Image Analysis

Rasmus R. Paulsen
Tim B. Dyrby

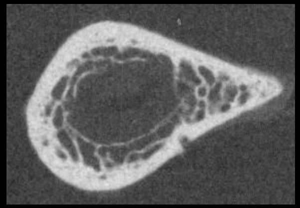
DTU Compute

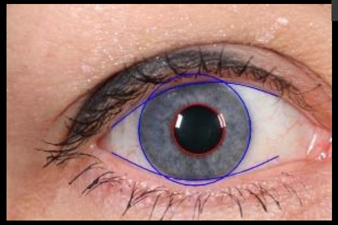
http://compute.dtu.dk/courses/02502

Week 1 - today

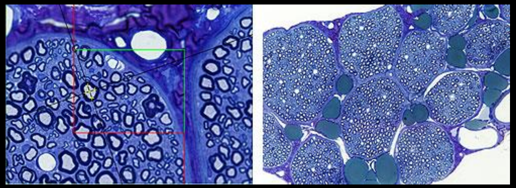
8:00 – 10:00	Exercises
10:00 – 12:00	Introduction and practical matters
	Lecture – Digital Images
	Lecture – A tutorial on Principal Component Analysis (PCA)

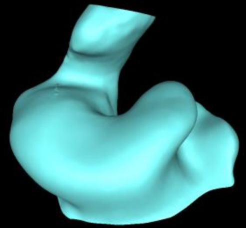
Rasmus R. Paulsen

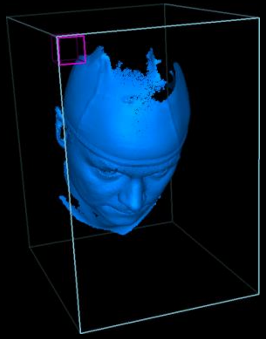




- Master of Science (Eng). DTU 1998
- Industrial PhD with Oticon A/S
- Research and development at Oticon A/S
- Associate Professor DTU Compute







Tim B. Dyrby

- Associate Professor at DTU Compute and Danish Research Centre for Magnetic Resonance (DRCMR)

Teaching Assistants

Mathias Micheelsen Lowes
Mathematical Modelling and Computation

Bjørn Marius Schreblowski Hansen
Mathematical Modelling and Computation

Paula Lopez Diez
Mathematical Modelling and Computation

Practical matters

- 13 days over the DTU 13 week semester
- Flipped class room
  - 8-10 Computer exercises (also on MS Teams)
  - 10-12 Lecture with quizzes

- Lectures are streamed, recorded and made available
  - Links to video on the homepage (under schedule)
  - Courses.compute.dtu.dk/02502

About this course

- Until 2017 the course responsible was Jens Michael Carstensen – CEO of Videometer
  - Now full time at Videometer
  - Will give a guest presentation at the *company presentation day*

- From 2018 Rasmus R. Paulsen is the course responsible
  - Major course revision
  - Other topics and new examples
  - Material from course 02512

- From 2019 Tim B. Dyrby is also teaching the course

New in 2021

- The exercises are now much more related to the exam
- Learning objectives stated in all exercises
- You will be examined in these learning objectives
- You will also be examined in the more theoretical learning objectives from the lectures
- We will expect you can run Matlab during the exam

Very Important!
Do the exercises!

Materials

- Book:
  - Rasmus R. Paulsen and Thomas B. Moeslund: *Introduction to Medical Image Analysis* **(MIA)**
  - Polyteknisk boghandel
  - http://mediabook.compute.dtu.dk

- Notes
  - Notes will be provided during the course

- At the end of the course a complete reading list will be published

DTU Learn and the homepage

- Homepage : The main entry to the course
  - http://courses.compute.dtu.dk/02502
  - Schedule / Exercises / Data
  - Updates happen!

- Course messages will be given through DTU Learn

#	Date	Topic	Video	Material	Exercise
1	31/8	Introduction and digital images. Introduction to Principal Component Analysis (PCA) (Rasmus)		MIA 1, 2, app. A. PCA Note (except Section VI (SVD) and App. A)	1
2	7/9	Image acquisition. Compression and storage (Tim)		MIA 2, 3	1 + 1b
3	14/9	Pixelwise operations, colour images and PCA Analysis on images (Rasmus)		MIA 4, 8 Eigenfaces article (only sections marked with yellow)	2
4	21/9	Neighborhood Processing (Filtering) and Morphology (Rasmus)		MIA 5, 6	3 + 3b
5	28/9	Blob analysis and object classification (Rasmus)		MIA 7	4 + 4b
6	5/10	Pixel classification and advanced classification (Tim)		MIA 9 + notes on DTU Learn	5
7	12/10	Geometric transformations and landmark based registration (Tim)		MIA 10, 11	6 + 6b
8	26/10	Boundary Tracing (Hough Transformation and Dynamic Programming) (Tim)		MIA 12	7
9	2/11	Statistical Models of Shape and Appearance (Rasmus)		Statistical Models of Appearance for Computer Vision (p. 12 - 20 and p. 29 - 33)	8
10	9/11	Advanced registration (Tim)		Notes on DTU Learn	9
11	16/11	Industry presentations		none	10
12	23/11	Active Shape Models (Rasmus)		Statistical Models of Appearance for Computer Vision (p. 34 - 43)	Exercise catch-up
13	30/11	Advanced topics		none	Digital test exam

Book (MIA): Rasmus R. Paulsen og Thomas B. Moeslund. Introduction to Medical Image Analysis. Springer Nature.

Learning Objectives (Læringsmål)

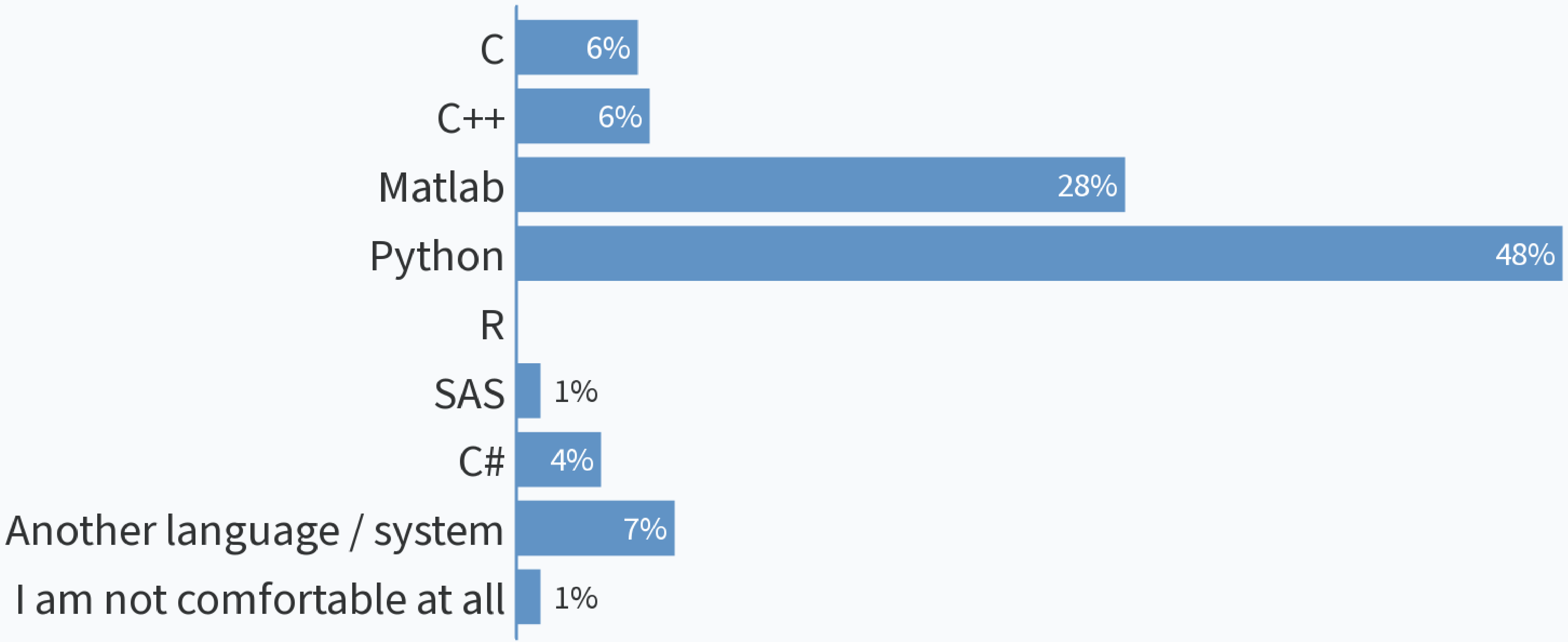
- A list of learning of objectives for each lecture and exercise
- A learning objective describes what you can do after the lecture/exercise
- If you fulfil all learning objectives you get 12
- Low-level learning objective
  - Apply the Prewitt edge filter to an image
- High-Level learning objective
  - Evaluate and compare the performance of a selection of image analysis algorithms

Exam

- Four hours multiple-choice exam
- Please see details here:
  - http://courses.compute.dtu.dk/02502/exam.html

- Previous exam sets are also available
  - Most relevant is from Spring 2021

What programming language are you most comfortable with?

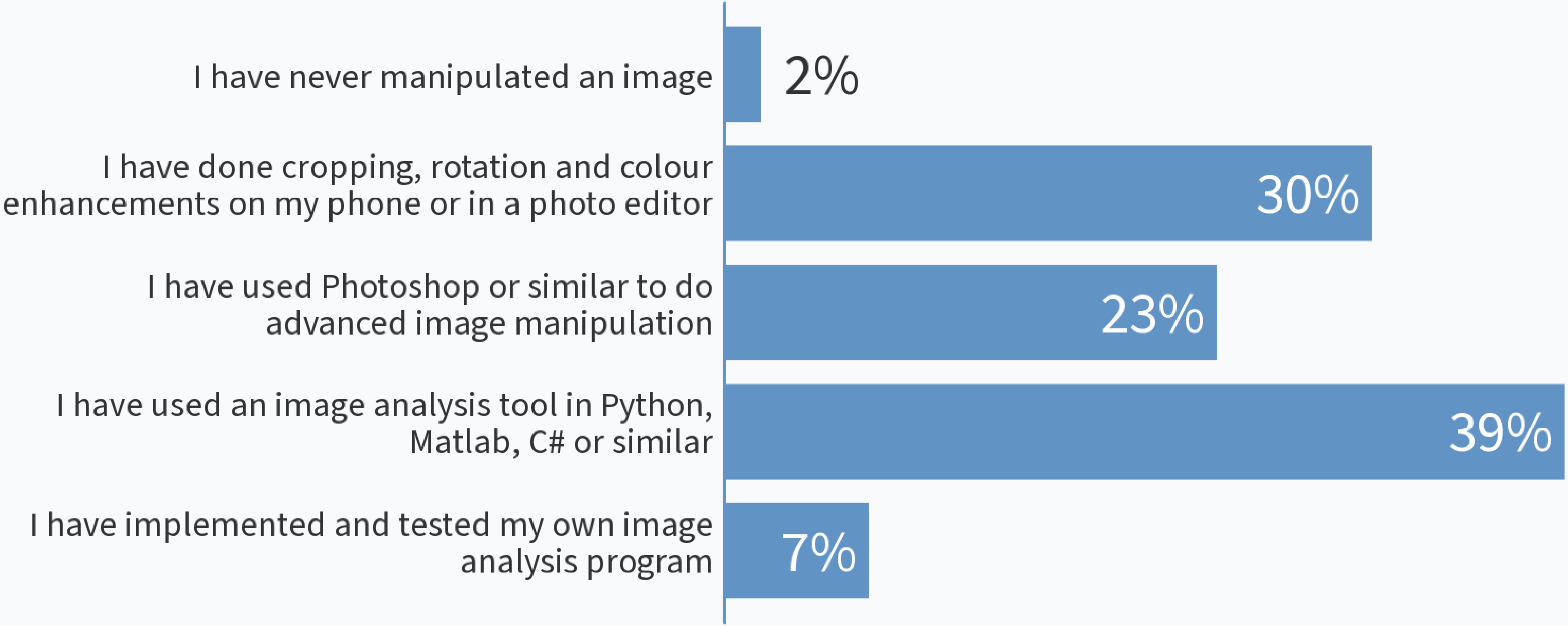


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Matlab and computers

- We assume that you can use your own portable computer with Matlab
- Python: Some exercises can potentially be made using Python
  - The TA will help the best they can
- We expect you to be able to run Matlab during exam
- Why Matlab:
  - Requires massive resources to do exercises, TA'ing and exams in several languages
  - Most important to learn the concepts of image analysis – implementation is dependent on the project and company

What is your experience with image manipulation, image processing and image analysis?

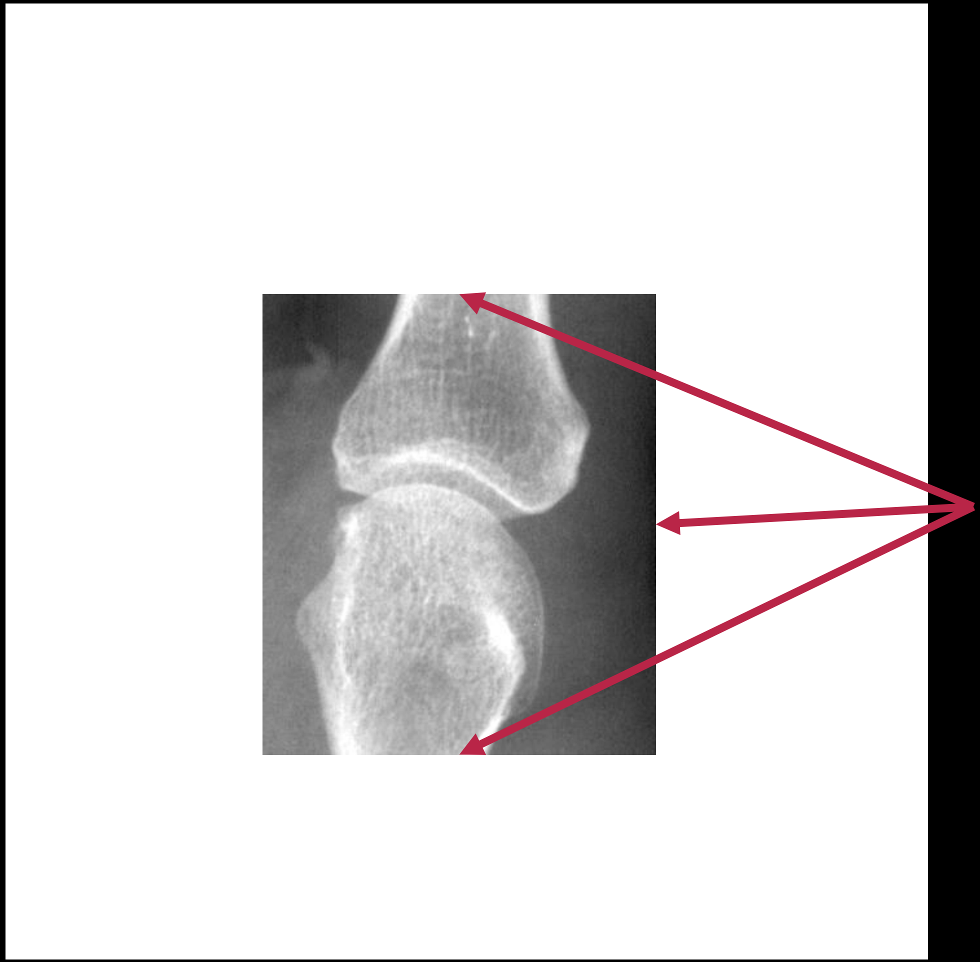


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Why are my slides black?

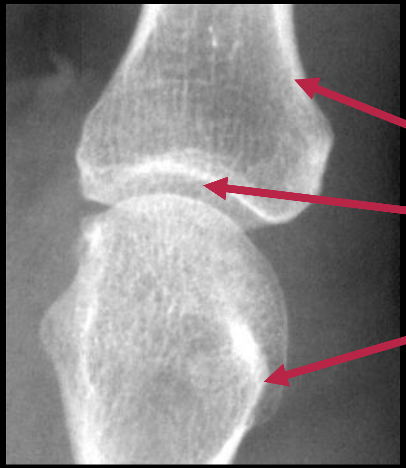
?

Norwegian Black Metal



With a white background, the strongest visual contrast is here

Why are my slides black?



With a dark background, the strongest visual contrast is here

(which I find more important)

What is image analysis

- Automatic extraction of information from images
- A sub-topic within
  - Pattern recognition
  - Machine learning
  - Deep learning

What is image processing

- Changing the information in images – but not necessarily getting any knowledge
  - Photoshopping
  - Changing the visual appearance of photos
  - Cropping / rotating
  - Filters / effects

Classical machine vision

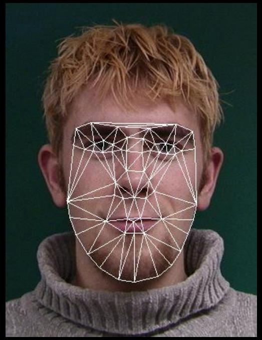
- Tomato sorting machine
  - Good tomatoes vs green/bad tomatoes
- Combination of
  - Very fast cameras
  - Fast classification algorithms
  - Robotics

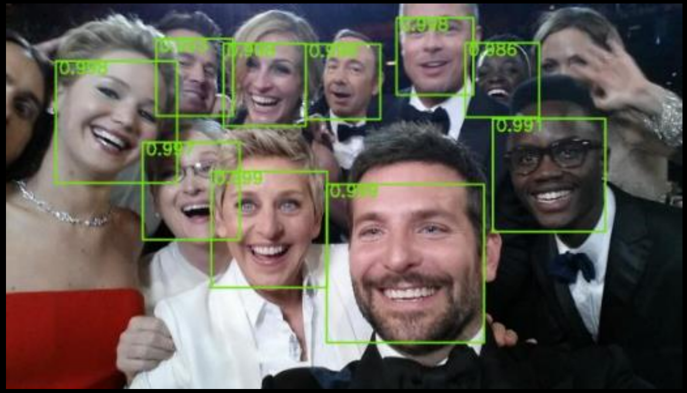
- https://www.youtube.com/watch?v=Lz88nsWL4kw

Local companies (some will be here on the company day):
JLI vision, Videometer, IHfood, Trivision

Face tracking – all features including eyes

- For digital cameras / phones
  - Automatic focus on the face + face beautification
- Tracking and manipulation for apps
  - Messenger / WhatsApp / SnapChat …
- Awareness tracking for car drivers
  - Warning if you fall a sleep





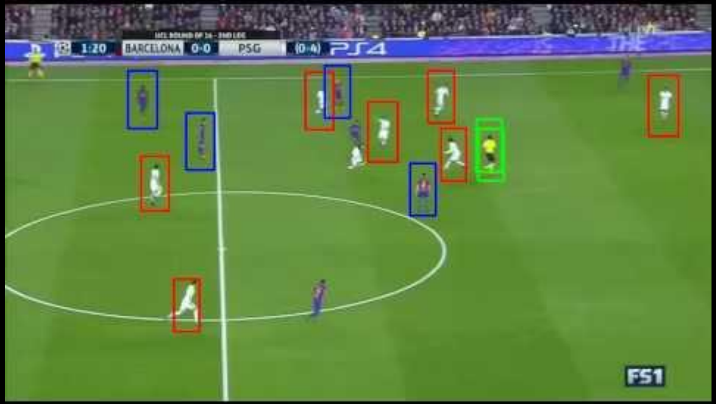
A 100 million $ industry

- This image is worth 100 of millions of dollars!
- Well – perhaps not that exact photo.
- The ability to track faces fast and accurate
  - Including estimates of 3D structure
  - App developers pays buckets of money for that
- It all started in 2001 with:
  P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features.“. CVPR 2001
- Suddenly you could track faces fast and relatively accurate
- Now it is all deep learning

Self driving cars

- Modern self driving cars rely on many sensors
  - Lidar – radar system
  - GPS
  - Accelerometers, gyroscopes, magnetometers etc.
  - Stereo cameras or multiple cameras
  - Lots of advanced image analysis – sensor fusion

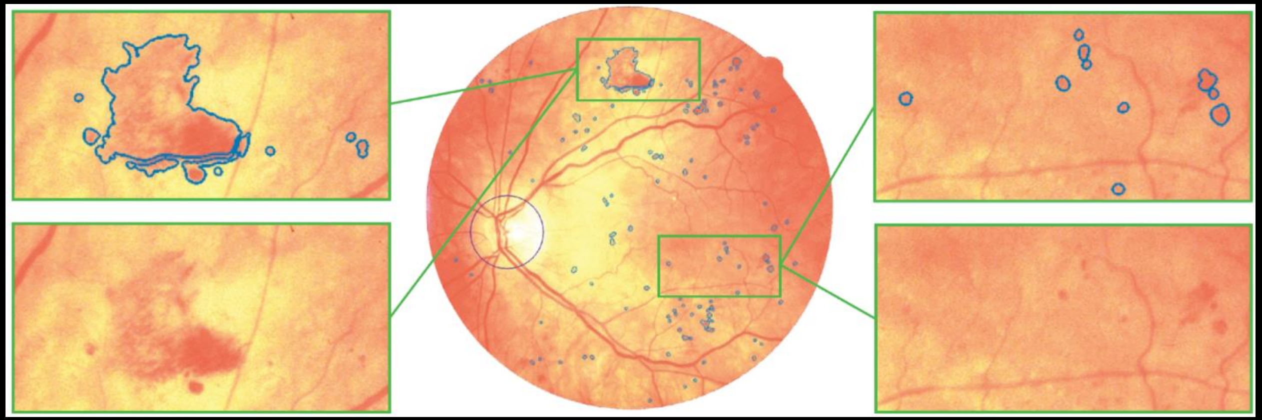
Sports tracking – human body tracking



- Huge commercial impact
- Lots of research in human body tracking
- Personal trainers

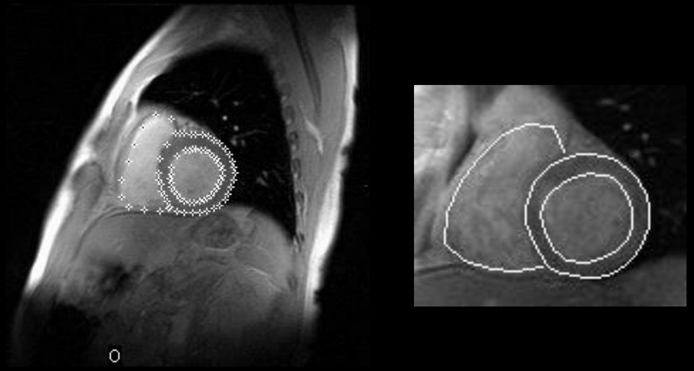
What is medical image analysis?

- Extraction of information from digital images
- Reproduce expert diagnostics
  - More accurate
  - Variation between doctors opinions removed
- Computer aided diagnostics – the doctor has the last word
- Can enhance the signs of diseases
  - Tumours
  - Bleedings



Automatically detected haemorrhages and micro aneurysms in digitized fundus images

Medical image analysis examples



Recognise and track the heart

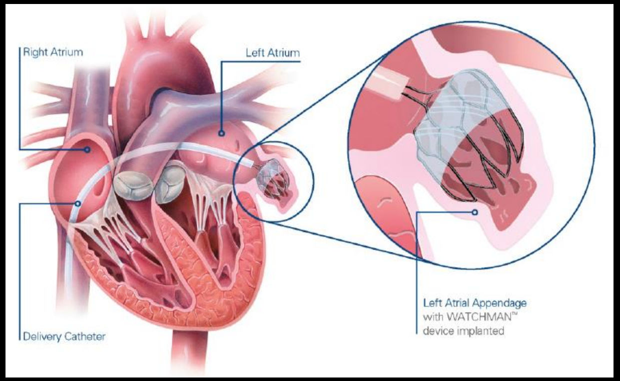
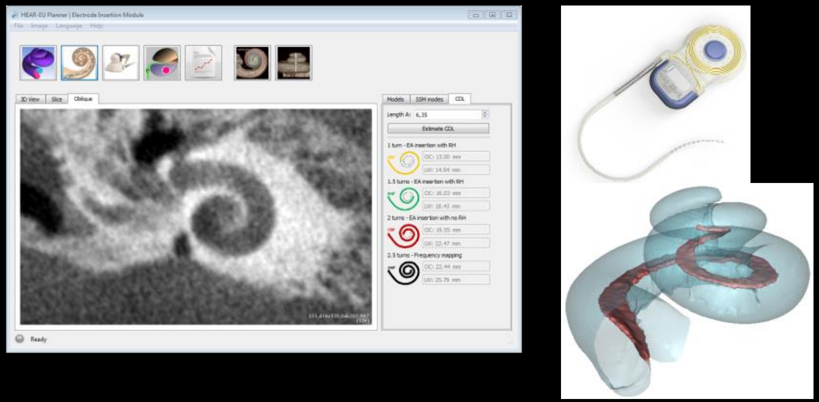
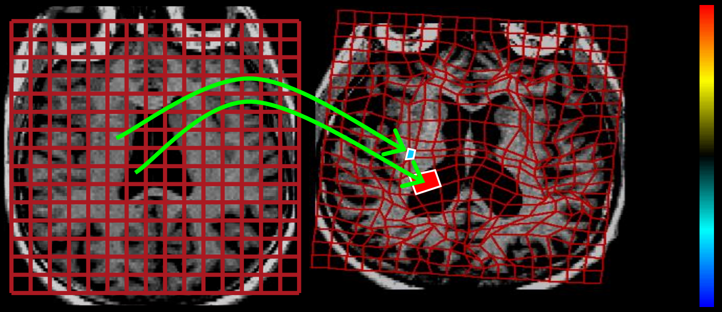


Image based surgery planning

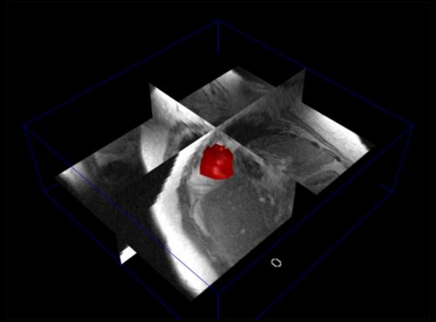


Cochlear implant planning



Shape changes in brain structures

Relevance



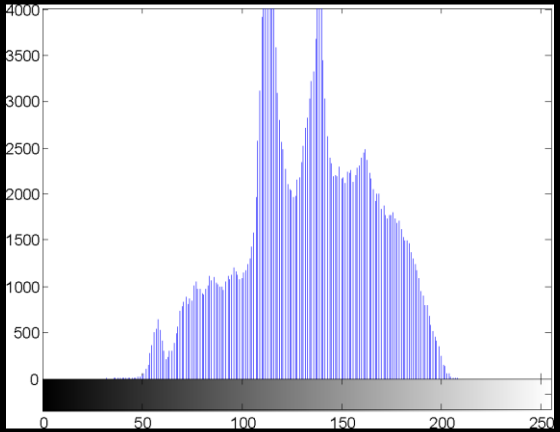
1980
Magnetic
resonance
prototype

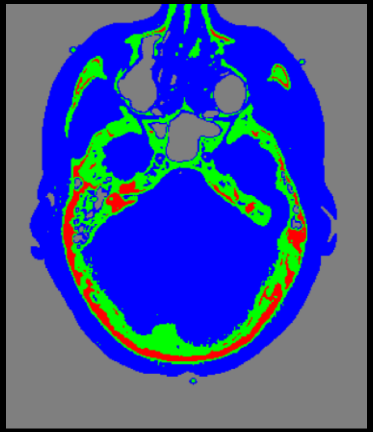
Now – PET/MR

- Images is an important tool in
  - Diagnosis
  - Treatment
  - Follow-up
- Very high-tech!
- New imaging technologies are developed all the time.

Digital Images – Learning Objectives

- Describe the fundamental properties of a digital image
- Read and show an image in Matlab
- Describe the commonly used image coordinate systems
- Describe the binary, the label, the multispectral, and the 16-bit image





A digital image

23	216	120	55
4	89	158	130
65	76	189	34
19	234	7	45

- Consists of pixels (picture elements)
- Each pixel has a value between 0 and 255? Why?

Bits and Bytes!

- A bit is a tiny tiny little switch that can be either 0 or 1 – the “memory of a computer” consists of insanely many bits
- One byte is 8 bits together. It is the “basic” unit in a computer.
- With 8 bits how many possible values can be made?
  - (2^8 = 256)

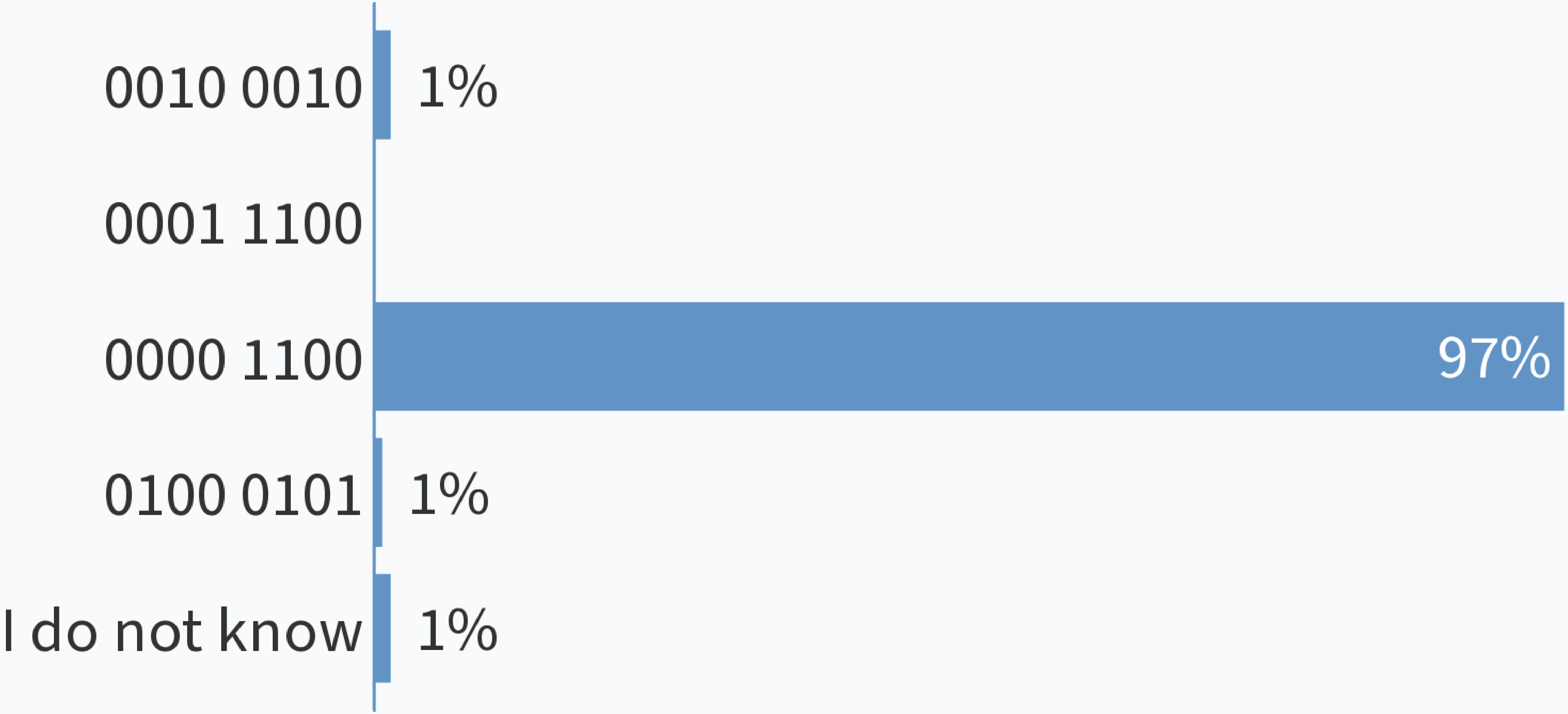
- 00000001 = 1
- 00000010 = 2
- 00000100 = 4
- 00001010 = 10
- 00001111 = 15
- 11111111 = 255

128	64	32	16	8	4	2	1
□	□	□	□	□	□	□	□

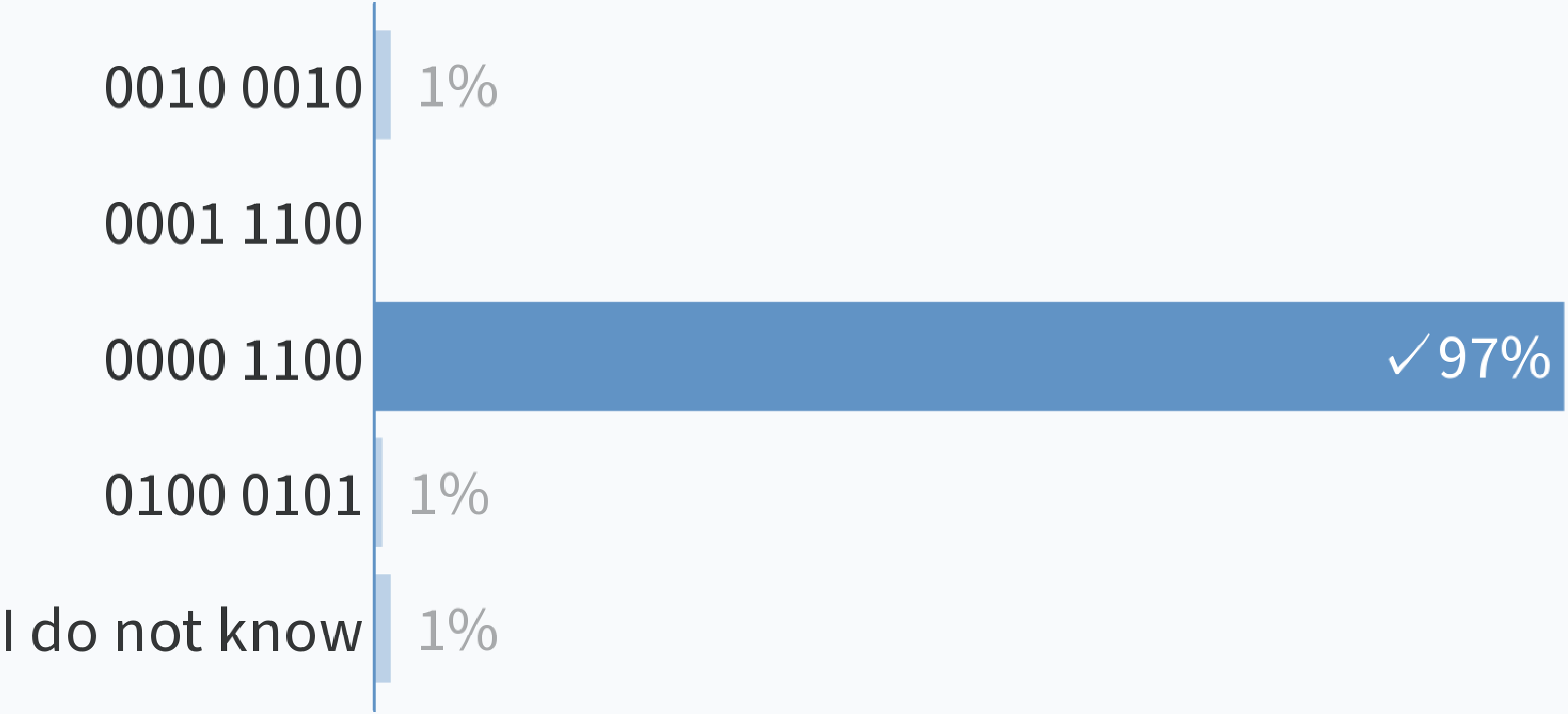
What is decimal 12 as a binary number?

- 0010 0010
- 0001 1100
- 0000 1100
- 0100 0101
- I do not know

What is decimal 12 as a binary number?



What is decimal 12 as a binary number?



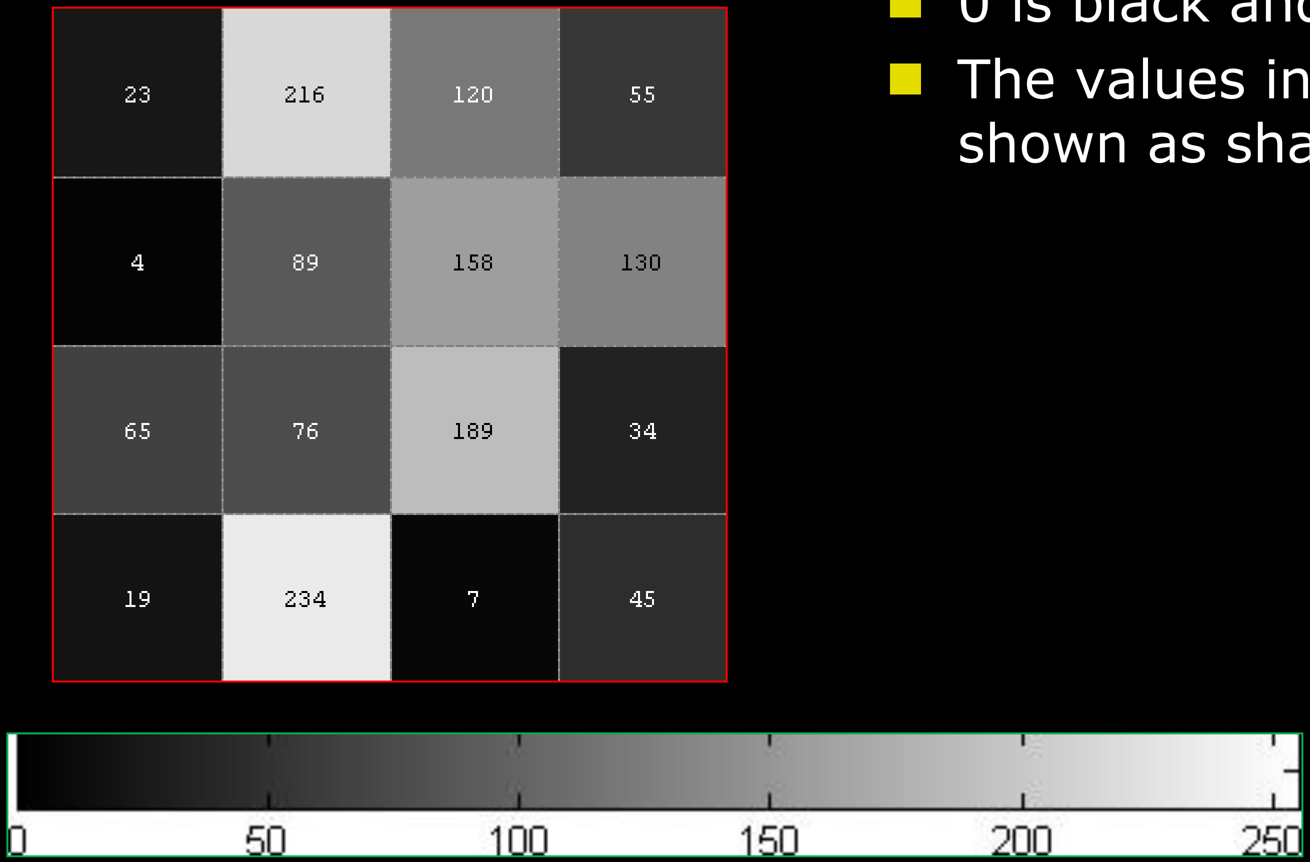
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

A digital image

23	216	120	55
4	89	158	130
65	76	189	34
19	234	7	45

- between 0 and 255.
- How many bytes do our image take up in the computer memory?
  - 16

Grayscale digital images



- 0 is black and 255 is white!
- The values in between are shown as shades of gray

Typical Grayscale image

- Traditional film X-ray
- Scanned on a flatbed scanner
- Bone is white and air is black
  - The more radiation the darker
- What are they used for?
  - Fractures
  - Arthrisis
  - Osteoporosis

Image Resolution

- Determines how much the image fills in the memory and on the hard disk
- Spatial resolution
- Gray level resolution

Spatial?

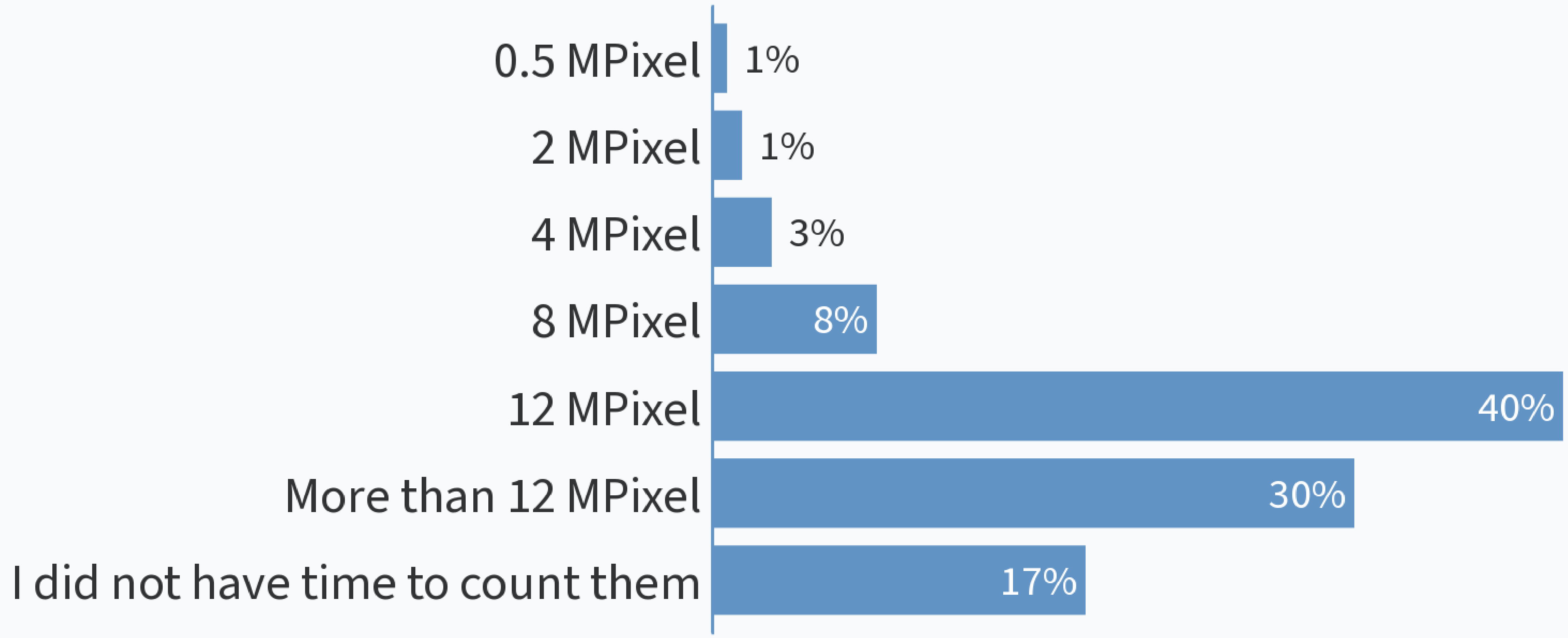
- Spatial
  - relating to the position, area and size of things
- Example:
  - This task is designed to test the child's *spatial* awareness

- Danish
  - Rumlig – barnet har en god rumlig forståelse

Spatial resolution

- The number of pixels used to represent the image
- 256 x 256
- 128 x 128
- 64 x 64
- 32 x 32
- 16 x 16
- 8 x 8

How many megapixels (approximately) do the photos you take with your camera or phone have?



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

How many pixels?

Width	Height	Pixels	Mega-pixels	Camera
320	240	10.000	0.01	Prototype 1975
1600	1200	1.920.000	2	Nikon Coolpix 950
4032	3024	12.192.768	12	Samsung Galaxy S7 edge
6240	4160	26.000.000	26	Canon EOS 6D M2
8984	6732	60.480.288	60.5	Phase One P65+

Gray level resolution

- The number of gray levels in the image
- 256
- 64
- 16
- 8
- 4
- 2

Image as a matrix

$$\begin{array}{c|cccc}
 & 1 & 2 & 3 & 4 \;\rightarrow c \\
\hline
1 & 23 & 216 & 120 & 55 \\
2 & 4 & 89 & 158 & 130 \\
3 & 65 & 76 & 189 & 34 \\
4 & 19 & 234 & 7 & 45 \\
\downarrow r & & & &
\end{array}$$

- An image is stored in the computer memory as a 2 dimensional matrix
- 4 rows and 4 columns
- Matlab image I – what is I(2,3) ?
- Can also be seen as a discrete function f(r, c)
- In Matlab a pixel is stored as an UINT8!
- UINT8 = Unsigned 8-bit integer = 1 byte

Pixel coordinates – Matlab matrix

r \ c	1	2	3	4
1	23	216	120	55
2	4	89	158	130
3	65	76	189	34
4	19	234	7	45

- Used in Matlab
- Origin is in upper left corner
- 1-based
- (row, column) system
- M rows and N columns

- What is the coordinates of the pixel with value 34?

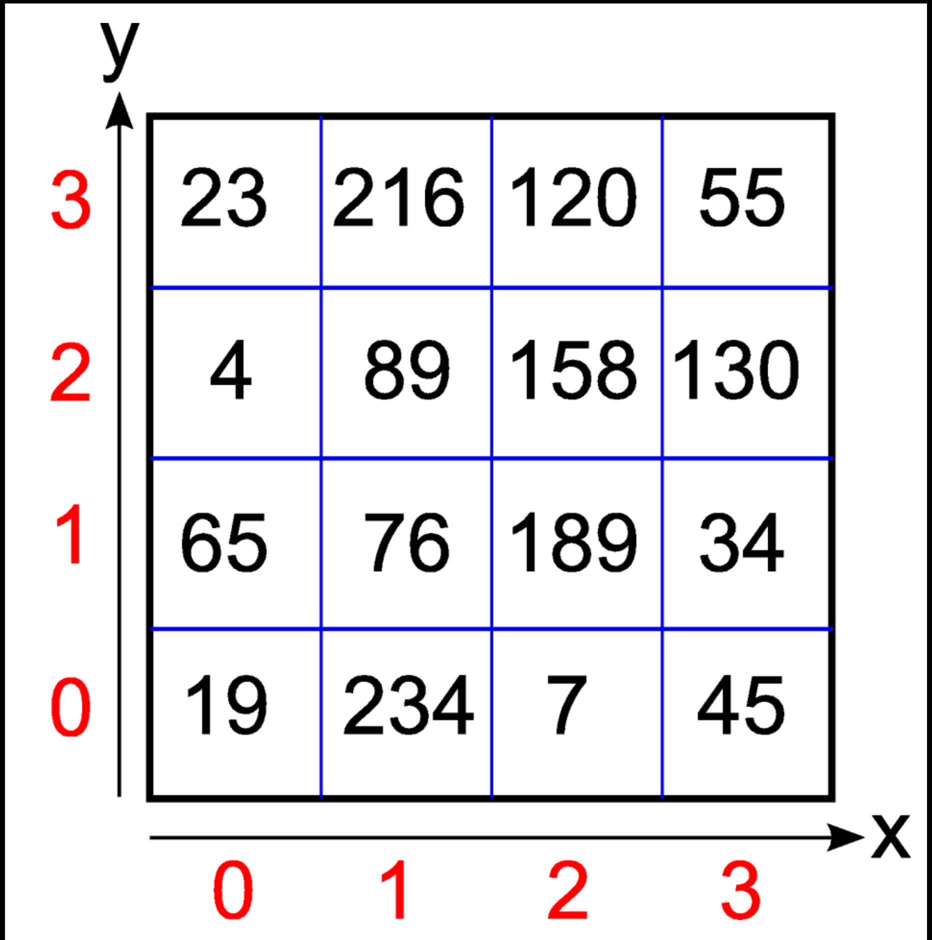
Pixel coordinates – Photoshop etc.

y \ x	0	1	2	3
0	23	216	120	55
1	4	89	158	130
2	65	76	189	34
3	19	234	7	45

- Used in many graphics programs
- Origin in upper left corner
- 0-based
- (X,Y) system

- What is the coordinates of the pixel with value 34?

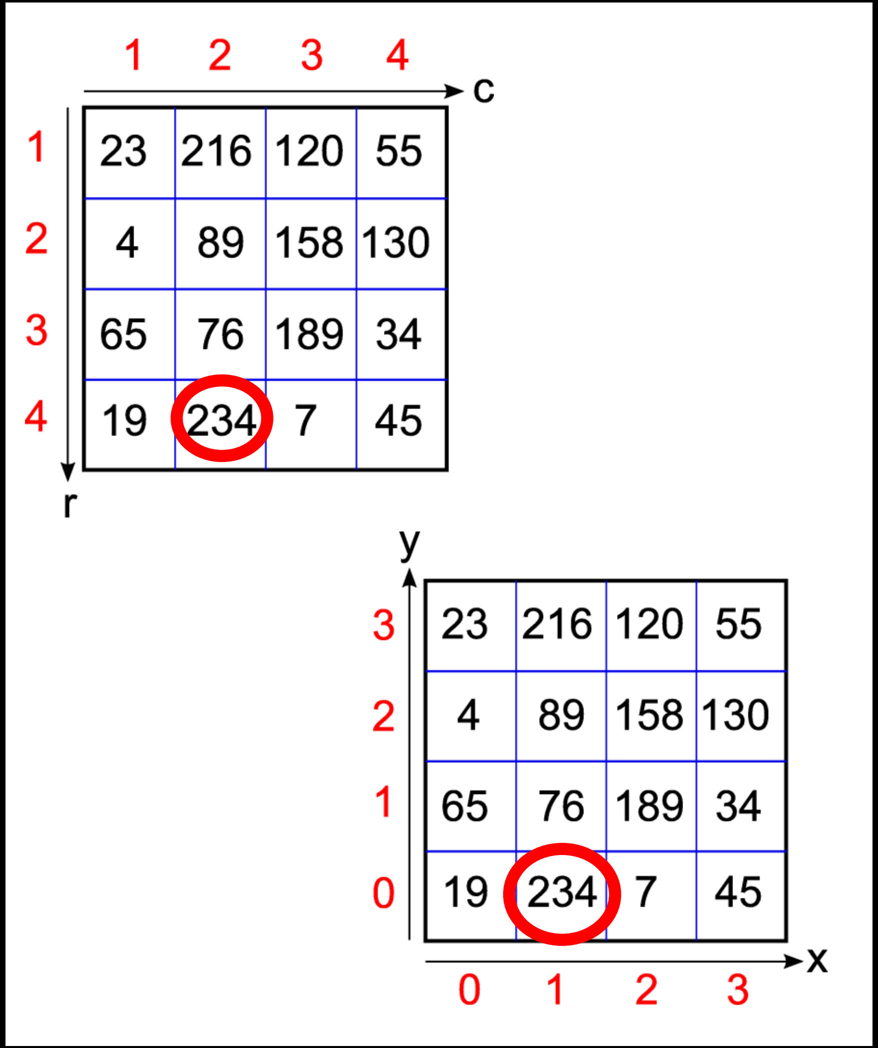
Pixel coordinates – Matlab plots



- Used when plotting – known from mathematics
- Origin in lower left corner
- 0-based
- (X,Y) system

- What is the coordinates of the pixel with value 34?

Why should I care?



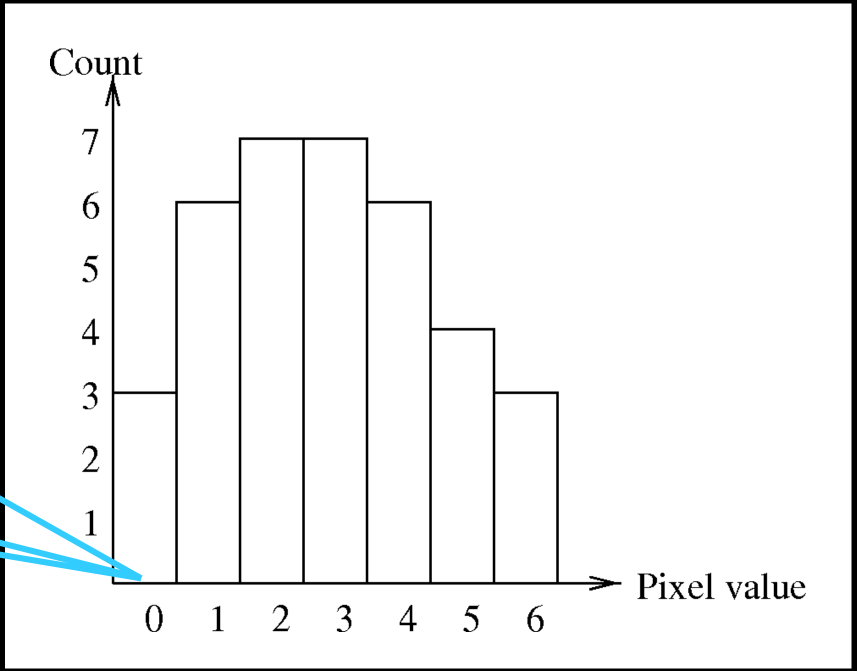
- You have a Matlab image in the matrix system
- Found the pixel with the maximum value
- Want to plot a red circle on top of it
- Plotting is done in the Matlab plot system
- How is this done in this image?
  - Max = 234 at (r,c) = (4,2)
  - Plot circle at (x,y) = (1,0)
- General conversion
  - x = c-1
  - y = M-r

The Image Histogram

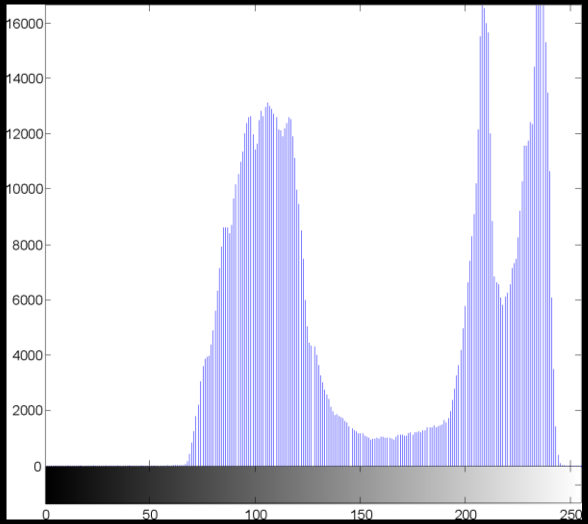
- A histogram normally contains the same number of "bins" as the possible pixel values
- A bin stores the number of pixel with that value

0	2	6	6	3	3
1	4	3	4	4	4
3	2	5	1	5	2
1	4	2	1	3	1
2	5	3	0	2	0
4	2	5	6	3	1

3



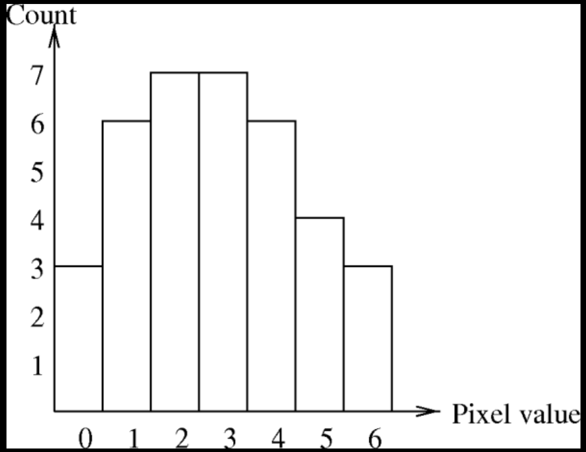
A real grayscale image histogram



- 256 gray levels in the image = 256 bins in the histogram
- The shape of the histogram tells us something about the image
- Can you “recognise” the flower in the histogram?
- What “colors” are missing?

The histogram function

0	2	6	6	3	3
1	4	3	4	4	4
3	2	5	1	5	2
1	4	2	1	3	1
2	5	3	0	2	0
4	2	5	6	3	1

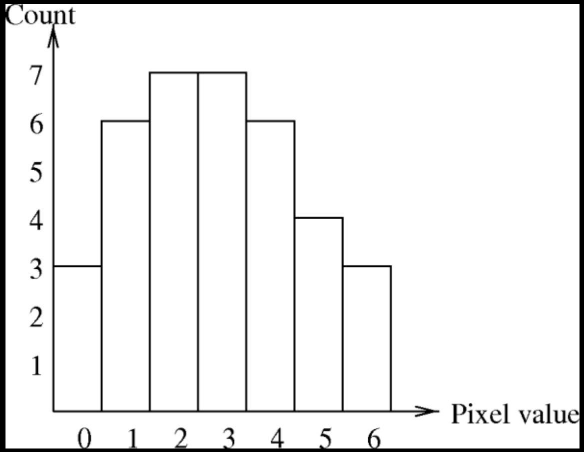


- Can be seen as a function h(v)
- v is the pixel value
- h(2) = 7
- h(5) = 4

- Total number of pixels is the sum of all h

Pixel value statistics

0	2	6	6	3	3
1	4	3	4	4	4
3	2	5	1	5	2
1	4	2	1	3	1
2	5	3	0	2	0
4	2	5	6	3	1



- Pick a random pixel in the image
- What is the probability of it having value 3? P(v=3)

- h(3) = 7
- Np = 36
- P(v=3) = 7/36 * 100%

A random pixel is chosen in the image. What is the probability that the value of the pixel is 3?

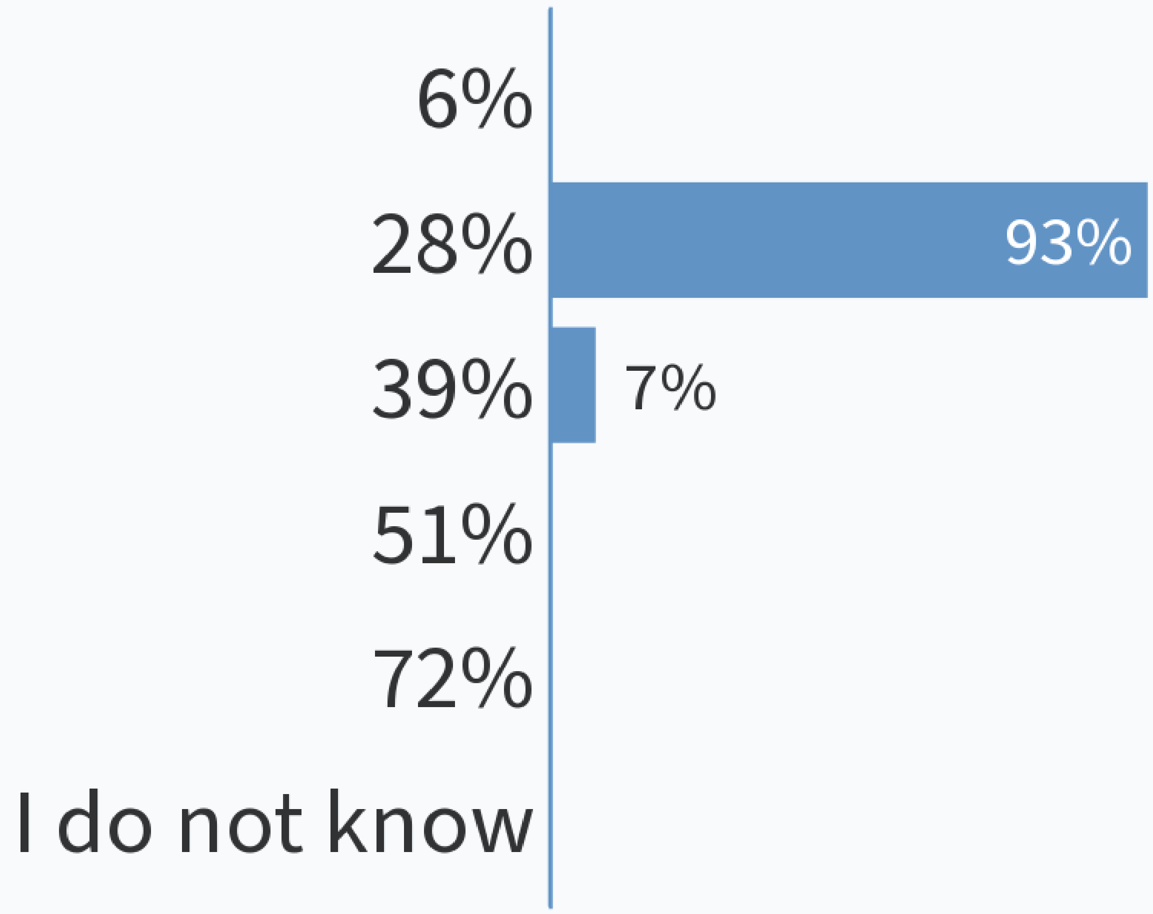
2	5	4	0	6	3
3	3	1	2	3	5
0	0	1	3	2	3
2	3	2	5	5	3
0	0	3	2	5	2
3	2	4	5	1	1

- 6%
- 28%
- 39%
- 51%
- 72%
- I do not know

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

A random pixel is chosen in the image. What is the probability that the value of the pixel is 3?

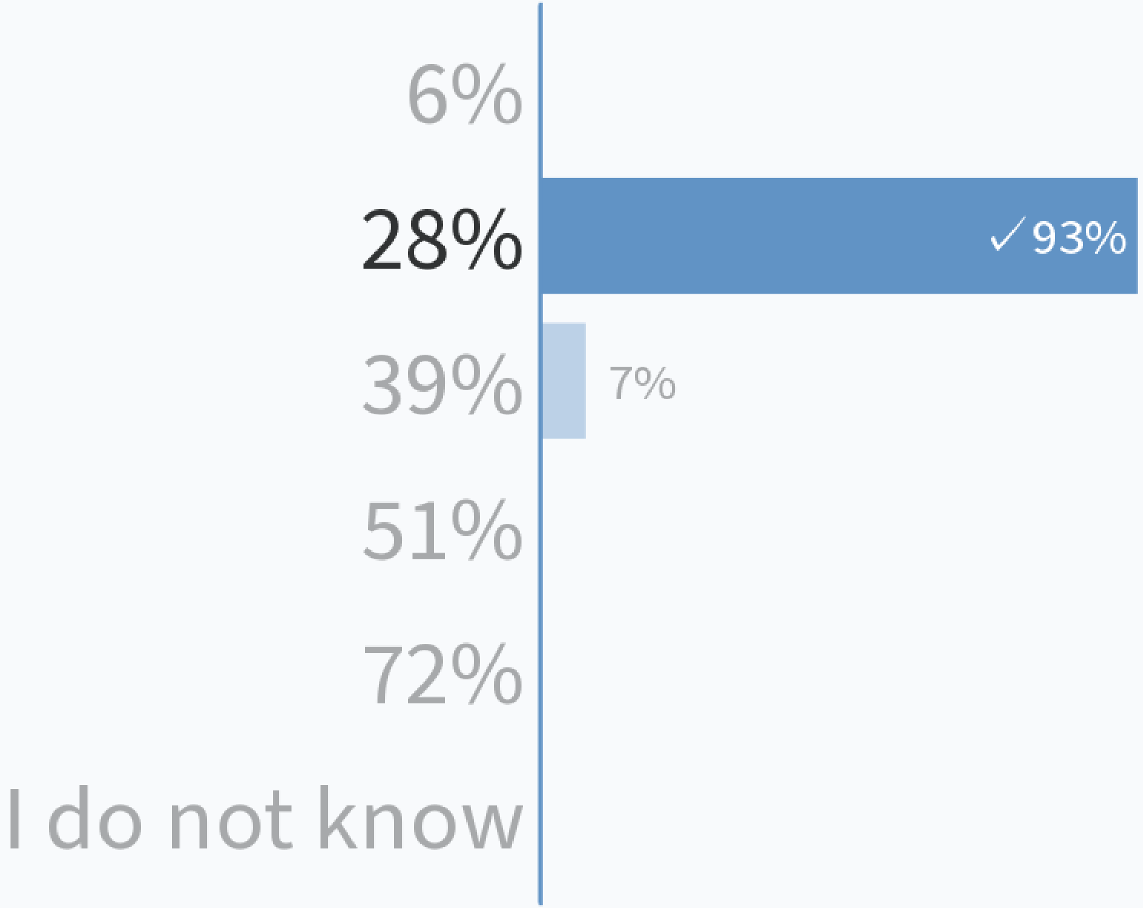
2	5	4	0	6	3
3	3	1	2	3	5
0	0	1	3	2	3
2	3	2	5	5	3
0	0	3	2	5	2
3	2	4	5	1	1



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

A random pixel is chosen in the image. What is the probability that the value of the pixel is 3?

2	5	4	0	6	3
3	3	1	2	3	5
0	0	1	3	2	3
2	3	2	5	5	3
0	0	3	2	5	2
3	2	4	5	1	1



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Normalised histogram

- A normalised histogram is made by dividing each bin count with the total number of pixels
- H(v) is the normalised histogram function
- H(v) is the probability that a random pixel has value v

- Equal to a probability density function

Other Image Types

- Colour images
- Binary Images
- Label Images
- 16-bit images

Colour images

- RGB = Red, Green, and Blue
- Television, computers, digital cameras use the "RGB color space"
- Additive colours: Final colour is made by mixing red, green, and blue
- Typically the values of R, G, and B lie between 0 and 255 (total 3 bytes)!

RGB Colours

RGB = (0,0,0)

RGB = (255,255,255)

- When alle three "Lamps" are turned of we get black
- When all three "lamps" are on what do we get?

(255,0,0)

(0,255,0)

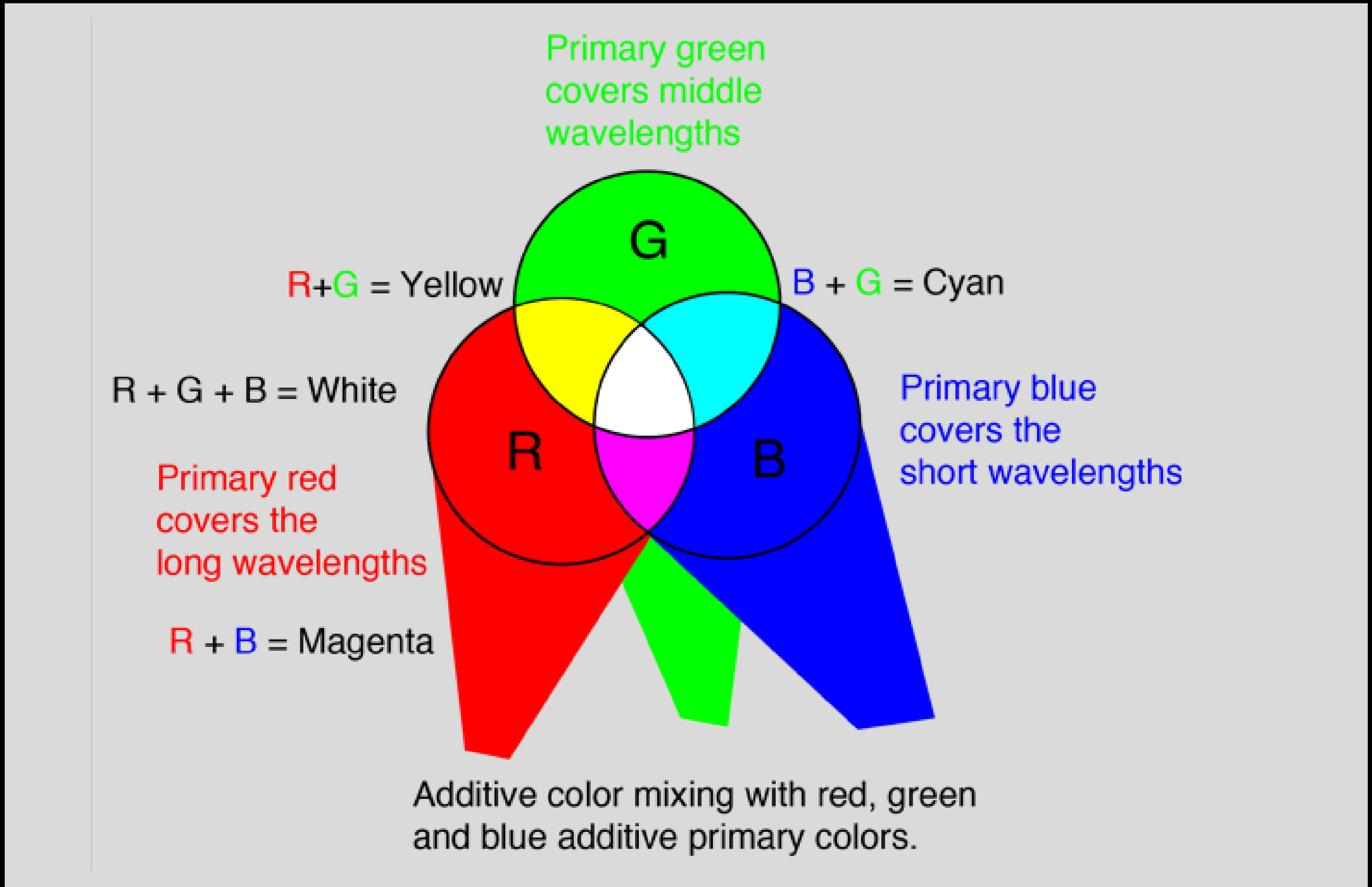
(0,0,255)

(255,255,0)

(0,255,255)

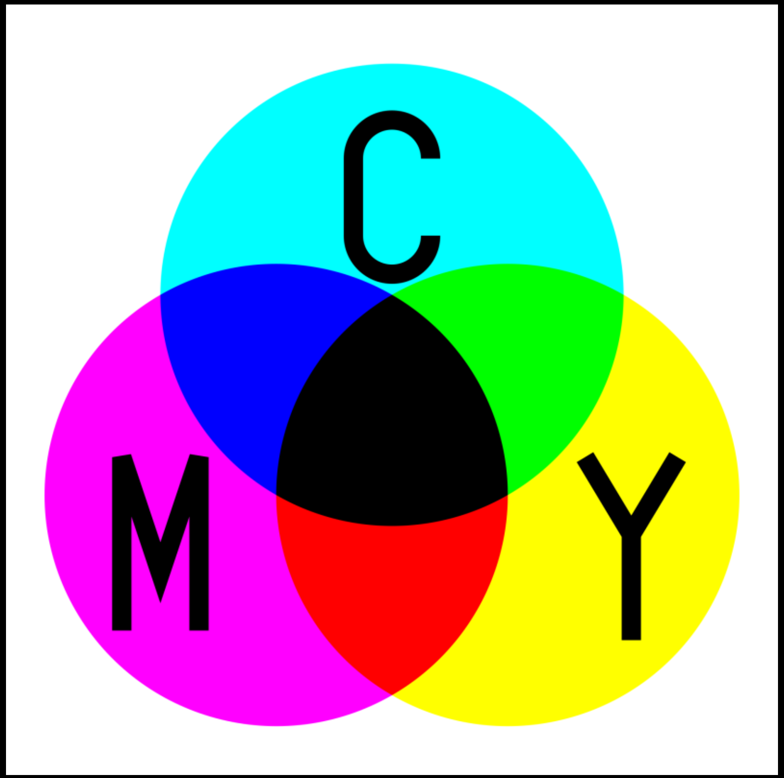
(255,0,255)

Additive color mixing



http://hyperphysics.phy-astr.gsu.edu/hbase/vision/addcol.html

Subtractive color mixing



Wikipedia

Processing RGB images

- Each pixel in a colour image contains 3 values
- Equal to a “vector function” in mathematics
- More complicated to analyse
- Medical images are typically grayscale
  - Why?
- Often images are converted from colours to grayscale before the analysis

Converting colour to grayscale

$$v = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

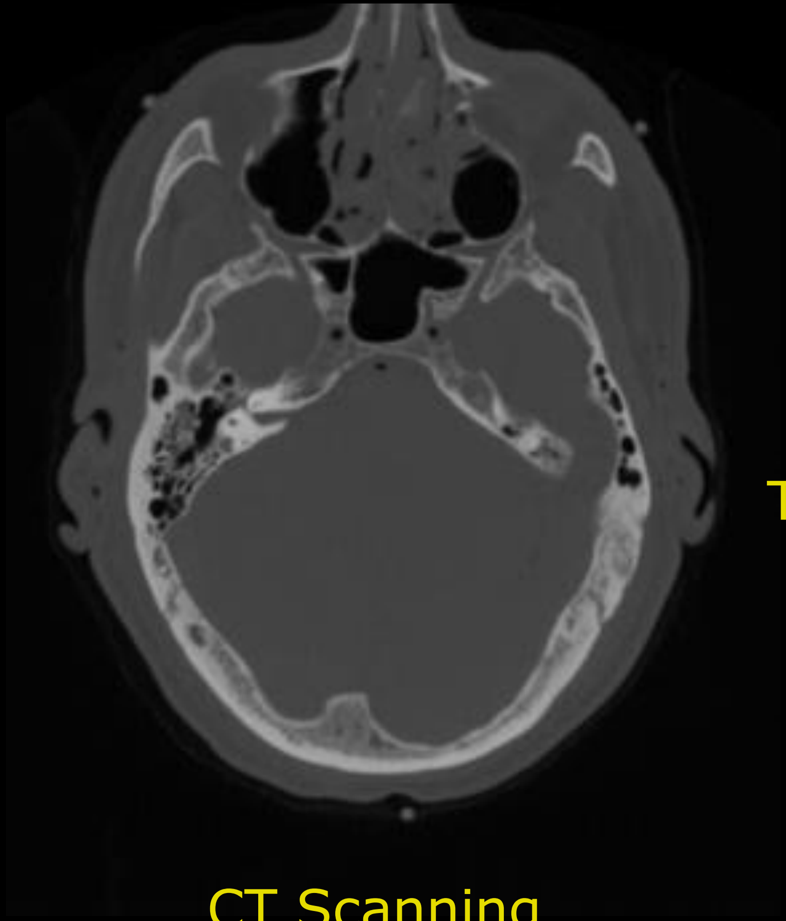
Is it possible to convert a grayscale image back to a color image?

Binary images

- Binary – means on or off
- Binary image – only two colors
- Background (0 = black)
- Foreground (1 = white)

- Simple representation of CT scanning of the head

Gray scale to Binary Image

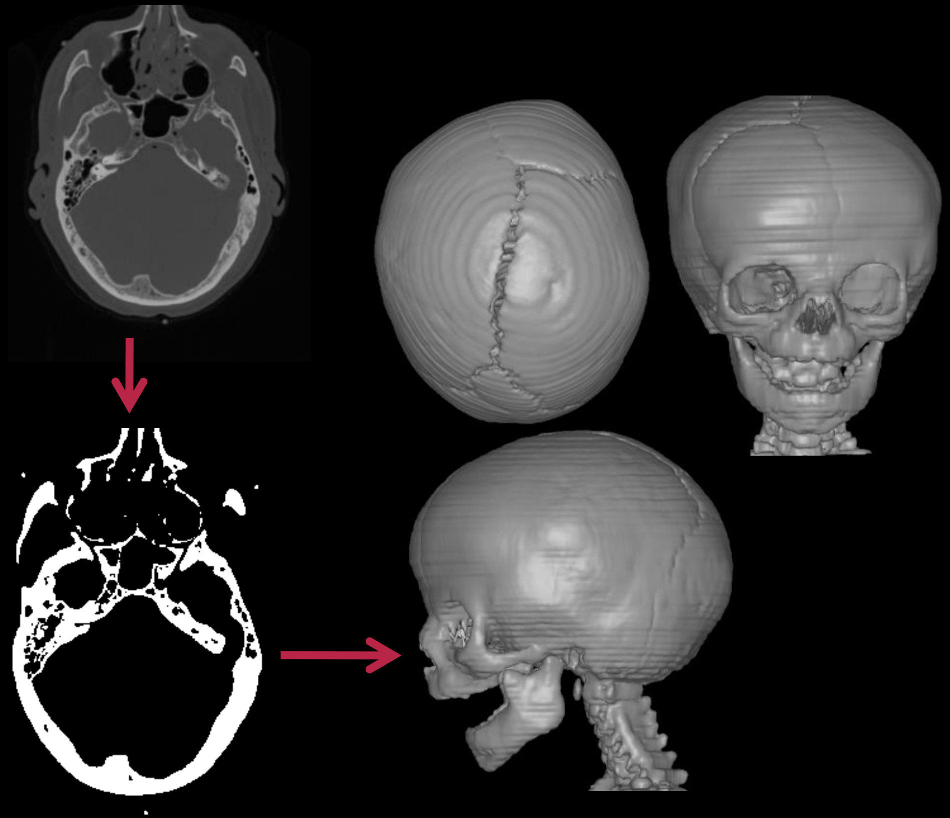


CT Scanning

Threshold

“Bone Image”

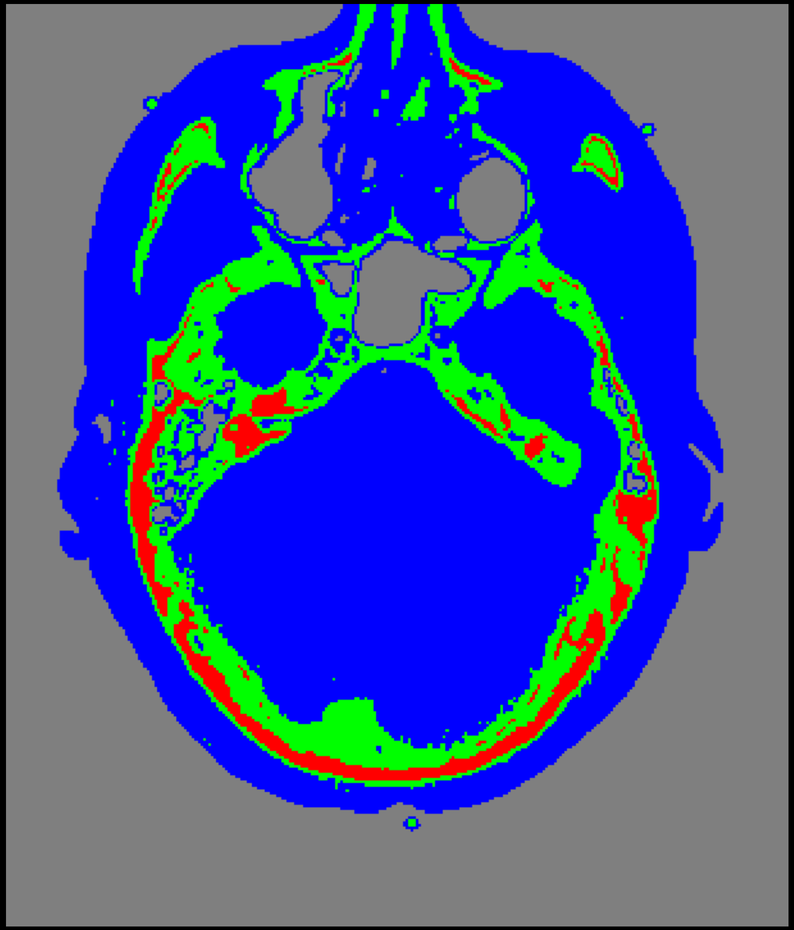
Binary image – why?



- Separating objects from background
- Count the number of the objects
- Measure the size and shape of objects
- Advanced 3D visualisations

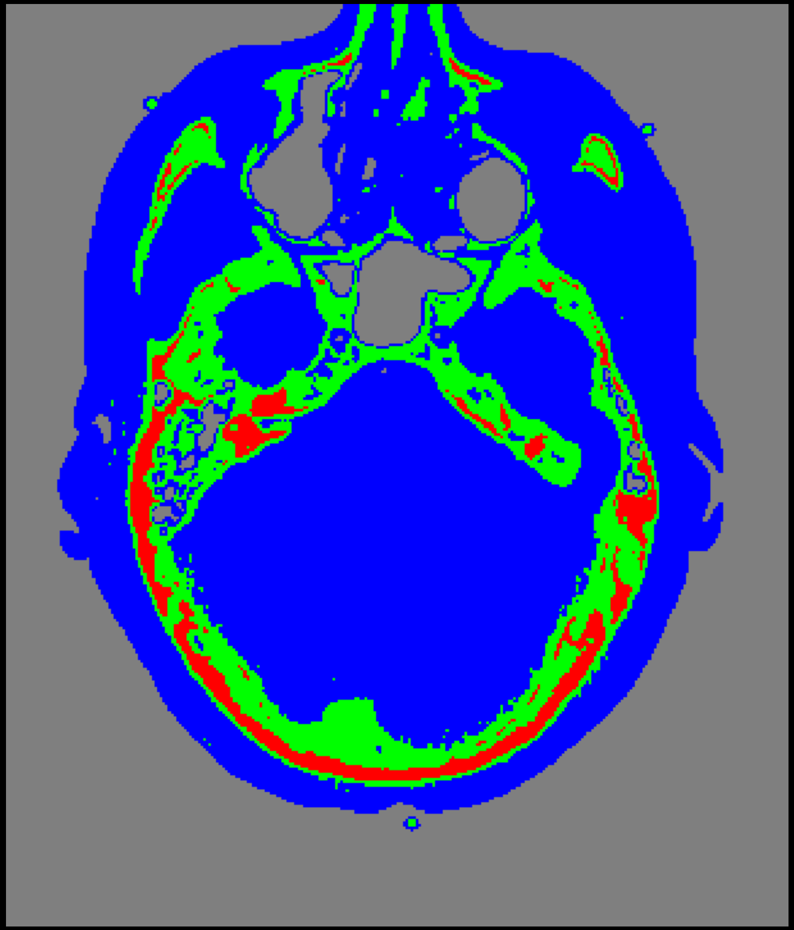
Image from 3D laboratory

Label Image



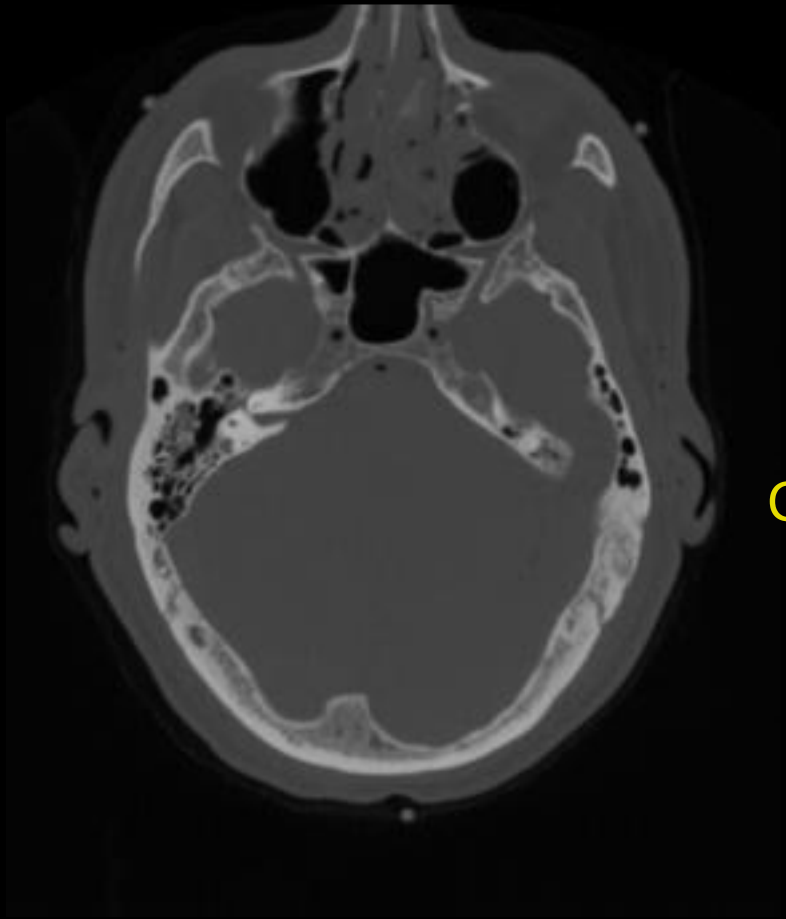
- The pixel value tells the *type* of the pixel
  - (0) Gray – background
  - (1) Blue – soft tissue
  - (2) Green – hard bone
  - (3) Red – spongy bone
- Only 4 different pixel values
- Colours made using a *look-up-table*

Label Image –why?

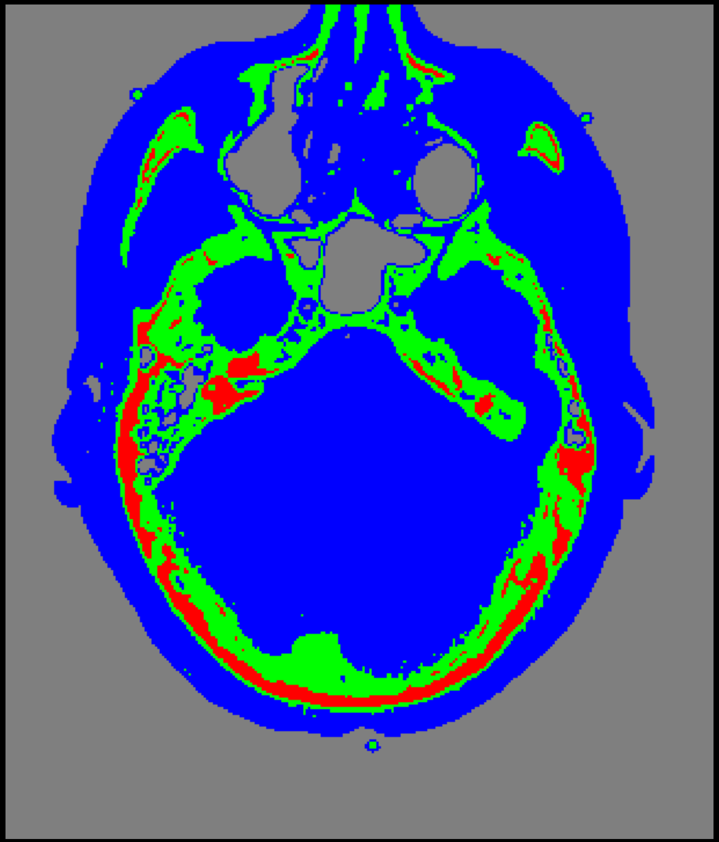


- How big is a tumour? (volume / percent)
- Bone density
- General anatomy recognition
  - Blood vessels
  - Calcifications

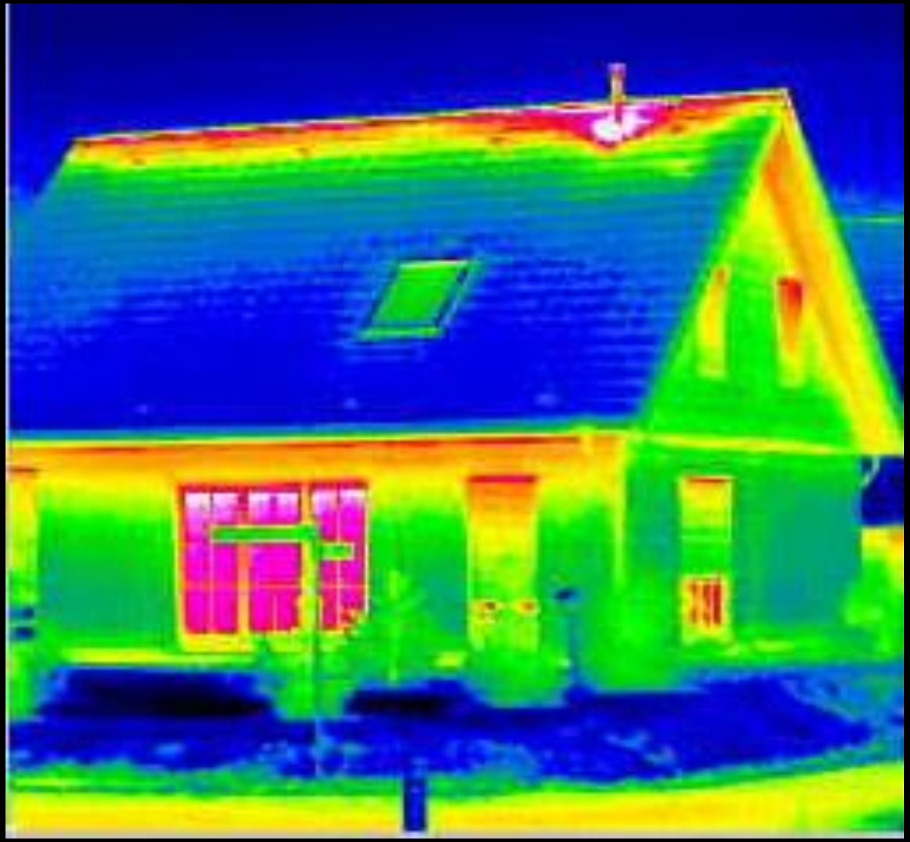
Label Image – how?



Classification



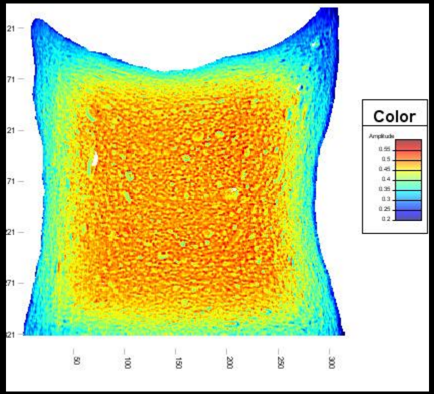
Multispectral images

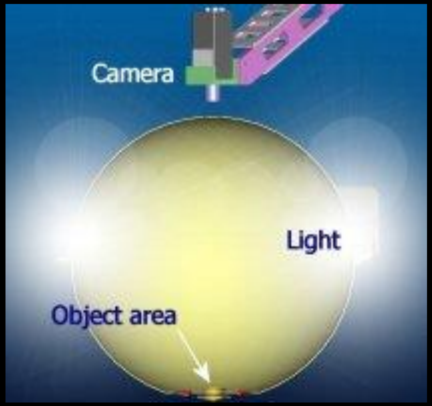


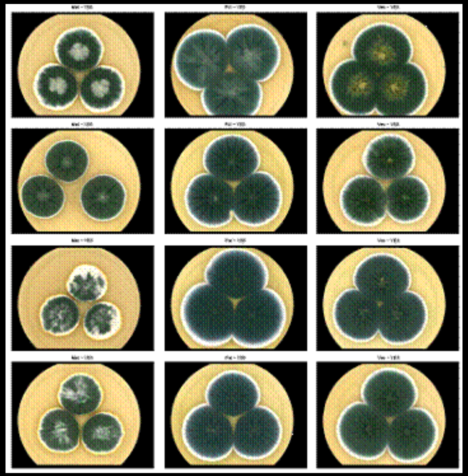
Infrared

- There are more visual information than what can be seen with the human eye
- Standard cameras captures the red, green, blue colours
- Capture systems that capture more bands and other frequencies exist
- Creates multispectral images
  - Each pixel contains perhaps 20 values from different spectral bands

Multispectral System - VideometerLab







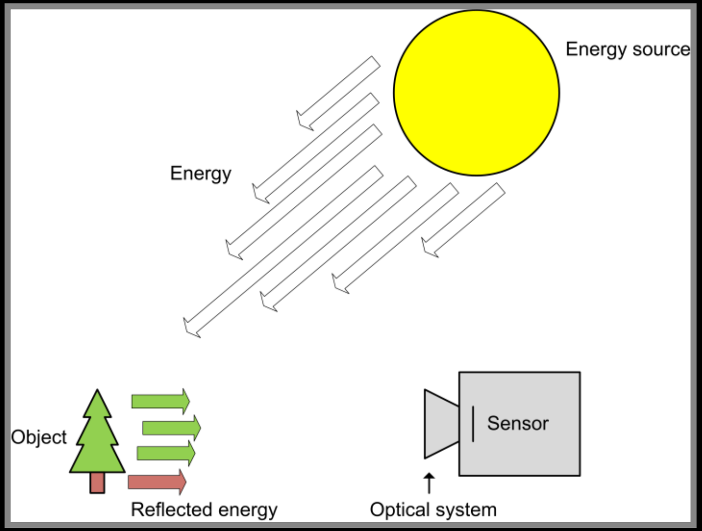
- Integrating sphere
- Light emitting diodes with different wavelengths
  - From near infrared to ultraviolet
- High resolution camera
- Water in bread
- Classification of fungi
- Skin diseases

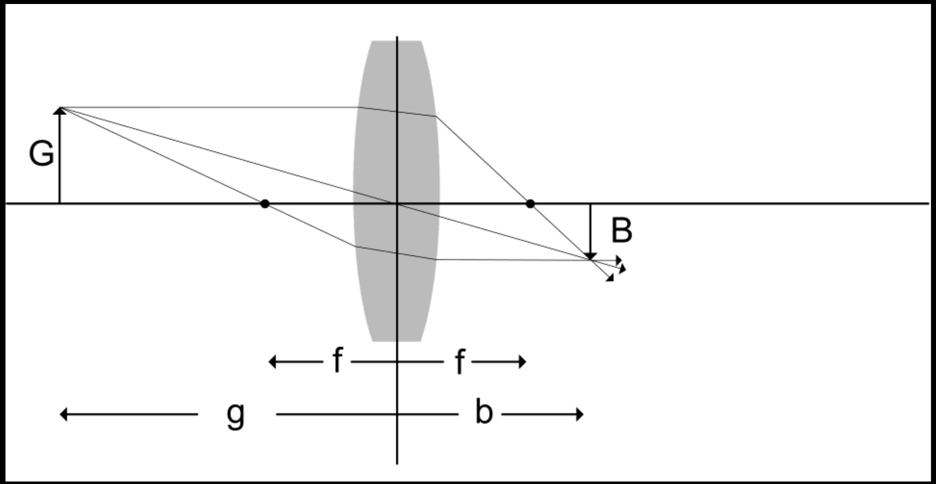
16-bit images

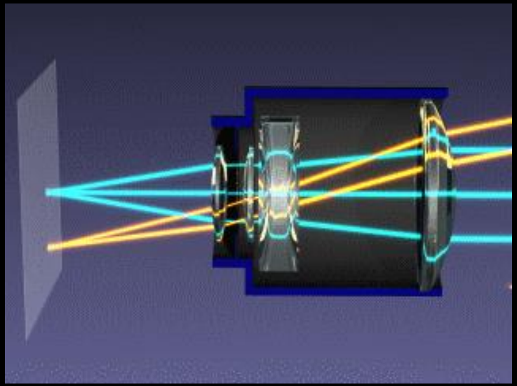
- 256 values fine for the human eye
- Pixel values not only for display
  - Physical meaning
- Computed Tomography
  - X-ray attenuation
- Hounsfield units
  - 0 water
  - -1000 air
  - -120 fat
  - 400+ bone

PCA Analysis

Next week:
Image acquisition, digital cameras, compression and storage







A tutorial on principal component analysis

Rasmus R. Paulsen

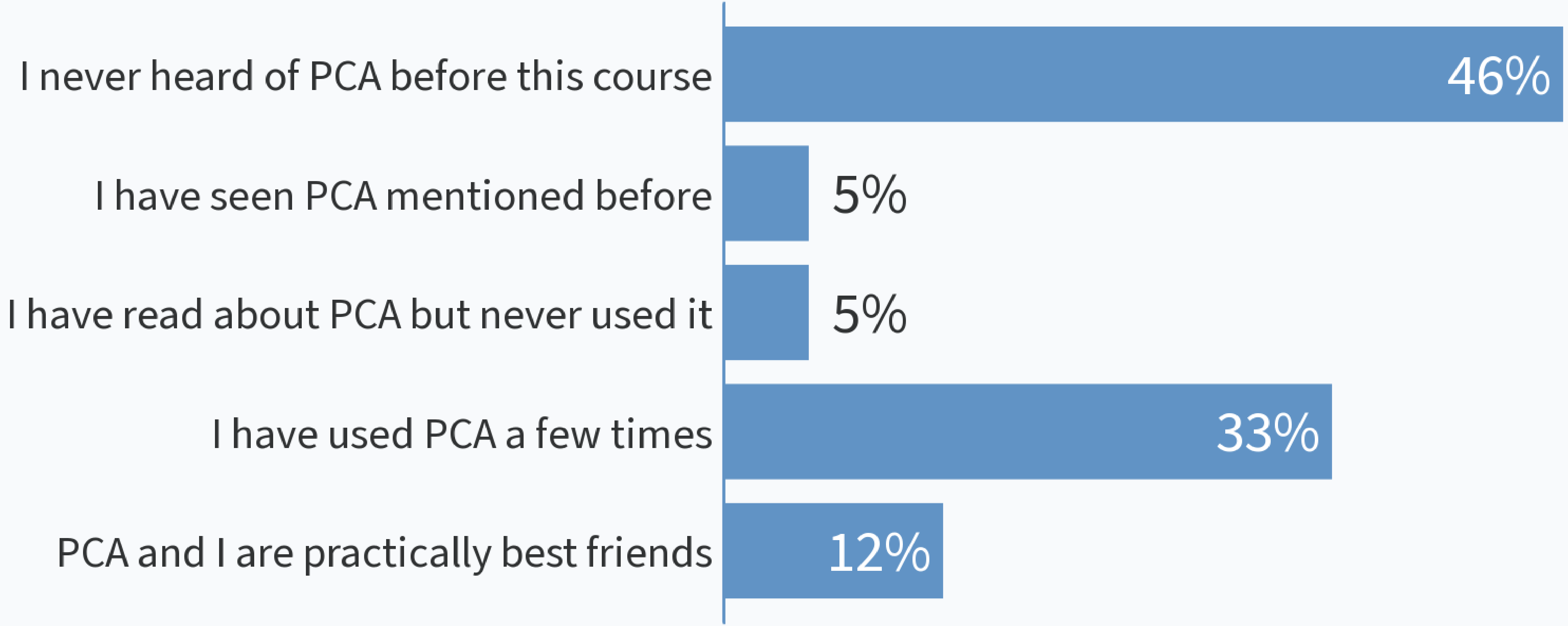
DTU Compute

Based on

Jonathan Shlens: A tutorial on Principal Component Analysis (version 3.02 – April 7, 2014)

http://compute.dtu.dk/courses/02502

What is your experience with Principal Component Analysis (PCA)



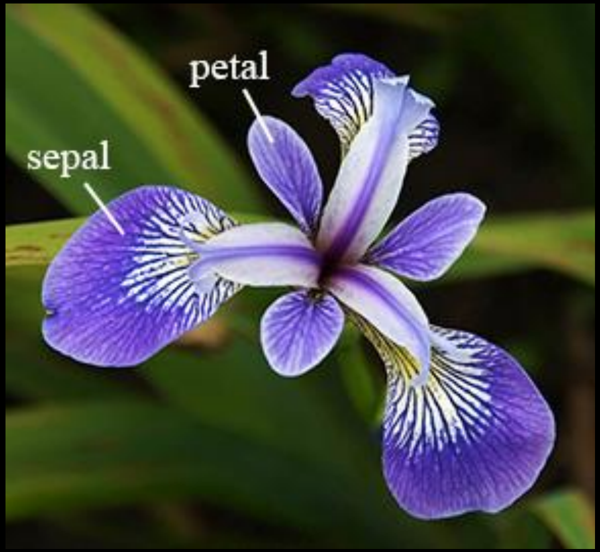
Principal Component Analysis (PCA) learning objectives

- Describe the concept of principal component analysis
- Explain why principal component analysis can be beneficial when there is high data redundancy
- Arrange a set of multivariate measurements into a matrix that is suitable for PCA analysis
- Compute the covariance of two sets of measurements
- Compute the covariance matrix from a set of multivariate measurements
- Compute the principal components of a data set using Eigenvector decomposition
- Describe how much of the total variation in the data set that is explained by each principal component

Iris data

The **Iris flower data**
set or Fisher's Iris data set is a data set introduced by Ronald Fisher in his 1936 paper *The use of multiple measurements in taxonomic problems*

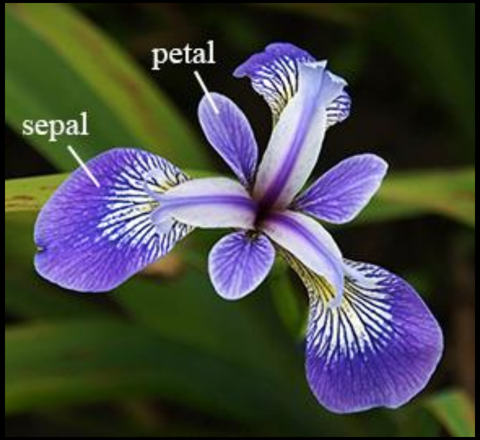
Iris data



- 3 Iris types
  - 50 flowers of each type
- For each flower
  - Sepal length
  - Sepal width
  - Petal length
  - Petal width
- We use one type as example
  - 50 measured flowers

Iris Data Matrix

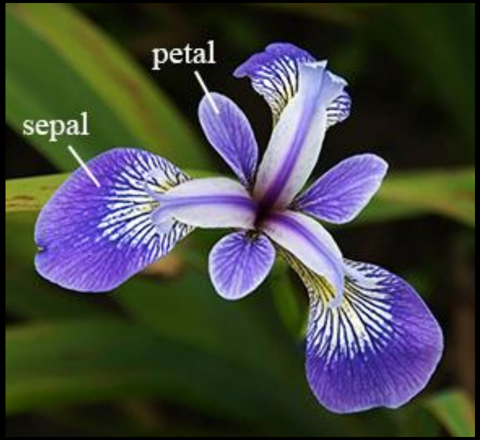
- One column is one flower
- One row is all measurements of one type



$$\mathbf{X} = \begin{array}{c} \begin{array}{ccc} 1 & \qquad\qquad & 50 \end{array} \\ \begin{bmatrix} \text{Sepal length}_1 & \cdots & \text{Sepal length}_{50} \\ \text{Sepal width}_1 & \cdots & \text{Sepal width}_{50} \\ \text{Petal length}_1 & \cdots & \text{Petal length}_{50} \\ \text{Petal width}_1 & \ldots & \text{Petal width}_{50} \end{bmatrix} \end{array}$$

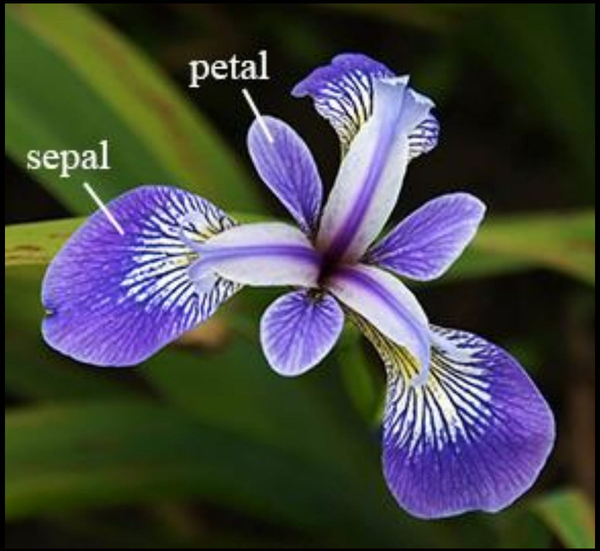
What can we use these data for?

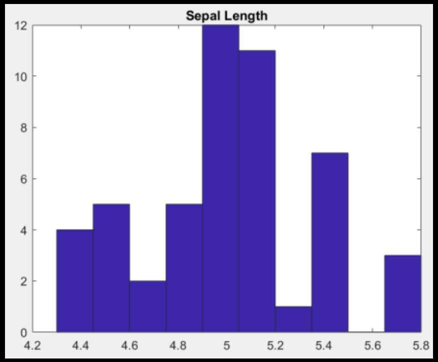


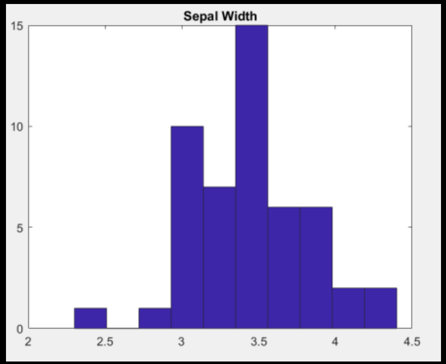
- The measurements can be used to:
  - Recognize a species of flowers
  - Classify flowers into groups
  - Describe the characteristics of the flower
  - Quantify growth rates
  - …
- Do we need all the measurements?
  - Can we *boil down* or *combine* some measurements?
- Are some measurements *redundant?*

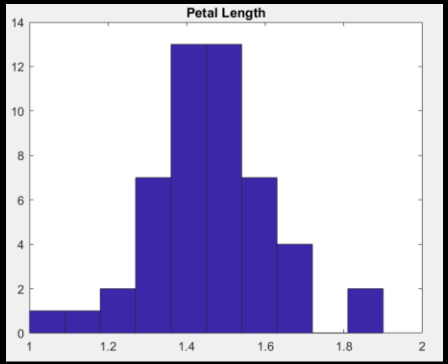
Variance

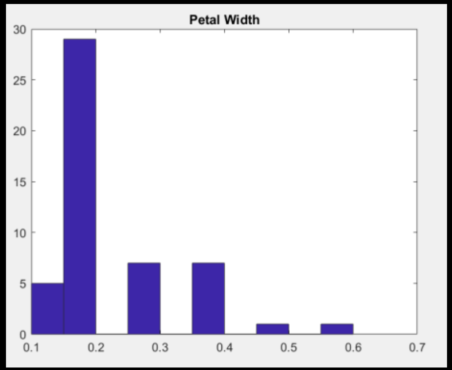
$$\begin{aligned}\sigma^2_{SL} &= 0.1242\\ \sigma^2_{SW} &= 0.1437\\ \sigma^2_{PL} &= 0.0302\\ \sigma^2_{PW} &= 0.0111\end{aligned}$$



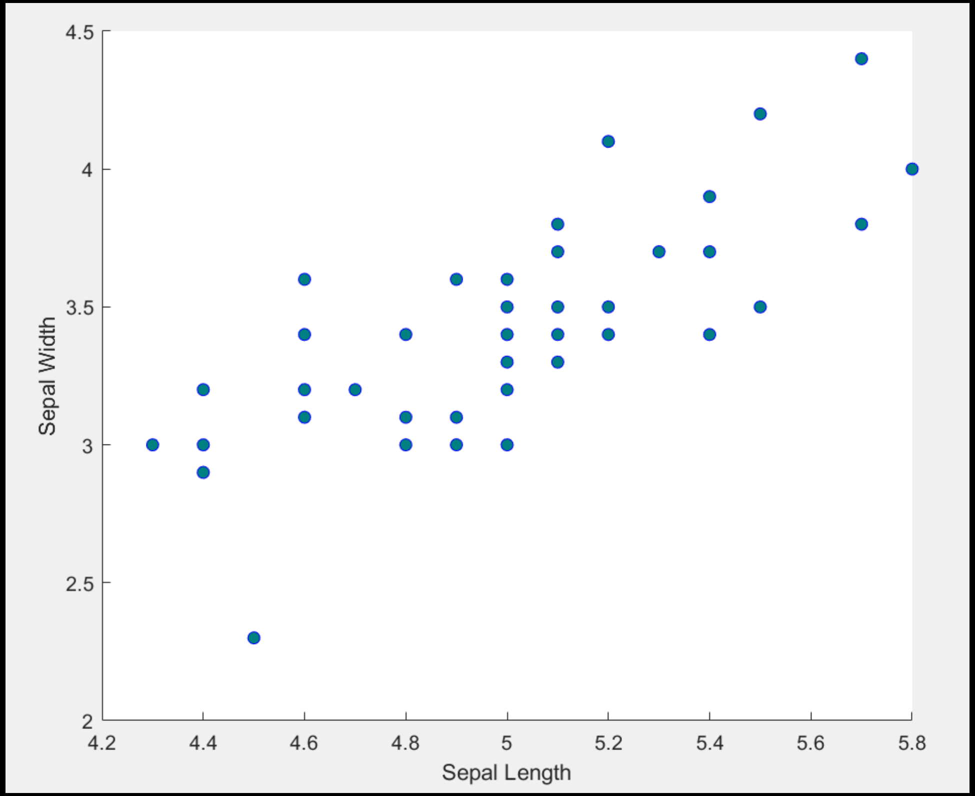




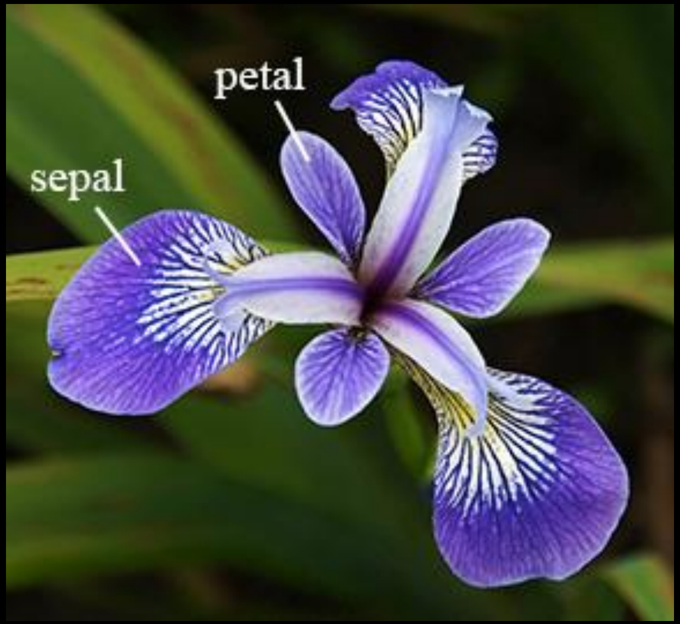




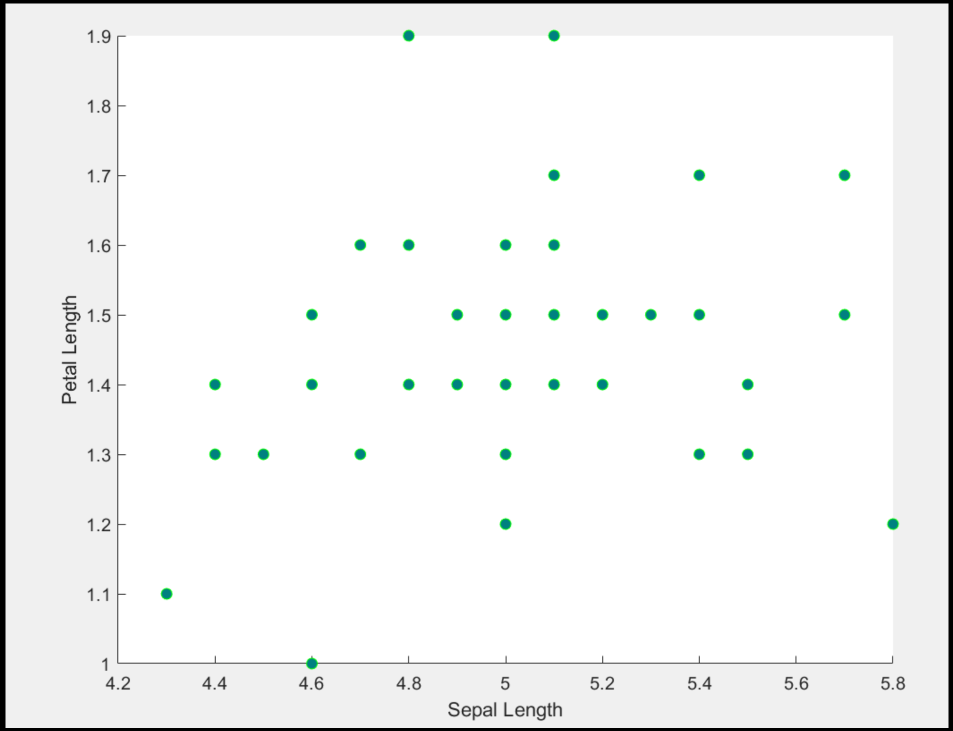
High Redundancy



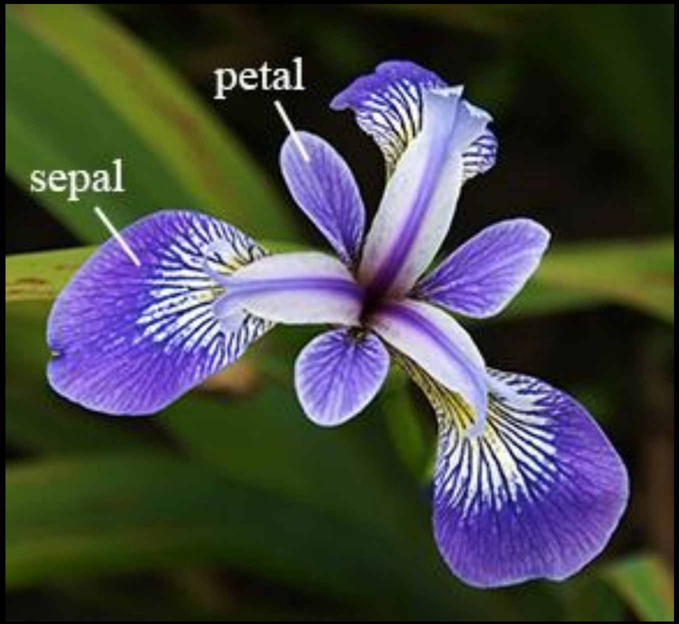
Observation: We can explain quite a lot of the sepal width if we know the sepal lengths



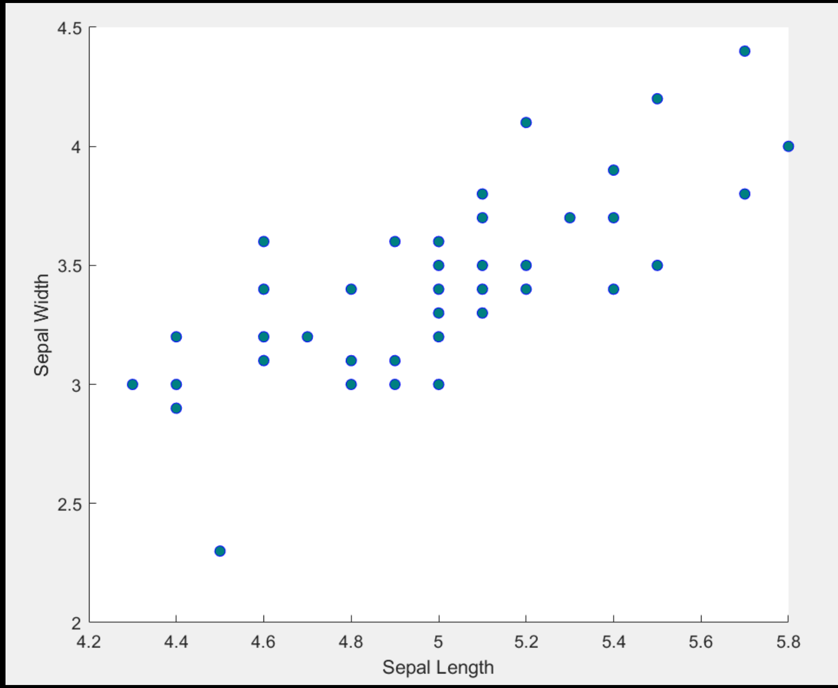
Low Redundancy

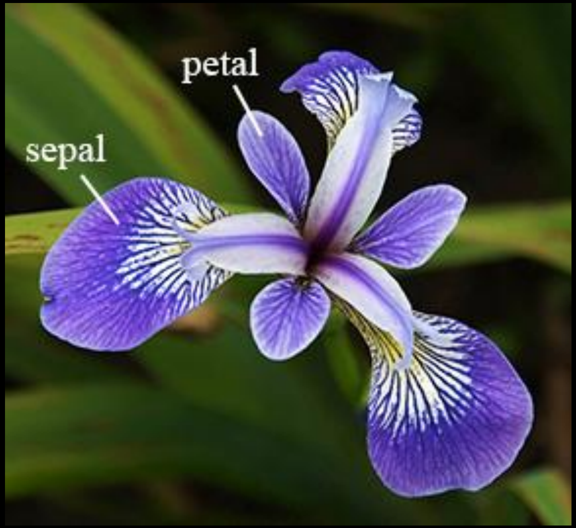


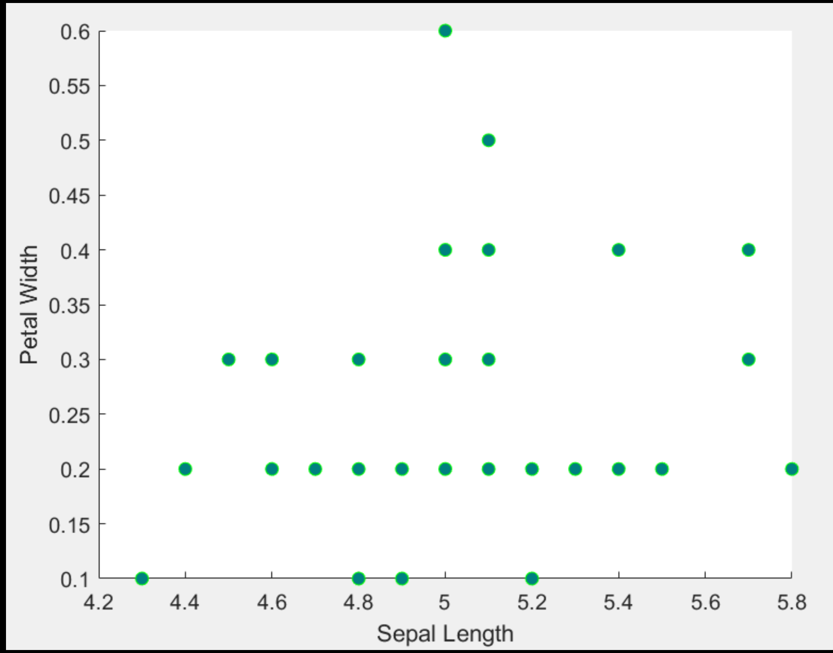
Observation: We can NOT explain the petal length if we know the sepal lengths



Covariance

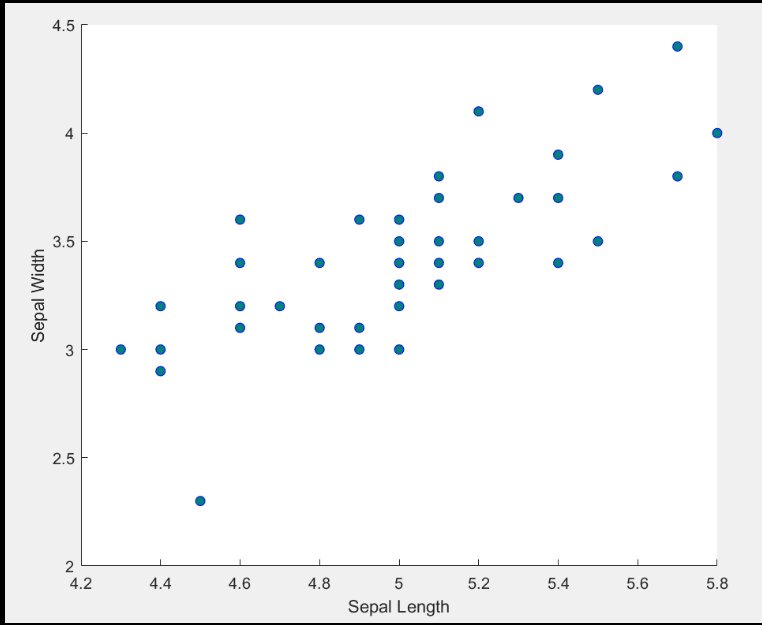


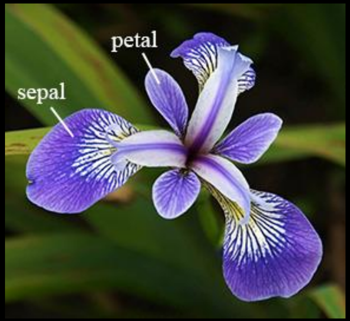




Covariance measures the *relationship* between measurements

High Covariance





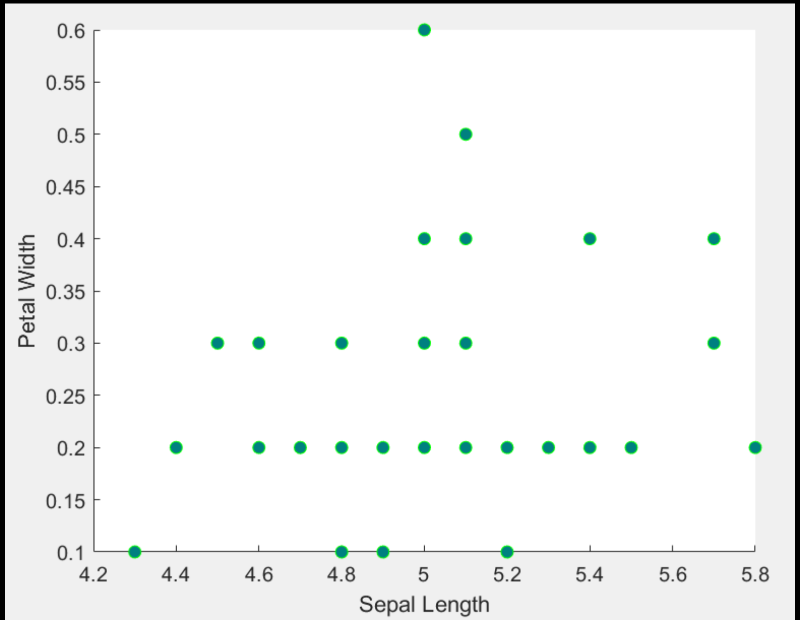
Sepal length and sepal width

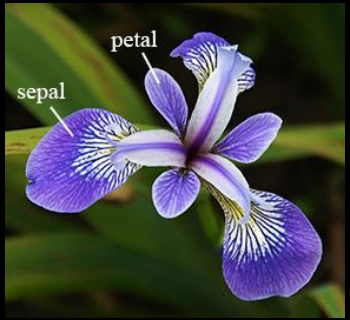
$$a_i = \mathrm{SL} = \{5.1, 4.9 \ldots, 5\}$$

$$\mathrm{b_i} = \mathrm{SW} = \{3.5, 3, \ldots, 3.3\}$$

$$\sigma^2_{\mathrm{SL,SW}} = \frac{1}{n}\sum_i a_i b_i = 17.2578$$

Low covariance

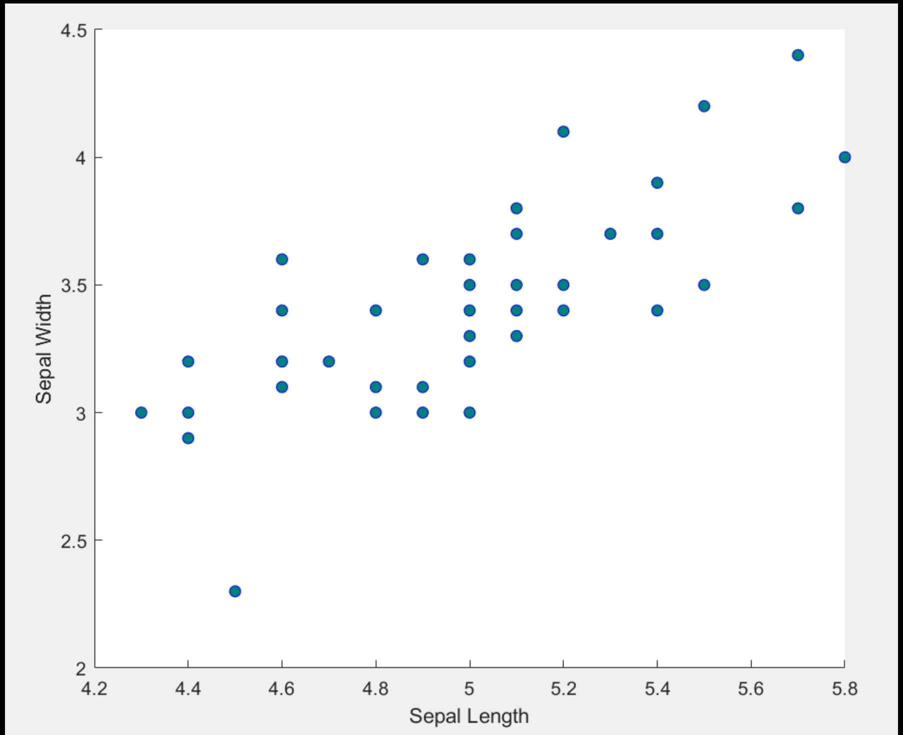




Sepal length and petal width

$$\sigma^2_{\mathrm{SL,PW}} = \frac{1}{n}\sum_i a_i b_i = 1.2416$$

Vector notation for covariance



Sepal length and sepal width

$$\mathbf{a} = \mathrm{SL} = [5.1, 4.9 \ldots, 5]$$

$$\mathbf{b} = \mathrm{SW} = [3.5, 3, \ldots, 3.3]$$

$$\sigma^2_{\mathrm{SL,SW}} = \frac{1}{n}\mathbf{a}\mathbf{b}^T$$

Matrix notation for covariance

m x n matrix (m=4 and n=50)

$$\mathbf{X} = \begin{bmatrix} \text{Sepal length}_1 & \cdots & \text{Sepal length}_{50} \\ \text{Sepal width}_1 & \cdots & \text{Sepal width}_{50} \\ \text{Petal length}_1 & \cdots & \text{Petal length}_{50} \\ \text{Petal width}_1 & \ldots & \text{Petal width}_{50} \end{bmatrix}$$

$$\mathbf{C_X} \equiv \frac{1}{n}\mathbf{X}\mathbf{X}^T$$

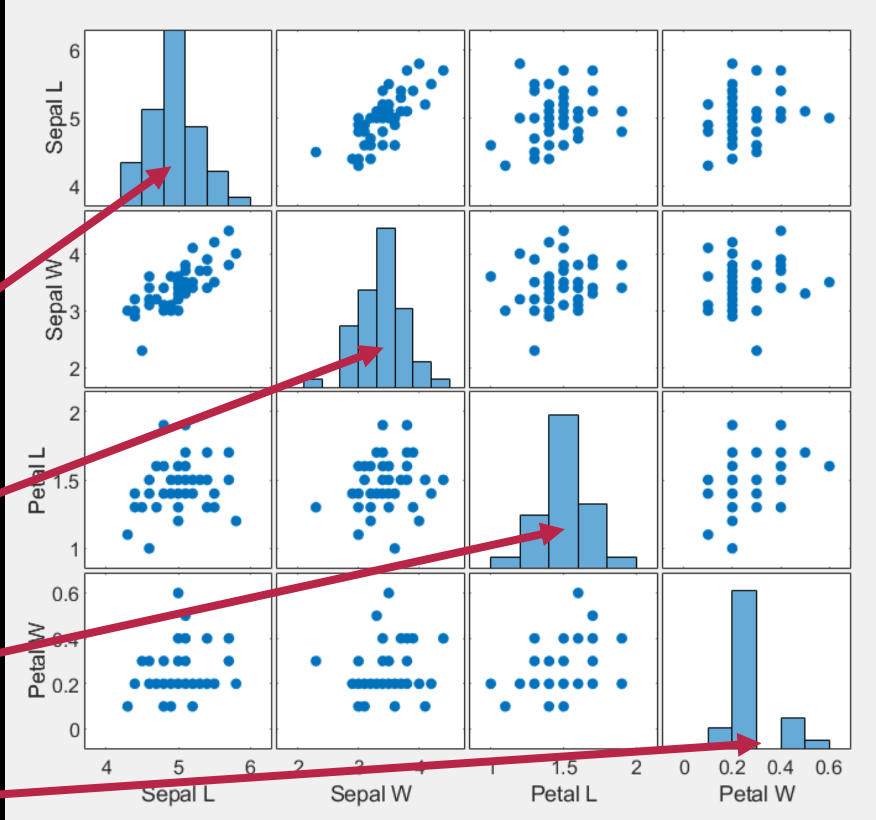
m x m square matrix (m=4)

Covariance matrix autopsy

$$\mathbf{C_x} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$$

The diagonal elements are the variances

$$\begin{aligned} \sigma^2_{SL} &= 0.1242 \\ \sigma^2_{SW} &= 0.1437 \\ \sigma^2_{PL} &= 0.0302 \\ \sigma^2_{PW} &= 0.0111 \end{aligned}$$



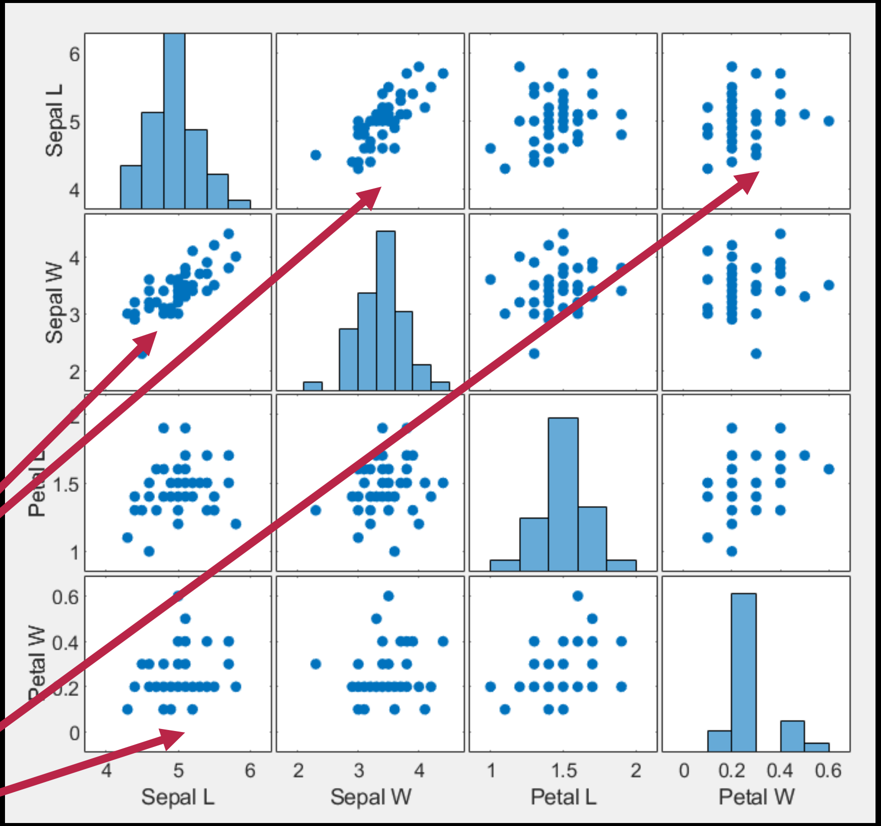
Covariance matrix autopsy II

$$\mathbf{C_x} \equiv \frac{1}{n}\mathbf{X}\mathbf{X}^T$$

The off-diagonal elements are the covariance

$$\sigma^2_{\mathrm{SL,SW}} = \frac{1}{n}\sum_i a_i b_i = 17.2578$$

$$\sigma^2_{\mathrm{SL,PW}} = \frac{1}{n}\sum_i a_i b_i = 1.2416$$

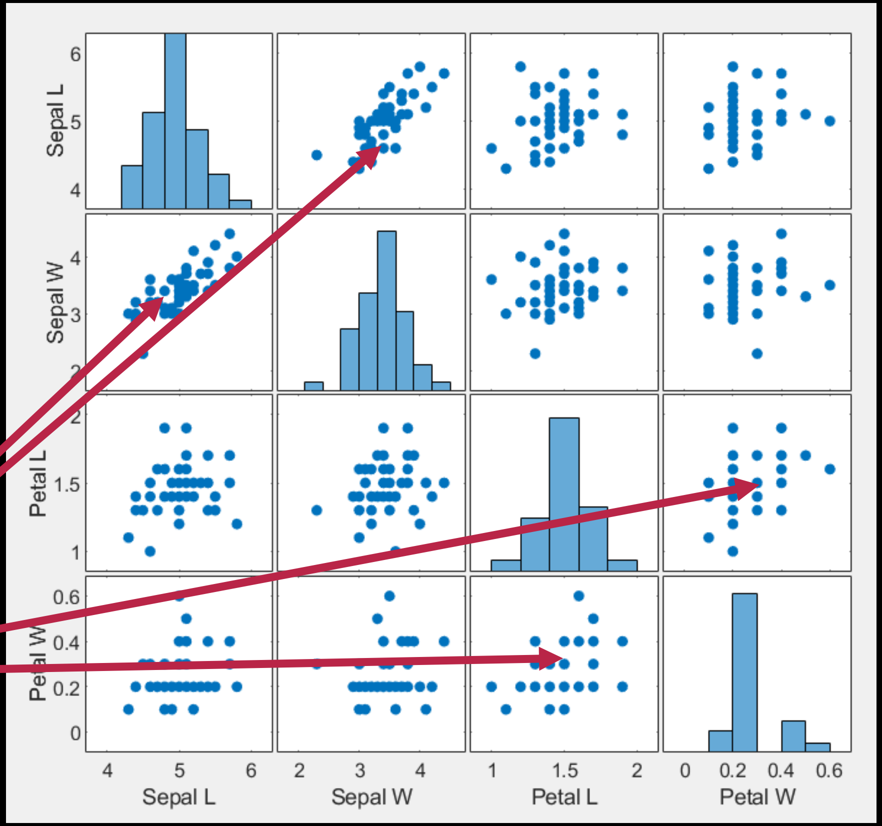


Symmetric!

Covariance matrix autopsy III

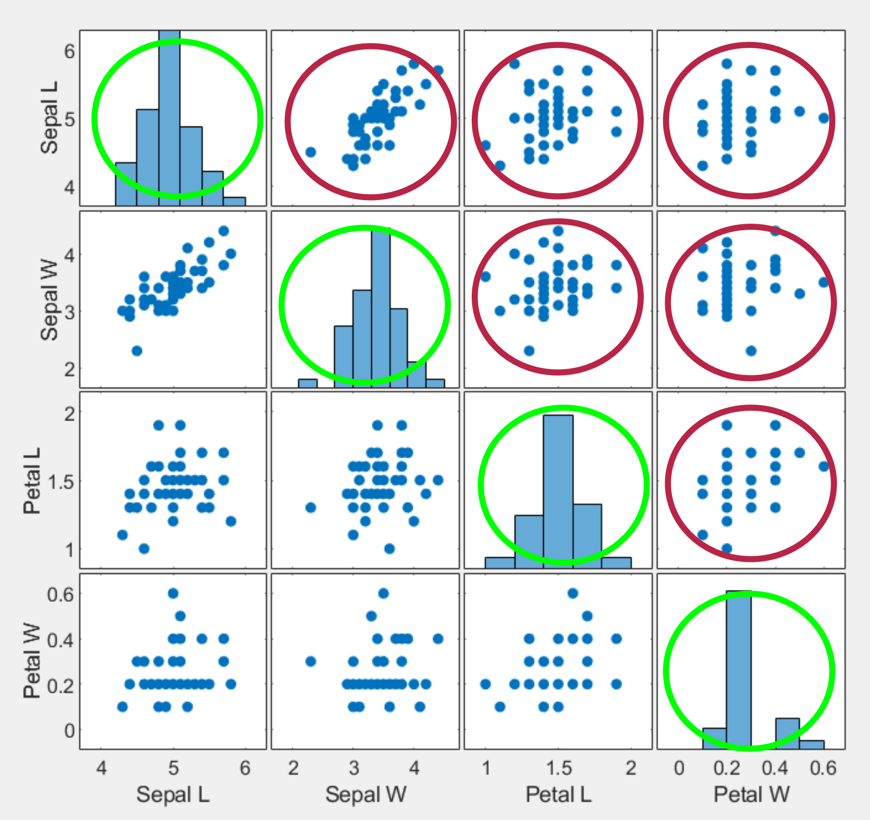
$$\mathbf{C_X} \equiv \frac{1}{n}\mathbf{X}\mathbf{X}^T$$

High redundancy



Symmetric!

Goals

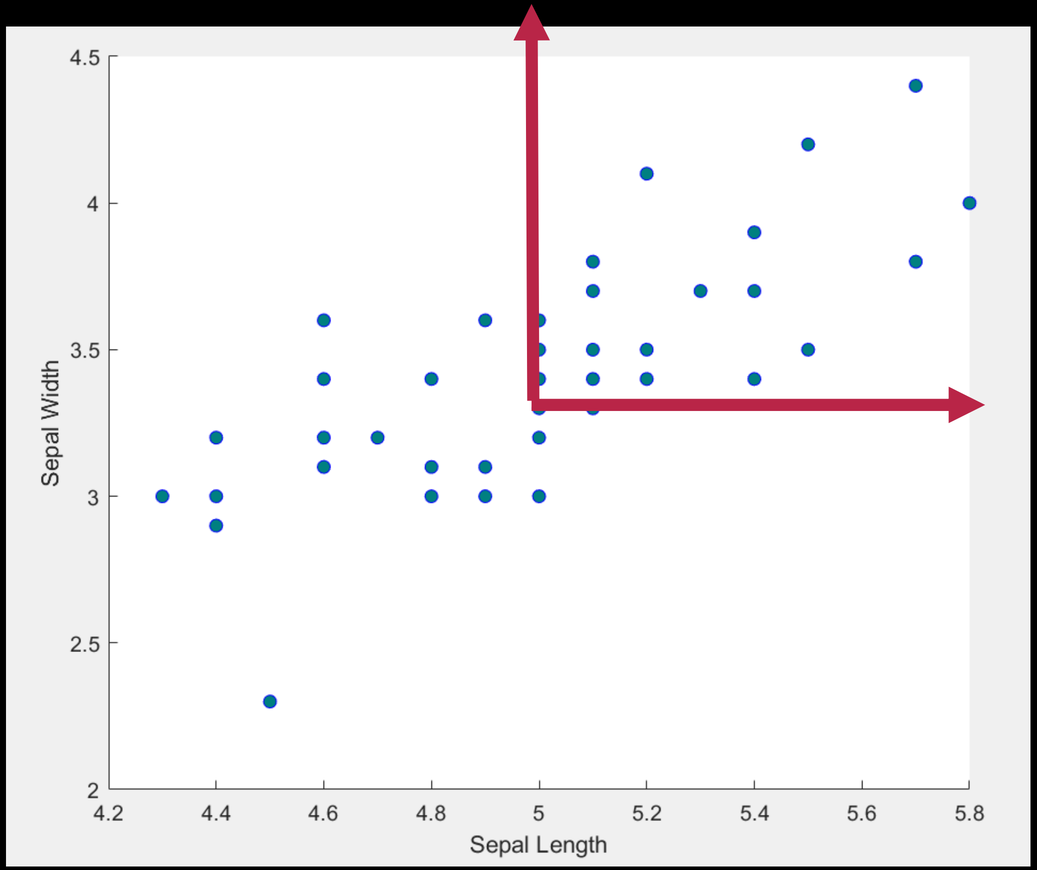


- Minimize redundancy
  - Covariance should be small
- Maximize signal
  - Variance should be large

Signal to noise ratio:

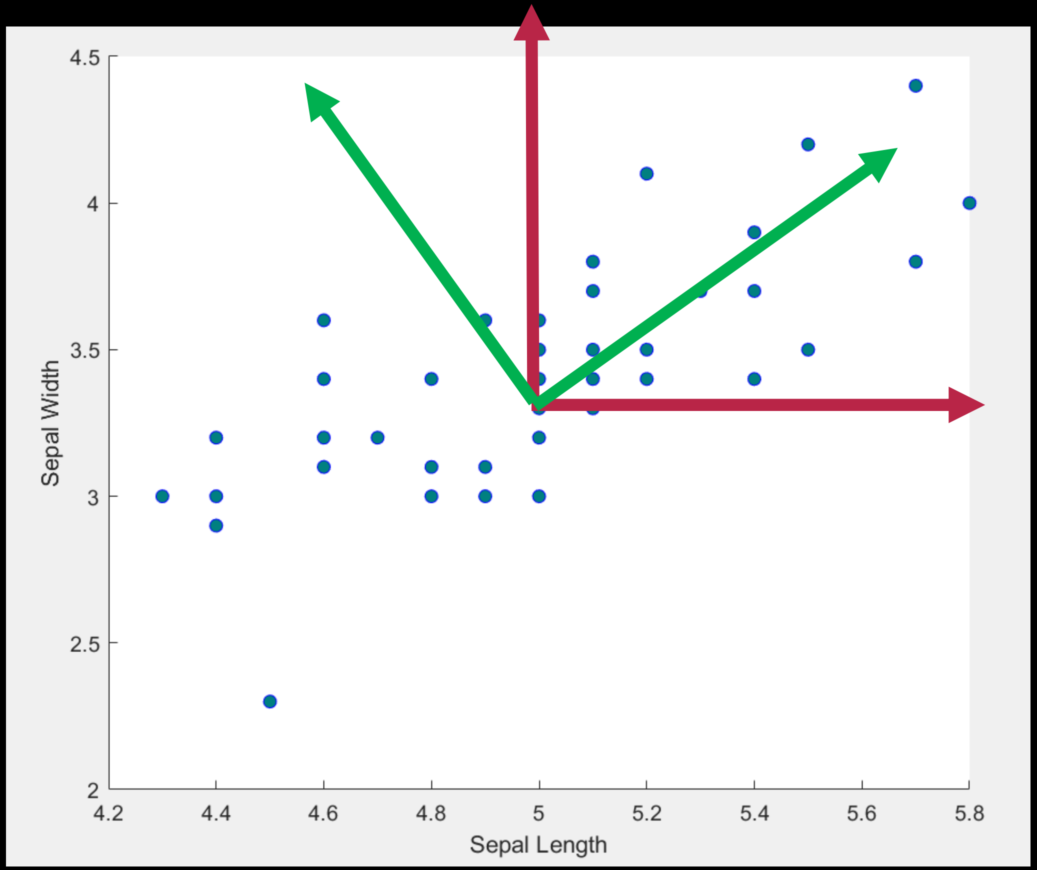
$$\mathrm{SNR} = \frac{\sigma^2_{\mathrm{signal}}}{\sigma^2_{\mathrm{noise}}}$$

Changing basis

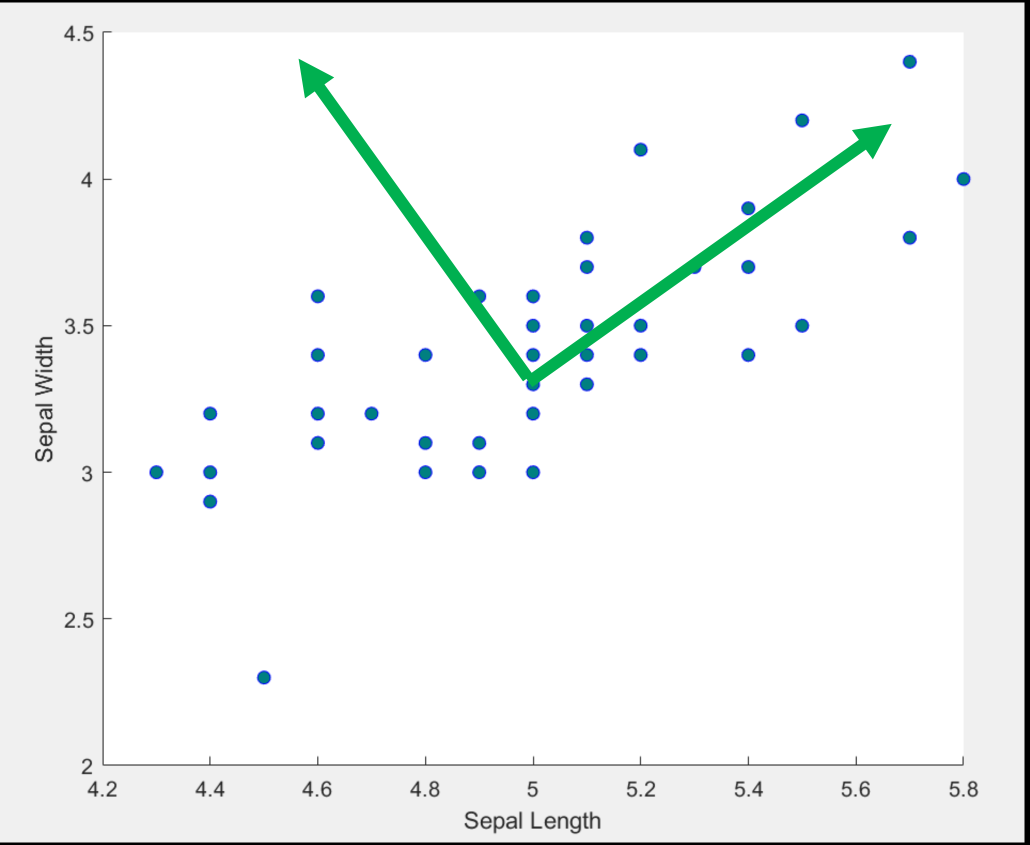


- We start by subtracting the mean
  - Centering data
- Red lines are the default basis

Changing basis

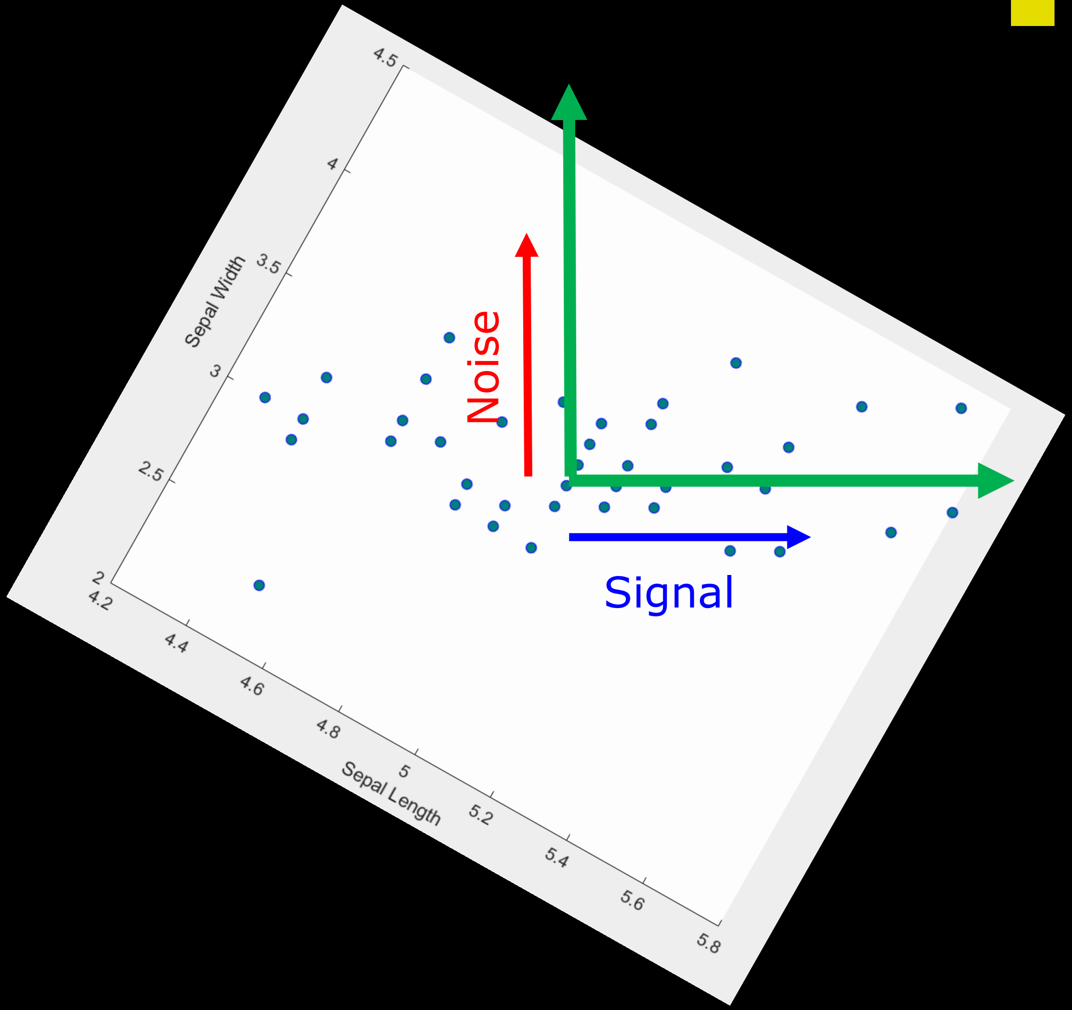


Changing basis



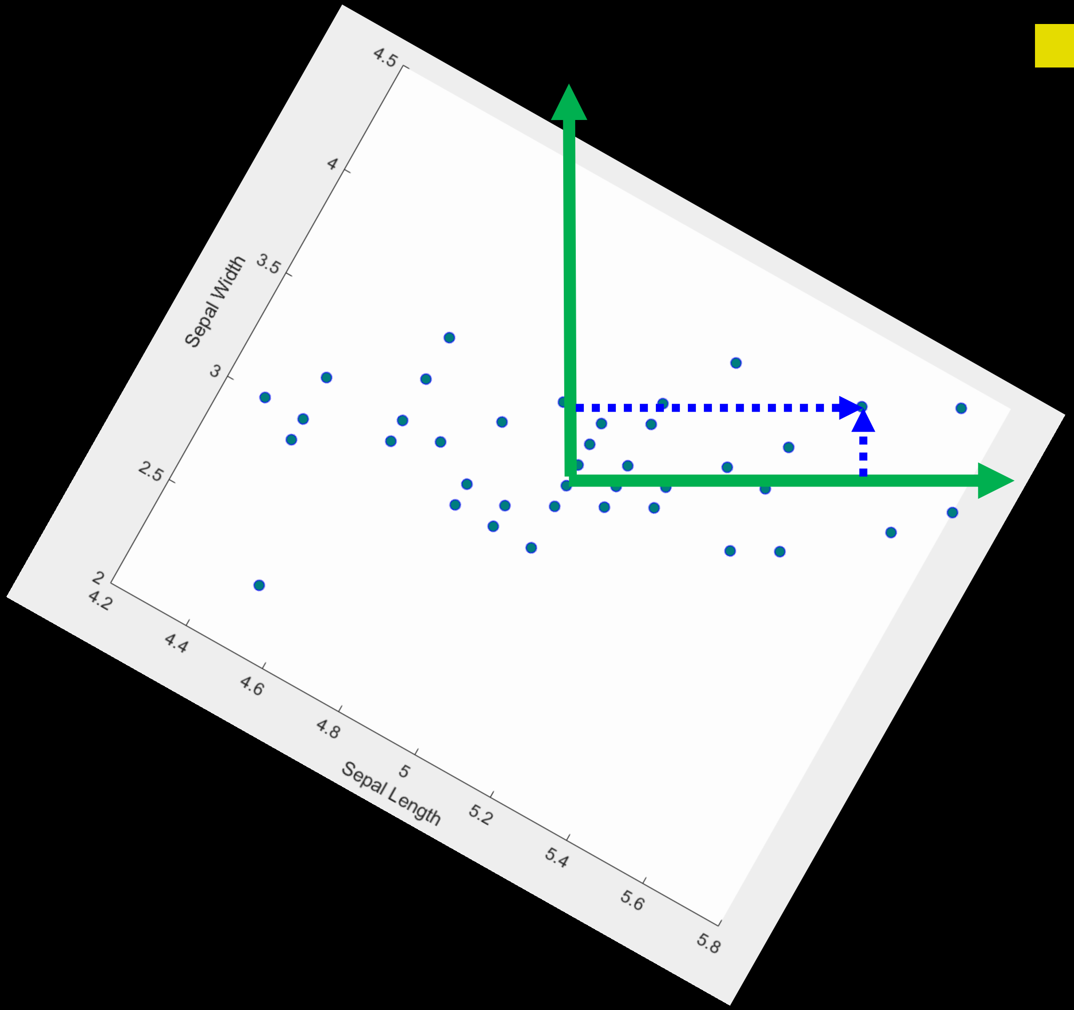
- A new basis that follows the *covariance* in the data

Changing basis



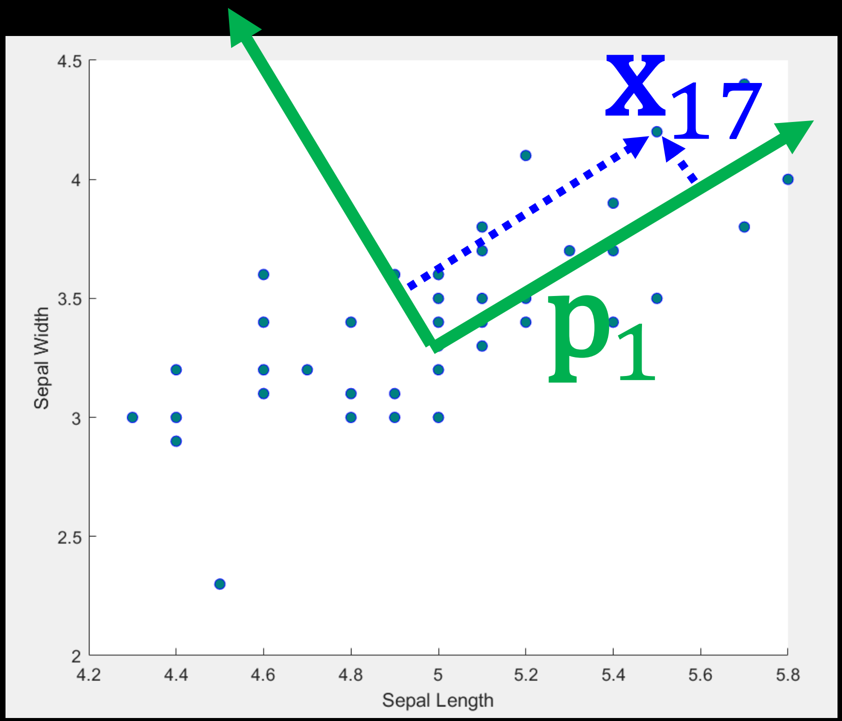
- Lets try to rotate the data – for visualisation

Changing basis



- Finding the measurement values in the new basis

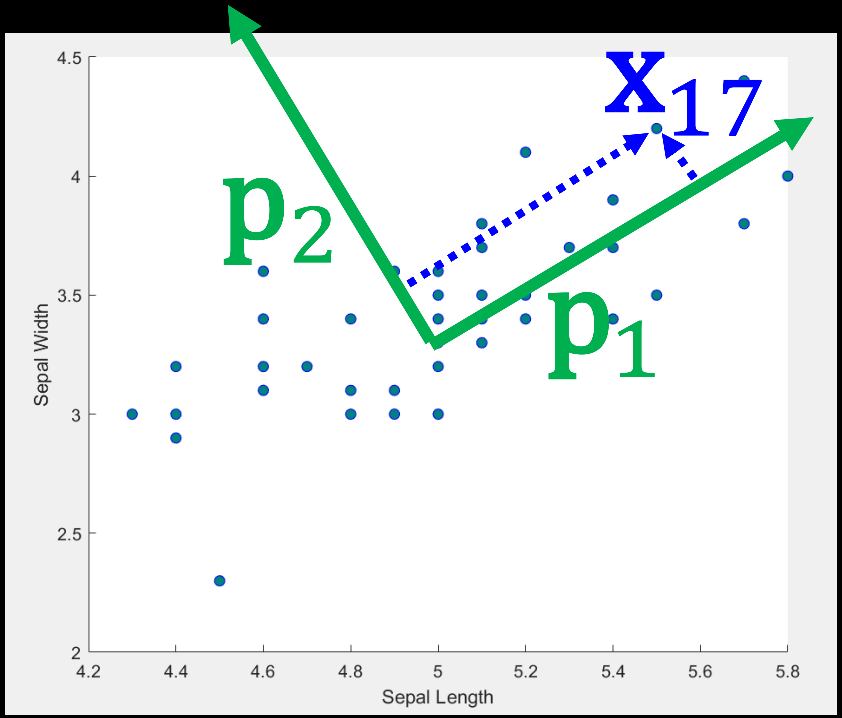
Changing basis



- The dot product projects a point down to a new axis

$$\mathbf{x}_{17,\text{new}} = x_{17} \cdot p_1$$

Changing basis



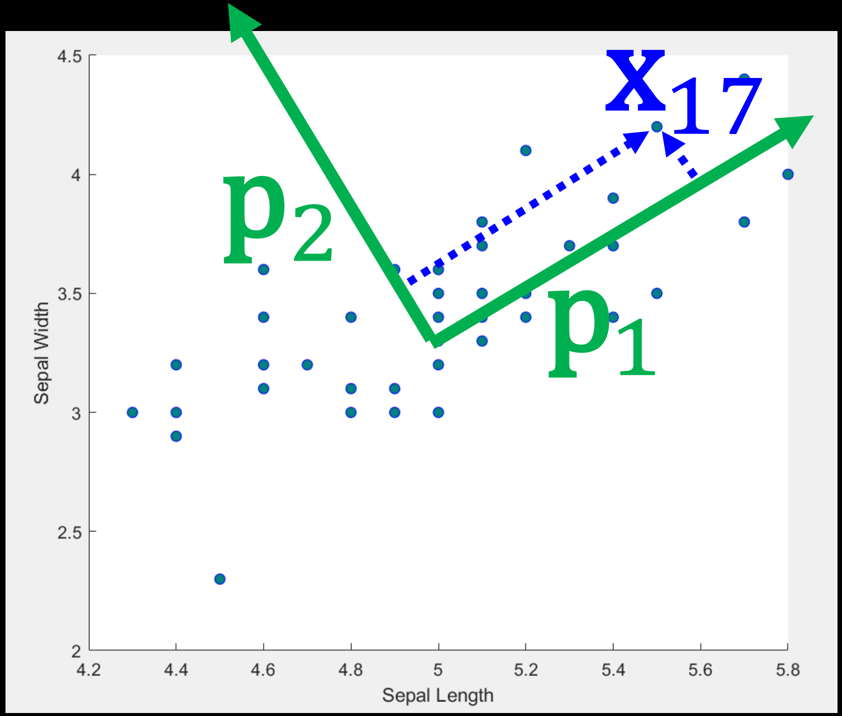
- The dot product projects a point down to a new axis

$$\mathbf{PX} = \mathbf{Y}$$

- $\mathbf{p}_1$ and $\mathbf{p}_2$ are the rows of $\mathbf{P}$

$$\mathbf{X} = \begin{bmatrix} \text{Sepal length}_1 & \cdots & \text{Sepal length}_{50} \\ \text{Sepal width}_1 & \cdots & \text{Sepal width}_{50} \\ \text{Petal length}_1 & \cdots & \text{Petal length}_{50} \\ \text{Petal width}_1 & \cdots & \text{Petal width}_{50} \end{bmatrix}$$

Changing basis

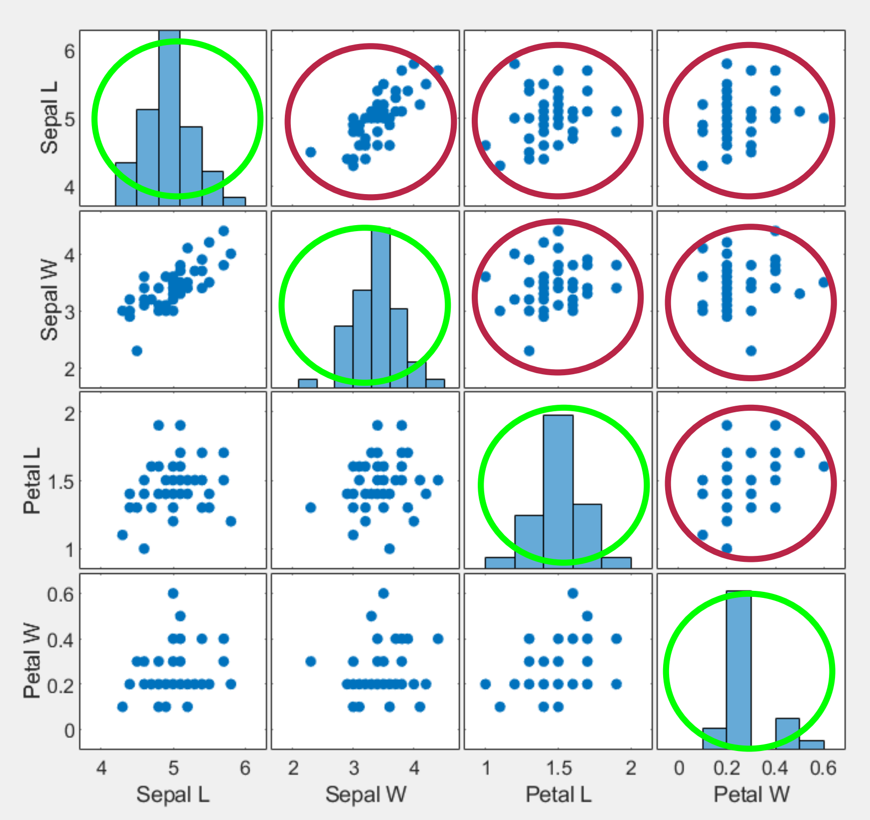


- The dot product projects a point down to a new axis

$$\mathbf{PX} = \mathbf{Y}$$

- Here **Y** contains the new coordinates/measurements per sample

Goals



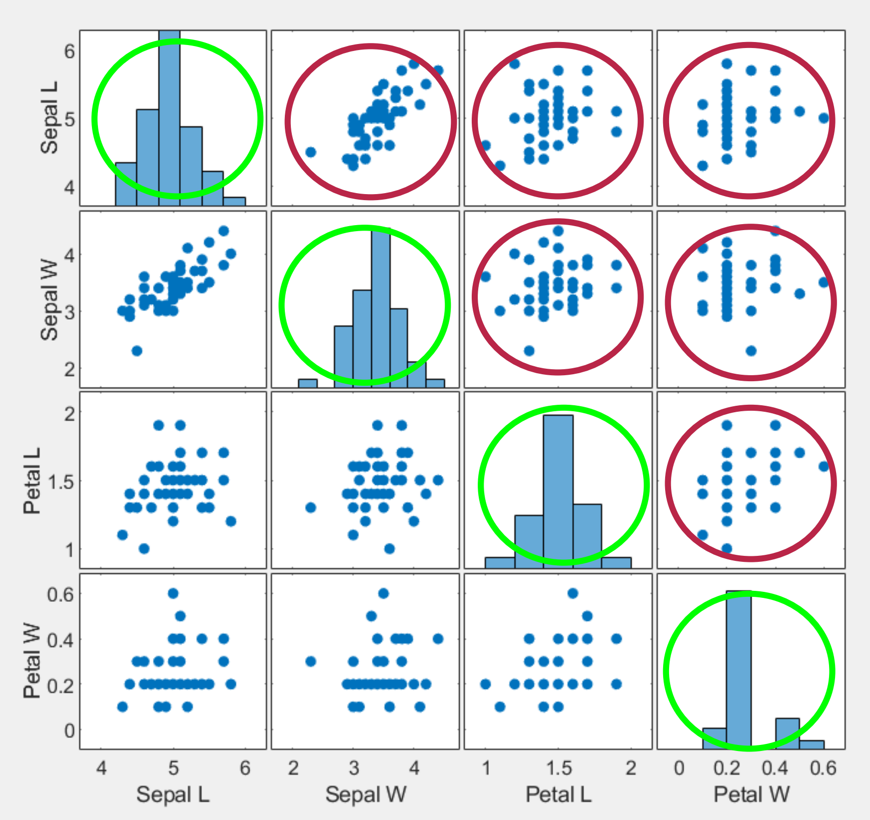
- Minimize redundancy
  - Covariance should be small
- Maximize signal
  - Variance should be large
- Transform our data
  - Rotating and scaling the basis

$$\mathbf{Y} = \mathbf{P}\mathbf{X}$$

- So it will have

$$\mathbf{C_Y} \equiv \frac{1}{n}\mathbf{Y}\mathbf{Y}^T$$

Goals



- The covariance matrix

$$\mathbf{C_Y} \equiv \frac{1}{n}\mathbf{YY}^T$$

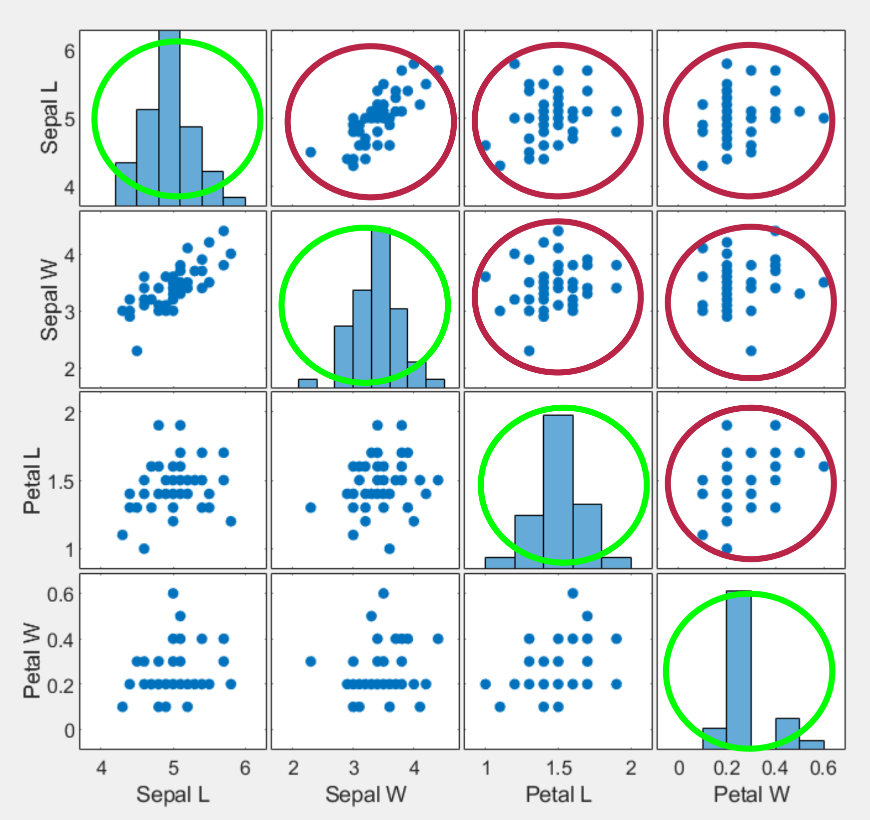
- Should be *as diagonal as possible*

- We do this by

$$\mathbf{Y} = \mathbf{PX}$$

  – Where **P** are the principal components

Computing the principal components

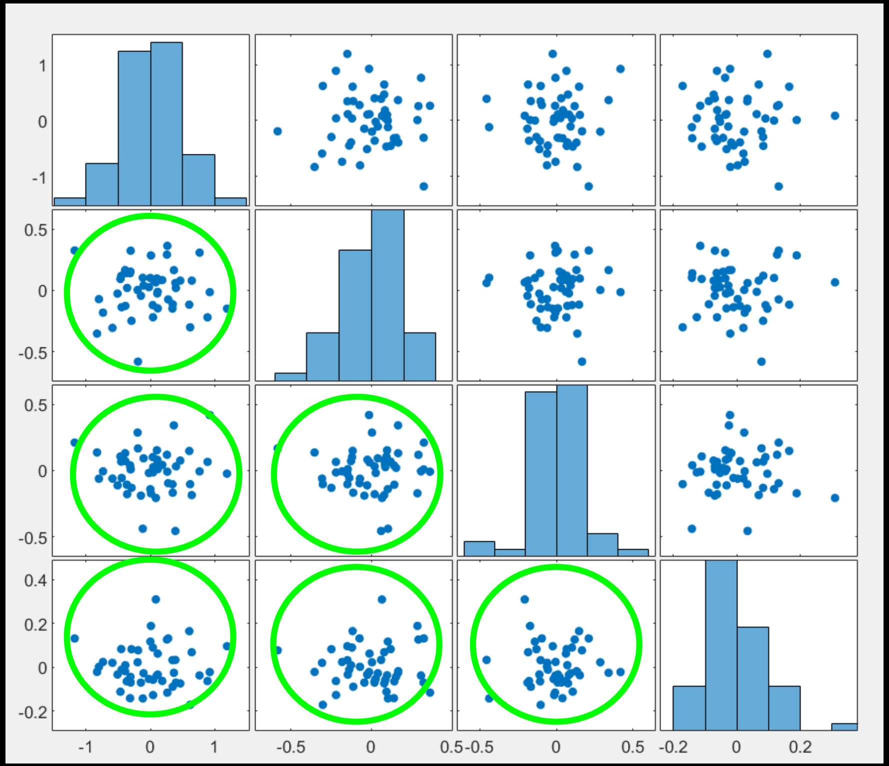


- The Principal Components of $\mathbf{X}$ are the eigenvectors of

$$\mathbf{C_X} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$$

- The i'th diagonal value of $\mathbf{C_Y}$ is the variance along principal component number i

New covariance matrix for Iris data



Covariance: 0

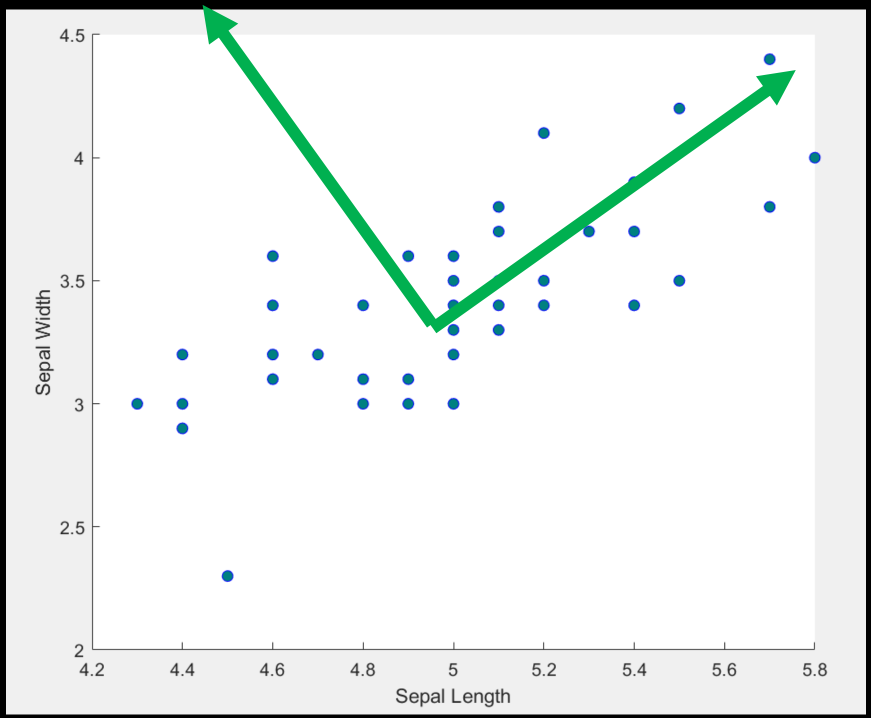
- The principal component are found and

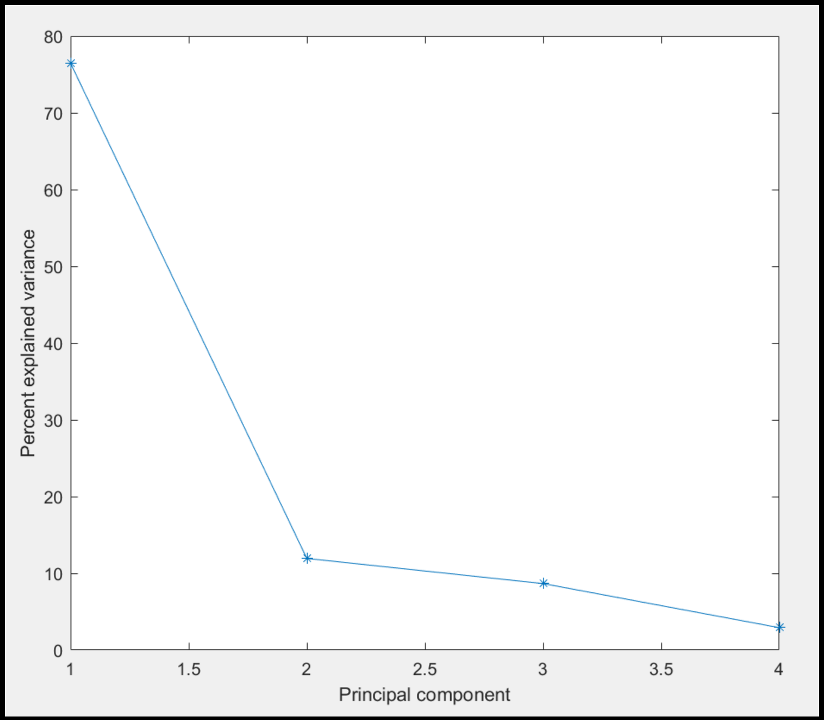
$$\mathbf{Y} = \mathbf{P}\mathbf{X}$$

- With the covariance matrix

$$\mathbf{C_Y} \equiv \frac{1}{n}\mathbf{Y}\mathbf{Y}^T$$

Explained variance





One component explains 75% of the total variation – so for each flower we can have one number that explains 75% percent of the 4 measurements!

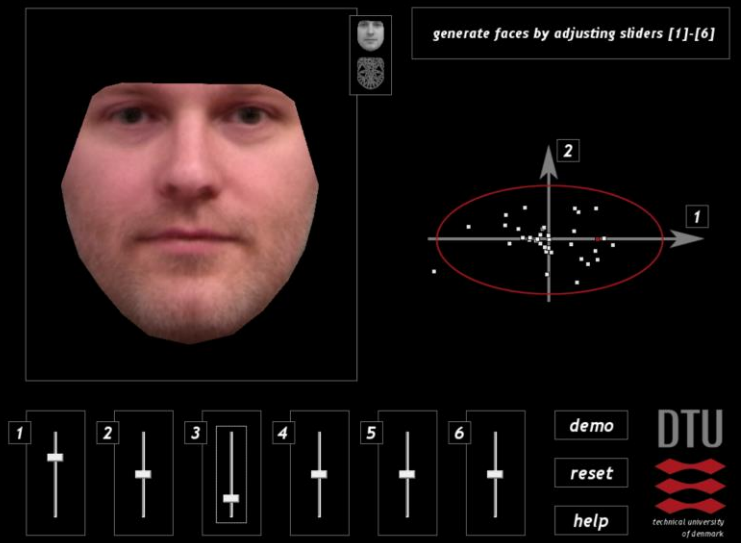
What can we use it for?

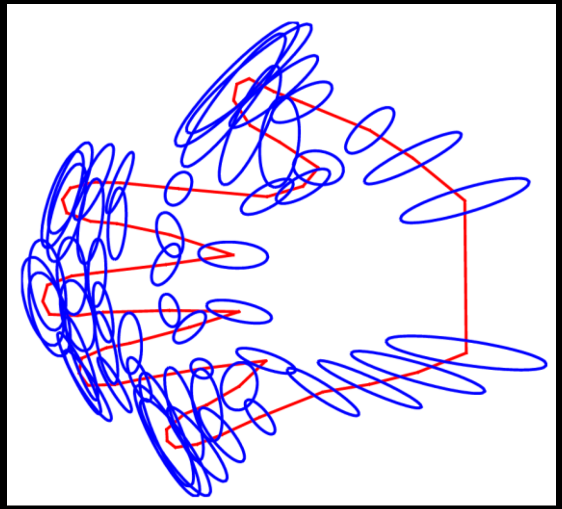
- Classification

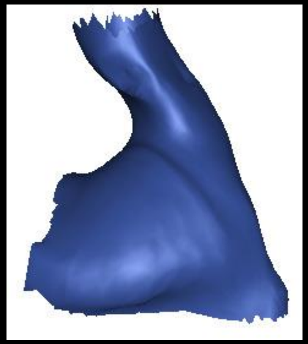
Based on one value instead of 4

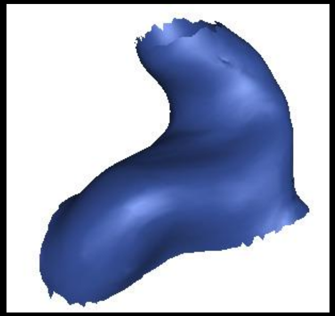
What can we use it for?

- Many more examples in the course









02502
Image Analysis
Lecture 2: Image acquisition, compression and storage

http://courses.compute.dtu.dk/02502/

Tim B. Dyrby (tbdy@dtu.dk)
Associate Professor, DTU Compute
Copenhagen University Hospital Hvidovre
Danish Research Centre MRI

&
Rasmus R. Paulsen (rapa@dtu.dk)
Associate Professor, DTU Compute

Image Analysis
The amazing adaptive brain

- How much is the brain involved when biking?

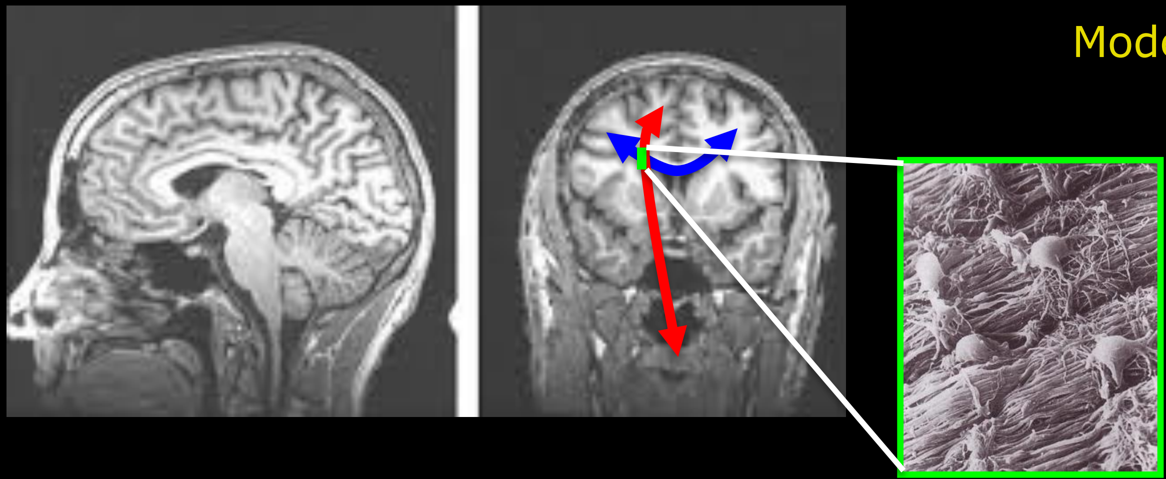
- Aging, dementia and Alzheimer

- Schizophrenia

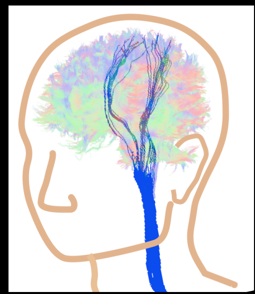
- Multiple sclerosis

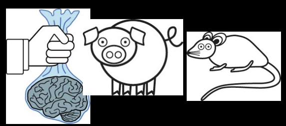
Mapping brain network and microstructure using MRI

- Magnetic Resonance Imaging (MRI) of the living brain

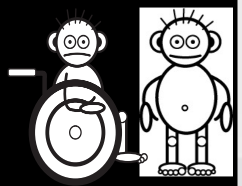


Model of the brain network
From MRI



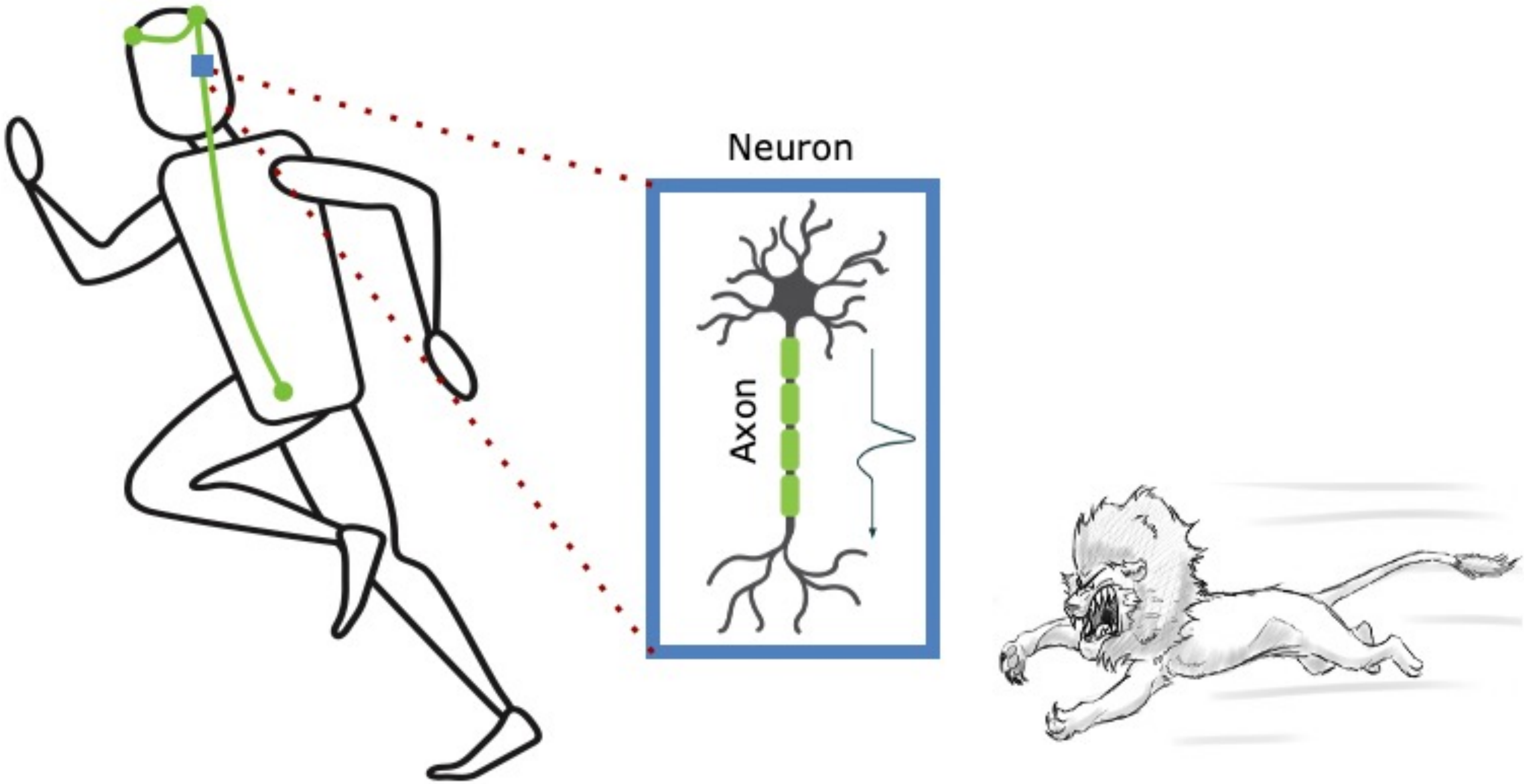


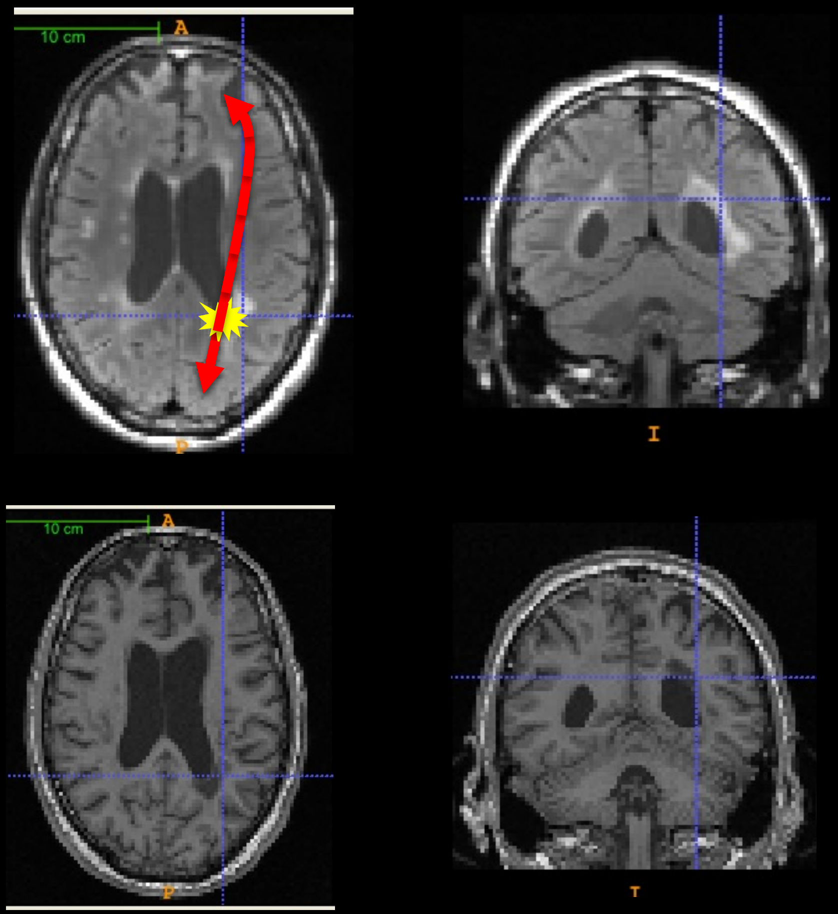
Preclinical MRI 7Tesla



Human 3Tesla MRI scanner

Communication speed in the brain network matters!

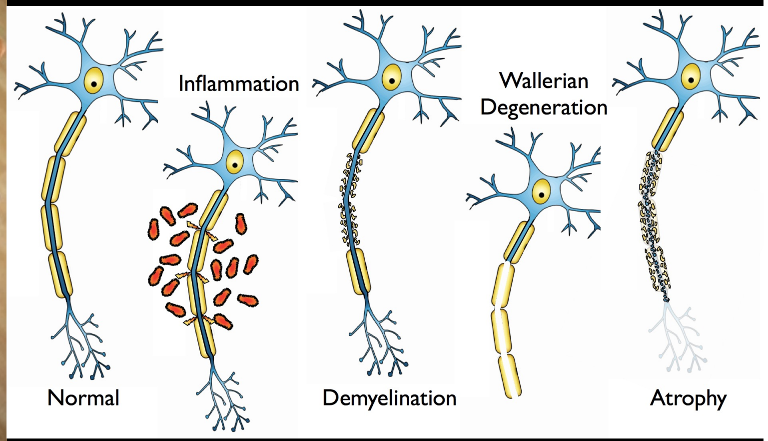




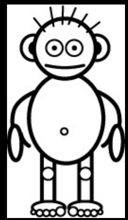
The challenges:

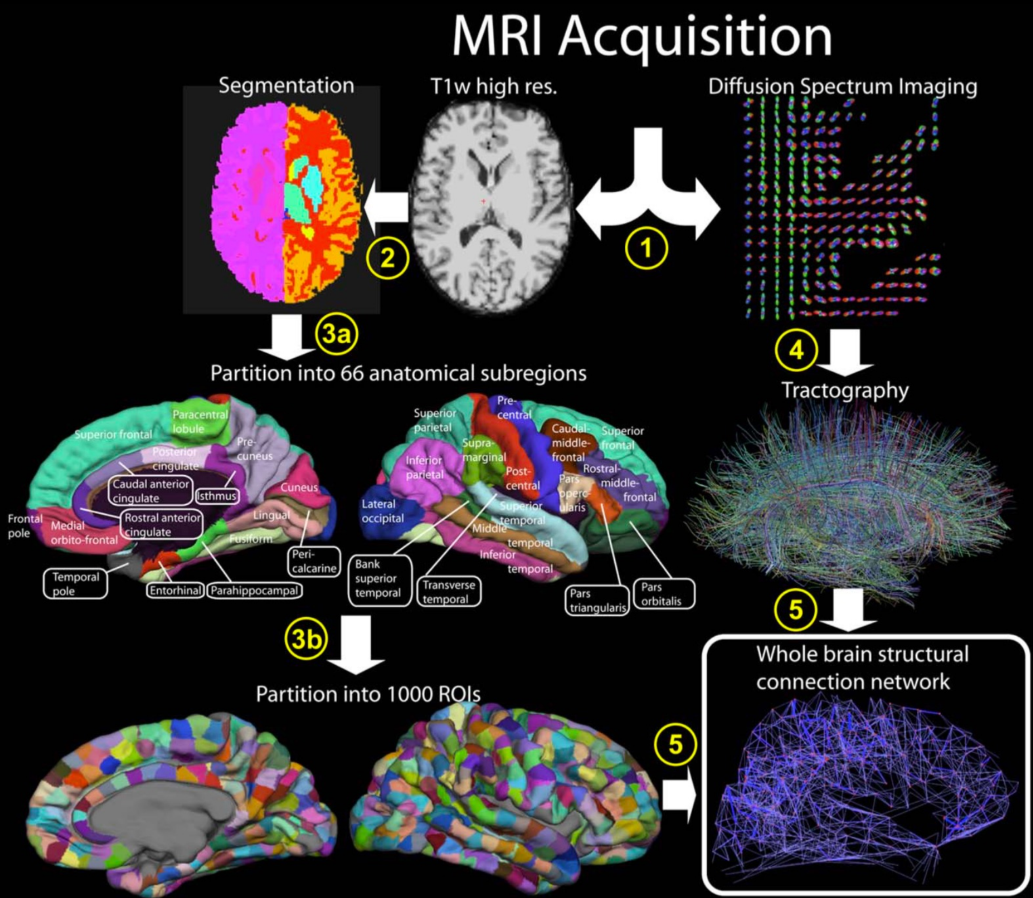
- Weakly correlating with clinical tests
- Clinical MRI is very sensitive to anatomical changes but often lacks specificity

Disease attacks the brain neurons



Human 3T MRI scanner





Hagmann et al., 2008, PloS Biol

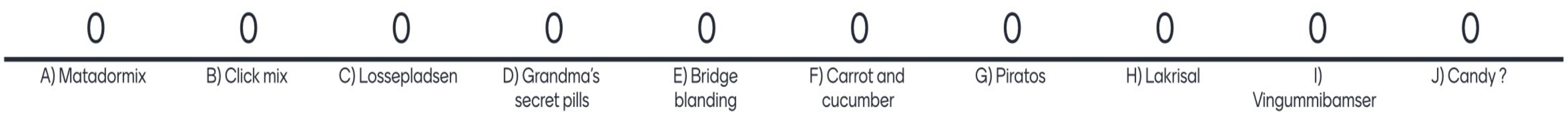
Image analysis:

- Image acquisition
- Data storage
- Point wise operations - Histograms
- Segment anatomical structures
- PCA analysis
- Design biophysical mathematical models of the brain network

Go to **www.menti.com** and use the code **6732 4259**

Mentimeter

Quiz 1: What is your favourite candy

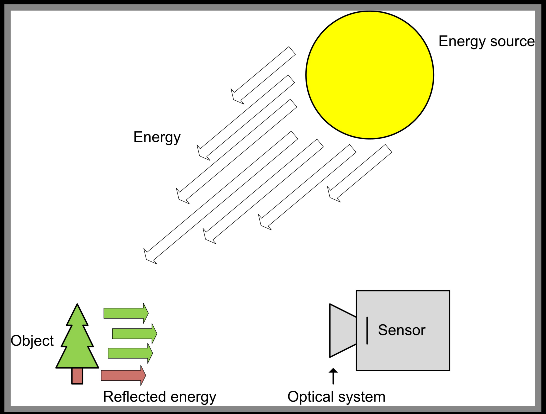


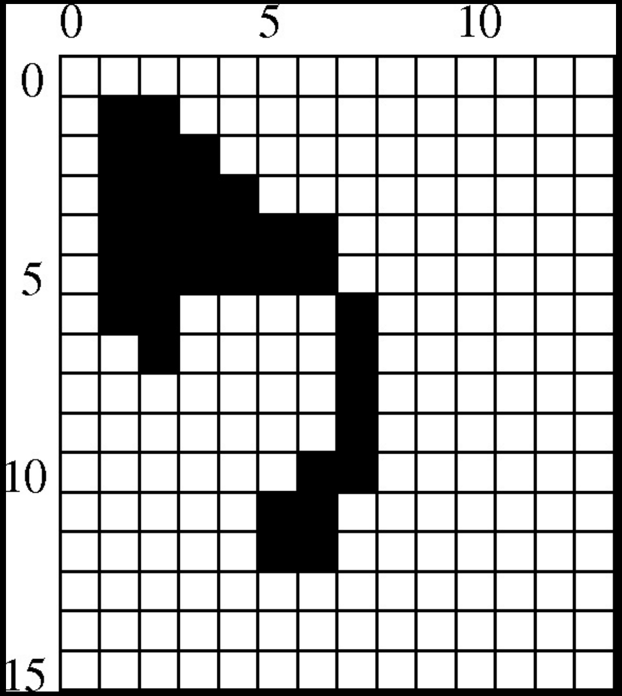
Quiz 1: What is your favourite candy

A) Matadormix
B) Click mix
C) Lossepladsen
D) Grandma's secret pills
E) Bridge blanding
F) Carrot and cucumber
G) Piratos
H) Lakrisal
I) Vingummibamser
J) Candy ?

Lecture 2

- Image acquisition, compression and storage

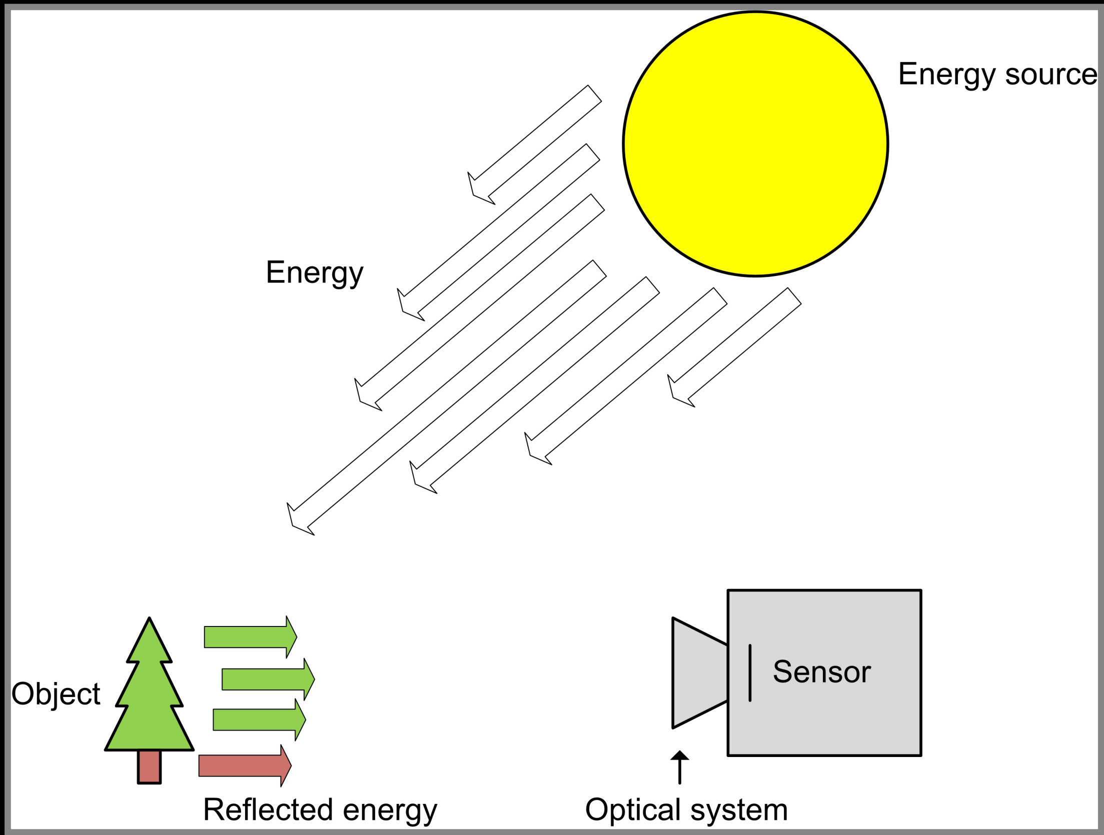




Today – What can you do after today?

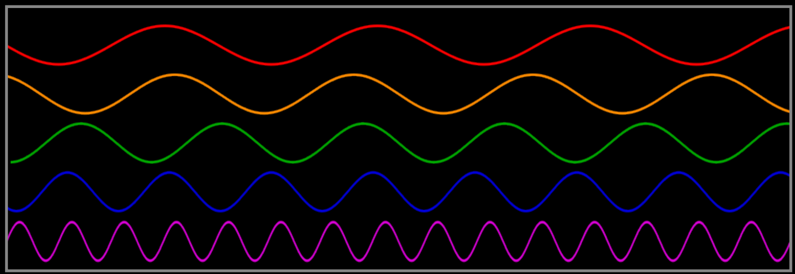
- Explain where visible light is in the electromagnetic spectrum
- Describe the pin hole camera
- Describe the properties of a thin-lens including focal-length, the optical center, and the focal point
- Estimate the focal length of a thin lens
- Compute the optimal placement of a CCD chip using the thin lens equation
- Describe depth-of-field
- Compute the field-of-view of a camera
- Explain the simple CCD model
- Compute the run-length code of a grayscale image
- Compute the chain coding of a binary image
- Compute the run length coding of a binary image
- Compute the compression ratio
- Describe the difference between a lossless and a lossy image format
- Decide if a given image should be stored using a lossless or a lossy image format
- Understand the principle of X-ray and MRI imaging methods

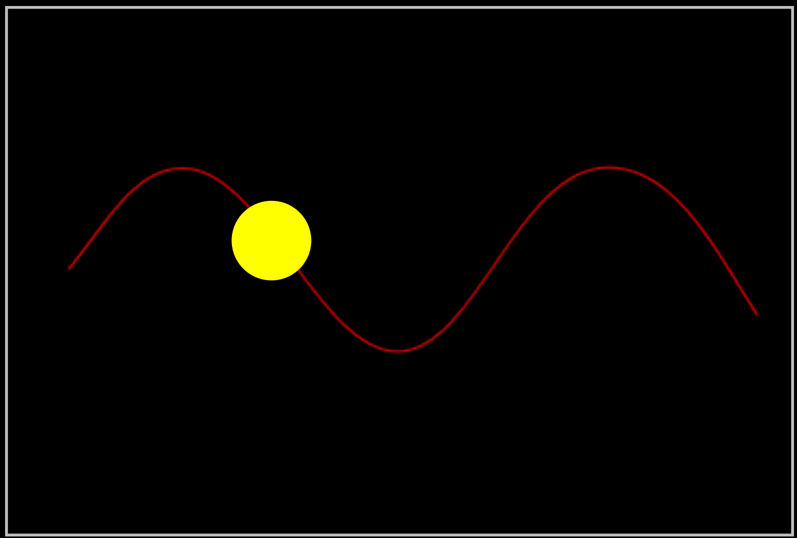
How is an image created?



This is just one way! Other methods will be described later in the course.

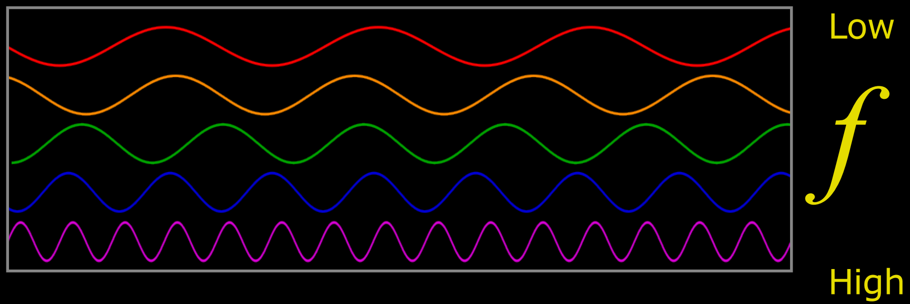
What is light?



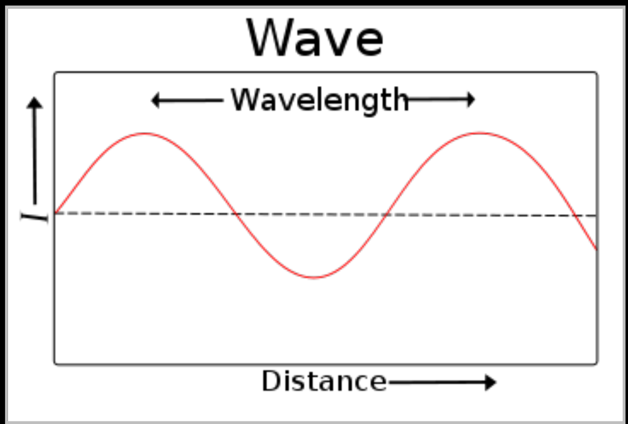


- Can be seen as electromagnetic waves
- Or as a photon (from Greek *phōtos*, “light”)
  - Mass less fundamental particle

Light as a wave



$$\lambda = \frac{c}{f}$$



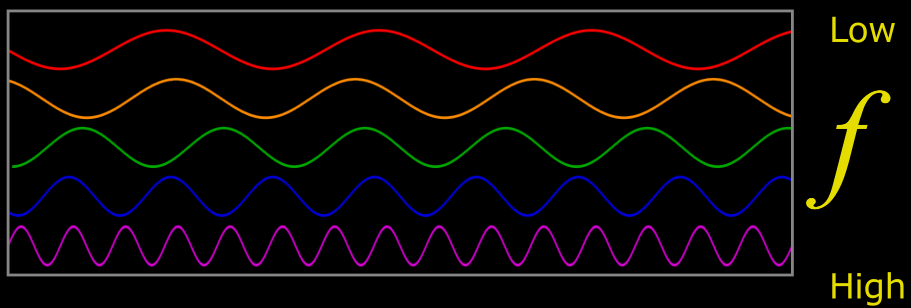
- It has a frequency f
  - Measured in Hertz [Hz]
- It has a wavelength λ (lambda)
  - Measured in meters [m]
- It has a speed
  - "The speed of light" c
  - 299.792 458 [m/s]

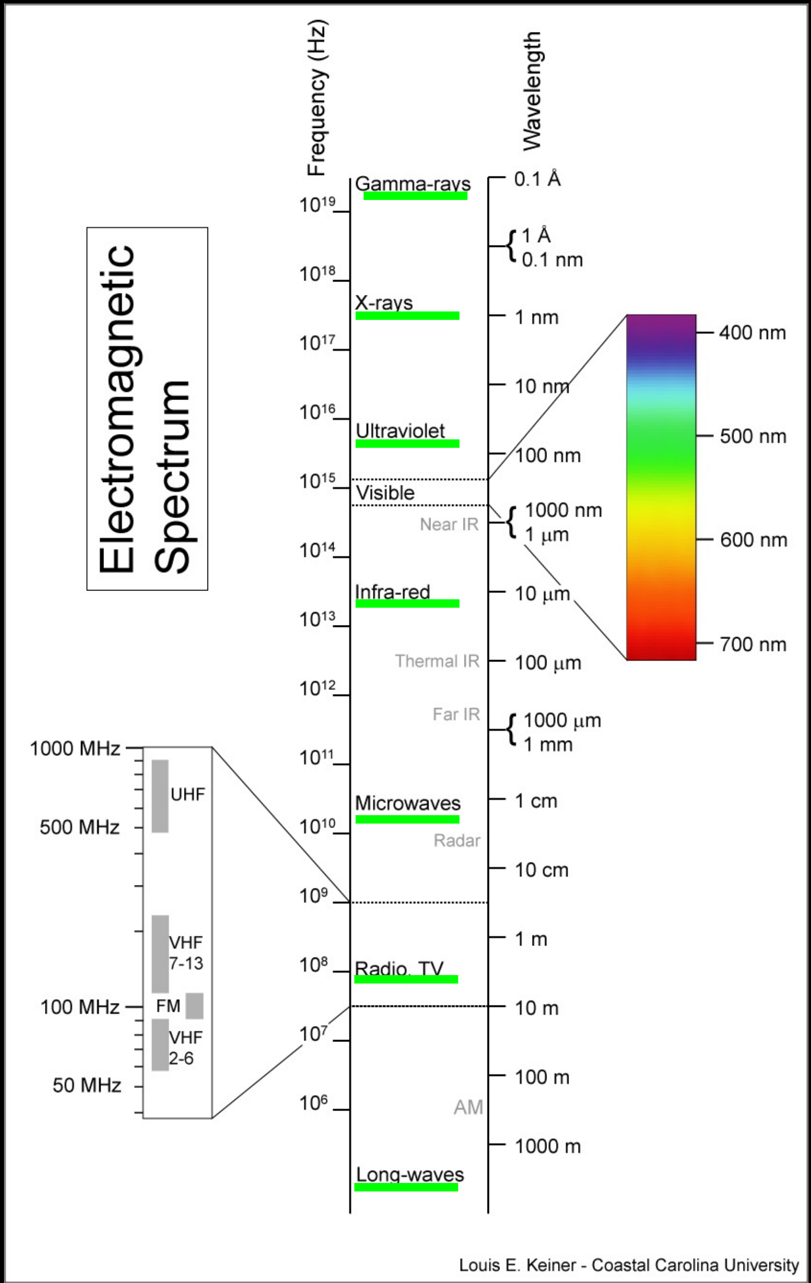
- High frequency -> short waves
- Low frequency -> long waves

Energy of light

$$E = h \cdot f$$

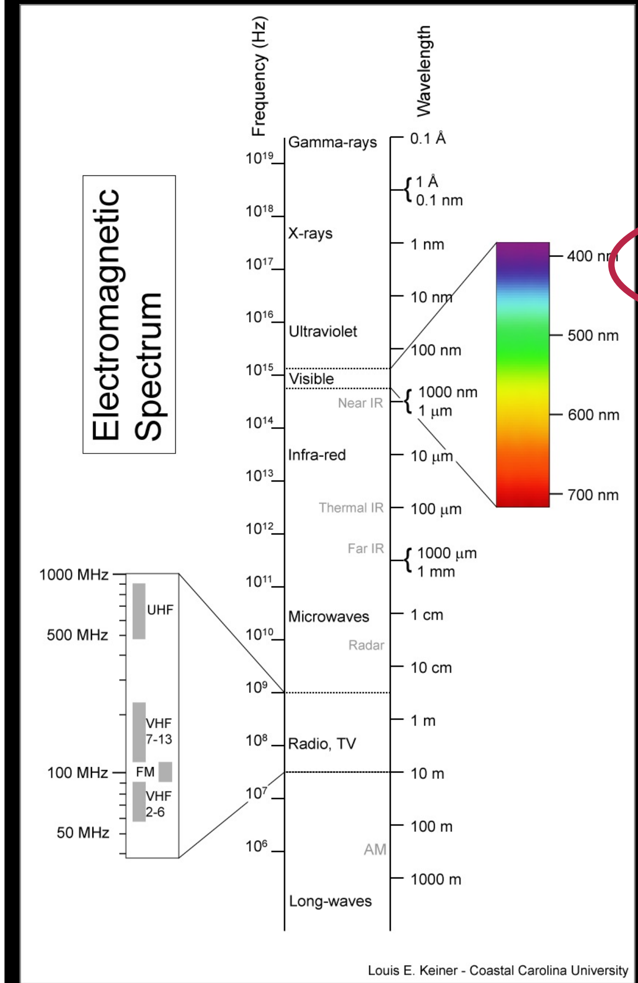
- Light has energy
  - You can feel it in the sun!
- Planck's constant h
- High frequency -> high energy
- Long waves -> low energy





- Electromagnetic spectrum
  - Range of all frequencies
  - Divided into 7 regions
- Wavelengths
  - 1 µm = 1 micrometer = 0.001 mm
  - 1 nm = 1 nanometer = 0.0000001mm

Quiz 2: What has the most energy?



A) Radio

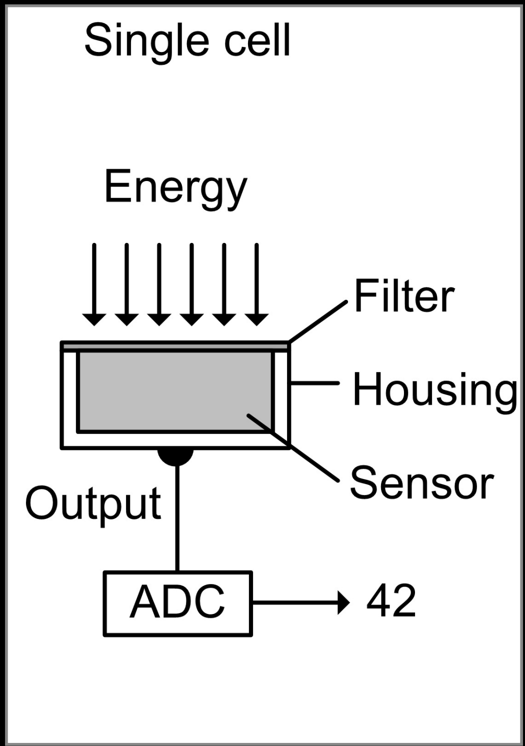
B) X-rays

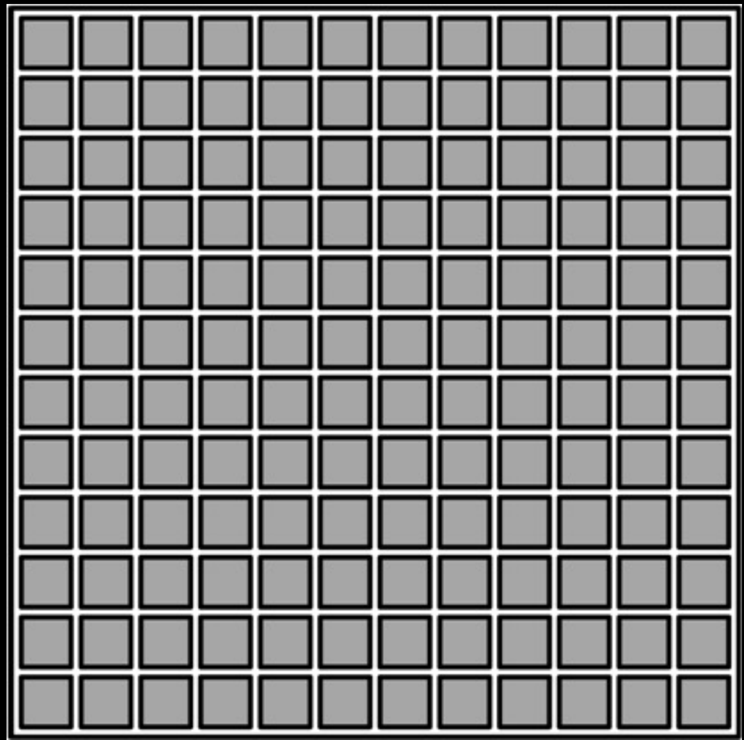
C) Red light

D) Microwave

E) Ultraviolet

How do light become a digital image?



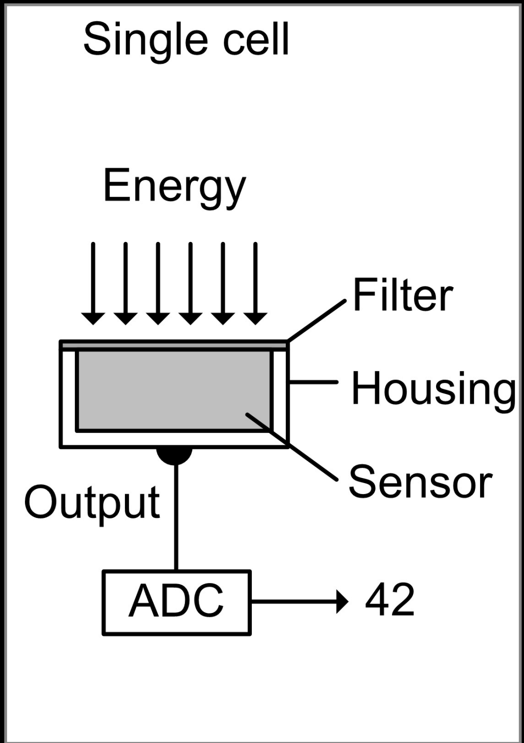


Charged coupled device

(CCD-chip)

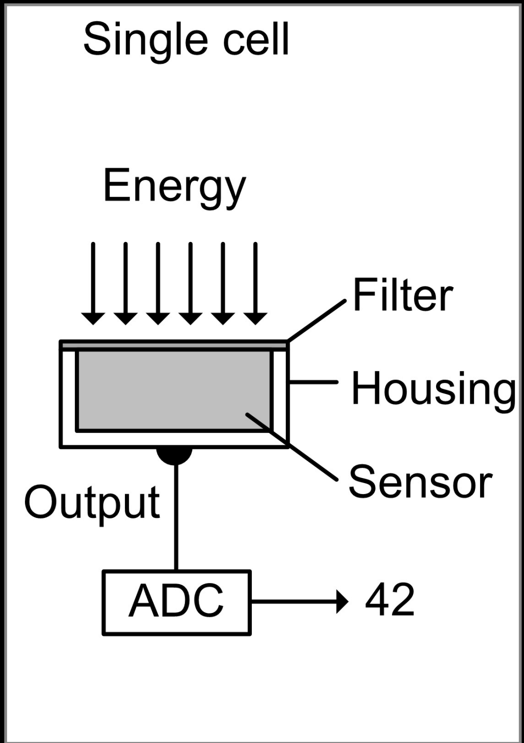
The digital film!

The CCD cell



- The cell can be seen as a well that collects energy
- It collect energy for a limited time (*to be charged*)
  - Exposure time
  - Integration time
  - Shutter

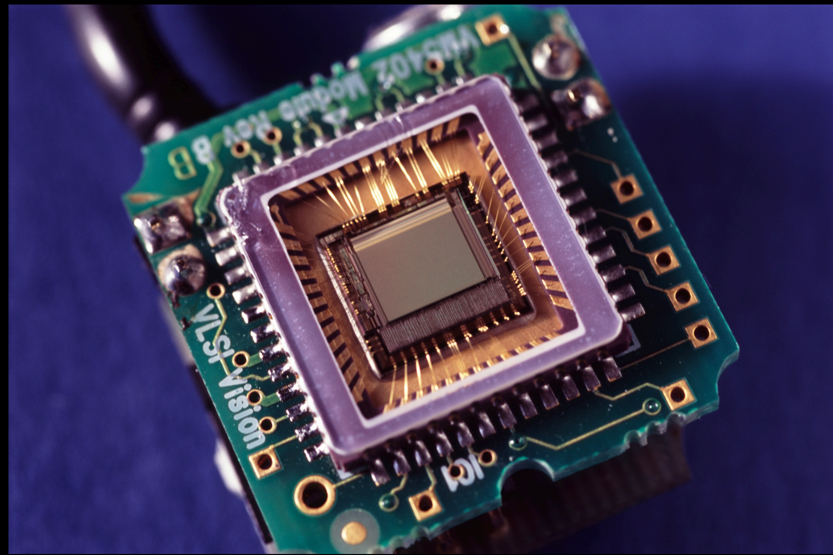
The CCD cell - conversion



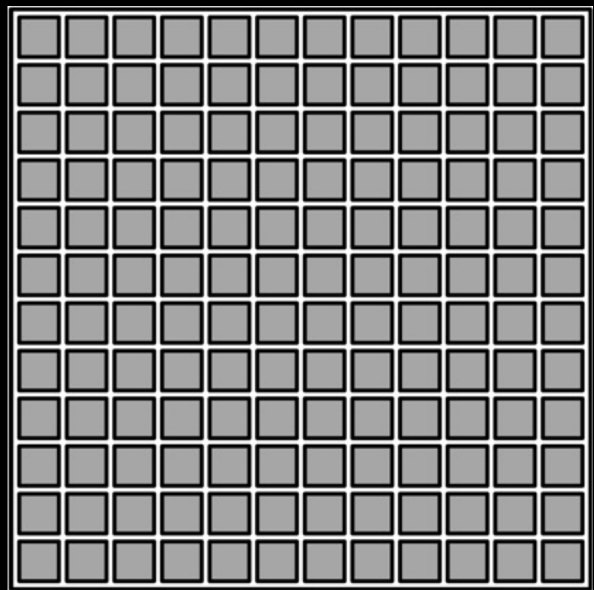
- Energy transformed to a digital number
  - Analog-to-Digital converter (ADC)
- Takes a an "analog signal" and converts it to a digital signal

$(01001001)_2 = (0 \times 2^7) + (1 \times 2^6) + (0 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = (73)_{10}$

CCD and images

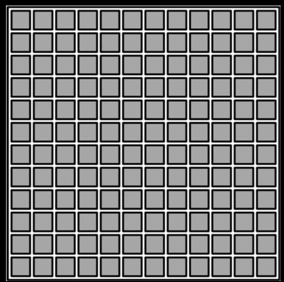


- Surprise! 1 CCD cell = 1 pixel
  - Only for grayscale images
  - More complex for RGB images
- 10 MPixel camera
  - 10 millions analog to digital conversions for one image!



Quiz 3: What is the size of a single CCD cell?

5.3 mm



7.2 mm

2048 x 1536 pixels

Solution:

7.2/2048= 3.5um

5.3/1536= 3.5um

A) 1 x 1 milimeter

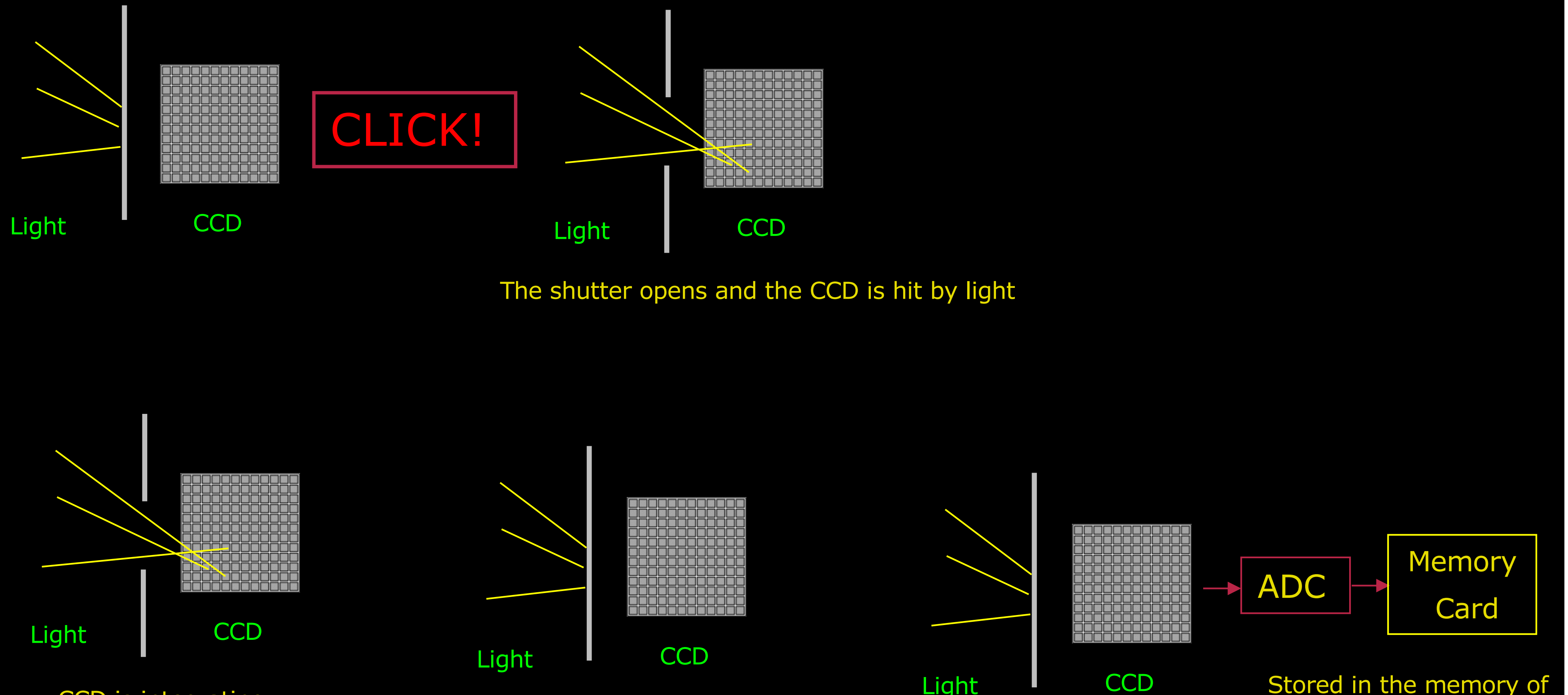
B) 3.5 x 3.5 micrometer

C) 0.002 x 0.002 milimeter

D) 5.6 x 5.6 micrometer

E) 0.4 x 0.4 milimeter

What happens when you press the button?



