Technische Universiteit Eindhoven University of Technology

Point-based image registration

Ruisheng Su,

Maureen van Eijnatten

Today:

- Image transformation
- Point-based image registration
- Optimization
- Evaluation of image registration accuracy

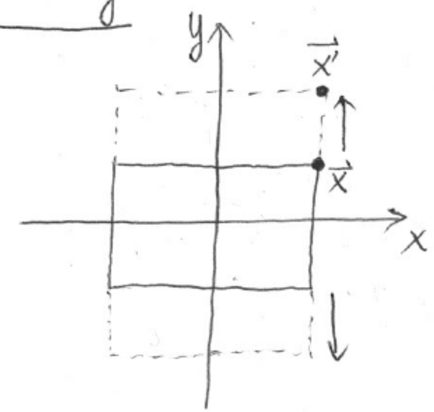
Medical image registration in 8DC00 in a nutshell

- **Week 1, day 1:** course introduction, introduction to medical image registration, review of linear algebra, geometrical transformations
- **Week 1, day 2:** point-based image registration,
- **Week 2, day 1:** intensity-based registration
- **Week 2, day 2:** evaluation metrics, active shape model
- **Week 7:** deep learning for image registration

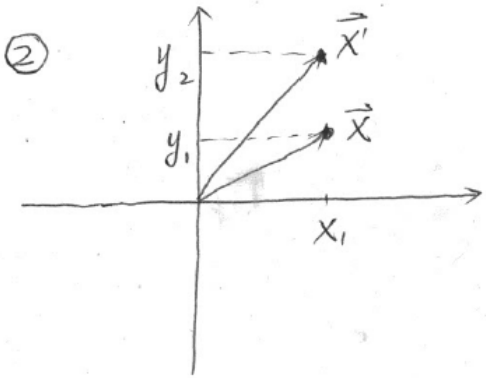
Scaling

scale ↑ 2 times in y direction

①



②



③. $\vec{x} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$ $\quad \vec{x}' = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$ $\quad \left.\begin{aligned} x_2 &= x_1 \\ y_2 &= 2y_1 \end{aligned}\right\} \Rightarrow \begin{aligned} x_2 &= 1 \cdot x_1 + 0 \cdot y_1 \\ y_2 &= 0 \cdot x_1 + 2y_1 \end{aligned}$

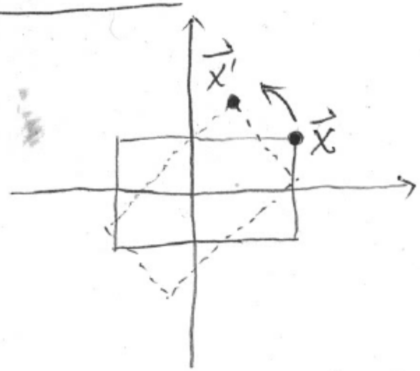
④. $\vec{x}' = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \cdot x_1 + 0 \cdot y_1 \\ 0 \cdot x_1 + 2 \cdot y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \vec{x}$

⑤. $f(\vec{x}) = \vec{x}'$ what is $f(\cdot)$? $f(\cdot)$

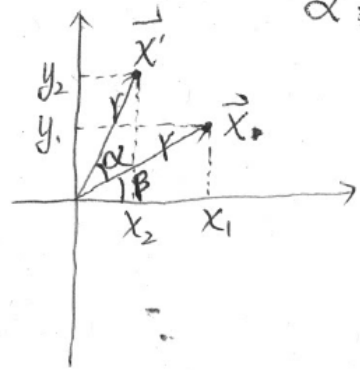
Rotation

α : angle of rotation

①



②



③ $\vec{x} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$ $\quad \vec{x}' = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$ $\quad x_1 = r\cos\beta \quad x_2 = r\cos(\alpha+\beta)$

$y_1 = r\sin\beta \quad y_2 = r\sin(\alpha+\beta)$

④ $\vec{x}' = f(\vec{x}) \quad f(\cdot) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \vec{x}' = \underline{\begin{bmatrix} a & b \\ c & d \end{bmatrix}} \vec{x}$

⑤. $x_2 = r\cos(\alpha+\beta) = \underline{r}\cos\alpha\,\underline{\cos\beta} - \underline{r}\sin\alpha\,\underline{\sin\beta}$

$= x_1\cos\alpha - y_1\sin\alpha$

$y_2 = r\sin(\alpha+\beta) = \underline{r}\sin\alpha\,\underline{\cos\beta} + \underline{r}\cos\alpha\,\underline{\sin\beta}$

$= x_1\sin\alpha + y_1\cos\alpha$

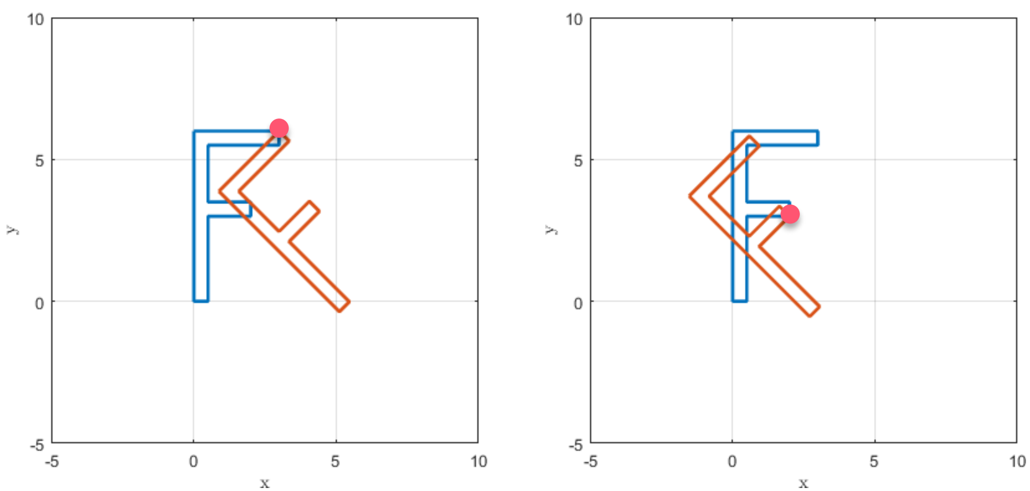
⑥

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\alpha \cdot x_1 - \sin\alpha \cdot y_1 \\ \sin\alpha \cdot x_1 + \cos\alpha \cdot y_1 \end{bmatrix} = \boxed{\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}}$$

Example – rotation around an arbitrary point $\mathbf{x}_r = \begin{bmatrix} x_r \\ y_r \end{bmatrix}$:

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

Example – rotation around an arbitrary point $\mathbf{x}_r = \begin{bmatrix} x_r \\ y_r \end{bmatrix}$:



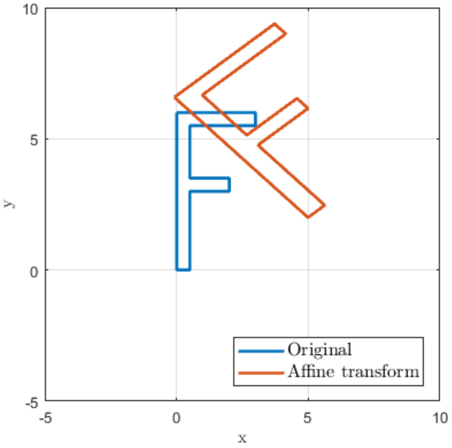
Recap (previous lecture)

Rigid transformations $\mathbf{x}' = \mathbf{R}\mathbf{x} + \mathbf{t}$

- Translation $\mathbf{t} =$
- Rotation $\mathbf{R} =$

Affine transformations: translation, rotations and $\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{t}$

- Scaling $\mathbf{S} =$
- Shearing $\mathbf{H} =$

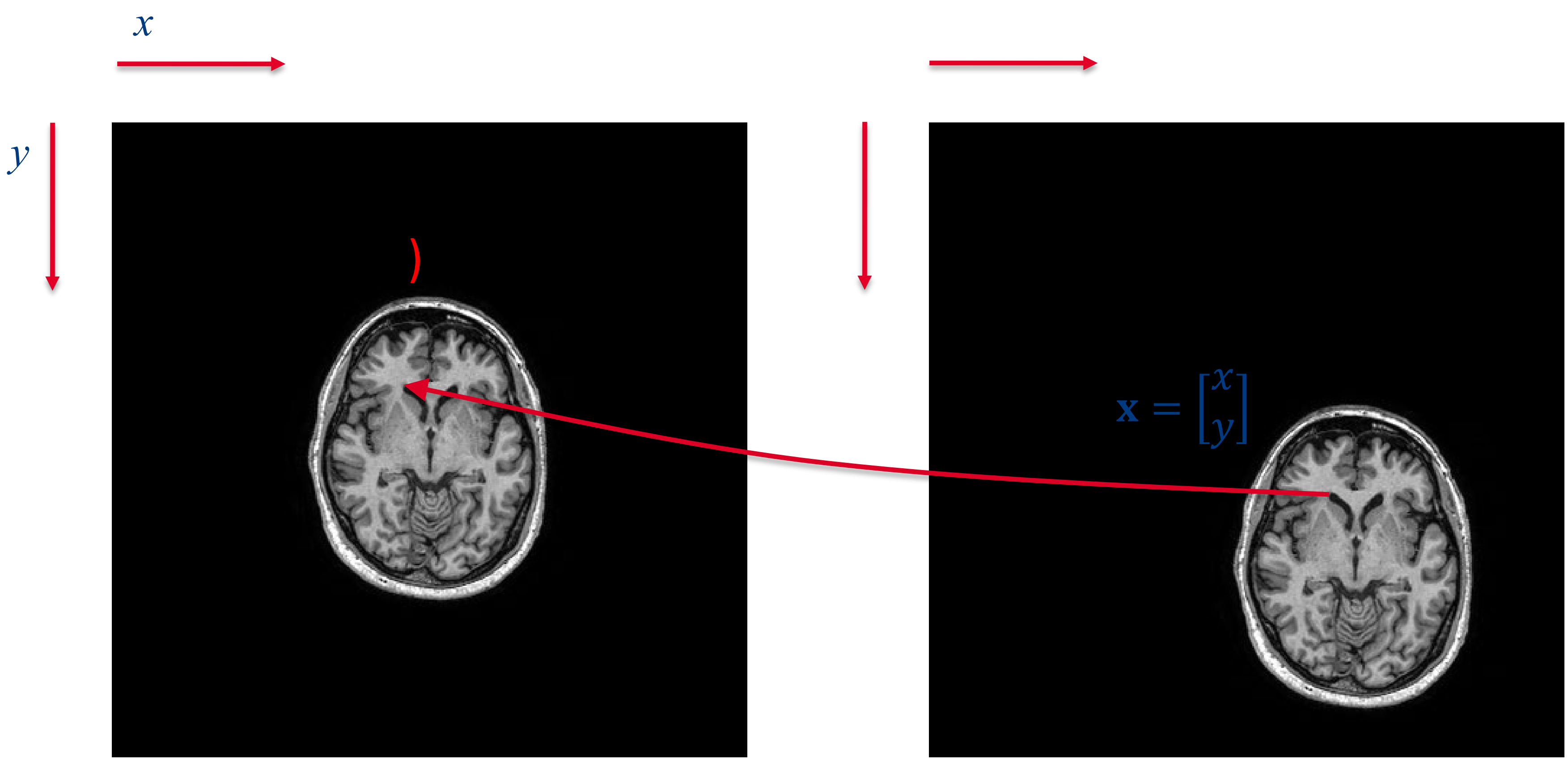


Recap of learning outcomes (previous lecture)

