

# Coursera Class

# Computational Anatomy

# and Neuroinformatics

[https://en.wikipedia.org/wiki/Computational\\_Anatomy](https://en.wikipedia.org/wiki/Computational_Anatomy)

Michael I. Miller

Tilak Ratnanather

Daniel Tward



# Matrix Groups Transforming Images

Michael I. Miller  
Tilak Ratnanather  
Daniel Tward

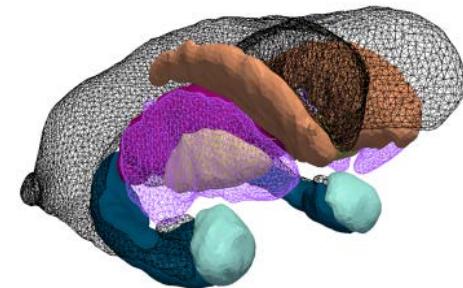


# Matrix Group Transformations

# Matrix Groups for Shifting, Scaling, Rotation

Transformation coordinates:  $x = (x_1, x_2) \in \mathbb{R}^2$

$$\phi(x) = \begin{pmatrix} \phi_1(x) \\ \phi_2(x) \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

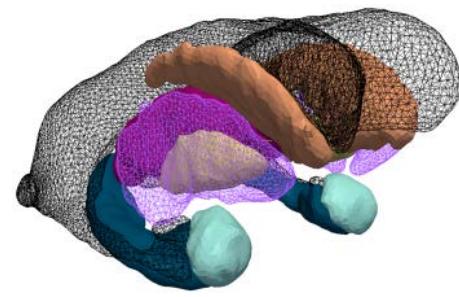


Translate:  $\phi(x) = \begin{pmatrix} x_1 + b_1 \\ x_2 + b_2 \end{pmatrix}$

Scale+Translate.:  $\phi(x) = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

Rotate+Trans.:  $\phi(x) = \underbrace{\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}}_{2 \times 2 \text{ Rotation}} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

# Vector Field Notation:



$$\phi(x) = x + v(x)$$

Spatial Coordinates and Vector Notation

$$x \doteq (x_1, x_2) \in \mathbb{R}^2$$

$$v(x) \doteq \begin{pmatrix} v_1(x_1, x_2) \\ v_2(x_1, x_2) \end{pmatrix}$$

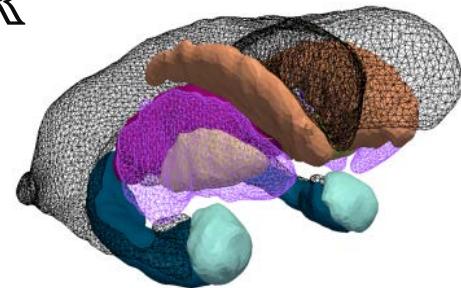
$$x \doteq (x_1, x_2, x_3) \in \mathbb{R}^3$$

$$v(x) \doteq \begin{pmatrix} v_1(x_1, x_2, x_3) \\ v_2(x_1, x_2, x_3) \\ v_3(x_1, x_2, x_3) \end{pmatrix}$$

# Vector fields for Translation-Rotation-Scale

Transformation coordinates:

$$\phi(x) = x + v(x), \quad x = (x_1, x_2) \in \mathbb{R}^2$$



Translate:  $v(x) = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$

Rotate:  $v(x) = \begin{pmatrix} \cos \theta - 1 & -\sin \theta \\ \sin \theta & \cos \theta - 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

Scale:  $v(x) = (s - 1) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

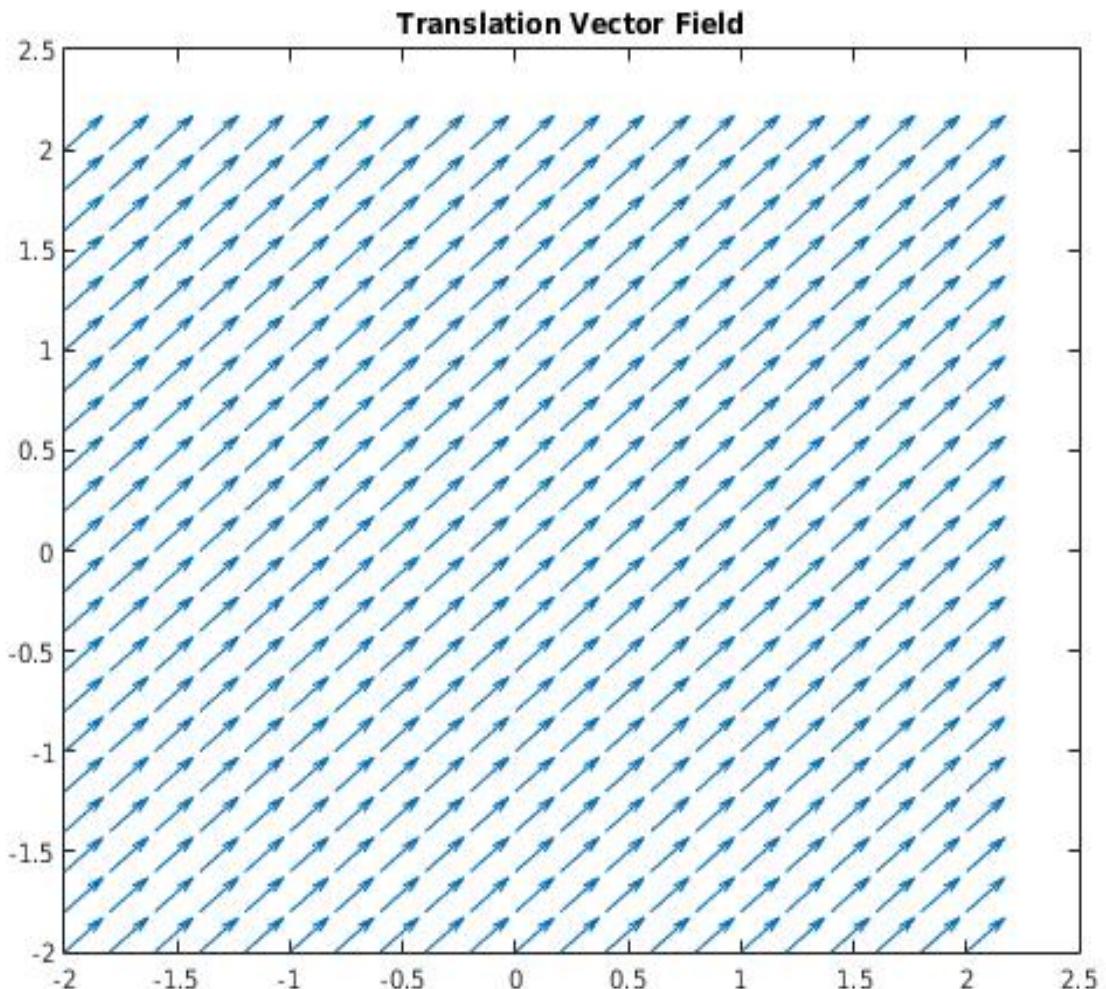
# Translation

Translation Vector Field  
in the plane

$$x = (x_1, x_2) \in \mathbb{R}^2$$

$$\phi(x) = \begin{pmatrix} x_1 + b_1 \\ x_2 + b_2 \end{pmatrix}$$

$$v(x) = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$



2-dimension

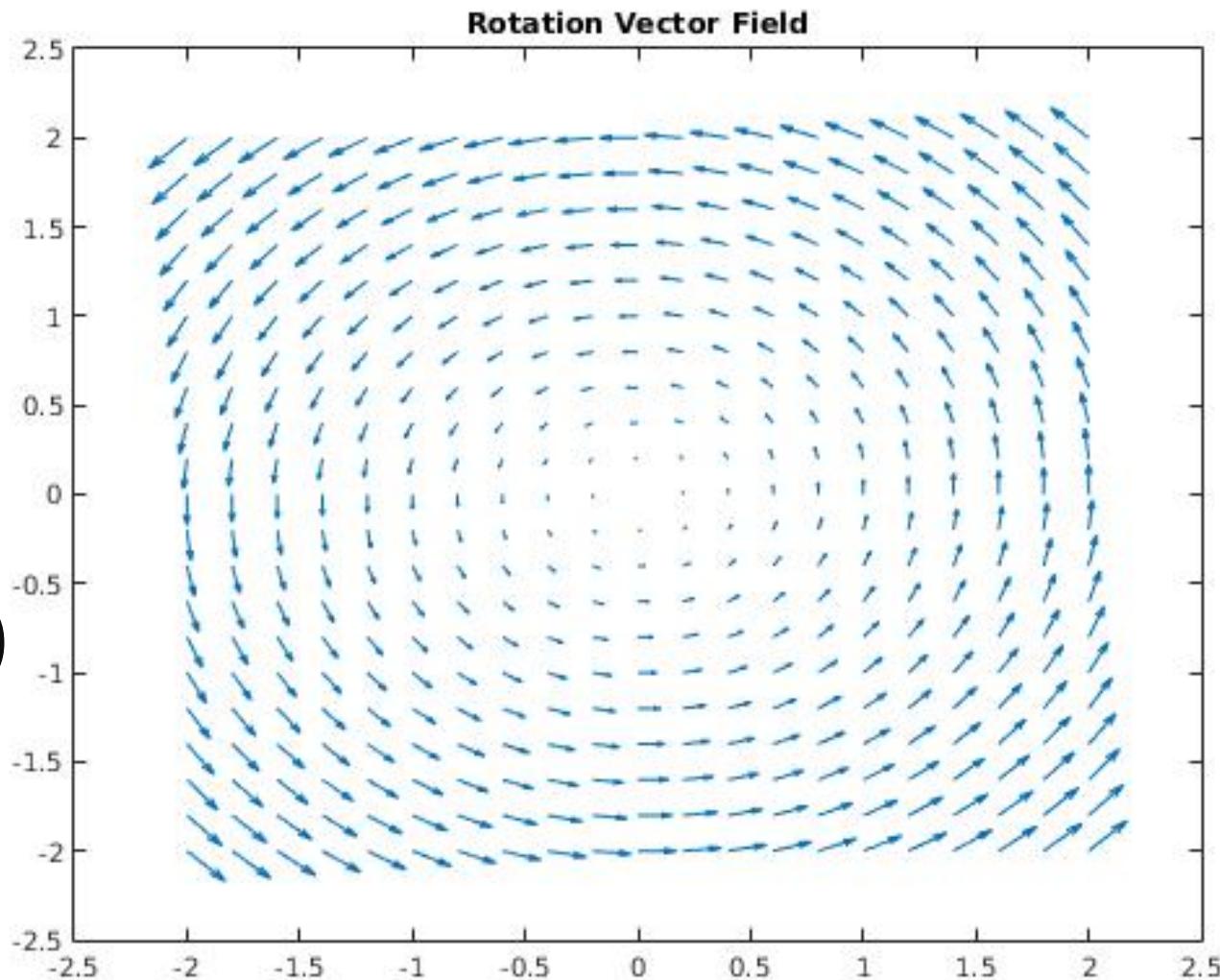
# Rotation

Rotation Vector Field  
in the plane

$$x = (x_1, x_2) \in \mathbb{R}^2$$

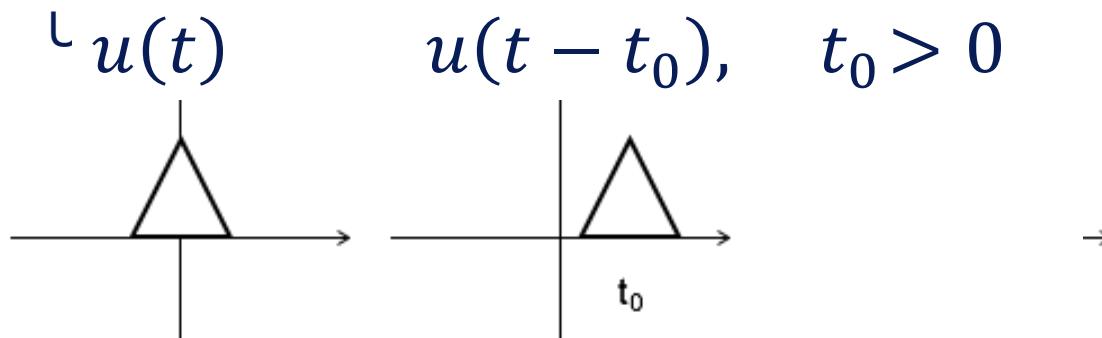
$$\phi(x) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$v(x) = \begin{pmatrix} \cos \theta - 1 & -\sin \theta \\ \sin \theta & \cos \theta - 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

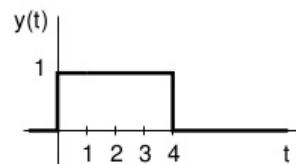
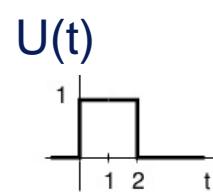


$\theta$ : 1-dimension

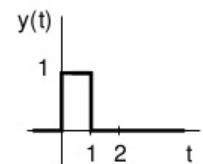
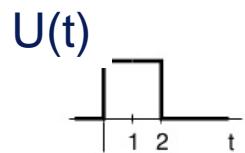
# Transformation of signal is “opposite”: shifting and scaling of time



$$U(t) = u_s(t) - u_s(t-2)$$
$$y(t) = U(t/2)$$



$$U(t) = u_s(t) - u_s(t-2)$$
$$y(t) = U(2t)$$



This has effectively “slowed down”  $U(t)$  by a factor of 2  
(What occurred at  $t=1$  now occurs at  $t=2$ )

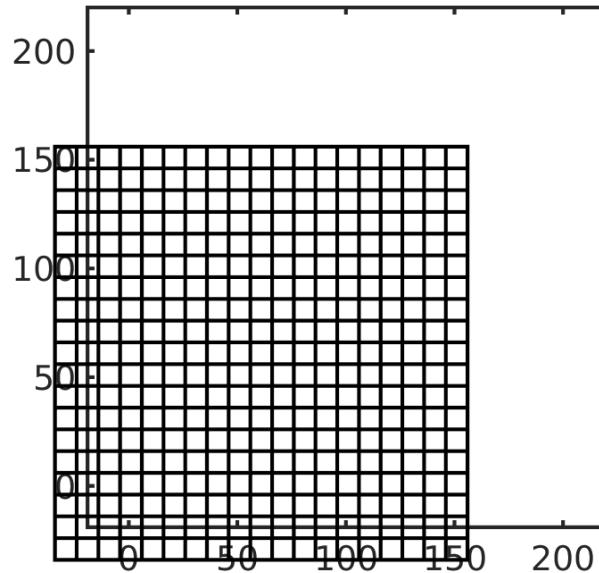
Computational Anatomy: Miller, Ratnanather, Tward9

This has effectively “sped up”  $U(t)$  by a factor of 2  
(What occurred at  $t=2$  now occurs at  $t=2/2=1$ )

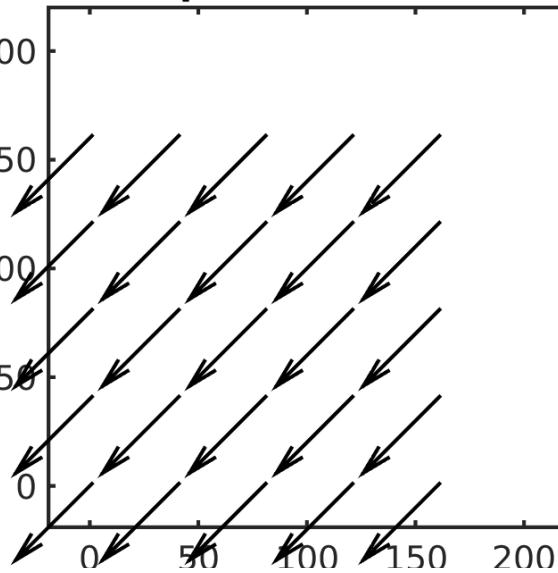
# **Matrix Group Transformation Model of Images**

# Image Translation: $I(x + b)$

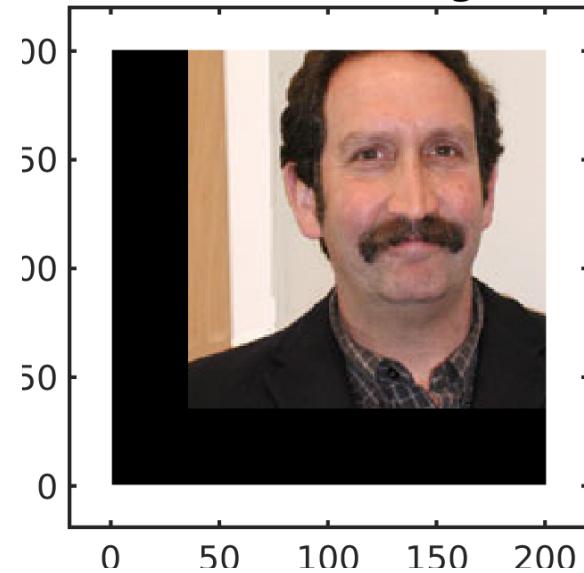
Deformation field



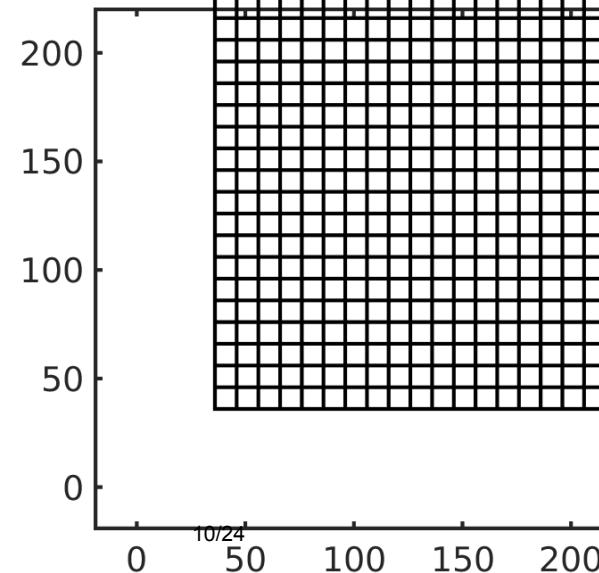
Displacement field



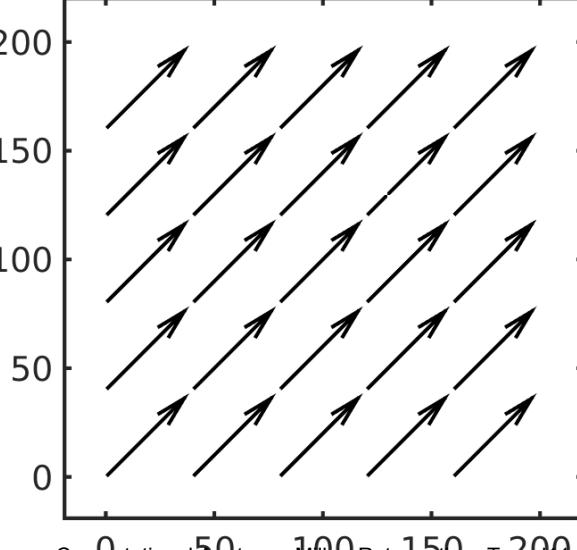
Deformed image



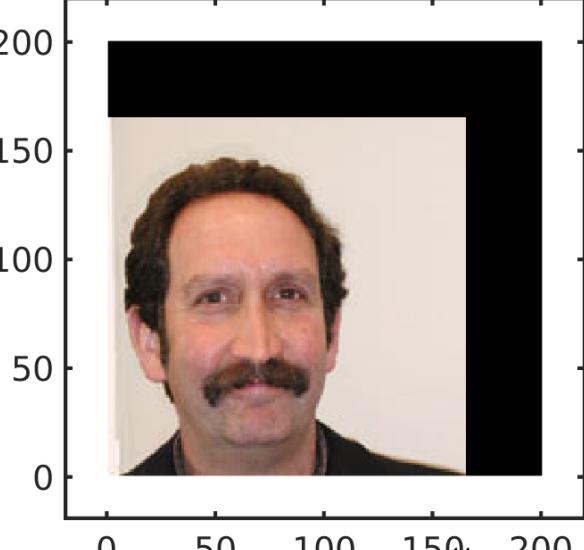
Deformation field



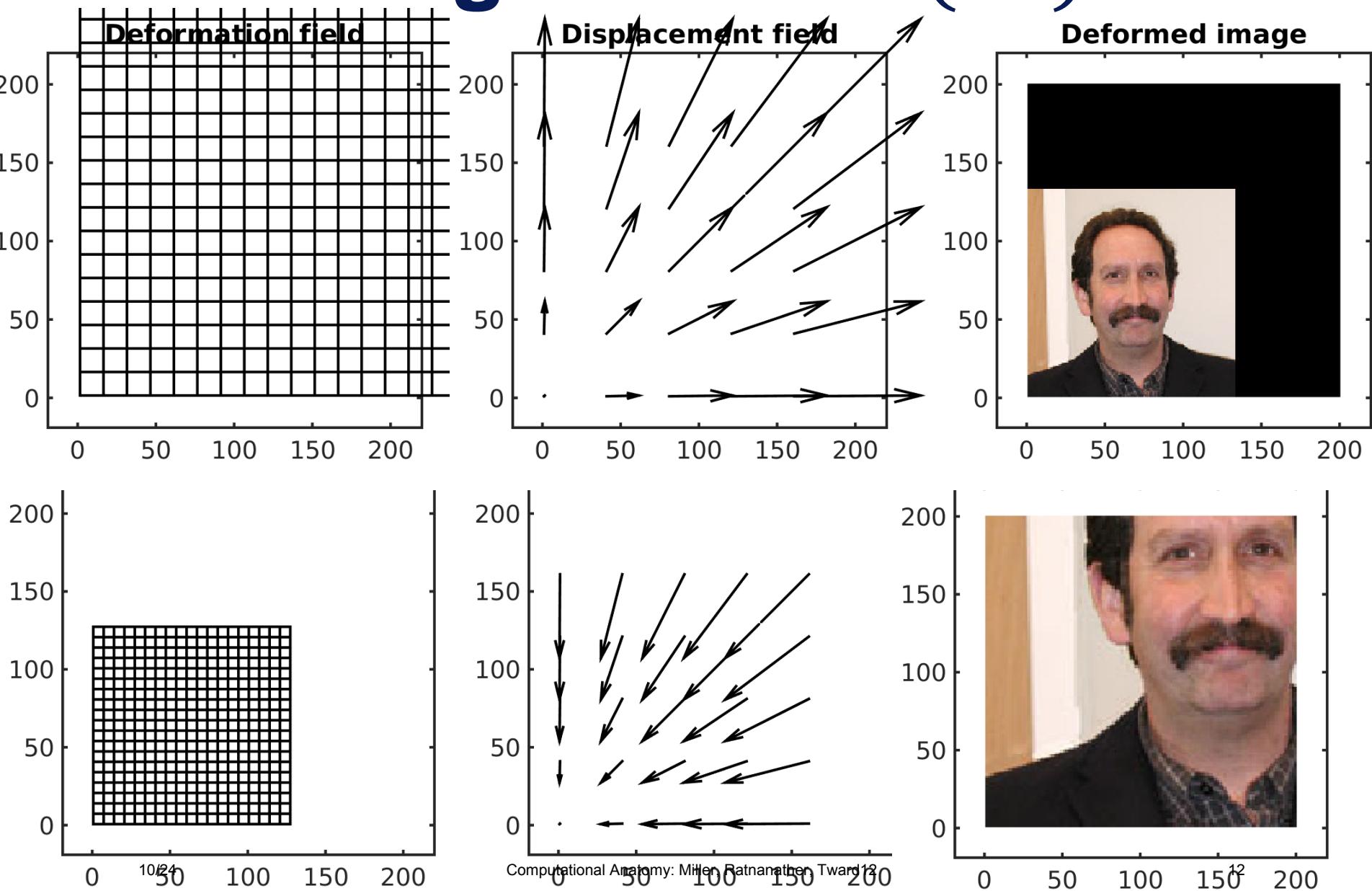
Displacement field



Deformed image

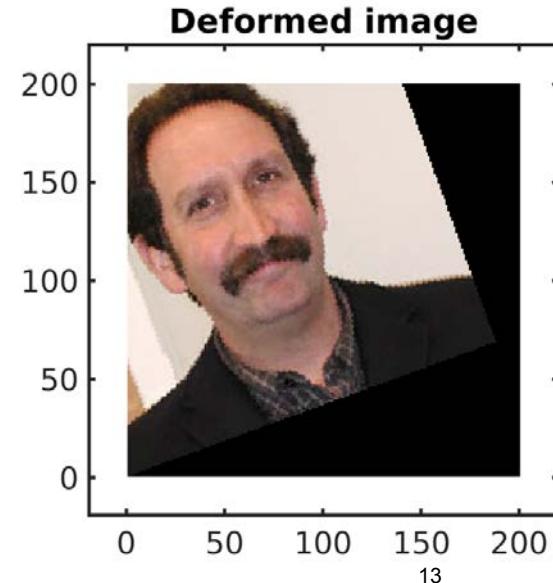
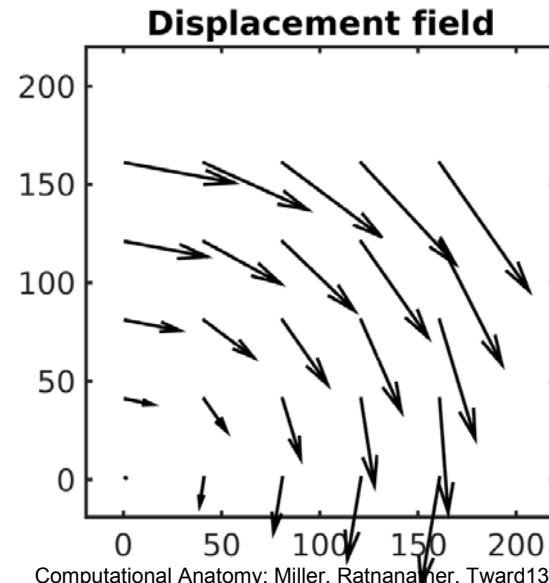
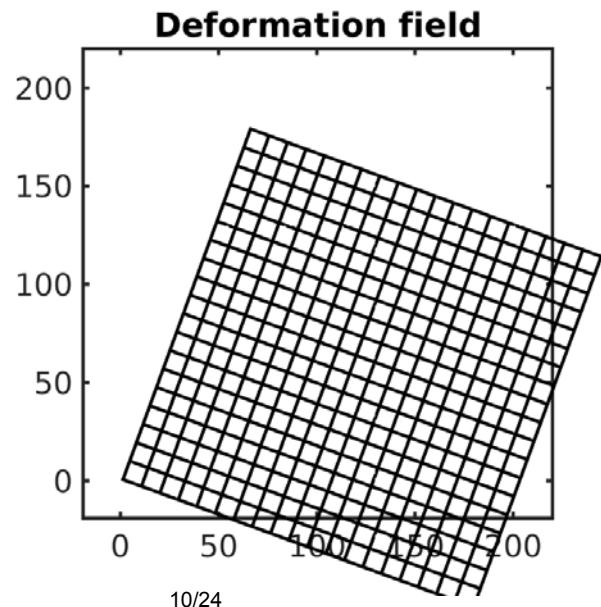
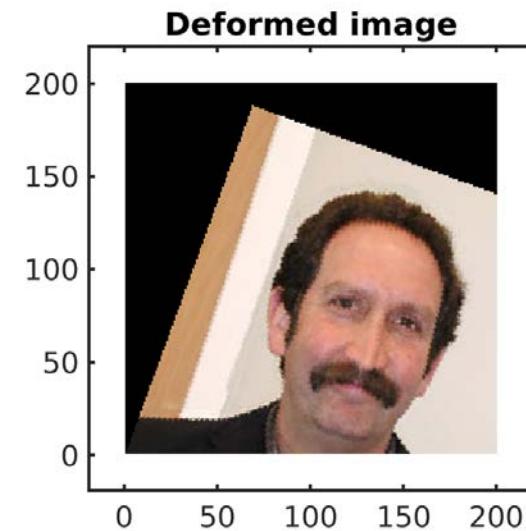
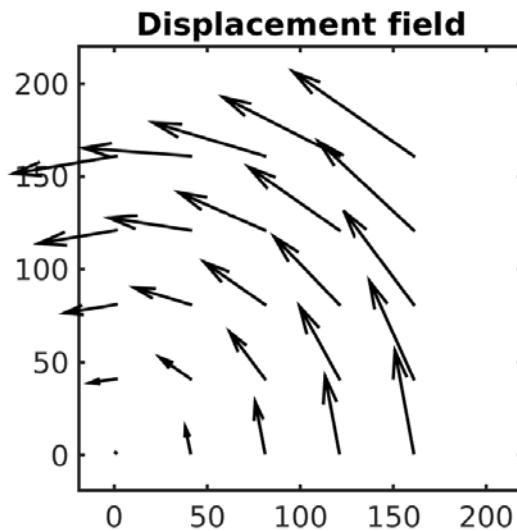
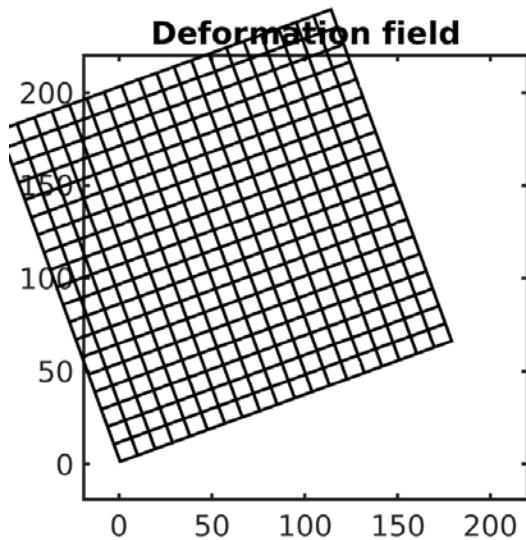