The shutter opens and the CCD is hit by light

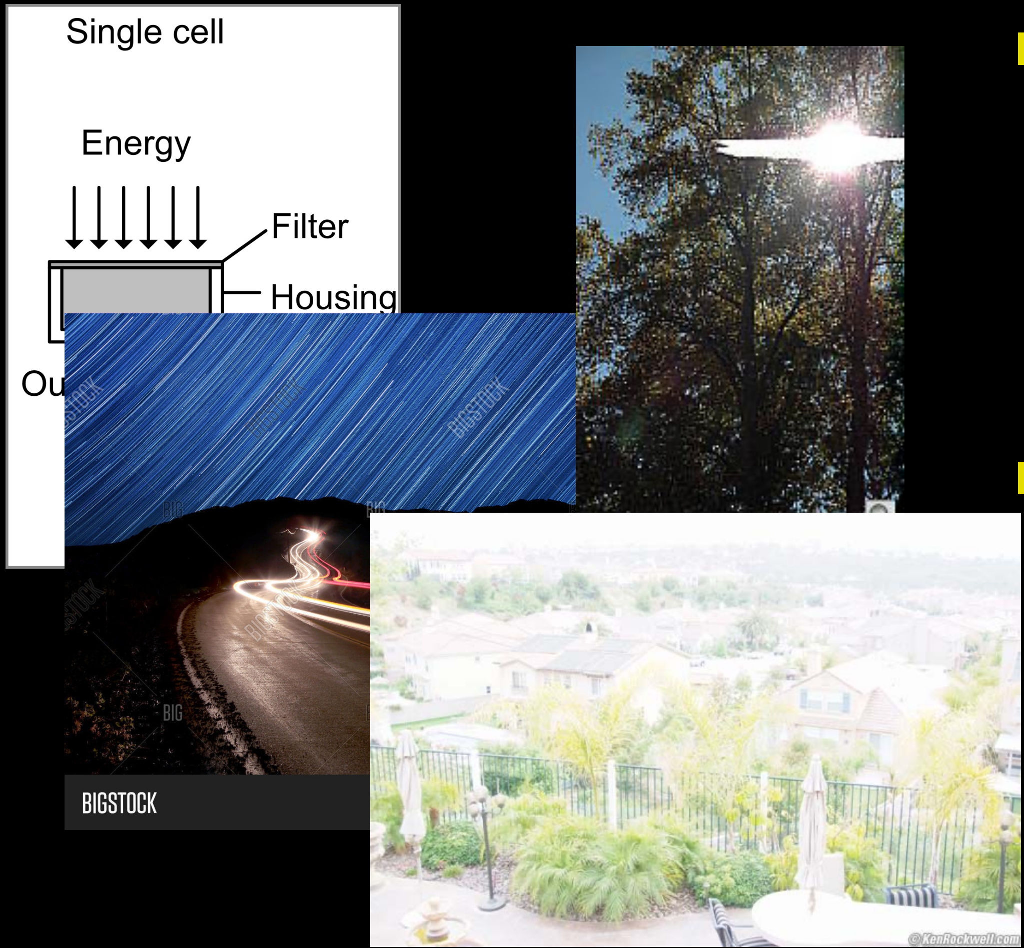
CCD is integrating

Shutter closes

The collected energies are transferred and converted to digital values

Stored in the memory of the camera

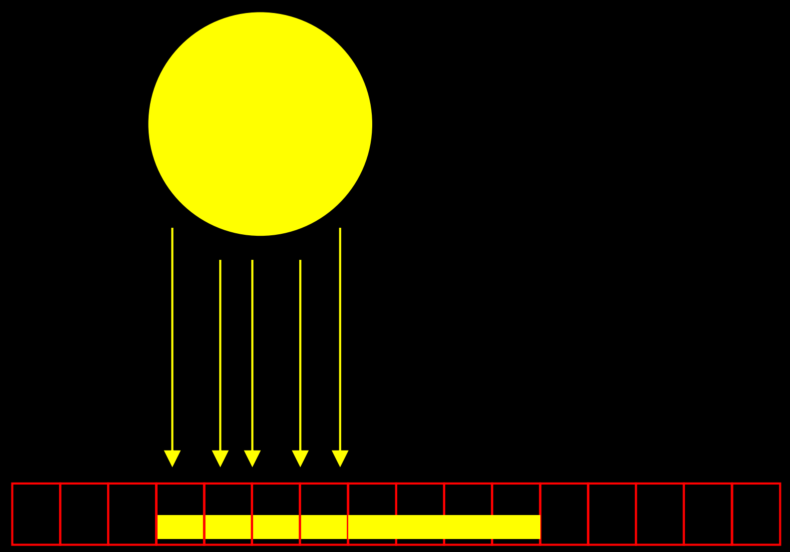
Question: Integration time



- What happens if we integrate over long time?
  - Motion blur
  - Over-exposure (the well is overrunning)
  - Blooming
- Short integration time
  - Noise
  - Lack of contrast

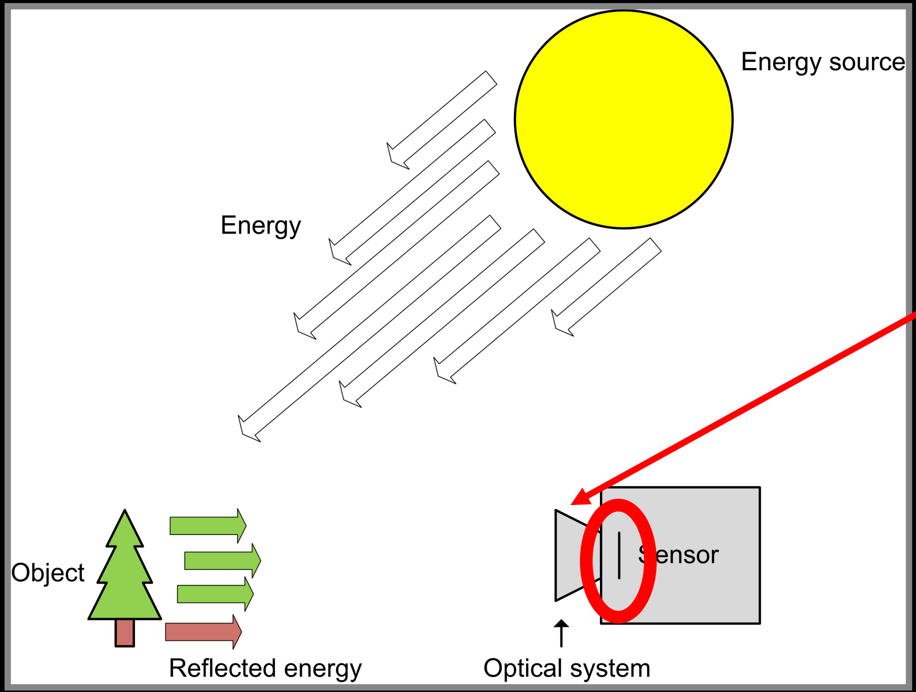
Motion blur

- Causes blurring of the moving object



The bigger picture

- A camera is more than a CCD!
- The CCD is the sensor!
- There is also "an optical system"



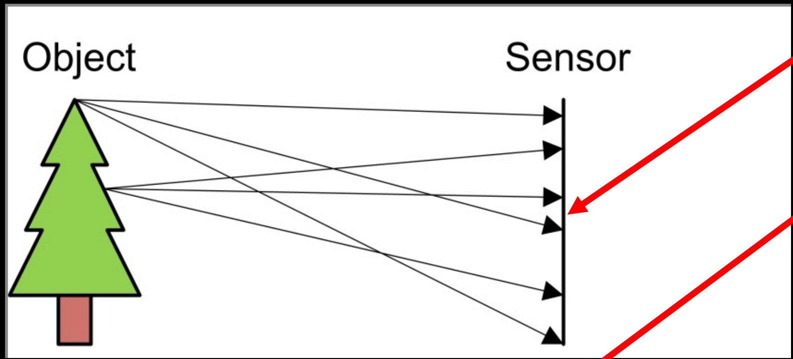
Optical system (lenses)

Optical system

- How do we get an image on the CCD?
- Light follows a straight line
- Light that hit one spot reflects in many directions

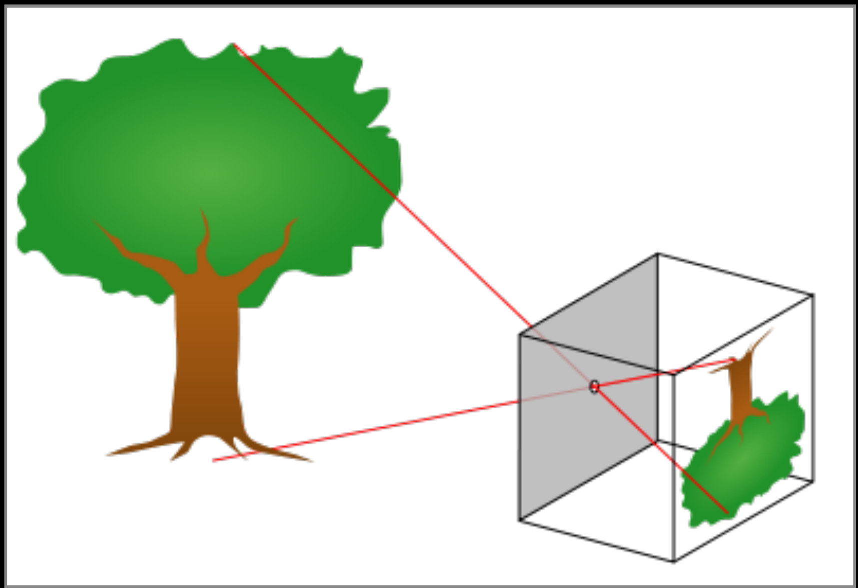
Same point hit by rays from
all over the object

Barrier with tiny hole

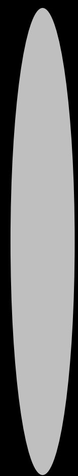


nera

Pinhole camera



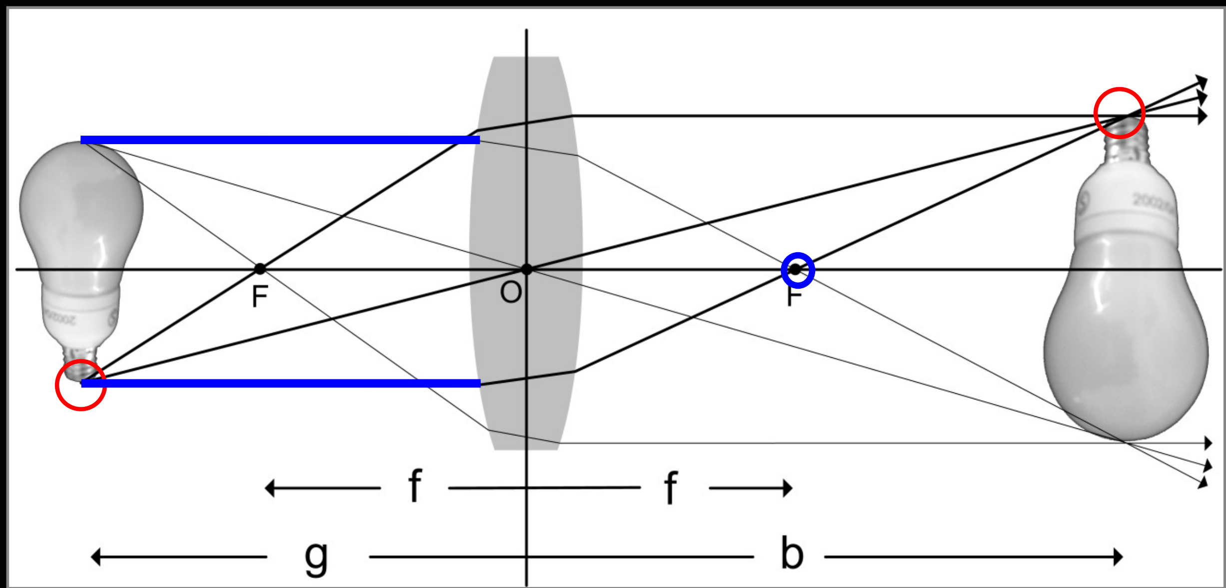
- Light coming through the tiny hole – any problems?
  - Very little light!
- How do we get more light inside the camera?
  - While keeping the focus?



A lens!

The lens

- A lens focuses a bundle of rays to one point
- Parallel rays pass through a focal point **F** at a distance f beyond the plane of the lens. f is the focal length
- O is the optical centre. F and O span the optical axis

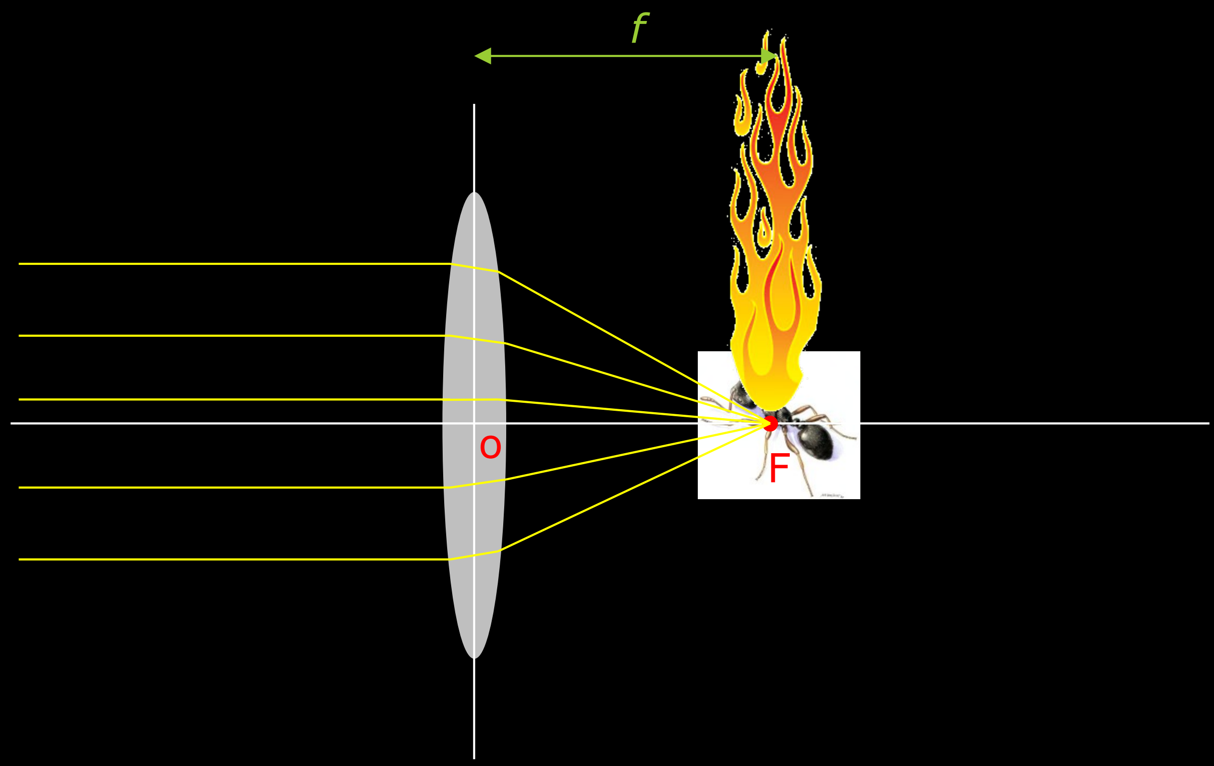


World

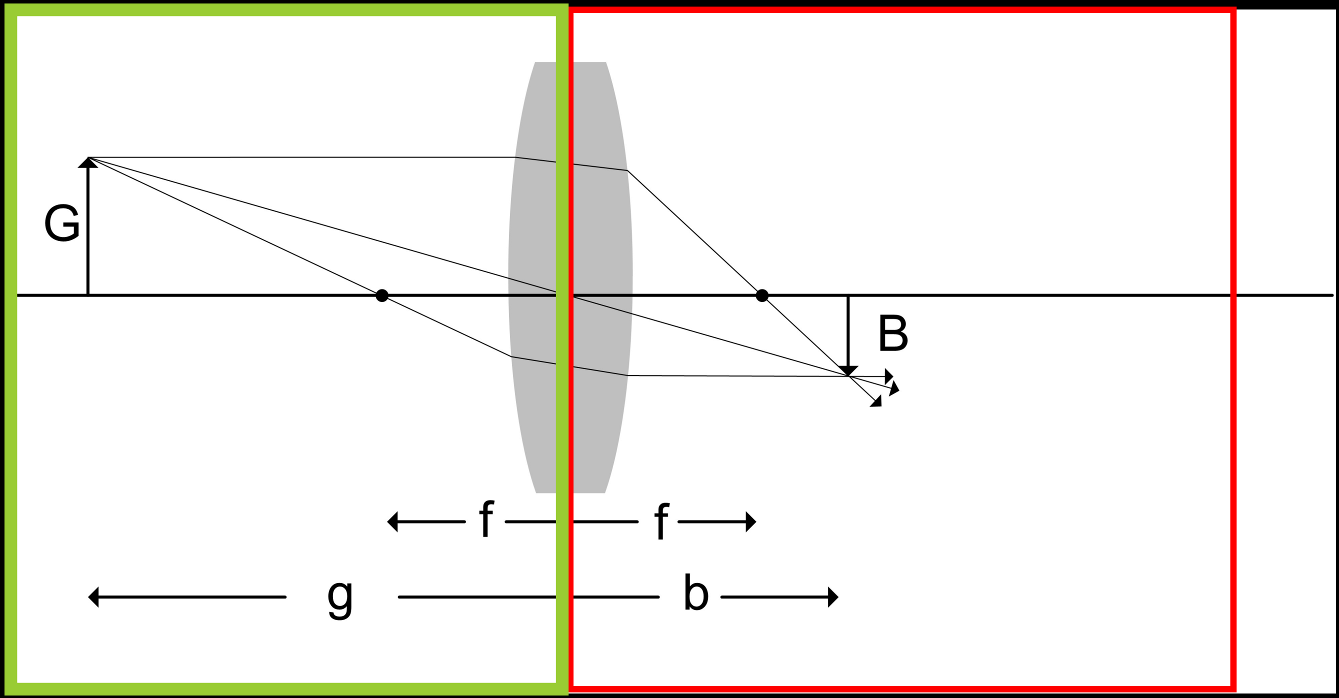
Inside camera

Focal point – focal length

- Light coming from "really far away" can be seen as parallel rays
- Rays intersect at the focal point
- Distance from optical centre O to focal point F is called *focal length f*



Where do non-parallel rays meet?



$$\frac{1}{g} + \frac{1}{b} = \frac{1}{f}$$

Thin lens equation

or

Gauss' lens equation

World

Camera

g – distance to object

b – distance to intersection

Quiz 4: Where do the rays meet

- Camera with focal length of 5 mm
- Rasmus is standing 3 meters away
- Where do the rays meet in the camera? (b)

$$\frac{1}{g} + \frac{1}{b} = \frac{1}{f}$$

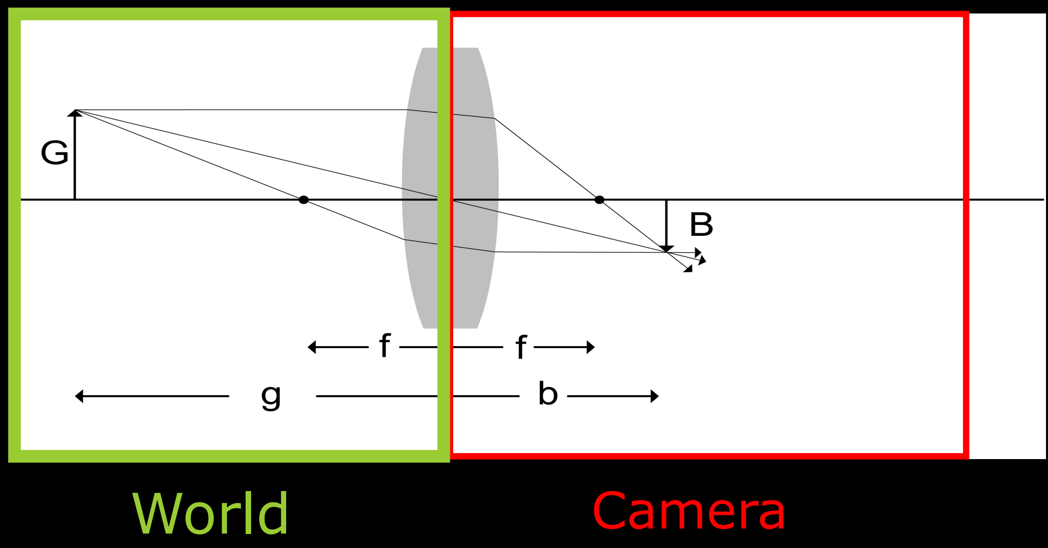
A) b = 1 mm

B) b = 4 mm

C) b = 5 mm

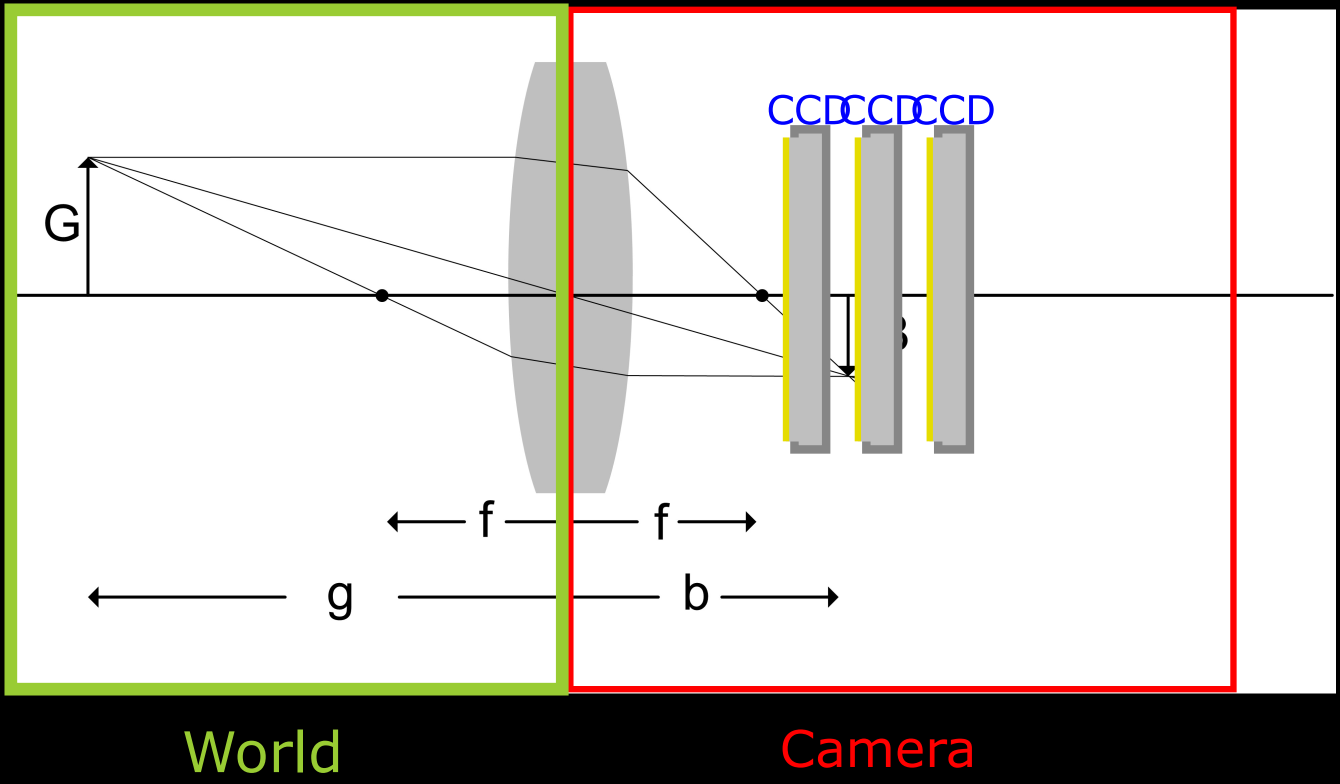
D) b= 6 mm

E) b = 7 mm



Focus or not to focus?

How do we make focused images? Placing the CCD right

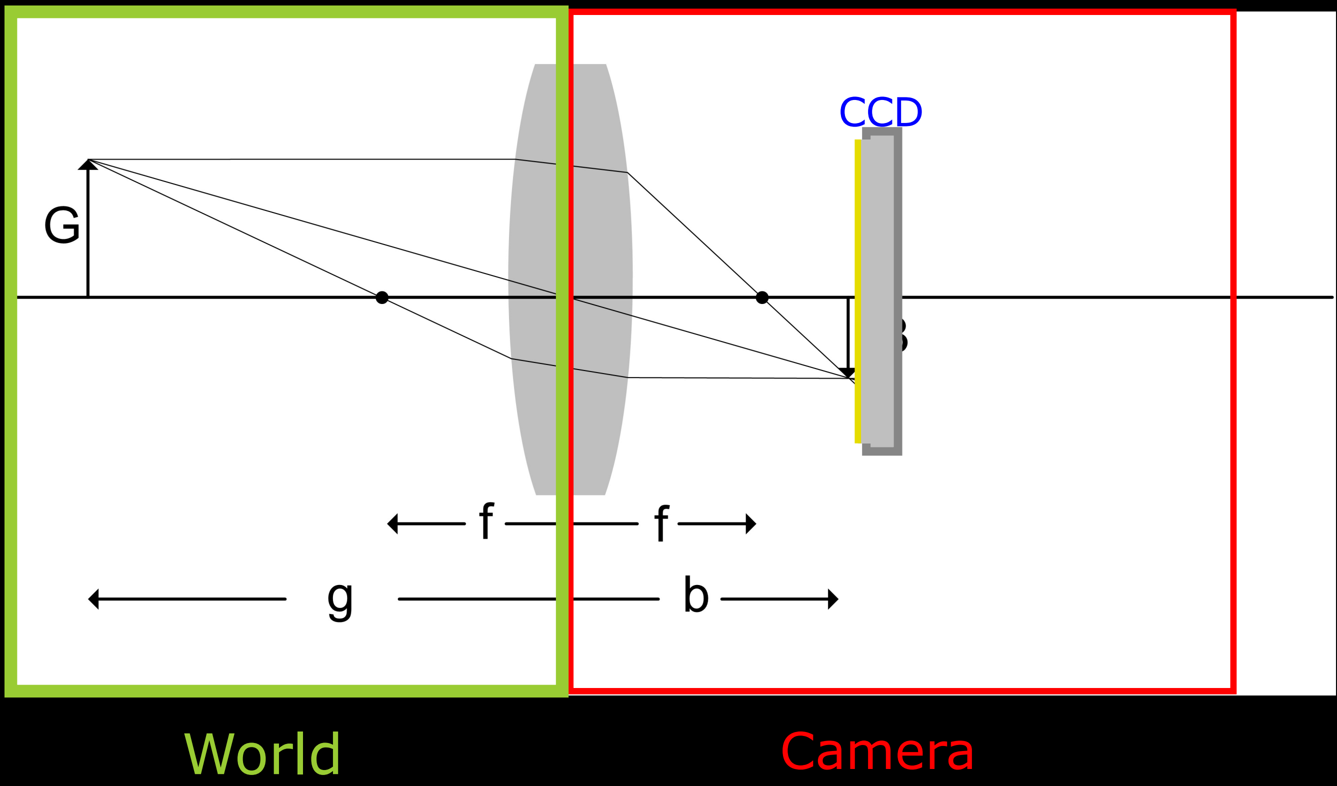


CCD should be placed at b!

g – distance to object

b – distance to intersection

Focusing



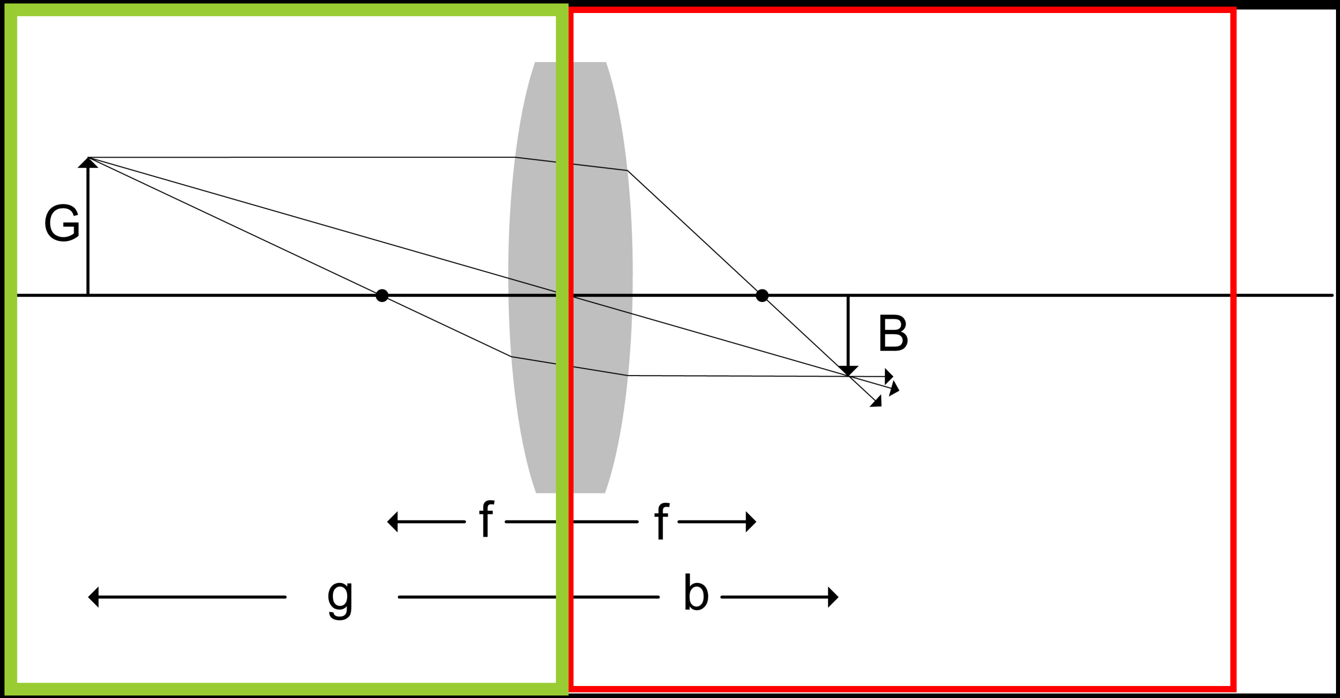
g – distance to object

b – distance to intersection

- We move the camera
- Distance to object (g) changes
- f is fixed
- b changes
- Move CCD to b
  - Focusing

$$\frac{1}{g} + \frac{1}{b} = \frac{1}{f}$$

Object size



What is the size of an object on the CCD?

World

g – distance to object

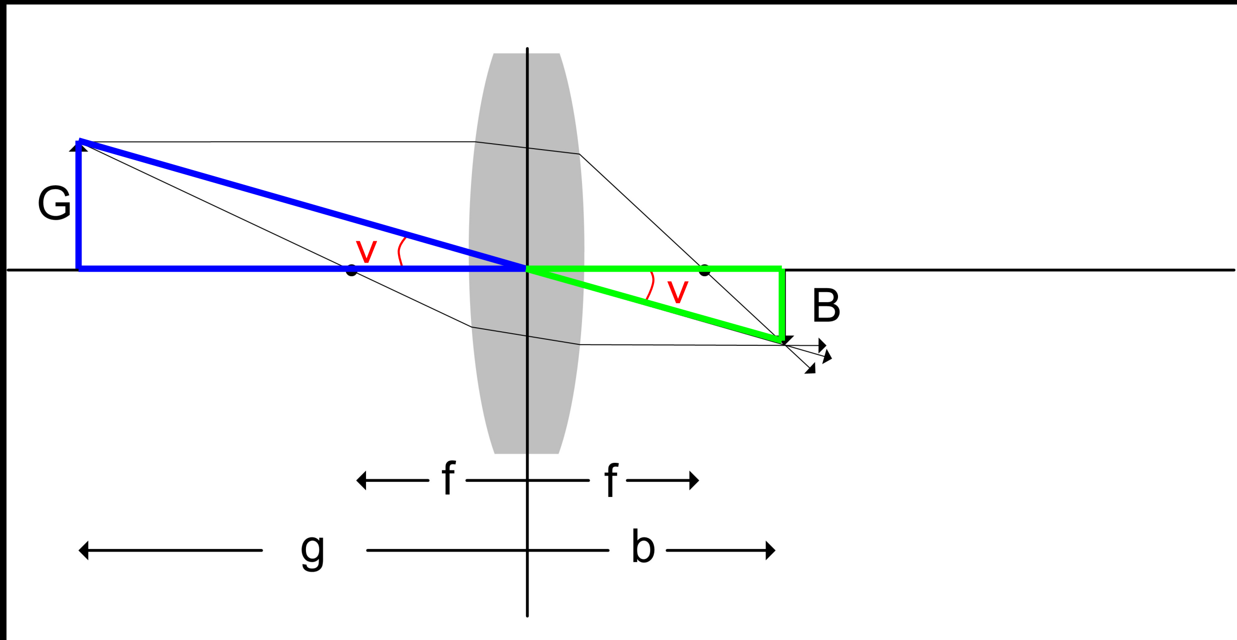
G – Object height

Camera

b – distance to intersection

B – object height on CCD

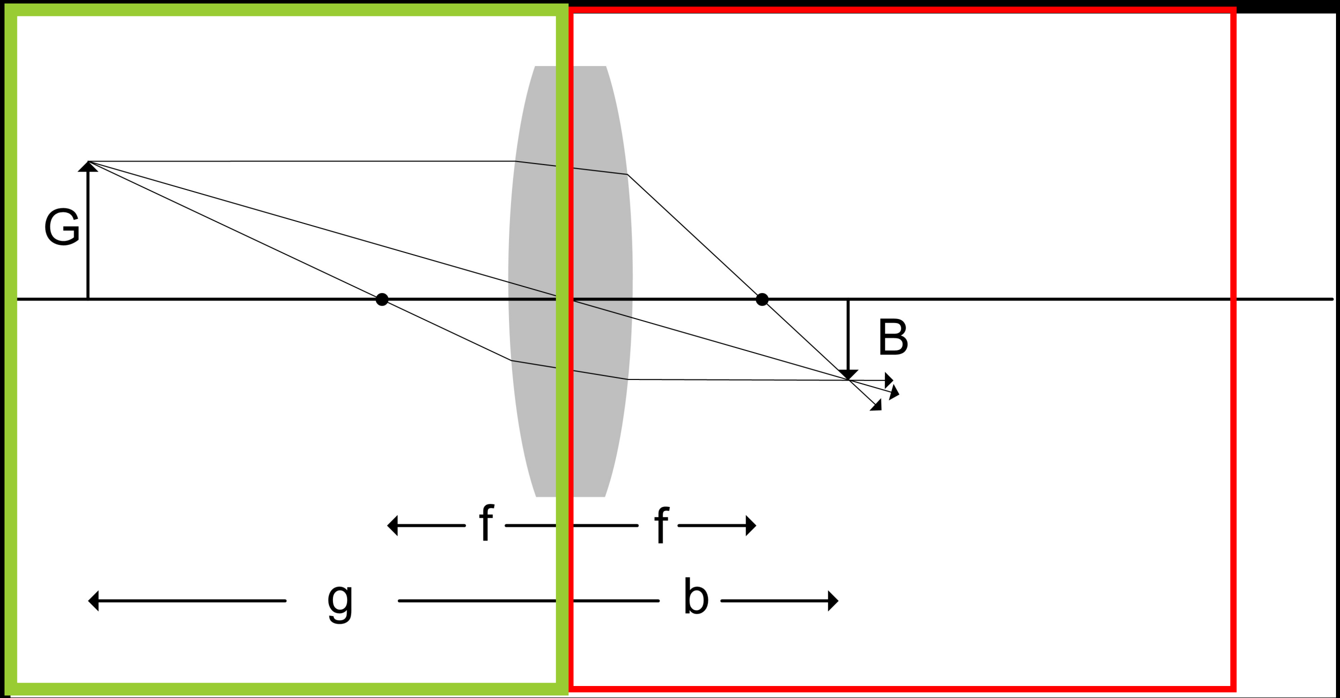
An important relation!



- Two triangles
- One with side length g and one with b
- B and G are related! – how?
- Hint - tangent

$$\frac{b}{B} = \frac{g}{G}$$

An important relation!



$$\frac{b}{B} = \frac{g}{G}$$

World

g – distance to object

G – Object height

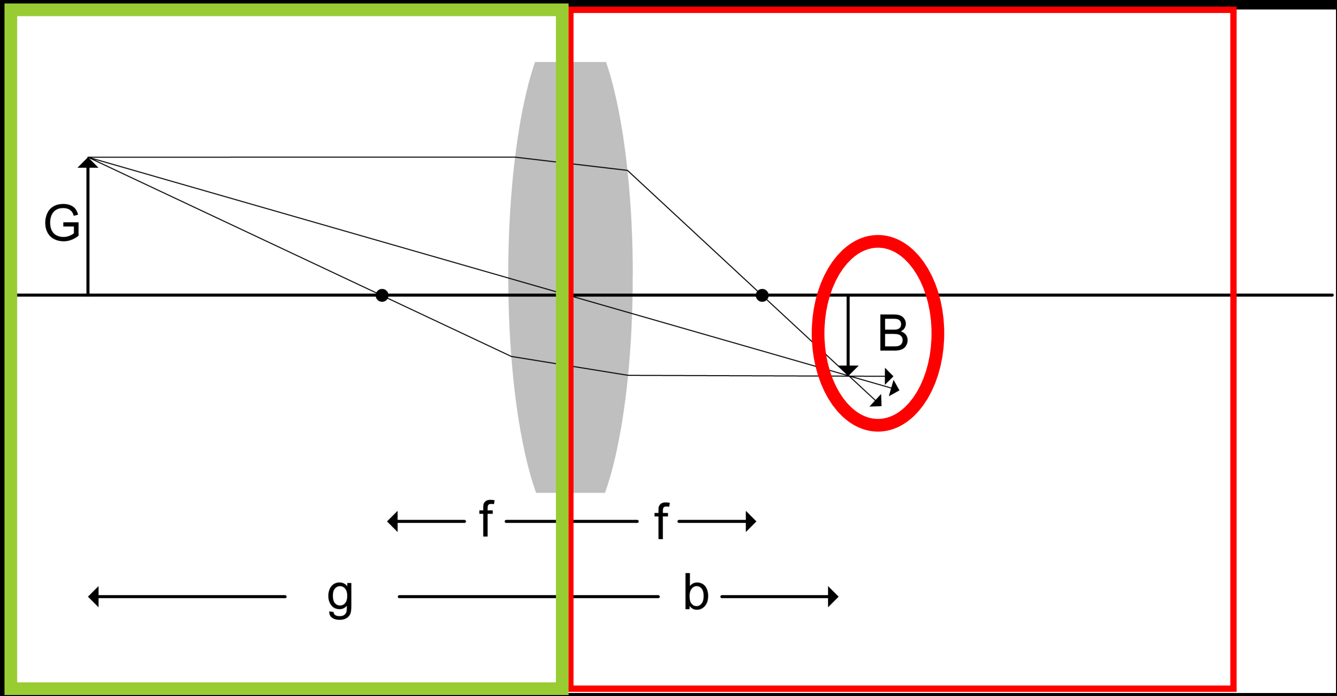
Camera

b – distance to intersection

B – object height on CCD

How do we Zoom ?

We want to make B larger! How?



World

g – distance to object

G – Object height

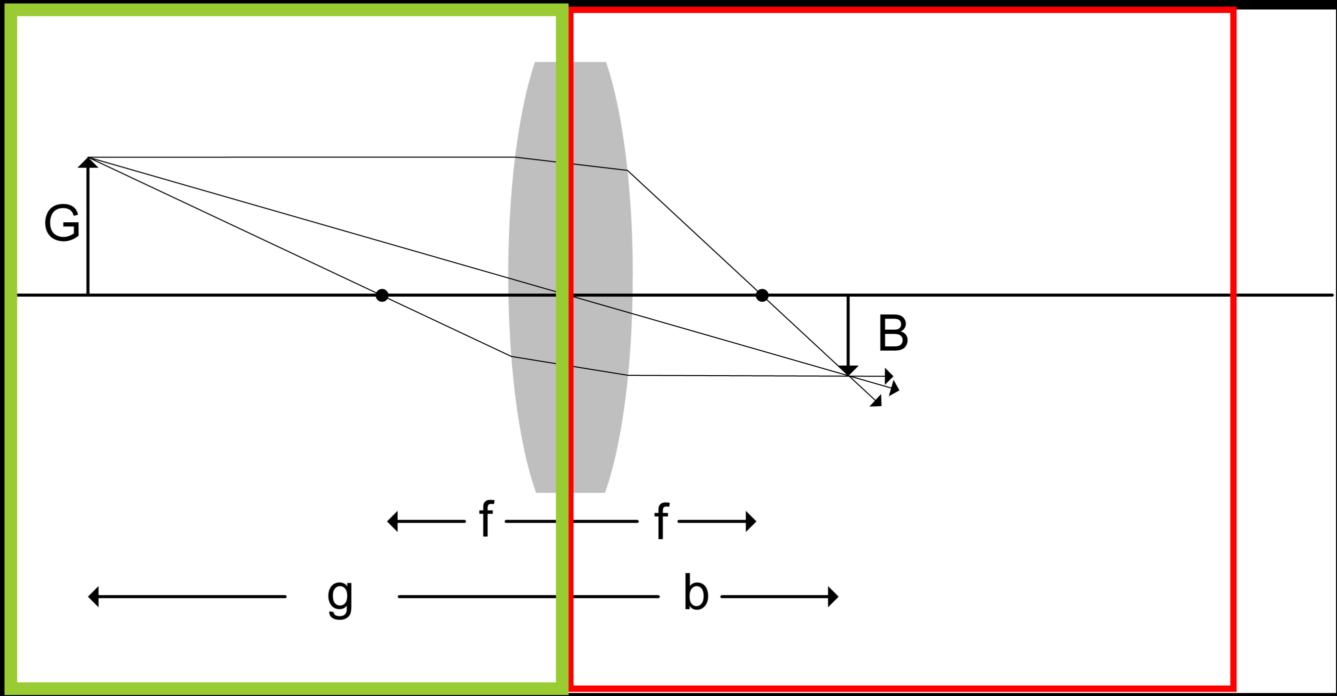
Camera

b – distance to intersection

B – object height on CCD

Zoom

We want to make B larger! How?



$$\frac{b}{B} = \frac{g}{G}$$

$$B = b\frac{G}{g}$$

Fixed

World

g – distance to object

G – Object height

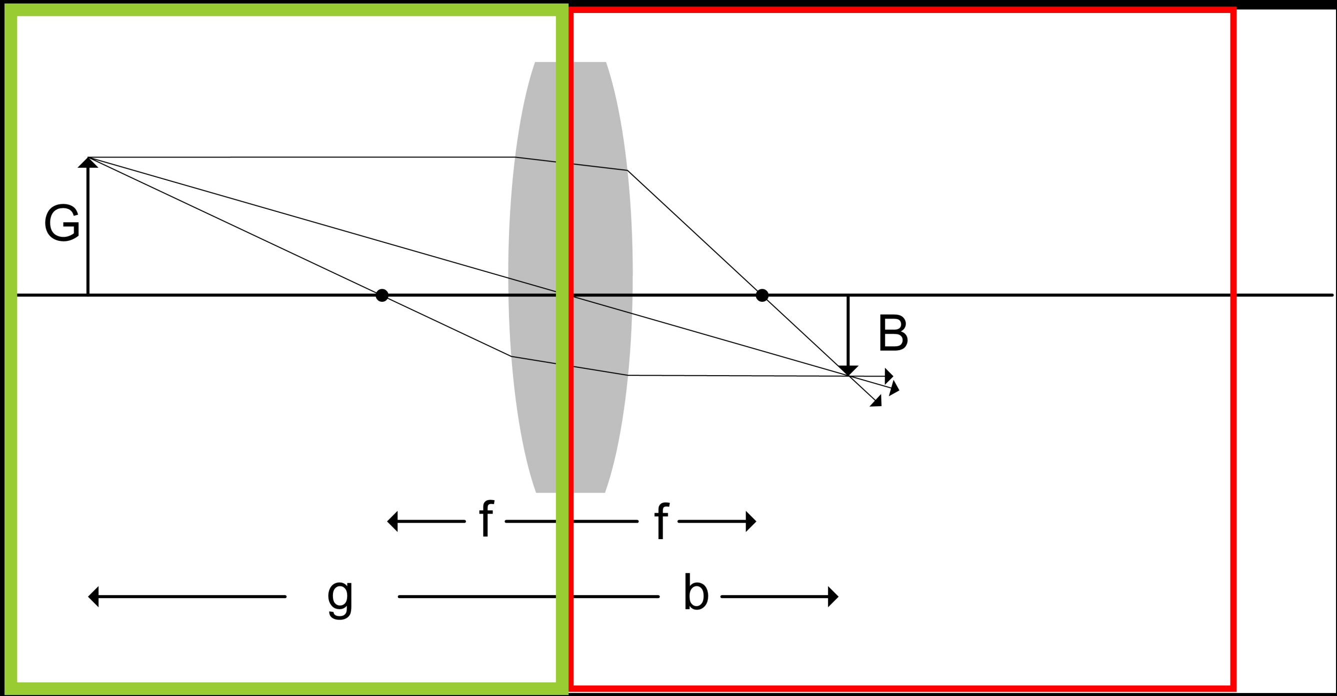
Camera

b – distance to intersection

B – object height on CCD

Zoom

We want to make B larger – changing b!



$$B = b\frac{G}{g}$$

$$\frac{1}{g} + \frac{1}{b} = \frac{1}{f}$$

constant

To change B we change the focal length!

World

g – distance to object

G – Object height

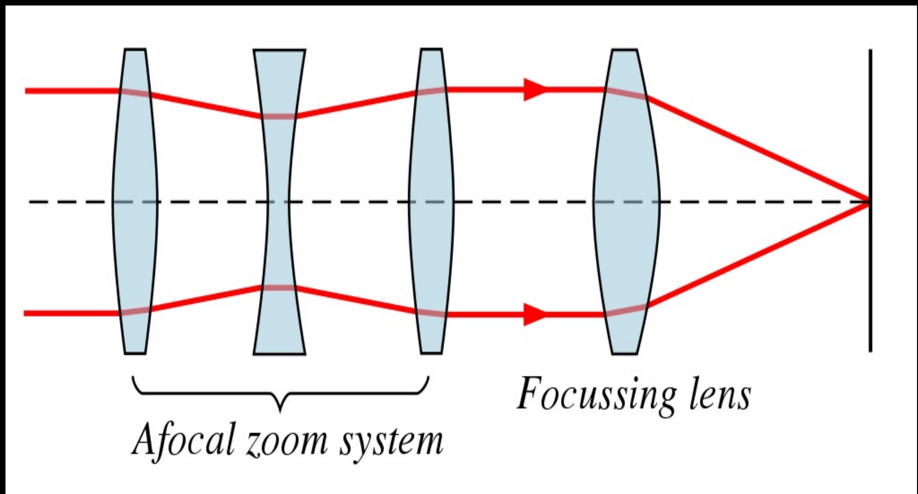
Camera

b – distance to intersection

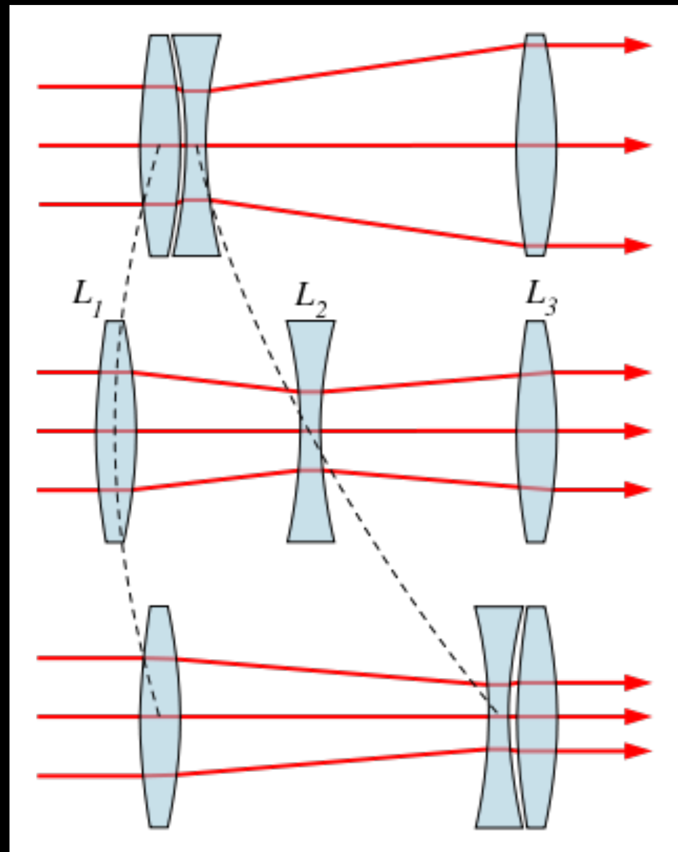
B – object height on CCD

Changing the focal length?

- Not possible on a simple lens
- Need a "zoom lens"
- Several lenses together



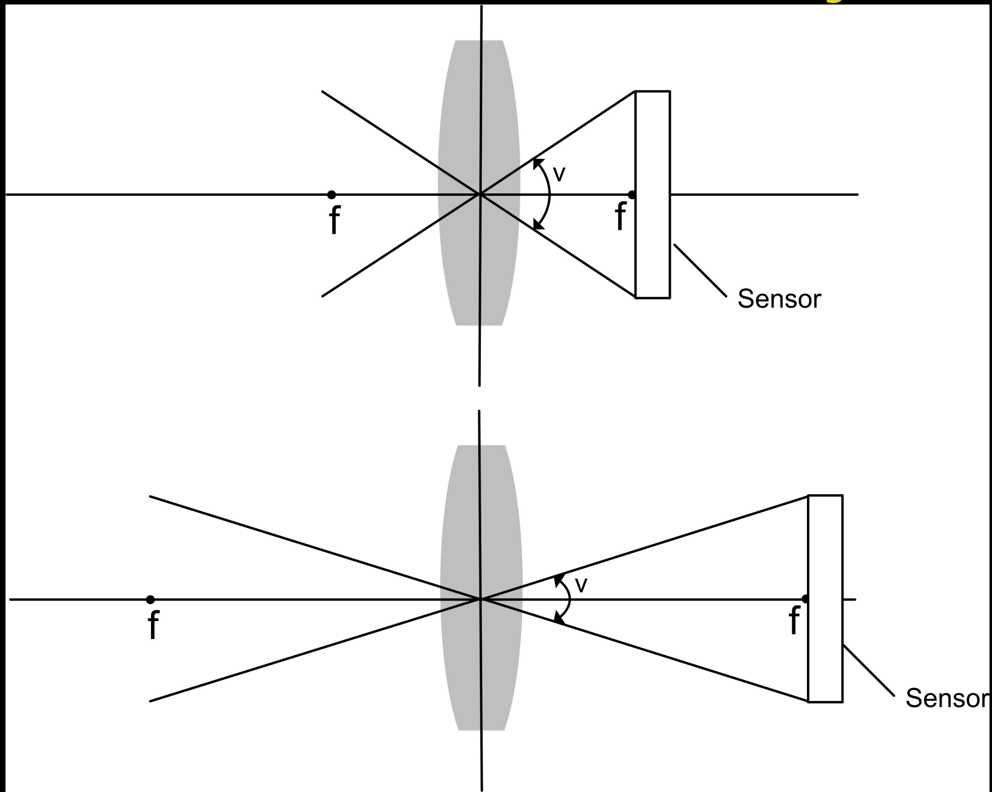
Afocal zoom principle



From WikiPedia: wikipedia.org/wiki/Zoom_lens

Field of view (FOV)

Two cameras with different focal length

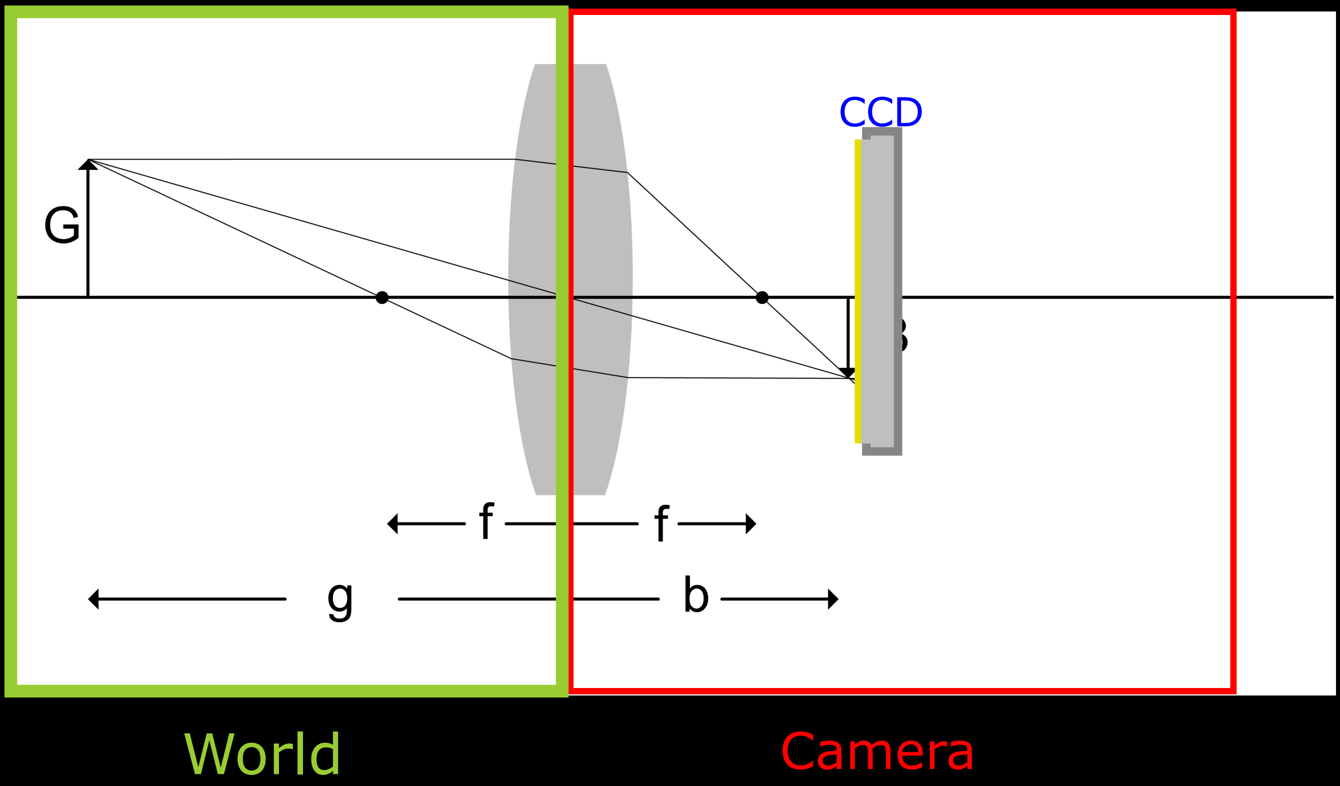


- Described by an angle
  - Large angle the larger FOV
- Depends on
  - CCD size
  - Focal length
- Fisheye lens
  - Small focal length
  - Large field of view

- CCD chip is a rectangle
  - Horizontal field of view
  - Vertical field of view
- Zoom changes field of view
  - Optical zoom
  - Digital zoom

Depth of field - dybdeskarphed

Depth of field

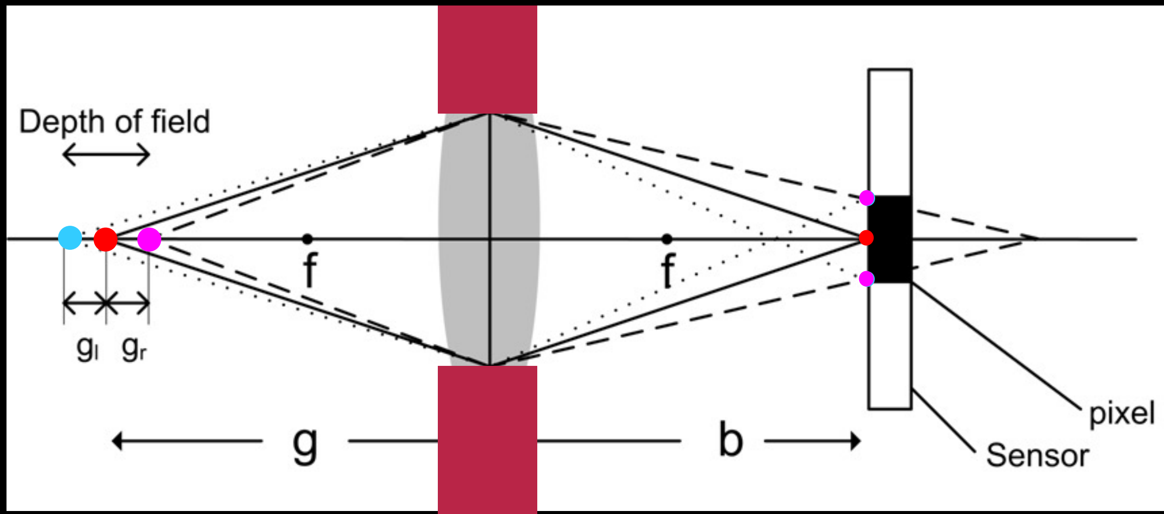


- g is fixed
- CCD should be placed at b
- g is fixed – only focus at one distance!

g – distance to object

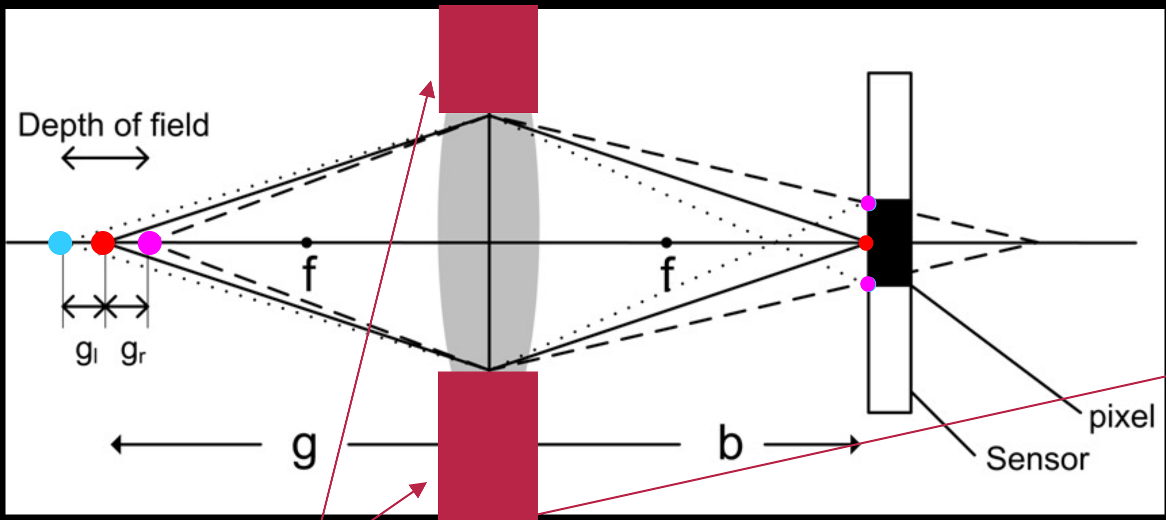
b – distance to intersection

Depth of field



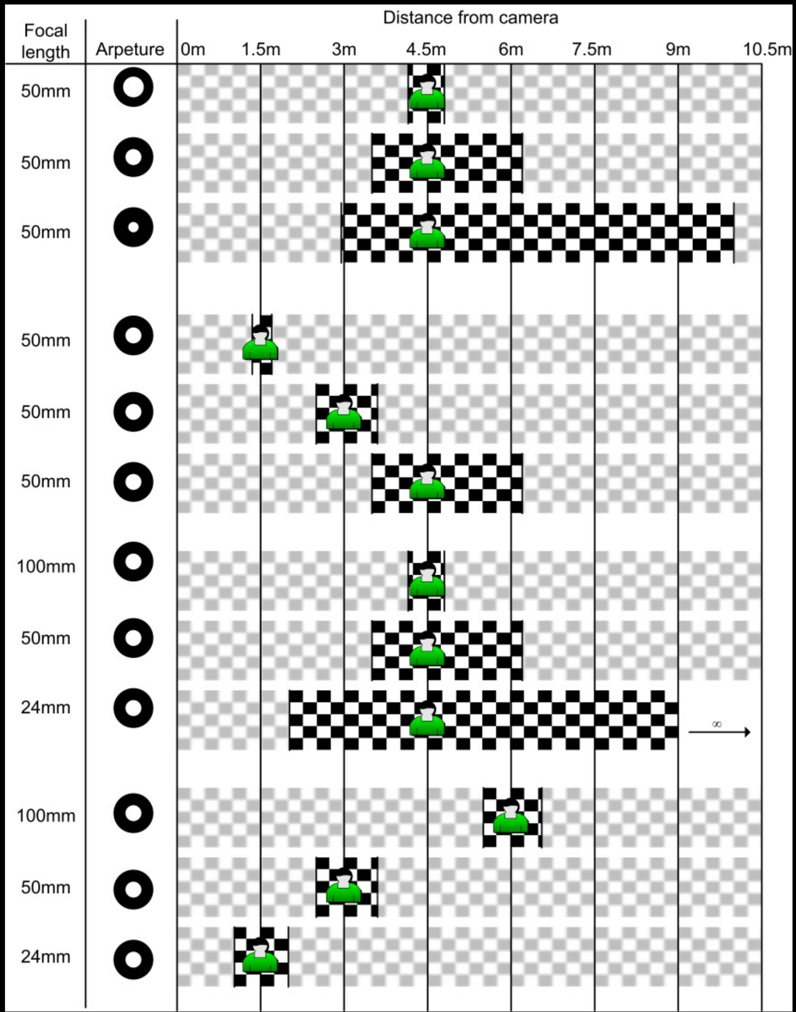
- Look at one pixel in the middle
- The object is placed at distance g
- How much can we move the object?
  - Light has to hit the same pixel
- Move it to the left (g_l)
- move it to the right (g_r) – still hit the same pixel (but twice)

Depth of field – Aperture (blænde)



- The aperture controls the amount of light
- Small aperture
  - large depth of field
  - Less light -> longer exposure

How to acquire a good image?



- Distance to object
- Motion of object
- Zoom
- Focus
- Depth-of-fields
- Focal length
- Shutter
- Field-of-view
- Aperture (DK: blænde)
- Sensor (size and type)

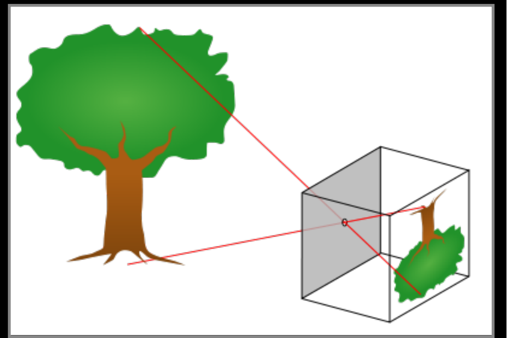
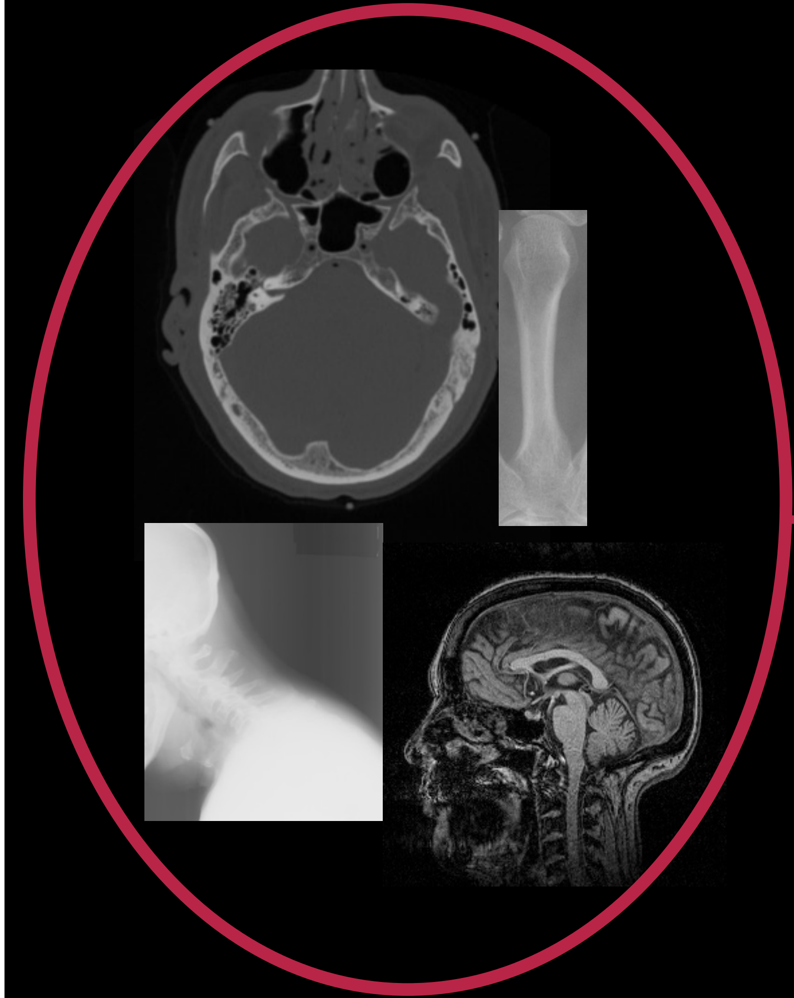


Image storage



Hard disks, memory cards, CDs etc

- Storage for bytes!
  - 500 GB?
  - 500 GigaBytes = 500.000.000.000 bytes!
- A hard disk do not know anything about images
- Stores data as lists of bytes
  - 17, 255, 1, 3, 87, 98, 11, …
- File on a hard disk
  - It has a length (in bytes, MB, GB)
  - Contains numbers! (Bytes)

We want to make an "image file"

Imagine

- You have a telephone. You are only allowed to say no or yes!
- You need to transfer an image to the person in the other end.
- How can we do that?

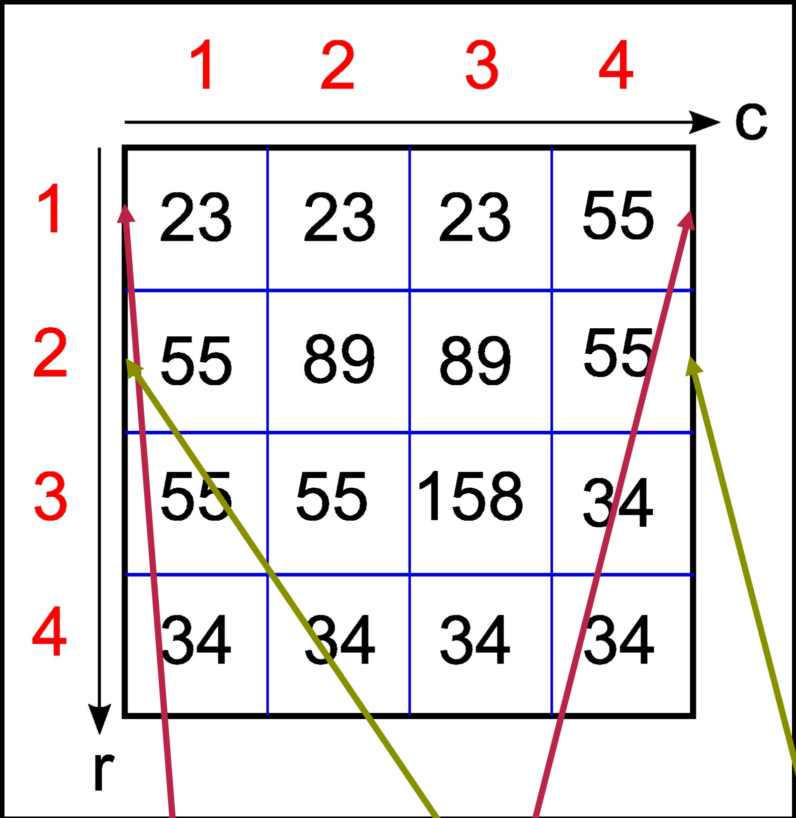
Size: 200 x 200

256 grayscales

Remember that each pixel is a byte

A byte is made out of 8 bit

Image as data



- How do we store this image as list of bytes?
- What do we need
  - Size of the image
    - Width as 2 bytes (0-65535)
    - Height as 2 bytes (0-65535)
  - The data

23,23,23,55,55,89,89,55,55,55,158,34,34,34,34,34

Simple image format

- Stores the image as
  - A header with information about size
  - Data with no compression
- Windows Bitmap Format (BMP)

r \ c	1	2	3	4
1	23	23	23	55
2	55	89	89	55
3	55	55	158	34
4	34	34	34	34

Compression - make something smaller

- Is there a more "compact" way to represent the data below?
- Look for patterns
  - A series of numbers can be represented how?
    - The count and the value
- What is the "count and value" code?
  - Reduced from 16 to 12 values

Run length encoding

23,23,23,55,55,89,89,55,55,55,158,34,34,34,34,34

3,23, 2,55, 2,89, 3,55, 1,158, 5,34

Run length encoding

- Simple but useful data compression
- General – not only for images
- Is also used by the Windows Bitmap Format (BMP)

Quiz 5: Run Length coding of image

A) 1 1 3 5 2 3 3 2 2 3 4 201 4 130 0 147 2 88

B) 1 1 2 5 2 3 2 3 3 201 3 19 5 147 4 130 1 147 2 88

C) 1 1 3 5 2 3 1 2 2 3 2 201 3 19 2 147 4 130 3 147 2 88

D) 5 1 1 5 2 3 3 2 2 3 2 201 3 19 2 147 3 130 1 147 5 88

E) 1 1 3 5 3 3 5 2 2 4 2 201 6 19 2 147 4 130 2 88

1	5	5	5	3
3	2	3	3	201
201	19	19	19	147
147	130	130	130	130
147	147	147	88	88

Compression ratio – how compressed?

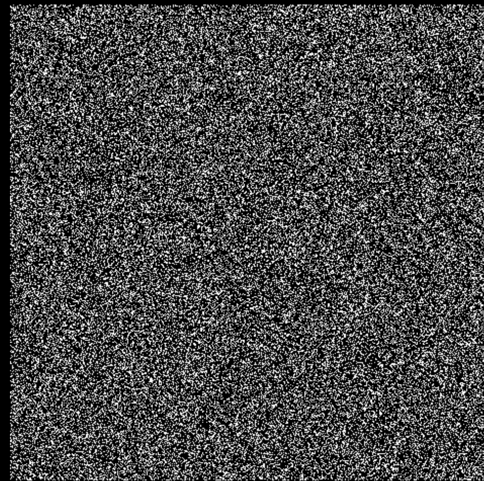
- Gives a measure for how much data is compressed
- Our example
  - From 16 to 12
  - 16 : 12 = 4 : 3
  - Ratio 1.33

Compression ratio = uncompressed size / compressed size

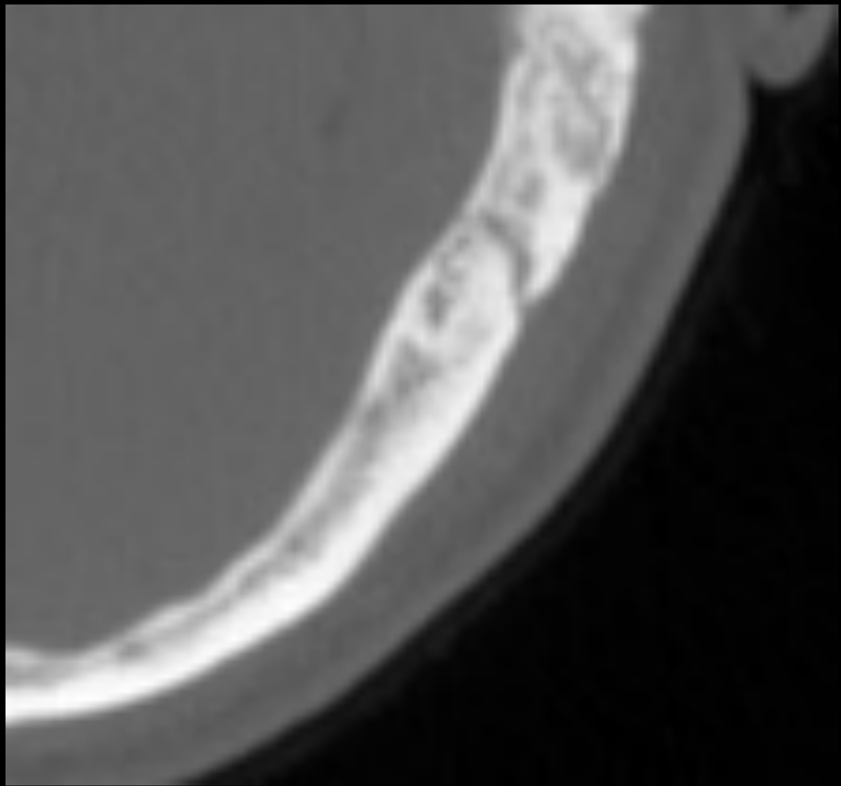
Lossless image formats

- Do not throw away information
- Good for storing medical images
  - We do not want to destroy any information
- Not very effective for photos. Why?
  - To many changes in the image
- PNG (portable network graphics) is a good format

Noisy image

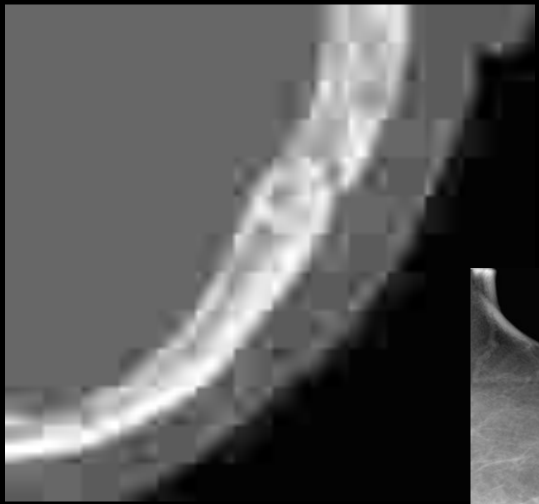


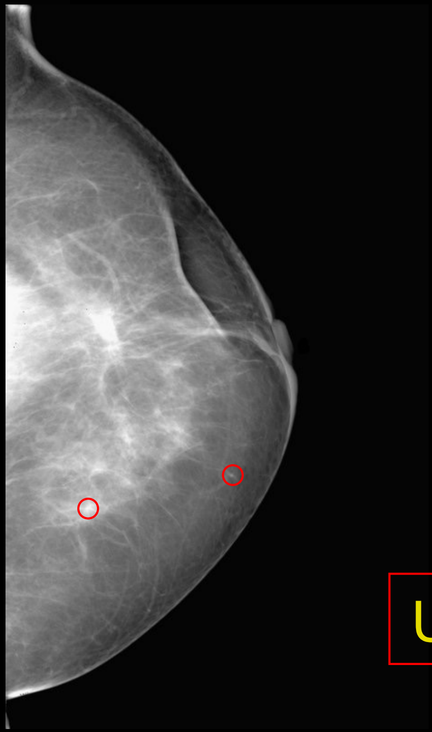
Lossy image formats



- Removes "unimportant" information
- JPEG is an example
- Removes the "high frequencies"
- Similar to the MP3 sound format

Compression artefacts





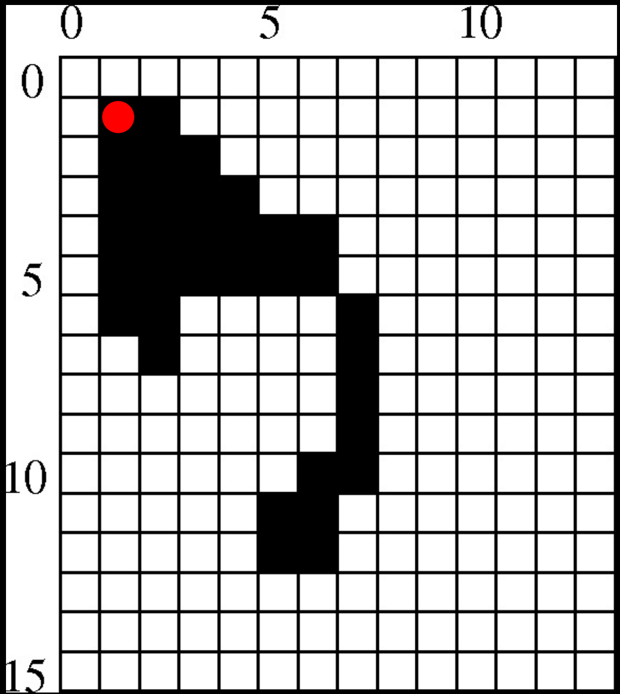
- Lossy compression changes the image
- Normally not a problem for photos
- BIG problem for medical images
- Mammogram
  - Looking for tiny bright spots
  - Would be changed by lossy compression

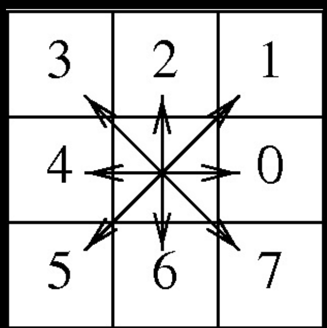
Use JPEG (JPG) for photos only

Binary images

- Binary – means on or off
- Binary image – only two colors
- Background (0 = black)
- Foreground (1 = white)

Chain coding of binary images



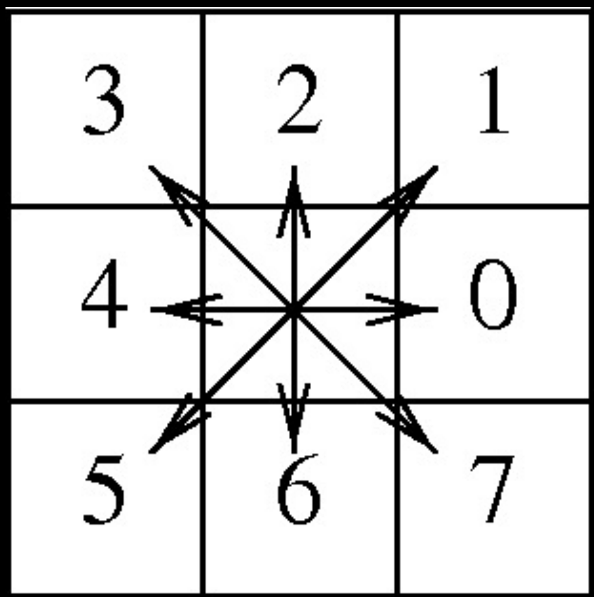


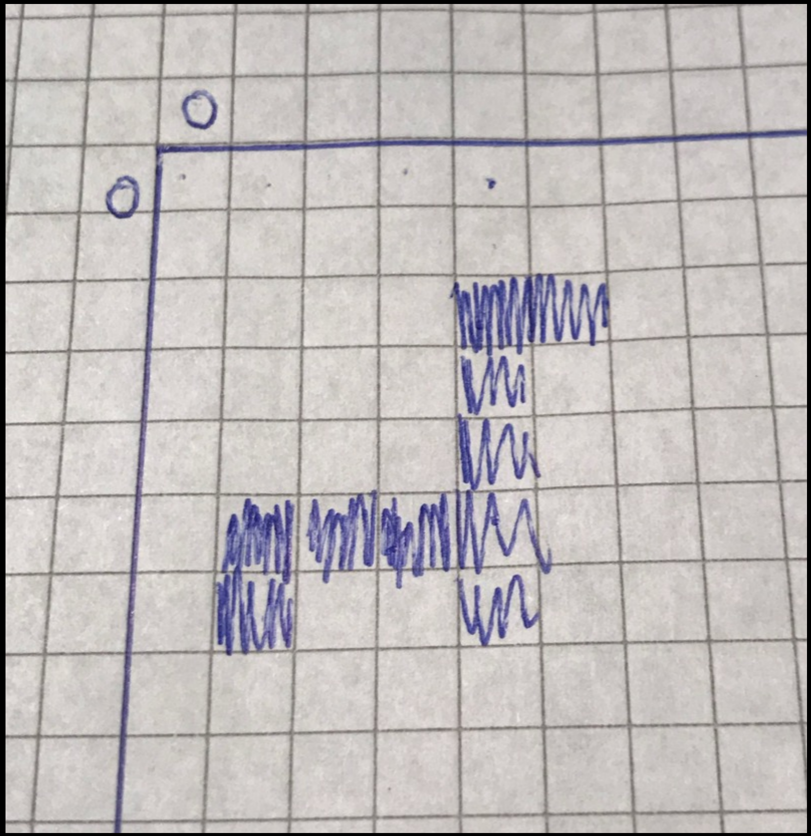
- Sufficient to describe the foreground
- Background given by the foreground
- The coordinates of the starting pixel is stored
- Secondly the sequence of step directions is stored

(1; 1) (0777067666656421122234445632222)

Quiz 6: Chain Coding – what is in the image?

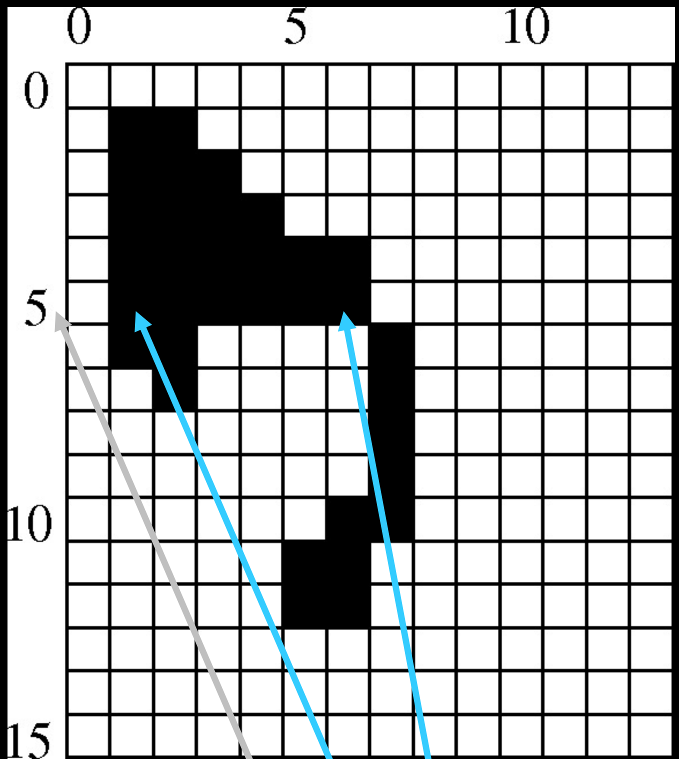
A) House
B) Chain
C) Flower
D) Giraffe
E) Dog
F) Teaport
G) Car
H) Glass
I) Bottle





(4;2)(04666624446)

Binary images – Run length coding



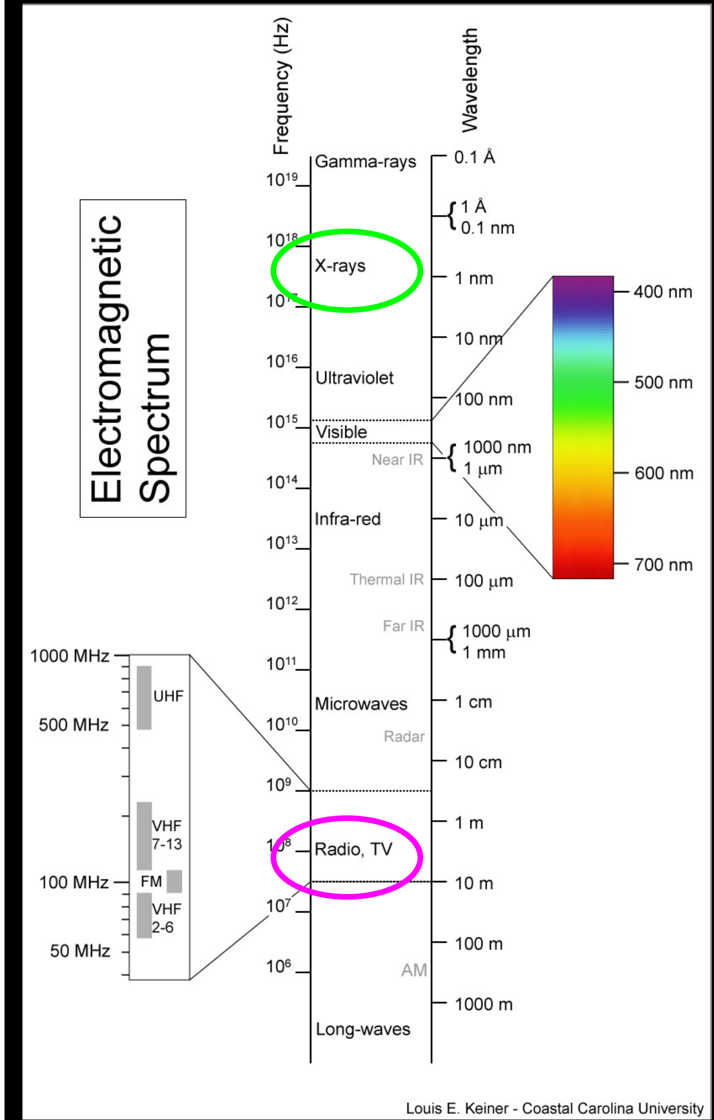
- Another way to represent binary images
- Again the foreground is described
- Each line of the image is described
- For each "run" the row number is stored
- Secondly, the start column and the end column is stored

[1; (1; 2)]; [2; (1; 3)]; [3; (1; 4)]; [4; (1; 6)]

[5; (1; 6)]; [6; (1; 2)(7; 7)]; [7; (2; 2)(7; 7)]; [8; (7; 7)]

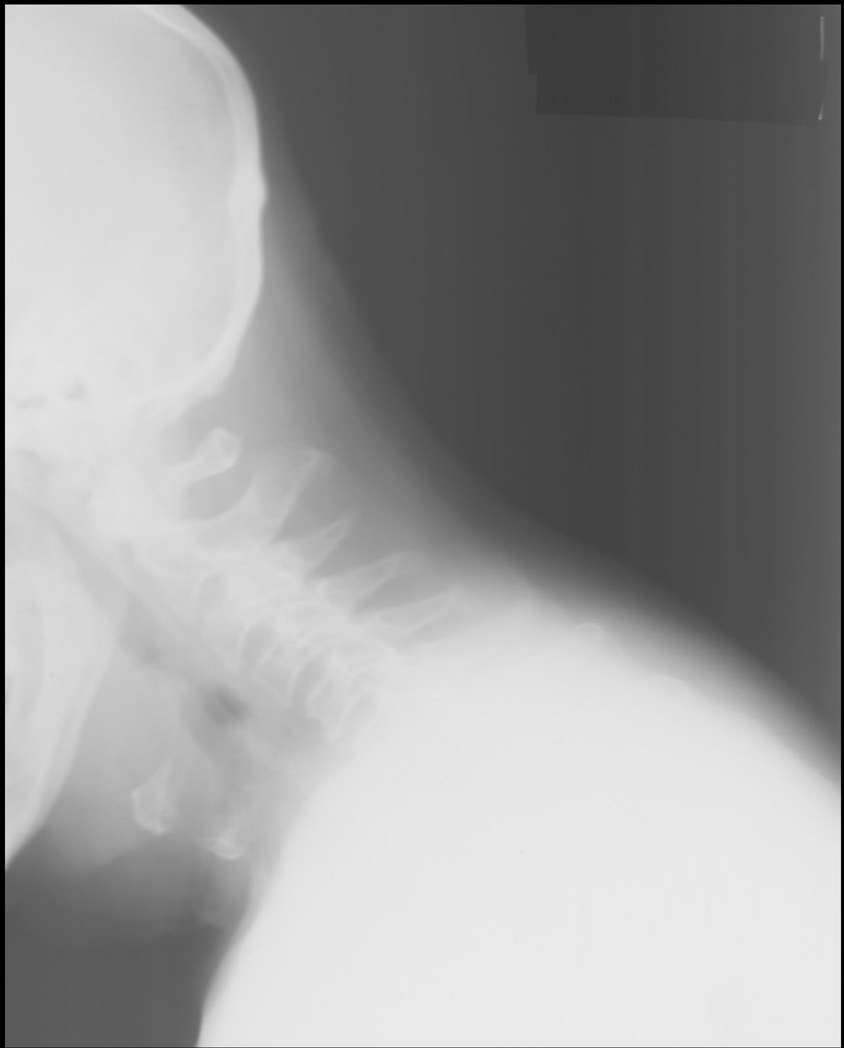
[9; (7; 7)]; [10; (6; 7)]; [11; (5; 6)]; [12; (5; 6)]

Beyond reflective light - To see the invisible



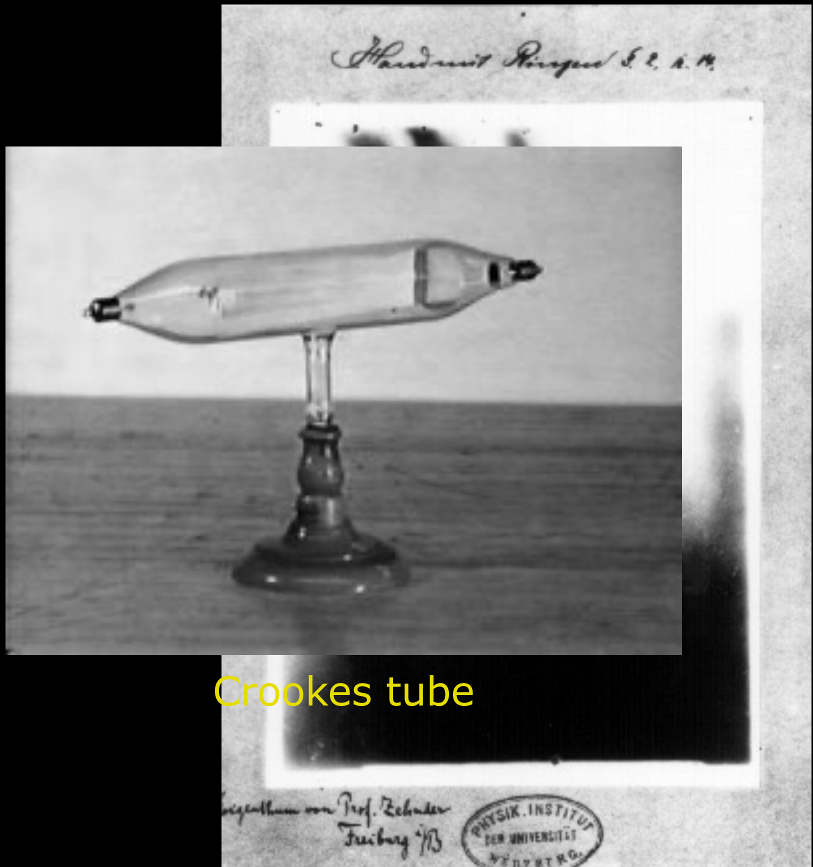
- X-ray imaging
  - High-energy light
  - Computed Tomography (CT) - Medical scanner (Hard tissue)
  - Synchrotron light – high Brilliance x-ray - nano-scope (soft and hard tissue)
- Magnetic Resonance Imaging (MRI)
  - Radio frequency
  - Medical imaging (soft tissue)

X-ray imaging



- The most used form of medical imaging
- Simple
- Cheap
- Fast
- Radiation

Wilhelm Conrad Röntgen

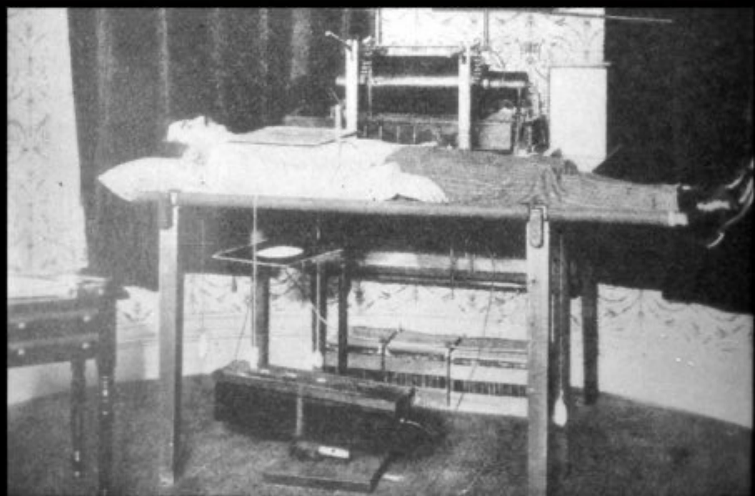


Crookes tube

- German physics professor
- Experimented with a *Crookes tube*
- Discovered that an unknown ray could be captured on photographic plates
- Named them X-rays
  - Other call them Röntgen-rays
- Had no idea they were dangerous
- Made an X-ray of his wife's hand
  - First medical X-ray

Wilhelm Röntgen's first *medical* X-ray, of his wife's hand, taken on 22 December 1895

Quick popularity



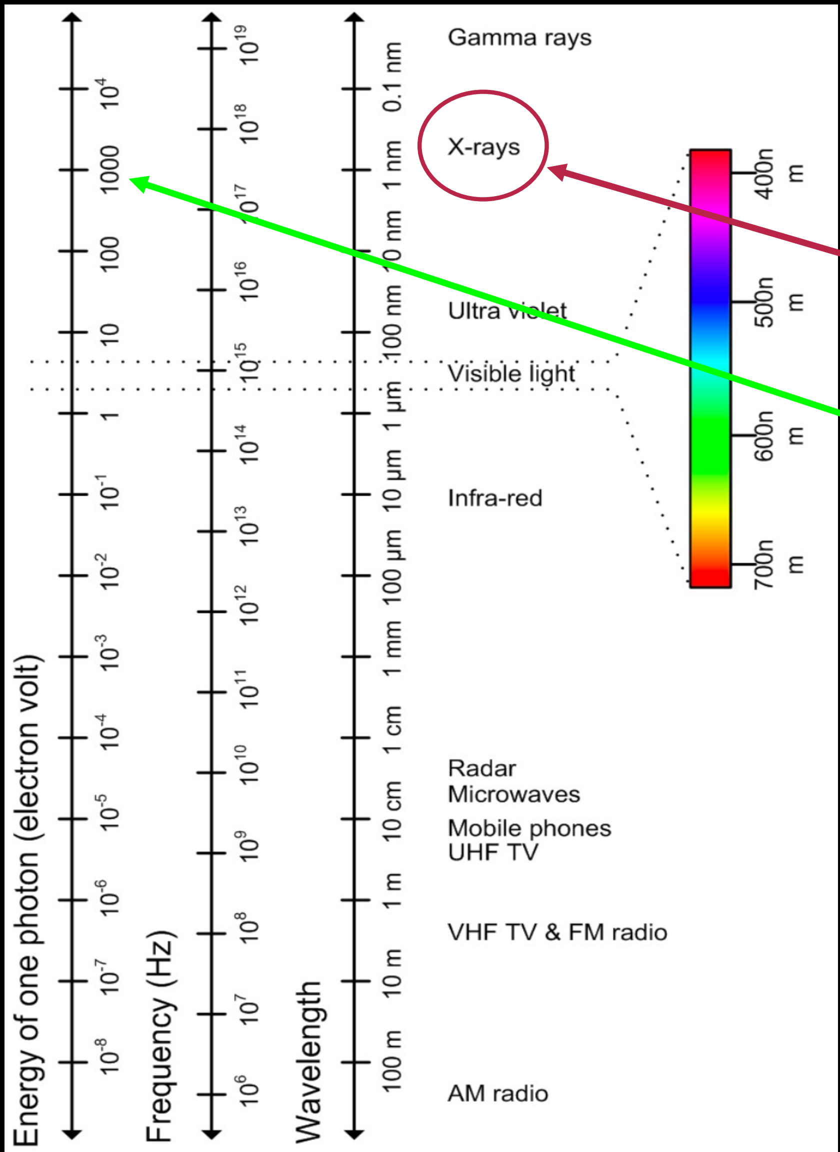
- X-ray became popular extremely fast
  - Shoe fitting
  - Examine your bones in coin machines
  - Wedding pictures
- X-ray clinics in small normal apartments

Dangers

Hands of X-ray pioneer
Mihran Kassabian

- People started to realise that exposure to X-rays could be dangerous

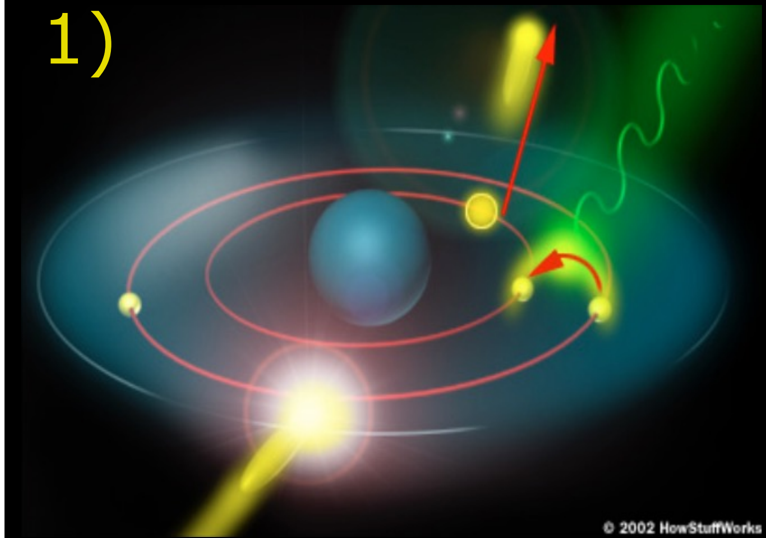
Electromagnetic spectrum



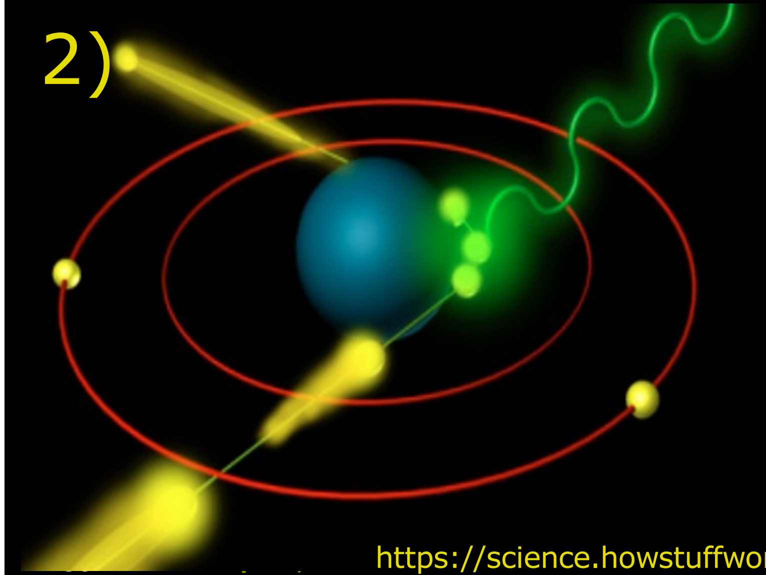
- Wavelength
  - 10 pm < λ < 10 nm
- pm = picometer = 1×10^{-12} m
- nm = nanometer = 1×10^{-9} m
- Small wavelength = high energy

How to generate light – emitting photons

1)



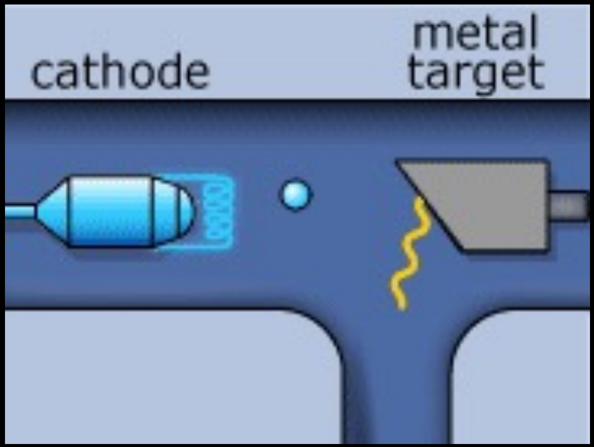
2)

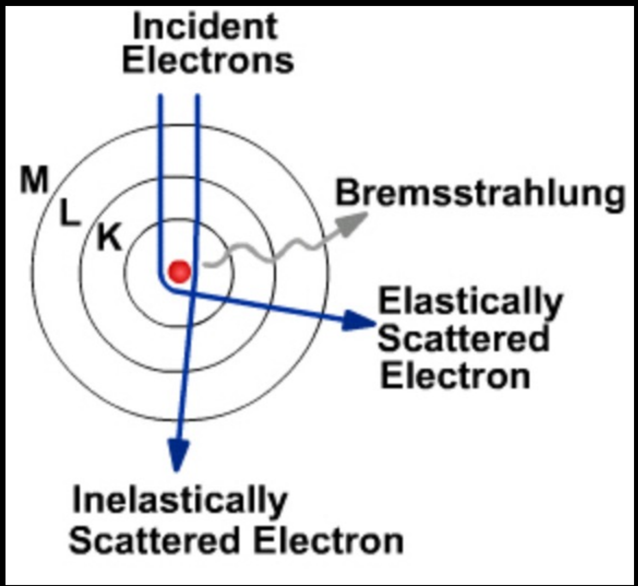


- A) Make a beam of accelerated electrons:
  - Close to the speed of light
  - High energies: kilo electron volts (KeV)
- B) Emitting photons:
  - 1) Incoming electron excite the atom:
    - Electron jump to higher energy level. Fall back to original energy level - release energy→emit a photon
  - 2) De-acceleration of electron → emit a photon

https://science.howstuffworks.com/x-ray2.htm

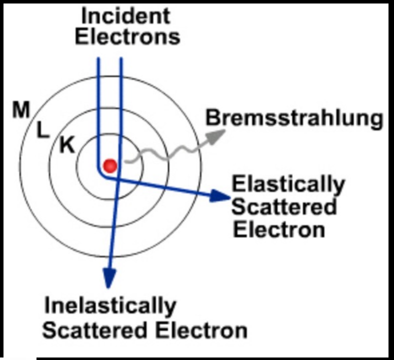
Production of X-rays

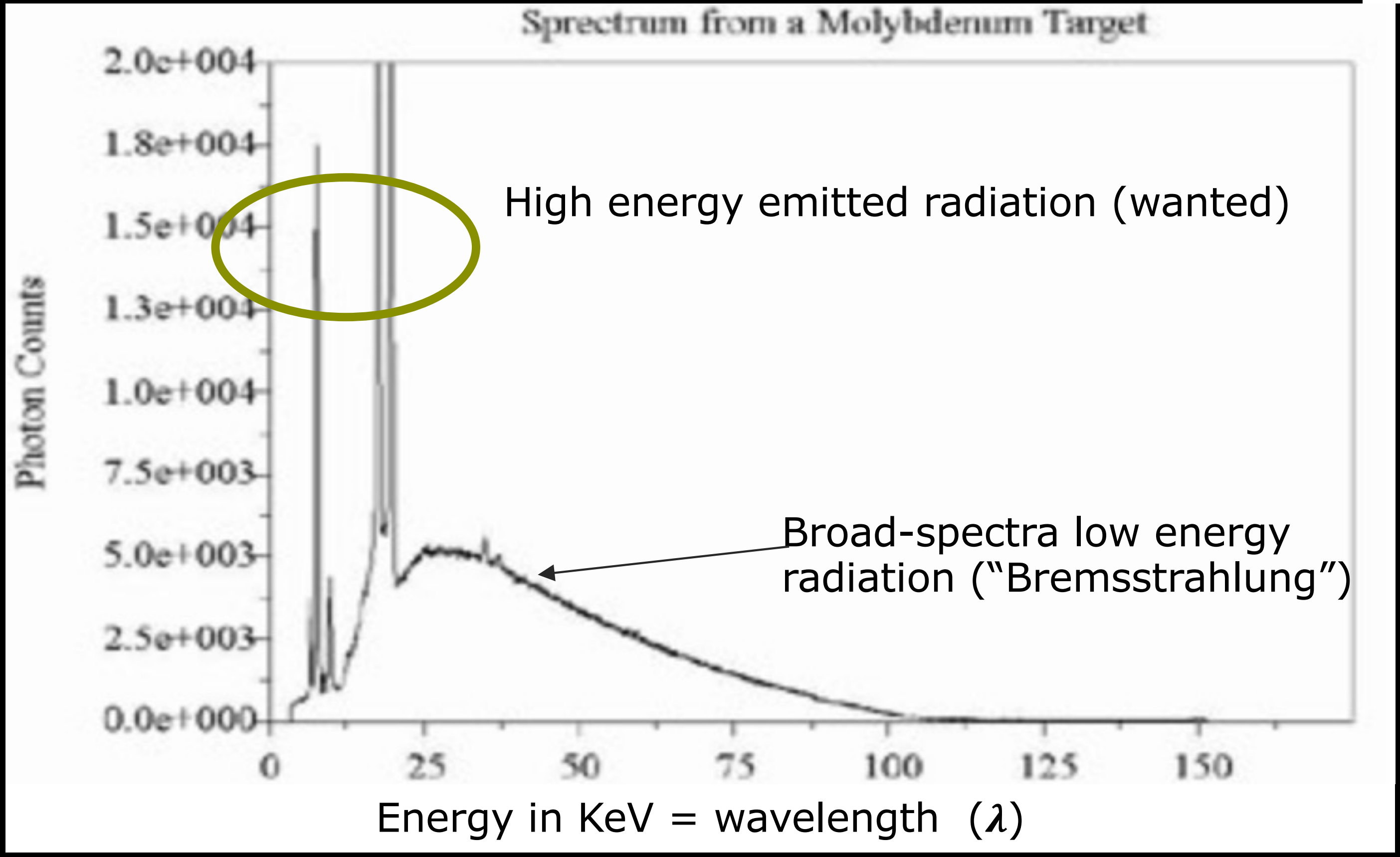




- Electrons are accelerated using a cathode
  - Heating up a filament release more electrons
- Some hit the anode (heavy metal target)
- Slowed down in the anode material
  - Generating heat
  - A small part of the energy is transformed to X-rays
- The electron comes very close to the nucleus
  - Electromagnetic interaction causes a deviation of the trajectory
  - The electron looses energy and an X-ray photon is emitted.

Production of X-rays

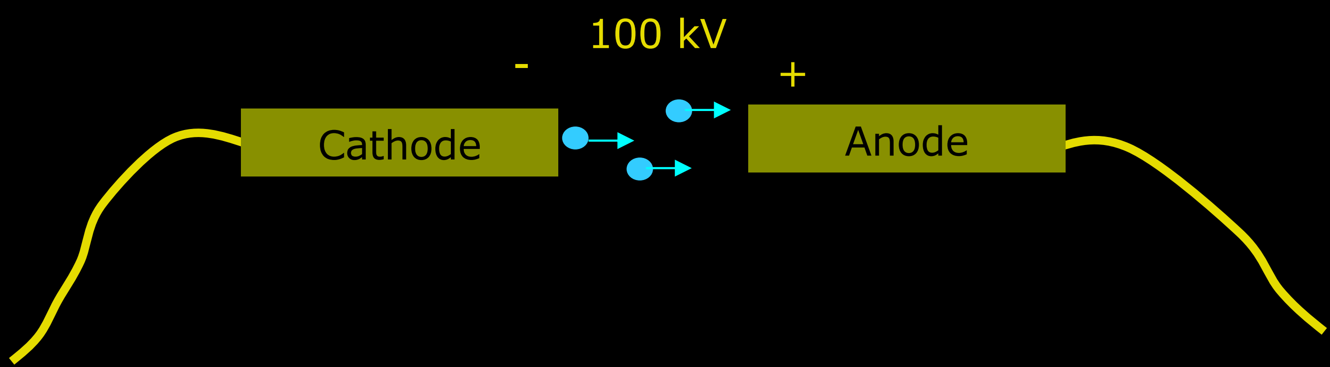




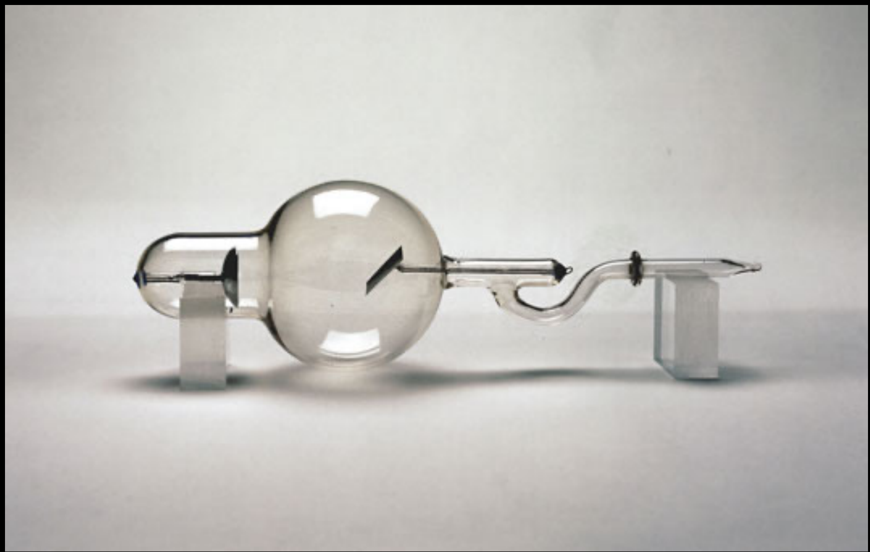
Energy of radiation vs the wavelength: $E = h/\lambda$ [eV]; h: Planck constant

Electron volts

- 1 eV is the energy increase that an electron experiences, when accelerated over a potential difference of 1 V.
- In medical imaging
  - 20 keV < E < 150 keV
- keV = kilo-electron-volts



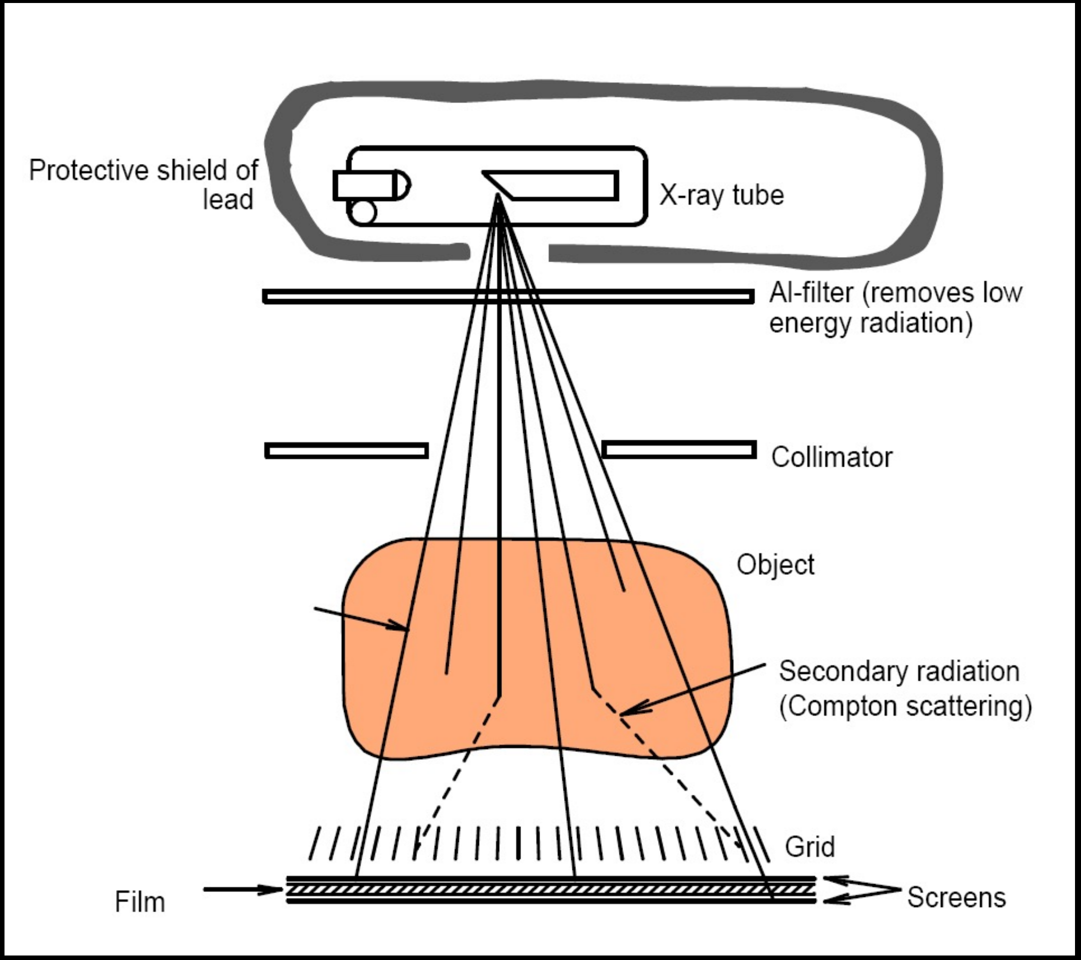
X-ray tube



Jackson X-ray tube, 1896.

Modern rotating anode tube

Full X-ray system

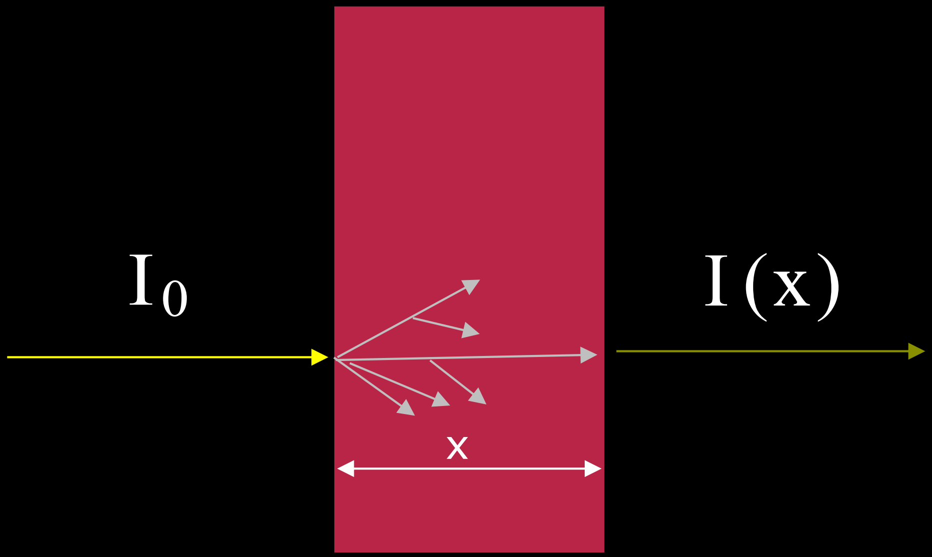


Contrast in X-ray images

Scanned X-ray film

- Some materials absorb more X-rays than others
- We see the X-rays that "got through"
- Dark area – high radiation
  - Air
  - Soft-tissue
  - Fat
- Bright area – low radiation
  - Metals
  - Bone

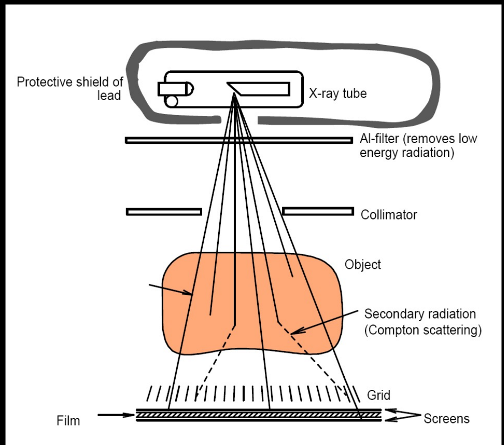
X-ray attenuation



$$I(x) = I_0 \boxed{\exp(-\mu x)}$$

- X-rays hits an object and travels through it
- I_0 is the intensity at the entrance
- $I(x)$ is what is left on the other side
  - after a length of x
- The rest disappears in several different ways
- Computed using Lambert-Beer's law: Different materials have different attenuation coefficients

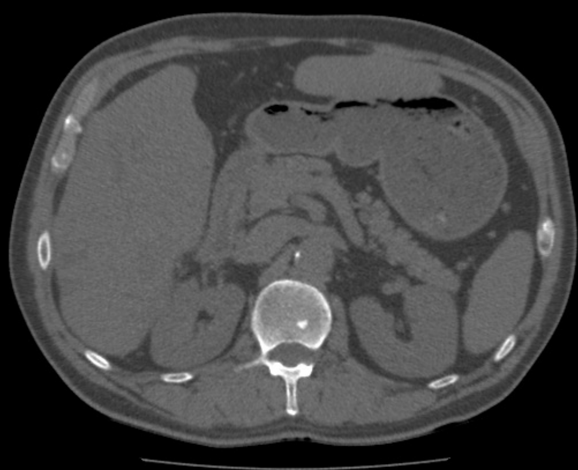
Computed Tomography (CT) scanning

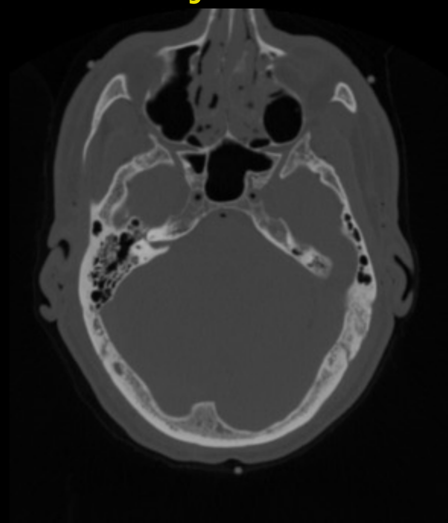


Scan single slice of whole object

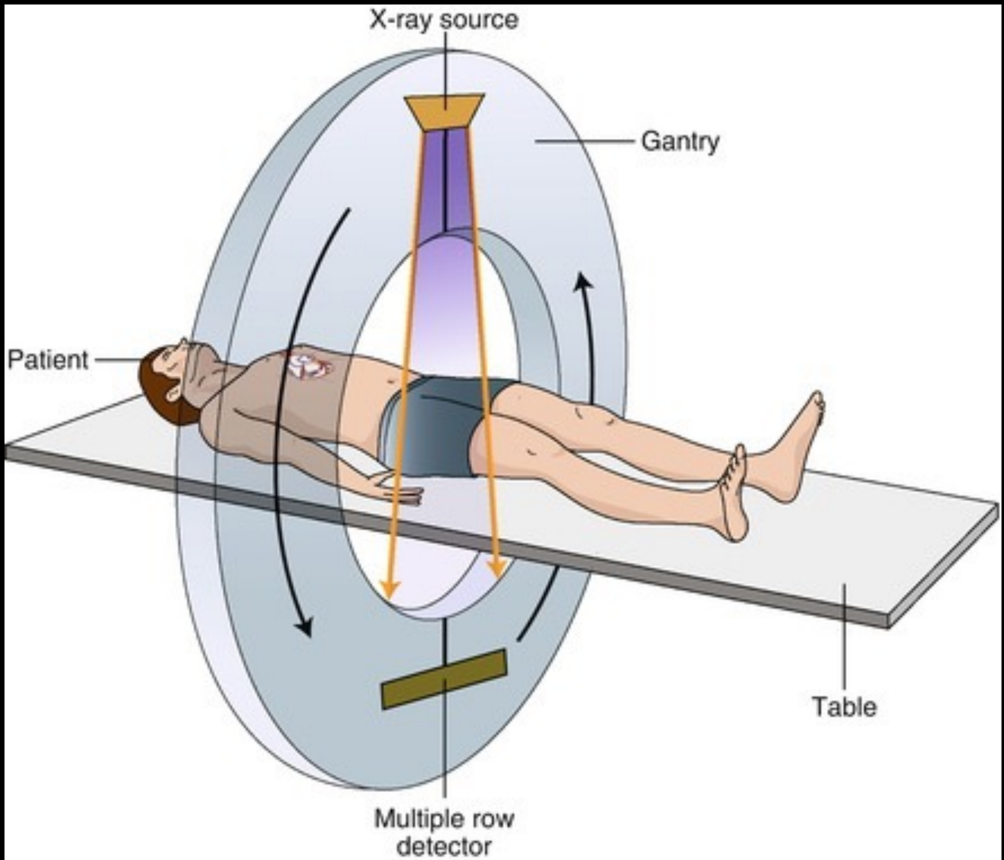
CT scanner (+multi-slice)

Scan single slice within an object





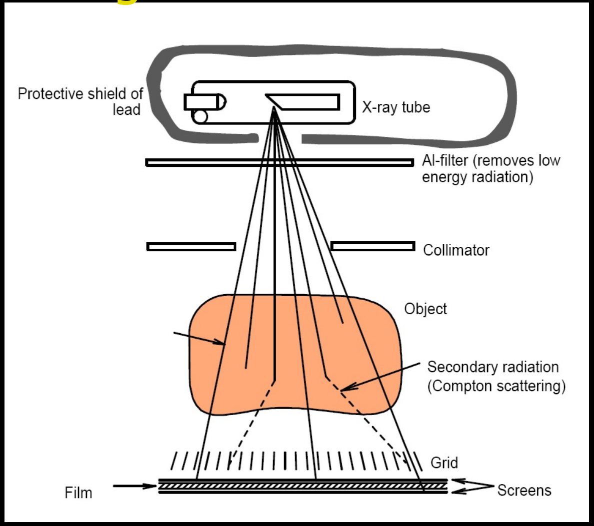
CT scanning - imaging by sections

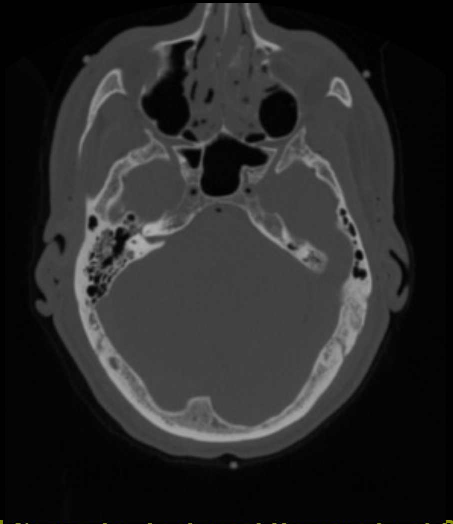


- 512 x 512 pixels per slice
- Many projections
- Image reconstruction
  - Enormous system of equations
  - Find each pixel attenuation coefficient (μ)
  - Hounsfield Units (HU) Calibrate units in medical imaging:
    - Air:1000
    - Water:0
- Not solvable by direct methods

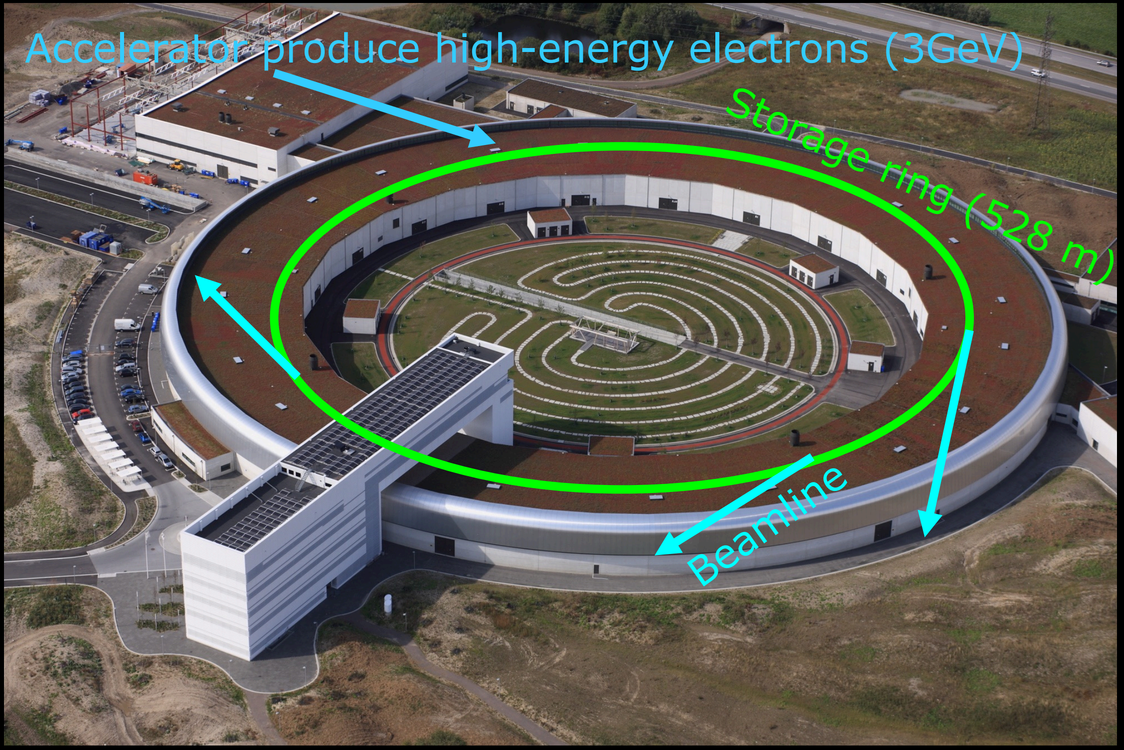
Synchrotron light: The brilliance of x-ray

Inhomogeneous electron beam





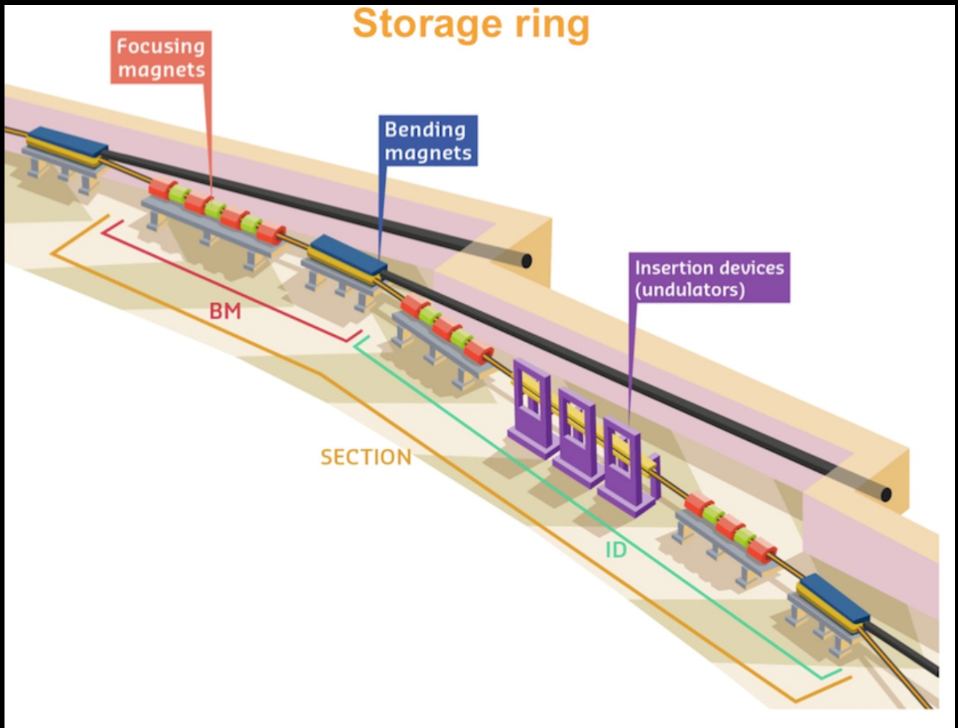
Very homogeneous electron beam

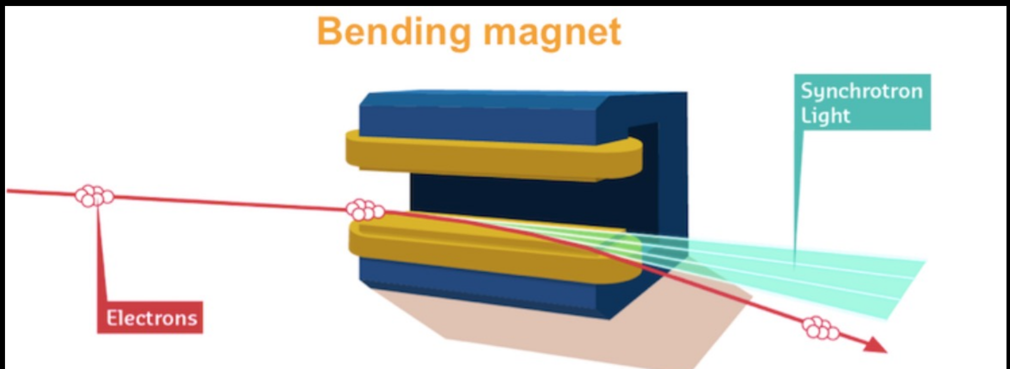


Large scale Research facility:

- MAXIV in Lund, Sweden

Synchrotron radiation: The principle

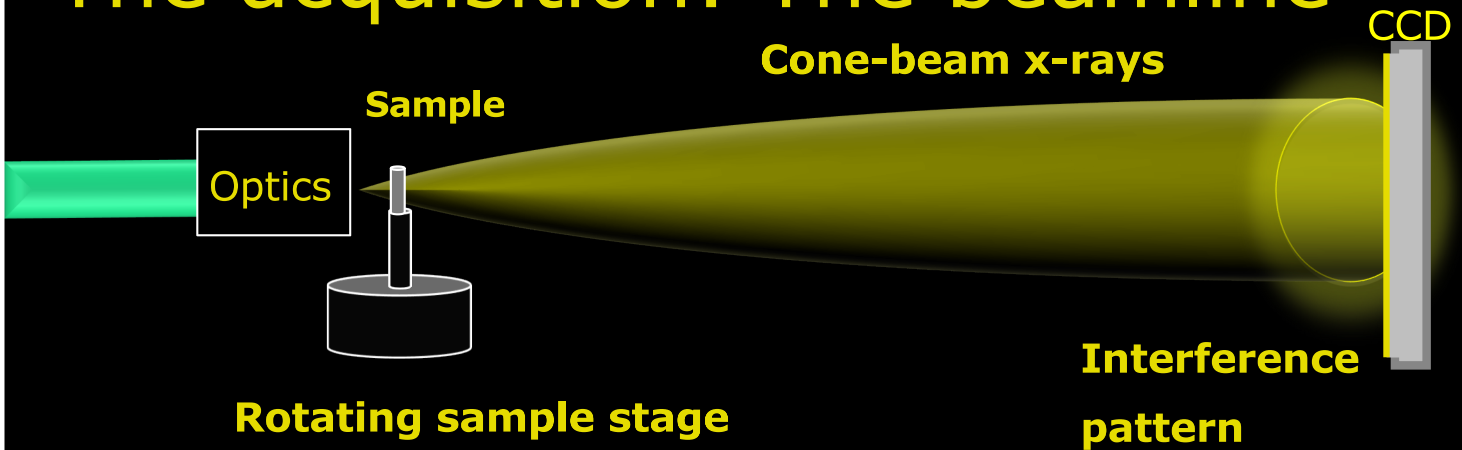


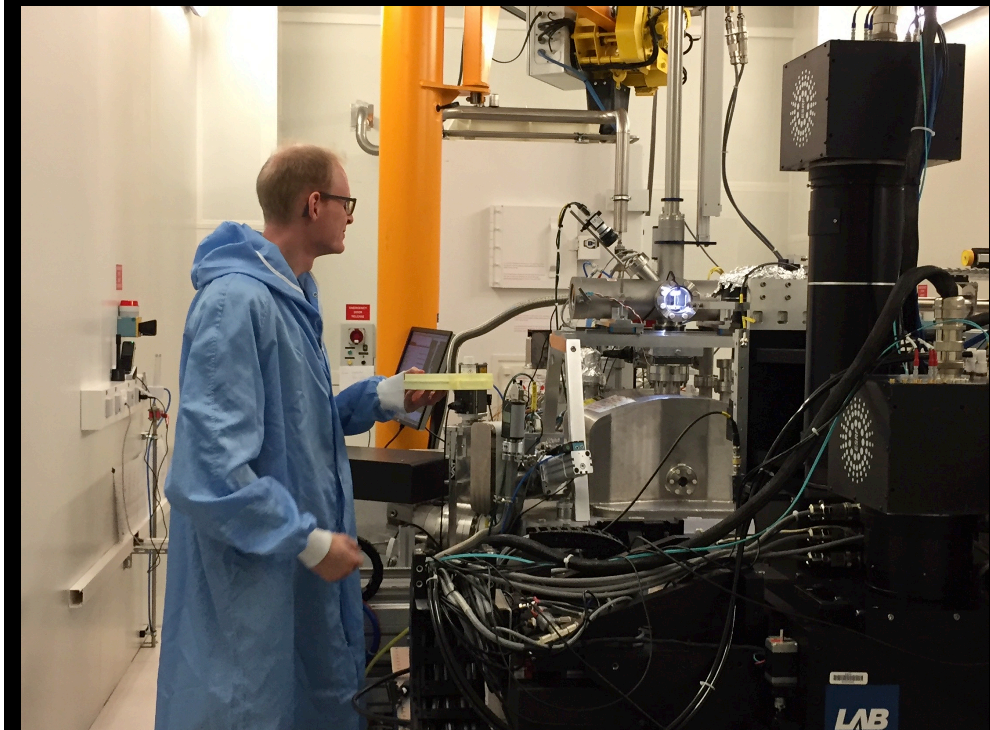


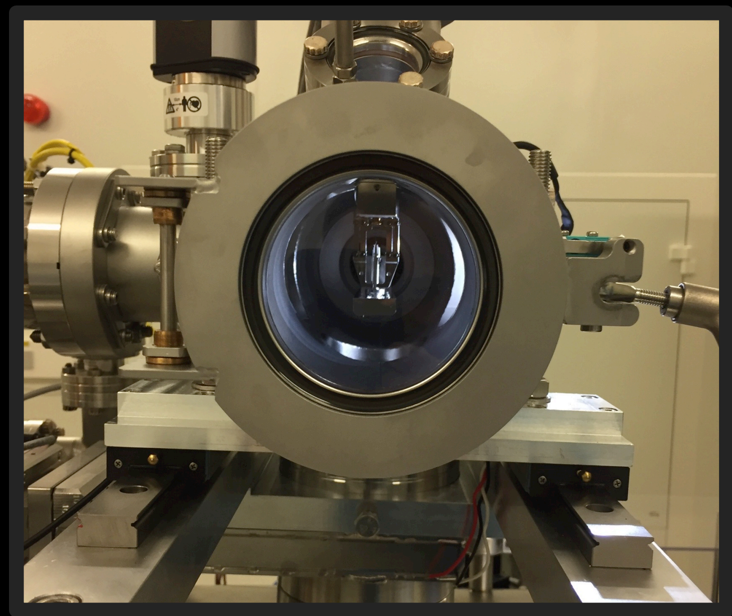
The storage ring

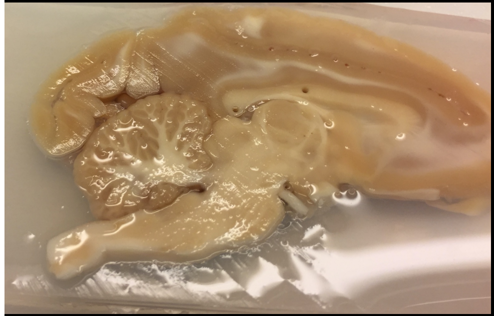
- Segment-wise linear
- Undulators
  - Magnets force electrons to follow a wavy trajectory: Improve the brilliance of the beam
- Bending magnets
  - Electrons deflected from their straight path emit x-ray tangentially (synchrotron light)

The acquisition: The beamline

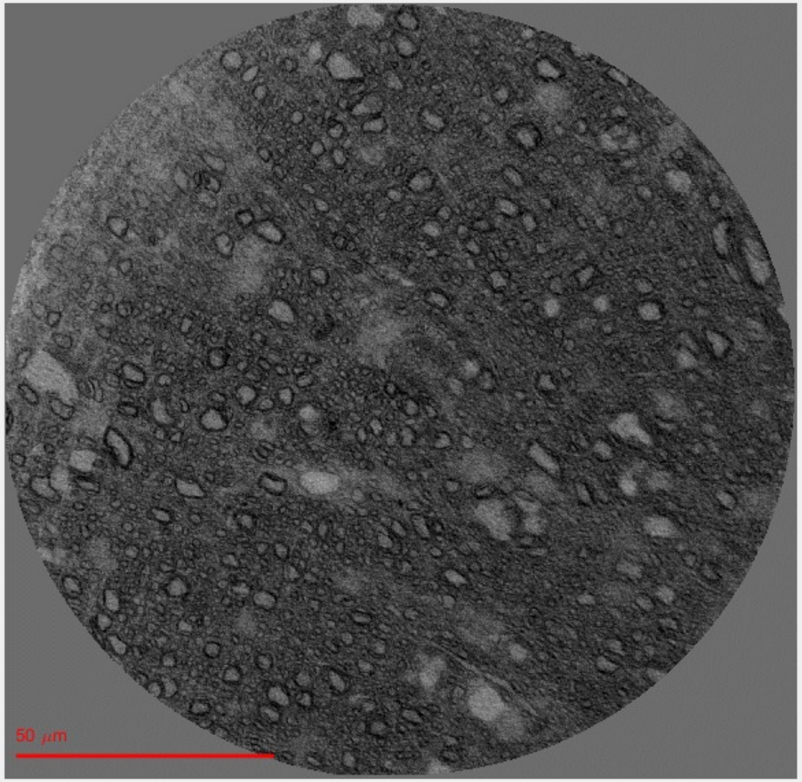




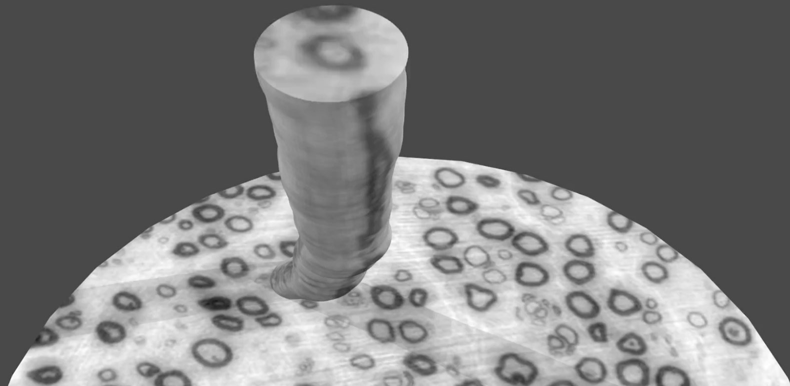


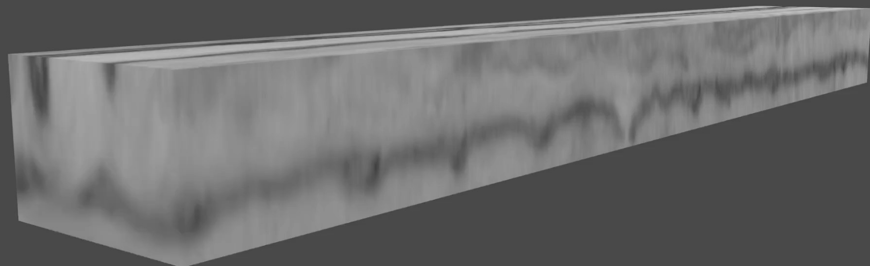


Synchrotron imaging – nano scale image resolution (75 nm)



3D image of nerves fibres in the brain.
(Andersson et al. 2020)
doi: https://doi.org/10.1101/2020.05.29.118737





Layered surface segmentation on human hand nerves (Kjer et al, in prep)

Magnetic Resonance Imaging (MRI)

3D structure of whole body: 10 min

DOI:https://doi.org/10.1016/j.ejrad.2009.09.014

- Magnetic and Radio Frequency in mega Hz
- Soft tissue + brain function
- Expensive compared to CT
- 3D imaging
- No documented danger
- Volume pixel: Voxel
- Clinical voxel sizes 3 to 1 mm^3, but can detect microstructures using biophysical models

Magnetic field

Preclinical MRI 7T

Human 1.5T, 3T, 7T MRI

0.00005 T

MRI and Safety

You most show respect for the invisible danger!!

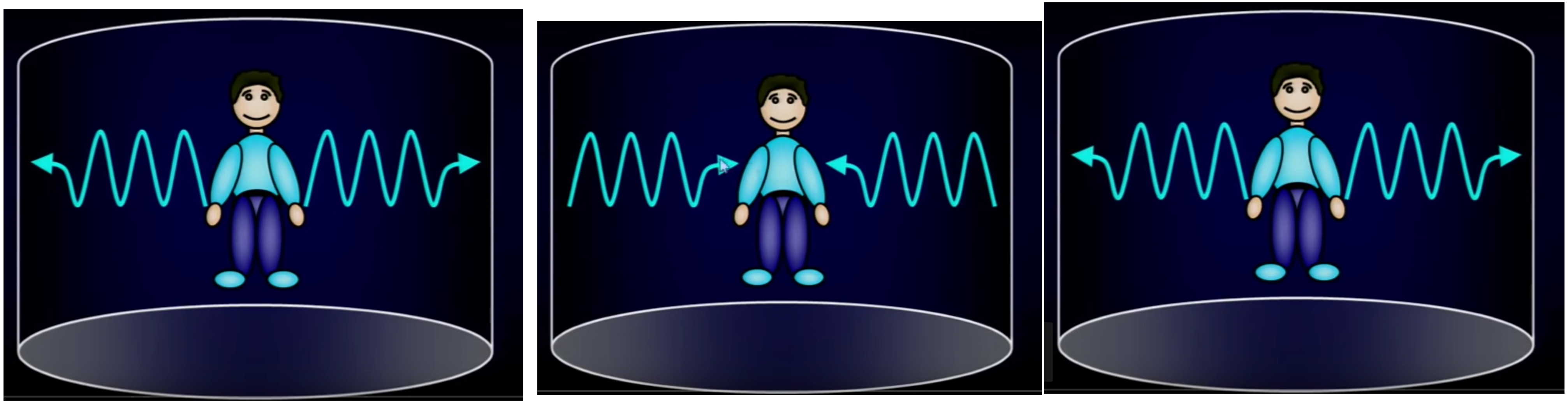

Two hospital workers spent four hours pinned between a highly magnetic MRI machine and a metal oxygen tank.

The 4ft tank was pulled across the room by the machine's magnetic field at Tata Memorial Hospital in New Delhi, India, leaving porter Sunil Jadhav and technician Swami Ramaiah seriously injured.

Hospital authorities launched an investigation into the incident, which was reportedly exacerbated when staff found they were unable to demagnetise the machine.

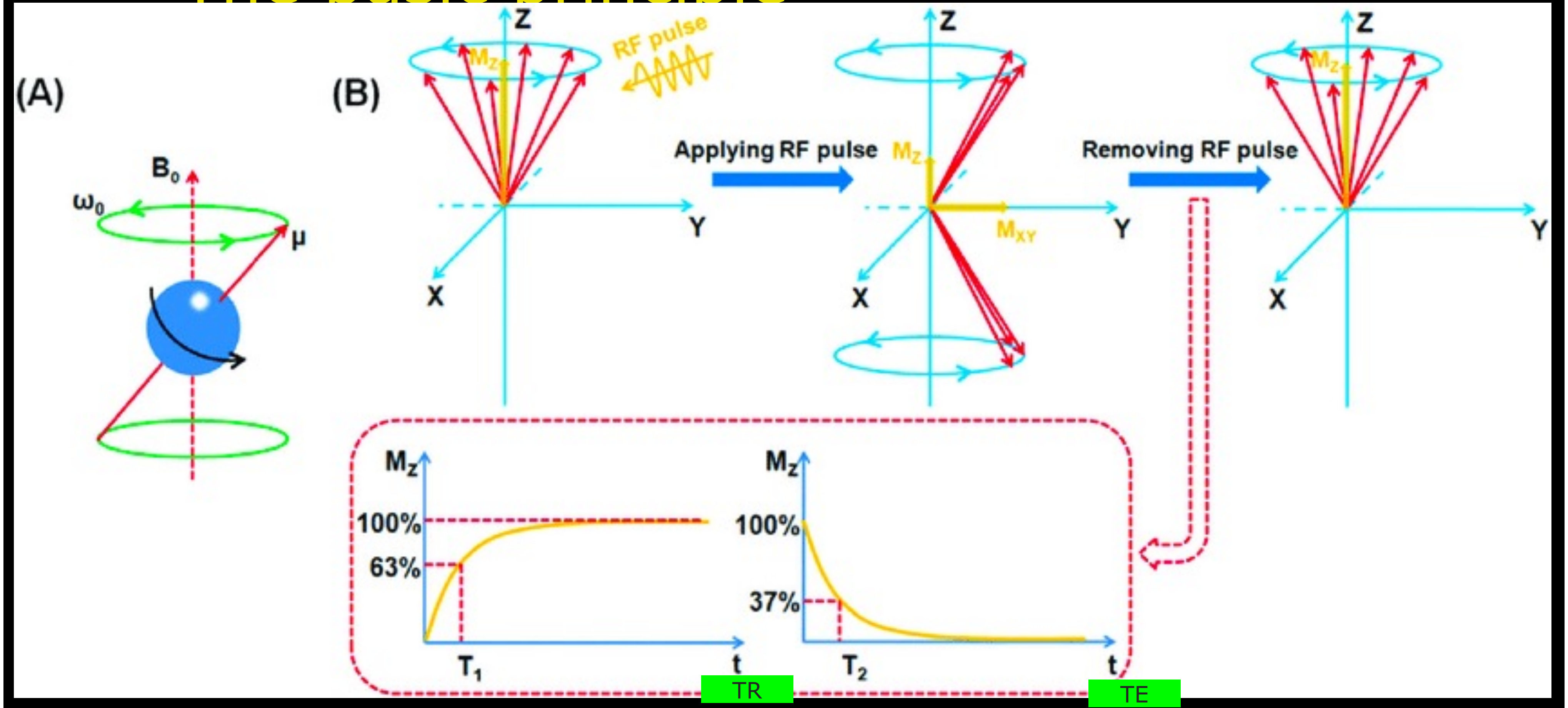
https://www.dailymail.co.uk/news/article-2890088/Two-hospital-workers-spend-FOUR-HOURS-pinned-MRI-machine-metal-oxygen-tank-catapulted-room-device-s-giant-magnet-turned-on.html

The MRI overview



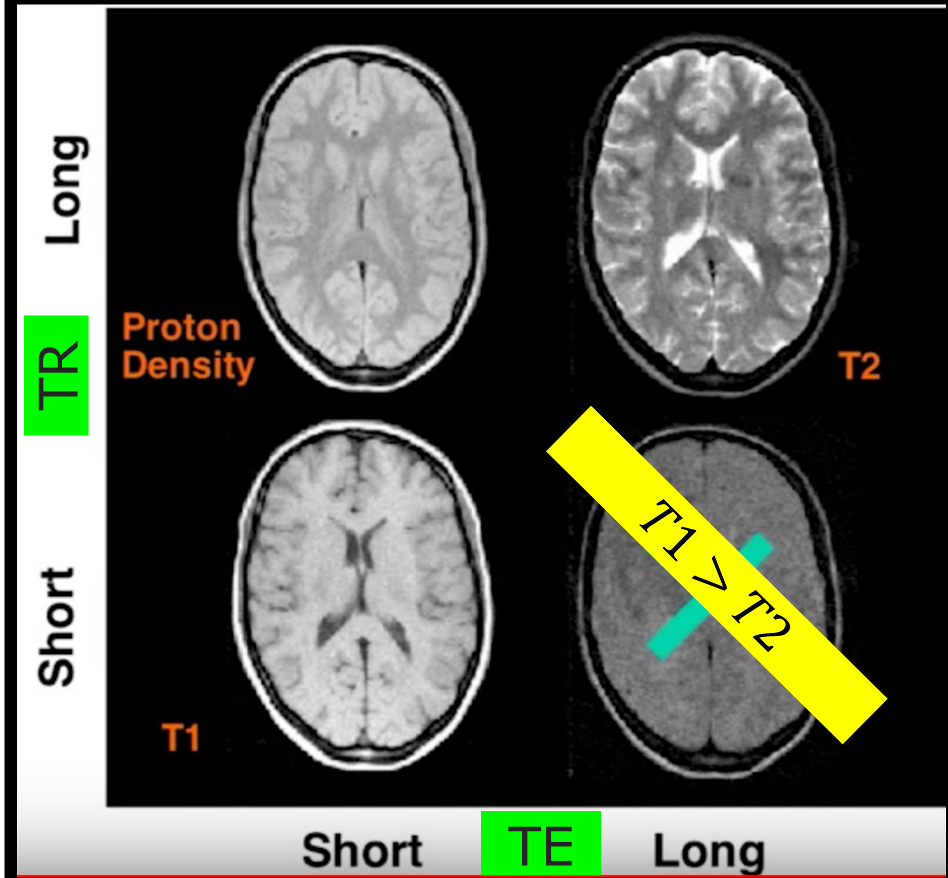
https://www.youtube.com/watch?v=tcGG5njW890

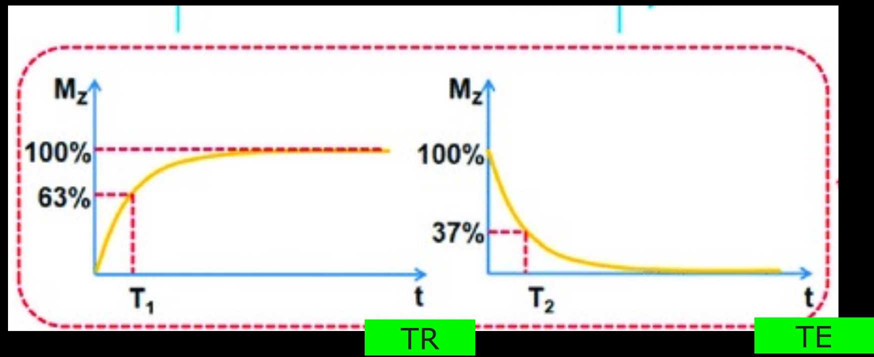
The basic principle



https://www.researchgate.net/figure/Schematic-illustration-of-the-mechanisms-of-MRI-A-Protons-precess-under-an-external_fig1_315899046

The basic principle

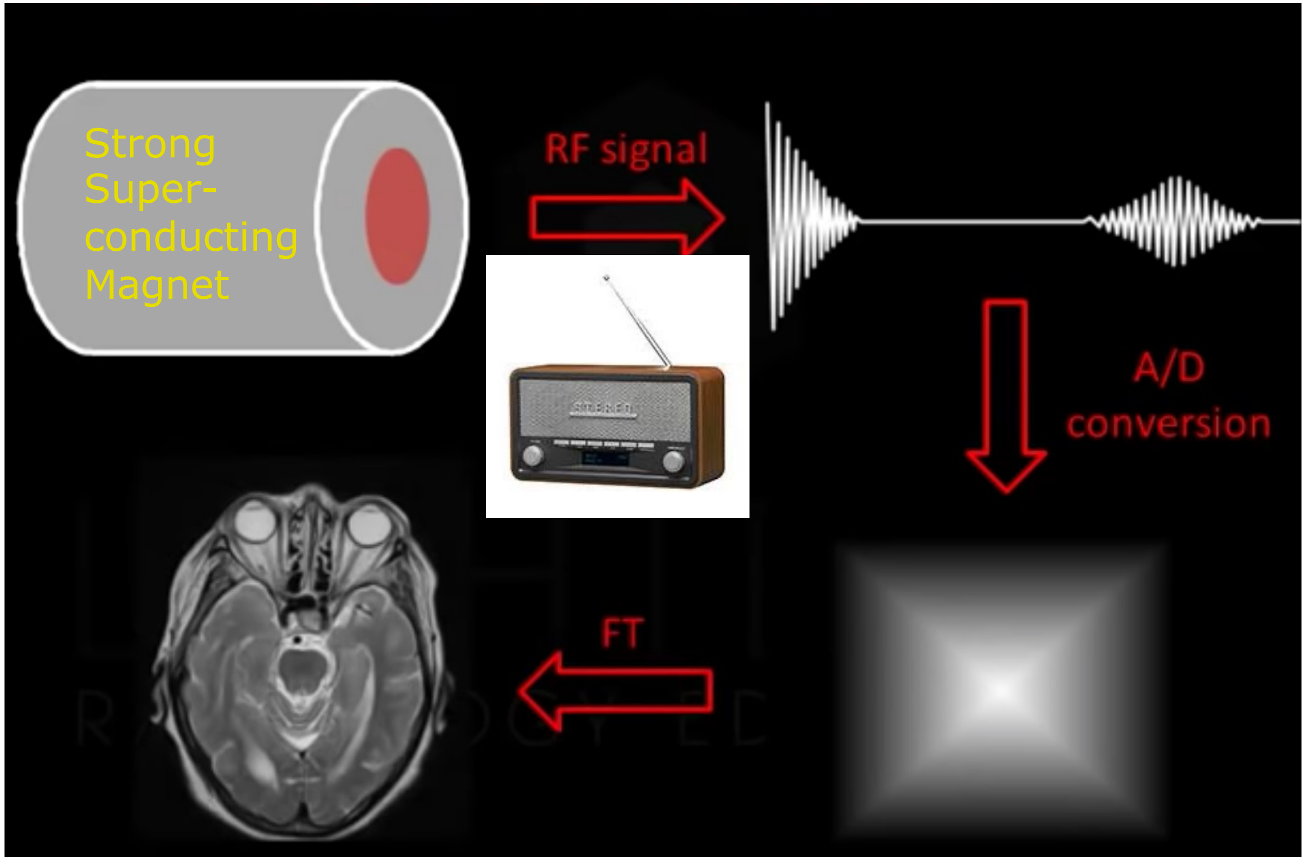




Signal contrast

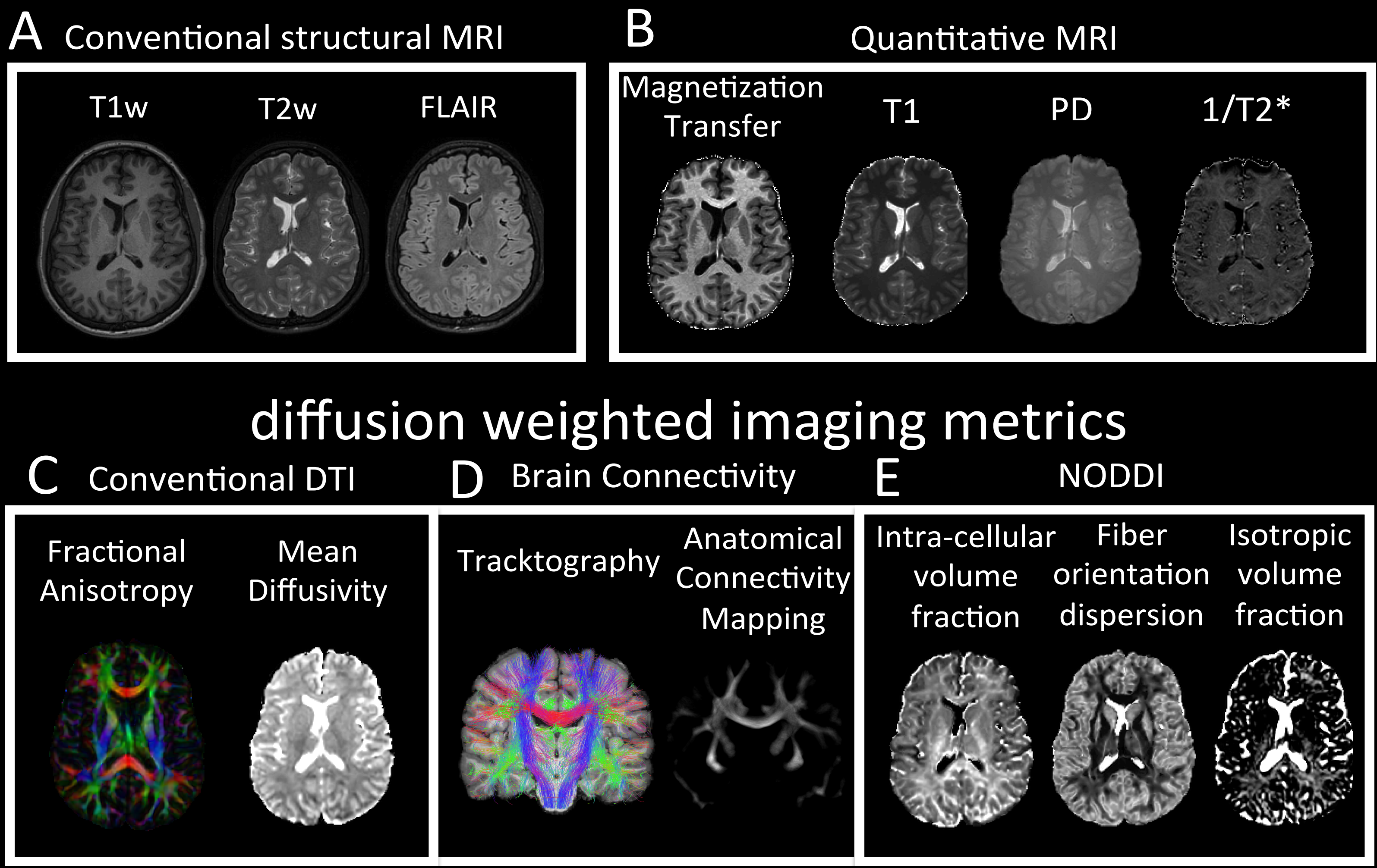
$$Mo = (1 - e^{-\frac{TR}{T1}})e^{-\frac{TE}{T2}}$$

The imaging principle of MRI



Multi-modality of the same subject

- Long scan times: High risk of motion

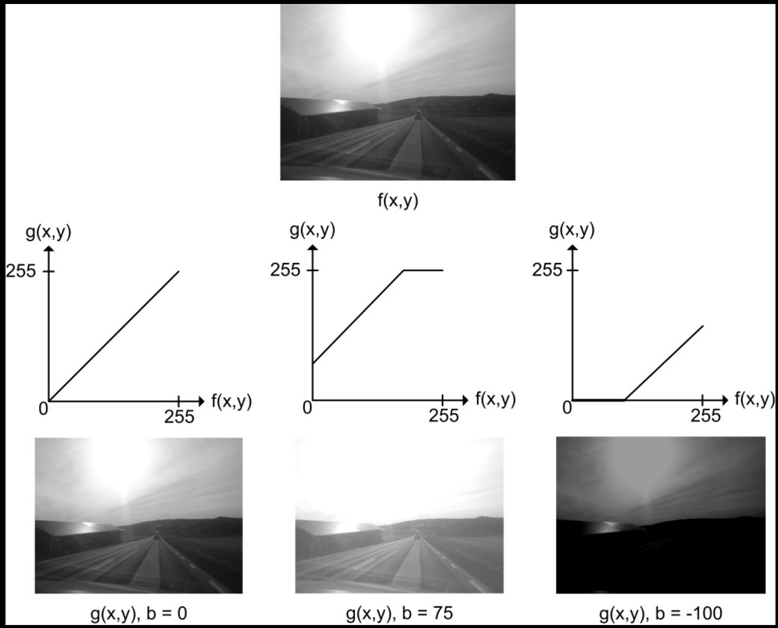


Today – What did you learn?

- Explain where visible light is in the electromagnetic spectrum
- Describe the pin hole camera
- Describe the properties of a thin-lens including focal-length, the optical center, and the focal point
- Estimate the focal length of a thin lens
- Compute the optimal placement of a CCD chip using the thin lens equation
- Describe depth-of-field
- Compute the field-of-view of a camera
- Explain the simple CCD model
- Compute the run-length code of a gray scale image
- Compute the chain coding of a binary image
- Compute the run length coding of a binary image
- Compute the compression ratio
- Describe the difference between a lossless and a lossy image format
- Decide if a given image should be stored using a lossless or a lossy image format
- Understand the principle of X-ray and MRI imaging methods

Next week

- Pixel wise operations
- Colour images
- MIA chapter 4
- MIA chapter 8



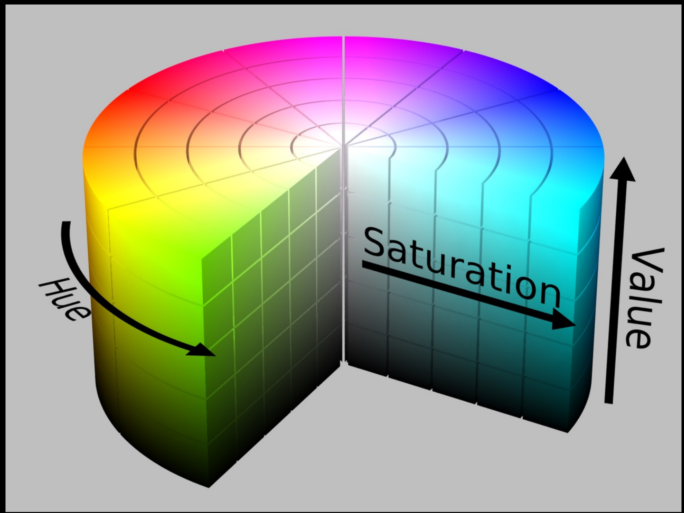


Image Analysis

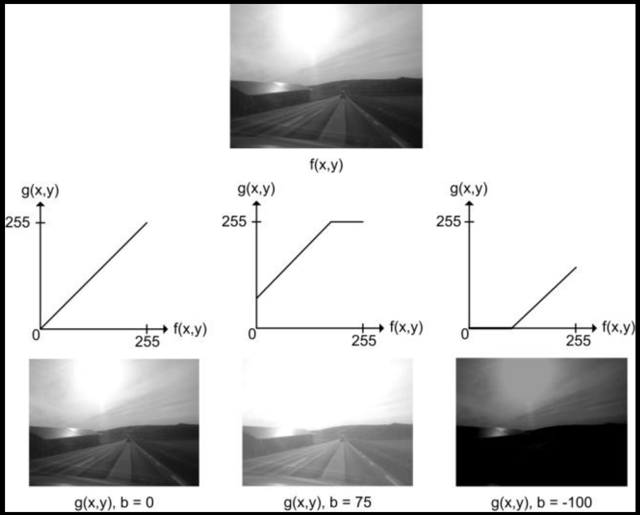
Rasmus R. Paulsen
Tim B. Dyrby

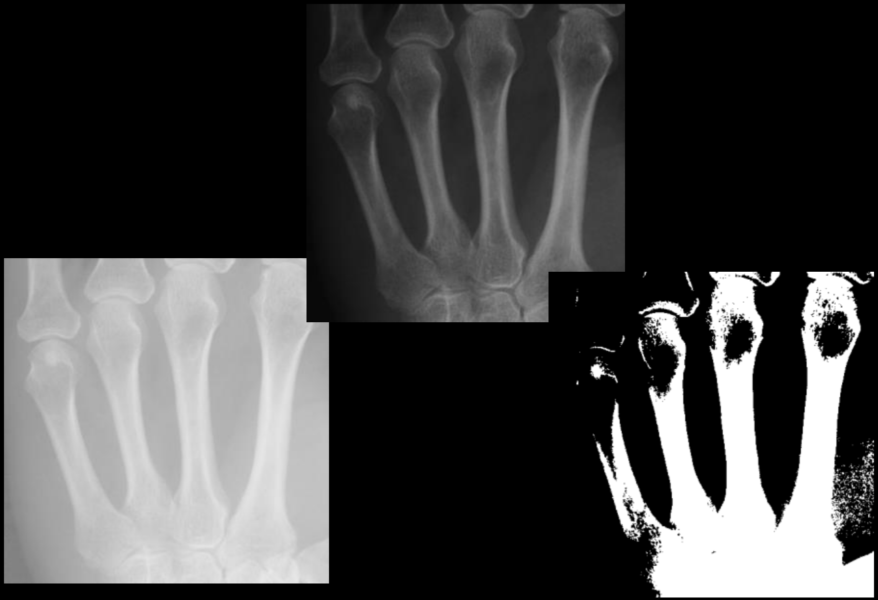
DTU Compute

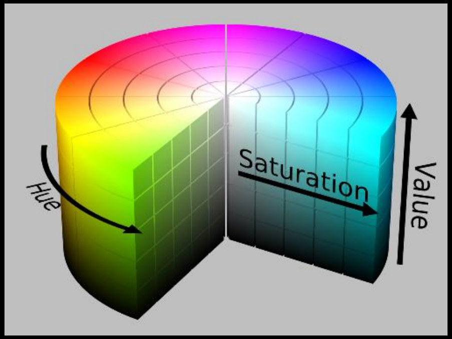
http://courses.compute.dtu.dk/02502

Plenty of slides adapted from Thomas Moeslunds lectures

Week 3
Pixelwise operations and colour images
PCA on images







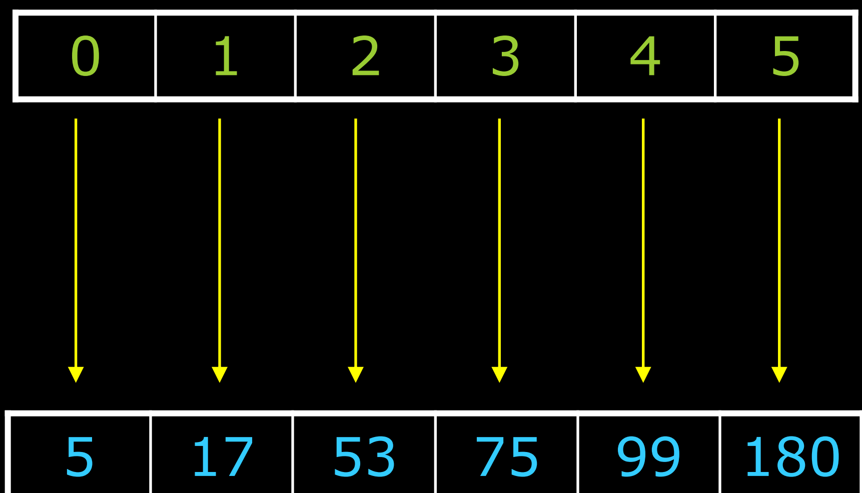
What can you do after today?

- Compute and apply a linear gray transformation
- Describe and compute the image histogram
- Implement and apply histogram stretching
- Implement and apply gamma transformation
- Implement and apply log and exp mappings
- Describe and use thresholding
- Describe and use automatic thresholding
- Perform conversions between bytes and doubles
- Use addition and subtraction of images
- Explain the benefits of bi-modal histograms
- Identify images where global thresholding can be used for object extraction

…and you can even more

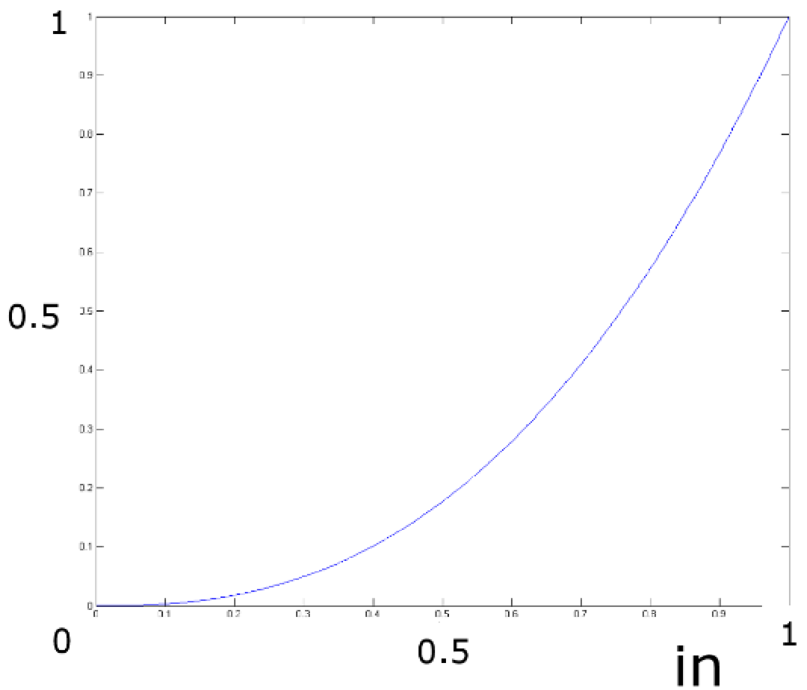
- Describe the basic human visual system including rods and cones
- Describe subtractive colors
- Describe additive colors
- Describe the RGB color space
- Describe the normalised RGB color representation
- Describe the use of the Bayer pattern in digital cameras
- Describe the HSI color space
- Convert from an RGB to a grey level value
- Convert from an RGB value to an HSI value
- Describe the use of different color spaces
- Implement and use color thresholding in RGB space
- Implement and use color thresholding in HSI space

Gray value mappings



- Mapping
  - To make correspondence between two sets of values
- Look-up-table
  - A table of mappings

Mapping Function



f(0.5)?

0.1

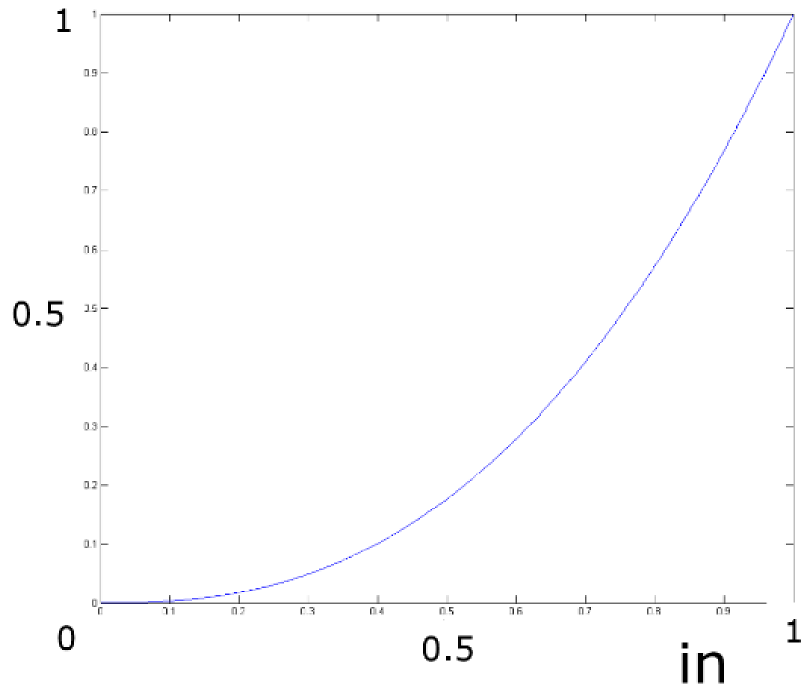
0.2

0.3

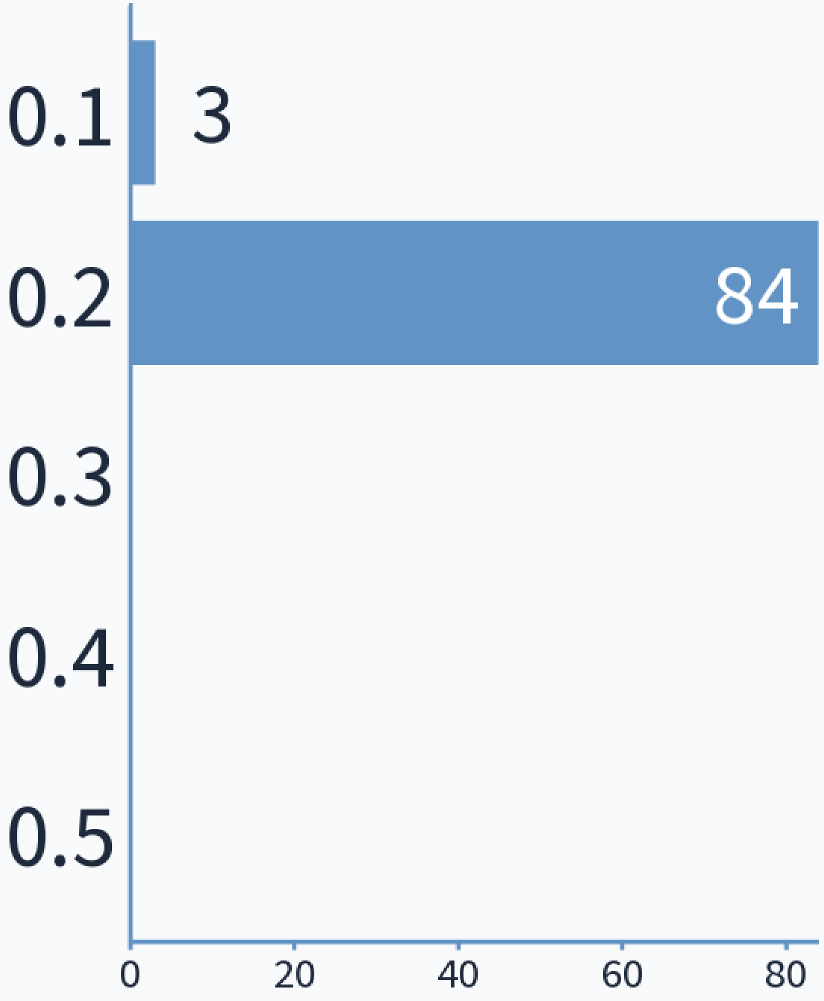
0.4

0.5

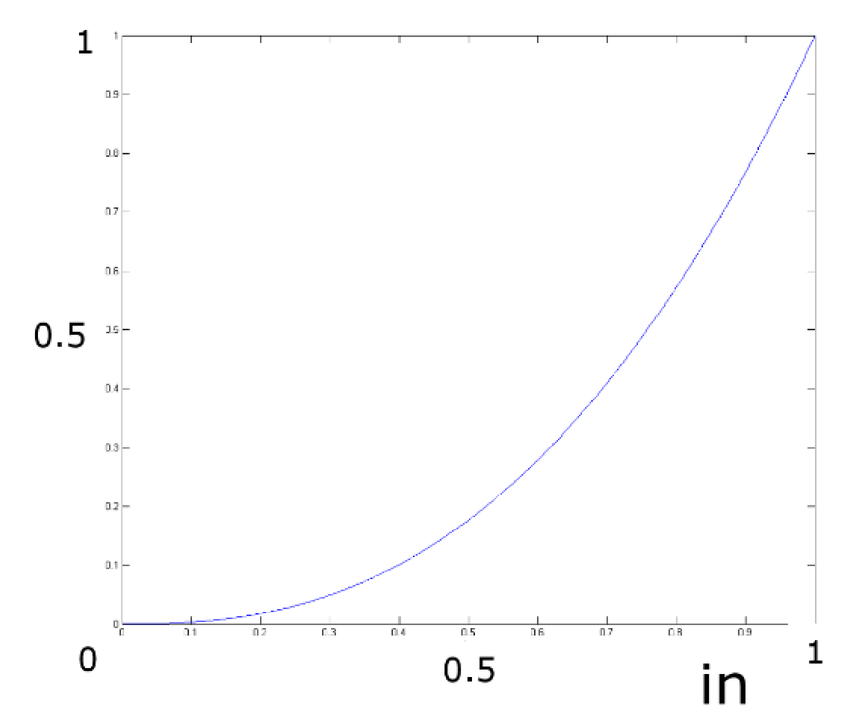
Mapping Function



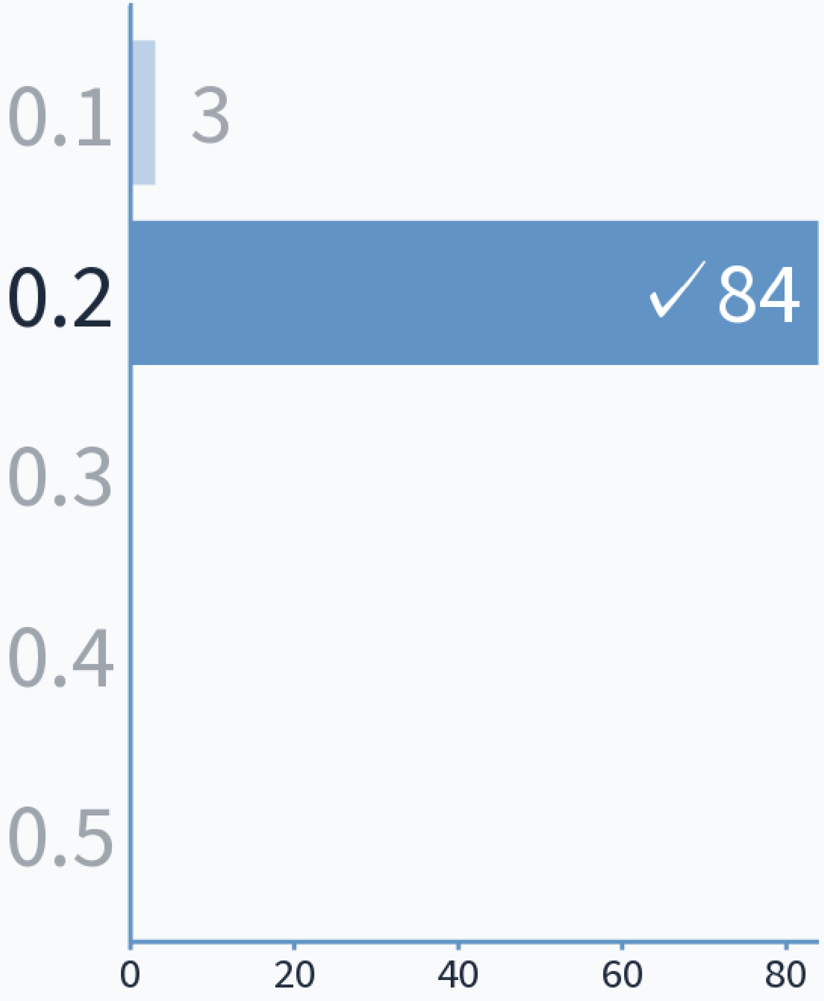
f(0.5)?



Mapping Function

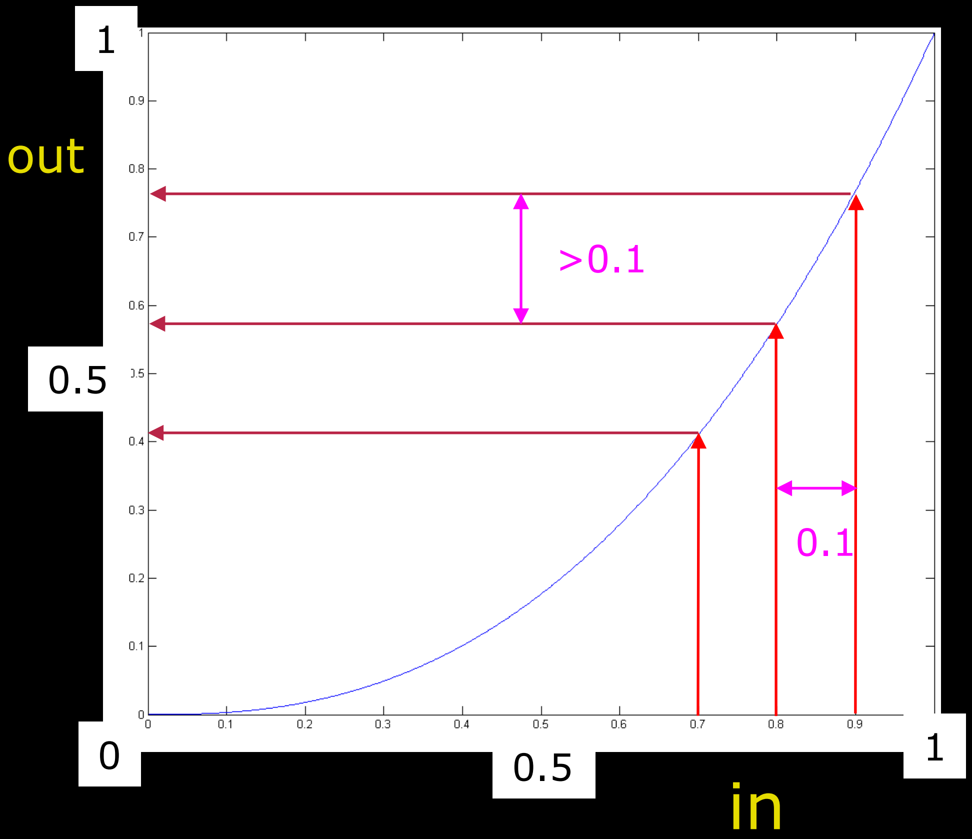


f(0.5)?



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Gray value mappings

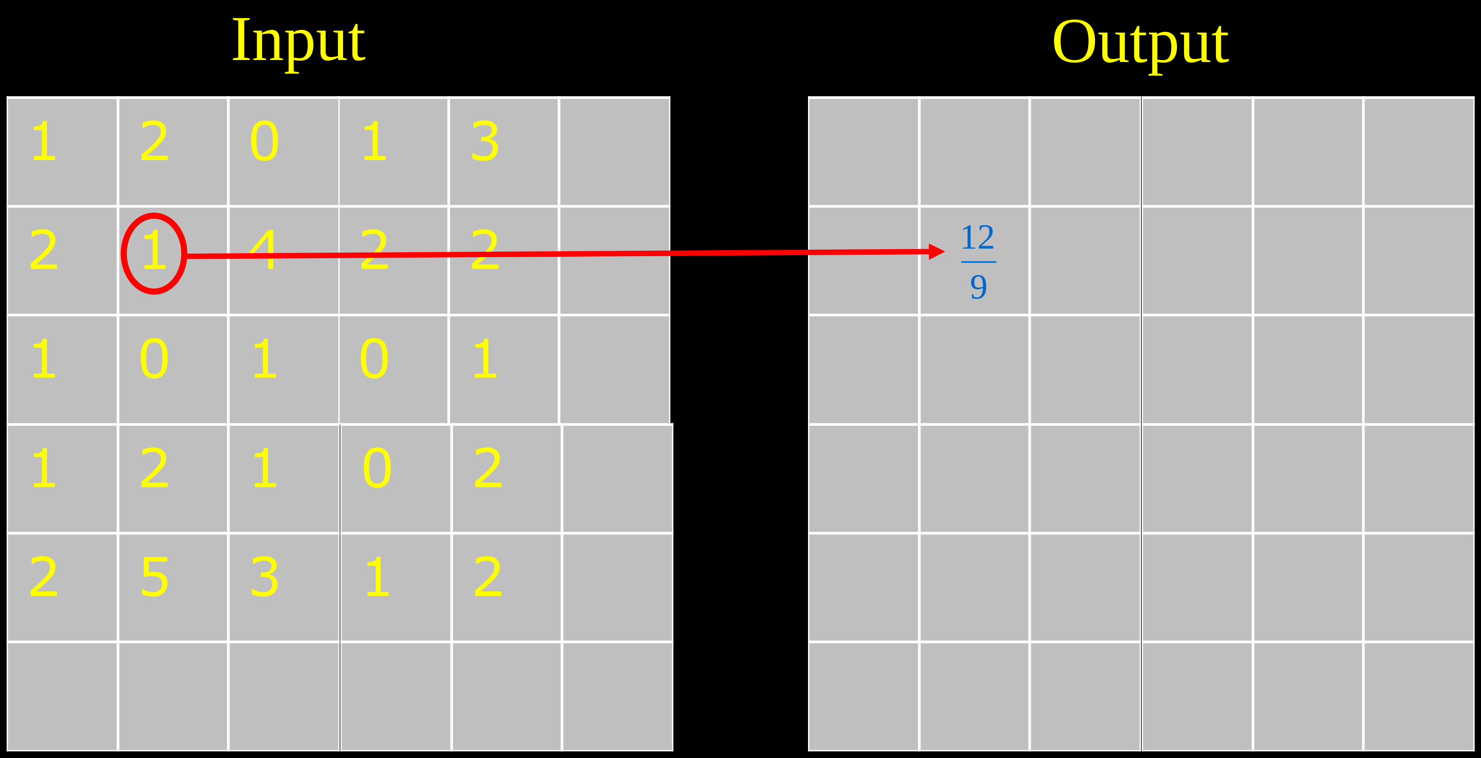


- Mapping
  - To make correspondence between two sets of values
- Mapping function
  - out = f(in)
- What happens with the values?
  - Values with difference 0.1
  - Output values "spread out"

Why change gray level values

- When could it be good to change the gray level values?
  - Lack of contrast
  - Make the image lighter
  - Make the image darker

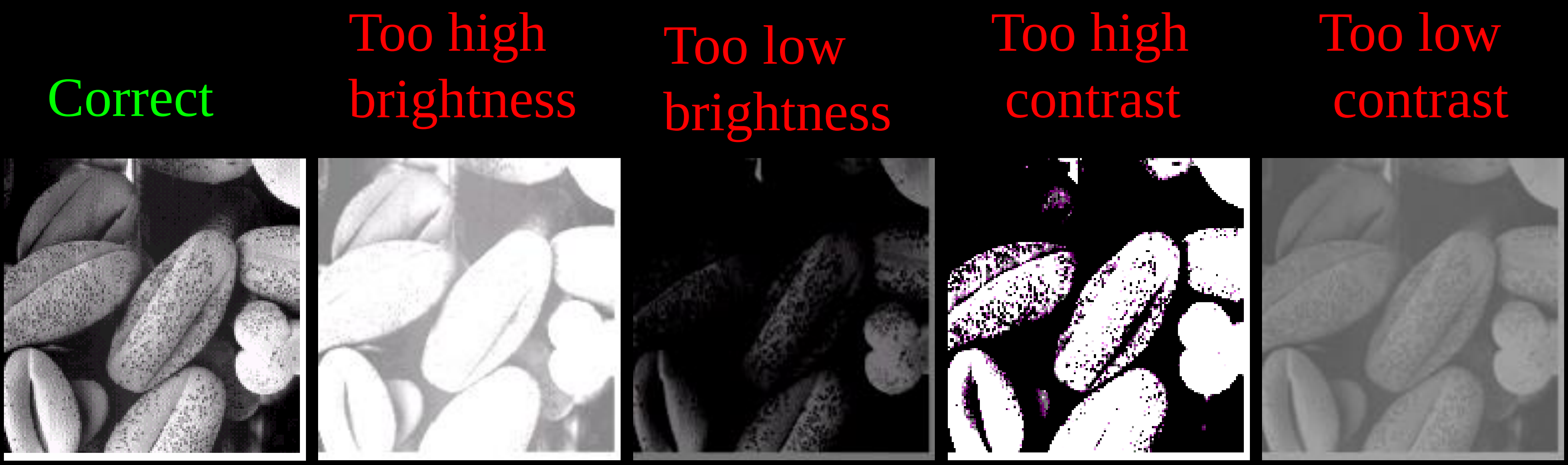
Point processing



- The value of the output pixel is only dependent on the value of one input pixel
- A global operation – changes all pixels

Point processing

- Grey level enhancement
  - Process one pixel at a time independent of all other pixels
  - For example used to correct Brightness and Contrast
    - Known from the television remote control

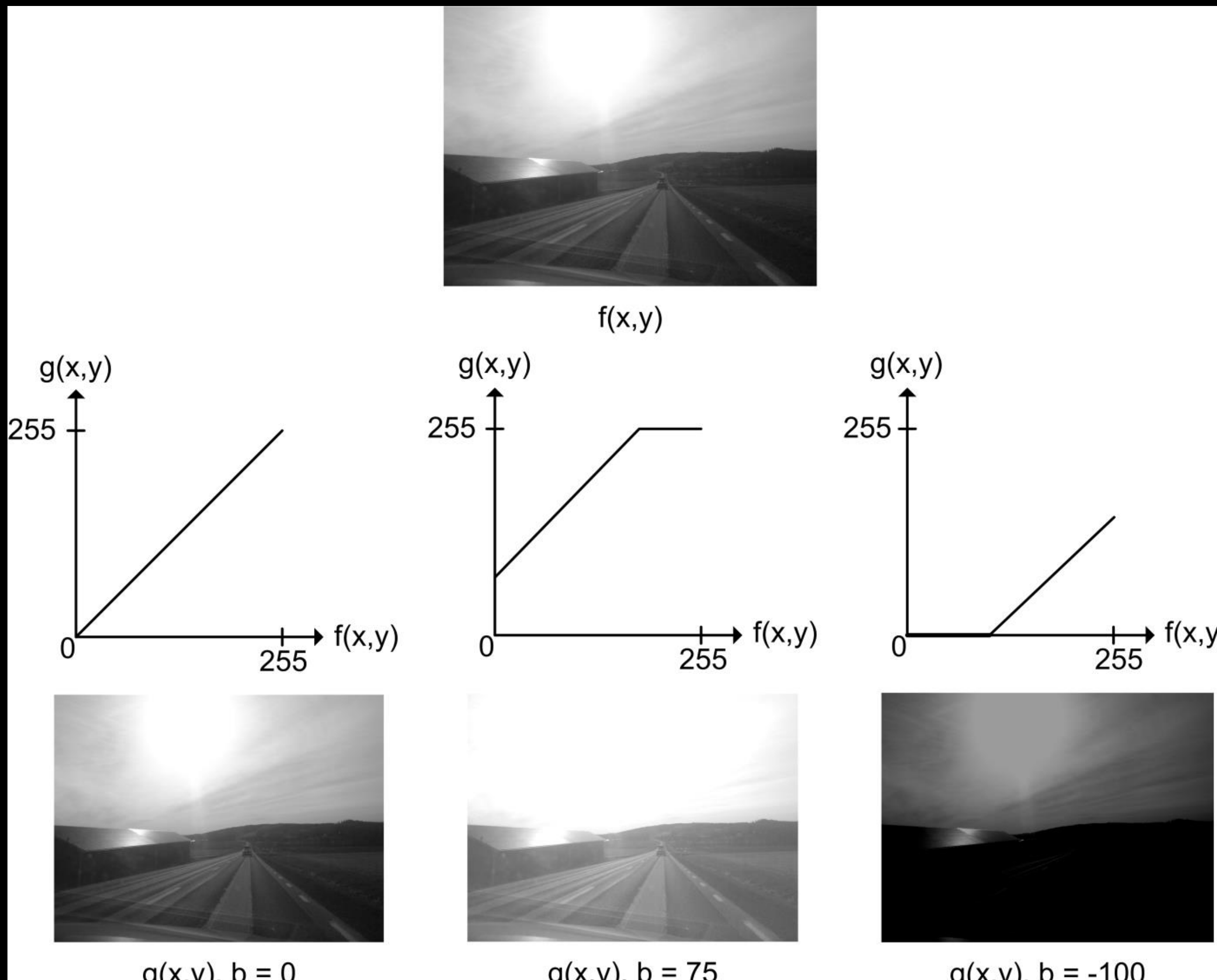


Brightness

- The brightness is the intensity
- Change brightness:
  - To each pixel is added the value *b*
  - f(x, y) is the input image
  - g(x, y) is the (enhanced) output image
- If *b*>0 : brighter image
- If *b*<0 : less bright image

$$g(x, y) = f(x, y) + b$$

Brightness


f(x,y)
g(x,y)
255
0
255
f(x,y)
g(x,y)
255
0
255
f(x,y)
g(x,y)
255
0
255
f(x,y)
g(x,y), b = 0
g(x,y), b = 75
g(x,y), b = -100

Contrast



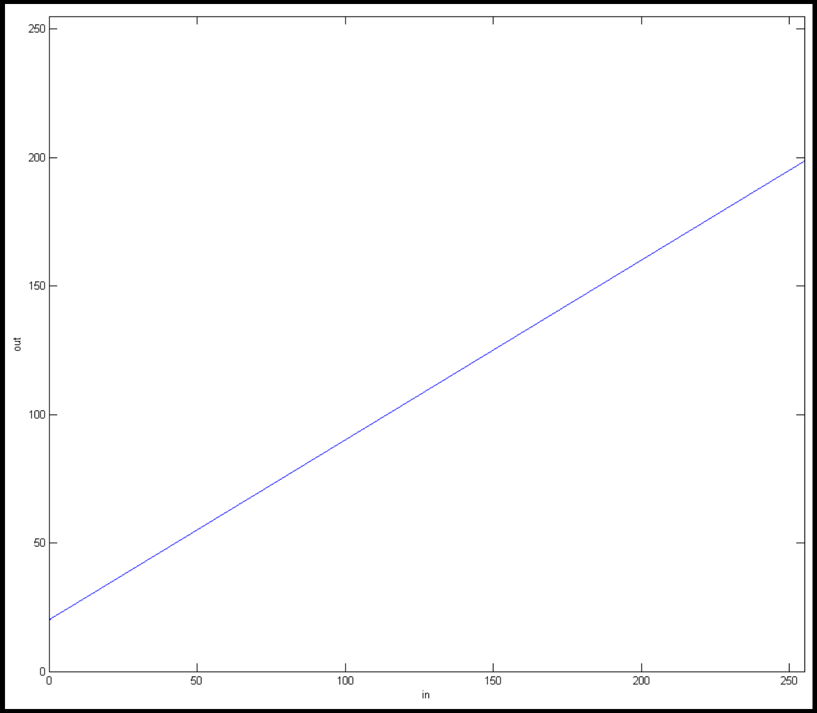
- The contrast describes the level of details we can see
- Change contrast
- Each pixel is multiplied by *a*
  - f(x, y) is the input image
  - g(x, y) is the (enhanced) output image
- If *a*>1 => more contrast
- If *a*<1 => less contrast

$$g(x,y) = a * f(x,y)$$

Combining brightness and contrast

- A straight line
- Called a *linear transformation*
- Here a = 0.7 and b = 20

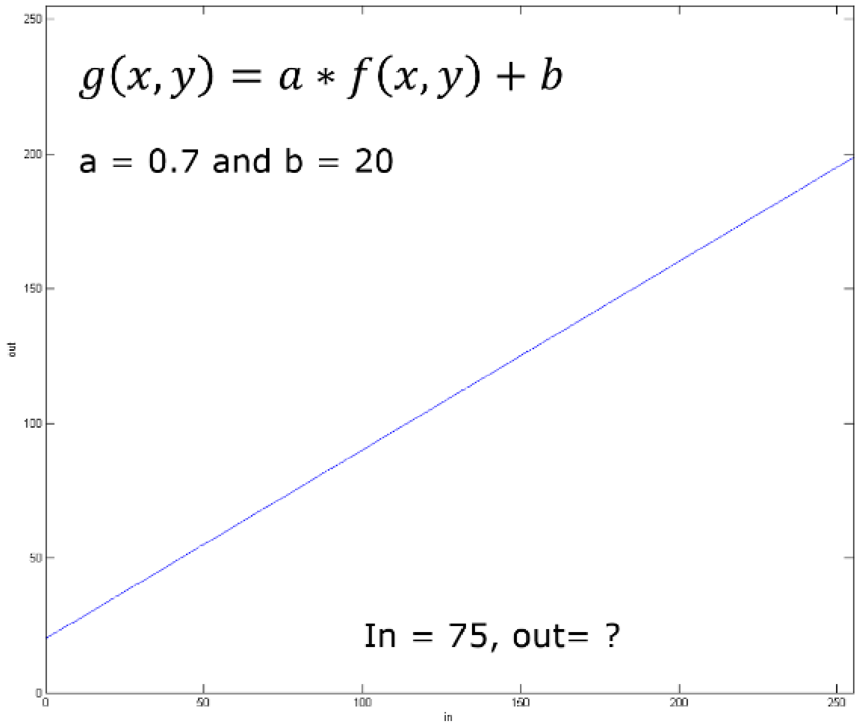
out



in

$$g(x,y) = a * f(x,y) + b$$

Linear Transformation



20

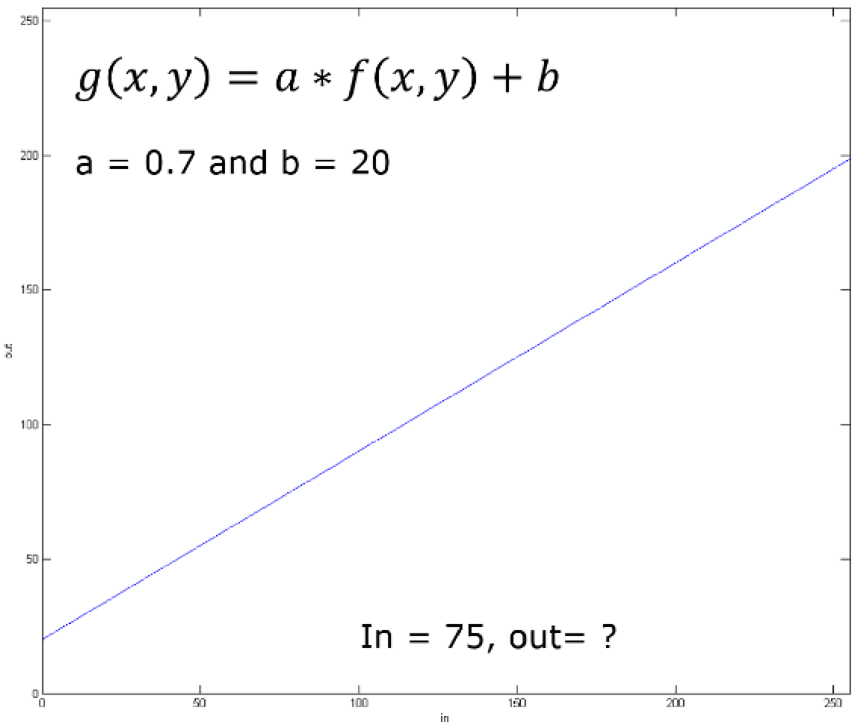
45

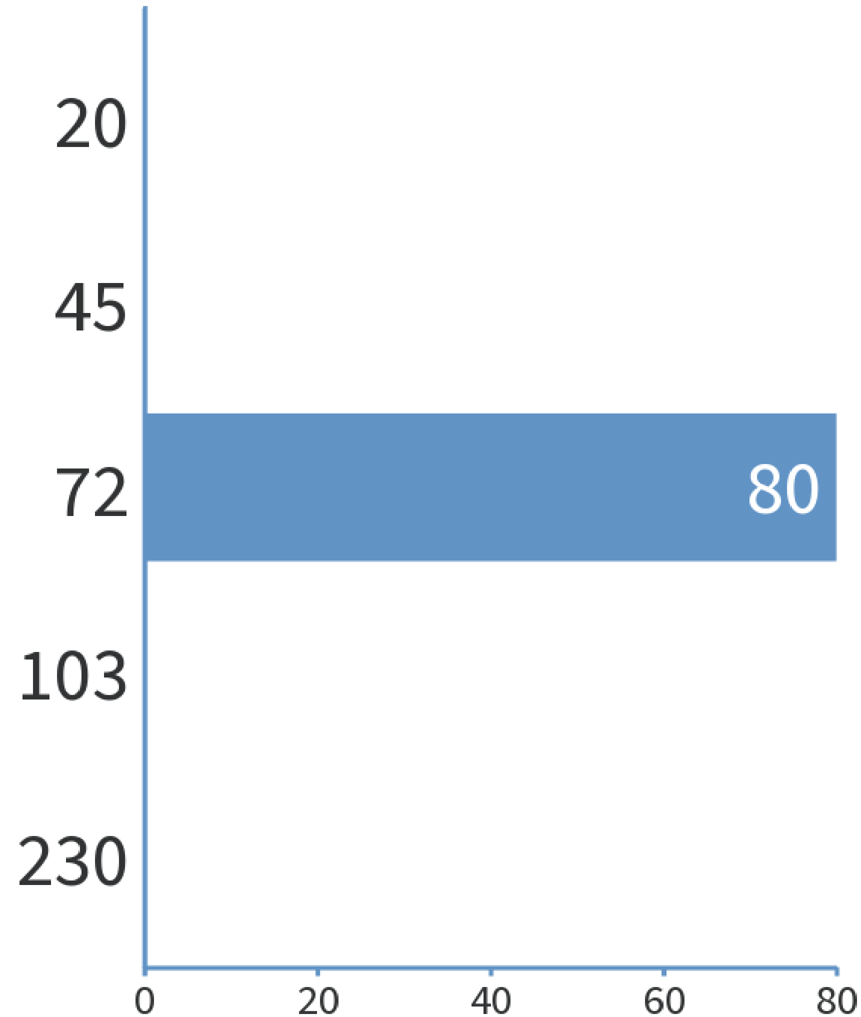
72

103

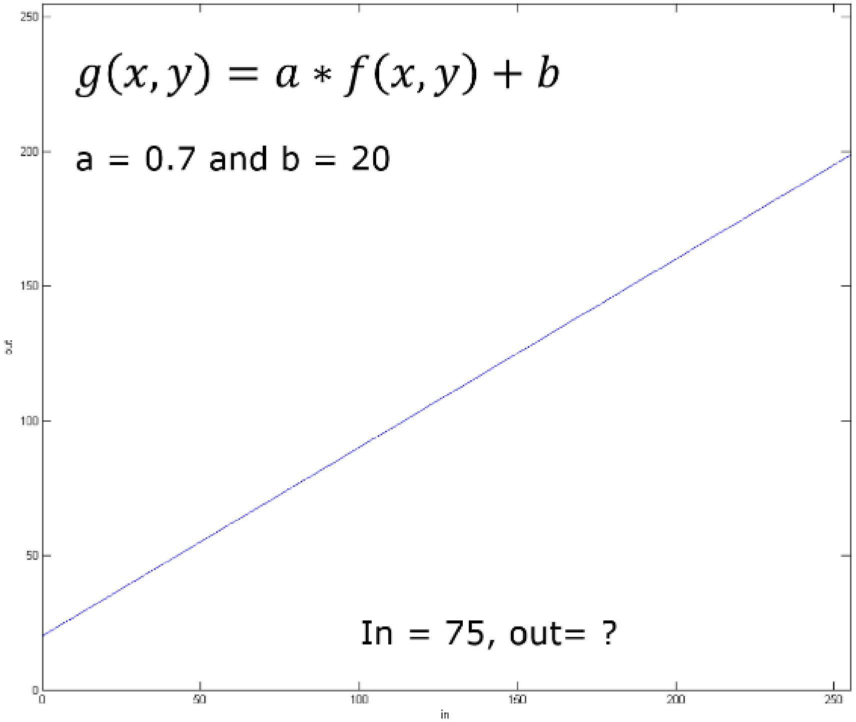
230

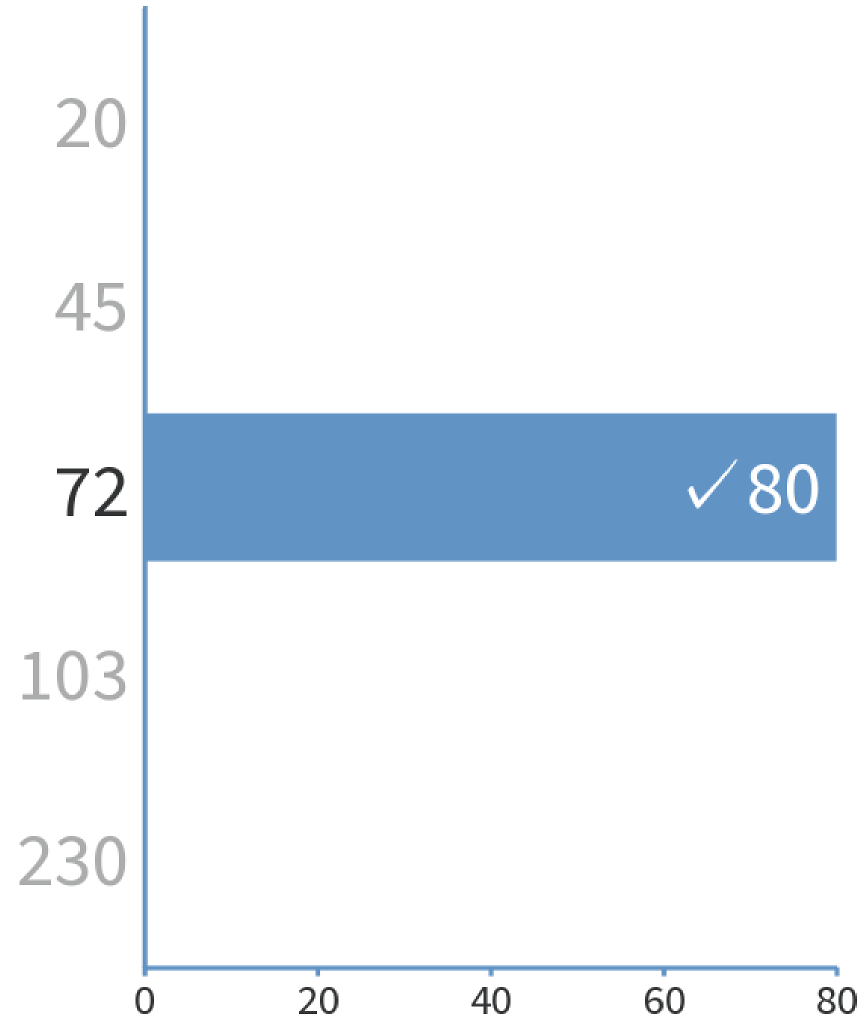
Linear Transformation





Linear Transformation



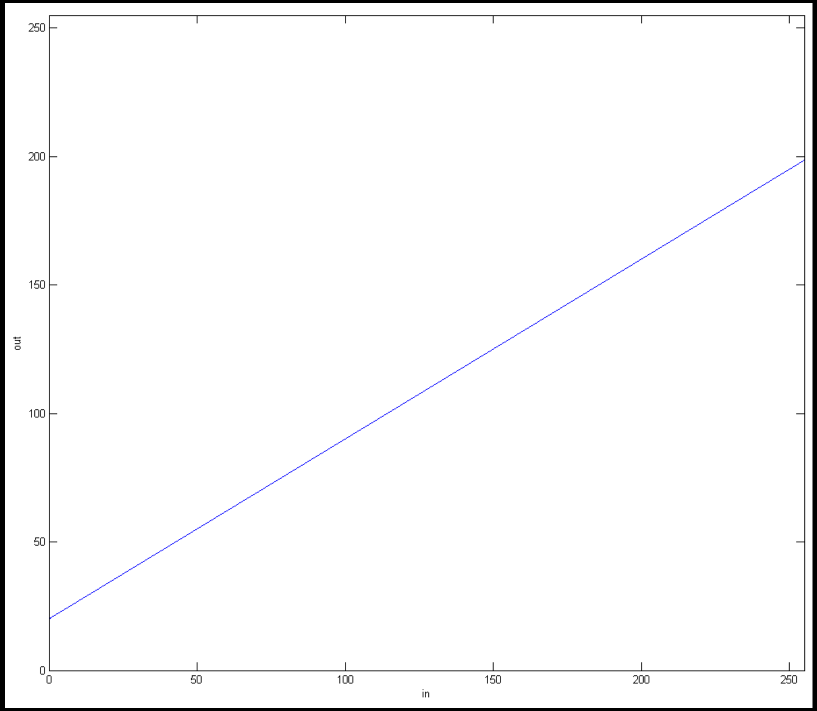


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Combining brightness and contrast

- A straight line
- Called a *linear transformation*
- Here a = 0.7 and b = 20
- What will the visual result be on the output image?
  - More bright (b > 0)
  - Less contrast (a < 1)

out



in

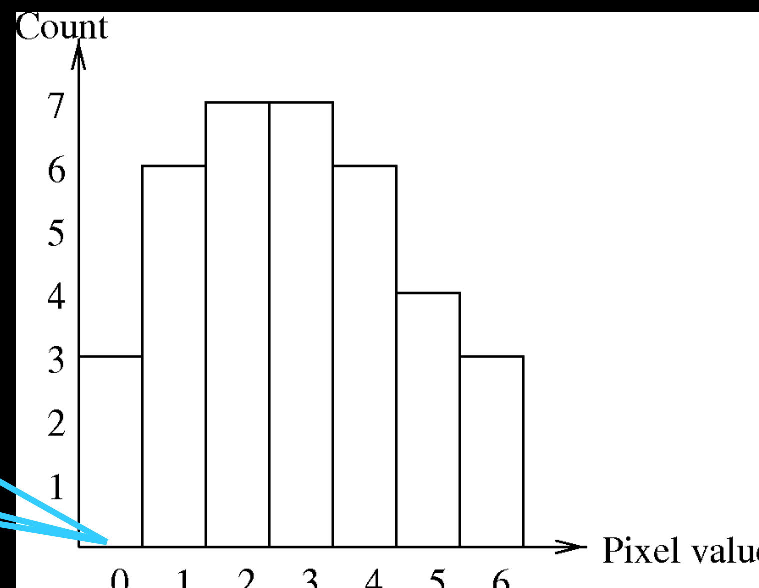
$$g(x, y) = a * f(x, y) + b$$

Histogram Reminder

- A histogram normally contains the same number of "bins" as the possible pixel values
- A bin stores the number of pixel with that value

0	2	6	6	3	3
1	4	3	4	4	4
3	2	5	1	5	2
1	4	2	1	3	1
2	5	3	0	2	0
4	2	5	6	3	1

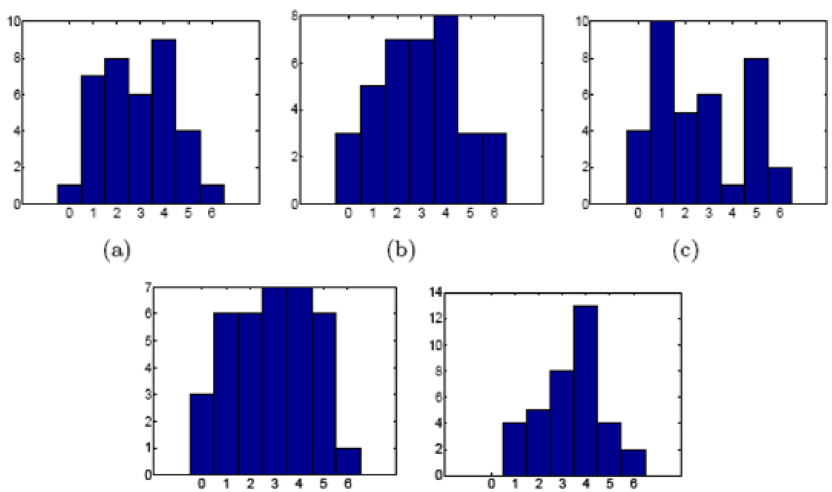
3



Choose the histogram that represents the image

0	5	3	5	2	1
3	5	5	3	3	1
1	1	1	3	2	3
6	2	2	1	0	0
0	2	1	5	1	5
5	5	1	4	1	6

Figur 6: Grayscale billede.



(a) (b) (c) (d) (e)

A

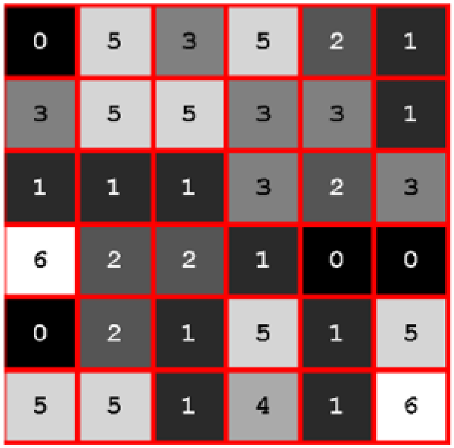
B

C

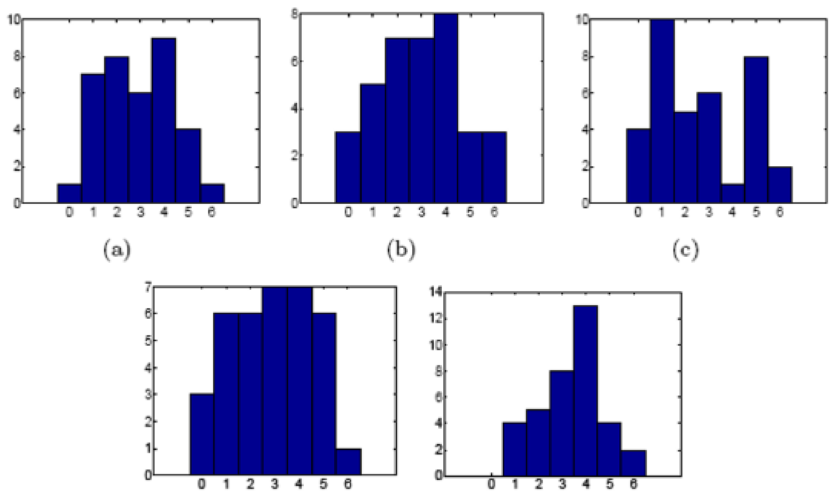
D

None of the above

Choose the histogram that represents the image

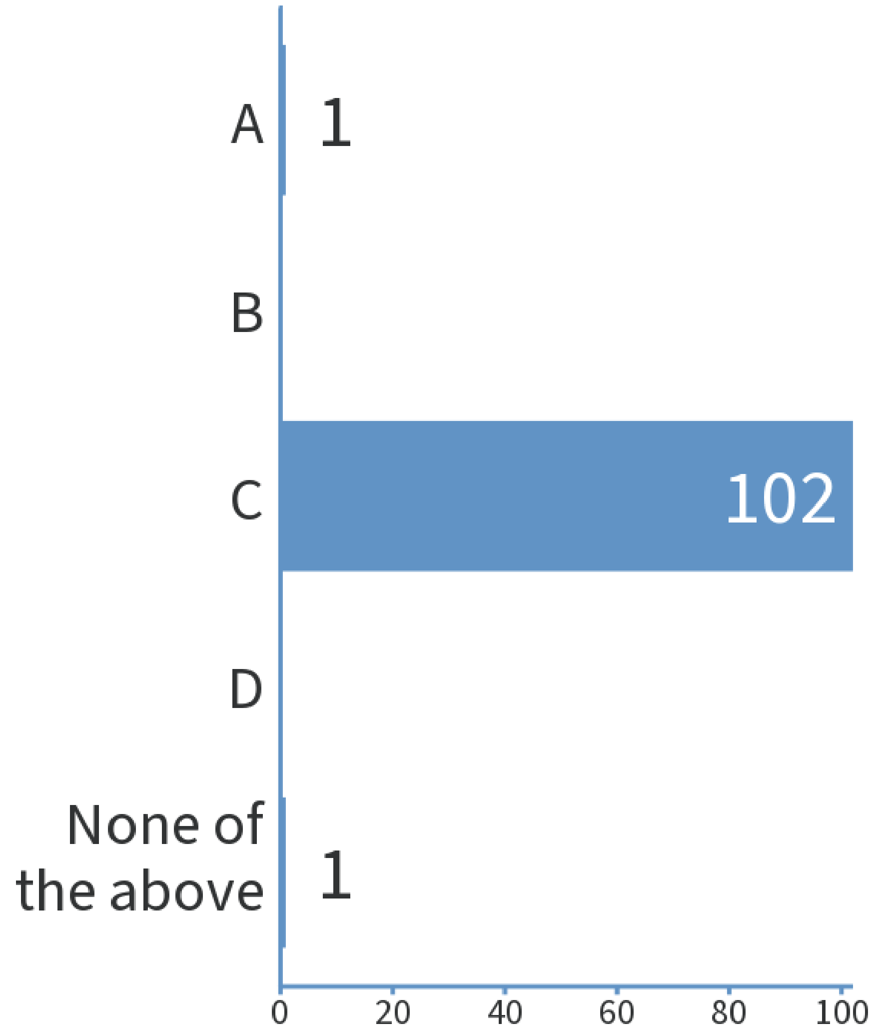


Figur 6: Grayscale billede.

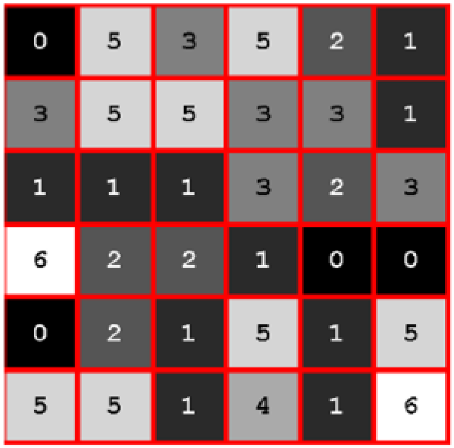


(a) (b) (c)

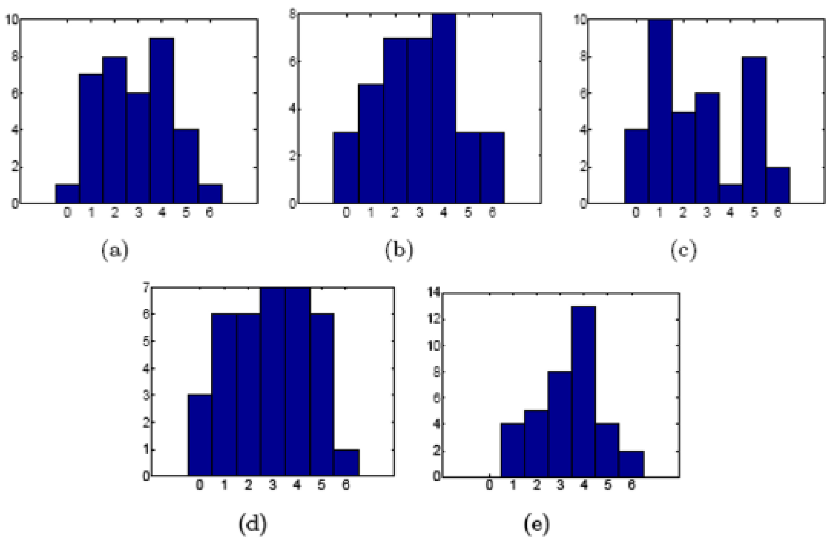
(d) (e)

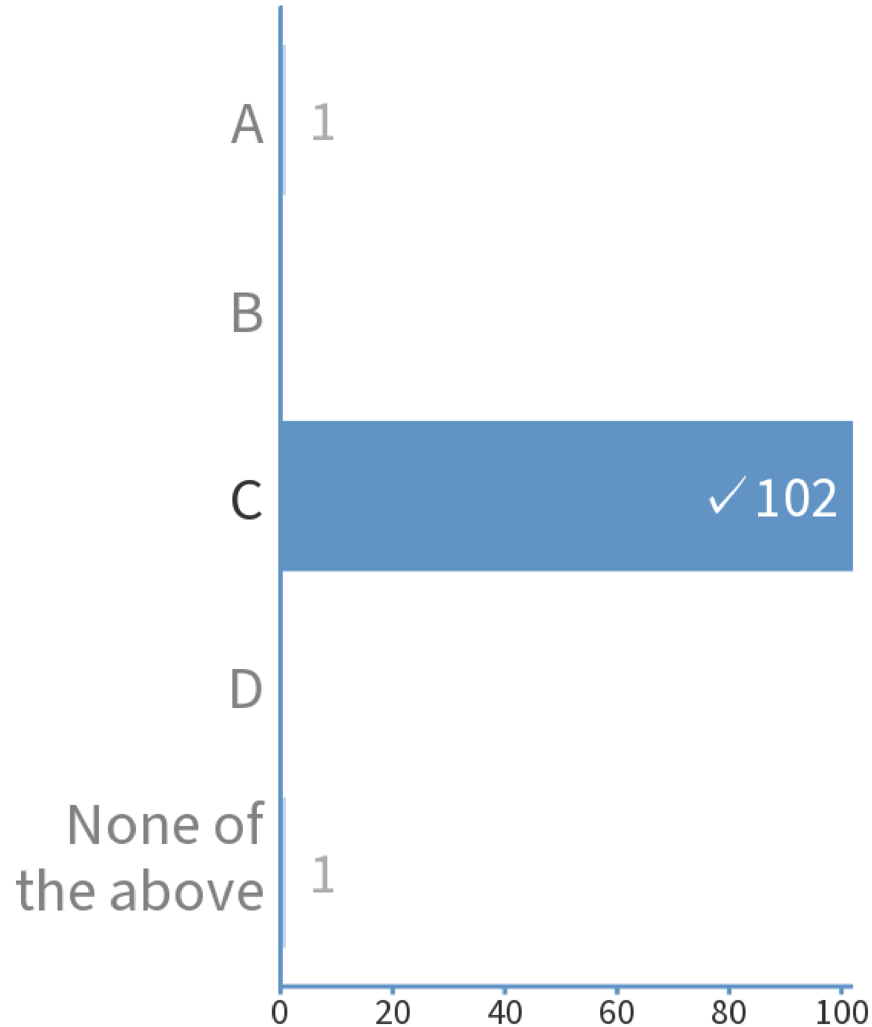


Choose the histogram that represents the image



Figur 6: Grayscale billede.

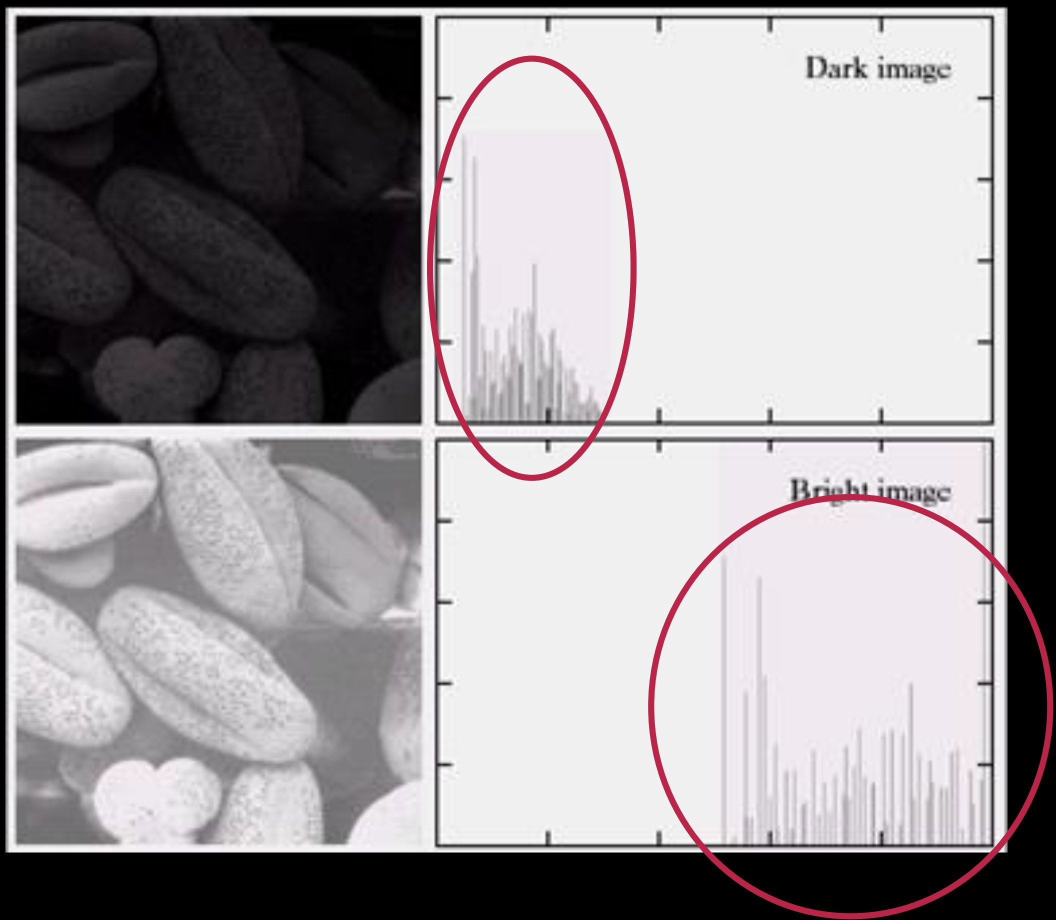




Back to the histogram

- The shape of the histogram tells us a lot!

