Nonlinear Registration

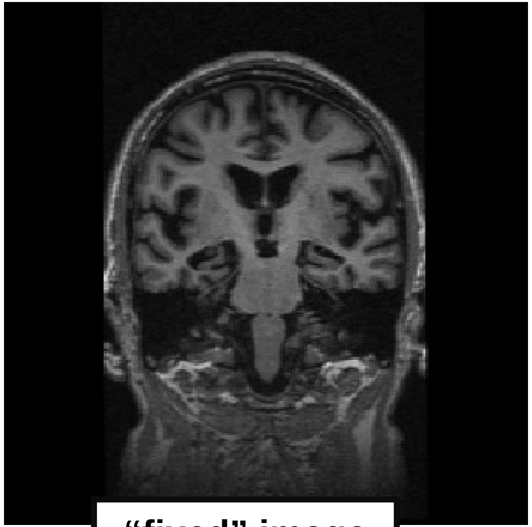
Aalto-yliopisto
Aalto-universitetet
Aalto University

Medical Image Analysis

Koen Van Leemput

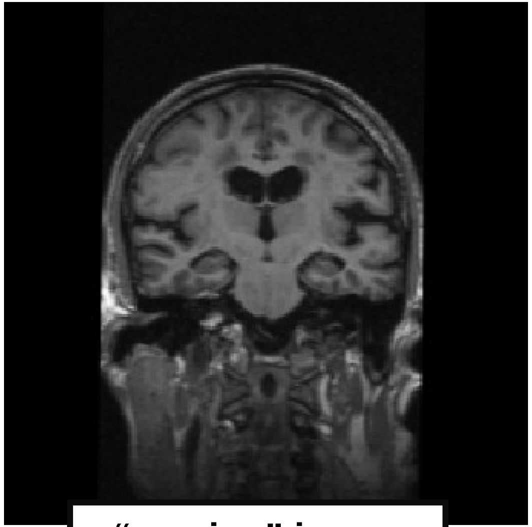
Fall 2024

Spatial transformations



“fixed” image

$$\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$$

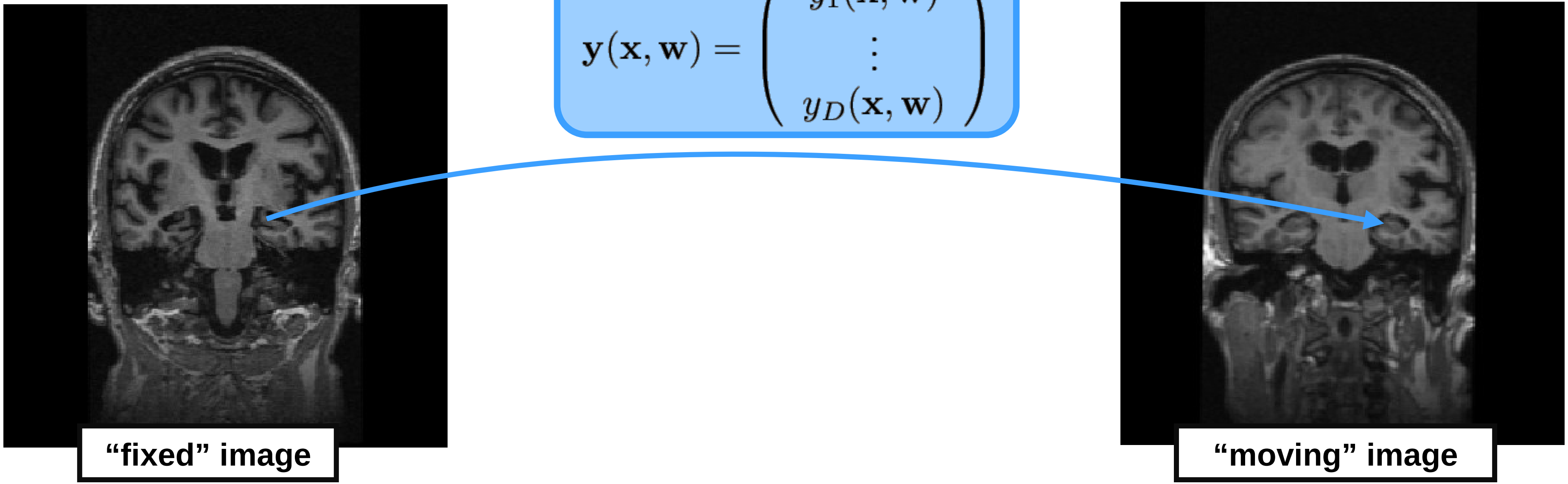


“moving” image

$$\mathbf{y} = (y_1, \ldots, y_D)^{\mathrm{T}}$$

Spatial transformations

$$\mathbf{y}(\mathbf{x},\mathbf{w}) = \begin{pmatrix} y_1(\mathbf{x},\mathbf{w}) \\ \vdots \\ y_D(\mathbf{x},\mathbf{w}) \end{pmatrix}$$



“fixed” image

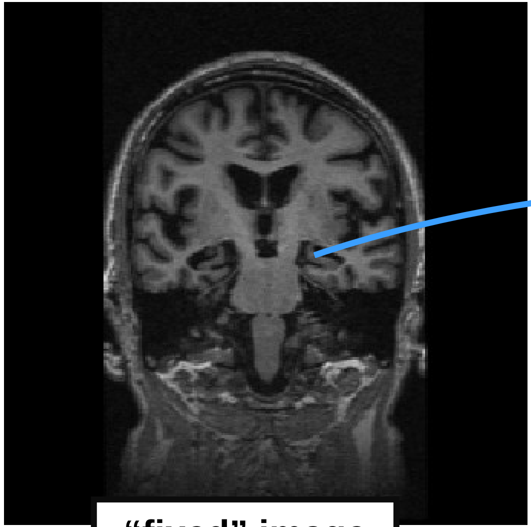
$\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$

“moving” image

$\mathbf{y} = (y_1, \ldots, y_D)^{\mathrm{T}}$

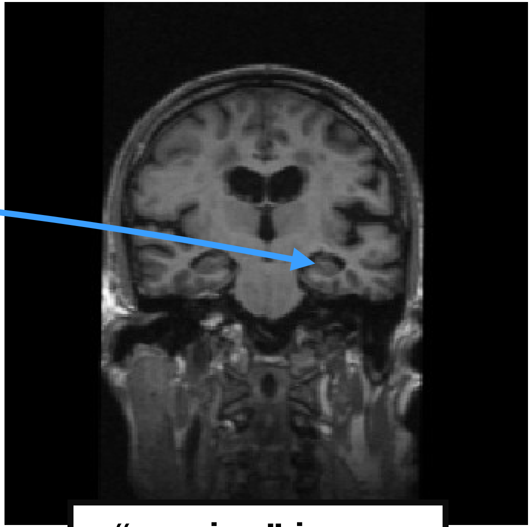
Spatial transformations

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \begin{pmatrix} y_1(\mathbf{x}, \mathbf{w}) \\ \vdots \\ y_D(\mathbf{x}, \mathbf{w}) \end{pmatrix}$$



“fixed” image

$\mathbf{x} = (x_1, \ldots, x_D)^{\mathrm{T}}$

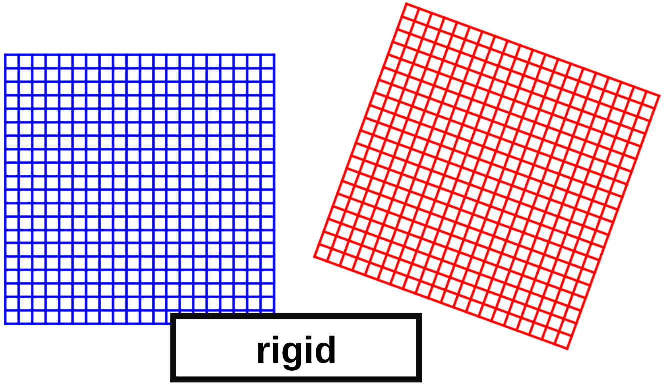


“moving” image

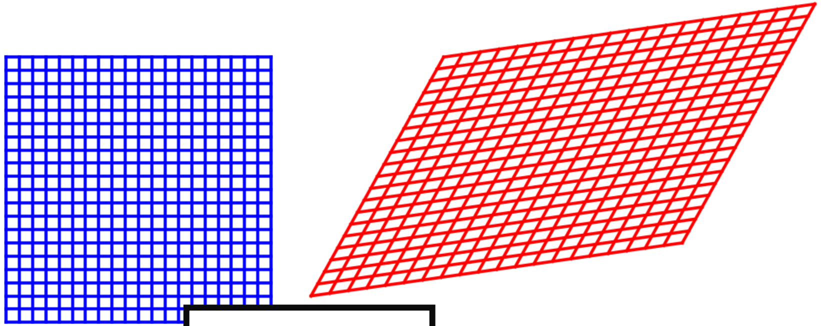
$\mathbf{y} = (y_1, \ldots, y_D)^{\mathrm{T}}$

$y_d(\mathbf{x}, \mathbf{w})$
controls how points $\mathbf{x}$ in the fixed image
move along the d-th direction in the moving image
as the parameters $\mathbf{w}$ are varied