# Image Scale: $I(sx)$



# Image Rotation

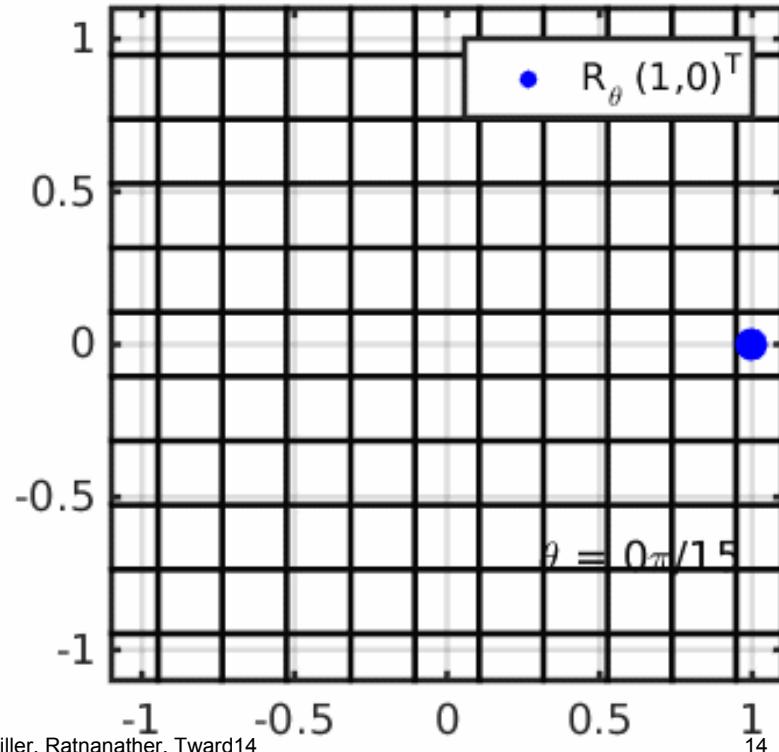
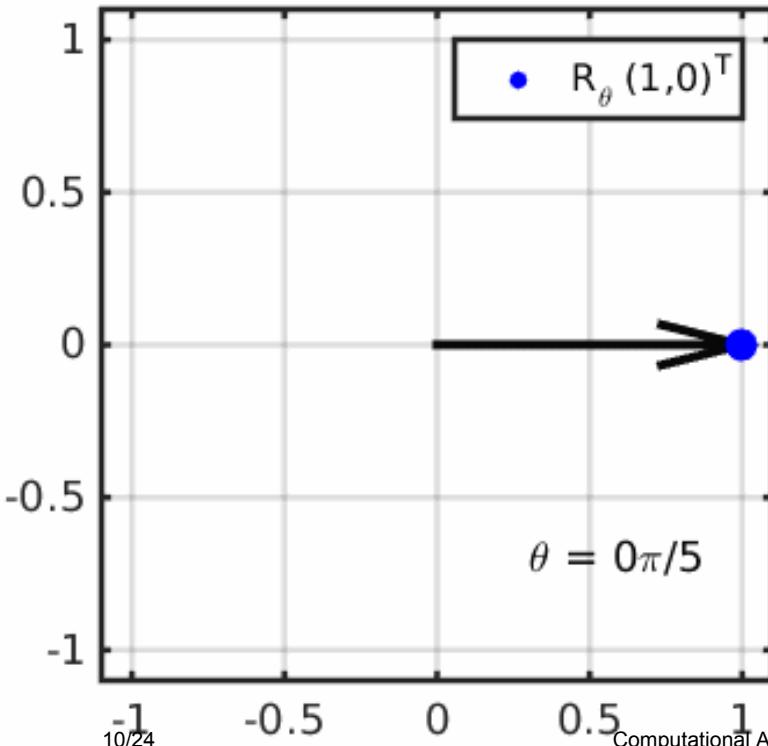
$$I \left( \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} x \right)$$



# Rotation Coordinates Counter Clockwise

Counter Clockwise

Rotation:  $\phi_\theta(x) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$



# Images via Template Transformation

Rotation:

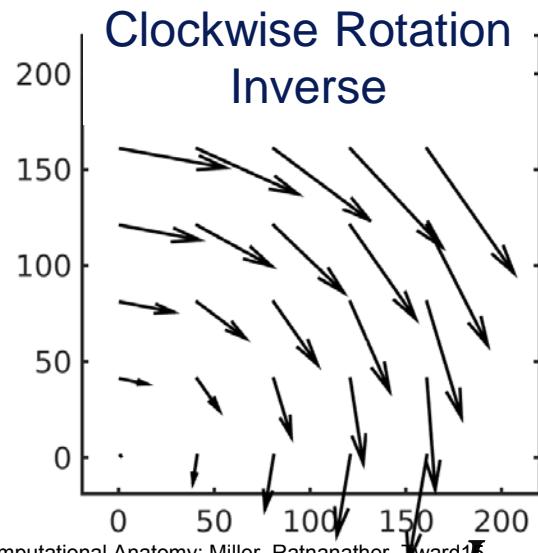
$$\phi_\theta(x) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Inverse:

$$\phi_\theta^{-1}(x) = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



$I_{temp}^{10/24}(x)$



$I_{temp} \circ \phi_{45}^{-1}, \theta = 45$

# Image Rotation

Rotation:

$$\phi_\theta x = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



$$I(\phi_{-45}x)$$

10/24



$$I(x)$$

Computational Anatomy: Miller, Rathnather, Tward16



$$I(\phi_{45}x)$$

16

In Metric Pattern Theory,  
Ulf Grenander coined the term  
*Deformable Template*  
to represent the space of  
objects generated via groups  
which transform  
exemplars.

[https://en.wikipedia.org/wiki/Computational\\_anatomy#The\\_deformable\\_template\\_orbit\\_model\\_of\\_computational\\_anatomy](https://en.wikipedia.org/wiki/Computational_anatomy#The_deformable_template_orbit_model_of_computational_anatomy)



# The Orbit of Images in Computational Anatomy

We call the  
*-ideal, prototype, exemplar, template, atlas-*  
 $I_{temp}$

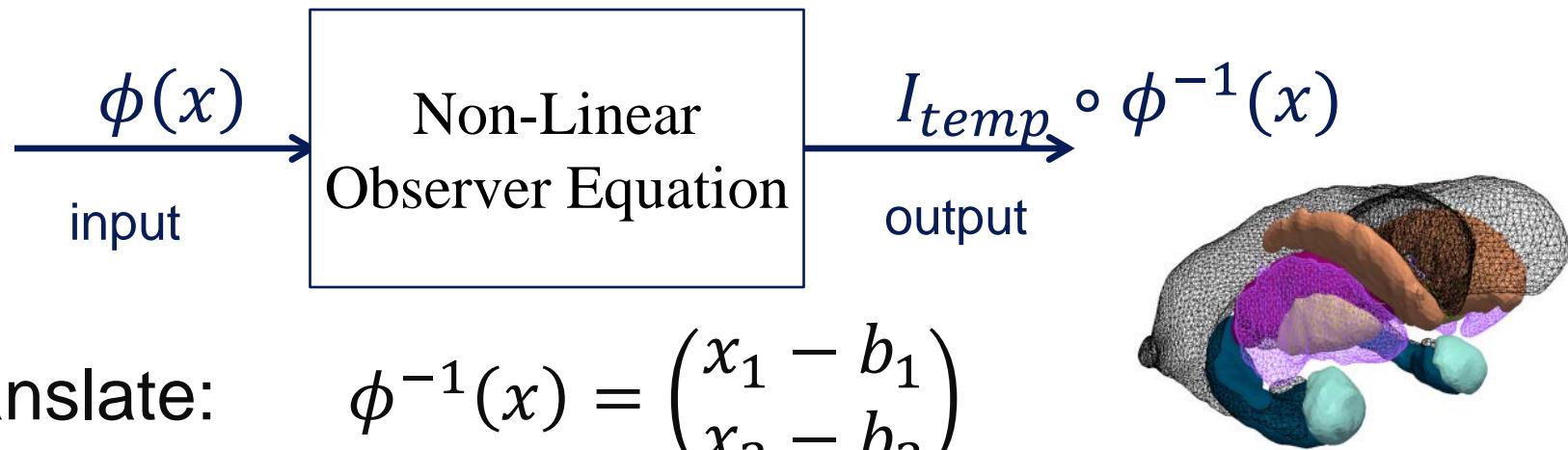
We call the set of images  
*-the Orbit-*

$$I = I_{temp} \circ \phi^{-1}$$

# The Imaging Model for Matrix Group

Transformation coordinates:  $x = (x_1, x_2) \in \mathbb{R}^2$

$$\phi: x \mapsto \phi(x) = \begin{pmatrix} \phi_1(x) \\ \phi_2(x) \end{pmatrix}$$



Translate:  $\phi^{-1}(x) = \begin{pmatrix} x_1 - b_1 \\ x_2 - b_2 \end{pmatrix}$

Scale.:  $\phi^{-1}(x) = \begin{pmatrix} 1/s & 0 \\ 0 & 1/s \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

Rotate:  $\phi^{-1}(x) = \begin{pmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

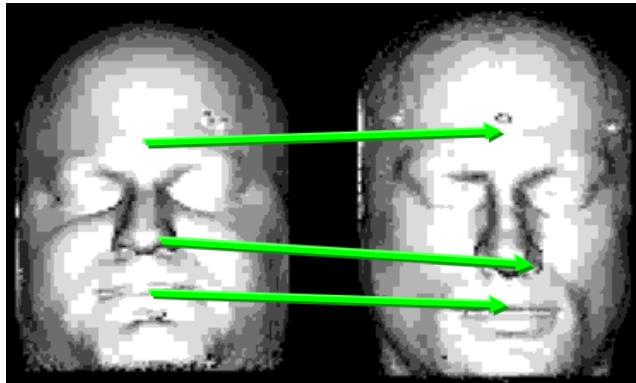
# Estimating Matrix Transformations

Daniel Tward:

- Program for Manual Transformation

Derivation of Transformations

# Translation Registration: 2D



- Given landmark B.C.'s

$$\begin{pmatrix} x_{11}, y_{11} \\ x_{21}, y_{21} \end{pmatrix}, \begin{pmatrix} x_{12}, y_{12} \\ x_{22}, y_{22} \end{pmatrix}, \dots \begin{pmatrix} x_{1n}, y_{1n} \\ x_{2n}, y_{2n} \end{pmatrix}$$

- Translation 2 dimensions

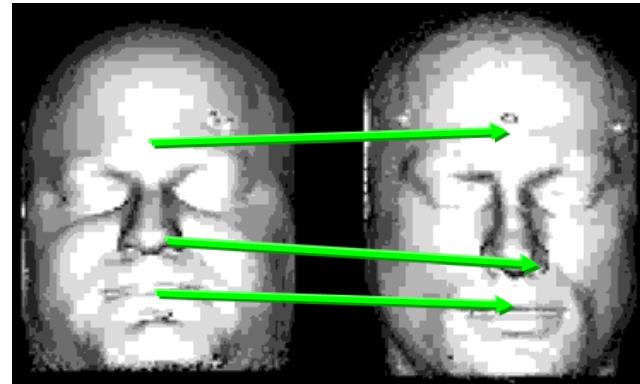
$$y_i = x_i + b = \begin{pmatrix} x_{1i} \\ x_{2i} \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

- Problem: Estimate Translation

$$\min_{b \in \mathbb{R}^2} \sum_i \underbrace{|y_i - (x_i + b)|^2}_{|b|^2 = b \cdot b = b^T b = \sum_i b_i b_i}$$

Computational Anatomy: Miller, Rathnaather, Tward21

# Matrix Registration



- Given landmarks

$$\left( \begin{matrix} x_{11}, y_{11} \\ x_{21}, y_{21} \end{matrix} \right), \left( \begin{matrix} x_{12}, y_{12} \\ x_{22}, y_{22} \end{matrix} \right), \dots \left( \begin{matrix} x_{1n}, y_{1n} \\ x_{2n}, y_{2n} \end{matrix} \right)$$

- Affine matrix 4 dimensions

$$y_i = \begin{pmatrix} y_{1i} \\ y_{2i} \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}}_A \underbrace{\begin{pmatrix} x_{1i} \\ x_{2i} \end{pmatrix}}_{x_i}$$

- Problem: Estimate affine matrix

$$\min_{A \in \mathbb{R}^{2 \times 2}} \sum_i \frac{|y_i - Ax_i|^2}{|a|^2 = a \cdot a = a^T a = \sum a_j a_j}$$

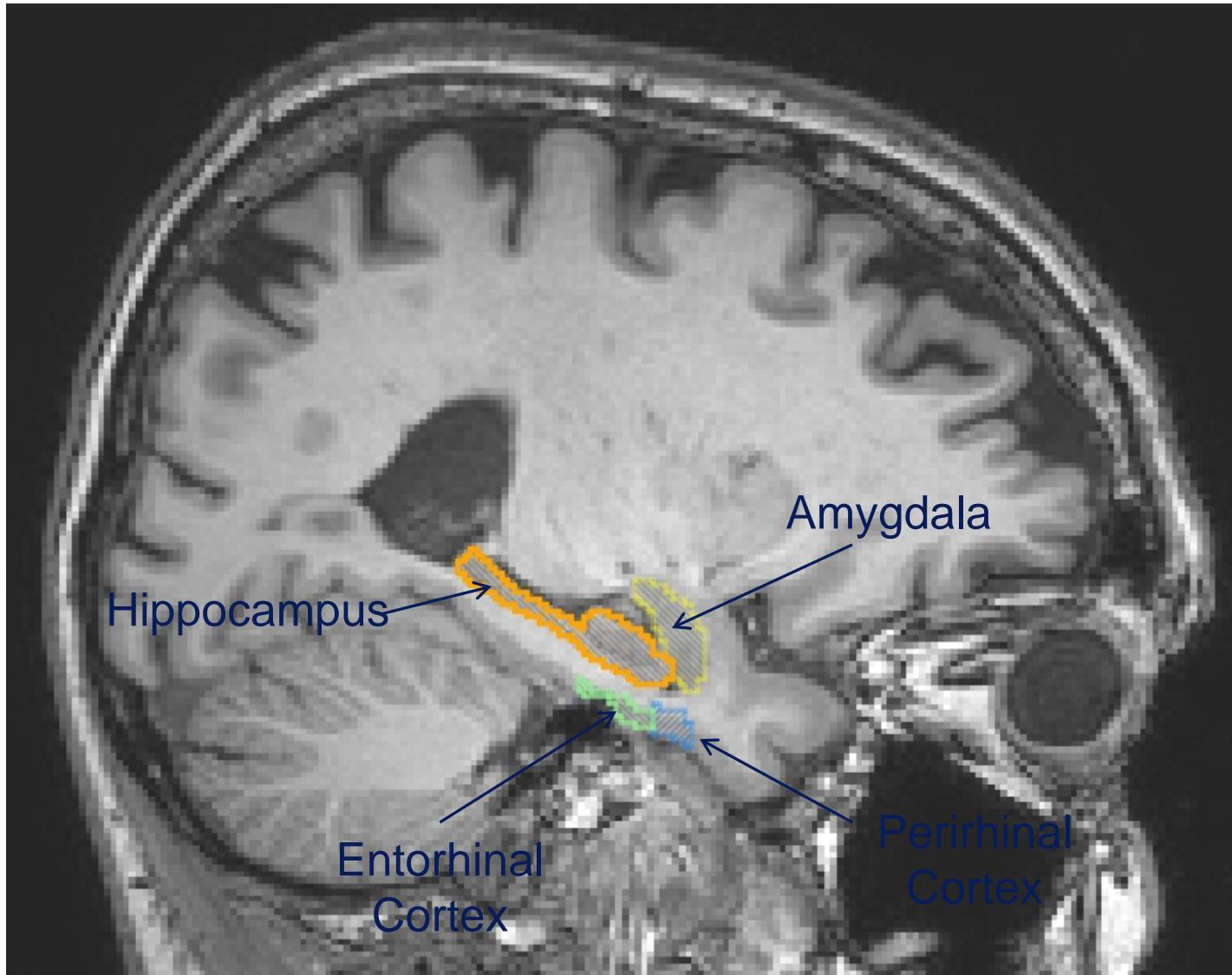
# Represent Matrix as column vector rewriting minimization

- $X_i = \begin{pmatrix} x_{1i} & x_{2i} & 0 & 0 \\ 0 & 0 & x_{1i} & x_{2i} \end{pmatrix}$        $a = \begin{pmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \end{pmatrix}$

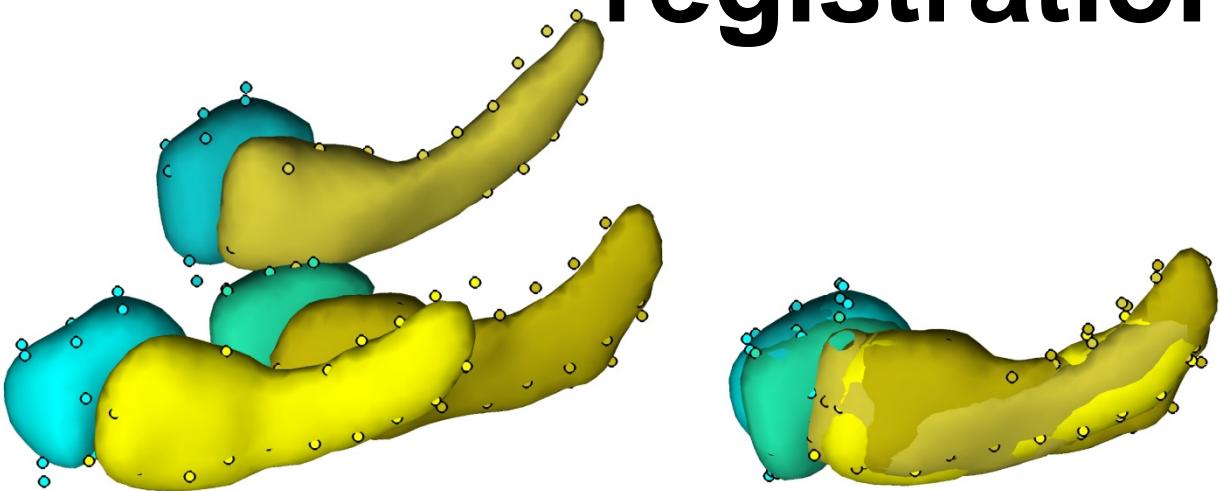
$$\min_{a \in \mathbb{R}^4} \sum_i |y_i - X_i a|^2$$

Prove:

$$\hat{a} = \underbrace{\left( \sum_i X_i^T X_i \right)^{-1}}_{4 \times 4 \text{ matrix}} \underbrace{\left( \sum_i X_i^T y_i \right)}_{4 \times 1 \text{ column}}$$



# Three amygdalas and hippocampi – before and after registration



Before  
registration

After rigid registration  
(rotation, translation, scale)<sup>1</sup>

# **Homework:**

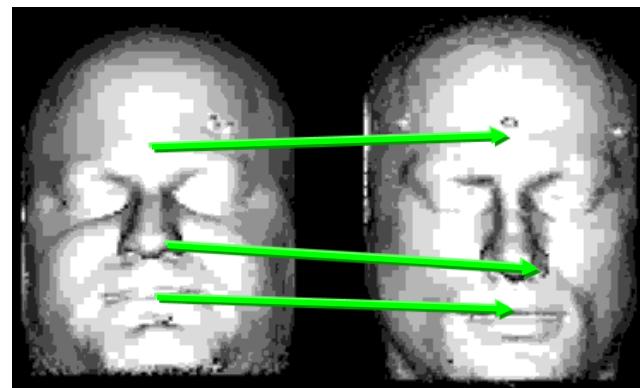
# **Translation and Matrix**

# **Group Registration**

# Homework

## SCALE

## Registration:2D



- Given landmark B.C.'s  $(x_1, y_1), (x_2, y_2) \dots, (x_n, y_n)$

- Scale 1 dimension

$$y_i = \begin{pmatrix} y_{1i} \\ y_{2i} \end{pmatrix} = s \begin{pmatrix} x_{1i} \\ x_{2i} \end{pmatrix} = sx_i$$

- Problem: Estimate Scale

$$\min_{s \in \mathbb{R}^1} \sum_i \frac{|y_i - sx_i|^2}{|b|^2 = b \cdot b = b^T b = \sum_i b_i b_i}$$

Computational Anatomy: Miller, Rathnather, Tward27

# **Homework cont'd:**

**Manual Adjustment of  
Anatomical Coordinate  
Systems  
Tool Matlab  
Daniel**

# Homework Cont'd

Homework

## 1 Examining the observer equation

In this homework you will calculate a transformation that matches one set of landmarks to another, and apply it to transform an image according to the nonlinear observer equation.

The assignment should be performed in matlab. Please hand in any code you write, in addition to anything else the problems ask for.

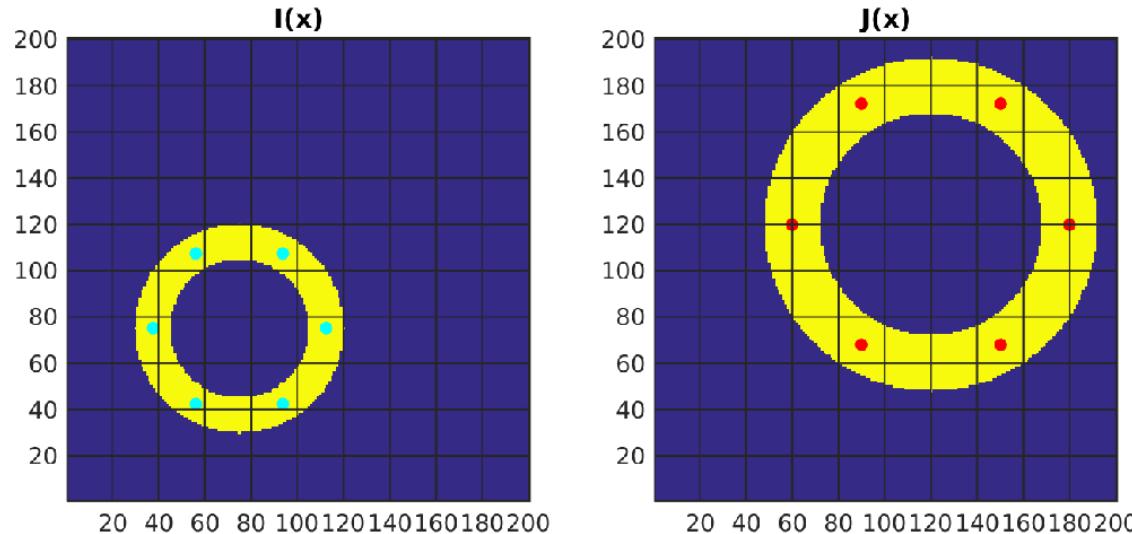


Figure 1: Images of two annuli with 6 corresponding labelled landmarks.

## 1.1 Generate the image $I(x)$

Use this code to generate your image of an annulus, corresponding to the left side of Fig. 1, in matlab.

# Homework cont'd

```
% The location of each pixel  
nX = 200; % number of columns  
nY = 200; % number of rows  
xj = 1 : nX; % x location of each column  
yi = 1 : nY; % y location of each row  
[xij,yij] = meshgrid(xj,yi); % x,y location of each pixel  
  
% define an image of an annulus  
cx = 75; % x component of center  
cy = 75; % y component of center  
r1 = 30; % inner radius in pixels  
r2 = 45; % outer radius in pixels  
% define the image using binary operations  
I = ((xij - cx).^2 + (yij - cy).^2 <= r2^2) - ((xij - cx).^2 + (yij - cy).^2 < r1^2);
```

Use this code or something similar to display your image.

```
% display it  
figure;  
imagesc(I);  
axis image;  
title('I(x)');  
set(gca,'ydir','normal'); % put origin at bottom left
```

## 1.2 Record landmarks

Estimate the position of the 6 landmarks on  $L$ , and store them in the (2 rows by 6 columns) variable  $\mathbf{X}$ . Estimate the position of the 6 corresponding landmarks on  $J$ , and store them in the (2 rows by 6 columns) variable  $\mathbf{Y}$ .

# Homework cont'd

Let the rightmost landmark be the first in your list, and proceed adding the others counterclockwise.

Write down the values you use and turn them in with your assignment.

## 1.3 Find scale factor

Find the scale factor relating one to the other finding the value of  $s$  that minimizes the error

$$E = \sum_{i=1}^6 \sum_{j=1}^2 |sX_j(i) - Y_j(i)|^2 \quad (1)$$

You should find the general equation for  $s$  analytically by taking the derivative and setting to zero. You should compute the value for this problem on the computer.

Write down the value you calculate and turn it in with your assignment.

## 1.4 Transform the landmarks $\mathbf{X}$

Apply your scale factor to the landmarks in  $\mathbf{X}$ , call this variable  $\mathbf{sX}$ . Use a scatter plot to display the landmarks  $\mathbf{X}$  (cyan, 'c'), the landmarks  $\mathbf{sX}$  (blue, 'b'), and the landmarks  $\mathbf{Y}$  (red, 'r').

Comment on the proximity of  $\mathbf{sX}$  and  $\mathbf{Y}$ . Do they match exactly? Would you expect them to?

Write down the transformed landmarks and turn them in with your assignment.

## 1.5 Transform the image “naively” $I(x) \mapsto I(sx)$

# Homework cont'd

Transform your image in the variable  $I$  according to the incorrect equation  $I(x) \mapsto I(sx)$  using the following pseudocode. You can reuse this code for a variety of different transformations, just change the lines that say % you should implement this line to match your transformation.

```
% initialize an image of all zeros
ITransformed = zeros(size(I));
for i = 1 : nY % loop through each row
    for j = 1 : nX % loop through each column

        % we are looking for the value to assign to Isx(j,i)

        % find the position to look at in the image J
        % iLook = % you should implement this line
        % jLook = % you should implement this line

        % round them to the nearest integer
        iLookRound = round(iLook);
        jLookRound = round(jLook);

        % check if we're out of bounds,
        if iLookRound < 1 || iLookRound > nY || jLookRound < 1 || jLookRound > nX
            % if so, fill the image with the value zero
            ITransformed(j,i) = 0;
```

# Homework cont'd

```
else
    % otherwise, assign the value in our image at this point
    ITransformed(j,i) = I(jLookRound,iLookRound);
end
% don't forget to index your images by (row,column) and not (x,y) !
end
end
```

Comment on what happened to the image? Did it transform to match the image  $J$ ?  
Print out a figure showing this transformed image and turn it in with your assignment.