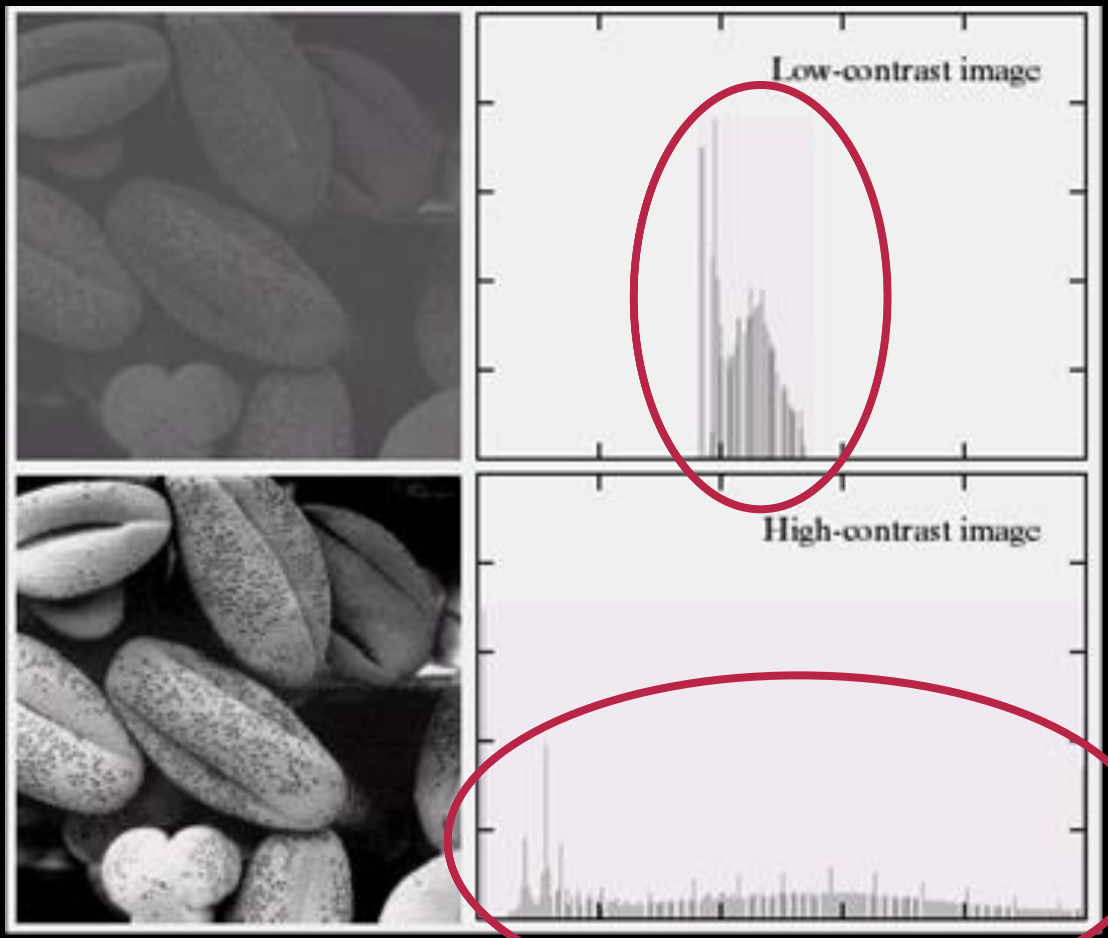
Histogram inspection



Dark image

Bright image

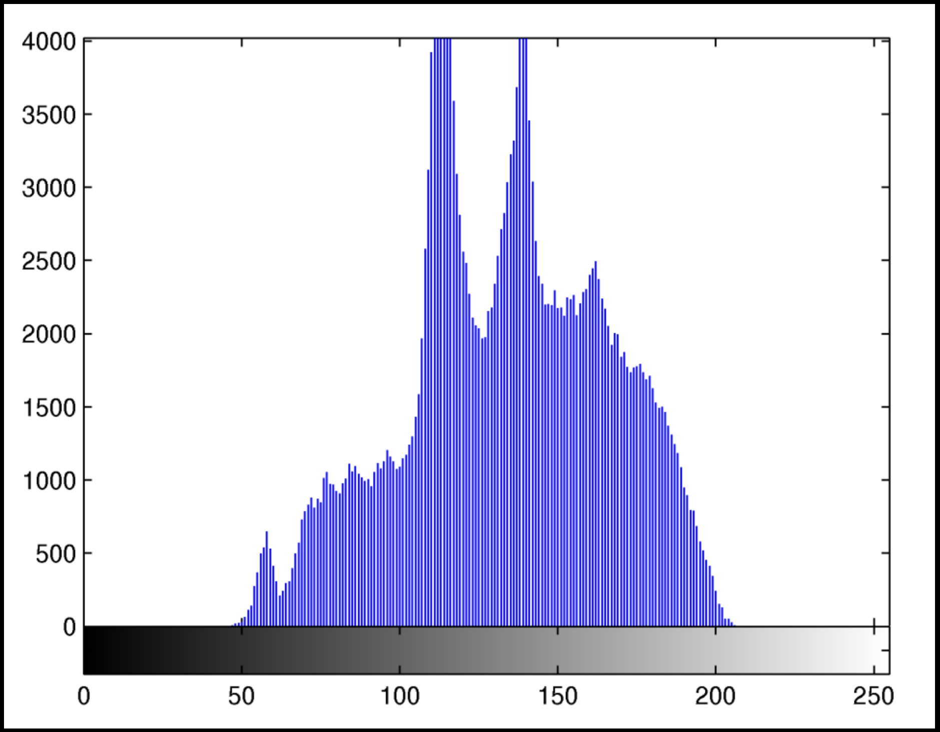
Histogram inspection



Low contrast

High contrast

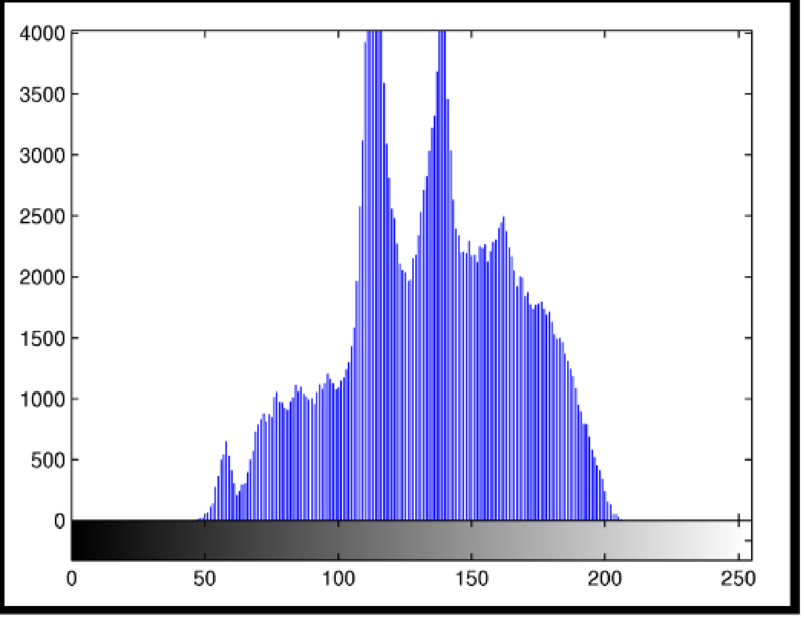
Histogram stretching



- How do we optimise the image using the histogram?
  - Minimum and maximum values?
  - Stretch it so new minimum = 0 and new maximum = 255

Histogram stretching

- We want
  - Min = 0
  - Max = 255
- We have
  - Min = 32
  - Max = 208



Using brightness

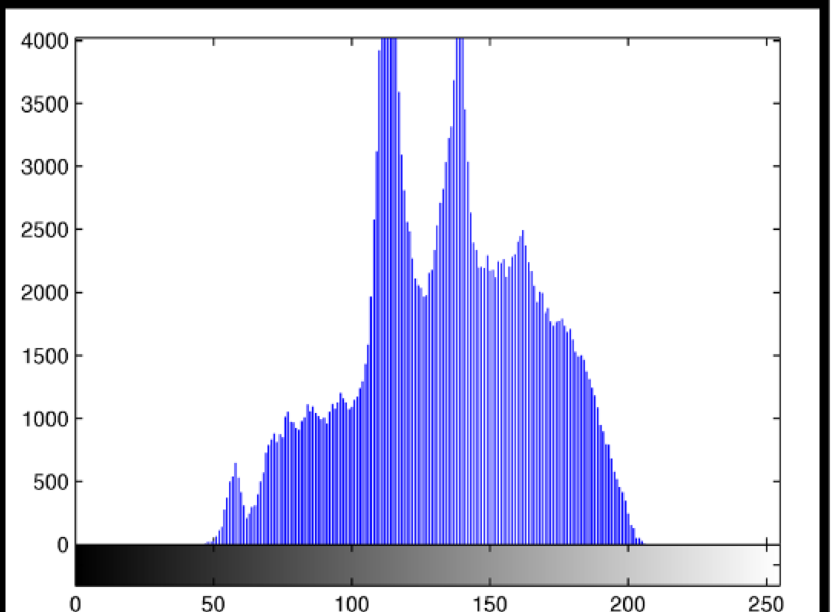
Using contrast

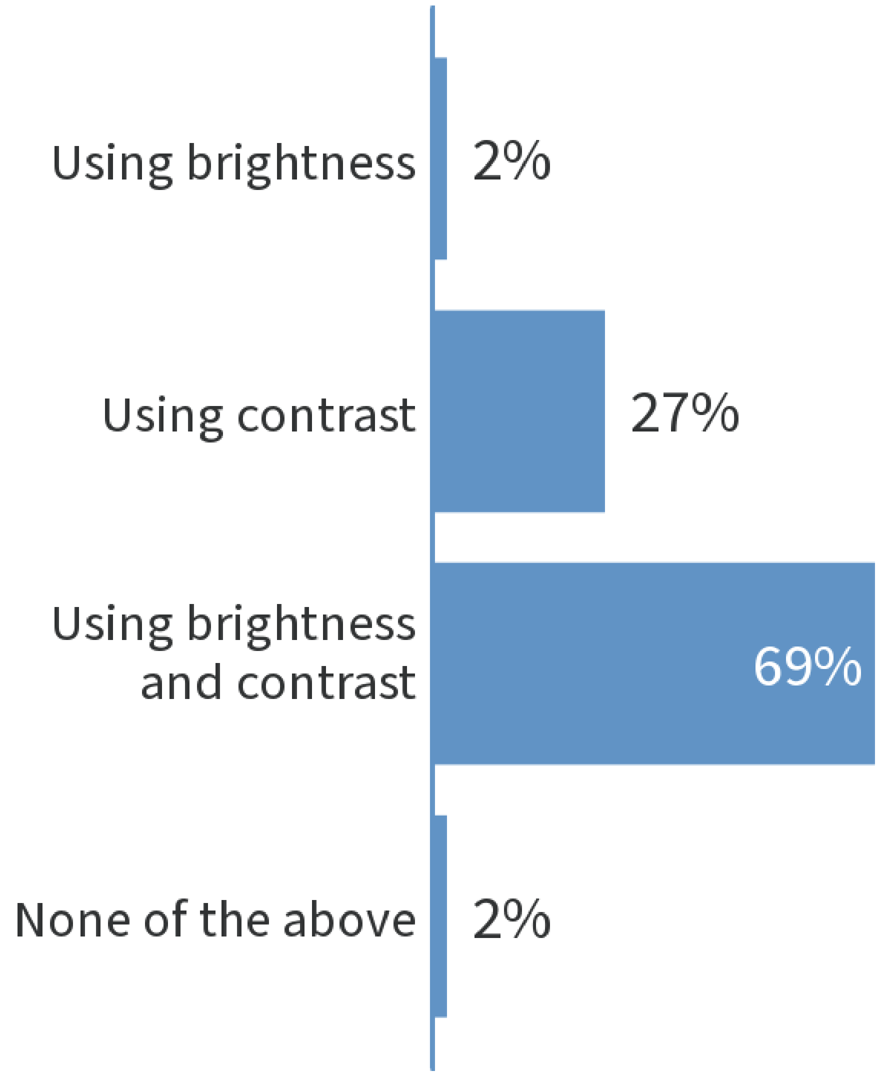
Using brightness and contrast

None of the above

Histogram stretching

- We want
  - Min = 0
  - Max = 255
- We have
  - Min = 32
  - Max = 208

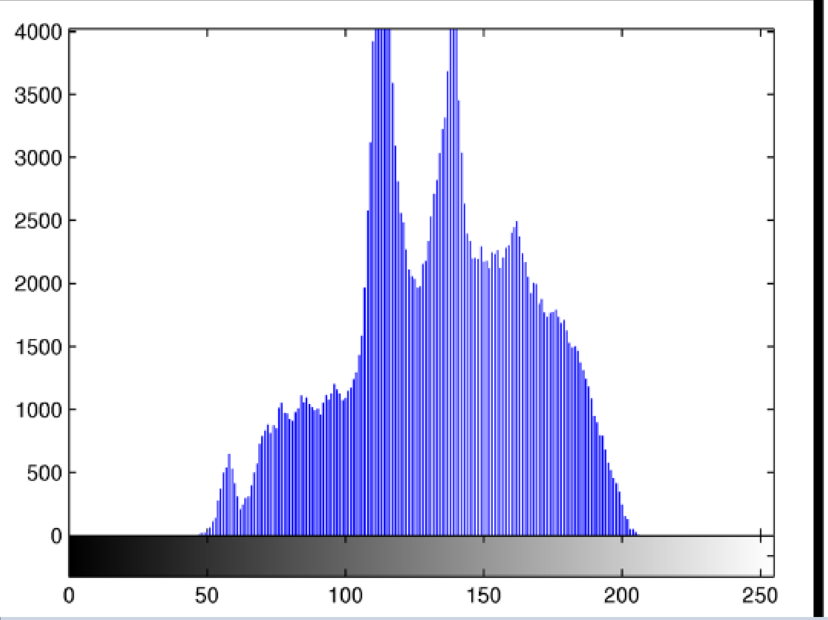


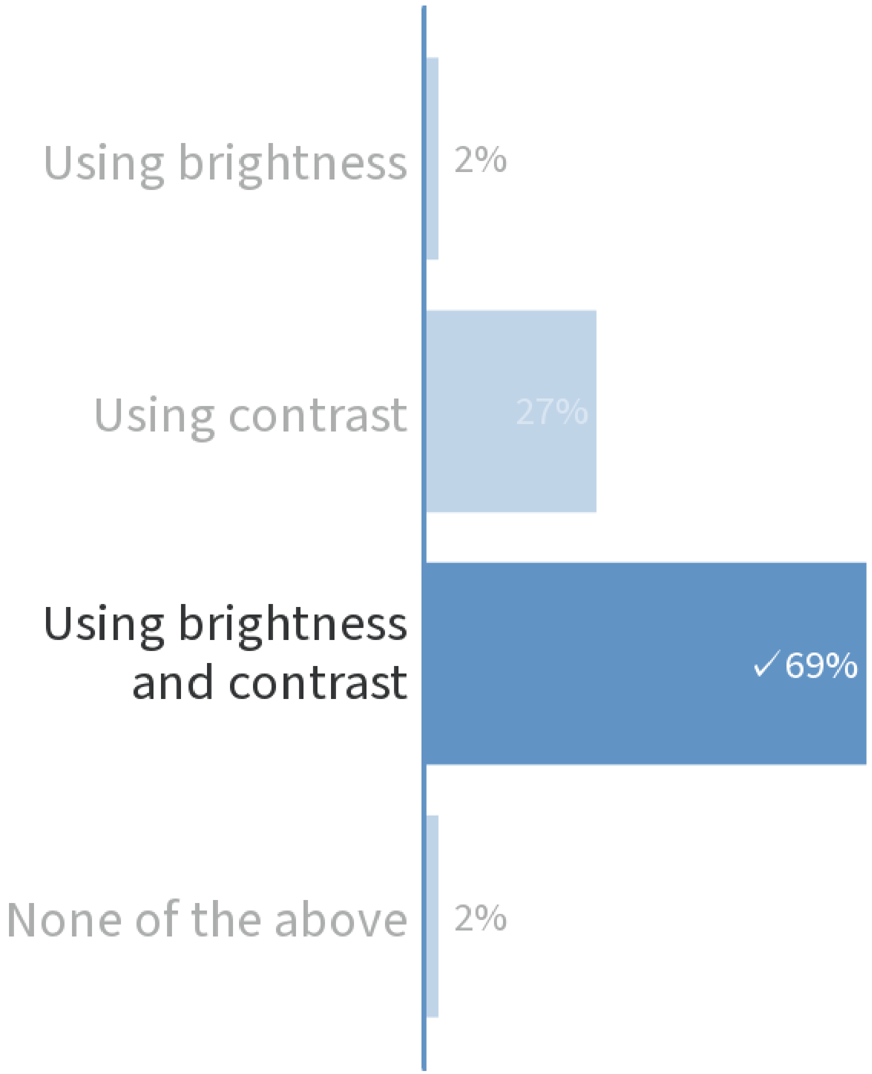


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

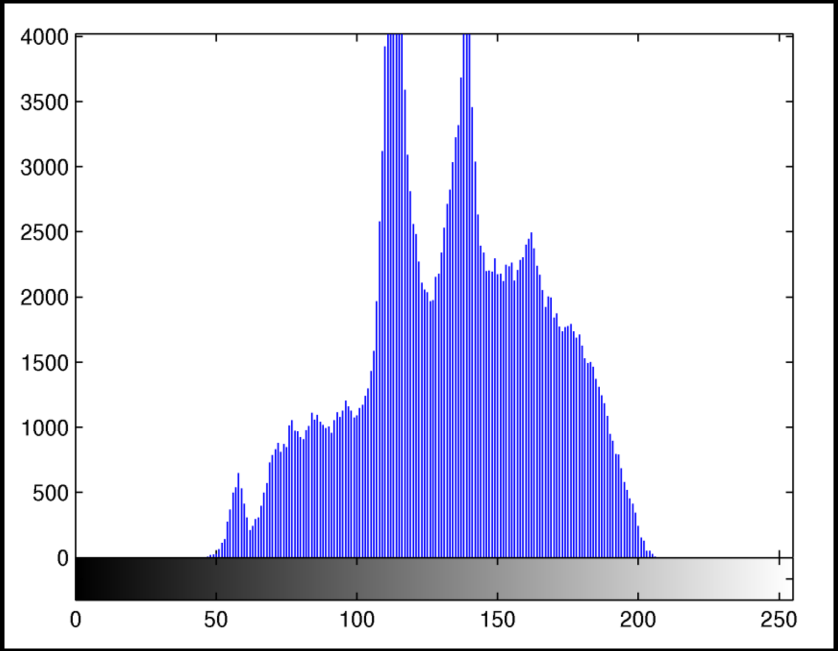
Histogram stretching

- We want
  - Min = 0
  - Max = 255
- We have
  - Min = 32
  - Max = 208





Histogram stretching



- We want
  - Min = 0
  - Max = 255
- We have
  - Min = 32
  - Max = 208

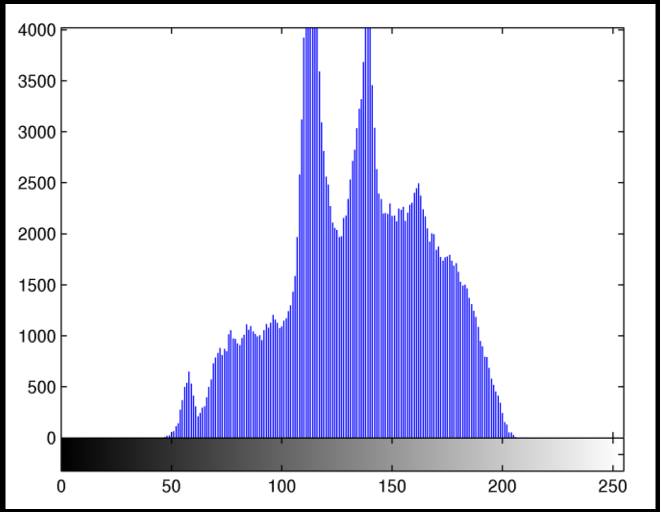
$$g(x,y) = \frac{v_{max,d} - v_{min,d}}{v_{max} - v_{min}}(f(x,y) - v_{min}) + v_{min,d}$$

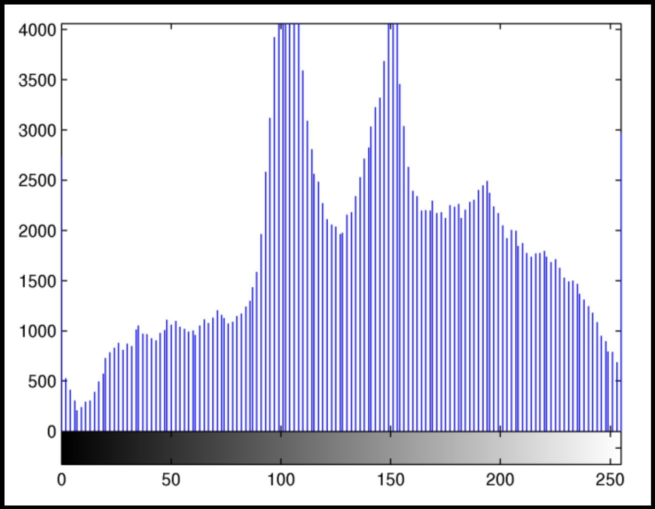
Histogram stretching formula

$$g(x,y) = \frac{v_{max,d} - v_{min,d}}{v_{max} - v_{min}}(f(x,y) - v_{min}) + v_{min,d}$$

- Desired min value $v_{min,d} = 0$
- Desired max value $v_{max,d} = 255$
- Current min value $v_{min} = 32$
- Current max value $v_{max} = 208$

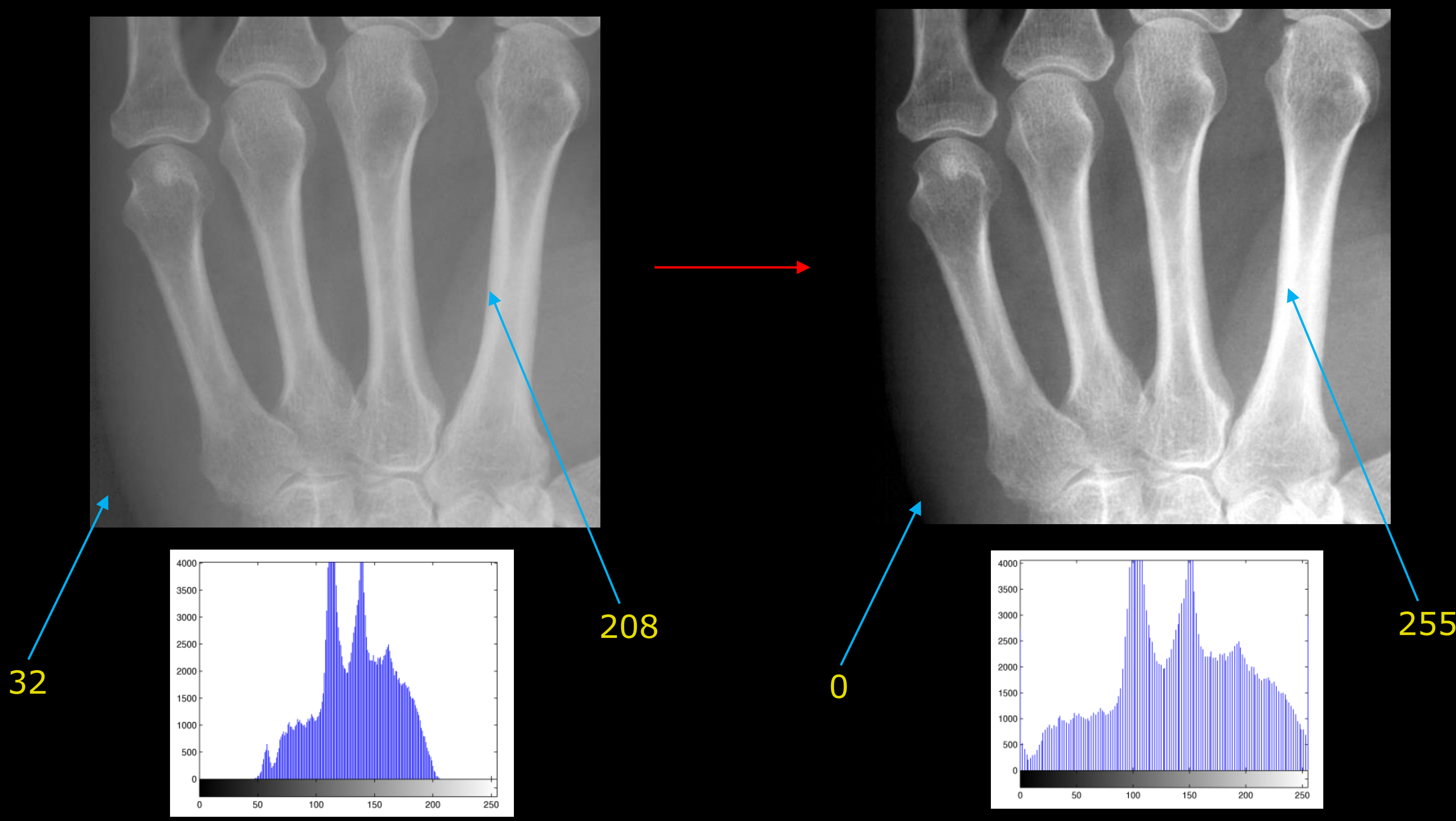
Histogram stretching





$$g(x,y) = \frac{255}{176}(f(x,y) - 32)$$

Effect of histogram stretching



Histogram stretching – weaknesses

- A single pixel value of 0 or 255 ruins it
- Sometimes you want
  - To stretch only the high pixel values
  - While “compressing” the low pixel values
  - Non-linear mapping

Linear mapping on an image

A linear mapping is performed on the image below. The mapping is performed so the mapped image has a maximum value of 255 and a minimum value of 0. What is the new value in the marked pixel?

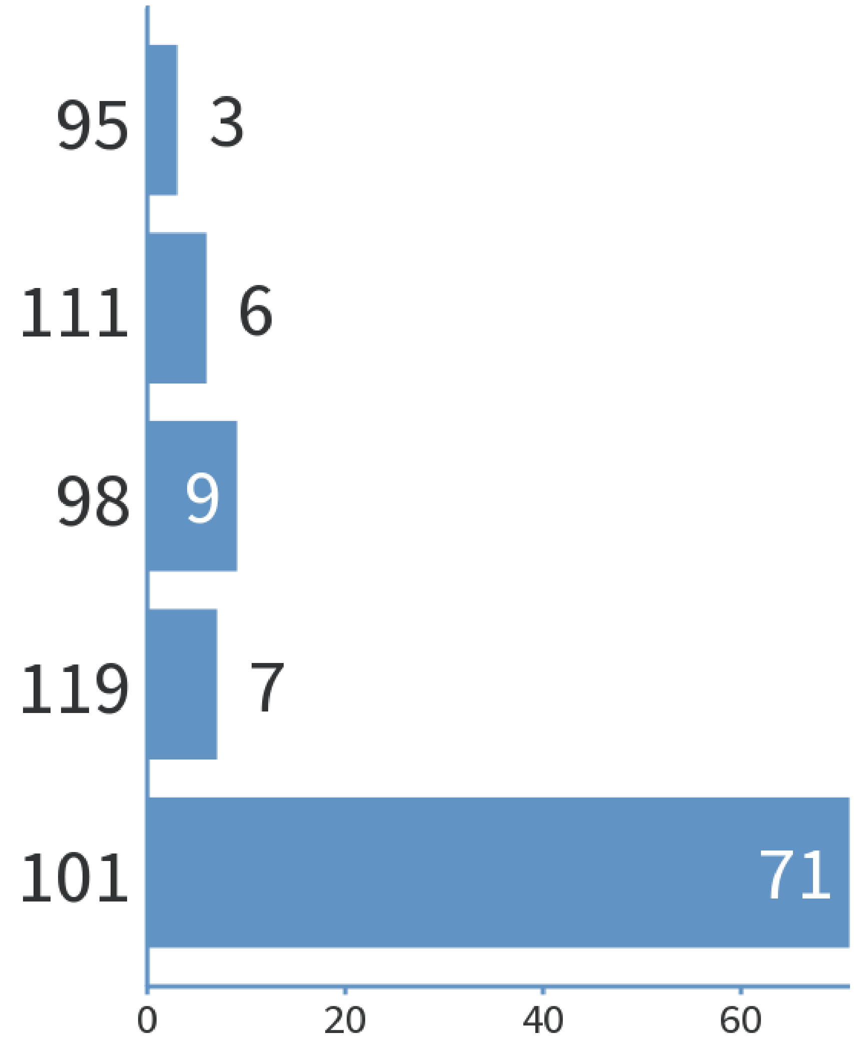
208	25	40	36	167
231	71	23	108	18
32	139	244	234	217
233	244	124	202	238
161	245	204	245	173

- 95
- 111
- 98
- 119
- 101

Linear mapping on an image

A linear mapping is performed on the image below. The mapping is performed so the mapped image has a maximum value of 255 and a minimum value of 0. What is the new value in the marked pixel?

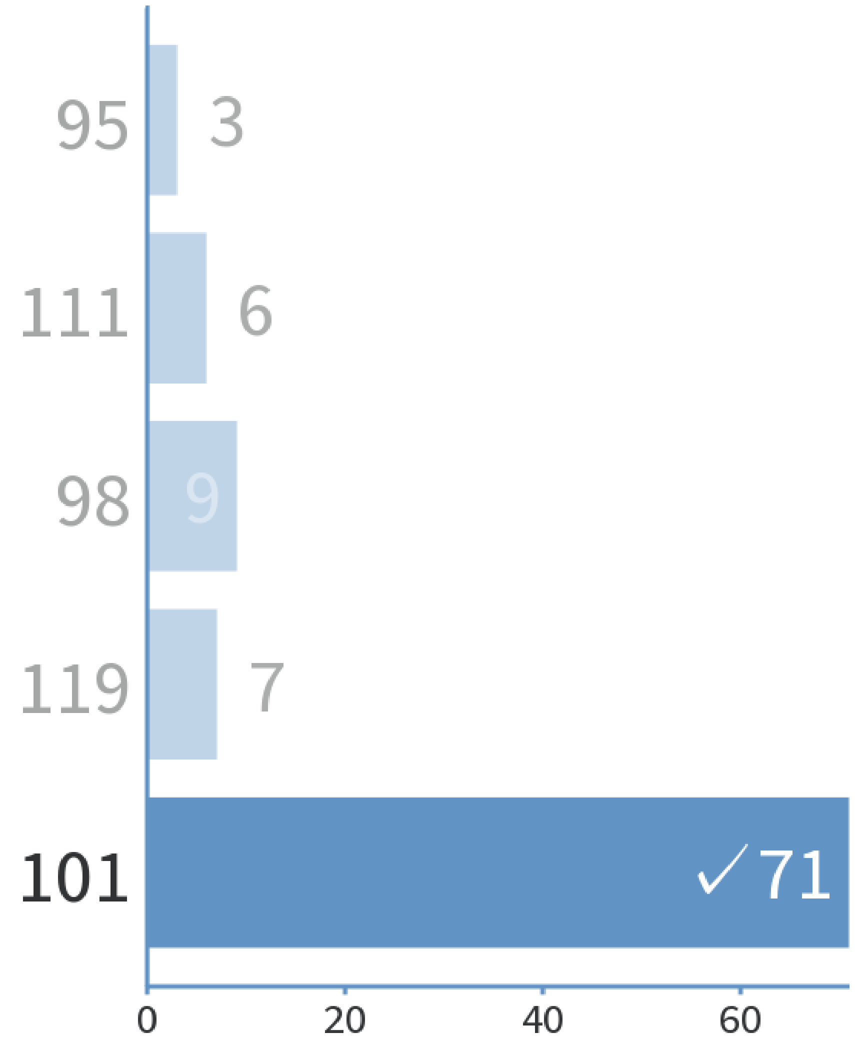
208	25	40	36	167
231	71	23	108	18
32	139	244	234	217
233	244	124	202	238
161	245	204	245	173



Linear mapping on an image

A linear mapping is performed on the image below. The mapping is performed so the mapped image has a maximum value of 255 and a minimum value of 0. What is the new value in the marked pixel?

208	25	40	36	167
231	71	23	108	18
32	139	244	234	217
233	244	124	202	238
161	245	204	245	173

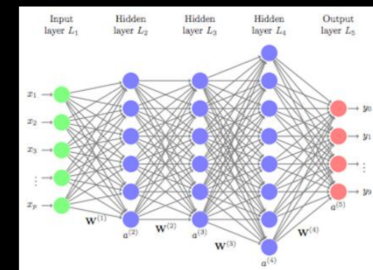


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

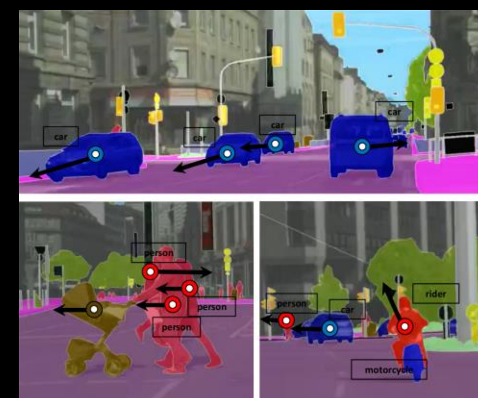
Deep learning and color/gray scale transformations

- Deep learning needs training data
  - Input image
  - Ground truth labels or classes

- When you lack data you can *augment* your data
  - Create artificial versions
  - Adding variation
  - Changing gray / color levels in the image
  - Point wise operations



http://uc-r.github.io/feedforward_DNN



Luc, Pauline, et al. "Predicting deeper into the future of semantic segmentation." IEEE International Conference on Computer Vision (ICCV). Vol. 1. 2017.

https://www.quora.com/What-does-the-term-semantic-segmentation-mean-in-the-context-of-Deep-Learning

Other mappings

- Non-linear mappings
- Not always nice to work with byte images
  - Better to work with image with values in [0,1]

Byte image

Conversion to [0,1]

Non-linear transformation

Back to bytes

Byte image

Working with bytes and doubles

- A byte contains integer values [0,255]
  - A byte can not store 127.4232
- A value of type *double* can contain “all numbers”
- Why not use doubles always?
  - One double = 8 bytes in the memory
  - Images become very large!
  - Many things can be done with bytes

Map pixels to [0,1]

- In Matlab it is easiest to create a new image of type double
  - Itemp = double(I);
  - (temp means temporary and is used by many programmers for variables that quickly are thrown away)
- Conversion to [0,1]

$$g(x,y) = \frac{1}{255} f(x,y)$$

Pixels back to bytes

- Input pixels are [0,1]
- We want them to be [0,255]
- Simple linear transformation equal to
  - Contrast?
  - Brightness?

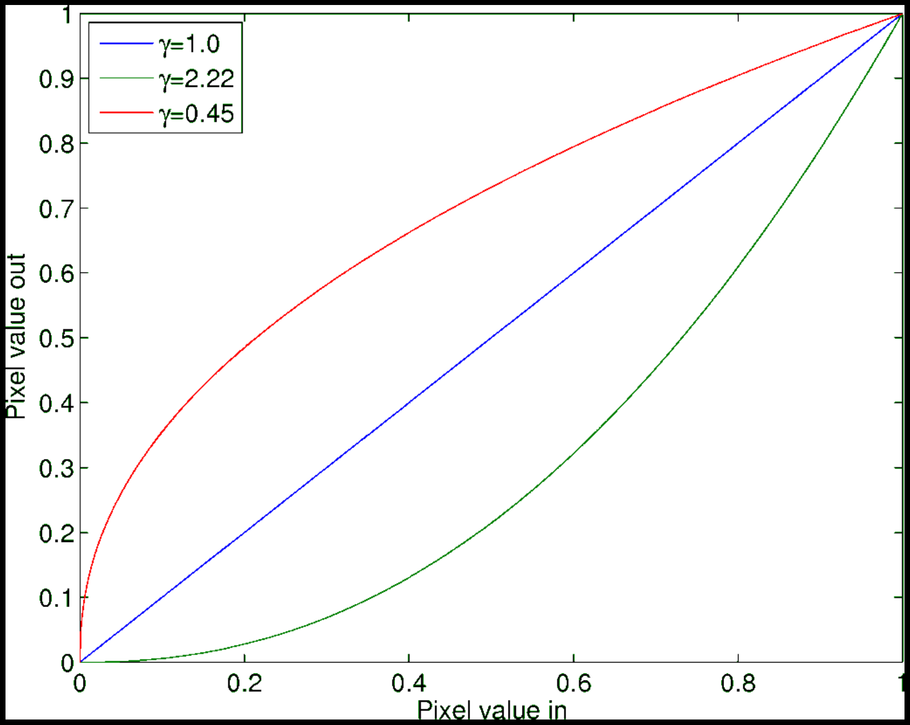
$$g(x, y) = 255 * f(x, y)$$

- Back to bytes
  - Ifinal = uint8(Itemp);

Gamma mapping

- Gamma mapping is used in televisions and flat panels
- Can increase the contrast (dynamics) in more selected part of the histogram
- Many games have a possibility for a gamma correction

Gamma curves



- Named after the Greek letter gamma
- What happens to the dark areas
  - With 0.45?
  - With 2.22?

$$g(x,y) = f(x,y)^{\gamma}$$

Perform the gamma mapping

Byte image

Conversion to [0,1]

Gamma mapping

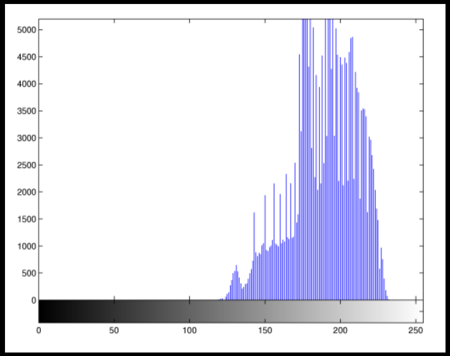
Back to bytes

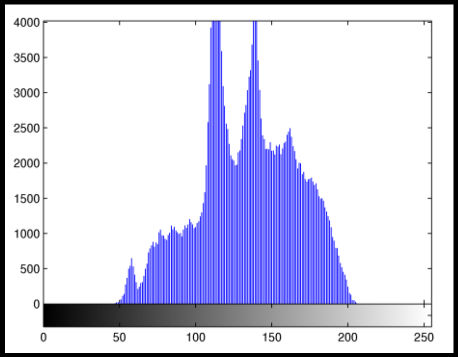
Byte image

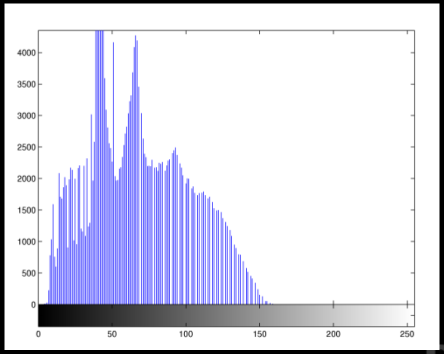
Results of gamma mapping

0.45

2.22







Gamma mapping on an image

A gamma mapping is performed on the image below with $\gamma = 1.3$. What is the minimum and maximum value in the mapped image?

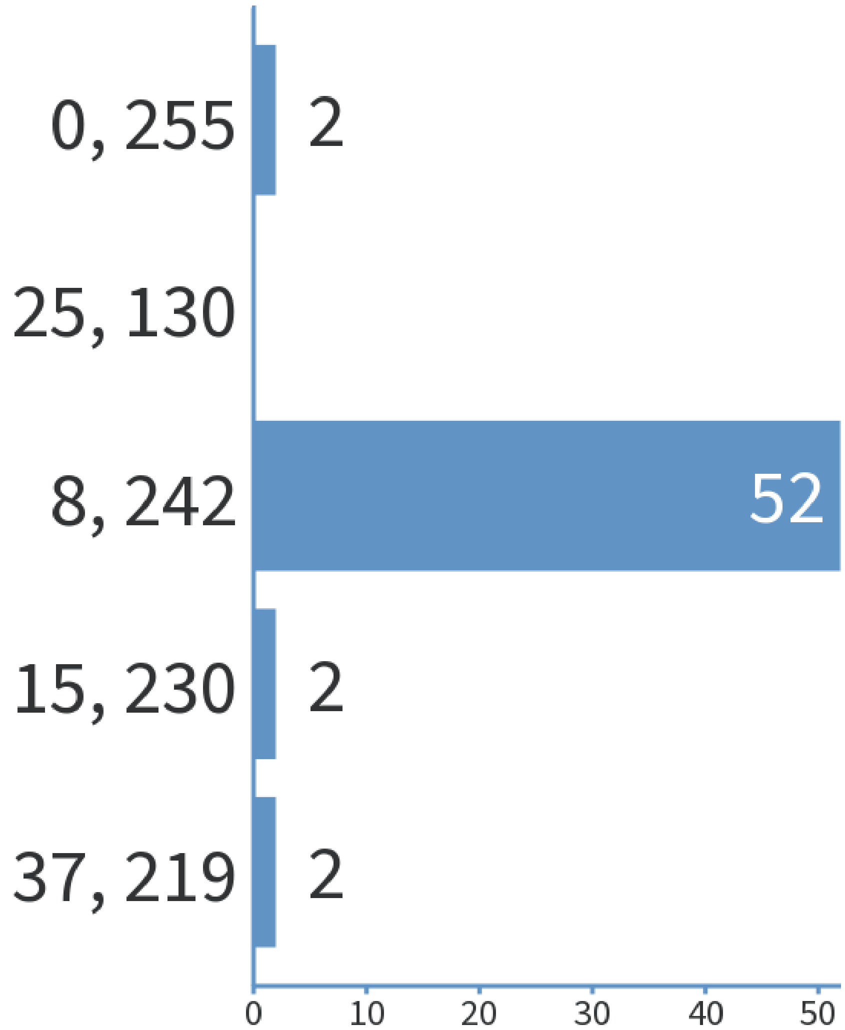
208	25	40	36	167
231	71	23	108	18
32	139	244	234	217
233	244	124	202	238
161	245	204	245	173

- 0, 255
- 25, 130
- 8, 242
- 15, 230
- 37, 219

Gamma mapping on an image

A gamma mapping is performed on the image below with $\gamma = 1.3$. What is the minimum and maximum value in the mapped image?

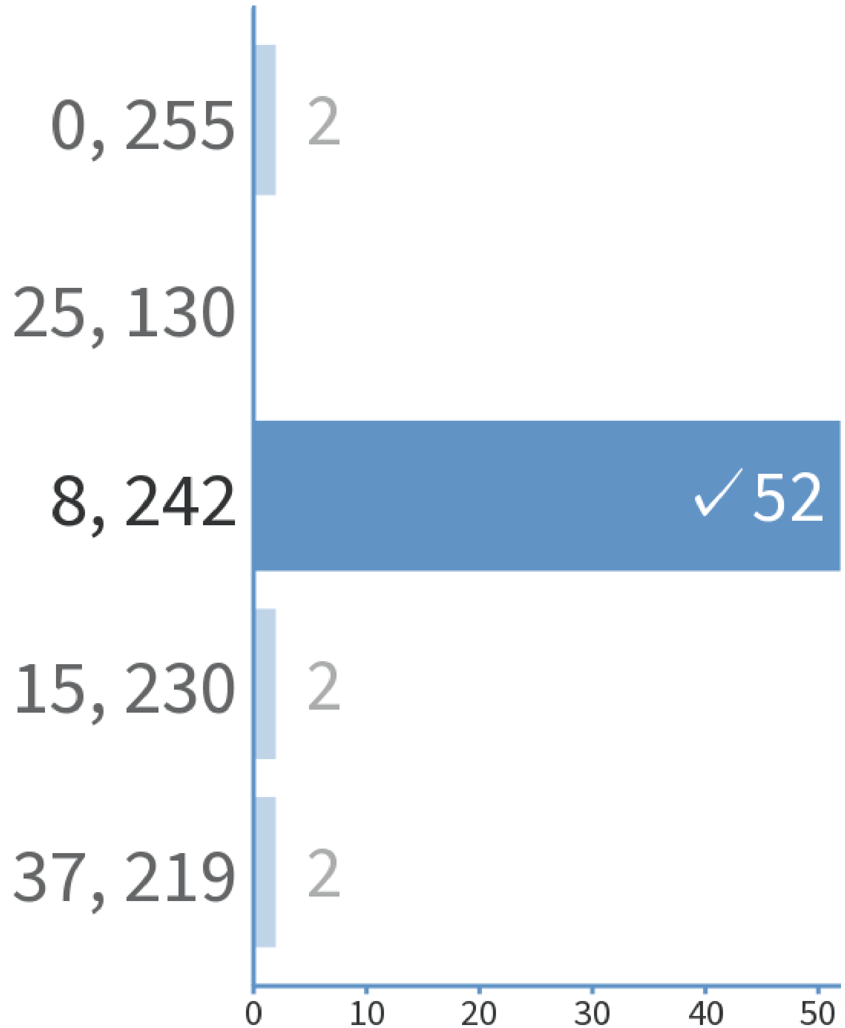
208	25	40	36	167
231	71	23	108	18
32	139	244	234	217
233	244	124	202	238
161	245	204	245	173



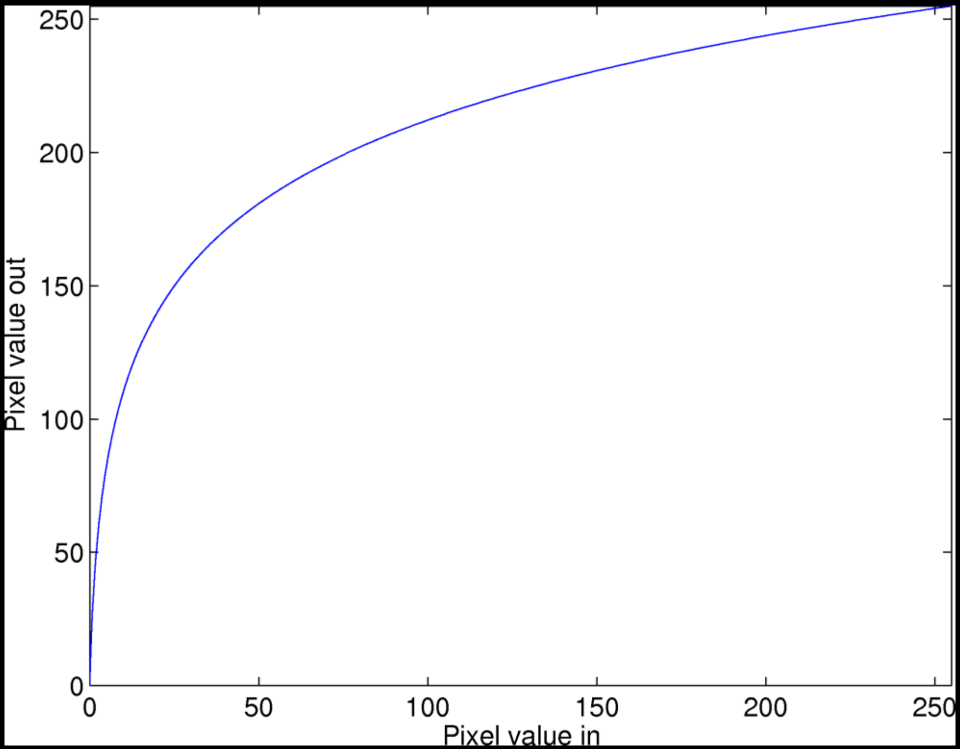
Gamma mapping on an image

A gamma mapping is performed on the image below with $\gamma = 1.3$. What is the minimum and maximum value in the mapped image?

208	25	40	36	167
231	71	23	108	18
32	139	244	234	217
233	244	124	202	238
161	245	204	245	173



Logarithmic mapping



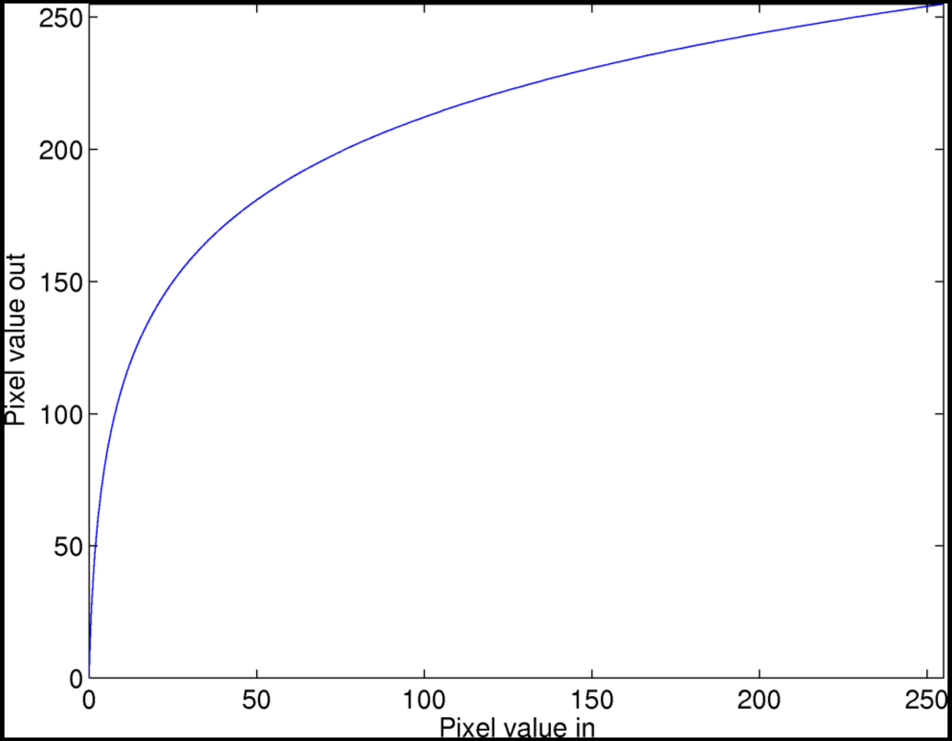
Maps from [0,255] to [0,255]

Why?

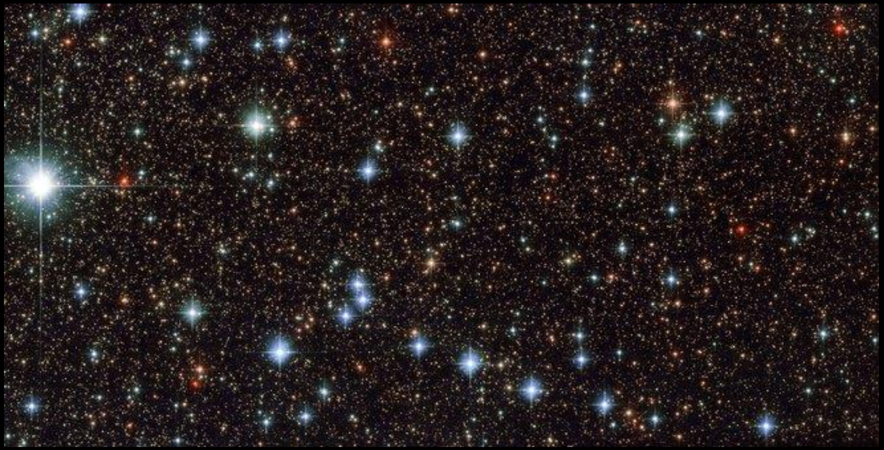
$$g(x,y) = c \log\left(1 + \mathrm{f}(\mathrm{x},\mathrm{y})\right)$$

$$c = \frac{255}{\log(1 + v_{max})}$$

Logarithmic mapping – when?



- For images with very bright spots
- Low intensity pixel values are enhanced

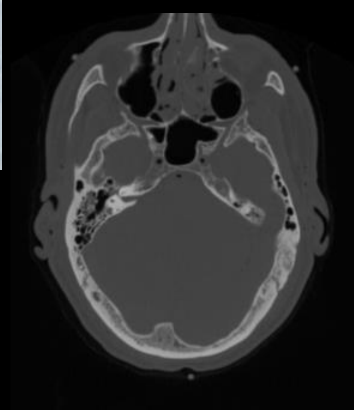


What do we get out of pixel mappings

- Spreading out or compressing pixel values
  - Better for humans to see
  - New information – no!

Now for something different

- Until now image processing
  - Input image transformed to output image
- Now for something more like image analysis
- Segmentation
  - Segment the image into regions
    - Background and objects for example

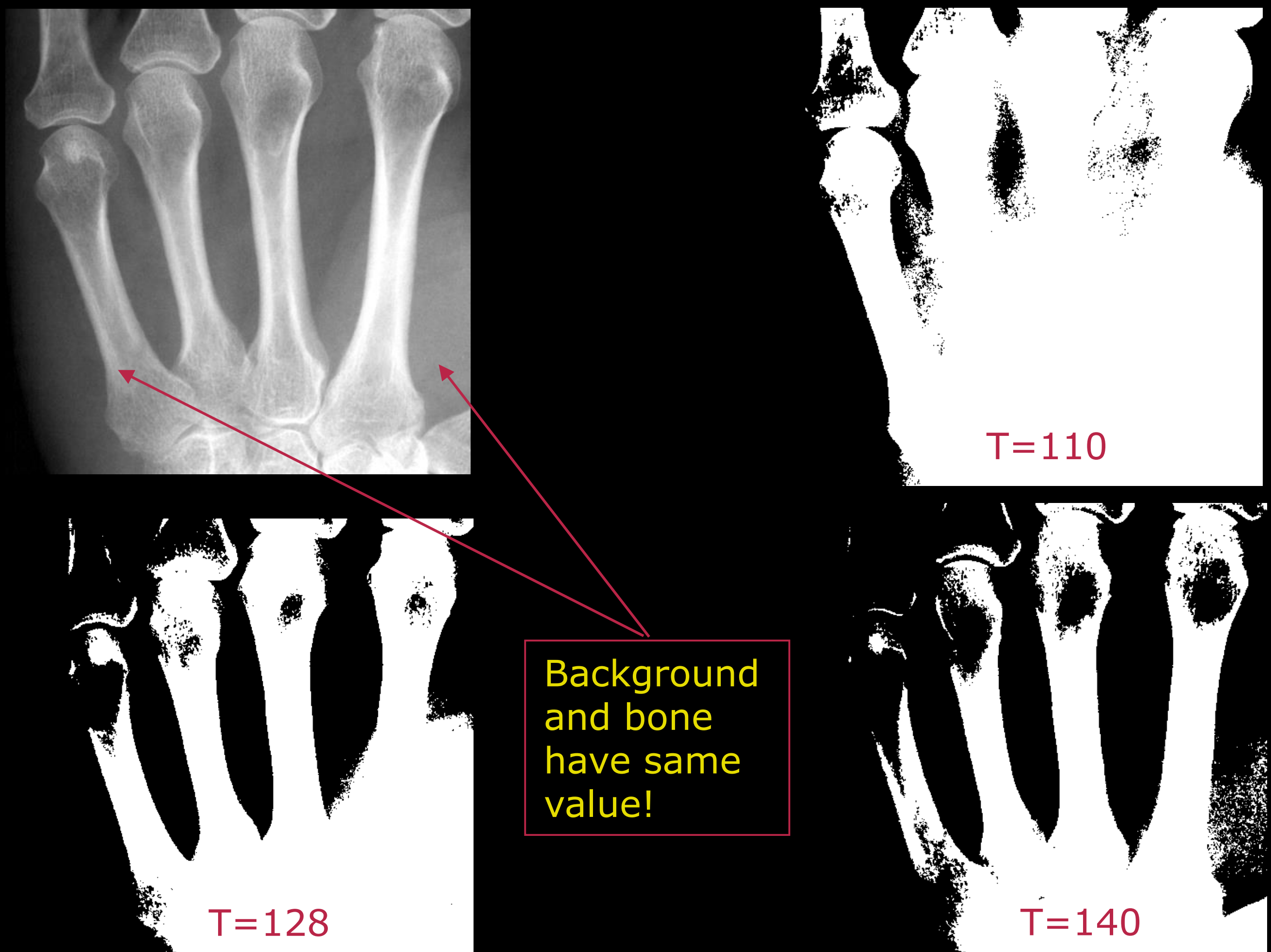
Thresholding

- A threshold *T* is a value
  - Pixels below that value is set to 0 (background)
  - Pixels equal or above is set to 1 (object)
- One threshold value for the entire image
  - Difficult to choose!

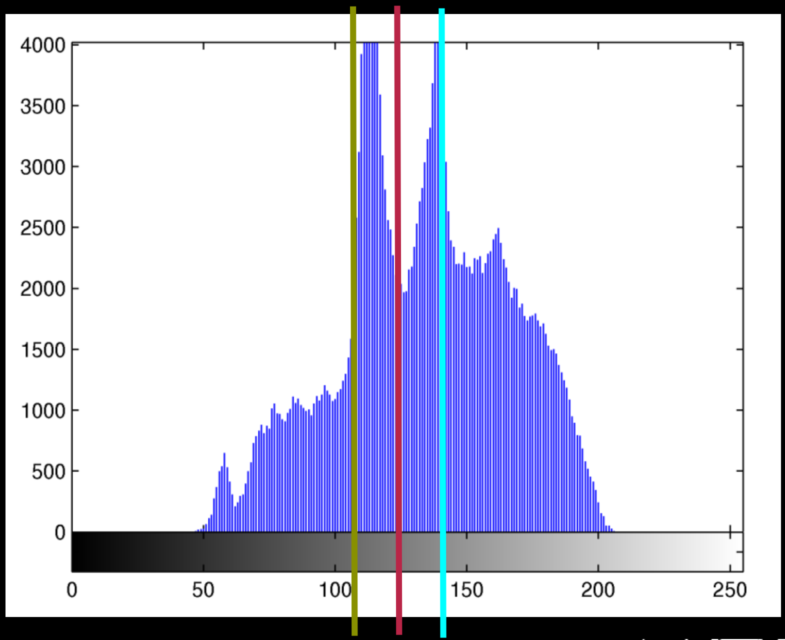
$$\text{if } f(x,y) \leq T \ \text{ then } g(x,y) = 0$$

$$\text{if } f(x,y) > T \ \text{ then } g(x,y) = 255$$

Thresholding

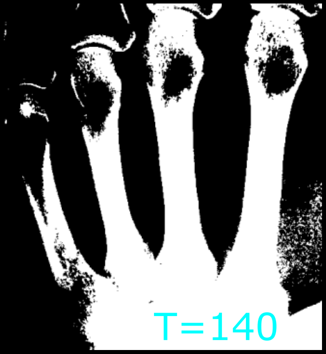


Thresholding based on the histogram

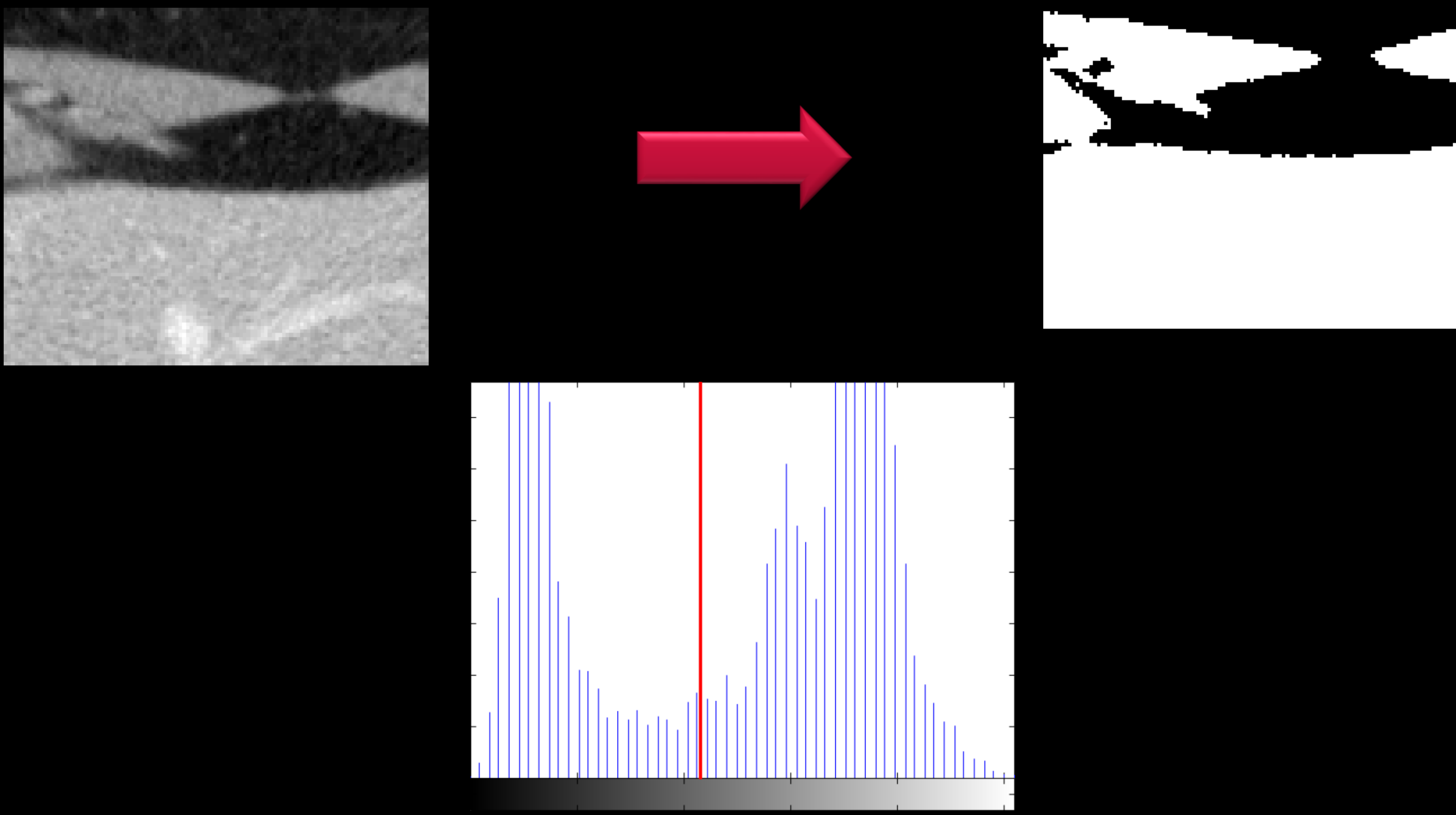


The bones are visible in the histogram!

But mixed with soft-tissue

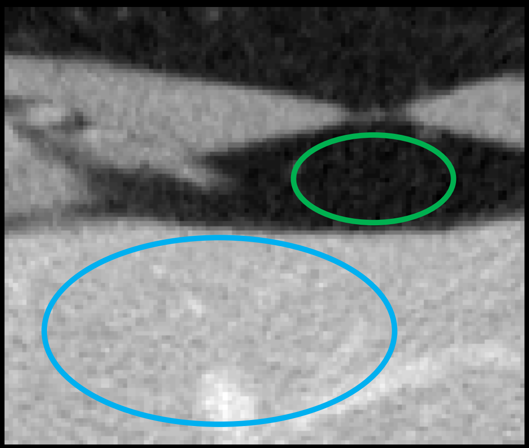


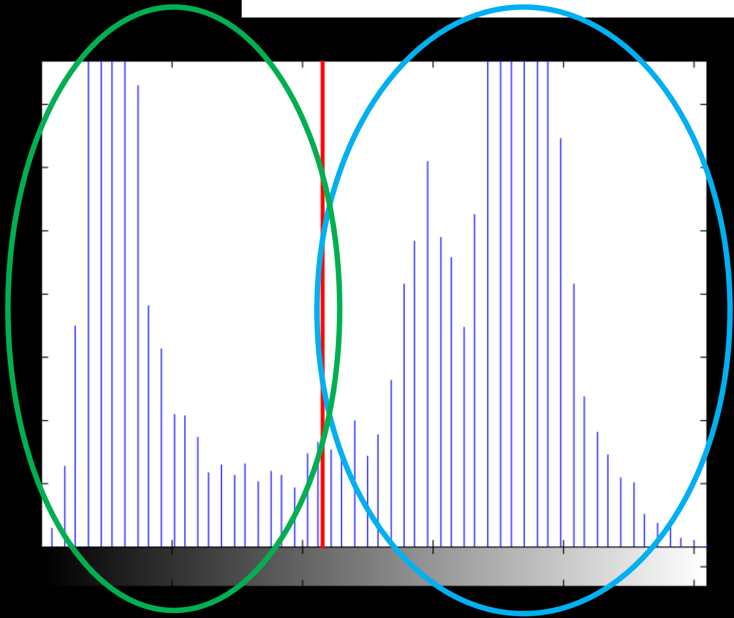
Automatic Tresholding



Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." IEEE transactions on systems, man, and cybernetics 9.1 (1979): 62-66.

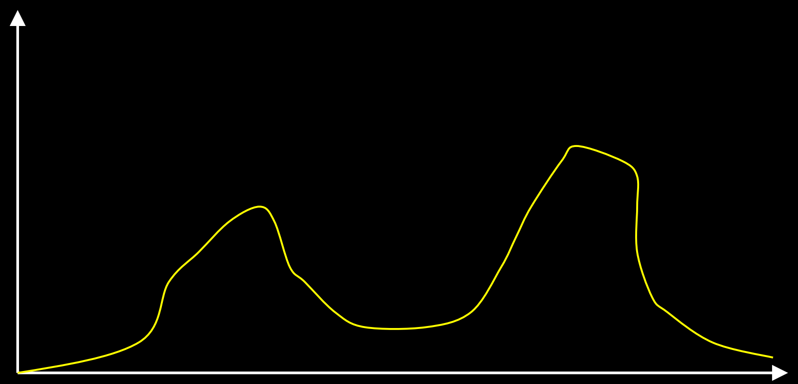
Automatic Tresholding
Otsu’s method

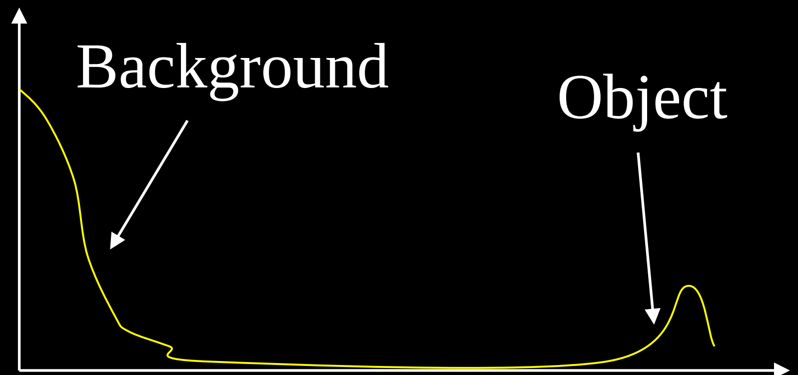




- Two classes: background and object
- T divides pixels into object and background
- Compute pixel value variance in each class
- Find T that minimises combined variance

Segmentation – histogram shaping

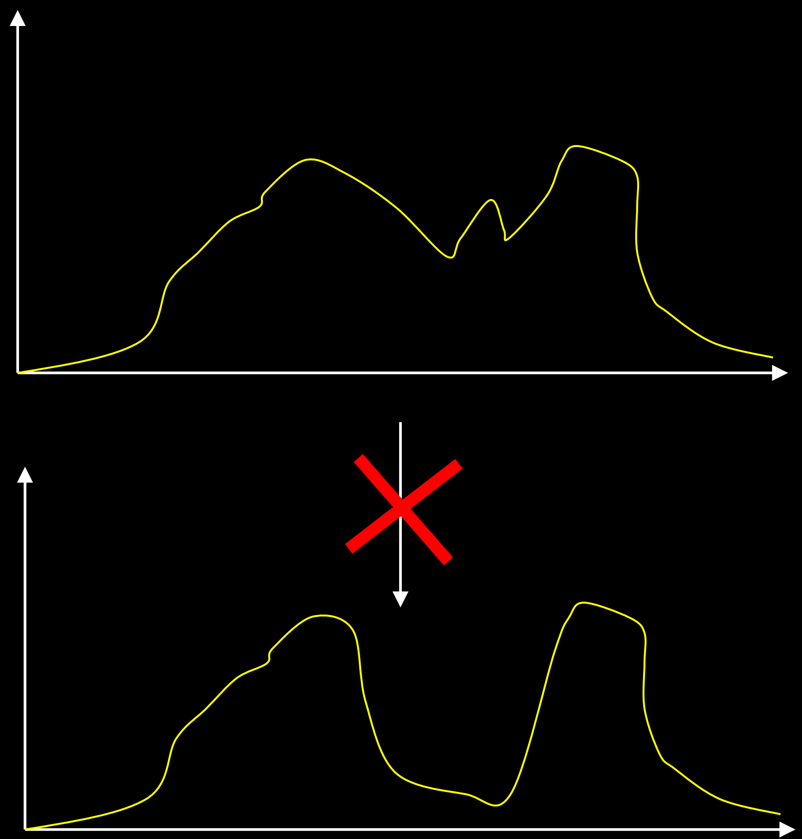




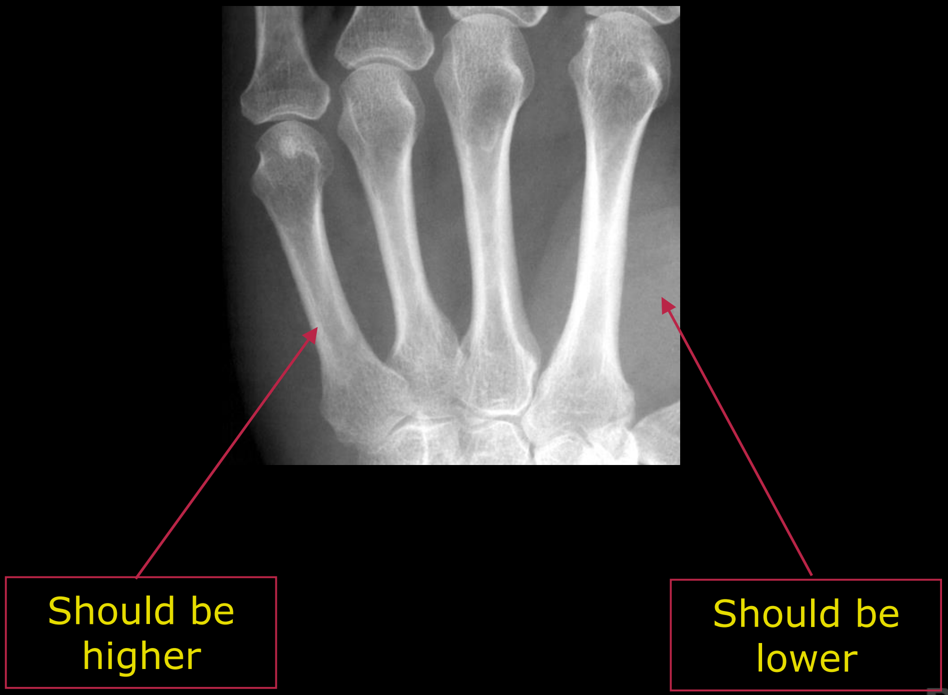
Ideal

- With a threshold you want a histogram with two peaks
  - *Bimodal*
- An ideal histogram has well separated peaks
- Obtaining a bi-modal histogram is very important in the image acquisition

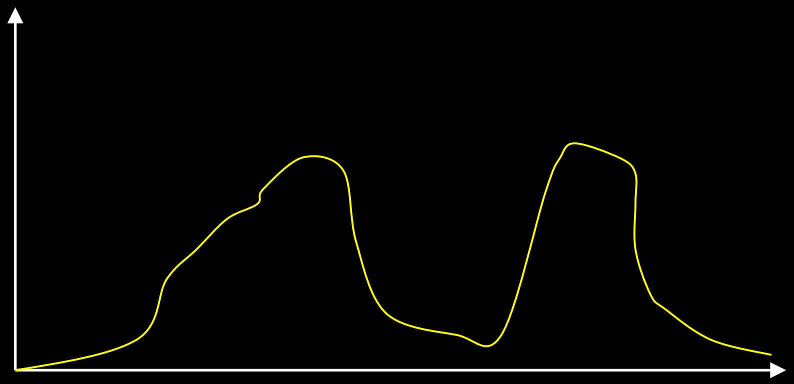
Histogram shaping



- It is not possible to “unmix” using gray level transformations

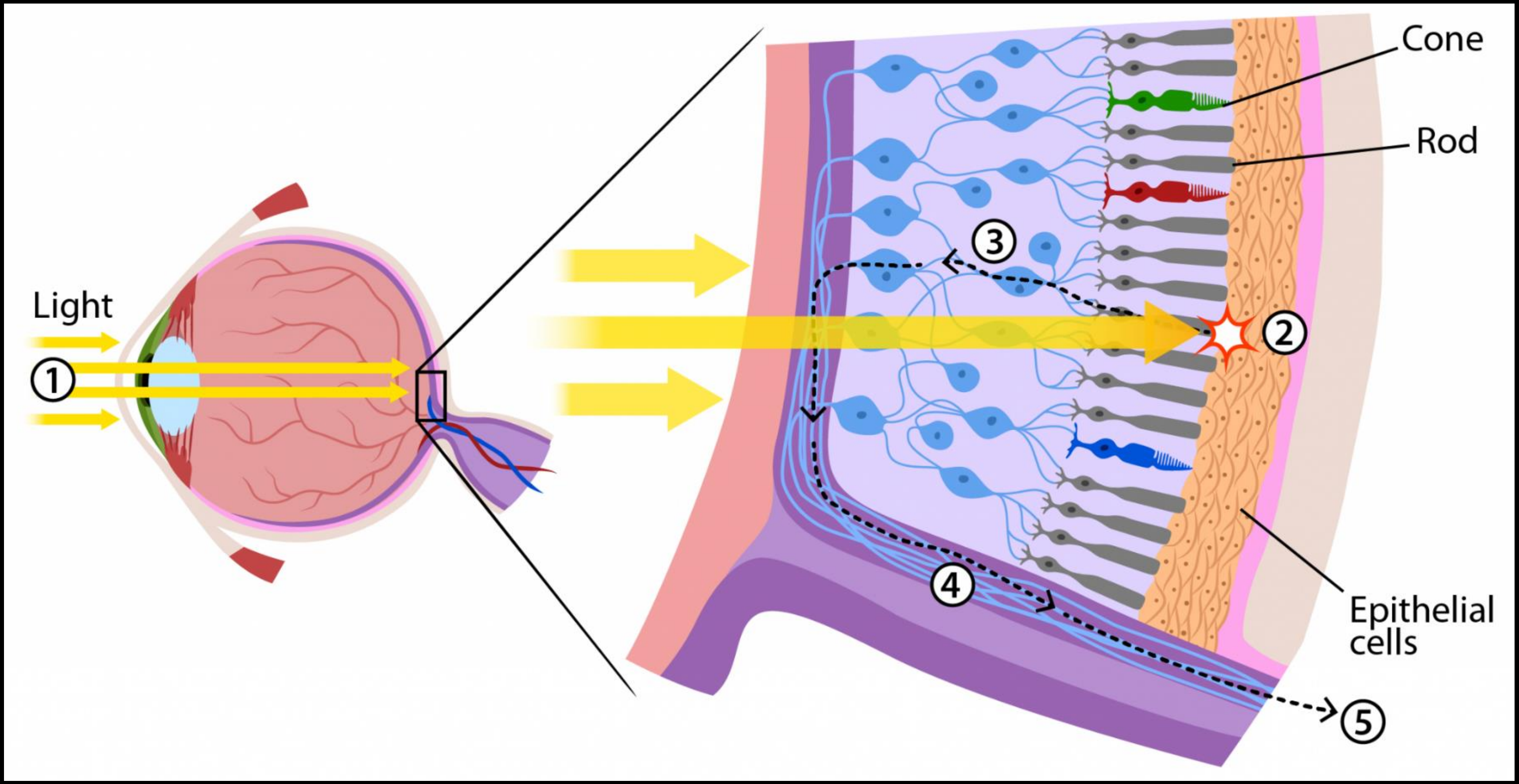


How to obtain good histograms



- With cameras
  - Light
  - Setup
  - Camera
  - Lens
  - Backlight?

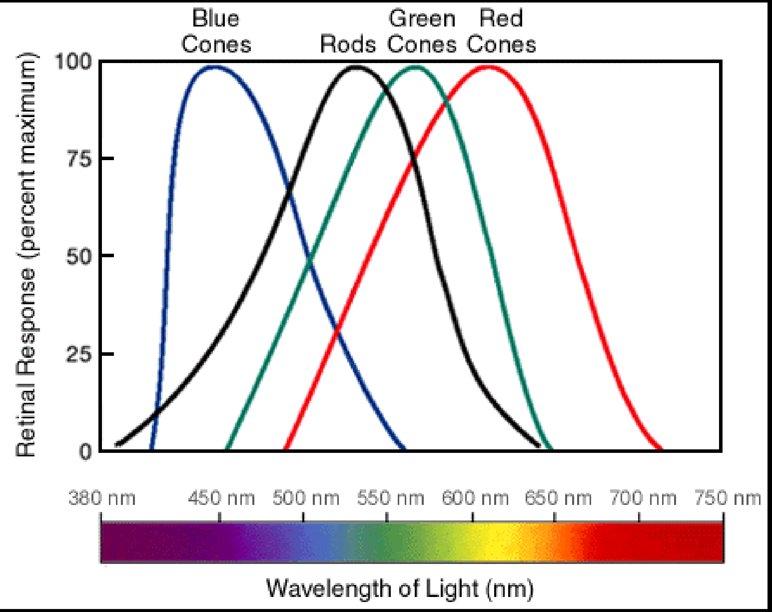
Colour images and colour perception
The Human Eye



https://askabiologist.asu.edu/rods-and-cones

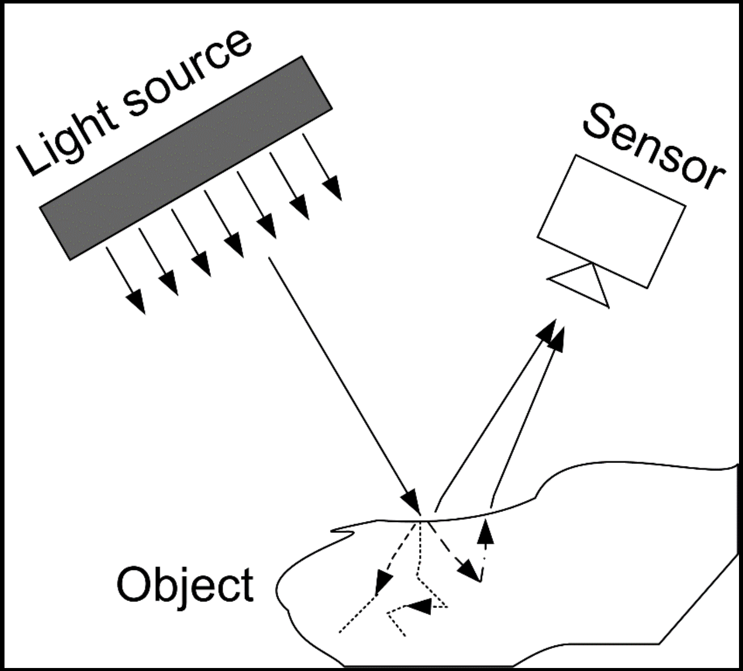
Color sensitivity

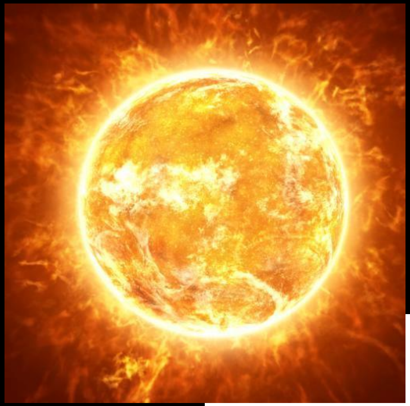
Photoreceptor cell	Wavelength in nanometers (nm)	Peak response in nanometer (nm)	Interpretation by the human brain
Cones (type L)	[400-680]	564	Red
Cones (type M)	[400-650]	534	Green
Cones (type S)	[370-530]	420	Blue
Rods	[400-600]	498	Shade of gray

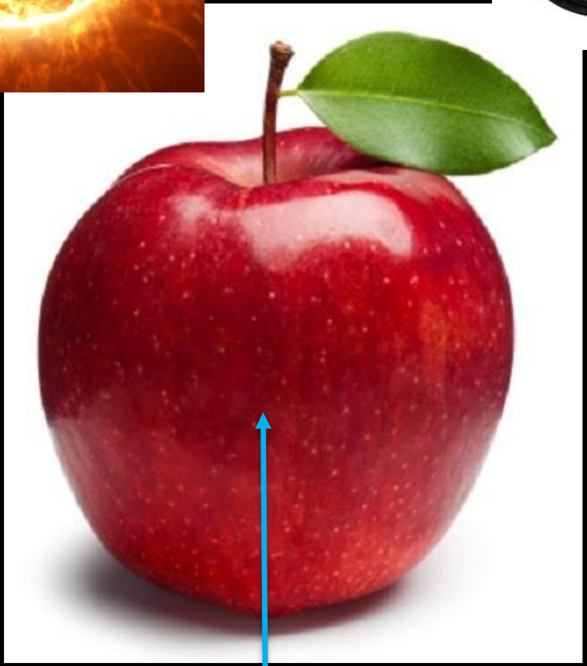


https://askabiologist.asu.edu/rods-and-cones

Object colors
Subtractive colors







All other colors than red absorbed

Object colors
Additive colors

- Additive colours: Final colour is made by mixing red, green, and blue
- RGB = Red, Green, and Blue
- Television, computers, digital cameras use the “RGB color space”
- Typically the values of R, G, and B lie between 0 and 255

RGB Colours

- When alle three “Lamps” are turned of we get black
- When all three “lamps” are on what do we get?

RGB = (0,0,0)

(255,0,0)

(0,255,0)

(0,0,255)

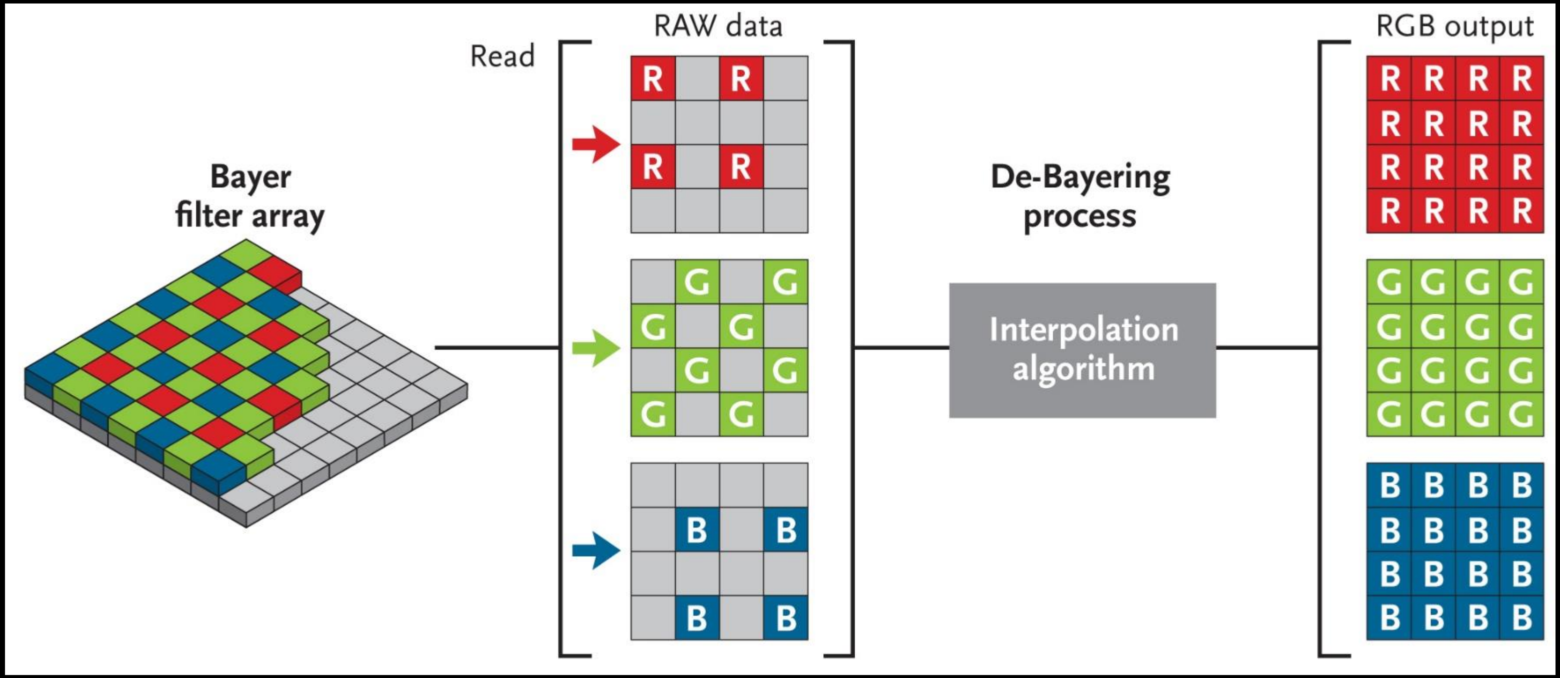
(255,255,0)

(0,255,255)

(255,0,255)

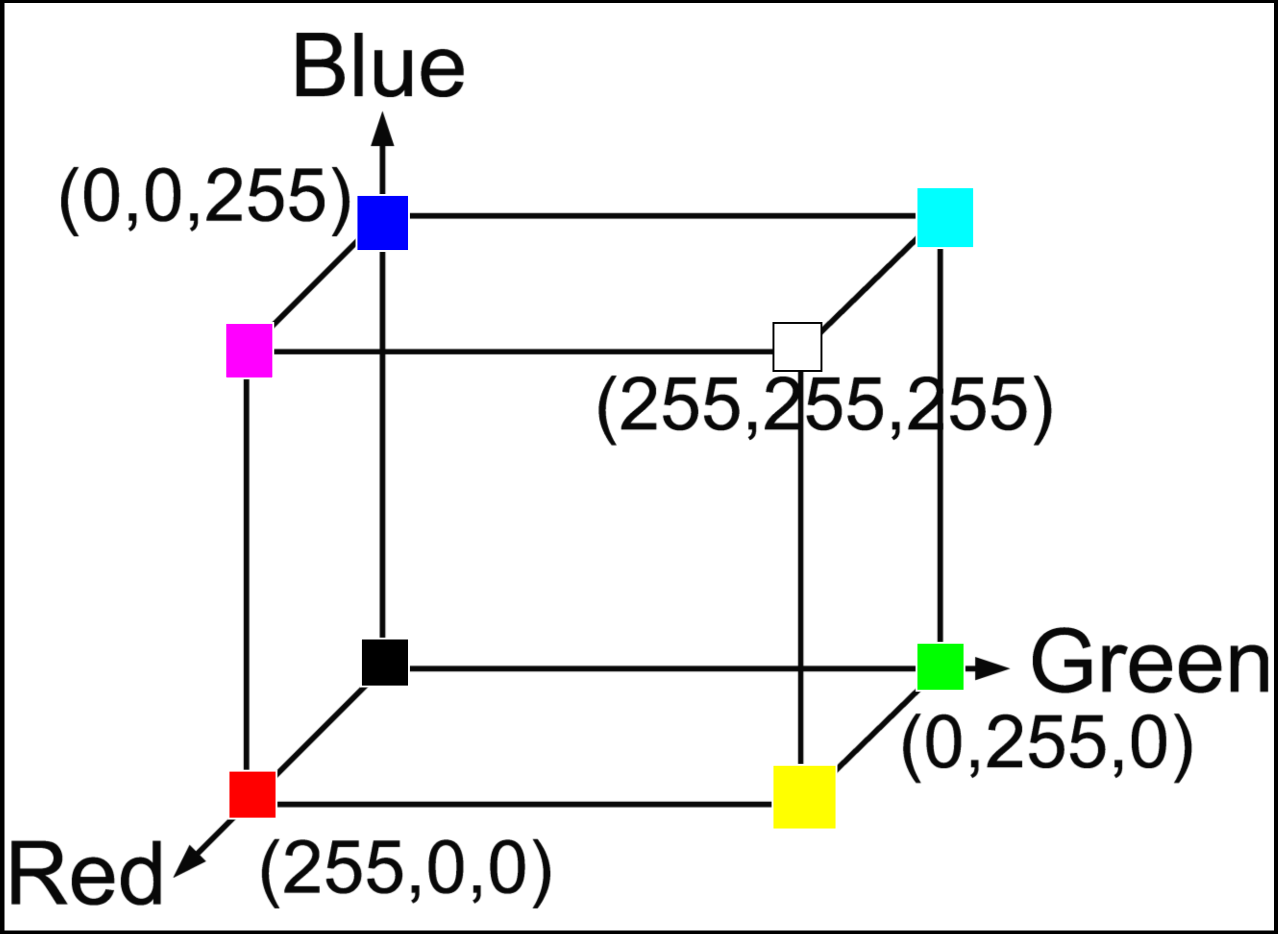
RGB = (255,255,255)

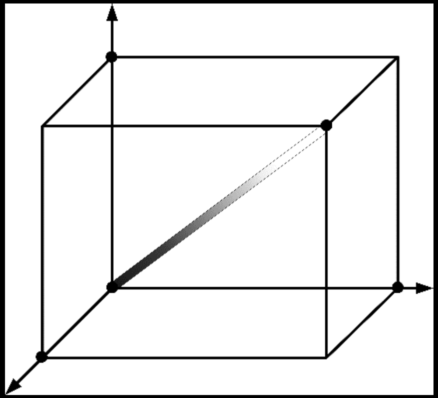
Color camera with one sensor



http://www.skyandtelescope.com/astronomy-resources/astrophotography-tips/redeeming-color-planetary-cameras/

RGB color space



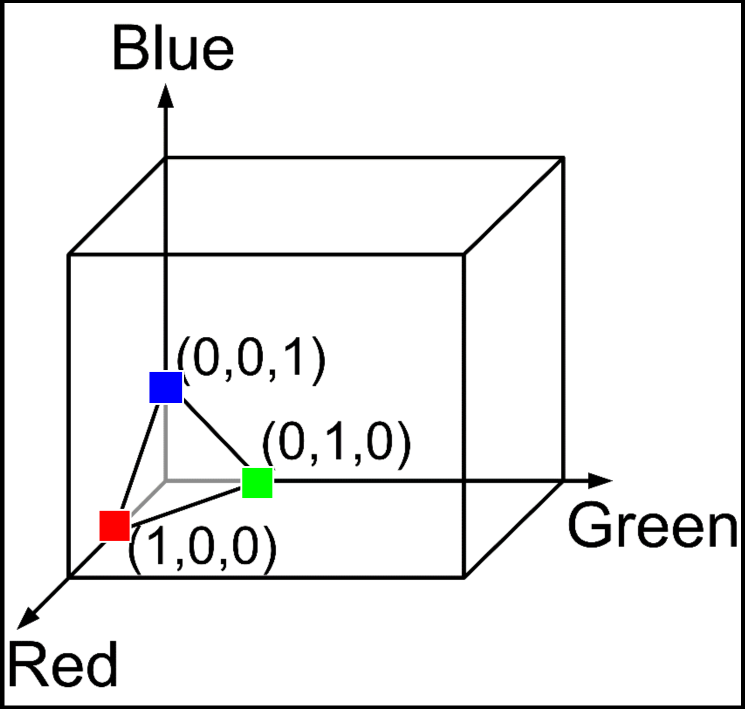


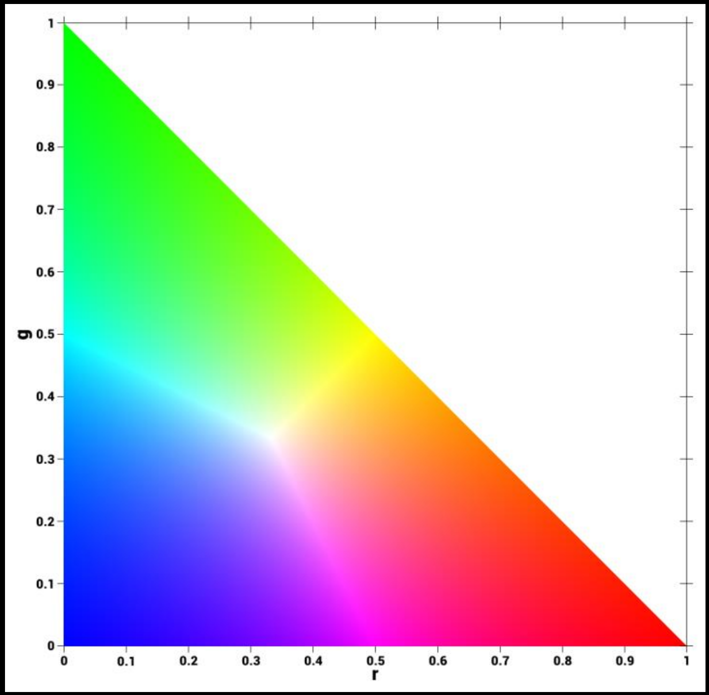
Converting colour to grayscale

$$v = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

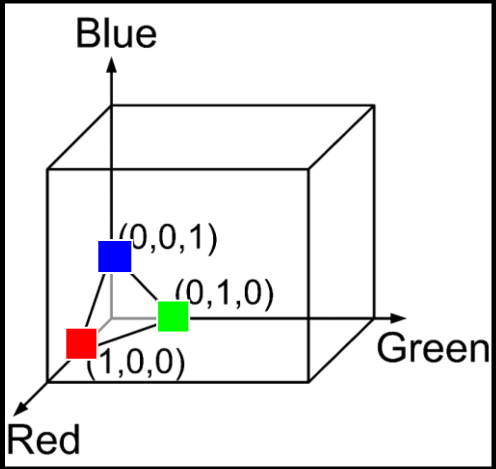
Normalised RGB colors

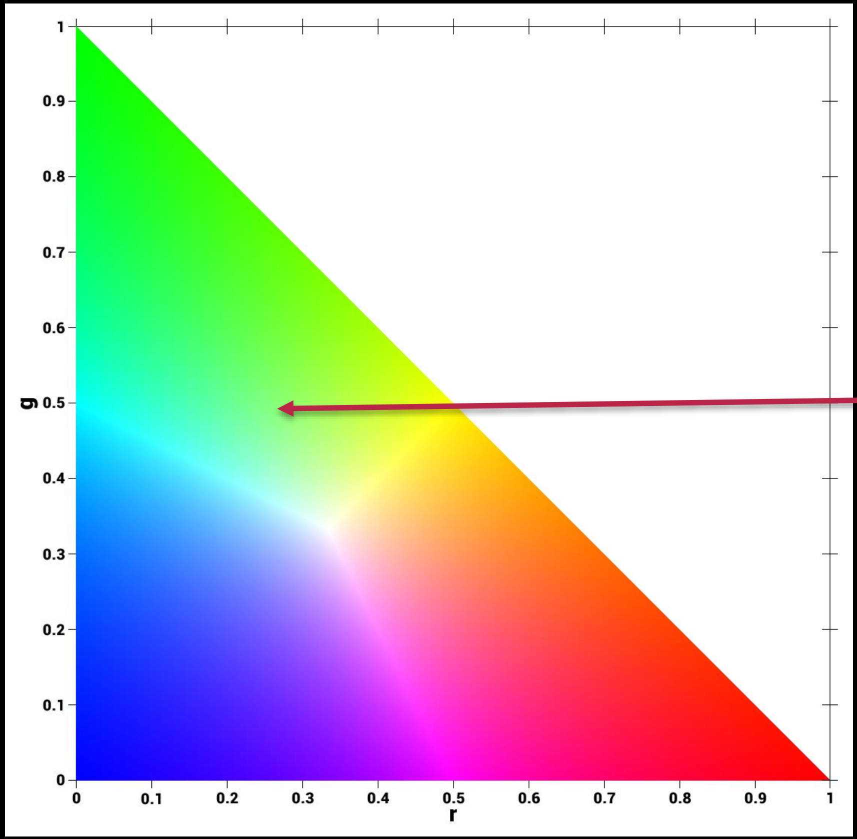
$$(r, g, b) = \left(\frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B} \right)$$





Another RGB representation



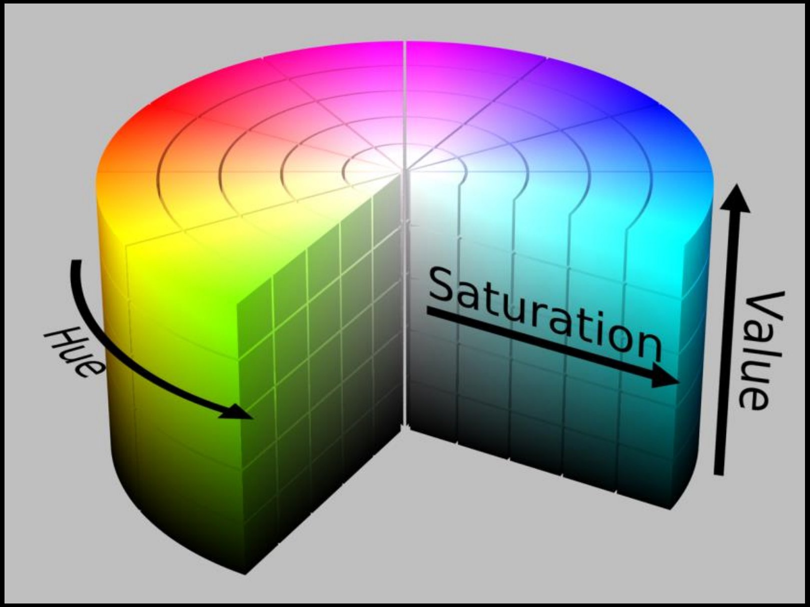


(r,g,I)

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

HSI Color Reprentation

- Hue – the dominant wave length in the perceived light (the pure color)
- Saturation – the purity of the color
- Intensity – the brightness of the color (sometimes called the value)



Converting between RGB and HSI

- You have an RGB value
- You want the corresponding HSI value

$$H = \begin{cases} \cos^{-1}\left(1/2 \cdot \frac{(R-G)+(R-B)}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}}\right), & \text{if } G \geq B; \\ 360^\circ - \cos^{-1}\left(1/2 \cdot \frac{(R-G)+(R-B)}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}}\right), & \text{Otherwise.} \end{cases} \quad (8.8)$$

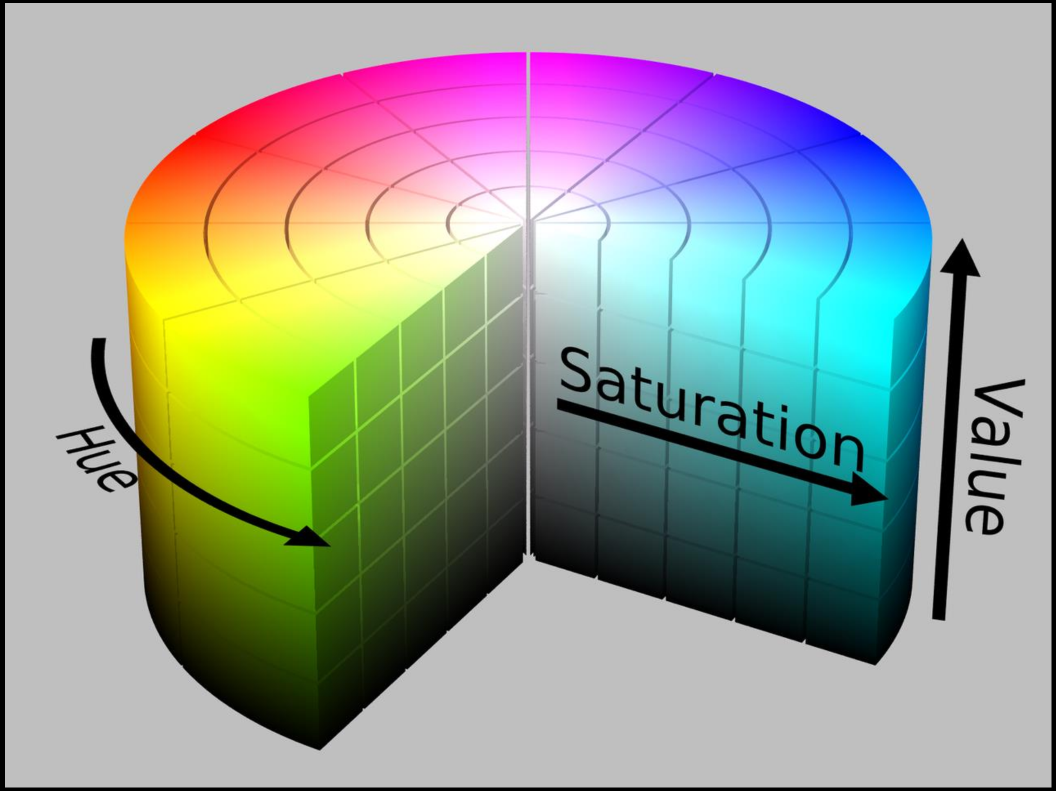
$$H \in [0, 360[$$

$$S = 1 - 3 \cdot \frac{\min\{R, G, B\}}{R+G+B} \qquad S \in [0, 1] \quad (8.9)$$

$$I = \frac{R+G+B}{3} \qquad I \in [0, 255] \ , \quad (8.10)$$

Why other colorspaces

- Why should we use for example HSI ?

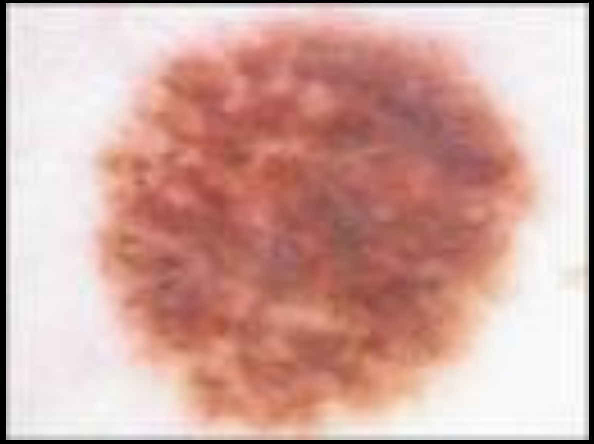


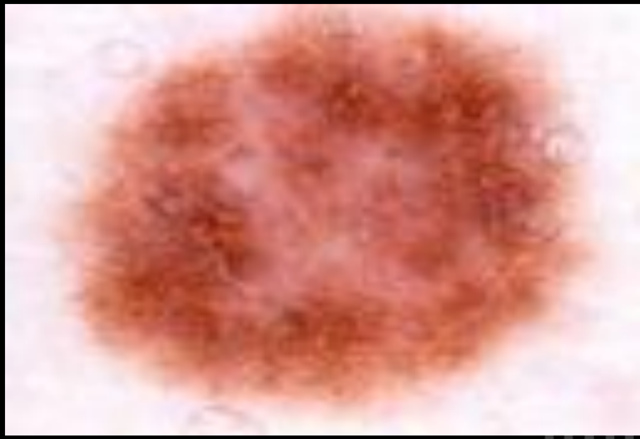
Melanoma segmentation

- An algorithm that can do pixelwise classification
  - Background / skin
  - Melanoma
- Use the colors

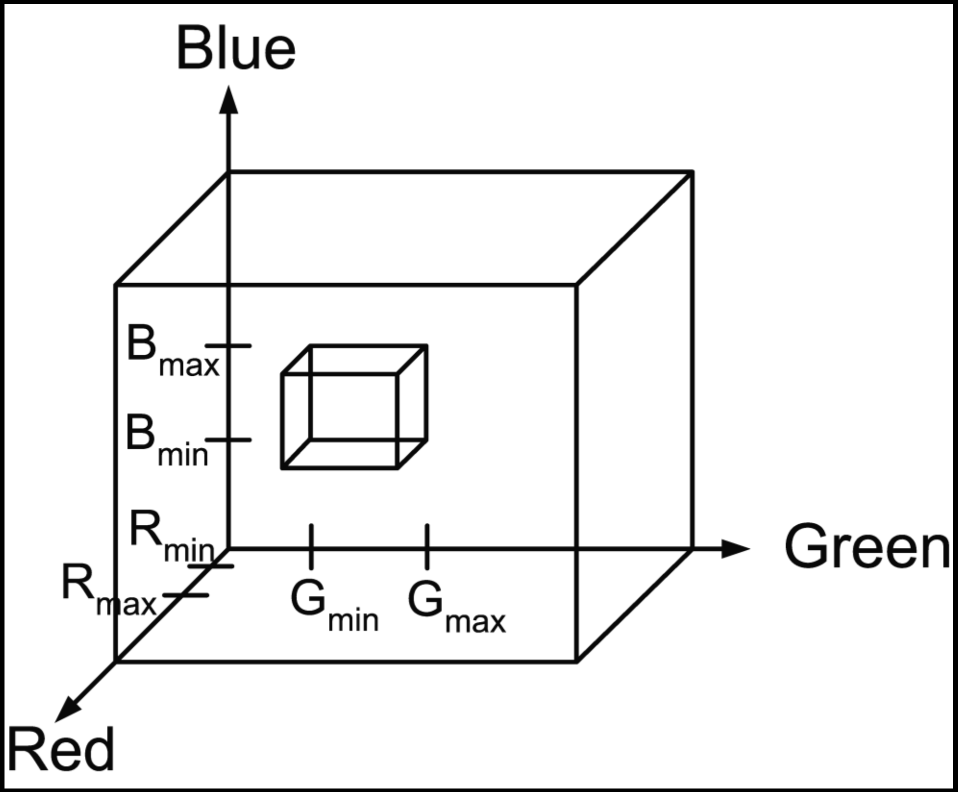
Melanoma segmentation – color variation





Color thresholding



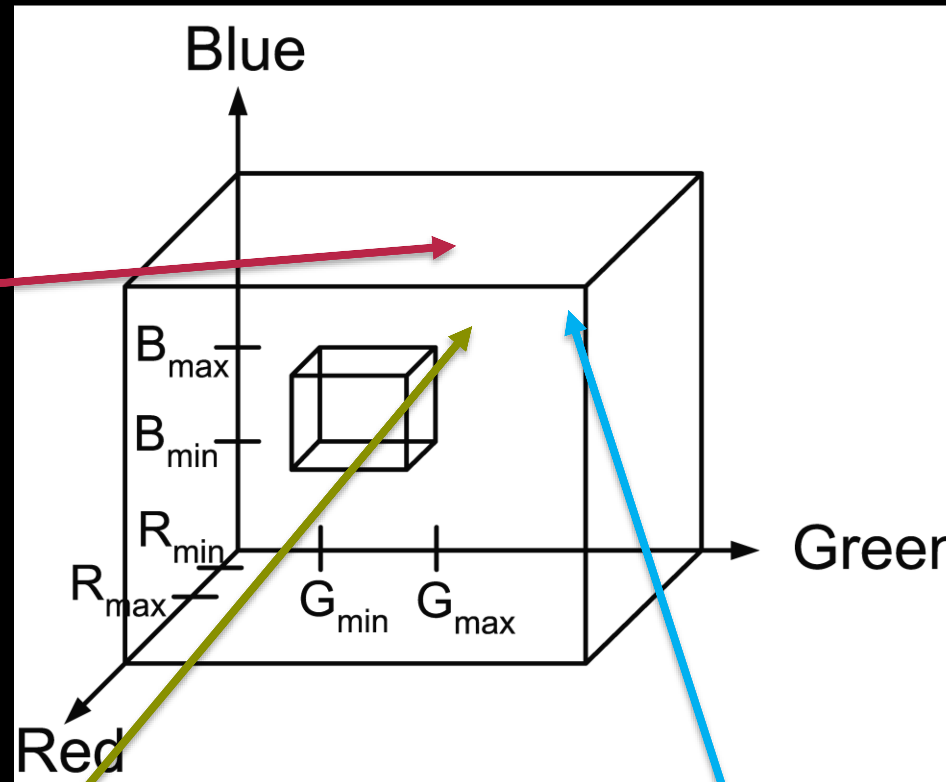
If

$$R > R_{min} \text{ and } R < R_{max} \text{ and}$$
$$G > G_{min} \text{ and } G < G_{max} \text{ and}$$
$$B > B_{min} \text{ and } B < B_{max}$$

Then $g(x, y) = 255$

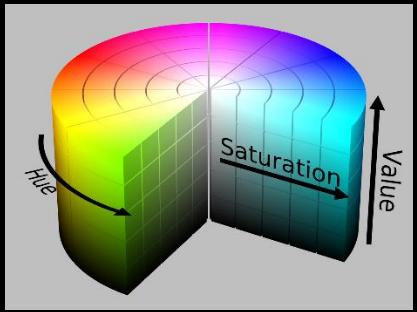
Else $g(x, y) = 0$

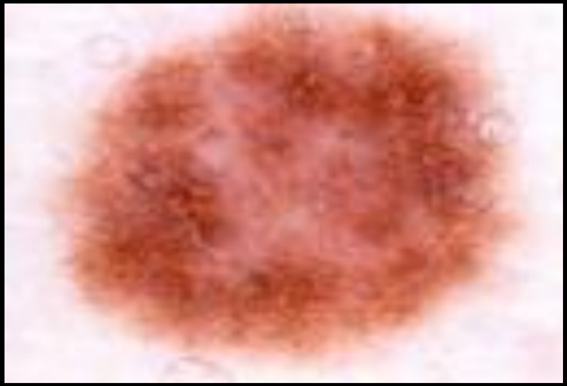
Color thresholding



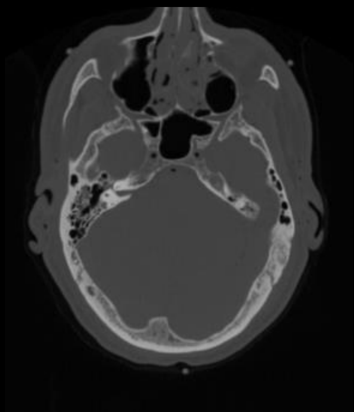
Color variation

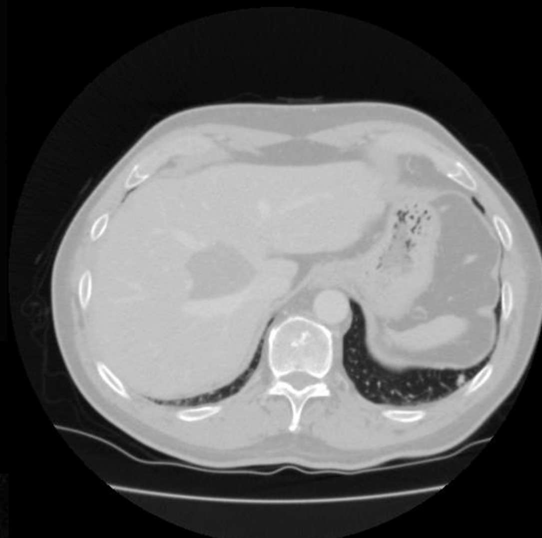
- The major variation is in the brightness
  - This will spread out the values in RGB space
- The Hue is rather constant
- HSI Space
  - HUE and saturation rather stable
  - Only variation in intensity / value





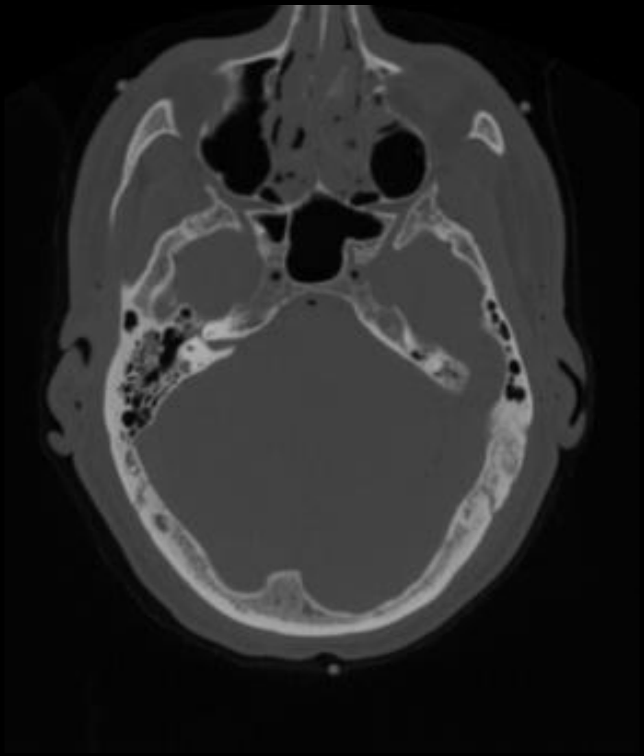
Contrast in medical images





- How do we optimise image acquisition when we want to look at
  - Bones
  - Brain structures
  - Cancer

Image acquisition - bone



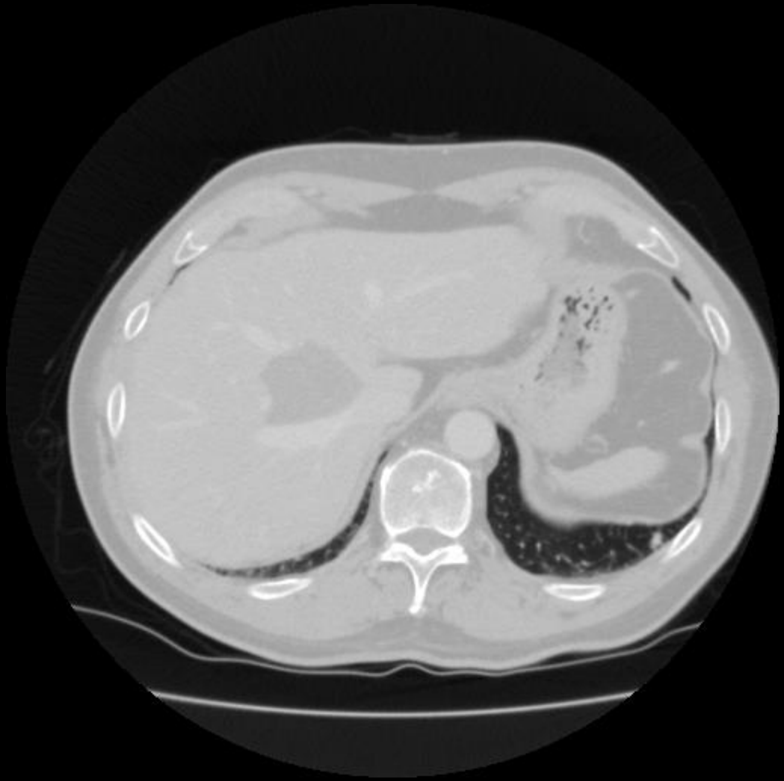
- X-rays
  - goes through soft tissue with little loss
  - are attenuated in bone
- CT scanners use X-rays
  - Good for imaging bones
- A simple threshold can often extract the bones
- Areas with only bone and soft-tissue will have a bimodal histogram

Attenuation - the gradual loss in intensity

Image acquisition – brain structures

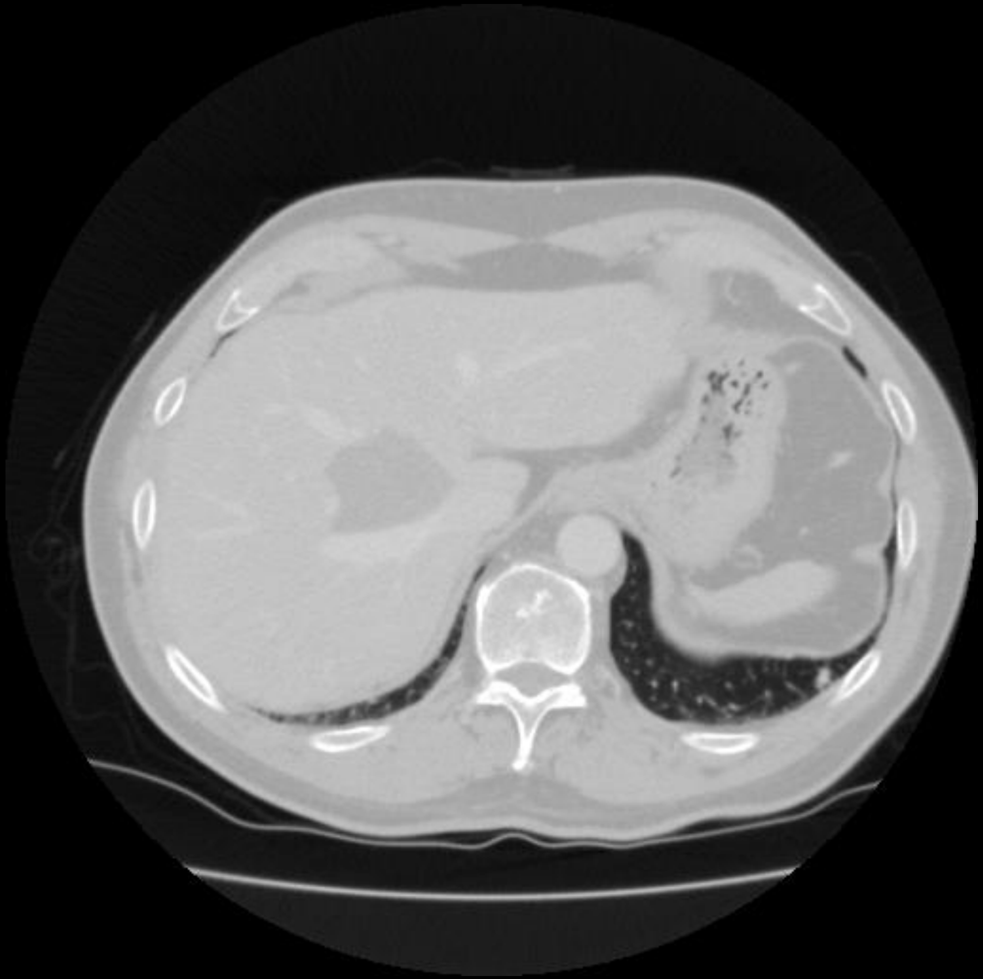
- Magnetic Resonance Imaging (MRI) is often used
- Much more difficult to explain!
  - Based on very powerful magnetic fields and radio waves
- Needs water molecules!
- Bone is black!

Image acquisition - cancer

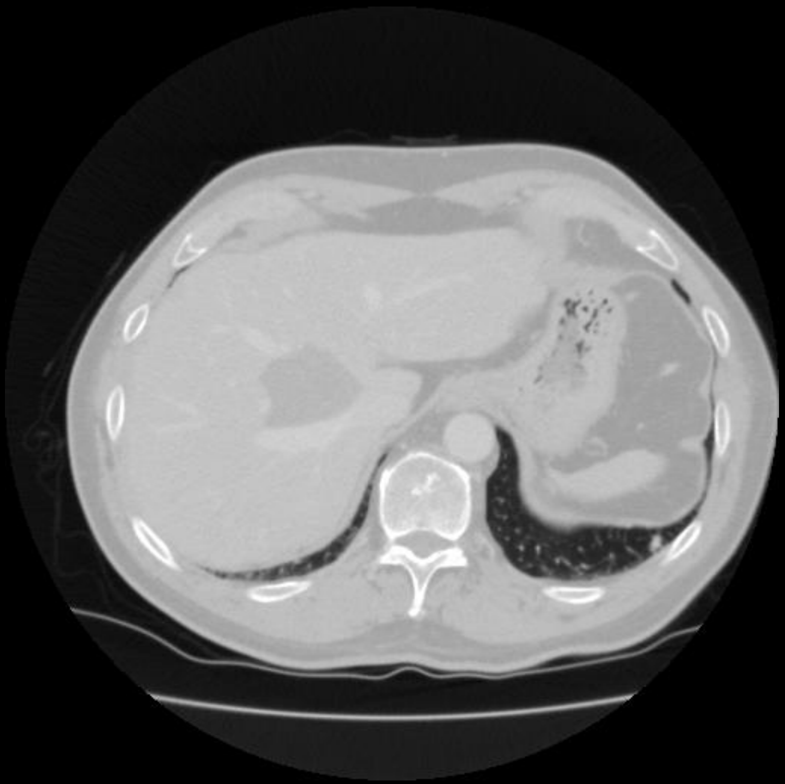


- CT scan
- Liver cancer
  - Very difficult to see

What makes cancer cells special?



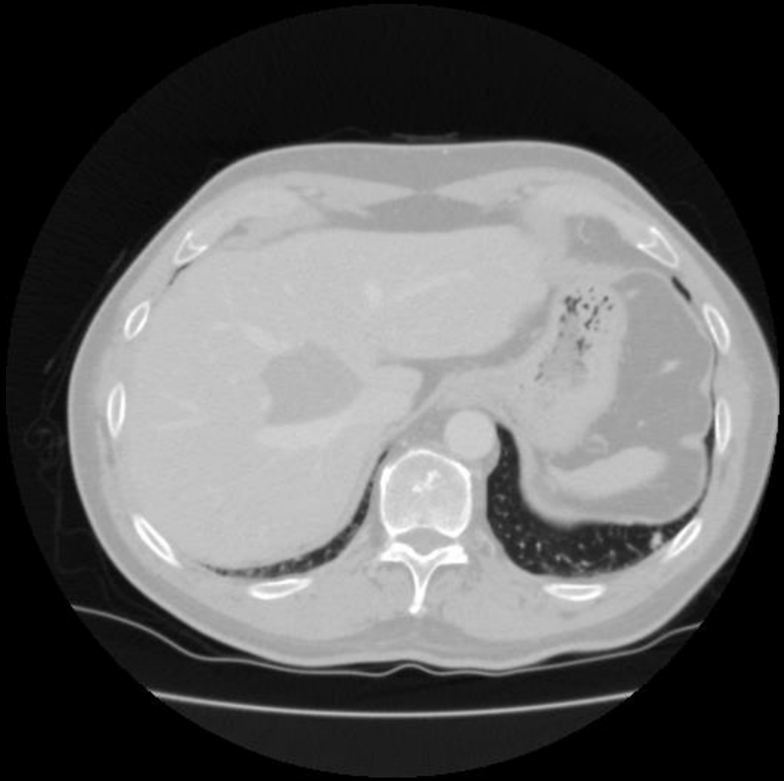
Cancer metabolism



- Cancer cells typically have a high metabolism
  - They eat more!

- Some substances are easier to see on different scanners
  - Bone on CT

Using the cancer metabolism



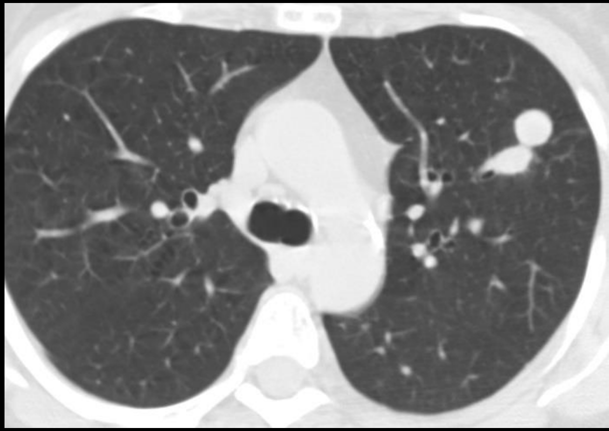
- Something that is to see

+

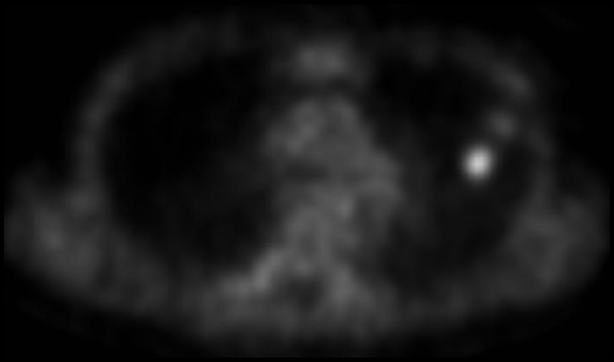
- Something that is being eaten by the cancer

- A tracer

Contrast using tracers



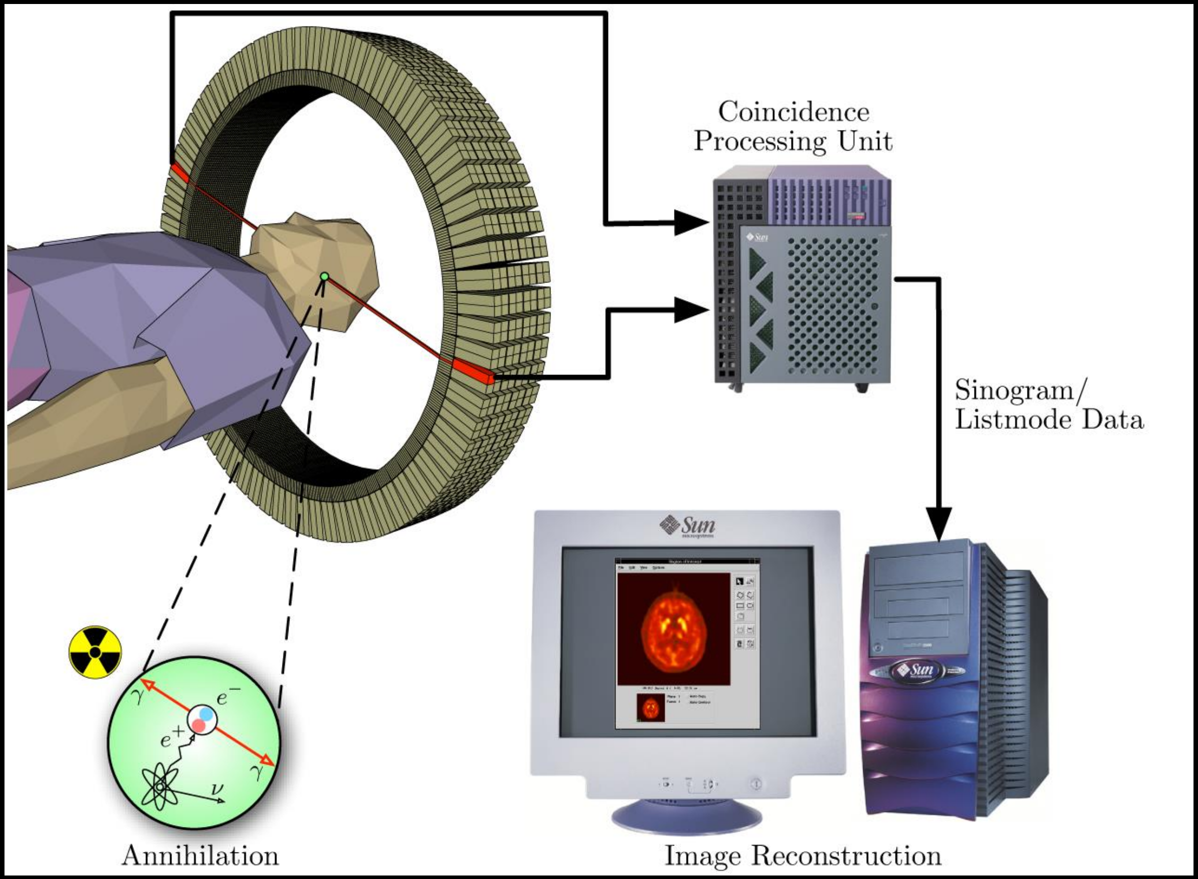
CT image



PET-FDG image

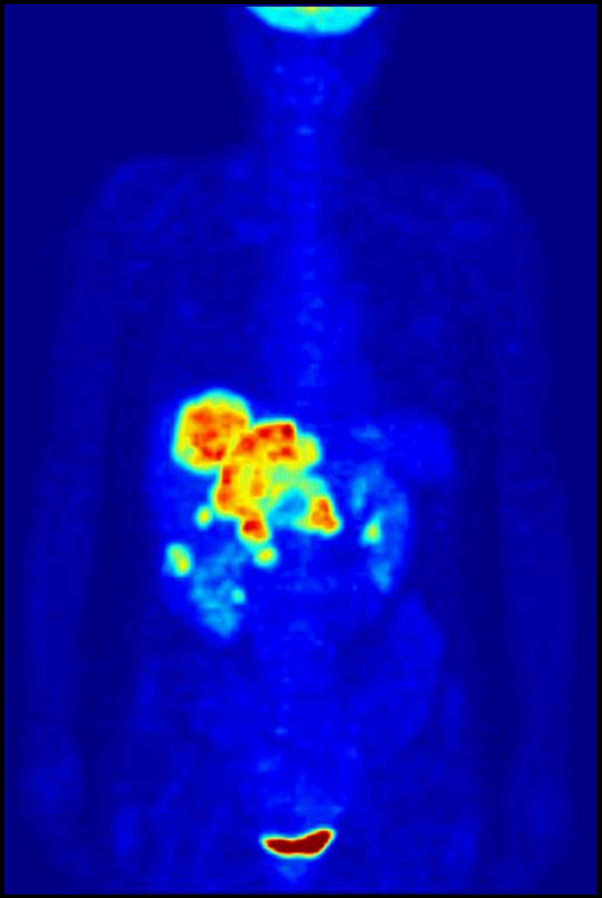
- A commonly used tracer is
  - 18F-FDG = 18F-fluorodeoxyglucose
- Used in *oncological PET*
  - *Oncology : Cancer*
  - *PET: positron emission tomography*
- Positron-emitting radioactive isotope fluorine-18
- Glucose is a “sugar”

PET



Wikipedia

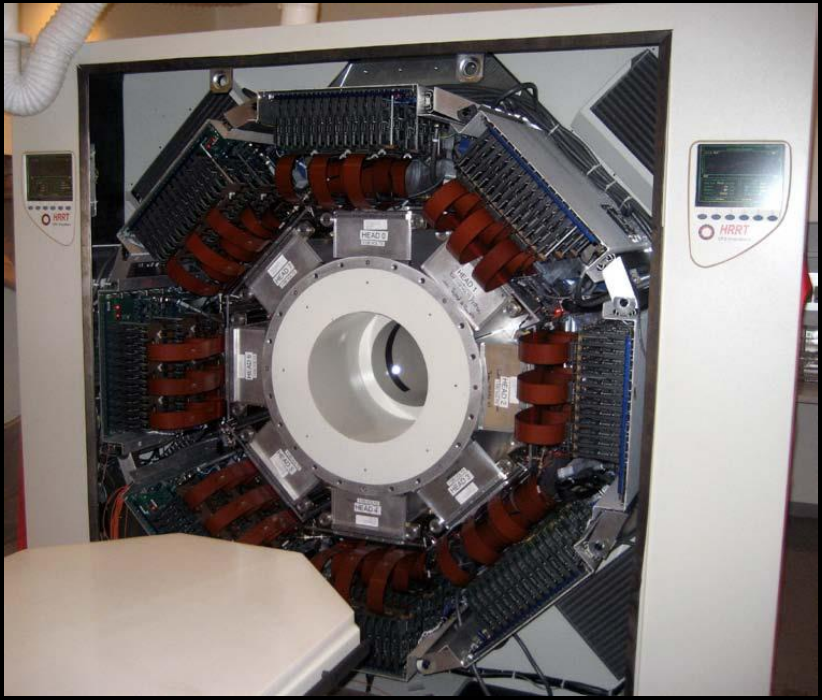
PET Image

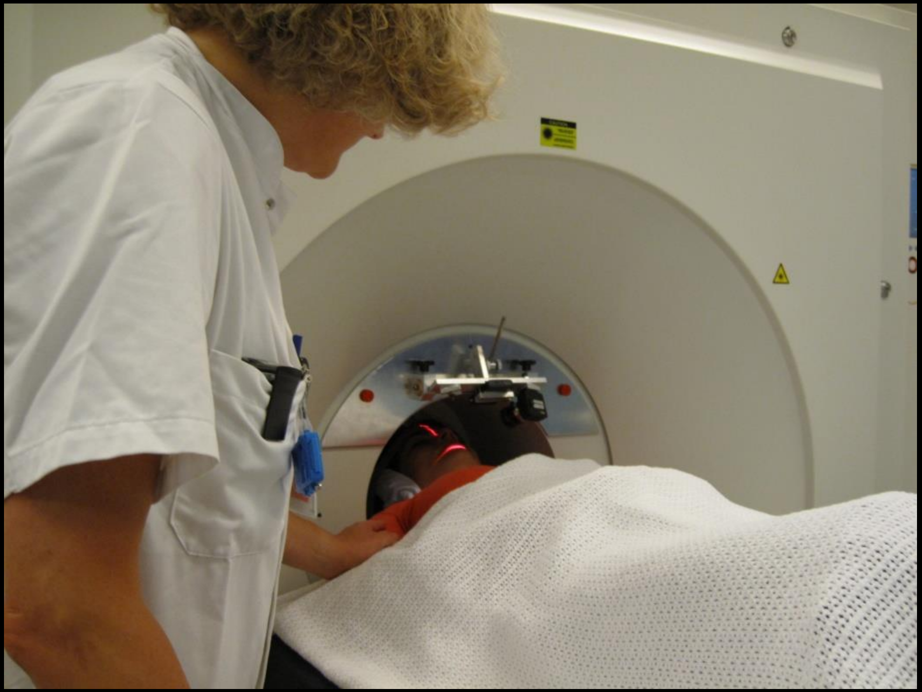


Wikipedia

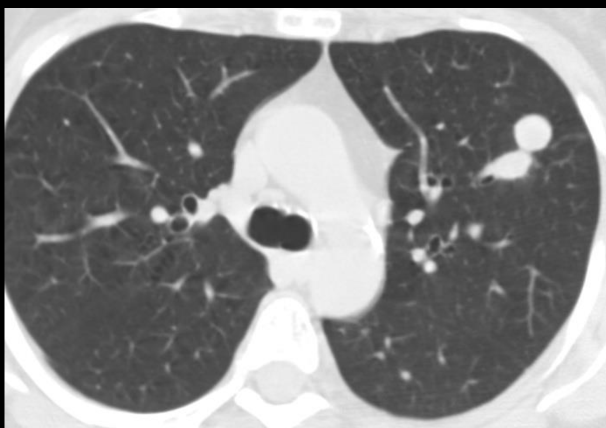
- Areas with high glucose intake will be brighter
  - Higher intake of radioactive molecules
- Bimodal histograms in areas with cancer cells
- Big research topic

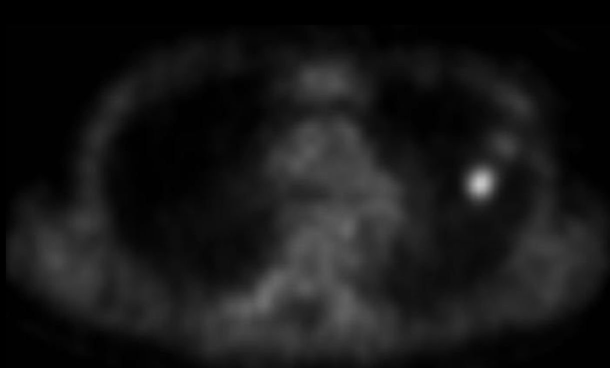
High-Resolution PET scanner at Rigshospitalet





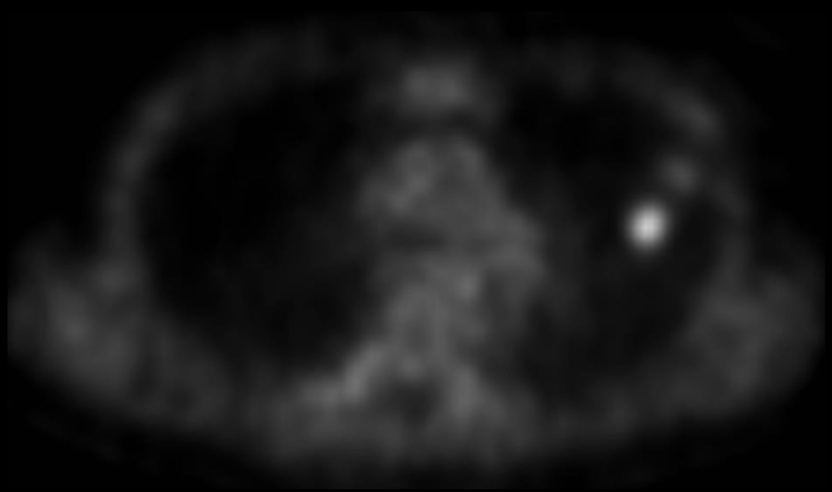
Combining Images





- CT is good for bone
- PET is good for cancer
- What if I want to see both?
- PET/CT scanner
  - Patient scanned in both a CT and a PET scanner
- Image registration
  - Take two or more separate images
  - Combine them using image registration
  - More about that later

Thresholds visited



- The tumour became much more separated from the background
- Perhaps a simple threshold is enough now?
- The best solution
  - Clever imaging techniques

    and
  - Intelligent image analysis

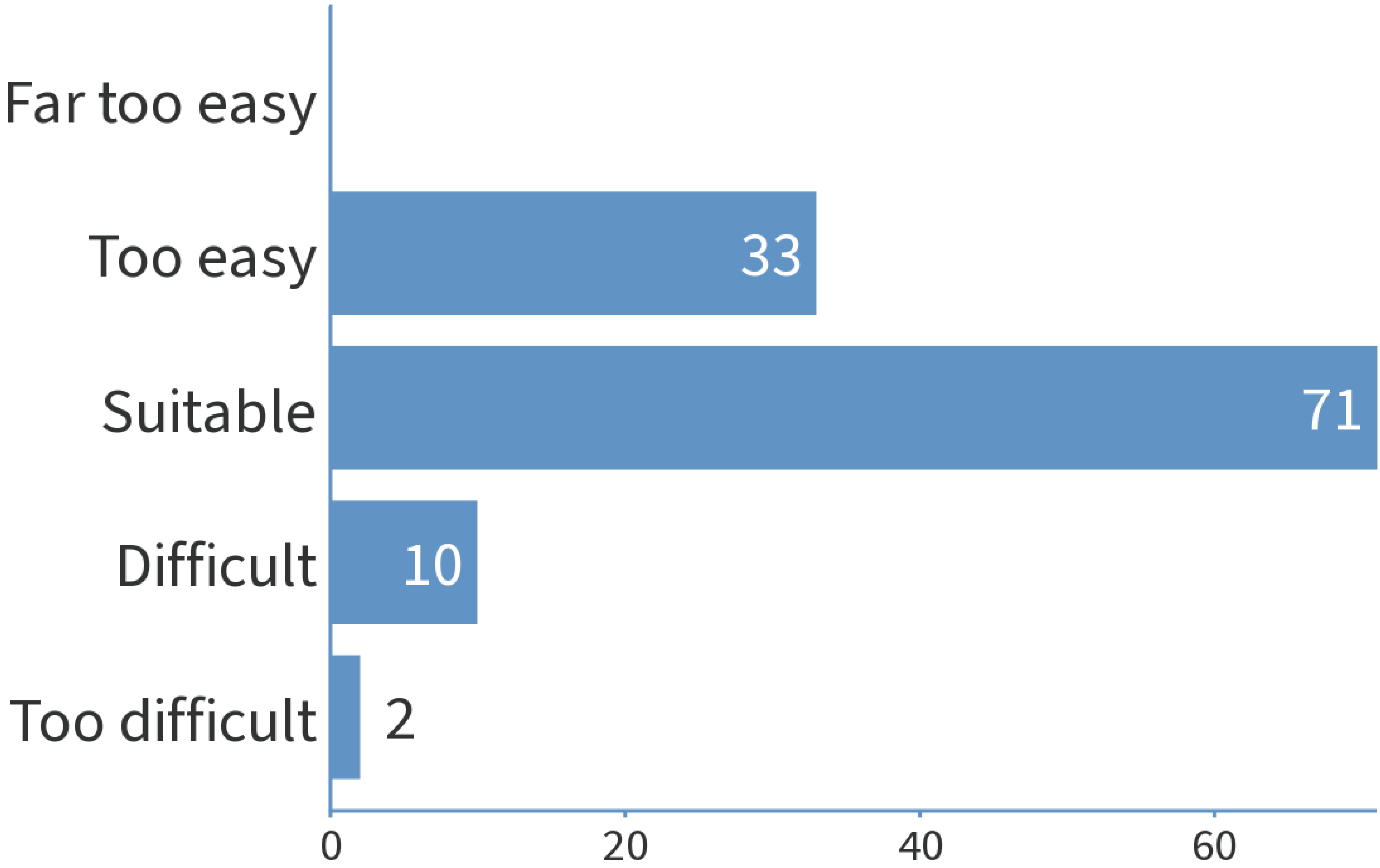
PCA on images

Level of the lectures

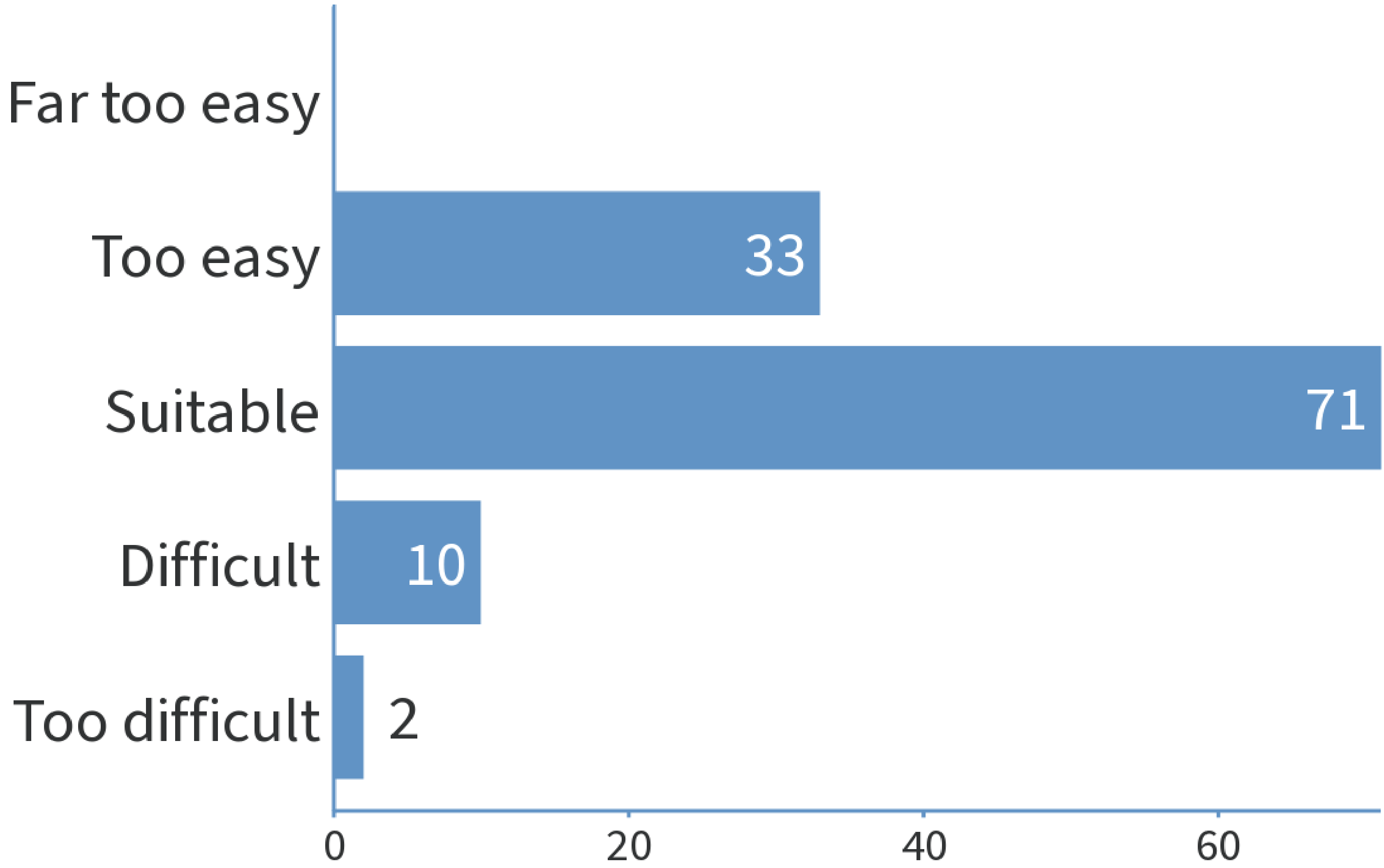
- Far too easy
- Too easy
- Suitable
- Difficult
- Too difficult

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Level of the lectures



Level of the lectures



Level of the exercises

Far too easy

Too easy

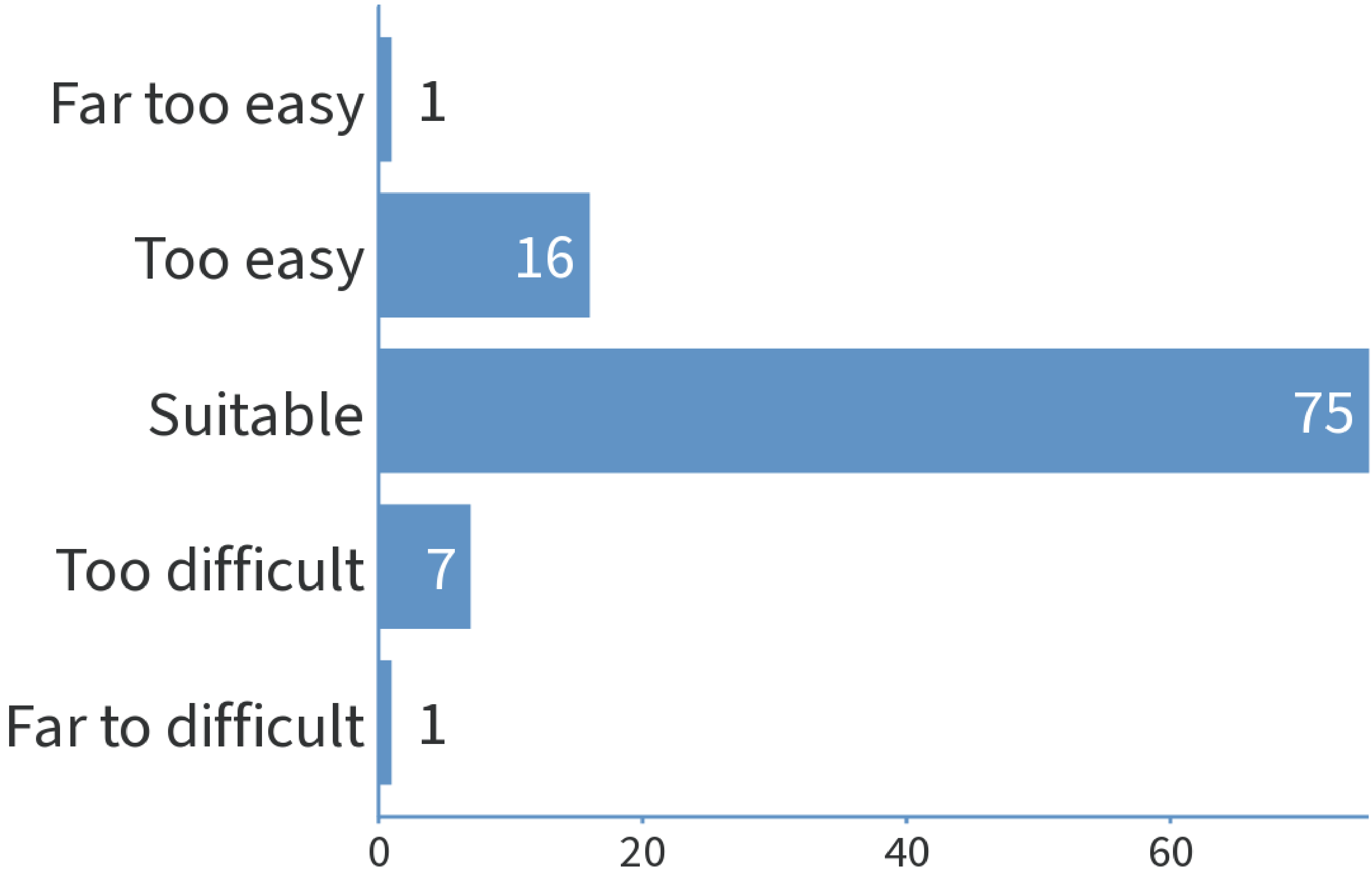
Suitable

Too difficult

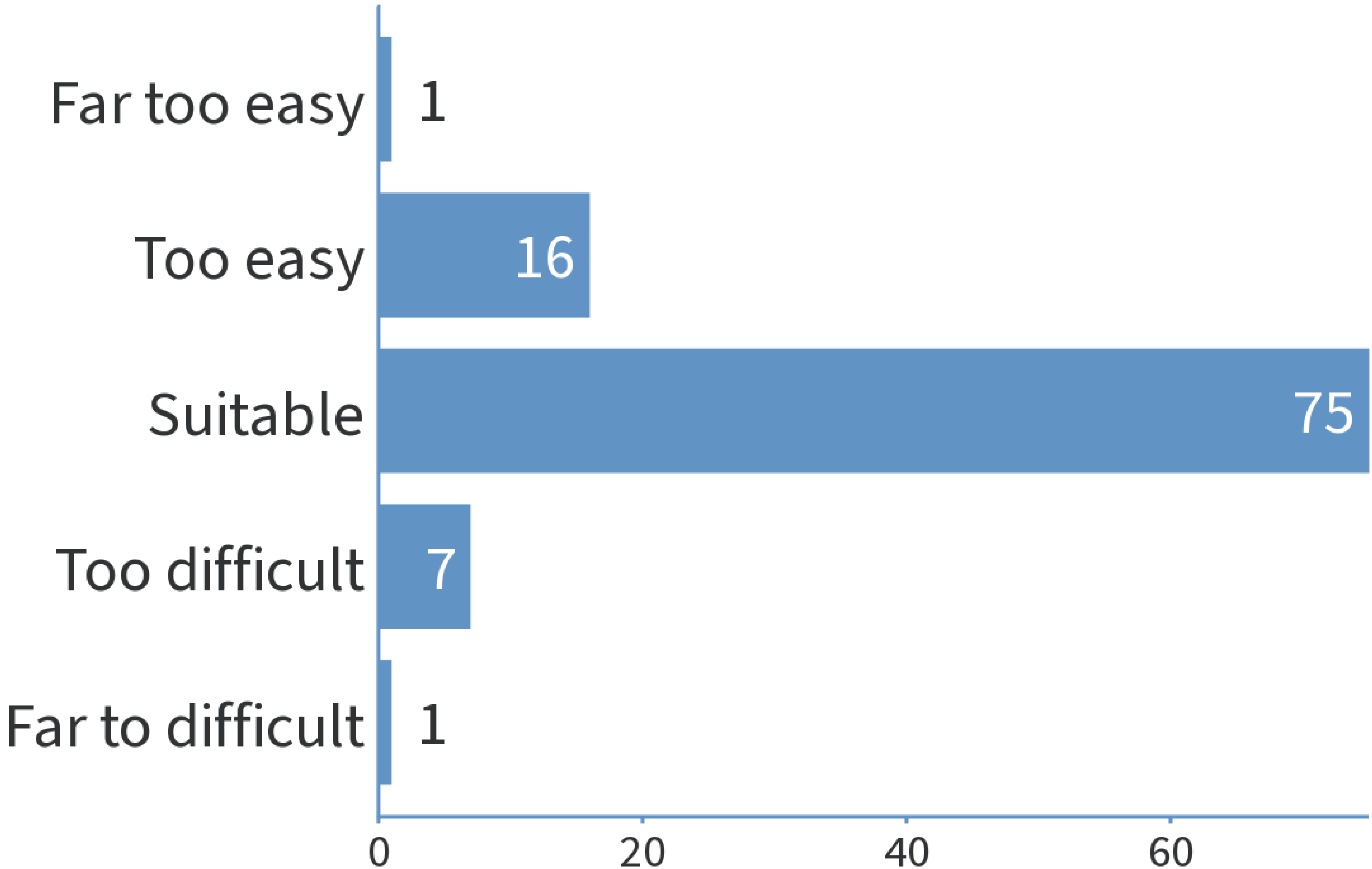
Far to difficult

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Level of the exercises



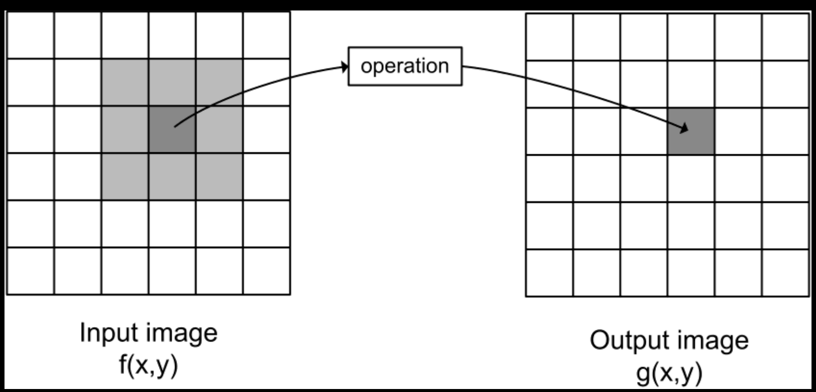
Level of the exercises

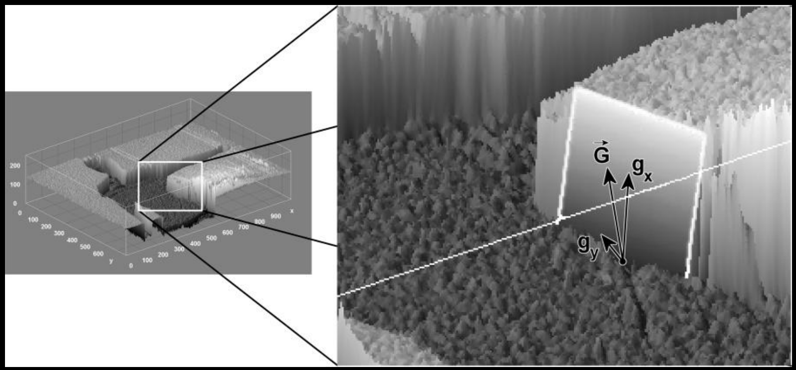


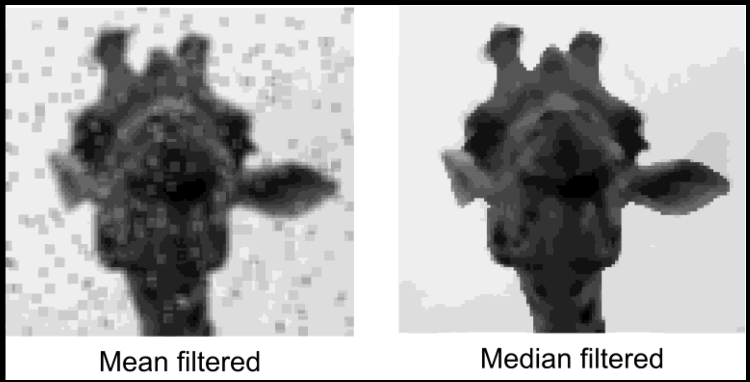
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Next week

- Neighbourhood processing (Filtering)
- Morphology







Principal component analysis on images

Rasmus R. Paulsen

DTU Compute

Based on

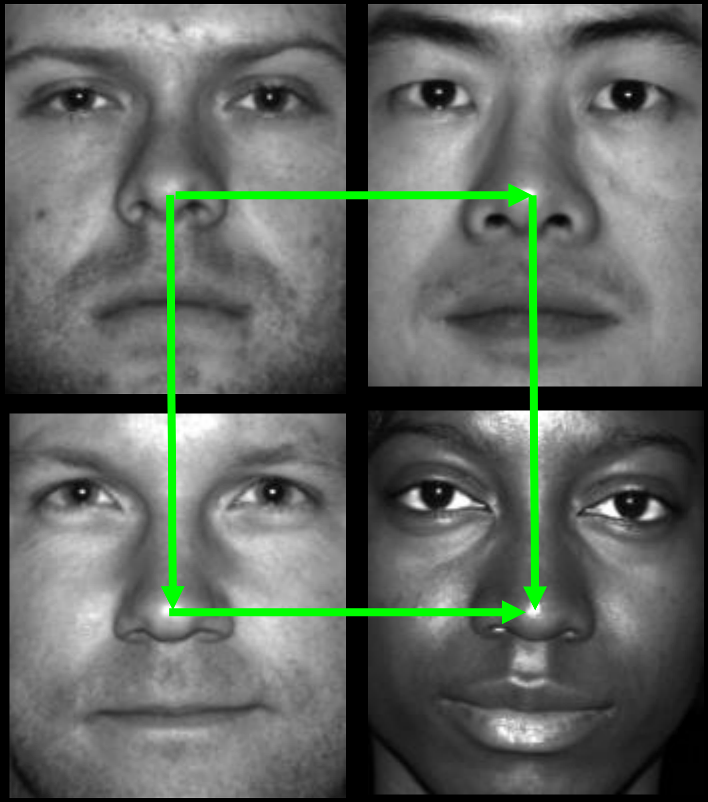
M. Turk and A. Pentland. *Face recognition using eigenfaces*. Computer Vision and Pattern Recognition, 1991.

http://compute.dtu.dk/courses/02502

Principal Component Analysis on images learning objectives

- Construct a column matrix from a single gray scale image
- Construct a data matrix from a set of gray scale images
- Compute and visualize an average image from a set of images
- Compute the principal components of a set of images
- Visualize the principal components computed from a set of images
- Synthesize an image by combining the average image and a linear combination of principal components

Face data



- 38 face images
  - 168 x 192 grayscale
- Aligned
  - The anatomy is placed ”in the same position in all image”
- Same illumination conditions on the images we use

The Extended Yale Face Database B

http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

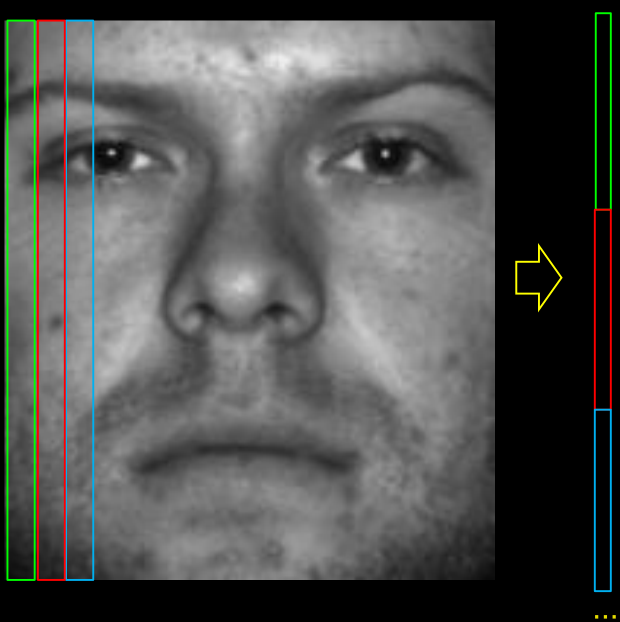
Principal component analysis on face images




- What is the main variation in face images?
  - The variation of appearance
  - Not the position in the image
  - Not the light conditions
  - Not the direction of the head

Putting images into matrices

- An image can be made into a column matrix
  - Stack all image columns into one column



$$I = \begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_m \end{bmatrix}$$

Face images in matrix form

- One column is one face
- n=38 faces
- m=168x192 = 32256 pixel values per image

$$X = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{bmatrix}$$

The average face

$$X=\begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{bmatrix}$$

- The average face
  - Average of each row
  - One column
  - Put it back into image shape
- Blurry around the eyes
  - Not perfectly aligned

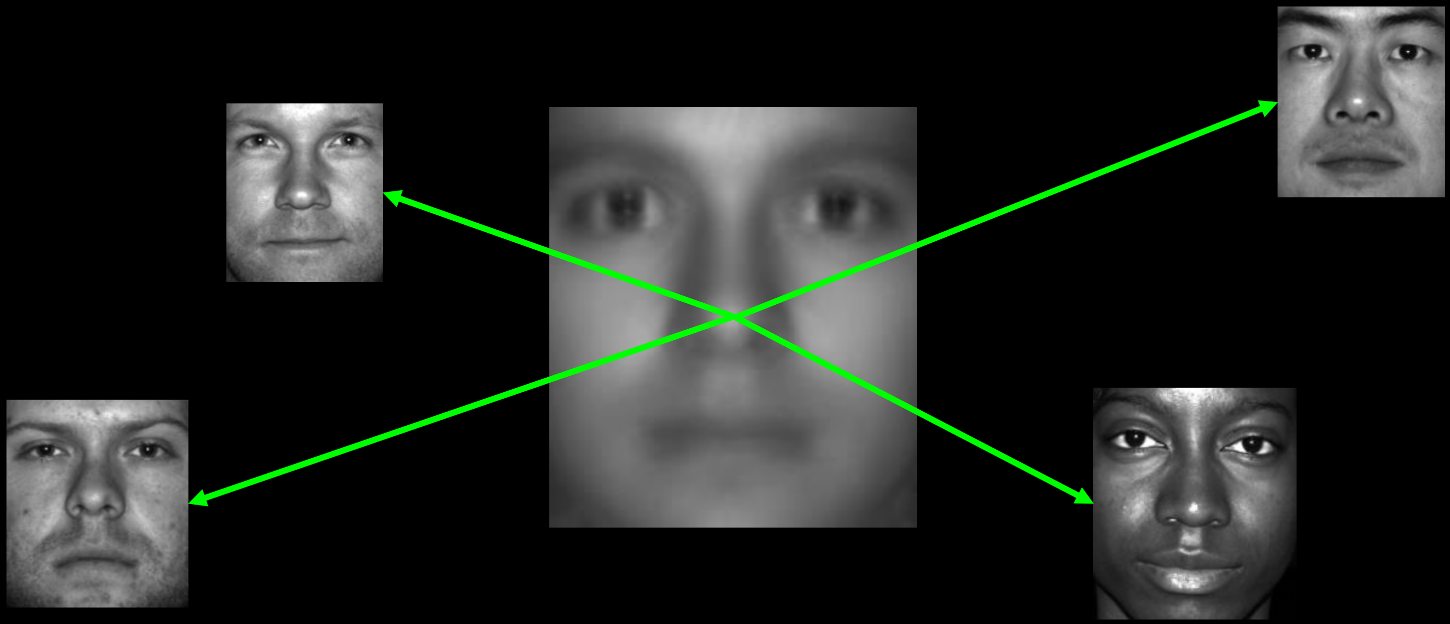
Subtracting the mean face

$$X' = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{bmatrix} - \bar{X}$$

- We subtract the mean face from all faces

Analyzing the deviation from the mean face

- We want to do the principal component analysis on the *deviations from the average face*

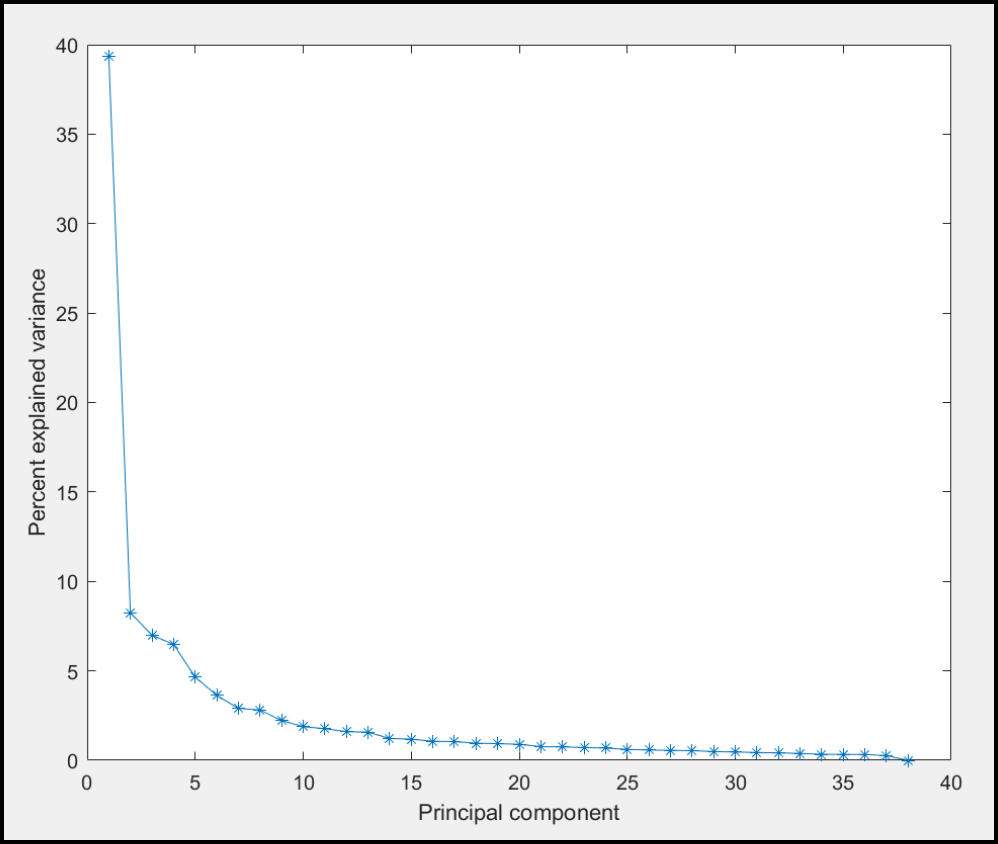


PCA Analysis on face data

$$X' = \begin{bmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{bmatrix} - \bar{X}$$

- We do the PCA analysis on the X' matrix
- X' is 32256 x 38
- Standard covariance matrix is 32256 x 32256
- Turk and Pentland found a trick:
  - Compute the PCA on the 38 x 38 matrix instead of the 32256x32256 matrix
  - Details in the paper
    - Beyond the scope here

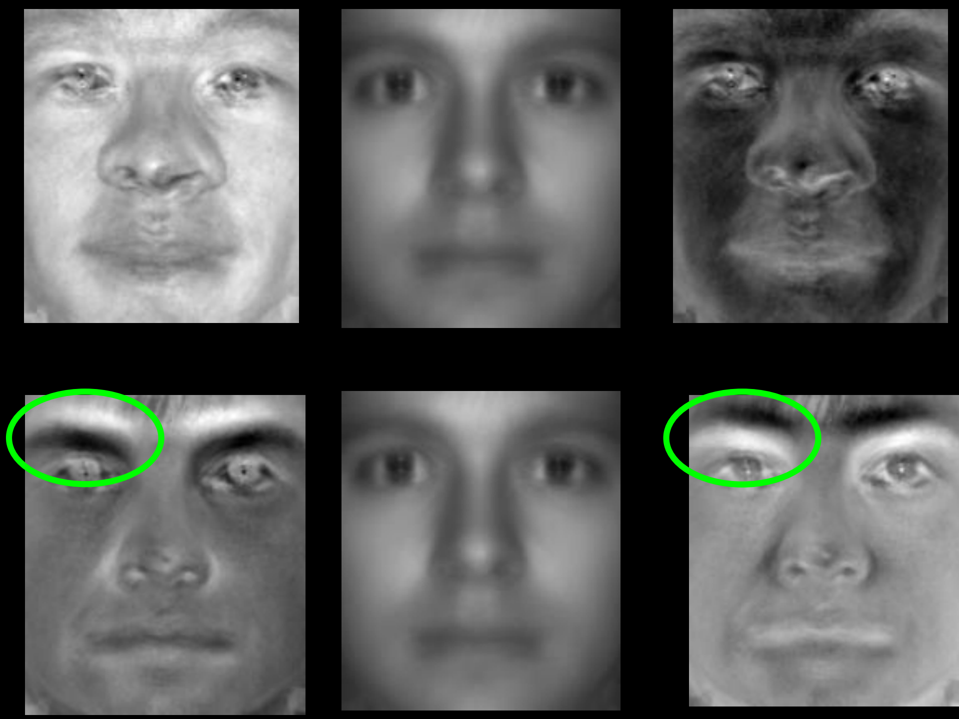
PCA on faces



- First eigenvector explains 40% of variation
- Second eigenvector explains 8% of variation

Visualizing the PCA faces

Main deviations from the average face



First PC – 40% of variation

Second PC – 8% of variation

A tool to see major variations – brow lifting

-PC

Average face

+PC

Synthesizing faces

- A new face can be created by combining
  - Average face
  - Linear combination of principal components

+ 0.05

- 0.12

=

Average face

PC1

PC2

Decomposing faces

- A given face can be reconstructed using
  - The average face
  - Linear combination of principal components
- Found by projecting the face on the principal components
- The weights can then be used for classification/identification

$+w_1$

$+w_2$

$\cong$

Average face

PC1

PC2

Face analysis plus plus?

- More examples later in the course

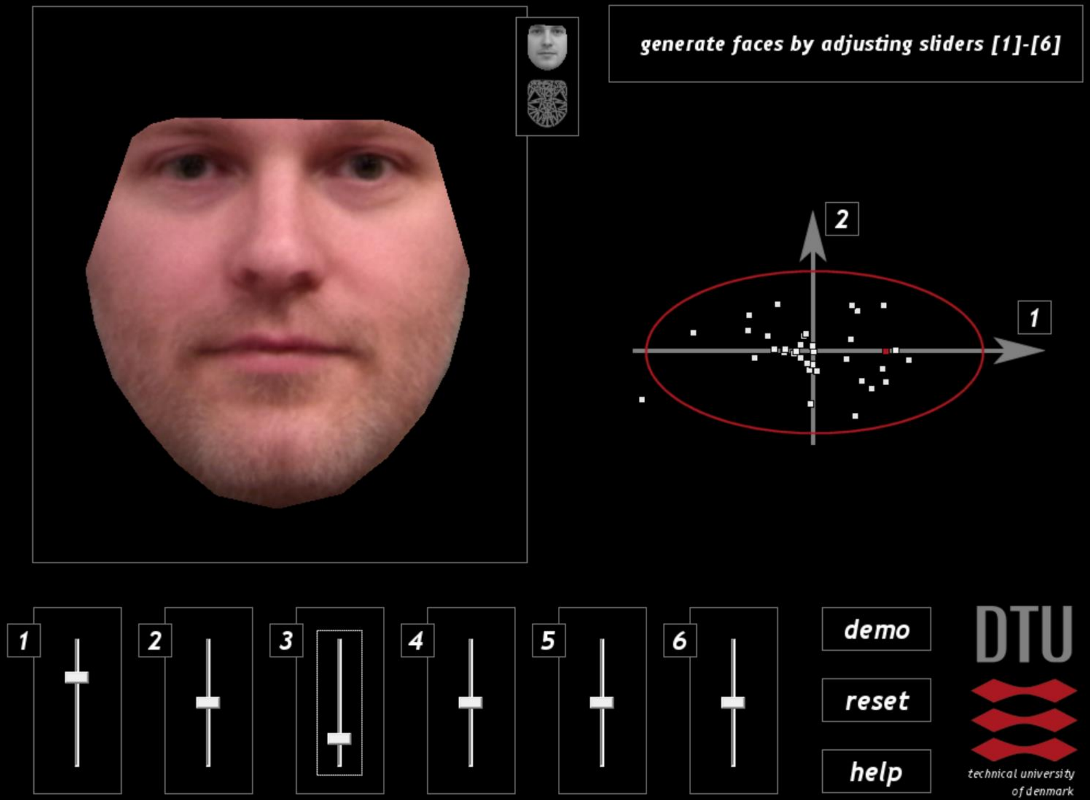


Image Analysis

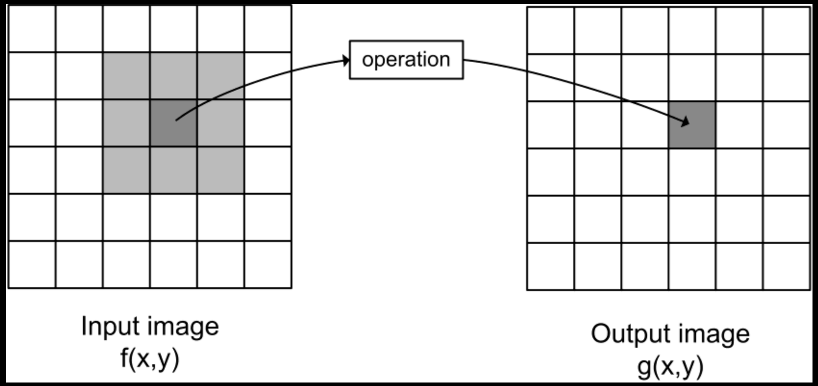
Rasmus R. Paulsen
Tim B. Dyrby
DTU Compute

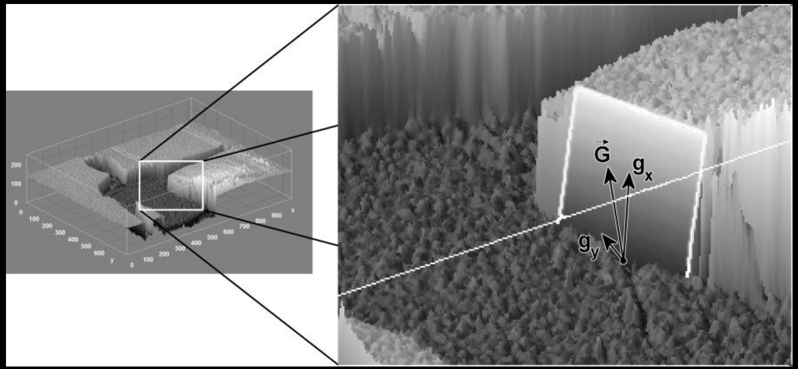
rapa@dtu.dk

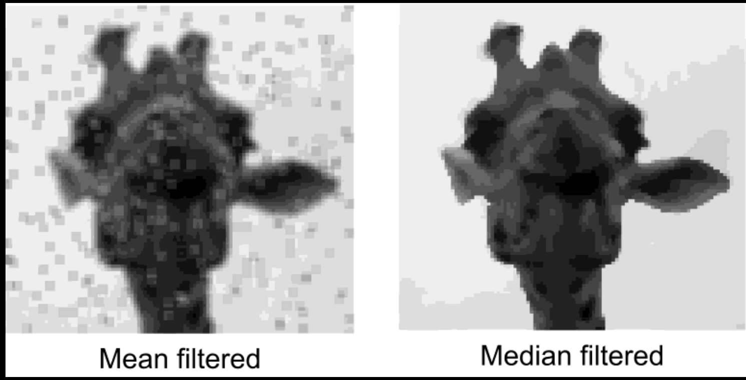
http://www.compute.dtu.dk/courses/02502

Plenty of slides adapted from Thomas Moeslunds lectures

Lecture 4
Neighbourhood Processing







What can you do after today?

- Describe the difference between point processing and neighbourhood processing
- Compute a rank filtered image using the min, max, median, and difference rank filters
- Compute a mean filtered image
- Decide if median or average filtering should be used for noise removal
- Choose the appropriate image border handling based on a given input image
- Implement and apply template matching
- Compute the normalised cross correlation and explain why it should be used
- Apply given image filter kernels to images
- Use edge filters on images
- Describe finite difference approximation of image gradients including the magnitude and the direction
- Compute the magnitude of the gradient
- Describe the concept of edge detection

Point processing

Input

1	2	0	1	3	
2	1	4	2	2	
1	0	1	0	1	
1	2	1	0	2	
2	5	3	1	2	

Output

	$\frac{12}{9}$				

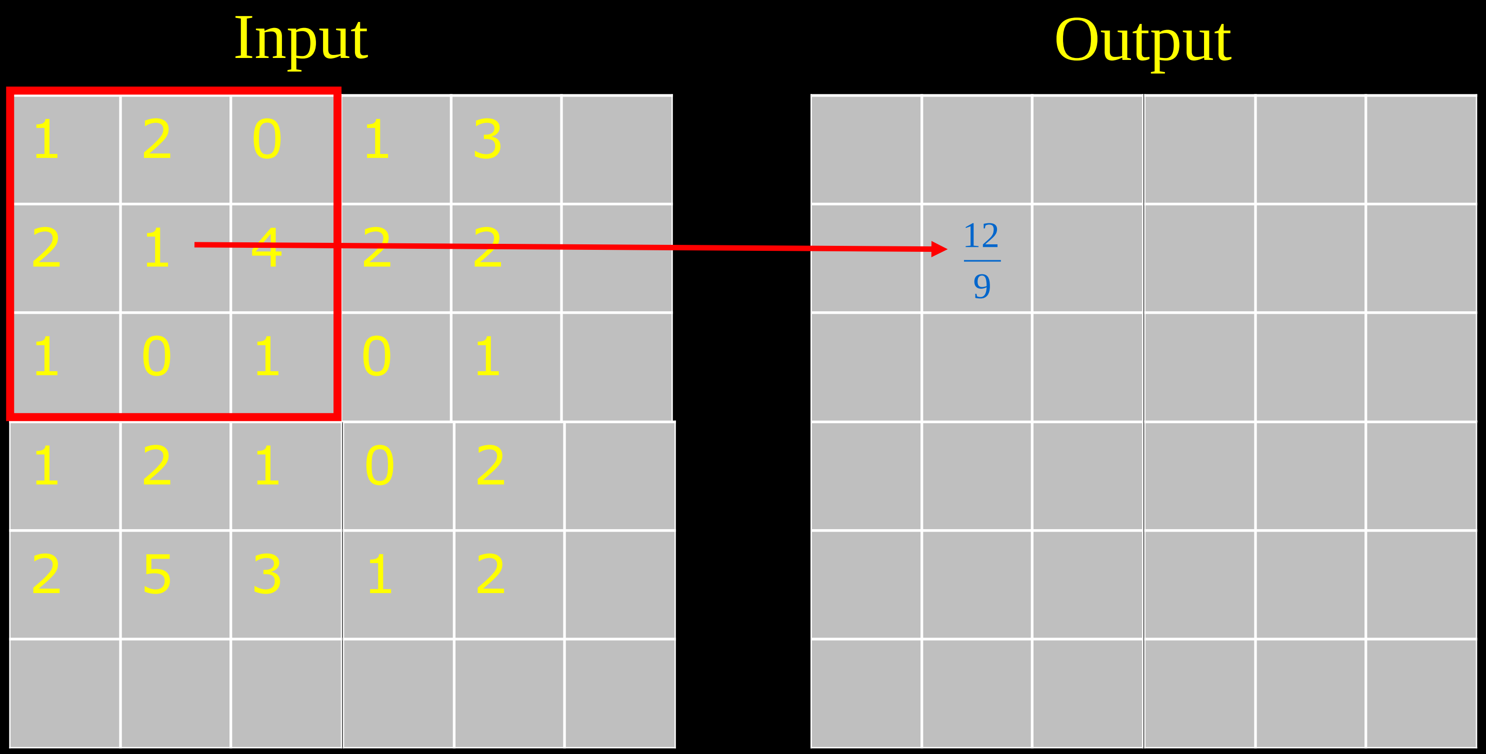
- The value of the output pixel is only dependent on the value of one input pixel
- A global operation – changes all pixels

Point processing

- Grey level enhancement
  - Process one pixel at a time independent of all other pixels
  - For example used to correct Brightness and Contrast

Correct | Too high brightness | Too low brightness | Too high contrast | Too low contrast

Neighbourhood processing



- Several pixels in the input has an effect on the output

Use of filtering

Noise removal

Enhance edges

Smoothing

- Image processing
- Typically done before actual image analysis

Salt and pepper noise

- Pixel values that are very different from their neighbours
- Very bright or very dark spots
- Scratches in X-rays

What is that?

Salt and pepper noise

- Fake example
  - Let us take a closer look at noise pixels

169	169	173	170	170	172	171	171	169
172	173	172	172	169	171	168	171	170
168	171	169	168	0	169	170	169	255
173	175	170	172	173	168	170	169	171
169	175	170	172	170	255	169	255	169
173	172	255	171	170	172	169	169	170
176	175	172	173	172	171	169	168	173
173	172	169	168	166	0	170	165	166
170	172	172	170	169	169	169	168	172
174	172	172	166	167	168	168	170	172

They are all 0 or 255

Should we just remove all the 0's and 255's from the image?

What is so special about noise?

169	169	173	170	170	172	171	171	169
172	173	172	172	169	171	168	171	170
168	171	169	168	0	169	170	169	255
173	175	170	172	173	168	170	169	171
169	175	170	172	170	255	169	255	169
173	172	255	171	170	172	169	169	170
176	175	172	173	172	171	169	168	173
173	172	169	168	166	0	170	165	166
170	172	172	170	169	169	169	168	172
174	172	172	166	167	168	168	170	172

172, 169, 171, 168, 0, 169, 172, 173, 168

- What is the value of the pixel compared to the neighbours?
- Average of the neighbours
  - 170
- Can we compare to the average?
  - Difficult – should we remove all values bigger than average+1 ?
- It is difficult to detect noise!

Noise – go away!

169	169	173	170	170	172	171	171	169
172	173	172	172	169	171	168	171	170
168	171	169	168	0	169	170	169	255
173	175	170	172	173	168	170	169	171
169	175	170	172	170	255	169	255	169
173	172	255	171	170	172	169	169	170
176	175	172	173	172	171	169	168	173
173	172	169	168	166	0	170	165	166
170	172	172	170	169	169	169	168	172
174	172	172	166	167	168	168	170	172

170

149

172, 169, 171, 168, 0, 169, 172, 173, 168

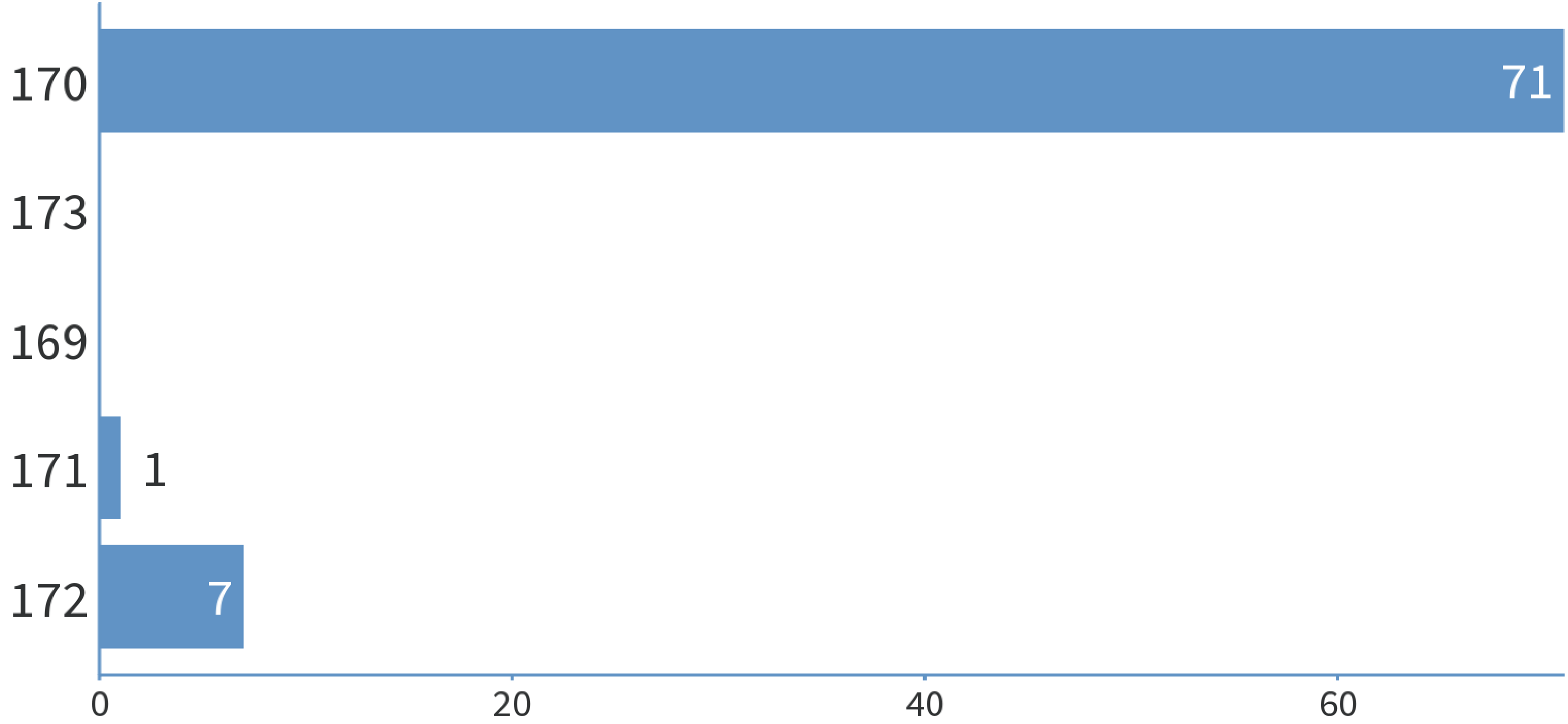
169, 168, 0, 170, 172,173, 170, 172, 170

- We can not tell what pixels are noise
- One solution
  - Set all pixels to the average of the neighbours (and the pixel itself)
- Oh no!
  - Problems!
  - The noise "pollutes" the good pixels

What is the median value of [169, 168, 0, 170, 172, 173, 170, 172, 170]?

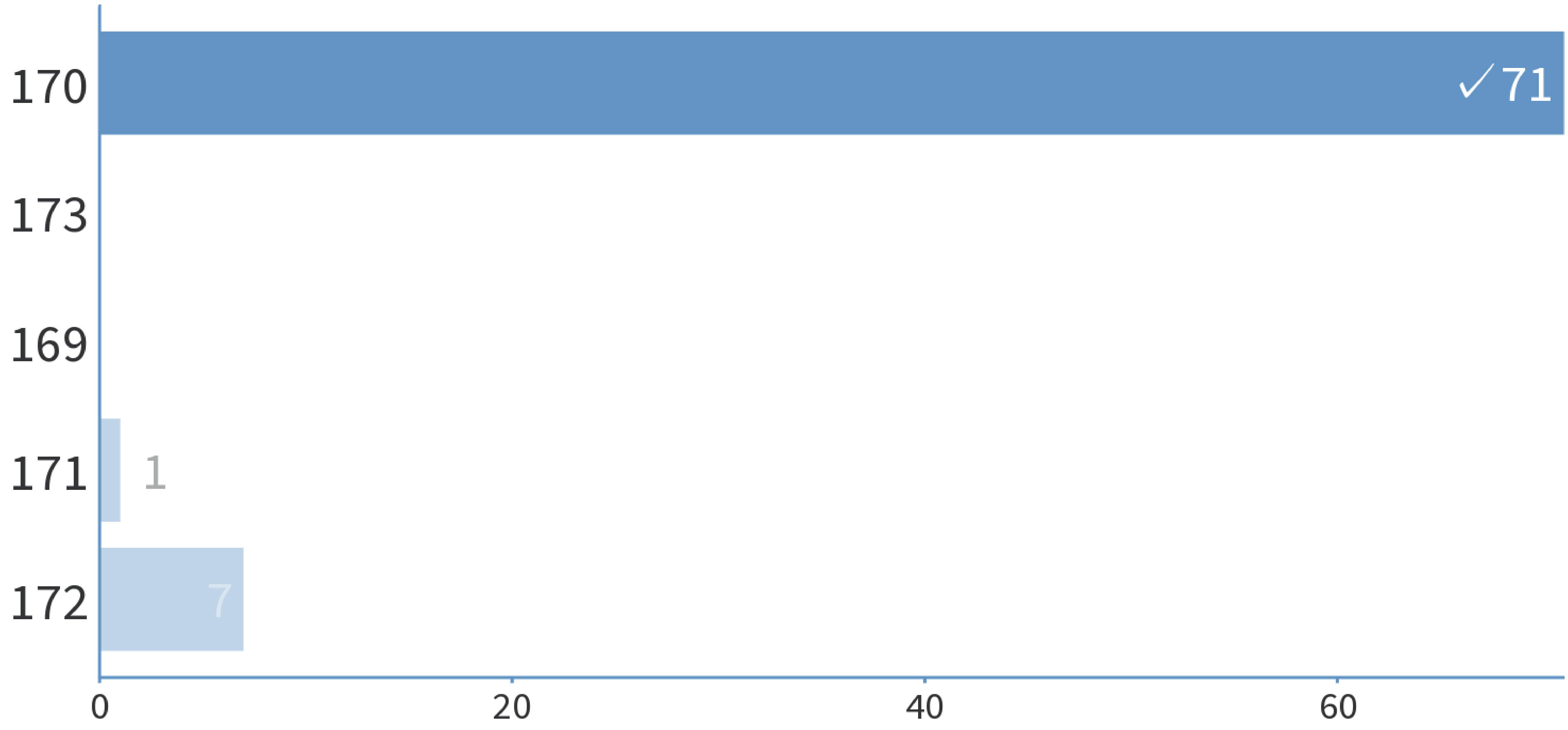
- 170
- 173
- 169
- 171
- 172

What is the median value of [169, 168, 0, 170, 172, 173, 170, 172, 170]?



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

What is the median value of [169, 168, 0, 170, 172, 173, 170, 172, 170]?



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

The median value

- The values are sorted from low to high
- The middle number is picked
  - The median value

169, 168, 0, 170,172, 173, 170, 172, 170

0,168, 169, 170, 170, 170,172, 172,173

Median

Noise has no influence on the median!

Noise away – the median filter

169	169	173	170	170	172	171	171	169
172	173	172	172	169	171	168	171	170
168	171	169	168	0	169	170	169	255
173	175	170	172	173	168	170	169	171
169	175	170	172	170	255	169	255	169
173	172	255	171	170	172	169	169	170
176	175	172	173	172	171	169	168	173
173	172	169	168	166	0	170	165	166
170	172	172	170	169	169	169	168	172
174	172	172	166	167	168	168	170	172

170

170

172, 169, 171, 168, 0, 169, 172, 173, 168

169, 168, 0, 170, 172,173, 170, 172, 170

- All pixels are set to the median of its neighbourhood (including the pixel itself)
- Noise pixels do not pollute good pixels

Noise removal – average filter

Scanned X-ray with salt and pepper noise

Average filter (3x3)

Noise removal – median filter

Scanned X-ray with salt and pepper noise

Median filter (3x3)

Image Filtering

169	169	173	170	170	172	171	171	169
172	173	172	172	169	171	168	171	170
168	171	169	168	0	169	170	169	255
173	175	170	172	173	168	170	169	171
169	175	170	172	170	255	169	255	169
173	172	255	171	170	172	169	169	170
176	175	172	173	172	171	169	168	173
173	172	169	168	166	0	170	165	166
170	172	172	170	169	169	169	168	172
174	172	172	166	167	168	168	170	172

- Creates a new *filtered* image
- Output pixel is computed based on a neighbourhood in the input image
- 3 x 3 neighbourhood
  - Filter size 3 x 3
  - Kernel size 3 x 3
  - Mask size 3 x 3
- Larger filters often used
  - Size
    - 7 x 7
  - Number of elements
    - 49

Median filter on image

The image is filtered with a 3 x 3 median filter. What is the result in the pixel marked with a circle?

66	222	102	230	199	147	166	175
204	148	19	241	99	15	187	47
110	140	61	125	62	60	165	94
232	37	31	125	103	90	115	160
46	218	47	86	25	209	139	199
67	159	61	230	34	4	76	21
37	89	106	94	240	11	190	237
35	131	13	28	244	43	48	198

25

90

198

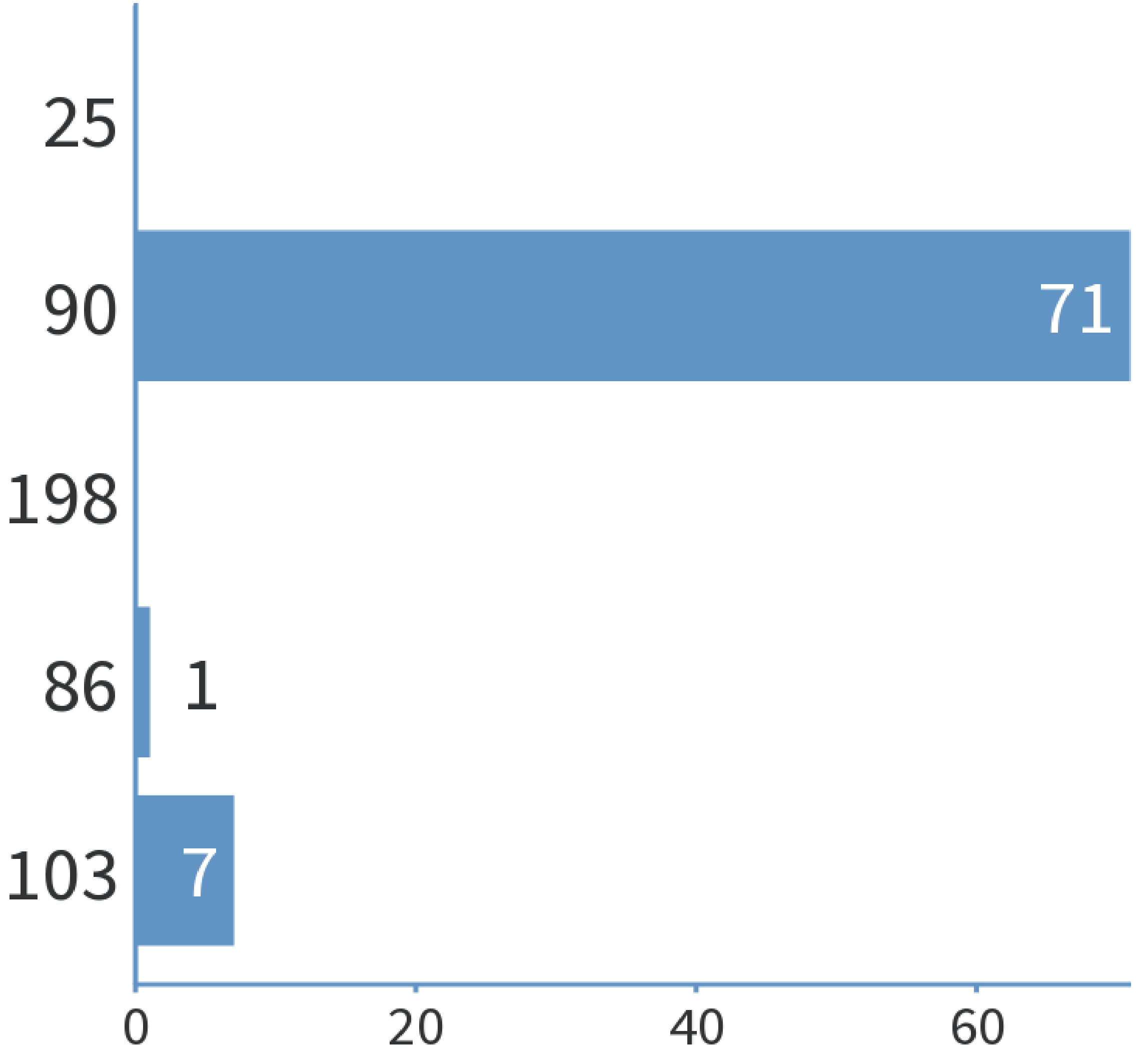
86

103

Median filter on image

The image is filtered with a 3 x 3 median filter. What is the result in the pixel marked with a circle?

66	222	102	230	199	147	166	175
204	148	19	241	99	15	187	47
110	140	61	125	62	60	165	94
232	37	31	125	103	90	115	160
46	218	47	86	25	209	139	199
67	159	61	230	34	4	76	21
37	89	106	94	240	11	190	237
35	131	13	28	244	43	48	198

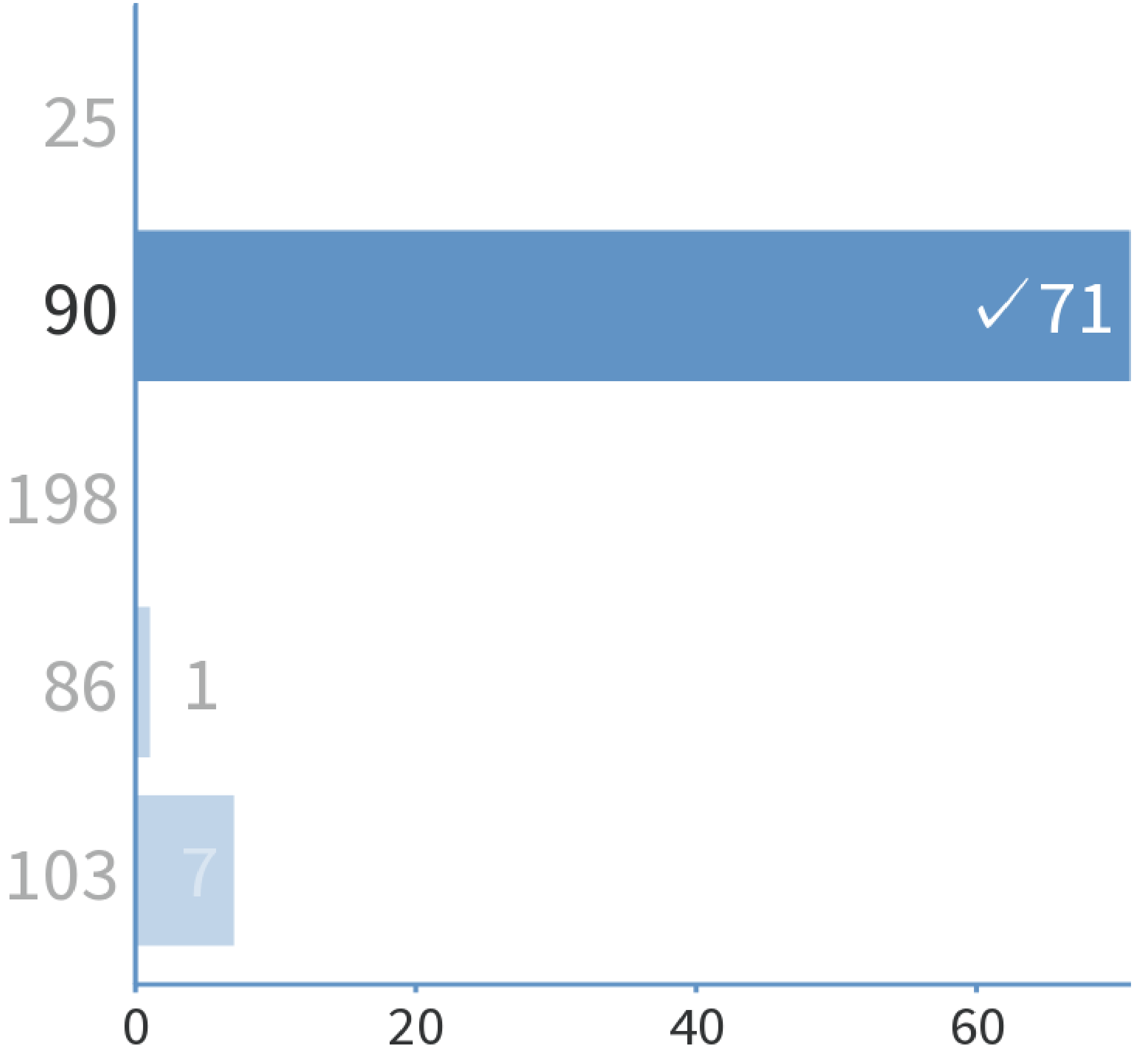


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Median filter on image

The image is filtered with a 3 x 3 median filter. What is the result in the pixel marked with a circle?

66	222	102	230	199	147	166	175
204	148	19	241	99	15	187	47
110	140	61	125	62	60	165	94
232	37	31	125	103	90	115	160
46	218	47	86	25	209	139	199
67	159	61	230	34	4	76	21
37	89	106	94	240	11	190	237
35	131	13	28	244	43	48	198



Rank filters

169	169	173	170	170	172	171	171	169
172	173	172	172	169	171	168	171	170
168	171	169	168	0	169	170	169	255
173	175	170	172	173	168	170	169	171
169	175	170	172	170	255	169	255	169
173	172	255	171	170	172	169	169	170
176	175	172	173	172	171	169	168	173
173	172	169	168	166	0	170	165	166
170	172	172	170	169	169	169	168	172
174	172	172	166	167	168	168	170	172

0 168, 169, 170, 170, 170,172, 172, 173

\-

- Based on sorting the pixel values in the neighbouring region
- Minimum rank filter
  - Darker image. Noise problems.
- Maximum rank filter
  - Lighter image. Noise problems.
- Difference filter
  - Enhances changes (edges)

Rank filters on image

The image is filtered with a 3 x 3 median filter (medI). The image (the original) is also filtered with a 5 x 5 minimum rank filter (minI). The final image is made by subtracting minI from medI. What is the result in the marked pixel?

67	159	61	230	34	4	76	21
37	89	106	94	240	11	190	237
35	131	13	28	244	43	48	198
222	102	230	199	147	166	175	124
148	19	241	99	15	187	47	111
140	61	125	62	60	165	94	114
37	31	125	103	90	115	160	78
218	47	86	25	209	139	199	130

3

84

112

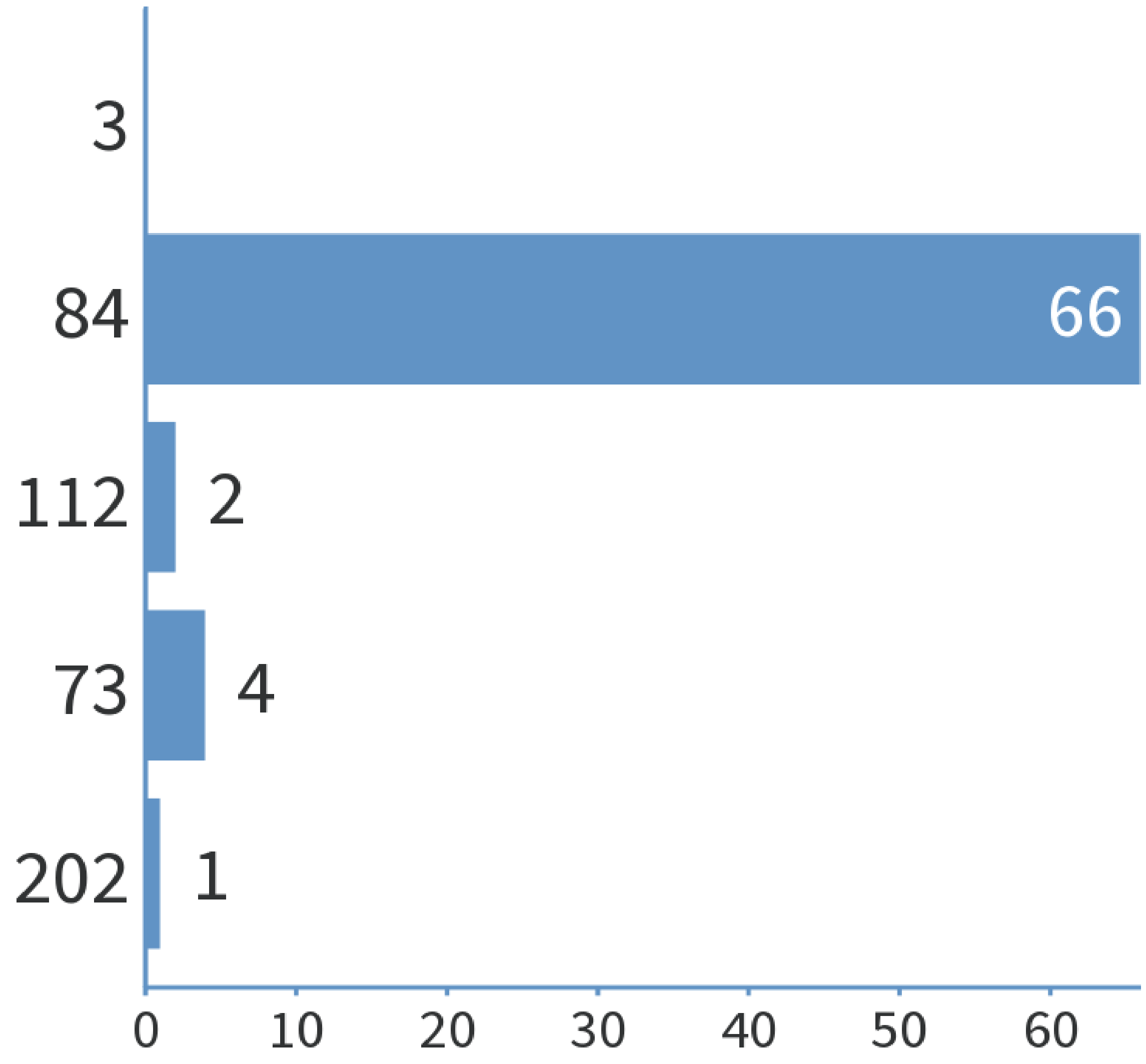
73

202

Rank filters on image

The image is filtered with a 3 x 3 median filter (medI). The image (the original) is also filtered with a 5 x 5 minimum rank filter (minI). The final image is made by subtracting minI from medI. What is the result in the marked pixel?

67	159	61	230	34	4	76	21
37	89	106	94	240	11	190	237
35	131	13	28	244	43	48	198
222	102	230	199	147	166	175	124
148	19	241	99	15	187	47	111
140	61	125	62	60	165	94	114
37	31	125	103	90	115	160	78
218	47	86	25	209	139	199	130

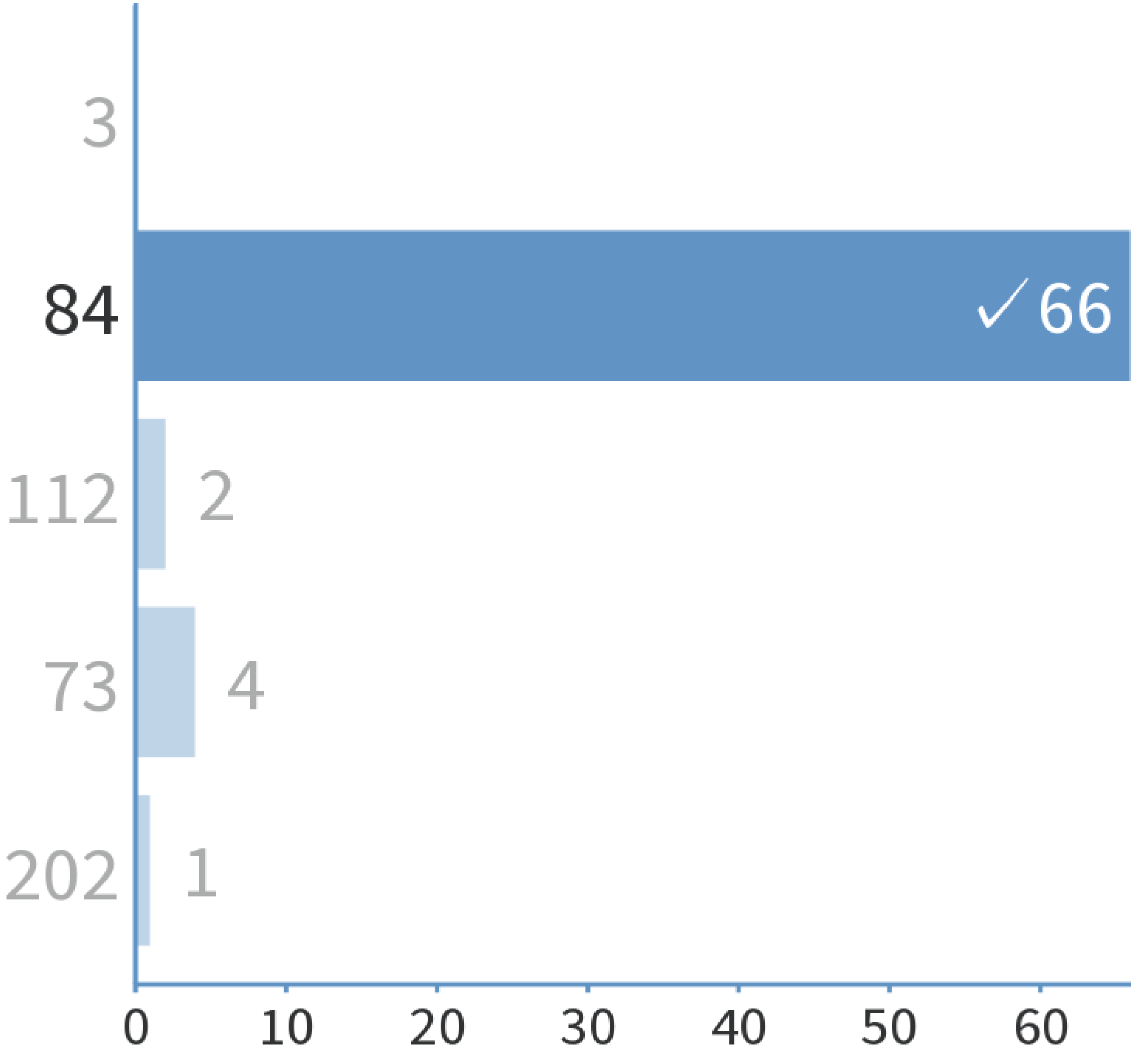


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

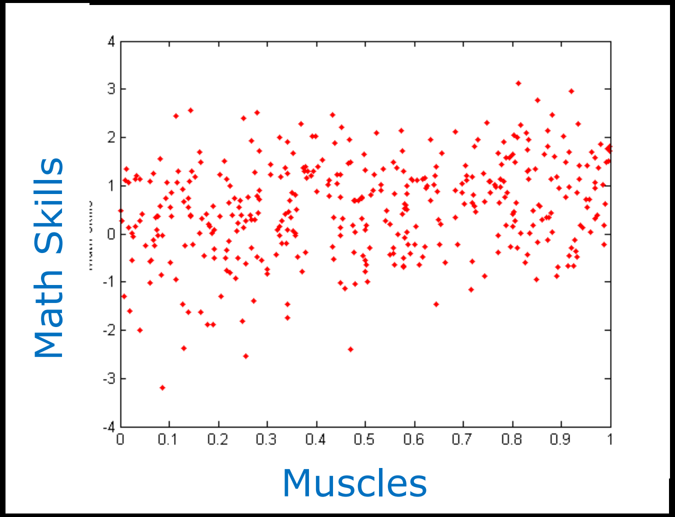
Rank filters on image

The image is filtered with a 3 x 3 median filter (medI). The image (the original) is also filtered with a 5 x 5 minimum rank filter (minI). The final image is made by subtracting minI from medI. What is the result in the marked pixel?

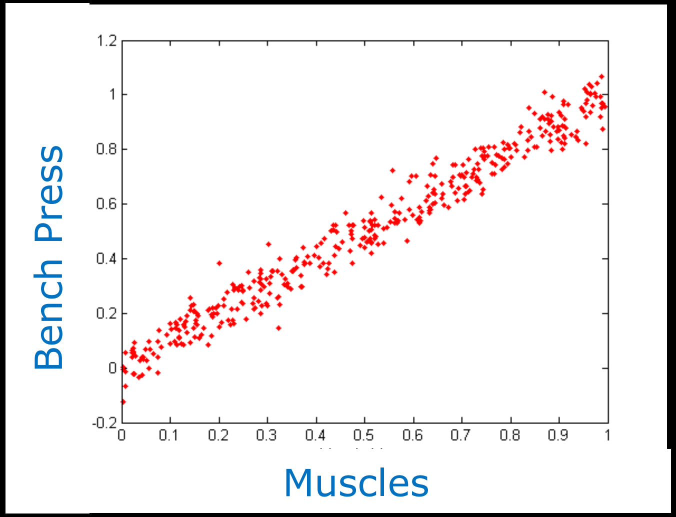
67	159	61	230	34	4	76	21
37	89	106	94	240	11	190	237
35	131	13	28	244	43	48	198
222	102	230	199	147	166	175	124
148	19	241	99	15	187	47	111
140	61	125	62	60	165	94	114
37	31	125	103	90	115	160	78
218	47	86	25	209	139	199	130



Correlation



Low

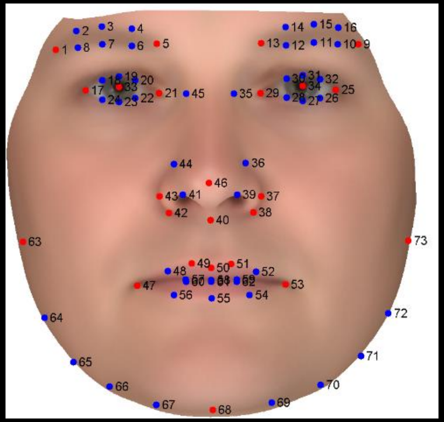


High

- What is it?
- Two measurements
  - Low correlation
  - High correlation
- High correlation means that there is a *relation* between the values
- They *look* the same
- Correlation is a *measure of similarity*

Why do we need similarity?

- Image analysis is also about recognition of patterns
- Often an example pattern is used
  - Often with some kind of meta data to apply to the targets
- We need something to tell us if there is a high match between our pattern and a part of the image



Example pattern

With meta information

Find matches

Correlation (1D)

Kernel

Signal (1D image)

$\frac{1}{4}$

1	2	1

1	1	2	2	1	1	2	2	1	1

$\boxed{1 * 1 + 2 * 1 + 1 * 2 = 5}$

Normalise! $\frac{1}{4}$

	$\frac{5}{4}$								

Correlation (1D)

$\frac{1}{4}$

1	2	1

1	1	2	2	1	1	2	2	1	1

	$\frac{5}{4}$	$\frac{7}{4}$	$\frac{7}{4}$	$\frac{5}{4}$	$\frac{5}{4}$	$\frac{7}{4}$	$\frac{7}{4}$	$\frac{5}{4}$	

Normalisation

- The sum of the kernel elements is used
- Keep the values in the same range as the input image

$\frac{1}{4}$

1	2	1

Sum is 4

1	1	2	2	1	1	2	2	1	1

1 * 1 + 2 * 1 + 1 * 2 = 5

Normalise! $\frac{1}{4}$

	$\frac{5}{4}$								

Normalisation

- Normalisation factor
  - Sum of kernel coefficients

$$h(x) \quad \boxed{1 \mid 2 \mid 1}$$

$$\sum_x h(x) = 1 + 2 + 1$$

Correlation on images

- The filter is now 2D

Kernel coefficients

$$\frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

Kernel

Input

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

Output

	$\frac{12}{9}$				

Correlation on images

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Input

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

Output

	$\frac{12}{9}$	$\frac{11}{9}$			

Correlation on images

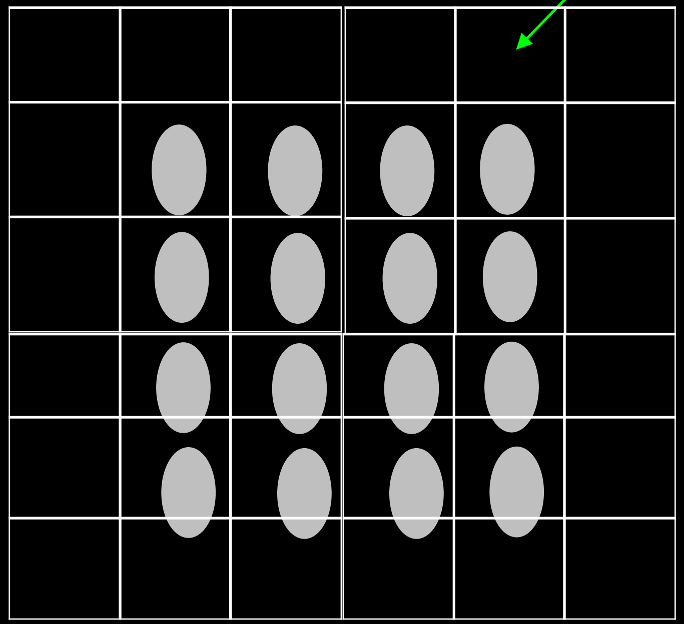
The mask is moved row by row

No values at the border!

Input

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

Output



Correlation on image - no kernel normalisation

1	2	1
1	3	1
1	2	1

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

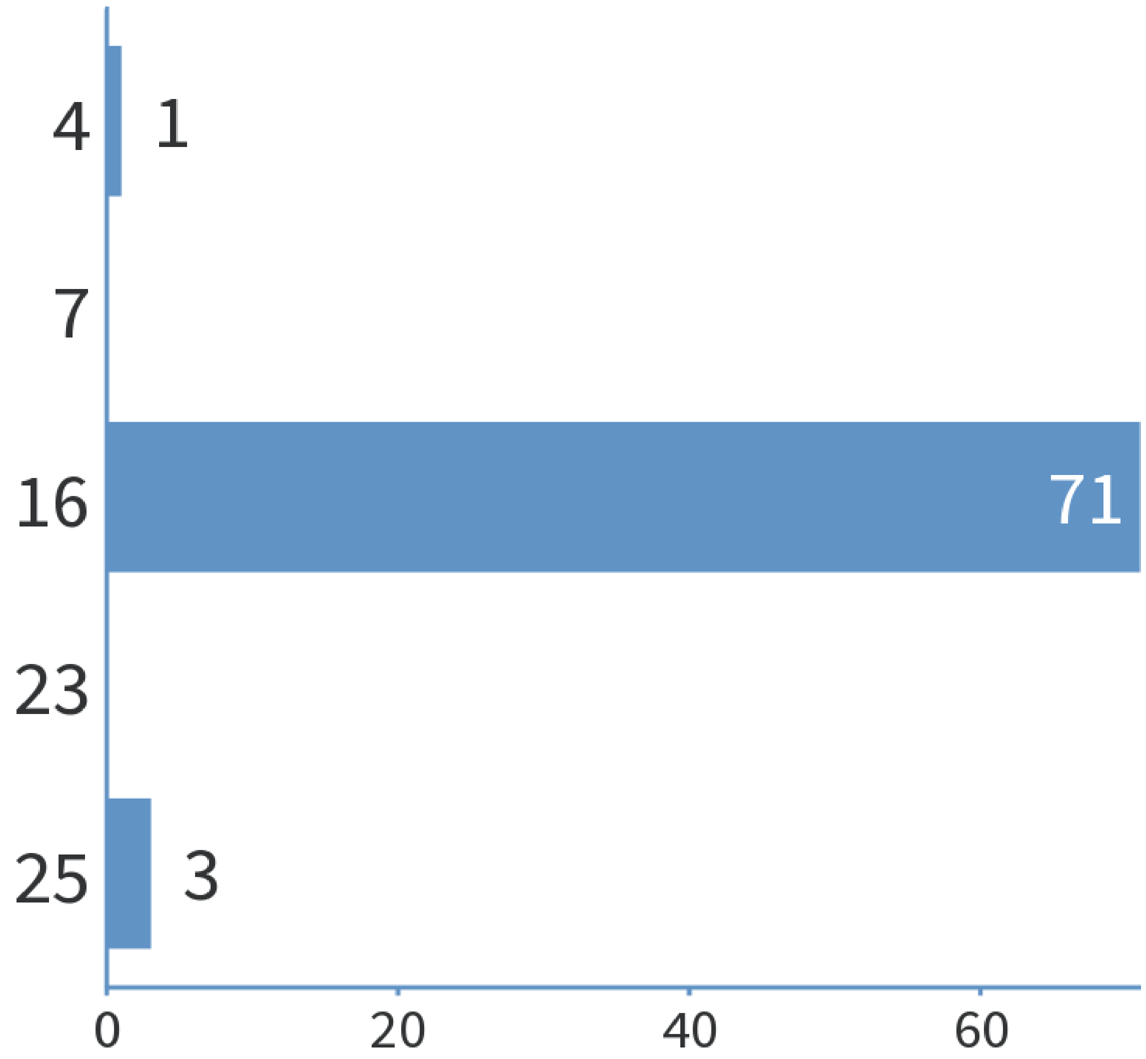
4
7
16
23
25

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Correlation on image - no kernel normalisation

1	2	1
1	3	1
1	2	1

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

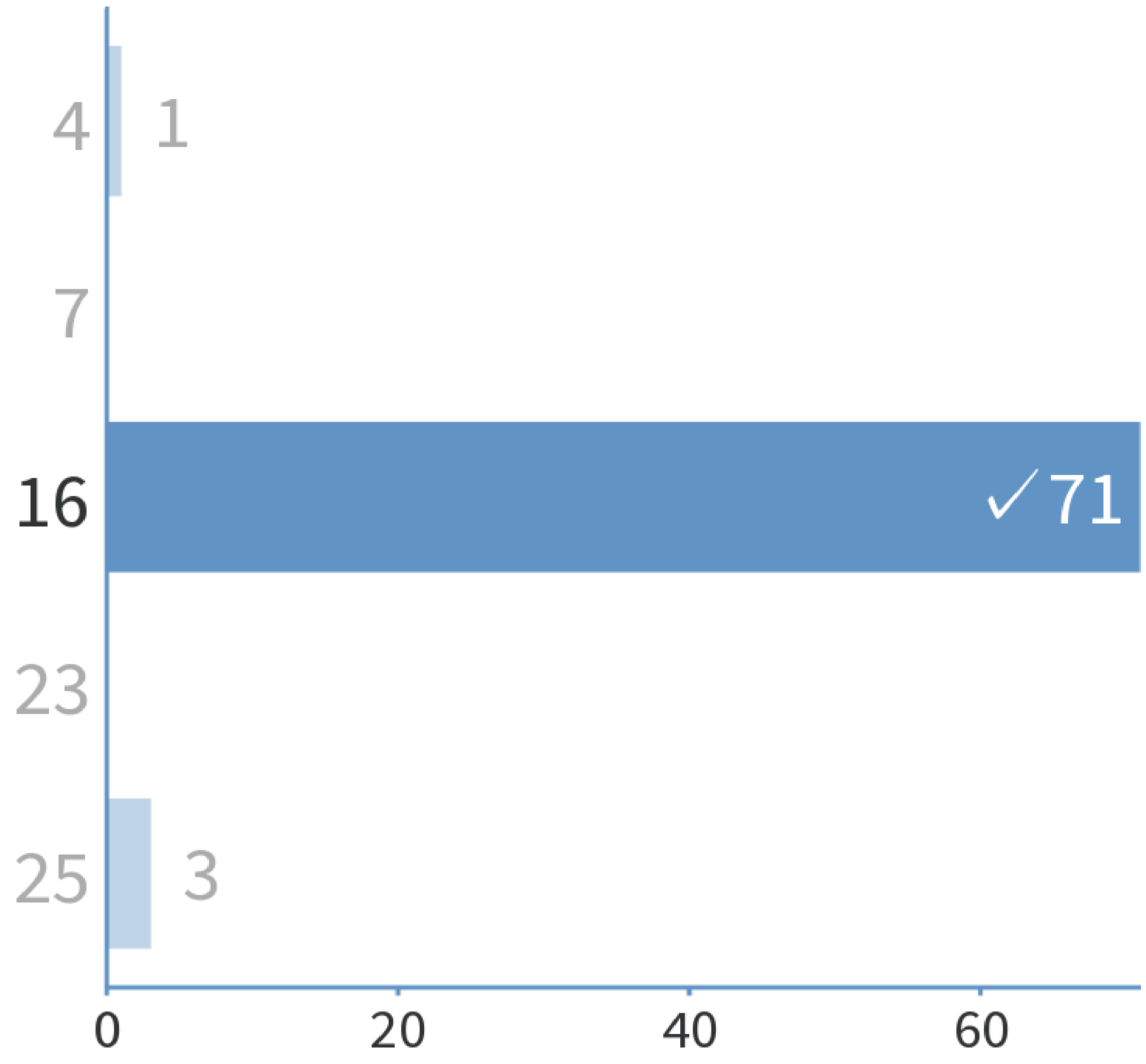


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Correlation on image - no kernel normalisation

1	2	1
1	3	1
1	2	1

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Correlation on image 2 - no kernel normalisation

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

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

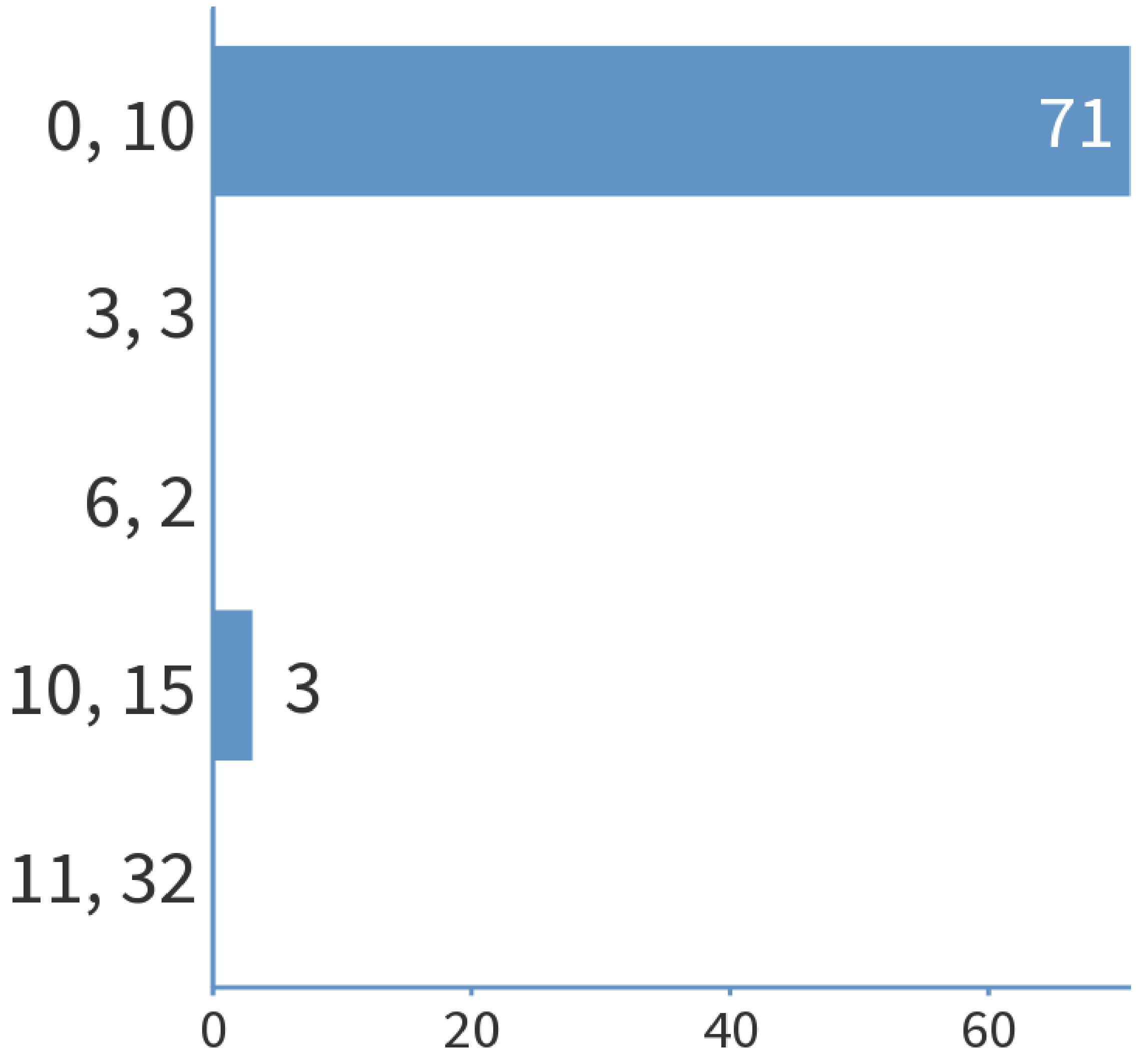
- 0, 10
- 3, 3
- 6, 2
- 10, 15
- 11, 32

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Correlation on image 2 - no kernel normalisation

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

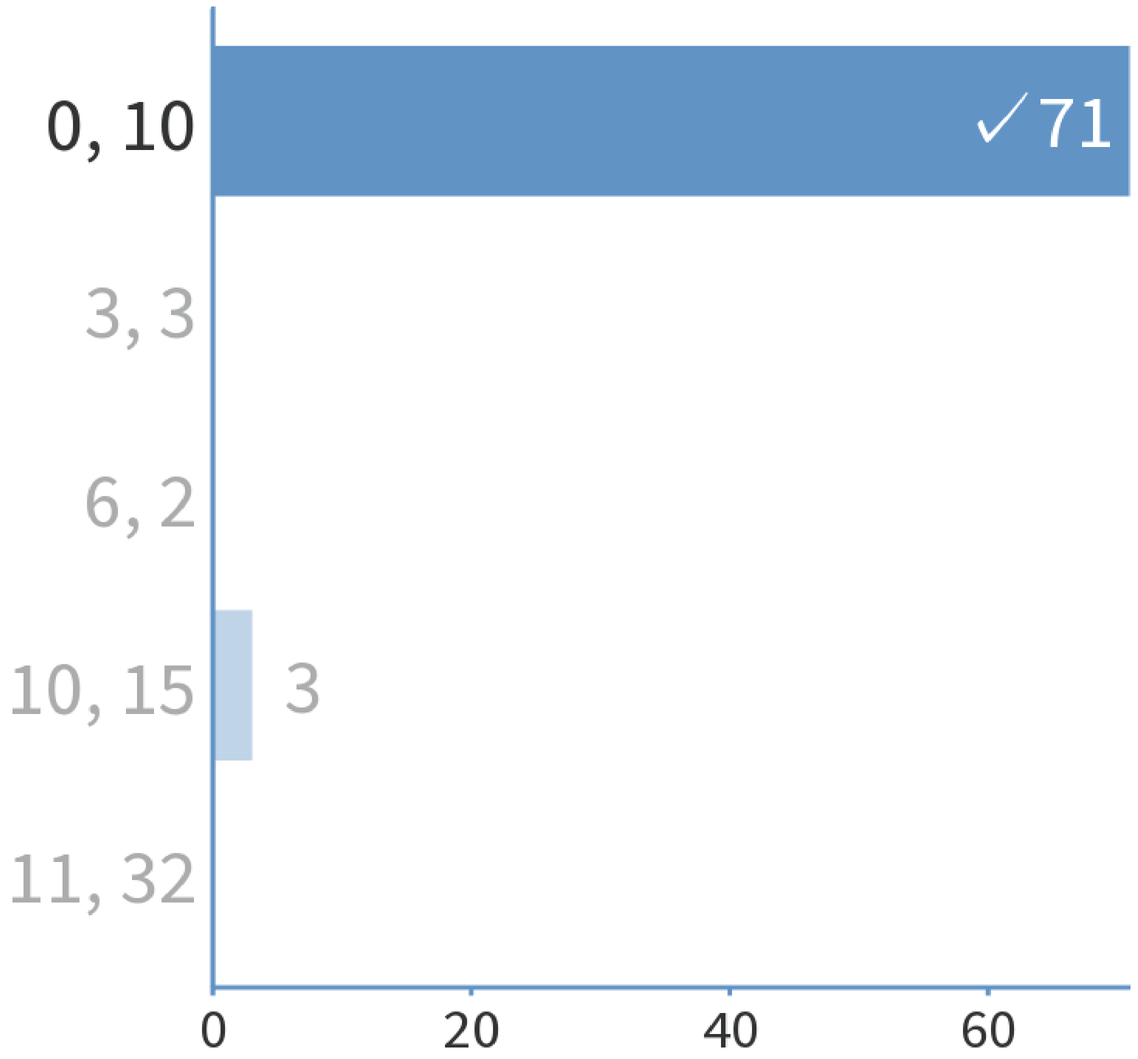
1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3



Correlation on image 2 - no kernel normalisation

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

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3



Mathematics of 2D Correlation

$$g(x,y) = f(x,y) \circ h(x,y)$$

Correlation operator

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

f

1	2	1
1	3	1
1	2	1

h

Mathematics of 2D Correlation

$$g(x,y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot f(x+i, y+j)$$

Example g(2,1)

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

f

i = -1, j = -1

i = 1, j = 0

1	2	1
1	3	1
1	2	1

h

h(0,0)

f(2,1)

Mathematics of 2D Correlation

$$g(x,y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot f(x+i, y+j)$$

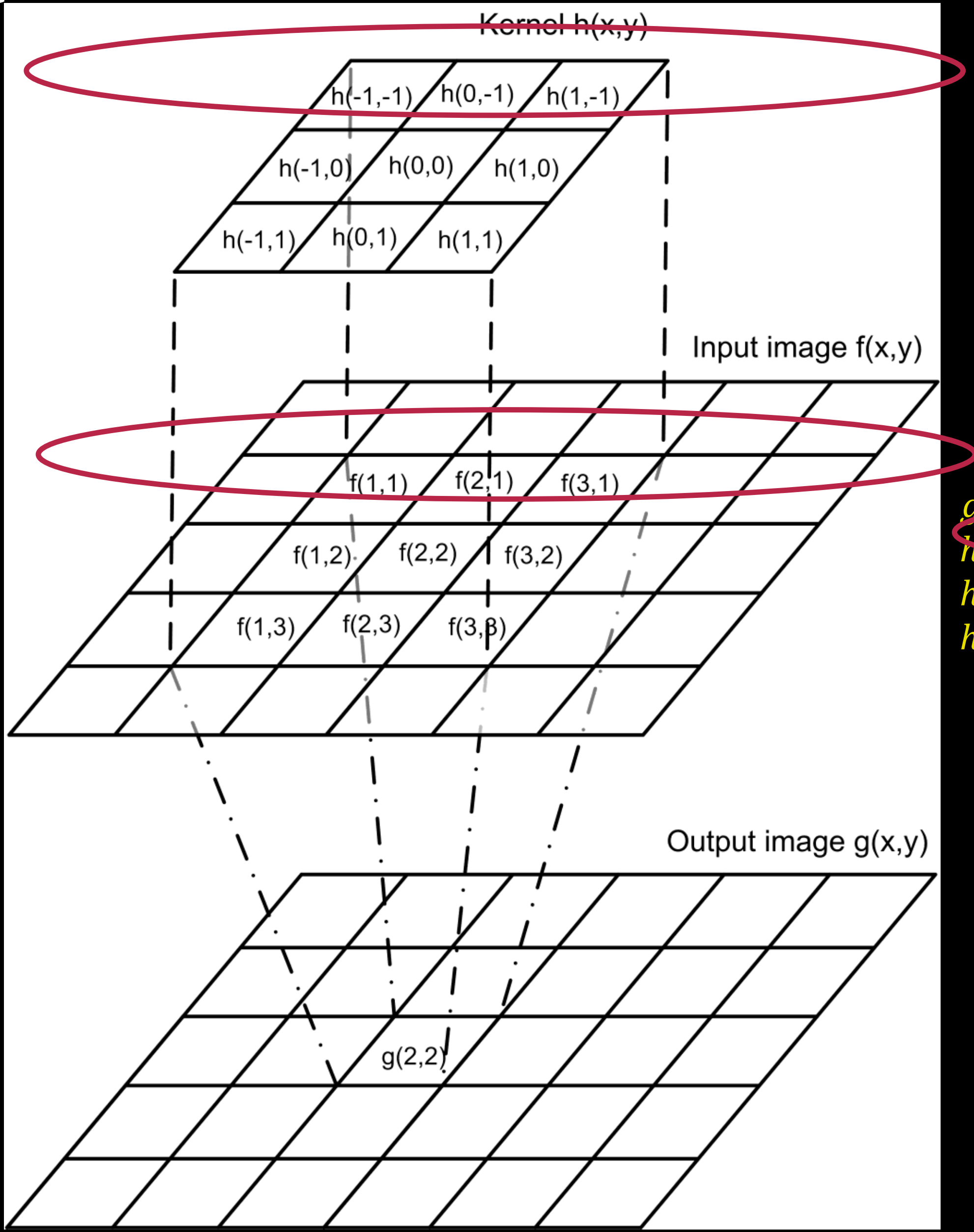
$$g(x,y) = 1 \cdot 2 + 2 \cdot 0 + 1 \cdot 1 + 1 \cdot 1 + 3 \cdot 4 + 1 \cdot 2 + 1 \cdot 0 + 2 \cdot 1 + 1 \cdot 0$$

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

f

1	2	1
1	3	1
1	2	1

h



$$g(x,y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot f(x+i, y+j)$$

$g(2,2) =$
$h(-1,-1) \cdot f(1,1) + h(0,-1) \cdot f(2,1) + h(1,-1) \cdot f(3,1) +$
$h(-1,0) \cdot f(1,2) + h(0,0) \cdot f(2,2) + h(1,0) \cdot f(3,2) +$
$h(-1,1) \cdot f(1,3) + h(0,1) \cdot f(2,3) + h(1,1) \cdot f(3,3)$

2D Kernel Normalisation

Normalisation factor:

$$\sum_{x}\sum_{y} h(x,y)$$

$$1+2+1+1+3+1+1+2+1=13$$

1	2	1
1	3	1
1	2	1

h

Template match on image

A template match is done on the image to the left with the template seen to the right. To find the best match the correlation is computed. What is the correlation in the marked pixel?

