A fast diffeomorphic image registration algorithm

John Ashburner

Wellcome Trust Centre for Neuroimaging, 12 Queen Square, London, WC1N 3BG, UK

This paper describes DARTEL, which is an algorithm for diffeomorphic image registration. It is implemented for both 2D and 3D image registration and has been formulated to include an option for estimating inverse consistent deformations. Nonlinear registration is considered as a local optimisation problem, which is solved using a Levenberg–Marquardt strategy. The necessary matrix solutions are obtained in reasonable time using a multigrid method. A constant Eulerian velocity framework is used, which allows a rapid scaling and squaring method to be used in the computations. DARTEL has been applied to intersubject registration of 471 whole brain images, and the resulting deformations were evaluated in terms of how well they encode the shape information necessary to separate male and female subjects and to predict the ages of the subjects.
© 2007 Elsevier Inc. All rights reserved.

Introduction

At its simplest, image registration involves estimating a smooth, continuous mapping between the points in one image and those in another. The relative shapes of the images can then be determined from the parameters that encode the mapping. The objective is usually to determine the single "best" set of values for these parameters. There are many ways of modelling such mappings, but these fit into two broad categories of parameterisation (Miller et al., 1997).

- The small-deformation framework does not necessarily preserve topology—although if the deformations are relatively small, then it may still be preserved.
- The large-deformation framework generates deformations (diffeomorphisms) that have a number of elegant mathematical properties, such as enforcing the preservation of topology.

Many registration approaches still use a small deformation model. These models parameterise a displacement field (u), which is simply added to an identity transform (x).

$$\Phi(x) = x + u(x) \tag{1}$$

In such parameterisations, the inverse transformation is sometimes approximated by subtracting the displacement. It is worth noting that this is only a very approximate inverse, which fails badly for larger deformations. As shown in Fig. 1, compositions of these forward and "inverse" deformations do not produce an identity transform. Small deformation models do not necessarily enforce a one-to-one mapping, particularly if the model assumes the displacements are drawn from a multivariate Gaussian probability density.

The large-deformation or diffeomorphic setting is a much more elegant framework. A diffeomorphism is a globally one-to-one (objective) smooth and continuous mapping with derivatives that are invertible (i.e. nonzero Jacobian determinant). If the mapping is not diffeomorphic, then topology[1] is not necessarily preserved. A key element of a diffeomorphic setting is that it enforces consistency under compositions of the deformations. A composition of two functions is essentially taking one function of the other in order to produce a new function. For two functions, Φ_2 and Φ_1 this would be denoted by

$$\Phi_2 \circ \Phi_1 \circ x = \Phi_2(\Phi_1(x)) \tag{2}$$

For deformations, the composition operation is achieved by resampling one deformation field by another.[2] If the deformations are diffeomorphic, then the result of the composition will also be diffeomorphic. In reality though, deformations are generally represented discretely with a finite number of parameters, so there may be some small violations—particularly if the composition is done using low degree interpolation methods. Perfect (i.e. infinitely dimensional) diffeomorphisms form a Lie group under the composition operation, as they satisfy the requirements of closure, associativity, inverse and identity (see Fig. 2).

---

[1] The word "topology" is used here in the same sense as in "Topological Properties of Smooth Anatomical Maps" (Christensen et al., 1995).

[2] Particular care is needed when dealing with the boundaries—particularly if the boundary conditions are circulant.

E-mail address: j.ashburner@fil.ion.ac.uk.

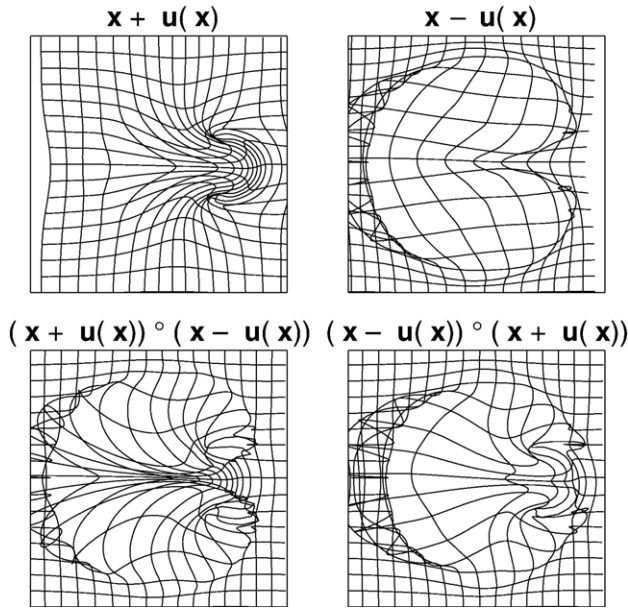**Available online on ScienceDirect (www.sciencedirect.com).**

Fig. 1. Inversion and composition in a small deformation setting. Top-left: a diffeomorphic deformation field. A displacement field was derived by subtracting the identity transform: $u(x)=\Phi(x)-x$. Top-right: an attempt at obtaining an inverse by subtracting the displacement. Although a forward transform may be one-to-one, an inverse obtained by subtracting the displacement may not be. Bottom row: compositions of the forward and "inverse" transformations. If the inverse was correct, then these would both be identity transforms.

The early diffeomorphic registration approaches were based on the greedy "viscous fluid" registration method of Christensen et al. (1994, 1996). In these models, finite difference methods are used to solve the differential equations that model one image as it "flows" to match the shape of the other. At the time, the advantage of these methods was that they were able to account for large displacements while ensuring that the topology of the warped image was preserved. They also provided a useful foundation from which later methods arose. Viscous fluid methods require the solutions to large sets of partial differential equations. The earliest implementations were computationally expensive because solving the equations used successive over-relaxation. Such relaxation methods are inefficient when there are large low frequency components to estimate. Since then, a number of faster ways of solving the differential equations have been devised. These include the use of Fourier transforms to convolve with the impulse response of the linear regularisation operator (Bro-Nielsen and Gramkow, 1996), or by convolving with a separable approximation (Thirion, 1995).

More recent algorithms for large deformation registration aim to find the smoothest possible solution. For example, the LDDMM (large deformation diffeomorphic metric mapping) algorithm (Beg et al., 2005) does not fix the deformation parameters once they have been estimated. It continues to update them using a gradient descent algorithm such that a geodesic distance measure is minimised. In principle, such models could be parameterised by an initial "momentum" field (Miller et al., 2006; Vaillant et al., 2004), which fully specifies how the velocities – and hence the deformations – evolve over unit time. Unfortunately though, the differential equations involved are difficult to work with, and it is easier to parameterise using a number of velocity fields corresponding to

different time periods over the course of the evolution of the diffeomorphism. If $u^{(t)}$ is a velocity field at time t, then the diffeomorphism evolves by

$$\frac{\mathrm{d}\Phi}{\mathrm{d}t} = u^{(t)}\left(\Phi^{(t)}\right) \qquad (3)$$

Diffeomorphisms are generated by initialising with an identity transform ($\Phi^{(0)}=x$) and integrating over unit time to obtain $\Phi^{(1)}$.

The framework described in this paper involves a single flow (velocity) field, which remains constant over unit time. It is similar to the log-Euclidean framework of Arsigny et al. (2006b,a). The algorithm is called DARTEL, standing for "Diffeomorphic Anatomical Registration using Exponentiated Lie algebra".

DARTEL has the advantage, over the small deformation setting, that the resulting deformations are diffeomorphic, easily invertible and can be rapidly computed. It does, however, have a number of disadvantages when compared to variable velocity models. To further understand these limitations, one needs to consider a single point in a brain as the deforming image evolves over unit time. As this point passes different locations of the flow field, then it will be assigned different velocities. Therefore, each of the parameters of such a model will relate to a position in the background space over which the brain deforms, rather than to points within the brain itself. Each voxel in the flow field corresponds to different brain structures at different times during the propagation of the deforming image. Because there is no simple association between a point in the flow field, with a point in the brain, this makes the model parameterisation less ideally suited to computational anatomy studies.

The parameterisation of the variable velocity framework has a more useful physical interpretation, which relates to the velocity of
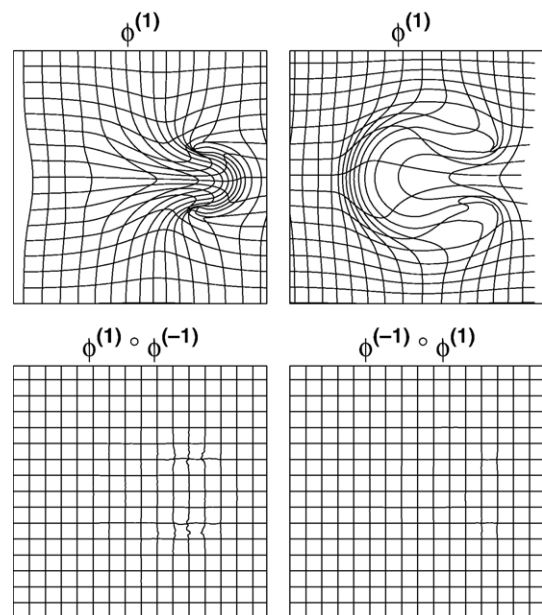

Fig. 2. Inversion and composition in a diffeomorphic setting. Top-left: a forward deformation. Top-right: the corresponding inverse deformation. Both the forward and inverse transforms are one-to-one. Bottom row: compositions of the forward and inverse transformations produce deformations that are close to the identity transform.

each point in the brain at each time during the course of the evolution. Registration involves simultaneously minimising a measure of difference between the image and the warped template, while also minimising an "energy" measure of the deformations used to warp the template. This energy, often thought of as a squared geodesic distance, is obtained by integrating the energy of the velocity fields over unit time. The fixed velocity field used by DARTEL has to encode the whole trajectory of an evolving diffeomorphism. This constraint may force the diffeomorphism to take very circuitous and high energy trajectories in order to achieve good correspondence between images. In fact, some diffeomorphic configurations, which would easily be achieved if velocities could vary over time, are impossible to reach using DARTEL's constant velocity framework.

A further limitation of the DARTEL model can be seen by registering an image pair and then registering the same image pair, but after first translating one of the images by a few pixels. Providing translations are not explicitly penalised, an ideal registration approach should produce deformation energy measures that are the same in both cases. Unfortunately, this does not happen within the fixed velocity DARTEL framework. Similarly, the shape of the deforming template at particular times during the evolution of the diffeomorphism is not invariant with respect to such an initial translation.

In the Method section, the basic theory behind the constant velocity framework used by DARTEL will be covered. The remainder of this section describes the algorithm that can be used to warp one image to match another. This algorithm involves optimising an objective function that consists of a prior term and a likelihood term. Optimisation is done using a method that uses the first and second derivatives of these terms, with respect to the parameterisation of the deformation. The large number of parameters means that computationally efficient methods are needed for solving the equations, so there is a specific focus on computationally efficient schemes that can handle extremely large, if sparse, matrices. Although the DARTEL model is technically inferior to variable velocity diffeomorphic models, it does have practical advantages in terms of the speed of execution.

The Results and discussion section applies the DARTEL registration scheme to 471 anatomical MR images. The resulting flow fields are used in order to assess the level of internal consistency of the method. The same 471 MR images are also brought into register with a small-deformation model, and the parameterisation of the small-deformation and DARTEL models is compared in terms of how well the information encoded can be used by pattern recognition procedures. A quantitative comparison of fixed velocity DARTEL registration with variable velocity diffeomorphic registration methods will be left for future work.

