# Diffeomorphic Image Registration by Geodesic Shooting

John Ashburner

Wellcome Trust Centre for Neuroimaging UCL Institute of Neurlogy London, UK.

ヨート

# Outline

- 1 Small Deformation Model
  - Differential Operator
  - Optimisation
- 2 Large-Deformation Diffeomorphic Metric Mapping
  - Diffeomorphisms & Geodesics
  - LDDMM Itself

#### 3 Geodesic Shooting

- Generating Deformations from Initial Velocities
- Optimising the Initial Velocities
- Initial Momentum
- Examples

・ 同 ト ・ ヨ ト ・ ヨ ト

Differential Operator Optimisation

#### Small Deformation Approximation

 $\mu^{\circ}(x-v(x))$ 



x-v(x)



|J<sup>x-v(x)</sup>|







x+v(x)







f	, (X+)	/(x))
---	--------	-------



x+v(x)







μ ° (x–v(x))











æ

<ロ> <同> <同> < 同> < 同>

Geodesic Shooting

Differential Operator Optimisation

# Small Deformation Model

- Should fit a model to the data (as opposed to fitting the data to the model).
- The objective function for least-squares fitting is often expressed as:

$$\mathcal{E} = \frac{1}{2} ||\mathbf{L}\mathbf{v}||^2 + \frac{1}{2\sigma^2} ||f - \mu \left( \mathrm{Id} - \mathbf{v} \right)||^2$$

- f is our image (data).
- $\mu$  is the template (model).
- v is a displacemet field.
- $\bullet~{\rm Id}$  is an identity transform.
- L is a differential operator.

Differential Operator Optimisation

# Linear Elasticity Operator

The differential operator used is essentially the same linear elasticity operator used by Christensen (where D indicates computing the Jacobian:

$$||\mathbf{L}\mathbf{v}||^2 = \int_{\mathbf{x}\in\Omega} \left(\frac{\lambda_1}{4} ||\mathbf{D}\mathbf{v} + (\mathbf{D}\mathbf{v})^{\mathcal{T}}||^2 + \lambda_2 \mathrm{tr}(\mathbf{D}\mathbf{v})^2 + \lambda_3 ||\mathbf{v}||^2\right) d\mathbf{x}$$

This has three terms that penalise different things.

- Length changes  $(\mathbf{D}\mathbf{v} + (\mathbf{D}\mathbf{v})^T)$ .
- **2** Volume changes (tr(Dv)).
- Solute displacement (v needed to make the operator invertable).

イロト イポト イヨト イヨト

#### Discretisation

Differential Operator Optimisation

- Images and deformations are really stored discretely, and made continuous by interpolation.
- Generalised interpolation involves representing fields etc by linear combinations of basis functions.
- Linear interpolation, which involves very local triangular basis functions  $(b_i(\mathbf{x}))$ , so:

$$\mathbf{v}(\mathbf{x}) = \sum_i w_i \mathbf{b}_i(\mathbf{x})$$

イロト イポト イヨト イヨト

Differential Operator Optimisation

# Differential Operator in 1D

• In 1D, a simple operator could compute the gradients by **Gw**, where:

$$\mathbf{G} = \begin{pmatrix} -1 & 1 & 0 & 0 & \dots \\ 0 & -1 & 1 & 0 & \dots \\ 0 & 0 & -1 & 1 & \dots \\ 0 & 0 & 0 & -1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

- Penalty  $(||\mathbf{Lv}||^2)$  could be the sum of squares of the gradients.
- Computed by  $(\mathbf{G}\mathbf{w})^T (\mathbf{G}\mathbf{w}) \equiv \mathbf{w}^T \mathbf{G}^T \mathbf{G}\mathbf{w} \equiv \mathbf{w}^T \mathbf{A}\mathbf{w}$ .
- Note that the norm  $||\mathbf{L}\mathbf{v}||$  (ie  $\sqrt{\mathbf{w}^T \mathbf{A} \mathbf{w}}$ ) can be thought of as a distance.

- 4 同 6 4 日 6 4 日 6

Differential Operator Optimisation

## Differential Operator in 2D

• For more dimensions, we can use Kroneker tensor products.

$$\mathbf{G} \otimes \mathbf{B} = \begin{pmatrix} g_{11}\mathbf{B} & \dots & g_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ g_{m1}\mathbf{B} & \dots & g_{mn}\mathbf{B} \end{pmatrix}$$

• So, computing horizontal and vertical gradients of a square image (reshaped to a column vector) may be achieved using the following matrix:

$$\begin{pmatrix} \mathsf{I}\otimes\mathsf{G}\\\mathsf{G}\otimes\mathsf{I} \end{pmatrix}$$

Differential Operator for Computing Jacobians (2D)

