Coursera Class Computational Anatomy and Neuroinformatics

https://en.wikipedia.org/wiki/Computational_Anatomy

Michael I. Miller
Tilak Ratnanather
Daniel Tward

Center for IMAGING SCIENCE
JOHNS HOPKINS UNIVERSITY

Matrix Groups Transforming Images

Michael I. Miller

Tilak Ratnanather

Daniel Tward

Center for IMAGING SCIENCE

JOHNS HOPKINS UNIVERSITY

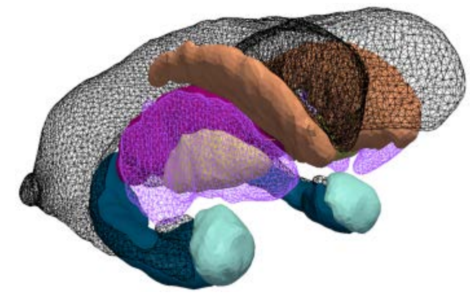
Matrix Group Transformations

Matrix Groups for Shifting, Scaling, Rotation

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

$\phi: (x_1, x_2) \in \mathbb{R}^2 \mapsto \phi(x) = (\phi_1(x), \phi_2(x)) \in \mathbb{R}^2$

Matrix notation:



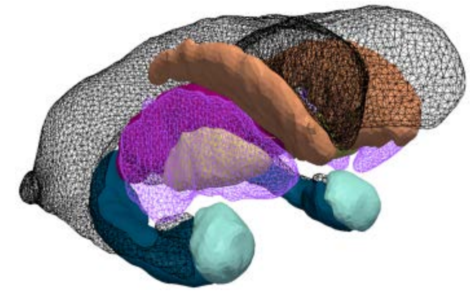
$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \phi(x) = \begin{pmatrix} \phi_1(x) \\ \phi_2(x) \end{pmatrix}$$

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

Vector Field Notation:



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

Spatial Coordinates and Matrix Notation

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

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

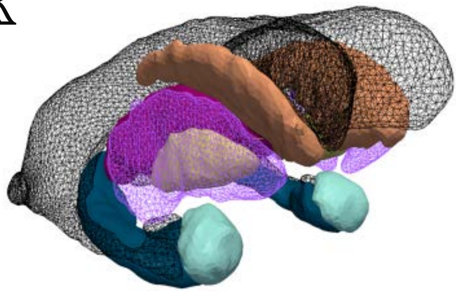
$$x \doteq (x_1, x_2, x_3) \in \mathbb{R}^3$$

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

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

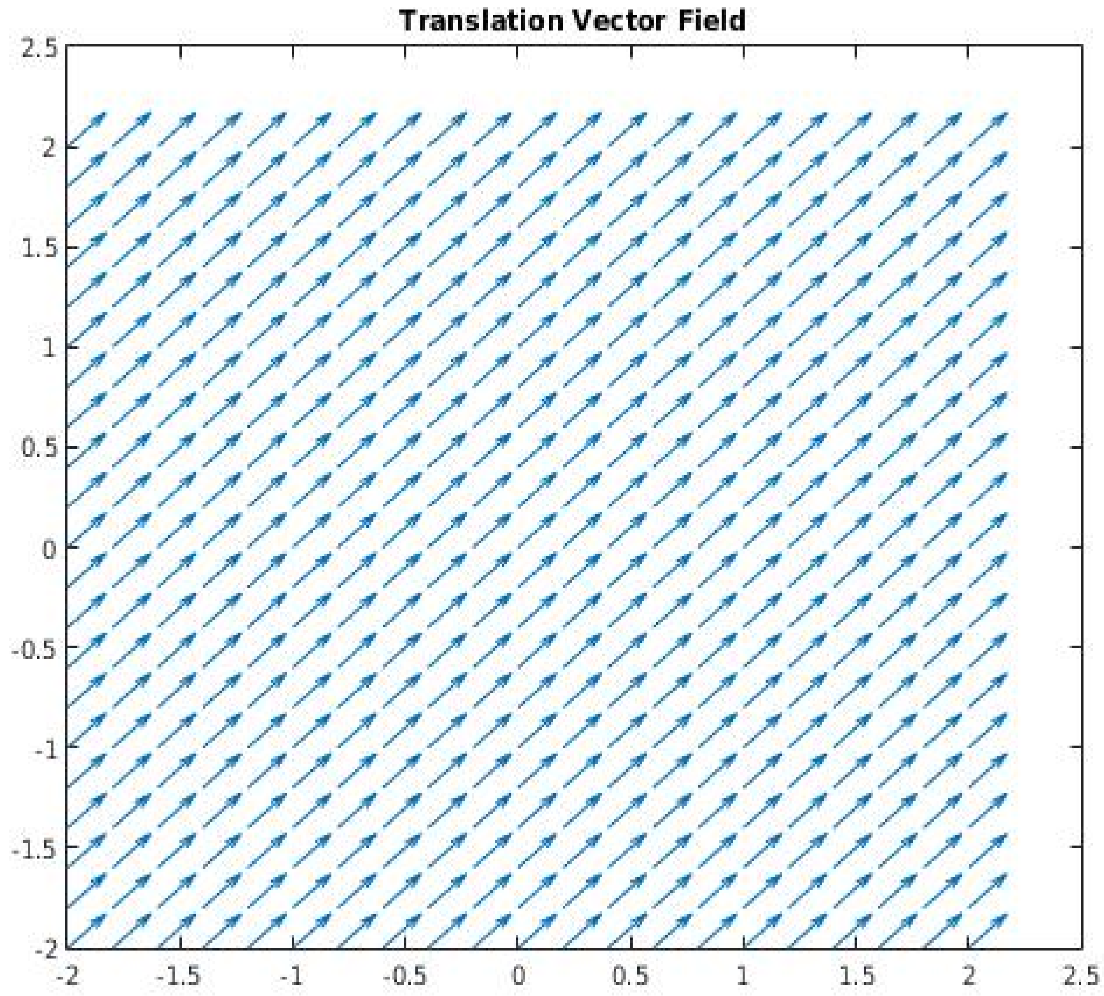
Scale: $v(x) = (s-1) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

Translation

Translation Vector Field in the plane
$x = (x_1, x_2\ ) \in \mathbb{R}^2$

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

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



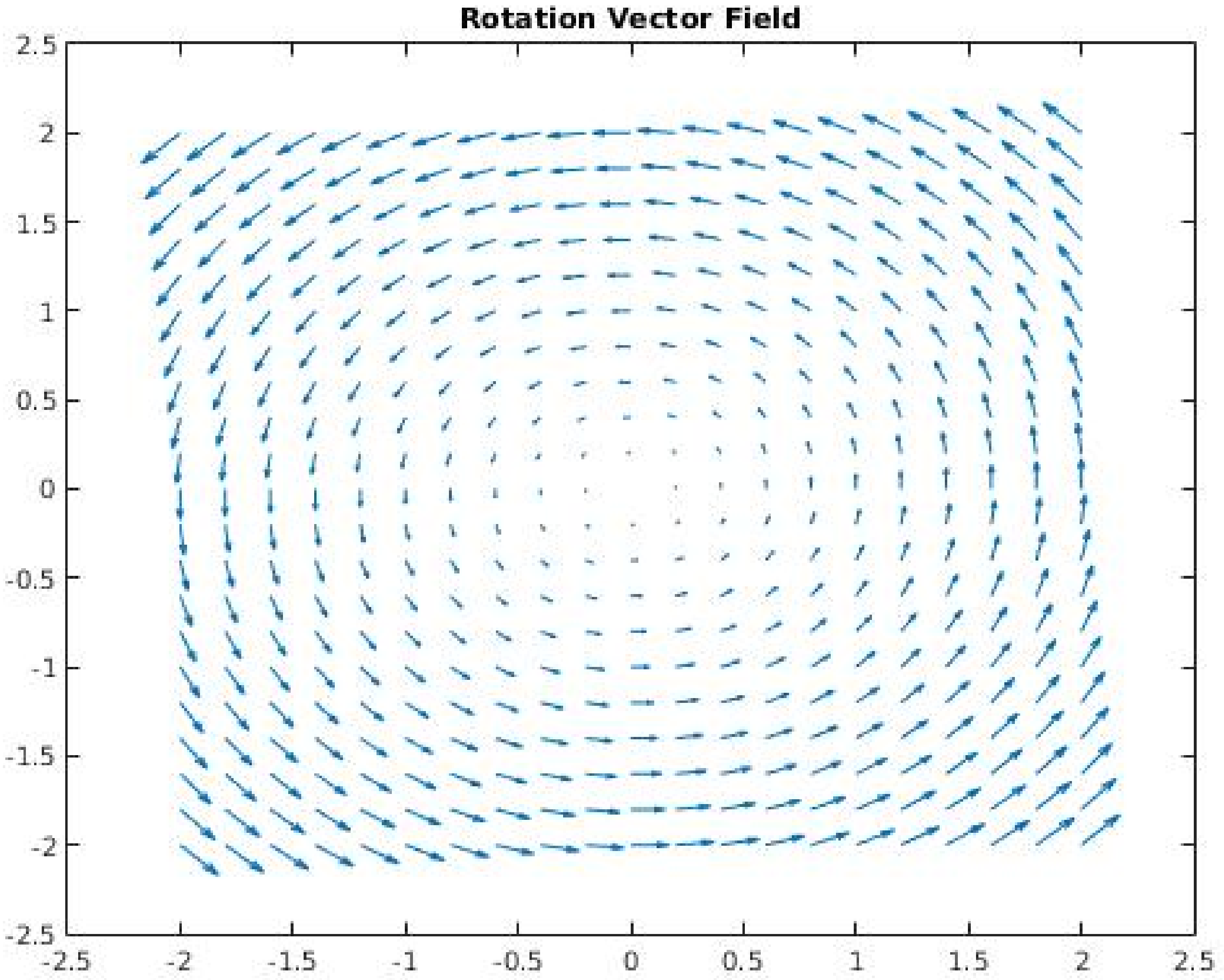
2-dimension

Rotation

Rotation Vector Field
in the plane
$x = (x_1, x_2) \in \mathbb{R}^2$

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

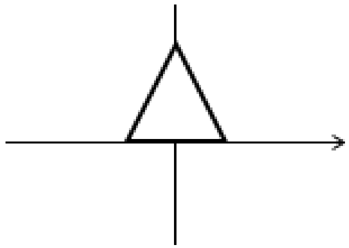
$v(x) = \begin{pmatrix} \cos\,\theta - 1 & -\sin\theta \\ \sin\theta & \cos\,\theta - 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

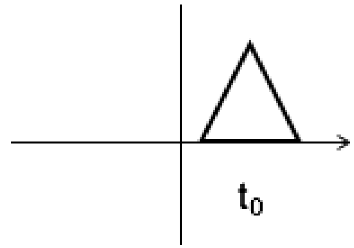


θ: 1-dimension

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

$u(t)$ $\qquad$ $u(t - t_0), \quad t_0 > 0$

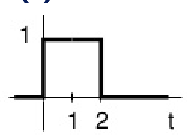


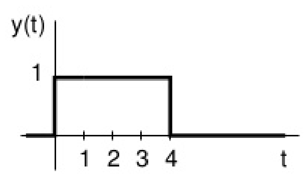


U(t) $= u_s(t) - u_s\,(t-2)$

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

U(t)



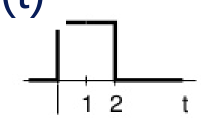


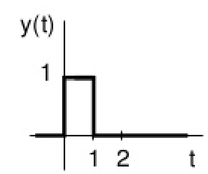
This has effectively “slowed dow U(t) by a factor of 2
(What occurred at t=1 now occurs at t=2)

U(t) $= u_s(t) - u_s\,(t-2)$

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

U(t)





This has effectively “sped u 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: $I(x+b)$

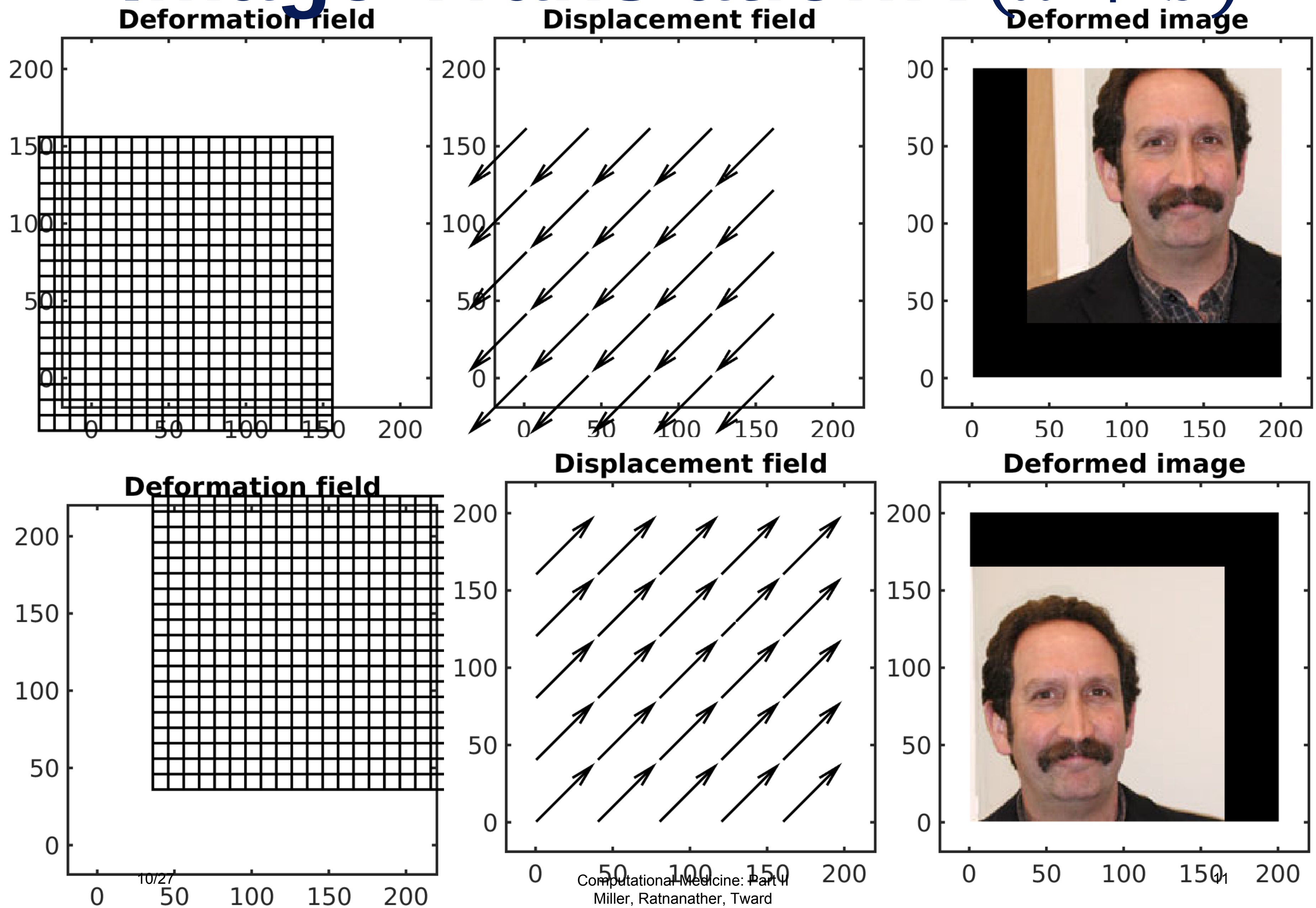


Image Scale: $I(sx)$

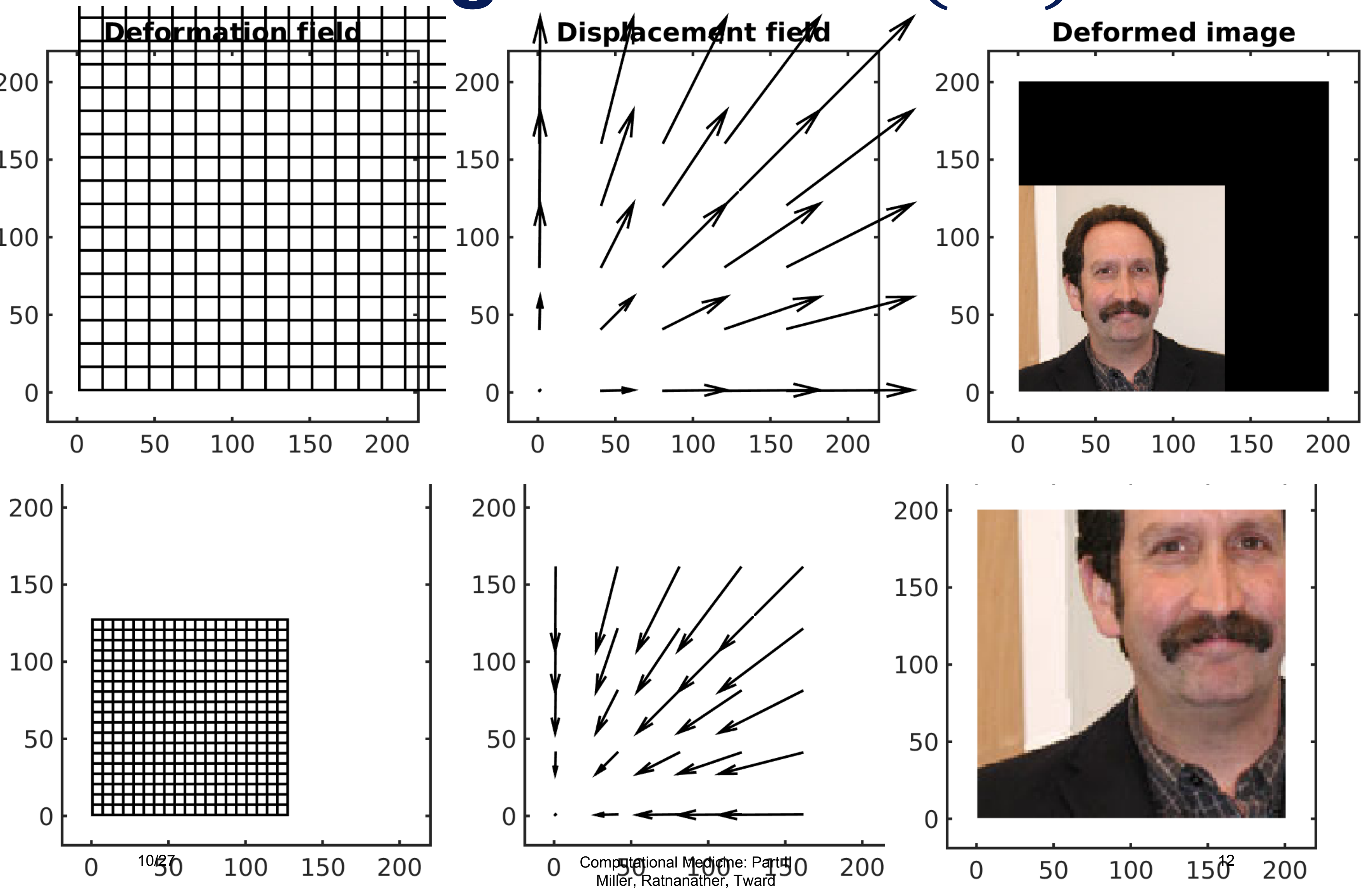
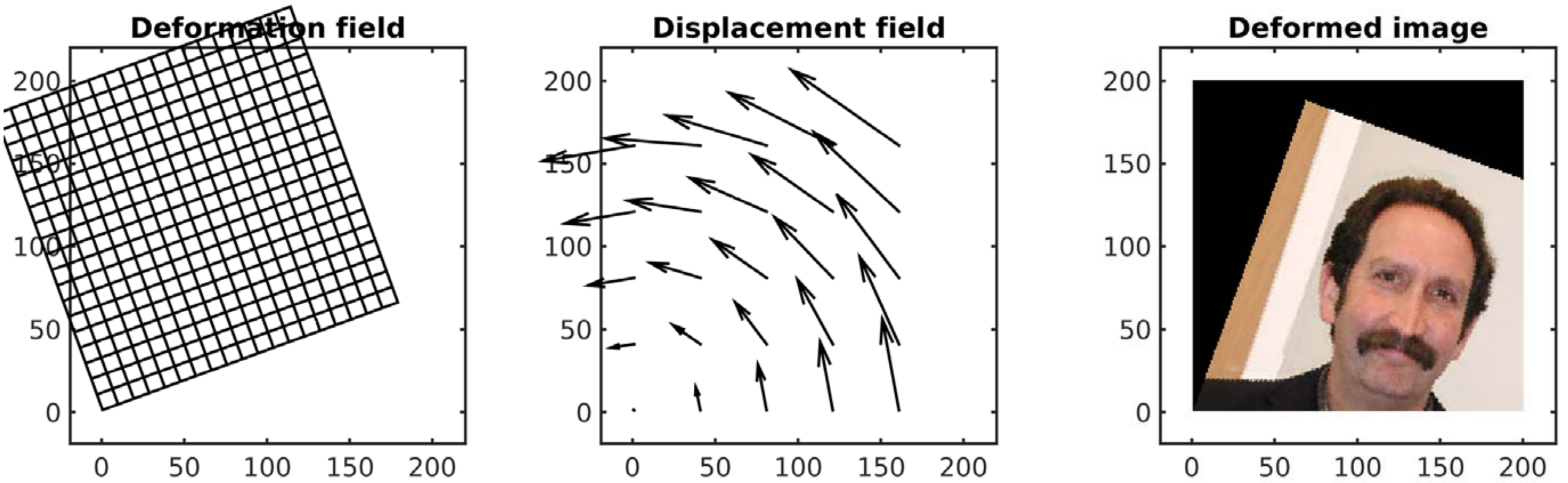


Image Rotation $I(\begin{pmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{pmatrix}x)$



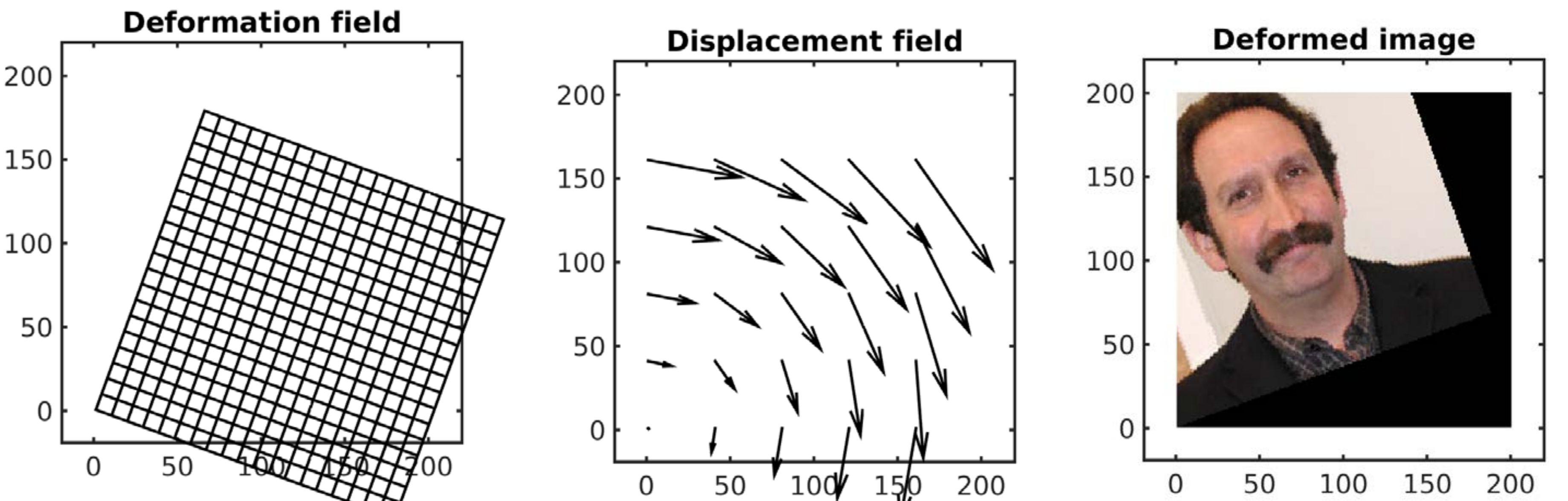
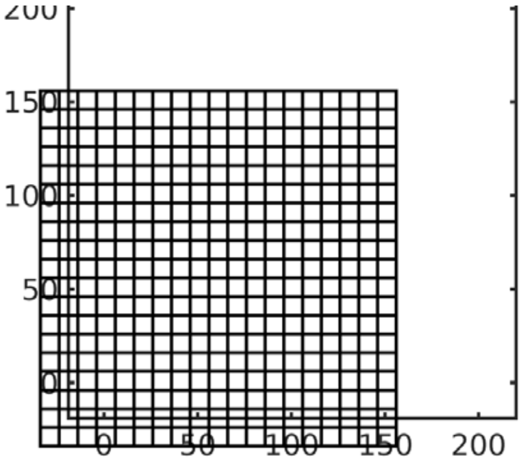
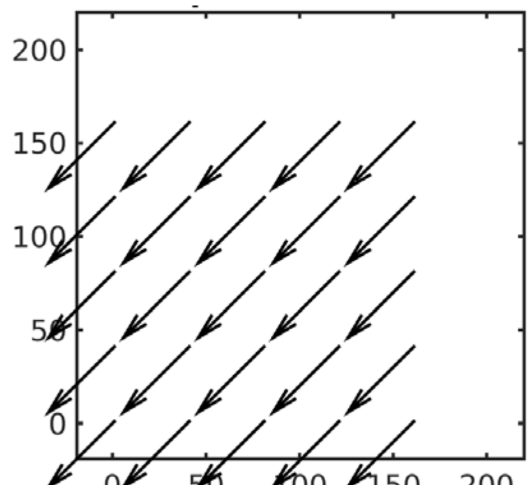
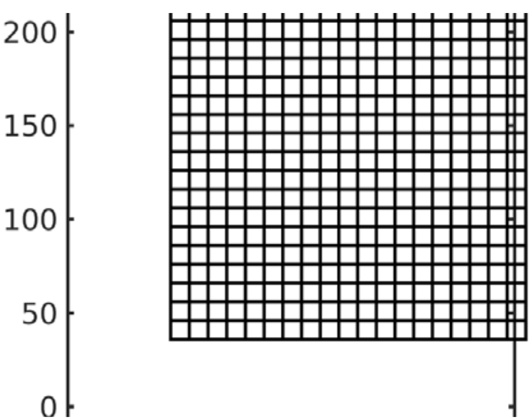


