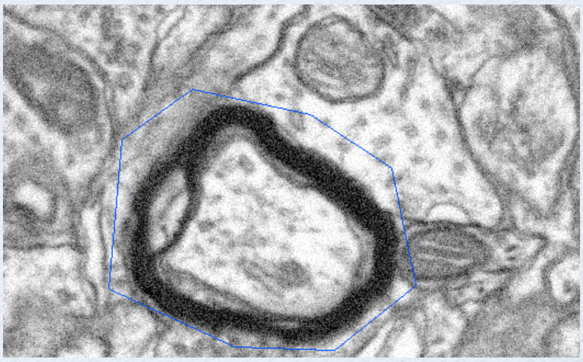
Segmentation, as before, consisted in keeping (connected) black objects larger than 50 pixels^2.
Image and operations from Fiji/ImageJ.

Active contours

$$E^*_{snake} = \int_0^1 E_{snake}(\mathbf{v}(s))\,ds = \int_0^1 (E_{internal}(\mathbf{v}(s)) + E_{image}(\mathbf{v}(s)) + E_{con}(\mathbf{v}(s)))\,ds$$

$$E_{internal} = (\alpha(s)\,|\mathbf{v}_s(s)|^2 + \beta(s)\,|\mathbf{v}_{ss}(s)|^2)/2$$

- **Minimize an energy** associated to the current contour as a sum of an internal and external energy:
  - The external energy is supposed to be minimal when the snake is at the **object boundary position**.
  - The most straightforward approach consists in giving low values when the regularized gradient around the contour position reaches its peak value.
- The internal energy is supposed to be minimal when the snake has a shape which is supposed to be relevant considering the shape of the sought object
- http://demo.ipol.im/demo/abmh_real_time_morphological_snakes_algorithm/
  - parameters: sigma = 20; dark center lines; contraction balloon force



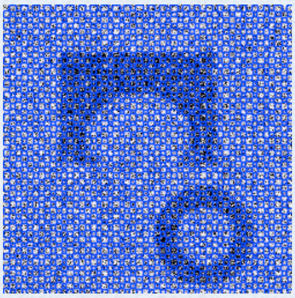
Chan-Vese Segmentation

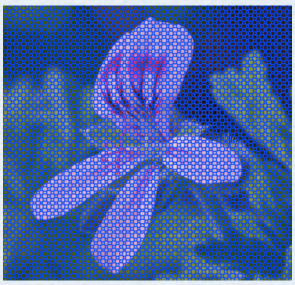
- Chan-Vese Segmentation
  - http://demo.ipol.im/demo/g_chan_vese_segmentation/
- "Active Contours Without Edges" method by Chan and Vese
- Optimally fits a two-phase piecewise constant model to the given image

For the numerical implementation, suppose that f is sampled on a regular grid $\Omega = \{0, \dots, M\} \times \{0, \dots, M\}$. The evolution of φ is discretized in space according to

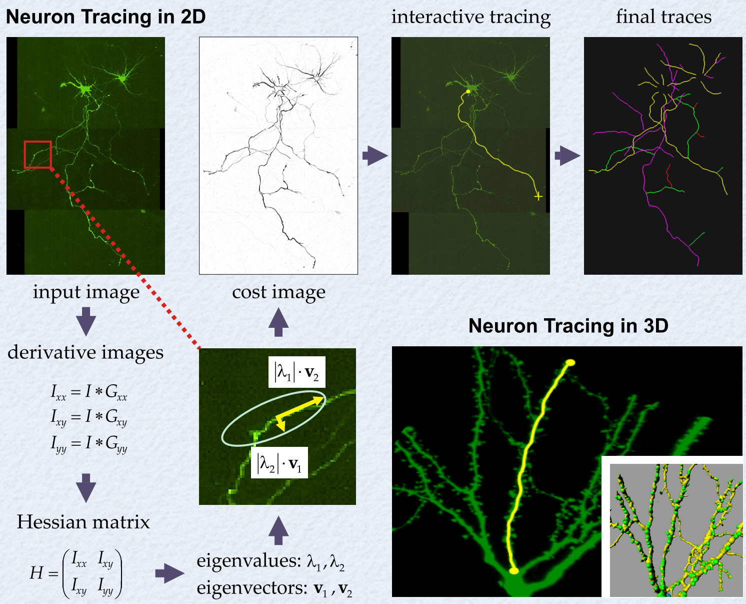
$$\frac{\partial \varphi_{i,j}}{\partial t} = \delta_\epsilon(\varphi_{i,j}) \Bigg[\mu \Bigg(\nabla_x^- \frac{\nabla_x^+ \varphi_{i,j}}{\sqrt{\eta^2 + (\nabla_x^+ \varphi_{i,j})^2 + (\nabla_y^0 \varphi_{i,j})^2}} + \nabla_y^- \frac{\nabla_y^+ \varphi_{i,j}}{\sqrt{\eta^2 + (\nabla_x^0 \varphi_{i,j})^2 + (\nabla_y^+ \varphi_{i,j})^2}} \Bigg) - \nu - \lambda_1 (f_{i,j} - c_1)^2 + \lambda_2 (f_{i,j} - c_2)^2 \Bigg], \quad i,j = 1, \dots, M-1, \tag{17}$$

