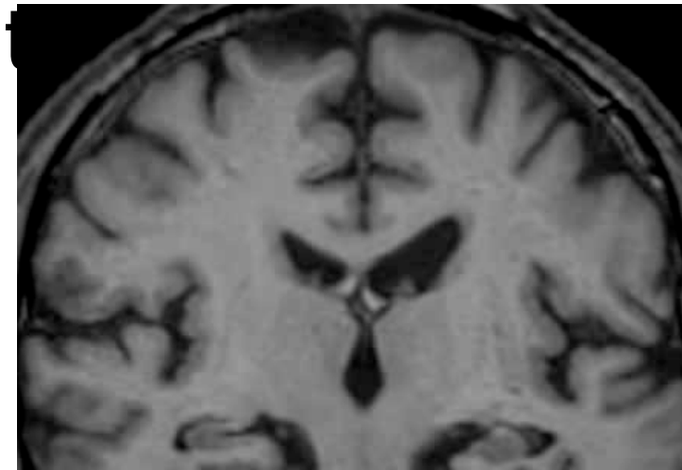
Medical Image Registration Transformations

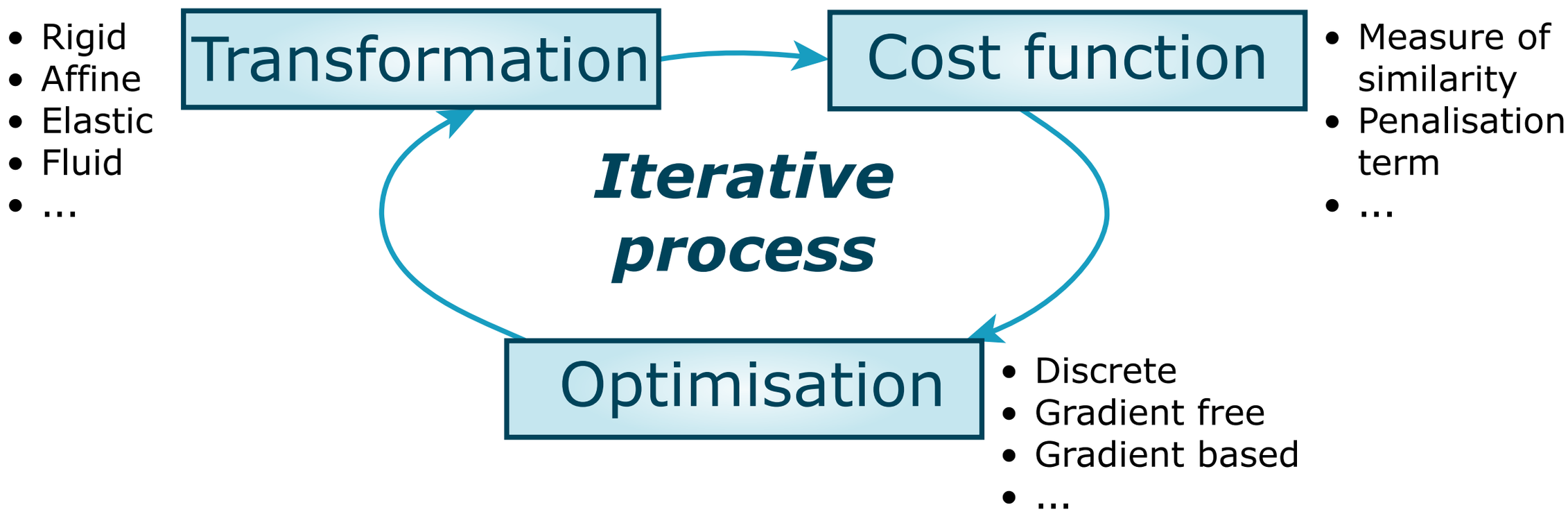
Marc Modat

Centre for Medical Image Comput
Dementia Research Centre
University College London



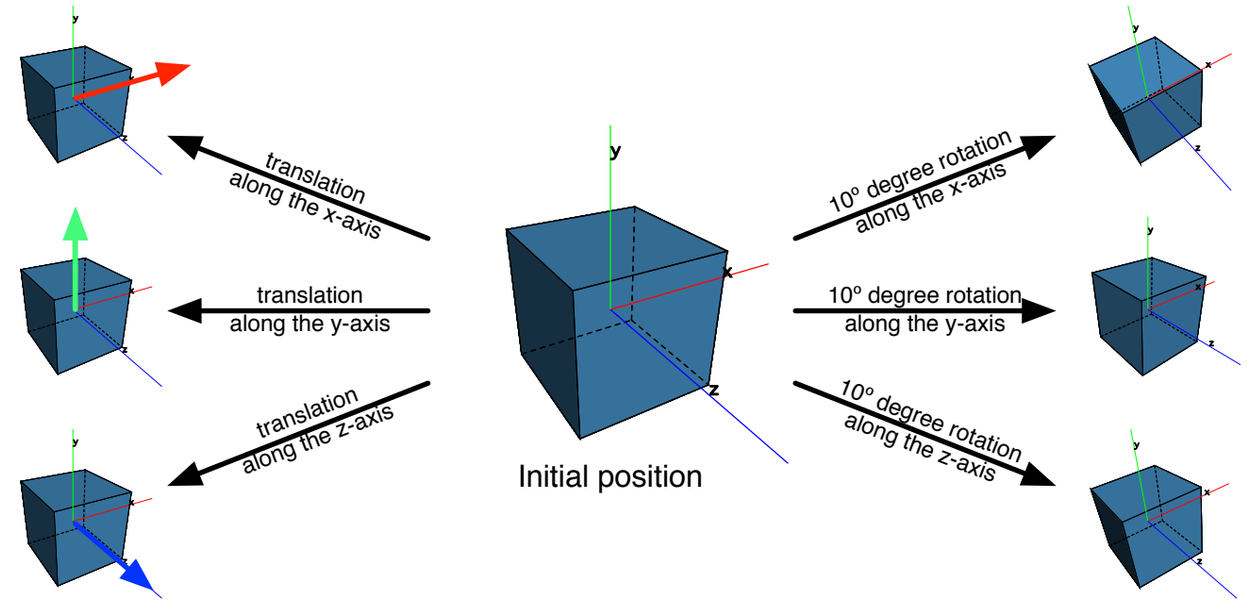
Medical image registration

- Overall scheme

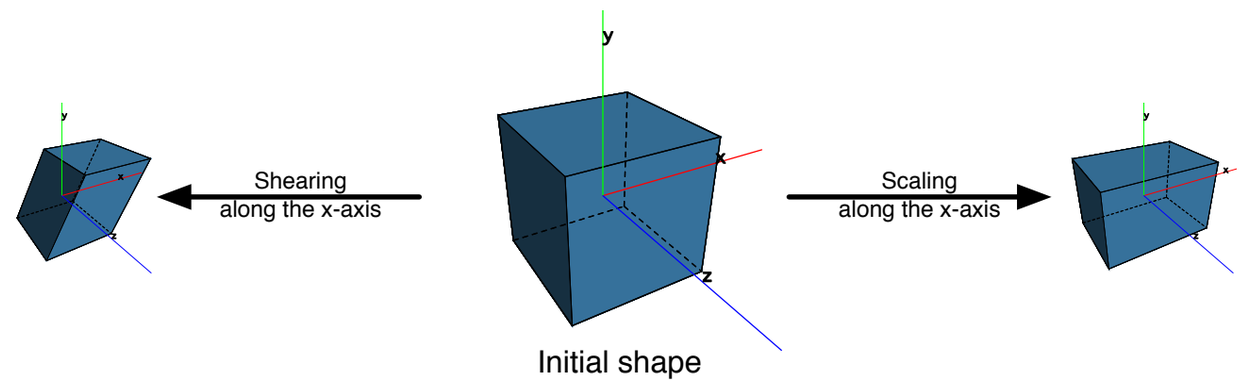


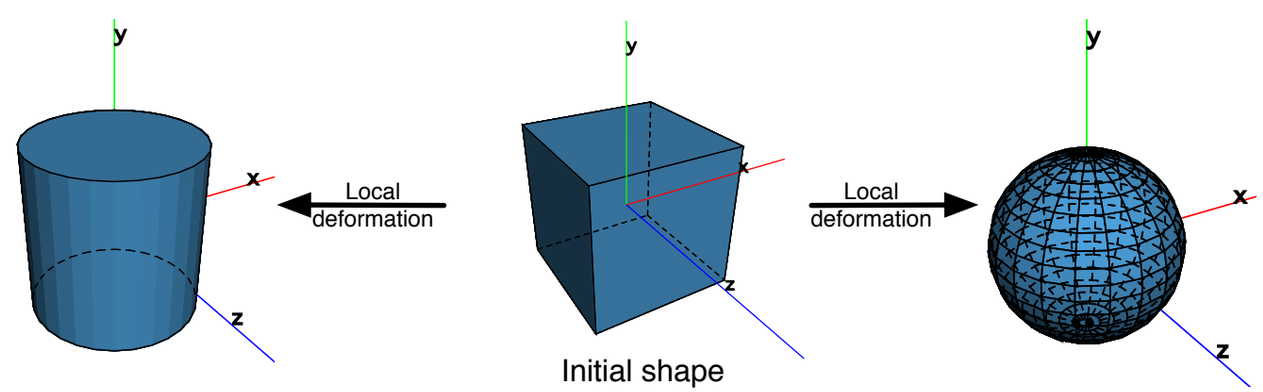
Outline

- Rigid /Affine

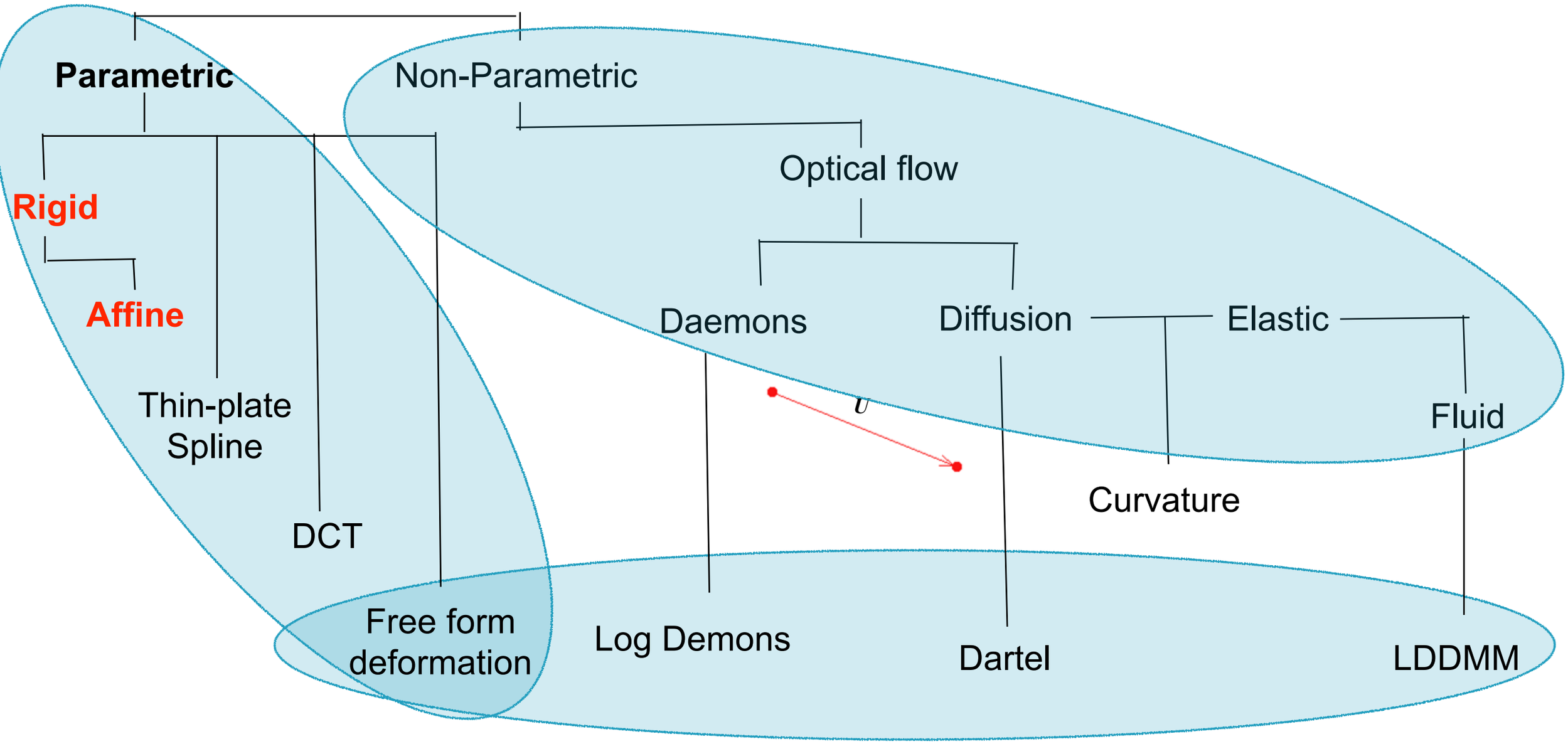


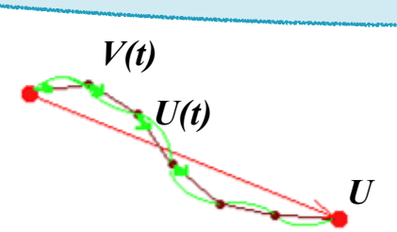
- Non-rigid
  - parametric
    - Thin plate spline
    - Discrete cosine function
    - Free form deformation
  - non-parametric
    - Optical flow
    - Elastic
    - Diffusion
    - Fluid

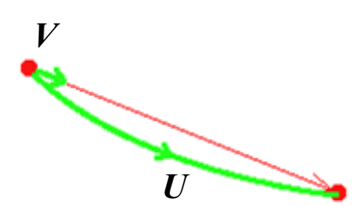


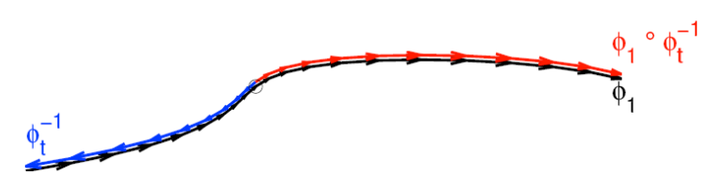


A taxonomy of reg. algorithms





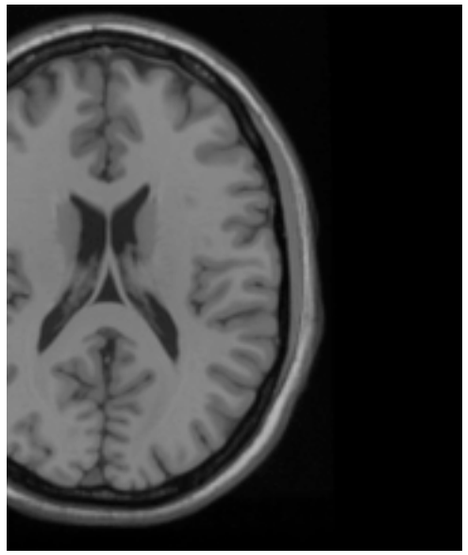




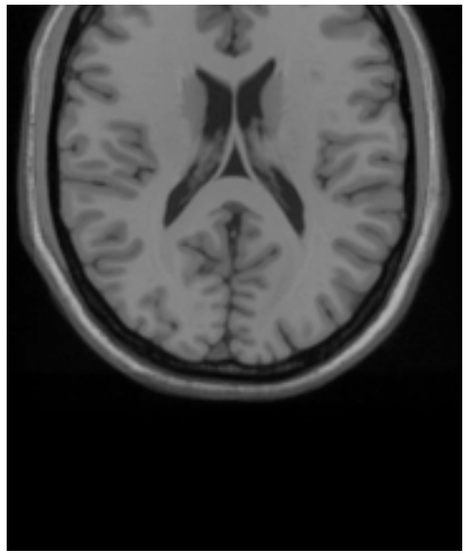
Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



