

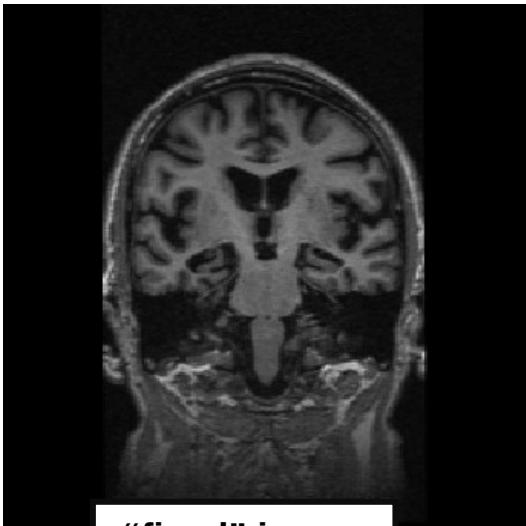
Nonlinear Registration

A”

Aalto-yliopisto
Aalto-universitetet
Aalto University

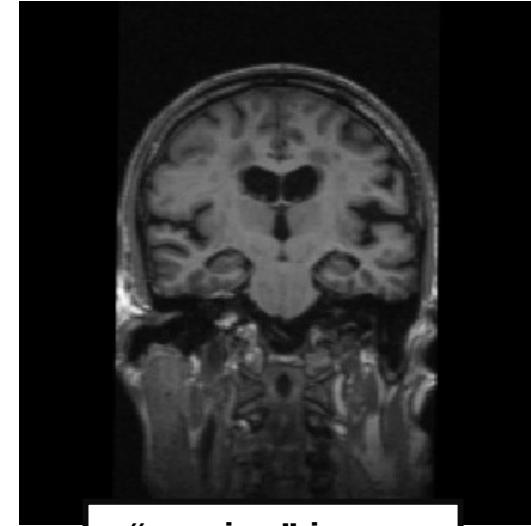
Medical Image Analysis
Koen Van Leemput

Spatial transformations



“fixed” image

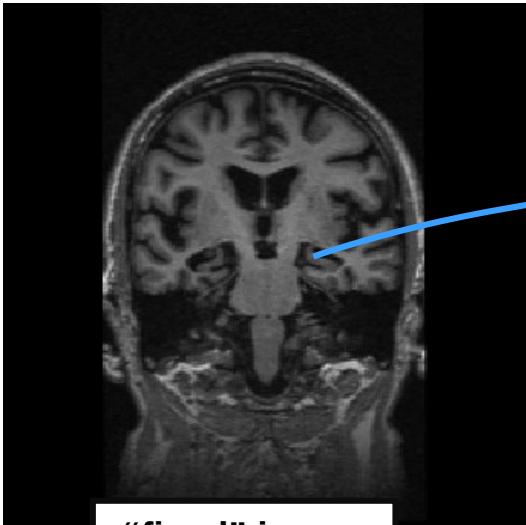
$$\mathbf{x} = (x_1, \dots, x_D)^T$$



“moving” image

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

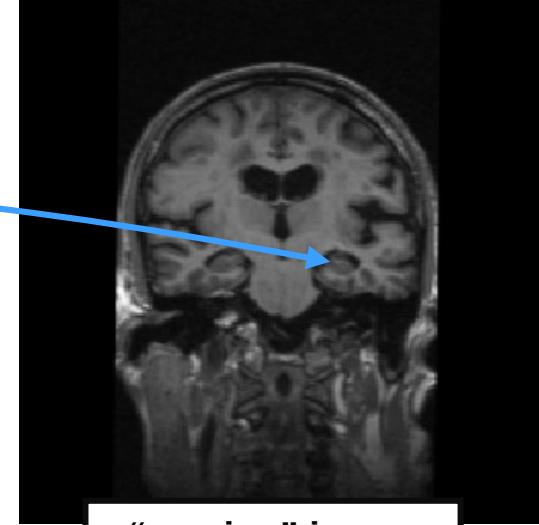
Spatial transformations



"fixed" image

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

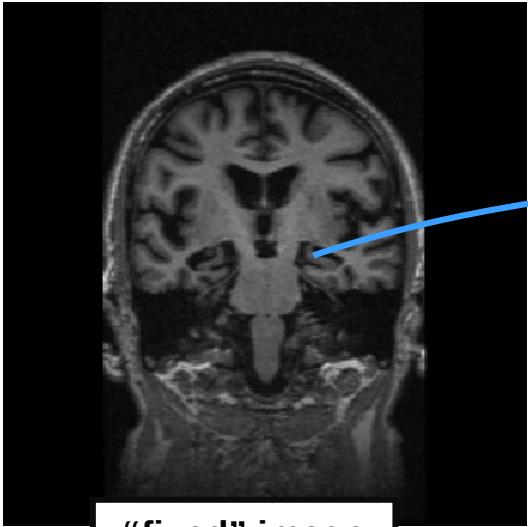
$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \begin{pmatrix} y_1(\mathbf{x}, \mathbf{w}) \\ \vdots \\ y_D(\mathbf{x}, \mathbf{w}) \end{pmatrix}$$



"moving" image

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

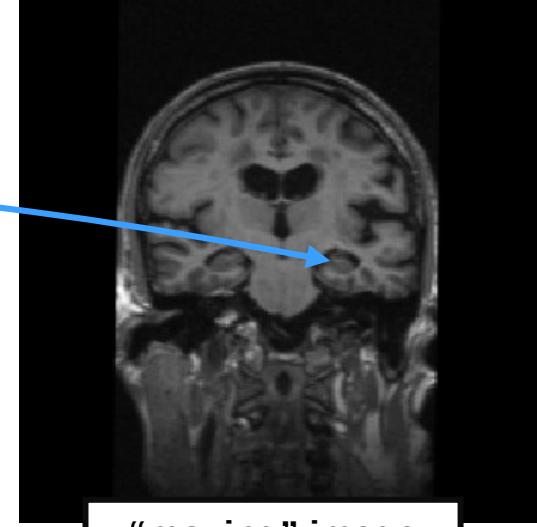
Spatial transformations



"fixed" image

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

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



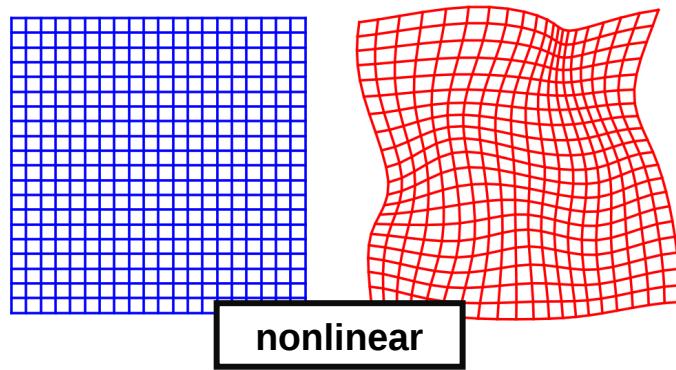
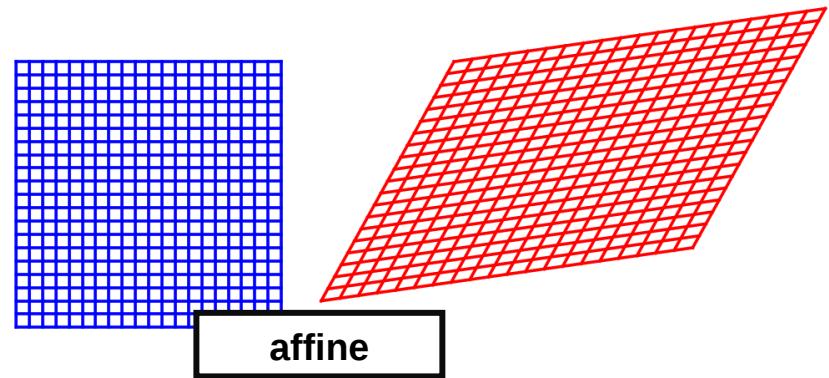
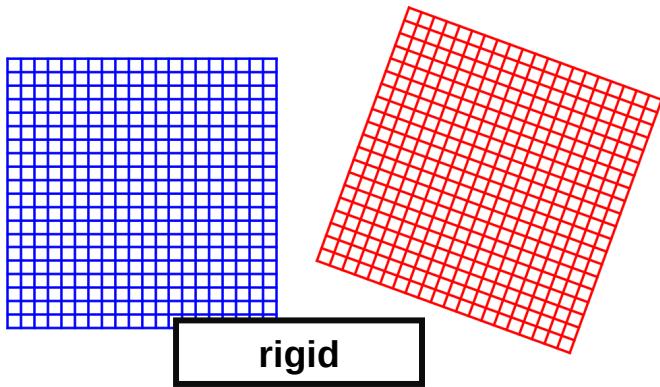
"moving" image