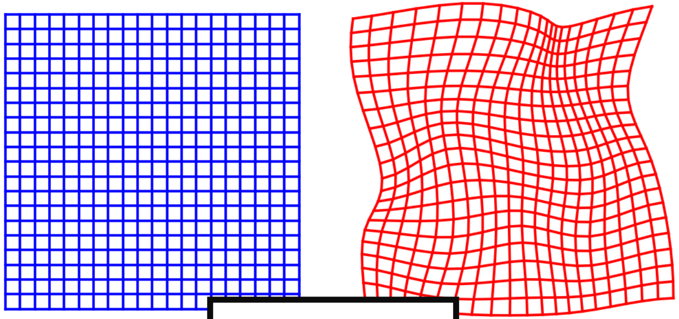
Spatial transformations



rigid

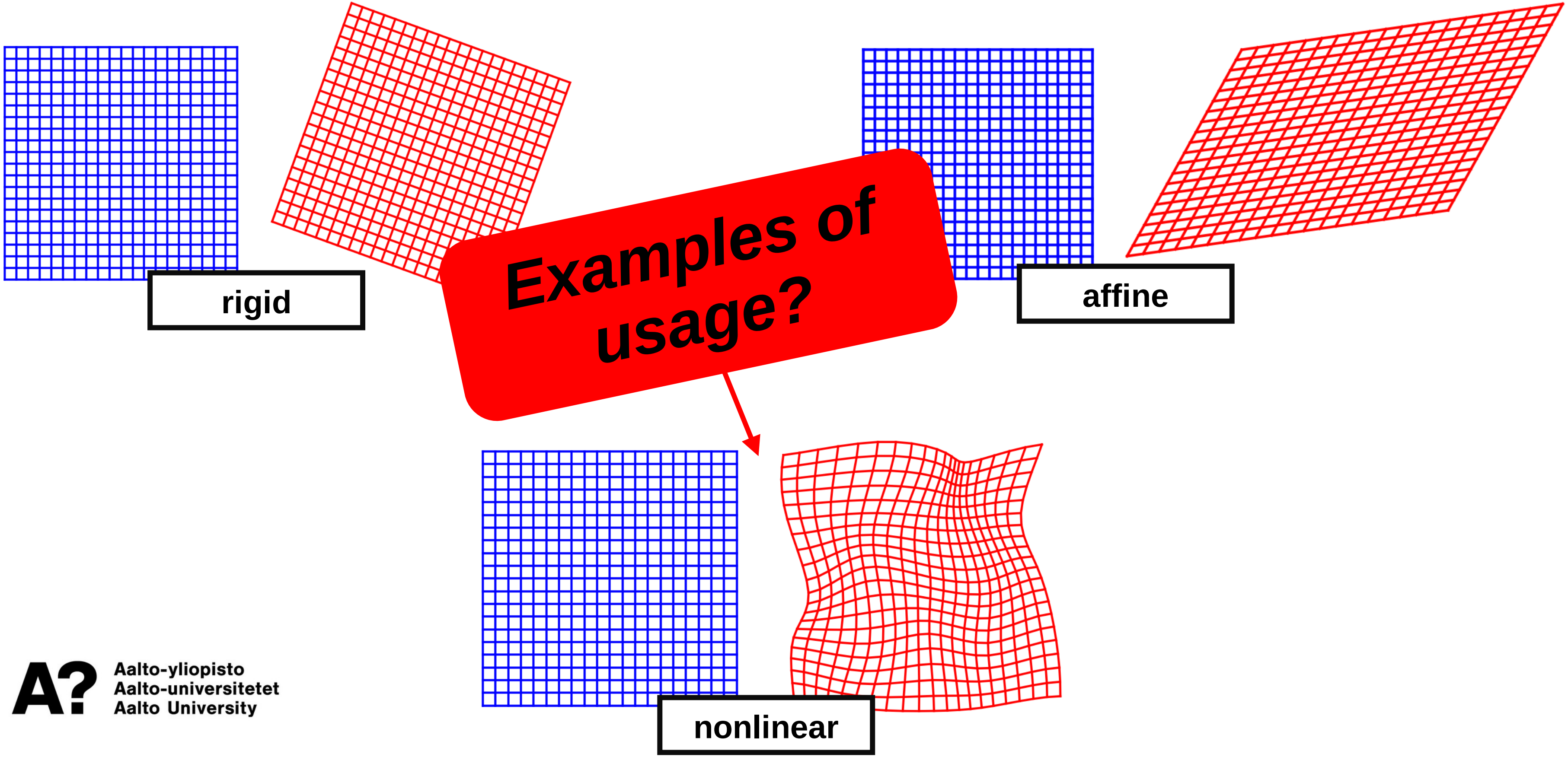


affine



nonlinear

Spatial transformations



rigid

affine

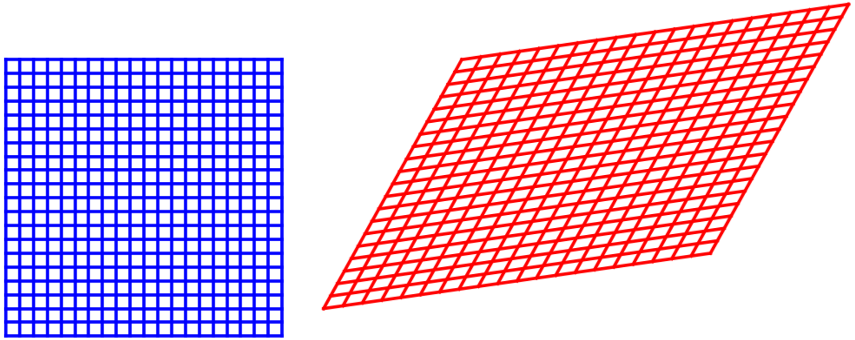
Examples of usage?

nonlinear

Affine transformation

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$

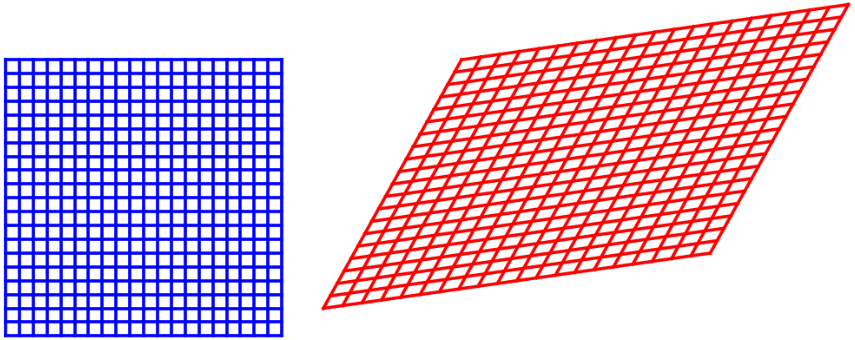
$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \quad \text{and} \quad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$



Affine transformation

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \quad \text{and} \quad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$



$y_d(\mathbf{x}, \mathbf{w})$
controls how points $\mathbf{x}$ in the fixed image
move along the d-th direction in the moving image
as the parameters $\mathbf{w}$ are varied

$$y_d(\mathbf{x}, \mathbf{w}_d) = t_d + a_{d,1}x_1 + \ldots + a_{d,D}x_D$$

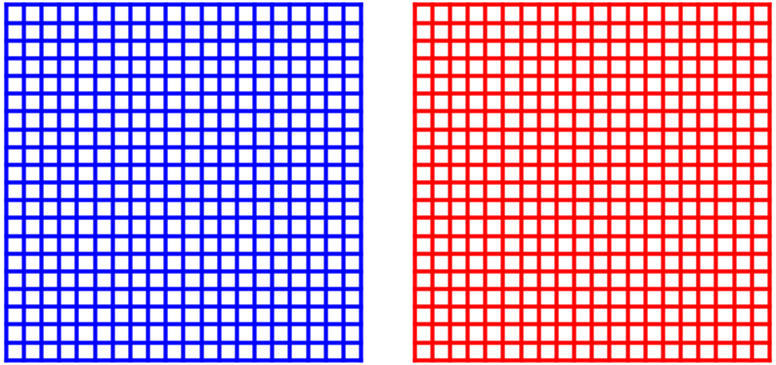
$$\mathbf{w}_d = (t_d, a_{d,1}, \ldots, a_{d,D})^{\mathrm{T}}$$

$$\mathbf{w} = (\mathbf{w}_1^{\mathrm{T}}, \ldots, \mathbf{w}_D^{\mathrm{T}})^{\mathrm{T}}$$

Aalto-yliopisto
Aalto-universitetet
Aalto University

Affine transformation

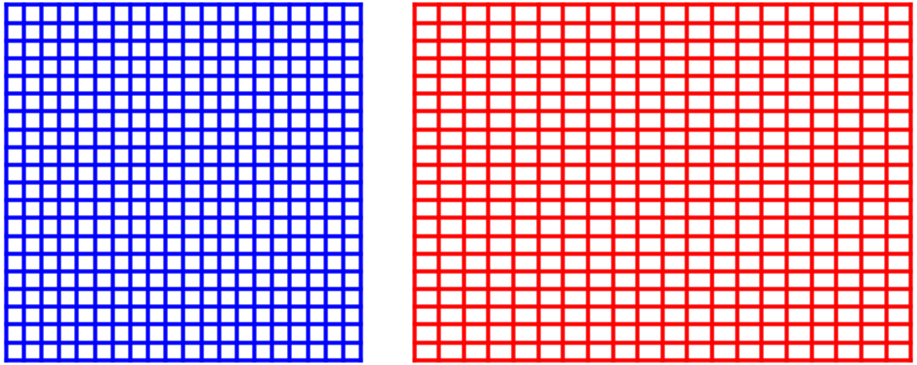
$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$



$$\mathbf{A} = \begin{pmatrix} \boxed{1.0 \quad 0.0} \\ 0.0 \quad 1.0 \end{pmatrix}, \quad \mathbf{t} = \begin{pmatrix} \boxed{23} \\ 0 \end{pmatrix}$$

Affine transformation

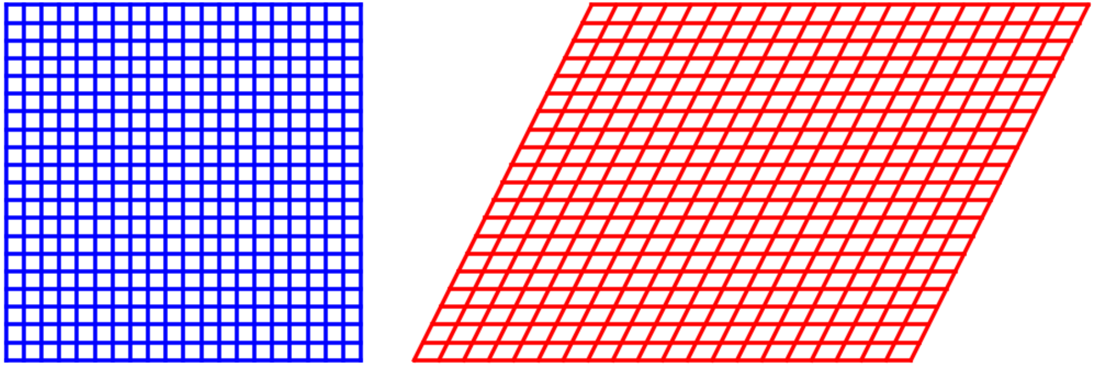
$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$



$$\mathbf{A} = \begin{pmatrix} \boxed{1.4 \quad 0.0} \\ 0.0 \quad 1.0 \end{pmatrix}, \quad \mathbf{t} = \begin{pmatrix} \boxed{23} \\ 0 \end{pmatrix}$$

Affine transformation

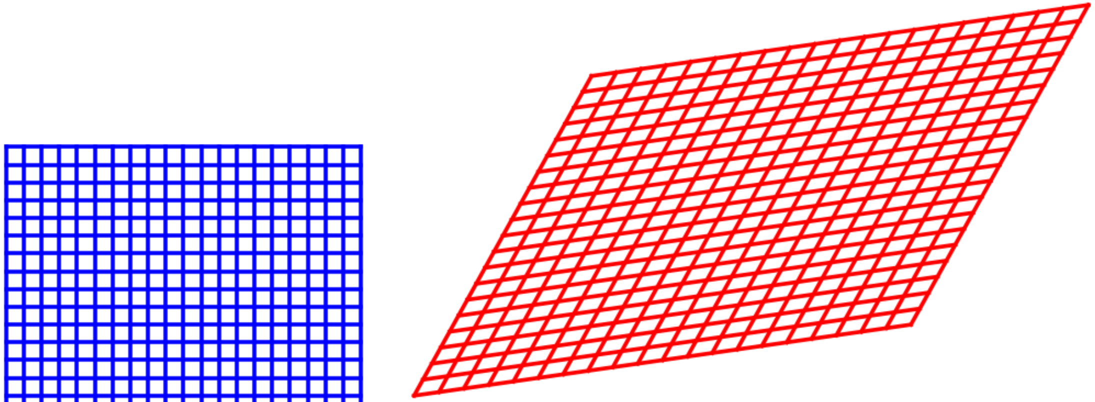
$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$



$$\mathbf{A} = \begin{pmatrix} \boxed{1.4 \quad 0.5} \\ 0.0 \quad 1.0 \end{pmatrix}, \quad \mathbf{t} = \begin{pmatrix} \boxed{23} \\ 0 \end{pmatrix}$$

Affine transformation

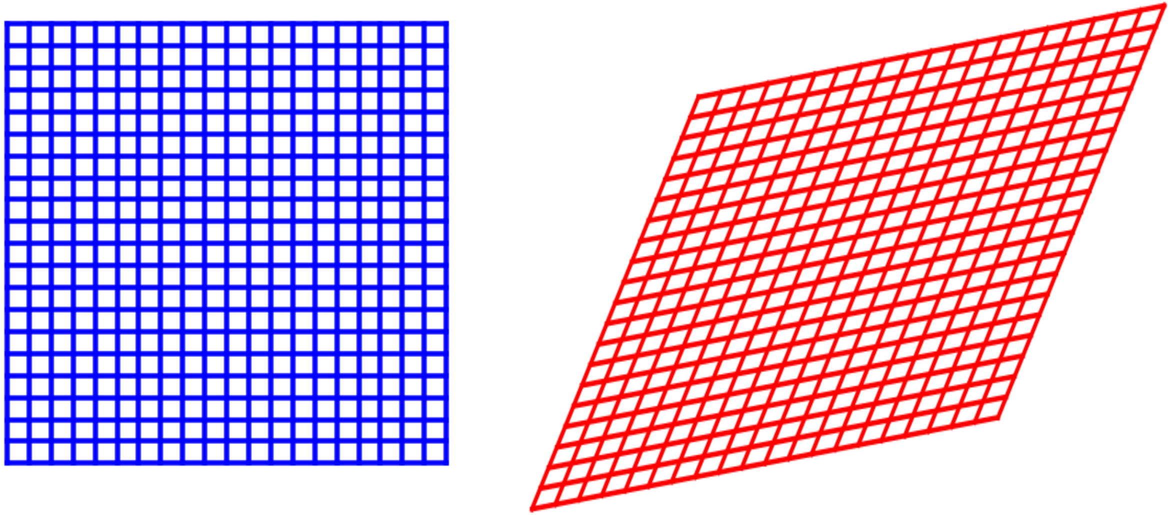
$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$



$$\mathbf{A} = \begin{pmatrix} 1.4 & 0.5 \\ 0.2 & 0.9 \end{pmatrix}, \quad \mathbf{t} = \begin{pmatrix} 23 \\ 6 \end{pmatrix}$$

Affine (*linear*) transformation...