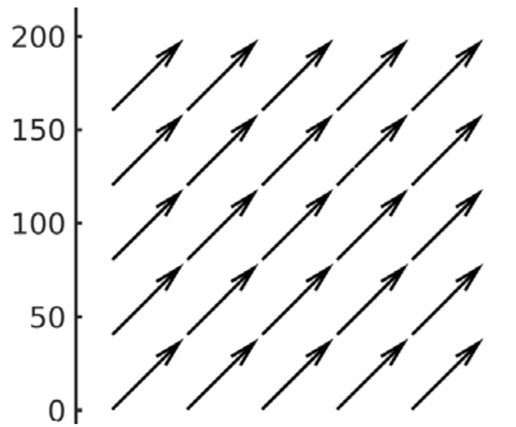
Image Translation: $I \circ \phi^{-1}(x)$

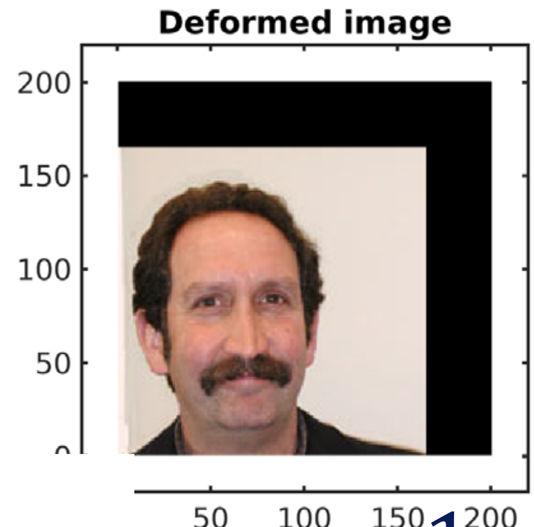




$$\phi(x) = x + \begin{pmatrix} -1 \\ -1 \end{pmatrix} \qquad v(x) = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$





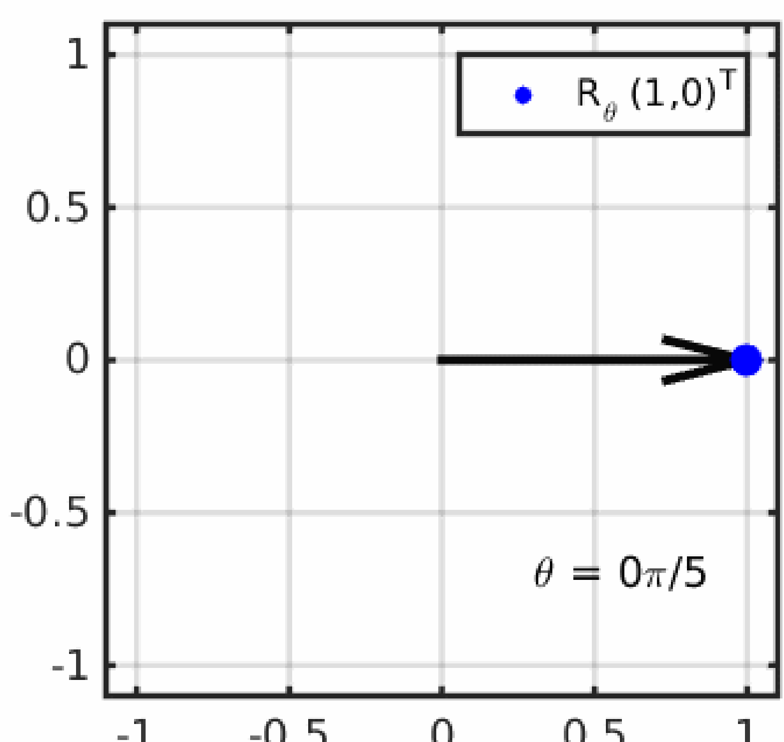


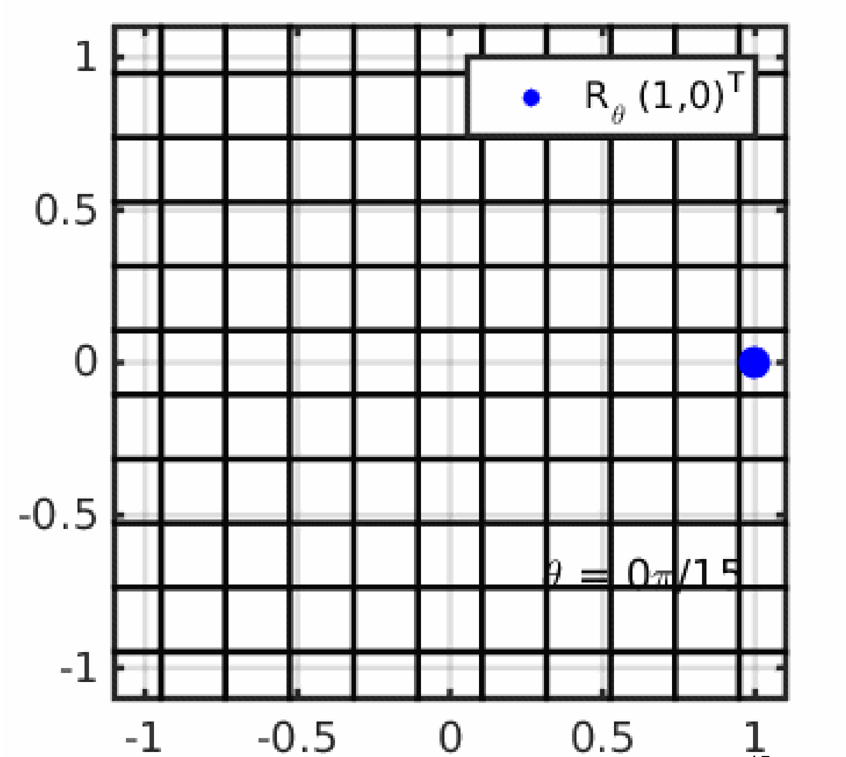
$$\phi^{-1}(x) = x + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad v(x) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad I \circ \phi^{-1}$$

Rotation Coordinates Counter Clockwise

Counter Clockwise

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

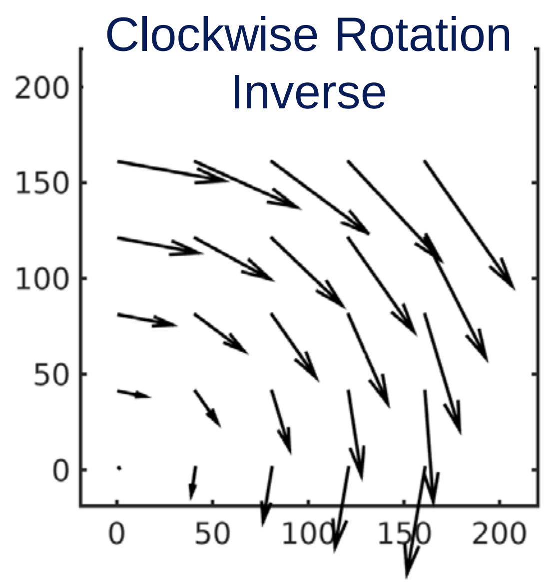




Images via Template Transformation

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

I



$$v(x) = \phi_{45}^{-1}(x) - x$$

$$I \circ \phi_{45}^{-1}$$

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

The Orbit of Images in Computational Anatomy

We call the

−ideal,protoype,exemplar,template, atlas−

$$I_{temp}$$

We call the set of images

-the Orbit-

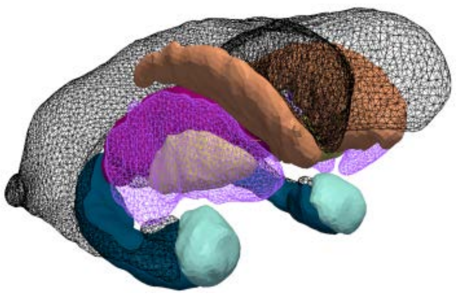
$$I = I_{temp} \circ \phi^{-1}$$

The Imaging Model for Matrix Group

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

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

$$\phi(x) \xrightarrow{\text{input}} \boxed{\text{Non-Linear Observer Equation}} \xrightarrow{\text{output}} I_{temp} \circ \phi^{-1}(x)$$



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

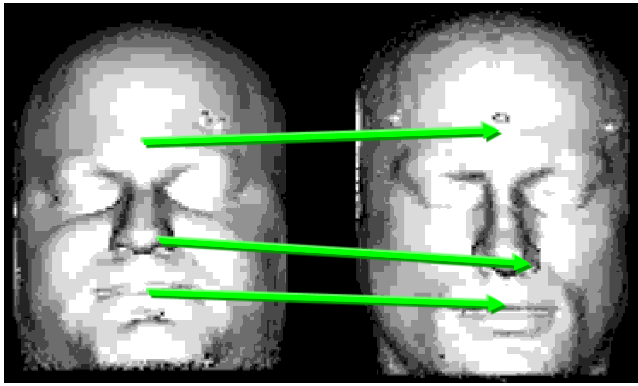
Estimating Matrix Transformations

Daniel Tward:

- Program for Manual Transformation

Derivation of Transformations

Translation Registration:2D



- Given landmark B.C.'s

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

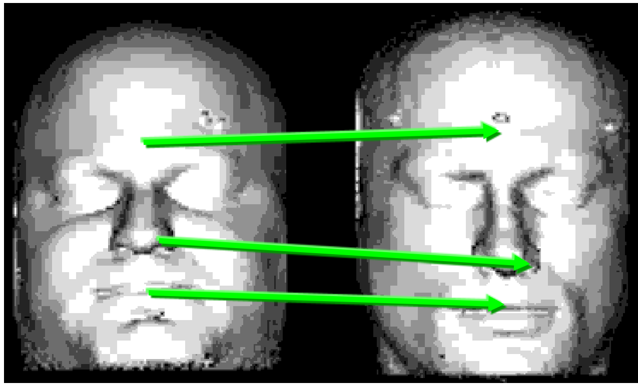
- Translation 2 dimensions

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

- Problem: Estimate Translation

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

Matrix Registration



- Given landmarks

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

- 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

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

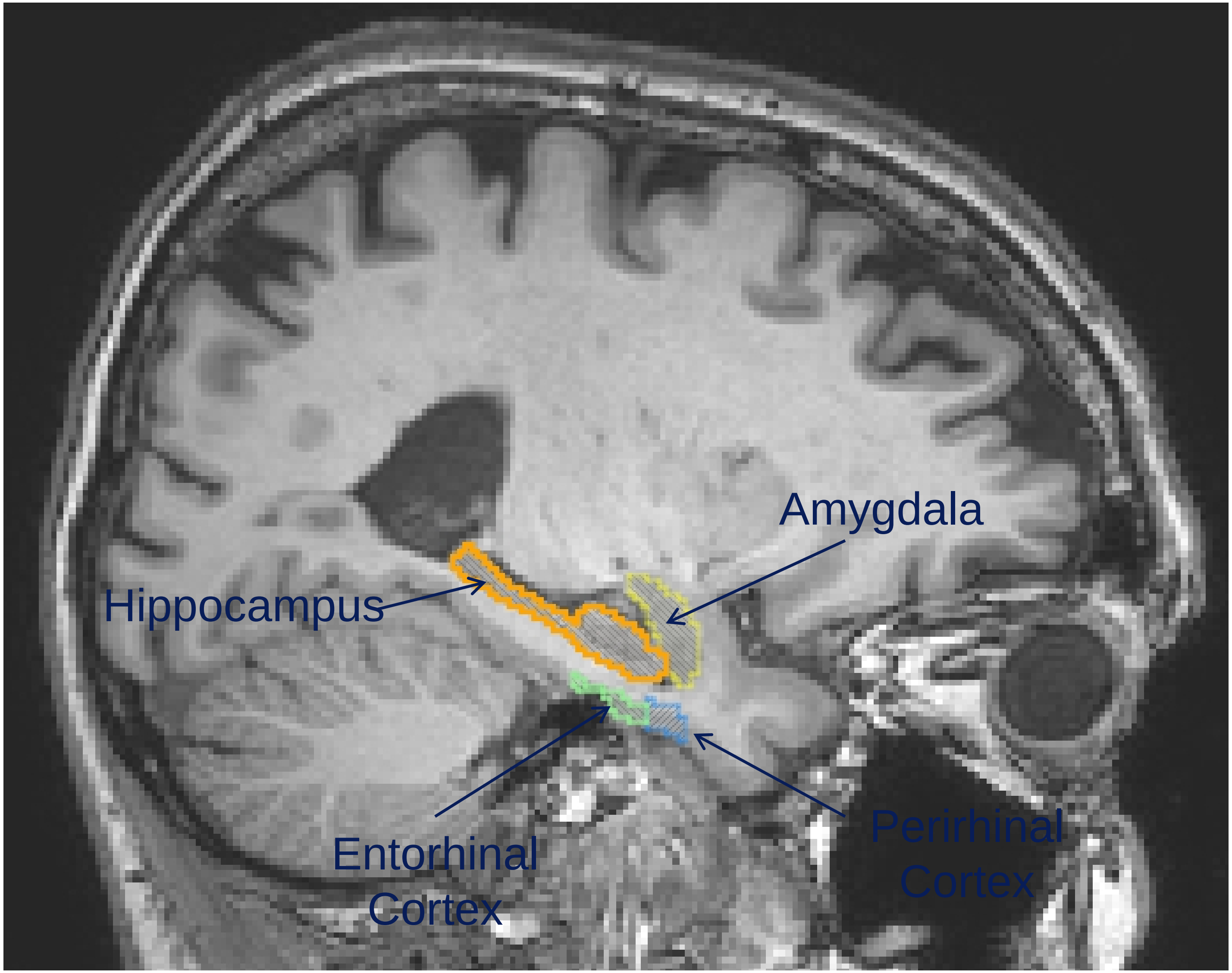
Represent Matrix as column vector rewriting minimization

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

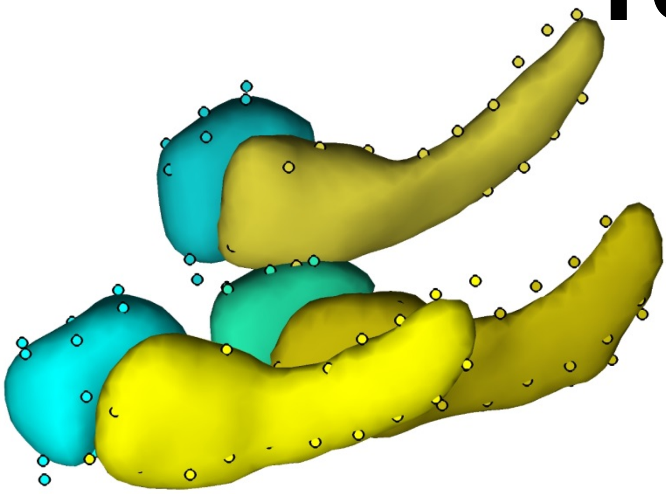
$$\min_{a \in \mathbb{R}^4} \sum_i |y_i - X_i a|^2$$

Prove:

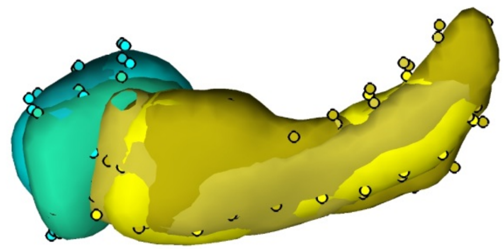
$$\hat{a} = \underbrace{\left(\sum_i X_i{}^T X_i\right)^{-1}}_{4\times 4 \text{ matrix}} \underbrace{\left(\sum_i X_i{}^T y_i\right)}_{4\times 1 \text{ column}}$$


Amygdala
Hippocampus
Entorhinal
Cortex
Perirhinal
Cortex

Three amygdalas and hippocampi – before and after registration



Before registration



After rigid registration (rotation, translation, scale)(

Homework: Translation and Matrix Group Registration

Homework: Maximum Entropy

$$p(x)\ a\ density, \qquad \int_{-\infty}^{\infty} p(x)dx = 1$$

$$p(x) \geq 0, x \in (-\infty, \infty)$$

Given two constraints:

Zero-mean: $\int_{-\infty}^{\infty} xp(x)dx = 0$

Variance: $\int_{-\infty}^{\infty} xp(x)dx = \sigma^2$

Prove the density maximizing entropy subject to mean and variance constraint is

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Homework: Rotations

- Show the rotations form a group. We have already shown inverse, and identity is identity matrix.

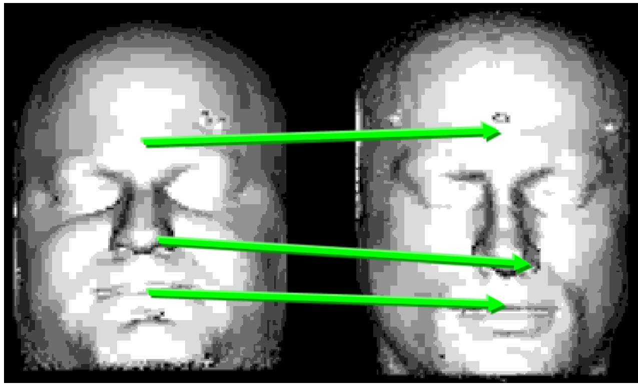
Show composition which is matrix multiplication generates a rotation matrix:

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

Show $\phi_\theta \circ \phi_{\theta'} = \phi_\gamma \ for \ some \ \gamma$

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

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

- Problem: Estimate Scale

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

Homework cont'd:

Manual Adjustment of Hippocampus and Amygdala Anatomical Coordinate Systems Tool Matlab Daniel Tward

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.

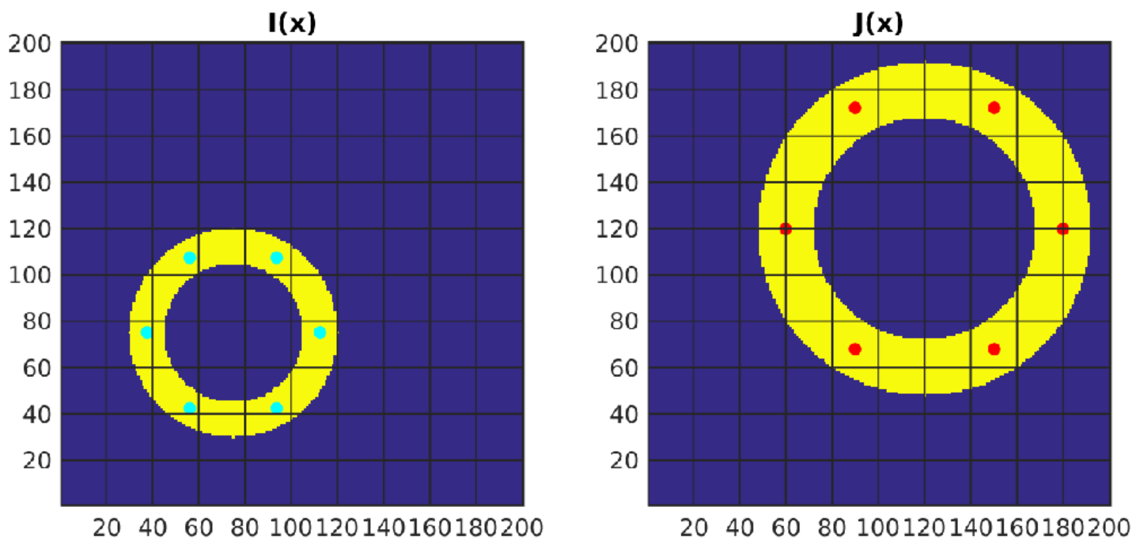


Figure 1: Images of two annuli with 6 corresponding labelled landmarks.

Homework cont'd

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.

```
% 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 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
```

Homework cont'd

1.2 Record landmarks

Estimate the position of the 6 landmarks on I, and store them in the (2 rows by 6 columns) variable `X`. Estimate the position of the 6 corresponding landmarks on J, and store them in the (2 rows by 6 columns) variable `Y`.

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

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

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 X

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

Comment on the proximity of `sX` and `Y`. Do they match exactly? Would you expect them to?