227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86	167	211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

66	232	37
204	46	35
110	67	222

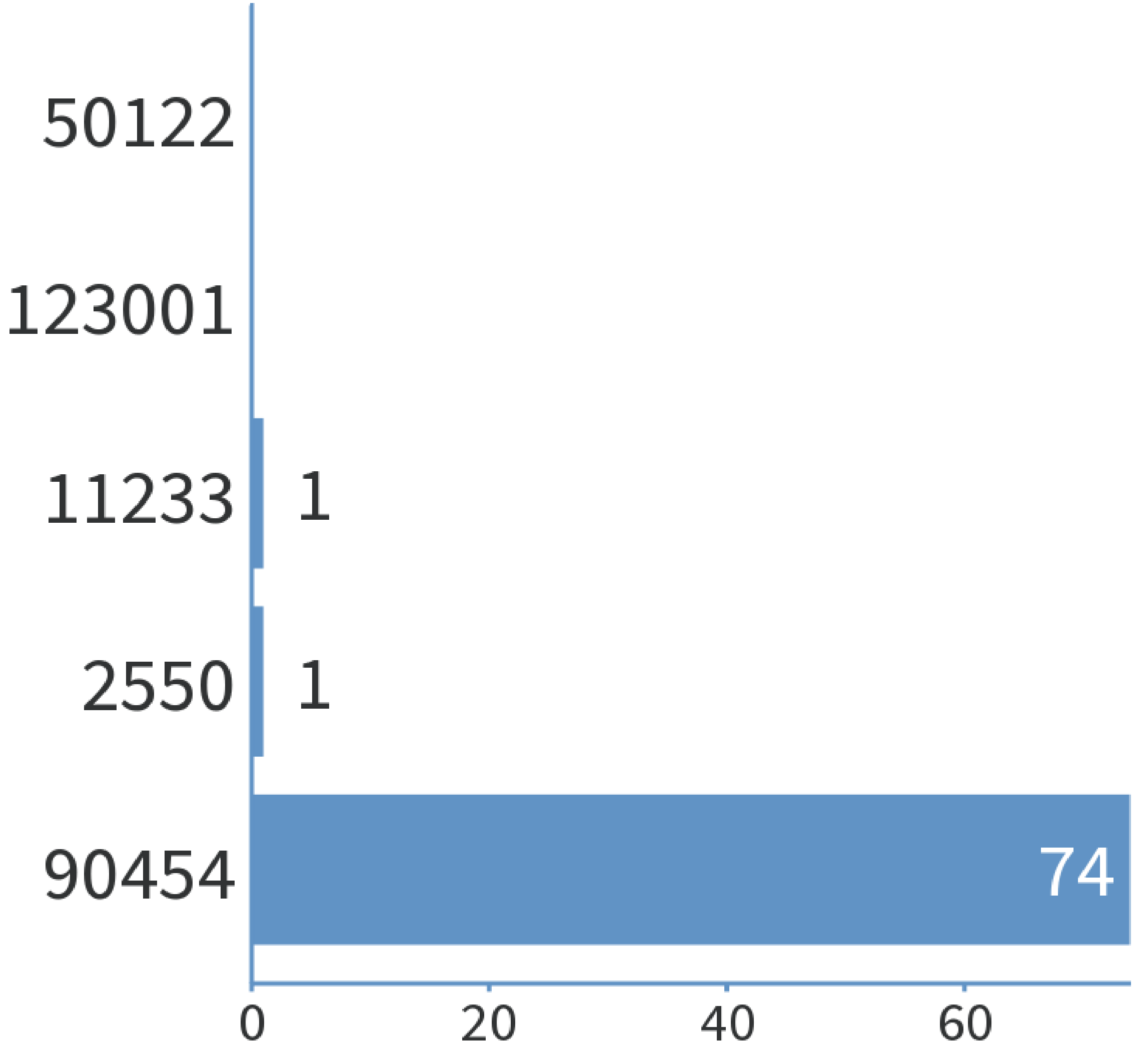
- 50122
- 123001
- 11233
- 2550
- 90454

Template match on image

A template match is done on the image to the left with the template seen to the right. To find the best match the correlation is computed. What is the correlation in the marked pixel?

227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86	167	211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

66	232	37
204	46	35
110	67	222



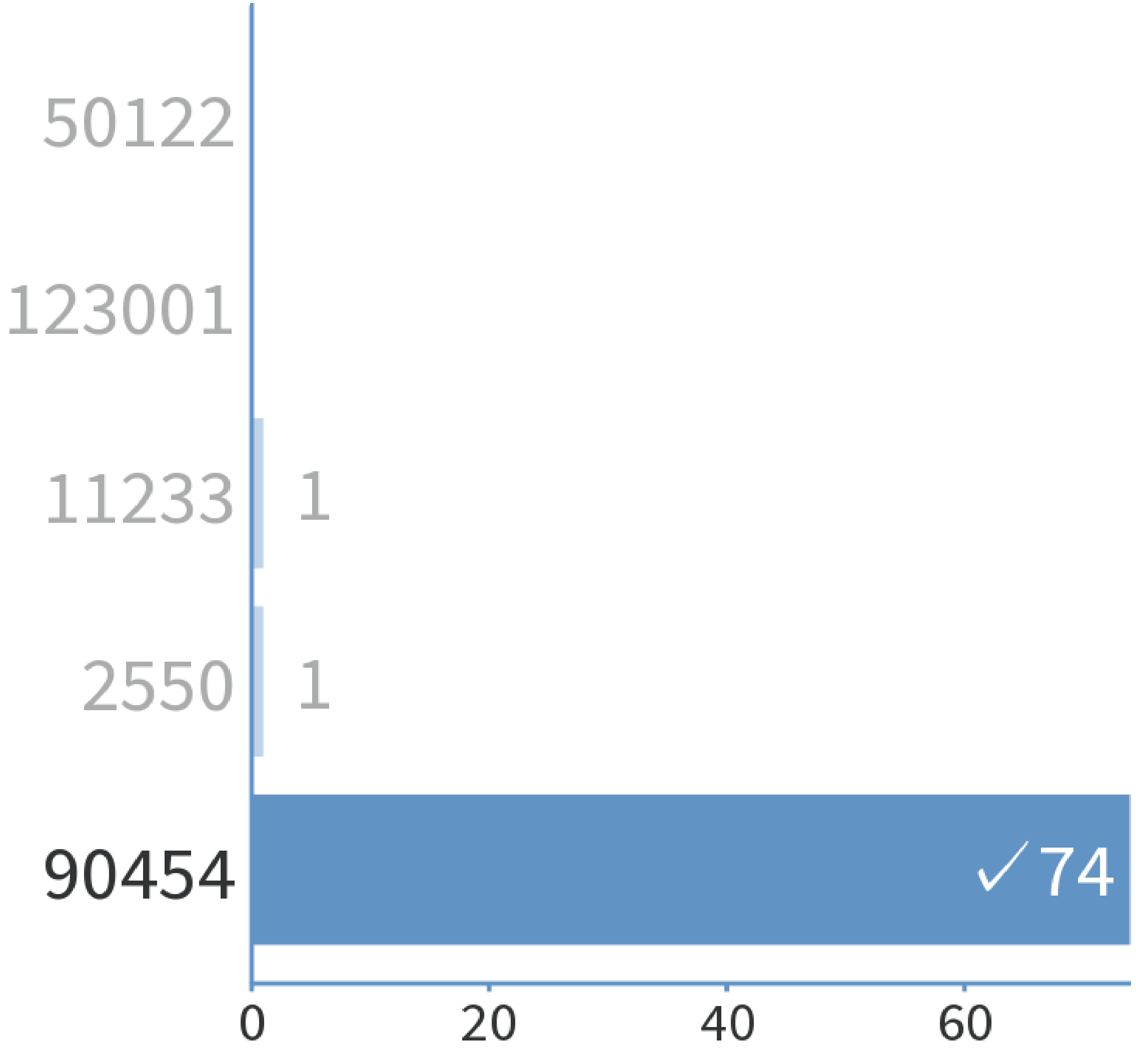
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Template match on image

A template match is done on the image to the left with the template seen to the right. To find the best match the correlation is computed. What is the correlation in the marked pixel?

227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86	167	211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

66	232	37
204	46	35
110	67	222



Smoothing filters

- Also know as
  - Smoothing kernel, Mean filter, Low pass filter, blurring
- The simplest filter:
  - *Spatial low pass filter*
  - Removes high frequencies

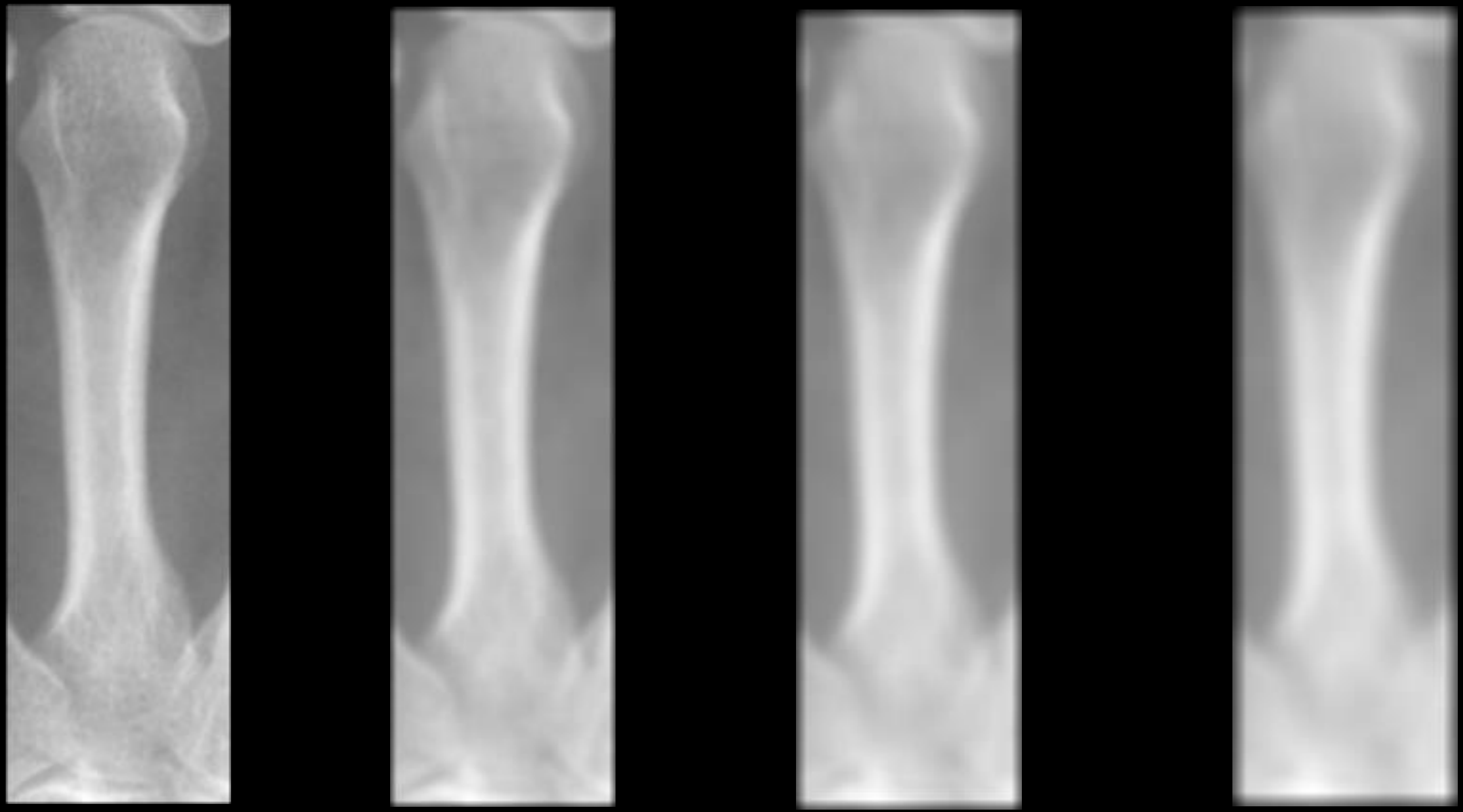
$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

- Another mask:
  - Gaussian filter

$$\frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

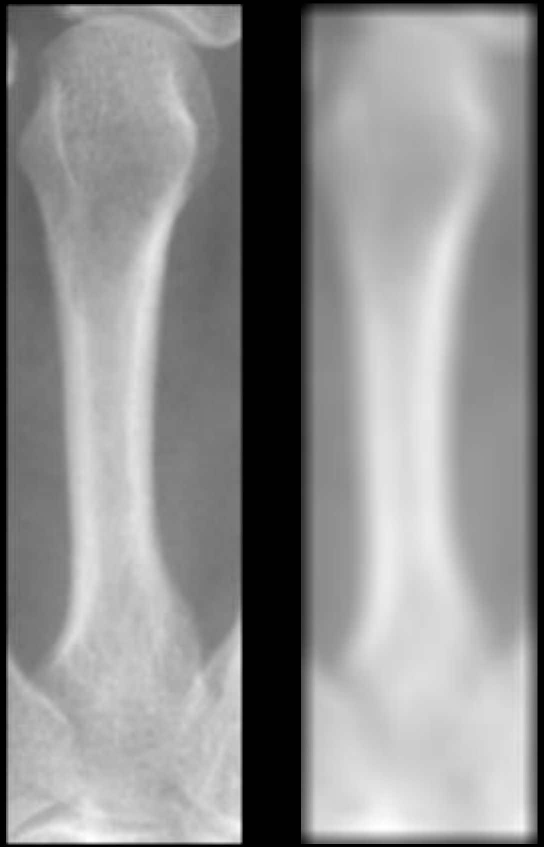
Why Gaussian?

Use of smoothing



3x3 7x7 11x11 15x15

Use of smoothing



3x3

15x15

- Large kernels smooth more
- Removes high frequency information
- Good at enhancing *big structures*

Mean filter on image - missing value

A 3x3 mean filter is applied to the image. The result in the marked pixel is 86. What is the value of the pixel, where the value is missing?

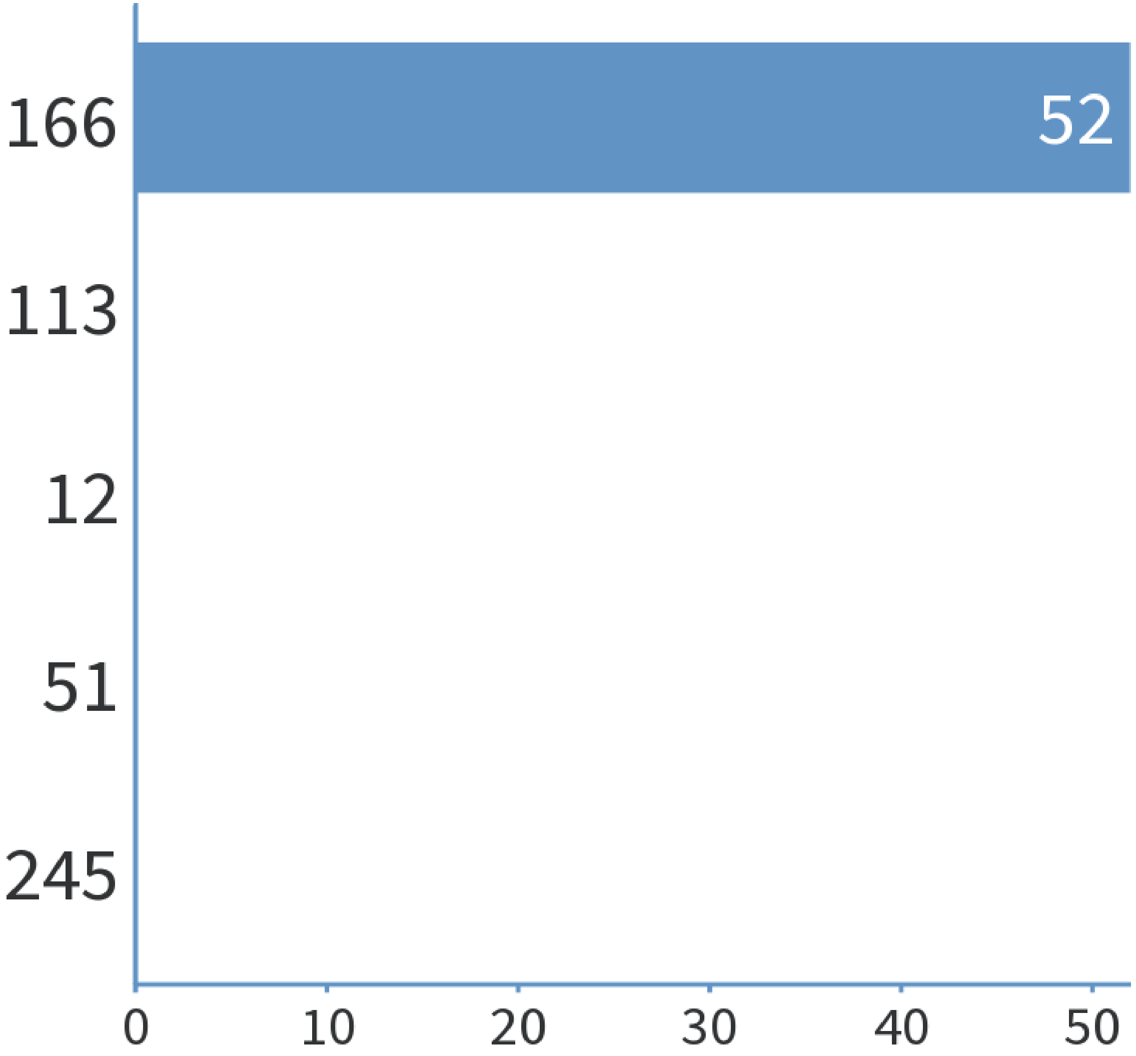
227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86		211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

- 166
- 113
- 12
- 51
- 245

Mean filter on image - missing value

A 3x3 mean filter is applied to the image. The result in the marked pixel is 86. What is the value of the pixel, where the value is missing?

227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86		211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

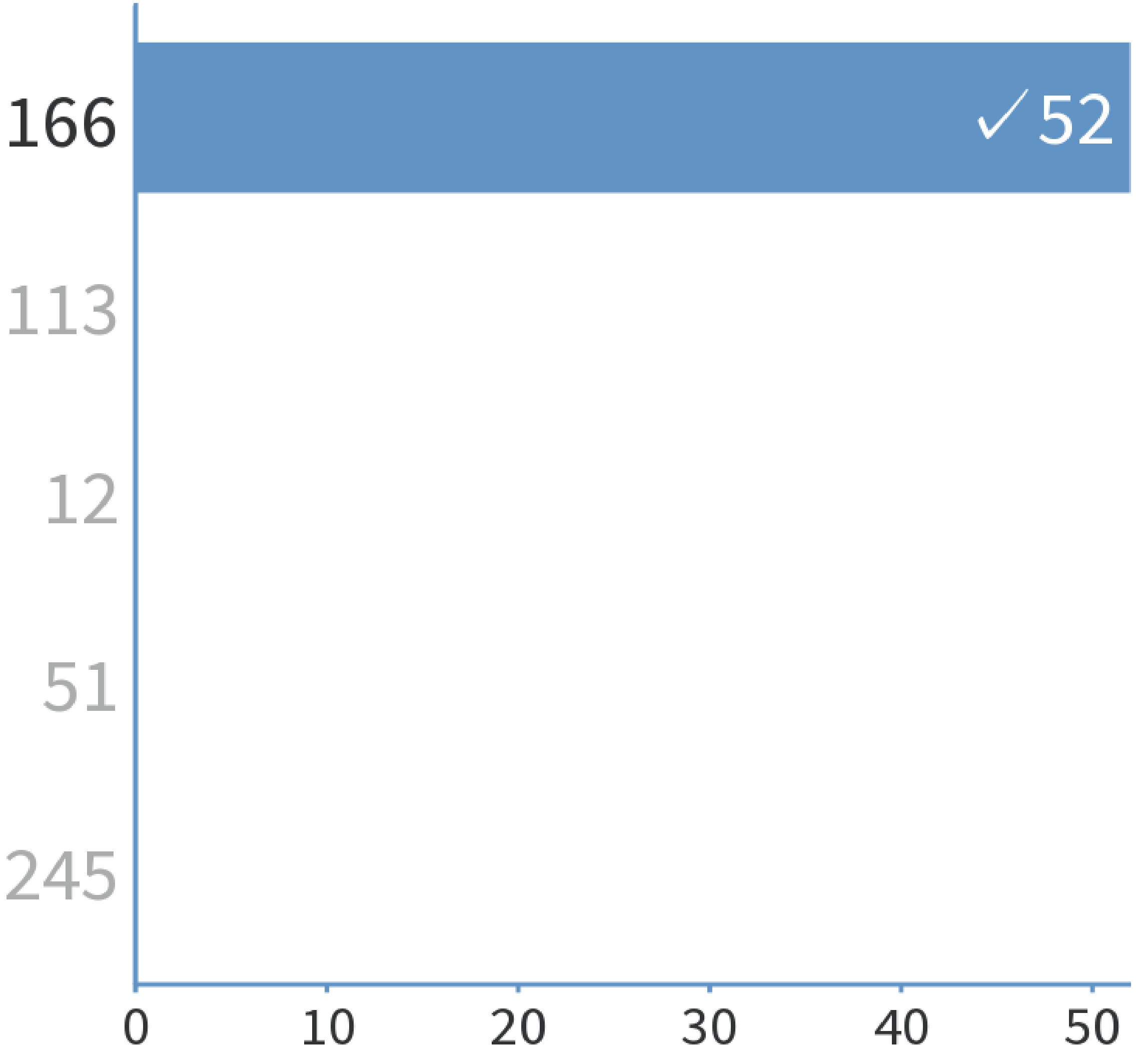


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Mean filter on image - missing value

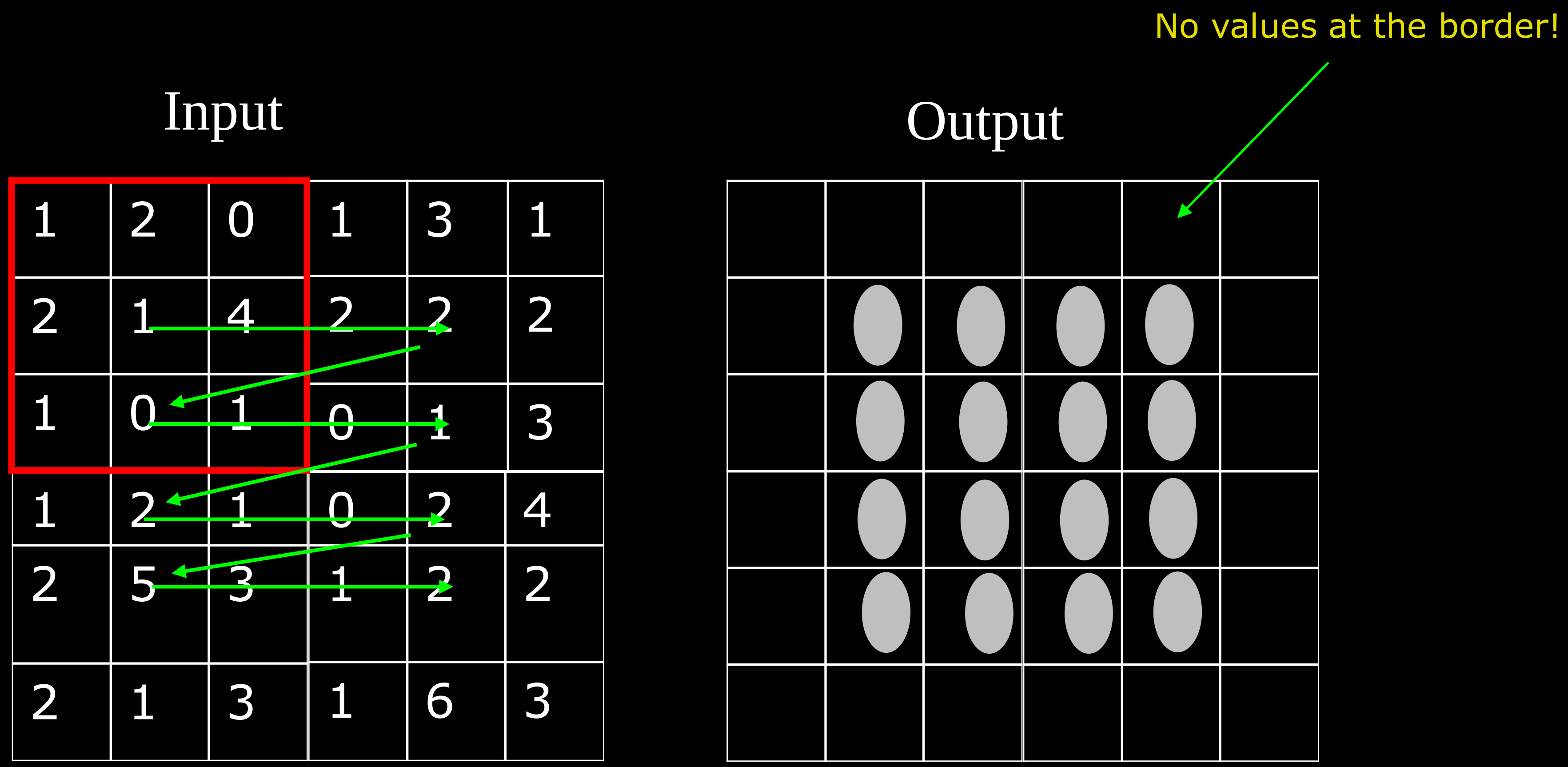
A 3x3 mean filter is applied to the image. The result in the marked pixel is 86. What is the value of the pixel, where the value is missing?

227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86		211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Border handling

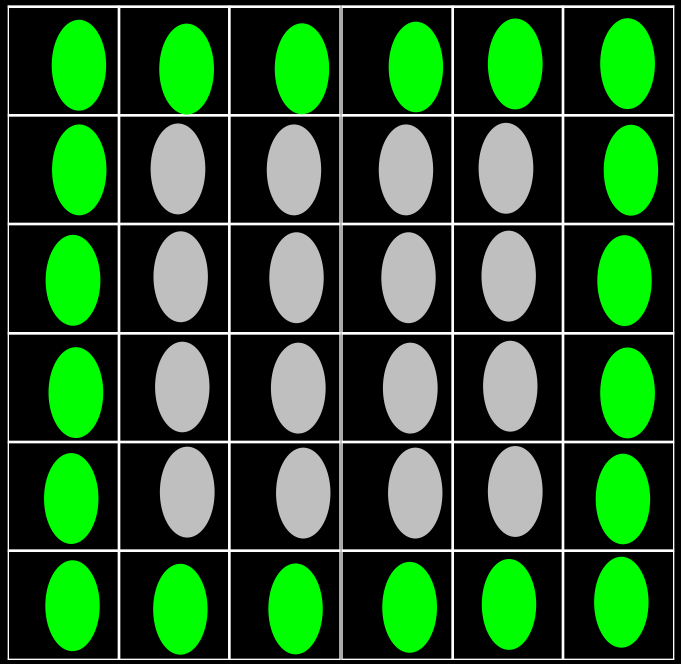


Border handling – extend the input

Input

0	0	0	0	0	0	
0	1	2	0	1	3	1
0	2	1	4	2	2	2
0	1	0	1	0	1	3
0	1	2	1	0	2	4
	2	5	3	1	2	2
	2	1	3	1	6	3

- Zero padding – what happens?
- Zero is black – creates dark border around the image



Correlation on image with zero padding

1	2	1
1	3	1
1	2	1

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

5, 8

12, 16

13, 3

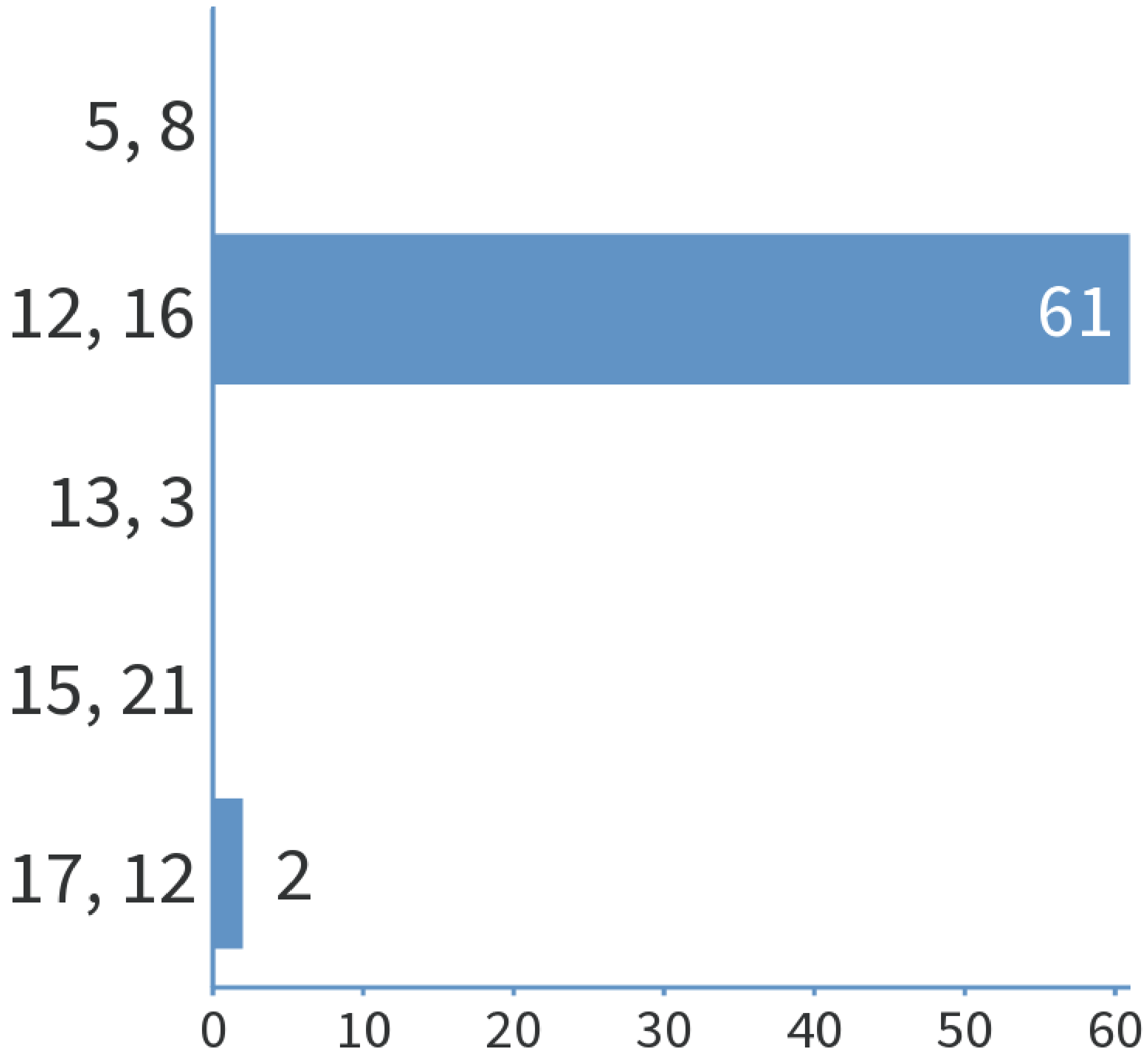
15, 21

17, 12

Correlation on image with zero padding

1	2	1
1	3	1
1	2	1

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

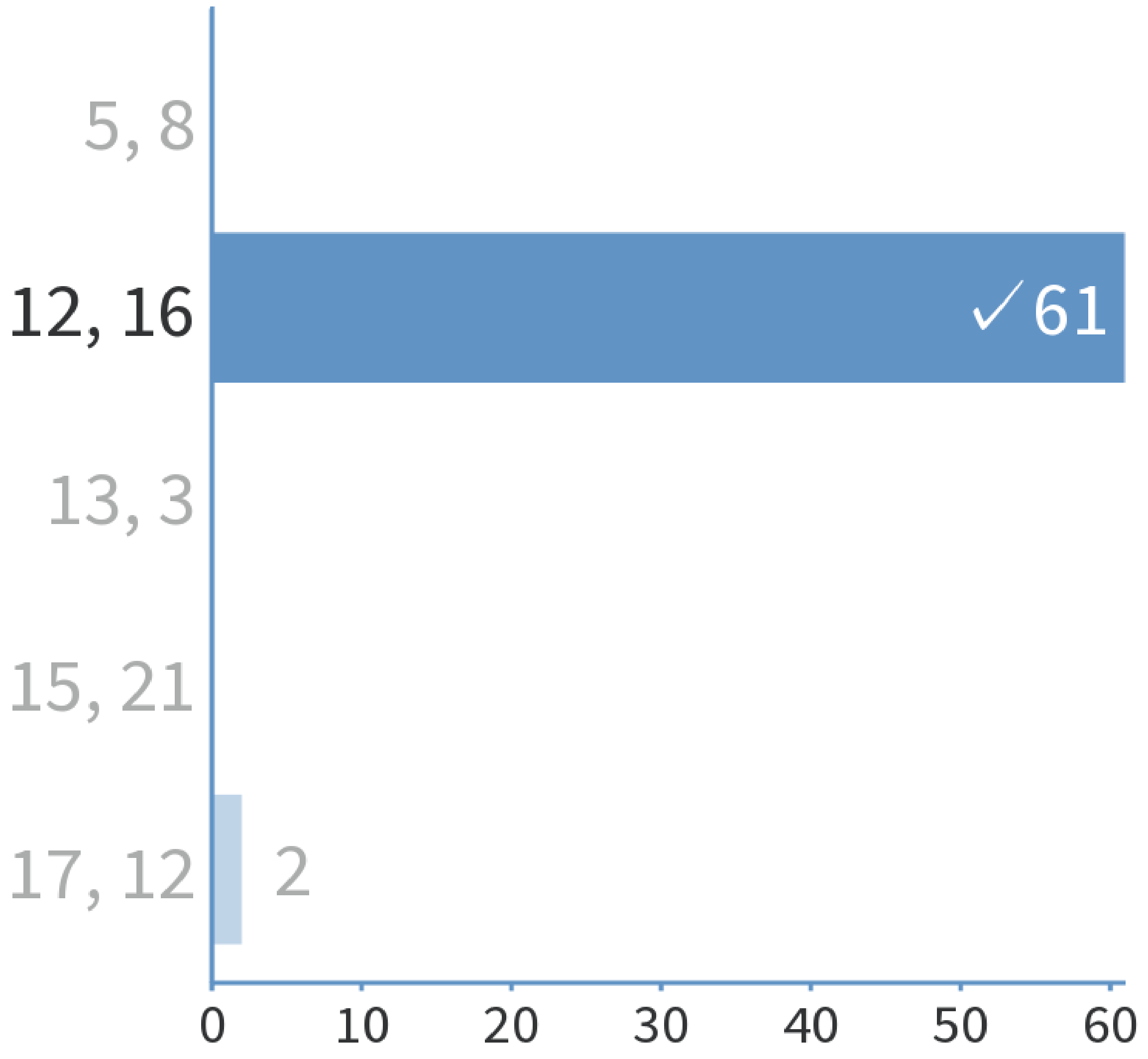


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Correlation on image with zero padding

1	2	1
1	3	1
1	2	1

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3



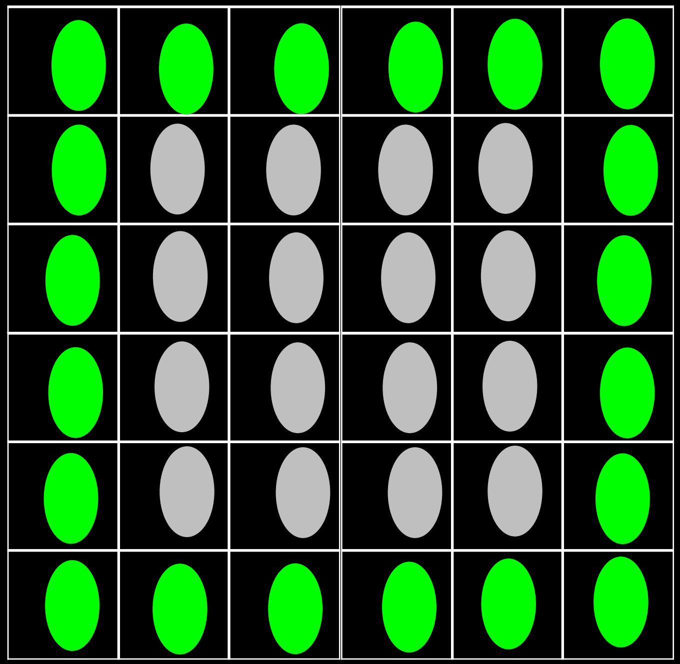
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Border handling – extend the input

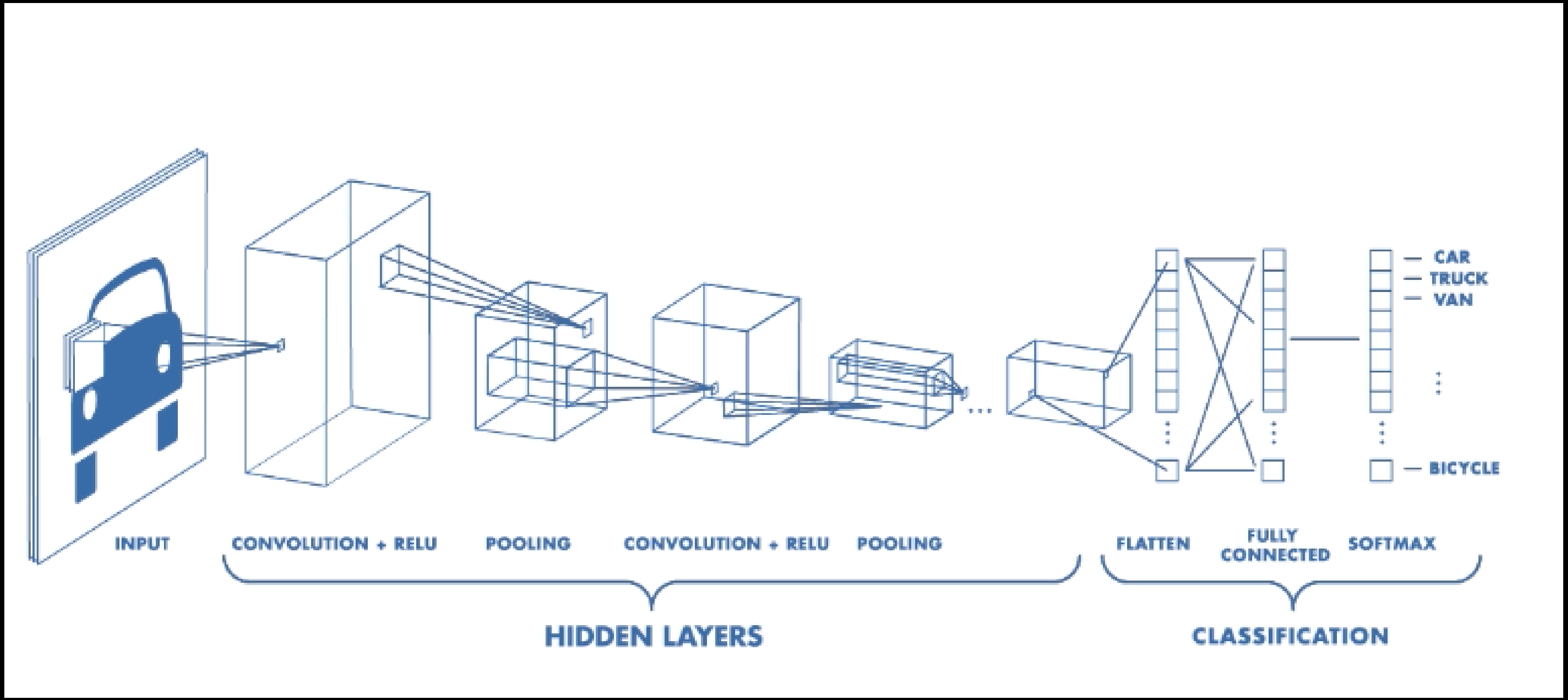
Input

1	1	2	0	1	3	
1	1	2	0	1	3	1
2	2	1	4	2	2	2
1	1	0	1	0	1	3
1	1	2	1	0	2	4
	2	5	3	1	2	2
	2	1	3	1	6	3

- Reflection
- Normally better than zero padding



What is the connection to deep learning?

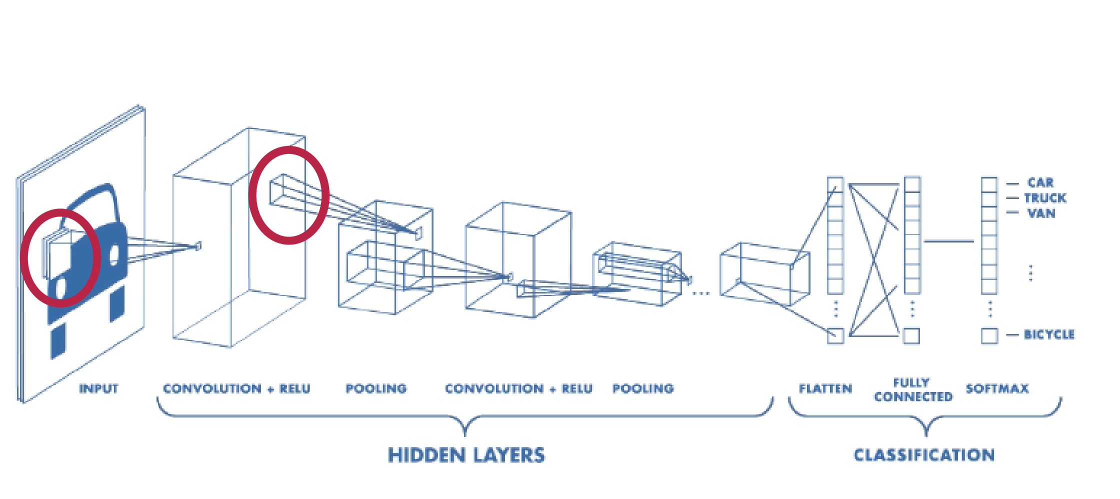


https://se.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html

Banks of filters

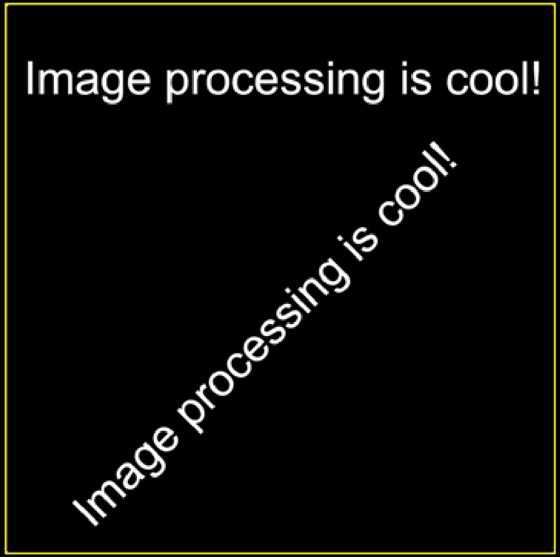
- The part of the network that touches the image consists of a bank of filters
  - Organised in a multi-level hierarchy

- The weights of the filters are adapted to the problem



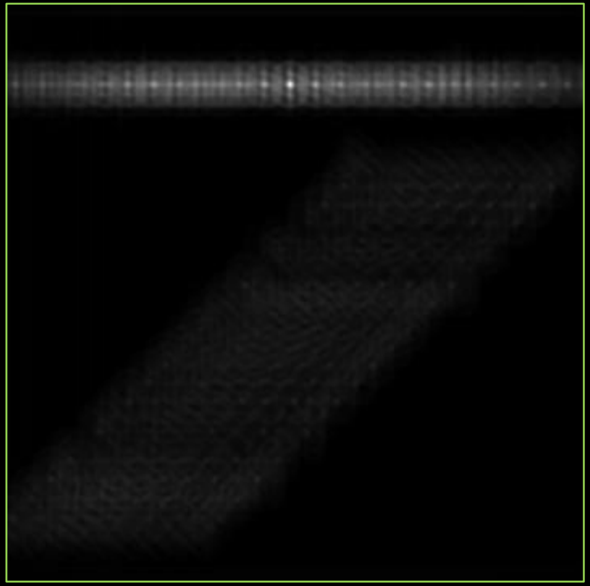
Template Matching

- Template
  - *Skabelon* på dansk
- Locates objects in images



Input

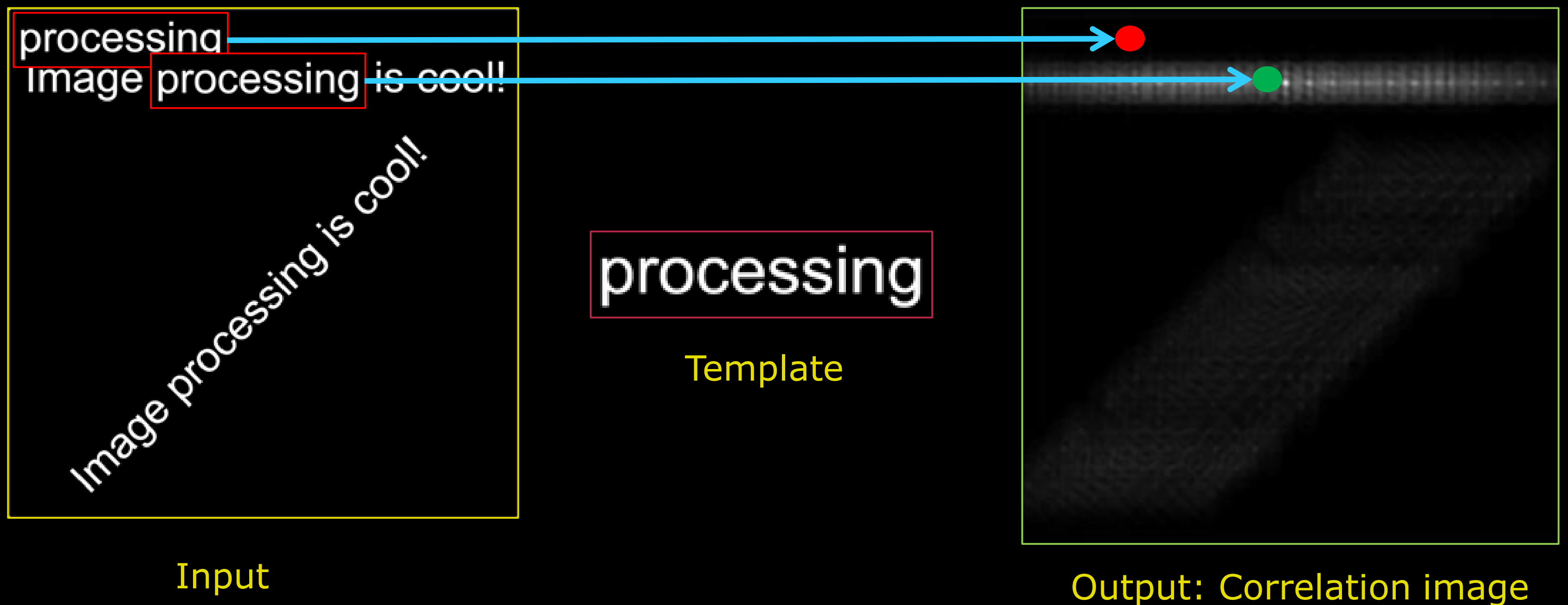
Template



Output: Correlation image

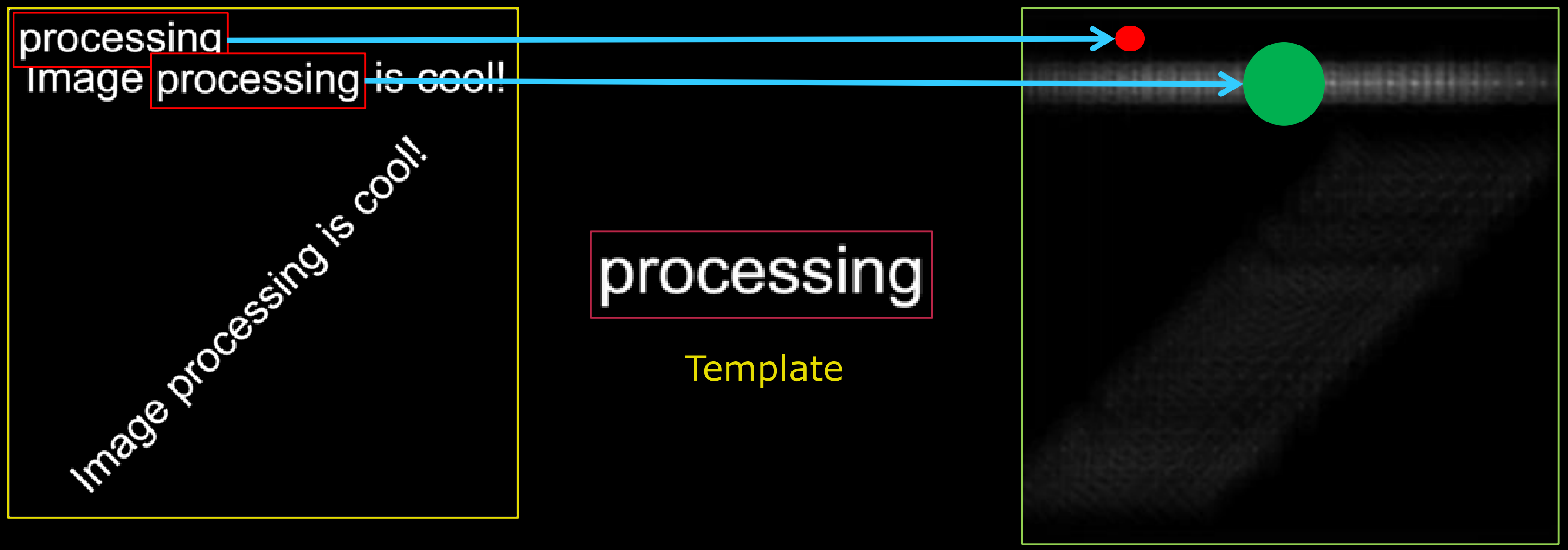
Template Matching

- The correlation between the template and the input image is computed for each pixel



Template Matching

- The pixel with the highest value is found in the output image
  - Here is the highest correlation

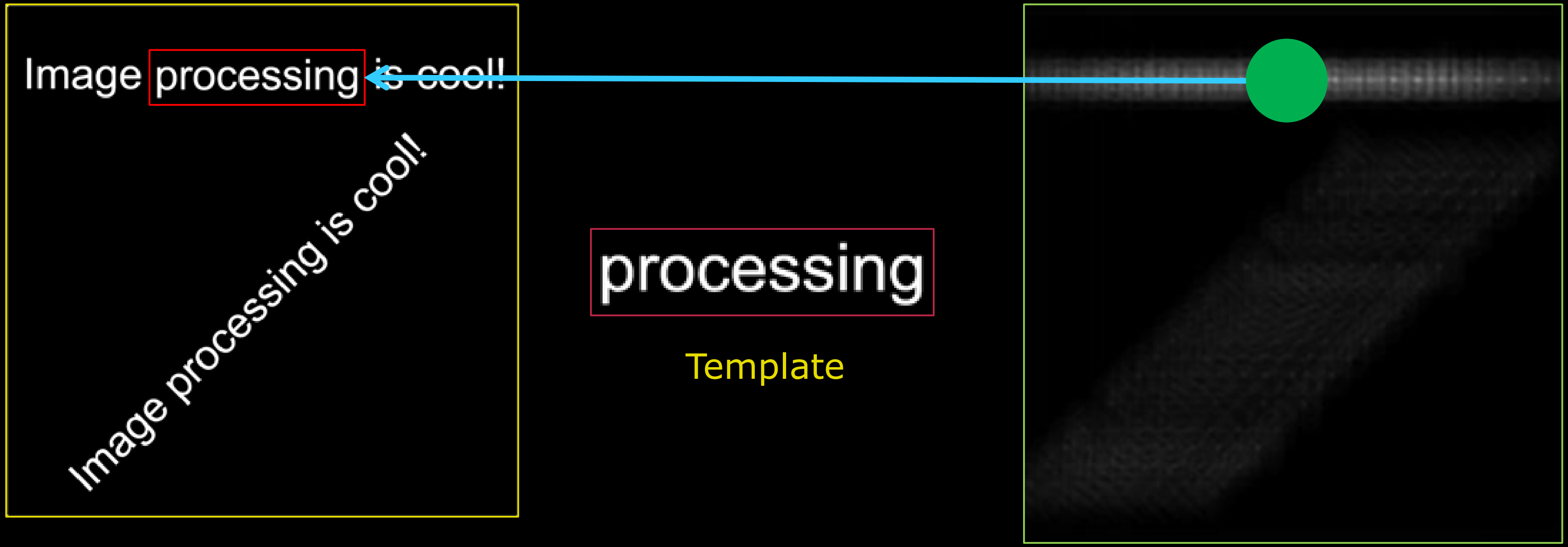


Input

Output: Correlation image

Template Matching

- This corresponds to the found pattern in the input image



Input

Output: Correlation image

Problematic Correlation

- Correlation matching has problem with light areas – why?

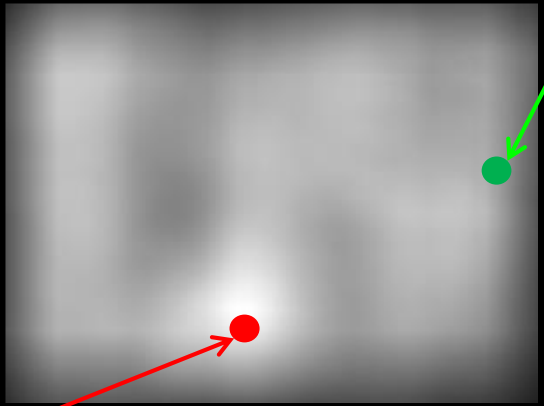
$$g(x,y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot f(x+i, y+j)$$

Very High!

Input (f)

Template (h)

Real max



Fake max

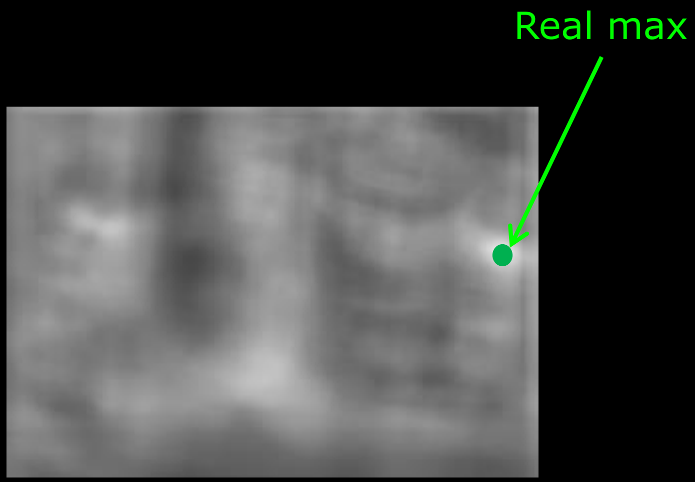
Output: Correlation image

Normalised Cross Correlation

$$\mathrm{NCC}(x, y) = \frac{\text{Correlation}}{\text{Length of image patch} \cdot \text{Length of template}}$$


Input (f)


Template (h)



Output: Correlation image

Length of template

- Vector length
  - Put all pixel values into a vector
  - Compute the length of this vector
- Describes the intensity of the template
  - Bright template has a large length
  - Dark template has a small length

$$\text{Length of template} = \sqrt{\sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot h(i,j)}$$

Template (h)

Length of image patch

- Vector length based on pixel values in image patch
- Describes the intensity of the image patch

Input (f) with patch

Template (h)

Normalised Cross Correlation

- The length of the image patch and the length of template normalise the NCC
- If the image is very bright the NCC will be "pulled down"

$$\mathrm{NCC}(x, y) = \frac{\text{Correlation}}{\text{Length of image patch} \cdot \text{Length of template}}$$

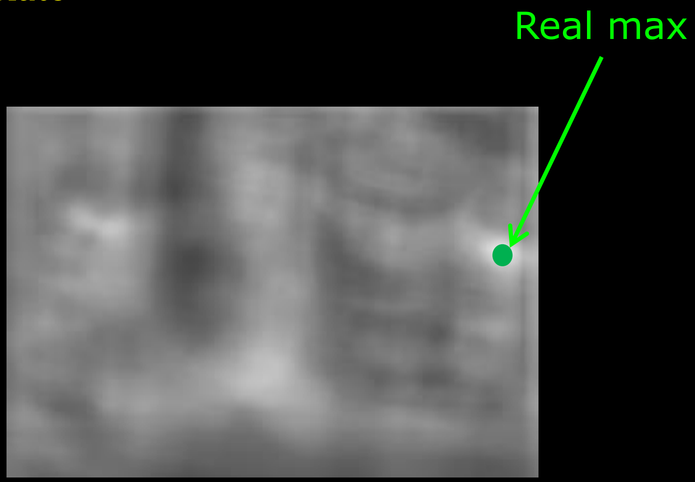
Normalised Cross Correlation

- NCC will be between
  - 0 : No similarity between template and image patch
  - 1 : Template and image patch are identical

$$\mathrm{NCC}(x,y) = \frac{\text{Correlation}}{\text{Length of image patch} \cdot \text{Length of template}}$$

Input (f)

Template (h)



Output: Correlation image

Normalised cross correlation on image

A template match using normalised cross correlation is performed. What is the resulting value in the marked pixel?

227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86	167	211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

66	232	37
204	46	35
110	67	222

0.10

0.33

0.83

0.62

0.98

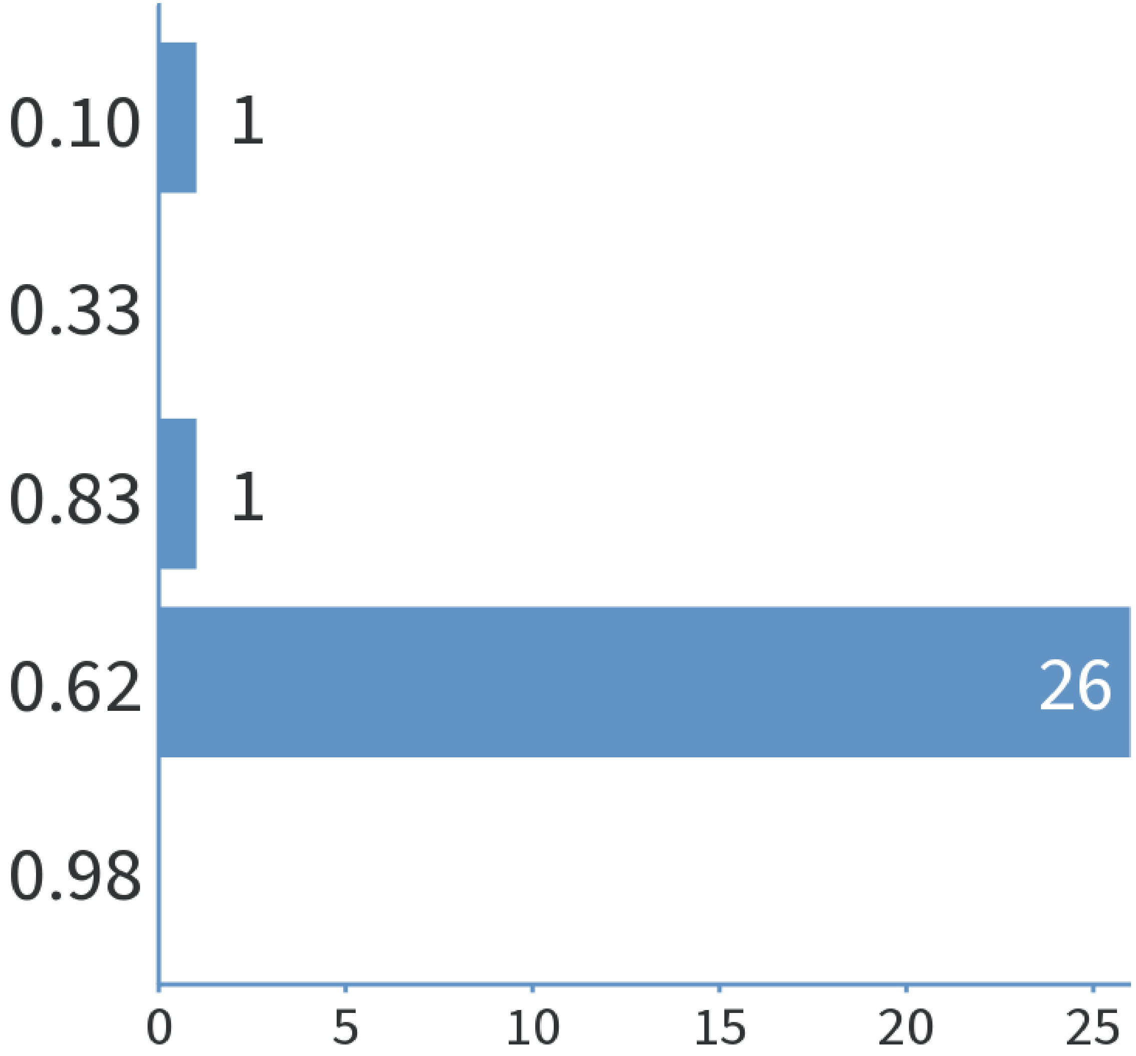
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Normalised cross correlation on image

A template match using normalised cross correlation is performed. What is the resulting value in the marked pixel?

227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86	167	211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

66	232	37
204	46	35
110	67	222

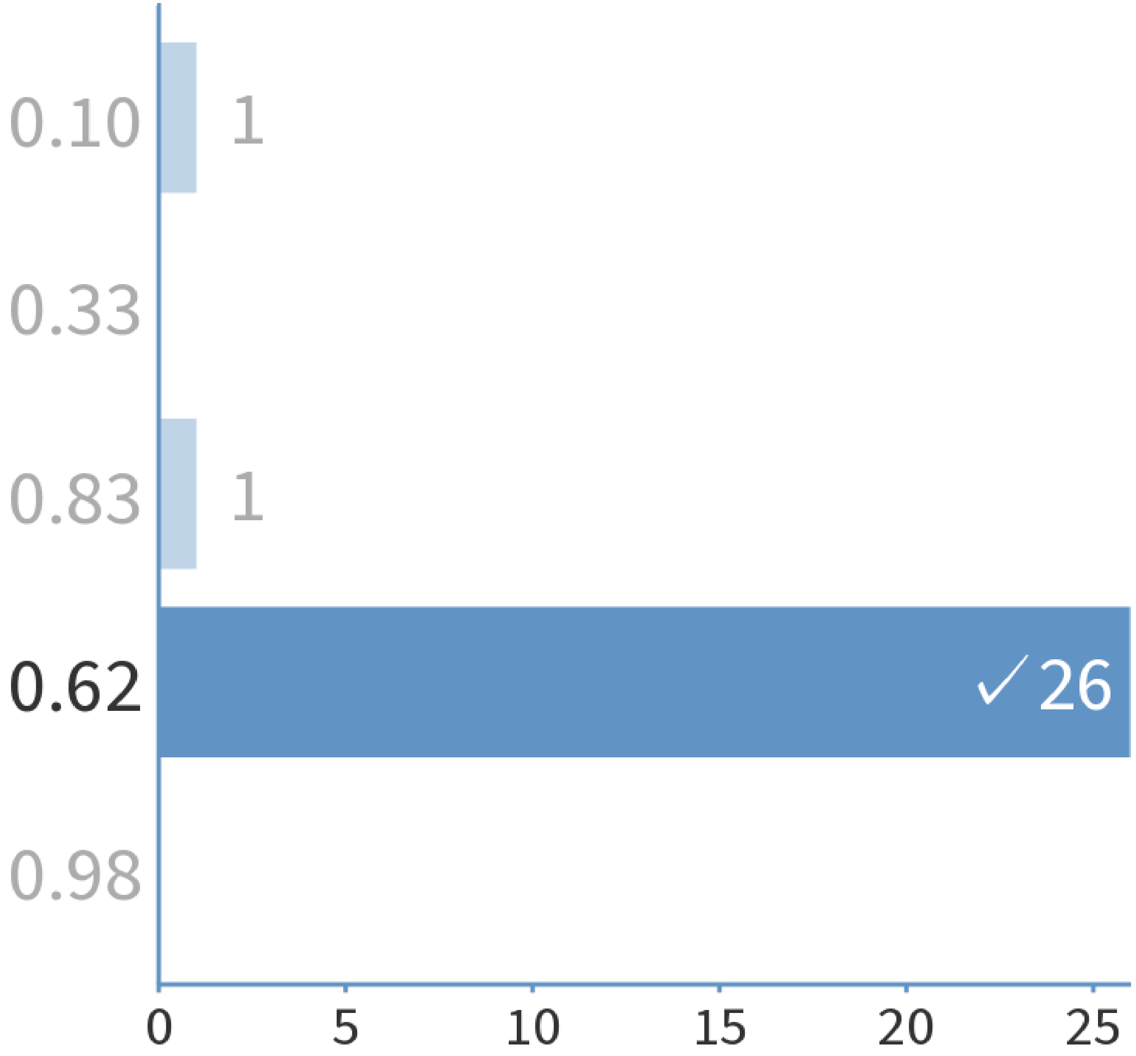


Normalised cross correlation on image

A template match using normalised cross correlation is performed. What is the resulting value in the marked pixel?

227	208	90	97	145	42	58	27
245	62	212	145	120	154	233	245
140	237	149	19	3	67	39	1
35	89	140	14	86	167	211	198
38	50	234	135	41	176	137	208
66	64	73	199	203	191	254	222
214	157	193	238	79	115	20	22
65	121	192	33	135	21	113	102

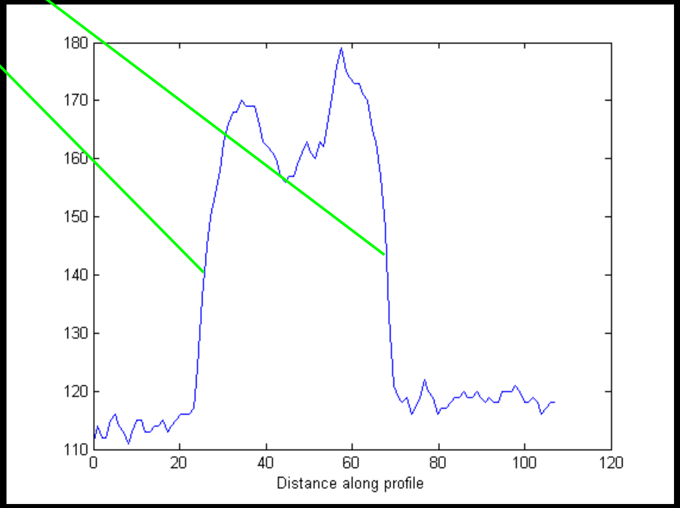
66	232	37
204	46	35
110	67	222



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

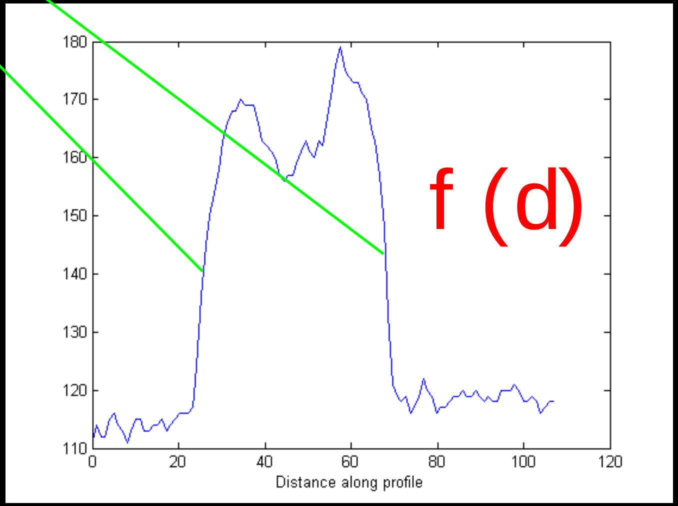
Edges

Gray level profile



- An edge is where there is a high change in gray level values
- Objects are often separated from the background by edges

Edges



$$f'(d)$$

- The profile as a function f(d)
- What value is high when there is an edge?
  - The slope of f
  - The slope of the tangent at d

Finite Difference

- Definition of slope

$$f'(d) = \lim_{h \to 0} \frac{f(d+h) - f(d)}{h}$$

- Approximation

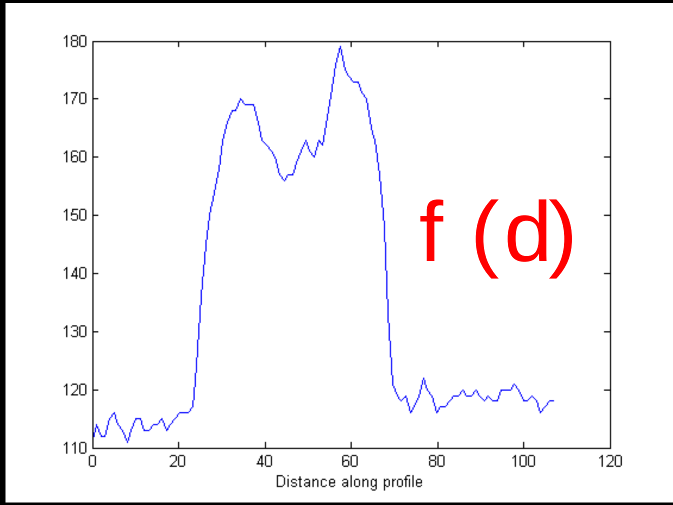
$$f'(d) \approx \frac{f(d+h) - f(d)}{h}$$

- Simpler approximation

h = 1

$$f'(d) \approx f(d+1) - f(d)$$

Edges



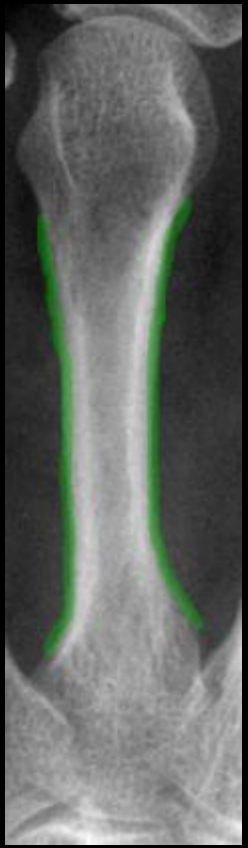
- Discrete approximation of f'(d)
- Can be implemented as a filter

$$f'(d) \approx f(d+1) - f(d)$$

-1	0	1

f(d-2)	f(d-1)	f(d)	f(d+1)	f(d+2)	f(d+3)

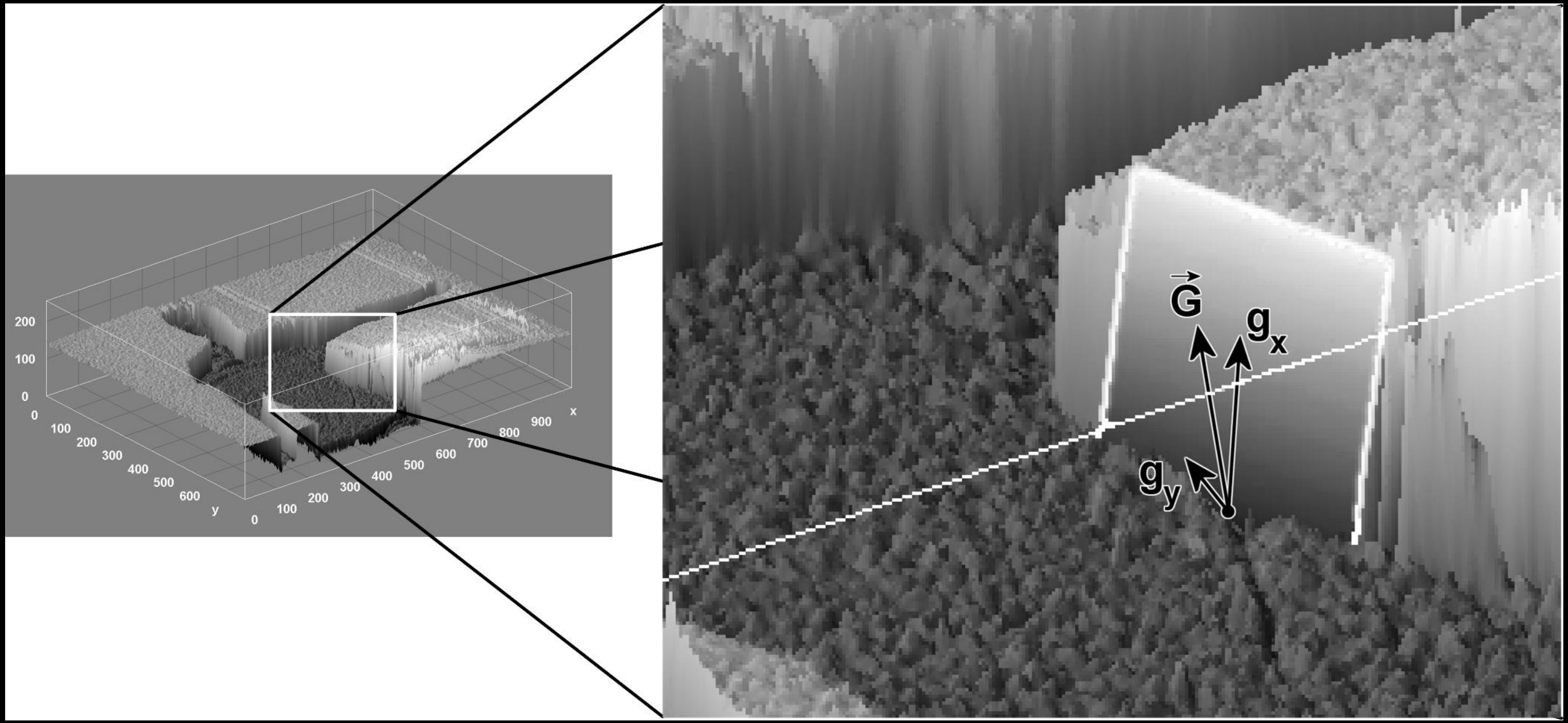
Edges in 2D



- Changes in gray level values
  - Image gradient
  - Gradient is the 2D derivative of a 2D function f(x,y)
  - Equal to the *slope* of the image
  - A steep slope is equal to an edge

$$\nabla f(x,y) = \vec{G}(g_x, g_y)$$

2D Gradient



$$\text{magnitude} = \sqrt{g_x^2 + g_y^2}$$

Edge filter kernel

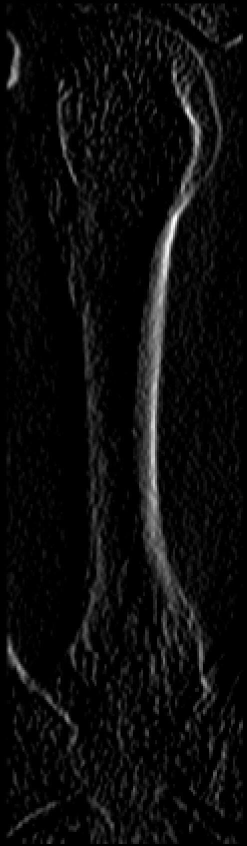
-1	0	1
-1	0	1
-1	0	1

Vertical Prewitt filter

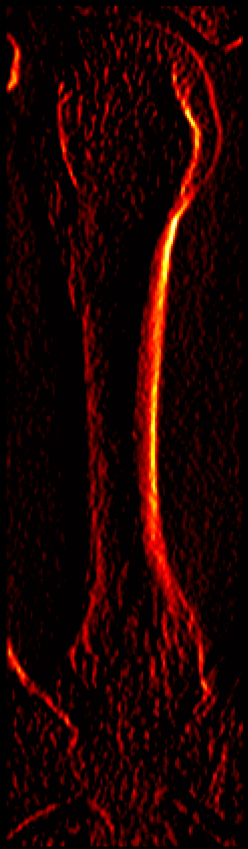
- The Prewitt filter is a typical edge filter
- Output image has high values where there are edges

Prewitt filter

Original

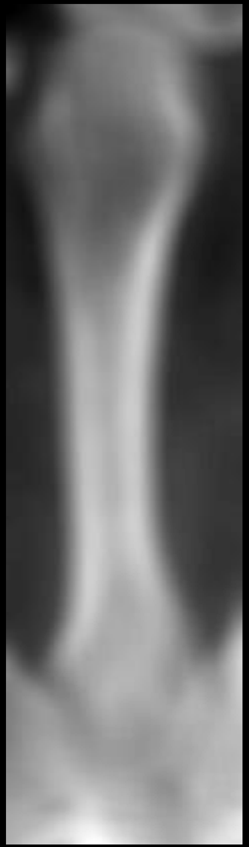


Prewitt

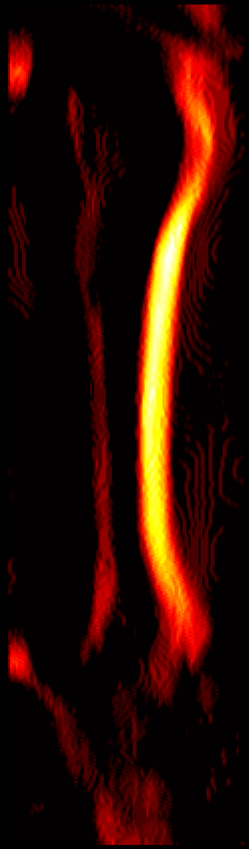


Prewitt

Hot colormap



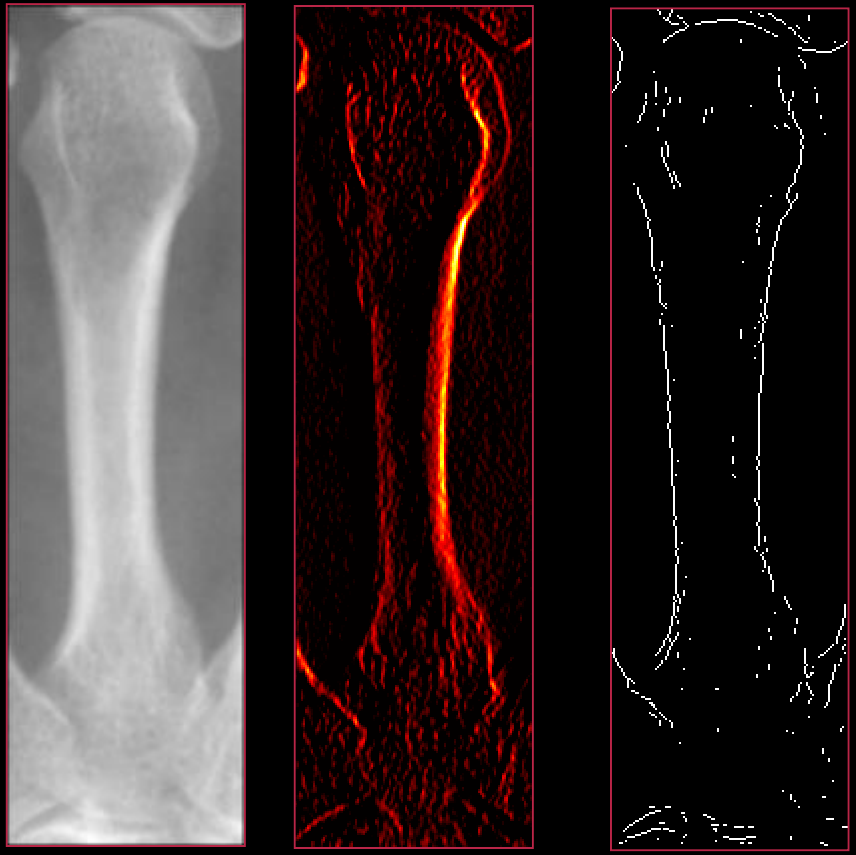
Smooth 15x15



Smooth 15x15

Prewitt

Edge detection



- Edge filter
  - Prewitt for example
- Thresholding
  - Separate edges from non-edges
- Output is binary image
  - Edges are white

Exercise data

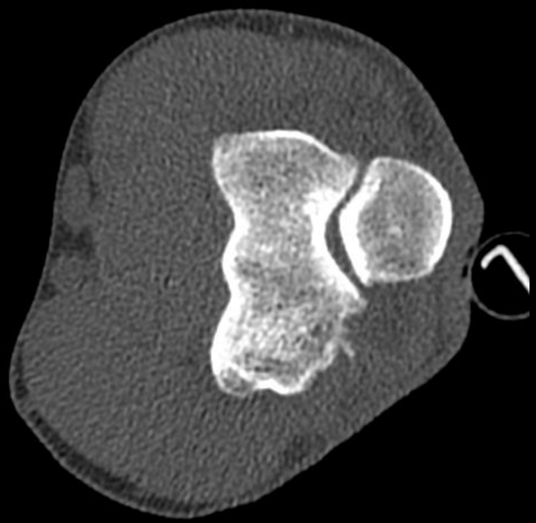


Image Analysis

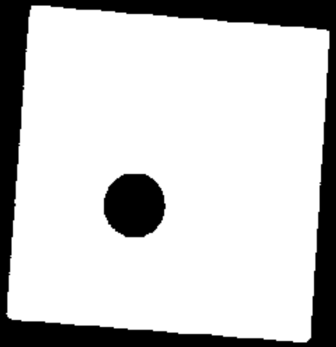
Rasmus R. Paulsen
Tim B. Dyrby
DTU Compute

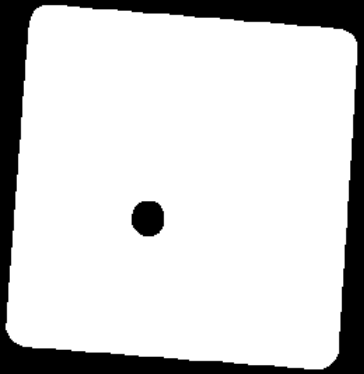
rapa@dtu.dk

http://www.compute.dtu.dk/courses/02502

Plenty of slides adapted from Thomas Moeslunds lectures

Lecture 4b – Morphology





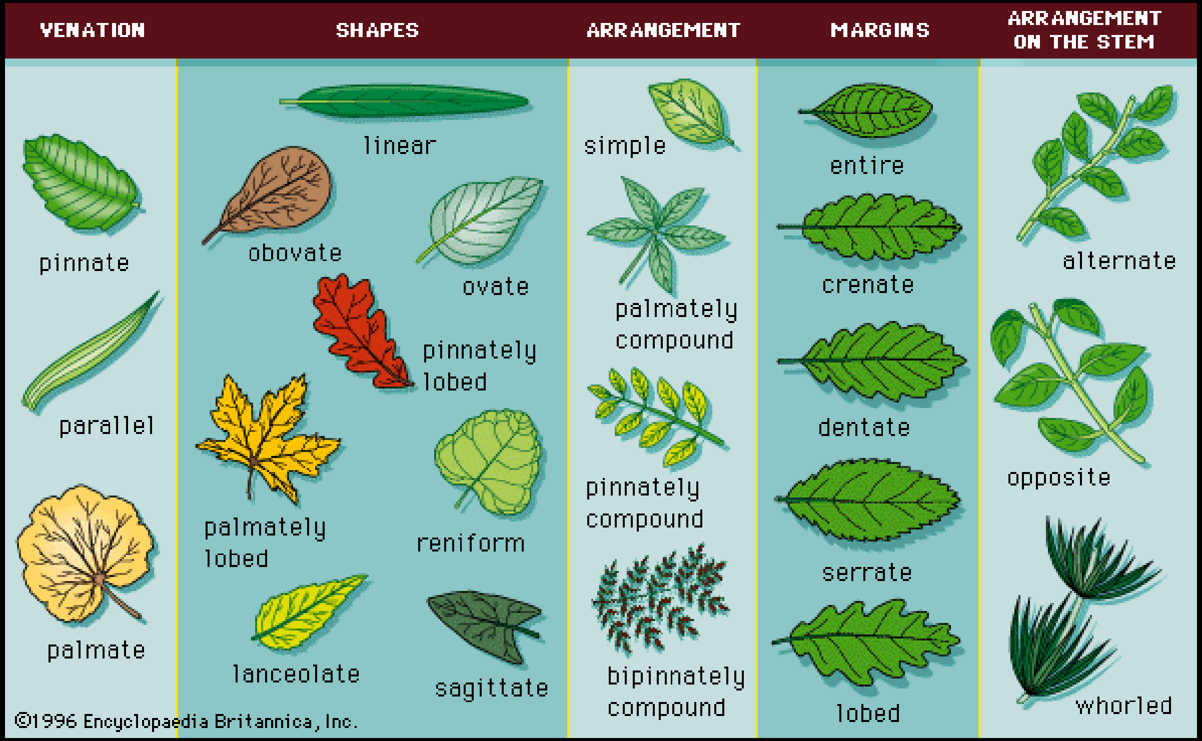
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0

What can you do after today?

- Describe the similarity between filtering and morphology
- Describe a structuring element
- Compute the dilation of a binary image
- Compute the erosion of a binary image
- Compute the opening of a binary image
- Compute the closing of a binary image
- Apply compound morphological operations to binary images
- Describe typical examples where morphology is suitable
- Remove unwanted elements from binary images using morphology
- Choose appropriate structuring elements and morphological operations based on image content

Morphology

- The science of *form, shape* and *structure*
- In biology: The form and structure of animals and plants



Common leaf morphologies

Mathematical morphology

Theorem 4.10

$$\begin{cases} \psi_m = \widetilde{\varphi}\,\widetilde{\gamma} = \widetilde{\gamma}\,\widetilde{\varphi}\,\widetilde{\gamma} = \psi\widetilde{\gamma} & , \\ \psi_M = \widetilde{\gamma}\widetilde{\varphi} = \widetilde{\varphi}\,\widetilde{\gamma}\,\widetilde{\varphi} = \psi\widetilde{\varphi} & , \\ \psi = \widetilde{\gamma}\,\psi = \widetilde{\varphi}\,\psi, & \\ \widetilde{\gamma} \le \psi_m \le \psi \le \psi_M \le \widetilde{\varphi} & . \end{cases}$$

The same theorem may be restated in another way. If $\mathcal{J}\mathrm{d}(\mathcal{B}) \neq \varnothing$ then let B_i be a family of elements of $\mathcal{B}$. We have $\vee\, B_i \in \sim B$, and thus $\widetilde{\gamma}(\vee\, B_i) = \vee B_i$. From the first relation above, it follows for any $\psi \in \mathcal{J}\mathrm{d}(\mathcal{B})$, that

$$\psi(\vee\, B_i) = \psi\widetilde{\gamma}(\vee\, B_i) = \widetilde{\varphi}\,\widetilde{\gamma}(\vee\, B_i)\,.$$

But $\widetilde{\gamma}(\vee\, B_i) = \vee\, B_i$, so that

$$\widetilde{\varphi}(\vee B_i) = \psi(\vee B_i) \in \mathcal{B}.$$

In the same way, we also obtain

$$\widetilde{\gamma}\widetilde{\varphi}(\wedge\, B_i) = \widetilde{\gamma}(\wedge\, B_i) = \psi(\wedge\, B_i) \in \mathcal{B}.$$

In other words, $\mathcal{B}$ is a *complete lattice* with respect to the ordering on $\mathcal{B}$ induced by $\le$, i.e. any family B_i in $\mathcal{B}$ has a smallest upper bound $\widetilde{\varphi}(\vee\, B_i)\,\mathcal{B}$ and a greatest lower bound $\widetilde{\gamma}(\wedge\, B_i) \in \mathcal{B}$.

Conversely, let us assume that $\mathcal{B}$ is a complete lattice. Thus, for any $A \in \mathcal{L}$, the family $\{B : B \in \mathcal{B}, B \ge A\}$ has in $\mathcal{B}$ a greatest lower bound, which is

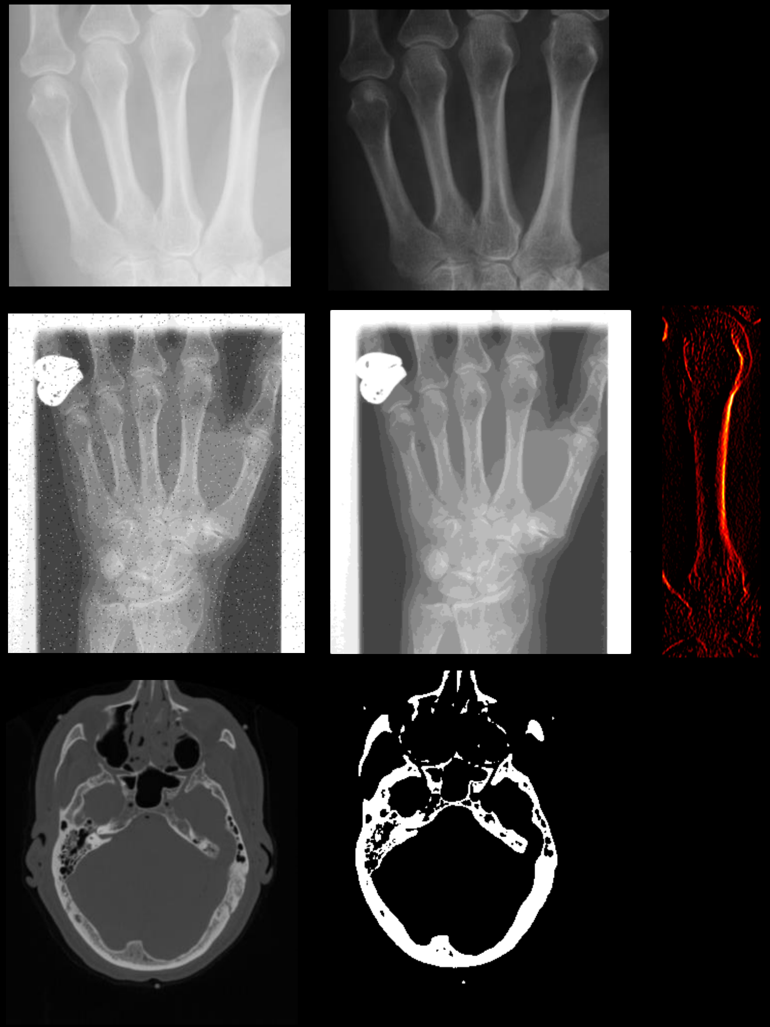
$$\widetilde{\gamma}(\wedge\{B : B \in \mathcal{B}, B \ge A\}) = \widetilde{\gamma}\,\widetilde{\varphi}(A) \in \mathcal{B}.$$

But this implies $\mathcal{B}_{\psi_M} \subseteq \mathcal{B}$ for the filter $\psi_M = \widetilde{\gamma}\,\widetilde{\varphi}$. Conversely, for any

- Developed in 1964
- Theoretical work done in Paris
- Used for classification of minerals in cut stone
- Initially used for binary images

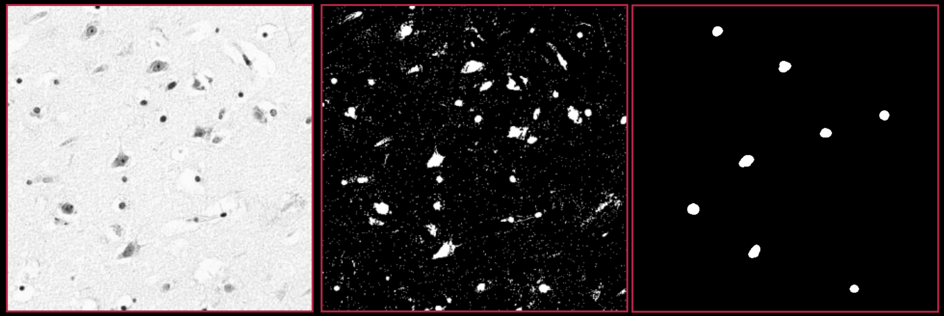
Do not worry! We use a much less theoretical approach!

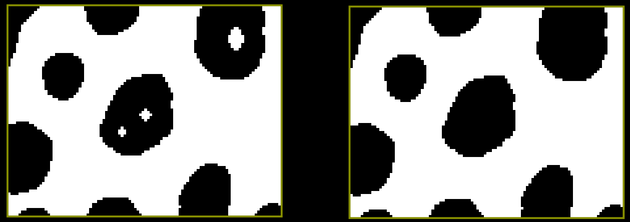
Relevance?

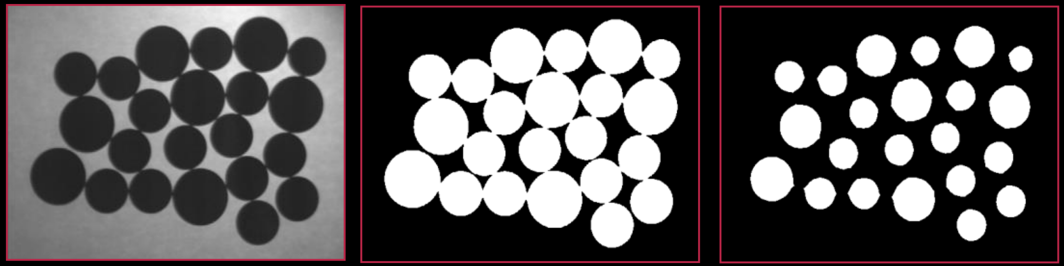


- Point wise operations
- Filtering
- Thresholding
  - Gives us objects that are separated by the background
- Morphology
  - Manipulate and enhance binary objects

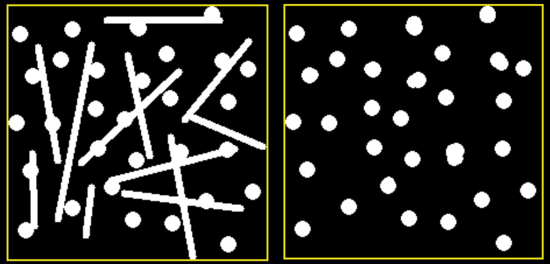
What can it be used for?



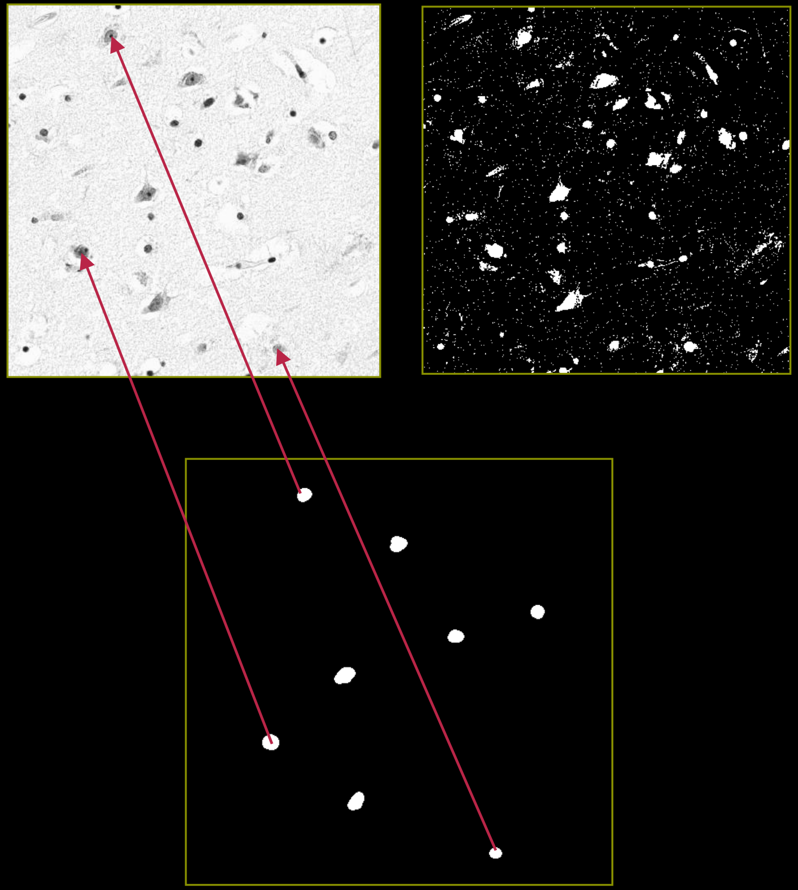




- Remove noise
  - Small objects
  - Fill holes
- Isolate objects
- Customized to specific shapes



How does it work?



- Grayscale image
- Preprocessing
  - Inversion
- Threshold => Binary image
- Morphology

Filtering and morphology

1	2	0	1	3	1
2	1	4	2	2	2
1	0	1	0	1	3
1	2	1	0	2	4
2	5	3	1	2	2
2	1	3	1	6	3

- Filtering
  - Gray level images
  - Kernel
  - Moves it over the input image
  - Creates a new output image

Filtering and morphology

0	1	0
1	1	1
0	1	0

Disk

1	1	1
1	1	1
1	1	1

Box

- Filtering
  - Gray level images
  - Kernel
  - Moves it over the input image
  - Creates a new output image
- Morphology
  - Binary images
  - Structuring element (SE)
  - Moves the SE over the input image
  - Creates a new binary output image

1D Morphology

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element (SE)

1	1	1

Output Image

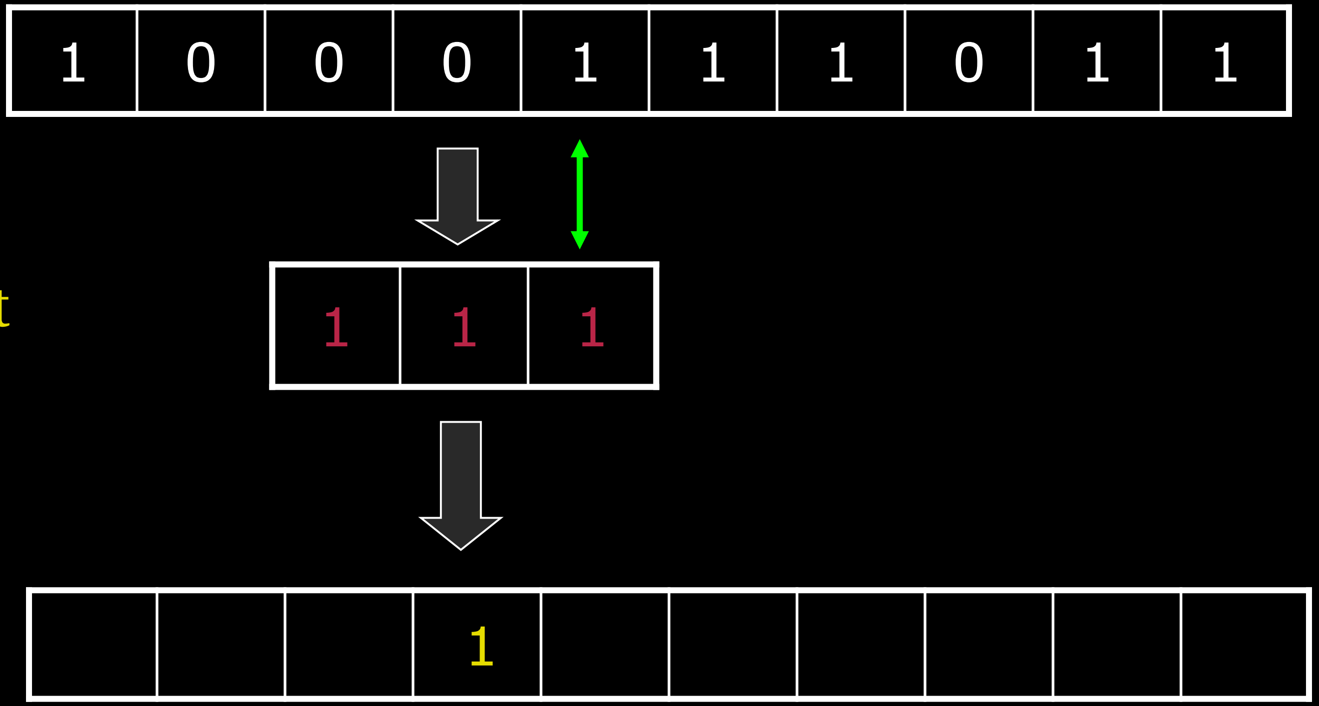
			?						

1D Morphology : The hit operation

Input image

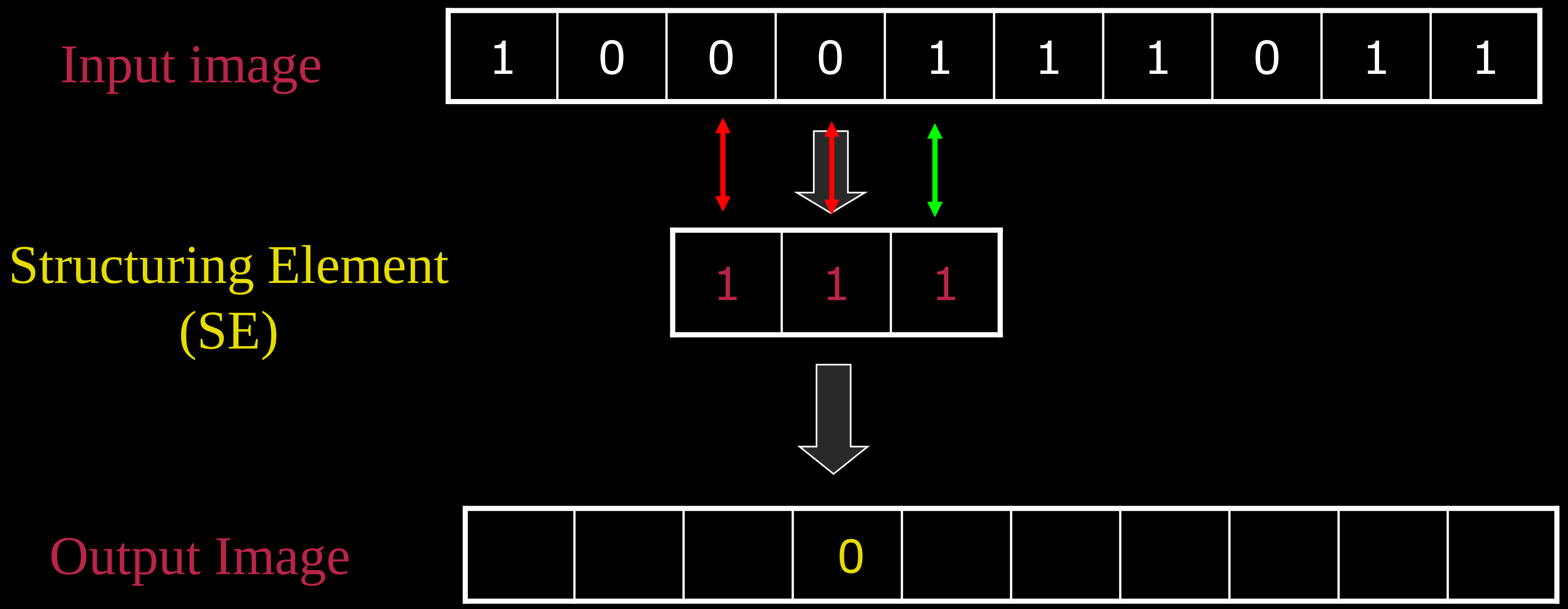
Structuring Element (SE)

Output Image



- If just one 1 in the SE match with the input
    - output 1
- else
    - output 0

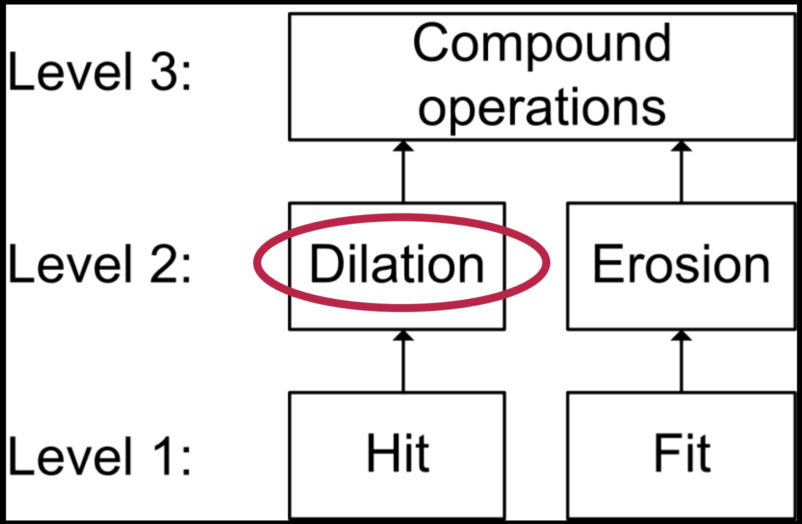
1D Morphology : The fit operation



- If all 1 in the SE match with the input
  - output 1
- else
  - output 0

1D Morphology : Dilation

- Dilate : To make wider or larger
  - Dansk : udvide
- Based on the *hit* operation



1D Dilation example

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

$$g(x) = f(x) \oplus SE$$

to make bigger

Output Image

	1								

Hit

- If just one 1 in the SE match with the input
    - output 1
- else
    - output 0

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0							

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1						

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1					

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1				

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1	1			

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1	1	1		

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

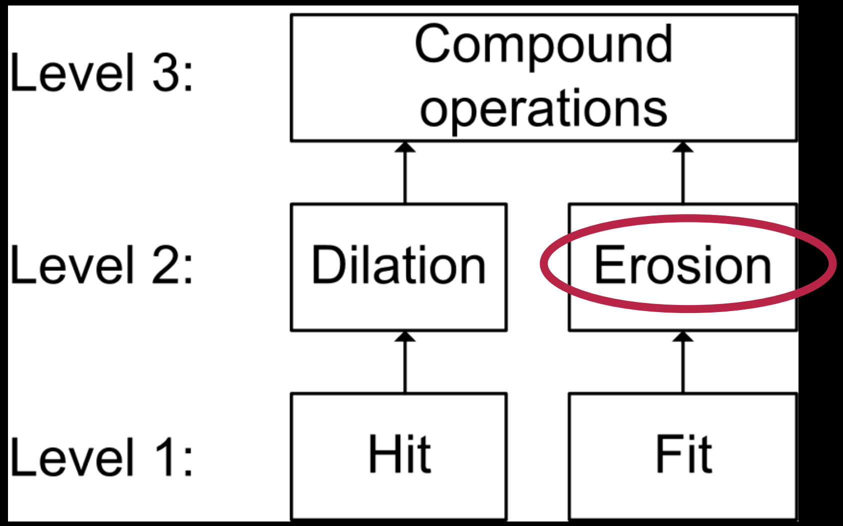
Output Image

	1	0	1	1	1	1	1	1	

The object gets bigger and holes are filled!

1D Morphology : Erosion

- Erode : To wear down (*Waves eroded the shore*)
  - Dansk : tære, gnave
- Based on the *fit* operation



Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

$$g(x) = f(x) \ominus SE$$

to make smaller

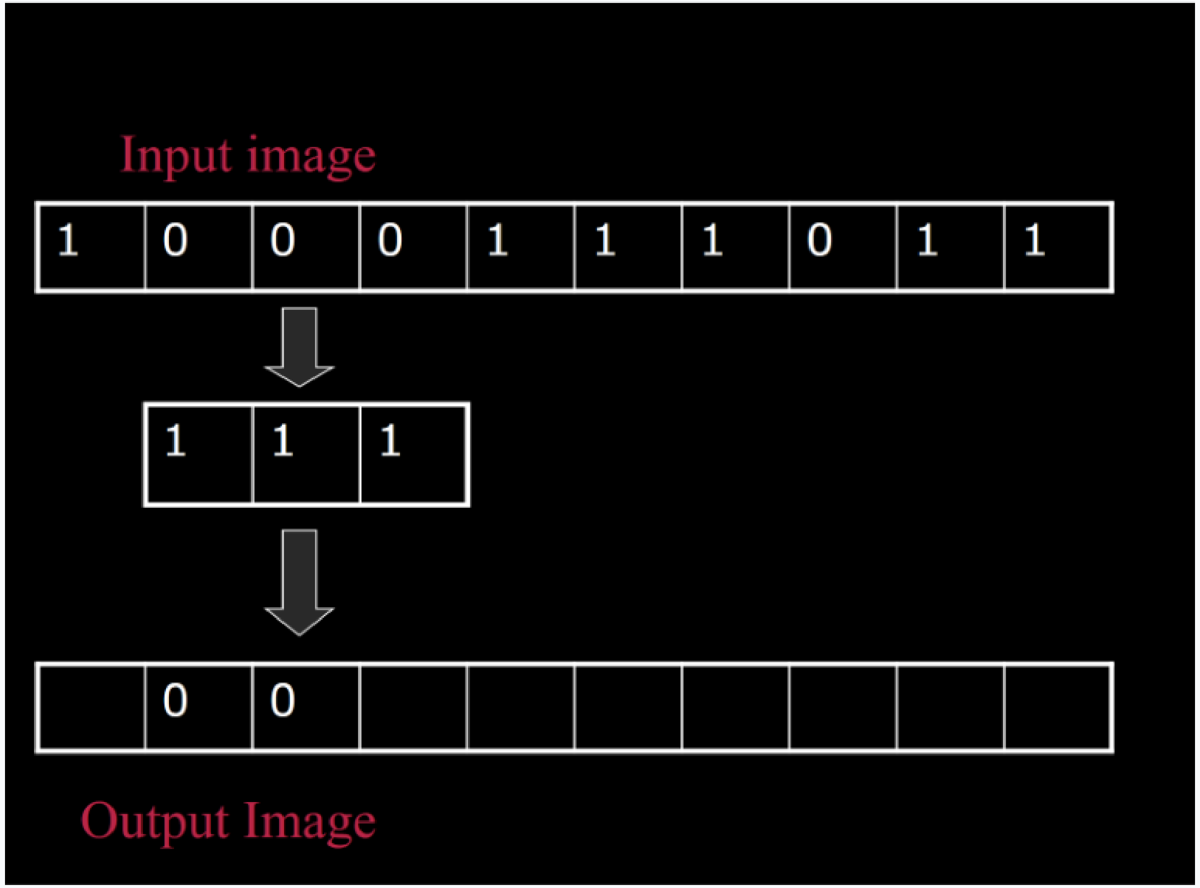
Output Image

	0								

Fit

- If all 1 in the SE match with the input
    - output 1
- else
    - output 0

1D Erosion



0 1 0 0 1 1 0 0

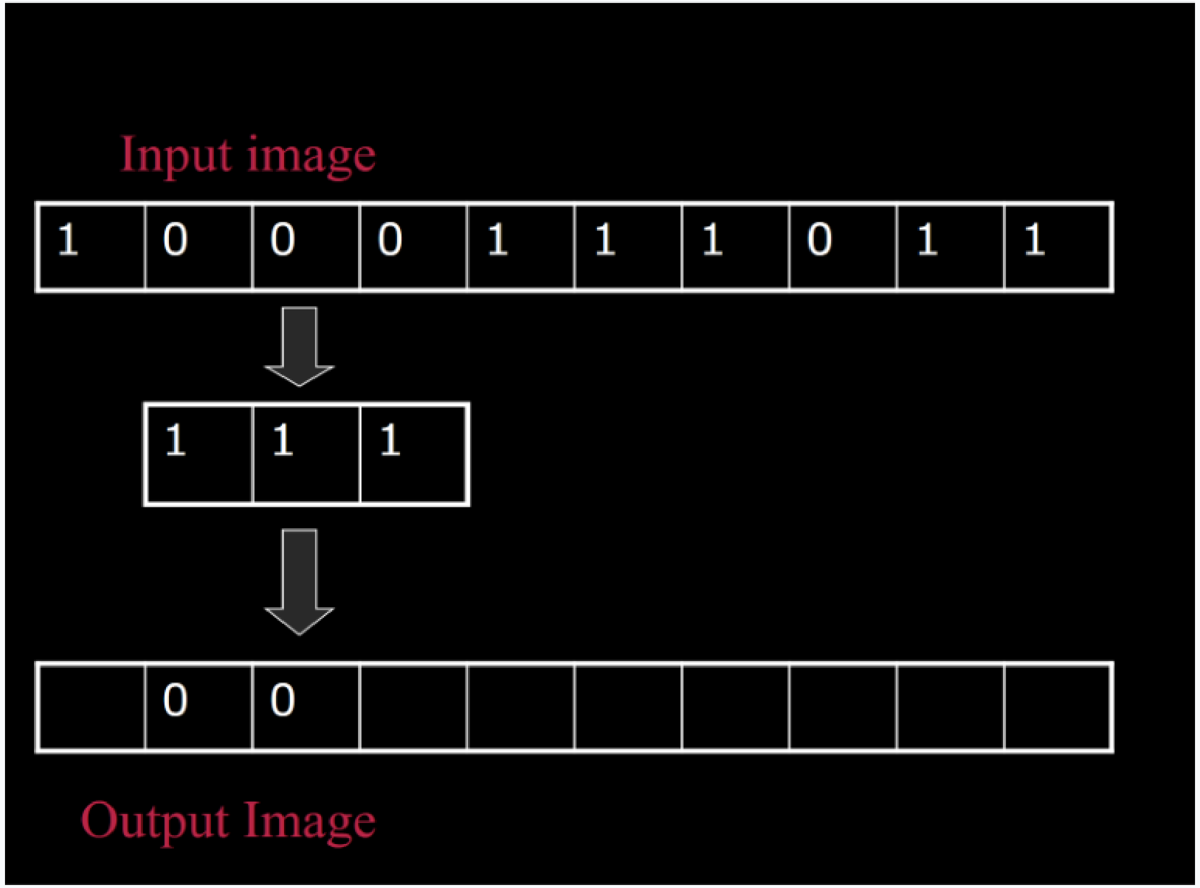
0 0 1 0 1 0 0 0

0 0 0 0 1 0 0 0

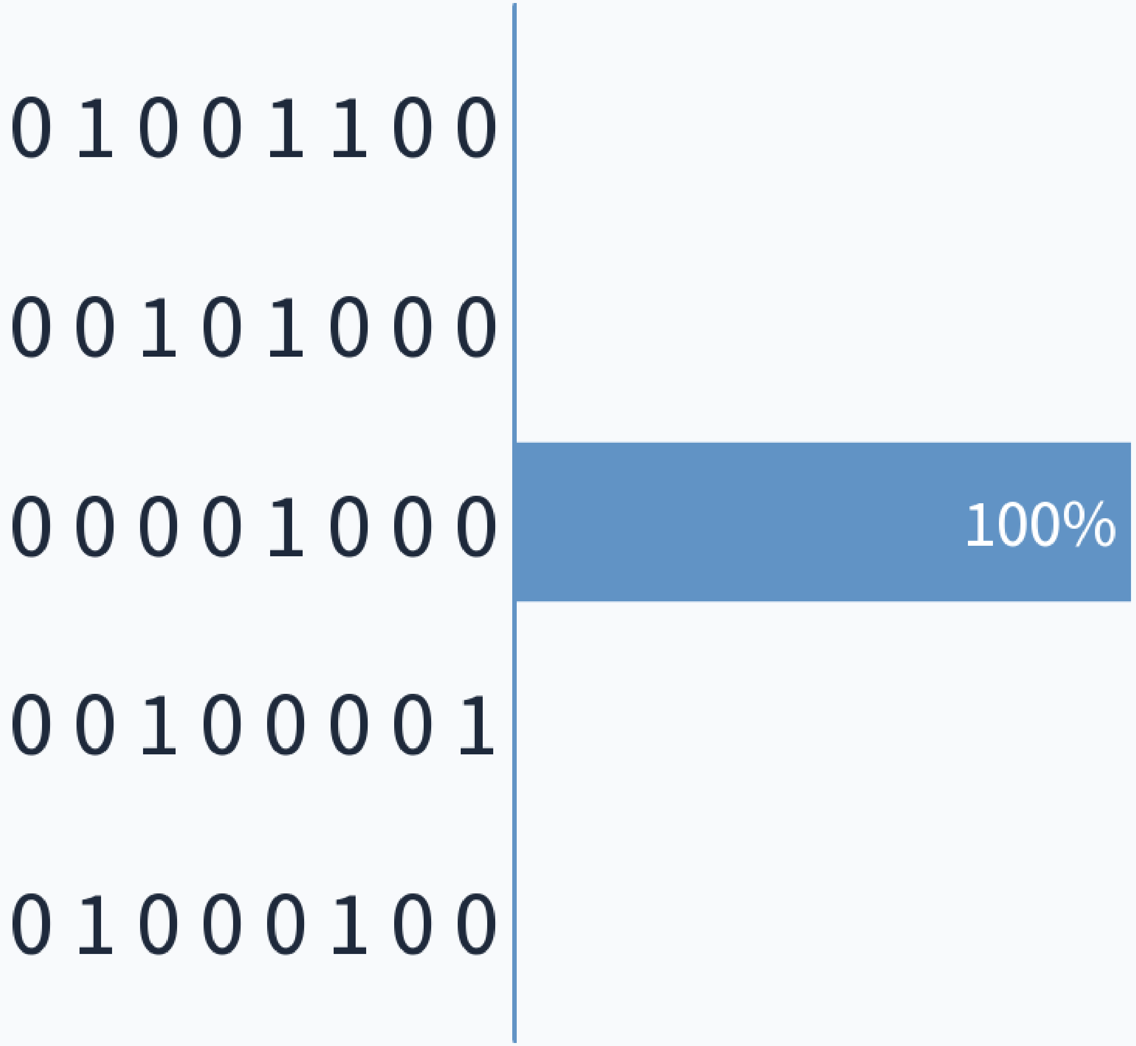
0 0 1 0 0 0 0 1

0 1 0 0 0 1 0 0

1D Erosion

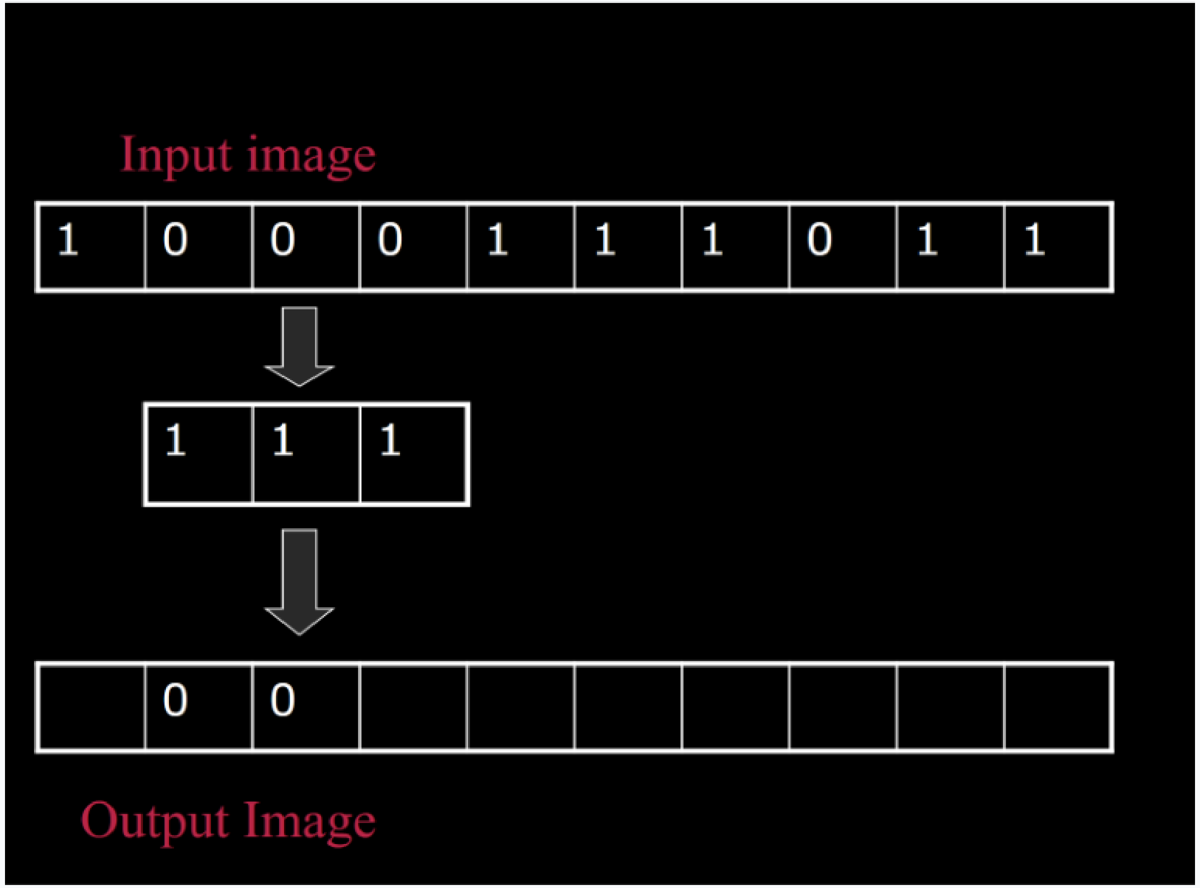


Input image
1 0 0 0 1 1 1 0 1 1
1 1 1
0 0
Output Image



0 1 0 0 1 1 0 0
0 0 1 0 1 0 0 0
0 0 0 0 1 0 0 0
100%
0 0 1 0 0 0 0 1
0 1 0 0 0 1 0 0

1D Erosion



- 0 1 0 0 1 1 0 0
- 0 0 1 0 1 0 0 0
- 0 0 0 0 1 0 0 0 — ✓ 100%
- 0 0 1 0 0 0 0 1
- 0 1 0 0 0 1 0 0

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0	0	1	0	0	0	

The object gets smaller

Structuring Element (Kernel)

0	1	0
1	1	1
0	1	0

Disk

1	1	1
1	1	1
1	1	1

Box

- Structuring Elements can have varying sizes
- Usually, element values are 0 or 1, but other values are possible (including none!)
- Structural Elements have an origin
- Empty spots in the Structuring Elements are *don't cares*!

		1	1	1		
	1	1	1	1	1	
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
	1	1	1	1	1	
		1	1	1		

Structuring Element Origin

0	1	0
1	1	1
0	1	0

1	1	1
1	1	1
1	1	1

- The origin is not always the center of the SE

Special structuring elements

- Structuring elements can be customized to a specific problem

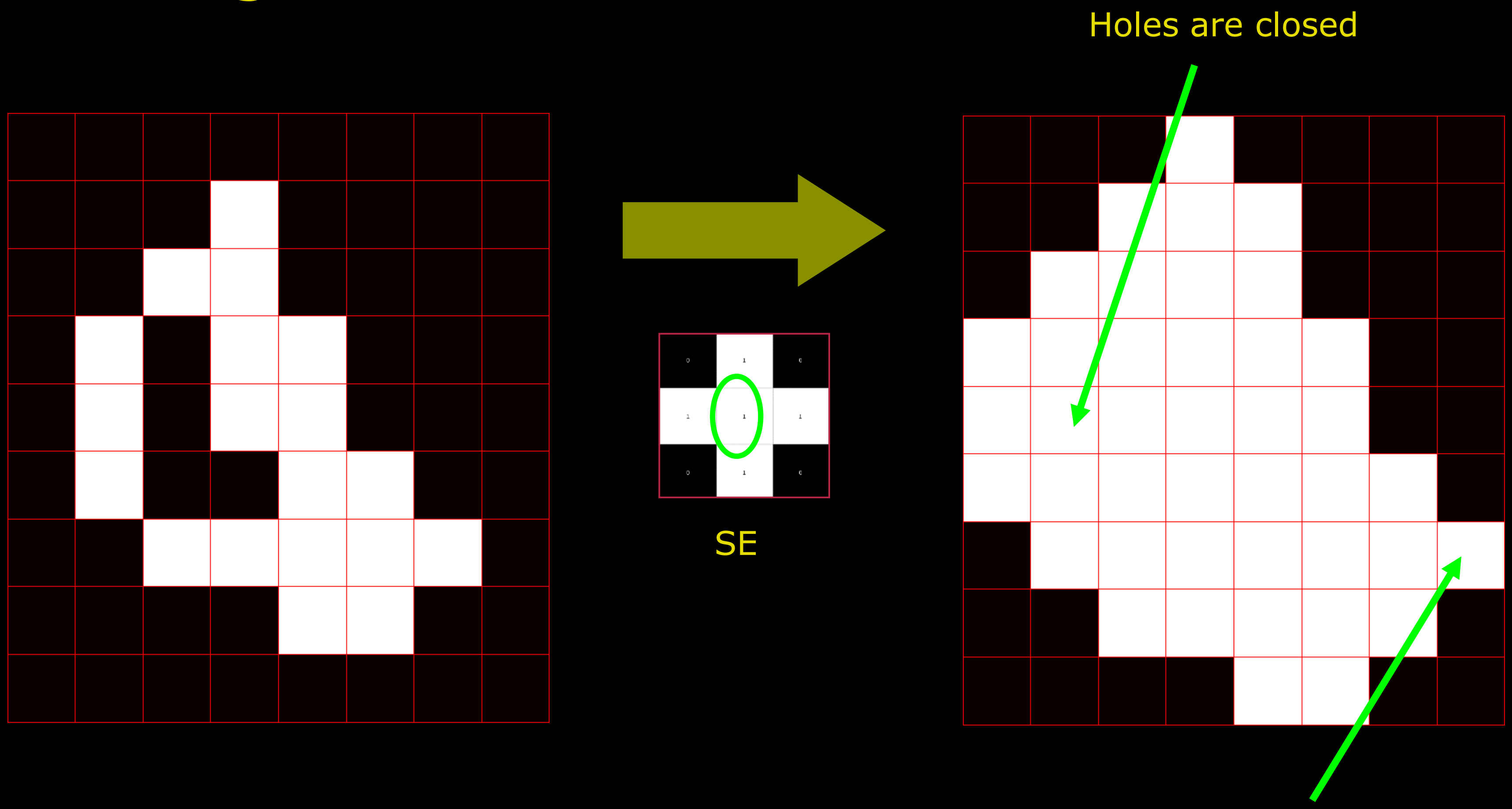
0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	0	1	0	0	0

Diamond

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

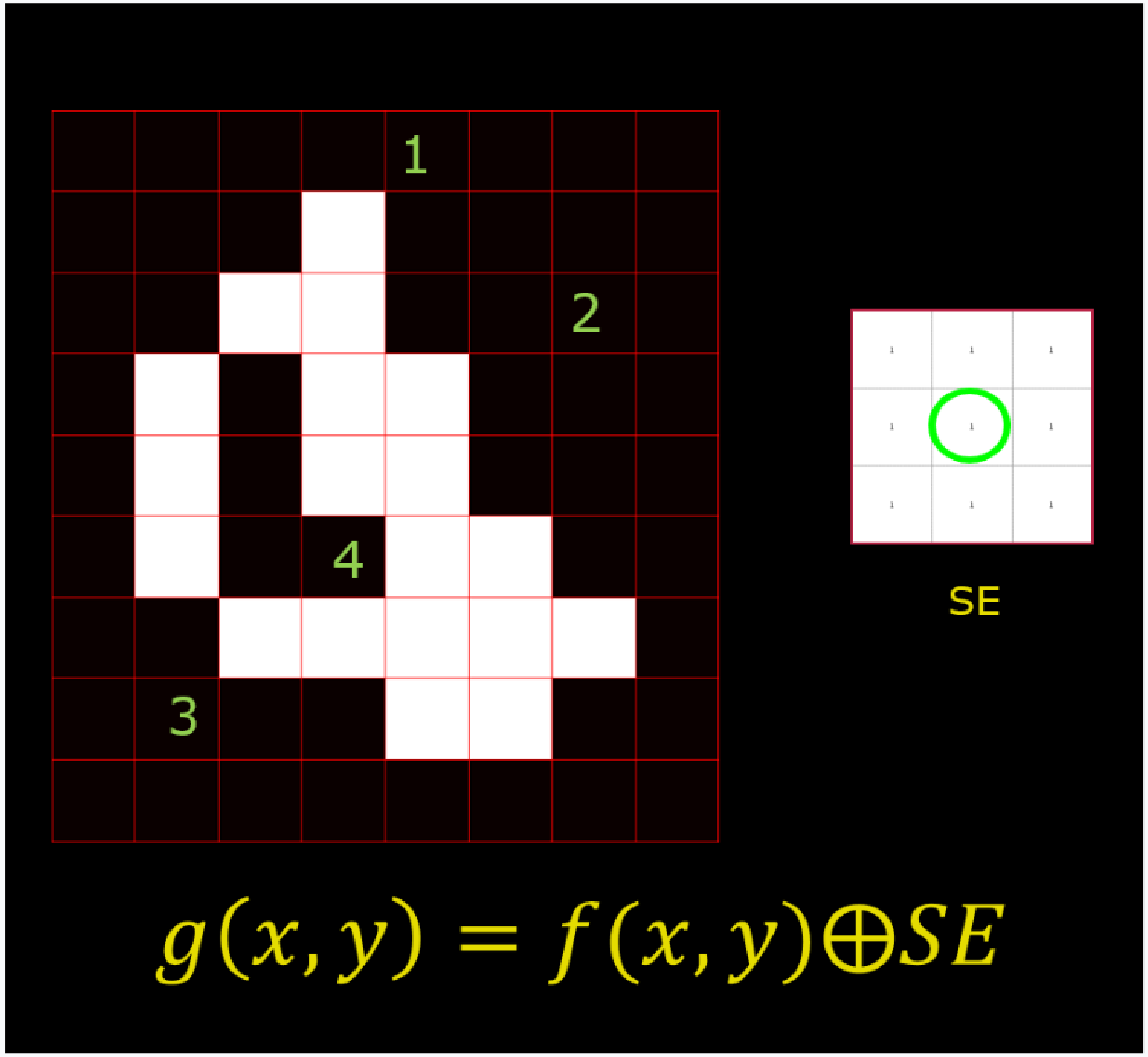
Line

Dilation on images - disk



$$g(x, y) = f(x, y) \oplus SE$$

Dilation on image - box



1 0 1 1

0 1 0 0

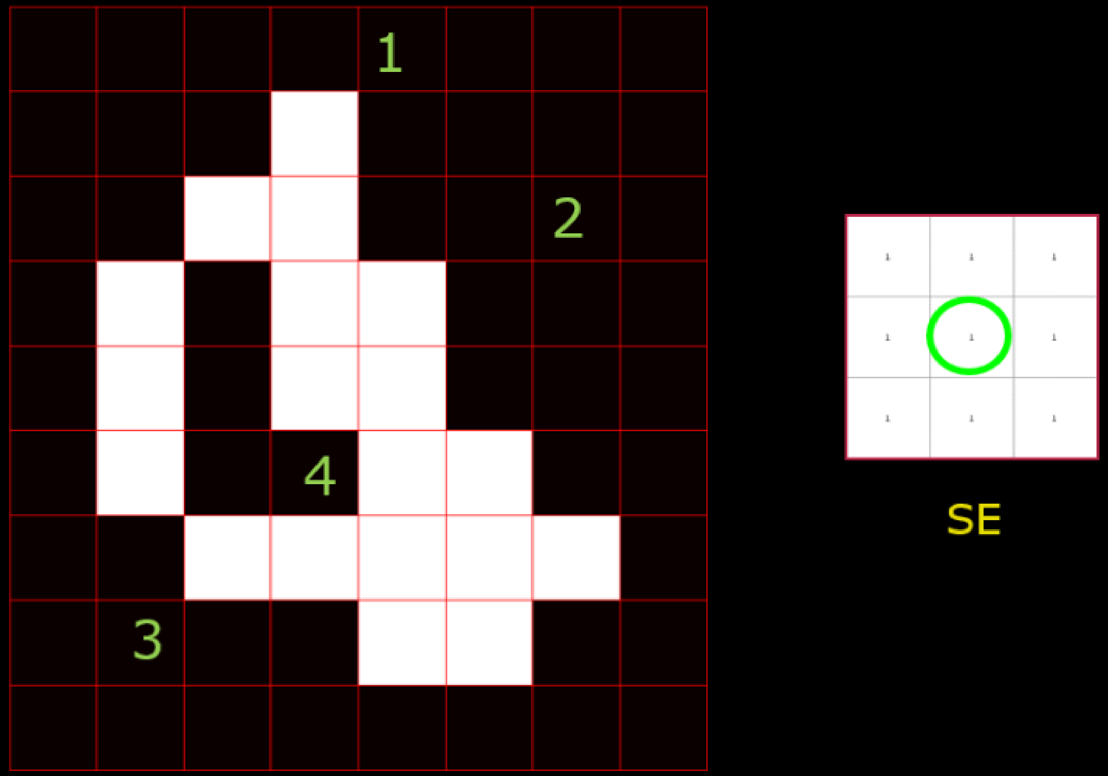
0 1 1 1

0 1 0 1

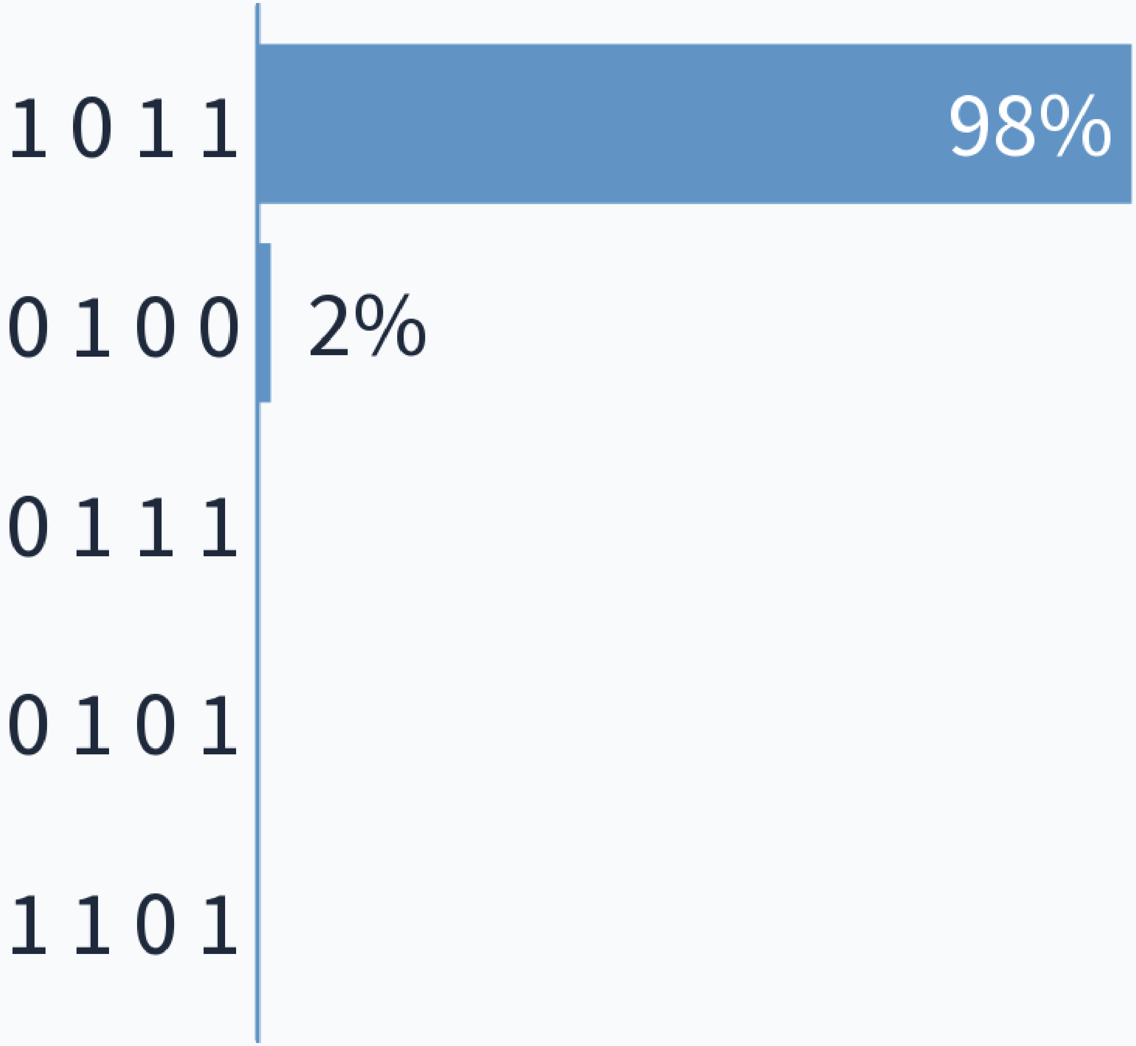
1 1 0 1

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Dilation on image - box

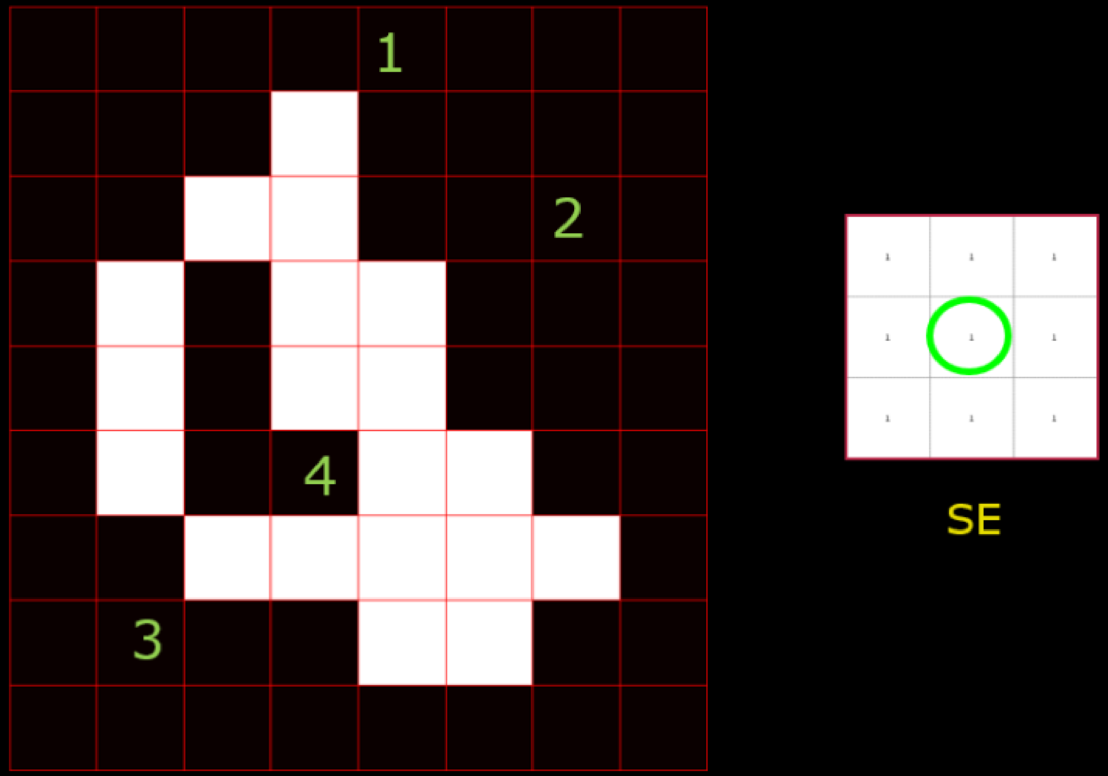


$$g(x,y) = f(x,y) \oplus SE$$

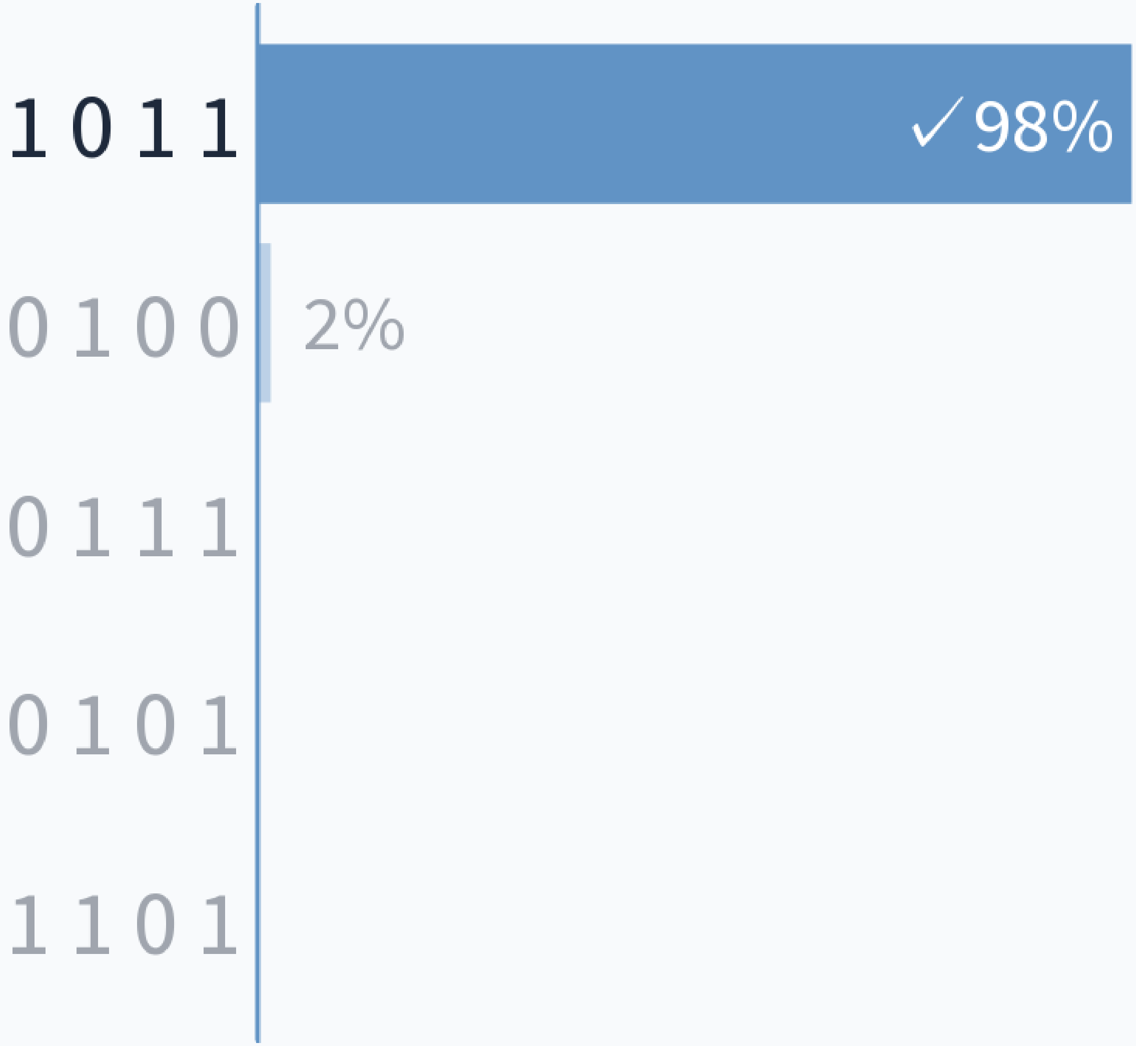


Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Dilation on image - box

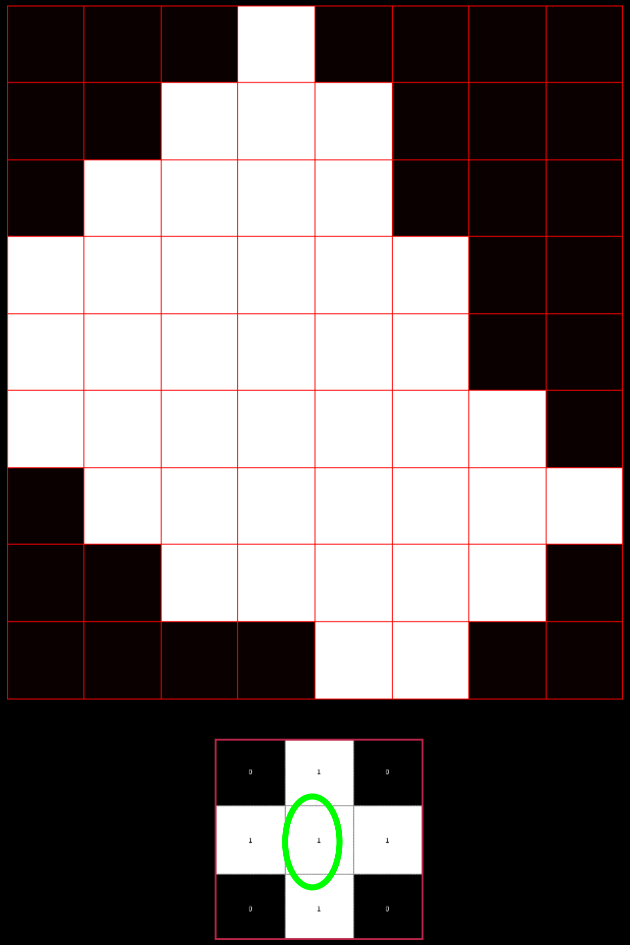


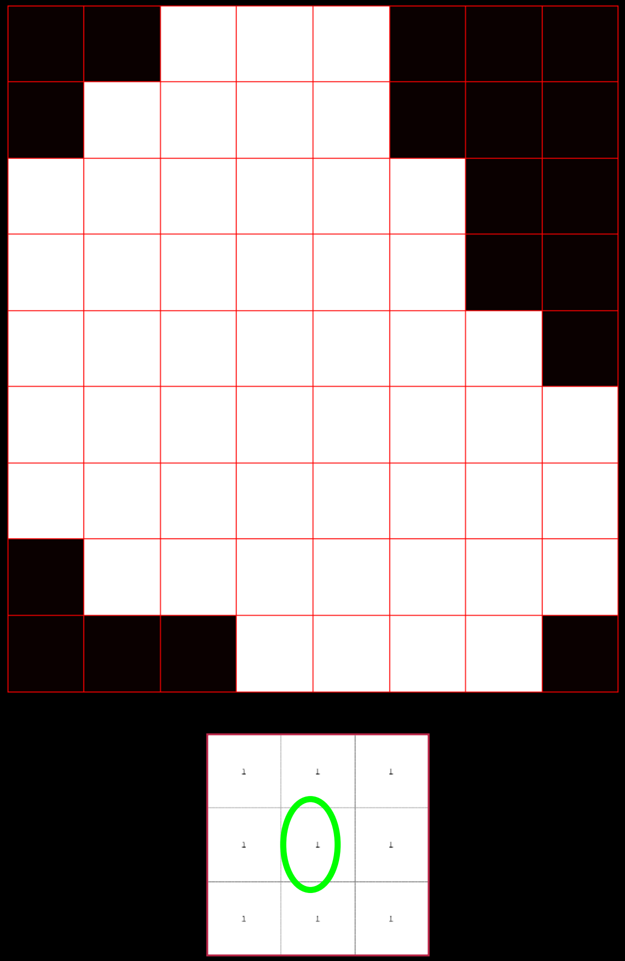
$$g(x,y) = f(x,y) \oplus SE$$



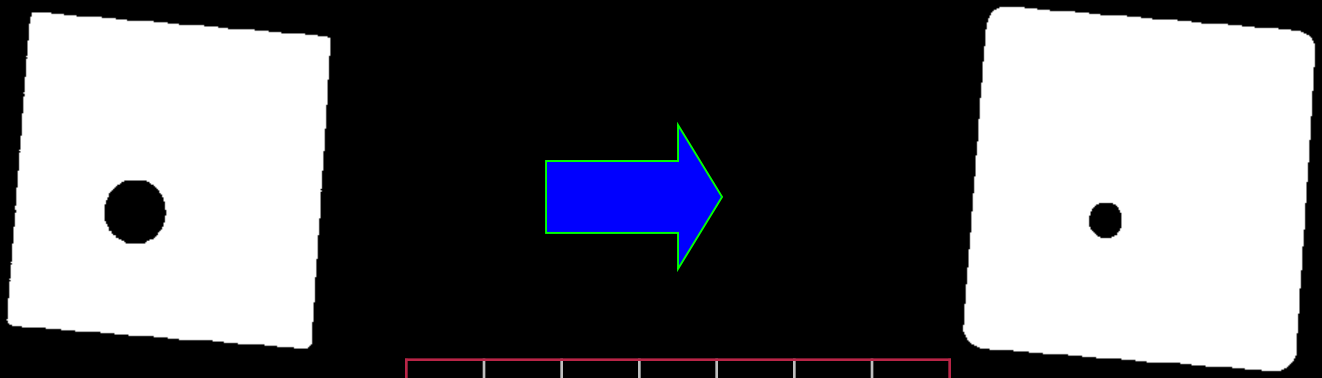
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Dilation – the effect of the SE





Dilation Example



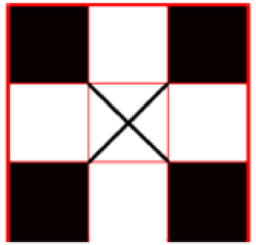
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0

- Round structuring element (disk)
- Creates round corners

Threshold and Dilation

A threshold of 200 is applied to the image and the result is a binary image. Now a dilation is performed with the structuring element below. How many foreground pixels are there in the resulting image?

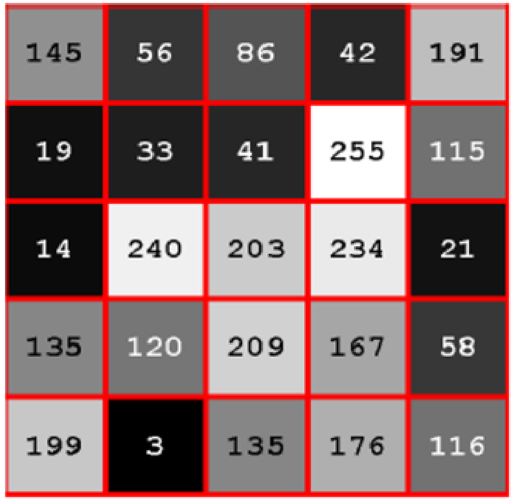
145	56	86	42	191
19	33	41	255	115
14	240	203	234	21
135	120	209	167	58
199	3	135	176	116

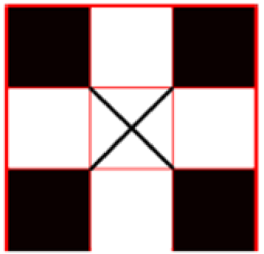


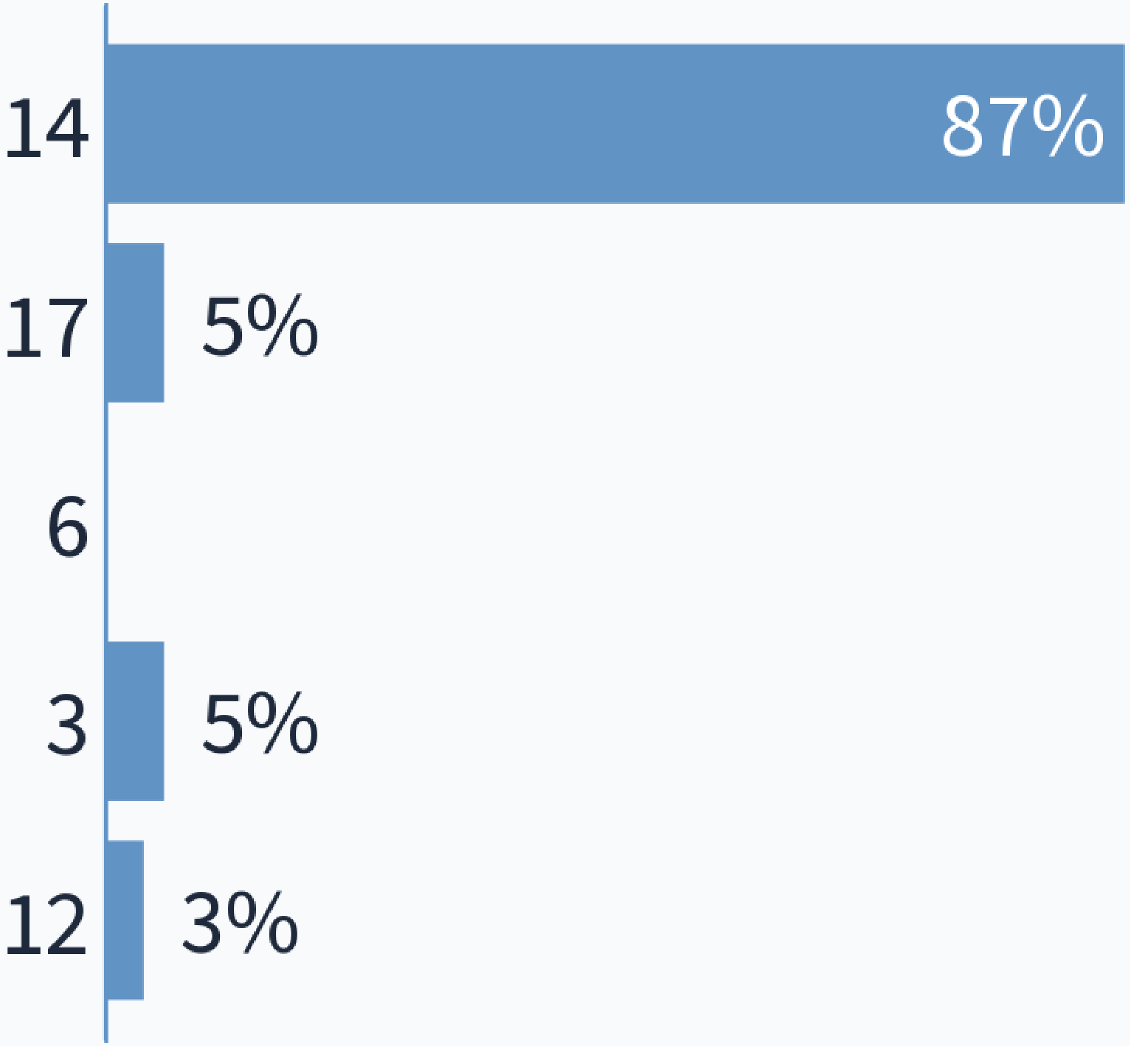
- 14
- 17
- 6
- 3
- 12

Threshold and Dilation

A threshold of 200 is applied to the image and the result is a binary image. Now a dilation is performed with the structuring element below. How many foreground pixels are there in the resulting image?





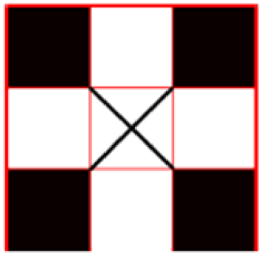


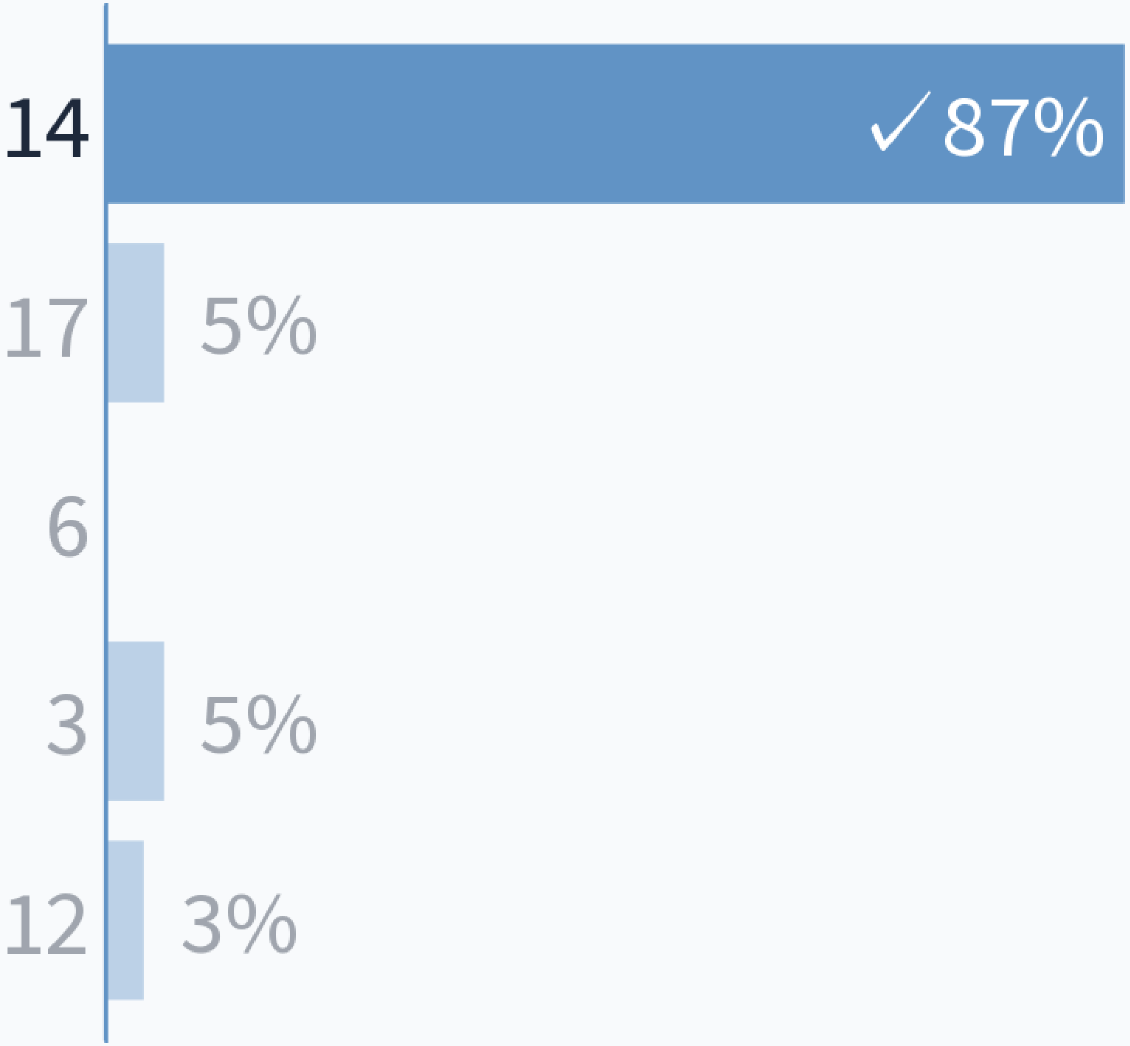
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Threshold and Dilation

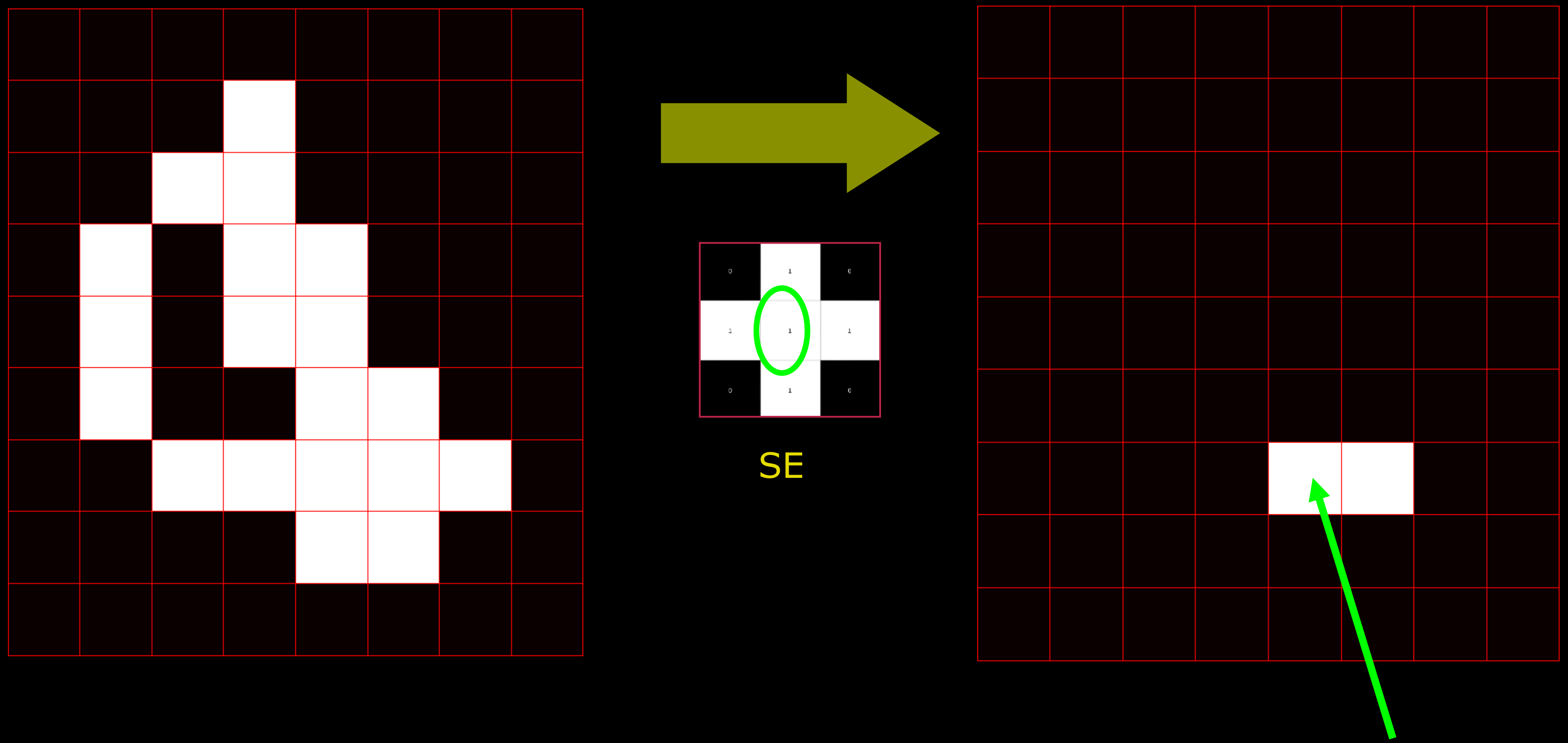
A threshold of 200 is applied to the image and the result is a binary image. Now a dilation is performed with the structuring element below. How many foreground pixels are there in the resulting image?

145	56	86	42	191
19	33	41	255	115
14	240	203	234	21
135	120	209	167	58
199	3	135	176	116



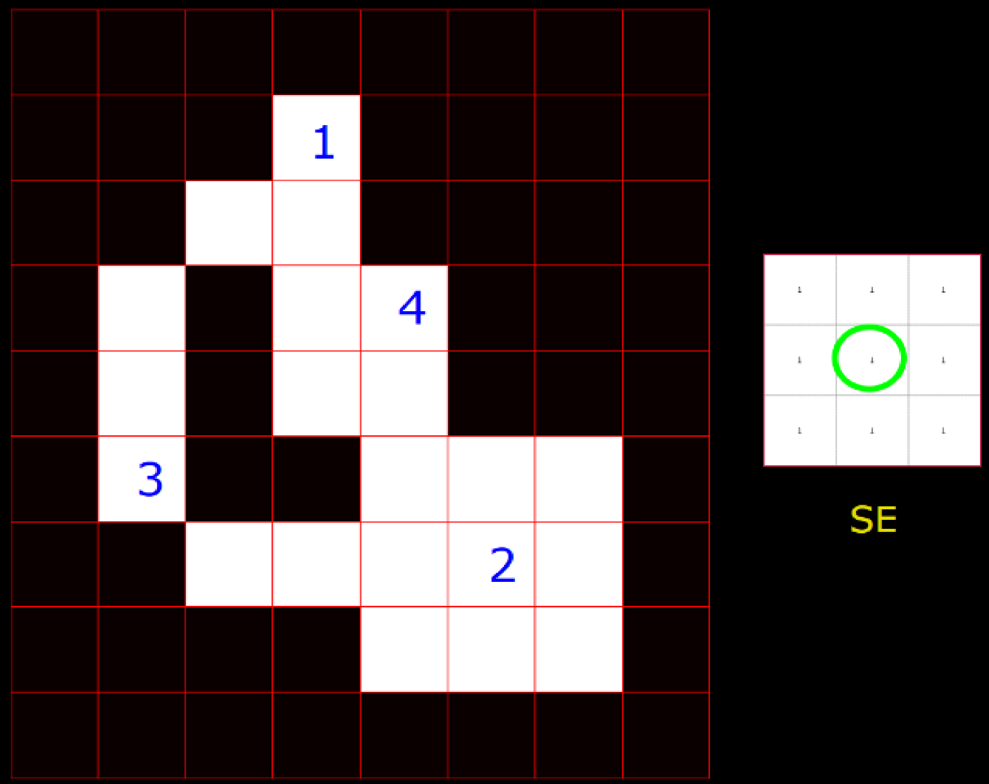


Erosion on images - disk



$$g(x,y) = f(x,y) \ominus SE$$

Erosion on images



$$g(x,y) = f(x,y) \ominus SE$$

0 0 1 0

1 0 1 0

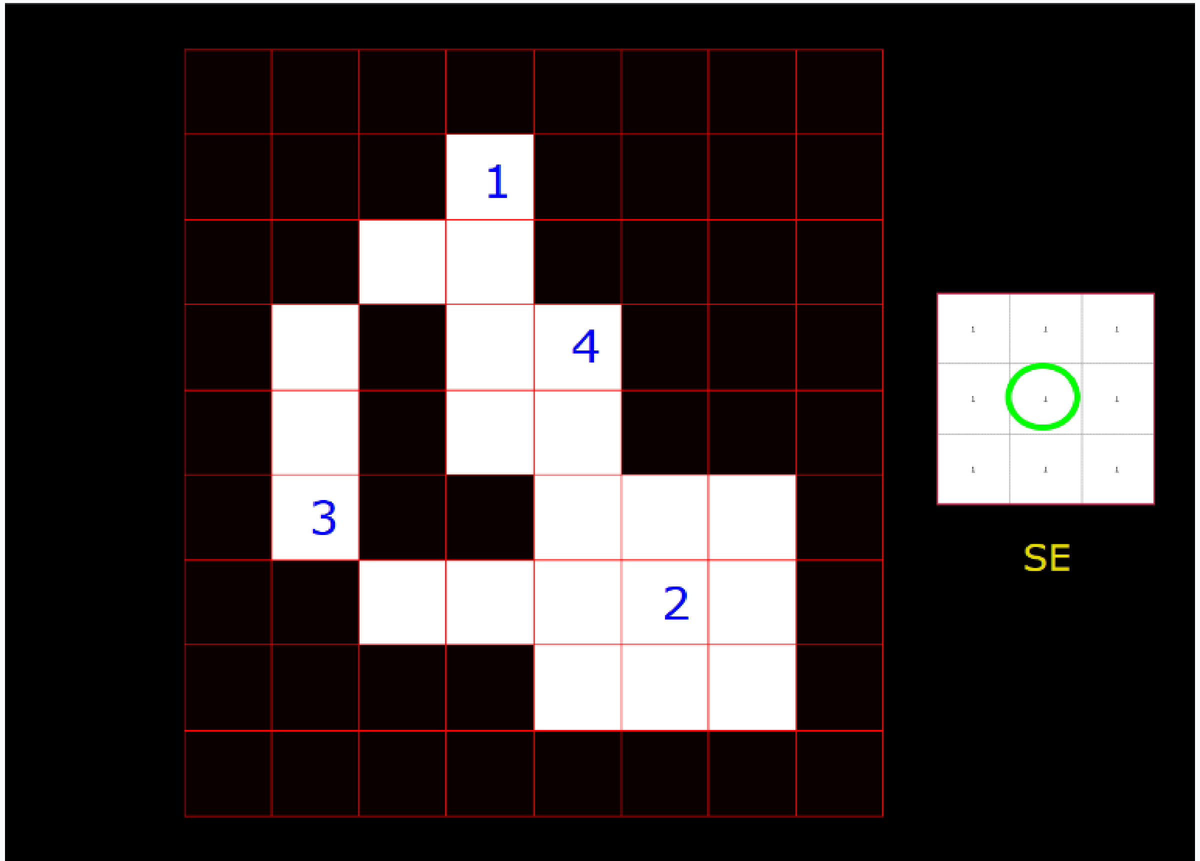
0 1 1 0

0 1 0 0

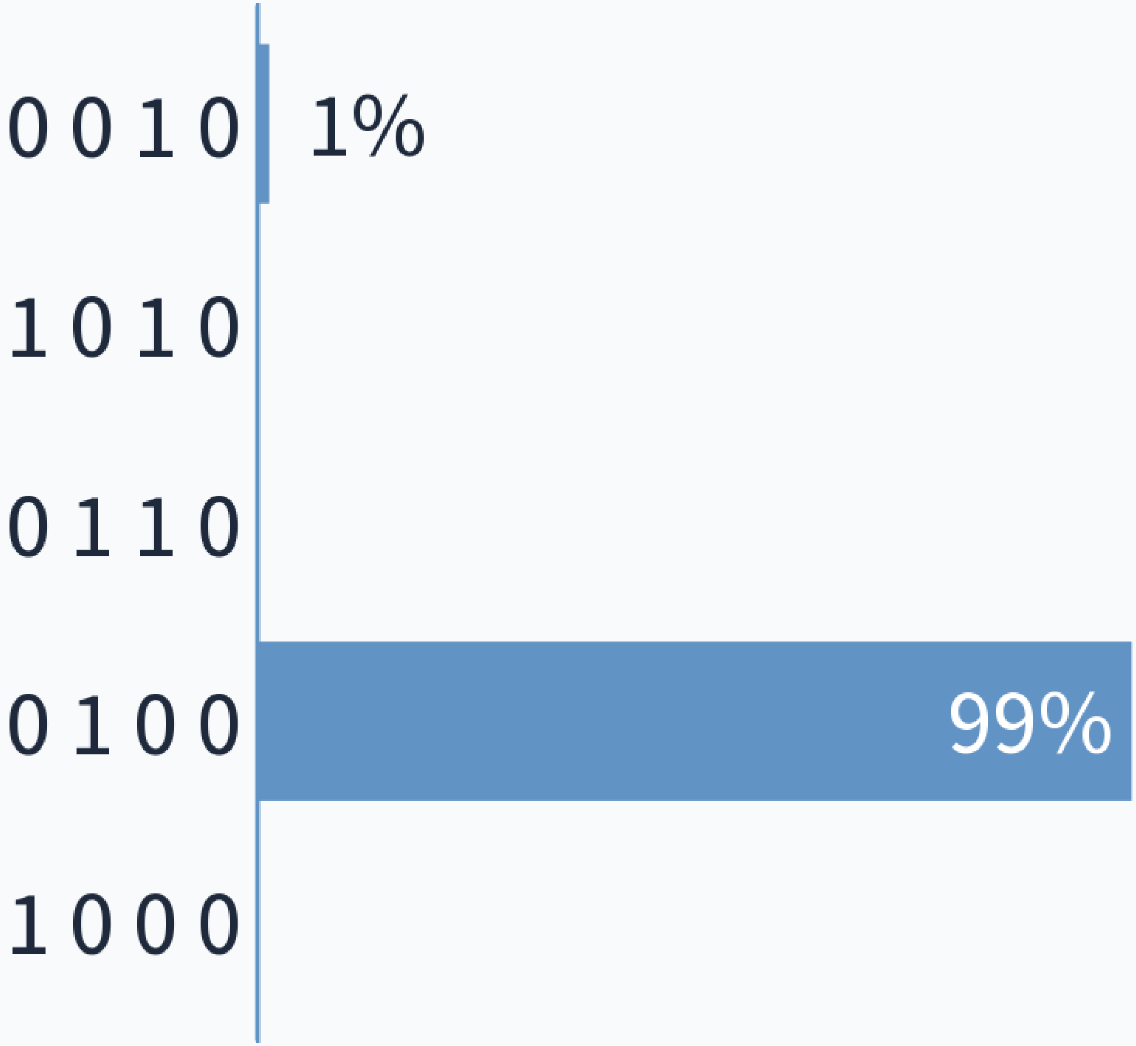
1 0 0 0

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

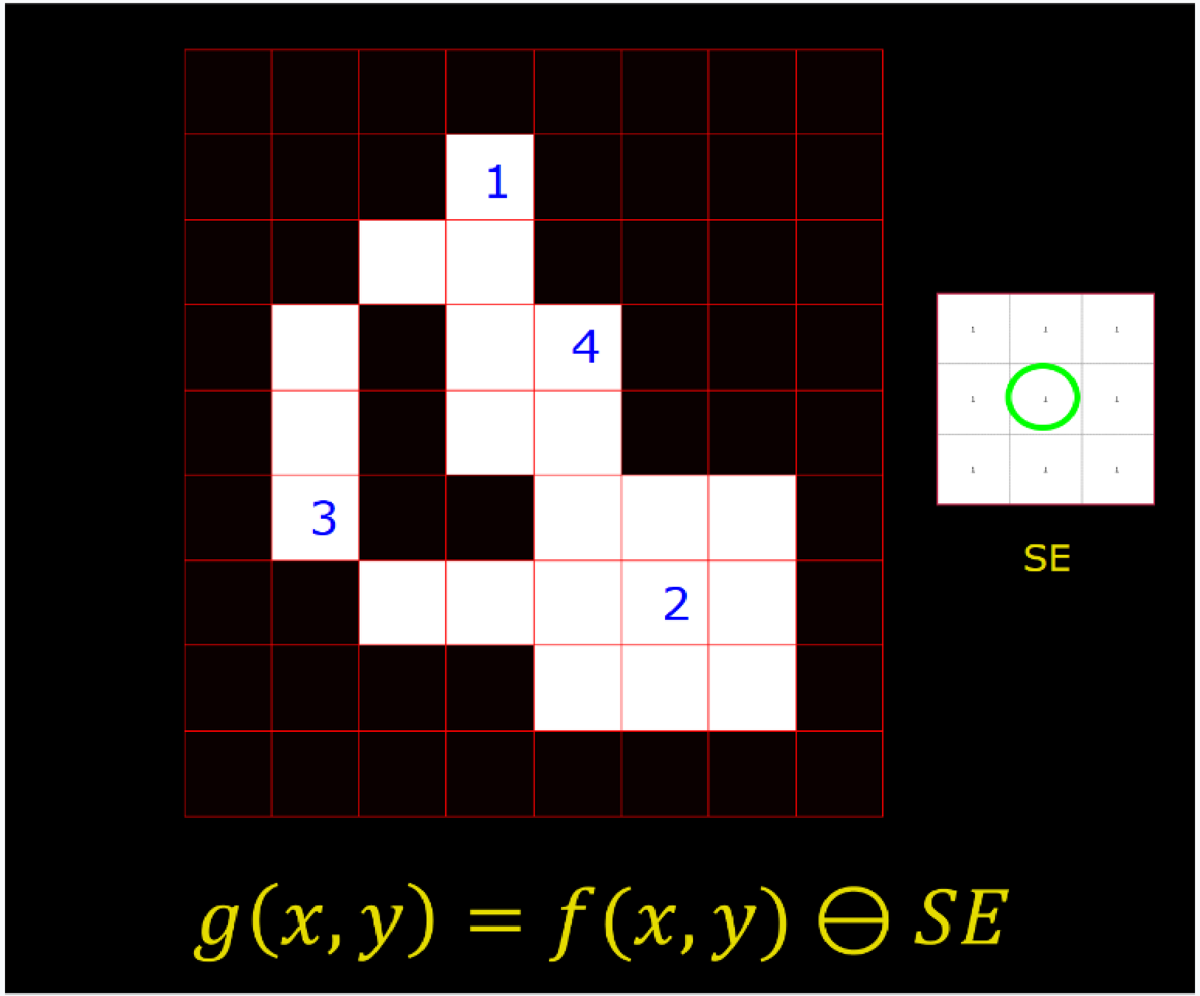
Erosion on images

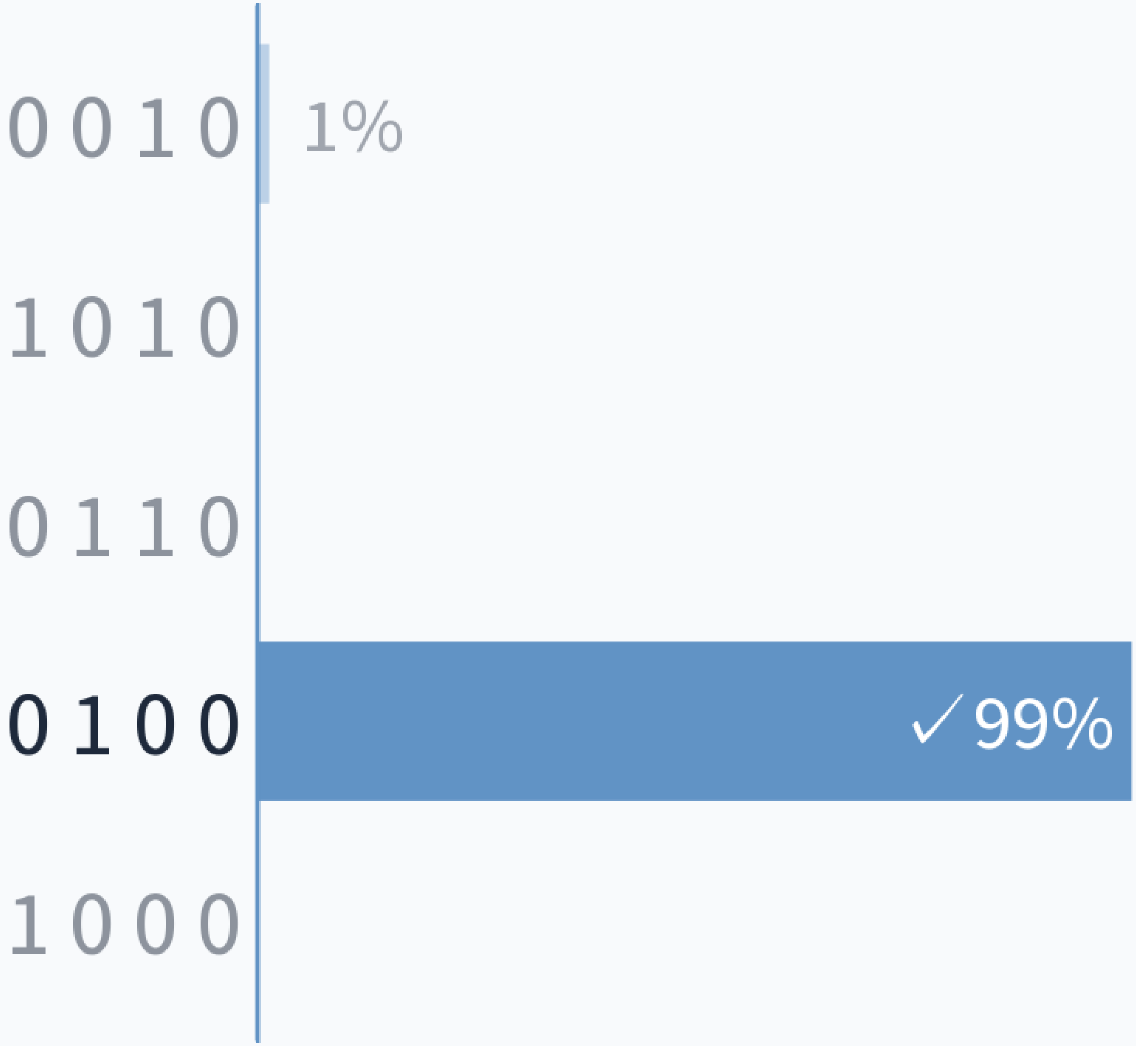


$$g(x,y) = f(x,y) \ominus SE$$



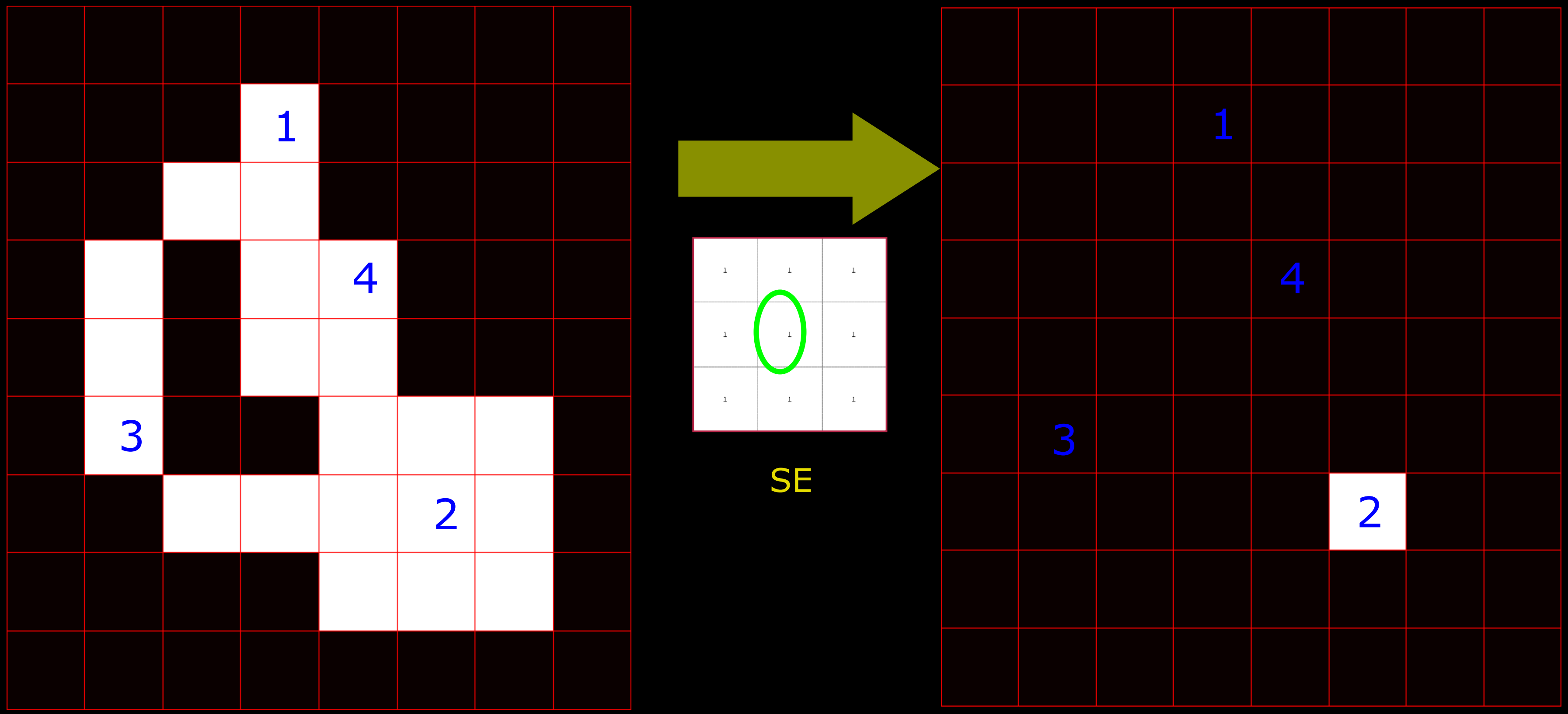
Erosion on images





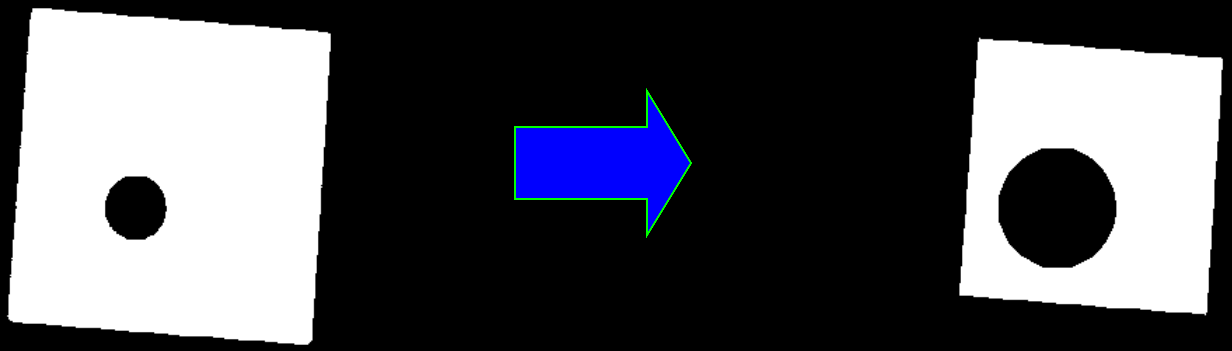
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Erosion on images – box (square)



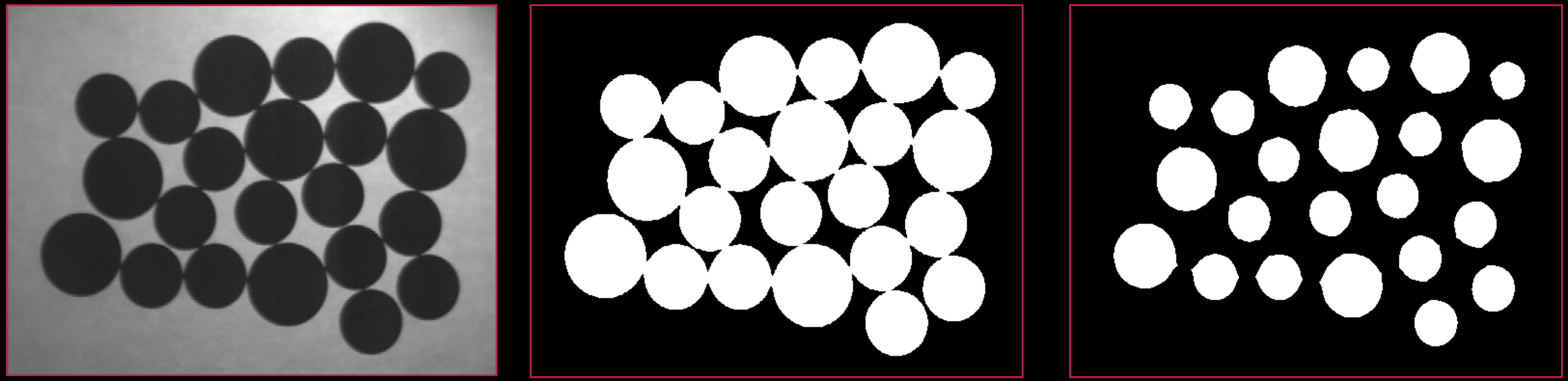
$$g(x, y) = f(x, y) \ominus SE$$

Erosion example

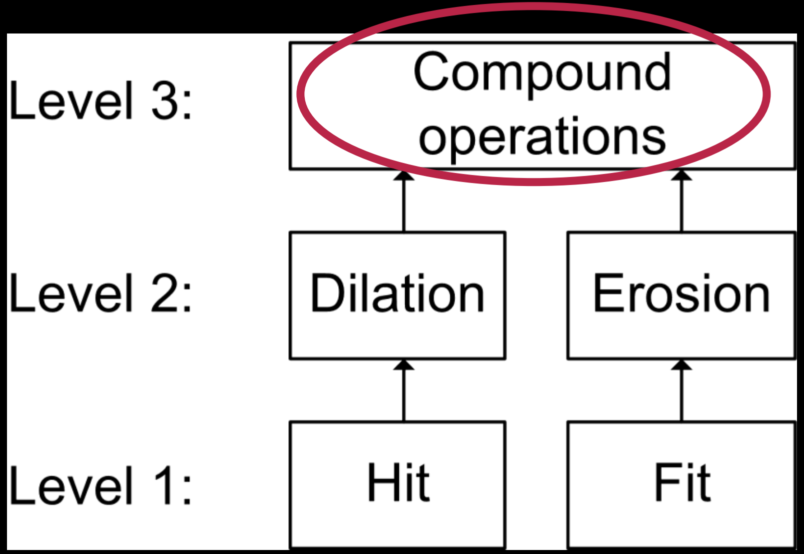


Counting Coins

- Counting these coins is difficult because they touch each other!
- Solution: Threshold and Erosion separates them!
- More on counting next time!



Compound operations

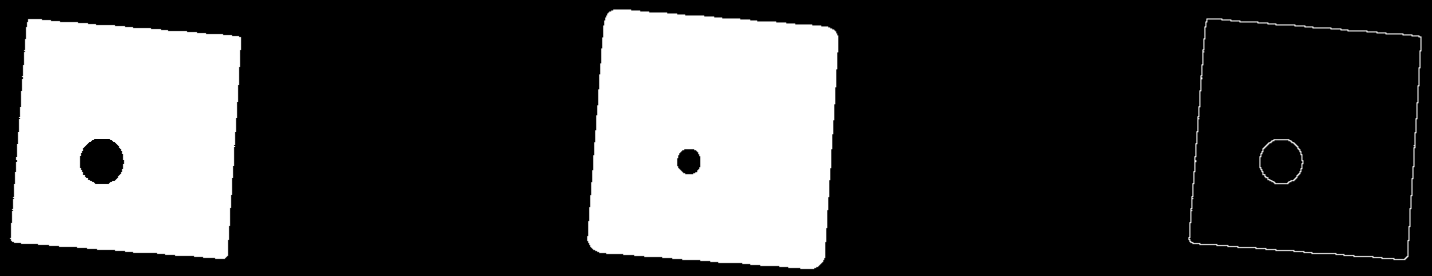


- Compound
  - *made of two or more separate parts or elements*
- Combining Erosion and Dilation into more advanced operations
  - Finding the outline
  - Opening
    - Isolate objects and remove small objects (better than Erosion)
  - Closing
    - Fill holes (better than Dilation)

Finding the outline

1. Dilate input image (object gets bigger)
2. Subtract input image from dilated image
3. The outline remains!

$$g(x,y) = (f(x,y) \oplus SE) - f(x,y)$$



Opening

- Motivation: Remove small objects BUT keep original size (and shape)
- Opening = Erosion + Dilation
  - Use the same structuring element!
  - Similar to erosion but less destructive
- Math:

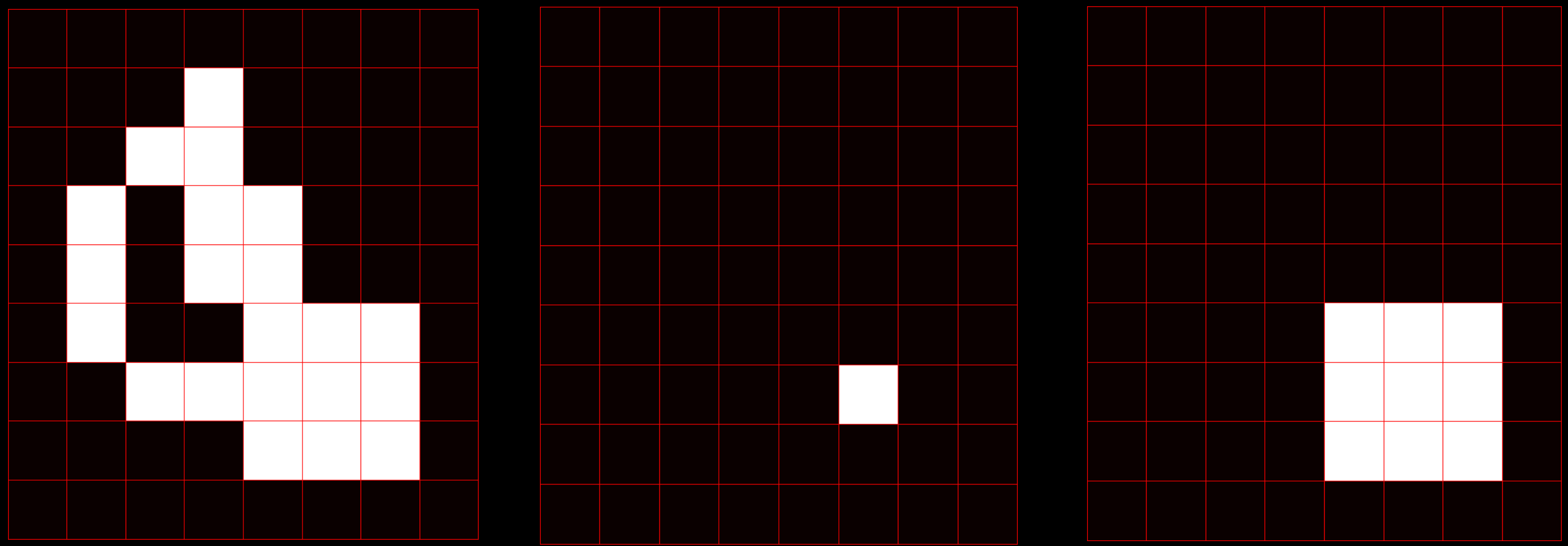
$$g(x,y) = f(x,y) \circ SE = (f(x,y) \ominus SE) \oplus SE$$

- Opening is **idempotent:** Repeated operations has no further effects!

$$f(x,y) \circ SE = (f(x,y) \circ SE) \circ SE$$

Opening

$$g(x,y) = (f(x,y) \ominus SE) \oplus SE$$

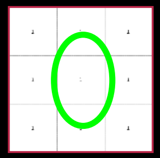


Original

Eroded

Dilated

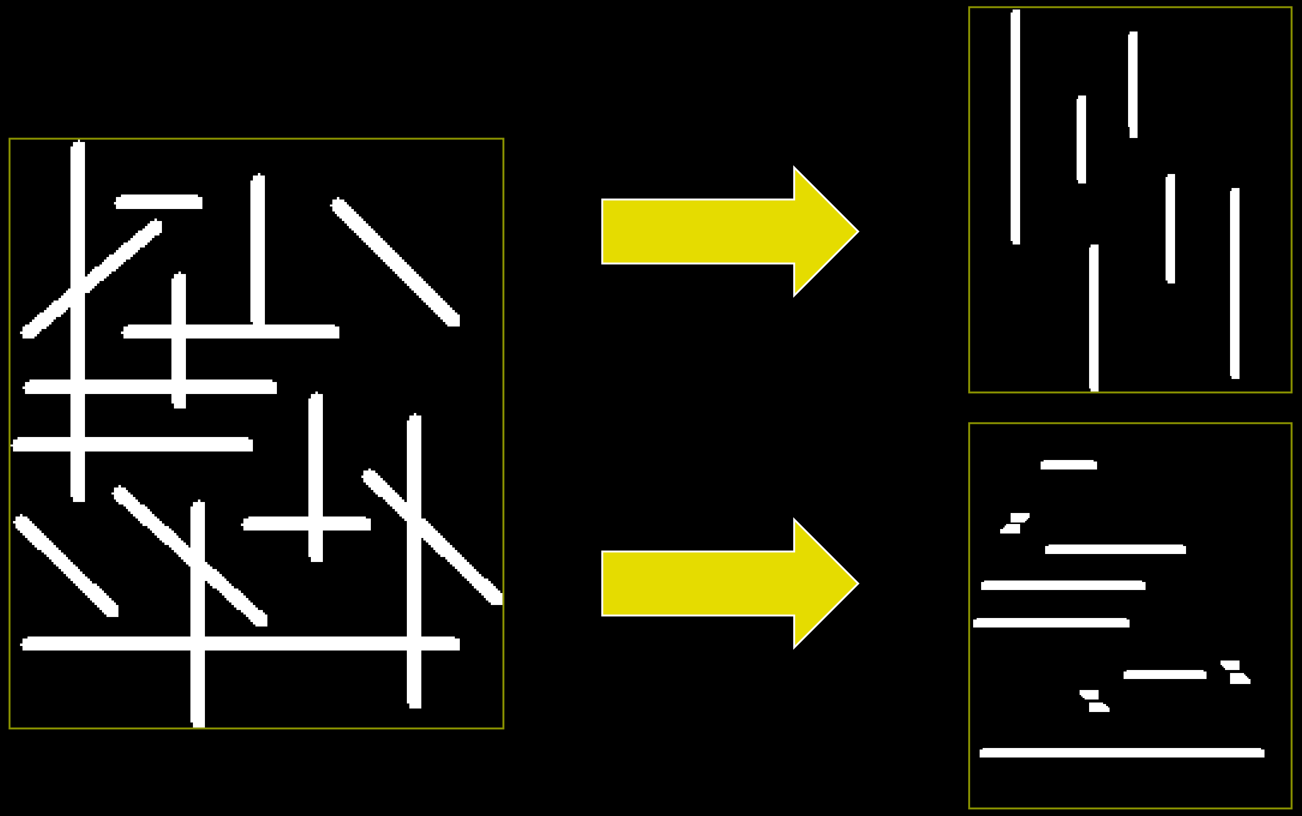
Opening = erosion+dilation



SE

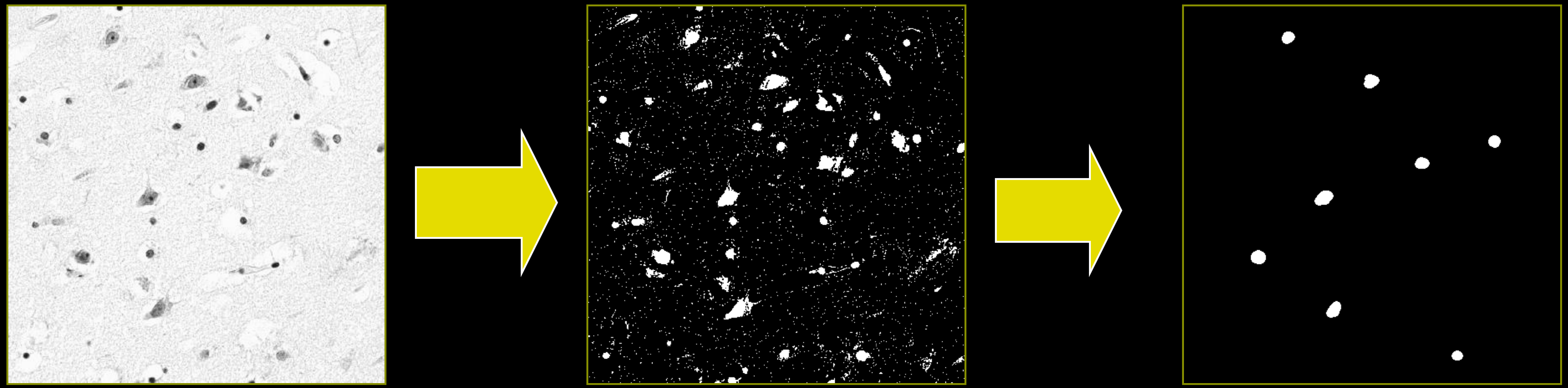
Opening Example

- 9x3 and 3x9 Structuring Elements



Opening example

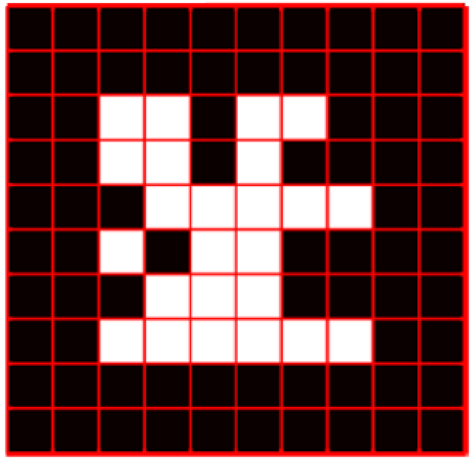
- Size of structuring element should fit into the smallest object to keep
- Structuring Element: 11 pixel disc

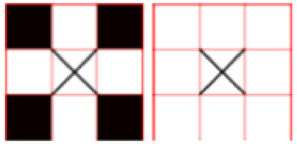


Compound operations on image

The compound morphological operation seen below is applied to the image. How many foreground pixels are there in the resulting image?

$$(I \ominus SE1) \oplus SE2,$$





3

23

11

36

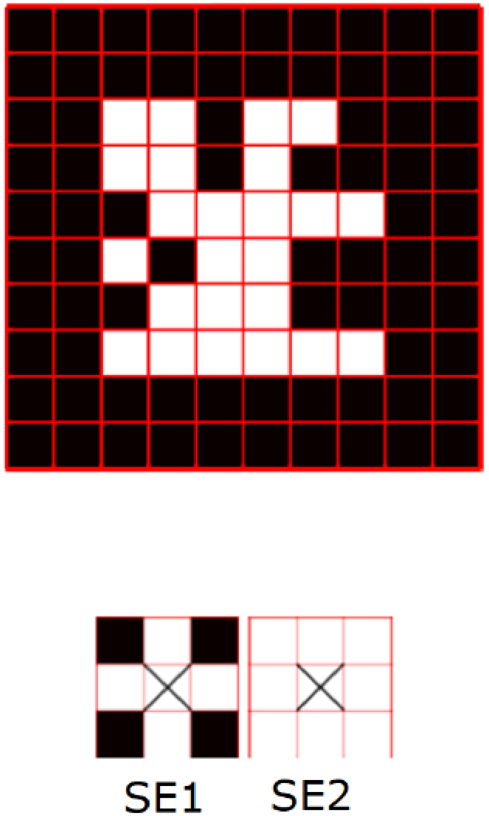
16

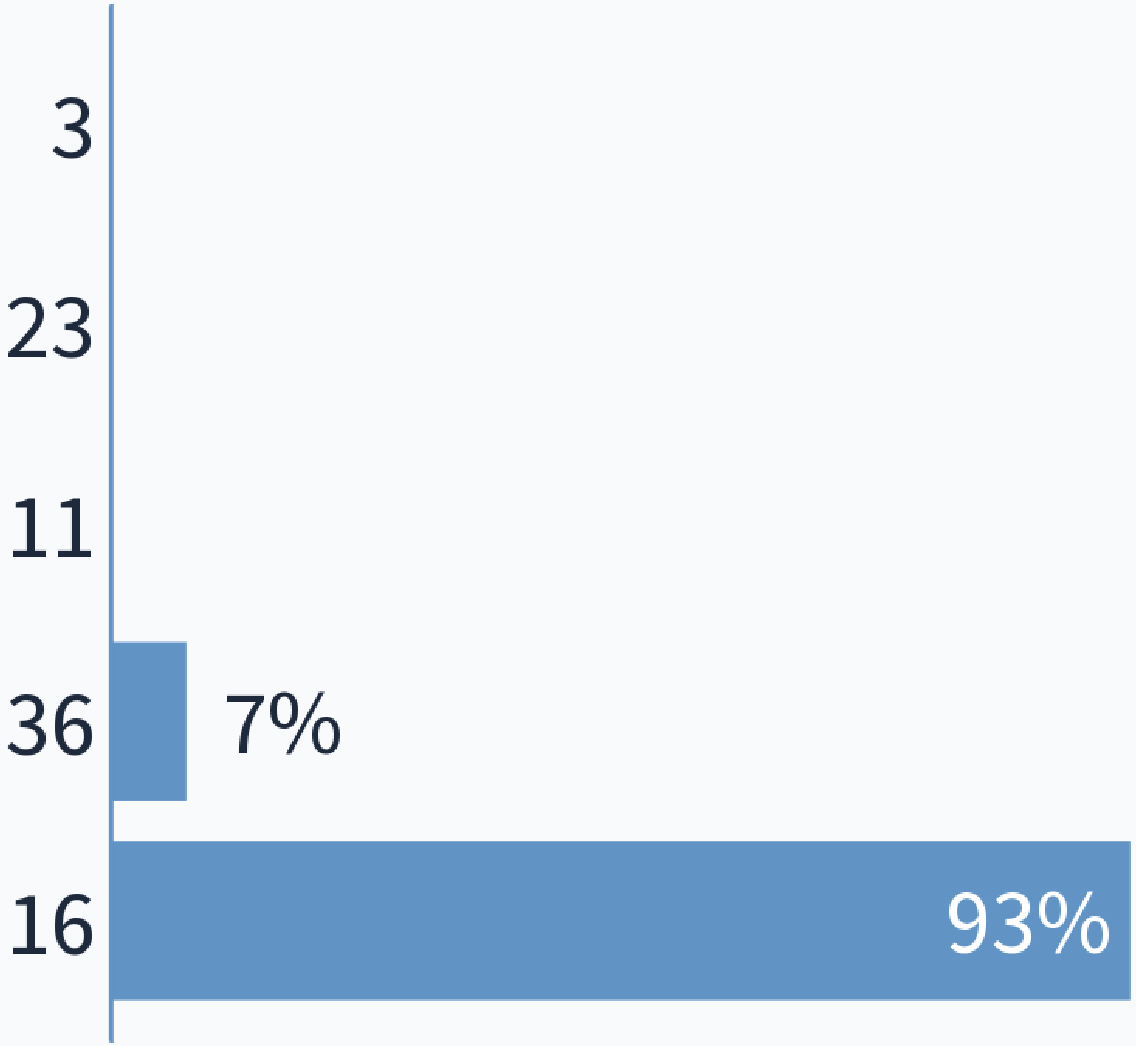
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Compound operations on image

The compound morphological operation seen below is applied to the image. How many foreground pixels are there in the resulting image?

$$(\mathrm{I} \ominus \mathrm{SE1}) \oplus \mathrm{SE2},$$



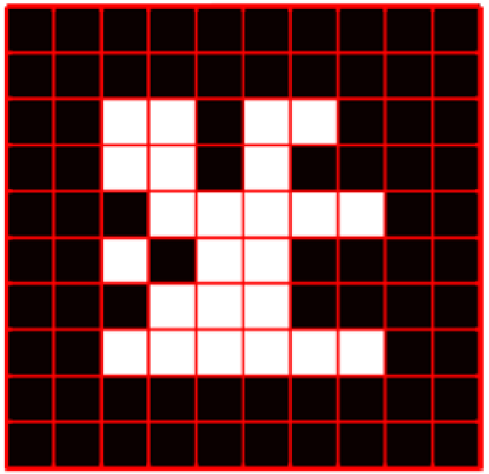


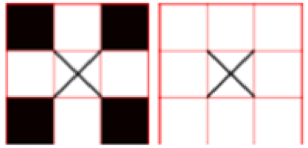
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Compound operations on image

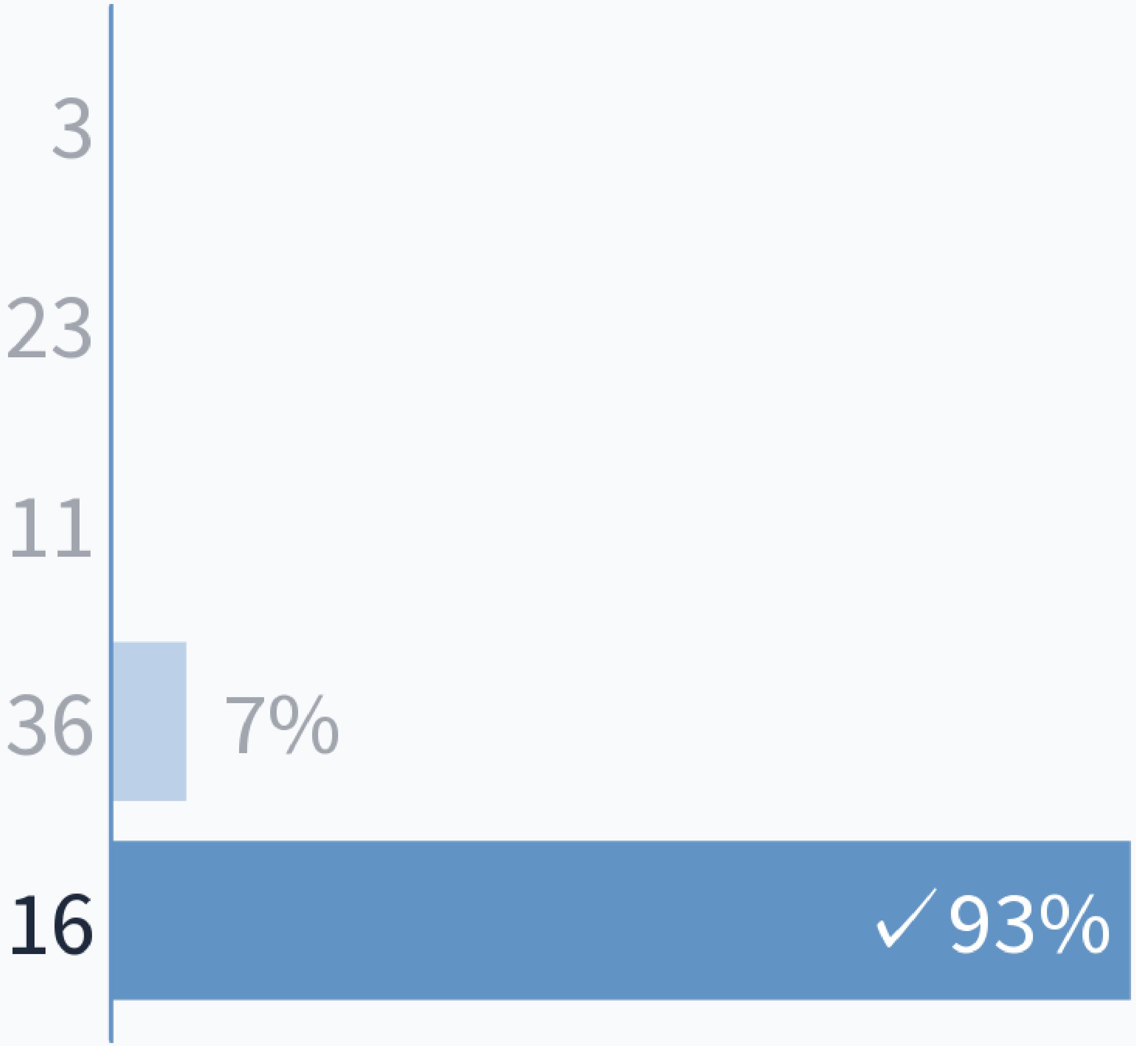
The compound morphological operation seen below is applied to the image. How many foreground pixels are there in the resulting image?

$$(\mathrm{I} \ominus \mathrm{SE1}) \oplus \mathrm{SE2},$$





SE1 SE2



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Closing

- Motivation: Fill holes BUT keep original size (and shape)
- Closing = Dilation + Erosion
  - Use the same structuring element!
  - Similar to Dilation but less destructive
- Math:

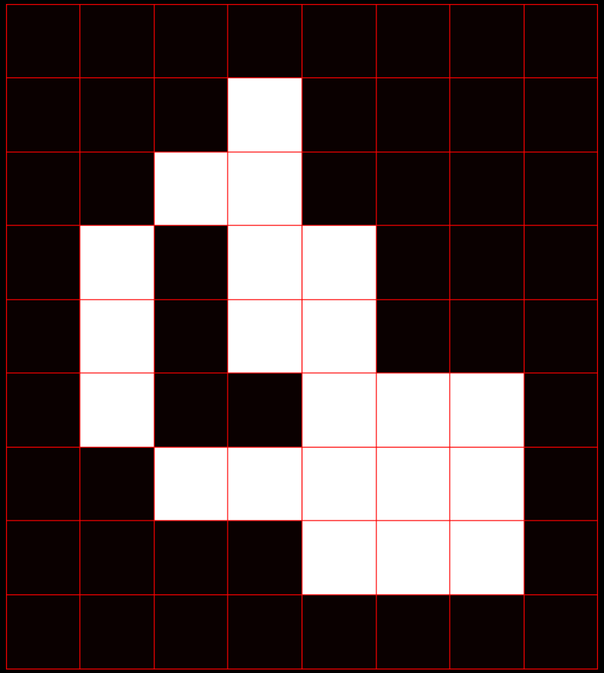
$$g(x,y) = f(x,y) \bullet SE = (f(x,y) \oplus SE) \ominus SE$$

- Closing is **idempotent:** Repeated operations has no further effects!

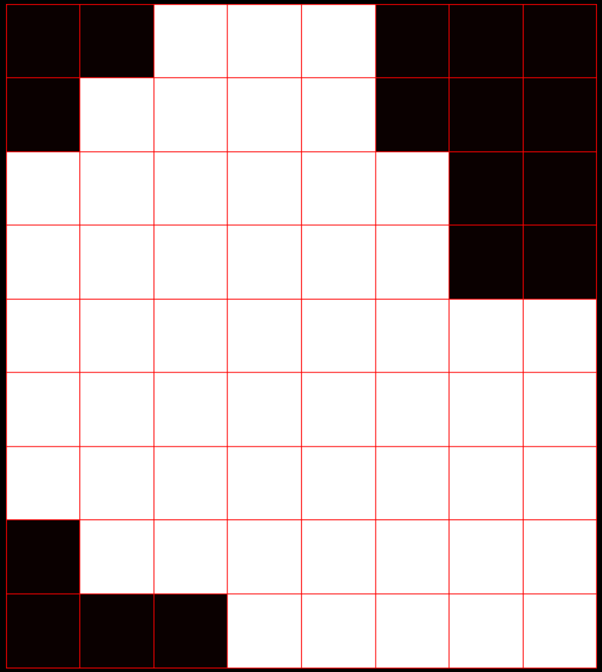
$$f(x,y) \circ SE = (f(x,y) \circ SE) \circ SE$$

Closing

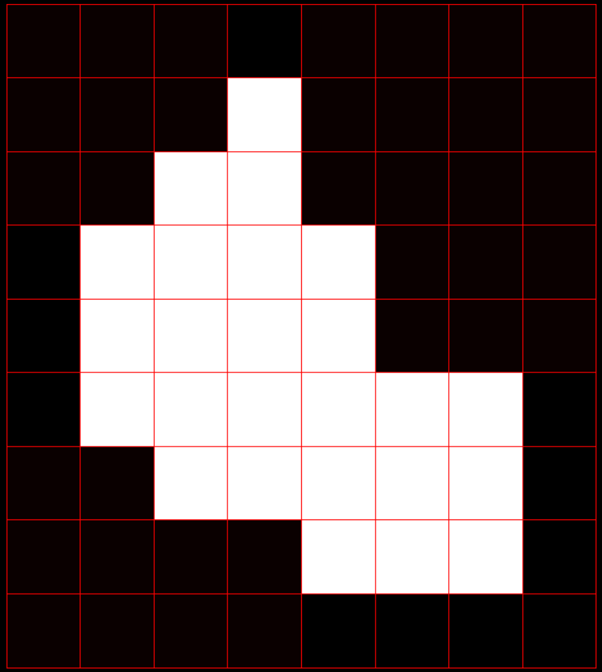
$$g(x,y) = (f(x,y) \oplus SE) \ominus SE$$



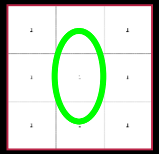
Original



Dilated



Eroded

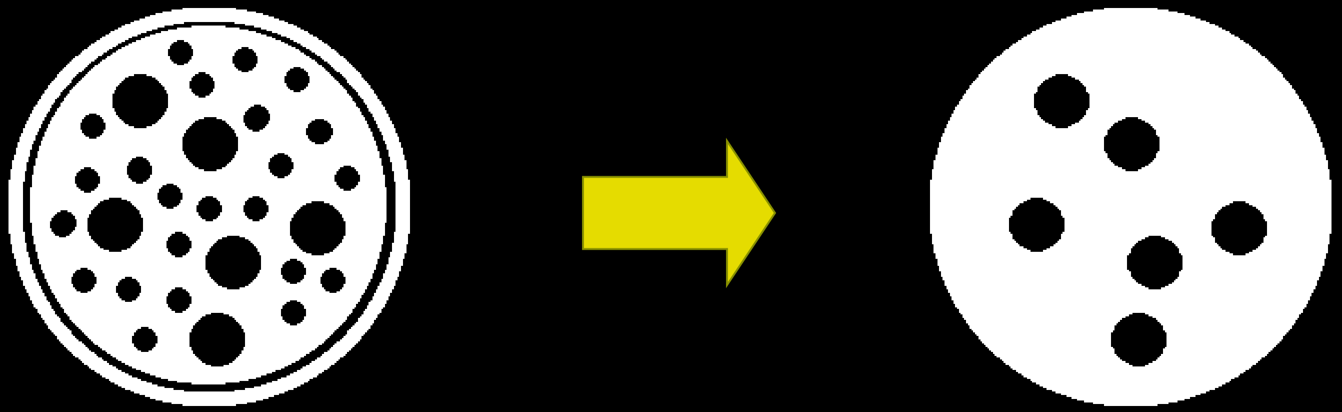


SE

Closing = dilation + erosion

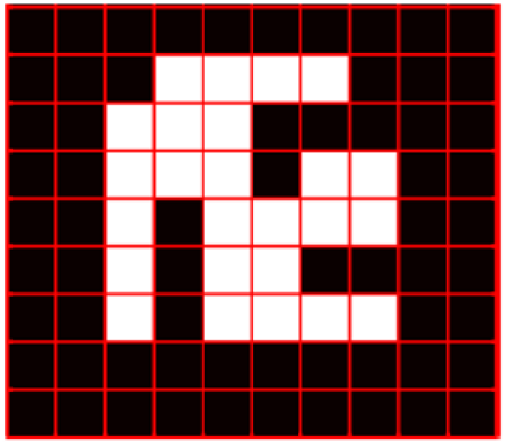
Closing Example

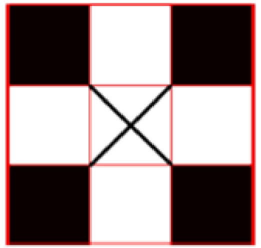
- Closing operation with a 22 pixel disc
- Closes small holes



Closing on image

Morphological closing is applied to the image using the structuring element below. How many foregrounds pixels are there in the resulting image?



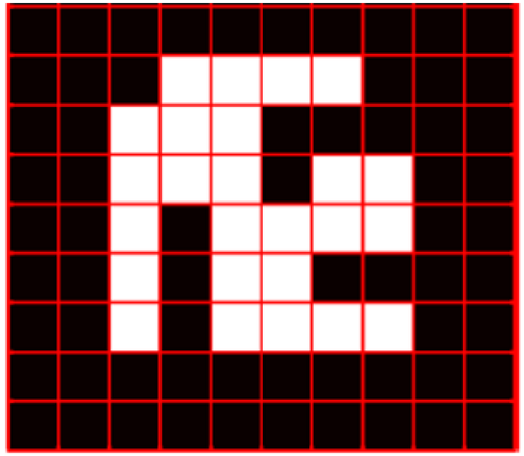


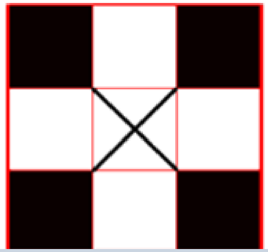
- 31
- 18
- 6
- 35
- 21

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Closing on image

Morphological closing is applied to the image using the structuring element below. How many foregrounds pixels are there in the resulting image?





31

18

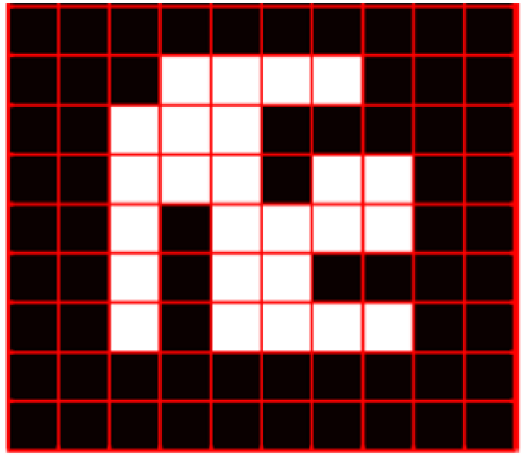
6

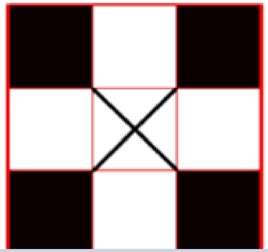
35

21

Closing on image

Morphological closing is applied to the image using the structuring element below. How many foregrounds pixels are there in the resulting image?





