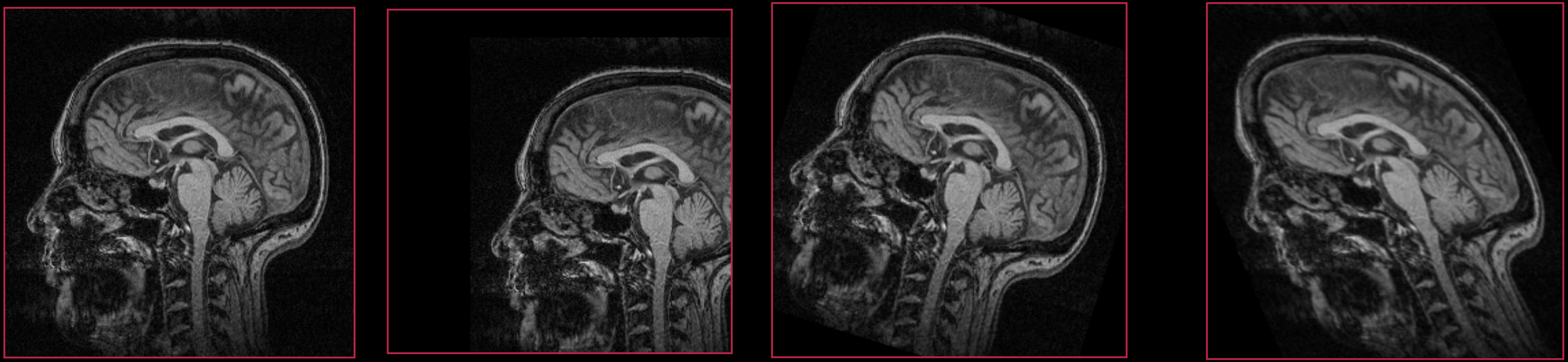
# Image Analysis

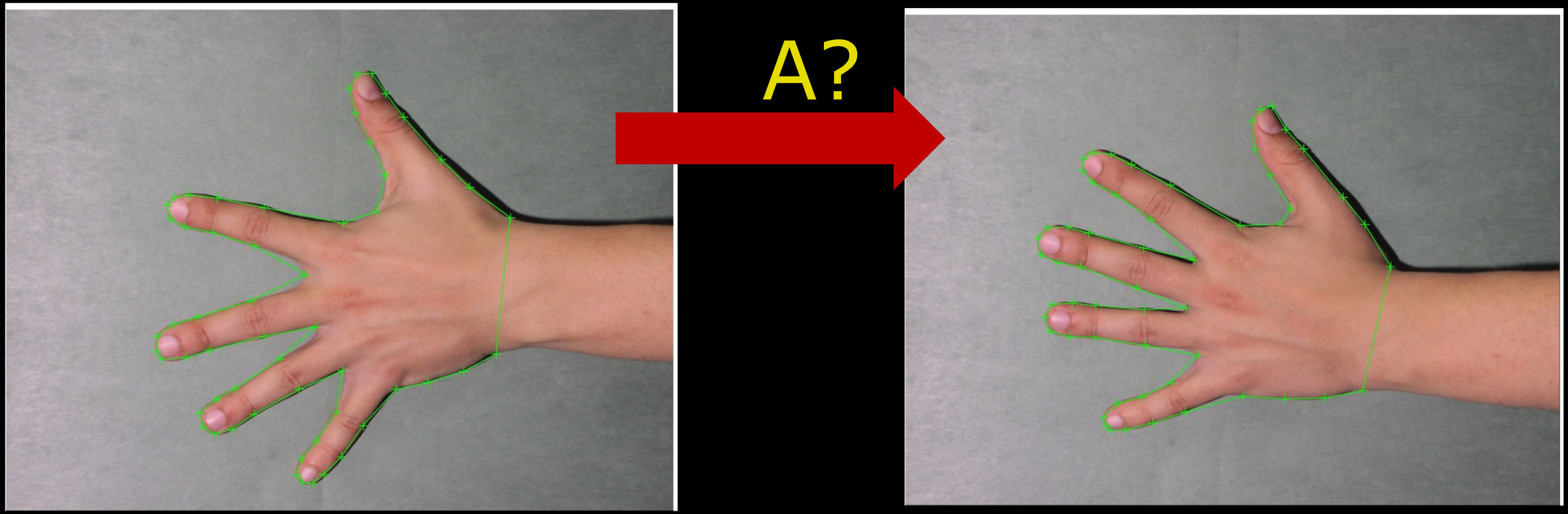
Tim B. Dyrby
Rasmus R. Paulsen
DTU Compute

tbdy@dtu.dk

http://www.compute.dtu.dk/courses/02503

# Lecture 8 – Geometric Transformation and image registration





## What can you do after today?

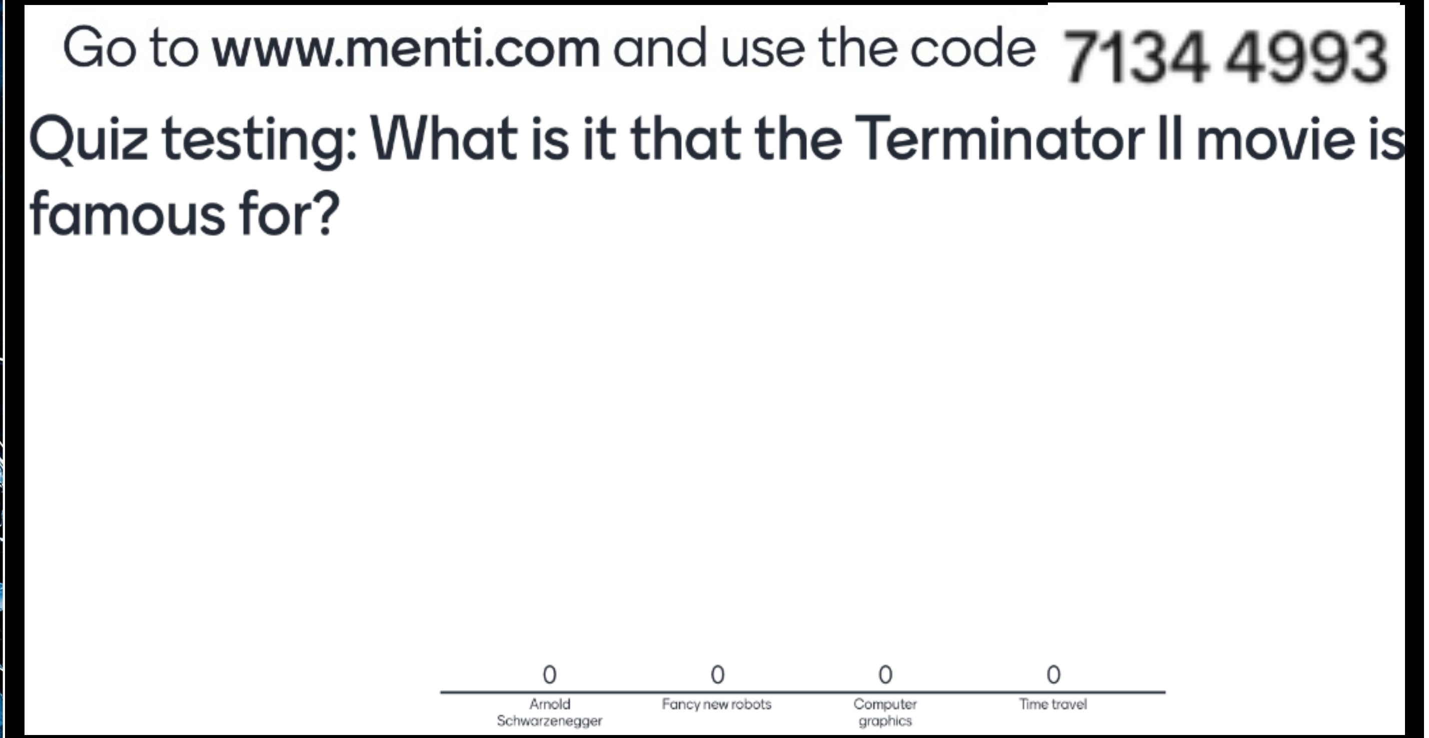
Geometrical transformation

- 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

From 1991



Go to www.menti.com and use the code 7134 4993

Quiz testing: What is it that the Terminator II movie is famous for?

1) Arnold Schwarzenegger

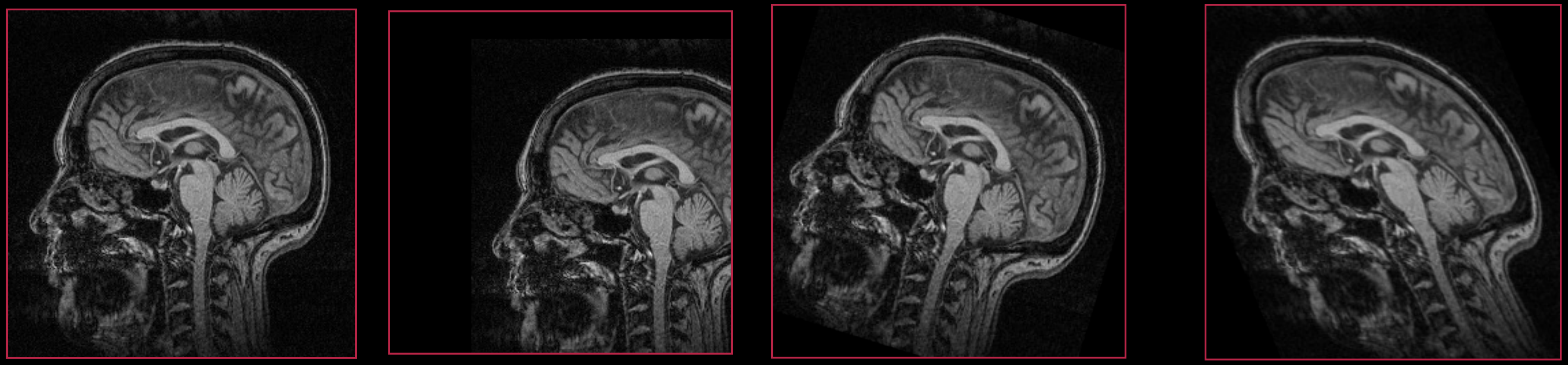
2) Fancy new robots

3) Computer graphics

4) Time travel

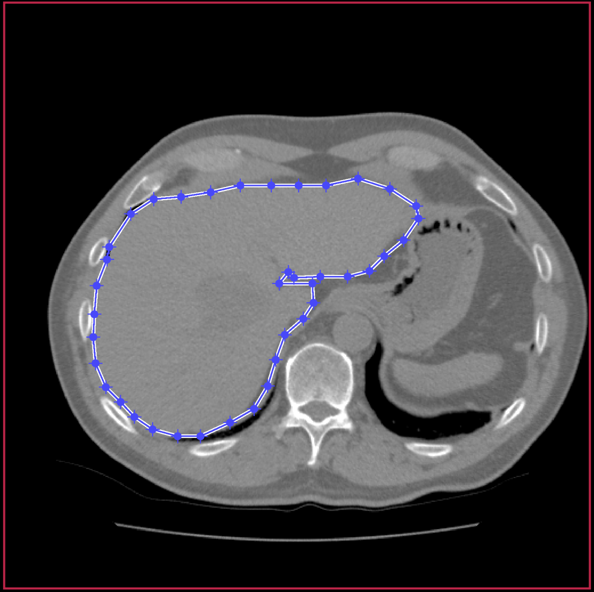
## Geometric transformation

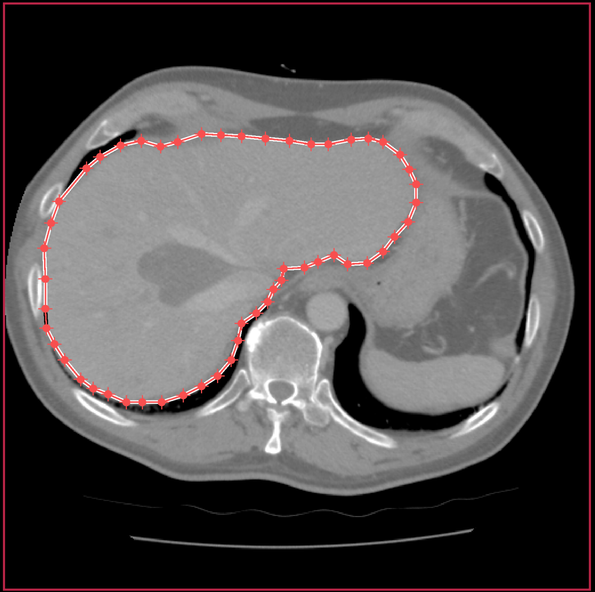
- Moving and changing the dimensions of images
- Why do we need it?



## Change detection

- Patient imaged before and after surgery
- What are the changes in the operated organ?
- Patient cannot be placed in the exact same position in the scanner

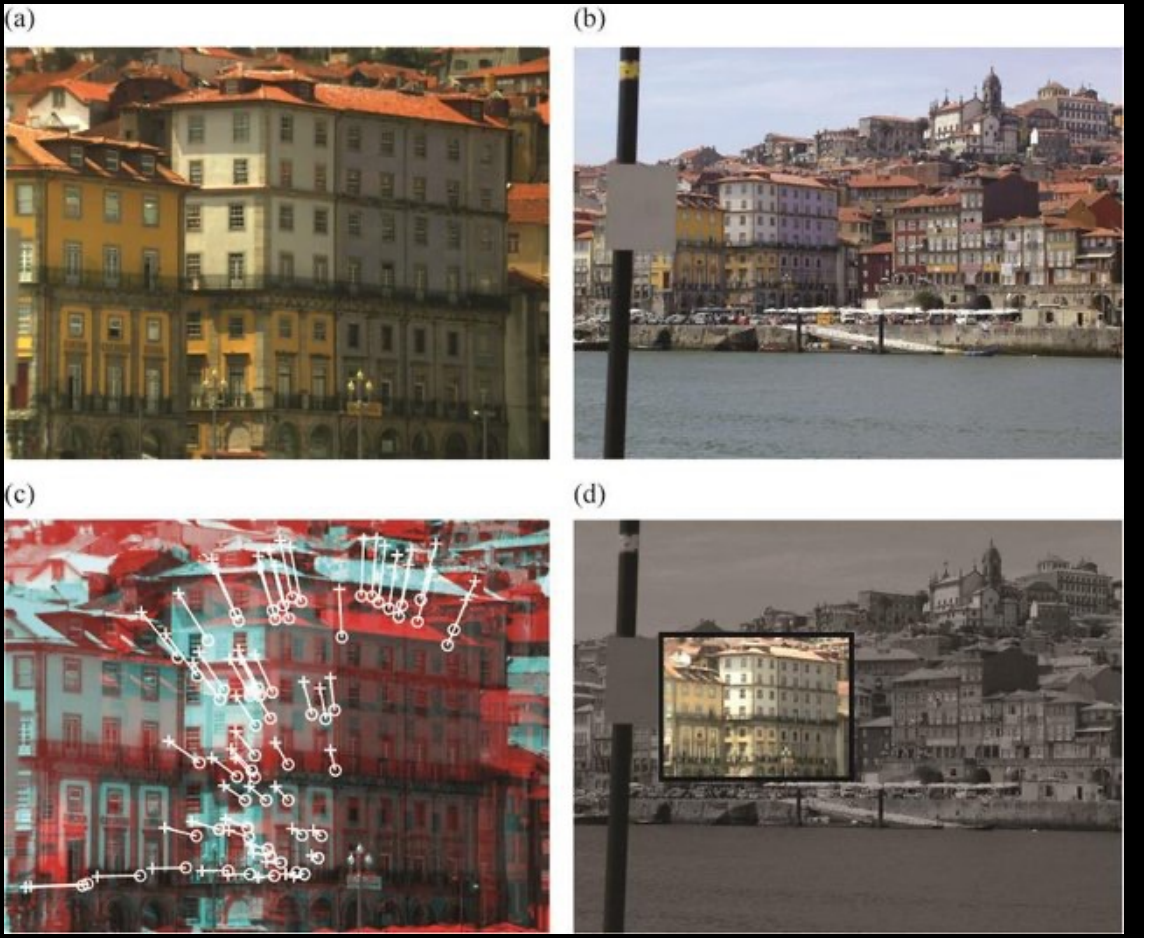

Before surgery


After surgery

Bachelor project: Image Guided Surgery Planning

## Similarity transform

- Objects at different distances

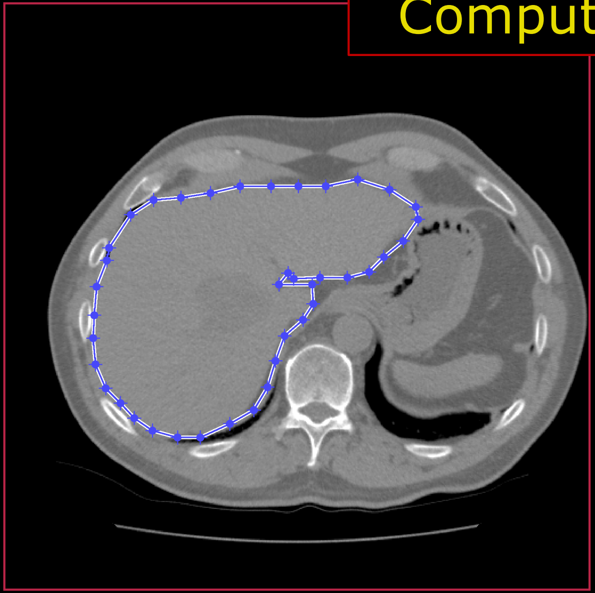


Amano et al 2016, DOI: 10.1051/matecconf/20166600024

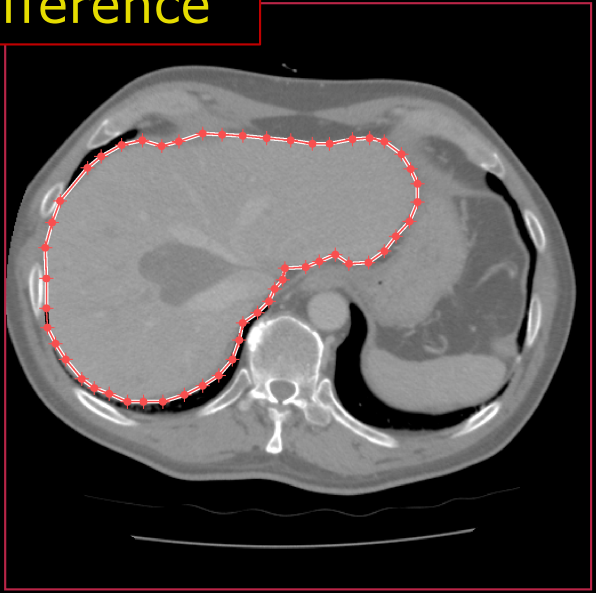
## Image Registration

- 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



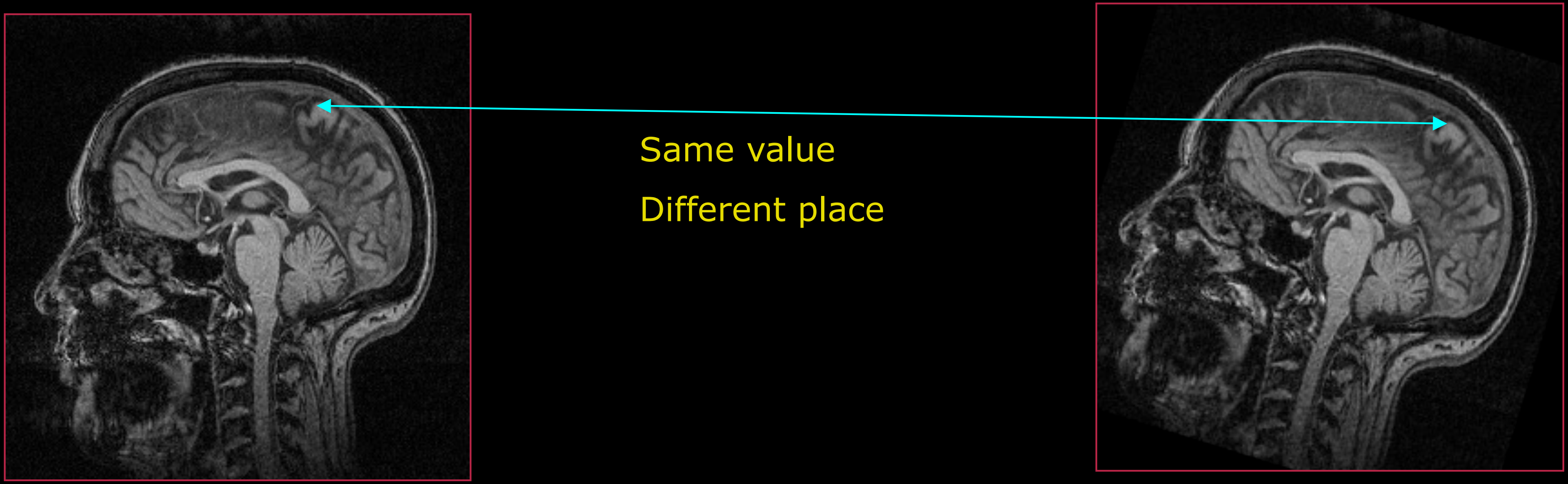
Template



Reference

## Geometric Transform

- The pixel intensities are not changed
- The "pixel values" just change positions



## Different transformations

- Translation
- Rotation
- Scaling
- Shearing

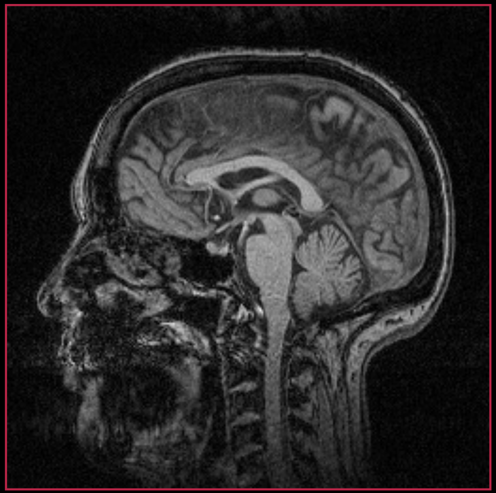
- Advanced transformations

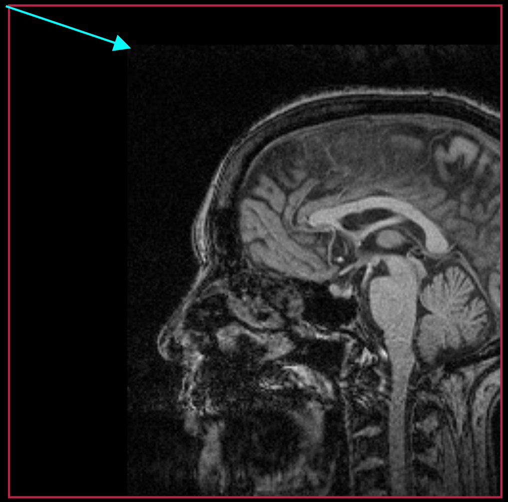
From Terminator 2 movie: Non-linear image transformation