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

Homework cont'd

1.5 Transform the image "naively" $I(x) \mapsto I(sx)$

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

Center for IMAGING SCIENCE

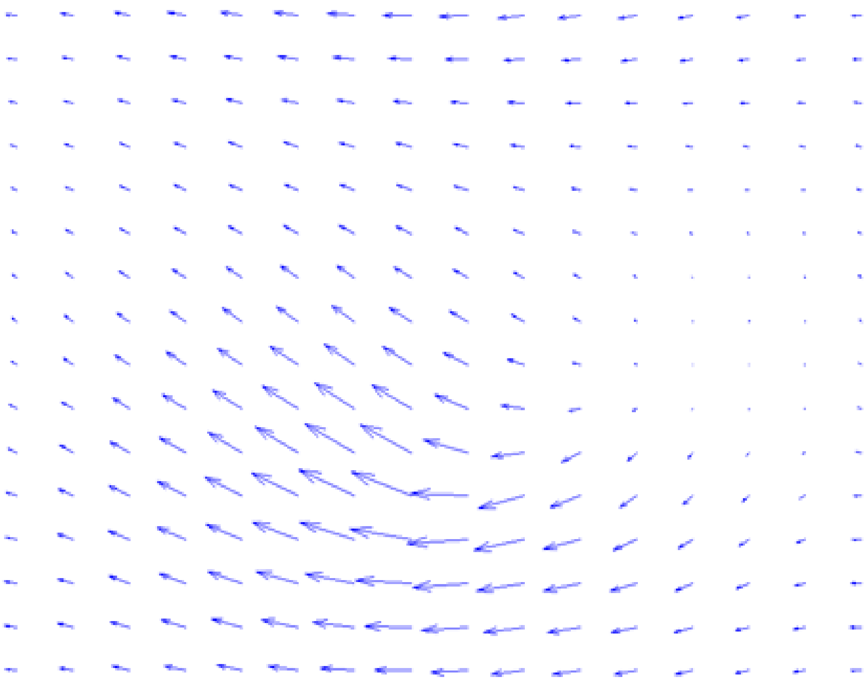
JOHNS HOPKINS UNIVERSITY

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

Vector Fields

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

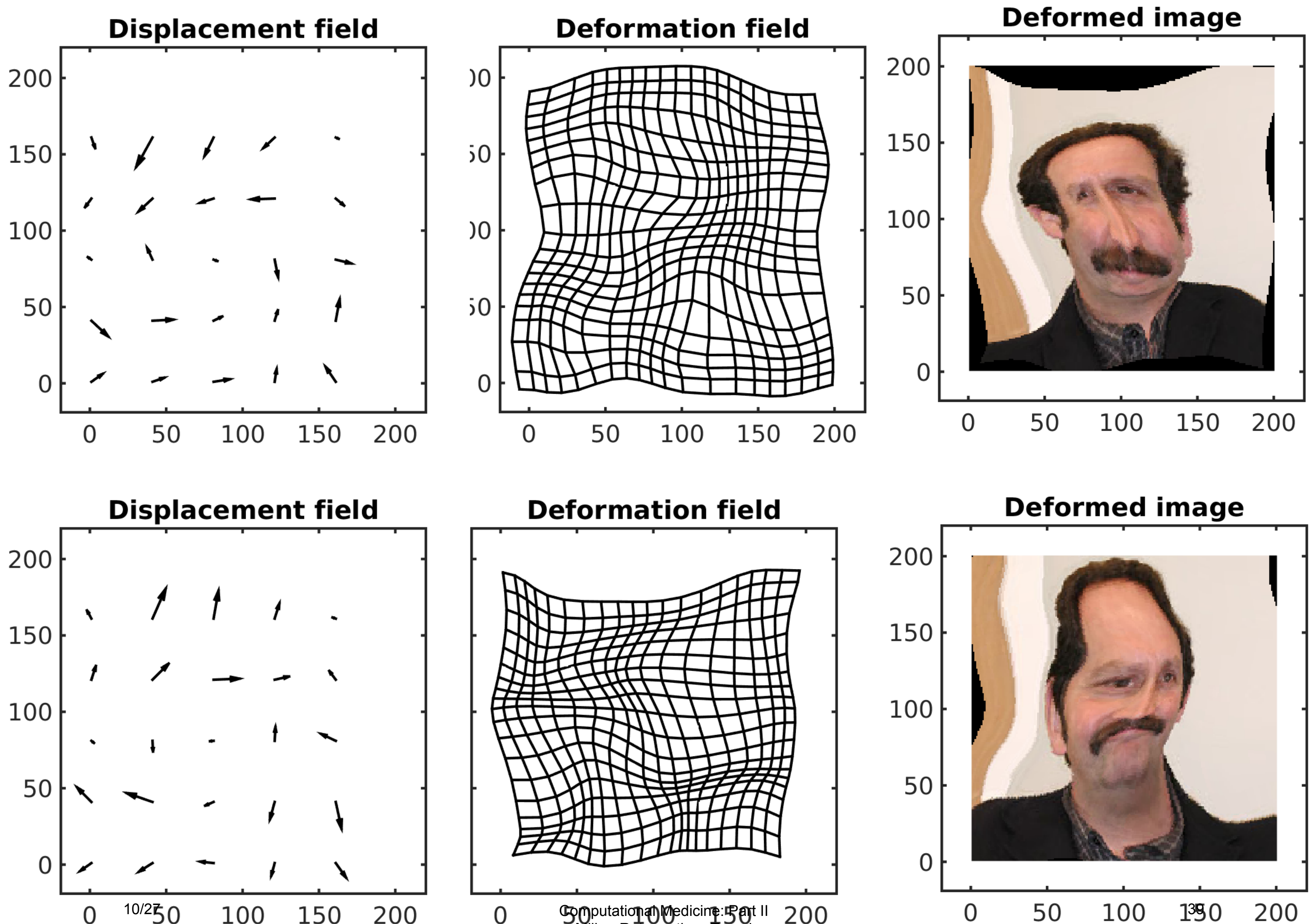
$$\phi(x) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \underbrace{\begin{pmatrix} v_1(x) \\ v_2(x) \end{pmatrix}}_{\substack{vector \\ field}}$$



16×16

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

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 ϕ_t, Velocity $v_t = \dot{\phi}_t$

Spring-Mass $m\ddot{\phi}_t + k\phi_t = 0$

$$\phi_t = \Phi_{init} \cos\left(\sqrt{k/m}\, t + \theta\right)$$

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

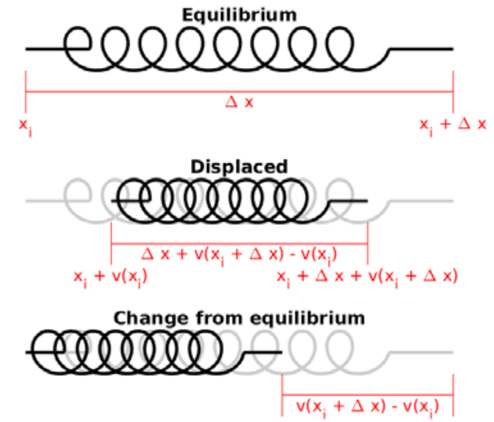
Charge q_t, Current $i_t = \dot{q}_t$

Kirchoff: $L\ddot{q}_t + \frac{1}{C} q_t = 0$

$$q_t = Q_{init} \cos(1/\sqrt{LC}\, t + \theta)$$

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

Energy of a Distributed Spring System



Spring Position:

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

Potential Energy:

$$\lim_{\Delta x \to 0} \sum_i \frac{1}{2} \frac{k}{\Delta x} \left| v(x_i + \Delta x) - v(x_i) \right|^2 \simeq \lim_{\Delta x \to 0} \sum_i \frac{1}{2}\, k \, |v'(x_i)|^2 \Delta x$$

$$= \frac{1}{2} \int k \left| \frac{dv(x)}{dx} \right|^2 dx$$

$$= \frac{1}{2} \int Av \cdot v \, dx + B.C.'s\,, \qquad A = -k \frac{\partial^2}{\partial x^2}$$

Show Integration by Parts

3D Energies from Continuum Mechanics

Energy: $$\frac{1}{2}\int_{\mathbb{R}^3}(Av)\cdot v dx$$

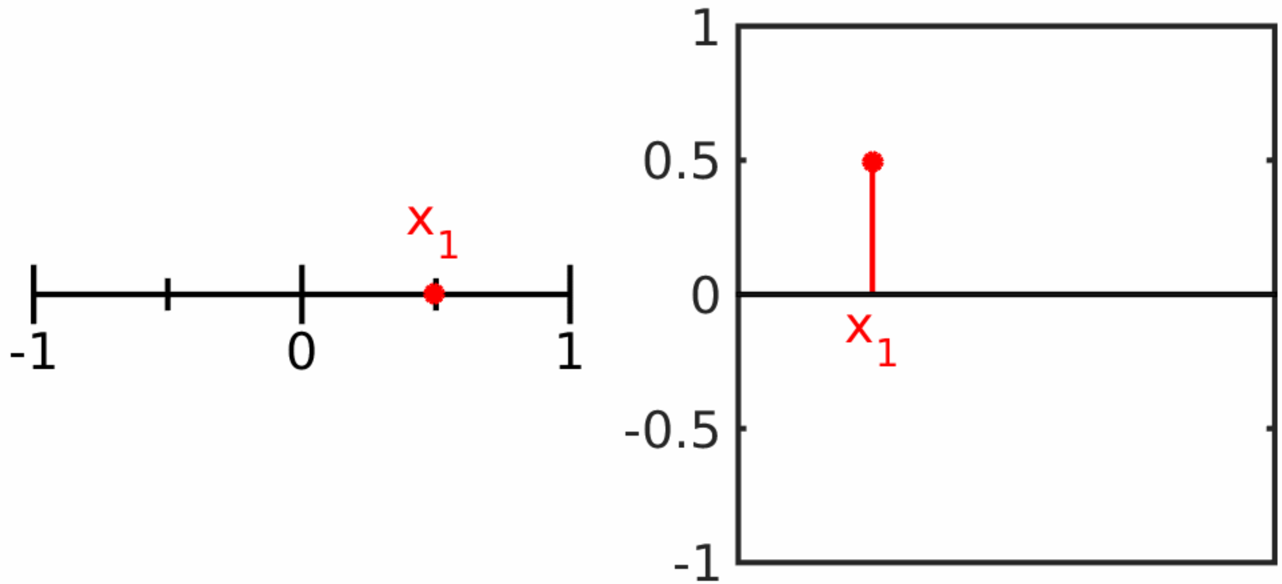
$$A = \sum_n a_n \frac{\partial^{p_n+q_n+r_n}}{\partial x_1^{p_n}\partial x_2^{q_n}\partial x_3^{r_n}}$$

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

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

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

Minimizing Energies via Calculus of Variations


x1
-1
0
1
1
0.5
0
-0.5
-1
x1

Maximizing or Minimizing

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

Vector: $$f = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \in R^n, h = \begin{pmatrix} h(x_1) \\ \vdots \\ h(x_n) \end{pmatrix} \in R^n$$

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

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

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

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

Scalar: $J(x) = x^2,\ J(\epsilon) = (x + \epsilon h)^2, h \in R$

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

Vector: $f = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \in R^n, h = \begin{pmatrix} h(x_1) \\ \vdots \\ h(x_n) \end{pmatrix} \in R^n$

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

$$= 2 \sum_i f(x_i) h(x_i) = 0$$

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

Vector: $f = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \in R^n, h = \begin{pmatrix} h(x_1) \\ \vdots \\ h(x_n) \end{pmatrix} \in R^n$

Energy: $J = \frac{1}{2} Af \cdot f = \frac{1}{2} f \cdot Af$

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

Minimizing Directional Derivative Condition:

$$\frac{d}{d\epsilon} J(\epsilon)|_{\epsilon=0} = Af \cdot h = 0$$

Proof:

Vector: $f = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} \in R^n, h = \begin{pmatrix} h(x_1) \\ \vdots \\ h(x_n) \end{pmatrix} \in R^n, A = A^T$

Energy: $J = \frac{1}{2} Af \cdot f = \frac{1}{2} f \cdot Af$

Prove: $\frac{d}{d\epsilon} J(\epsilon)|_{\epsilon=0} = Af \cdot h = 0$

$$\frac{d}{d\epsilon} J(\epsilon) = \frac{d}{d\epsilon} \frac{1}{2} A(f + \epsilon h) \cdot (f + \epsilon h)$$

$$= \frac{1}{2} Ah \cdot (f + \epsilon h) \Big|_{\epsilon=0} + \frac{1}{2} A(f + \epsilon h) \cdot h \Big|_{\epsilon=0}$$

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

$$Ah \cdot f = h^T A^T f = h^T Af = h \cdot Af = Af \cdot h$$

Minimizing Quadratic Energy

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

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

$$f^{\epsilon}(x) = f(x) + \epsilon h(x)$$

Minimizing Directional Derivative Condition:

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

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

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

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

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

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

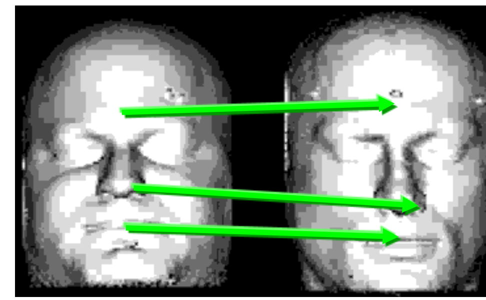
$$= \frac{1}{2}\int_X h \cdot Af \, dx + \ \frac{1}{2}\int_X Af \cdot h \, dx = \int_X Af \cdot h \, dx$$

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: $$y_i = x_i + v(x_i), i = 1, \ldots, n$$
$$\begin{pmatrix} y_{1i} \\ y_{2i} \end{pmatrix} = \begin{pmatrix} x_{1i} \\ x_{2i} \end{pmatrix} + \begin{pmatrix} v_1(x_i) \\ v_2(x_i) \end{pmatrix}$$

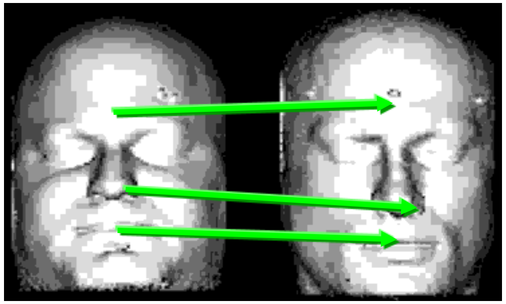


$$\min_{v(x), x \in \mathbb{R}^2} \sum_i |y_i - (x_i + v(x_i))|^2$$

- Solution:

$$v(x_i) = y_i - x_i$$
$$\hat{v}(x) = anything, for\ all\ x \neq x_i, i = 1, \ldots, n$$

Smooth vector field as variational problem.



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

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

$$\text{Green's kernel} \quad Ak = \delta$$

- Solution: A Superposition of Spline Green's Kernels

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

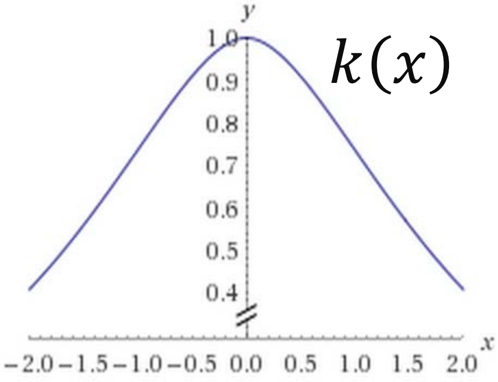
In Physics, the inverse of A is called the Green's function or the Green's kernel : it "splines" the landmark boundary conditions.

$$Ak = \delta$$

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

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

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



Green's Function (as an impulse response)

1D differential equation: $t \in \mathbb{R}^1$

No-Smoothness: $\dot{y}(t) + \alpha y(t) = x(t), y(-\infty) = 0$

- $A = \frac{d}{dt} + \alpha, \quad k(t) = e^{-\alpha t} u_s(t), \quad Ak(t) = \delta(t)$

$$-\dot{y}(t) + \alpha y(t) = x(t), y(\infty) = 0$$

- $A = -\frac{d}{dt} + \alpha, \; k(t) = e^{\alpha t} u_s(-t), \; Ak(t) = \delta(t)$

Continuity:

- $A = -\frac{d^2}{dt^2} + 1, \; k(t) = \frac{1}{2} e^{-|t|}, \quad Ak(t) = \delta(t)$

1-Derivative:

- $A = \left(-\frac{d^2}{dt^2} + 1\right)^2, \; k(t) = (1 + |t|)e^{-|t|}$

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:

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

Splines as Minimum Energy (cont'd)

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

$$J(v) = \frac{1}{2}\int_X \big(Av(x)\big) \cdot v(x)dx + \underbrace{\sum_i p_i \cdot (y_i - x_i + v(x_i))}_{Lagrange-Multipliers}$$

$$v \to v + \epsilon h \qquad \frac{d}{d\epsilon} J(v+\epsilon h)\Big|_{\epsilon=0}$$

$$= \frac{d}{d\epsilon}\Bigg(\frac{1}{2}\int_X A(v(x)+\epsilon h(x)) \cdot (v(x)+\epsilon h(x))dx$$

$$+ \sum_i \int_X p_i \cdot (y_i - x_i - v(x) - \epsilon h(x))\delta(x-x_i)\, dx\Bigg)\Bigg|_{\epsilon=0}$$

$$= \int_X \underbrace{(Av(x) - \textstyle\sum_i \delta(x-x_i)p_i)}_{=0} \cdot h(x)dx$$

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

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

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

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

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 $x \in \mathbb{R}^3$

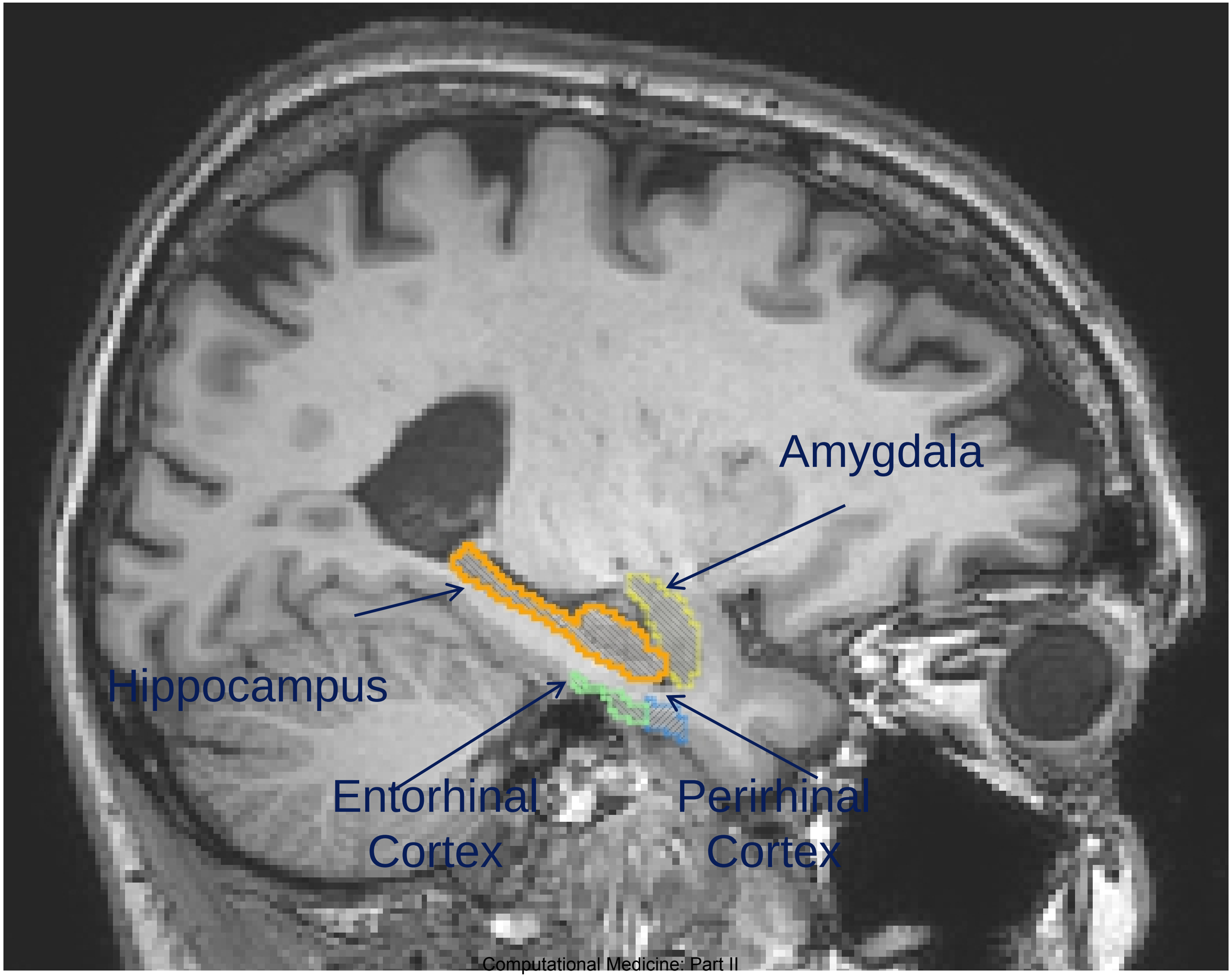
Splines smooth: $\phi(x) = x + v(x)$

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

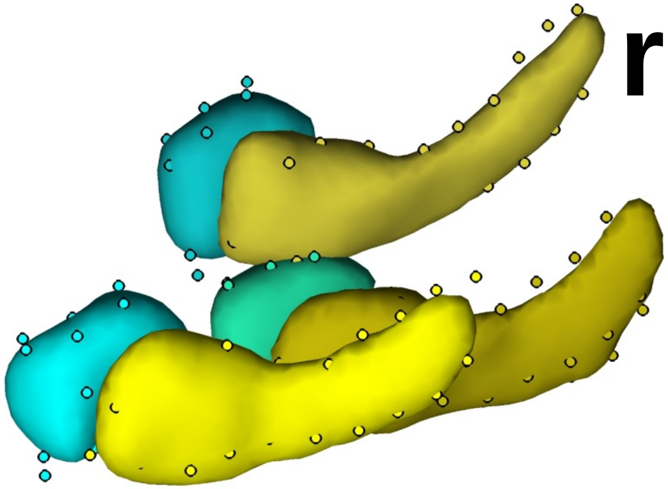
Matrix Notation:

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

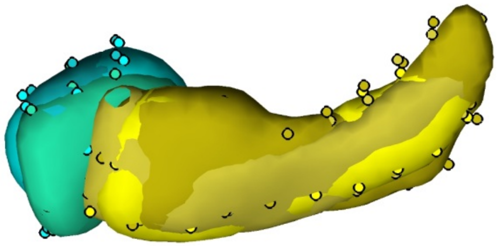
Landmark Based Splines



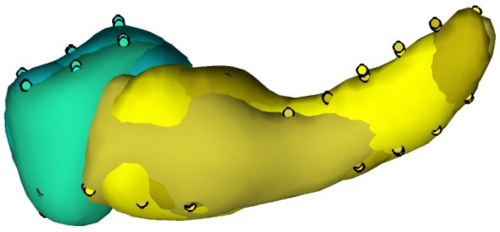
Three amygdalas and hippocampi – before and after registration



Before registration

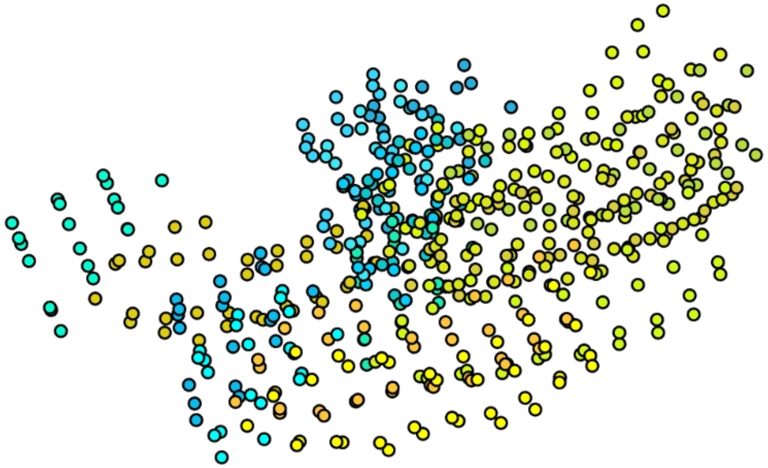


After rigid registration (rotation, translation, scale)

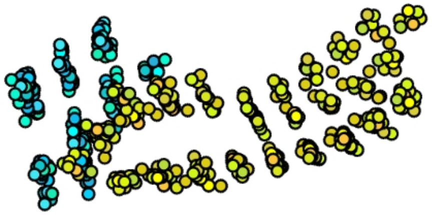


After non-rigid (Landmark matching actually via splines)

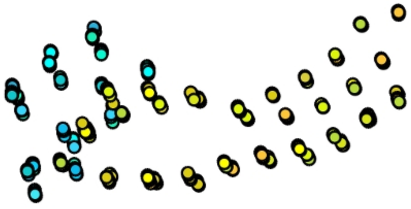
Nine amygdalas and hippocampi – before and after registration landmarks



Before registration

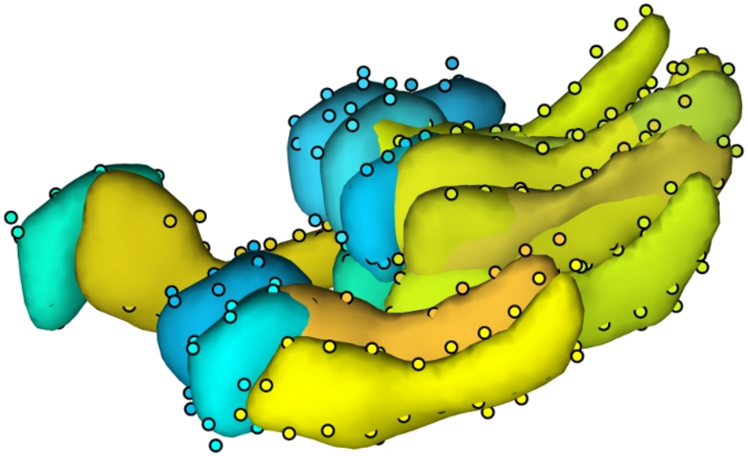


After rigid registration (rotation, translation, scale)

