# Point-based Image Registration

Dr. Ruisheng Su, Assistant Professor

Department of Biomedical Engineering, Medical Image Analysis group

IMAG/e TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

# Overview & course schedule

| Modules | Date | Topic |
|---|---|---|
| **Registration** | April 24 (Thursday) | Course introduction, geometrical transformations |
| | April 28 (Monday) | Point-based image registration |
| | May 1 (Thursday) | Intensity-based image registration |
| **Segmentation** | May 8 (Thursday) | Introduction to image segmentation |
| | May 15 (Thursday) | Segmentation in feature space |
| | May 19 (Monday) | Segmentation using graph-cuts |
| | May 22 (Thursday) | Statistical shape models |
| **Deep learning for MIA** | May 26 (Monday) | Convolutional neural networks |
| | June 2 (Monday) | Deep learning applications (registration) |
| | June 5 (Thursday) | Guest lecture by **Danny Ruijters** (principal scientist @ Philips, full professor @ TU/e) |
| | June 10 (Tuesday) | Deep learning applications (segmentation) |
| | June 12 (Thursday) | Unsupervised deep learning for medical image analysis |

# Outline

- Recap of previous lecture
- Image transformation
- Point-based registration
- Evaluation metrics for point-based registration

# Intended learning outcomes

- explain the role of inverse mapping and interpolation when transforming an image and compute inverse transformation matrices

- design a general algorithm to register two images based on fiducials (i.e., composing an error function and minimizing this function w.r.t. the transformation *T*)
- use optimization to find the minimum of this error function
- recall the exact solution for *T* (matrix notation) when constrained to <u>affine</u> registration
- give at least four reasons why perfect alignment of multiple fiducials is not possible in practice
- describe the algorithm required to find the optimal parameters of *T* when constrained to <u>rigid</u> registration (orthogonal Procrustes problem)
- explain the principle behind the iterative closest point algorithm

- use the target registration error (TRE) to evaluate image registration

# Outline

- Recap of previous lecture
- Image transformation
- Point-based registration
- Evaluation metrics for point-based registration

# Recap

**Quiz**: Which of the following is not a typical cause of misalignment?

A. Distortions caused by the imaging system
B. Patient's voluntary motion during scan
C. Changes in viewing angle between sessions
D. Fluctuations in scanner room lighting

# Recap

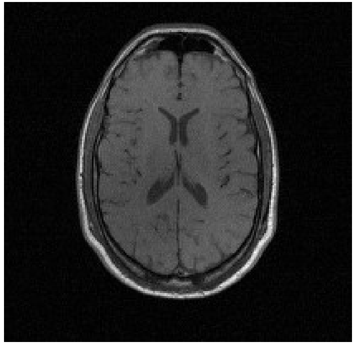
**Quiz**: We register the MRI and PET images of the same patient. Which of the following is correct about the category of registration?

A. Inter-patient

B. Intra-modal

C. Inter-modal

D. Atlas-based



MRI, information about anatomical structures

PET, information about function

# Recap

**Quiz**: An object is transformed using the following transformation matrix: $\begin{bmatrix} 2 & 0 \\ 0 & -0.5 \end{bmatrix}$.
What can we derive about the area and reflection of the transformed object?

A. Magnified

B. Shrunk

C. Rotated

D. Reflected

# Recap

**Quiz**: Which of the following does **not** necessarily hold for a rotation matrix $\boldsymbol{R}$?

*A.* $\boldsymbol{R}\boldsymbol{R}^{-1} = \boldsymbol{I}$

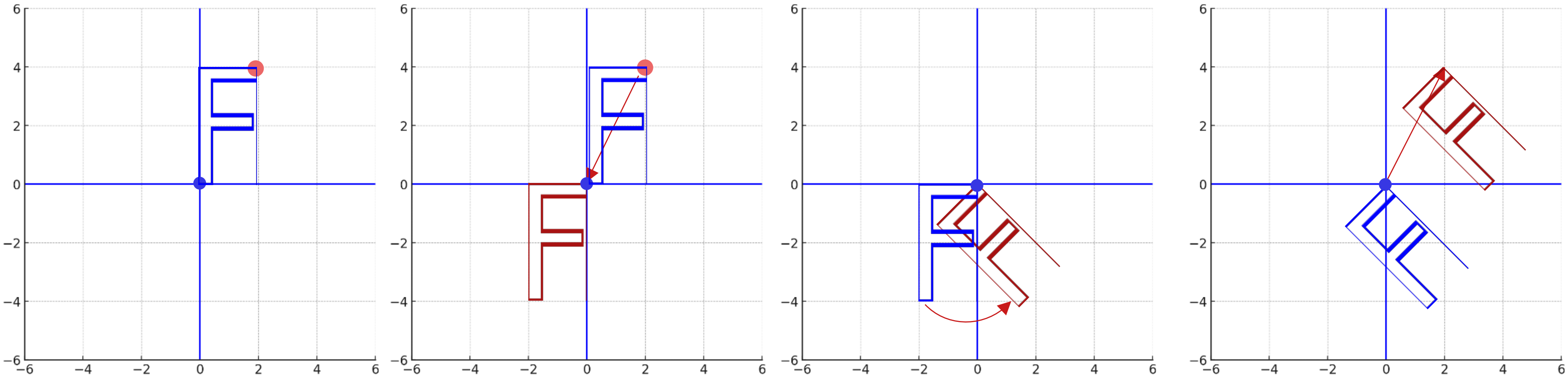
*B.* $\boldsymbol{R}\boldsymbol{R}^{T} = \boldsymbol{I}$

*C.* $\boldsymbol{R} = \boldsymbol{R}^{T}$

*D.* $\boldsymbol{det}(\boldsymbol{R}) = \boldsymbol{1}$

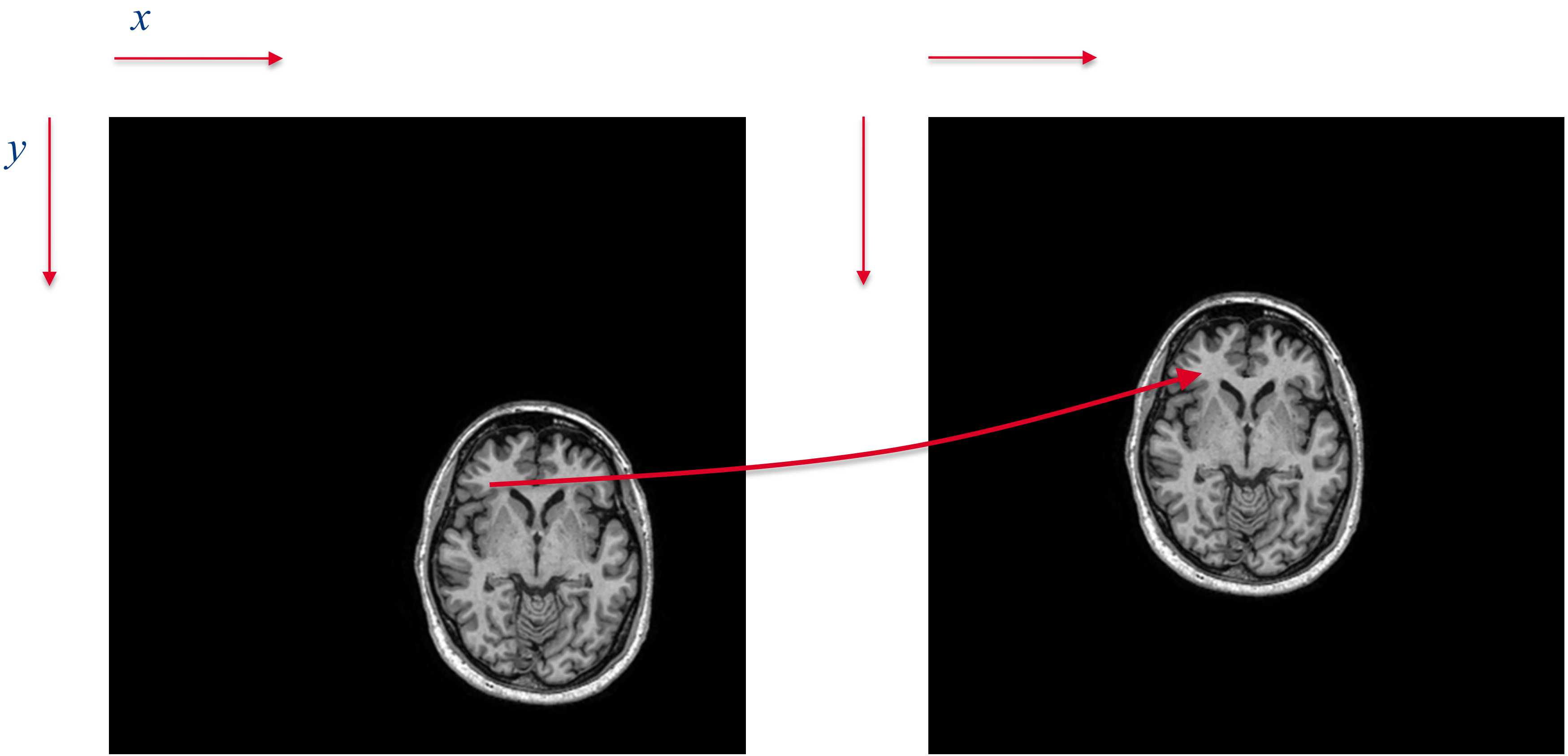
# Recap – Rotation around an arbitrary point

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_r \\ 0 & 1 & y_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_r \\ 0 & 1 & -y_r \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



