Collocation for diffeomorphic deformations in medical image registration

Darkner, Sune; Pai, Akshay Sadananda Uppinakudru; Liptrot, Matthew George; Sporring, Jon

Published in:
I E E E Transactions on Pattern Analysis and Machine Intelligence

DOI:
10.1109/TPAMI.2017.2730205

Publication date:
2018

Collocation for Diffeomorphic Deformations in Medical Image Registration

Sune Darkner, Akshay Pai, Matthew G. Liptrot and Jon Sporring

Abstract—Diffeomorphic deformation is a popular choice in medical image registration. A fundamental property of diffeomorphisms is invertibility, implying that once the relation between two points A to B is found, then the relation B to A is given per definition. Consistency is a measure of a numerical algorithm's ability to mimic this invertibility, and achieving consistency has proven to be a challenge for many state-of-the-art algorithms.

We present CDD (Collocation for Diffeomorphic Deformations), a numerical solution to diffeomorphic image registration, which solves for the Stationary Velocity Field (SVF) using an implicit A-stable collocation method. CDD guarantees the preservation of the diffeomorphic properties at all discrete points and is thereby consistent to machine precision.

We compared CDD's collocation method with the following standard methods: Scaling and Squaring, Forward Euler, and Runge-Kutta 4, and found that CDD is up to 9 orders of magnitude more consistent. Finally, we evaluated CDD on a number of standard bench-mark data sets and compared the results with current state-of-the-art methods: SPM-DARTEL, Diffeomorphic Demons and SyN. We found that CDD outperforms state-of-the-art methods in consistency and delivers comparable or superior registration precision.

Index Terms—Registration, Ordinary Differential Equation, Convergence and Stability, Model Validation and Analysis, Image Processing and Computer Vision

1 Introduction

Medical image registration is the process of transforming one anatomy on to another anatomy of the same class. It forms the basis for computational anatomy such as the study of neuroanatomy and the changes hereof across both time and populations. We study the popular class of diffeomorphic transformations induced by ordinary differential equations (ODE's) and their numerical approximation. In the continuous domain, a diffeomorphism is a smooth, i.e., differentiable, one-to-one, invertible mapping of one image domain onto another. However in practice we need to handle discrete images and this work is concerned with explicitly preserving the one-to-one and invertible properties of the discrete solution of the image registration problem. The work presented herein preserves the consistency in diffeomorphic registration. It generates diffeomorphisms with a minimal (to machine precision) difference between the identity mapping (**Id**) and the result of composing a forward (ϕ) with a backward mapping (ψ) ($\|\mathbf{Id} - \psi \circ \phi\| \simeq 0$) [1], [2].

Corresponding author: Sune Darkner (darkner@di.ku.dk)
Sune Darkner, Akshay Pai, Matthew G. Liptrot and Jon Sporring are with the Department of Computer Science, University of Copenhagen

Without consistency a solution is ambiguous. One example is the automatic transport of labels between two images $\boldsymbol{I}$ and $\boldsymbol{J}$, where the choice of deformation, $\boldsymbol{I} \circ \boldsymbol{\phi} \sim \boldsymbol{J}$ or $\boldsymbol{J} \circ \boldsymbol{\psi} \sim \boldsymbol{I}$ under some similarity measure, is ambiguous. In contrast, consistent registration overcomes this ambiguity as only one invertible mapping exists $\boldsymbol{I} \circ \boldsymbol{\phi} \sim \boldsymbol{J}$ and $\boldsymbol{J} \circ \boldsymbol{\phi}^{-1} \sim \boldsymbol{I}$, where per definition $\mathbf{Id} = \boldsymbol{\phi}^{-1} \circ \boldsymbol{\phi}$. Diffeomorphisms are by definition consistent, thus any lack of consistency is purely related to the numerical approximation of diffeomorphisms.

Diffeomorphisms are often defined by an ODE to which the true solution for the deformation is unknown. Although many state-of-the-art frameworks claim to be diffeomorphic, investigations report significant consistency errors [3], [4], [5], [6], [7], and in some cases divergence [4], [5]. A consequence of poor consistency induces ambiguity in derived averages, metrics and statistics. It is worth noting that high consistency does not imply accurate numerical approximation to the true underlying ODE. In this work we characterize collocation for diffeomorphic deformation as a numerical approximation to the ODE of the diffeomorphism, and which is inherently consistent to machine precision. To this end, we present a simple, stable and accurate registration framework: **CDD** (**C**ollocation for **D**iffeomorphic **D**eformations). The framework uses Normalized Mutual Information [8], Locally Orderless Registration (LOR) [9] and Stationary Velocity Fields (SVF) [10], is fully symmetric and produces consistent diffeomorphic registrations to machine precision.

It is important to clarify the role of symmetry in image registration, and in particular the difference between symmetry and consistency. Symmetry [11] is invariance to the ordering of the input images in the registration, and is a property of the registration model, whilst consistency is a property of the numerical implementation of the deformation. Symmetry may be desirable when no a priori knowledge exists regarding the relationships between the objects to be registered, e.g., for the inter-subject registrations performed in this paper. Asymmetric registration may be useful when considering time sequences of the same object, e.g., a developing pathology over time or multi-modal registration. It has been observed [12] that symmetry improves the numerics of registration by enabling the use of gradient information from both images during optimization. A similar observation was made in [13] for homographies regarding the use of gradient information from both images.

Our main contribution is the use of an implicit A-stable collocation method for solving the ODE of the SVF and the derivation of the partial derivatives of the solution to the ODE with respect to the initial conditions. This approach permits the use of gradient-based optimization. The collocation property of the integration scheme guarantees that the discrete solution reflects the properties of the continuous formulation of the ODE at all discrete evaluation points, regardless of the accuracy of the approximation to the true solution of the ODE. Thus all solutions will have diffeomorphic properties. The proposed framework thereby offers numerical stability with guaranteed convergence and preservation of the properties of the diffeomorphic mapping at all discrete points, and does so up to machine precision for all solutions. This major difference between **CDD** and existing state-of-the-art methods is reflected in the consistency, i.e., the difference between the identity mapping and the result of composing a forward mapping with a backward one. The benefits are 3-fold: consistent mapping, guaranteed convergence and a simple symmetric formulation. Note that the first of these ensures accurate inverse mapping and consistent deformation measures, e.g., a distance measure based on the deformation.