## 1.6 Transform the image with the observer equation $I(x) \mapsto I(s^{-1}x)$

Transform your image in the variable  $I$  according to the nonlinear observer equation  $I(x) \mapsto I(s^{-1}x)$  using the above pseudocode.

Comment on what happened to the image? Did it transform to match the image  $J$ ?  
Print out a figure showing this transformed image and turn it in with your assignment.

# High-Dimensional Vector Fields and Splines via Energy Methods

Michael I. Miller

Tilak Ratnanather

Daniel Tward

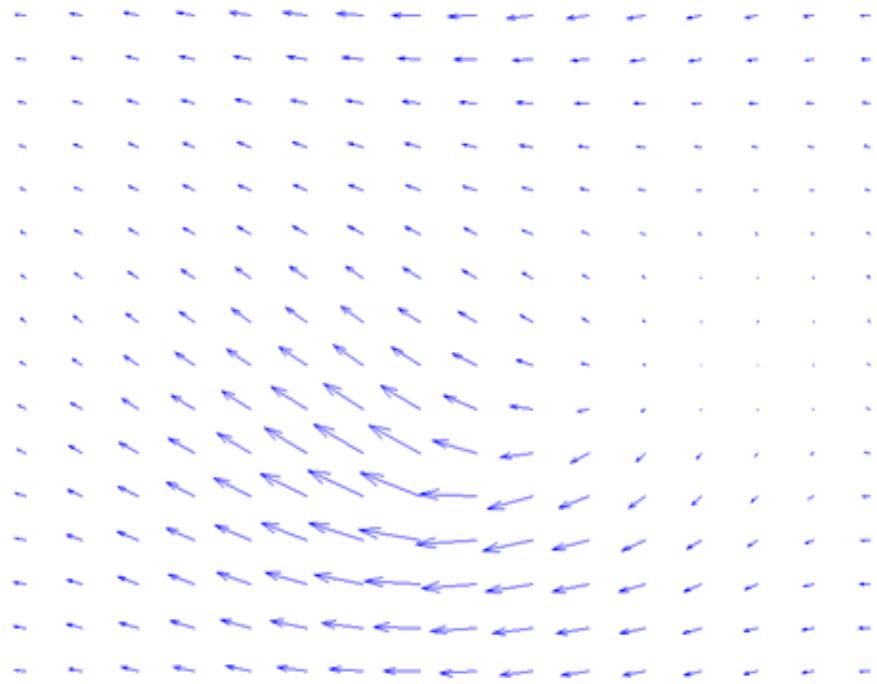


**Vector fields are high dimensional transformations; we need high dimensions for human anatomy.**

# Vector Fields

Vector Field  $x \in \mathbb{R}^2$

$$\phi(x) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \underbrace{\begin{pmatrix} v_1(x) \\ v_2(x) \end{pmatrix}}_{\text{vector field}}$$

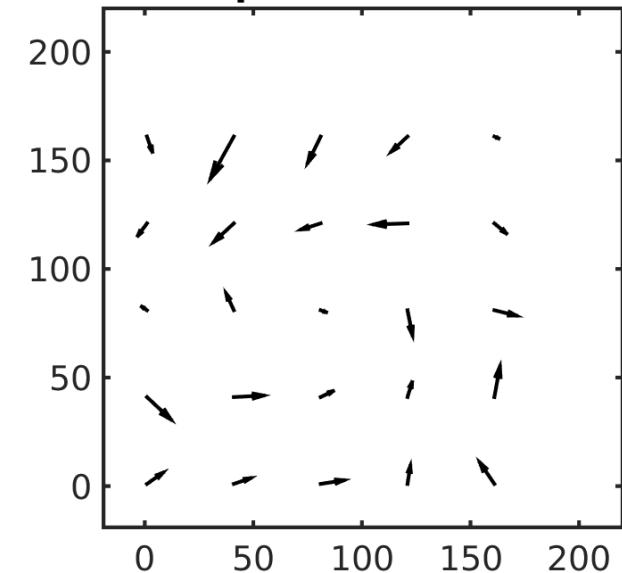


$16 \times 16$

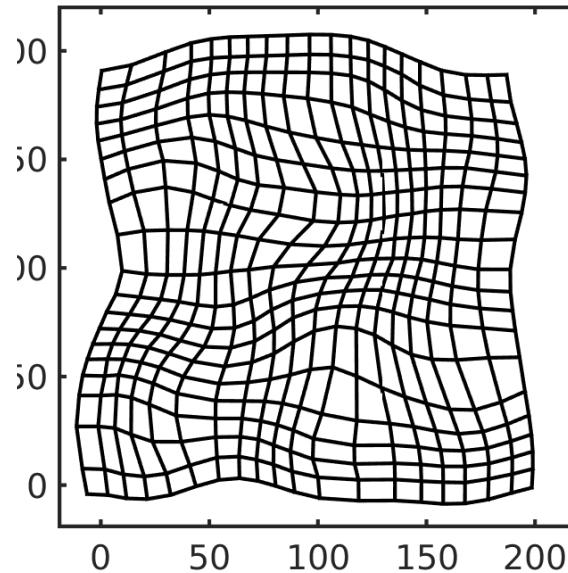
$v \in \mathbb{R}^{2 \times 256}$

High-dimensional

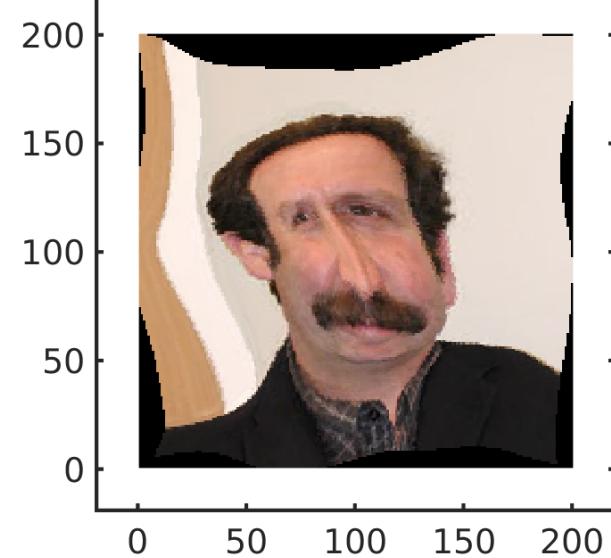
**Displacement field**



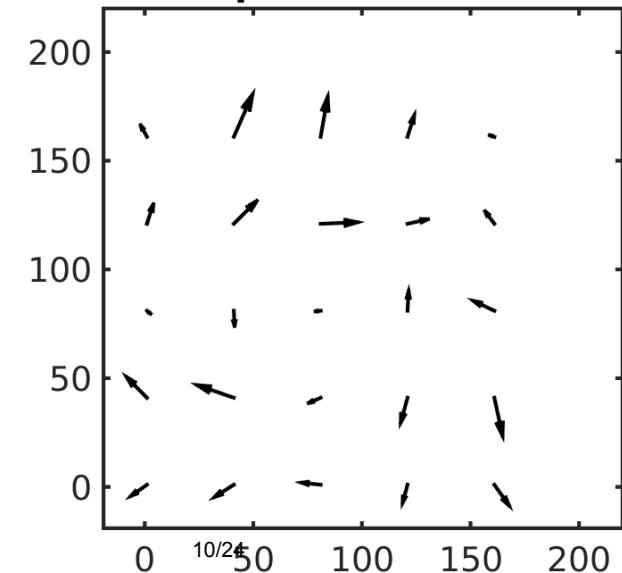
**Deformation field**



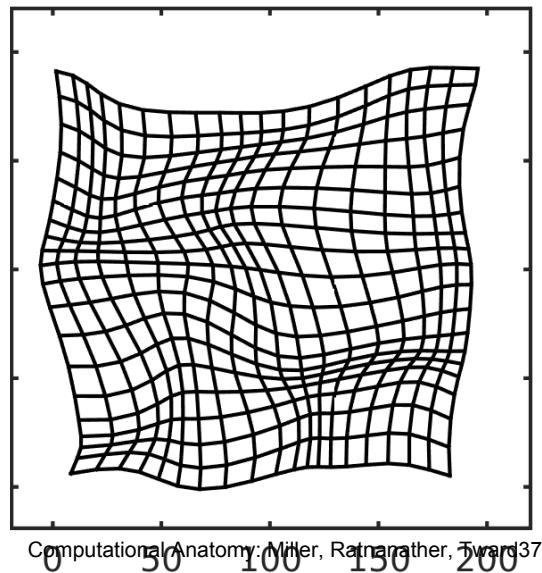
**Deformed image**



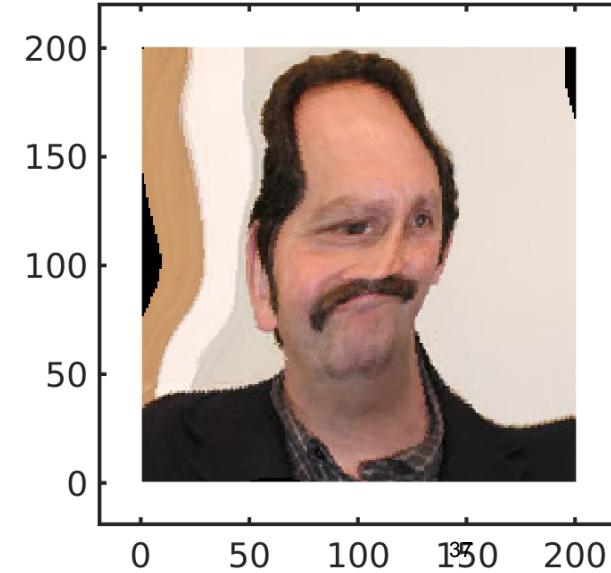
**Displacement field**



**Deformation field**



**Deformed image**



**Vector fields have too many dimensions so we use Energy methods to constrain them.**

**You are familiar with Energies from lumped, mechano-electrical models.**

# Energies of lumped mechanical or electrical systems: harmonic oscillators, without resistivity

Position  $\phi_t$ , Velocity  $v_t = \dot{\phi}_t$    Charge  $q_t$ , Current  $i_t = \dot{q}_t$

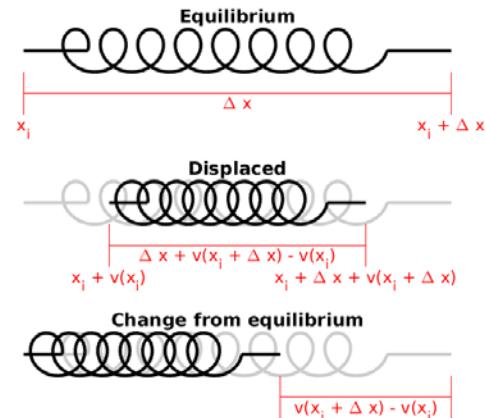
Spring-Mass    $m\ddot{\phi}_t + k\phi_t = 0$   
 $\phi_t = \Phi_{init} \cos(\sqrt{k/m} t + \theta)$

K. E. + P. E. =  $\frac{1}{2}m\dot{\phi}_t^2 + \frac{1}{2}k\phi_t^2$   
=  $k\Phi_{init}^2$

Kirchoff:    $L\ddot{q}_t + \frac{1}{C}q_t = 0$   
 $q_t = Q_{init} \cos(1/\sqrt{LC}t + \theta)$

K. E. + P. E. =  $\frac{1}{2}L\dot{q}_t^2 + \frac{1}{2C}q_t^2$   
=  $1/CQ_{init}^2$

# Energy of a Distributed Spring System



Spring Position:

$$\phi: x \mapsto x + v(x)$$

Potential Energy:

$$\begin{aligned} \lim_{\Delta x \rightarrow 0} \sum_i \frac{1}{2} \frac{k}{\Delta x} |v(x_i + \Delta x) - v(x_i)|^2 &\simeq \lim_{\Delta x \rightarrow 0} \sum_i \frac{1}{2} k |v'(x_i)|^2 \Delta x \\ &= \frac{1}{2} \int k \left| \frac{dv(x)}{dx} \right|^2 dx \\ &= \frac{1}{2} \int A v \cdot v dx + B.C.'s, \quad A = -k \frac{\partial^2}{\partial x^2} \end{aligned}$$

# 3D Energies from Continuum Mechanics

Energy:

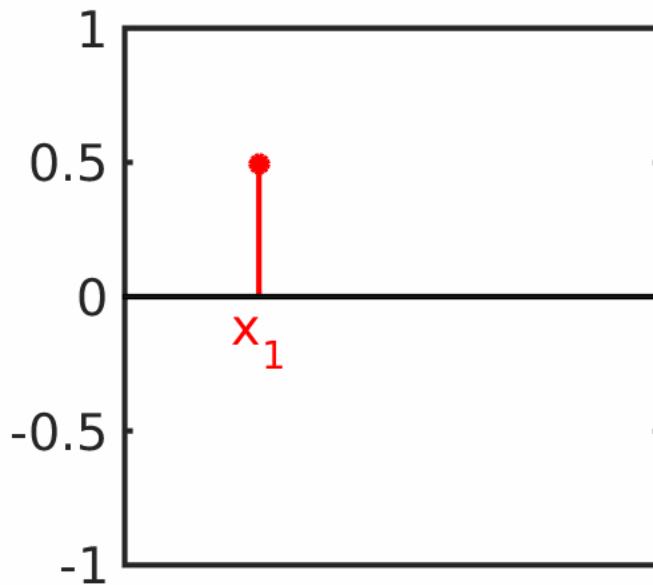
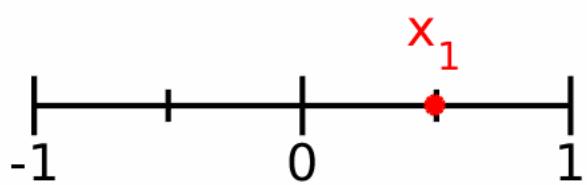
$$\frac{1}{2} \int_{\mathbb{R}^3} (\mathbf{A}\mathbf{v}) \cdot \mathbf{v} dx$$
$$\mathbf{A} = \sum_n a_n \frac{\partial^{p_n+q_n+r_n}}{\partial x_1^{p_n} \partial x_2^{q_n} \partial x_3^{r_n}}$$

- 3D Elasticity in Volumes  $x = (x_1, x_2, x_3) \in \mathbb{R}^3$

$$\int_{\mathbb{R}^3} (-\nabla^2 v(x) + v(x))^2 dx, \quad \nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2}$$

$$\int_{\mathbb{R}^3} \left( -\underbrace{c_1}_{\substack{\text{bulk} \\ \text{modulus}}} \nabla^2 v(x) + \underbrace{c_2}_{\substack{\text{shear} \\ \text{modulus}}} \nabla \nabla \cdot v(x) + v(x) \right)^2 dx$$

# Minimizing Energies via Calculus of Variations



# Maximizing or Minimizing

Scalar:

$$G(x), \quad J(\epsilon) = G(x + \epsilon h), h \in R$$
$$\frac{d}{d\epsilon} J(\epsilon) \Big|_{\epsilon=0} = 0$$

Vector:  $f = (f(x_1), \dots, f(x_n)), h = (h(x_1), \dots, h(x_n))$

$$J(\epsilon) = G(f + \epsilon h), \quad \frac{d}{d\epsilon} J(\epsilon) \Big|_{\epsilon=0} = 0$$

Function:  $f = f(x), x \in X, h = h(x), x \in X$

$$J(\epsilon) = G(f + \epsilon h), \quad \frac{d}{d\epsilon} J(\epsilon) \Big|_{\epsilon=0} = 0$$

*Proof:*  $J(\epsilon) = J(\epsilon) \Big|_{\epsilon=0} + \epsilon \underbrace{\frac{d}{d\epsilon} J(\epsilon) \Big|_{\epsilon=0}}_{must=0} + o(\epsilon)$

# Quadratic Form $G(\cdot) = |\cdot|^2$

Scalar:

$$J(\epsilon) = G(x + \epsilon h), h \in R$$

$$\frac{d}{d\epsilon} J(\epsilon) \Big|_{\epsilon=0} = 2xh = 0$$

Vector:

$$f = (f(x_1), \dots, f(x_n)), h = (h(x_1), \dots, h(x_n))$$

$$\frac{d}{d\epsilon} J(\epsilon) \Big|_{\epsilon=0} = \frac{d}{d\epsilon} \sum_i (f(x_i) + \epsilon h(x_i))^2 \Big|_{\epsilon=0} = 2 \sum_i f(x_i)h(x_i) = 0$$

Function:

$$f = f(x), x \in X, h = h(x), x \in X$$

$$\frac{d}{d\epsilon} J(\epsilon) \Big|_{\epsilon=0} = \frac{d}{d\epsilon} \int_X (f(x) + \epsilon h(x))^2 dx \Big|_{\epsilon=0}$$

$$= 2 \int_X f(x)h(x) dx = 0$$

# Minimizing Quadratic Energy

Energy:  $J = \frac{1}{2} \int_X A f \cdot f dx = \frac{1}{2} \int_X f \cdot A f dx$

Perturbed energy:  $J(\epsilon) = \frac{1}{2} \int_X A f^\epsilon \cdot f^\epsilon dx$   
 $f^\epsilon(x) = f(x) + \epsilon h(x)$

Minimizing Directional Derivative Condition:

$$\frac{d}{d\epsilon} J(\epsilon)|_{\epsilon=0} = \int_X A f \cdot h dx = 0$$

*Proof:*  $J(\epsilon) = J(\epsilon)|_{\epsilon=0} + \underbrace{\epsilon \frac{d}{d\epsilon} J(\epsilon)|_{\epsilon=0}}_{must=0} + o(\epsilon)$

*Prove Minimizer Condition:*  $\frac{d}{d\epsilon} J(\epsilon)|_{\epsilon=0} = \int_X Af \cdot h \, dx$

$$f^\epsilon = f + \epsilon h,$$

$$\frac{d}{d\epsilon} J(\epsilon) \Big|_{\epsilon=0} = \frac{d}{d\epsilon} \left( \frac{1}{2} \int_X A(f + \epsilon h) \cdot (f + \epsilon h) dx \Big|_{\epsilon=0} \right)$$

$$= \frac{1}{2} \int_X Ah \cdot f dx + \frac{1}{2} \int_X Af \cdot h dx$$

$$= \frac{1}{2} \int_X h \cdot Af dx + \frac{1}{2} \int_X Af \cdot h dx = \int_X Af \cdot h dx$$

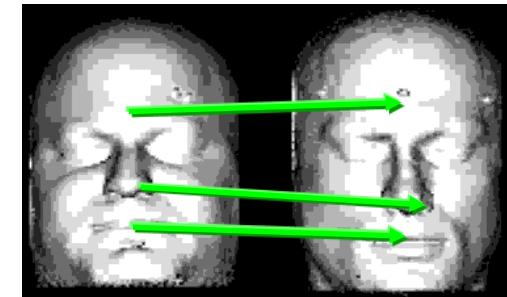
# Spline constrained vector fields as minimum energy solutions.

# Unconstrained vector fields have too many dimensions

- Problem: vector field  $\phi(x) = x + v(x), x \in \mathbb{R}^2$

B.C.s:  $y_i = x_i + v(x_i), i = 1, \dots, n$

$$\begin{pmatrix} y_{1i} \\ y_{2i} \end{pmatrix} = \begin{pmatrix} x_{1i} \\ x_{2i} \end{pmatrix} + \begin{pmatrix} v_1(x_i) \\ v_2(x_i) \end{pmatrix}$$



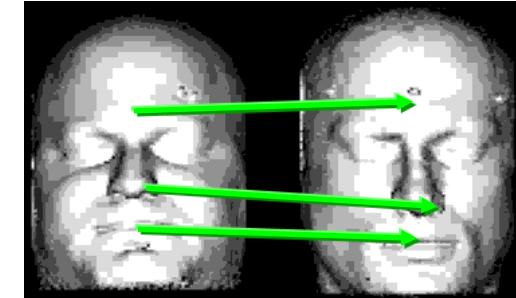
$$\min_{v(x), x \in \mathbb{R}^2} \sum_i |y_i - (x_i + v(x_i))|^2$$

- Solution:

$$v(x_i) = y_i - x_i$$

$\hat{v}(x) = \text{anything, for all } x \neq x_i, i = 1, \dots, n$

# Smooth vector field as variational problem.



- Vector Field:  $\phi(x) = x + v(x)$

$$\min_v \frac{1}{2} \int_{\mathbb{R}^2} (Av) \cdot v dx \text{ s.t. } y_i = x_i + v(x_i), i = 1, \dots, n$$

$$A = \sum_l a_l \frac{\partial^{p_l+q_l}}{\partial x_1^{p_l} \partial x_2^{q_l}} \quad \text{Green's kernel } Ak = \delta$$

- Solution: A Superposition of Spline Green's Kernels

$$v(x) = \sum_i \underbrace{k(x - x_i)}_{\substack{\text{Green's kernel} \\ \text{spline}}} p_i$$

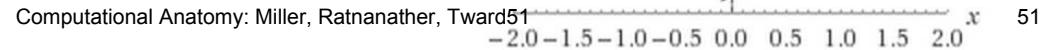
In Physics, the inverse of A is called the Green's function or the Green's kernel : it “splines” the landmark boundary conditions.

$$Ak = \delta$$

- Signals-Systems Differential equation:  $x \in \mathbb{R}$

$$A = \left( -\frac{d^2}{dx^2} + id \right)^2$$

$$k(x) = (1 + |x|) e^{-|x|}$$



# Splines as Minimum Energy (cont'd)

B.C.S:  $y_i = x_i + v(x_i), i = 1, 2, \dots$

$$\begin{aligned}v^\epsilon &= v + \epsilon h, \frac{d}{d\epsilon} J(v + \epsilon h) \Big|_{\epsilon=0} \\&= \frac{d}{d\epsilon} \left( \frac{1}{2} \int_X A(v(x) + \epsilon h(x)) \cdot (v(x) + \epsilon h(x)) dx \right. \\&\quad \left. + \sum_i \int_X p_i \cdot (y_i - x_i - v(x) - \epsilon h(x)) \delta(x - x_i) dx \right) \Big|_{\epsilon=0} \\&= \underbrace{\int_X (A v(x) - \sum_i \delta(x - x_i) p_i) \cdot h(x) dx}_{= 0}\end{aligned}$$

$$Av(x) = \sum_i \delta(x - x_i) p_i$$

# Green's Function

3D partial differential equation:  $(x_1, x_2, x_3) \in \mathbb{R}^3$

Laplacian:  $\nabla^2 = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2}$

Continuity:

- $A = (-\nabla^2 + id)^2, k(x) = e^{-|x|}$

1-Derivative:

- $A = (-\nabla^2 + id)^3, k(x) = 2(1 + |x|)e^{-|x|}$

2-Derivatives:

- $A = (-\nabla^2 + id)^4, k(x) = 4(3 + 3|x| + |x|^2)(e^{-|x|})$

# Vector Fields via Splines $x \in \mathbb{R}^2$

**Splines smooth:**  $\phi(x) = x + v(x)$

$$\begin{aligned}v_1(x) &= \sum_i k(x - x_i) p_{1i} \\v_2(x) &= \sum_i k(x - x_i) p_{2i}\end{aligned}$$

**Matrix Notation:**

$$\underbrace{\begin{pmatrix} v_1(x) \\ v_2(x) \end{pmatrix}}_{\substack{2 \times 1 \\ vector}} = \sum_i k(x - x_i) \underbrace{\begin{pmatrix} p_{1i} \\ p_{2i} \end{pmatrix}}_{\substack{2 \times 1 \\ vector}}$$

# Vector Fields via Splines $x \in \mathbb{R}^3$

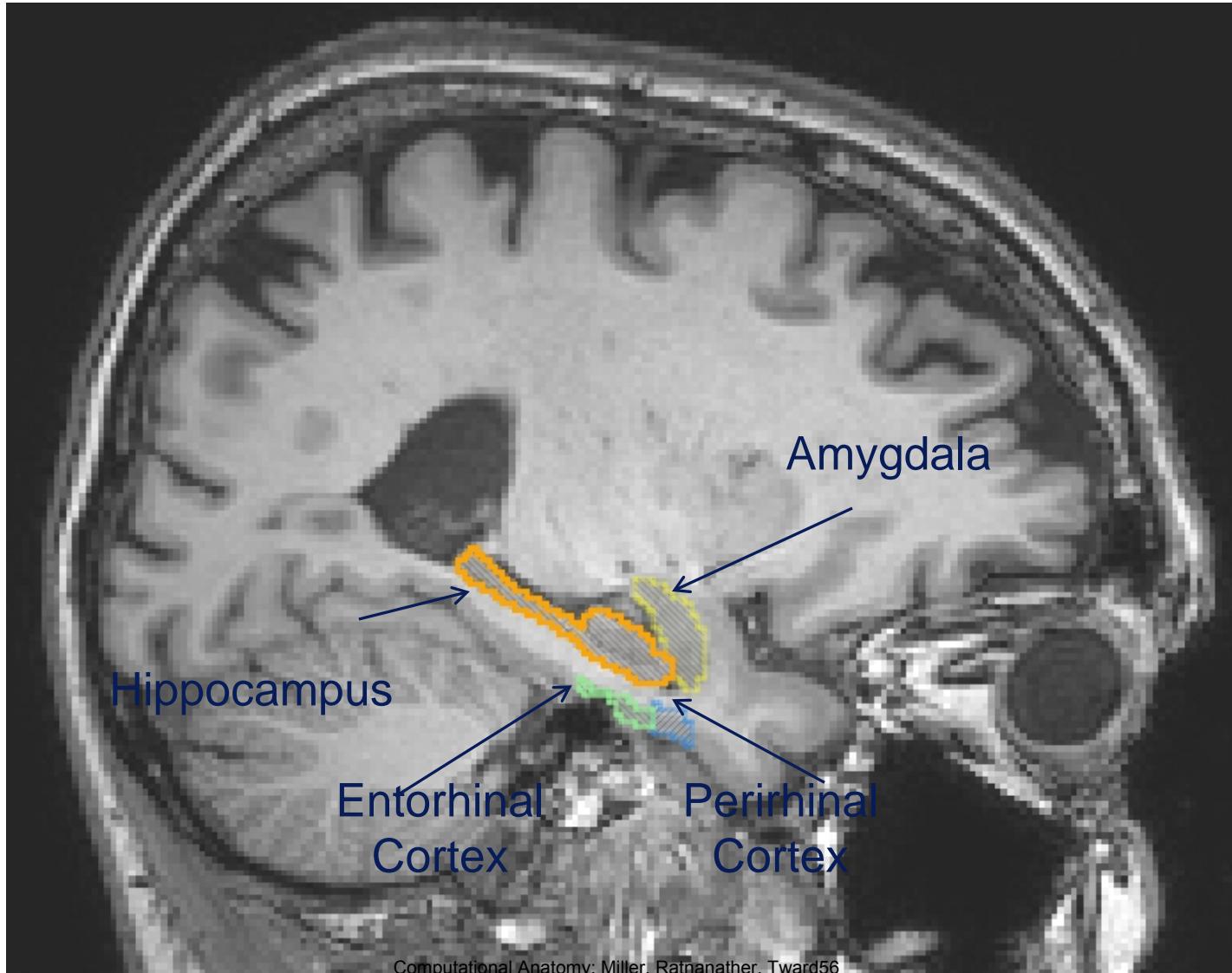
**Splines smooth:**  $\phi(x) = x + v(x)$

$$\begin{pmatrix} v_1(x) = \sum_i k(x - x_i) p_{1i} \\ v_2(x) = \sum_i k(x - x_i) p_{2i} \\ v_3(x) = \sum_i k(x - x_i) p_{3i} \end{pmatrix}$$

