DMIP – Exercise

Rigid Registration

Yan Xia, Marco Bögel
Pattern Recognition Lab (CS 5)

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Registration – Overview

- Images from different modalities (CT, MR, PET, etc.) or from same modality but different exams (e.g., w/wo contrast) probably do not overlap
- Registration establishes a mapping between two coordinate systems to solve this
- *Rigid registration*: Assume a rigid body, only rotation, translation
- *Deformable registration*: Also allow non-rigid deformations (soft tissue!)

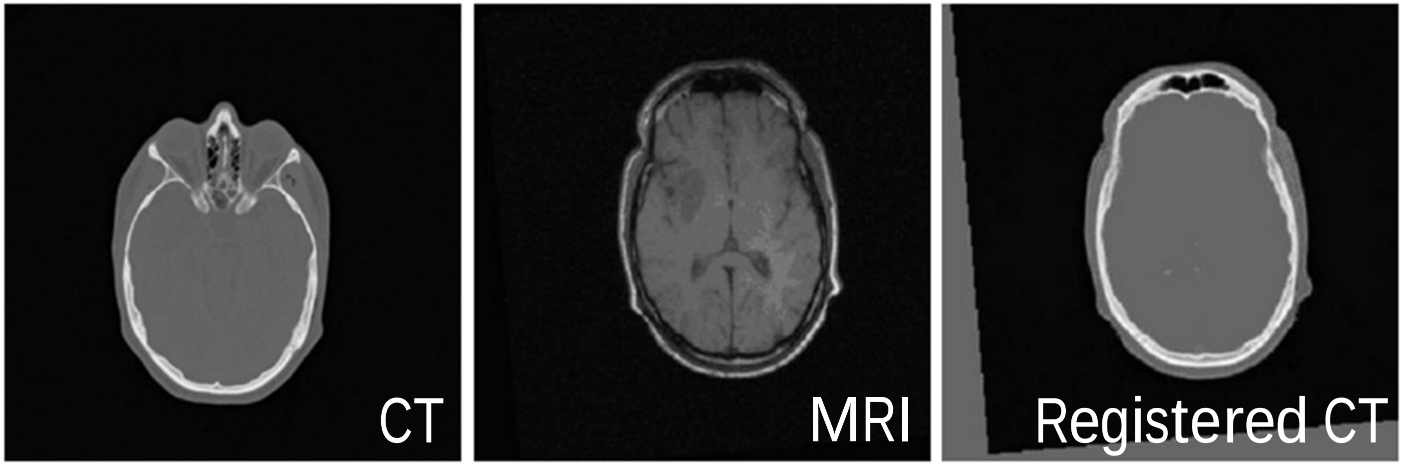


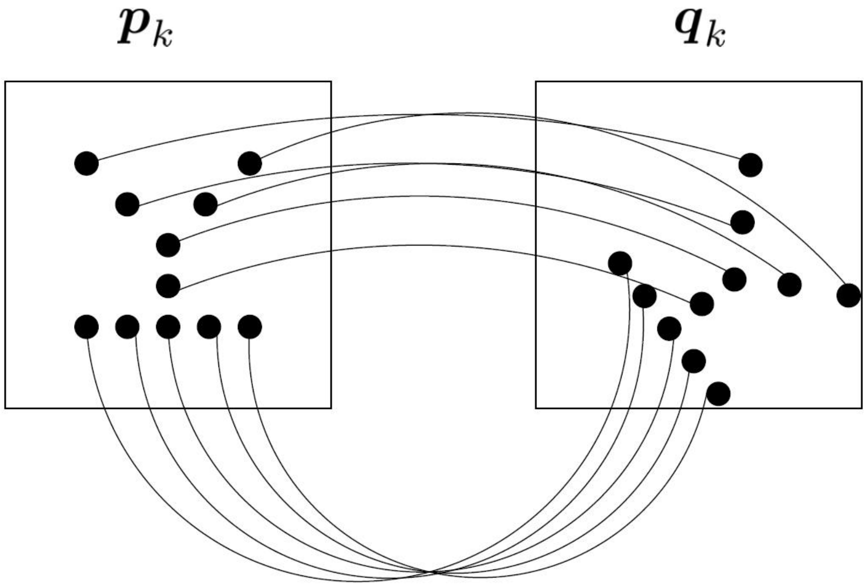
Image: ITK Software Guide

Registration – More Formal

- Assume a set of corresponding 2-D points in two different images

$$C = \{(\boldsymbol{p}_k, \boldsymbol{q}_k)\,;\, k = 1, 2, \ldots, N\}$$

where $\boldsymbol{p}_k, \boldsymbol{q}_k \in \mathbb{R}^2$ is the k-th pair of corresponding image points.



- How are correspondences established?

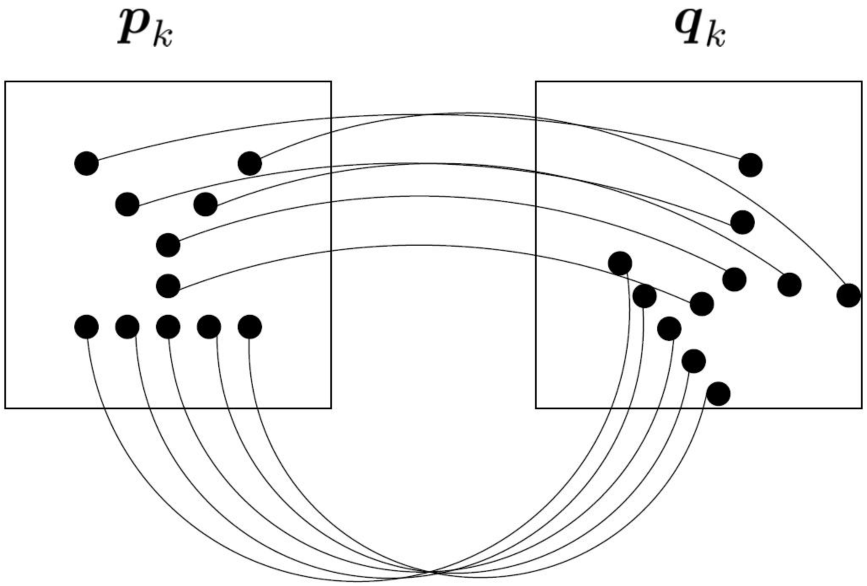
- How do you compute the transform that maps all $\boldsymbol{p}_k$ to all $\boldsymbol{q}_k$?

Registration – More Formal

- Assume a set of corresponding 2-D points in two different images

  $$C = \{(\boldsymbol{p}_k, \boldsymbol{q}_k)\,; k = 1, 2, \ldots, N\}$$

  where $\boldsymbol{p}_k, \boldsymbol{q}_k \in \mathbb{R}^2$ is the k-th pair of corresponding image points.



- How are correspondences established? Markers or similarity metrics
- How do you compute the transform that maps all $\boldsymbol{p}_k$ to all $\boldsymbol{q}_k$? Registration!

Rigid Registration

- A rigid transformation in 2-D:

  $$\boldsymbol{p}_k = \boldsymbol{R}\boldsymbol{q}_k + \boldsymbol{t}$$

  where $\boldsymbol{R}$ is the rotation matrix $\boldsymbol{R} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$

  and $\boldsymbol{t}$ is the translation vector

- Least squares problem:

  $$\arg\min_{\varphi,\boldsymbol{t}} \sum_{k=1}^{N} \|\boldsymbol{p}_k - \boldsymbol{R}\boldsymbol{q}_k - \boldsymbol{t}\|^2$$

- But we would prefer a linear problem!