• If we stack the x component of the displacement field on top of the y component, then we can compute elements of the Jacobians by:

$$\begin{pmatrix} \mathbf{I} \otimes \mathbf{G} & \mathbf{0} \\ \mathbf{G} \otimes \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \otimes \mathbf{G} \\ \mathbf{0} & \mathbf{G} \otimes \mathbf{I} \end{pmatrix} \mathbf{w}$$

• The divergence of the displacement is simply the trace of the Jacobian, so can be computed using:

$$(I \otimes G \quad G \otimes I) w$$

・ 同 ト ・ ヨ ト ・ ヨ ト

Differential Operator Optimisation

# Symmetric and Anti-symmetric Matrices

- A square matrix (J) can be decomposed into the sum of a symmetric and anti-symmetric (skew symmetric) matrix.
  - The symmetric part is  $(\mathbf{J} + \mathbf{J}^T)/2$ .
  - The anti-symmetric part is  $(\mathbf{J} \mathbf{J}^T)/2$ .
- The symmetric elements of the Jacobian of the displacement can be obtained from:

$$\begin{pmatrix} \mathsf{I} \otimes \mathsf{G} & \mathsf{0} \\ \mathsf{0} & \mathsf{G} \otimes \mathsf{I} \\ \frac{1}{2}\mathsf{G} \otimes \mathsf{I} & \frac{1}{2}\mathsf{I} \otimes \mathsf{G} \end{pmatrix} \mathsf{w}$$

• Similar large sparse matrices can be generated for 3D data.

< ロ > < 同 > < 回 > < 回 > .

Differential Operator Optimisation

# Matrix Exponentials

• A matrix exponential ("expm" in MATLAB) is given by:

$$\mathsf{e}^\mathsf{J} = \sum_{k=0}^\infty rac{1}{k!} \mathsf{J}^k$$

• For matrices very close to zero,  $e^{\mathbf{J}} \simeq \mathbf{I} + \mathbf{J}$ .

1

- The determinant  $|e^{\mathbf{J}}|$  may be computed from  $e^{\operatorname{tr}(\mathbf{J})}$ .
- If **J** is anti-symmetric, then  $e^{J}$  is orthogonal (ie a rigid rotation).

$$\mathbf{J} = w_1 \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + w_2 \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} + w_3 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$$

• We do not wish to penalise rotations, so only penalise the symmetric parts of the Jacobians.

#### Greens Function

- Note that the Green's function (K) of the operator may be computed from the (pseudo)inverse of A, although it is often easier to use Fourier methods.
- If you think of applying a differential operator as a convolution, then applying the Green's function is a deconvolution.



< □ > <

Differential Operator

Optimisation

Differential Operator Optimisation

#### Gauss-Newton Optimisation

An iterative procedure, using both first and second derivatives.

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - \left[ \left. \frac{\partial^2 \mathcal{E}}{\partial \mathbf{w}^2} \right|_{\mathbf{w}^{(i)}} \right]^{-1} \left[ \left. \frac{\partial \mathcal{E}}{\partial \mathbf{w}} \right|_{\mathbf{w}^{(i)}} \right]$$

$$rac{\partial \mathcal{E}}{\partial \mathbf{w}} = \mathbf{A}\mathbf{w} + \mathbf{g}$$

$$\frac{\partial^2 \mathcal{E}}{\partial w^2} = \mathbf{A} + \mathbf{H}$$

э

Gradients

$$\mathbf{g} = \begin{bmatrix} g_{11} \\ g_{21} \\ g_{31} \\ \vdots \\ g_{l1} \\ g_{l1} \\ g_{l1} \\ g_{21} \\ g_{l1} \\ g_{21} \\ g_{21} \\ g_{31} \\ \vdots \\ g_{l1} \end{bmatrix}$$
Using trilinear interpolation to represent the displacement fields, where the density of the control points is equal to the pixel spacing. Therefore  $\vartheta(\mathbf{x}_i) = [x_{i1} - w_{i1}, x_{i2} - w_{i2}].$ 

Differential Operator

Optimisation

$$g_{il} = \frac{1}{\sigma^2} \left( f(\mathbf{x}_i) - \mu(\vartheta(\mathbf{x}_i)) \right) \left( (\nabla_l \mu) \circ (\vartheta(\mathbf{x}_i)) \right)$$

æ

Differential Operator Optimisation

#### Hessian



$$h_{ilm} = \frac{1}{\sigma^2} ((\nabla_l \mu) \circ \vartheta(\mathbf{x}_i)) ((\nabla_m \mu) \circ \vartheta(\mathbf{x}_i))$$

John Ashburner

Geodesic Shooting

æ

#### Matrix Solution

A multigrid approach used to solve  $(\mathbf{A} + \mathbf{H})^{-1}(\mathbf{Aw} + \mathbf{g})$ . Based on a relaxation scheme at several resolutions. See Numerical Recipes (2nd edition onwards) for more details.