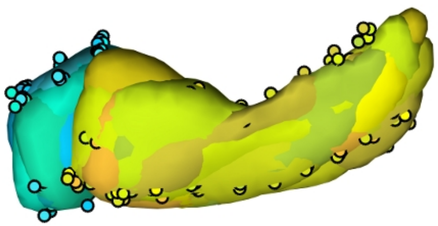


After non-rigid (Landmark matching actually via splines)

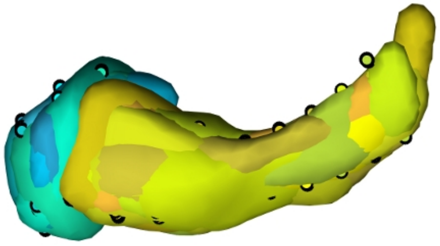
Nine amygdalas and hippocampi – before and after registration



Before registration



After rigid registration (rotation, translation, scale)



After non-rigid (Landmark matching actually via splines)

Solve some Green’s Kernel Examples

Daniel Tward

Circuits

Differential equations

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

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

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

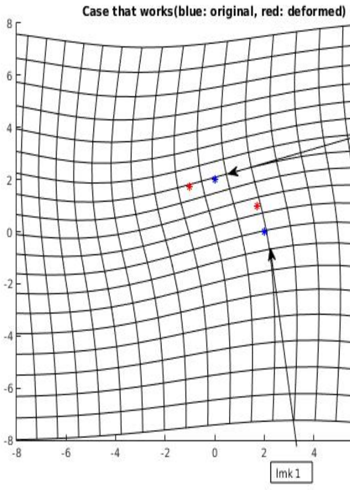
$$A = -\frac{d^2}{dx^2} + id \text{ with } Ak(x) = \delta(x)$$

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

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

Derive matrix equation for p's:

Gaussian Kernels:



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

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

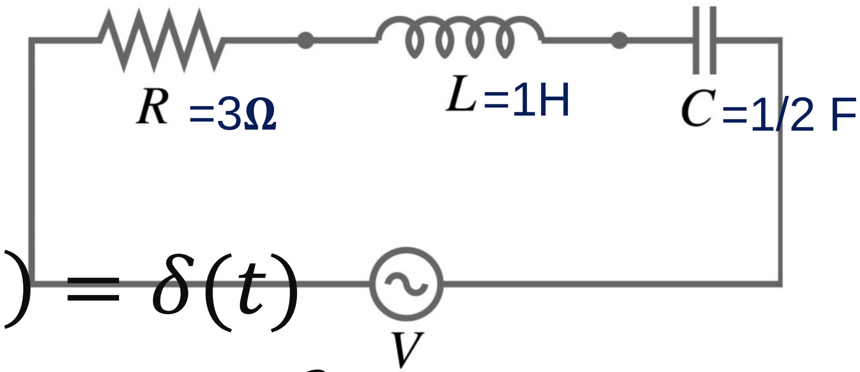
$$v_2(x) = \sum_i \frac{1}{2\pi\sigma^2} e^{\frac{-|x-x_i|^2}{2\sigma^2}} p_{2i}$$

- Boundary Conditions:

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

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

Circuit Examples: Impulse Response



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

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

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

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

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

Prf.: $\dot{k}(t)=(2e^{-2t}-e^{-t})u_s(t)+(e^{-t}-e^{-2t})\delta(t)$

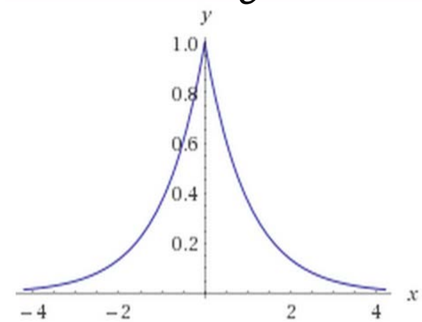
$$\ddot{k}(t)=(e^{-t}-4e^{-2t})u_s(t)+(2e^{-2t}-e^{-t})\delta(t)$$

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

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

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

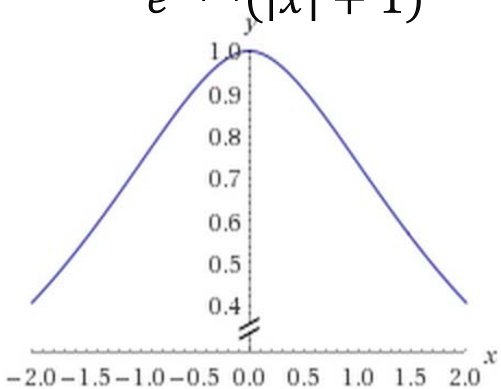
$e^{-|x|}$



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

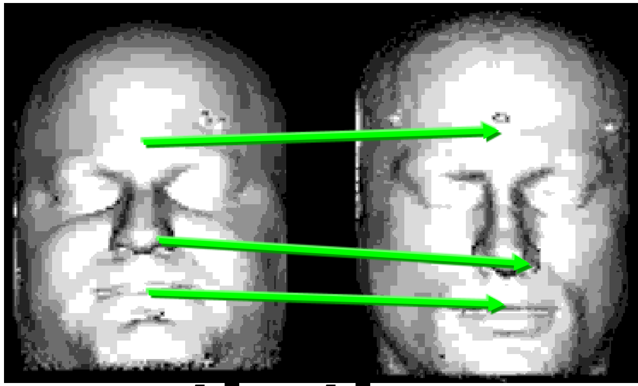
$$k(x) = e^{-|x|}(|x| + 1)$$
$$\text{with } Ak = \delta$$

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



Homework:

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



- Problem: Penalize Vector Field with Energy

$$\min_{v(x),x\in R^d} \frac{1}{2}\int_X v\cdot v dx + \sum_i |y_i - (x_i + v(x_i))|^2$$

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

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

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

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

HOMEWORK

Computational Homework 2

2 Calculating Linear Transformations and Jacobians

In this set of exercises you will calculate transformations to match one pair of landmarks to another, visualize the transformation as a grid, and calculate the Jacobian. You will demonstrate that that the model $\phi(x) = x + v(x)$ may result in transformations which do not have an inverse.

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

2.1 Generate a grid

We will calculate the value of our transformations φ at each point on a grid. Generate a grid as in the previous homework.

```
% 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
```

Homework

2.2 Generate the landmarks

Start with a pair of landmarks at locations $(125, 100)$ and $(150, 50)$. Store each landmark as a column in the variable `X`.

For $\theta = 30$ degrees, generate a rotation matrix stored in the variable `R`.

Define a pair of target landmarks `Y` by rotating `X` by 30 degrees.

2.3 Plot the landmarks and grid

Show the landmarks `X` as a scatterplot in cyan. Show the landmarks `Y` in the same scatterplot in red. Draw an untransformed grid in the same plot as follows.

```
down = 10; % downsampling is important so you can see things clearly
xijdown = xij(1:down:end,1:down:end);
yijdown = yij(1:down:end,1:down:end);
% This is a trick to plot a grid easily.
% We actually plot a 3D surface,
% but view it directly from above so it looks 2D
surf(xijdown,yijdown,ones(size(xijdown)),'facecolor','none','edgecolor','k');
```

Homework

2.4 Calculate an optimal 2×2 matrix transformation

Find the $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, a 2×2 matrix, which brings `X` to `Y` by minimizing the error

$$E = \sum_{i=1}^{2} |AX(i) - Y(i)|^2 = \sum_{i=1}^{2} \sum_{j=1}^{2} |\sum_{k=1}^{2} a_{jk} X_k(i) - Y_j(i)|^2 \tag{1}$$

Minimize this error analytically. You should be able to derive a matrix equation whose solution gives the optimal A.

In matlab, numerically calculate A for these landmarks.

Note that you should already know what the optimal A is for this example (what is it?), use this to check your work.

2.5 Plot the transformed landmarks and grid

Transform your landmarks `X` by left multiplying with `A`. Show the transformed landmarks, `AX`, as a scatterplot in blue. Show the landmarks `Y` in the same scatterplot in red. Draw a transformed grid in the same plot.

```
Axij = % ... you should implement this
Ayij = % ... you should implement this
Axijdown = Axij(1:down:end,1:down:end);
Ayijdown = Ayij(1:down:end,1:down:end);
surf(Axijdown,Ayijdown,ones(size(Axijdown)),'facecolor','none','edgecolor','k');
```

Homework

2.6 Calculate the Jacobian

Use matlab's `gradient` function to calculate the Jacobian of this transformation (`Axij` and `Ayij`) and its determinant everywhere on the 200×200 grid. Visualize the Jacobian determinant as an image with a colorbar.

What do you expect the value of the Jacobian determinant to be? You should use this to check your work.

2.7 Calculate an optimal Gaussian kernel transformation

Chose the standard deviation $\sigma = 50$ for this exercise.

We calculate a displacment vector field of the form

$$
\begin{aligned}
v_1(x) &= \sum_{i=1}^{2} \exp\left(-\frac{1}{2\sigma^2}|x - X(i)|^2\right) p_1(i) \\
v_2(x) &= \sum_{i=1}^{2} \exp\left(-\frac{1}{2\sigma^2}|x - X(i)|^2\right) p_2(i)
\end{aligned} \tag{2}
$$

while satisfying boundary conditions.

The boundary conditions

$$
\begin{aligned}
v_1(X(1)) = Y_1(1) - X_1(1) &= \sum_{i=1}^{2} \exp\left(-\frac{1}{2\sigma^2}|X(1) - X(i)|^2\right) p_1(i) \\
v_1(X(2)) = Y_1(2) - X_1(2) &= \sum_{i=1}^{2} \exp\left(-\frac{1}{2\sigma^2}|X(2) - X(i)|^2\right) p_1(i)
\end{aligned} \tag{3}
$$

Homework

can be written as a 2×2 matrix vector equation for the x component of the transformation, $V_1 = \hat{K} P_1$, where $\hat{K}$ is a 2×2 matrix, and V_1 and P_1 are 2×1 vectors storing x components of v and p respectively. Write out this equation for the x component of the $p(i)$. You should solve it analytically, and computationally in matlab.

Do the same for the y component of $p(i)$ by writing the boundary conditions

$$v_2(X(1)) = Y_2(1) - X_2(1) = \sum_{i=1}^{2} \exp\left(-\frac{1}{2\sigma^2}|X(1) - X(i)|^2\right) p_2(i)$$
$$v_2(X(2)) = Y_2(2) - X_2(2) = \sum_{i=1}^{2} \exp\left(-\frac{1}{2\sigma^2}|X(2) - X(i)|^2\right) p_2(i)$$

as a matrix equation.

2.8 Plot the transformed landmarks and grid

Transform your landmarks by adding $v(X(i))$ to them. Plot these in blue as a scatterplot. On the same plot, show `Y` as a scatterplot in red.

Calculate the transformation at every point on your grid.

```
% initialize to identity, we will add the displacment v
phix = xij;
phiy = yij;
for i = 1 : nY
    for j = 1 : nX
        % add the displacement for each p(k) in the sum
```

Homework

```
        for k = 1 : size(X,2) % number of landmarks
            Kij = % ... implement this, the kernel evaluated at (j,i) - X(k)
            phix(i,j) = phix(i,j) + % ... add the x component for p(k)
            phiy(i,j) = phiy(i,j) + % ... add the y component for p(k)
        end
    end
end
```

Visualize the deformed grid as above.

```
phixdown = phix(1:down:end,1:down:end);
phiydown = phiy(1:down:end,1:down:end);
surf(phixdown,phiydown,ones(size(phixdown)),'facecolor','none','edgecolor','k');
```

2.9 Calculate the Jacobian

Calculate the Jacobian of `phix` and `phiy` and its determinant as above. Visualize it as an image with a colorbar.

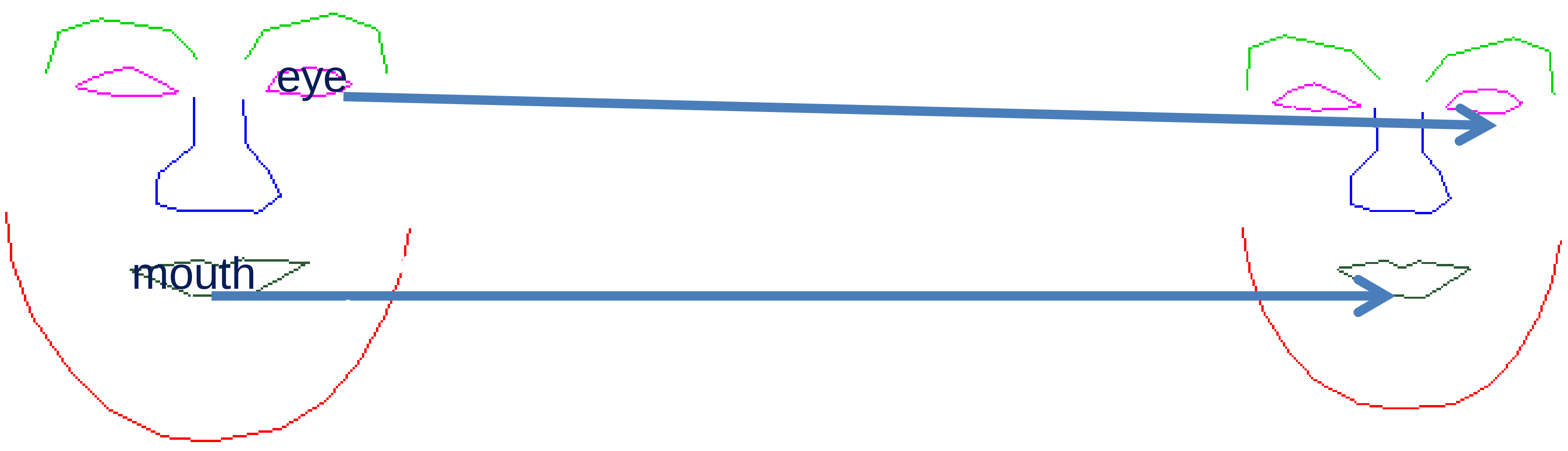
2.10 Repeat the exercise for $\theta = 45$ degrees

Describe what you notice about the deformed grid, the determinant of the Jacobian, and the invertibility of the transformation.

Image Analysis and Image Understanding Via Deformable Templates

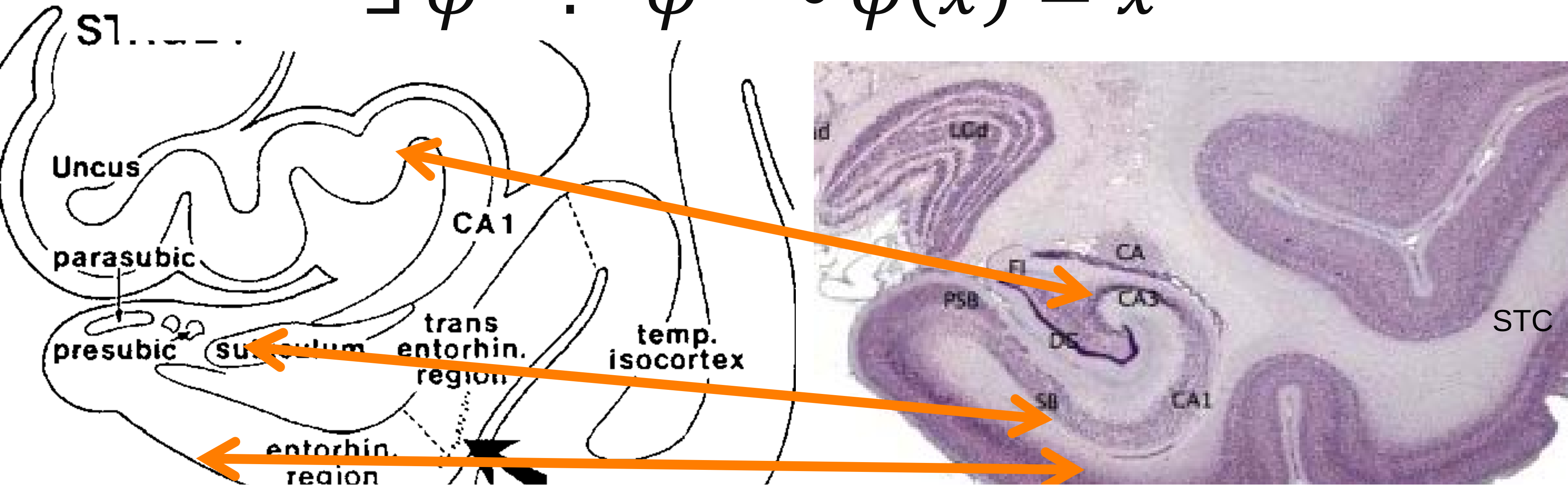
Michael I. Miller
Tilak Ratnanather
Daniel Tward

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



1-1 ONTO requires an inverse

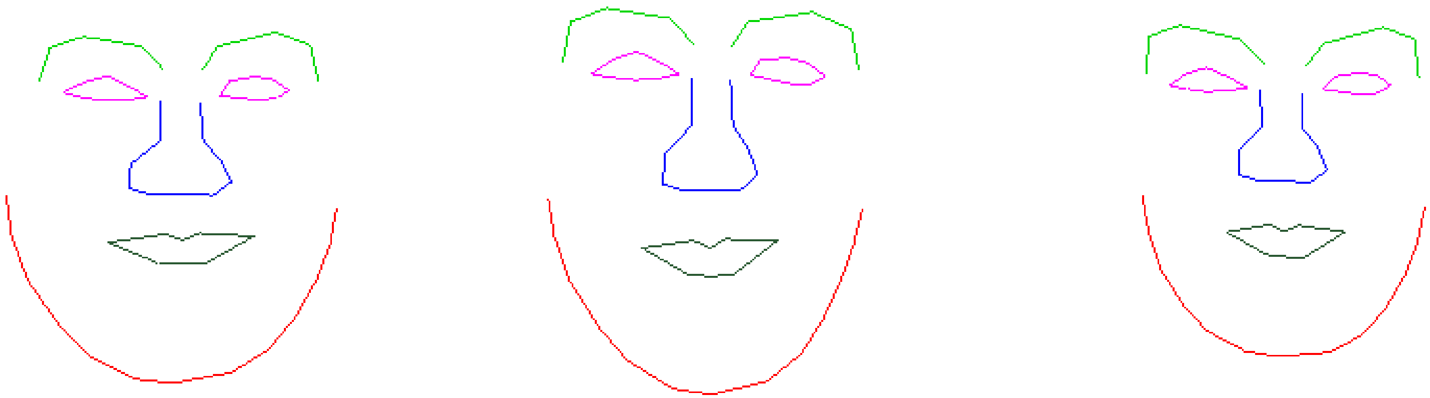
$$\exists\, \phi^{-1}: \quad \phi^{-1} \circ \phi(x) = x$$

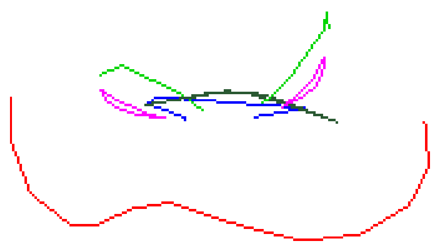


Inverse properties are determined by Jacobian of transformation

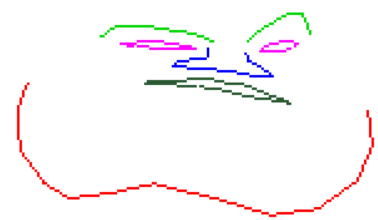
In $\mathbb{R}^2$, Jacobian $\partial_X \phi \doteq \begin{pmatrix} \dfrac{\partial \phi_1}{\partial x_1} & \dfrac{\partial \phi_1}{\partial x_2} \\ \dfrac{\partial \phi_2}{\partial x_1} & \dfrac{\partial \phi_2}{\partial x_2} \end{pmatrix}$

One of these does not have a bijective correspondence with others



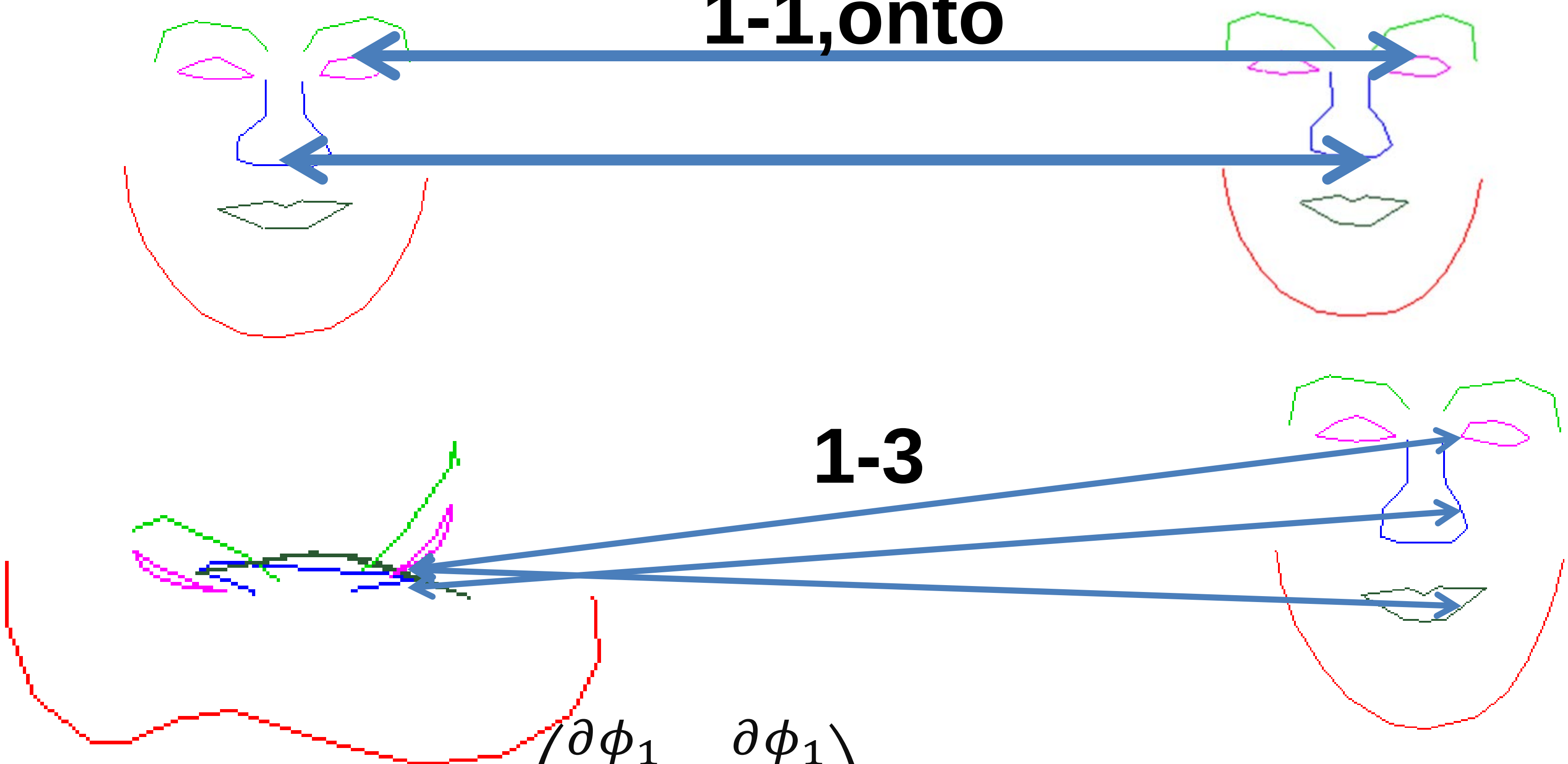


Not a bijection



Bijection

Neuroimage 2005
Vaillant, Miller, Trouvé & Younes



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

Neuroimage 2005
Vaillant, Miller, Trouvé & Younes

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

Matrix Groups and Inverse

Transform coordinates:

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

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

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

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

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

Matrix inverse implies 1-1 and onto.