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

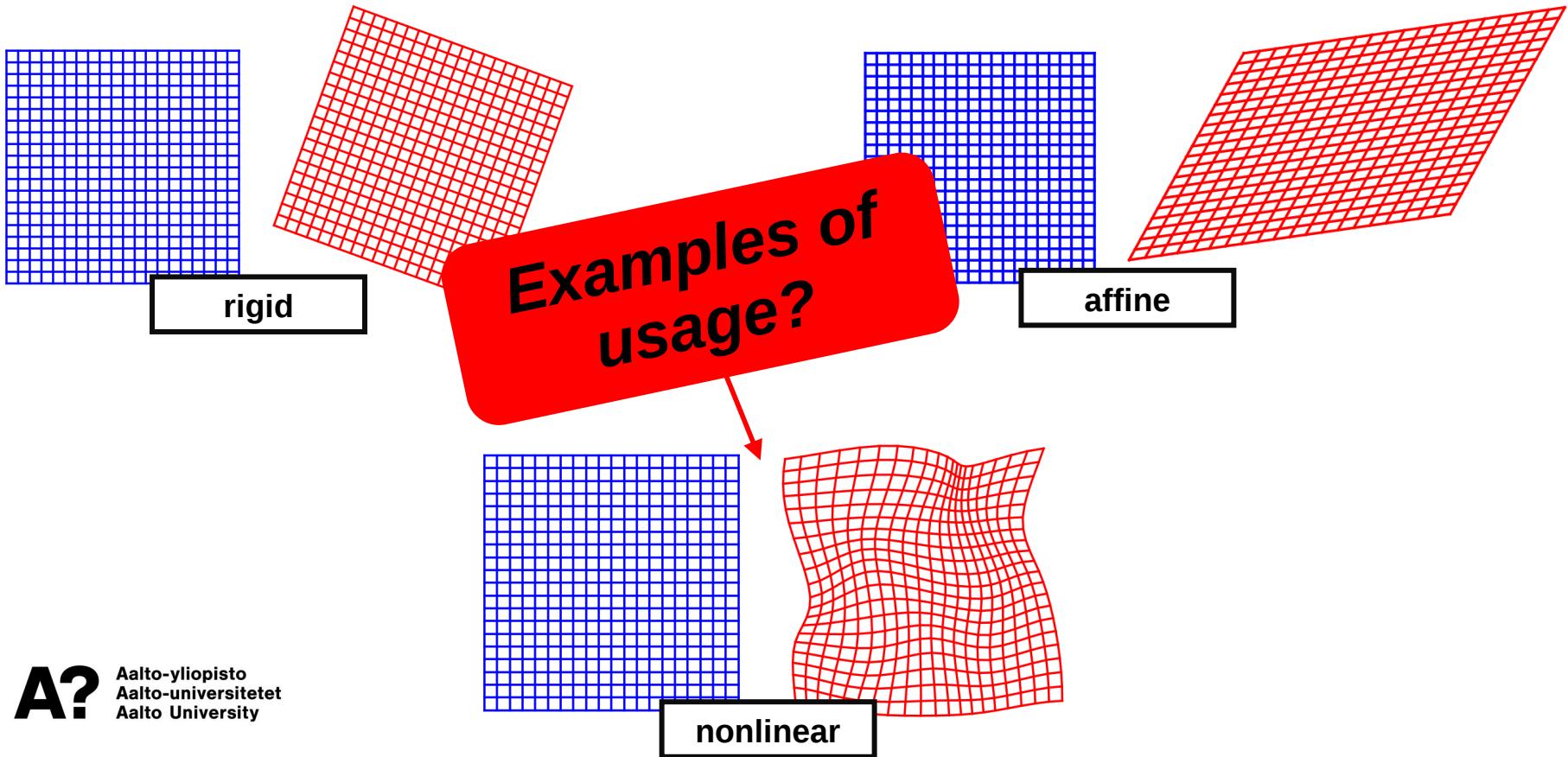
$$y_d(\mathbf{x}, \mathbf{w})$$

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



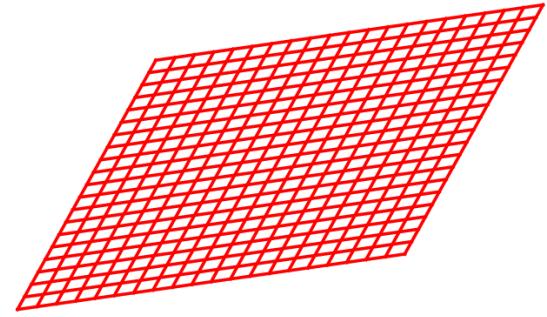
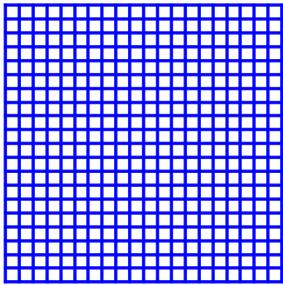
Spatial transformations



Affine transformation

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

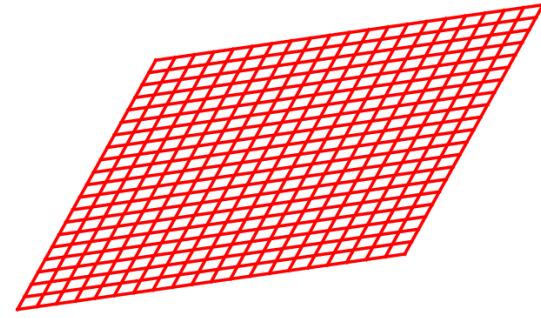
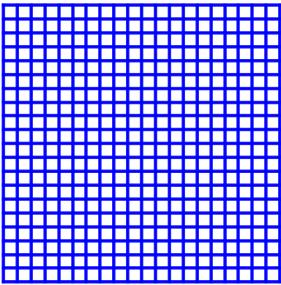
$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \quad \text{and} \quad \mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$



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

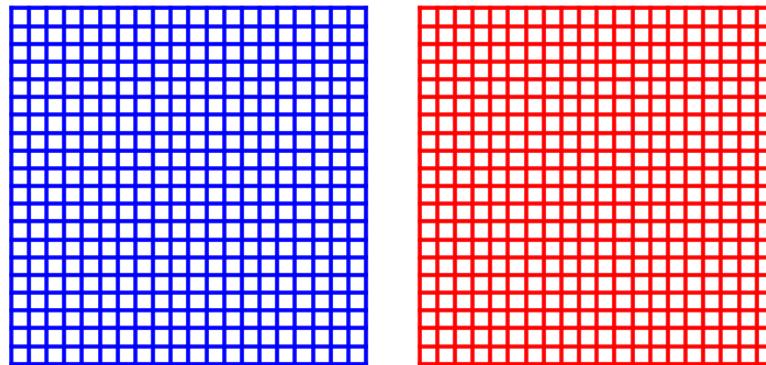
$$y_d(\mathbf{x}, \mathbf{w}_d) = t_d + a_{d,1}x_1 + \dots + a_{d,D}x_D$$

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

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

Affine transformation

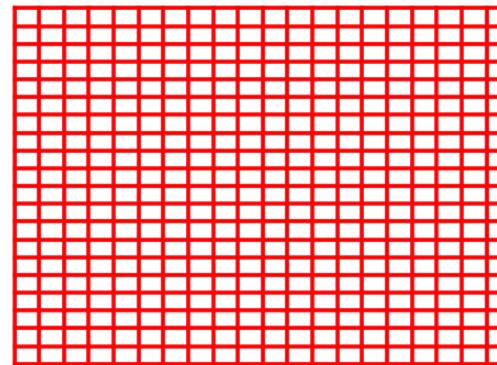
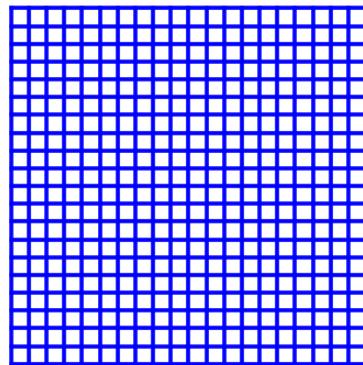
$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$



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

Affine transformation

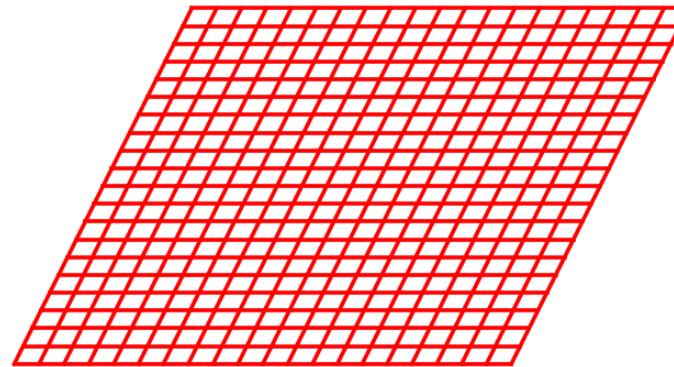
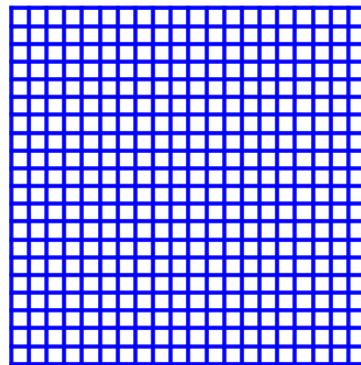
$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$



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

Affine transformation

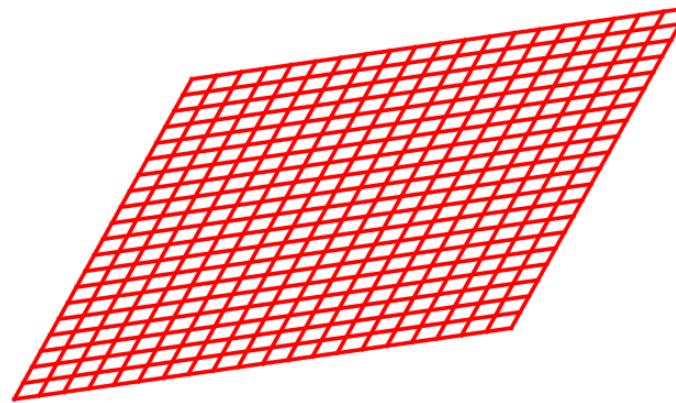
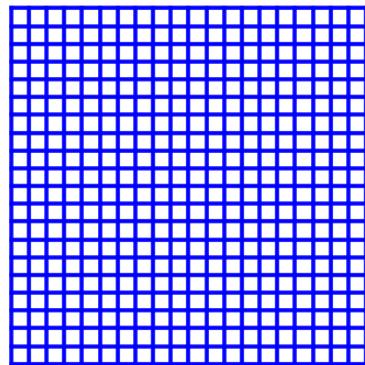
$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$