Image: A image: A

Differential Operator

Optimisation

Differential Operator Optimisation

#### At The Solution

At the solution, the derivatives of the objective function are zero.

$$rac{\partial \mathcal{E}}{\partial \mathbf{w}} = \mathbf{A}\mathbf{w} + \mathbf{g} = \mathbf{0}$$

Therefore

$$\mathbf{w} = \mathbf{A}^{-1}\mathbf{g}$$

We see that the displacements are a smoothed version of a difference image multiplied by the warped gradients of the template.

$$\mathsf{v} = \mathsf{K}\left((\mathsf{f} - \boldsymbol{\mu}(artheta))((
abla \boldsymbol{\mu}) \circ artheta)
ight)$$

Diffeomorphisms & Geodesics LDDMM Itself

・ 同 ト ・ ヨ ト ・ ヨ ト

# Outline

- 1 Small Deformation Model
  - Differential Operator
  - Optimisation
- 2 Large-Deformation Diffeomorphic Metric Mapping
  - Diffeomorphisms & Geodesics
  - LDDMM Itself

#### 3 Geodesic Shooting

- Generating Deformations from Initial Velocities
- Optimising the Initial Velocities
- Initial Momentum
- Examples

Diffeomorphisms & Geodesics LDDMM Itself

# Nonlinearity of Shapes

According to David Mumford (Fields Medal, 1974):

"Shapes are the ultimate non-linear sort of thing"

Relative shapes can not be added and subtracted (ie, they are nonlinear). Instead, deformations should combined by composing them together.

Deformations that are smooth and invertable are known as *diffeomorphisms*, and form a mathematical *group*.



Diffeomorphisms & Geodesics LDDMM Itself

#### Linear Methods for Data on Manifolds

We are dealing with non-Euclidean geometry, so conceptualise the curved spaces as manifolds embedded in higher dimensions.



・ロッ ・ 一 ・ ・ ・ ・

Diffeomorphisms & Geodesics LDDMM Itself

# Diffeomorphism Groups

A group is a set, *G*, together with an operation,  $\circ$ , that combines any two elements *a* and *b* to form another element, denoted  $a \circ b$ . To qualify as a group, the set and operation must satisfy four requirements known as the group axioms:

- **Closure**. For all a, b in G, the result of the operation,  $a \circ b$ , is also in G.
- Associativity. For all a, b and c in G,  $(a \circ b) \circ c = a \circ (b \circ c)$ .
- Identity element. There exists an element Id in G, such that for every element a in G, the equation  $Id \circ a = a \circ Id = a$  holds.
- Inverse element. For each *a* in *G*, there exists an element *b* in *G* such that  $a \circ b = b \circ a = Id$ .

The order in which the group operation is carried out matters for diffeomorphisms  $(a \circ b \neq b \circ a)$ . They are not *abelian*.

Diffeomorphisms & Geodesics LDDMM Itself

#### Tiny Small Deformation Approximation

Positive real numbers form a group under the multiplication operation. We can not multiply numbers together by addition, but if they are very close to 1 (the identity element of the group), then we almost can. For example  $1.3 \times 0.8 = 1.04$ , 1 + (1.3 - 1) + (0.8 - 1) = 1.1 $1.03 \times 0.98 = 1.0094$ , 1 + (1.03 - 1) + (0.98 - 1) = 1.01 $1.003 \times 0.998 = 1.000994$ , 1 + (1.003 - 1) + (0.998 - 1) = 1.001

Similarly, we can treat extremely small displacement fields as almost linear.

Diffeomorphisms & Geodesics LDDMM Itself

イロン 不同 とくほう イロン

3

#### Large Deformations

We can consider a large deformation as the composition of a series of small deformations:

$$\varphi_1 = \left( \mathrm{Id} + \frac{\mathbf{v}_{t_{N-1}}}{N} \right) \circ \left( \mathrm{Id} + \frac{\mathbf{v}_{t_{N-2}}}{N} \right) \circ ... \circ \left( \mathrm{Id} + \frac{\mathbf{v}_{t_1}}{N} \right) \circ \left( \mathrm{Id} + \frac{\mathbf{v}_0}{N} \right)$$

The inverse of the deformation can be computed from:

$$\boldsymbol{\vartheta}_{1} = \left( \mathrm{Id} - \frac{\mathbf{v}_{0}}{N} \right) \circ \left( \mathrm{Id} - \frac{\mathbf{v}_{t_{1}}}{N} \right) \circ \dots \circ \left( \mathrm{Id} - \frac{\mathbf{v}_{t_{N-2}}}{N} \right) \circ \left( \mathrm{Id} - \frac{\mathbf{v}_{t_{N-1}}}{N} \right)$$