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$



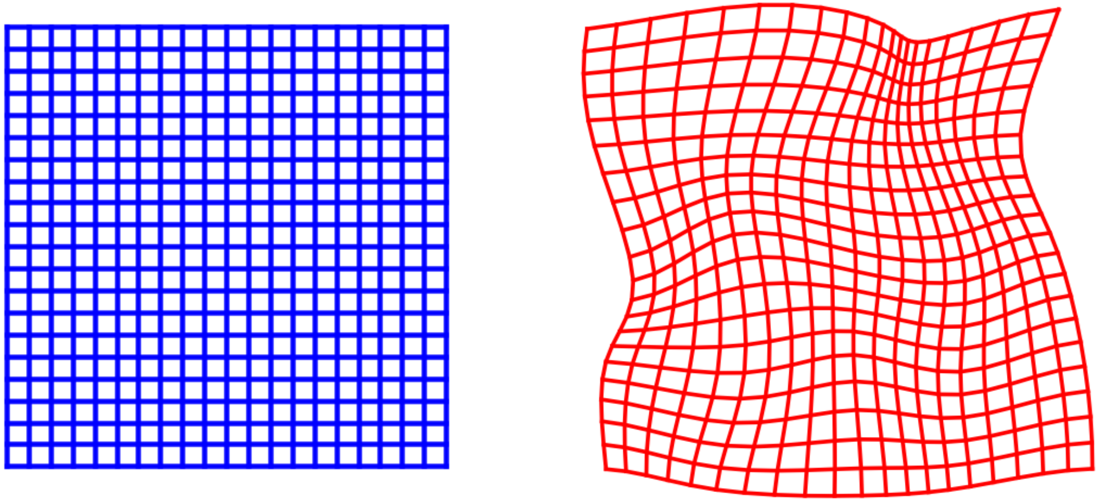
$$y_d(\mathbf{x}, \mathbf{w}_d) = t_d + a_{d,1}x_1 + \ldots + a_{d,D}x_D$$

$$\mathbf{w}_d = (t_d, a_{d,1}, \ldots, a_{d,D})^{\mathrm{T}}$$

$$\mathbf{w} = (\mathbf{w}_1^{\mathrm{T}}, \ldots, \mathbf{w}_D^{\mathrm{T}})^{\mathrm{T}}$$

...vs. *nonlinear* transformation

$$\mathbf{y}(\mathbf{x},\mathbf{w}) = \mathbf{x} + \boldsymbol{\delta}(\mathbf{x},\mathbf{w})$$



$$y_d(\mathbf{x},\mathbf{w}_d) = x_d + \delta_d(\mathbf{x},\mathbf{w}_d) \quad \text{with} \quad \delta_d(\mathbf{x},\mathbf{w}_d) = \sum_{m=0}^{M-1} w_{d,m}\phi_m(\mathbf{x})$$

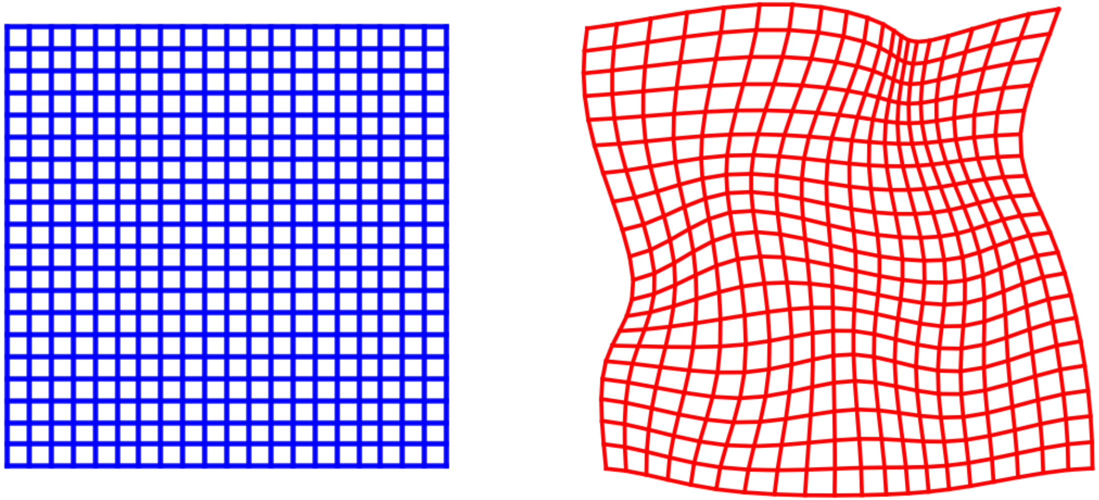
$$\mathbf{w}_d = (w_{d,0}, \ldots w_{d,M-1})^{\mathrm{T}}$$

$$\mathbf{w} = (\mathbf{w}_1^{\mathrm{T}}, \ldots, \mathbf{w}_D^{\mathrm{T}})^{\mathrm{T}}$$

...vs. ***<u>nonlinear</u>*** transformation

"residual deformation"

$$\mathbf{y}(\mathbf{x},\mathbf{w}) = \mathbf{x} + \delta(\mathbf{x},\mathbf{w})$$



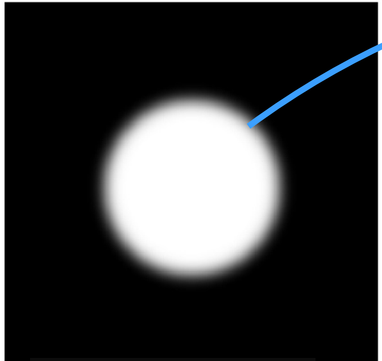
$$y_d(\mathbf{x},\mathbf{w}_d) = x_d + \delta_d(\mathbf{x},\mathbf{w}_d) \quad \text{with} \quad \delta_d(\mathbf{x},\mathbf{w}_d) = \sum_{m=0}^{M-1} w_{d,m}\phi_m(\mathbf{x})$$

$$\mathbf{w}_d = (w_{d,0}, \ldots w_{d,M-1})^{\mathrm{T}}$$

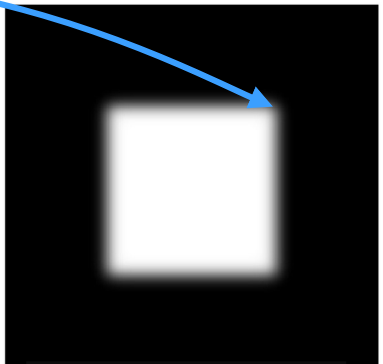
$$\mathbf{w} = (\mathbf{w}_1^{\mathrm{T}}, \ldots, \mathbf{w}_D^{\mathrm{T}})^{\mathrm{T}}$$

nonlinear basis functions

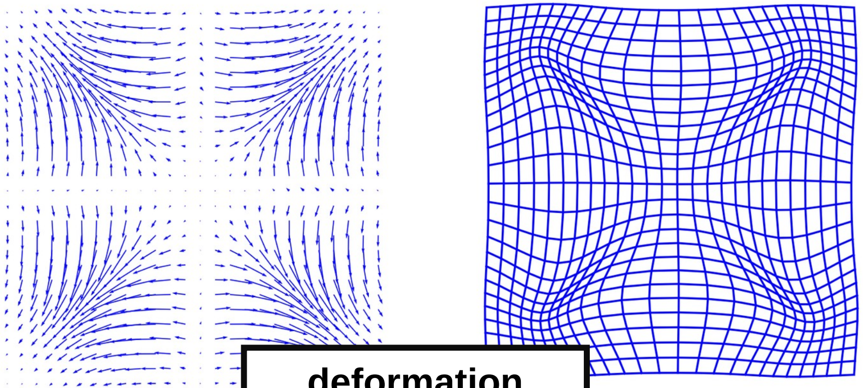
$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{x} + \boldsymbol{\delta}(\mathbf{x}, \mathbf{w})$$