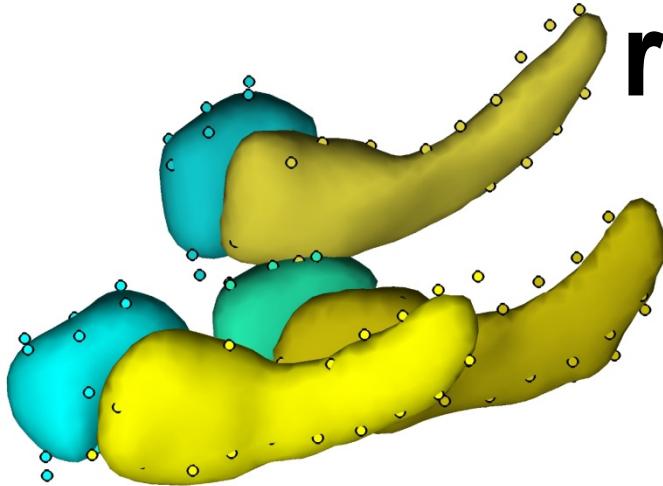
**Matrix Notation:**

$$\underbrace{\begin{pmatrix} v_1(x) \\ v_2(x) \\ v_3(x) \end{pmatrix}}_{\substack{3 \times 1 \\ vector}} = \sum_i k(x - x_i) \underbrace{\begin{pmatrix} p_{1i} \\ p_{2i} \\ p_{3i} \end{pmatrix}}_{\substack{3 \times 1 \\ vector}}$$

# Landmark Based Spines

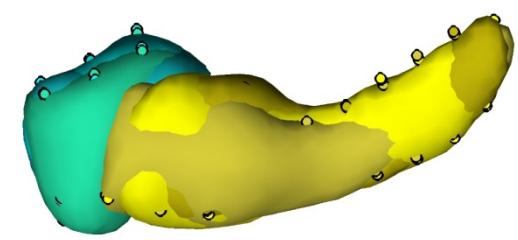


# Three amygdalas and hippocampi – before and after registration



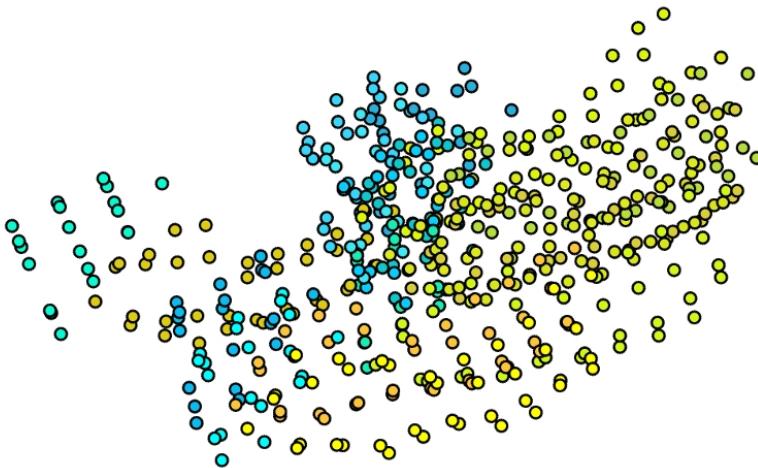
Before  
registration

After rigid registration  
(rotation, translation, scale)

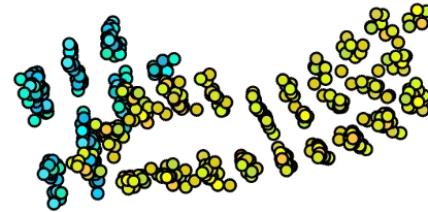


After non-rigid  
(Landmark matching actually  
via splines)

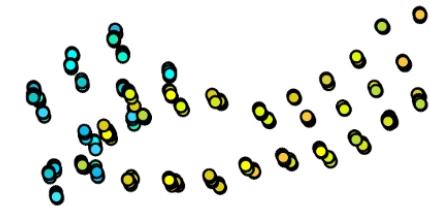
# Nine amygdalas and hippocampi – before and after registration landmarks



Before  
registration

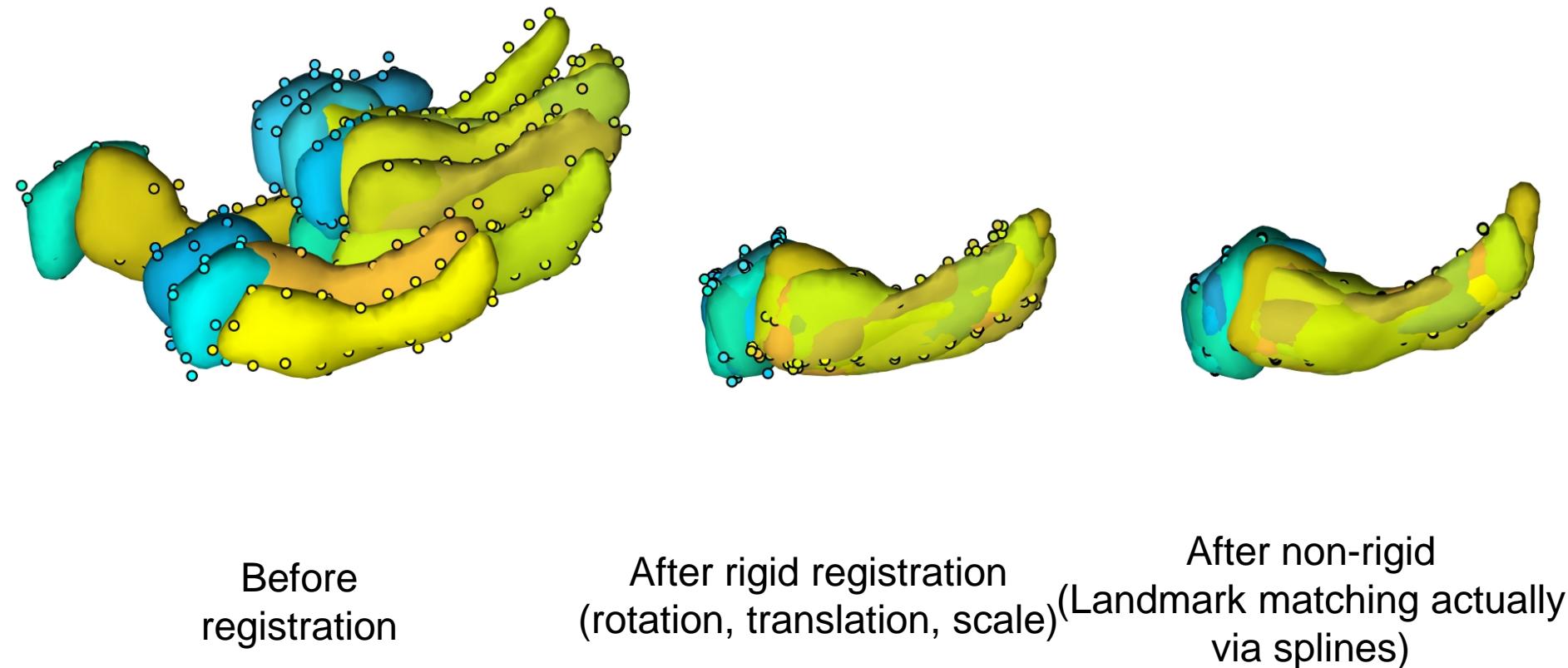


After rigid registration  
(rotation, translation, scale)



After non-rigid  
registration  
(Landmark matching actually  
via splines)

# Nine amygdalas and hippocampi – before and after registration



# Solve some Green's Kernel Examples

Daniel Tward

Circuits

Differential equations

# Differential Equation $x \in \mathbb{R}$

Given  $p(x) = -\frac{d^2}{dx^2}v(x) + v(x)$ , at rest

Green's Kernel  $k(x) = \frac{1}{2}e^{-|x|}$  for

$$A = -\frac{d^2}{dx^2} + id \text{ with } Ak(x) = \delta(x)$$

Given BC's:  $v(x_i) = V_i, i = 1, \dots, n$

Prove Solution:  $v(x) = \sum_i \frac{1}{2}e^{-|x-x_i|}p_i$

Derive matrix equation for p's:

# Gaussian Kernels:

$$v(x) = \sum_i k(x - x_i)p_i, p_i = \begin{pmatrix} p_{1i} \\ p_{2i} \end{pmatrix}$$

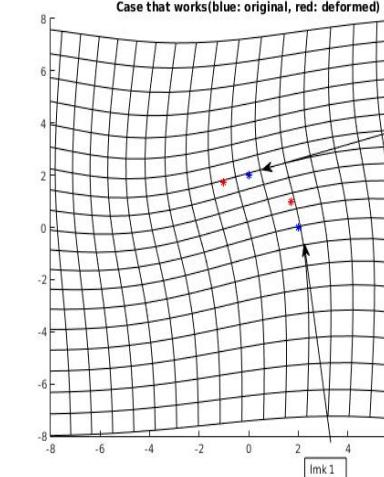
$$v_1(x) = \sum_i \frac{1}{2\pi\sigma^2} e^{\frac{-|x-x_i|^2}{2\sigma^2}} p_{1i}$$

$$v_2(x) = \sum_i \frac{1}{2\pi\sigma^2} e^{\frac{-|x-x_i|^2}{2\sigma^2}} p_{2i}$$

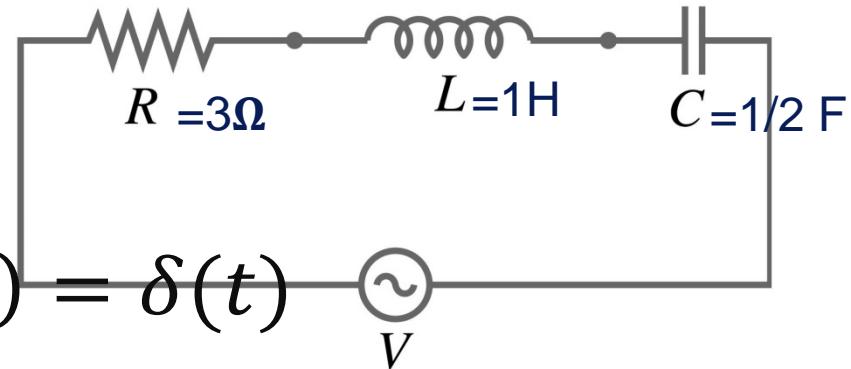
- Boundary Conditions:

$$v(x_1) = \begin{pmatrix} y_{11} - x_{11} \\ y_{21} - x_{21} \end{pmatrix}, \quad v(x_2) = \begin{pmatrix} y_{12} - x_{12} \\ y_{22} - x_{22} \end{pmatrix}$$

- Write matrix equation satisfying  $v(x_i) = \begin{pmatrix} V_{1i} \\ V_{2i} \end{pmatrix}, i = 1, 2, \dots$



# Circuit Examples: Impulse Response



$$R-C, L=0: \quad 3 \frac{d}{dt} k(t) + 2k(t) = \delta(t)$$

$$\text{Prove for } A = 3 \frac{d}{dt} + 2, \quad k(t) = \frac{1}{3} e^{-\frac{2}{3}t} u(t)$$

$$\frac{d}{dt} u(t) = \delta(t)$$

$$R-L-C: \frac{d^2}{dt^2} k(t) + 3 \frac{d}{dt} k(t) + 2k(t) = \delta(t)$$

$$\text{Prove for } A = \frac{d^2}{dt^2} + 3 \frac{d}{dt} + 2, \quad k(t) = (e^{-t} - e^{-2t})u(t)$$

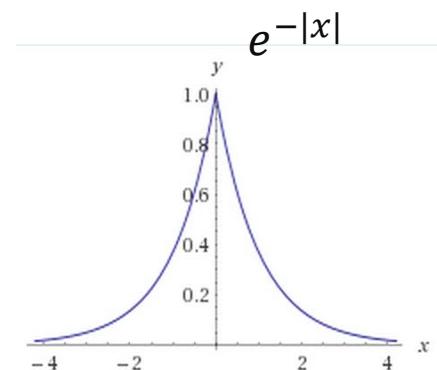
$$\text{Proof: } \dot{k}(t) = (2e^{-2t} - e^{-t})u(t) + (e^{-t} - e^{-2t})\delta(t)$$

$$\ddot{k}(t) = (e^{-t} - 4e^{-2t})u(t) + (2e^{-2t} - e^{-t})\delta(t)$$

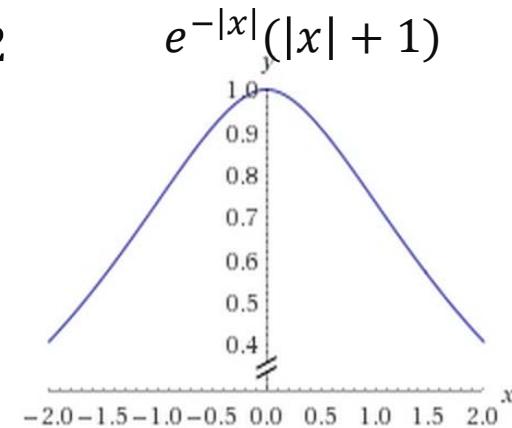
# Homework: Green's Function $x \in \mathbb{R}$

- Prove for diff. eqn.  $A = -\frac{d^2}{dx^2} + id$

$$k(x) = \frac{1}{2}e^{-|x|} \text{ with } Ak = \delta$$

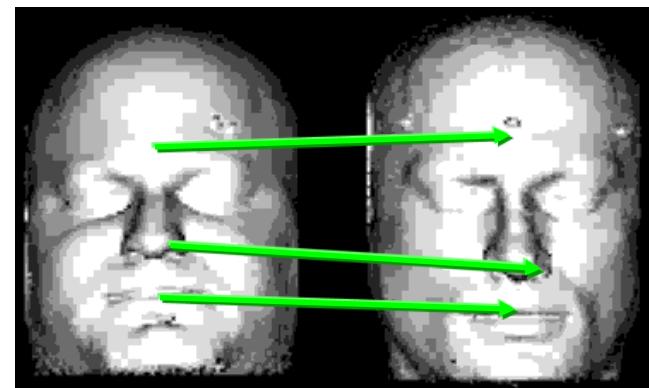


- Prove for diff. eqn.  $A = \left(-\frac{d^2}{dx^2} + id\right)^2$
- $$k(x) = e^{-|x|}(|x| + 1)$$
- with  $Ak = \delta$



# Homework:

## Using the squared-error and variation condition, show energy does not smooth the vector field without derivatives.



- Problem: Penalize Vector Field with Energy

$$\min_{v(x), x \in R^d} \int_X v \cdot v dx + \sum_i |y_i - (x_i + v(x_i))|^2$$

$$\hat{v}(x) = 0, \text{ for all } x \neq x_i, i = 1, \dots, n$$

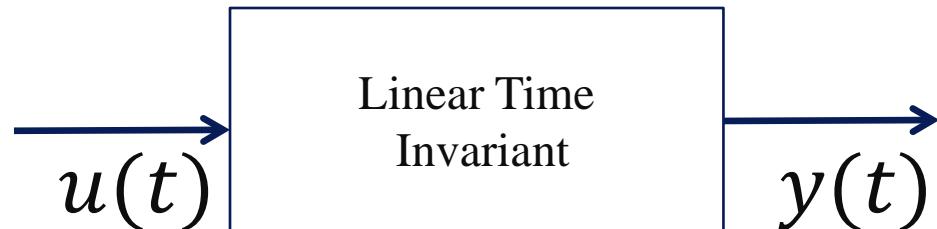
Not Smooth

# Homework: Parseval's Theorem

- Parseval's Theorem

$$\int_{-\infty}^{\infty} u(t) \cdot u(t) dt = \int_{-\infty}^{\infty} |U(f)|^2 df$$

- LTI Systems



$$u(t) = \dot{y}(t) + \alpha y(t), \text{ at rest}$$

Prove:  $\int_{-\infty}^{\infty} |U(f)|^2 df = \int_{-\infty}^{\infty} (\alpha^2 + (2\pi f)^2) |Y(f)|^2 df$

# Image Analysis and Image Understanding Via Deformable Templates

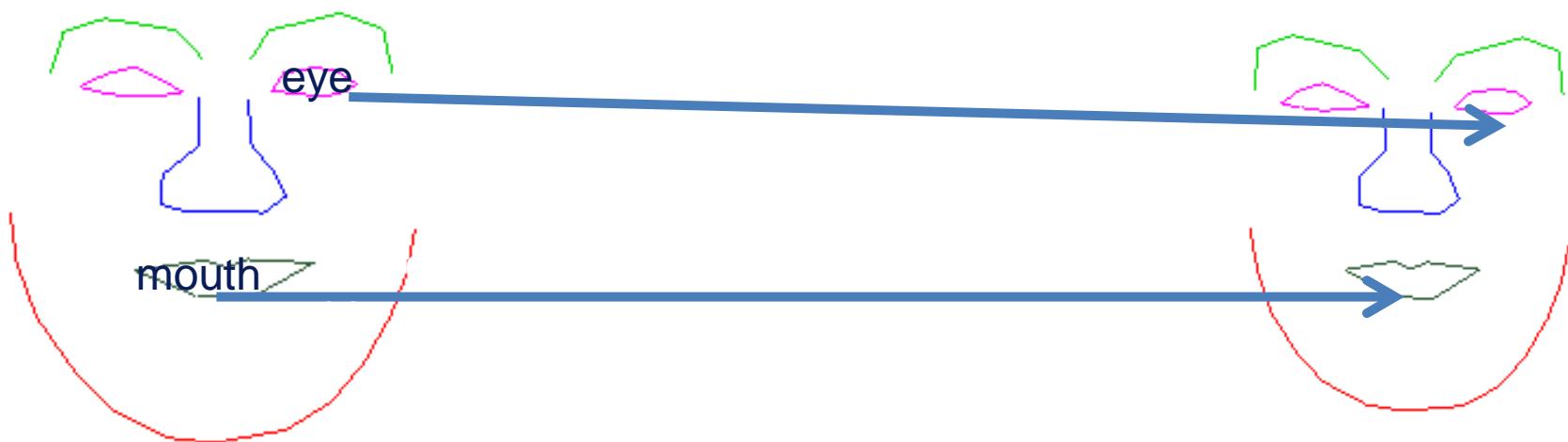
Michael I. Miller

Tilak Ratnanather

Daniel Tward

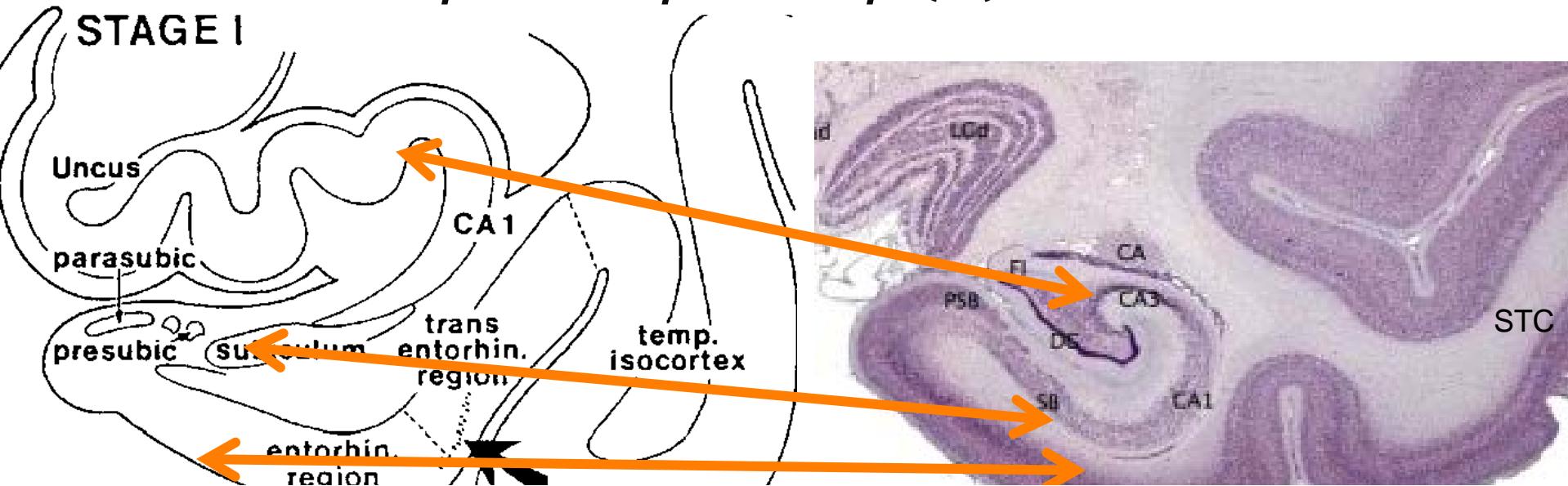


# 1-1 and Onto (Bijections) are the basis for Image Analysis



# 1-1 ONTO requires an inverse

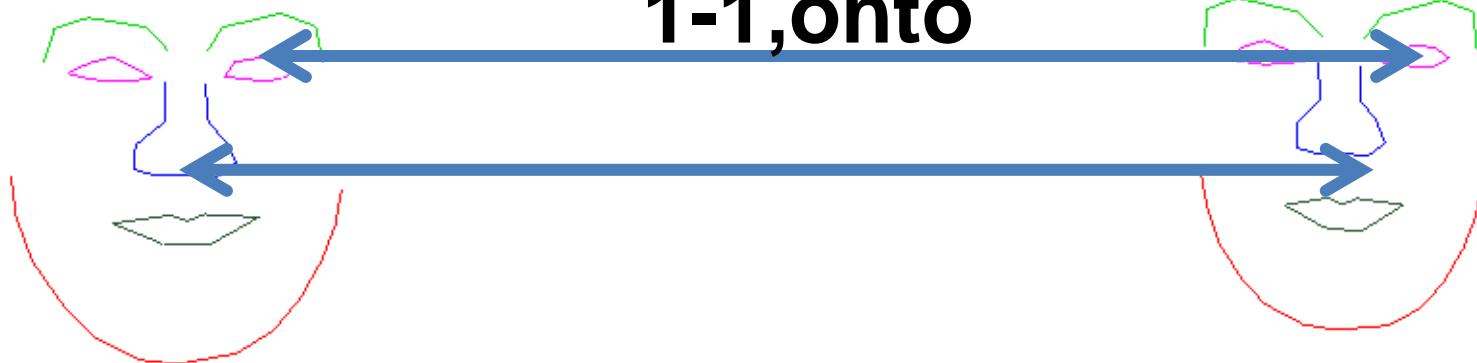
$$\exists \phi^{-1}: \phi^{-1} \circ \phi(x) = x$$



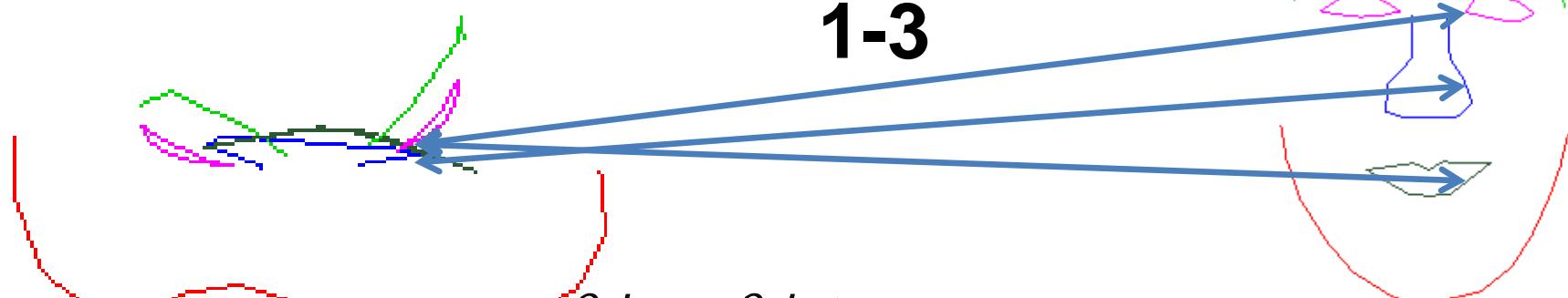
Inverse properties are determined by Jacobian of transformation

$$\text{In } \mathbb{R}^2, \text{ Jacobian } \partial_X \phi \doteq \begin{pmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_2} \\ \frac{\partial \phi_2}{\partial x_1} & \frac{\partial \phi_2}{\partial x_2} \end{pmatrix}$$

**1-1,onto**



**1-3**

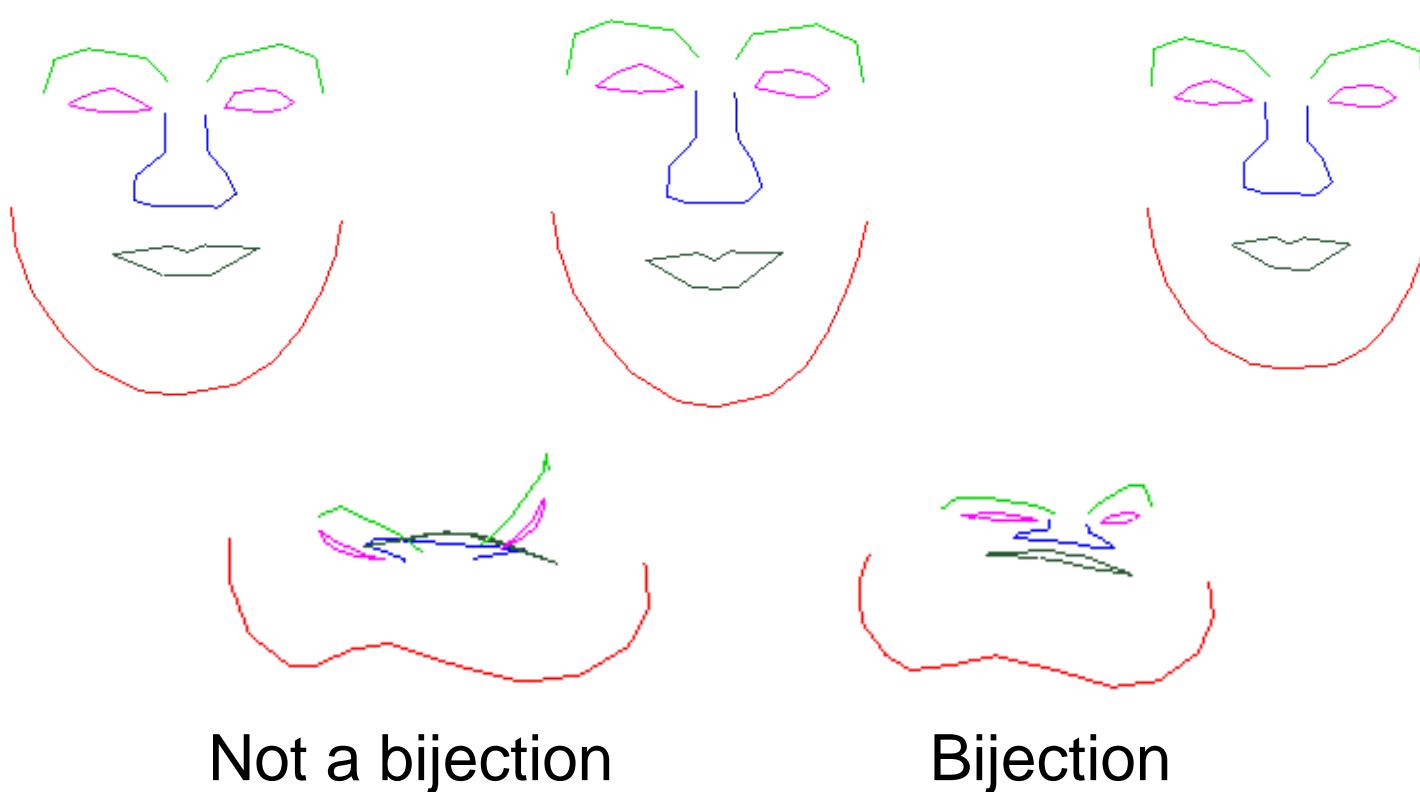


$$\det \partial_X \phi(x) = \det \begin{pmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_2} \\ \frac{\partial \phi_2}{\partial x_1} & \frac{\partial \phi_2}{\partial x_2} \end{pmatrix} = 0$$

Computational Anatomy: Miller, Ratnanather, Tward<sup>70</sup>

Neuroimage 2005  
Vaillant, Miller, Trouvé & Younes

# One of these does not have a bijective correspondence with others



**For linear matrix  
transformations,  
having a matrix  
inverse implies they  
are 1-1 and onto.**

# Matrix Groups and Inverse

Transform coordinates:

$$\phi: (x_1, x_2) \in \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$\phi(x) \doteq Ax = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Group: composition  $\circ$ , identity  $Id$ , inverse  $A^{-1}$

$$A \circ B \doteq \begin{pmatrix} \sum_i a_{1i} b_{i1} & \sum_i a_{1i} b_{i2} \\ \sum_i a_{2i} b_{i1} & \sum_i a_{2i} b_{i2} \end{pmatrix}, \quad Id \doteq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$A^{-1} \doteq \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix}$$