where ∇_x^+ denotes forward difference in the x dimension, ∇_x^- denotes backward difference, and $\nabla_x^0 := (\nabla_x^+ + \nabla_x^-)/2$ is central difference, and similarly in the y dimension. The curvature is regularized by





Neuron Tracing



NeuronJ: plugin for ImageJ

Meijering and van Cappellen. Biological Image Analysis Primer. (2006)

Feature Extraction

- After segmentation, measure **features**
  - area, circumference, ellipticity, ...
  - orientation, pose,
  - intensity (mean, std, max, min)
  - Haralick features (texture)
- Use features to answer specific question, or
- input features to **machine learning** algorithm to
- **sort objects** into different **categories**

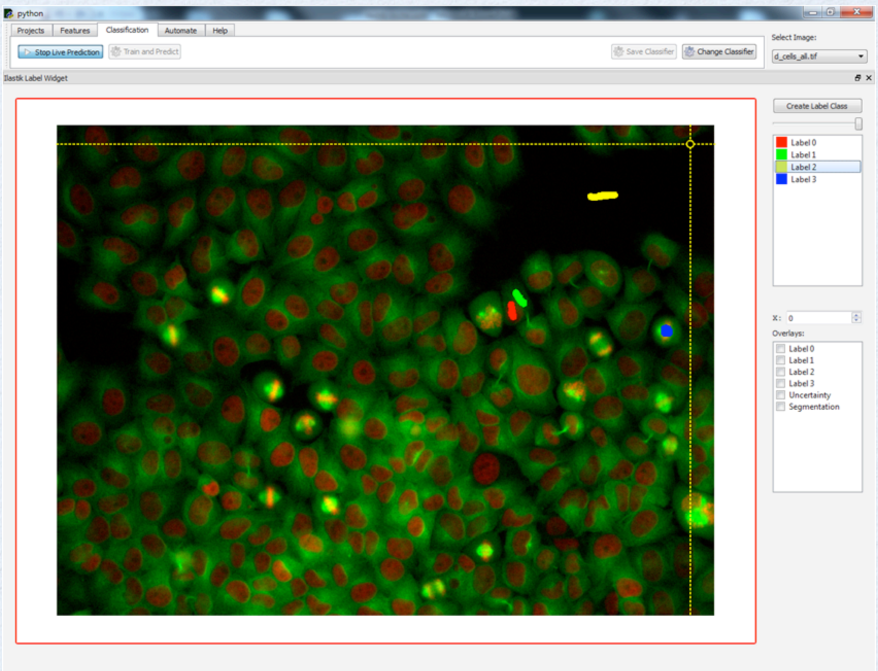
- This is the **analysis** part (returns numbers)
  - allows compact summary of relevant aspects of the images analyzed
  - quantified features can be analyzed statistically

www.ilastik.org

- What is ilastik?
  - Interactive Learning and Segmentation Toolkit
  - first released in 2011 by scientists at University of Heidelberg
  - user-friendly tool for image **classification** and **segmentation** in up to three spatial and one spectral dimension
  - allows user to **annotate** an arbitrary number of **classes** in images with a mouse interface
  - **features** includes **color**, **edge** and **texture** descriptors
  - advanced users can add their own problem-specific features
  - trains a random forest classifier
  - bases its predictions on local (not global) properties
  - used in CellProfiler (image-based screening)