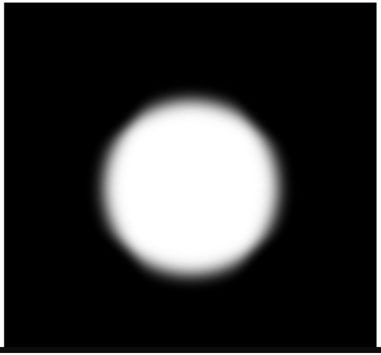
fixed image
$\mathcal{F}(\mathbf{x})$



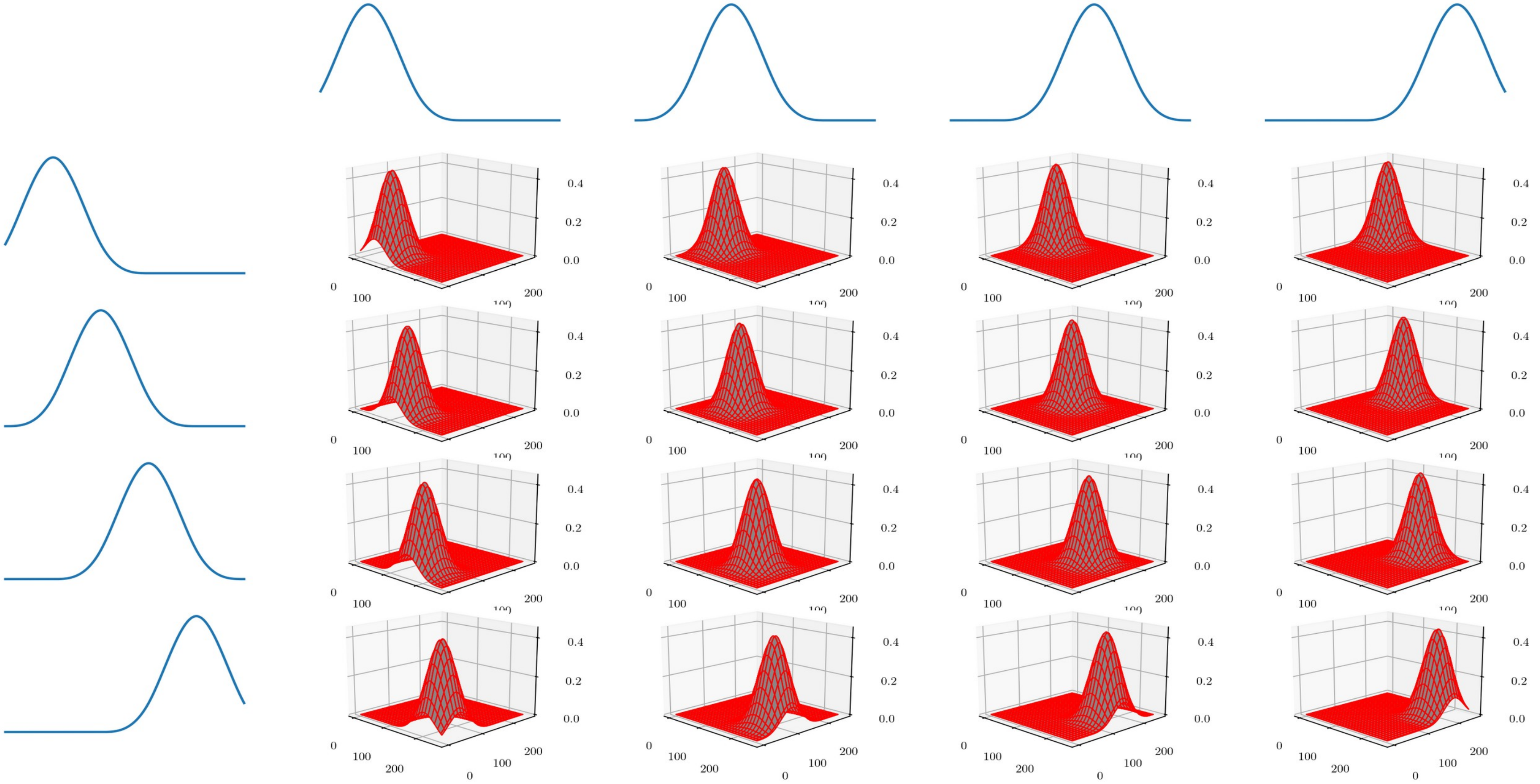
moving image
$\mathcal{M}(\mathbf{y})$



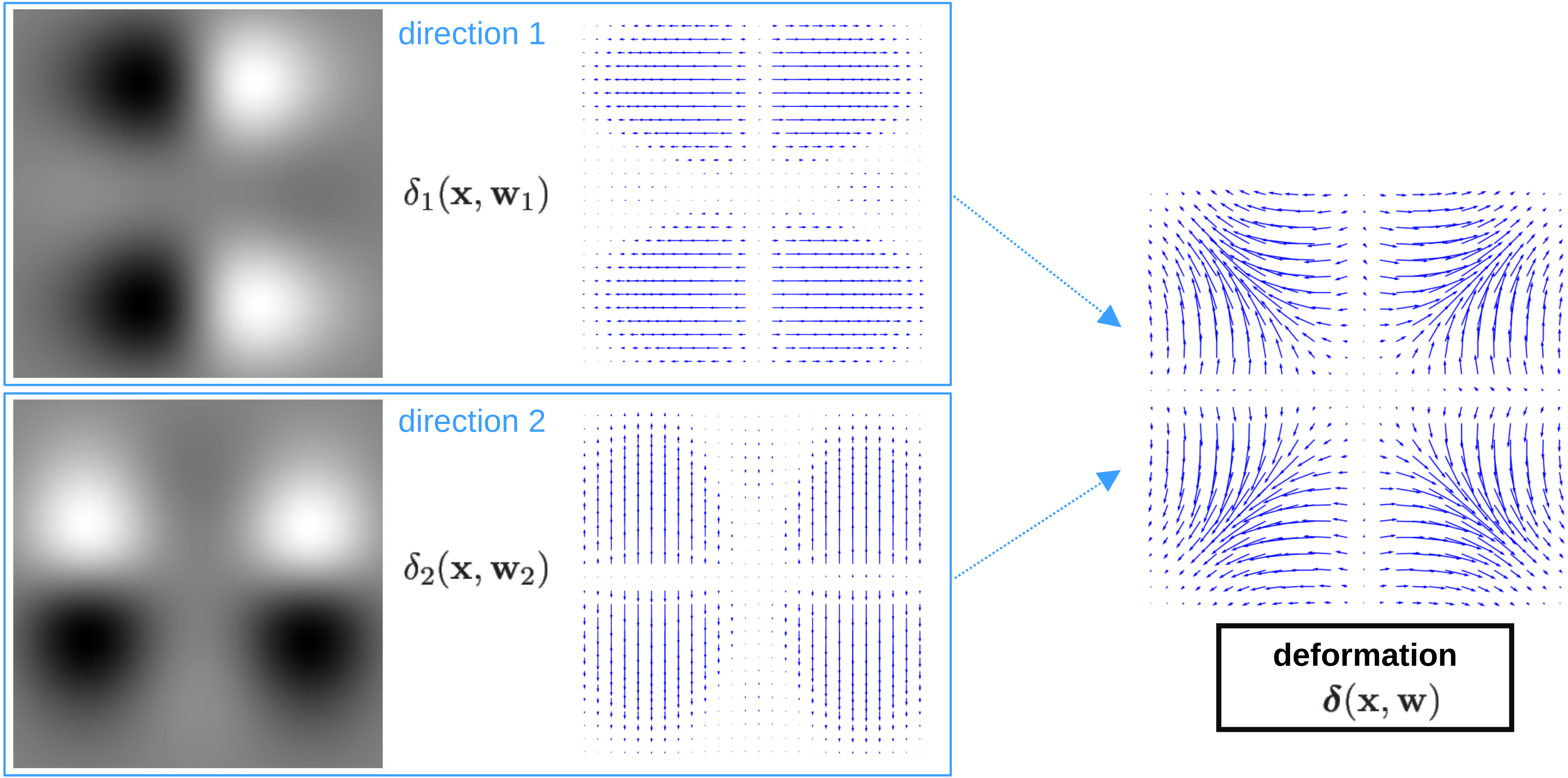
deformation
$\boldsymbol{\delta}(\mathbf{x}, \mathbf{w})$



interpolated moving image
$\mathcal{M}(\mathbf{y}(\mathbf{x}, \mathbf{w}))$



16 nonlinear basis functions



direction 1

$\delta_1(\mathbf{x}, \mathbf{w}_1)$

direction 2

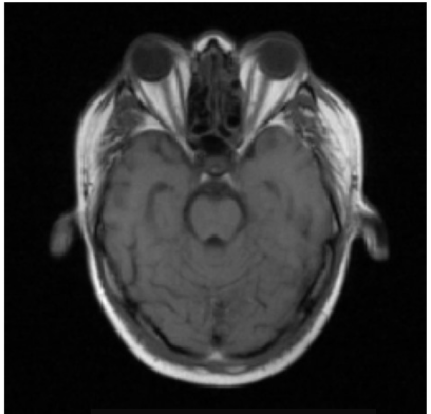
$\delta_2(\mathbf{x}, \mathbf{w}_2)$

deformation

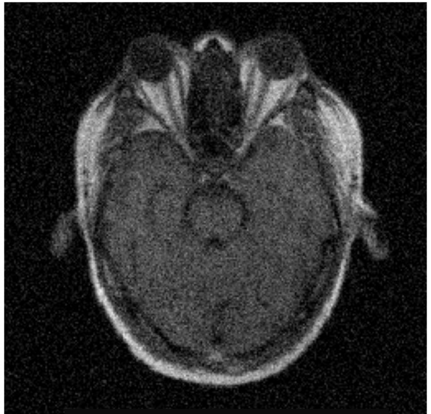
$\delta(\mathbf{x}, \mathbf{w})$

Focus on intra-modal registration

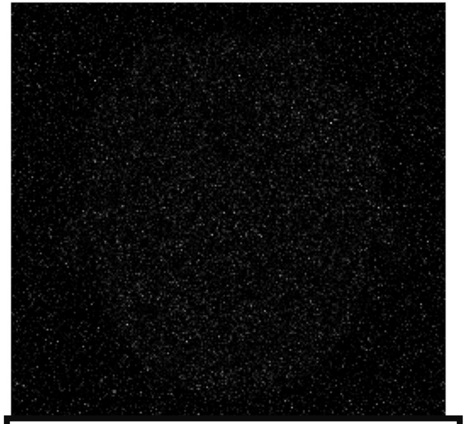
Images have similar intensity characteristics



$\mathcal{F}(\mathbf{x})$



$\mathcal{M}(\mathbf{y}(\mathbf{x},\mathbf{w}))$



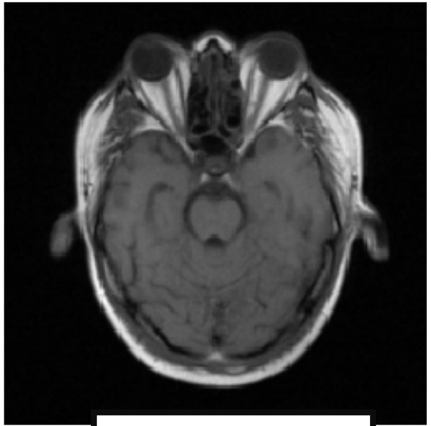
$[\mathcal{F}(\mathbf{x})-\mathcal{M}(\mathbf{y}(\mathbf{x},\mathbf{w}))]^2$

$$\boxed{E(\mathbf{w})=\sum_{n=1}^{N}[\mathcal{F}(\mathbf{x}_n)-\mathcal{M}(\mathbf{y}(\mathbf{x}_n,\mathbf{w}))]^2}$$

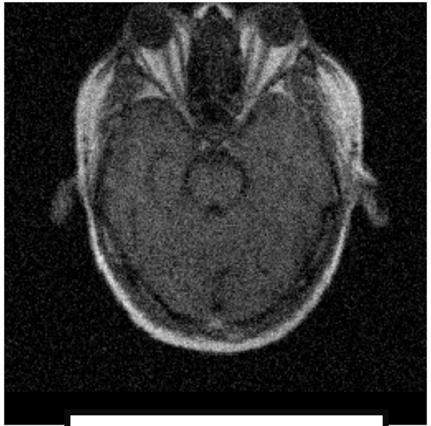
sum over all voxels

Focus on intra-modal registration

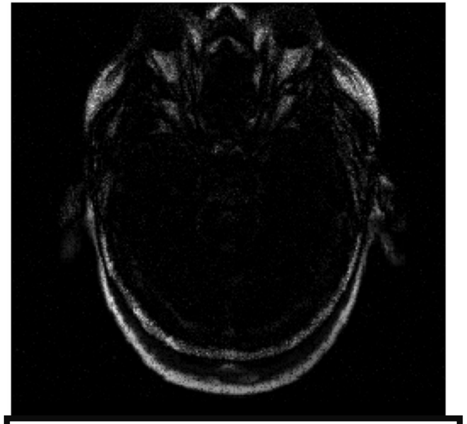
Images have similar intensity characteristics



$\mathcal{F}(\mathbf{x})$



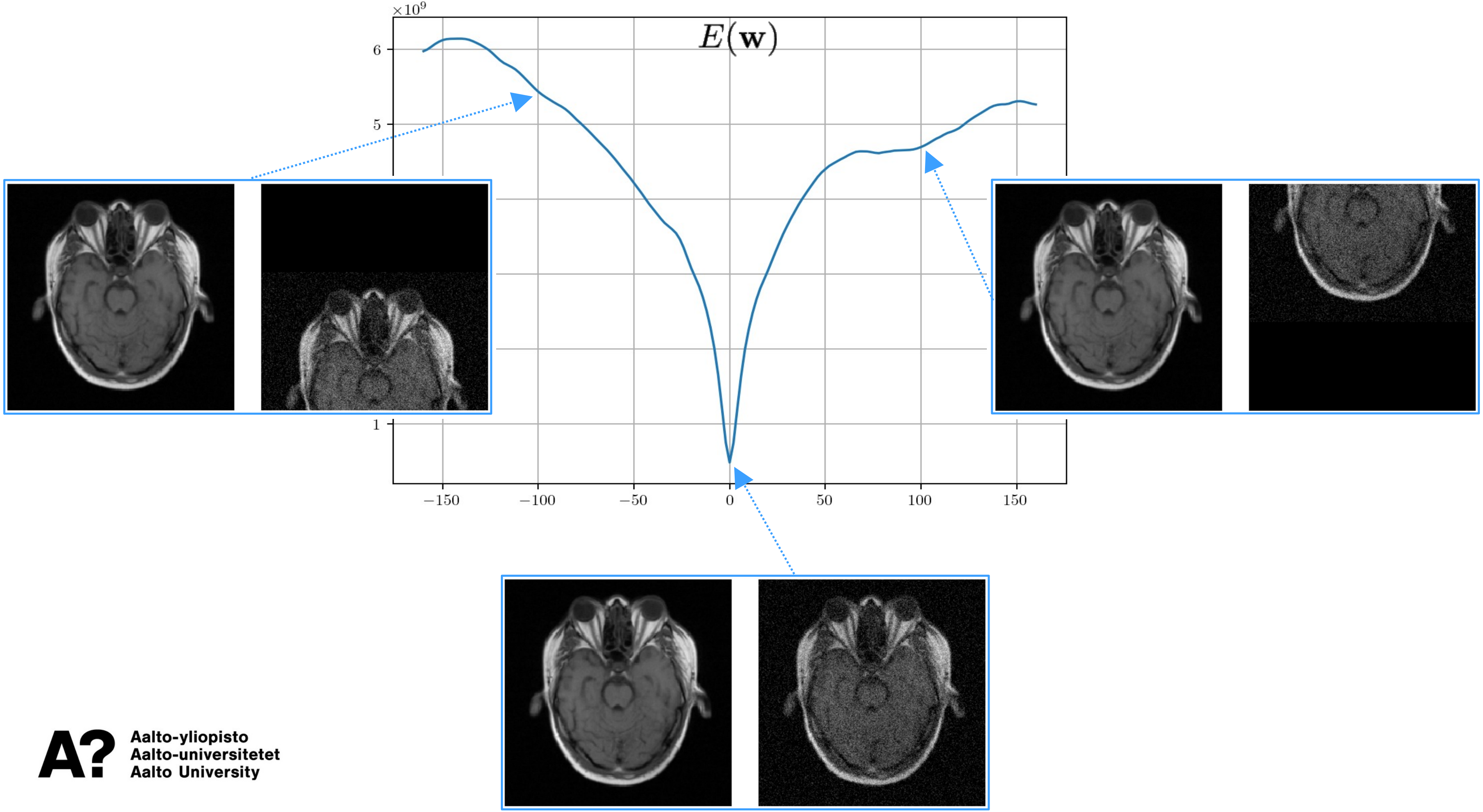
$\mathcal{M}(\mathbf{y}(\mathbf{x},\mathbf{w}))$