# Matrix inverse implies 1-1 and onto.

$\phi: \mathbb{R}^2 \rightarrow \mathbb{R}^2$  is 1-1 and onto with

$$\phi(x) \doteq Ax = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Properties:

$$\begin{aligned} \text{1-1: } \phi(x) = \phi(y) &\Rightarrow \phi(x - y) = A(x - y) = 0 \\ &\Rightarrow A^{-1} \circ A(x - y) = (x - y) = 0 \end{aligned}$$

Onto:

$$\forall y \in \mathbb{R}^2, A^{-1}y = x \text{ for some } x \in \mathbb{R}^2 \Rightarrow Ax = y$$

# For Matrix Group: Matrix=Jacobian

Transform coordinates:

$$\phi: (x_1, x_2) \in \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$\phi(x) \doteq Ax = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$\partial_x \phi(x) = \begin{pmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_2} \\ \frac{\partial \phi_2}{\partial x_1} & \frac{\partial \phi_2}{\partial x_2} \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}}_A$$

$\phi$  is 1<sup>10/24</sup>– 1, onto if  $A$  invertible (Jacobian not singular)

**For high dimensional  
vector fields,  
Jacobian must be  
positive definite  
everywhere for the  
inverse of the  
transformation  
to exist.**

# High Dimensional Vector Field

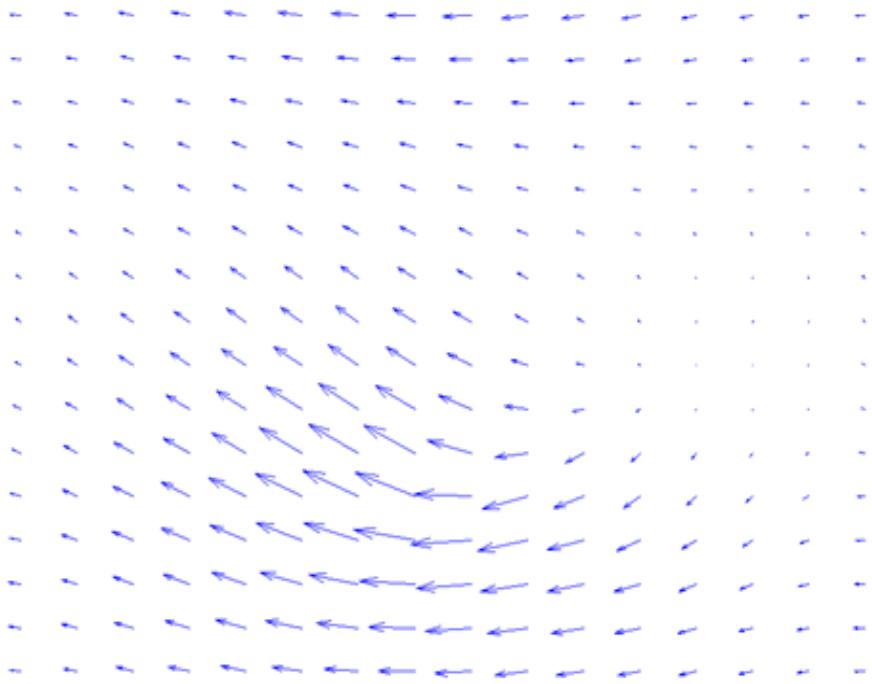
$$\phi: (x_1, x_2) \in \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$\phi(x) = (\phi_1(x), \phi_2(x))$$

Vector Field  $\in \mathbb{R}^2$

$$\phi_1(x) = x_1 + v_1(x)$$

$$\phi_2(x) = x_2 + v_2(x)$$



2x2 Jacobian Matrix

$$\partial_x \phi(x)$$

$$= \begin{pmatrix} \frac{\partial v_1(x)}{\partial x_1} + 1 & \frac{\partial v_1(x)}{\partial x_2} \\ \frac{\partial v_2(x)}{\partial x_1} & 1 + \frac{\partial v_2(x)}{\partial x_2} \end{pmatrix}$$

$16 \times 16$

$$v \in R^{2 \times 256}$$

High-dimensional

# Splines have small deformation inverse Jacobian determines the inverse

Vector field transformation:

$$\phi(x) = x + v(x)$$

Inverse (does not exist in general):

$$\phi^{-1}(x) = x - \partial_x(\phi^{-1})|_x v(x) + o(v)$$

Implicit Function Theorem

$$\partial_x(\phi^{-1})|_x = (\partial_x \phi|_{\phi^{-1}(x)})^{-1}$$

Small Deformation inverse:

small  $v \Rightarrow \partial_x(\phi^{-1}) \approx \text{identity}$

$$\phi^{-1}(x) \approx x - v(x)$$

# Neuroinformatics at the Radiomics Scale

Michael I. Miller

Tilak Ratnanather

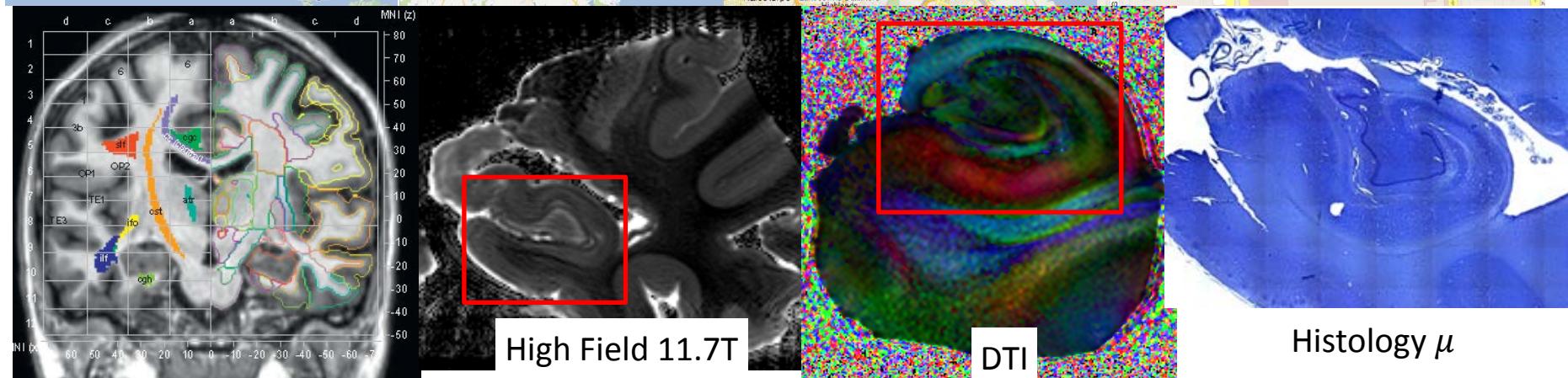
Daniel Tward



In high throughput imaging informatics, we position the 100's of structures in the brain using high-dimensional vector fields. This is a positioning engine.

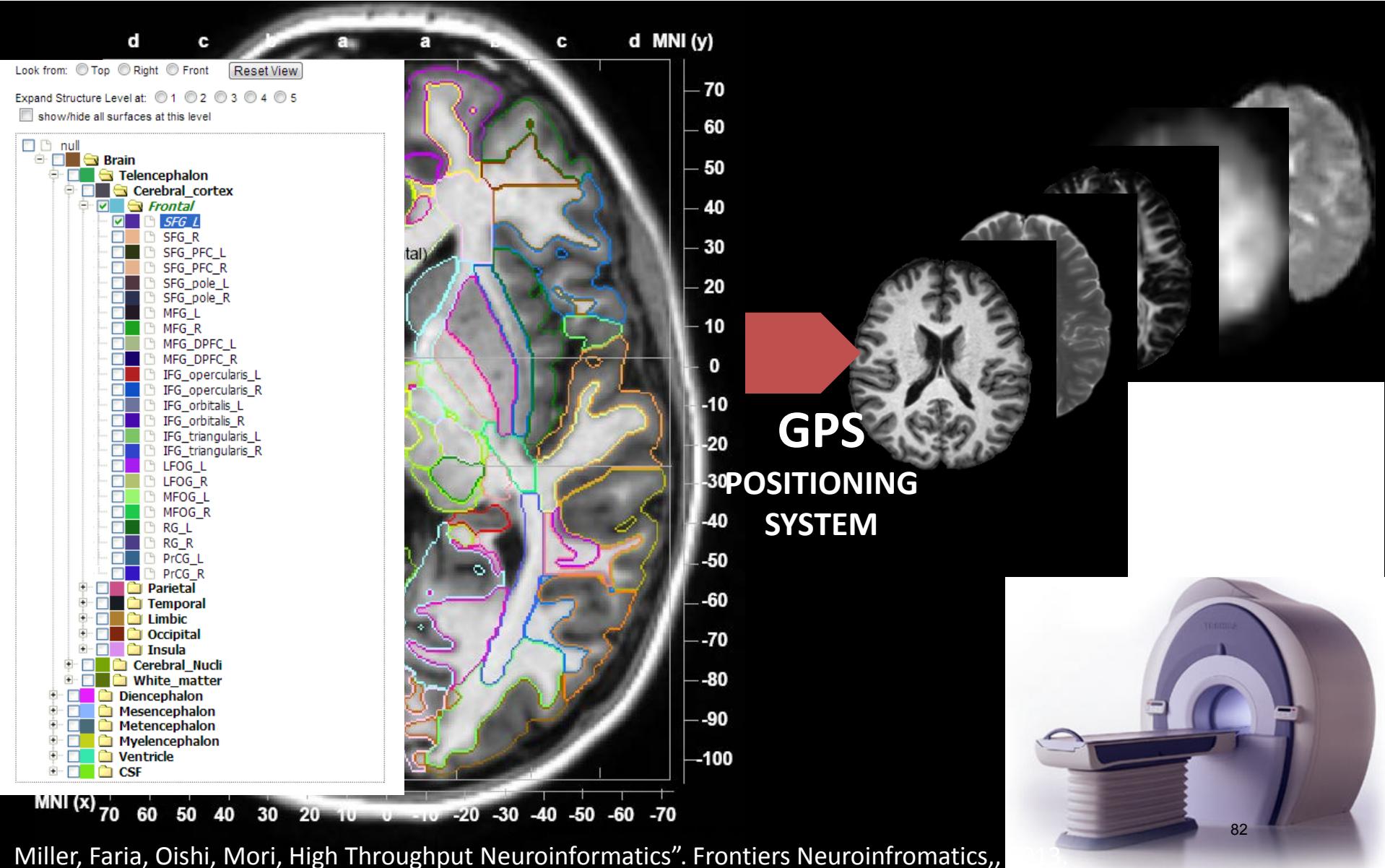
# Our GPS positions information across scales using high-dimensional vector fields.

[www.MRICloud.org](http://www.MRICloud.org)



Miller, Torte, Younes, Diffeomorphometry and Geodesic Positioning Systems in Human Anatomy, Technology, 2013.

# High Throughput Neuroinformatics



# Positioning by Registering Coordinate Systems via Small Deformation Image Matching

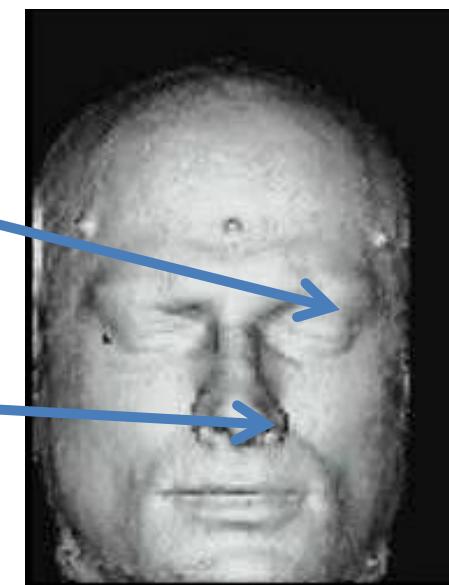
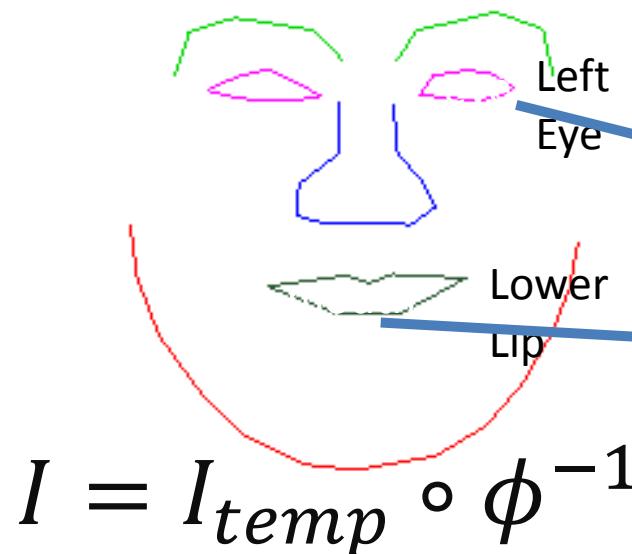
Michael I. Miller

Tilak Ratnanather

Daniel Tward



# Anatomy is an element in the orbit; the vector field positions the information.



Daniel Tward:

**Positioning is a computational code.  
Examine Small Deformation  
image matching based on splines;  
assume inverse exists.**

# Small Deformation Image Matching

$$\min_{\nu} \quad \frac{1}{2} \int_X A\nu \cdot \nu \, dx + \frac{1}{2} \int_X (I \circ \phi^{-1} - I')^2 \, dx$$

$$\nu^\epsilon = \nu + \epsilon w, \phi^\epsilon = id + (\nu + \epsilon w)$$

$$\phi^{-1} = id - \partial_X(\phi^{-1})\nu + o(\nu), \phi^{\epsilon-1} = \phi^{-1} - \epsilon(\partial_X\phi^{-1})w + o(\epsilon)$$

$$\frac{d}{d\epsilon} \frac{1}{2} \int_X A\nu^\epsilon \cdot \nu^\epsilon \, dx + \int_X (I(\phi^{-1} - \epsilon \partial_X(\phi^{-1})w) - I')^2 \, dx \Big|_{\epsilon=0}$$

$$\begin{aligned} &= \int_X A\nu \cdot w \, dx - \int_X (I \circ \phi^{-1} - I') \cdot \underbrace{\nabla I \Big|_{\phi^{-1}} \cdot \partial_X(\phi^{-1})w}_{\nabla(I \circ \phi^{-1}) \cdot w} \, dx \\ &= \int_X (A\nu - (I \circ \phi^{-1} - I') \nabla(I \circ \phi^{-1})) \cdot w \, dx \end{aligned}$$

- Necessary Condition:

$$A\nu(x) = (I \circ \phi^{-1}(x) - I'(x)) \nabla(I \circ \phi^{-1})(x), x \in X$$

# Small Deformation Image Matching

$$\min_{\nu} \quad \frac{1}{2} \int_X A\nu \cdot \nu \, dx + \frac{1}{2} \int_X |I \circ \phi^{-1} - I'|^2 \, dx$$

Necessary Condition:

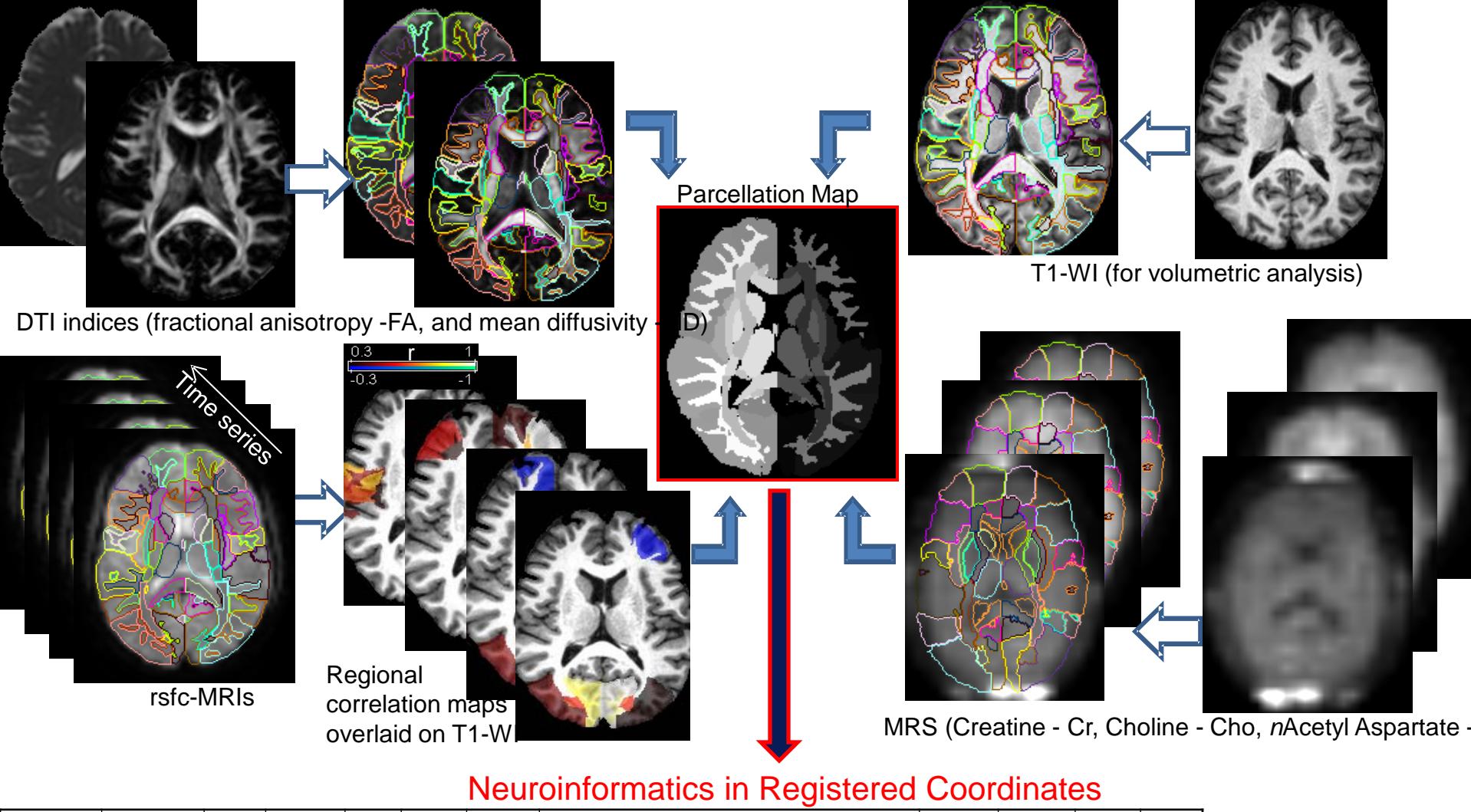
$$A\nu = (I \circ \phi^{-1} - I') \nabla (I \circ \phi^{-1})$$

Non-Linear Iteration:

$$A\nu^{new} = A\nu^{old} + \epsilon(A\nu^{old} - (I \circ \phi^{-1old} - I') \nabla (I \circ \phi^{-1old}))$$

$$\phi^{new} = id + \partial_X(\phi^{-1old}) \nu^{new},$$

$$\phi^{-1new} = id - \left( \partial_X(\phi^{new})|_{\phi^{-1old}(x)} \right)^{-1} \nu^{new}$$

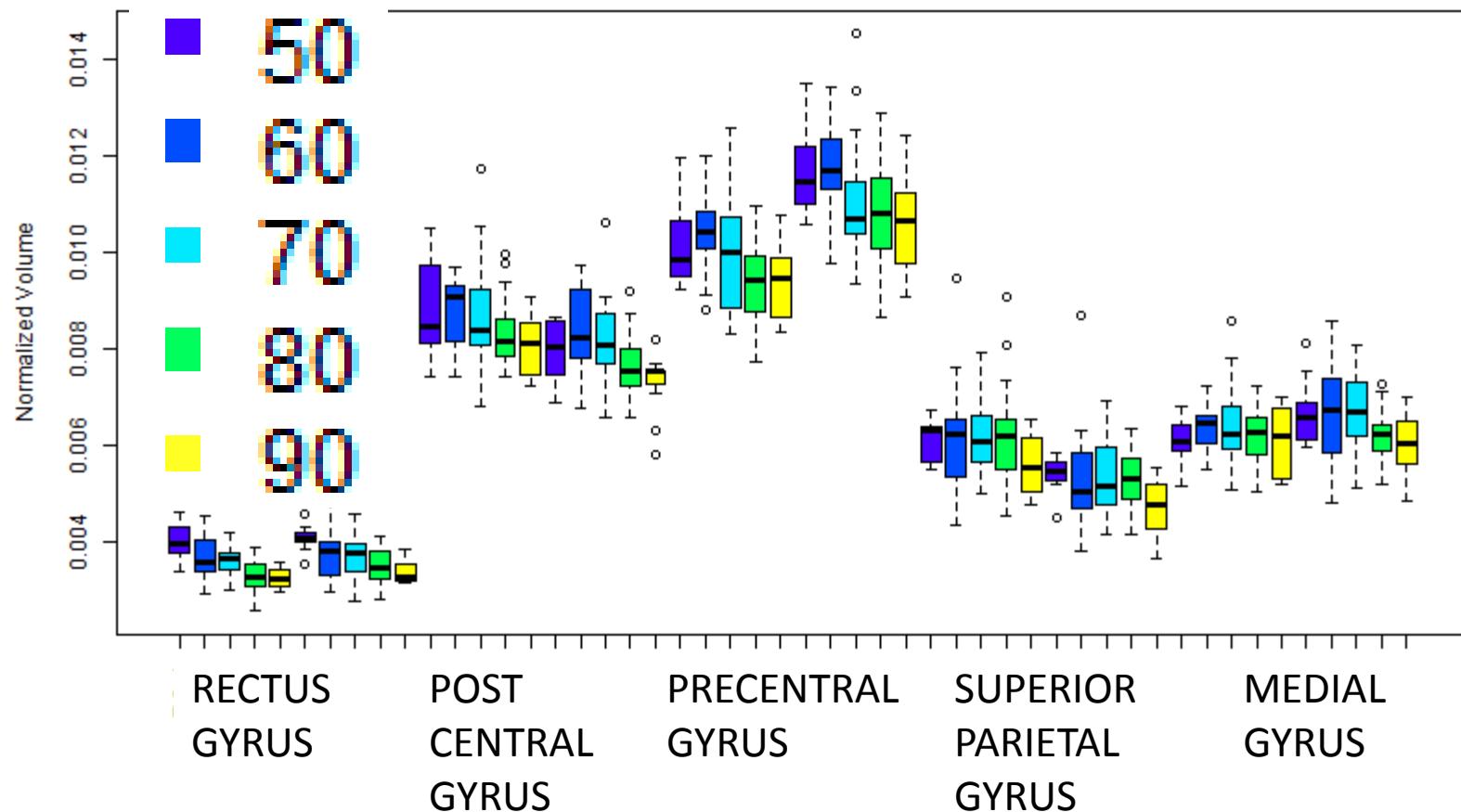


Subject n	volme	FA	MD	Cr	Cho	NAA	rsfc-fMRI correlation vector				GE	LE	PL	CC			
SFL	Subject 3	volme	FA	MD	Cr	Cho	NAA	rsfc-fMRI correlation vector				GE	LE	PL	CC		
CingL	SFL	Subject 2	volme	FA	MD	Cr	Cho	NAA	rsfc-fMRI correlation vector				GE	LE	PL	CC	
PrCuL	CingL	SFL	Subject 1	volme	FA	MD	Cr	Cho	NAA	rsfc-fMRI correlation vector				GE	LE	PL	CC
...	PrCuL	CingL	SFL	23985	0.141	10.6	5.4	1.4	13.2	1, 0.44, 0.39, ..., SFL vs. region n				0.44	0.73	2.55	0.49
region n	...	PrCuL	CingL	14551	0.14	9.3	5.2	1.3	13	0.44, 1, 0.5, ..., CingL vs. region n				0.52	0.64	2.2	0.35
...	...	PrCuL	...	7633	0.131	10.3	5.1	1.5	12.8	0.39, 0.51, ..., PrCuL vs. region n				0.49	0.68 <sub>68</sub>	2.4	0.38
region	...	...	...	...	...	...	...	...	...	...				...	...	...	...
region n	n-volume	n-FA	n-MD	n-Cr	n-Cho	n-NAA	SFL vs. n, CingL vs. n, PrCuL vs. n, ..., 1	n-GE	n-LE	n-PL	n-CC	10/24	Computational Anatomy: Miller, Rathnam, Twardosz				

Neurodevelopmental diseases have brain changes which start many years before clinical symptoms.

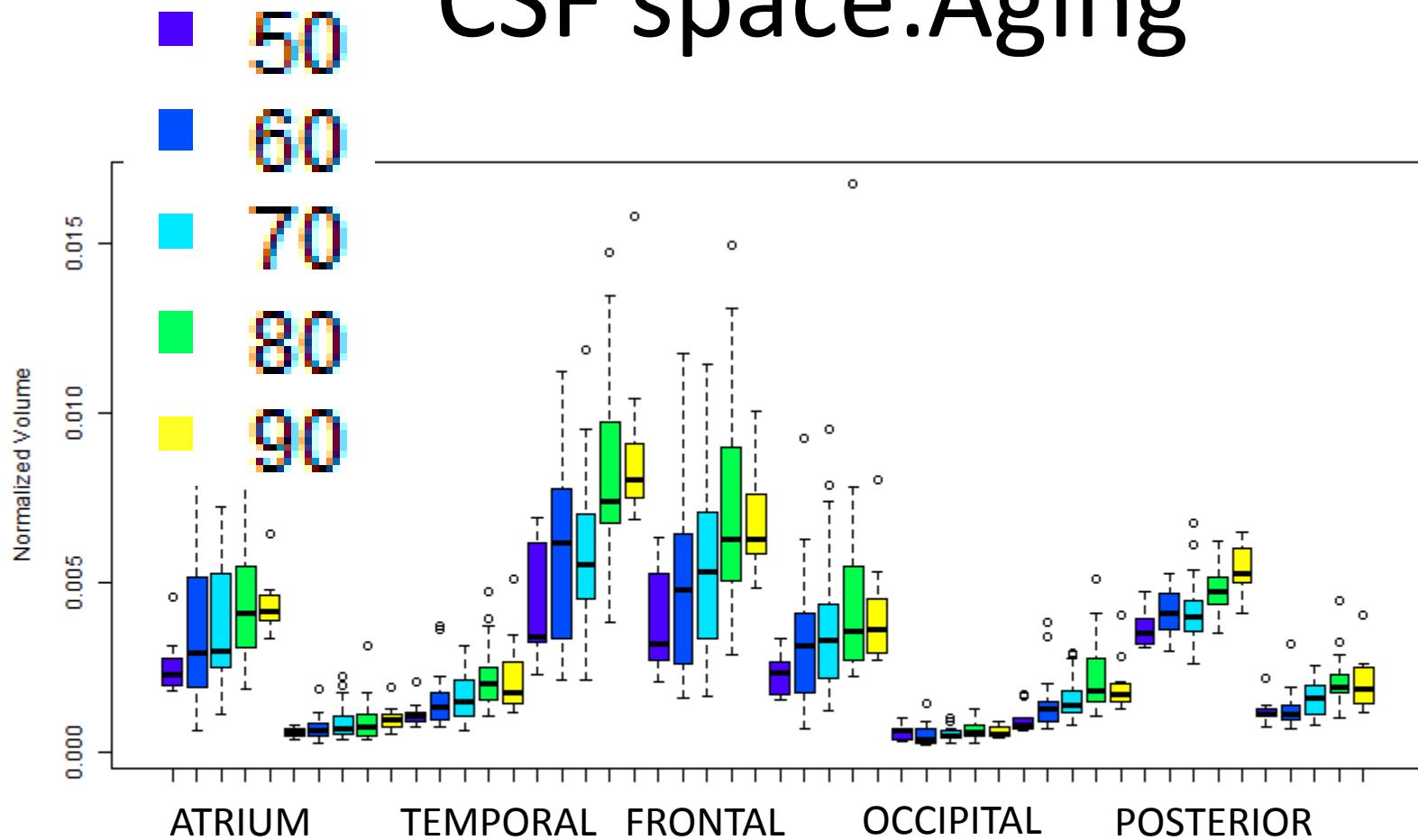
...aging does as well.