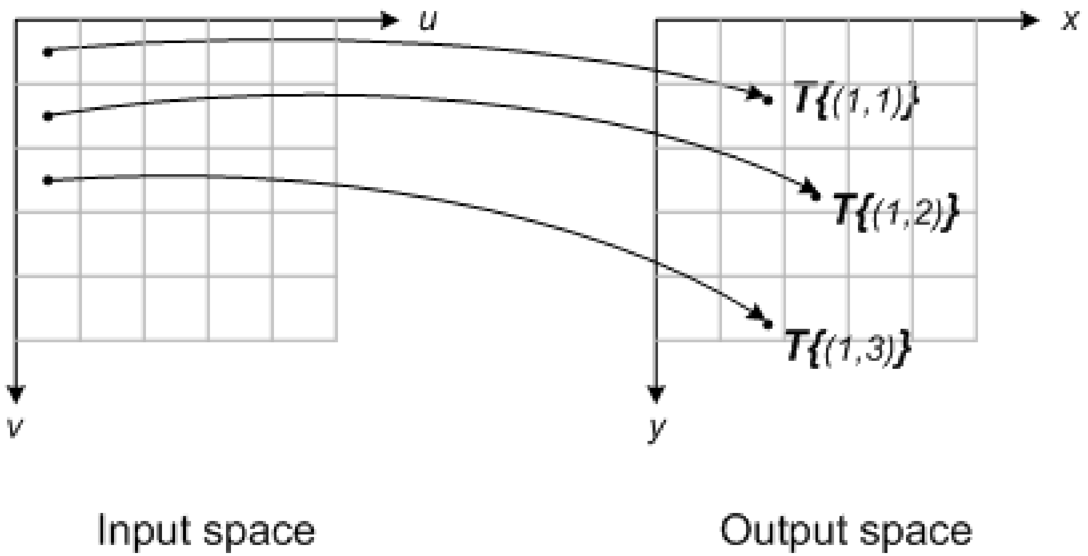
# Outline

- Recap of previous lecture
- Image transformation
- Point-based registration
- Evaluation metrics for point-based registration

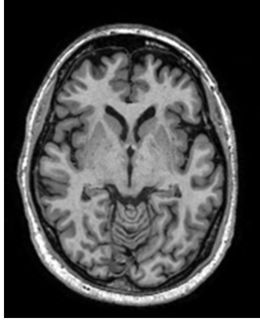

x
y

# Image transformation

Transforming an image means transforming the spatial coordinates of the pixels.



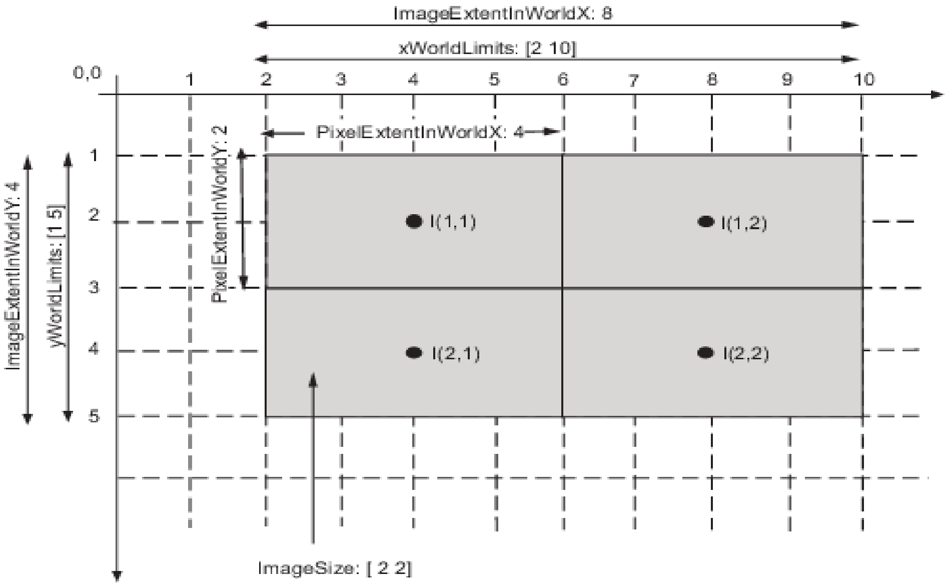
# Pixels



Images are stored as arrays where each element corresponds to a pixel intensity value.

In addition to the **intensity**, in medical imaging each pixel is associated with:

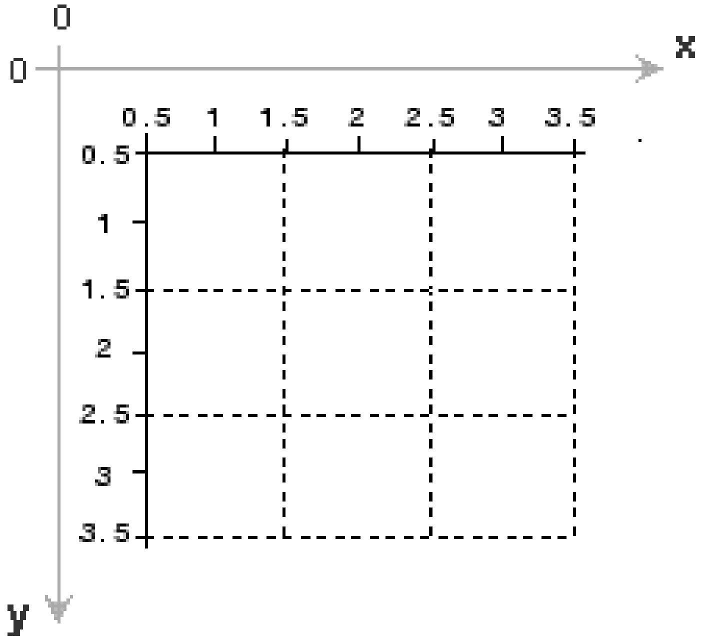
- **Spatial coordinates** – the coordinates in some world coordinate system where the pixel intensity value “appears”.
- **Extent** – the physical size of the pixel.



# Assumptions

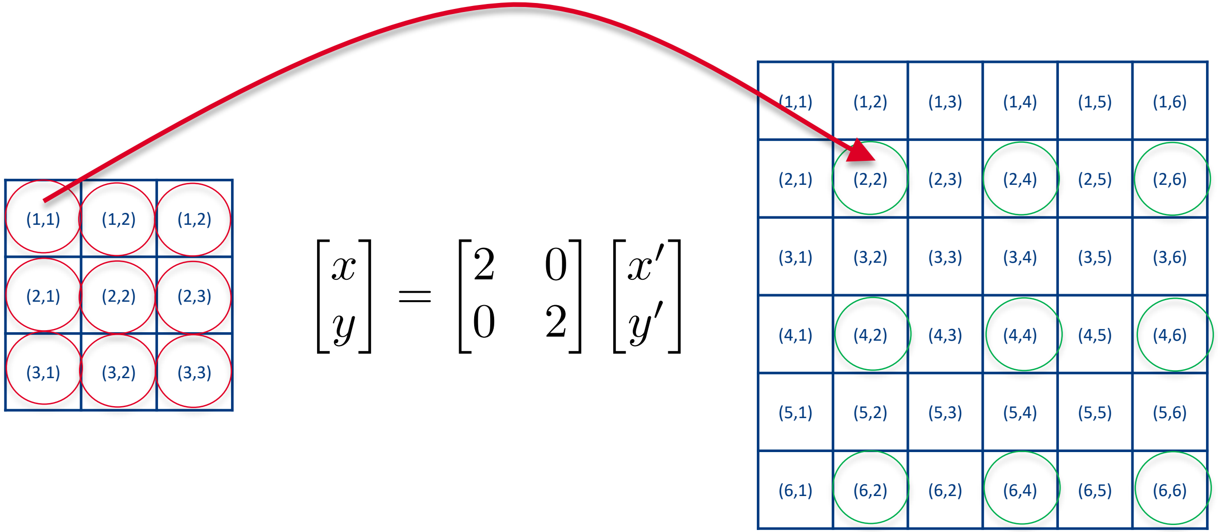
- To simplify the discussion and the implementation in practice, we are going to assume that the pixel indices correspond to the spatial coordinates.
- Furthermore, we are assuming that all images have pixels of the same size and shape (unit size isotropic).

However, note that in practical applications the concepts of physical pixel size and spatial coordinates are very important.



# Forward mapping

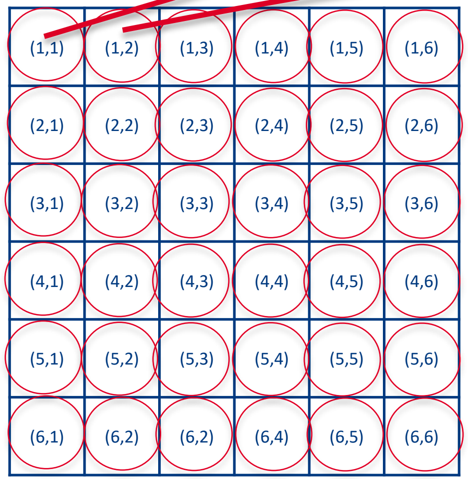
The problem with forward mapping of the coordinates: **gaps**.



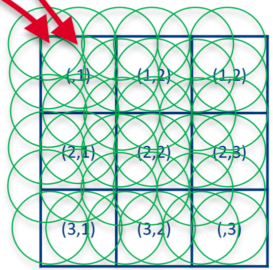

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

# Forward mapping

The problem with forward mapping of the coordinates: **overlaps**.

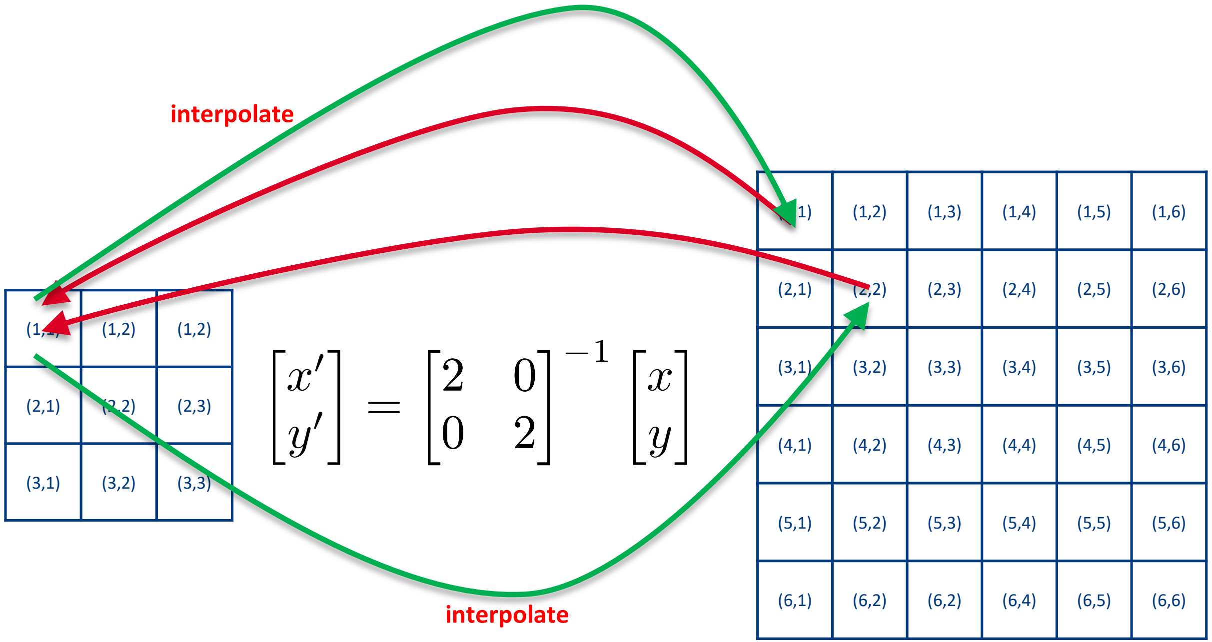



$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$



# Inverse mapping

Gaps and multiple values can be avoided by a performing **inverse mapping and interpolation**.