$[\mathcal{F}(\mathbf{x}) - \mathcal{M}(\mathbf{y}(\mathbf{x},\mathbf{w}))]^2$

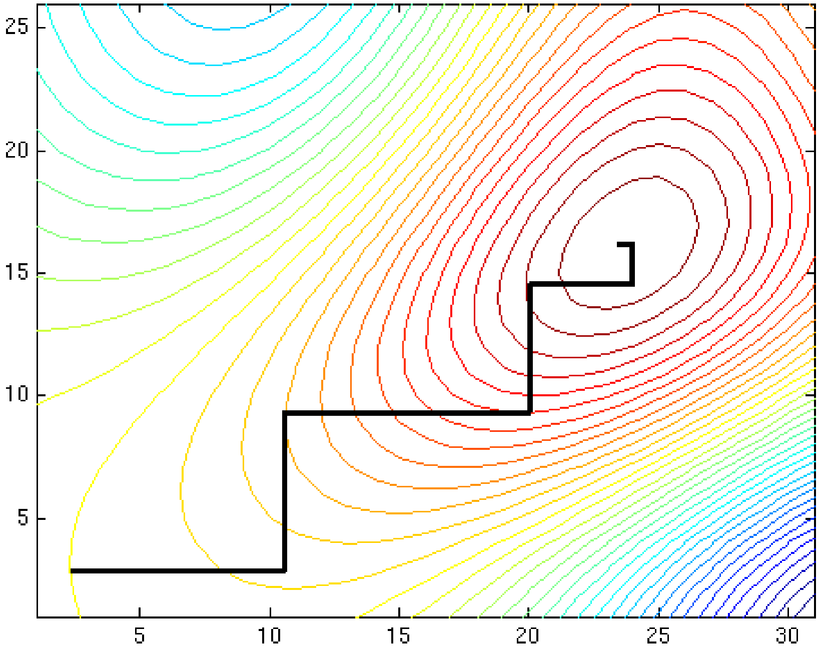
$$E(\mathbf{w}) = \sum_{n=1}^{N} [\mathcal{F}(\mathbf{x}_n) - \mathcal{M}(\mathbf{y}(\mathbf{x}_n,\mathbf{w}))]^2$$

sum over all voxels


×10^9
E(w)
6
5
1
−150
−100
−50
0
50
100
150

Numerical optimization

Find transformation parameters $\mathbf{w}$ **that minimize** $E(\mathbf{w})$



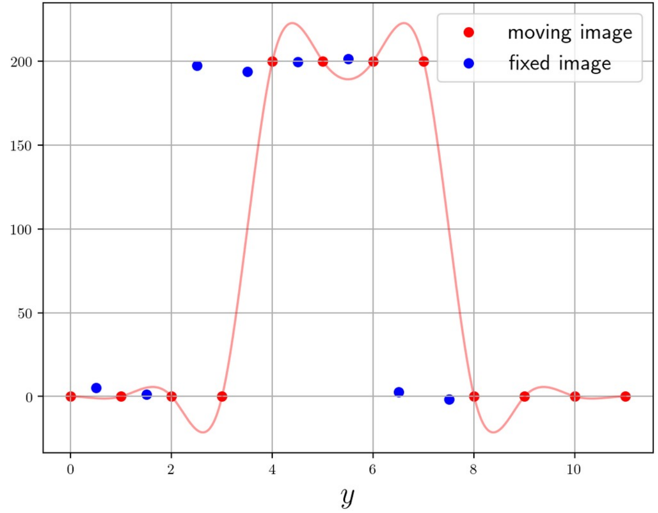
Gauss-Newton optimization

Toy example: 1D and translation only

- ✔ transformation model: $y(x,t) = x + t$

parameter to be optimized

- ✔ energy: $E(t) = \sum_{n=1}^{N} E_n(t)$ with $E_n(t) = [\mathcal{F}(x) - \mathcal{M}(y(x_n,t))]^2$



before optimization

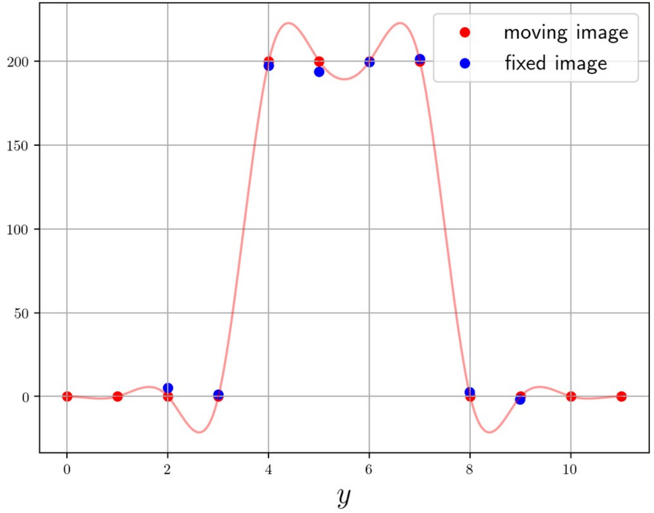
Gauss-Newton optimization

Toy example: 1D and translation only

- ✔ transformation model: $y(x,t) = x + \boxed{t}$

parameter to be optimized

- ✔ energy: $E(t) = \sum_{n=1}^{N} E_n(t)$ with $E_n(t) = [\mathcal{F}(x) - \mathcal{M}(y(x_n,t))]^2$

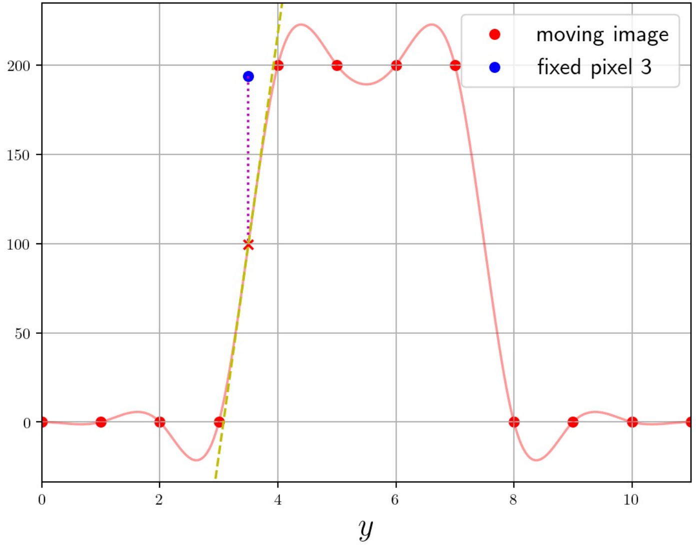


after optimization

Gauss-Newton optimization

$$g_n = \left.\frac{\mathrm{d}\mathcal{M}(y)}{\mathrm{d}y}\right|_{y=x_n+t}$$

Idea: for a *small* deviation ϵ around current estimate of t: $\mathcal{M}(y(x_n, t+\epsilon)) \simeq \mathcal{M}(y(x_n, t)) + g_n \cdot \epsilon$



- ✔ Energy: $E(t+\epsilon) = \sum_{n=1}^{N} E_n(t+\epsilon)$ with

$$E_n(t+\epsilon) = [\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t+\epsilon))]^2$$
$$\simeq [\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t)) - g_n \cdot \epsilon]^2$$

Task: what is ϵ minimizing $E(t+\epsilon)$?

Gauss-Newton optimization

Solution: standard linear regression!

$$\epsilon = (\boldsymbol{\psi}^T \boldsymbol{\psi})^{-1} \boldsymbol{\psi}^T \boldsymbol{\tau}$$

Now update t **:**

$$t \leftarrow t + \epsilon$$

where $\boldsymbol{\tau} = \begin{pmatrix} \tau_1 \\ \vdots \\ \tau_N \end{pmatrix}$ with $\tau_n = \mathcal{F}(x_n) - \mathcal{M}(y(x_n, t))$

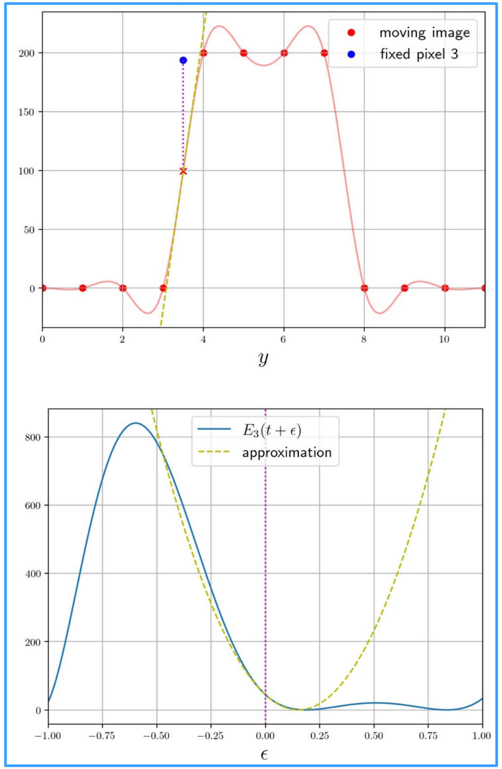
and $\boldsymbol{\psi} = \begin{pmatrix} g_1 \\ \vdots \\ g_N \end{pmatrix}$

one basis function

initialization

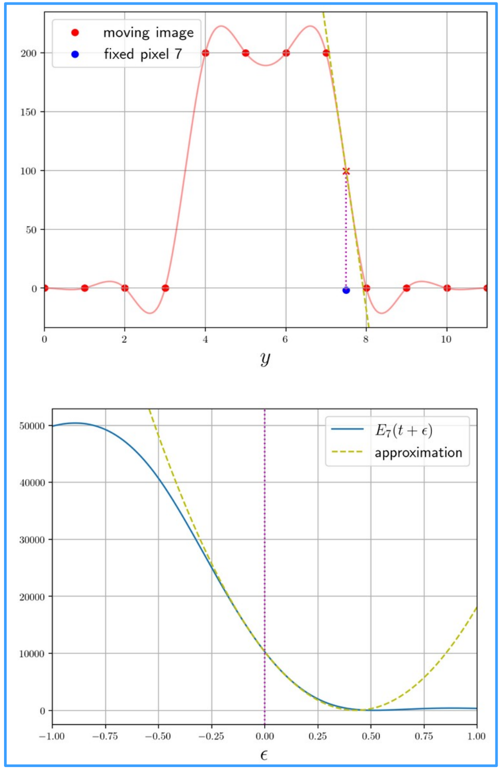
pixel 3

… +



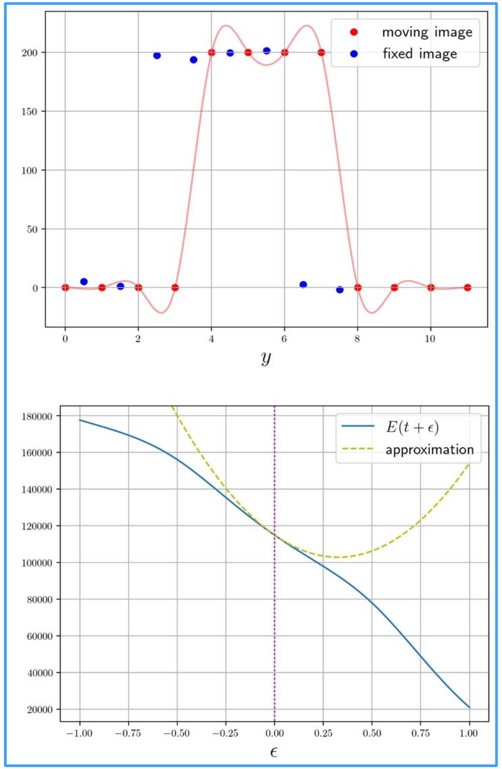
… + …

pixel 7



+ … =

all pixels



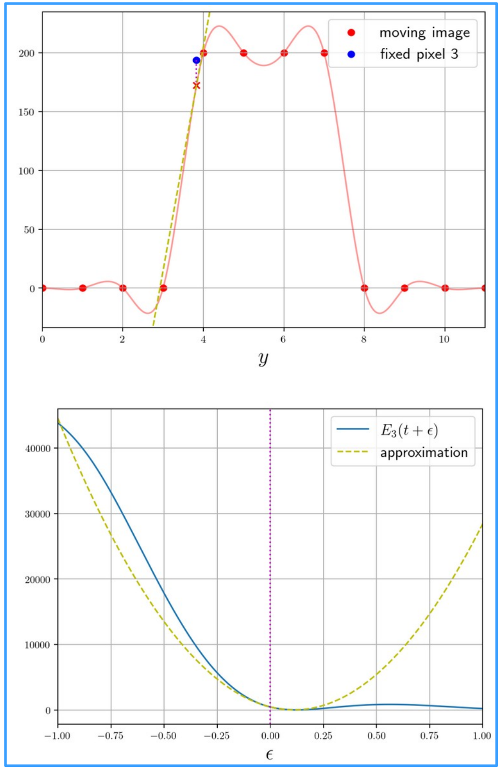
✔ Energy: $E(t+\epsilon) = \sum_{n=1}^{N} E_n(t+\epsilon)$ with