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

Affine transformation

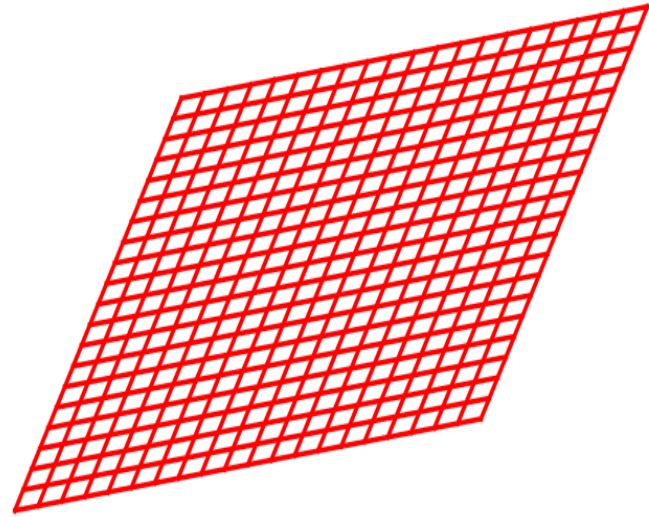
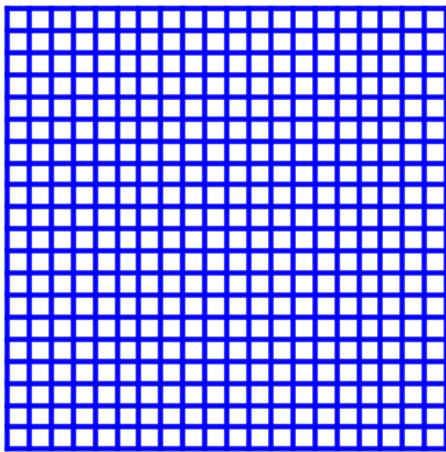
$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{A}\mathbf{x} + \mathbf{t}$$



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

Affine (linear) transformation...

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



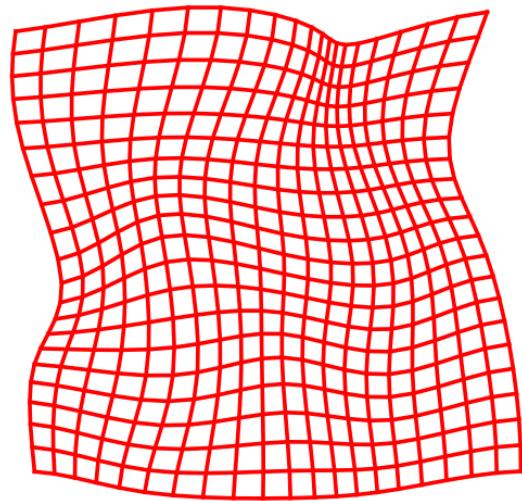
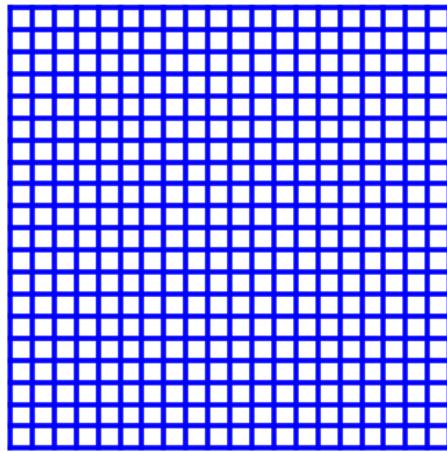
$$y_d(\mathbf{x}, \mathbf{w}_d) = t_d + a_{d,1}x_1 + \dots + a_{d,D}x_D$$

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

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

...vs. *nonlinear* transformation

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



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

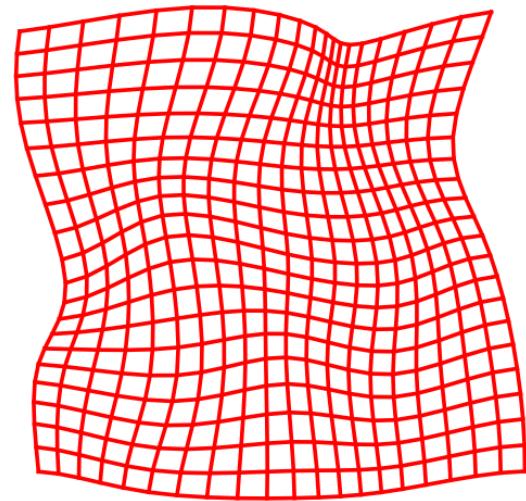
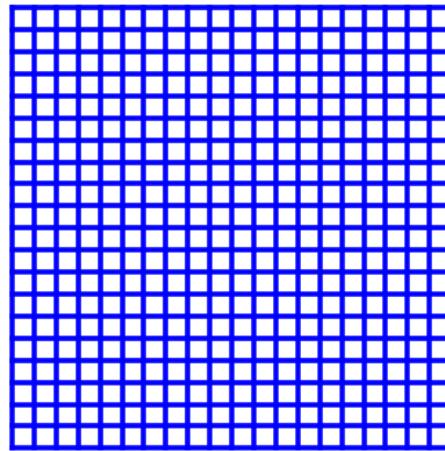
$$\mathbf{w} = (\mathbf{w}_1^T, \dots, \mathbf{w}_D^T)^T$$

$$\delta_d(\mathbf{x}, \mathbf{w}_d) = \sum_{m=0}^{M-1} w_{d,m} \phi_m(\mathbf{x})$$

...vs. *nonlinear* transformation

“residual deformation”

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

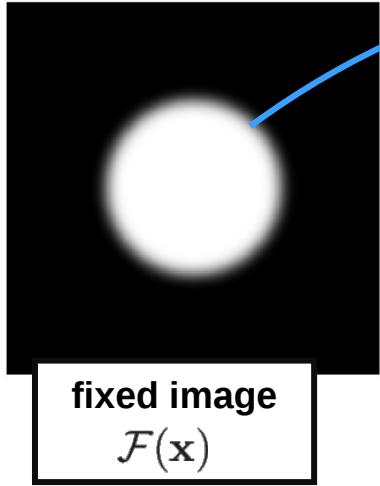


$$y_d(\mathbf{x}, \mathbf{w}_d) = x_d + \delta_d(\mathbf{x}, \mathbf{w}_d) \quad \text{with} \quad \mathbf{w}_d = (w_{d,0}, \dots, w_{d,M-1})^T$$

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

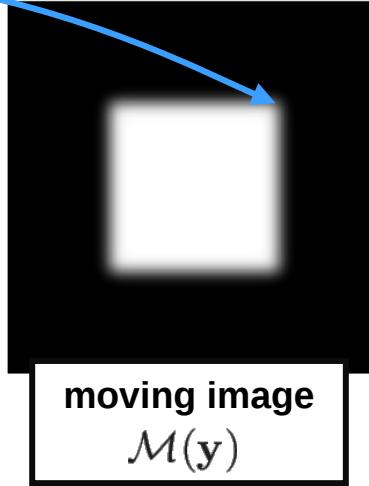
$$\delta_d(\mathbf{x}, \mathbf{w}_d) = \sum_{m=0}^{M-1} w_{d,m} \phi_m(\mathbf{x})$$

