Registration. Lecture 1

Bulat Ibragimov

bulat@di.ku.dk

Department of Computer Science
University of Copenhagen

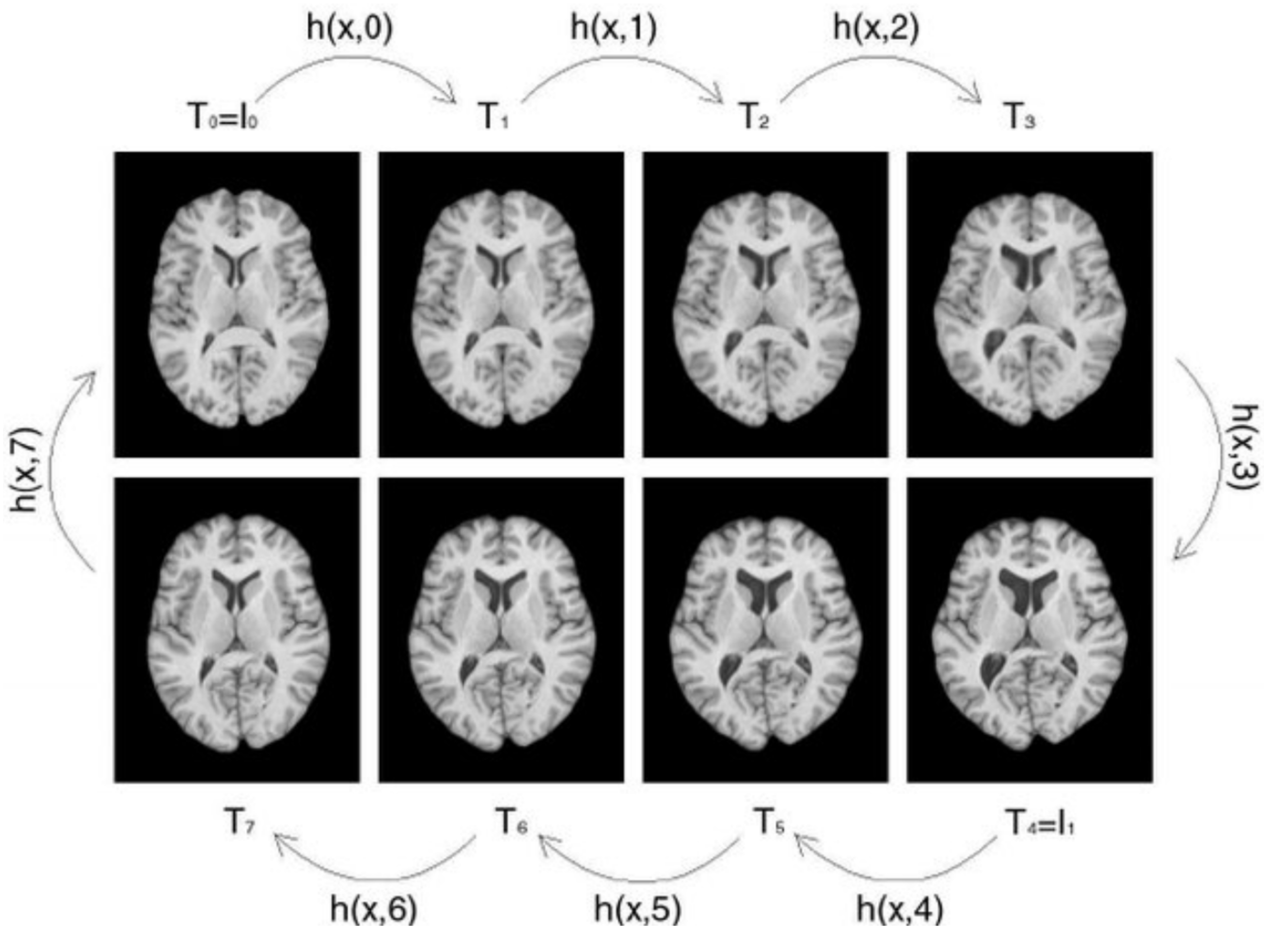
UNIVERSITY OF COPENHAGEN

Todays Learning Objectives

- Why do we need registration?
- Similarity measures
- Rigid registration

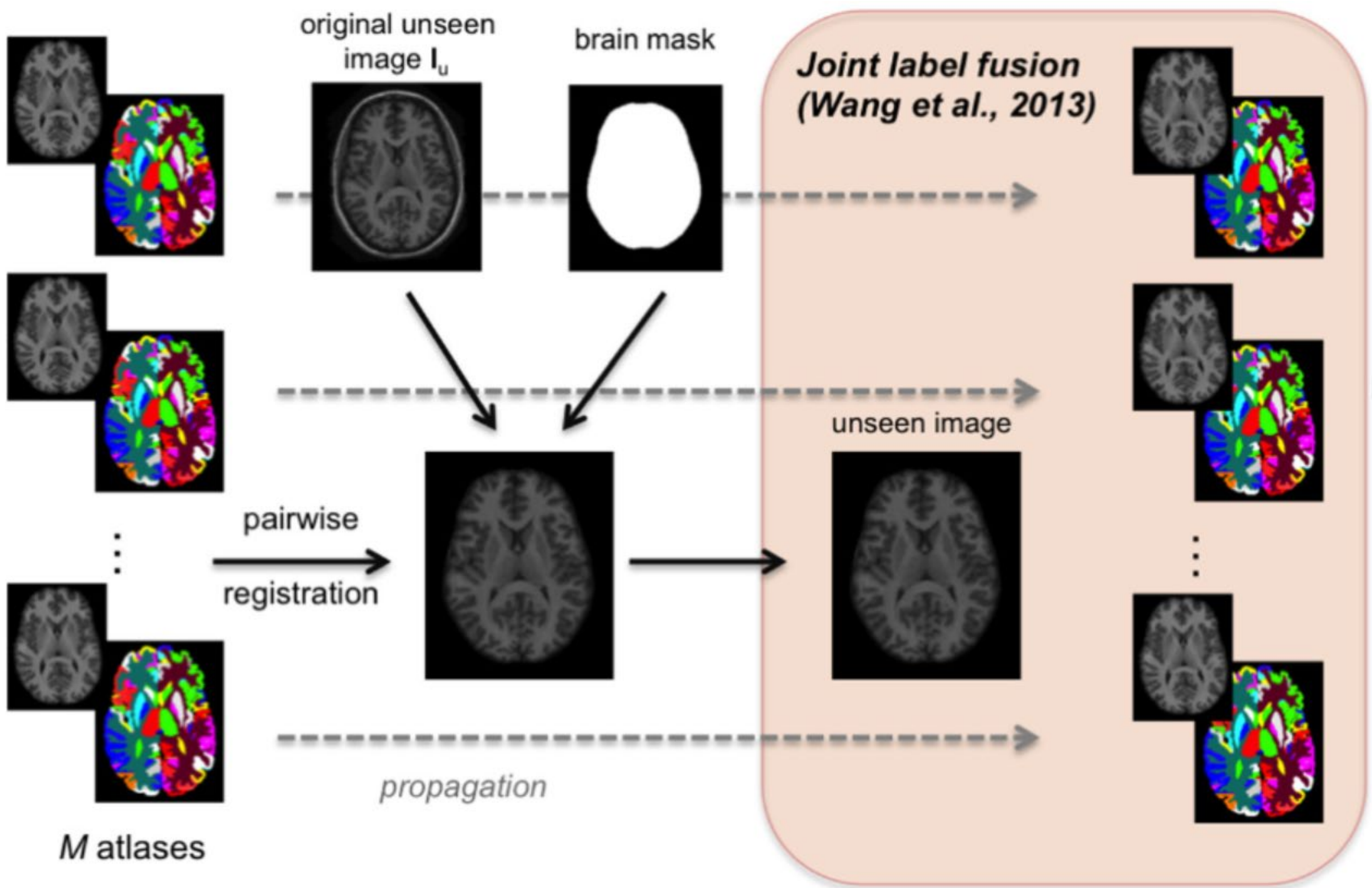
Why do we need registration?

Registration geometrically transforms one image into another



Why do we need registration?

Atlas-based segmentation



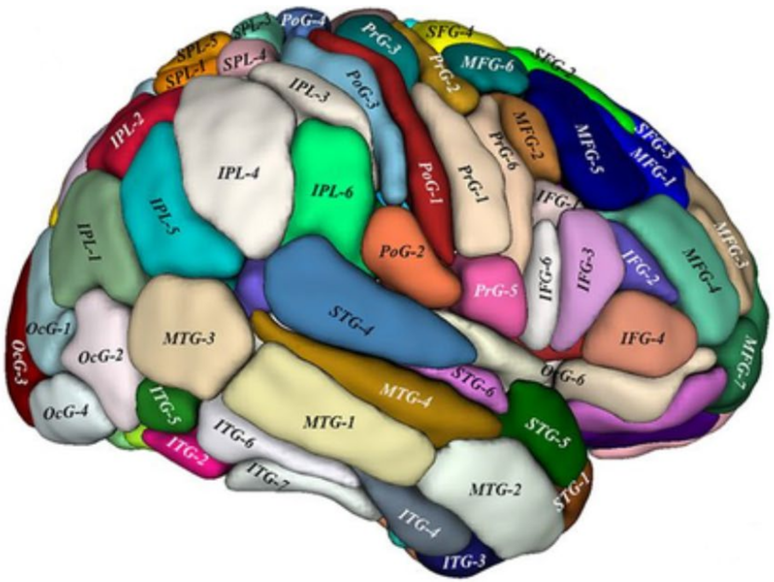
*Robust whole-brain segmentation: Application to traumatic brain injury

Why do we need registration?