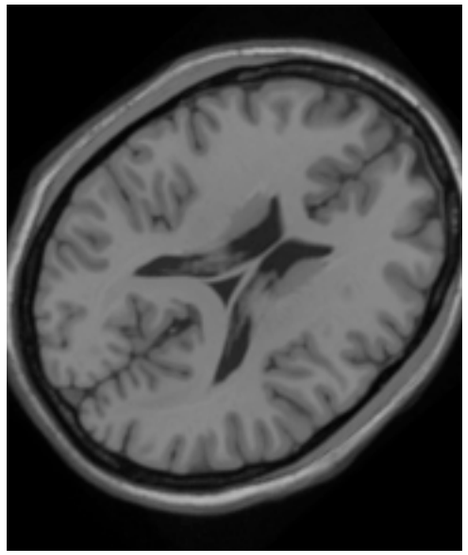
$$\mathbf{T}(\vec{x}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

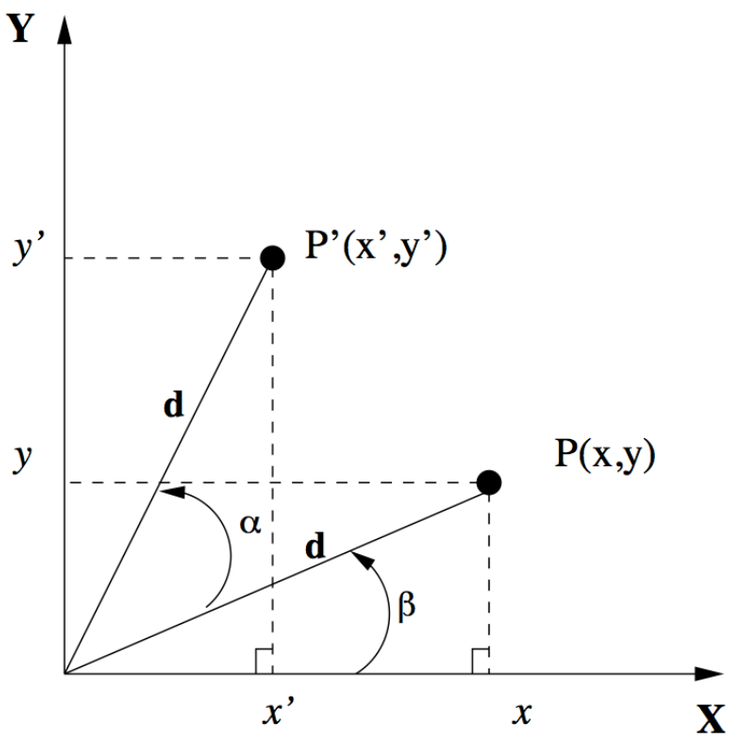


Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$





$$x = d\cos(\beta)$$

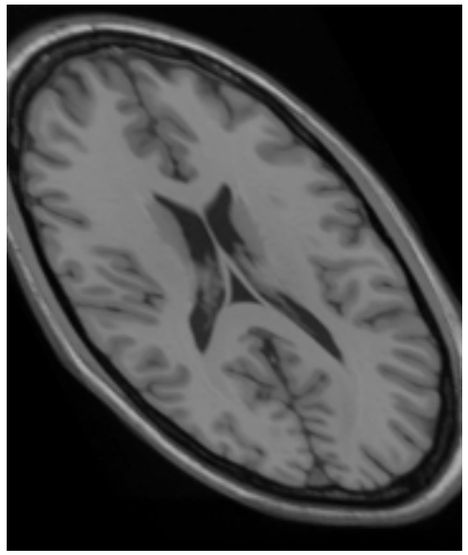
$$y = d\sin(\beta)$$

$$\cos(-\alpha) = \cos(\alpha) \text{ and } \sin(-\alpha) = -\sin(\alpha)$$

Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system

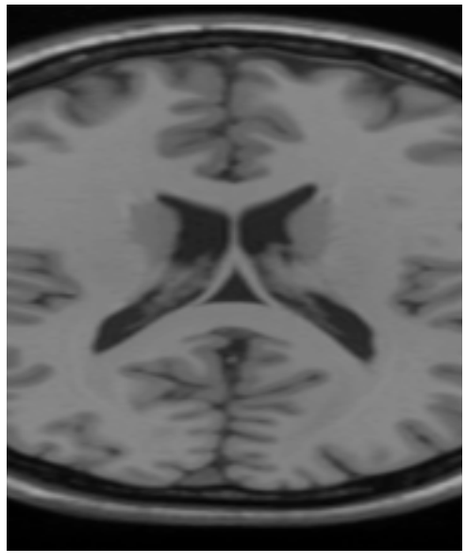
$$\mathbf{T}(\vec{x}) = \begin{pmatrix} 1 & \alpha & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



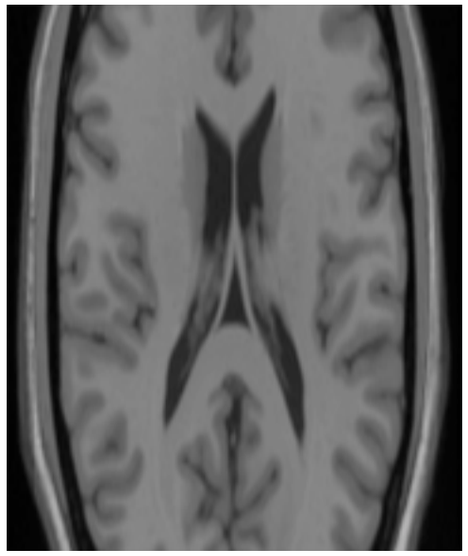
Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} S_x & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



$$\mathbf{T}(\vec{x}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system
  - All degrees of freedom are parametrised by 1 matrix

$$\mathbf{Aff}(\vec{x}) = \mathbf{Sh} \times \mathbf{Sc} \times \mathbf{R_x} \times \mathbf{T_x} \times \mathbf{T_y} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\mathbf{T}_{2D}(\vec{x}) = \begin{pmatrix} \alpha & \beta & t_x \\ \gamma & \delta & t_y \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Rigid / Affine parametrisation

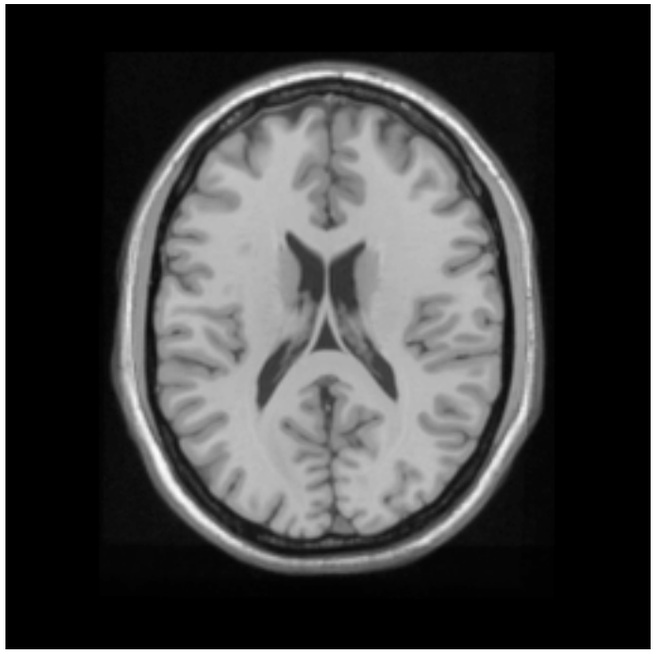
- Transformation model - Homogenous coordinate system
  - All degrees of freedom are parametrised by 1 matrix

$$\mathbf{Aff}(\vec{x}) = \mathbf{Sh} \times \mathbf{Sc} \times \mathbf{R_x} \times \mathbf{R_y} \times \mathbf{R_z} \times \mathbf{T_x} \times \mathbf{T_y} \times \mathbf{T_z} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\mathbf{T}_{3D}(\vec{x}) = \begin{pmatrix} \alpha & \beta & \gamma & t_x \\ \delta & \epsilon & \zeta & t_y \\ \eta & \theta & \iota & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system
  - Change of origin
  - How to register these two images:
    - Rotation by 90º clockwise



Reference image

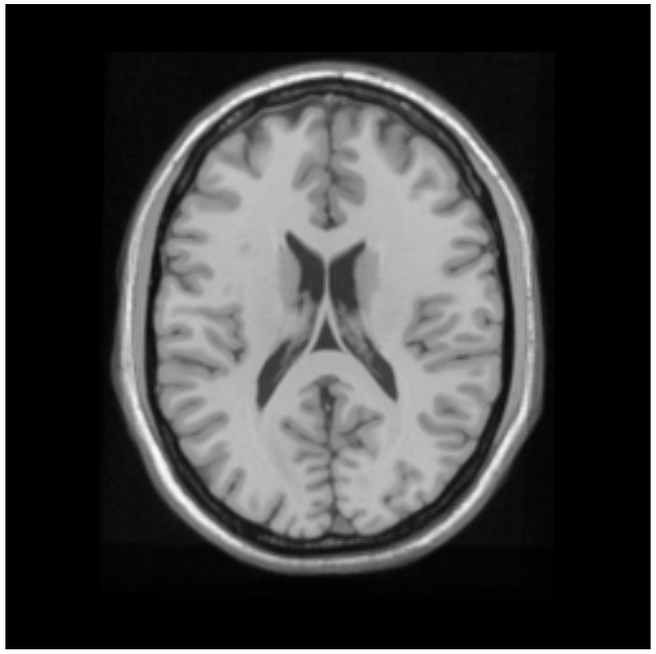
Floating image

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

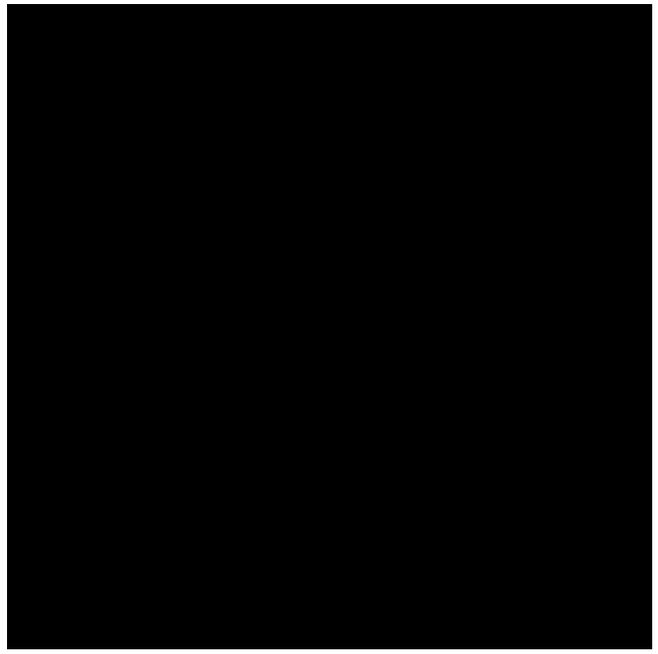
$$\text{with } \theta = -90 \times \pi/180$$

Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system
  - Change of origin
  - How to register these two images:
    - Rotation by 90º clockwise



Reference image



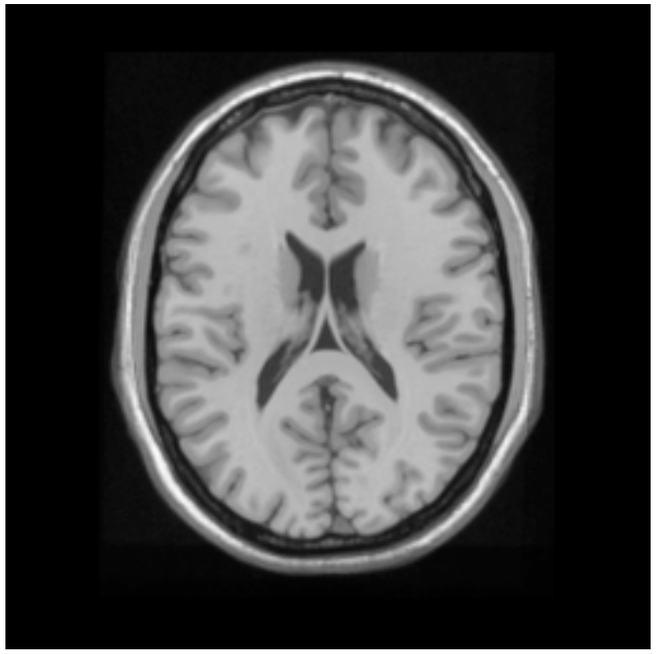
Warped floating image

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

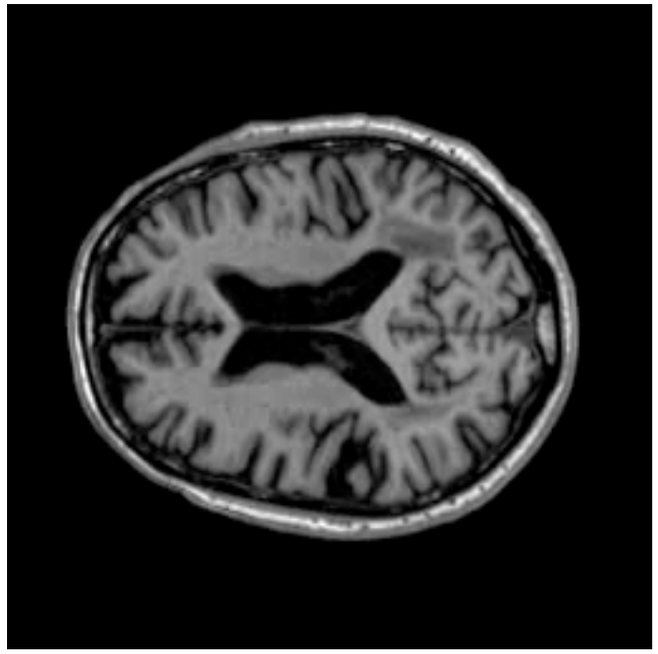
$$\text{with } \theta = -90 \times \pi/180$$

Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system
  - Change of origin
  - How to register these two images:
    - Rotation by from 0º to 90º clockwise



Reference image



Warped floating image

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\text{with } \theta = -[0:90] \times \pi/180$$