31 ✓0%

18

6

35

21

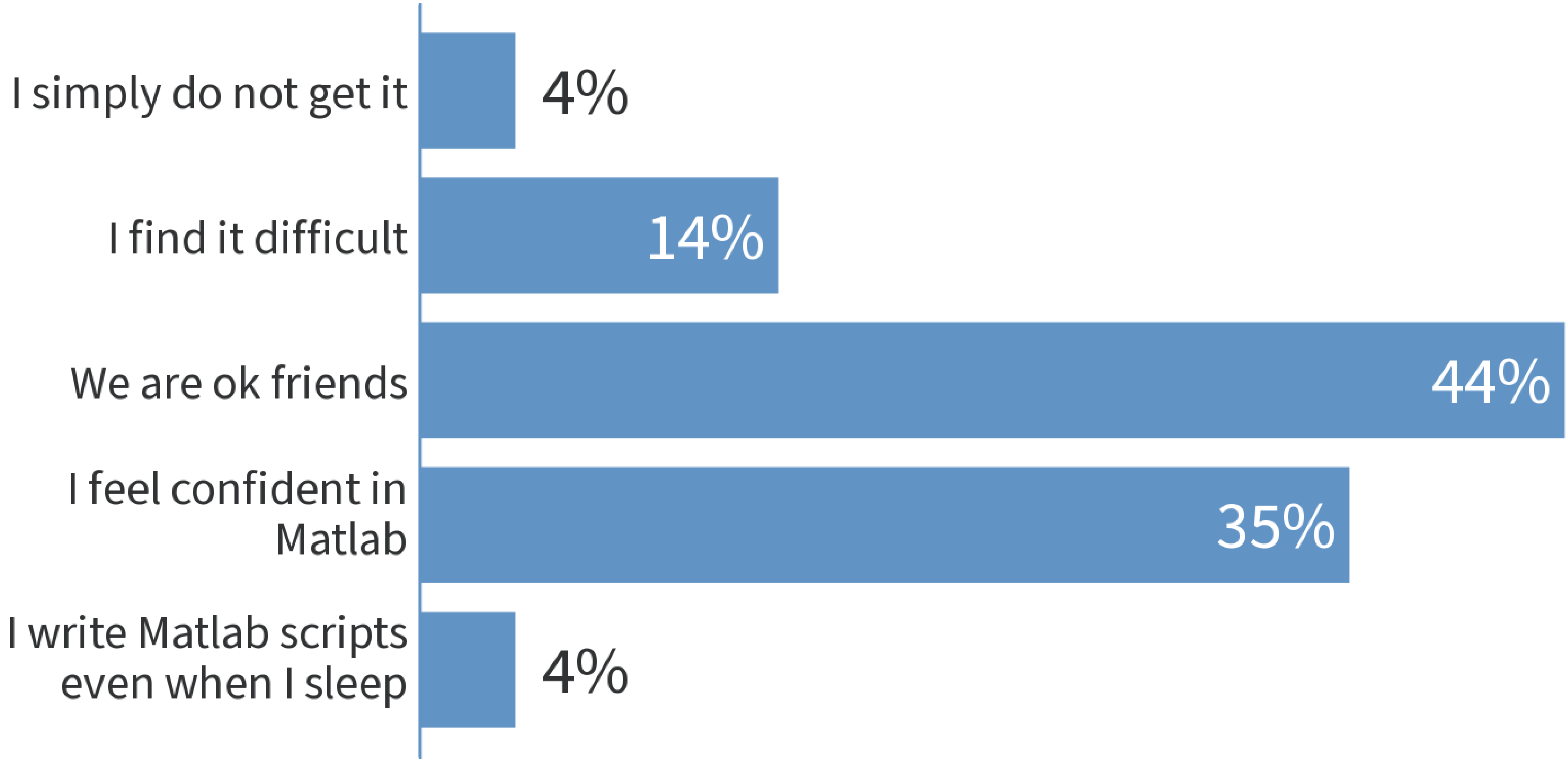
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

How do I feel about Matlab

- I simply do not get it
- I find it difficult
- We are ok friends
- I feel confident in Matlab
- I write Matlab scripts even when I sleep

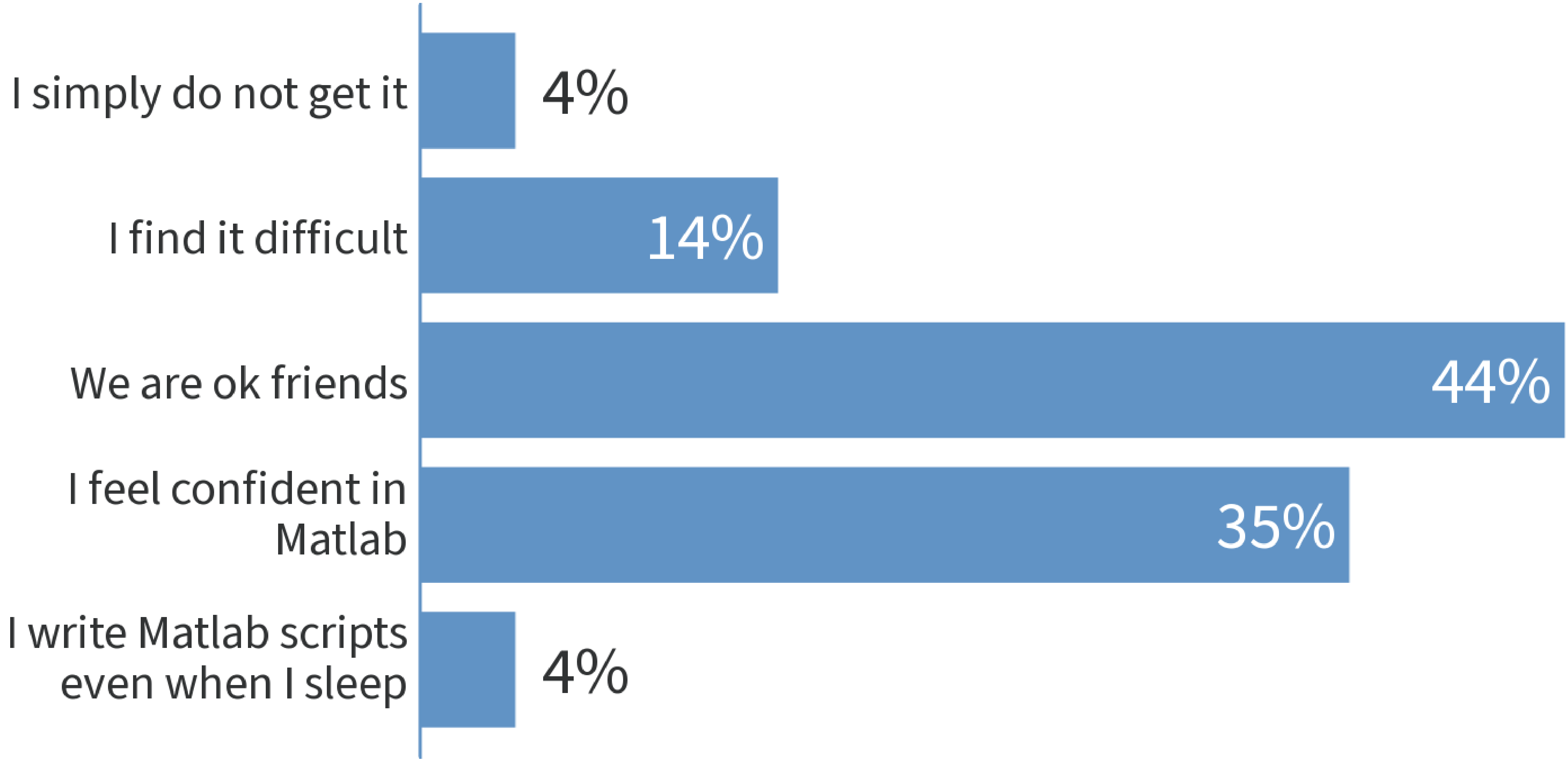
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

How do I feel about Matlab



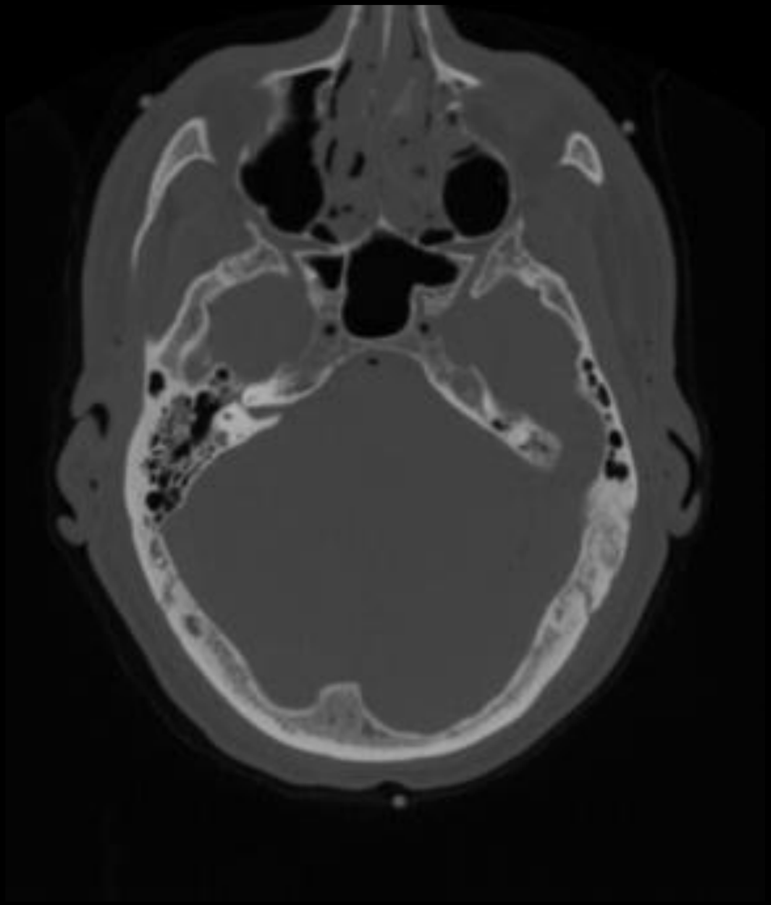
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

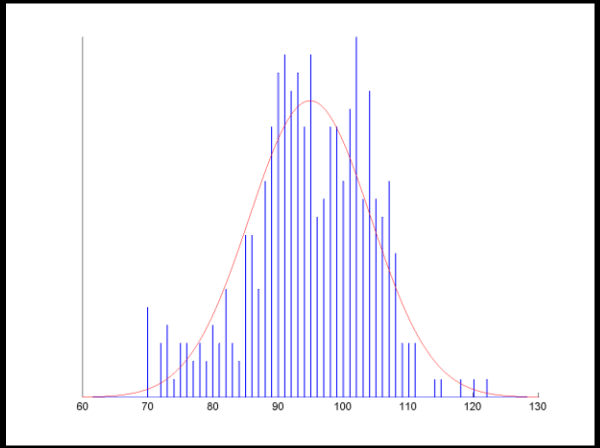
How do I feel about Matlab



Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Next week: Pixel classification





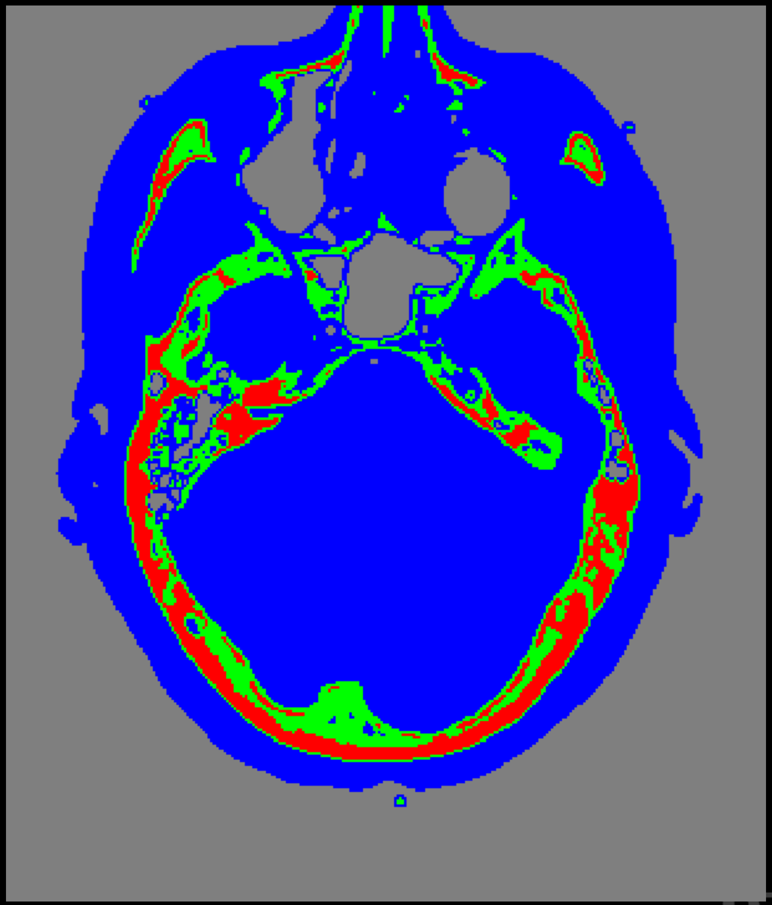


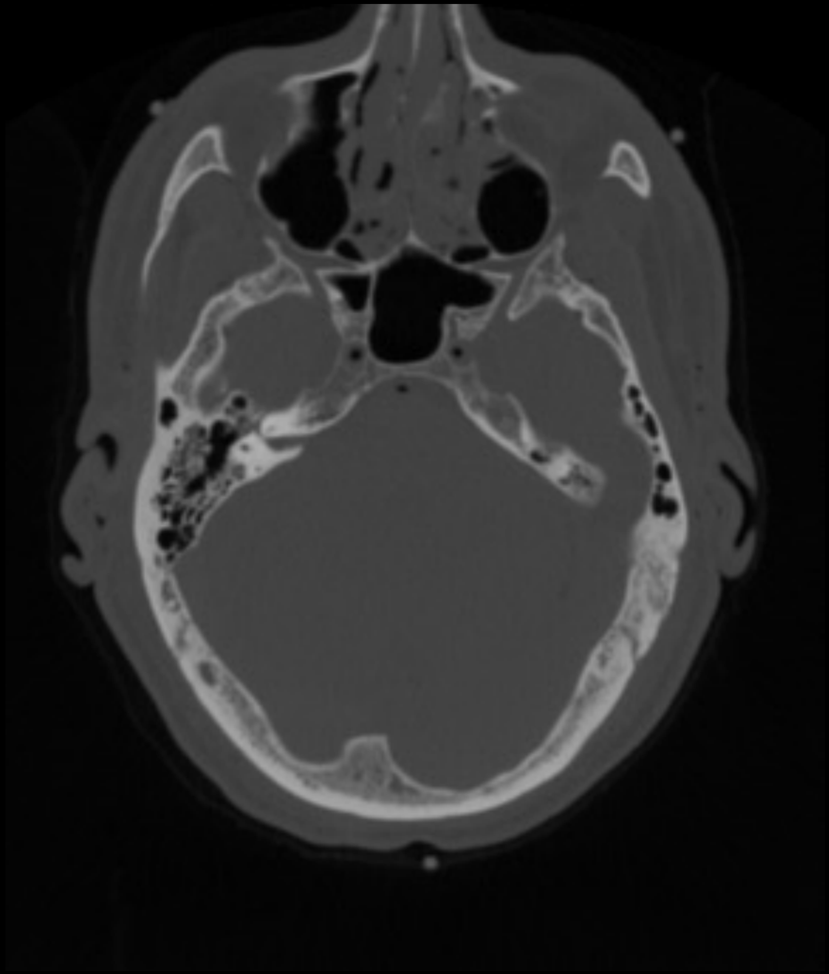
Image Analysis

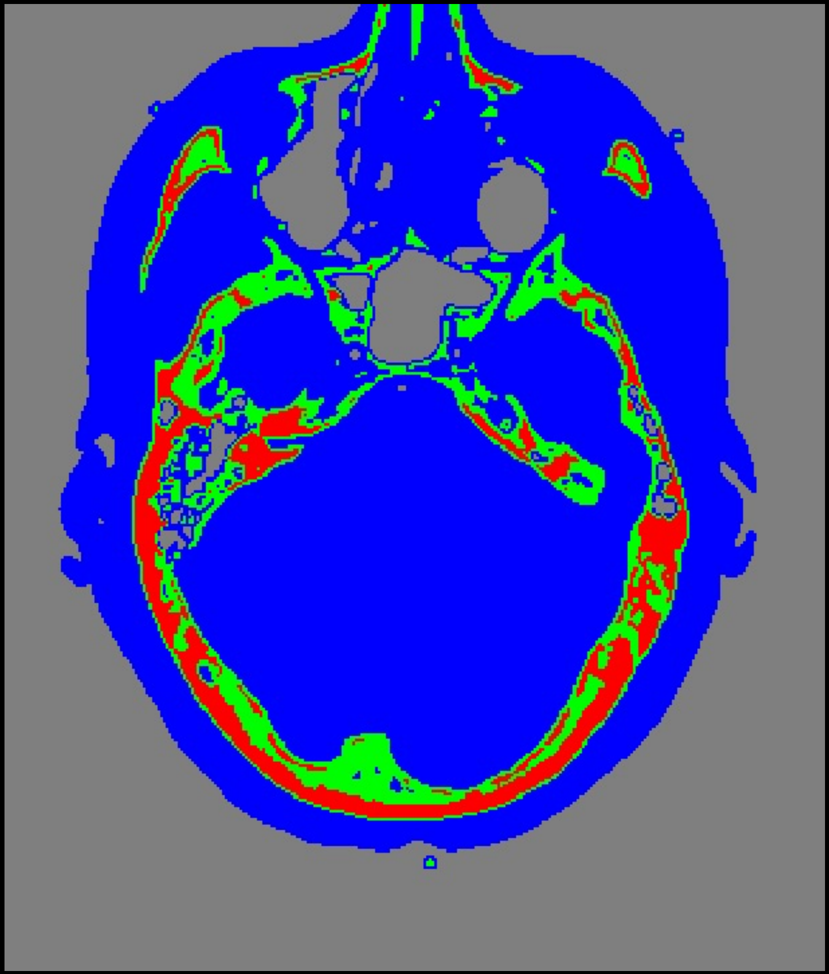
Tim B. Dyrby
Rasmus R. Paulsen
DTU Compute

tbdy@dtu.dk

http://www.compute.dtu.dk/courses/02502

Lecture 5 – Pixel Classification and advanced segmentation



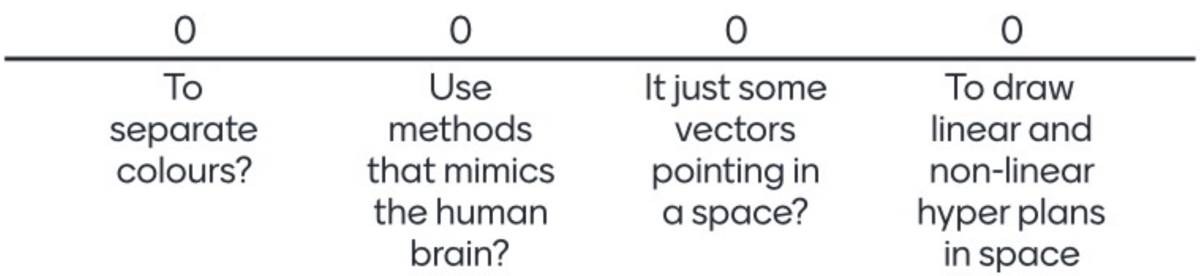


What can you do after today?

- Describe the concept of pixel classification
- Compute the pixel value ranges in a minimum distance classifier
- Implement and use a minimum distance classifier
- Approximate a pixel value histogram using a Gaussian distribution
- Implement and use a parametric classifier
- Decide if a minimum distance or a parametric classifier is appropriate based on the training data
- Explain the concept of Bayesian classification
- Understand the use of 1D vs 2D feature space
- Implement and use the linear discriminant analysis (LDA) classifier
- Understand the use of linear vs non-line hyper-plans for segmentation

Go to **www.menti.com** and use the code **5648 1375**

Quiz 0: What is advanced segmentation?



Classification

- Take a measurement and put it into a class

Measurement

Wheels: 2

HP: 50

Weight: 200

Classifier →

Classes

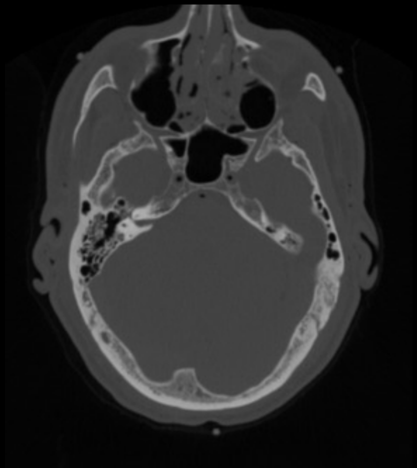
- Bike
- Truck
- Car
- Motorbike
- Train
- Bus

General Classification

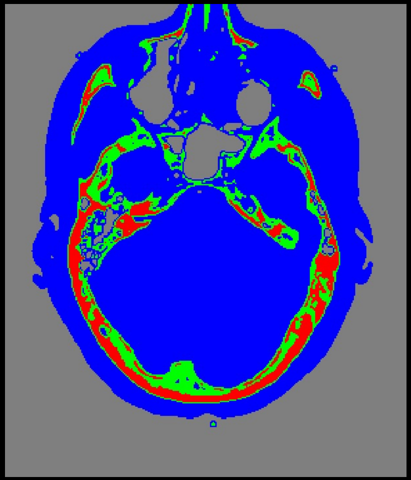
- Multi-dimensional measurement
- Pre-defined classes
  - Can also be found automatically – can be very difficult!

Pixel Classification

CT scan of human head



Pixel wise segmentation



Four Class labels

Background

Soft-Tissue

Trabecular Bone

Hard Bone

- Classify each pixel
  - Independent of neighbours
- Also called labelling
  - Put a label on each pixel
- We look at the pixel value and assign them a label
- Labels already defined

Quiz 1: Two class pixel classification?

Background and object

A) Median filter
B) Threshold
C) Brightness
D) Morphological Erosion
E) BLOB analysis

Pixel Classification – formal definition

Pixel value (the measurement) $v \in R$

k classes $C = c_1, \dots, c_k$

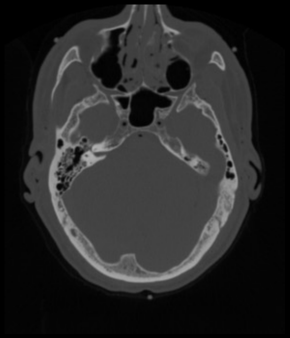
Classification rule $c: R \longrightarrow \{c_1, \dots, c_k\}$

Pixel Classification – example

Pixel value $v \in [0{,}255]$

Set of 4 classes C={background, soft-tissue, trabeculae, bone}

Classification rule $c: v \longrightarrow \{\text{background}, \text{soft} - \text{tissue}, \text{trabeculae}, \text{bone}\}$



How do we construct a classification rule?

Pixel classification rule

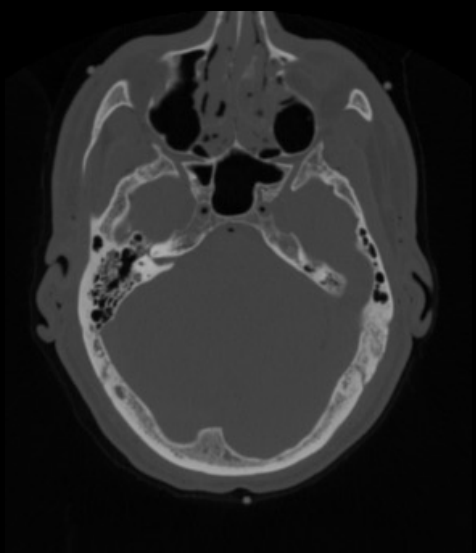
$$c: v \longrightarrow \{\text{background}, \text{soft} - \text{tissue}, \text{trabeculae}, \text{bone}\}$$

background

trabeculae

soft-tissue

bone

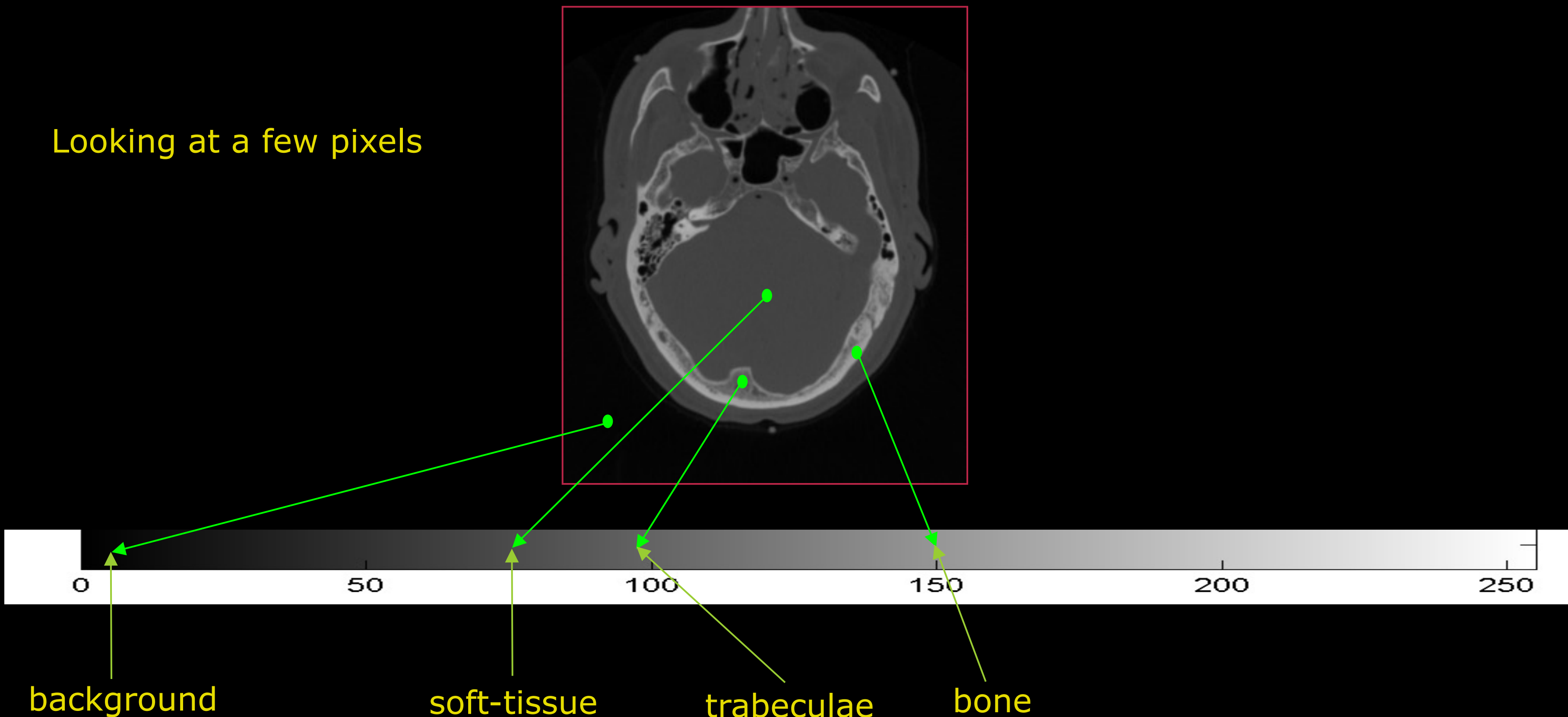


How do we do this?

Pixel classification rule – manual inspection

$c: v \longrightarrow \{\text{background}, \text{soft} - \text{tissue}, \text{trabeculae}, \text{bone}\}$

Looking at a few pixels

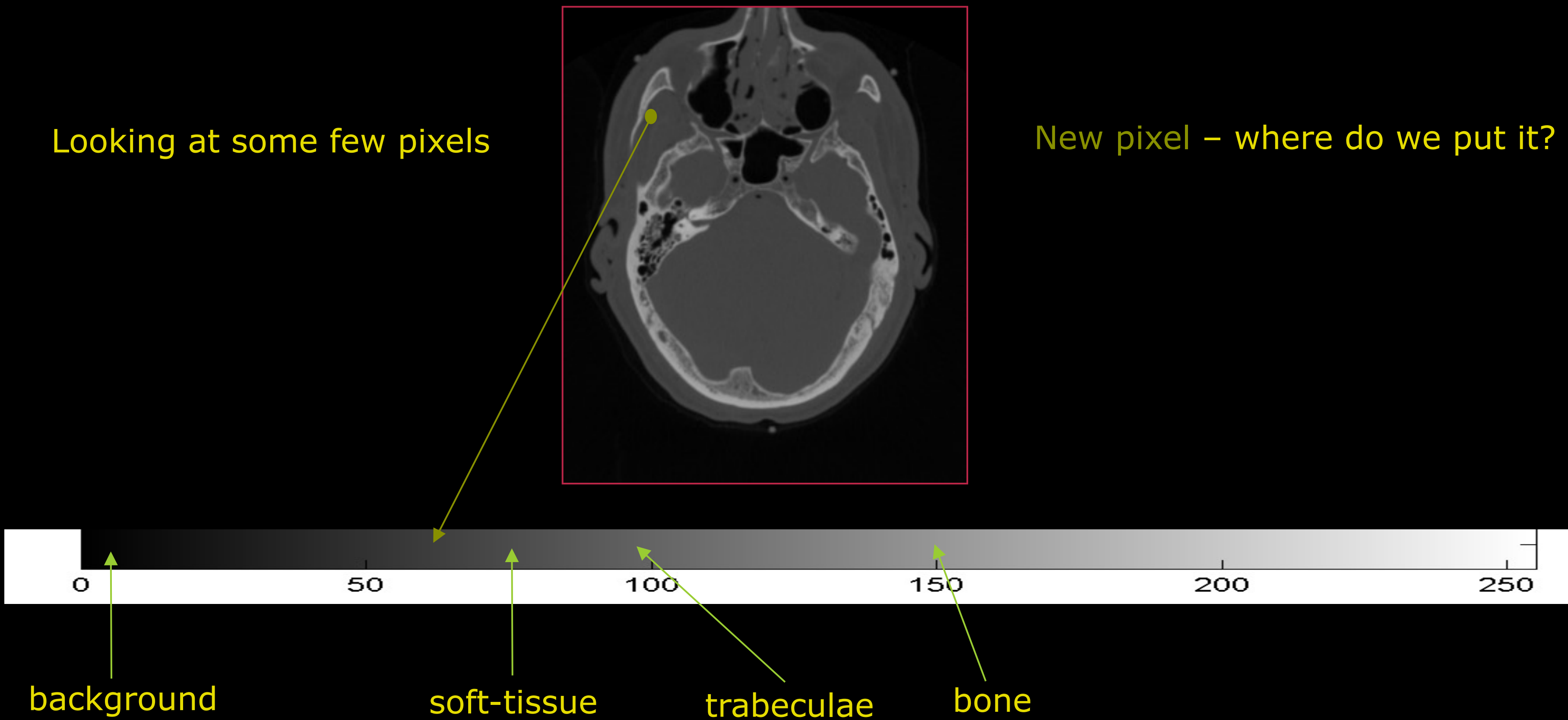


Pixel classification rule – manual inspection

$$c: v \longrightarrow \{\text{background}, \text{soft} - \text{tissue}, \text{trabeculae}, \text{bone}\}$$

Looking at some few pixels

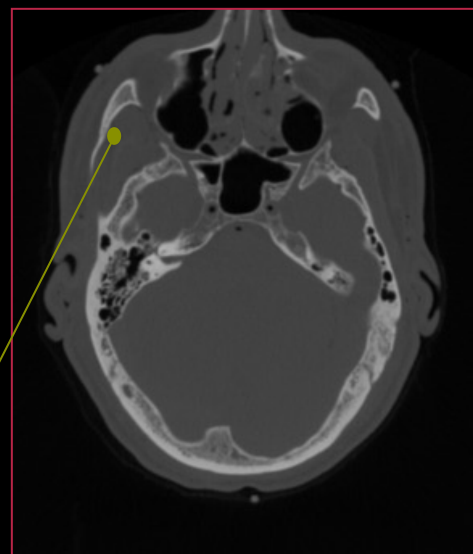
New pixel – where do we put it?



Pixel classification rule – manual inspection

$$c: v \longrightarrow \{\text{background}, \text{soft}-\text{tissue}, \text{trabeculae}, \text{bone}\}$$

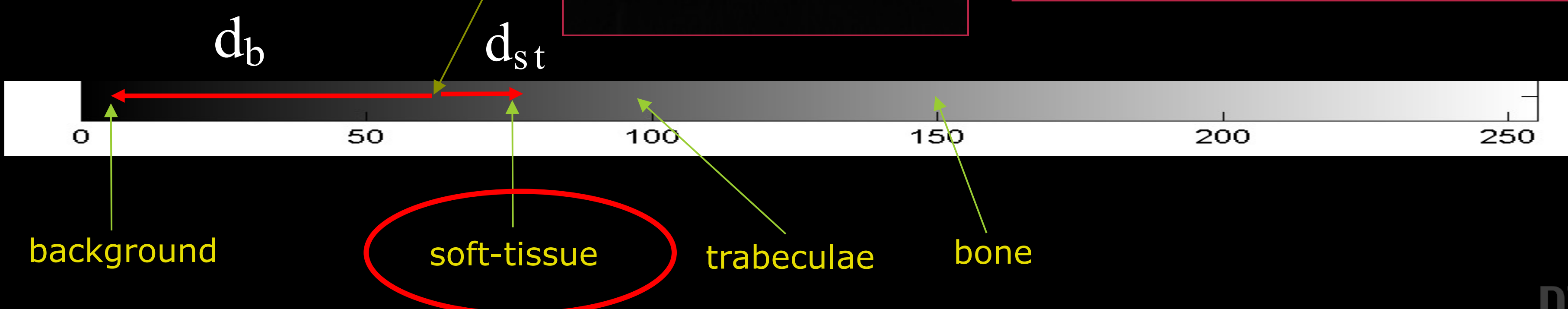
Looking at some few pixels



New pixel – where do we put it?

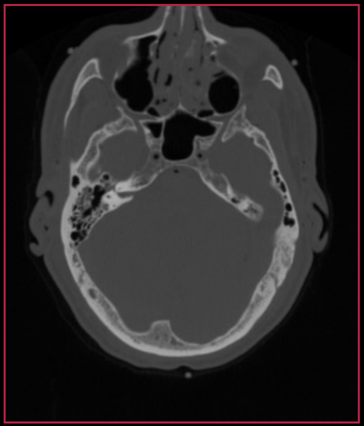
- Measure the "distance" to the other classes
- Select the closest class

Minimum distance classification

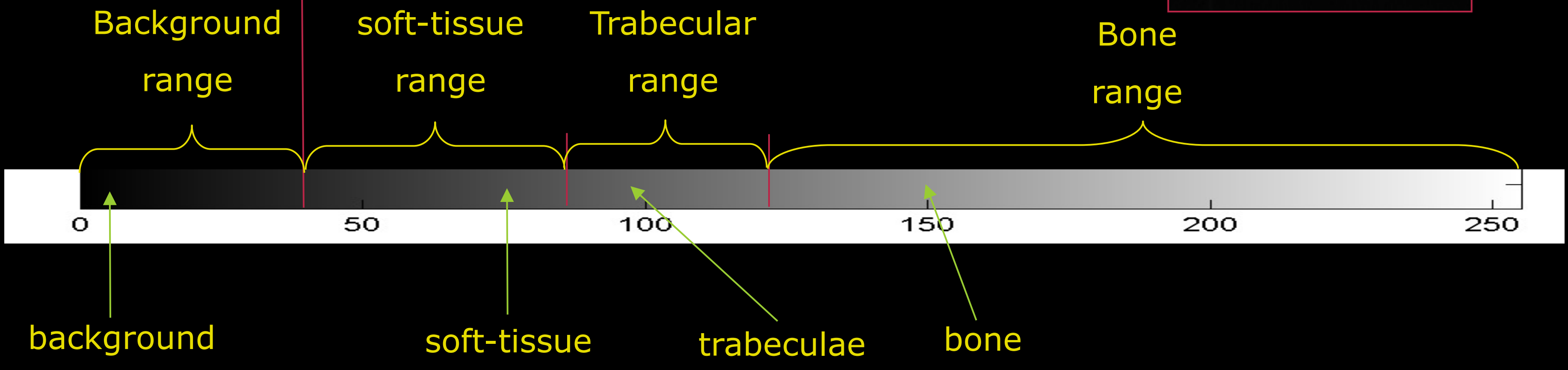


Pixel classification rule
Minimum Distance Classification

The possible pixel values are divided into ranges



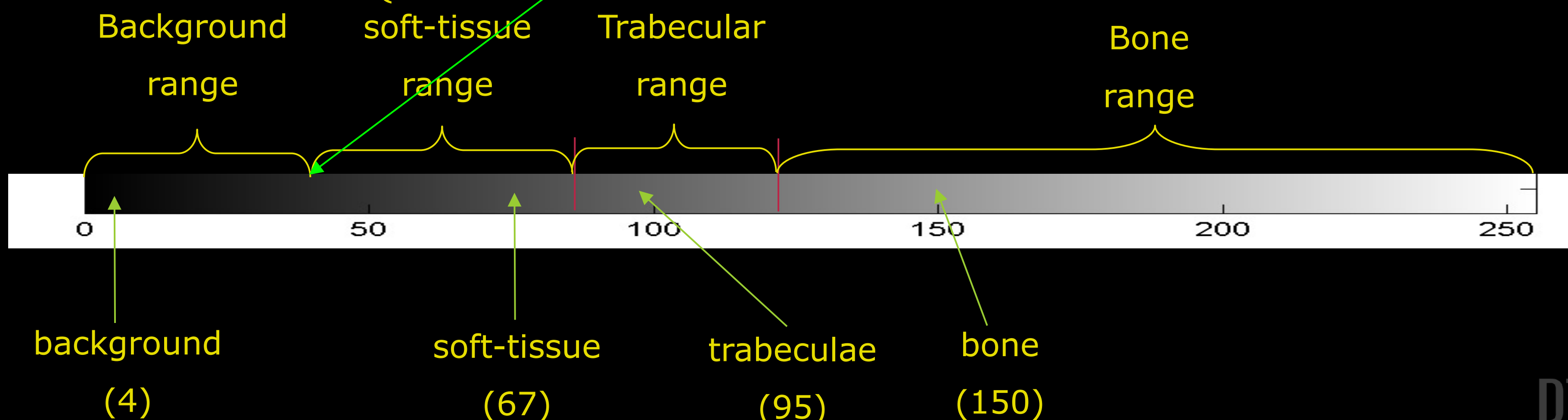
Here the distance to "background" is equal to "soft-tissue"



Pixel classification rule

Minimum Distance Classification

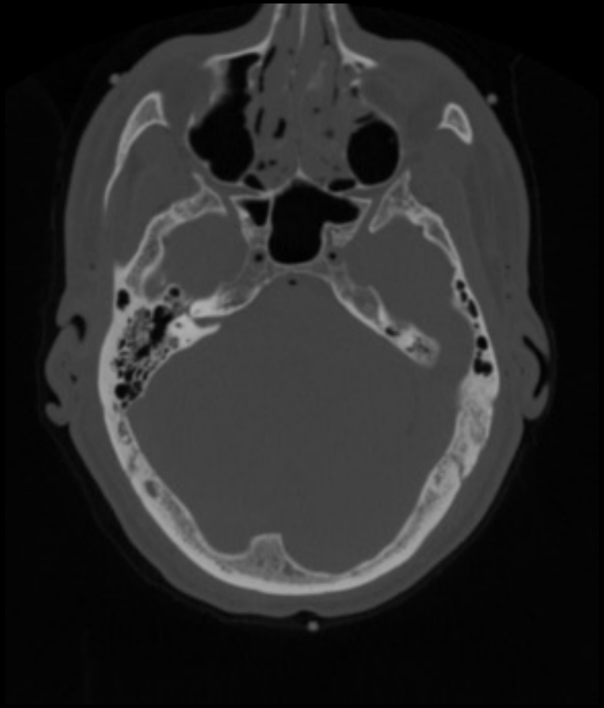
$$c(v) = \begin{cases} \text{background, if } v \le (4+67)/2 \\ \text{soft} - \text{tissue, if } \dfrac{(4+67)}{2} < v \le \dfrac{67+95}{2} \\ \text{trabeculae, if } \dfrac{67+95}{2} < v \le \dfrac{95+150}{2} \\ \text{bone, if } v > \dfrac{95+150}{2} \end{cases}$$



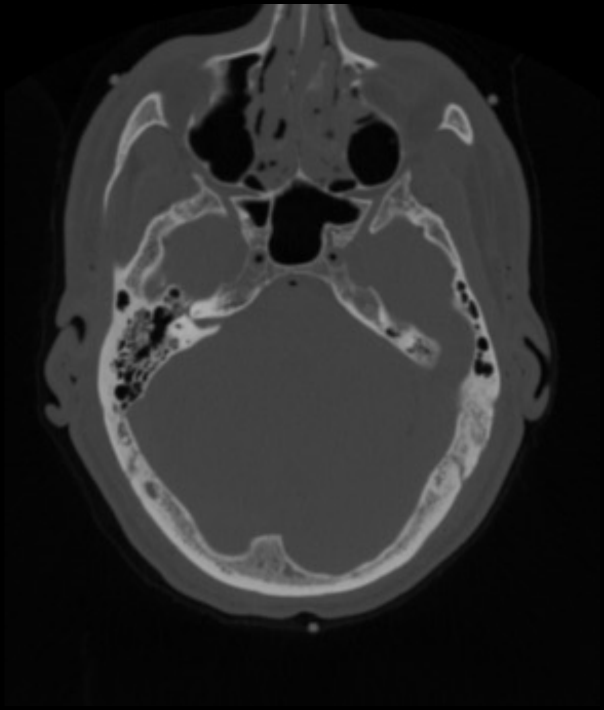
Pixel classification rule

- For all pixel in the image do

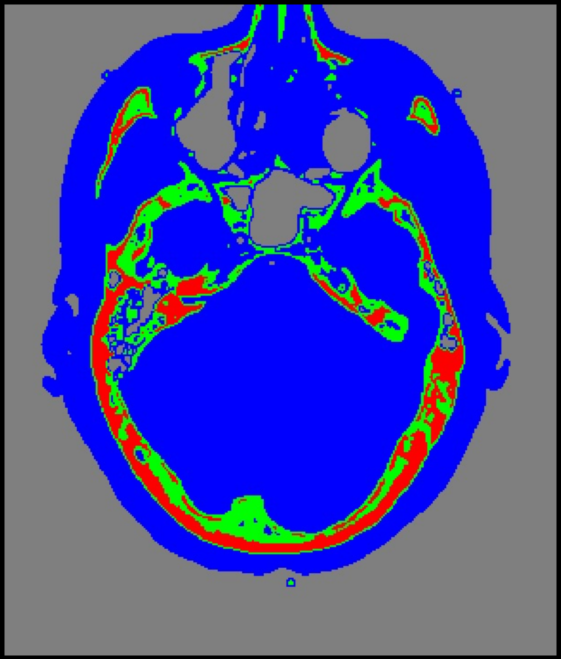
$$c(v) = \begin{cases} \text{background, if } v \leq (4+67)/2 \\ \text{soft} - \text{tissue, if } \dfrac{(4+67)}{2} < v \leq \dfrac{67+95}{2} \\ \text{trabeculae, if } \dfrac{67+95}{2} < v \leq \dfrac{95+150}{2} \\ \text{bone, if } v > \dfrac{95+150}{2} \end{cases}$$



Pixel Classification example



CT scan of human head



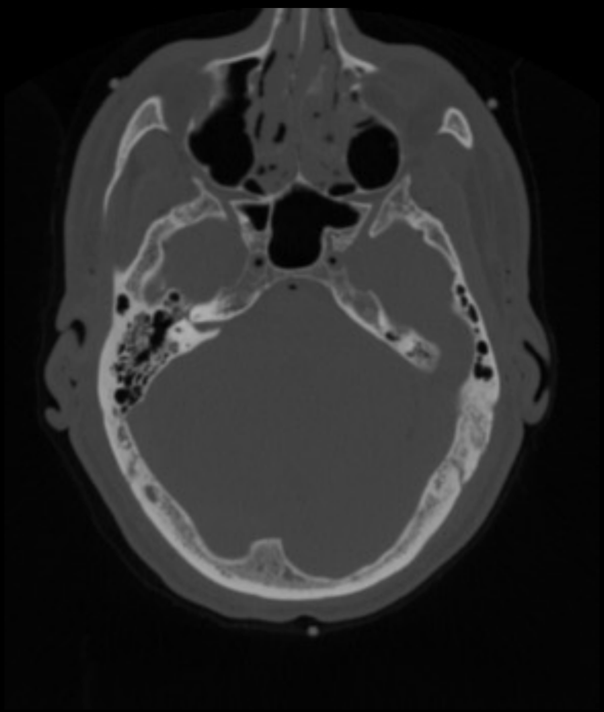
Background

Soft-Tissue

Trabecular Bone

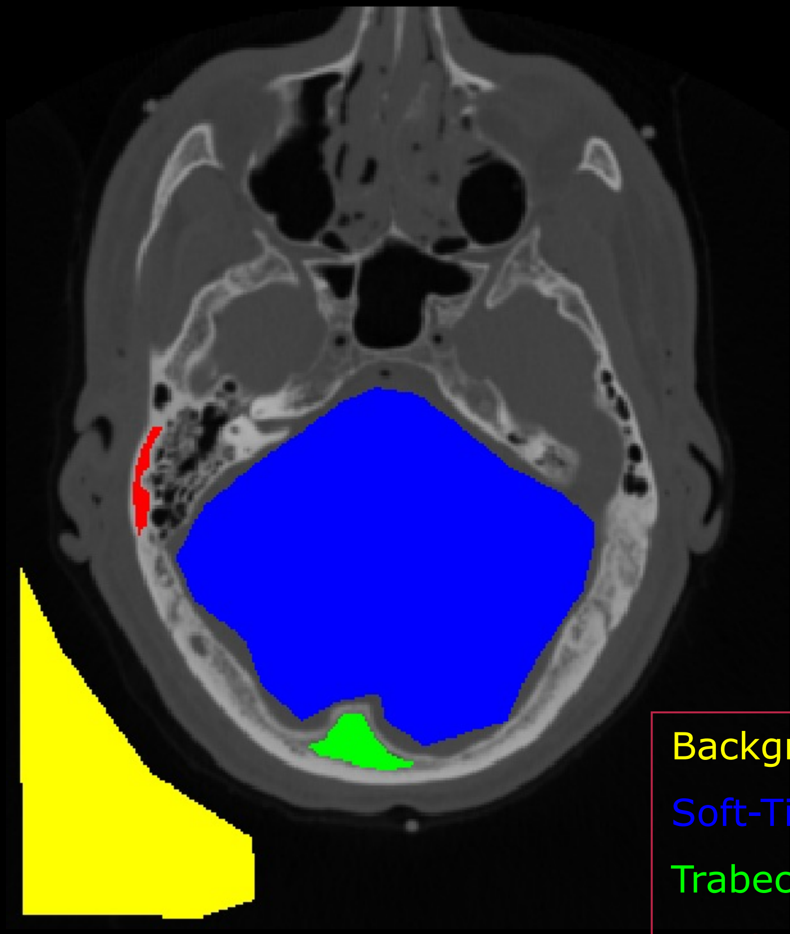
Hard Bone

Better range selection



- Guessing range values is not a good idea
- Better to use “training data”
- Start by selecting representative regions from an image
- *Annotation*
  - To mark points, regions, lines or other significant structures

Classifier training - annotation



- An “expert” is asked how many different tissue types that are possible
- Then the expert is asked to mark representative regions of the selected tissue types

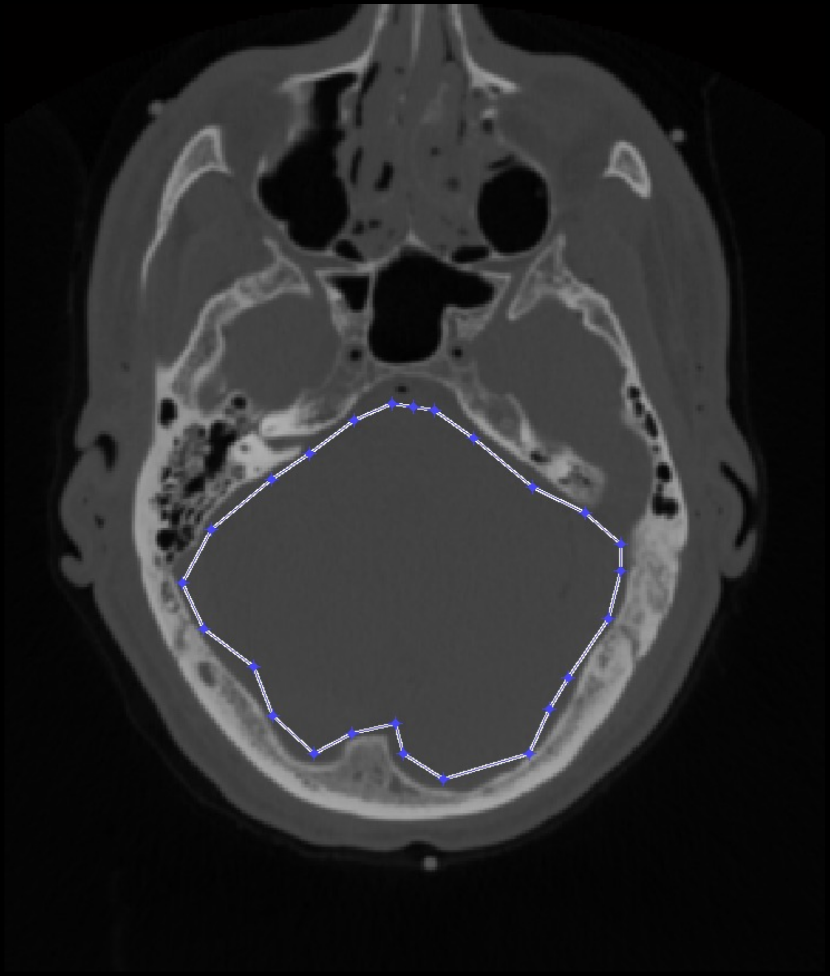
Background

Soft-Tissue

Trabecular Bone

Hard Bone

Classifier training – region selection

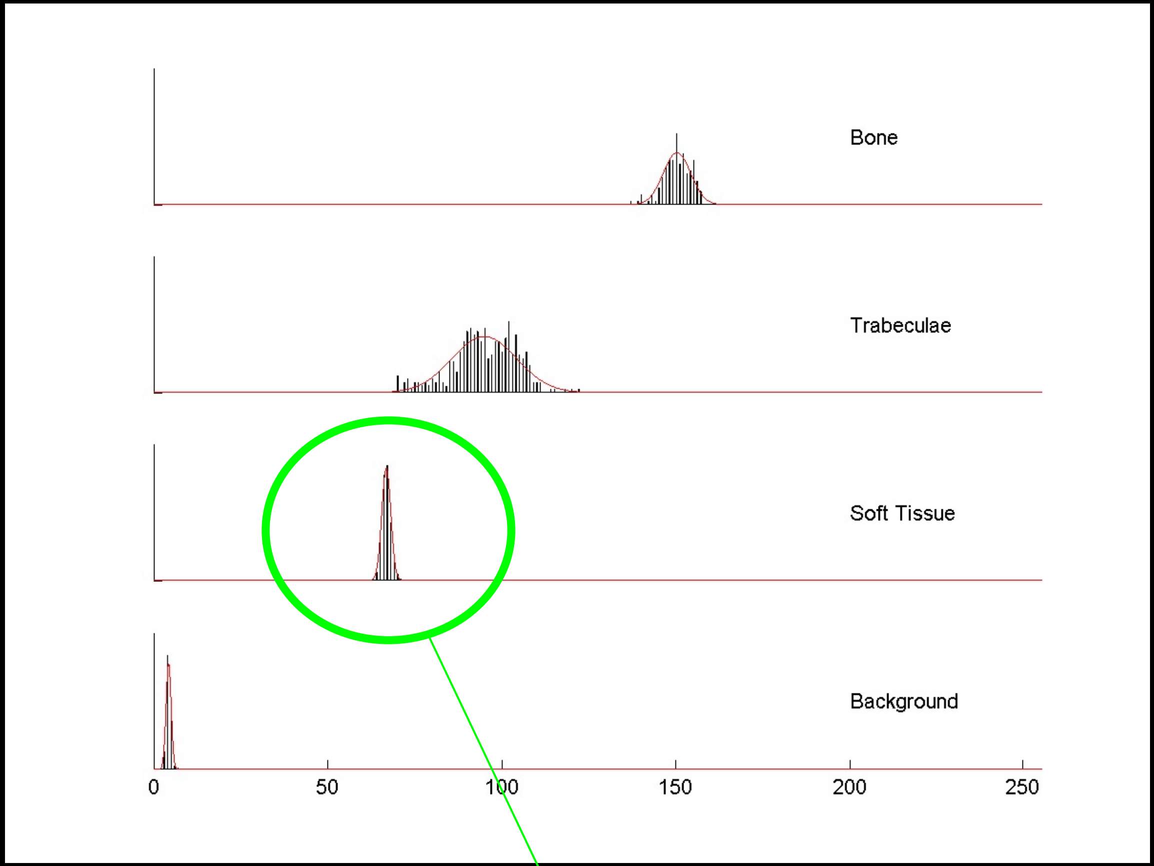


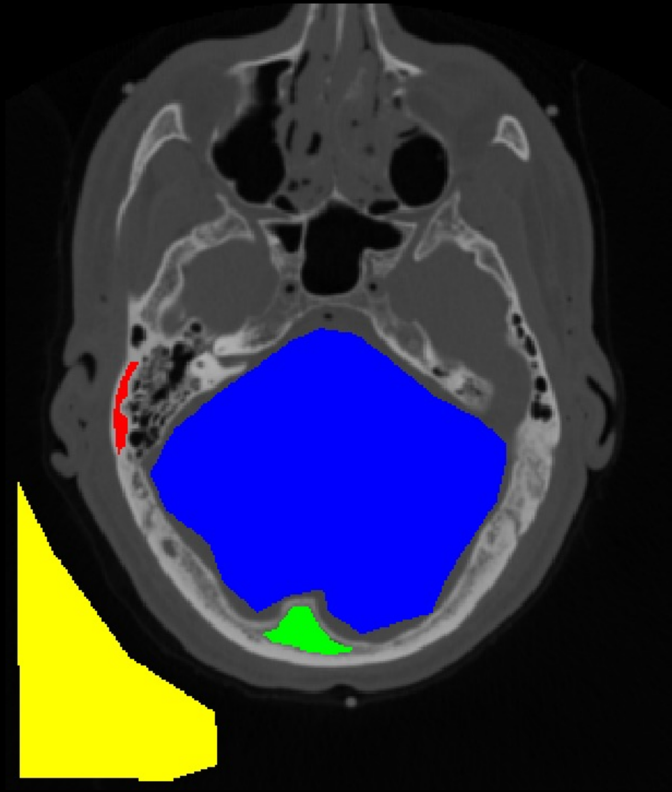
- Many tools exist
- Matlab tool roipoly
  - Select closed regions using a piecewise polygon

Training is only done once!

Optimally, the training can be used on many pictures that contains the same tissue types

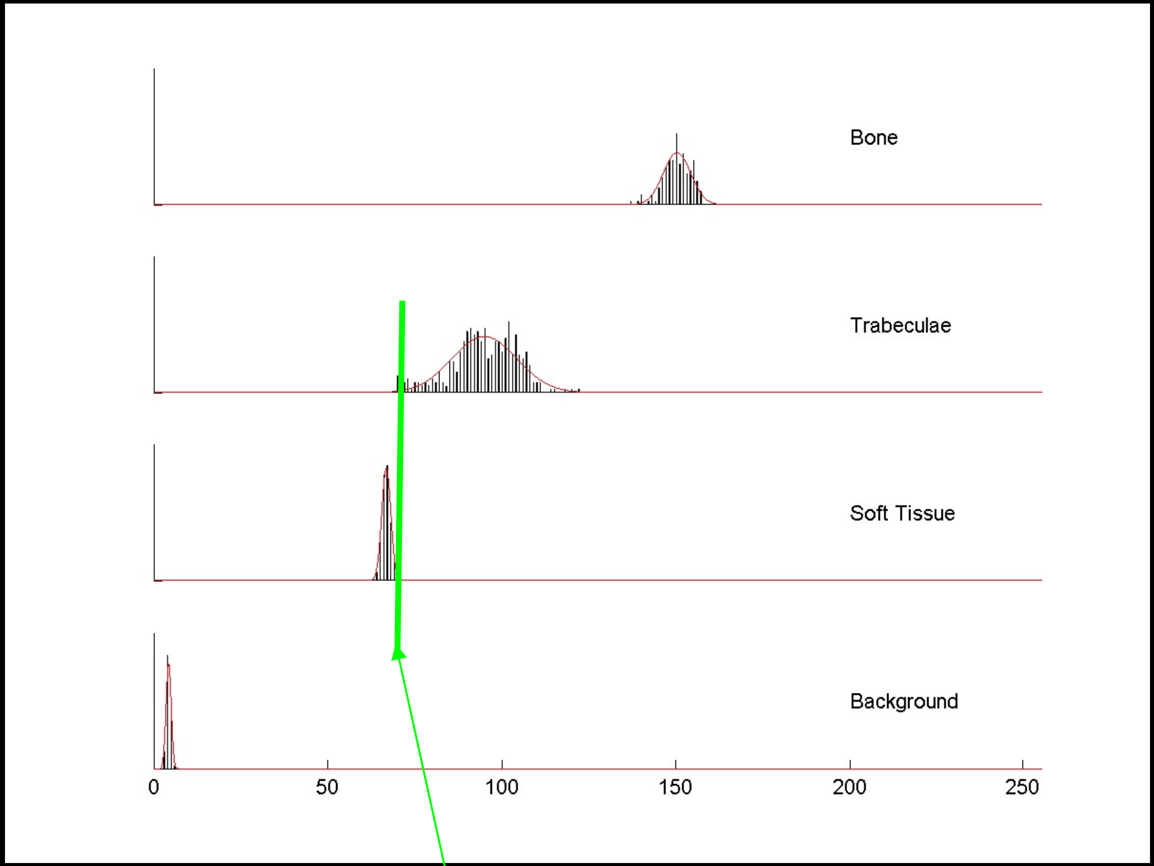
Initial analysis - histograms

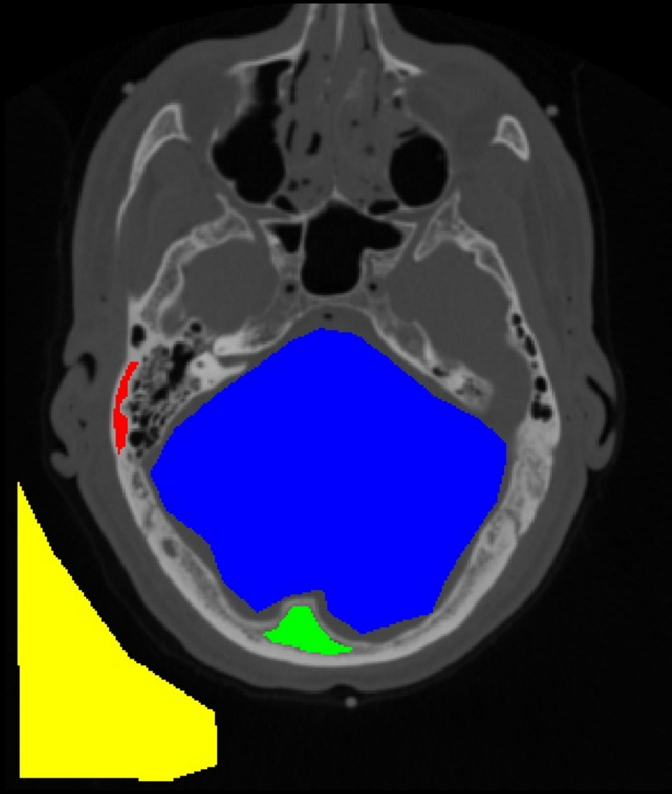




Gaussian

Initial analysis - histograms

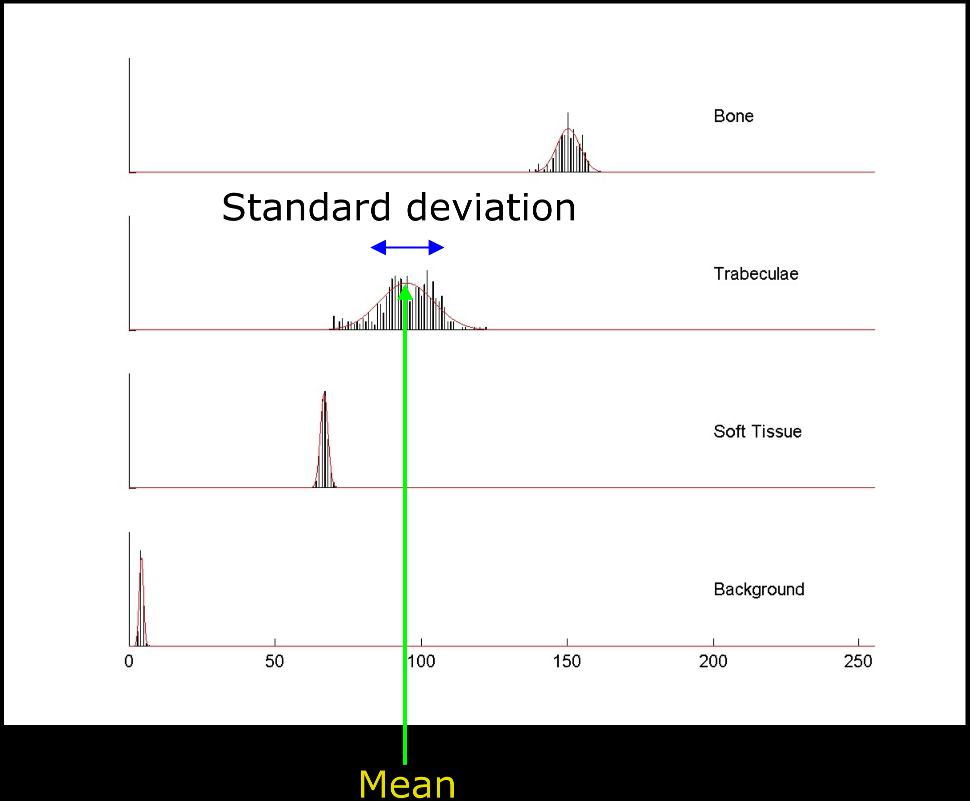


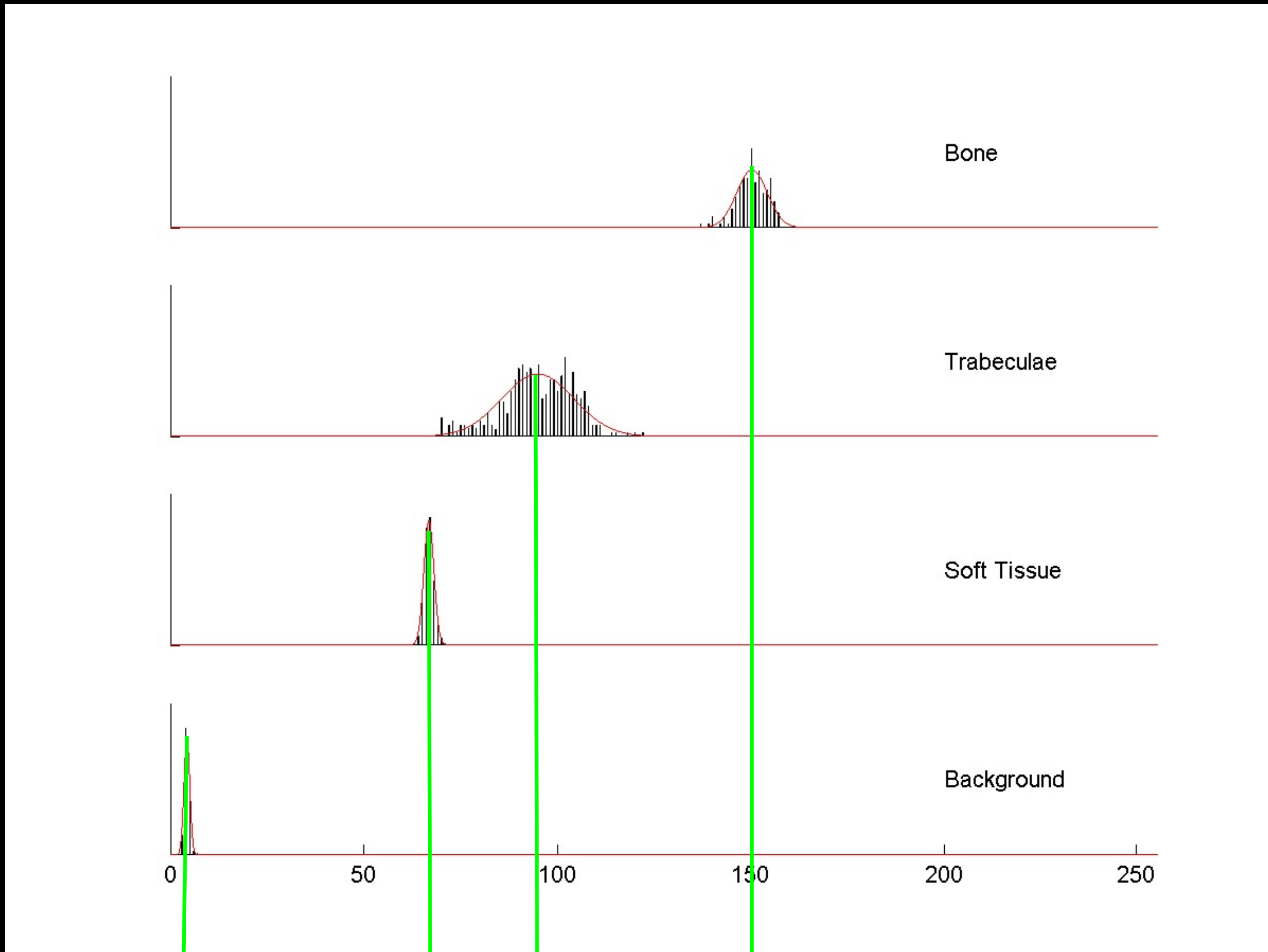


Class separation

Simple pixel statistics

- Calculate the mean and the standard deviation of each class

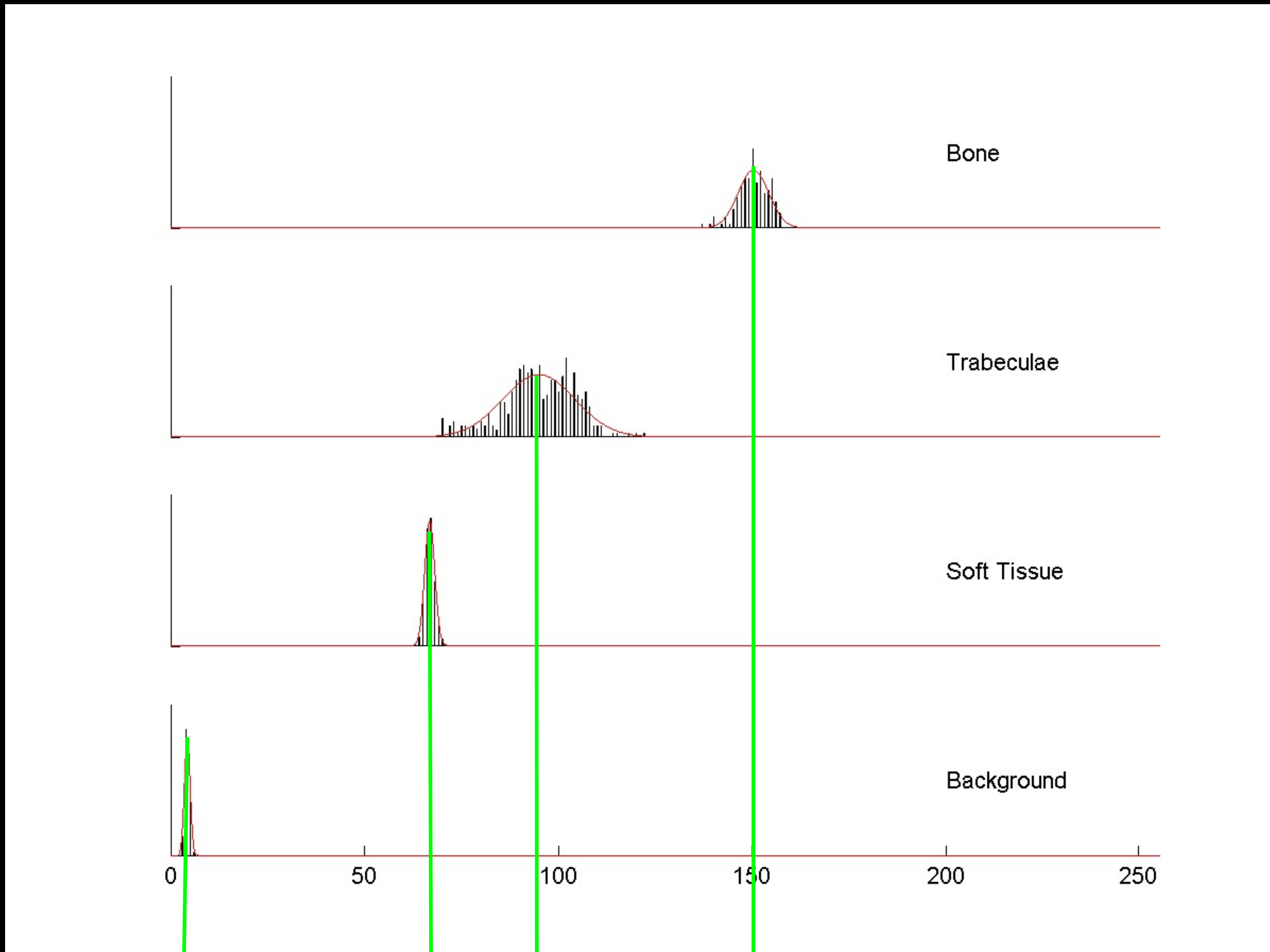


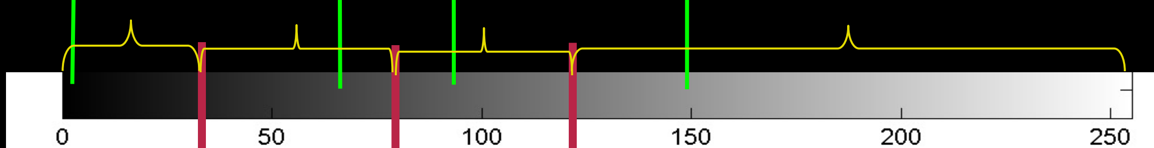


Bone
Trabeculae
Soft Tissue
Background
0
50
100
150
200
250

0
50
100
150
200
250

Minimum distance classification





Any objections?

The pixel value ranges are not always in good correspondence with the histograms?

Quiz 2: Minimum distance classification

A) Background

B) Soft tissue

C) Fat

D) Bone

E) None of the above

Solution:

Green: (6+4+9+5)/4=6

Blue: (132+130+134+133)/4= 132,25

Yellow: (178+175+176+174)/4=175,75

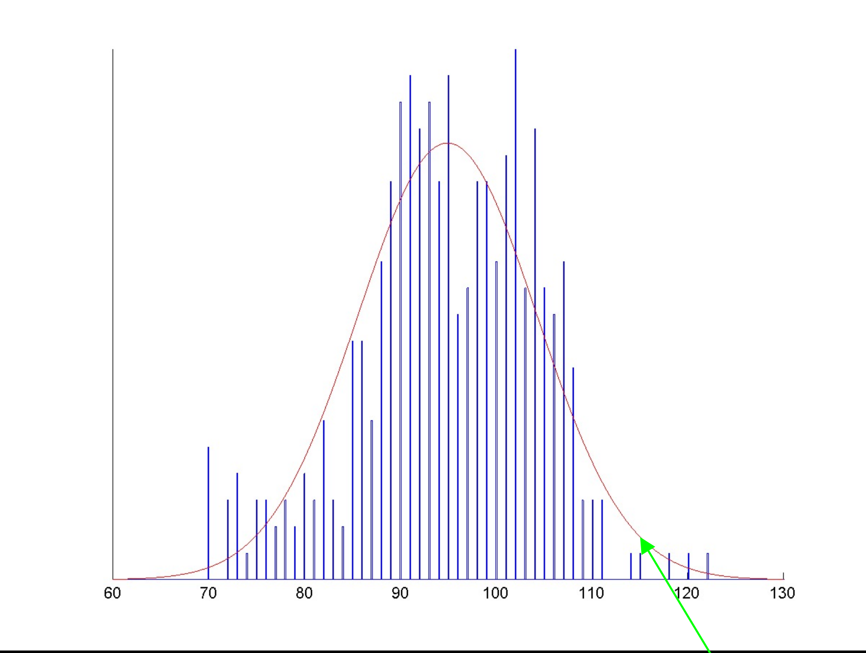
Purple: (222+220+219+221)/4=220

Blue: 158 is closes to 175,75 (yellow)= fat

To make a pixel classification an expert has selected representative regions in the image. They contain background (green), soft tissue (blue), fat (yellow), and bone (purple). The goal is to classify the pixel marked with a light blue circle. Using a minimum distance classifier it is classified as?

5	6	5	81	180	182	222	220
8	9	4	108	181	175	219	221
7	8	132	130	148	182	174	223
58	231	134	133	61	173	178	175
44	250	181	130	117	101	176	174
5	6	7	204	246	94	86	175
156	158	6	7	7	252	173	230
157	161	7	6	6	10	35	227

Parametric classification



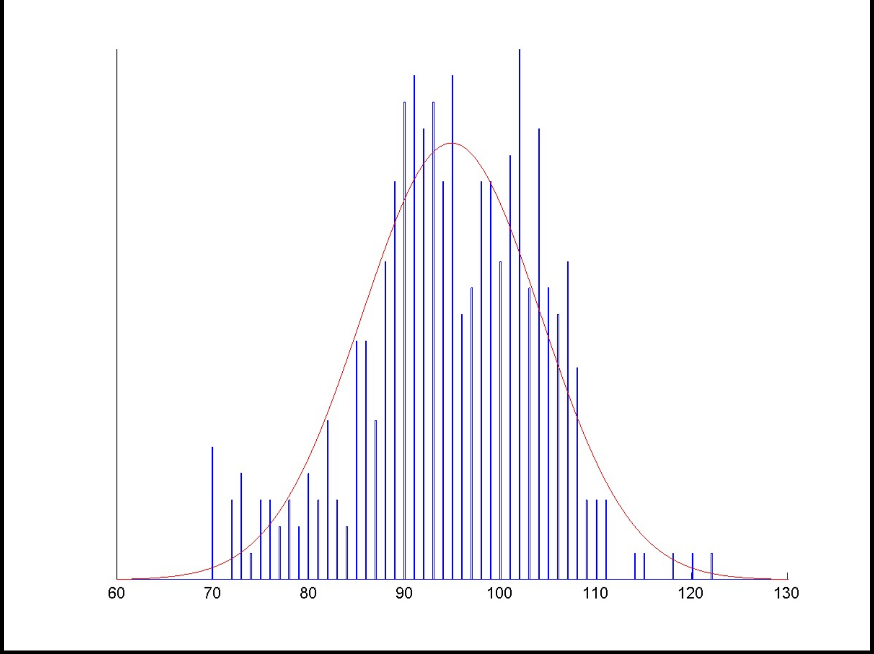
Trabecular bone

- Describe the histogram using a few parameters
- Assume a “model” describing the signal values
- Model: Gaussian/Normal distribution
  - The mean μ
  - Standard deviation σ
  - $\mathcal{N}(\mu, \sigma)$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Only two values needed

Parametric classification



Trabecular bone

Training pixel values (Belonging to one class) $v_1, v_2, \ldots, v_n$,

Estimated mean $$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} v_i$$

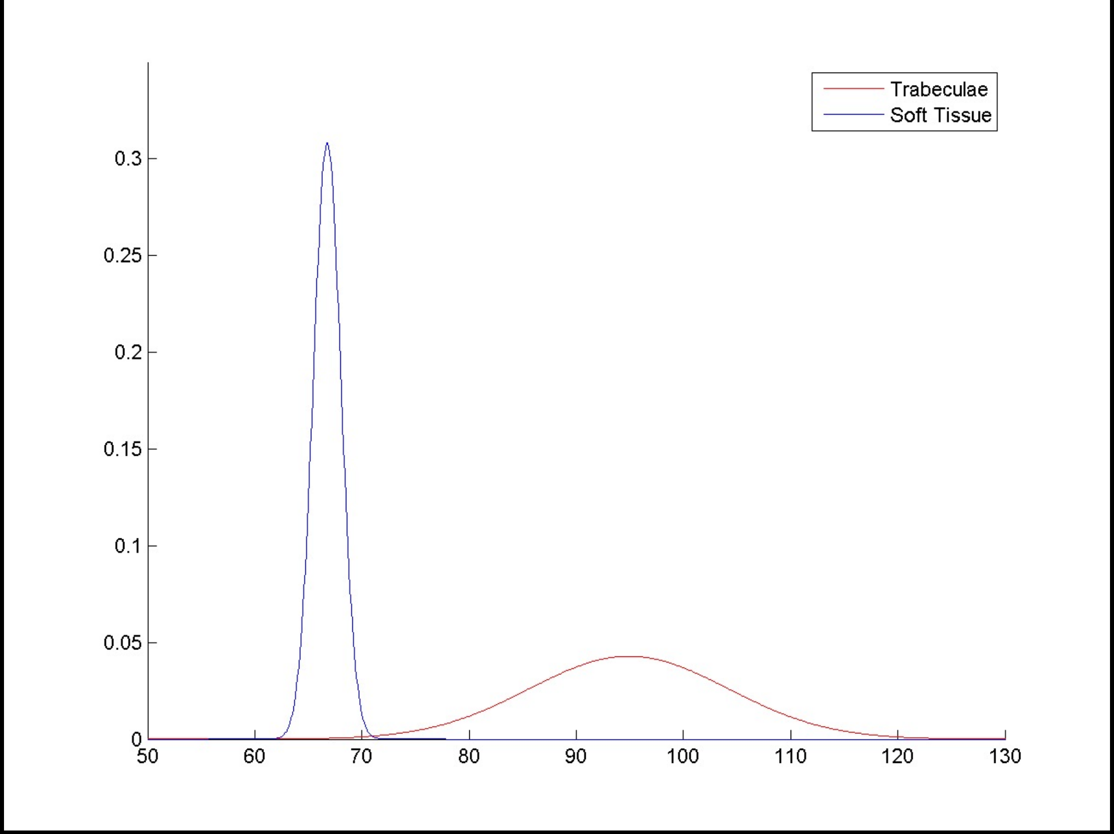
Estimated standard deviation $$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (v_i - \hat{\mu})$$

The “signal model” is a Gaussian distribution

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Parametric classification

- Fit a Gaussian to the training pixels for all classes

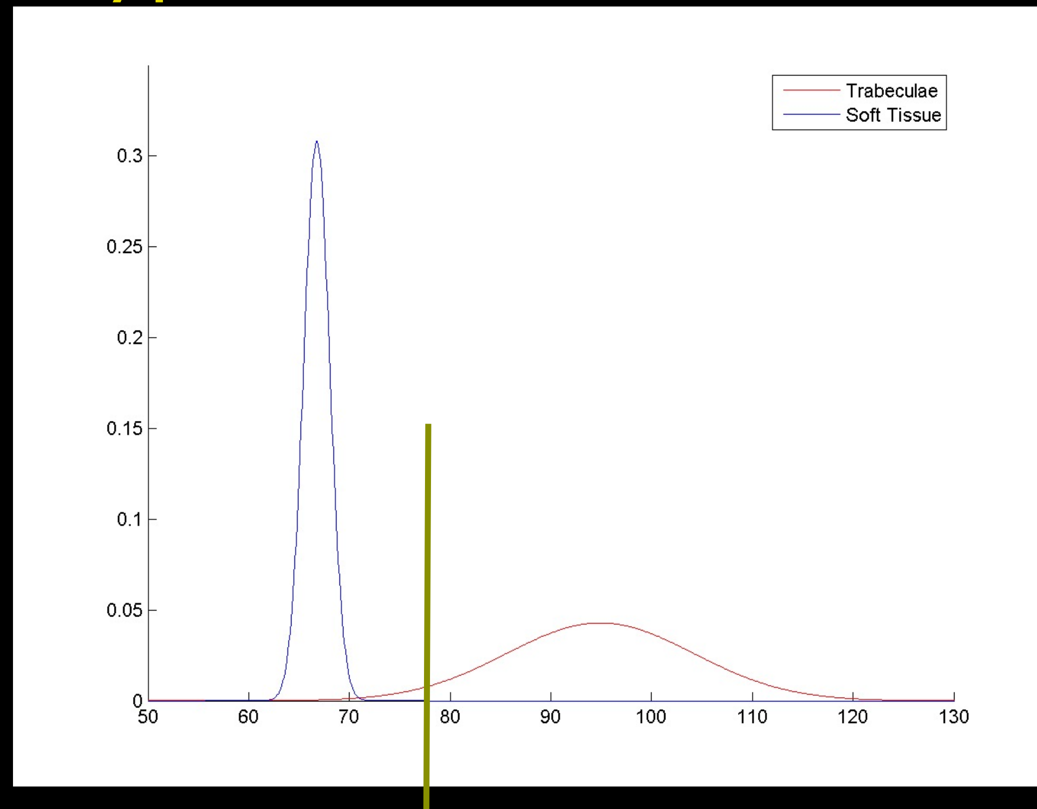


What do we see here?

What is the difference between the two classes?

Trabeculae has much higher variation in the pixel values

Quiz 3: Two tissue types – minimum distance v = 78



v = 78

Which tissue class?

A) Trabeculae

B) Soft-tissue

Solution: Minimum distance classifier

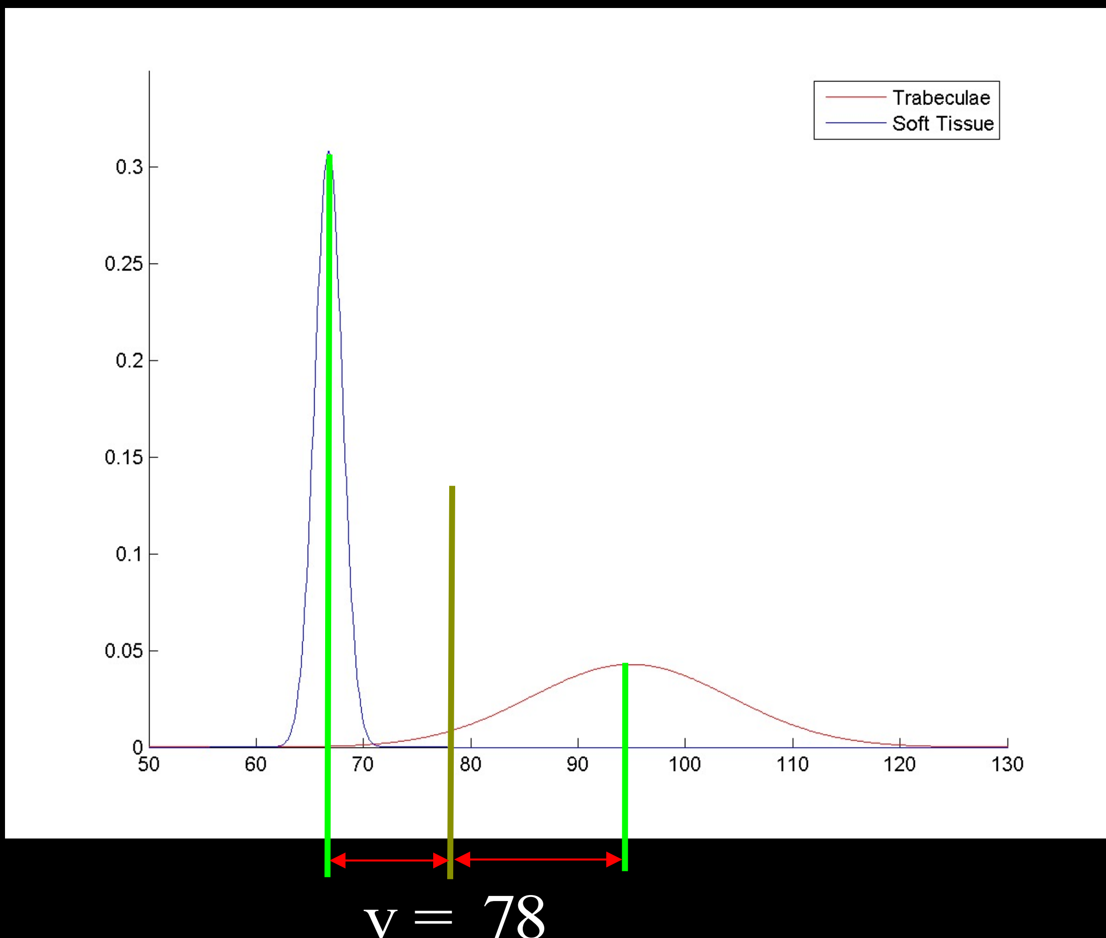
First we find the threshold, T:

B: mean(Soft Tissue)=68 and A: mean(Trabeculae)= 95

T = (95+68)/2 = 81,5

Then we classify/segment v=78: A if v>81,5 or <u>B if v< 81,5</u>

Parametric classification



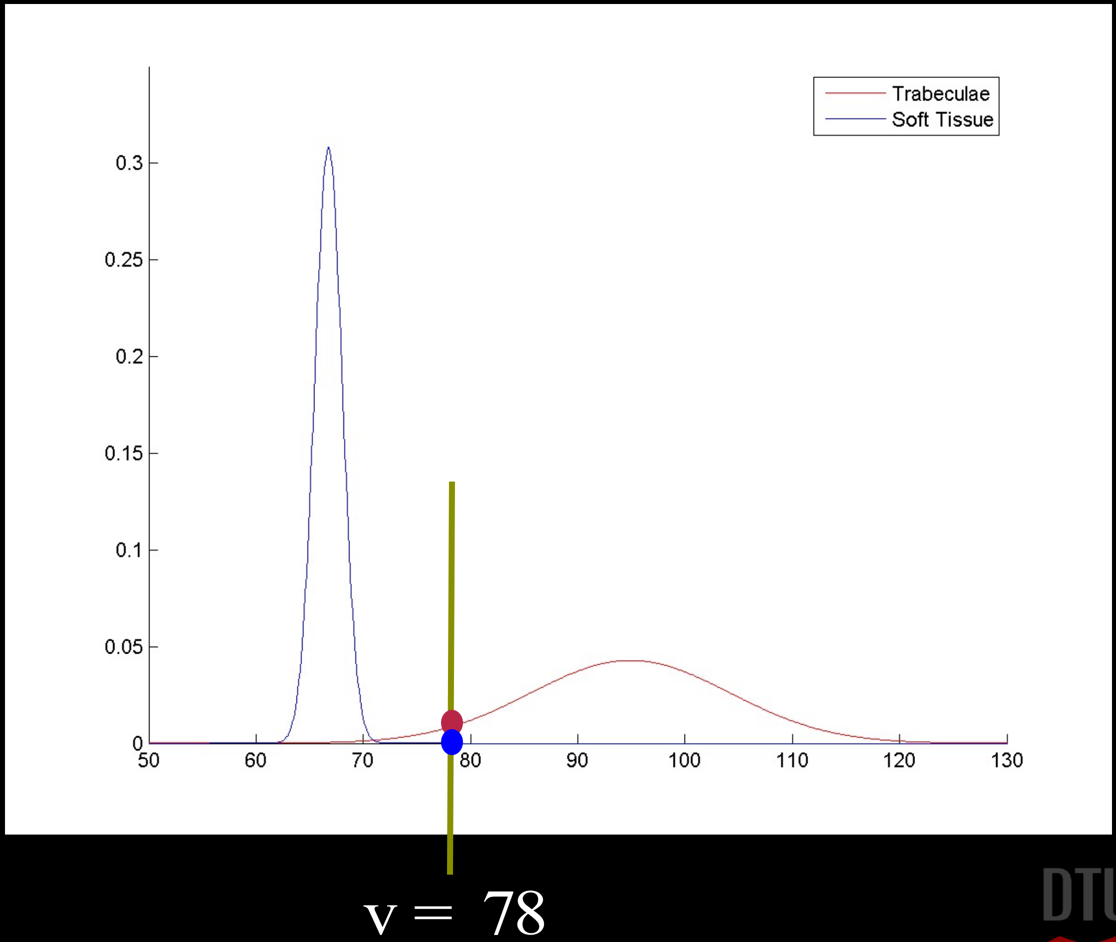
- New pixel with value 78
  - Is it soft-tissue or trabecular bone?
- Minimum distance classifier?
  - Soft-tissue
- Is that fair?
  - Soft-tissue Gaussian says "Extremely low probability that this pixel is soft-tissue"

Quiz 4: Two tissue types – parametric classification

Which tissue class?

A) Trabeculae

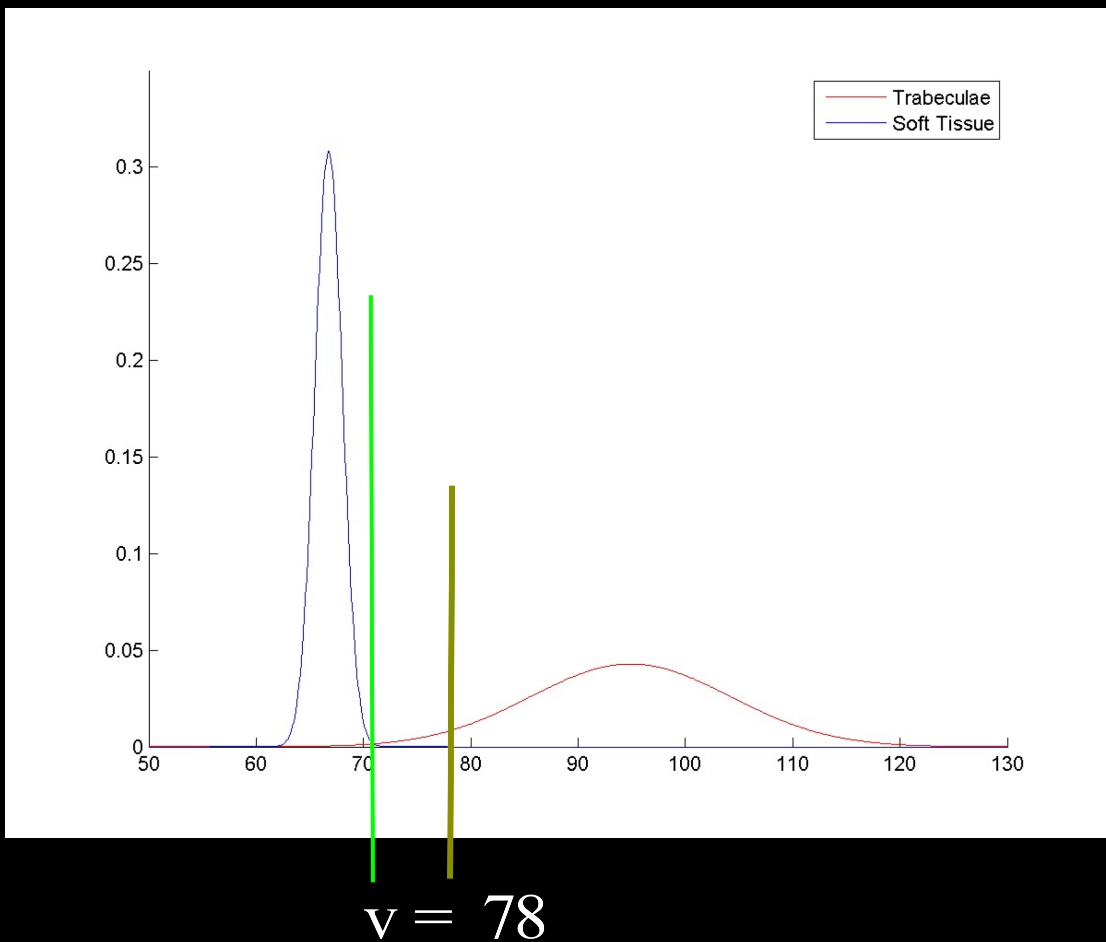
B) Soft-tissue



Solution:

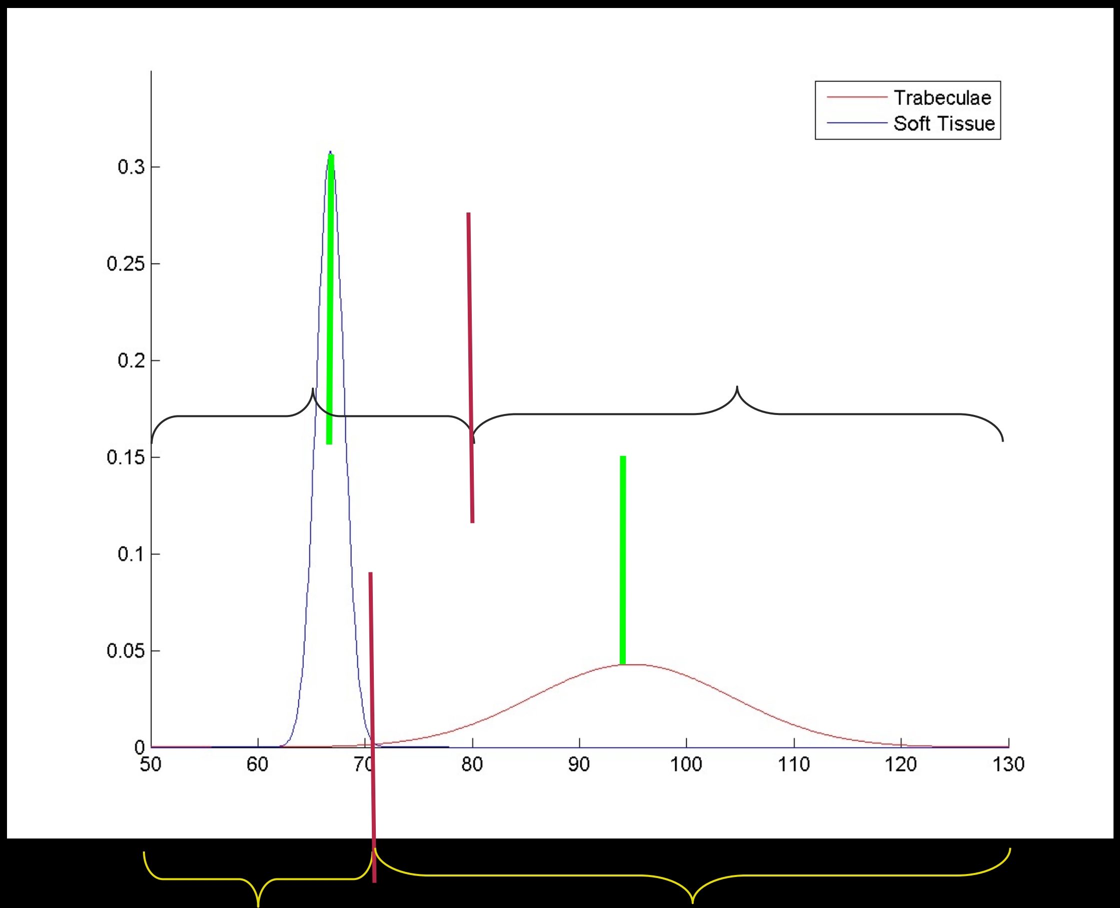
The A distribution (red) is higher than B (blue) at v=78

Parametric classification – repeat the question



- New pixel with value 78
  - Is it soft-tissue or trabecular bone?
  - Most probably trabecular bone
- Where should we set the limit?
  - Where the two Gaussians cross!

Parametric classification – ranges

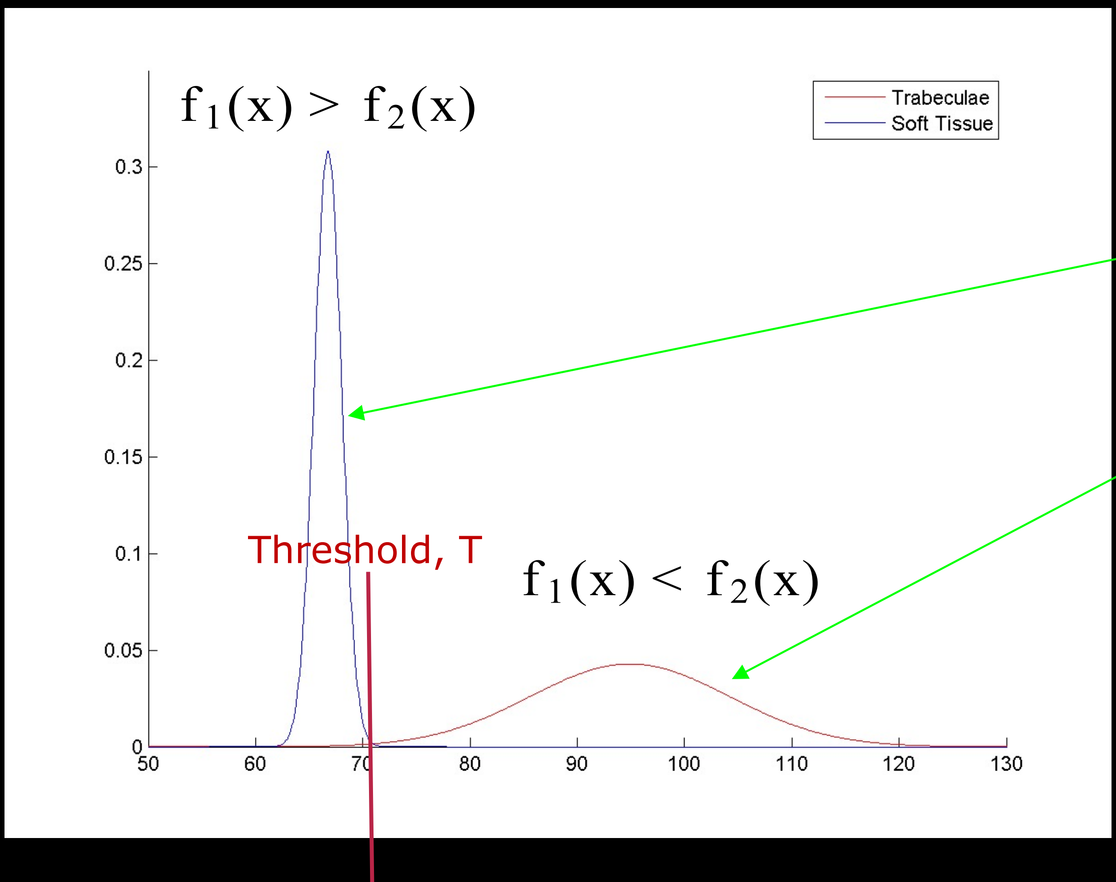


- The pixel value ranges depends on
  - The mean
  - The standard deviation
- Compared to the minimum distance classifier
  - Only the average

Parametric classification – how to

- Select training pixels for each class
- Fit Gaussians ($\mathcal{N}(\mu_i, \sigma_i)$) to each class
- Use Gaussians to determine pixel value ranges

Parametric classifier - ranges



- We want to compute where they cross

$$f_1(x) = \frac{1}{\sigma_1\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right)$$

$$f_2(x) = \frac{1}{\sigma_2\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right)$$

Create a lookup table:

- Run through all 256 possible pixel values
- Check which Gaussian is the highest
- Store the [value, class] in the table

Alternatively – analytic solution

The two Gaussians

$$\frac{1}{\sigma_1\sqrt{2\pi}}\exp\left(-\frac{(v-\mu_1)^2}{2\sigma_1^2}\right)=\frac{1}{\sigma_2\sqrt{2\pi}}\exp\left(-\frac{(v-\mu_2)^2}{2\sigma_2^2}\right)$$

Intercept at

$$v=\frac{{\sigma_1}^2\mu_2-{\sigma_2}^2\mu_1\pm\sqrt{-{\sigma_1}^2{\sigma_2}^2\left(2\,\mu_2\,\mu_1-{\mu_2}^2-2\,{\sigma_2}^2\ln\left(\frac{\sigma_2}{\sigma_1}\right)-{\mu_1}^2+2\,{\sigma_1}^2\ln\left(\frac{\sigma_2}{\sigma_1}\right)\right)}}{-{\sigma_2}^2+{\sigma_1}^2}$$

Quiz 5: Class ranges

A) [0,45],]45, 75],]75,255]

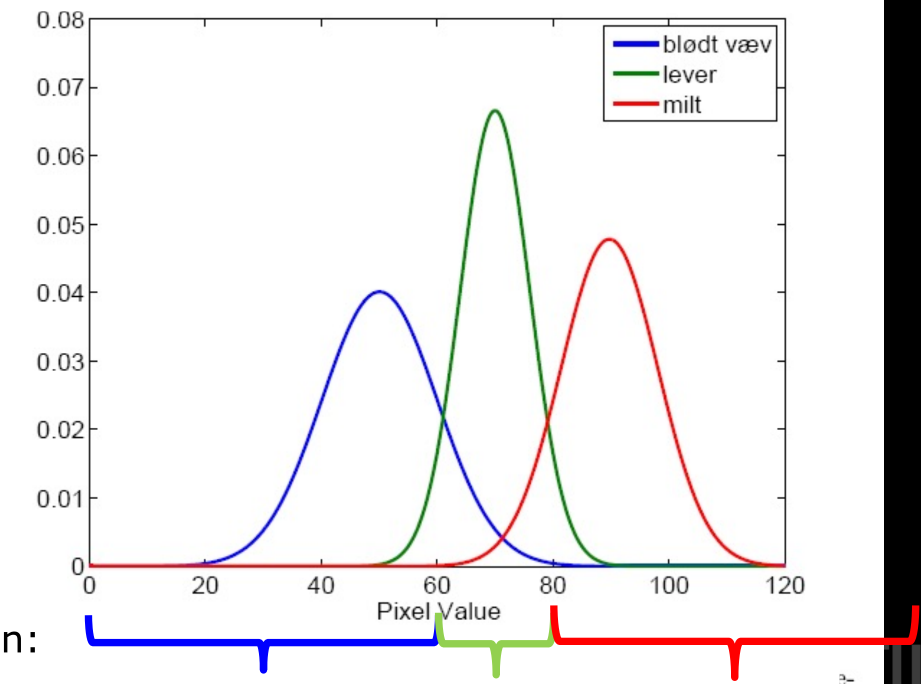
B) [40,60],]60,100],]100,140]

C) [0, 60],]60,80],]80,255]

D) [0,60],]60,100],]100,255]

E) [0,75],[75,100],]100,255]

An expert have chosen representative regions in an image that contains soft tissue, liver and spleen. The image pixel minimum and maximum values are 0 and 255. To make a parametric classification, the histograms are parameterized using Gaussian distributions as seen in the image. What are the class ranges?



Solution:

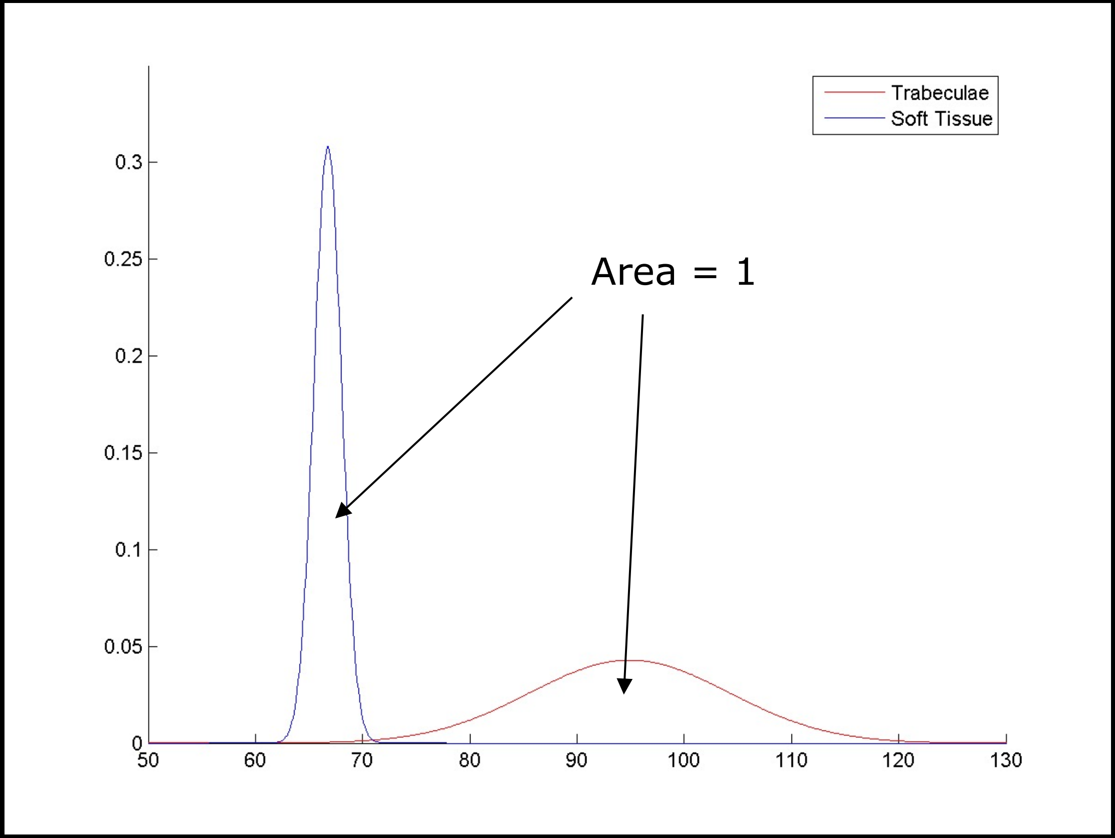
Thomas Bayes

Wikipedia

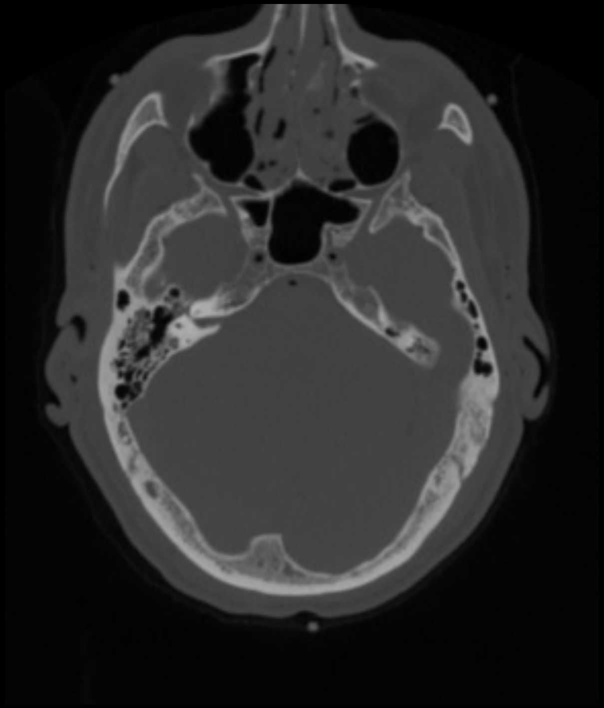
- 1702-1761
- English mathematician and Presbyterian minister
- Bayes' theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

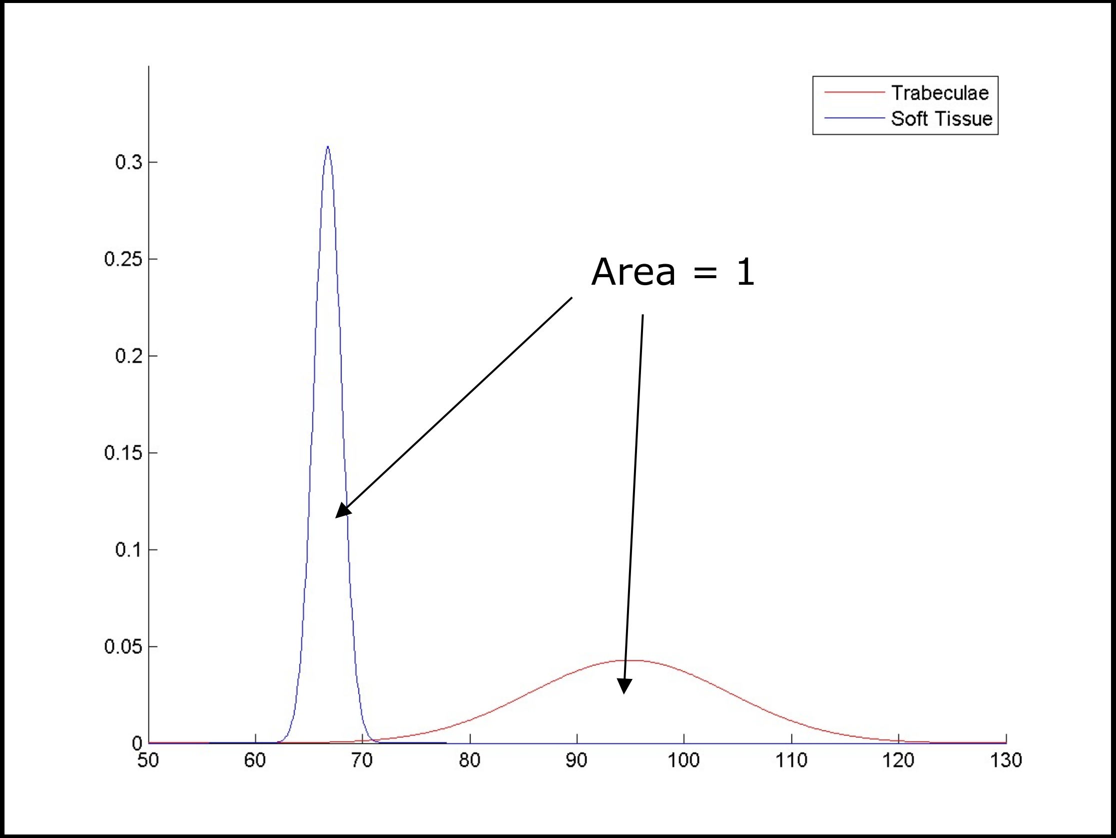
Bayesian Classification



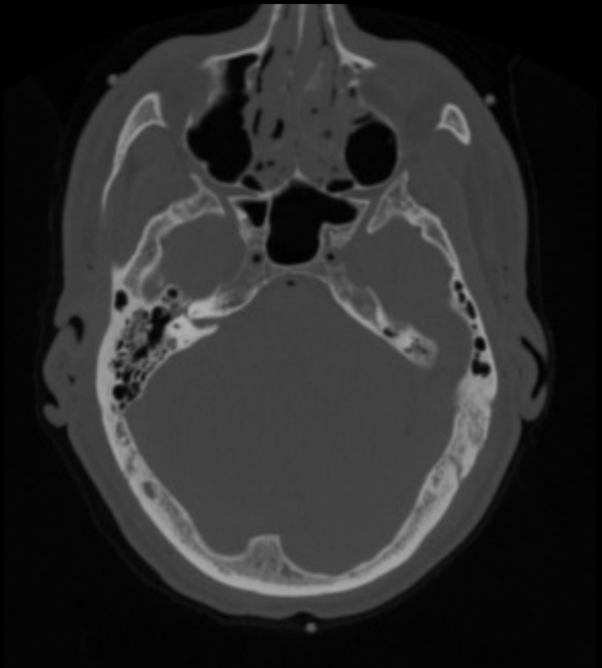
Pure parametric classifier assumes equal amount of different tissue types



Bayesian Classification



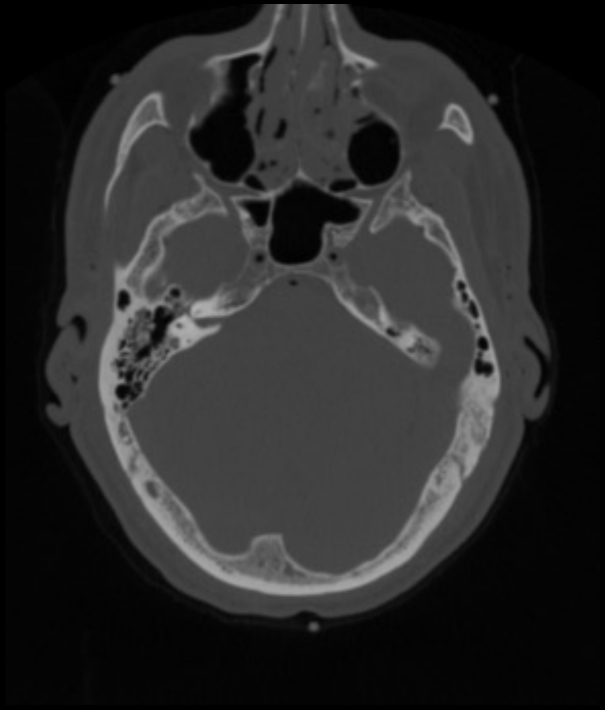
But much more soft-tissue than trabecular bone



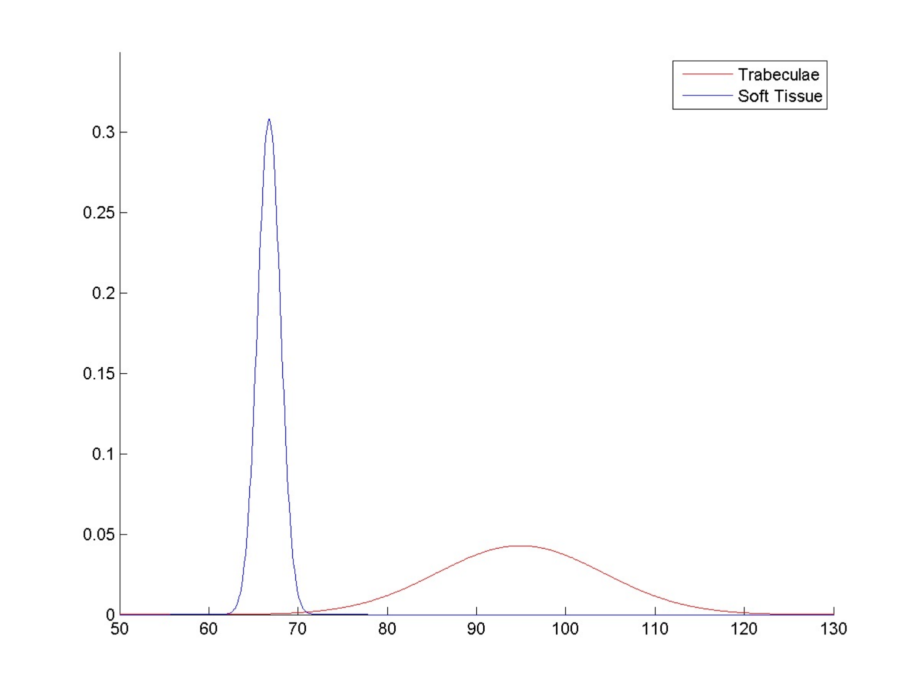
How do we handle that?

Bayesian Classification

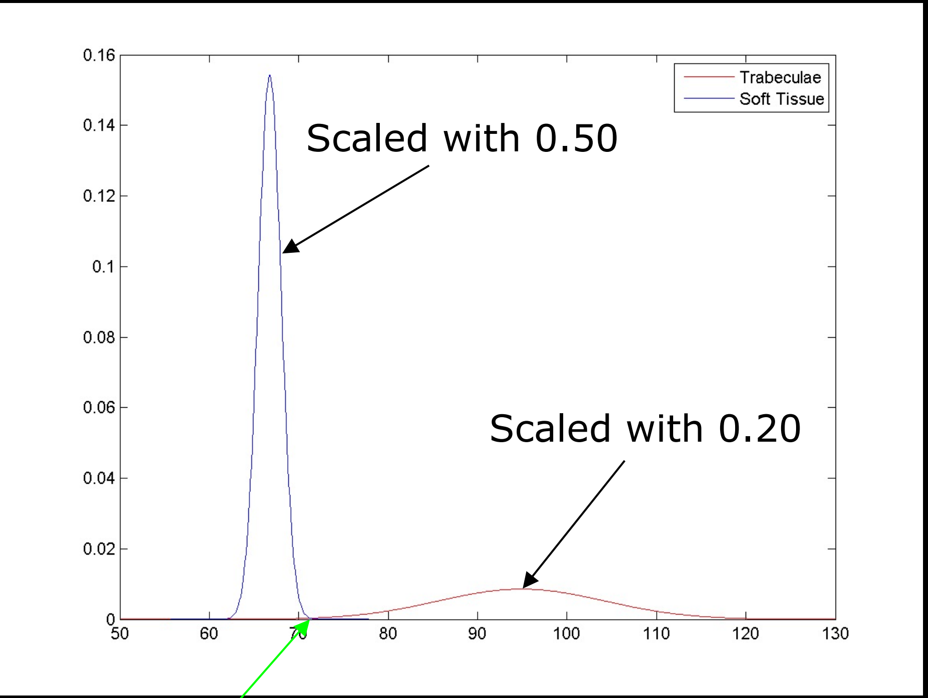
- An expert tells us that a CT scan of a head contains
  - 20% Trabecular bone
  - 50% Soft-tissue
- Picking a random pixel in the image
  - 20% Chance that it is trabecular bone
  - 50% Chance that it is soft-tissue
- How to use that?



Bayesian Classification – histogram scaling



Parametric classifier



Bayesian classifier

Little change in class border
(sometimes significant changes)

Formal definition

- Given a pixel value v
- What is the probability that the pixel belongs to class c_i

Example: If the pixel value is 78, what is the probability that the pixel is bone

$$P(c_i|v) = \frac{P(v|c_i)P(c_i)}{P(v)}$$

Formal definition

Constant – ignored from now on

$$P(c_i|v) = \frac{P(v|c_i)P(c_i)}{P(v)}$$

Formal definition

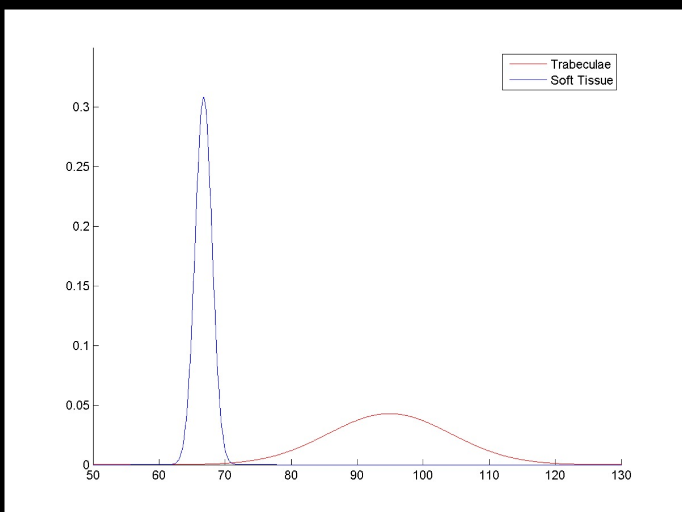
- The *a priori probability* (what is known from before)

Example: From general biology it is known that 20% of a brain CT scan is trabecular bone. Therefore P(trabecular) = 0.20

$$P(c_i|v) = \frac{P(v|c_i)P(c_i)}{P(v)}$$

Formal definition

- The *class conditional probability*
- Given a class, what is the probability of a pixel with value v

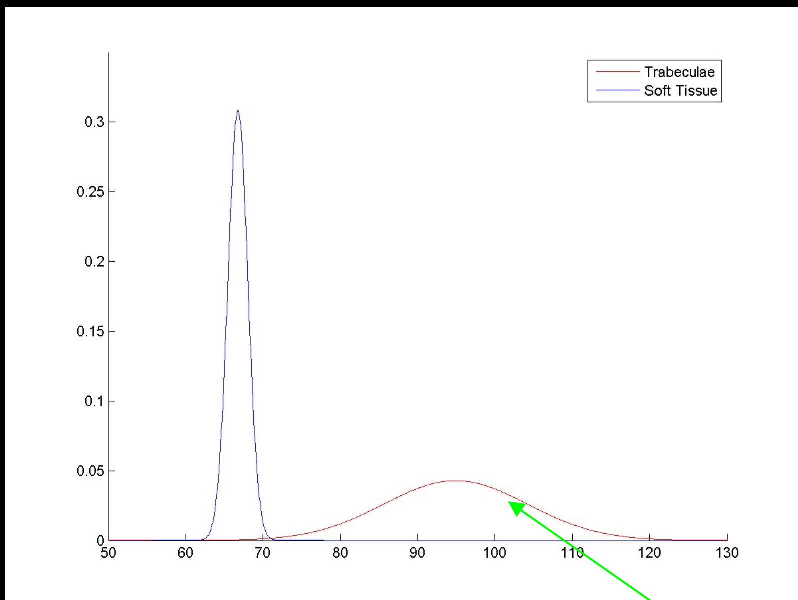


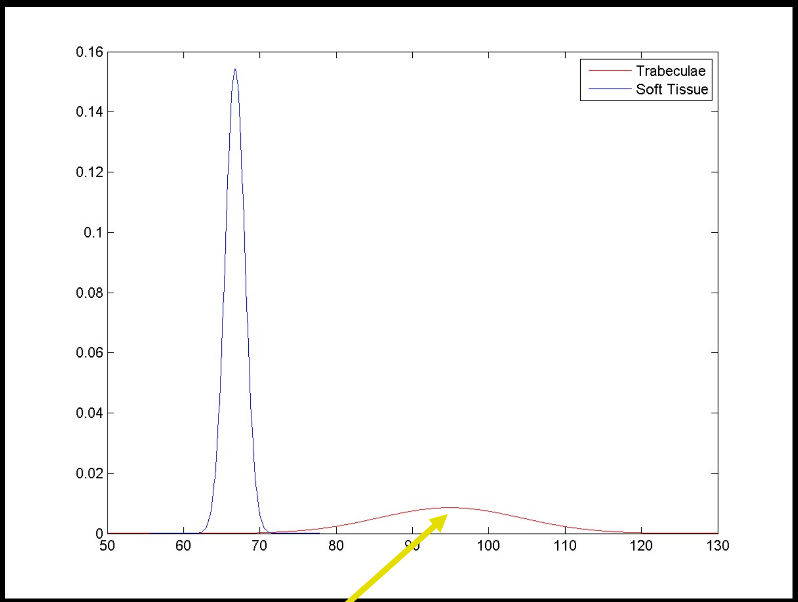
Example: If we consider class = soft-tissue. What is the probability that the pixel value is 78?

Very low

$$P(c_i|v) = \frac{P(v|c_i)P(c_i)}{P(v)}$$

Formal definition – sum up





$$P(c_i|v) = \frac{P(v|c_i)P(c_i)}{P(v)}$$

$$c_i = \text{trabeculae}$$

Bayesian classification – how to

- Select training pixels for each class
- Fit Gaussians to each class
- Ask an expert for the prior probabilities (how much there normally is in total of each type)
- For each pixel in the image
  - Compute $P(c_i|v)$ for each class (the *a posterior probability*)
  - Select the class with the highest $P(c_i|v)$

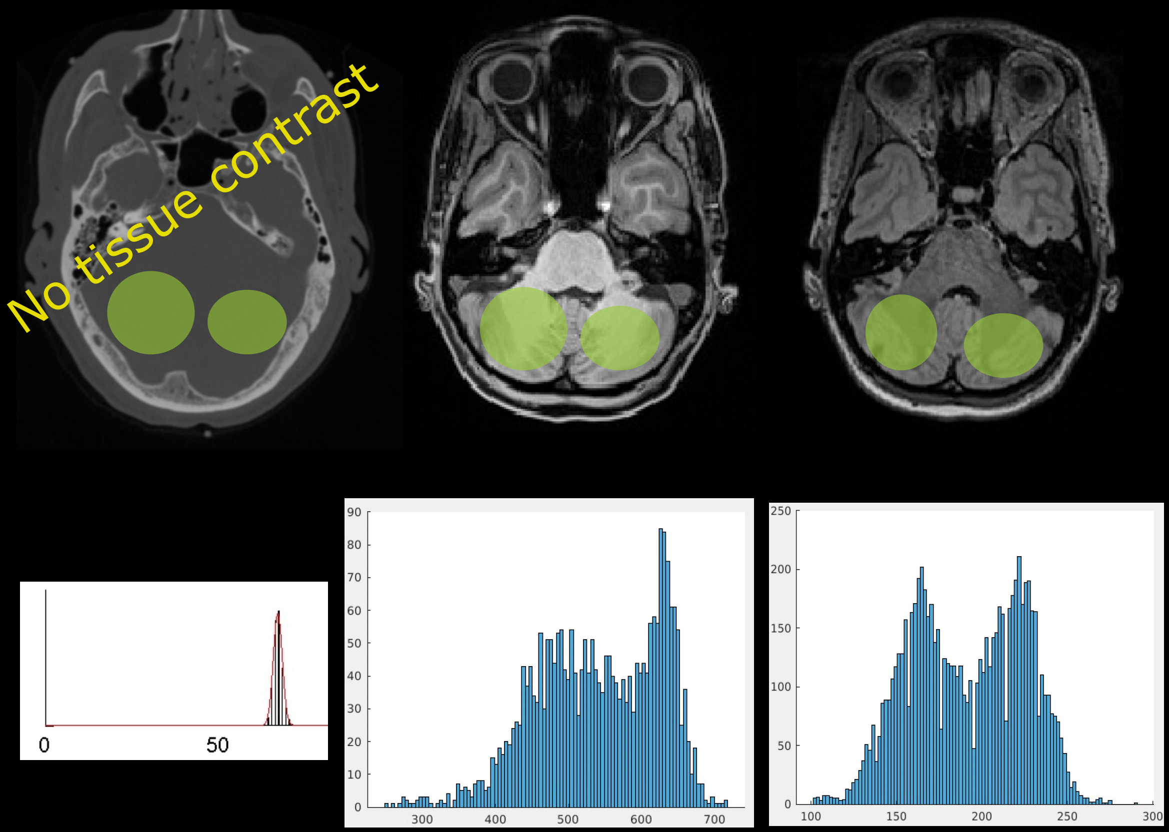
$$P(c_i|v) = \frac{P(v|c_i)P(c_i)}{P(v)}$$

When to use Bayesian classification

- The *parametric classifier* is good when there are approximately the same amount of all type of tissues
- Use *Bayesian classification* if there are very little or very much of some types
- A more general formulation for segmentation
- When going to higher dimensional feature space

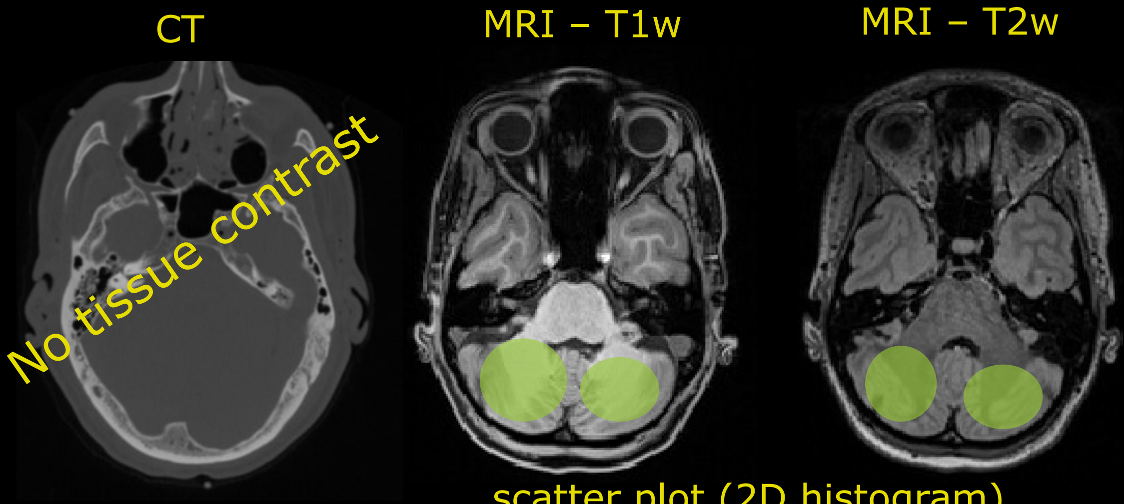
High dimensional feature space

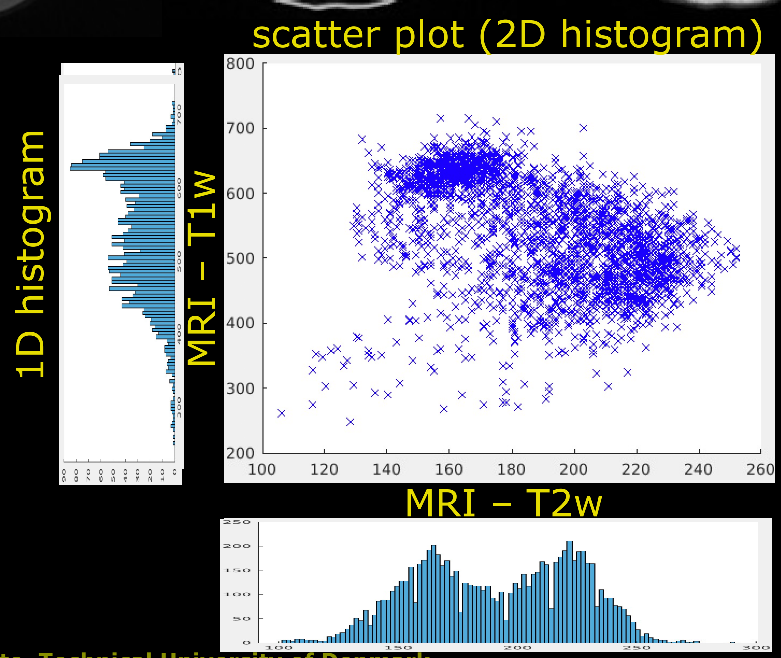
CT MRI – T1w MRI – T2w



- Combine different features input to **improve** segmentation
  - Different image modalities e.g. CT vs MRI
  - Subject groups
    - Healthy vs disease
  - Different angles of object e.g. cars

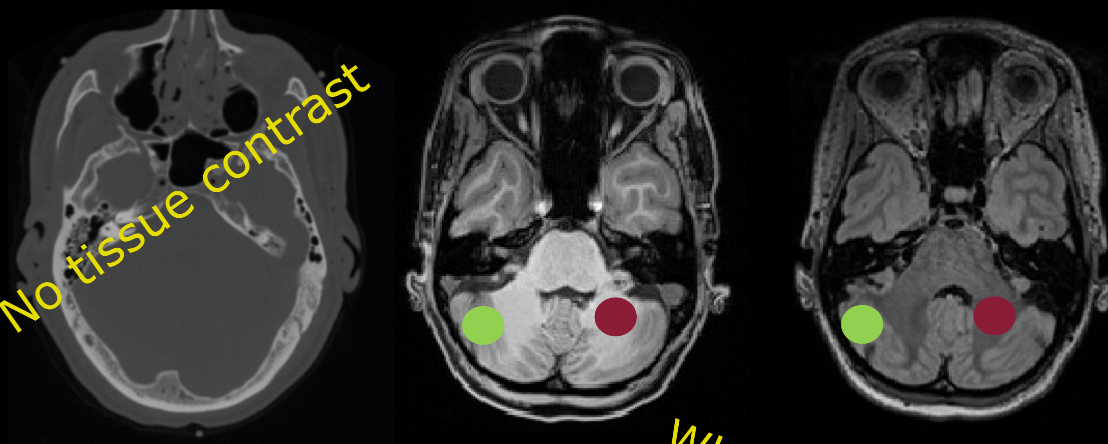
High dimensional feature space



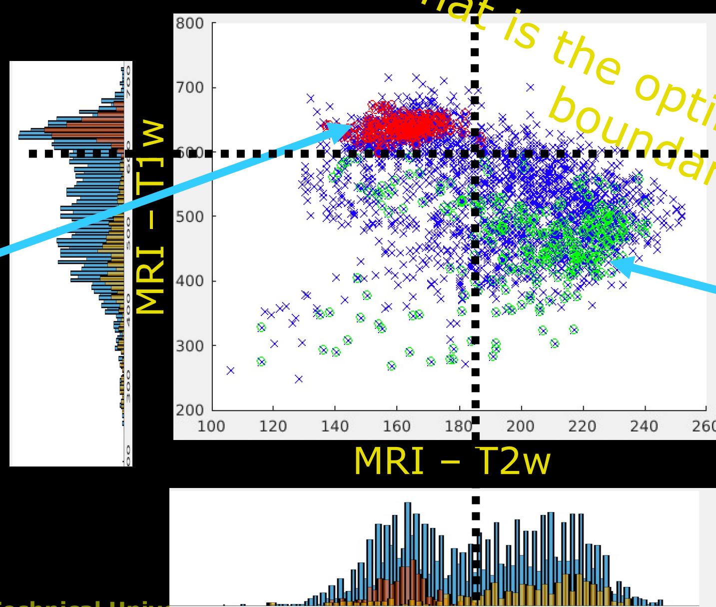


- Feature space:
  - 1D is a histogram
  - 2D is a scatterplot i.e. 2D histogram
  - >2D is bit more complicated to show
- Here we stay in 2D feature space for optimal visualisation

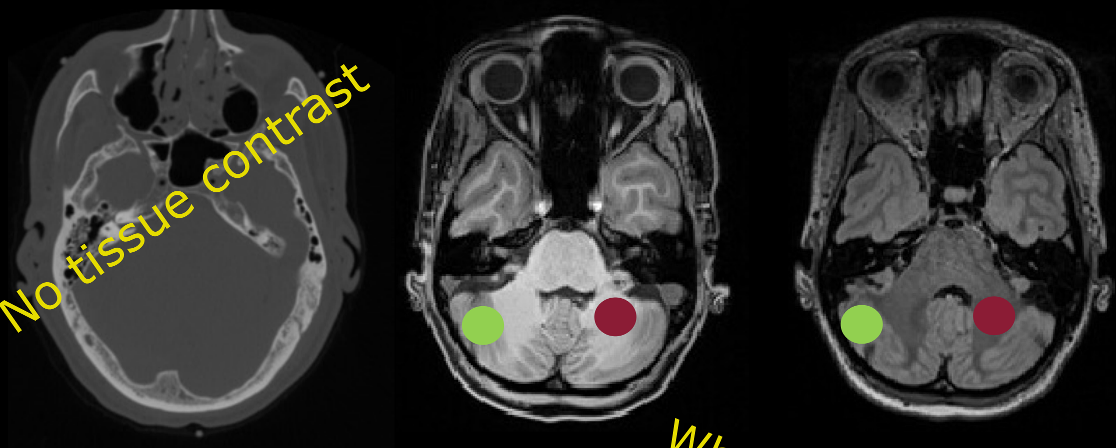
High dimensional feature space

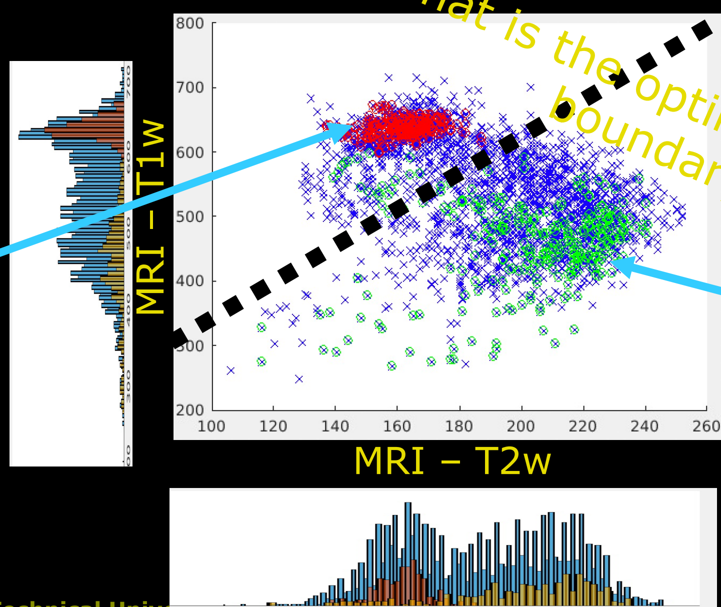


- Segmentation with more feature inputs
- To train our "model" with class examples
  - Draw tissue specific regions for each class
  - Class 1 and Class 2
  - Tissue type 1 and type 2
- Segmentation:
  - How to define the decision boundaries
  - 1D vs 2D



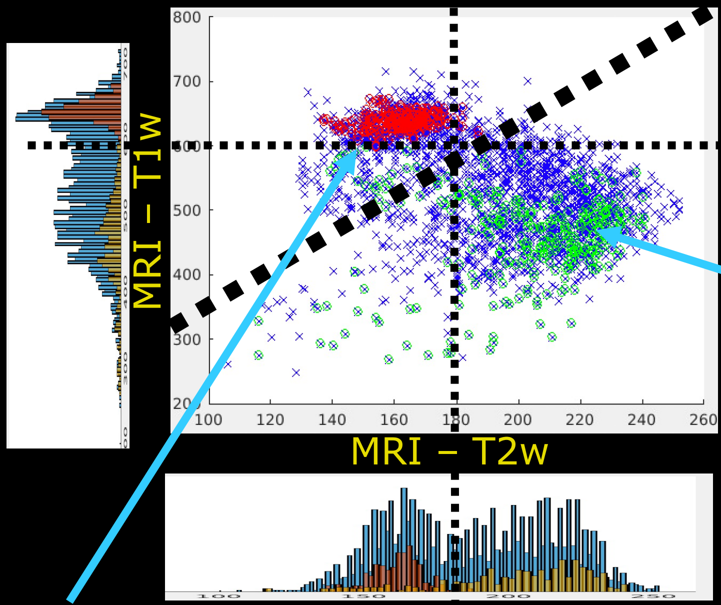
High dimensional feature space





- Segmentation with more feature inputs
- To train our "model" with class examples
  - Draw tissue specific regions for each class
  - Class 1 and Class 2
  - Tissue type 1 and type 2
- Segmentation:
  - How to define the decision boundaries
  - 1D vs 2D

Decision boundary



Class 1
$\mathcal{N}(\mu_1, \Sigma_1)$

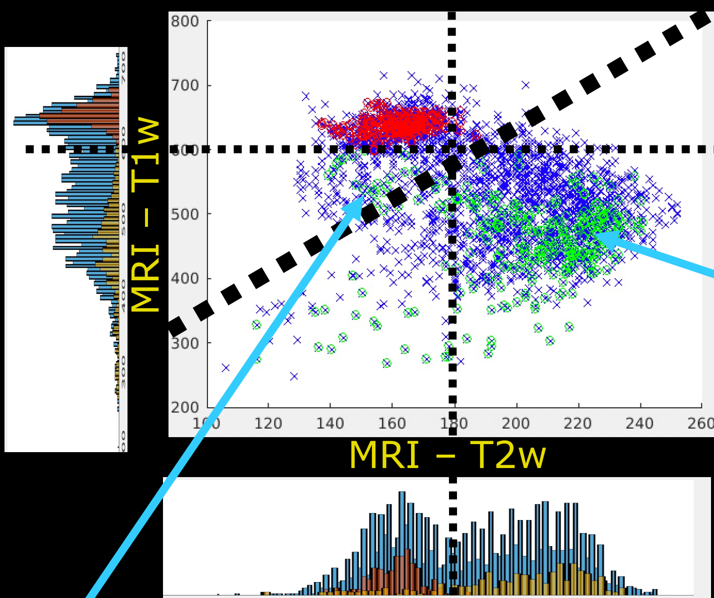
Class 2
$\mathcal{N}(\mu_2, \Sigma_2)$

- 2D feature space
  - Better class separation vs 1D?
- Model assumption
  - Type of distribution?
- Intensity histograms per class looks Gaussian like?
  - We assume Gaussian distributions: $\mathcal{N}(\mu_i, \Sigma_i)$
- Optimal decision boundary using Bayes theorem:
  - Likelihood ratio for belonging to C2:

$$\frac{P(C2|\boldsymbol{x})}{P(C1|\boldsymbol{x})} > T$$

Optimal decision boundary "T" – How to?

Decision boundary



- We wish to find T using Bayes:

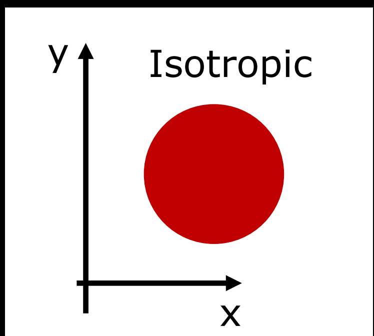
$$\frac{P(C2|\boldsymbol{x})}{P(C1|\boldsymbol{x})} > T$$

- The posterior probability
  - $P(Ci|\boldsymbol{x}) = P(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\boldsymbol{P}_{\boldsymbol{Ci}}$
- The class specific Gaussian model

$$P(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i,) = K_i \exp((\boldsymbol{x} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i))$$

  - Data points:
    - $\boldsymbol{x}_i = [x1, x2]^T$
  - Training set:
    - $\boldsymbol{t}_{x \in \boldsymbol{C1}} = 0$ and $\boldsymbol{t}_{x \in \boldsymbol{C2}} = 1$
  - The class mean of training
    - $\boldsymbol{\mu}_i = \frac{1}{N} \sum_{n \in Ci} x_n$
  - The covariance matrix of training
    - $\boldsymbol{\Sigma}_i = (\boldsymbol{x} - \boldsymbol{\mu}_i)^T (\boldsymbol{x} - \boldsymbol{\mu}_i)$

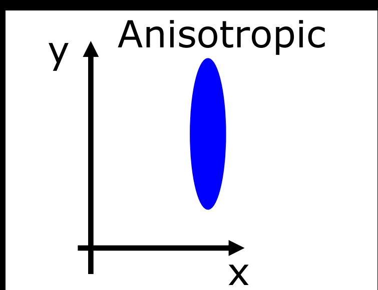
Gaussian in 2D: The covariance matrix



Rotational invariant

$$\Sigma = \begin{bmatrix} \sigma_{xx} & 0 \\ 0 & \sigma_{yy} \end{bmatrix}$$

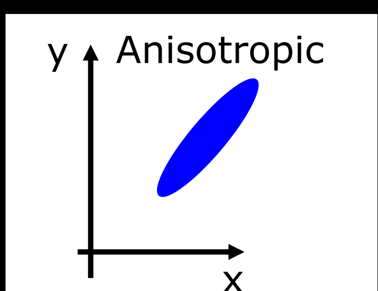
$\sigma_{xx} = \sigma_{yy}$



Aligned with coordinate system

$$\Sigma = \begin{bmatrix} \sigma_{xx} & 0 \\ 0 & \sigma_{yy} \end{bmatrix}$$

$\sigma_{xx} \neq \sigma_{yy}$



Not aligned with coordinate system

$$\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix}$$

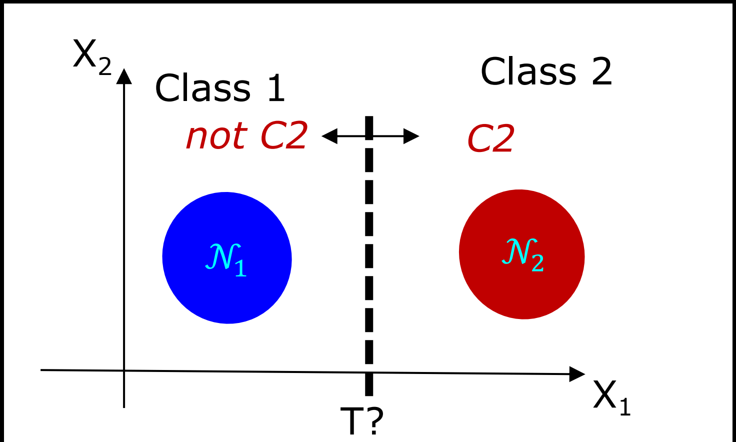
QUICK REFRESH:

- The covariance matrix:

  $$\boldsymbol{\Sigma_i} = (\boldsymbol{x} - \boldsymbol{\mu_i})^T (\boldsymbol{x} - \boldsymbol{\mu_i})$$

- Expresses the orientation of anisotropic variance in relation to coordinate system

Back to the Decision boundary



- Classifier: If ***x*** belongs to C_2 or not:

$$\frac{P(C2|\boldsymbol{x})}{P(C1|\boldsymbol{x})} > T$$

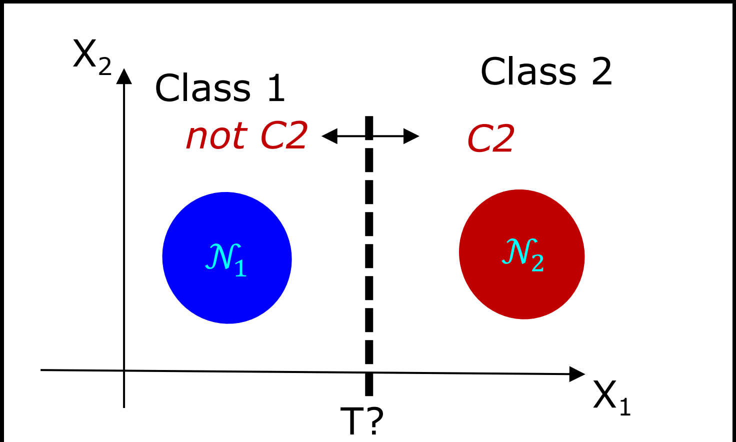
- Taking the logarithmn

$$ln\big(P(C2|x)\big) - ln\big(P(C1|x)\big) > T$$

$\mathcal{N}_1(\mu_1, \Sigma_1)$ $\mathcal{N}_2(\mu_2, \Sigma_2)$

Inspiration: https://en.wikipedia.org/wiki/Linear_discriminant_analysis

Back to the Decision boundary



$\mathcal{N}_1(\mu_1, \Sigma_1)$ $\mathcal{N}_2(\mu_2, \Sigma_2)$

- Classifier: If $\boldsymbol{x}$ belongs to C_2:

$$\frac{P(C2|\boldsymbol{x})}{P(C1|\boldsymbol{x})} > T$$

- Taking the logarithmn

$$ln\left(P(C2|x)\right) - ln\left(P(C1|x)\right) > T$$

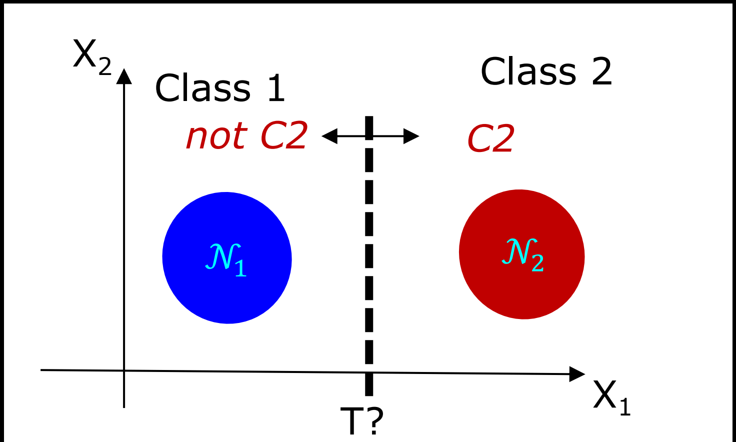
- Where the log posterior probability:

$$\ln(P(\boldsymbol{Ci}|\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i) + \ln(K_i) + \ln(Pi)$$

- P_i is the prior probability for class C_i

Inspiration: https://en.wikipedia.org/wiki/Linear_discriminant_analysis

Back to the Decision boundary



$\mathcal{N}_1(\mu_1, \Sigma_1)$ $\mathcal{N}_2(\mu_2, \Sigma_2)$

- Classifier: If $\boldsymbol{x}$ belongs to C_2 or not:

$$\frac{P(C2|\boldsymbol{x})}{P(C1|\boldsymbol{x})} > T$$

- Taking the logarithmn

$$ln\big(P(C2|x)\big) - ln\big(P(C1|x)\big) > T$$

- Where the log posterior distribution:

$$\ln(P(\boldsymbol{Ci}|\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \Sigma_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i) + \ln(K_i) + \ln(Pi)$$

- P_i is the prior probability for class C_i

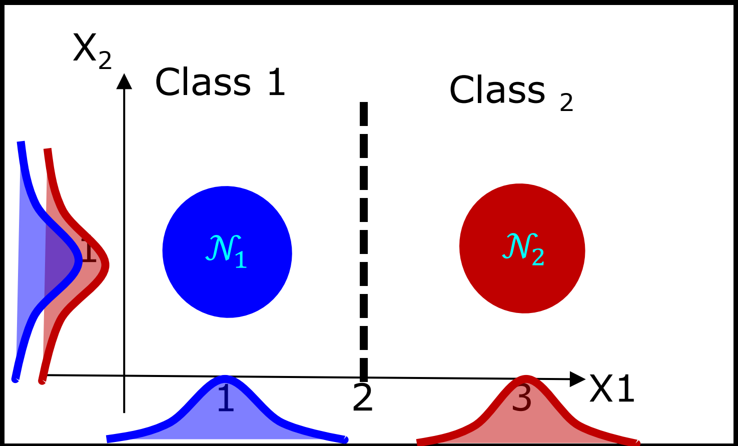
- Inserting and assuming homoscedasticity ($\Sigma_1 = \Sigma_2 = \Sigma_0$) we have a Linear discrimenant Analysis (LDA) classifier model (reorganise the expression)

$$ln\frac{P2}{P1} + \frac{1}{2}(\boldsymbol{\mu_2} + \boldsymbol{\mu_1})^T \Sigma_0^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1}) - \boldsymbol{x}^T \Sigma_0^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1}) > \boldsymbol{T}$$

- We train the classifier to find T with examples obtained from the two distributions N1 and N2

Inspiration: https://en.wikipedia.org/wiki/Linear_discriminant_analysis

Quiz 6 - LDA - Optional Decision boundary



- Define T for x belonging to C_2:

$$\frac{P(C2|\boldsymbol{x})}{P(C1|\boldsymbol{x})} > T$$

- Using Linear Discriminat Analysis (LDA):

$$ln\frac{P2}{P1} + \frac{1}{2}(\boldsymbol{\mu_2} + \boldsymbol{\mu_1})^T \Sigma_0^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1}) - \boldsymbol{x}^T \Sigma_0^{-1}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1}) > \boldsymbol{T}$$

$\Sigma_1 = \Sigma_2 = \Sigma_0 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ Prior probabilities: P1=P2=0,5

At which x value is the optimal decision boundary, T found i.e. using $\ln(\frac{P(C2|\boldsymbol{x})}{P(C1|\boldsymbol{x})})$?

A) 1,5
B) 2,7
C) 2,3
D) 2,0
E) 0,7

Solution – we see that x for optimal T is a threshold only along X1 i.e. a solution in 1D:

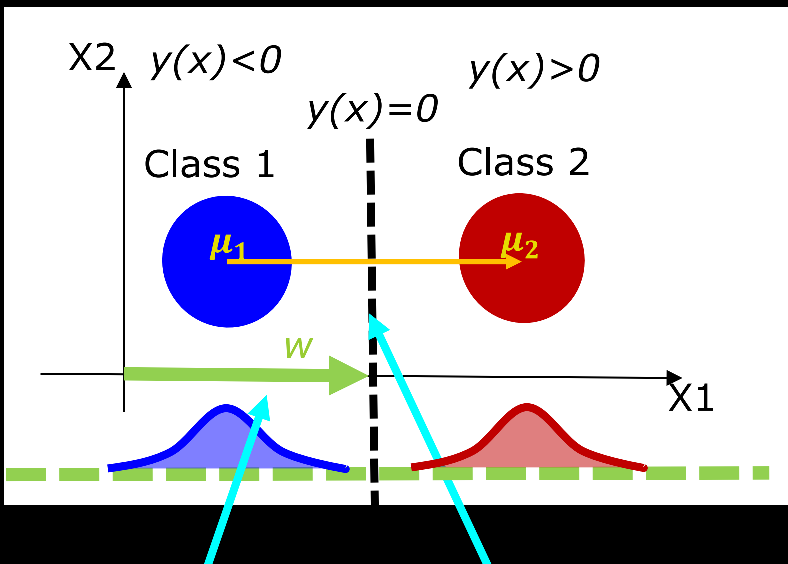
$$ln\frac{P2}{P1} + \frac{1}{2}(\boldsymbol{\mu_2} + \boldsymbol{\mu_1})\frac{(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})}{\sigma_0} = \text{x1}\frac{(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})}{\sigma_0}$$

$$ln\frac{0,5}{0,5} + \frac{1}{2}(3+1)\frac{(3-1)}{2} = x\frac{(3-1)}{2}$$

x1=2

X2= all values

Hyper plan and projections in feature space



Hyper plan

- *w* projects in the class mean direction i.e. the weight vector
- *w is normal to the hyper plan yi(x)=0*
- $x^T w$ is a *dot product i.e.* x and c are projected onto *w* ($a^T b = \|a\|\|b\|\cos(\theta)$)

- We wish to predict the C_2:

$$\frac{P(C2|x)}{P(C1|x)} > T$$

- The LDA function for C_2

$$\underbrace{\underbrace{ln\frac{P1}{P2} + \frac{1}{2}(\mu_2+\mu_1)^T}_{c}\underbrace{\Sigma_0^{-1}(\mu_2-\mu_1)}_{w}}_{w_0} - x^T\underbrace{\Sigma_0^{-1}(\mu_2-\mu_1)}_{w} > T$$

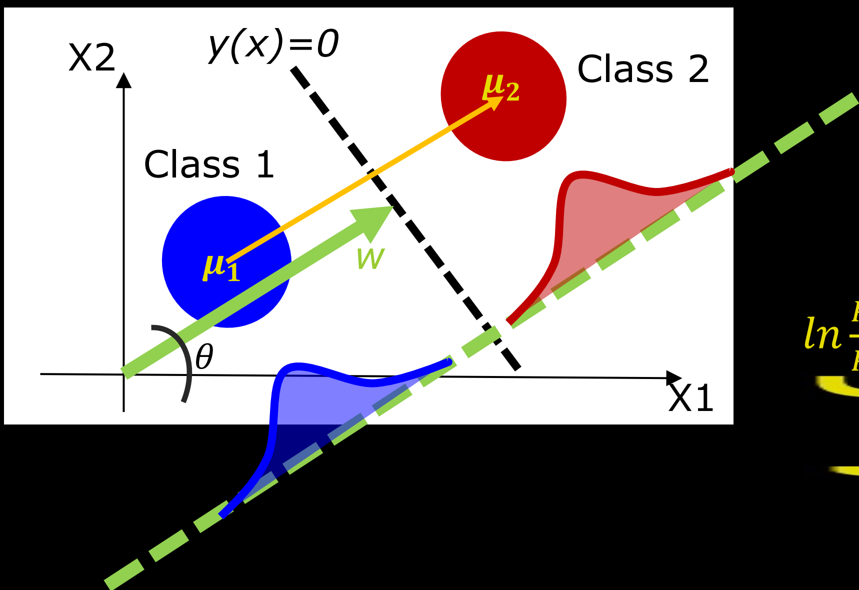
- The linear discriminat function

$$y_{C\in 2}(x) = \mathbf{x}^T\mathbf{w} + \mathbf{w_0}$$

-where negtive Wo is the threshold

- x is assigned to C2 if $y_{C\in 2}(x) > 0$
- $y_i(x) = 0$ defines a *hyper plan for* the dicision boundary

Hyper plan and projections in feature space



- We wish to predict the C_2:

$$\frac{P(C2|\boldsymbol{x})}{P(C1|\boldsymbol{x})} > T$$

- The LDA function for C_2

$$\underbrace{\underbrace{\ln\frac{P1}{P2} + \frac{1}{2}(\boldsymbol{\mu_2}+\boldsymbol{\mu_1})^T}_{\mathbf{c}}\underbrace{\Sigma_0^{-1}(\boldsymbol{\mu_2}-\boldsymbol{\mu_1})}_{\mathbf{w}}}_{\mathbf{w_0}} - \boldsymbol{x}^T\underbrace{\Sigma_0^{-1}(\boldsymbol{\mu_2}-\boldsymbol{\mu_1})}_{\mathbf{w}} > T$$

- The linear discriminat function

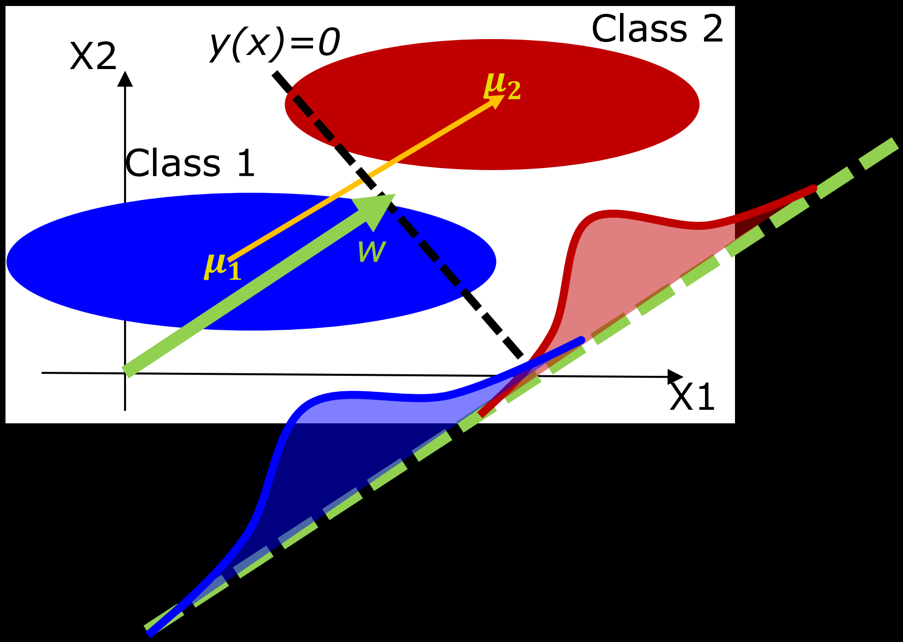
$$y_{C\in 2}(\boldsymbol{x}) = \mathbf{x}^T\mathbf{w} + \mathbf{w_0}$$

-where negtive Wo is the threshold

- x is assigned to C2 if $y_{C\in 2}(x) > 0$
- $y_i(x) = 0$ defines a *hyper plan for* the dicision boundary

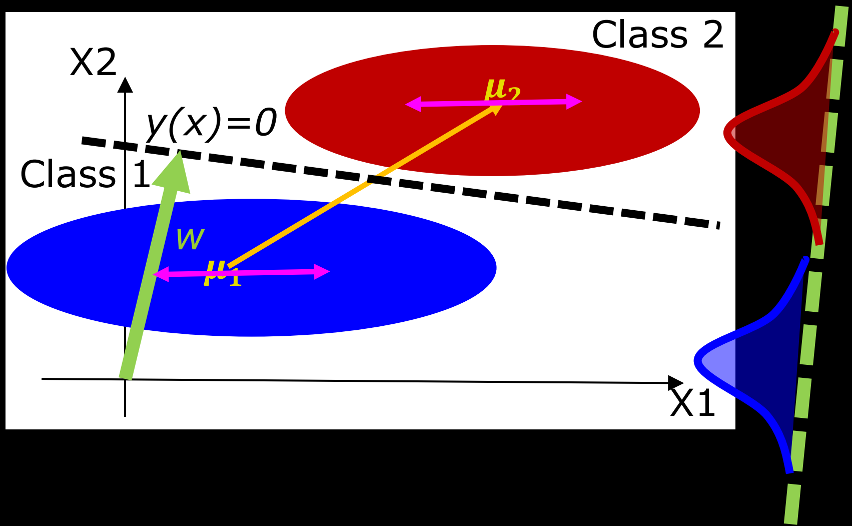
- *w* projects in the class mean direction i.e. the weight vector
- *w is normal to the hyper plan yi(x)=0*
- $x^T w$ is a *dot product i.e.* x and c are projected onto *w* $(\boldsymbol{a}^T\boldsymbol{b} = \|\boldsymbol{a}\|\|\boldsymbol{b}\|\cos(\theta))$

Hyper plan and projections in feature space



- If convariance is *anisotropic* ie not identity matrix.
  - Not optimal placement of hyper plan based on mean seperation
  - Not optimal segmentation results
  - Hyper plan does not ensure optimal seperation!
- To improve the seperation
  - We need to adjust the weigth vector ***w***

Hyper plan and projections in feature space



Optimal class separation:

- The *weight vector* ***w*** now account for both for class means and variances

- Fisher's linear discrimiment:
  - Uses: *between-class (means) covariance*:

  $$\boldsymbol{S_B} = (\boldsymbol{\mu_2} - \boldsymbol{\mu_1})^T(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})$$

  - and: optimise *(total) within-class covariance*

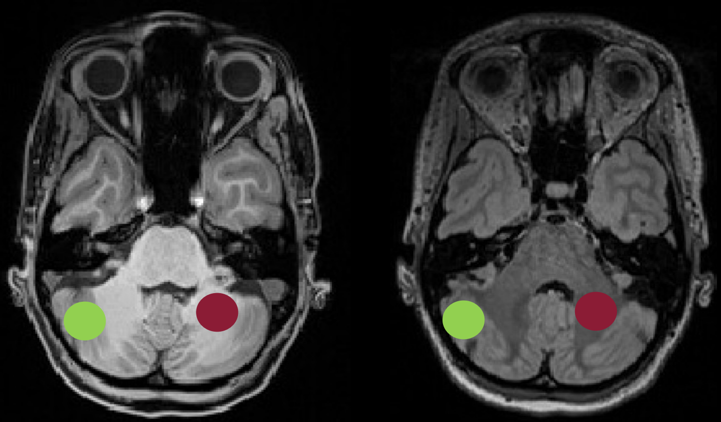
  $$\boldsymbol{S_W} = \boldsymbol{\Sigma_1} + \boldsymbol{\Sigma_2}$$

- Find projection *w* using a cost function:
  - $J(w) = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}}$ and differentiate: $\frac{\partial J(w)}{\partial w}=0$
  - which gives (simple solution):

  $$\boldsymbol{w} \propto \boldsymbol{S_W^{-1}}(\boldsymbol{\mu_2} - \boldsymbol{\mu_1})$$

Segmentation of brain data using LDA

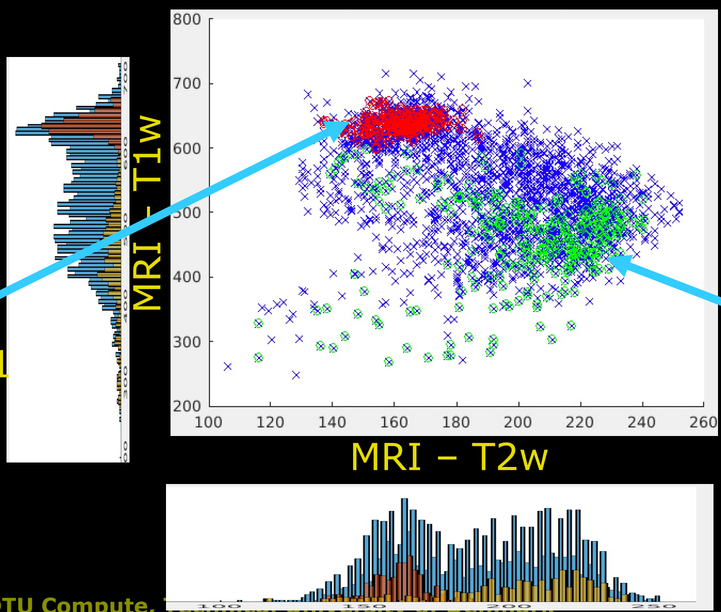
MRI – T1w MRI – T2w



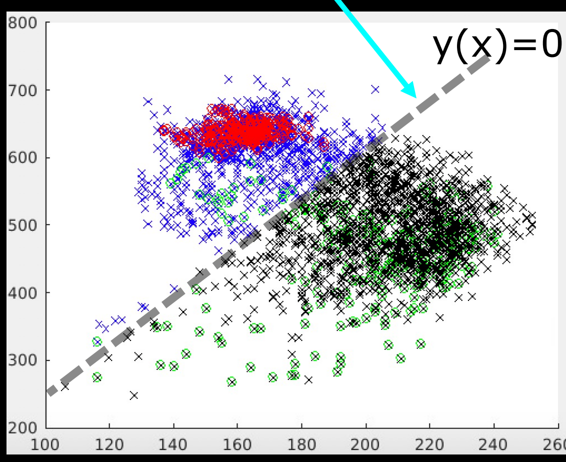
- Fisher's linear discriminent
- Use Matlab function:
  - LDA.m

Found Hyper plan

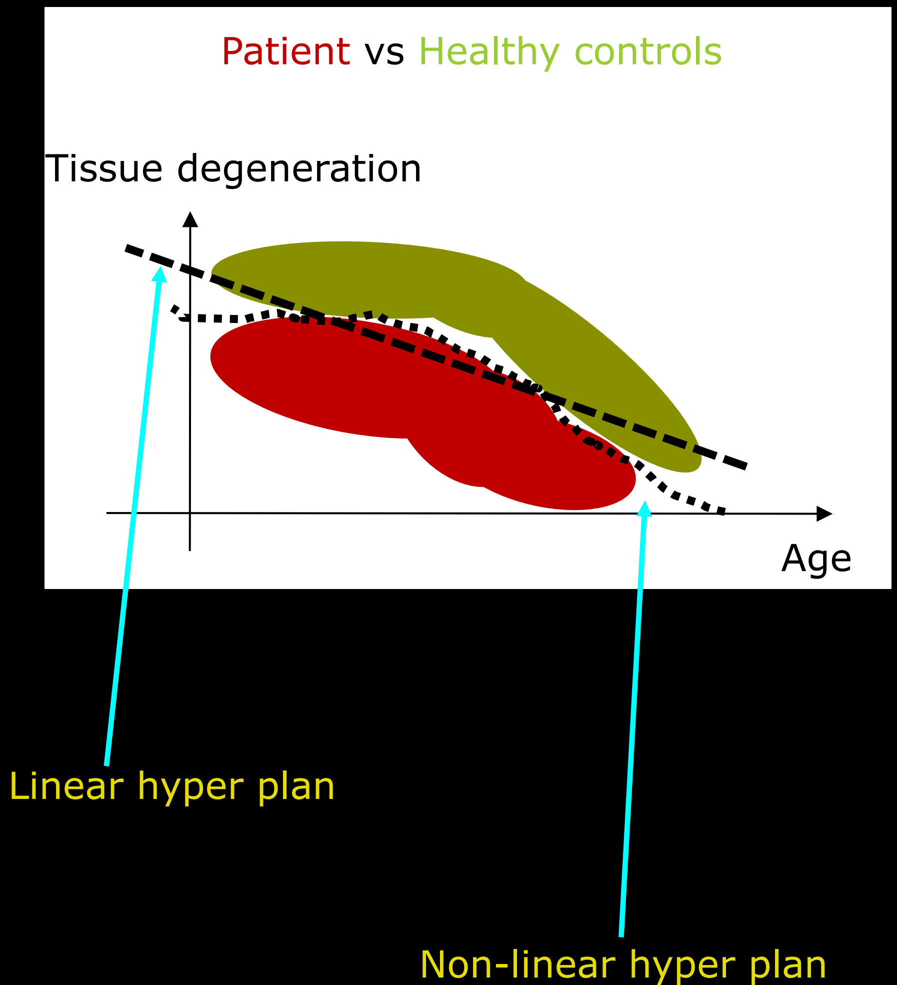
Decision boundary?



Segmentation result: Fisher's LDA

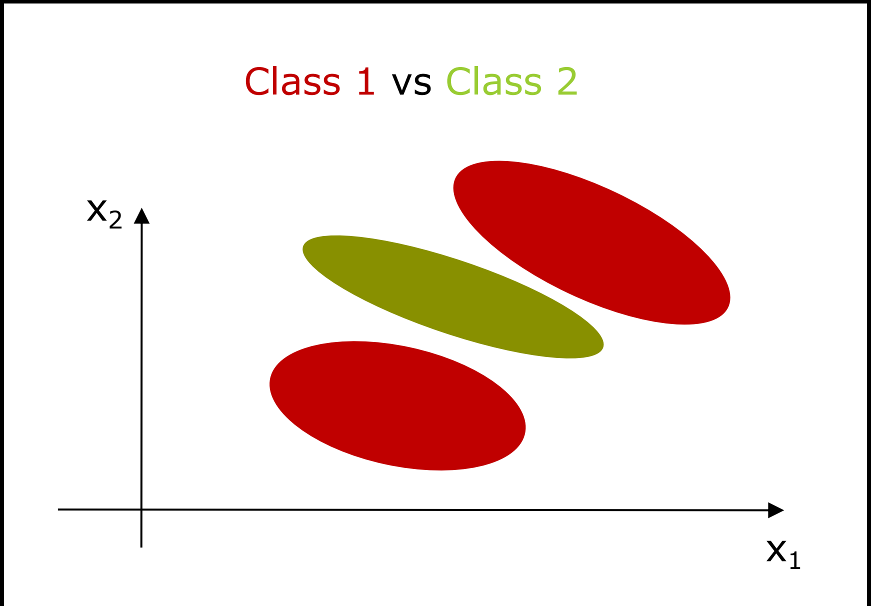


Limitations of LDA



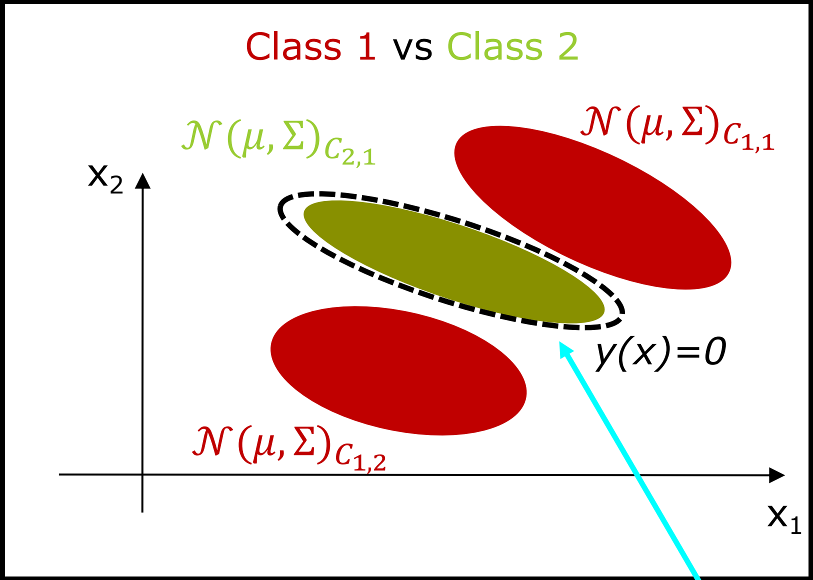
- Linear discriminant analysis (LDA)
  - Only linear hyper plans
- Non-linear hyper plans?
- Example:
  - I wish to make a classifier
  - Features (2D):
  - Age vs. Tissue degeneration
- Classes
  - Healthy controls vs Patient

Limitations of LDA



- One class can be separated
  - A non-linear problem

Non-linear Hyper plans



- Class 1: $\mathcal{N}(\mu,\Sigma)_{C_{1,1}} + \mathcal{N}(\mu,\Sigma)_{C_{1,2}}$
- Class 2: $\mathcal{N}(\mu,\Sigma)_{C_{2,1}}$

Non-linear hyper plan

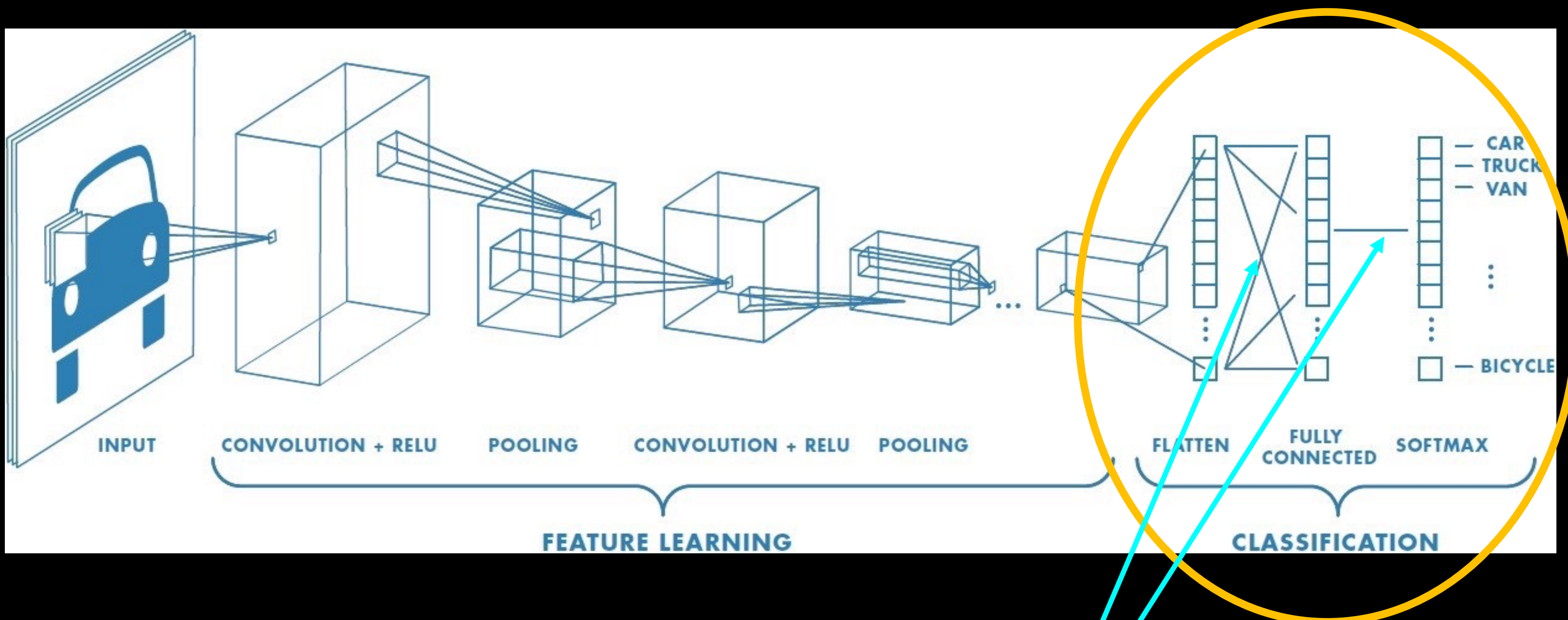
Non-linear classifier (Machine learning):

Example:

- Gaussian Mixture Model
  - Each class is modelled using a number of Gauss distributions e.g. class 1
- Again use Bayes theorem also for Gaussian Mixture Model
- Optimisation:
  - We derive $\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}=0$ for a Gaussian mixture model
  - Iterative optimisation algorithm is used to find ***w***

Segmentation - Non-linear Hyper plans

- Convolutional neural network and classification

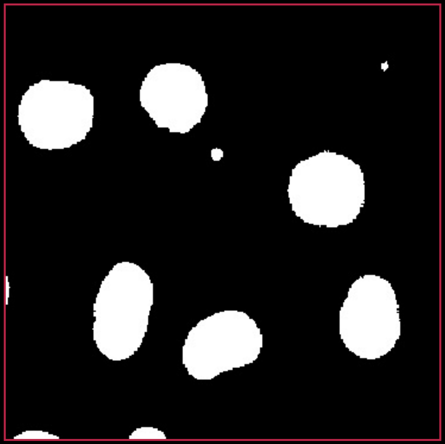


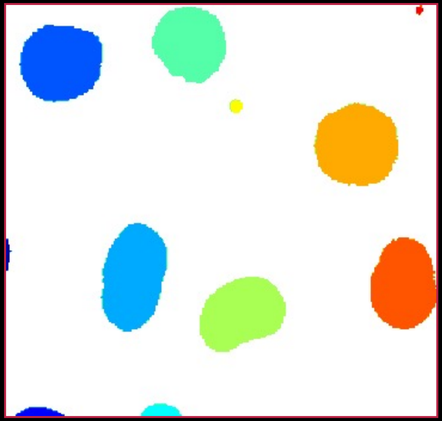
Weights are non-linear sigmoid functions: $y_k = \phi(x, w, w0)$

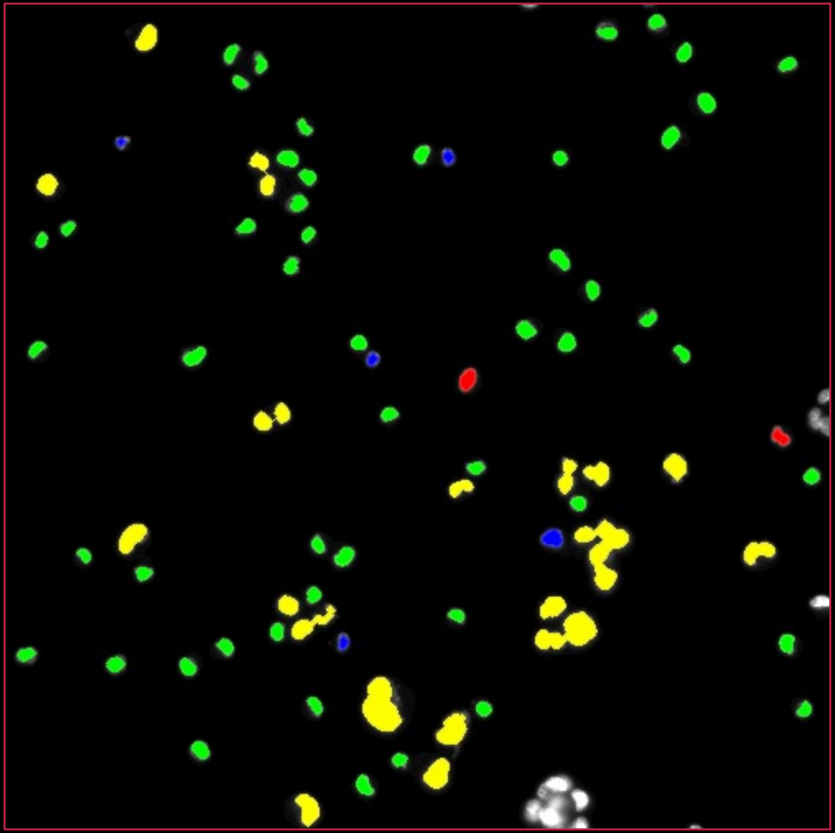
What did you learn today?

- Describe the concept of pixel classification
- Compute the pixel value ranges in a minimum distance classifier
- Implement and use a minimum distance classifier
- Approximate a pixel value histogram using a Gaussian distribution
- Implement and use a parametric classifier
- Decide if a minimum distance or a parametric classifier is appropriate based on the training data
- Explain the concept of Bayesian classification
- Understand the use of 1D vs 2D feature space
- Implement and use the linear discriminant analysis (LDA) classifier
- Understand the use of linear vs non-line hyper-plans for segmentation

Lecture 6 – BLOB analysis and feature based classification







Teaching – the speed of the lecture

A) Come oooooon! I am so bored

B) I can easily follow and knit my sweater

C) The speed is fine

D) I need to concentrate a lot to follow

E) Hey! Wait! You are too fast

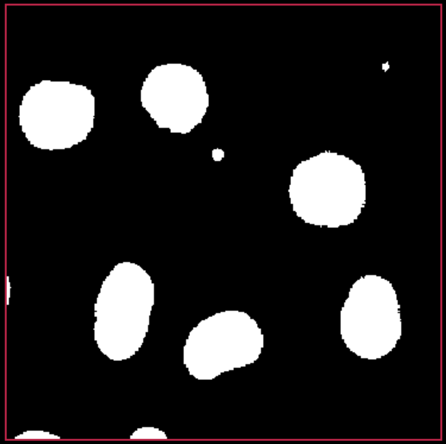
Image Analysis

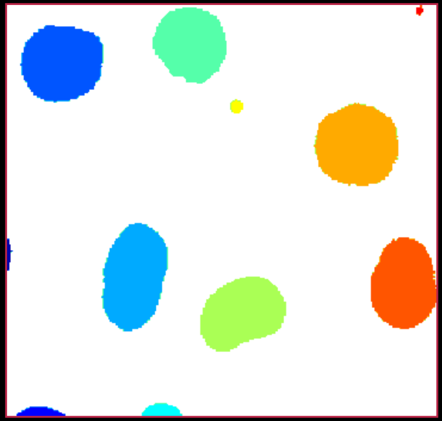
Rasmus R. Paulsen
Tim B. Dyrby
DTU Compute

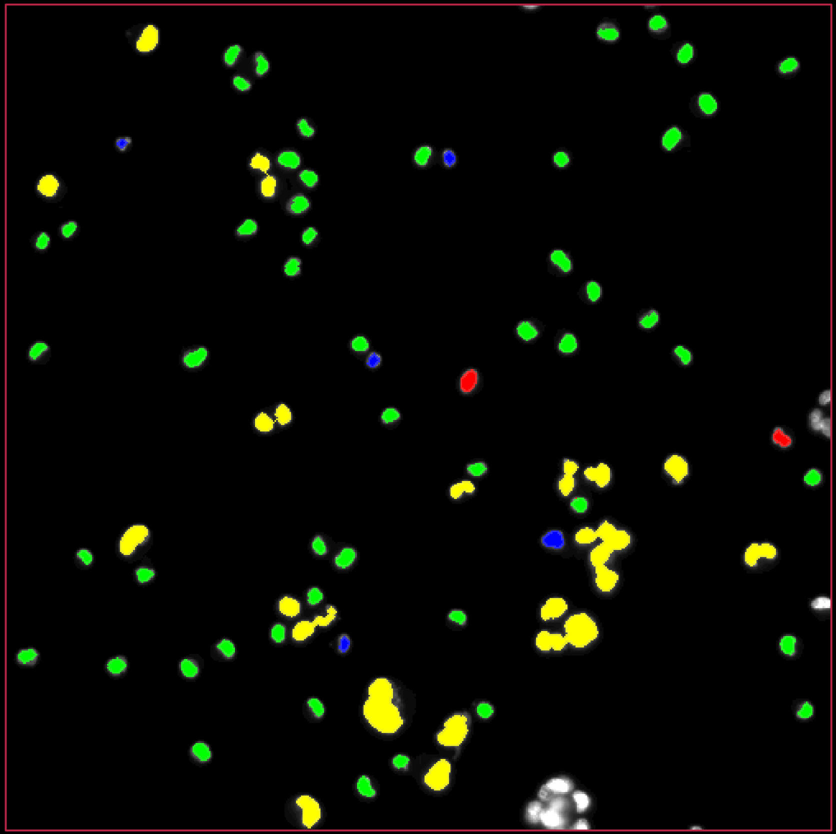
rapa@dtu.dk

http://www.compute.dtu.dk/courses/02502

Lecture 6 – BLOB analysis and feature based classification





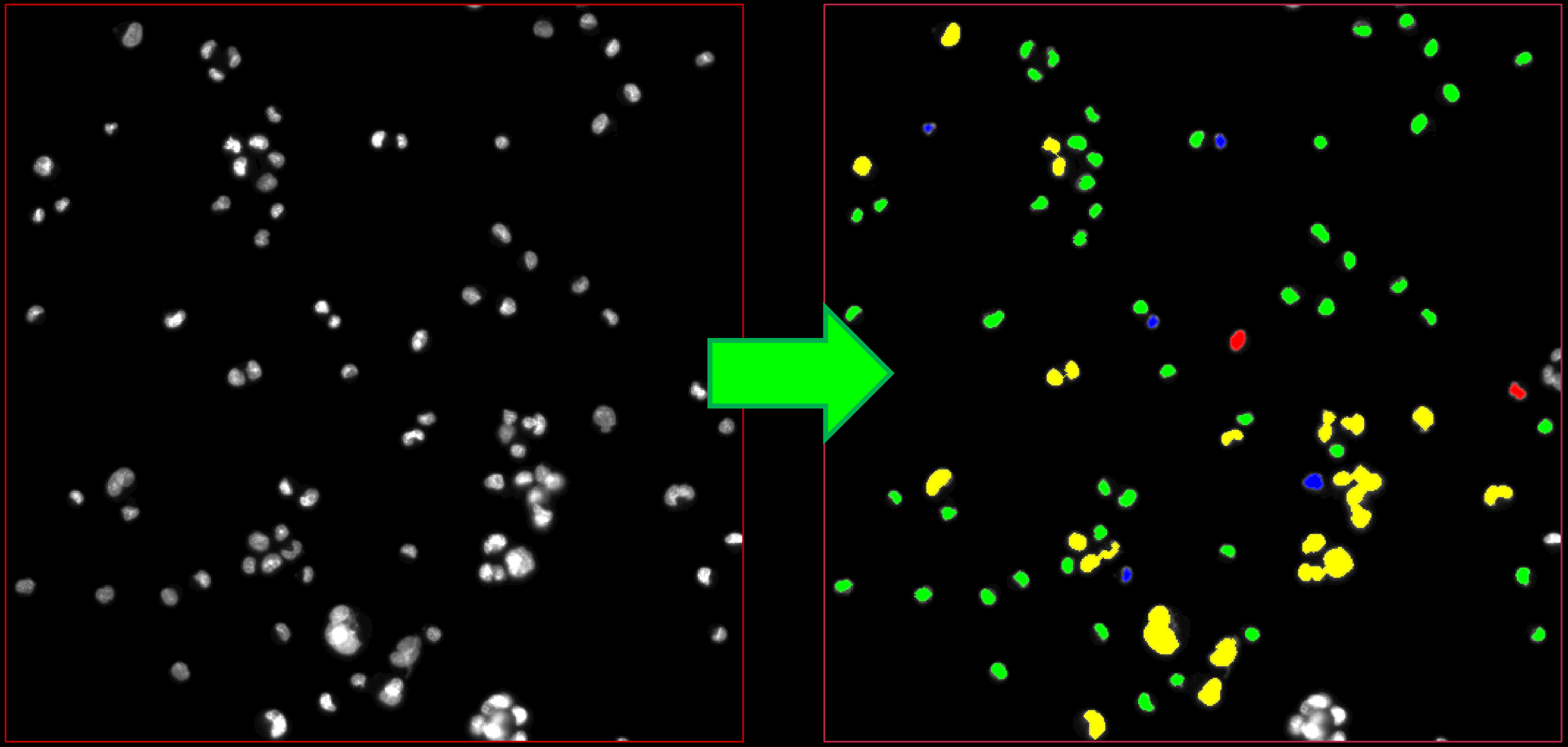


What can you do after today?

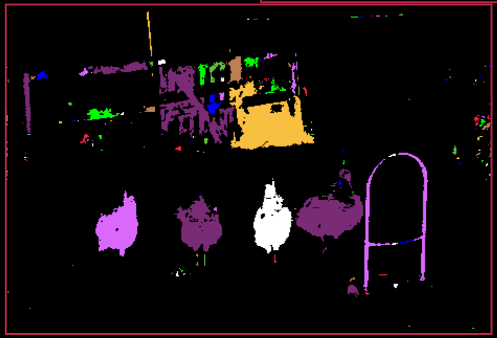
- Calculate the connected components of a binary image. Both using 4-connected and 8-connected neighbours
- Compute BLOB features including area, bounding box ratio, perimeter, center of mass, circularity, and compactness
- Describe a feature space
- Compute blob feature distances in feature space
- Classify binary objects based on their blob features
- Estimate feature value ranges using annotated training data
- Compute a confusion matrix
- Compute rates from a confusion matrix including sensitivity, specificity and accuracy
- Determine and discuss what is the importance of sensitivity and specificity given an image analysis problem

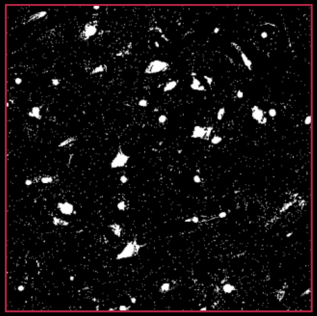
Object recognition

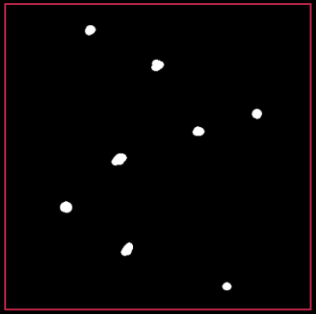
- Recognise objects in images
- Put them into different classes

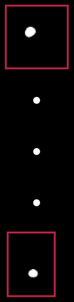


BLOB – what is it?



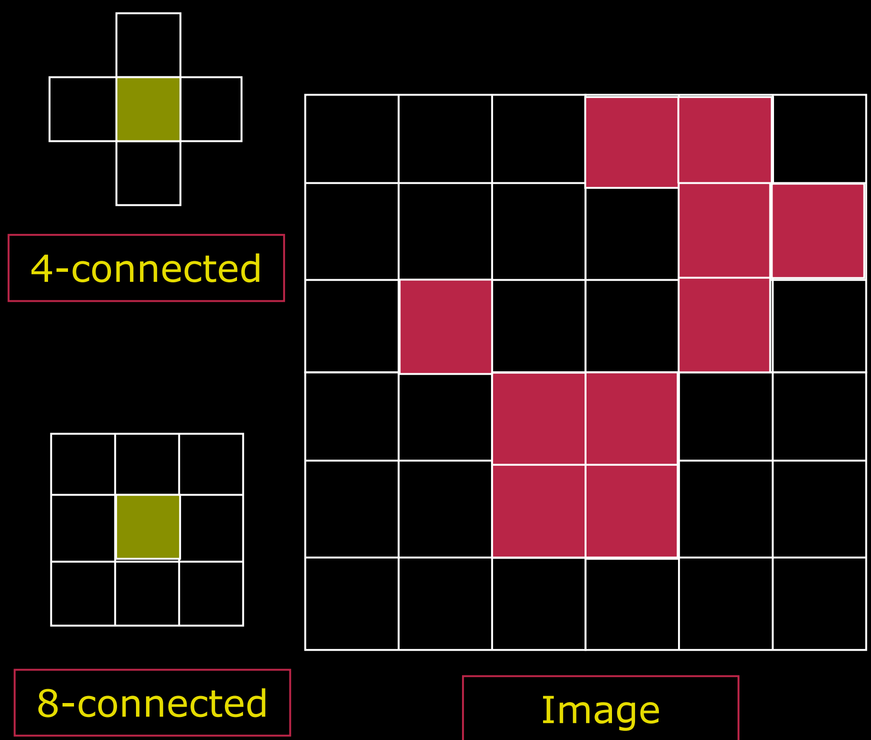






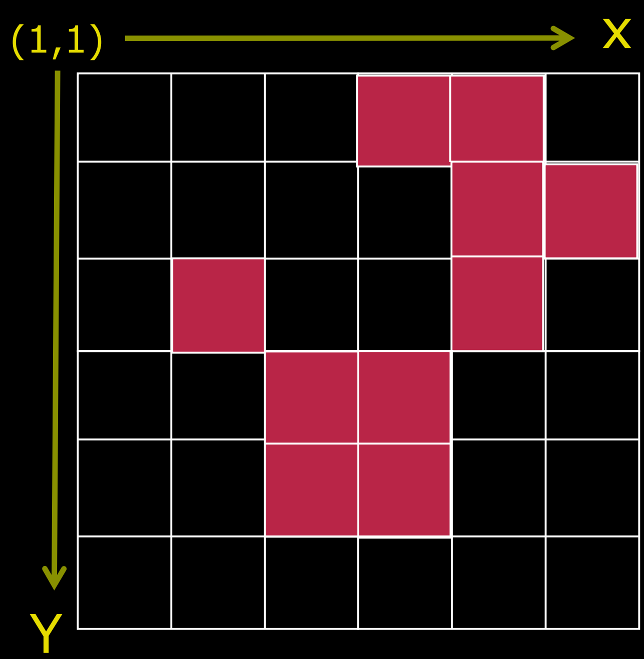
- BLOB = Binary Large Object
  - Group of connected pixels
- BLOB Analysis
  - *Connected component analysis*
  - *Object labelling*

Isolating a BLOB



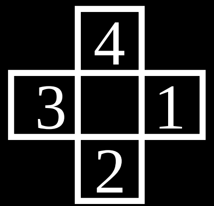
- What we want:
  - For each object in the image, a list with its pixels
- How do we get that?
  - Connected component analysis
- Connectivity
  - Who are my neighbors?
  - 4-connected
  - 8-connected

Connected component analysis



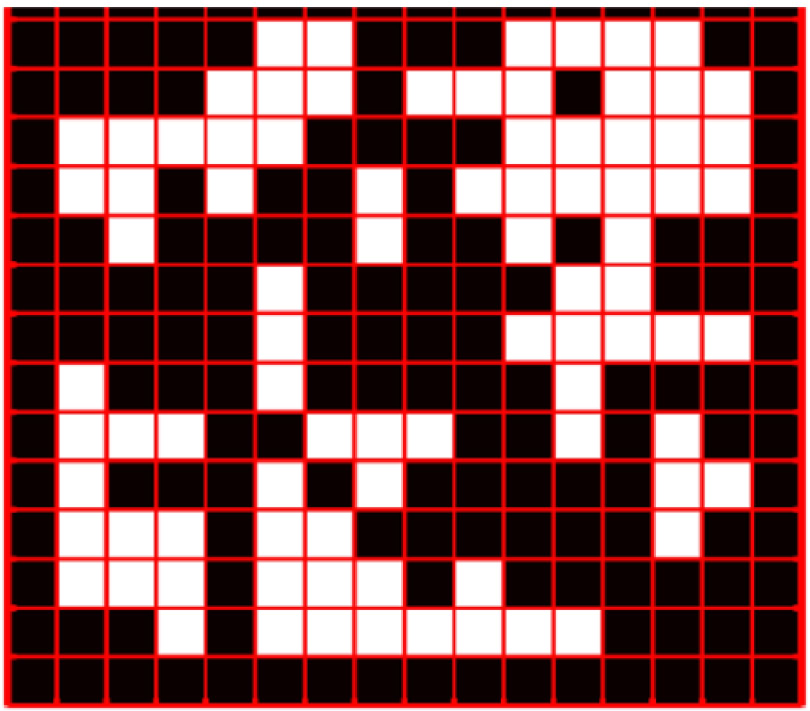
- Binary image
- Seed point: where do we start?
- *Grassfire* concept
  - Delete (burn) the pixels we visit
  - Visit all *connected* (4 or 8) neighbors

4-connected



BLOBs with 4- and 8- connectivity

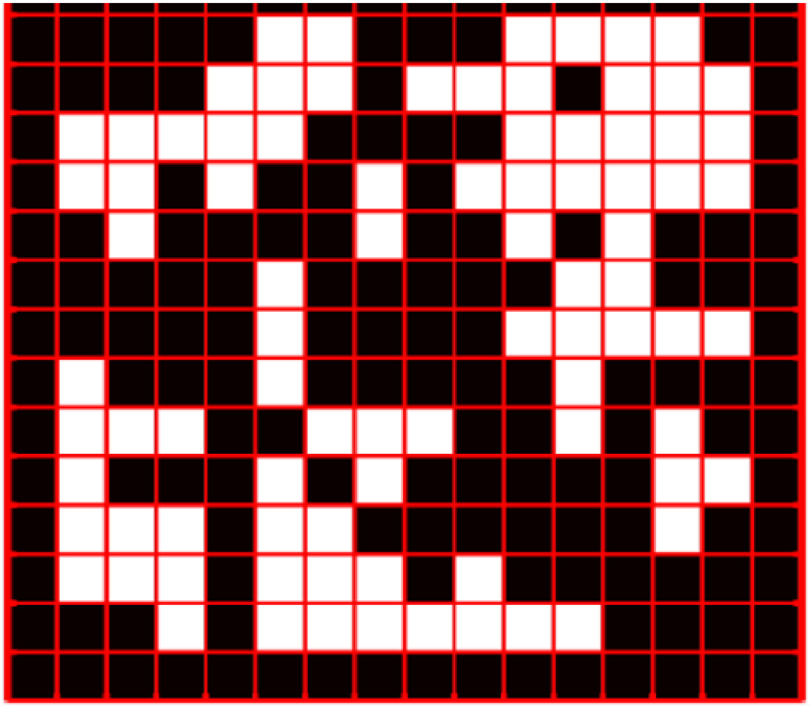
A BLOB analysis is performed using both 4- and 8- connectivity. How many BLOBS are found using the two different connectivities?



- 3 and 7
- 9 and 5
- 8 and 6
- 7 and 5
- 4 and 5

BLOBs with 4- and 8- connectivity

A BLOB analysis is performed using both 4- and 8- connectivity. How many BLOBS are found using the two different connectivities?



3 and 7

9 and 5

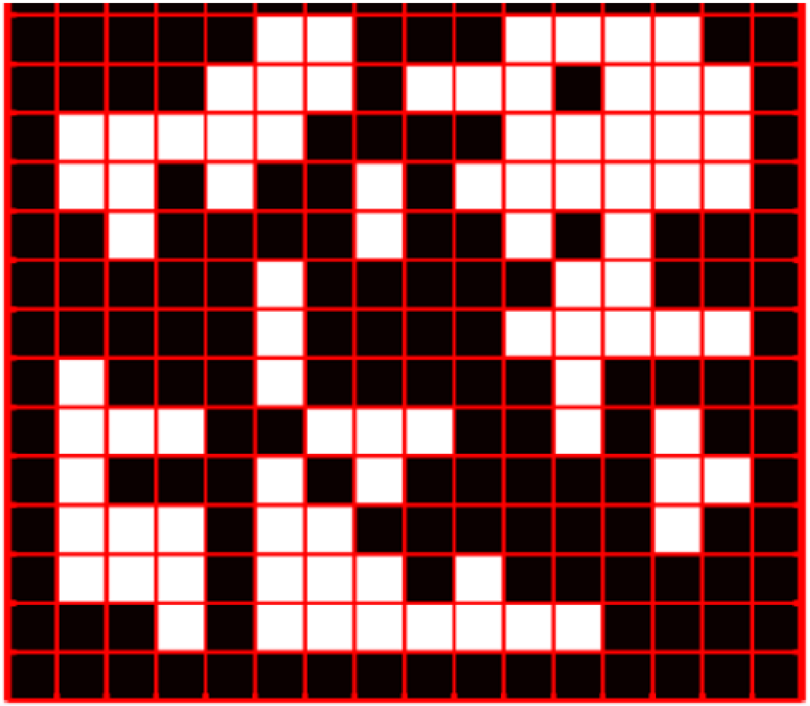
8 and 6

7 and 5

4 and 5

BLOBs with 4- and 8- connectivity

A BLOB analysis is performed using both 4- and 8- connectivity. How many BLOBS are found using the two different connectivities?



3 and 7

9 and 5

8 and 6 ✓ 0%

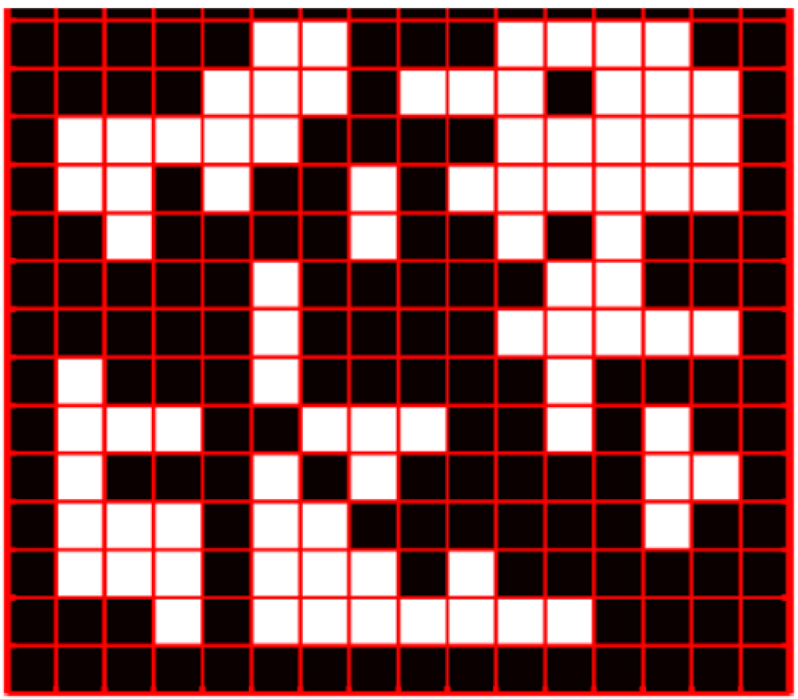
7 and 5

4 and 5

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

BLOBs with 4- and 8- connectivity

A BLOB analysis is performed using both 4- and 8- connectivity. How many BLOBS are found using the two different connectivities?



- 3 and 7
- 9 and 5
- 8 and 6
- 7 and 5
- 4 and 5

BLOB extraction and binary run length coding

A BLOB analysis using 4-connectivity is performed on the image seen in Figure 4. It is a 0-based (x,y) coordinate system with origin in the upper left corner. The largest **BLOB** is kept and the resulting image is coded using binary run-length coding. The code is:

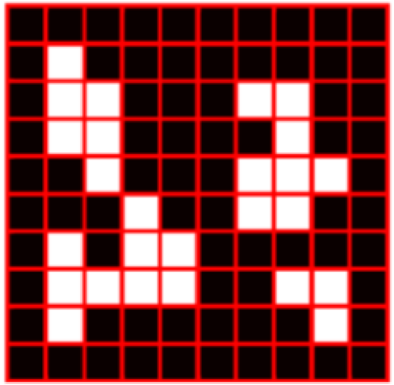


Figure 4: Binary image I. White pixels are foreground (1) and black pixels are background (0).

[5; (3; 3)]; [6; (1; 1)]; [6; (3; 4)]; [7; (1; 4)]; [8; (1; 1)]

[4; (3; 3)]; [6; (1; 2)]; [7; (3; 4)]; [7; (1; 4)]; [8; (1; 1)]

[5; (2; 3)]; [6; (1; 1)]; [6; (3; 4)]; [8; (1; 5)]; [8; (1; 1)]

[4; (3; 3)]; [6; (1; 2)]; [6; (3; 5)]; [7; (1; 4)]; [8; (1; 3)]

[5; (2; 3)]; [6; (1; 1)]; [6; (3; 4)]; [7; (2; 4)]; [8; (2; 3)]

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

BLOB extraction and binary run length coding

A BLOB analysis using 4-connectivity is performed on the image seen in Figure 4. It is a 0-based (x,y) coordinate system with origin in the upper left corner. The largest BLOB is kept and the resulting image is coded using binary run-length coding. The code is:

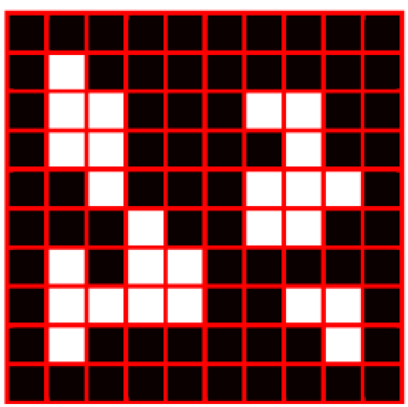


Figure 4: Binary image I. White pixels are foreground (1) and black pixels are background (0).

[5; (3; 3)]; [6; (1; 1)]; [6; (3; 4)]; [7; (1; 4)]; [8; (1; 1)]

[4; (3; 3)]; [6; (1; 2)]; [7; (3; 4)]; [7; (1; 4)]; [8; (1; 1)]

[5; (2; 3)]; [6; (1; 1)]; [6; (3; 4)]; [8; (1; 5)]; [8; (1; 1)]

[4; (3; 3)]; [6; (1; 2)]; [6; (3; 5)]; [7; (1; 4)]; [8; (1; 3)]

[5; (2; 3)]; [6; (1; 1)]; [6; (3; 4)]; [7; (2; 4)]; [8; (2; 3)]

BLOB extraction and binary run length coding

A BLOB analysis using 4-connectivity is performed on the image seen in Figure 4. It is a 0-based (x,y) coordinate system with origin in the upper left corner. The largest BLOB is kept and the resulting image is coded using binary run-length coding. The code is:

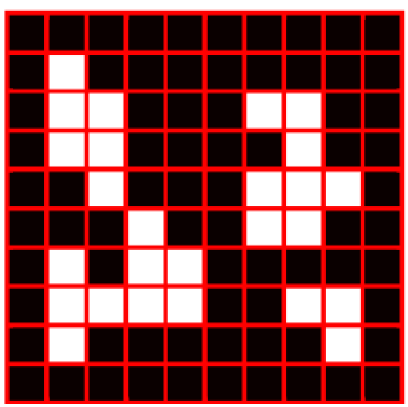


Figure 4: Binary image I. White pixels are foreground (1) and black pixels are background (0).

[5; (3; 3)]; [6; (1; 1)]; [6; (3; 4)]; [7; (1; 4)]; [8; (1; 1)]

✓ 0%

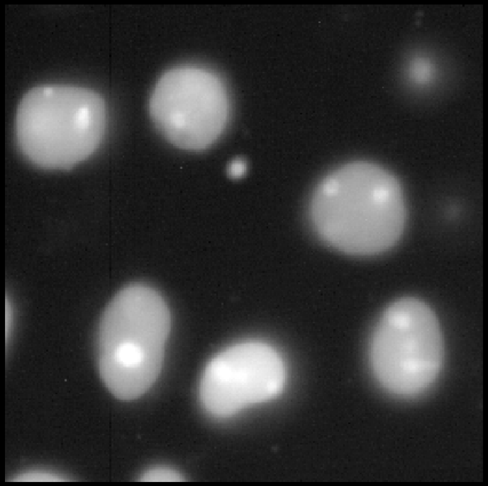
[4; (3; 3)]; [6; (1; 2)]; [7; (3; 4)]; [7; (1; 4)]; [8; (1; 1)]

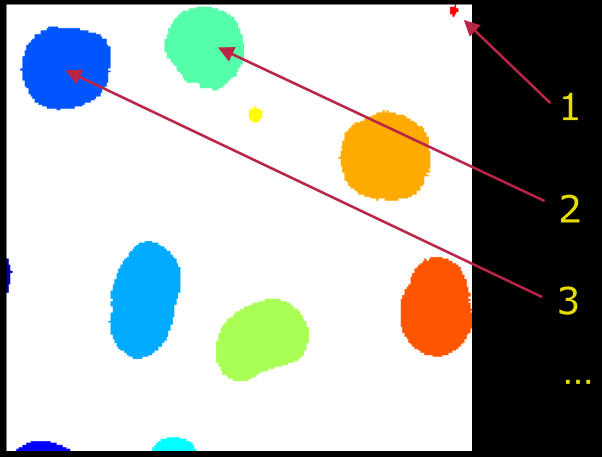
[5; (2; 3)]; [6; (1; 1)]; [6; (3; 4)]; [8; (1; 5)]; [8; (1; 1)]

[4; (3; 3)]; [6; (1; 2)]; [6; (3; 5)]; [7; (1; 4)]; [8; (1; 3)]

[5; (2; 3)]; [6; (1; 1)]; [6; (3; 4)]; [7; (2; 4)]; [8; (2; 3)]

The result of connected component analysis





- An image where each BLOB (component) is labelled
- Each blob now has a unique ID number
- What do we do with these blobs?

Features

- Feature
  - A prominent or distinctive aspect, quality, or characteristic
  - *This radio has many good features*
- Car (Ford-T) features
  - 4 wheels
  - 2 doors
  - 540 kg
  - 20 hp

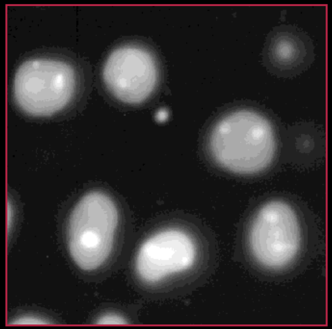
Feature vector

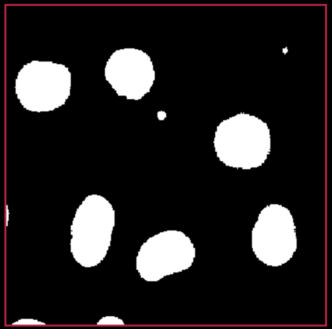
f=[4, 2, 540, 20]

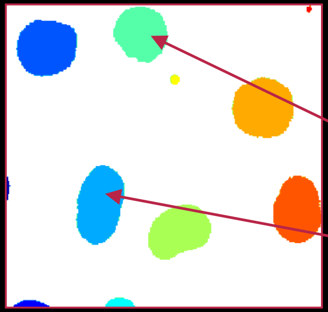
f=[4, 3, 1100, 90]

- Feature vector
  - Vector with all the features for one object
- Ford-T features
  - 4 wheels
  - 2 doors
  - 540 kg
  - 20 hp
- Ford Fiesta features
  - 4 wheels
  - 3 doors
  - 1100 kg
  - 90 hp

Feature extractions





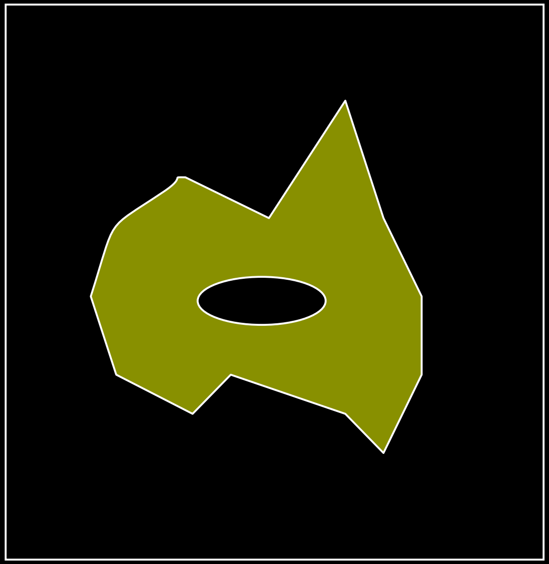


- Compute features for each BLOB that can be used to identify it
  - Size
  - Shape
  - Position
- From image operations to mathematical operations
  - Input: a list of pixel positions
  - Output: Feature vector
- First step: remove invalid BLOBS
  - too small or big- using morphological operations for example
  - border BLOBs

Feature vector = [2,1,…,3]

Feature vector = [4,7,…,0]

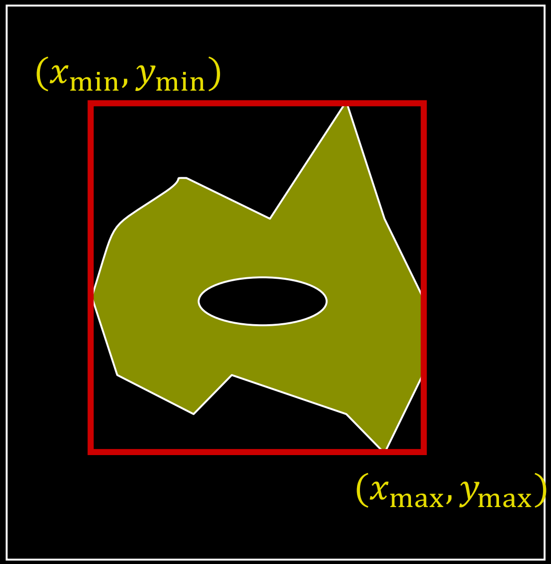
BLOB Features



One BLOB

- Area
  - number of pixels in the BLOB
  - Can be used to remove noise (small BLOBS)

BLOB Features



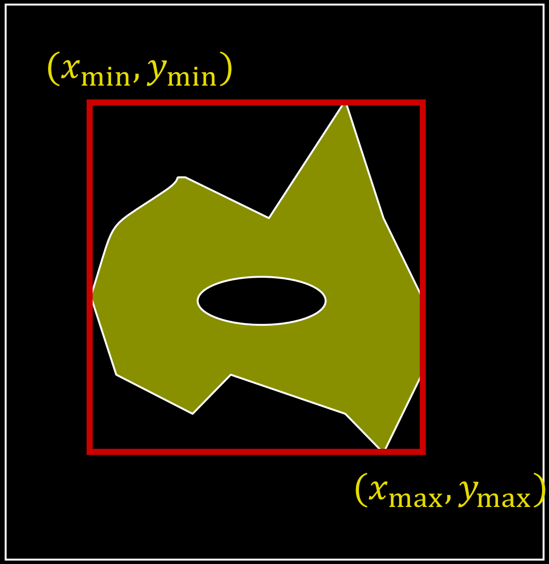
One BLOB

- Bounding box
  - Minimum rectangle that contains the BLOB
  - Height: $y_{\max} - y_{\min}$
  - Width: $x_{\max} - x_{\min}$
  - Bounding box ratio:

    $$\frac{y_{\max} - y_{\min}}{x_{\max} - x_{\min}}$$

  - tells if the BLOB is elongated

BLOB Features



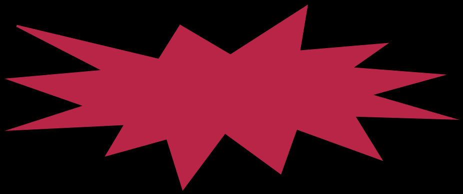
One BLOB

- Bounding box
  - Bounding box area:

$$(y_{\max} - y_{\min}) \cdot (x_{\max} - x_{\min})$$

  - Compactness of BLOB

$$\text{Compactness} = \frac{\text{BLOB Area}}{(y_{\max} - y_{\min}) \cdot (x_{\max} - x_{\min})}$$

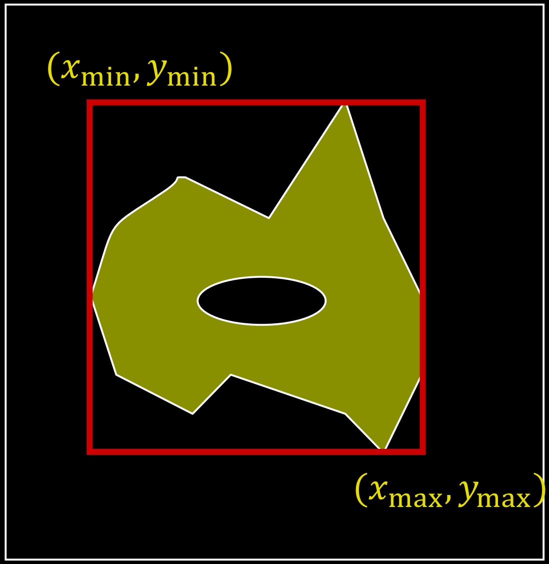


Not compact

Compact

BLOB Features

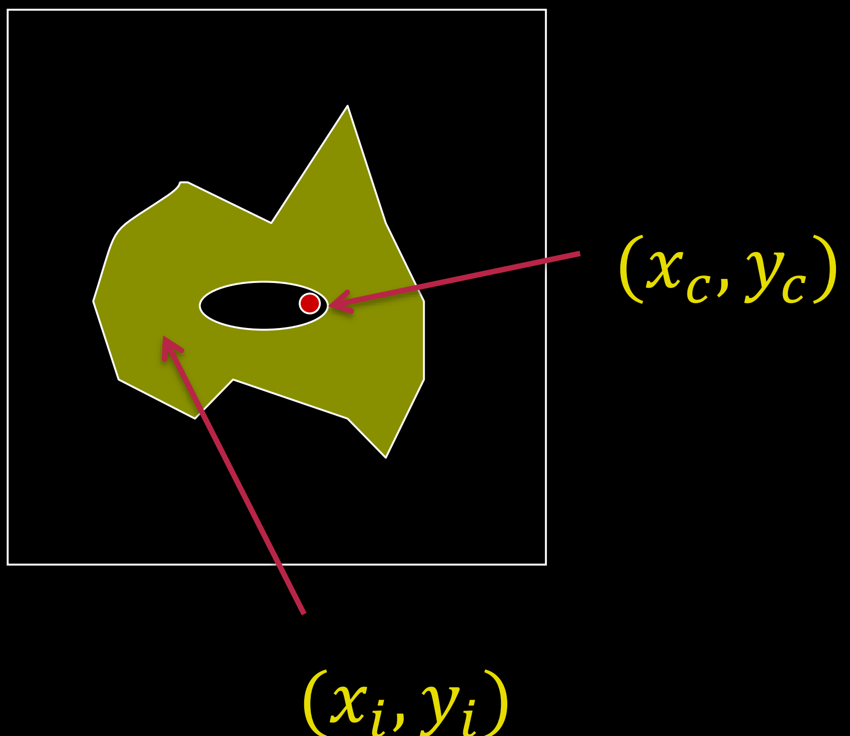
- Bounding box ratio
  - Bounding box height divided by the width



One BLOB

BLOB Features

- Center of mass (x_c, y_c)

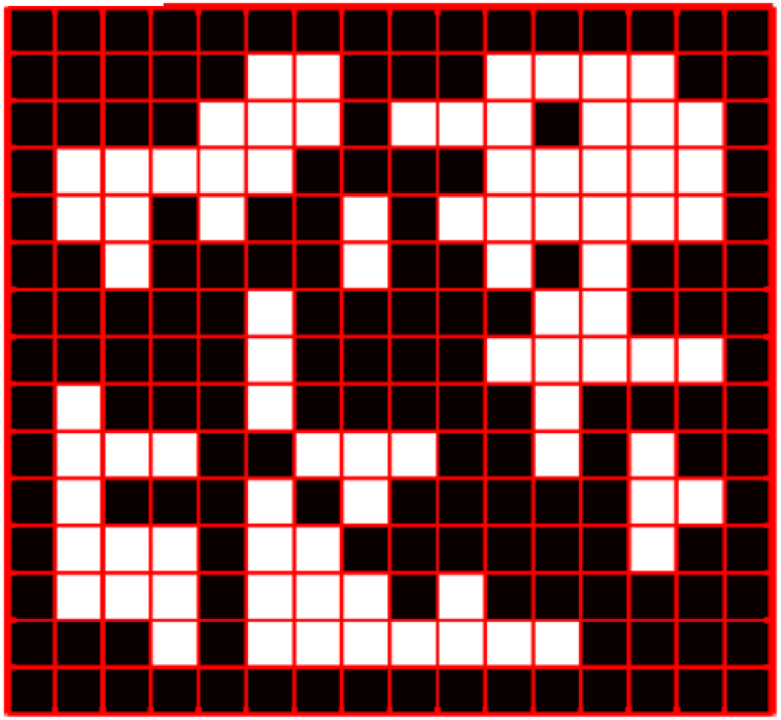


$$x_c = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$y_c = \frac{1}{N}\sum_{i=1}^{N} y_i$$

BLOB Center of Mass

The smallest BLOB is found using 4-connectivity. What is the center of mass of this BLOB. The image has origin (0,0) and uses a (x,y) coordinate system.



(12, 1.5)

(5, 8.5)

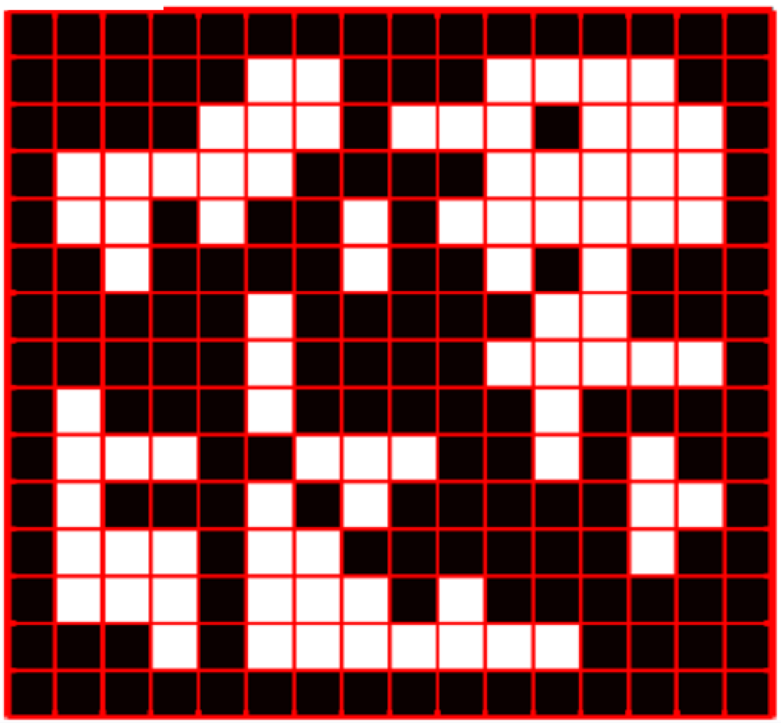
(6.5, 3.5)

(4.5, 0.5)

(7, 4.5)

BLOB Center of Mass

The smallest BLOB is found using 4-connectivity. What is the center of mass of this BLOB. The image has origin (0,0) and uses a (x,y) coordinate system.



(12, 1.5)

(5, 8.5)

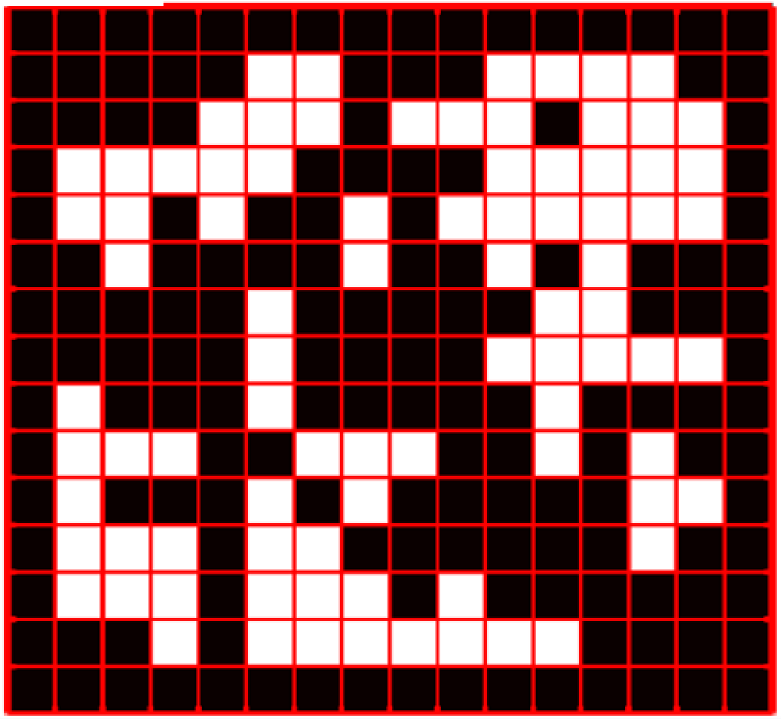
(6.5, 3.5)

(4.5, 0.5)

(7, 4.5)

BLOB Center of Mass

The smallest BLOB is found using 4-connectivity. What is the center of mass of this BLOB. The image has origin (0,0) and uses a (x,y) coordinate system.