Method

The DARTEL model assumes a flow field ($\boldsymbol{u}$) that remains constant over time. With this model, the differential equation describing the evolution of a deformation is

$$\frac{\mathrm{d}\boldsymbol{\Phi}}{\mathrm{d}t} = \boldsymbol{u}\left(\boldsymbol{\Phi}^{(t)}\right) \tag{4}$$

Generating a deformation involves starting with an identity transform ($\boldsymbol{\Phi}^{(0)}=\boldsymbol{x}$) and integrating over unit time to obtain $\boldsymbol{\Phi}^{(1)}$. The Euler method is a simple integration approach, which involves computing new solutions after many successive small time-steps (h).

$$\boldsymbol{\Phi}^{(t+h)} = \boldsymbol{\Phi}^{(t)} + h\boldsymbol{u}(\boldsymbol{\Phi}^{(t)}) \tag{5}$$

Each of these Euler steps is equivalent to

$$\boldsymbol{\Phi}^{(t+h)} = (\boldsymbol{x} + h\boldsymbol{u})\circ\boldsymbol{\Phi}^{(t)} \tag{6}$$

The small deformation setting can be conceptualised as an Euler integration with a single time step. The use of a large number of small time steps will produce a more accurate solution, such that the trajectory of the points follows a curved path over unit time (Fig. 3). For example, with eight time steps, the Euler integration method is equivalent to

$$\begin{aligned}
\boldsymbol{\Phi}^{(1/8)} &= \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})/8 \\
\boldsymbol{\Phi}^{(2/8)} &= \boldsymbol{\Phi}^{(1/8)}\circ\boldsymbol{\Phi}^{(1/8)} \\
\boldsymbol{\Phi}^{(3/8)} &= \boldsymbol{\Phi}^{(1/8)}\circ\boldsymbol{\Phi}^{(2/8)} \\
&\vdots \\
\boldsymbol{\Phi}^{(8/8)} &= \boldsymbol{\Phi}^{(1/8)}\circ\boldsymbol{\Phi}^{(7/8)}
\end{aligned} \tag{7}$$

If the number of time steps is a power of two, then the solution can be determined by a scaling and squaring approach (Moler and Van Loan, 2003; Arsigny et al., 2006b,a).

$$\begin{aligned}
\boldsymbol{\Phi}^{(1/8)} &= \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})/8 \\
\boldsymbol{\Phi}^{(1/4)} &= \boldsymbol{\Phi}^{(1/8)}\circ\boldsymbol{\Phi}^{(1/8)} \\
\boldsymbol{\Phi}^{(1/2)} &= \boldsymbol{\Phi}^{(1/4)}\circ\boldsymbol{\Phi}^{(1/4)} \\
\boldsymbol{\Phi}^{(1)} &= \boldsymbol{\Phi}^{(1/2)}\circ\boldsymbol{\Phi}^{(1/2)}
\end{aligned} \tag{8}$$

In practice, rather more than eight time steps would be used to compute a more accurate solution. In Group theory, the flow field may be considered as a member of the *Lie algebra*, which is *exponentiated* to produce a deformation, which is a member of a *Lie group*. A useful
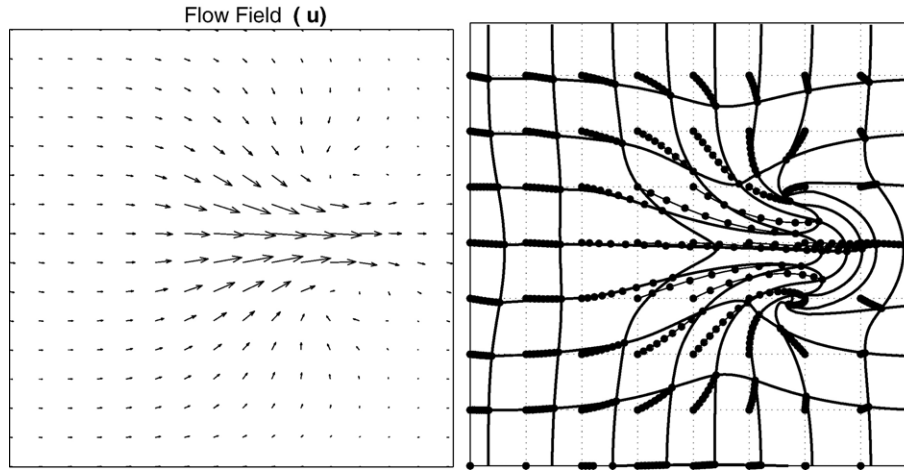
Flow Field (**u**)

Fig. 3. Points follow a curved trajectory as the differential equation is integrated.

heuristic here is that the Jacobian of a deformation that conforms to an exponentiated flow field is always positive (in the same way that the exponential of a real number is always positive). This ensures the mapping is diffeomorphic and, implicitly, assures that the forward and inverse transformations can be generated from the same flow field (Fig. 4):

$$\boldsymbol{\Phi}^{(1)} = \mathrm{Exp}(\boldsymbol{u}) \tag{9}$$

Inverse consistency (Christensen, 1999) is an area of interest within the field of image registration. The extreme case of an inconsistency between a forward and inverse transformation is when the one-to-one mapping between the images breaks down. This can be avoided by using a framework that is diffeomorphic. In order to implement inverse consistent algorithms, it is useful to be able to integrate backwards as well as forwards (see Fig. 5). The inverse of the spatial transformation $\boldsymbol{\Phi}^{(-1)}$ can be achieved by backward integration

$$\begin{aligned}
\boldsymbol{\Phi}^{(-1/8)} &= \boldsymbol{x} - \boldsymbol{u}(\boldsymbol{x})/8 \\
\boldsymbol{\Phi}^{(-1/4)} &= \boldsymbol{\Phi}^{(-1/8)} \circ \boldsymbol{\Phi}^{(-1/8)} \\
\boldsymbol{\Phi}^{(-1/2)} &= \boldsymbol{\Phi}^{(-1/4)} \circ \boldsymbol{\Phi}^{(-1/4)} \\
\boldsymbol{\Phi}^{(-1)} &= \boldsymbol{\Phi}^{(-1/2)} \circ \boldsymbol{\Phi}^{(-1/2)}
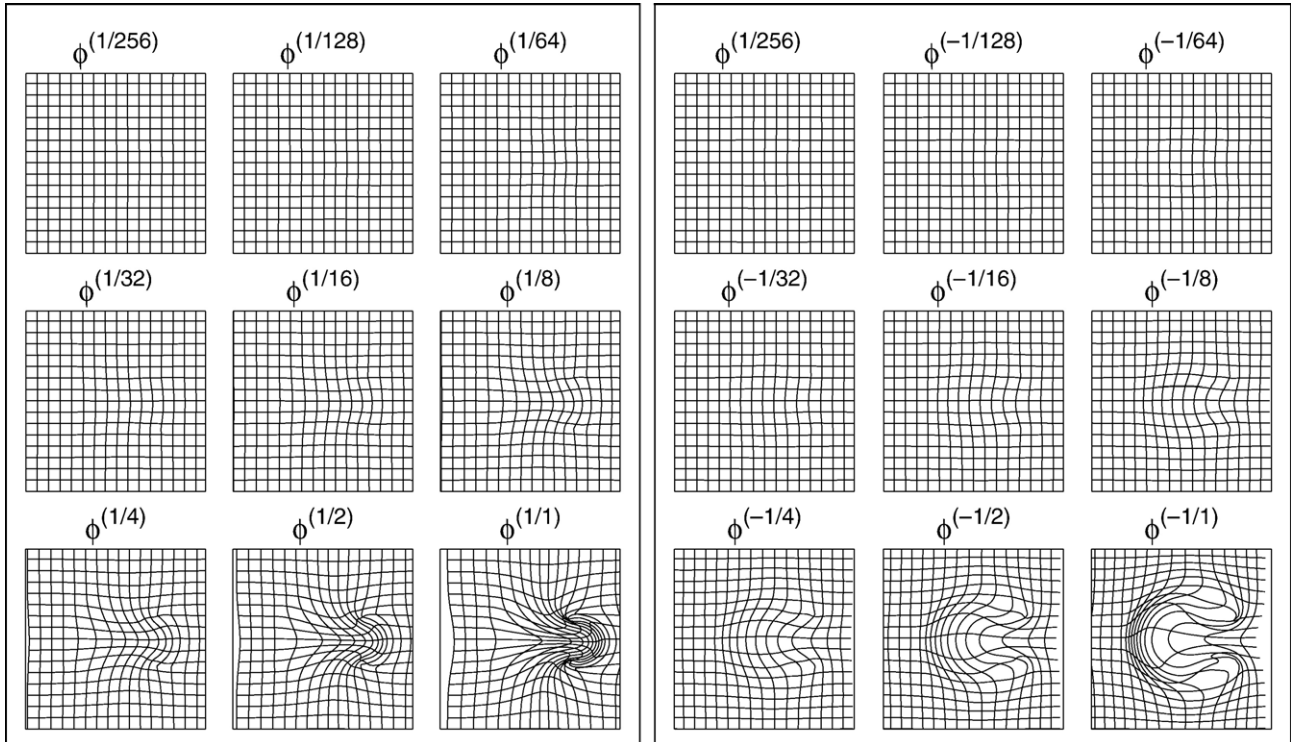\end{aligned} \tag{10}$$

Fig. 4. A scaling and squaring procedure can be used for computing a deformation by exponentiating a flow field (left), as well as an inverse deformation (right).
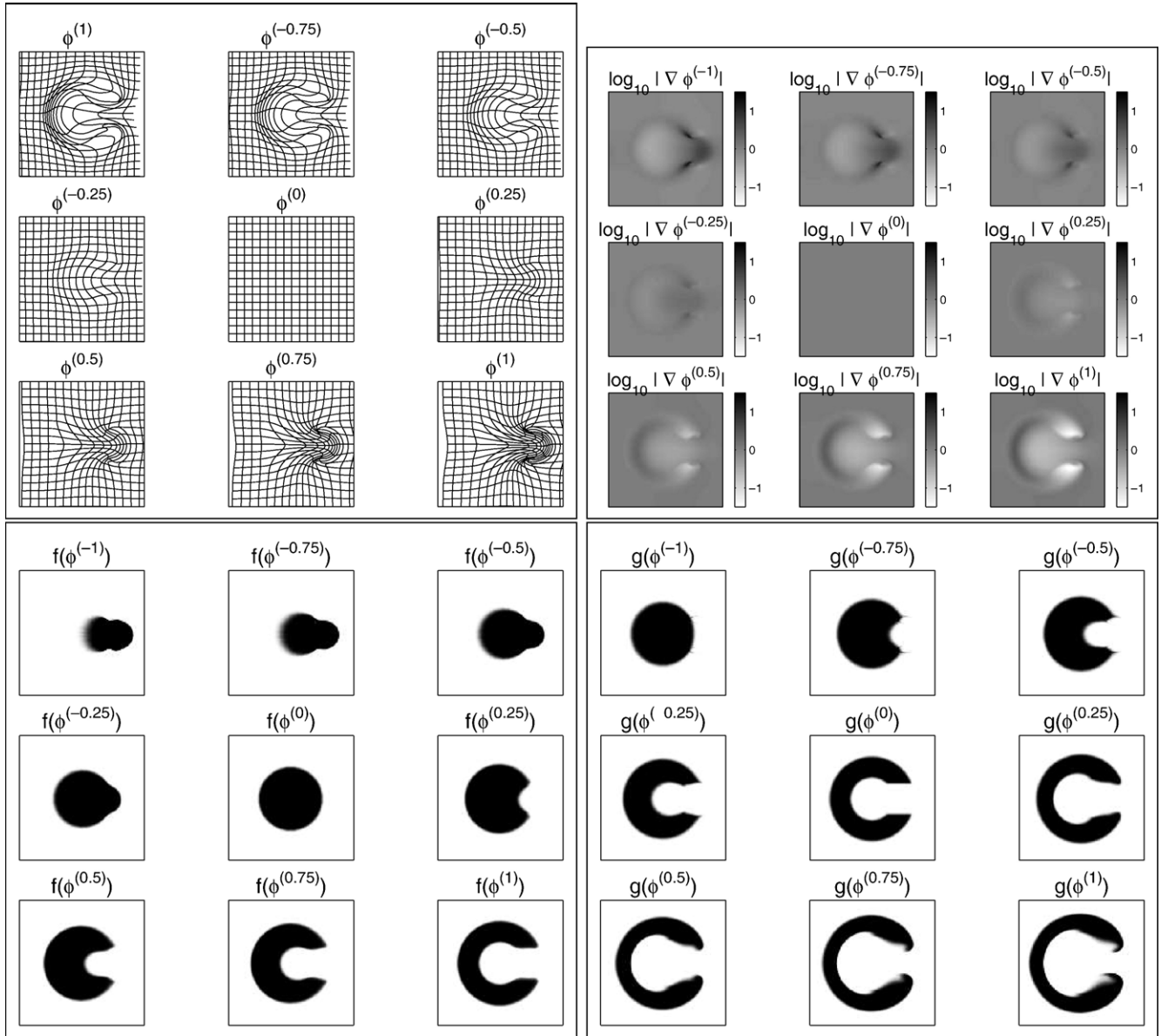
Fig. 5. A deformation at different times (top-left), shown next to the logarithms of the corresponding Jacobian determinants (top-right). A one-to-one mapping is preserved, as illustrated by the Jacobian determinants being greater than zero. The bottom row shows a pair of images transformed with the deformations shown at the top. Note that $f(\boldsymbol{\Phi}^{(0)})$ (the undeformed version) matches $g(\boldsymbol{\Phi}^{(-1)})$ and $f(\boldsymbol{\Phi}^{(1)})$ matches $g(\boldsymbol{\Phi}^{(0)})$. In general, $g(\boldsymbol{\Phi}^{(t)})$ matches $f(\boldsymbol{\Phi}^{(t+1)})$.