$$E_n(t+\epsilon) = \left[\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t+\epsilon))\right]^2$$
$$\simeq \left[\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t)) - g_n \cdot \epsilon\right]^2$$

after 1 iteration

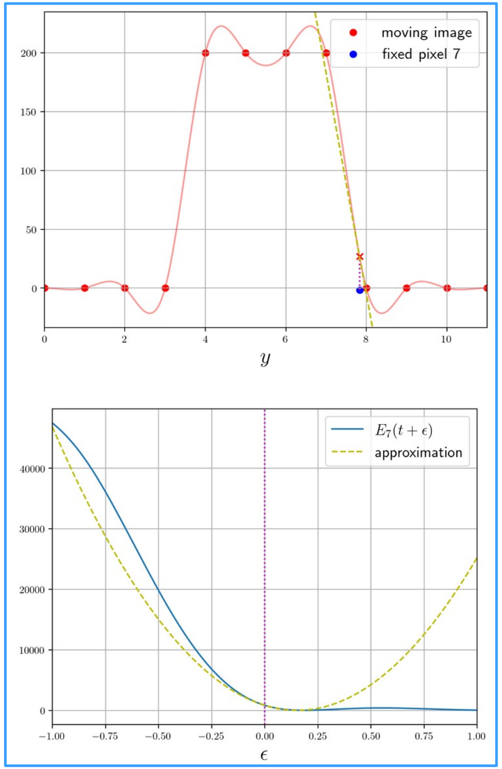
… +

pixel 3



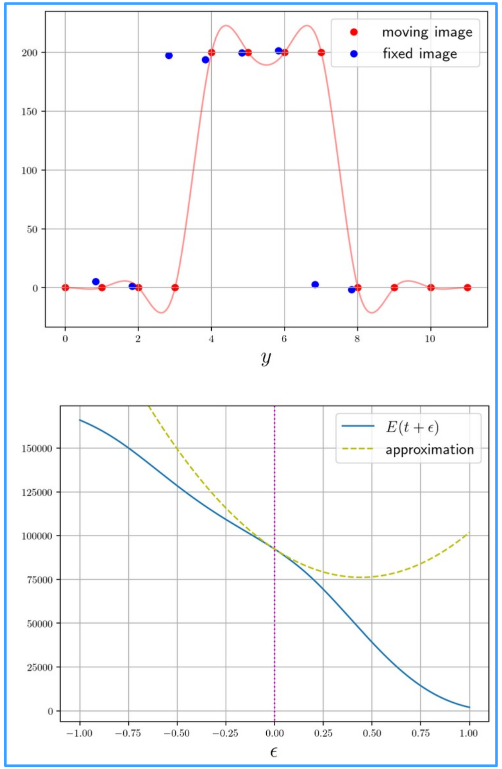
… + …

pixel 7



+ … =

all pixels



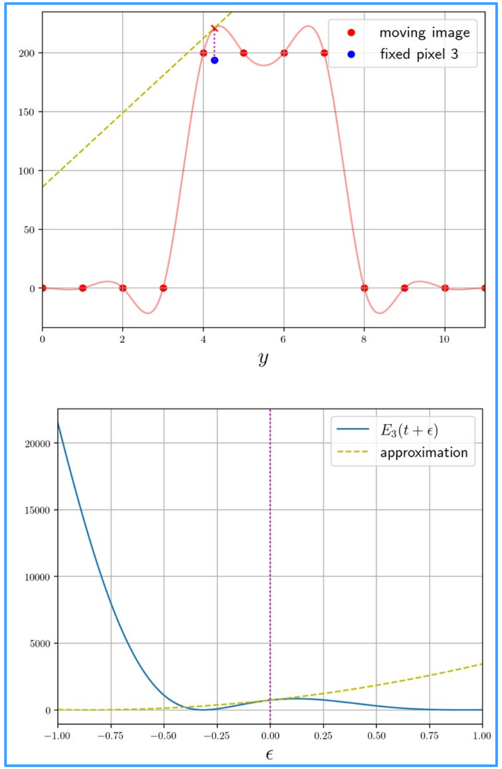
✔ Energy: $E(t+\epsilon) = \sum_{n=1}^{N} E_n(t+\epsilon)$ with

$$\begin{aligned} E_n(t+\epsilon) &= \left[\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t+\epsilon))\right]^2 \\ &\simeq \left[\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t)) - g_n \cdot \epsilon\right]^2 \end{aligned}$$

after 2 iterations

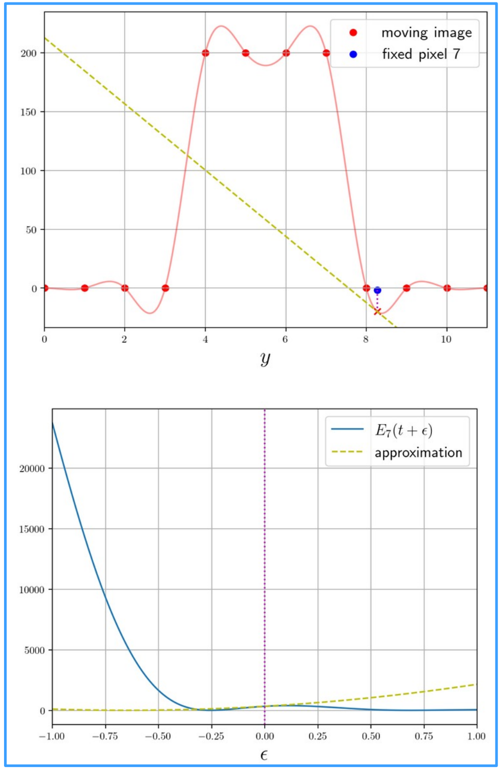
… +

pixel 3



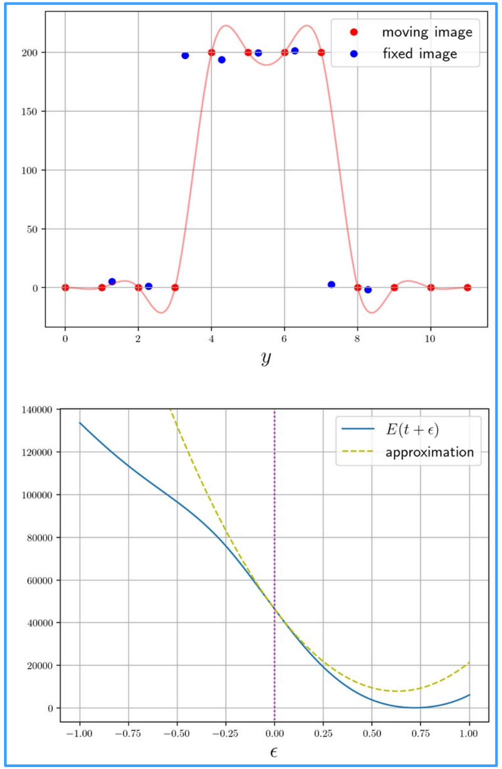
… + …

pixel 7



+ … =

all pixels



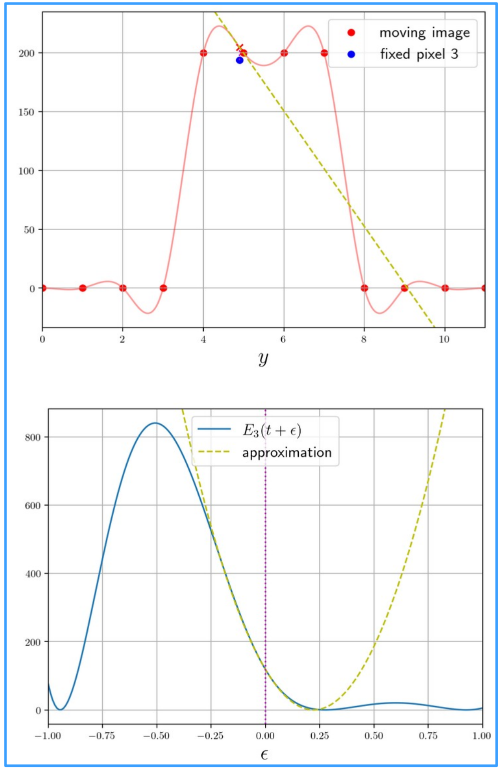
✔ Energy: $E(t+\epsilon) = \sum_{n=1}^{N} E_n(t+\epsilon)$ with $E_n(t+\epsilon) = \left[\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t+\epsilon))\right]^2$

$$\simeq \left[\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t)) - g_n \cdot \epsilon\right]^2$$

after 3 iterations

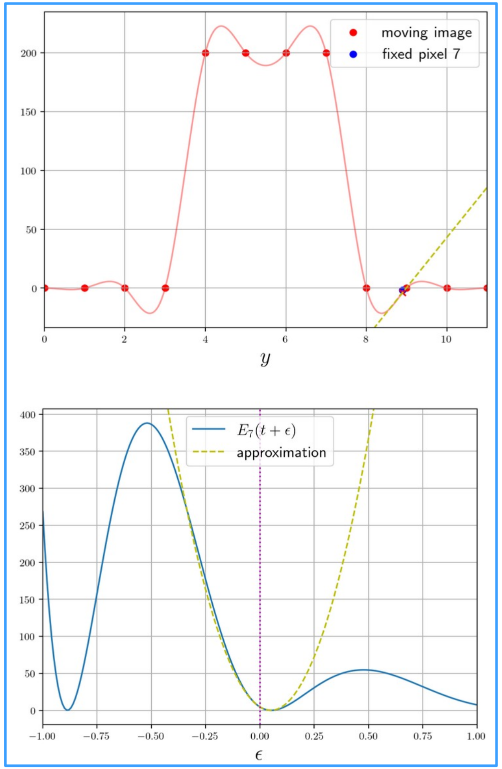
pixel 3

… +



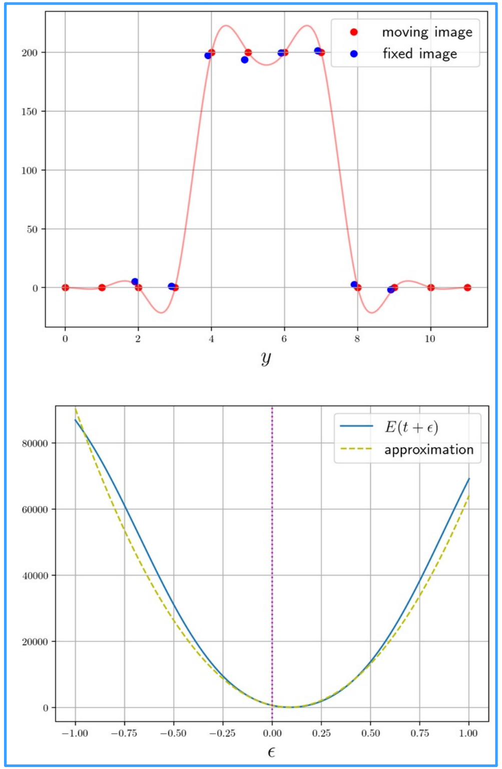
… + …

pixel 7



+ … =

all pixels



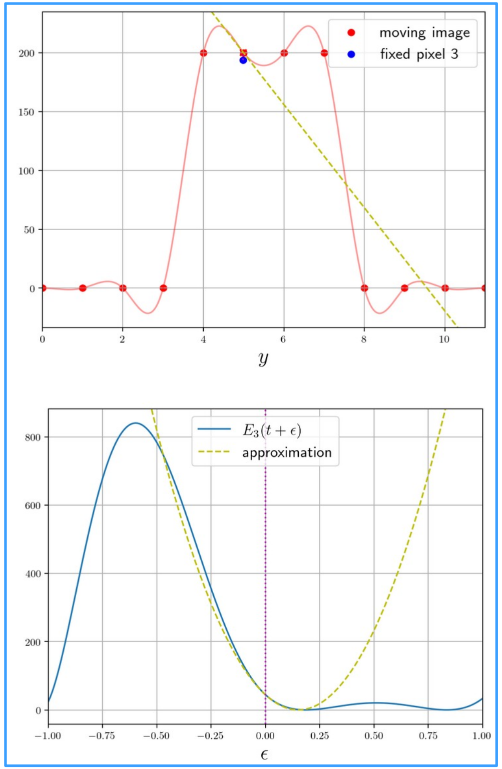
✔ Energy: $E(t+\epsilon)=\sum_{n=1}^{N} E_n(t+\epsilon)$ with $E_n(t+\epsilon) = [\mathcal{F}(x_n) - \mathcal{M}(y(x_n,t+\epsilon))]^2$