www.ilastik.org

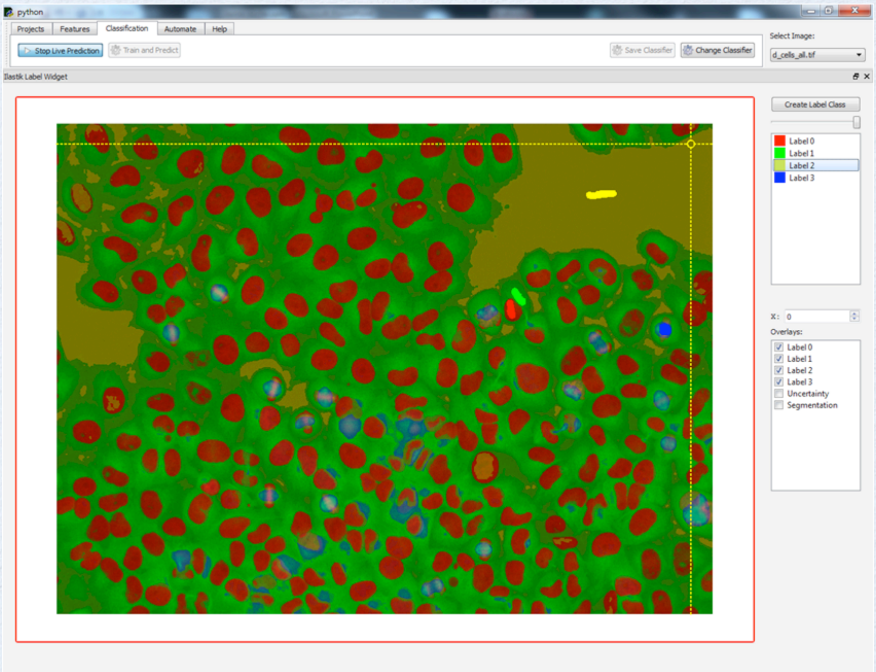
Initial labels



nucleus
cytoplasm
background
weird stuff

www.ilastik.org

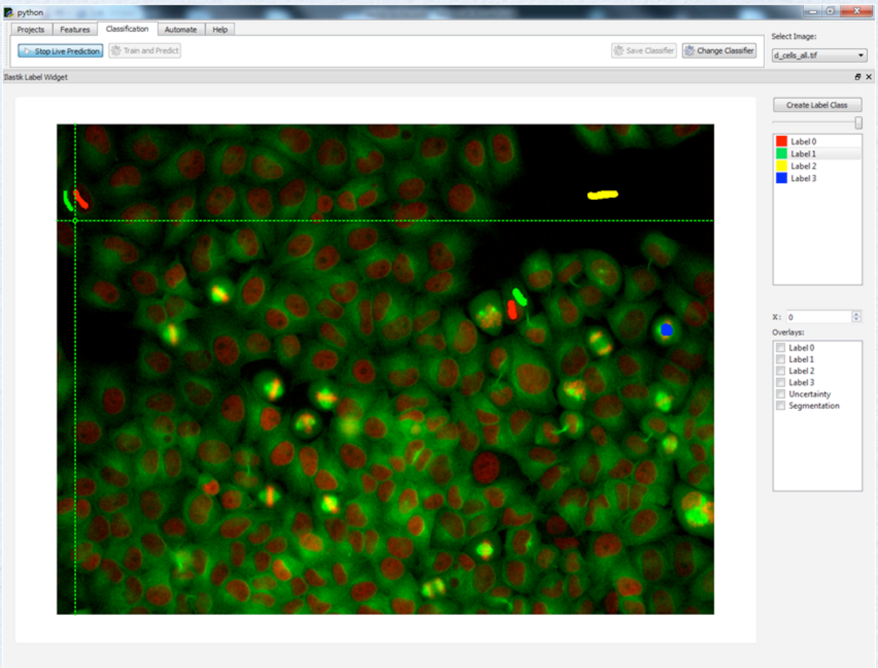
Interactive prediction



nucleus
cytoplasm
background
weird stuff

www.ilastik.org

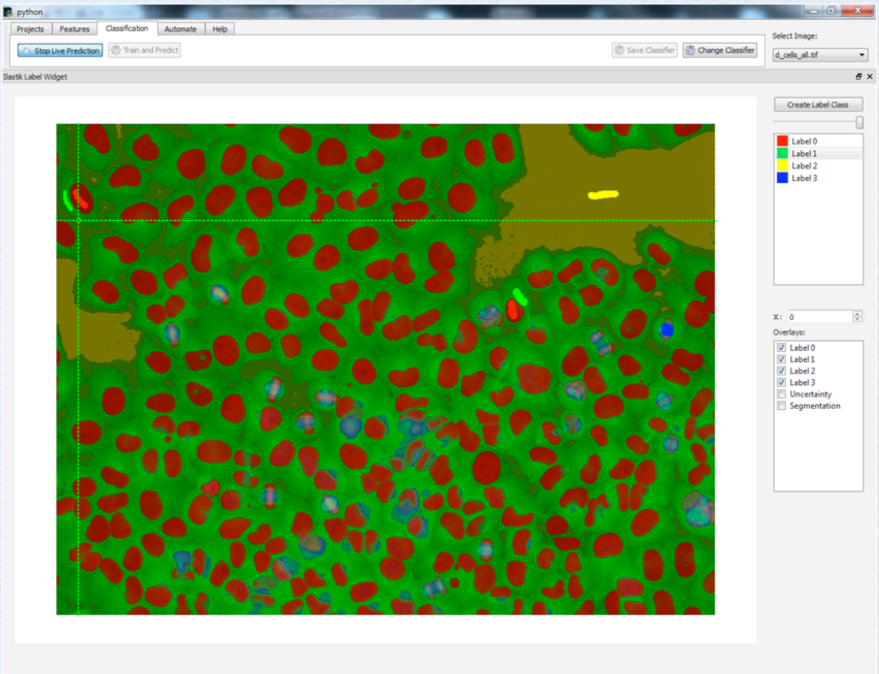
Refinement of labels



nucleus
cytoplasm
background
weird stuff

www.ilastik.org

Updated result

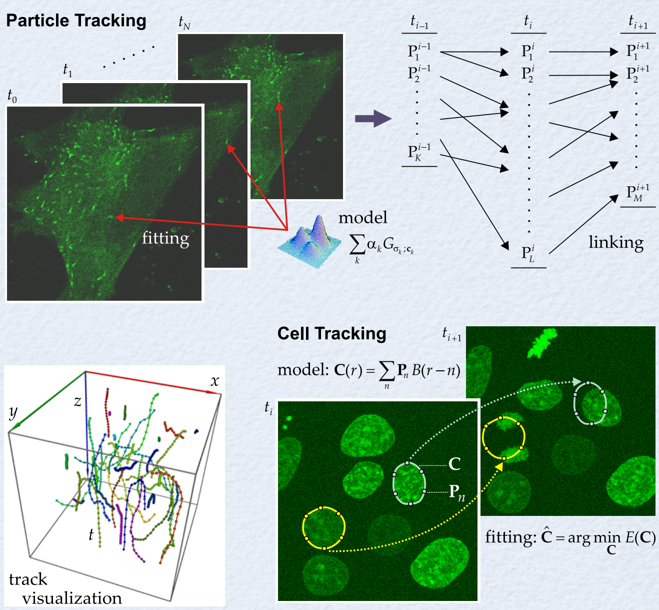


nucleus
cytoplasm
background
weird stuff

Putting it all together

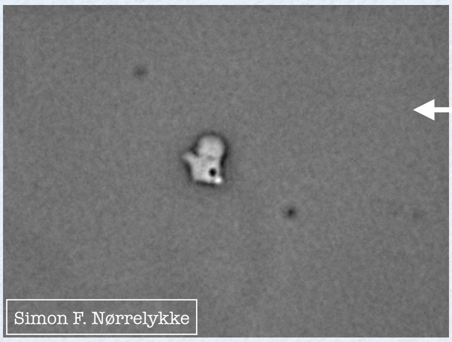
TRACKING

- **Tracking** of objects
  - particles (point objects)
  - cells (extended, deformable objects)
- has two separate steps
  - **detection** of objects in each frame
  - **linking** of objects between frame
- Detection can use
  - external knowledge (point spread function)
  - constraints (mechanistic model for shape)
  - information from previous slide
- Linking can use
  - only the past, to predict future (on the fly)
  - all pre-recorded data (off-line processing)



Meijering and van Cappellen. Biological Image Analysis Primer. (2006)