Rotations in 2-D – Complex Numbers

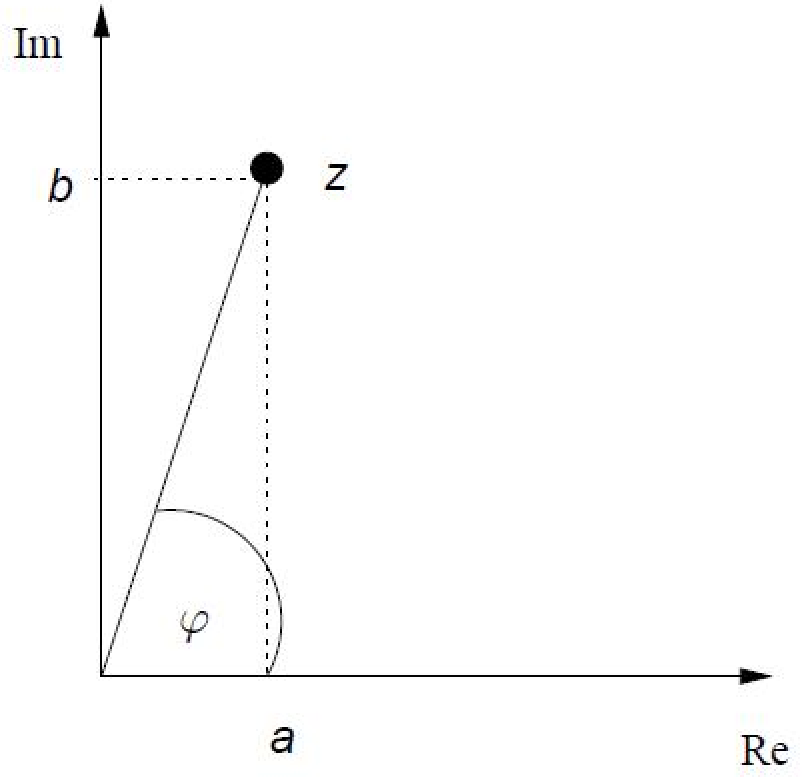
- A complex number can be seen as a (vector to a) point in 2-D:

$$\boldsymbol{z} = a + \mathrm{i}b$$

- There is also the Euler representation of complex numbers:

$$\boldsymbol{z} = |\boldsymbol{z}|\,\mathrm{e}^{\mathrm{i}\varphi}$$

where $|\boldsymbol{z}| = \sqrt{a^2 + b^2}$
is the length of the vector and
$\varphi = \operatorname{atan2}(b, a)$ the angle



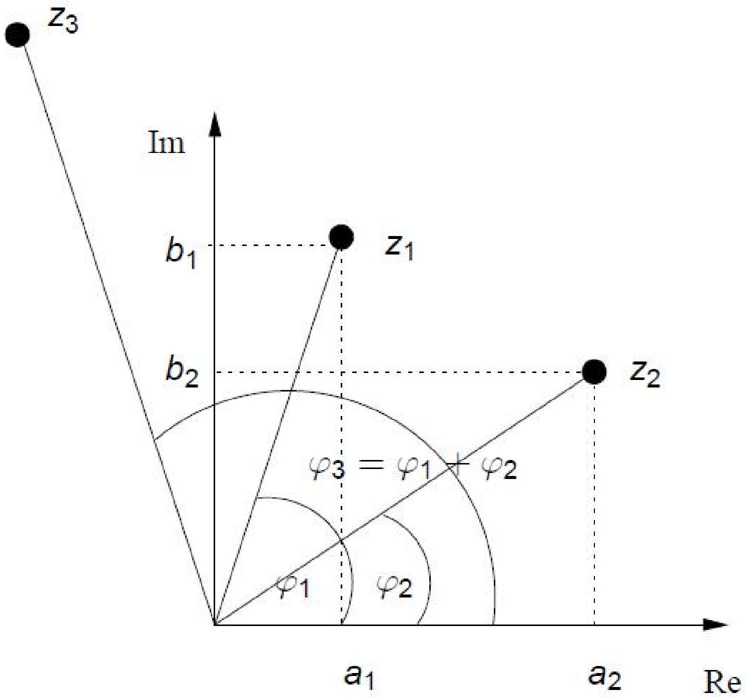
Rotations in 2-D – Complex Numbers

- Multiplication of complex numbers:

$$\boldsymbol{z}_1 \cdot \boldsymbol{z}_2 = (a_1 + \mathrm{i}b_1) \cdot (a_2 + \mathrm{i}b_2)$$

- In Euler notation:

$$\begin{aligned} \boldsymbol{z}_1 \cdot \boldsymbol{z}_2 &= |\boldsymbol{z}_1|\, \mathrm{e}^{\mathrm{i}\varphi_1} \cdot |\boldsymbol{z}_2|\, \mathrm{e}^{\mathrm{i}\varphi_2} \\ &= |\boldsymbol{z}_1|\, |\boldsymbol{z}_2|\, \mathrm{e}^{\mathrm{i}(\varphi_1+\varphi_2)} \end{aligned}$$



Rotations in 2-D – Complex Numbers

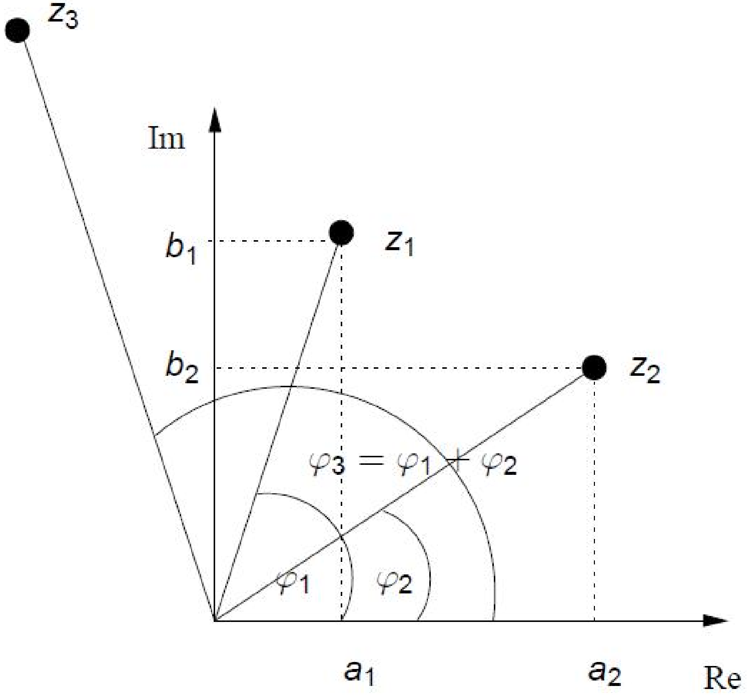
- Multiplication of complex numbers:

$$z_1 \cdot z_2 = (a_1 + ib_1) \cdot (a_2 + ib_2)$$

- In Euler notation:

$$\begin{aligned} z_1 \cdot z_2 &= |z_1|\, \mathrm{e}^{\mathrm{i}\varphi_1} \cdot |z_2|\, \mathrm{e}^{\mathrm{i}\varphi_2} \\ &= |z_1|\, |z_2|\, \mathrm{e}^{\mathrm{i}(\varphi_1 + \varphi_2)} \end{aligned}$$

- If $|z_2| = 1$, this is a rotation!



Rigid Registration

- Now we get:

$$p_{k,1} + \mathrm{i}p_{k,2} = \overset{\texttt{rotation}}{(r_1 + \mathrm{i}r_2)}\left(q_{k,1} + \mathrm{i}q_{k,2}\right) + \overset{\texttt{translation}}{t_1 + \mathrm{i}t_2}$$

  where $r_1^2 + r_2^2 = 1$

- This can be rewritten into two linear equations:

$$\begin{aligned} p_{k,1} &= r_1 q_{k,1} - r_2 q_{k,2} + t_1 = (q_{k,1}, -q_{k,2}, 1, 0)\begin{pmatrix} r_1 \\ r_2 \\ t_1 \\ t_2 \end{pmatrix} \\ p_{k,2} &= r_1 q_{k,2} + r_2 q_{k,1} + t_2 = (q_{k,2}, q_{k,1}, 0, 1)\begin{pmatrix} r_1 \\ r_2 \\ t_1 \\ t_2 \end{pmatrix} \end{aligned}$$