Rigid / Affine parametrisation

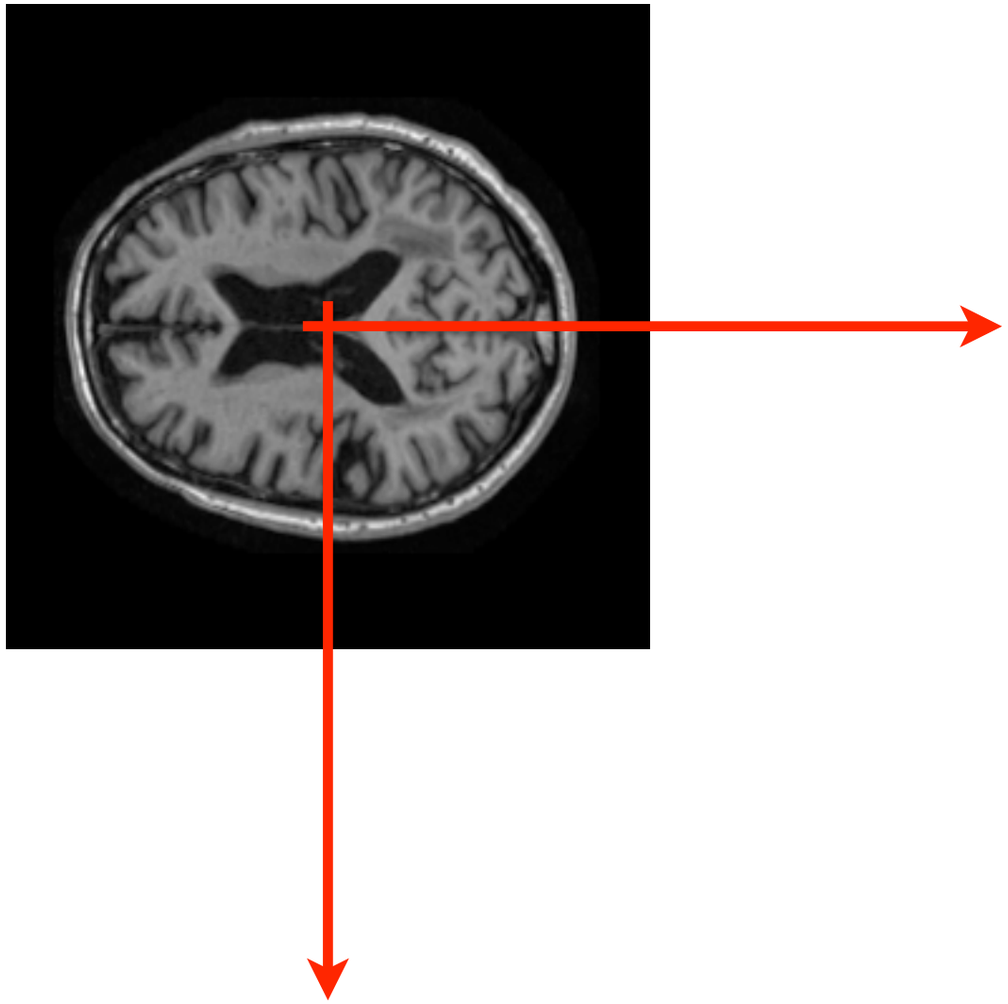
- Transformation model - Homogenous coordinate system
  - Change of origin
    - From 0,0 (top left in the example)
    - To geometrical centre of the image

Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system
  - Change of origin
    - From 0,0 (top left in the example)
    - To geometrical centre of the image

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\text{with } t_x = \frac{\dim_x}{2} \text{ and } t_y = \frac{\dim_y}{2}$$

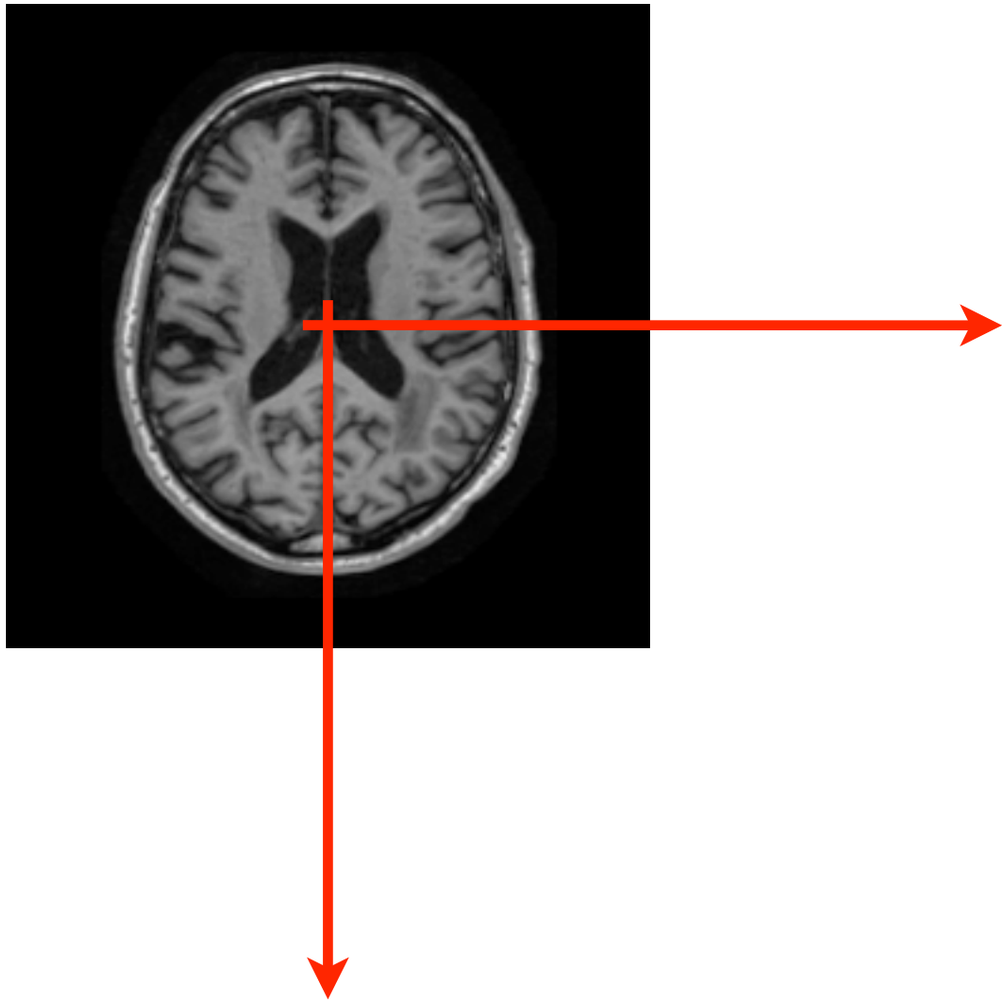


Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system
  - Change of origin
    - From 0,0 (top left in the example)
    - To geometrical centre of the image

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

with $\theta = -90 \times \pi/180$

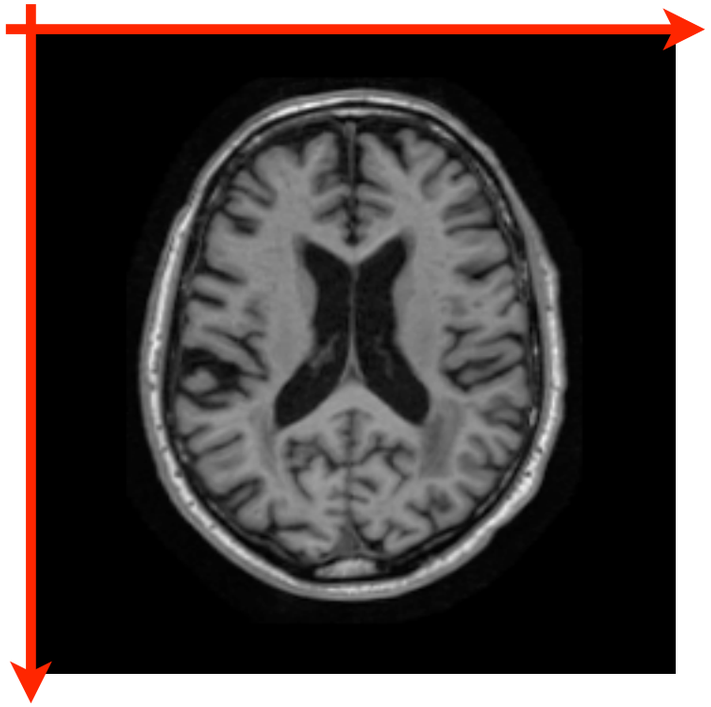


Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system
  - Change of origin
    - From 0,0 (top left in the example)
    - To geometrical centre of the image

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

$$\text{with } t_x = \frac{\dim_x}{2} \text{ and } t_y = \frac{\dim_y}{2}$$



Rigid / Affine parametrisation

- Transformation model - Homogenous coordinate system
  - Change of origin
    - One can use composition of transformation

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

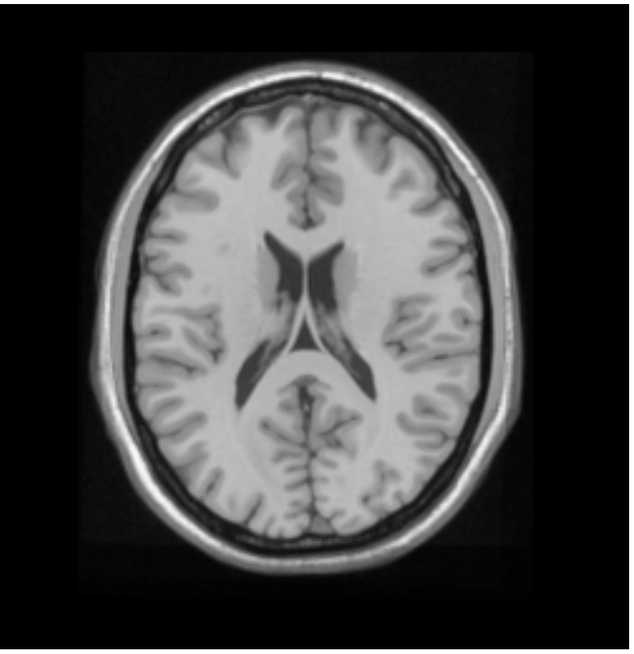
$$\text{with } \theta = -90 \times \pi/180 \text{ , } t_x = \frac{\dim_x}{2} \text{ and } t_y = \frac{\dim_y}{2}$$

$$\mathbf{T}(\vec{x}) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & t_x(1-\cos(\theta)) + t_y sin(\theta) \\ -\sin(\theta) & \cos(\theta) & -t_x sin(\theta) + t_y(1-\cos(\theta)) \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

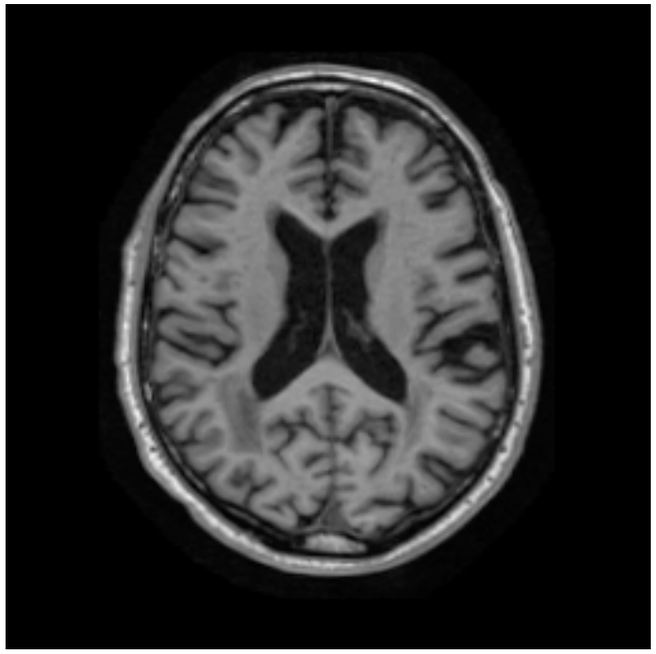
$$\text{with } \theta = -90 \times \pi/180 \text{ , } t_x = \frac{\dim_x}{2} \text{ and } t_y = \frac{\dim_y}{2}$$

Rigid / Affine parametrisation

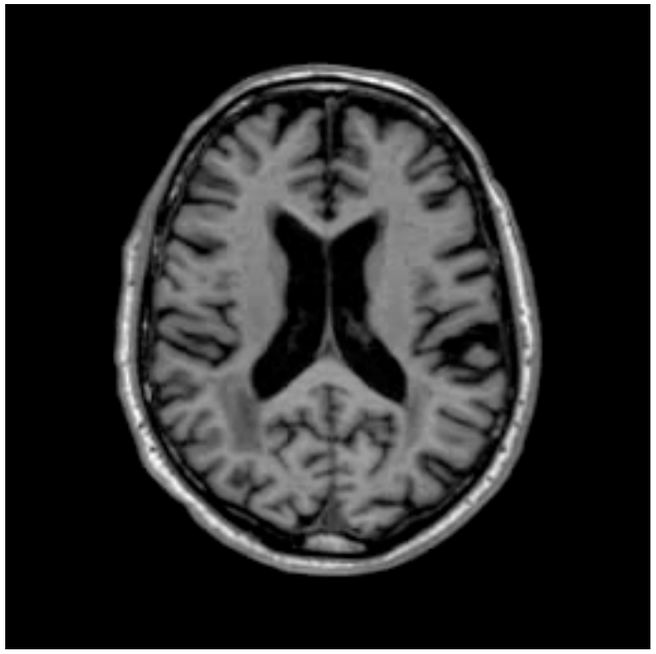
- Transformation model - Homogenous coordinate system
  - Change of origin
    - Note that the same applies for all degrees of freedom
      - example with scaling