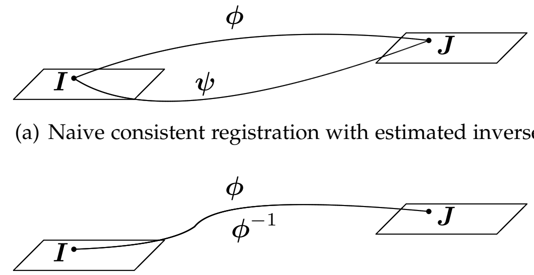
We compared **CDD** to state-of-the-art diffeomorphic alternatives on the standard LPBA [14], MGH10 [15], CUMC12 [16], and IBSR[1] datasets. **CDD** was found to have consistently and significantly higher mean of mean overlap and target overlap across all data sets, significantly higher median of the mean overlap on IBSR, CUMC12 and MGH10, and significantly higher median of the target overlap on the LPBA, CUMC12 and MGH10 datasets while maintaining consistency to machine precision.

We compared the collocation method, using the SVF obtained of the MGH10 dataset, to the Scaling and Squaring, the standard method for integrating SVF's, as well as the standard textbook explicit methods, namely the Forward Euler and the Runge-Kutta 4. We demonstrate that **CDD** is 8-9 orders of magnitude more consistent than both Scaling and Squaring (used by SPM-DARTEL, Diffeomorphic Demons and NiftyReg) and Forward Euler integration schemes, and it is 4-5 orders of magnitude more consistent than the Runge-Kutta 4 scheme. In absolute errors, **CDD** has a maximum inconsistency of approximately 10^{-9} voxels (10^{-11} average), which is in sharp contrast to SPM-DARTEL, which has been reported to have an inconsistency as high as 4 voxels (0.2 average), to Diffeomorphic Demons (approximately 0.5 voxels on average) and SyN (0.2-0.03 voxels on average) [3], [17]. Finally we applied SyN on the MGH dataset and found an inconsistency as high as 2.1 voxels, thus showing that **CDD** offers an improvement in consistency of around 8-9 orders of magnitude over current state-of-the-art.

2 Previous Work

Diffeomorphic mappings [18], which originate from registration based on fluid dynamics [19], are a popular choice for image registration (diffeomorphic registration). They were subsequently developed into the Large Deformation

1. http://www.cma.mgh.harvard.edu/ibsr/



(a) Naive consistent registration with estimated inverse

(b) Diffeomorphic consistent registration used in **CDD**

Fig. 1. A schematic overview of consistent deformation. (a) Naive consistent deformation. This is the approach used in SyN for each deformation to the midpoint for representing the deformation and its inverse by post-registration estimation. This approach is similar to other implementations, where the symmetry is obtained by maintaining 2 deformations with 2 different parameterizations. (b) is the consistency obtained by **CDD**, by accurately using the diffeomorphic property.

Diffeomorphic Metric Mapping (LDDMM) [20] framework. Since the introduction of diffeomorphic registration, several extensions and alternative solutions have been proposed, such as sparse [21] and fast [22] solutions, linear- and hyper elasticity [2], [23] and deformation constraints [24] upon the popular B-spline [25] deformations, used in IRTK and FSL [26], [27].

The consistency of existing ODE based approximations to a diffeomorphism is not guaranteed. Lack of consistency is an indication of a numerical approximation error of the diffeomorphism which then may provide inferior or invalid estimates of desirable properties such as associated metrics. The presented **CDD** framework accurately implements a diffeomorphic deformation model, which guarantees consistency to machine precision.

One popular approach to obtain a diffeomorphism is the stationary velocity field (SVF) [10], which offers easy computation of the inverse by integrating the negated velocity field. Popular SVF-based frameworks include SPM-DARTEL [4] and Diffeomorphic Demons [6], both of which have made SVF a common choice of deformation modeling in clinical studies [28], [29]. These two implementations are based on integration by scaling-and-squaring (SS) [10] also known as the phase flow method [30], a principle that allows for computing the integration of the velocity field (computing the Lie-group exponential) of n Euler steps in $\log_2(n)$ computations. Although an error analysis shows that the SS is asymptotically as good as a standard RK method [31], in practice significant errors [4], [5] and lack of consistency [6], [4], [7], [3] have been reported, including divergence after about 7 squaring operations, equivalent to about 128 regular Euler steps. A general set of strategies for handling inconsistency of a registration includes maintaining two parameterizations [2], [27], [7], penalization [32], [33] or parameterization computed at the midpoint [6], [34], [17]. Computing the deformation at the midpoint produces the deformation $\boldsymbol{\phi}$ and an approximation of its inverse $\boldsymbol{\psi} \approx \boldsymbol{\phi}^{-1}$, but the problem of inferior consistency remains due to an inferior numerical approximation, $(\boldsymbol{\psi} \circ \boldsymbol{\phi} \neq \mathbf{Id})$. In fact the estimated diffeomorphism will differ from the optimization problem due to the inconsistency in the numerical approximation. This numerical inaccuracy can be reduced using higher order schemes, but this also increases numerical complexity. Figure 1 shows consistency in registration using estimated inverses (a) and using the diffeomorphic property (b). One particular approach used in [7] is estimating the inverse after registration. This will produce a consistent (to the precision of the estimation), but not necessarily diffeomorphic mapping (Figure 1(a))

Comparison of the registration accuracy across different registration algorithms is very difficult. A comparison of current state-of-the-art was proposed by Klein et al. [35]. In this study SyN [7], [17], a LDDMM [20] based registration framework, was the best performing method on average. Although that study includes results for SyN [17], SPM-DARTEL [4], IRTK [25], ANIMAL [36], ART [37], AIR [38], ROMEO [39], JRD Fluid [40], SICLE [2], FNIRT [27] and Diffeomorphic Demons [32] we restrict the comparison to the diffeomorphic methods SyN, SPM-DARTEL and Diffeomorphic Demons. Other popular methods include NiftiReg [41] and DRAMMS [42]. DRAMMS was excluded as the method is not diffeomorphic and NiftyReg uses the same integration as both SPM-DARTEL and Diffeomorphic Demons and the consistency should be comparable to these. Interestingly, none of the compared algorithms achieved the best performance across all 4 benchmark datasets, and even the general type of deformation model differed, with ART [35] being homeomorphic and SyN diffeomorphic. Hence, the role of the individual components of the non-rigid registration algorithm is unclear. Herein we address the implementation issues for achieving diffeomorphic registration. In general, registration problems also exist where the use of diffeomorphisms may be debatable, such as pre- to post-surgery of cancer, or registration to a template [43]. However such a discussion is beyond the scope of this paper.