$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix}$$

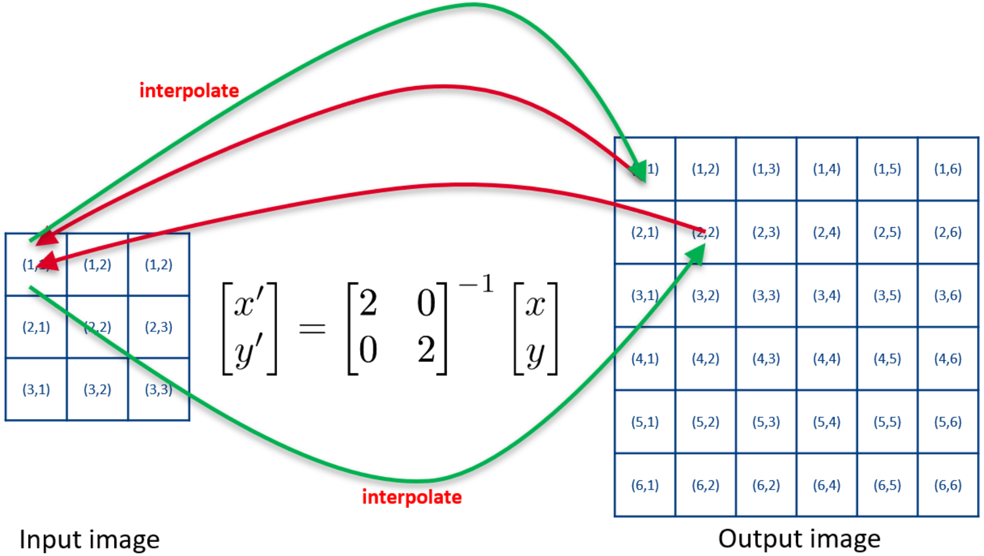
Input image

Output image

# Inverse mapping

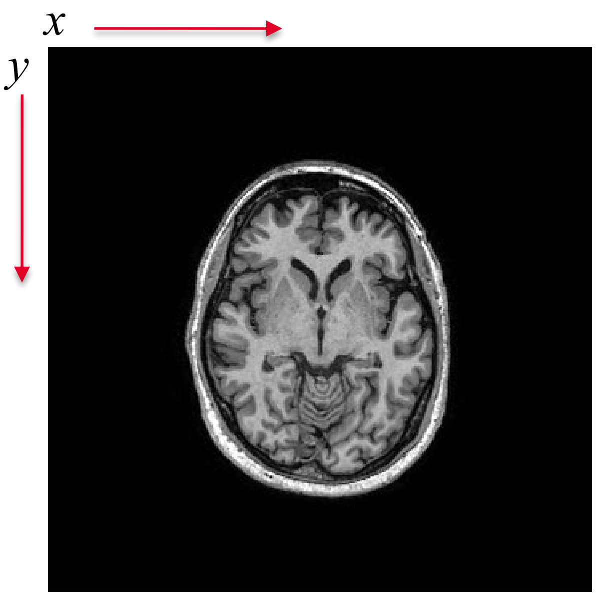
Transforming an image with the transformation **T** by inverse mapping:

1. Define the grid of the output image
2. Map the points on the grid to the input image with $\mathbf{T}^{-1}$
3. Determine the intensity value at those locations with image interpolation

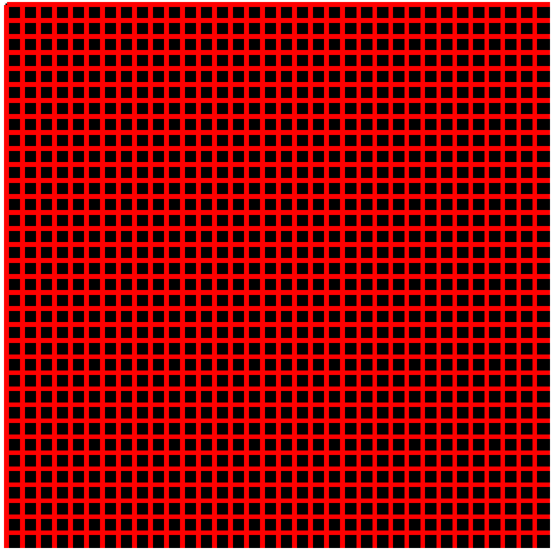



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix}$$

# Example - Inverse mapping

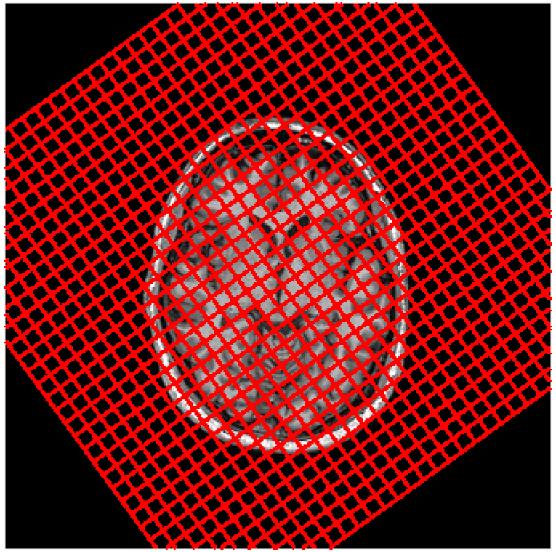



Input image

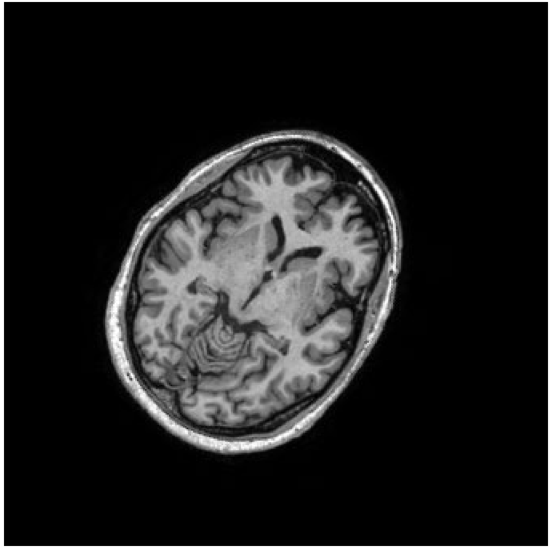


Transformed image (now empty) –
we want a π/5 rotation of the input

# Example - Inverse mapping



Grid of the output transformed with the inverse transformation

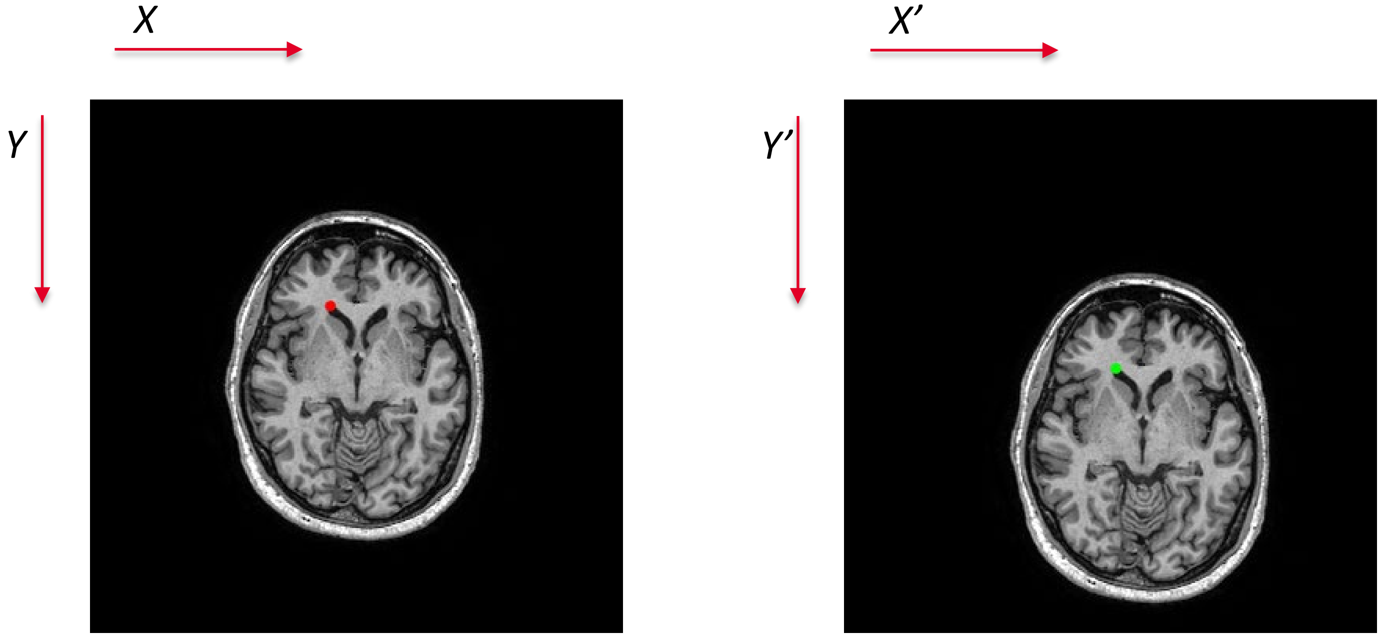


Output image

# Outline

- Recap of previous lecture
- Image transformation
- Point-based registration
- Evaluation metrics for point-based registration