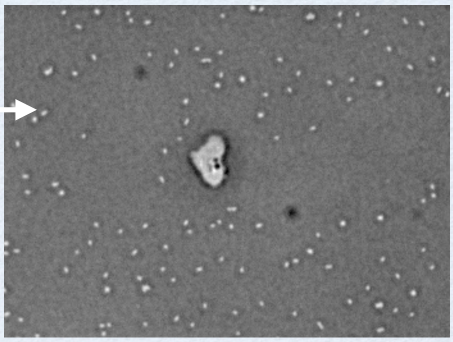
SAME CELL WITHOUT/WITH FOOD

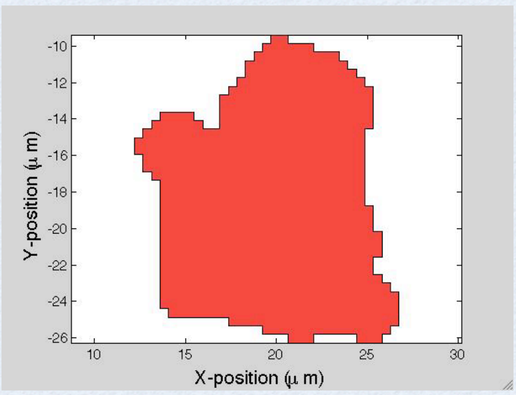


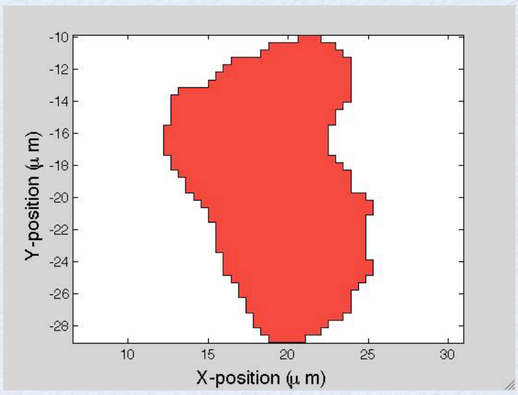
300x real time
(1s = 5min)

163µm x 122µm;
100min total

not looping



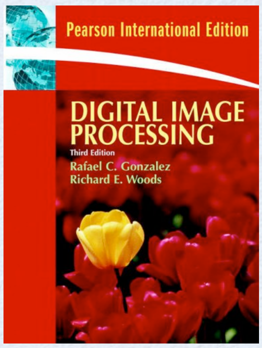




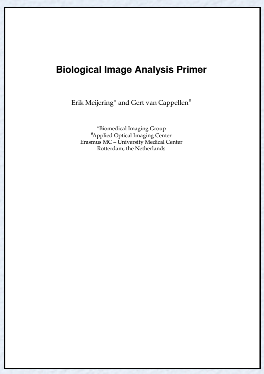
FURTHER

FURTHER READING

- **Digital Image Processing, 3rd edition**
  - by Gonzales and Woods (2008)
  - ~950 pages, 66 EUR (amazon.de)
- Website for the book:
  - http://www.imageprocessingplace.com/DIP-3E/dip3e_main_page.htm
  - all images and figures from the book
  - solutions to selected problemsets
- Assessment:
  - Thorough
  - Not bio-image specific
  - Does not assume any/much math knowledge



- **Biological Image Analysis Primer**
  - Meijering and van Cappellen (2006)
  - 37 pages, free pdf:
  - http://www.imagescience.org/meijering/publications/download/biap2006.pdf