Several numerical schemes for solving ODE's are available [44], both explicit ones such as the Forward Euler and the RK4 from the family of Runge-Kutta methods, as well as implicit ones, e.g., the Backward Euler. A subset of these methods are A-stable. Consider the test equation, $\frac{dy(t)}{dt} = ky(t),\ k \in \mathbb{C}$, which has the solution $y(t) = e^{kt}$, and if $\mathrm{Re}(k) < 0$ for $y(0) = 1$, then $y(t) \to 0$ as $t \to \infty$. If the numerical method exhibits the same properties on this equation, then it is said to be A-stable [44]. An A-stable method guarantees convergence, in contrast to methods that are conditionally stable such as RK4 or Forward Euler, which require a convergence study of every obtained solution to guarantee stability. The implicit trapezoidal method is a standard method for solving ODE's and has been used in various other applications, e.g. [45], [46]. It should not be confused with the trapezoidal rule for integration, as employed by [12] for estimating the norm of the geodesic. The implicit trapezoidal method is a collocation method [47], which ensures that the properties of the integrated ordinary differential equation (ODE) are upheld at each numerical evaluation point in space and time. Therefore, the obtained numerical solution will have the same properties as the exponential of the SVF. This allows for a single parameterization of the deformation and its inverse to machine precision.

CDD uses the implicit trapezoidal method. The use of the trapezoidal method allows us to implement a simple but fully symmetric energy measure, where the similarity measure is Normalized Mutual Information NMI [8] using Locally Orderless Registration (LOR) [9], and the regularization term penalizes the magnitude of the first order derivative of the mapping. We optimize the energy by quasi-Newton gradient descent and provide the analytical derivatives needed. The system parameter is a single SVF for forward and backward mapping, where the backward mapping is obtained by negating the same SVF. Thus, **CDD** is a simple, stable and accurate registration framework, and the consequence is that, in contrast to existing state-of-the-art registration methods, **CDD** accurately implements the consistency property of diffeomorphic registration.

3 Method

The **CDD** pipeline consists of two symmetric registrations: affine registration followed by non-rigid registration. In the following we describe the details of the **CDD** registration framework and it's implementation, including the derivation of the first order information needed for optimization. We start by defining consistency and symmetry in the context of registration.

3.1 Consistent and Symmetric Image registration

We consider only diffeomorphic deformations, which implies that the inverse of a mapping exists and is continuously differentiable. We focus on the numerical schemes and implementations that are invertible to high accuracy, i.e.,

$$\|\boldsymbol{x} - (\boldsymbol{\psi} \circ \boldsymbol{\phi})(\boldsymbol{x})\| \approx 0. \tag{1}$$

We consider the case, where $\boldsymbol{\psi} = \boldsymbol{\phi}^{-1}$, such that $\mathbf{Id} = \boldsymbol{\psi} \circ \boldsymbol{\phi}$. Hence, any deviation from 0 will be due to implementation inaccuracies, and we define *consistency* [1] as the degree to which a numerical scheme approximates the above relation.

Consider two images $\boldsymbol{I}, \boldsymbol{J} : \boldsymbol{\Omega} \to \boldsymbol{\Gamma}$, where $\boldsymbol{\Omega} \subseteq \mathbb{R}^N$ is the image domain, and the value domain typically is $\boldsymbol{\Gamma} \subseteq \mathbb{R}$, and define the registration as the process of seeking a bijective mapping $\boldsymbol{\phi} : \boldsymbol{\Omega} \to \boldsymbol{\Omega}$, such that $\boldsymbol{I} \circ \boldsymbol{\phi}$ improves similarity to $\boldsymbol{J}$ and $\boldsymbol{J} \circ \boldsymbol{\phi}^{-1}$ improves similarity to $\boldsymbol{I}$ under some functional, $\mathcal{F}$, i.e., such that

$$\mathcal{F}(\boldsymbol{I}, \boldsymbol{J}, \boldsymbol{\phi}) = \mathcal{M}(\boldsymbol{I} \circ \boldsymbol{\phi}, \boldsymbol{J}) + \lambda \mathcal{S}(\boldsymbol{\phi}), \tag{2}$$

is minimal. Here $\mathcal{S}$ is a regularization term, $\lambda \in \mathbb{R}$ is a free parameter, and $\mathcal{M}$ is a dissimilarity measure that depends indirectly on $\boldsymbol{\phi}$ in the sense that it measures the dissimilarity between $\boldsymbol{I}$ and $\boldsymbol{J}$ under the mapping $\boldsymbol{\phi}$. We assume that $\mathcal{F}$ is differentiable, and in compliance with much of the literature, we consider the image domain to be continuous and $\boldsymbol{\phi}$ to be diffeomorphic. Given $\boldsymbol{I}$, $\boldsymbol{J}$, and $\boldsymbol{\phi}$, a *symmetric* functional further requires [1],

$$\mathcal{F}_{\text{sym}}(\boldsymbol{I}, \boldsymbol{J}, \boldsymbol{\phi}) = \mathcal{F}_{\text{sym}}(\boldsymbol{J}, \boldsymbol{I}, \boldsymbol{\phi}^{-1}). \tag{3}$$

Symmetry is trivially obtained as,

$$\begin{aligned} \mathcal{F}_{\text{sym}} &= \mathcal{F}(\boldsymbol{I}, \boldsymbol{J}, \boldsymbol{\phi}) + \mathcal{F}(\boldsymbol{J}, \boldsymbol{I}, \boldsymbol{\phi}^{-1}) \\ &= \mathcal{M}(\boldsymbol{I} \circ \boldsymbol{\phi}, \boldsymbol{J}) + \mathcal{S}(\boldsymbol{\phi}) + \mathcal{M}(\boldsymbol{J} \circ \boldsymbol{\phi}^{-1}, \boldsymbol{I}) + \mathcal{S}(\boldsymbol{\phi}^{-1}). \end{aligned} \tag{4}$$

where arguments have been removed for brevity and for any $\mathcal{M}$ and $\mathcal{S}$, which may be verified by insertion.