# Example – point-based registration




Fixed

Moving



IMAG/e TU/e

# Example – point-based registration

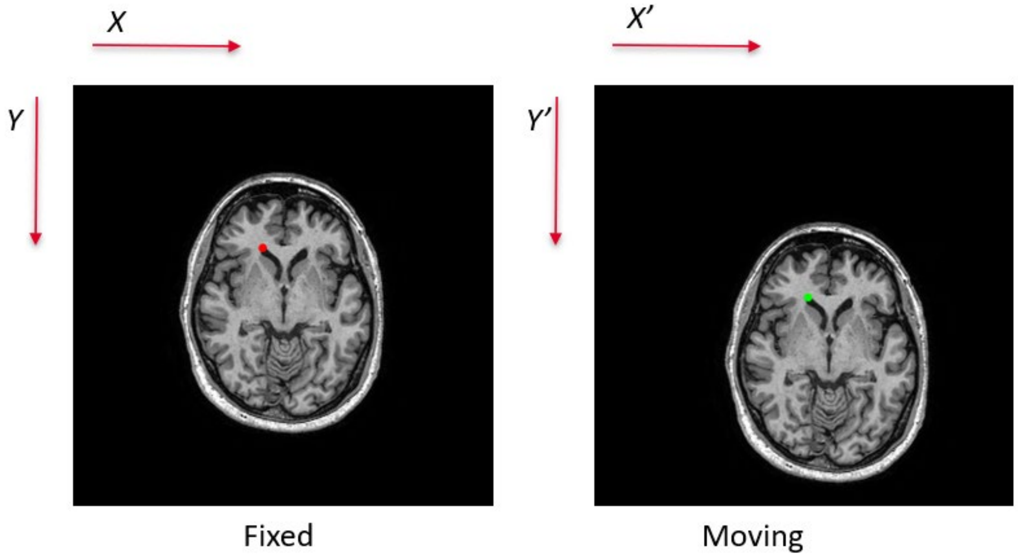
**Assumption:**

Two images are misaligned only with a **translation**:

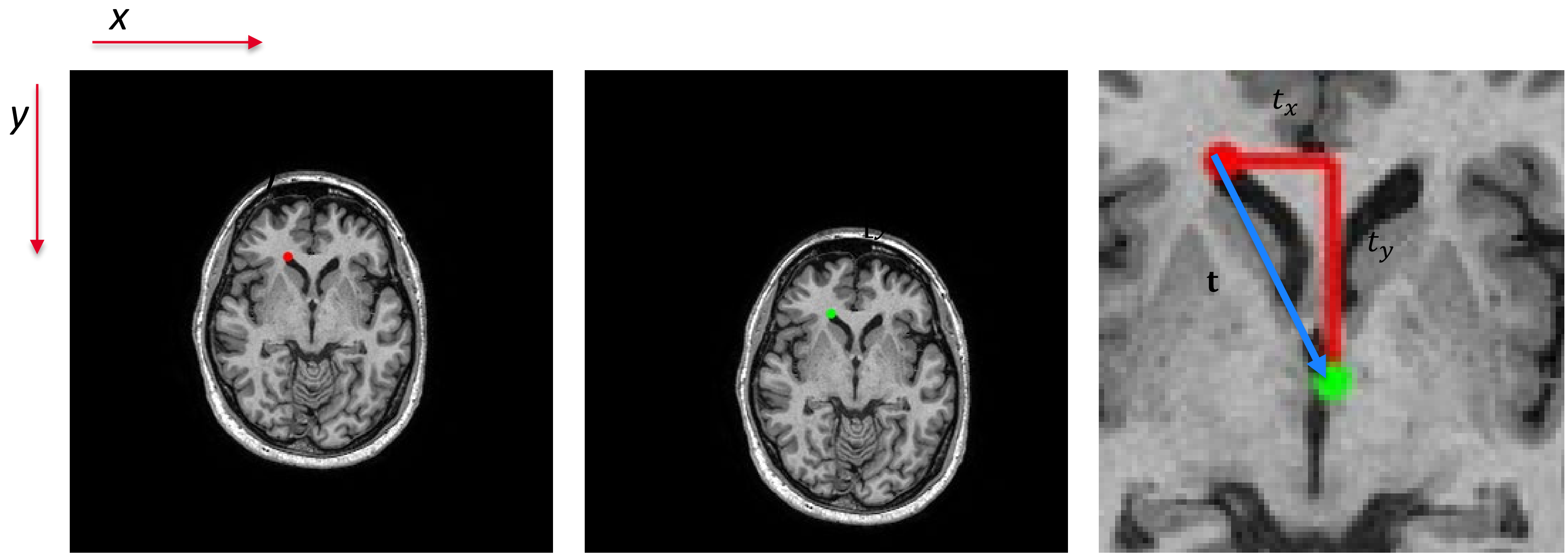
$$\mathbf{x}' = \mathbf{x} + \mathbf{t}$$

**Solution:**

1. Mark the location of some well **discernable feature** in the fixed image.
2. Mark the corresponding location in the moving image.
3. Compute the translation as: $\mathbf{t} = \mathbf{x}' - \mathbf{x}$
4. Transform the moving image by translating it by -**t**



# Example – point-based registration




Fixed

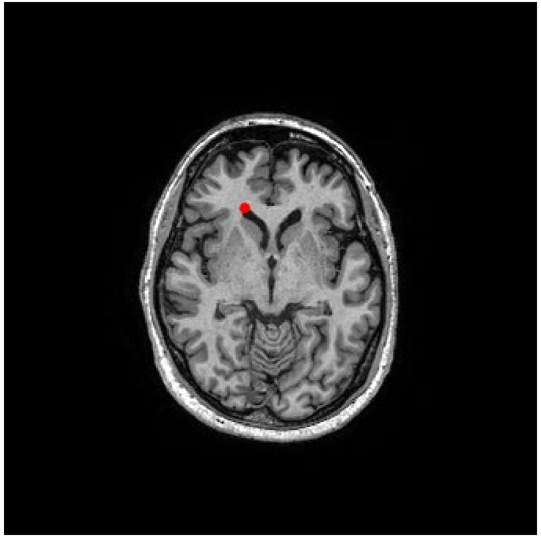
Moving

# Fiducial points

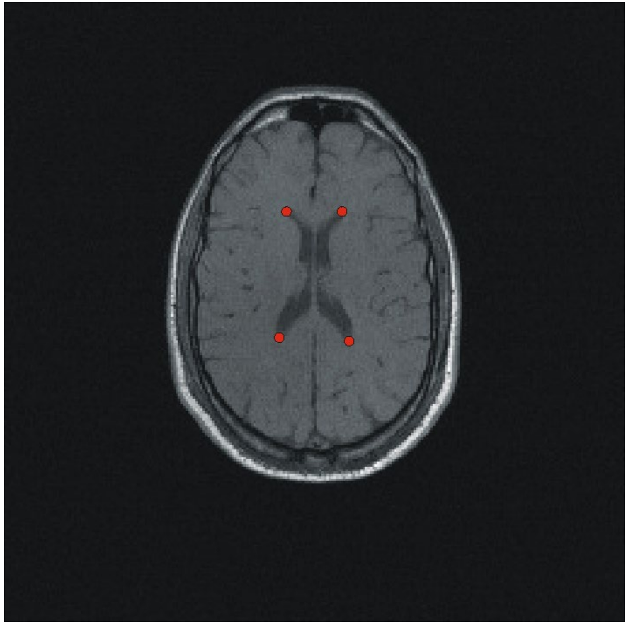
Such points that are taken to be reliable for image registration are called **fiducial points** or **fiducials**.

Fiducials can be placed at either intrinsic or extrinsic features.

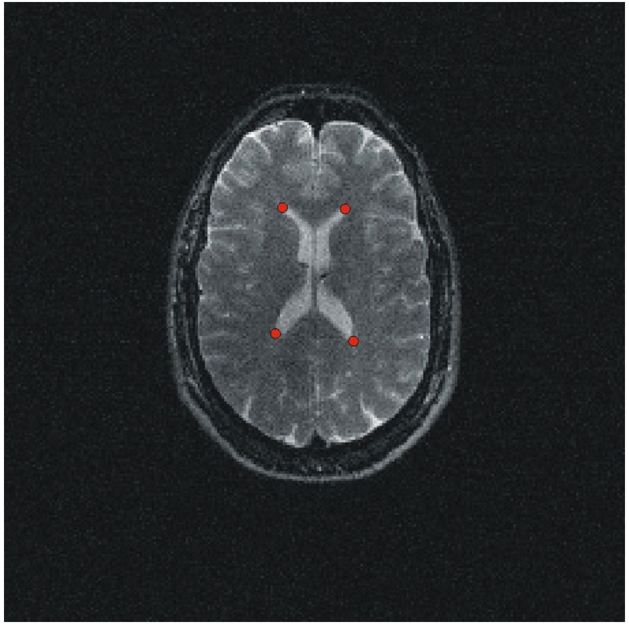
- **Intrinsic features**: e.g. anatomical landmarks.
- **Extrinsic features**: e.g. implanted markers.