$$\simeq [\mathcal{F}(x_n) - \mathcal{M}(y(x_n,t)) - g_n \cdot \epsilon]^2$$

after 4 iterations

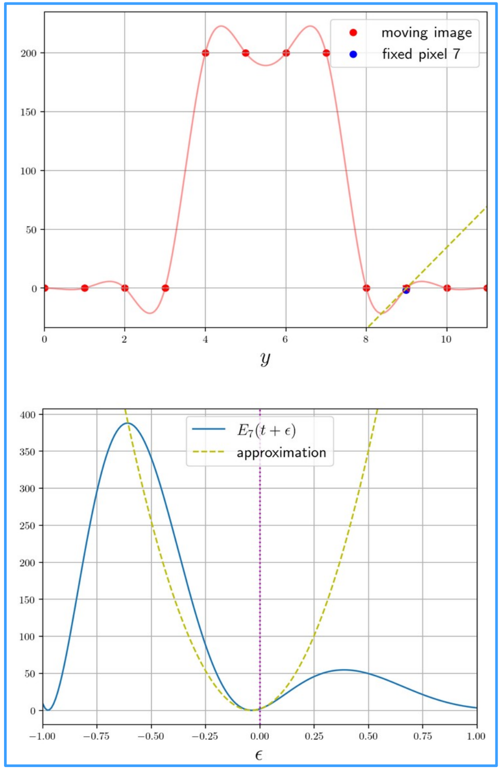
… +

pixel 3



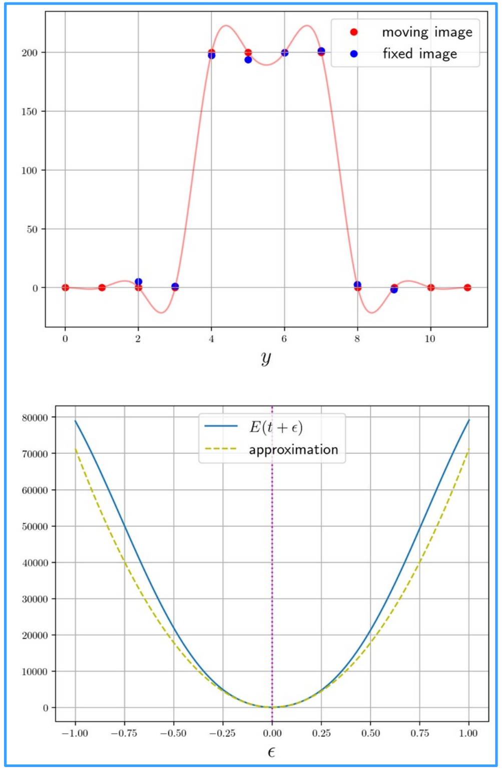
… + …

pixel 7



+ … =

all pixels



✔ Energy: $E(t+\epsilon) = \sum_{n=1}^{N} E_n(t+\epsilon)$ with

$$\begin{aligned} E_n(t+\epsilon) &= \left[\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t+\epsilon))\right]^2 \\ &\simeq \left[\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t)) - g_n \cdot \epsilon\right]^2 \end{aligned}$$

Gauss-Newton optimization

Solution: standard linear regression!

$$\epsilon = (\boldsymbol{\psi}^T\boldsymbol{\psi})^{-1}\boldsymbol{\psi}^T\boldsymbol{\tau}$$

where $\boldsymbol{\tau} = \begin{pmatrix} \tau_1 \\ \vdots \\ \tau_N \end{pmatrix}$ with $\tau_n = \mathcal{F}(x_n) - \mathcal{M}(y(x_n, t))$

and $\boldsymbol{\psi} = \begin{pmatrix} g_1 \\ \vdots \\ \underbrace{g_N}_{\text{one basis function}} \end{pmatrix}$

Now update t **:**

$$t \leftarrow t + \epsilon$$

Gauss-Newton optimization

Solution: standard linear regression!

$$\epsilon = (\boldsymbol{\psi}^T\boldsymbol{\psi})^{-1}\boldsymbol{\psi}^T\boldsymbol{\tau}$$

Now update t **:**

$$t \leftarrow t + \epsilon$$

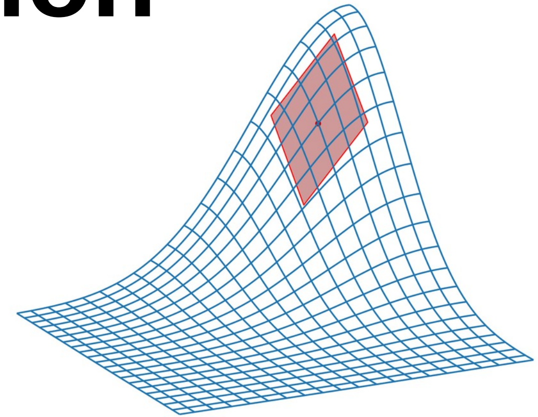
where $\boldsymbol{\tau} = \begin{pmatrix} \tau_1 \\ \vdots \\ \tau_N \end{pmatrix}$ with $\tau_n = \mathcal{F}(x_n) - \mathcal{M}(y(x_n, t))$

and $\boldsymbol{\psi} = \begin{pmatrix} g_1 \\ \vdots \\ g_N \end{pmatrix} = \underbrace{\begin{pmatrix} g_1 & & \\ & \ddots & \\ & & g_N \end{pmatrix}}_{\mathbf{G}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$

one basis function

Aalto-yliopisto
Aalto-universitetet
Aalto University

Gauss-Newton optimization



In real situations, exactly the same idea but:

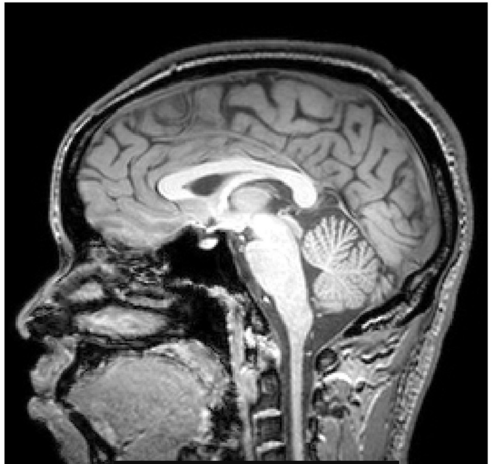
- ✔ $D > 1$ spatial dimensions
- ✔ $M > 1$ basis functions

$$\mathcal{M}(\mathbf{y}(\mathbf{x}_n, \mathbf{w} + \boldsymbol{\epsilon})) \simeq \mathcal{M}(\mathbf{y}(\mathbf{x}_n, \mathbf{w})) + \sum_{d=1}^{D} \sum_{m=0}^{M} \left(g_{d,n}\phi_m(\mathbf{x}_n)\right)\epsilon_{d,m}$$

Solution: $\boldsymbol{\epsilon} = \left(\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\tau}$ where $\boldsymbol{\Psi} = \left(\ \mathbf{G}_1\boldsymbol{\Phi} \mid \cdots \mid \mathbf{G}_D\boldsymbol{\Phi}\ \right)$

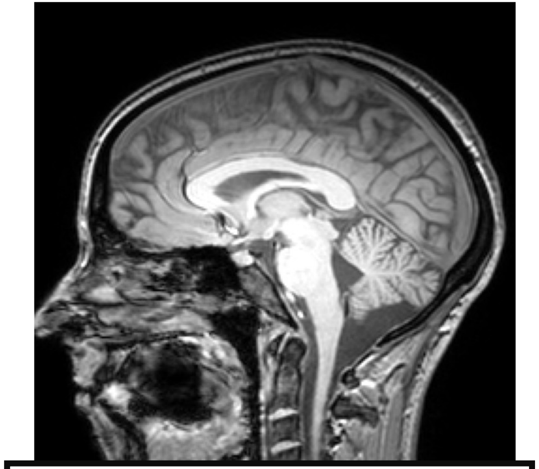
and $\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \dots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \dots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$

initialization



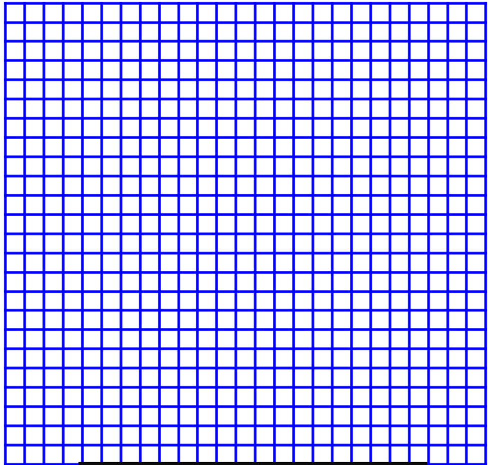
fixed image

$\mathcal{F}(\mathbf{x})$



interpolated moving image

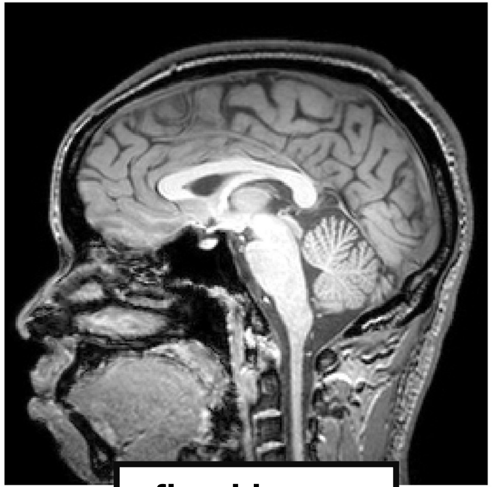
$\mathcal{M}(\mathbf{y}(\mathbf{x},\mathbf{w}))$



deformation

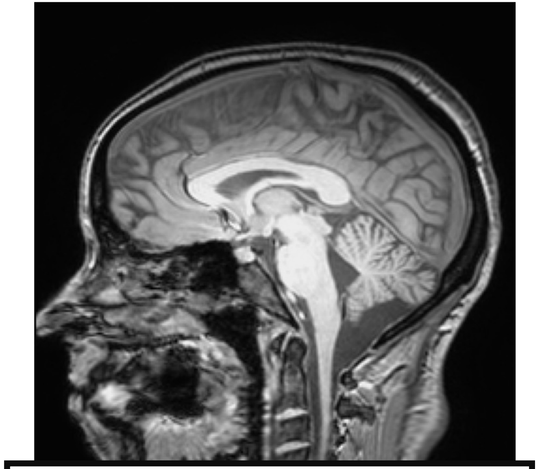
$\delta(\mathbf{x},\mathbf{w})$

after 10 iterations



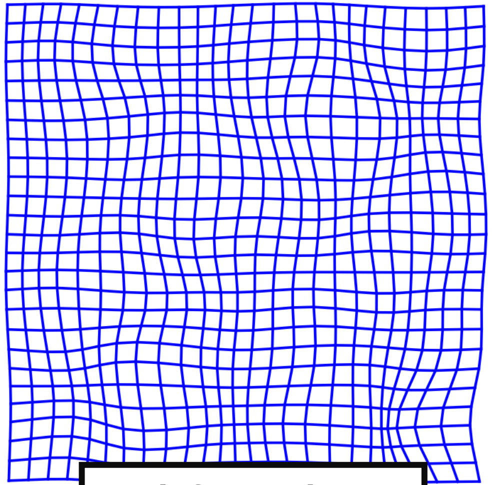
fixed image

$\mathcal{F}(\mathbf{x})$



interpolated moving image

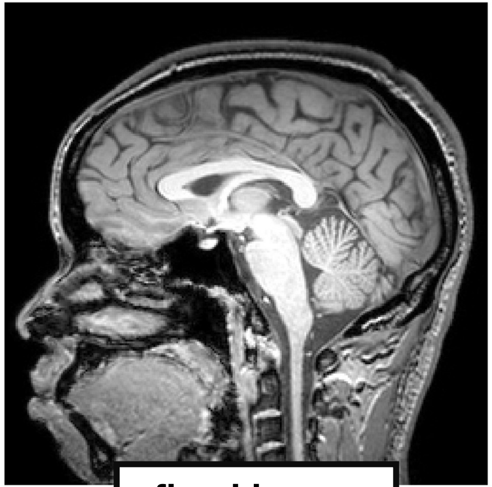
$\mathcal{M}(\mathbf{y}(\mathbf{x},\mathbf{w}))$