3.2 Diffeomorphic Mappings Induced by Stationary Velocity Fields (SVF)

A Stationary Velocity Field (SVF), $\boldsymbol{v}$, is a time independent vector field inducing a diffeomorphic mapping $\boldsymbol{\phi}$. Consider pairs of points in the image domain $\boldsymbol{x}_0, \boldsymbol{x}_t \in \Omega, t \in \mathbb{R}_+$, such that $\boldsymbol{x}_t = \boldsymbol{\phi}_t(\boldsymbol{x}_0)$. Here $\boldsymbol{\phi}$ is a family of C^r-diffeomorphic mappings with $r \geq 1$, parametrized by t, and defined as,

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{v}(\boldsymbol{x}),\ \boldsymbol{x}(0) = \boldsymbol{x}_0,\ \boldsymbol{\phi}_t(\boldsymbol{x}_0) = \boldsymbol{x}_t \tag{5}$$

We think of t as time, although it need not have any relation to the physical notion of time. Note that 1) given the mapping, the vector field is found as $\frac{\partial \boldsymbol{\phi}_t(\boldsymbol{x})}{\partial t} = \boldsymbol{v}(\boldsymbol{x})$; 2) the mapping $\boldsymbol{\phi}$ observes the semi-group property that $\boldsymbol{\phi}$ can be realised by n compositions $\boldsymbol{\phi}_T = \boldsymbol{\phi}_{t_n} \circ \boldsymbol{\phi}_{t_{n-1}} \circ \ldots \circ \boldsymbol{\phi}_{t_1}$, where $T = \sum_{i=1}^{n} t_i$; 3) the inverse mapping is found as the integration of $-\boldsymbol{v}$, i.e., for $\boldsymbol{y}_0, \boldsymbol{y}_t \in \Omega$, $\boldsymbol{y}_t = \boldsymbol{\psi}_t(\boldsymbol{y}_0)$,

$$\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} = -\boldsymbol{v}(\boldsymbol{y}),\ \boldsymbol{y}(0) = \boldsymbol{y}_0,\ \boldsymbol{\psi}_t(\boldsymbol{y}_0) = \boldsymbol{y}_t \tag{6}$$

As a consequence $\boldsymbol{x}_0 = (\boldsymbol{\psi}_t \circ \boldsymbol{\phi}_t)(\boldsymbol{x}_0)$, and $\boldsymbol{\phi}_t^{-1} = \boldsymbol{\psi}_t$.

Prior to fitting the SVF, we assume that $\boldsymbol{I}$ and $\boldsymbol{J}$ have been affinely registered using $\mathcal{M}(\boldsymbol{I} \circ \boldsymbol{\phi}, \boldsymbol{J}) + \mathcal{M}(\boldsymbol{J} \circ \boldsymbol{\phi}^{-1}, \boldsymbol{I})$ as the energy measure.

As similarity measure we use the Normalized Mutual Information [8],

$$\mathcal{M}(\boldsymbol{I}, \boldsymbol{J}) = \frac{\mathcal{H}(\boldsymbol{I}) + \mathcal{H}(\boldsymbol{J})}{\mathcal{H}(\boldsymbol{I}, \boldsymbol{J})}, \tag{7}$$

where H are the entropies, $\mathcal{H}(\boldsymbol{I}) = -\int q_{\boldsymbol{I}}(i) \log q_{\boldsymbol{I}}(i)\,\mathrm{d}i$, $\mathcal{H}(\boldsymbol{I}, \boldsymbol{J}) = -\int\int q_{\boldsymbol{I},\boldsymbol{J}}(i,j) \log q_{\boldsymbol{I},\boldsymbol{J}}(i,j)\,\mathrm{d}i\,\mathrm{d}j$, and q are the estimated joint and marginal intensity distributions estimated using Locally Orderless Images [48]. For the regularization term we use,

$$\boldsymbol{S}(\boldsymbol{\phi}) = |\Omega|^{-1} \int_{\Omega} \left\| \frac{\partial \boldsymbol{\phi}_t(\boldsymbol{x})}{\partial t} \right\|_2^2 \mathrm{d}\boldsymbol{x}, \tag{8}$$

which does not regularize the affine term, and trivially gives $\boldsymbol{S}(\boldsymbol{\phi}) = \boldsymbol{S}(\boldsymbol{\phi}^{-1}) = |\Omega|^{-1} \int_{\Omega} \|\boldsymbol{v}(\boldsymbol{x})\|_2^2\,\mathrm{d}\boldsymbol{x}$. Since $\boldsymbol{v}(\boldsymbol{x})$ is independent of time, it follows that $\boldsymbol{S}(\boldsymbol{\phi})$ is to.

3.3 A Consistent Numerical Scheme for Fitting Stationary Velocity Fields

The optimal registration is found by minimising (2) using a quasi-Newton scheme such as LBFGS [49]. To this end, the key issue is to find the derivative of $\mathcal{F}_{\text{sym}}$ in (2) with respect to the deformation. For the sake of notation, consider $\mathcal{M}_{\text{sym}}$ and $\mathcal{M}$ to be functions of two arguments named $(\boldsymbol{U}, \boldsymbol{V})$ such that we may write $\mathcal{M}(\boldsymbol{U}, \boldsymbol{V})$ and likewise for $\boldsymbol{I}(\boldsymbol{x})$, $\boldsymbol{J}(\boldsymbol{x})$, and $\mathcal{S}(\boldsymbol{\theta})$, in that case,

$$\begin{aligned}\frac{\partial \mathcal{F}_{\text{sym}}}{\partial \boldsymbol{\phi}} = {} & \left.\frac{\partial \mathcal{M}}{\partial \boldsymbol{U}}\right|_{\boldsymbol{U}=\boldsymbol{I}\circ\boldsymbol{\phi}} \left.\frac{\partial \boldsymbol{I}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{\phi}} + \left.\frac{\partial \mathcal{S}}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\phi}} \\ & + \left(\left.\frac{\partial \mathcal{M}}{\partial \boldsymbol{U}}\right|_{\boldsymbol{U}=\boldsymbol{J}\circ\boldsymbol{\phi}^{-1}} \left.\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{\phi}^{-1}} + \left.\frac{\partial \mathcal{S}}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\phi}^{-1}} \right) \frac{\partial \boldsymbol{\phi}^{-1}}{\partial \boldsymbol{\phi}}.\end{aligned} \tag{9}$$

A numerically sound and fast scheme for calculating these Jacobians for a range of similarity measures is found in [9]. That work presents a framework for density estimation for image similarity in image registration, from which CDD uses a combination of Normalized Mutual Information as similarity measure, a scale-space formulation of density estimators [48], and a Parzen window approach for image registration.