# Intrinsic fiducials

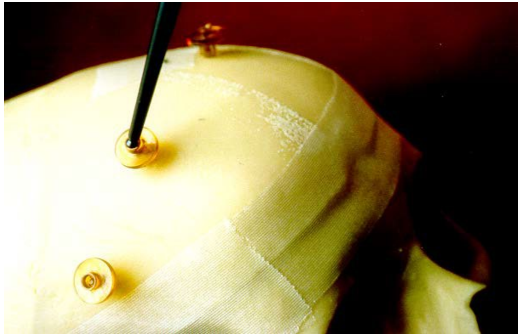


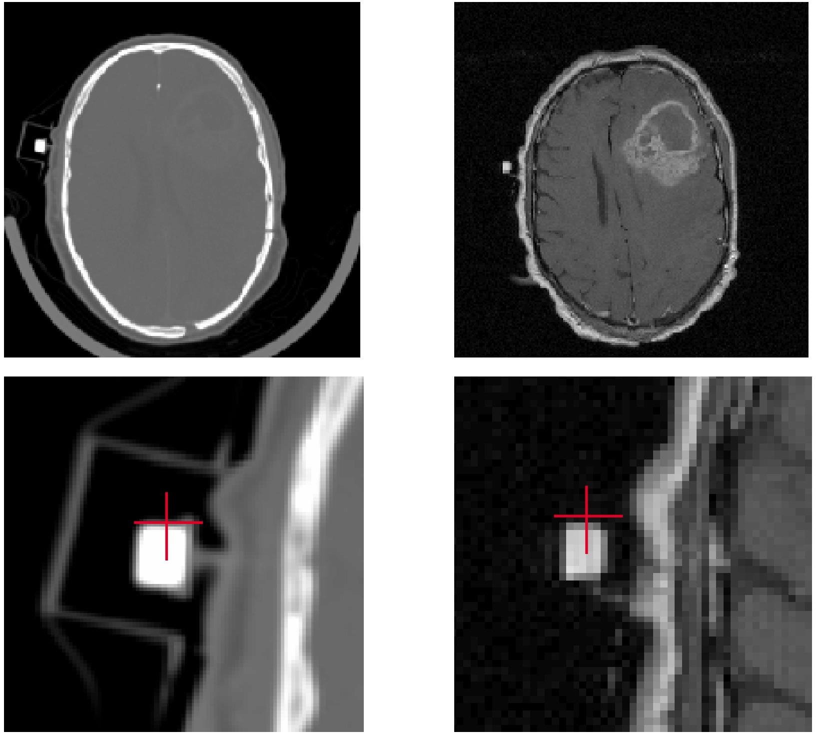
MR T1 weighted



MR T2 weighted

# Extrinsic fiducials





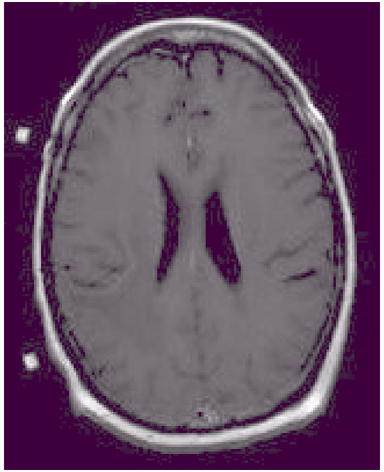
CT MR

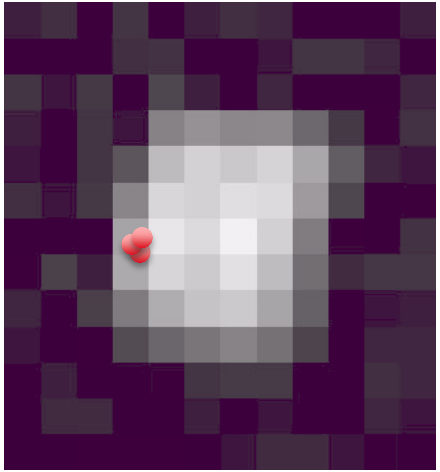
# Fiducial alignment

A perfect alignment of the fiducials is usually not possible.

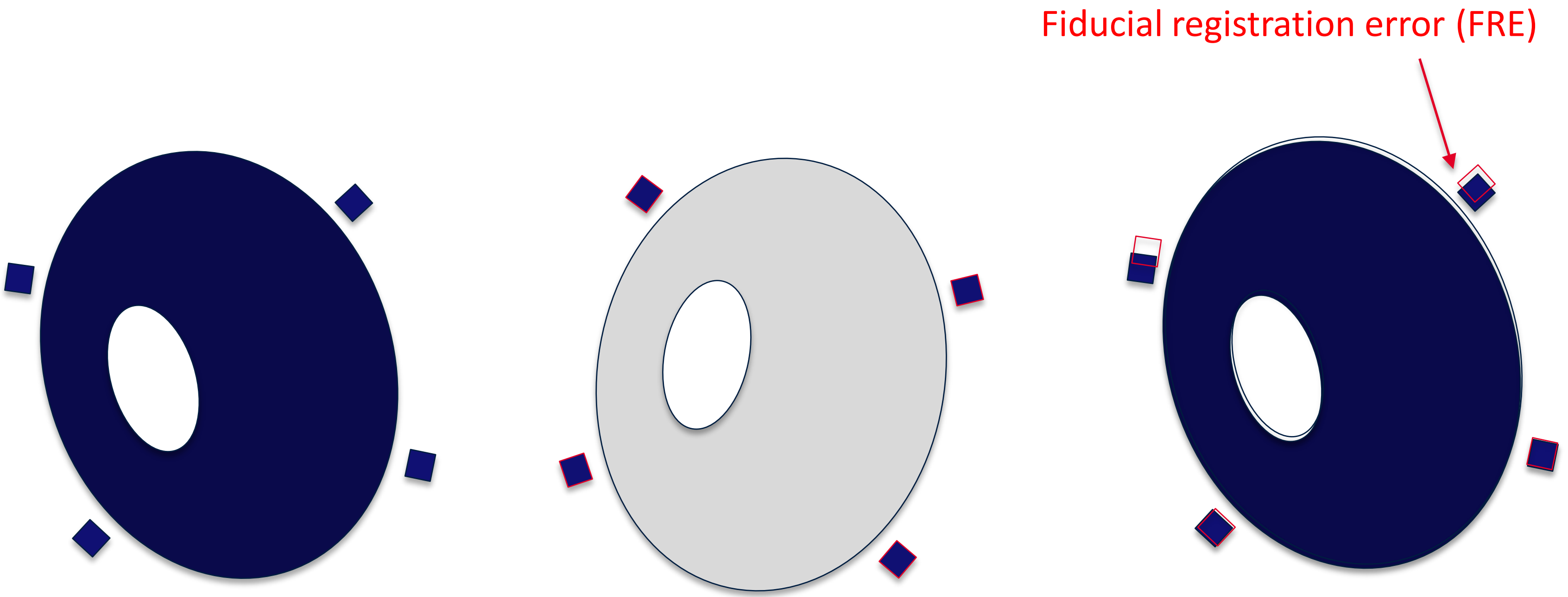
**Reasons:**

- Localization error of the marker (a.k.a. fiducial localization error, FLE)
- Image distortion
- Shift of extrinsic markers
- Incorrect model assumption
- …





# Fiducial registration error (FRE)




Fixed image

Moving image

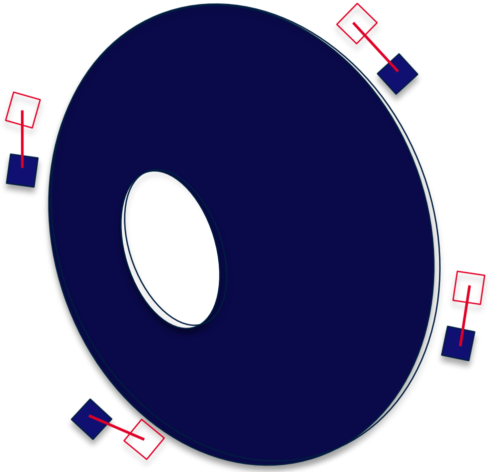
Registered images

# Point-based registration - Goal

It is thus not realistic to look for an algorithm that will find a transformation that results in a **perfect alignment** of all corresponding point pairs.

However, we can design an algorithm that will find a transformation that results in the **best possible alignment** given that fact that there will always be some error.

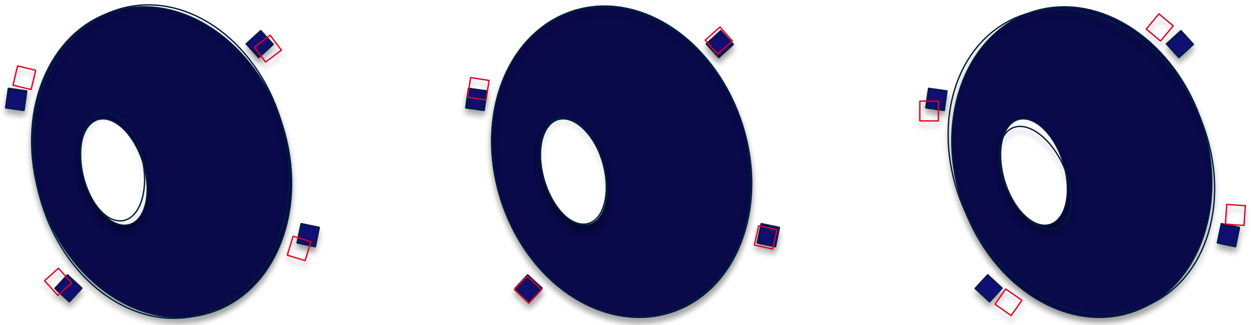
**Best possible alignment = lowest error.**



# Point-based registration – How?

How to find such a transformation?

1. Write the error as a function of the transformation.
2. Find the minimum of the error function w.r.t. the transformation.

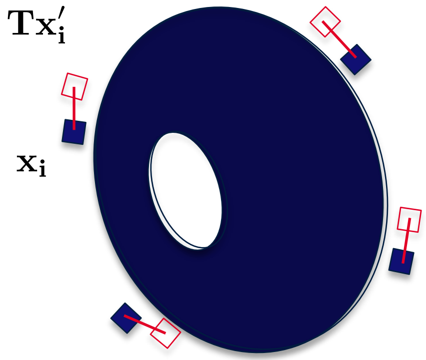


Different transformations will result in different errors.
Our goal: find the transformation that results in the lowest error.