(12, 1.5)

(5, 8.5)

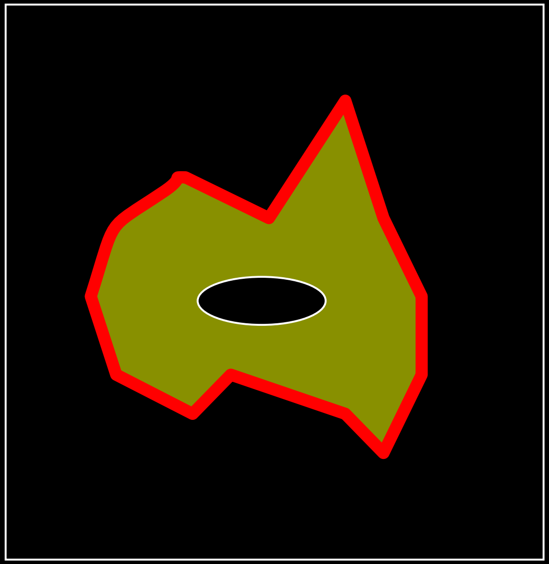
(6.5, 3.5)

(4.5, 0.5)

(7, 4.5)

✓0%

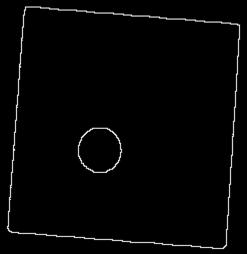
BLOB Features



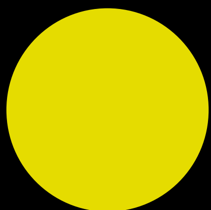
One BLOB

- Perimeter
  - Length of perimeter
  - How can we compute that?
- In practice (in Matlab) it is computed differently and more accurately

$$\sum((f(x,y) \oplus \mathrm{SE}) - f(x,y))$$



BLOB Features - circularity

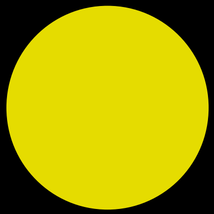


Circle like

Not circle like

- How much does it look like a circle?
- Circle
  - Area $A = \pi r^2$
  - Perimeter $P = 2\pi r$
- New object assumed to be a circle
  - Measured perimeter P_m
  - Measured area A_m
- Estimate perimeter from (measured) area
  - Estimated perimeter $P_e = 2\sqrt{\pi A_m}$

BLOB Features - circularity



Circle like

Not circle like

- Compare the perimeters
  - Measured perimeter P_m
  - Estimated perimeter $P_e = 2\sqrt{\pi A_m}$
- Circularity 1:

$$\text{Circularity} = \frac{P_m}{P_e} = \frac{P_m}{2\sqrt{\pi A_m}}$$

Circularity math

$$\text{Circularity} = \frac{P_m}{P_e} = \frac{P_m}{2\sqrt{\pi A_m}}$$

$P_m < P_e$

$P_m = P_e$

$P_m > P_e$

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Circularity math

$$\text{Circularity} = \frac{P_m}{P_e} = \frac{P_m}{2\sqrt{\pi A_m}}$$

$$P_m < P_e$$

$$P_m = P_e$$

$$P_m > P_e$$

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Circularity math

$$\text{Circularity} = \frac{P_m}{P_e} = \frac{P_m}{2\sqrt{\pi A_m}}$$

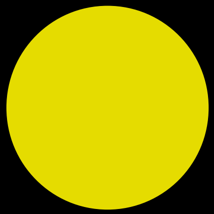
$P_m = P_e$

$P_m > P_e$

✓0%

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

BLOB Features - circularity



Circle like

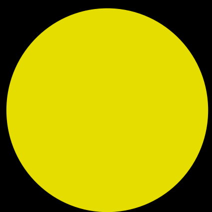
Not circle like

- Compare the perimeters
  - Measured perimeter P_m
  - Estimated perimeter $P_e = 2\sqrt{\pi A_m}$
- Circularity:

$$\text{Circularity} = \frac{P_m}{P_e} = \frac{P_m}{2\sqrt{\pi A_m}}$$

- This measure will normally be ≥1

BLOB Features – circularity inverse



Circle like

Not circle like

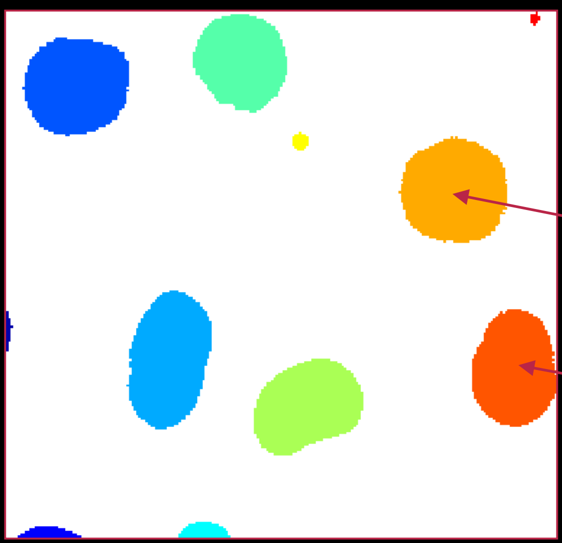
- Compare the perimeters
  - Measured perimeter P_m
  - Estimated perimeter $P_e = 2\sqrt{\pi A_m}$
- Circularity (inverse):

$$\text{Circularity inverse} = \frac{P_e}{P_m} = \frac{2\sqrt{\pi \mathrm{A}_m}}{P_m}$$

- This measure will normally be ≤1

After feature extraction

Area, compactness, circularity etc calculated for all BLOB



Feature vector = [2,1,…,3]

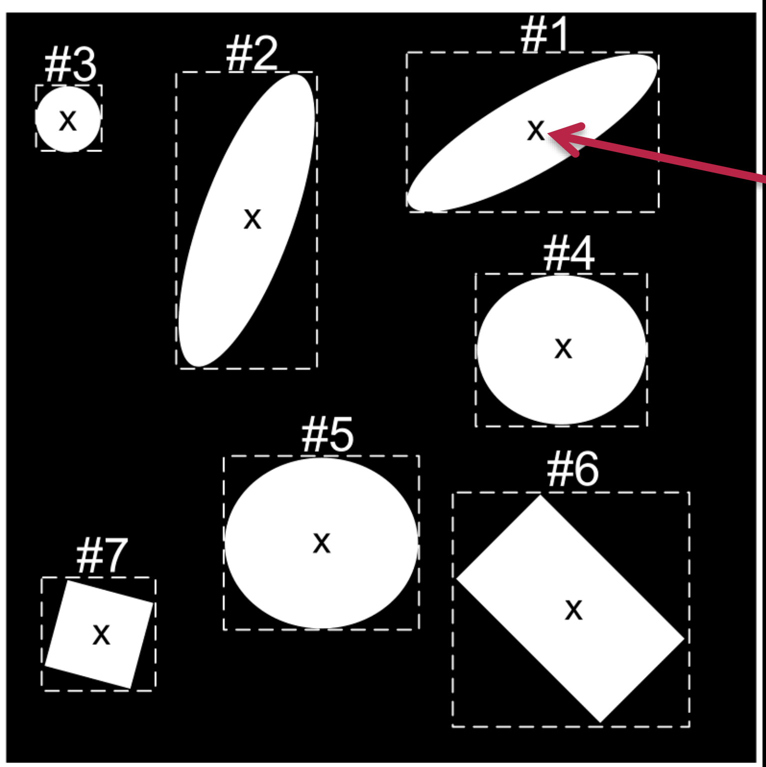
Feature vector = [4,7,…,0]

One feature vector per blob

BLOB Classification

- Classification
  - Put a BLOB into a *class*
- *Classes* are normally pre-defined
  - *Car*
  - *Bus*
  - *Motorcycle*
  - *Scooter*
- Object recognition

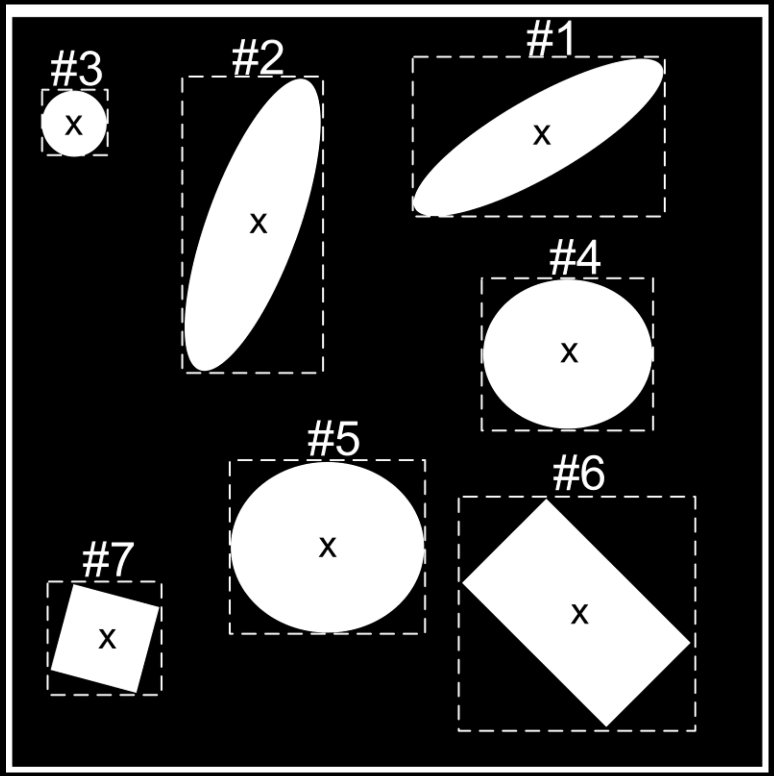
Object recognition: Circle example



BLOB number	Circu-larity	Area (pixels)
1	0.31	6561
2	0.40	6544
3	0.98	890
4	0.97	6607
5	0.99	6730
6	0.52	6611
7	0.75	2073

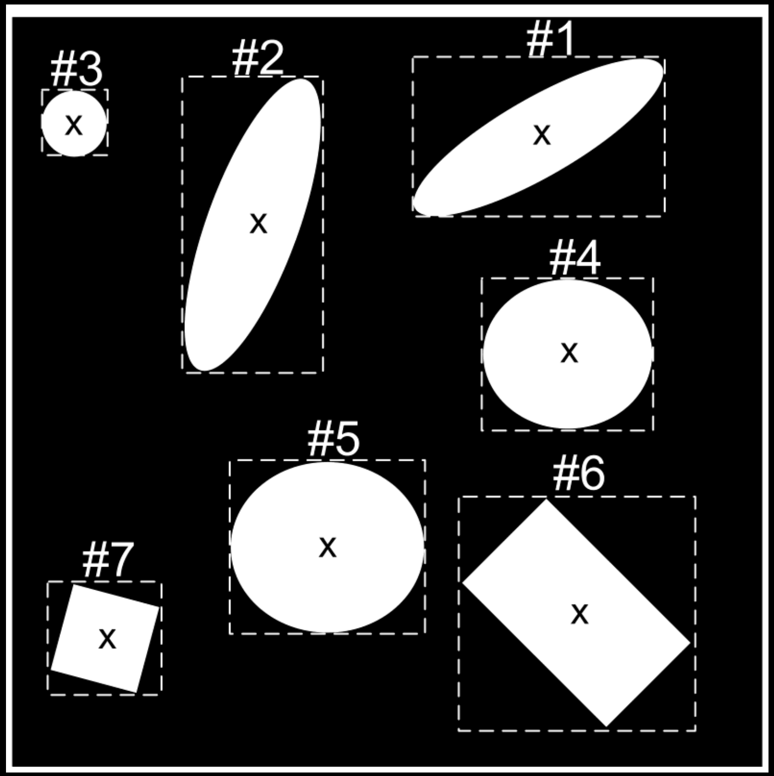
Which objects are circles?

Circle classification



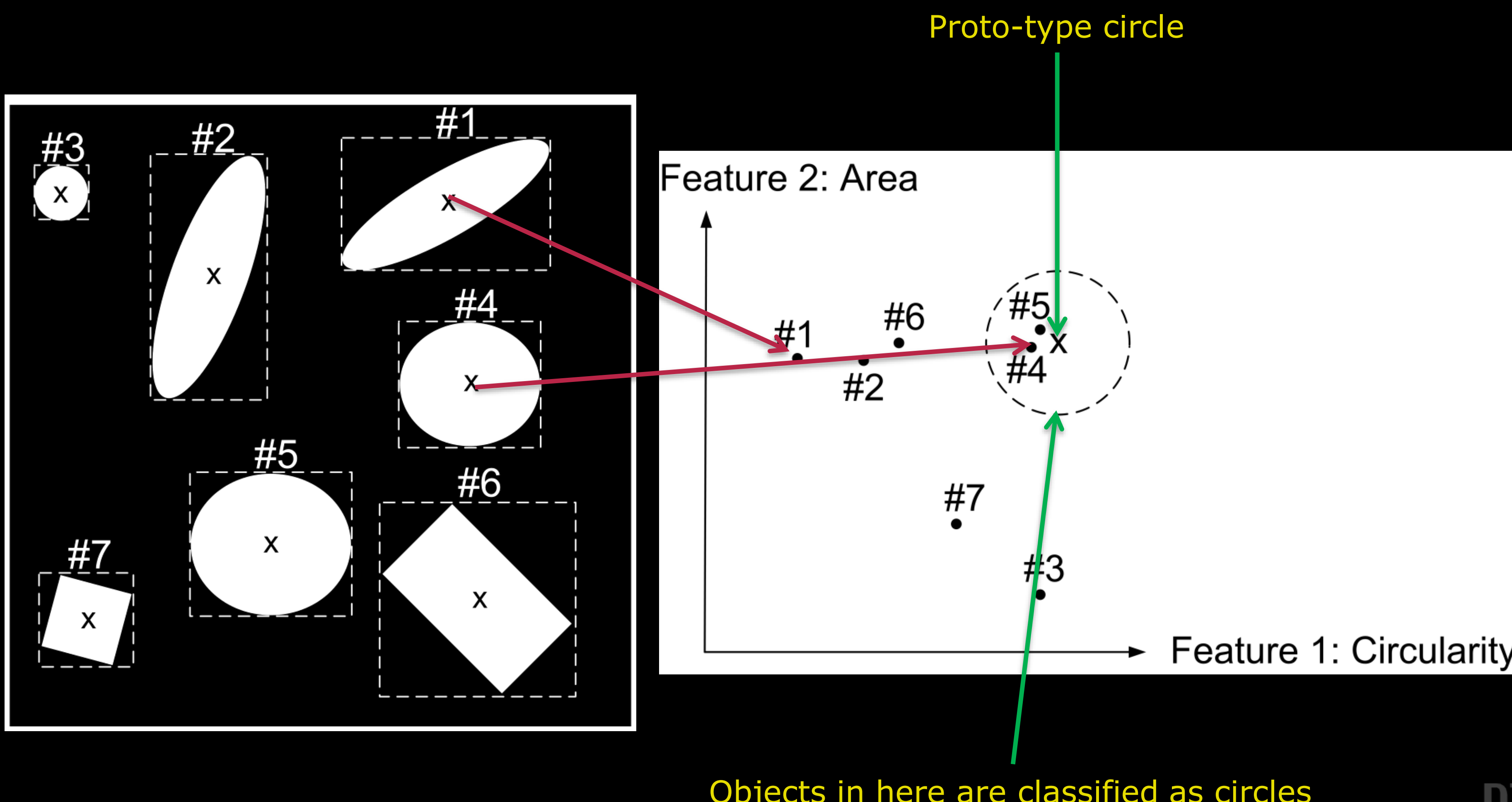
- Two classes:
  - Circle
  - Not-circle
- Lets make a model of a *proto-type* circle

Circle classification

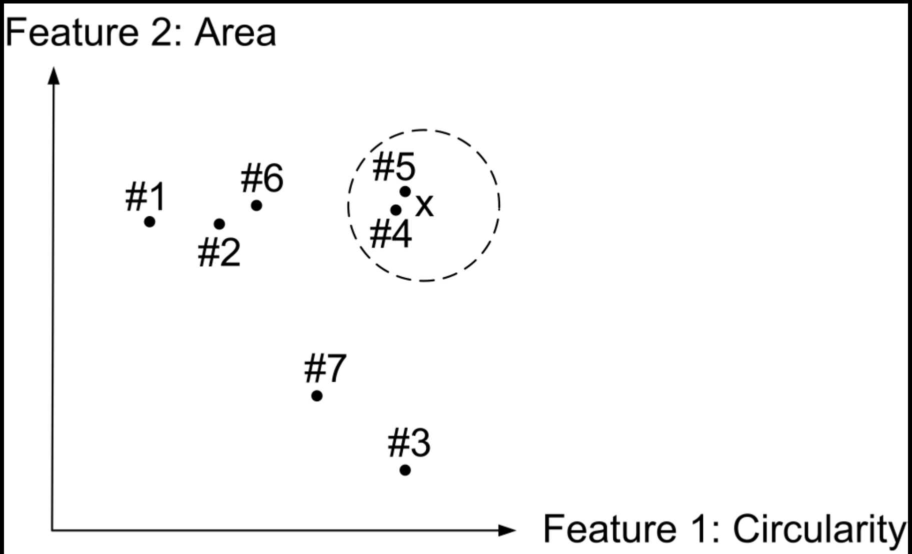


- Proto-type circle
  - Circularity : 1
  - Area: 6700

Feature Space

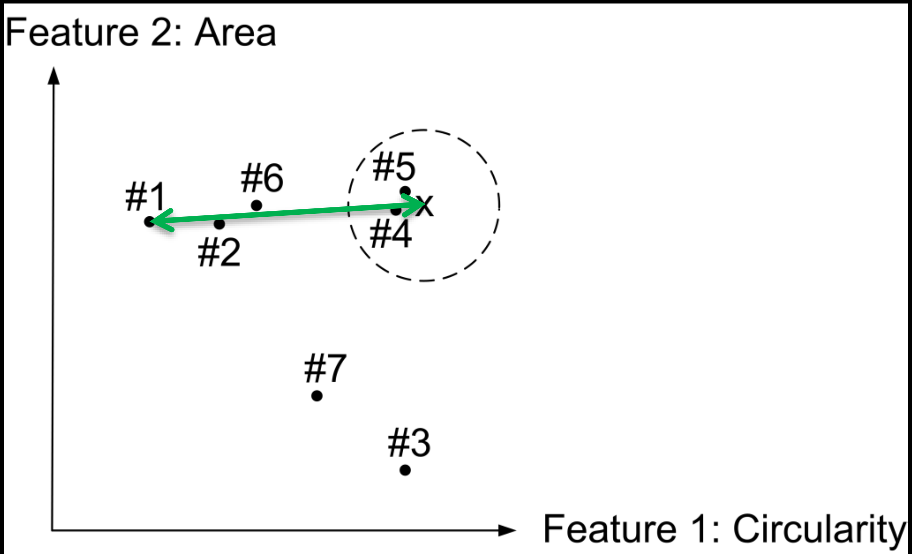


Feature space



- Proto-type circle
  - Circularity : 1
  - Area: 6700
- Some slack is added to allow non-perfect circles
  - Circularity: 1 +/- 0.15

Feature space - distances



- How do we decide if an object is inside the circle?
- Feature space distance
- Euclidean distance in features space

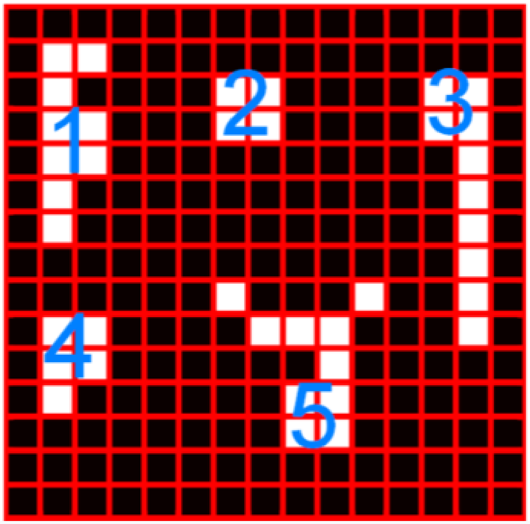
Blob 1: circularity: 0.31, Area : 6561

$$D = \sqrt{(0.31 - 1)^2 + (6561 - 6700)^2}$$

Dominates all! – normalisation needed

BLOB Classification

A BLOB analysis using 8-connectivity has been performed on the image seen in Figure 12 and the five found BLOBs have been marked with numbers. The BLOB features *area* and *compactness* have been computed for the five BLOBs. A reference BLOB has an area of 10 pixels and a compactness of 0.5. The Euclidean distance in feature space has been computed between the five BLOBs and the reference BLOB. Which of the five BLOBs has the minimum distance?



1

2

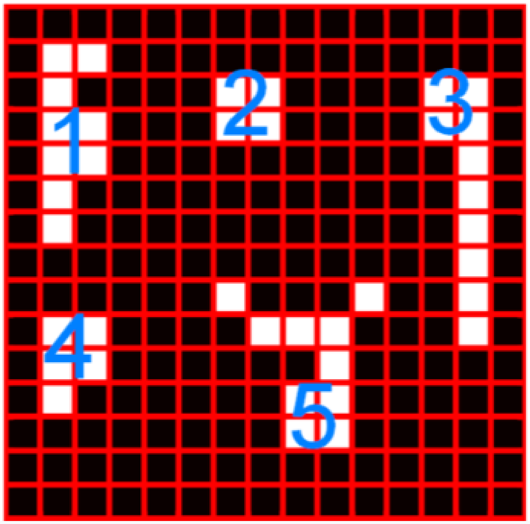
3

4

5

BLOB Classification

A BLOB analysis using 8-connectivity has been performed on the image seen in Figure 12 and the five found BLOBs have been marked with numbers. The BLOB features *area* and *compactness* have been computed for the five BLOBs. A reference BLOB has an area of 10 pixels and a compactness of 0.5. The Euclidean distance in feature space has been computed between the five BLOBs and the reference BLOB. Which of the five BLOBs has the minimum distance?



1

2

3

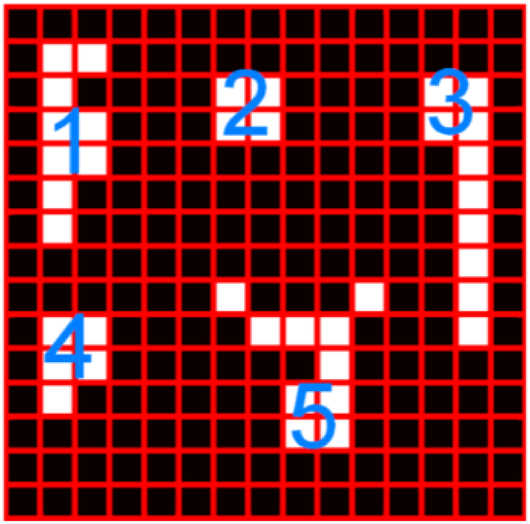
4

5

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

BLOB Classification

A BLOB analysis using 8-connectivity has been performed on the image seen in Figure 12 and the five found BLOBs have been marked with numbers. The BLOB features *area* and *compactness* have been computed for the five BLOBs. A reference BLOB has an area of 10 pixels and a compactness of 0.5. The Euclidean distance in feature space has been computed between the five BLOBs and the reference BLOB. Which of the five BLOBs has the minimum distance?



1

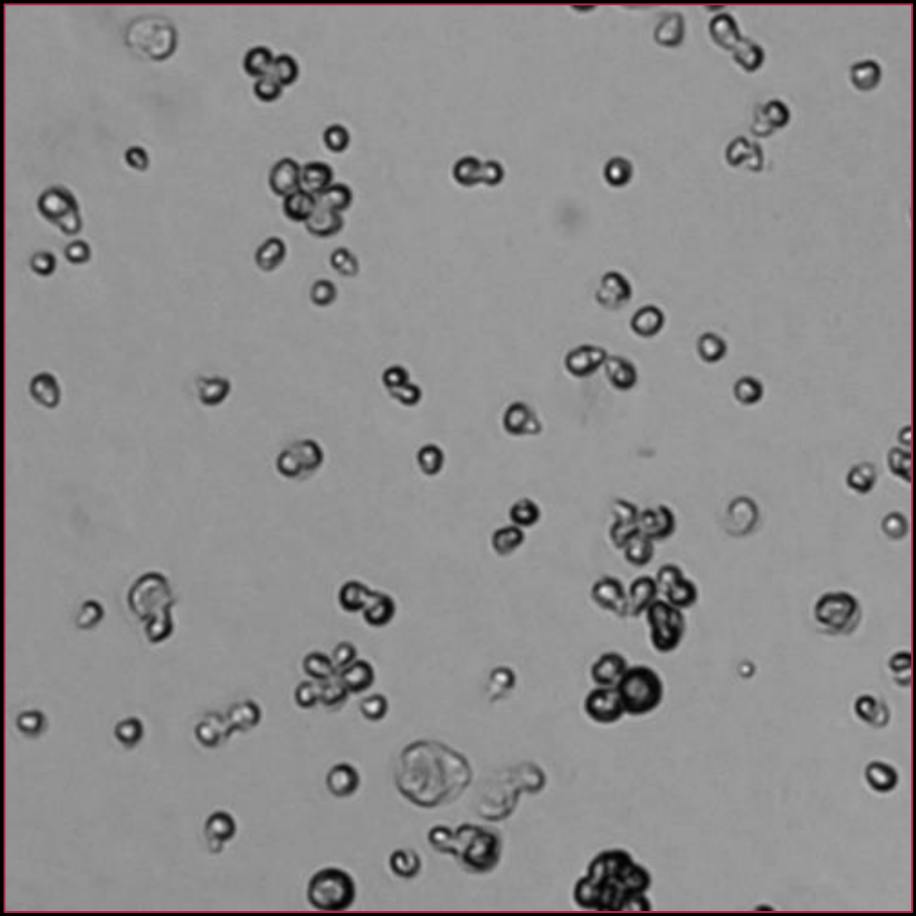
2

3

4

5 ✓ 0%

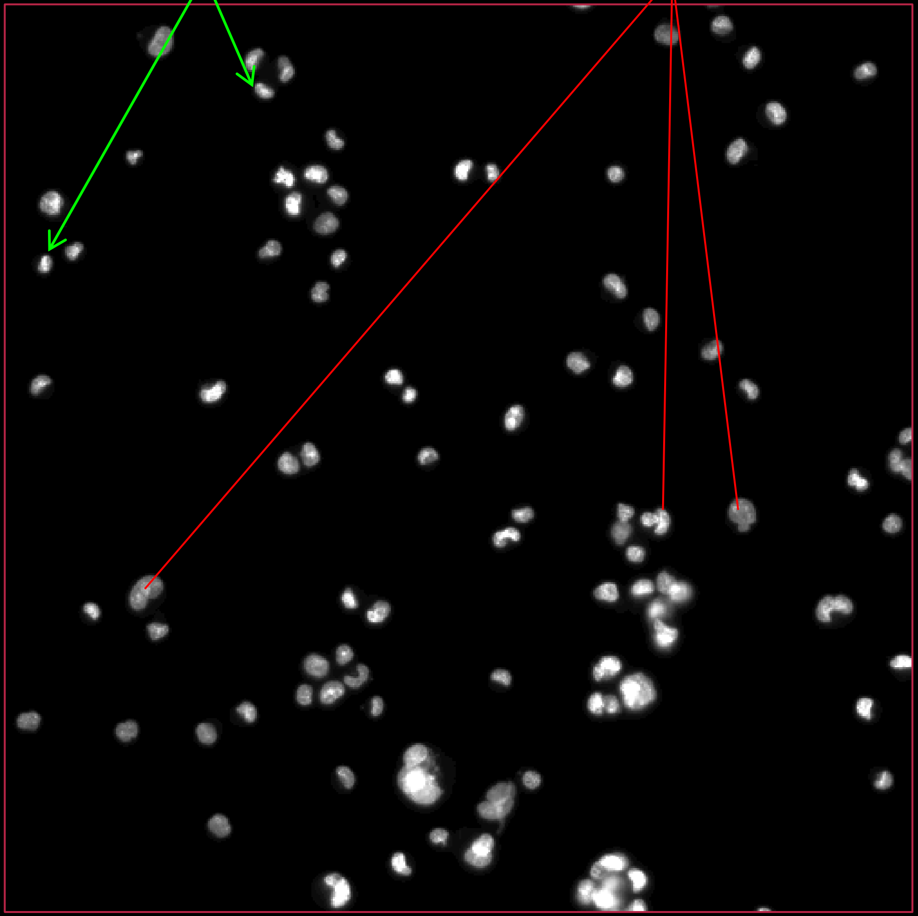
Cell classification



UV Microscopy

Single Nuclei

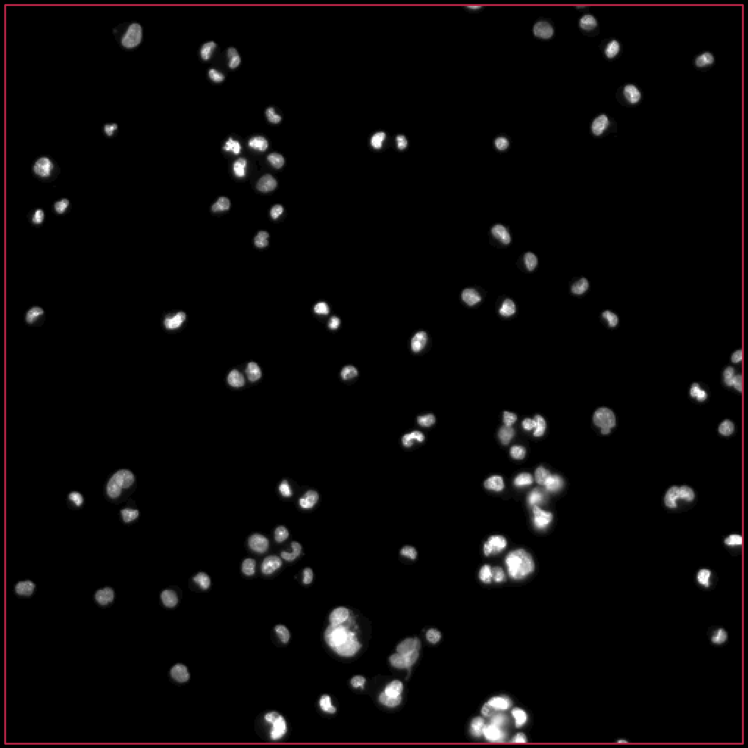
Multiple Nuclei



Fluorescence Microscopy (DAPI)

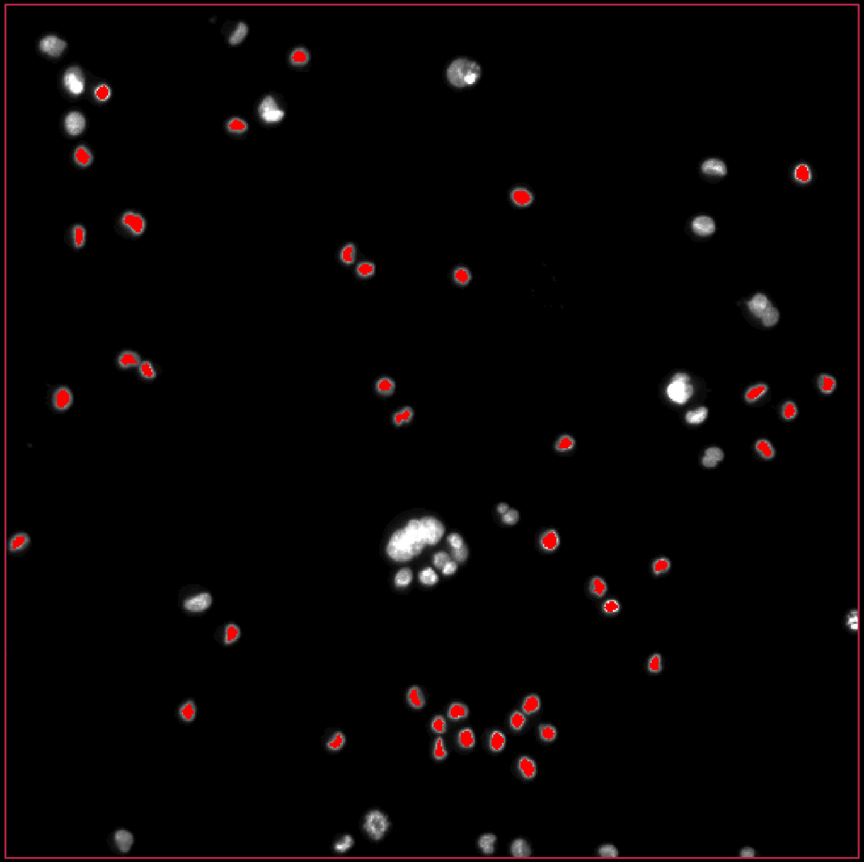
Images from ChemoMetec A/S

Nuclei classification



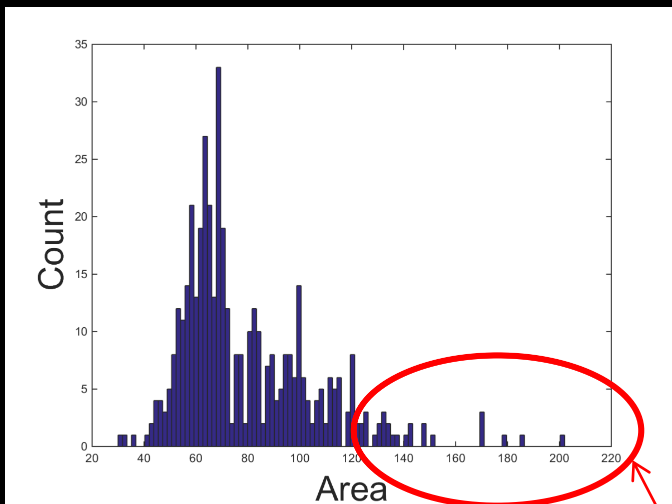
- DAPI image
- Two classes
  - Single nuclei
  - Noise
    - Multiple nuclei together
    - Debris
    - Other noise

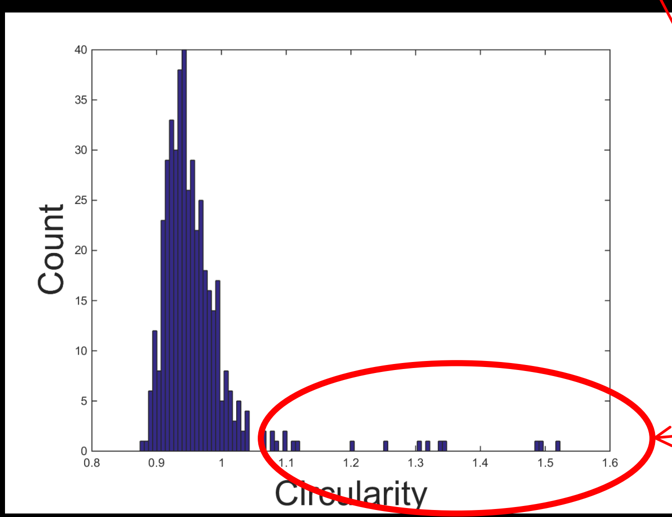
Training and annotation



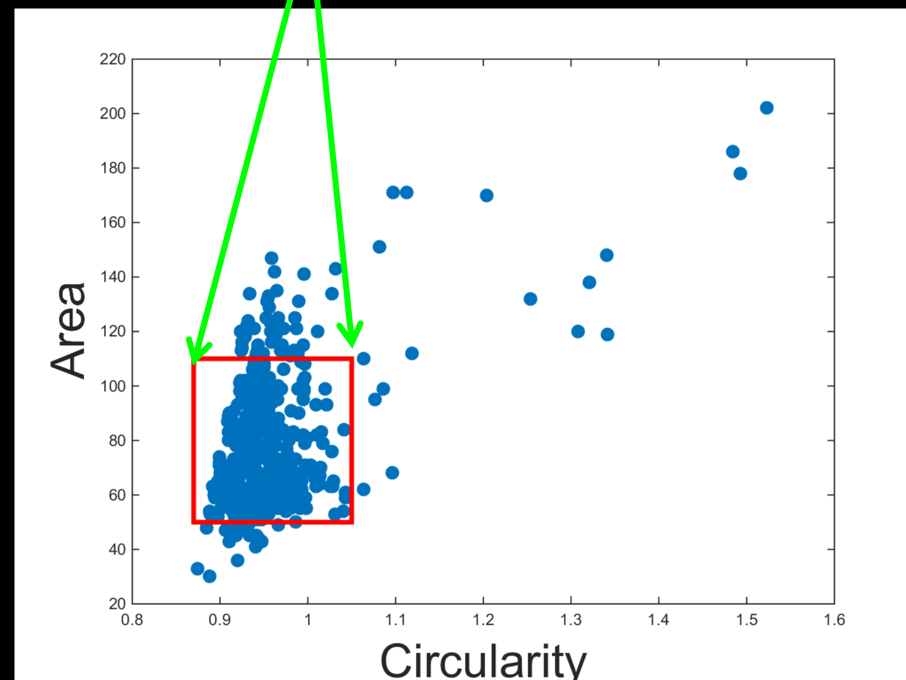
- Selection of true single nuclei marked
- Thresholding
- BLOB Analysis
  - Circularity
  - Area

Training data - analysis



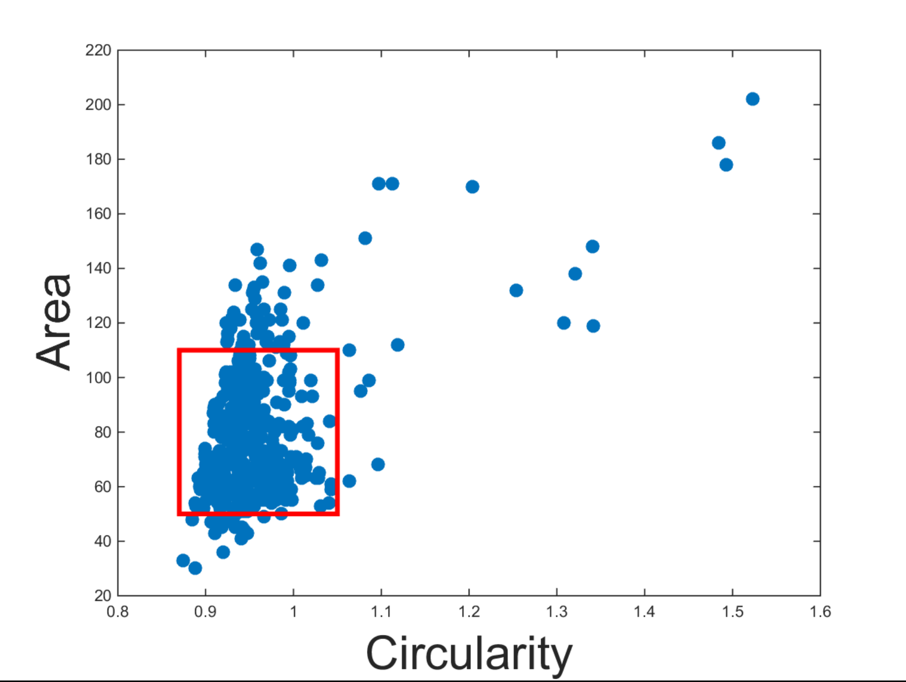


Acceptance area



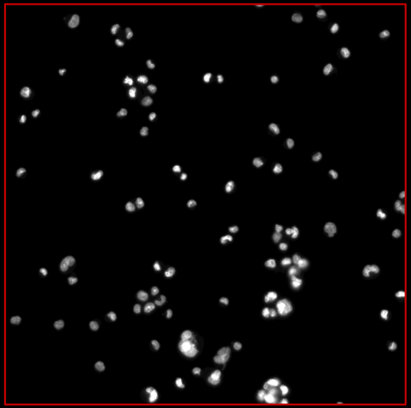
Probably outliers

Feature ranges



Feature	Min	Max
Area	50	110
Circularity	0.87	1.05

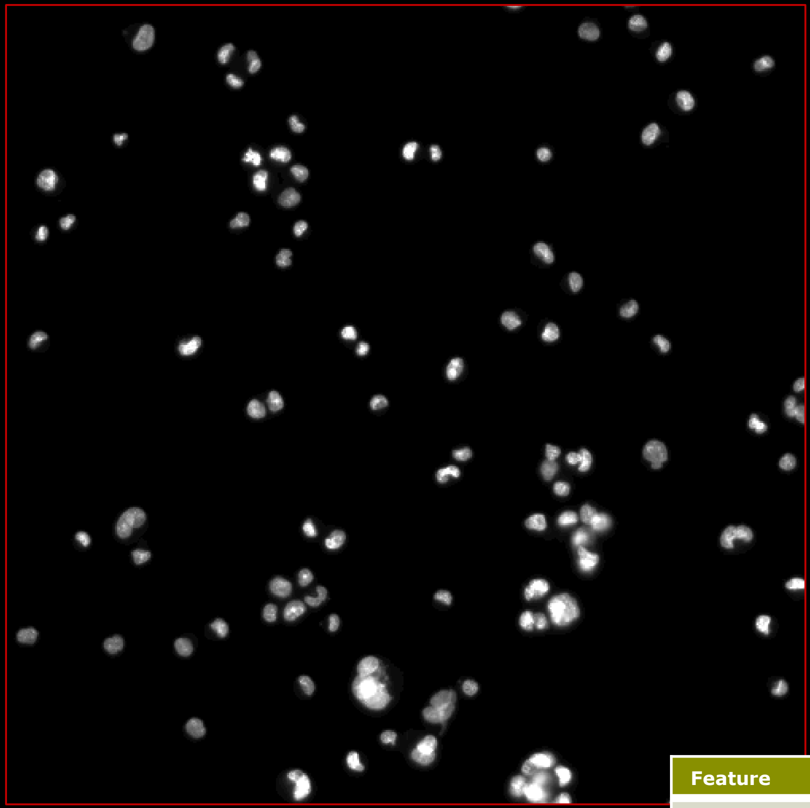
Using the classifier



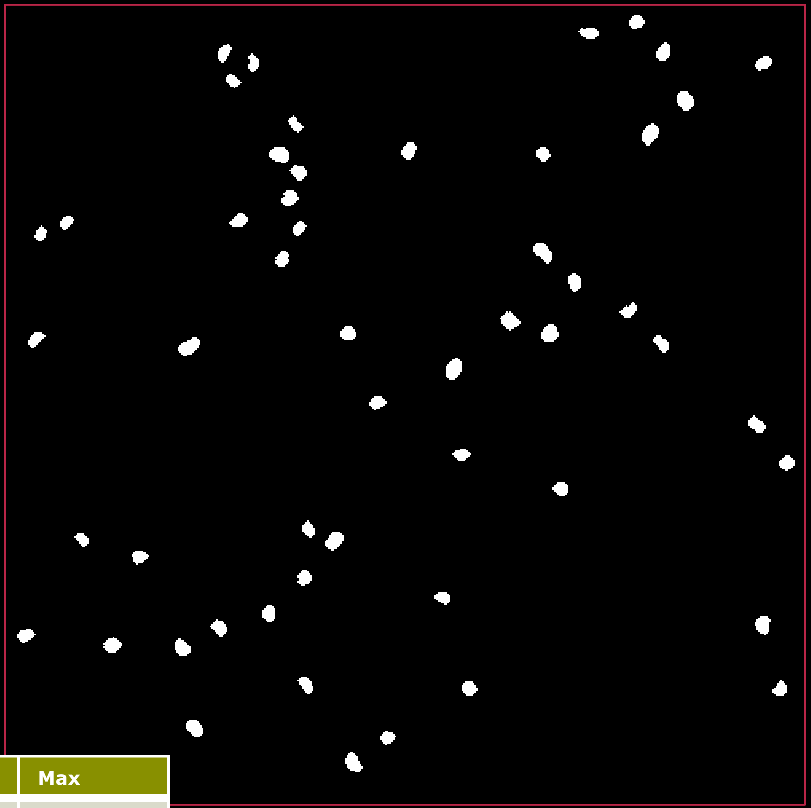
DAPI input image

- Threshold input image
- Morphological opening (SE 5x5)
- Morphological closing (SE 5x5)
- BLOBs found using 8-neighbours
- Border BLOBS removed
- Border features computed
  - Area + circularity
- BLOBs with features inside the acceptance range are single-nuclei

Using the classifier



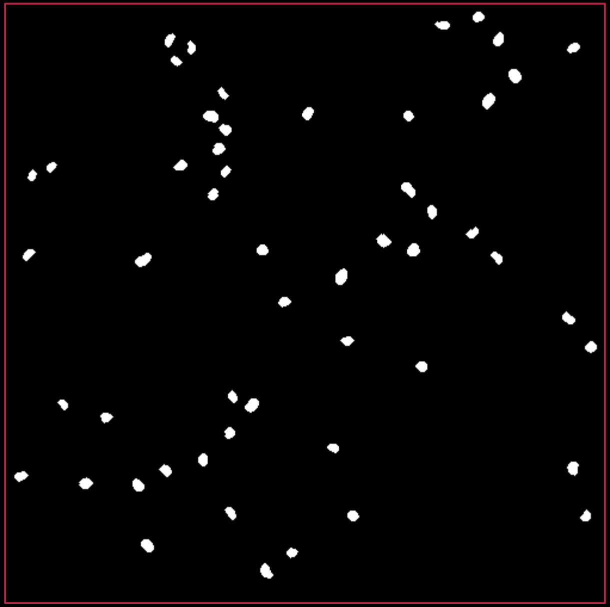
DAPI input image



Found single nuclei

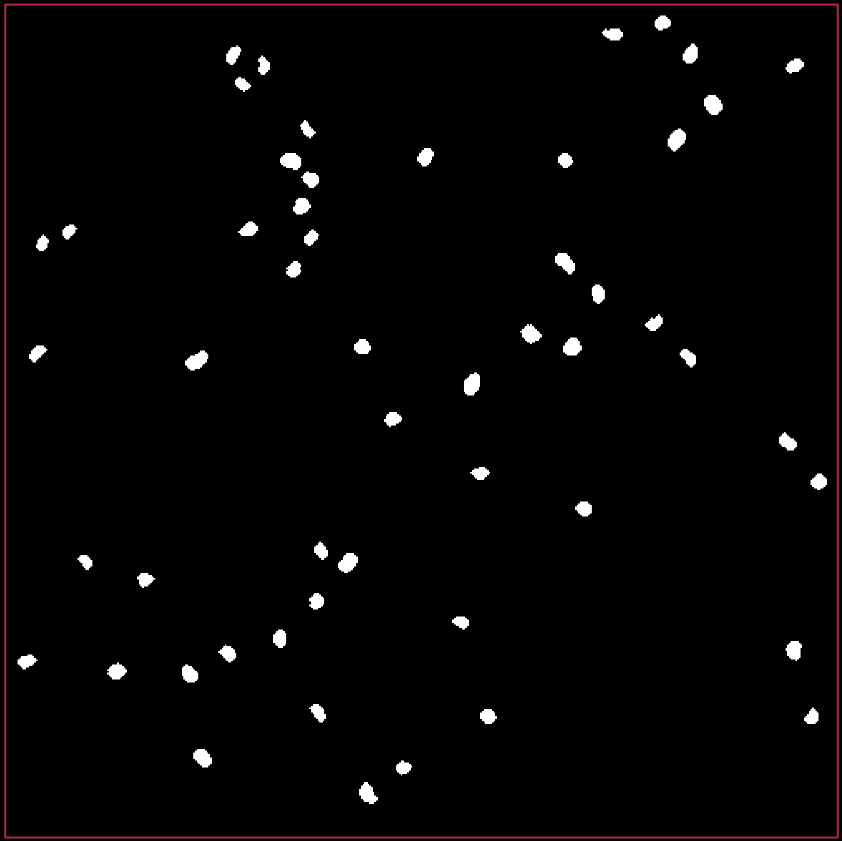
Feature	Min	Max
Area	50	110
Circularity	0.87	1.05

How well does it work?

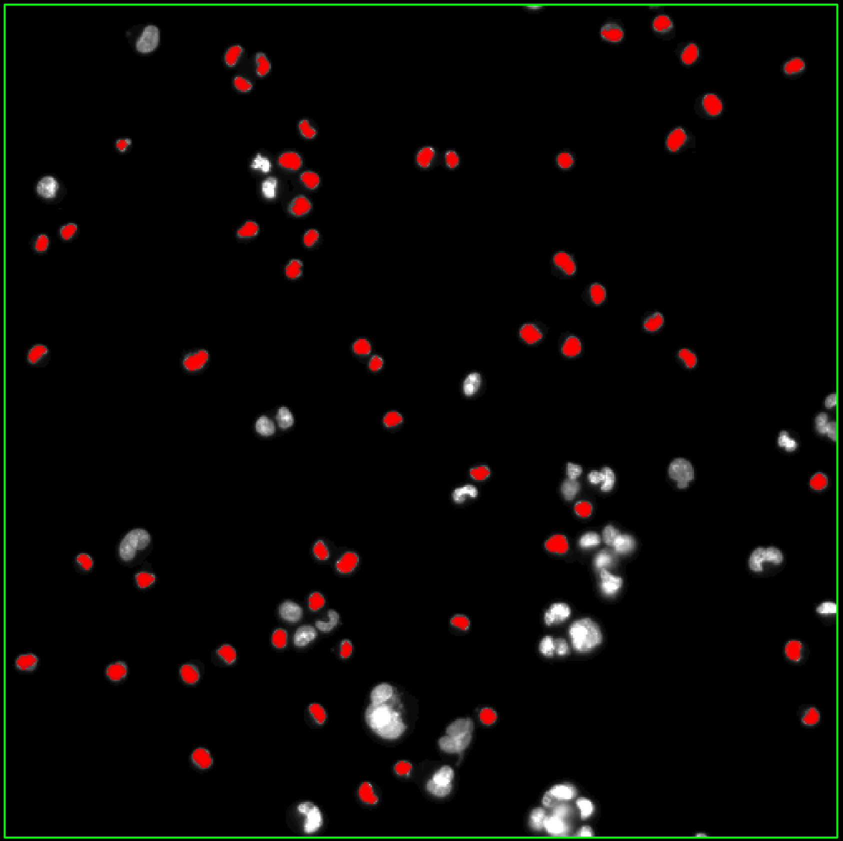


- We say we have a great algorithm!
- Strangely the doctor/biochemist do not trust this statement!
  - They need numbers!
- How do we report the performance?

Creating ground truth – expert annotations



Found single nuclei



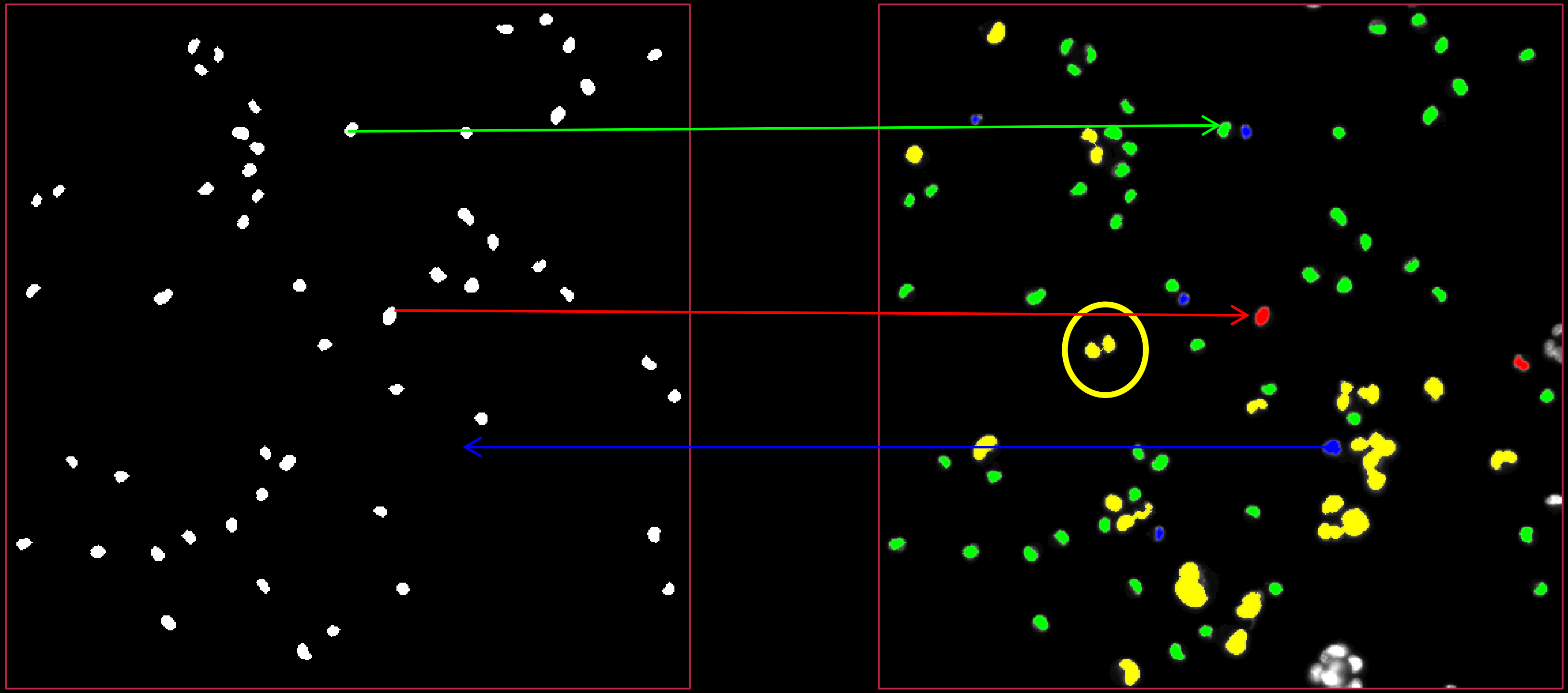
Expert opinion on true single nuclei

Red markings: Single nuclei

Not marked: Noise

Four cases

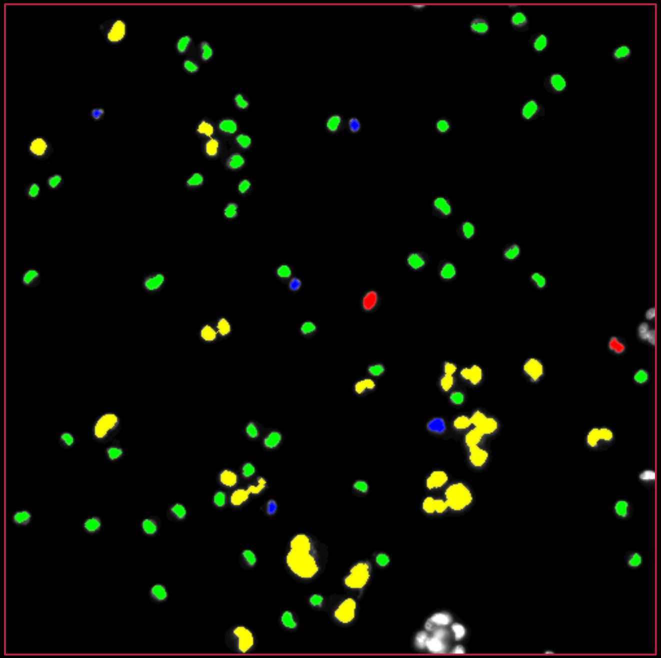
- True Positive (TP): A nuclei is classified as a nuclei
- True Negative (TN): A noise object is classified as noise object
- False Positive (FP): A noise object is classified as a nuclei
- False Negative (FN): A nuclei is classified as a noise object



Found single nuclei

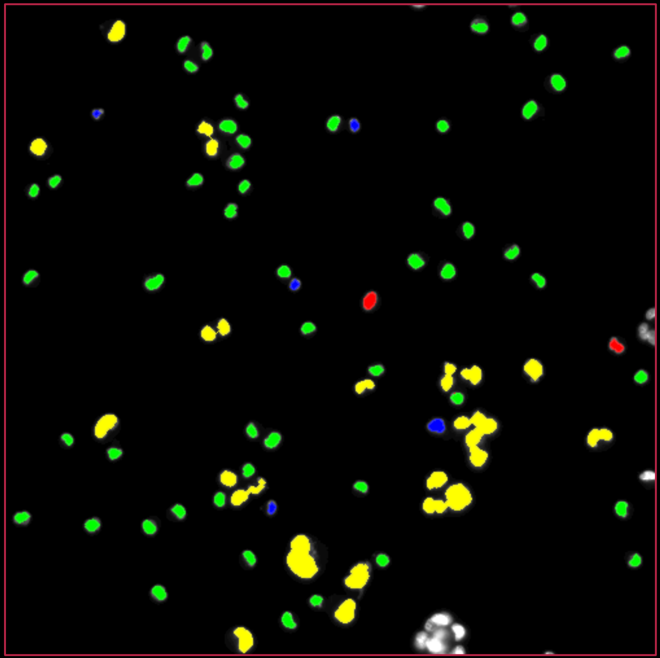
Confusion matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise		
Actual single-nuclei		



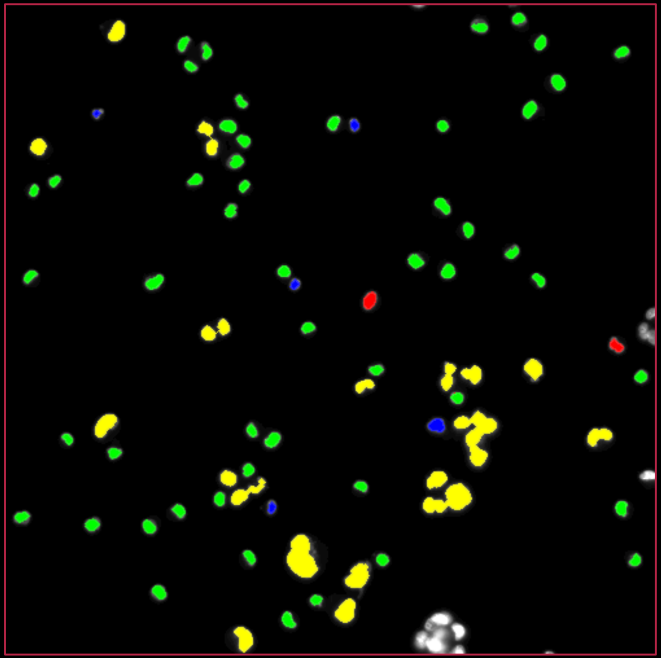
Confusion matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	
Actual single-nuclei		



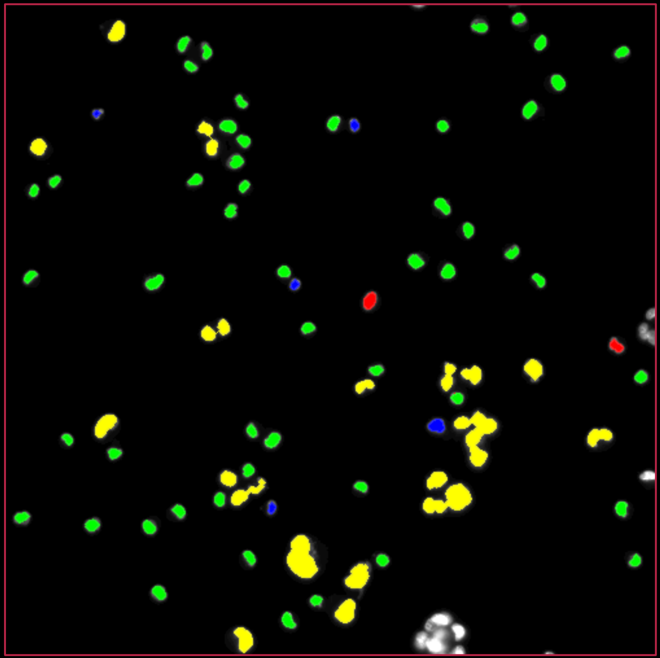
Confusion matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	
Actual single-nuclei		TP=51



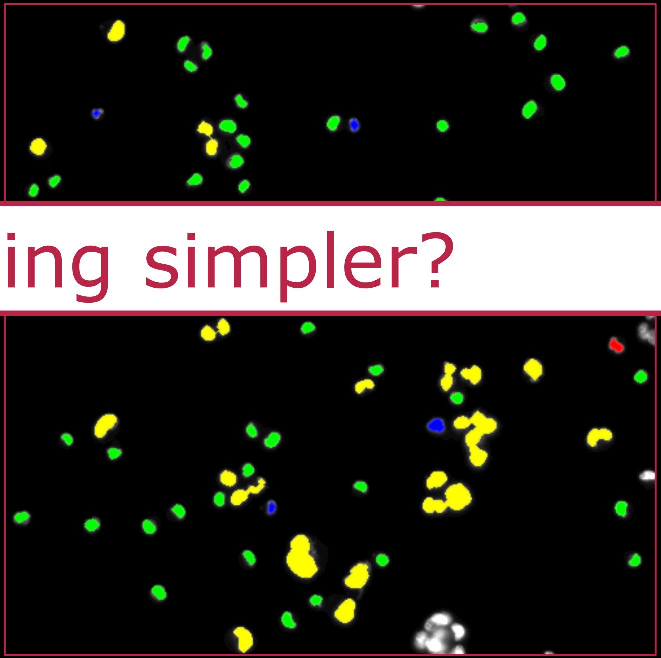
Confusion matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei		TP=51



Confusion matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51



Something simpler?

Accuracy

- Tells how often the classifier is correct

$$\text{Accuracy} = \frac{TP + TN}{N}$$

- N is the total number of annotated objects

$$N = TN + TP + FP + FN$$

Accuracy from Confusion Matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51

42%

65%

77%

91%

97%

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Accuracy from Confusion Matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51

42%

65%

77%

91%

97%

Accuracy from Confusion Matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51

42%

65%

77%

91% ✓0%

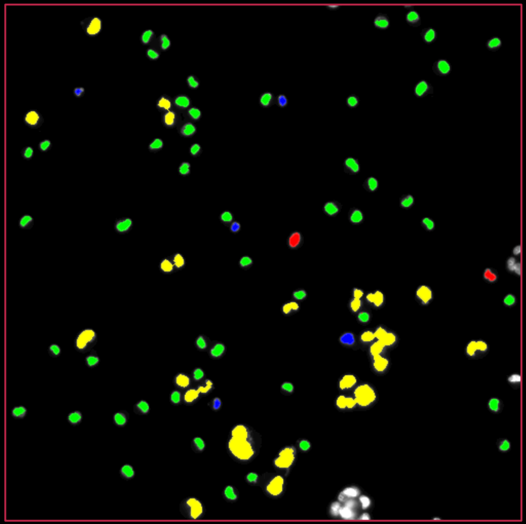
97%

True positive rate (sensivity)

- How often is a positive predicted when it actually is positive

$$\text{Sensivity} = \frac{TP}{FN+TP}$$

All the experts true single-nuclei



Sensitivity from Confusion Matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51

62%

65%

71%

91%

93%

Sensitivity from Confusion Matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51

62%

65%

71%

91%

93%

Sensitivity from Confusion Matrix

	Predicted as noise	Predicted as single-nuclei
Actual noise	TN=19	FP=2
Actual single-nuclei	FN=5	TP=51

62%

65%

71%

91% ✓ 0%

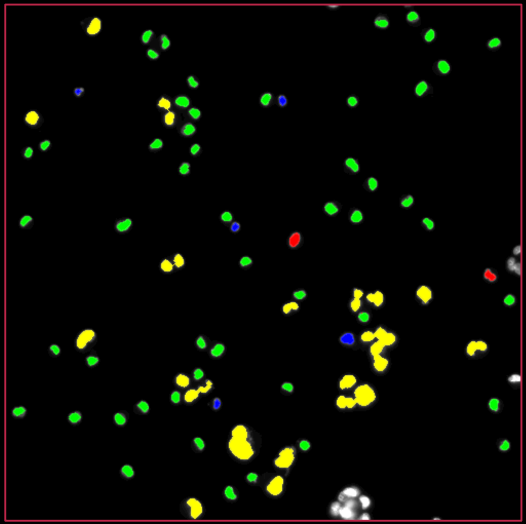
93%

Specificity

- How often is a negative predicted when it actually is negative

$$\text{Specificity} = \frac{TN}{TN+FP}$$

All the experts true noise objects



True positive rate

You have made an algorithm that can locate neon fish in an aquarium. An expert has marked all neon fish in an image as seen in Figure 1 (left). The result of your algorithm is seen in Figure 1 (right). What is the true positive rate of your algorithm?

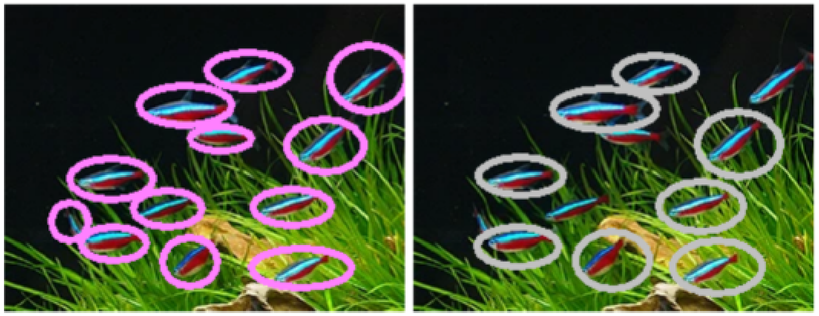


Figure 1: Image of aquarium with neon fish. Left: Expert markings are shown as ellipses. Right: Algorithm markings are shown as ellipses.

77%

92%

81%

55%

67%

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

True positive rate

You have made an algorithm that can locate neon fish in an aquarium. An expert has marked all neon fish in an image as seen in Figure 1 (left). The result of your algorithm is seen in Figure 1 (right). What is the true positive rate of your algorithm?

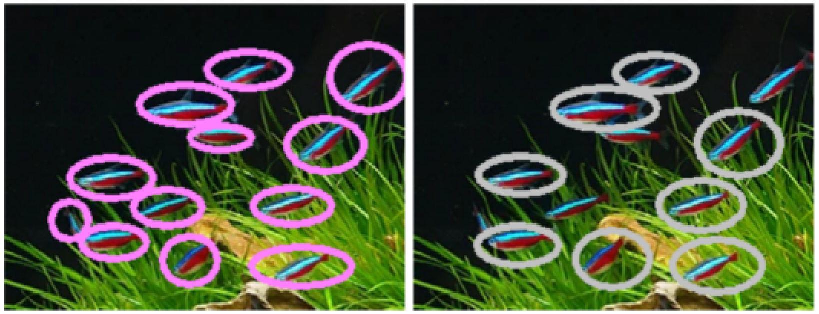


Figure 1: Image of aquarium with neon fish. Left: Expert markings are shown as ellipses. Right: Algorithm markings are shown as ellipses.

77%

92%

81%

55%

67%

True positive rate

You have made an algorithm that can locate neon fish in an aquarium. An expert has marked all neon fish in an image as seen in Figure 1 (left). The result of your algorithm is seen in Figure 1 (right). What is the true positive rate of your algorithm?

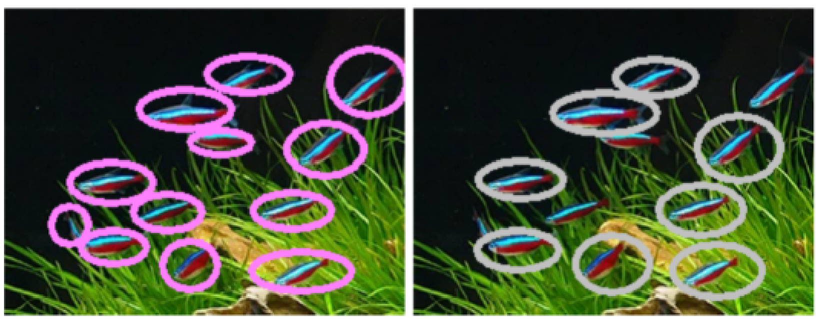


Figure 1: Image of aquarium with neon fish. Left: Expert markings are shown as ellipses. Right: Algorithm markings are shown as ellipses.

77%

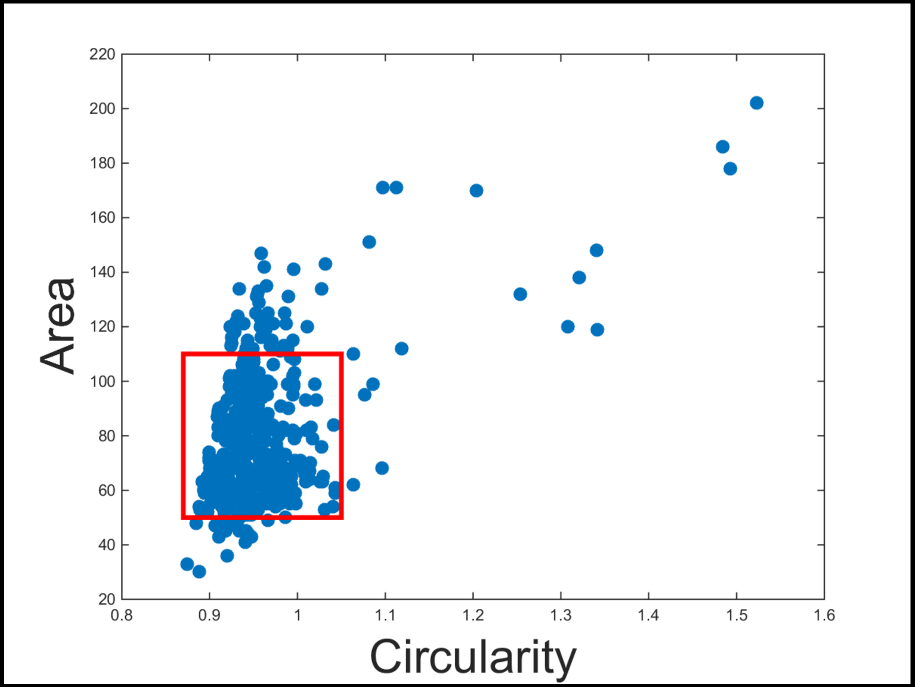
92%

81%

55%

67% ✓ 0%

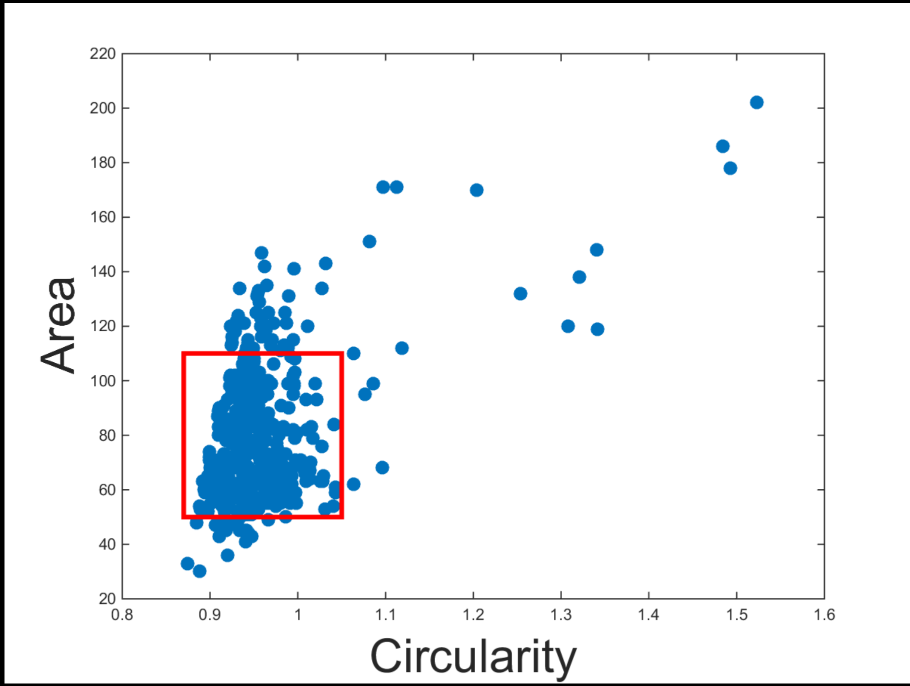
Optimising the classification



- Changing the classification limits
- The rates will be changed:
  - Accuracy
  - Sensitivity
  - Specificity
  - …
- Very dependent on the task what is optimal

Dependencies

- Increasing true positive rate
  - Increased false positive rate
  - Decreased precision



Example – cell analysis

- We want only single-nuclei cells
  - For further analysis
- We do not want to do an analysis of a noise object
- We are not interested in the true number of single nuclei

What measure is the most important?

- We want only single-nuclei cells
  - For further analysis
- We do not want to do an analysis of noise objects
- We are not interested in the true number of single nuclei

Low false positives

High true positives

High true negatives

Low false negatives

What measure is the most important?

- We want only single-nuclei cells
  - For further analysis
- We do not want to do an analysis of noise objects
- We are not interested in the true number of single nuclei

Low false positives

High true positives

High true negatives

Low false negatives

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

What measure is the most important?

- We want only single-nuclei cells
  - For further analysis
- We do not want to do an analysis of noise objects
- We are not interested in the true number of single nuclei

Low false positives ✓ 0%

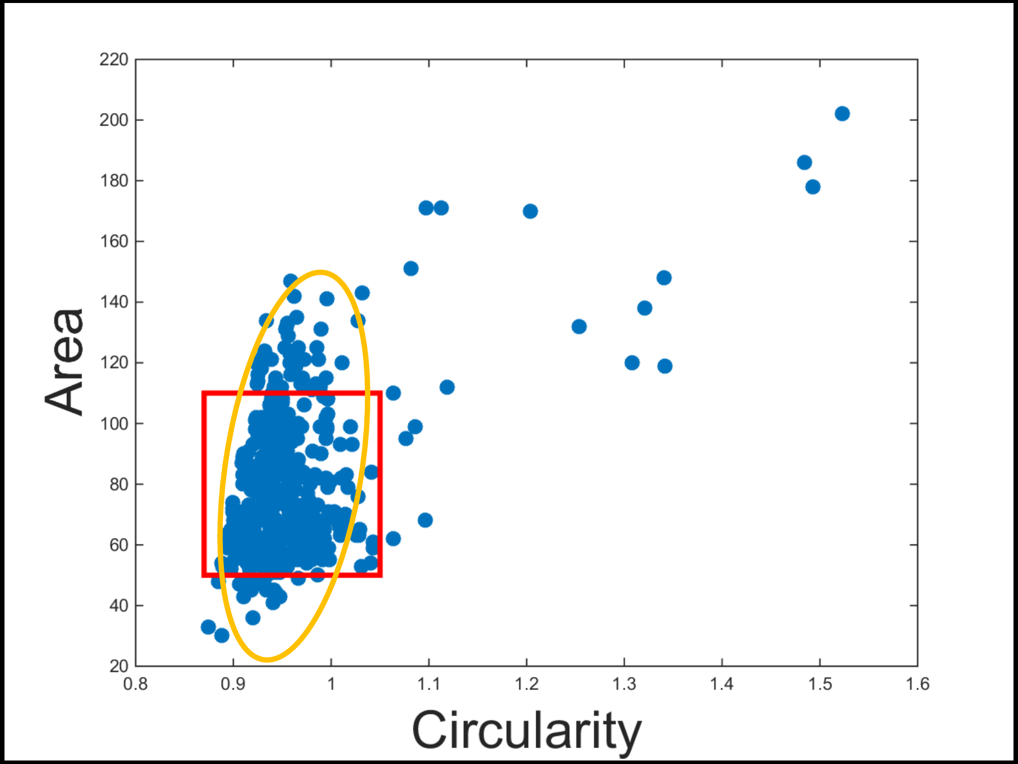
High true positives

High true negatives

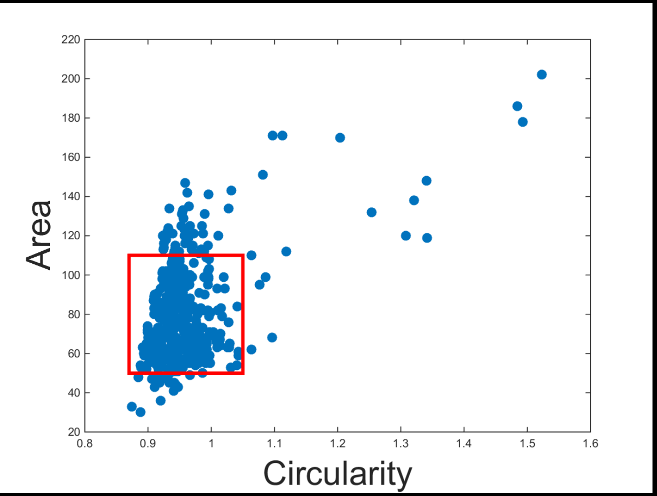
Low false negatives

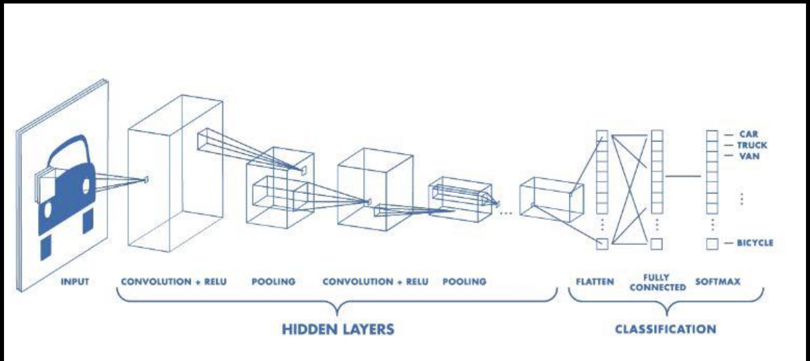
Advanced classification

- Fitting more advanced functions to the samples
- Multivariate Gaussians
- Mahalanobis distances



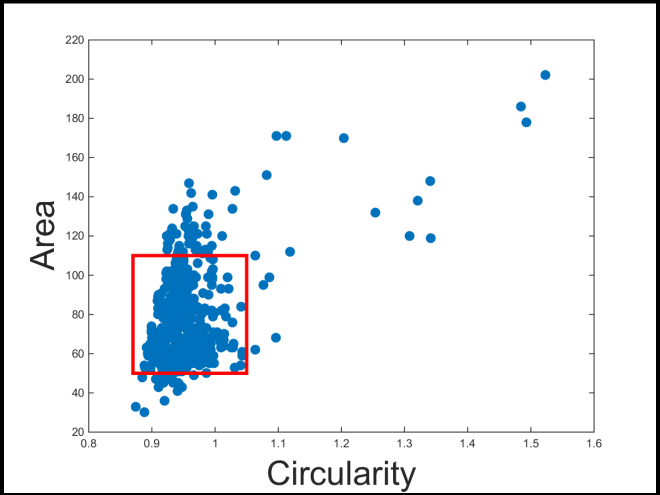
Feature Engineering vs. Deep learning





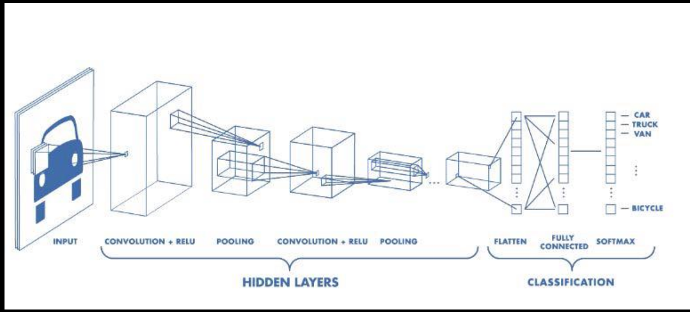
- Until around 5-7 years ago feature engineering was the way to go
- Now deep learning beats everything
- However – feature engineering is still important

Feature engineering



- Given a classification problem
  - Cars vs. Pedestrians
- Use background knowledge to select relevant features
  - Area
  - Shape
  - Appearance
  - …
- Use multivariate statistics to classify
- Depending on the selected features

Deep learning

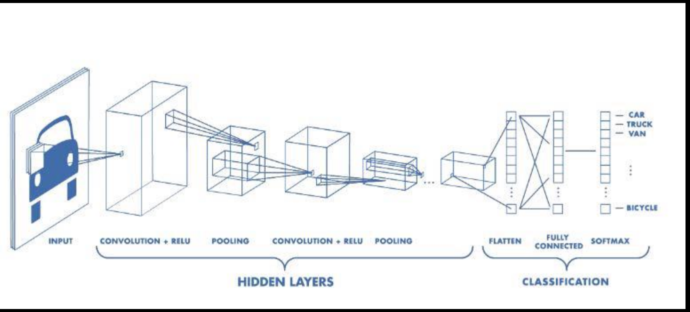


- You start with a dummy classifier
- Feed it with lots and lots of data with given labels
- The network learns the optimal features
- Layer/network engineering

Feature Engineering vs. Deep learning

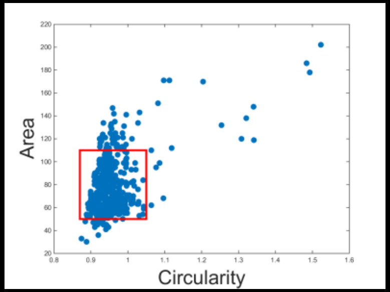
Deep Learning

- When you have lot of annotated data
- Where it is not clear what features work



Manual features

- When you have limited data
- When it is rather obvious what features can discriminate



The level of the lecture

Far too easy - my hamster could understand it

Too easy - I need more

Suitable - I am generally learning what I want

Too hard - slow down please

Far too hard - my head is exploding

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

The quizzes

Not enough quizzes - I want more more

Fine with the quizzes - no more no less

Argghh! These quizzes...I want less

Next week

- Geometric transformations
- Landmark based registration

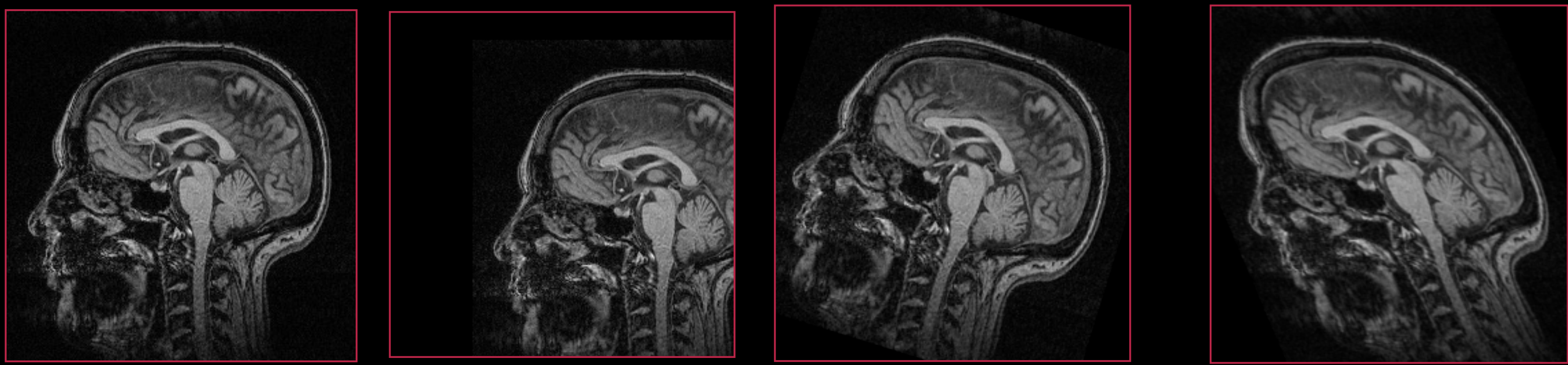
Image Analysis

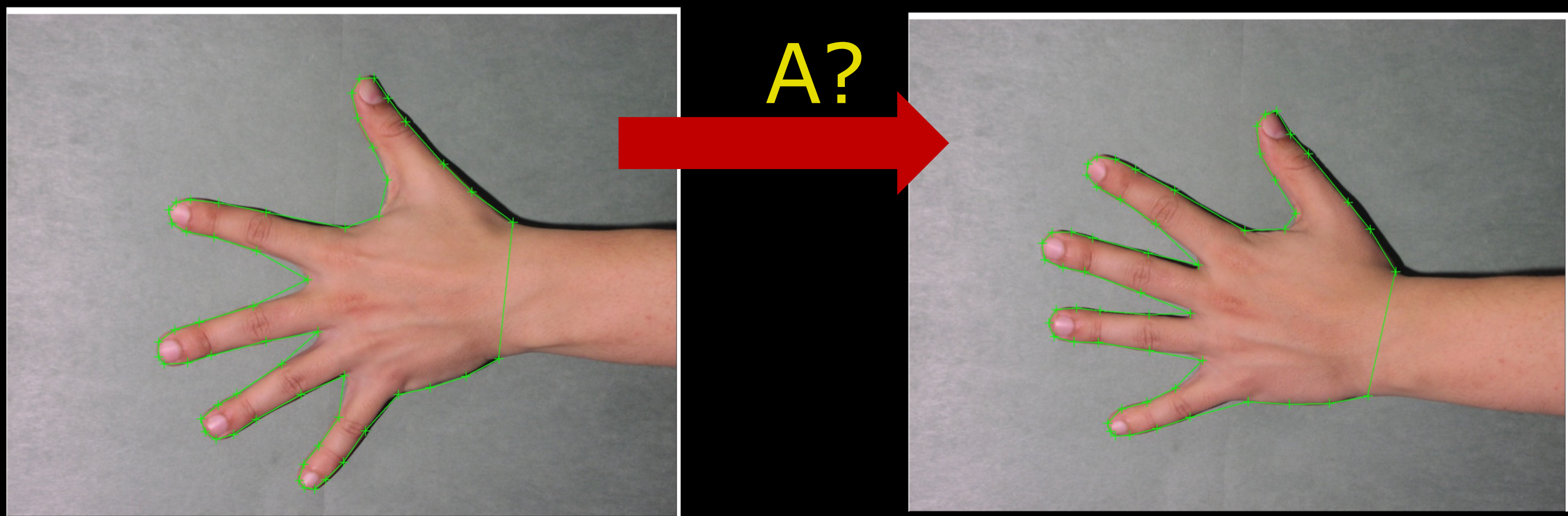
Tim B. Dyrby
Rasmus R. Paulsen
DTU Compute

tbdy@dtu.dk

http://www.compute.dtu.dk/courses/02502

Lecture 7 – Geometric Transformation and image registration





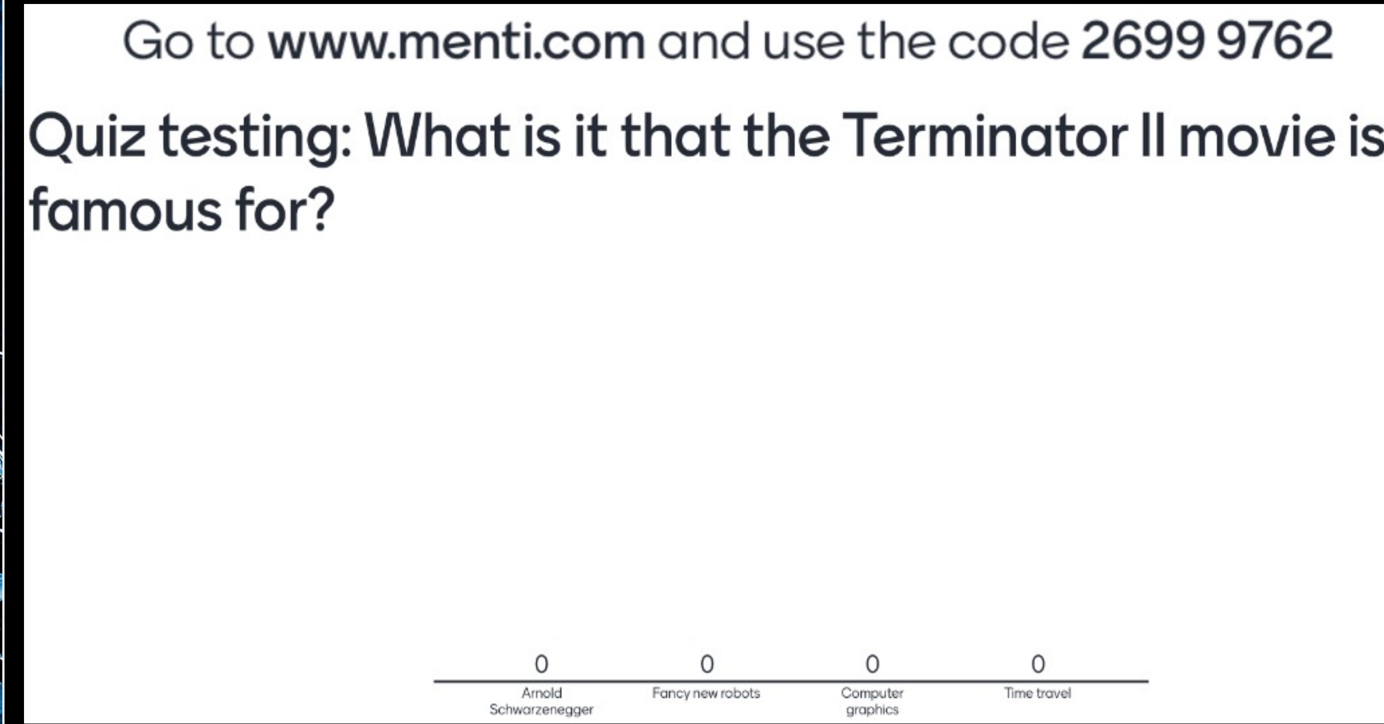
What can you do after today?

Geometrical transformation

- Construct a translation, rotation, scaling, and shearing transformation matrix of a point
- Use transformation matrices to perform point transformations
- Describe the difference between forward and backward mapping
- Transform an image using backward mapping and bilinear interpolation

Image registration

- Describe the image registration
- Describe the different types of landmarks
- Annotate a set of image using anatomical landmarks
- Describe the objective function used for landmark and joint histogram based registration
- Compute the optimal translation between two sets of landmarks
- Use the rigid body transformation for image registration
- Describe the general ”pipeline” for image registration



1) Arnold Schwarzenegger

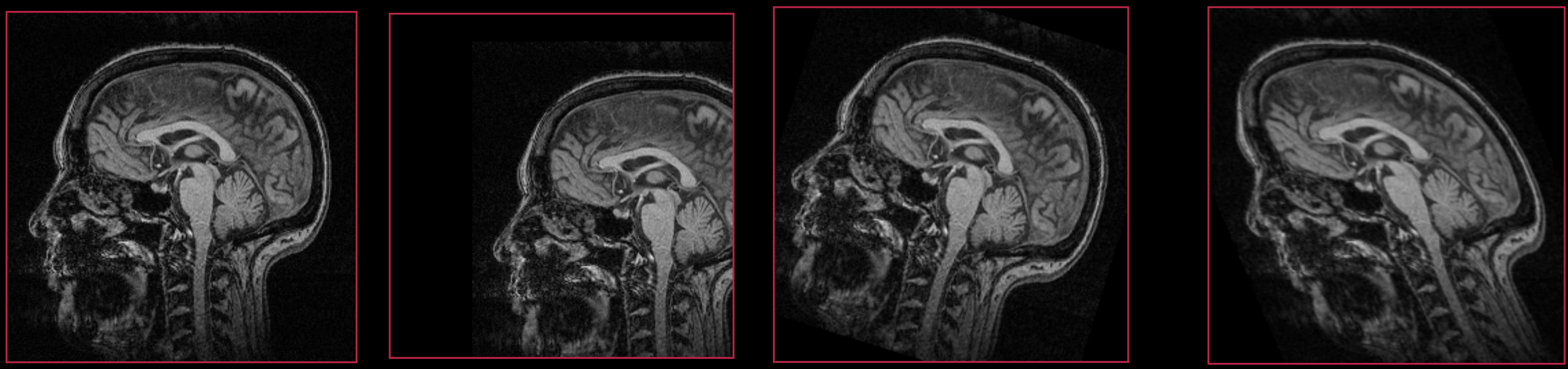
2) Fancy new robots

3) Computer graphics

4) Time travel

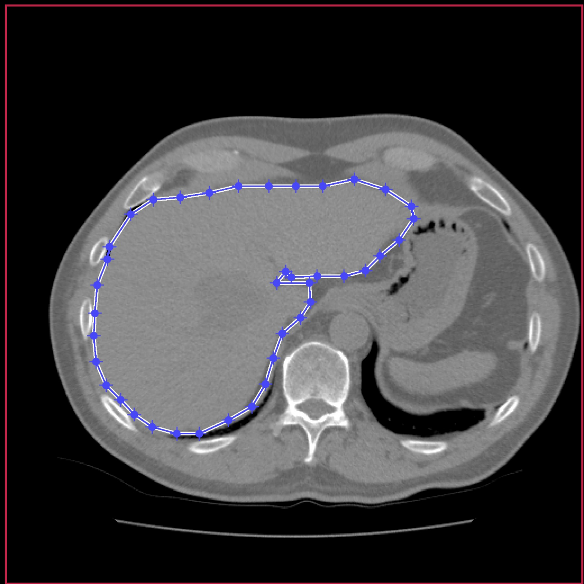
Geometric transformation

- Moving and changing the dimensions of images
- Why do we need it?

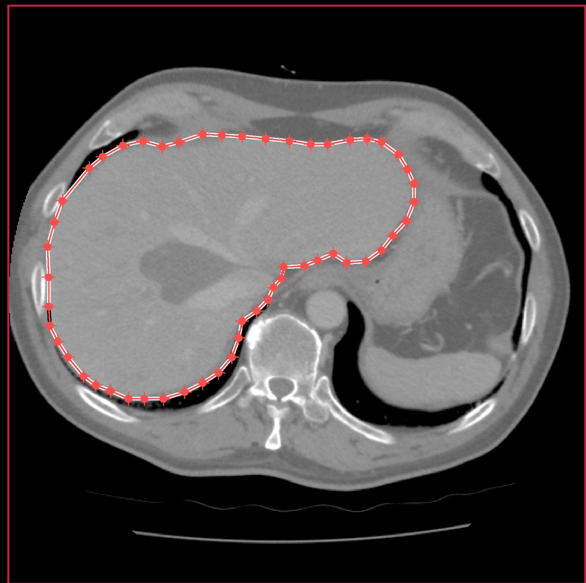


Change detection

- Patient imaged before and after surgery
- What are the changes in the operated organ?
- Patient can not be placed in the exact same position in the scanner



Before surgery

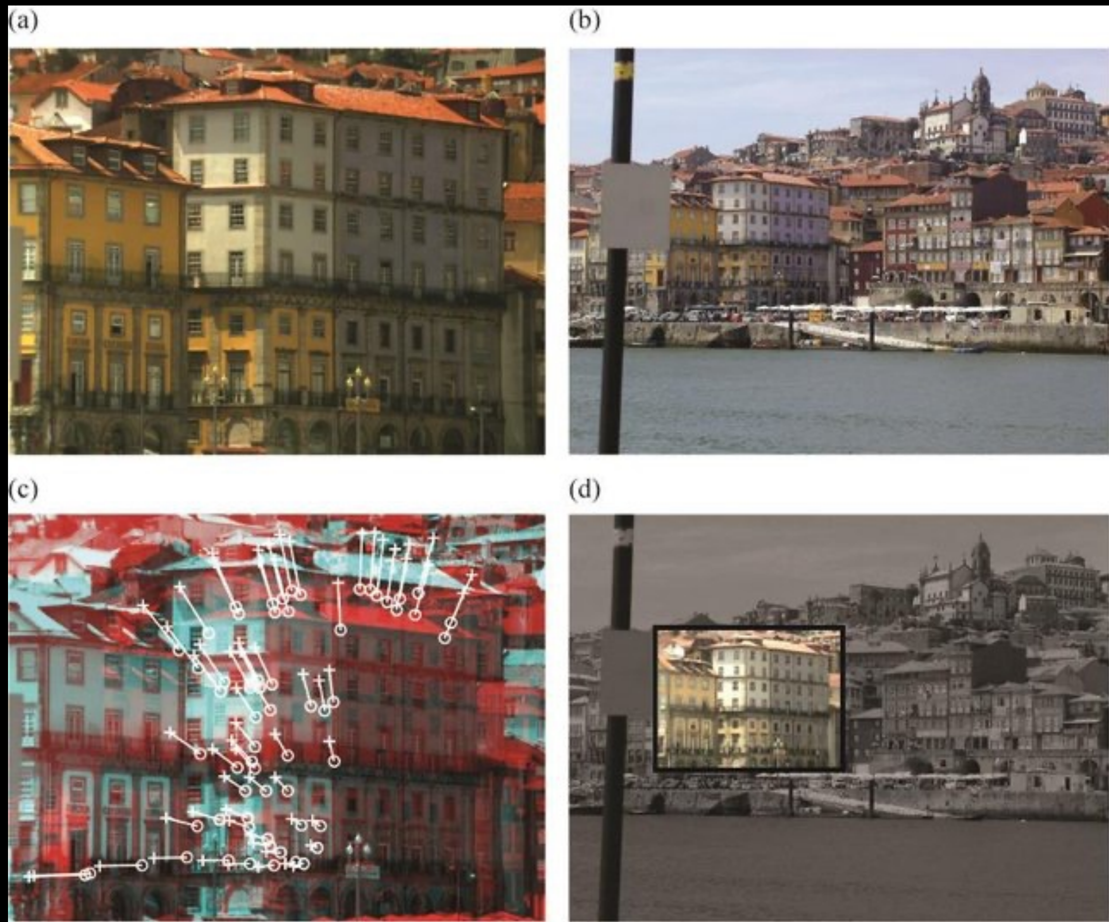


After surgery

Bachelor project: Image Guided Surgery Planning

Similarity transform

- Objects at different distances

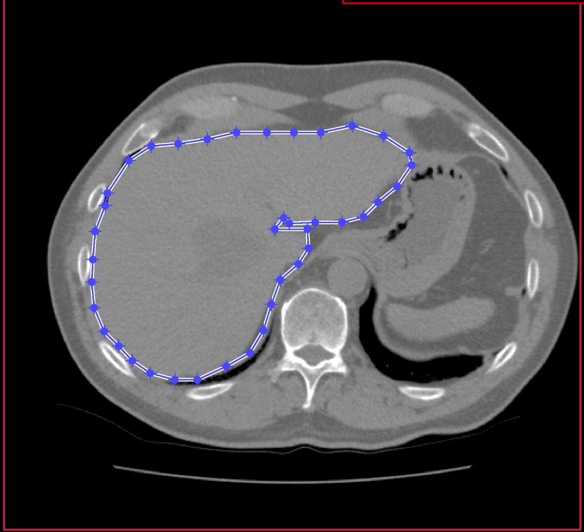


Amano et al 2016, DOI: 10.1051/matecconf/20166600024

Image Registration

- Change one of the images so it fits with the other
- Formally
  - Template image
  - Reference image
  - Template is moved to fit the reference

Compute the difference

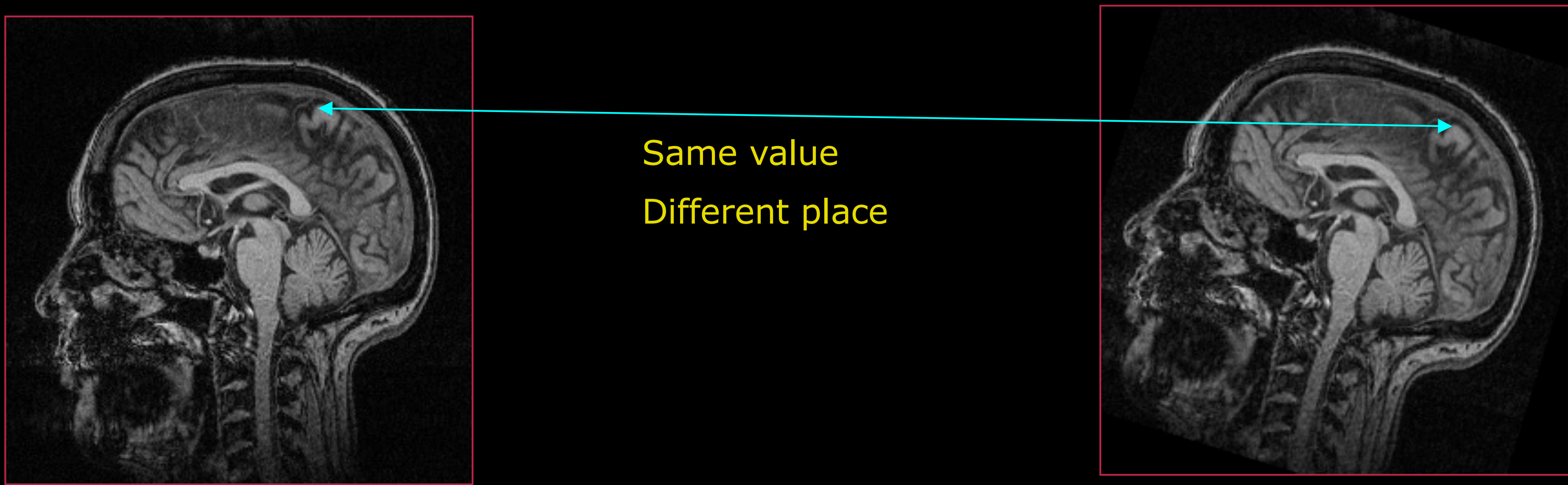


Template

Reference

Geometric Transform

- The pixel intensities are not changed
- The "pixel values" just change positions



Different transformations

- Translation
- Rotation
- Scaling
- Shearing

- Advanced transformations

From Terminator 2 movie: Non-linear image transformation

Translation

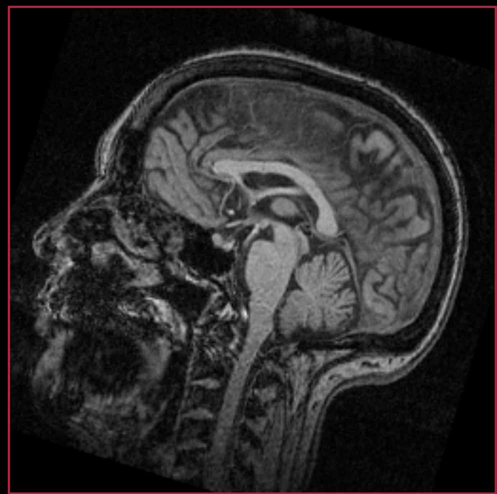
- The image is shifted – both vertically and horizontally

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 60 \\ 20 \end{bmatrix}$$

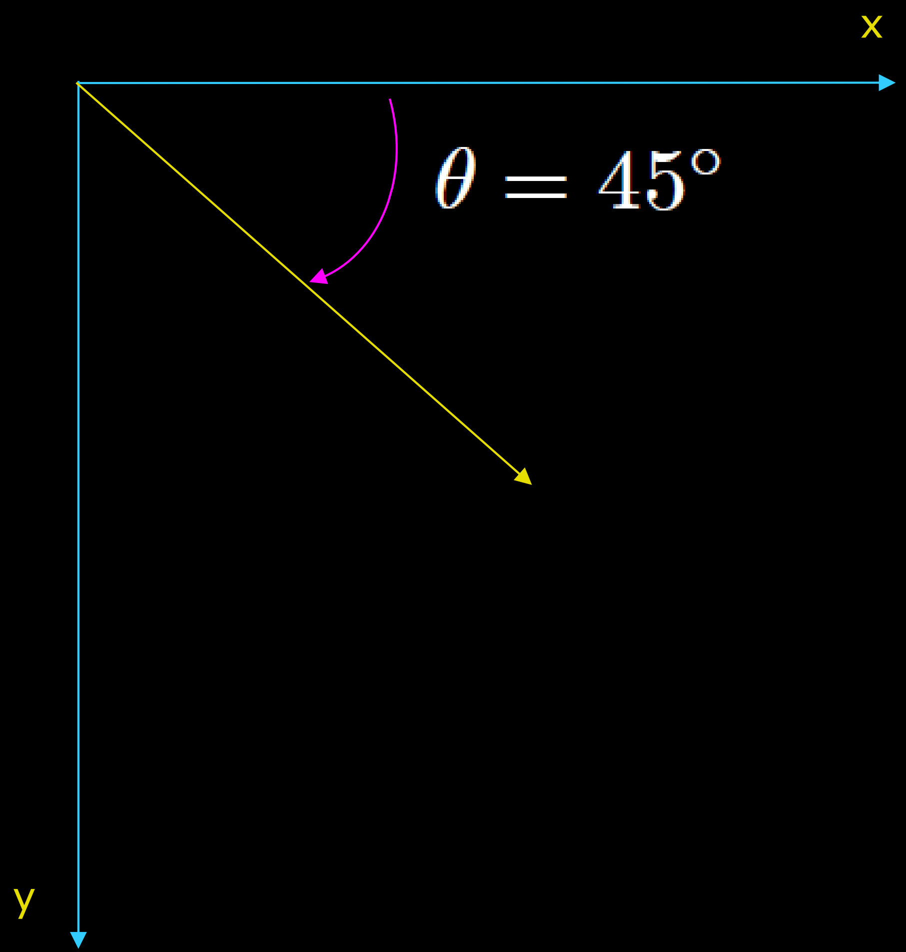
Rotation

- The image is rotated around the centre or the upper left corner
- Remember to use degrees and radians correctly
  - Matlab uses radians
  - Degrees easier for us humans

$$\theta = 15^\circ$$

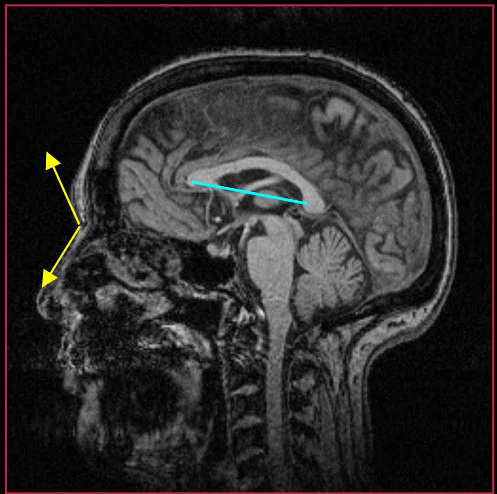


Rotation coordinate system



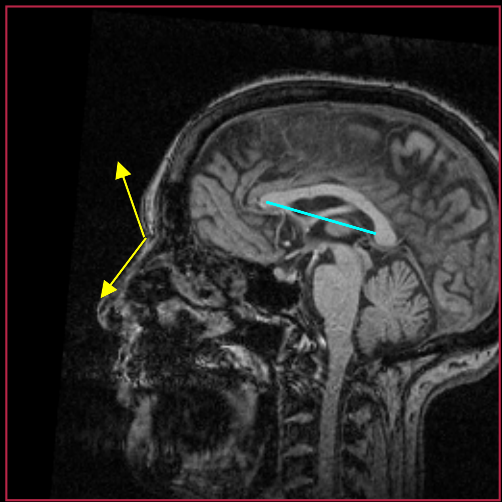
Rigid body transformation

- Translation and rotation
- Rigid body
- Angles and distances are kept



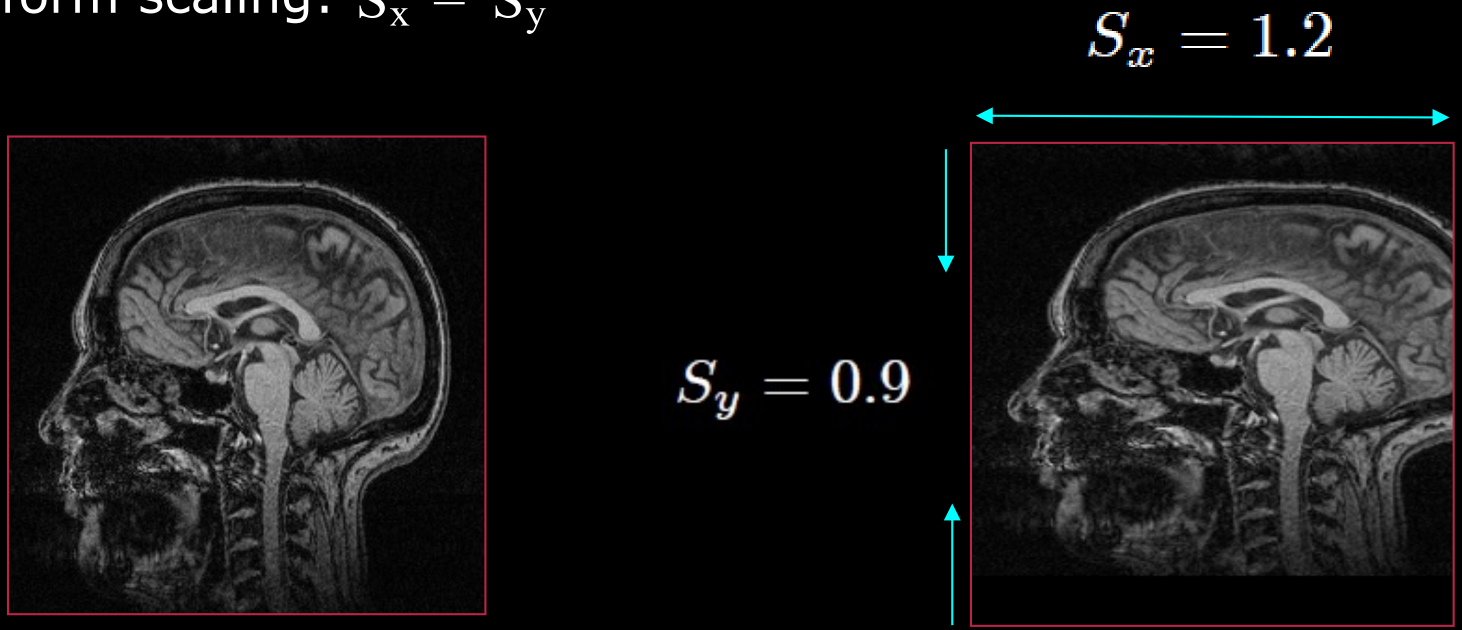
$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 60 \\ 20 \end{bmatrix}$$

$$\theta = 5^\circ$$



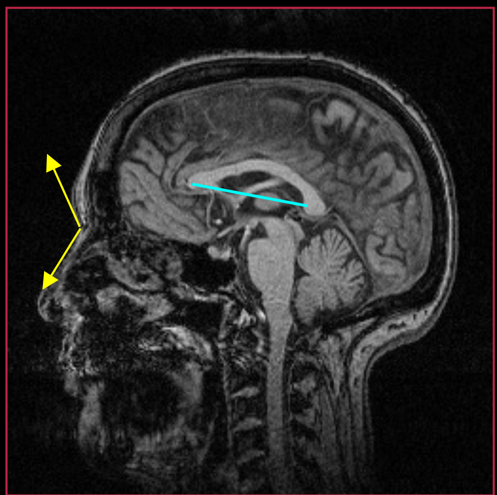
Scaling

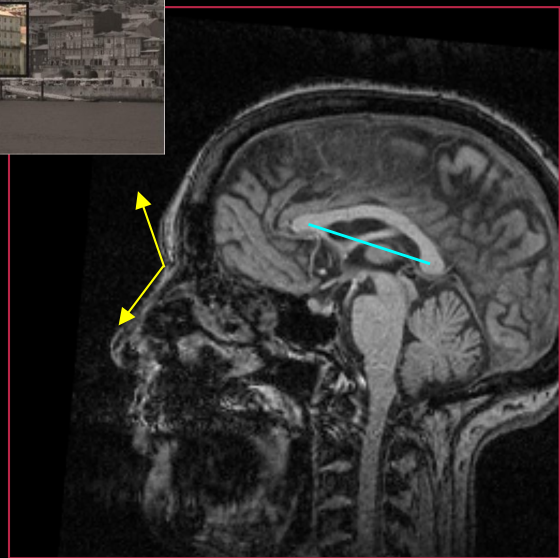
- The size of the image is changed
- Scale factors
  - X-scale factor S_x
  - Y-scale factor S_y
- Uniform scaling: $S_x = S_y$



Similarity transformation

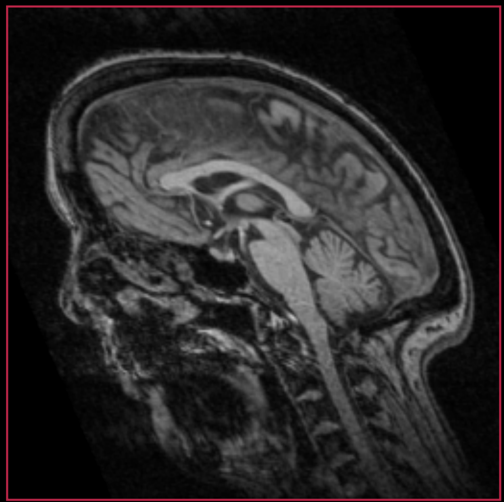
- Translation, and uniform scaling
- Angles are kept
- Distances change





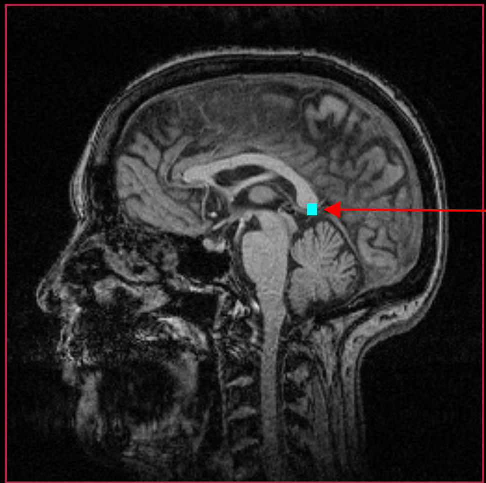
Shearing

- Pixel shifted horizontally or/and vertically
- Shearing factors
  - X-shear factor B_x
  - Y-shear factor B_y
- Is less used than translation, rotation, and scaling



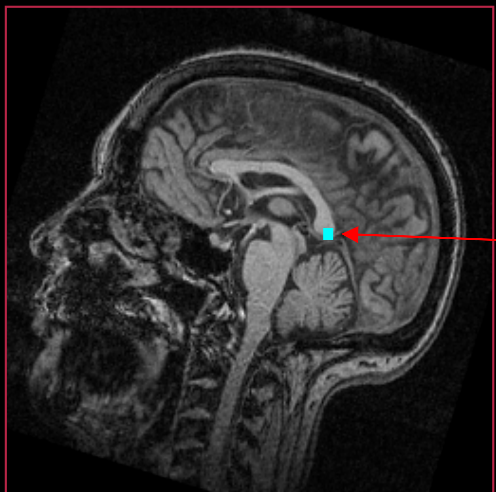
Transformation Mathematics

f



(x,y)

g



(x′,y′)

- Transformation of *positions*
- Structure found at position (x,y) in the input image f
- Now at position (x′,y′) in output image g
- A *mapping function* is needed

$$x' = A_x(x, y)$$
$$y' = A_y(x, y)$$

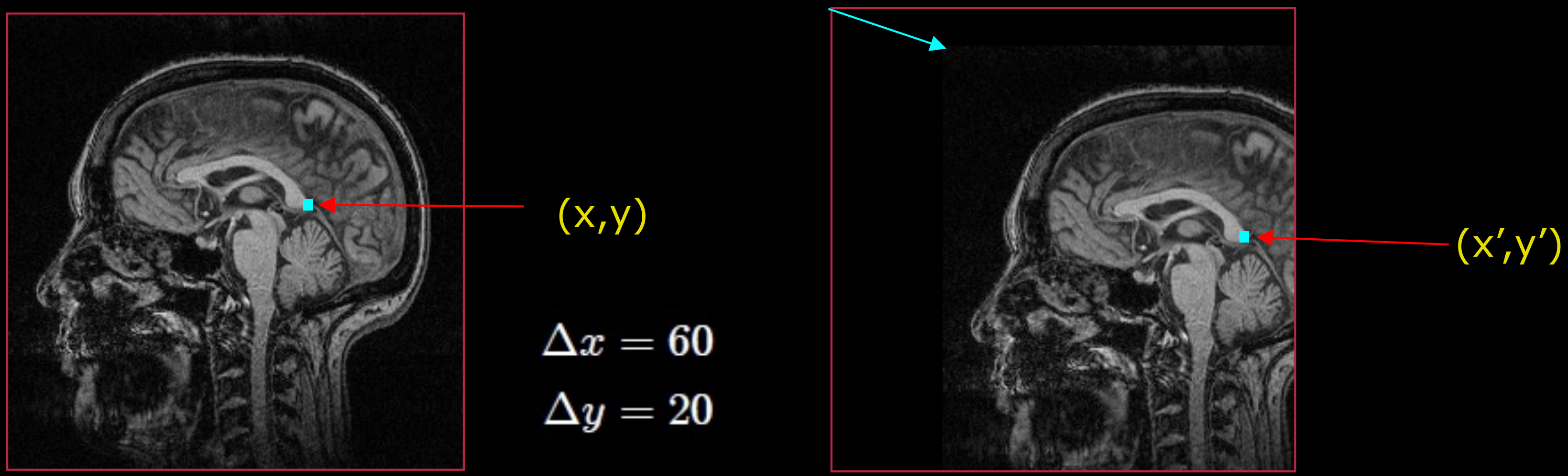
Depends on both x and y!

Translation mathematics

- The image is shifted – both vertically and horizontally

$$x' = x + \Delta x$$

$$y' = y + \Delta y$$



Matrix notation

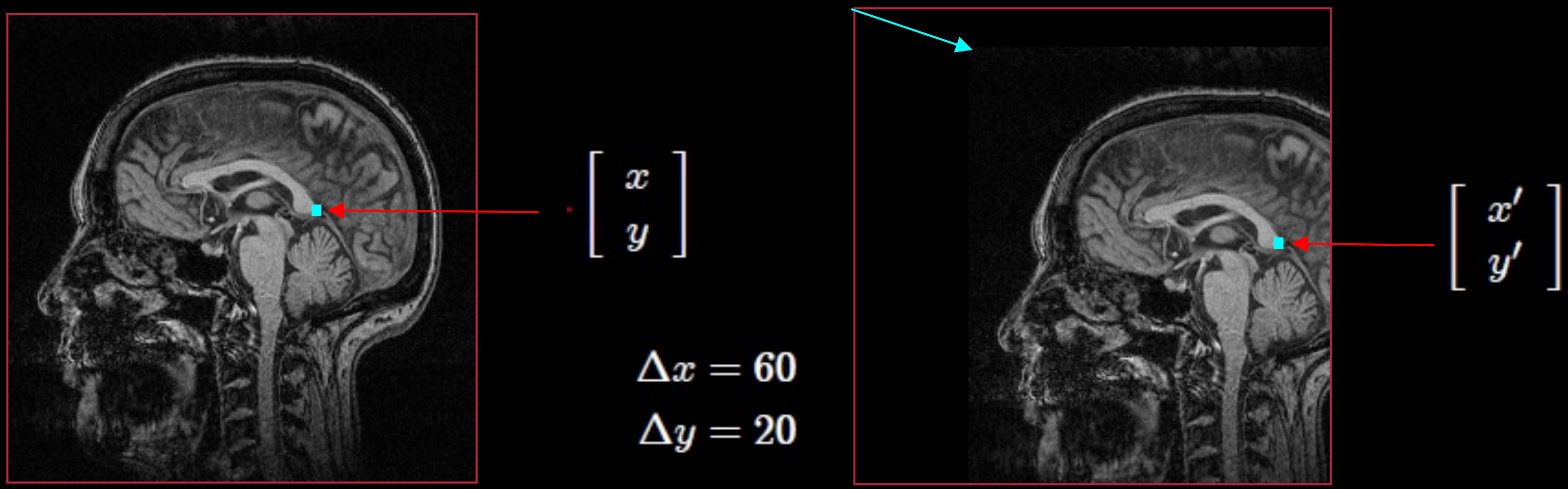
- Coordinates in column matrix format

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

Translation mathematics in matrix notation

- The image is shifted – both vertically and horizontally

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

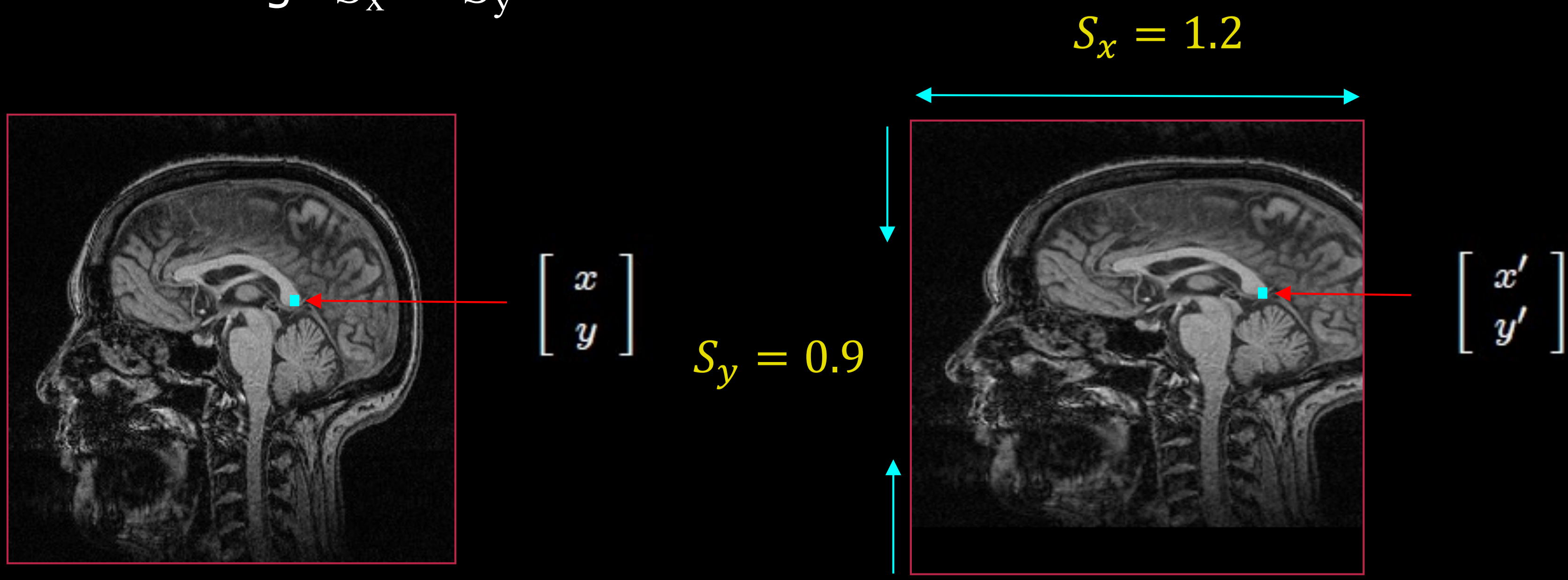


Scaling

- The size of the image is changed
- Scale factors
  - X-scale factor S_x
  - Y-scale factor S_y

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

- Uniform scaling: $S_x = S_y$



Matrix multiplication details

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Is equal to:

$$x' = x \cdot S_x$$
$$y' = y \cdot S_y$$

Transformation matrix

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

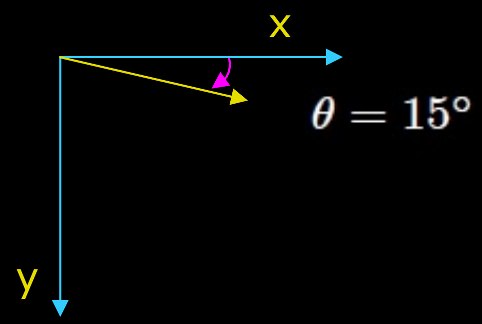
Can be written as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Where

$$\mathbf{A} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

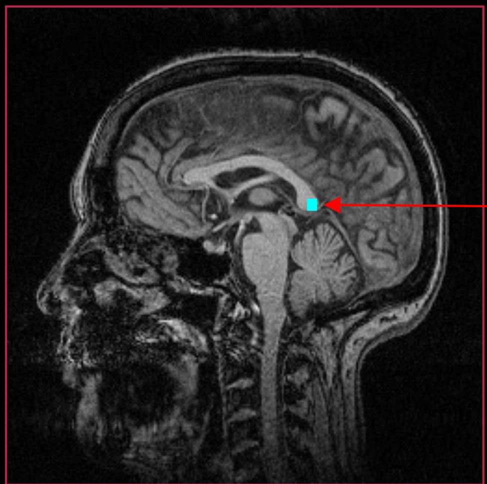
is a *transformation matrix*



Rotation

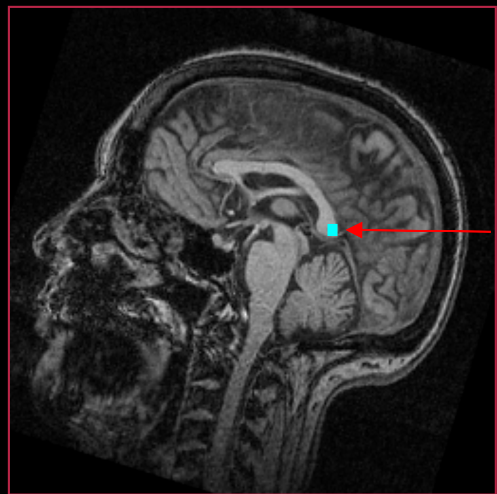
- A rotation matrix is used

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$



$$\begin{bmatrix} x \\ y \end{bmatrix}$$

$$\theta = 15°$$



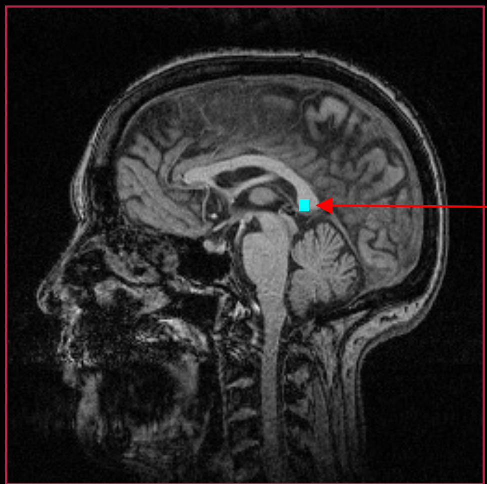
$$\begin{bmatrix} x' \\ y' \end{bmatrix}$$

Shearing

- Pixel shifted horizontally or/and vertically

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & B_x \\ B_Y & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

New x value depends on x and y

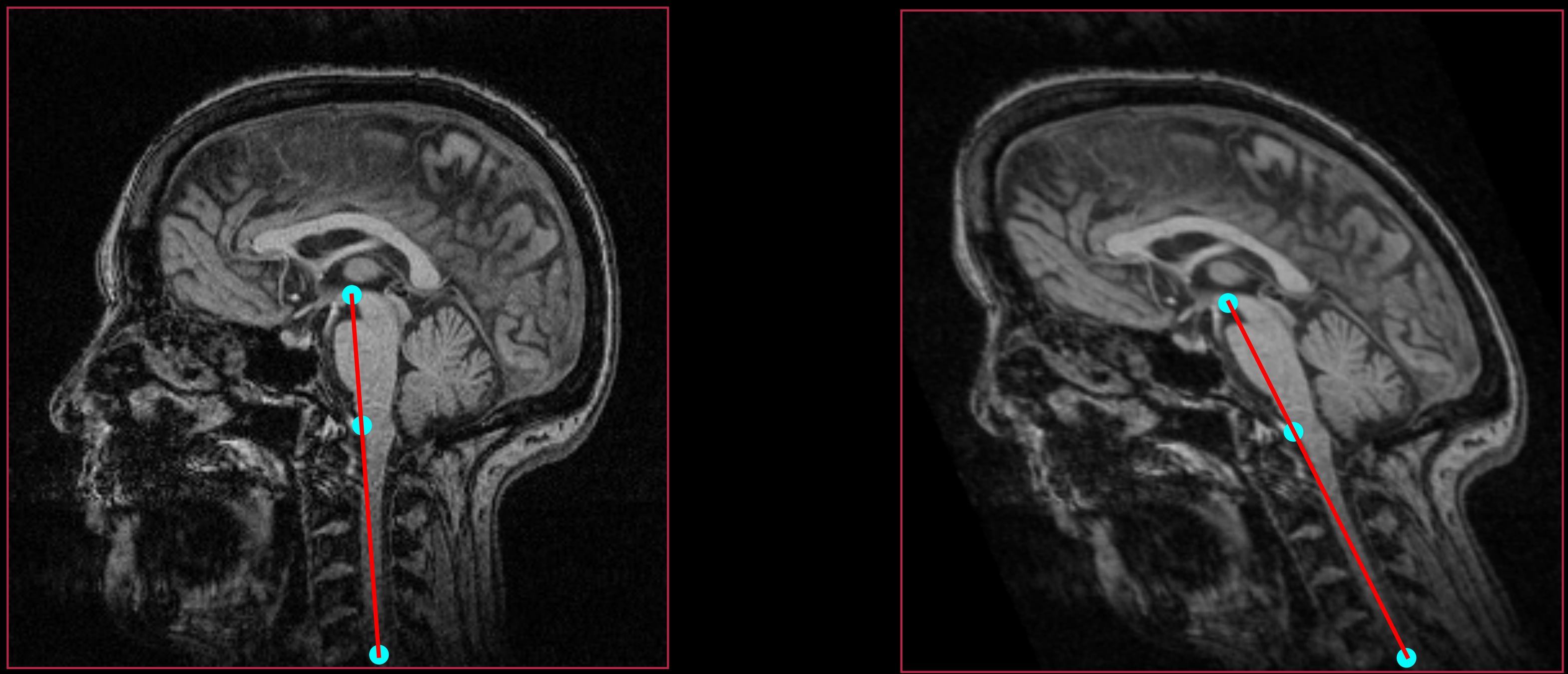


$\begin{bmatrix} x \\ y \end{bmatrix}$

$\begin{bmatrix} x' \\ y' \end{bmatrix}$

Affine transformation

- The collinearity relation between points, i.e., three points which lie on a line continue to be collinear after the transformation



Combining transformations

Scaling $S_x = S_y = 1.10$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation $\theta = 5^\circ$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

- Suppose you first want to rotate by 5 degrees and then scale by 10%

How do we combine the transformations?

Combining transformations

- Combination is done by matrix multiplication

Scaling

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Combined

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Combining transformations

- Compact notation

Scaling

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & \mathbf{0} \\ \mathbf{0} & S_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A}_S \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A}_R \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Combined

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{A}_S \cdot \mathbf{A}_R \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Remember: The order of matrix multiplications matters!

Quiz 1: Combining transforms

The point (x,y)=(5,6) is transformed. First with:

$$\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \tag{2}$$

and then with:

$$\begin{bmatrix} 0.9239 & 0.3827 \\ -0.3827 & 0.9239 \end{bmatrix} \tag{3}$$

The result is:

1. (16.12, 12.8)
2. (2.35, 20.46)
3. (11.3, 1.21)
4. (-1.2, 3.13)
5. (-30.8, 24.21)

Solution:

$$\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}\begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 2*5+0*6 \\ 0*5.+3*6 \end{bmatrix} = \begin{bmatrix} 10 \\ 18 \end{bmatrix}$$

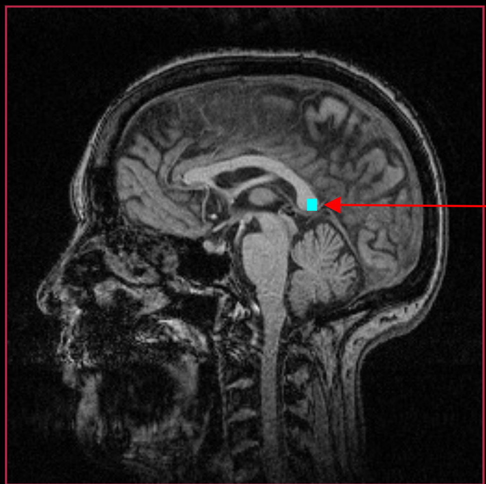
$$\begin{bmatrix} 0.9239 & 0.3827 \\ -0.3827 & 0.9239 \end{bmatrix}\begin{bmatrix} 10 \\ 18 \end{bmatrix} = \begin{bmatrix} 0.9239*10+0.3827*18) \\ -0.3827*10+0.9239*18) \end{bmatrix} = \begin{bmatrix} 16.12 \\ 12.8 \end{bmatrix}$$

What do we have now?

- We can pick a position in the input image f and find it in the output image g

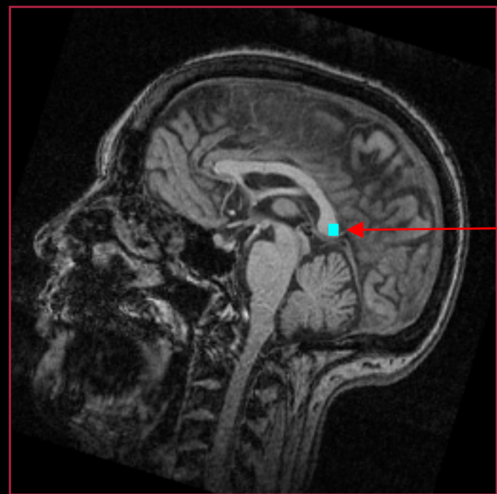
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

We can transfer one pixel – what about the whole image?



$\begin{bmatrix} x \\ y \end{bmatrix}$

f



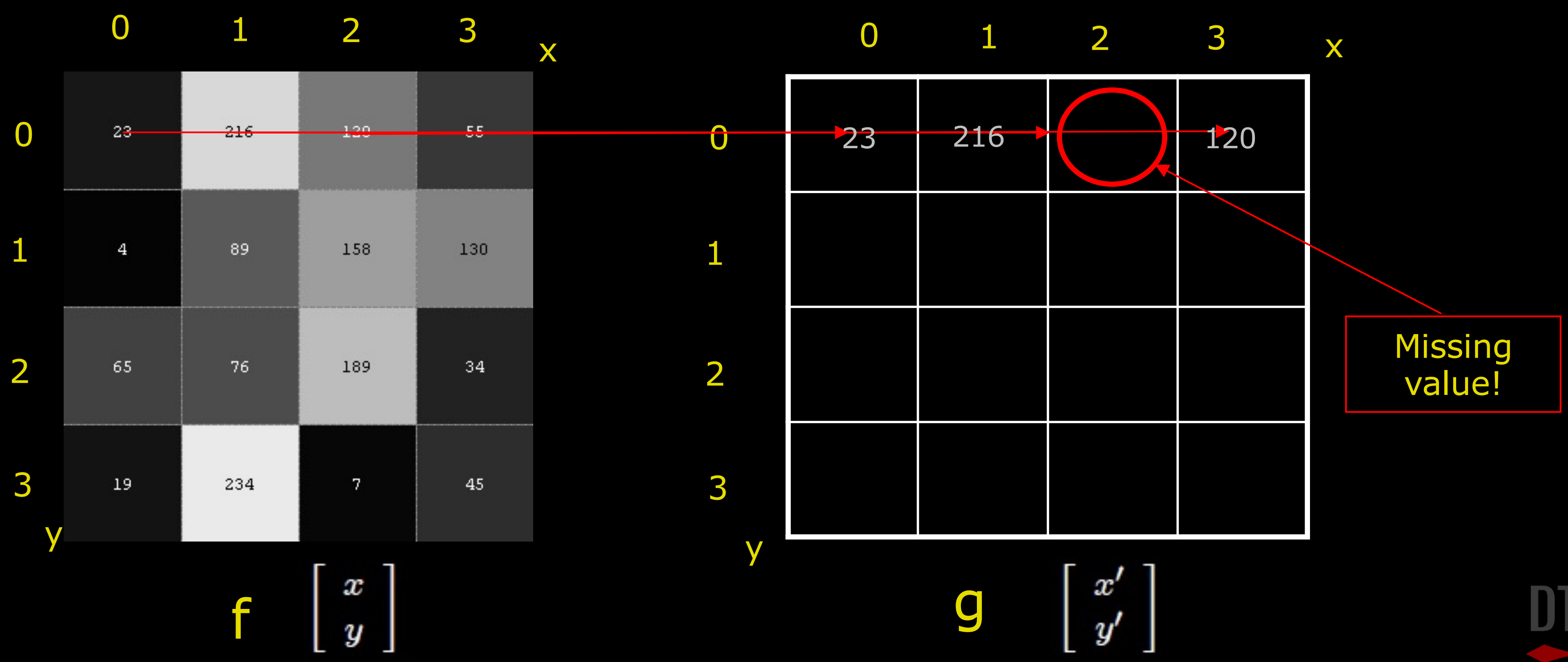
$\begin{bmatrix} x' \\ y' \end{bmatrix}$

g

Solution 1 : Input-to-output

- Run through all pixel in input image
- Find position in output image and set output pixel value

Scaling example $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$



Input-to-Output

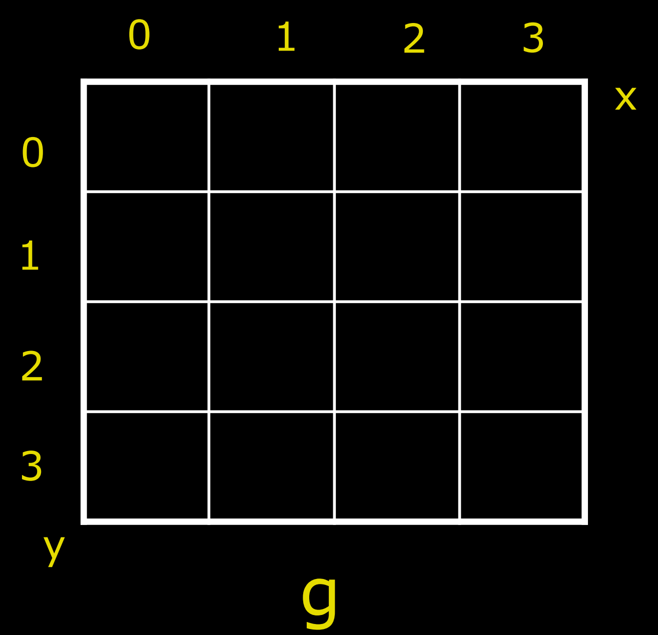
- The input to output transform is not good!
- It creates holes and other nasty looking stuff
- What do we do now?

Some observations

- We want to fill all the pixels in the output image
  - Not just the pixels that are "hit" by the pixels in the input image
- Run through all pixels in the output image?
  - Pick the relevant pixels in the input image?

We need to go "backwards"

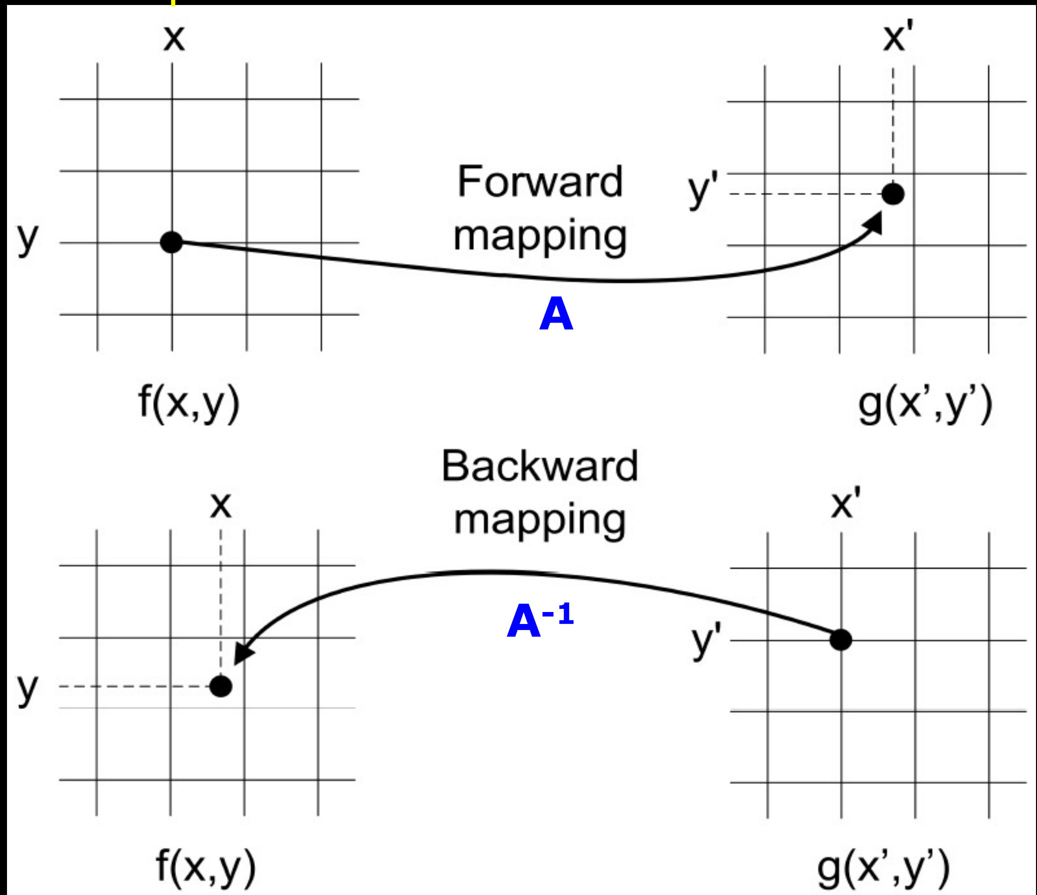
From the output to the input



Forward vs Backward mapping

- In a nut shell
  - Going backward we need to invers the transformation

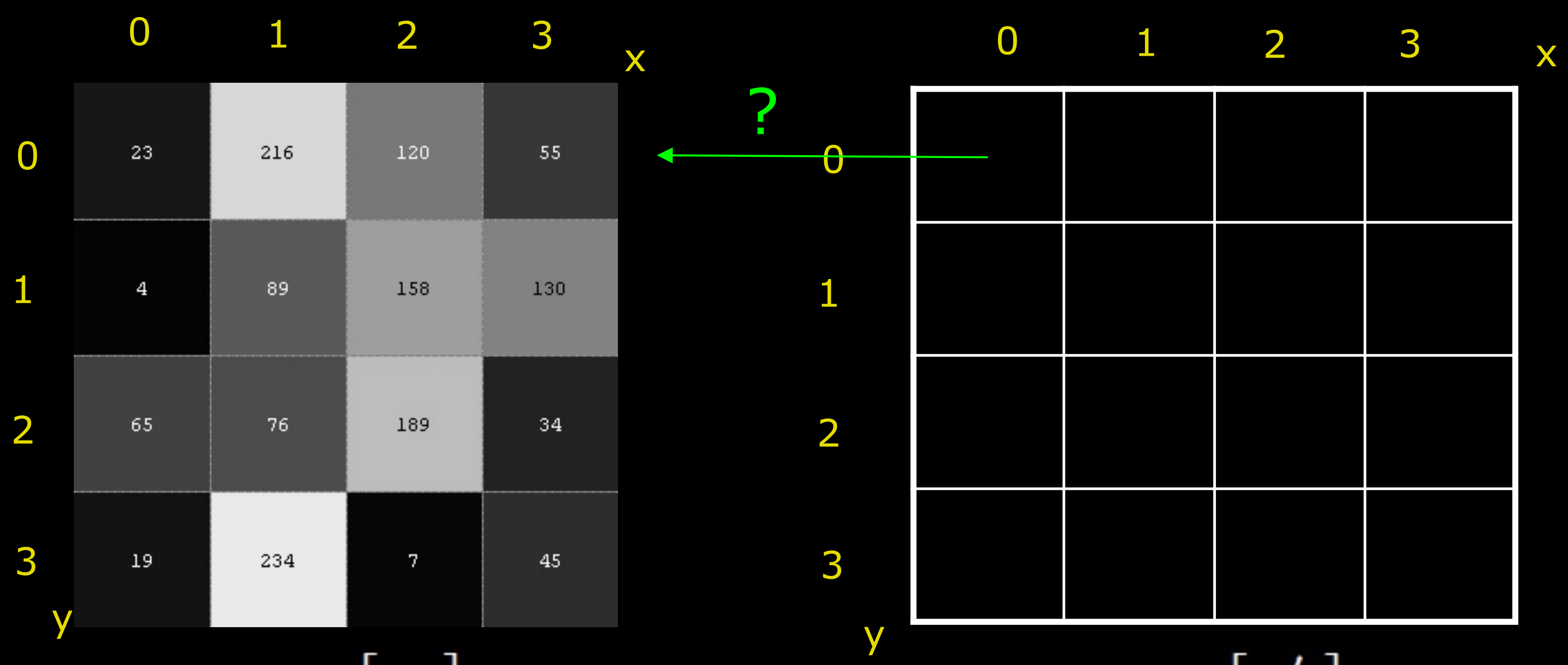
Template	Reference



Inverse transformation

- We want to go from the output to the input

Scaling example $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$ $\xrightarrow{\text{inverse}}$ $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/1.5 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$

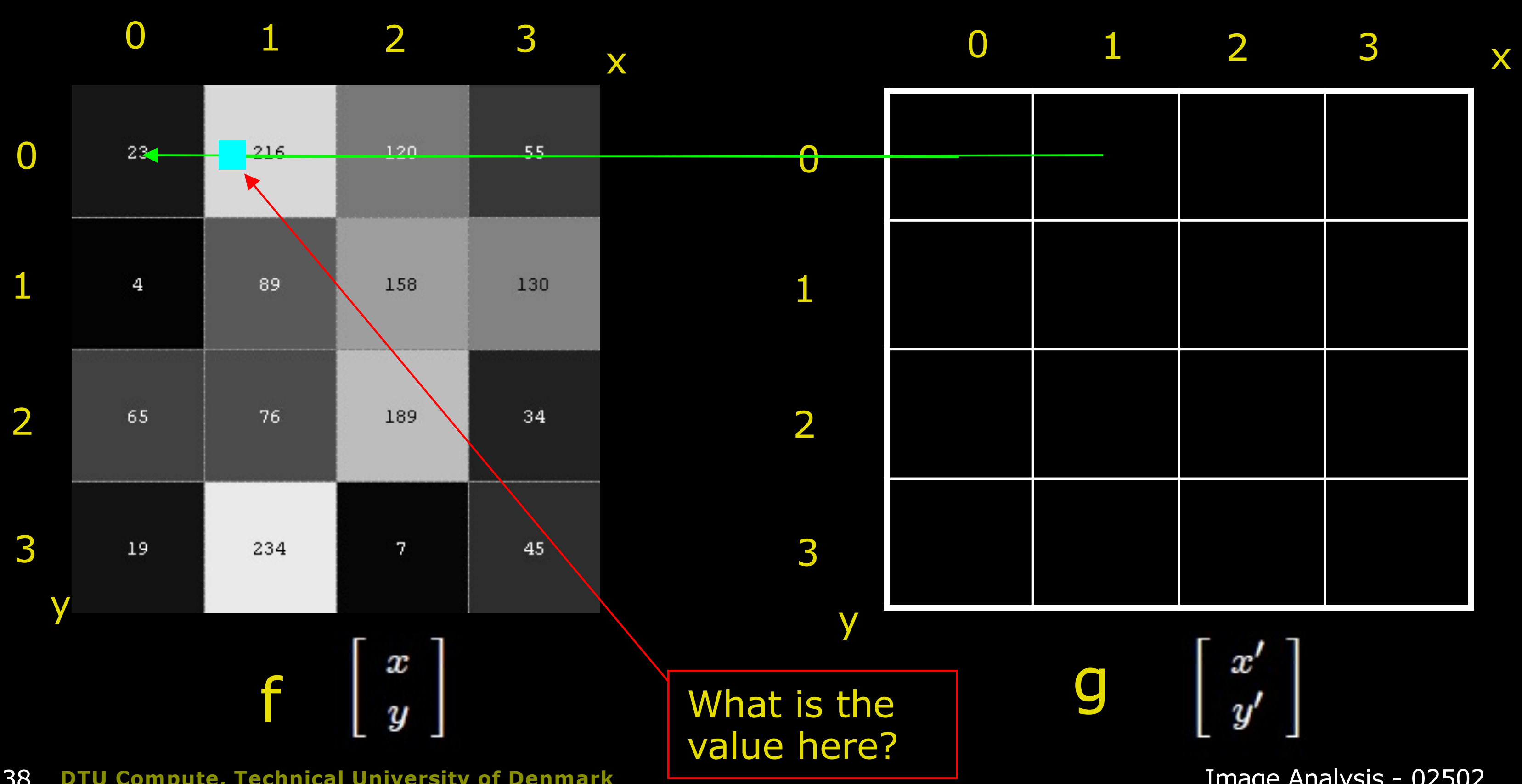


f $\begin{bmatrix} x \\ y \end{bmatrix}$ g $\begin{bmatrix} x' \\ y' \end{bmatrix}$

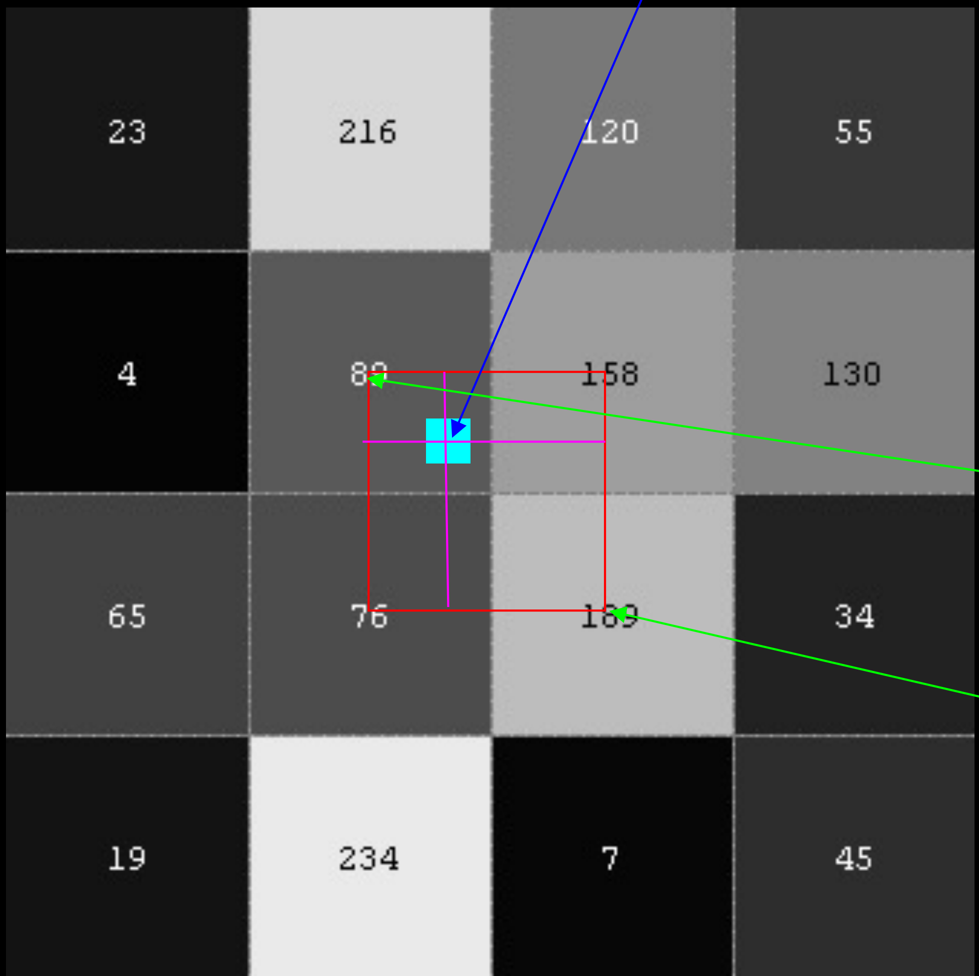
Output-to-input transformation
Backward mapping

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/1.5 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$$

- Run through all pixel in output image
- Find position in input image and *get the value*



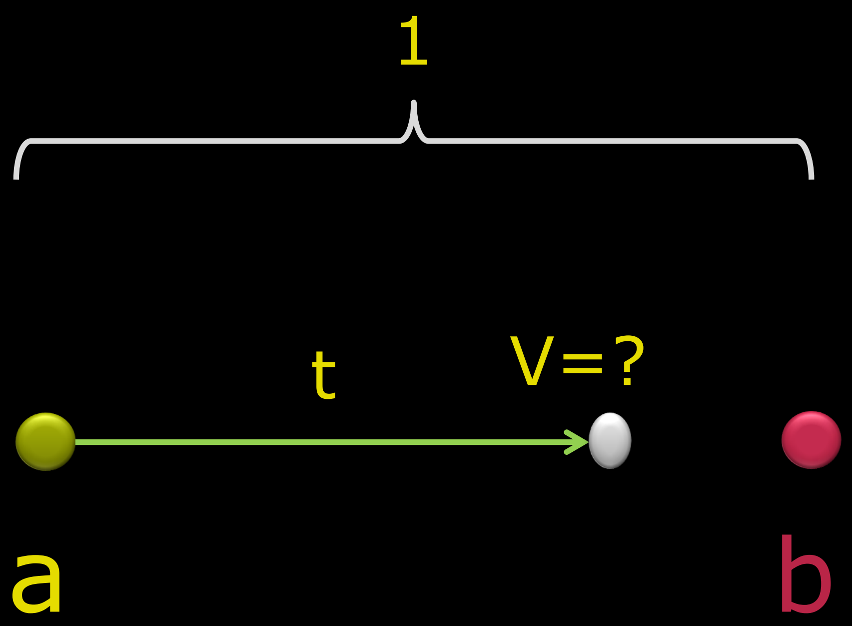
Bilinear Interpolation



- The value is calculated from 4 neighbours
- The value is based on the distance to the neighbours

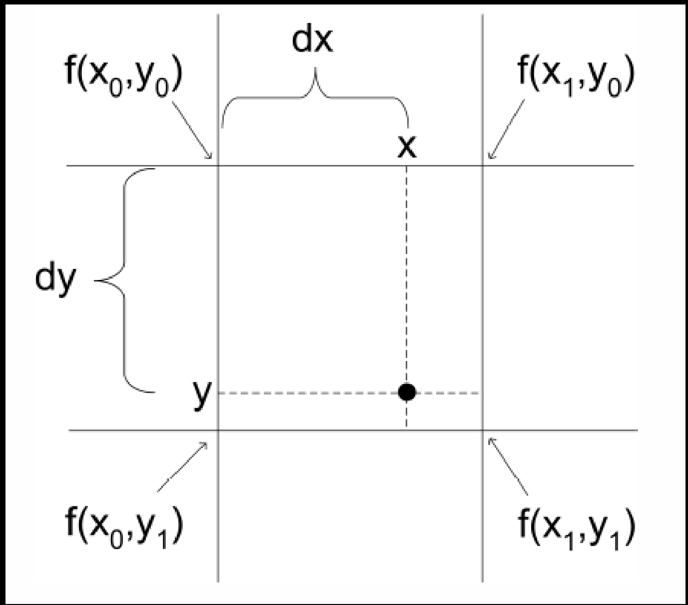
Linear Interpolation (1D)

$$v = tb + (1 - t)a$$



Bilinear interpolation (2D)

$$
\begin{aligned}
g(x', y') \quad = \quad & f(x_0, y_0) \cdot (1 - dx)(1 - dy) + \\
& f(x_1, y_0) \cdot (dx)(1 - dy) + \\
& f(x_0, y_1) \cdot (1 - dx)(dy) + \\
& f(x_1, y_1) \cdot (dx \cdot dy) \quad ,
\end{aligned}
$$



Quiz 2: Bilinear interpolation

Bilinear interpolation is used to create a line profile from an image. In a given point (x,y) = (173.1, 57.8), the four nearest pixels are:

x	y	værdi
173	57	110
174	57	140
173	58	156
174	58	101

What is the interpolated value in the point:

1. 131
2. 143
3. 128
4. 151
5. 139

Solution:

Distance between grid points is 1

hence: dx=0.1 and dy=0.8

Do the interpolation (see previous slide)

$g(173.1, 57.8)=$

$110*(1-0.1)*(1-0.8)+$

$140*(0.1)*(1-0.8)+$

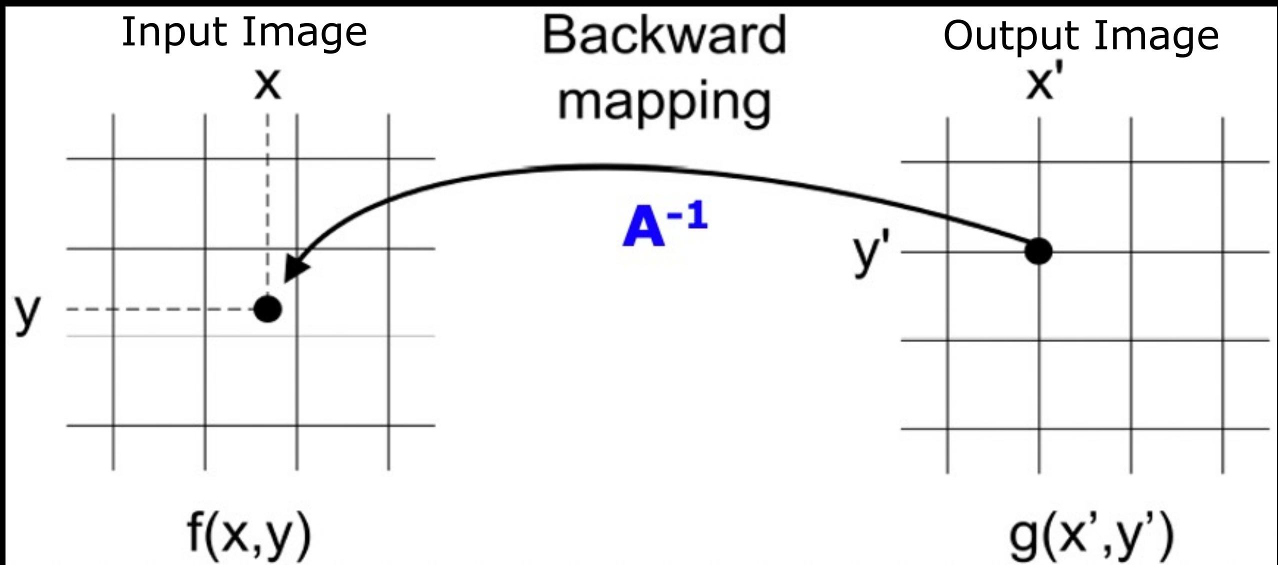
$156*(1-0.1)*(0.8)+$

$101*(0.1)*(0.8)$

$=143$

Output-to-input transformation
Backward mapping

- Run through all the pixel in the output image
- Use the inverse transformation to find the position in the input image
- Use bilinear interpolation to calculate the value
- Put the value in the output image



Inverse transformation

- We can calculate the inverse transformation for the scaling
- What about the others?

Scaling

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Inverse

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/1.5 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$$

General inverse transformation

Affine transformation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Inverse transformation

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A}^{-1} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Where

$$\mathbf{A}^{-1} \cdot \mathbf{A} = \mathbf{I}$$

- The transformation is expressed as a transformation matrix A
- The *matrix inverse* of A gives the inverse transformation

Quiz 3: Transformation

The point (x,y) = (45, 23) is transformed using:

$$\begin{bmatrix} 0.5 & 2 \\ 2 & 0.8 \end{bmatrix} \quad (1)$$

And the result is translated with (-15, 20). The result is:

1. (53.5, 128.4)
2. (3.4, -10.3)
3. (45.3, 80.2)
4. (150.8, 32.4)
5. (-20.5, 22.6)

Solution:

$$(x', y') = \begin{bmatrix} -15 \\ 20 \end{bmatrix} + \begin{bmatrix} 0.5 & 2 \\ 2 & 0.8 \end{bmatrix} \begin{bmatrix} 45 \\ 23 \end{bmatrix}$$

$$= \begin{bmatrix} -15 \\ 20 \end{bmatrix} + \begin{bmatrix} 68.5 \\ 108.4 \end{bmatrix}$$

$$= \begin{bmatrix} 53.5 \\ 128.4 \end{bmatrix}$$

Image Registration

Image Registration

- The act of adjusting something to match a standard
- Align images

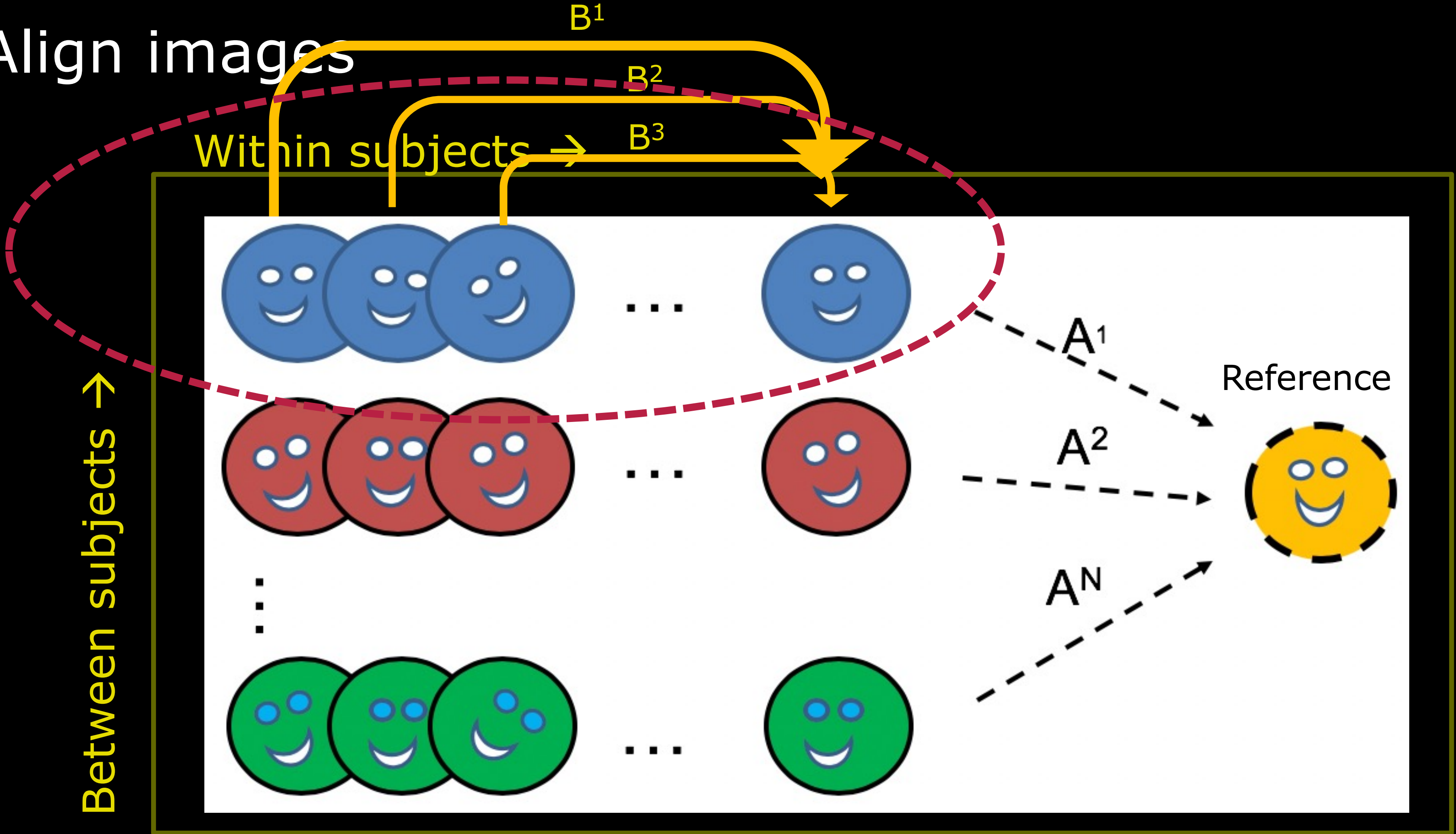


Image registration

- Monitoring of change in the individual
- Fusion of information from different sources in a meaningful way
- Comparison of one subject with others
- Comparison of groups with others
- Comparing with an atlas

Data fusion

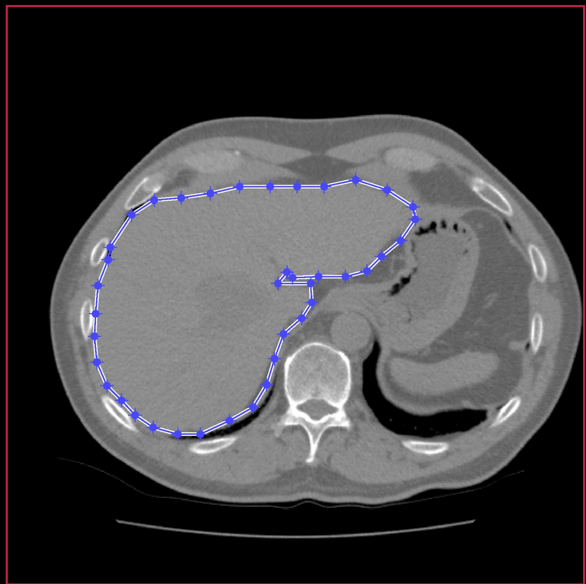
Same patient – two scans

MR

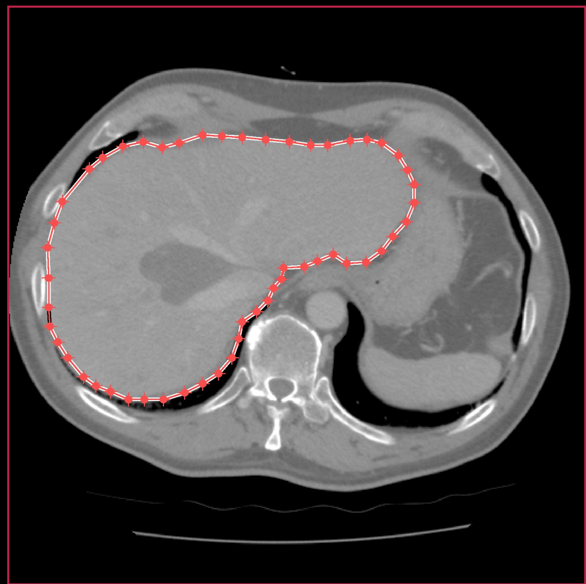
CT

Change detection

- Patient image before and after operation
- What has changed?
- Images need to be aligned before comparison

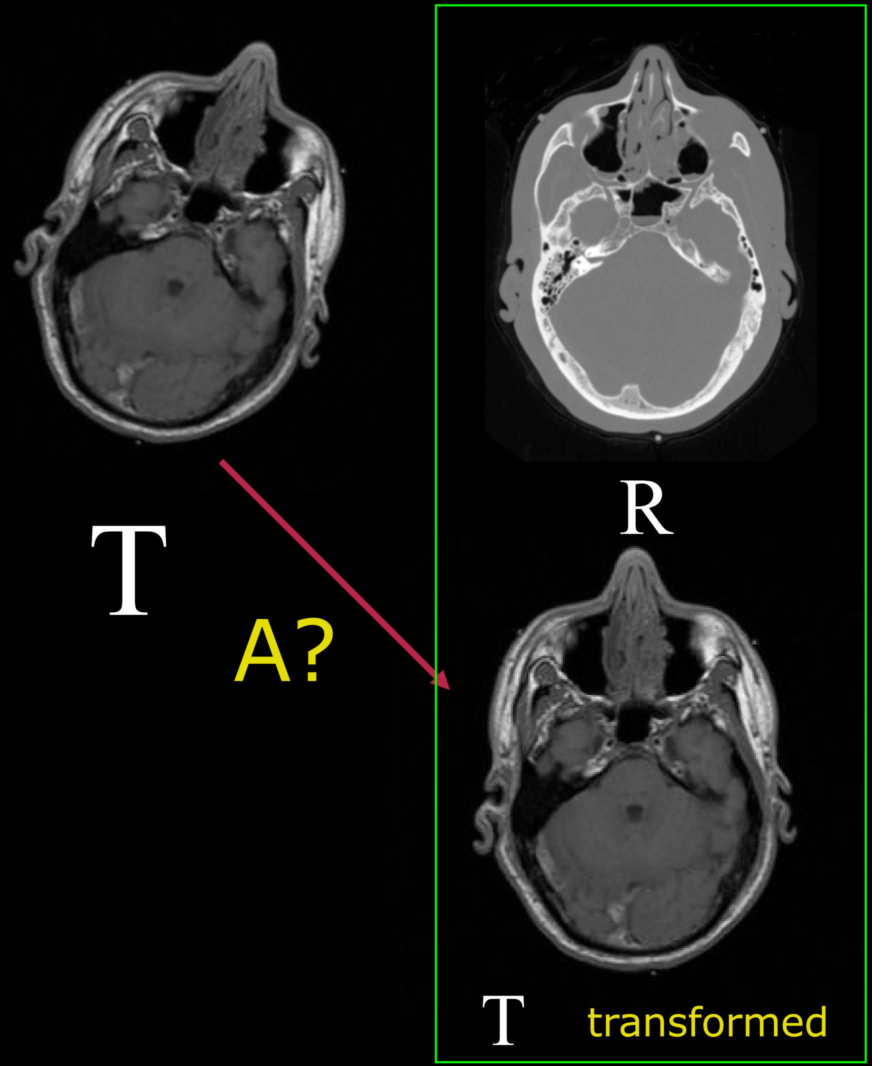


Before operation



After operation

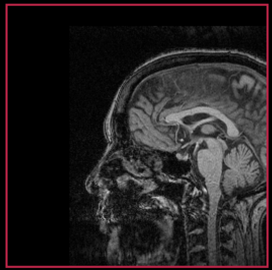
Reference and template image

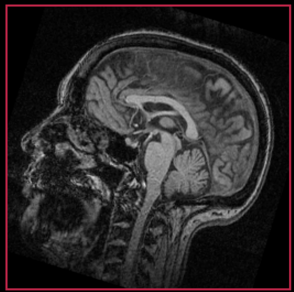


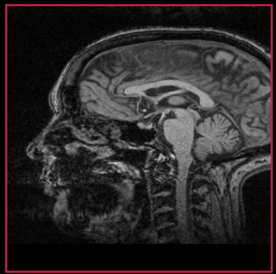
- The reference image R
- Template image T
- Transform the template so it fits the reference
- Combine geometrical transformations
- Find the transformation matrix, A for the best match

The transformations

- Translation
- Rotation
- Scaling





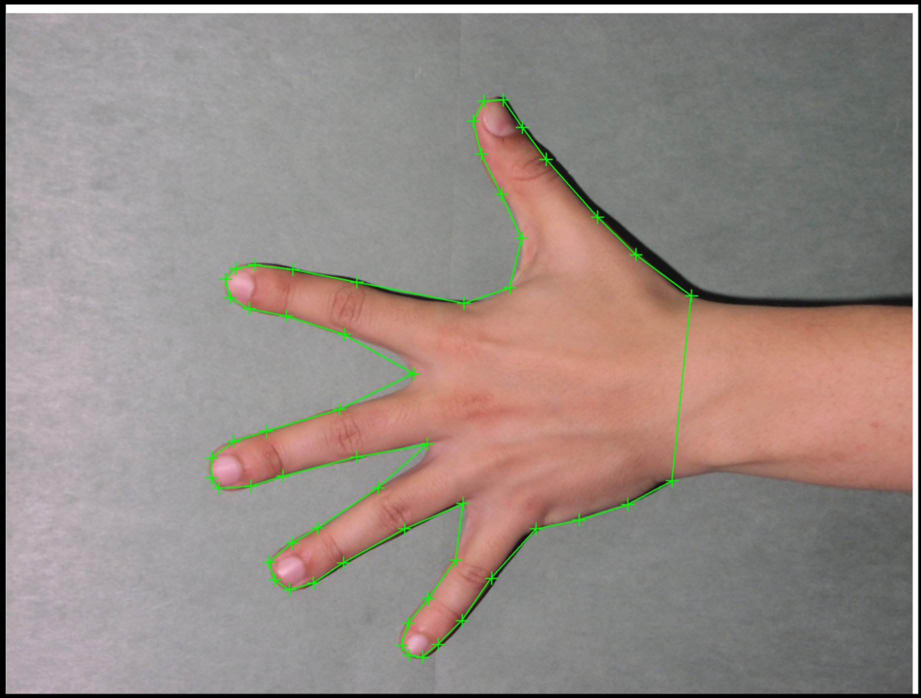


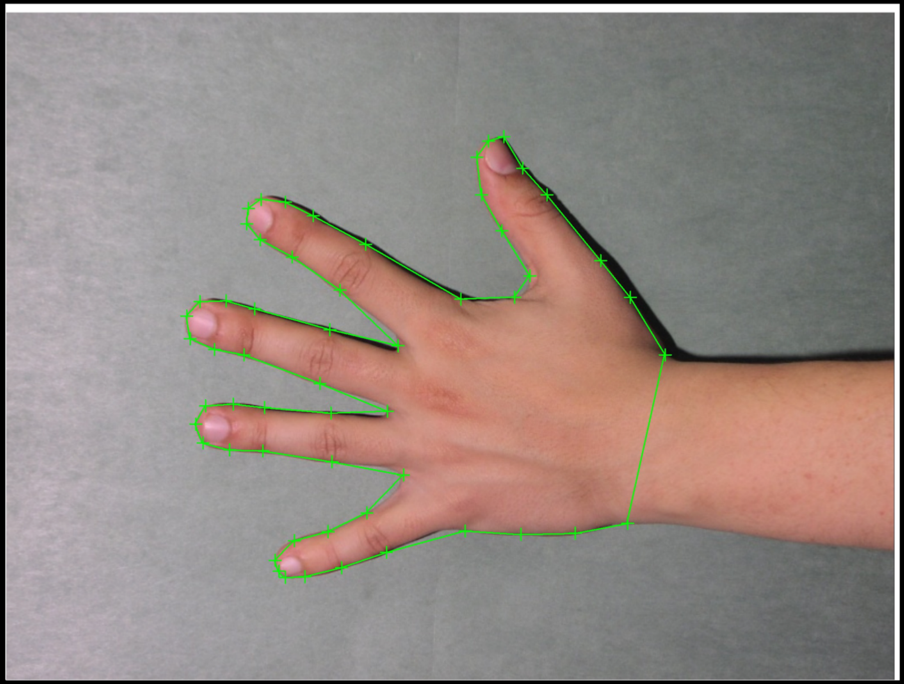
Similarity measures

- The aim is to transform the template so it *looks like* the reference
- Looks like = Similarity measure
- Image similarity
  - Subtract the two images and see "what is left"
- Landmark similarity
  - Landmarks from the two images should be "close together"

Landmark Based Registration

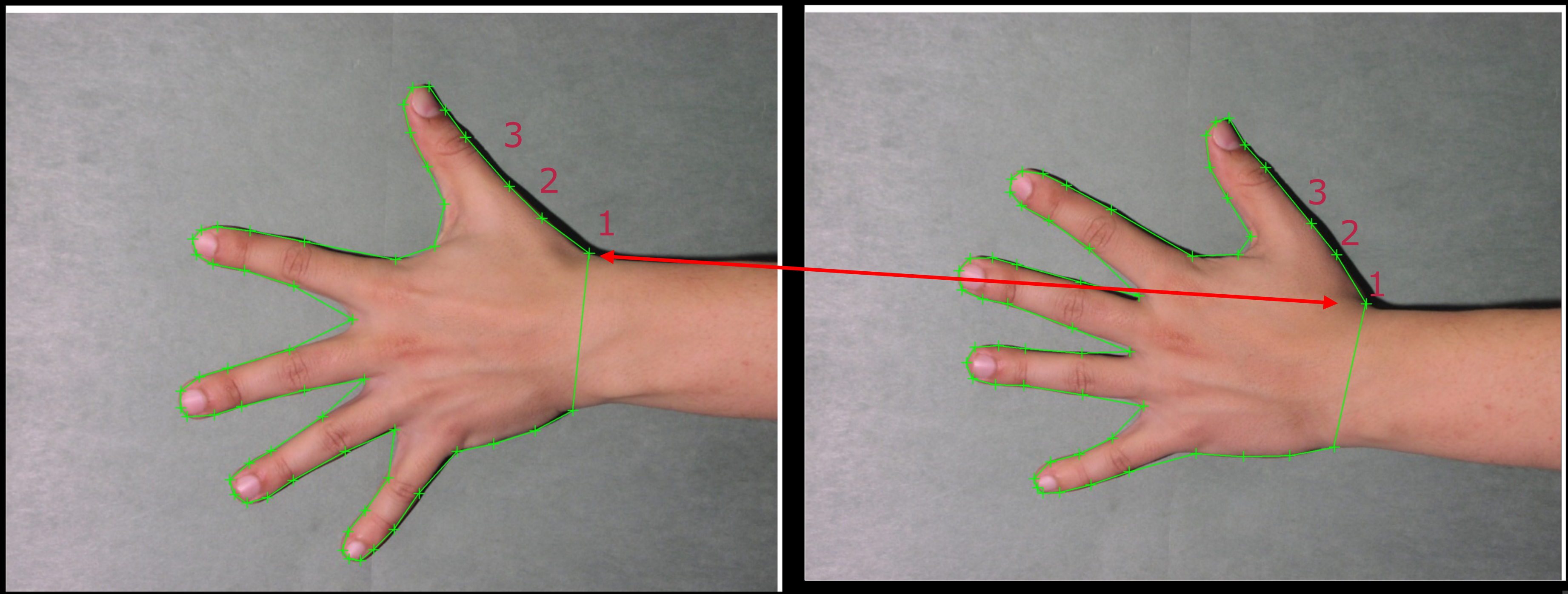
- Landmarks placed on both reference and template image
- The landmark should have *correspondence*



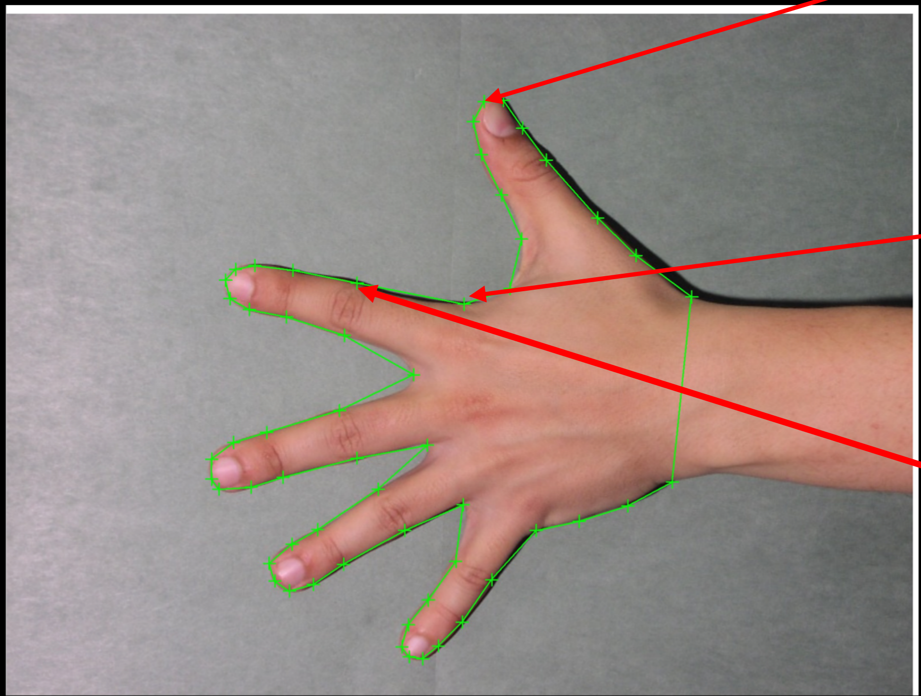


Point correspondence

- Landmarks are numbered
- Each landmark should be placed the same place on both images



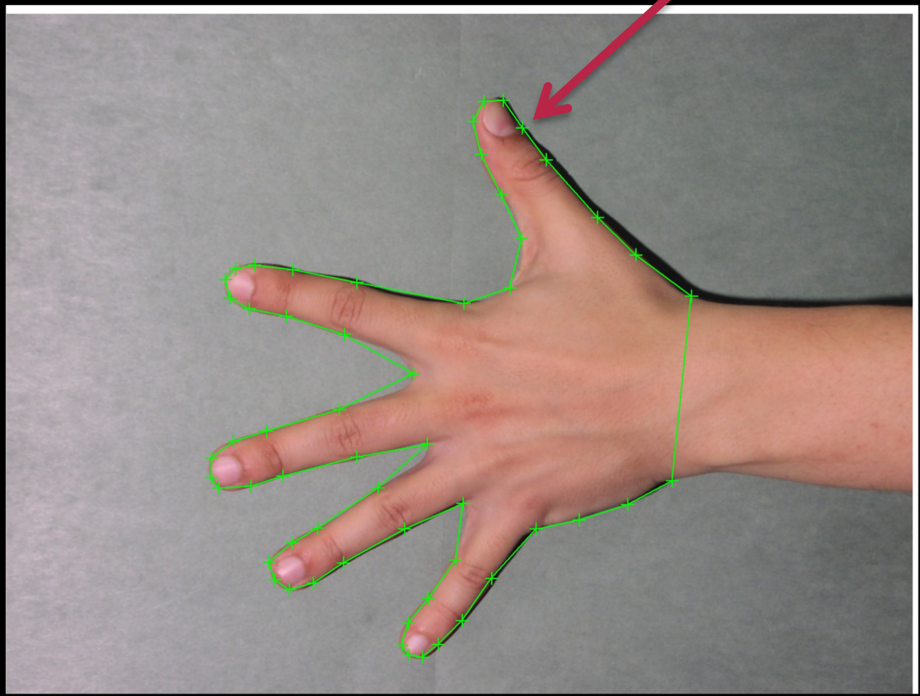
Landmark types



- Anatomical landmark
  - a mark assigned by an expert that corresponds between objects in a biologically meaningful way
- Mathematical landmark
  - a mark that is located on a curve according to some mathematical or geometrical property
- Pseudo landmark
  - a mark that is constructed on a curve based on anatomical or mathematical landmarks

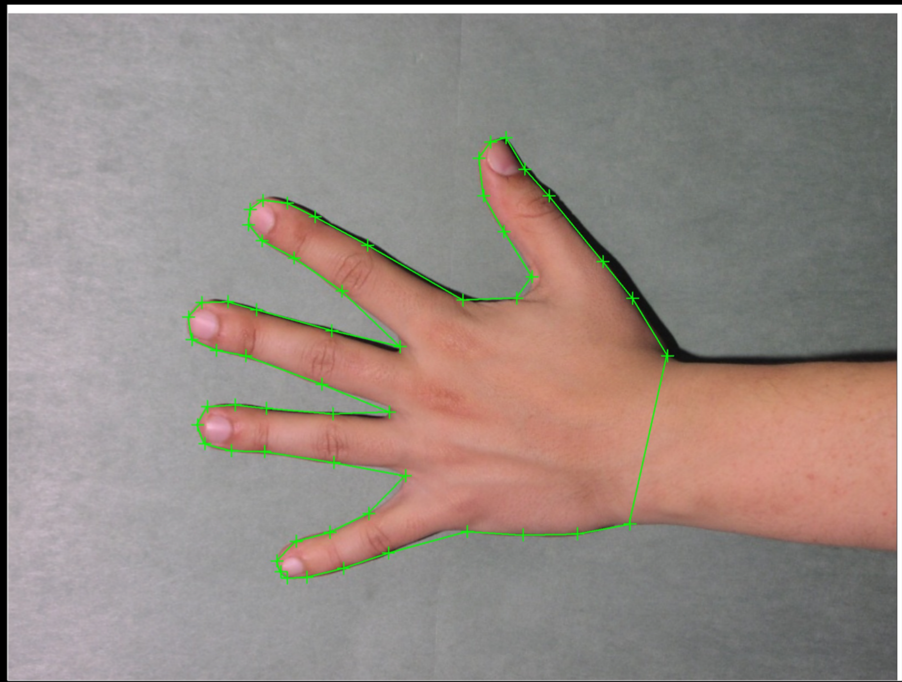
Landmarks

$$a_5 = (412, 55)$$



a_i

Reference image R



b_i

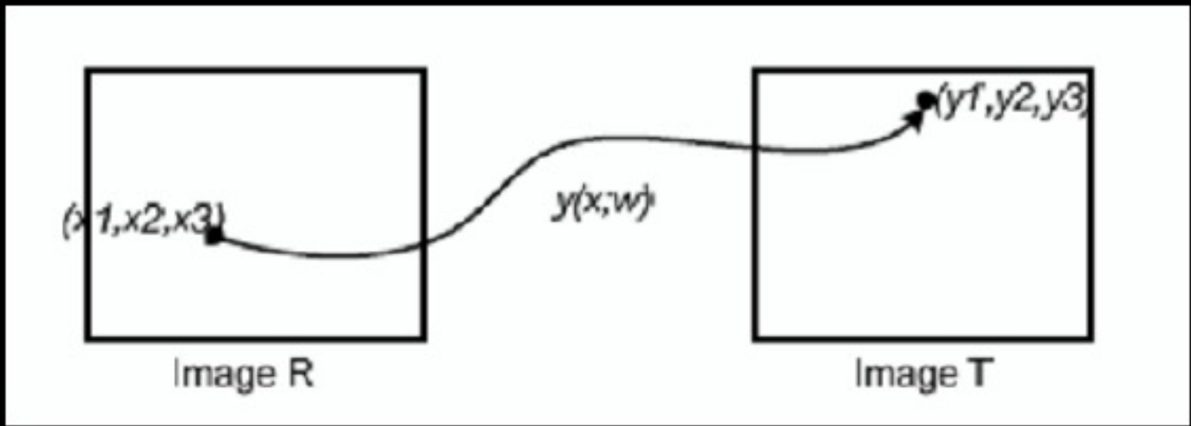
Template image T

The aim of registration

- We have selected Landmark points
- Find a transformation that maps the coordinates of the reference to the coordinates of the template
  - Why not the template to the reference?

    Sampling of template image:

    Backward mapping -> inverse transform



The Transformation

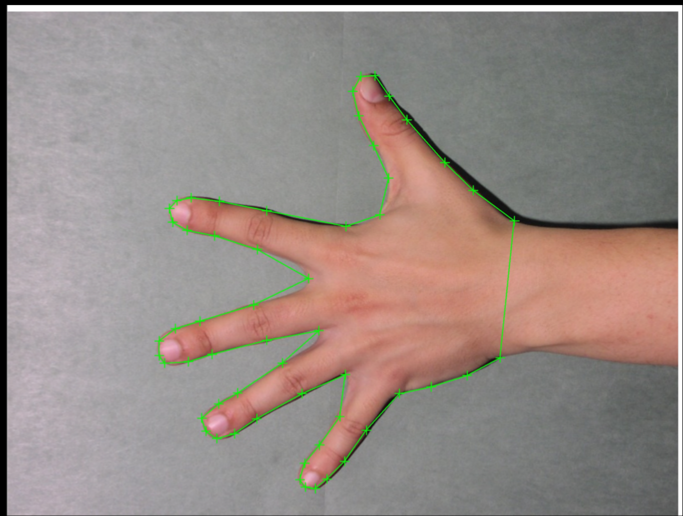
$$p' = T(p)$$

- Transforms point *p*
- Into point *p′*
- T is for example geometrical transformations eg. a
  - Translation
  - Rotation
  - Rigid body transform
  - Similarity transform

The Transformation

$$a_i' = T(a_i)$$

- Transforms points from the reference



a_i

The parameters

$$w \in R^p$$

parameters

- The parameters is a vector with p elements
  - The type of transformation determines the number of parameters
  - Translation p = 2
  - Rotation p = 1
  - Scaling p = 1

Quiz 4: Rigid body transform
How many parameters?

$$w \in R^p$$

A) 1
B) 2
C) 3
D) 4
E) 5

Solution:

We have:

- Translation in x and y axis p= 2
- Rotation P= 1

In total 3 parameters for rigid transformation

$$w = (\Delta x, \Delta y, \theta)$$

Objective function

$$F = \sum_{i=1}^{N} D(T(a_i), b_i)^2$$

Points from the template image

Transformed points from the reference image

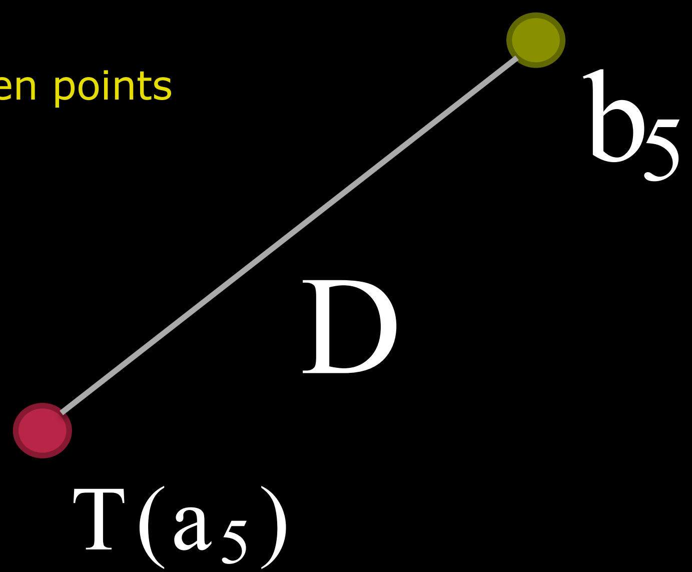
- The *objective function* measures how well two point sets match
- It uses a *cost function that describe how to evaluate the match*
- Here the cost function is a sum-of-squares distance function
- Point sets could be landmarks

Objective function

$$F = \sum_{i=1}^{N} D(T(a_i), b_i)^2$$

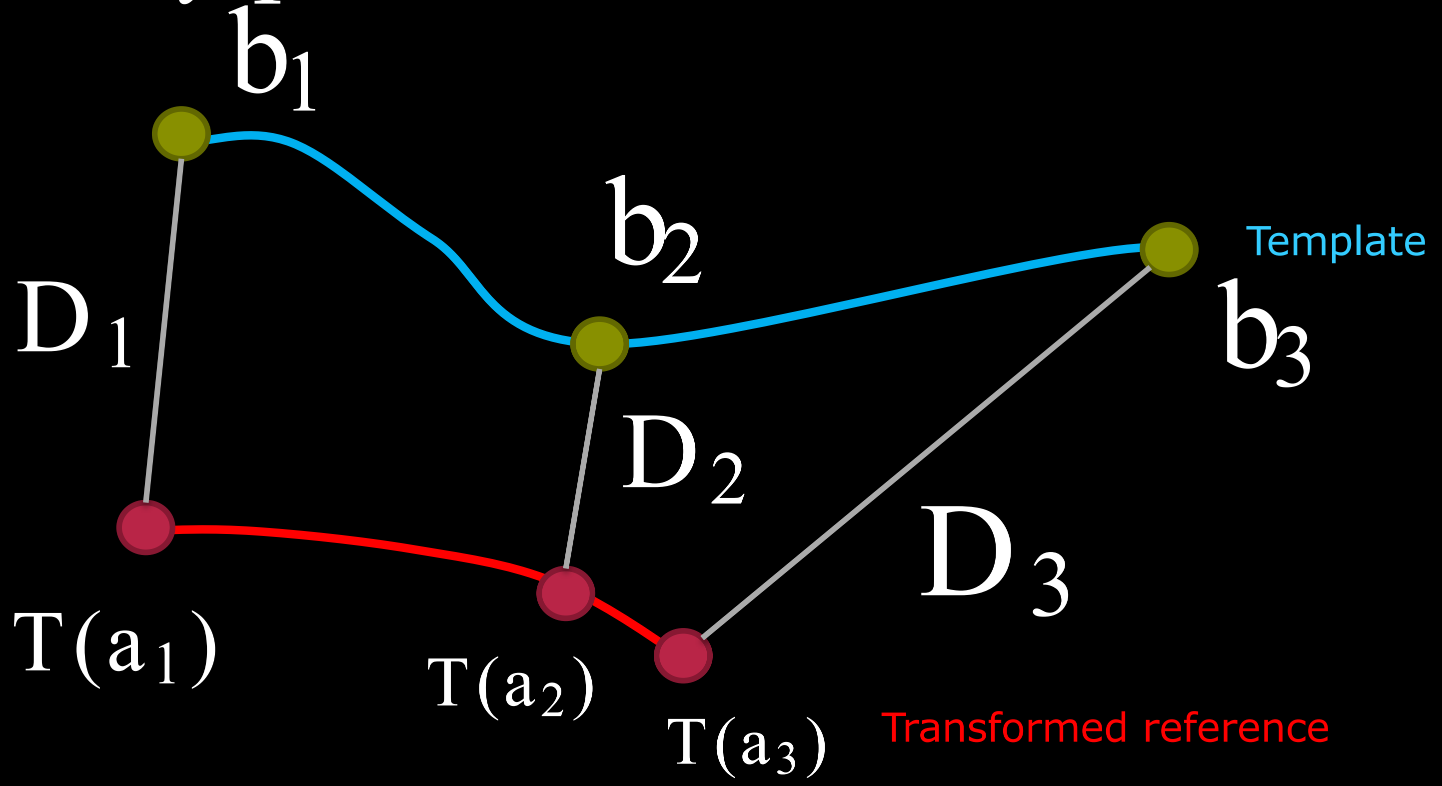
Distance between points

- The *objective function* measures how well two point sets match



Objective function

$$F = \sum_{i=1}^{3} D(T(a_i), b_i)^2 = D_1^2 + D_2^2 + D_3^2$$

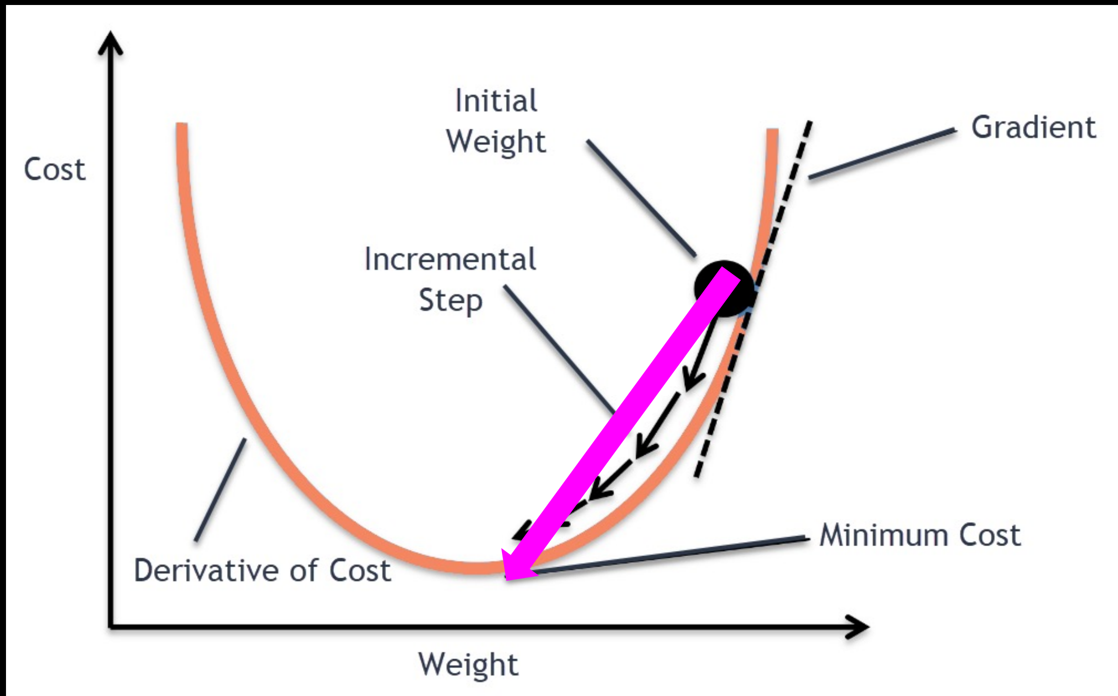


Minimization / Optimization

$$F = \sum_{i=1}^{N} D(T(a_i), b_i)^2$$

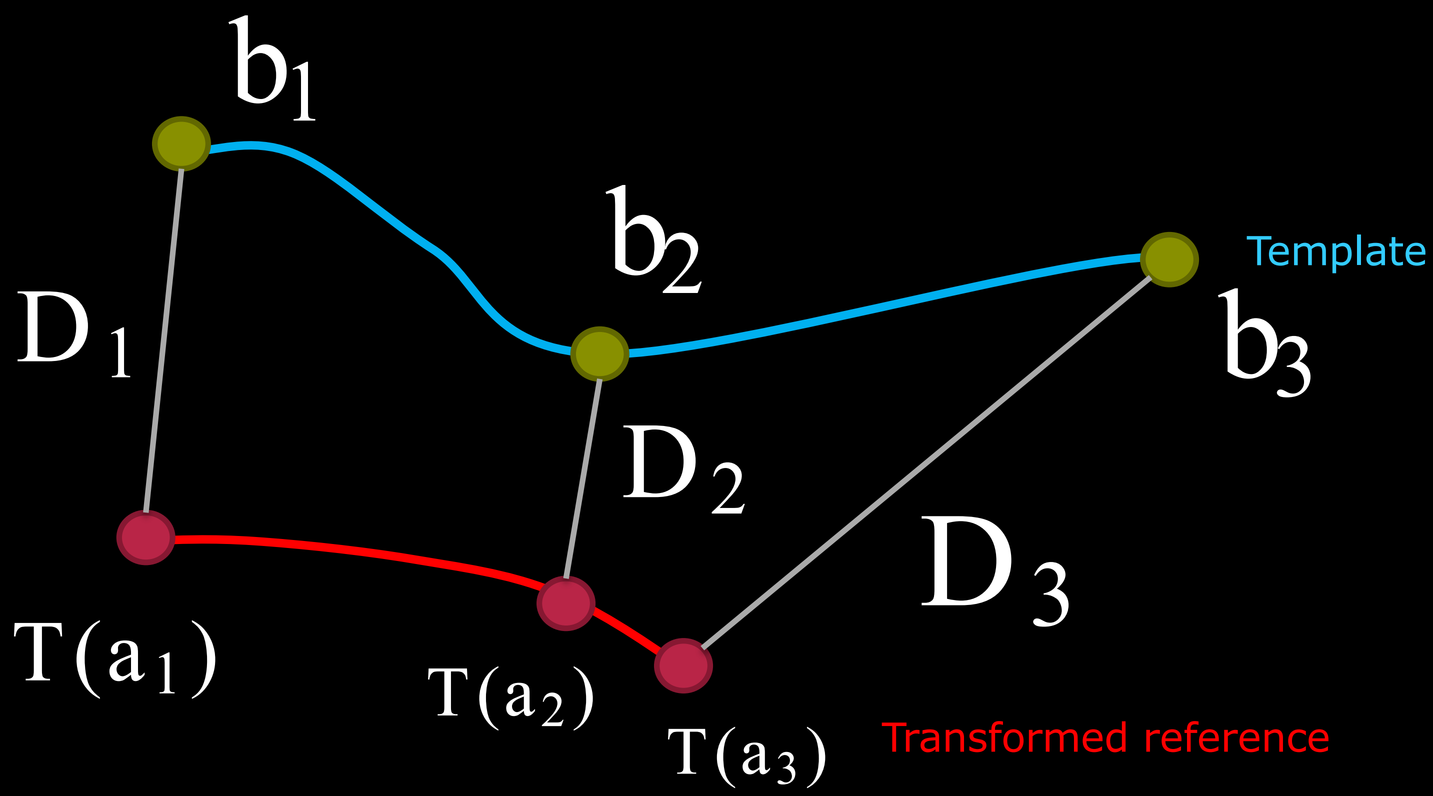
$$\boxed{\hat{w} = \arg\min_{w} F}$$

- Find the set of parameters that minimizes the objective function
- Optimisation strategy: Analytic (exact solution) vs Numerical?



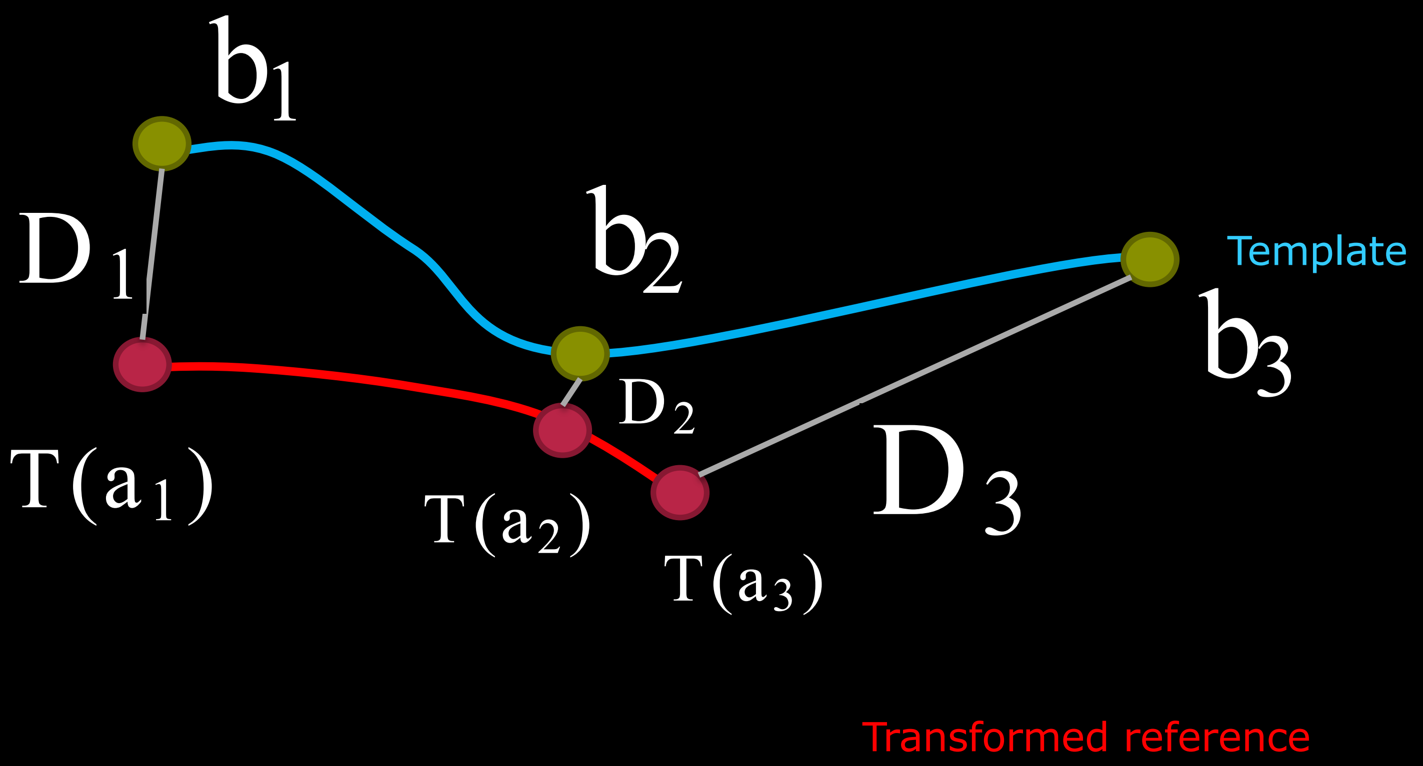
Minimization – pure translation

$$F = D_1^2 + D_2^2 + D_3^2$$



Minimization – pure translation

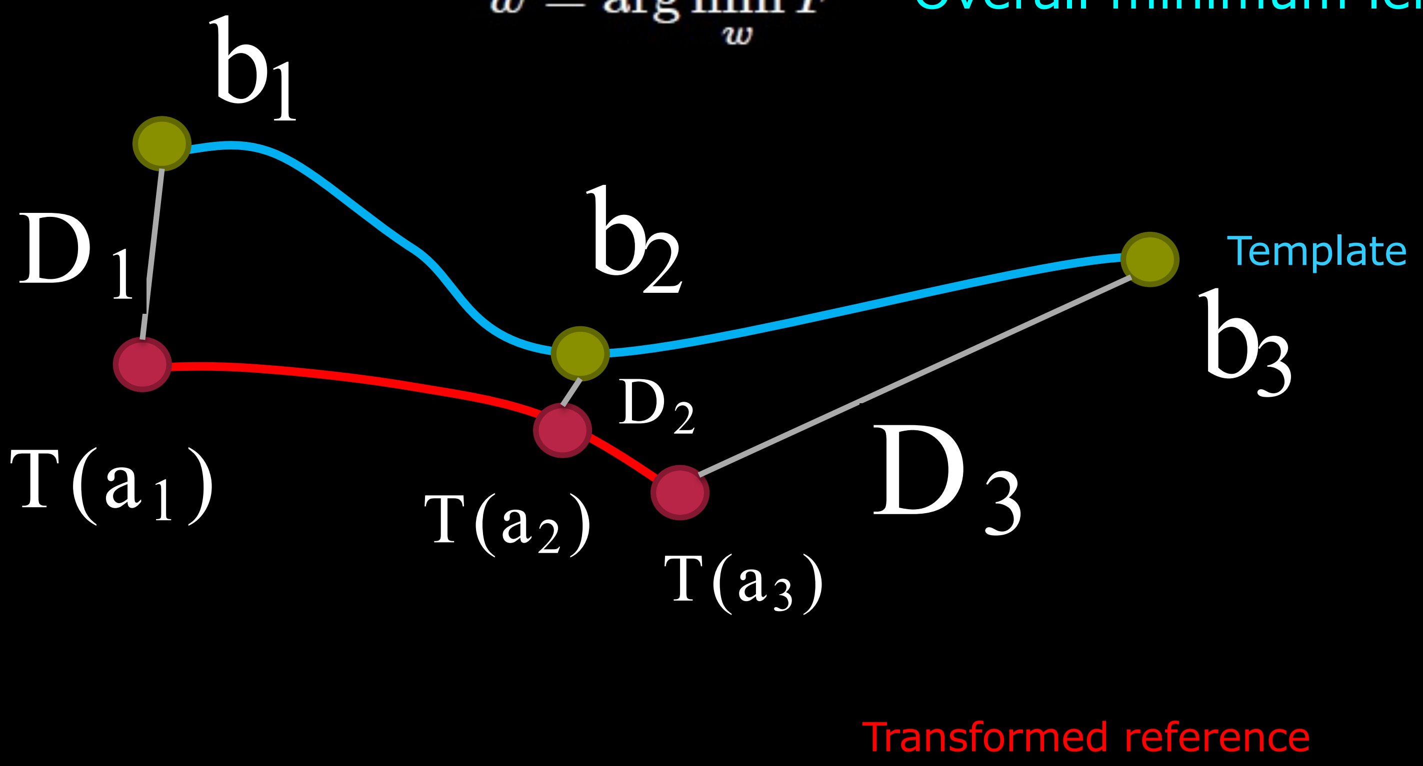
$$F = D_1^2 + D_2^2 + D_3^2$$ Decreased!



Minimization – pure translation

$$F = D_1^2 + D_2^2 + D_3^2$$ Decreased!

$$\hat{w} = \arg\min_{w} F$$ Overall minimum lengths!



DTU Compute

Quiz 5: Objective function

A) 600
B) 50
C) 100
D) 900
E) 300

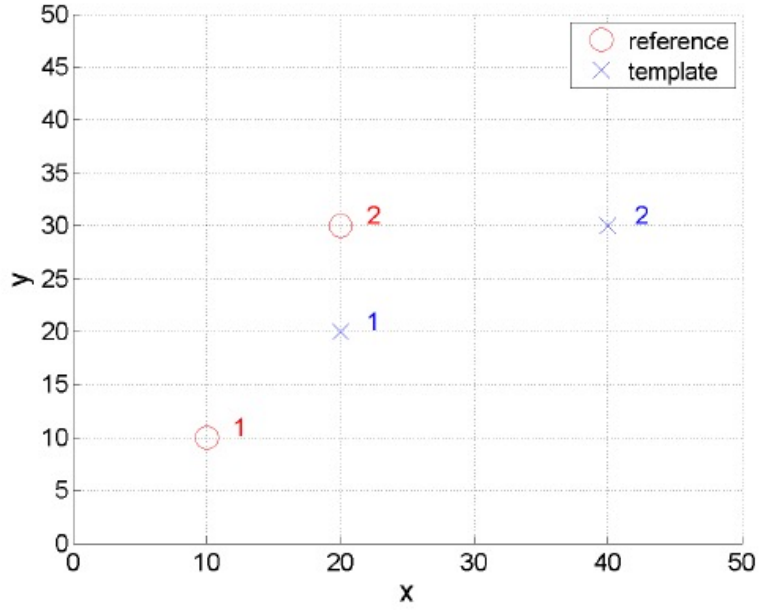
Solution:

$$D1^2 = \left\| \begin{bmatrix}10\\10\end{bmatrix} - \begin{bmatrix}20\\20\end{bmatrix} \right\|^2 = \left\| \begin{matrix}10\\10\end{matrix} \right\|^2 = 200$$

$$D2^2 = \left\| \begin{bmatrix}20\\30\end{bmatrix} - \begin{bmatrix}40\\30\end{bmatrix} \right\|^2 = \left\| \begin{matrix}20\\0\end{matrix} \right\|^2 = 400$$

An expert has placed two sets of landmark in the image below. We want to find the optimal translation. First we compute the objective function *F* and it is:

1. 600
2. 50
3. 100
4. 900
5. 300
6. Ved ikke

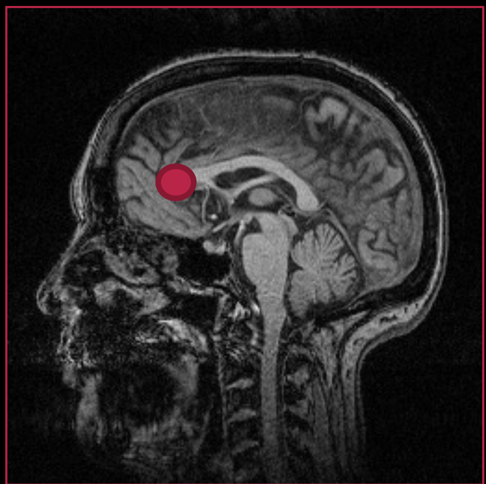


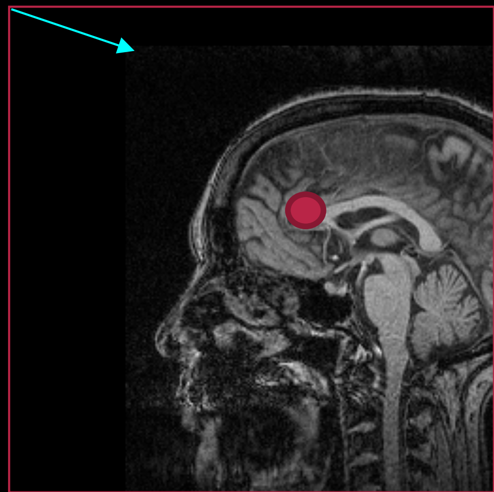
Translation

- Simple shift of coordinates

$$T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = (x, y) + t$$

parameters $w = (\Delta x, \Delta y)$





Objective function for translation

Objective function

$$F = \sum_{i=1}^{N} D(T(a_i), b_i)^2$$

Translation

$$a_i' = a_i + t$$

Objective function for translation

$$F = \sum_{i=1}^{N} \|(a_i + t) - b_i\|^2$$

Optimal function value

$$F = \sum_{i=1}^{N} D(T(a_i), b_i)^2$$

To find: $\hat{w} = \arg\min_{w} F$

We simply differentiate w.r.t. *w*:

$$\boxed{\frac{\partial F}{\partial w} = 0}$$

Optimal translation

Objective function

$$F = \sum_{i=1}^{N} \|(a_i + t) - b_i\|^2$$

Parameters

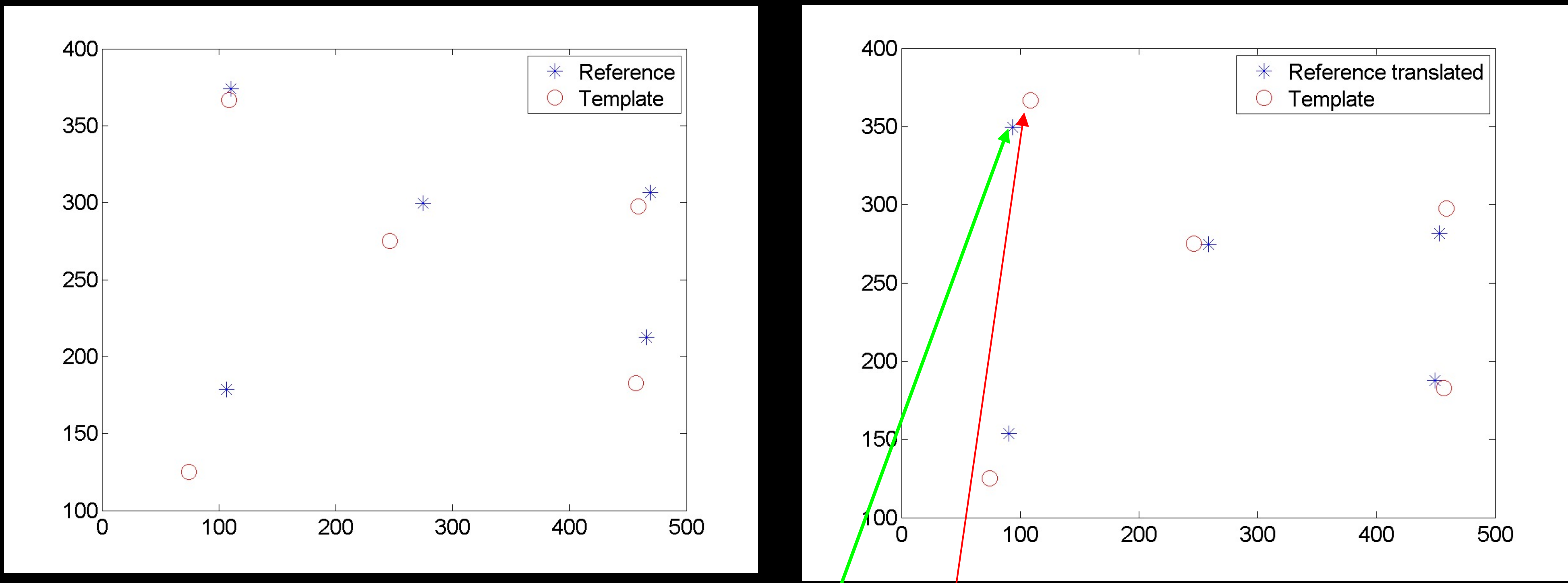
$$w = (\Delta x, \Delta y) = t$$

Optimal translation

$$\hat{t} = \bar{b} - \bar{a} \qquad \bar{a} = \frac{1}{N}\sum_{i=1}^{N} a_i$$

Average point = centre of mass

Optimal translation



Original landmarks

Reference points translated

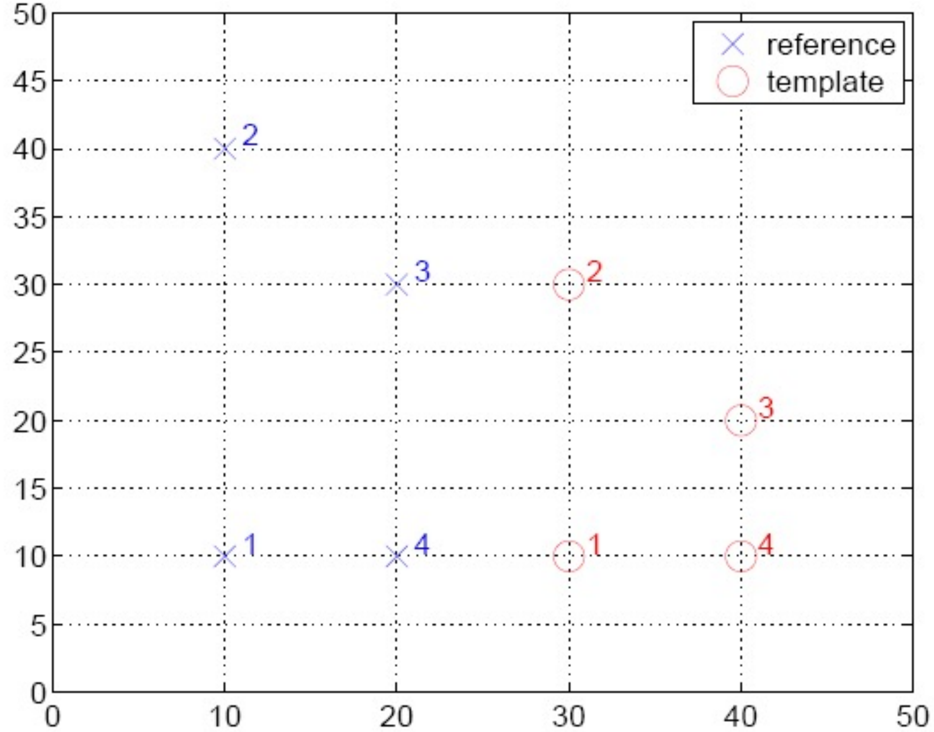
$$F = \sum_{i=1}^{N} \|(a_i + t) - b_i\|^2$$

Quiz 6: Optimal translation

A) (-10, 10)
B) (20,5)
C) (20,-5)
D) (15,-5)

An expert has placed four landmarks in two images. The optimal translation that brings the landmarks from the reference image over in the landmarks from the template image. What is this translations?

2. (-10, -10)
3. (20, 5)
4. (20, -5)
5. (15, 5)
6. Ved ikke



Solution:

$$\hat{t} = \bar{b} - \bar{a}$$

$$\bar{a} = \frac{1}{4}\left(\begin{bmatrix}10\\10\end{bmatrix} + \begin{bmatrix}10\\40\end{bmatrix} + \begin{bmatrix}20\\30\end{bmatrix} + \begin{bmatrix}20\\10\end{bmatrix}\right) = \begin{bmatrix}15\\22{,}5\end{bmatrix}$$

$$\bar{b} = \frac{1}{4}\left(\begin{bmatrix}30\\10\end{bmatrix} + \begin{bmatrix}30\\40\end{bmatrix} + \begin{bmatrix}40\\20\end{bmatrix} + \begin{bmatrix}40\\10\end{bmatrix}\right) = \begin{bmatrix}35\\17{,}5\end{bmatrix}$$

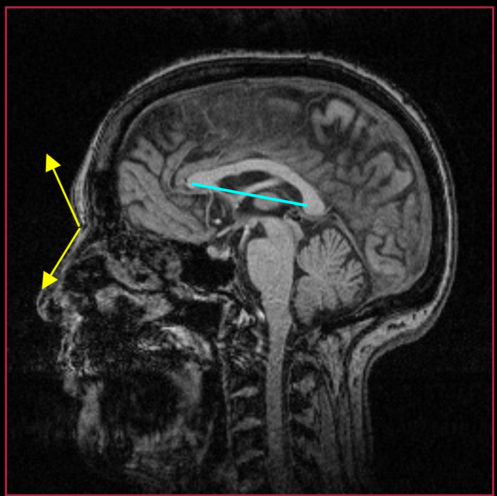
Rigid body transformation

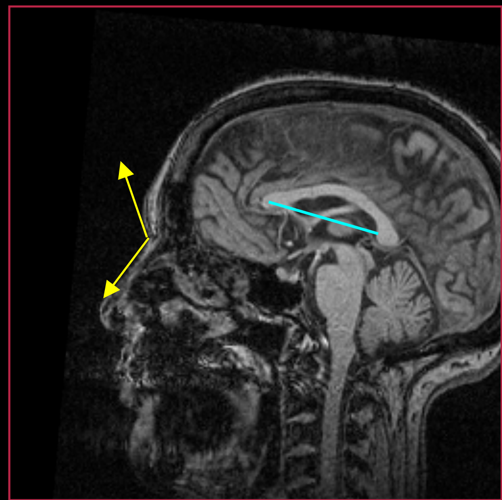
- Translation and rotation
- Rigid body
- Angles and distances are kept

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$a_i' = R a_i + t$$

$$w = (\Delta x, \Delta y, \theta)$$





Rigid body transformation

Transformation

$$a_i' = Ra_i + t$$

Rotation matrix

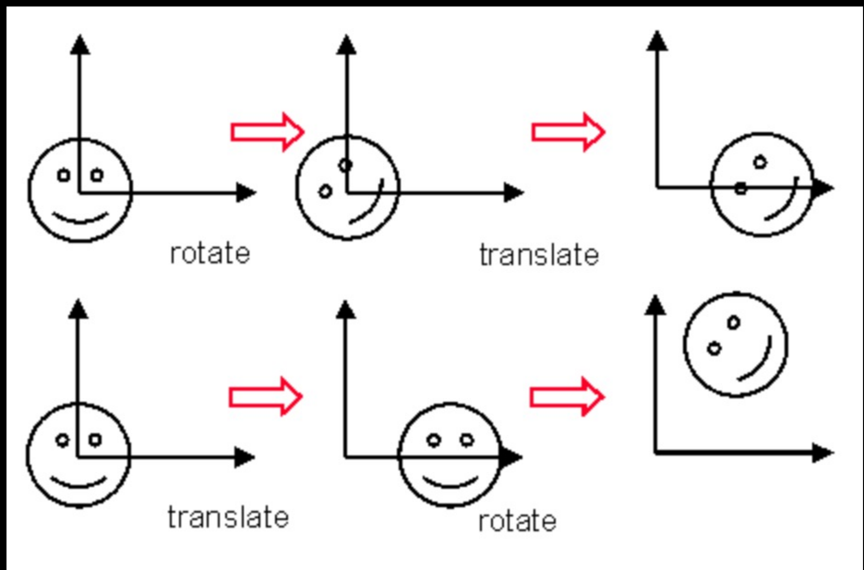
$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Objective function

$$F = \sum_{i=1}^{N} \|(Ra_i + t) - b_i\|^2$$

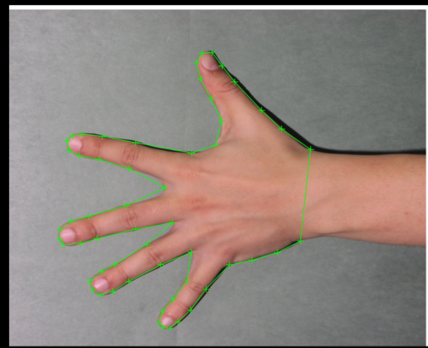
Optimal rigid body transformation

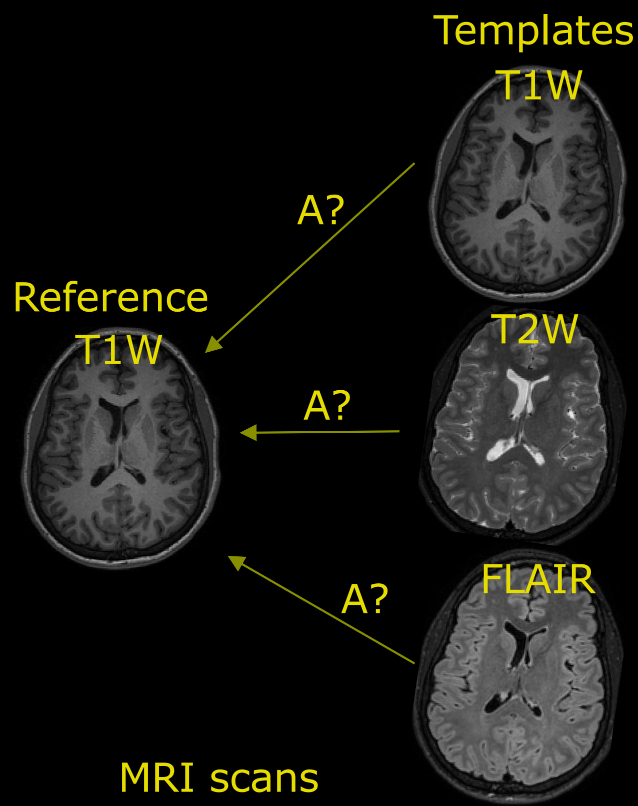
- The minimum of the objective function can be found in several ways
- The rotation can be found analytically by *singular value decomposition*
- A tip: Always start by matching the centre of masses



Similarity measures

- **Landmarks** - time consuming to obtained
- Alternative: **joint intensity histograms?**





- Same subject
- Same intensity histogram

- Same subject
- Different intensity histogram

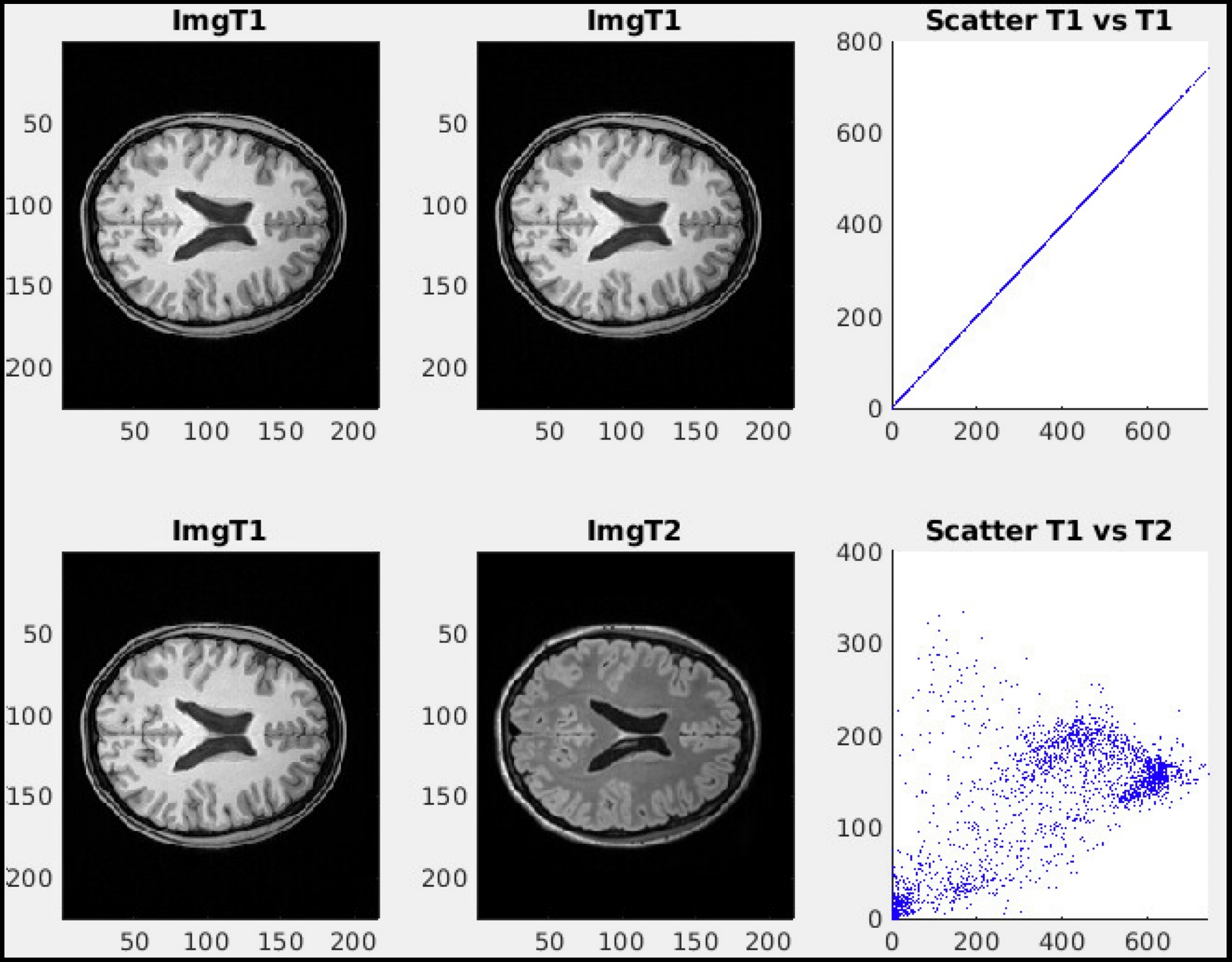
- Same subject
- Different intensity histogram

Joint intensity histograms

- Perfect registered: Optimal joint intensity agreement

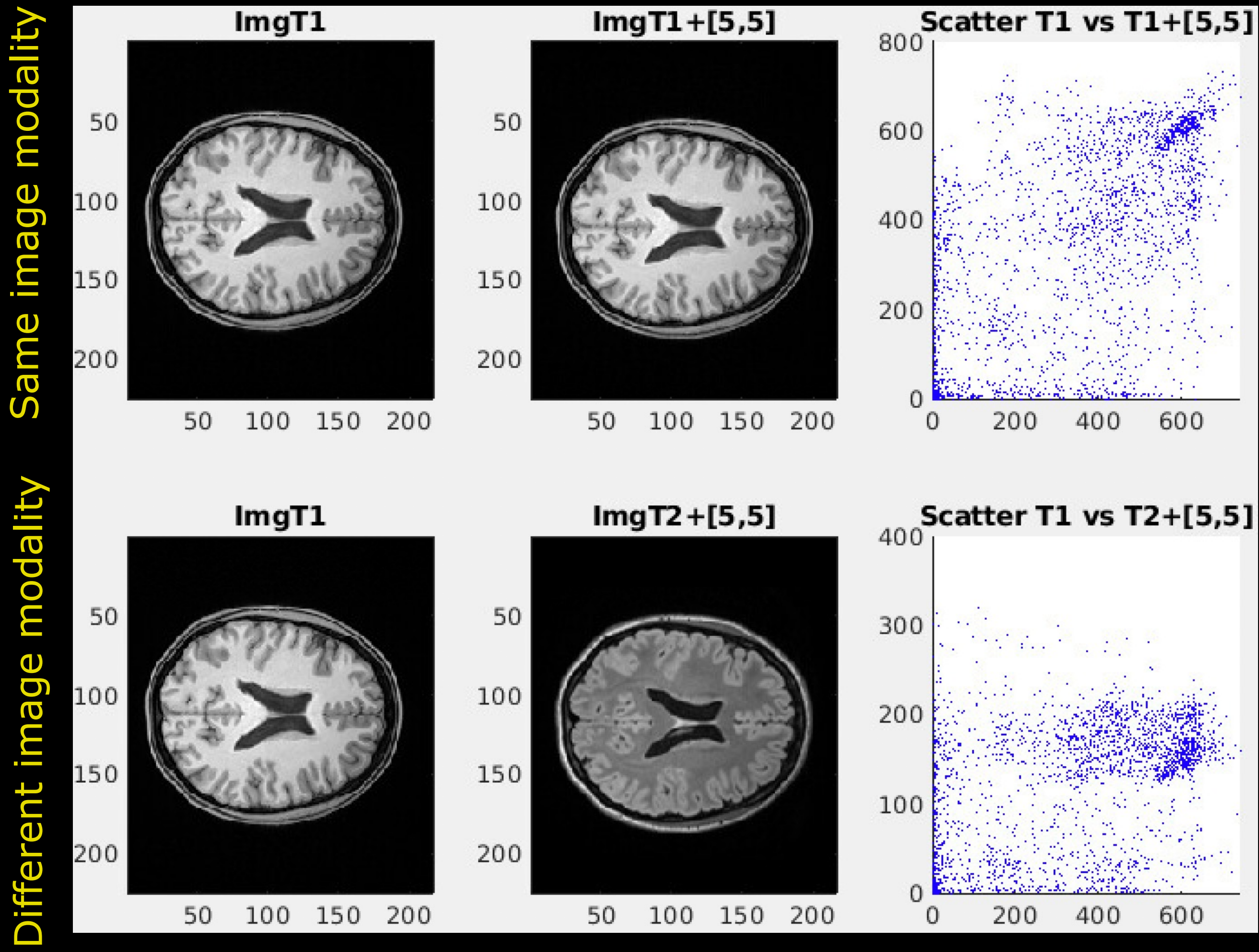
Same image modality

Different image modality



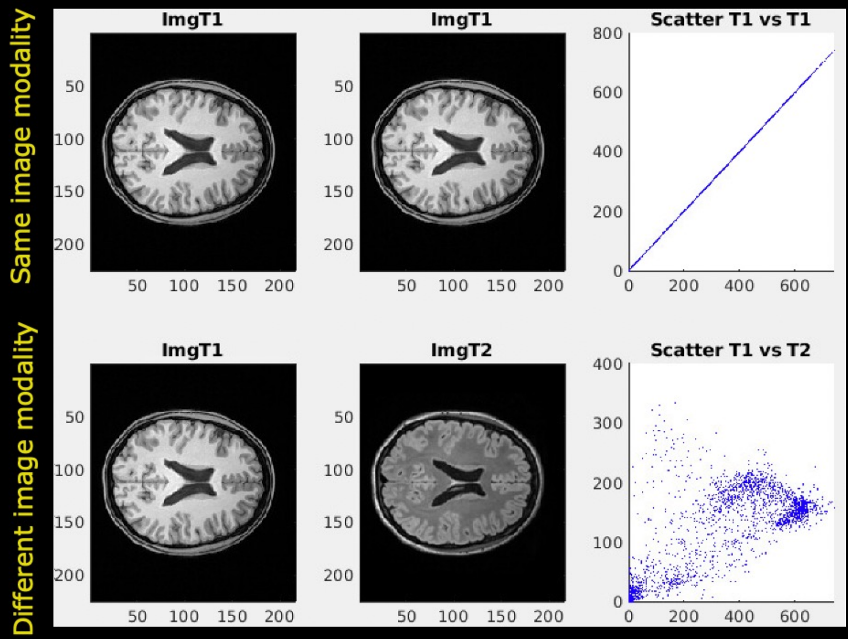
Joint intensity histograms

- Small translation difference: Lower joint intensity agreement



Joint intensity histograms

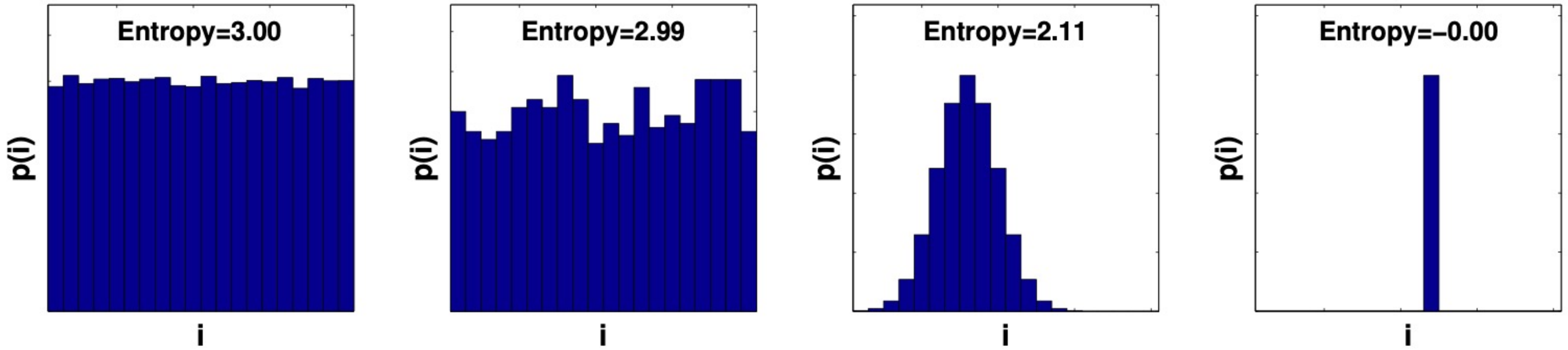
- Similarity measures to find transformation
- Many methods exist, but two types dominate:
  - Cross-correlation based
    → Fast to estimate, not optimal choice if different image modalities
  - Joint entropy based also known as Mutual Information (MI)
    → Slow to estimate, robust also if different modalities



Similarity measure - Entropy

- A information content measure
- Entropy (Shannon-Weiner):

$$H = -\sum_i p_i \log p_i$$

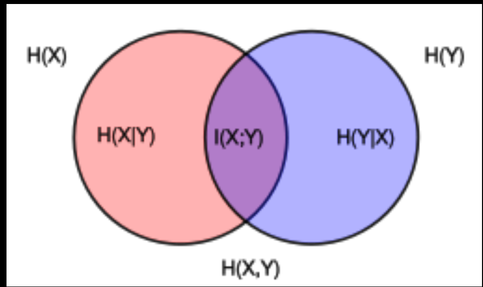


Joint entropy - Mutual information

- Joint entropy $H = - \sum_{X,Y} p_{X,Y} \log\, p_{X,Y}$

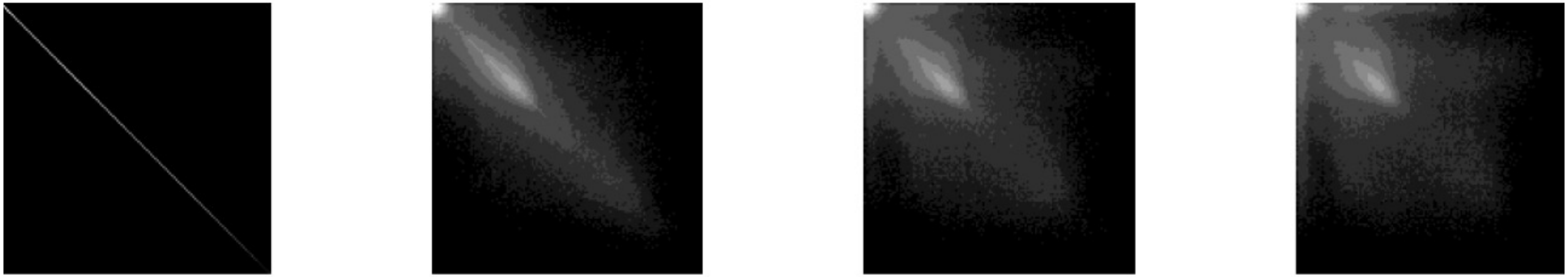
- Similarity measure: The more similar the distributions, the lower the joint entropy compared to the sum of the individual entropies

$$H(X,Y) \leq H(X) + H(Y)$$



en.wikipedia.org/wiki/Mutual_information

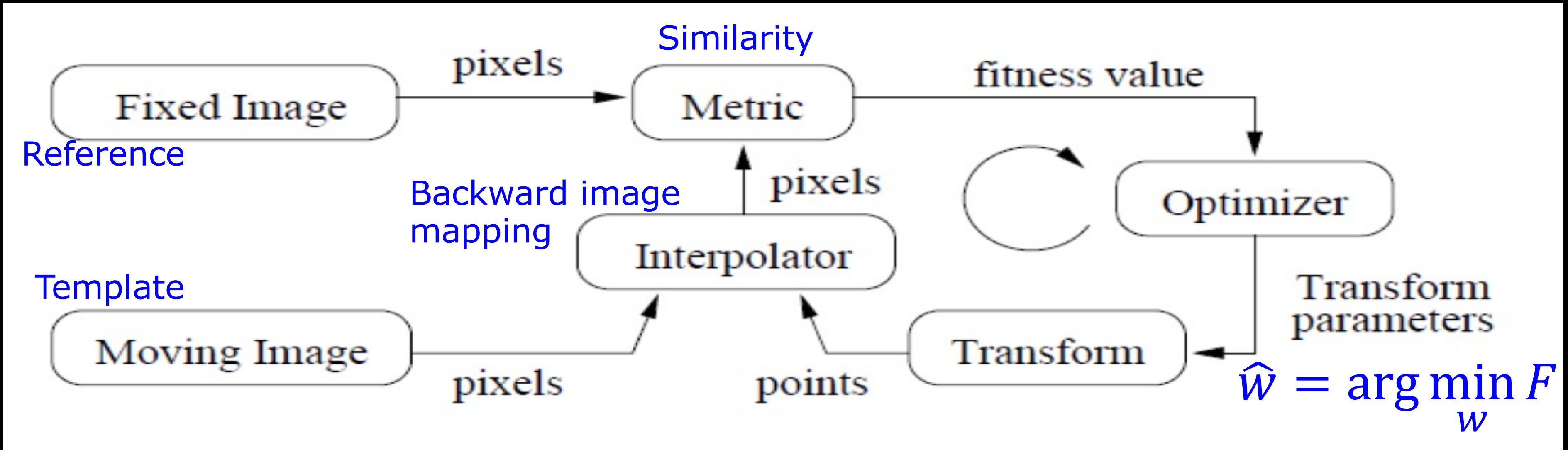
- Example (Pluim et al., 2003, TMI)



3.82 6.79 6.98 7.15

The image registration “pipeline”

- Register *Template image* to *Reference image via geometrical transformations*
- Select a similarity measure to map coordinates from template
- Objective function - Find optimal parameters: $\hat{w} = \arg\min_{w} F$
- The solution is often found by numerical optimisation (optimizer)



... Or use existing methods !!