nonlinear basis
functions

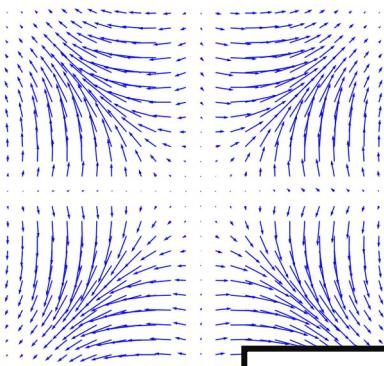


fixed image
 $\mathcal{F}(x)$

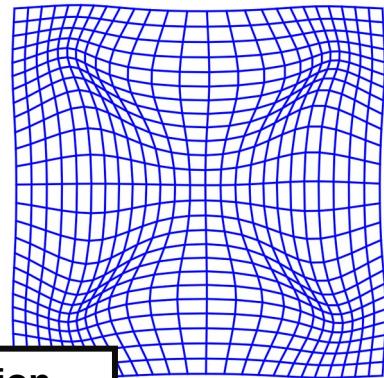
$$y(x, w) = x + \delta(x, w)$$



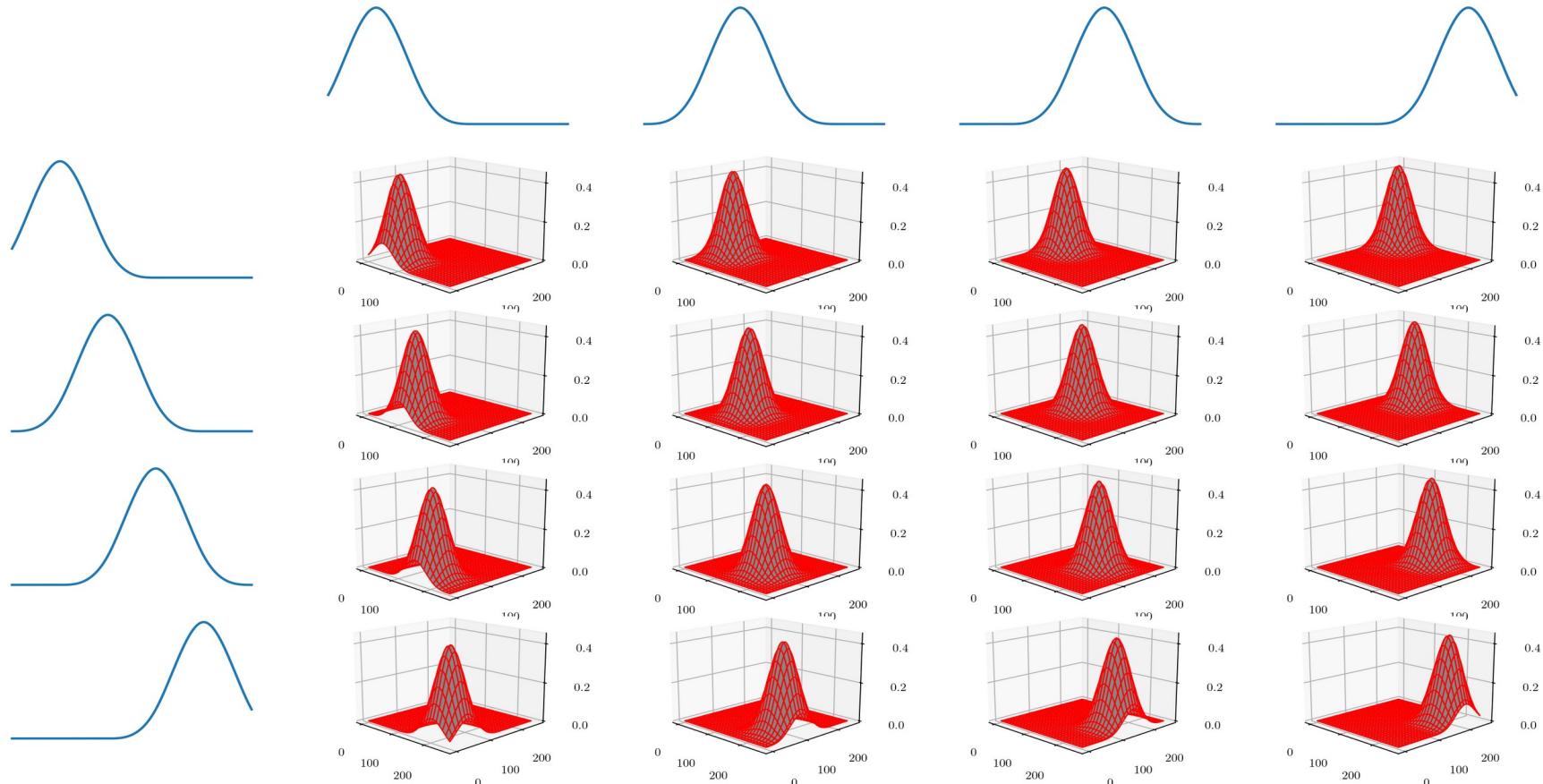
moving image
 $\mathcal{M}(y)$



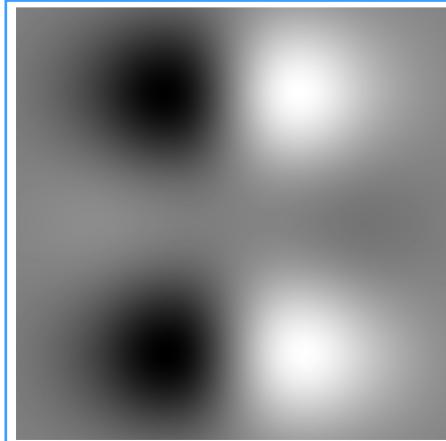
deformation
 $\delta(x, w)$



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

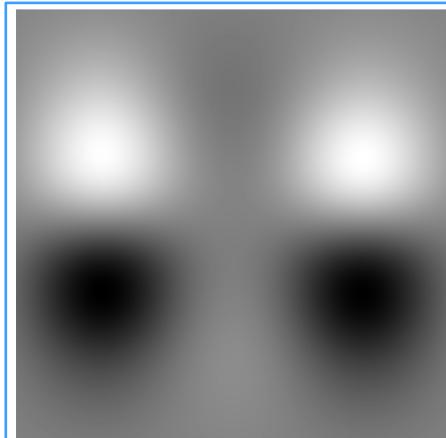
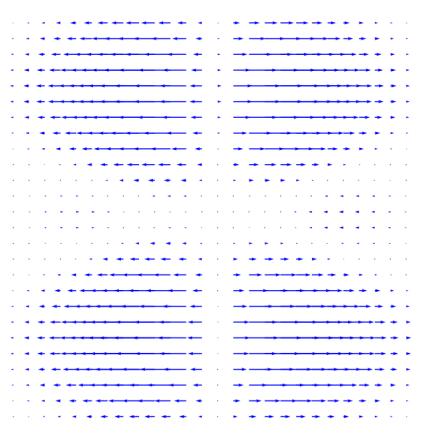


**16 nonlinear
basis functions**



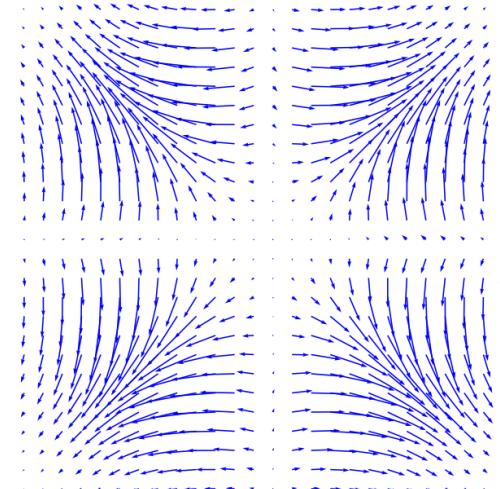
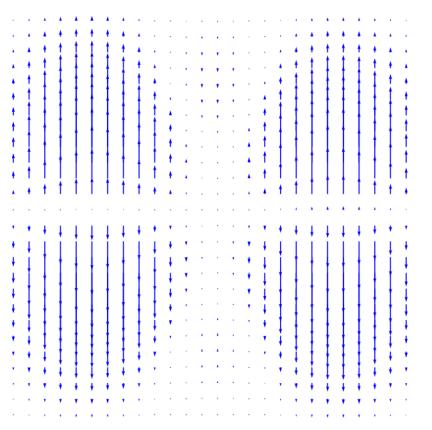
direction 1

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



direction 2

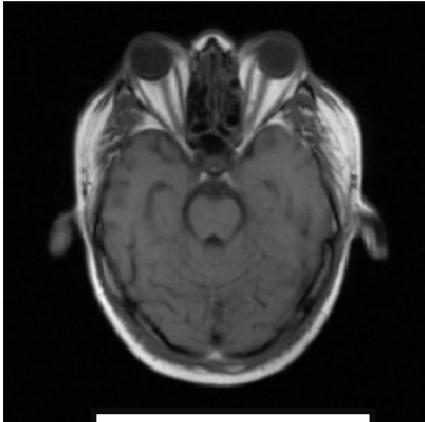
$$\delta_2(\mathbf{x}, \mathbf{w}_2)$$



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

Focus on intra-modal registration

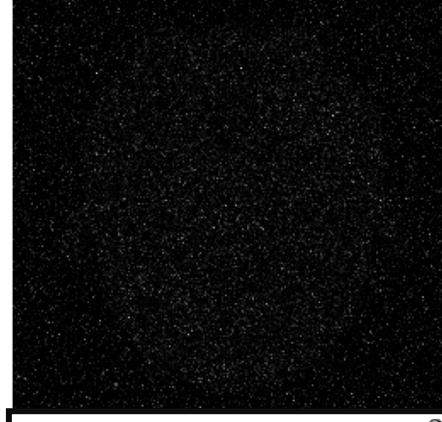
Images have similar intensity characteristics



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



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



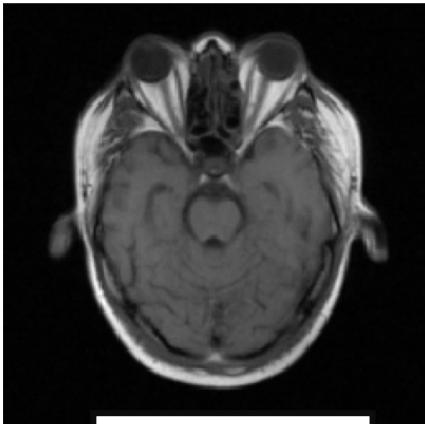
$[\mathcal{F}(\mathbf{x}) - \mathcal{M}(\mathbf{y}(\mathbf{x}, \mathbf{w}))]^2$

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

sum over all voxels

Focus on intra-modal registration

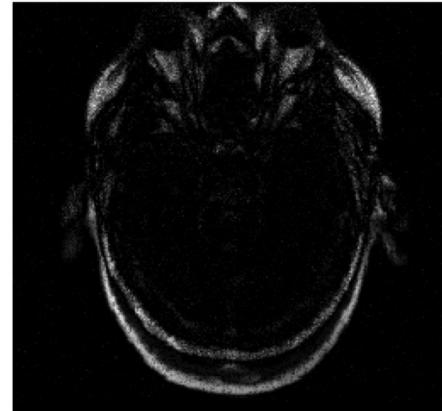
Images have similar intensity characteristics



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



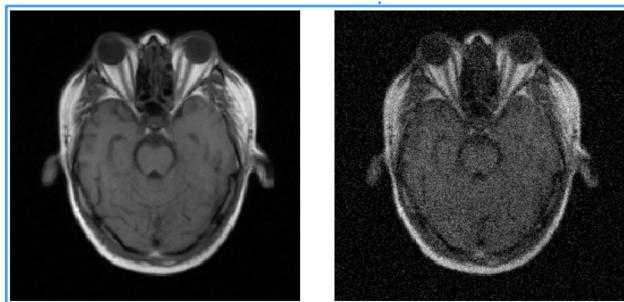
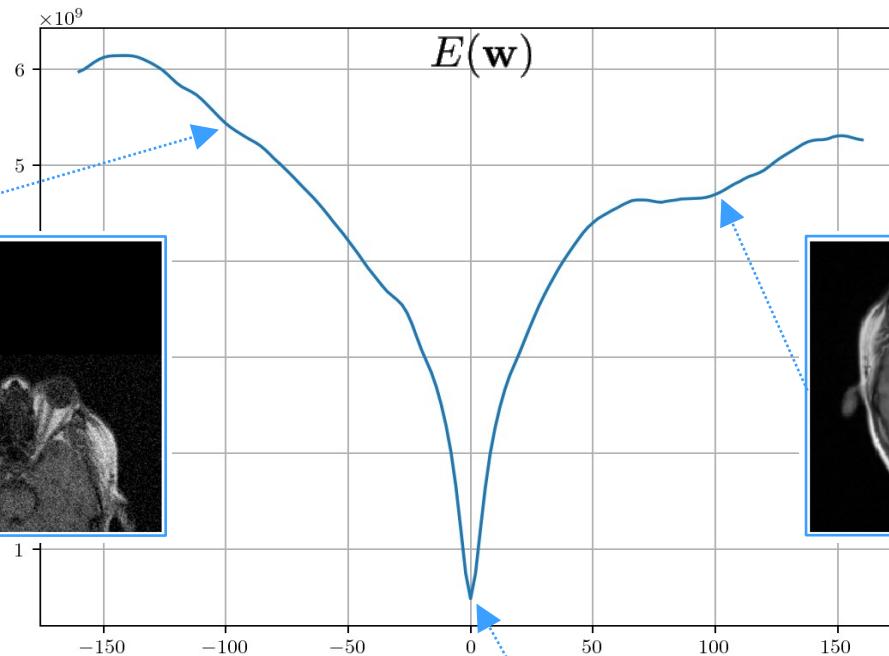
$$\mathcal{M}(\mathbf{y}(\mathbf{x}, \mathbf{w}))$$