# Point-based registration – How?

**Step 1:** write the error as a function of the transformation.

$$E(\mathbf{T}) = \sum_{i=1}^{n} ||\mathbf{T}\mathbf{x}_\mathbf{i}' - \mathbf{x}_\mathbf{i}||_2^2$$




affine registration $\mathbf{T} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix}$

# Point-based registration – How?

**Step 2:** find the minimum of the error w.r.t. to the parameters.

$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T} \in A} \sum_{i=1}^{n} ||\mathbf{T}\mathbf{x}'_{\mathbf{i}} - \mathbf{x}_{\mathbf{i}}||_2^2 \qquad \mathbf{T} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

The optimal T is the one that minimizes the error

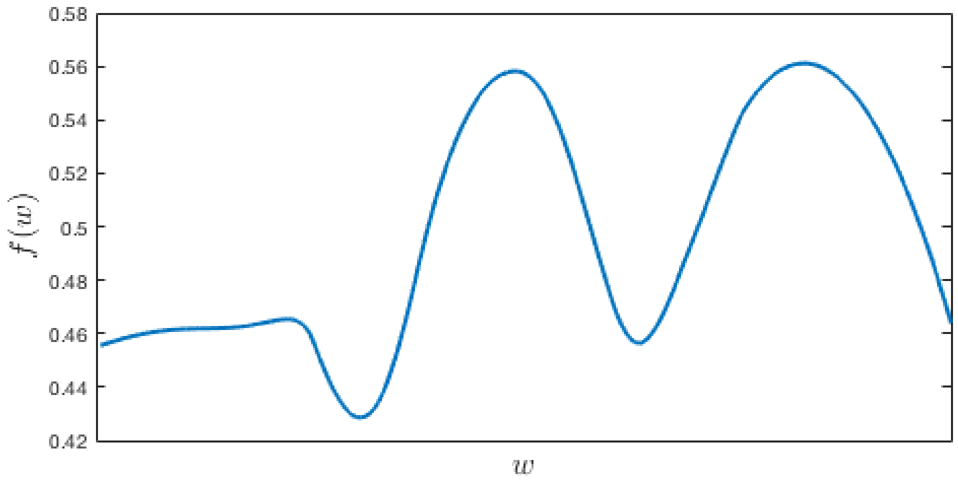
T belongs to the set of all affine transformations



IMAG/e TU/e

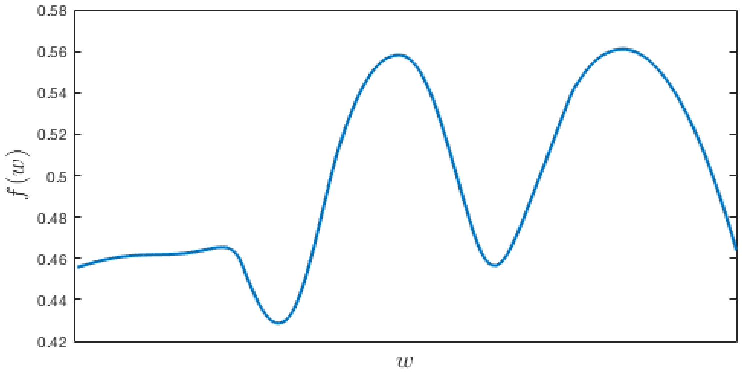
# Optimization

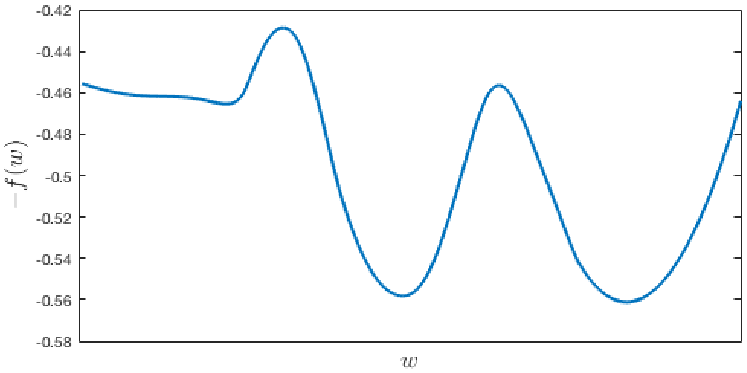
**Optimization** involves finding the "best" parameters according to an "objective function", which is either minimised or maximised.



# Optimization

If we have a method that finds the maximum of a function,
it can be easily used to find a minimum by inverting the function.

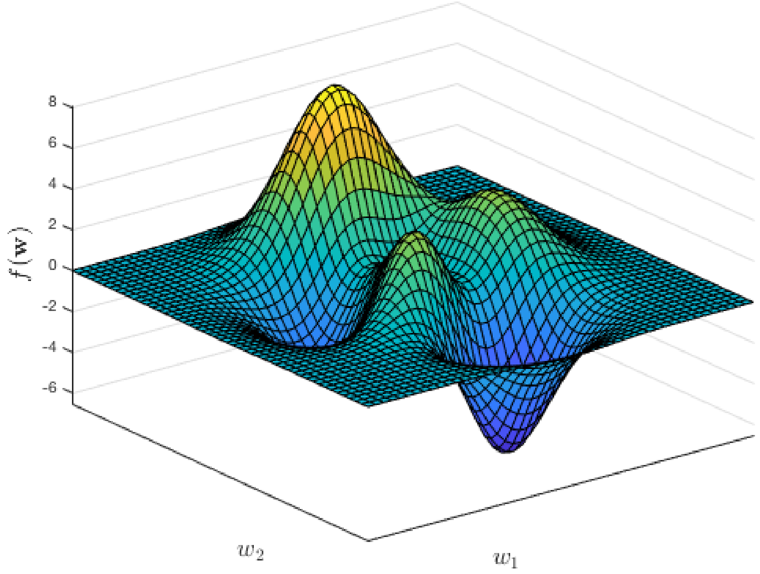


# Optimization

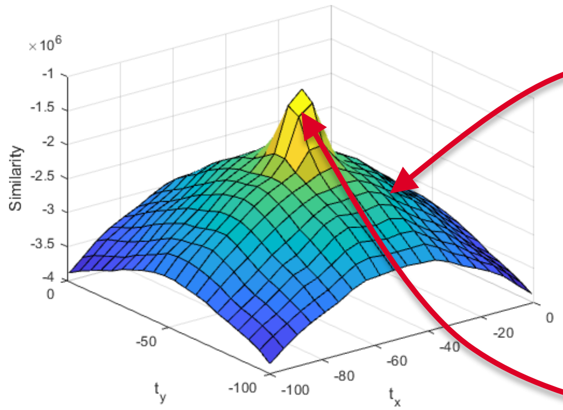
**Optimization** involves finding some "best" parameters according to an "objective function", which is either minimised or maximised.

**2D example:**

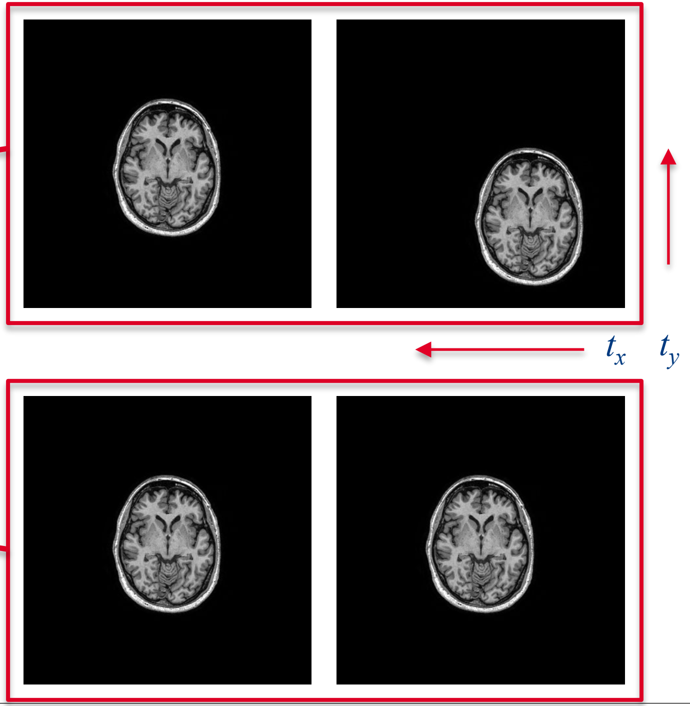


# Optimization

One approach to optimization is **full search** of the parameter space:




Similarity as a function of translation



# Optimization

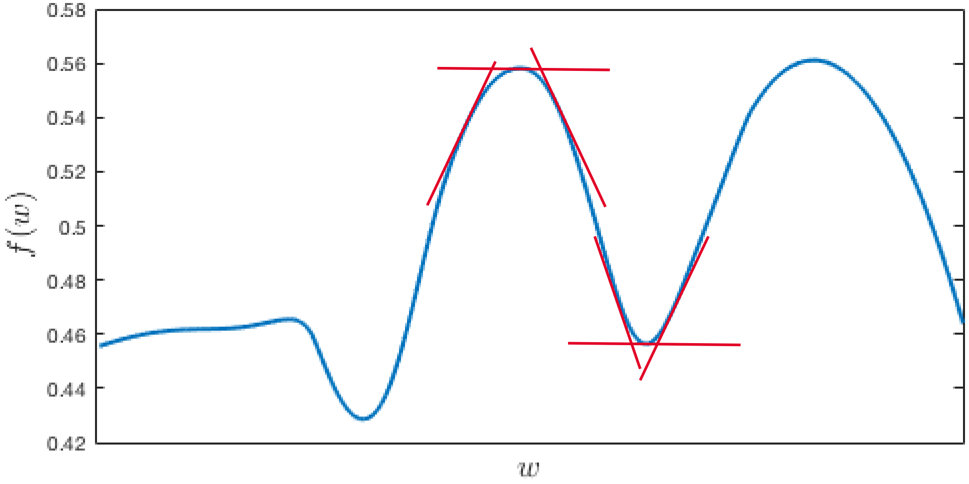
However, this is computationally expensive, even for a small number of parameter.

- Consider affine registration of 2D images that has 6 parameters. If we want to evaluate every parameter at 20 values (e.g. 20 rotation angles), all possible combinations would be $20^6$ = 64 000 000.

- If one evaluation takes just **0.01 second**, it would take just **over a week** to register a pair of images.

# Optimization

How to find the min. and max. of this function **analytically**?

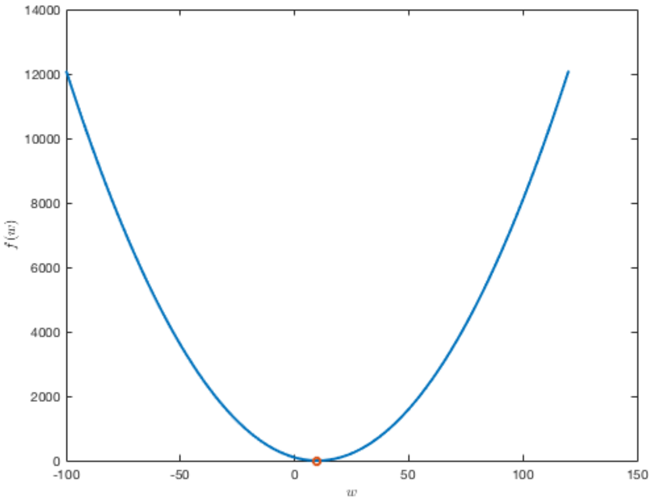


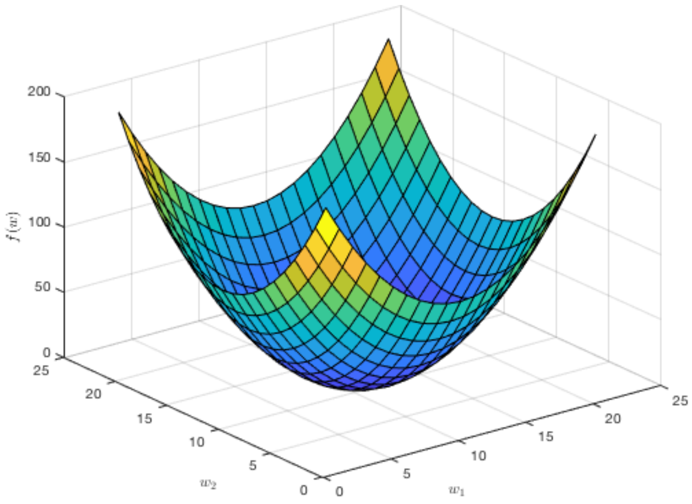

Compute the derivative and set it to zero.