If $\boldsymbol{\Phi}^{(0)} = \boldsymbol{x}$ (the identity transform) and sufficient time steps are used, then the following should hold within this framework.

$$\boldsymbol{\Phi}^{(1)} \circ \boldsymbol{\Phi}^{(-1)} = \boldsymbol{\Phi}^{(-1)} \circ \boldsymbol{\Phi}^{(1)} = \boldsymbol{\Phi}^{(0)} \tag{11}$$

The derivatives (*Jacobian* matrices) of the deformations form a second order tensor field.

$$\mathbf{J}_{\boldsymbol{\Phi}}(\boldsymbol{x}) = \left(\nabla \boldsymbol{\Phi}^T\right) \circ \boldsymbol{x} = \begin{pmatrix} \dfrac{\partial \phi_1(\boldsymbol{x})}{\partial x_1} & \dfrac{\partial \phi_1(\boldsymbol{x})}{\partial x_2} & \dfrac{\partial \phi_1(\boldsymbol{x})}{\partial x_3} \\[2mm] \dfrac{\partial \phi_2(\boldsymbol{x})}{\partial x_1} & \dfrac{\partial \phi_2(\boldsymbol{x})}{\partial x_2} & \dfrac{\partial \phi_2(\boldsymbol{x})}{\partial x_3} \\[2mm] \dfrac{\partial \phi_3(\boldsymbol{x})}{\partial x_1} & \dfrac{\partial \phi_3(\boldsymbol{x})}{\partial x_2} & \dfrac{\partial \phi_3(\boldsymbol{x})}{\partial x_3} \end{pmatrix} \tag{12}$$

These Jacobian matrices encode the local stretching, shearing and rotating of the deformation field. Useful measures that can be derived from the matrices are the determinants, which indicate relative volumes before and after spatially transforming. A region of negative determinants would indicate that the one-to-one mapping has been lost.

If $\boldsymbol{\Phi}_C$ is the deformation that results from the composition of two deformations $\boldsymbol{\Phi}_B$ and $\boldsymbol{\Phi}_A$ (i.e. $\boldsymbol{\Phi}_C = \boldsymbol{\Phi}_B \circ \boldsymbol{\Phi}_A$), then the resulting Jacobian field can be obtained by the matrix multiplication $\mathbf{J}_{\boldsymbol{\Phi}_C} = (\mathbf{J}_{\boldsymbol{\Phi}_B} \circ \boldsymbol{\Phi}_A) \, \mathbf{J}_{\boldsymbol{\Phi}_A}$. This leads to a similar scaling and squaring approach that can be used for computing the Jacobian matrices of deformations.

Optimisation

Image registration procedures use a mathematical model to explain the data. Such a model will contain a number of unknown parameters that describe how an image is deformed. A true diffeomorphism has an infinite number of dimensions and is infinitely differential. The implementation described here, and which is used to generate the examples, is based on a finite dimensional approximation for a fixed lattice. Bi- or trilinear interpolation is used so that the functions can be treated as continuous, but this renders them differentiable only once. It would be possible to use a higher-degree interpolation (see e.g. Thévenaz et al., 2000), but linear interpolation was used for speed. This discrete parameterisation of the velocity field, $\boldsymbol{u}(\boldsymbol{x})$, can be considered as a linear combination of basis functions

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_i v_i \boldsymbol{\rho}_i(\boldsymbol{x}) \tag{13}$$

where $\boldsymbol{v}$ is a vector of coefficients and $\boldsymbol{\rho}_i(\boldsymbol{x})$ is the ith first degree B-spline basis function at position $\boldsymbol{x}$. The algorithm is implemented so that functions wrap around at the boundary, so as a point disappears off the right side of field of view, it will appear again on the left. Fixed or sliding boundary conditions could also have been used, but boundaries that are completely free to move are precluded because the necessary compositions can not easily be performed.

The aim is to estimate the single "best" set of values for these parameters ($\boldsymbol{v}$). The objective function, which is the measure of "goodness", is formulated as the most probable deformation, given the data (D).

$$p(\boldsymbol{v}|D) = \frac{p(D|\boldsymbol{v})p(\boldsymbol{v})}{p(D)} \tag{14}$$

This *posterior probability* of the parameters given the image data ($p(\boldsymbol{v}|D)$) is proportional to the probability of the image data given the parameters ($p(D|\boldsymbol{v})$—the *likelihood*), times the *prior probability* of the parameters ($p(\boldsymbol{v})$). The probability of the data ($p(D)$) is a constant. The objective is to find the most probable parameter values and not the actual probability density, so this factor is ignored. The single most probable estimate of the parameters is known as the *maximum a posteriori* (MAP) estimate. There is a monotonic relationship between a value and its logarithm and, in practice, the objective function is normally the logarithm of the probability (in which case it is maximised) or the negative logarithm (which is minimised). It can therefore be considered as the sum of two terms: a prior term and a likelihood term.

$$-\log p(\boldsymbol{v}, D) = -\log p(\boldsymbol{v}) - \log p(D|\boldsymbol{v}) \tag{15}$$

or

$$\mathcal{E}(\boldsymbol{v}) = \mathcal{E}_1(\boldsymbol{v}) + \mathcal{E}_2(\boldsymbol{v}) \tag{16}$$

Many nonlinear registration approaches search for a maximum a posteriori (MAP) estimate of the parameters defining the warps, which corresponds to the mode of the probability density. In practice, there are a number of technical difficulties that can preclude a simple Bayesian interpretation of the problem, as probability densities of continuous functions do not really exist. For this reason, it is more straightforward to interpret registration as a minimum energy estimation procedure. There are many optimisation algorithms that try to find the mode, but most of them only perform a local search. It is possible to use relatively simple strategies for fitting models with few parameters, but as the number of parameters increases, the time required to estimate them will increase dramatically.

The Levenberg–Marquardt (LM) algorithm is a very good general purpose optimisation strategy (see Press et al. (1992) for more information). The procedure is a local optimisation, so it needs reasonable initial starting estimates. It uses an iterative scheme to update the parameter estimates in such a way that the objective function is usually improved each time. Each iteration requires the first and second derivatives of the objective function, with respect to the parameters. In the following scheme, $\mathbf{I}$ is an identity matrix and ζ is a scaling factor. The choice of ζ is a trade-off between speed of convergence and stability. A value of zero for ζ gives the Newton–Raphson or Gauss–Newton optimisation scheme, which may be unstable if the probability density is not well approximated by a Gaussian. Increasing ζ will slow down the convergence, but increase the stability of the algorithm.

$$\boldsymbol{v}^{(n+1)} = \boldsymbol{v}^{(n)} - \left(\left. \frac{\partial^2 \mathcal{E}(\boldsymbol{v})}{\partial \boldsymbol{v}^2} \right|_{\boldsymbol{v}^{(n)}} + \zeta \mathbf{I} \right)^{-1} \left. \frac{\partial \mathcal{E}(\boldsymbol{v})}{\partial \boldsymbol{v}} \right|_{\boldsymbol{v}^{(n)}} \tag{17}$$

The prior term and its derivatives

The prior term reflects the prior probability of a deformation occurring—effectively biasing the deformations to be realistic. The probability of the parameterisation of a flow field ($\boldsymbol{v}$) can most easily be approximated by a probability density that is close to a zero-mean

multivariate Gaussian distribution. In the maximum entropy characterisation of Pennec et al. (2006), the matrix $\mathbf{H}$ is known as a concentration matrix and is analogous to the inverse of a covariance matrix. Z is a normalisation constant.

$$p(\mathbf{v}) = \frac{1}{Z}\exp\left(-\frac{1}{2}\mathbf{v}^T\mathbf{H}\mathbf{v}\right) \tag{18}$$

By taking the negative logarithm of this probability, we obtain

$$\mathcal{E}_1(\mathbf{v}) = -\log p(\mathbf{v}) = \log Z + \frac{1}{2}\mathbf{v}^T\mathbf{H}\mathbf{v} \tag{19}$$

The first and second derivatives of $\mathcal{E}_1(\mathbf{v})$, with respect to the parameters, are required for the registration. These are

$$\frac{\partial \mathcal{E}_1}{\partial \mathbf{v}} = \mathbf{H}\mathbf{v} \text{ and } \frac{\partial^2 \mathcal{E}_1}{\partial \mathbf{v}^2} = \mathbf{H} \tag{20}$$

In most implementations, the matrix $\mathbf{H}$ has a simple numerical form that assumes a similar amount of variability in all spatial locations. In reality, the best model of anatomical variability is very likely to differ from region to region (Lester et al., 1999), so a matrix that models nonstationary variability could, in theory, be a more accurate model. If the true variability of the parameters is known (somehow derived from a large number of subjects), then a suitable model could be determined empirically. The choice of prior will influence how the estimated deformations interpolate between features in the images. As this variability is unknown, the implementation of DARTEL is currently able to use a variety of priors (defined by matrix $\mathbf{H}$). These are based on either membrane, bending or linear elastic energy.

- The membrane energy model is also known as the Laplacian model and is given in 3D by

$$\mathcal{E}_1(\mathbf{v}) = \frac{\lambda}{2}\int_{\mathbf{x}\in\Omega}\sum_{i=1}^{3}\sum_{j=1}^{3}\left(\frac{\partial u_i(\mathbf{x})}{\partial x_j}\right)^2 \mathrm{d}x \tag{21}$$

In the above equations, λ is a constant that encodes the amount of variability. Larger values of λ indicate that the flow field should be smoother. The matrix $\mathbf{H}$ is very large and sparse, but because the operation of $\mathbf{H}\mathbf{v}$ is actually a convolution, it is relatively straightforward to compute. The function with which $\mathbf{v}$ is convolved can be derived from the rows of $\mathbf{H}$. For example, in the case of the membrane energy model in two dimensions, $\mathbf{H}\mathbf{v}$ would be obtained by convolving the horizontal and vertical components of $\mathbf{v}$ by

$$\begin{pmatrix} 0 & -\lambda\delta_1^{-2} & 0 \\ -\lambda\delta_2^{-2} & 2\lambda(\delta_1^{-2}+\delta_2^{-2}) & -\lambda\delta_2^{-2} \\ 0 & -\lambda\delta_1^{-2} & 0 \end{pmatrix} \tag{22}$$

where δ_1 is the height of a voxel and δ_2 is the width.

- The bending energy (biharmonic or thin plate model) is given by

$$\mathcal{E}_1(\mathbf{v}) = \frac{\lambda}{2}\int_{\mathbf{x}\in\Omega}\sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\left(\frac{\partial^2 u_i(\mathbf{x})}{\partial x_j \partial x_k}\right)^2 \mathrm{d}x \tag{23}$$

In two dimensions, the multiplication $\mathbf{H}\mathbf{v}$ is obtained by convolving each component of $\mathbf{v}$ with

$$\begin{pmatrix} 0 & 0 & \lambda\delta_1^{-4} & 0 & 0 \\ 0 & 2\lambda\delta_1^{-2}\delta_2^{-2} & -4\lambda\delta_1^{-2}(\delta_1^{-2}+\delta_2^{-2}) & 2\lambda\delta_1^{-2}\delta_2^{-2} & 0 \\ \lambda\delta_2^{-4} & -4\lambda\delta_2^{-2}(\delta_1^{-2}+\delta_2^{-2}) & \lambda(6\delta_1^{-4}+6\delta_2^{-4}+8\delta_1^{-2}\delta_2^{-2}) & -4\lambda\delta_2^{-2}(\delta_1^{-2}+\delta_2^{-2}) & \lambda\delta_2^{-4} \\ 0 & 2\lambda\delta_1^{-2}\delta_2^{-2} & -4\lambda\delta_1^{-2}(\delta_1^{-2}+\delta_2^{-2}) & 2\lambda\delta_1^{-2}\delta_2^{-2} & 0 \\ 0 & 0 & \lambda\delta_1^{-4} & 0 & 0 \end{pmatrix} \tag{24}$$

- The linear elastic energy is given by

$$\mathcal{E}_1(\mathbf{v}) = \frac{1}{2}\int_{\mathbf{x}\in\Omega}\sum_{j=1}^{3}\sum_{k=1}^{3}\left(\lambda\left(\frac{\partial u_j(\mathbf{x})}{\partial x_j}\right)\left(\frac{\partial u_k(\mathbf{x})}{\partial x_k}\right) + \frac{\mu}{2}\left(\frac{\partial u_j(\mathbf{x})}{\partial x_k} + \frac{\partial u_k(\mathbf{x})}{\partial x_j}\right)^2\right)\mathrm{d}x \tag{25}$$

Here, λ encodes the variance of the trace of the Jacobian matrix (the *divergence*) at each point in $\mathbf{v}$. Larger values will tend to cause volumes to be preserved during the transformation. Jacobian matrices can be decomposed into the sum of symmetric and skew-symmetric (anti-symmetric) matrices. The μ parameter encodes the amount of variance in the elements of the symmetric component and this tends toward penalising scaling and shearing, while allowing rotations to occur more freely. Again, the multiplication $\mathbf{H}\mathbf{v}$ is performed as a convolution

operation (see Fig. 6), but it is more complex as it involves mixing computations on the vertical and horizontal components of the flow fields. In order to obtain the convolved vertical component, it is convolved with

$$
\begin{pmatrix}
0 & -(2\mu+\lambda)\delta_1^{-2} & 0 \\
-\mu\delta_2^{-2} & \mu(4\delta_1^{-2}+2\delta_2^{-2})+2\lambda\delta_1^{-2} & -\mu\delta_2^{-2} \\
0 & -(2\mu+\lambda)\delta_1^{-2} & 0
\end{pmatrix}
\tag{26}
$$

and this is added to the horizontal component convolved with

$$
\begin{pmatrix}
-\dfrac{\mu+\lambda}{4}\delta_1^{-1}\delta_2^{-1} & 0 & \dfrac{\mu+\lambda}{4}\delta_1^{-1}\delta_2^{-1} \\
0 & 0 & 0 \\
\dfrac{\mu+\lambda}{4}\delta_1^{-1}\delta_2^{-1} & 0 & -\dfrac{\mu+\lambda}{4}\delta_1^{-1}\delta_2^{-1}
\end{pmatrix}
\tag{27}
$$

The convolved horizontal component is by convolving the vertical component with the array in Eq. (27) and adding it to the horizontal component convolved with

$$
\begin{pmatrix}
0 & -\mu\delta_1^{-2} & 0 \\
-(2\mu+\lambda)\delta_2^{-2} & \mu(4\delta_2^{-2}+2\delta_1^{-2})+2\lambda\delta_2^{-2} & -(2\mu+\lambda)\delta_2^{-2} \\
0 & -\mu\delta_1^{-2} & 0
\end{pmatrix}
\tag{28}
$$

Currently, the best form of regularisation is unknown. Future work will attempt to learn the optimal settings for the priors from image data itself. In principle, this is just a Type-II Maximum Likelihood problem (with Laplace approximations). Unfortunately, there are a number of technical challenges to overcome before the approach could become practically feasible for problems of this scale.