Atlas-based segmentation:

- (-) Segmentation speed linearly depends on the number of training images
- (+) Segmentation speed does not depend on the number of target structures
- (+) Needs way less training samples than deep learning

More examples at the end



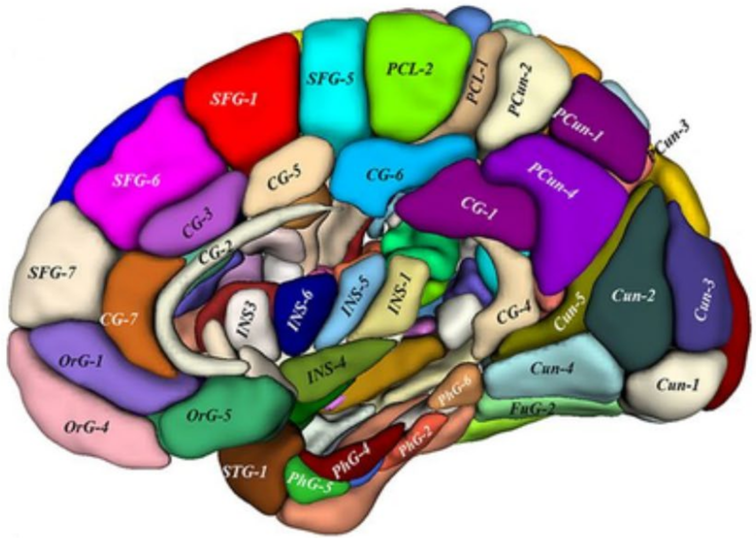


Image registration

Information type

- Intrinsic or extrinsic

Similarity measure

- Pixelwise difference, correlation, mutual information

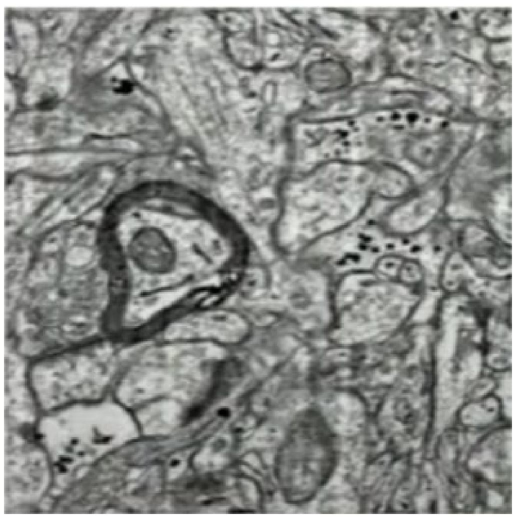
Transformation

- Rigid, non-rigid

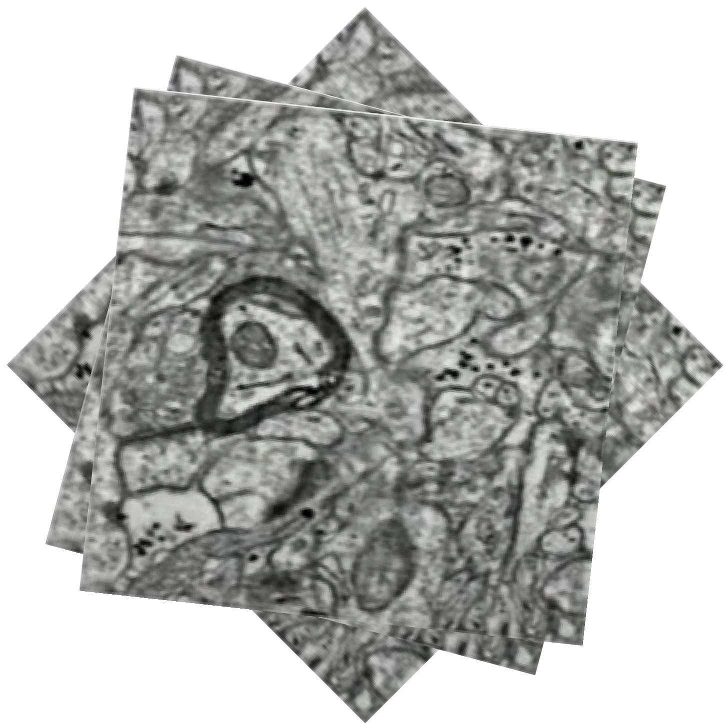
Image registration: information type

Intrinsic information

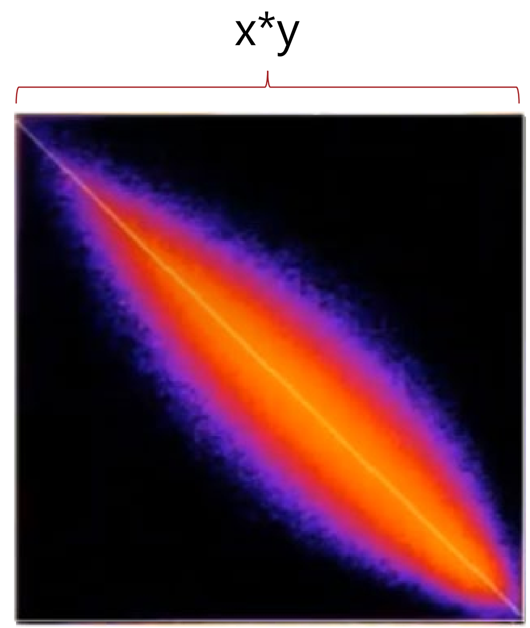
- The images are visually similar (non-necessarily by absolute intensities):



Target image



Moving image



Histogram matching

*WMIS

Image registration: information type

Extrinsic information

- The images are too different from each other visually
- We need to help registration by providing correspondences

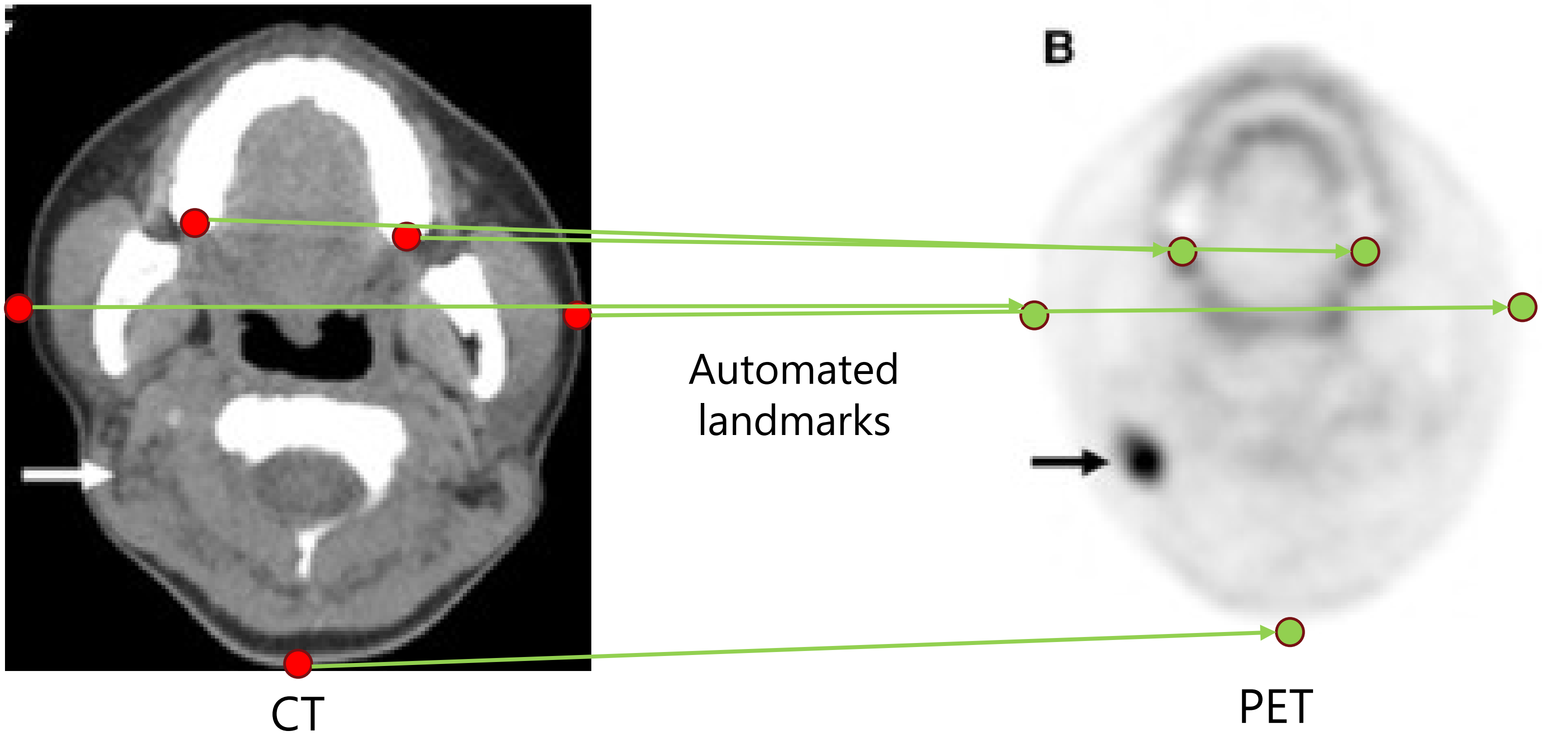
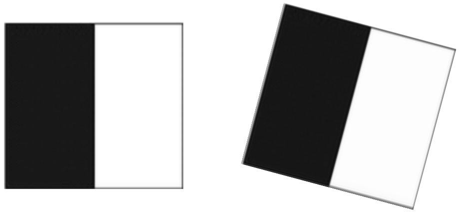


Image registration: similarity measures

Mean sum of squared differences:

$$MSE = \frac{1}{n}\frac{1}{m}\sum_{x=1}^{n}\sum_{y=1}^{m}\left(I(x,y) - J(x,y)\right)^2$$



$$MSE\left(\begin{matrix} 2 & 2 & 4 & 4 \\ 2 & 2 & 4 & 4 \\ 2 & 2 & 4 & 4 \\ 2 & 2 & 4 & 4 \end{matrix}, \begin{matrix} 2 & 2 & 2 & 4 \\ 2 & 2 & 4 & 4 \\ 2 & 2 & 4 & 4 \\ 2 & 4 & 4 & 4 \end{matrix}\right) =$$

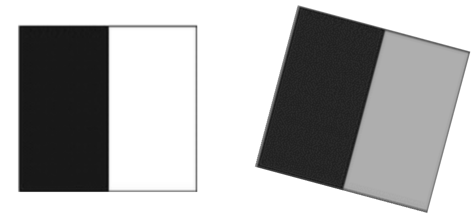
$$\frac{1}{4}\frac{1}{4}(0 + \cdots + (4-2)^2 + (2-4)^2 + \cdots + 0) = 0.5$$