If the function has more than one variable, set the partial derivatives (or gradient vector) to zero.

# Optimization

Quadratic functions:



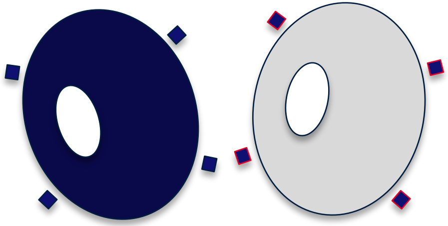
# Optimization (affine transformation)

**Step 2**, find the minimum of the error w.r.t. to the parameters.

$$E(\mathbf{T}) = \sum_{i=1}^{n} ||\mathbf{T}\mathbf{x}_\mathbf{i}' - \mathbf{x}_\mathbf{i}||_2^2$$

$$E(\mathbf{T}) = ||\mathbf{T}\mathbf{X}' - \mathbf{X}||_F^2$$

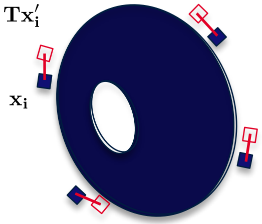
Frobenius norm



$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T} \in A} ||\mathbf{T}\mathbf{X}' - \mathbf{X}||_F^2$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_i & \dots & \mathbf{x}_n \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,i} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,i} & \dots & x_{2,n} \\ 1 & 1 & \dots & 1 & \dots & 1 \end{bmatrix}$$



# Optimization (affine transformation)

**Step 2**, find the minimum of the error w.r.t. to the parameters.

$$E(\mathbf{T}) = ||\mathbf{T}\mathbf{X}' - \mathbf{X}||_F^2$$

**Set of equations:**

$$\nabla_{\mathbf{T}} E(\mathbf{T}) = 0$$

**Solution for affine transformation:**

$$\mathbf{T} = \mathbf{X}'\mathbf{X}^\intercal(\mathbf{X}\mathbf{X}^\intercal)^{-1}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_i & \dots & \mathbf{x}_n \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,i} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,i} & \dots & x_{2,n} \\ 1 & 1 & \dots & 1 & \dots & 1 \end{bmatrix}$$

# Optimization (rigid transformation)

Can we use the same approach to **find the parameters of a rigid registration** (rotation and translation only)?

- **Step 1**, write the error as a function of the transformation:

$$E(\mathbf{T}) = \sum_{i=1}^{n} ||\mathbf{T}\mathbf{x_i'} - \mathbf{x_i}||_2^2$$

- **Step 2**, find the minimum of the error function w.r.t. the transformation:

$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T} \in R} \sum_{i=1}^{n} ||\mathbf{T}\mathbf{x_i'} - \mathbf{x_i}||_2^2$$

# Optimization (rigid transformation)

Compared with affine registration, the solution is now **<u>constrained</u>** to transformation matrices that only do **rotation** and **translation**:

$$\mathbf{T} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{T} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & t_x \\ \sin(\phi) & \cos(\phi) & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

**Affine** **Rigid**

Using the same approach as before will **NOT** guarantee rigid transformation.

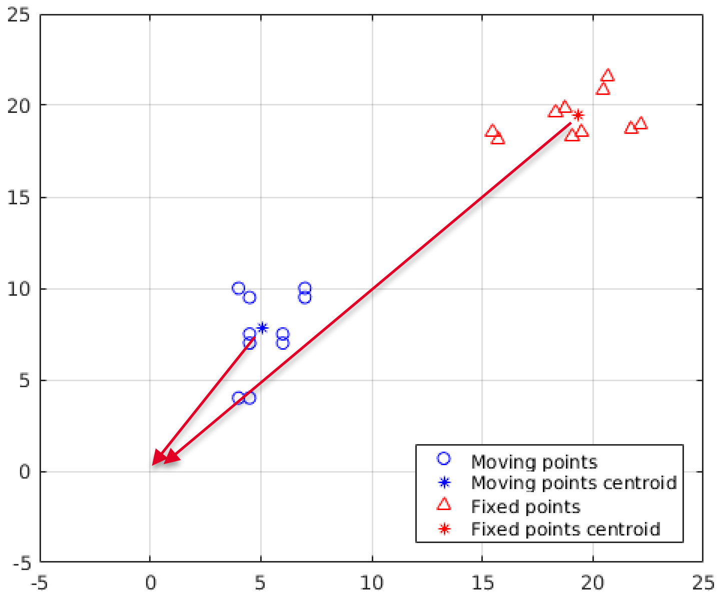
# Optimization (rigid transformation)

The minimization of the error of rigid registration is called “**orthogonal Procrustes problem**”.

**Algorithm**:

1. Compute the centroids of the fiducial points in the fixed and moving image.

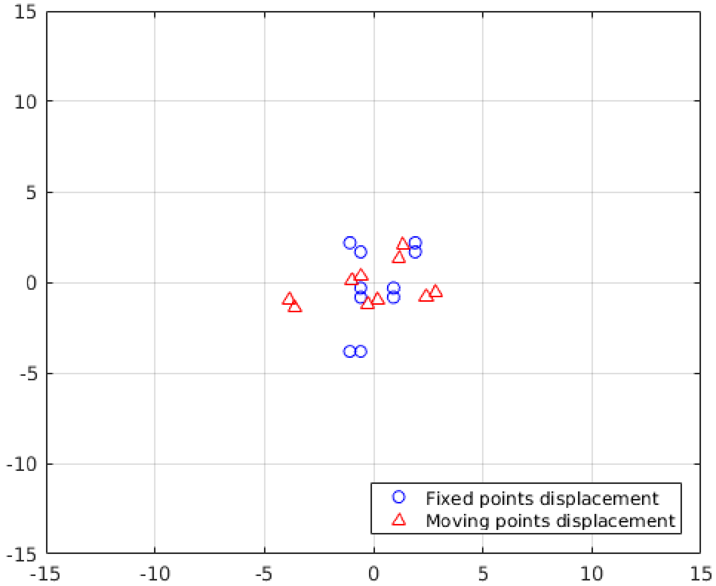
$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i;\ \bar{\mathbf{x}}' = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}'_i$$



# Optimization (rigid transformation)

2. Compute the displacements:

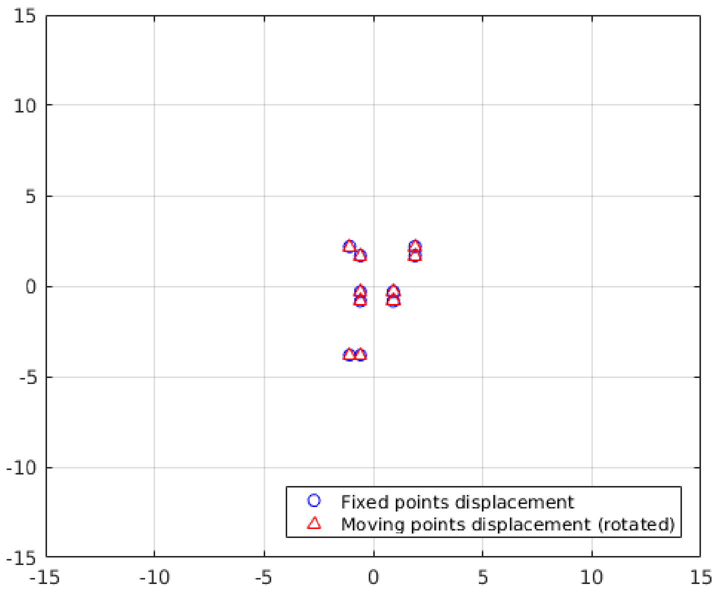
$$\hat{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$$
$$\hat{\mathbf{x}}'_i = \mathbf{x}'_i - \bar{\mathbf{x}}'$$



# Optimization (rigid transformation)

3. Compute the rotation matrix:

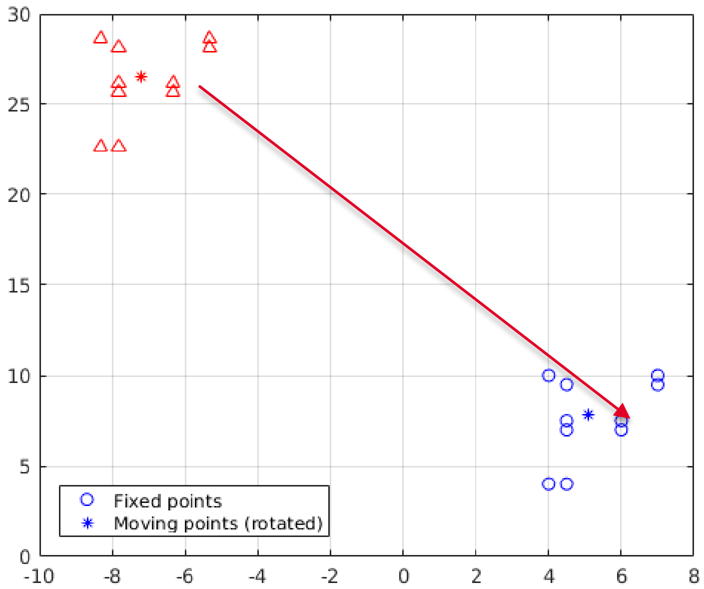
$$\mathbf{R} = \mathbf{V}\mathrm{diag}(1, 1, \mathbf{VU})\mathbf{U}^\intercal$$



# Optimization (rigid transformation)

4. Compute the translation:

$$\mathbf{t} = \bar{\mathbf{x}} - \mathbf{R}\bar{\mathbf{x}}'$$



# Optimization (rigid transformation)

1. Compute the centroids of the fiducial points in the fixed and moving image.

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i;\ \bar{\mathbf{x}}' = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}'_i$$

2. Compute the displacement of each fiducial point

$$\hat{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}};\ \hat{\mathbf{x}}'_i = \mathbf{x}'{}_i - \bar{\mathbf{x}}'$$

3. Compute the covariance matrix of the fiducials

$$\mathbf{H} = \sum_{i=1}^{n}\hat{\mathbf{x}}'_i\hat{\mathbf{x}}_i^{\intercal}$$

4. Perform singular value decomposition of the covariance matrix.

$$\mathbf{H} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{\intercal}$$

5. Compute the rotation matrix:

$$\mathbf{R} = \mathbf{V}\mathrm{diag}(1, 1, \mathbf{VU})\mathbf{U}^{\intercal}$$

→ The diagonal term ensures that we get a proper rotation matrix with determinant equal 1.

