Image Registration / Alignment

Suyash P. Awate

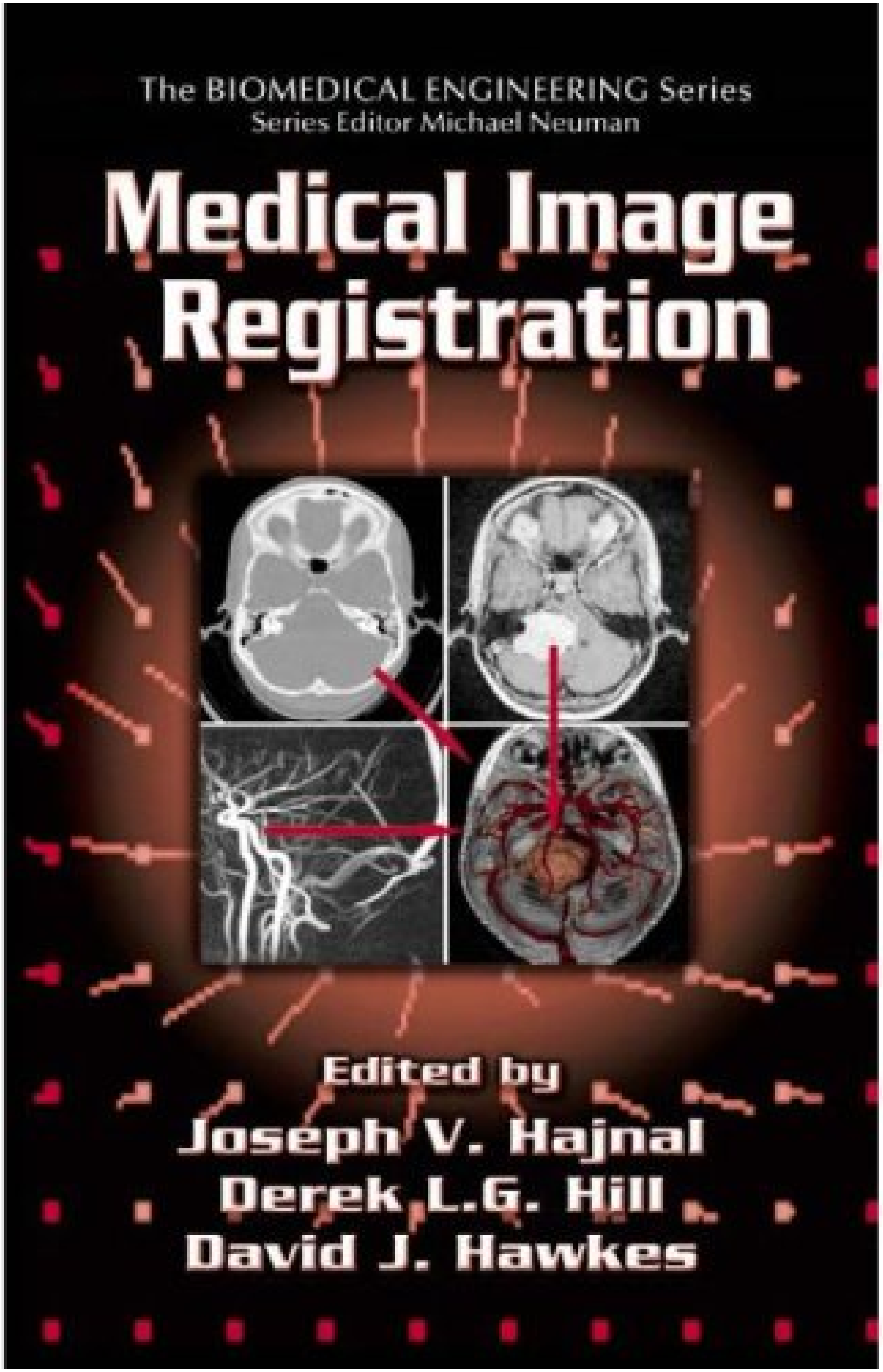


Image Registration

- Given: 2 images
- Goal: “Transform” one image to align to the other
- Components

  (1) Model for spatial deformation / warp

  - Geometric transformation on the spatial domain

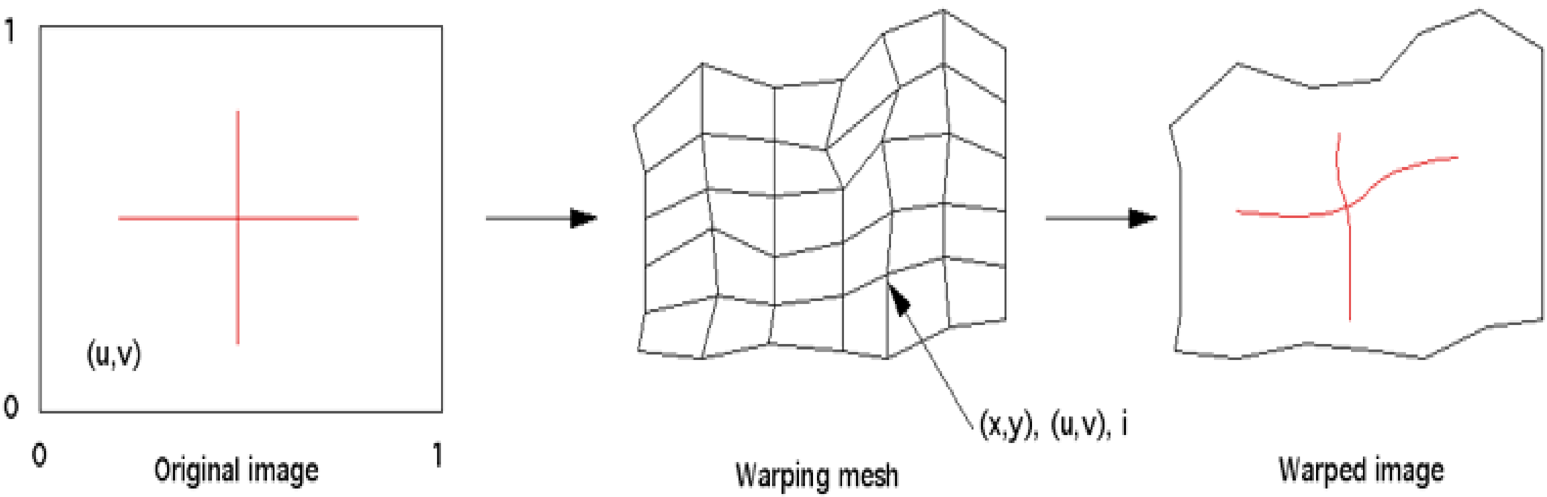
  (2) Similarity measure between images

Image Registration

- Components

  (1) Model for spatial deformation / warp

  - Geometric transformation on the spatial domain



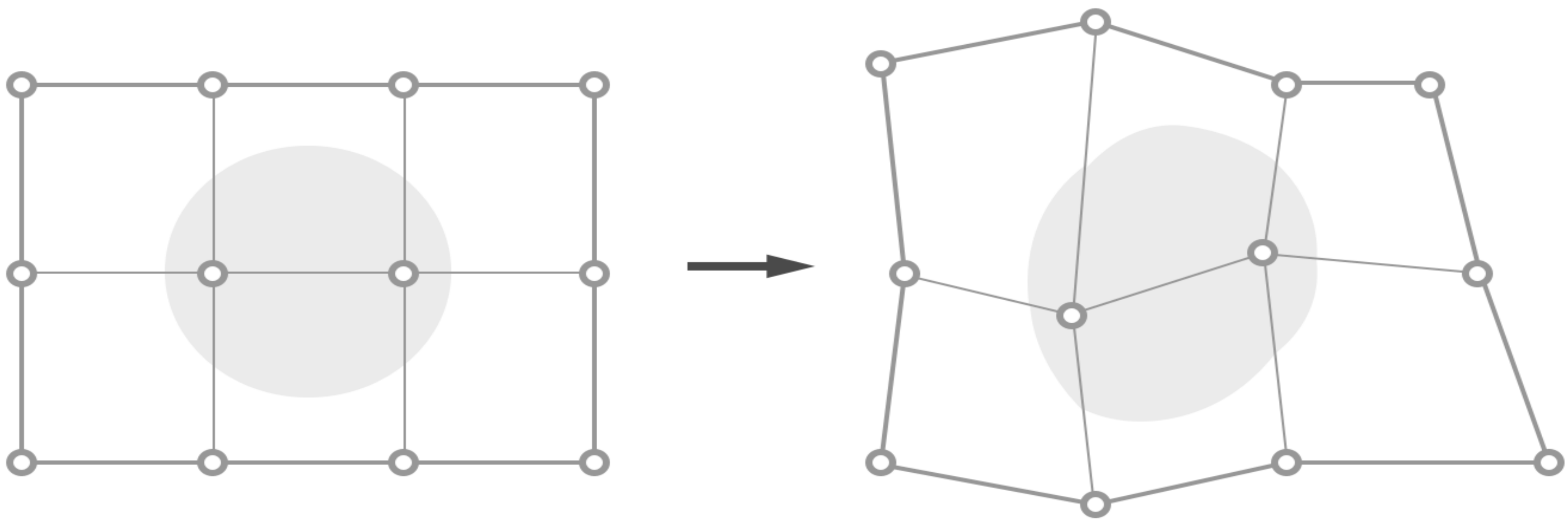


Image Registration

- Components

  (1) Model for spatial deformation / warp

  - Geometric transformation on the spatial domain

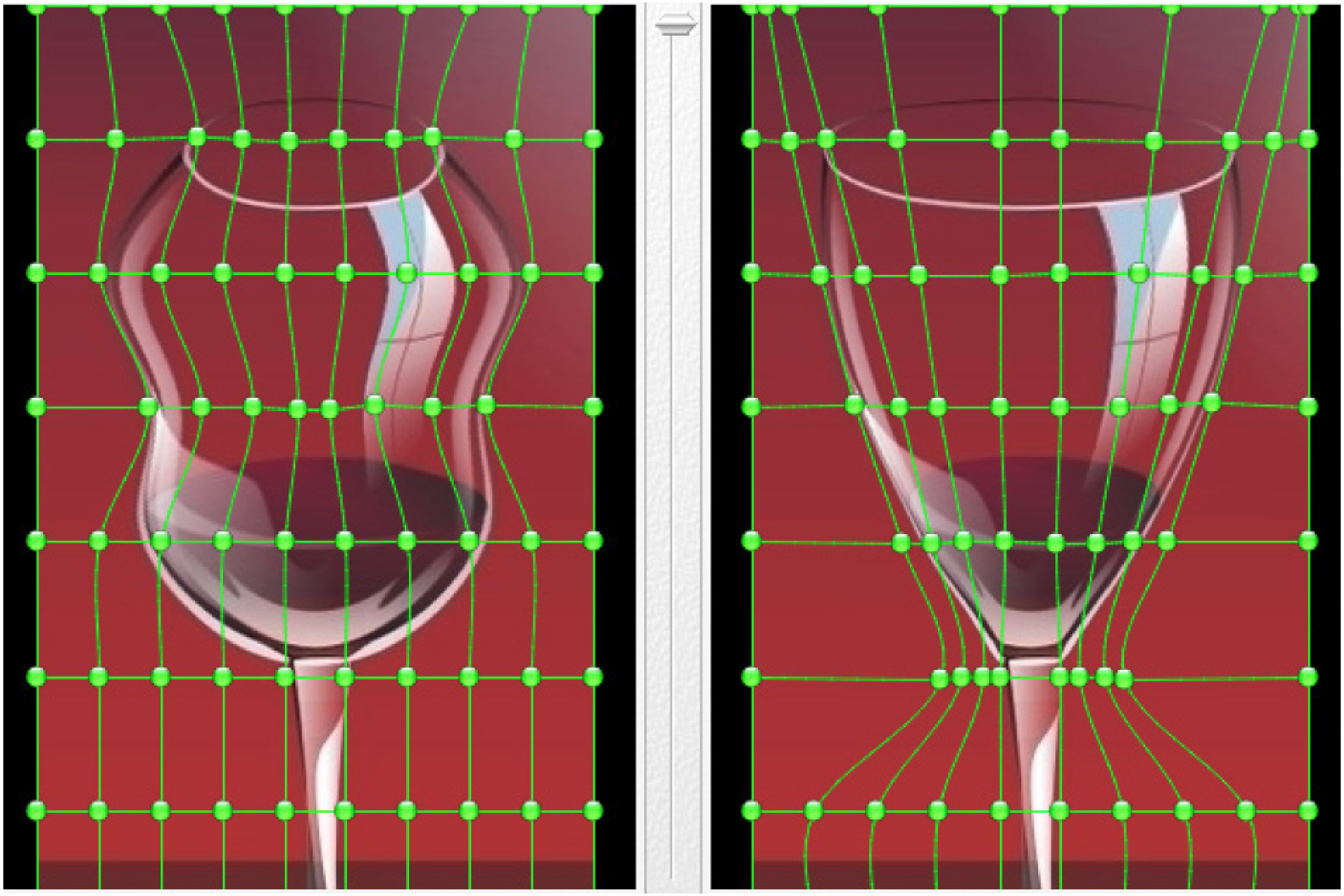


Image Registration

- Components
  - (1) Model for spatial deformation / warp
    - Geometric transformation on the spatial domain
      - Moving control points