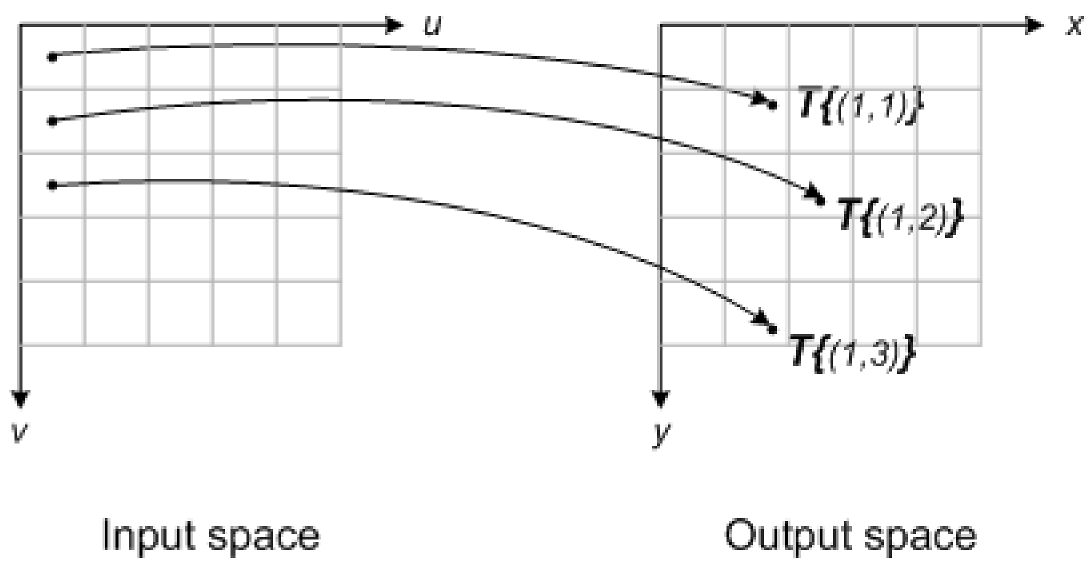
The student can:

- name possible causes of misalignment in medical images
- name different applications of medical image registration
- classify medical image registration methods using eight different criteria
- apply the basic principles of linear algebra (i.e., matrix-vector, vector-matrix products, transpose, norms, orthogonality, determinant) to image registration tasks
- use the determinant of a transformation matrix T to predict the orientation and magnification of an object transformed with T
- compose and combine rigid and affine transformations in 2D and 3D (and rewrite them using homogeneous coordinates)
- explain the difference between affine and non-linear registrations

Image transformation


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

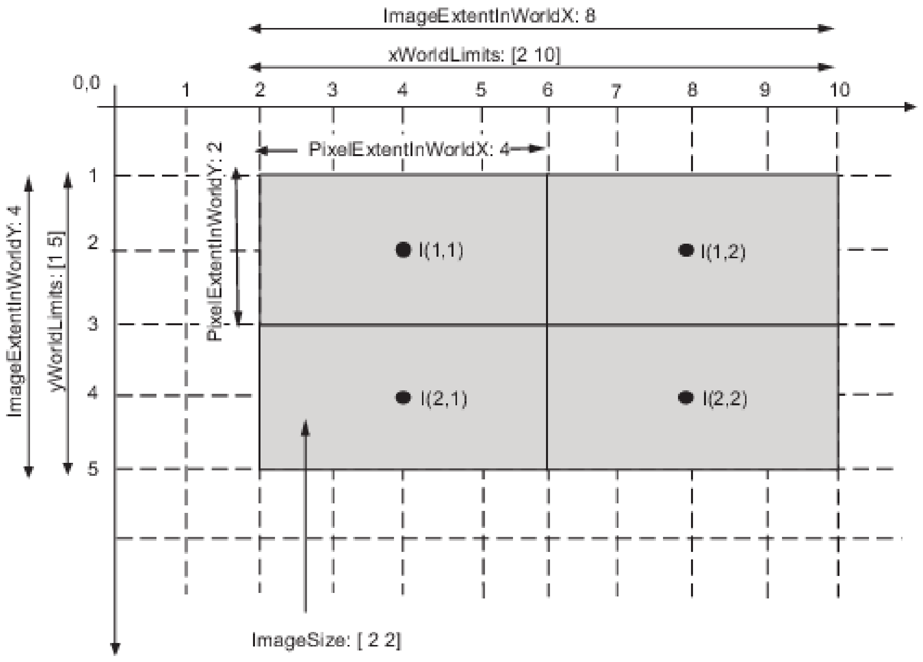
Transforming an image means transforming the spatial coordinates of the pixels.



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:

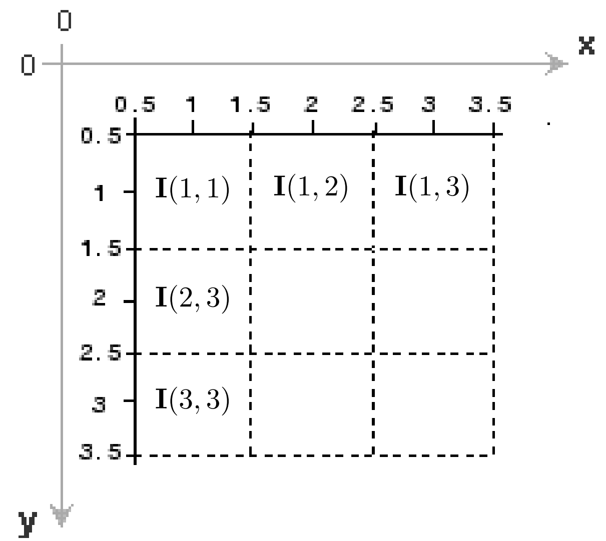
- Spatial coordinates – the coordinates in some world coordinate system where the pixel intensity value “appears”.
- Extent – the physical size of the pixel.



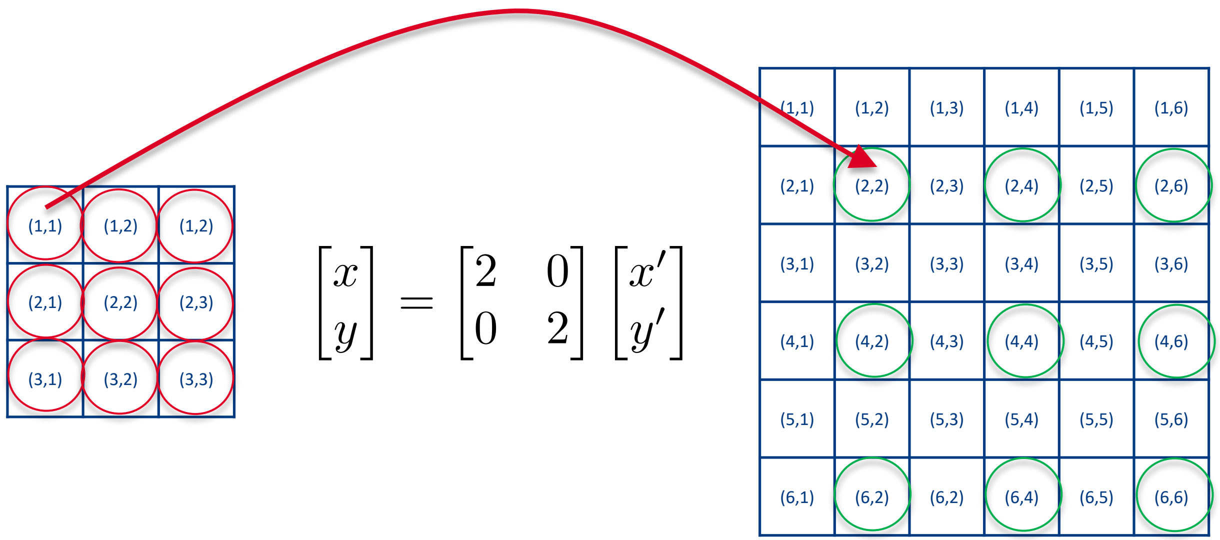
To simplify the discussion and the implementation in the practicals, we are going to assume that the pixel indices correspond to the spatial coordinates.

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.

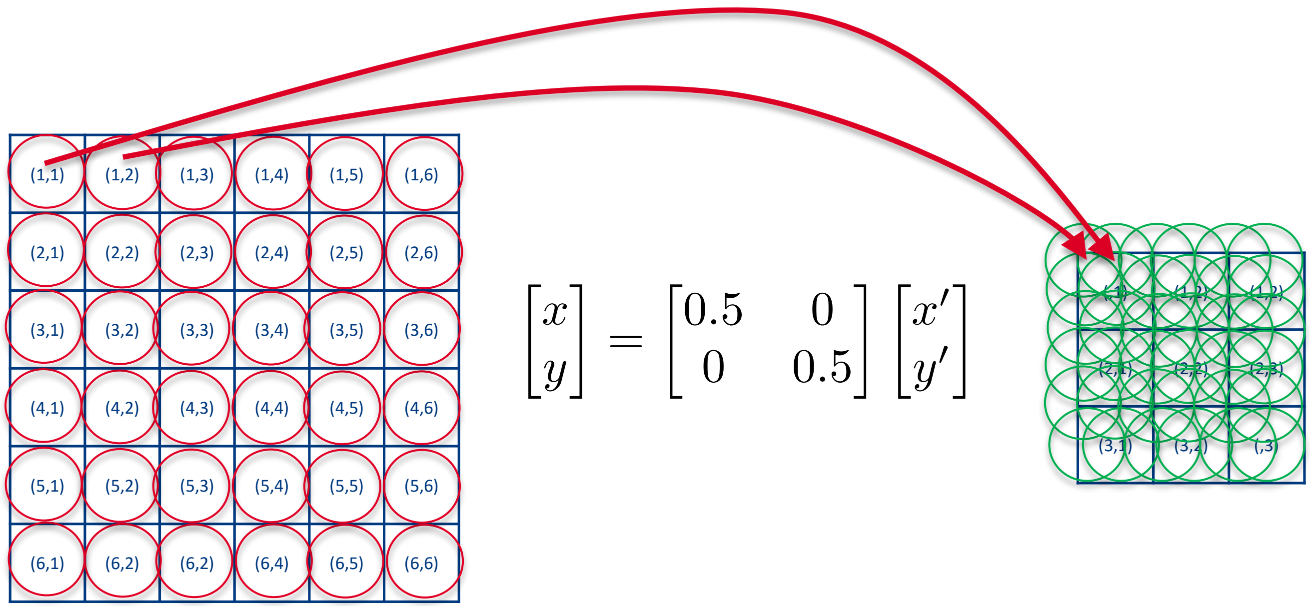


The problem with forward mapping of the coordinates: gaps.