6. Compute the translation vector:

$$\mathbf{t} = \bar{\mathbf{x}} - \mathbf{R}\bar{\mathbf{x}}'$$

# Optimization (unknown correspondence)

Thus far we assumed that the correspondence between the points in the moving and fixed images is know. What if we do not know the correspondence?

We introduce a **correspondence function** that is now **subject to optimization**:

$$E(\mathbf{T}) = \sum_{i=1}^{n} ||\mathbf{T}C(\mathbf{x}_i, \{\mathbf{x}_j^{'}\}) - \mathbf{x}_i||^2$$

# Iterative optimization

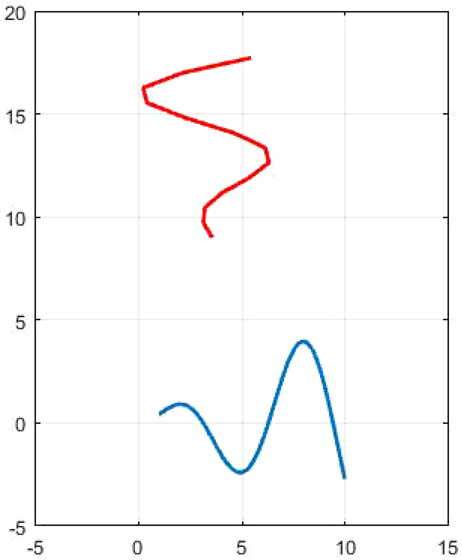
This leads to optimization problems that are usually solved with an **iterative procedure**.

Example: **iterative closest point**.

Iterate until convergence:

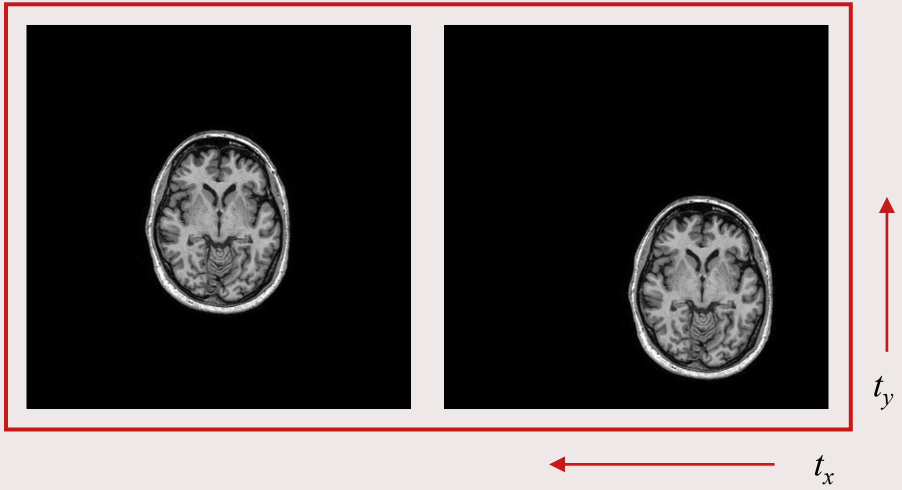
- For each point in the moving shape find the closest point in the fixed shape.
- Assume the found correspondence is correct and register (using one of the algorithms that work with correspondence).

# Iterative closest point

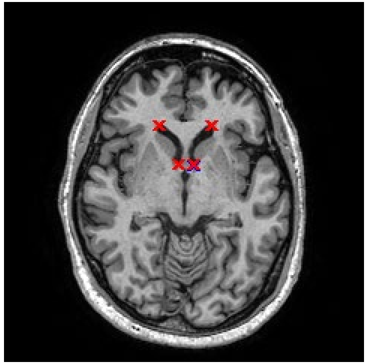


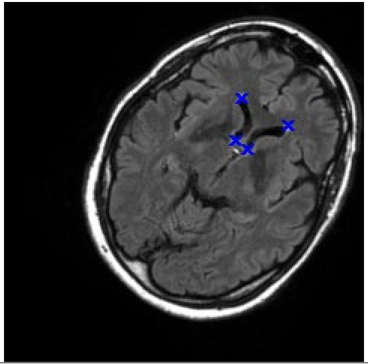
# Outline

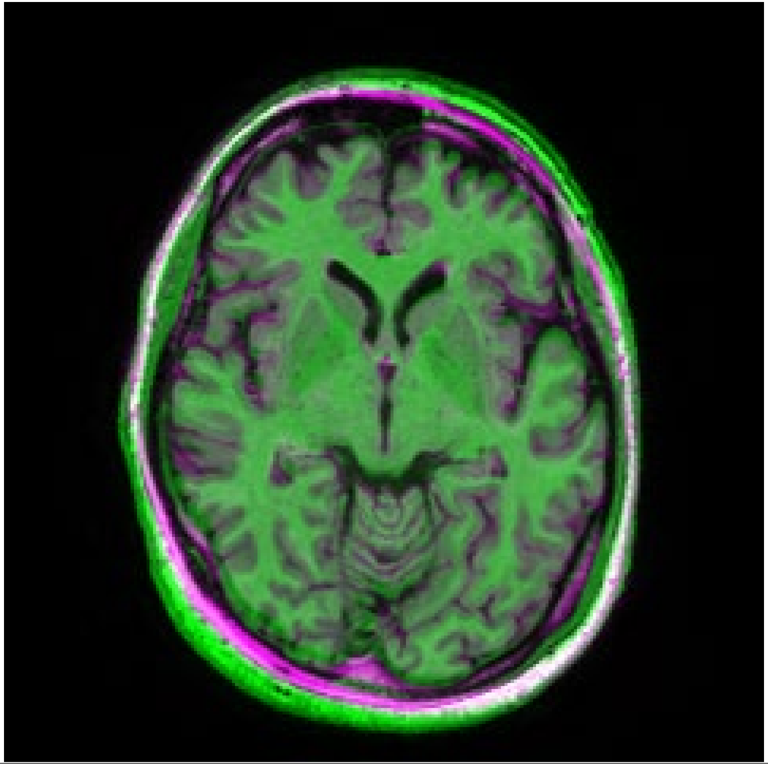
- Recap of previous lecture
- Image transformation
- Point-based image registration
- Evaluation of registration accuracy



# Evaluation of registration accuracy







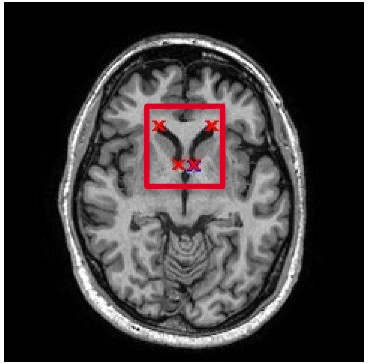
# Evaluation of registration accuracy

Image registration can be evaluated by computing the registration error for some target corresponding point pairs.

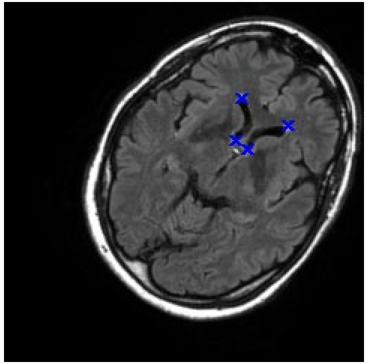
→ **Target registration error (TRE)**

- The target points should be selected in locations that are relevant for some treatment or diagnosis.
- Basically, this is the same as the procedure for selecting corresponding point pairs to compute the transformation.
- <u>However, the target corresponding points must be different from the points used to compute the transformation!</u> Why?

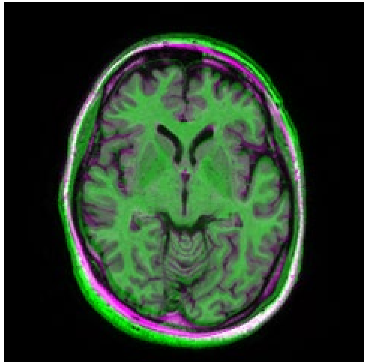
(Poor) Example of affine registration:



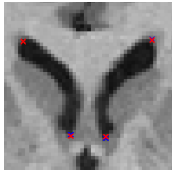
Fixed



Moving



Registered



**Note that the registration error for the fiducials is very low. This is because we minimize this error! The error outside of these locations is larger.**

# Evaluation procedure

1. Perform image registration (compute the transformation matrix $T$)

2. Annotate some target corresponding point pairs in the fixed and moving images. These must be different from the corresponding points used to compute $T$ and at locations that are relevant for some treatment or diagnosis.

3. Transform the points from the moving image

4. Compute the target registration error as the average distance between the points in the fixed image and the transformed moving points.

# Summary

The student can:

- explain the role of inverse mapping and interpolation when transforming an image and compute inverse transformation matrices

- design a general algorithm to register two images based on fiducials (i.e., composing an error function and minimizing this function w.r.t. the transformation *T*)
- use optimization to find the minimum of this error function
- recall the exact solution for *T* (matrix notation) when constrained to <u>affine</u> registration
- give at least four reasons why perfect alignment of multiple fiducials is not possible in practice
- describe the algorithm required to find the optimal parameters of *T* when constrained to <u>rigid</u> registration (orthogonal Procrustes problem)
- explain the principle behind the iterative closest point algorithm

- use the target registration error (TRE) to evaluate image registration

# Thank you

r.su@tue.nl

**Next: intensity-based registration**