# T1 Gray matter: Aging



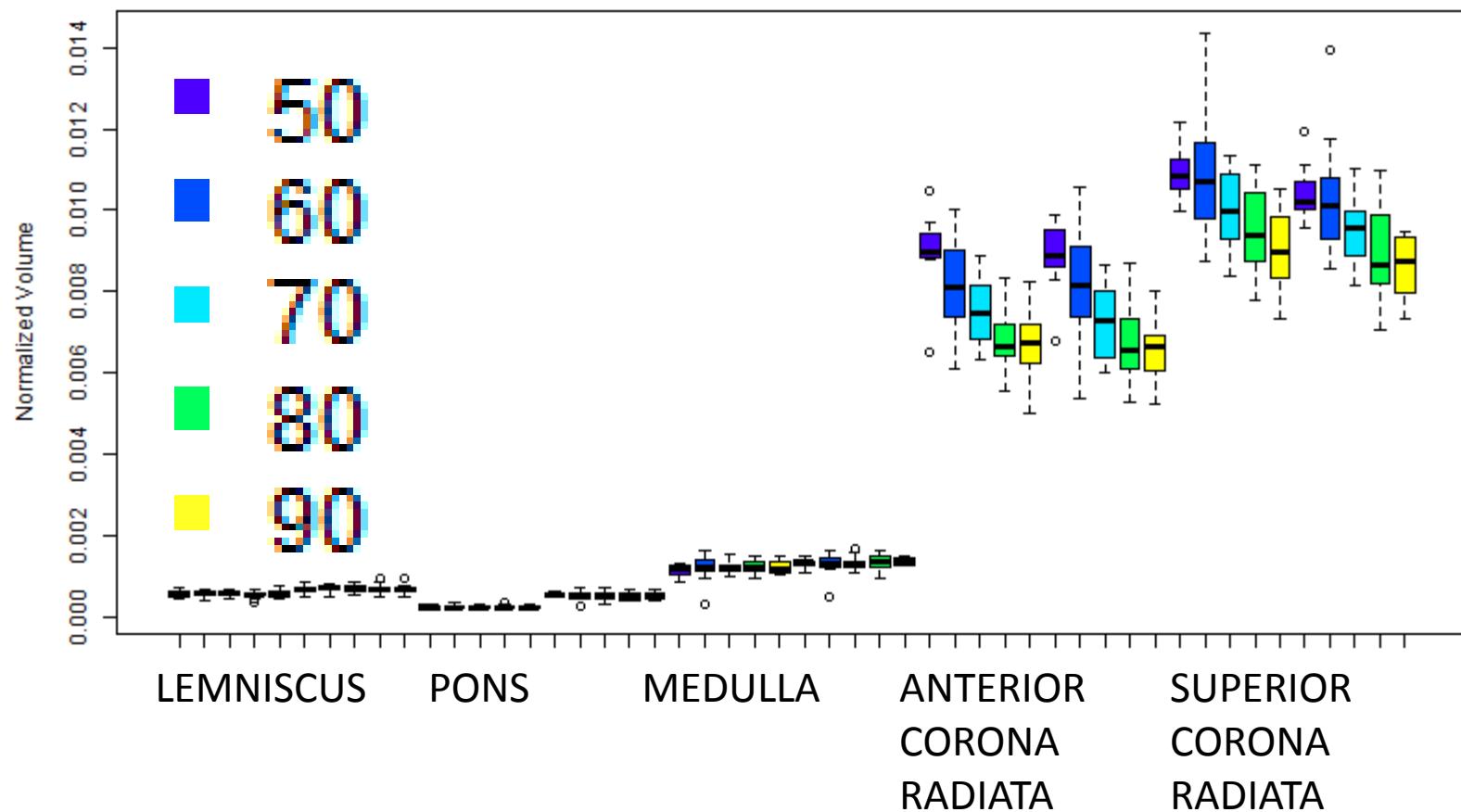
Gray matter spaces are decreasing!

# CSF space:Aging



Spinal fluid spaces are growing!

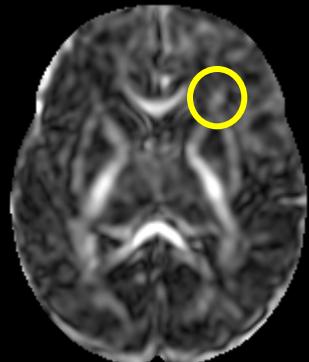
# DWI White Matter: Aging



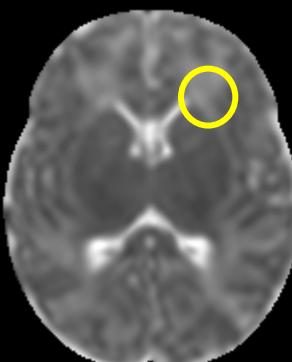
# Growth as well

# Neonatal Growing Brain (K. Oishi)

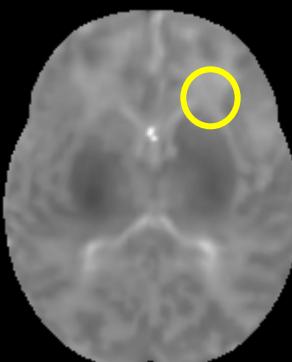
FA map



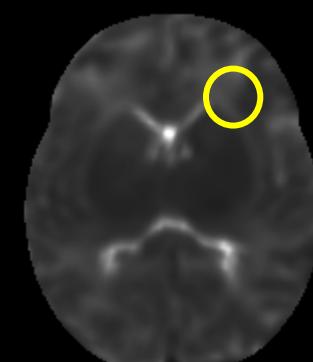
MD map



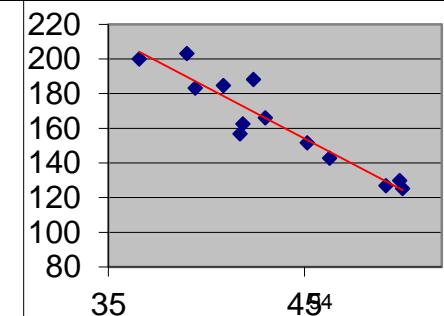
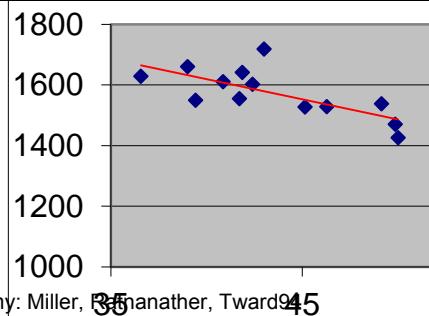
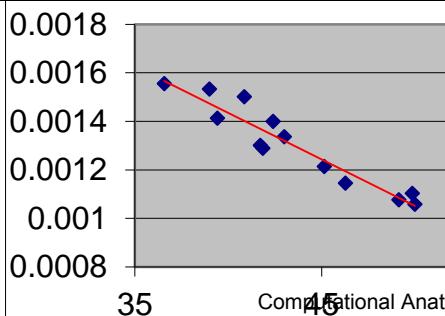
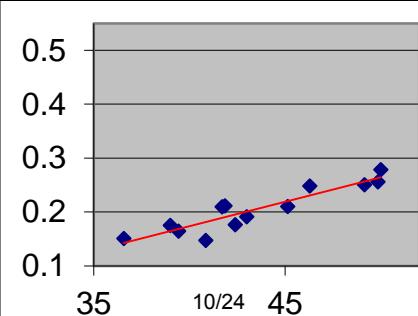
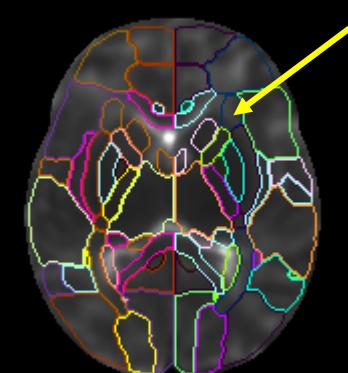
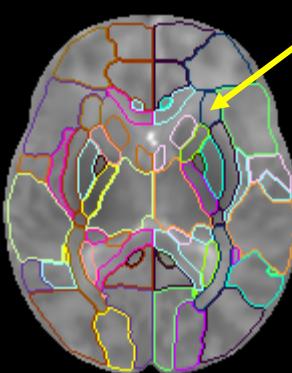
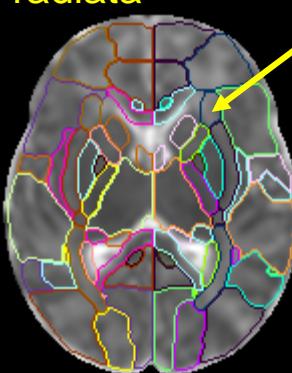
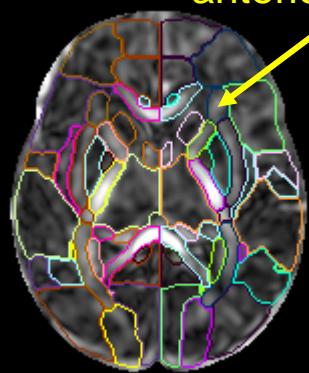
T<sub>1</sub> map



T<sub>2</sub> map



anterior corona radiata



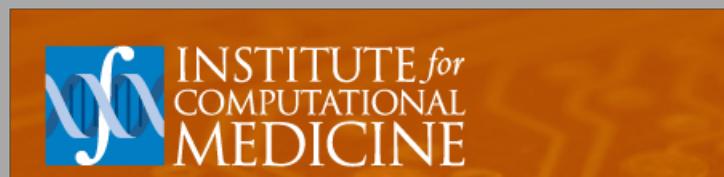
post conceptional weeks at scan

Computational Anatomy: Miller, Shanahan, Tward

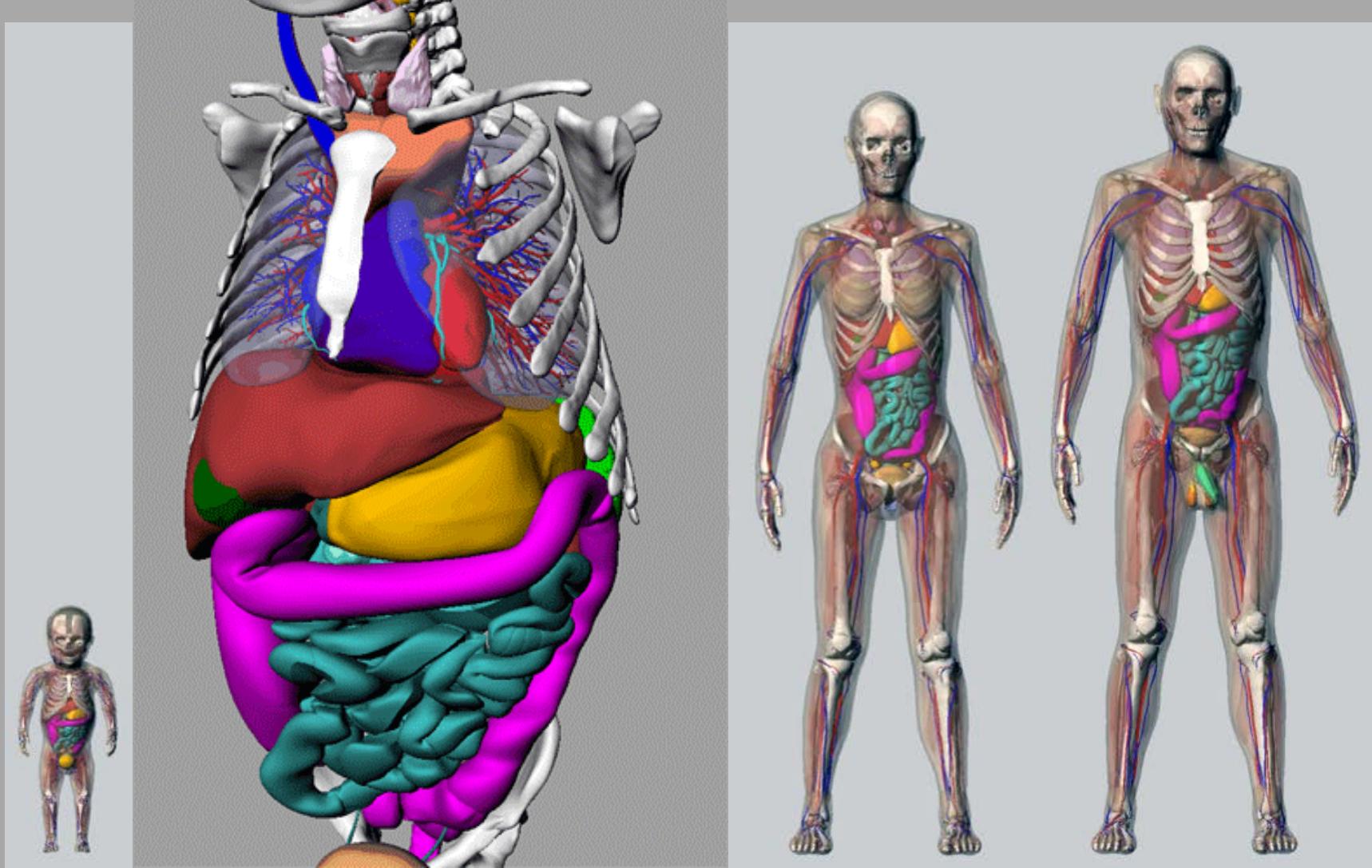
# Cardiac

# Whole Body Radiomics

- W. Paul Segars, Ph.D. (Duke University)
- Michael I. Miller, Ph.D. (JHU)
- Daniel J. Tward (JHU)
- J. Tilak Ratnanather, D.Phil. (JHU)



# New XCAT Phantoms

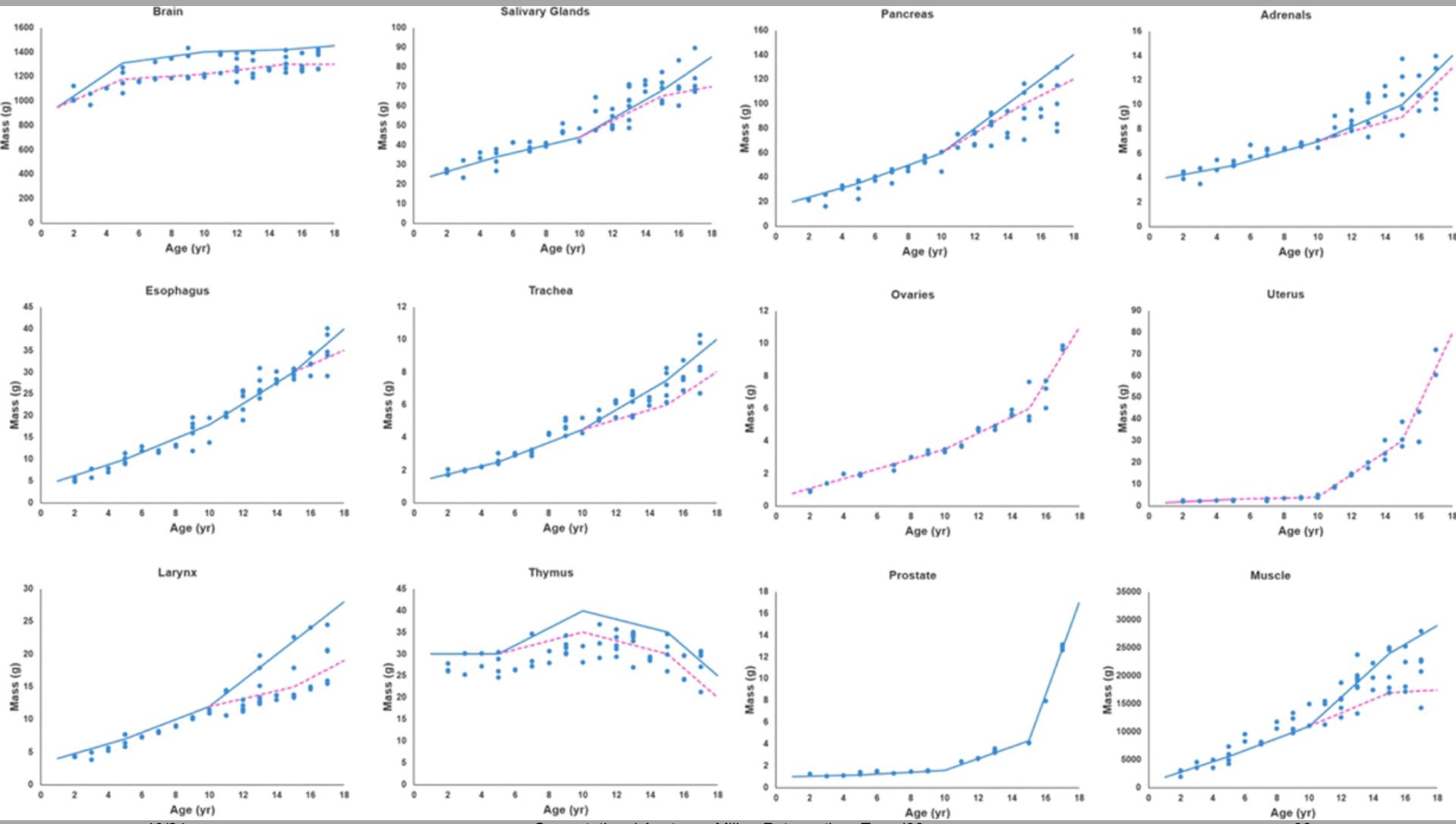


Segars el al, Population of anatomically variable 4D XCAT adult phantoms for imaging research and optimization, Med Phys, 40, (2013).

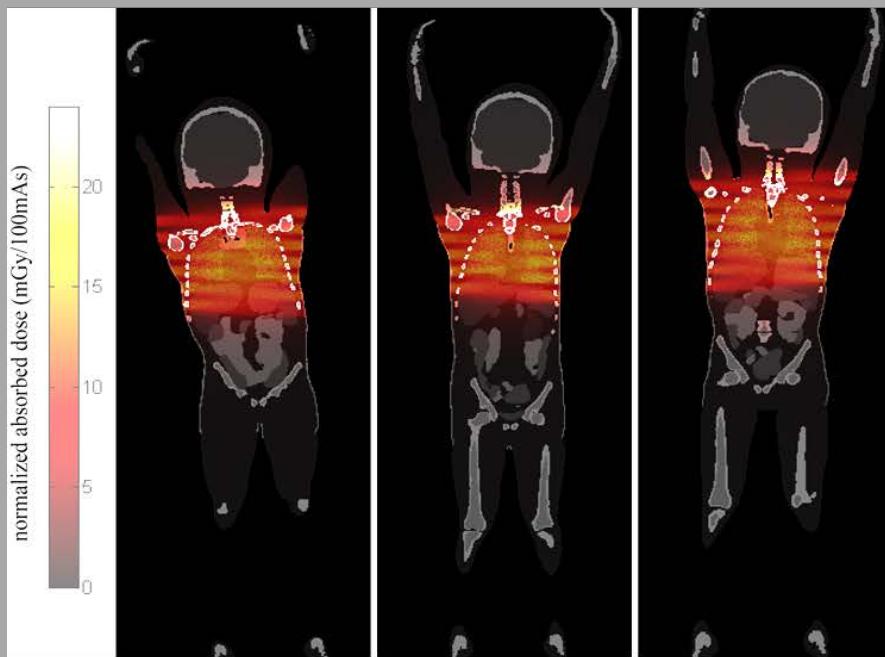
Norris el al, A set of 4D Pediatric XCAT Reference Phantoms for Multimodality Research, Med Phys, 41, (2014).

Segars el al, The development of a population of 4D pediatric XCAT phantoms for imaging research and optimization, Med Phys, (2015).

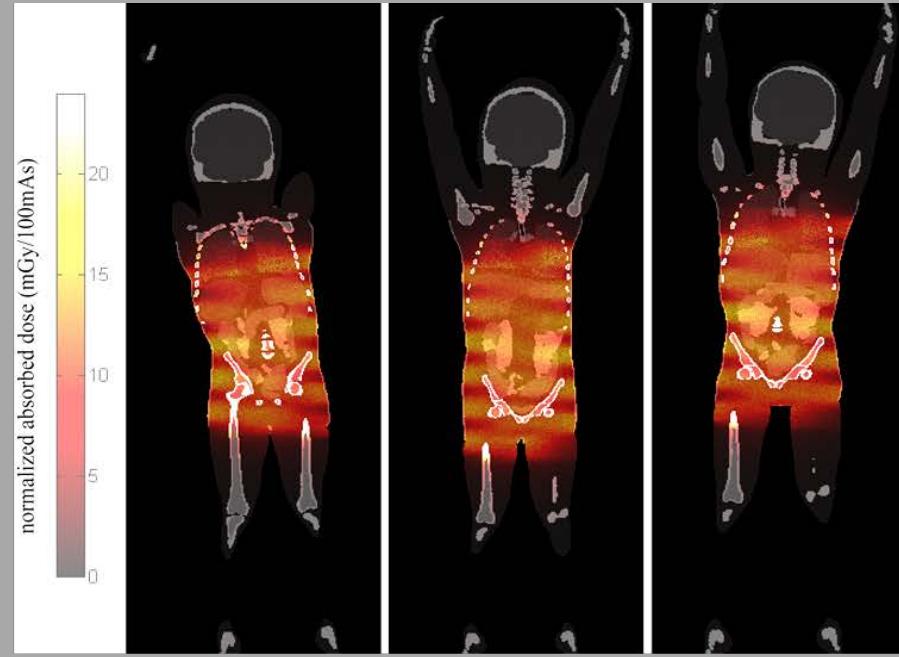
# Mass versus age for the organs and structures predicted by the MC-LDDMM transform



# Accurate Dose Estimation from Imaging Protocols



**Chest Scan**



**Abdomen Scan**

Coronal dose estimation in three 2 year-old phantoms for chest and abdomen CT imaging protocols

Brief Review: [Norris et al. \(2014\) Proc. of SPIE Vol. 9033, 90331V](#)  
Computational Anatomy: Miller, Rathnather, Tward99  
Experiments: [Li et al. \(2011\) Med Phys. 38: 397-407 and 408-419](#)

# Dynamical Systems Models Based on Hamilton's Principles of Least-Action

Michael I. Miller

Tilak Ratnanather

Daniel Tward



**Matrix groups support large motions (rigid & affine) but are not of sufficiently high-dimension for geometric change at the morphome scale.**

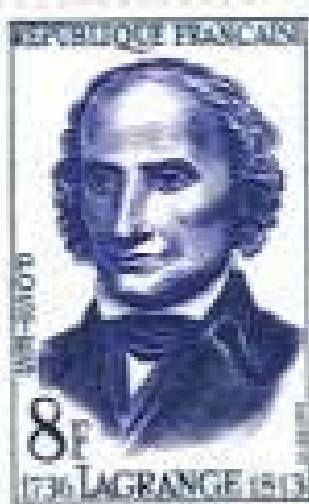
**In general high-dimensional spline vector fields do not represent transformations which are topology preserving for large change.**

**We introduce flows, a composition of spline vector fields, because they are 1-1 and onto for large deformations handling the self-intersecting problems of coordinate systems.**

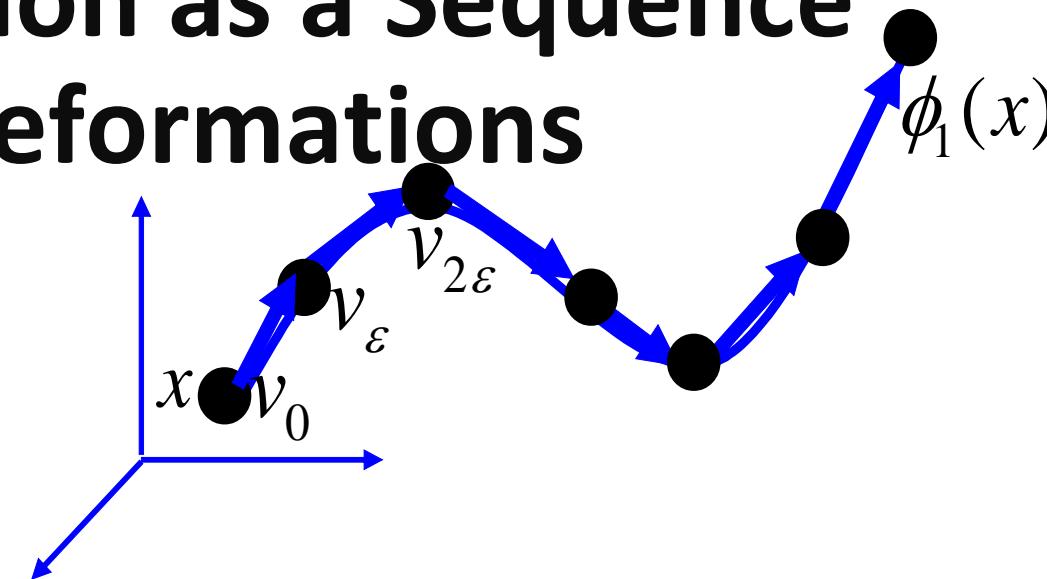
# Lagrangian and Eulerian Flows for Medical Imaging

[https://en.wikipedia.org/wiki/Computational\\_anatomy#Lagrangian\\_and\\_Eulerian\\_flows\\_for\\_generating\\_diffeomorphisms](https://en.wikipedia.org/wiki/Computational_anatomy#Lagrangian_and_Eulerian_flows_for_generating_diffeomorphisms)

Lagrange and Euler appreciated how large deformations can be constructed via composition of many small deformations.