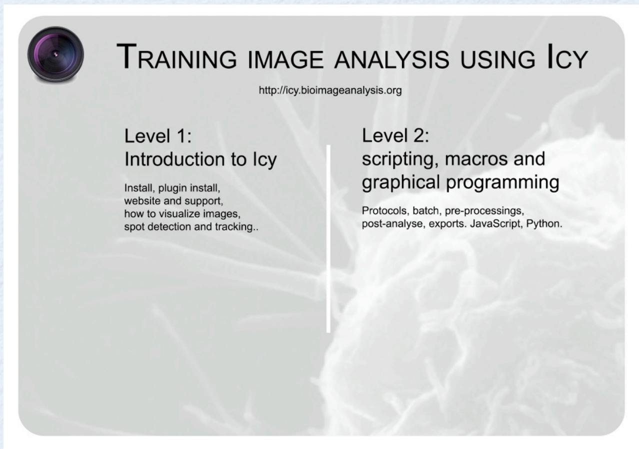


Web-resources

- “Digital Image Processing” (the book):
  - http://www.imageprocessingplace.com
  - image databases
  - solutions to problem sets from the book
- Interactive tools to learn about image processing:
  - http://learn.hamamatsu.com/tutorials/flash/imageprocessing
- Everything light-microscopy related:
  - http://www.olympusmicro.com
- Basic Properties of Digital Images:
  - http://learn.hamamatsu.com/articles/digitalimagebasics.html
- Free online courses:
  - Coursera: www.coursera.org
    - Image and video processing: From Mars to Hollywood with a stop at the hospital
    - Computer Vision: From 3D Reconstruction to Visual Recognition
    - Fundamentals of Digital Image and Video Processing
  - Udacity: https://www.udacity.com
  - EdX: https://www.edx.org

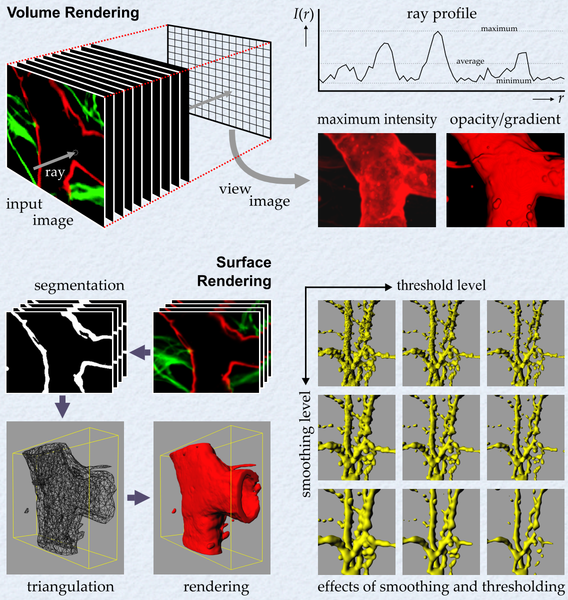
ICY

- An open community platform for bioimage informatics
  - http://icy.bioimageanalysis.org
- Practical on Thursday 12th
- Flying in two ICY developers from Paris



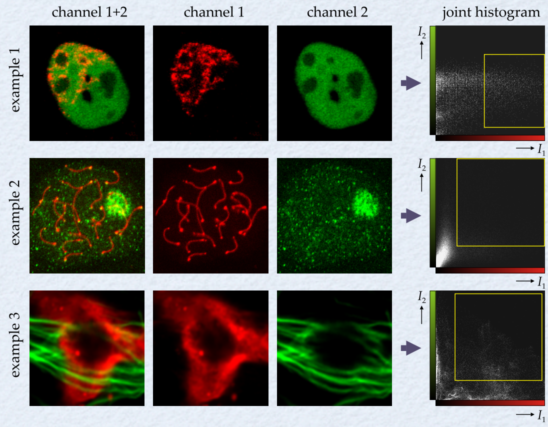
Extra Slides

Volume Rendering



Meijering and van Cappellen. Biological Image Analysis Primer. (2006)

Co-localization



Colocalization Measures

Pearson's correlation coefficient: $r_P = \frac{\sum (I_1 - \bar{I}_1)(I_2 - \bar{I}_2)}{\sqrt{\sum (I_1 - \bar{I}_1)^2 \sum (I_2 - \bar{I}_2)^2}}$

example	r_P	r	k_1	k_2	m_1	m_2
1	0.49	0.58	0.79	0.43	0.99	0.15
2	0.39	0.71	0.81	0.62	0.35	0.24
3	-0.08	0.37	0.24	0.57	0.23	0.38

overlap coefficient: $r = \sqrt{k_1 k_2}$ with

$$k_1 = \frac{\sum I_1 I_2}{\sum I_1^2} \quad \text{and} \quad k_2 = \frac{\sum I_1 I_2}{\sum I_2^2}$$

Manders' colocalization coefficients:

$$m_1 = \frac{\sum I_{1\,\mathrm{coloc}\,2}}{\sum I_1} \quad \text{and} \quad m_2 = \frac{\sum I_{2\,\mathrm{coloc}\,1}}{\sum I_2}$$

- A statistical analysis of spatial colocalization using Ripley's K function.
  - Lagache et al. (ISBI 2013)
  - Implemented as plugin in ICY

Meijering and van Cappellen. Biological Image Analysis Primer. (2006)

Line-segment detector

- http://demo.ipol.im/demo/gjmr_line_segment_detector/
- find parameters