What did you learn today?

Geometrical transformation

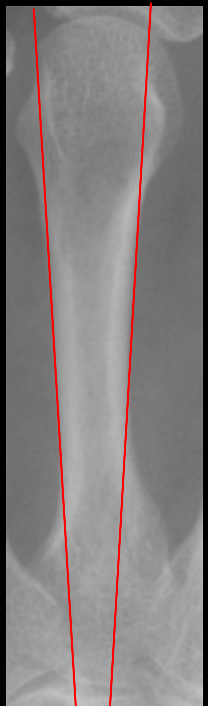
- Construct a translation, rotation, scaling, and shearing transformation matrix of a point
- Use transformation matrices to perform point transformations
- Describe the difference between forward and backward mapping
- Transform an image using backward mapping and bilinear interpolation

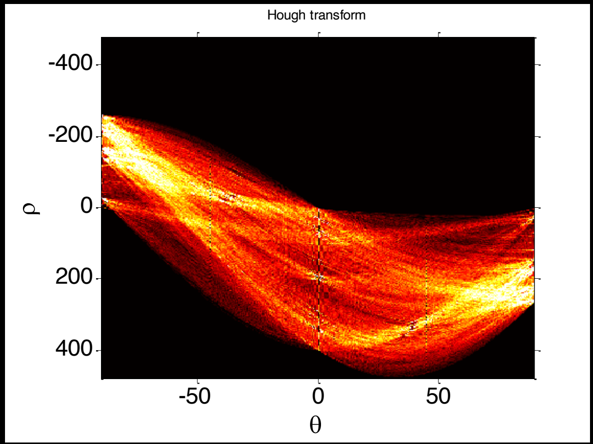
Image registration

- Describe the image registration
- Describe the different types of landmarks
- Annotate a set of image using anatomical landmarks
- Describe the objective function used for landmark and joint histogram based registration
- Compute the optimal translation between two sets of landmarks
- Use the rigid body transformation for image registration
- Describe the general ”pipeline” for image registration

Next week

- Hough Transformation and Path Tracing





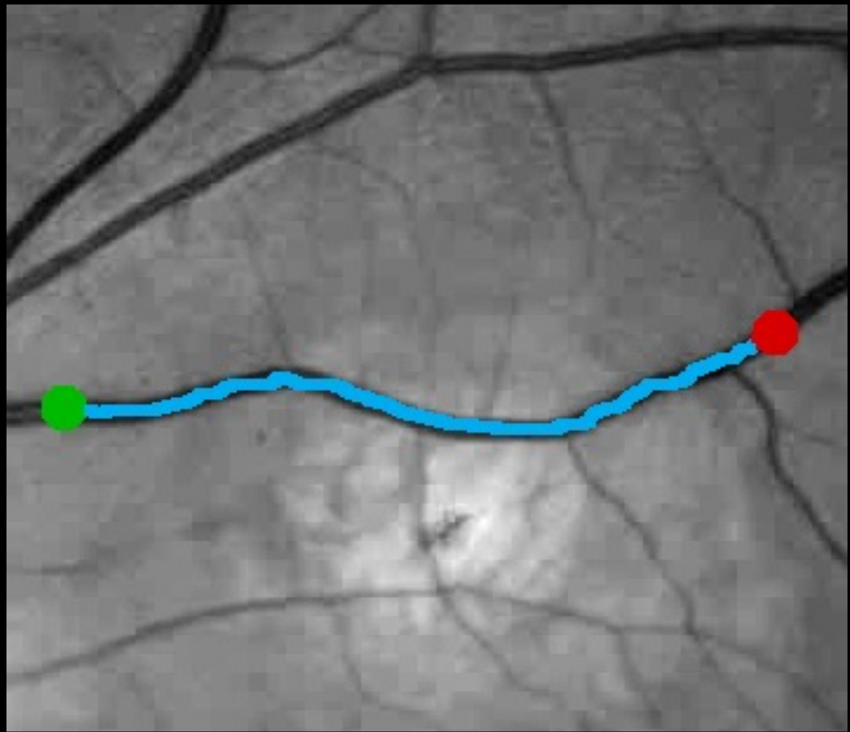


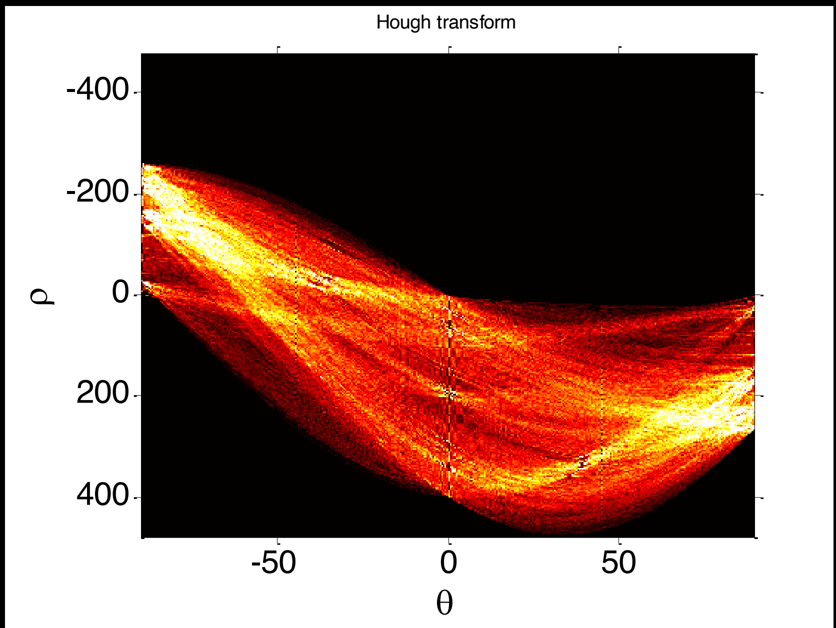
Image Analysis

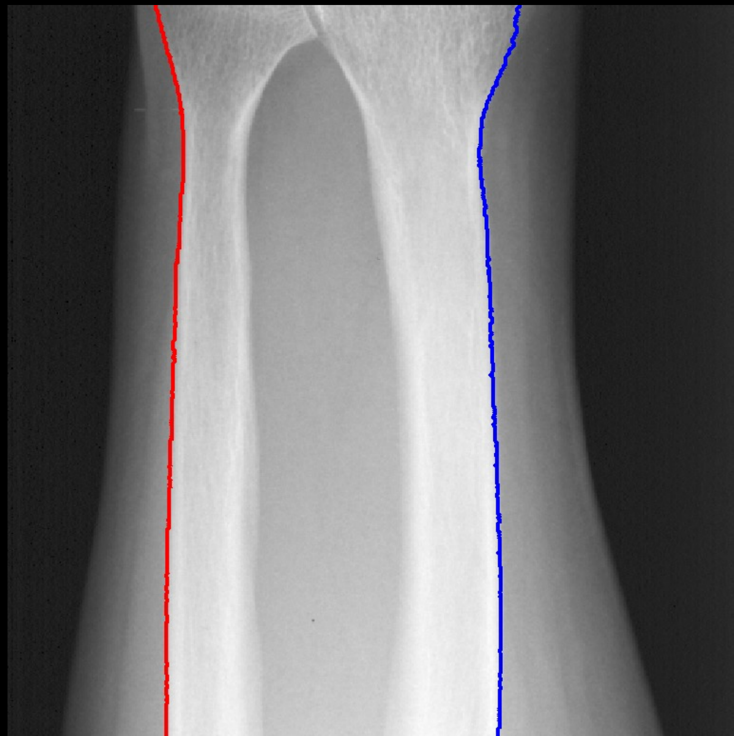
Tim B. Dyrby
Rasmus R. Paulsen
DTU Compute

tbdy@dtu.dk

http://www.compute.dtu.dk/courses/02502

Lecture 8 – Hough Transformation and Path Tracing





Go to **www.menti.com** and use the code **4006 2653**

Quiz testi… …ow long time did it take to develop the Dijkst…

Breaking NEWS!

One morning I was shopping in Amsterdam with my young fiancée, and tired, we sat down on the café terrace to drink a cup of coffee and I was just thinking about whether I could do this, and I then designed the algorithm for the shortest path. As I said, it was a twenty-minute invention.

— Edsger Dijkstra, in an interview with Philip L. Frana, Communications of the ACM, 2001[3]

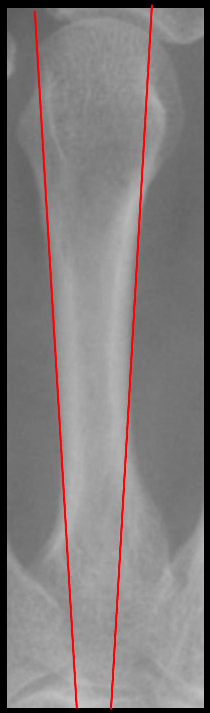
0	0	0	0	0	0
3 years	1 year	2 weeks	1 hour	20 min	37 sec

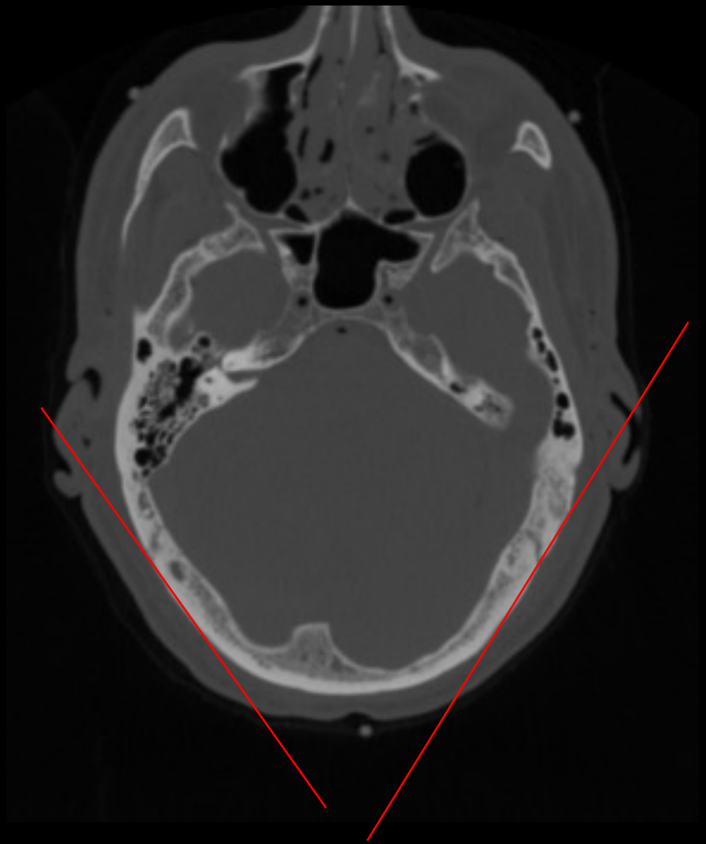
What can you do after today?

- Use the Hough transform for line detection
- Describe the slope-intercept, the general form and the normalised form of lines
- Describe the connection between lines and the Hough space
- Use edge detection to enhance images for use with the Hough transform
- Use dynamic programming to trace paths in images
- Describe how an image can be used as a graph
- Describe the fundamental properties of a cost image
- Compute the cost of path
- Compute an accumulator image for path tracing
- Compute a back tracing image for path tracing
- Choose appropriate pre-processing steps for path tracing
- Describe how circular structures can be located using path tracing

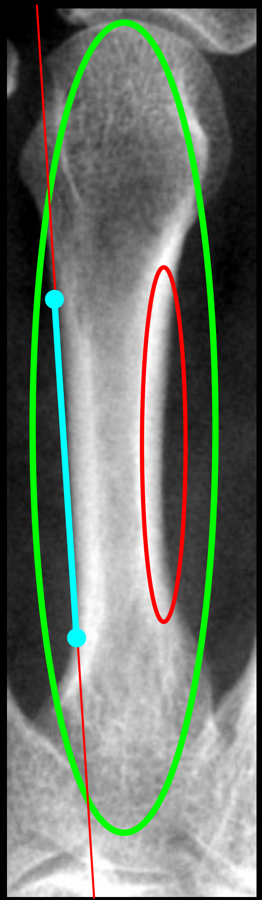
Line Detection

- Find the lines in an image





What is a line?



- It can be the entire object
  - Large scale
- Can also be the border between an object and the background
  - Small scale
- Normally only locally defined

Enhancing the lines

Original Prewitt Edge

- We want to locate the borders
  - Enhance them
- Filtering (Prewitt)
- Edge detection

Prewitt:

Vertical

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

Horizontal

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

What is a line II?

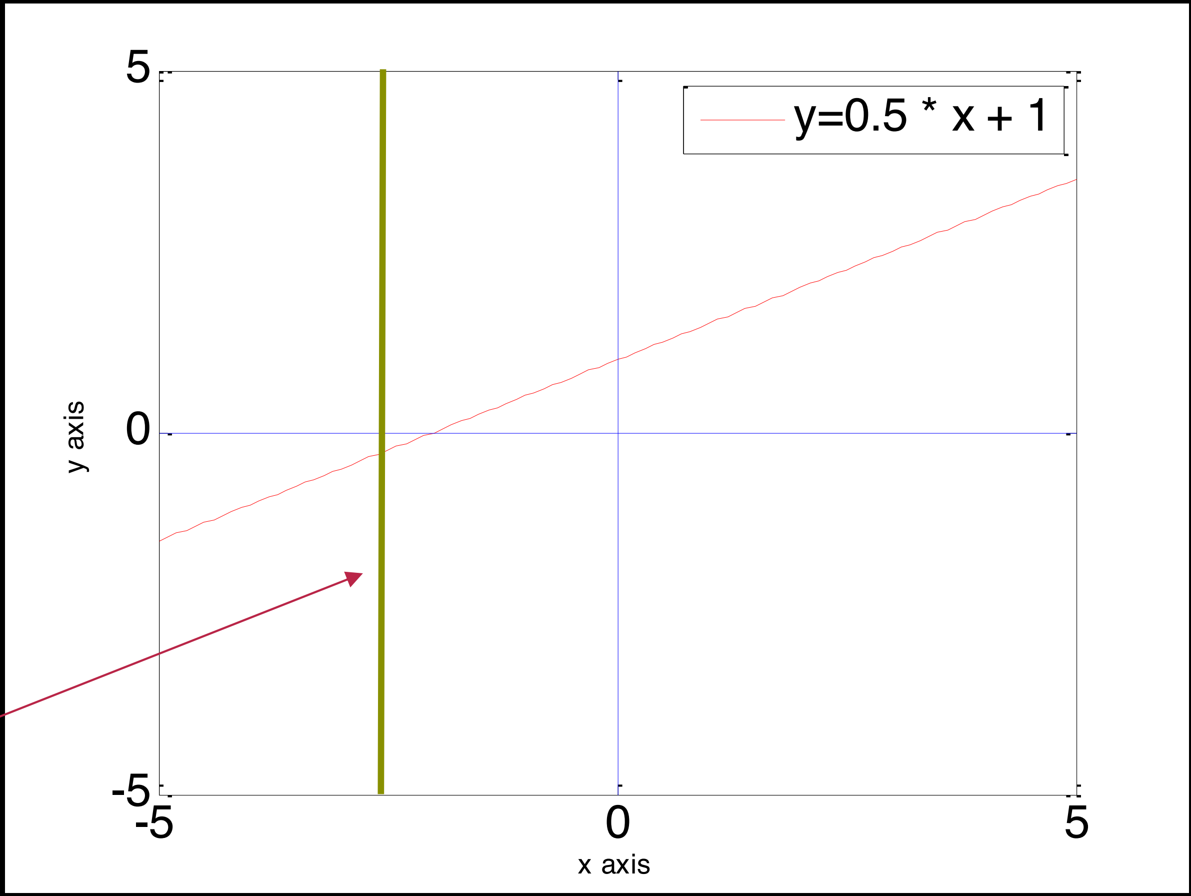
- Result of the edge filter is a selection of white pixels
- Some of them define a line
  - Not a perfect straight line
  - “Linelike”
- How do we find the collection of points that define a line?

Mathematical line definition

- The classical definition (slope-intercept form)

$$y = ax + b$$

Slope Intercept

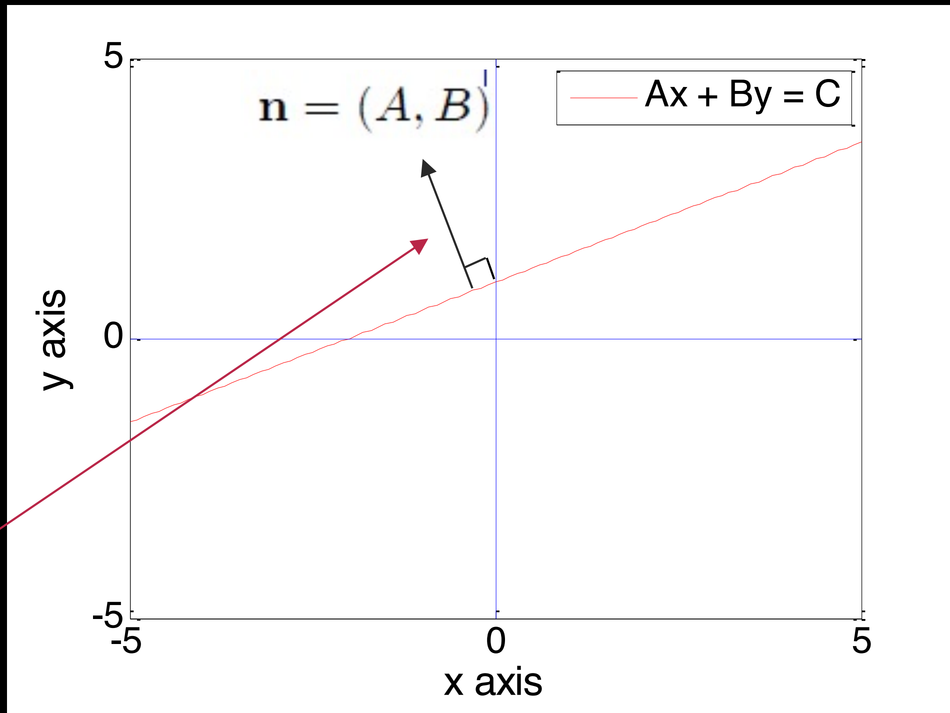


Can not represent lines that are vertical

Mathematical line definition

- General definition (the normal form)

$$Ax + By = C$$



- With

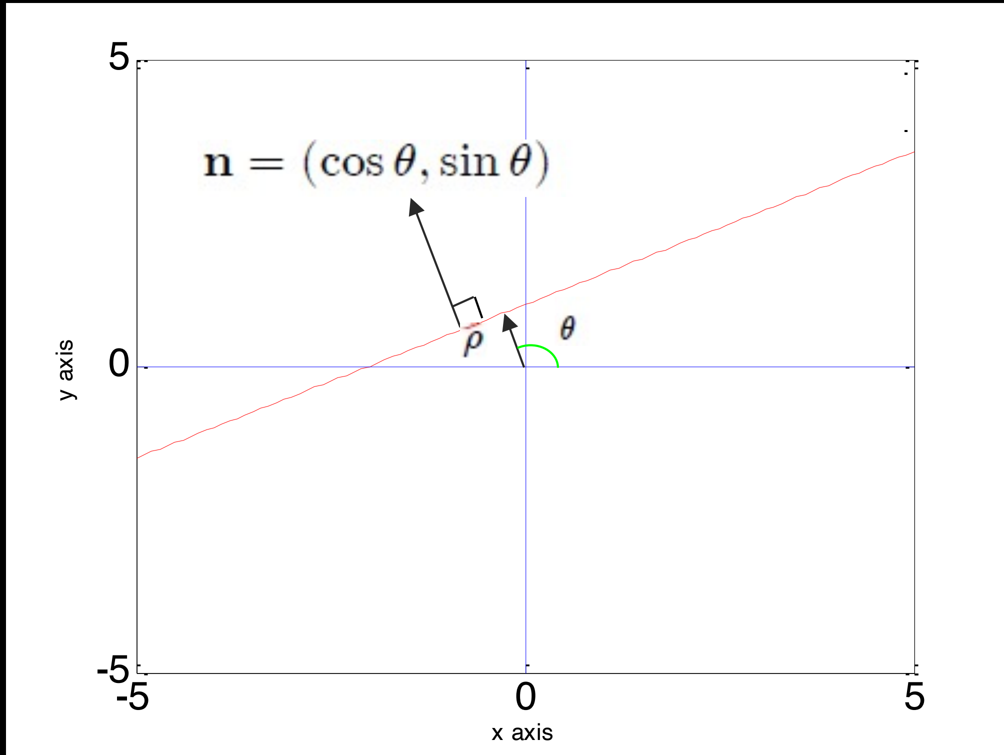
$$A^2 + B^2 = 1$$

Line normal

Mathematical line definition

- Normal form parameterisation

$$x\cos\theta + y\sin\theta = \rho$$



- where
  - ρ is the distance from the origin
  - θ is the angle

$$(\cos\theta)^2 + (\sin\theta)^2 = 1$$

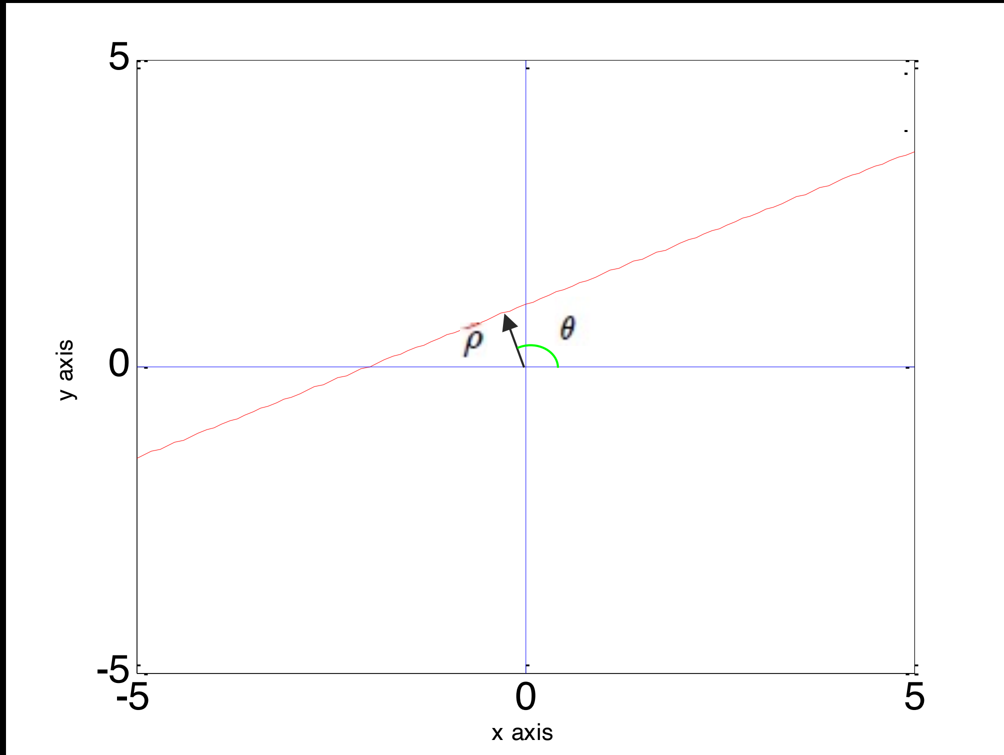
$$A^2 + B^2 = 1$$

Mathematical line definition

- Normal form parameterisation

$$x\cos\theta + y\sin\theta = \rho$$

- Therefore a line can be defined by two values
  - ρ
  - θ
- A line can therefore also be seen as a *point* in a (θ,ρ)-space



Converting lines between definitions

- From normal from to the slop-intercept form

The normal form: $p = x\cos\theta + y\sin\theta$

The slope-intercept form: $\mathrm{y} = ax + b$

Start: $p = x\cos\theta + y\sin\theta$

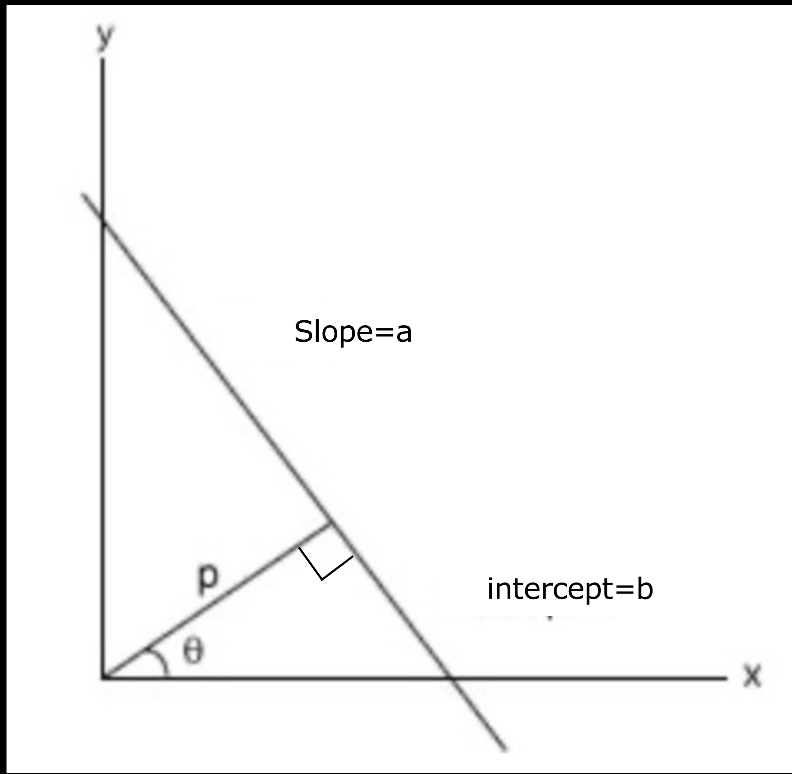
$$\text{-x}\cos\theta + p = \text{ysin}\,\theta$$

$$-x\ cot\theta + p\ cosec\theta = y$$

$$y = x * \boxed{(-cot\theta)} + \boxed{p(cosec\theta)}$$

Slope=a

Intercept=b



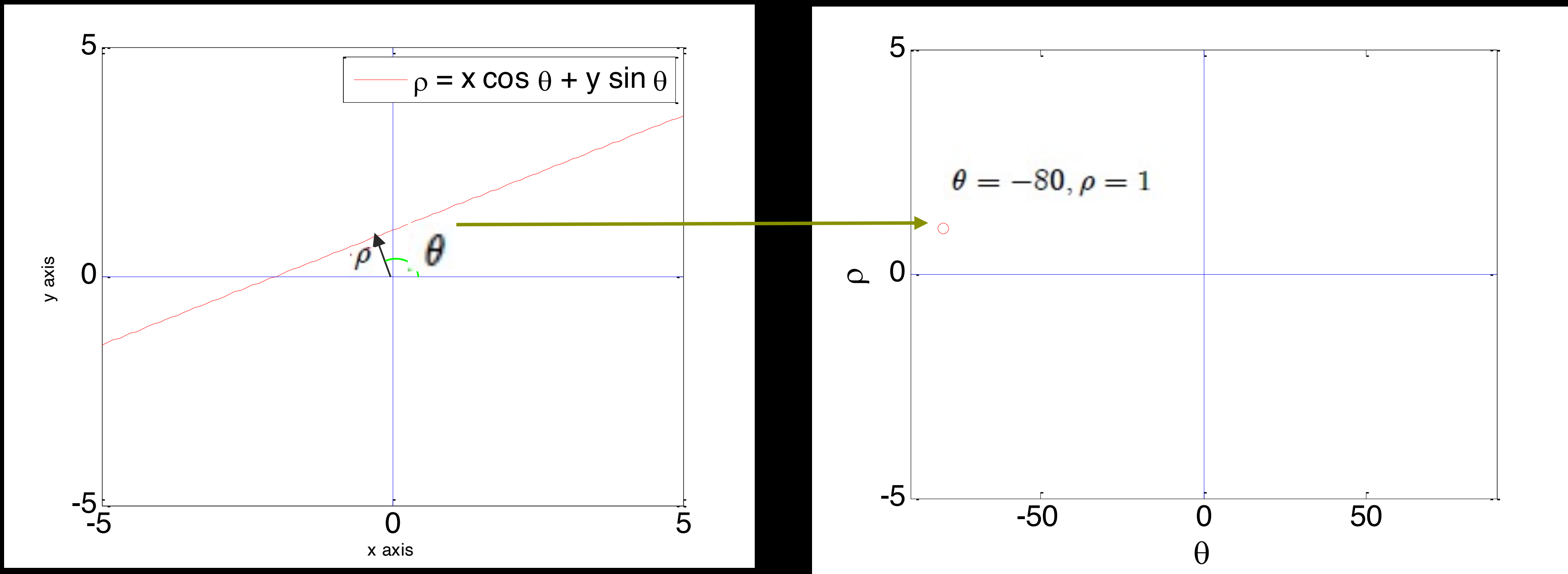
Something about angles

$\theta \in [0^o, 180^o[$ In the course notes

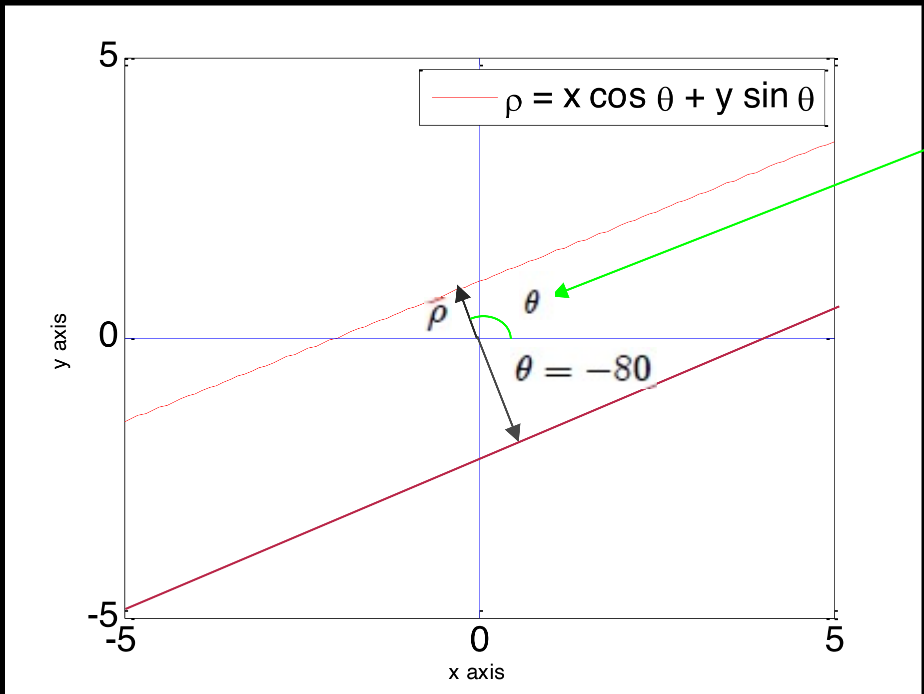
$\theta \in [-90^o, 90^o[$ In Matlab and in this presentation

Hough Space

$$-90^o < \theta < 90^o$$



More about angles



$$\theta = -80^o$$

Why?

$$\theta = 100^o$$

but Matlab only allows

$$-90^o < \theta < 90^o$$

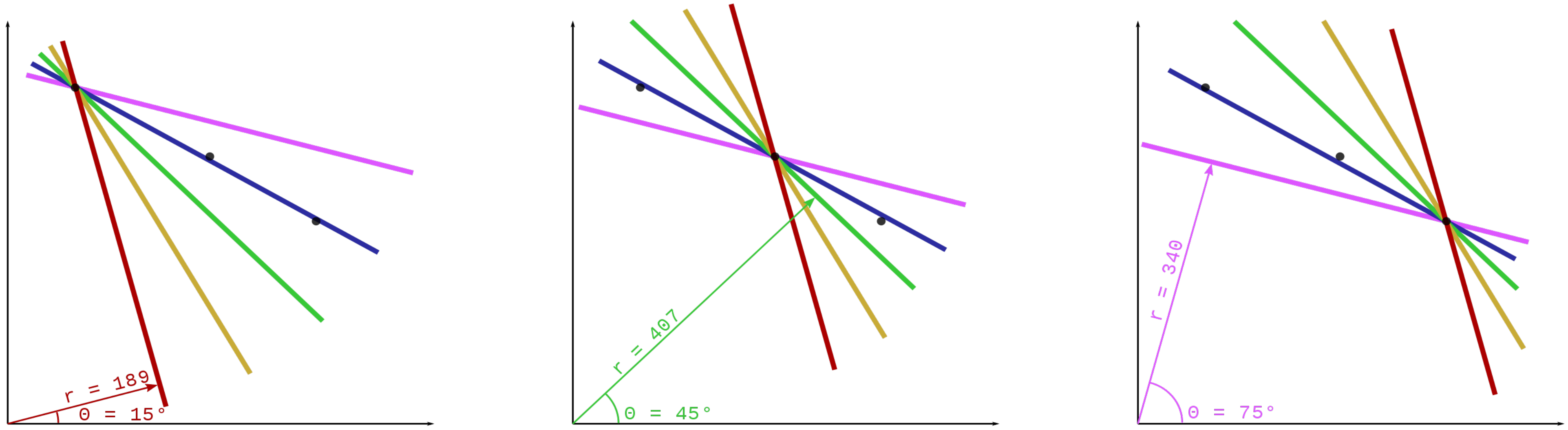
look at the mirror-projection of the normal

ρ is used to determine if it is the “upper” or “lower line”

Hough space: Let’s vote for general line …

- Basically a tool to find a line through points.
    1) Point coordinates: (x,y)
    2) Define origin
    3) Hough parameters space:(r,θ)
    4) Map all possible lines through a point for different θ
    5) “Vote” which line fit best through points

Points



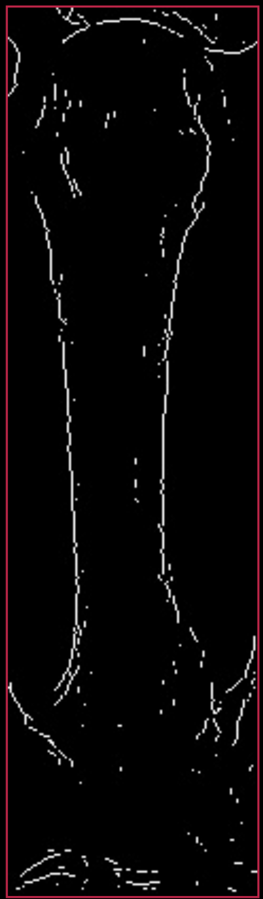
Hough space

θ	r
15	189.0
30	282.0
45	355.7
60	407.3
75	429.4

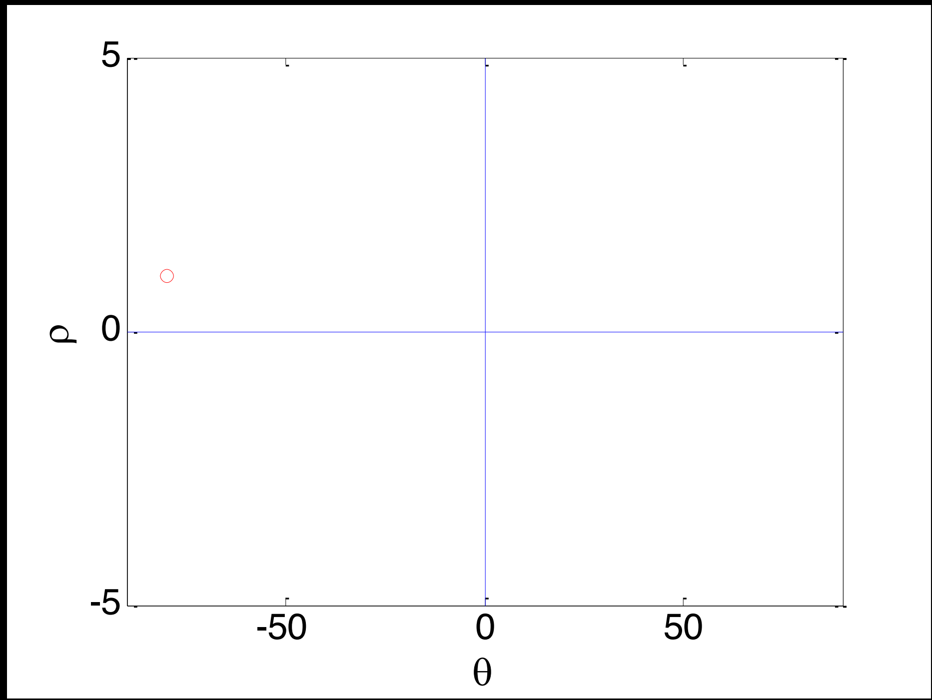
θ	r
15	318.5
30	376.8
45	407.3
60	409.8
75	385.3

θ	r
15	419.0
30	443.6
45	438.4
60	402.9
75	340.1

How do we use the Hough space?

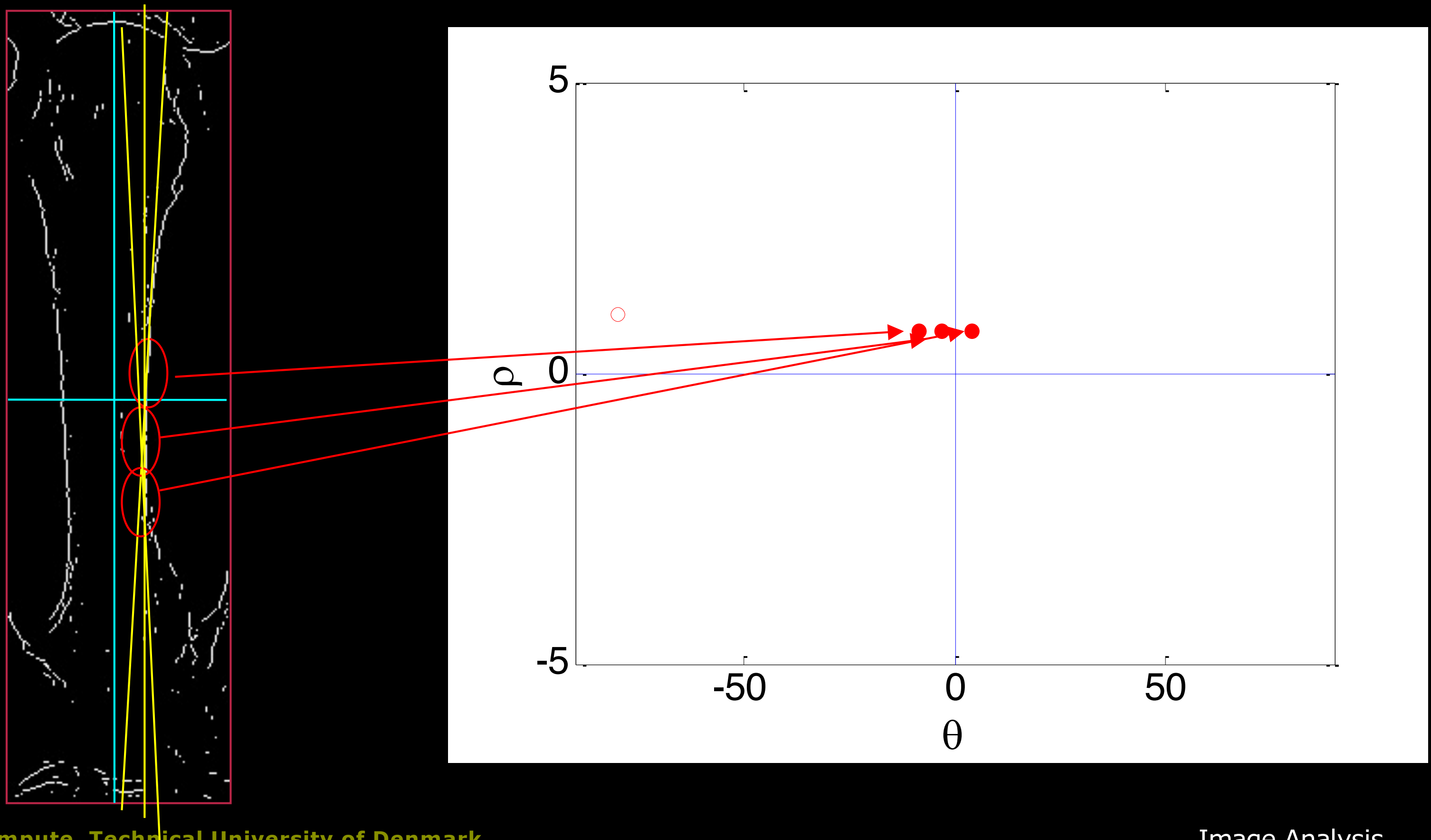


?



How do we use the Hough space?

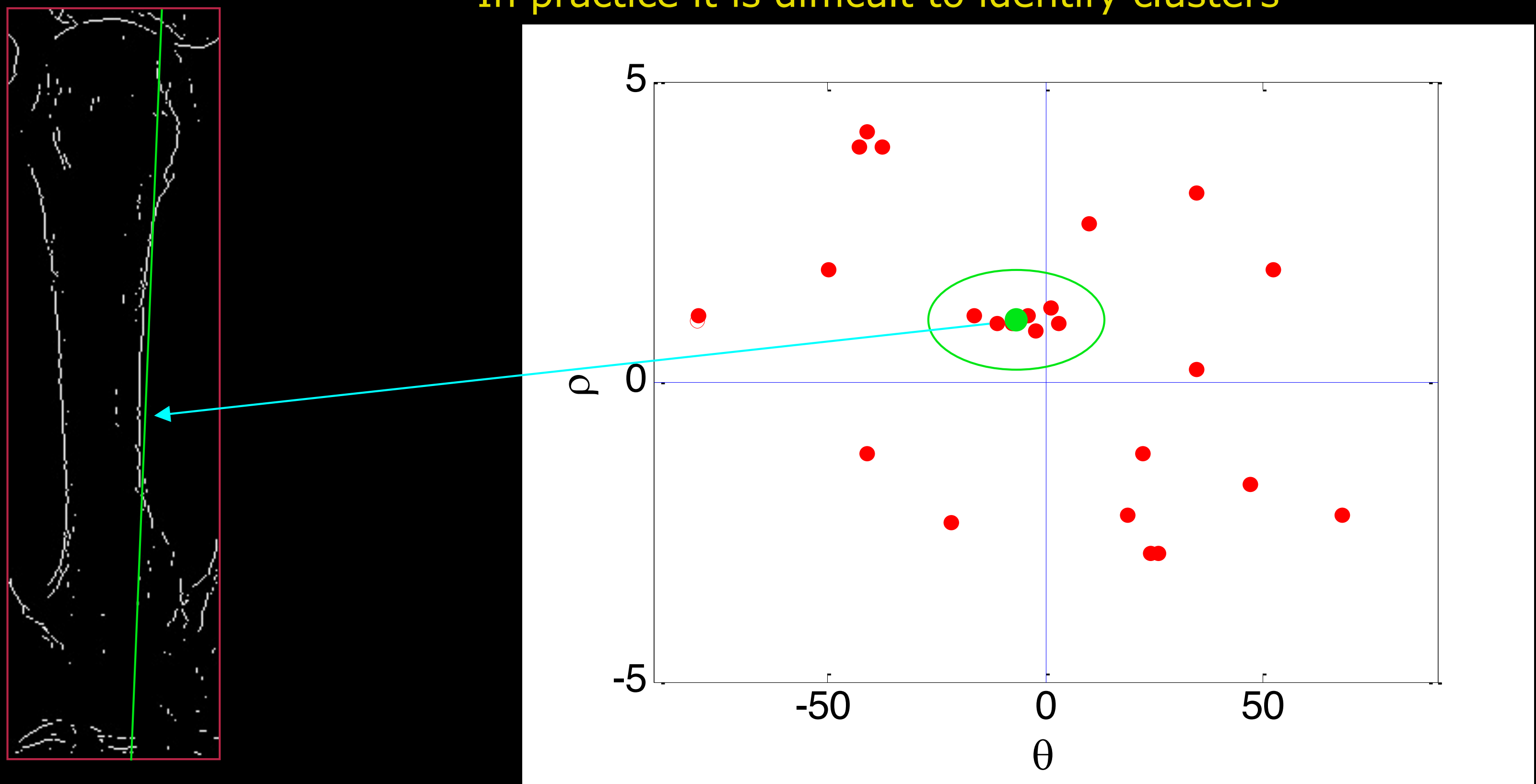
- What if every little "line-segment" was plotted in the Hough-space?



Filled Hough-Space

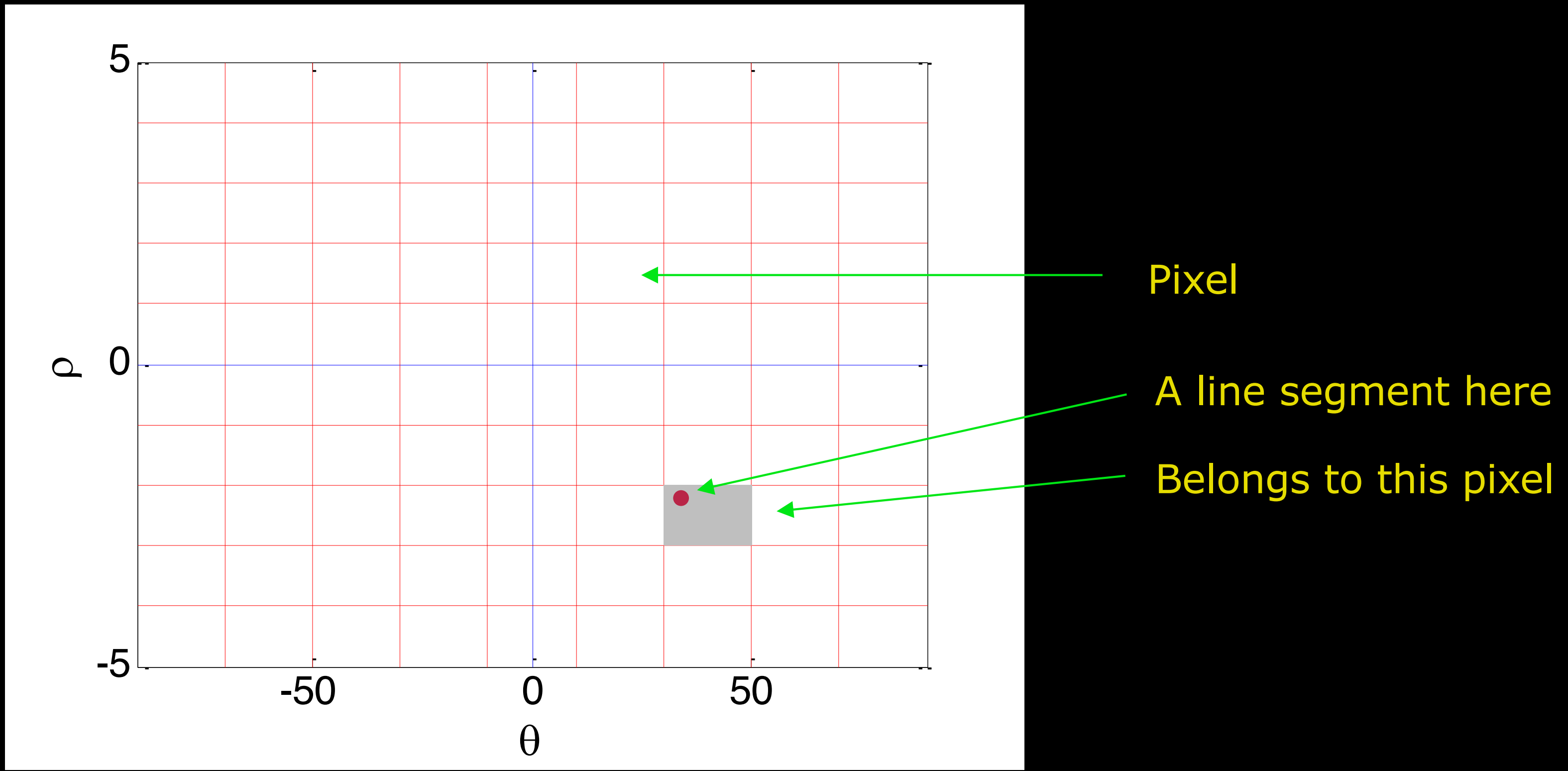
- All “line segments” in the image examined
- A “global line” can now be found as a cluster of points

In practice it is difficult to identify clusters



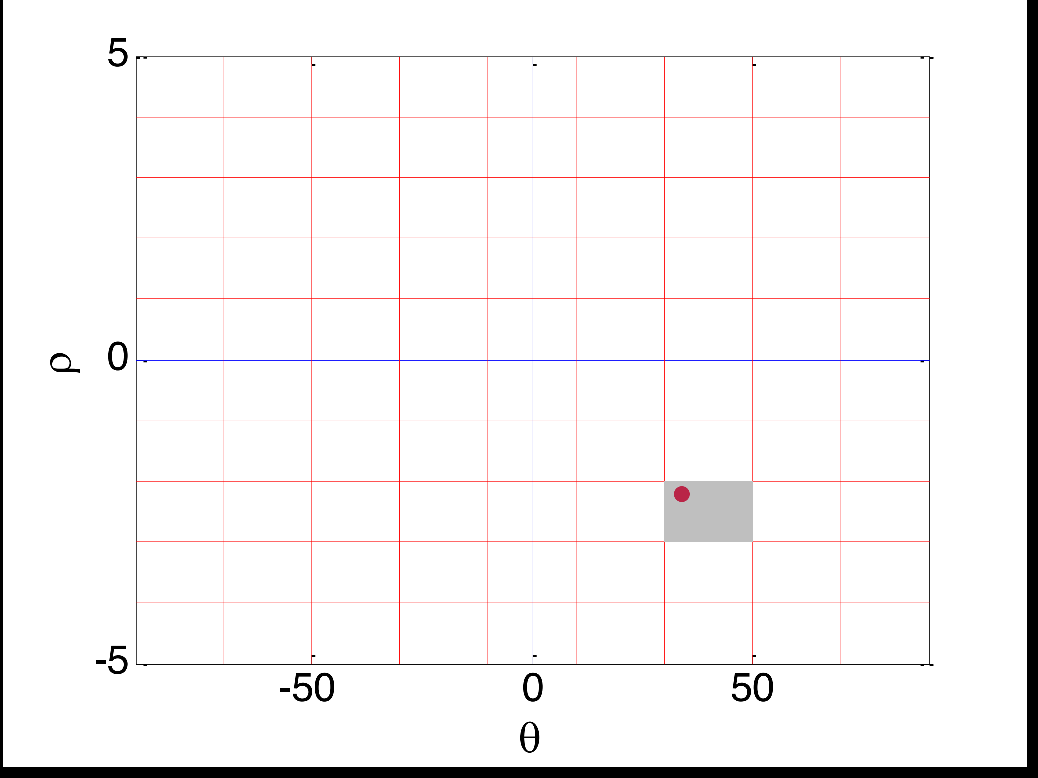
Hough transform in practise

- Hough Space is represented as an image
- It is *quantisized* – made into finite boxes



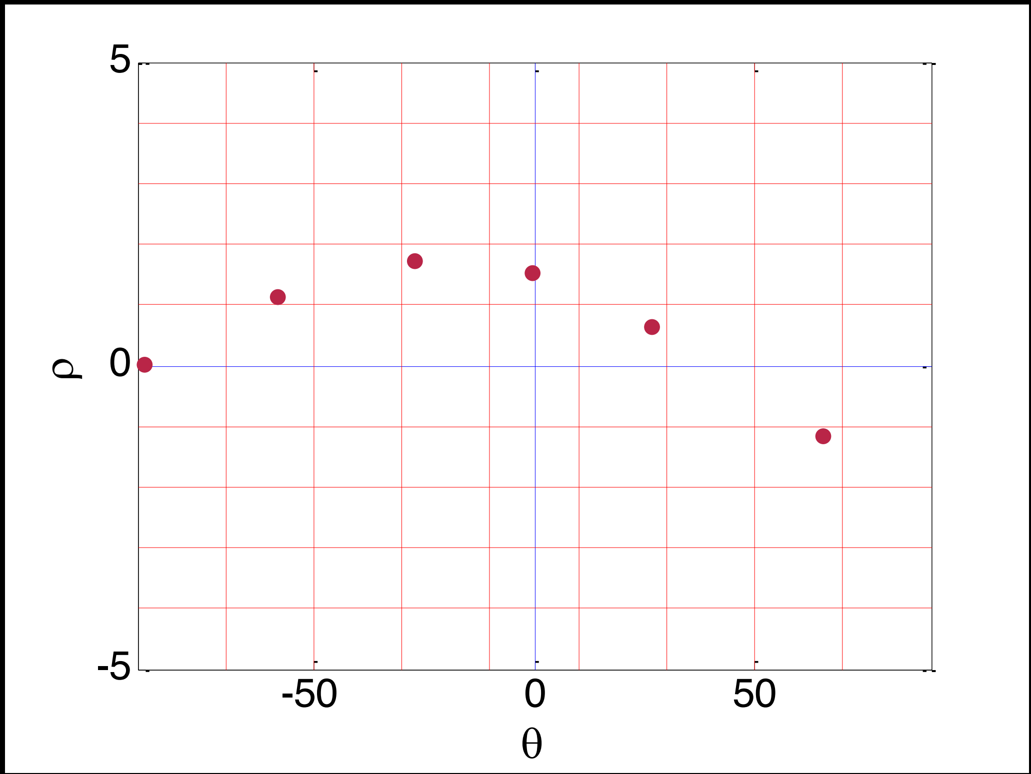
Hough transform as a voting scheme

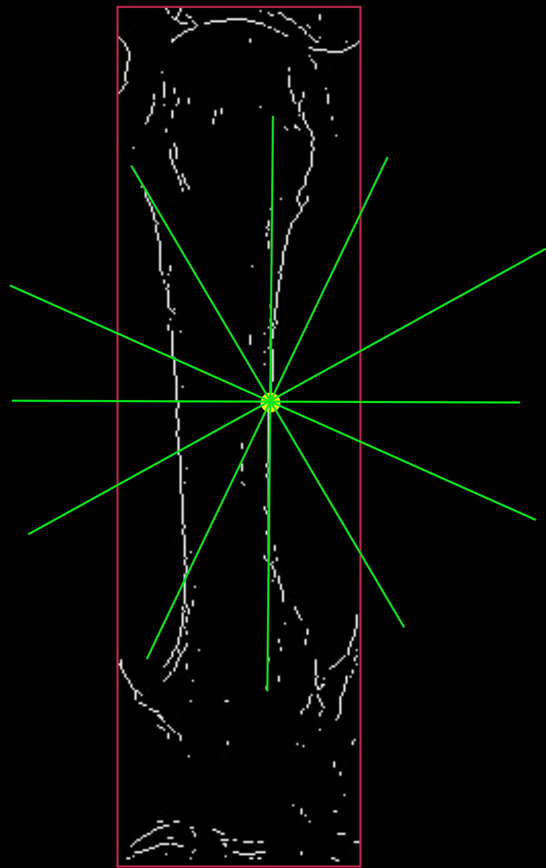
- The pixels in the Hough space are used to *vote* for lines.
- Each *line segment* votes by putting *one vote* in a pixel
- The pixels are also called *accumulator cells*



Hough transform per pixel

- In practise we do not use line segments
- Each pixel in the input image votes for all potential lines going through it.





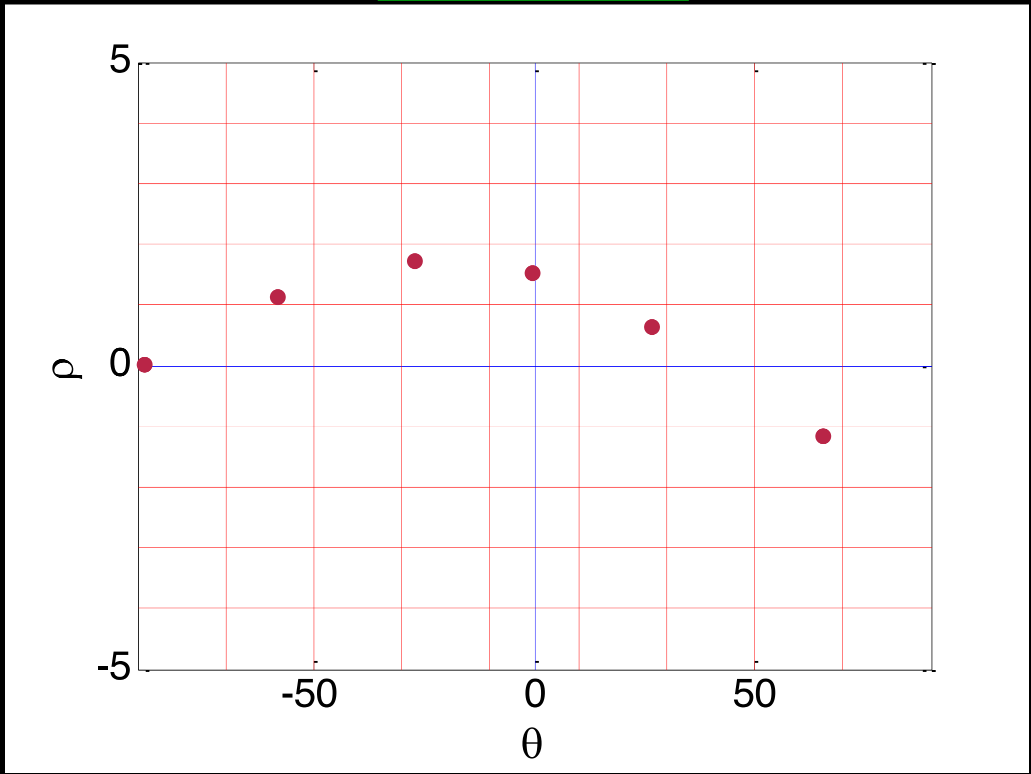
Hough transform per pixel

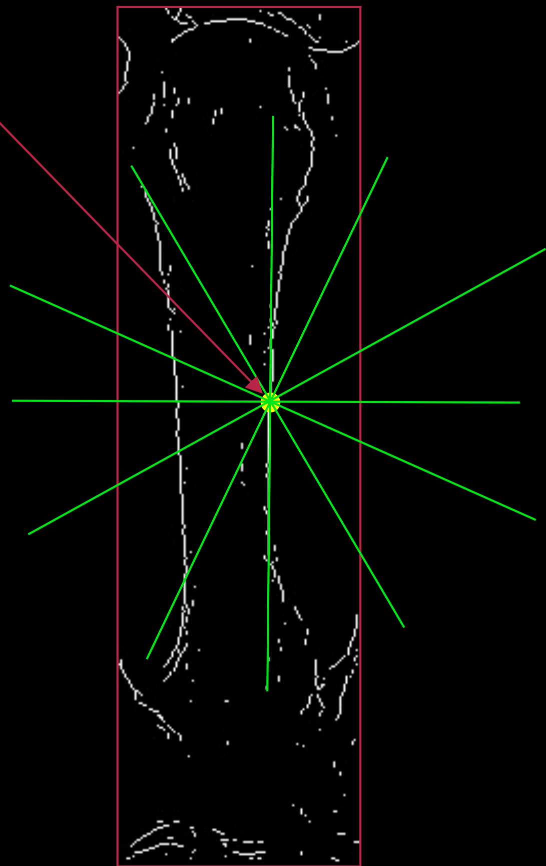
$$x\cos\theta + y\sin\theta = \rho$$

Go through all θ and calculate ρ

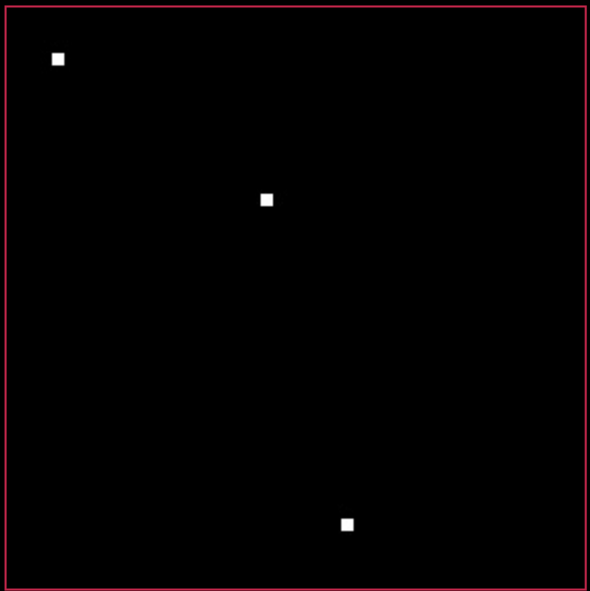
(x, y) are fixed

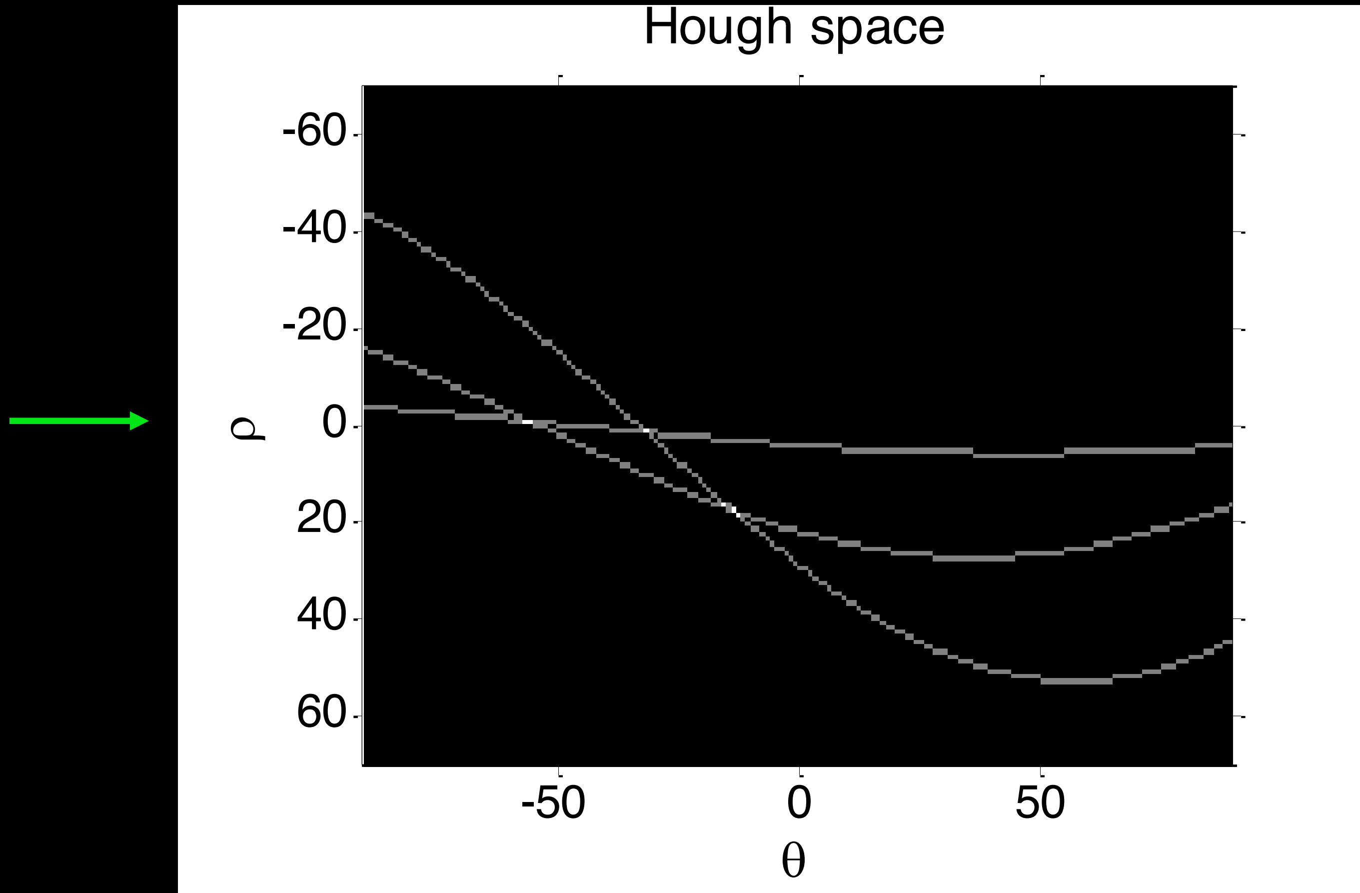
Sinusoid!



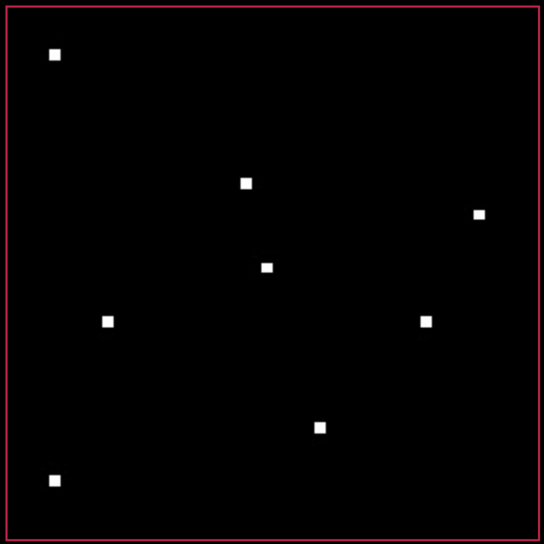


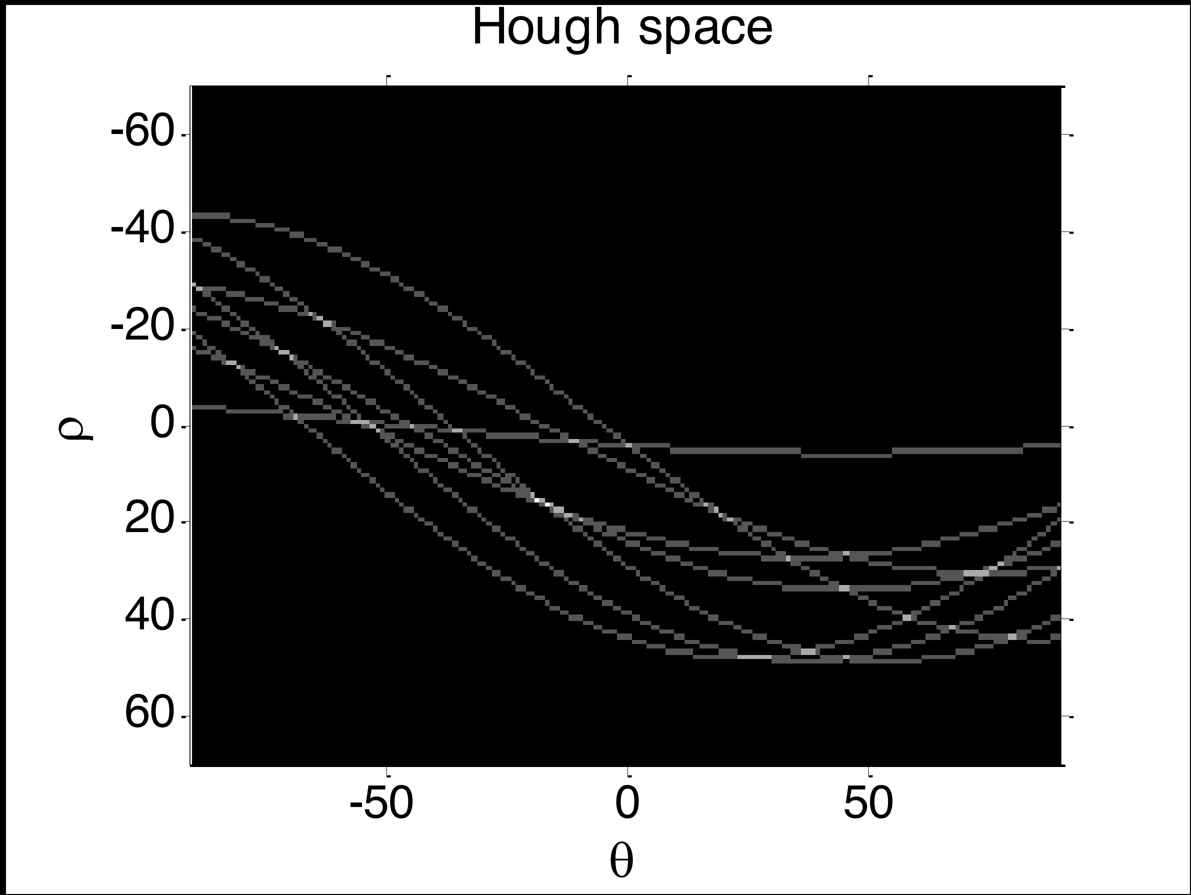
Real Hough Transform





Real Hough Transform II

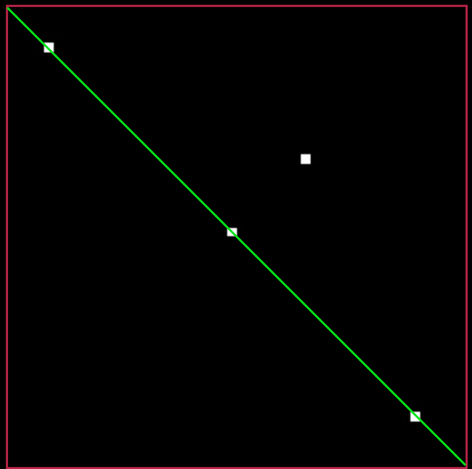


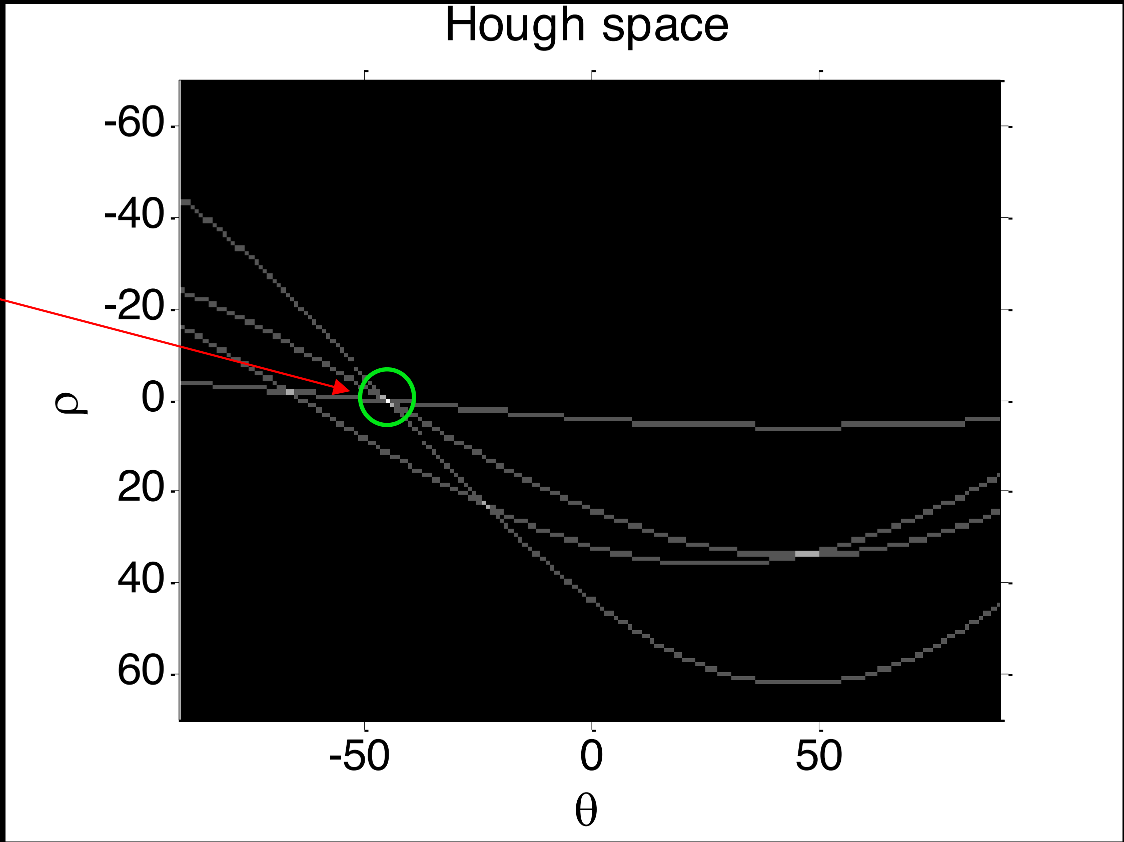


Real Hough Transform and lines

Spot the line!

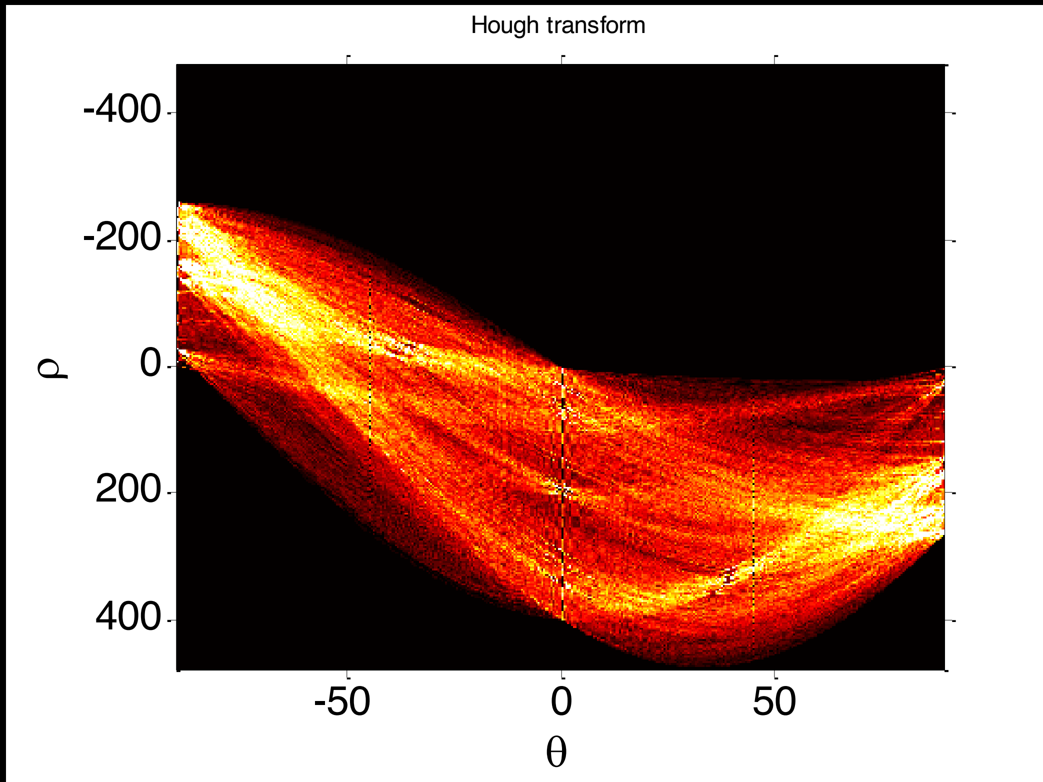
A maximum where Hough pixel has value 3



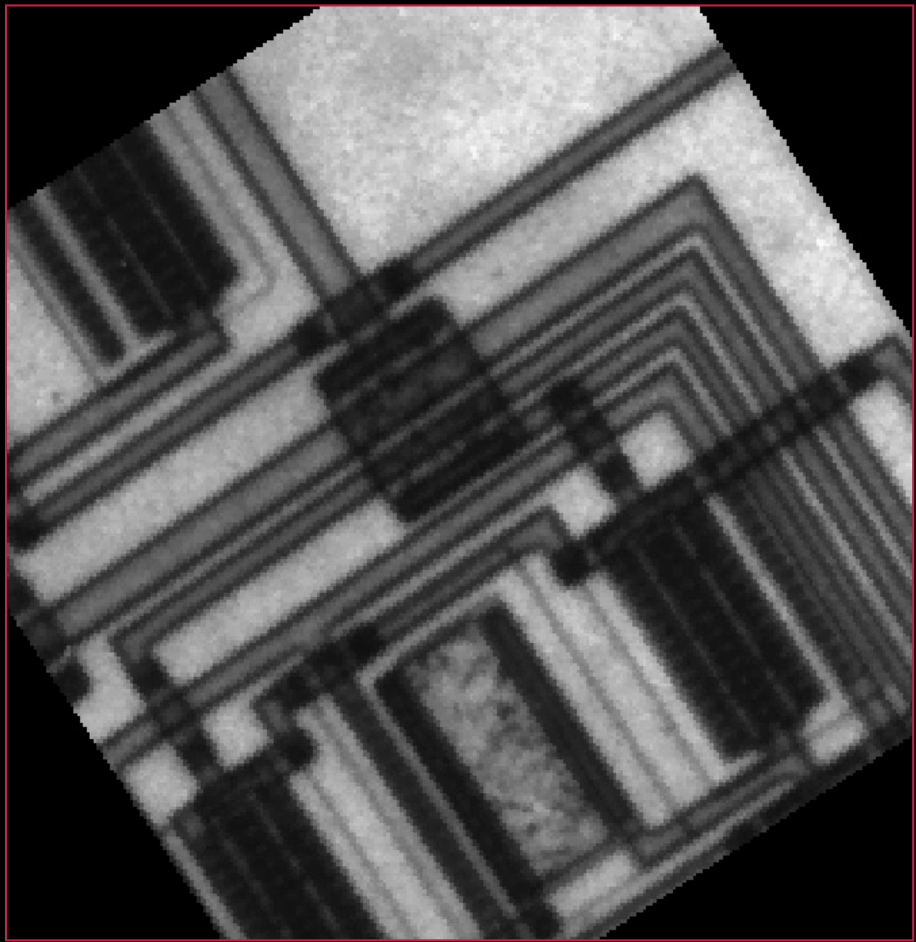


Finding the lines in Hough space

- The lines are found in Hough space where *most pixels have voted for there being a line*
- Can be found by searching for maxima in Hough Space

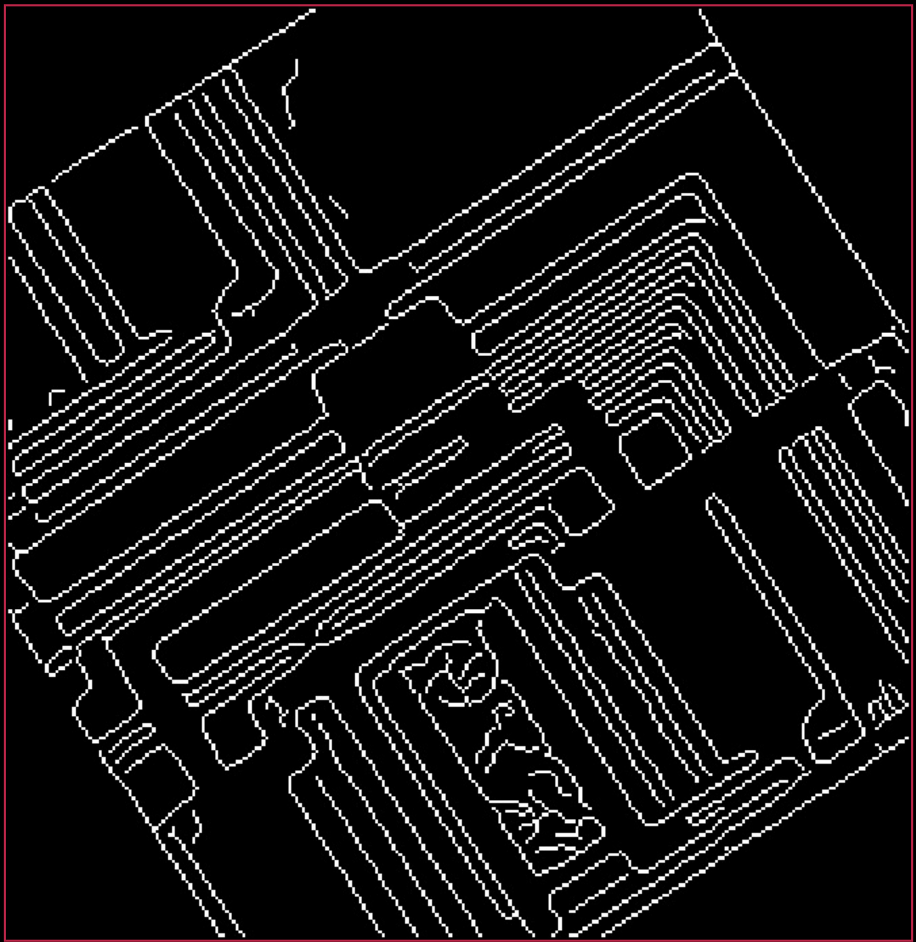


The practical guide to the Hough Transform



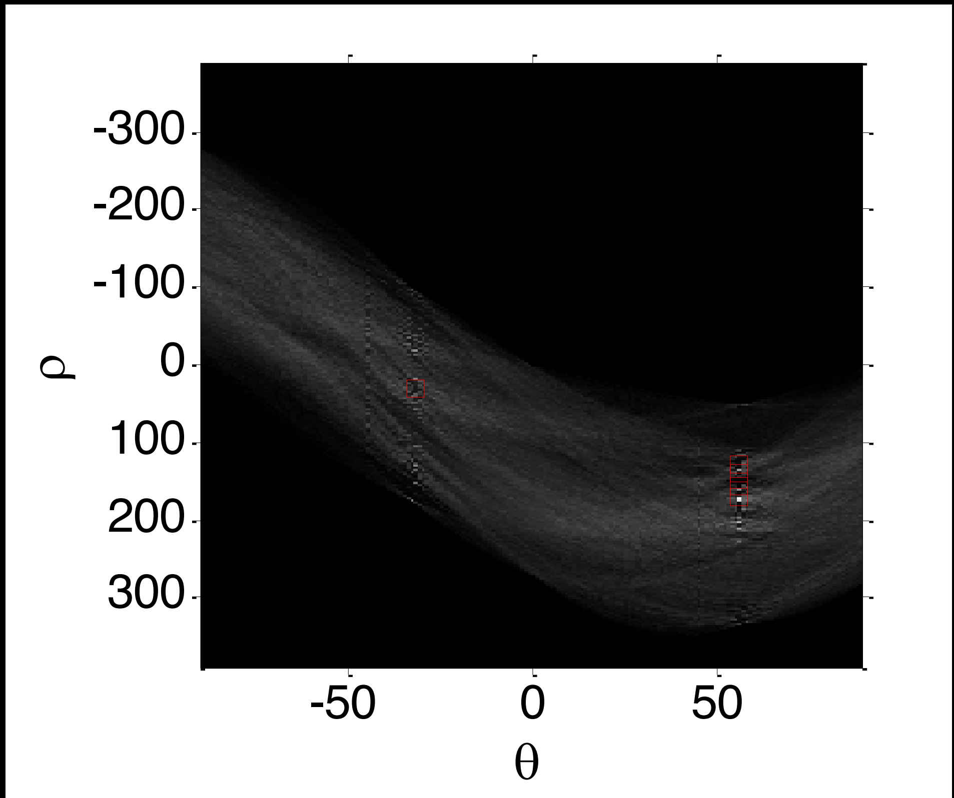
- Start with an input image

The practical guide to the Hough Transform



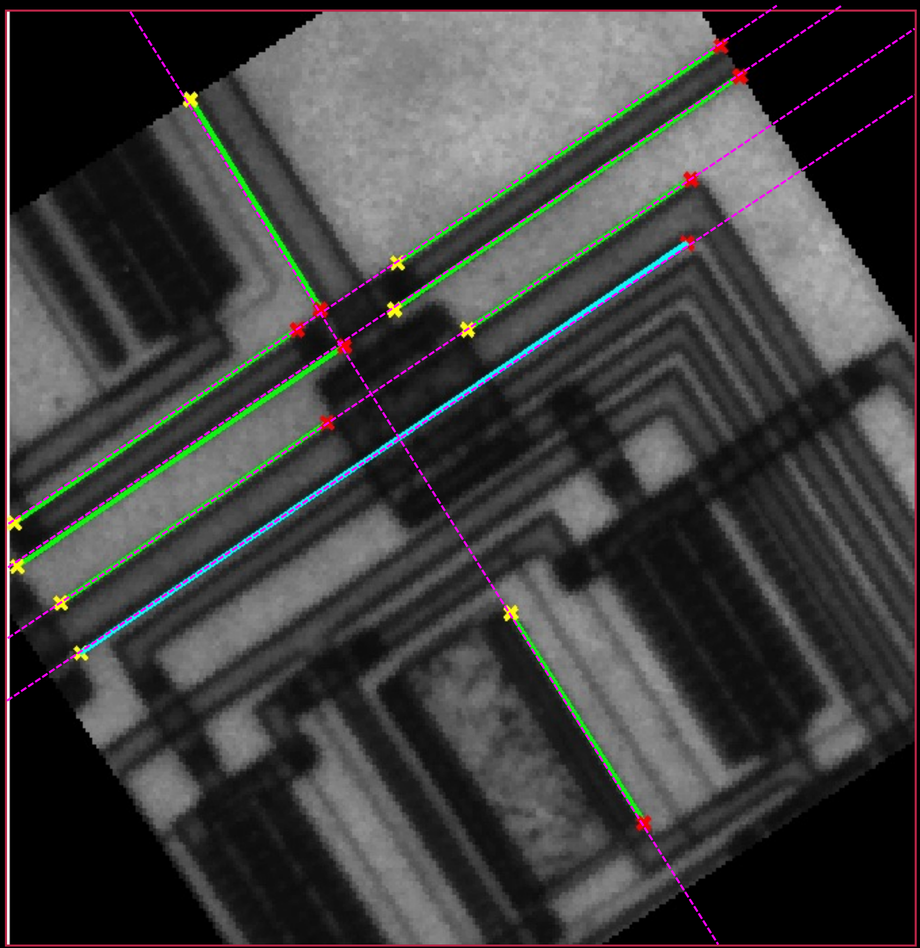
- Detect edges and create a binary image

The practical guide to the Hough Transform



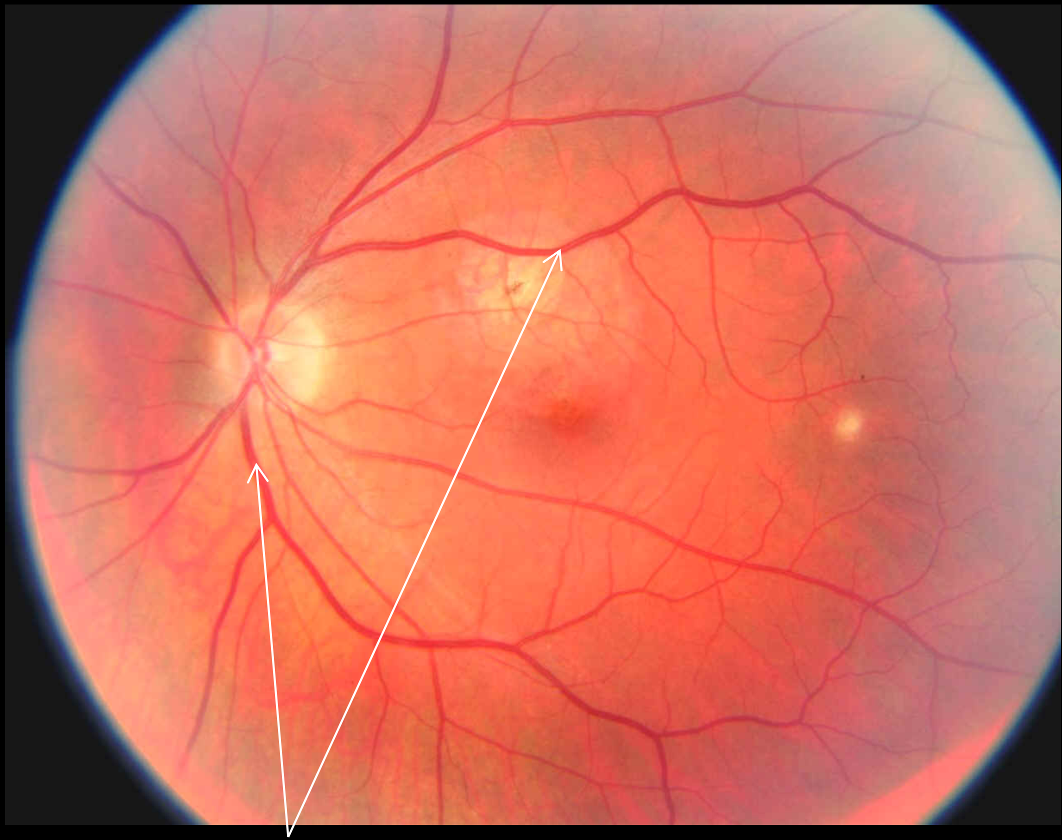
- Compute Hough transform and locate the maxima

The practical guide to the Hough Transform



- Draw the lines corresponding to the found maxima
- Here the cyan line is the longest

Path Tracing

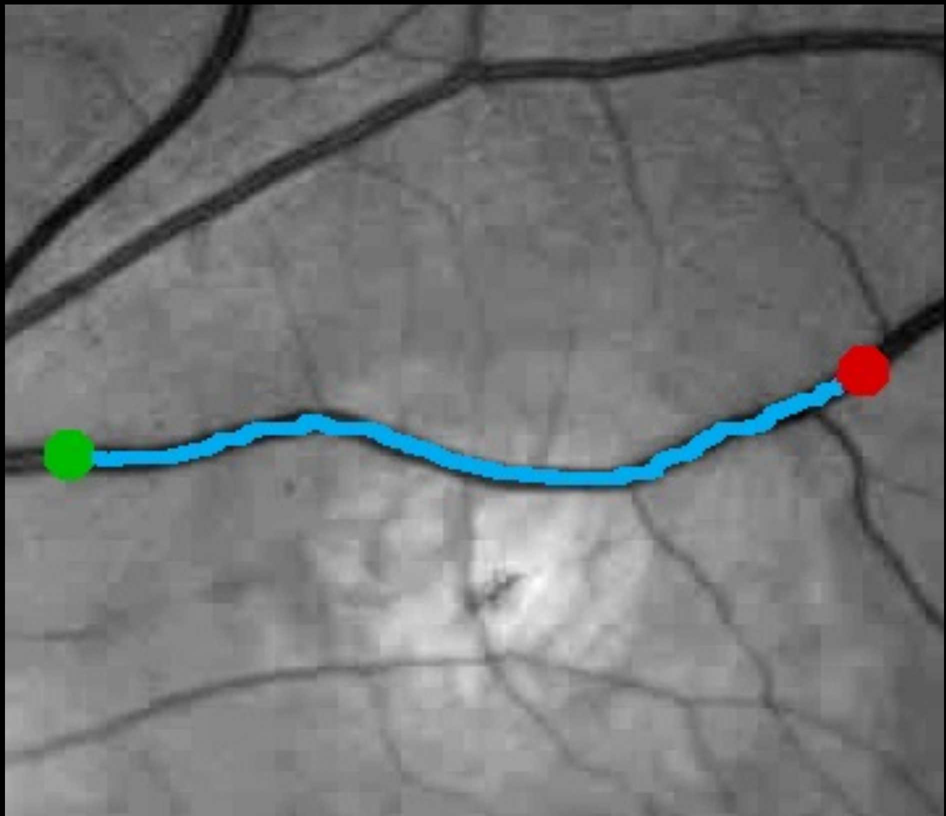


Arteries and veins

Fundus image

- The diameter as function of the distance to the optic cup tells something about the patients health
- We need to find the arteries and veins
- Path tracing is one solution

Path tracing



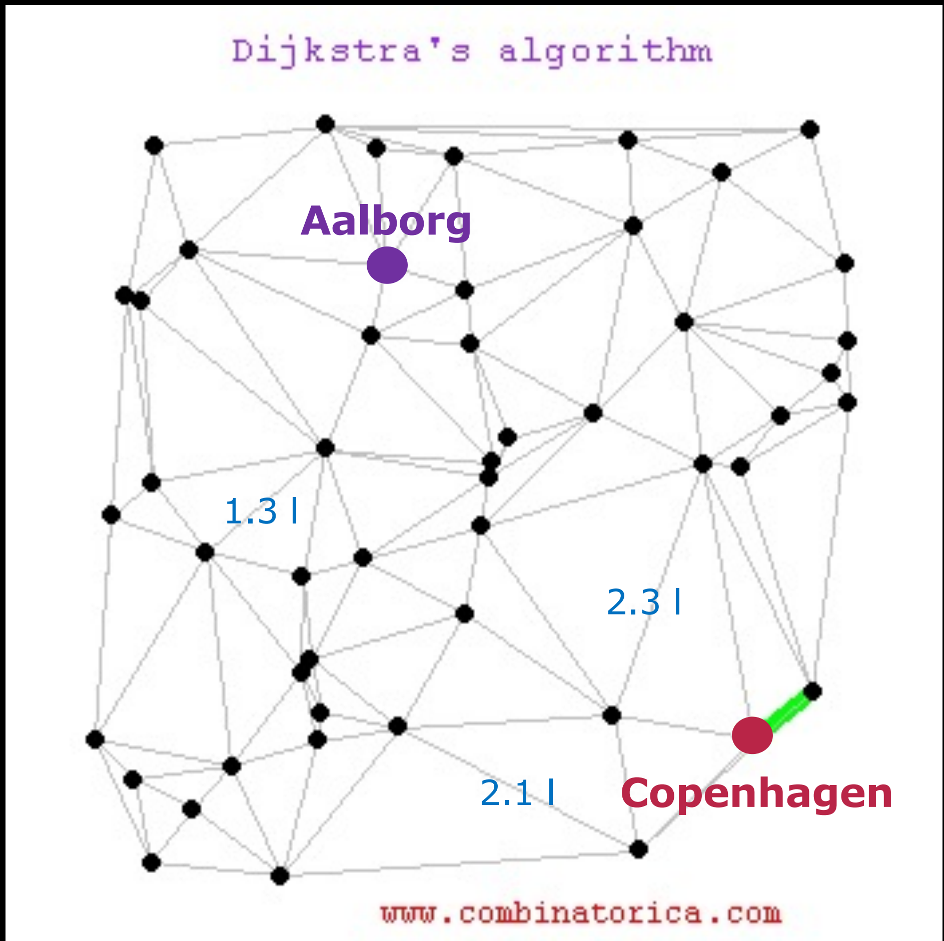
- A path is defined as a curve in an image defined as *something that is different from the background*
- In this case it is a dark line
- Pre-processing can for example turn edges into dark lines.

Dynamic Programming

- Break up large problem into many small sub-problems
- A classic algorithm:
  - Dijkstra’s algorithm
  - One source to all nodes shortest path
- We will look at a simplified variant

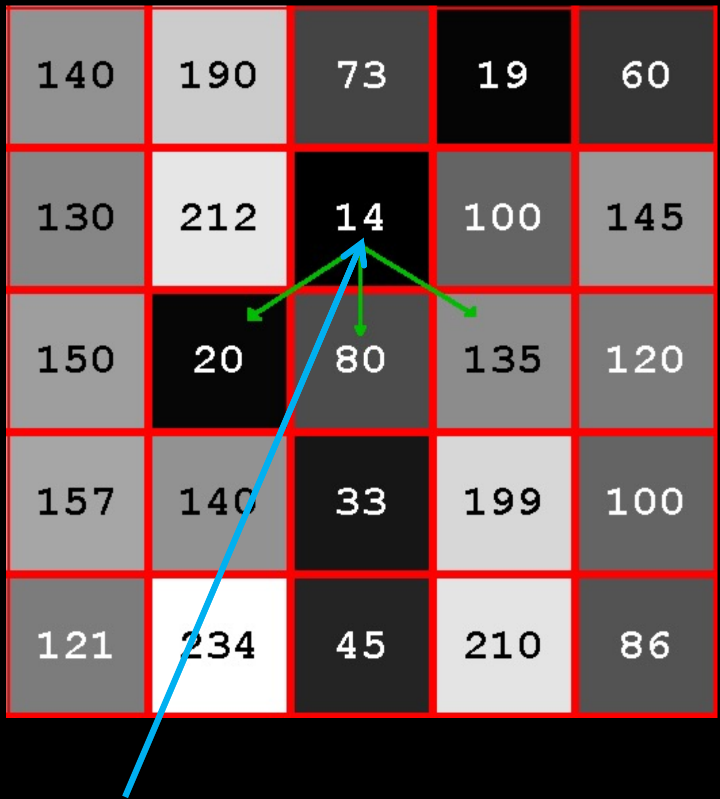
Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". Numerische Mathematik. 1: 269–271.

Path tracing



- A GPS device uses path tracing
- Based on *graph algorithms*
  - A city is a node
  - A road is an edge. The weight of the edge is the fuel cost
- How do we come from Copenhagen to Aalborg using the least fuel?
- Dijkstra's algorithm

Images as graphs



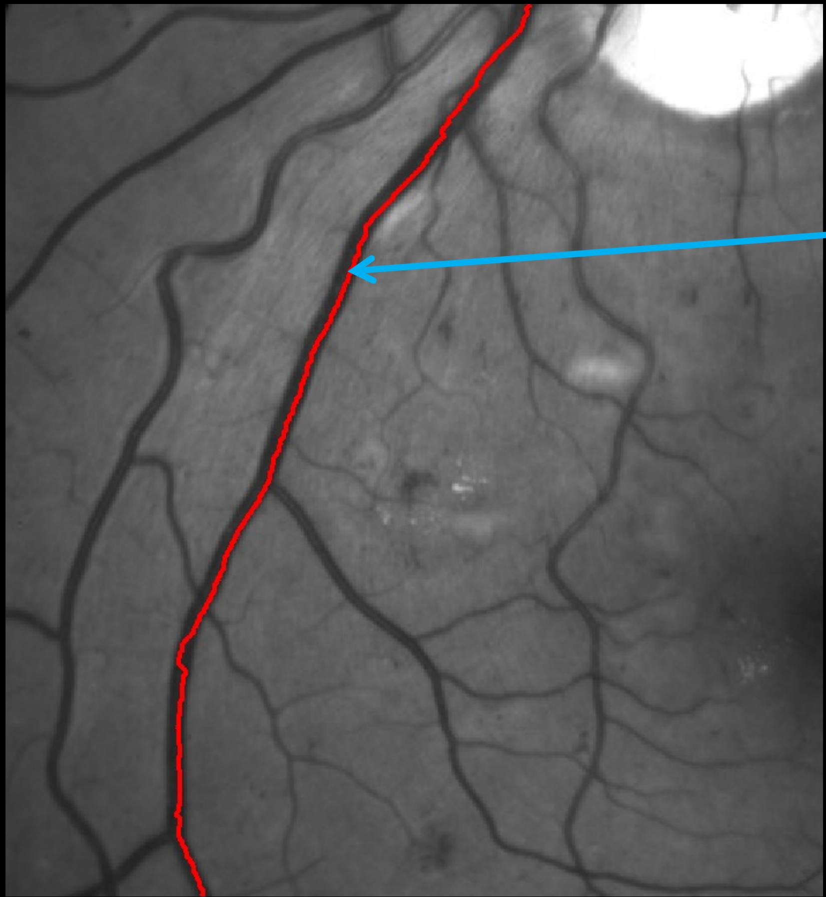
$$C(2,3) = 14$$

- Each pixel is a node
- Pixel neighbours are connected by edges
- The edge cost ($c(r,c)$)is the pixel value
- Directed graph
- Imagine a car driving on the image
- Called a *cost image*

Simplified problem

- Track dark lines
- Path going from top to bottom
- No sharp turns – smooth
- Problem:
  - from the top to the bottom
  - Sum of pixel values should be minimal

Simplified problem



- Pixel value at (r,c) equals the cost

  $C(r,c)$

- The path P consist of pixels
- The sum of pixel values in the path

$$C_{tot} = \sum_{(r,c)\in P} C(r,c)$$

Path cost

140	190	73	19	60
130	212	14	100	145
150	20	80	135	120
157	140	33	199	100
121	234	45	210	86

P = [(1,3), (2,3), (3, 2), (4,3), (5,4)]

- A path is defined as (r,c) coordinates

$$\mathcal{C}_{tot} = \sum_{(r,c)\in \mathcal{P}} \mathcal{C}(r,c)$$

Quiz 1: Total cost – what is C_{tot}?

A) 167

B) 350

C) 403

D) 270

E) 345

140	190	73	19	60
130	212	14	100	145
150	20	80	135	120
157	140	33	199	100
121	234	45	210	86

P = [(1,3), (2,3), (3, 2), (4,3), (5,4)]

Path cost

140	190	73	19	60
130	212	14	100	145
150	20	80	135	120
157	140	33	199	100
121	234	45	210	86

P = [(1,3), (2,3), (3, 2), (4,3), (5,4)]

- This is *NOT* the optimal path
- How do we compute the path P that has minimum C_{tot}?
- Test all possible paths?
  - No! Impossible amount of possibilities

Quiz 2: Path Cost

A path has been found in the image P=[(1,4),(2,4),(3,5),(4,5),(5,5),(6,4)]. A Matlab matrix coordinate system is used. What is the total cost of the path?

A) 196

B) 154

C) 201

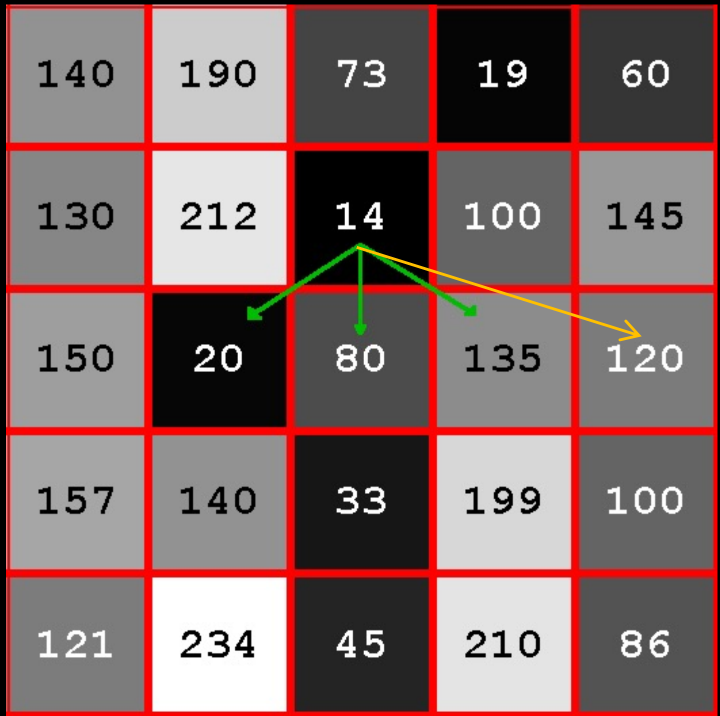
D) 185

E) 132

208	157	234	19	145	79
62	121	73	14	120	135
237	90	193	135	3	42
89	212	192	199	86	154
50	149	97	238	41	67
64	140	145	33	203	167

Figur 1: Grayscale billede

Path restriction: The rules



- Path is only allowed to
  - Go down
  - Move one pixel left or right
- Longer jumps not allowed

Accumulator image

140	190	73	19	60
270	285	33	119	164
420	53	113	168	239
210	193	86	312	268
314	320	131	296	354

- Keeps track of the accumulated cost for efficient paths finding
- Path ending here has cost 296

Computing the accumulator image

140	190	73	19	60
130	212	14	100	145
150	20	80	135	120
157	140	33	199	100
121	234	45	210	86

140	190	73	19	60
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Step 1: Copy first row of input image

Computing the accumulator image

140	190	73	19	60
130	212	14	100	145
150	20	80	135	120
157	140	33	199	100
121	234	45	210	86

140	190	73	19	60
270	285	33	119	164
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Step 2: Fill second row

$$\mathbf{A}(r,c) = \mathbf{I}(r,c) + \min\left(\mathbf{A}(r-1,c-1), \mathbf{A}(r-1,c), \mathbf{A}(r-1,c+1)\right)$$

Computing the accumulator image

140	190	73	19	60
130	212	14	100	145
150	20	80	135	120
157	140	33	199	100
121	234	45	210	86

140	190	73	19	60
270	285	33	119	164
420	53	113	168	239
210	193	86	312	268
314	320	131	296	354

Step 3: Fill all rows by looking at the previous row

$$\mathbf{A}(r,c) = \mathbf{I}(r,c) + \min\left(\mathbf{A}(r-1,c-1), \mathbf{A}(r-1,c), \mathbf{A}(r-1,c+1)\right)$$

Quiz 3: Accumulator Image

A) 57

B) 167

C) 301

D) 241

E) 145

An optimal path has been found in the image. What is the value of the accumulator image in the marked pixel?

117	163	74	210
223	244	171	57
132	61	110	170
241	172	17	215

Using the accumulator image

140	190	73	19	60
130	212	14	100	145
150	20	80	135	120
157	140	33	199	100
121	234	45	210	86

140	190	73	19	60
270	285	33	119	164
420	53	113	168	239
210	193	86	312	268
314	320	131	296	354

Step 4: The end of the optimal path can now be found

The backtracing image

140	190	73	19	60
270	285	33	119	164
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

0	0	0	0	0
1	3	4	4	4
1	3	3	3	4
2	2	2	3	4
2	3	3	3	5

- Keeps track of where the path *came* from
- Each pixel stores the column number

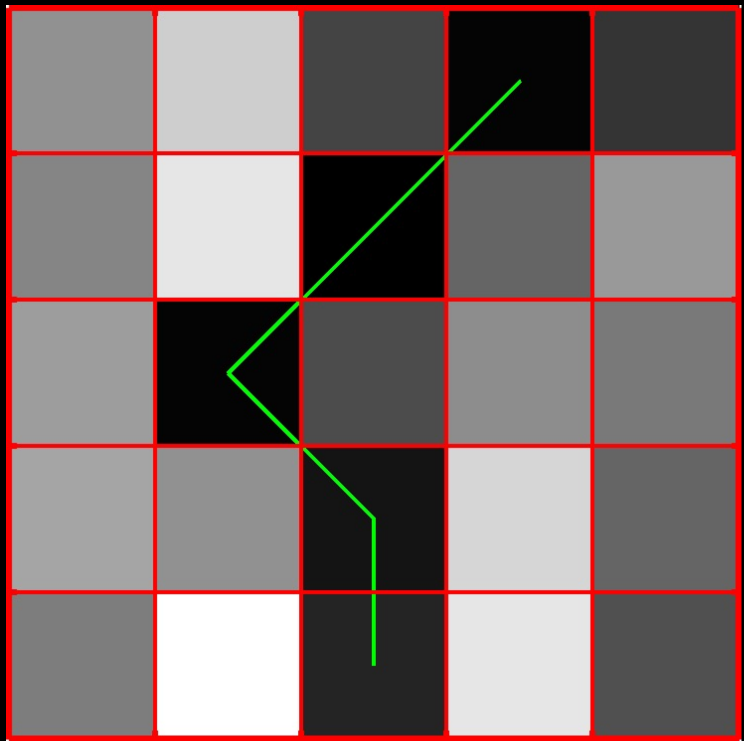
Using the backtracing image

140	190	73	19	60
270	285	33	119	164
420	53	113	168	239
210	193	86	312	268
314	320	131	296	354

0	0	0	0	0
1	3	4	4	4
1	3	3	3	4
2	2	2	3	4
2	3	3	3	5

Step 5: Trace the path in the backtracing image

Using the backtracing image



0	0	0	0	0
1	3	4	4	4
1	3	3	3	4
2	2	2	3	4
2	3	3	3	5

Quiz 4: Backtracing

A) 1

B) 2

C) 3

D) 4

E) 5

An optimal path has been found in an image. The backtracing image is seen below and the optimal path ends in the marked pixel. A Matlab coordinate system is used. What is the optimal path?

0	0	0	0	0
1	3	3	3	5
2	2	2	4	4
1	1	4	5	5
1	1	4	4	4

1. $\mathcal{P} = [(1,3),(2,2),(3,1),(4,1),(5,2)]$
2. $\mathcal{P} = [(1,3),(2,2),(3,2),(4,2),(5,2)]$
3. $\mathcal{P} = [(1,2),(2,2),(3,2),(4,1),(5,2)]$
4. $\mathcal{P} = [(1,3),(2,1),(3,1),(4,1),(5,2)]$
5. $\mathcal{P} = [(1,2),(2,1),(3,1),(4,2),(5,2)]$

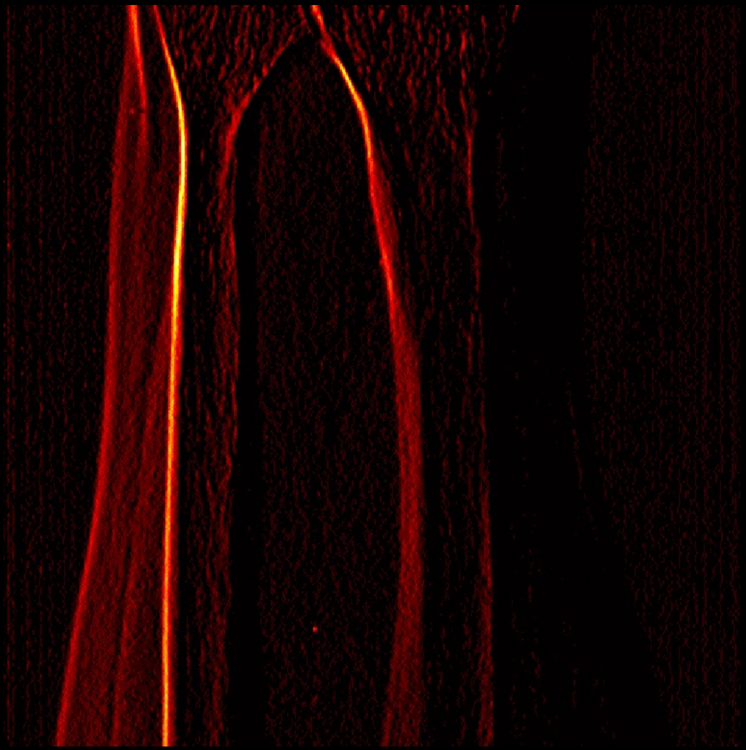
Pre-processing

- We would like to track paths that are not dark curves

Quiz 5 : X-ray preprocessing

A) Gaussian smoothing
B) 255-I
C) Gradient filter
D) Registration
E) Morphological operation

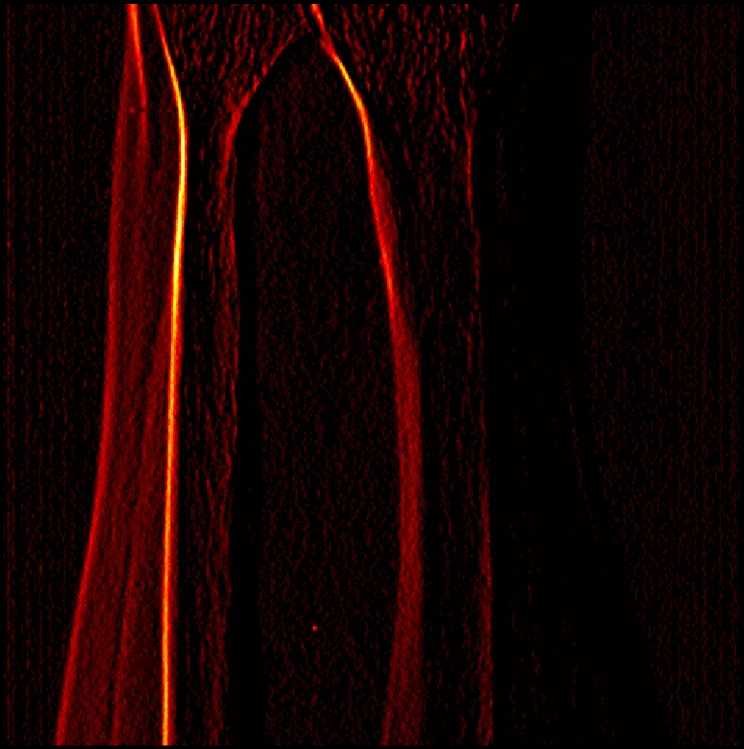
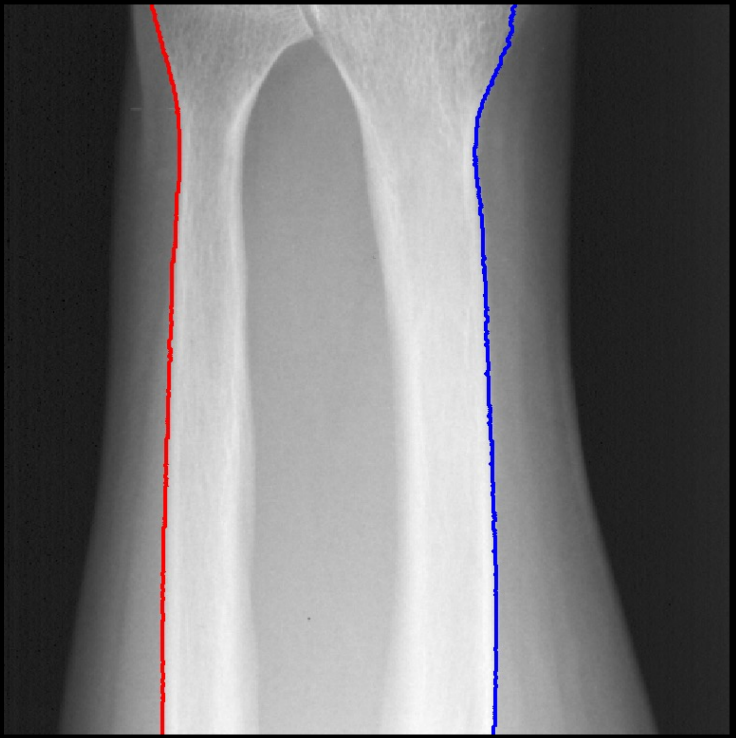
Pre-processing



Edge filtered image

(Gaussian smoothing followed by Prewitt)

Path tracing on pre-processed image

Paths found on pre-processed image and intensity inverted

Quiz 6: Optimal Path 2

A) 81

B) 64

C) 11

D) 73

E) 51

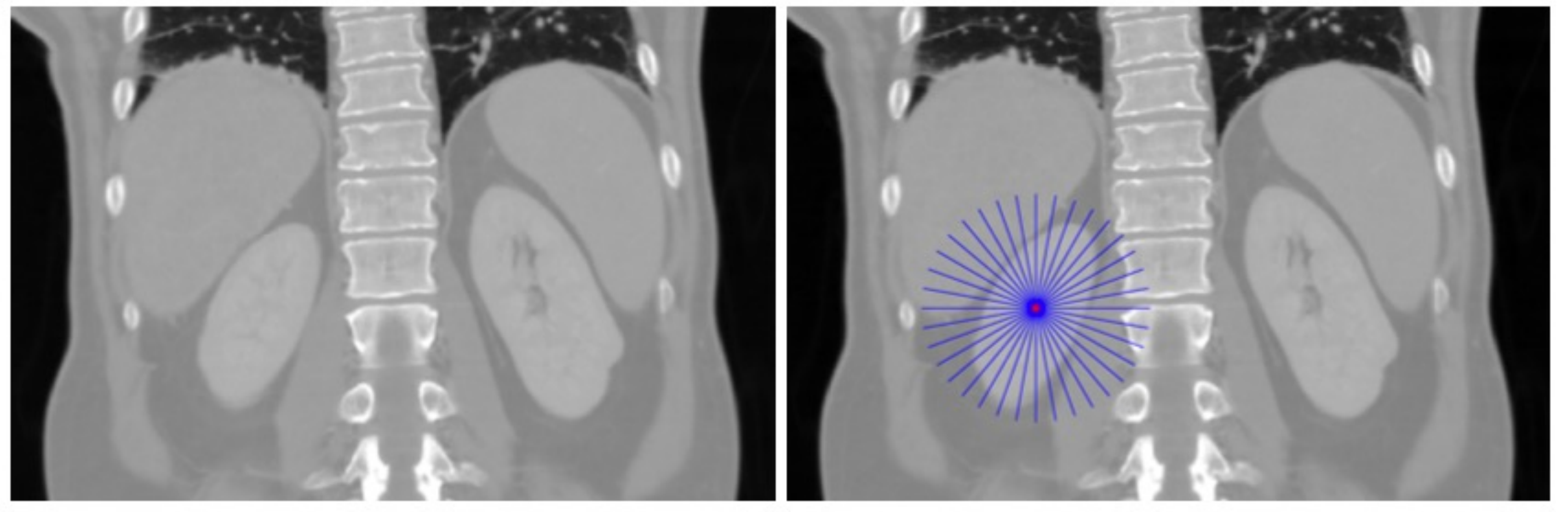
A 5 x 5 image is filled with values given the gray level run length encoding: 2, 180, 1, 15, 3, 112, 1, 8, 4, 177, 1, 20, 4, 195, 1, 12, 3, 242, 2, 25, 3, 9. After that the optimal path is found. What is the total cost?

Solution:

15+8+20+12+9=64

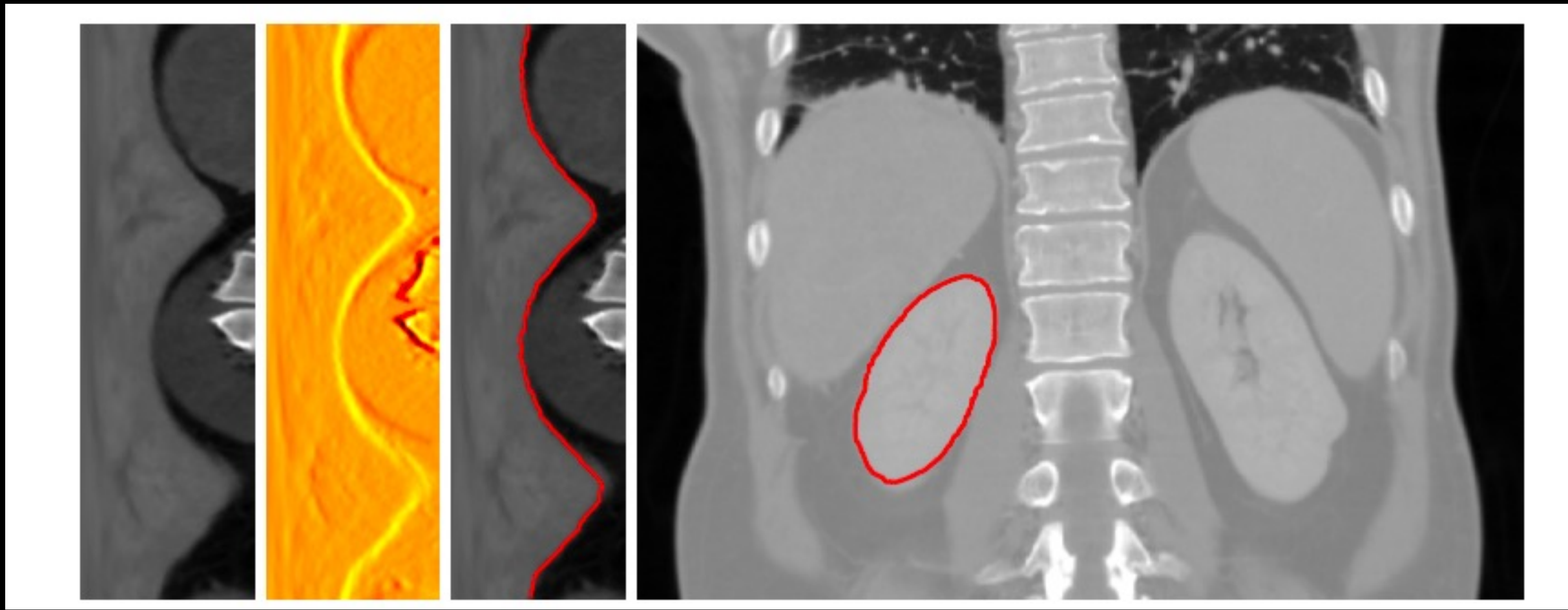
180	180	15	112	112
112	8	177	177	177
177	20	195	195	195
195	12	242	242	242
25	25	9	9	9

Locating Circular Structures



- Define origin inside structure
- Send out spokes

Locating Circular Structures

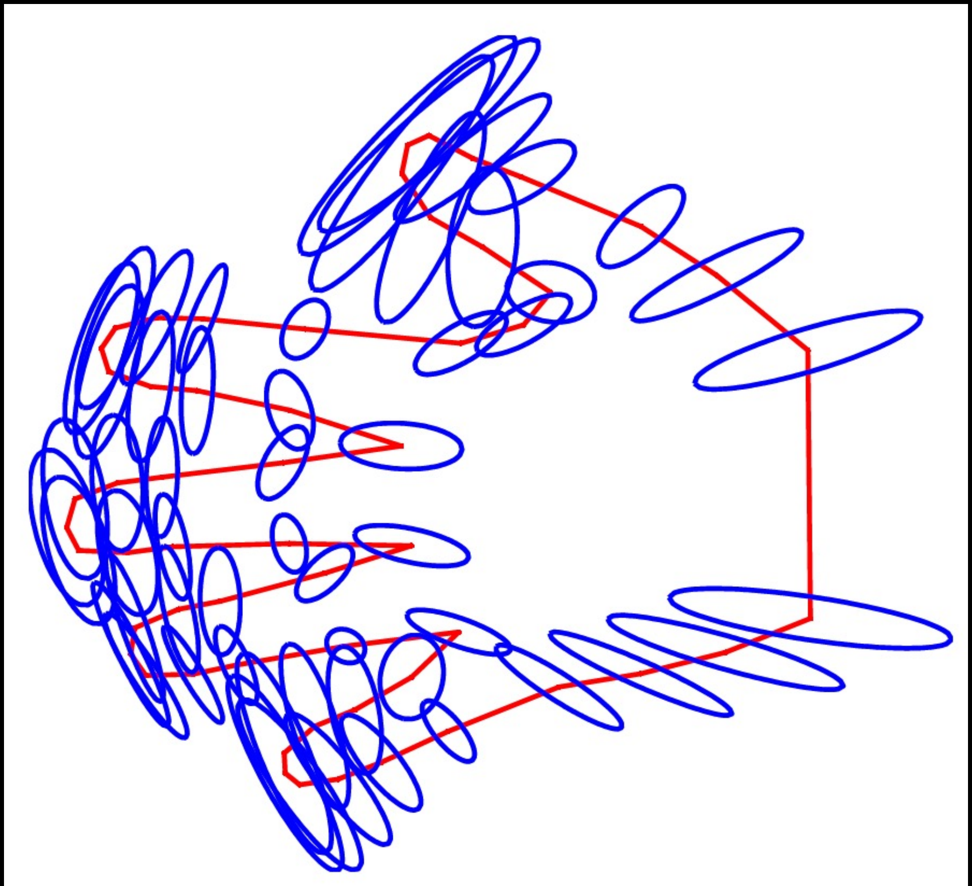


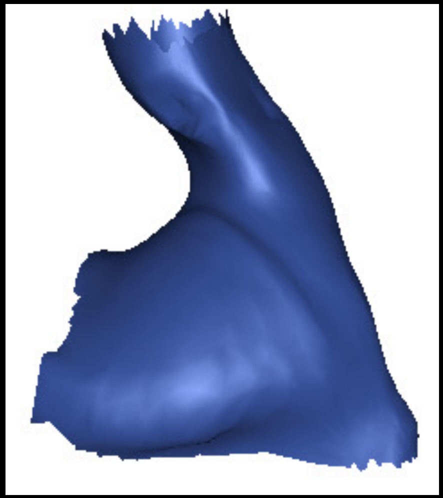
- Each spoke is a line in a new image (surface- layer detection)
- Prewitt
- Dijkstra's algorithm
- Map back the spokes into image

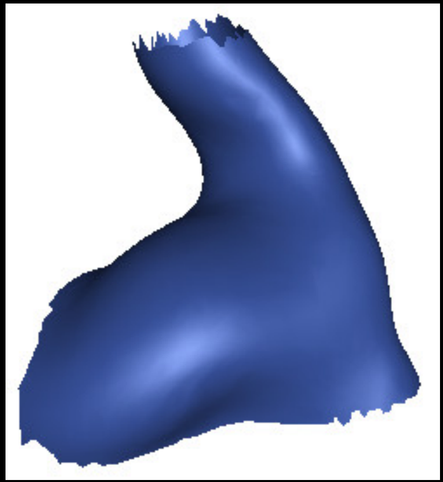
What did you learn today?

- Use the Hough transform for line detection
- Describe the slope-intercept, the general form and the normalised form of lines
- Describe the connection between lines and the Hough space
- Use edge detection to enhance images for use with the Hough transform
- Use dynamic programming to trace paths in images
- Describe how an image can be used as a graph
- Describe the fundamental properties of a cost image
- Compute the cost of path
- Compute an accumulator image for path tracing
- Compute a back tracing image for path tracing
- Choose appropriate pre-processing steps for path tracing
- Describe how circular structures can be located using path tracing

Lecture 9 – Statistical models of shape and appearance







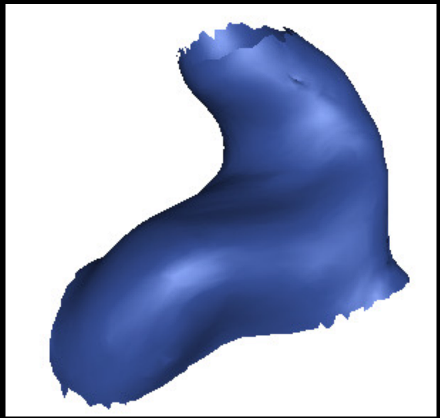


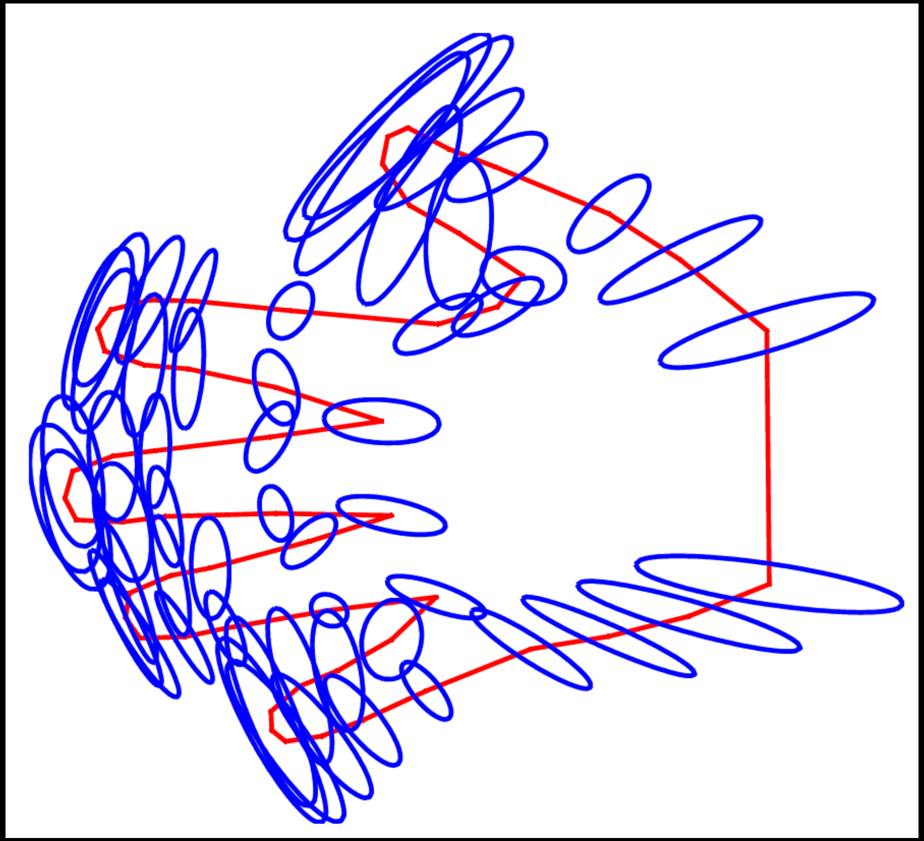
Image Analysis

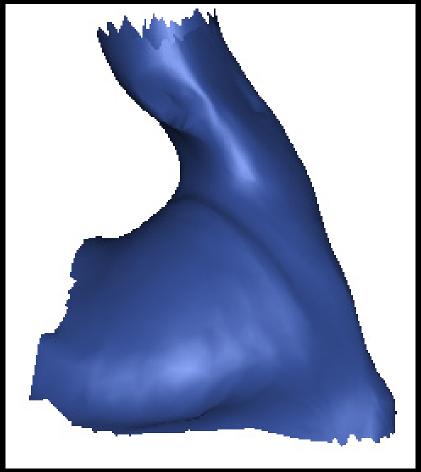
Rasmus R. Paulsen
Tim B. Dyrby
DTU Compute

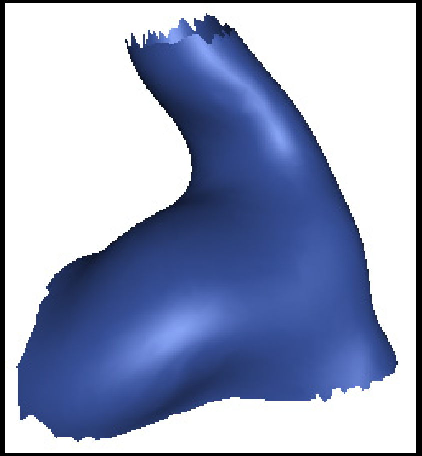
rapa@dtu.dk

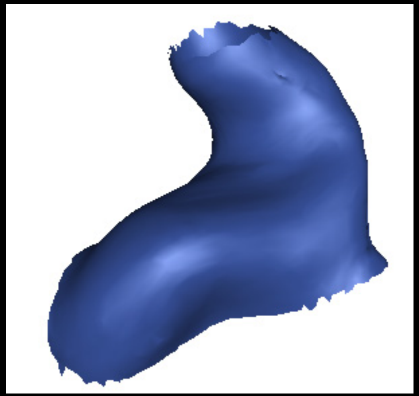
http://courses.compute.dtu.dk/02502

Lecture 9 – Statistical models of shape and appearance









Today's Learning Objectives

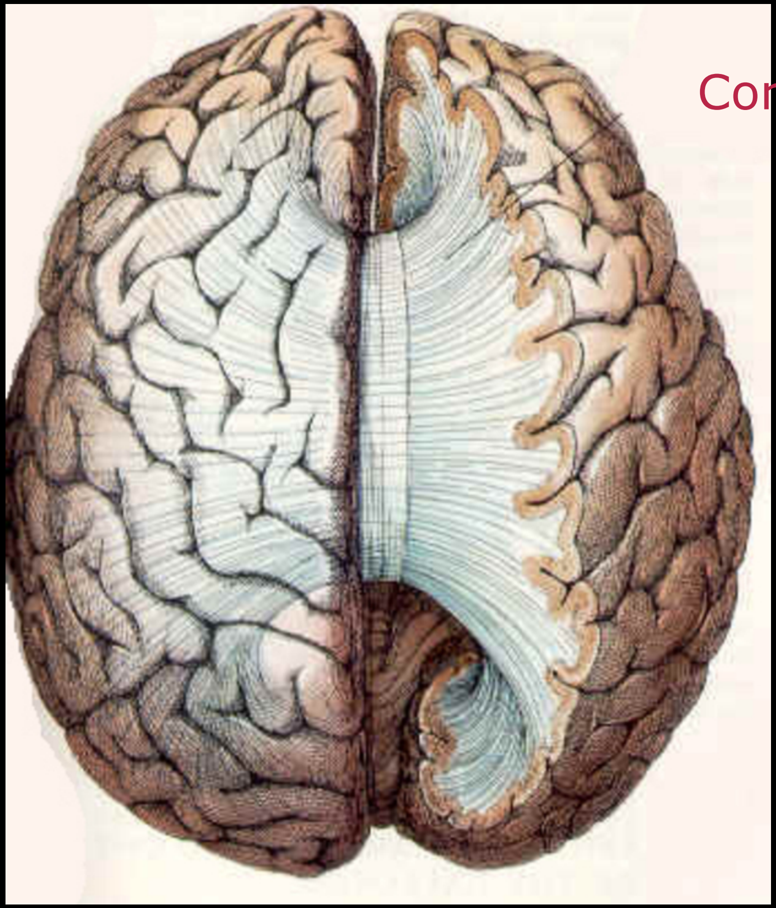
- Describe the concept of shape models
- Define the shape of an object using landmarks
- Describe point correspondence
- Describe and use the vector representation of a shape
- Describe how a shape can be seen as a point in high-dimensional space
- Explain how shapes can be aligned
- Describe how principal component analysis can be used to model shape variation
- Explain the similarity between Eigenfaces and shape and appearance models

A typical scenario

- Doctor X believes that he can “see” on a hand X-ray if the patient is in risk of arthritis!
- Specifically Doctor X is sure that the *shape of the joints* is an estimator for arthritis!

Can we verify that?

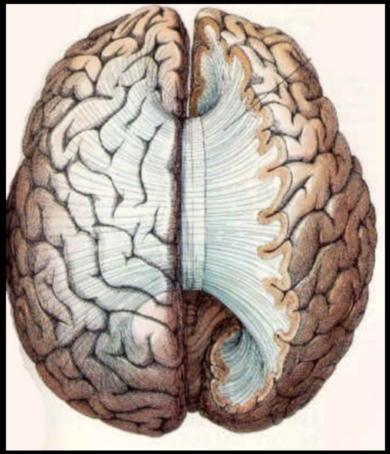
Scenario II



Corpus Callosum

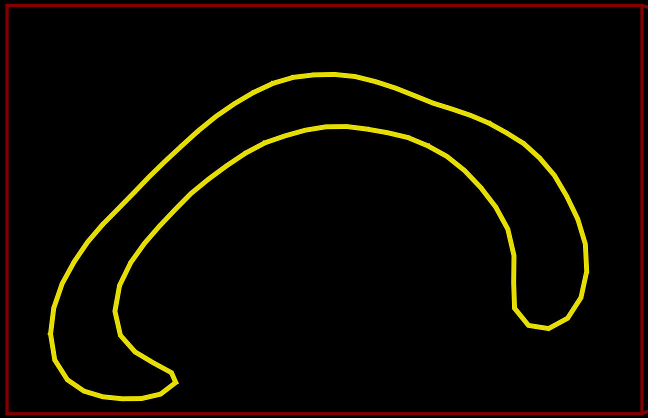
- MR images have been captured of a large group of people
- Cognitive abilities measured as well
- Is there a correlation between *how the brain looks* and how we behave?
- Does the shape of corpus callosum tell us something?

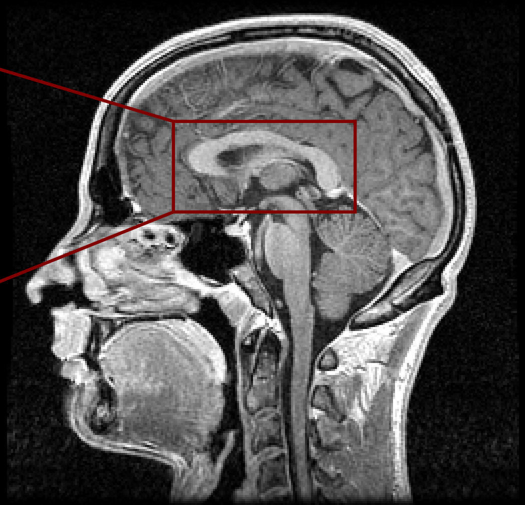
Scenario II



Corpus Callosum

- We can get the MR slice with the corpus callosum from all the patients



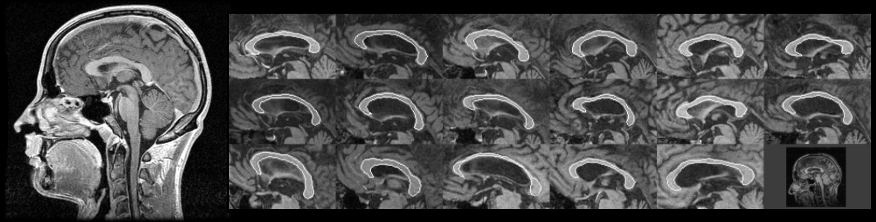


Scenario III

- An experienced hearing aid fitter has seen a lot of ears!
- Some hearing aid users are very difficult to fit. Why?
- Large variation in the shape of ears
- Ear canals change shape when people chews
- Is it possible to learn about the shape and use it?

Shape Analysis

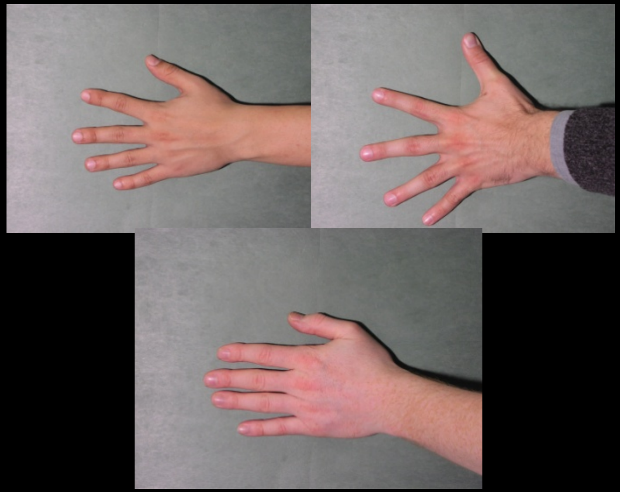
- What can we learn from shape?
- What can we use it for?
- How do we do it?



600 MR scans and behavioural data

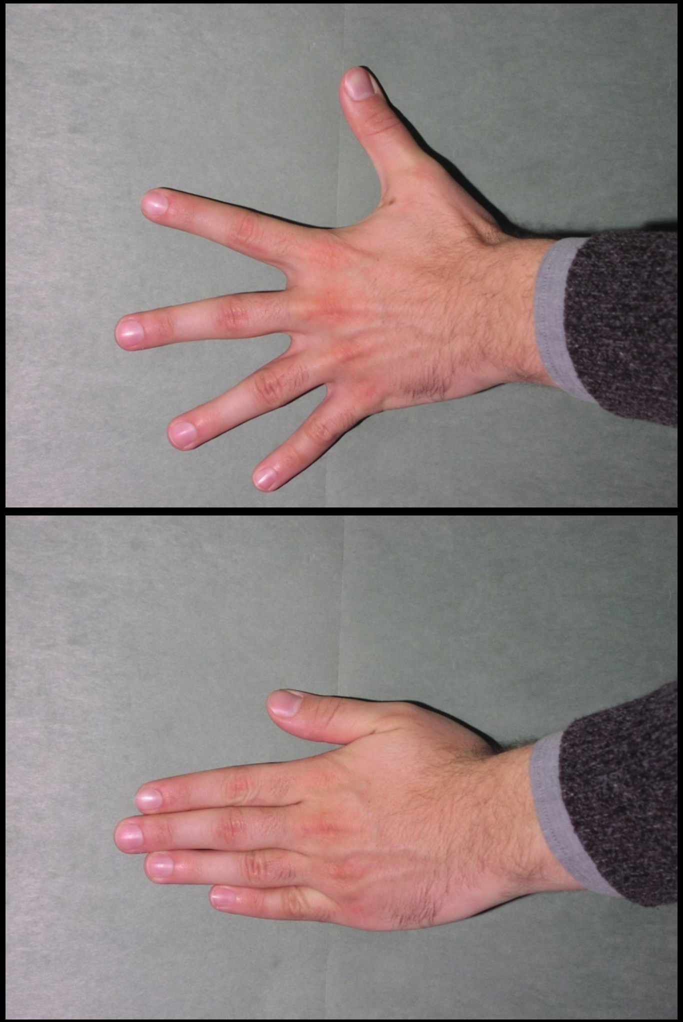
1000 historical X-rays

A boxful of something that look like ear canals



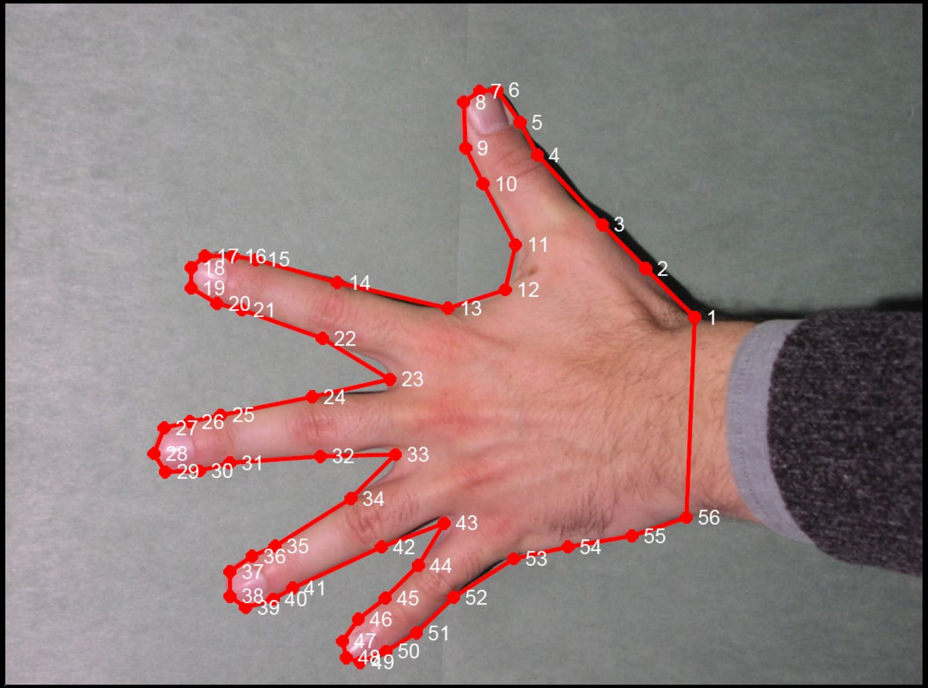
A set of hand photographs

What is shape?



- How do we define the *shape* of this hand?
- What is the shape difference between the two hands?

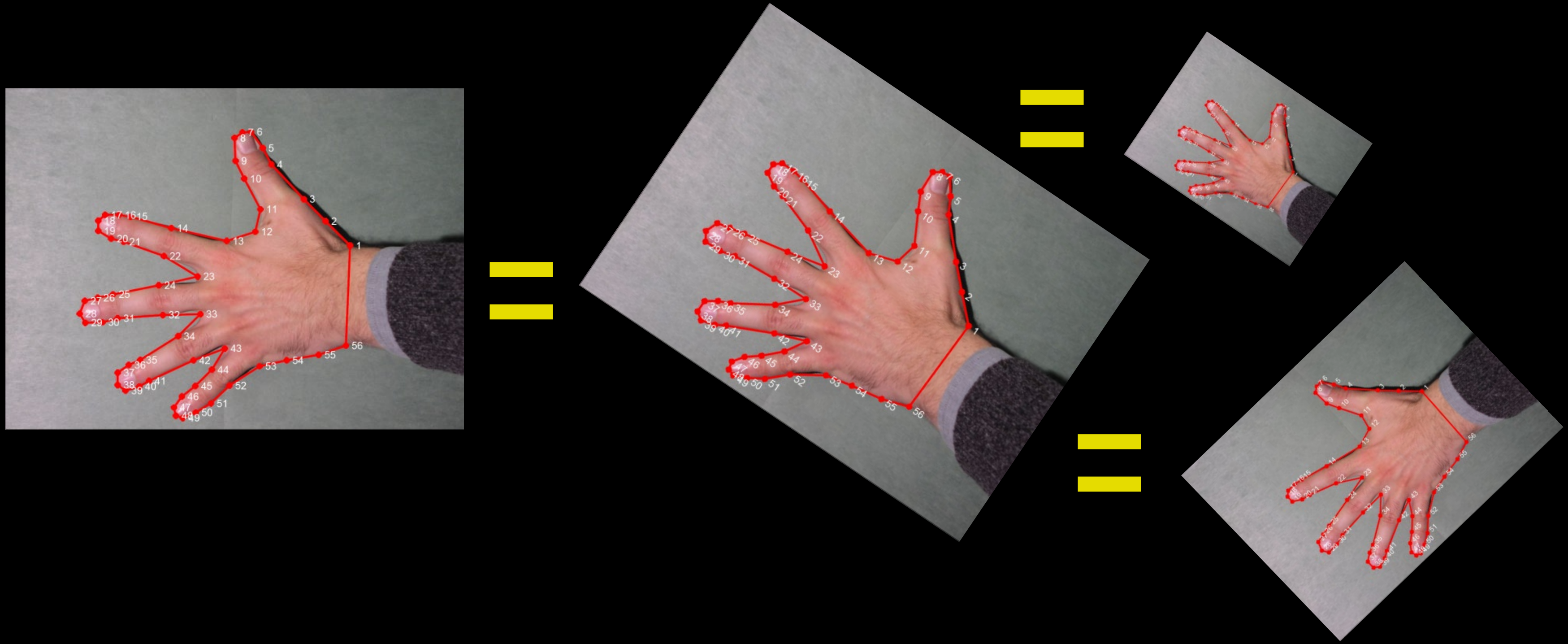
Shape definition



- Shape is defined using landmarks
  - Placed by an expert
- In this case the outer contour of the hand
- Just one of many ways!

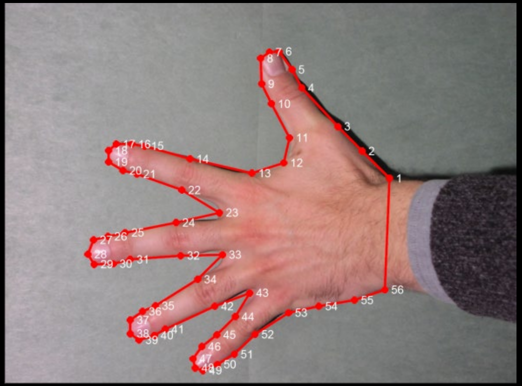
Shape definition

Shape is all geometrical information that remains when location, scale, and rotational effects are removed

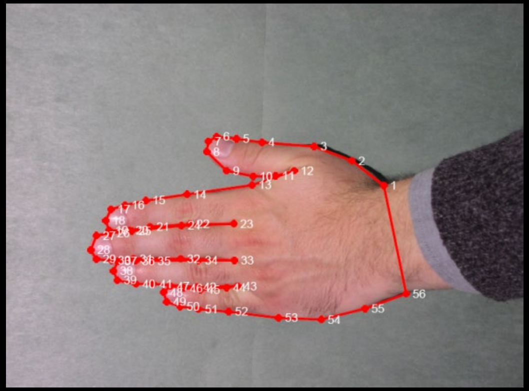


Shape definition

Shape is all geometrical information that remains when location, scale, and rotational effects are removed

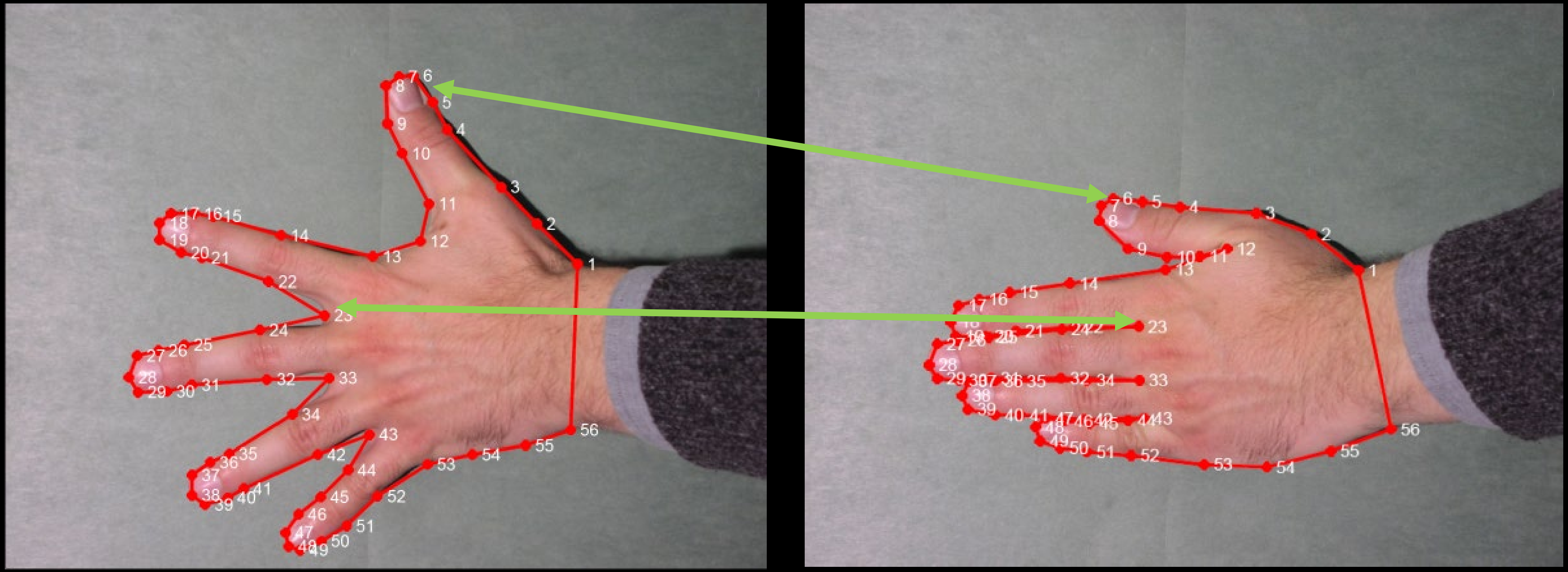


≠



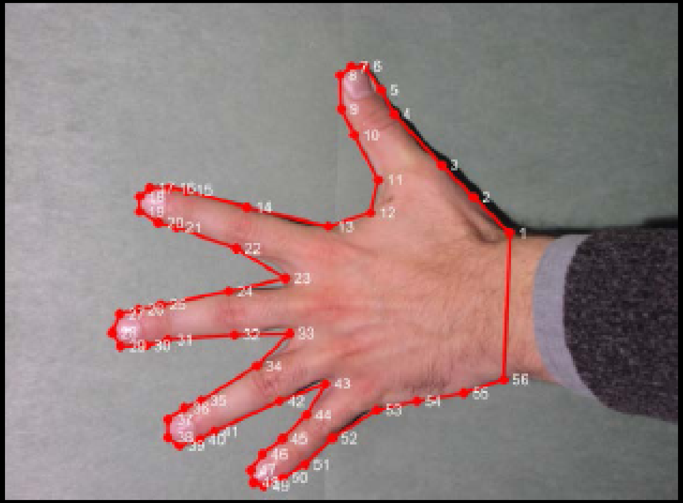
Landmarks and point correspondence

Landmarks are placed on the same place on all shapes in the training set



Shape as a vector

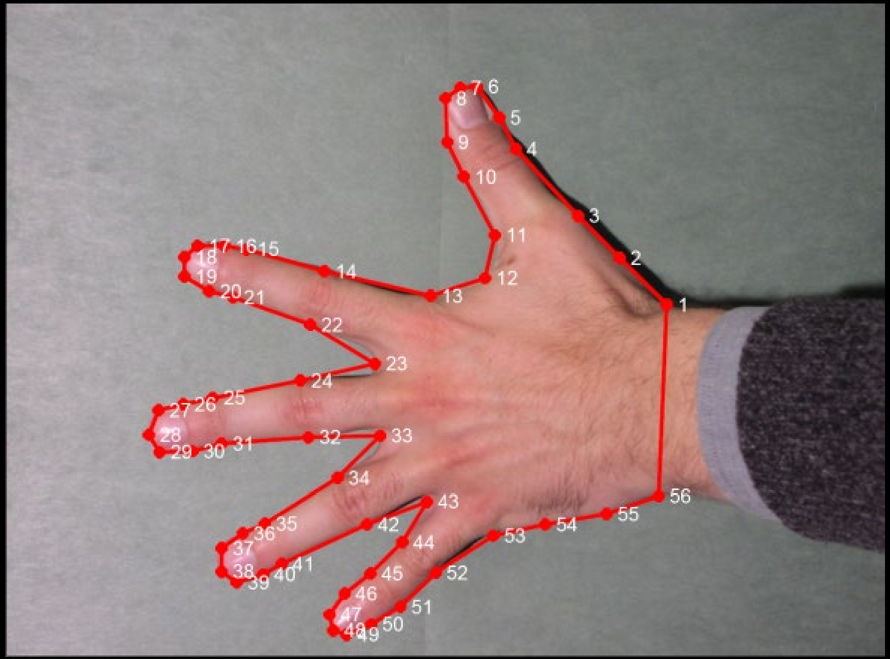
$$1 : (x_1, y_1)$$
$$2 : (x_2, y_2)$$
$$\vdots$$
$$N : (x_n, y_n)$$



- The shape is represented as an array of (x,y) coordinates
- Trick number one!
  All coordinates are put into one vector!
- n=56 points
  - Vector with 112 elements!

$$\mathbf{x} = [x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n]^T$$

Shapes in high-dimensional space

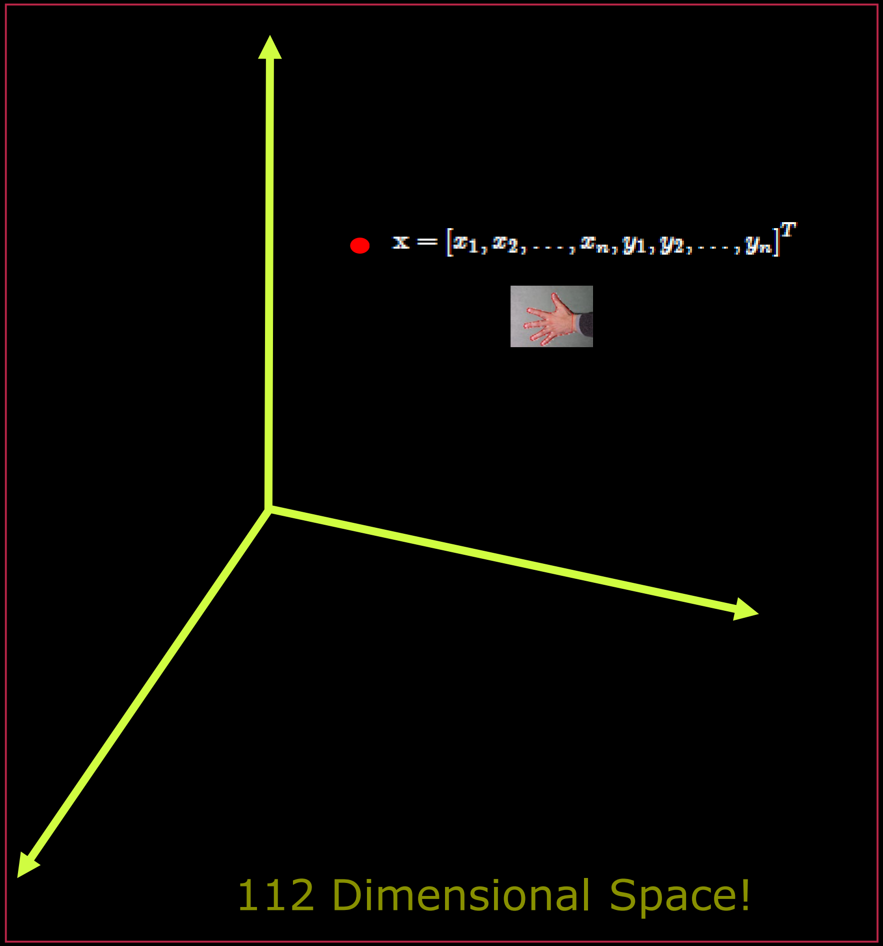


$$\mathbf{x} = [x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n]^T$$

- One hand is now described using one vector
- A vector can also be seen as a point in space!

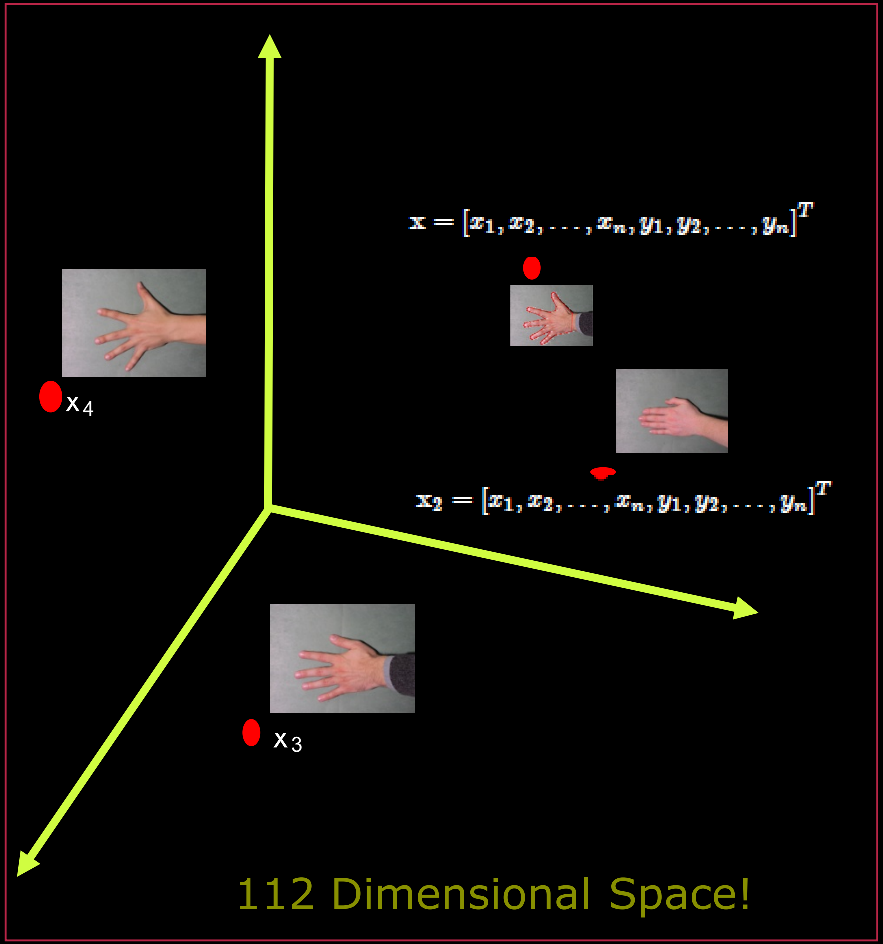
Trick number two!

Coordinates in space



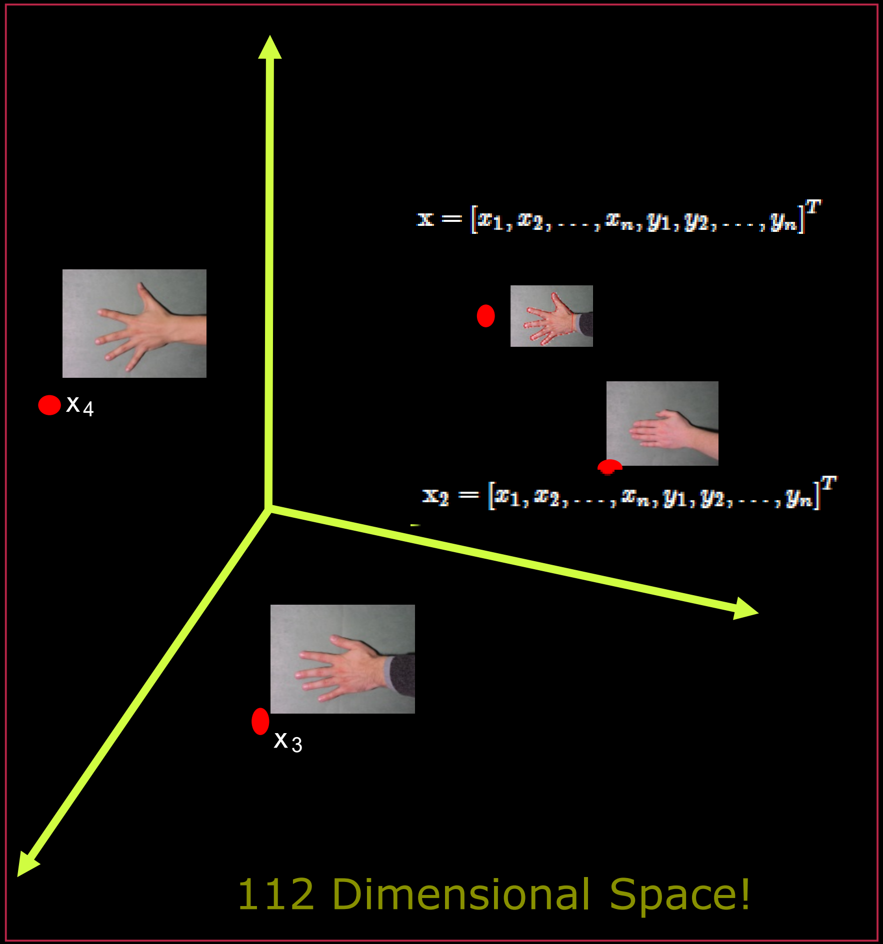
- On hand is now described using one vector
- A vector can also be seen as a coordinate in space!
- Not 2D space, not 3D space, not 4D space…
- 112 Dimensional Space!
- A hand has a position in this space!

Hands in Space



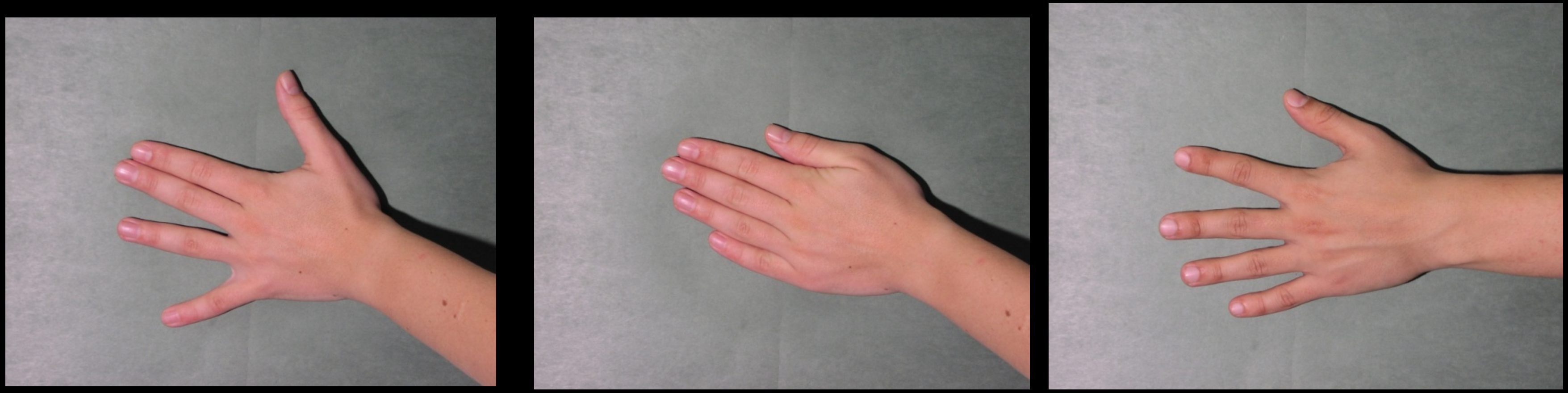
- A hand has a position in space!
- Another hand appears
  - in the same space
  - different position = different shape
- All hands have a place in this space!

Shape Analysis



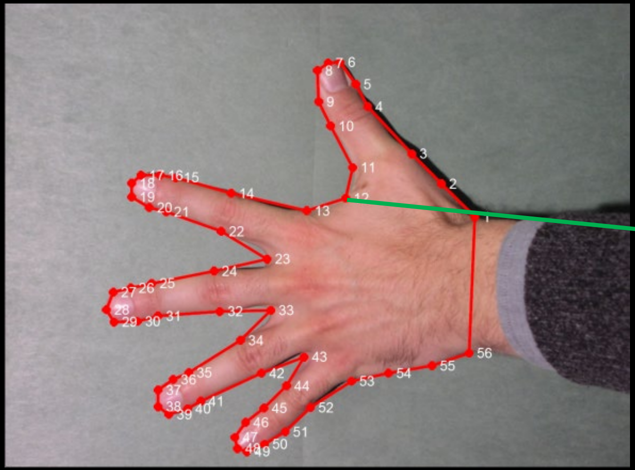
- Shape analysis
  - Similar shapes are placed on "planes" in the shape-space
  - Also called a manifold

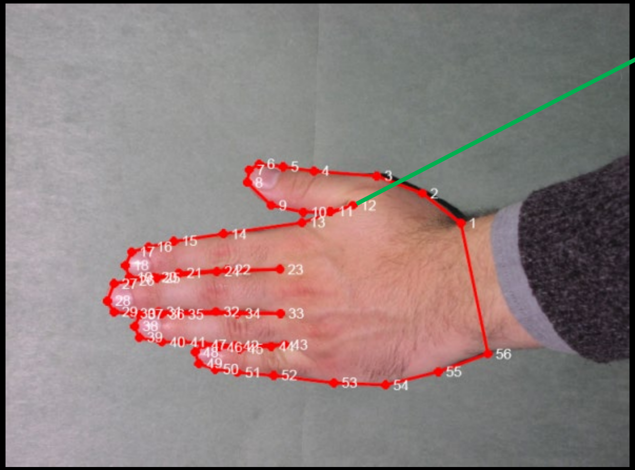
Shape alignment

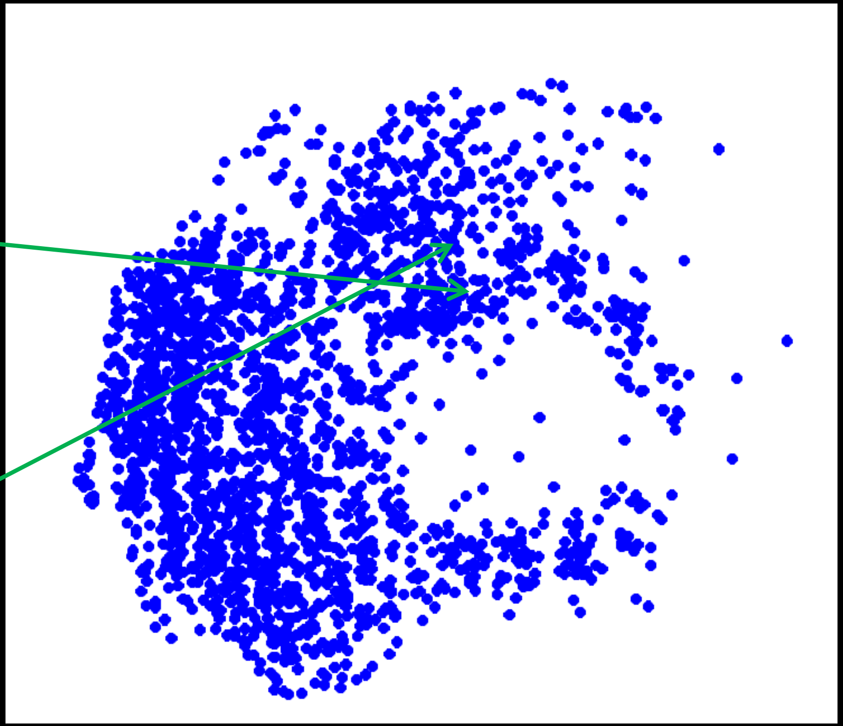


- 40 training images of hands
- 56 landmarks on each
- Placed in random location (translation+rotation)

Shape alignment



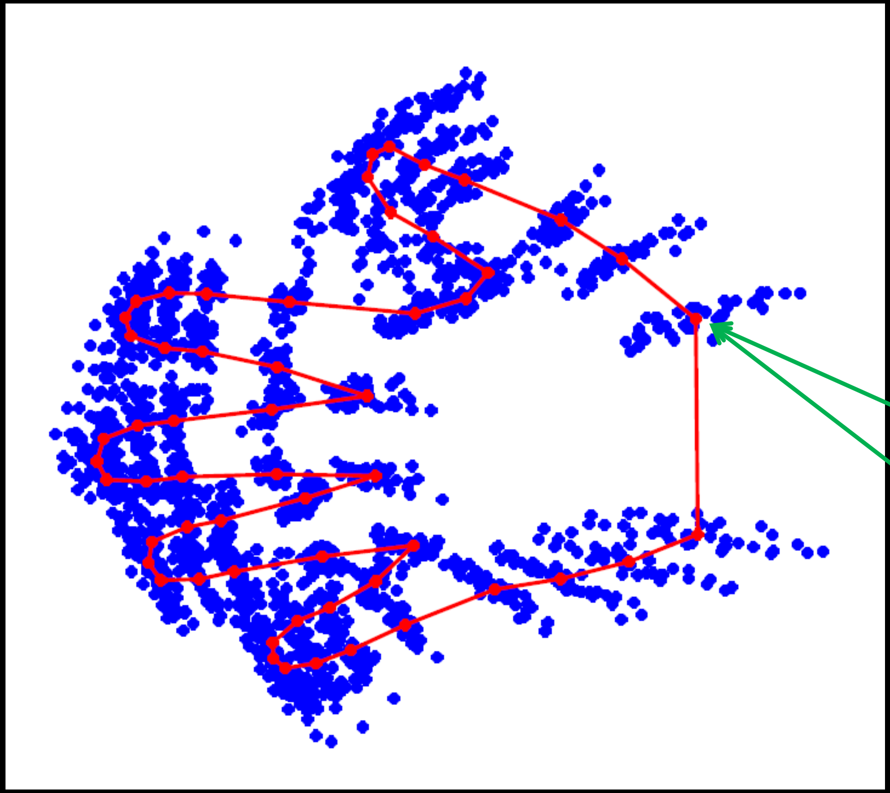




Landmarks from all hands

Needs alignment!

What is alignment?



Average shape

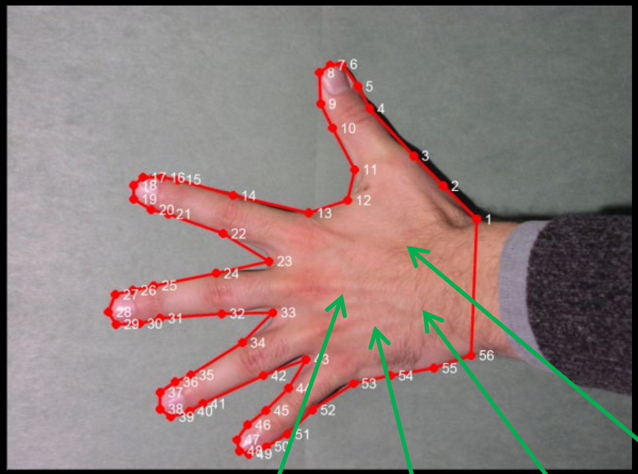
- Group wise registration
  - Not one-to-one
  - All to the average shape

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$$

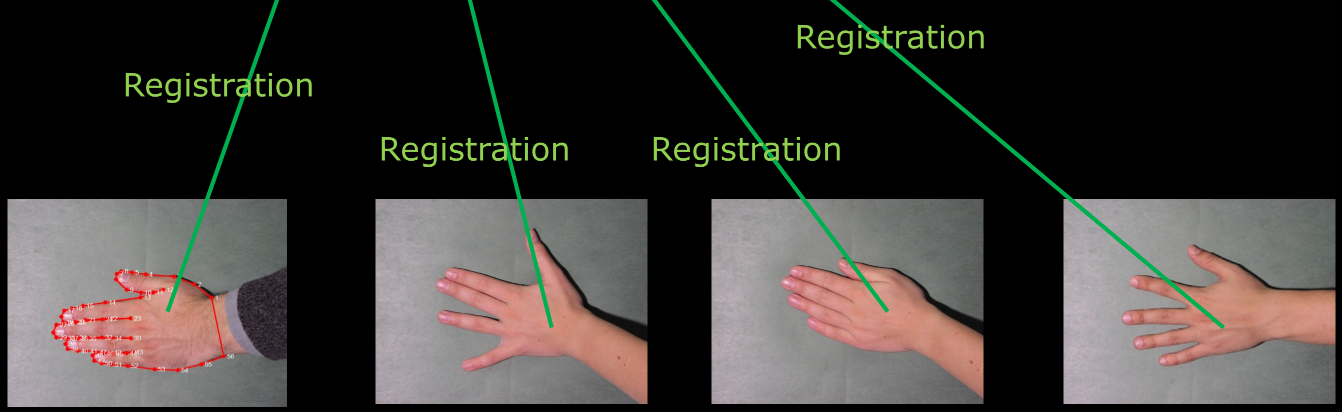
$$\bar{\mathbf{x}} = [\bar{x_1}, \bar{x_2}, \ldots, \bar{x_n}, \bar{y_1}, \bar{y_2}, \ldots, \bar{y_n}]^T$$

But hey! We do not have an average shape?

Procrustes Analysis (alignment)



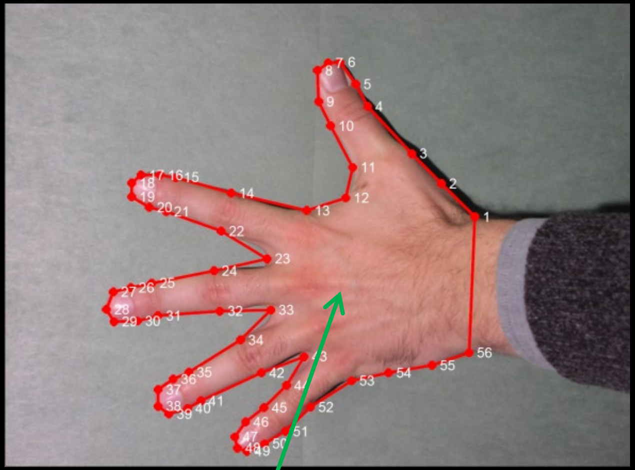
"Average shape"



Shape #2

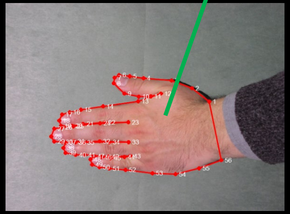
- We start by defining
  - Average shape = Shape #1
- Align shape #2 to shape #1
- Align all shapes to shape #1

Landmark based registration



"Average shape"

Registration

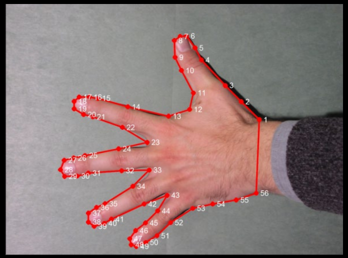


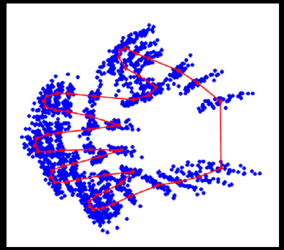
Shape #2

- Shape #2 is transformed to fit the average shape
  - Translation
  - Rotation
  - Scaling
  - = Similarity Transform
- Result
  - Shape #2 is placed *on top of* the average shape

Procrustes Analysis

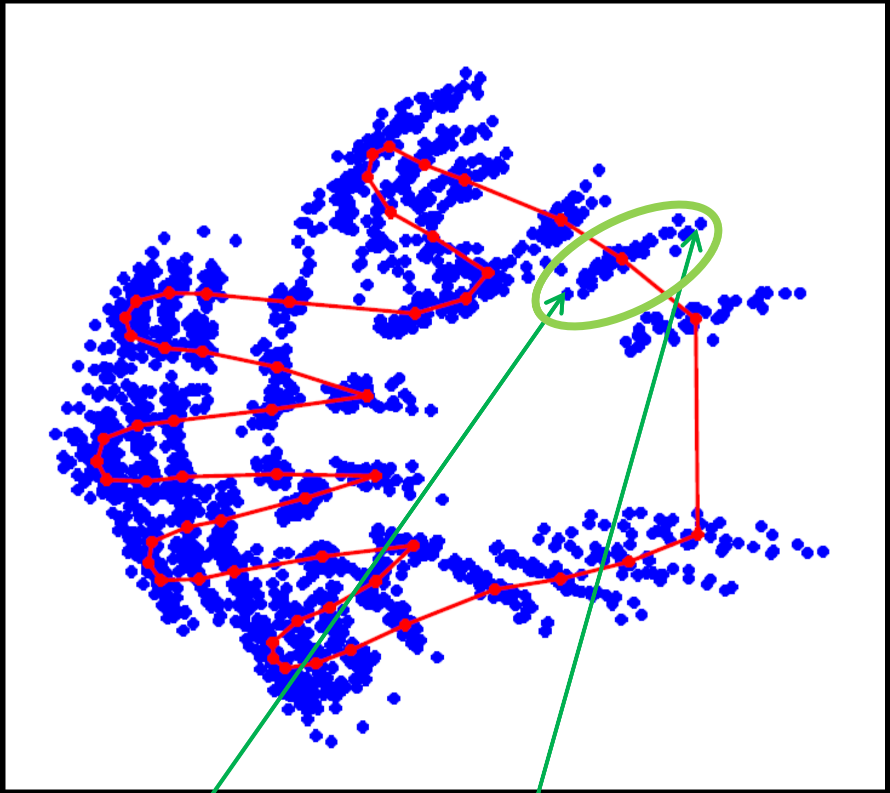
1. Average shape is set to shape #1
2. Register all shapes to the average shape
   – Landmark based registration
3. Recompute the average shape
4. If average shape changed return to step 2.





$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$$

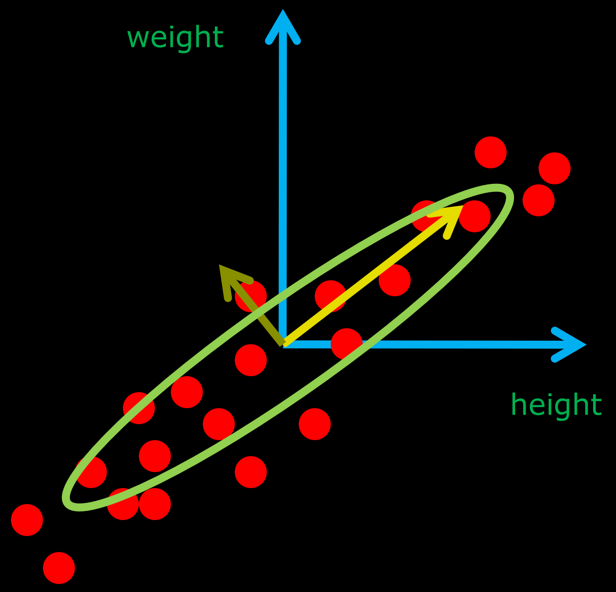
Aligned shapes – what now



Shape #16   Shape #27

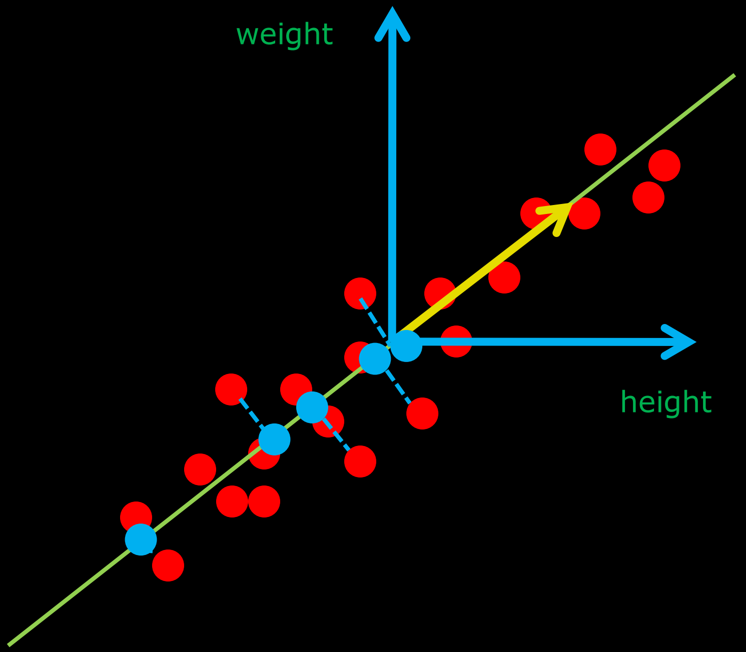
- Individual landmark variation
  - Over the training set
- What shape is the variation?

Principal Component Analysis (PCA)



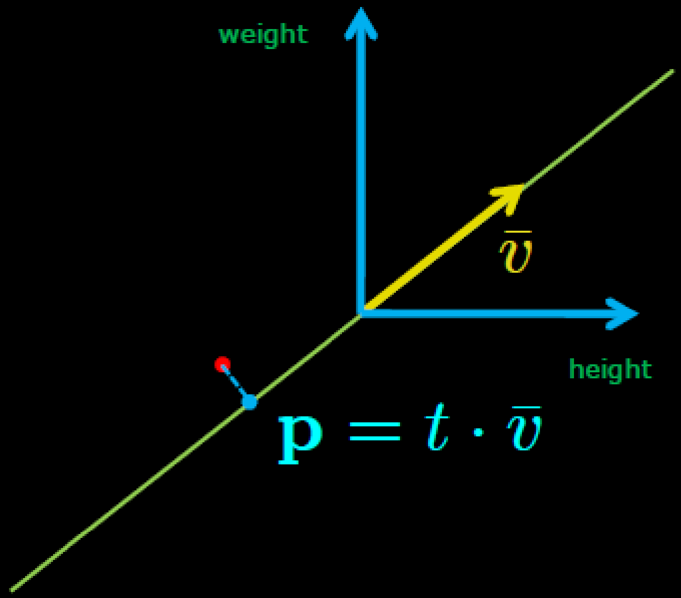
- PCA
  - Main axis in data
  - Eigenvectors
  - Eigenvalues
- Size of Eigenvalues describe explained variance

Principal Component Analysis (PCA)



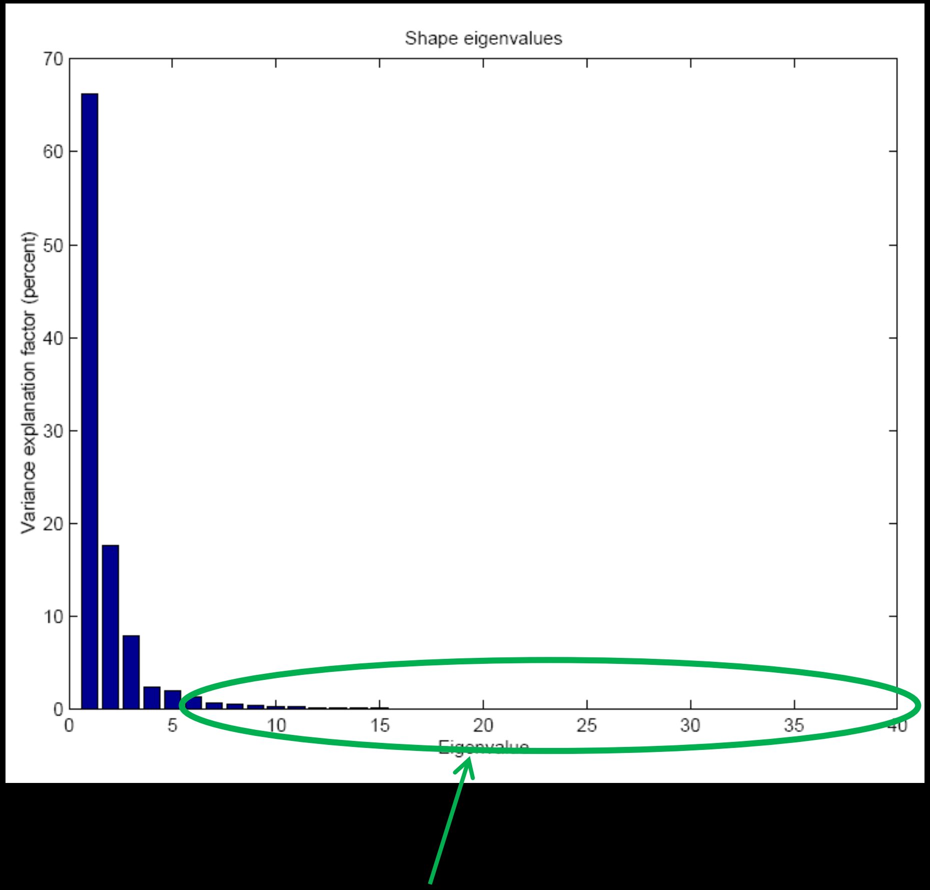
- We throw away the *noise dimensions*
- Points projected to the line

Principal Component Analysis (PCA)



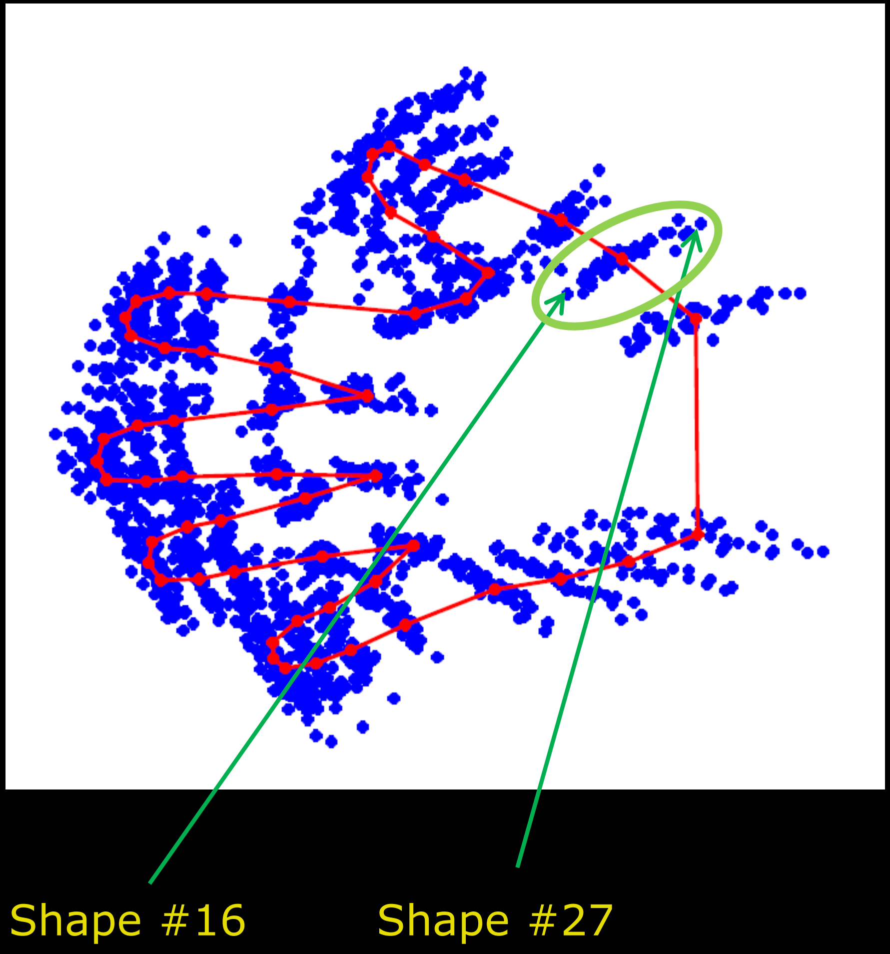
- We throw away the *noise dimensions*
- Points projected to the line
- A point can now be described by one parameter *t*
- We have reduced the number of dimensions

How many dimensions should we keep?



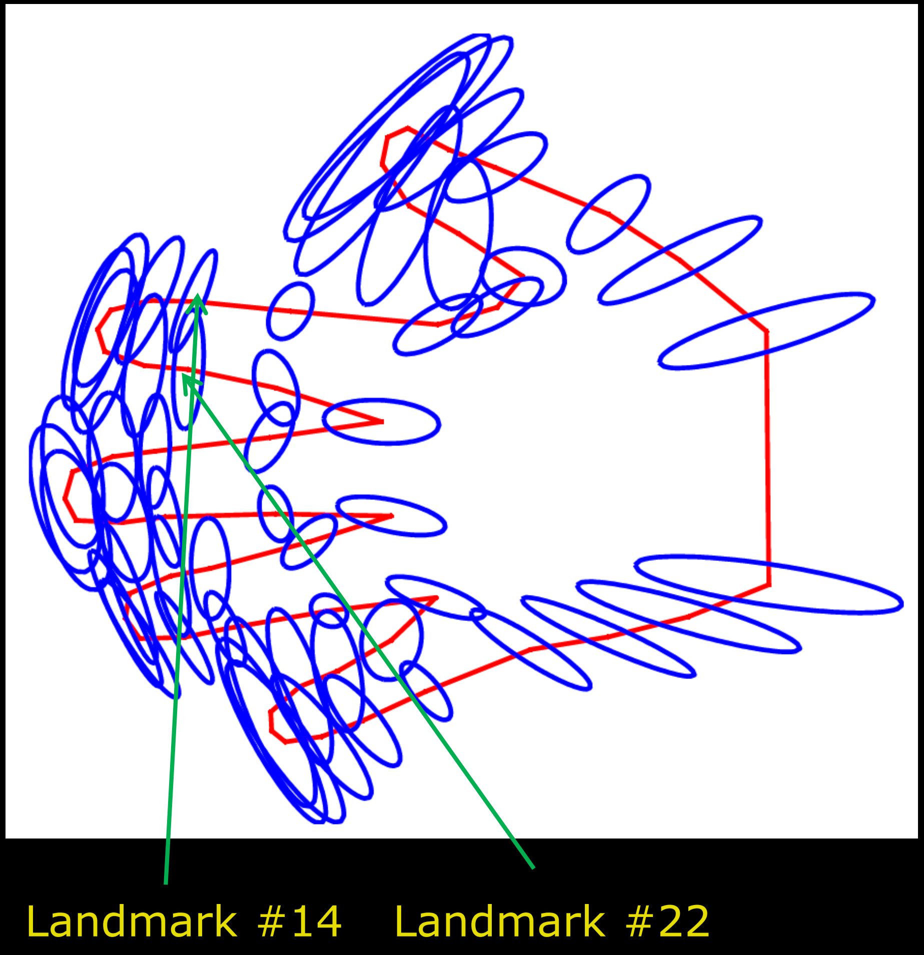
- Plot the Eigenvalues
- Explains how *important* each dimension is
- Cut away noise dimensions

Aligned shapes – what now



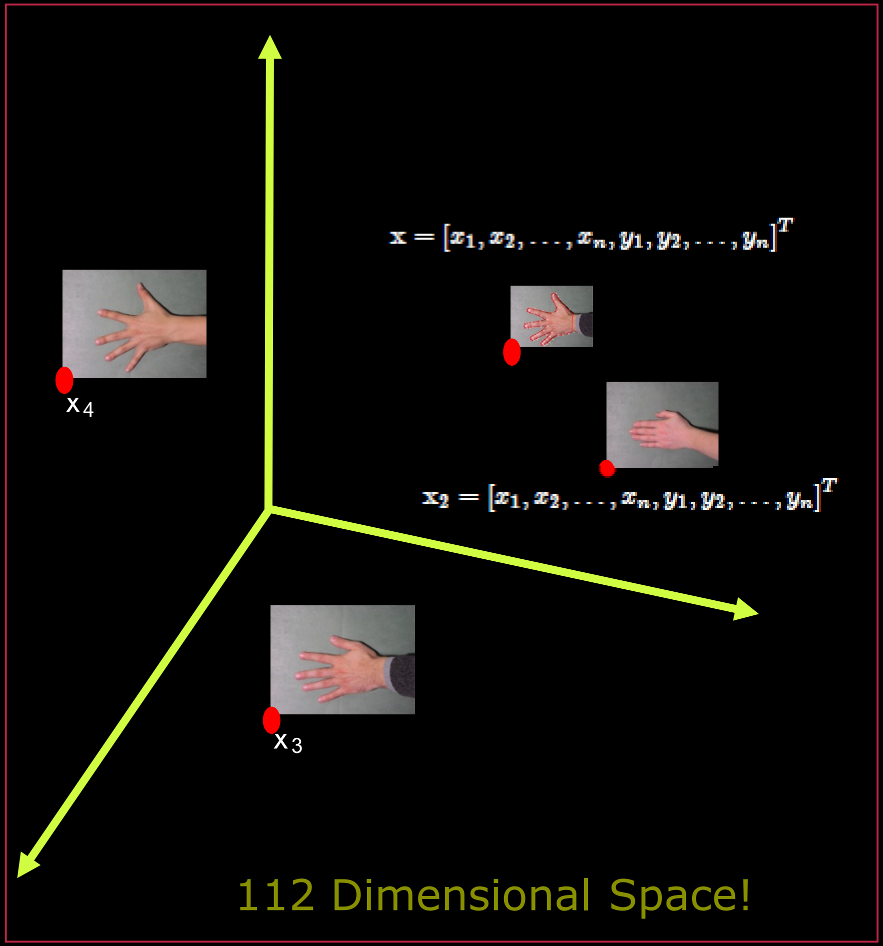
- Individual landmark variation
  - Over the training set
- What shape is the variation?

PCA Analysis



- PCA analysis on individual landmarks
- Describes the major direction of variation
- Landmarks are correlated!
- The movement over the shape is connected
- Return to shape space

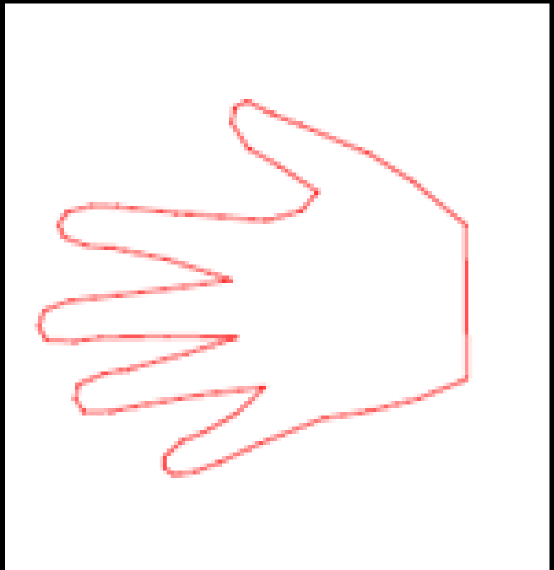
PCA in shape space



- Instead of doing PCA on 2D points we do it on 112D points
- Examine if our 40 *shapes is lying on a plane* in 112D space
- We find the directions that spans the maximum variation in shape space

Start by computing the shape average

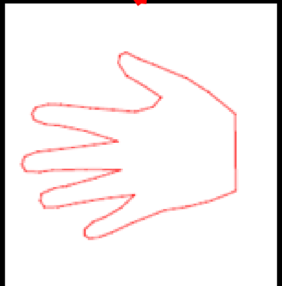
$$\bar{\mathbf{x}} = \frac{1}{s}\sum_{i=1}^{s}\mathbf{x}_i$$



Do the eigenvector analysis

$$\mathbf{S} = \frac{1}{s-1}\sum_{i=1}^{s}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

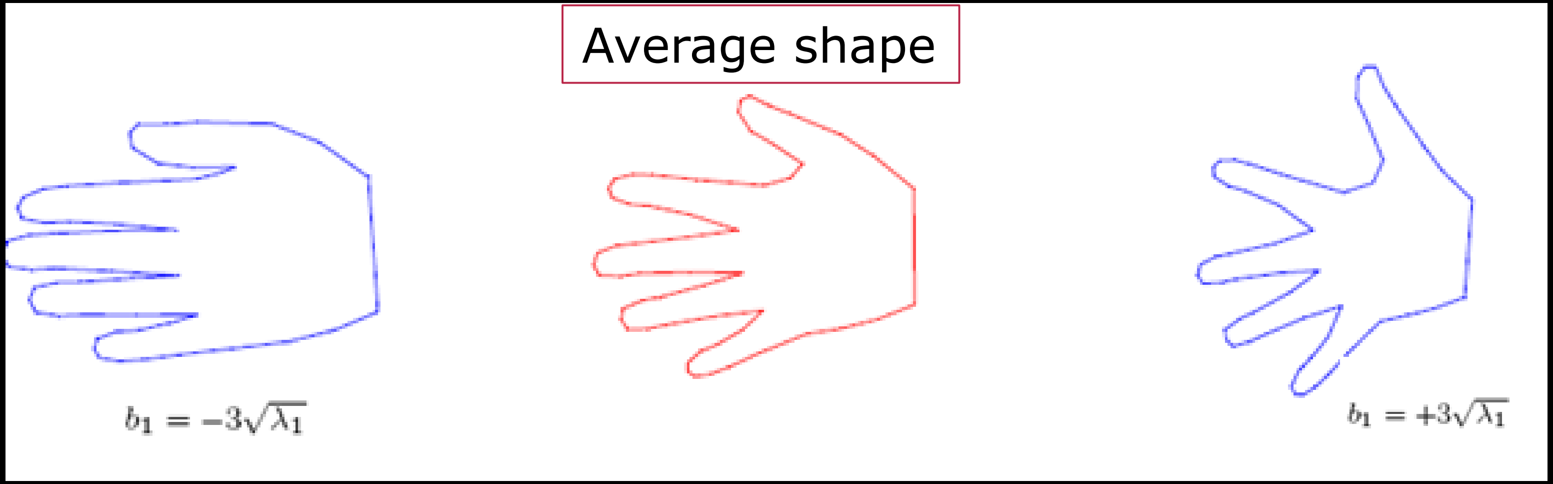
- Computing the covariance of the shape data



Average shape

Shape number i in the training set

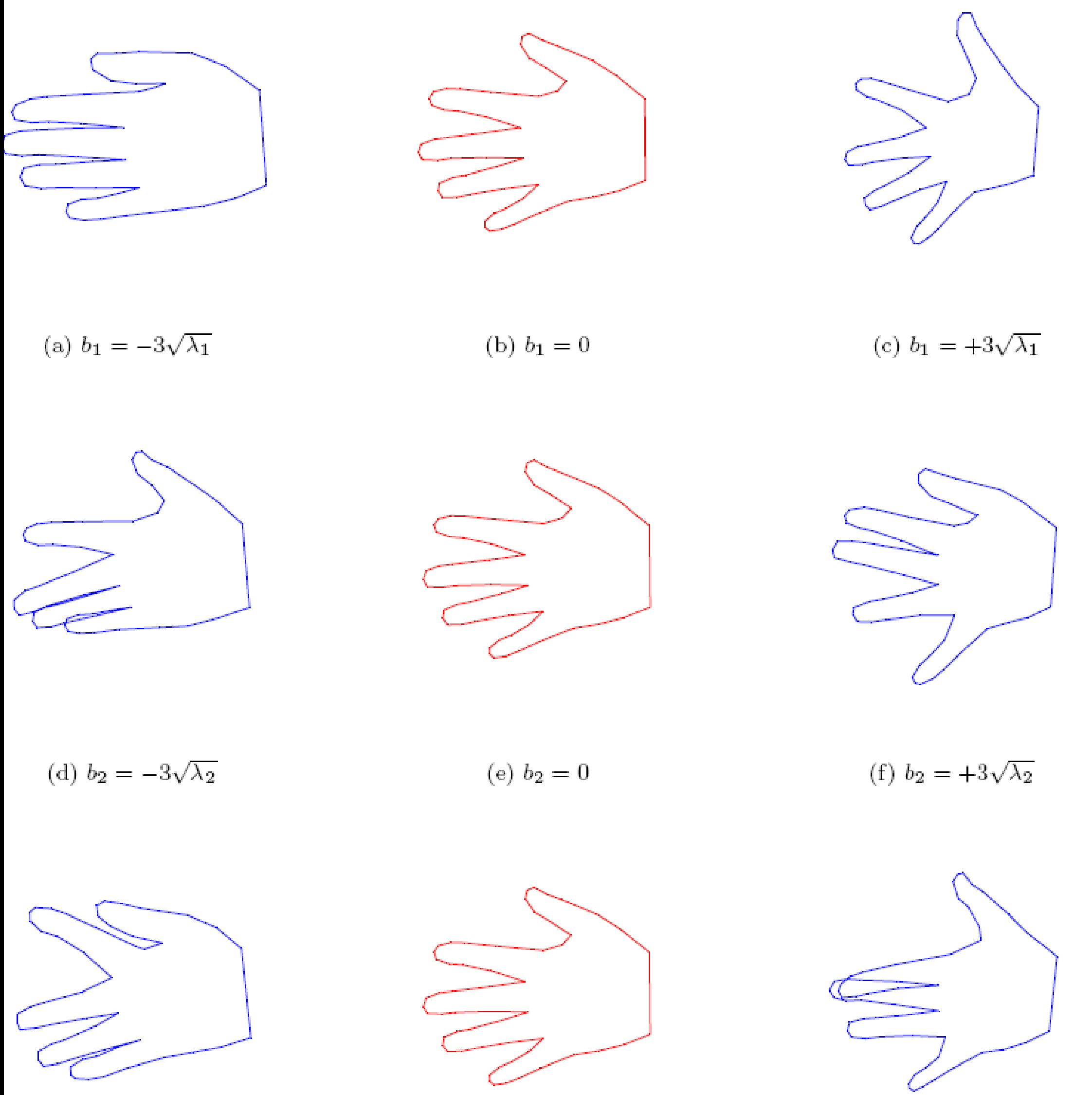
Visualizing variation



Visualizing the first principal component

$$\mathbf{x} \approx \bar{\mathbf{x}} + \Phi \mathbf{b}$$

Φ contains the t eigenvectors



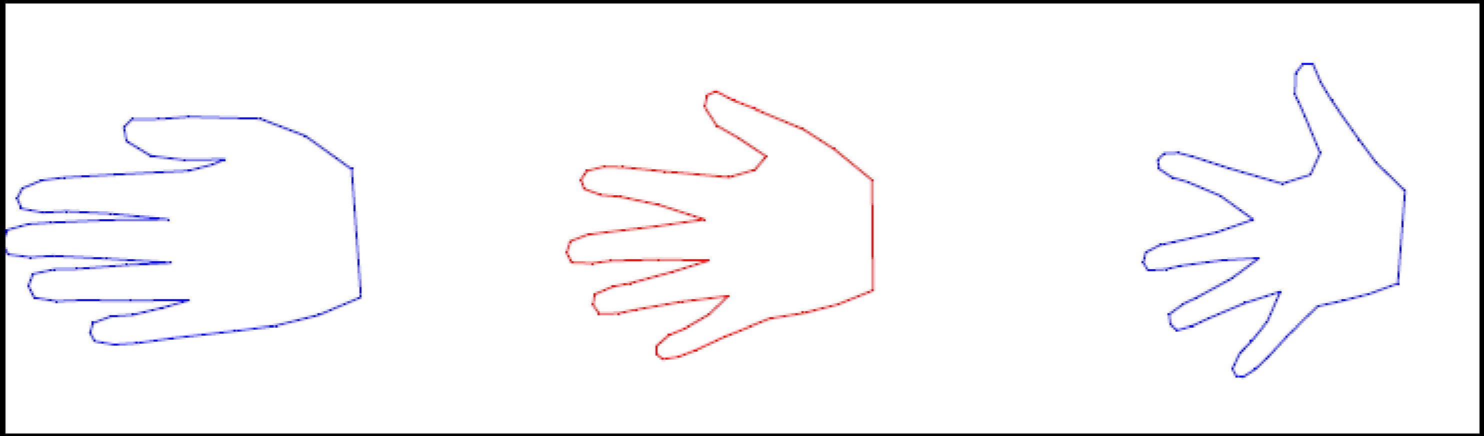
(a) $b_1 = -3\sqrt{\lambda_1}$ (b) $b_1 = 0$ (c) $b_1 = +3\sqrt{\lambda_1}$

(d) $b_2 = -3\sqrt{\lambda_2}$ (e) $b_2 = 0$ (f) $b_2 = +3\sqrt{\lambda_2}$

(g) $b_3 = -3\sqrt{\lambda_3}$ (h) $b_3 = 0$ (i) $b_3 = +3\sqrt{\lambda_3}$

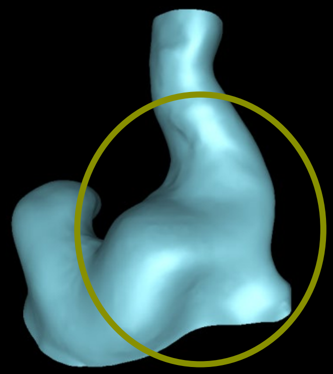
Results of Shape Analysis

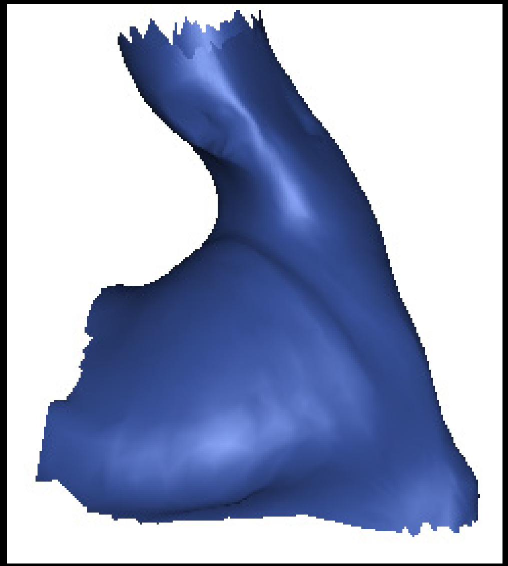
- Visualisation of the major variation of the shape over a population



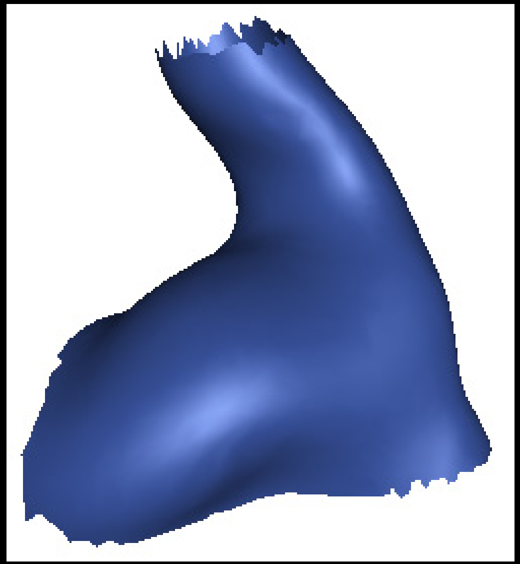
Hearing Aid Design

- Main variation of the shape of the ear canal
- Found using principal component analysis
- First mode of variation
- 7 modes explain 95% of the total variation

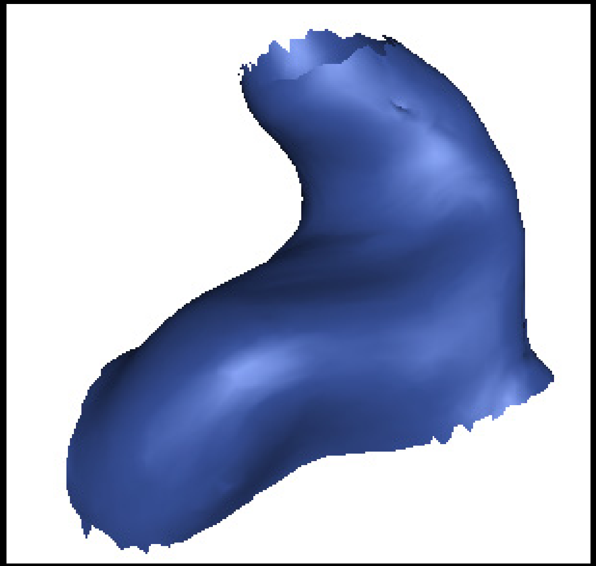




Average-1. mode



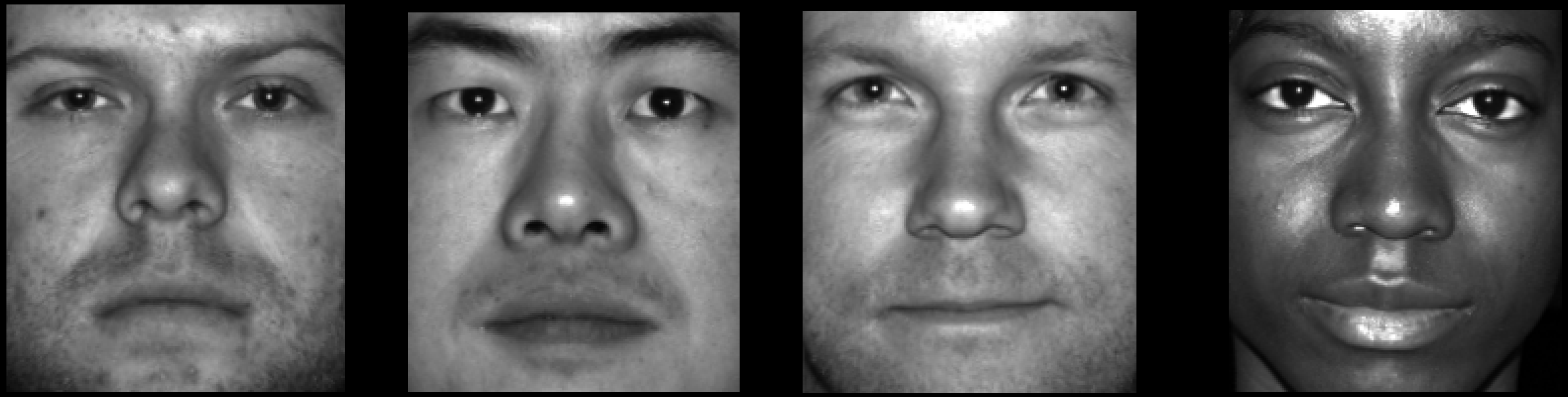
Average



Average+1. mode

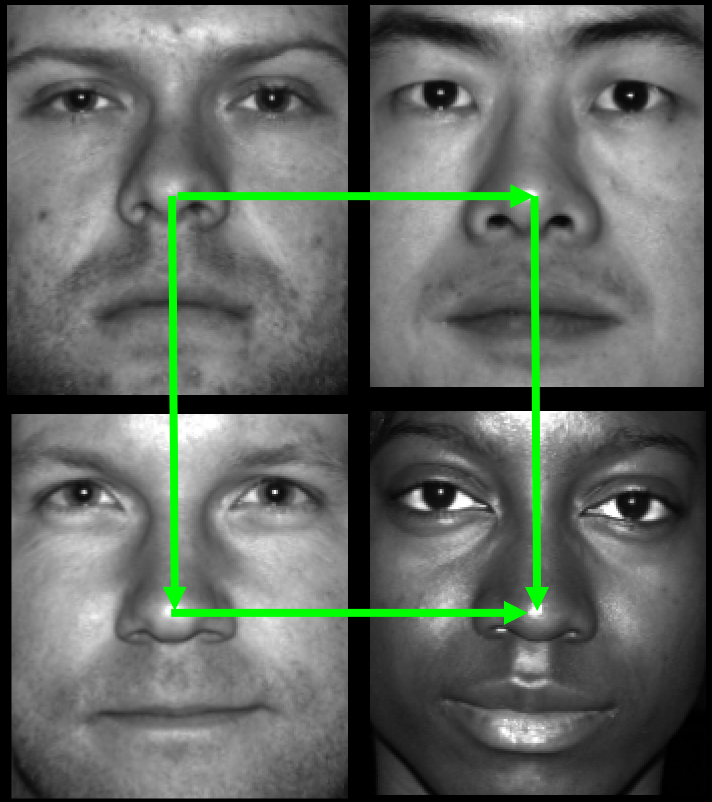
Modelling shape and appearance

- A model that can both model the shape of an object and the appearance (the texture)
- Texture: The pattern of intensities (or colors) across an image patch



Back to lecture 3: Eigenfaces

Face data

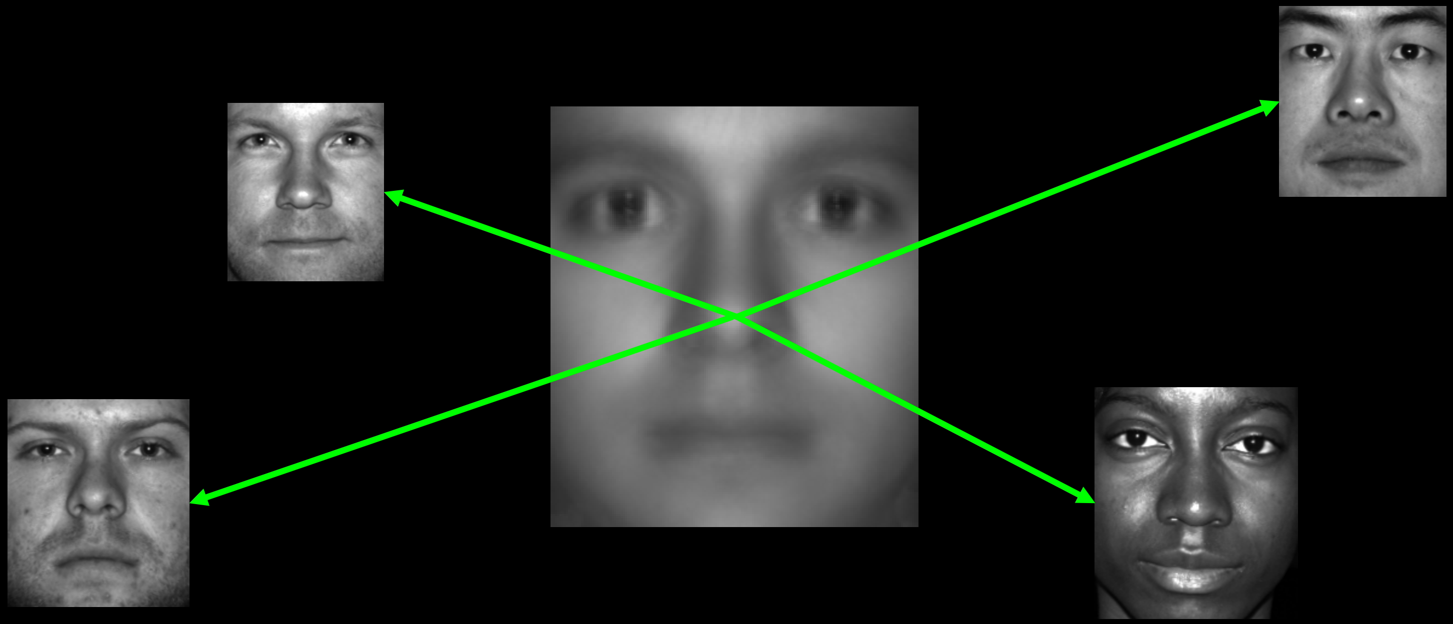


- 38 face images
  - 168 x 192 grayscale
- Aligned
  - The anatomy is placed ”in the same position in all image”
- Same illumination conditions on the images we use

The Extended Yale Face Database B
http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

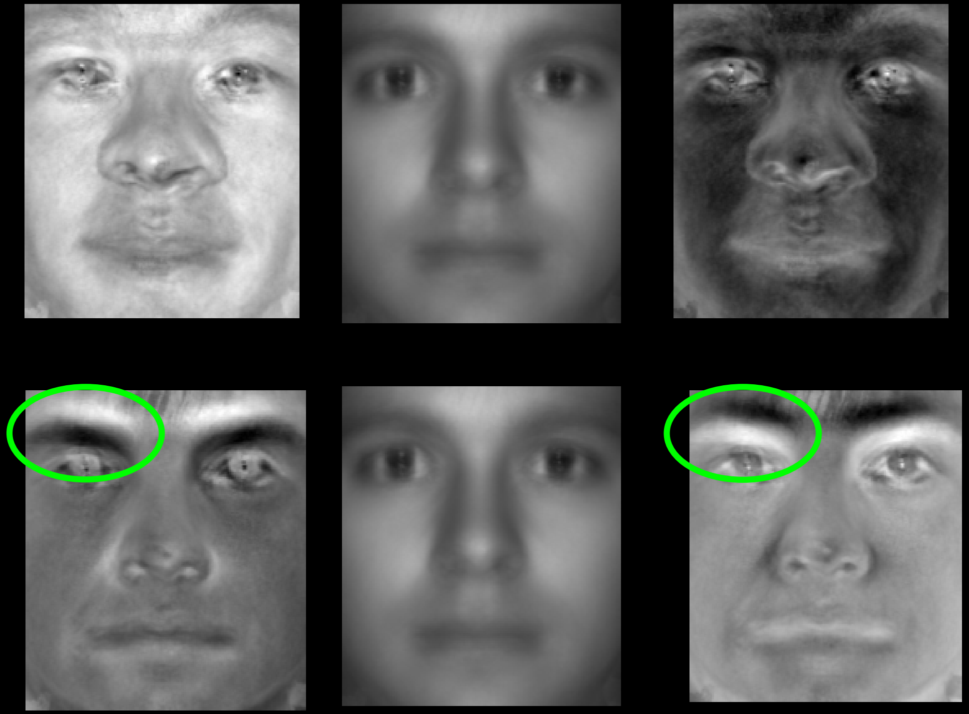
Analyzing the deviation from the mean face

- We want to do the principal component analysis on the *deviations from the average face*



Visualizing the PCA faces

Main deviations from the average face



First PC – 40% of variation

Second PC – 8% of variation

-PC

Average face

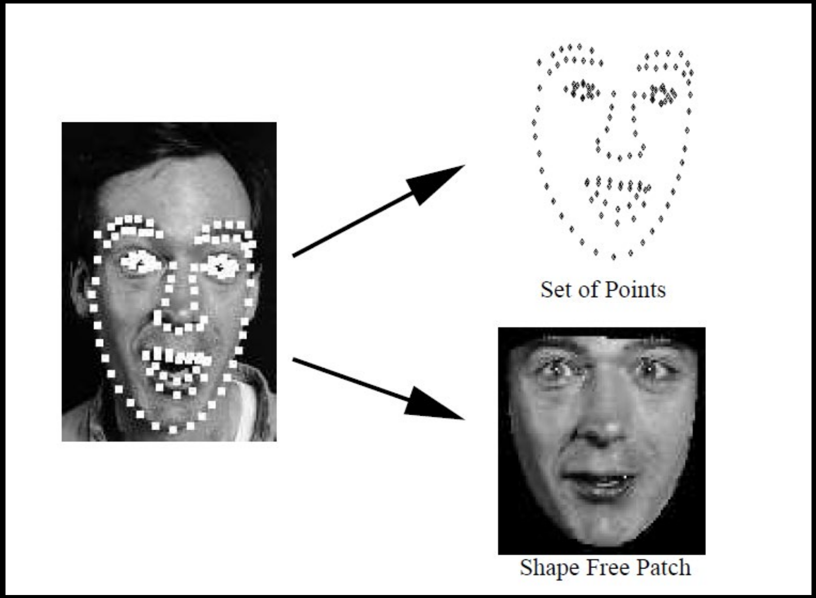
+PC

A tool to see major variations – brow lifting

Eigenfaces: Modelling texture

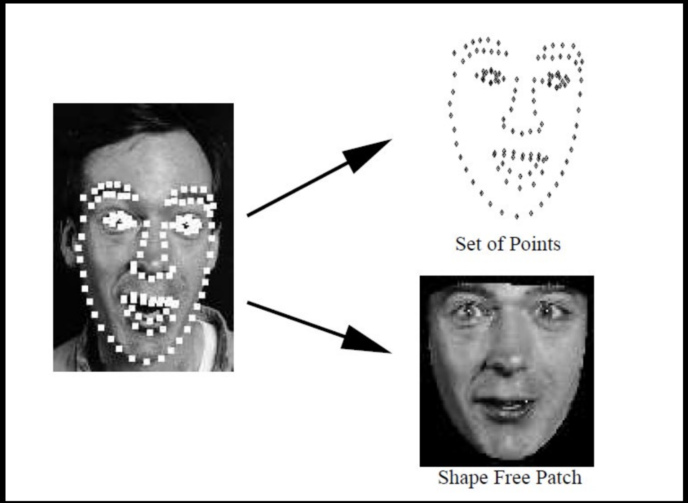
- The modelling of the pure appearance
- Without removing variation in shape
- No *decoupling* of shape and appearance

Decoupling shape and texture



- Warp each face to average shape using the landmarks
- Non-linear geometrical transformation
- Sample the texture from the warped face

Eigenfaces on warped faces



- Same PCA modelling as in the Eigenfaces approach
- Just slightly different notation

$$\mathbf{g} = \bar{\mathbf{g}} + \mathbf{P}_g \mathbf{b}_g$$

Combined shape and appearance model

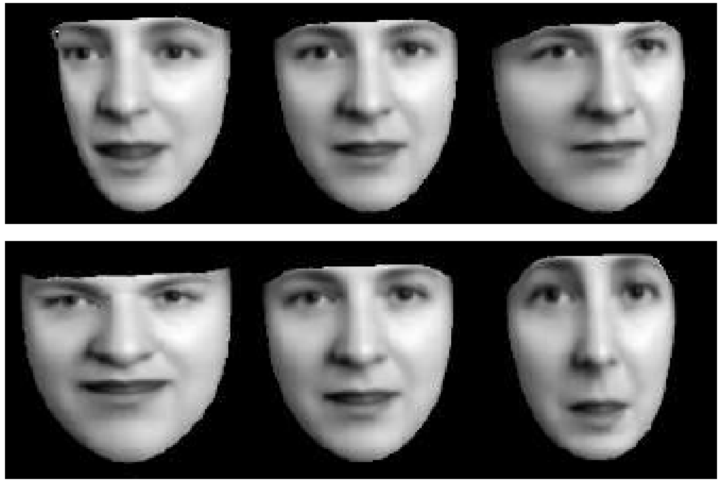


Figure 5.2: First two modes of shape variation (± 3 sd)

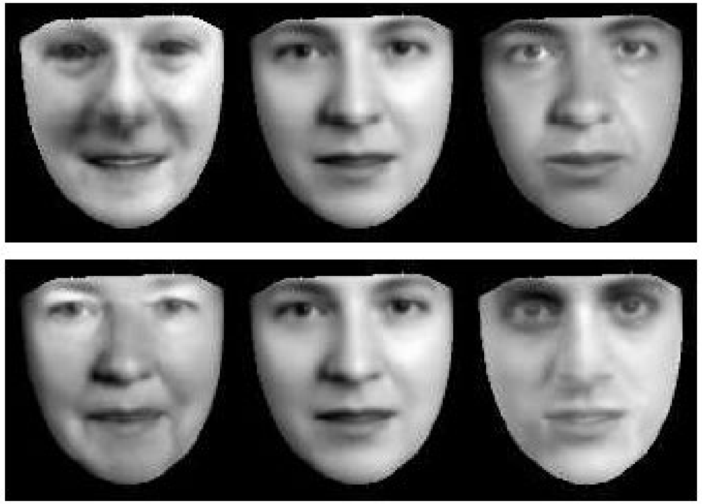


Figure 5.3: First two modes of grey-level variation (± 3 sd)

Facial Analysis

- Demo of AAM explorer

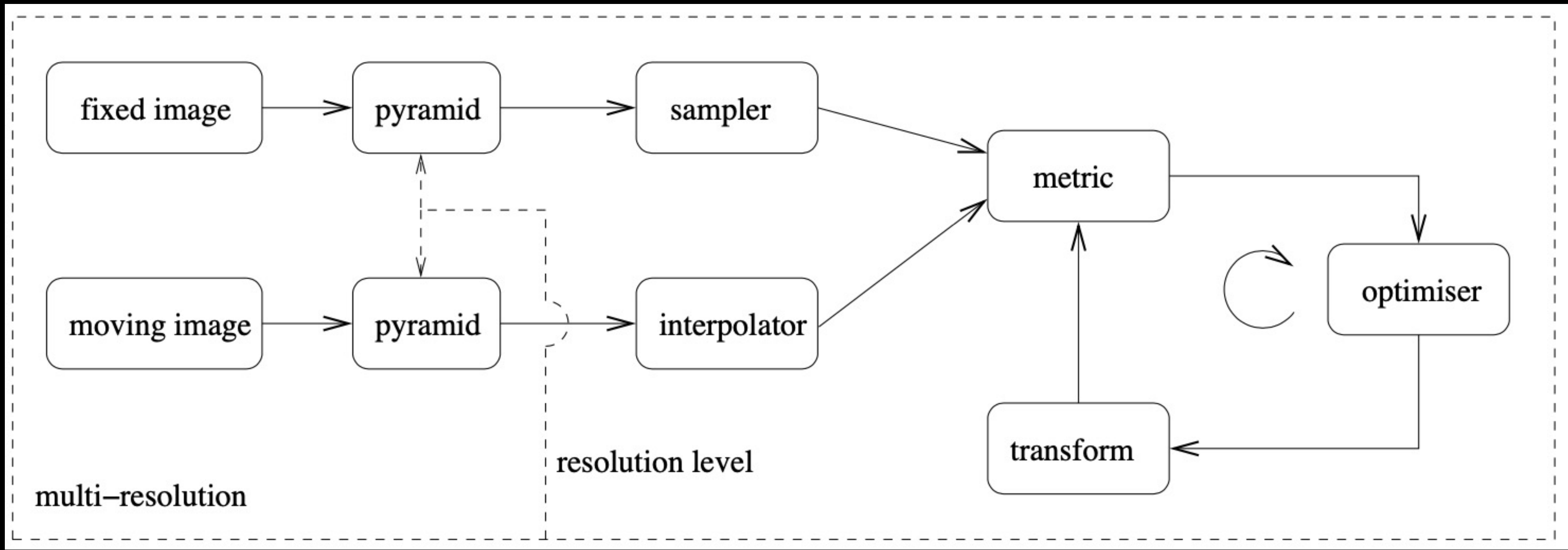
Image Analysis

Tim B. Dyrby
Rasmus R. Paulsen
DTU Compute

tbdy@dtu.dk

http://www.compute.dtu.dk/courses/02502

Lecture 10 – Advanced image registration



Klein et al 2010. (IEEE Trans Med Img)

https://elastix.lumc.nl

What can you do after today?