## Translation

- The image is shifted – both vertically and horizontally

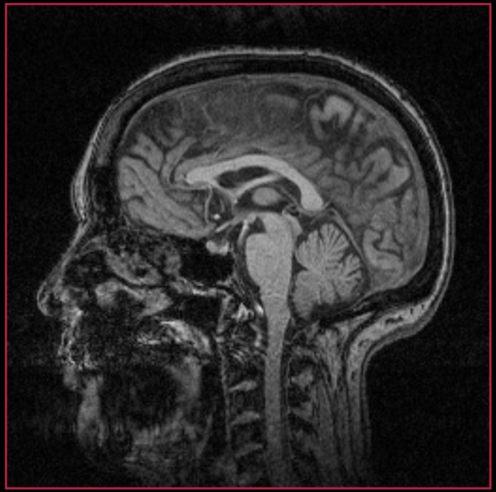
$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 60 \\ 20 \end{bmatrix}$$



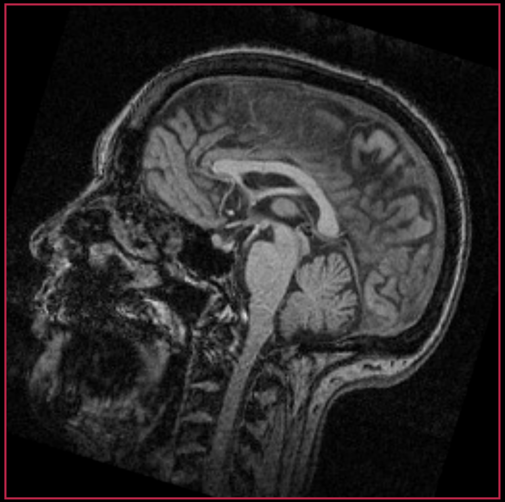


## Rotation

- The image is rotated around origin (0,0) i.e., upper left corner or a user defined position e.g. around the image center
- Remember to use degrees and radians correctly
  - Python uses radians
  - Degrees easier for us humans

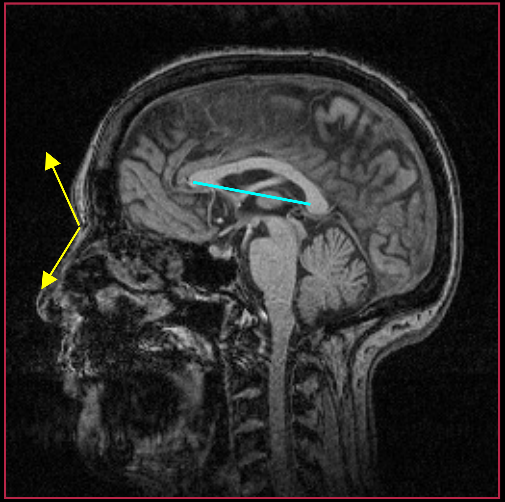


$\theta = 15^\circ$



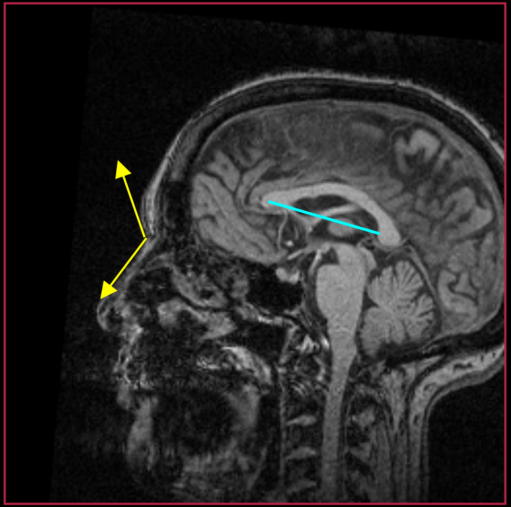
## Rigid body transformation

- Translation and rotation
- Rigid body
- Angles and distances are kept



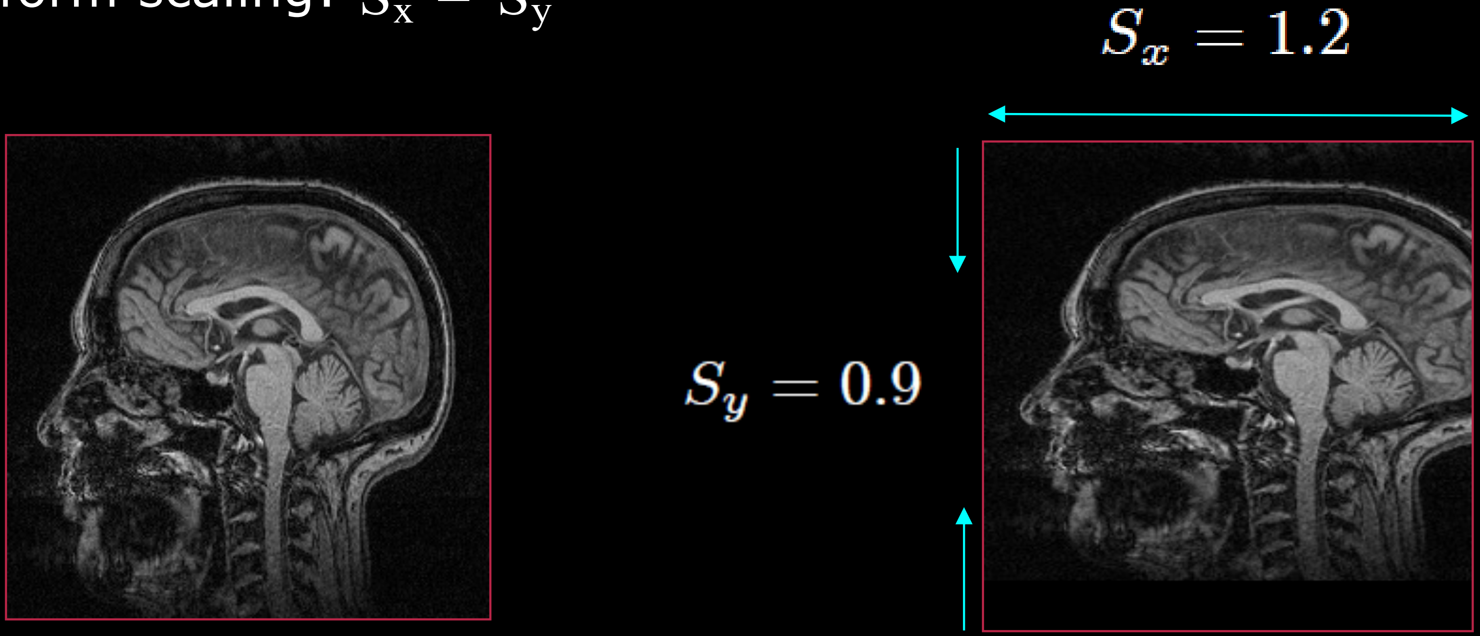
$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 60 \\ 20 \end{bmatrix}$$

$$\theta = 5^\circ$$



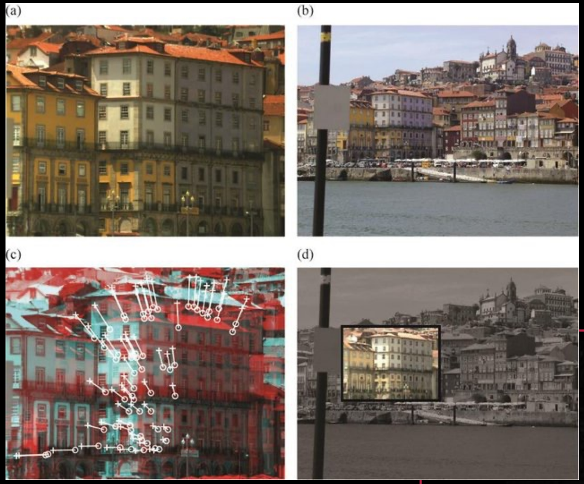
## Scaling

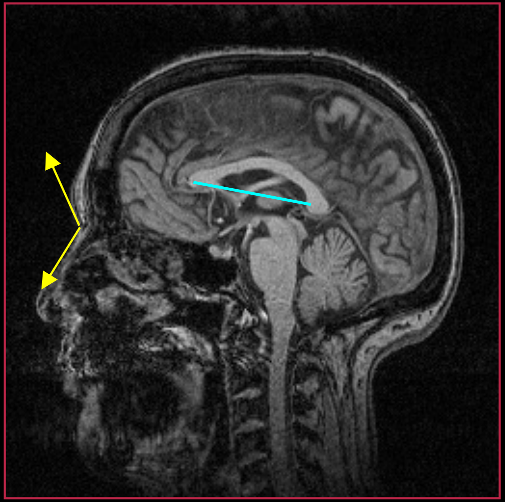
- The size of the image is changed
- Scale factors
  - X-scale factor $S_x$
  - Y-scale factor $S_y$
- Uniform scaling: $S_x = S_y$

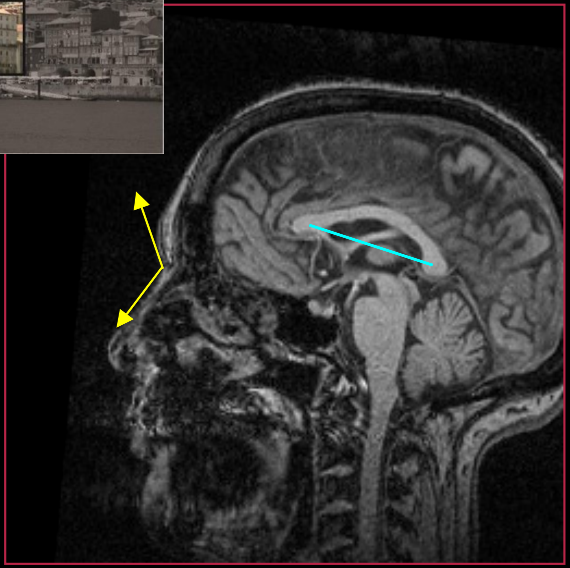


## Similarity transformation

- Translation, and uniform scaling
- Angles are kept
- Distances change

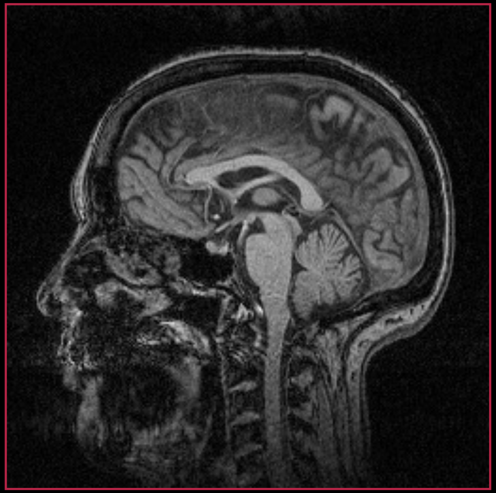


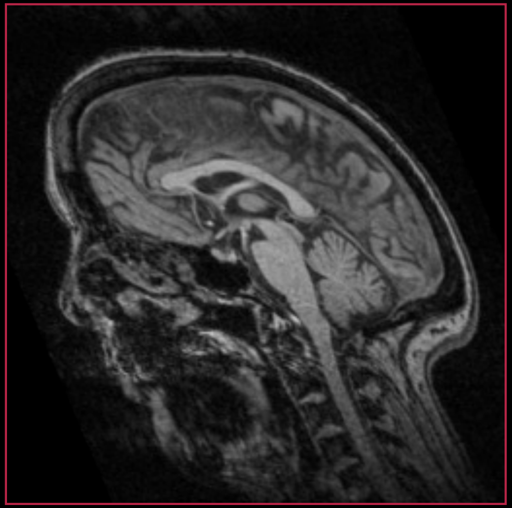




## Shearing

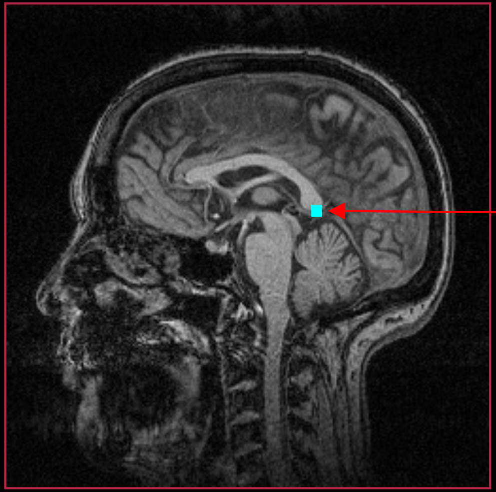
- Pixel shifted horizontally or/and vertically
- Shearing factors
  - X-shear factor $B_x$
  - Y-shear factor $B_y$
- Is less used than translation, rotation, and scaling



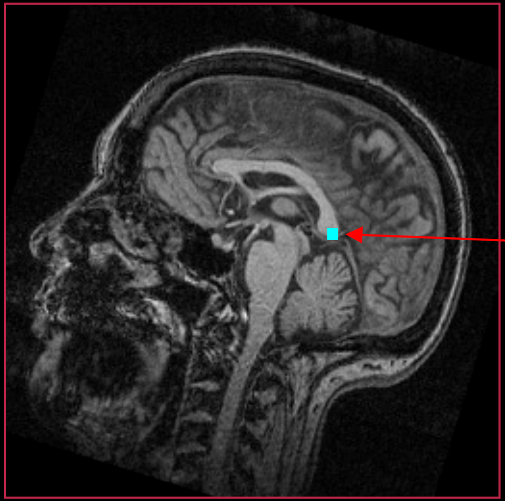


## Transformation Mathematics

f



g



- 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

$$x' = \boxed{A_x(x,y)}$$
$$y' = A_y(x,y)$$

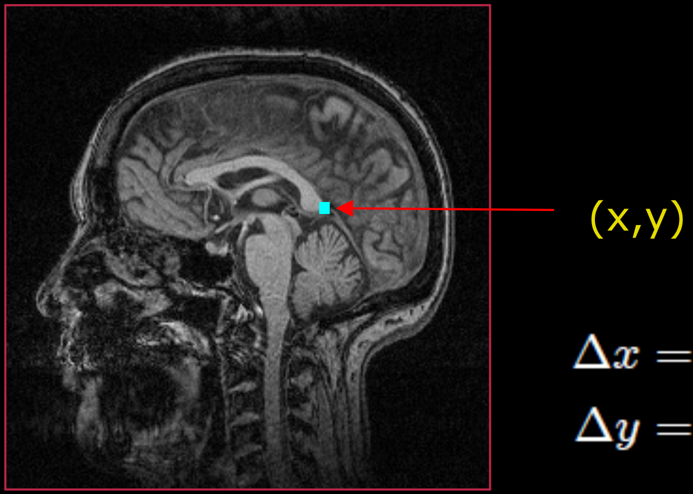
Depends on both x and y!

## Translation mathematics

- The image is shifted – both vertically and horizontally

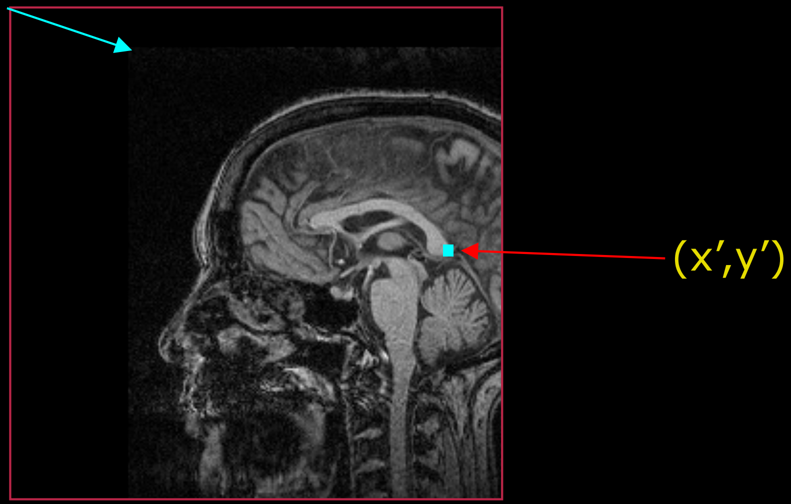
$$x' = x + \Delta x$$

$$y' = y + \Delta y$$



$$\Delta x = 60$$

$$\Delta y = 20$$



## Matrix notation

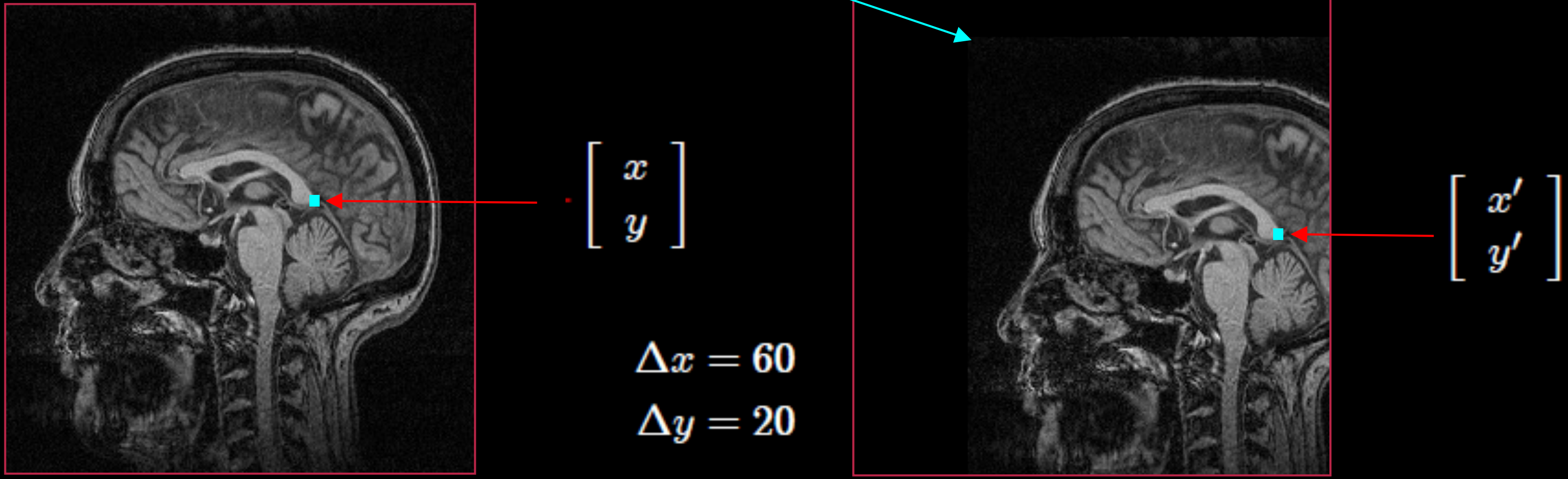
- Coordinates in column matrix format

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

## Translation mathematics in matrix notation

- The image is shifted – both vertically and horizontally

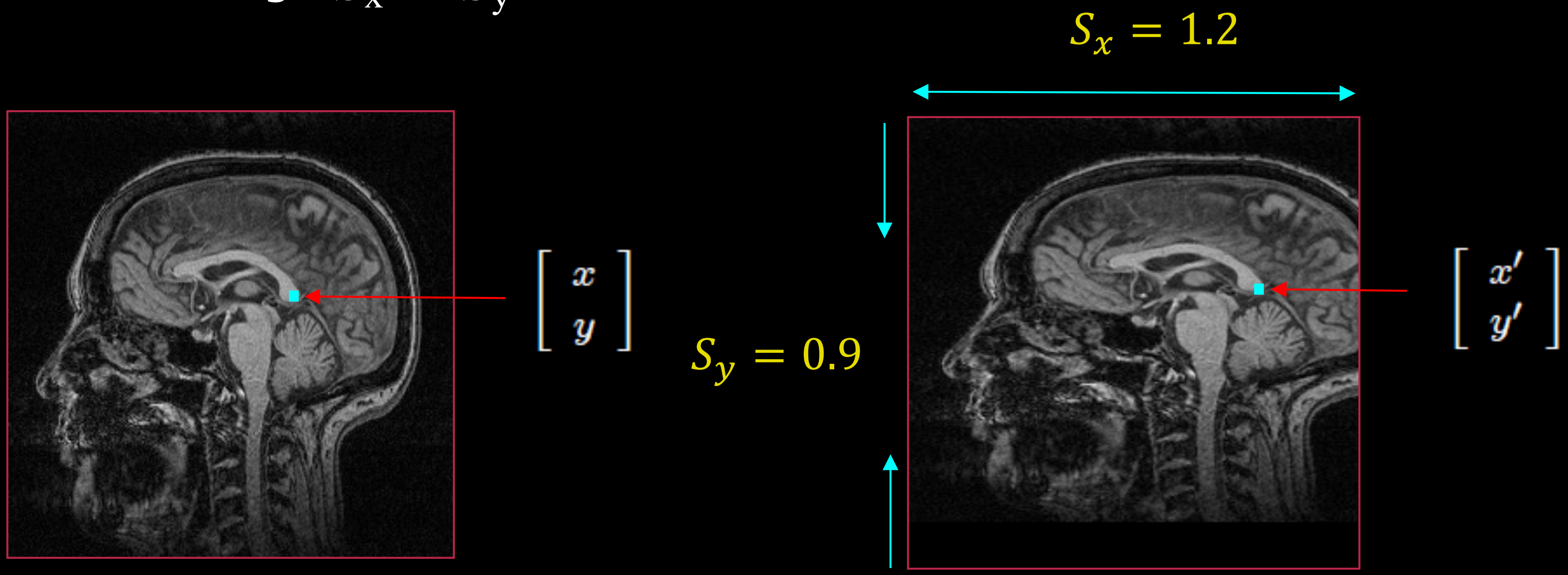
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$


$\begin{bmatrix} x \\ y \end{bmatrix}$
$\Delta x = 60$
$\Delta y = 20$
$\begin{bmatrix} x' \\ y' \end{bmatrix}$

## Scaling

- The size of the image is changed
- Scale factors
  - X-scale factor $S_x$
  - Y-scale factor $S_y$
- Uniform scaling: $S_x = S_y$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$


$S_x = 1.2$
$\begin{bmatrix} x \\ y \end{bmatrix}$
$S_y = 0.9$
$\begin{bmatrix} x' \\ y' \end{bmatrix}$

## 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

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

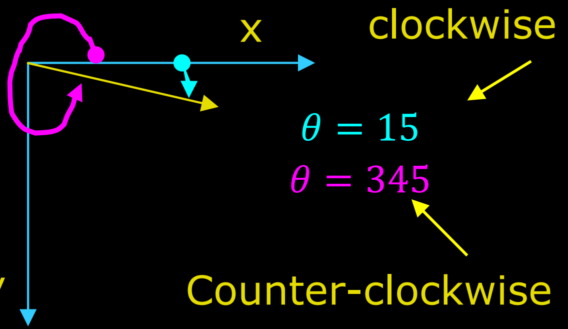
Where

$$\mathbf{A} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$$

is a *transformation matrix*

## Rotation

- A rotation matrix is used
- Is it a clockwise rotation or counter-clockwise?