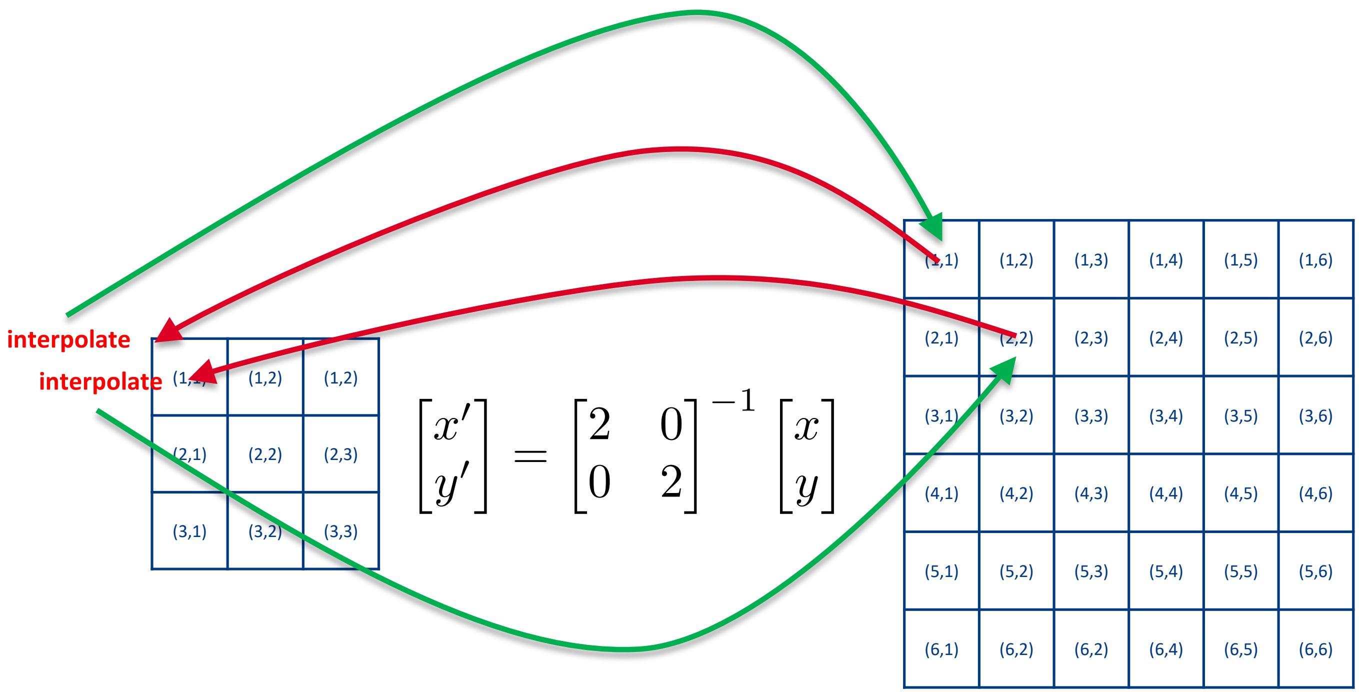
$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

The problem with forward mapping of the coordinates: overlaps.



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

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

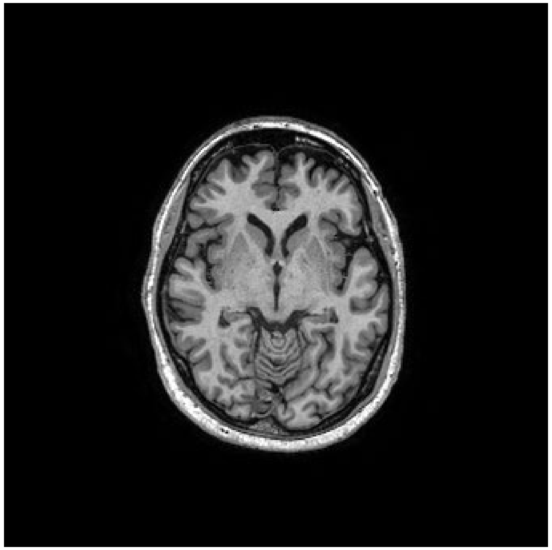


$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix}$$

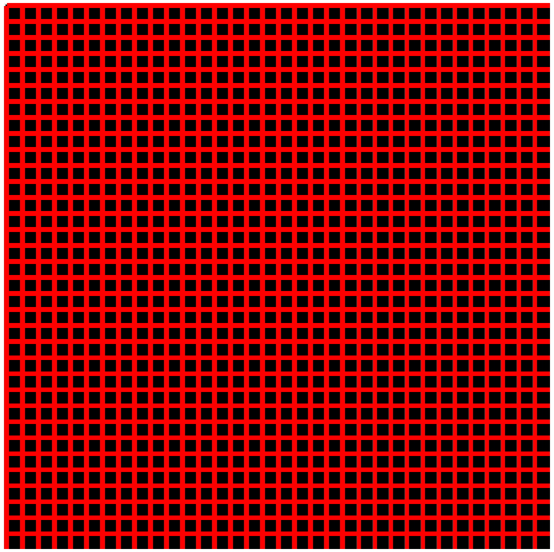
Transforming an image with the transformation $\mathbf{T}$ by inverse mapping:

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

Example:

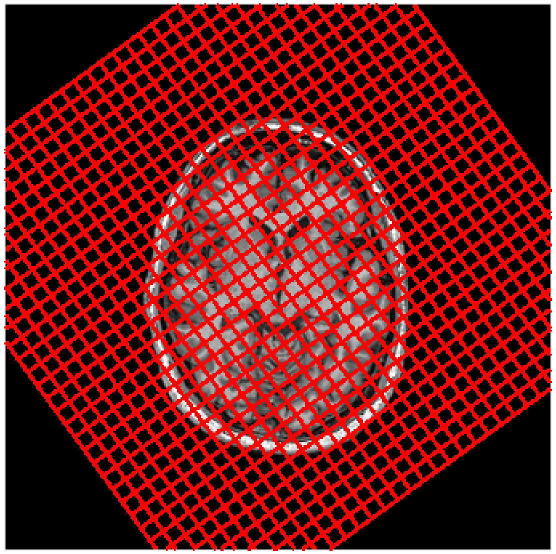


Input image

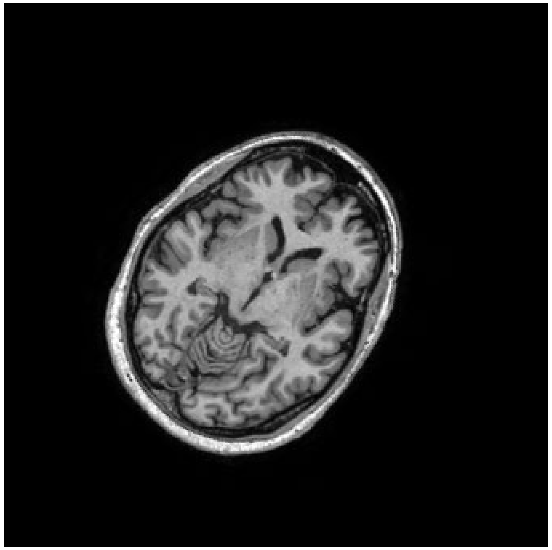


Transformed image (now empty) – we want a π/5 rotation of the input

Example:



Grid of the output transformed
with the inverse transformation



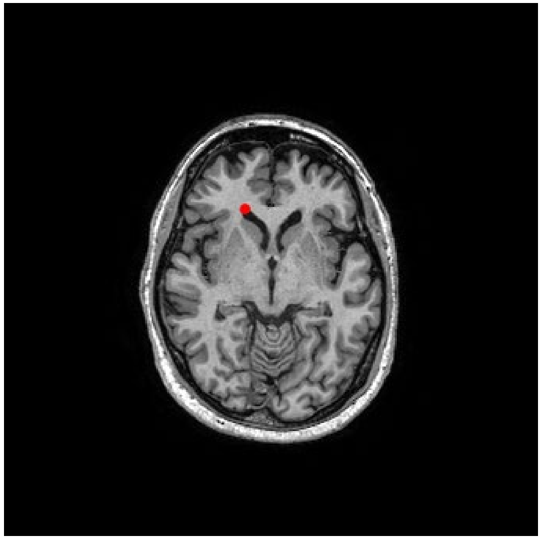
Output image

Point-based image registration

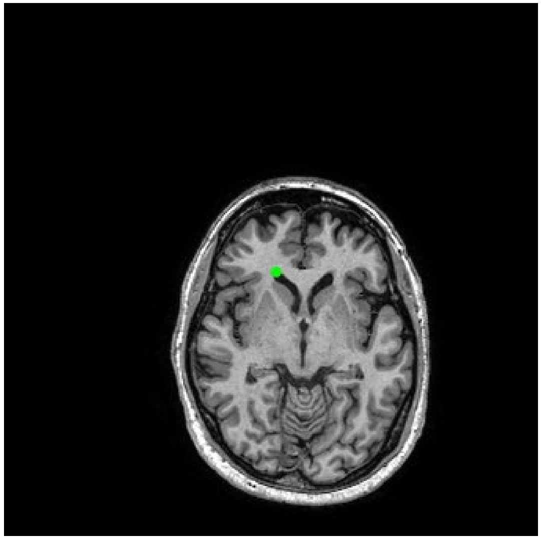
Example:

x

y



Fixed



Moving

Assume two images are misaligned only with a translation:

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

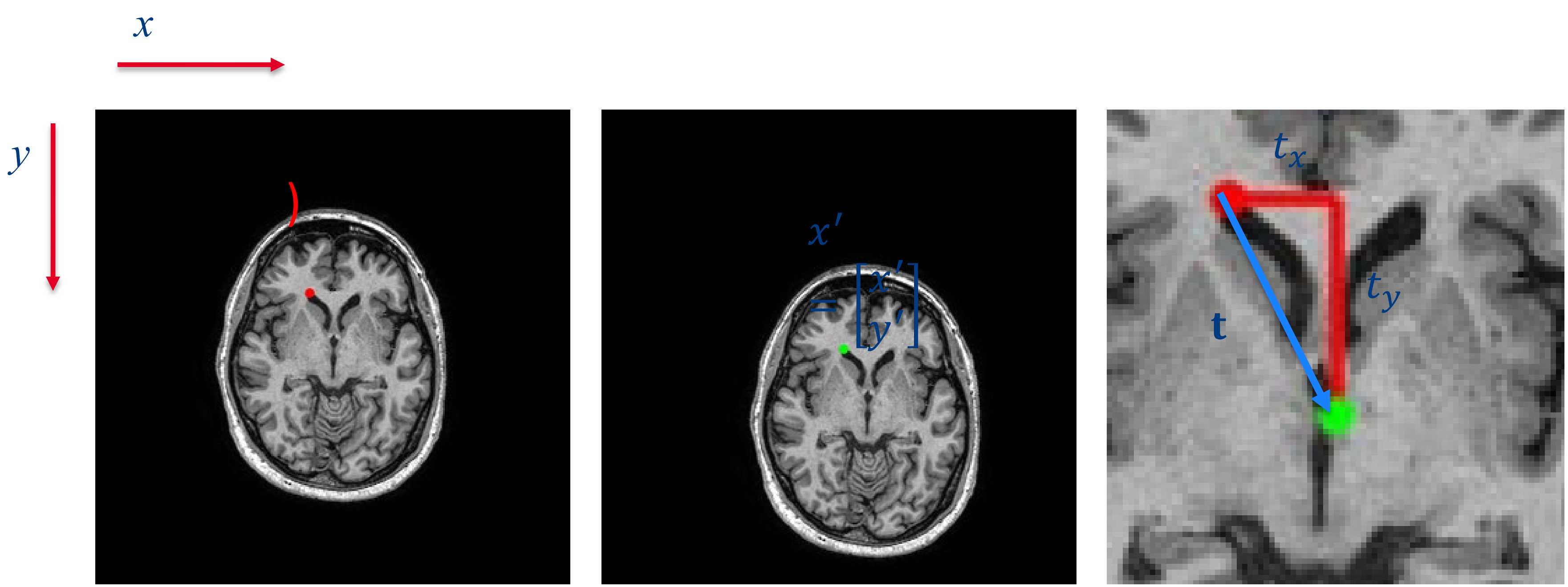
A simple registration algorithm that works for this example:

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:

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

1. Transform the moving image by translating it by -$\mathbf{t}$

Example:



Fixed Moving

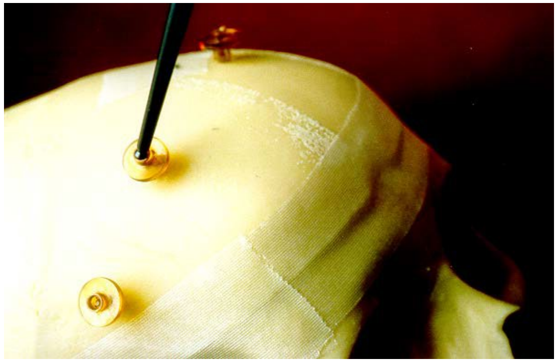
Such points that are taken to be reliable for image registration are called **fiducial points** or **fiducials**.