Rigid Registration

rotation

translation

- Now we get:

$$p_{k,1} + \mathrm{i}p_{k,2} = (r_1 + \mathrm{i}r_2)\,(q_{k,1} + \mathrm{i}q_{k,2}) + t_1 + \mathrm{i}t_2$$

where $r_1^2 + r_2^2 = 1$

- This can be rewritten into two linear equations:

$$p_{k,1} \quad = \quad r_1 q_{k,1} - r_2 q_{k,2} + t_1 = (q_{k,1}, -q_{k,2}, 1, 0) \begin{pmatrix} r_1 \\ r_2 \\ t_1 \\ t_2 \end{pmatrix}$$

real

$$p_{k,2} \quad = \quad r_1 q_{k,2} + r_2 q_{k,1} + t_2 = (q_{k,2}, q_{k,1}, 0, 1) \begin{pmatrix} r_1 \\ r_2 \\ t_1 \\ t_2 \end{pmatrix}$$

imag

Rigid Registration

- Putting it all together, we get:

$$\boldsymbol{A}\boldsymbol{x} = \begin{pmatrix} q_{1,1} & -q_{1,2} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ q_{N,1} & -q_{N,2} & 1 & 0 \\ q_{1,2} & q_{1,1} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ q_{N,2} & q_{N,1} & 0 & 1 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ t_1 \\ t_2 \end{pmatrix} = \begin{pmatrix} p_{1,1} \\ \vdots \\ p_{N,1} \\ p_{1,2} \\ \vdots \\ p_{N,2} \end{pmatrix} = \boldsymbol{b}$$

Rigid Registration

- Democratic algorithm: Rotation and translation are estimated simultaneously
- Linear problem! How do we solve it?
- The solution needs to be scaled so that $r_1^2 + r_2^2 = 1$

Rigid Registration

- Democratic algorithm: Rotation and translation are estimated simultaneously
- Linear problem! How do we solve it? Pseudoinverse (SVD)
- The solution needs to be scaled so that $r_1^2 + r_2^2 = 1$

Rigid Registration – Exercise

- Find the solution to the following problem:

q	(1, 2)	(3, 6)	(4, 6)	(3, 4)
p	(-0.7, 2.1)	(-2.1, 6.4)	(-1.4, 7.1)	(-0.7, 4.9)

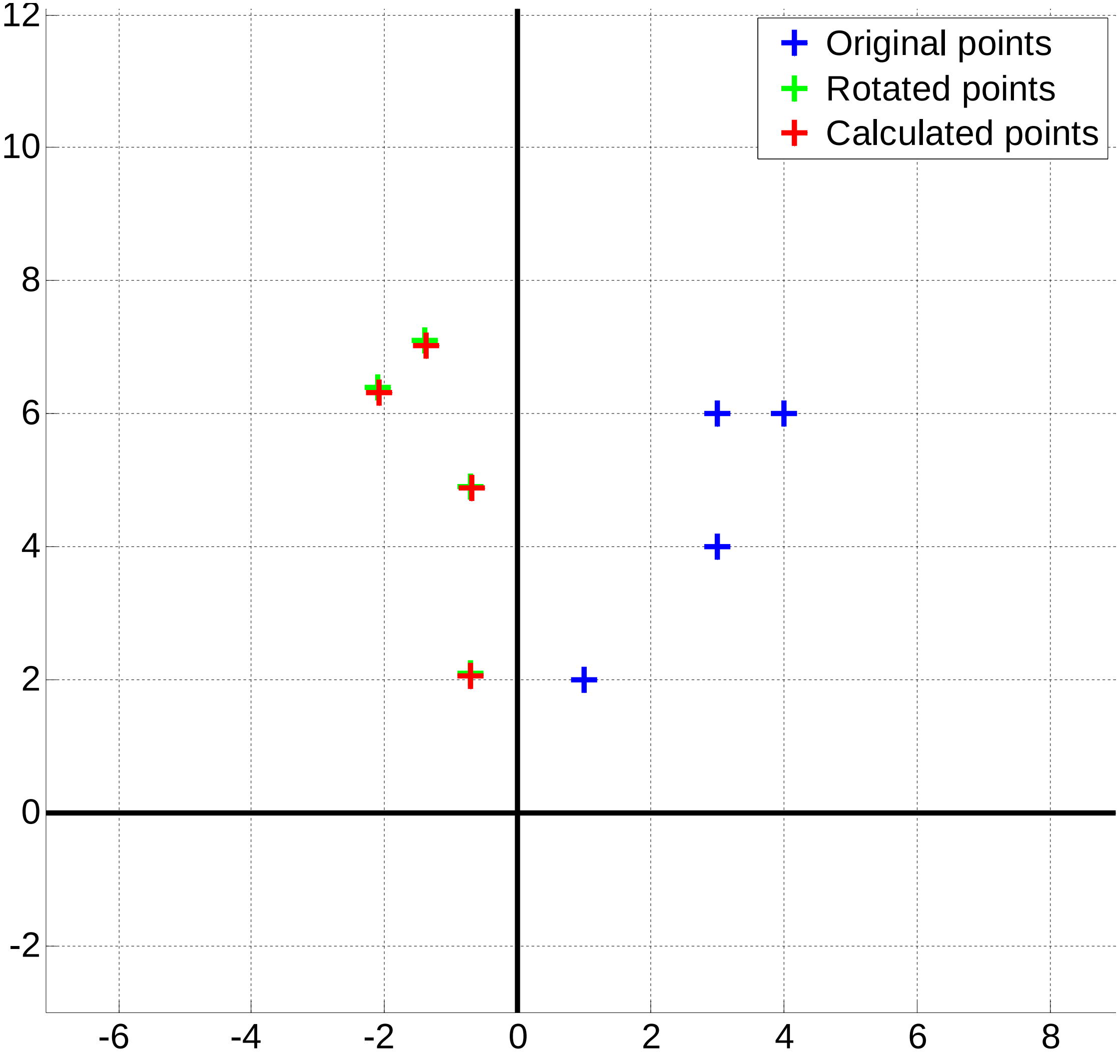
- What is the rotation angle?

Rigid Registration – Exercise

```
% Point correspondences
qk = [1 2;
       3 6;
       4 6;
       3 4];

pk = [-0.7 2.1;
     -2.1 6.4;
     -1.4 7.1;
     -0.7 4.9];
```

Rigid Registration – Exercise



Rigid Registration – Intensity Based

- Point based:

$$\arg\min_{\varphi,\boldsymbol{t}} \sum_{k=1}^{N} \left\|\boldsymbol{p}_k - \boldsymbol{R}\boldsymbol{q}_k - \boldsymbol{t}\right\|^2$$

- Intensity based:

$$\arg\min_{\varphi,\boldsymbol{t}} D\left(\boldsymbol{S}\left(\boldsymbol{x}\right), \boldsymbol{M}\left(\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}\right)\right)$$

where D is a similarity or distance function, $\boldsymbol{S}$ is the static or reference image, $\boldsymbol{M}$ is the moving or template image and $\boldsymbol{x}$ is a point in the image.

Rigid Registration – Similarity Metrics

- Sum of Squared Distances (SSD)

$$SSD\left(\boldsymbol{I_1}\left(\boldsymbol{x}\right),\boldsymbol{I_2}\left(\boldsymbol{x}\right)\right)=\sum_{i=1}^{N}\left(\boldsymbol{I_1}\left(\boldsymbol{x}_i\right)-\boldsymbol{I_2}\left(\boldsymbol{x}_i\right)\right)^2$$

  - Direct intensity comparison: Same intensities shown?
  - Simple and fast
  - Requires same intensity ranges for same tissues

Rigid Registration - Implementation

- Task: Generate a Shepp-Logan phantom. Use `imtransform` to apply a rotation of 45 ° and translation of `t = [20, 1]`.

Rigid Registration - Implementation

- Task: Generate a Shepp-Logan phantom. Use `imtransform` to apply a rotation of 45 ° and translation of `t = [20, 1]`.

```
image = …;
…
a = pi/4;
tmat = …;
tform = maketform('affine', tmat);
transimage2 = imtransform(image, tform);
```

Rigid Registration - Implementation

- Task: Now we want to start with the registration. Filter the image to avoid local minima during optimization and transform it using `transform()`, `rotation`, `transx`, and `transy` to obtain a moving image.

Rigid Registration - Implementation

- Task: Now we want to start with the registration. Filter the image to avoid local minima during optimization and transform it using `transform`, `rotation`, `transx`, and `transy` to obtain a moving image.

```
Image1=padarray(mat2gray(phantom(is)),[paddings,paddings]);
size(Image1)
nice = fspecial('gaussian', 10, 4);
Image1 = …;
Image2 = …;
```

Rigid Registration - Implementation

- Task: Minimizing the similarity function `sim` is a non-linear optimization problem. Use `fminsearch` for this task allowing a maximum number of 150 similarity function evaluations.