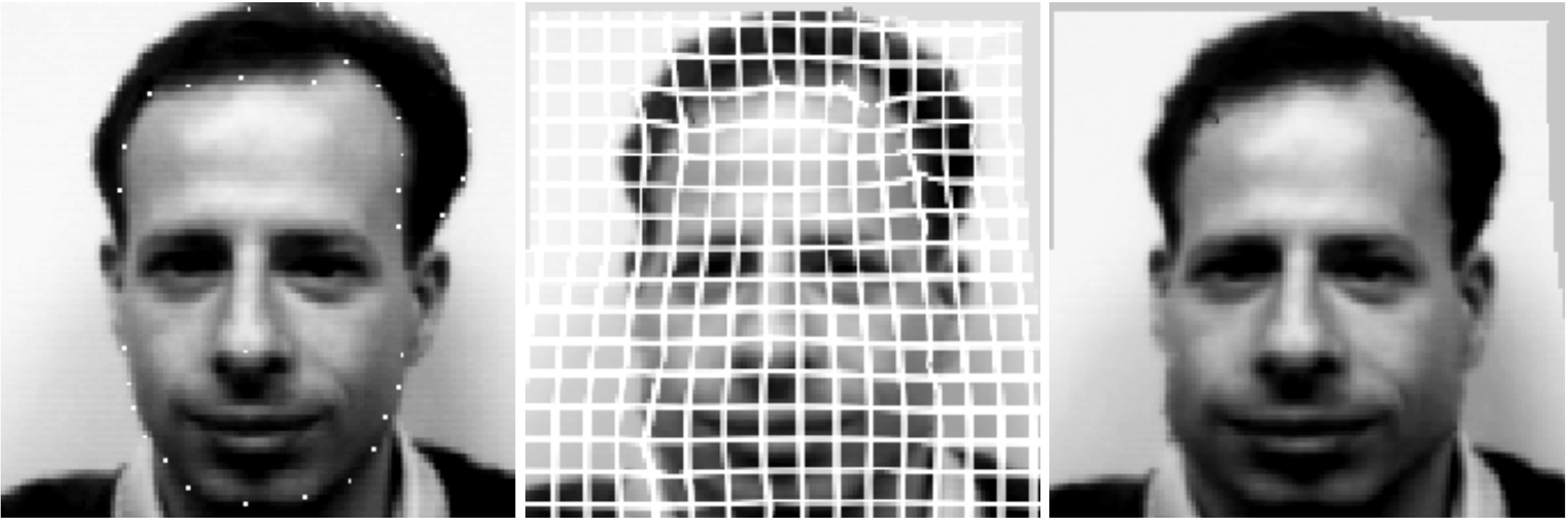


Image Registration

- Components

  (1) Model for spatial deformation / warp

  - Geometric transformation on the spatial domain

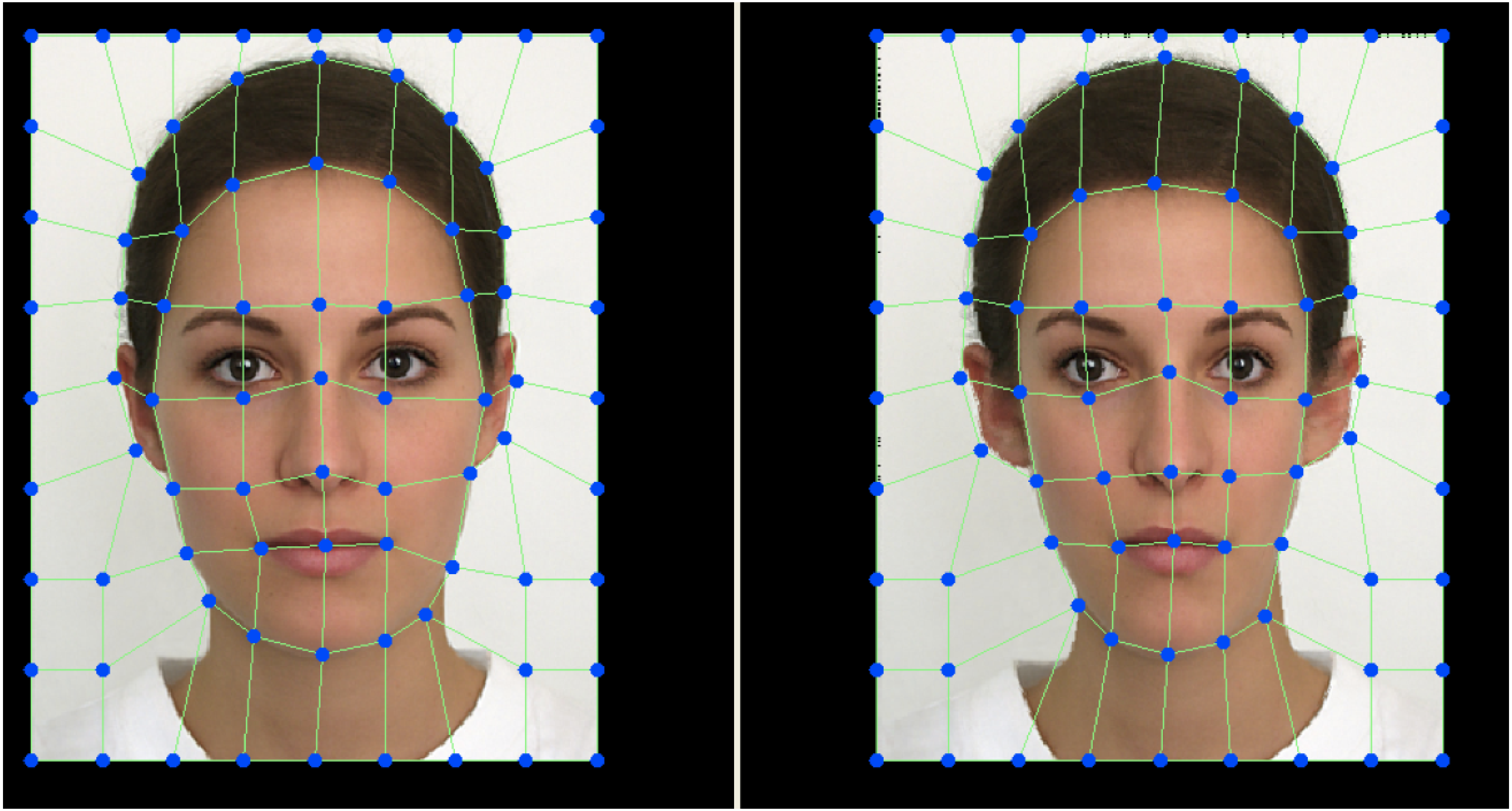


Image Registration

- Components

  (1) Model for spatial deformation / warp

  - Geometric transformation on the spatial domain

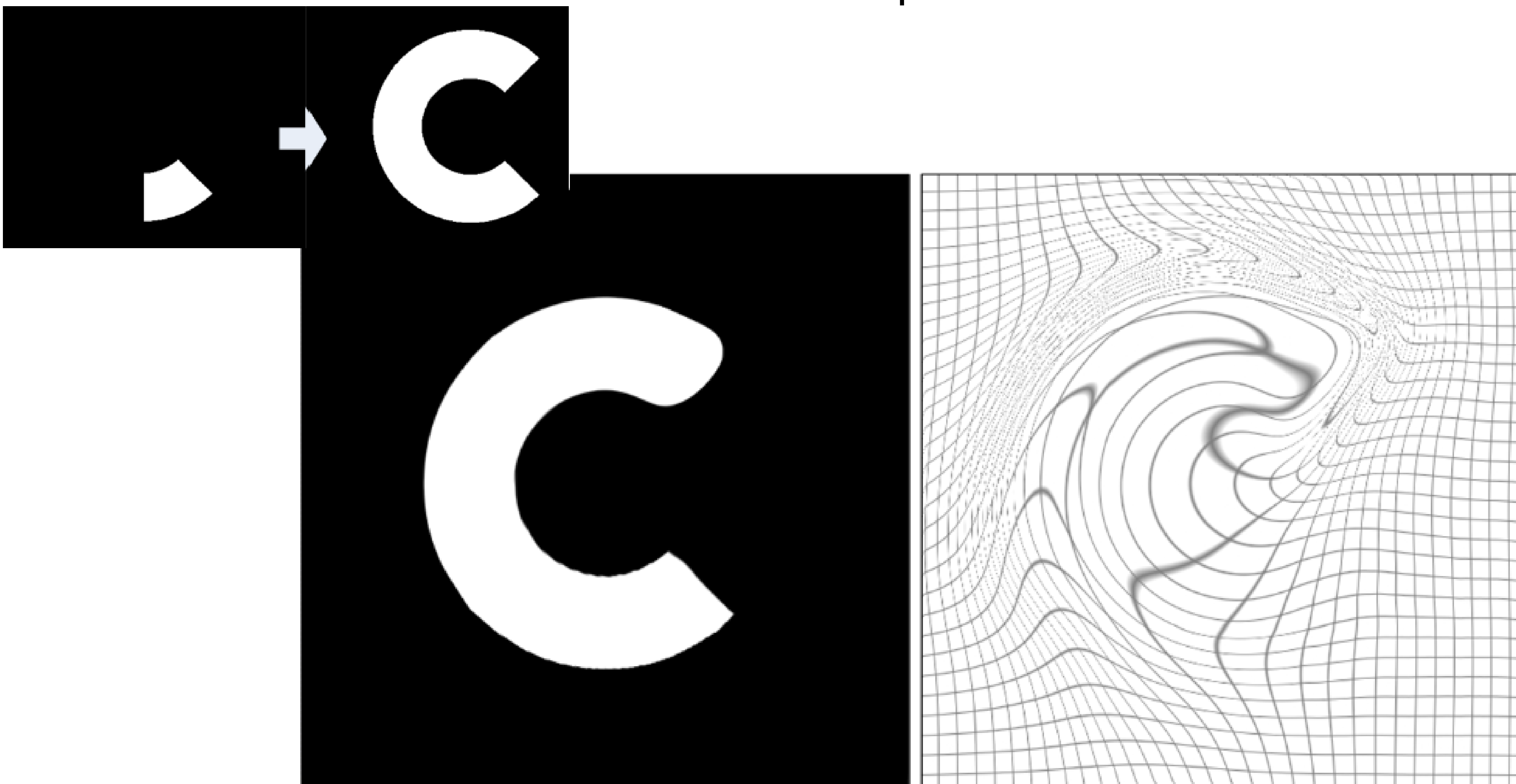


Image Registration

- Components

  (1) Model for spatial deformation

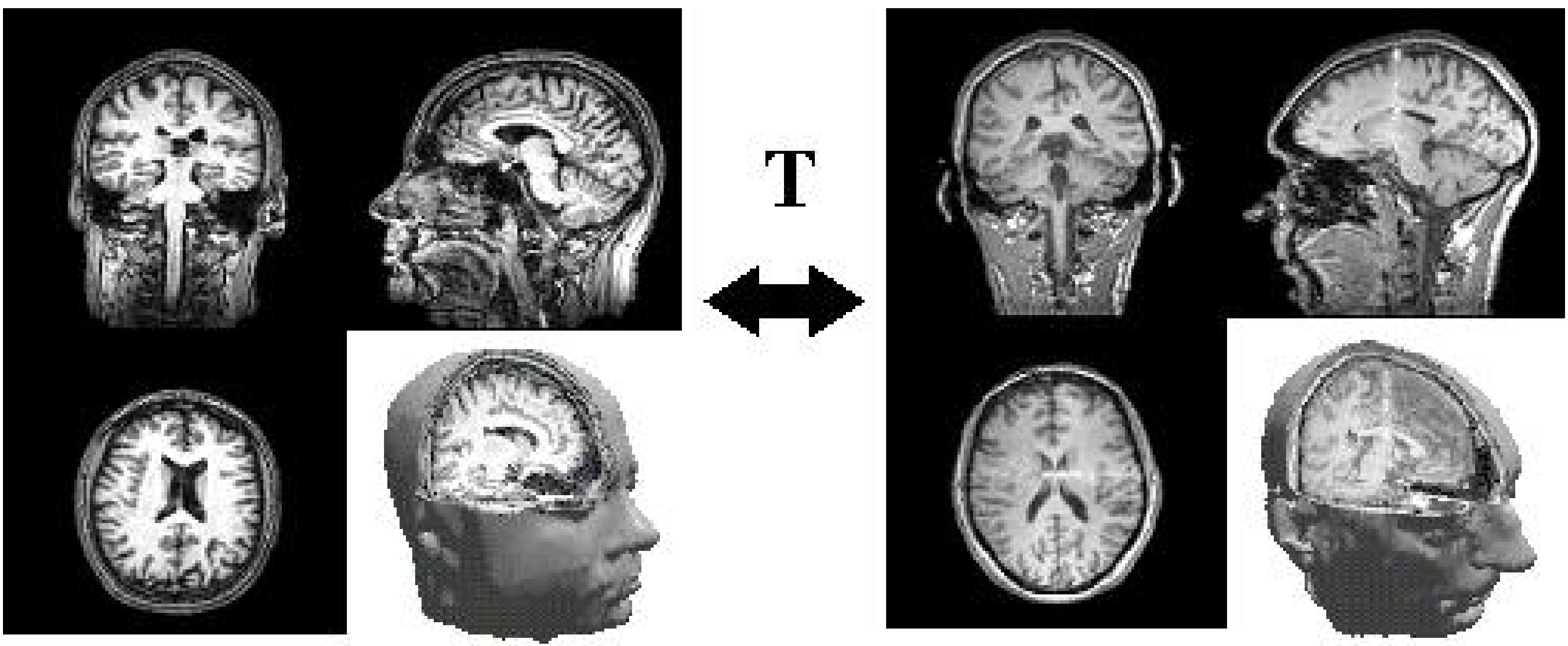
  - Rigid transformation
  - Similarity transformation
  - Linear transformation
  - Nonlinear transformation
    - Diffeomorphic (smooth, invertible)
    - Parametric, e.g., B spline
    - Nonparametric / dense, e.g., vector field

Image Registration

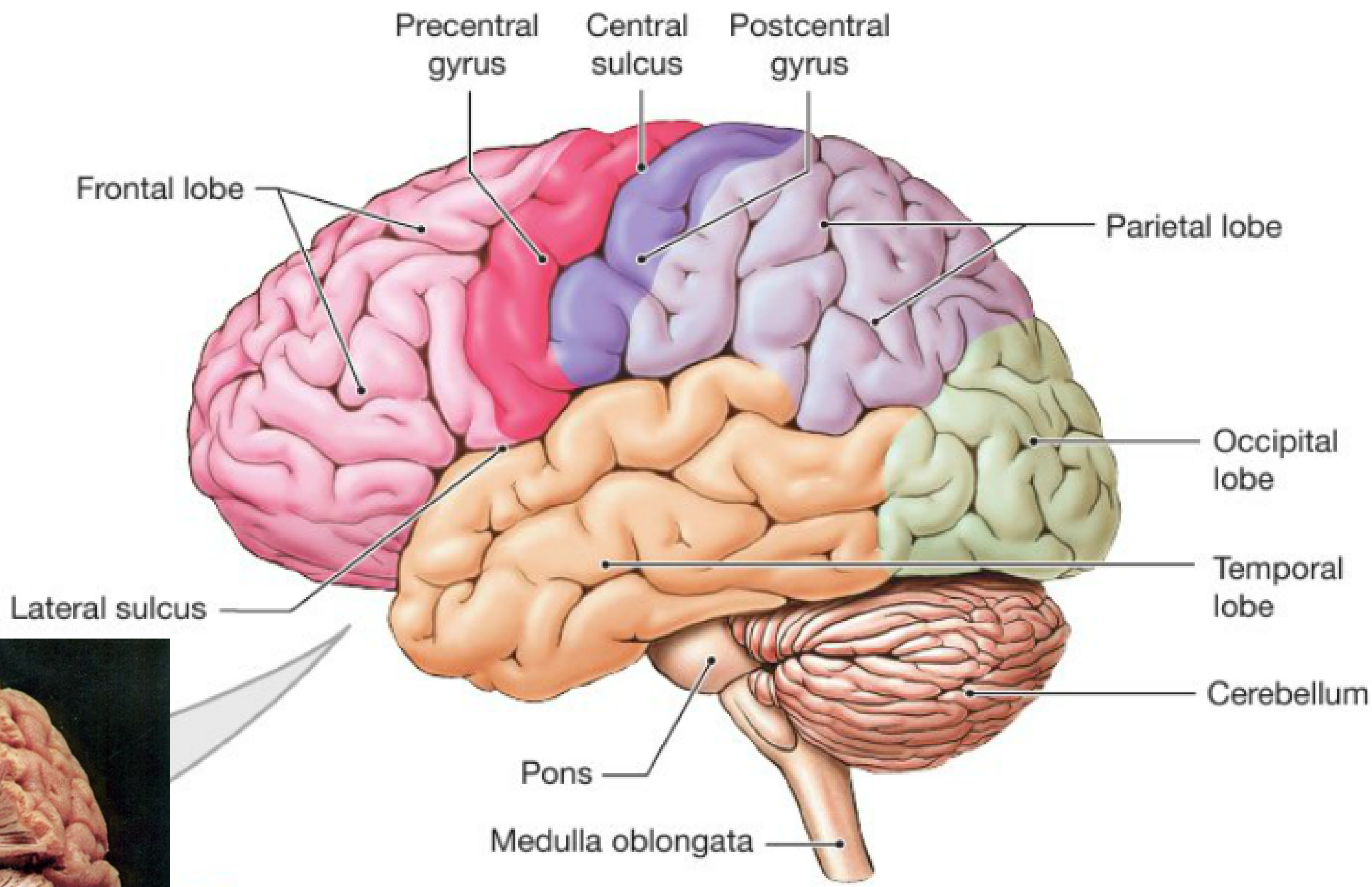
- Components

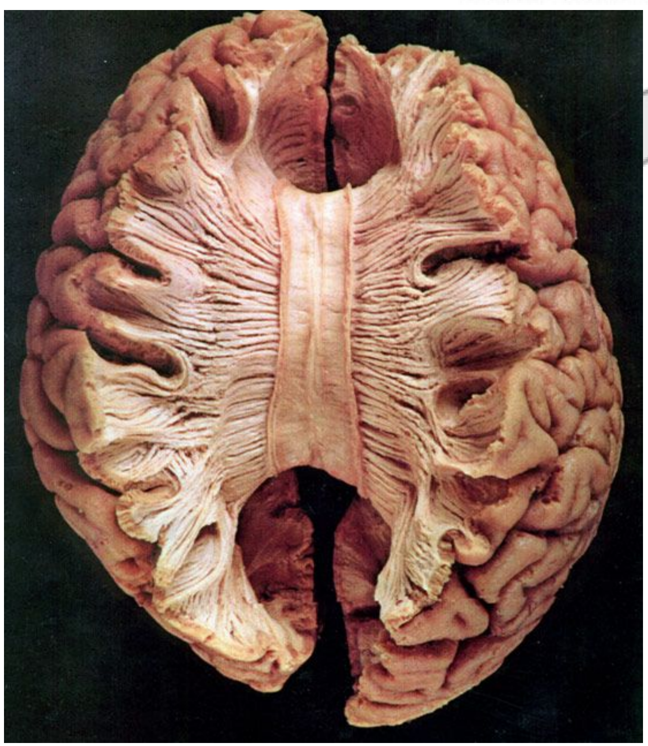
  (2) Similarity measure between images
  (when the task is of image alignment)

  - Are images of same modality ?
    - Squared error between corresponding pixel intensities