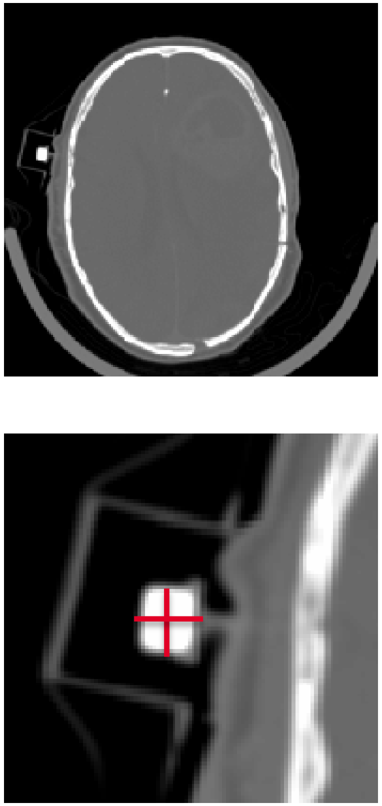
Fiducials can be placed at either intrinsic or extrinsic features.

Intrinsic features: e.g. anatomical landmarks.

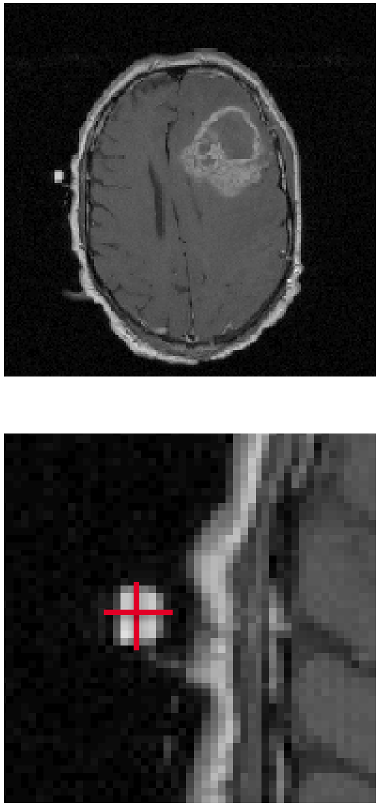
Extrinsic features: e.g. implanted markers.

Extrinsic fiducials



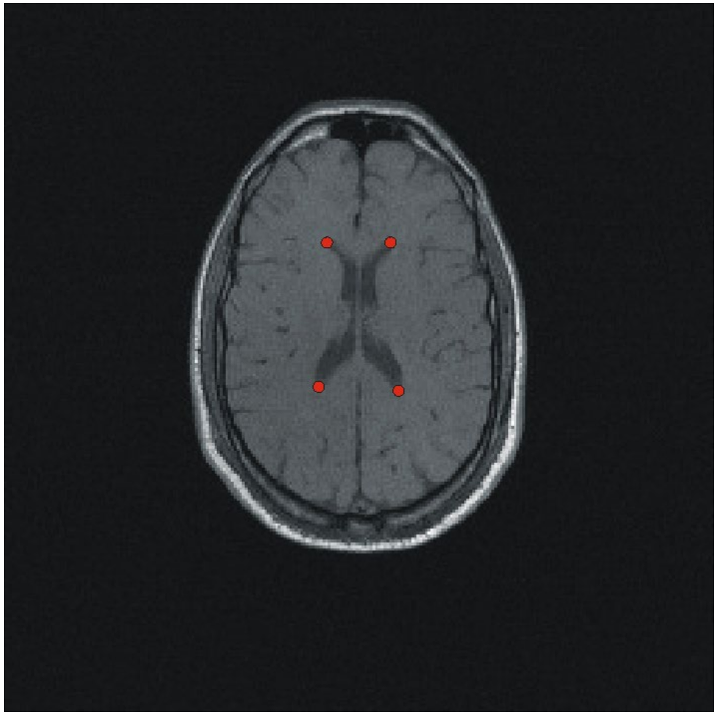


CT

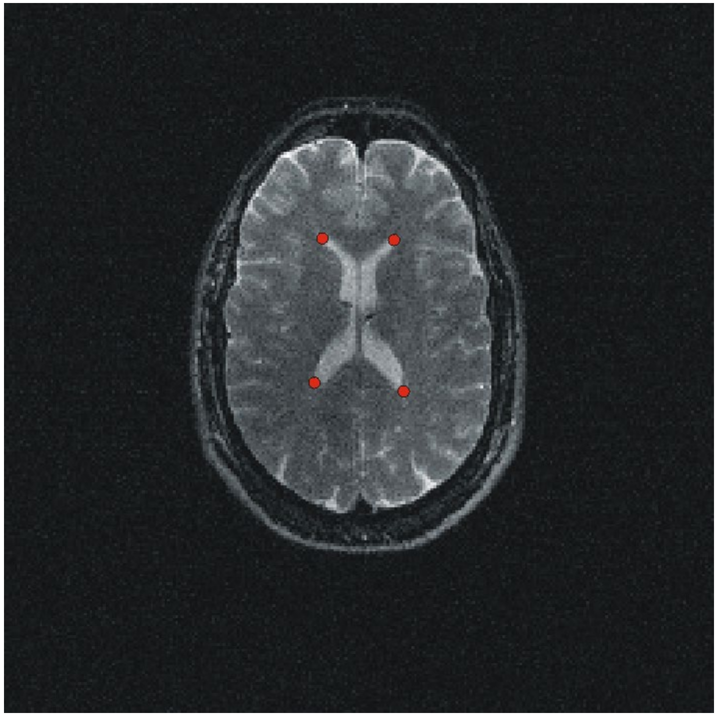


MR

Intrinsic fiducials



MR T1 weighted

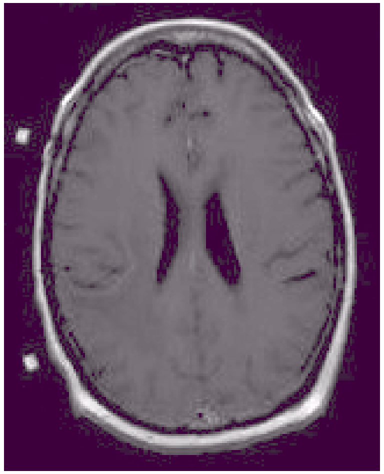


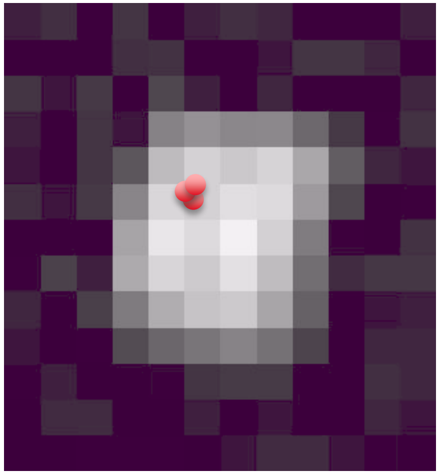
MR T2 weighted

A perfect alignment of the fiducials is usually not possible.

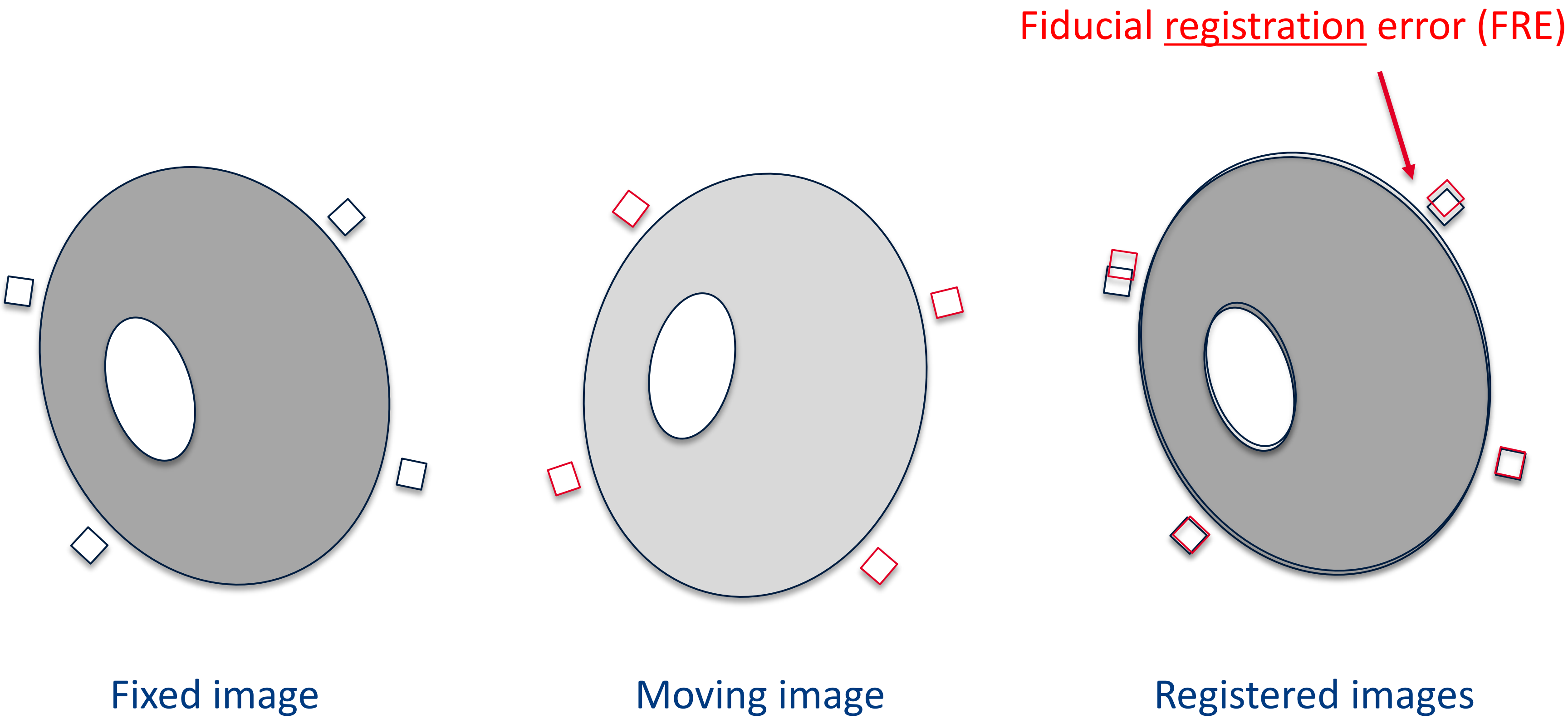
Reasons:

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





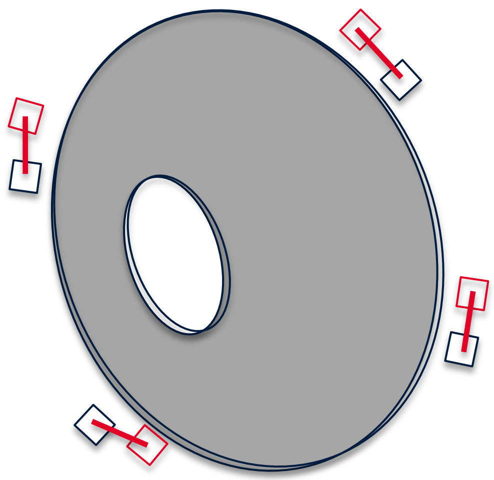
Example:


Fiducial registration error (FRE)
Fixed image
Moving image
Registered images

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.