Rigid Registration - Implementation

- Task: Minimizing the similarity function `sim` is a non-linear optimization problem. Use `fminsearch` for this task allowing a maximum number of 150 similarity function evaluations.

```
[x, fval] = fminsearch(…);
```

Rigid Registration - Implementation

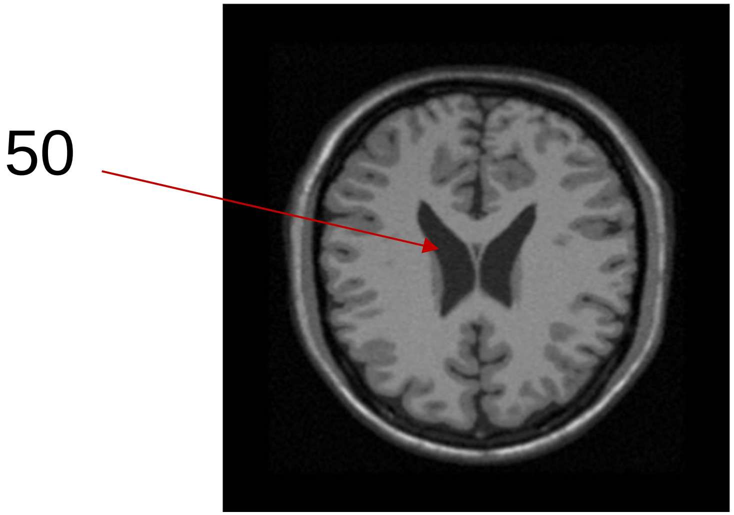
- Task: Implement the Sum of Squared Distances (SSD).

$$SSD\left(\boldsymbol{I}_{\mathbf{1}}\left(\boldsymbol{x}\right),\boldsymbol{I}_{\mathbf{2}}\left(\boldsymbol{x}\right)\right)=\sum_{i=1}^{N}\left(\boldsymbol{I}_{\mathbf{1}}\left(\boldsymbol{x}_{i}\right)-\boldsymbol{I}_{\mathbf{2}}\left(\boldsymbol{x}_{i}\right)\right)^{2}$$

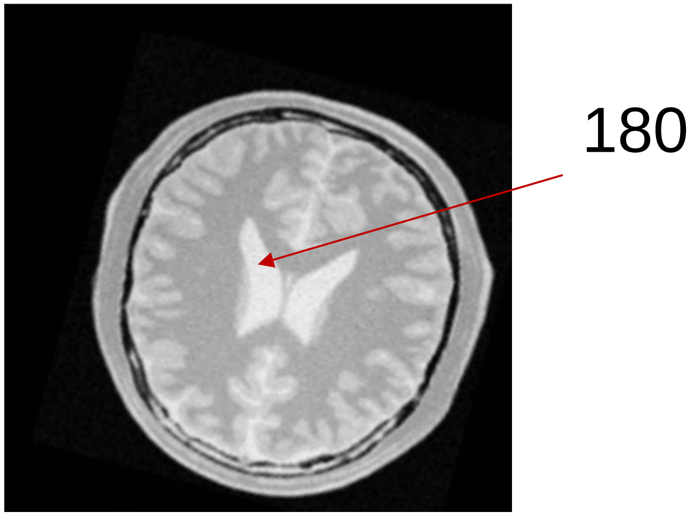
```
% distance measure SSD
???
```

Rigid Registration – Similarity Metrics

- How about for images that have the different intensity ranges?



T1 MR



Proton Density MR

Rigid Registration – Similarity Metrics

- Mutual Information (MI) – a measure of mutual dependence of images

$$MI\left(\boldsymbol{I_1}\left(\boldsymbol{x}\right),\boldsymbol{I_2}\left(\boldsymbol{y}\right)\right) = H\left(\boldsymbol{I_1}\right) + H\left(\boldsymbol{I_2}\right) - H\left(\boldsymbol{I_1},\boldsymbol{I_2}\right)$$
$$= H\left(\boldsymbol{I_1},\boldsymbol{I_2}\right) - H\left(\boldsymbol{I_1}|\boldsymbol{I_2}\right) - H\left(\boldsymbol{I_2}|\boldsymbol{I_1}\right)$$

where $H\left(\cdot\right)$ is the marginal entropy, $H\left(\cdot,\cdot\right)$ is the joint entropy, $H\left(\cdot|\cdot\right)$and is the conditional entropy

- Comparison of intensity distribution: Same information shared?
- Complex computation
- More robust, if intensities for same tissues differ between images

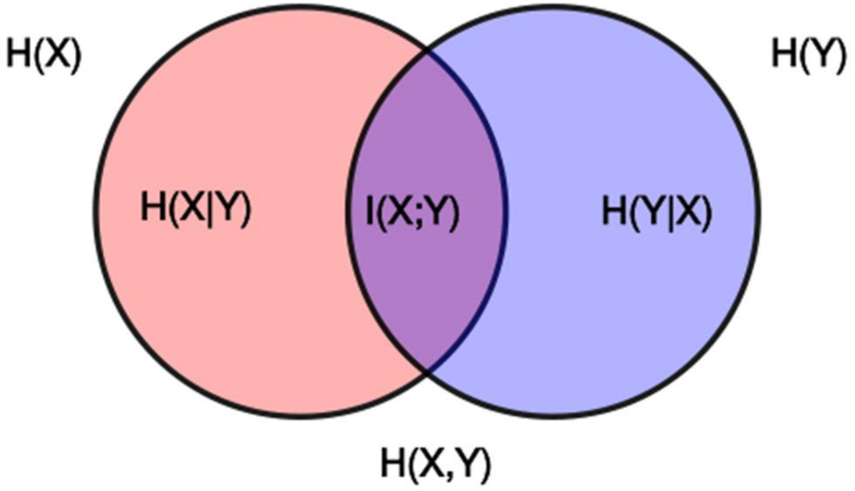


Image: Wikipedia

Rigid Registration – Similarity Metrics

- Mutual Information (MI) – a measure of mutual dependence of images

$$
\begin{aligned}
MI\left(\boldsymbol{I_1}\left(\boldsymbol{x}\right), \boldsymbol{I_2}\left(\boldsymbol{y}\right)\right) &= H\left(\boldsymbol{I_1}\right) + H\left(\boldsymbol{I_2}\right) - H\left(\boldsymbol{I_1}, \boldsymbol{I_2}\right) \\
&= \boxed{H\left(\boldsymbol{I_1}, \boldsymbol{I_2}\right)} - H\left(\boldsymbol{I_1}|\boldsymbol{I_2}\right) - H\left(\boldsymbol{I_2}|\boldsymbol{I_1}\right)
\end{aligned}
$$