ϕ: $\mathbb{R}^2 \to \mathbb{R}^2$ 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:

1-1: $\phi(x) = \phi(y) \Longrightarrow \phi(x-y) = A(x-y) = 0$

$$\Longrightarrow \; A^{-1} \circ A(x-y) = (x-y) = 0$$

Onto:

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

For Matrix Group: Matrix=Jacobian

Transform coordinates:

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

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

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

ϕ *is* $1-1$, *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

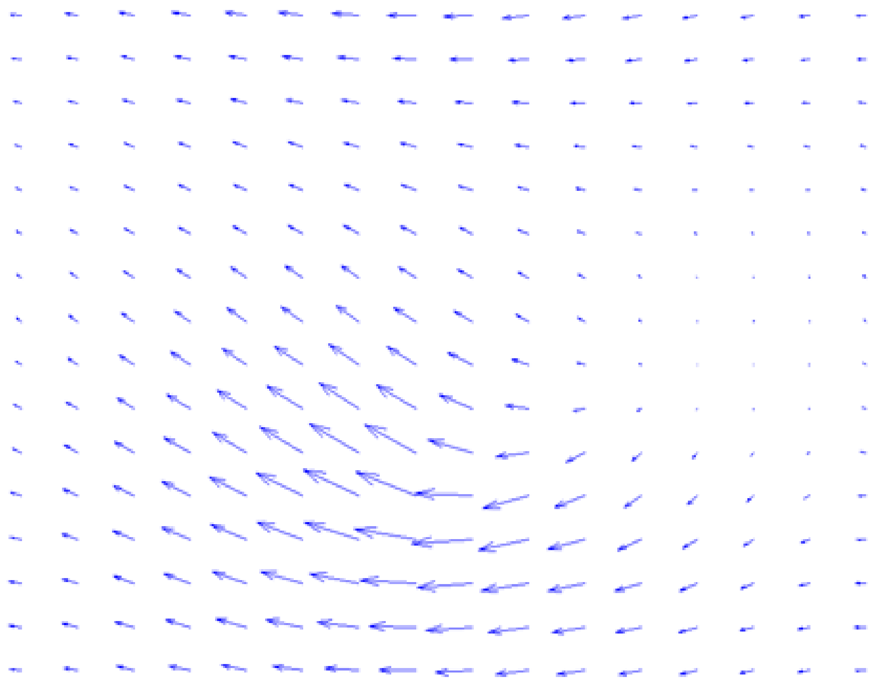
$$\phi: (x_1, x_2) \in \mathbb{R}^2 \to \mathbb{R}^2$$
$$\phi(x) = (\phi_1(x), \phi_2(x))$$

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

$$\phi_1(x) = x_1 + v_1(x)$$
$$\phi_2(x) = x_2 + v_2(x)$$

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

Taylor series

Scalar function of two-variables: $f(x_1, x_2)$

Vector field transformation:

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

Taylor series $f(\phi(x))$ at x:

$$f(x + v(x)) = f(x) + \underbrace{\left(\frac{\partial f}{\partial x_1}\bigg|_x v_1(x) + \frac{\partial f}{\partial x_2}\bigg|_x v_2(x) \right)}_{\partial_X f(x) v(x)} + o(v)$$

Taylor series

Vector function,two-variables: $\phi(x) = x + v(x)$

$$f(x) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, v(x) = \begin{pmatrix} v_1(x) \\ v_2(x) \end{pmatrix}$$

Taylor series $f(\phi(x))$ at x:

$$f(x + v(x))$$

$$= f(x) + \underbrace{\begin{pmatrix} \frac{\partial f_1}{\partial x_1}\Big|_x v_1(x) + \frac{\partial f_1}{\partial x_2}\Big|_x v_2(x) \\ \frac{\partial f_2}{\partial x_1}\Big|_x v_1(x) + \frac{\partial f_2}{\partial x_2}\Big|_x v_2(x) \end{pmatrix}}_{\partial_X f(x) v(x)} + o(v)$$

Example of Jacobian with no inverse

Vector field transformation:

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

Inverse (does not exist in general):

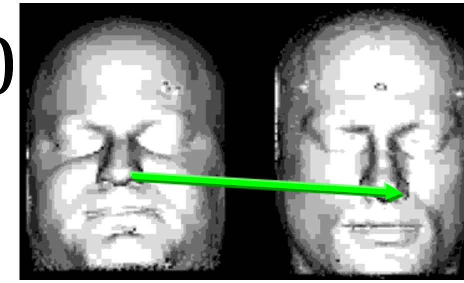
$$\phi^{-1}(x) = x - \partial_X(\phi^{-1})_{|x} v(x) + o(v)$$

Implicit Function Theorem

$$\partial_X(\phi^{-1})_{|x} = (\partial_X \phi)\Big|_{\phi^{-1}(x)}^{-1}$$

$$= (id + \partial_X v)\Big|_{\phi^{-1}(x)}^{-1}$$

1-landmark,Jacobian $|\partial_X \phi(x)| = 0$



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

$$\min_v \frac{1}{2}\int_{\mathbb{R}^2} (Av)\cdot v dx \ \text{ s.t. } \ \phi(x_1) = x_1 + v(x_1)$$

Solution: A Superposition of Spline Green's Kernels

$$v(x) = \underbrace{k(x - x_1)}_{\frac{1}{2\pi} e^{-\frac{1}{2}|x-x_1|^2}} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{2\pi} e^{-\frac{1}{2}|x-x_1|^2} p_1 \\ \frac{1}{2\pi} e^{-\frac{1}{2}|x-x_1|^2} p_2 \end{pmatrix}$$

Horizontal motion from origin: $\quad p = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad x_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$$v(x) = \frac{1}{2\pi} e^{-\frac{1}{2}|x|^2} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\partial_X \phi(x) = \partial_X\big(x + v(x)\big) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{2\pi} e^{-\frac{1}{2}|x|^2} \begin{pmatrix} -\frac{1}{2\pi} x_1 & -\frac{1}{2\pi} x_2 \\ 0 & 0 \end{pmatrix}$$

Neuroinformatics at the Radiomics Scale

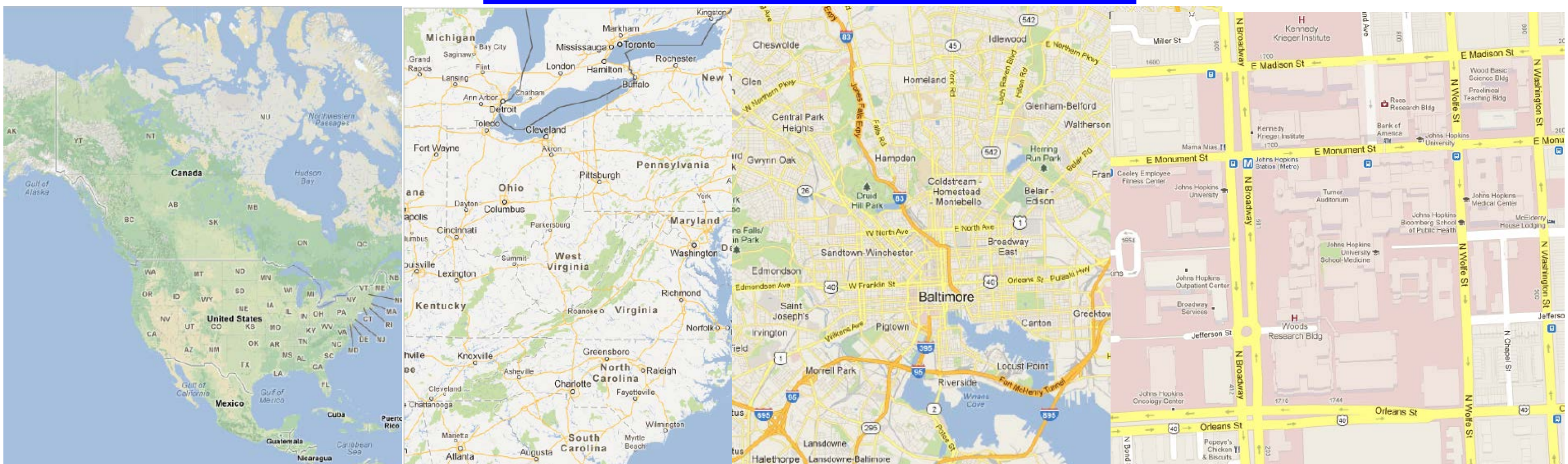
Michael I. Miller
Tilak Ratnanather
Daniel Tward

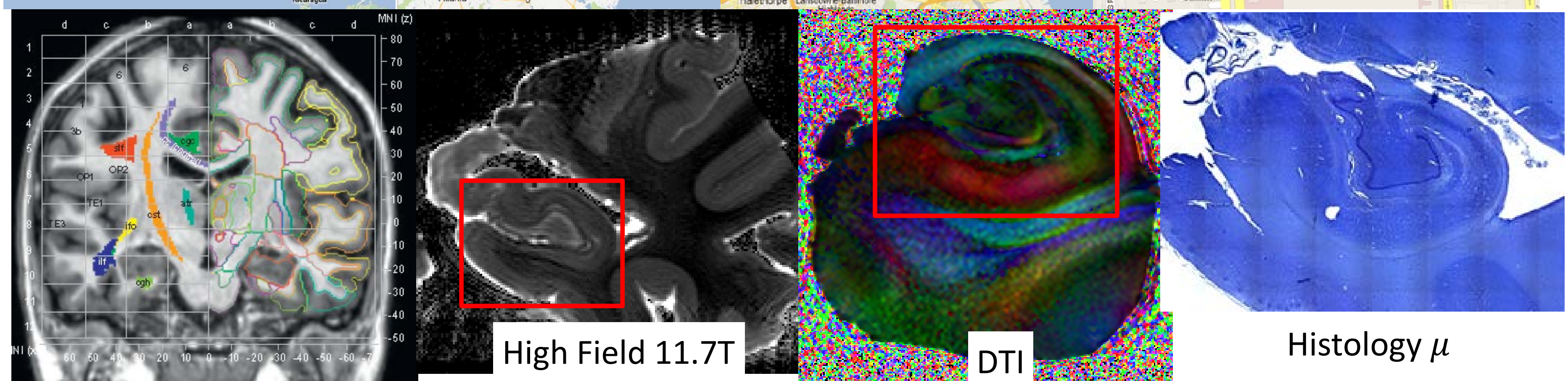
Center for IMAGING SCIENCE
JOHNS HOPKINS UNIVERSITY

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.

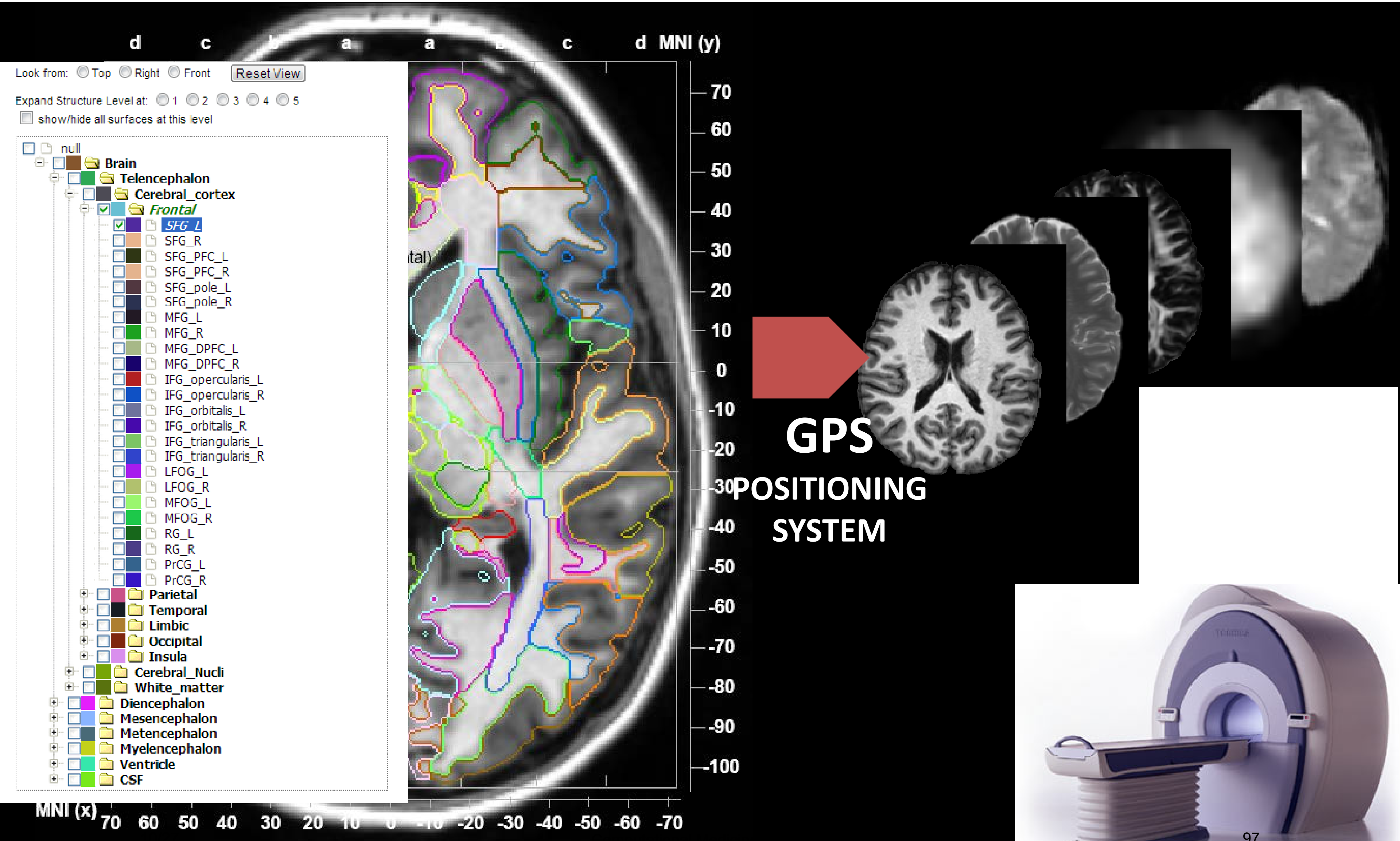
www.MRICloud.org





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

High Throughput Neuroinformatics



Miller, Faria, Oishi, Mori, High Throughput Neuroinformatics". Frontiers Neuroinfromatics,, 2013.

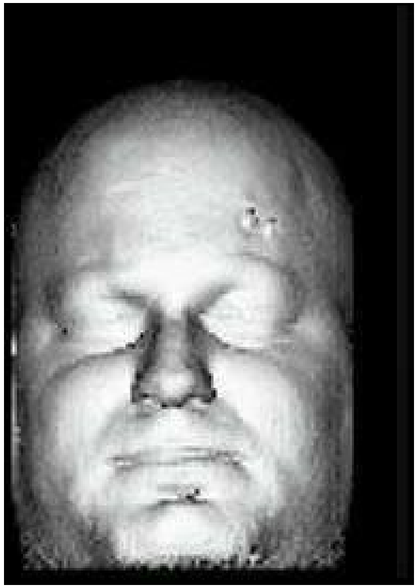
Positioning by Registering Coordinate Systems via Small Deformation Image Matching

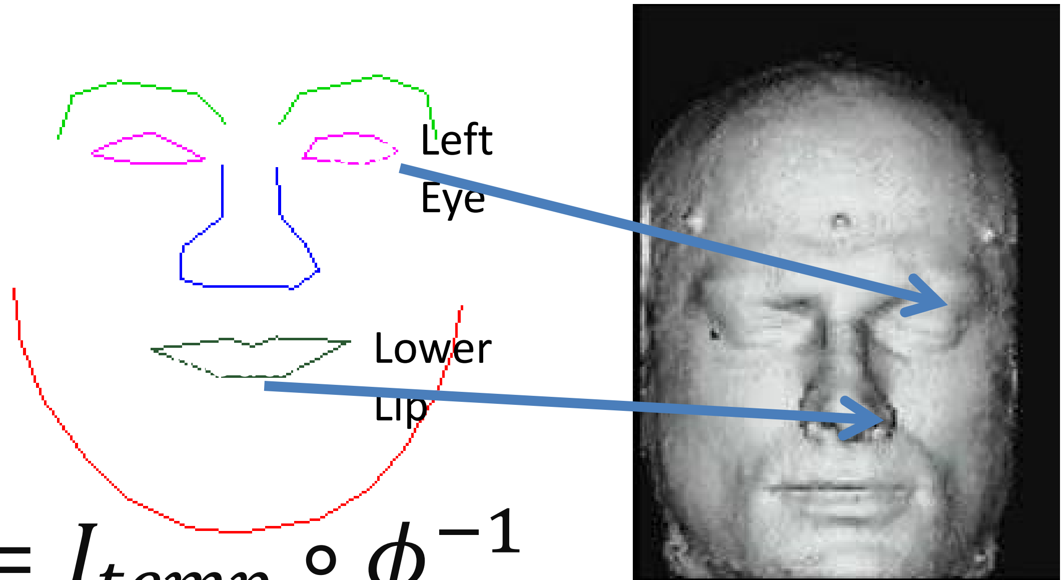
Michael I. Miller

Tilak Ratnanather

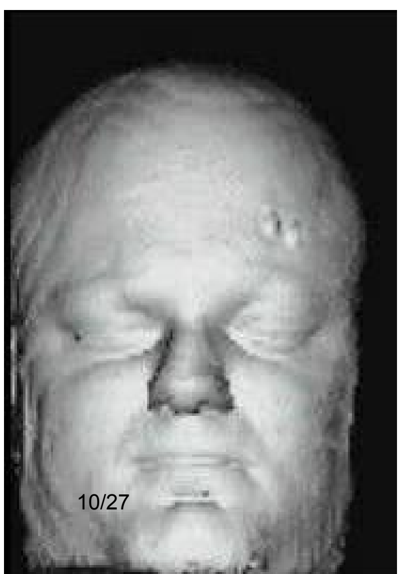
Daniel Tward

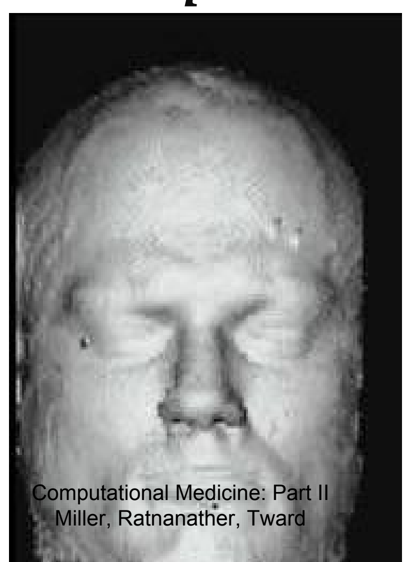
Anatomy is an element in the orbit; the vector field positions the information.

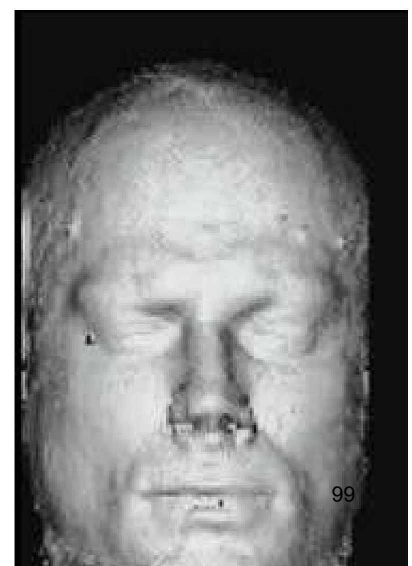




$$I = I_{temp} \circ \phi^{-1}$$







Daniel Tward:

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

Small Deformation Image Matching

$$\min_v \quad \frac{1}{2}\int_X Av \cdot v \, dx + \frac{1}{2}\int_X (I \circ \phi^{-1} - I')^2 \, dx$$

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

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

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

$$= \int_X Av \cdot w dx - \int_X (I \circ \phi^{-1} - I') \cdot \underbrace{\nabla I \Big|_{\phi^{-1}} \cdot \partial_X \, (\phi^{-1}) w}_{\nabla(I \circ \phi^{-1}) \cdot w} \, dx$$

$$= \int_{\mathrm{X}} (Av - (I \circ \phi^{-1} - I')\nabla(I \circ \phi^{-1})) \cdot w \, dx$$

- Necessary Condition:

$$Av(x) = \left(I \circ \phi^{-1}(x) - I'(x)\right)\nabla(I \circ \phi^{-1})(x), x \in X$$

Small Deformation Image Matching

$$\min_{v} \quad \frac{1}{2}\int_X Av \cdot v \, dx + \frac{1}{2}\int_X \left|I \circ \phi^{-1} - I'\right|^2 dx$$

Necessary Condition:

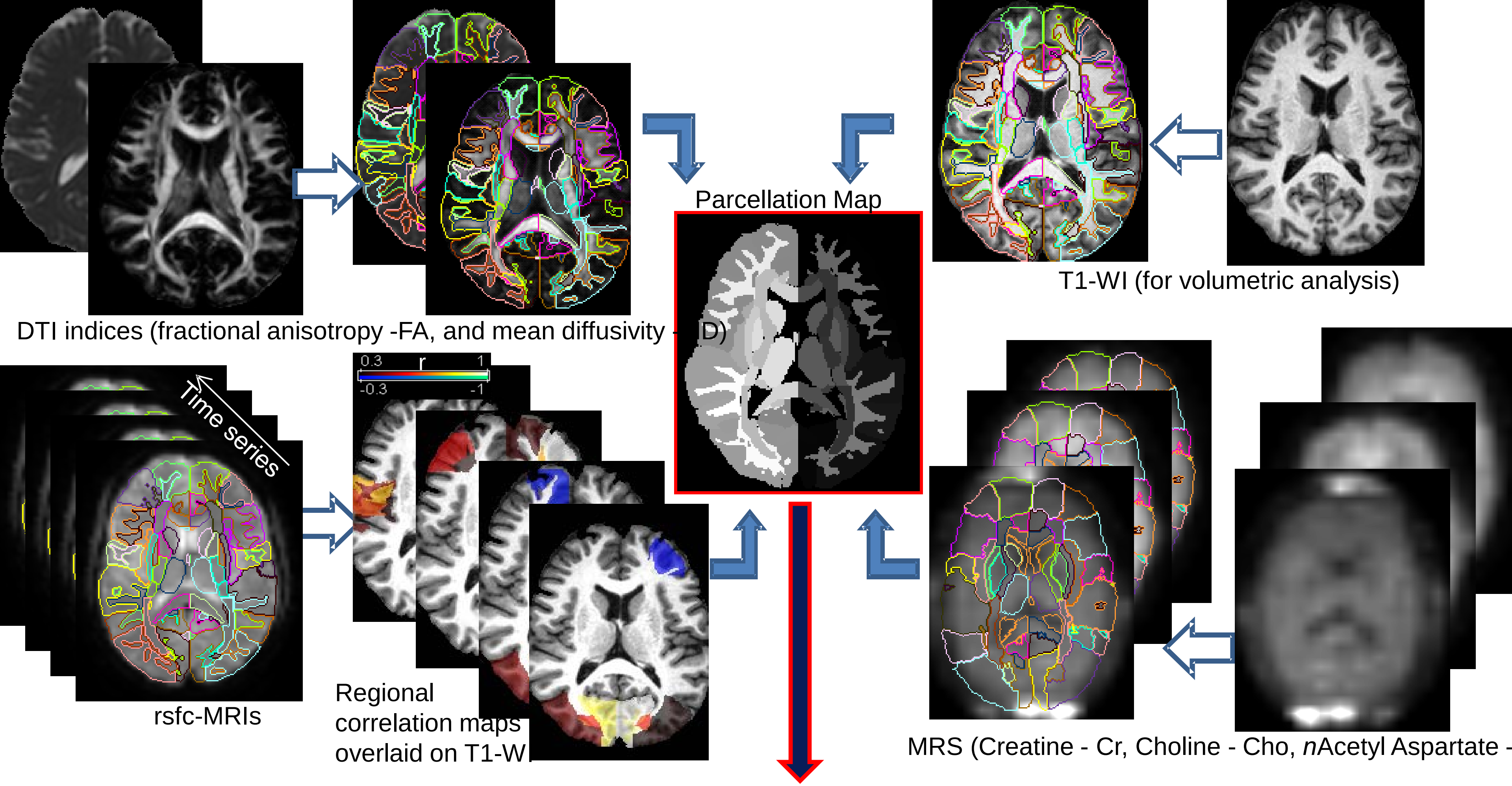
$$Av = (I \circ \phi^{-1} - I')\nabla(I \circ \phi^{-1})$$