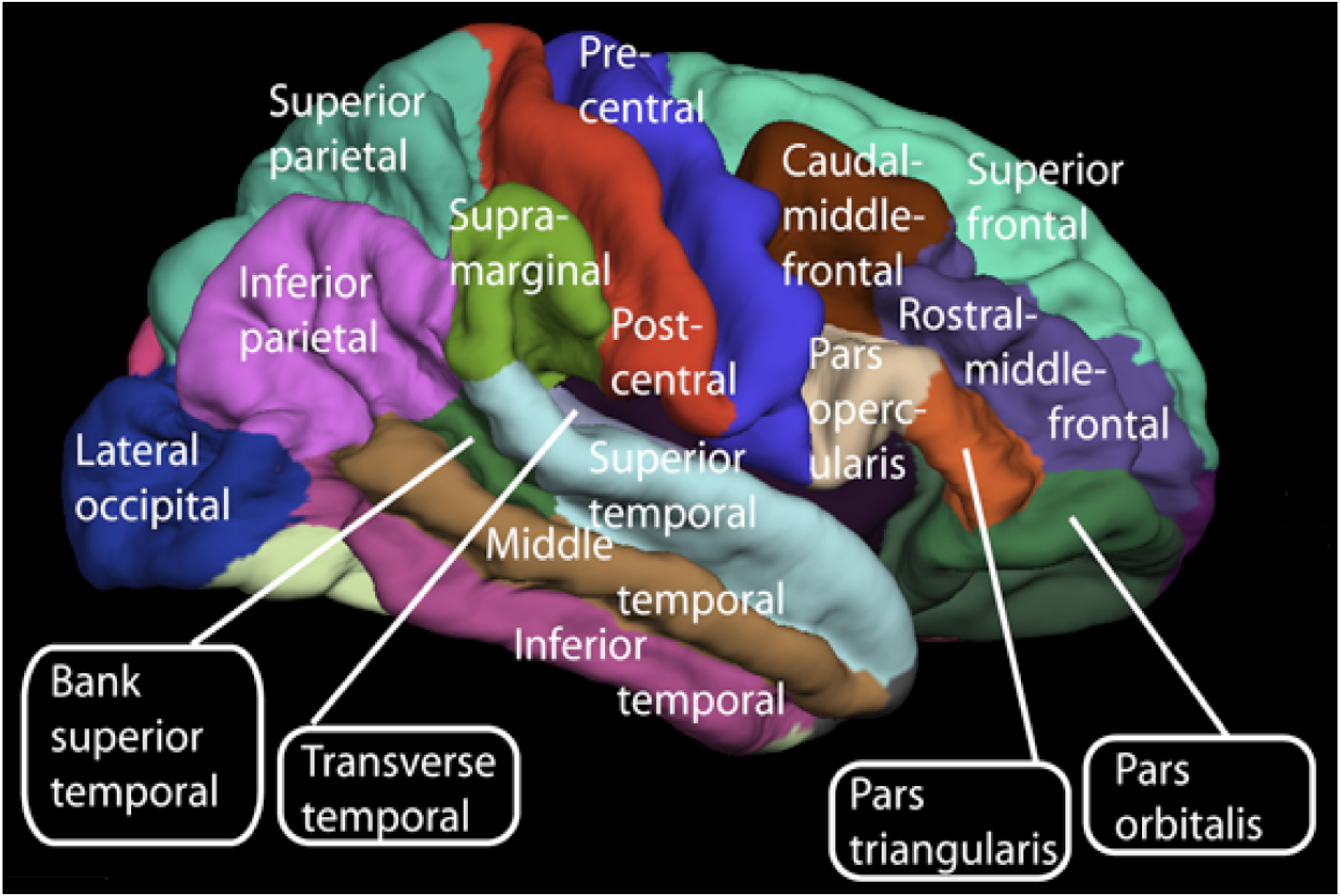


Brain Structure

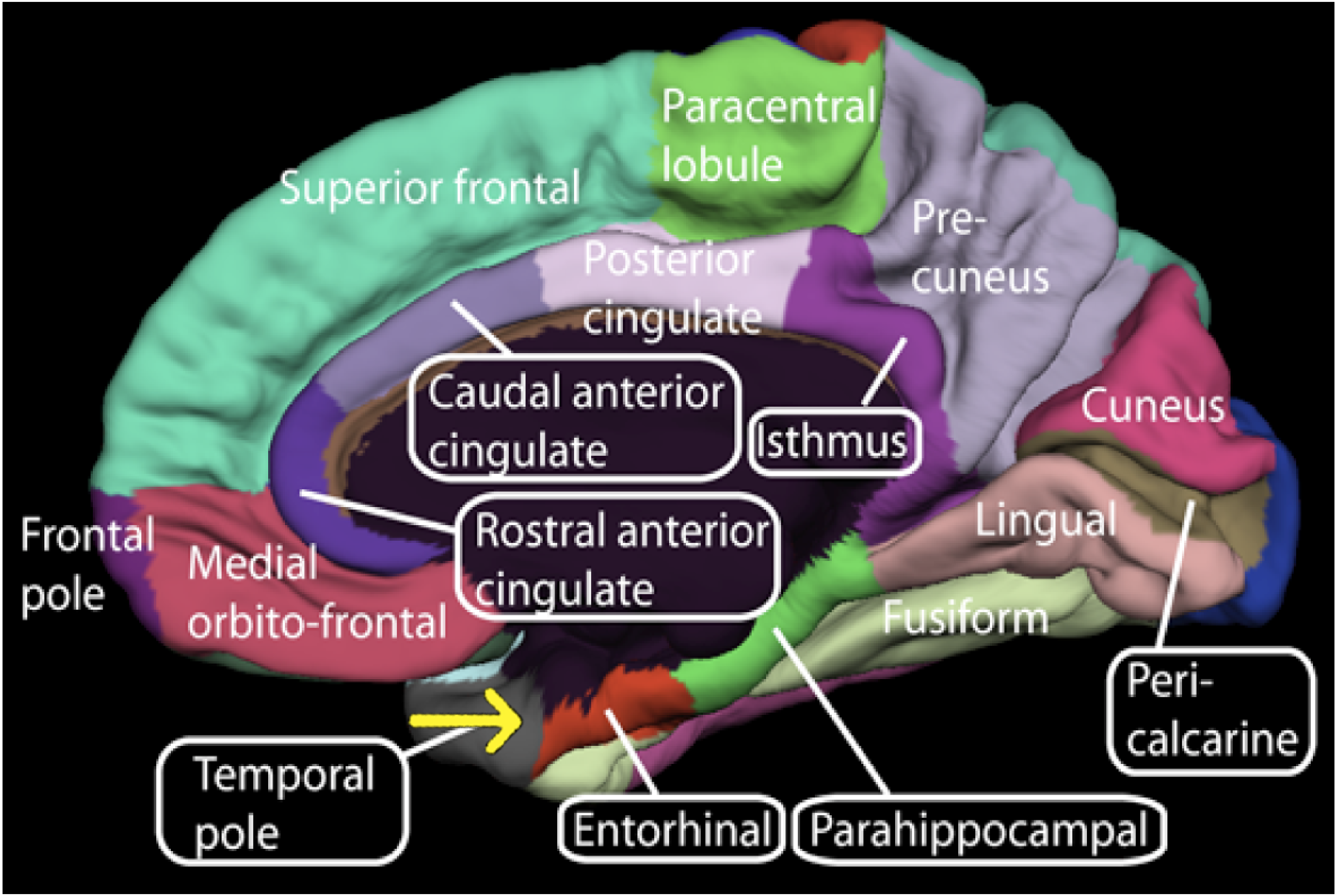




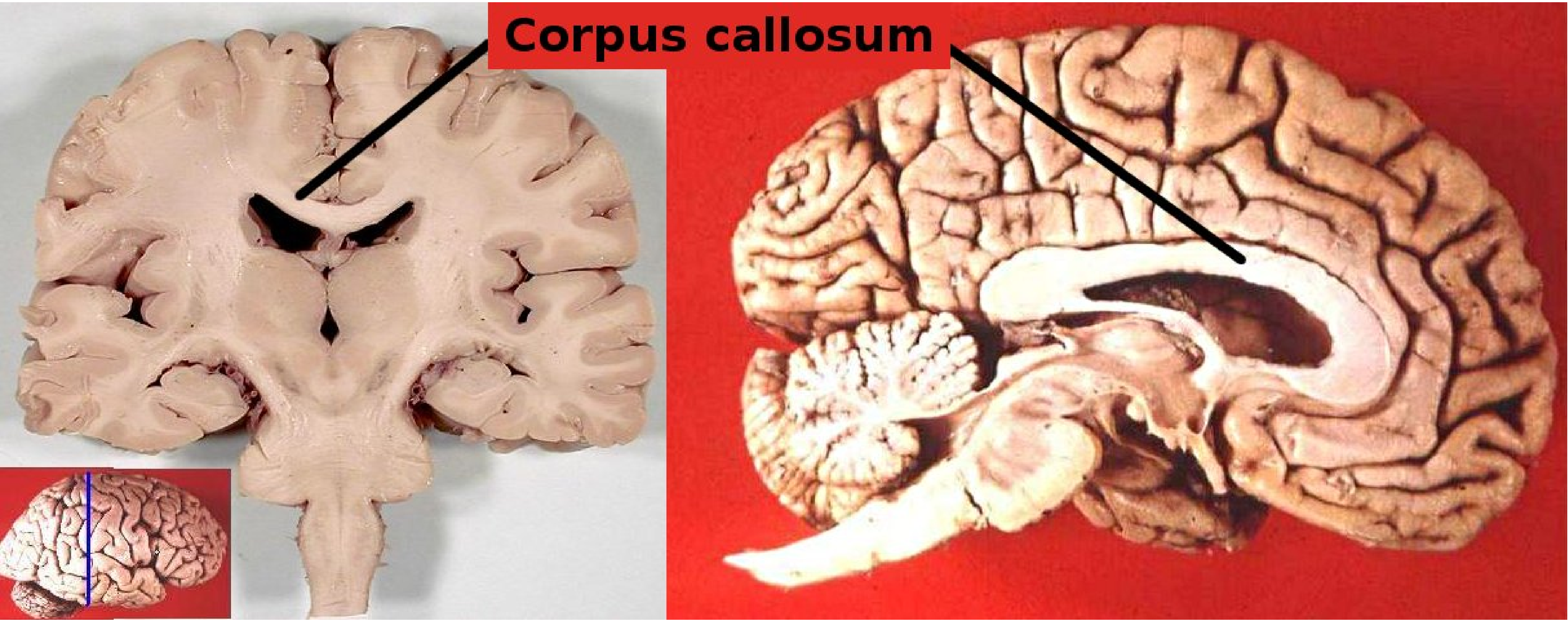
Brain Structure



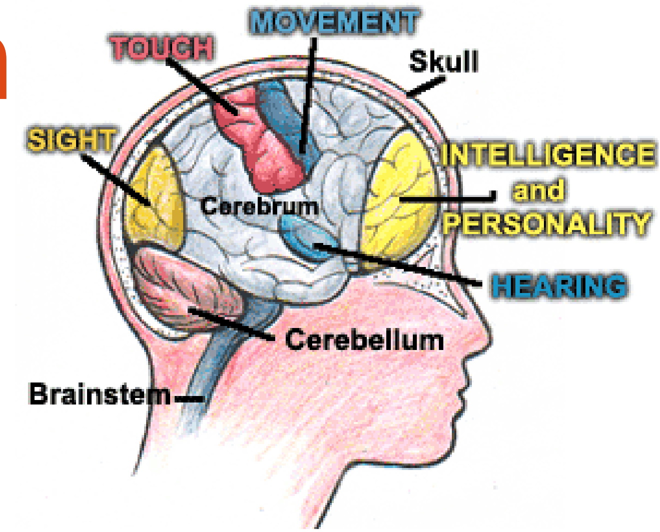
Brain Structure



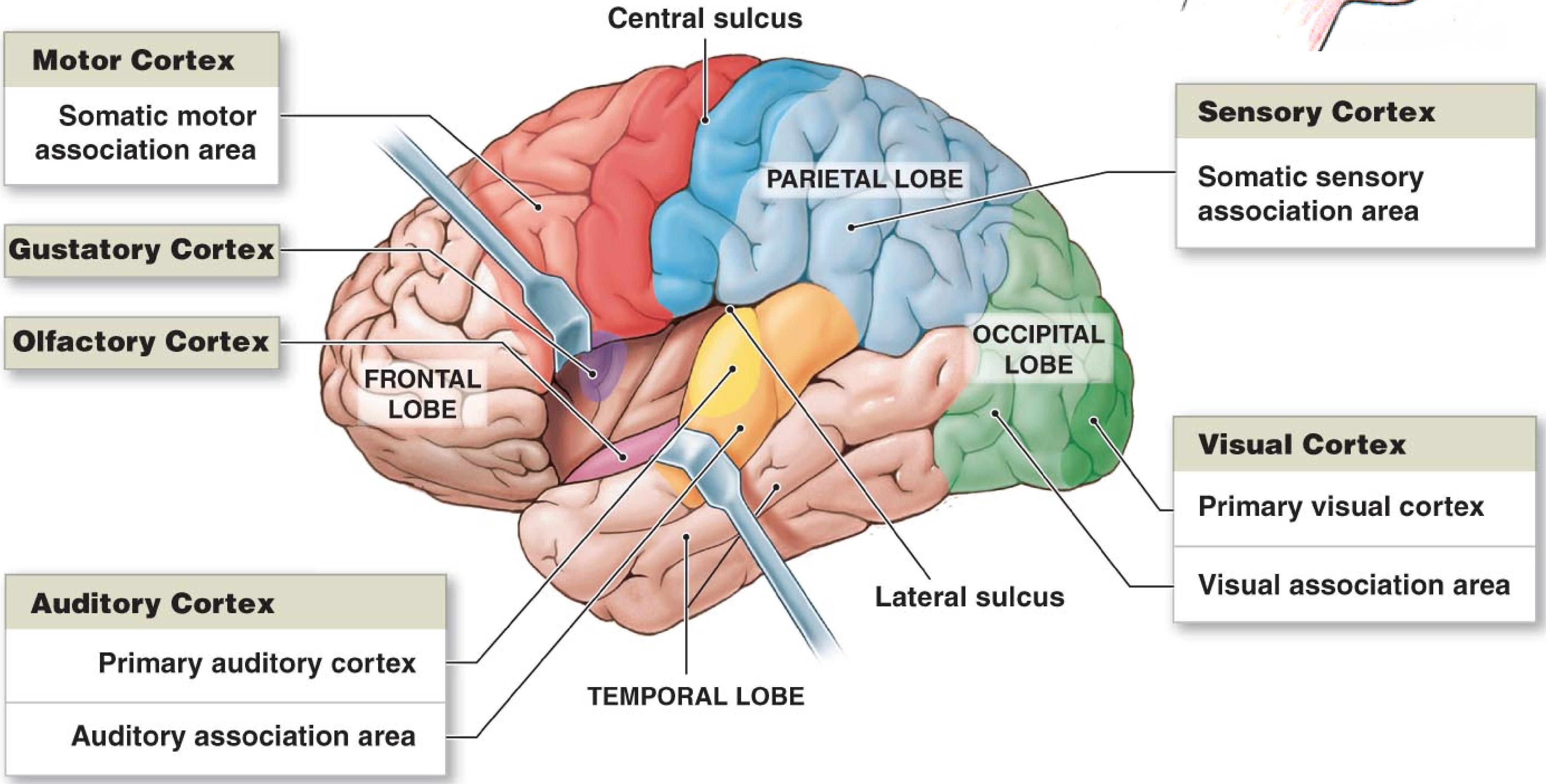
Brain Structure



Brain Function



The motor and sensory cortexes and the association areas for each



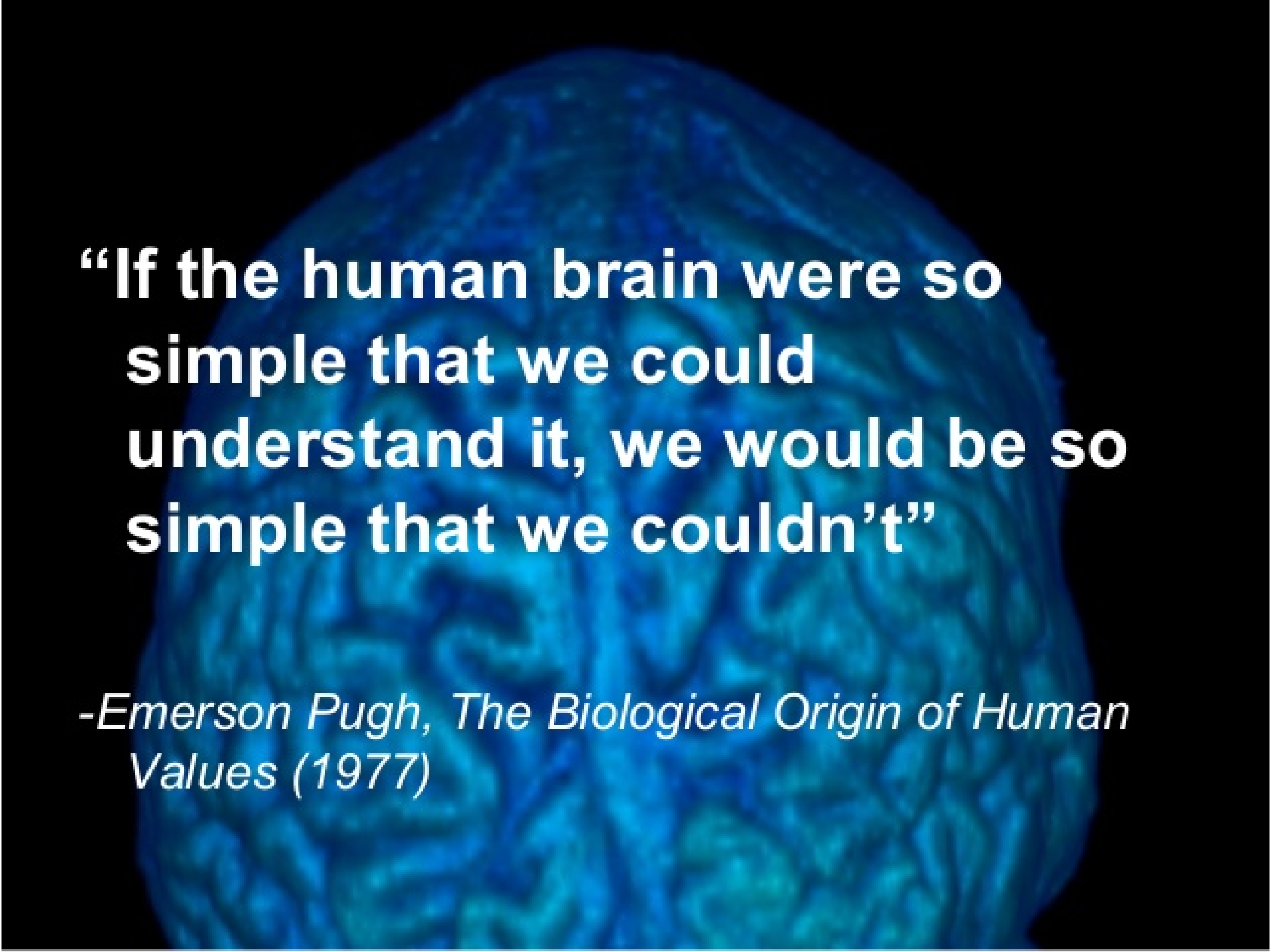

"If the human brain were so simple that we could understand it, we would be so simple that we couldn't"
-Emerson Pugh, The Biological Origin of Human Values (1977)

Image Registration

- Components

  (2) Similarity measure between image
  (when the task is of image alignment)

  - Are images of same modality ?
    - Squared error between corresponding pixel intensities
  - Are images of different modality ? e.g., MRI and CT ?
    - Cross-correlation of patch, around each pixel
    - Mutual information

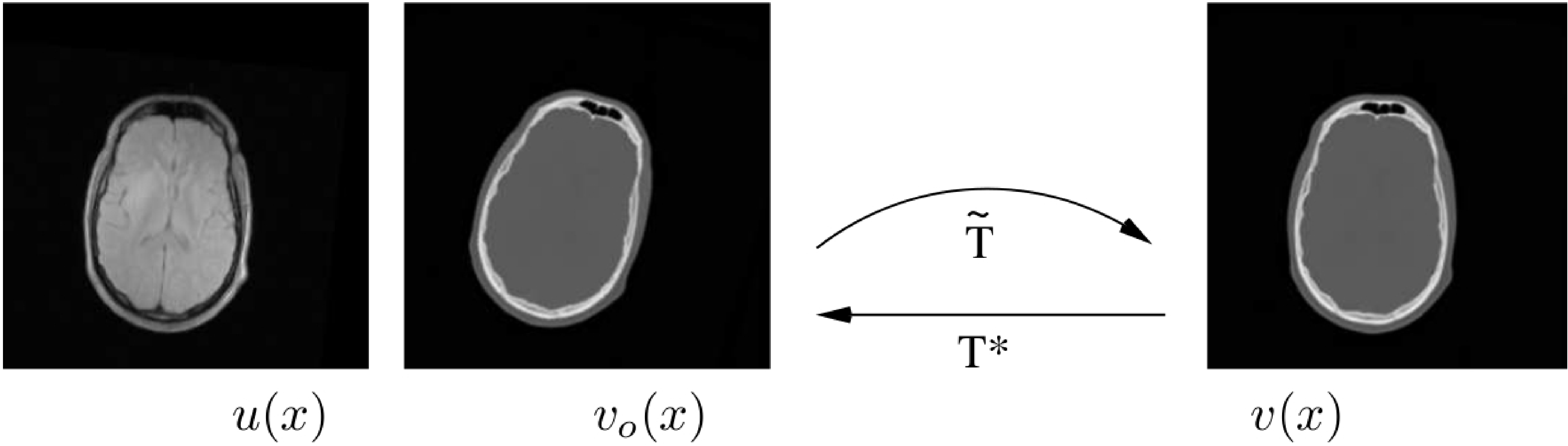


Image Registration

- Applications

  (1) Average appearance or geometry → Atlas

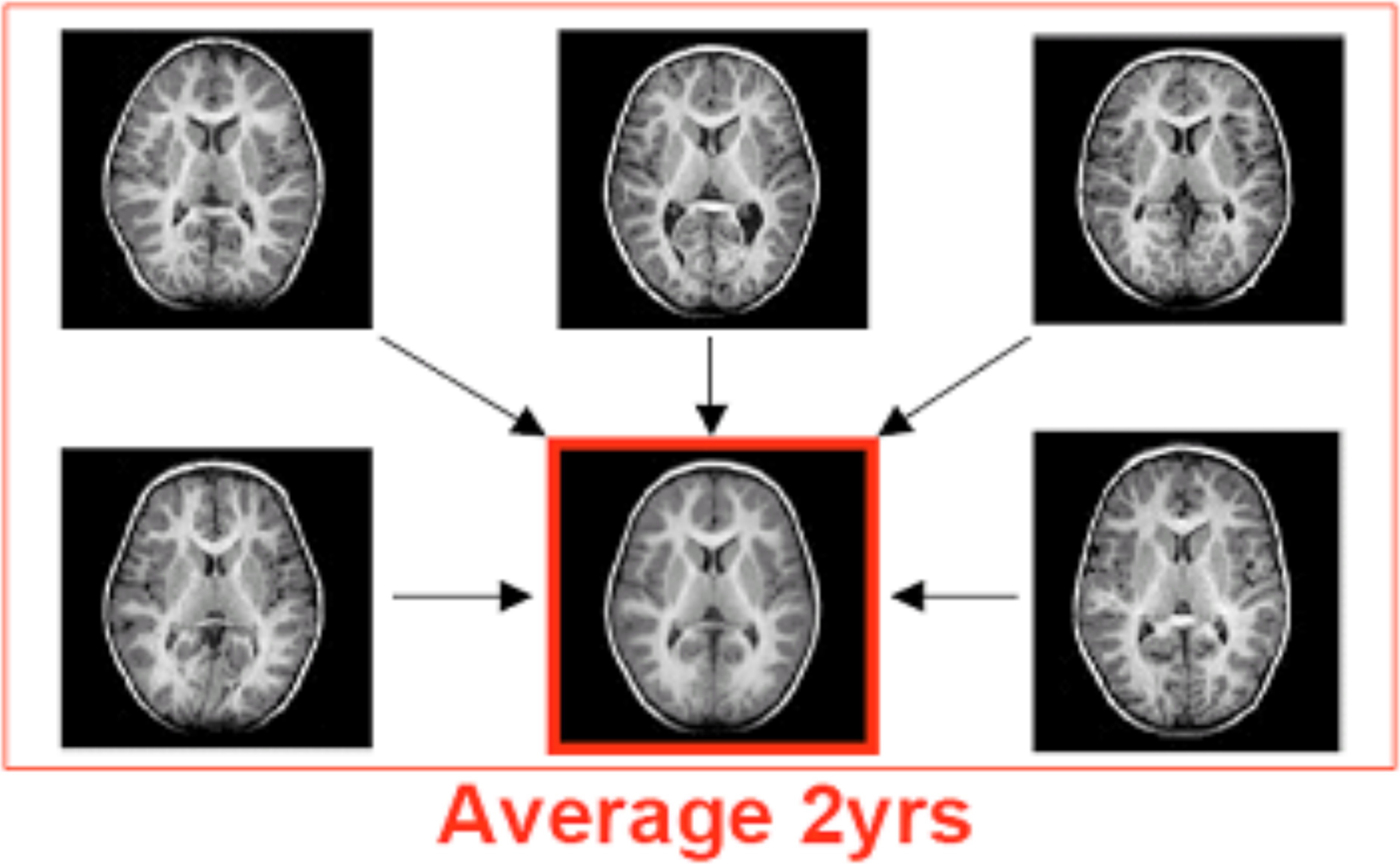
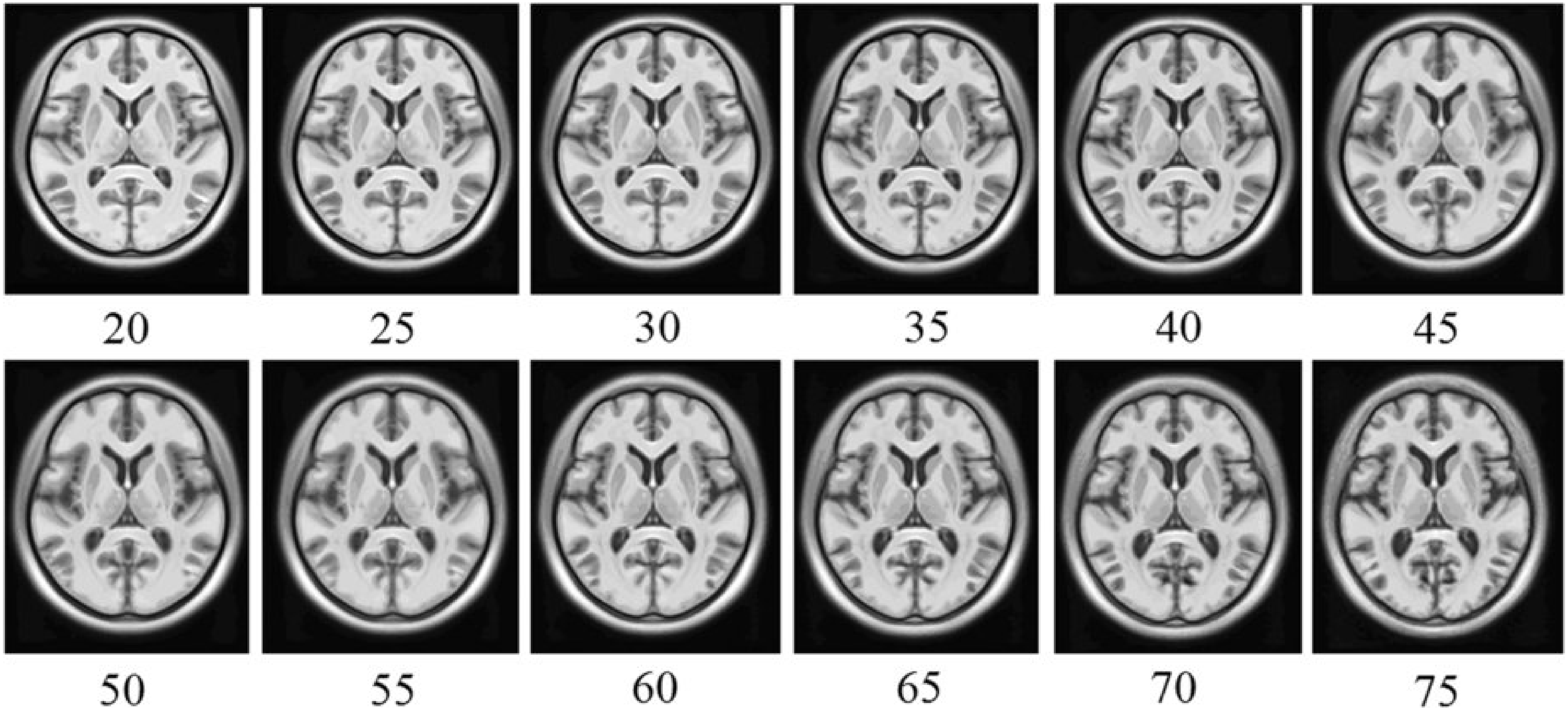


Image Registration

- Applications

  (1) Average appearance or geometry → Atlas



Nature 2015. Construction of brain atlases based on a multi-center MRI dataset of 2020 Chinese adults.