Our numerical scheme approximates (5) as,

$$\boldsymbol{x}_t = \boldsymbol{\phi}_t(\boldsymbol{x}_{t-1}) \approx \boldsymbol{x}_{t-1} + f(\boldsymbol{v}, \boldsymbol{x}_{t-1}, \Delta t) \tag{10}$$

for some small value of the step size Δt, and where f may depend on both past and future positions. The Forward Euler integration scheme, $f = \boldsymbol{v}(\boldsymbol{x}_{t-1})\Delta t$, is an explicit scheme, which is very simple to implement, has a local approximation order of $\mathcal{O}(\Delta t^2)$, but typically requires very small step-sizes in order to converge. A popular alternative method is the fourth order explicit Runge-Kutta, RK4 [50]. RK4 is algorithmically more complicated, has a local approximation order of $\mathcal{O}(\Delta t^5)$, but only offers marginally improved stability [50]. Implicit integration schemes are algorithmically complicated, but typically offer far better stability than explicit schemes [50]. One such integration scheme is the Trapezoidal method, $f = \frac{1}{2}(\boldsymbol{v}(\boldsymbol{x}_{t-1}) + \boldsymbol{v}(\boldsymbol{x}_t))\Delta t$, which has a local approximation order of $\mathcal{O}(\Delta t^3)$ and is A-stable. In fact, the Trapezoidal method is the most accurate linear multistep method, that is also A-stable [44].

We propose to use the Trapezoidal method for SVF. Approximating,

$$\boldsymbol{v}(\boldsymbol{x}) \approx \boldsymbol{B}(\boldsymbol{x})\boldsymbol{p} \tag{11}$$

using a linear basis $\boldsymbol{B}$ and a K-dimensional parameter vector $\boldsymbol{p}$, we use $S(\boldsymbol{\phi}) \approx \lambda \|\boldsymbol{p}\|_2^2$ to approximate (8) as $S(\boldsymbol{\phi})$ is positive and bounded by $\lambda \|\boldsymbol{p}\|_2^2$ for positive and constant λ. As $\|v(x)\|^2 = \|\boldsymbol{B}(\boldsymbol{x})p\|^2 = \langle (p^t \boldsymbol{B}(\boldsymbol{x})^t), \boldsymbol{B}(\boldsymbol{x})p \rangle = \langle \boldsymbol{B}(\boldsymbol{x})^t \boldsymbol{B}(\boldsymbol{x}), pp^t \rangle$ we then get $\langle \boldsymbol{B}(\boldsymbol{x})^t \boldsymbol{B}(\boldsymbol{x}), pp^t \rangle \leq |\boldsymbol{B}(\boldsymbol{x})^t \boldsymbol{B}(\boldsymbol{x})|_F |pp^t|_F$ using the Cauchy-Schwartz inequality. Since $|pp^t|_F = \|p\|^2$ and splines are bounded on Ω, $|\boldsymbol{B}(\boldsymbol{x})^t \boldsymbol{B}(\boldsymbol{x}))|_F |\Omega| < \infty$ given $|\Omega| < \infty$ is finite we get, $\int_{\Omega} |\boldsymbol{B}(\boldsymbol{x})^t \boldsymbol{B}(\boldsymbol{x}))| dx \leq \sup_{\boldsymbol{x}\in\Omega} |\boldsymbol{B}(\boldsymbol{x})^t \boldsymbol{B}(\boldsymbol{x})|_F |\Omega|$. Setting $\lambda := \sup_{\boldsymbol{x}\in\Omega} |\boldsymbol{B}(\boldsymbol{x})^t \boldsymbol{B}(\boldsymbol{x})|_F < \infty$, we get $\|v(\boldsymbol{x})\|^2 \leq \lambda \|p\|^2$. In this approximation (11), the Trapezoidal method gives,

$$\boldsymbol{x}_t = \boldsymbol{\phi}(\boldsymbol{x}_{t-1}, \boldsymbol{p}) = \boldsymbol{x}_{t-1} + \frac{\Delta t}{2} \left(\boldsymbol{B}(\boldsymbol{x}_{t-1}) + \boldsymbol{B}(\boldsymbol{x}_t) \right) \boldsymbol{p}. \tag{12}$$

which is solved using fixed point iterations. The inverse is found by rearranging the above,

$$\begin{aligned}\boldsymbol{x}_{t-1} &= \boldsymbol{x}_t - \frac{\Delta t}{2} \left(\boldsymbol{B}(\boldsymbol{x}_{t-1}) + \boldsymbol{B}(\boldsymbol{x}_t) \right) \boldsymbol{p} && (13)\\ &= \boldsymbol{x}_t + \frac{\Delta t}{2} \left(\boldsymbol{B}(\boldsymbol{x}_t) + \boldsymbol{B}(\boldsymbol{x}_{t-1}) \right) (-\boldsymbol{p}) && (14)\\ &= \boldsymbol{\phi}^{-1}(\boldsymbol{x}_t, \boldsymbol{p}), && (15)\end{aligned}$$

and we find that $\boldsymbol{\phi}$ fulfills (1) by direct substitution, thus $\boldsymbol{x}_0 = \boldsymbol{\phi}^{-1}(\boldsymbol{\phi}(\boldsymbol{x}_0, \boldsymbol{p}), \boldsymbol{p})$.

In out approximation, $\boldsymbol{\phi}$ and hence $\mathcal{F}_{\text{sym}}$ depends on $\boldsymbol{p}$, thus the Jacobian to be used in the minimisation process is found as,

$$\frac{\partial \mathcal{F}_{\text{sym}}}{\partial \boldsymbol{p}} = \frac{\partial \mathcal{F}_{\text{sym}}}{\partial \boldsymbol{\phi}} \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{p}}. \tag{16}$$

The Jacobian of (12) is found by vectorizing,

$$\begin{aligned} \boldsymbol{x}_t &= \mathrm{vec}(\boldsymbol{x}_t) \\ &= \mathrm{vec}(\boldsymbol{x}_{t-1}) + \frac{\Delta t}{2}\mathrm{vec}((\boldsymbol{B}(\boldsymbol{x}_t) + \boldsymbol{B}(\boldsymbol{x}_{t-1}))\,\boldsymbol{p}) \qquad (17) \\ &= \boldsymbol{x}_{t-1} + \frac{\Delta t}{2}\left(\boldsymbol{p}^T \otimes \mathbf{Id}_N\right)\left(\mathrm{vec}(\boldsymbol{B}(\boldsymbol{x}_t)) + \mathrm{vec}(\boldsymbol{B}(\boldsymbol{x}_{t-1}))\right), \qquad (18) \end{aligned}$$

where $\mathbf{Id}_N$ is an identity matrix of size $N \times N$; N is the dimensionality of the image domain; $\mathrm{vec}(\cdot)$ is the vec-operator, which for any matrix $\boldsymbol{A}$ with columns $\boldsymbol{A}_{*i}$, $\mathrm{vec}(\boldsymbol{A}) = [\boldsymbol{A}_{*1}^T\, \boldsymbol{A}_{*2}^T \ldots]^T$; and where $\cdot \otimes \cdot$ is the Kronecker product, i.e., for any pairs of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\boldsymbol{A} \otimes \boldsymbol{B} = [a_{11}\boldsymbol{B}\; a_{12}\boldsymbol{B}\; \ldots]$. By direct computation we find the partial derivative w.r.t. $\boldsymbol{p}$ as,