Non-Linear Iteration:

$$Av^{new} = Av^{old} + \epsilon(Av^{old} - \left(I \circ \phi^{-1old} - I'\right)\nabla\left(I \circ \phi^{-1old}\right))$$

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

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



Neuroinformatics in Registered Coordinates

Subject n	volme	FA	MD	Cr	Cho	NAA	rsfc-fMRI correlation vector	GE	LE	PL	CC
SFL											
CingL											
PrCuL											
…											
region n											

Subject 3	volme	FA	MD	Cr	Cho	NAA	rsfc-fMRI correlation vector	GE	LE	PL	CC
SFL											
CingL											
PrCuL											
…											
region n											

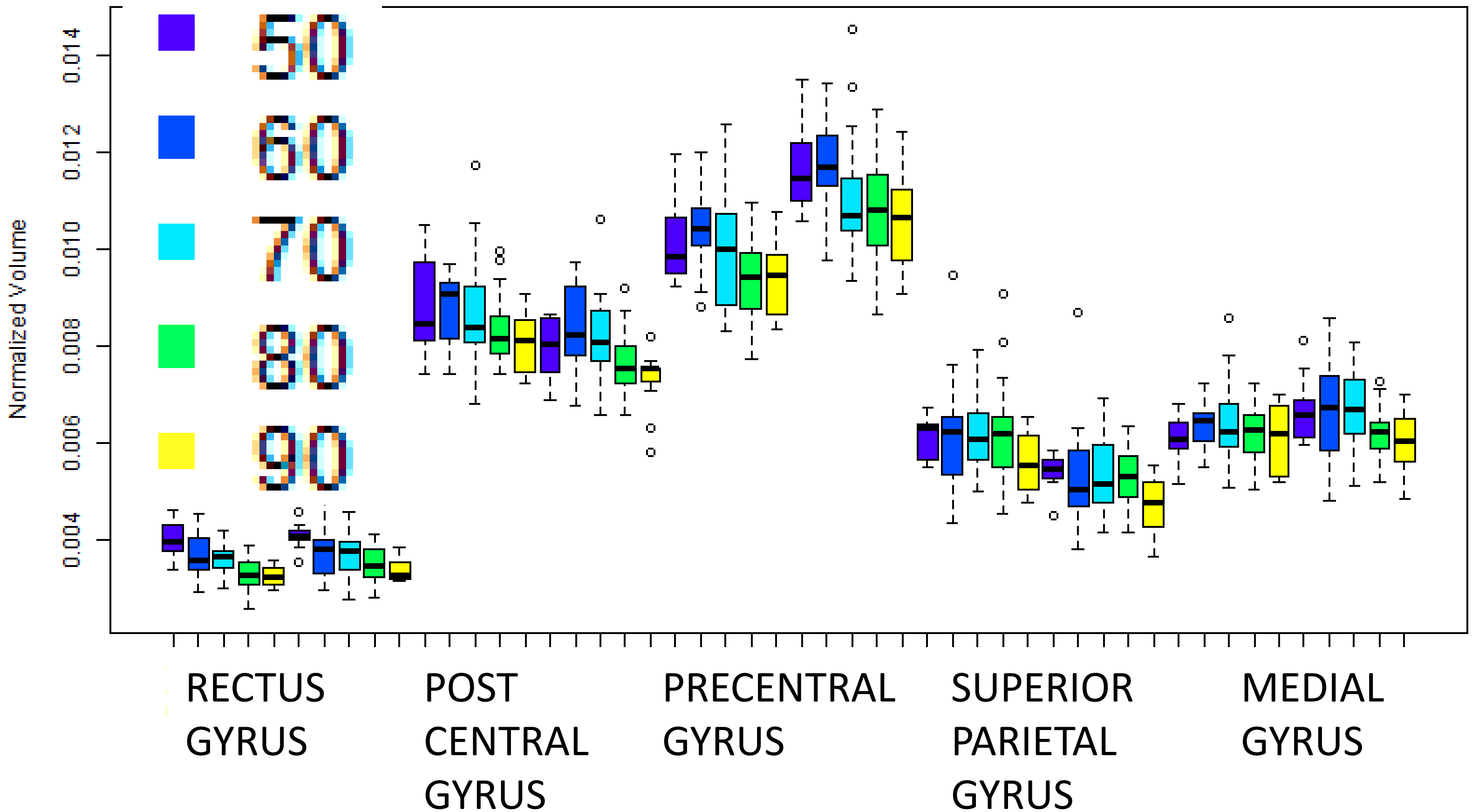
Subject 2	volme	FA	MD	Cr	Cho	NAA	rsfc-fMRI correlation vector	GE	LE	PL	CC
SFL											
CingL											
PrCuL											
…											
region											

Subject 1	volme	FA	MD	Cr	Cho	NAA	rsfc-fMRI correlation vector	GE	LE	PL	CC
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
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, 0.5,1, …, PrCuL vs. region n	0.49	0.68	2.4	0.38
…	…	…	…	…	…	…	…	…	…	…	…
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

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

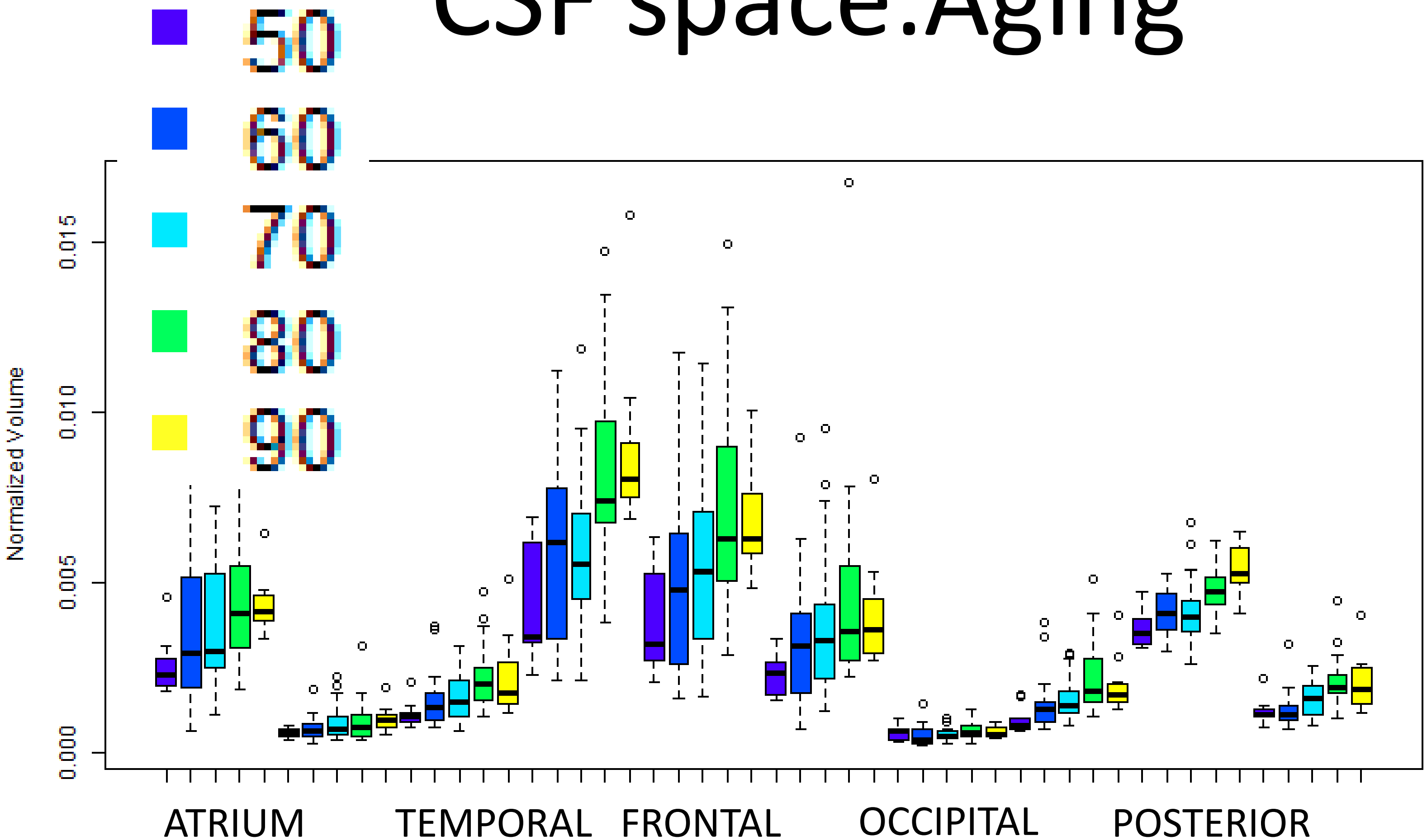
…aging does as well.

T1 Gray matter: Aging



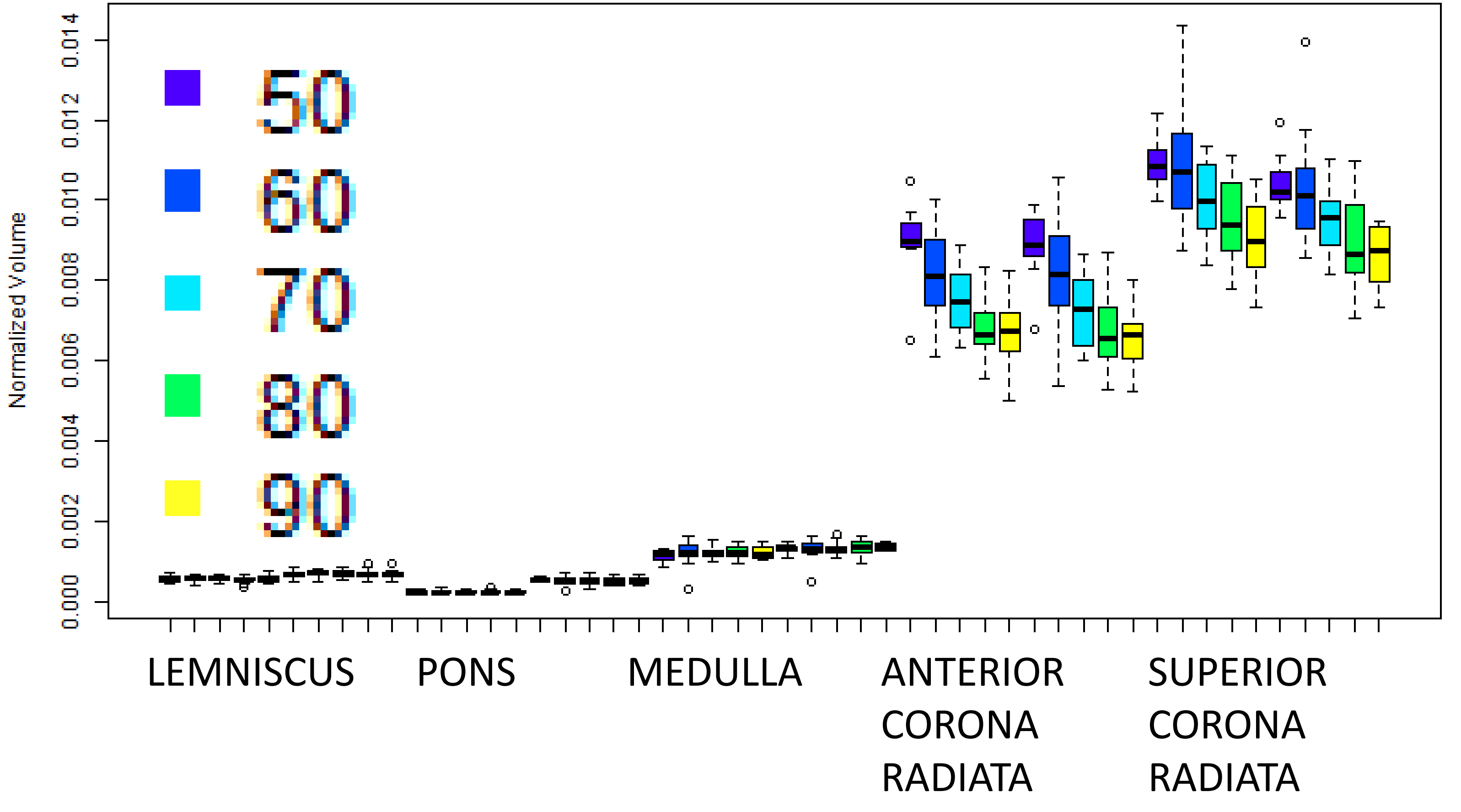
Gray matter spaces are decreasing!

CSF space:Aging



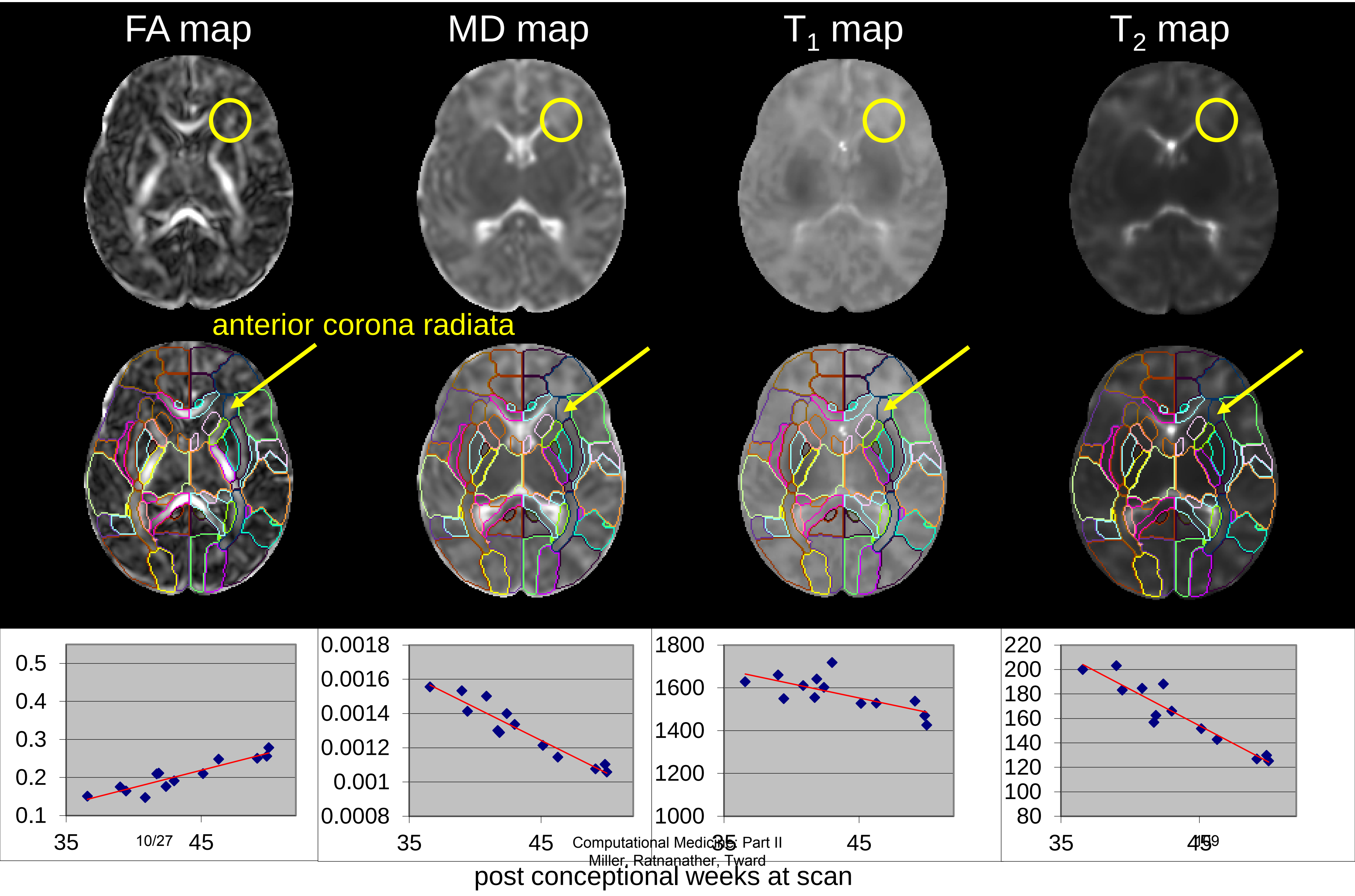
Spinal fluid spaces are growing!

DWI White Matter: Aging



Growth as well

Neonatal Growing Brain (K. Oishi)

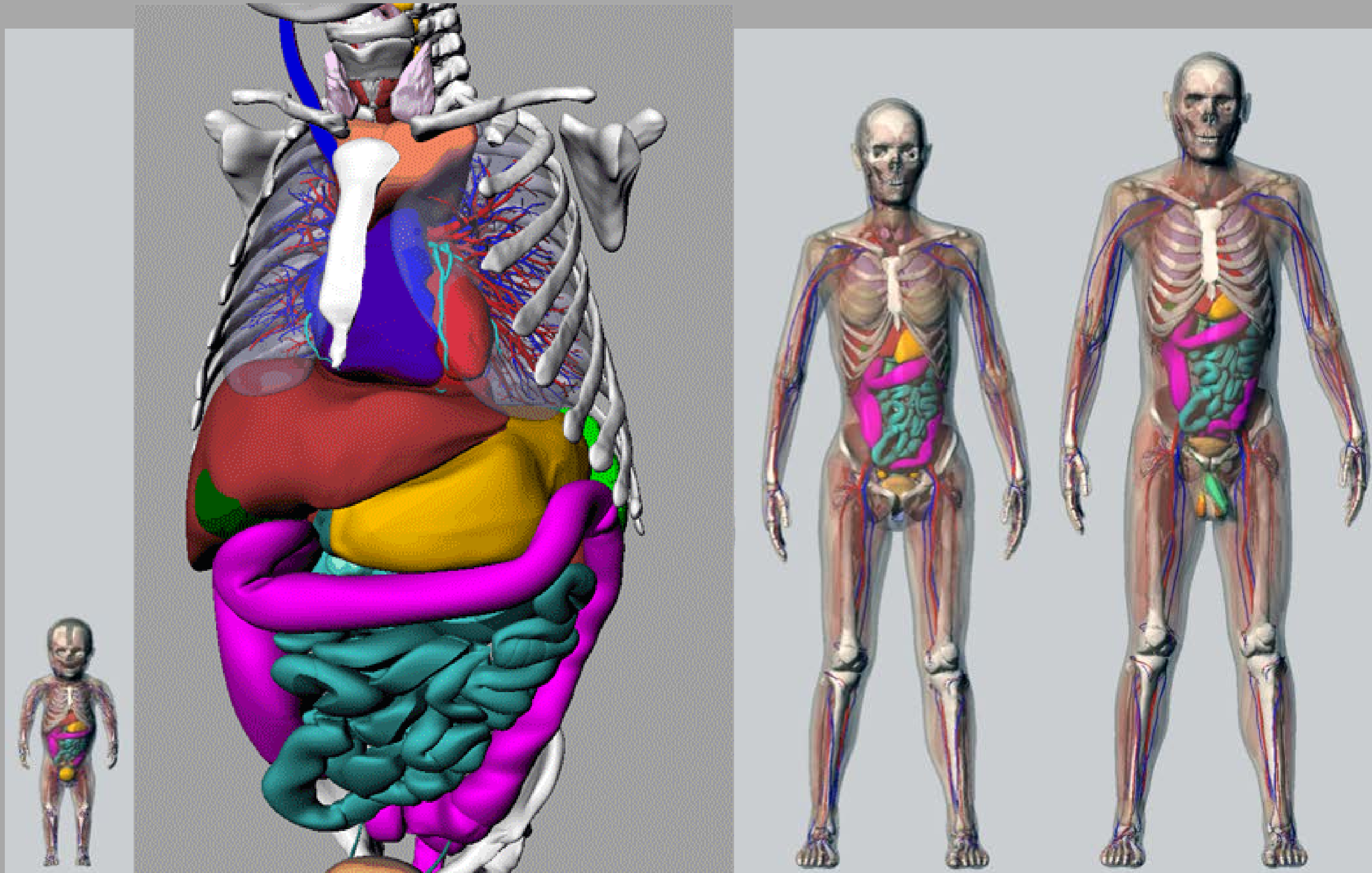


Cardiac

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

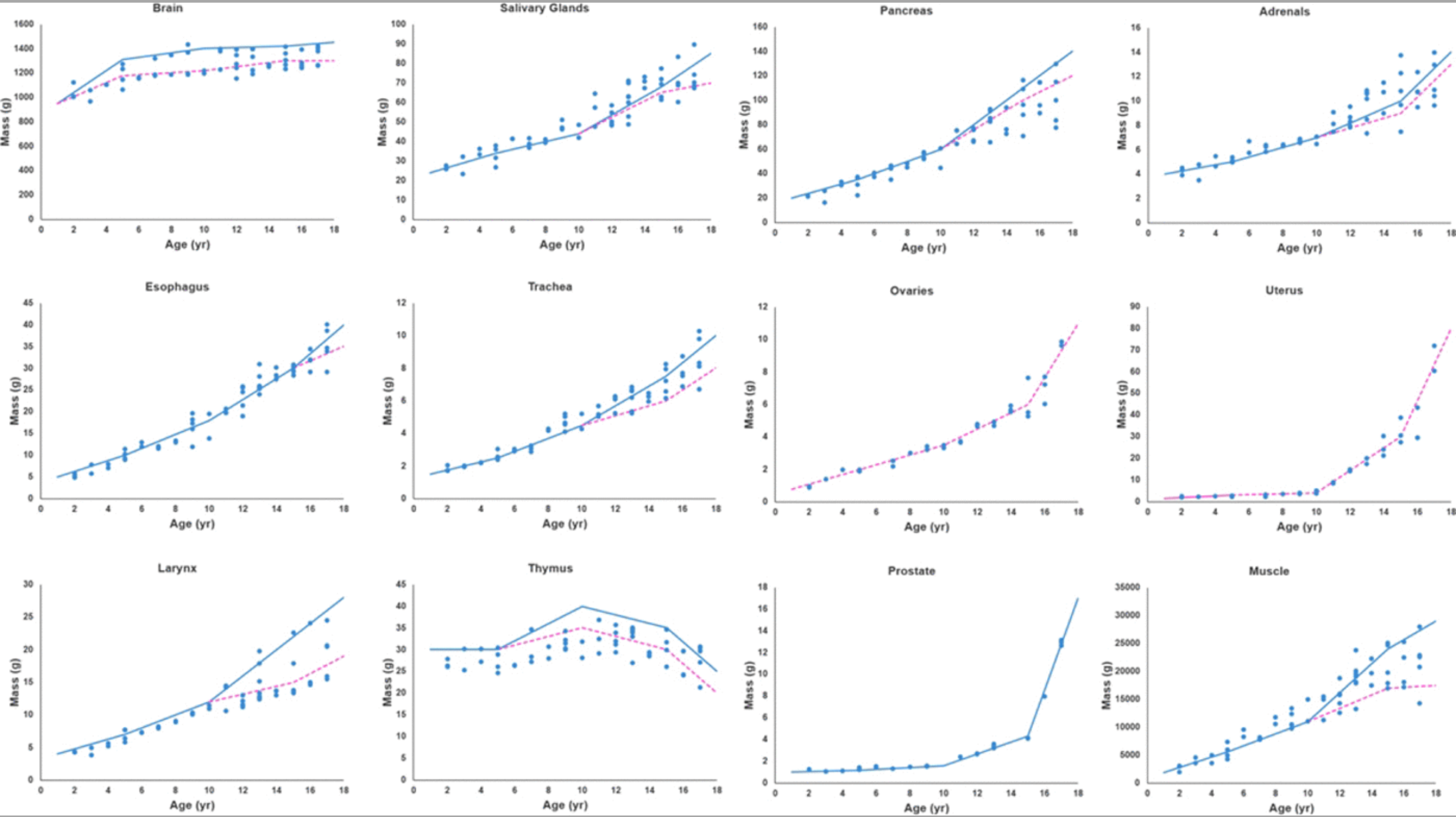


Segars el al, Population of anatomically variable 4D XCAT adult phantoms for imaging research and optimization, Med Phys, 40, (2013).