MINIMIZATION

Image registration: similarity measures

Normalized sum of squared differences:

$$I^* = \frac{I - \bar{I}}{\sigma(I)}$$



1	1	1	3
1	1	3	3
1	1	3	3
1	3	3	3

$\bar{I} = 2$
$\sigma = 1$

-1	-1	-1	1
-1	-1	1	1
-1	-1	1	1
-1	1	1	1

MSE = 1.5

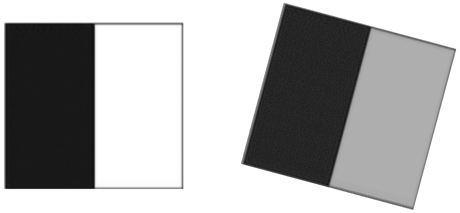
MINIMIZATION

NMSE = 0.5

Image registration: similarity measures

Normalized cross-correlation:

$$NCC = \frac{\sum_{x,y}((I(x,y) - \bar{I}) \cdot (J(x,y) - \bar{J}))}{\sqrt{\sum_{x,y}(I(x,y) - \bar{I})^2 \sum_{x,y}(J(x,y) - \bar{J})^2}}$$



$$NCC = \frac{12}{\sqrt{16 \cdot 16}} = 0.75$$

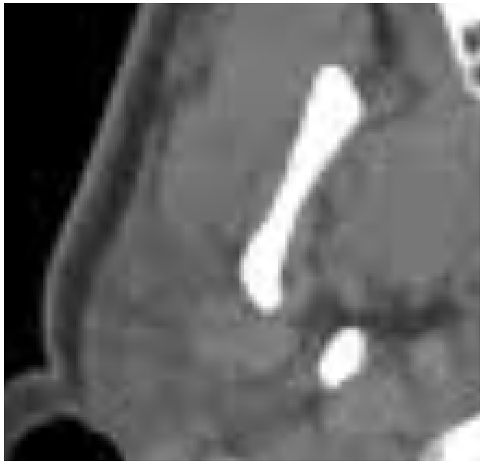
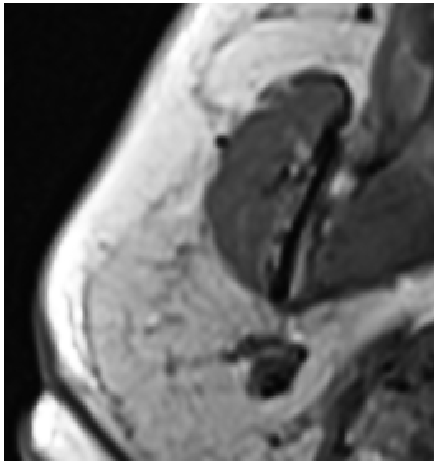
2	2	4	4
2	2	4	4
2	2	4	4
2	2	4	4

1	1	1	3
1	1	3	3
1	1	3	3
1	3	3	3

MAXIMIZATION

Image registration: similarity measures

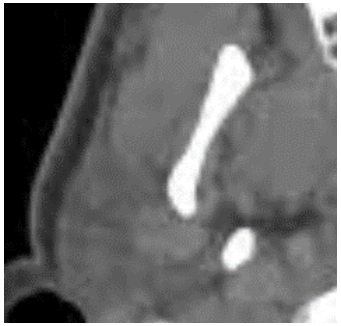
Will MSE and NCC work for CT-MR image registration?

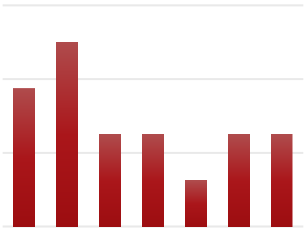



MSE and NCC try to match high intensity with high intensity, and low intensity with low intensity.

But we want to match patterns not individual intensities.

Image registration: similarity measures





Mutual image information:

$$MI = E(I,J)$$

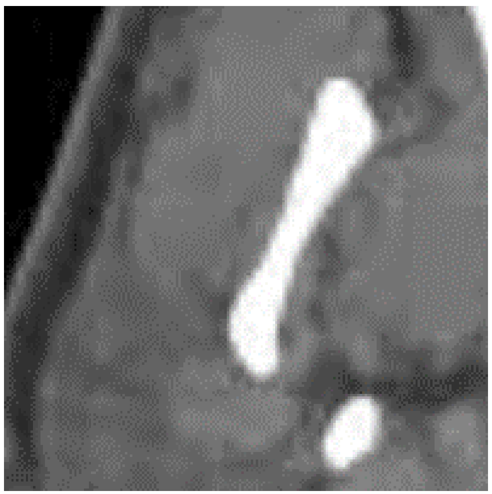
Joint entropy

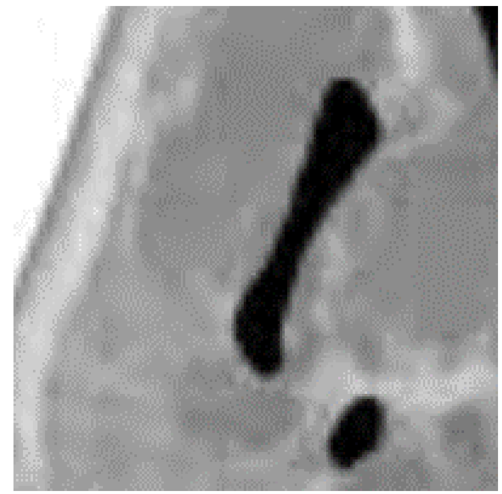
$$E(I,J) = -\sum_i \sum_j P(i,j) \log_2 \frac{P(i,j)}{P(i) \cdot P(j)}$$

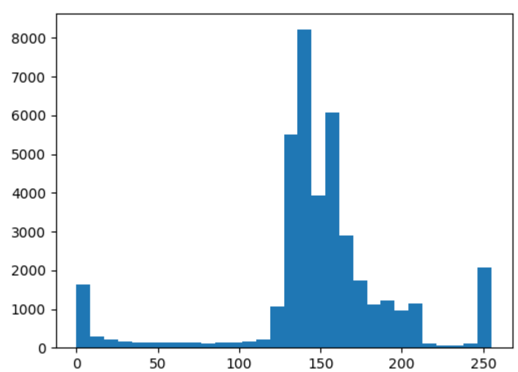
Image registration: similarity measures

Mutual image information:

$$MI = E(I,J)$$







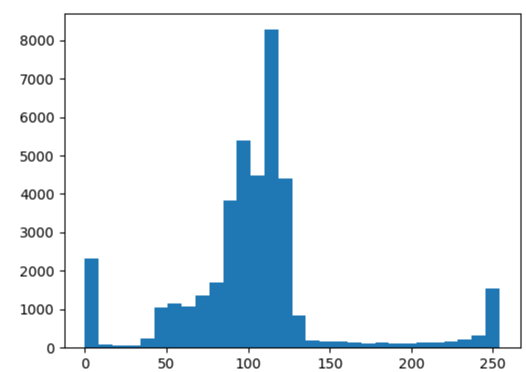
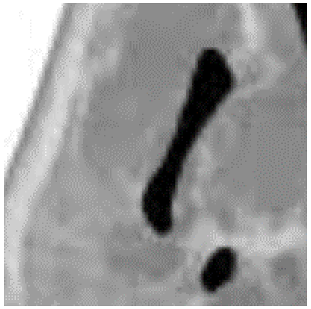
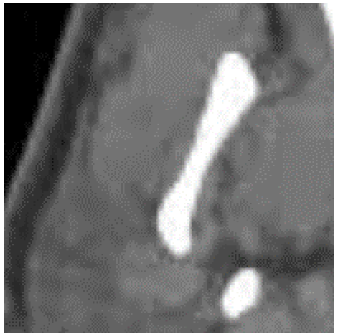


Image registration: similarity measures

Mutual image information:



Every pixel is a dot with coordinates defined by colors in two images



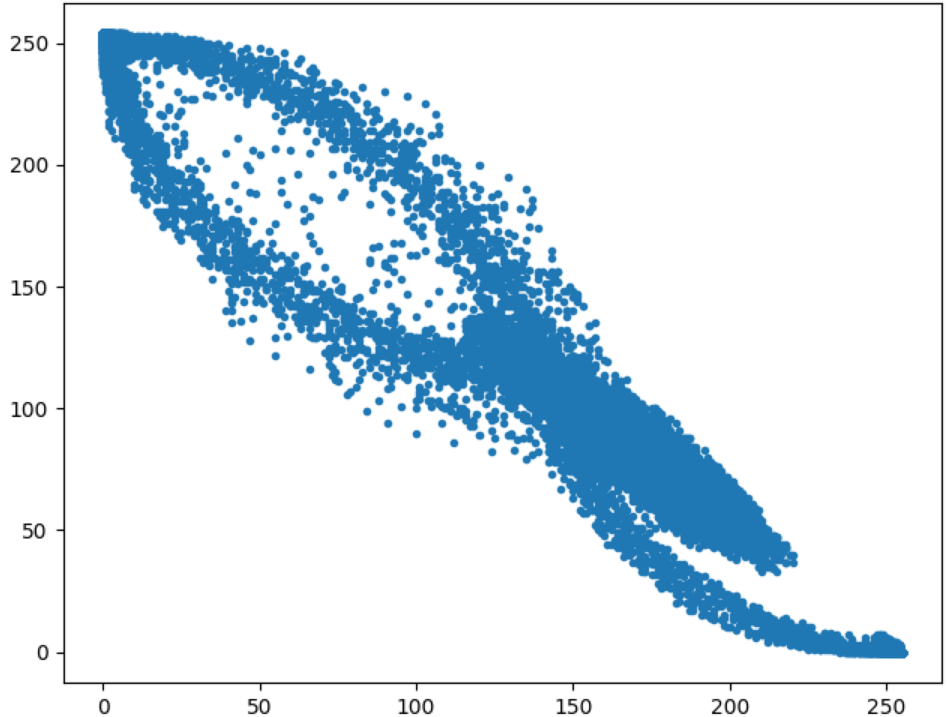


Image registration: similarity measures

Mutual image information:

$$E(I,J) = -\sum_{i\in I}\sum_{j\in J} P(i,j)\log_2\frac{P(i,j)}{P(i)\cdot P(j)}$$

$$P(i) = \sum_{j} P(j,i)\sum_{j} P(i,j)$$

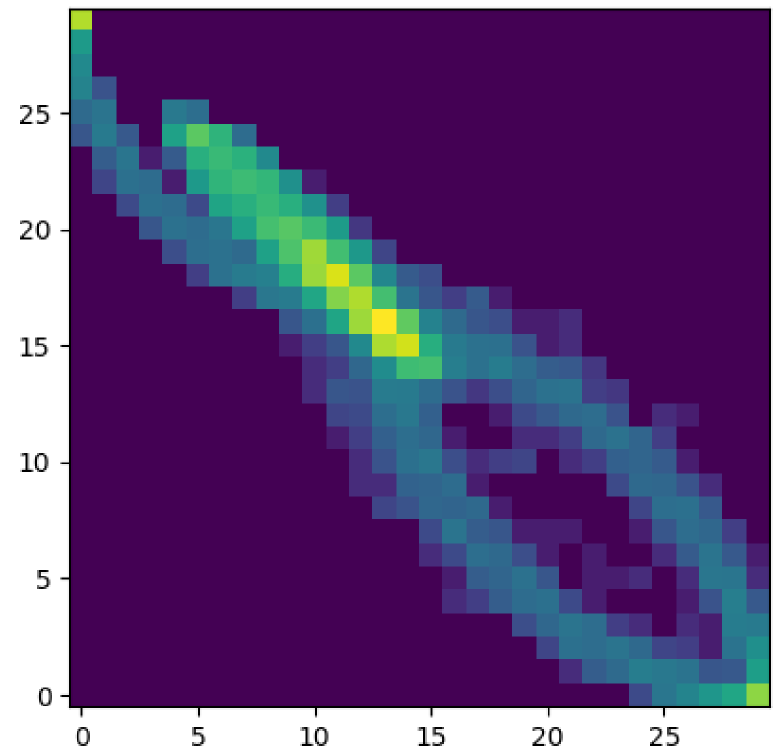
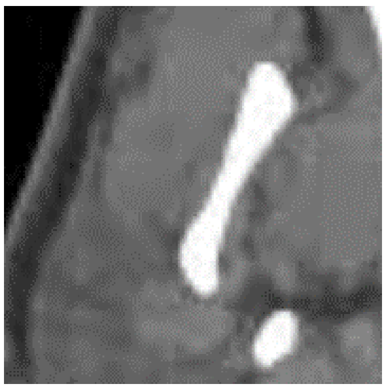
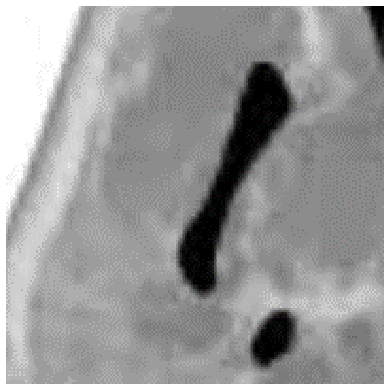
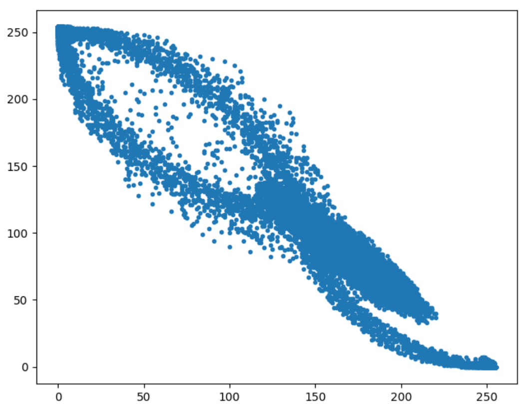