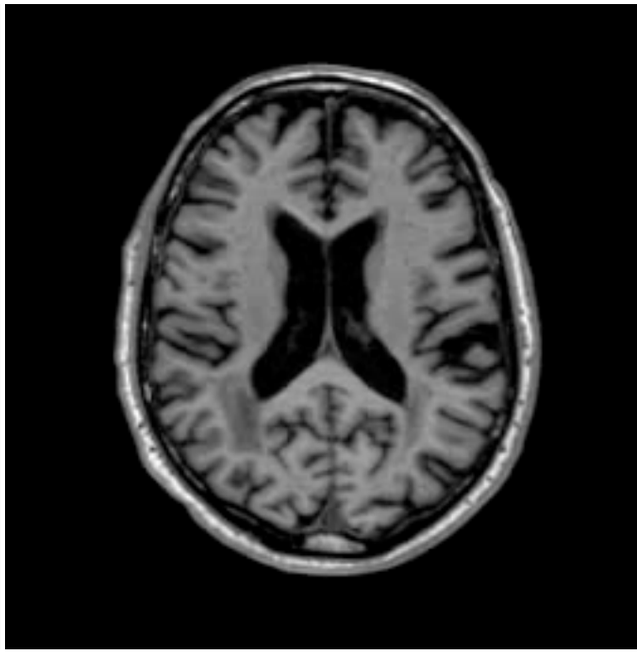
Reference image



Floating image



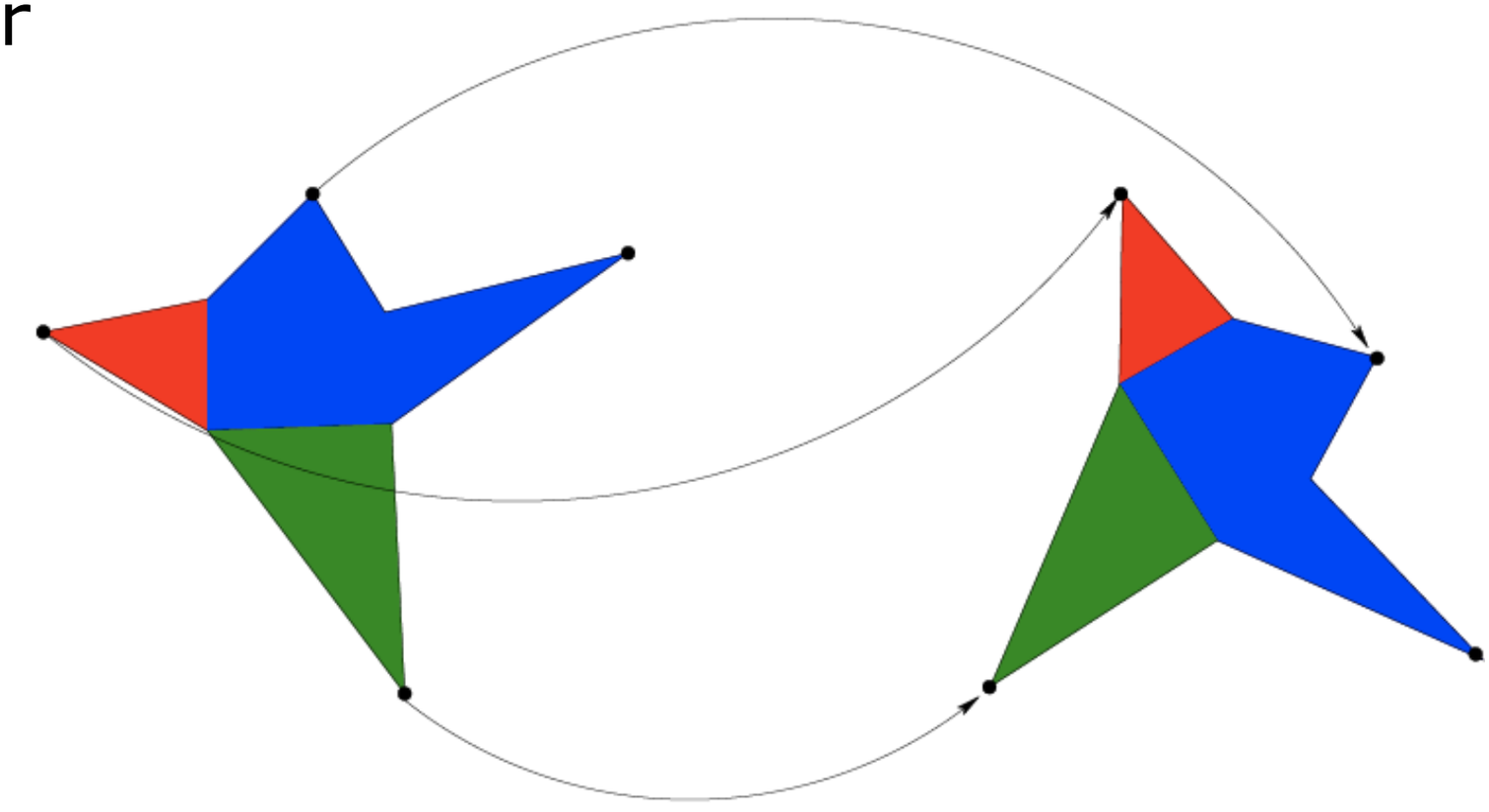
Scaled floating image, scaling along the origin



Scaled floating image, scaling along the centre

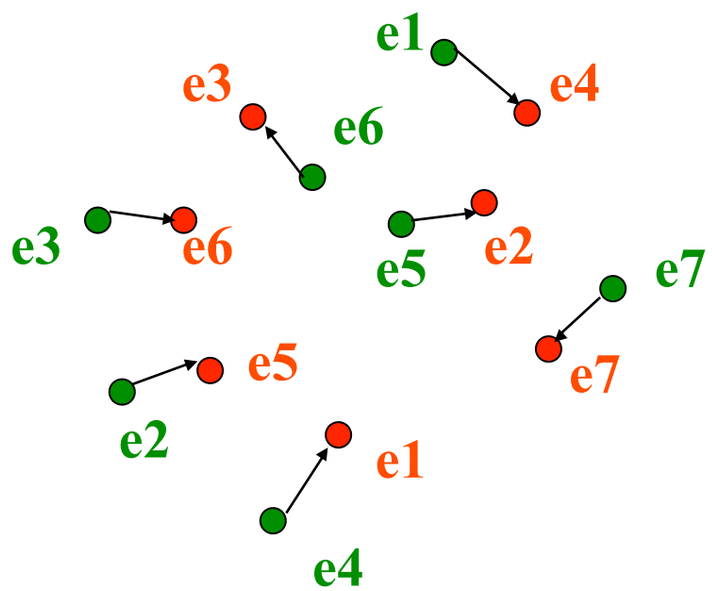
Rigid / Affine parametrisation

- Example of landmark based registration
  - Extract primitives (example: points with high curvature)
    - Often, we do need a segmentation step to extract these primitives
  - Find set of corresponding primitives
  - Compute the transformation to align these primitives together



Rigid / Affine parametrisation

- Example of landmark based registration
  - Step 1: find the correspondences between the primitives



primitives from I
primitives from J

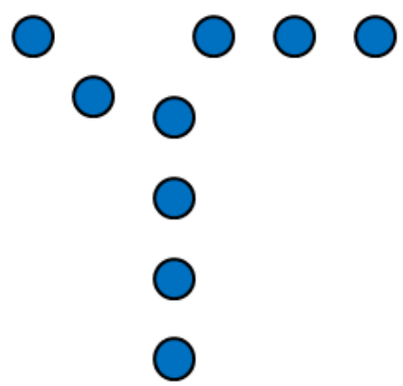
find the « closest » primitive (using e.g. Euclidean distance)

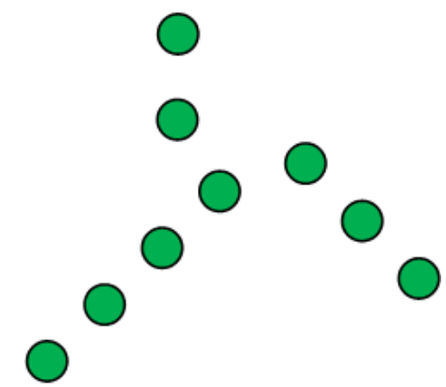
  - Step 2: Estimate the transformation

  - Repeat step 1 and 2 until convergence

Rigid / Affine parametrisation

- Example of landmark based registration
  - Consider two point sets:
    $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}$ and $\mathbf{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_N\}$





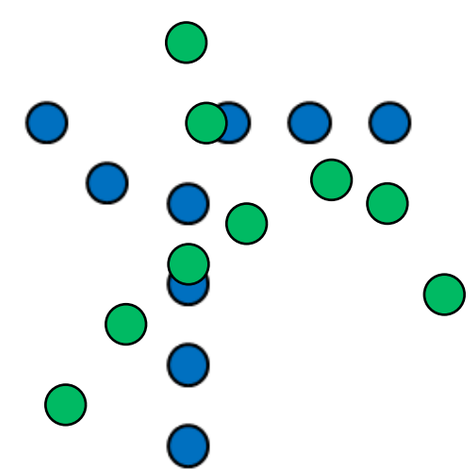
  - Problem is to find a rigid transformation (R, t) to minimise the Fiducial Registration Error:

$$FRE = \frac{1}{N}\sum_i \left(\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\right)^2$$

    - First step is to demean the points:

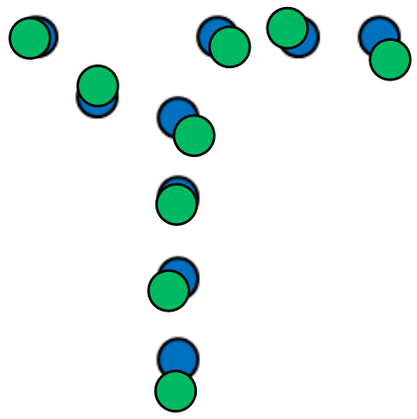
$$\bar{\mathbf{p}} = \frac{1}{N}\sum_i \mathbf{p}_i \qquad \bar{\mathbf{q}} = \frac{1}{N}\sum_i \mathbf{q}_i$$

$$\mathbf{p}'_i = \mathbf{p}_i - \bar{\mathbf{p}} \qquad \mathbf{q}'_i = \mathbf{q}_i - \bar{\mathbf{q}}$$



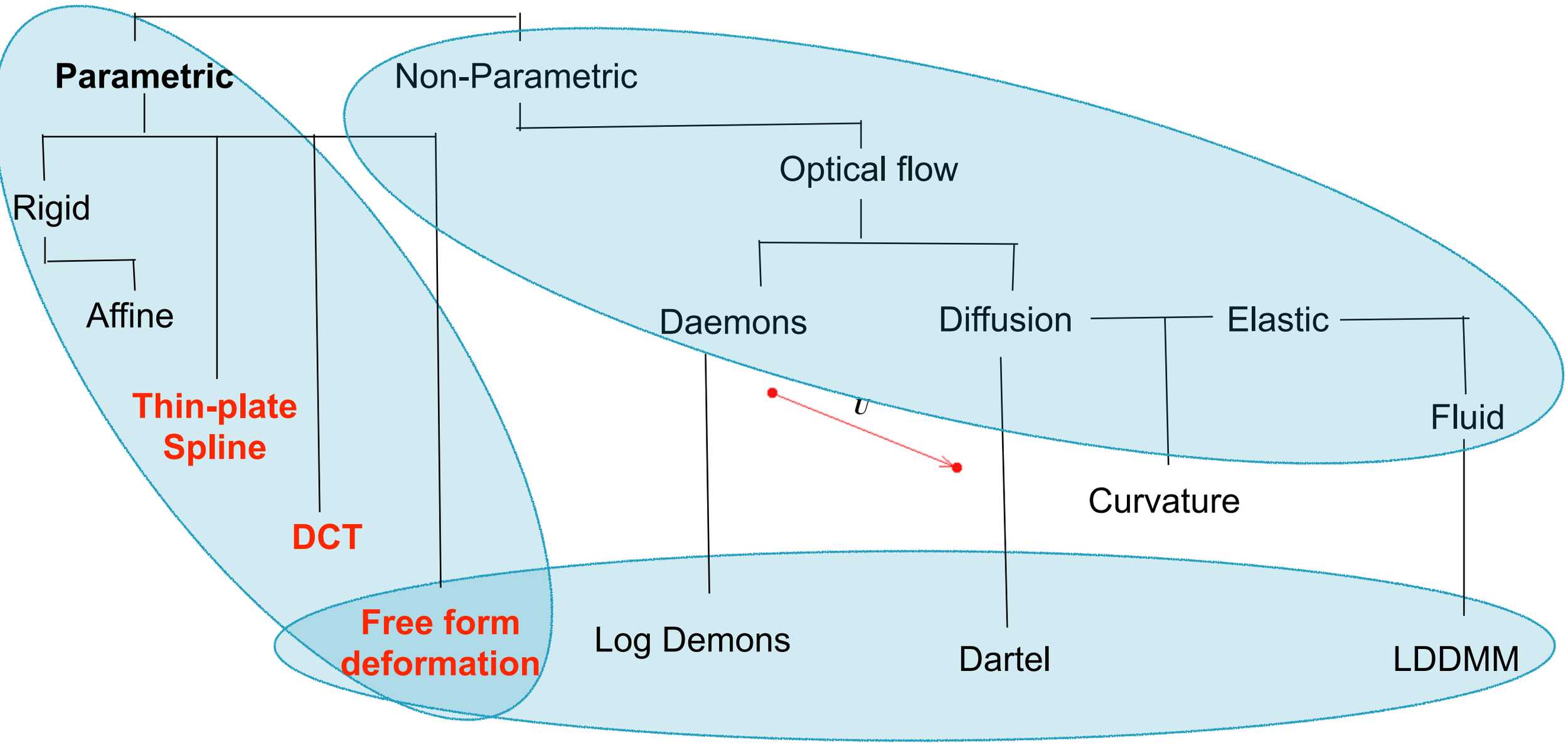
Rigid / Affine parametrisation

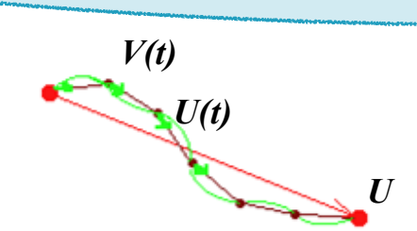
- Example of landmark based registration
  - Second derive the rigid rotation.
    - Several methods but all depend on forming covariance matrix: $\mathbf{H} = \sum_i \mathbf{p}_i' \mathbf{q}_i'^t$
    - Most common method uses Singular Value Decomposition.
    - In brief the steps are:
      - Perform SVD of H $\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^t$
      - Optimal R is then $\mathbf{R} = \mathbf{V}\mathbf{U}^t$

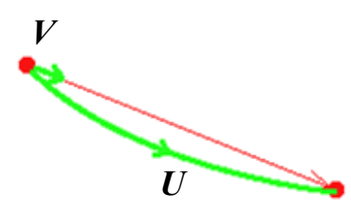


  - Update the translation parameter
  - Iterate

A taxonomy of reg. algorithms

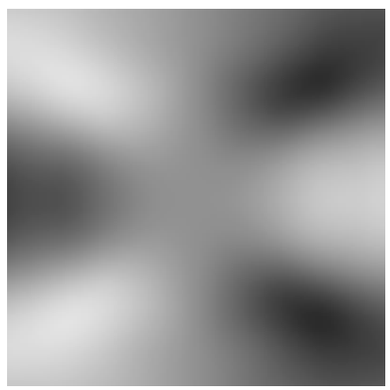




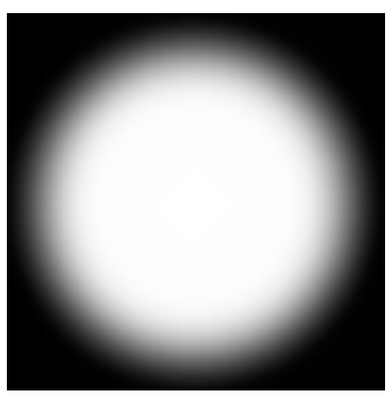


Parametric registration - Concept

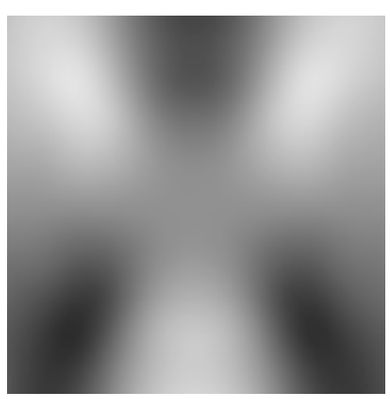
- Two families of non-rigid registration algorithms
  1. non-parametric
     - Direct optimisation of the deformation field **u**
     - 2 (or 3) degrees of freedom per pixel (voxel)
     - $x' = T(x;\mathbf{u})$



Disp. X axis

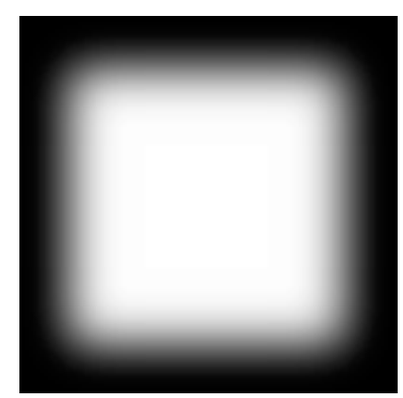


Reference image



Disp. Y axis

Warped image



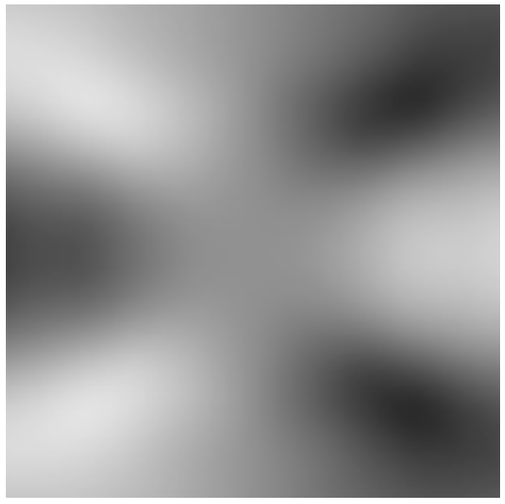
Floating image

Parametric registration - Concept

- Two families of non-rigid registration algorithms
  2. parametric
     - The deformation field **u** is parametrized using a mathematical function and its associated degrees of freedom **μ**
     - $x' = T(x; \boldsymbol{\mu})$
     - Usually less degrees of freedom than pixels (voxels)