$$\begin{aligned} \frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{p}} = \frac{\partial \boldsymbol{x}_{t-1}}{\partial \boldsymbol{p}} &+ \frac{\Delta t}{2}\left(\boldsymbol{p}^T \otimes \mathbf{Id}_N\right) \cdot \\ &\left(\frac{\partial \mathrm{vec}(\boldsymbol{B}(\boldsymbol{x}_t))}{\partial \boldsymbol{x}_t}\frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{p}} + \frac{\partial \mathrm{vec}(\boldsymbol{B}(\boldsymbol{x}_{t-1}))}{\partial \boldsymbol{x}_{t-1}}\frac{\partial \boldsymbol{x}_{t-1}}{\partial \boldsymbol{p}}\right) \\ &+ \frac{\Delta t}{2}\left(\boldsymbol{B}(\boldsymbol{x}_t) + \boldsymbol{B}(\boldsymbol{x}_{t-1})\right). \qquad (19) \end{aligned}$$

This equation may be rearranged on matrix form, as,

$$\left(\mathbf{Id}_N - \boldsymbol{Q}_t\right)\boldsymbol{X}_t = \left(\mathbf{Id}_N + \boldsymbol{Q}_{t-1}\right)\boldsymbol{X}_{t-1} + \boldsymbol{B}_t + \boldsymbol{B}_{t-1}, \qquad (20)$$

where $\boldsymbol{X}_s = \frac{\partial \boldsymbol{x}_s}{\partial \boldsymbol{p}}$ is an $N \times K$ matrix, $\boldsymbol{Q}_s = \frac{\Delta t}{2}\left(\boldsymbol{p}^T \otimes \mathbf{Id}_N\right)\frac{\partial \mathrm{vec}(\boldsymbol{B}(\boldsymbol{x}_s))}{\partial \boldsymbol{x}_s}$ is an $N \times N$ matrix, and $\boldsymbol{B}_s = \frac{\Delta t}{2}\boldsymbol{B}(\boldsymbol{x}_s)$ is an $N \times K$ matrix. Since the derivatives are evaluated after solving for $\boldsymbol{x}_t$ then both $\boldsymbol{B}_t$ and $\boldsymbol{B}_{t-1}$ are known. For a 3-dimensional image, $N = 3$, and K is typically proportional to the number of voxels in the image. However, for a spline basis, each evaluation point $\boldsymbol{x}$ is only influenced by a small number of parameters $k \ll K$, effectively reducing the dimensionality of $\boldsymbol{X}_s$ to $3 \times k$ for each evaluation point. We solve these systems for $\boldsymbol{X}_t$ using Cramer's rule, as the problem is well conditioned and the method is simple, exact and constant in time.

For comparison with other schemes in Section 5, we will now derive the computational complexity of solving for $\frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{p}}$ using (20). The matrix $\boldsymbol{Q}_t$ needs to be calculated once per iteration and has complexity $\mathcal{O}(N^3K)$ assuming that the Kronecker product and the calculation of the Jacobian of the $\boldsymbol{B}$ are negligible. The product of has complexity $\left(\mathbf{Id}_N + \boldsymbol{Q}_{t-1}\right)\boldsymbol{X}_{t-1}$ is $\mathcal{O}(N^2K)$, and the two additions of $\boldsymbol{B}$ has complexity $\mathcal{O}(NK)$. Solving using Cramer's rule has complexity $\mathcal{O}(N!\,N)$. Thus, the complexity of solving for $\frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{p}}$ using (20) is $\mathcal{O}(N^3K + N!\,N)$. Cramer's rule becomes inefficient for values of $N > 3$ in which case it may be replaced by, e.g., Gauss-Jordan elimination with complexity $\mathcal{O}(N^3)$, and which will be dominated by $\mathcal{O}(N^3K)$.

Constructing a registration scheme as (4), the algorithm for each direction can be formulated as Algorithm 1. For computational efficiency and memory compactness we use a slicing technique, and for simplicity we assume that x_0 are voxel positions. We compute $\boldsymbol{x}_t$ from (12) by fixed point iterations, initially setting $\boldsymbol{x}_t = \boldsymbol{x}_{-1}$ and use (12) recursively until convergence (4-5 recursions). (12) is guaranteed to converge if $\boldsymbol{B}(\boldsymbol{x})\boldsymbol{p}$ is Lipshitz continuous and Δt sufficiently small [51].

```
Data: I, J, p
Result: F_sym and ∂F_sym/∂p
for each direction do
    for each evaluation point i do
        for each time step Δt do
            Compute x_t^i using (12) recursively ;
        end
    end
    Given x_{t=1}^i, compute F_sym(I, J, φ) and store
      derivative ;
    for each evaluation point i do
        for each time step Δt do
            Compute x_t^i (12) recursively;
            use x_t^i to compute X_t^i of (20) (a 3 × k
              system) using Cramer's rule;
        end
        multiply X_t^i with derivative of F_sym;
    end
end
```

Algorithm 1: A slicing approach to the implementation of the trapezoidal method including derivatives. Empirical results show that the cost of solving for x_t^i is roughly 20% of the outer loop.

4 Experiments and Results

We performed a series of non-rigid registrations to illustrate the performance of **CDD** using the 4 publicly available data sets of T1 weighted brain images used in [35]. The following sections describes the data and the experiments along with our results.

4.1 Data

We used the same images as in [35] which were preprocessed according to the methods described therein, with a few minor changes as detailed below.

LPBA The dataset contains 40 brain images with 56 labels[2] each, used to construct the LONI Probabilistic Brain Atlas (LPBA40) [14], and is available online[3]. Note although errors were found by [35], we use the same dataset as in that paper to give a fair comparison.

IBSR The dataset contains 18 brain images with 84 labeled regions, available in the IBSR v2.0[4]. The images have been aligned to Talairach [52] and processed by the Center for Morphometric Analysis, Massachusetts General Hospital (MGH).

CUMC12 The dataset consist of 12 subjects with 128 labeled regions that were scanned at the Columbia University Medical Center and manually labeled by one technician trained according to the Cardviews labeling scheme [16].

MGH10 The dataset consists of 10 subjects with 106 labeled regions scanned at the A. Martinos Center for Biomedical Imaging. The data has been corrected for inhomogeneity and affine registered to the MNI152 template [15].