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

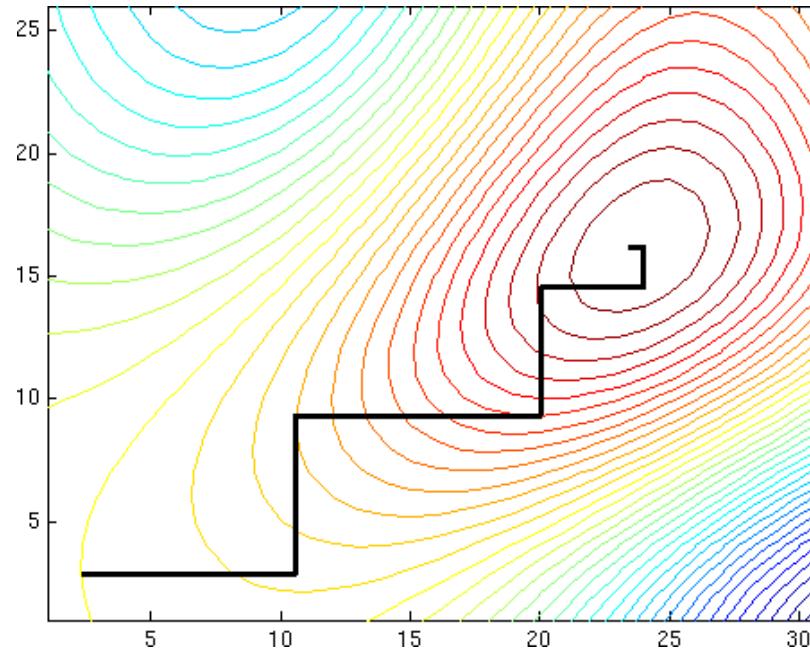
$$E(\mathbf{w}) = \sum_{n=1}^N [\mathcal{F}(\mathbf{x}_n) - \mathcal{M}(\mathbf{y}(\mathbf{x}_n, \mathbf{w}))]^2$$

sum over all voxels



Numerical optimization

Find transformation parameters w that minimize $E(w)$

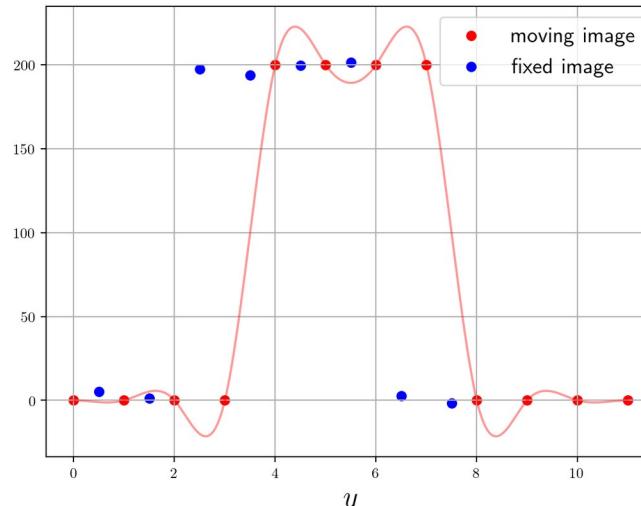


Gauss-Newton optimization

Toy example: 1D and translation only

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

parameter to be
optimized

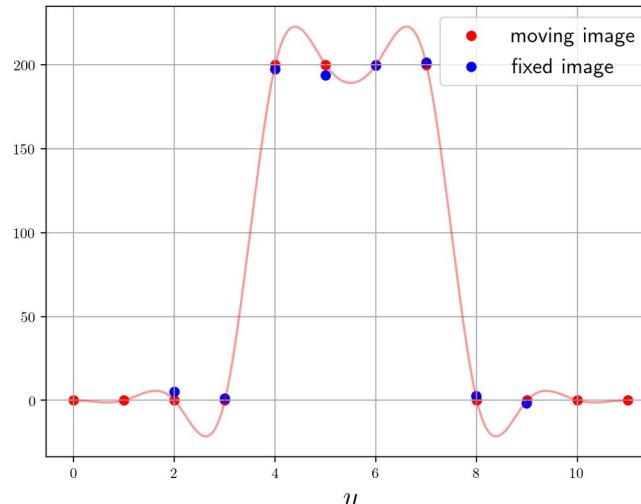


before optimization

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 $E_n(t) = [\mathcal{F}(x) - \mathcal{M}(y(x_n, t))]^2$

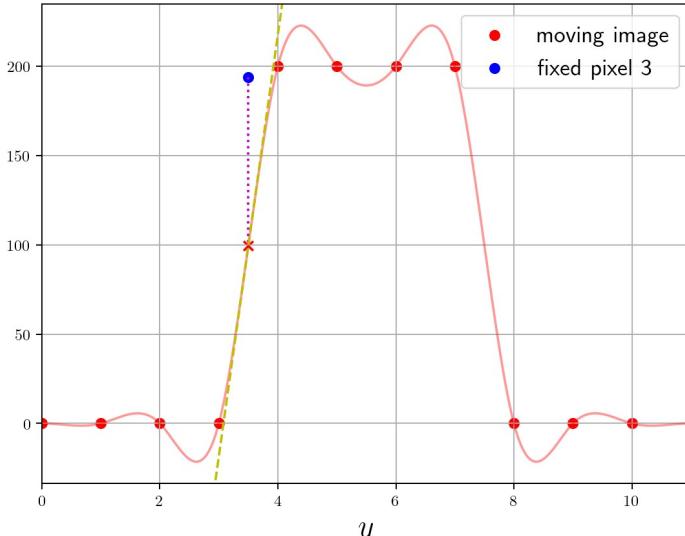


after optimization

Gauss-Newton optimization

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

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!

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

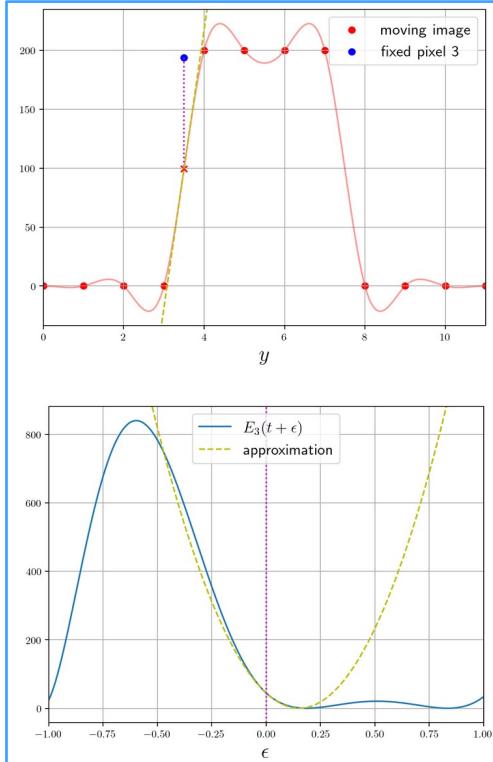
Now update t :
 $t \leftarrow t + \epsilon$

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

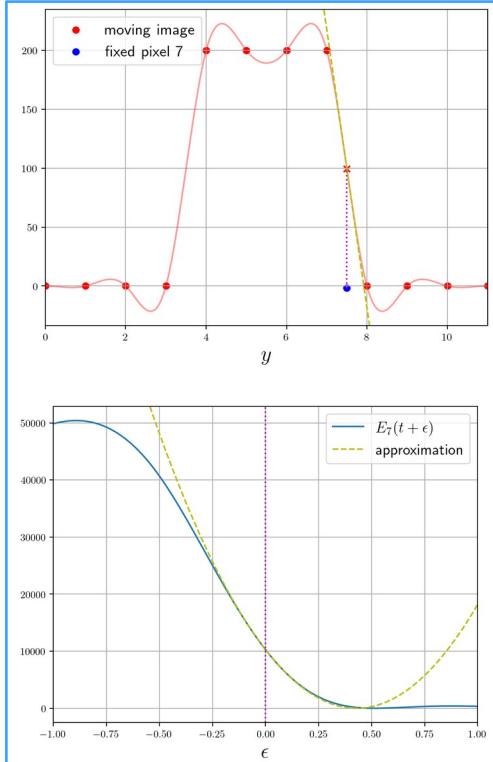
and $\psi = \begin{pmatrix} g_1 \\ \vdots \\ g_N \end{pmatrix}$