Image Registration

- Applications

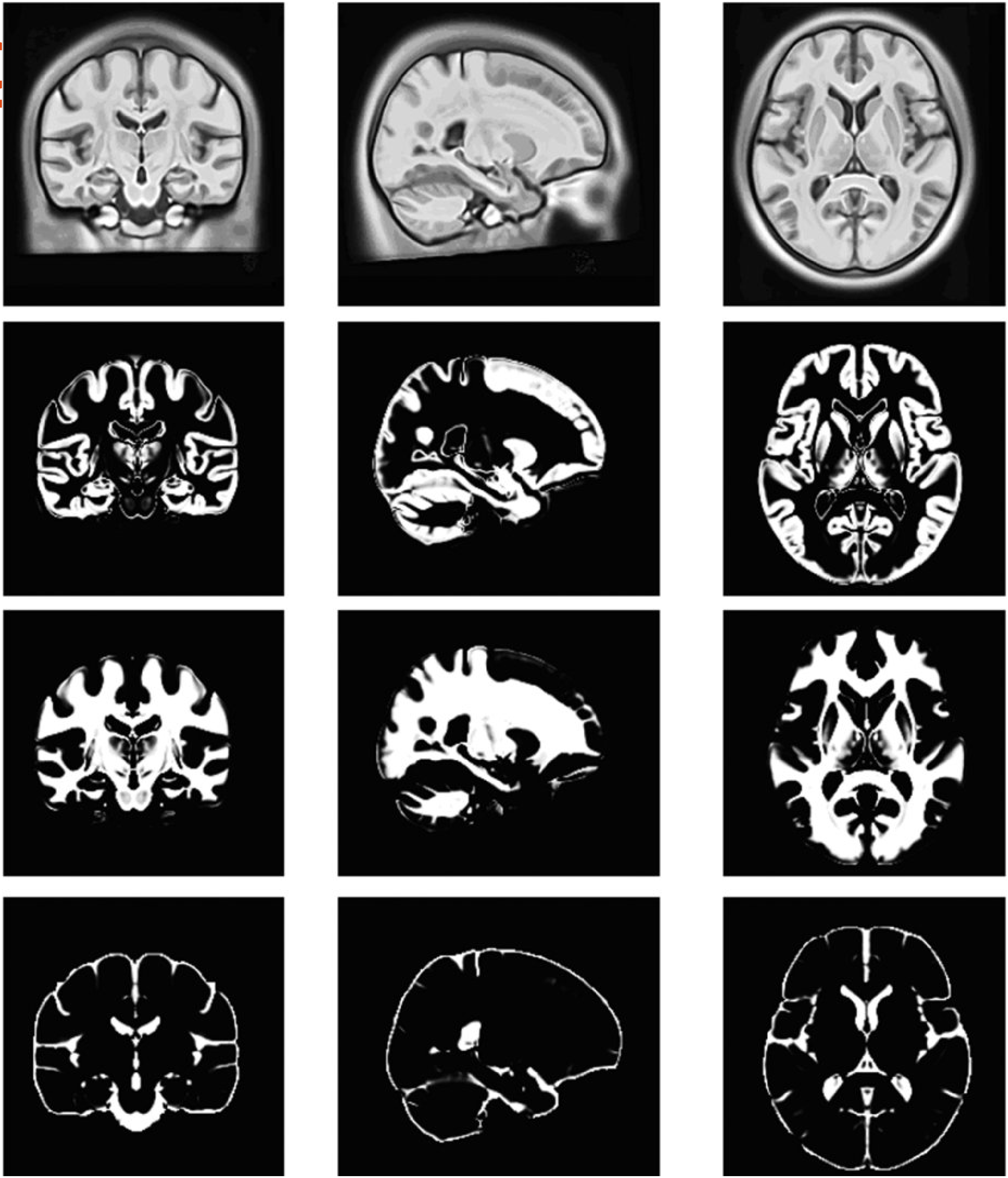
  (1) Average appearance or geometry → Atlas

  (2) Understanding variability in appearance / geometry

  (3) Atlas-based segmentation

Imag

- Applications
  - Atlas with tissue probability images



Nature 2015. Construction of brain atlases based on a multi-center MRI dataset of 2020 Chinese adults.

Image Reg

- Applications
  - Atlas with tissue probability images

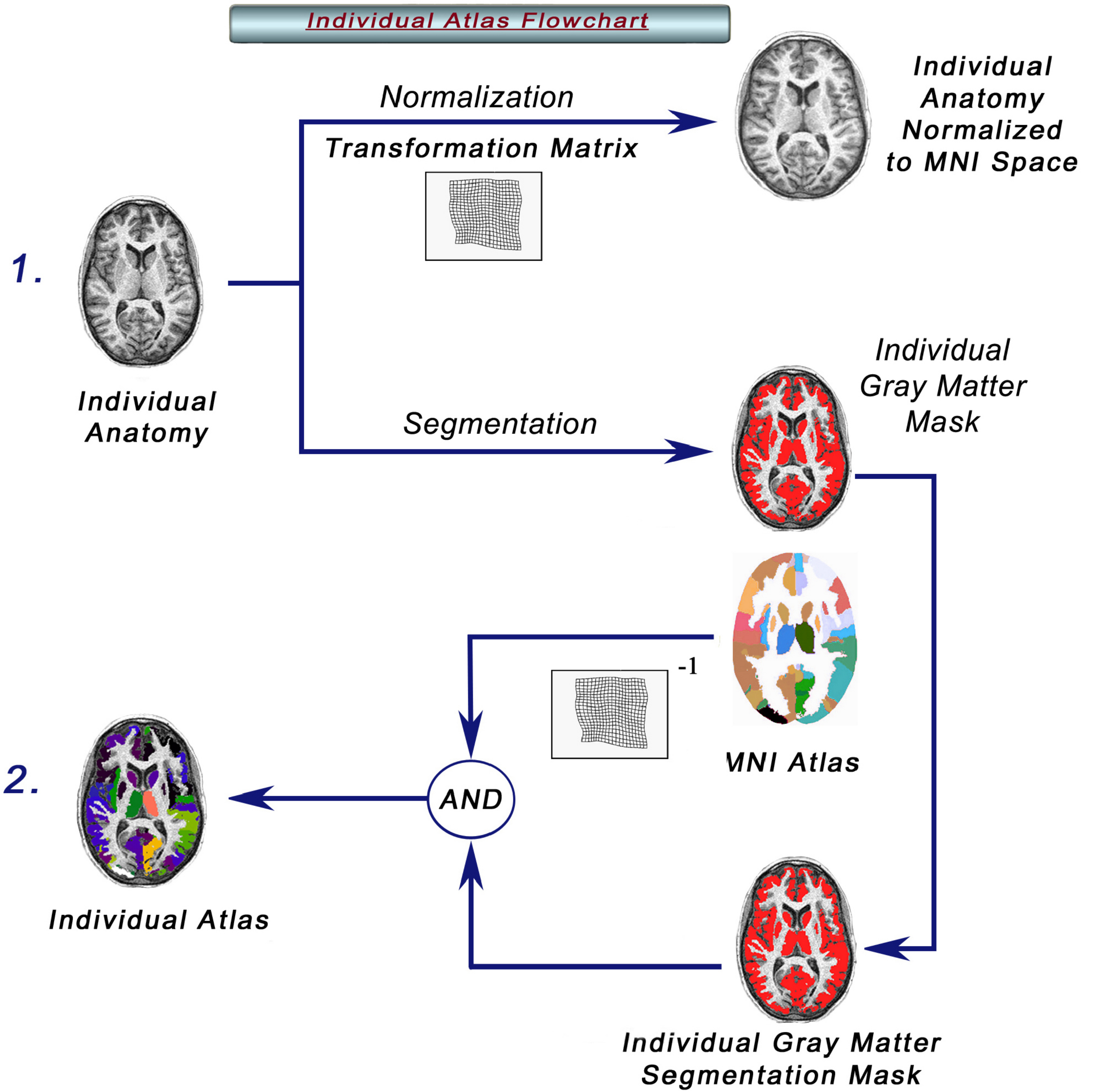


Image Registration

- Applications

  (1) Average appearance or geometry → Atlas

  (2) Understanding variability in appearance / geometry

  (3) Atlas-based segmentation

  (4) Tracking

  - Organ / structure location and size over time
  - Tumor location and size over time

Image Registration

- Group registration
  - Method similar to that used for statistical shape analysis
    - Find mean, align to mean, repeat

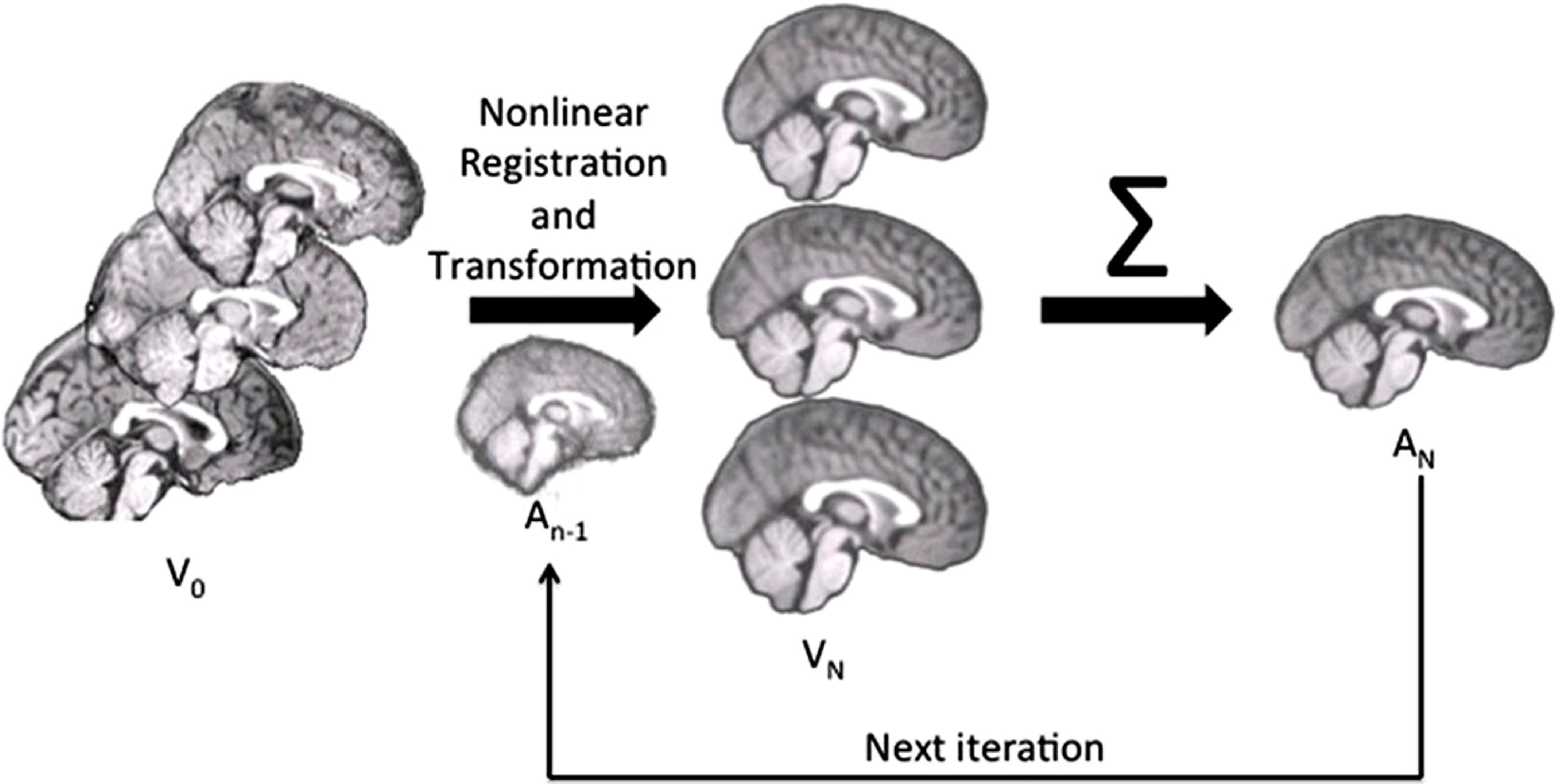


Image Registration

- Let $x \in R^3$ be a (pixel) location in domain Ω
- Given
  - $R(x)$ = reference image
  - $T(x)$ = target image
- Goal
  - Align image T to image R

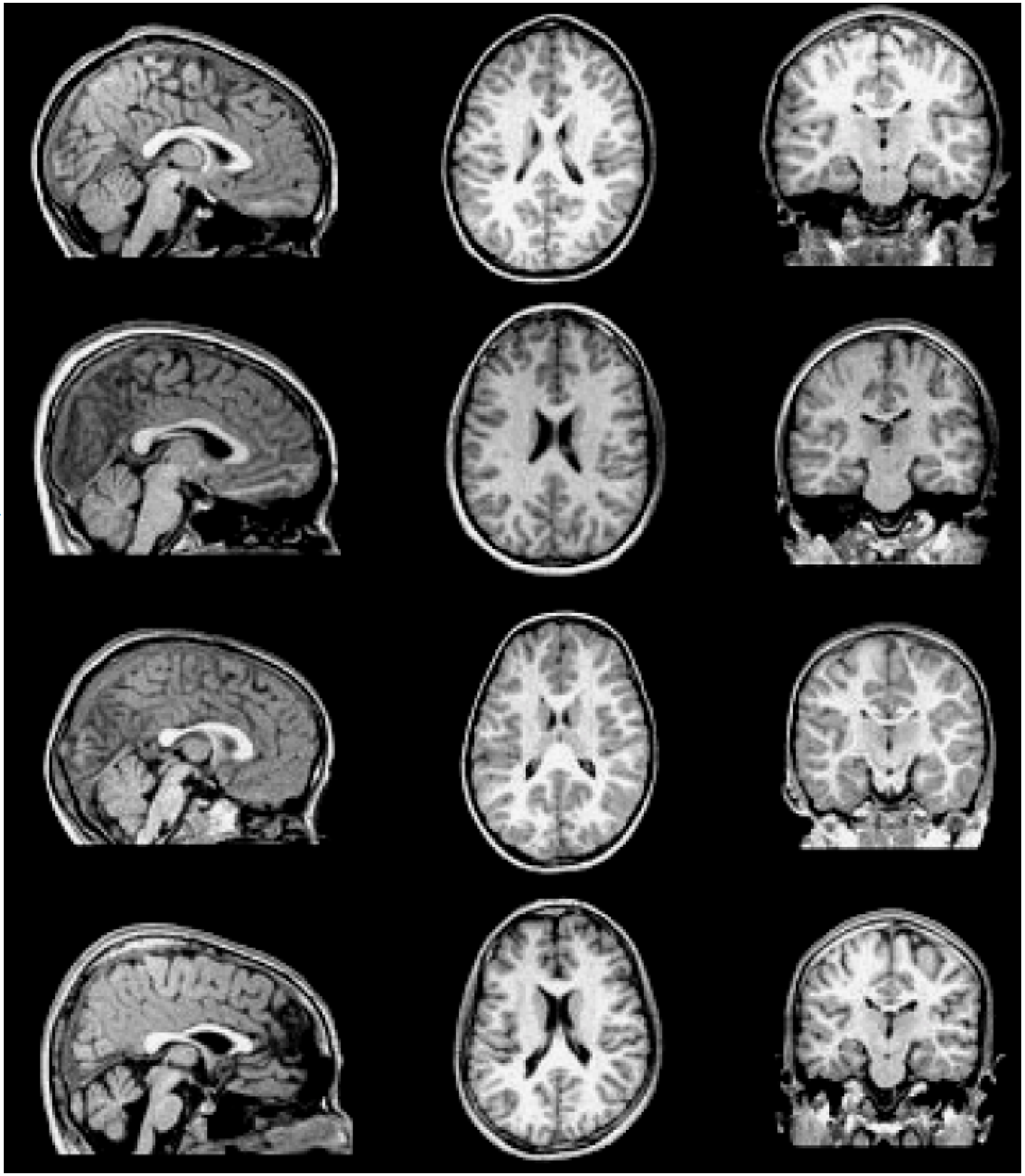


Image Registration

- Formulation
  - Let Φ(x) be a spatial transformation function
  - Special case: Φ(x) := Ax + b
    - Linear transformation, $b \in R^3$, $A \in R^{3x3}$ (invertible)
  - Assume dissimilarity → squared-difference
  - R(.) = discretized image, N pixels with locations $x_i \in R^3$
  - Objective function:

$$J(A, b) := \sum_{i=1}^{N} \Big(T(\phi(x_i; A, b)) - R(x_i) \Big)^2$$

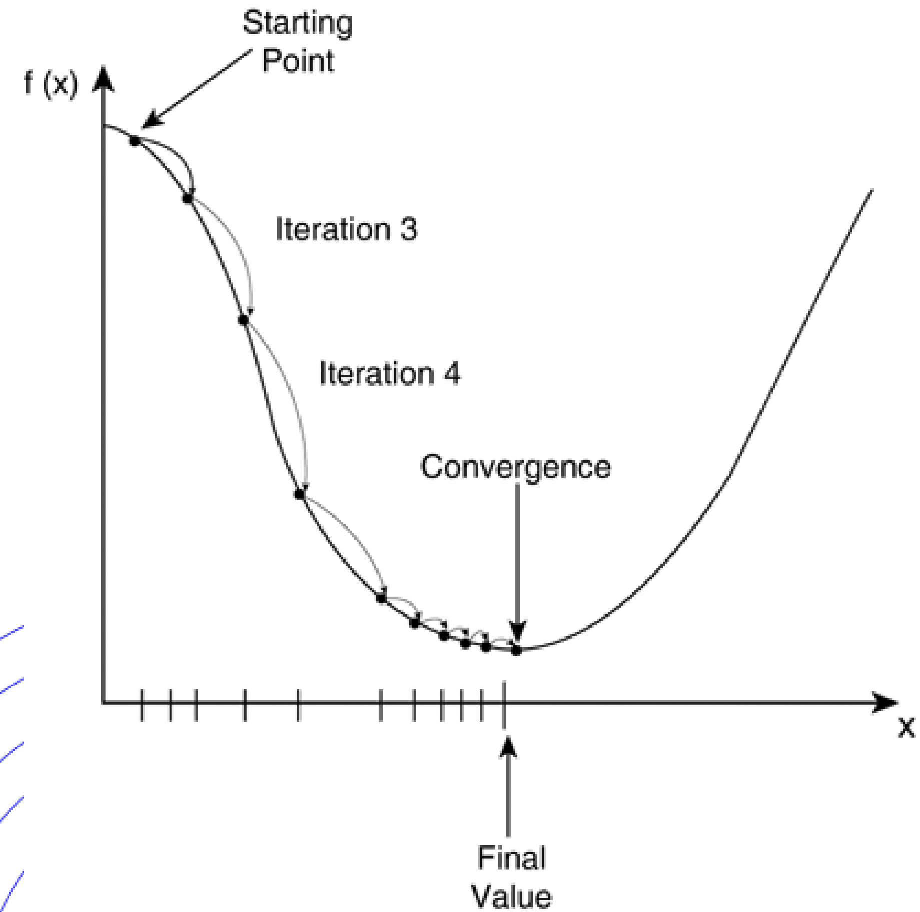
    - Optimize parameters A,b to minimize J(A, b)
    - How ?

Image Registration

- Gradient-descent algorithm
  - Iterative optimization
  - Function to be optimized = “Objective Function”
  - Step 0: Start with an estimate of the solution
  - Step 1: Improve estimate a little bit
    - Update based on gradient of objective function
  - Step 2: Repeat previous step until no improvement

Image Registration

- Gradient-descent algorithm
  - Blue Curves = Contours of objective function f(A,B)
  - Let current solution = (a,b)
  - Improved solution
    - Compute gradient at (a,b)
      - g(a,b) = (df/dA, df/dB) evaluated at (A=a,B=b)
    - Improved solution (a',b') = (a,b) – s g(a,b)
    - s > 0 is a small step size
    - How to choose s ?