Comparative Registration Evaluations For the comparison we used the scores of SyN, SPM-DARTEL, and Diffeomorphic Demons provided as supplementary material for [35]

2. http://www.loni.ucla.edu/Protocols/LPBA40
3. http://www.loni.ucla.edu/Atlases/LPBA40
4. http://www.cma.mgh.harvard.edu/ibsr/

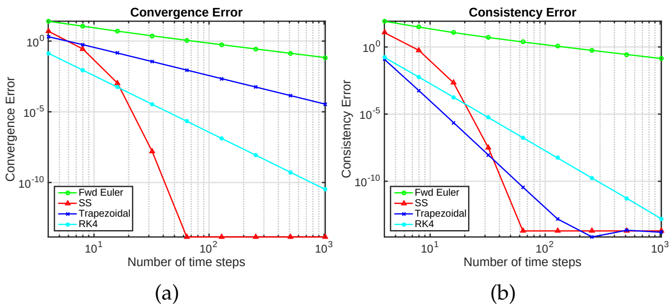


Fig. 2. The convergence plot (a) and the consistency plot (b) of integrating a circle with the different methods. Their similarity demonstrate that the consistency is a good surrogate for convergence, which is useful when the true solution of the ODE is unknown.

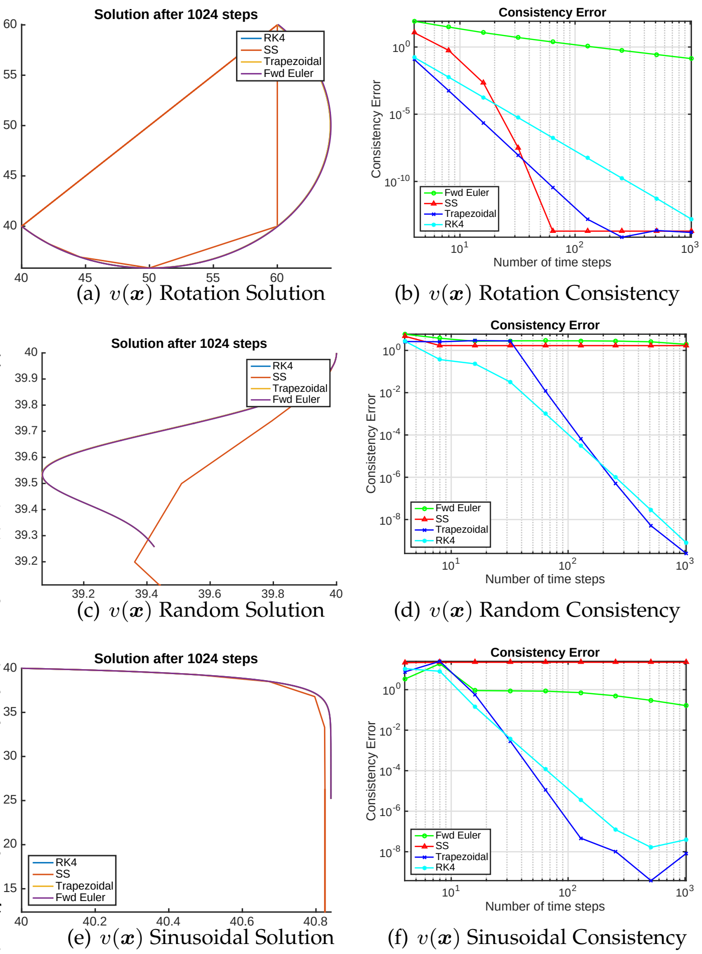


Fig. 3. The solutions for a single point obtained by the use of Fwd Euler, Trapezoidal, SS and RK4 methods on 3 different velocity fields (top row) and the corresponding consistency plots (bottom row). a) & b) are for a full rotation of π with radius 10. c) & d) is a random velocity field with mean 0 and std 10, and e) & f) a sinusoidal function. The plots clearly illustrate that for very well behaved velocity fields (rotation (b)) the SS is far superior, however for velocity fields which are slightly more challenging then SS fails to converge ((d) and (f)).

where the data is detailed further. As IRTK, ANIMAL, ART, AIR, JRD Fluid, SICLE and FNIRT are non-diffeomorphic registration methods, these where excluded.

In addition we included a data sample with pathologies to illustrate the method's performance on non-standard data: A small Alzheimer's Disease (AD) dataset, which consists of 8 subjects from the ADNI database labelled using the harmonized hippocampus protocol [53]. All scans were intensity-normalized using N3 [54].

4.1.1 *Preprocessing and Changes*

All non-brain regions of the images were removed before registration using the corresponding brain mask, as in [35]. Due to an error in the segmentation mask and atlas of subject 'm2' from the CUMC12 dataset, this subject was excluded.

4.2 Artificial 2D examples

To illustrate the basic behaviour of the Forward Euler, Trapezoidal, SS and RK4 methods we computed the exponential of 3 artificial 2D velocity fields. 1) a pure rotation of π with radius 10 and with a known solution, 2) a random velocity field with average magnitude of 0 and std of 10 and 3) the velocity field $v_x(x, y) = 10sin(x), v_y(x, y) = 10cos(x)$. The velocity fields (1) and (3) were defined analytically and the random vector field was sampled on a 1x1 isotropic grid. SS used a 1x1 representation for the squaring operation, as in [10]. Figure 2 shows the convergence and consistency of the rotational velocity, and due to their similar nature it is fair to assume that consistency is a good surrogate for convergence when the true solution of the ODE is unknown. The results in Figure 3 from the rotational velocity field correspond well with the theory, with linear convergence for Forward Euler, Trapezoidal and RK4, and with quadratic convergence for SS. The results from the random and sinusoidal velocity fields shows a consistency similar to that of the rotation for the Forward Euler, Trapezoidal and RK4 methods, whereas the SS appears to not converge, given the poor consistency, and produces inferior solutions (Figure 3 (b) and (c)).

4.3 Consistency

Consistency is the property that differentiates **CDD** from other methods. To illustrate this difference we compared the convergence of a number of numerical schemes that approximates (5) by measuring the consistency (1). The statistics of the consistency were evaluated over the 45 velocity fields generated by the registration of the MGH10 dataset, over all voxel positions x_0, and as a function of the number of time steps. We compared the Trapezoidal method used by **CDD**, Scaling and Squaring (SS) used by SPM-DARTEL and Diffeomorphic Demons, the standard explicit methods Forward Euler, and RK4. For SS, the velocity field of $5 \times 5 \times 5$ mm^3 was re-sampled to 1 mm in order to compute the exponential at voxel resolution. Note that SyN is not mentioned, as it is not an SVF but an LDDMM method. The the maximum error and mean consistency errors are shown in Figure 4.