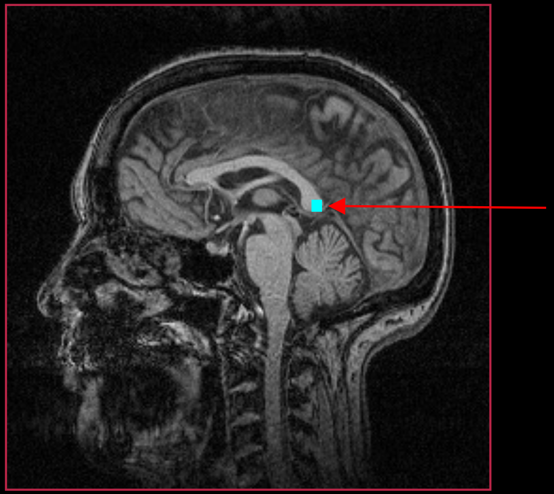


Clockwise

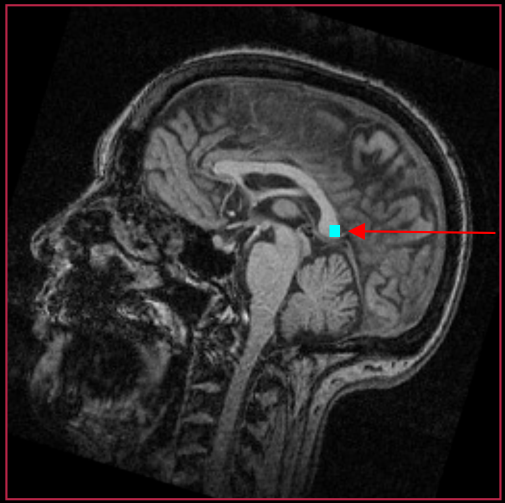
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Counter-clockwise

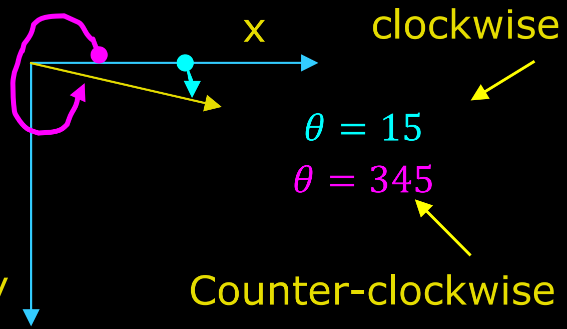
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$



$$\begin{bmatrix} x \\ y \end{bmatrix}$$



$$\begin{bmatrix} x' \\ y' \end{bmatrix}$$



## Rotation

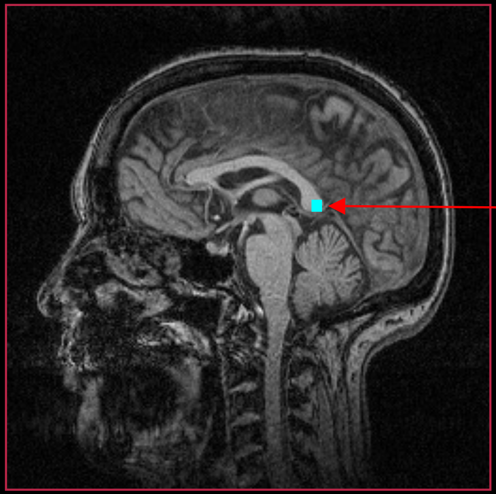
- A rotation matrix is used
- Is it a clockwise rotation or counter-clockwise?
  - In this course we use a counter-clockwise rotation

Clockwise

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

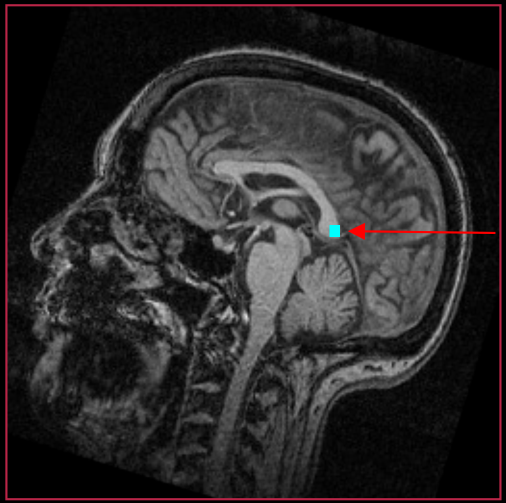
Counter-clockwise

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$



$$\begin{bmatrix} x \\ y \end{bmatrix}$$

$$\theta = 345°$$



$$\begin{bmatrix} x' \\ y' \end{bmatrix}$$

## Rotation

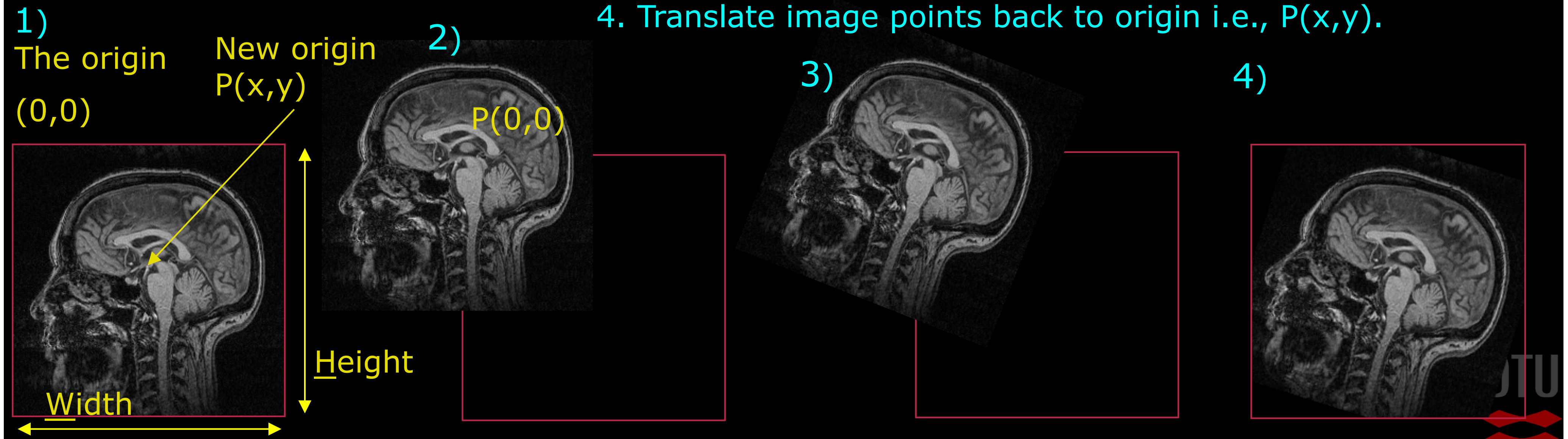
- Default rotation is around the origin (0,0) i.e., the top left corner
- How to change rotation to e.g. around the image center, P(H/2,W/2)

1. Select the point P(x,y) about which you want to rotate the 2D image e.g. image center.

2. Translate the image points such that P becomes new origin i.e., $\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -H/2 \\ -w/2 \end{bmatrix}$. P is (0,0) now i.e., the new origin.

3. Apply 2d rotation for the points in the image.

4. Translate image points back to origin i.e., P(x,y).

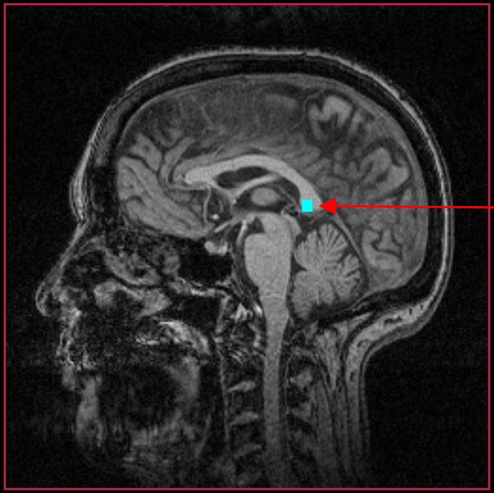


## Shearing

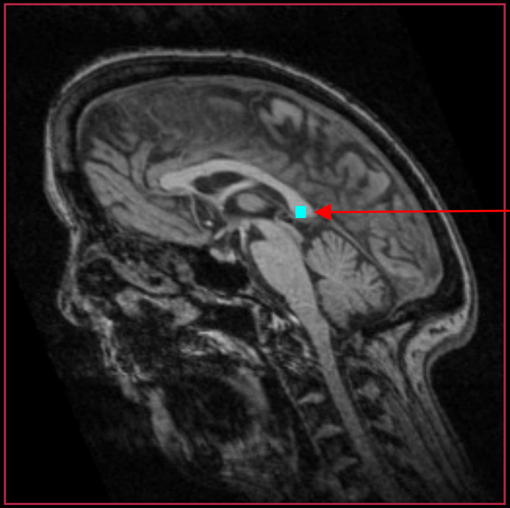
- Pixel shifted horizontally or/and vertically

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & B_x \\ B_Y & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

New x value depends on x and y



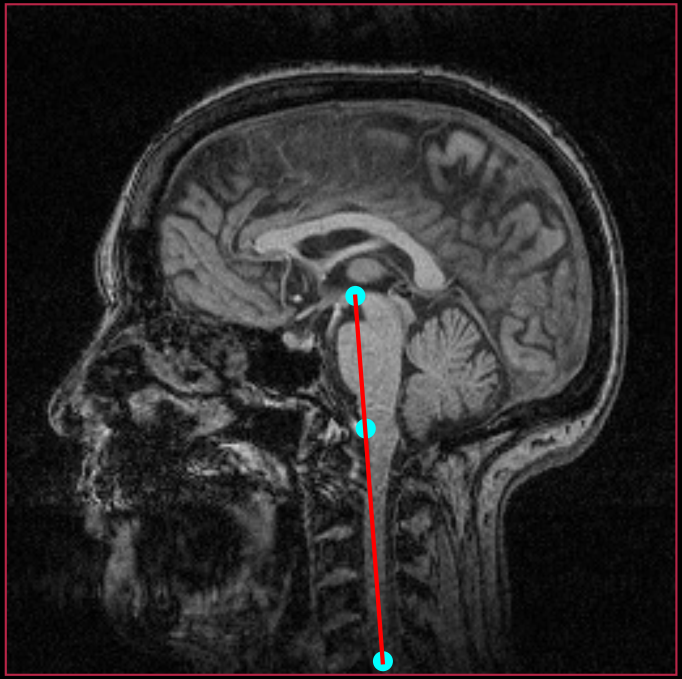
$\begin{bmatrix} x \\ y \end{bmatrix}$

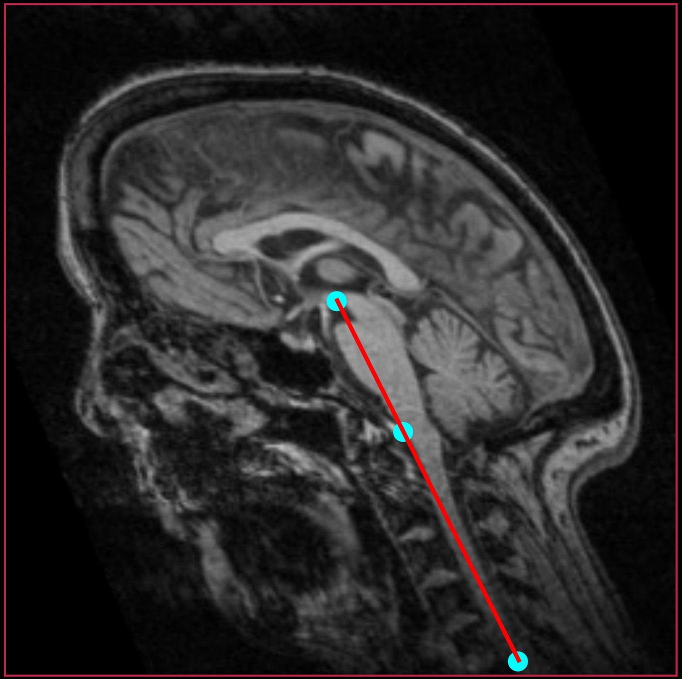


$\begin{bmatrix} x' \\ y' \end{bmatrix}$

## 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

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A}_S \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A}_R \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

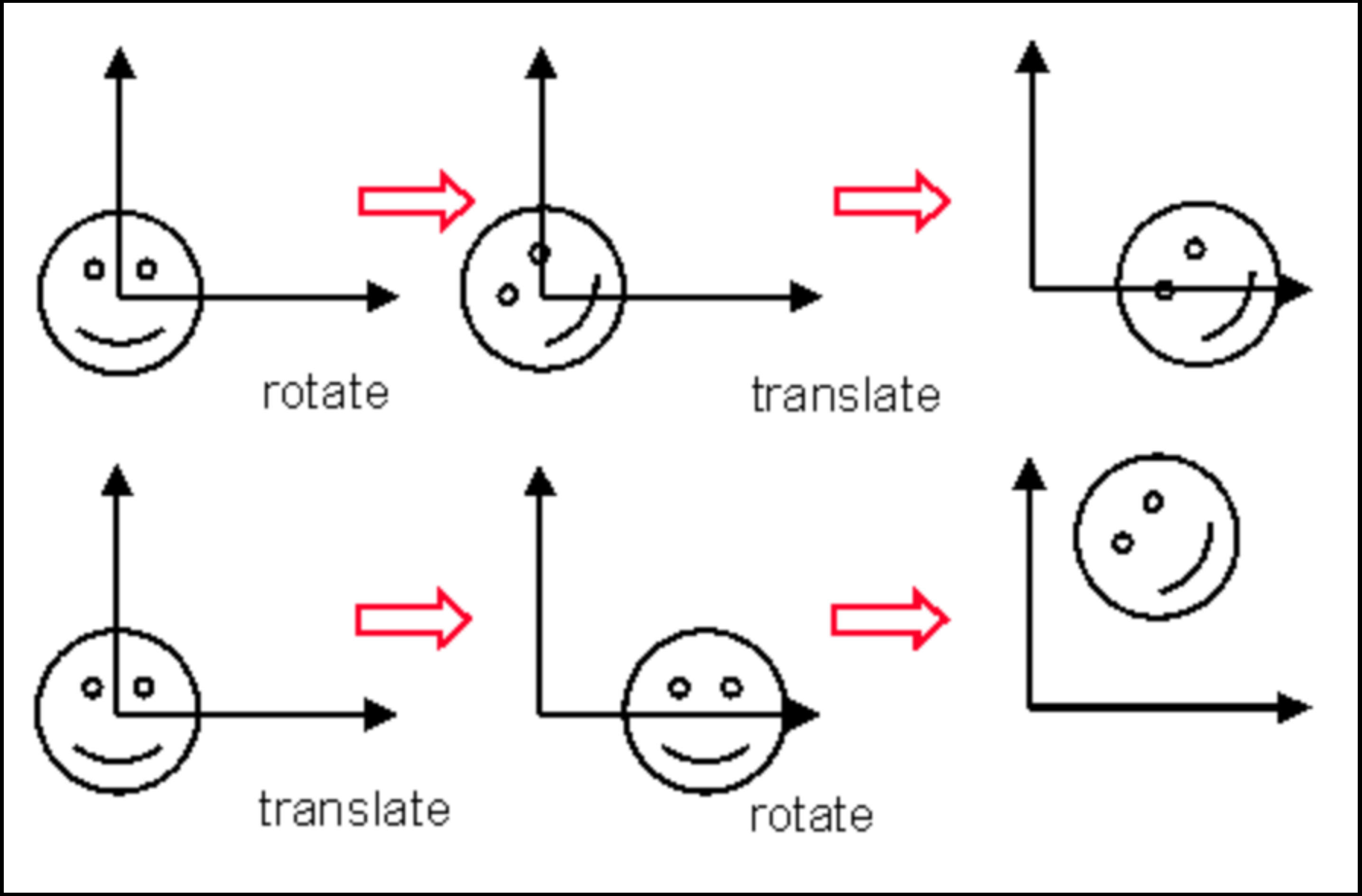
Combined

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{A}_S \cdot \mathbf{A}_R \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

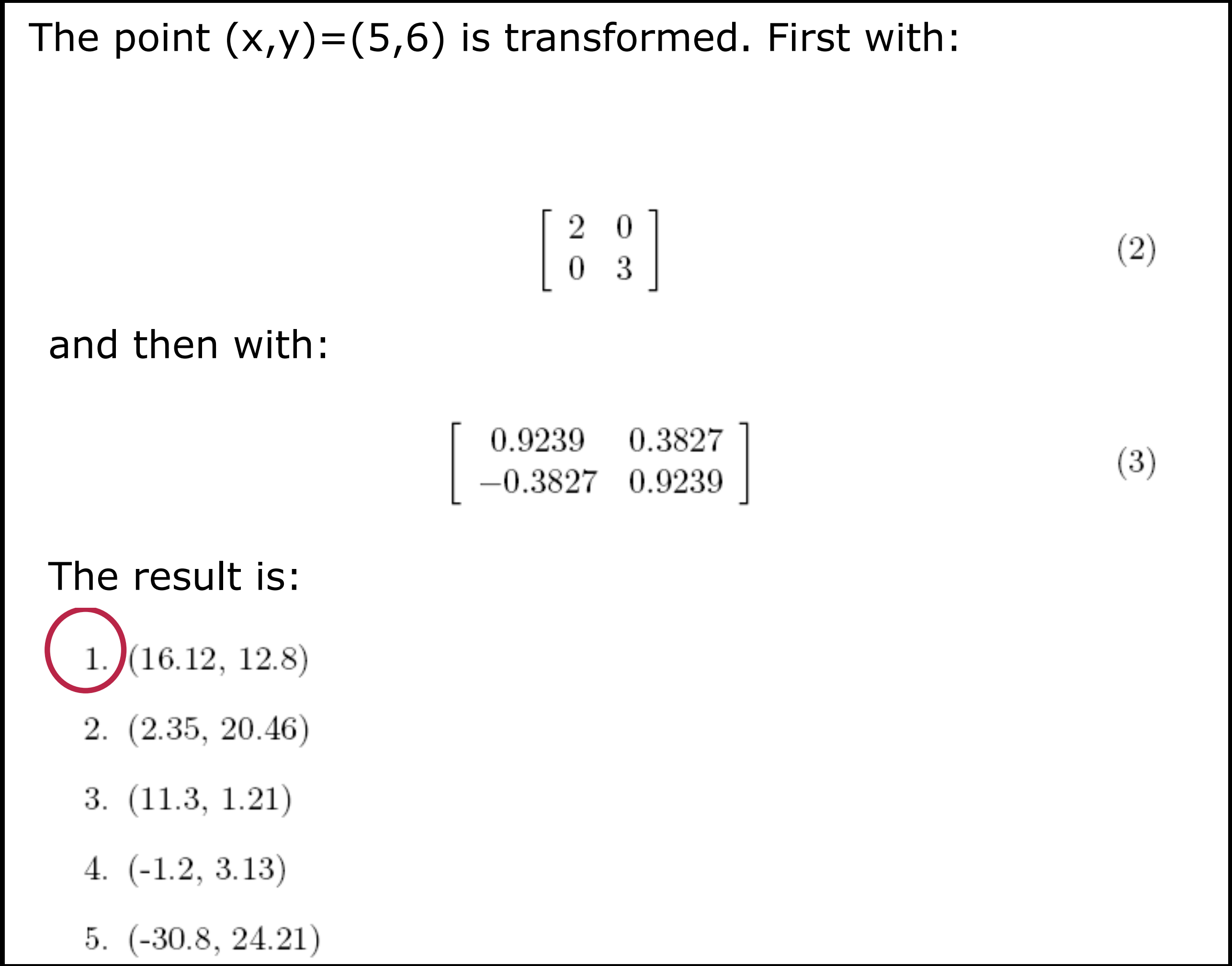
Remember: The order of matrix multiplications matters!

## Compositions of transformations

- Rigid body transformation: translation and rotation
- Composition of transformations matters = product of metrices
- Remember: order matters!