The likelihood term and its derivatives

This section only considers a likelihood term based upon the mean-squared difference between a pair of images. The model assumes that the individual image g is generated from the template image f by

$$
g(\mathbf{x}) = f(\boldsymbol{\Phi}^{(1)}(\mathbf{x})) + \epsilon(\mathbf{x})
\tag{29}
$$

where $\epsilon(\mathbf{x})$ is drawn from a zero mean Gaussian distribution, which is assumed to be independent and identically distributed over voxels. Ignoring the constant terms and assuming the variance of $\epsilon(\mathbf{x})$ is one, the negative log-likelihood is obtained by summing over the centres of the i voxels

$$
\mathcal{E}_2 = \frac{1}{2}\sum_{i=1}^{I}\left(g_i - f_i\left(\boldsymbol{\Phi}^{(1)}\right)\right)^2
\tag{30}
$$

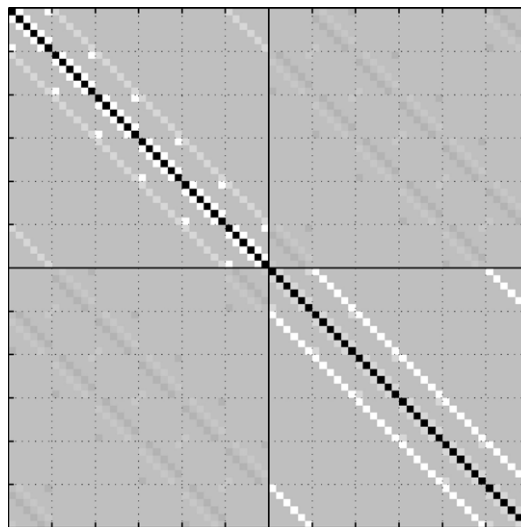where $f_i(\boldsymbol{\Phi}^{(1)})$ denotes the ith voxel of the warped template.

Fig. 6. The **H** matrix for computing the linear elastic energy of a 2D 6×6 flow field, with wrapped boundaries. This matrix uses a value for μ that is twice that used for λ.

For clarity, in what follows, the dependence of the flow and other quantities on location x is dropped. The diffeomorphic mapping, $\boldsymbol{\Phi}^{(1)}$ is the solution to $\dot{\boldsymbol{\Phi}} = \boldsymbol{u}(\boldsymbol{\Phi})$ at unit time. The starting point of the integration is an identity transform ($\boldsymbol{\Phi}^{(0)} = \boldsymbol{x}$). For ease of terminology, this section assumes that the images, flow fields, deformations, etc., are all smooth continuous fields. Implementational details relating to interpolation and boundary conditions are ignored.

The first derivatives of the likelihood term, with respect to changes in velocity are a vector field $\boldsymbol{b}(\boldsymbol{x})$. Within a continuous time representation, the first derivative at any point is given by

$$\boldsymbol{b} = \int_{t=0}^{1} |\mathbf{J}_{\boldsymbol{\Phi}}^{(-t)}| (g^{(-t)} - f^{(1-t)})(\nabla f^{(1-t)})\mathrm{d}t \tag{31}$$

where $g^{(-t)} \equiv g(\boldsymbol{\Phi}^{(-t)})$, and $f^{(1-t)} \equiv f(\boldsymbol{\Phi}^{(1-t)}) \equiv f(\boldsymbol{\Phi}^{(1)} \circ \boldsymbol{\Phi}^{(-t)}) \equiv f(\boldsymbol{\Phi}^{(-t)} \circ \boldsymbol{\Phi}^{(1)})$. The image gradients and Jacobian matrices are denoted by the ∇ and $\mathbf{J}$ operators. At each point in a vector field, there is assumed to be a column vector of values.

The second derivatives can be treated as a symmetric second order tensor field $\mathbf{A}(x)$. Ignoring second derivatives in the image data, these can be obtained in a similar way (see Appendix A).

$$\mathbf{A} = \int_{t=0}^{1} |\mathbf{J}_{\boldsymbol{\Phi}}^{(-t)}| (\nabla f^{(1-t)})^T (\nabla f^{(1-t)})\mathrm{d}t \tag{32}$$

Within a discrete time representation, the registration can be conceptualised as a series of intermediate small deformation registration steps, which are optimised simultaneously. The first and second derivatives are then

$$\boldsymbol{b} = \frac{1}{N} \sum_{n=0}^{N-1} |\mathbf{J}_{\boldsymbol{\Phi}}^{(-n/N)}| \left(g^{(-n/N)} - f^{((N-n)/N)} \right)$$
$$\times \left((\nabla f^{((N-1-n)/N)}) \circ \boldsymbol{\Phi}^{(1/N)} \right) \tag{33}$$

$$\mathbf{A} = \frac{1}{N} \sum_{n=0}^{N-1} |\mathbf{J}_{\boldsymbol{\Phi}}^{(-n/N)}| \left((\nabla f^{((N-1-n)/N)}) \circ \boldsymbol{\Phi}^{(1/N)} \right)^T$$
$$\times \left((\nabla f^{((N-1-n)/N)}) \circ \boldsymbol{\Phi}^{(1/N)} \right) \tag{34}$$

where $g^{(-n/N)} \equiv g(\boldsymbol{\Phi}^{(-n/N)})$. Note that if $N = 1$, these are equivalent to the derivatives for registration within the small-deformation setting.

The DARTEL algorithm uses a recursive procedure for computing an approximation to the derivatives, using a value for N which is a power of two ($N = 2^K$). It begins by first computing $\boldsymbol{\Phi}^{>(1)}$ and the Jacobian matrix $\mathbf{J}_{\boldsymbol{\Phi}}^{(1)}$, according to the current estimates of the flow field. This is done by a scaling and squaring procedure, which begins with a small deformation approximation.

$$\boldsymbol{\Phi}^{(1/2^K)} = \boldsymbol{x} + \frac{1}{2^K}\boldsymbol{u} \tag{35}$$

$$\mathbf{J}_{\boldsymbol{\Phi}}^{(1/2^K)} = \mathbf{I} + \frac{1}{2^K}\mathbf{J}_{\boldsymbol{u}} \tag{36}$$

Then for $k = 0..K-1$ steps, the small deformation approximation is recursively squared in order to generate $\boldsymbol{\Phi}^{(1)}$ and $\mathbf{J}_{\boldsymbol{\Phi}}^{(1)}$.

$$\boldsymbol{\Phi}^{(2^{k+1}/2^K)} = \boldsymbol{\Phi}^{(1)} \circ \boldsymbol{\Phi}^{(2^k/2^K)} \tag{37}$$

$$\mathbf{J}_{\boldsymbol{\Phi}}^{(2^{k+1}/2^K)} = \left((\mathbf{J}_{\boldsymbol{\Phi}}^{(2^k/2^K)}) \circ \boldsymbol{\Phi}^{(2^k/2^K)} \right) \left(\mathbf{J}_{\boldsymbol{\Phi}}^{(2^k/2^K)} \right) \tag{38}$$

The first and second derivatives are initialised by

$$\boldsymbol{b}^{(0)} = \frac{1}{2^K} \left(g^{(0)} - f^{(1)} \right) \mathbf{h} \tag{39}$$

$$\mathbf{A}^{(0)} = \frac{1}{2^K} \mathbf{h}^T \mathbf{h} \tag{40}$$

where

$$\mathbf{h} = \left((\mathbf{J}_{\boldsymbol{\Phi}}^{(1)})(\mathbf{J}_{\boldsymbol{\Phi}}^{(1/2^K)})^{-1} \right)^T \left((\nabla f^{(0)}) \circ \boldsymbol{\Phi}^{(1)} \right) \tag{41}$$

Backward transforms are initialised by

$$\boldsymbol{\Phi}^{(-1/2^K)} = \boldsymbol{x} - \frac{1}{2^K}\boldsymbol{u} \tag{42}$$

$$\mathbf{J}_{\boldsymbol{\Phi}}^{(-1/2^K)} = \mathbf{I} - \frac{1}{2^K}\mathbf{J}_u \tag{43}$$

Then the following are computed recursively for $k=0..K-1$ steps

$$\boldsymbol{b}^{(k+1)} = \boldsymbol{b}^{(k)} + |\mathbf{J}_{\boldsymbol{\Phi}}^{(-2^k/2^K)}| \left(\mathbf{J}_{\boldsymbol{\Phi}}^{(-2^k/2^K)}\right)^T \left(\boldsymbol{b}^{(k)} \circ \boldsymbol{\Phi}^{(-2^k/2^K)}\right) \tag{44}$$

$$\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} + |\mathbf{J}_{\boldsymbol{\Phi}}^{(-2^k/2^K)}| \left(\mathbf{J}_{\boldsymbol{\Phi}}^{(-2^k/2^K)}\right)^T \left(\mathbf{A}^{(k)} \circ \boldsymbol{\Phi}^{(-2^k/2^K)}\right) \left(\mathbf{J}_{\boldsymbol{\Phi}}^{(-2^k/2^K)}\right) \tag{45}$$

$$\boldsymbol{\Phi}^{(-2^{k+1}/2^K)} = \boldsymbol{\Phi}^{(-2^k/2^K)} \circ \boldsymbol{\Phi}^{(-2^k/2^K)} \tag{46}$$

$$\mathbf{J}_{\boldsymbol{\Phi}}^{(-2^{k+1}/2^K)} = \left((\mathbf{J}_{\boldsymbol{\Phi}}^{(-2^k/2^K)}) \circ \boldsymbol{\Phi}^{(-2^k/2^K)}\right)\left(\mathbf{J}_{\boldsymbol{\Phi}}^{(-2^k/2^K)}\right) \tag{47}$$

If $K=0$, the derivatives are exactly equivalent to those used for small deformation registration. Larger values of K produce the derivatives for diffeomorphic registration. In practice, these recursively computed derivatives are only an approximation because of the effect of iteratively resampling (Eqs. (44) and (45)). In particular, it is not really clear how to optimally and efficiently resample (interpolate) the tensor field $\mathbf{A}(x)$ such that the positive definite (and other) properties are best retained (Pennec et al., 2006; Arsigny et al., 2006c). Currently, the individual scalar fields that comprise both $\boldsymbol{b}(x)$ and $\mathbf{A}(x)$ are sampled using trilinear interpolation.

DARTEL has been implemented to include an option for inverse consistent registration. In this formulation, the likelihood part of the objective function is

$$\mathcal{E}_2 = \frac{1}{2}\sum_{i=1}^{I}\left(g_i - f_i(\boldsymbol{\Phi}^{(1)})\right)^2 + \frac{1}{2}\sum_{i=1}^{I}\left(f_i - g_i(\boldsymbol{\Phi}^{(-1)})\right)^2 \tag{48}$$

This inverse consistency was achieved by making the first and second derivatives exactly symmetric by adding them to derivatives computed by integrating the other way. This forward integration is very similar to that shown for backward integration, except for a few small changes. The results of such a formulation are *exactly* inverse consistent spatial transformations.

This section has treated the first and second derivatives as smooth continuous vector and tensor fields. In the next section, the vector field of first derivatives will be treated as a column vector ($\boldsymbol{b}$) and the tensor field of second derivatives as a large sparse matrix ($\mathbf{A}$). This discrete representation corresponds to sampling the fields on a fine regular grid and assumes a good lattice approximation.

Solving the equations

Each Levenberg–Marquardt iteration involves the update

$$\boldsymbol{v}^{(n+1)} = \boldsymbol{v}^{(n)} - (\mathbf{A} + \mathbf{H} + \zeta\mathbf{I})^{-1}(\boldsymbol{b} + \mathbf{H}\boldsymbol{v}^{(n)}) \tag{49}$$

This requires the solution to the following set of equations

$$(\mathbf{A} + \mathbf{H} + \zeta\mathbf{I})^{-1}\left(\boldsymbol{b} + \mathbf{H}\boldsymbol{v}^{(n)}\right) \tag{50}$$

The model is very high-dimensional, so storing a full matrix of second derivatives is not possible because of memory limitations. For this reason, the optimisation uses a method for solving systems of sparse equations. Initial attempts used a conjugate gradient approach (Gilbert et al., 1992), but this was found to be slow. Instead, a full multigrid (FMG) approach (Haber and Modersitzki, 2006) is used to solve the update equations. This is based upon the explanations in Chapter 19 of Press et al. (1992).