Image registration: similarity measures

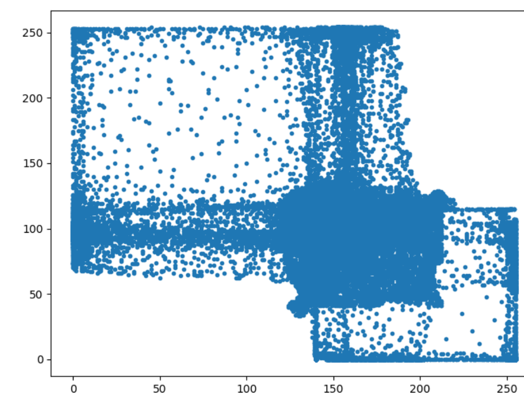
How does mutual information behaves?

$MI = 1.15$

$MI = 0.23$

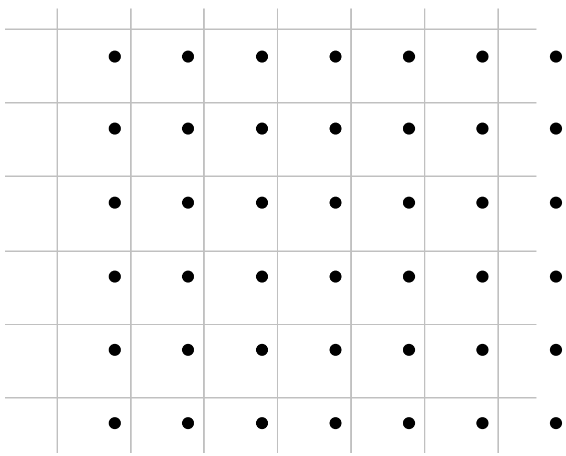




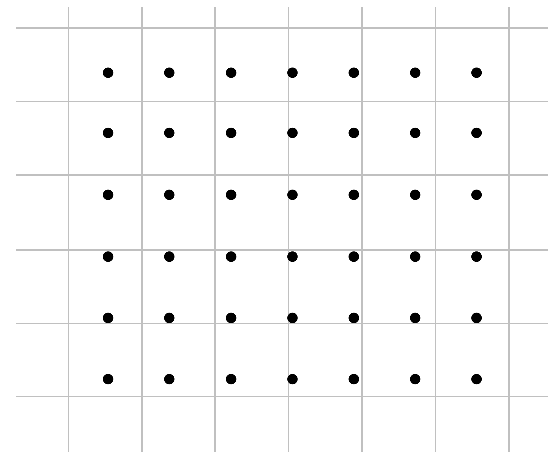
MAXIMIZATION

Image registration: rigid transformations

Translation



Scaling



Rotation

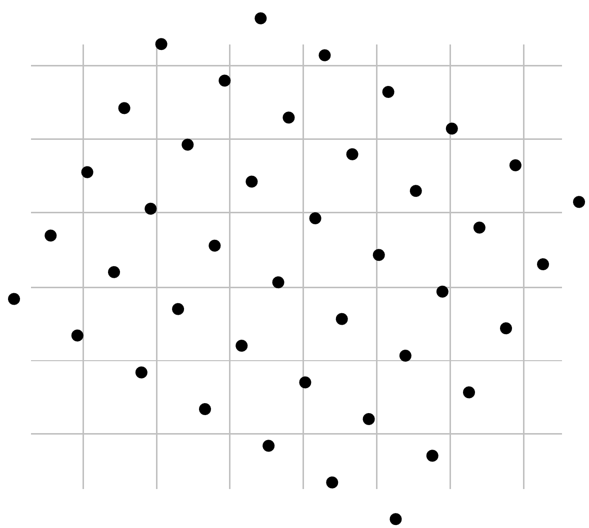


Image registration: transformations

Interpolation

The contribution of I(0,0):

- Proportional to the opposite “rectangle”

$$(1-x)\cdot(1-y)\cdot I(0,0)$$

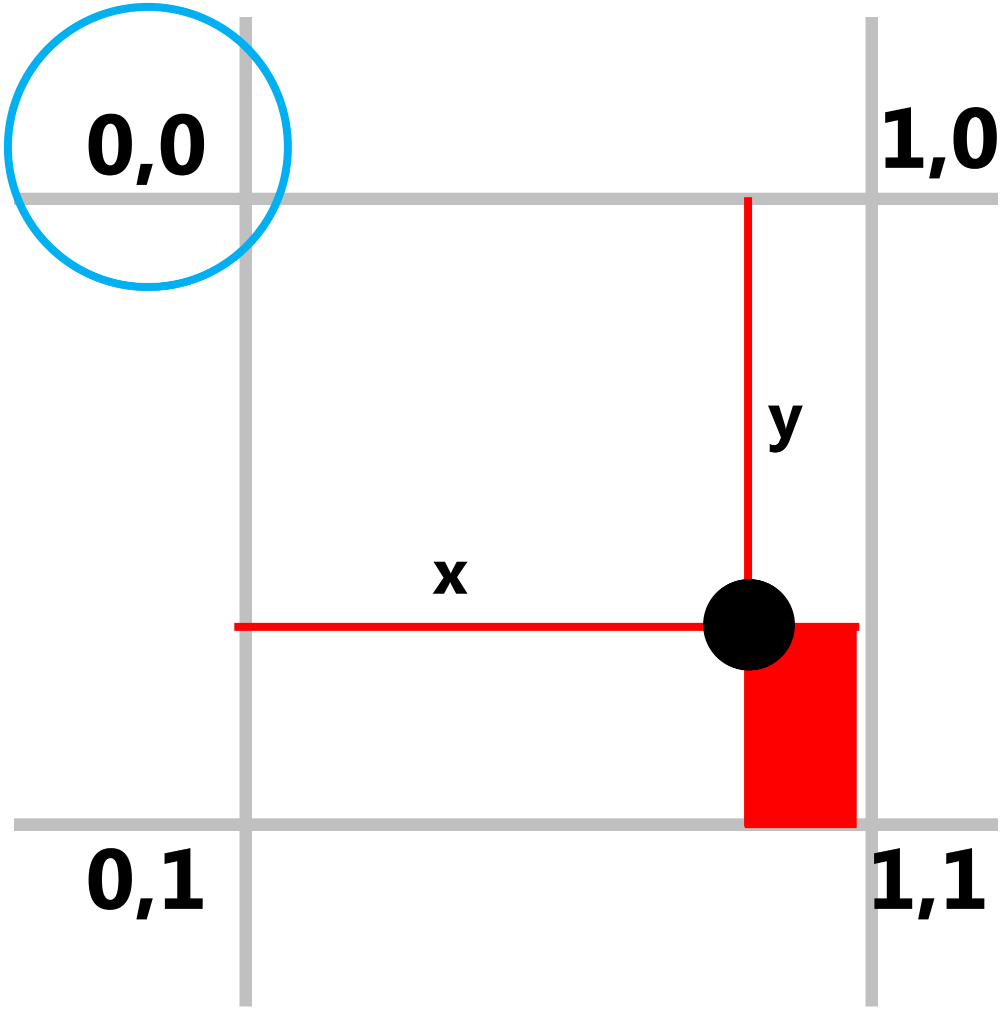


Image registration: transformations

Interpolation

$$
\begin{aligned}
&I(\bullet) \\
&= (1-x)\cdot(1-y)\cdot I(0,0) \\
&+ x\cdot(1-y)\cdot I(1,0) \\
&+ (1-x)\cdot y\cdot I(0,1) \\
&+ \quad x\cdot y\cdot I(1,1)
\end{aligned}
$$