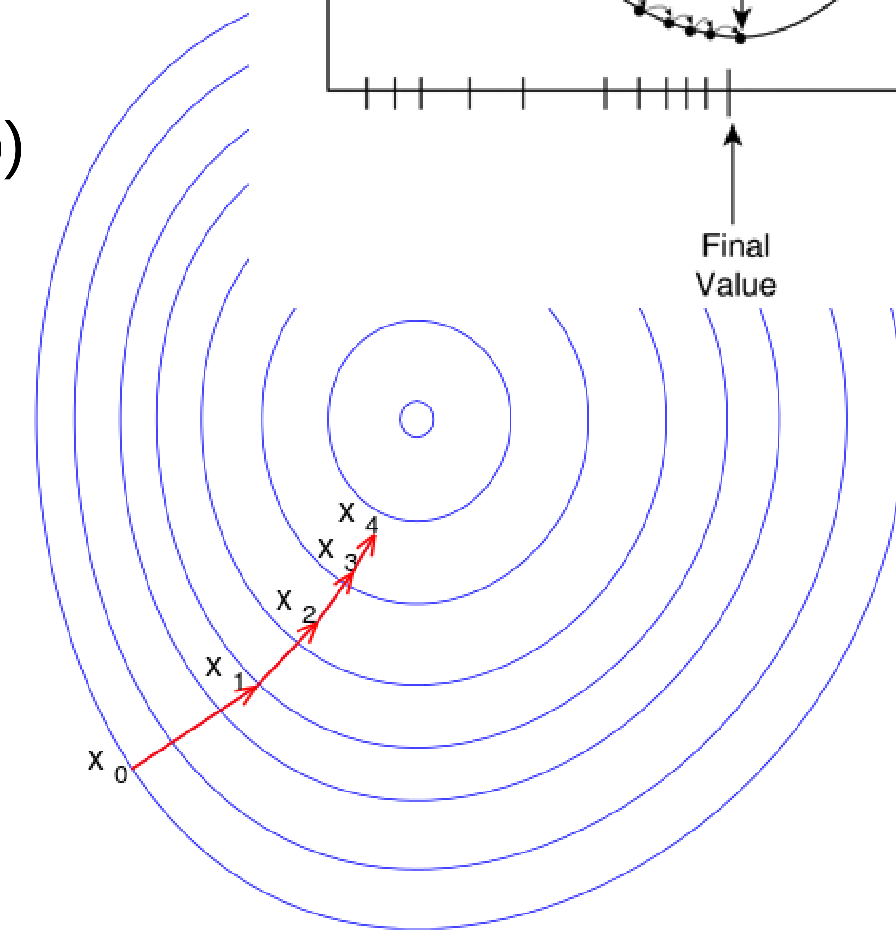


Image Registration

- Gradient-descent algorithm (how to choose step size)
  - Start with some 's'
  - Guess updated solution
    (a',b') = (a,b) – s g(a,b)
  - Check if f(.) value reduces
    - If f(a',b') < f(a,b),
      improved solution found
      increase s by 10%
    - If f(a',b') >= f(a,b),
      reduce s by 50%
  - Repeat Guess+Check until
    improved solution found OR
    s becomes zero

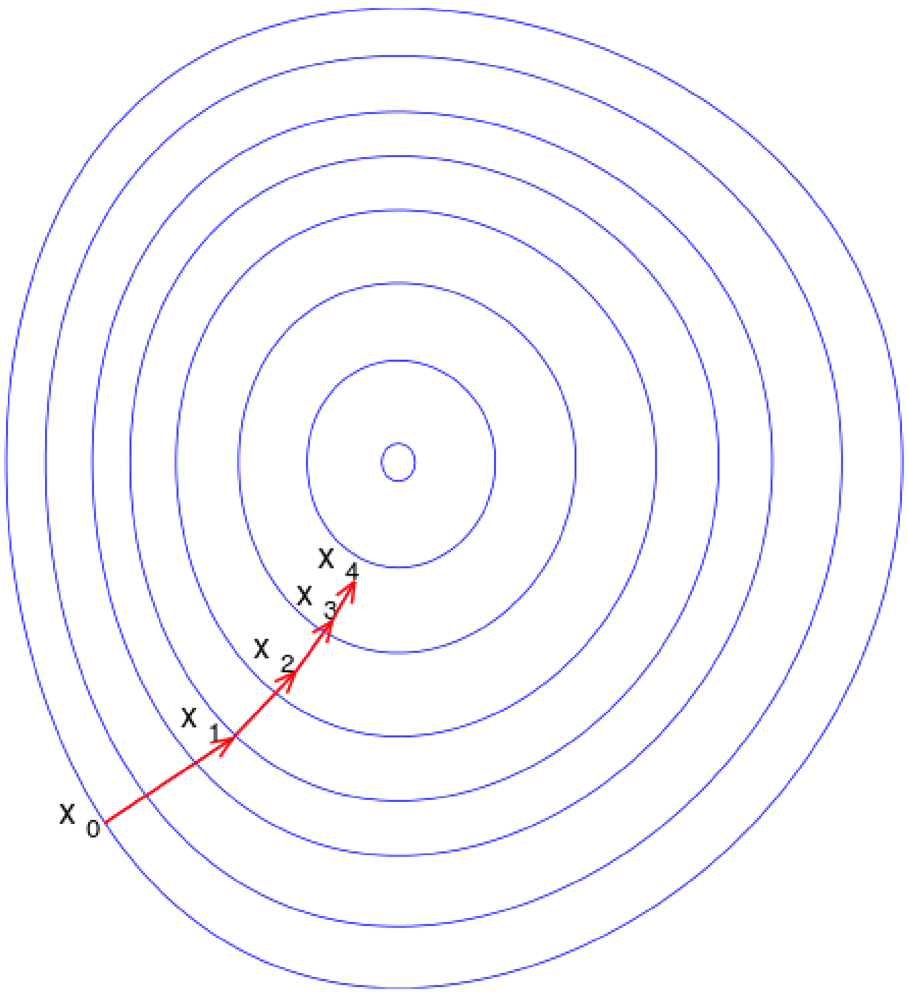


Image Registration

- Gradient-descent algorithm
  - Let current solution = (a,b)
  - Improved solution
    - Compute gradient at (a,b)
      - g(a,b) = (df/dA, df/dB) evaluated at (A=a,B=b)
    - Improved solution (a',b') = (a,b) – s g(a,b)
    - s = chosen step size
  - Repeat last step until improvement is negligible

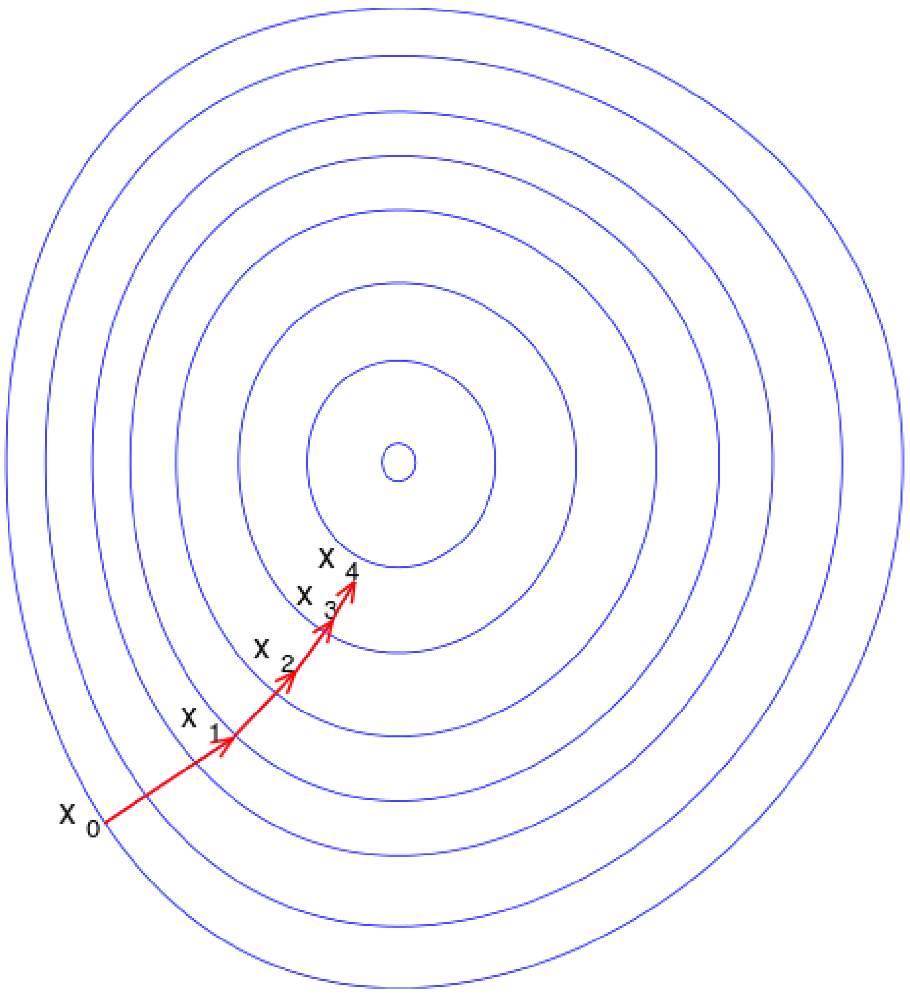


Image Registration

- Gradient descent guaranteed to convergence
  - Given sufficiently small step size
  - If maximum curvature of objective function is known, then step size can be fixed appropriately

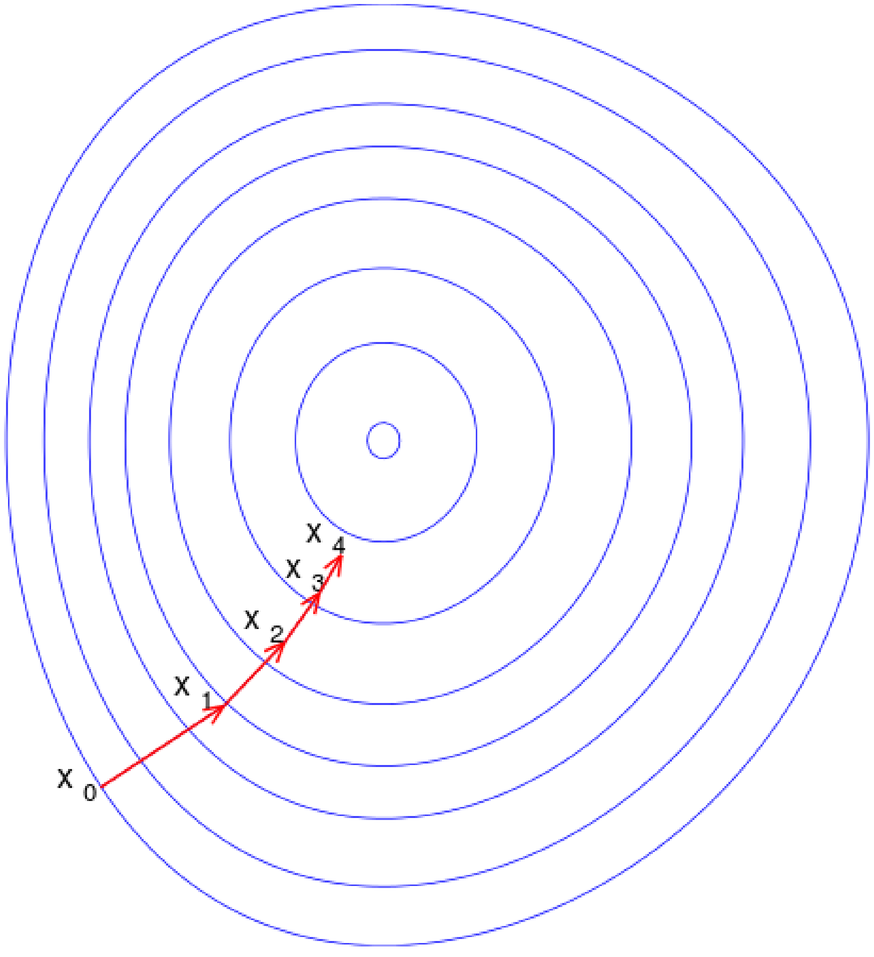
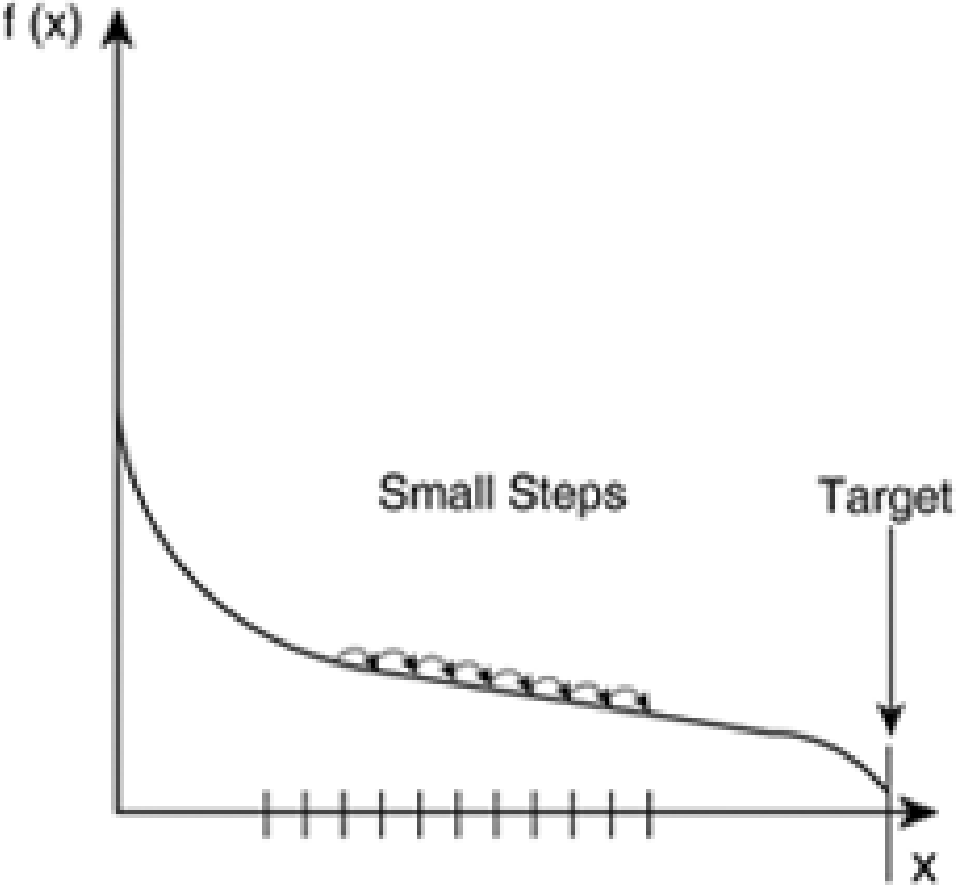


Image Registration

- Gradient descent



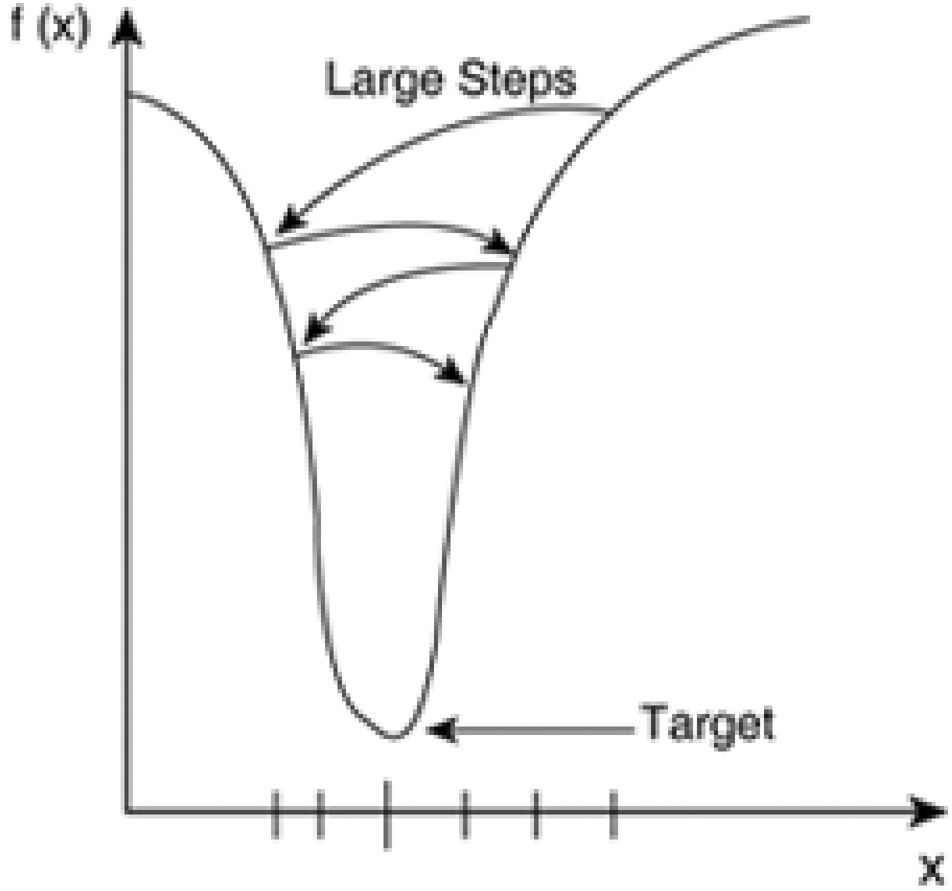
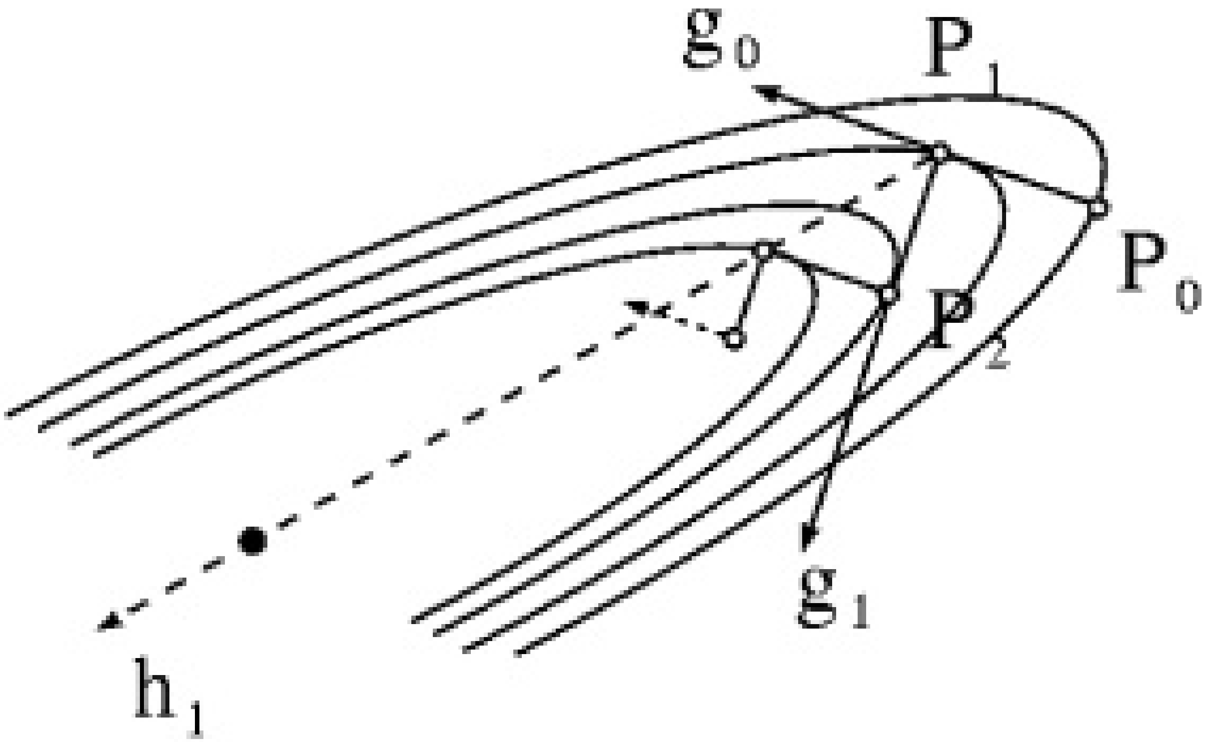


Image Registration

- Gradient



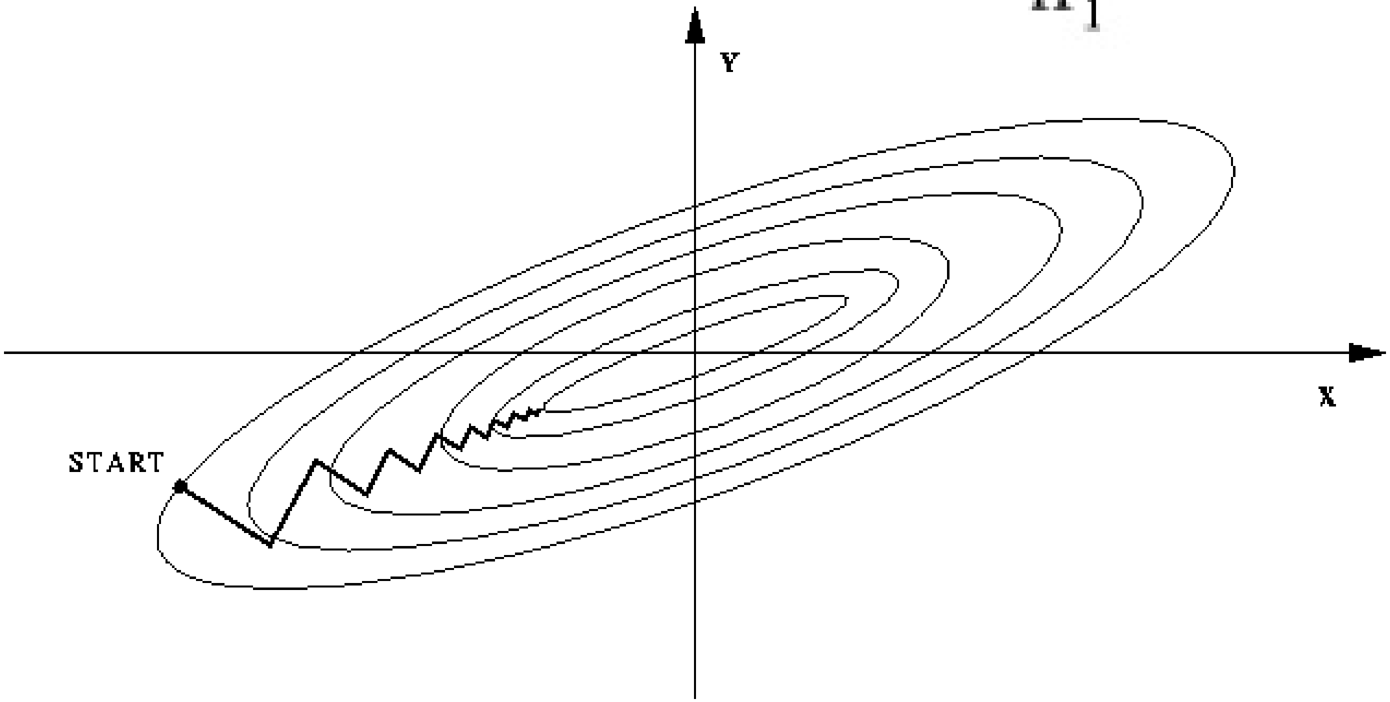
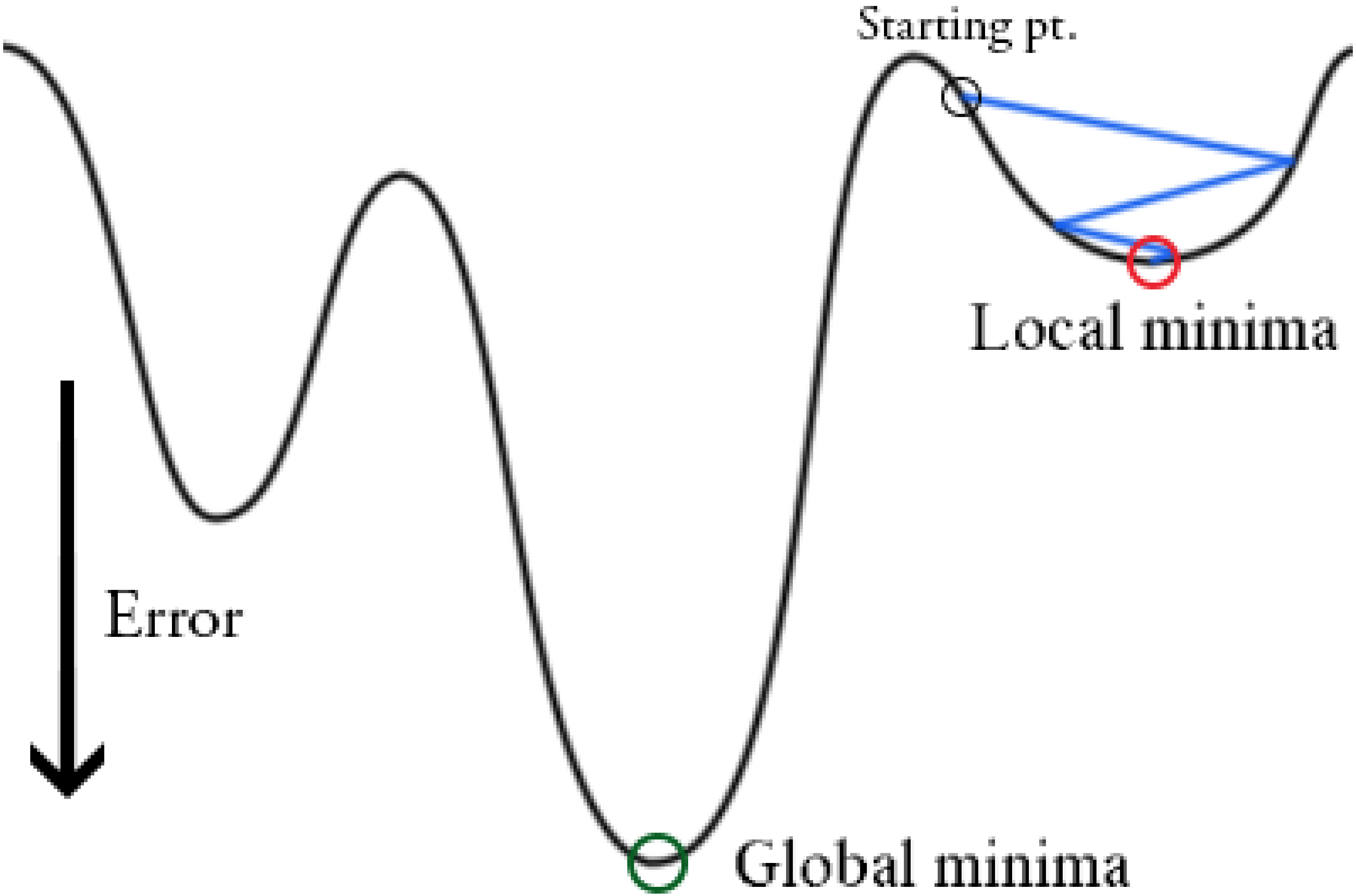