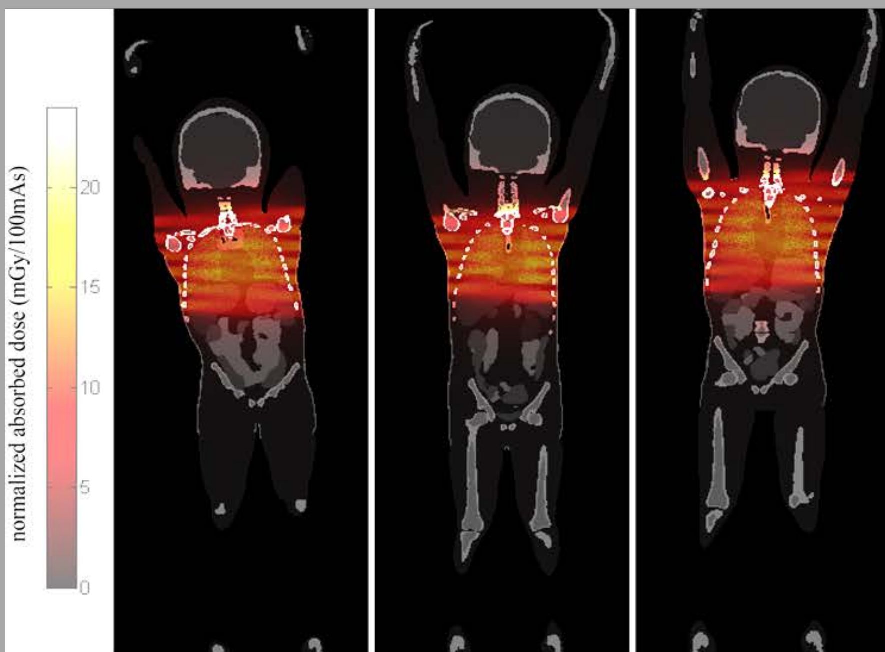
Norris el al, A set of 4D Pediatric XCAT Reference Phantoms for Multimodality Research, Med Phys, 41, (2014).

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

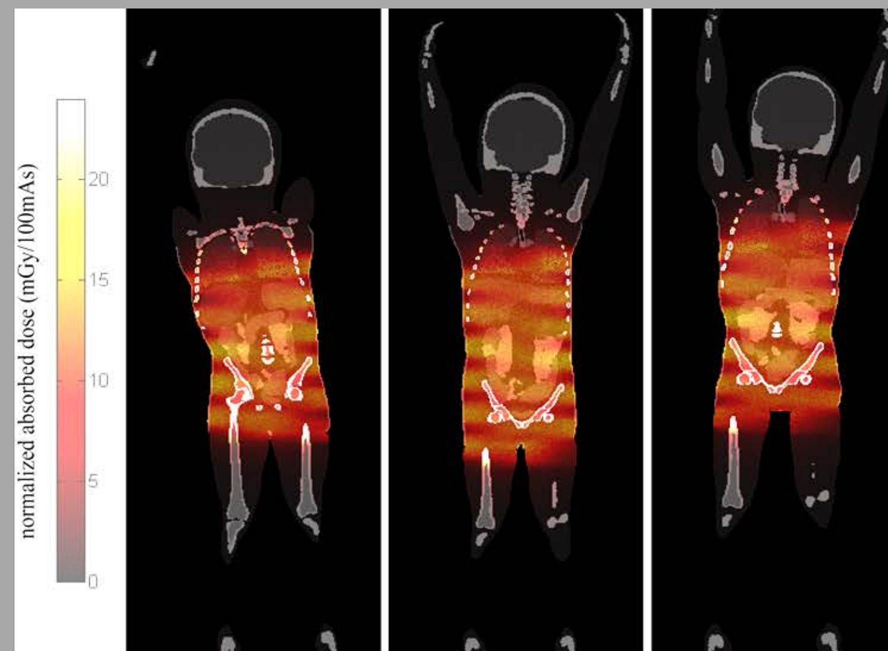
Mass versus age for the organs and structures predicted by the MC-LDDMM transform



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

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

Center for IMAGING SCIENCE
JOHNS HOPKINS UNIVERSITY

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.

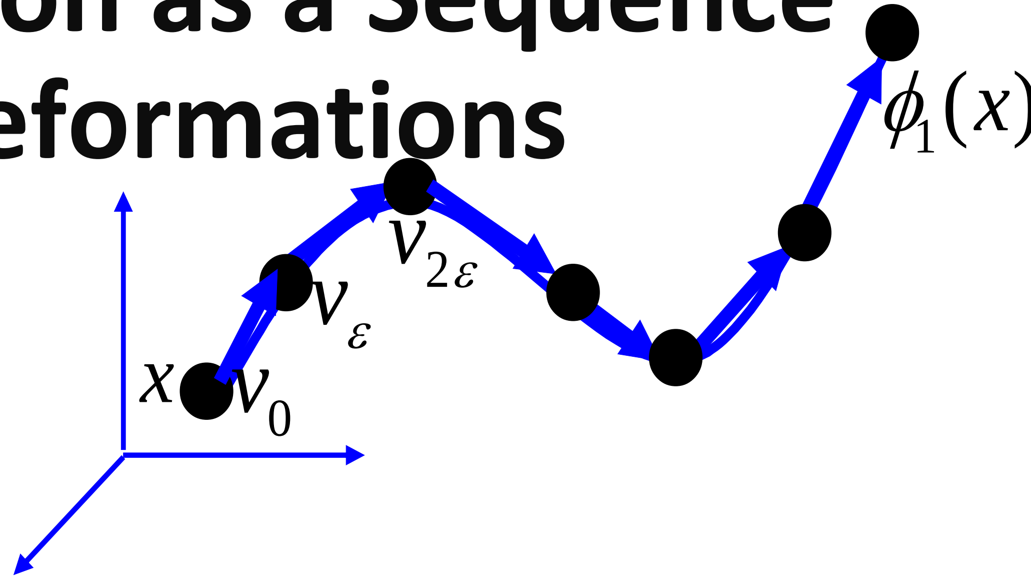
We introduce flows, a composition of spline vector fields, because they are 1-1 and onto for large deformations handling the self-intersecting problems of coordinate systems.

Lagrangian and Eulerian Flows for Medical Imaging

https://en.wikipedia.org/wiki/Computational_anatomy#Lagrangian_and_Eulerian_flows_for_generating_diffeomorphisms

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

Large Deformation as a Sequence of Small Deformations



$$\phi_0(x) = x$$

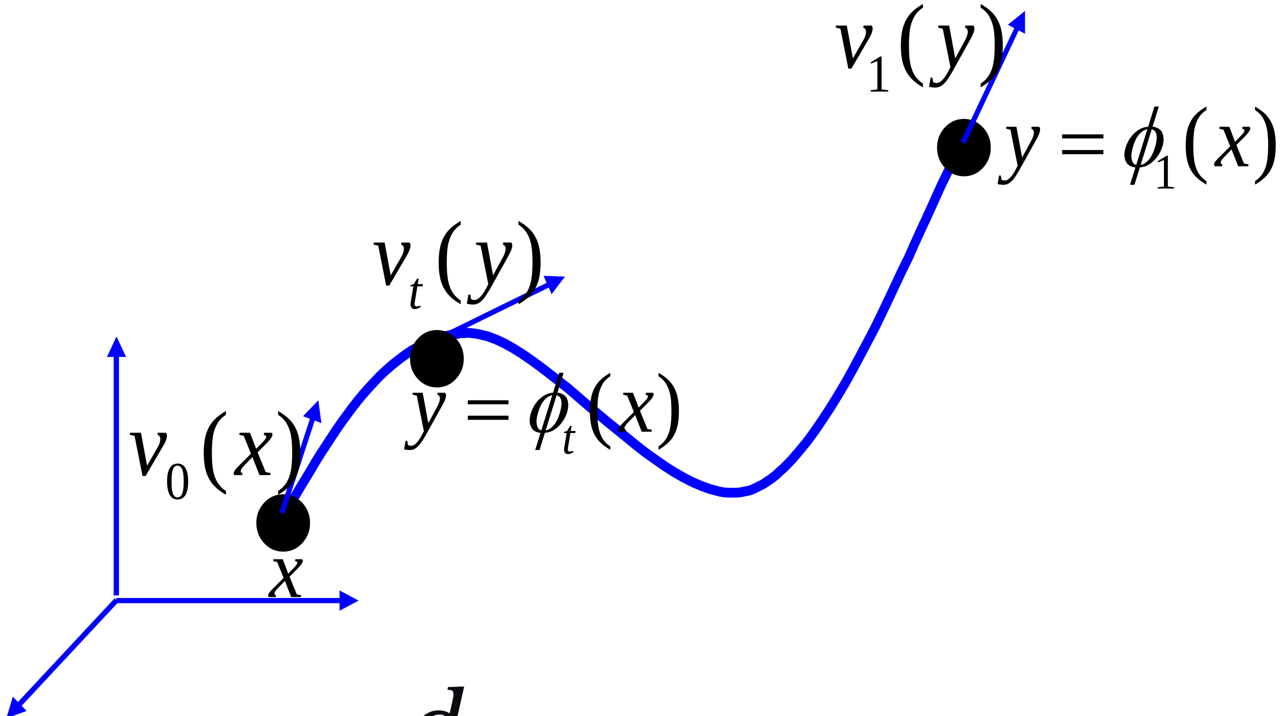
$$\phi_\epsilon(x) = x + v_0\Big|_{\phi_0(x)}$$

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

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

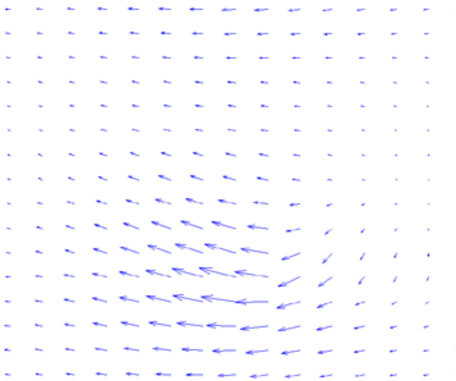
$$\phi_t : x \mapsto x + \sum_{i=0,1,\ldots} v_{i\epsilon} \circ \phi_{i\epsilon}(x), \quad v\ vector\ fields$$

Large Deformation Flow

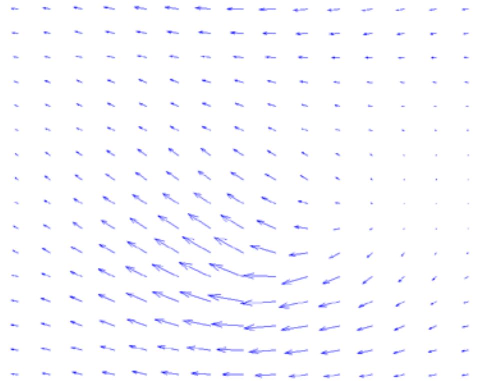


$$\frac{d}{dt}\phi_t = v_t \circ \phi_t$$

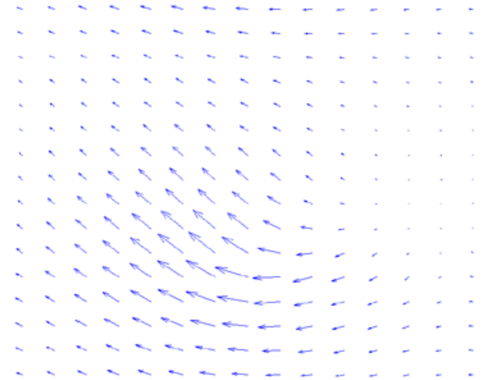
$$\phi_t : x \mapsto x + \int_0^t v_s \circ \phi_s(x)\,dx\,,$$



v_0

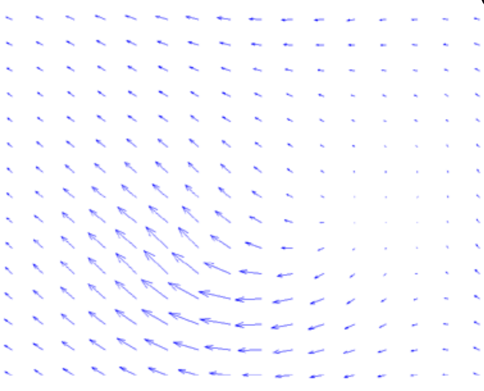


v_6

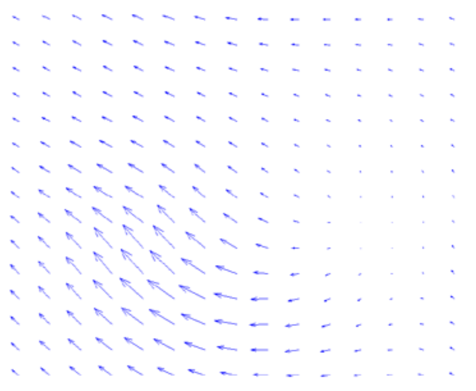


v_{12}

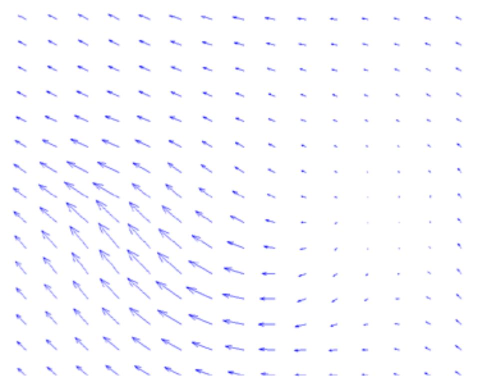
$$\frac{d}{dt}\phi_t = v_t \circ \phi_t$$



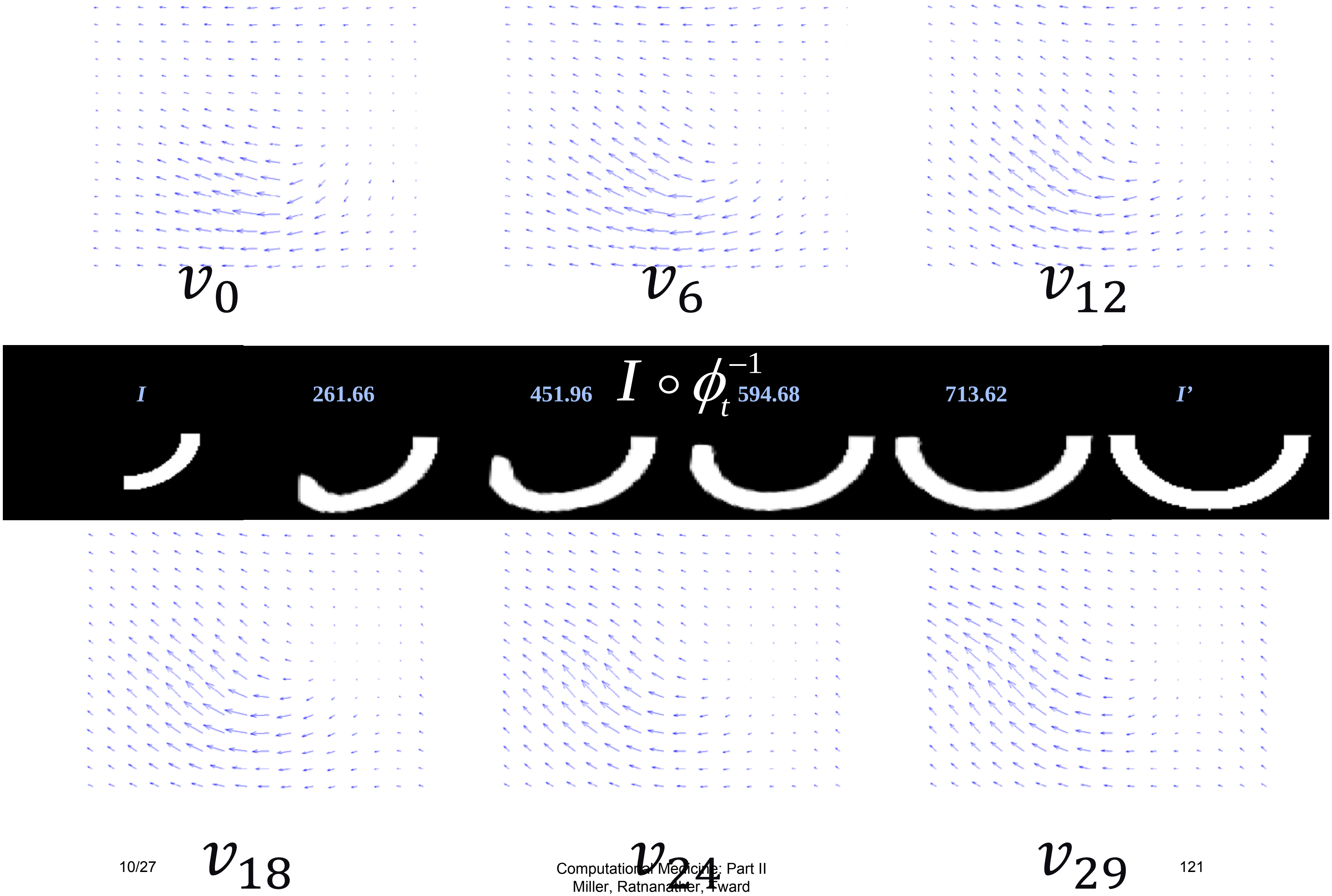
v_{18}



v_{24}



v_{29}

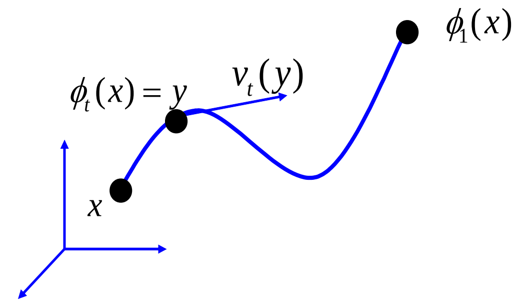

v_0
v_6
v_{12}
I
261.66
451.96
$I \circ \phi_t^{-1}$
594.68
713.62
I'
v_{18}
v_{24}
v_{29}

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

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



$$\phi_t = \phi_{t-\epsilon} + \phi_t - \phi_{t-\epsilon}$$
$$\approx (id + \epsilon v_{t-\epsilon}) \circ \phi_{t-\epsilon}$$
$$= (id + \epsilon v_{t-\epsilon}) \circ (\phi_{t-2\epsilon} + \phi_{t-\epsilon} - \phi_{t-2\epsilon})$$
$$\approx (id + \epsilon v_{t-\epsilon}) \circ (id + \epsilon v_{t-2\epsilon}) \circ \phi_{t-2\epsilon}$$
$$\approx (id + \epsilon v_{t-\epsilon}) \circ (id + \epsilon v_{t-2\epsilon}) \circ \cdots \circ (id + \epsilon v_0)$$

Splines are Addition

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

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

Registering Coordinate Systems via Large Deformation Image Matching (LDDMM)

https://en.wikipedia.org/wiki/Large_deformation_diffeomorphic_metric_mapping

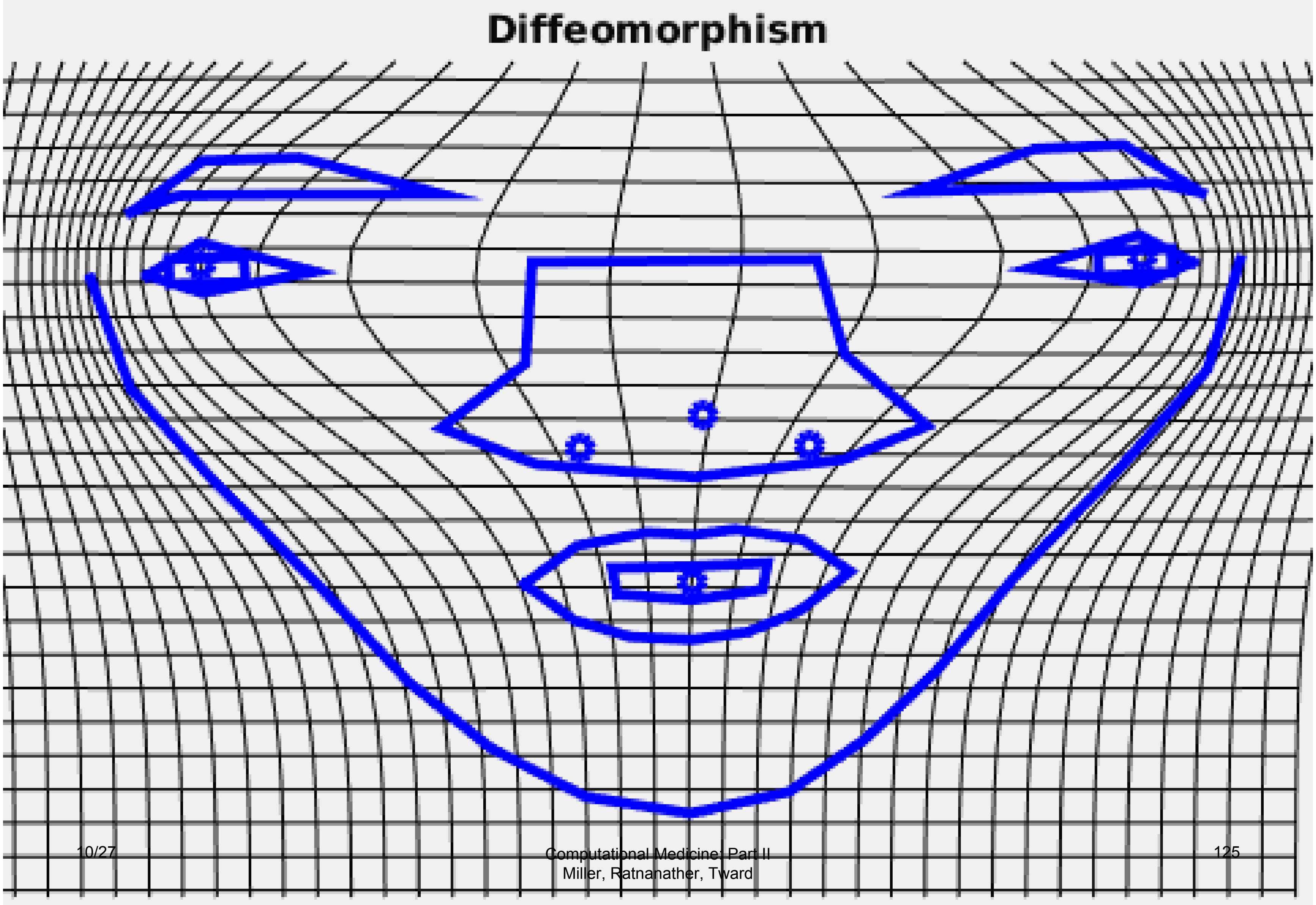
Michael I. Miller

Tilak Ratnanather

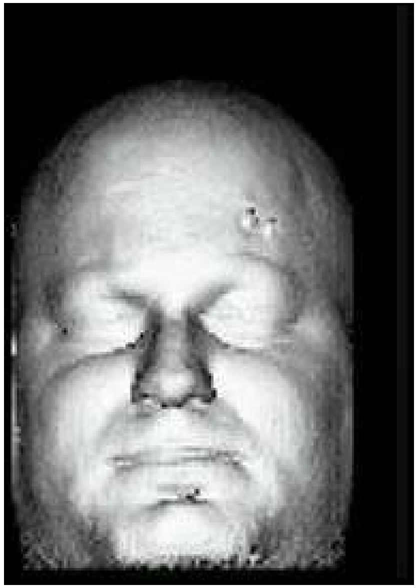
Daniel Tward

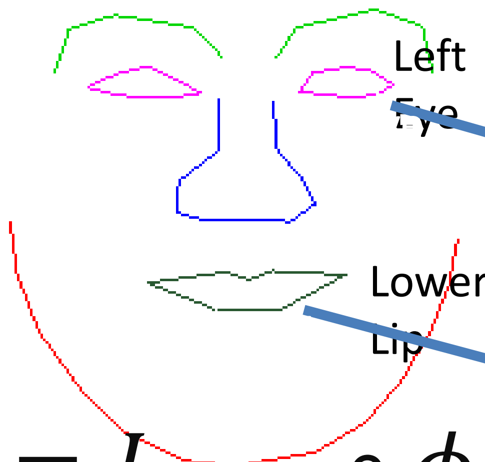
Center for IMAGING SCIENCE
JOHNS HOPKINS UNIVERSITY

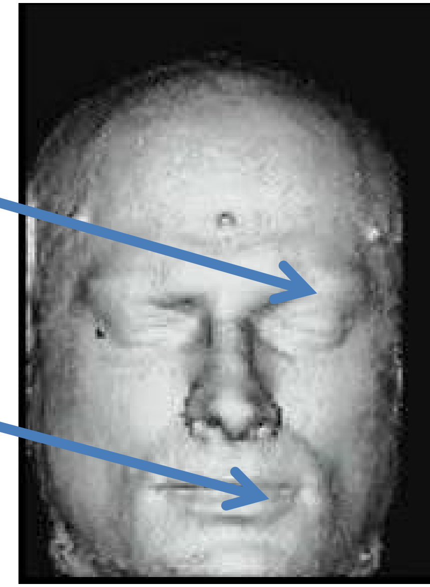
The computational algorithm for the GPS is called Large Deformation Diffeomorphic Metric Mapping.