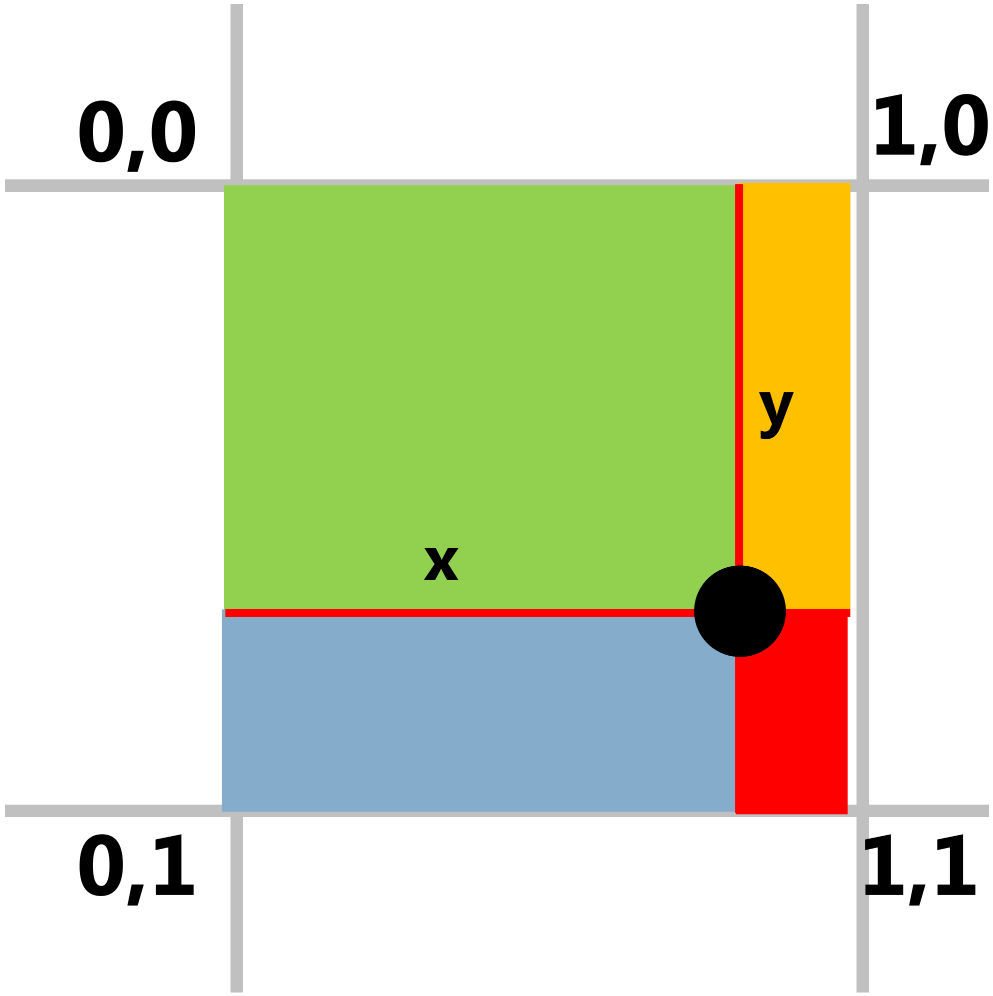
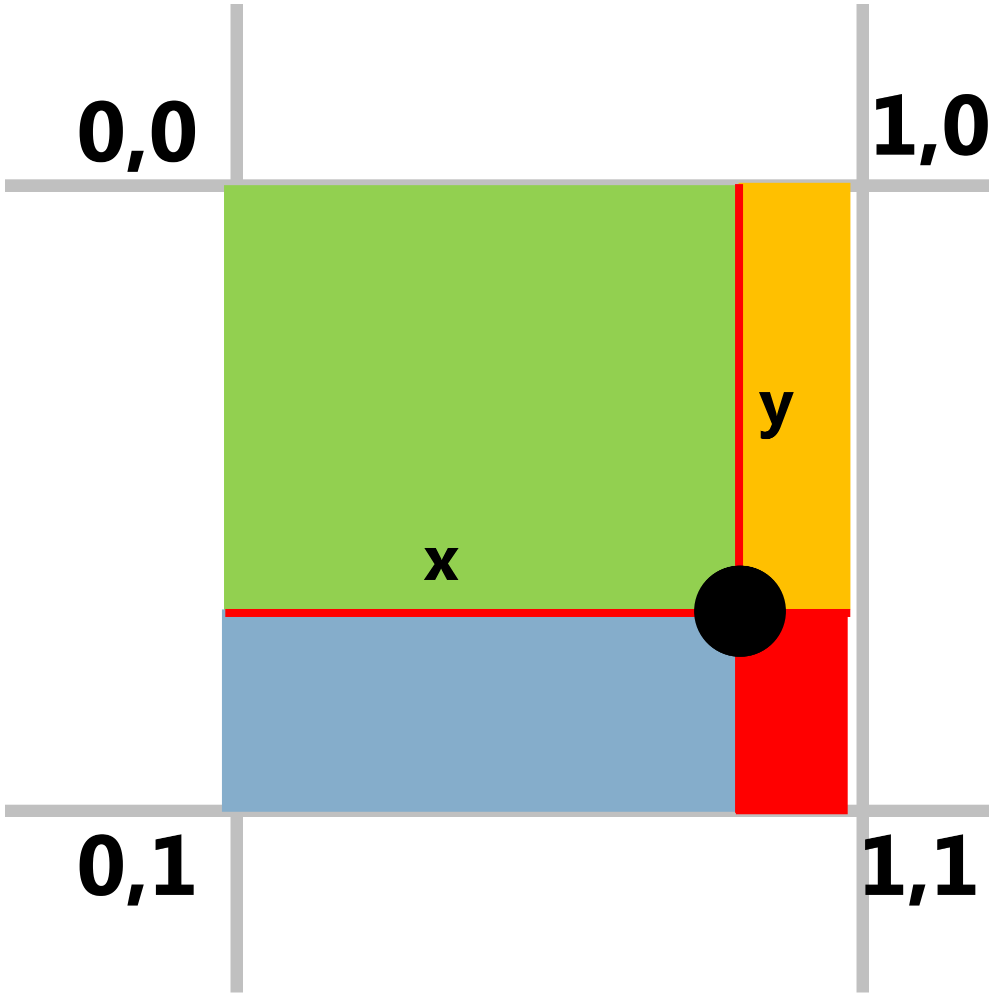


Image registration: transformations

Interpolation

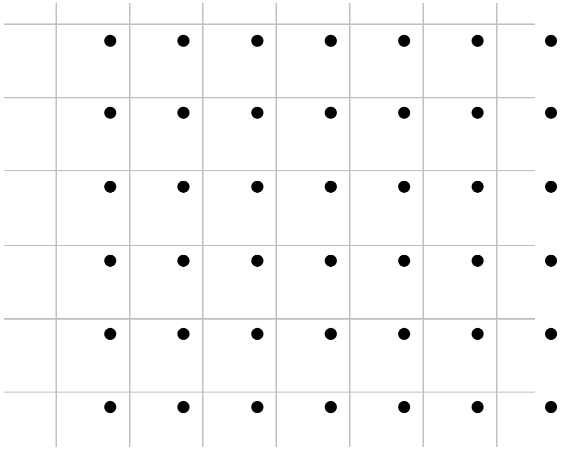
$$
\begin{aligned}
&I(\bullet) \\
&= (1-x)\cdot(1-y)\cdot I(0,0) \\
&+ x\cdot(1-y)\cdot I(1,0) \\
&+ (1-x)\cdot y\cdot I(0,1) \\
&+ \quad x\cdot y\cdot I(1,1)
\end{aligned}
$$



Break

Image registration: optimization

We want to optimize:



$$\min \sum_{i,j} d(I(i,j) - J(x(i,j,\Theta), y(i,j,\Theta)))$$

$d(I,J)$ – similarly measure

$x(i,j,\Theta)$ – transformation for first coordinate

y(i,j,Θ) – transformation for second coordinate

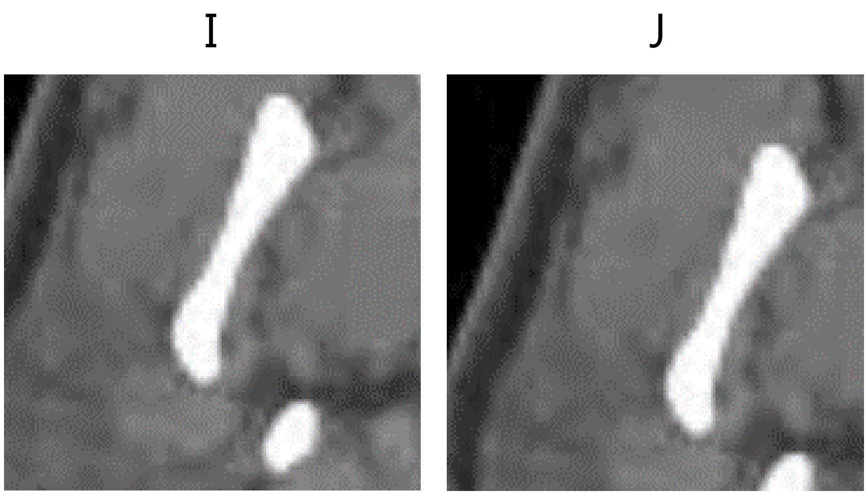


Image registration: optimization

We want to optimize:

$$\min \sum_{i,j} d(I(i,j) - J(x(i,j,\Theta), y(i,j,\Theta)))$$

$$d(I(i,j), J(x,y)) = (I(i,j) - J(x,y))^2$$

$f(I,J,\Theta)$ – translation using Θ

$$f(i,j,\Theta) = [i + \Theta_x, j + \Theta_y]$$

Image registration: optimization

We want to optimize:

$$E(\Theta) = \sum_{i,j} (I(i,j) - J(x(i,j,\Theta), y(i,j,\Theta)))^2$$

Gradient decent algorithm:

$$\nabla E(\Theta) = \frac{\partial E(\Theta)}{\partial \Theta}$$

$$\Theta_{t+1} = \Theta_t + \eta \nabla E(\Theta)$$

Image registration: optimization

Gradient:

$$\nabla E(\Theta) = \nabla \sum_{i,j} \left(I(i,j) - J(x(i,j,\Theta), y(i,j,\Theta))\right)^2 =$$

$$-\sum_{i,j} 2\left(I(i,j) - J(x(i,j,\Theta), y(i,j,\Theta))\right) \cdot \frac{\partial J}{\partial \Theta}$$

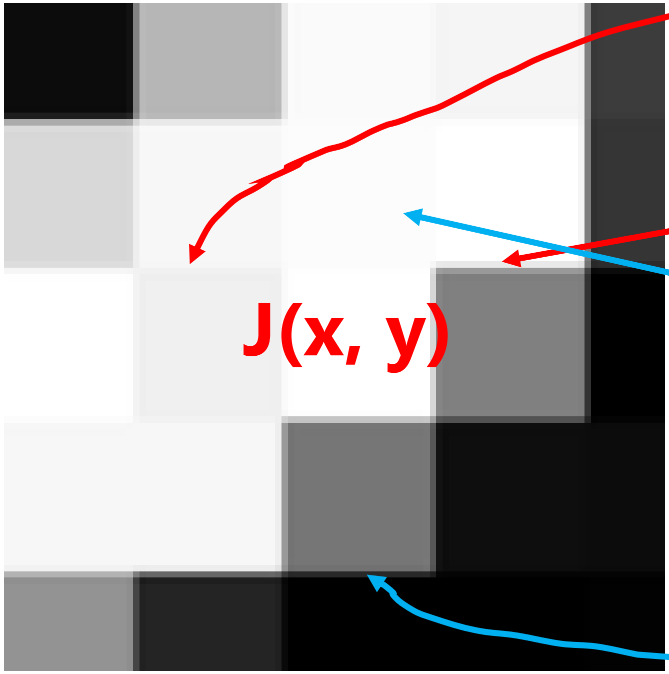
$$\frac{\partial J}{\partial \Theta} = \frac{\partial J}{\partial \mathrm{X}} \cdot \frac{\partial X}{\partial \Theta}$$