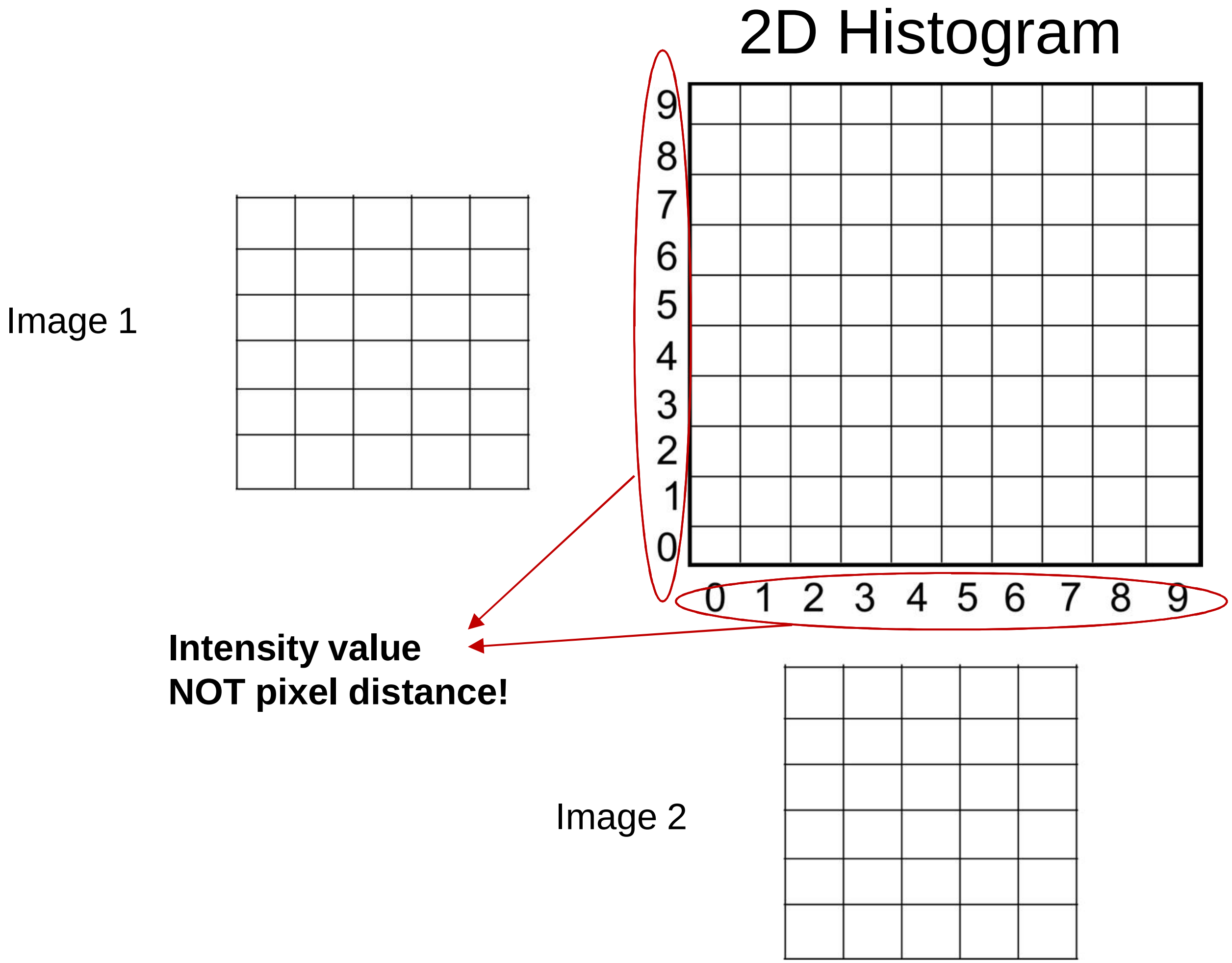
where $H\left(\cdot\right)$ is the marginal entropy, $H\left(\cdot,\cdot\right)$ is the joint entropy, $H\left(\cdot|\cdot\right)$and is the conditional entropy

$$H\left(\boldsymbol{I}_1, \boldsymbol{I}_2\right) = -\sum\sum P\left(x, y\right)\log_2\left[P\left(x, y\right)\right]$$

where $P\left(\cdot,\cdot\right)$ is the probability of the values occurring together