## Quiz 1: Combining transforms


The point (x,y)=(5,6) is transformed. First with:

$$\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \qquad (2)$$

and then with:

$$\begin{bmatrix} 0.9239 & 0.3827 \\ -0.3827 & 0.9239 \end{bmatrix} \qquad (3)$$

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:

$$\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 2*5+0*6 \\ 0*5+3*6 \end{bmatrix} = \begin{bmatrix} 10 \\ 18 \end{bmatrix}$$

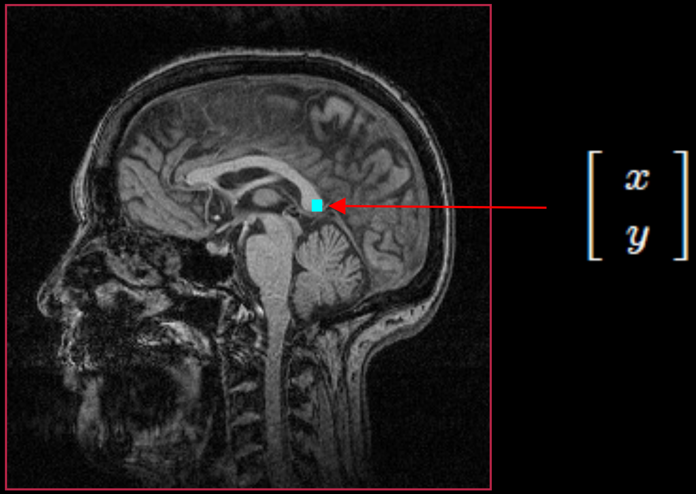
$$\begin{bmatrix} 0.9239 & 0.3827 \\ -0.3827 & 0.9239 \end{bmatrix} \begin{bmatrix} 10 \\ 18 \end{bmatrix} = \begin{bmatrix} 0.9239*10+0.3827*18) \\ -0.3827*10+0.9239*18) \end{bmatrix} = \begin{bmatrix} 16.12 \\ 12.8 \end{bmatrix}$$

## What do we have now?

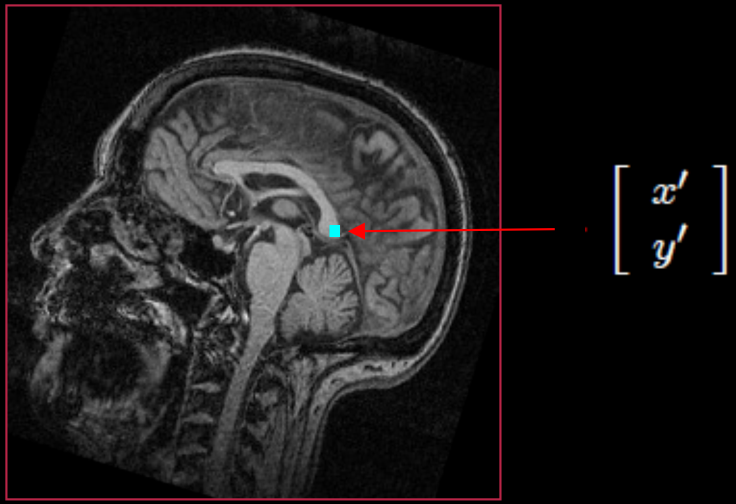
- We can pick a position in the input image f and find it in the output image g

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

We can transfer one pixel – what about the whole image?



f

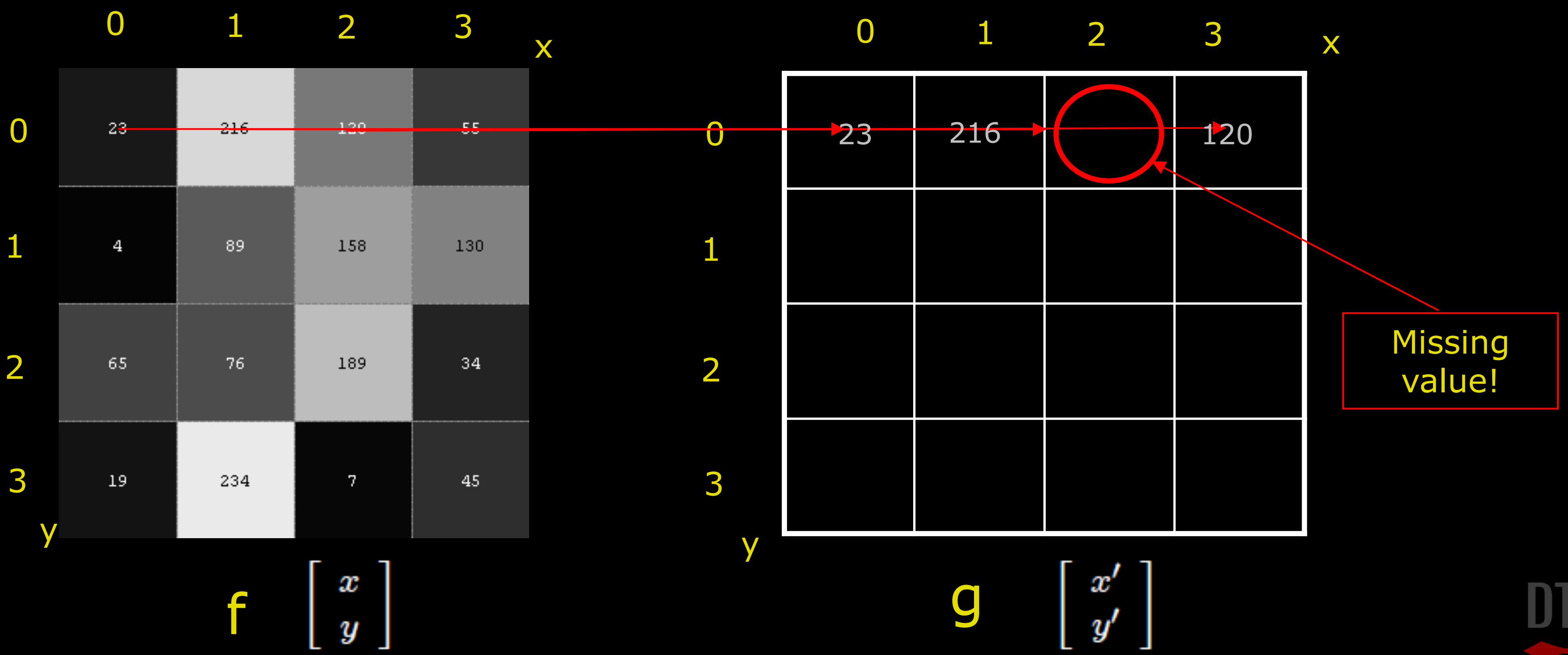


g

## 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

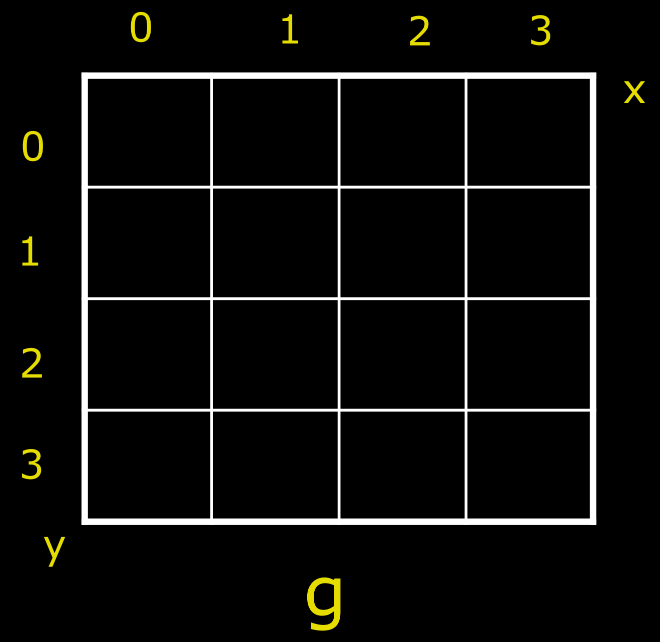
- The input to output transform is not good!
- It creates holes and other nasty looking stuff
- What do we do now?

## 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?

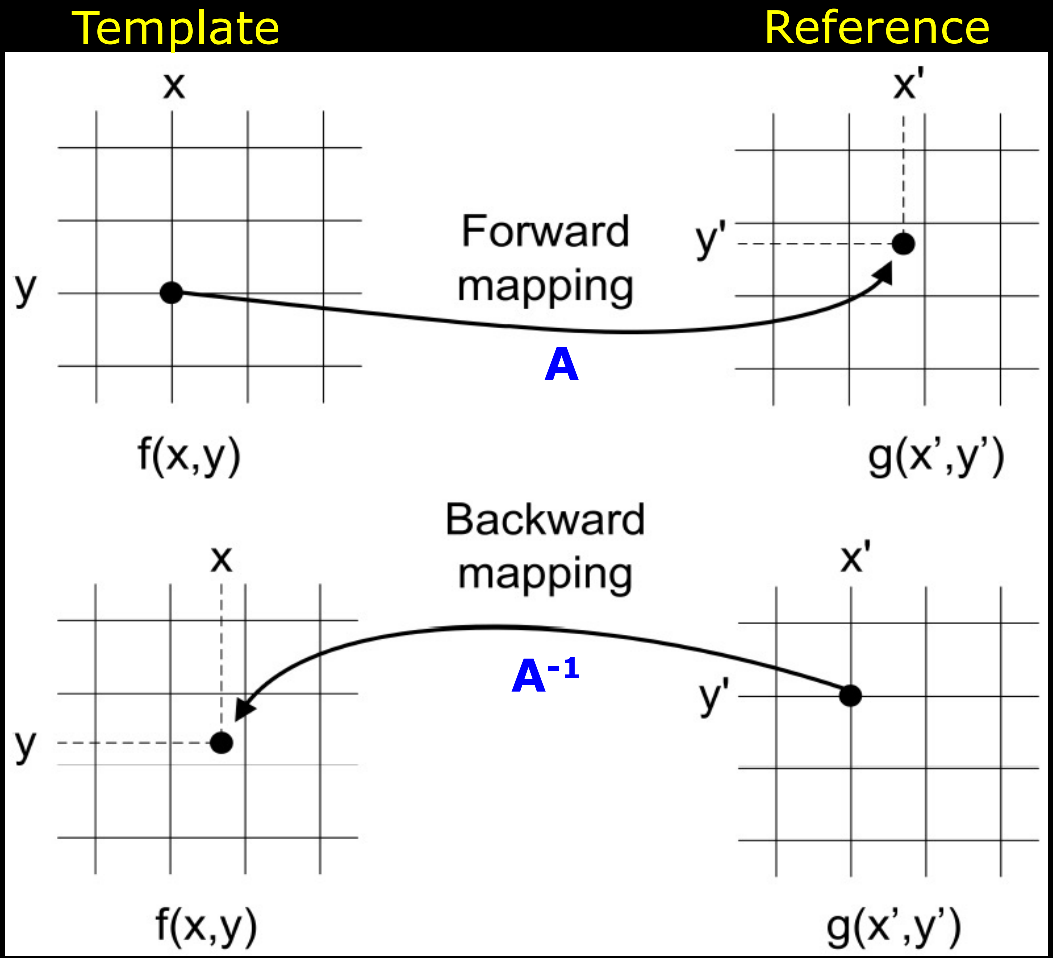
We need to go “backwards”

From the output to the input



## Forward vs. Backward mapping

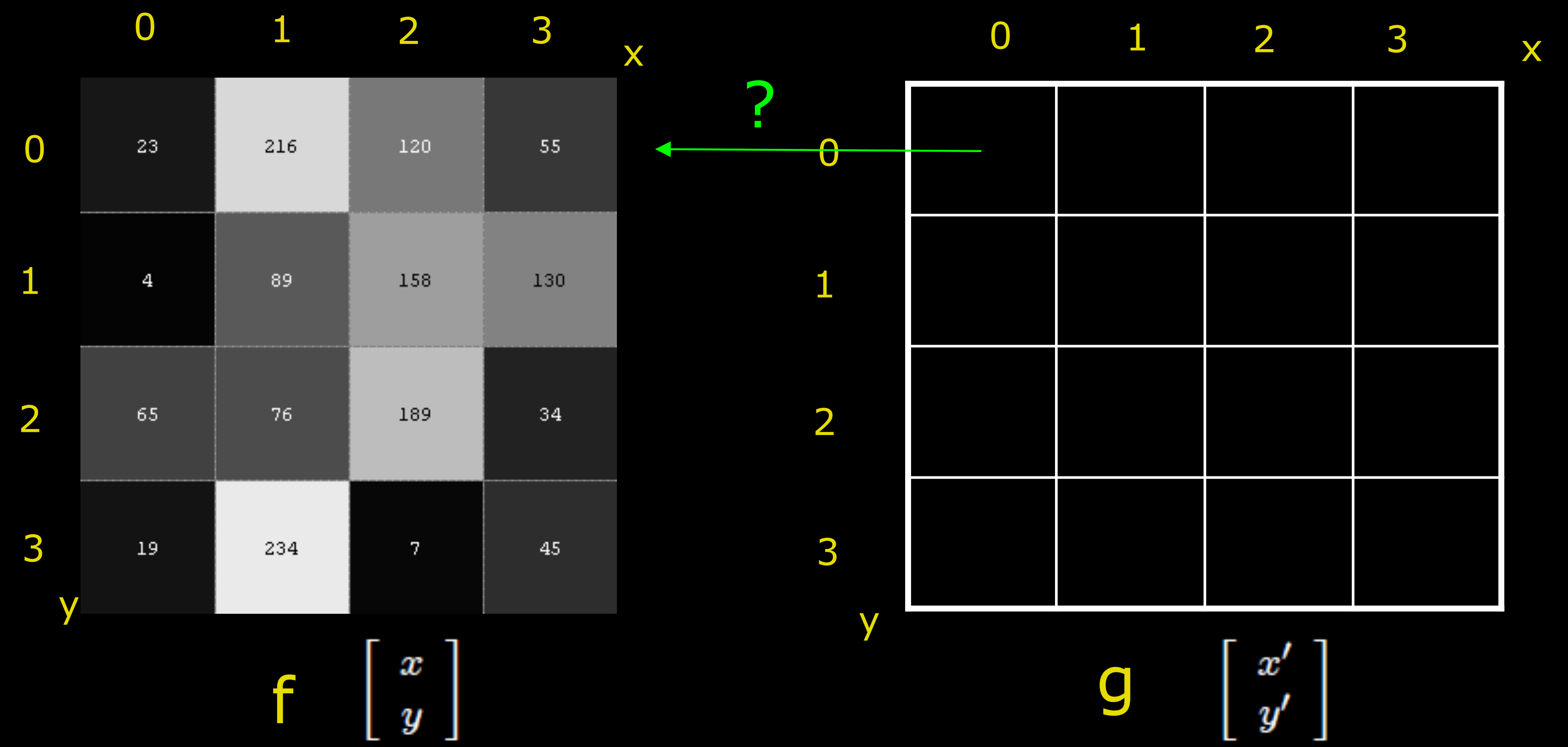
- In a nutshell
  - Going backward we need to invers the transformation



## Inverse transformation

- We want to go from the output to the input

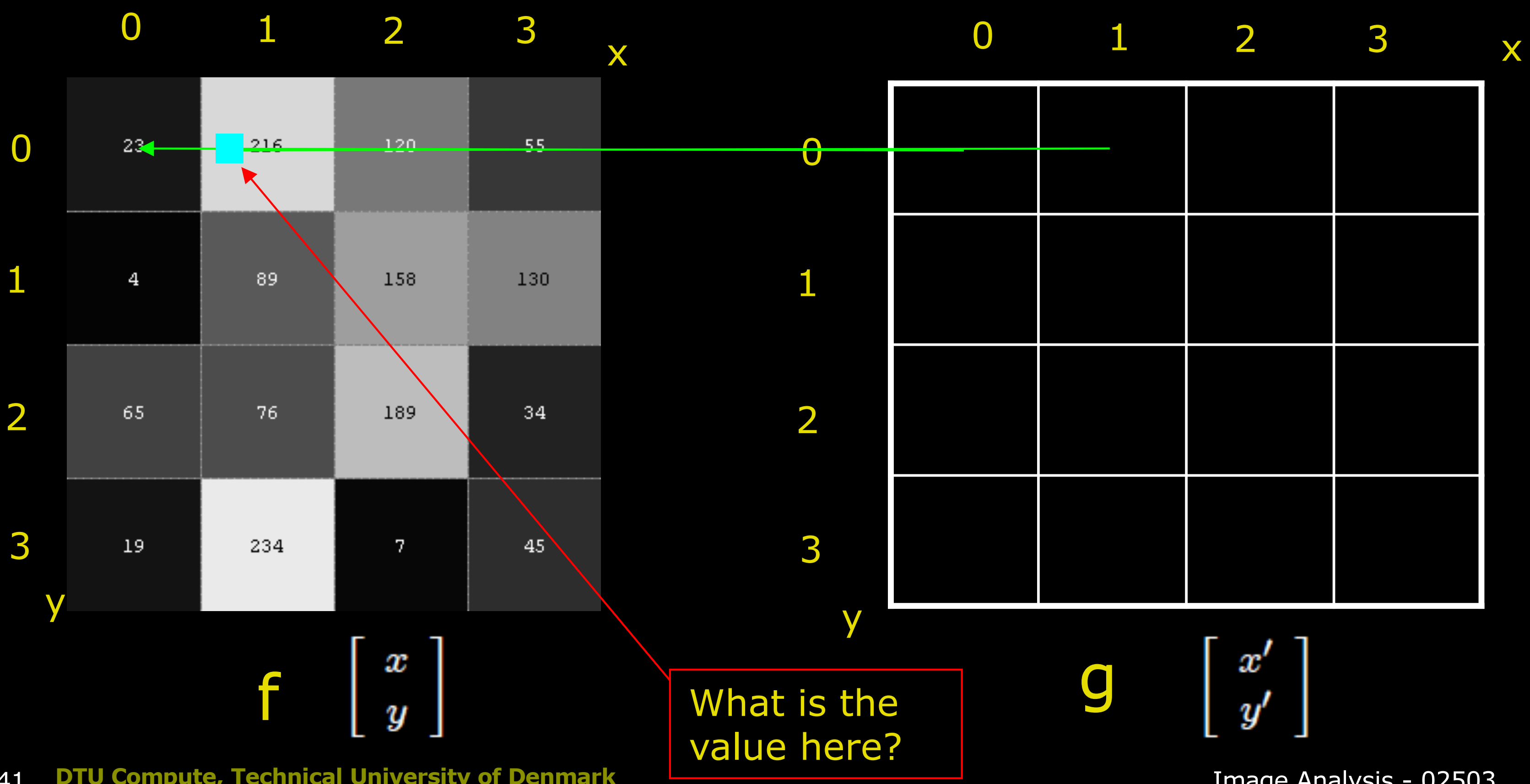
Scaling example $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$ $\xrightarrow{\text{inverse}}$ $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/1.5 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$


0
1
2
3
x
y
23
216
120
55
4
89
158
130
65
76
189
34
19
234
7
45
?
f
$\begin{bmatrix} x \\ y \end{bmatrix}$
g
$\begin{bmatrix} x' \\ y' \end{bmatrix}$

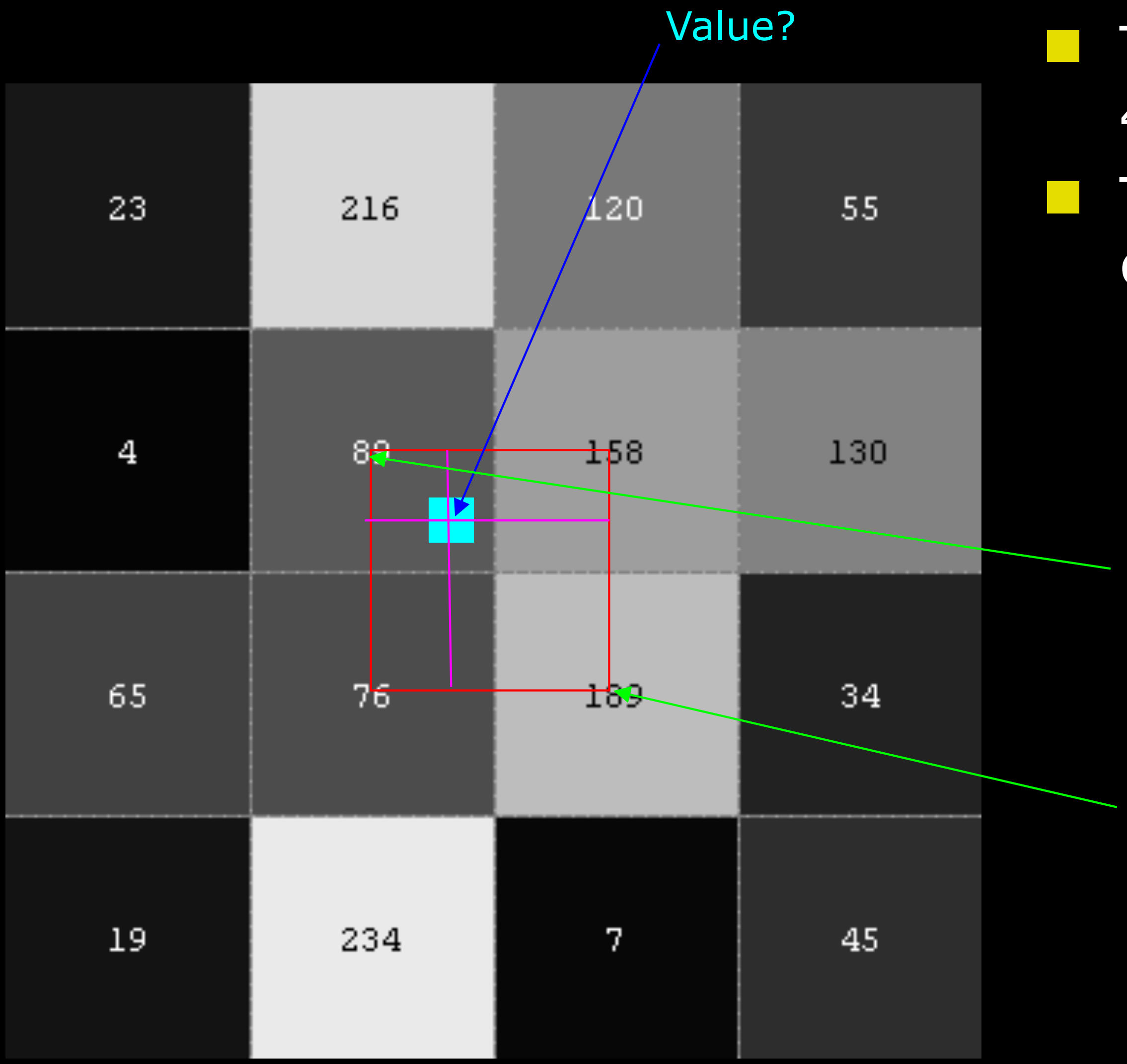
# Output-to-input transformation *Backward mapping*

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/1.5 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$$

- Run through all pixel in output image
- Find position in input image and *get the value*



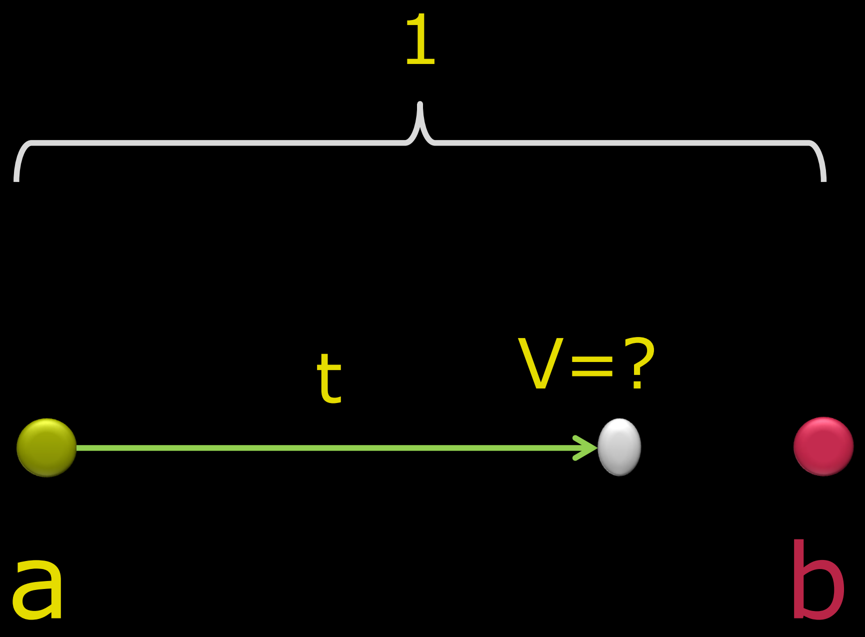
# Bilinear Interpolation



- The value is calculated from 4 neighbours
- The value is based on the distance to the neighbours