Reference image

Degrees of freedom + Parametrisation Model =



Deformation field

Parametric registration - TPS

- Parametrisation through Thin plate spline
  - From a set of point $\mathbf{p}_i$ with $i \in [0;N[$
  - Each voxel position can be computed as:

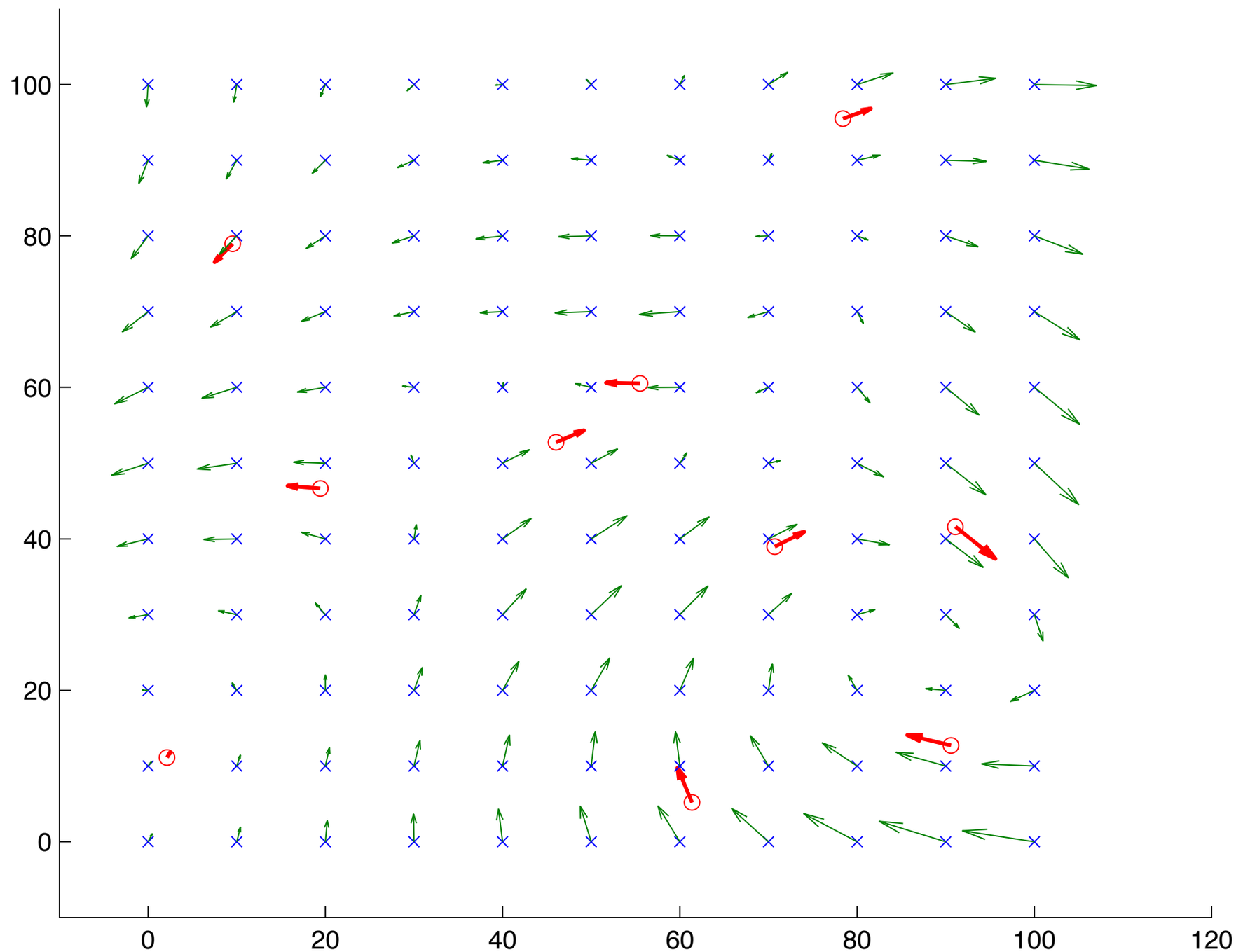
$$\mathbf{T}(\vec{x}) = \sum_{i=0}^{N-1} \vec{c}_i \times \phi(\|\vec{x} - \vec{p}_i\|)$$

where $\boldsymbol{c}_i$ are coefficients computed from the positions of the points $\mathbf{p}_i$ (minimising the curvature) and