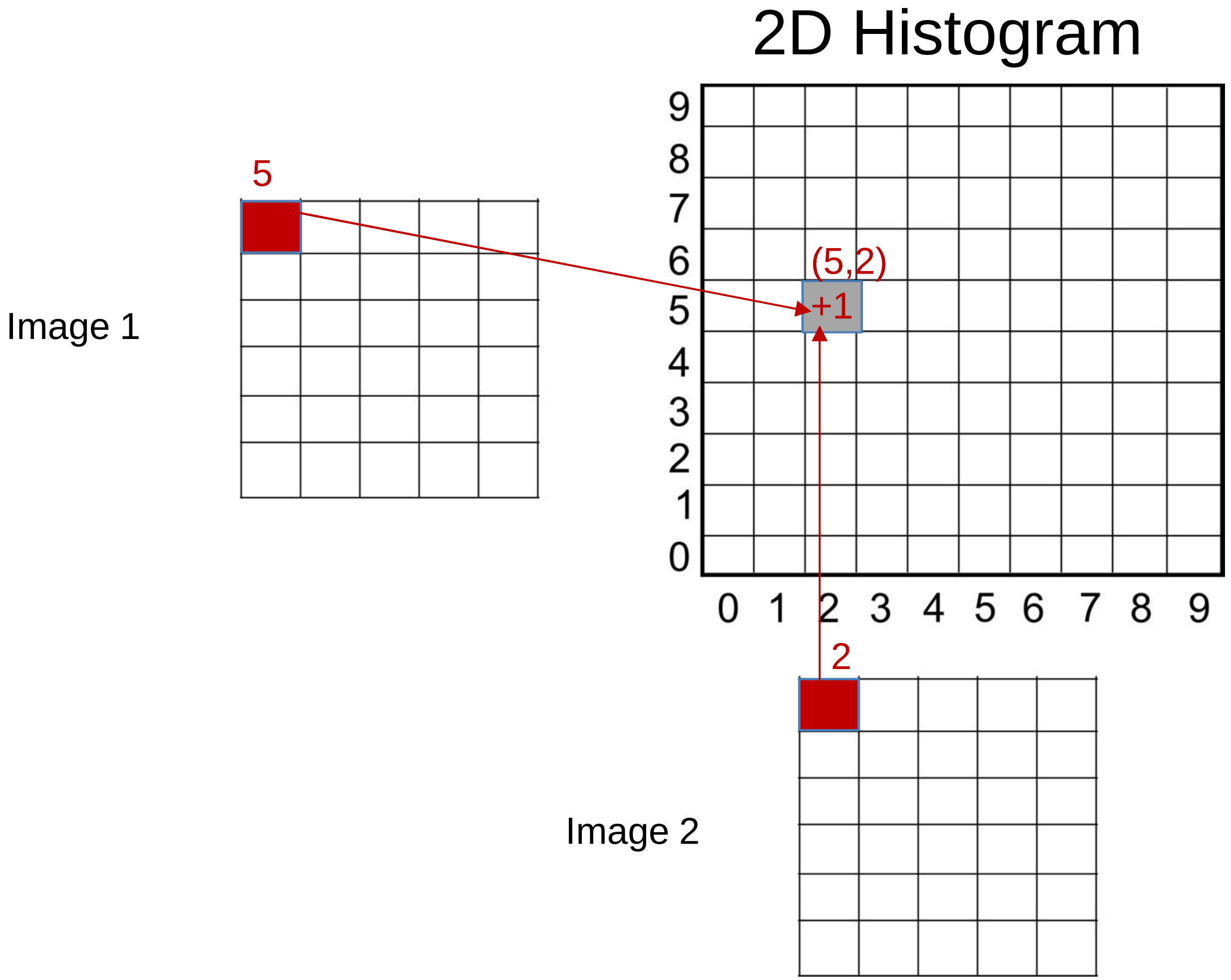
→ Joint histogram

Rigid Registration – Similarity Metrics

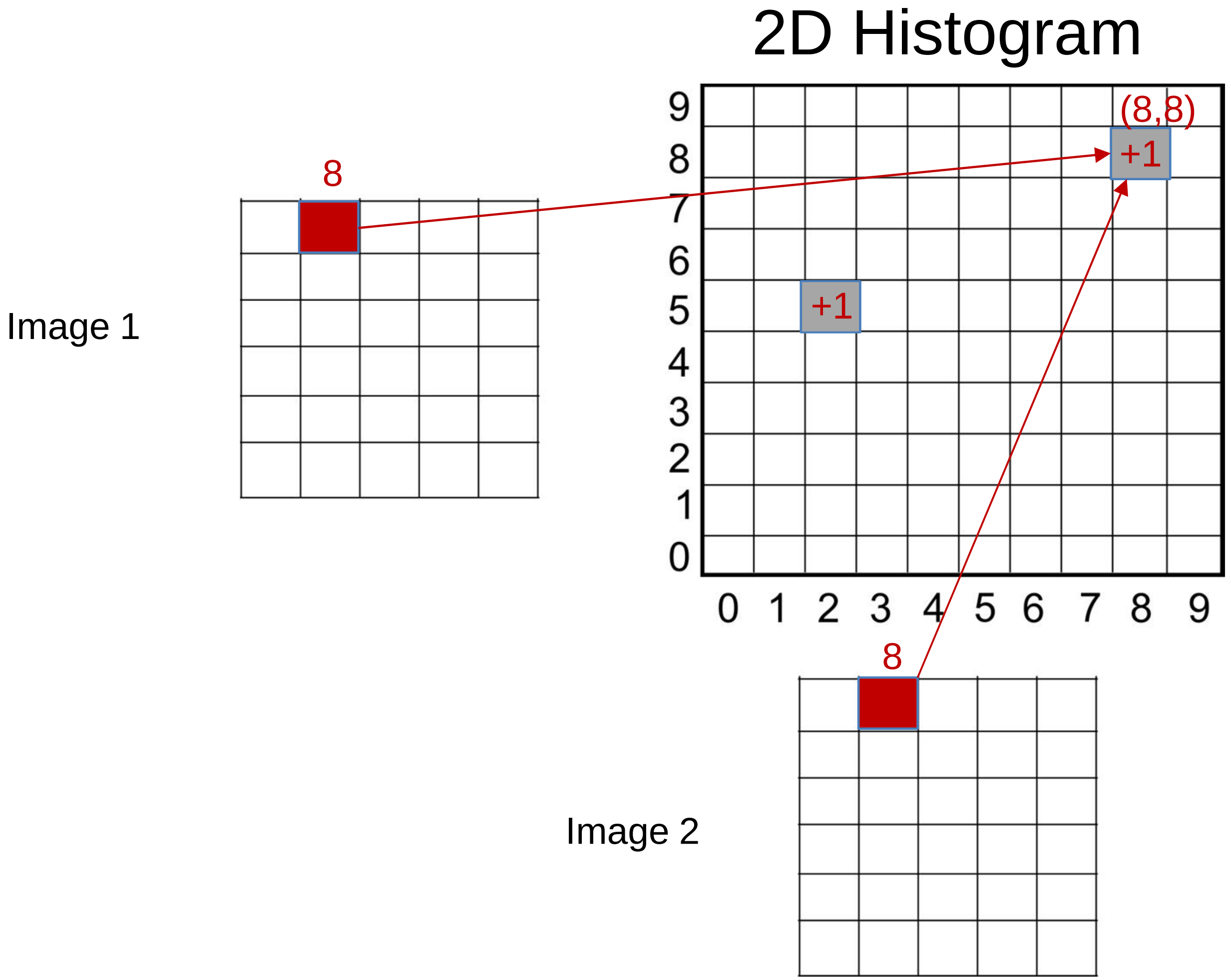


Intensity value
NOT pixel distance!