one basis function

pixel 3

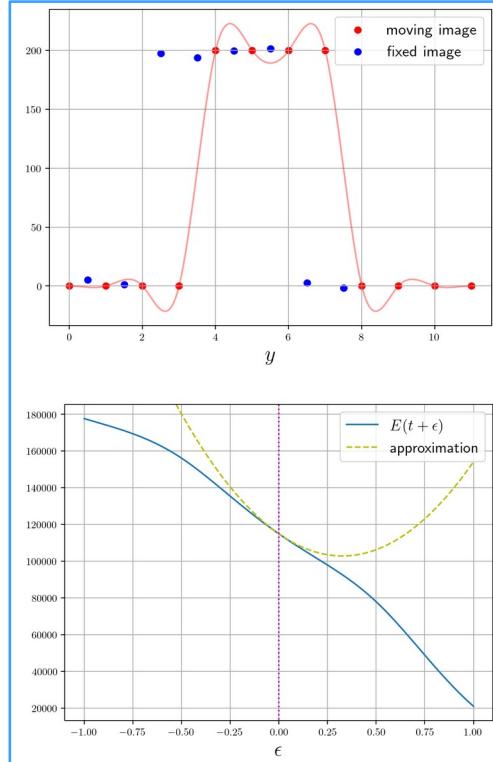


pixel 7



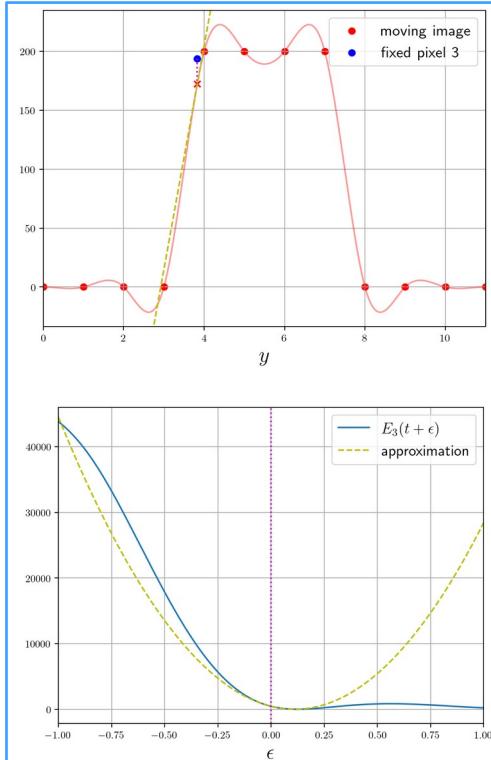
initialization

all pixels

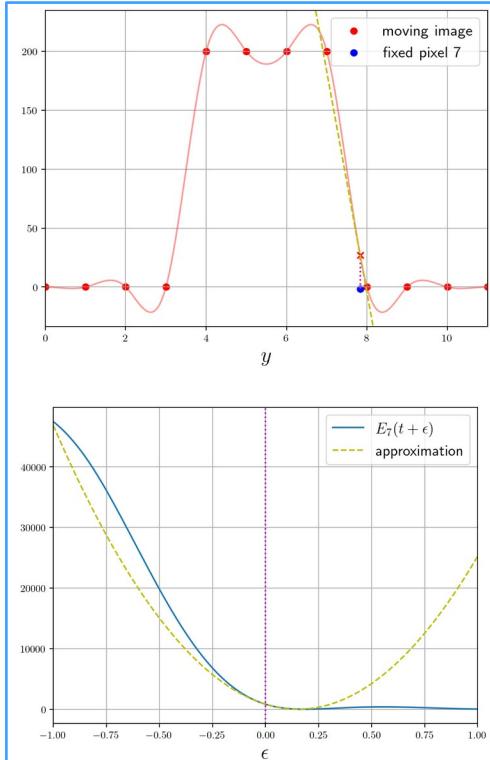


- ✓ Energy: $E(t+\epsilon) = \sum_{n=1}^N E_n(t+\epsilon)$ with $E_n(t+\epsilon) = [\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t+\epsilon))]^2$
- $$\simeq [\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t)) - g_n \cdot \epsilon]^2$$

pixel 3

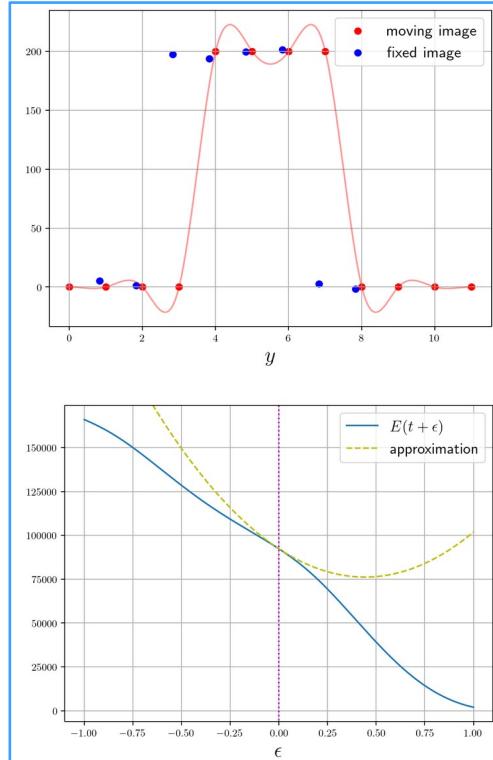


pixel 7



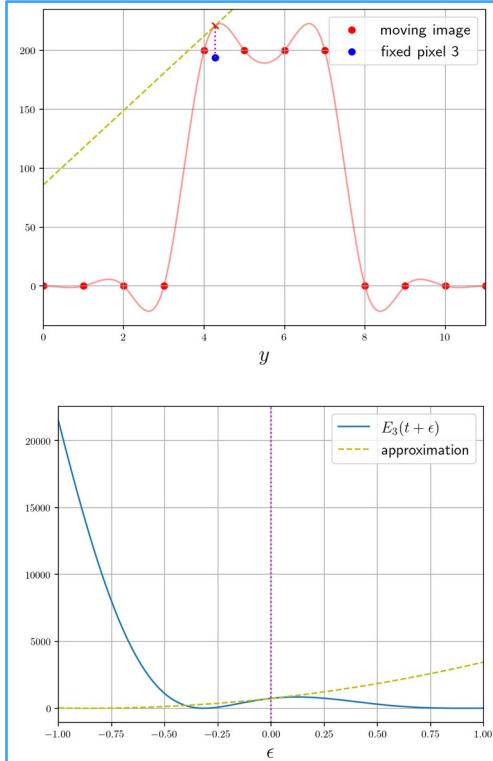
after 1 iteration

all pixels

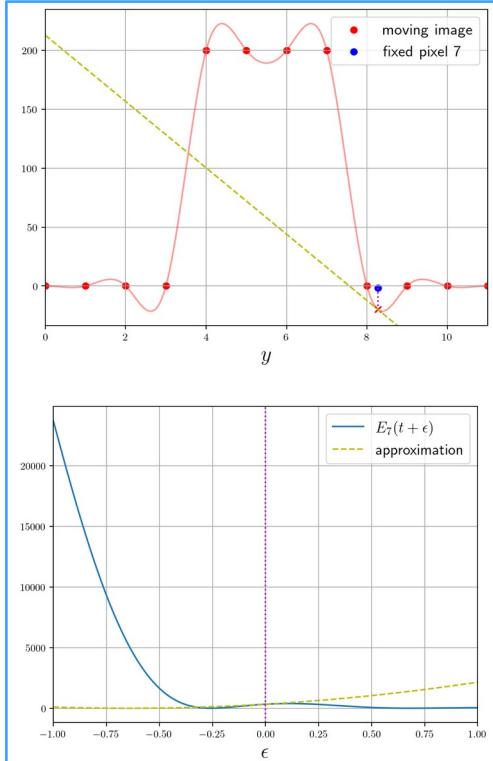


- ✓ Energy: $E(t+\epsilon) = \sum_{n=1}^N E_n(t+\epsilon)$ with $E_n(t+\epsilon) = [\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t+\epsilon))]^2$
- $$\simeq [\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t)) - g_n \cdot \epsilon]^2$$