Transformed coordinates

Image registration: optimization

Gradient $\frac{\partial J}{\partial \Theta} = \frac{\partial J}{\partial X} \cdot \frac{\partial X}{\partial \Theta}$:

$$\frac{\partial J}{\partial X} = \left[\frac{\partial J(x(i,j,\Theta), y(i,j,\Theta))}{\partial x}; \frac{\partial J(x(i,j,\Theta), y(i,j,\Theta))}{\partial y}\right]$$



$$\frac{\partial J}{\partial x} = \frac{1}{2}(I(x+1,y) - I(x-1,y))$$

$$\frac{\partial J}{\partial y} = \frac{1}{2}(I(x,y+1) - I(x,y-1))$$

Image registration: optimization

Gradient $\frac{\partial J}{\partial \Theta} = \frac{\partial J}{\partial \mathrm{X}} \cdot \frac{\partial X}{\partial \Theta}$:

$$\mathrm{T}(x, y) = \begin{bmatrix} 1 & 0 & \Theta_x \\ 0 & 1 & \Theta_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [x + \Theta_x, y + \Theta_y]$$

$$\frac{\partial X}{\partial \Theta} = \left[\frac{\partial X}{\partial \Theta_x}; \frac{\partial X}{\partial \Theta_y} \right] = [1, 1]$$

$$\frac{\partial J}{\partial \Theta} = [\frac{1}{2}\big(I(x+1, y) - I(x-1, y)\big), \frac{1}{2}(I(x, y+1) - I(x, y-1))]$$

Image registration: optimization

Gradient:

$$\nabla E(\Theta) = \begin{bmatrix} -\sum_{i,j} 2\big(I(i,j) - J(x(i,j,\Theta), y(i,j,\Theta))\big) \cdot \frac{1}{2}\big(I(x+1,y) - I(x-1,y)\big) \\ -\sum_{i,j} 2\big(I(i,j) - J(x(i,j,\Theta), y(i,j,\Theta))\big) \cdot \frac{1}{2}\big(I(x,y+1) - I(x,y-1)\big) \end{bmatrix}$$

$$\Theta_{t+1} = \Theta_t + \eta \nabla E(\Theta)$$

Image registration: optimization

Problem with optimization:

$$\min \sum_{i,j} d(I(i,j) - J(x(i,j,\Theta), y(i,j,\Theta)))$$

- $d(I,J)$ – can be complex and expensive to compute (mutual information)
- Θ - can contain many variables:
  - 9 for similarity transformation

$$\begin{bmatrix} \Theta_1 & \Theta_2 & \Theta_3 \\ \Theta_4 & \Theta_5 & \Theta_6 \\ \Theta_7 & \Theta_8 & \Theta_9 \end{bmatrix}$$

  - thousands for non-rigid transformations

How to simplify this?

Image registration: optimization

Multi-resolution pyramid registration:

- Downscale (and smooth) reference and moving images x16
- Register them and memorize transformation
- Downscale (and smooth) reference and moving images x8, apply memorized transformation
- Register images and combine transformations from both steps
- Continue

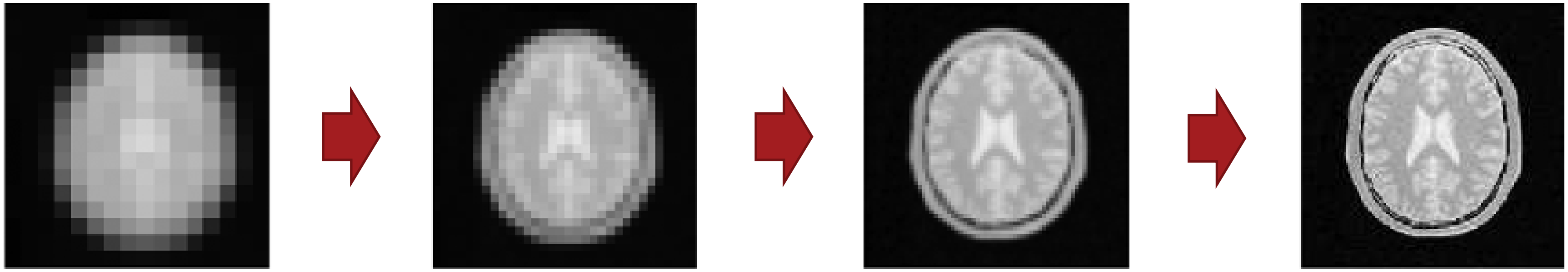
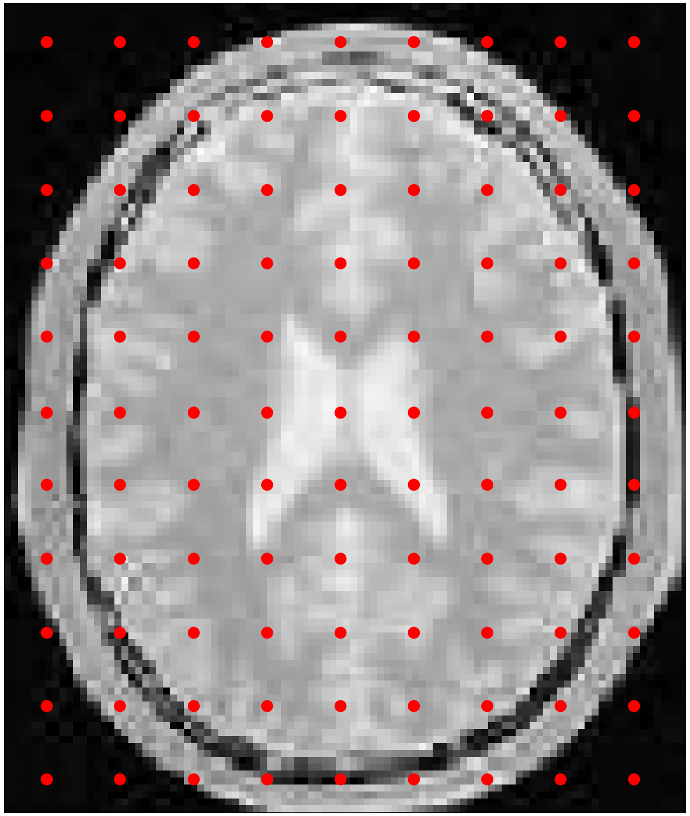


Image registration: optimization

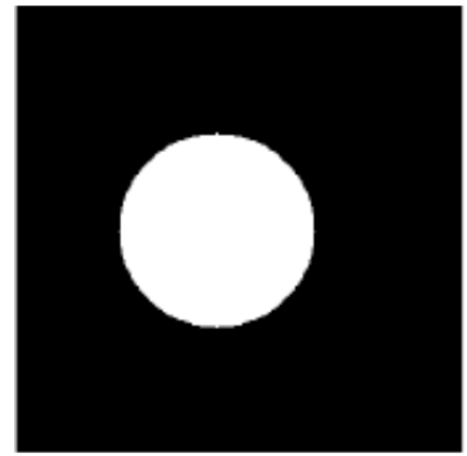
Grid-based registration:

- Do not perform registration per pixel but use a sparse grid

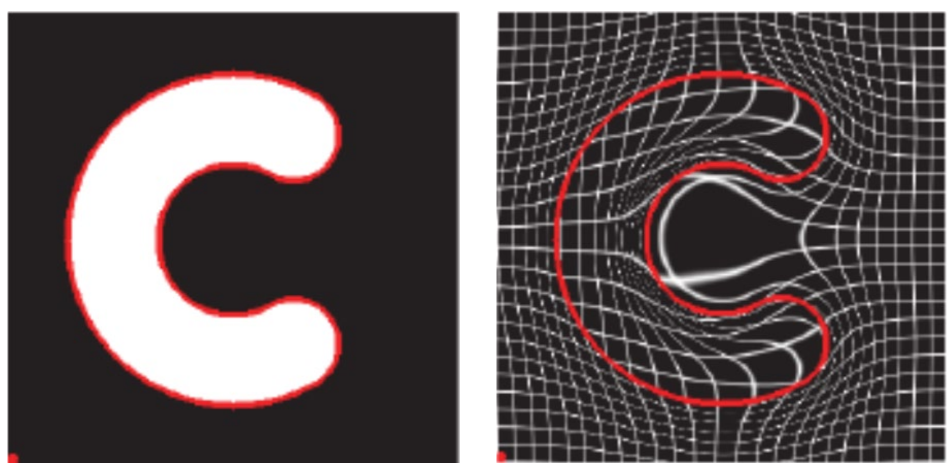


Non-rigid image registration

Reference



Moving



Transformation

Non-rigid image registration

Transformation field:

- The connections between points are not straight lines
- The connections are kinda smooth but how to generate them?

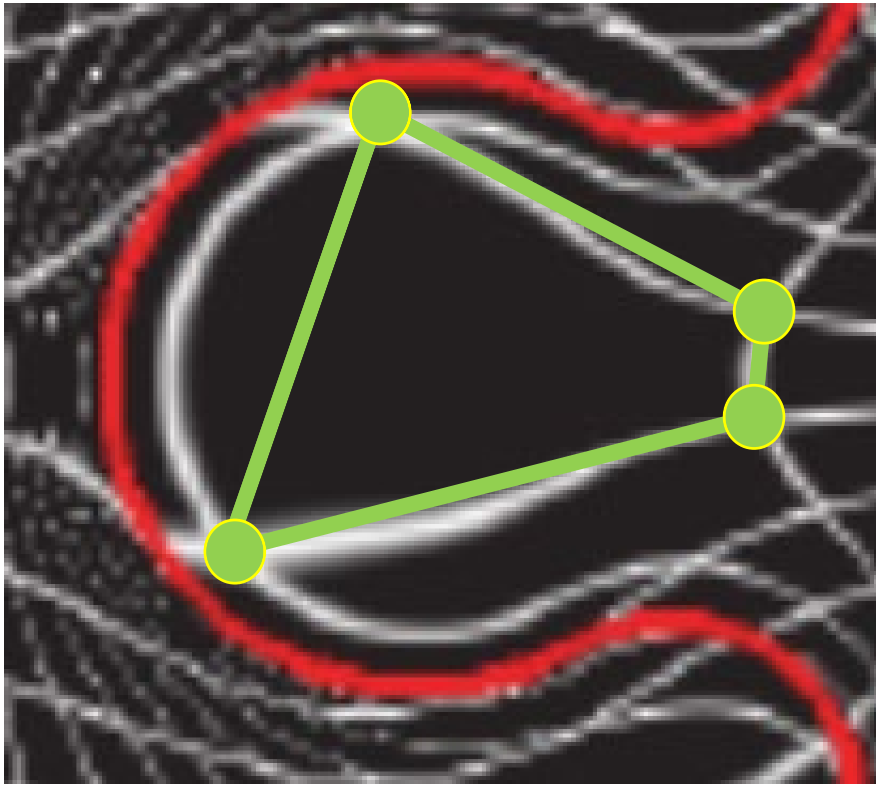


Image registration: spline interpolation

Linear interpolation:

- We connect knots with straight lines
- The problem is that the curve break at knots