Rigid Registration – Similarity Metrics

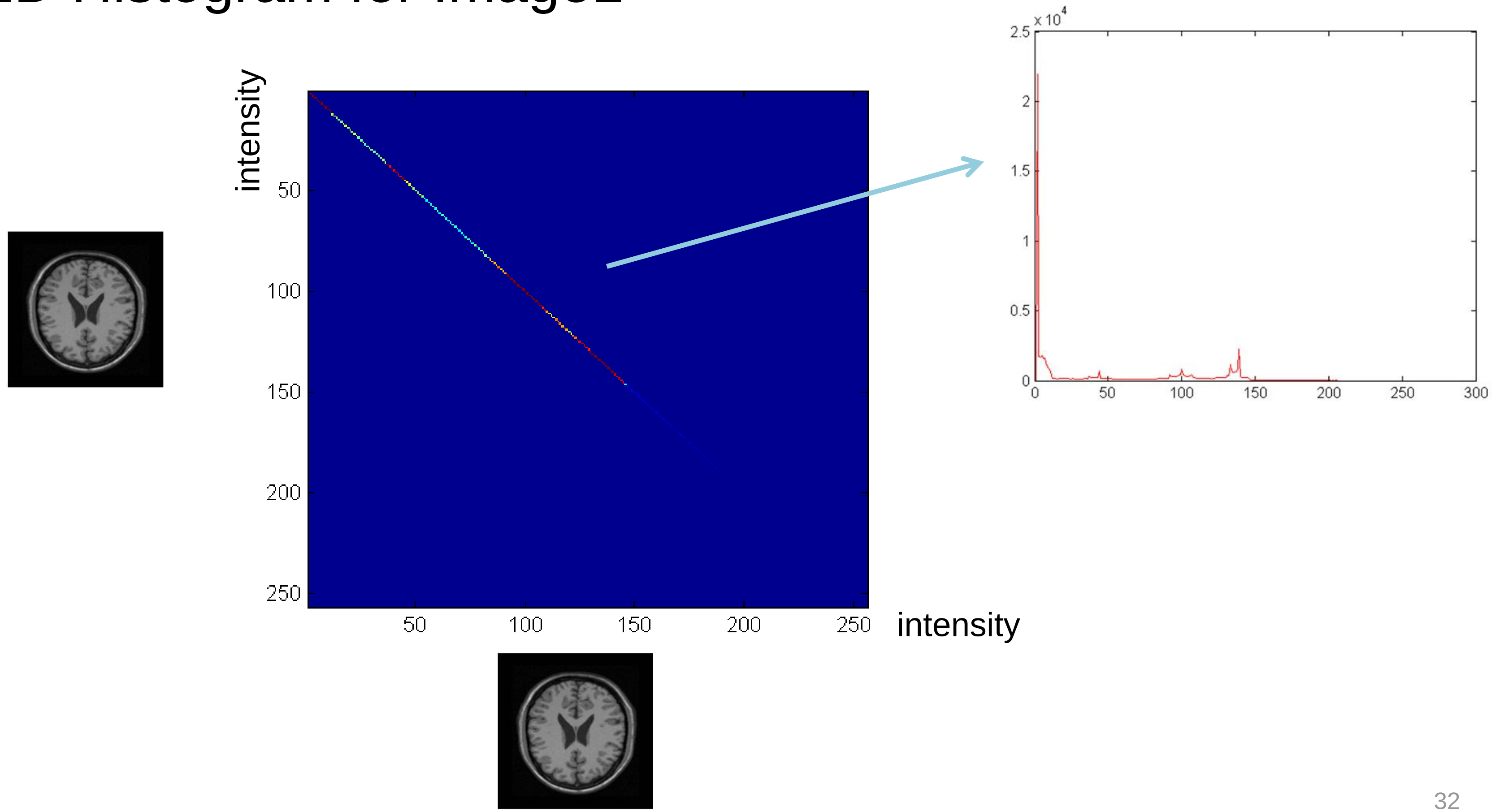


Rigid Registration – Similarity Metrics



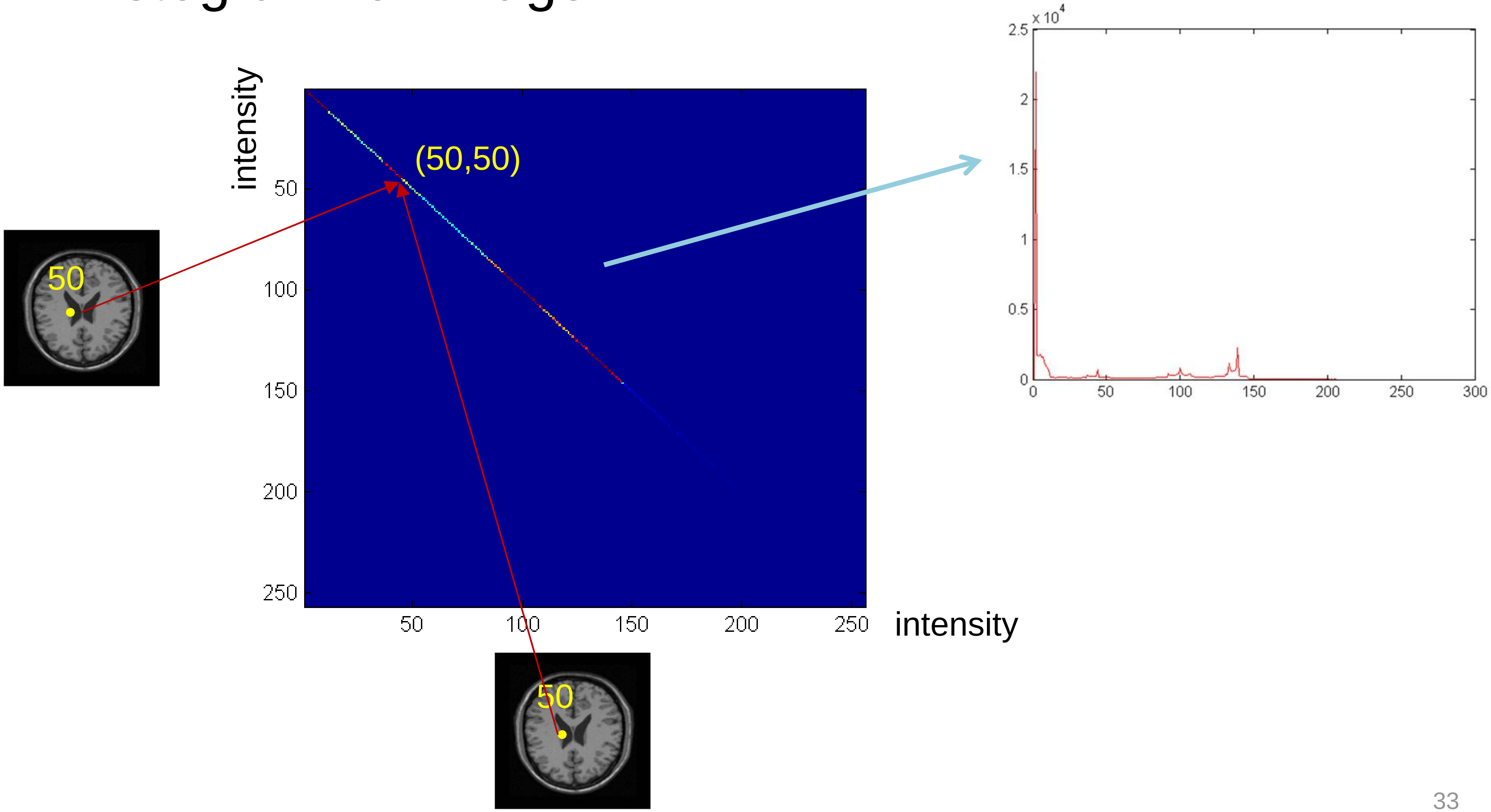
Rigid Registration – Similarity Metrics

2D Histogram for Image1



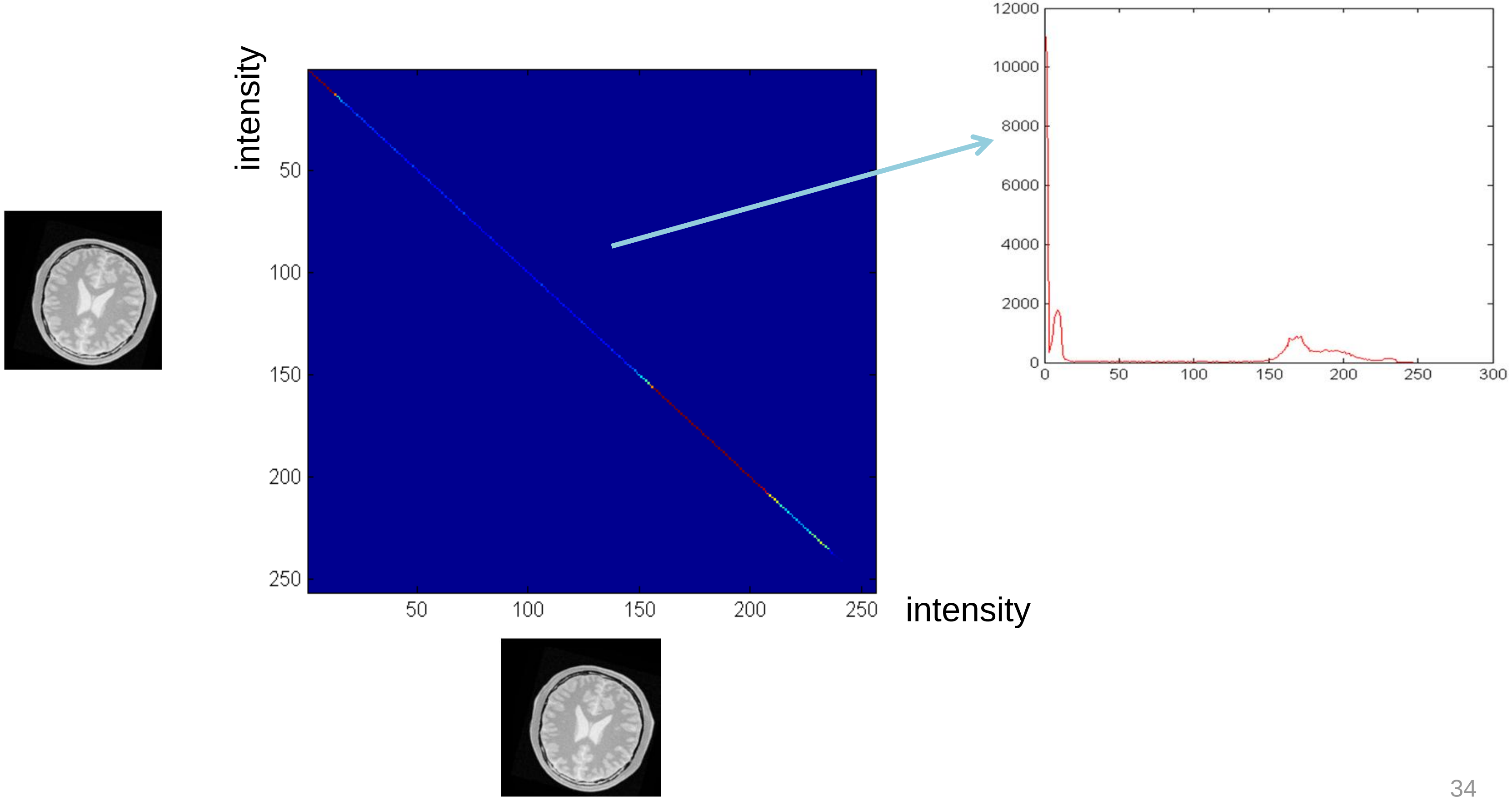
Rigid Registration – Similarity Metrics

2D Histogram for Image1



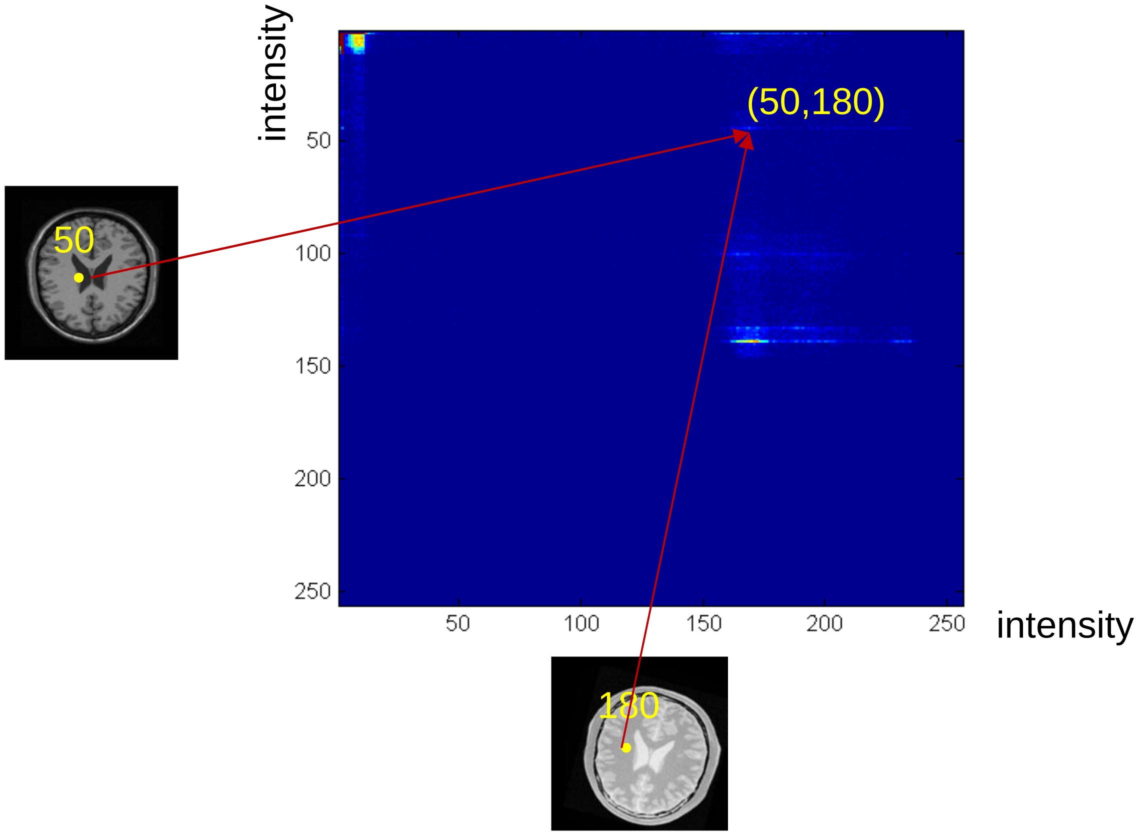
Rigid Registration – Similarity Metrics

2D Histogram for Image2



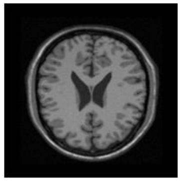
Rigid Registration – Similarity Metrics

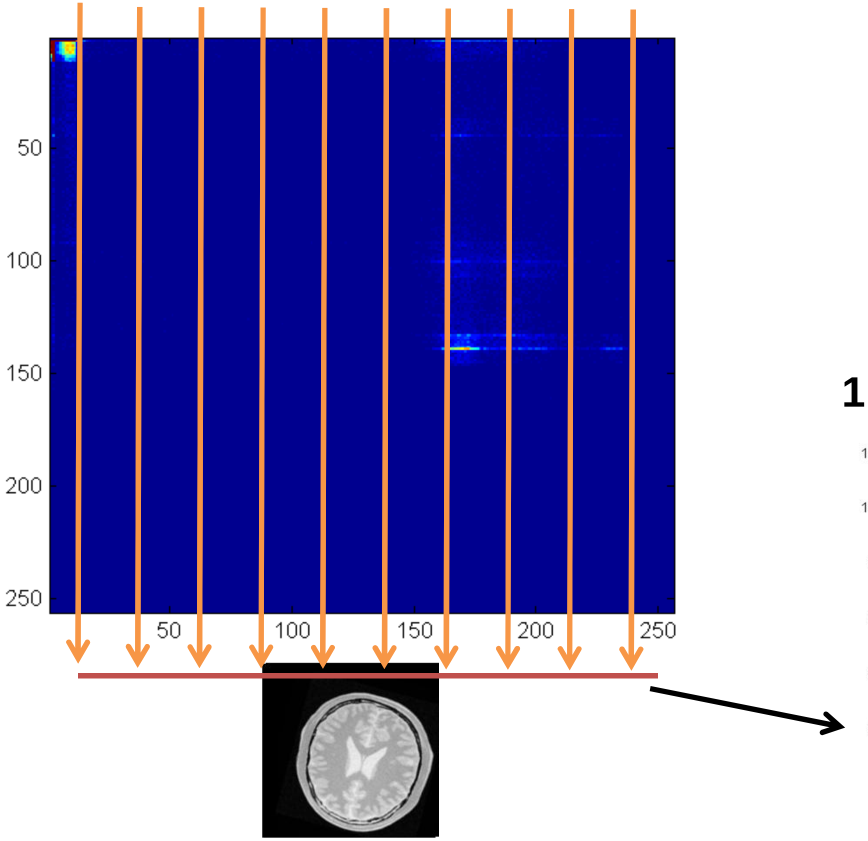
2D Joint Histogram for both



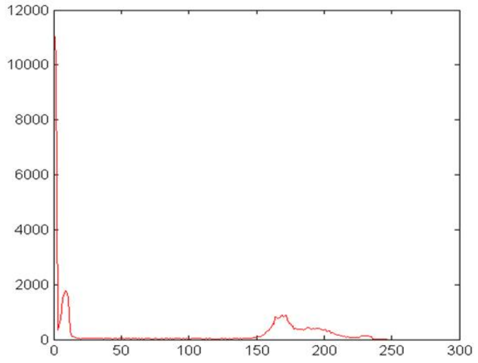
Rigid Registration – Similarity Metrics

2D Joint Histogram for both



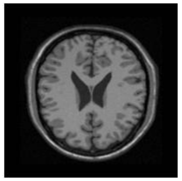


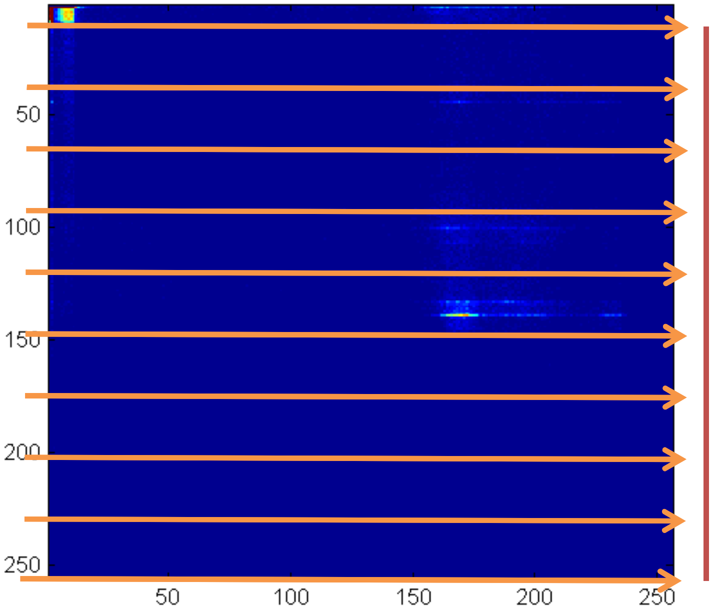
1D Histogram for Image 2

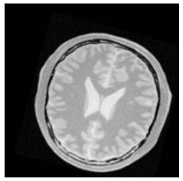


Rigid Registration – Similarity Metrics

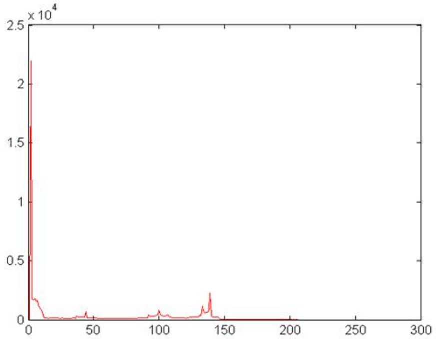
2D Joint Histogram for both







1D Histogram for Image 1



Rigid Registration - Implementation

- Task: Implement the helper function `jointH` to obtain a joint histogram.

```
function h = jointH(im1, im2)
   is = size(Image1);
   h = zeros(256);
   ???
end
```

Rigid Registration – Similarity Metrics

- Mutual Information (MI) – a measure of mutual dependence of images

$$MI\left(\boldsymbol{I_1}\left(\boldsymbol{x}\right), \boldsymbol{I_2}\left(\boldsymbol{y}\right)\right) = H\left(\boldsymbol{I_1}\right) + H\left(\boldsymbol{I_2}\right) - H\left(\boldsymbol{I_1}, \boldsymbol{I_2}\right)$$