FMG approaches are based on relaxation methods, which are performed at multiple scales in order to enhance the speed. Relaxation methods for obtaining a least-squares solution to a set of equations of the form $\mathbf{M}\boldsymbol{w}=\boldsymbol{c}$ involve splitting the matrix into $\mathbf{M}=\mathbf{E}+\mathbf{F}$, where $\mathbf{E}$ is easy to invert and $\mathbf{F}$ is the remainder. The procedures are iterative and involve assigning initial estimates for $\boldsymbol{w}$, and then updating at iteration n according to

$$\boldsymbol{w}^{(n+1)} = \mathbf{E}^{-1}\left(\boldsymbol{c} - \mathbf{F}\boldsymbol{w}^{(n)}\right) \tag{51}$$

Usually, $\mathbf{E}$ is simply a diagonal matrix. Providing $\mathbf{M}$ is strictly diagonally dominant[3], then the updates of Eq. (51) are guaranteed to converge. This is the case when using a membrane energy model for the prior potential.

---

[3] For each row, the magnitude of the diagonal element must be greater than the sum of the magnitudes of the off-diagonal elements.

A different update strategy is required if diagonal dominance conditions are not satisfied, as is the case when modelling the prior potential with bending energy or linear elasticity. This can be derived by re-writing Eq. (51) as

$$w^{(n+1)} = w^{(n)} + \mathbf{E}^{-1}\left(c - \mathbf{F}w^{(n)} - \mathbf{E}w^{(n)}\right) \tag{52}$$

Providing $\mathbf{M}$ is positive definite, then the following regularised update strategy will ensure convergence, where s is chosen to ensure diagonal dominance of $\mathbf{M} + s\mathbf{I}$. This is a similar stabilising strategy to that used by Levenberg–Marquardt optimisation.

$$w^{(n+1)} = w^{(n)} + (\mathbf{E} + s\mathbf{I})^{-1}\left(c - \mathbf{M}w^{(n)}\right) \tag{53}$$

In practice, the updates are performed in place so that the updated values can be used immediately in the current iteration. This is the *Gauss–Seidel* method, as opposed to *Jacobi*'s method, which uses only the values of w from the previous iteration. The Gauss–Seidel method is faster than Jacobi's method and also requires less memory (only one copy of w instead of two). The ordering of the updates of a Gauss–Siedel iteration can be tuned to optimise performance. A red–black checkerboard updating scheme is best if using membrane energy. This involves alternating between updates of all the "red" voxels and then all the "black" voxels. For other prior potential models, the updates can be done by sweeping through w along a variety of different directions.

In most descriptions of relaxation methods, the $\mathbf{E}$ matrix is diagonal, but this is not the case in the current implementation. For volumetric registration, inverting $\mathbf{E}$ consists of inverting a series of symmetric 3×3 matrices, whereas a series of 2×2 matrices would be inverted for a 2D implementation (see Fig. 7).

Relaxation methods take a very long time to estimate the low spatial frequency components of w, whereas the higher frequency components are estimated relatively quickly. Multigrid methods are a way of achieving more rapid convergence by using relaxation methods at various different spatial scales. The full-multigrid (FMG) method is a recursive approach, which involves cycling through the scales. Press et al. (1992) describe the full-multigrid method for solving a relatively simple problem. This algorithm was extended so that the FMG method can be applied to 3D images of any dimensions, with circulant boundary conditions and more complex second derivatives of the types described above. The full details of the approach are omitted, but a brief summary of the procedure is illustrated in Fig. 8 and some of the ideas are elaborated below.

Multigrid methods usually begin by estimating the field at the coarsest scale, and then zooming this coarse estimate to the next higher resolution (*prolongation*). The lower frequencies of the zoomed version tend to be fairly accurate, but the high-frequencies require a few iterations of relaxation to refine them. This refined version is then prolongated to the next higher resolution and so on until the highest resolution solution is reached.

Such a single ascent through the various scales is rarely enough to achieve an accurate solution. Further refinement is needed, and this is obtained by computing the field that needs to be added to w, such that the *defect* (the residuals, $c - \mathbf{M}w$) is minimised. This is achieved by
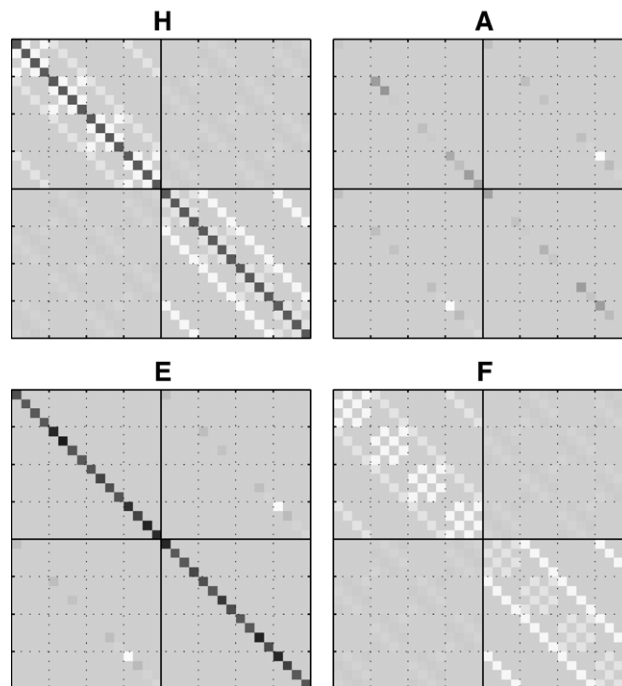
Fig. 7. This figure shows a schematic of the matrices involved in the optimisation. Because of the large dimensions involved, the matrices shown here are only for 2D registration of 4×4 images. Top-left: the $\mathbf{H}$ matrix (for linear elasticity, where $\mu = 1$ and $\lambda = 0$), which is used to regularise the registration. Top-right: the $\mathbf{A}$ matrix that encodes the second derivatives of the likelihood term. Bottom-left: the $\mathbf{E}$ matrix, which contains selected diagonals of $\mathbf{M}$ (where $\mathbf{M}$ consists of $\mathbf{A} + \mathbf{H} + \zeta\mathbf{I}$). Inverting this matrix involves inverting a series of symmetric 2×2 matrices. Bottom-right: the $\mathbf{F}$ matrix ($\mathbf{M} - \mathbf{E}$).

Fig. 8. A schematic of the FMG algorithm for two cycles and four different scales. The algorithm proceeds from left to right. The different heights of the boxes indicate the grid coarseness, where the coarsest grid is at the bottom, and the finest at the top. (A) A coarse solution for w is interpolated up to the resolution of the current grid (*prolongation*), and added to the current estimate for *w*. This solution is refined by a few relaxation iterations, and the residuals (*defect*) computed. This defect is then projected down to a coarser grid (*restriction*) for use as the *c* vector in the next step. (B) A coarse solution for *w* is prolongated to the current grid and added to the current *w*. Relaxation is used to refine the solution, before it is prolonged for use in the next step. (C) The *c* vector is a restricted version of the residuals from the previous step. The initial estimate for *w* is uniformly zero, but this is refined by relaxation. The residuals are computed, and these restricted to a coarser grid for use as the *c* vector in the next step. (E) An exact solution is obtained on the coarsest grid.

computing the defect and restricting it to a coarser grid. At this coarse scale, it serves as the *c* vector. The equations are solved on this grid, and the solution is prolongated back to a finer grid and added to the existing *w*. This step reduces more of the low frequency signal in the defect. A few relaxation steps are then performed in order to reduce the high frequencies. This procedure can be cycled through a number of times, each time reducing the defect.

Solving the equations on the coarse grid would involve restricting the defect to an even coarser grid, solving, prolongating the coarser solution and adding it to the coarse solution. This would be done at all spatial scales, but at the coarsest scale, the solution can be obtained directly because the matrices are much smaller.

Results and discussion

Evaluation of warping methods is a complex area. Generally, the results of an evaluation are specific only to the data used to evaluate the model. MR images vary a great deal with different subjects, field strengths, scanners, sequences, etc., so a model that is good for one set of data may not be appropriate for another. Validation should therefore relate to both the data and the algorithm. The question should be about whether it is appropriate to apply a model to a data set, given the assumptions made by the model.

Very soon, the *Non-rigid Image Registration Evaluation Project* (Christensen et al., 2006) will be ready for use. This framework will allow the developers of nonlinear registration algorithms to compare their algorithms on the same data sets, using the same measures of accuracy. Currently, most developers use their own data and measures to assess registration accuracy, which precludes proper comparisons between competing models.

There are various approaches that are currently used for evaluating registration models. An assessment based on colocalisation of manually defined landmarks would be a useful validation strategy, because it allows the models to be compared with human expertise (Hellier et al., 2001, 2002, 2003). The use of simulated images with known underlying deformations is not really appro-

priate for proper validation of nonlinear registration methods. This is particularly true if the deformation model is the same for the simulations as it is for the registration, because this only illustrates whether or not the optimisation strategy works. Another commonly used form of "evaluation" involves examining the residual difference after registration. Such a strategy would ignore the possibility of over-fitting and tends to favour those models with less regularisation.

The appropriateness of an evaluation depends on the particular application that the deformations are to be used for. For example, if the application was spatial normalisation of functional images of different subjects, then the most appropriate evaluation may be based on assessing the sensitivity of voxel-wise statistical tests (Gee et al., 1997; Miller et al., 2005). Because the warping procedure is based only on structural information, it is blind to the locations of functional activation. If the locations of activations can be brought into close correspondence in different subjects, then it is safe to say that the spatial normalisation procedure works well.

Another application of intersubject registration may involve identifying shape differences among populations of subjects. In this case, the usefulness of the warping algorithm could be assessed by how well the deformation fields can be used to distinguish between the populations (Lao et al., 2004). This approach can be considered as forms of cross-validation, because it

assesses how well the registration helps to predict additional information that it is not explicitly provided with. This is the main evaluation strategy described in this section.

Group-wise registration

Instead of simply matching a pair of images, the objective of intersubject registration is often to align the images of multiple subjects. This is sometimes done by registering all the images with a single template image. Such a procedure would produce different results depending upon the choice of template, so this is an area where internal consistency should be considered. A more optimal template would be some form of average (Avants and Gee, 2004; Davis et al., 2004; Lorenzen et al., 2004). Registering such a template with a brain image generally requires smaller (and therefore less error prone) deformations than would be necessary for registering to an unusually shaped template. Such averages generally lack some of the detail present in the individual subjects' images. Structures that are more difficult to match are generally slightly blurred in the average, whereas those structures that can be more reliably matched are sharper. Such an average generated from a large population of subjects would be ideal for use as a general purpose template.

Four hundred and seventy-one T1 weighted MRI scans were used to create such a template. Details of acquisition parameters, etc. can be found in Good et al. (2001). This experiment used the same 465 scans, plus a few others. The subjects consisted of 264 males and 207 females, with ages ranging from 17 to 79. The mean age was 31.8 (see Fig. 9 for more details).

These data were segmented using the algorithm in SPM5 (Ashburner and Friston, 2005). This procedure includes a component whereby pre-defined tissue probability maps (generated from a large number of subjects) are approximately registered with each image undergoing segmentation. A rigid body transformation was extracted from the nonlinear deformations estimated by the segmentation algorithm using a Procrustes method, weighted by a grey matter tissue probability map (Ashburner et al., 1998). These rigid-body transformations were used to reslice tissue probability images of grey and white matter for each subject, such that they were in approximate alignment. The resliced images were of size $121 \times 145 \times 121$ voxels, and had an isotropic resolution of 1.5 mm.

To illustrate an application of internally consistent registration, the DARTEL algorithm is demonstrated through the construction of average image templates. The scheme involves iterations of DARTEL to map the scans above to their average, to form a new average. This cycle is repeated 18 times in the hope of improving the spatial precision of the average and selecting those features that are conserved and are informative for registering over subjects.

Intensity averages of the grey and white matter images were generated to serve as an initial template for DARTEL registration (see top row of Fig. 10). An inverse-consistent formulation[4] was used to register each individual brain with the template. The likelihood term of the registration was based on the sum of squared difference between the grey matter and grey matter mean, plus that of the white matter and that of the remainder (i.e. one minus grey and white). The prior term was based on linear elasticity, with a value for μ that was 10 times greater than the value for λ. A value for K of 6 was used, which would be analogous to an Euler integration scheme using 64 time points. Registration was done with eighteen Gauss–Newton iterations and, after every three iterations, the mean was re-computed. For the first six iterations, μ was set to 0.5. For the second six, it was 0.25, and for the last six, it was set to 0.125.