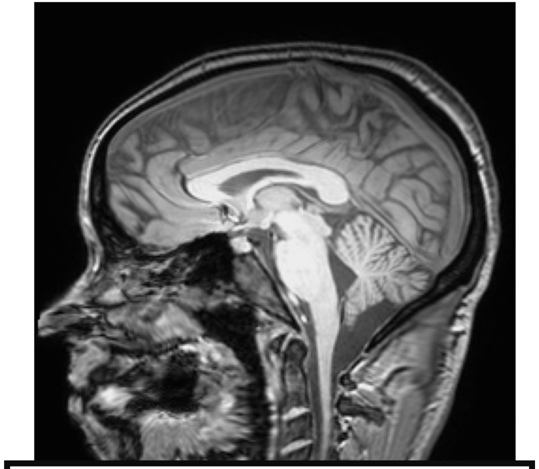
deformation

$\boldsymbol{\delta}(\mathbf{x},\mathbf{w})$

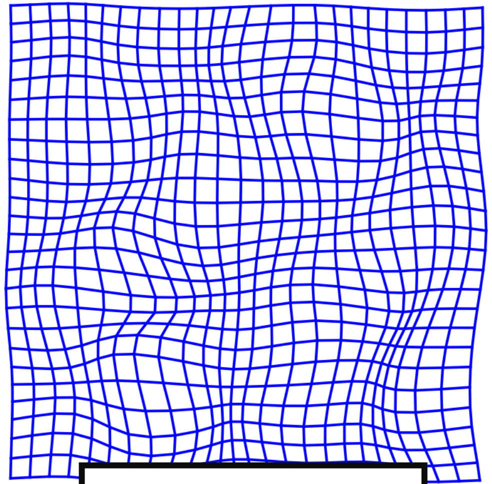
after 30 iterations



fixed image
$\mathcal{F}(\mathbf{x})$

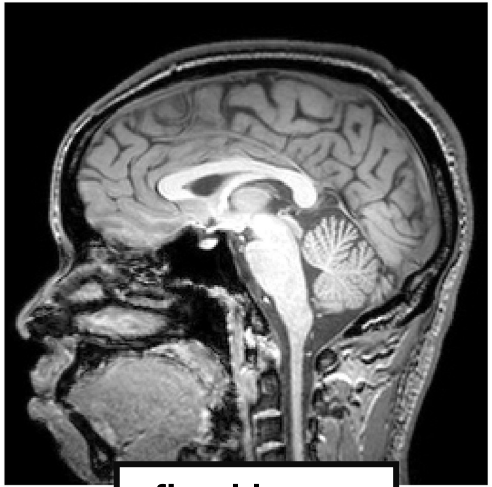


interpolated moving image
$\mathcal{M}(\mathbf{y}(\mathbf{x},\mathbf{w}))$

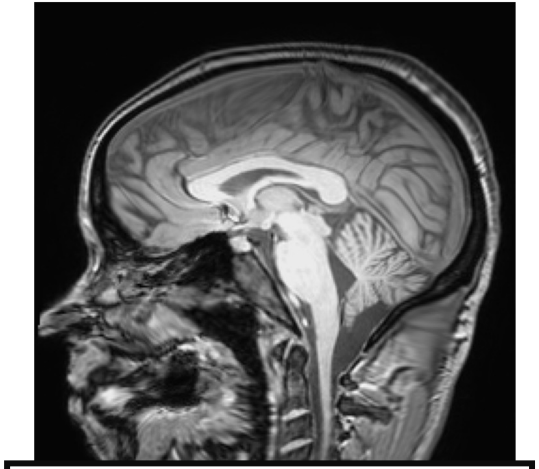


deformation
$\boldsymbol{\delta}(\mathbf{x},\mathbf{w})$

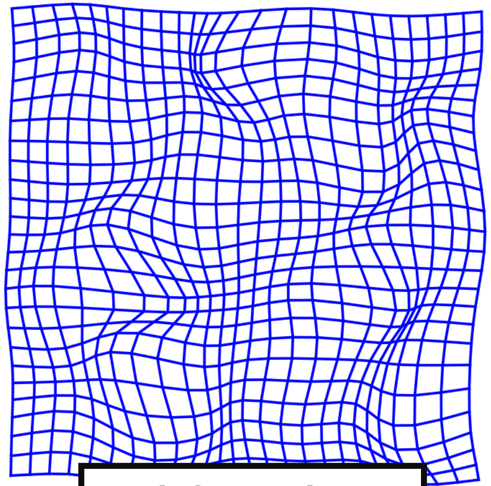
after convergence



fixed image
$\mathcal{F}(\mathbf{x})$

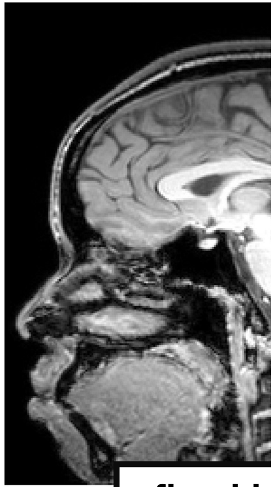


interpolated moving image
$\mathcal{M}(\mathbf{y}(\mathbf{x},\mathbf{w}))$

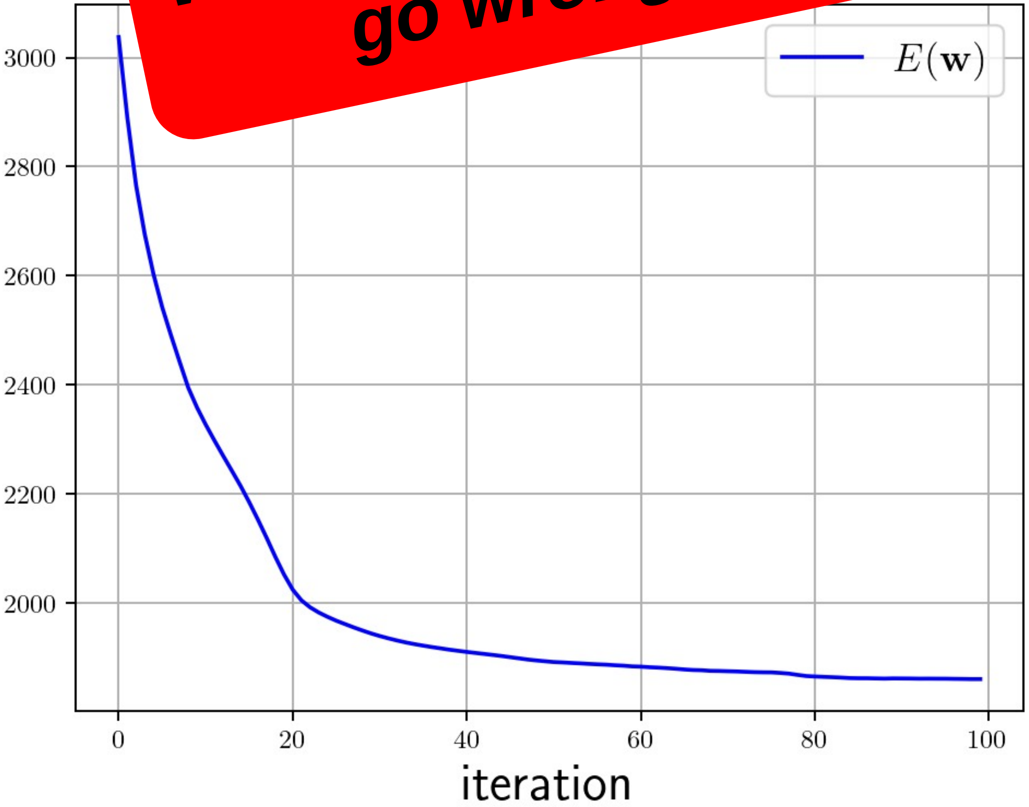


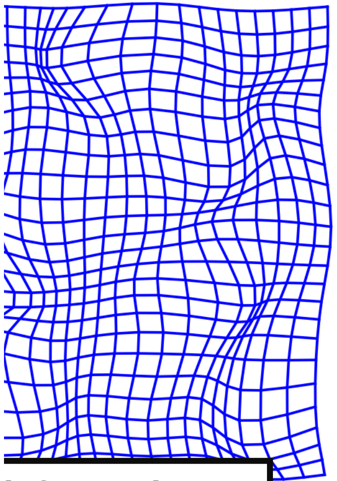
deformation
$\boldsymbol{\delta}(\mathbf{x},\mathbf{w})$

What could possibly go wrong?



fixed i
$\mathcal{F}(\mathbf{x}$





deformation
$\boldsymbol{\delta}(\mathbf{x}, \mathbf{w})$

Aalto-yliopisto
Aalto-universitetet
Aalto University

Safety rails...

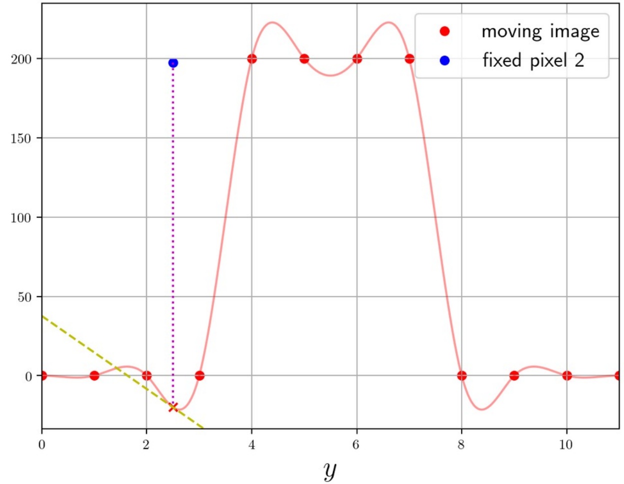
We have assumed that ϵ is small

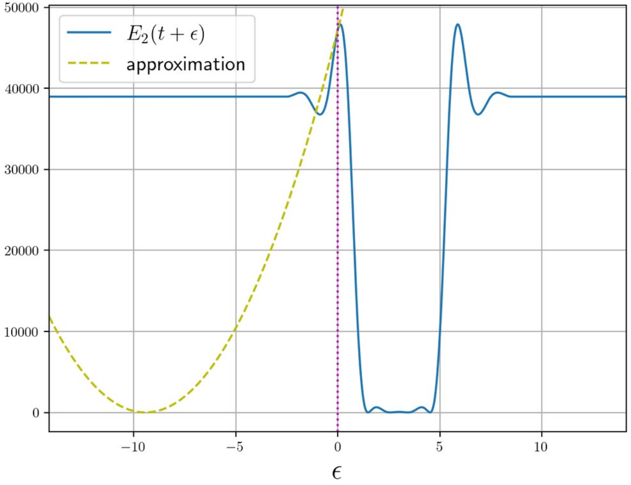
- ✔ What if it isn't? Energy $E(\mathbf{w})$ could go *up* instead of *down*!

- ✔ Solution: Levenberg–Marquardt

$$\boldsymbol{\epsilon} = \left(\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\Psi} + \lambda\mathbf{I}\right)^{-1}\boldsymbol{\Psi}^{\mathrm{T}}\boldsymbol{\tau}$$

tunable


moving image
fixed pixel 2
y


$E_2(t+\epsilon)$
approximation
ϵ