# Large Deformation as a Sequence of Small Deformations



$$\phi_0(x) = x$$

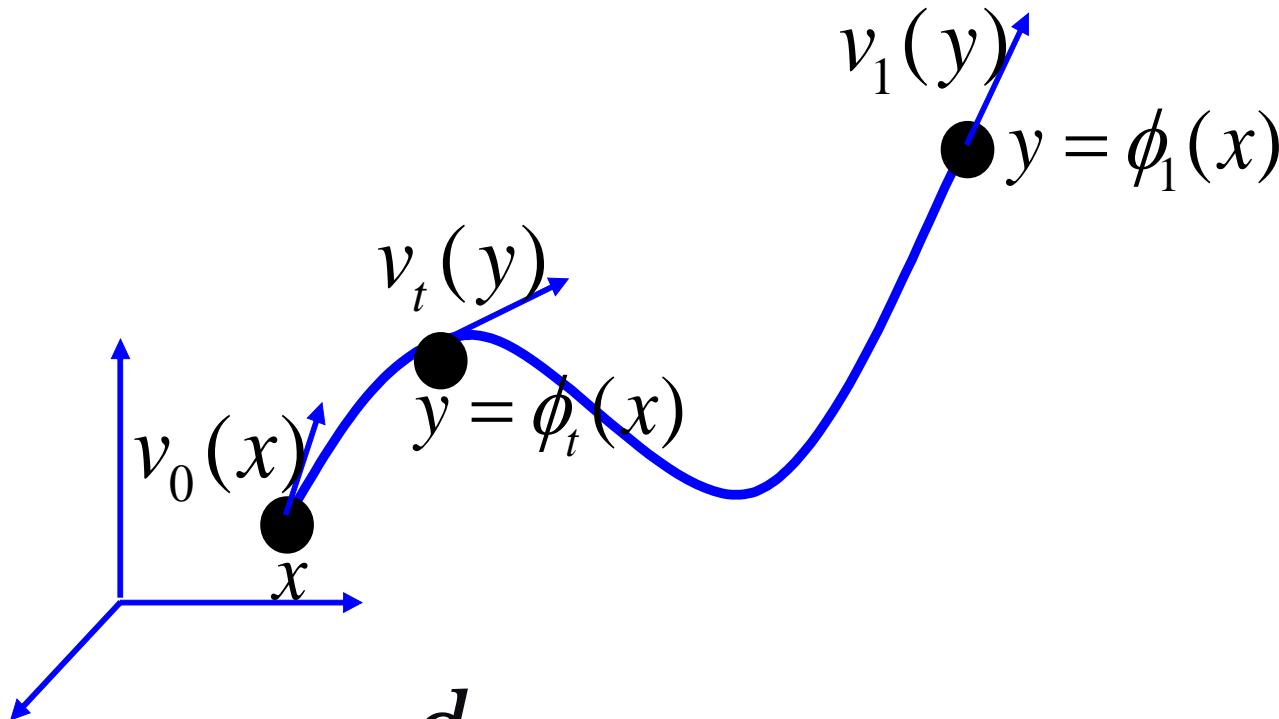
$$\phi_\epsilon(x) = x + v_0 \Big|_{\phi_0(x)}$$

$$\phi_{2\epsilon}(x) = x + v_0 \Big|_{\phi_0(x)} + v_\epsilon \Big|_{\phi_\epsilon(x)}$$

$$\phi_t(x) = x + v_0 \Big|_{\phi_0(x)} + v_\epsilon \Big|_{\phi_\epsilon(x)} + v_{2\epsilon} \Big|_{\phi_{2\epsilon}(x)} + \dots$$

$$\phi_t: x \mapsto x + \sum_{i=0,1,\dots} v_{i\epsilon} \circ \phi_{i\epsilon}(x), \quad v \text{ vector fields}$$

# Large Deformation Flow



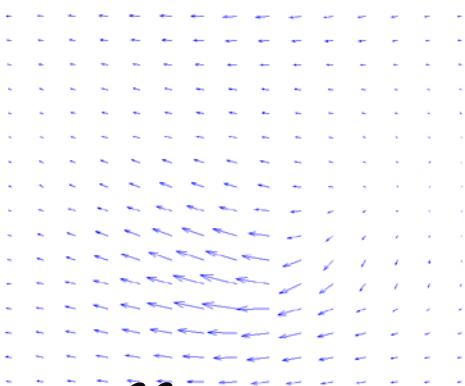
$$\frac{d}{dt} \phi_t = v_t \circ \phi_t$$

$$\phi_t: x \mapsto x + \int_0^t v_s \circ \phi_s(x) dx ,$$

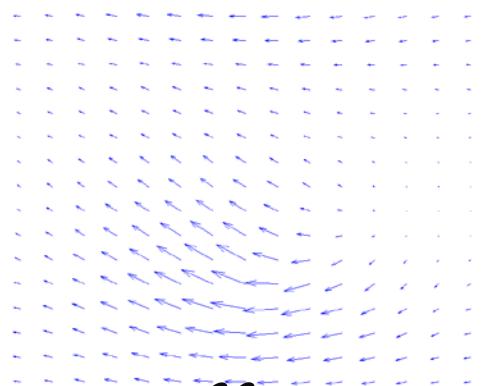
$v_0$  $v_6$  $v_{12}$ 

$$\frac{d}{dt} \phi_t = v_t \circ \phi_t$$

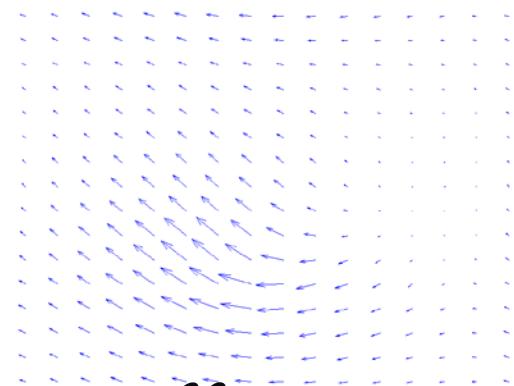
 $v_{18}$  $v_{24}$



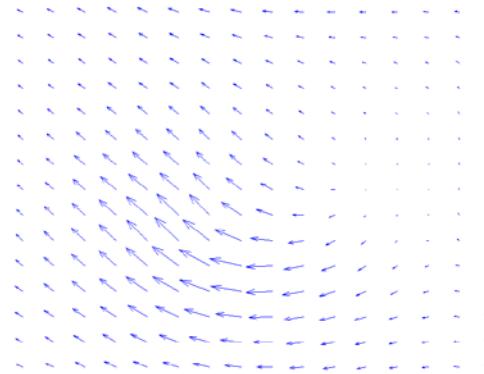
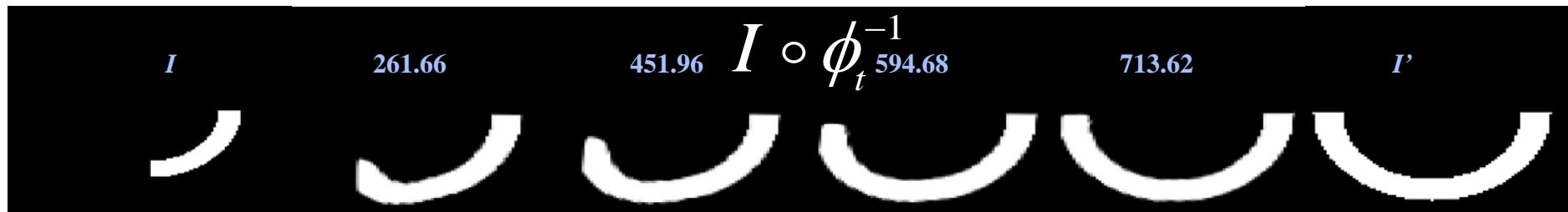
$v_0$



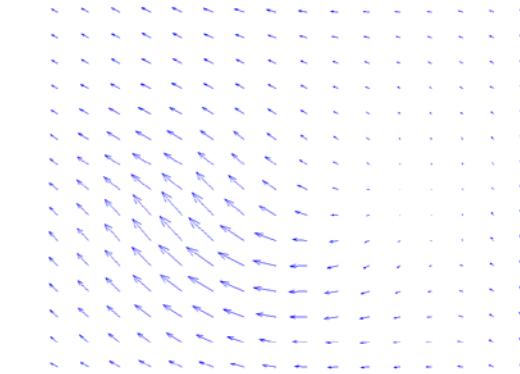
$v_6$



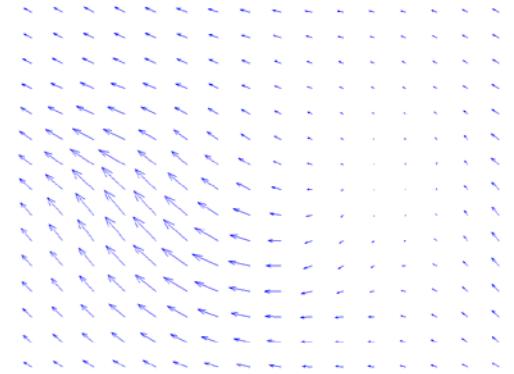
$v_{12}$



$v_{18}$



Computational Anatomy Miller, Romanather, Tward 106  
 $v_{24}$



$v_{29}$  106

**Flows emphasize dynamics of composition of many small deformations, rather than addition.**

**Splines are a special case reducing to addition.**

# Large deformation flows are Function Composition

$$\phi_t: x \mapsto x + \int_0^t v_s \circ \phi_s(x) ds ,$$

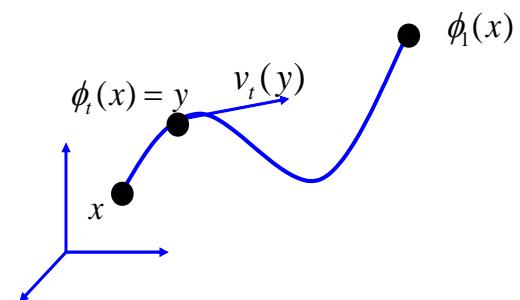
$$\phi_t = \phi_{t-\epsilon} + \phi_t - \phi_{t-\epsilon}$$

$$\approx (id + \epsilon v_{t-\epsilon}) \circ \phi_{t-\epsilon}$$

$$= (id + \epsilon v_{t-\epsilon}) \circ (\phi_{t-2\epsilon} + \phi_{t-\epsilon} - \phi_{t-2\epsilon})$$

$$\approx (id + \epsilon v_{t-\epsilon}) \circ (id + \epsilon v_{t-2\epsilon}) \circ \phi_{t-2\epsilon}$$

$$\approx (id + \epsilon v_{t-\epsilon}) \circ (id + \epsilon v_{t-2\epsilon}) \circ \dots \circ (id + \epsilon v_0)$$



## Splines are Addition

$$\phi_t: x \mapsto x + \int_0^t v_s(x) ds ,$$

$$\phi_t \approx id + \epsilon v_{t-\epsilon} + \epsilon v_{t-2\epsilon} + \dots + \epsilon v_0$$

# Registering Coordinate Systems via Large Deformation Image Matching (LDDMM)

[https://en.wikipedia.org/wiki/Large\\_deformation\\_diffeomorphic\\_metric\\_mapping](https://en.wikipedia.org/wiki/Large_deformation_diffeomorphic_metric_mapping)

Michael I. Miller

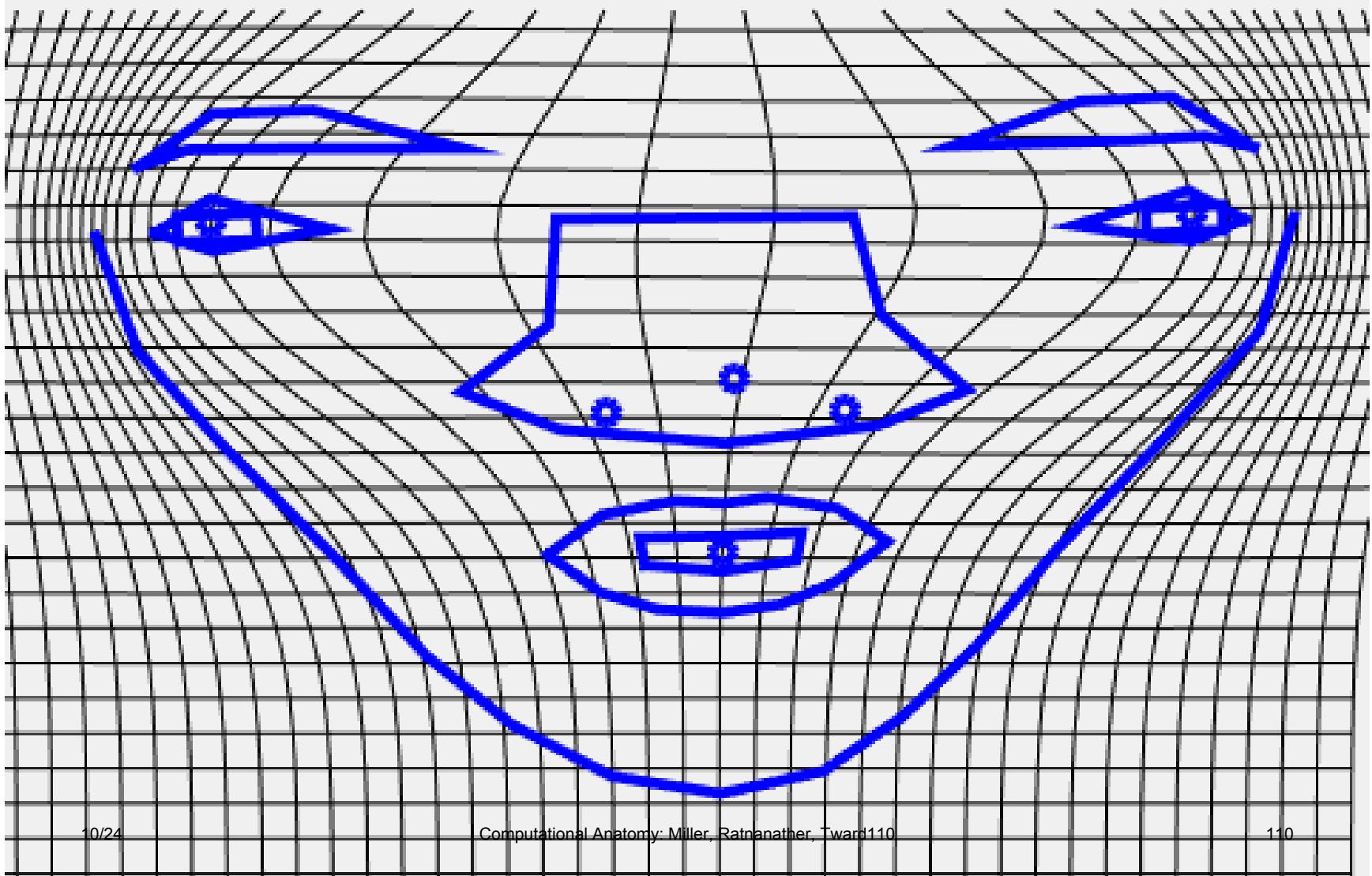
Tilak Ratnanather

Daniel Tward

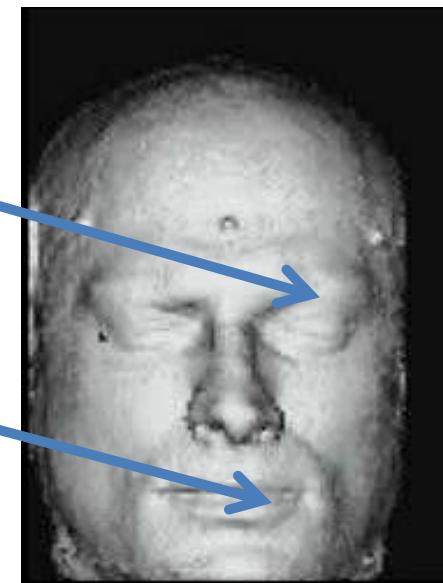
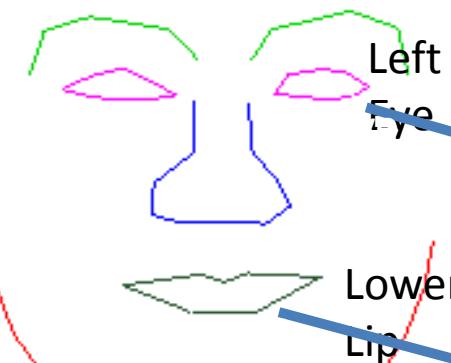


The computational algorithm for the GPS is called Large Deformation Diffeomorphic Metric Mapping.

### Diffeomorphism

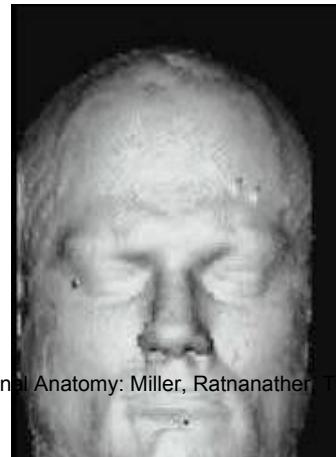


# Anatomy is an element in the orbit; the flow positions the information.



$$I = I_{temp} \circ \phi_1^{-1}$$

$$\frac{d}{dt} \phi_t = v_t \circ \phi_t, \phi_0 = id$$



# LDDMM Image Matching

[https://en.wikipedia.org/wiki/Large\\_deformation\\_diffeomorphic\\_metric\\_mapping#  
LDDMM algorithm for dense image matching](https://en.wikipedia.org/wiki/Large_deformation_diffeomorphic_metric_mapping#LDDMM_algorithm_for_dense_image_matching)

$$\dot{\phi}_t = v_t \circ \phi_t, \phi_0 = id$$

$$J(v) = \frac{1}{2} \int_0^1 \int_X A v_t \cdot v_t \, dx dt + \frac{1}{2} \int_X |I \circ \phi_1^{-1} - I'|^2 \, dx$$

$$\phi_{t,1} = \phi_1 \circ \phi_t^{-1}, \quad \frac{d}{d\epsilon} J(v + \epsilon w) \Big|_{\epsilon=0} = \\ \int_0^1 \int_X \underbrace{(A v_t - (I \circ \phi_t^{-1} - I' \circ \phi_{t,1}) \nabla (I \circ \phi_t^{-1}) |\partial_X(\phi_{t,1})|)}_{=0} \cdot w_t \, dx dt$$

Necessary Condition:

$$A v_t = (I \circ \phi_t^{-1} - I' \circ \phi_{t,1}) \nabla (I \circ \phi_t^{-1}) |\partial_X(\phi_{t,1})|$$

# Iteration for LDDMM Image Matching

[https://en.wikipedia.org/wiki/Large\\_deformation\\_diffeomorphic\\_metric\\_mapping#LDDMM\\_algorithm\\_for\\_dense\\_image\\_matching](https://en.wikipedia.org/wiki/Large_deformation_diffeomorphic_metric_mapping#LDDMM_algorithm_for_dense_image_matching)

$$\min_{\nu} \quad \frac{1}{2} \int_0^1 \int_X A\nu_t \cdot \nu_t \, dxdt + \frac{1}{2} \int_X |I \circ \phi_1^{-1} - I'|^2 \, dx$$

$$\phi_{t,1} = \phi_1 \circ \phi_t^{-1}$$

Necessary Condition:

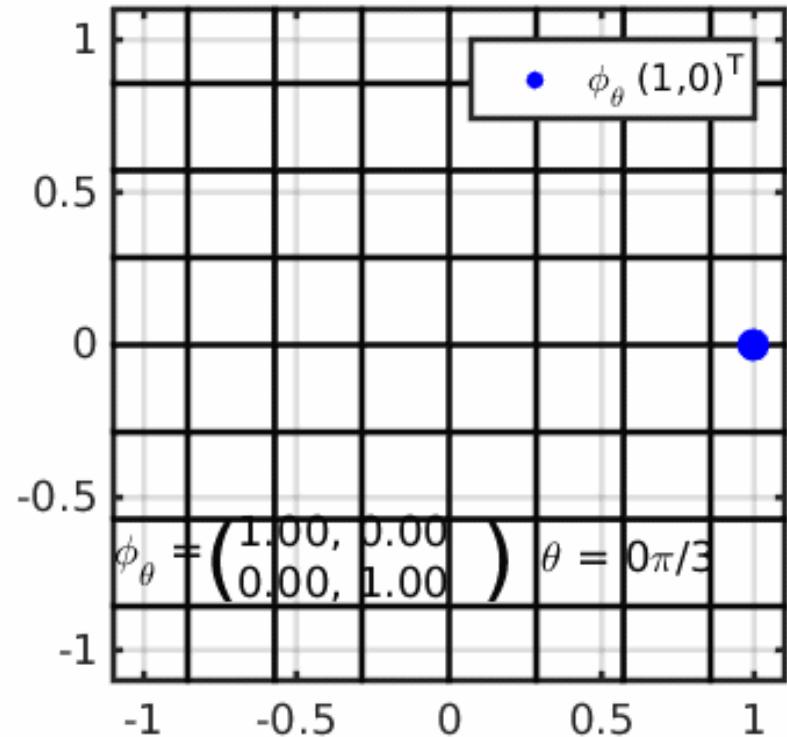
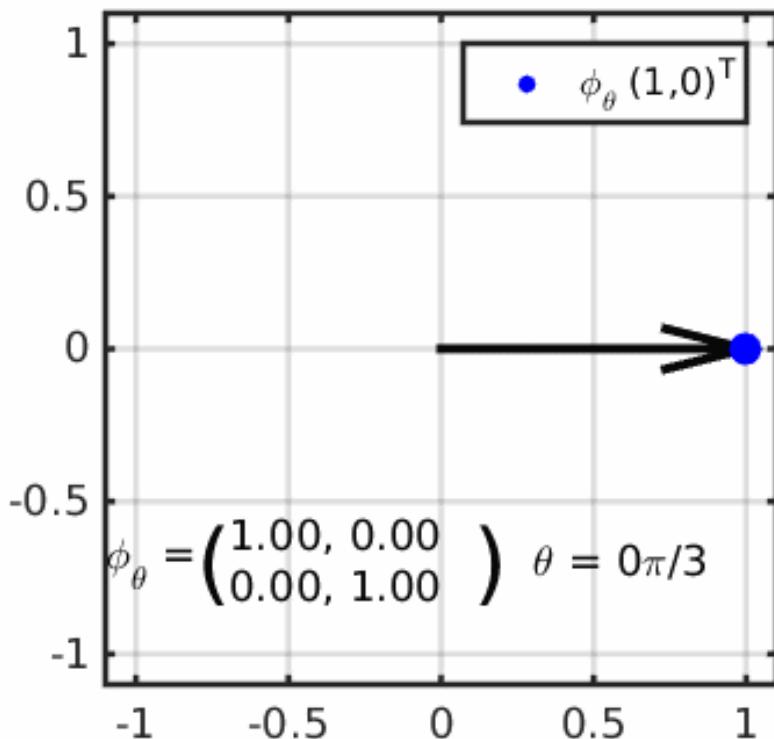
$$A\nu_t = (I \circ \phi_t^{-1} - I' \circ \phi_{t,1}) \nabla (I \circ \phi_t^{-1}) |\partial_X(\phi_{t,1})|$$

Non-Linear Iteration:

$$A\nu_t^{new} = A\nu_t^{old} + \epsilon \left( A\nu_t^{old} - (I \circ \phi_t^{-1, old} - I' \circ \phi_{t,1}^{old}) \right. \\ \left. \nabla (I \circ \phi_t^{old-1}) |\partial_X(\phi_{t,1}^{old})| \right)$$

**Daniel Tward:  
Do sequence of rotations  
as large deformation.**

# Large rotation as a composition of small rotations.



$$\underbrace{\varphi_n}_{\begin{pmatrix} \cos 30 & -\sin 30 \\ \sin 30 & \cos 30 \end{pmatrix}} \circ$$

$$\underbrace{\varphi_{n-1}}_{\begin{pmatrix} \cos 30 & -\sin 30 \\ \sin 30 & \cos 30 \end{pmatrix}} \circ \dots \circ$$

$$\underbrace{\phi_1}_{\begin{pmatrix} \cos 30 & -\sin 30 \\ \sin 30 & \cos 30 \end{pmatrix}}$$

# LDDMM Landmark Matching

[https://en.wikipedia.org/wiki/Large\\_deformation\\_diffeomorphic\\_metric\\_mapping#  
LDDMM registered landmark matching](https://en.wikipedia.org/wiki/Large_deformation_diffeomorphic_metric_mapping#LDDMM_registered_landmark_matching)

$$\min_{\nu} J = \frac{1}{2} \int_0^1 \int_X A \nu_t \cdot \nu_t \, dx dt + \frac{1}{2} \sum_i |y_i - \phi_1(x_i)|^2$$

$$\begin{aligned} \frac{d}{d\epsilon} J(\nu + \epsilon w) &= \\ \int_0^1 \int_X \left( A \nu_t - \sum_i \delta_{\phi_1(x_i)}(x) \partial_X(\phi_{t,1}) \right) \Big|_{\phi_1(x_i)}^T & (\phi_1(x_i) - y_i) \cdot w_t dx dt \end{aligned}$$

Non-Linear Iteration:

$$\begin{aligned} A \nu_t^{new} &= A \nu_t^{old} + \epsilon \left( A \nu_t^{old} \right. \\ &\quad \left. - \sum_i \delta_{\phi_1^{old}(x_i)}(y) \partial_X(\phi_{t,1}^{old}) \right) \Big|_{\phi_1^{old}(x_i)}^T (\phi_1^{old}(x_i) - y_i) \end{aligned}$$

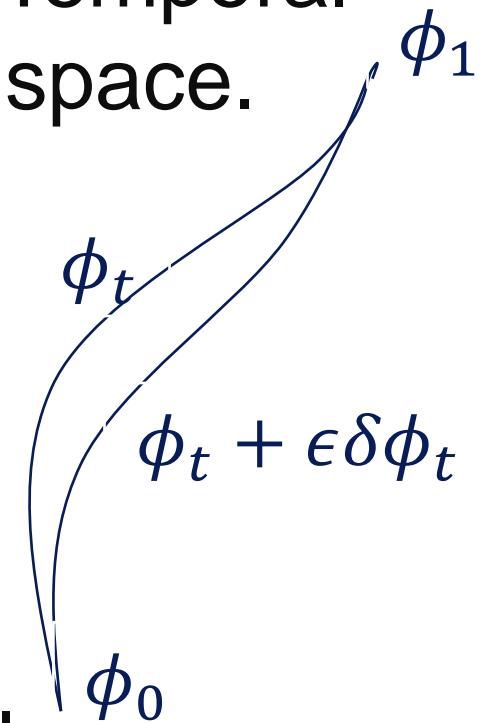
**There are many possible flows.**

**We use Hamilton's Principle of Least Action.**

[https://en.wikipedia.org/wiki/Computational\\_anatomy#The action integral for Hamilton.27s principle on diffeomorphic flows](https://en.wikipedia.org/wiki/Computational_anatomy#The_action_integral_for_Hamilton.27s_principle_on_diffeomorphic_flows)

# Hamilton's Principle of Least Action & the Lagrangian

- Coordinates  $\phi_t$  of system at time  $t$ . Temporal evolution is a curve in configuration space.
- Lagrangian  $\mathcal{L}(\phi_t, \dot{\phi}_t) = \text{K.E.-P.E.}$ , action integral  $J = \int_0^1 \mathcal{L}(\phi_t, \dot{\phi}_t) dt$
- PRINCIPLE: true evolution of system is stationary for action integral, a point with zero variation

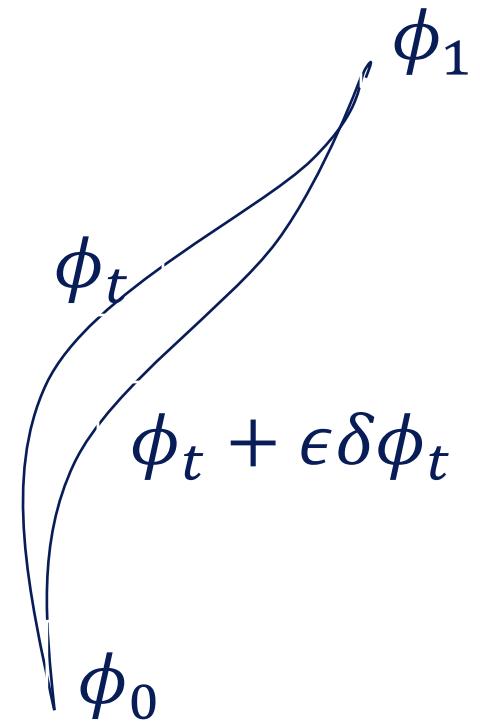


# Euler-Lagrange proof

- $f_t \doteq \phi_t, g_t \doteq \dot{\phi}_t, f^\epsilon = \phi + \epsilon h, g^\epsilon = \dot{\phi} + \epsilon \frac{d}{dt} h$
- $$\frac{d}{d\epsilon} J(\epsilon) \Big|_{\epsilon=0} = \int_0^1 \int_X \frac{d}{d\epsilon} L(f_t^\epsilon(x), g_t^\epsilon(x)) \Big|_{\epsilon=0} dx dt$$
- $$= \int_0^1 \int_X \left( \frac{\partial L(f_t^\epsilon, g_t^\epsilon)}{\partial f} \cdot \frac{df_t^\epsilon}{d\epsilon} + \frac{\partial L(f_t^\epsilon, g_t^\epsilon)}{\partial g} \cdot \frac{dg_t^\epsilon}{d\epsilon} \right) \Big|_{\epsilon=0} dx dt$$
- $$= \int_0^1 \int_X \left( \frac{\partial L(\phi_t, \dot{\phi}_t)}{\partial \phi} \cdot h_t + \frac{\partial L(\phi_t, \dot{\phi}_t)}{\partial \dot{\phi}} \cdot \frac{d}{dt} h_t \right) dx dt$$
- $$= \int_0^1 \int_X \underbrace{\left( \frac{\partial L(\phi_t, \dot{\phi}_t)}{\partial \phi} - \frac{d}{dt} \frac{\partial L(\phi_t, \dot{\phi}_t)}{\partial \dot{\phi}} \right)}_{=0 \text{ Euler-Lagrange}} \cdot h_t dx dt$$

# Euler-Lagrange Equation for Hamilton's Principle

- $J(\phi) = \int_0^1 \underbrace{\int_X L(\phi_t(x), \dot{\phi}_t(x)) dx}_{\text{Lagrangian } \mathcal{L}(\phi_t, \dot{\phi}_t)} dt$