Diffeomorphisms & Geodesics LDDMM Itself

#### Geodesic Distances

By modelling the trajectories as piecewise linear, geodesic distances (locally shortest distances along a curved manifold) can be computed using:

$$d = \frac{1}{N} \sum_{n=0}^{N-1} \sqrt{\mathbf{v}_{t_n}^T \mathbf{A} \mathbf{v}_{t_n}} = \frac{1}{N} \sum_{n=0}^{N-1} ||\mathbf{L} \mathbf{v}_{t_n}||$$

If N approaches infinity, the evolution of a deformation may be conceptualised as integrating the following equation:

$$rac{darphi}{dt} = \mathbf{v}_t(arphi)$$

Geodesic distances (from zero) are then measured by:

$$d = \int_{t=0}^{1} ||\mathbf{L}\mathbf{v}_t|| dt$$

- 4 同 6 4 日 6 4 日 6

Diffeomorphisms & Geodesics LDDMM Itself

#### LDDMM

Large Deformation Diffeomorphic Metric Mapping is an image registration algorithm that minimises the following:

$$\mathcal{E} = \frac{1}{2} \int_{t=0}^{1} ||\mathbf{L}\mathbf{v}_t||^2 dt + \frac{1}{2\sigma^2} ||f - \mu\left(\varphi_1^{-1}\right)||^2$$
  
where  $\varphi_0 = \text{Id}, \frac{d\varphi}{dt} = \mathbf{v}_t\left(\varphi_t\right)$ 

The first term minimises the squared distance measure of the deformations, whereas the second term simply minimises the difference between the warped template and the individual scan. The objective is to estimate a series of velocity fields  $(\mathbf{v}_t)$ .

Diffeomorphisms & Geodesics LDDMM Itself

#### Change of Variables

When we warp images, we should usually account for expansion/contraction via a change of variables.

$$\int_{\mathbf{x}\in\varphi(\Omega)}f(\mathbf{x})d\mathbf{x}=\int_{\mathbf{x}\in\Omega}f(\varphi(\mathbf{x}))|(\mathbf{D}\varphi)(\mathbf{x})|d\mathbf{x}$$

where  $(\mathbf{D}\varphi)(\mathbf{x})$  means the Jacobian of  $\varphi$  at  $\mathbf{x}$ .

Diffeomorphisms & Geodesics LDDMM Itself

# Matching Term



The matching term of the objective function is:

$$rac{1}{\sigma^2}\int_{x\in\Omega}(f\circ {f x}-\mu\circ {m arphi}_1^{-1}\circ {f x})^2d{f x}$$

This may be re-written (including a change of variables via  $\varphi_1\circ\varphi_t^{-1})$  as:

$$\frac{1}{\sigma^2}\int_{x\in\Omega}|\mathsf{D}(\varphi_1\circ\varphi_t^{-1})\circ\mathsf{x}|(f\circ\varphi_1\circ\varphi_t^{-1}\circ\mathsf{x}-\mu\circ\varphi_t^{-1}\circ\mathsf{x})^2d\mathsf{x}$$

< ロ > < 同 > < 回 > < 回 >

Diffeomorphisms & Geodesics LDDMM Itself

# Matching Term



After registration.

- $f(\mathbf{x})$  matches  $\mu(\varphi_1^{-1}(\mathbf{x}))$ .
- $\mu(\mathbf{x})$  matches  $f(\varphi_1(\mathbf{x}))$ .
- $f(\varphi_1(\varphi_t^{-1}(\mathbf{x})))$  matches  $\mu(\varphi_t^{-1}(\mathbf{x}))$ .

▲ 同 ▶ → 三 ▶

Derivatives for LDDMM (need to check these equations)

For n = 0..N, the objective function is:

$$\frac{1}{\sigma^2}\int_{x\in\Omega}|\mathsf{D}(\varphi_1\circ\varphi_{\frac{n}{N}}^{-1})\circ\mathsf{x}|(f\circ\varphi_1\circ\varphi_{\frac{n}{N}}^{-1}\circ\mathsf{x}-\mu\circ\varphi_{\frac{n}{N}}^{-1}\circ\mathsf{x})^2d\mathsf{x}|$$

We can create two images  $f' = f \circ \varphi_1 \circ \varphi_{\frac{n}{N}}^{-1}$  and  $\mu' = \mu \circ \varphi_{\frac{n-1}{N}}^{-1}$ , as well as some weights  $p = |\mathbf{D}(\varphi_1 \circ \varphi_{\frac{n}{N}}^{-1})|$ . This (almost) means that the updates at each time step can be treated as small deformations.