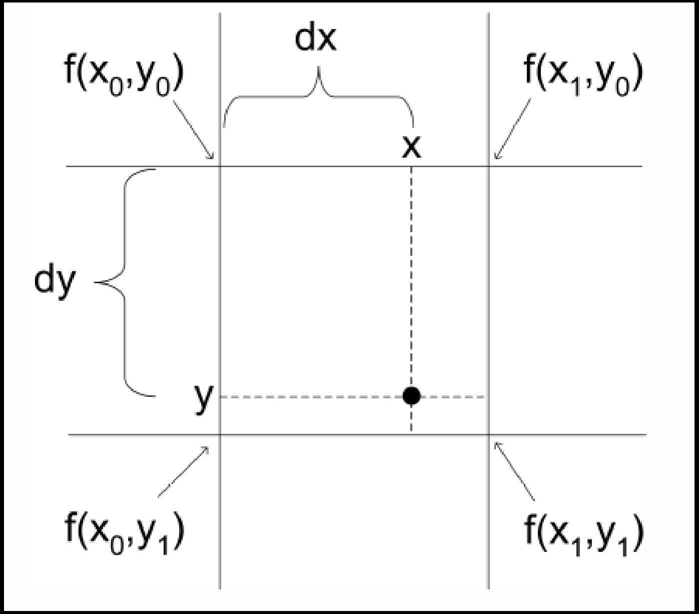
## Linear Interpolation (1D)

$$v = tb + (1 - t)a$$



## 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)

$g(173.1, 57.8)=$

$110*(1-0.1)*(1-0.8)+$

$140*(0.1)*(1-0.8)+$

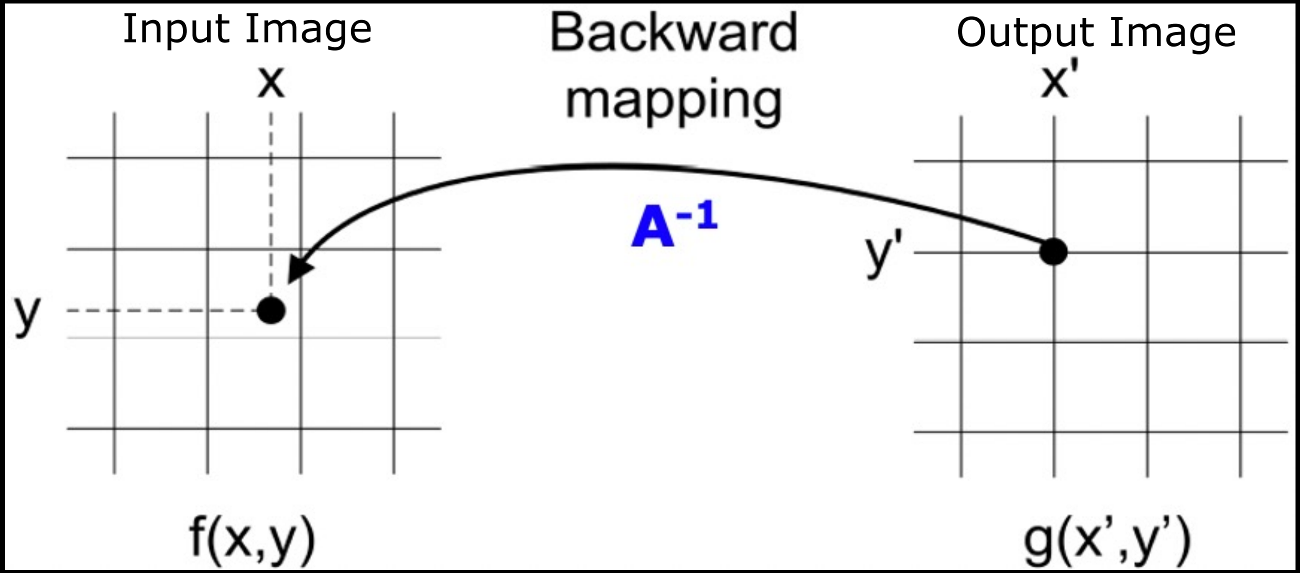
$156*(1-0.1)*(0.8)+$

$101*(0.1)*(0.8)$

$=143$

## 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

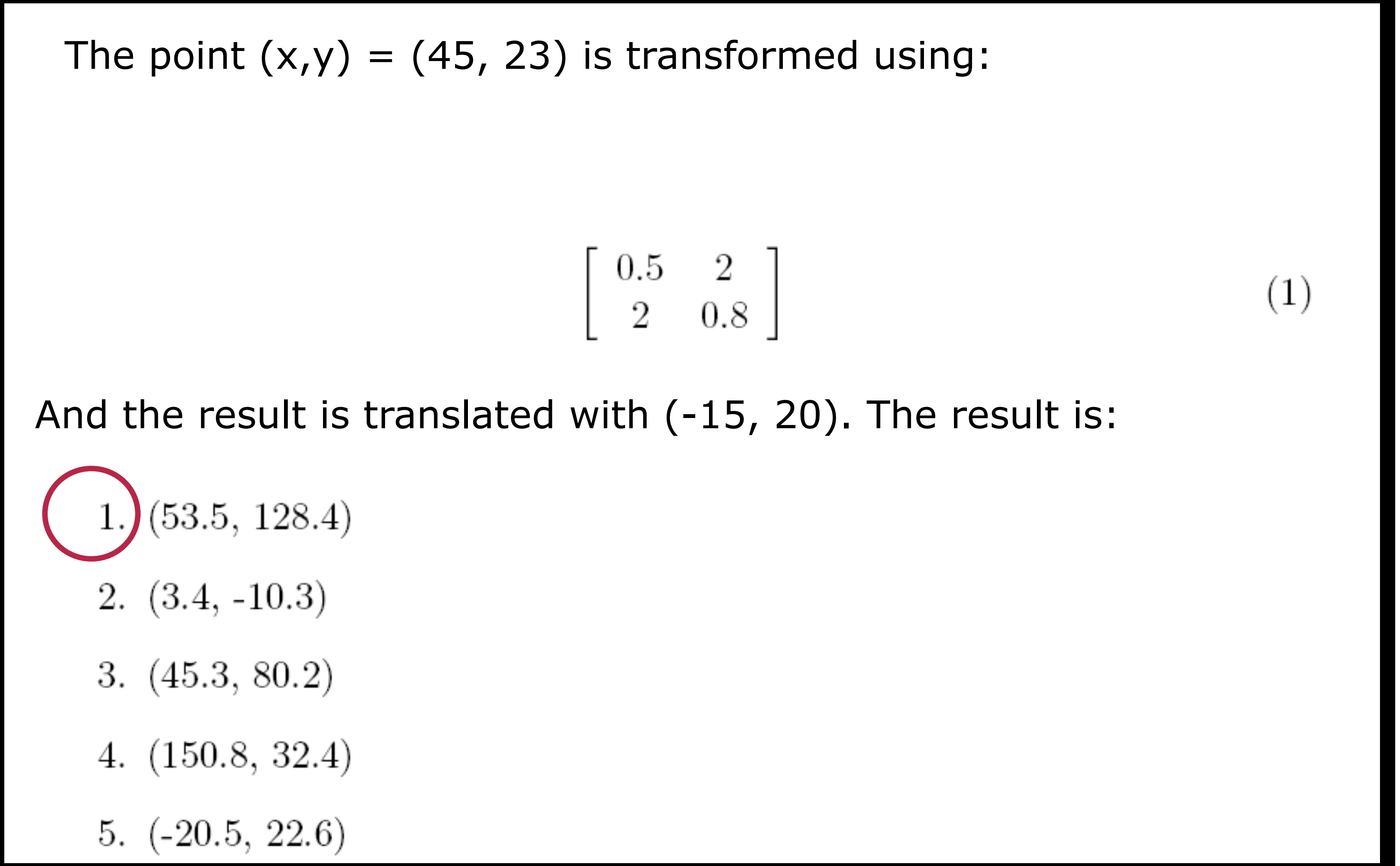
$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{A}^{-1} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$$

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:

$$\begin{bmatrix} 0.5 & 2 \\ 2 & 0.8 \end{bmatrix} \tag{1}$$

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:

$$(x', y') = \begin{bmatrix} -15 \\ 20 \end{bmatrix} + \begin{bmatrix} 0.5 & 2 \\ 2 & 0.8 \end{bmatrix} \begin{bmatrix} 45 \\ 23 \end{bmatrix}$$

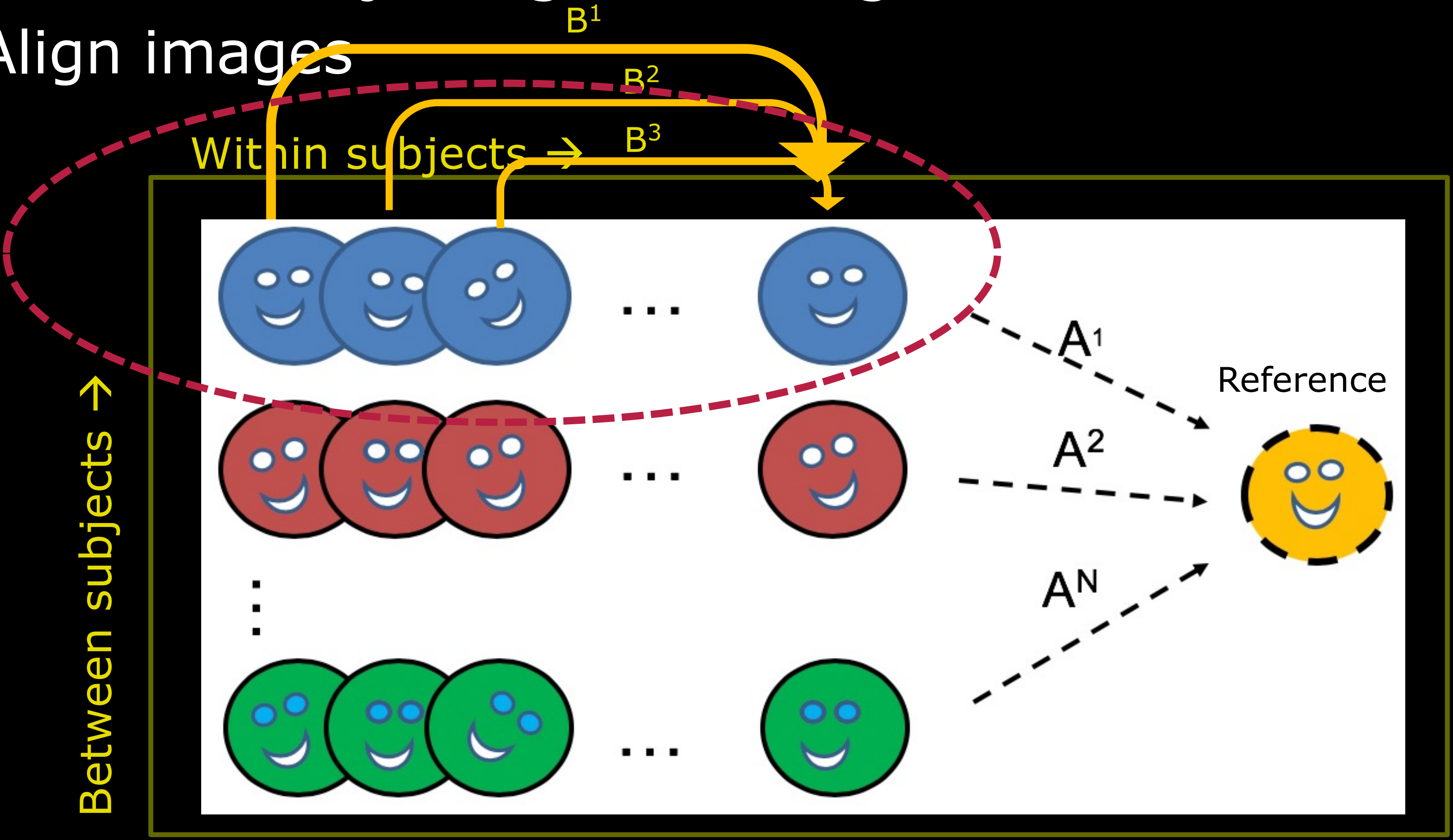
$$= \begin{bmatrix} -15 \\ 20 \end{bmatrix} + \begin{bmatrix} 68.5 \\ 108.4 \end{bmatrix}$$

$$= \begin{bmatrix} 53.5 \\ 128.4 \end{bmatrix}$$

## Image Registration

## Image Registration

- The act of adjusting something to match a standard
- Align images

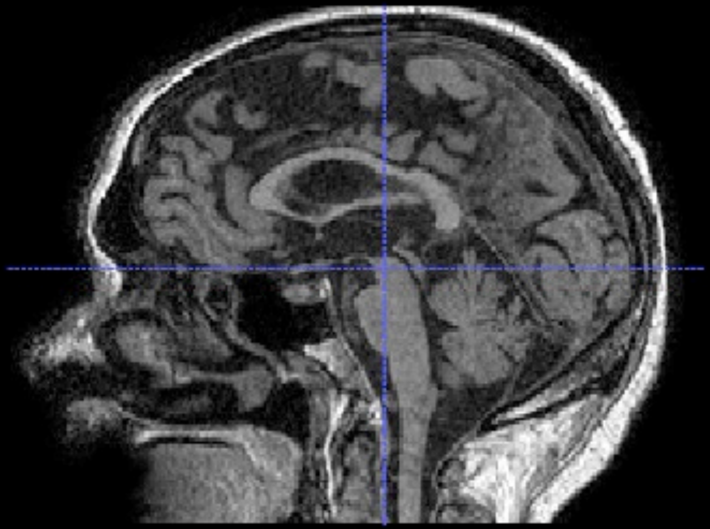


## Image registration

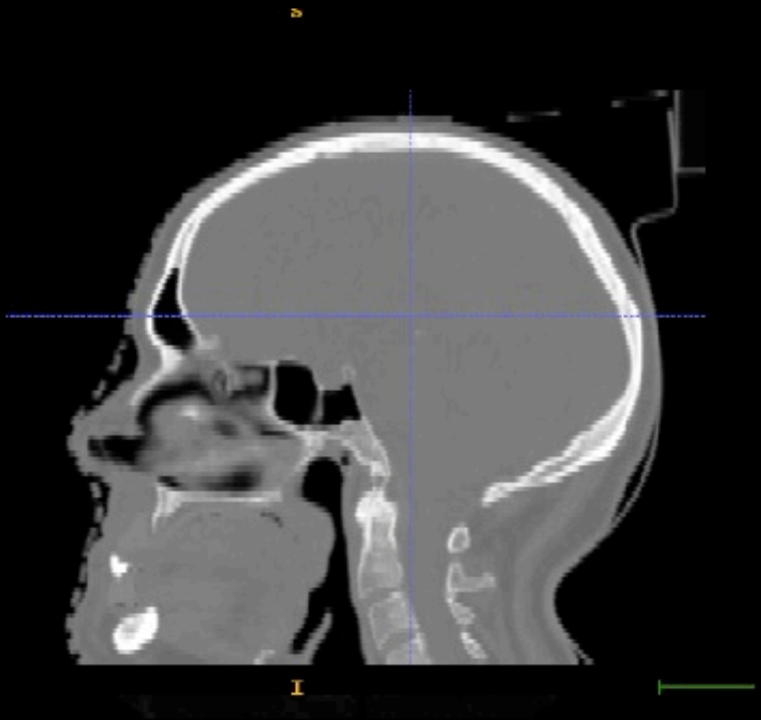
- Monitoring of change in the individual
- Fusion of information from different sources in a meaningful way
- Comparison of one subject with others
- Comparison of groups with others
- Comparing with an atlas

## Data fusion

Same patient – two scans



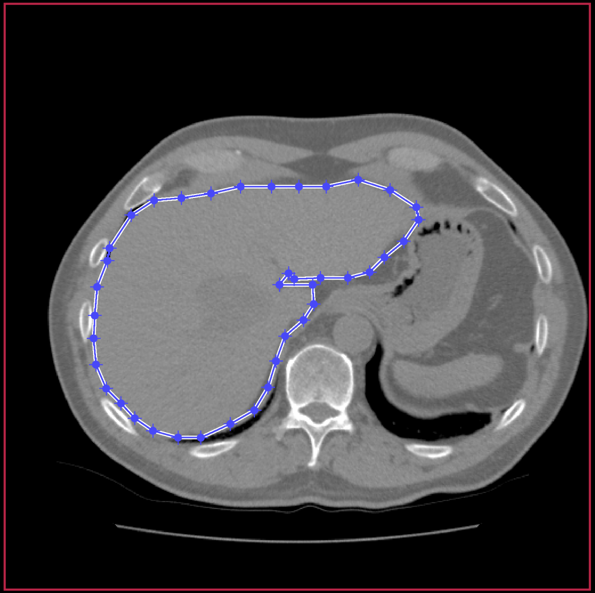
MR

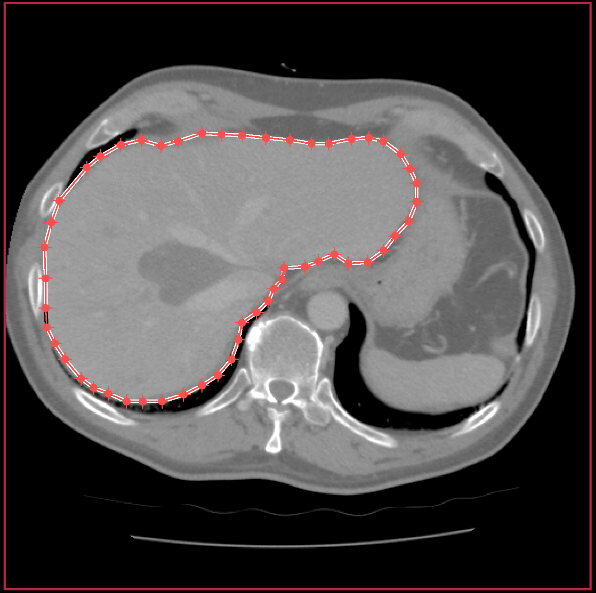


CT

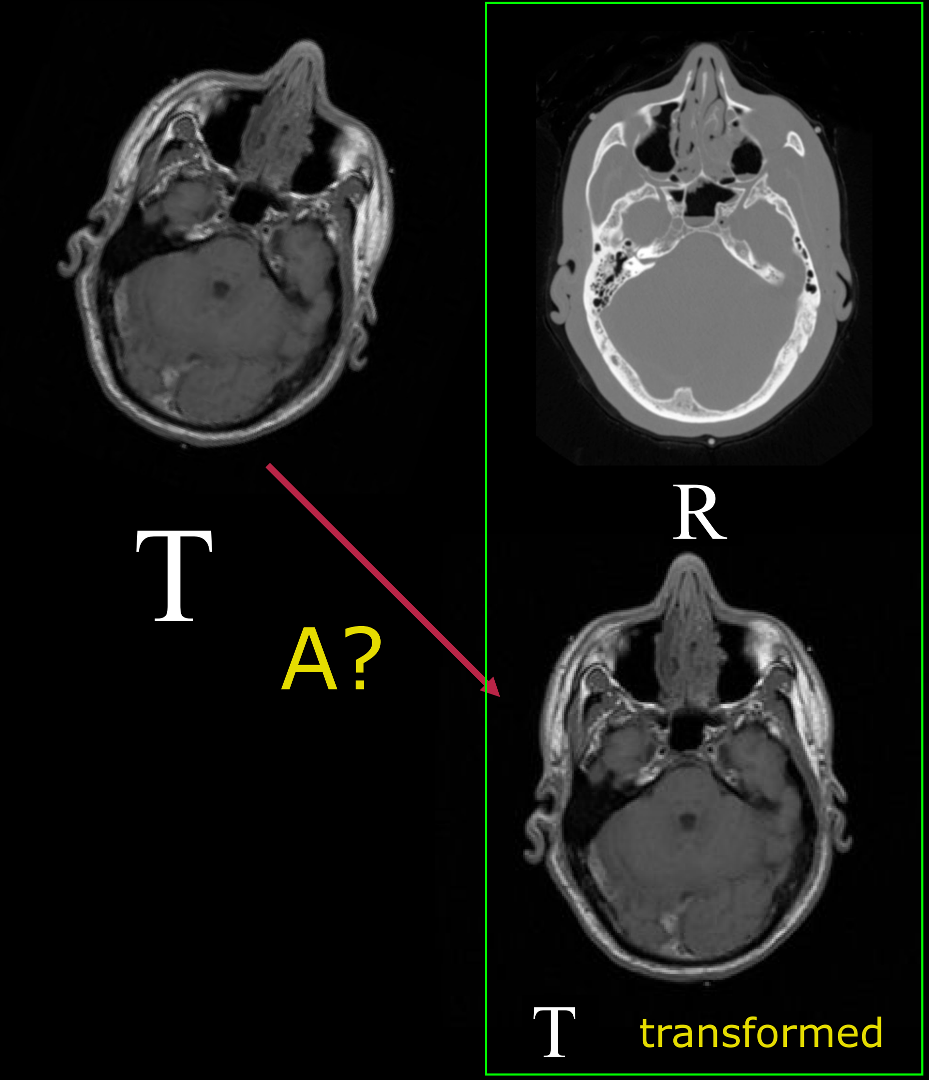
## Change detection

- Patient image before and after operation
- What has changed?
- Images need to be aligned before comparison


Before operation


After operation

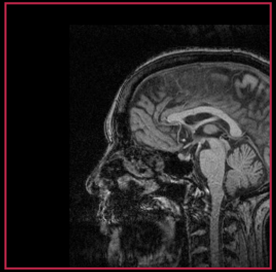
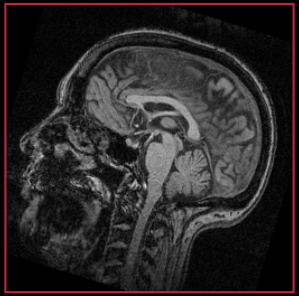
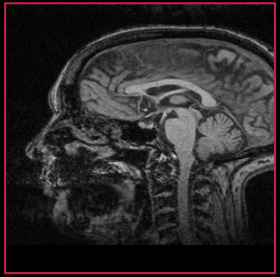
## Reference and template image