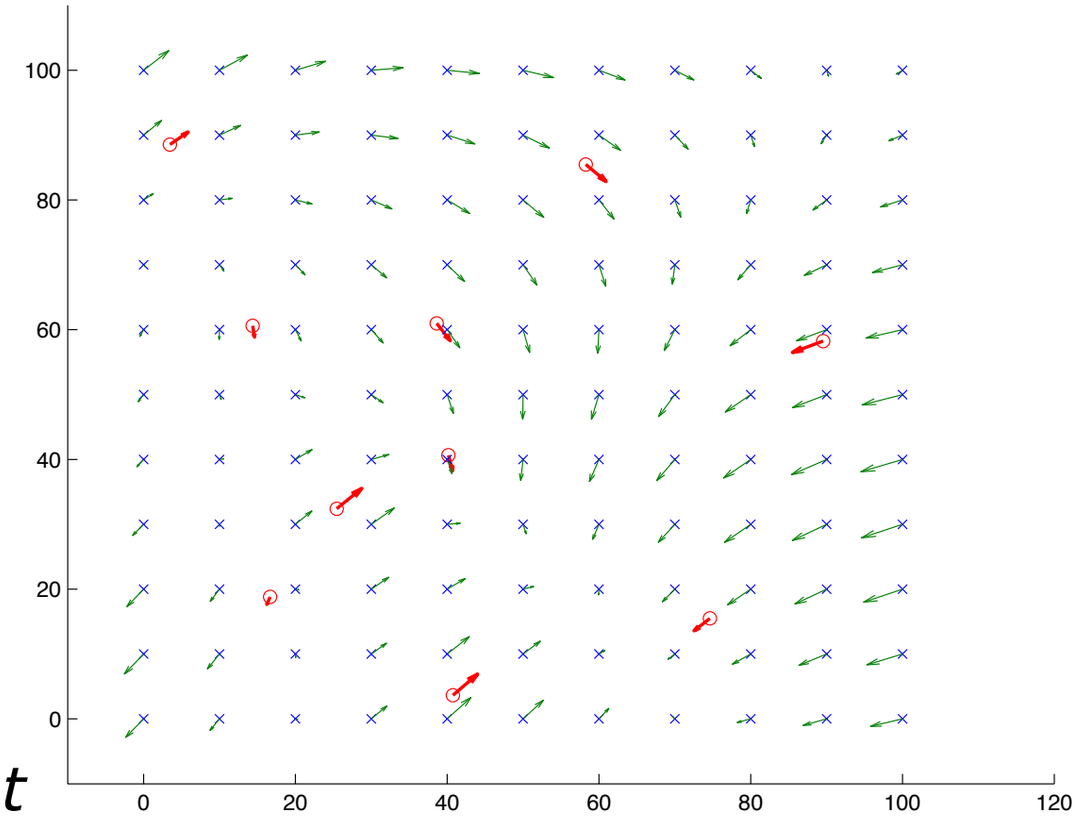
$$\phi(d) = d^2 \log(d)$$

➡ Smooth transformation
➡ Continuous first order derivatives

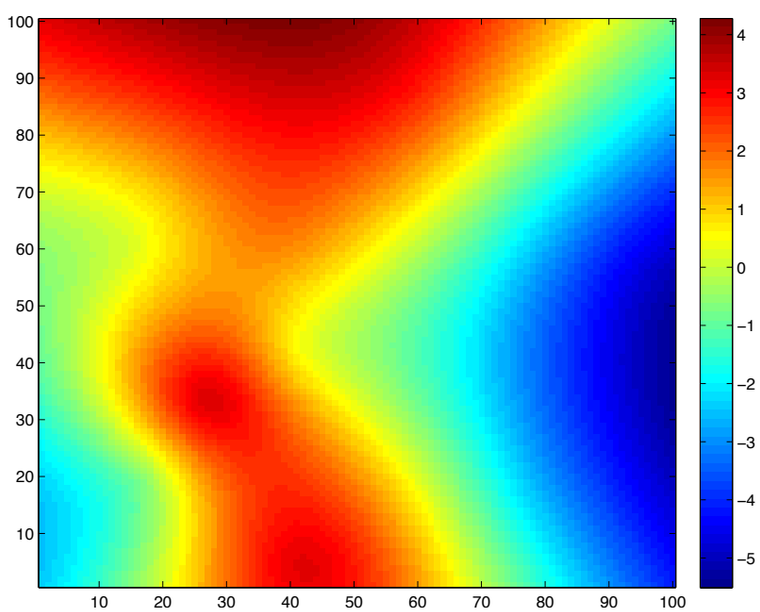
Parametric registration - TPS



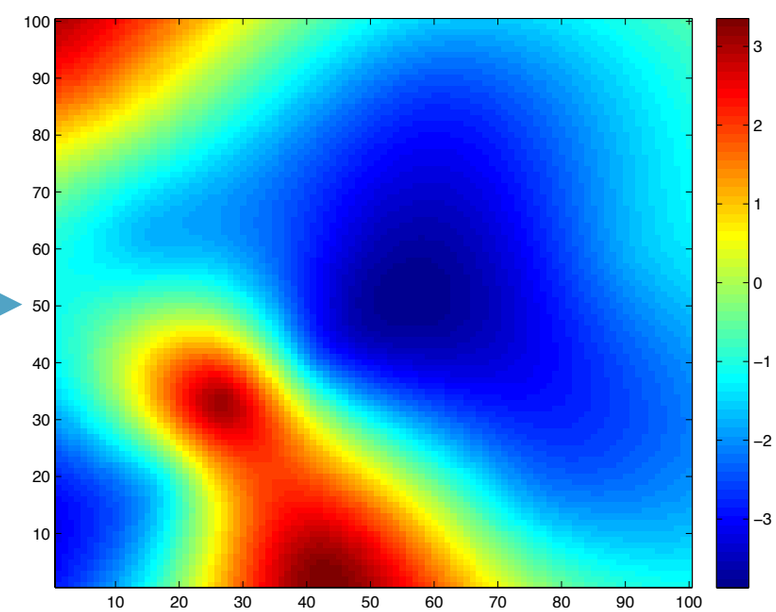
Parametric registration - TPS



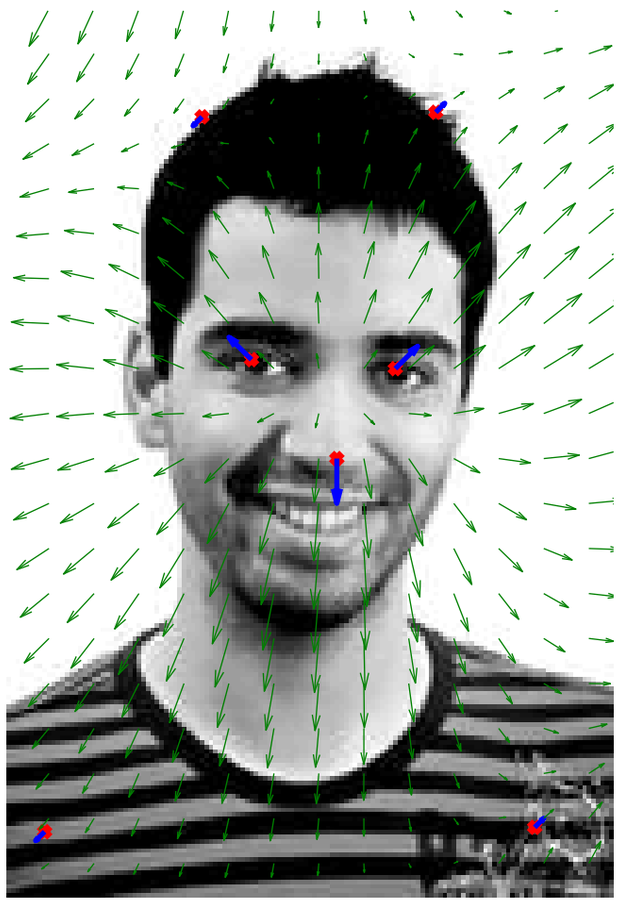
x-axis displacement

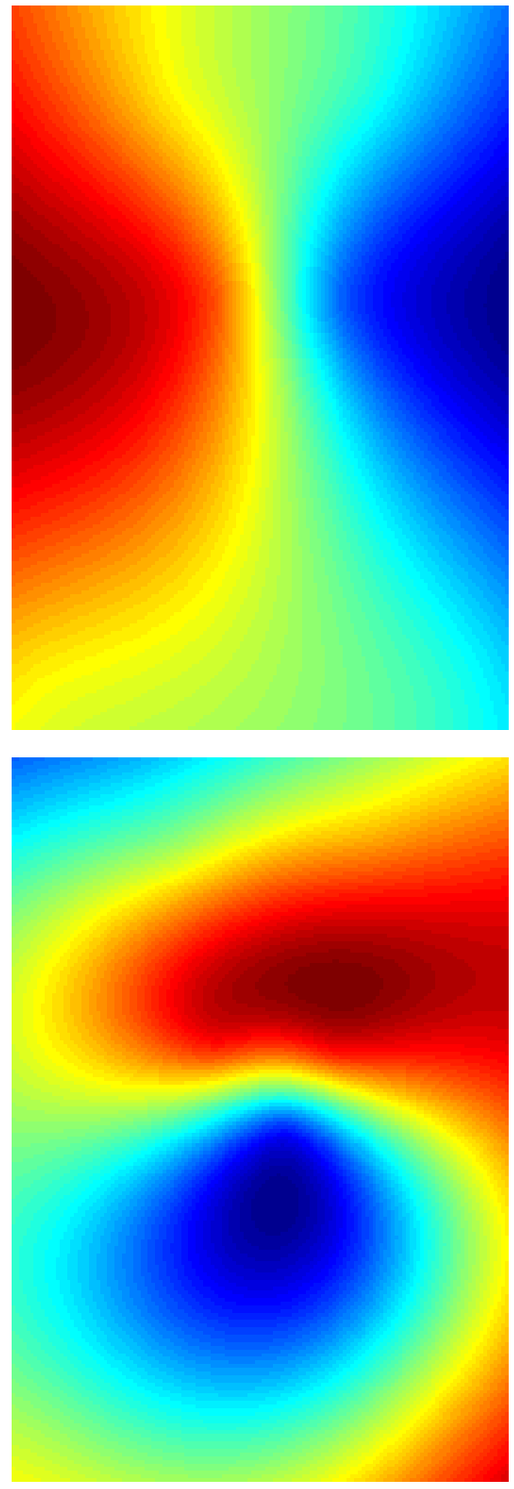


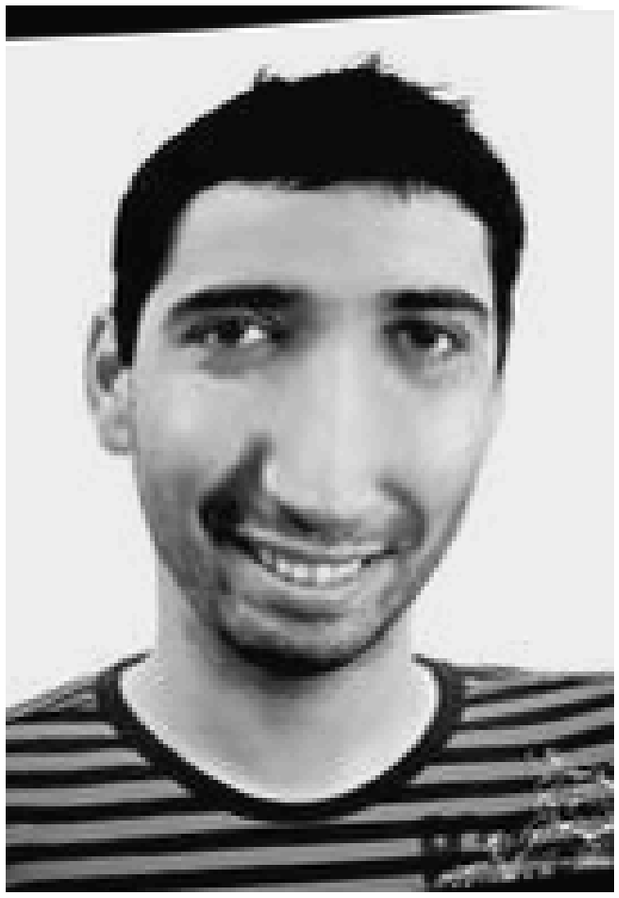
y-axis displacement



Parametric registration - TPS



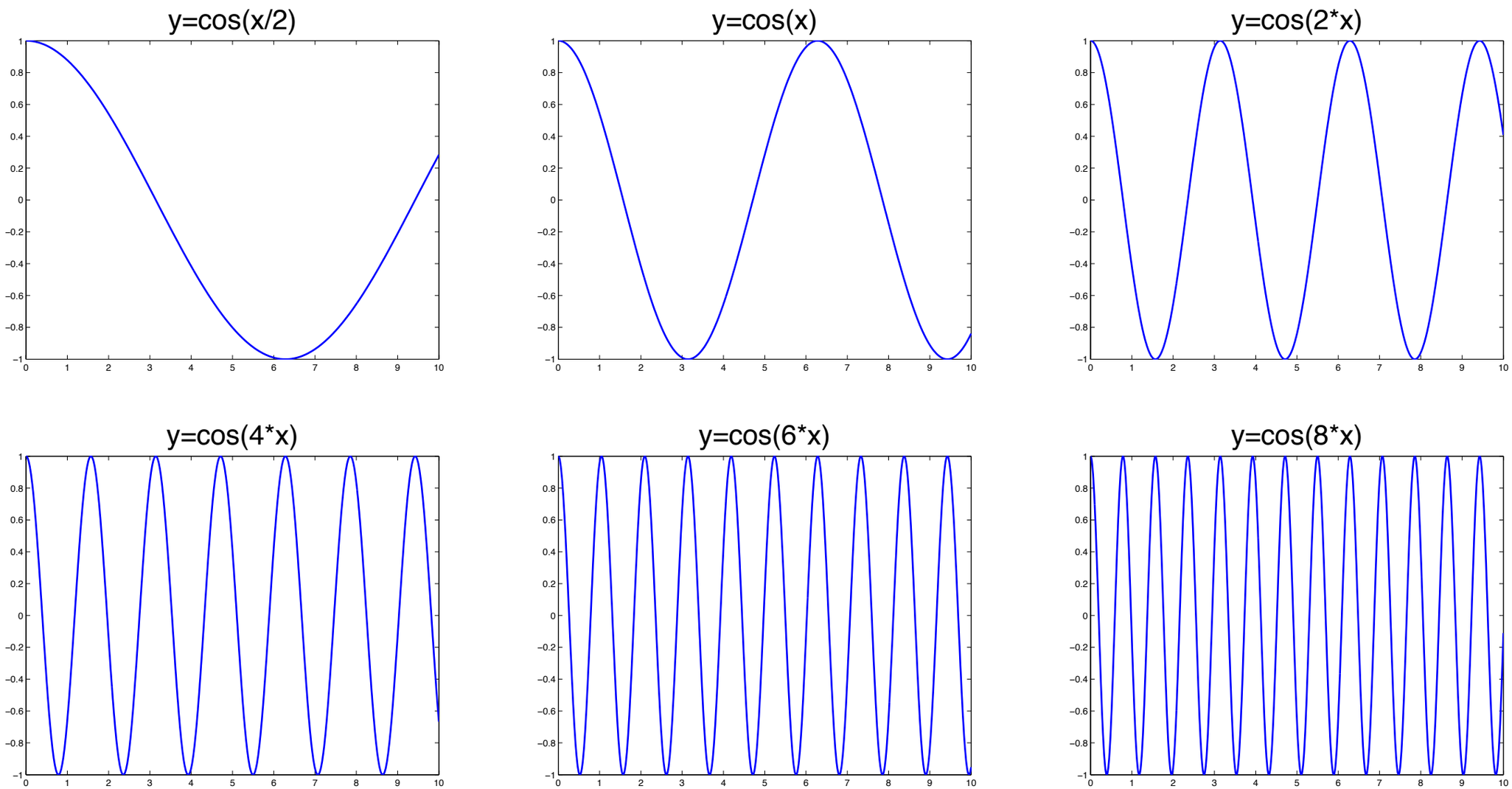


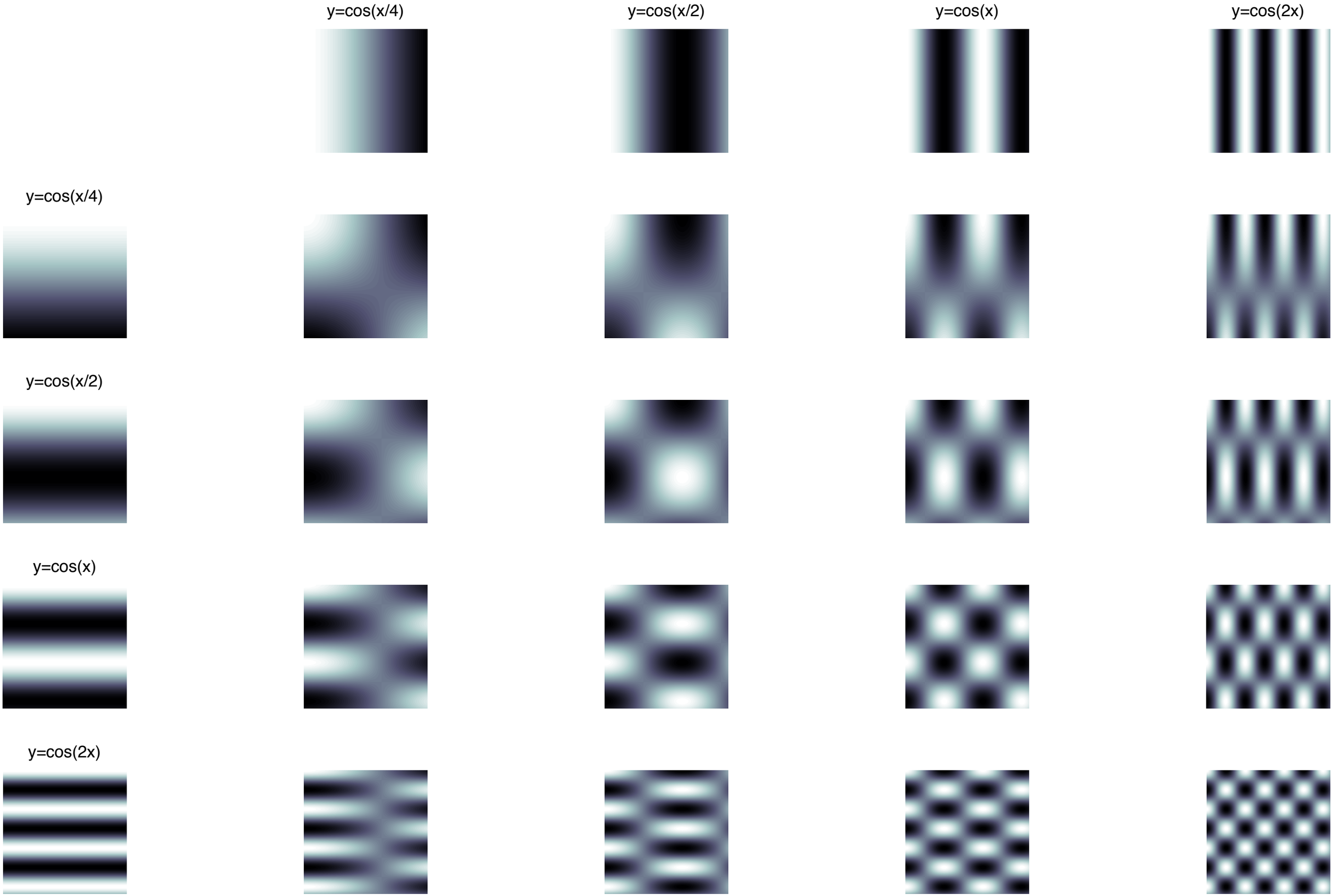


Parametric registration - TPS

Parametric registration - Basis function

- Deformation field can be parametrised using a set of Discrete Cosine Transform functions

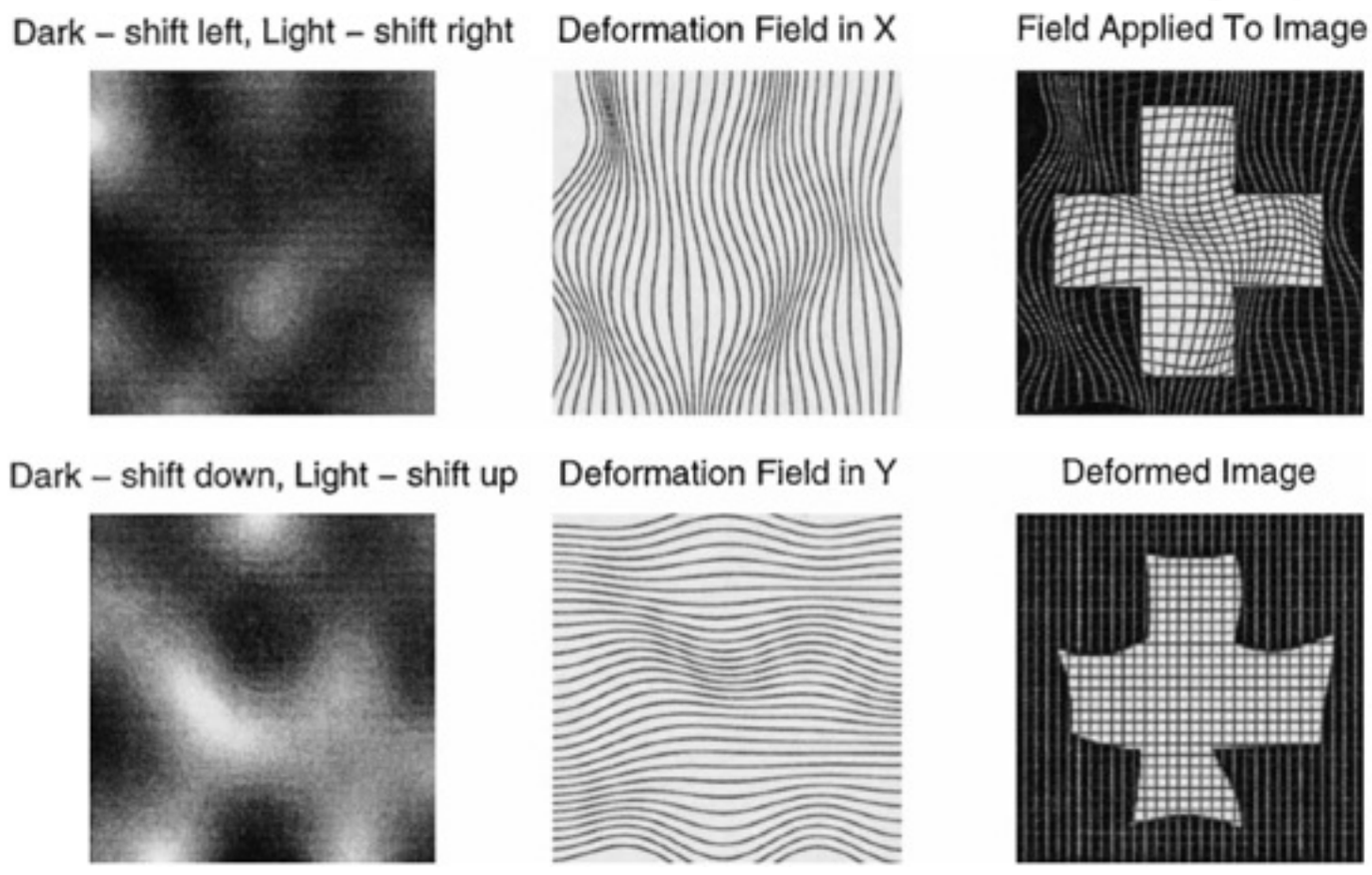



y=cos(x/4)
y=cos(x/2)
y=cos(x)
y=cos(2x)
y=cos(x/4)
y=cos(x/2)
y=cos(x)
y=cos(2x)

Parametric registration - Basis function

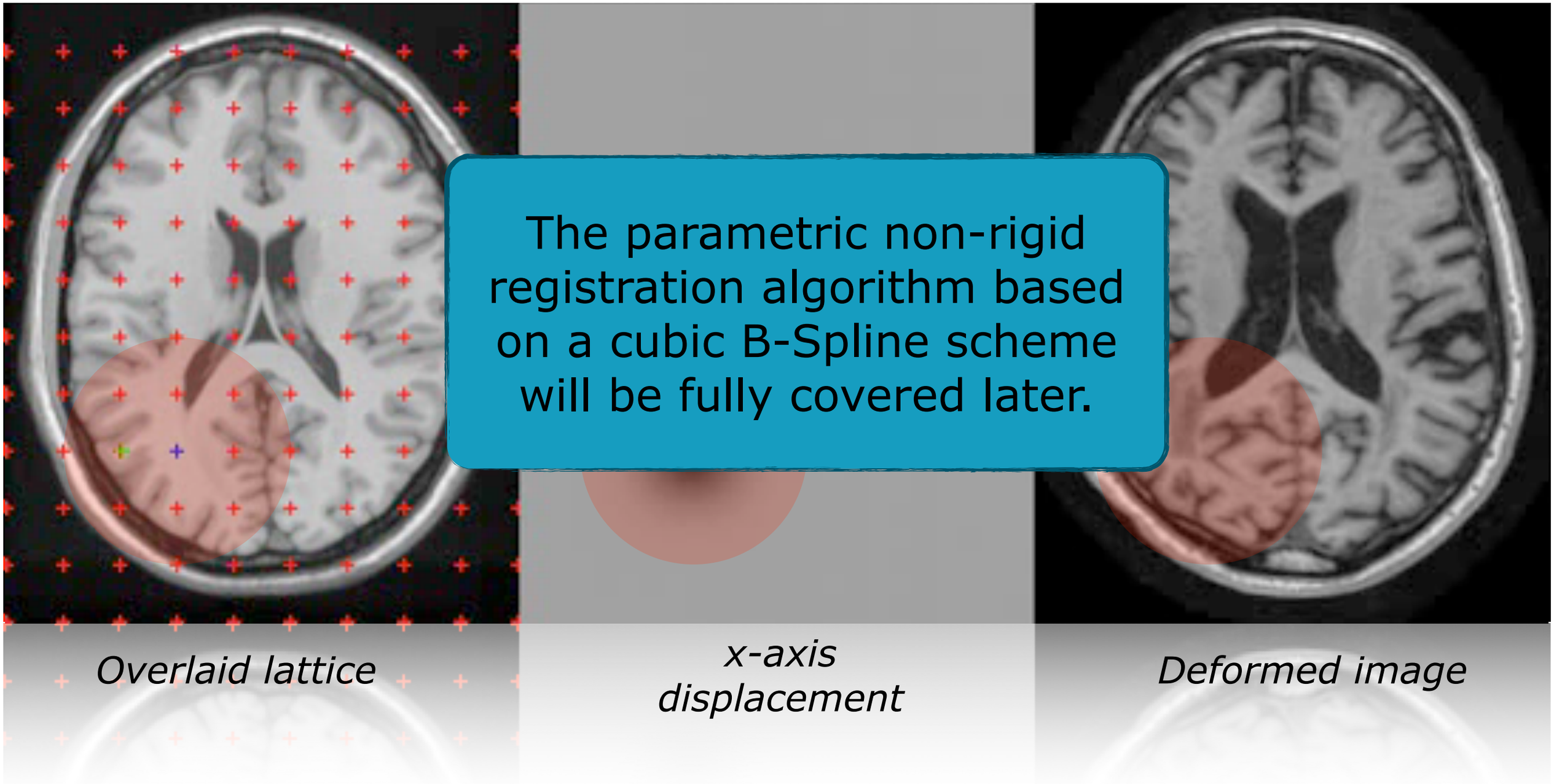
- A coefficient *t* is applied to each basis function