$$\mathcal{E}_t = \frac{1}{2} ||\mathbf{L}\mathbf{v}_t||^2 + \frac{1}{2\sigma^2} \int_{\mathbf{x}} p(\mathbf{x}) \left( f'(\mathbf{x}) - \mu'(\mathbf{x} - \frac{\mathbf{v}_t(\mathbf{x})}{N}) \right)^2 d\mathbf{x}$$

Diffeomorphisms & Geodesics LDDMM Itself

## Derivatives for LDDMM

The derivatives for the matching term turn out to be:

$$\frac{1}{\sigma^2} | \mathbf{D}(\varphi_1 \circ \varphi_t^{-1}) | (\nabla(\mu \circ \varphi_t^{-1}))(f \circ \varphi_1 \circ \varphi_t^{-1} - \mu \circ \varphi_t^{-1})$$

Note that:

• 
$$\nabla(\mu \circ \varphi_t^{-1}) = (\mathbf{D}\varphi_t^{-1})^T((\nabla \mu) \circ \varphi_t^{-1})$$
  
•  $|\mathbf{D}(\varphi_1 \circ \varphi_t^{-1})| = |\mathbf{D}\varphi_t^{-1}|(|\mathbf{D}\varphi_1| \circ \varphi_t^{-1}).$   
•  $(f \circ \varphi_1 \circ \varphi_t^{-1} - \mu \circ \varphi_t^{-1}) = (f \circ \varphi_1 - \mu) \circ \varphi_t^{-1}$   
•  $\varphi_t^{-1} = \vartheta_t$ 

So the derivatives can be re-written:

$$\frac{1}{\sigma^2} |\mathbf{D}\vartheta_t| (|\mathbf{D}\varphi_1| \circ \vartheta_t) (\mathbf{D}\vartheta_t)^{\mathsf{T}} ((\nabla \mu) \circ \vartheta_t) \left( (f \circ \varphi_1 - \mu) \circ \vartheta_t \right)$$

$$|\mathbf{D}\vartheta_t| (\mathbf{D}\vartheta_t)^T \left( \left( \frac{1}{\sigma^2} |\mathbf{D}\varphi_1| (\nabla \mu) (f \circ \varphi_1 - \mu) \right) \circ \vartheta_t \right)$$

Diffeomorphisms & Geodesics LDDMM Itself

#### Gradient Descent for LDDMM

LDDMM is optimised via gradient descent on the so called *Hilbert* gradient:

$$\mathbf{v}_{t}^{(i+1)} = \mathbf{v}_{t}^{(i)} - \epsilon \mathbf{v}_{t}^{($$

Note that Gauss-Newton can not easily be used because of covariance between velocities at different times.

Also note that lots of deformations and velocity fields need to be computed, which either involves using loads of memory, or writing loads of temporary files to disk.

イロト イポト イヨト イヨト

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

- 4 同 6 4 日 6 4 日 6

# Outline

- 1 Small Deformation Model
  - Differential Operator
  - Optimisation
- 2 Large-Deformation Diffeomorphic Metric Mapping
  - Diffeomorphisms & Geodesics
  - LDDMM Itself

#### 3 Geodesic Shooting

- Generating Deformations from Initial Velocities
- Optimising the Initial Velocities
- Initial Momentum
- Examples

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

# Euler-Lagrange Equations

At the solution, the derivatives of the objective function are zero, which means that the velocity at any time point is given by:

$$\mathbf{v}_t + \mathbf{K}\left(|\mathbf{D}\vartheta_t| (\mathbf{D}\vartheta_t)^T \left(\left(\frac{1}{\sigma^2}|\mathbf{D}\varphi_1|(\nabla\mu)(f\circ\varphi_1-\mu)\right)\circ\vartheta_t\right)\right) = 0$$

If we introduce something that we'll call initial momentum:

$$\mathbf{u}_0 = \mathbf{A}\mathbf{v}_0 = rac{1}{\sigma^2} |\mathbf{D} oldsymbol{arphi}_1| (
abla \mu) (\mu - f \circ oldsymbol{arphi}_1)$$

we see that the velocity at any time point is determined by the initial momentum (or velocity), according to:

$$\mathbf{v}_t = \mathbf{K} \left( |\mathbf{D} artheta_t| (\mathbf{D} artheta_t)^T (\mathbf{u}_0 \circ artheta_t) 
ight)$$