- 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

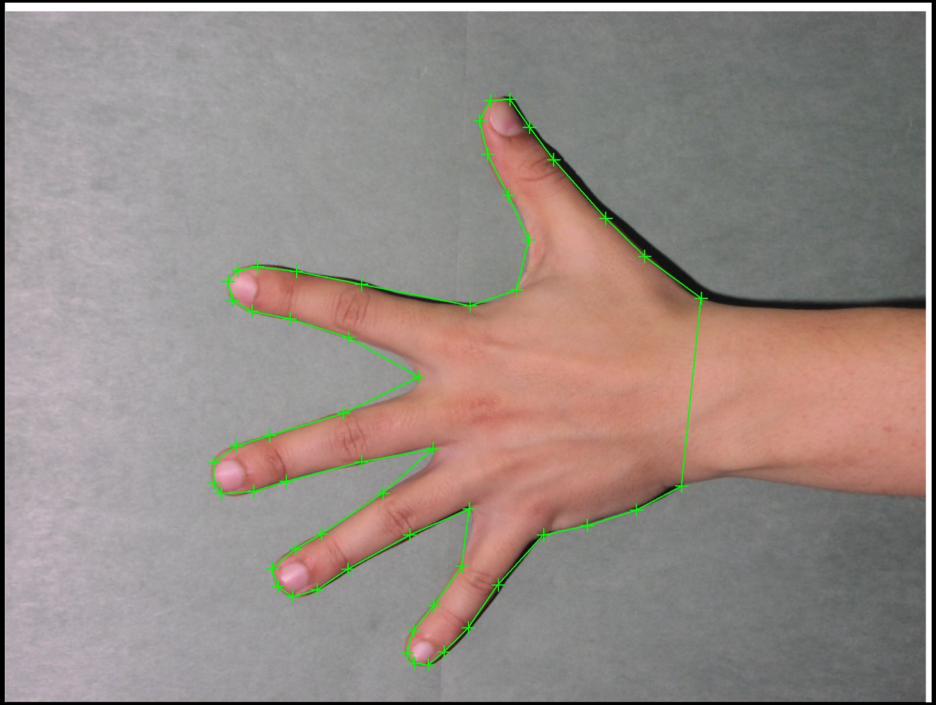




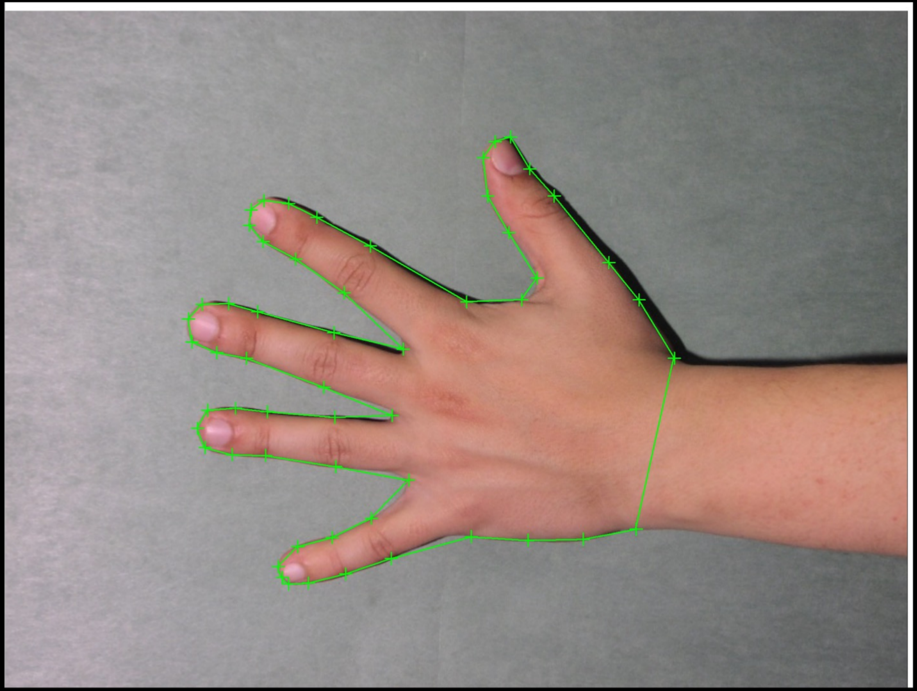
## Similarity measures

- 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

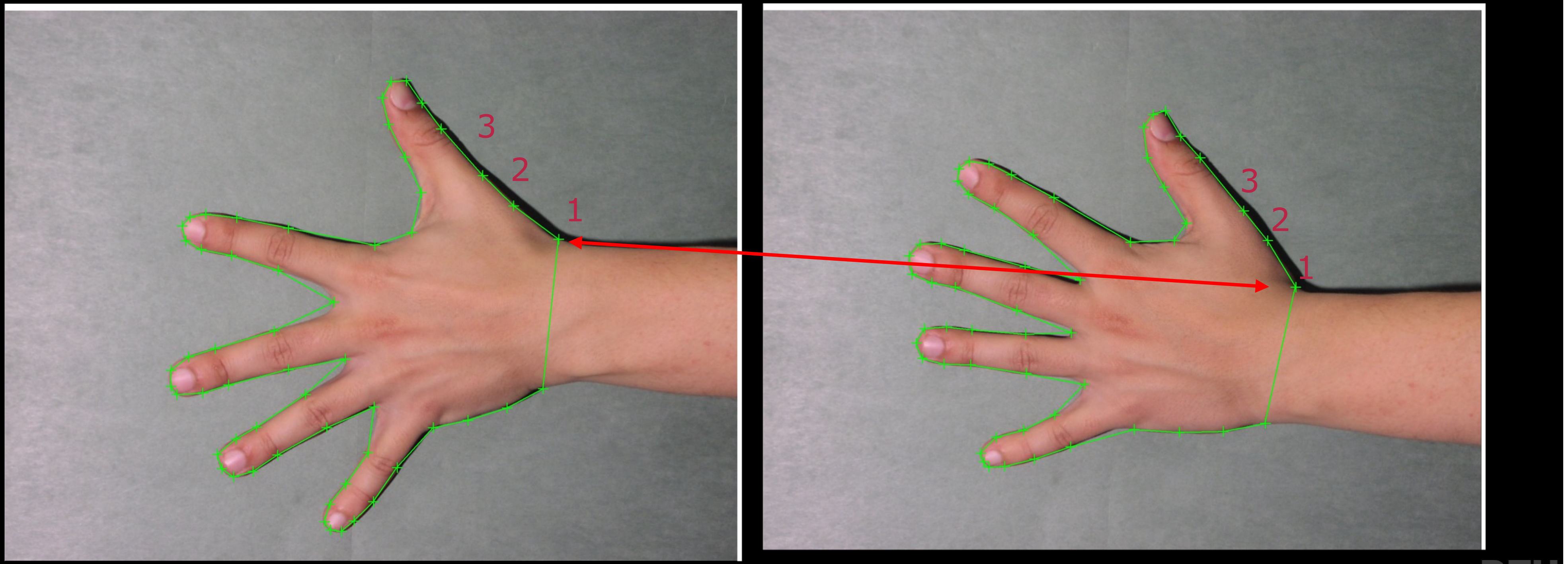
- Landmarks placed on both reference and template image
- The landmark should have *correspondence*



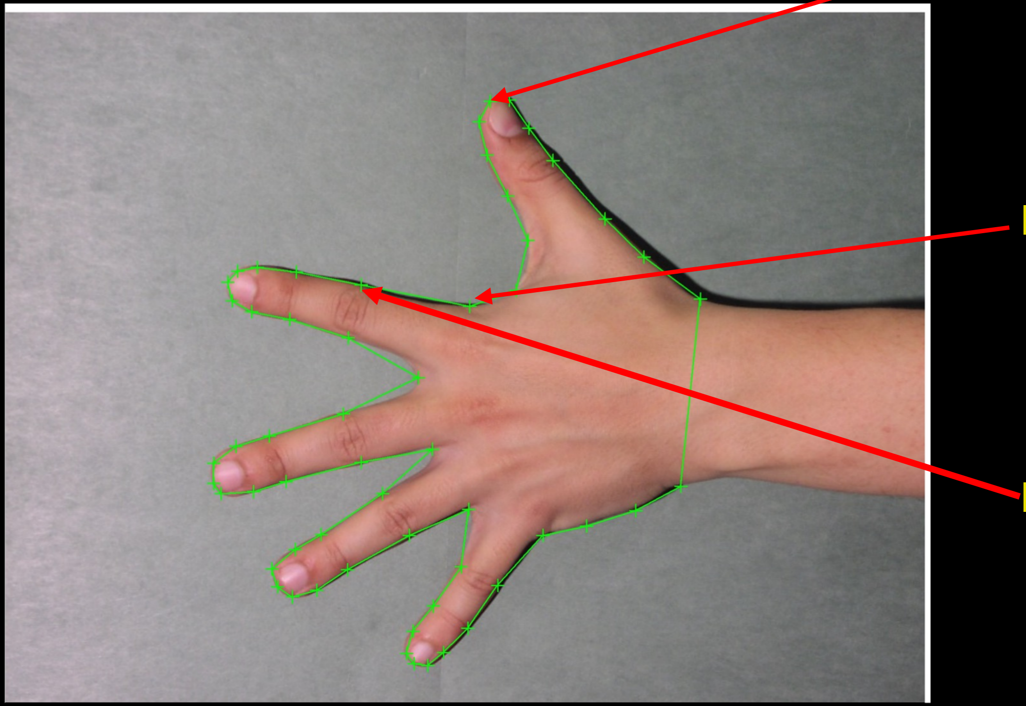


## Point correspondence

- Landmarks are numbered
- Each landmark should be placed the same place on both images



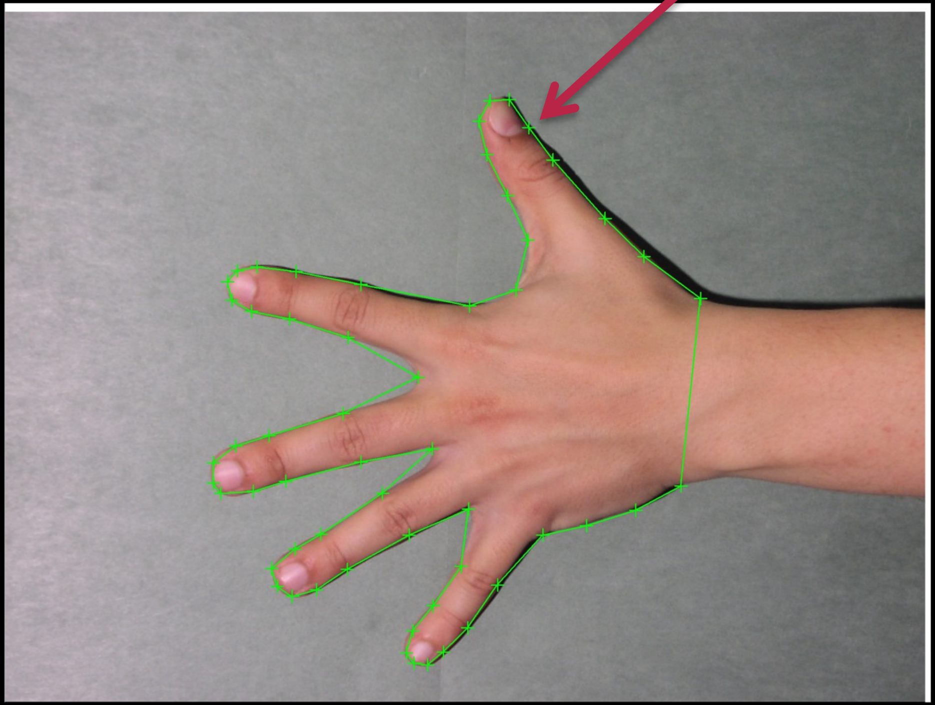
## Landmark types



- 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

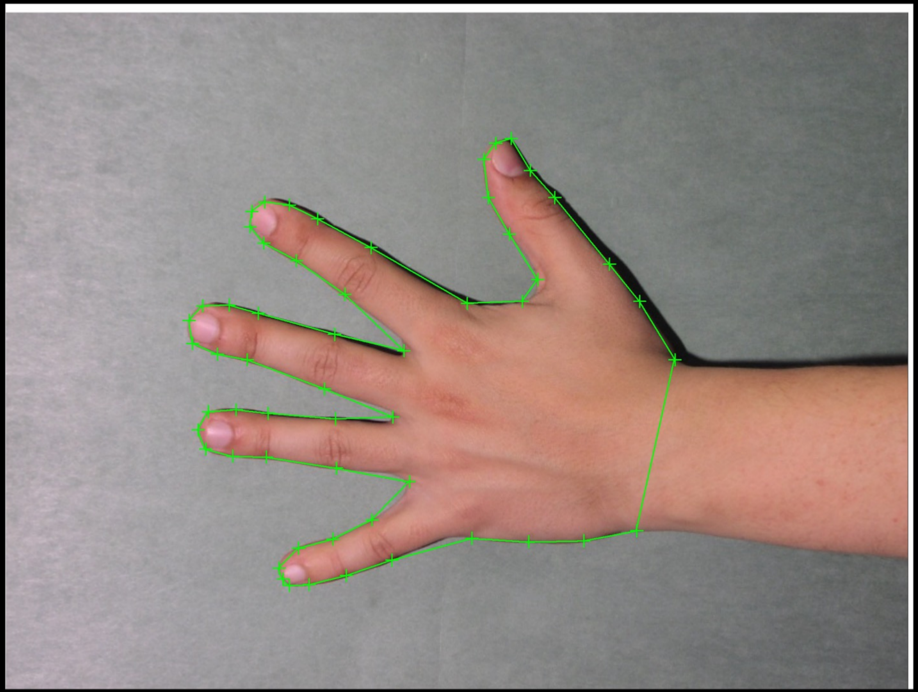
## Landmarks

$a_5 = (412, 55)$



$a_i$

Reference image $\mathrm{R}$



$b_i$

Template image $\mathrm{T}$

## The aim of registration

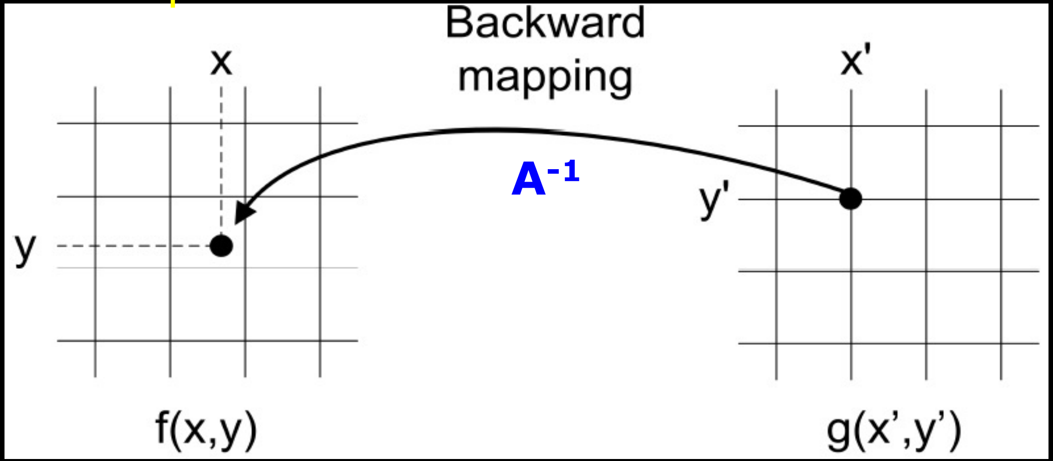
- We have selected Landmark points
- Find a transformation that maps the coordinates of the reference to the coordinates of the template
  - Why not the template to the reference?

    Sampling the template image:

    Backward mapping -> inverse transform

Template

Reference



## The Transformation

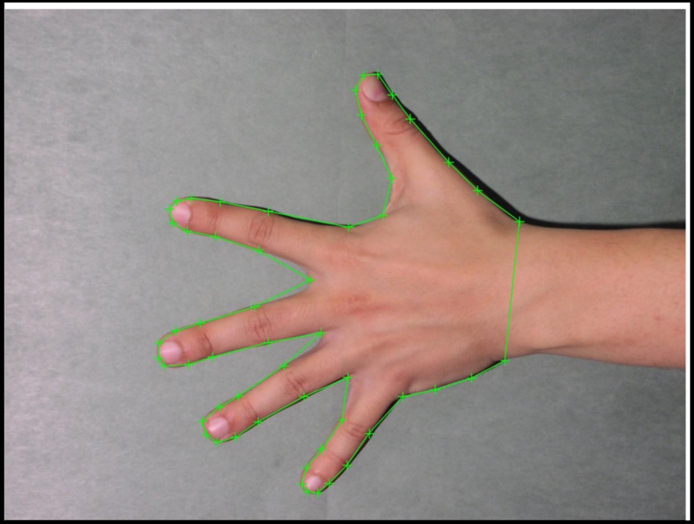
$$p' = T(p)$$

- 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

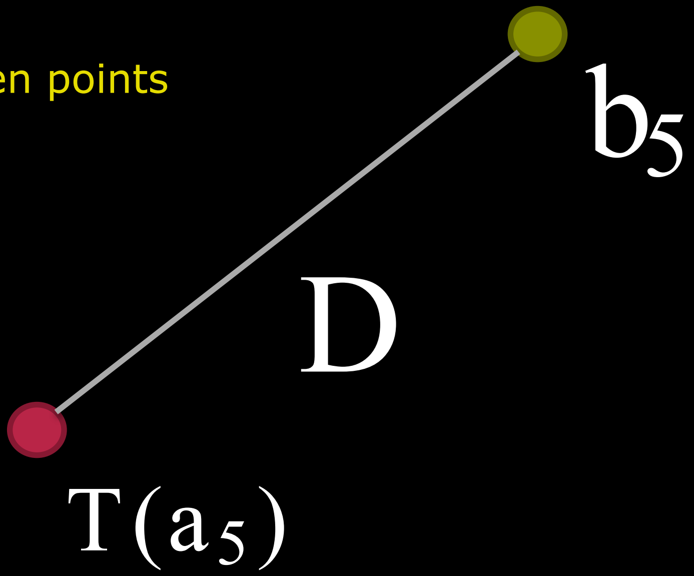
- The *objective function* measures how well two point sets match
- It uses a *cost function that describe how to evaluate the match*
- Here the cost function is a sum-of-squares distance function
- Point sets could be landmarks