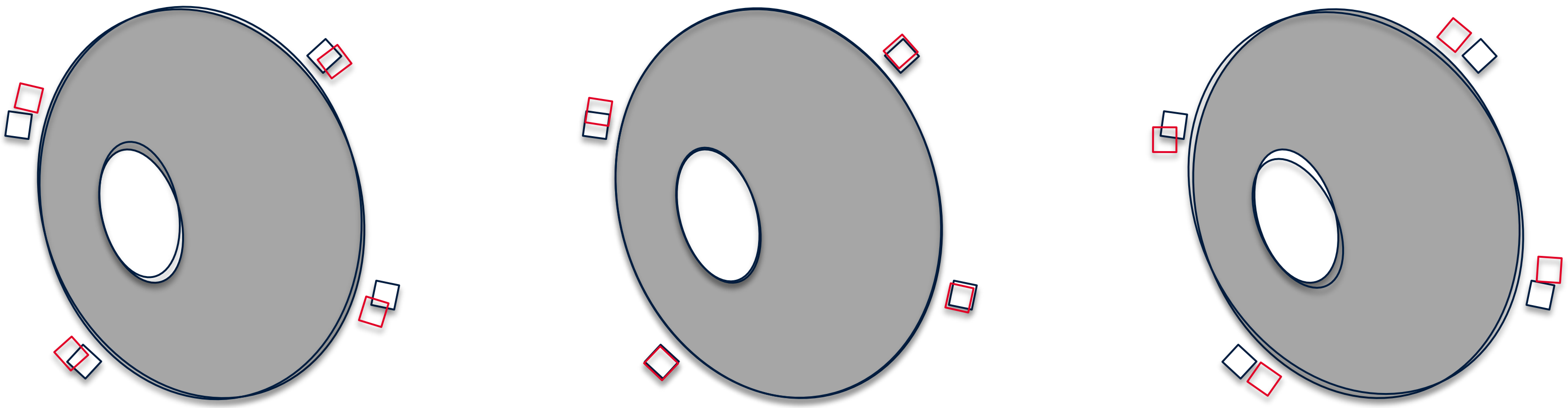
However, we can design an algorithm that will find a transformation that results in the best possible alignment given that fact that there will always be some error.

Best possible alignment = lowest error.



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.



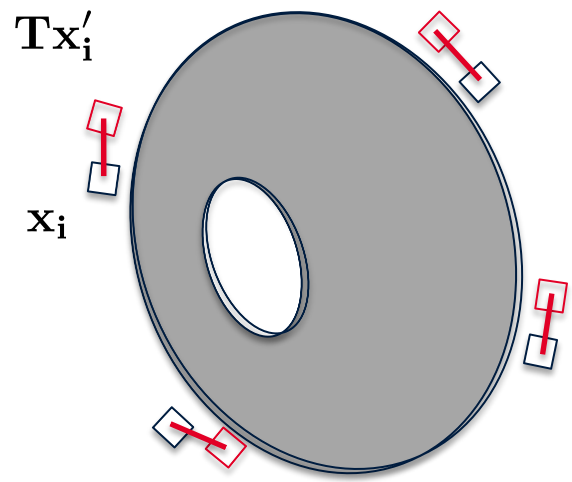
Different transformations will result in different errors.
Our goal: find the transformation that results in the lowest error.

Step 1, affine registration:

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

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

Error as a function of
the transformation T



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

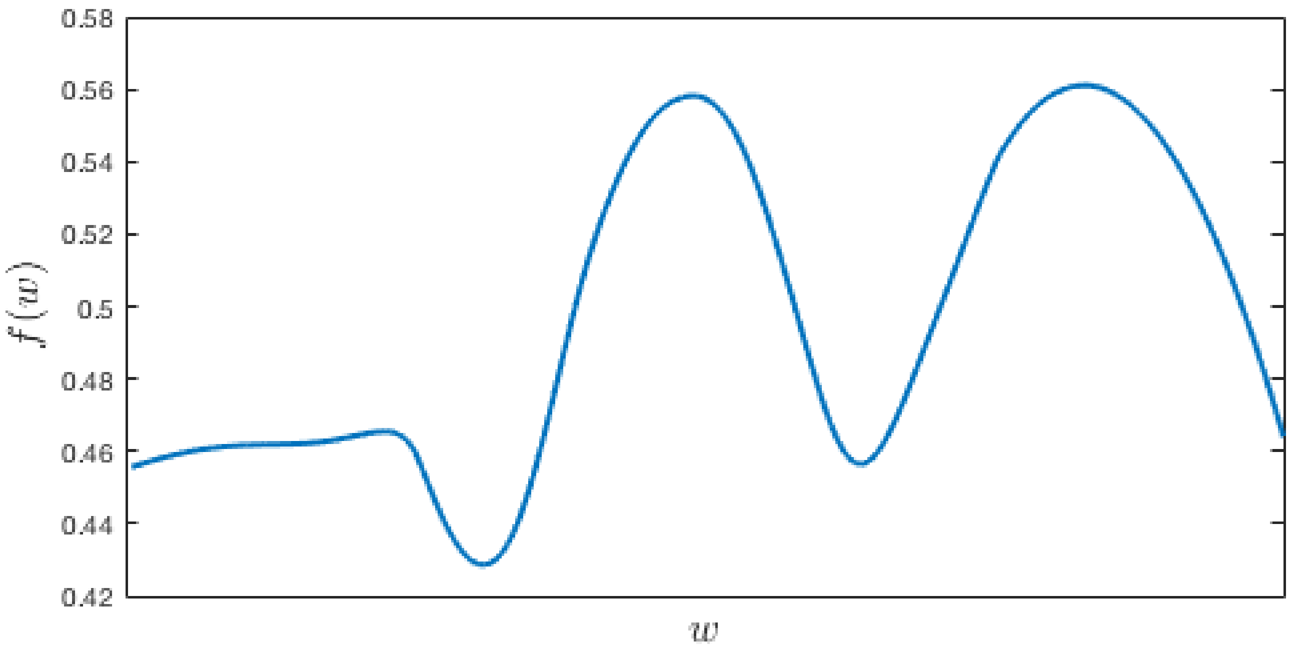
$$\hat{\mathbf{T}} = \arg\min_{\mathbf{T} \in A} \sum_{i=1}^{n} ||\mathbf{T}\mathbf{x}'_\mathbf{i} - \mathbf{x_i}||_2^2 \qquad \mathbf{T} = \begin{bmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

The optimal $\mathbf{T}$ is the one that minimizes the error

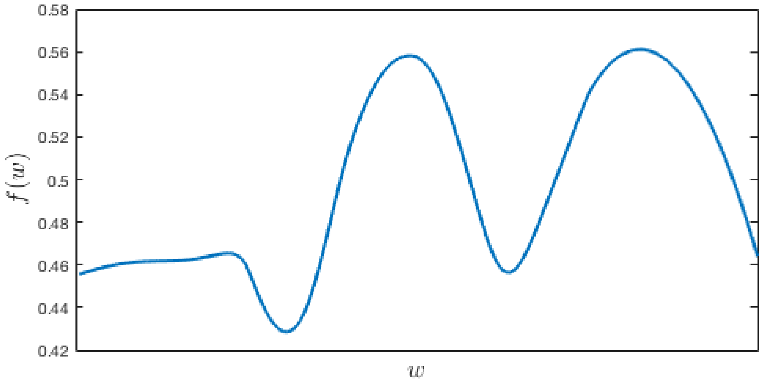
$\mathbf{T}$ belongs to the set of all affine transformations

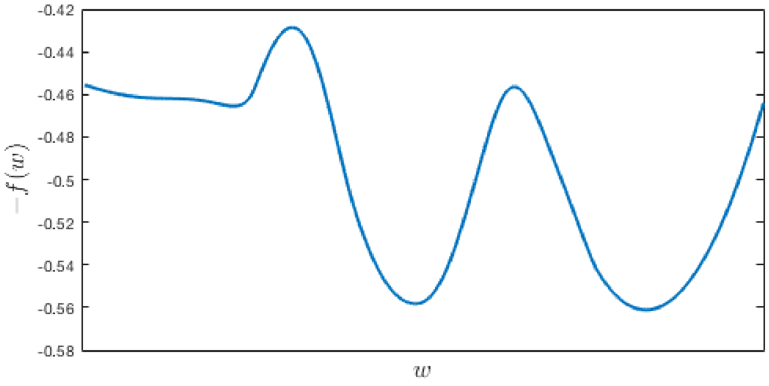
Optimization

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

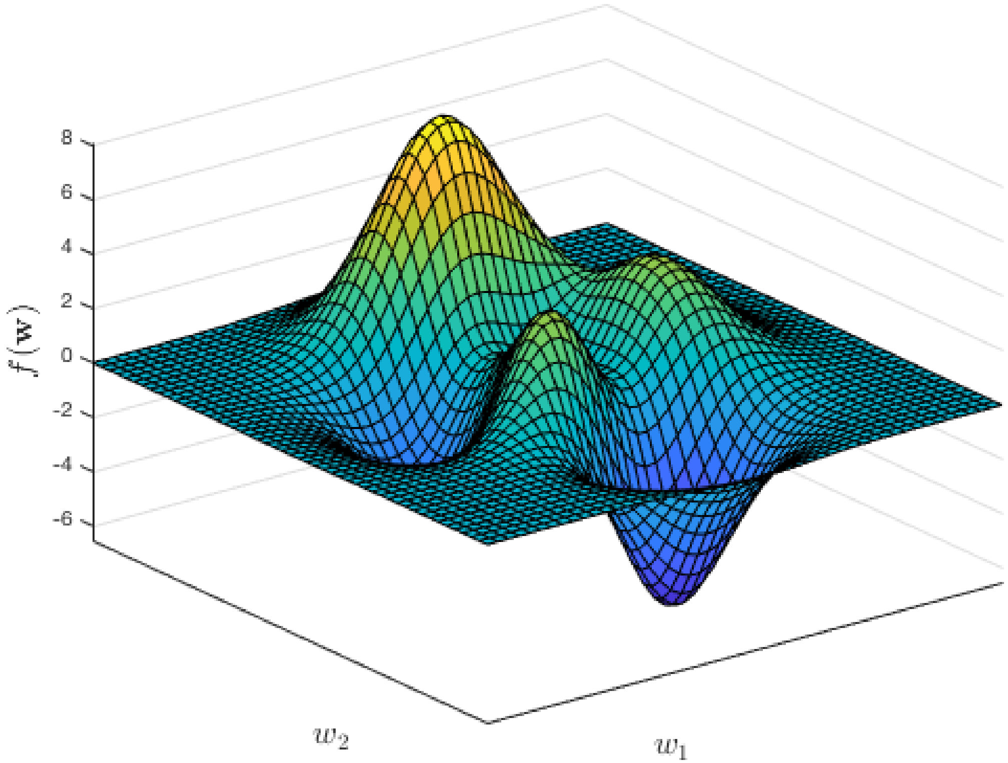


If we have a method that finds the maximum of a function it can be easily used to find a minimum by inverting the function.