< 日 > < 同 > < 三 > < 三 >

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

## Geodesic Shooting

This all means that we do not need to estimate a series of velocity fields. We just need to estimate an initial velocity  $(\mathbf{v}_0)$ , from which we compute the initial momentum by  $\mathbf{u}_0 = \mathbf{A}\mathbf{v}_0$ . We set the deformation at time 0 to an identity transform  $(\varphi_0 = Id)$ , and then evolve the following dynamical system for unit time:

$$\frac{d\varphi}{dt} = \mathbf{v}_t(\varphi_t)$$

$$\mathbf{v}_t = \mathsf{K}\left(|\mathbf{D} arphi_t^{-1}| (\mathbf{D} arphi_t^{-1})^{\mathcal{T}} (\mathbf{u}_0 \circ arphi_t^{-1}) 
ight)$$



Geodesic Shooting

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

#### Evolution



John Ashburner

Geodesic Shooting

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

(a)

#### **Objective Function**

LDDMM objective function:

$$\mathcal{E} = rac{1}{2} \int_{t=0}^{1} ||\mathbf{L}\mathbf{v}_t||^2 dt + rac{1}{2\sigma^2} ||f - \mu\left( oldsymbol{arphi}_1^{-1} 
ight)||^2$$

A possible Geodesic Shooting objective function:

$$\mathcal{E} = \frac{1}{2} ||\mathbf{L}\mathbf{v}_0||^2 dt + \frac{1}{2\sigma^2} ||f - \mu\left(\boldsymbol{\varphi}_1^{-1}\right)||^2$$

The objective function I use:

$$\mathcal{E} = rac{1}{2} ||\mathbf{L} \mathbf{v}_0||^2 dt + rac{1}{2\sigma^2} \int_{\mathbf{x} \in \Omega} |(\mathbf{D} arphi_1) \circ \mathbf{x}| \left(f\left(arphi_1(\mathbf{x})
ight) - \mu(\mathbf{x})
ight)^2 d\mathbf{x}$$

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

#### Gradient Descent for Geodesic Shooting

Could simply optimise  $\mathbf{v}_0$  via gradient descent:

$$\mathbf{v}_t^{(i+1)} = \mathbf{v}_t^{(i)} - \epsilon \mathbf{v}_t^{(i)} - \epsilon \mathbf{K} \left( \frac{|\mathbf{D}\varphi_1^{(i)}|}{\sigma^2} (\nabla \mu) (f \circ \varphi_1^{(i)} - \mu) \right)$$

We no longer need save a series of deformations and velocity fields. Note also that Gauss-Newton can now be used because all the parameters are in  $\mathbf{v}_0$  - so no covariance to worry about.

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

< ロ > < 同 > < 回 > < 回 > .

#### Gauss-Newton Algorithm

- Set the initial velocity  $\mathbf{v}_0$  (parameterised by  $\mathbf{w}$ ) to zero.
- Repeat the following until convergence or for a fixed number of iterations
  - Shoot from the initial velocity  $\mathbf{v}_0$  to obtain  $\varphi_1$ .
  - Compute the objective function, and approximate gradient and Hessian (*E*, g and H), using the current φ<sub>1</sub>.
  - Check  $\mathcal{E}$  improves.
  - Gauss-Newton update of the coefficients, which parameterise  $\textbf{v}_0.$

#### Gradients

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

$$\mathbf{g} = \begin{bmatrix} g_{11} \\ g_{21} \\ g_{31} \\ \vdots \\ g_{l1} \\ g_{11} \\ g_{21} \\ g_{31} \\ \vdots \\ g_{l1} \end{bmatrix}$$

$$g_{il} = \frac{1}{\sigma^2} |\mathbf{J}_1^{\boldsymbol{\varphi}}(\mathbf{x}_i)| \left( f(\boldsymbol{\varphi}_1(\mathbf{x}_i)) - \mu(\mathbf{x}_i) \right) \left( (\nabla_l \mu) \circ \mathbf{x}_i \right)$$

John Ashburner

Geodesic Shooting

э

#### Hessian

	[h <sub>111</sub> 0 0	0 <i>h</i> <sub>211</sub> 0	0 0 h <sub>311</sub>	· · · · · · ·	0 0 0	h <sub>121</sub> 0 0	0 <i>h</i> 221 0	0 0 h <sub>321</sub>	· · · · · · ·	0   0   0
	:	÷	÷	·	÷	÷	÷	÷	·	:
ы	0	0	0		h <sub>/11</sub>	0	0	0		h <sub>121</sub>
п =	h <sub>112</sub>	2 0	0		0	h <sub>122</sub>	0	0		0
	0	h <sub>212</sub>	0		0	0	h <sub>222</sub>	0		0
	0	0	h <sub>312</sub>		0	0	0	h <sub>322</sub>		0
	:	÷	÷	۰.	÷	÷	÷	÷	·	:
	L o	0	0		h <sub>112</sub>	0	0	0		h <sub>122</sub>

$$h_{ilm} = \frac{1}{\sigma^2} |\mathbf{J}_1^{\varphi}(\mathbf{x}_i)| ((\nabla_i \mu) \circ \mathbf{x}_i) ((\nabla_m \mu) \circ \mathbf{x}_i)$$