- Describe difference between a pixel and voxel
- Describe the general image-to-image registration pipeline
- Describe 3D geometrical affine transformations
- Choose a suitable similarity metric given the image modalities to register
- Compute the normalized correlation coefficient (NNC) between two images
- Compute Entropy
- Describe the concept of iterative optimizers
- Compute steps in the gradient descent optimization steps
- Describe the pyramidal principle for multi-resolution strategies
- Select a relevant registration strategy: 2D to 3D, Within- and between objects and moving images

Go to **www.menti.com** and use the code **1682 8098**

Associations to a mountain view

Mount Everest - Himalayas

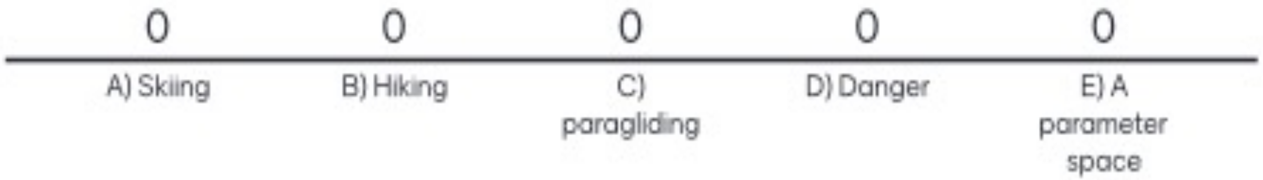


Image Registration pipeline

- The input images
  - Fixed image: Reference image
  - Moving image: Template image

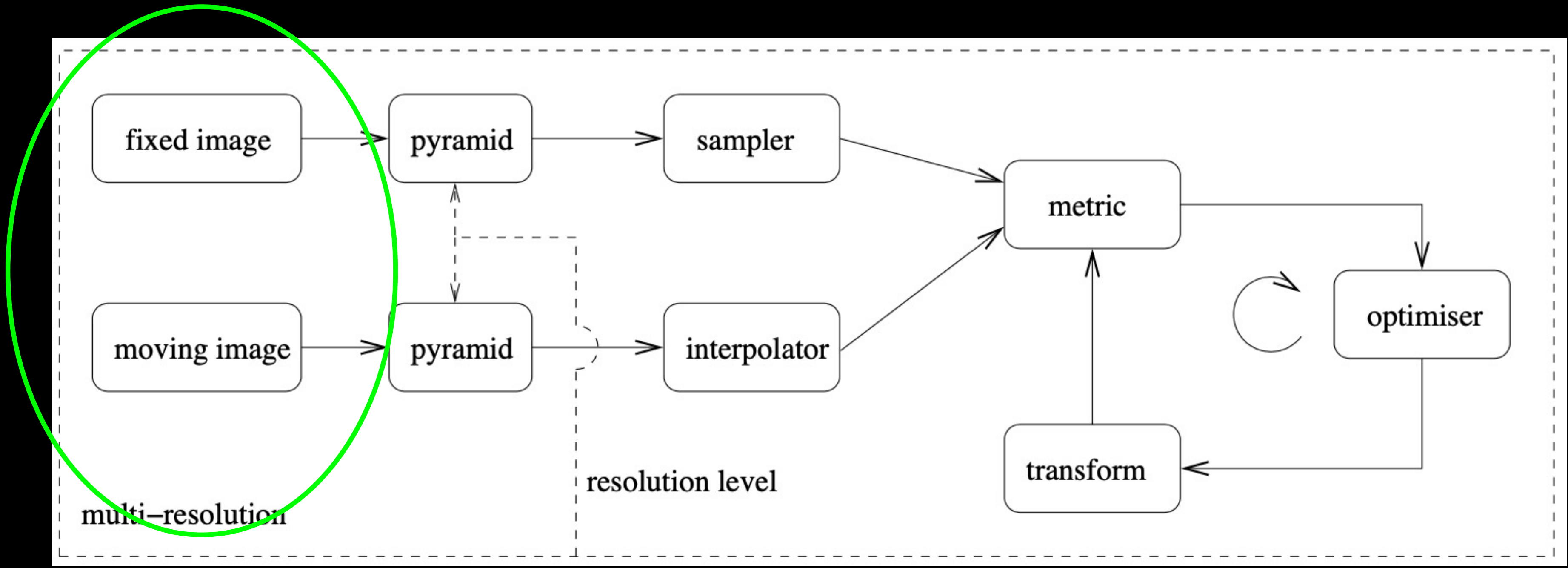
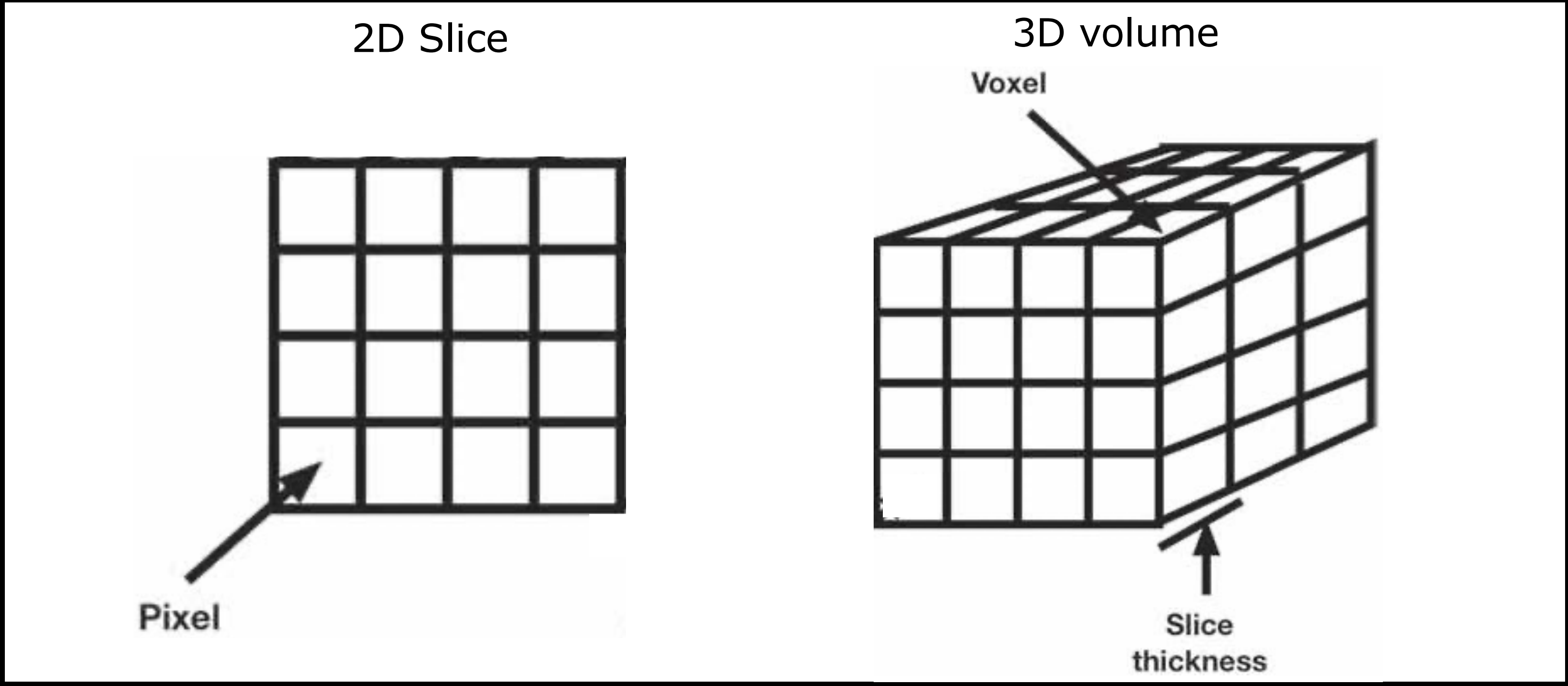
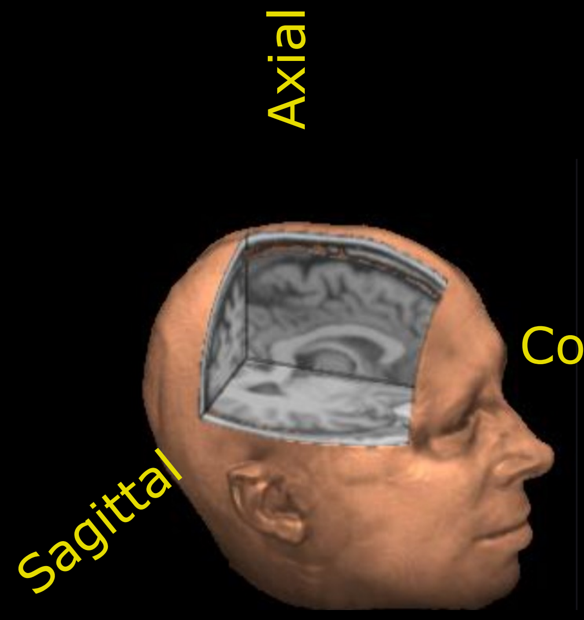


Image volumes

- Image slice: 2D (NxM) matrix of pixels
- Image volumes: 3D (NxMxP) matrix of voxels
  - An element is a volume pixel i.e. voxel
- Pixel vs voxel intensity
  - Integrated information within an area or volume



3D rendering



3D image viewing

- Three orthogonal views
  - Fine structural details at slice level
  - Hard to get 3D surface insight
- Rendering of surfaces
  - Surface insight
  - Limited to clear surfaces

Slices three orthogonal views

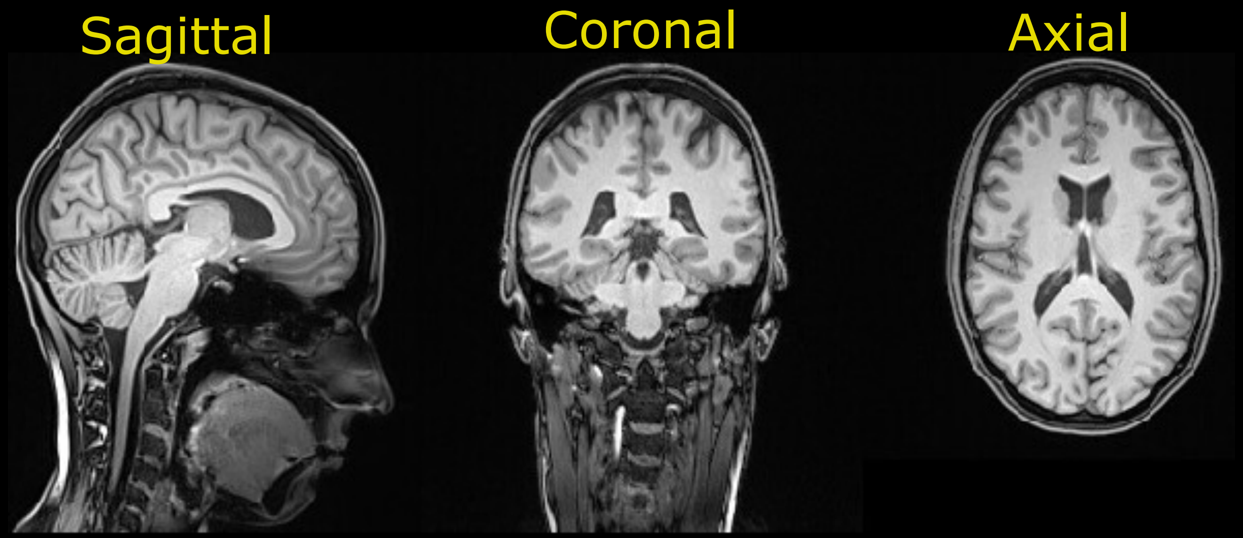


Image volumes

- Stacked slices: 2D to 3D
  - Object cut into slices, imaged and stacked
  - Still pixels – not voxel
- Registration challenges
  - Geometrical distortions between slices

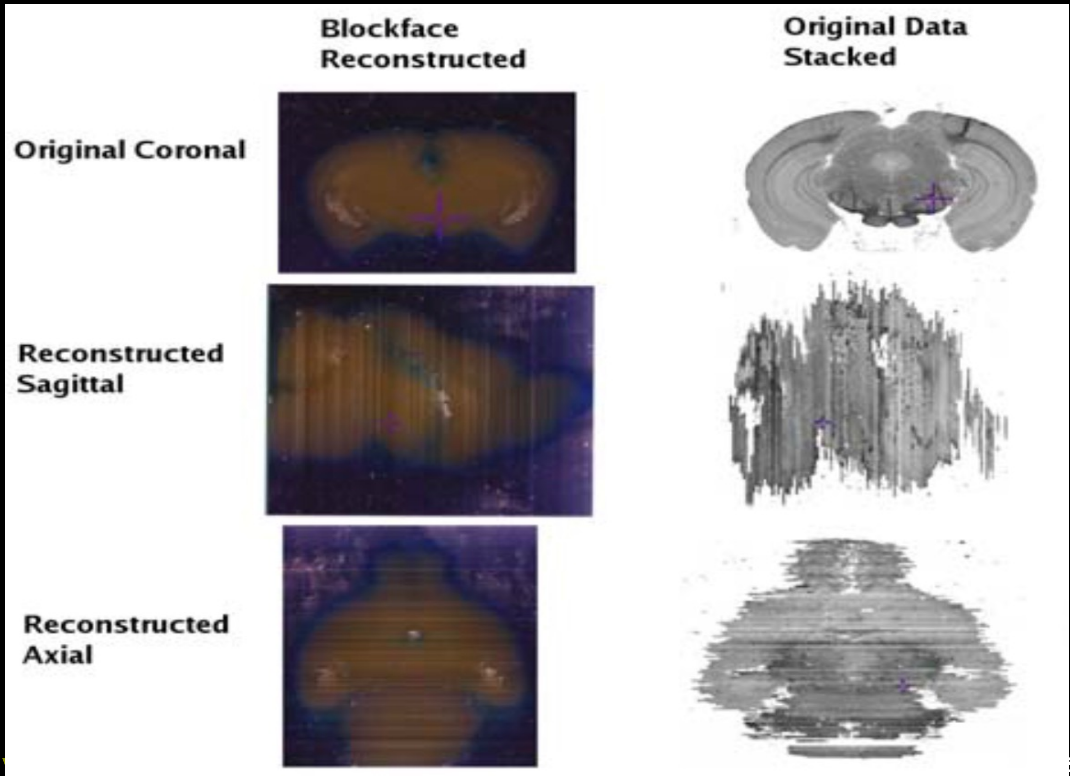
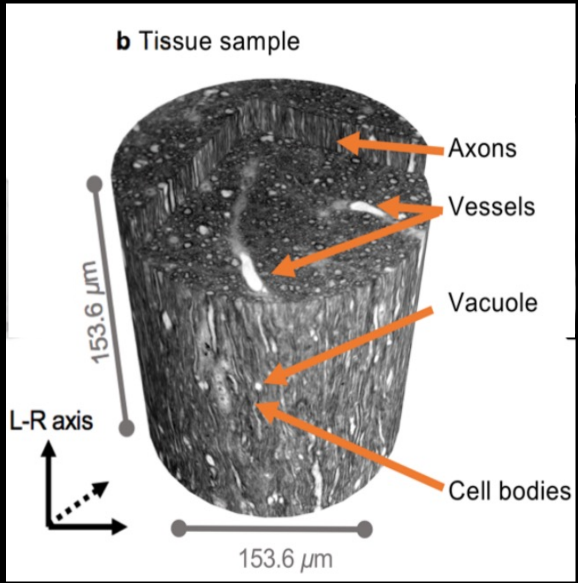


Image volumes

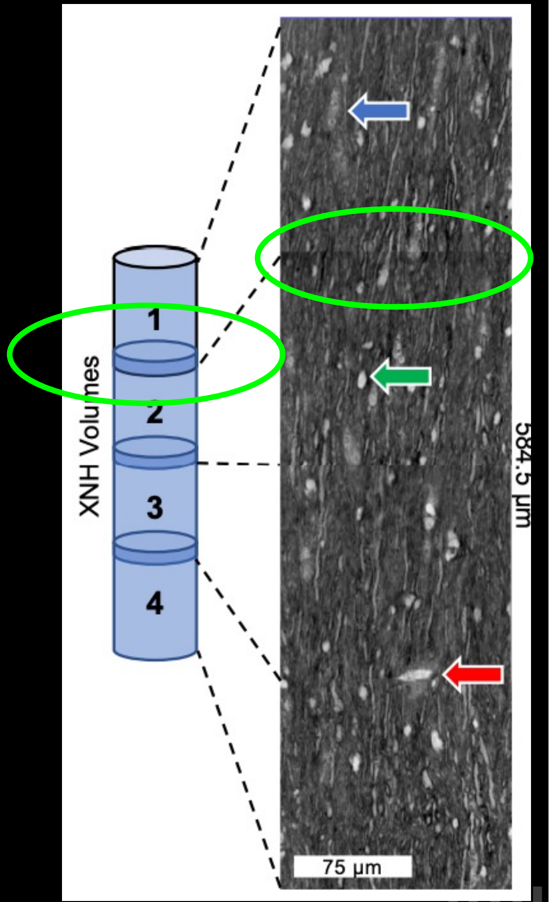
- Intact sample
  - No sample cutting
- Registration challenges:
  - Stacking 3D volumes

Synchrotron x-ray imaging
Tissue sample 1mm
75 nm isotropic resolution voxels



Andersson et al, 2020 (PNAS)

Stacked 3D volumes



MRI
Whole brain
1 mm isotropic resolution voxels

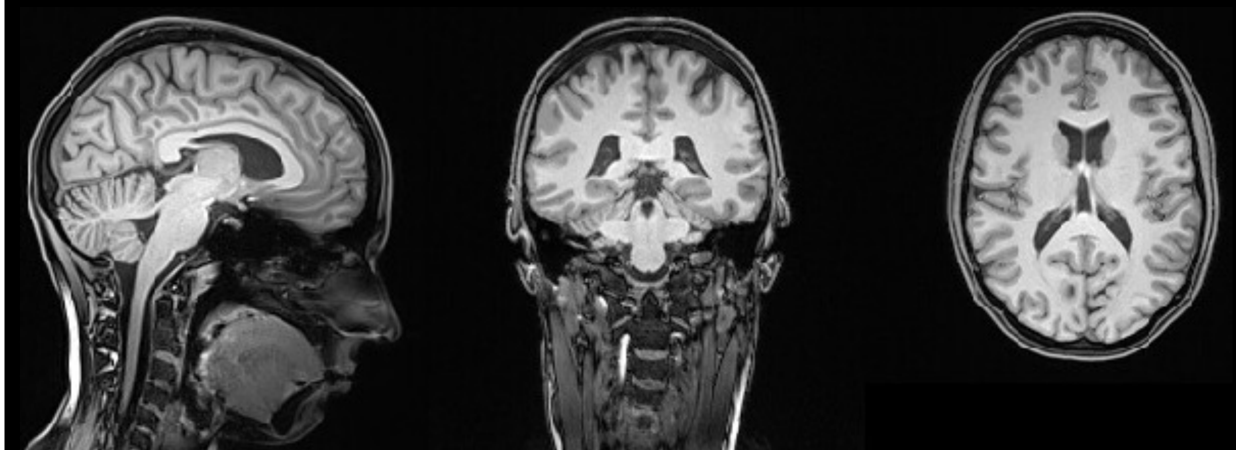
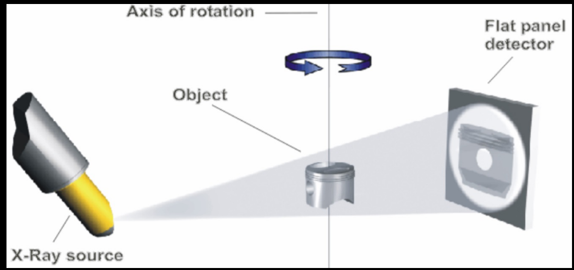


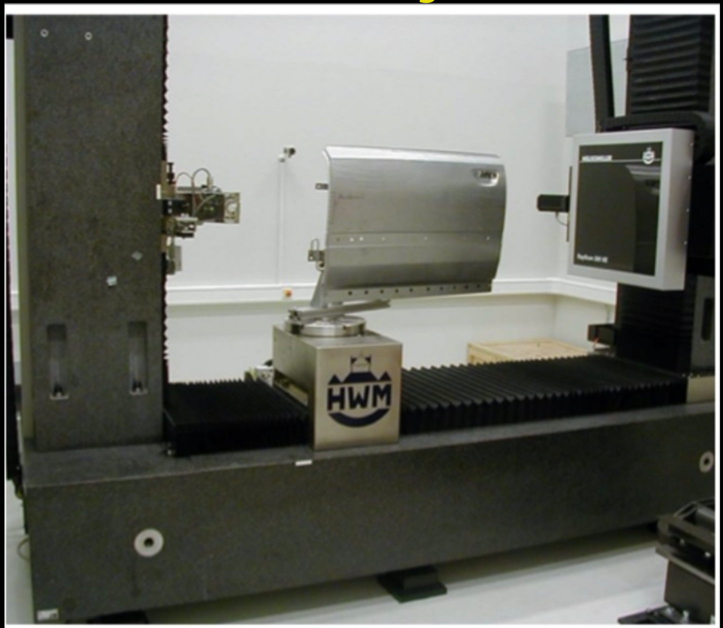
Image volumes

- Image of intact sample
  - No sample cutting
- Registration challenges:
  - Multi image resolution: Fit Region-of-interest image to whole object image

Rotating sample in x-ray

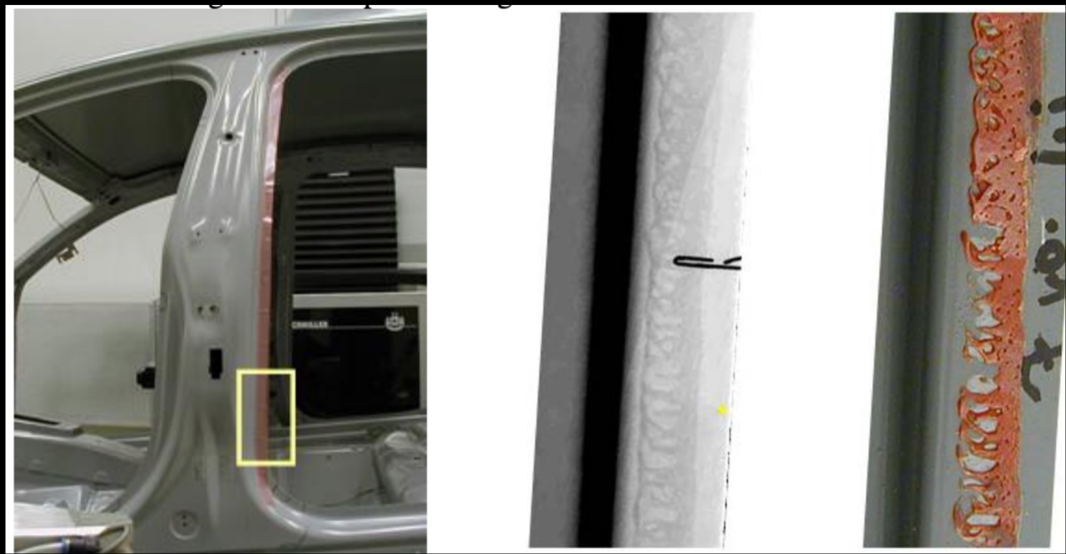


CT scanning



Car door AUDI A8, size: 1150 mm

Region of interest (ROI) | CT of ROI (non-destructive) | Microscope (destructive)

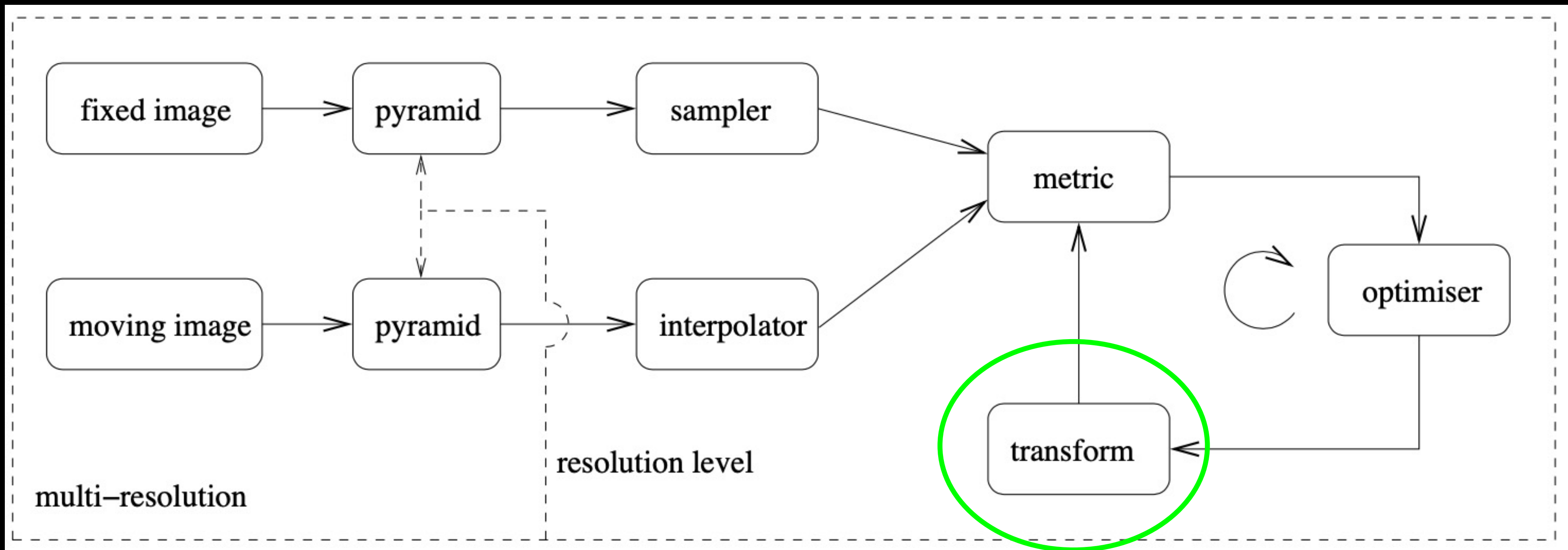


The inspection of a glued joint of a car body

Simon et al, 2006 (ECNDT)

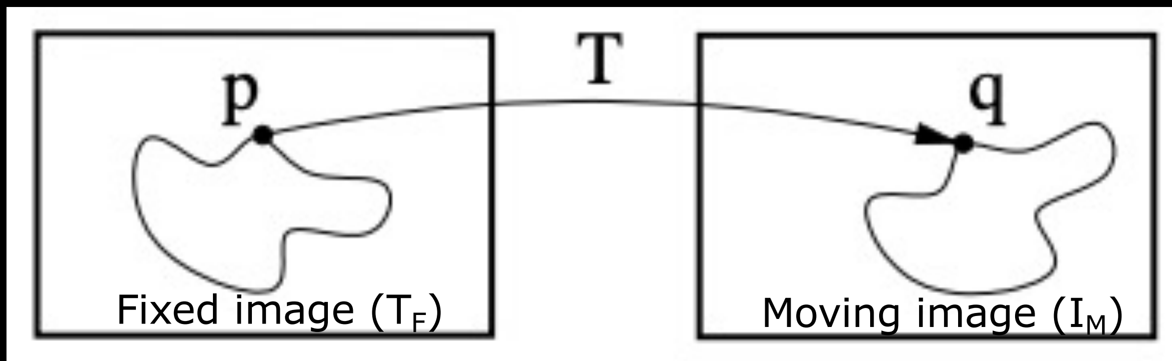
Image Registration pipeline

- Geometrical transformations



Geometric transformations

- Translation
- Rotation
- Scaling
- Shearing



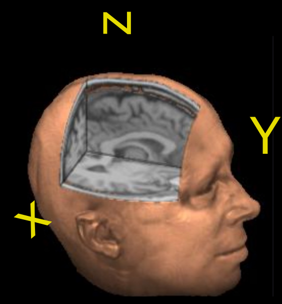
$$\hat{T} = \arg\min_{T} C(T; I_F, I_M)$$

Translation 2D vs 3D

- The image is shifted
  - 2D: Inspect one slice plan
  - 3D:Inspect three slice plans

3D: (x,y,z)-plans

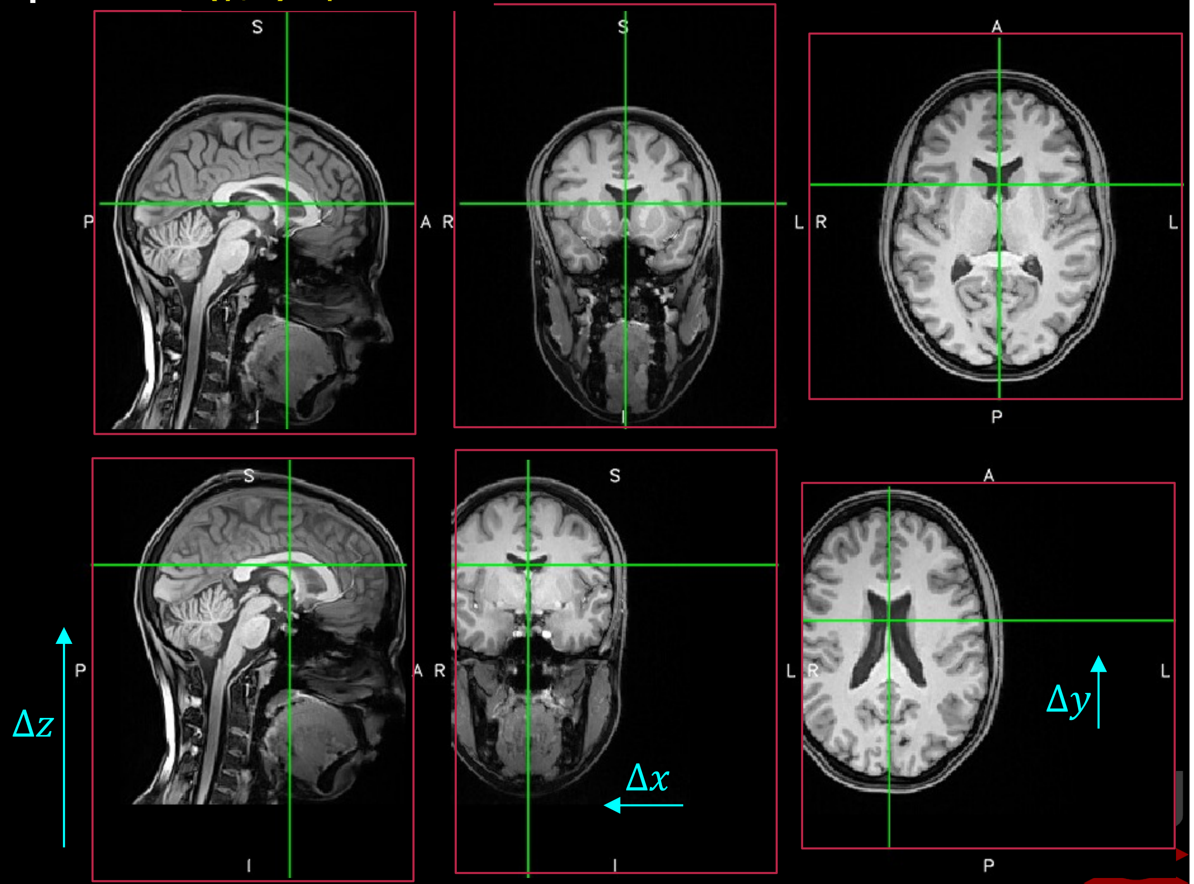
$$\begin{bmatrix}\Delta x\\ \Delta y\\ \Delta z\end{bmatrix} = \begin{bmatrix}60\\ 20\\ 15\end{bmatrix}$$



(y,z) -plan

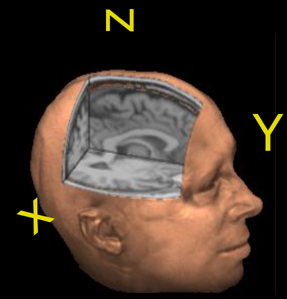
(x,z)-plan

(x,y)-plan



2D: (x,y)-plan

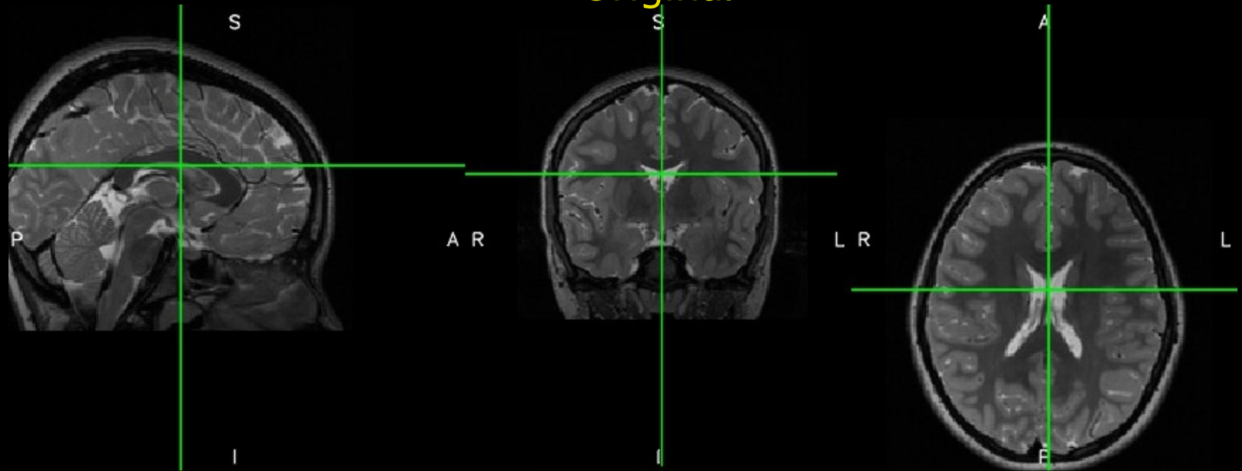
$$\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = \begin{bmatrix}60\\ 20\end{bmatrix}$$



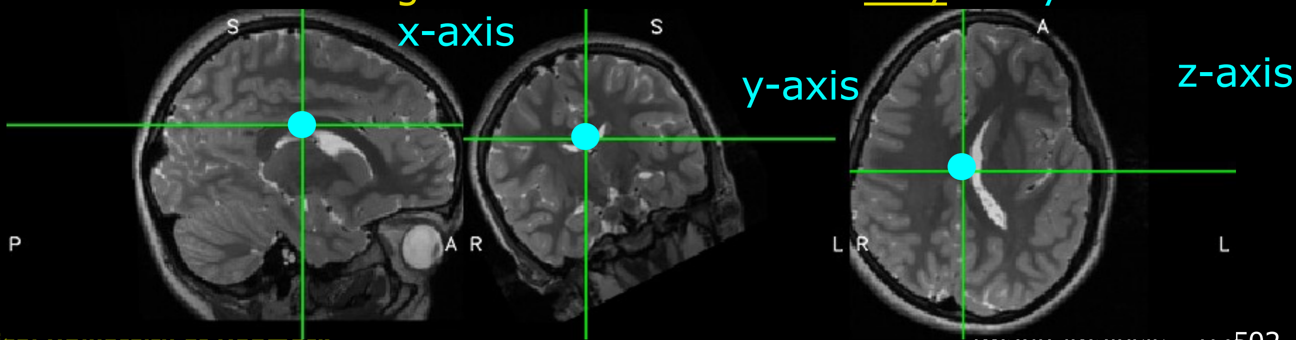
Rotation 3D

- The image is rotated around a origin (e.g. the centre-of-mass)
- Rotate the object around three axis hence three angles.
  - Inspect all three views to identify a rotation

Original

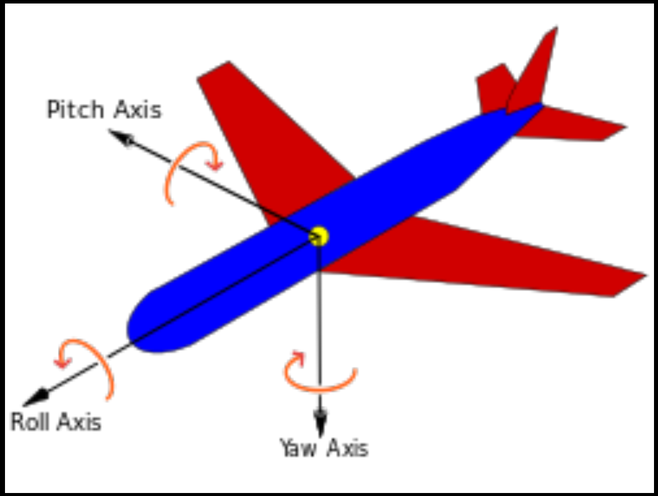


Rotated: 27 degree clock wise around <u>only</u> the y-axis

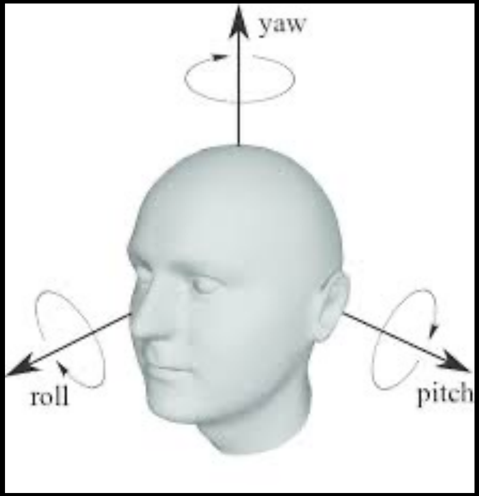


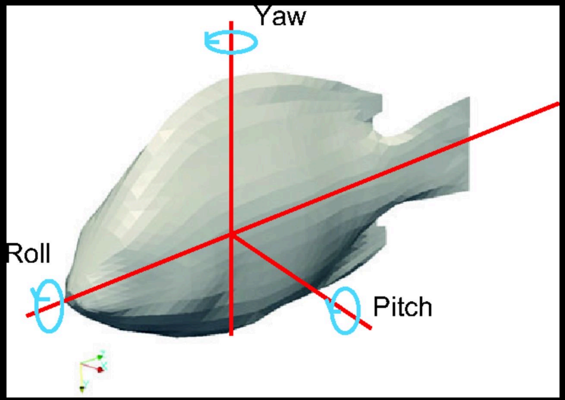
3D Rotation coordinate system

- Three element rotations round the axes of the coordinate system
- Pitch, Yaw and Roll
  - Defined differently for different systems (typ. related to the forward direction)



The principal axes of an aircraft according to the air norm DIN 9300

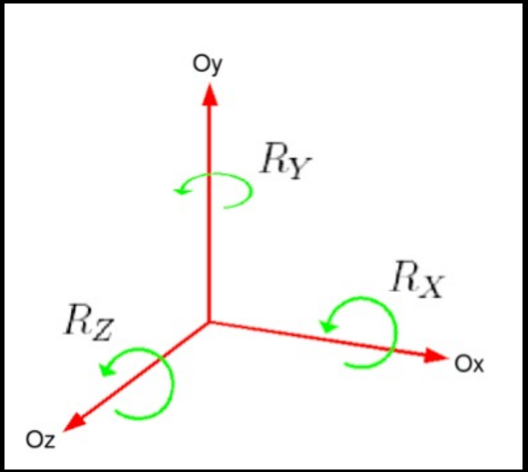




3D Rotation coordinate system

- Three composed element rotations
  - Angles: α, β, γ
- The order matters
  - Several conventions exist
- Remember: Know your origin!

Axis-Angle representation

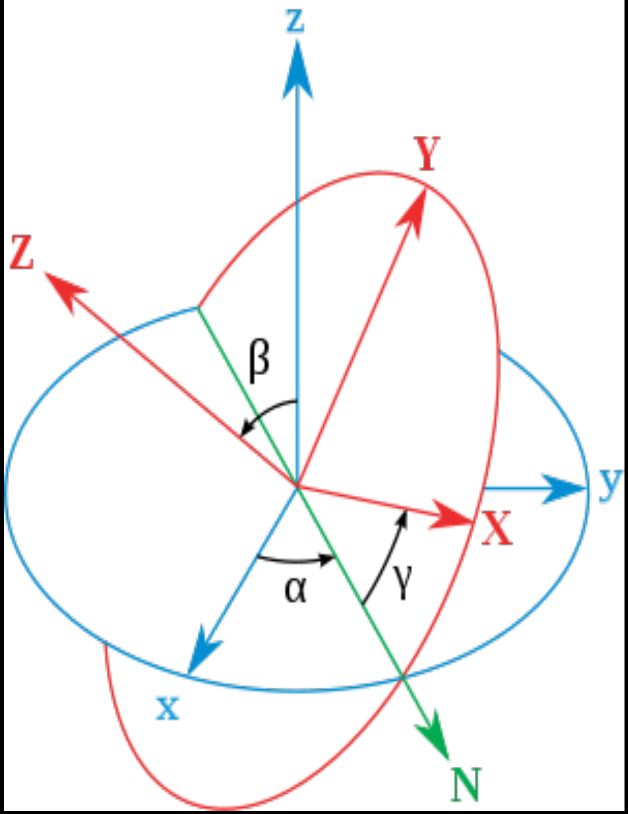


$$\boldsymbol{R}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \quad \boldsymbol{R}_Y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad \boldsymbol{R}_Z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

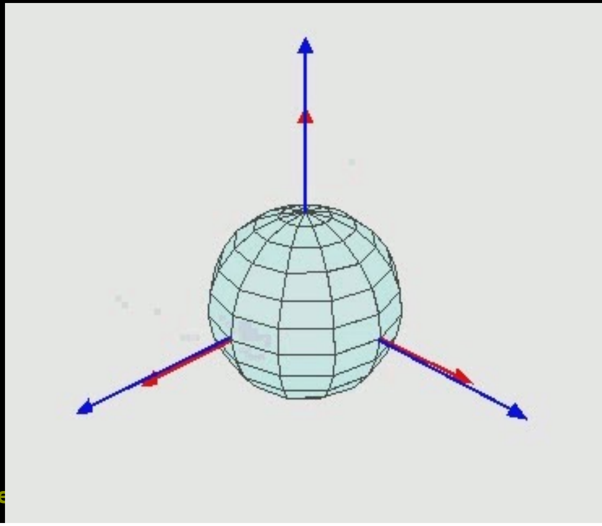
Pitch Roll Yaw

3D Rotation coordinate system

- The Euler angel convention:
  - α: Around the z-axis. Defines the line of nodes (N)
  - β: Around the X-axis defined by N
  - γ: Around the Z-axis from N
- The order of coordinate system rotations:
  - Rotation order around the:
  - z-axis: Initial: Original frame (x,y,z): α
  - *X-axis: First coordinate system rotation (X,Y,Z): β*
  - *Z-axis: Second coordinate system rotation (X,Y,Z): γ*



wikipedia.org/wiki/Euler_angles



Quiz 1: Affine 3D transformation

How many parameters?

A) 6

B) 5

C) 16

D) 12

E) 3

SOLUTION:

Translation: P=3

Rotation: p=3

Scaling: p=3

Shearing: p=3

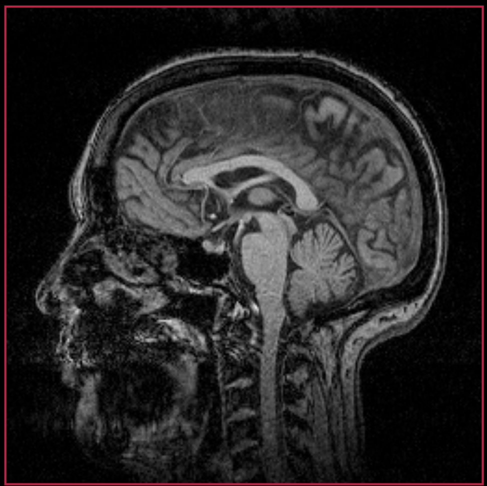
Scaling in 3D

- The size of the image is changed
- Three parameters:
  - X-scale factor, S_x
  - Y-scale factor, S_y
  - Z-scale factor, S_z

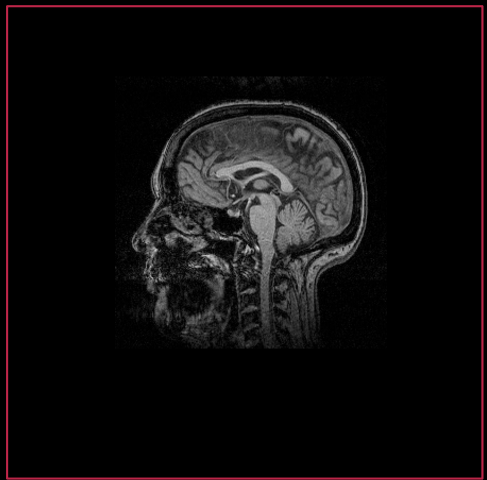
$$A = \begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & Sz \end{bmatrix}$$

- Anisotropic scaling:

z



y



$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$$

Shearing in 3D

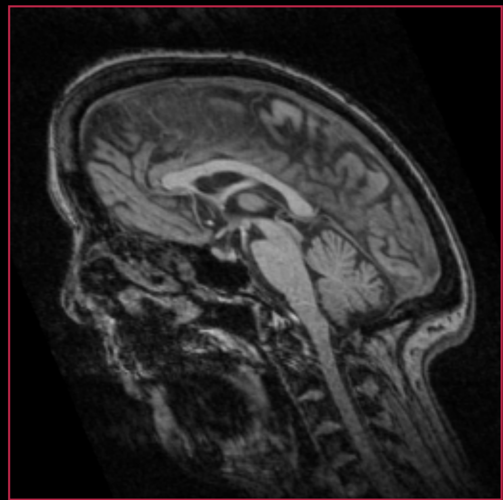
- Pixel shifted horizontally or/and vertically
- Three parameters

$$A = \begin{bmatrix} 1 & Syx & Szx \\ Sxy & 1 & Syz \\ Sxz & Syz & 1 \end{bmatrix}$$

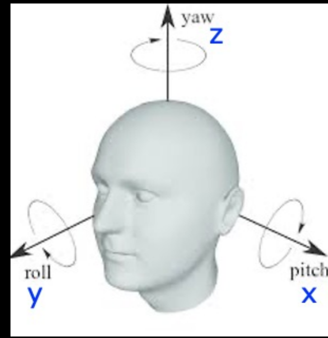
z

y

Shearing (z,y)-plan



Axis-Angle representation



Combining transformations

Translation: $$A_T = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotations: $$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) & 0 \\ 0 & -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_y = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_z = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- x=pitch
- y=roll
- z=yaw

Scaling: $$A_s = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Shear: $$A_z = \begin{bmatrix} 1 & Sxy & Sxz & 0 \\ 0 & 1 & Syz & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Affine transformation: $$A = \underbrace{A_T * (R_x * R_y * R_z)}_{\text{Rigid}} * A_z * A_s$$

github.com/fieldtrip/fieldtrip/blob/master/external/spm8/spm_matrix.m

Different transformations

- Linear: Affine transformation
- Non-linear: Piece-wise affine or B-spline
  - Remember: First apply the linear transformations!

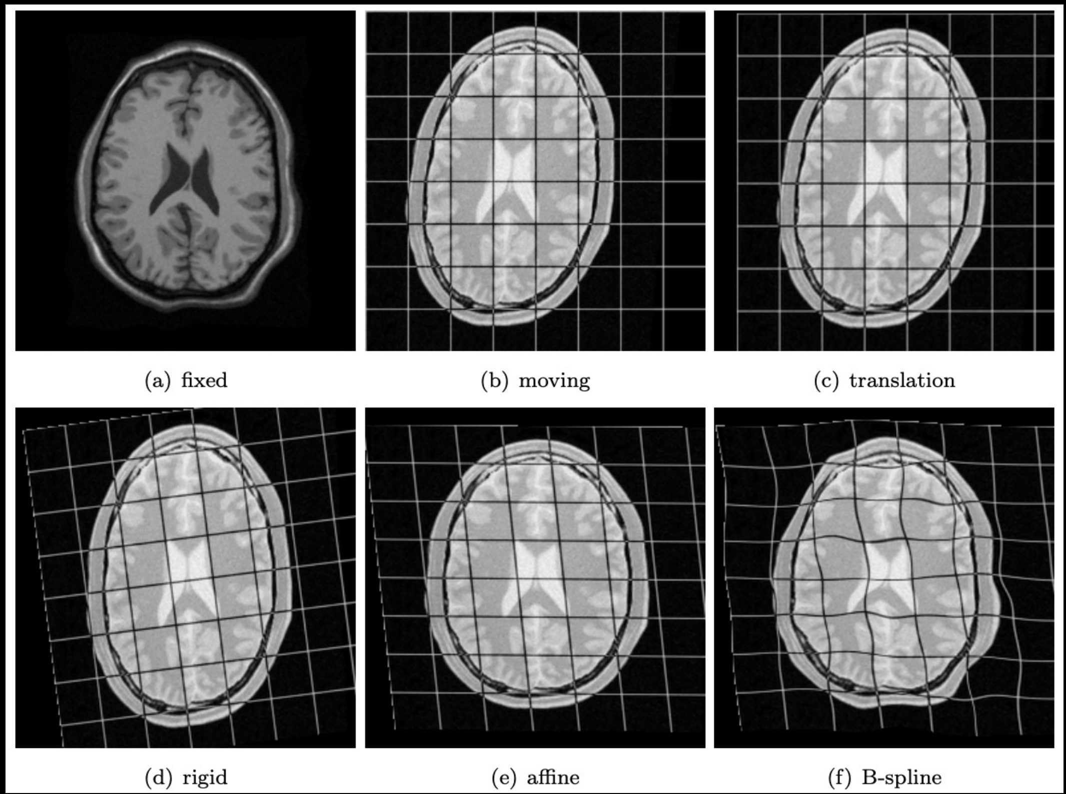
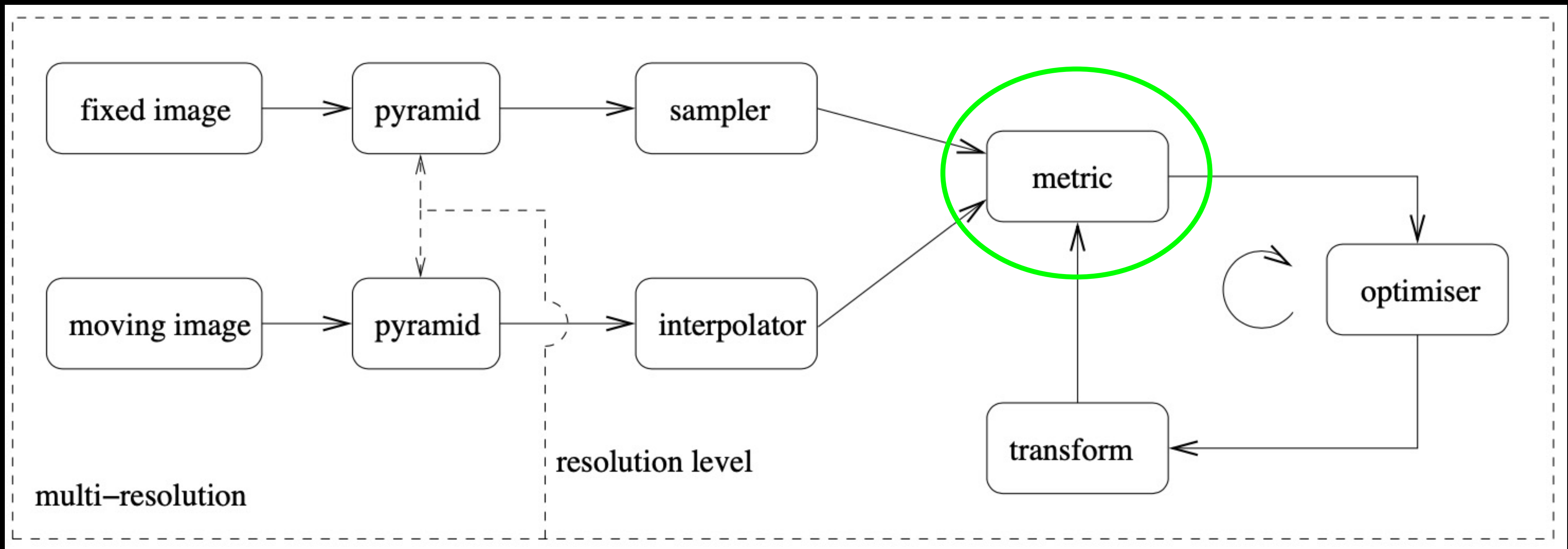


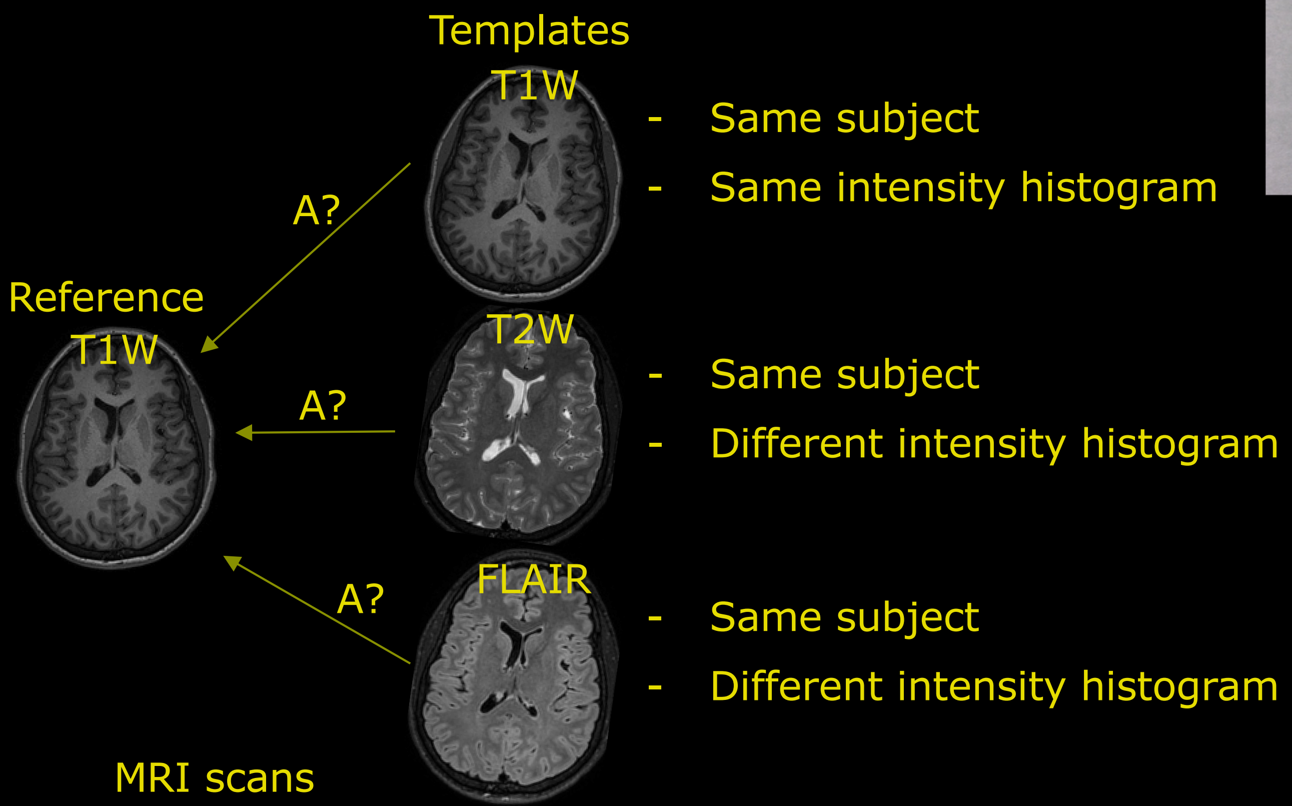
Image Registration pipeline

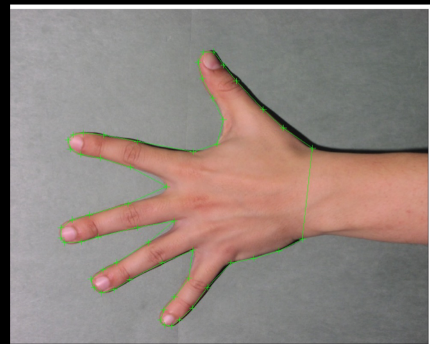
- Similarity measures



Similarity measures

- Anatomical Landmarks
  - time consuming to obtain positions manually
  - Alternative: Joint intensity histogram





Similarity measure: Mean squared difference (MSD)

- Compare difference in intensities.
  - Same similarity measure we used for anatomical landmarks (positions) in Lecture 7
  - Super fast to estimate
- Many local minima's (sub optimal solutions)
  - Intensities are not optimal for this similarity metric

$$\mathrm{MSD}(\boldsymbol{\mu}; I_F, I_M) = \frac{1}{|\Omega_F|} \sum_{\boldsymbol{x}_i \in \Omega_F} \left(I_F(\boldsymbol{x}_i) - I_M(\boldsymbol{T}_{\boldsymbol{\mu}}(\boldsymbol{x}_i)) \right)^2 ,$$

Similarity measure: Cross-correlation

- Cross-correlation of intensities in two images
  - Fast to estimate
- Risk of local minima's (sub optimal solutions)
  - Less robust if image modalities have different intensity histograms
  - Normalise: Reduce the impact of outlier regions

$$\mathrm{NCC}(\boldsymbol{\mu}; I_F, I_M) = \frac{\sum_{\boldsymbol{x}_i \in \Omega_F} \left(I_F(\boldsymbol{x}_i) - \overline{I_F}\right)\left(I_M(\boldsymbol{T}_{\boldsymbol{\mu}}(\boldsymbol{x}_i)) - \overline{I_M}\right)}{\sqrt{\sum_{\boldsymbol{x}_i \in \Omega_F} \left(I_F(\boldsymbol{x}_i) - \overline{I_F}\right)^2 \sum_{\boldsymbol{x}_i \in \Omega_F} \left(I_M(\boldsymbol{T}_{\boldsymbol{\mu}}(\boldsymbol{x}_i)) - \overline{I_M}\right)^2}},$$

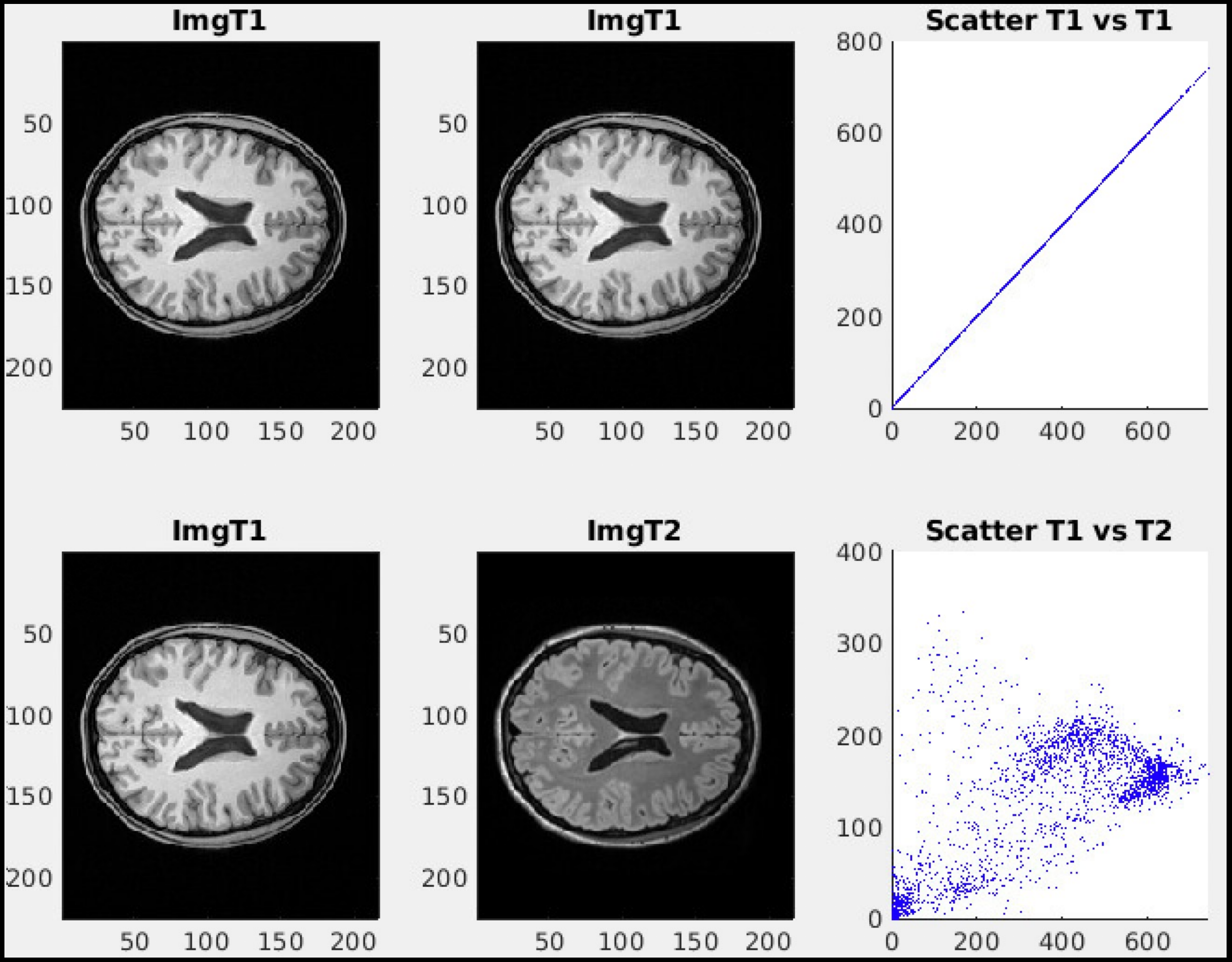
with the average grey-values $\overline{I_F} = \frac{1}{|\Omega_F|} \sum I_F(\boldsymbol{x}_i)$ and $\overline{I_M} = \frac{1}{|\Omega_F|} \sum I_M(\boldsymbol{T}_{\boldsymbol{\mu}}(\boldsymbol{x}_i))$.

Joint intensity histograms

- Perfect registered: Optimal joint intensity agreement

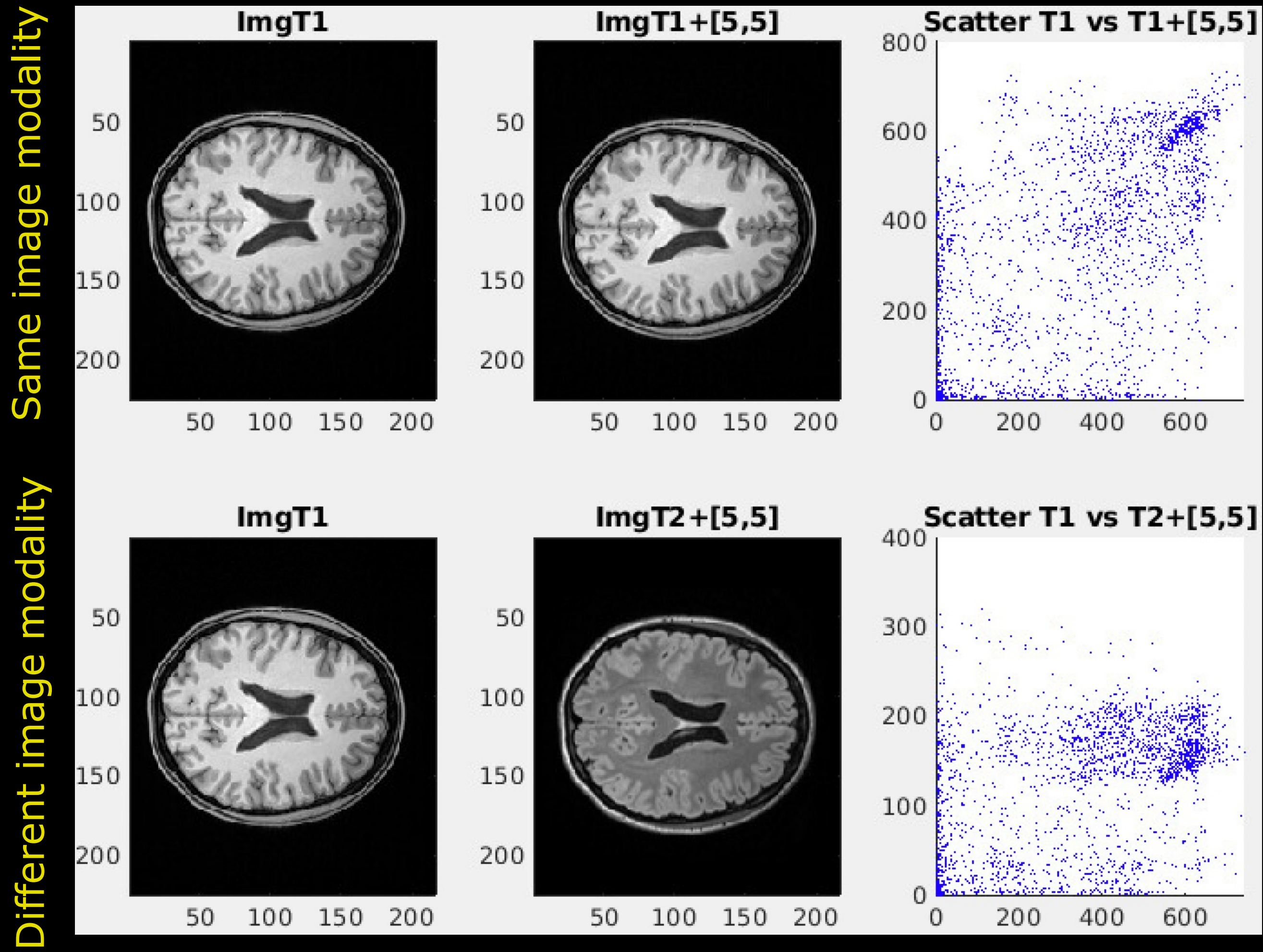
Same image modality

Different image modality



Joint intensity histograms

- Small translation difference: Lower joint intensity agreement

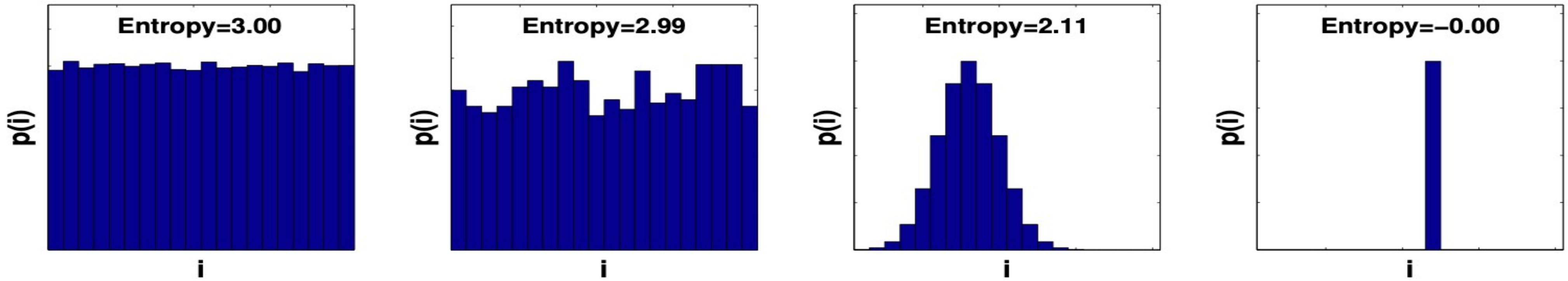


Similarity measure - Entropy

- Comes from information theory.
  - The higher then entropy the more the information content.
- Entropy (Shannon-Weiner):

$$H = -\sum_i p_i \log_b p_i$$

Where *b*: the base of the logarithm
- Bits: *b*=2 and bans: *b*=10
- Entropy is typically in bits i.e. typical used in digital information



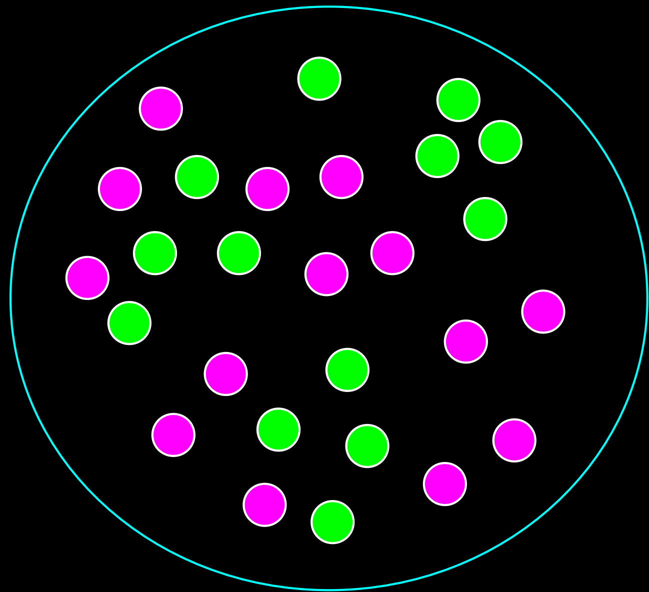
Quiz 2: Highest entropy?

I went to the candy shop and wanted to select the cady mixture that have the highest entropy. Each candy mixture include in total 27 pieces. Which one should I select?

A) Mix 1
B) Make a new choice
C) Contain no liquorice
D) Mix 2
E) It is not healthy

Quiz 3: What is the entropy of the candy mix 1?

Candy mix 1



A) 0.38

B) 0.99

C) 0.45

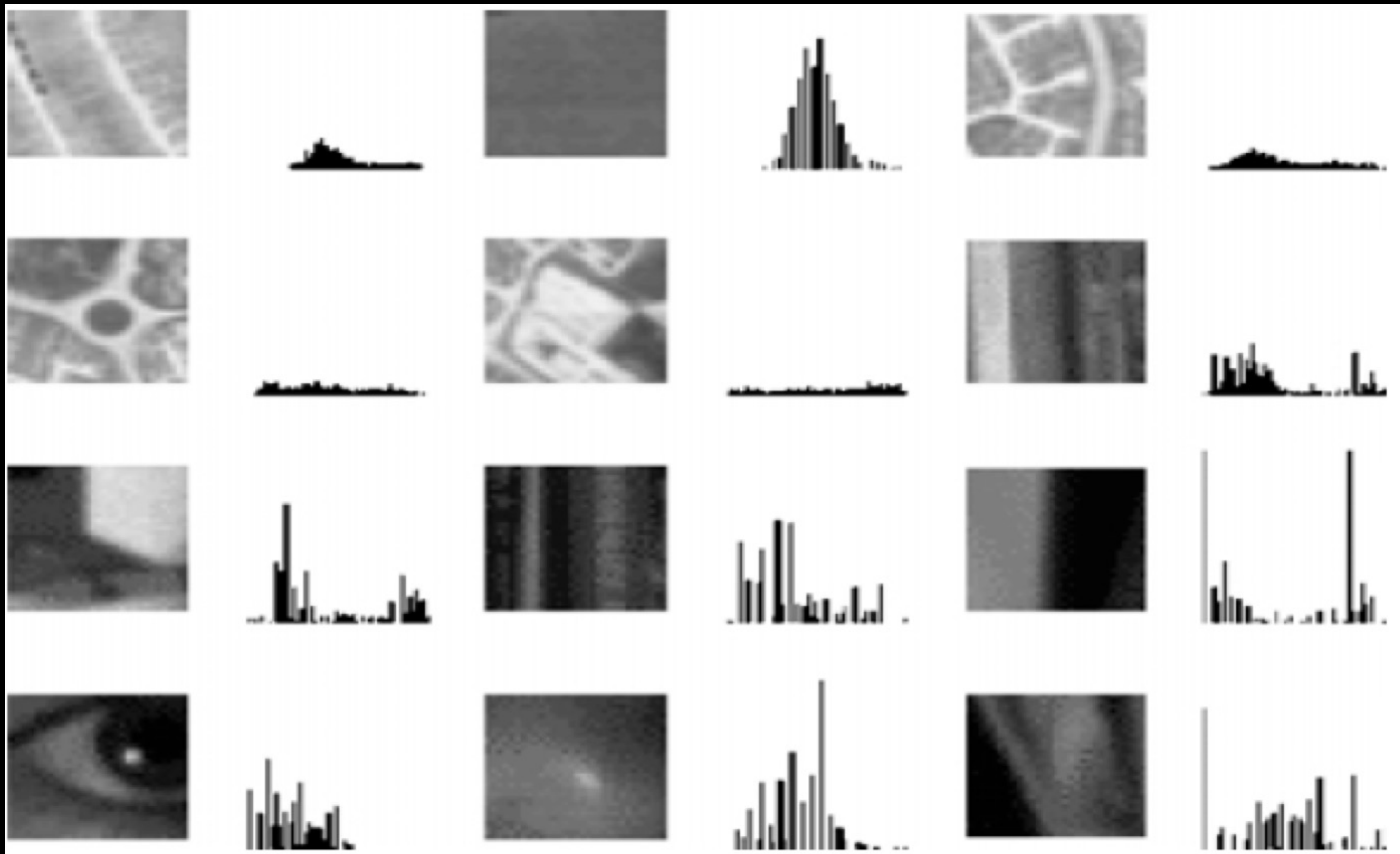
D) 0.23

E) 0.00

SOLUTION:
Green=13
Pink=14
Total=27

pG=13/27
pP=14/27
Entropy= $pG*\log_2(pG)+pP*\log_2(pP)=0.99$

Histograms of images

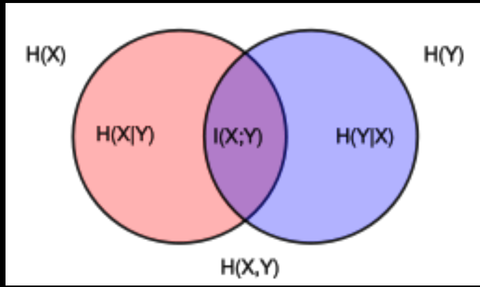


Joint entropy - Mutual information

- Joint entropy $H = -\sum_{X,Y} p_{X,Y} \, log \, p_{X,Y}$

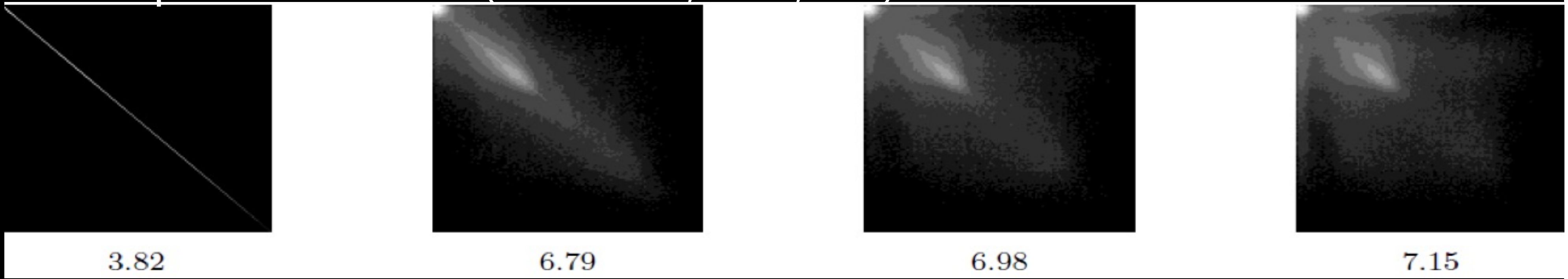
- Similarity measure: The more similar the distributions, the lower the joint entropy compared to the sum of the individual entropies

$$H(X,Y) \leq H(X) + H(Y)$$



en.wikipedia.org/wiki/Mutual_information

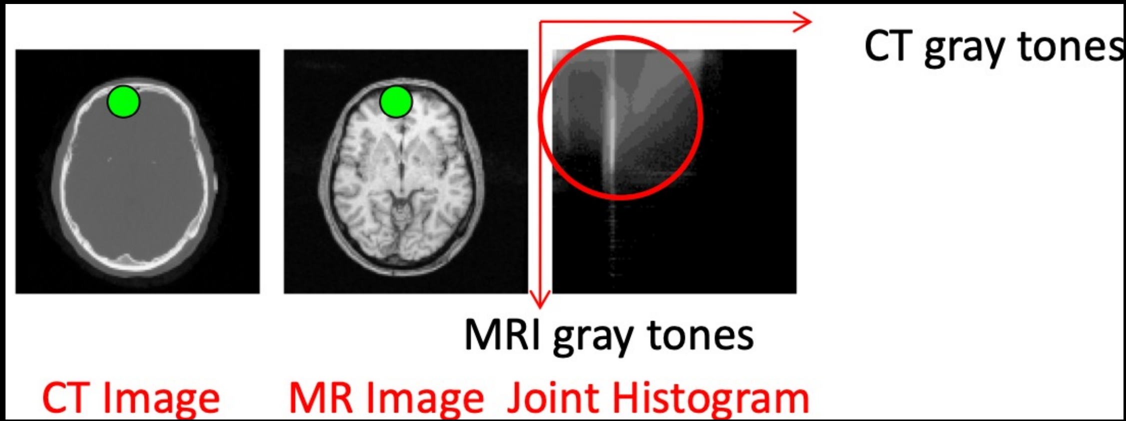
- Example of rotation (Pluim et al., 2003, TMI)



0 degrees 2 degrees 5 degrees 10 degrees

Contrast in joint histograms

- The histogram of the two images must reflect contrast to similar structures for image registration to be successful



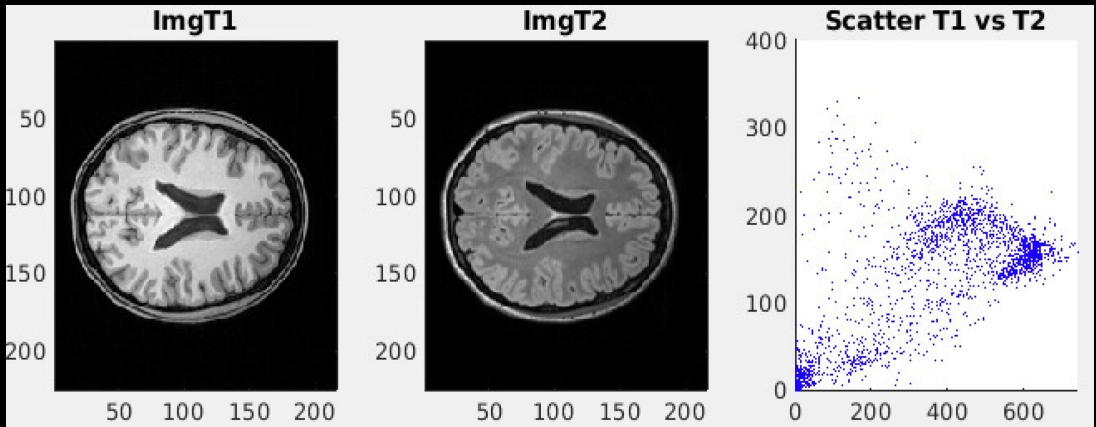
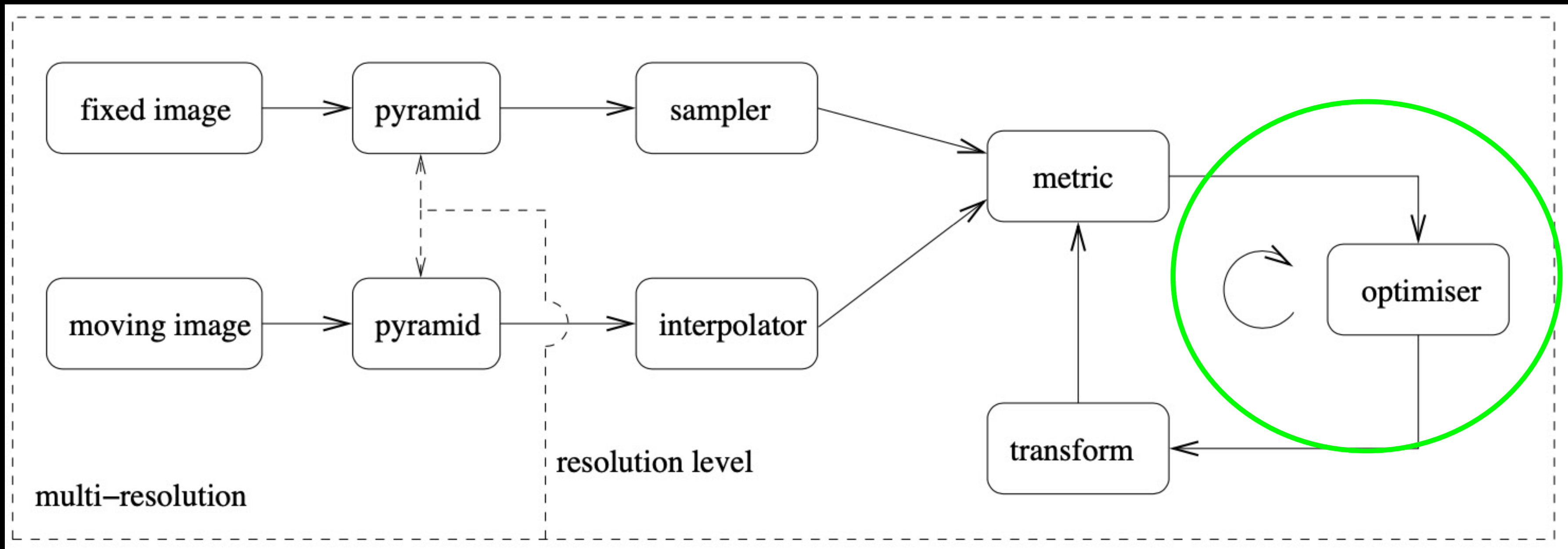


Image Registration pipeline

- The optimiser
  - How to find the transformation parameters?



The optimizer

- We have an objective function describing:
  - A cost function (C) based on a similarity metric
    - Quantifying how well a geometrical transformation ($T(w)$) map an image (Reference/moving, I_M) into an other (Template/fixed, I_F)
- Hence, a good match is a minimum difference:

$$\hat{T}_w = \arg\min_{T_w} \mathrm{C}\left(T_w; I_F, I_M\right)$$

The parameters

$$w \in \mathcal{R}^p$$

parameters

- The parameters is a vector with p elements
- The type of transformation and the dimension of the dataset set the number of parameters
  - Translation p = 2 or 3 (3D)
  - Rotation p = 1 or 3 (3D)
  - Scaling p = 1

Optimization by minimization

- Find the parameter set that minimizes the objective function
- How to find the solution?
  - Analytical: Works fine for landmark registration with few points
  - Numerical: Iterative approaches to search for a solution

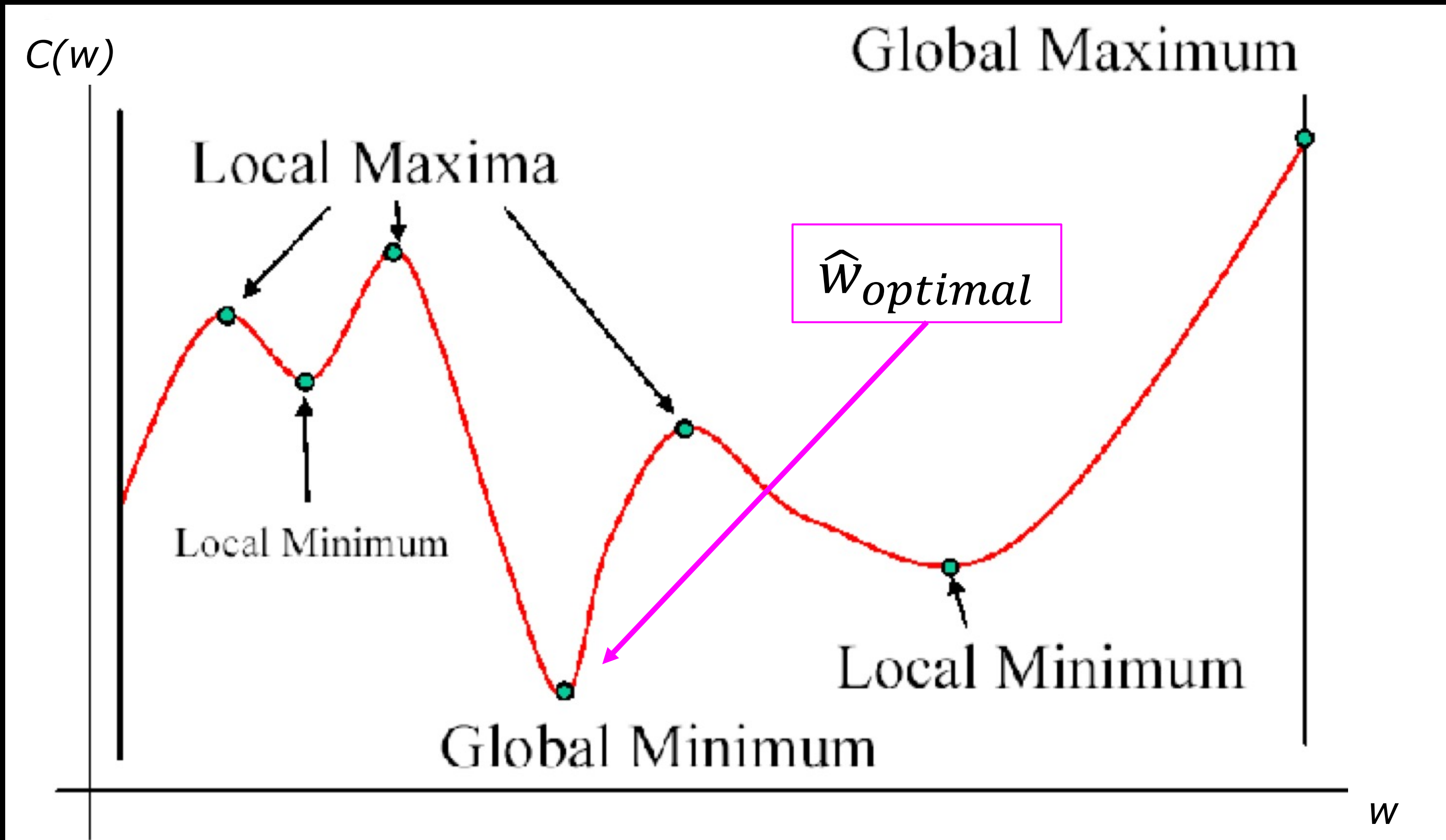
To find: $$\hat{w} = \arg\min_{w} C$$

We simply differentiate w.r.t. *w*:

$$\boxed{\frac{\partial C}{\partial w} = 0}$$

The challenge

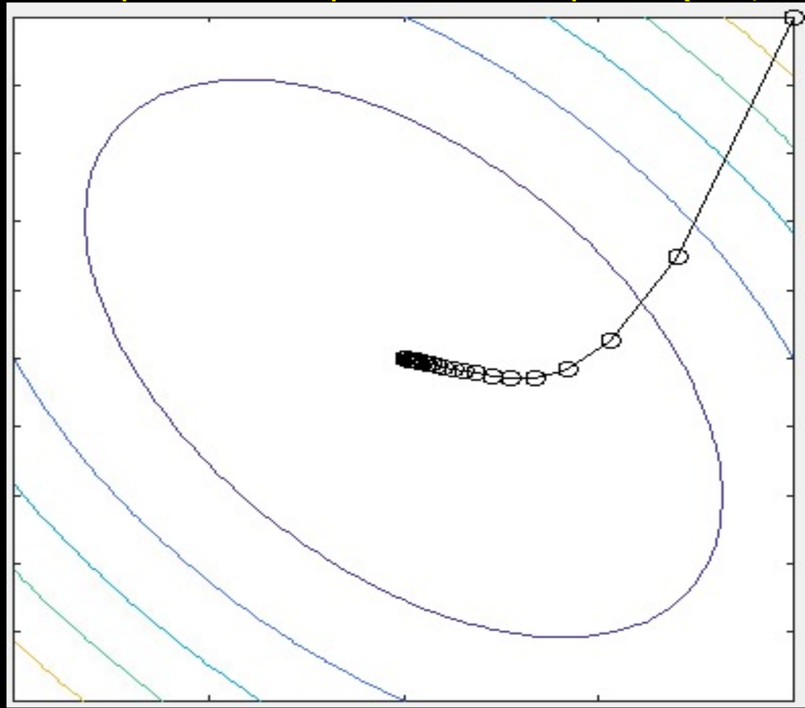
- ***w*** span a p-dimensional space $\boldsymbol{w}=[w_1,w_2,\ldots,w_p]^T$
- Complex parameter space with many data points
  - Finding the lowest place in mountains



Iterative optimisation

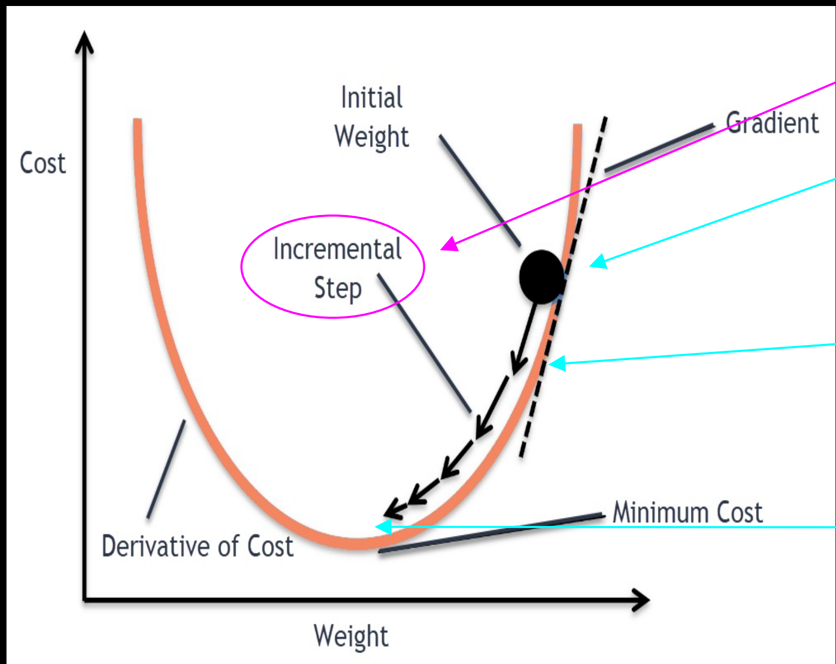
- Aim: Find in parameter space w: $\frac{\partial C}{\partial w} = 0$ i.e. a global minima
  - Search all possible combinations of w? (not a good idea)
  - Systematically search the parameter space = Good idea
- Iterative optimisation strategies
  - Step-wise searching the parameter space
- Many methods exist
  - Gradient based
  - Genetic evolution
  - ...

Contour plot of 2D parameter space (w1,w2)



Gradient descent

- Definition: C($\boldsymbol{w}$) is differentiable in neighbourhood of a point w_n
- C($\boldsymbol{w}$) decreases in the *negative* gradient direction of w_n.
- $w_{n+1} = w_n - \gamma \nabla C(w_n)$
  - $\nabla C(w_n)$: Gradient direction at point w_n
  - γ: Step length --> If small enough: $C(\boldsymbol{w}_n) \geq C(\boldsymbol{w}_{n+1})$



Procedure:

γ 0) Define a step length

$\nabla C(w_0)$ 1) Start guess of a position

2) Find gradient

3) Take a step

$\nabla C(w_1)$ 4) Repeat 2)+3)

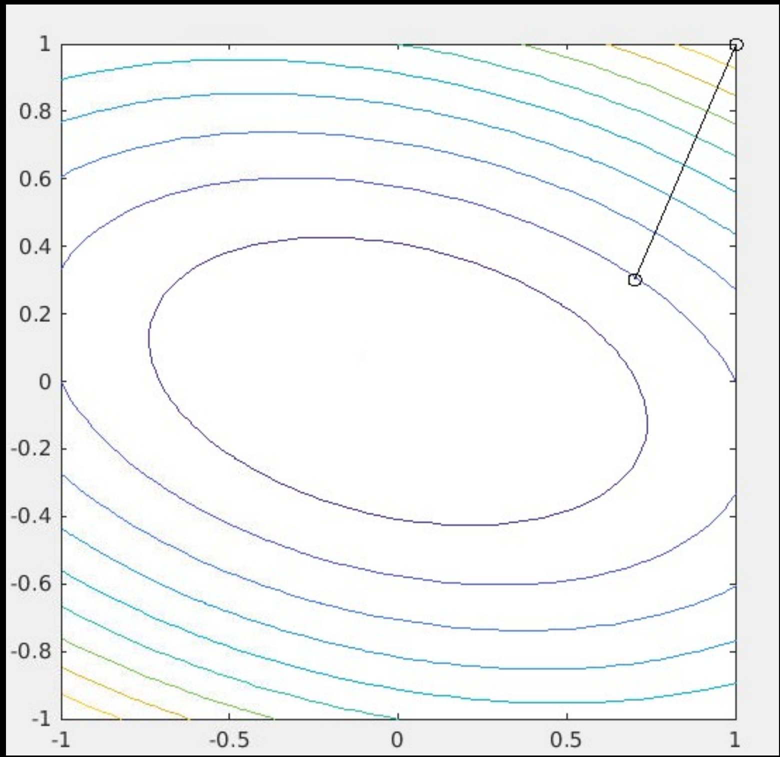
5) Solution: Global minima

$$\nabla C(w_{n+1}) = \frac{\partial C}{\partial w} \approx 0$$

Gradient descent

- Cost function: $\mathrm{C}(\mathrm{x}) = x_1^2 + x_1 x_2 + 3x_1^2$
- Gradient at point x_n: $-\nabla C(x_n) = -\begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$
- Step length: γ=0.1;
- Max steps: 1000
- Start position: $x_0=[1,1]^T$

Iteration:1

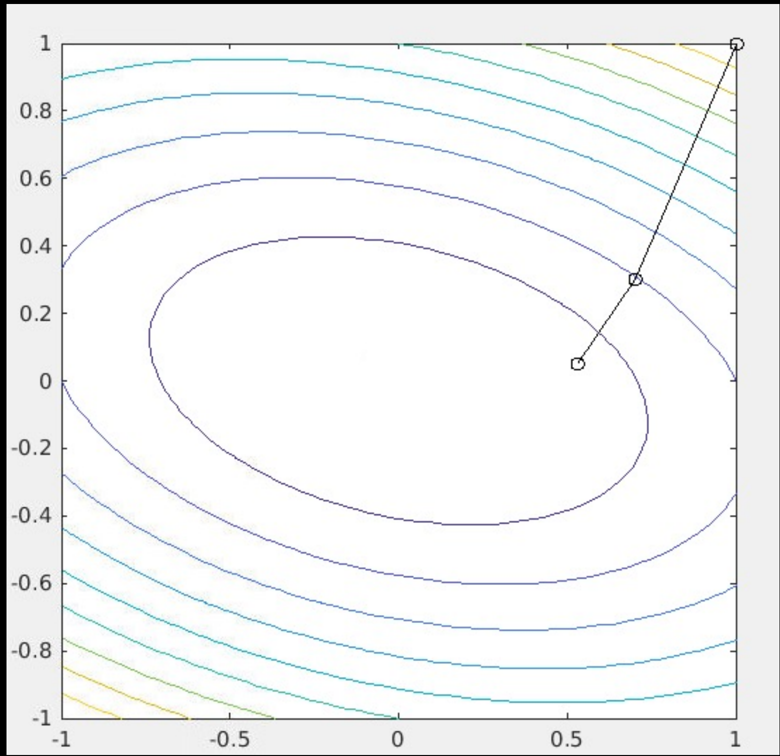


From Matlab function: *grad_descent.m*
By James T. Allison

Gradient descent

- Cost function: $C(x) = x_1^2 + x_1 x_2 + 3x_1^2$
- Gradient at point x_n: $-\nabla C(x_n) = -\begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$
- Step length: γ=0.1;
- Max steps: 1000
- Start position: $x_0=[1,1]^T$

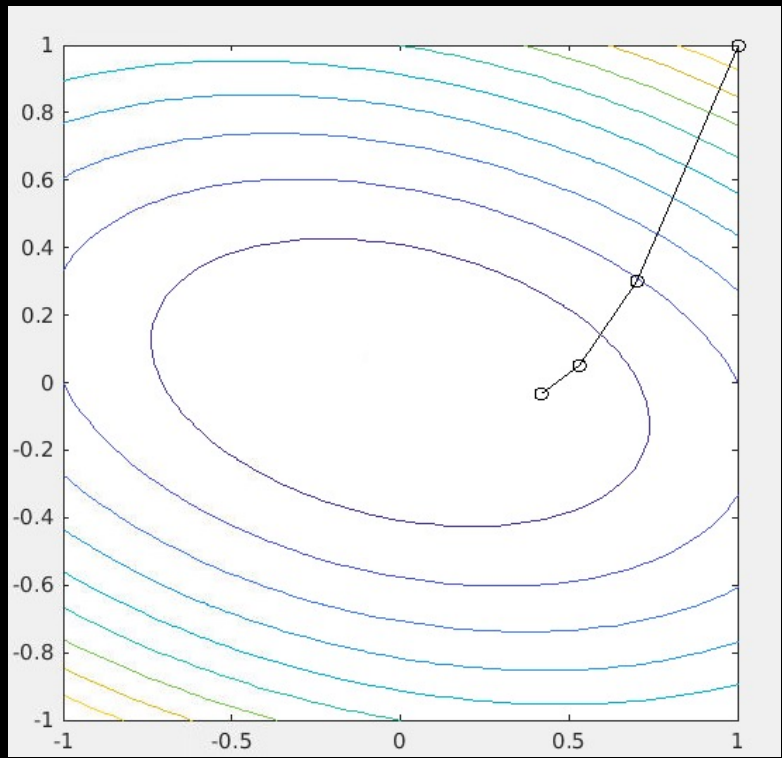
Iteration:2



Gradient descent

- Cost function: $\mathrm{C(x)} = x_1^2 + x_1 x_2 + 3x_1^2$
- Gradient at point x_n: $-\nabla C(x_n) = -\begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$
- Step length: γ=0.1;
- Max steps: 1000
- Start position: $x_0=[1,1]^T$

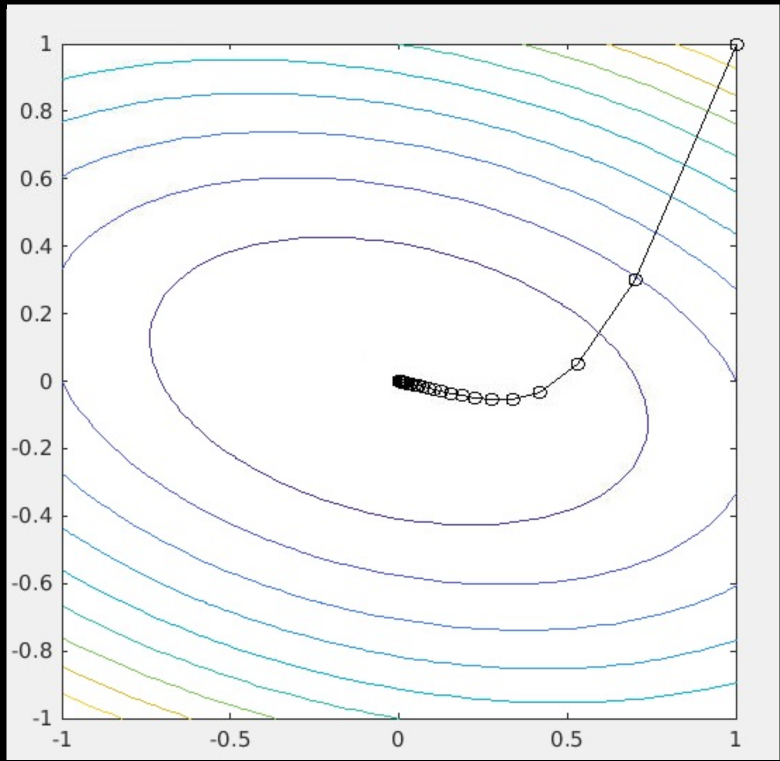
Iteration:3



Gradient descent

- Cost function: $C(x) = x_1^2 + x_1 x_2 + 3x_1^2$
- Gradient at point x_n: $-\nabla C(x_n) = -\begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$
- Step length: γ=0.1;
- Max steps: 1000
- Start position: $x_0=[1,1]^T$

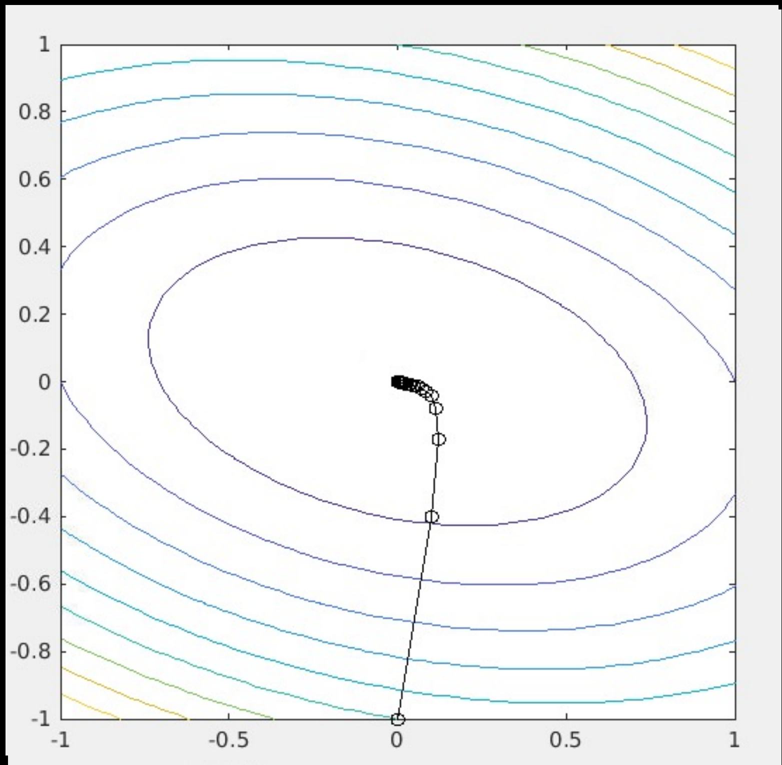
Iteration:37 (final)



Gradient descent

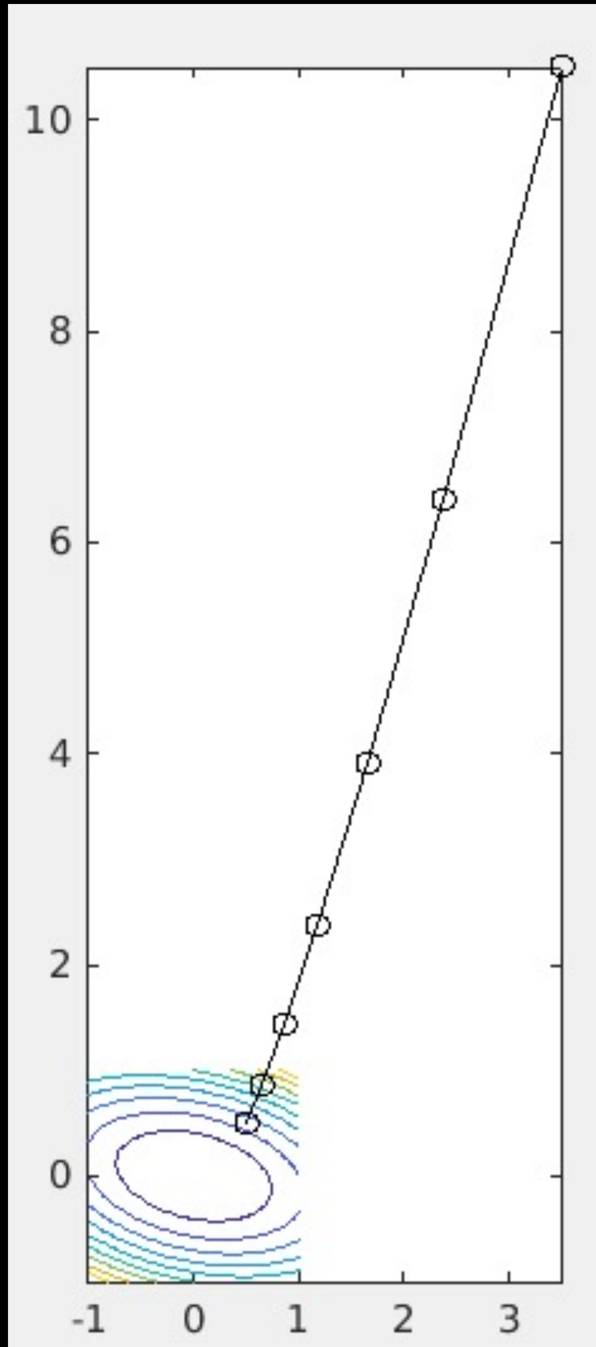
- Cost function: $C(x) = x_1^2 + x_1 x_2 + 3x_1^2$
- Gradient at point x_n: $-\nabla C(x_n) = -\begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$
- Step length: γ=0.1;
- Max steps: 1000
- Start position: $x_0=[0,-1]^T$
- Can find solution from any place
- No local minima's nearby

Iteration:31 (final)



Gradient descent

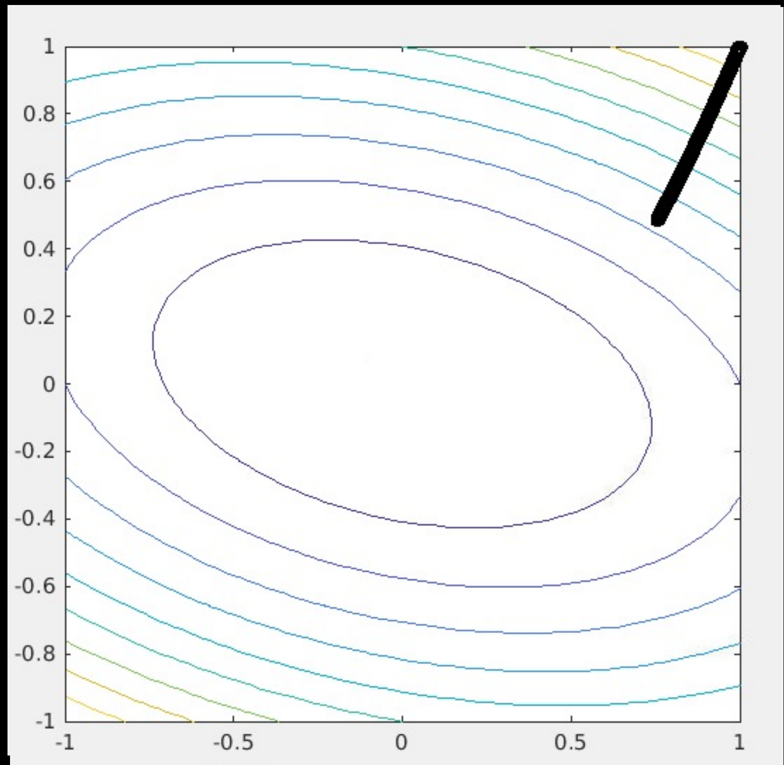
- Cost function: $C(x) = x_1^2 + x_1 x_2 + 3x_1^2$
- Gradient at point x_n: $+\nabla C(x_n) = +\begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$
- Step length: γ=0.1;
- Max steps: 1000
- Start position: $x_0=[0.5,0.5]^T$
- If use positive gradient
  - WRONG DIRECTION!



Gradient descent

- Cost function: $\mathrm{C(x)} = x_1^2 + x_1 x_2 + 3x_1^2$
- Gradient at point x_n: $-\nabla C(x_n) = -\begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$
- Step length: γ=0.0001;
- Max steps: 1000
- Start position: $x_0 = [1,1]^T$
- Too small step size –many steps
- Do not find a solution

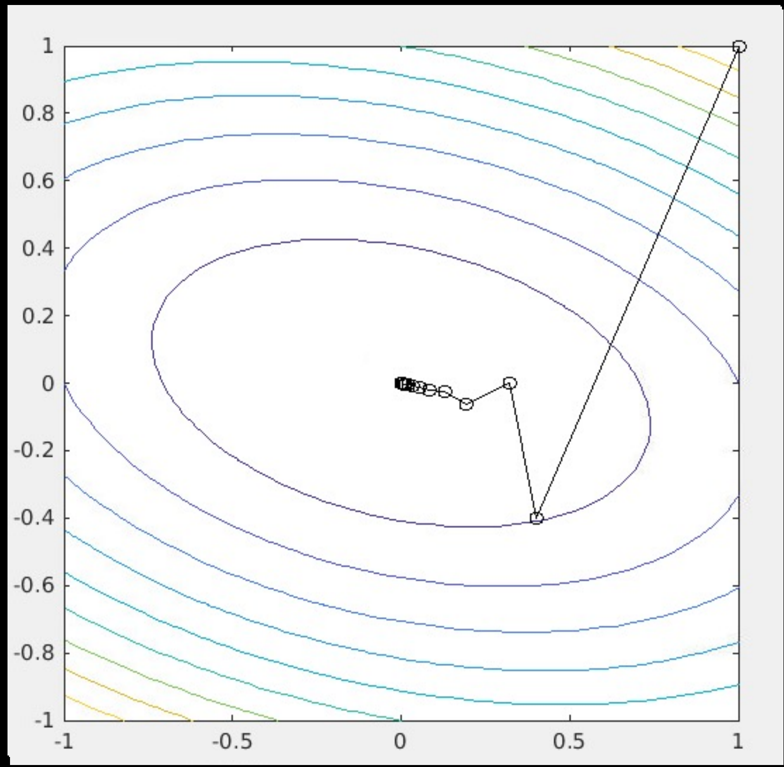
Iteration:1000 (final)



Gradient descent

- Cost function: $C(x) = x_1^2 + x_1 x_2 + 3x_1^2$
- Gradient at point x_n: $-\nabla C(x_n) = -\begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$
- Step length: γ=0.2 (optimal)
- Max steps: 1000
- Start position: $x_0=[1,1]^T$
- Few steps: Optimal step size

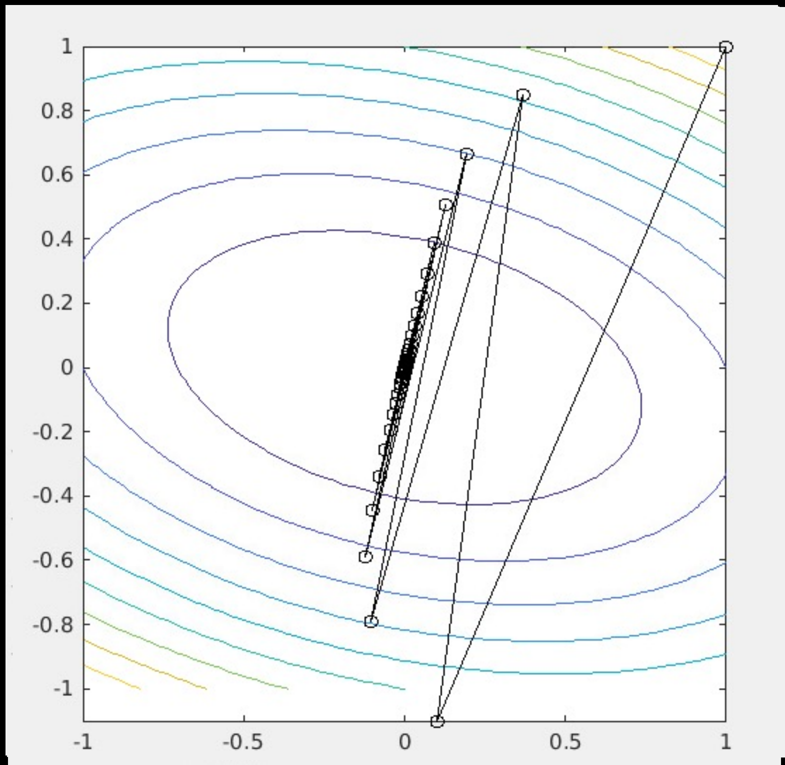
Iteration:17 (final)



Gradient descent

- Cost function: $C(\mathrm{x}) = x_1^2 + x_1 x_2 + 3x_1^2$
- Gradient at point x_n: $-\nabla C(x_n) = -\begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$
- Step length: γ=0.3
- Max steps: 1000
- Start position: $x_0=[1,1]^T$
- Too large step size – unstable
- Sensitive to local minima’s
- Solution: Dynamic step length

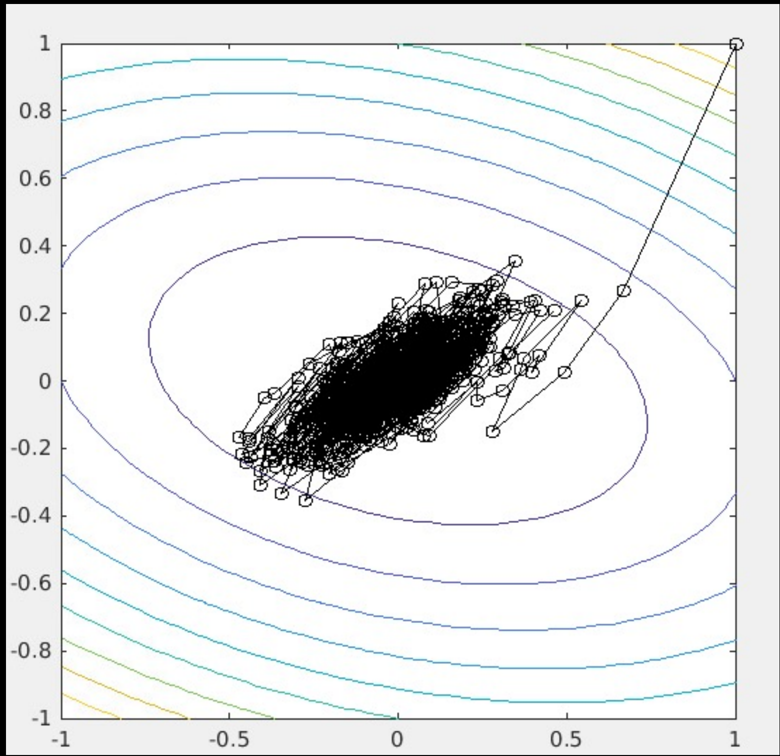
Iteration:65 (final)



Gradient descent

- Cost function: $C(x) = x_1^2 + x_1x_2 + 3x_1^2$
- Gradient at point x_n: $-\nabla C(x_n) = -\begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$
- Step length: γ=0.1
- Max steps: 1000
- Start position: $x_0=[1,1]^T$
- Noisy data: Cannot find optimum

Iteration:1000 (final)



Quiz 4: What is the updated position xnew?

Model fitting uses an a cost function: $C(x) = x_1^2 + x_1x_2 + 3x_2^2$
and an iterative optimizer: Gradient descent with step length of 0.2

What is the new position of xnew $=[?,?]^T$ after one step from position $x=[1, 0]^T$?

A) $[0.3,2.3]^T$
B) $[-1.7,0.3]^T$
C) [1.4,0.2]T
D) $[0.6,-0.2]^T$
E) $[5.2,2.2]^T$

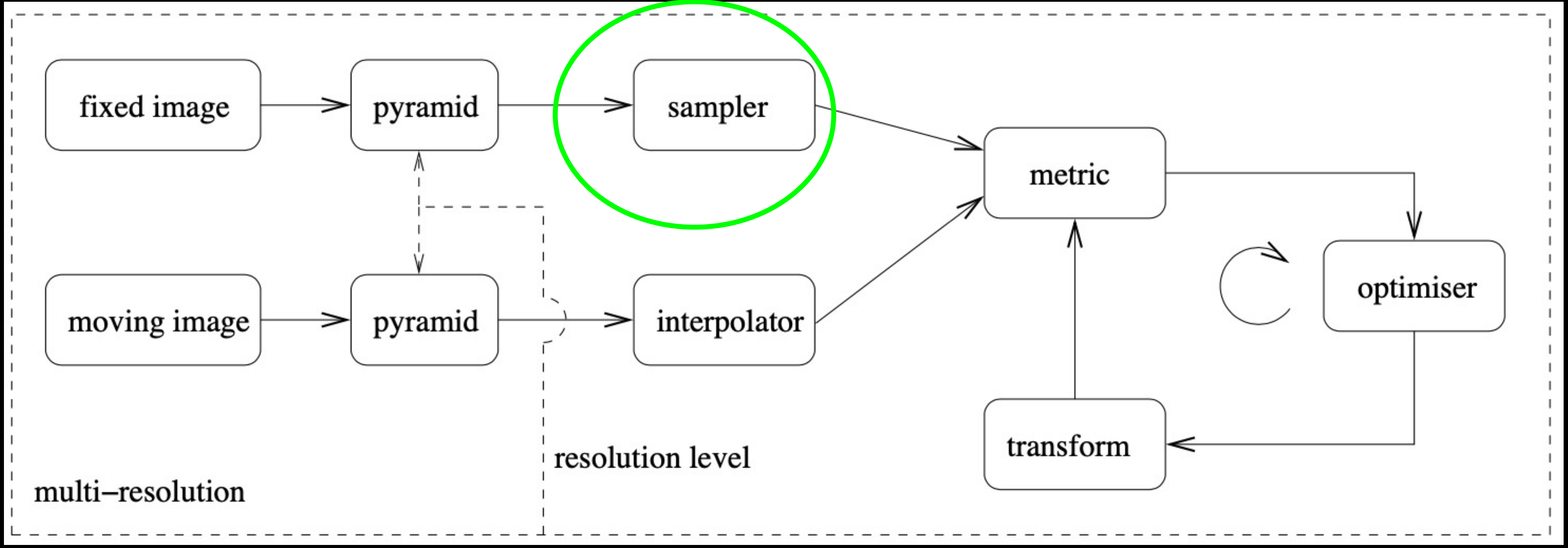
Solution:
1) Calculate the gradient for $x=[1,0]^T$
- differentiate C: $\nabla C(x) = \begin{bmatrix} 2x_1 + x_2 \\ x_1 + 6x_2 \end{bmatrix}$

$\nabla C([1,0]^T) = [2,1]^T$

2) Update the step: $x_{new}=x - \nabla C$*stepLength
- $xnew=[1,0]^T-0.2*[2,1]^T=[0.6, -0.2]^T$

Image Registration pipeline

- The sampler
  - How many data points for a robust similarity measure?

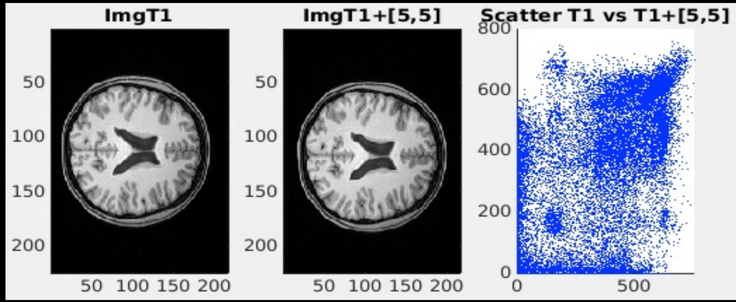


The sampler

- Calculating the similarity metrics:
  - Summing over all pixels/voxels in an image is VERY time consuming
- Selecting a sparse sampling strategy
  - Reducing CPU load and reduce memory load when
  - Efficient selection of image points

with translation

All samples



The sampler

- Sparser sampling: Similar scatter plot
  - Define a good compromise (sample the whole image)
- Ordered vs Random
  - Spatial dependency: Dependent on large homogeneous structures
  - Very sparse sampling: Risk not sampling small structures

Ordered | Random

Every 2nd

Every 10th

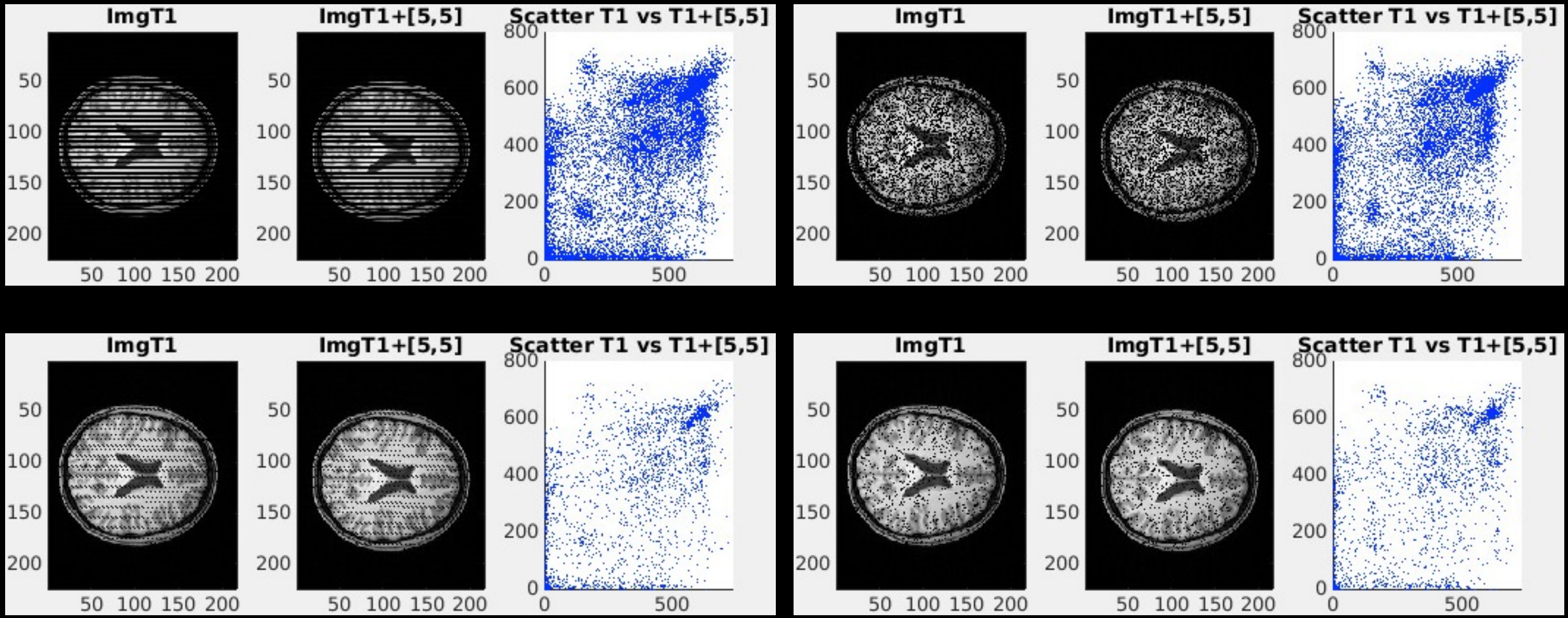
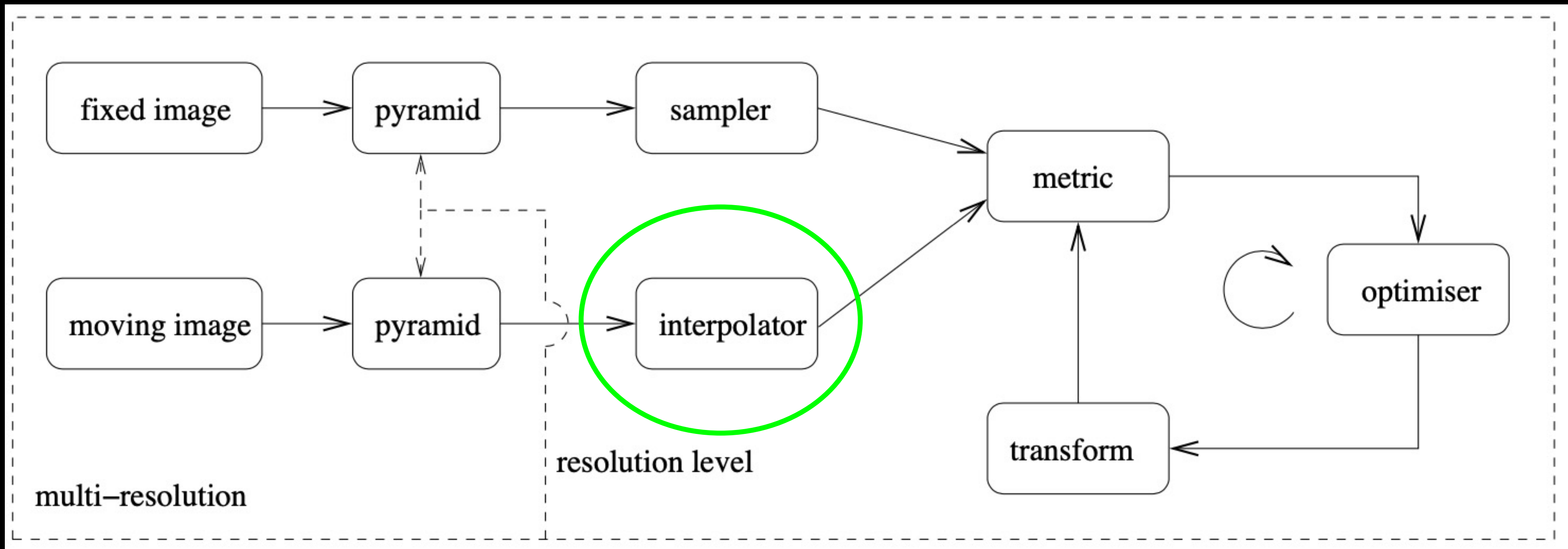


Image Registration pipeline

- Interpolation
  - To map the intensities from the template image to the grid of the reference image via a transformation matrix

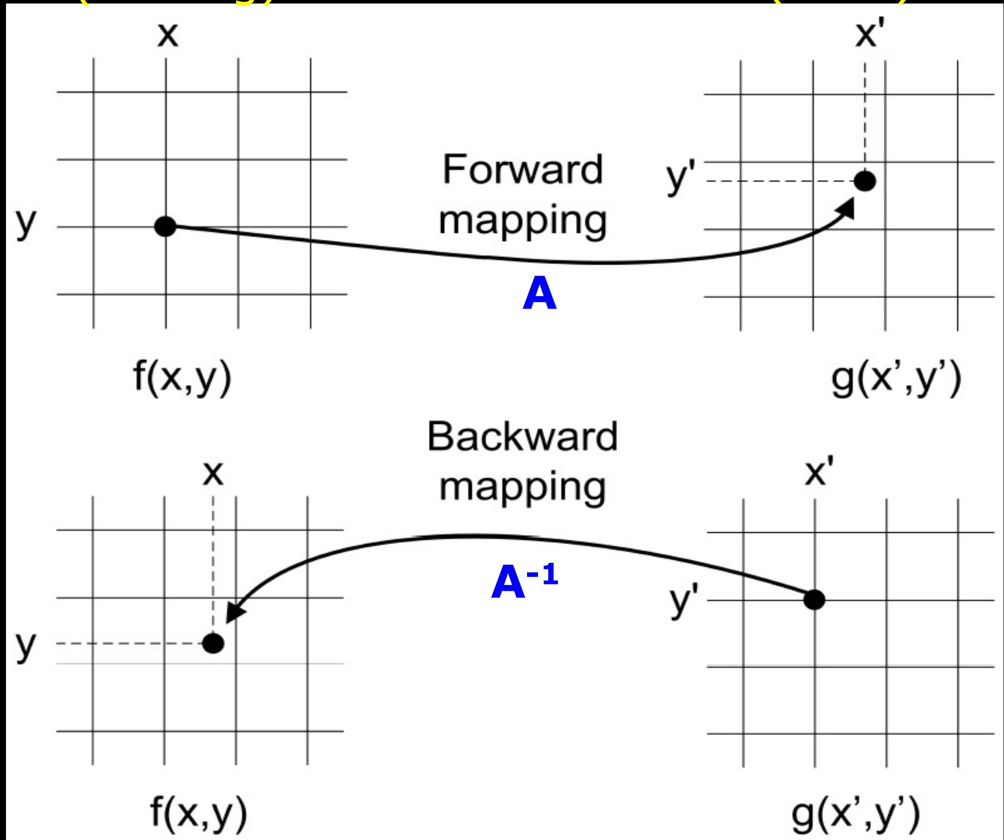


A FLASH BACK to Lecture 7:
Forward vs Backward mapping

- In a nut shell
  - Going backward we need to invers the transformation

Template (moving)

Reference (fixed)



Interpolation methods

- Enhances structural boundaries
  - Higher-order interpolation methods: Reduce blurring
- May visually appear “sharper”
  - Do not change image information!
  - Only if combining interpolated images w. different information of the same object
  - Different angles of moving object e.g. car

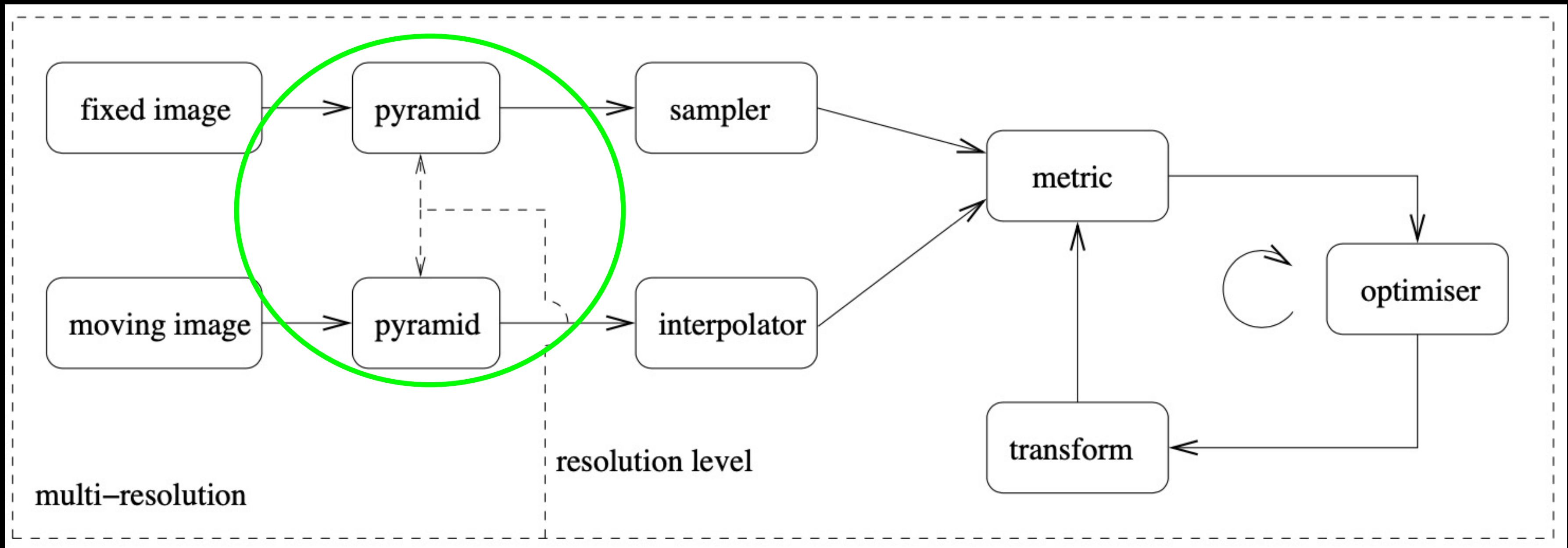
    → Super resolution (another topic)

(a) (b) (c) (d) (e)

Figure 2.4: Interpolation. (a) nearest neighbour, (b) linear, (c) B-spline $N = 2$, (d) B-spline $N = 3$, (e) B-spline $N = 5$.

Image Registration pipeline

- Pyramid



The Pyramid Principle

- To ensure robust image registration

Some stones?

Pretty close

Walking distance

From a bird

From space?

Very detailed

Good overview

Too coarse

The Pyramid Principle

- To ensure robust image registration

Some stones?

Pretty close

Walking distance

From a bird

From space?

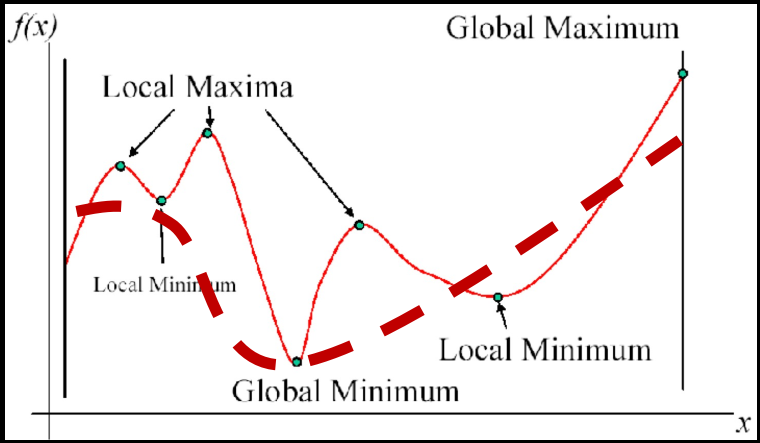
Very detailed

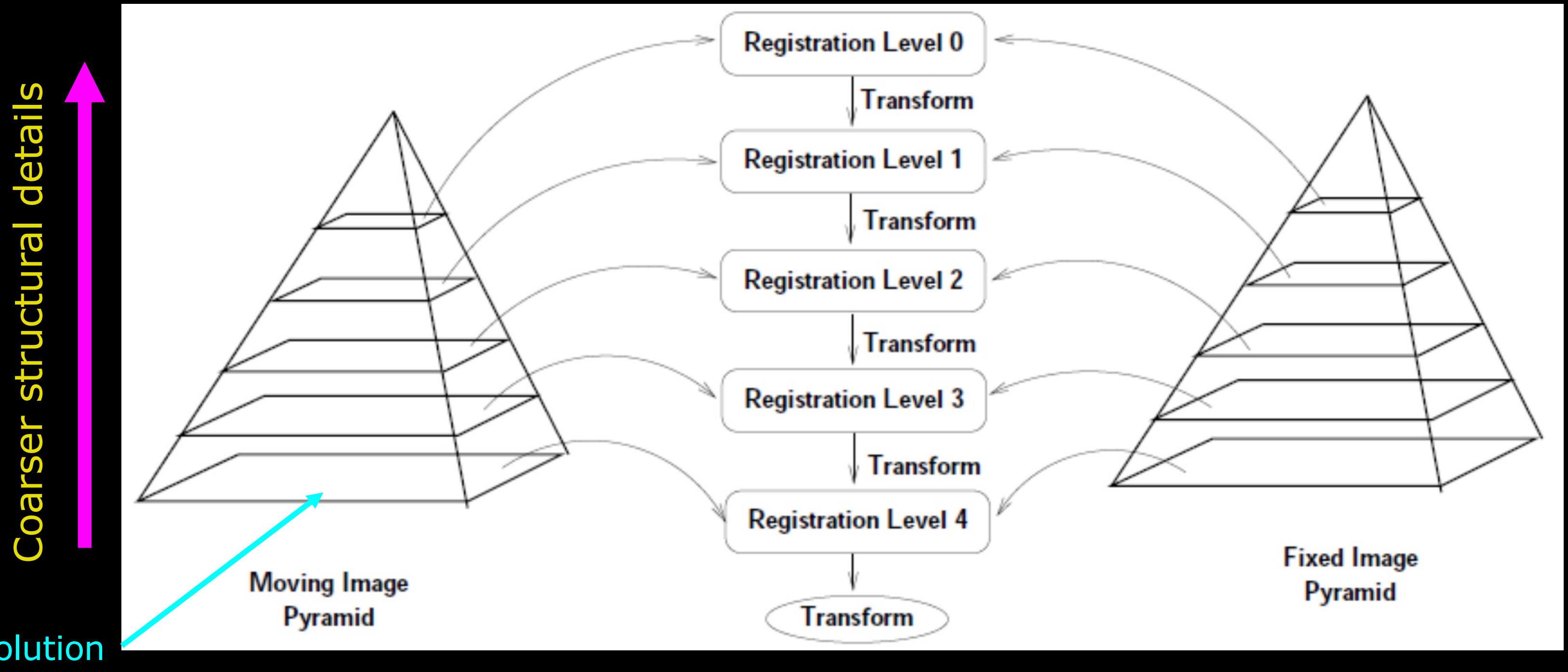
Good overview

Too coarse

The Pyramid Principle

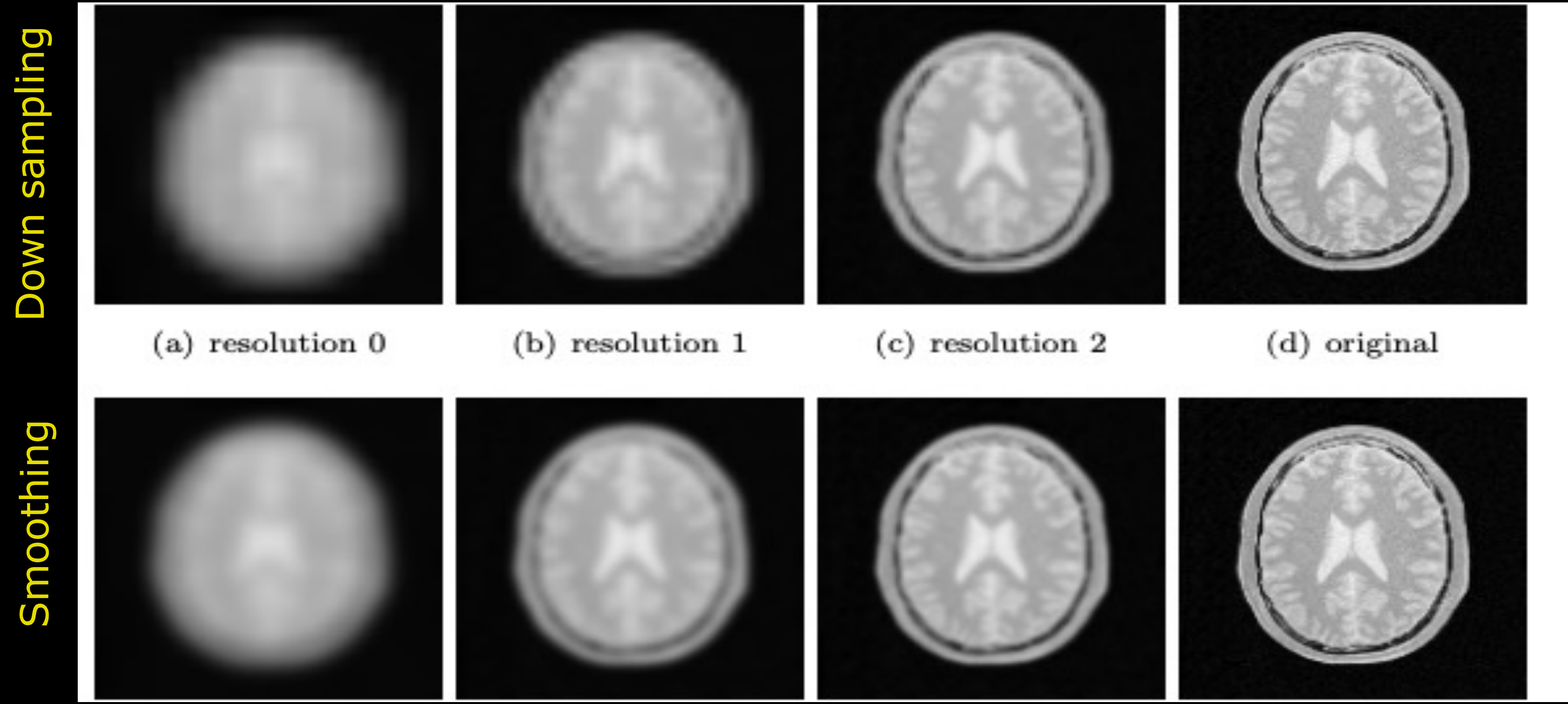
- A Multi-resolution strategy
- To ensure robust image registration
  - To reduce local minima's
  - What is a prober image resolution level ?





The Pyramid Principle

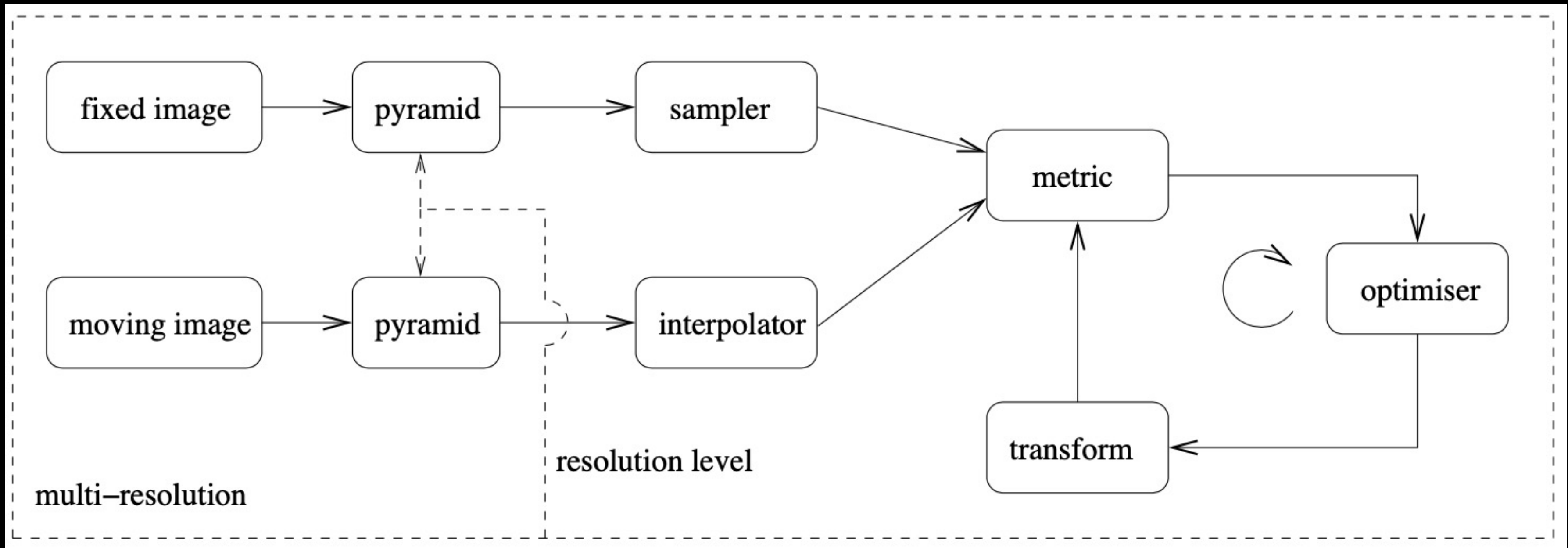
- Lower image resolution
  - Down sampling (memory reduction, fewer data)
- Less structural details
  - Smoothing (Complex method settings become more general)



(a) resolution 0 (b) resolution 1 (c) resolution 2 (d) original

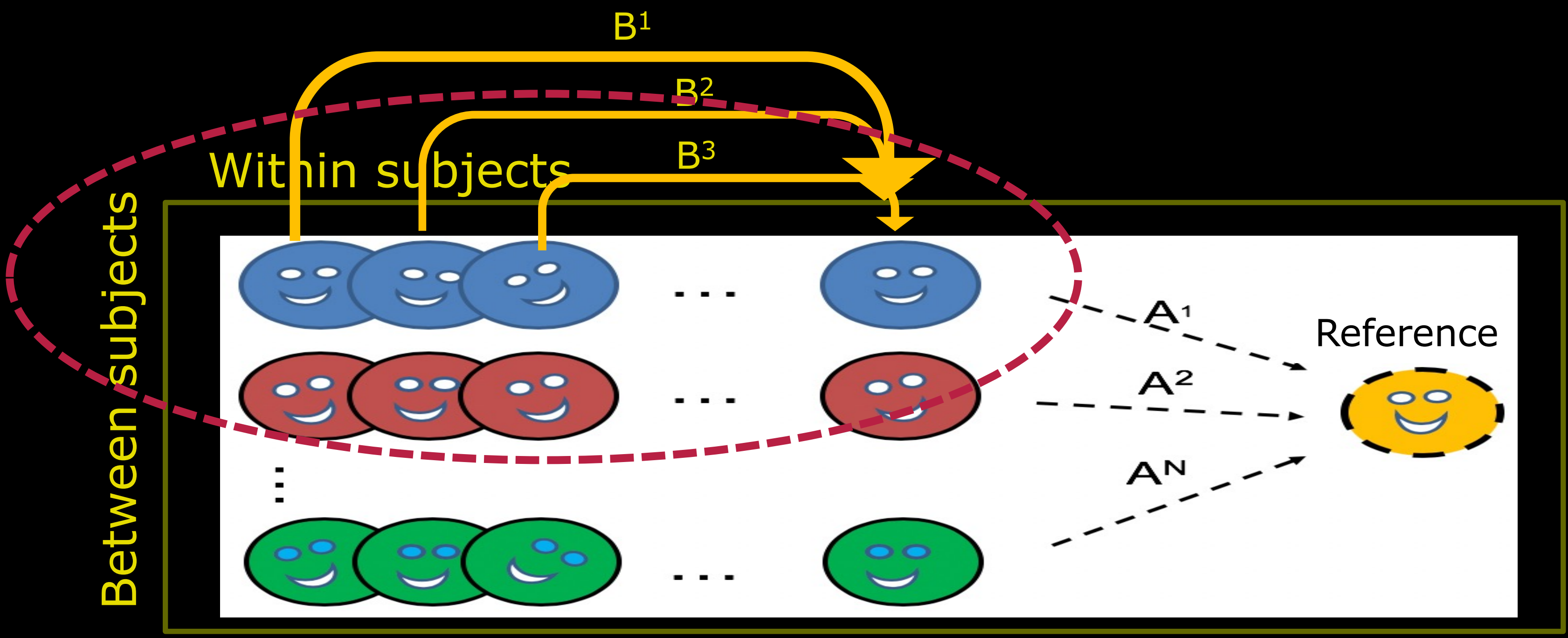
Image Registration pipeline

- At the end we just select an existing tool
- Still, we need how too select method settings ☺
  - This was the first step in the registration pipeline



Combining Image Registration pipelines

- First step : Within subjects (Same structure + temporal)
- Second step: Between subjects (different structure+ temporal)
  - Can use an iterative procedure to improve registration
- Combine subject-wise transformation metrics by multiplication
  - Apply only one interpolation at the end to minimise blurring



Quiz 5: Quality inspection - How

How to quality assurance (QA) the image registration results?

A) Use a similarity measure
B) Visual inspection
C) No need it to - just works
D) Sum of square difference
E) Search the internet for experience

Image Registration pipeline strategy

- Within subjects and between challenges
  - E.g. Histology 2D → 3D: Structural difference between slices
  - Visually inspect your results!!

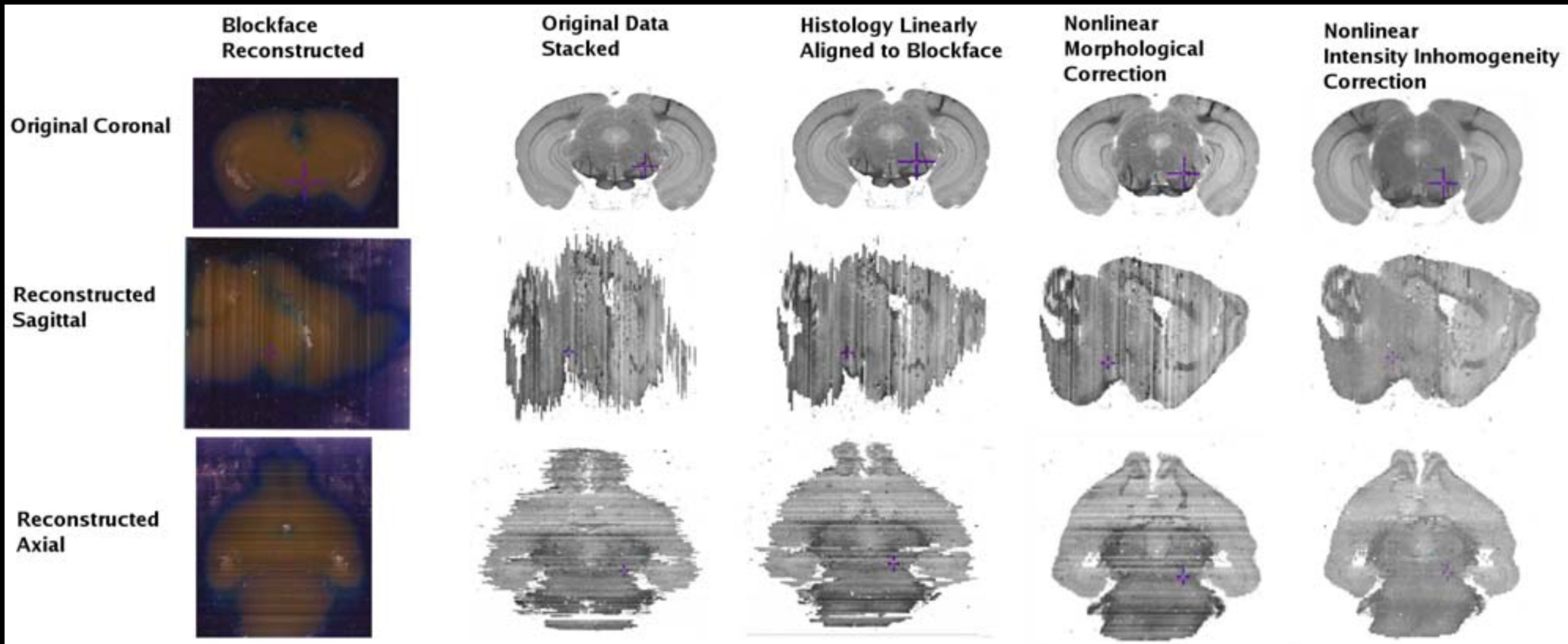
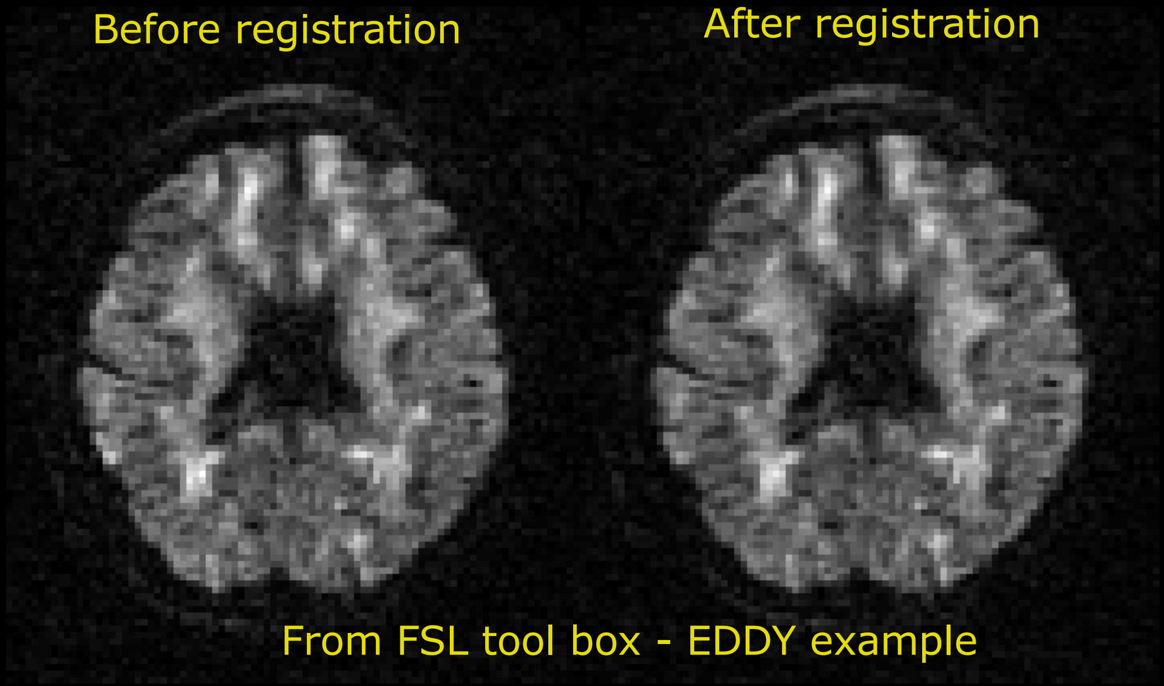


Image Registration pipeline strategy

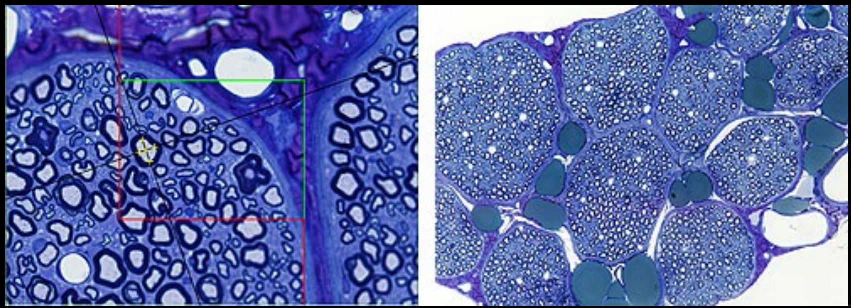
- Within subjects across time points (temporal)
  - Remove image distortions + subjection motion
- Visually inspect your results!!



What did you learn today?

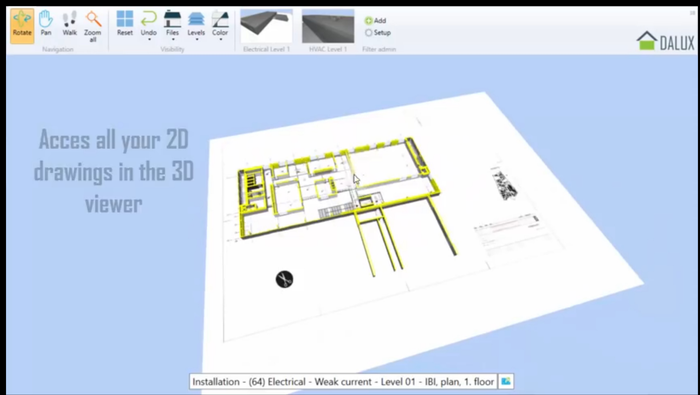
- Describe difference between a pixel and voxel
- Describe the general image-to-image registration pipeline
- Describe 3D geometrical affine transformations
- Choose a suitable similarity metric given the image modalities to register
- Compute the normalized correlation coefficient (NNC) between two images
- Compute Entropy
- Describe the concept of iterative optimizers
- Compute steps in the gradient descent optimization steps
- Describe the pyramidal principle for multi-resolution strategies
- Select a relevant registration strategy: 2D to 3D, Within- and between objects and moving images

Next week – Company presentations

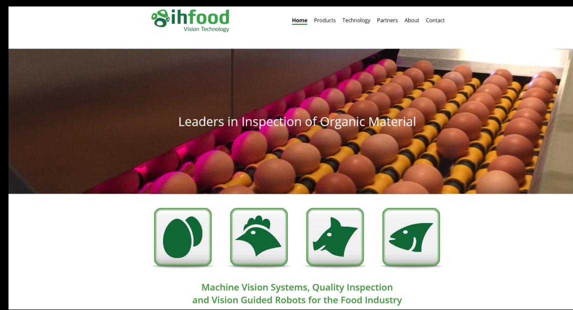


Visiopharm

JLI Vision



Dalux



IH Food

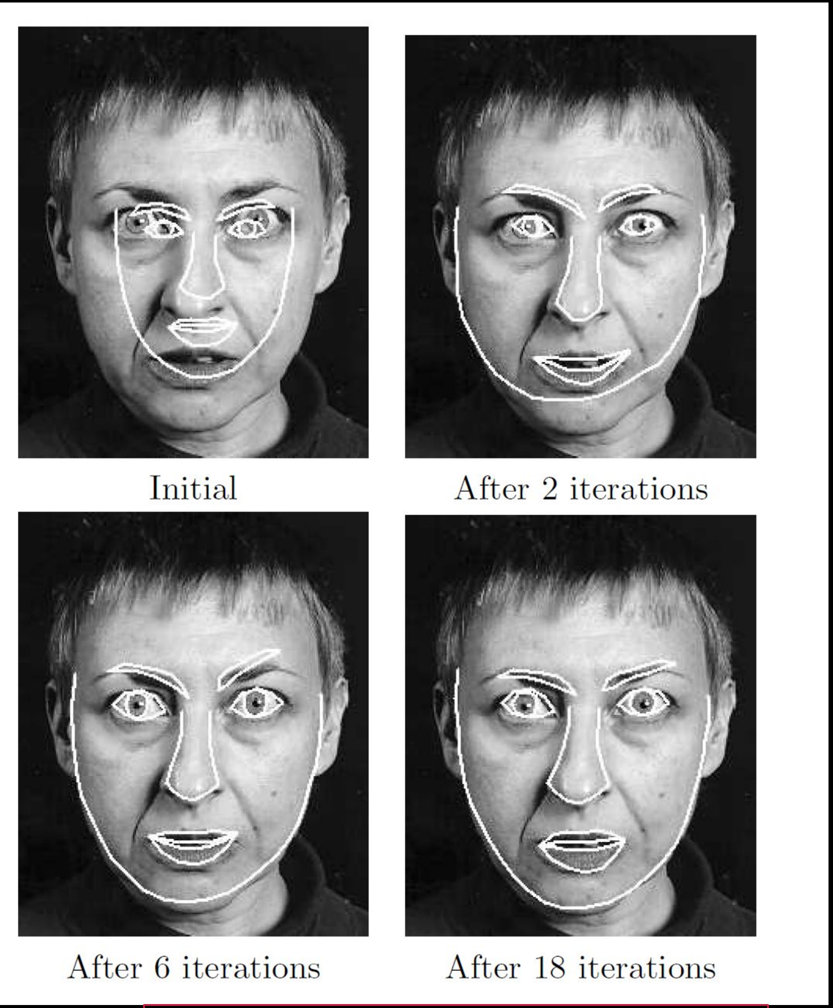
Image Analysis

Rasmus R. Paulsen
Tim B. Dyrby
DTU Compute

rapa@dtu.dk

http://courses.compute.dtu.dk/02502

Lecture 12 – Active shape models

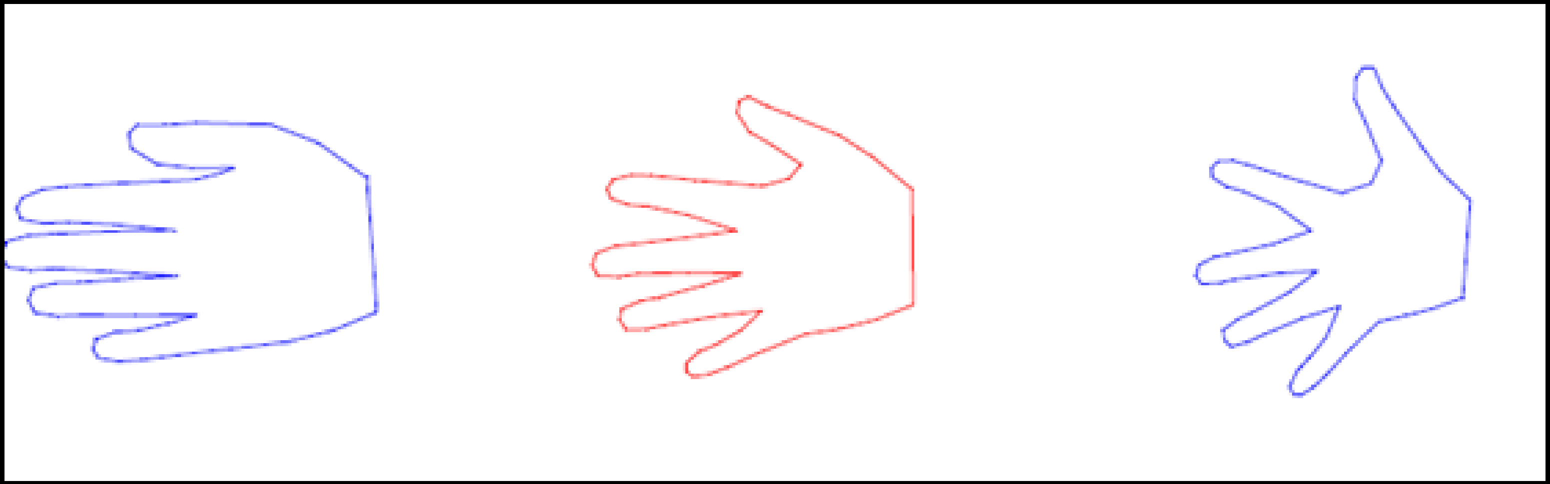


Tim Cootes: Active shape models

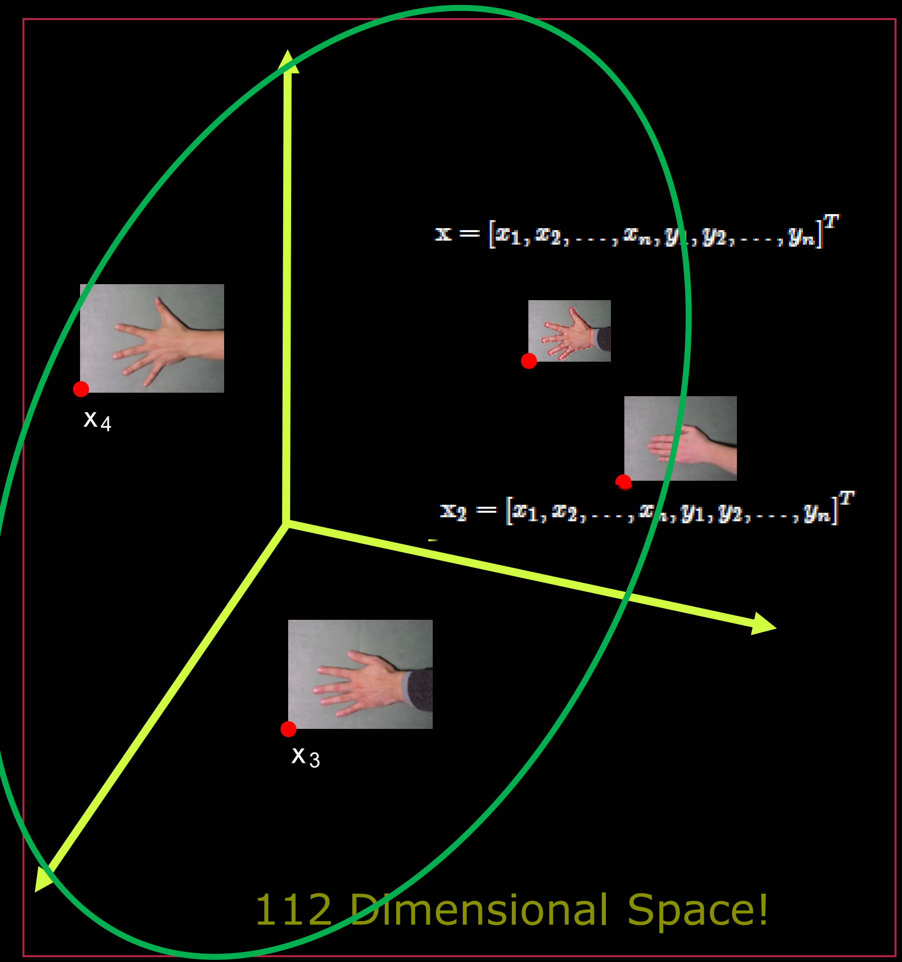
Today's Learning Objectives

- Describe how shapes can be synthesized using the shape space
- Describe the generative model based on a statistical shape model
- Describe the concept of analysis by synthesis
- Describe how the Eigenvectors and Eigenvalues can be used to constrain a shape model
- Describe how a statistical shape model can be fitted using the gradients in an image
- Describe how a statistical shape model can be fitted by modelling local variation
- Explain the problem of strong priors in statistical models

We have a statistical model of shape

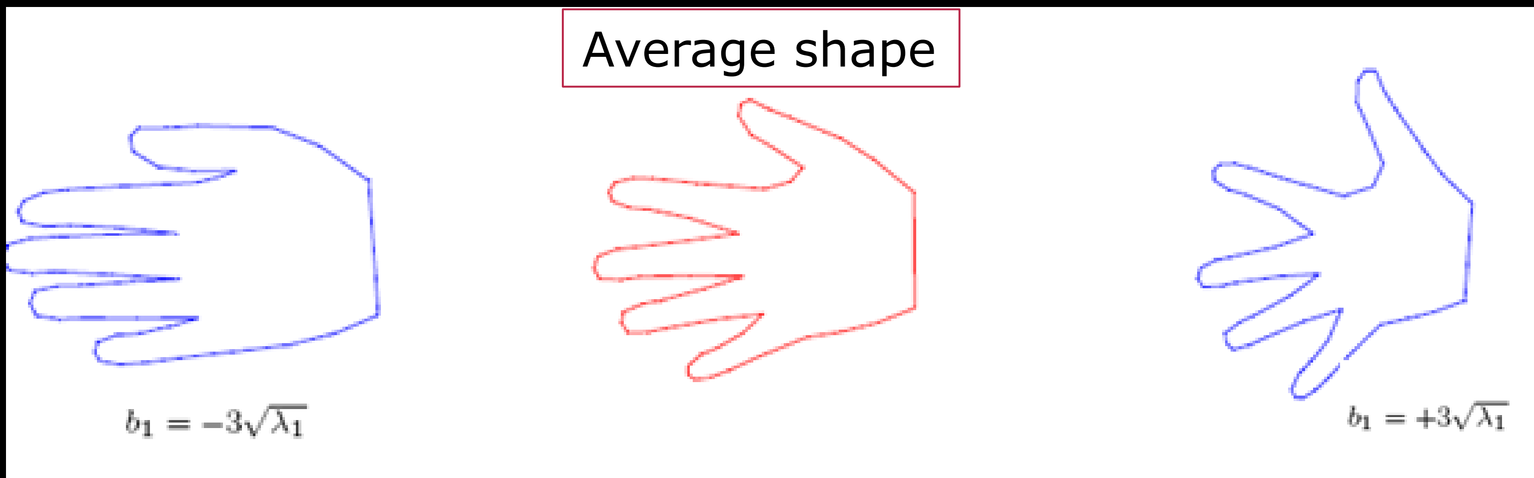


Shape space



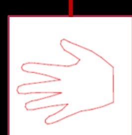
- A mapping of the shape space
- PCA based description of the "hand space"

Synthezising new shapes

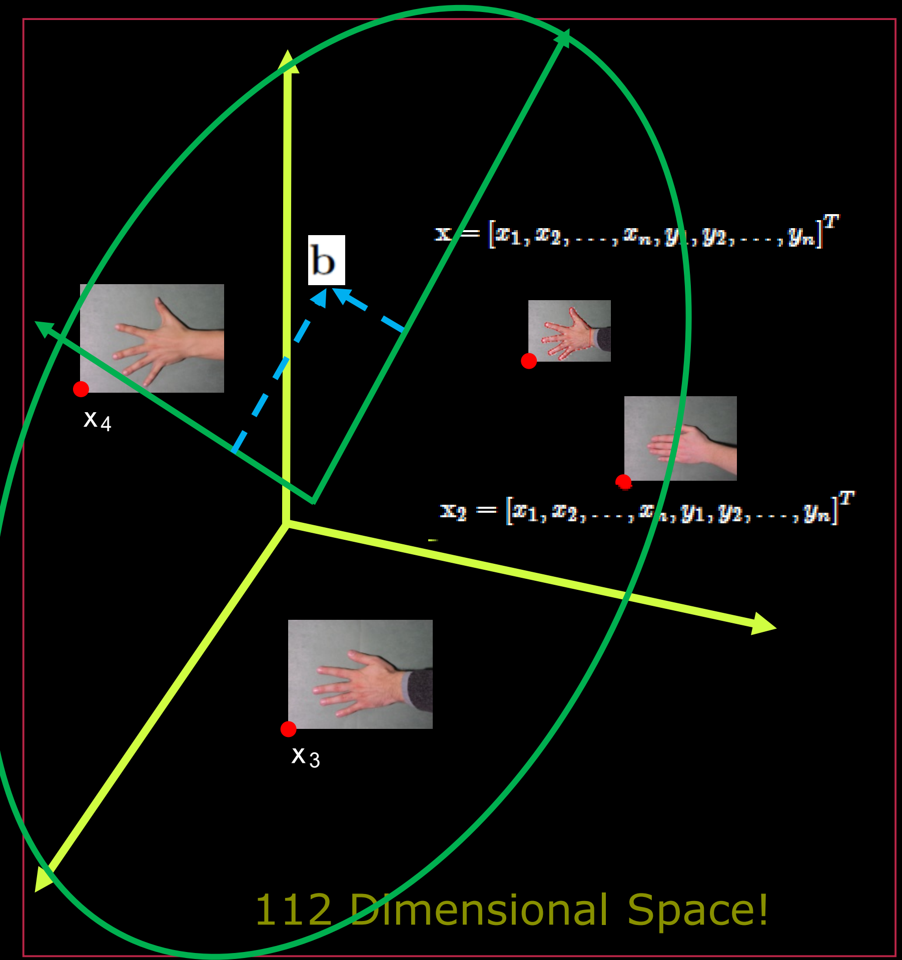


$$\mathbf{x} \approx \bar{\mathbf{x}} + \boldsymbol{\Phi}\mathbf{b}$$

Φ contains the t eigenvectors



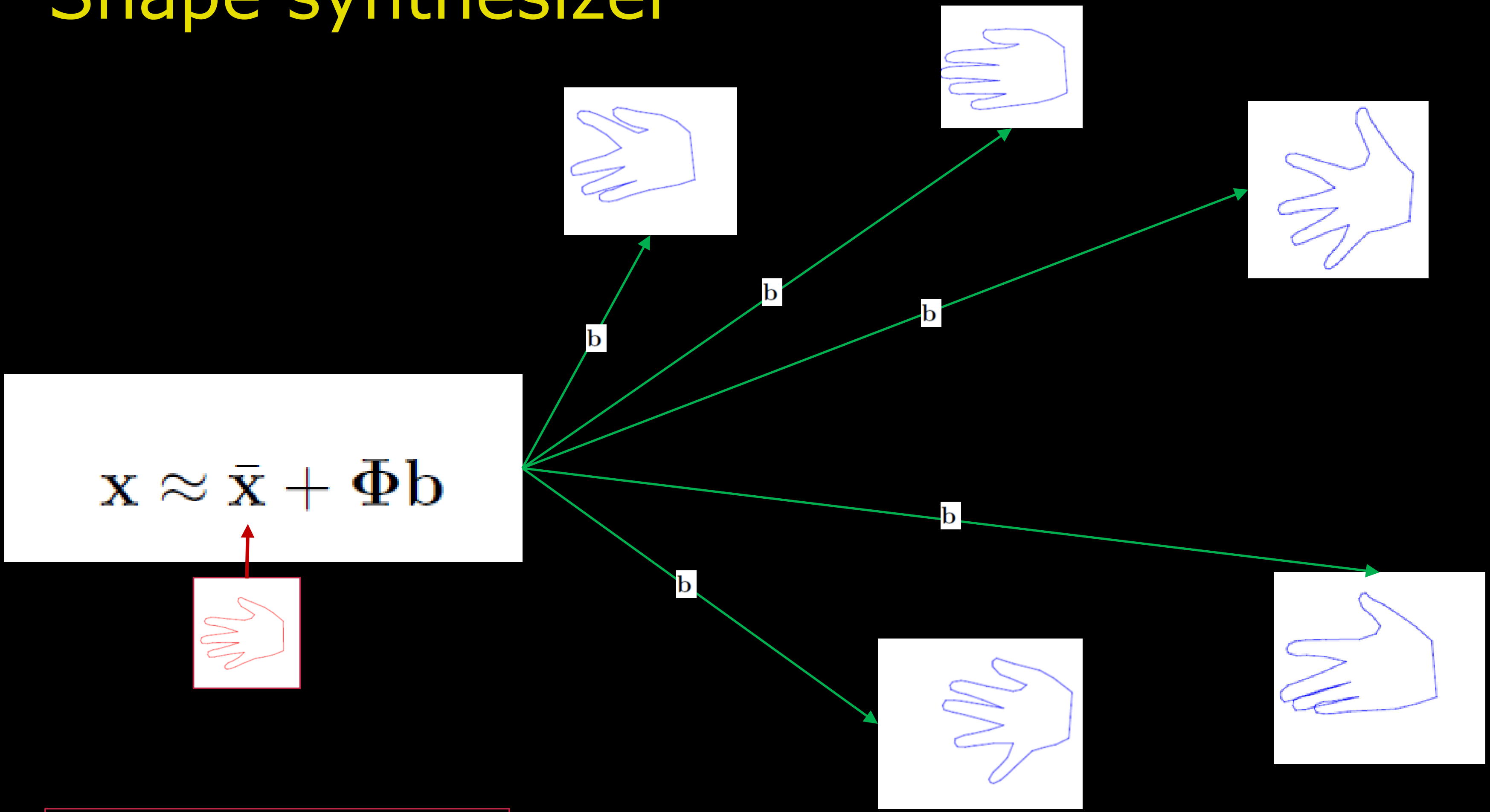
Shape space



- We can *sample* new shapes by moving around in shape space
- b are the *coordinates* in shape space
- The shape space is defined by the Eigenvectors
- b are the coordinates on the Eigenvectors

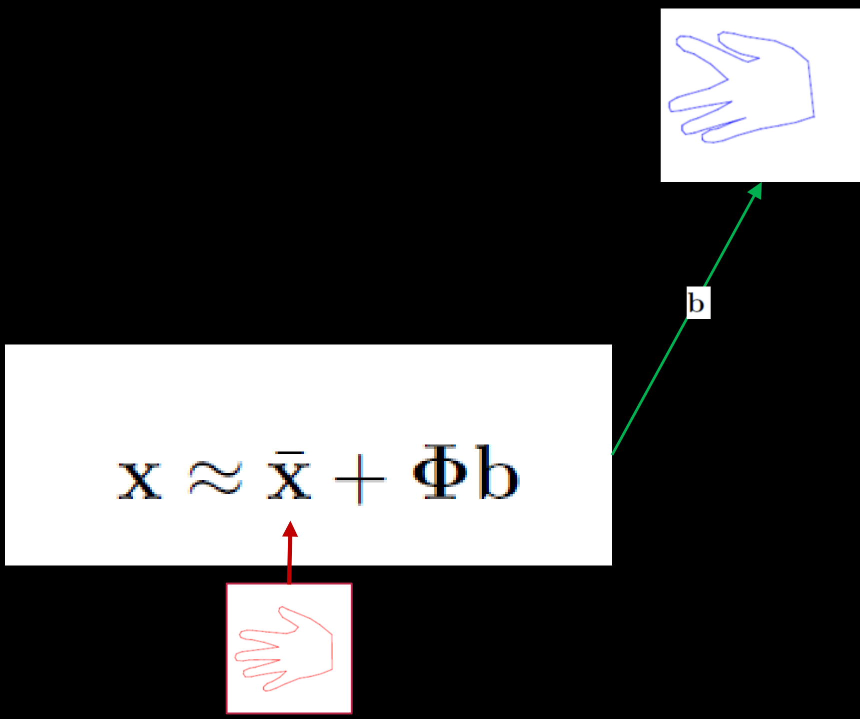
$$\mathbf{x} \approx \bar{\mathbf{x}} + \Phi \mathbf{b}$$

Shape synthesizer



A *generative* model

Shape synthesizer

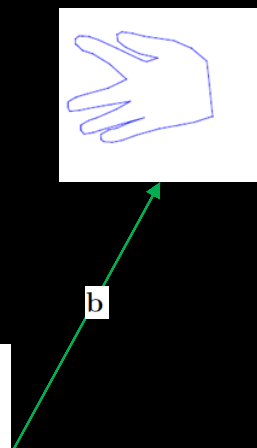


- b needs to be *constrained*
- Should be bounded by the learned shape space
- Using the size of the Eigenvalues

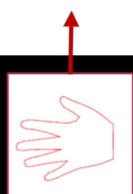
$$-3\sqrt{\lambda_1} < b_1 < 3\sqrt{\lambda_1}$$

A *generative* model

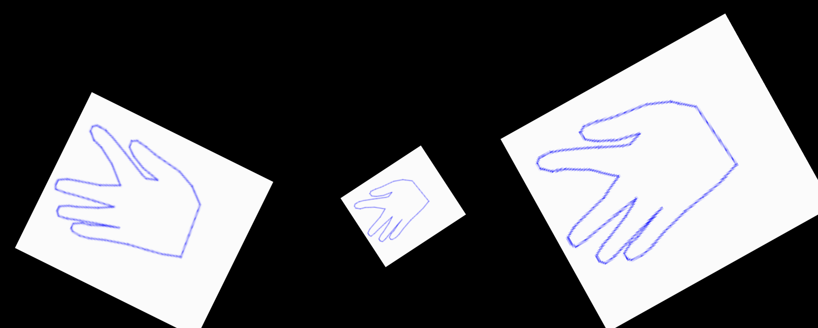
Shape synthesizer + geometrical transformation



$$\mathrm{x} \approx \bar{\mathrm{x}} + \Phi \mathrm{b}$$



- Adding a geometrical transformation
  - Translation X_t, Y_t
  - Scale s
  - Rotation θ

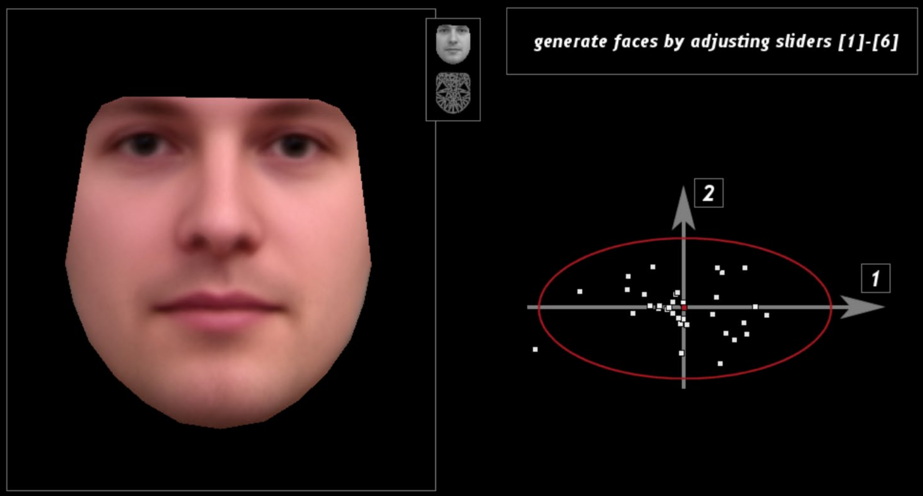


A *generative* model

Pattern recognition

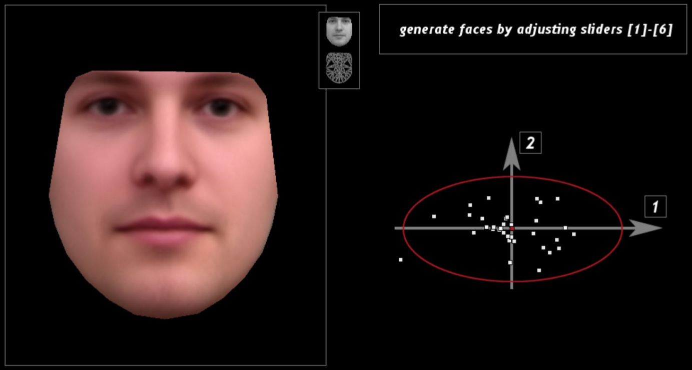
- Classical image analysis
- Hand crafting features
  - Eye detector
  - Nose detector
  - Mouth detector

Analysis by synthesis



- We have a generative model
  - A face synthesizer
  - A face is represented by a *few* parameters: *b*

Analysis by synthesis



Generative model

Target

- Compare synthetic face with target face
  - Sum of squared differences
- Change parameters of model until difference is minimal
  - Position, rotation, scaling
  - b vector

Similar to image registration with a deformable *moving image*

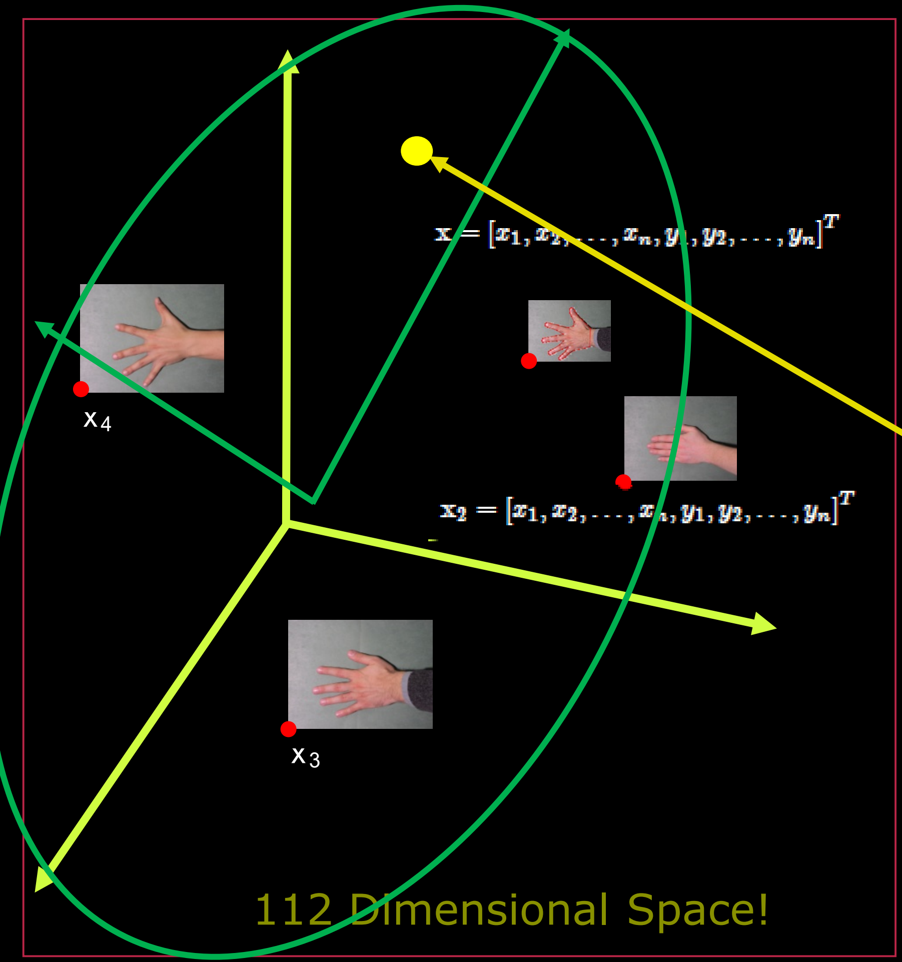
Fitting a shape and appearance model

- Finding the optimal set of parameters: position, rotation, size and b vector of model
- An *optimization* problem
- In general very hard
- Custom solutions exist

Left: Fitted model
Right: Real photo

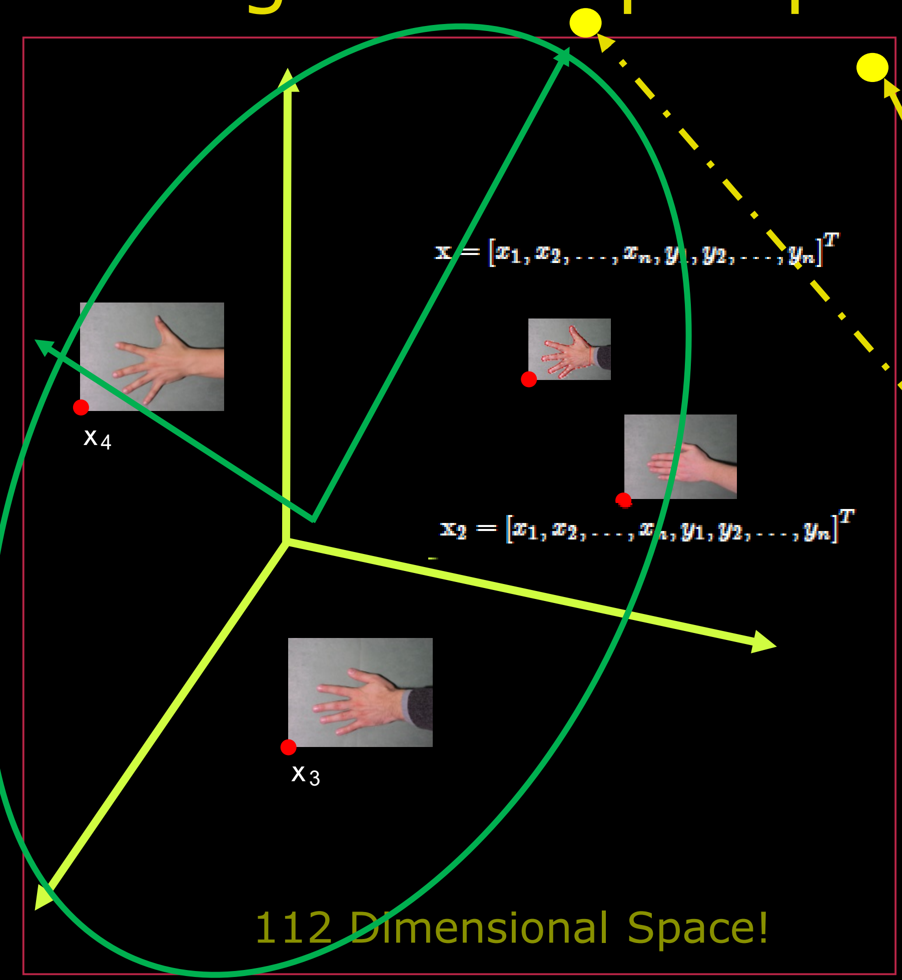
Tim Cootes: Active Appearance models

Using the shape space



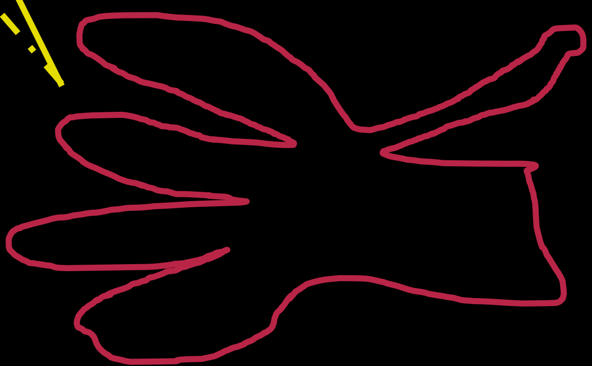
- Given a shape
  - It can be placed in shape space

Using the shape space

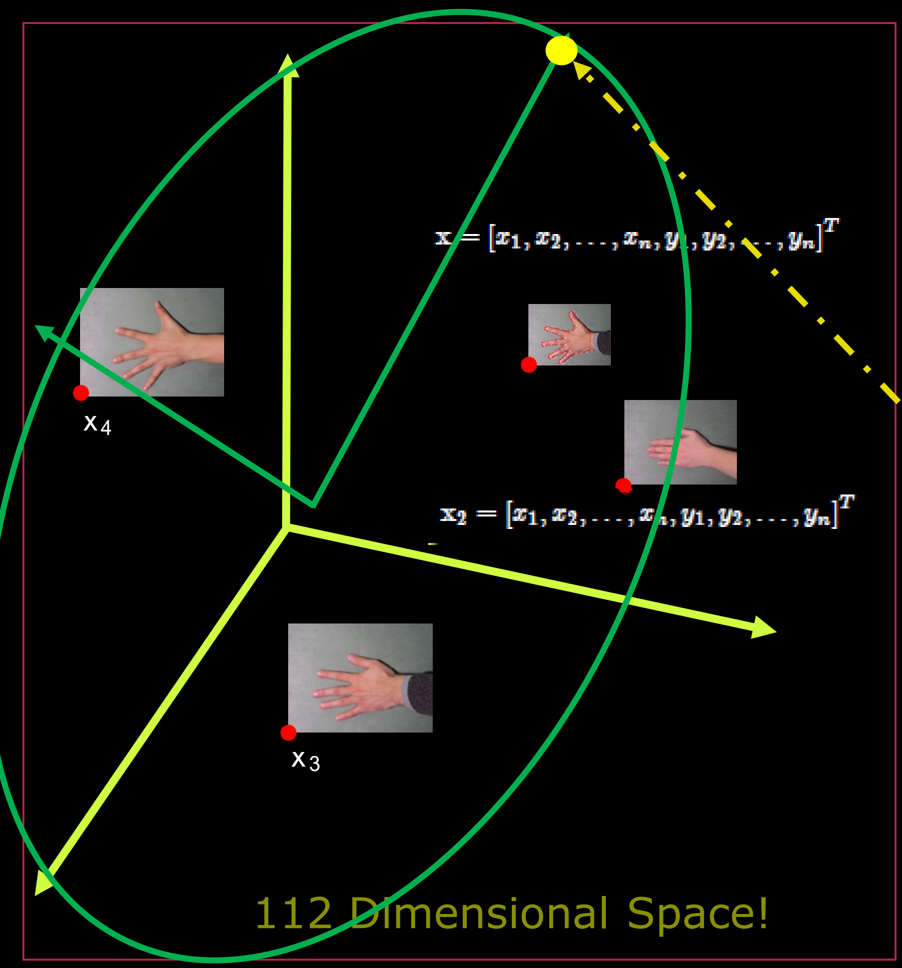


- Given a shape
  - It can be placed in shape space
- It can be projected to the Eigenvectors

Not anatomically plausible



Using the shape space



- Given a shape
  - It can be placed in shape space
- It can be projected to the Eigenvector
- And bounded by the Eigenvalues

$$\boxed{-3\sqrt{\lambda_1} < b_1 < 3\sqrt{\lambda_1}}$$

Closest anatomically plausible shape

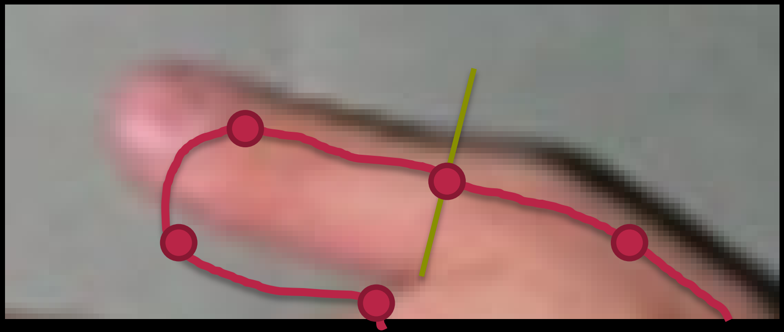
Course evaluation

- Please all use DTU Inside to evaluate the course!
- What did you like and what worked well?
- What do you think can be improved?
  - Perhaps how?

Fitting the active shape model to a new image

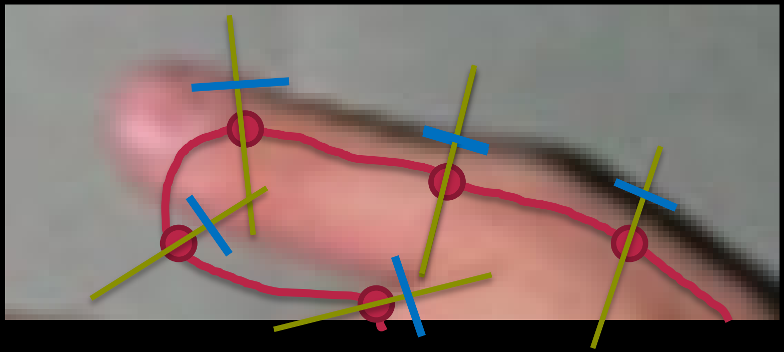
- Place the average shape on top
- Fit model points to actual image

Fitting the active shape model to a new image



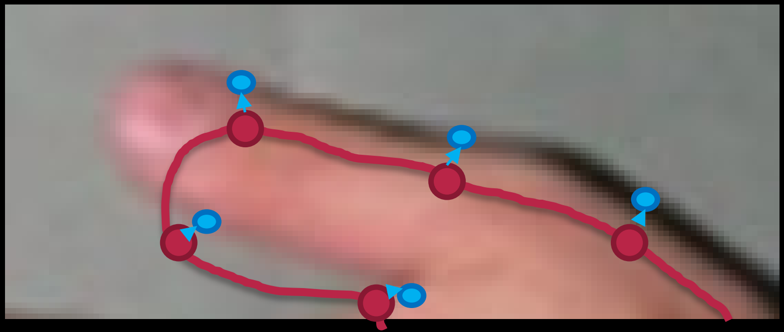
- Fit model points to actual image
- For each point:
  - Search along normal direction
  - Find highest grey level gradient

Fitting the active shape model to a new image



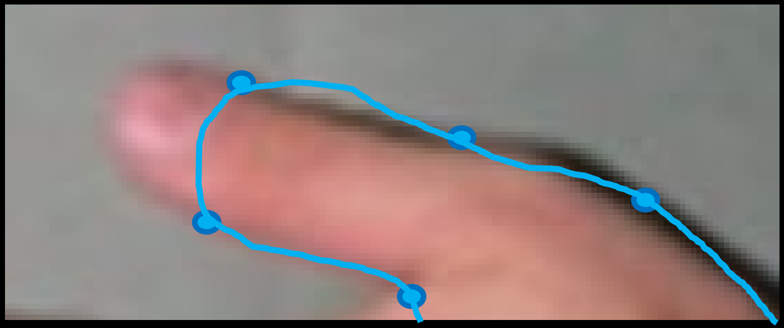
- Fit model points to actual image
- For each point:
  - Search along normal direction
  - Find highest grey level gradient

Fitting the active shape model to a new image



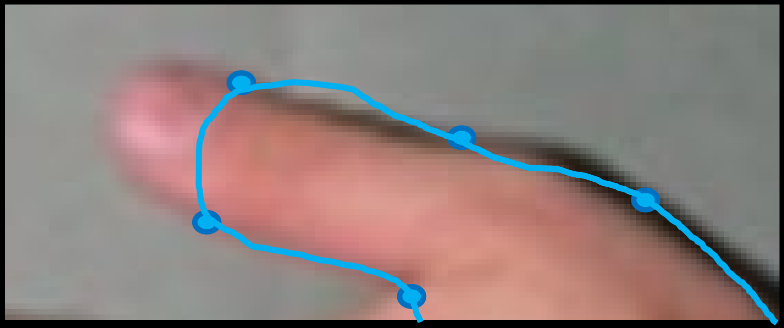
- Compute translation, rotation and scaling
  - Landmark based registration
- Move points to create new shape

Fitting the active shape model to a new image



- Compute translation, rotation and scaling
  - Landmark based registration
- Move points to create new shape

Fitting the active shape model to a new image



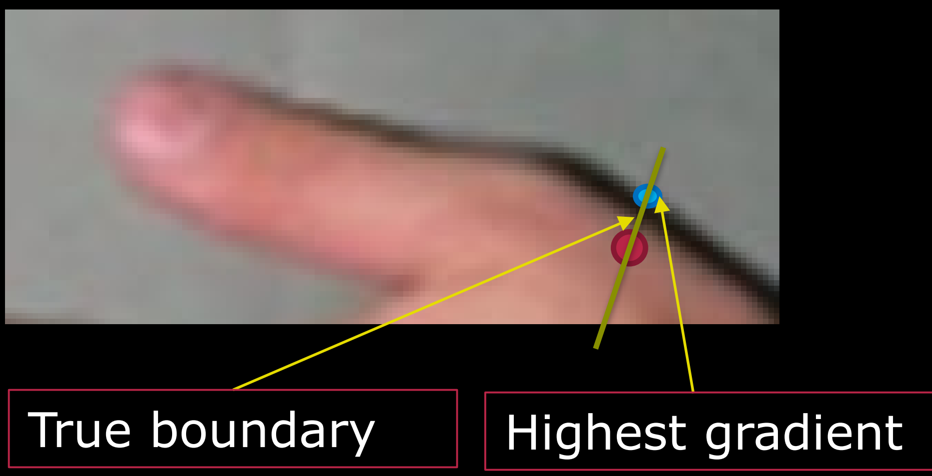
- Put new shape in shape space
- Project on Eigenvectors
- Constrain using Eigenvalues
  - Also called *regularization*

Result: Shape that matches image and that is anatomically plausible

$$-3\sqrt{\lambda_1} < b_1 < 3\sqrt{\lambda_1}$$

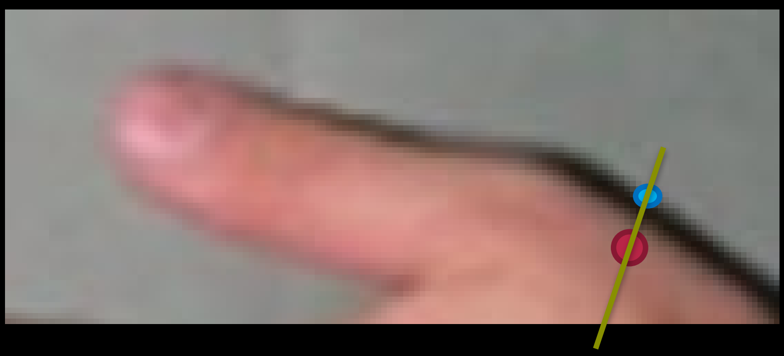
Modelling local structure

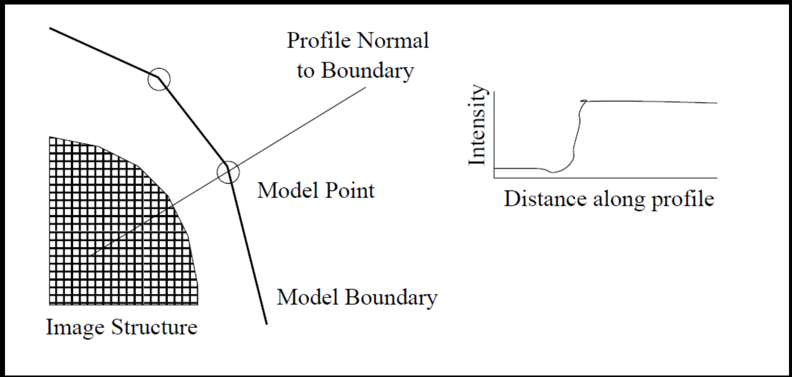
- The boundary is not always where there is highest gradient



Modelling local structure

- Sample along profile
- Normalise using sum of values



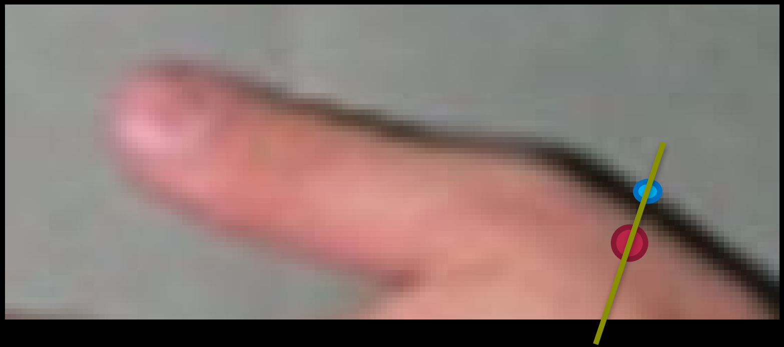


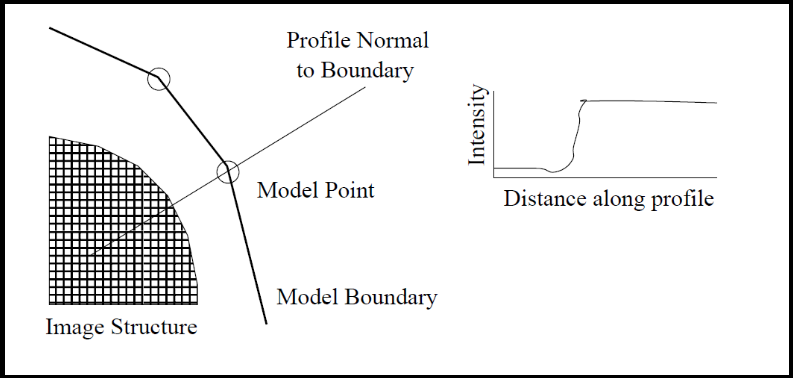
$$\mathbf{g}_i \rightarrow \frac{1}{\sum_j |g_{ij}|}\mathbf{g}_i$$

Modelling local structure

- Approximate distribution of samples
  - Multivariate Gaussian

$$\text{mean } \bar{\mathbf{g}} \text{ and covariance } \mathbf{S}_g$$





$$\mathbf{g}_i \rightarrow \frac{1}{\sum_j |g_{ij}|}\mathbf{g}_i$$

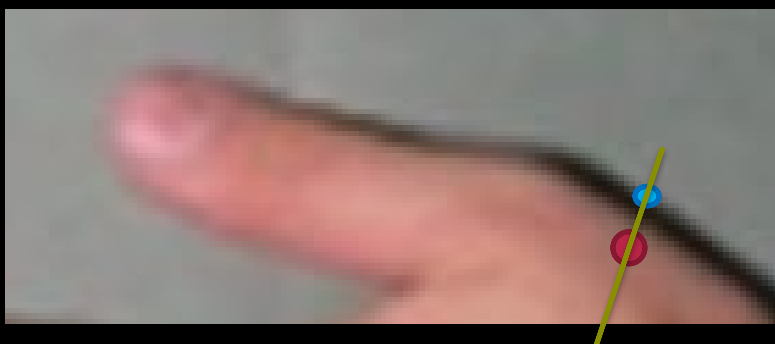
Modelling local structure

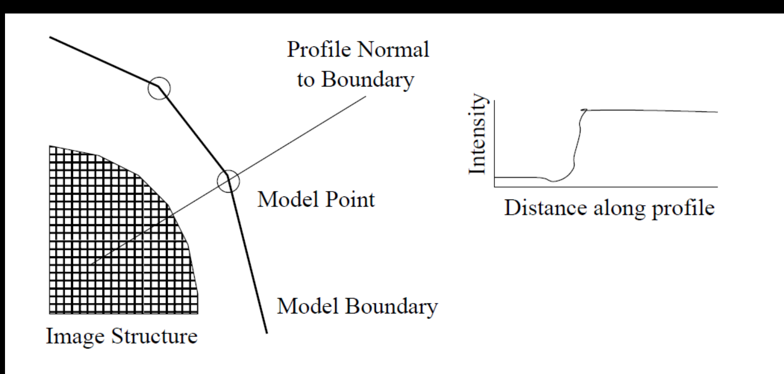
- Instead of using the gradient to search, a quality of fit is used:

The quality of fit of a new sample, $\mathbf{g}_s$, to the model is given by

$$f(\mathbf{g}_s) = (\mathbf{g}_s - \bar{\mathbf{g}})^T \mathbf{S}_g^{-1} (\mathbf{g}_s - \bar{\mathbf{g}})$$

This is the Mahalanobis distance of the sample from the model mean.





$$\mathbf{g}_i \rightarrow \frac{1}{\sum_j |g_{ij}|} \mathbf{g}_i$$

Tim Cootes: Active Appearance models

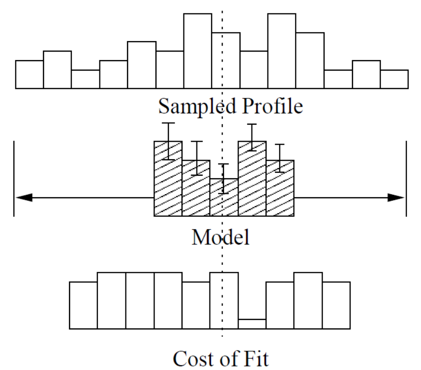
Modelling local structure

- Instead of using the gradient to search, a quality of fit is used:

The quality of fit of a new sample, $\mathbf{g}_s$, to the model is given by

$$f(\mathbf{g}_s) = (\mathbf{g}_s - \bar{\mathbf{g}})^T \mathbf{S}_g^{-1} (\mathbf{g}_s - \bar{\mathbf{g}})$$

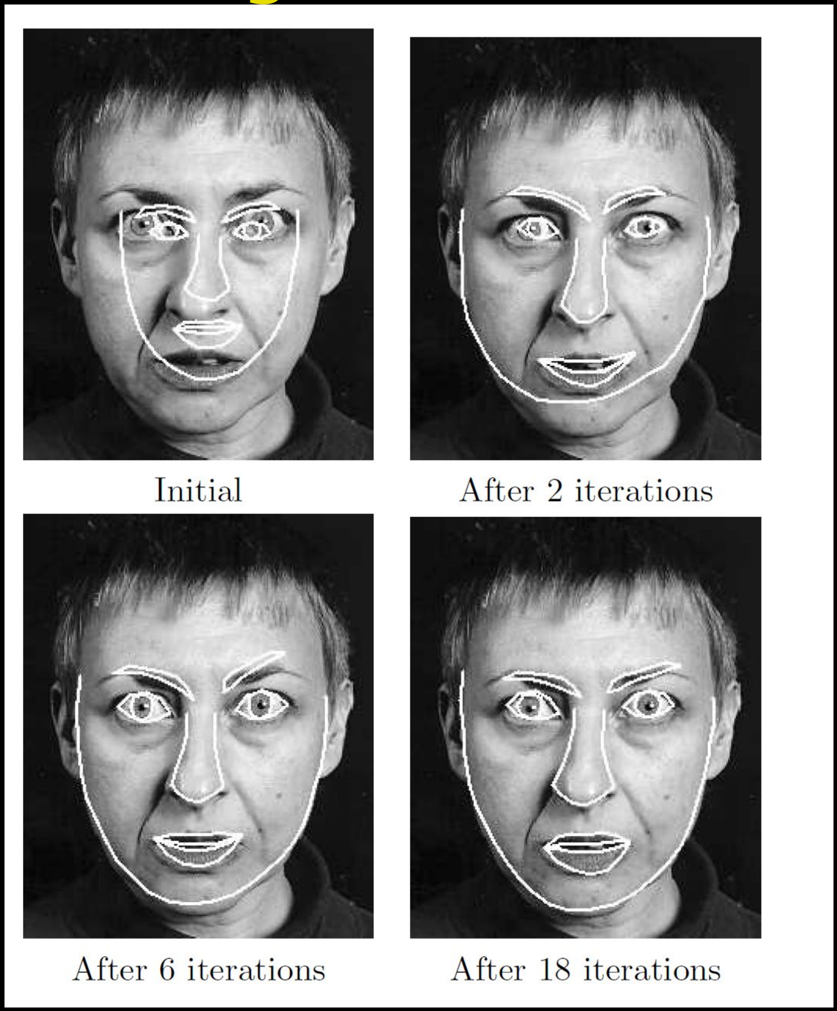
This is the Mahalanobis distance of the sample from the model mean.



Search along sampled profile to find best fit of grey-level model

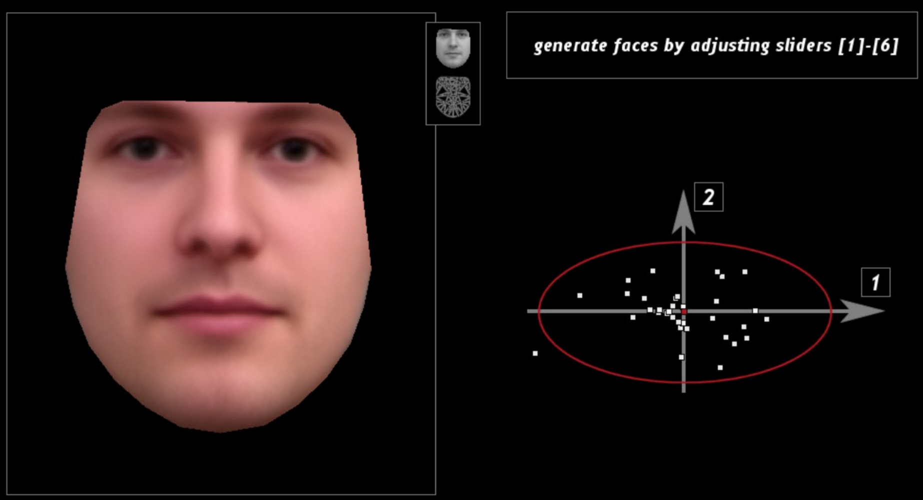
Tim Cootes: Active Appearance models

Fitting to a new shape



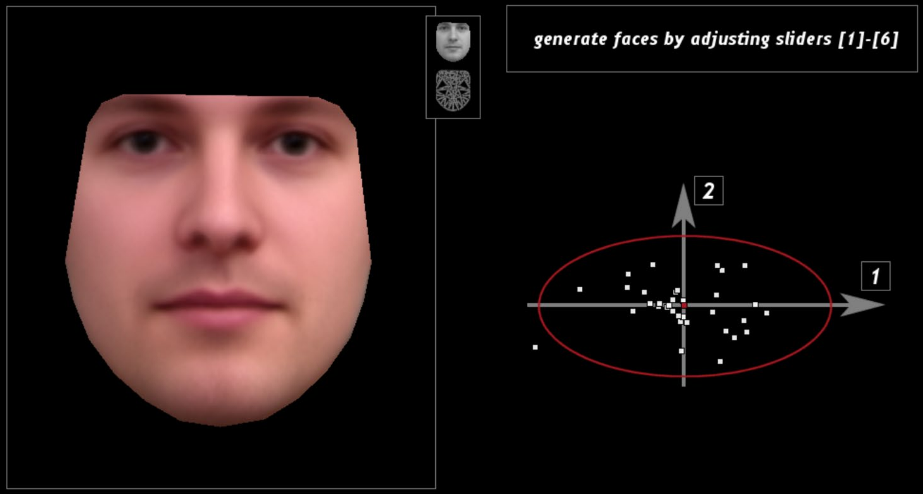
Tim Cootes: Active Appearance models

The problem with strong priors



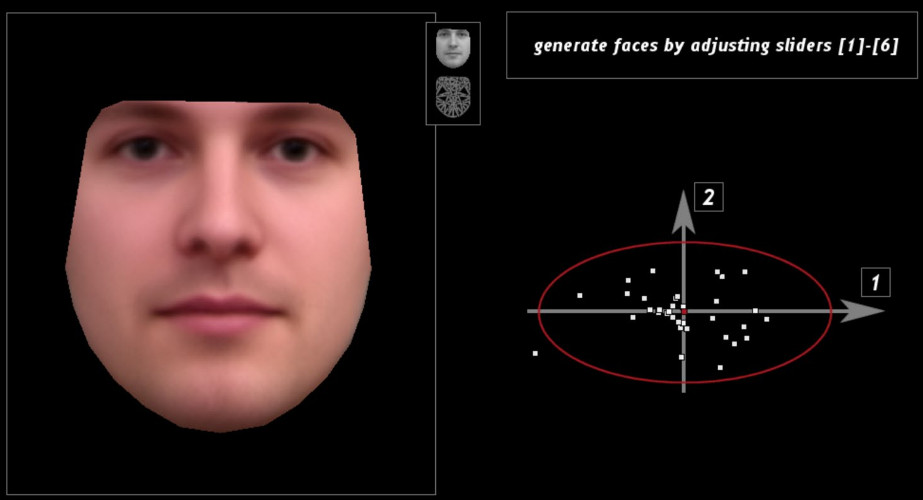
- A *prior*
  - What was known before
  - A statistical shape model

The problem with strong priors



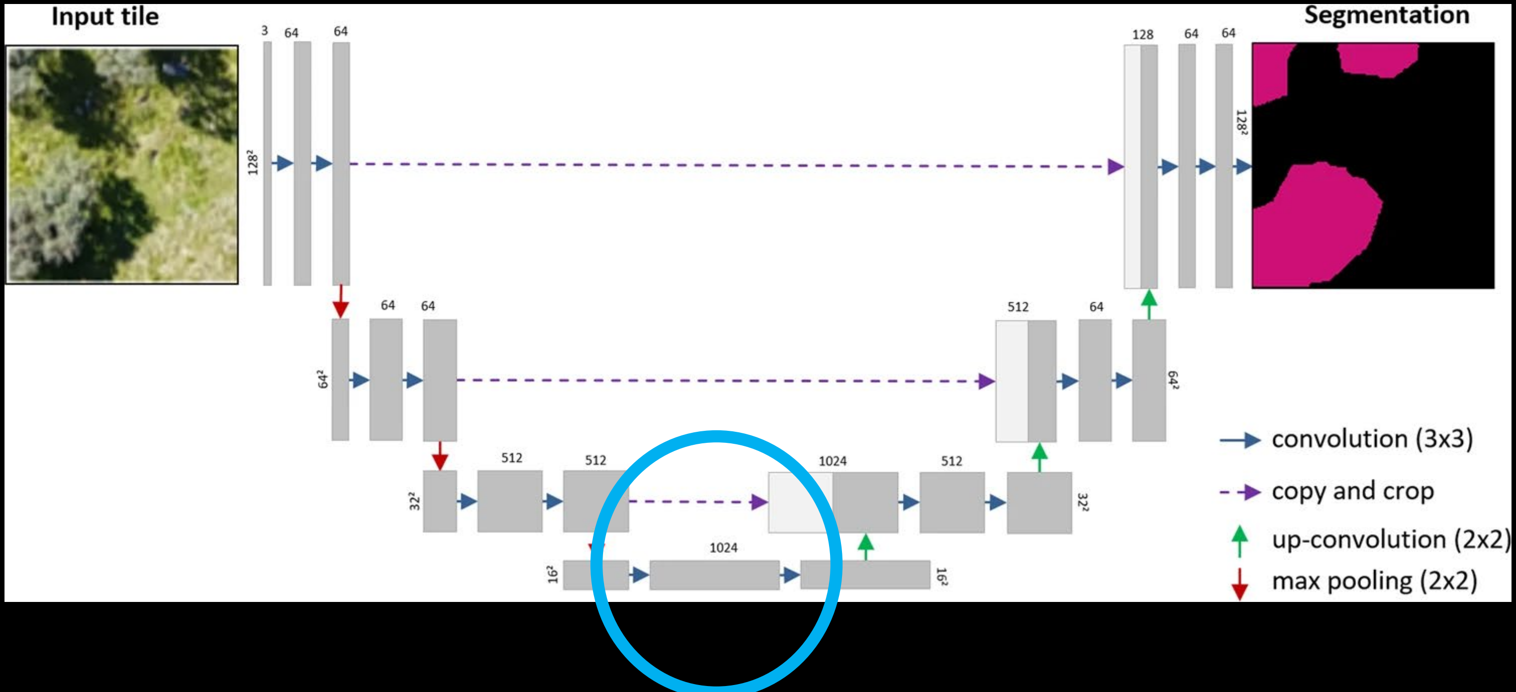
- Model is trained on images of adults
- Will try to force all fits to *look like adults*
- Will not work well with images outside the *prior*

Testing the model



- Important to the model on independent data
- How it *generalizes*
- Is the prior too strong?

PCA space vs. Latent space



Latent space

Kattenborn, T., Eichel, J. & Fassnacht, F.E. Convolutional Neural Networks enable efficient, accurate and fine-grained segmentation of plant species and communities from high-resolution UAV imagery. *Sci Rep* **9,** 17656 (2019).

Next week

- Digital test exam
- Examples of advanced topics in image analysis