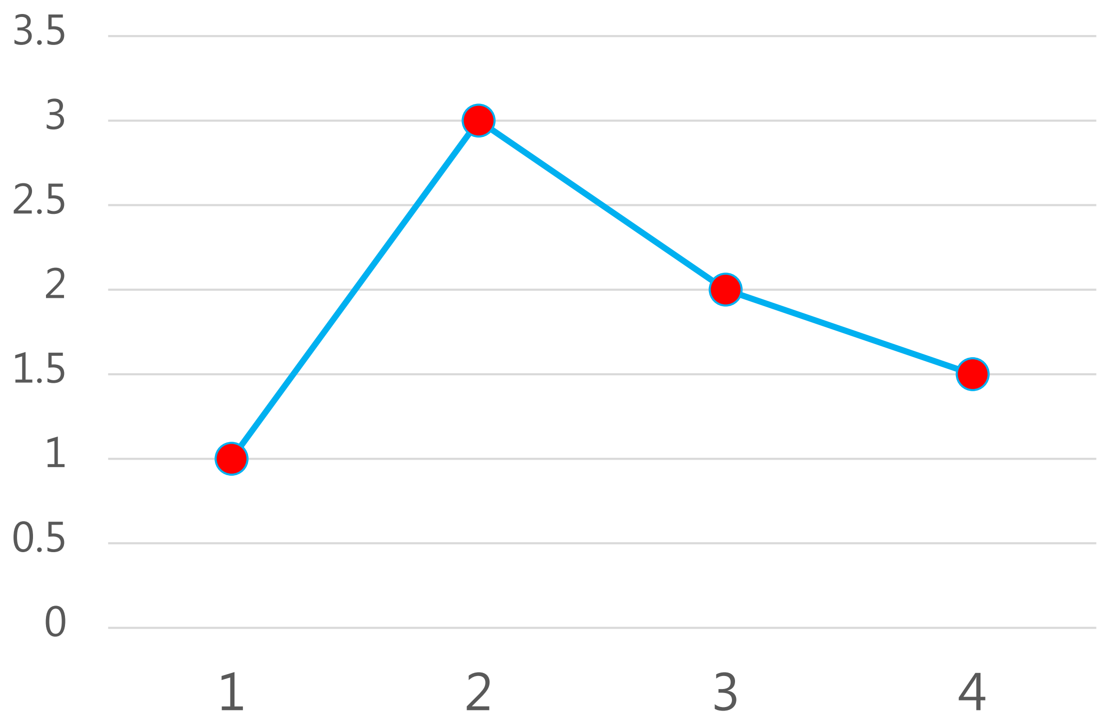
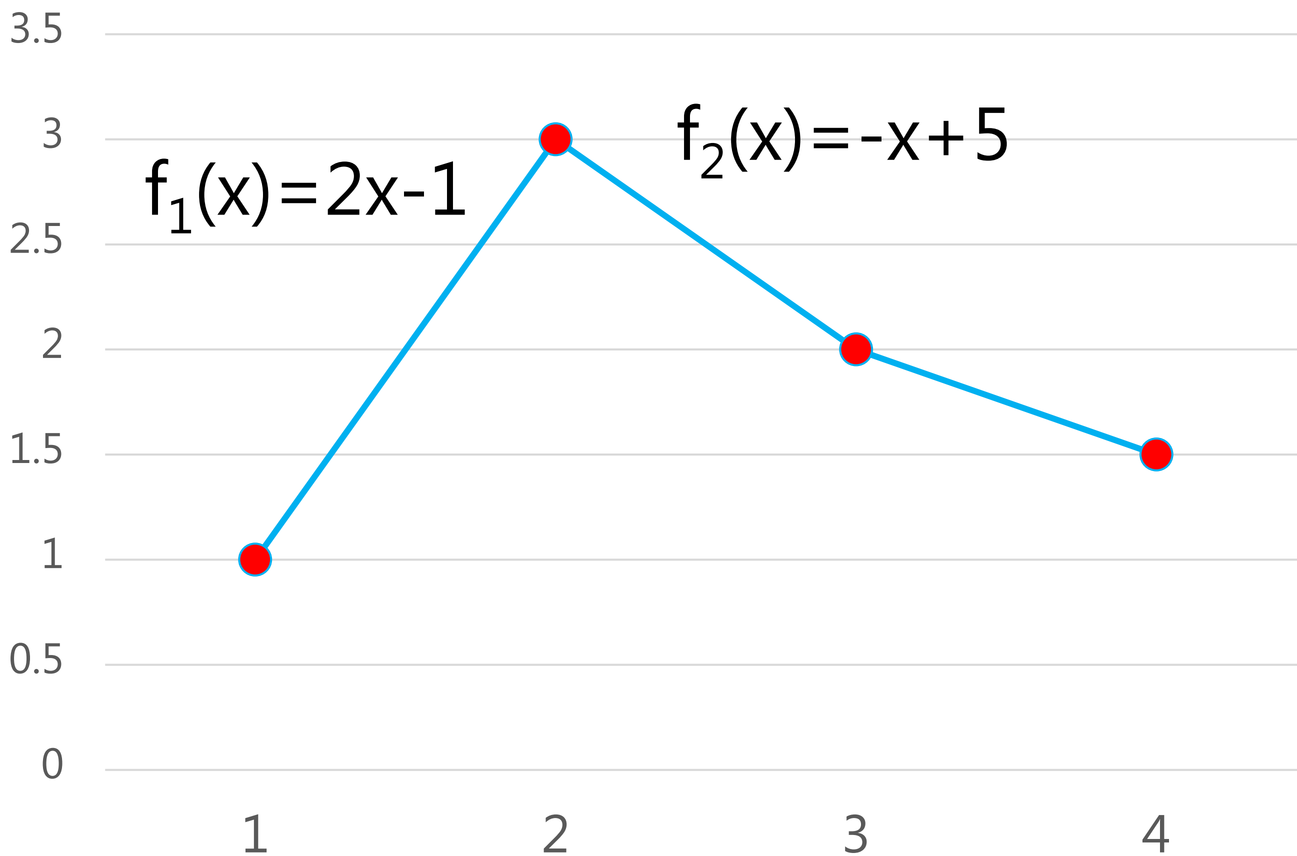


Image registration: spline interpolation

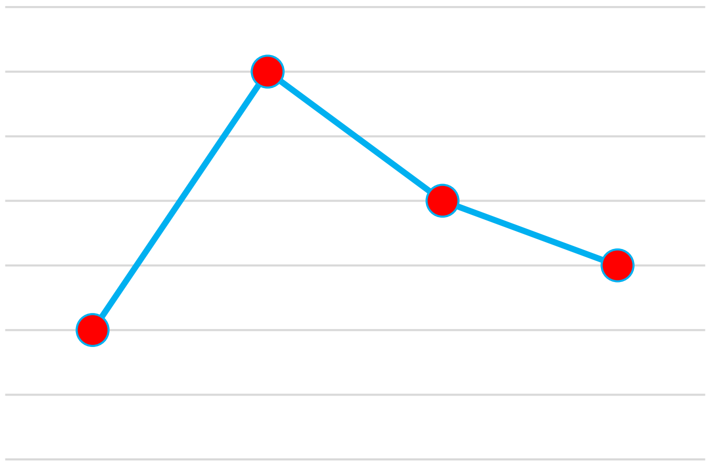
Curve breaks at non-differentiable points:



$df_1(2) = 2$ $df_2(2) = -1$ $df_1(2) \neq df_2(2)$

Image registration: spline interpolation

Curve breaks at non-differentiable points:



- Use quadratic polynomials

$$f_n(x) = a_n x^2 + b_n x + c_n$$

- We have:
  - 3*3 variables, i.e. three segments f_n with 3 unknowns each
  - 2*3 conditions, i.e. each of the three segments should pass through the corresponding knots

Image registration: spline interpolation

Curve breaks at non-differentiable points:

- Three more equations are needed

- Let's ensure smoothness of the curve

$$df_1(2) = df_2(2)$$

$$df_2(3) = df_3(3)$$

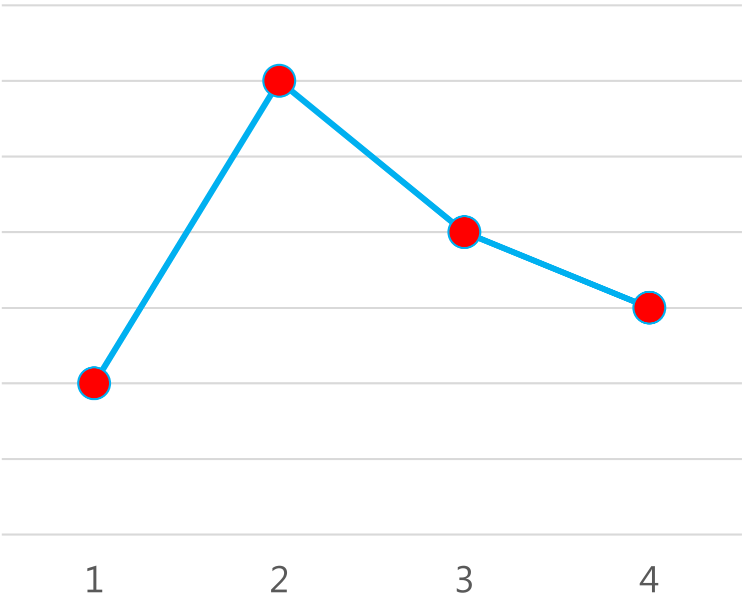


Image registration: spline interpolation

Spline interpolation solution:

$$a_1 \cdot 1 + b_1 \cdot 1 + c_1 = 1$$

$$a_1 \cdot 4 + b_1 \cdot 2 + c_1 = 3$$

$$2a_1 \cdot 2 + b_1 = 2a_2 \cdot 2 + b_2$$

$$2a_2 \cdot 3 + b_2 = 2a_3 \cdot 3 + b_3$$

$$a_2 \cdot 4 + b_2 \cdot 2 + c_2 = 3$$

$$a_2 \cdot 9 + b_2 \cdot 3 + c_2 = 2$$

$$a_3 \cdot 9 + b_3 \cdot 3 + c_3 = 2$$

$$a_3 \cdot 16 + b_3 \cdot 4 + c_3 = 1.5$$

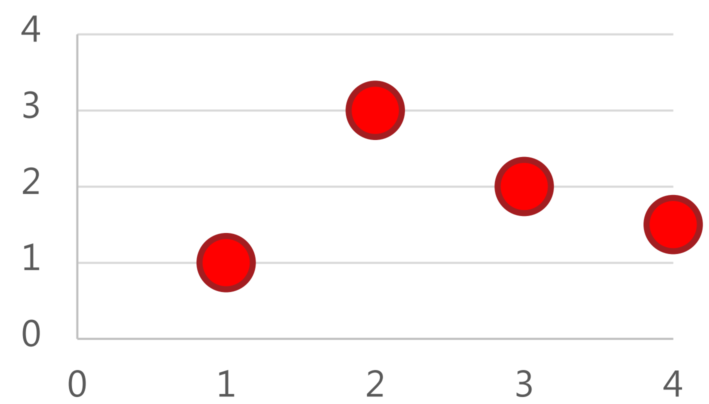
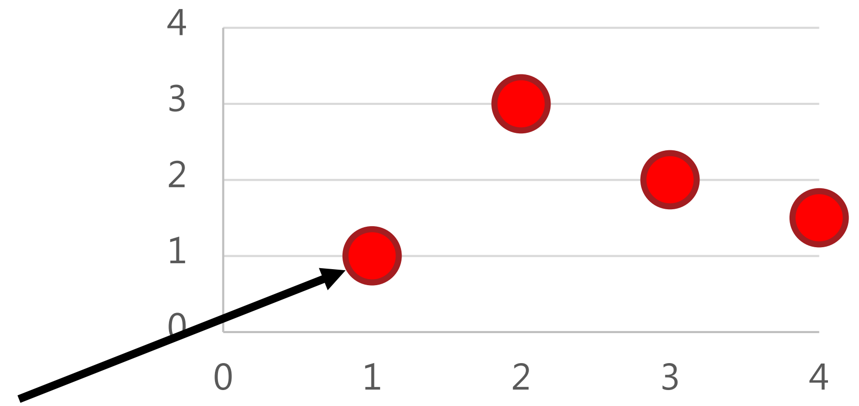