## Objective function

$$F = \sum_{i=1}^{N} D(T(a_i), b_i)^2$$

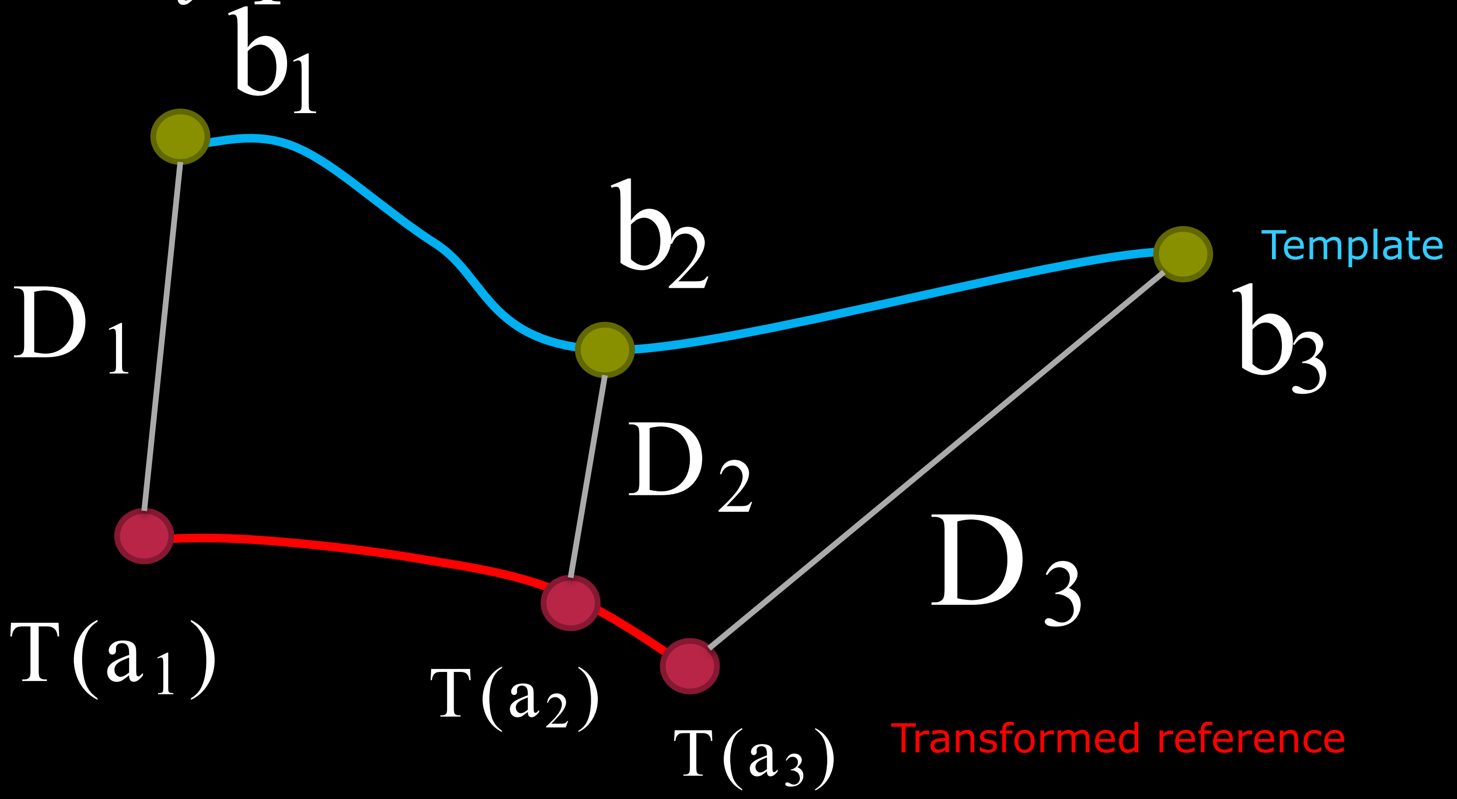
Distance between points

- The *objective function* measures how well two point sets match



## Objective function

$$F = \sum_{i=1}^{3} D(T(a_i), b_i)^2 = D_1^2 + D_2^2 + D_3^2$$

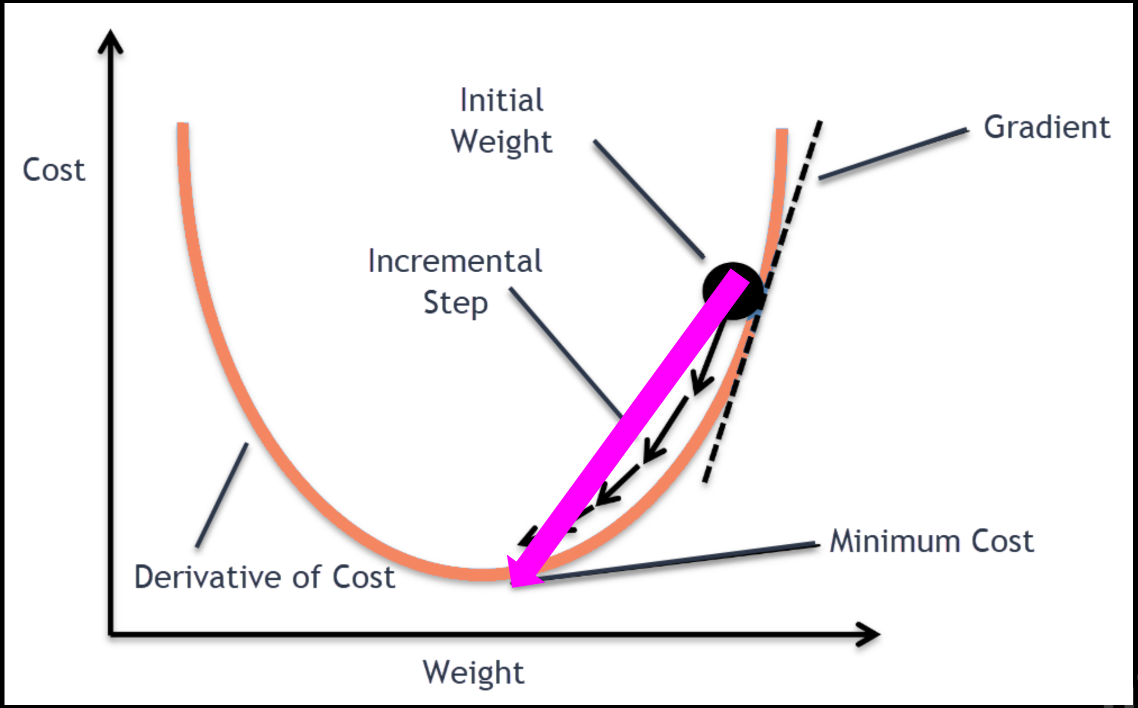


## Minimization / Optimization

$$F = \sum_{i=1}^{N} D(T(a_i), b_i)^2$$

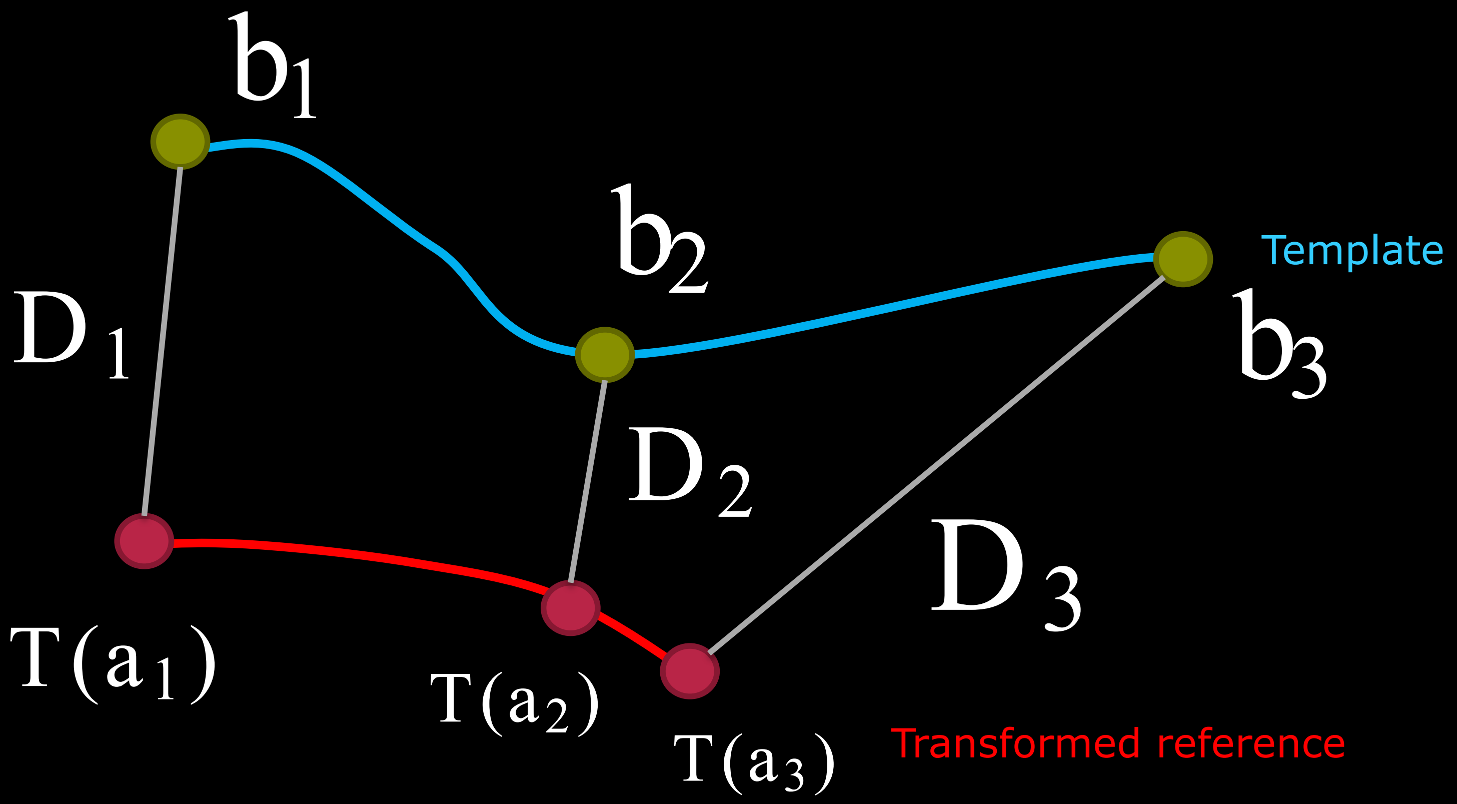
$$\boxed{\hat{w} = \arg\min_{w} F}$$

- Find the set of parameters that minimizes the objective function
- Optimisation strategy: Analytic (exact solution) vs Numerical?



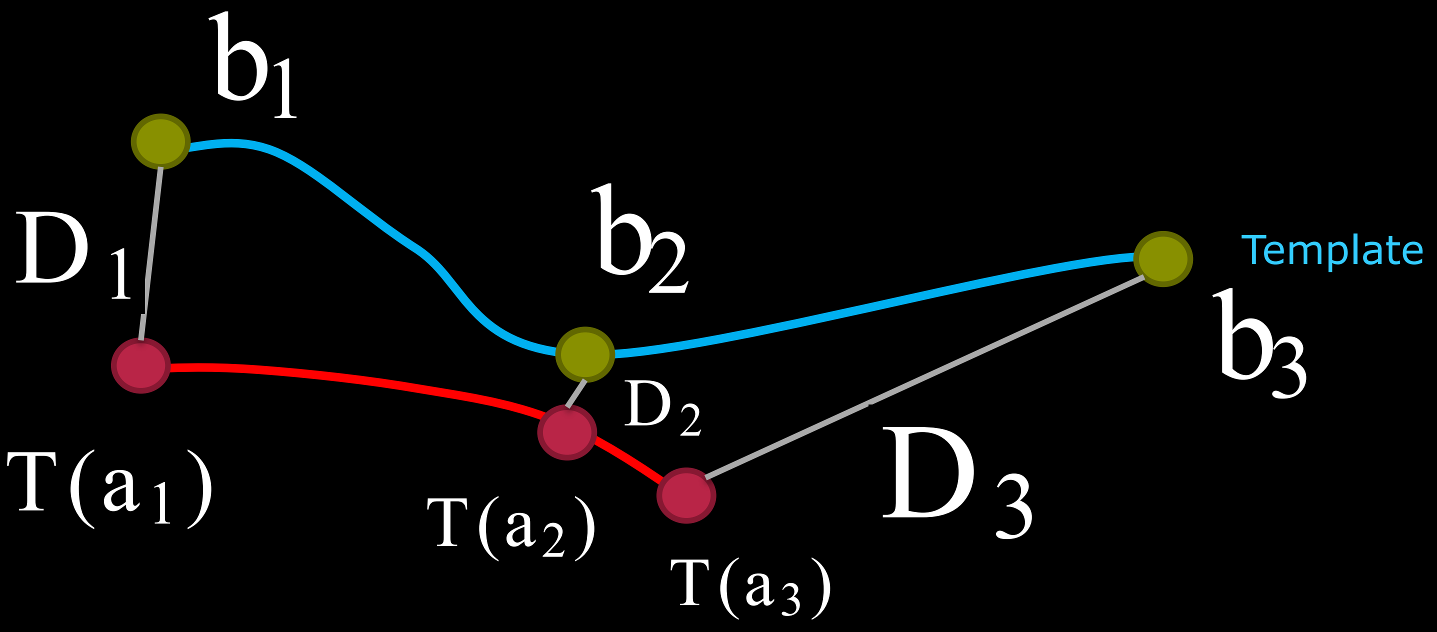
## Minimization – pure translation

$$F = D_1^2 + D_2^2 + D_3^2$$



## Minimization – pure translation

$$F = D_1^2 + D_2^2 + D_3^2$$ Decreased!

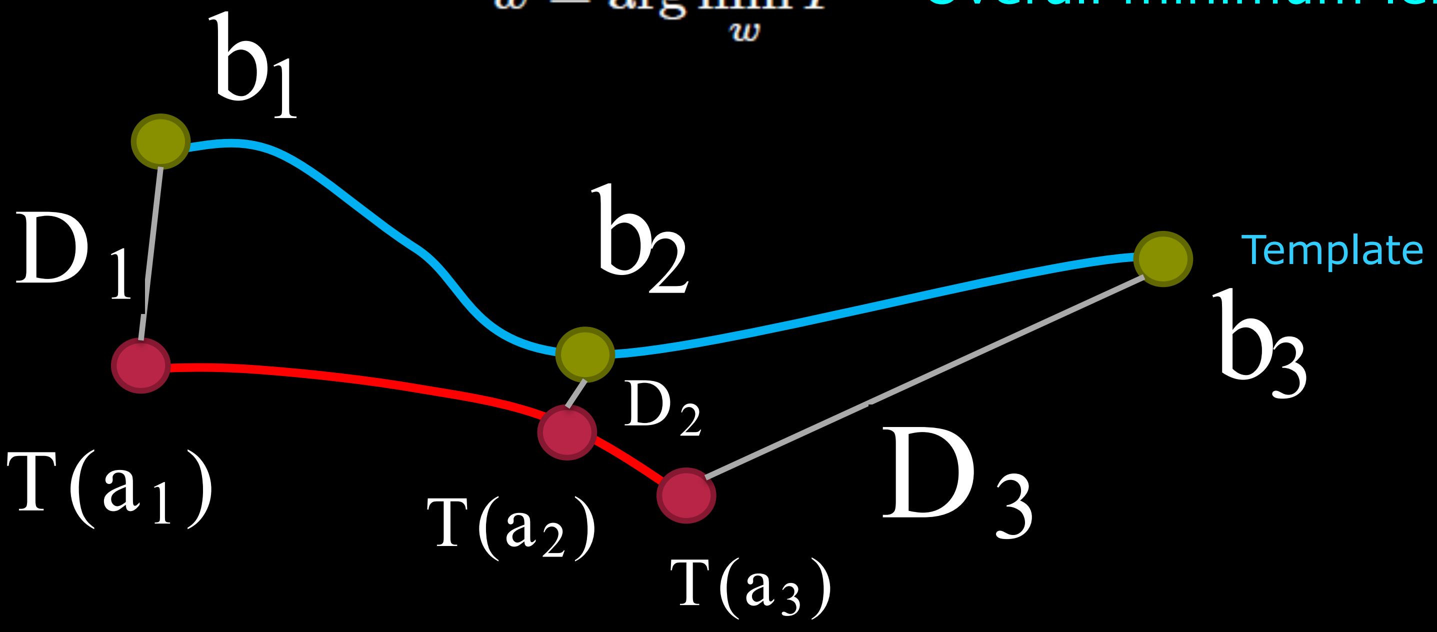


Transformed reference

## Minimization – pure translation

$F = D_1^2 + D_2^2 + D_3^2$ Decreased!

$\hat{w} = \arg\min_{w} F$ Overall minimum lengths!

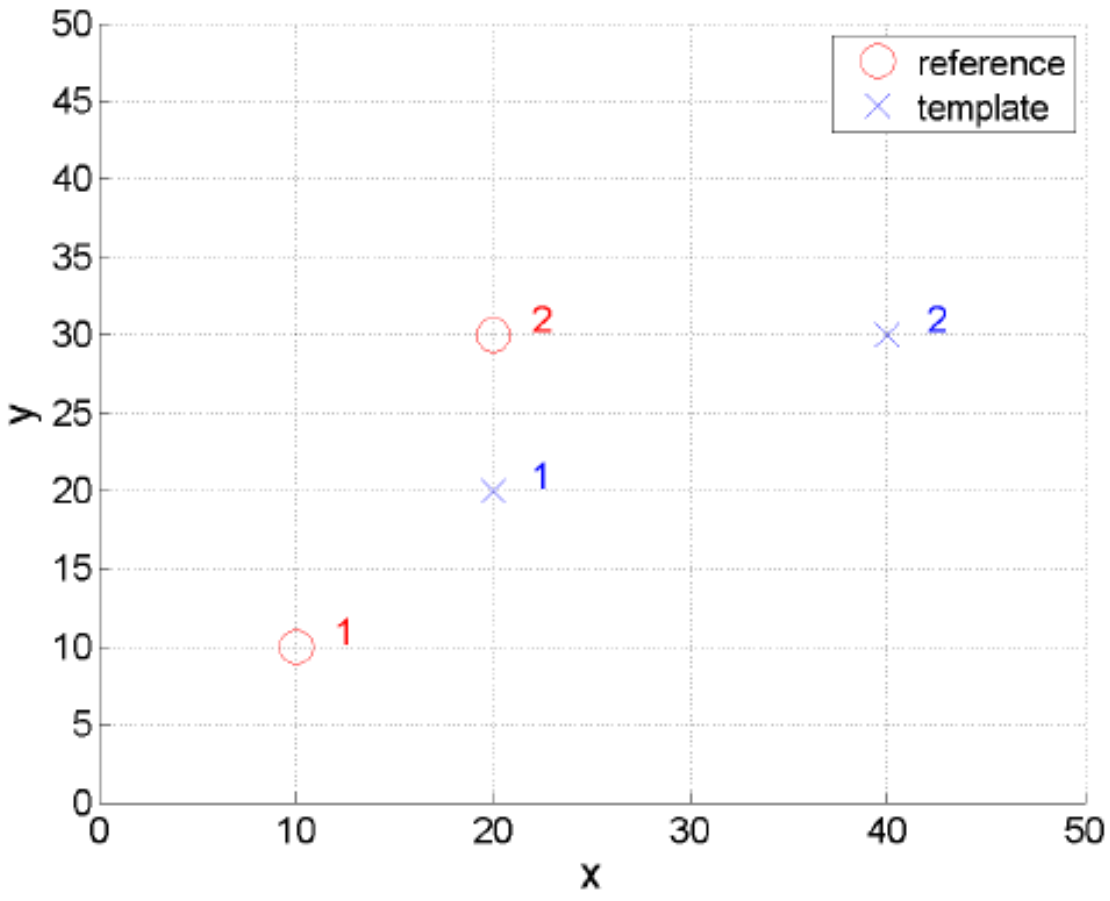


Transformed reference

## Quiz 5: Objective function

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:

[✓ A) 600]
B) 50
C) 100
D) 900
E) 300



Solution:

$$D1^2 = \left\| \begin{bmatrix}10\\10\end{bmatrix} - \begin{bmatrix}20\\20\end{bmatrix} \right\|^2 = \left\| \begin{matrix}10\\10\end{matrix} \right\|^2 = 200$$

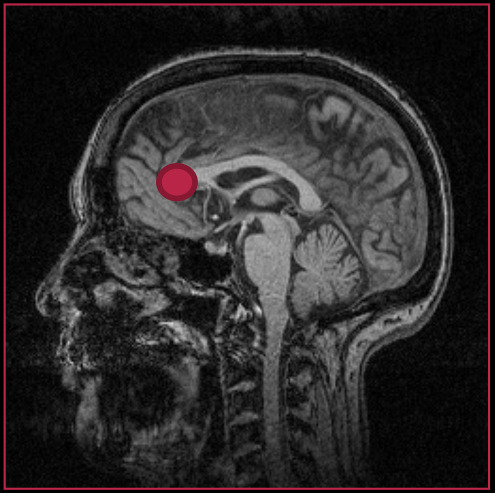
$$D2^2 = \left\| \begin{bmatrix}20\\30\end{bmatrix} - \begin{bmatrix}40\\30\end{bmatrix} \right\|^2 = \left\| \begin{matrix}20\\0\end{matrix} \right\|^2 = 400$$

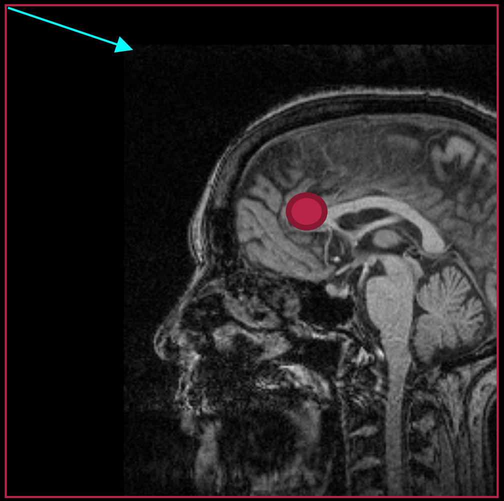
## Translation

- 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

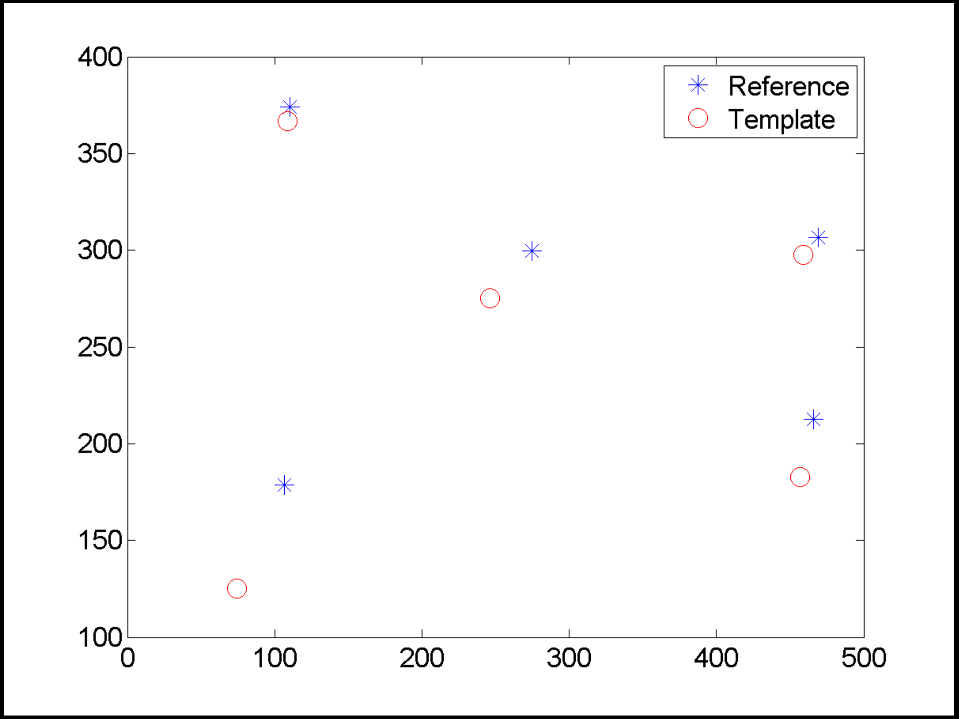
$$w = (\Delta x, \Delta y) = t$$

Optimal translation

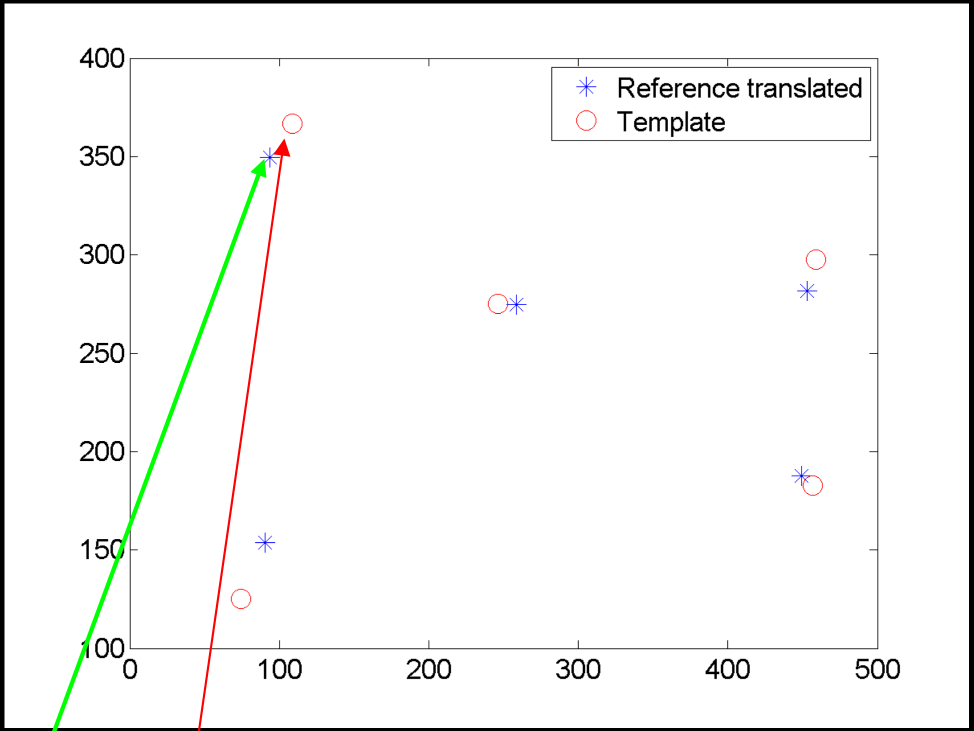
$$\hat{t} = \boxed{\bar{b}} - \bar{a} \qquad \bar{a} = \frac{1}{N}\sum_{i=1}^{N} a_i$$

Average point = centre of mass

# Optimal translation



Original landmarks

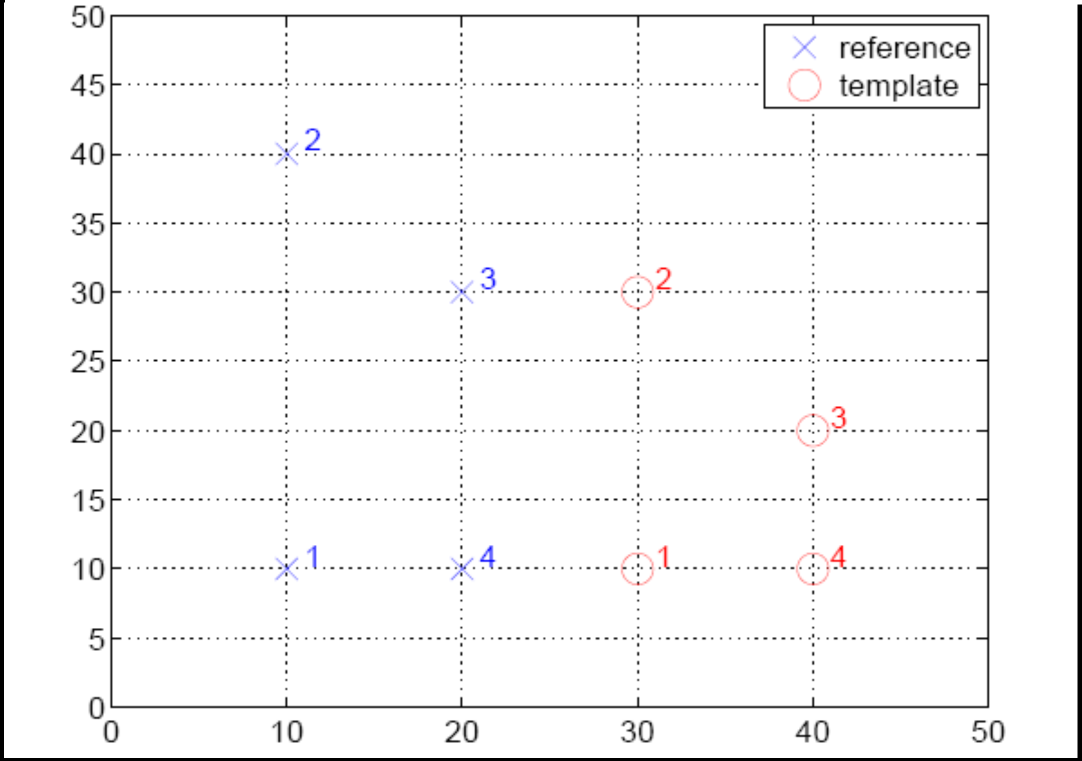


Reference points translated

$$F = \sum_{i=1}^{N} \|(\underbrace{a_i + t}) - \underbrace{b_i}\|^2$$

# Quiz 6: Optimal translation

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?