$$= H\left(\boldsymbol{I_1}, \boldsymbol{I_2}\right) - H\left(\boldsymbol{I_1}|\boldsymbol{I_2}\right) - H\left(\boldsymbol{I_2}|\boldsymbol{I_1}\right)$$

where $H\left(\cdot\right)$ is the marginal entropy, $H\left(\cdot,\cdot\right)$ is the joint entropy, $H\left(\cdot|\cdot\right)$and is the conditional entropy

$$H\left(\boldsymbol{I}_1\right) = -\sum P\left(x_i\right)\log_2\left[P\left(x_i\right)\right]$$

→ $P\left(\cdot\right)$ is the histogram of $\boldsymbol{I_1}$

Rigid Registration - Implementation

- Task: Implement the helper function `marginalE` to obtain a joint histogram. Note: `marginalE` expects a joint histogram as input.

$$H\left(\boldsymbol{I}_1\right) = -\sum P\left(x_i\right)\log_2\left[P\left(x_i\right)\right]$$

```
function h = marginalE(hxy)
    marginal = sum(hxy);
    h=0;
    ???
end
```

Rigid Registration - Implementation

- Task: Calculate the joint histogram of both images. Use it to obtain the joint entropy. Finally, implement the Mutual Information (MI). Note: MI is maximum for best match, but we require a minimization problem.

$$MI\left(\boldsymbol{I_1}\left(\boldsymbol{x}\right),\boldsymbol{I_2}\left(\boldsymbol{x}\right)\right)=H\left(\boldsymbol{I_1}\right)+H\left(\boldsymbol{I_2}\right)-H\left(\boldsymbol{I_1},\boldsymbol{I_2}\right)$$

$$H\left(\boldsymbol{I}_1,\boldsymbol{I}_2\right)=-\sum\sum P\left(x,y\right)\log_2\left[P\left(x,y\right)\right]$$

```
 % joint histogram of fiximage and moved
???
% joint entropy of the joint histogram of fiximage and
% moved
???
% mutual information formula (inverted)
???
```