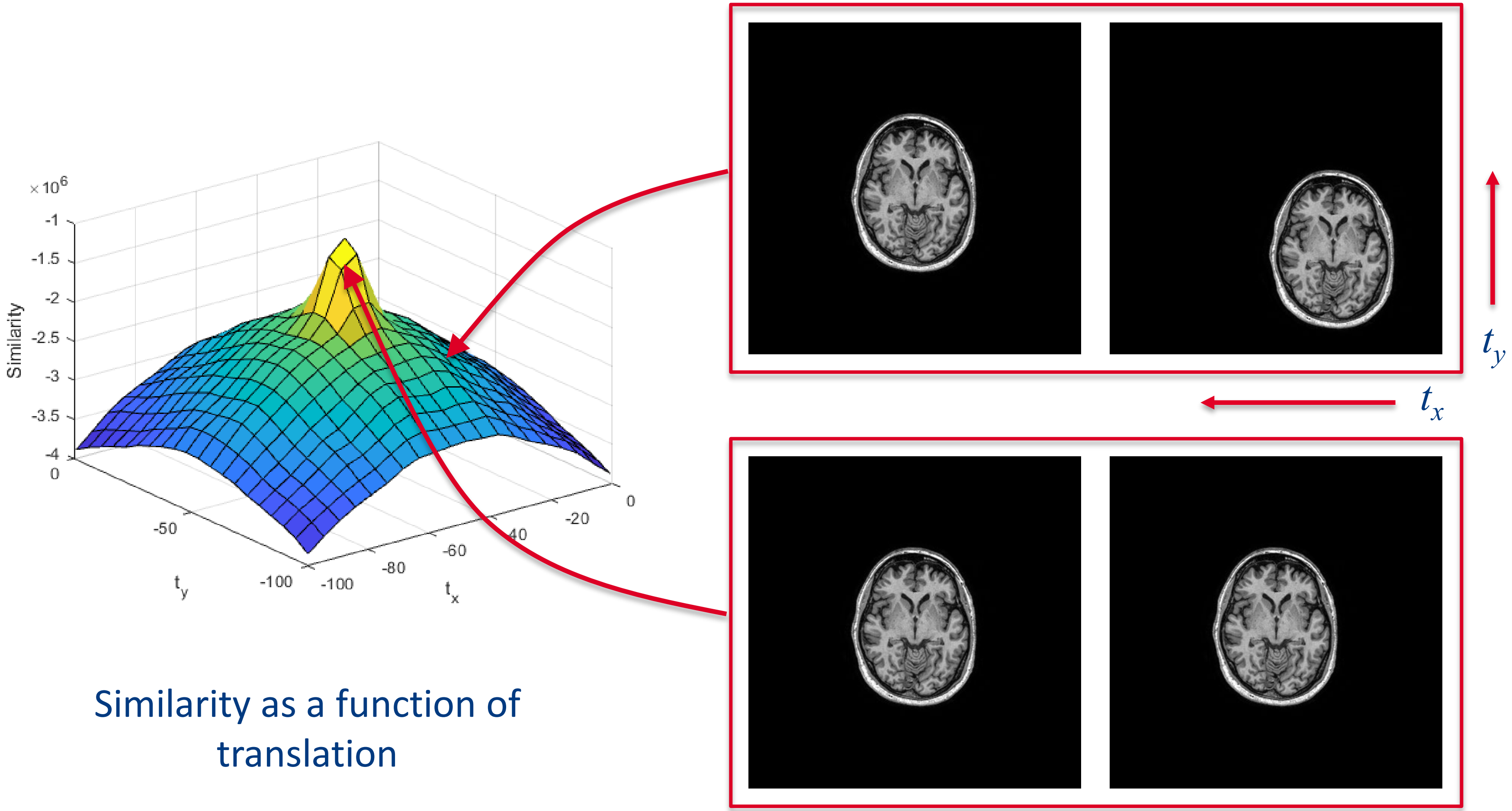




Optimization involves finding some “best” parameters according to an “objective function”, which is either minimised or maximised. 2D example:



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



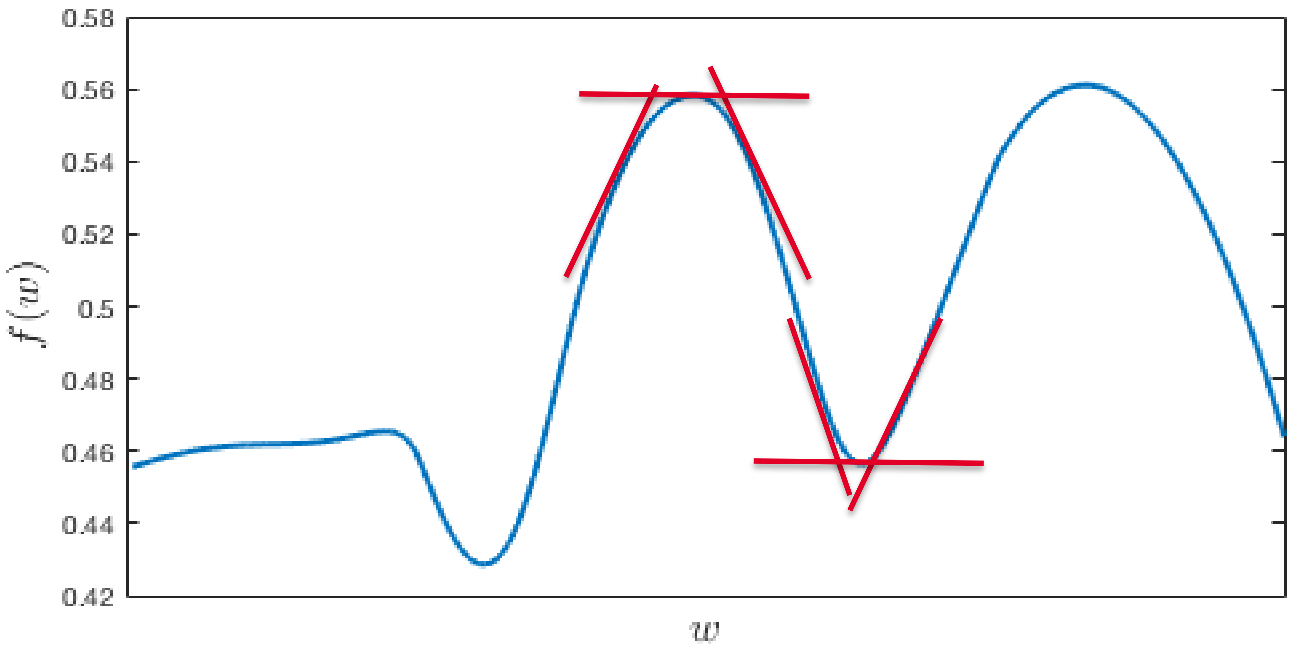
Similarity as a function of translation

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 $20^6 = 64000000$.

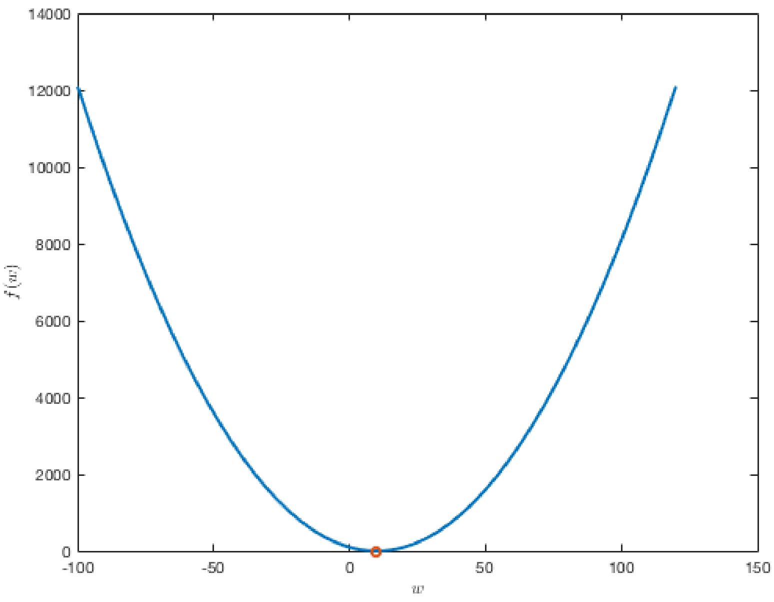
If one evaluation takes just 0.01 second, it would take just over a week to register a pair of images.

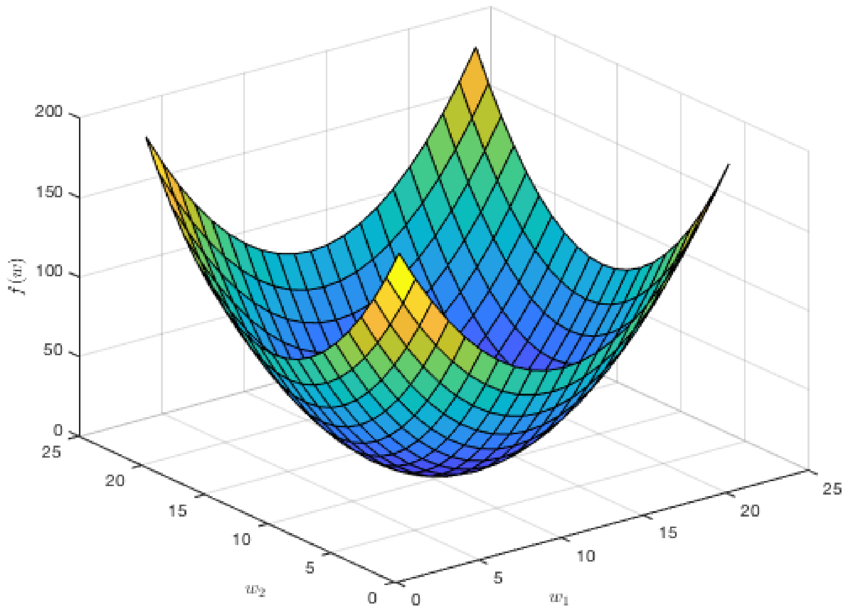
How to find the min. and max. of this function analytically?



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.

Quadratic functions:



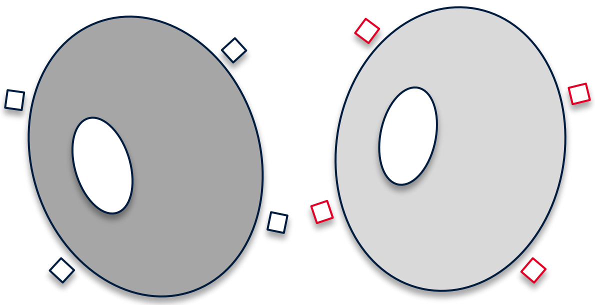


Step 2, find the minimum of the error w.r.t. to the parameters.

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

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

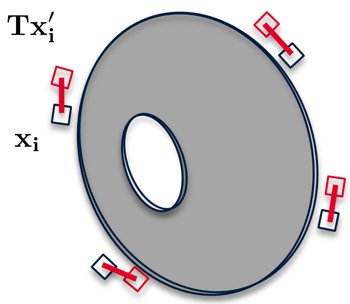
Frobenius norm



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

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

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



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:

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

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

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

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:

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

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

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

Compared with affine registration, the solution is now **constrained** 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.