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

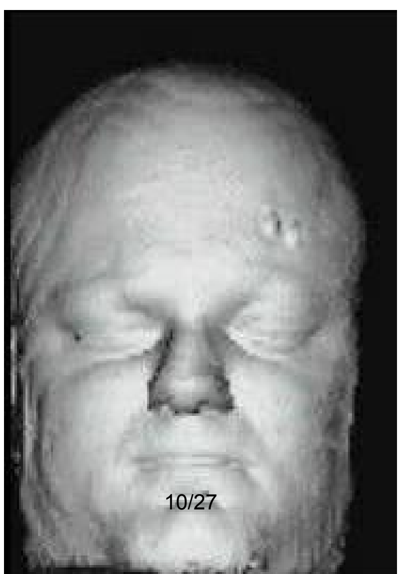


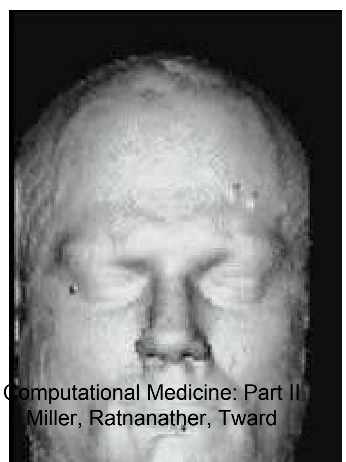


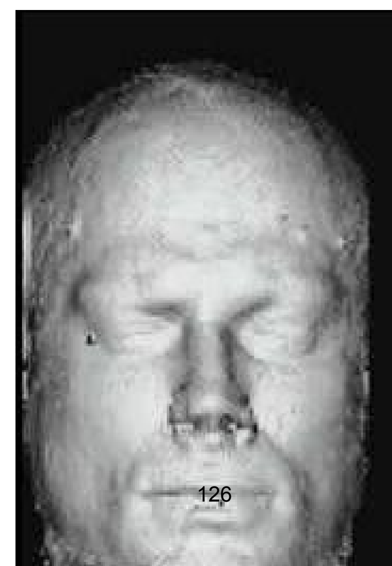


$$I = I_{temp} \circ \phi_1^{-1}$$

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







LDDMM Image Matching

https://en.wikipedia.org/wiki/Large_deformation_diffeomorphic_metric_mapping#LDDMM_algorithm_for_dense_image_matching

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

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

$$\phi_{t,1} = \phi_1 \circ \phi_t^{-1}, \qquad \frac{d}{d\epsilon} J(v + \epsilon w)\Big|_{\epsilon=0} =$$

$$\int_0^1 \int_X \underbrace{\left(Av_t - \left(I \circ \phi_t^{-1} - I' \circ \phi_{t,1}\right)\nabla(I \circ \phi_t^{-1})\left|\partial_X(\phi_{t,1})\right|\right)}_{=0} \cdot w_t dxdt$$

Necessary Condition:

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

Iteration for LDDMM Image Matching

https://en.wikipedia.org/wiki/Large_deformation_diffeomorphic_metric_mapping#LDDMM_algorithm_for_dense_image_matching

$$\min_{v} \ \frac{1}{2}\int_0^1 \int_X Av_t \cdot v_t \, dxdt + \frac{1}{2}\int_X \left|I \circ \phi_1^{-1} - I'\right|^2 dx$$

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

Necessary Condition:

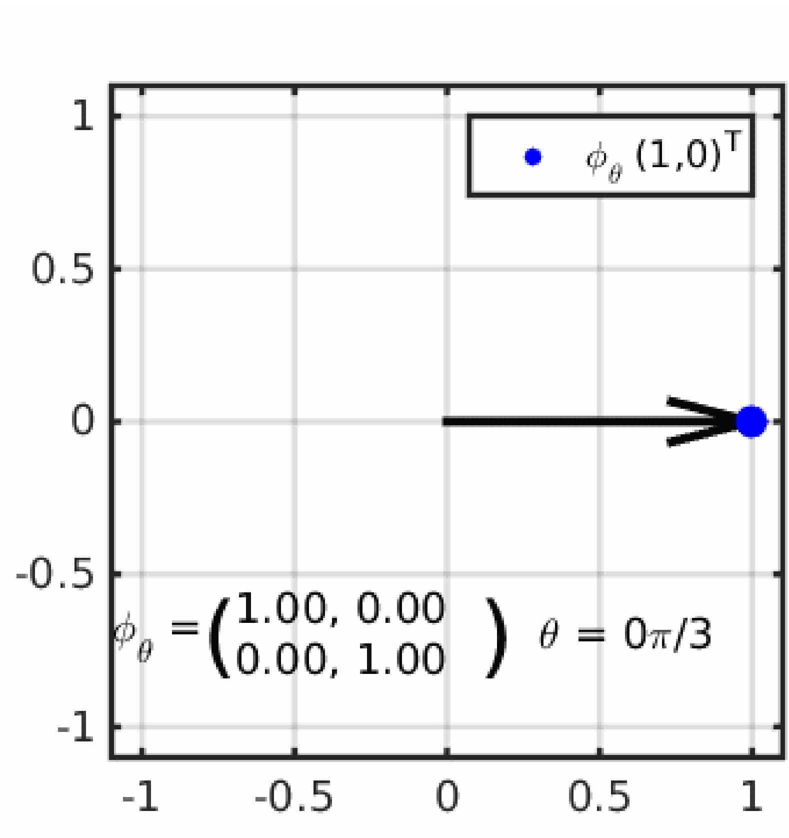
$$Av_t = \left(I \circ \phi_t^{-1} - I' \circ \phi_{t,1}\right)\nabla(I \circ \phi_t^{-1})\left|\partial_X(\phi_{t,1})\right|$$

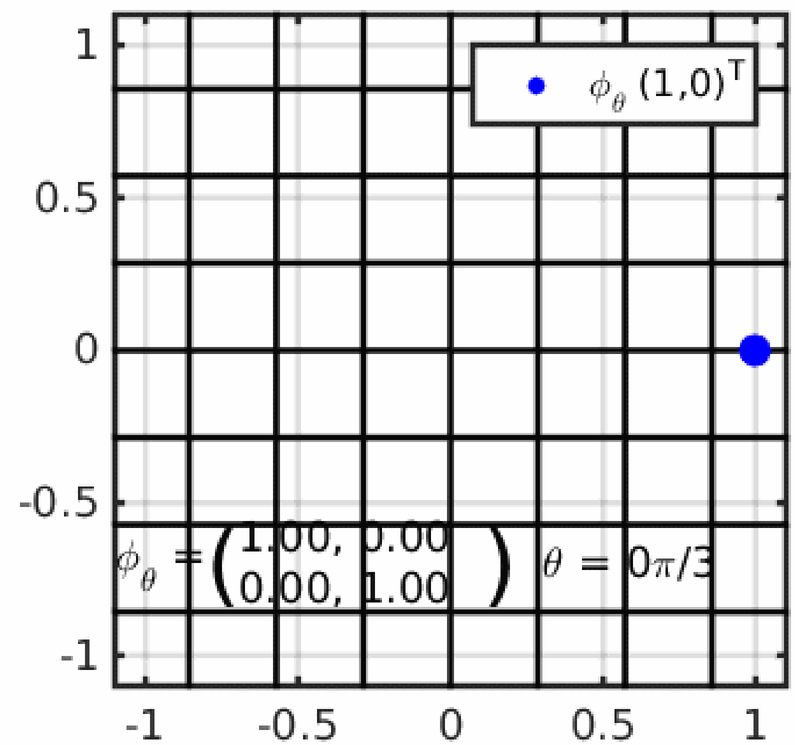
Non-Linear Iteration:

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

Daniel Tward:
Do sequence of rotations as large deformation.

Large rotation as a composition of small rotations.





$$\underbrace{\varphi_n}_{\begin{pmatrix} \cos 30 & -\sin 30 \\ \sin 30 & \cos 30 \end{pmatrix}} \circ \underbrace{\varphi_{n-1}}_{\begin{pmatrix} \cos 30 & -\sin 30 \\ \sin 30 & \cos 30 \end{pmatrix}} \circ \cdots \circ \underbrace{\phi_1}_{\begin{pmatrix} \cos 30 & -\sin 30 \\ \sin 30 & \cos 30 \end{pmatrix}}$$

LDDMM Landmark Matching

https://en.wikipedia.org/wiki/Large_deformation_diffeomorphic_metric_mapping#LDDMM_registered_landmark_matching

$$\min_v J = \frac{1}{2}\int_0^1 \int_X Av_t \cdot v_t \, dxdt + \frac{1}{2}\sum_i |y_i - \phi_1(x_i)|^2$$

$$\frac{d}{d\epsilon} J(v+\epsilon w) =$$

$$\int_0^1 \int_X \left(Av_t - \sum_i \delta_{\phi_1(x_i)}(x) \;\; \partial_X(\phi_{t,1}) \right) \Big|_{\phi_1(x_i)}^T (\phi_1(x_i) - y_i) \cdot w_t dxd$$

Non-Linear Iteration:

$$Av_t^{new} = Av_t^{old} + \epsilon(\left(Av_t^{old} \right.$$

$$\left. - \sum_i \delta_{\phi_1^{old}(x_i)}(y) \;\; \partial_X(\phi_{t,1}^{old}) \right) \Big|_{\phi_1^{old}(x_i)}^T (\phi_1^{old}(x_i) - y_i)$$

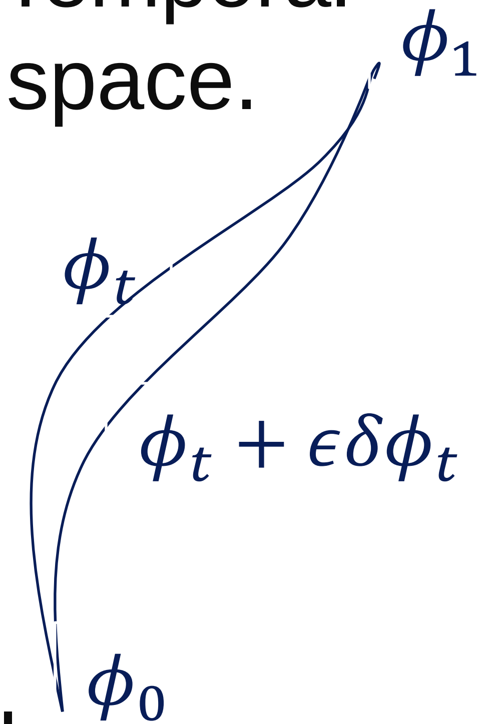
There are many possible flows.

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

Hamilton's Principle of Least Action & the Lagrangian

- Coordinates ϕ_t of system at time t. Temporal evolution is a curve in configuration space.



- Lagrangian $\mathcal{L}(\phi_t, \dot{\phi}_t)$ = 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

- $f_t \doteq \phi_t, g_t \doteq \dot{\phi}_t, f^\epsilon = \phi + \epsilon h, g^\epsilon = \dot{\phi} + \epsilon \frac{d}{dt} h$

$$\frac{d}{d\epsilon} J(\epsilon)\Big|_{\epsilon=0} = \int_0^1 \int_X \frac{d}{d\epsilon} L(f_t^\epsilon(x), g_t^\epsilon(x))\Big|_{\epsilon=0} dxdt$$

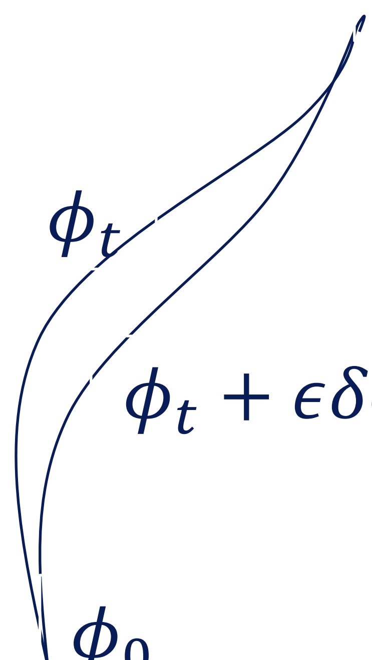
$$= \int_0^1 \int_X \left(\frac{\partial L(f_t^\epsilon, g_t^\epsilon)}{\partial f} \cdot \frac{d f_t^\epsilon}{d\epsilon} + \frac{\partial L(f_t^\epsilon, g_t^\epsilon)}{\partial g} \cdot \frac{d g_t^\epsilon}{d\epsilon}\right)\Big|_{\epsilon=0} dx\, dt$$

$$= \int_0^1 \int_X \left(\frac{\partial L\left(\phi_t, \dot{\phi}_t\right)}{\partial \phi} \cdot h_t + \frac{\partial L\left(\phi_t, \dot{\phi}_t\right)}{\partial \dot{\phi}} \cdot \frac{d}{dt} h_t\right) dx\, dt$$

$$= \int_0^1 \int_X \underbrace{\left(\frac{\partial L(\phi_t, \dot{\phi}_t)}{\partial \phi} - \frac{d}{dt} \frac{\partial L(\phi_t, \dot{\phi}_t)}{\partial \dot{\phi}}\right)}_{=0 \quad Euler-Lagrange} \cdot h_t \, dxdt$$

Euler-Lagrange Equation for Hamilton's Principle

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



- 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

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

Euler-Lagrange of Lumped Mass-Spring Mechanical System

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

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:

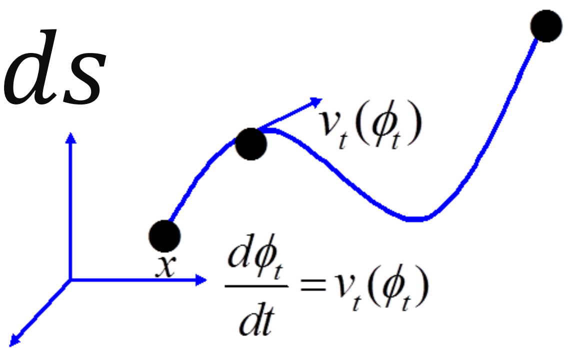
$$m\ddot{\phi}_t + k\phi_t = 0, \phi_t = \Phi_{init}\cos(\sqrt{k/m}\, t + \theta)$$

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#Surface_geodesics
https://en.wikipedia.org/wiki/Computational_anatomy#Volume_geodesics

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



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

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

$$\dot{p}_t = - \left. (\partial_X v_t^T) \right|_{\phi_t} p_t, \quad p_0 = A v_0$$

Euler-Lagrange Large Deformations

https://en.wikipedia.org/wiki/Computational_anatomy#Landmark_or_pointset_geodesics
https://en.wikipedia.org/wiki/Computational_anatomy#Surface_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

$$p_t(x) = \frac{\partial L(\phi_t(x), \dot{\phi}_t(x))}{\partial \dot{\phi}} = Av_t \circ \phi_t(x) \, |\partial_X \phi_t(x)|$$

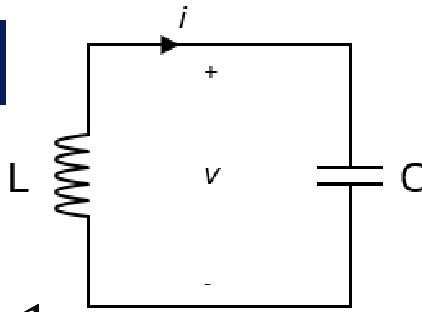
- Euler-Lagrange and Geodesic vector fields

$$\dot{p}_t = - \left. (\partial_X v_t^T) \right|_{\phi_t} p_t, \;\; p_0 = Av_0$$

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

Daniel Tward Examples

Euler-Lagrange of Lumped L-C Electrical System



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

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:

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

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

$$\dot{p}_{tx} = 0, \qquad \dot{p}_{tz} = -mg$$

- Geodesic Solution $\phi_{tx} = t, \;\; \phi_{tz} = -\frac{1}{2} g t^2 + \frac{1}{2} g t$

Cannonball cont'd

- Given Least Action Euler-Lagrange Equation,

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

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

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

$$\dot{p}_{tx} = m\ddot{\phi}_{tx} = 0 \Rightarrow \phi_{tx} = at + b \Rightarrow b = \phi_{00} = 0$$

$$\phi_{1x} = 1 \Rightarrow a = 1 \Rightarrow \phi_{tx} = t$$

$$\dot{p}_{tz} = -mg \Rightarrow m\ddot{\phi}_{tz} = -mg \Rightarrow \dot{\phi}_{tz} = -gt + c$$

$$\Rightarrow \phi_{tz} = -\frac{1}{2}gt^2 + ct + d$$

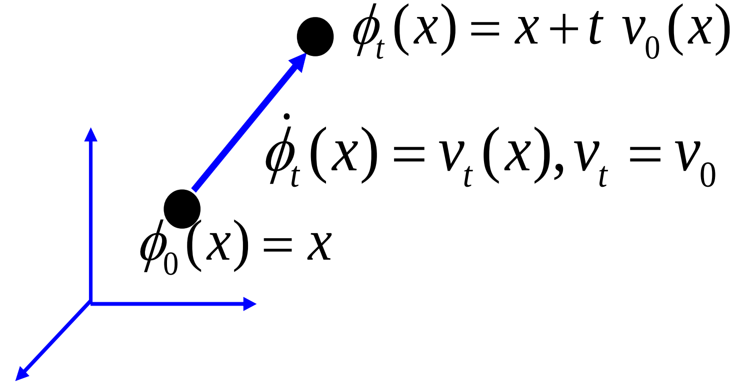
$$\phi_{0z} = 0 \Rightarrow d = 0, \phi_{1z} = 0 \Rightarrow c = \frac{1}{2}g$$

Daniel Tward

Small Deformation Splines are Least Action

Splines are Small Deformation Flows following Constant Speed Straight Lines

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



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

$$v_t(x) = \int_X k(x-y)p_0(y)dy = v_0(x)\,, p_0 = Av_0$$

Euler-Lagrange: Small Deformation Splines

- Velocity (3D): $\dot{\phi}_t(x) = v_t(x), x \in X$
- K.E. Density: $L\big(\phi_t(x), \dot{\phi}_t(x)\big) = \frac{1}{2} A v_t(x) \cdot v_t(x)$
$$= \frac{1}{2} A \dot{\phi}_t(x) \cdot \dot{\phi}_t(x)$$
- Canonical Momentum:
$$p_t(x) = \frac{\partial L\big(\phi_t(x), \dot{\phi}_t(x)\big)}{\partial \dot{\phi}} = A v_t(x)$$
- Euler-Lagrange and Geodesic vector fields
$$\frac{\partial L}{\partial \phi} = 0, \qquad \dot{p}_t = 0, \quad p_0 = A v_0$$
$$v_t(x) = \int_X k(x-y) p_t(y) dy$$

Systems Models of Medical Imaging

Michael I. Miller

Tilak Ratnanather

Daniel Tward

Center for IMAGING SCIENCE

JOHNS HOPKINS UNIVERSITY

Physicists use Energy methods; engineers use systems methods.

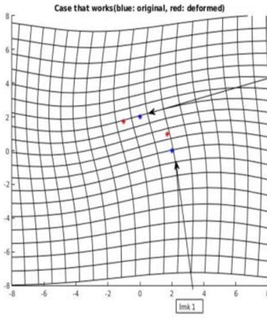
Examine our systems model of imagery.

Landmark Based Spline Models of Images

Michael I. Miller
Tilak Ratnanather
Daniel Tward

Fred Bookstein
Rohlf Medal

Small Deformation Spline Model of Medical Imaging



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

$$\sum_i p_i \delta(x - x_i) \xrightarrow{\text{input}} \boxed{\text{Linear Shift Invariant System Kernel } k(x)} \xrightarrow{\text{output}} v(x) = \int_X k(x-y)p(y)\,dy$$

$$\phi(x) = x + v(x) \xrightarrow{\text{input}} \boxed{\text{Linear Observer Equation}} \longrightarrow \phi(x_i), i = 1, \dots, n$$

Non − Linear Image Transformation (Small):

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

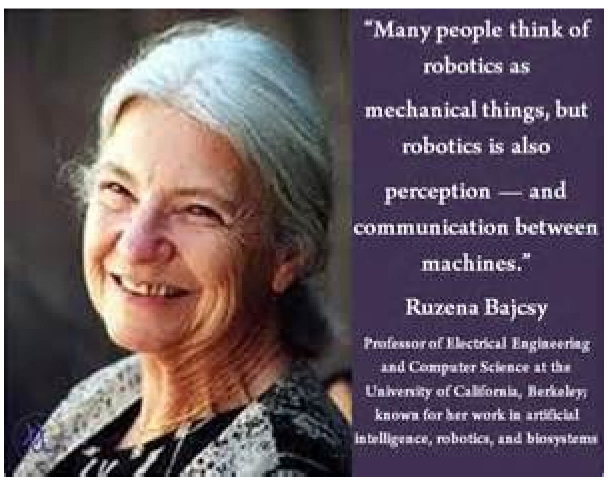
$$\phi(x) = x + v(x) \xrightarrow{\text{input}} \boxed{\text{Non-Linear Observer Equation}} \xrightarrow{\text{output}} I_{temp}\,(x - v(x))$$

Small Deformation Spline Models of Images

Michael I. Miller

Tilak Ratnanather

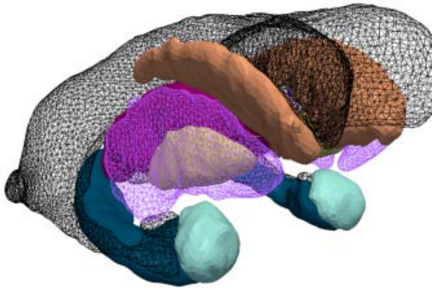
Daniel Tward



Ruzena Bajcsy

Center for IMAGING SCIENCE
JOHNS HOPKINS UNIVERSITY

Small Deformation Spline Model of Medical Imaging



Controller: Linear Shift Invariant System

input $p(x)$ → **Linear Shift Invariant System Kernel $k(x)$** → output

$$v(x) = \int_X k(x-y)p(y)\,dy$$

Image Transformation (Small):

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

input $\phi(x) = x + v(x)$ → **Non-Linear Observer Equation** → output

$$I_{temp}\,(x - v(x))$$

Dynamical Systems Model of Images


Gary Christensen


Richard Rabbitt

Michael I. Miller
Tilak Ratnanather
Daniel Tward

Dynamical System Model of Medical Imaging

Controller: Linear Shift Invariant System

$p_t(x)$ input → Control LSI: Kernel K → output $v_t(x) = \int K\big(x - \phi_t(y)\big)p_t(y)dy$

Dynamical System Constraint

$v_t(x), t \in [0,T]$ input → Non-Linear Dynamical System → output $\phi_t(x) = x + \int_0^t v_s \circ \phi_s(x)ds$

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

$\phi_t(x)$ input → Non-Linear Observer Equation → output $I_{temp} \circ \phi_t^{-1}(x), x \in X$

Geodesic Shooting Dynamical Systems Model of Medical Imagery

Geodesic Momentum Dynamics with Linear Controller

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

$$p_t(x) \xrightarrow{\text{input}} \boxed{\text{LSI Kernel k}} \xrightarrow{\text{output}} v_t(x) = \int k\big(x - \phi_t(y)\big) p_t(y) dy$$

Eulerian Dynamics

$$v_t(x) \xrightarrow{\text{input}} \boxed{\text{Non-Linear Dynamics}} \xrightarrow{\text{output}} \phi_t(x) = x + \int_0^t v_s \circ \phi_s(x) ds$$

Image Transformation (large):

$$\phi_t(x) \xrightarrow{\text{input}} \boxed{\text{Non-Linear Observer Equation}} \xrightarrow{\text{output}} I_{temp} \circ \phi_t^{-1}(x), x \in X$$

Homework