- Only the coefficients have to be optimised
  - fewer degrees of freedom
  - (usually about 1000 for a brain MR image)



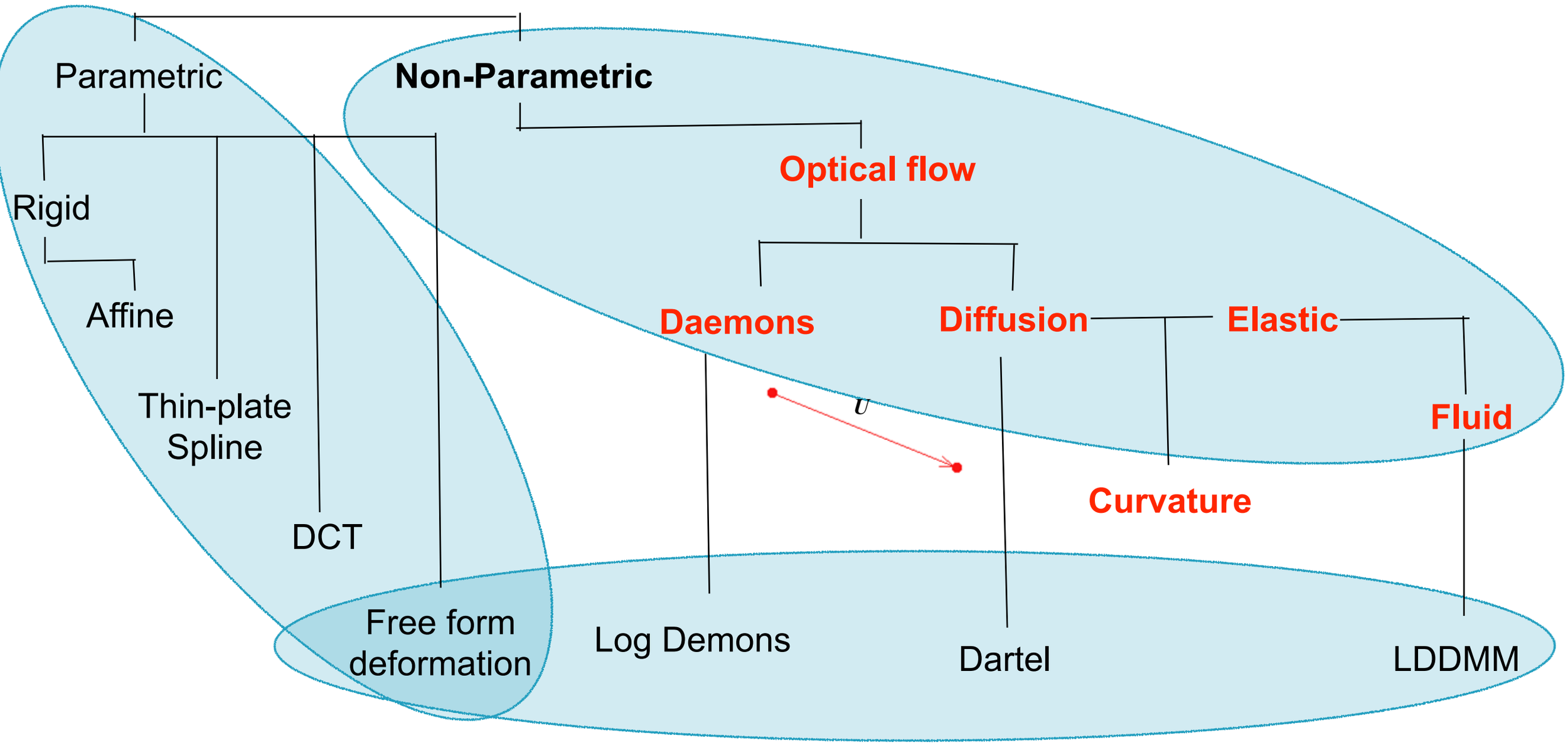
From Ashburner and Friston. Nonlinear spatial normalization using basis functions. Hum. Brain Mapp. (1999) vol. 7 pp. 254-266

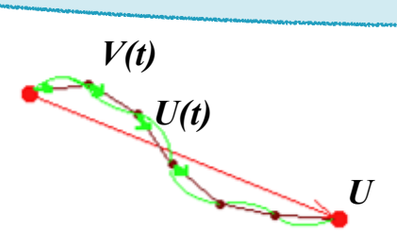
Parametric registration - Cubic B-Spline

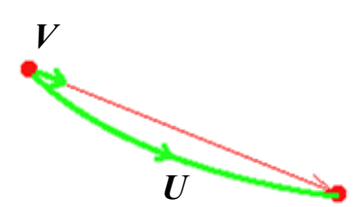


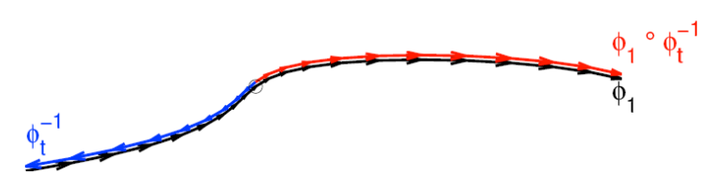
The parametric non-rigid registration algorithm based on a cubic B-Spline scheme will be fully covered later.

A taxonomy of reg. algorithms



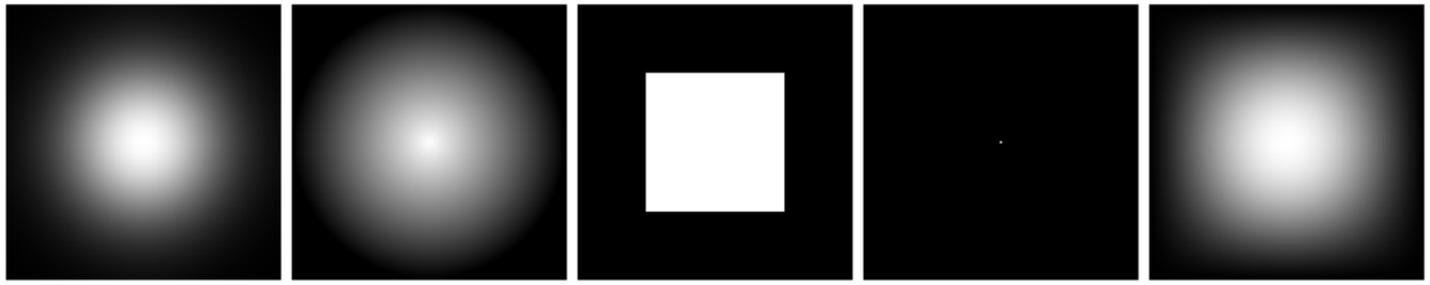


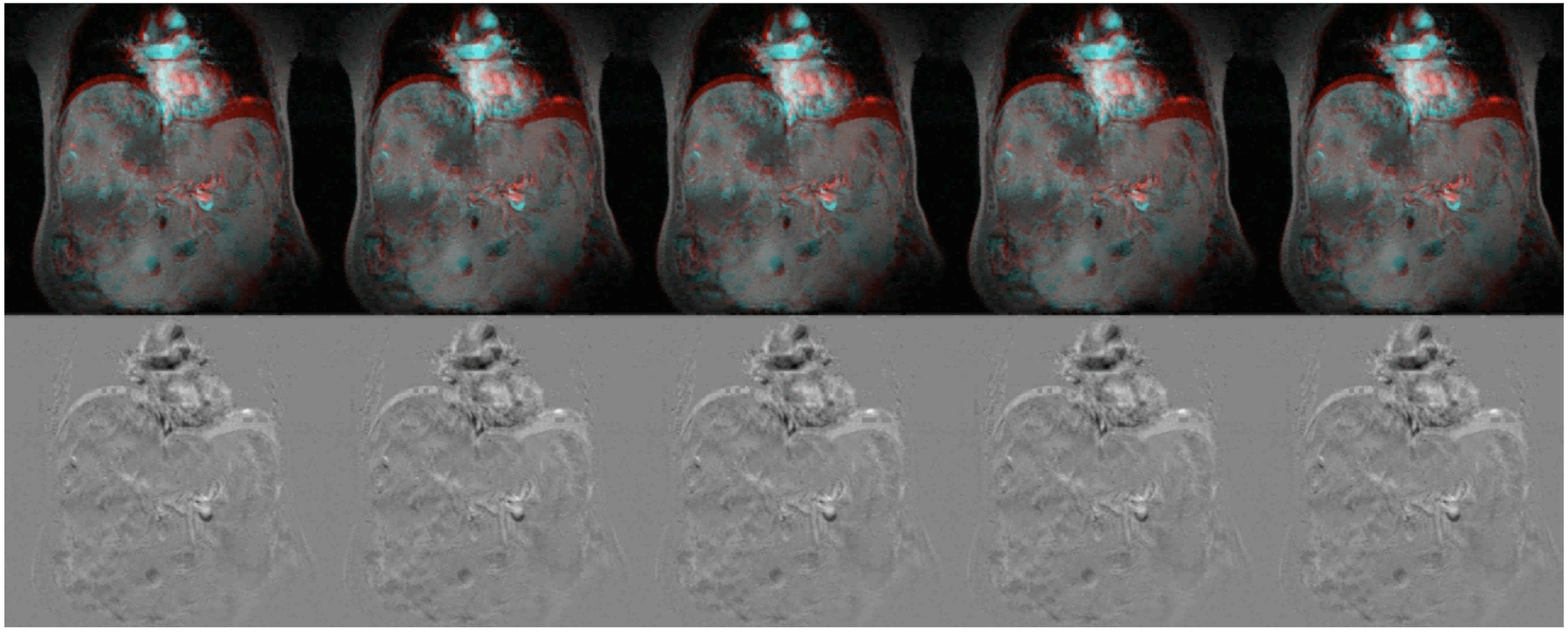




Optical flow

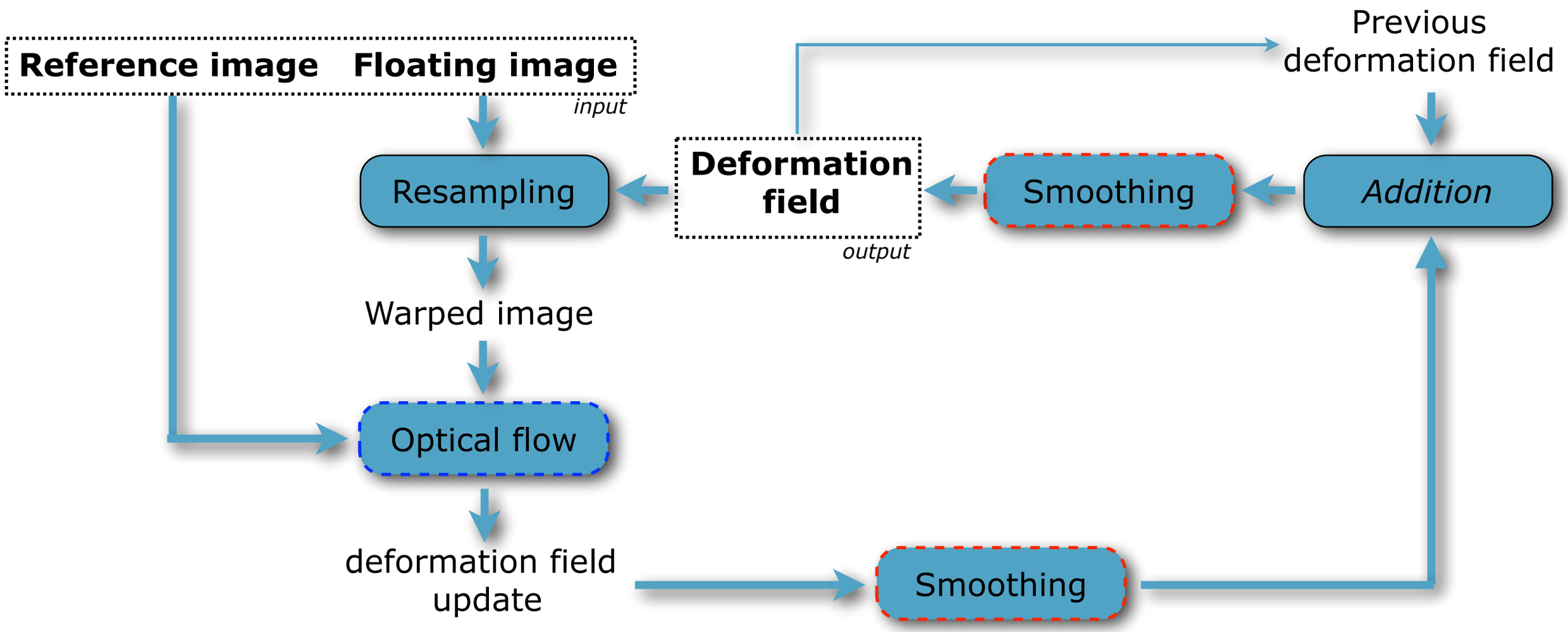
- Minimising the SSD
  - Need for regularisation
    - e.g. convolutions





The demons algorithm

- Regularised by Gaussian smoothing
  - Elastic-like behavior
  - Fluid-like behavior



The demons algorithm

- Recursive smoothing method
  - Elastic like

$$\mathbf{u}^{t+1} = \mathbf{G} * (\mathbf{u}^t + \delta\mathbf{u})$$

  - Fluid-like

$$\mathbf{u}^{t+1} = \mathbf{u}^t + \mathbf{G} * \delta\mathbf{u}$$

PDE-based registration methods

- Diffusion equation

$$\int \left[(A - B(\mathbf{u}))^2 + ||\nabla \mathbf{u}||_2^2\right] \delta\Omega$$

$$(\frac{\delta U_1}{\delta x})^2 + (\frac{\delta U_1}{\delta y})^2 + (\frac{\delta U_2}{\delta x})^2 + (\frac{\delta U_2}{\delta y})^2$$