Image Registration

- Gradient descent



Matrix Derivatives

- Consider a function f : $R^n \rightarrow R^m$
  - f(.) = m-valued function of n variables x1, · · ·, xn
    - x = [x1, ..., xn]'
  - Function value components :
    f1(x1,· · ·,xn), · · · , fm(x1,· · ·,xn)
  - Represent f (x) as a column vector :
    [f1(x1,· · ·,xn), · · · , fm(x1,· · ·,xn)]'
- Derivative ∂f (x)/∂x = Jacobian matrix = D
  - $D_{ij} := \dfrac{\partial f_i(\cdot)}{\partial x_j}$
  - Number of rows = number of values m
  - Number of columns = number of variables n

Matrix Derivatives

- Multivariate Gradient-Descent Optimization
  - Let x = [· · · , xi , · · ·]' = column vector of variables
  - Let f (x) = real-valued function of x
  - Derivative ∂f(x)/∂x = is a row vector g'(x)

$$g'(x) = \left[\cdots, \frac{\partial f(x)}{\partial x_i}, \cdots\right]$$

- For the special case of real-valued functions
  - Transpose of derivative (row vector g (x))
    is gradient (column vector g(x))

Image Registration

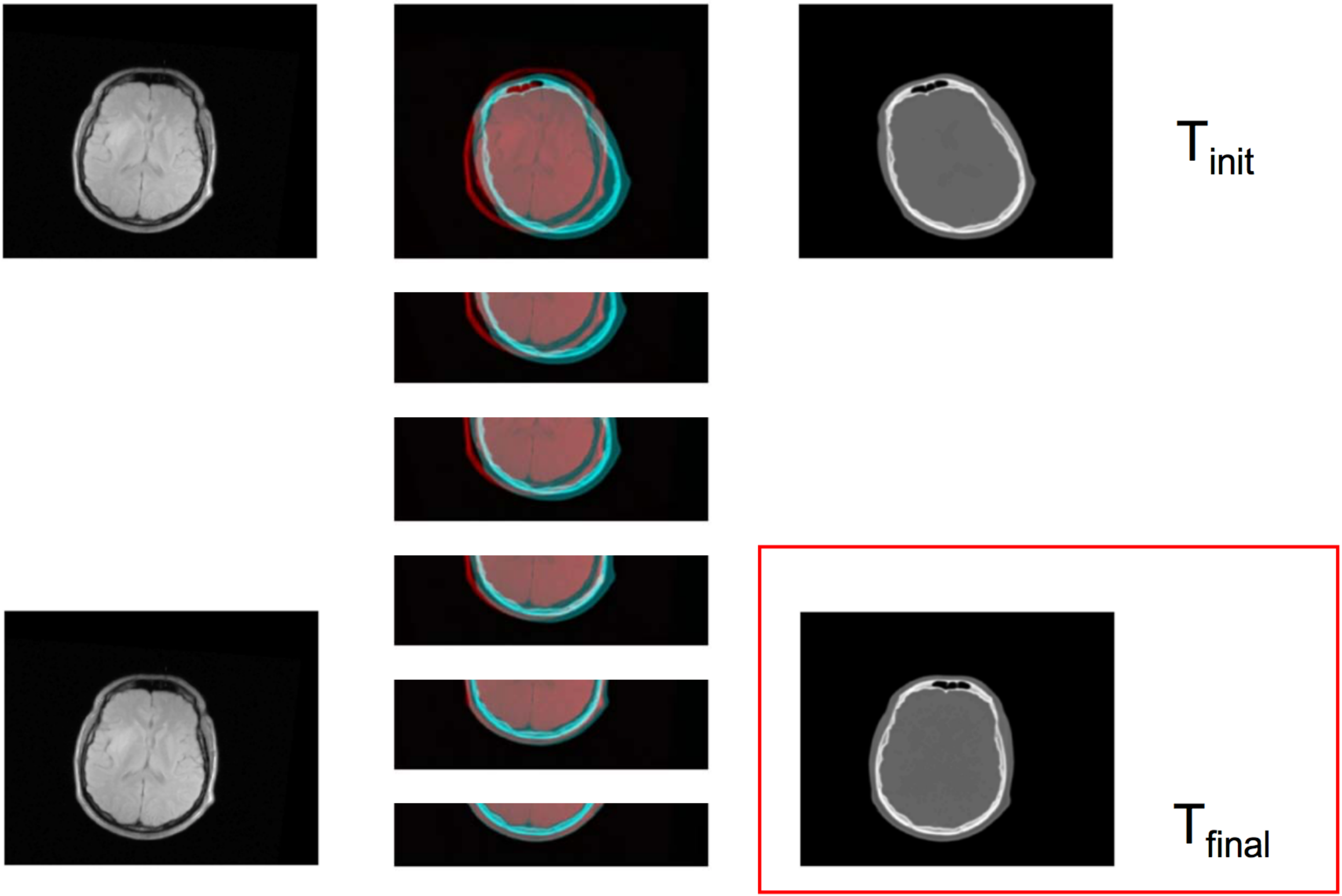


Image Registration

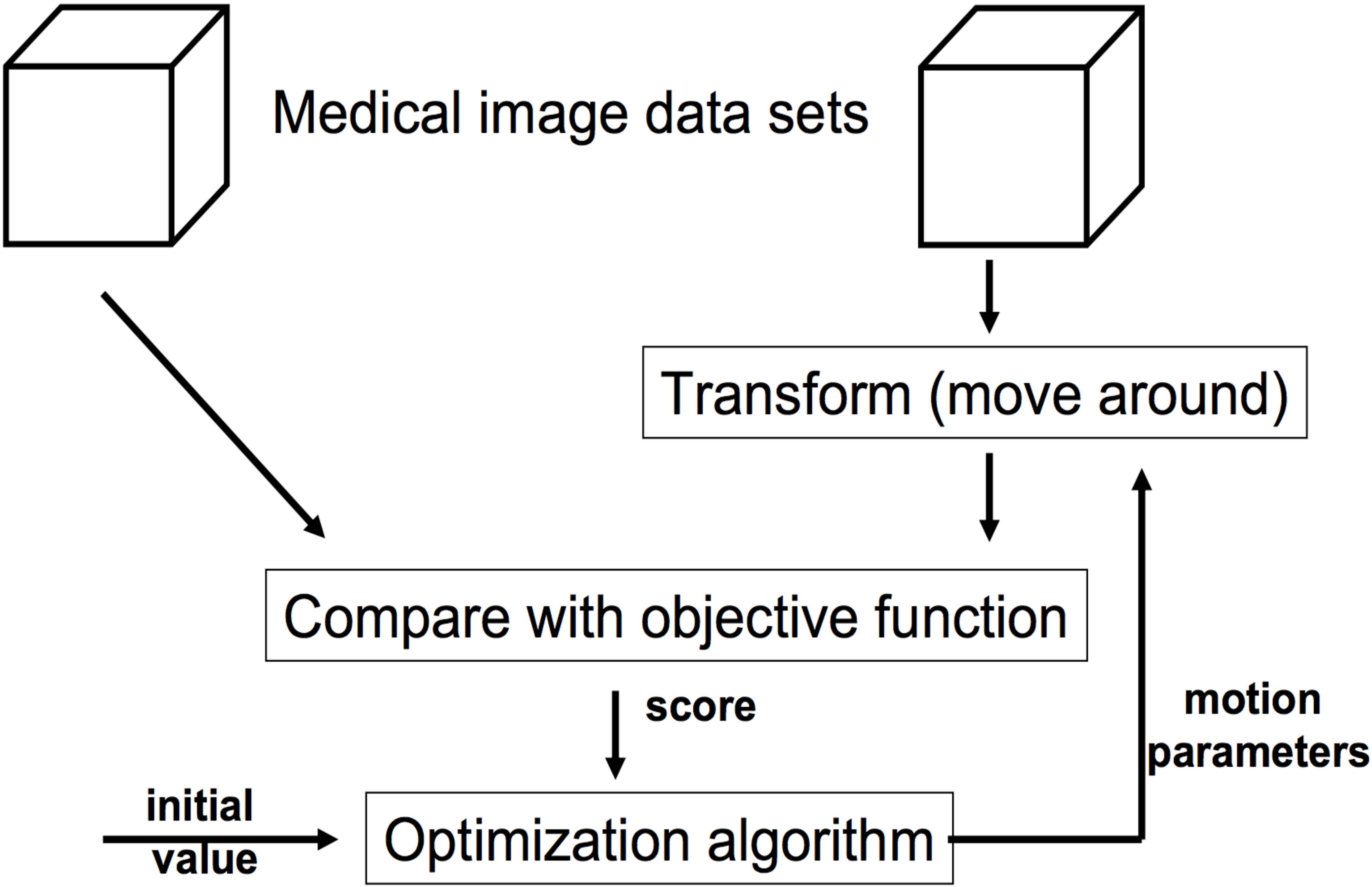


Image Registration

$$J(A,b) := \sum_{i=1}^{N} \Big(T(\phi(x_i; A, b)) - R(x_i) \Big)^2$$

- **Gradient descent w.r.t. translation 'b'**
  - Using chain rule, partial derivative of objective function:

$$\frac{\partial J(A,b)}{\partial b} := \sum_{i=1}^{N} 2\Big(T(\phi(x_i; A, b)) - R(x_i) \Big) \frac{\partial T(\phi(x_i; A, b))}{\partial \phi(x_i; A, b)} \frac{\partial \phi(x_i; A, b)}{\partial b}$$

$$= \sum_{i=1}^{N} 2\Big(T(\phi(x_i; A, b)) - R(x_i) \Big) \frac{\partial T(\phi(x_i; A, b))}{\partial \phi(x_i; A, b)}$$

because $\frac{\partial}{\partial b}\phi(x) = \frac{\partial}{\partial b}(Ax + b) = \mathbf{I}_{3\times 3}$

  - Spatial derivative of T(.) = row vector
    - Jacobian of real-valued function T(.)
  - Gradient = transpose of derivative

Image Registration

- **Gradient descent w.r.t. translation 'b'**
  - Example

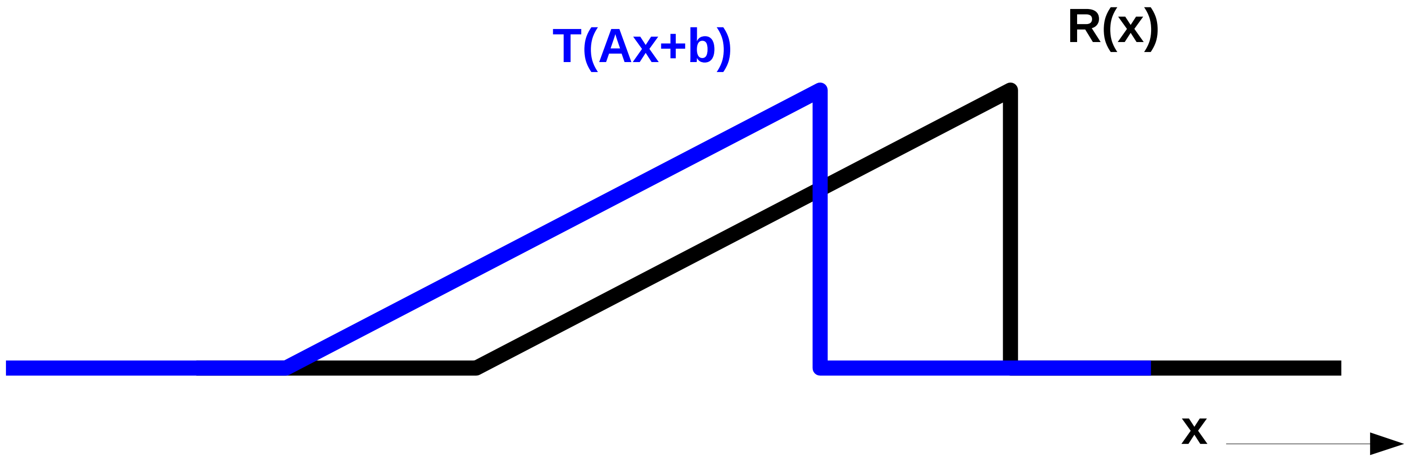


Image Registration

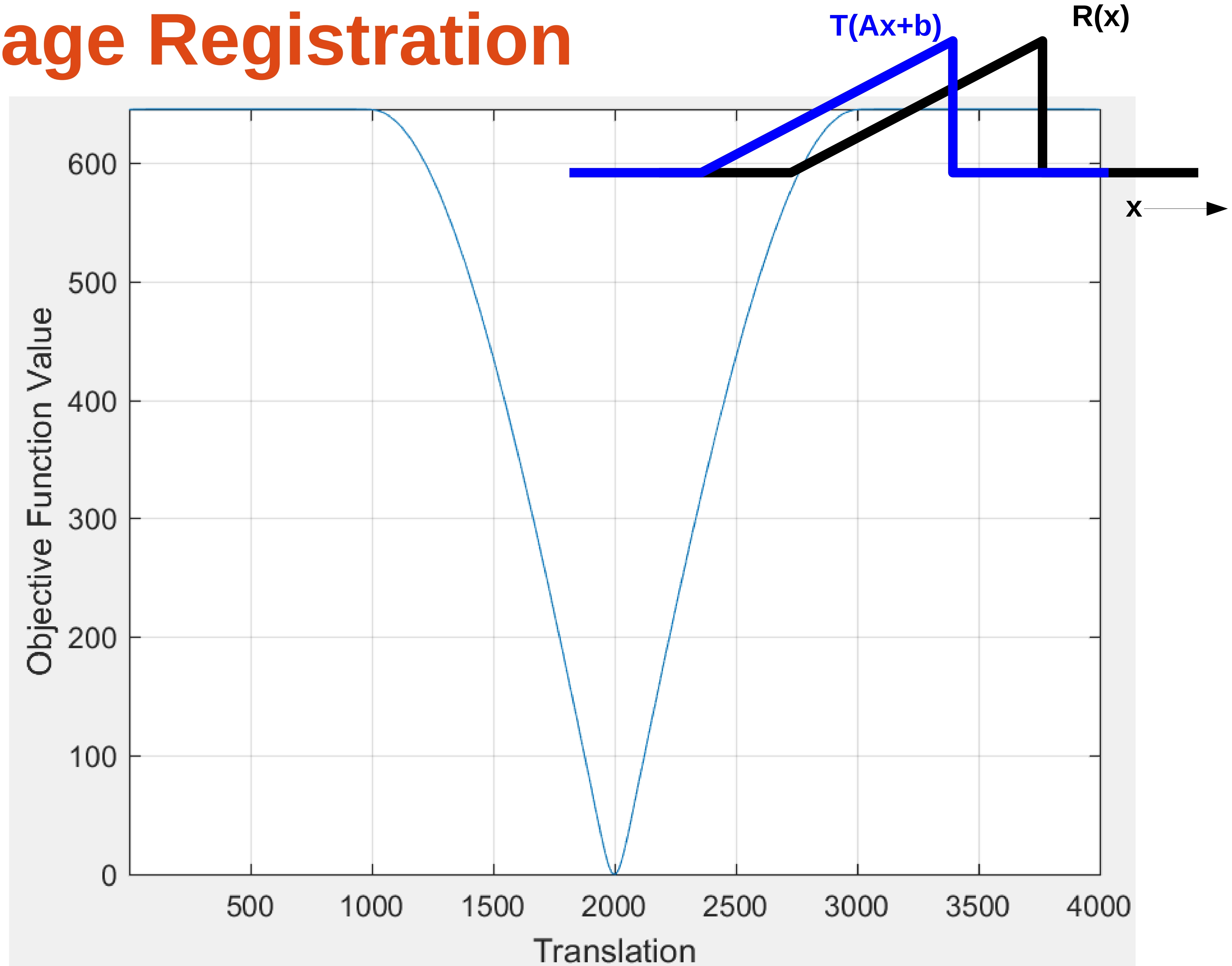
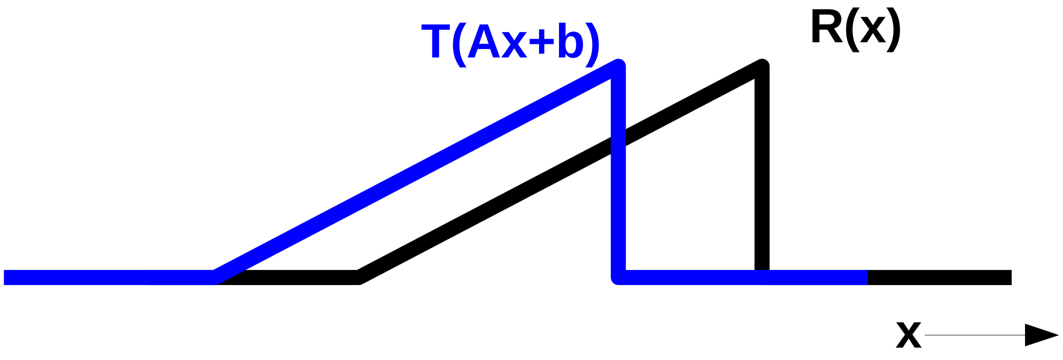


Image Registration

- Gradient descent w.r.t. translation 'b'
  - Example
    - Φ(x) := Ax + b



$$b \leftarrow b - \tau \frac{\partial J(A,b)}{\partial b}^{\top}$$

$$\sum_{i=1}^{N} 2\Big(T(\phi(x_i;A,b)) - R(x_i)\Big) \frac{\partial T(\phi(x_i;A,b))}{\partial \phi(x_i;A,b)}$$

  - 2^{nd} term = spatial derivative of T → ‘always’ pointing right
  - 1^{st} term = positive (at ‘all’ 'x' where 2^{nd} term > 0)
  - Gradient descent reduces 'b', i.e., pushes 'b' left
    - Shifting coordinate axis to left moves image T to the right