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.

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

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

$$\mathbf{H} = \sum_{i=1}^{n} \hat{\mathbf{x}}_i' \hat{\mathbf{x}}_i^{\mathsf{T}}$$

4. Perform singular value decomposition of the covariance matrix.

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

5. Compute the rotation matrix:

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

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

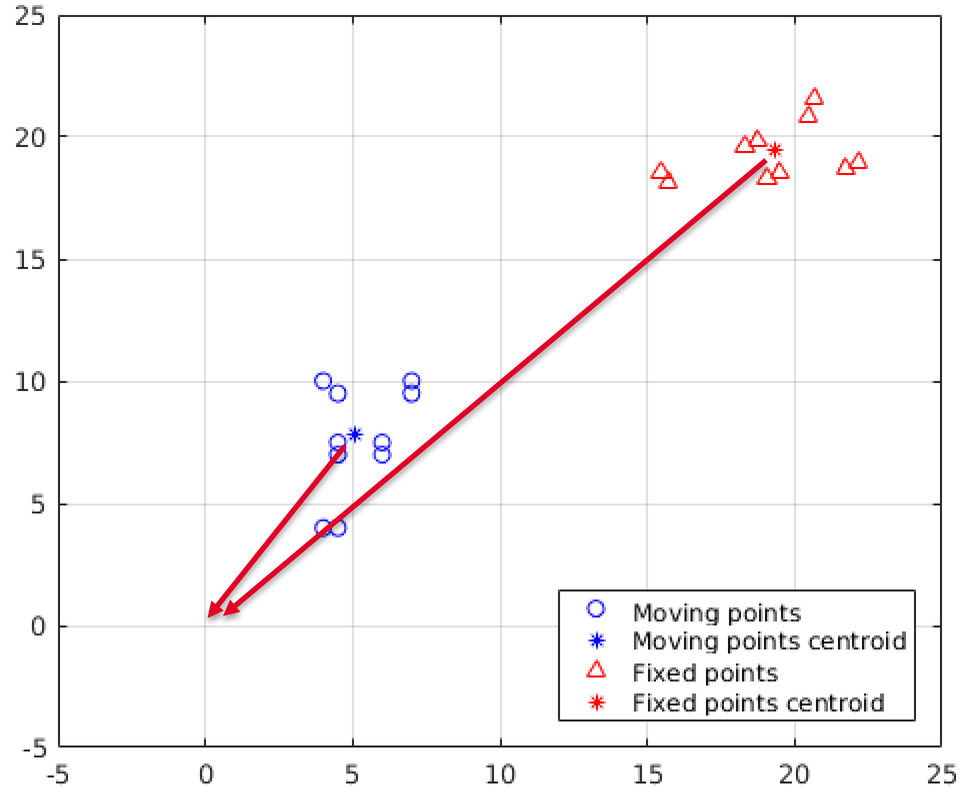
6. Compute the translation vector:

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

Compute the centroids of the two point sets:

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

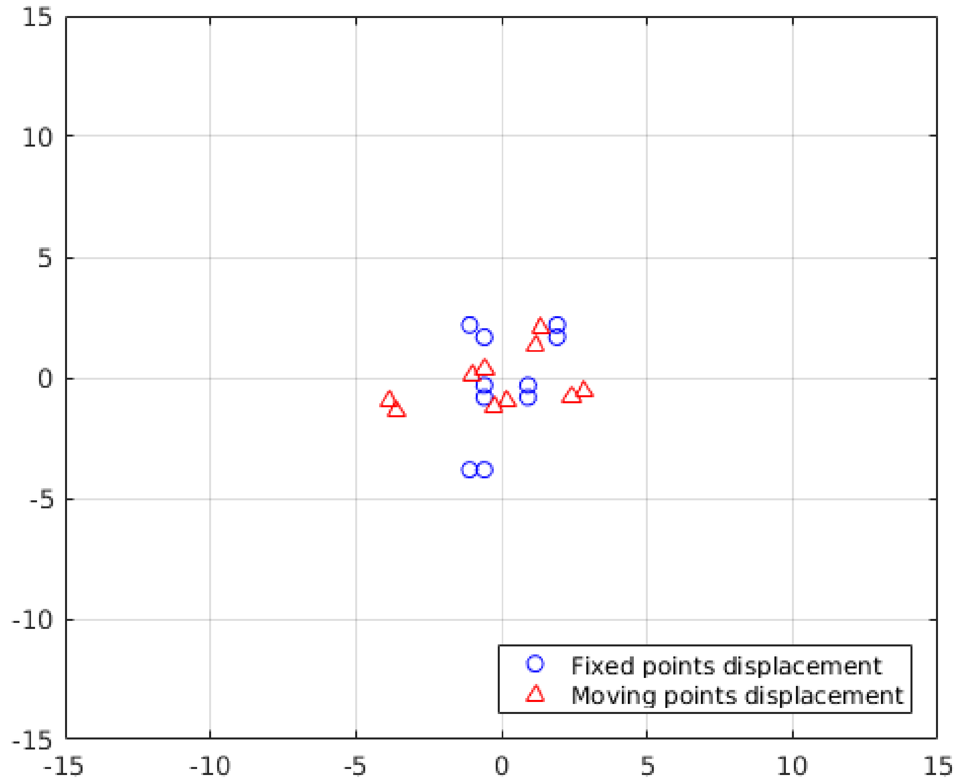
$$\bar{\mathbf{x}}' = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x'}_i$$



Compute the displacements:

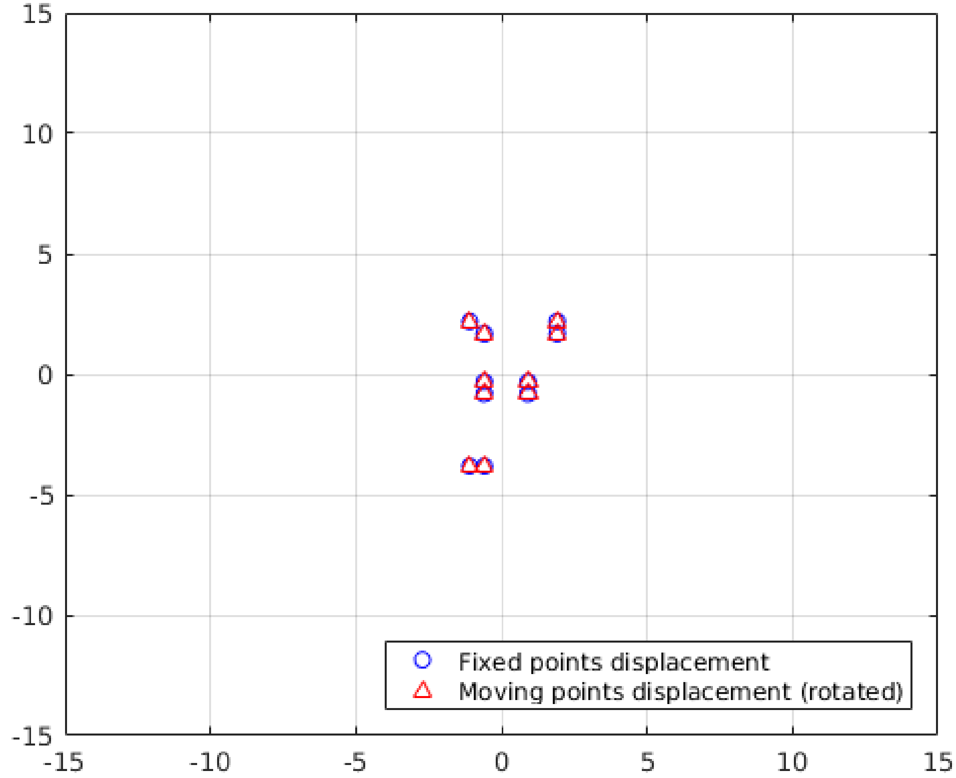
$$\hat{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$$

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



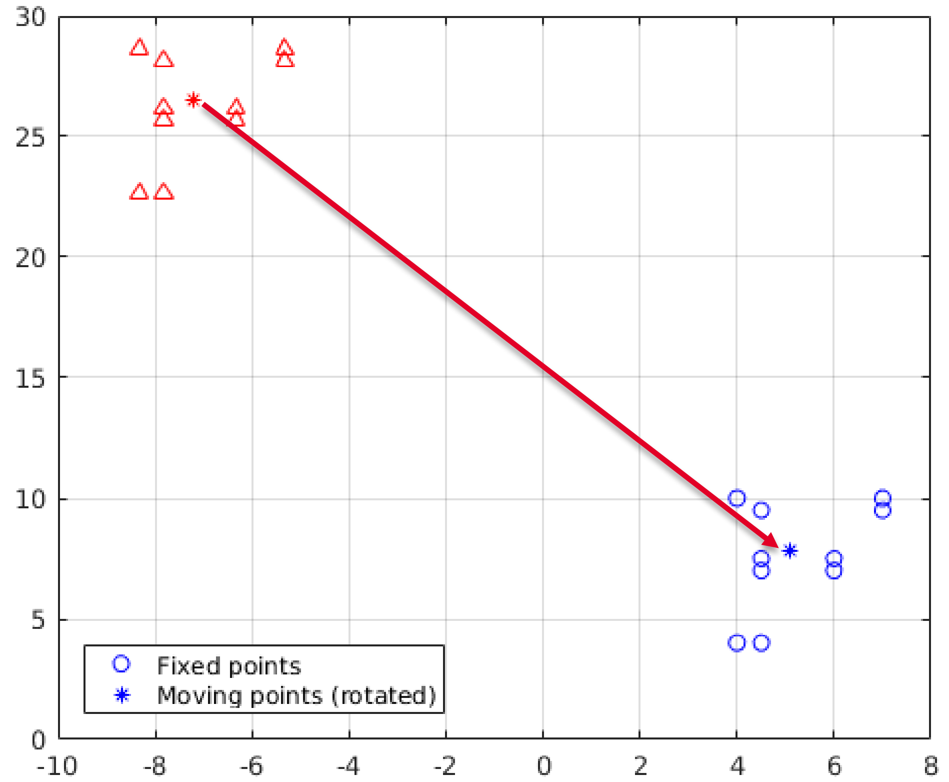
Compute the rotation matrix:

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



Compute the translation:

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



Thus far we assumed that the correspondence between the points in the moving and fixed images is know.

What if we have the two sets of points but we do not know the correspondence?