John Ashburner

Geodesic Shooting

æ

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

(日) (同) (日) (日)

#### Initial Momentum

Remember that

$$\mathbf{u}_0 = rac{1}{\sigma^2} |\mathbf{D} arphi_1| (
abla \mu) (\mu - f \circ arphi_1)$$

If a population of subjects are all aligned with the same template image,  $\frac{1}{\sigma^2}(\nabla \mu)$  will be the same for all subjects. Deviations from the template are encoded by the residuals,  $|\mathbf{D}\varphi_1|(\mu - f \circ \varphi_1)$ . This is a scalar field, and in principle is all that is needed (along with the template) to reconstruct the original images.

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

## Evolution



John Ashburner

Geodesic Shooting

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

# Example Images

#### Some example images.



Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

## Residuals

#### Residuals after aligning the example images to a common template.



John Ashburner

Geodesic Shooting

. nac

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

#### Reconstructed Images

#### Images reconstructed using just the template and residuals.



Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

#### Parameterisations



John Ashburner

Geodesic Shooting

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

#### Geodesic Shooting versus Dartel



John Ashburner

Geodesic Shooting

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

#### Convergence of Gauss-Newton



John Ashburner Geodesic Shooting

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

# Template (IBSR18)



John Ashburner

Geodesic Shooting

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

# Label Overlaps (IBSR18)



Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum **Examples** 

# Template (LPBA40)









(日)

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

# Label Overlaps (LPBA40)

#### LPBA40

		_	_	_	Τ.		_			_			-					_	_		1			1				1		1		Т		_	
R lateral orbitofrontal gyrus	E i				- Yr	1.		11	* ±				÷.				÷.				211			- 1				( r. r.		111		1			
H superior occipital gyrus	E :		11	11	16.5	: *	1.1*	:*			11	1.1	12	111			121	111						112	111	111					 	18	111	111	1.1.2
L superior occipital gyrus	E I				111			112	ř.		÷.	Τ.	12								211							200		111		11			
L angular gyrus	F.							7	- 5	1.7	÷+		ьáн																	11.		- 5			
B angular gyrus	L .													н.	•													÷		÷.,		- 1			
R cuneus	F.				- 5 - 6							++	· +	.++	- i - i															4.4.9	$\sim \sim \sim 10$				
L inferior temporal gyrus	H 1								- 6						+ +	•	+0							4				ŝe e		1.1.1		+ 2			
L precuneus	F 1				- 2.1				12					4	H+ -	٠												( t. t.		111		12			
L inferior occipital gyrus	F.												22	112		÷	+-																		
H supramarginal gyrus	<b>-</b> '				111				12				÷÷.	11							201							111		111		11			
L cuneus	E :												л.	11	۲.L																				
L supramarginal gyrus	E								12				17	Ξ.	÷.						200							22.2		111		11			
R middle orbitofrontal gyrus	F.											· +			- 1	- 6	+ 1													1.1.1					
L avrus rectus	⊢.								- 61							+ 🖷	++				10 A 4							i.e.		1.1.1		12			
L middle temporal gyrus	F -								$\sim 2\pi$						+ +	- ÷												1.1.1				$\sim 10$			
R precuneus	Ŀ١				- ÷ +											++	•													1.1.1		- 2			
R inferior occipital gyrus	F 1								1.41						1.4	•												* * *							
R middle occipital gyrus	F .				12.1				12.							÷+	÷.											\$1.5.		111		12			
H interior temporal gyrus	F 1															÷.,	. 7	e : :												111					
E middle occipital gyrus	С.				12.2								10			177		14.1	1											 111	 	11			
R parahippocampal gyrus	L .								- 61						1.14	÷		÷.,										i.e.		1		12			
R cinculate gyrus	⊢ .								$\sim \infty$						·+		- +	÷	+-									÷							
R fusiform gyrus	F.				- 6 -												+	+	) + +		-0-1-							( e. e.		1.00		13			
L cingulate gyrus	F .								1.0							++			₩ +		(1,1,1,1)							S 1 1							
L parahippocampal gyrus	F.				- 2.1				12							:++	÷.	::•	•+		211							<u>(</u>		111		12			
L fusiform gyrus	F 1															+· ·	121	<u>*</u> : :	•											111		14			
H middle temporal gyrus					1.1				12	114				111	÷ 1.			<u> </u>	۰.					115				20 C.		111		11			
L posicerinar gyrus	E									17				Τ.,	T. 1	511	1.	4.1.1	**																
B superior parietal gyrus	⊢.																÷		÷÷	÷.								2		÷					
L lingual gyrus	H -																+		++	۰ē															
R postcentral gyrus	F.								1.0				+	+ •	+ -	+					1 I							n + n		(1, 2, 2)					
R inferior frontal gyrus	Ŀ١				- 2.1												· (#		··#	··+•	<u></u>							<u> ( ) ( )</u>		111		1.5			
L inferior frontal gyrus	F 1																	111	<u>+</u> :	*:	•									111		14			
H lingual gyrus	<b>-</b> '				12.1				12				12						+ .	111	•••	. L						111		1.1.1		11			
L superior temporal gyrus	E :																	14.1			11														
E nippocampus	Ε.								12				12				÷		+-	÷	T	1.7						222		 111	 	11			
R caudate	L .																4	. + .		1	et e s		4							÷					
R hippocampus	⊢ .								- 6									- 14			+++ -	· + i	•					i.e.		1.1.1					
R superior temporal gyrus	F.								$\sim 2$													++	•					e		1 + 1		-5			
L caudate	H 1				- 0.1										- +			+ -			<b>€</b> ⊡									111					
R putamen	F .				12.1				12									: <b>†</b> :			-+-		+•		+			22.2		111		11			
L precentral gyrus	F 1				- 2 -											+		÷.:		+ +				110	12.1										
E putamen B ingular cortex	С.				16.3				14									17					Π.3	1.1	27 A.			22.2		111		11			
I middle frontel aurue	С.																						7.2												
B middle frontal gyrus	L .								. ÷.								÷.,				de e		- <b>#</b>	ā.,				÷		1		11			
L insular cortex	⊢.				. ÷ .				с. è.											+ 1		+	7.	÷÷	· •+					÷					
R superior frontal gyrus	H -																							-	-+÷ (	<b>H</b> -				1.00		13			
brainstem	Ŀ١								$+\infty$	• • •								• • •						+ • +	+ -	٠				(1,1)					
L superior frontal gyrus	F.									• • •							- 21				0.1				++	+++		6 M M		1.11		12		• • •	
cerebellum	F 1				10.1				11				11				11				111			111				<u>۱</u> ۰+		 111		11			
	-					_	-	-	-	_			0.00				0.7	-						-	•			05				0.0		-	
d				0	1.55				U.6				0.65				U./				J./5			0.	в		0	.00		 0.9		U.9	2		
																			(	٦v	erla	an													
						_	-	-	_	-	_							_	`	~	011	uμ													
												1	oh	n	Ac	h	111	ne	r			G	201	les	ic	Sh	oot	ine	7						
																						~ 15													