Image registration: spline interpolation

Curve breaks at non-differentiable points:

- Three more equations are needed
- Last equations could be anything reasonable

$$df_1(1) = 0$$



$$2a_1 \cdot 1 + b_1 = 0$$

The curve will be oriented horizontally at this point

Image registration: spline interpolation

Spline interpolation solution:

$$2a_1 + b_1 = 0 \quad \rightarrow \quad b_1 = -2a_1$$

$$a_1 \cdot 1 + b_1 \cdot 1 + c_1 = 1 \quad \rightarrow \quad c_1 - a_1 = 1$$

$$a_1 \cdot 4 + b_1 \cdot 2 + c_1 = 3 \quad \rightarrow \quad c_1 = 3; a_1 = 2; b_1 = -4$$

$$f_1(x) = 2x^2 - 4x + 3$$

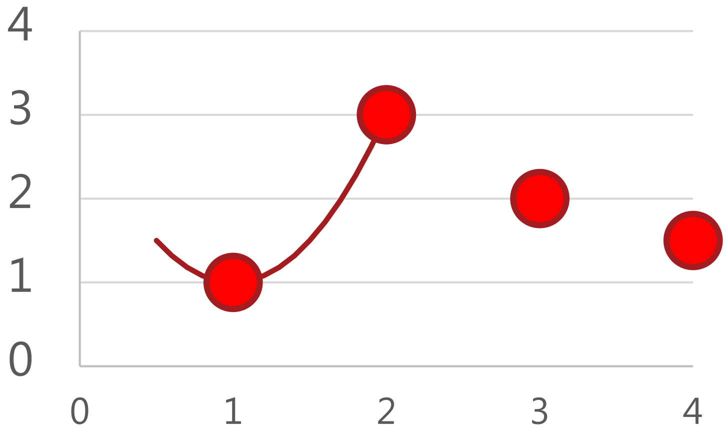


Image registration: spline interpolation

Spline interpolation solution:

$a_1 = 2; b_1 = -4; c_1 = 3$

$$2a_1 \cdot 2 + b_1 = 2a_2 \cdot 2 + b_2 \quad \rightarrow \quad 4a_2 + b_2 = 4$$

$$a_2 \cdot 4 + b_2 \cdot 2 + c_2 = 3 \quad \rightarrow \quad b_2 + c_2 = -1$$

$$a_2 \cdot 9 + b_2 \cdot 3 + c_2 = 2 \quad \rightarrow \quad a_2 = -5; b_2 = 24; c_2 = -25$$

$$\|$$

$$a_2 + 2(4a_2 + b_2) + (b_2 + c_2)$$

$$f_2(x) = -5x^2 + 24x - 25$$

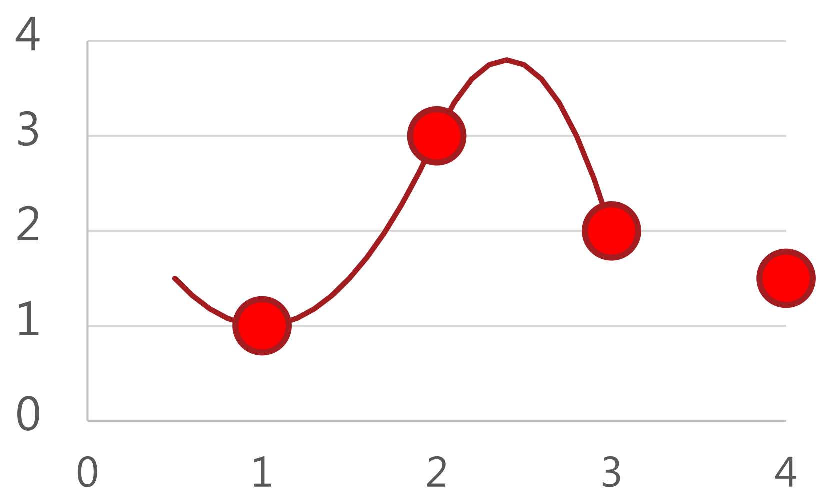


Image registration: spline interpolation

Spline interpolation solution:

$a_2 = -5; b_2 = 24; c_2 = -25$

$$2a_2 \cdot 3 + b_2 = 2a_3 \cdot 3 + b_3 \quad \rightarrow \quad 6a_3 + b_3 = -6$$

$$a_3 \cdot 9 + b_3 \cdot 3 + c_3 = 2$$

$$a_3 \cdot 16 + b_3 \cdot 4 + c_3 = 1.5$$

$$a_3 = 5.5;$$
$$b_3 = -39;$$
$$c_3 = 69.5$$

$$f_3(x) = 5.5x^2 - 39x + 69.5$$

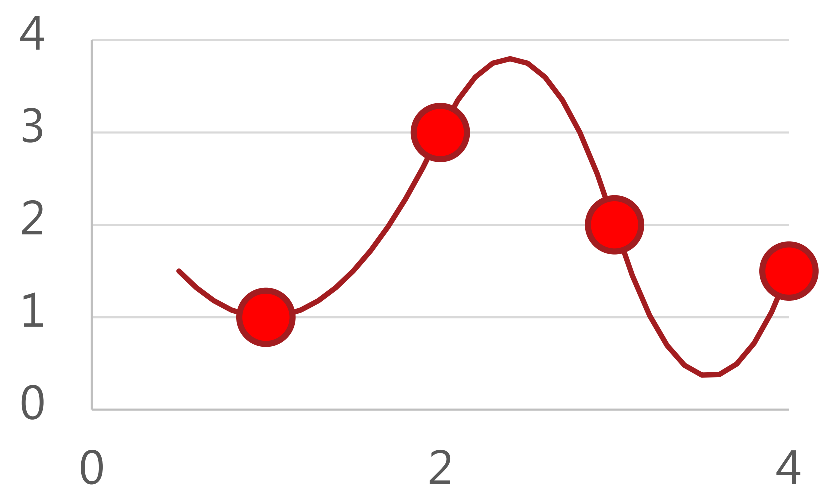
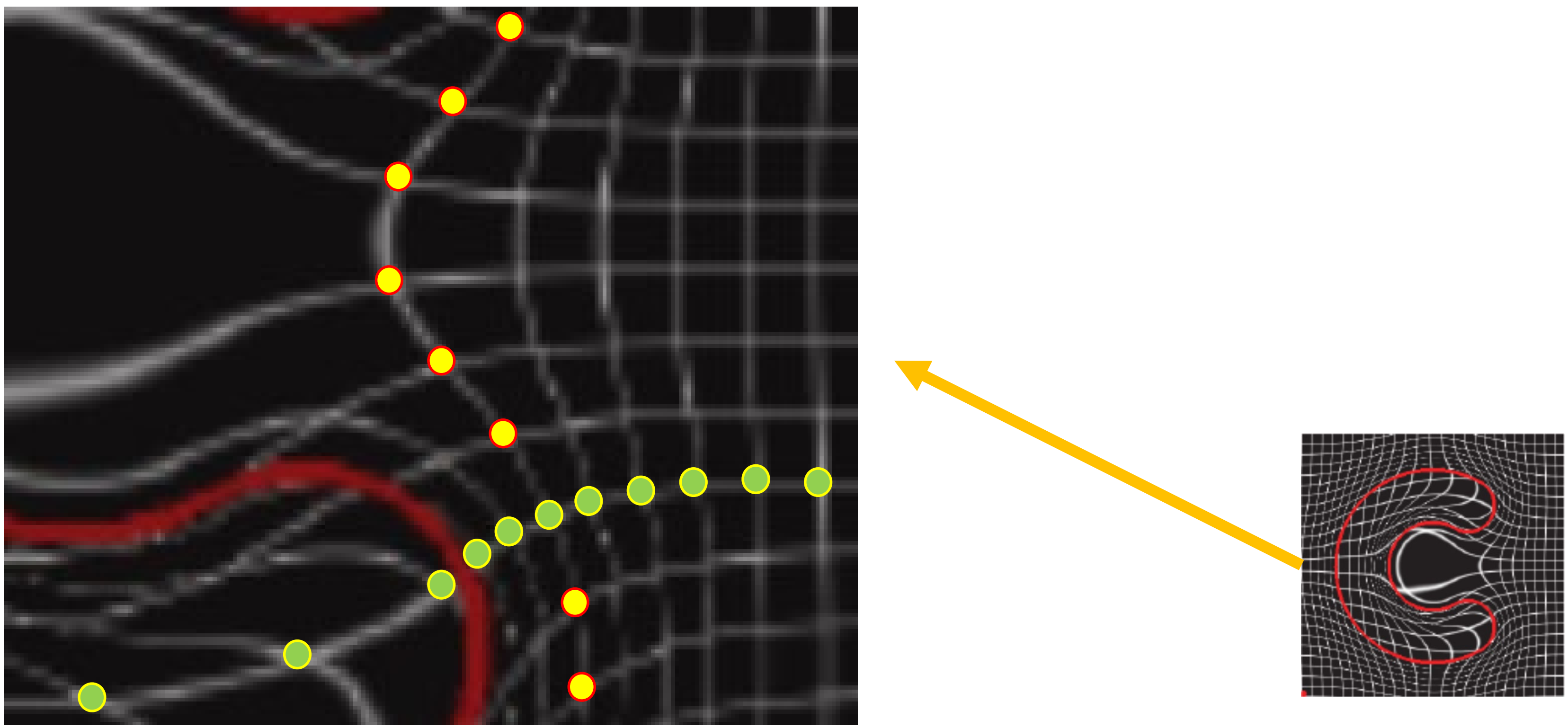


Image registration: spline interpolation

We can apply spline interpolation for each row and column of the grid



Questions?