A) (-10, 10)

B) (20,5)

[✓ C) (20,-5)]

D) (15,-5)

Solution:

$$\hat{t} = \bar{b} - \bar{a}$$

$$\bar{a} = \frac{1}{4}\left(\begin{bmatrix}10\\10\end{bmatrix} + \begin{bmatrix}10\\40\end{bmatrix} + \begin{bmatrix}20\\30\end{bmatrix} + \begin{bmatrix}20\\10\end{bmatrix}\right) = \begin{bmatrix}15\\22{,}5\end{bmatrix}$$

$$\bar{b} = \frac{1}{4}\left(\begin{bmatrix}30\\10\end{bmatrix} + \begin{bmatrix}30\\40\end{bmatrix} + \begin{bmatrix}40\\20\end{bmatrix} + \begin{bmatrix}40\\10\end{bmatrix}\right) = \begin{bmatrix}35\\17{,}5\end{bmatrix}$$

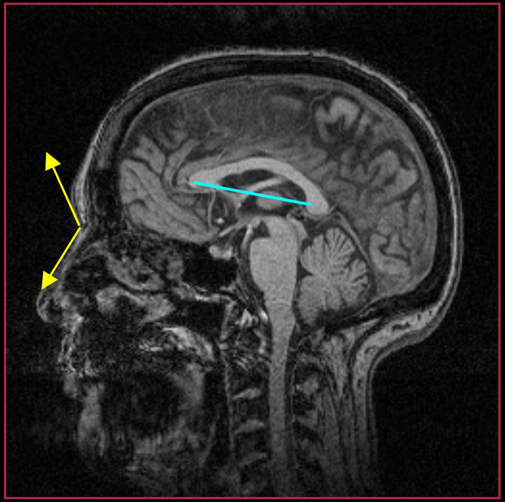
## Rigid body transformation

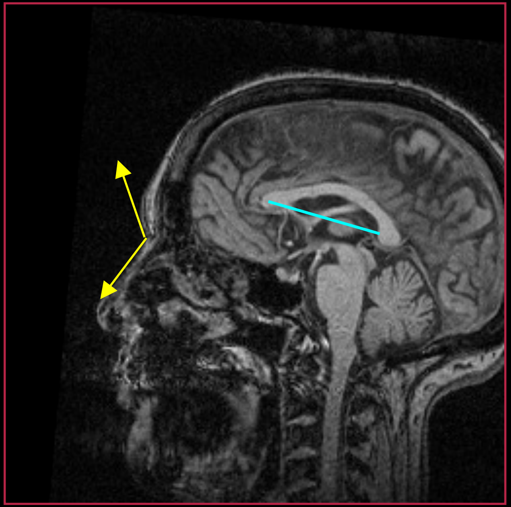
- Translation and rotation
- Rigid body
- Angles and distances are kept

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$a'_i = Ra_i + t$$

$$w = (\Delta x, \Delta y, \theta)$$





## 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

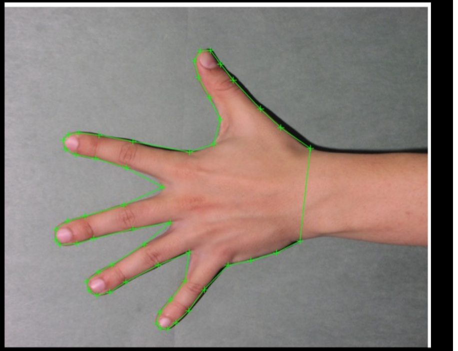
$$F = \sum_{i=1}^{N} \|(Ra_i + t) - b_i\|^2$$

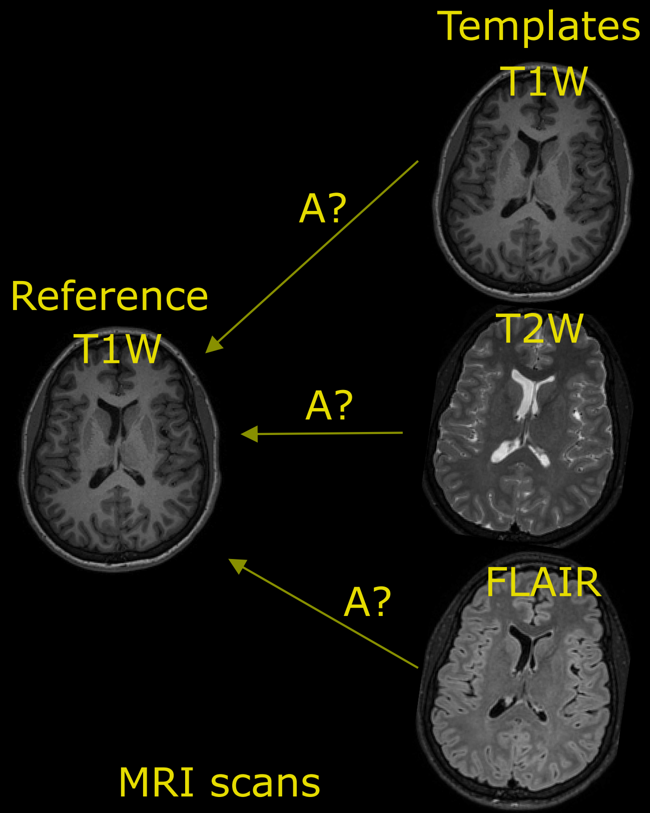
## Optimal rigid body transformation

- The minimum of the objective function can be found in several ways
- The rotation can be found analytically by *singular value decomposition*

## Similarity measures

- Landmarks - time consuming to obtained
- Alternative: joint intensity histograms?





- Same subject
- Same intensity histogram

- Same subject
- Different intensity histogram

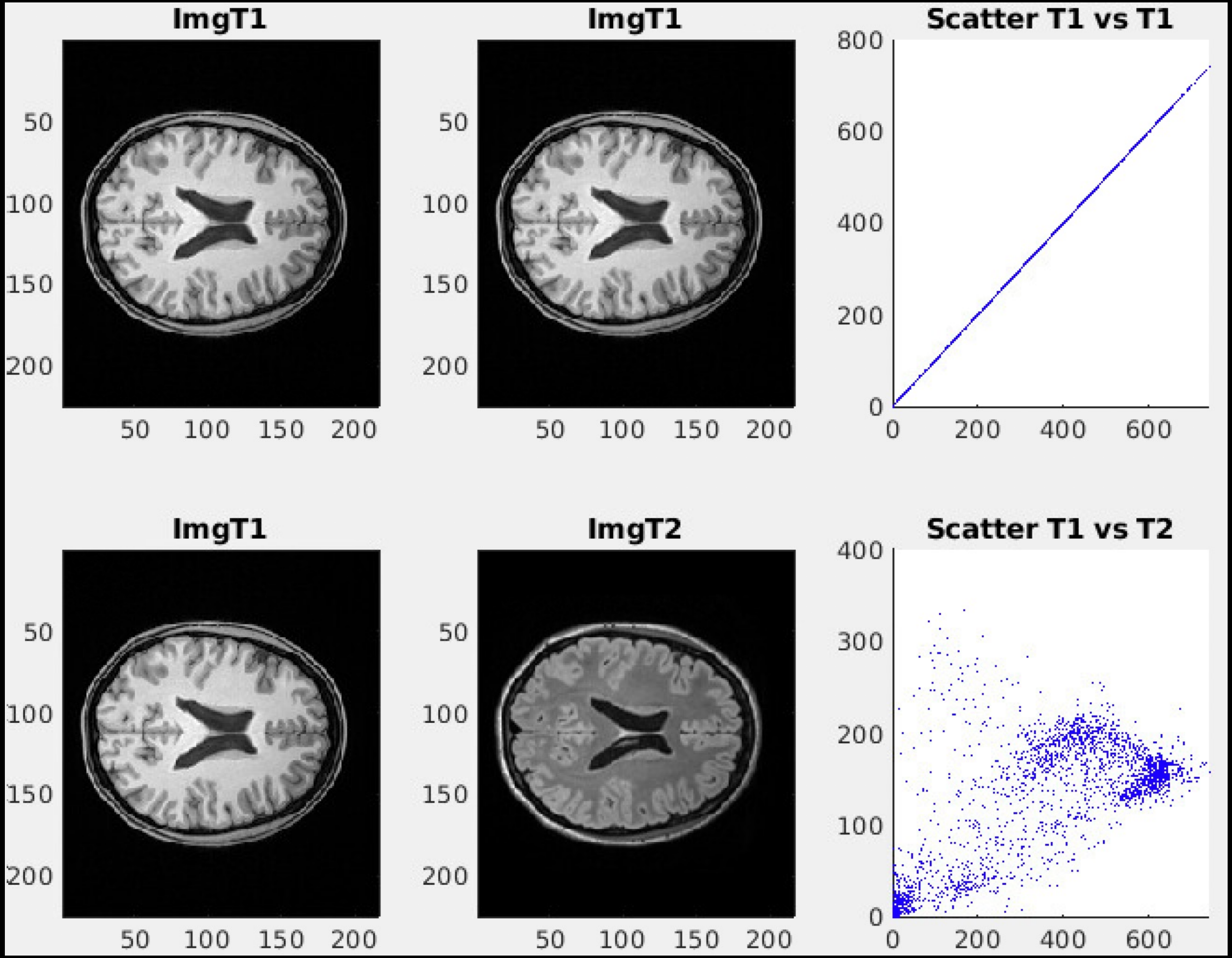
- Same subject
- Different intensity histogram

## Joint intensity histograms

- Perfect registered: Optimal joint intensity agreement

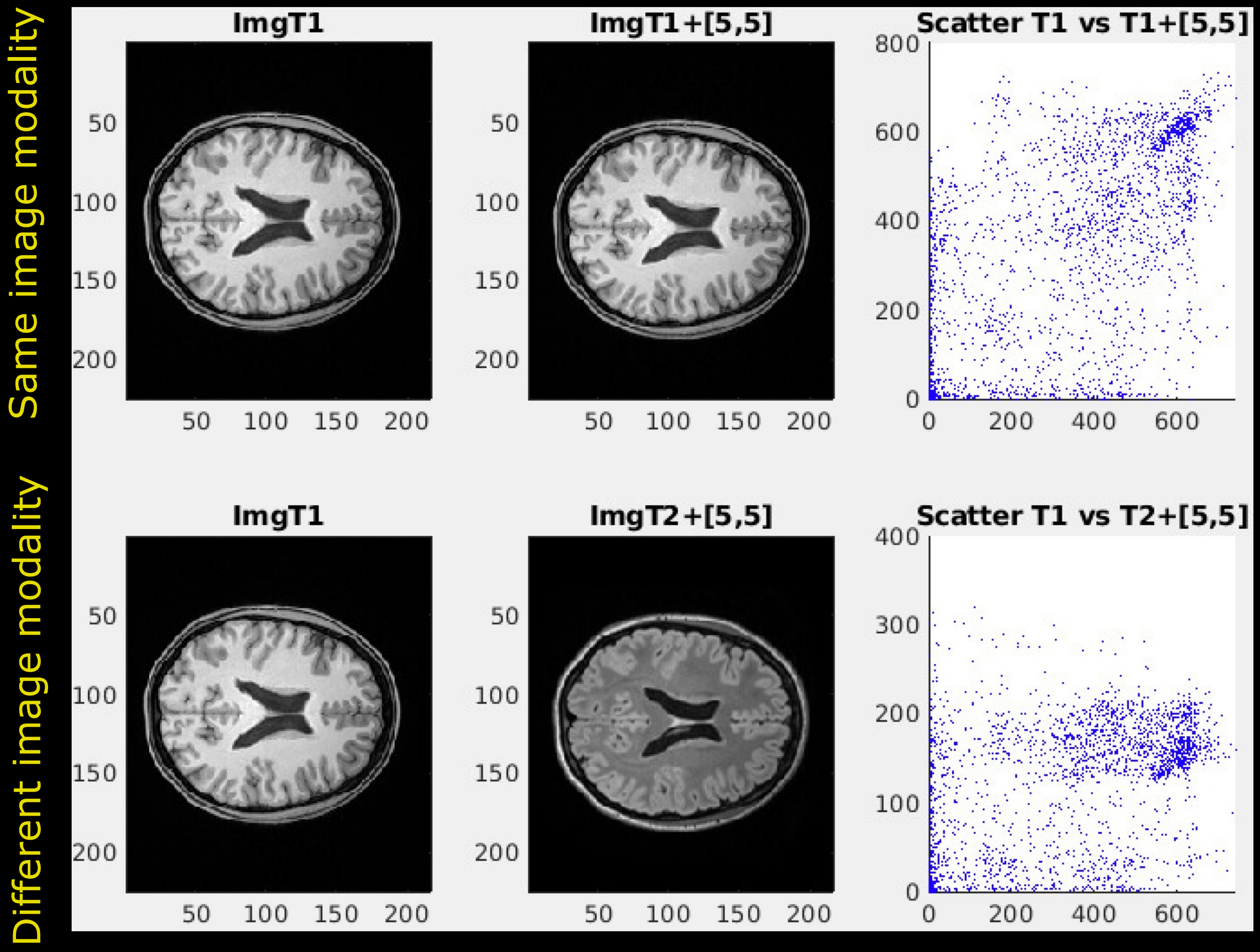
Same image modality

Different image modality



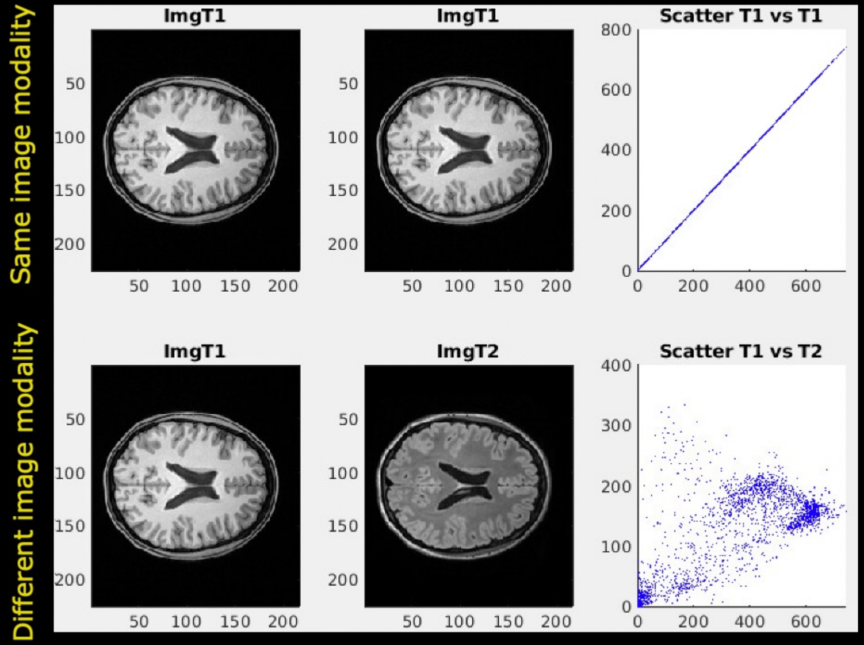
# Joint intensity histograms

- Small translation difference: Lower joint intensity agreement



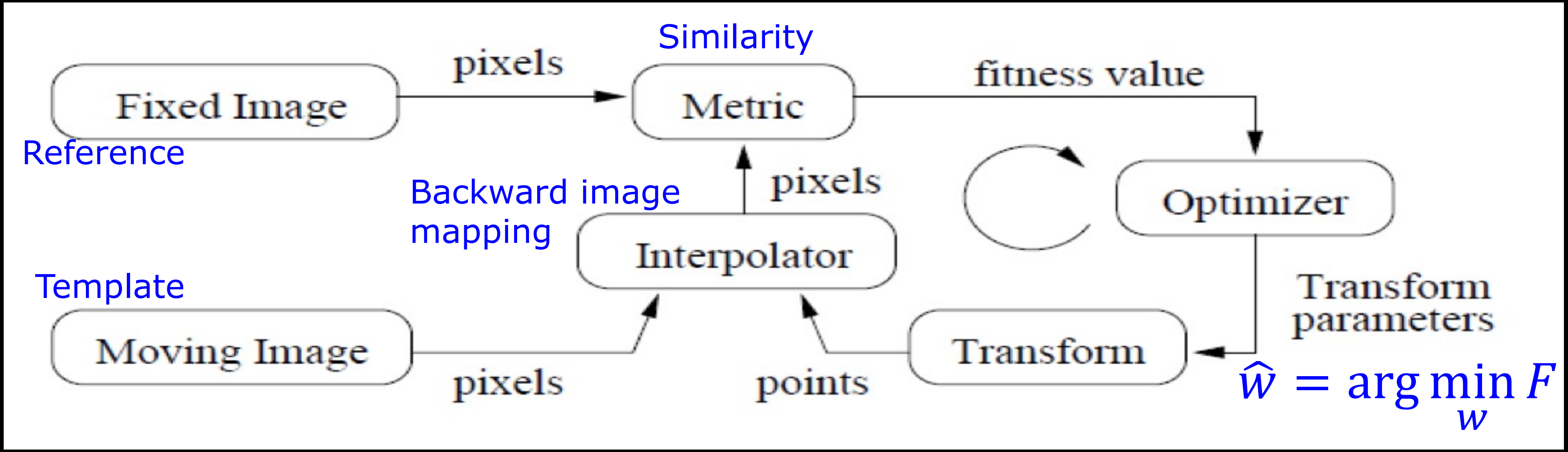
## Joint intensity histograms

- Objective function i.e. a similarity measure to find the optimal transformation
- Many methods exist but two types dominate:
  - Cross-correlation based
    → Fast to estimate, not optimal choice if different image modalities (histograms)
  - Joint entropy based also known as Mutual Information (MI)
    → Slow to estimate, robust when image modalities (i.e., histograms) are different



## 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?

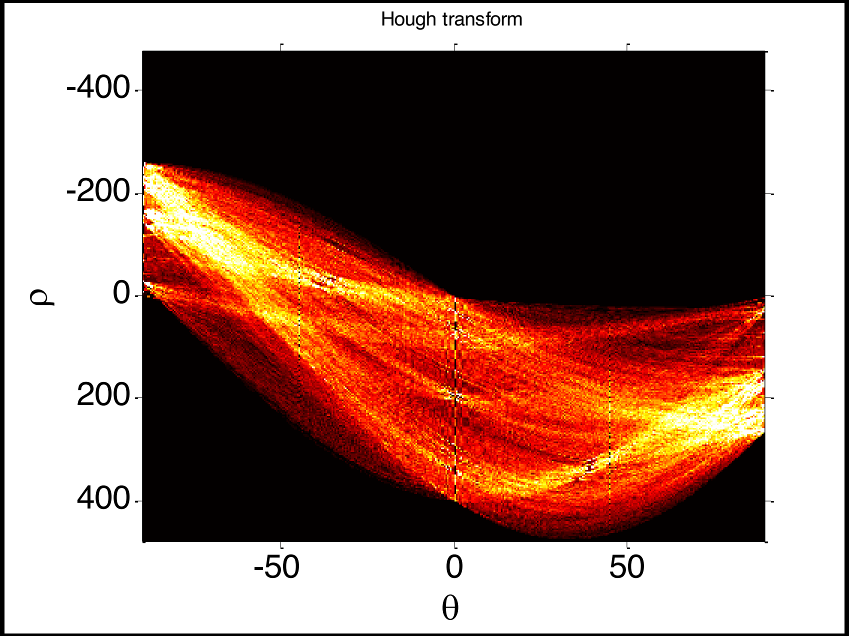
Geometrical transformation

- 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: Boundary Tracing (Hough Transformation) and Dynamic programming


Hough transform
ρ
θ