The initial template was quite smooth, but it became sharper each time it was re-generated, resulting in a natural coarse-to-fine registration scheme (see Fig. 10). The aim of the heavier regularisation in the early iterations was to avoid some of the potential local minima. Registration of all 471 images took 2 weeks on a standard 3 year old desktop PC[5] Spatially normalised images of selected subjects are shown in Fig. 11.

The whole procedure was repeated in an identical way, except that a small deformation setting was used. All settings were identical, except that K was set to zero in order to achieve small deformation registrations. The resulting displacement fields were later compared with those generated using the diffeomorphic setting.

Exponentiation

Computational precision is finite. For example, using double precision floating point representation (64 bits), a value of about $1 + 2.2 \times 10^{-16}$ is indistinguishable from a value of 1. Similarly, for
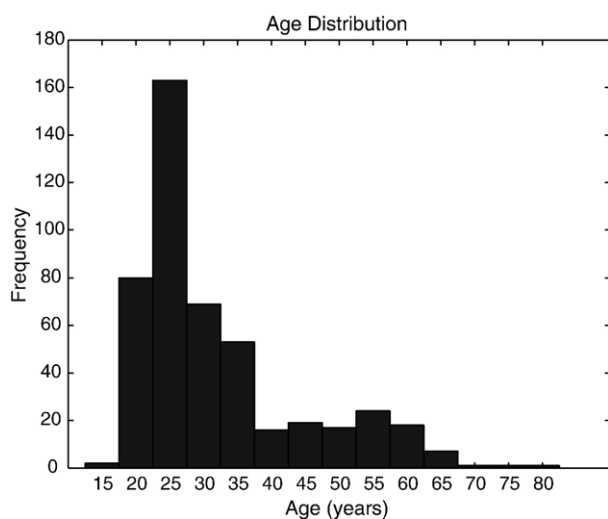
---

[4] From a generative modelling perspective, it would have been more appropriate to use an asymmetric formulation whereby the template was warped to match each individual image. The objective, however, was to demonstrate the ease with which exactly inverse consistent registration could be achieved with DARTEL. Within the functional imaging field, it is also common practice to "spatially normalise" by warping the individual images to match a common template, rather than match the template to the individual images. The recommended strategy would normally be to use the correct asymmetric model.

[5] A single iteration of the asymmetric formulation of DARTEL is rather faster than the symmetric formulation. On the same PC, each iteration (with $K=6$) takes 1 min. An iteration of the small deformation model ($K=0$) is faster than this, taking about 8.7 s. Much of the work in many current registration methods consists of convolving gradients with the Green's function of the regularisation operator. In three-dimensions, this requires six 3D Fourier transforms. To obtain an idea of the speed of the PC, the MATLAB *fftn* function requires 8 s to compute these six Fourier transforms on a $128 \times 128 \times 128$ volume.

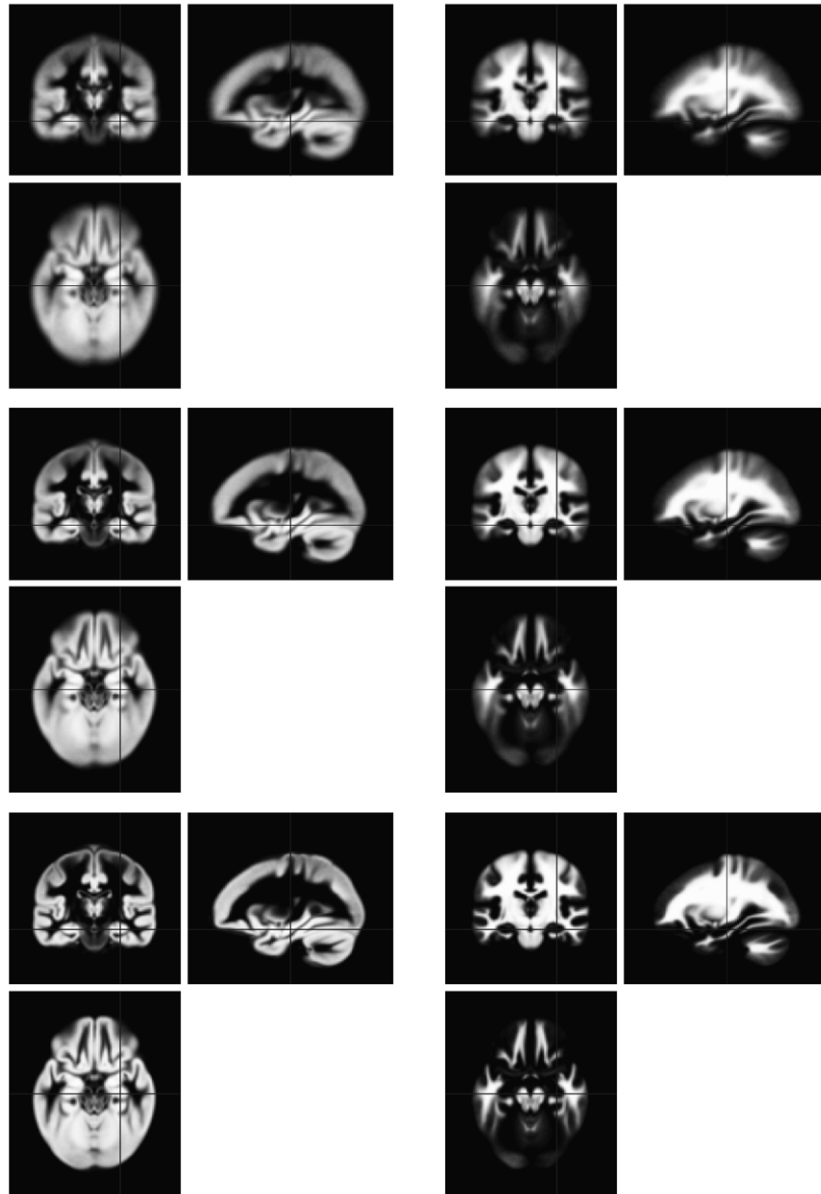Fig. 9. The age distribution of the 471 subjects.

Fig. 10. This figure shows the intensity averages of the grey (left) and white (right) matter images after different numbers of iterations. The top row shows the average after initial rigid-body alignment. The middle row shows the images after three iterations, and the bottom row shows them after all 18 iterations.

single precision representations (32 bits), the relative accuracy is about 1.1×10^{-7}. For this reason, a scaling and squaring algorithm for exponentiating a deformation can only involve squaring a finite number of times. Exponentiating with too many squaring steps leads to numerical problems. The ensuing challenge is to determine a suitable number (K) of steps.

A typical flow field used for matching brains was exponentiated using a range of values of K. Image sampling during each squaring step was done using trilinear interpolation. The root-mean squared (RMS) difference between the deformations derived using K steps and $K-1$ steps was then computed. An optimal value for K was chosen around the point where the RMS difference was minimal. The results are plotted in Fig. 12, showing that, for these data and using single precision floating point representations, a value of around 6 or 7 appears to be optimal (i.e. 64 to 128 time steps).

Inverse consistency

This section assesses the inverse consistency of the deformations. The composition of a transform with its inverse should result in an identity transform. In practice, this is rarely achieved exactly because of the discrete representation of the deformations. The resulting disparity (with the identity transform) was compared with the inverse consistency that would be achieved by using a small deformation approximation.

A typical flow field is exponentiated to produce a forward deformation $\Phi^{(1)}$, and the negative of the flow field is exponentiated to produce the inverse deformation $\Phi^{(-1)}$. Six squaring steps (i.e. 64 time points) were used during the exponentiation. These were composed both ways (i.e. $\Phi^{(1)} \circ \Phi^{(-1)}$ and $\Phi^{(-1)} \circ \Phi^{(1)}$) and the mean and maximum RMS deviation from an identity transform was measured. The RMS differences were found to be
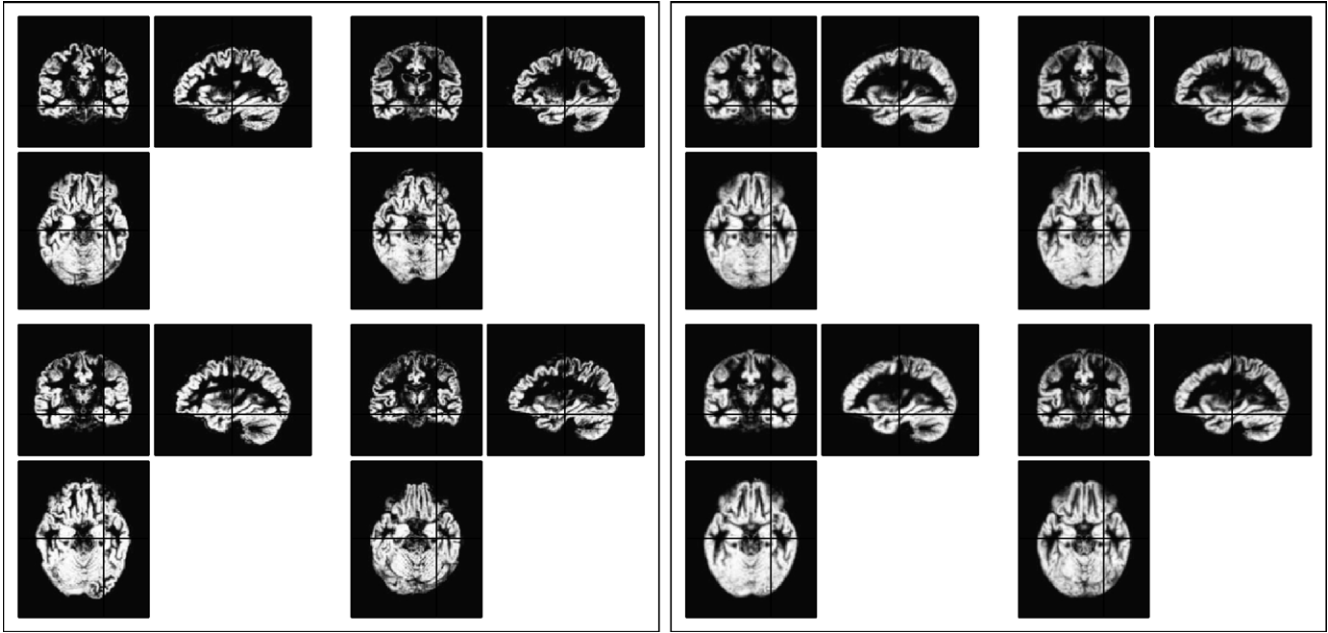
Fig. 11. The left panel shows rigid-body aligned grey matter tissue probability maps of four subjects: an 18 year old female (top left), a 79 year old female (top right), a 17 year old male (bottom left) and a 67 year old male (bottom right). These represent the extremes in age of the subjects. The right panel shows the same subjects data, but after spatial normalisation by warping to their average using the DARTEL algorithm.

0.023 and 0.022 voxels (0.034 and 0.032 mm), and the maximum differences anywhere within the volumes were 0.40 and 0.30 voxels.

A small deformation inverse was generated by $2\boldsymbol{x} - \boldsymbol{\Phi}^{(1)}$, and this was composed both ways with $\boldsymbol{\Phi}^{(1)}$. Similarly, $(2\boldsymbol{x} - \boldsymbol{\Phi}^{(-1)}) \circ \boldsymbol{\Phi}^{(-1)}$ and $\boldsymbol{\Phi}^{(-1)} \circ (2\boldsymbol{x} - \boldsymbol{\Phi}^{(-1)})$ were computed. The RMS deviations of these small deformation approximations from the identity were 0.15, 0.16, 0.16 and 0.17 voxels, and the maximum differences were 2.4, 3.4, 2.5 and 4.0 voxels.

This demonstrates a clear advantage of the current framework over that of the small deformation setting.

Kernel pattern recognition

In this section, we address one aspect of validity using pattern recognition schemes. The idea here is that large-scale deformations should capture or encode relevant and important anatomical features. This means that we can use classification performance as a surrogate measure of the quality of the features encoded by DARTEL. To demonstrate this validation approach, support-vector machines were used to classify images according to sex, and relevance-vector machines to estimate the ages of subjects based upon their images. In brief, the assessment is of whether the diffeomorphic setting will enable pattern recognition approaches to attain better performance, relative to the small-deformation setting. Clearly, this does not represent an exhaustive validation of DARTEL; however, it does show how one can establish the utility of DARTEL in the context of pattern recognition problems.

The first challenge was to predict the sexes of the subjects. An off-the-shelf linear support vector classification (SVC) algorithm[6]

was used (setting C, the regularisation constant, to infinity). The kernel matrix was generated from inner products of the flow fields, such that

$$\mathbf{K} = \mathbf{V}^T \mathbf{H} \mathbf{V} \tag{54}$$

where $\mathbf{V}$ is a matrix, with each column containing the parameters of the flow field for a subject. $\mathbf{H}$ is as in Eq. (19), and encodes linear elasticity with $\mu = 1$ and $\lambda = 0$.