pixel 3

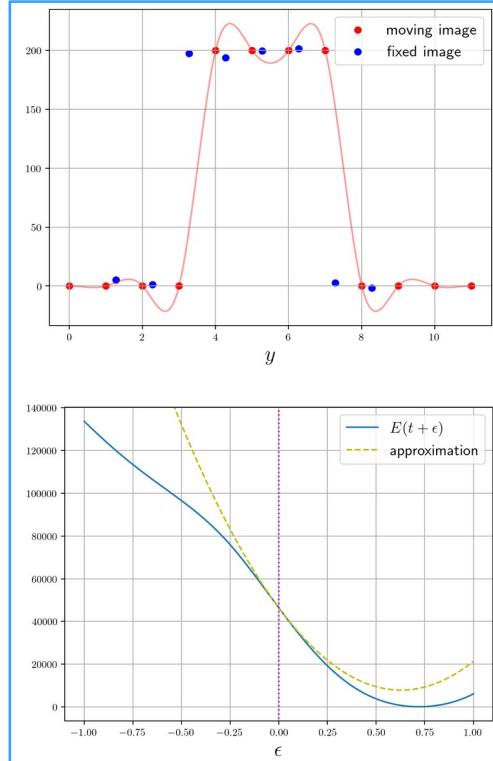


pixel 7



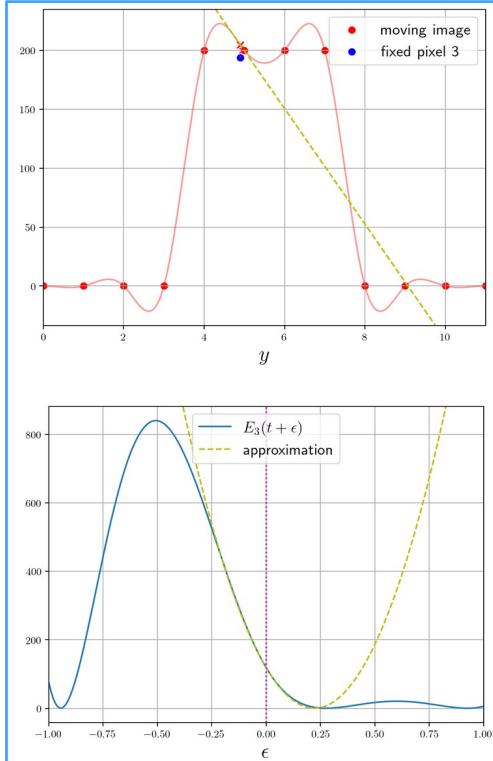
after 2 iterations

all pixels

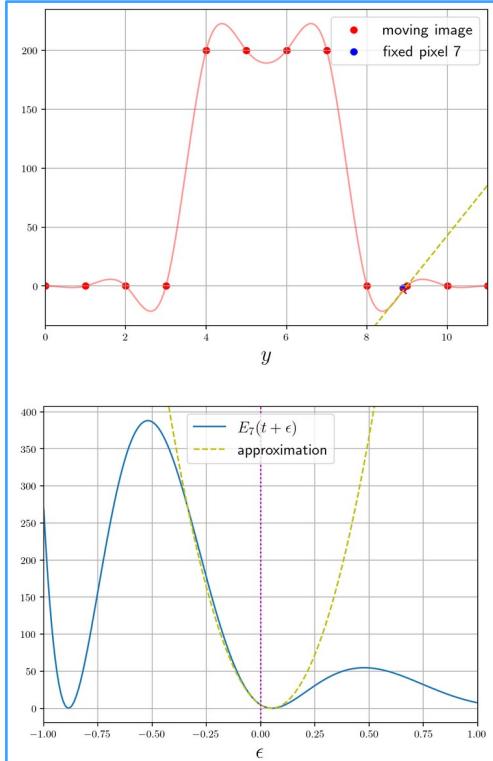


- ✓ Energy: $E(t+\epsilon) = \sum_{n=1}^N E_n(t+\epsilon)$ with $E_n(t+\epsilon) = [\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t+\epsilon))]^2$
- $$\simeq [\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t)) - g_n \cdot \epsilon]^2$$

pixel 3

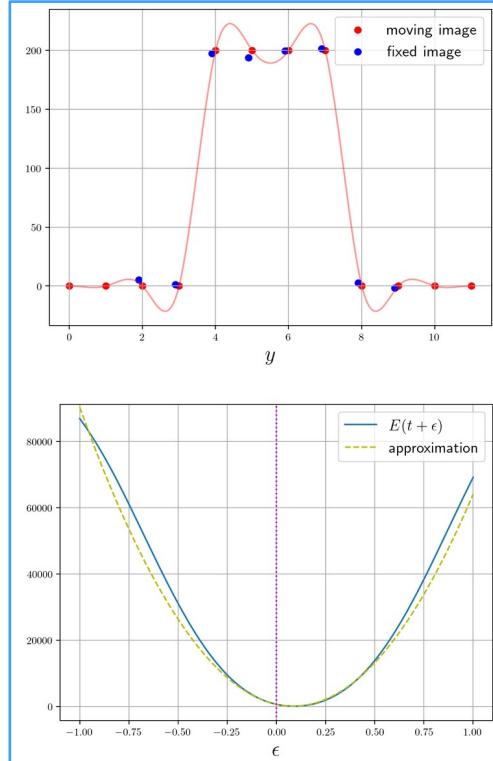


pixel 7



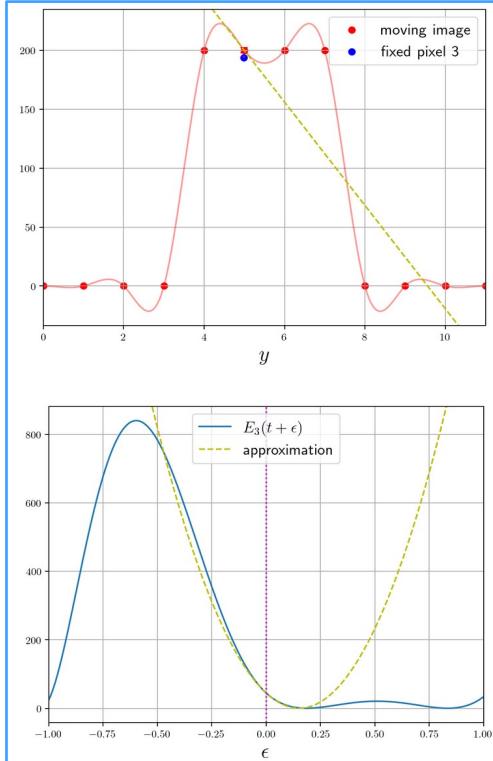
after 3 iterations

all pixels

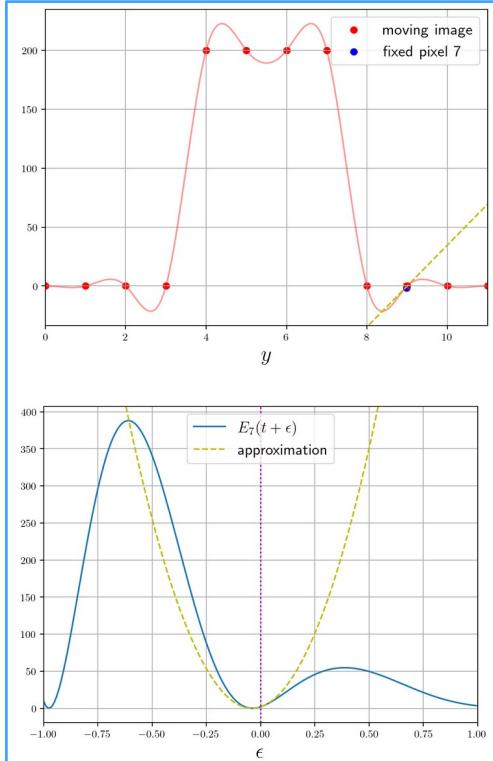


- ✓ Energy: $E(t+\epsilon) = \sum_{n=1}^N E_n(t+\epsilon)$ with $E_n(t+\epsilon) = [\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t+\epsilon))]^2$
- $$\simeq [\mathcal{F}(x_n) - \mathcal{M}(y(x_n, t)) - g_n \cdot \epsilon]^2$$

pixel 3

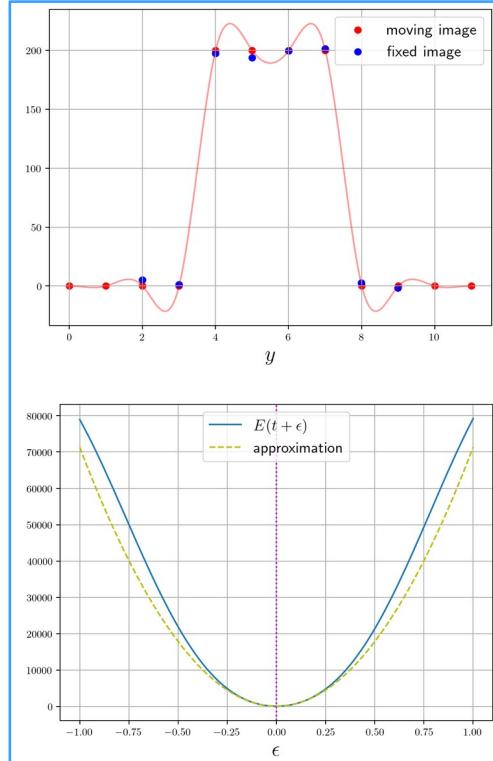


pixel 7



after 4 iterations

all pixels



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

Gauss-Newton optimization

Solution: standard linear regression!

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

Now update t :
 $t \leftarrow t + \epsilon$

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

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

one basis function

Gauss-Newton optimization

Solution: standard linear regression!

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

Now update t :
 $t \leftarrow t + \epsilon$

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

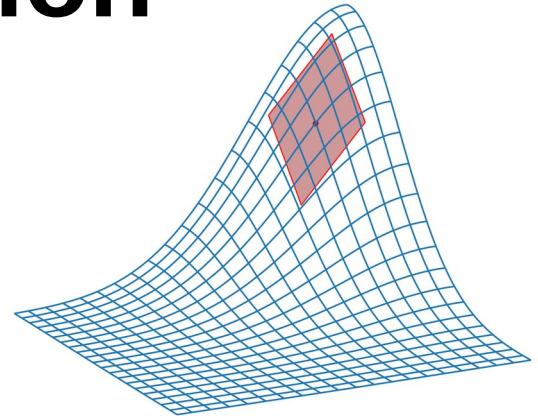
and $\psi = \begin{pmatrix} g_1 \\ \vdots \\ g_N \end{pmatrix} = \underbrace{\begin{pmatrix} g_1 & & \\ & \ddots & \\ & & g_N \end{pmatrix}}_{\mathbf{G}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$

one basis function

Gauss-Newton optimization

In real situations, exactly the same idea but:

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



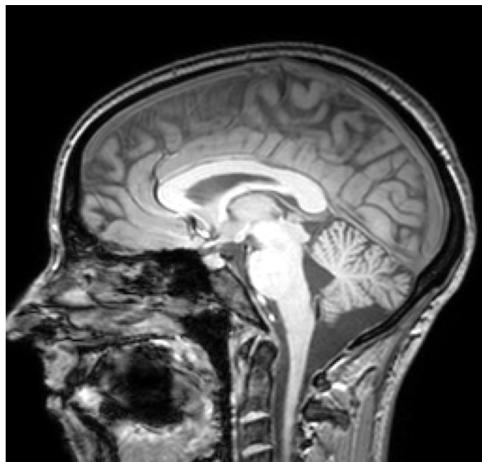
$$\mathcal{M}(\mathbf{y}(\mathbf{x}_n, \mathbf{w} + \boldsymbol{\epsilon})) \simeq \mathcal{M}(\mathbf{y}(\mathbf{x}_n, \mathbf{w})) + \sum_{d=1}^D \sum_{m=0}^M (g_{d,n} \phi_m(\mathbf{x}_n)) \epsilon_{d,m}$$