Image Registration

- Aligning ramps (w.r.t. translation)
  - Area under magenta curve > 0

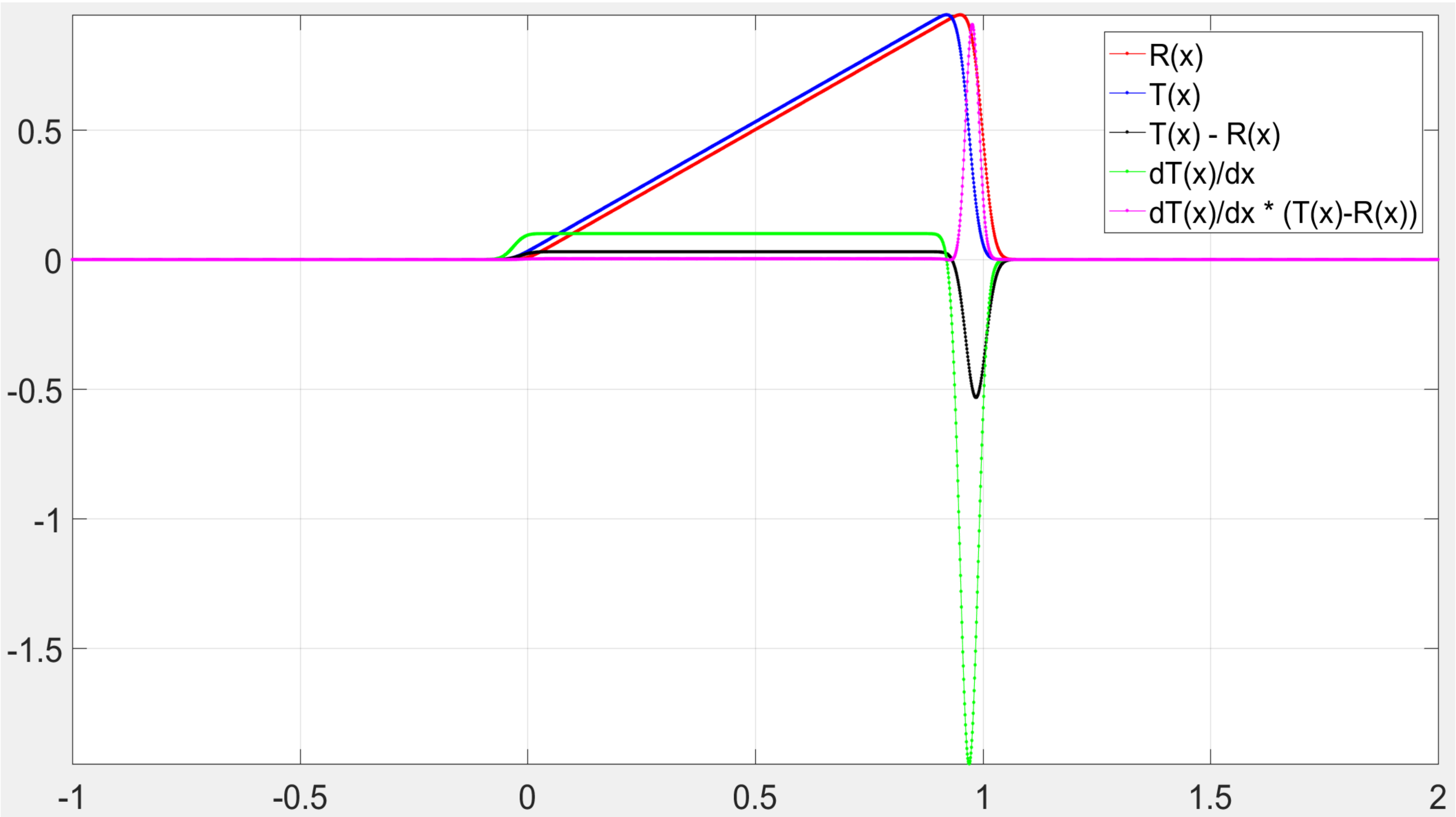


Image Registration

- Aligning ramps (w.r.t. translation)
  - Area under magenta curve > 0

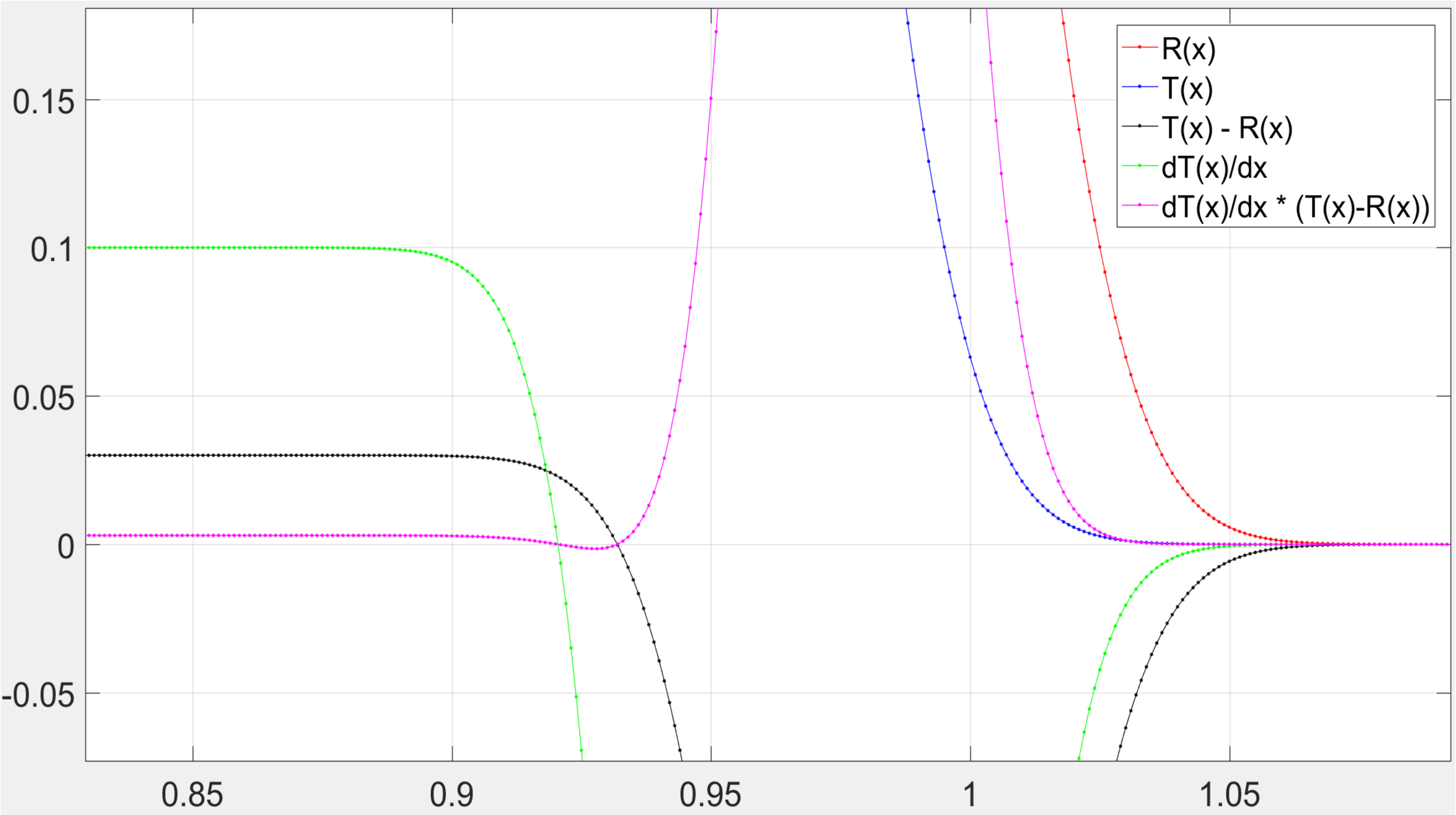


Image Registration

- Aligning ramps (w.r.t. translation)
  - Area under magenta curve **< 0**

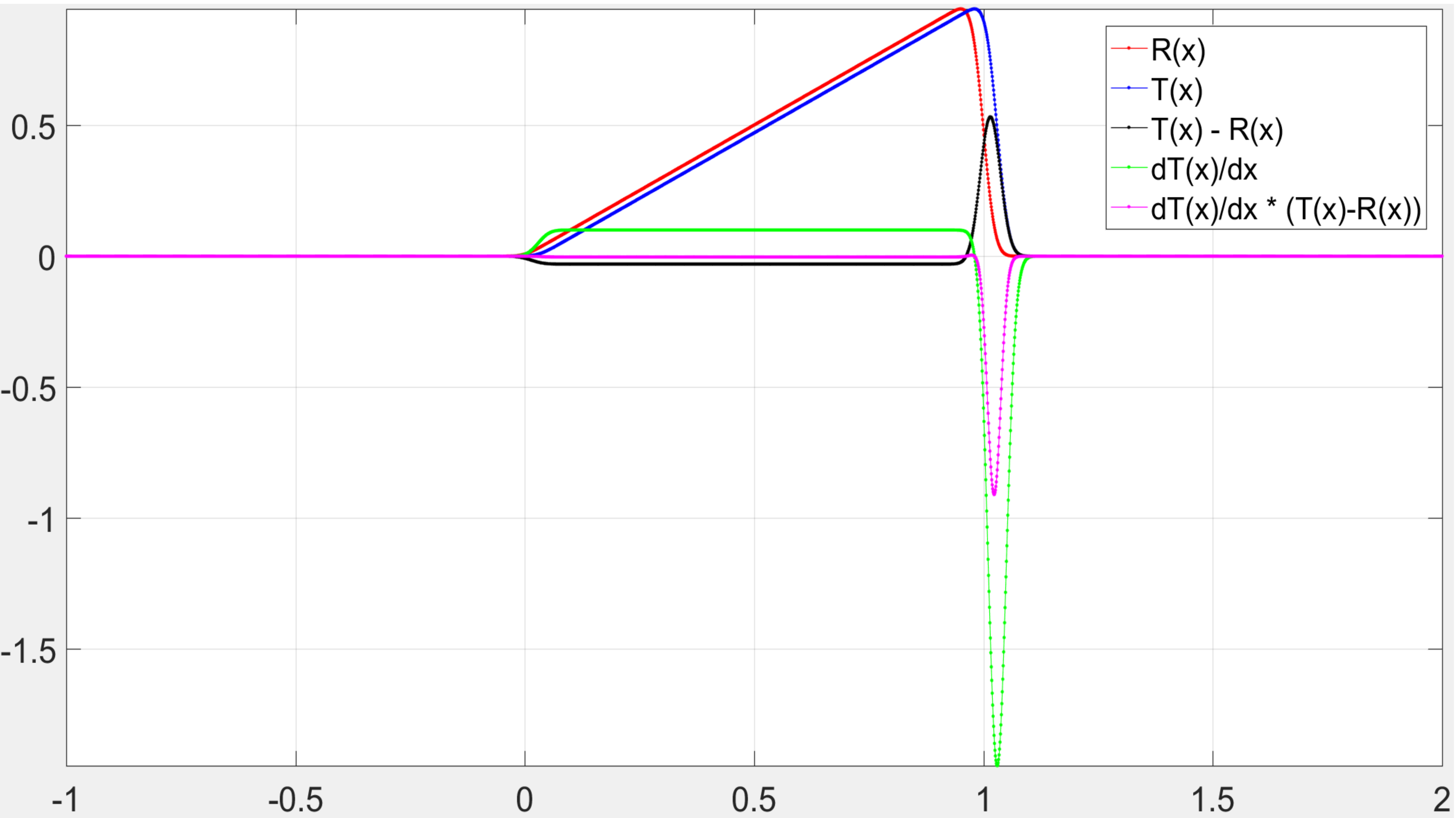


Image Registration

- Aligning ramps (w.r.t. translation)
  - Area under magenta curve **< 0**

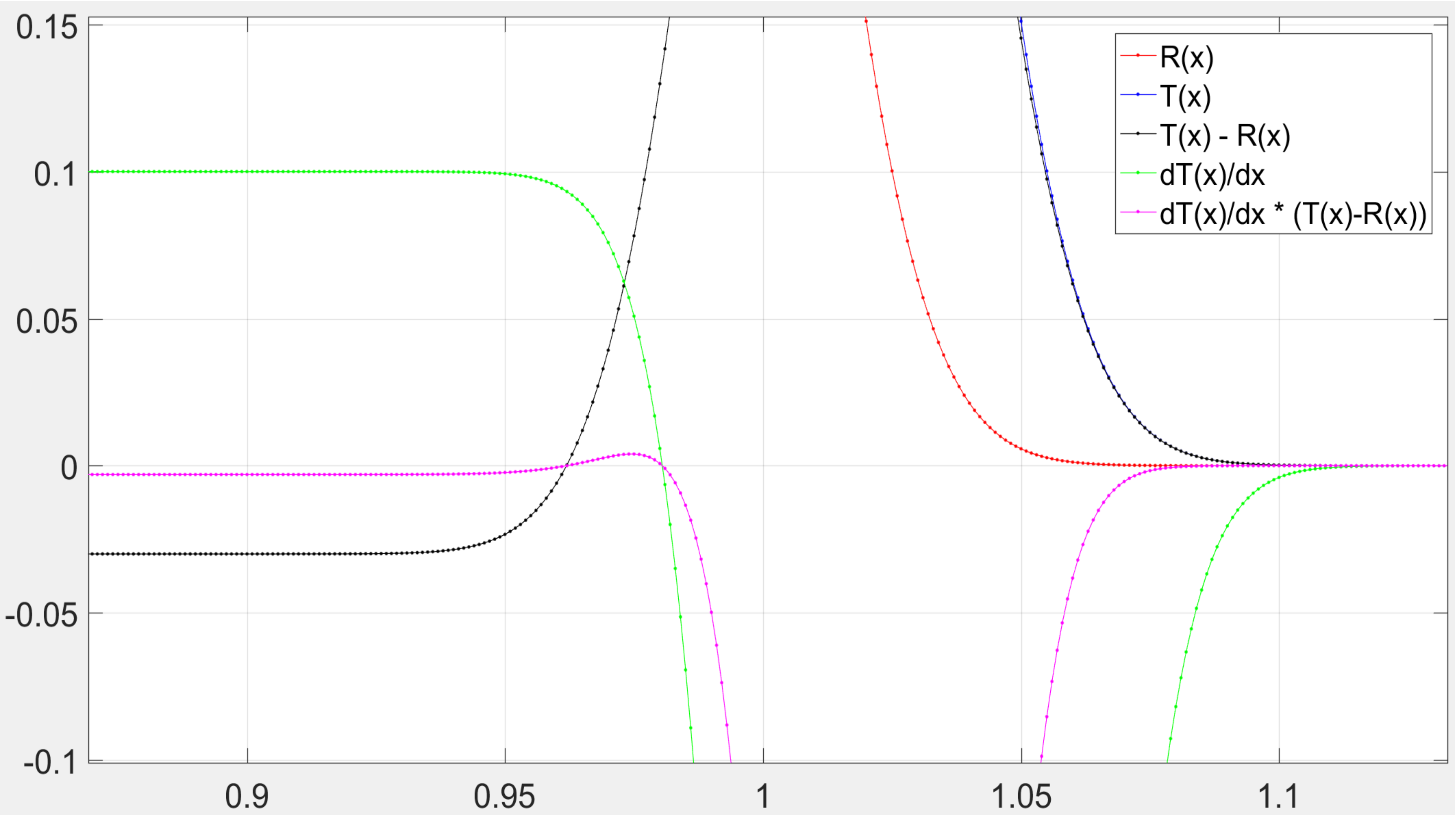


Image Registration

$$J(A,b) := \sum_{i=1}^{N} \Big(T(\phi(x_i;A,b)) - R(x_i) \Big)^2$$

- **Gradient descent w.r.t. 'A'**
  - Using chain rule, partial derivative of objective function is

$$\frac{\partial J(A,b)}{\partial A:} := \sum_{i=1}^{N} 2\Big(T(\phi(x_i;A,b)) - R(x_i) \Big) \frac{\partial T(\phi(x_i;A,b))}{\partial \phi(x_i;A,b)} \frac{\partial \phi(x_i;A,b)}{\partial A:}$$

$$= \sum_{i=1}^{N} 2\Big(T(\phi(x_i;A,b)) - R(x_i) \Big) \frac{\partial T(\phi(x_i;A,b))}{\partial \phi(x_i;A,b)} \Big(x_i^\top \otimes \mathbf{I}_{3\times 3} \Big)$$

because $\frac{\partial}{\partial A:}\phi(x) = \frac{\partial}{\partial A:}(Ax+b) = x^\top \otimes \mathbf{I}_{3\times 3}$

- A: = vectorized A by concatenating columns
- ⊗ = Kronecker product

Image Registration

- Gradient descent w.r.t. 'A'
  - Let spatial derivative $\frac{\partial T(\phi(x_i;A,b))}{\partial \phi(x_i;A,b)}$ equal the row vector $[\nabla T_{i1}\ \nabla T_{i2}\ \nabla T_{i3}]$
  - Let x_i be $[x_{i1}\ x_{i2}\ x_{i3}]^\top$
  - Then, what is $\frac{\partial T(\phi(x_i;A,b))}{\partial \phi(x_i;A,b)}\left(x_i^\top \otimes \mathbf{I}_{3\times 3}\right)$

$$[\nabla T_{i1}\ \nabla T_{i2}\ \nabla T_{i3}]\begin{bmatrix} x_{i1} & 0 & 0 & x_{i2} & 0 & 0 & x_{i3} & 0 & 0 \\ 0 & x_{i1} & 0 & 0 & x_{i2} & 0 & 0 & x_{i3} & 0 \\ 0 & 0 & x_{i1} & 0 & 0 & x_{i2} & 0 & 0 & x_{i3} \end{bmatrix}$$

$$=[\ \nabla T_{i1}x_{i1}\ \nabla T_{i2}x_{i1}\ \nabla T_{i3}x_{i1}\ \ \nabla T_{i1}x_{i2}\ \nabla T_{i2}x_{i2}\ \nabla T_{i3}x_{i2}\ \ \nabla T_{i1}x_{i3}\ \nabla T_{i2}x_{i3}\ \nabla T_{i3}x_{i3}\ ]$$

Image Registration

- Gradient descent w.r.t. 'A'
  - derivative

$$=[\ \nabla T_{i1}x_{i1}\ \nabla T_{i2}x_{i1}\ \nabla T_{i3}x_{i1}\ \ \nabla T_{i1}x_{i2}\ \nabla T_{i2}x_{i2}\ \nabla T_{i3}x_{i2}\ \ \nabla T_{i1}x_{i3}\ \nabla T_{i2}x_{i3}\ \nabla T_{i3}x_{i3}\ ]$$

  - Gradient = transpose of derivative
    - Same size as A:
  - Reshape gradient as a matrix (of same shape as A)

$$\begin{bmatrix} \nabla T_{i1}x_{i1} & \nabla T_{i1}x_{i2} & \nabla T_{i1}x_{i3} \\ \nabla T_{i2}x_{i1} & \nabla T_{i2}x_{i2} & \nabla T_{i2}x_{i3} \\ \nabla T_{i3}x_{i1} & \nabla T_{i3}x_{i2} & \nabla T_{i3}x_{i3} \end{bmatrix} = \begin{bmatrix} \nabla T_{i1} \\ \nabla T_{i2} \\ \nabla T_{i3} \end{bmatrix} [x_{i1}\ x_{i2}\ x_{i3}]$$

    - Outer product of
      spatial gradient col. vector with
      pixel location row vector

Image Registration

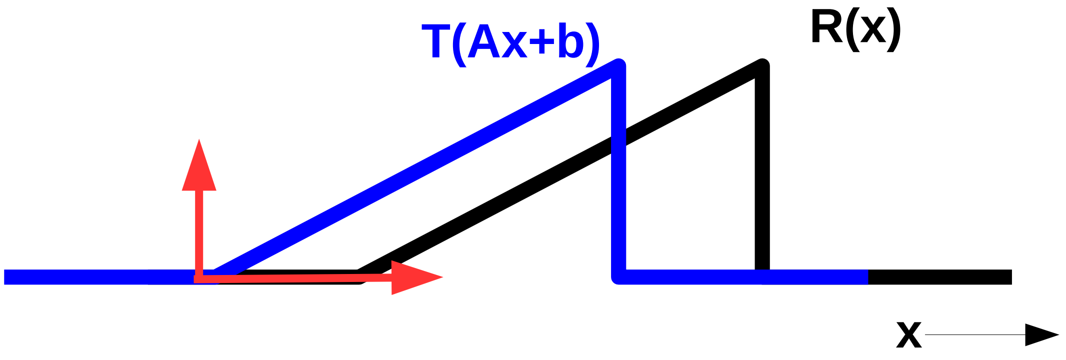
- Gradient descent w.r.t. 'A'

$$A \leftarrow A - \tau \sum_{i=1}^{N} 2\Big(T(\phi(x_i; A, b)) - R(x_i)\Big) \begin{bmatrix} \nabla T_{i1} \\ \nabla T_{i2} \\ \nabla T_{i3} \end{bmatrix} [x_{i1} \ x_{i2} \ x_{i3}]$$

$$= A - \tau \sum_{i=1}^{N} 2\Big(T(\phi(x_i; A, b)) - R(x_i)\Big) \nabla T(\phi(x_i)) x_i^\top$$

Image Registration

- Gradient descent w.r.t. “scaling” ‘a’ (in 1D)
  - Example
    - Φ(x) := ax + b
    - Assume a > 0



$$A - \tau \sum_{i=1}^{N} 2\Big(T(\phi(x_i; A, b)) - R(x_i)\Big) \nabla T(\phi(x_i)) x_i^\top$$

  - 2nd term = spatial derivative of T → ‘always’ positive
  - 1st term = positive (at ‘all’ 'x' where 2nd term > 0)
  - If ‘x_i’ all positive,
    gradient descent reduces 'a', i.e., shrinks the x-axis scale
    - Shrinking the coordinate axis → moves image T to the right

Image Registration

- Gradient descent w.r.t. 'A'

$$A \leftarrow A - \tau \sum_{i=1}^{N} 2\Big(T(\phi(x_i; A, b)) - R(x_i)\Big) \begin{bmatrix} \nabla T_{i1} \\ \nabla T_{i2} \\ \nabla T_{i3} \end{bmatrix} [x_{i1} \; x_{i2} \; x_{i3}]$$

$$= A - \tau \sum_{i=1}^{N} 2\Big(T(\phi(x_i; A, b)) - R(x_i)\Big) \nabla T(\phi(x_i)) x_i^\top$$

  - This requires computing **gradient** at sub-pixel locations, for each descent step
  - This can become computationally cheaper
  - How ?