Cross-validation (with smoothing) was used to assess the classification accuracy. This involved training with a random selection of 400 of the subjects, and then making predictions about the remaining 71 subjects. Training and testing were done by picking out the appropriate rows and columns of the $\mathbf{K}$ matrix for the whole data set. Accuracy was assessed by how well the predictions matched known information about those 71 subjects. Cross-validation was repeated 50 times in order to obtain a more precise measure of accuracy.

Nonlinear classification was also performed using a radial basis function (RBF) classifier. The "kernel trick"[7] was used to convert the inner products into distance measures, which were then used to compute the radial basis function kernels. For flow fields parameterised by $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$, the value in the corresponding element of the kernel matrix is

$$\exp\left(-\frac{1}{2\sigma^2} (\boldsymbol{v}_A - \boldsymbol{v}_B)^T \mathbf{H} (\boldsymbol{v}_A - \boldsymbol{v}_B)\right) \tag{55}$$

A range of values for σ^2 were tried, which varied from half the variance of the distances, through to 32 times this variance. Results are shown in Table 1.

---

[6] The quadratic programming algorithm was the implementation of A. J. Smola, using the wrapper written by R. Vanderbei and S. Gunn. It can be downloaded from http://www.isis.ecs.soton.ac.uk/isystems/kernel/.

[7] The "kernel trick" is based on $(\boldsymbol{v}_A - \boldsymbol{v}_B)^T \mathbf{H} (\boldsymbol{v}_A - \boldsymbol{v}_B)$ being equivalent to $\boldsymbol{v}_A^T \mathbf{H} \boldsymbol{v}_A + \boldsymbol{v}_B^T \mathbf{H} \boldsymbol{v}_B - 2 \boldsymbol{v}_A^T \mathbf{H} \boldsymbol{v}_B$, so the required distances can be derived from the inner products. Note that $\mathbf{H}$ is symmetric.
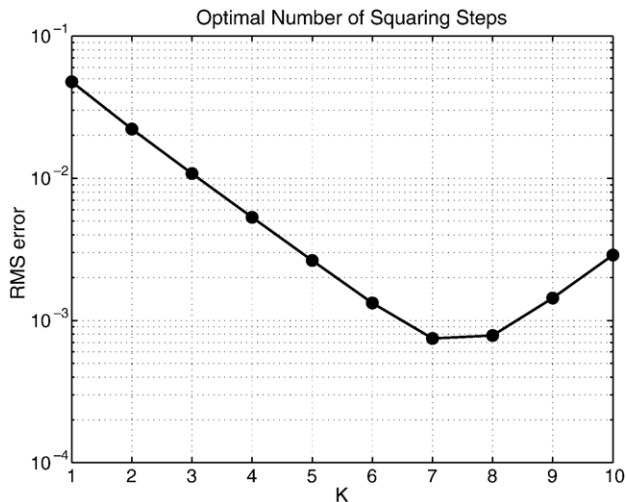
Fig. 12. Determining the optimal number of squaring steps by finding the value of K that produces the lowest RMS difference between deformations generated with K and $K-1$ squaring steps. The RMS difference is given in units of voxels.

A virtually identical procedure was repeated, but using the displacement fields derived from a small deformation setting. The objective was to compare the classification accuracy in the diffeomorphic setting, with the accuracy obtained from a comparable small-deformation model. Cross-validation was done for linear, as well as RBF classification, and the results are shown in Table 2. Overall, the DARTEL registration produced slightly more accurate results than the small deformation model, but the improvement was only in the region of about half of a percent and may not be significant.

The second challenge involved a comparison of how accurately the subjects' ages could be predicted both with and without using the diffeomorphic setting. Relevance-vector regression (Tipping, 2001) was used for making the predictions. This approach is based on kernel matrices similar to those employed by SVMs, and the kernels that were used were the same as those for the sex classification. Cross-validation was performed in a similar way to that for the classification (i.e. repeatedly training with 400 scans and testing with 71—repeating 50 times).

Both linear and RBF regression were performed, both for small deformation and diffeomorphic models, and the results reported as the root mean squared error (in years) and as correlation co-

Table 1
Sex prediction from the diffeomorphic model

	Percent M identified as M	Percent F identified as F	Percent classed as M being M	Percent classed as F being F	Overall percent correct	κ statistic
Linear	91.0	83.6	87.4	88.1	87.7	0.749
RBF 0.5	91.0	80.7	85.5	87.7	86.4	0.722
RBF 1.0	91.1	82.4	86.6	88.1	87.2	0.739
RBF 2.0	91.1	82.9	86.9	88.2	87.5	0.745
RBF 4.0	91.0	83.2	87.1	88.1	87.5	0.746
RBF 8.0	91.0	83.3	87.2	88.1	87.5	0.747
RBF 16	91.0	83.4	87.2	88.1	87.6	0.748
RBF 32	91.0	83.4	87.2	88.2	87.6	0.748

Table 2
Sex prediction from the small deformation model

	Percent M identified as M	Percent F identified as F	Percent classed as M being M	Percent classed as F being F	Overall percent correct	κ statistic
Linear	90.9	82.2	86.4	87.9	87.0	0.736
RBF 0.5	90.9	80.2	85.1	87.6	86.1	0.717
RBF 1.0	90.9	81.7	86.1	87.8	86.8	0.731
RBF 2.0	90.8	82.2	86.4	87.8	87.0	0.734
RBF 4.0	90.8	82.0	86.3	87.7	86.9	0.733
RBF 8.0	90.8	82.2	86.4	87.8	87.0	0.734
RBF 16	90.9	82.3	86.5	87.8	87.0	0.736
RBF 32	90.9	82.3	86.5	87.8	87.1	0.737

efficients. Brain shape changes with age tend to require higher spatial frequency distortions to encode them (cortical thinning, ventricular enlargements, etc.) than the sex effects (total brain size encodes much of the sex differences). This means that predicting the ages of subjects may be a better test for the high-spatial frequency deformations. The results of these tests are presented in Table 3. A plot of true ages, versus estimated ages using the diffeomorphic framework with the optimal RBF regression is shown in Fig. 13. The small deformation model gave slightly better predictions for linear regression, whereas the predictions were slightly more accurate for the diffeomorphic model when a RBF kernel was used. Again, the differences are small and may not be significant.

The constant velocity framework of DARTEL may limit the power of using such flow fields with pattern recognition approaches. Others have suggested a variable velocity framework for computational anatomy, whereby the analyses are based upon "initial momentum" maps (Miller et al., 2006; Vaillant et al., 2004). Future work will evaluate DARTEL with respect to a variable velocity registration strategy, and examine the feasibility of using DARTEL registration results to approximate such initial momentum maps.

Conclusions

In this paper, we have described DARTEL, a principled and efficient diffeomorphic framework for registering images. Optimisation is performed by a Levenberg–Marquardt strategy, and

Table 3
Age prediction accuracy for both the small deformation and diffeomorphic models

	Small deformation		Large deformation	
	RMS error	Correlation	RMS error	Correlation
Linear	7.55	0.826	7.90	0.813
RBF 0.5	7.64	0.816	7.34	0.830
RBF 1.0	7.07	0.842	6.84	0.850
RBF 2.0	6.84	0.851	6.64	0.857
RBF 4.0	6.74	0.854	6.56	0.859
RBF 8.0	6.70	0.856	6.52	0.860
RBF 16	6.68	0.856	6.50	0.861
RBF 32	6.80	0.849	6.64	0.854

The standard deviation of the subjects' ages was 12.24, so the RMS errors all show clear improvements over this figure.
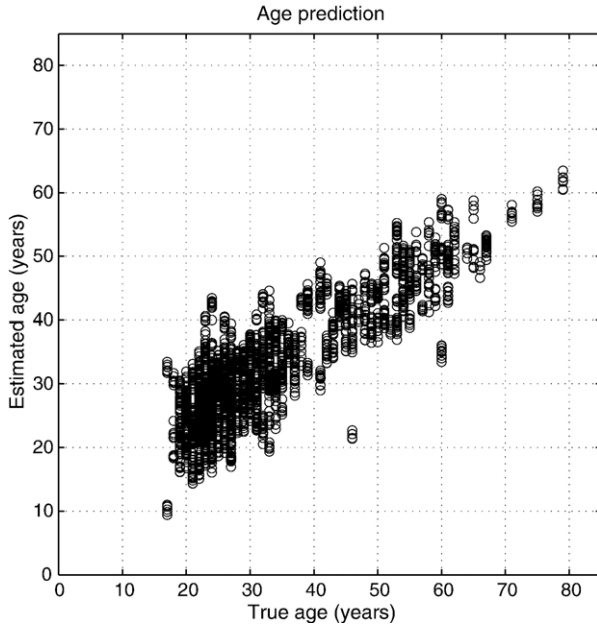
Age prediction

Fig. 13. A plot of true versus estimated ages derived from diffeomorphic flow fields and relevance vector regression (RBF 16).

requires matrix solutions for some very large sparse matrices. The main contribution of this work is the efficient recursive approach used to compute the first and second derivatives used by the optimisation, and the use of a full-multigrid method for solving the equations. This report has focused on underlying theory, the algorithm and operational details.

The performance of this constant velocity diffeomorphic registration scheme has been evaluated in relation to a small-deformation approach, using classification and regression based upon anatomical features encoded by the deformations. The flow fields computed within this constant velocity diffeomorphic framework appeared to confer only a slight advantage for pattern recognition approaches, when compared to displacement fields of a small deformation model.

Acknowledgments

Appendix A. Deriving derivatives

Rigorous derivations of first derivatives in a continuous time representation are given by Beg et al. (2005), but an alternative derivation is provided here. Derivatives are computed with respect to the parameterisation of a flow field (v), from which the mapping $\Phi^{(1)}$ is computed. Within a continuous spatial representation, the objective function is obtained by

$$\mathcal{E}_2 = \frac{1}{2}\int_{x\in\Omega}\Big(g(x) - f(\Phi^{(1)}(x))\Big)^2 \mathrm{d}x \tag{56}$$

The introduction of a second, arbitrary, diffeomorphism (θ), renders the objective function unchanged—provided that the Jacobian determinant of θ is accounted for by a Jacobian change of variables.

$$\mathcal{E}_2 = \frac{1}{2}\int_{x\in\Omega}|\mathbf{J}_\theta(x)|\Big(g(\theta(x)) - f(\Phi^{(1)}(\theta(x)))\Big)^2 \mathrm{d}x \tag{57}$$

Similarly, it can also be obtained by considering the evolution of some θ over time

$$\mathcal{E}_2 = \frac{1}{2}\int_{t=0}^{1}\int_{x\in\Omega}|\mathbf{J}_\theta^{(-t)}(x)|\Big(g(\theta^{(-t)}(x)) - f(\Phi^{(1)}(\theta^{(-t)}(x)))\Big)^2 \mathrm{d}x\mathrm{d}t \tag{58}$$

Within a discrete time representation, a large deformation can be considered as a composition of a series of small deformations. This is analogous to an Euler integration, and becomes increasingly accurate as N, the number of time steps, approaches infinity. First and second derivatives of $\mathcal{E}_2$ will be derived for a variable velocity framework, before constraining the model to constant velocity. In the following, each of the small deformation displacements will be denoted by u_n, where n runs from 0 to $N-1$. The notation $\Phi^{(A,B)}$ is used to denote the composition of $(x+u_A)\circ(x+u_{A-1})\circ\ldots\circ(x+u_B)$. If the number of components is zero, then $\Phi^{(A,B)}$ is simply the identity transform. Similarly, for the evolving second diffeomorphism, $\theta^{(B,A)}$ is used to denote $(x-u_B)\circ(x-u_{B+1})\circ\ldots\circ(x-u_A)$.

$$\mathcal{E}_2 = \frac{1}{2N}\sum_{n=0}^{N-1}\int_{x\in\Omega}|\mathbf{J}_\theta^{(0,n-1)}(x)|\Big(g(\theta^{(0,n-1)}(x))$$
$$-f(\Phi^{(N-1,0)}\circ\theta^{(0,n-1)}\circ x)\Big)^2 \mathrm{d}x \tag{59}$$

For any value of n, $\Phi^{(N-1,0)}$ is equivalent to $\Phi^{(N-1,n+1)}\circ(x+u_n)\circ\Phi^{(n-1,0)}$. Under the assumption of infinitesimally small steps, $(x+u_n)\circ(x-u_n)$ will approach the identity transform, so $\Phi^{(n-1,0)}\circ\theta^{(0,n-1)}$ will also approach the identity.

$$\mathcal{E}_2 = \frac{1}{2N}\sum_{n=0}^{N-1}\int_{x\in\Omega}|\mathbf{J}_\theta^{(0,n-1)}(x)|\Big(g(\theta^{(0,n-1)}(x))$$
$$-f(\Phi^{(N-1,n+1)}\circ(x+u_n(x)))\Big)^2 \mathrm{d}x \tag{60}$$

The discrete parameterisation of a field, $u_n(x)$, is normally by a linear combination of basis functions. Even the so-called free-form models, which usually obtain continuity via trilinear interpolation, are essentially parameterised by a set of first degree B-spline basis functions.