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum **Examples** 

# Convergence (LPBA40)



Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum **Examples** 

# Convergence (LPBA40)



Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

## More GS versus Dartel

To assess the effects of larger displacements, the IBSR40 images were all translated along the anterior-posterior direction by 12mm (8 voxels), and re-registered with the template previously generated from un-translated versions of the data. Reproducibility was assessed from correlation coefficients with original GS velocities:

$$r_{ab} = \frac{\mathbf{w}_a^T \mathbf{A} \mathbf{w}_b}{\sqrt{\mathbf{w}_a^T \mathbf{A} \mathbf{w}_a} \sqrt{\mathbf{w}_b^T \mathbf{A} \mathbf{w}_b}}$$

	No Translations	Poor Estimates	Good Estimates
GS	0.98	0.52	0.98
Dartel	0.84	0.19	0.47

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

#### More GS versus Dartel



John Ashburner

Geodesic Shooting

(日) (同) (日) (日) (日)

## Inverse Consistency

Could have achieved inverse consistency by aligning scans to a template at the "half-way point".

The objective here was to try to achieve inverse consistent registration, but with the template aligned to one of the original scans:

$$\mathcal{E} = \frac{1}{2} \int_{t=0}^{1} ||\mathbf{L}\mathbf{v}_{t}||^{2} dt + \frac{1}{2\sigma^{2}} ||f_{1} - \mu\left(\varphi_{1}^{-1}\right)||^{2} + \frac{1}{2\sigma^{2}} ||f_{0} - \mu||^{2}$$
  
where  $\varphi_{0} = \text{Id}$ ,  $\frac{d\varphi}{dt} = \mathbf{v}_{t}\left(\varphi_{t}\right)$ ,  $\mu = (|\mathbf{J}_{1}^{\varphi}|f_{1}(\varphi_{1}) + f_{0})/(|\mathbf{J}_{1}^{\varphi}| + 1)$ 

Generating Deformations from Initial Velocities Optimising the Initial Velocities Initial Momentum Examples

э

#### Inverse Consistency



John Ashburner Geodesic Shooting