Image Registration

- Gradient descent w.r.t. 'A'

$$A - \tau \sum_{i=1}^{N} 2\Big(T(\phi(x_i; A, b)) - R(x_i) \Big) \nabla T(\phi(x_i)) x_i^\top$$

  - Substitute y := Φ(x)
  - Replace summation / integration over x to summation / integration over y

$$A \leftarrow A - \tau \sum_{i=1}^{N} 2\Big(T(y_i) - R(\phi^{-1}(y_i; A, b)) \Big) \nabla T(y_i) \phi^{-1}(y_i; A, b)^\top |A|^{-1}$$

  - ∇ T (yi) = gradient image T at pixel location yi
  - $|A|^{-1}$= inverse det. Jacobian of transformation Φ(x) = Ax+b
    - Can be absorbed in step size τ, and ignored

Image Registration

- Atlas construction
  - Effect of the choice of the deformation model

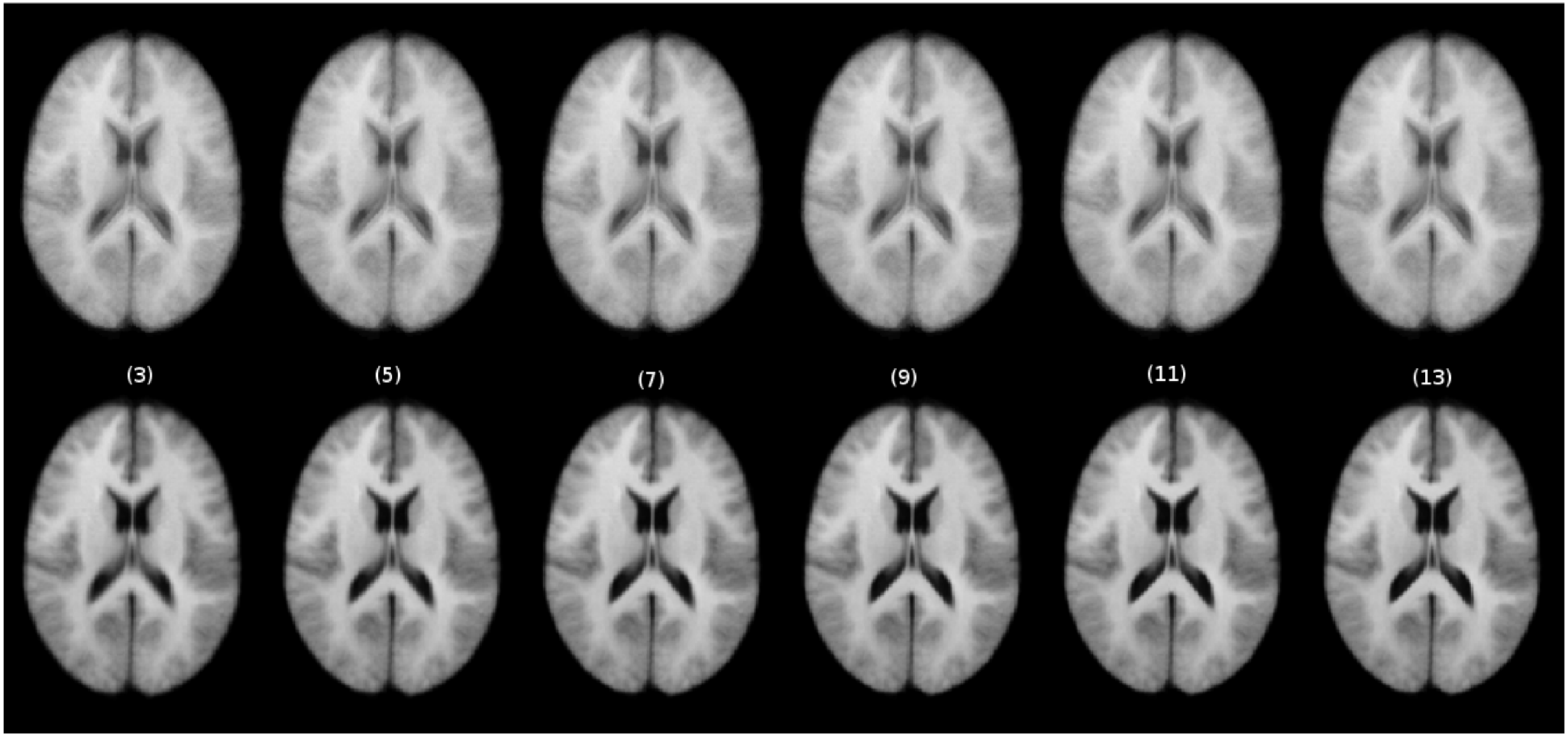


Figure 11: Atlas results with 3, 5, 7, 9, 11 and 13 inputs constructed by (a) arithmetically averaging rigidly aligned images (top row) and (b) Greedy Iterative Average template construction (bottom row)

Multi-scale Unbiased Diffeomorphic Atlas Construction on Multi-GPUs. Ha, Kruger, Joshi, Silva. 2010.

Image Registration

- How to reduce chances of getting stuck in local minima for textured images ?
  - Optimization at multiple image scales
    - (Gaussian) scale-space pyramid
    - Similar to annealing
  - First align coarse-scale structures
  - Gradually move to alignment of fine-scale features

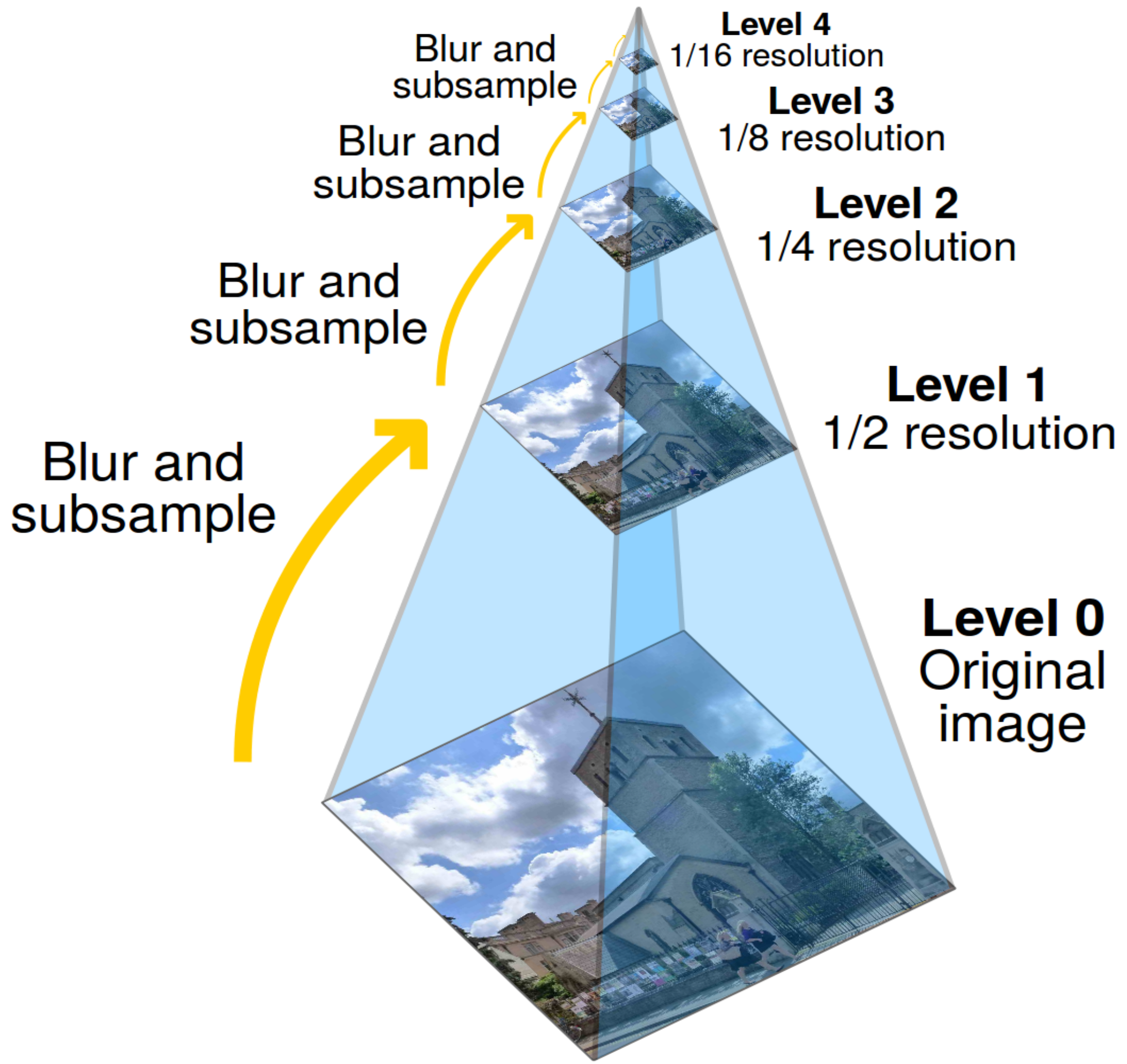


Image Registration

- Align 2 Gaussians of same height (w.r.t. translation)
  - Area under magenta curve > 0

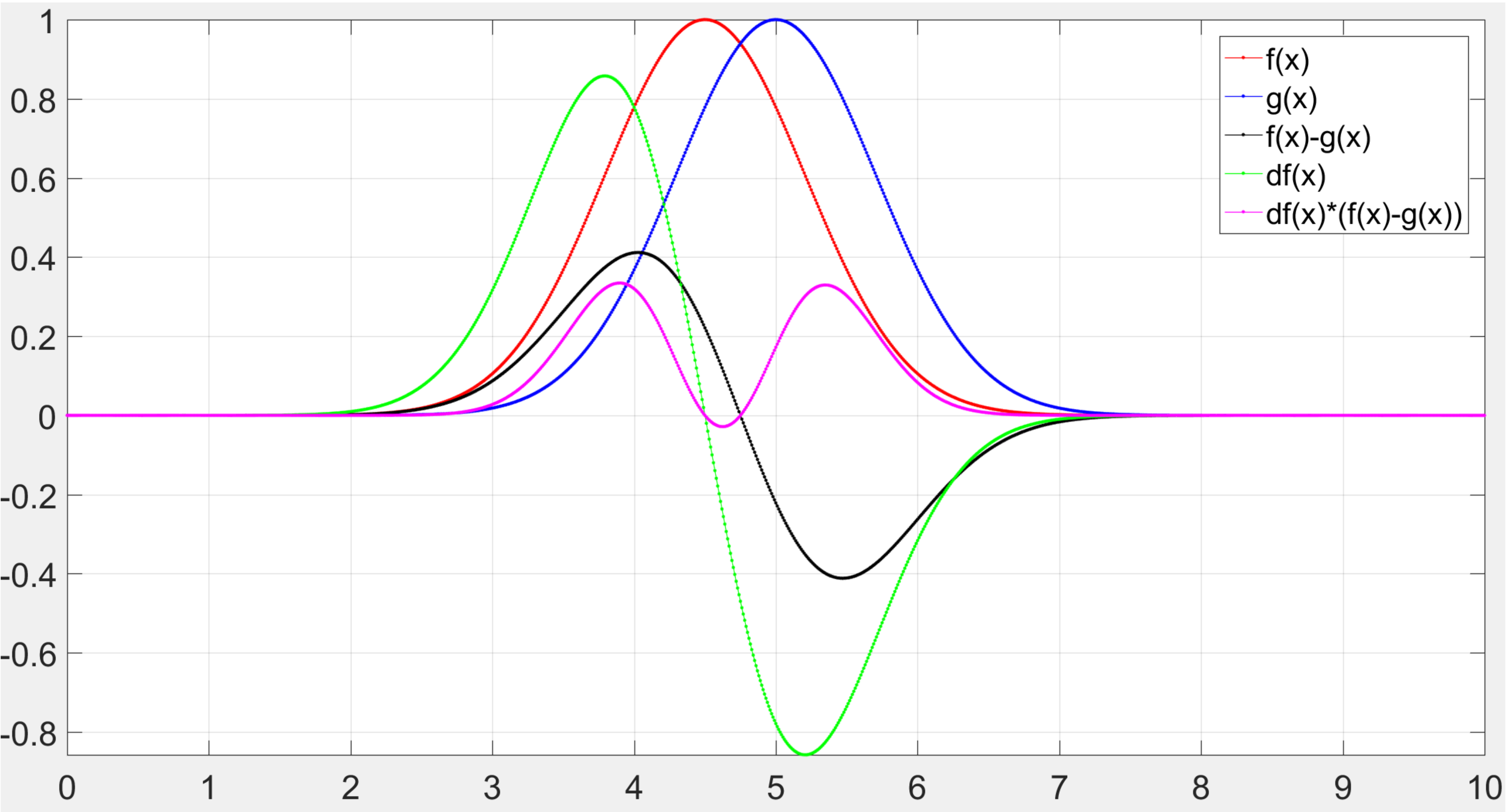


Image Registration

- Align 2 Gaussians of same height (w.r.t. translation)
  - Area under magenta curve < 0

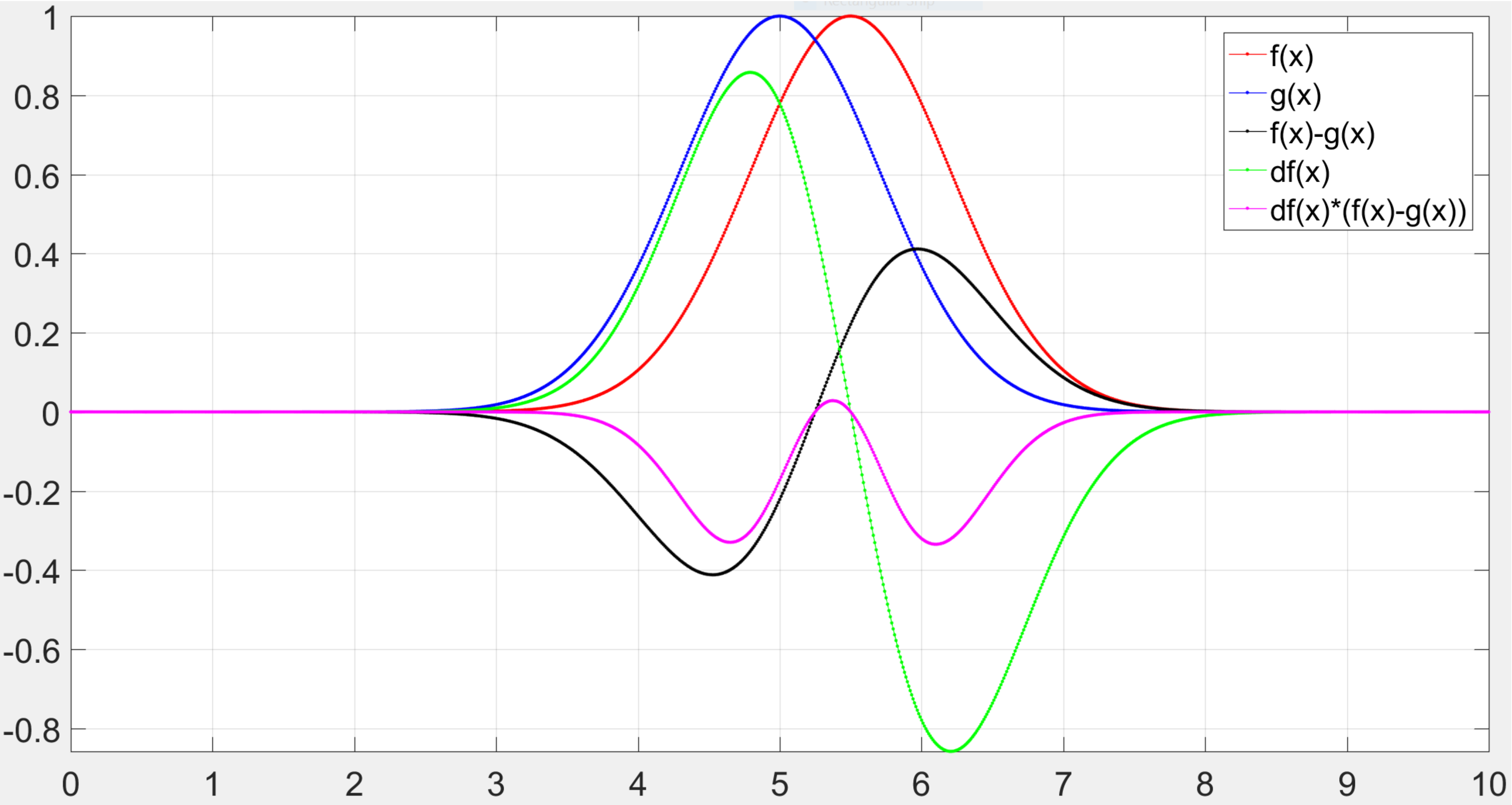


Image Registration

- Align 2 Gaussians of **diff.** height (w.r.t. translation)
  - Area under magenta curve > 0

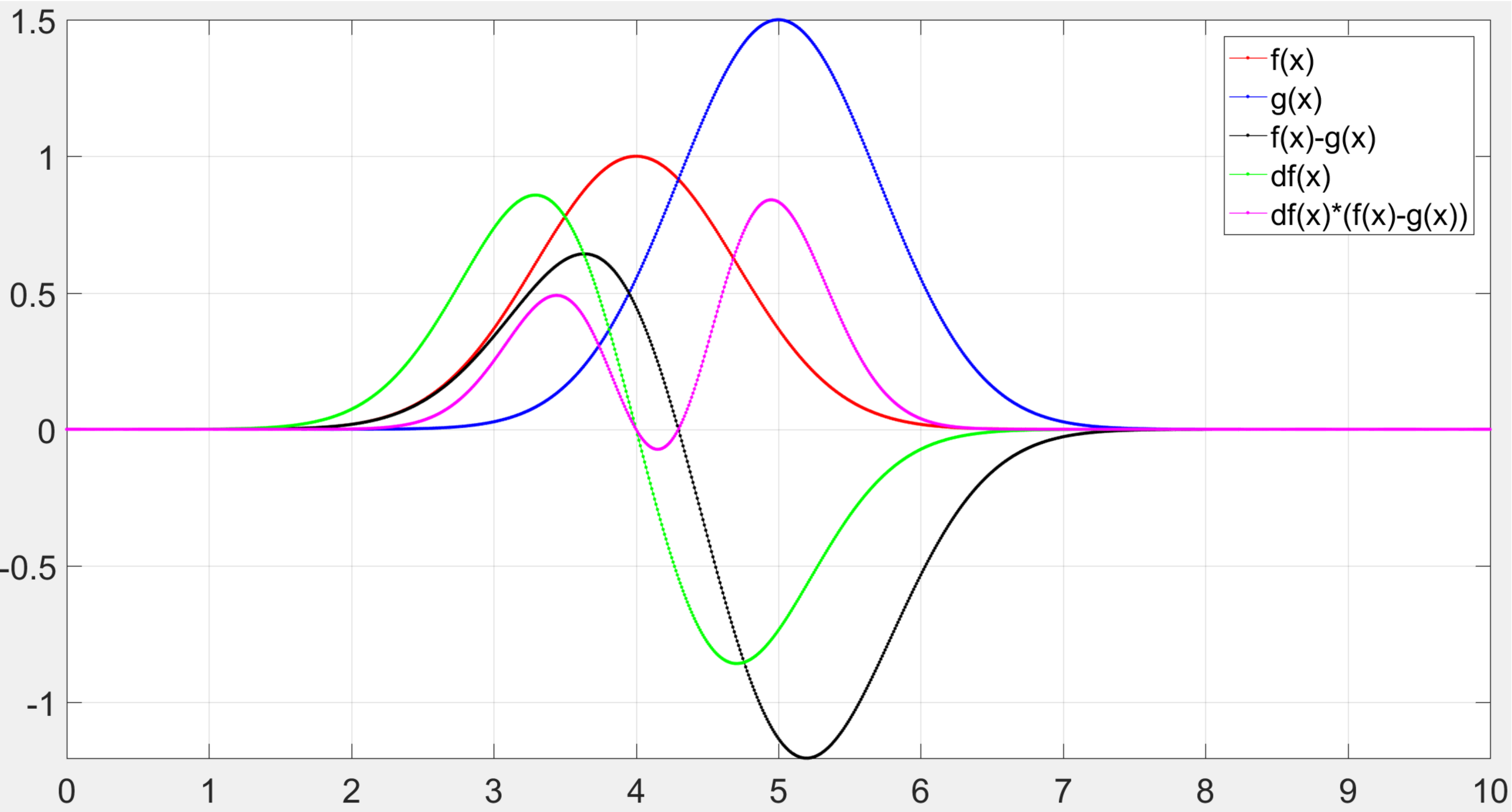


Image Registration

- Align 2 Gaussians of **diff.** height (w.r.t. translation)
  - Area under magenta curve < 0