- Hamiltonian Minimizer Satisfies  
 $\phi^\epsilon = \phi + \epsilon \delta \phi, \delta \phi_0 = \delta \phi_1 = 0$

$$\frac{d}{d\epsilon} \Big|_{\epsilon=0} J(\phi^\epsilon) = 0$$

- Euler-Lagrange Equation

$$-\frac{d}{dt} \frac{\partial L(\phi_t(x), \dot{\phi}_t(x))}{\partial \dot{\phi}} + \frac{\partial L(\phi_t(x), \dot{\phi}_t(x))}{\partial \phi} = 0$$

# Euler-Lagrange of Lumped Mass-Spring Mechanical System

Position  $\phi_t$  P. E. =  $\frac{1}{2}k\phi_t^2$    Velocity  $v_t = \dot{\phi}_t$  K. E. =  $\frac{1}{2}m\dot{\phi}_t^2$

Action Integral:

$$J = \int \underbrace{(K.E. - P.E.)}_{\text{Lagrangian } \mathcal{L}(\phi, \dot{\phi}_t)} dt$$

Lagrangian:

$$\mathcal{L}(\phi_t, \dot{\phi}_t) = \frac{1}{2}m\dot{\phi}_t^2 - \frac{1}{2}k\phi_t^2$$
$$-\frac{d}{dt} \frac{\partial \mathcal{L}(\phi_t, \dot{\phi}_t)}{\partial \dot{\phi}} + \frac{\partial \mathcal{L}(\phi_t, \dot{\phi}_t)}{\partial \phi} = -m\ddot{\phi}_t - k\phi_t = 0$$

Equation of Motion:

$$m\ddot{\phi}_t + k\phi_t = 0, \phi_t = \Phi_{init} \cos(\sqrt{k/m} t + \theta)$$

# **Large Deformation Geodesic Flows as Least Action**

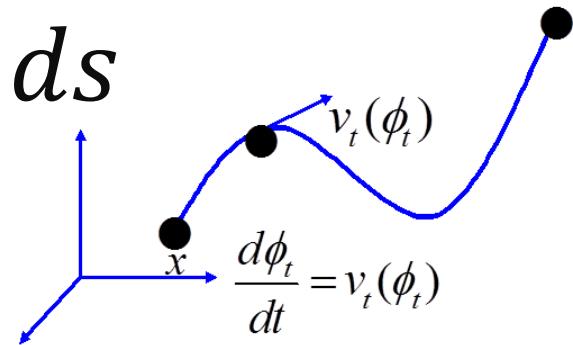
# Least Action (Geodesic) Large Deformations

[https://en.wikipedia.org/wiki/Computational\\_anatomy#Landmark or pointset geodesics](https://en.wikipedia.org/wiki/Computational_anatomy#Landmark_or_pointset_geodesics)

[https://en.wikipedia.org/wiki/Computational\\_anatomy#Surface geodesics](https://en.wikipedia.org/wiki/Computational_anatomy#Surface_geodesics)

[https://en.wikipedia.org/wiki/Computational\\_anatomy#Volume geodesics](https://en.wikipedia.org/wiki/Computational_anatomy#Volume_geodesics)

$$\phi_t: x \mapsto x + \int_0^t v_s \circ \phi_s(x) ds$$



$$\frac{d}{dt} \phi_t = v_t \circ \phi_t, \phi_0(x) = x$$

$$v_t(x) = \int_X k(x - \phi_t(y)) p_t(y) dy$$

$$\dot{p}_t = - (\partial_x v_t^T) \Big|_{\phi_t} p_t, \quad p_0 = A v_0$$

# Euler-Lagrange Large Deformations

[https://en.wikipedia.org/wiki/Computational\\_anatomy#Landmark or pointset geodesics](https://en.wikipedia.org/wiki/Computational_anatomy#Landmark_or_pointset_geodesics)

[https://en.wikipedia.org/wiki/Computational\\_anatomy#Surface geodesics](https://en.wikipedia.org/wiki/Computational_anatomy#Surface_geodesics)

[https://en.wikipedia.org/wiki/Computational\\_anatomy#Volume geodesics](https://en.wikipedia.org/wiki/Computational_anatomy#Volume_geodesics)

Velocity  $\dot{\phi}_t = v_t \circ \phi_t$

K.E.:  $L(\phi_t, \dot{\phi}_t) = \frac{1}{2} A \dot{\phi}_t \circ \phi_t^{-1} \cdot \dot{\phi}_t \circ \phi_t^{-1}$

Canonical Momentum

$$p_t(x) = \frac{\partial L(\phi_t(x), \dot{\phi}_t(x))}{\partial \dot{\phi}} = A v_t \circ \phi_t(x) |\partial_x \phi_t(x)|$$

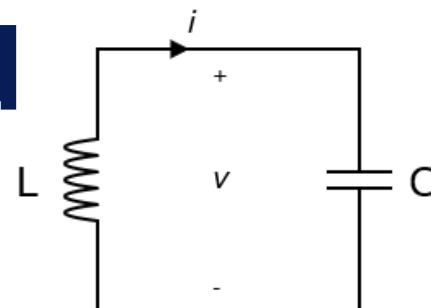
- Euler-Lagrange and Geodesic vector fields

$$\dot{p}_t = - (\partial_x v_t^T) \Big|_{\phi_t} p_t, \quad p_0 = A v_0$$

$$v_t(x) = \int_X k(x - \phi_t(y)) p_t(y) dy$$

# Daniel Tward Examples

# Euler-Lagrange of Lumped L-C Electrical System



Charge  $q_t$  P. E. =  $\frac{1}{2C} q_t^2$    Current  $i_t = \dot{q}_t$  K. E. =  $\frac{1}{2} L \dot{q}_t^2$

Action Integral:

$$J = \int \underbrace{(K.E. - P.E.)}_{\text{Lagrangian } \mathcal{L}(q, \dot{q}_t)} dt$$

Lagrangian:  $\mathcal{L}(q_t, \dot{q}_t) = \frac{1}{2} L \dot{q}_t^2 - \frac{1}{2C} q_t^2$

$$-\frac{d}{dt} \frac{\partial \mathcal{L}(q_t, \dot{q}_t)}{\partial \dot{q}} + \frac{\partial \mathcal{L}(q_t, \dot{q}_t)}{\partial q} = -L \ddot{q}_t - \frac{1}{C} q_t = 0$$

Kirchoff's Voltage Law of Motion:

$$L \ddot{q}_t + 1/C q_t = 0, q_t = Q_{init} \cos(\sqrt{1/LC} t + \theta)$$

# Euler-Lagrange Cannonball

- Position-Velocity:  $\phi_t = \begin{pmatrix} \phi_{tx} \\ \phi_{tz} \end{pmatrix}$ ,  $\dot{\phi}_t = \begin{pmatrix} \dot{\phi}_{tx} \\ \dot{\phi}_{tz} \end{pmatrix} = v_t$   
 $\phi_{0x} = \phi_{0z} = \dot{\phi}_{1z} = 0, \phi_{1x} = 1$
- Lagrangian: Kinetic minus Potential  
$$\mathcal{L}(\phi_t, \dot{\phi}_t) = \frac{1}{2} m \dot{\phi}_t \cdot \dot{\phi}_t - mg \phi_{tz}$$
- Canonical Momentum:  
$$p_t = \frac{\partial \mathcal{L}(\phi_t, \dot{\phi}_t)}{\partial \dot{\phi}} = m \dot{\phi}_t$$
- Euler-Lagrange Equation  
$$\dot{p}_{tx} = 0, \quad \dot{p}_{tz} = -mg$$
- Geodesic Solution  $\phi_{tx} = t, \phi_{tz} = -\frac{1}{2}gt^2 + \frac{1}{2}gt^{127}$

# Cannonball cont'd

- Given Least Action Euler-Lagrange Equation,

$$\dot{p}_{tx} = 0, \quad \dot{p}_{tz} = -mg, \quad p_t = m \dot{\phi}_t$$

with B.C.'s  $\phi_{0x} = \phi_{0z} = \phi_{1z} = 0, \phi_{1x} = 1$ .

Prove geodesic:  $\phi_{tx} = t, \quad \phi_{tz} = -\frac{1}{2}gt^2 + \frac{1}{2}gt$

$$\dot{p}_{tx} = m\ddot{\phi}_{tx} = 0 \Rightarrow \phi_{tx} = at + b \Rightarrow b = \phi_{00} = 0$$
$$\phi_{1x} = 1 \Rightarrow a = 1 \Rightarrow \phi_{tx} = t$$
$$\dot{p}_{tz} = -mg \Rightarrow m\ddot{\phi}_{tz} = -mg \Rightarrow \dot{\phi}_{tz} = -gt + c$$
$$\Rightarrow \phi_{tz} = -\frac{1}{2}gt^2 + ct + d$$
$$\phi_{0z} = 0 \Rightarrow d = 0, \phi_{1z} = 0 \Rightarrow c = \frac{1}{2}g$$

# Daniel Tward

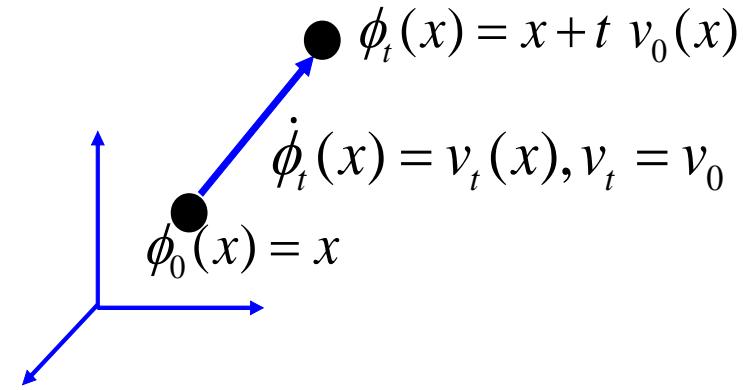
# Small Deformation Splines are Least Action

# Splines are Small Deformation Flows following Constant Speed Straight Lines

$$\phi_t: x \mapsto x + \int_0^t v_s(x) ds$$

$$\frac{d}{dt} \phi_t = v_t, \phi_0(x) = x$$

$$v_t(x) = \int_X k(x - y)p_0(y)dy = v_0(x), p_0 = A v_0$$



# Euler-Lagrange: Small Deformation Splines

- Velocity (3D):  $\dot{\phi}_t(x) = v_t(x), x \in X$
- K.E. Density: 
$$L(\phi_t(x), \dot{\phi}_t(x)) = \frac{1}{2} A v_t(x) \cdot v_t(x)$$
$$= \frac{1}{2} A \dot{\phi}_t(x) \cdot \dot{\phi}_t(x)$$
- Canonical Momentum:

$$p_t(x) = \frac{\partial L(\phi_t(x), \dot{\phi}_t(x))}{\partial \dot{\phi}} = A v_t(x)$$

- Euler-Lagrange and Geodesic vector fields

$$\frac{\partial L}{\partial \dot{\phi}} = 0, \quad \dot{p}_t = 0, \quad p_0 = A v_0$$

$$v_t(x) = \int_X k(x - y) p_t(y) dy$$

# Systems Models of Medical Imaging

Michael I. Miller

Tilak Ratnanather

Daniel Tward



**Physicists use Energy methods; engineers use systems methods.**

**Examine our systems model of imagery.**

# Landmark Based Spline Models of Images

Michael I. Miller

Tilak Ratnanather

Daniel Tward

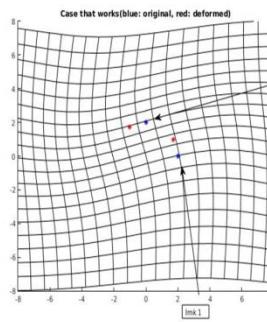


Fred Bookstein  
Rohlf Medal

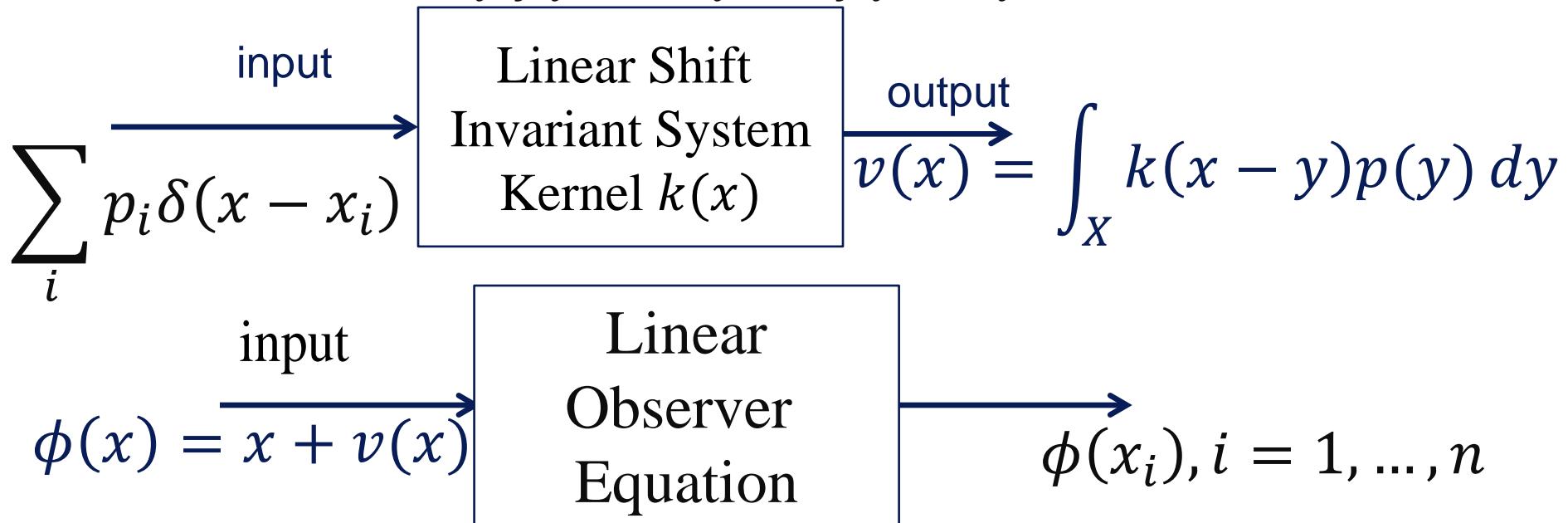


Computational Anatomy: Miller, Ratnanather, Tward134

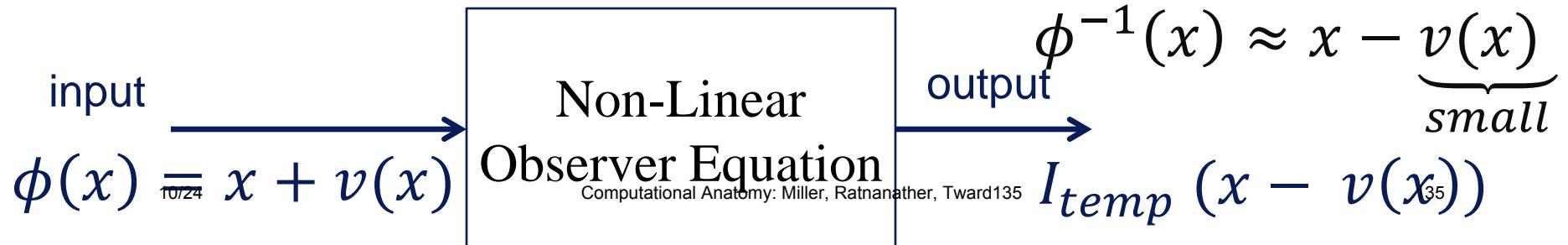
# Small Deformation Spline Model of Medical Imaging



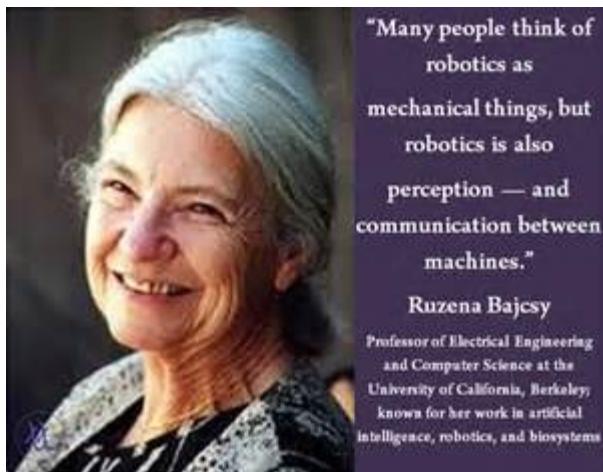
- Given B.C.'s  $(x_i, y_i), v(x_i) = y_i - x_i \quad i = 1, \dots, n$



Non – Linear Image Transformation (Small):



# Small Deformation Spline Models of Images



Ruzena Bajcsy

10/24

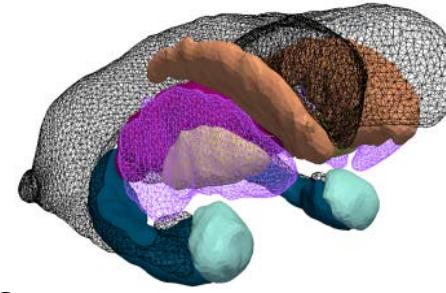
Michael I. Miller  
Tilak Ratnanather  
Daniel Tward



Computational Anatomy: Miller, Ratnanather, Tward 136

136

# Small Deformation Spline Model of Medical Imaging



Controller: Linear Shift Invariant System

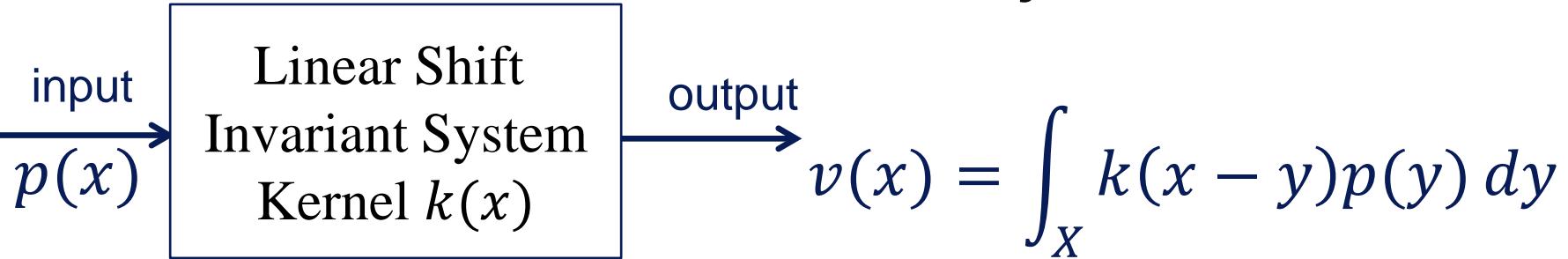
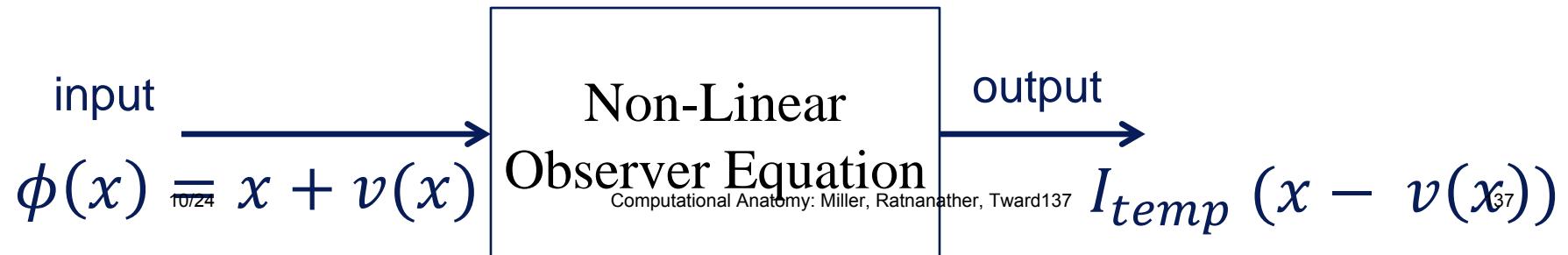


Image Transformation (Small):

$$\phi^{-1}(x) \approx x - \underbrace{v(x)}_{small}$$



# Dynamical Systems Model of Images



Gary Christensen



Richard Rabbitt

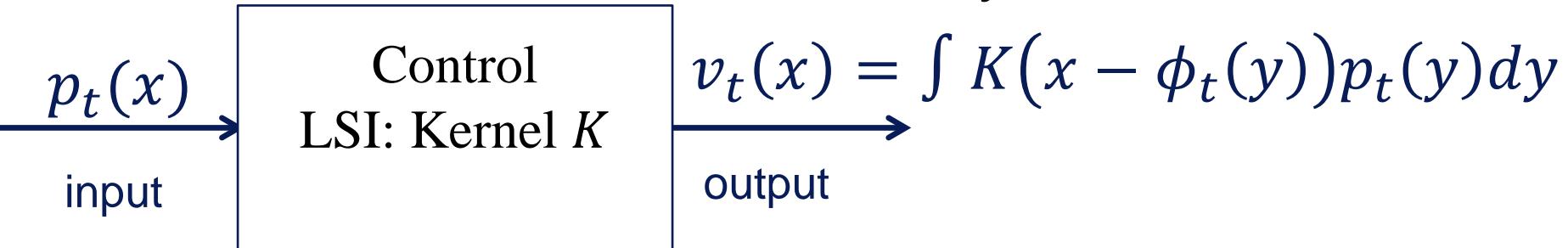
Michael I. Miller  
Tilak Ratnanather  
Daniel Tward



Computational Anatomy: Miller, Ratnanather, Tward 138

# Dynamical System Model of Medical Imaging

Controller: Linear Shift Invariant System



Dynamical System Constraint

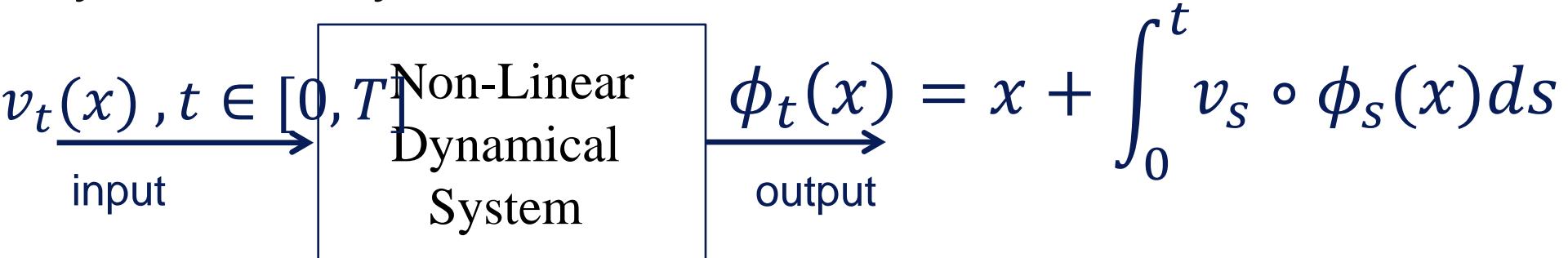
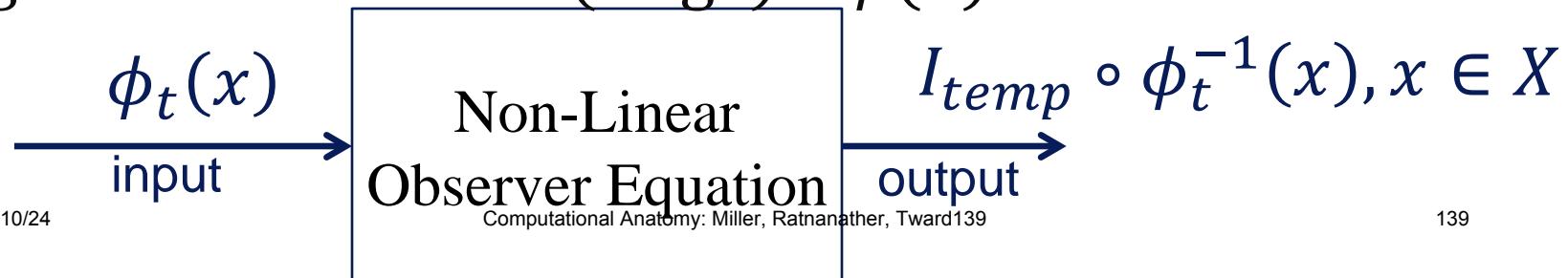


Image Transformation (large):  $\phi(x)$



# Geodesic Shooting Dynamical Systems Model of Medical Imagery

Geodesic Momentum Dynamics with Linear Controller

Geodesic Momentum  $\dot{p}_t(x) = (-\partial_x v_t)^T p_t, p_0 \text{ init. cond.}$

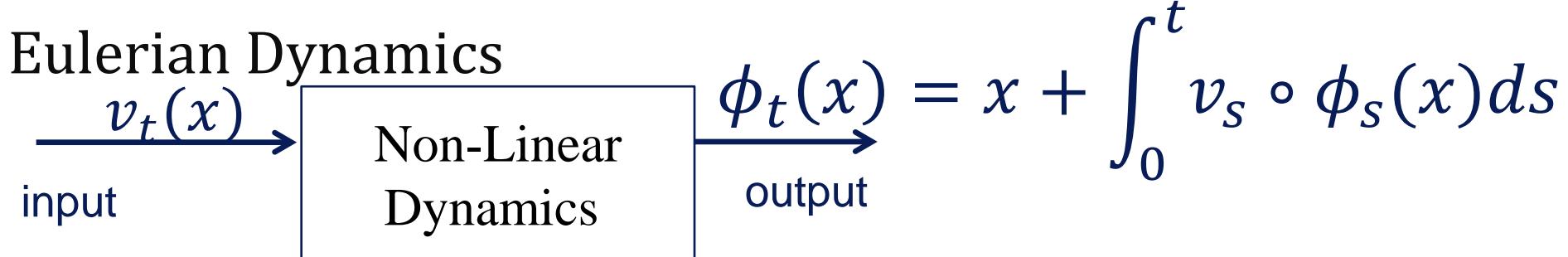
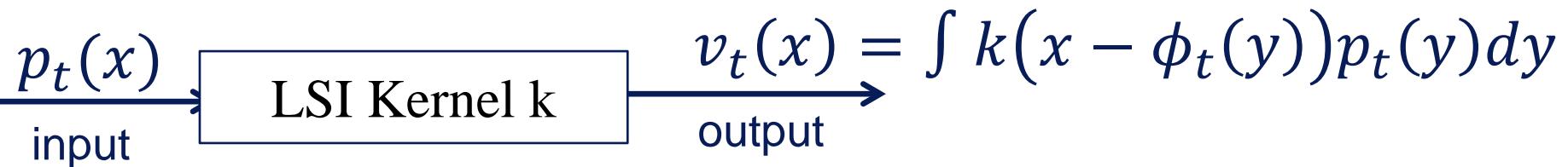
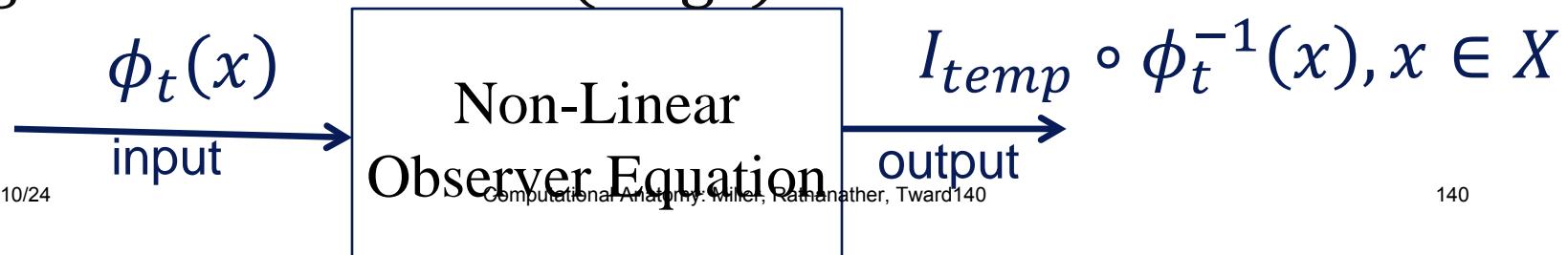


Image Transformation (large):



# Homework