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

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

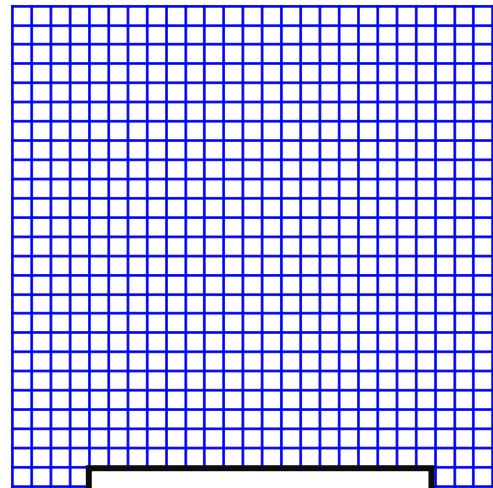


fixed image
 $\mathcal{F}(x)$



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

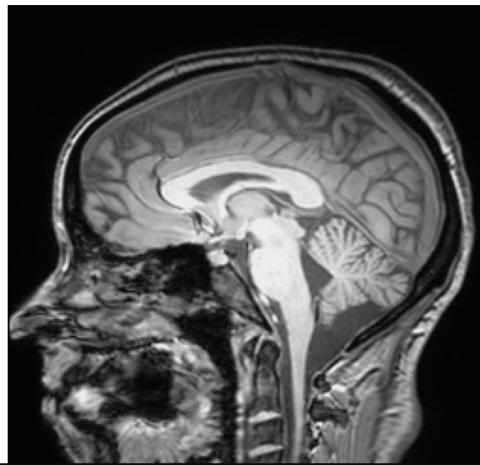
initialization



deformation
 $\delta(x, w)$

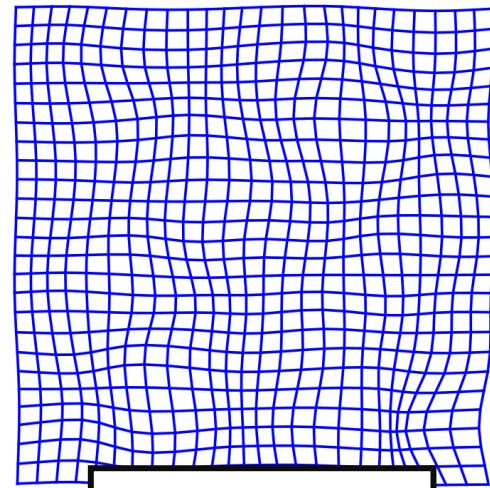


fixed image
 $\mathcal{F}(x)$



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

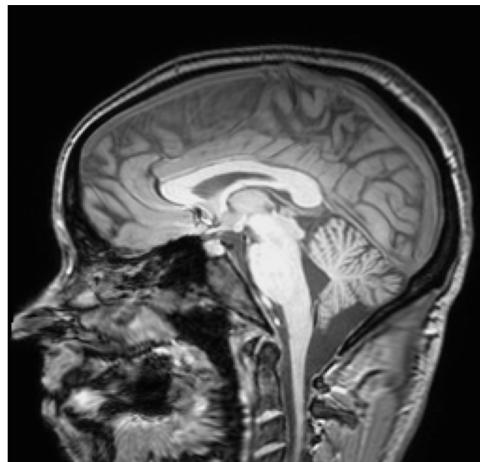
after 10 iterations



deformation
 $\delta(x, w)$

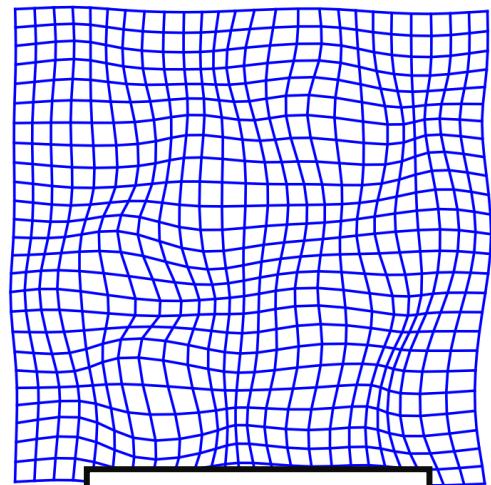


fixed image
 $\mathcal{F}(x)$



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

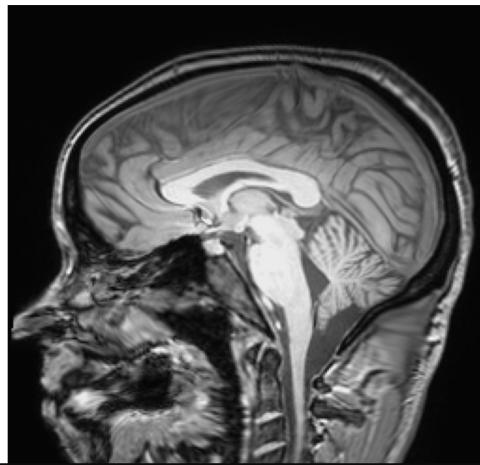
after 30 iterations



deformation
 $\delta(x, w)$

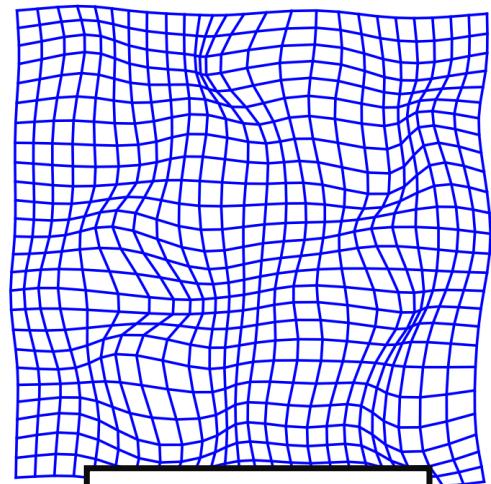


fixed image
 $\mathcal{F}(x)$

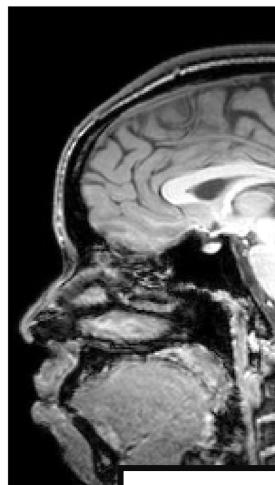


interpolated moving image
 $\mathcal{M}(y(x, w))$

after convergence

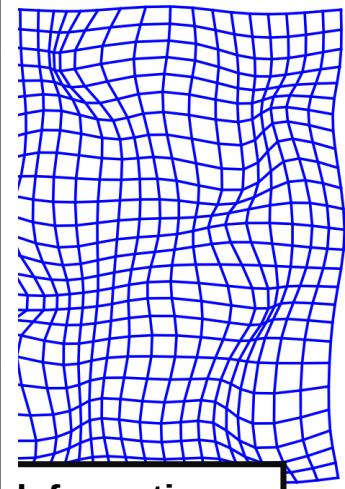
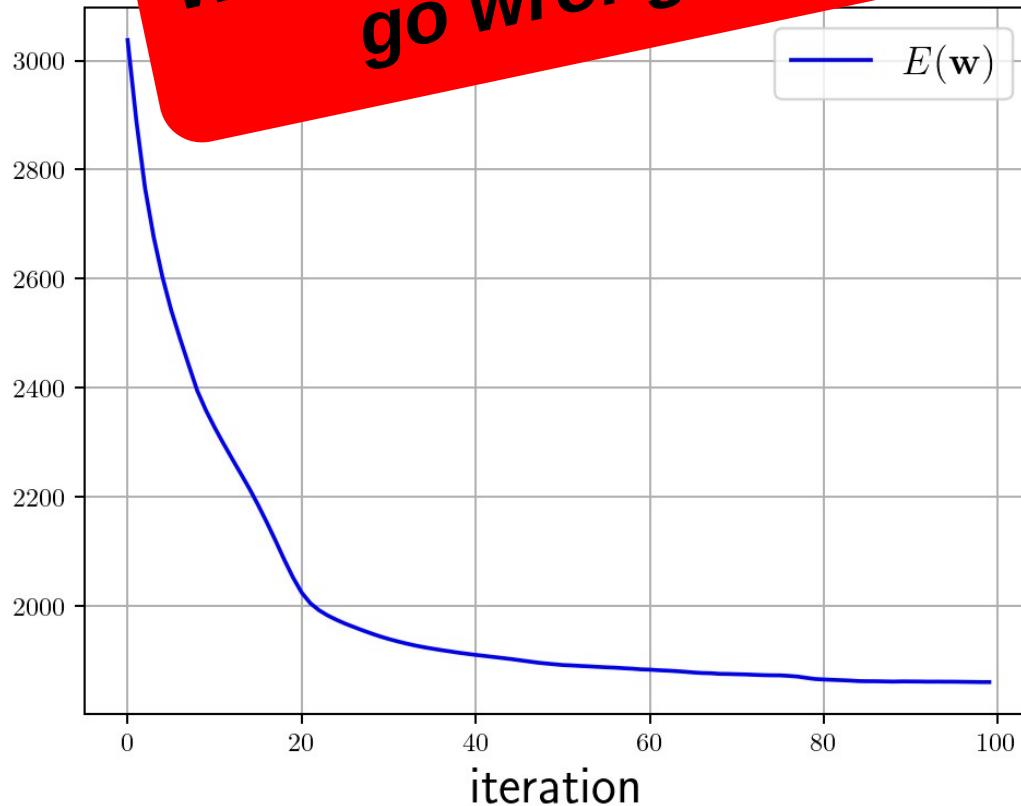


deformation
 $\delta(x, w)$



fixed ir
 $\mathcal{F}(x)$

What could possibly
go wrong?



leformation
 $\delta(x, w)$

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

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

tunable