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

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

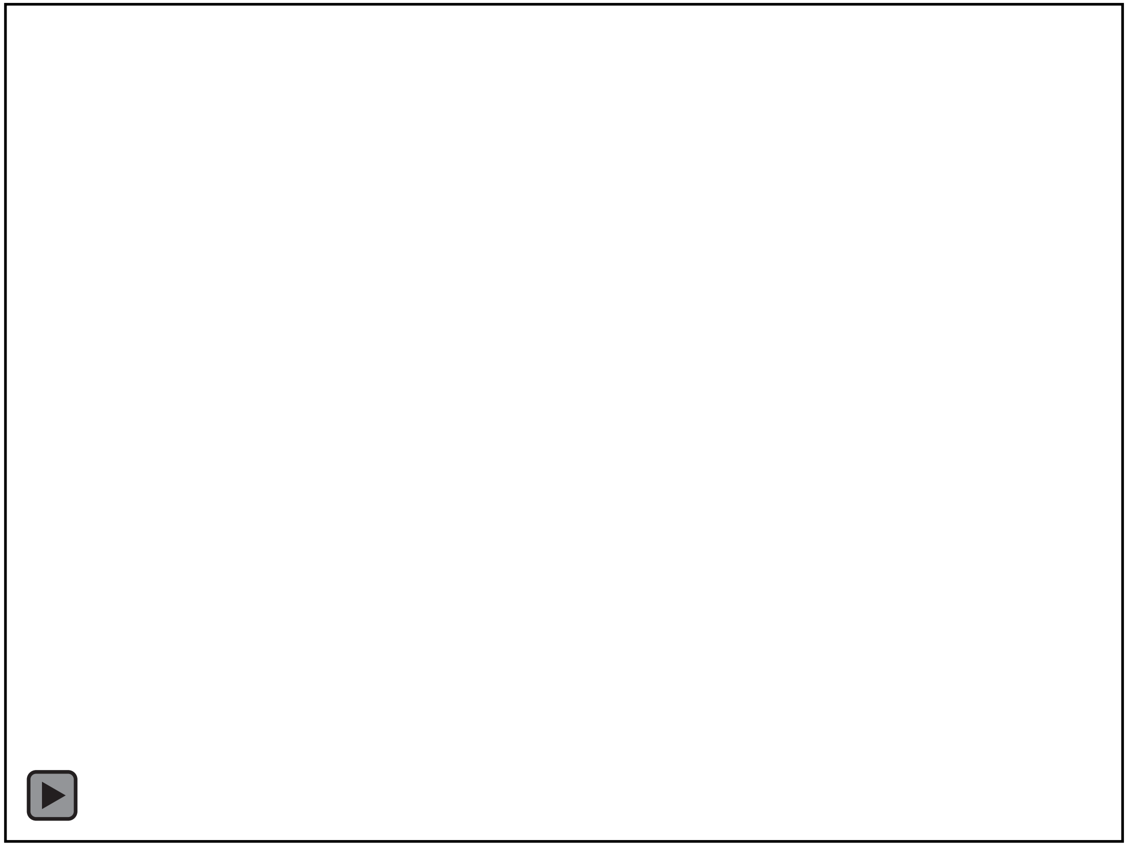
This leads optimization problems that are usually solved with an iterative procedure.

Example: iterative closest point.

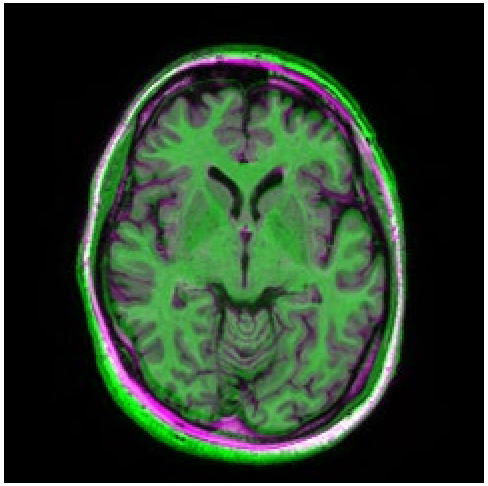
Iterate until convergence:

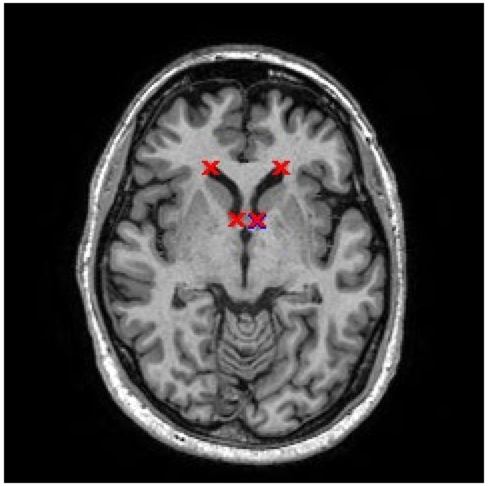
- For each point in the moving shape find the closest point in the fixed shape
- Assume the found correspondence is correct and register (using one of the algorithms that work with correspondence)

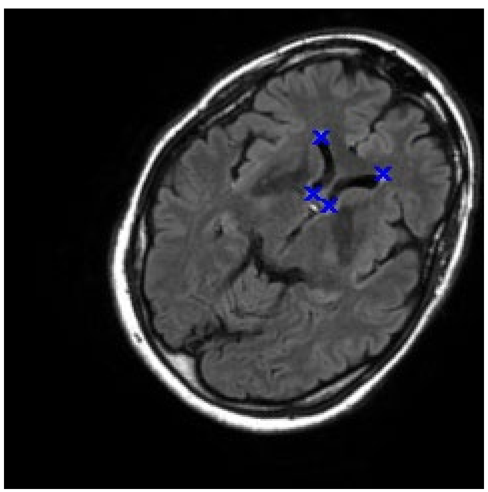
Example: iterative closest point.



Evaluation of the registration accuracy







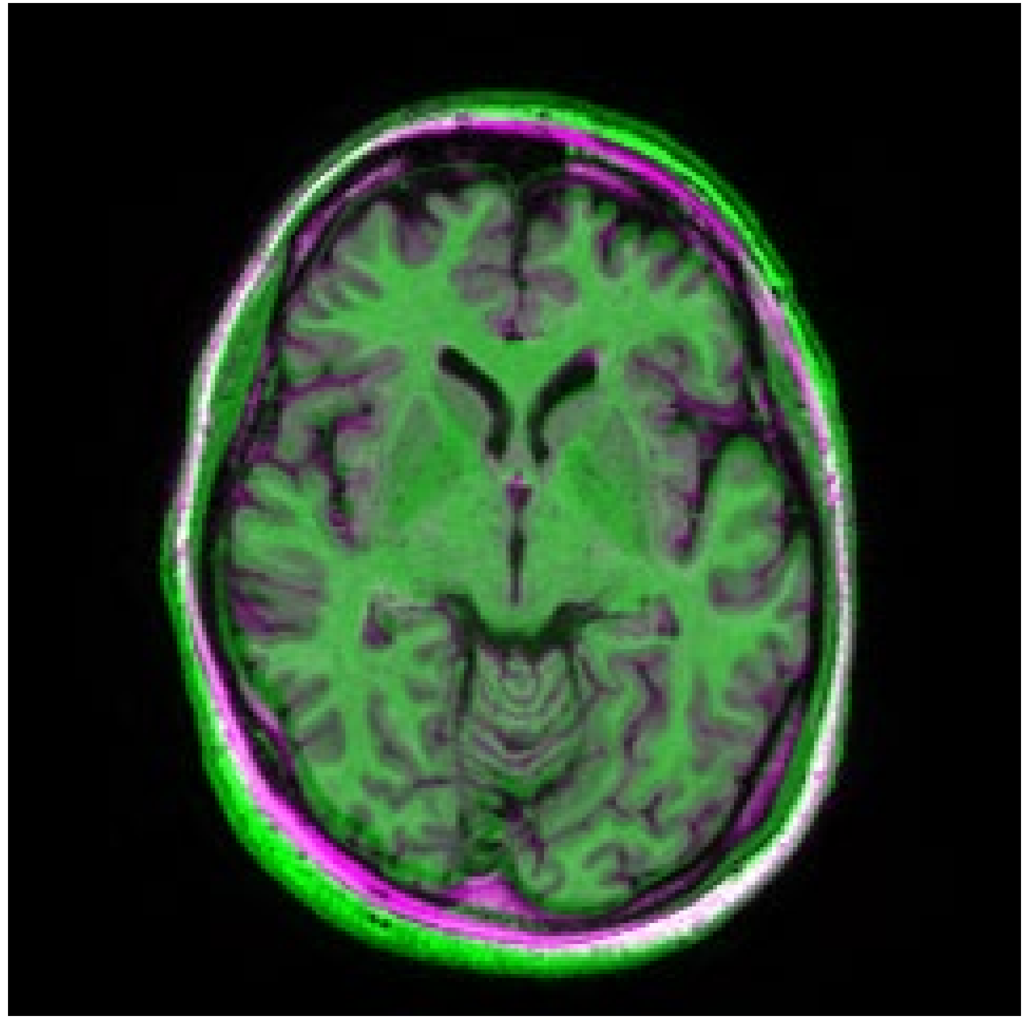


Image registration can be evaluated by computing the registration error for some target corresponding point pairs.

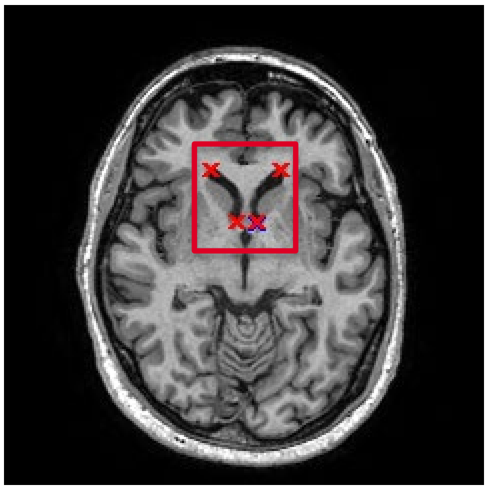
→ Target registration error (TRE)

The target points should be selected in locations that are relevant for some treatment or diagnosis.

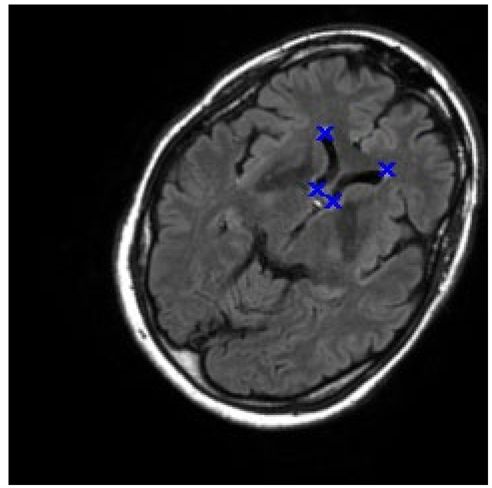
Basically, this is the same as the procedure for selecting corresponding point pairs to compute the transformation.

However, the target corresponding points must be different from the points used to compute the transformation! Why?

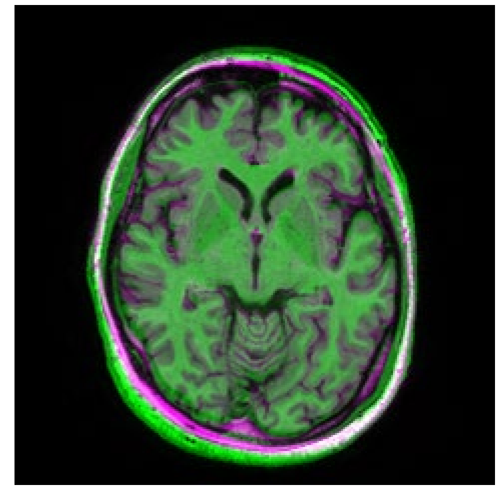
(Poor) Example of affine registration:



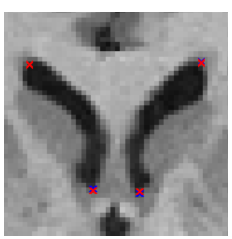
Fixed



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:

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.

Recap of learning outcomes

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 affine 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 rigid registration (orthogonal Procrustes problem)
- explain the principle behind the iterative closest point algorithm
- use the target registration error (TRE) to evaluate image registration