$$u_n(x) = \sum_k v_{kn}\rho_k(x) \tag{61}$$

Therefore

$$\frac{\partial}{\partial v_{in}}f\Big(\Phi^{(N-1,n+1)}\circ(x+u_n(x))\Big)$$
$$= \Big(\Big(\nabla f(\Phi^{(N-1,n+1)})\Big)\circ(x+u_n(x))\Big)^T \rho_i(x) \tag{62}$$

For a variable velocity framework, the first derivatives of $\mathcal{E}_2$ are therefore

$$\frac{\partial \mathcal{E}_2}{\partial v_{in}} = \frac{1}{N} \int_{x \in \Omega} |\mathbf{J}_{\boldsymbol{\theta}}^{(0,n-1)}(\boldsymbol{x})| \left(g(\boldsymbol{\theta}^{(0,n-1)}(\boldsymbol{x})) - f(\boldsymbol{\Phi}^{(N-1,n)}(\boldsymbol{x})) \right)$$
$$\times \left(\left(\nabla f(\boldsymbol{\Phi}^{(N-1,n+1)}) \right) \circ (\boldsymbol{x} + \boldsymbol{u}_n(\boldsymbol{x})) \right)^T \boldsymbol{\rho}_i(\boldsymbol{x}) \mathrm{d}x$$

$$(63)$$

Rather than using the exact second derivatives for optimisation, it is more practical to use an approximation that is guaranteed to be positive definite. This is the approximation used by the Gauss–Newton optimisation algorithm, as opposed to the Newton–Raphson algorithm. Press et al. (1992) says more about the pros and cons of one version over the other.

$$\frac{\partial^2 \mathcal{E}_2}{\partial v_{in} \partial v_{jn}} = \frac{1}{N} \int_{x \in \Omega} |\mathbf{J}_{\boldsymbol{\theta}}^{(0,n-1)}(\boldsymbol{x})| \left(\left(\left(\nabla f(\boldsymbol{\Phi}^{(N-1,n+1)}) \right) \circ (\boldsymbol{x} + \boldsymbol{u}_n(\boldsymbol{x})) \right)^T \boldsymbol{\rho}_i(\boldsymbol{x}) \right)$$
$$\times \left(\left(\left(\nabla f(\boldsymbol{\Phi}^{(N-1,n+1)}) \right) \circ (\boldsymbol{x} + \boldsymbol{u}_n(\boldsymbol{x})) \right)^T \boldsymbol{\rho}_j(\boldsymbol{x}) \right) \mathrm{d}x$$

$$(64)$$

The derivatives for a constant velocity framework are simply obtained by summing over the derivatives that would be used for variable velocity. Note that the notation is changed to the simpler version that can be used for the constant velocity model.

$$\frac{\partial \mathcal{E}_2}{\partial v_i} = \frac{1}{N} \sum_{n=0}^{N-1} \int_{x \in \Omega} |\mathbf{J}_{\boldsymbol{\theta}}^{(-n/N)}| \left(g(\boldsymbol{\Phi}^{(-n/N)}(\boldsymbol{x})) - f(\boldsymbol{\Phi}^{((N-n)/N)}(\boldsymbol{x})) \right)$$
$$\times \left(\left(\nabla f(\boldsymbol{\Phi}^{((N-1-n)/N)}) \right) \circ \boldsymbol{\Phi}^{(1/N)}(\boldsymbol{x}) \right)^T \boldsymbol{\rho}_i(\boldsymbol{x}) \mathrm{d}x$$

$$(65)$$

and

$$\frac{\partial \mathcal{E}_2}{\partial v_i \partial v_j} = \frac{1}{N} \sum_{n=0}^{N-1} \int_{x \in \Omega} |\mathbf{J}_{\boldsymbol{\Phi}}^{(-n/N)}| \left(\left(\left(\nabla f(\boldsymbol{\Phi}^{((N-1-n)/N)}) \right) \circ \boldsymbol{\Phi}^{(1/N)}(\boldsymbol{x}) \right)^T \boldsymbol{\rho}_i(\boldsymbol{x}) \right)$$
$$\times \left(\left(\left(\nabla f(\boldsymbol{\Phi}^{((N-1-n)/N)}) \right) \circ \boldsymbol{\Phi}^{(1/N)}(\boldsymbol{x}) \right)^T \boldsymbol{\rho}_j(\boldsymbol{x}) \right) \mathrm{d}x$$

$$(66)$$

When working with continuous functions, the main text treats the first derivatives as a continuous vector field ($\boldsymbol{b}(x)$), and the second derivatives as a tensor field ($\mathbf{A}(x)$). For the actual optimisation of the parameters ($\boldsymbol{v}$), these derivatives are considered as a vector and a square matrix, respectively. For simplification, the indexing by x is often omitted.

References

Arsigny, V., Commowick, O., Pennec, X., Ayache, N., 2006a. A Log–Euclidean framework for statistics on diffeomorphisms. Proc. of the 9th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI'06), Lecture Notes in Computer Science, 2–4 October 2006a. To appear.

Arsigny, V., Commowick, O., Pennec, X., Ayache, N., 2006b. A Log–Euclidean polyaffine framework for locally rigid or affine registration. In: Pluim, J.P.W., Likar, B., Gerritsen, F.A. (Eds.), Proceedings of the Third International Workshop on Biomedical Image Registration

(WBIR'06), Utrecht, The Netherlands, 9–11 July 2006. Lecture Notes in Computer Science, vol. 4057. Springer Verlag, pp. 120–127.

Arsigny, V., Fillard, P., Pennec, X., Ayache, N., 2006c. Geometric means in a novel vector space structure on symmetric positive-definite matrices. SIAM J. Matrix Anal. Appl. 29 (1), 328–347.

Ashburner, J., Friston, K.J., 2005. Unified segmentation. NeuroImage 26, 839–851.

Ashburner, J., Hutton, C., Frackowiak, R.S.J., Johnsrude, I., Price, C., Friston, K.J., 1998. Identifying global anatomical differences: deformation-based morphometry. Hum. Brain Mapp. 6 (5), 348–357.

Avants, B., Gee, J.C., 2004. Geodesic estimation for large deformation anatomical shape averaging and interpolation. NeuroImage 23, S139–S150.

Beg, M.F., Miller, M.I., Trouvé, A., Younes, L., 2005. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. Int. J. Comput. Vis. 61 (2), 139–157 (February).

Bro-Nielsen, M., Gramkow, C., 1996. Fast fluid registration of medical images. In: Hhne, K-H., Kikinis, R. (Eds.), Proc. Visualization in Biomedical Computing (VBC). Lecture Notes in Computer Science, vol. 1131. Springer-Verlag, Berlin, pp. 267–276.

Christensen, G.E., 1999. Consistent linear elastic transformations for image matching. In: Kuba, A., Samal, M., Todd-Pokropek, A. (Eds.), Proc. Information Processing in Medical Imaging (IPMI). Lecture Notes in Computer Science, vol. 1613. Springer-Verlag, Berlin, pp. 224–237.

Christensen, G.E., Rabbitt, R.D., Miller, M.I., 1994. 3D brain mapping using a deformable neuroanatomy. Phys. Med. Biol. 39, 609–618.

Christensen, G.E., Rabbitt, R.D., Miller, M.I., Joshi, S.C., Grenander, U., Coogan, T.A., Van Essen, D.C., 1995. Topological properties of smooth anatomic maps. In: Bizais, Y., Barillot, C., Di Paola, R. (Eds.), Proc. Information Processing in Medical Imaging (IPMI). Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 101–112.

Christensen, G.E., Rabbitt, R.D., Miller, M.I., 1996. Deformable templates using large deformation kinematics. IEEE Trans. Image Process. 5, 1435–1447.

Christensen, G.E., Geng, X., Kuhl, J.G., Bruss, J., Grabowski, T.J., Pirwani, I.A., Vannier, M.W., Allen, J.S., Damasio, H., 2006. Introduction to the non-rigid image registration evaluation project (NIREP). In: Pluim, J.P.W., Likar, B., Gerritsen, F.A. (Eds.), Proc. .27wThird International Workshop on Biomedical Image Registration. Lecture Notes in Computer Science, vol. 4057. Springer-Verlag, Berlin, pp. 128–135.

Davis, B., Lorenzen, P., Joshi, S., 2004. Large deformation minimum mean squared error template estimation for computational anatomy. Proc. IEEE Int. Symp. Biomed. Imag. (ISBI) 173–176.

Gee, J.C., Alsop, D.C., Aguirre, G.K., 1997. Effect of spatial normalization on analysis of functional data. In: Hanson, K.M. (Ed.), Proc. SPIE Medical Imaging 1997: Image Processing, pp. 312–322.

Gilbert, J.R., Moler, C., Schreiber, R., 1992. Sparse matrices in MATLAB: design and implementation. SIAM J. Matrix Anal. Appl. 13 (1), 333–356 (URL http://citeseer.csail.mit.edu/article/gilbert92sparse.html).

Good, C.D., Johnsrude, I.S., Ashburner, J., Henson, R.N.A., Friston, K.J., Frackowiak, R.S.J., 2001. A voxel-based morphometric study of ageing in 465 normal adult human brains. NeuroImage 14, 21–36.

Haber, E., Modersitzki, J., 2006. A multilevel method for image registration. SIAM J. Sci. Comput. 27, 1594–1607.

Hellier, P., Barillot, C., Corouge, I., Gibaud, B., Le Goualher, G., Collins, D.L., Evans, A.C., Malandain, G., Ayache, N., 2001. Retrospective evaluation of inter-subject brain registration. In: Niessen, W.J., Viergever, M.A. (Eds.), Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI). Lecture Notes in Computer Science, vol. 2208. Springer-Verlag, Berlin, pp. 258–265.

Hellier, P., Ashburner, J., Corouge, I., Barillot, C., Friston, K.J., 2002. Inter subject registration of functional and anatomical data using SPM. Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI). Lecture Notes in Computer Science, vol. 2489. Springer-Verlag, Berlin, pp. 587–590.

Hellier, P., Barillot, C., Corouge, I., Gibaud, B., Le Goualher, G., Collins, D.

L., Evans, A., Malandain, G., Ayache, N., Christensen, G.E., Johnson, H.J., 2003. Retrospective evaluation of inter-subject brain registration. IEEE Trans. Med. Imag. 22 (9), 1120–1130.

Lao, Z., Shen, D., Xue, Z., Karacali, B., Resnick, S.M., Davatzikos, C., 2004. Morphological classification of brains via high-dimensional shape transformations and machine learning methods. NeuroImage 21 (1), 46–57.

Lester, H., Arridge, S.R., Jansons, K.M., Lemieux, L., Hajnal, J.V., Oatridge, A., 1999. Non-linear registration with the variable viscosity fluid algorithm. In: Kuba, A., Samal, M., Todd-Pokropek, A. (Eds.), Proc. Information Processing in Medical Imaging (IPMI). Lecture Notes in Computer Science, vol. 1613. Springer-Verlag, Berlin, pp. 238–251.

Lorenzen, P., Davis, B., Gerig, G., Bullitt, E., Joshi, S., 2004. Multi-class posterior atlas formation via unbiased Kullback–Leibler template estimation. In: Barillot, C., Haynor, D.R., Hellier, P. (Eds.), Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI). Lecture Notes in Computer Science, vol. 3216. Springer-Verlag, Berlin, pp. 95–102.

Miller, M., Banerjee, A., Christensen, G., Joshi, S., Khaneja, N., 1997. Statistical methods in computational anatomy. Stat. Methods Med. Res. 6, 267–299.

Miller, M.I., Beg, M.F., Ceritoglu, C., Stark, C.E.L., 2005. Increasing the power of functional maps of the medial temporal lobe using large deformation metric mapping. Proc. Natl. Acad. Sci. U. S. A. 102, 9685–9690.

Miller, M.I., Trouvé, A., Younes, L., 2006. Geodesic shooting for computational anatomy. J. Math. Imaging Vis. 24 (2), 209–228 (ISSN 0924-9907).

Moler, C., Van Loan, C., 2003. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. SIAM Rev. 45 (1), 3–49.

Pennec, X., Fillard, P., Ayache, N., 2006. A Riemannian framework for tensor computing. Int. J. Comput. Vis. 66 (1), 41–66 (January) URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0920-5691&volume=66&issue=1&spage=41. A preliminary version appeared as INRIA Research Report 5255, July 2004.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes in C, 2nd ed. Cambridge Univ. Press, Cambridge, UK.

Thévenaz, P., Blu, T., Unser, M., 2000. Interpolation revisited. IEEE Trans. Med. Imag. 19 (7), 739–758.

Thirion, J.-P. 1995. Fast non-rigid matching of 3D medical images. Technical Report 2547, Institut National de Recherche en Informatique et en Automatique, May 1995. Available from http://www.inria.fr/rrrt/rr-2547.html.

Tipping, M.E., 2001. Sparse bayesian learning and the relevance vector machine. J. Mach. Learn. Res. 1, 211–244.

Vaillant, M., Miller, M.I., Younes, L., Trouvé, A., 2004. Statistics on diffeomorphisms via tangent space representations. NeuroImage 23, S161–S169.