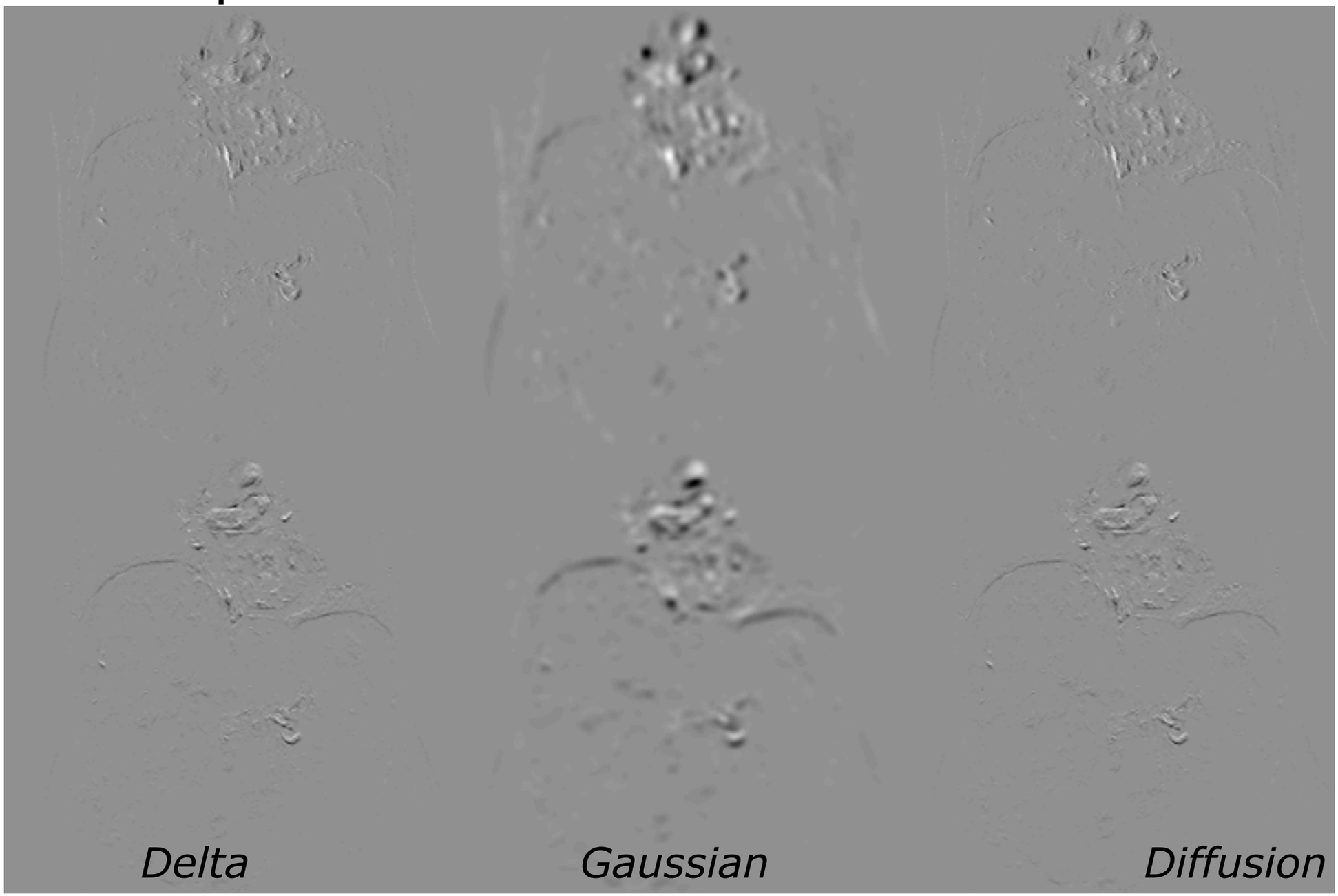
$$\mathbf{u}_1 \qquad \mathbf{u}_2$$

$$(A - B(\mathbf{u}))\nabla B(\mathbf{u}) = \nabla^2 \mathbf{u}$$

- Multiple ways to solve a PDE
  - » Successive over-relaxation
  - » Multi-grid approach
  - » FFT

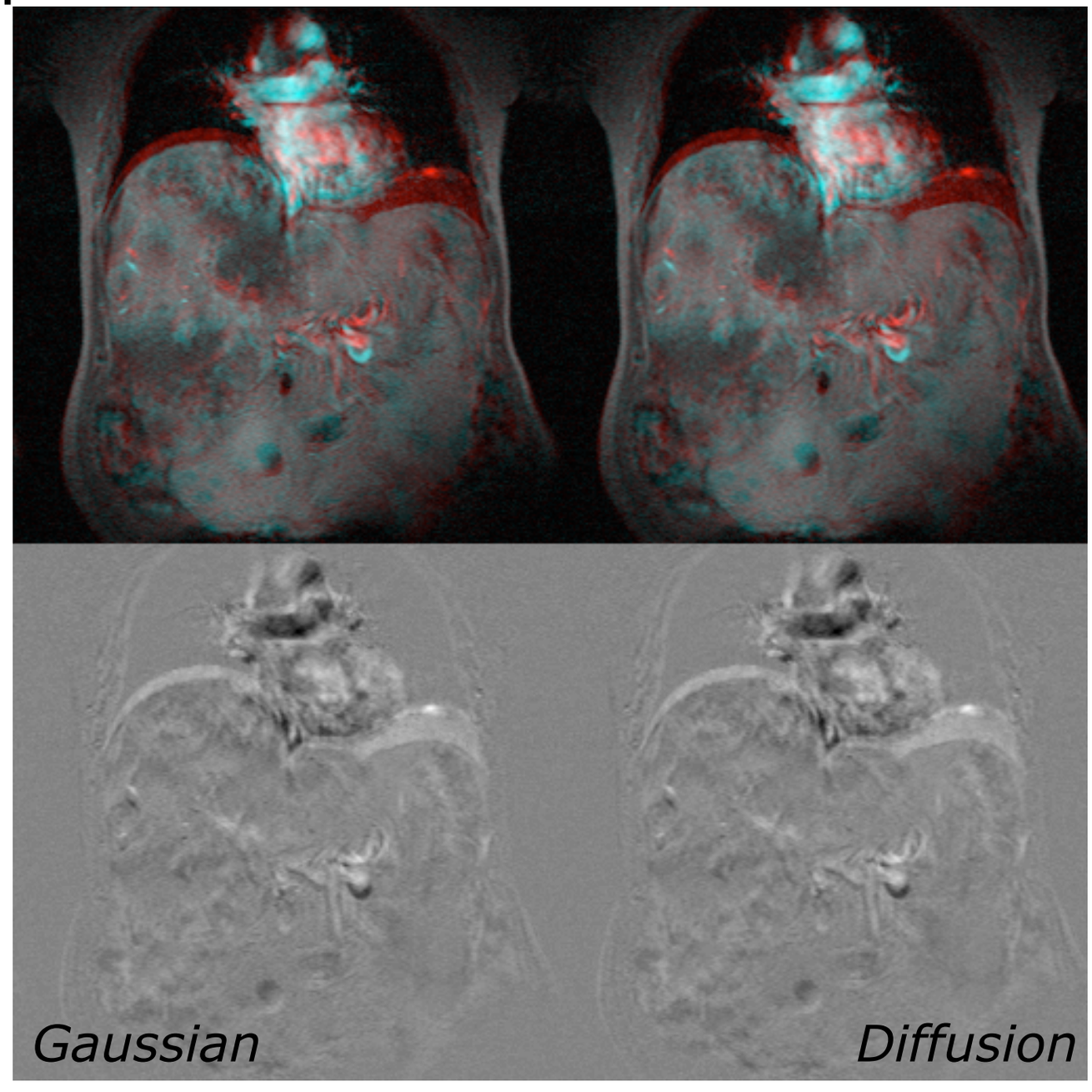
PDE-based registration methods

- Diffusion equation



PDE-based registration methods

- Diffusion equation



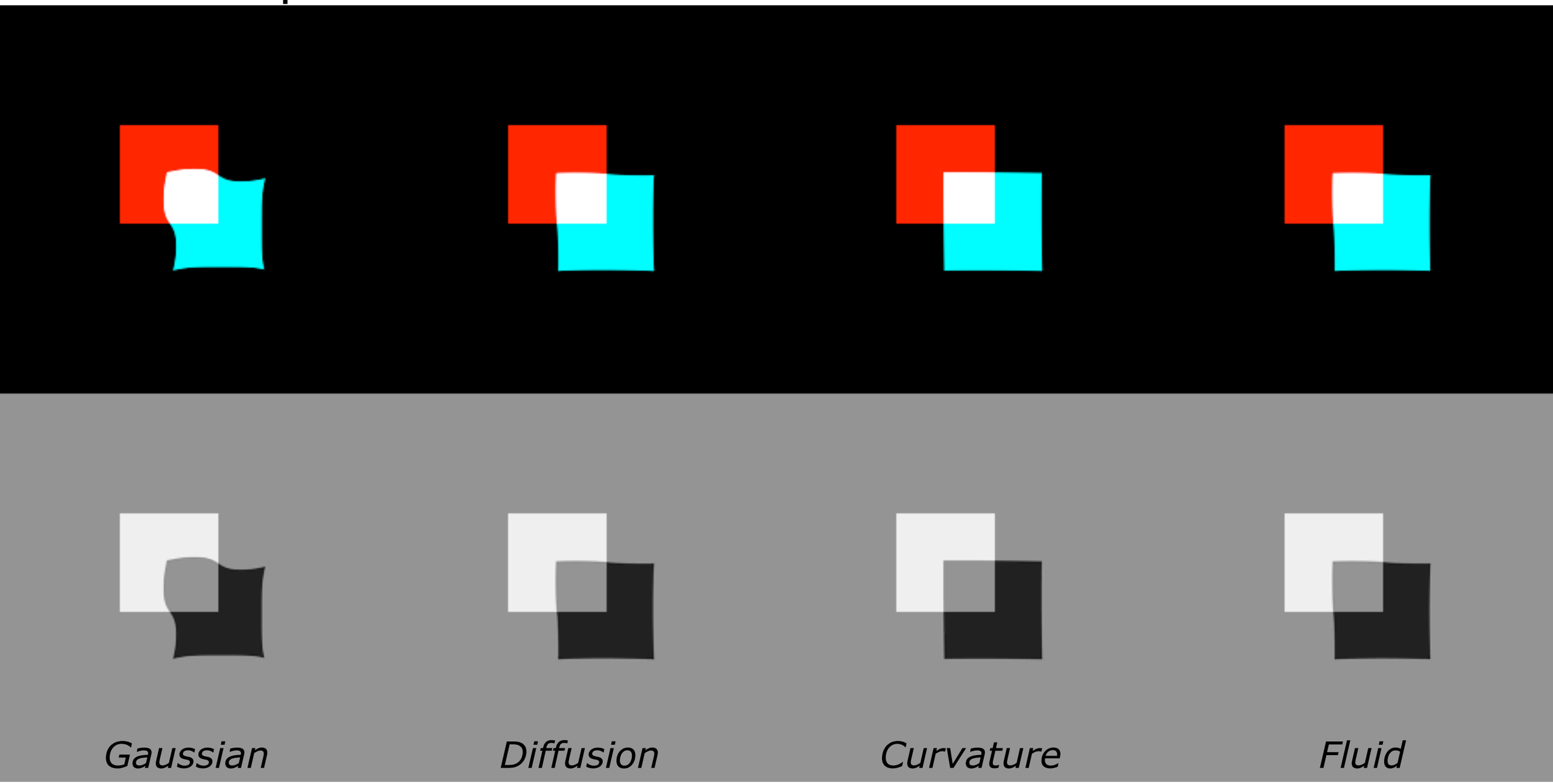
PDE-based registration methods

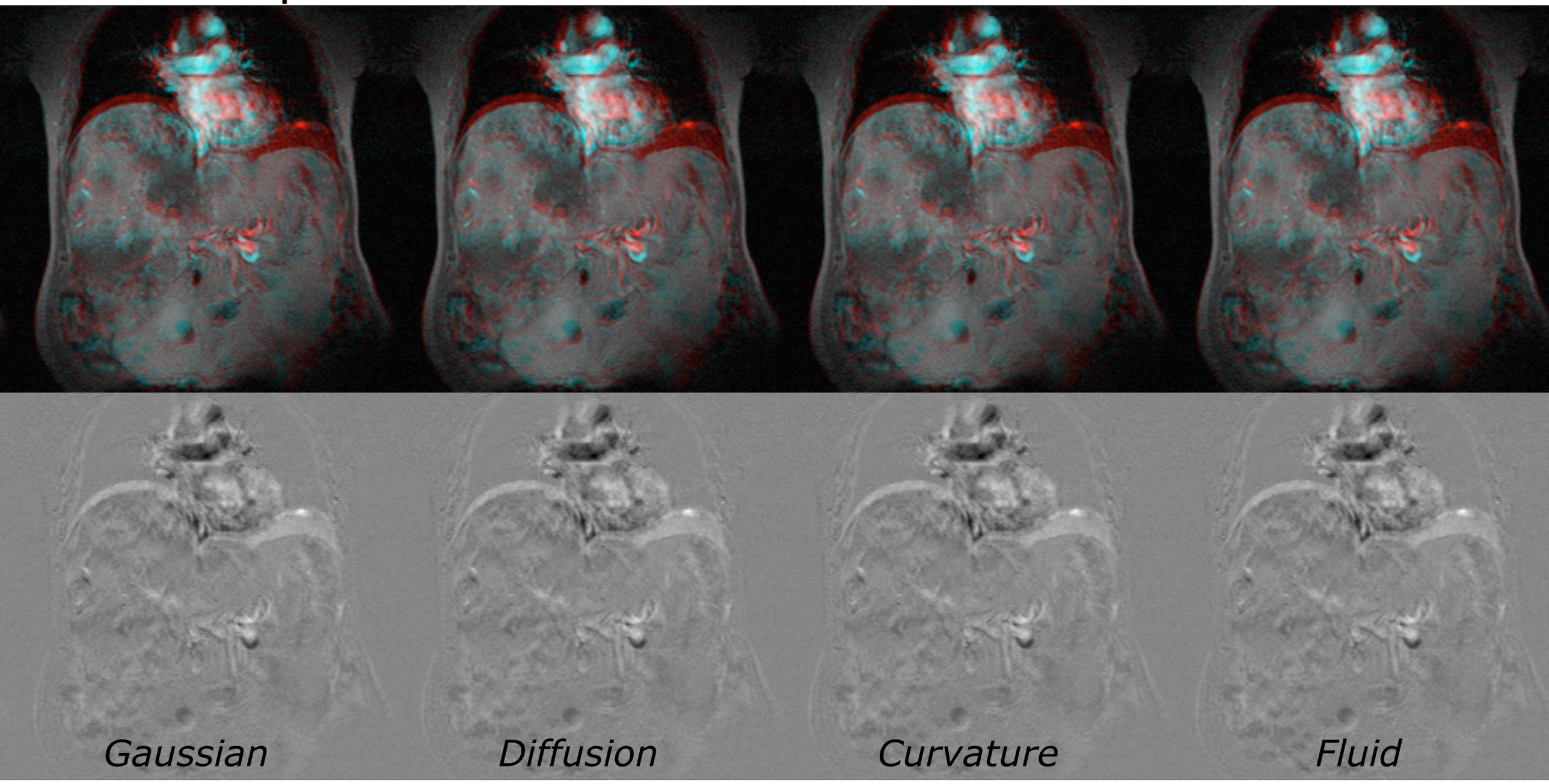
- Fluid equation

$$(A - B(\mathbf{u}))\nabla B(\mathbf{u}) = \mu\nabla^2\mathbf{v} + (\mu + \lambda)\nabla(\nabla \cdot \mathbf{v})$$

PDE-based registration methods

- Fluid equation



PDE-based registration methods

- Fluid equation