Figure 4 show that the Trapezoidal method improved the mean and maximum consistency by 5-8 orders of magnitude compared to the other SVF integration techniques and 9 orders of magnitude compared to state-of-the-art SVF frameworks [4], [6], SPM-DARTEL and Diffeomorphic Demons, including SyN [7], [3]. In addition, the Trapezoidal method

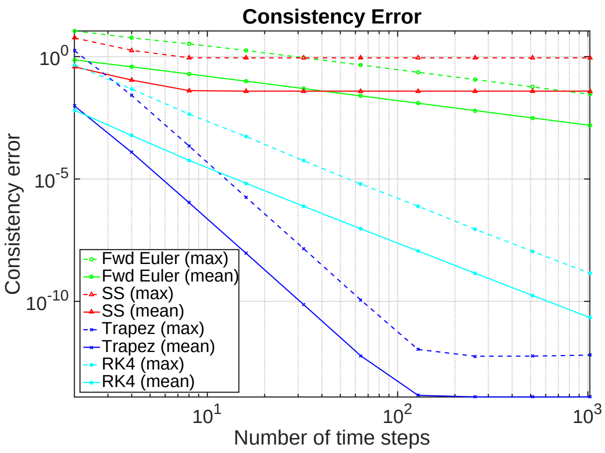


Fig. 4. The reduction in the maximum and mean consistency accuracy error as a function of the number of steps using $1st$-order B-Splines using the SVF's obtained from the 45 pairwise registrations of the MGH10 dataset. As the figure shows, the consistency accuracy for the Euler improves by a factor of 2 when doubling the number of time steps, the RK4 with a factor of 10 and the trapezoidal is around a factor of 50.

produced the fastest reduction in the consistency error in terms of time steps and was fully converged to machine precision after 128 time steps. See Section 5 for a comparison of the computational complexity of each time step for each method. Note how well the convergence in consistency in Figure 3 and Figure 4 corresponds. Extrapolating the results of Figure 4, the RK4 would converge after roughly 4000 time steps, whereas the Euler scheme consistently reduced the error by a factor of 2 indicating that the number of time steps should have been increased with a factor of $10^6 - 10^7$ in order to achieve an accuracy equal to that of the Trapezoidal method attained after just 16 time steps. The SS converged to its maximum precision after only 8 squaring operations with an average consistency of 0.039 mm and a max of 0.888 mm.

Figure 4 demonstrates that **CDD** preserves the consistency property of diffeomorphic mappings up to machine precision, and is between 5 and 9 orders of magnitude more consistent than the SS, Forward Euler and RK4. The RK4 is expensive to compute and is, from a theoretical point of view, less stable than the Trapezoidal method, and as the results clearly show, it offers significantly lower consistency. The extrapolation of the results show that the Forward Euler scheme requires 10^7 time steps, and RK4 requires 4000 time steps to achieve the same consistency as **CDD**. Popular state-of-the-art methods such as SPM-DARTEL, Diffeomorphic demons and LDDMM of SyN only have an accuracy on the order of a single voxel, although for SyN this is user specified (with 0.5 mm as the default because high accuracy is computationally expensive). Assuming symmetry and consistency, we applied SyN to the MGH10 dataset with the setting from [35] and found an inconsistency as high as 2.1 voxels and a mean of 0.01 voxels across the dataset. The popular SS algorithm has been shown to diverge [4], [5], and in our experiments delivers the most inferior consistency of the four methods tested. While the registration results produced by SS may appear to be good (e.g., SPM-DARTEL in Figure 8), our application of SS to MGH10 shows that the deformations produced are far from consistent and clearly violate this basic property of diffeomorphisms. Although it would be interesting to investigate SS further, a full analysis is beyond the scope of this work.

4.4 Registration of Benchmark data

The trivial solution to accurate consistency in image registration is the identity transformation. To illustrate that **CDD** achieves state-of-the-art registrations while preserving consistency, we repeated the registration experiments performed in [35] on the LPBA, CUMC12, IBSR and MGH10 datasets. Each non-rigid registration was initialized with a symmetric affine registration using NMI. The registration framework using the trapezoidal method was realized as a continuation method using 4 scales of 1^{st} order uniform B-Spline. The algorithm uses 40 mm, 20 mm, 10 mm and 5 mm knot spacing respectively as the representation of velocity field $\boldsymbol{v}(\boldsymbol{x})$, where the finest level has a grid of size $45 \times 45 \times 45$ knots, i.e., $K = 3 \cdot 45^3$. Using 40 time steps, the numerical integration of (4) was performed through integration points, distributed on a uniform grid with 2 mm spacing in each image within the brain-masks. For regularization we used $\lambda = 0.0005$ in (2), which was kept constant across all scales. The framework used the LOR [9], [55] formulation for NMI as image similarity. Consistent comparison with respect to image scale was ensured by accounting for the affine mapping, enabling the use of the same image interpolation kernel directly onto both images. At each level the registration problem was solved and the solution was projected onto the next and finer grid by simple knot splitting. This ensured that each velocity knot in the coarse grid existed in the corresponding high resolution velocity grid. The configuration and choice of parameters was not optimized but chosen based on experience. An example of a registration result can be found in Figure 5. We computed the target overlap and mean overlap for all pairs of images directly from the corresponding atlases as in [35]. The results are presented in Figure 6, 7, 8 and 9 and are complemented by the publicly available results from [35].

Figure 6, 7, 8 and 9 show that the new **CDD** method achieves state-of-the-art registration results with significantly higher mean of mean overlap and target overlap across all data sets, significantly higher median of the mean overlap on IBSR, CUMC12 and MGH10 and significantly higher median of the target overlap on the LPBA, CUMC12 and MGH10 datasets. We compared the proposed method to each of the other methods using double-sided paired t-tests for mean and double sided paired Wilcoxon signed rank tests for zero median. The full set of test results for each dataset can be found in following tables: LPBA Table 1 , IBSR Table 2, CUMC12 Table 3, and MGH10 Table 4. In summary, only one case returned a score significantly higher than **CDD**, namely the median of SyN on the mean overlap on the LPBA dataset. In all other cases **CDD** produced significantly more accurate scores according to the tests performed, except for one which was insignificantly lower than the proposed method: SPM-DARTEL's median of the Target Overlap on IBSR with a p value of 0.09.

The results demonstrated that **CDD** performs significantly better than existing state-of-the-art methods on the benchmark data. In contrast to the results reported in [35]

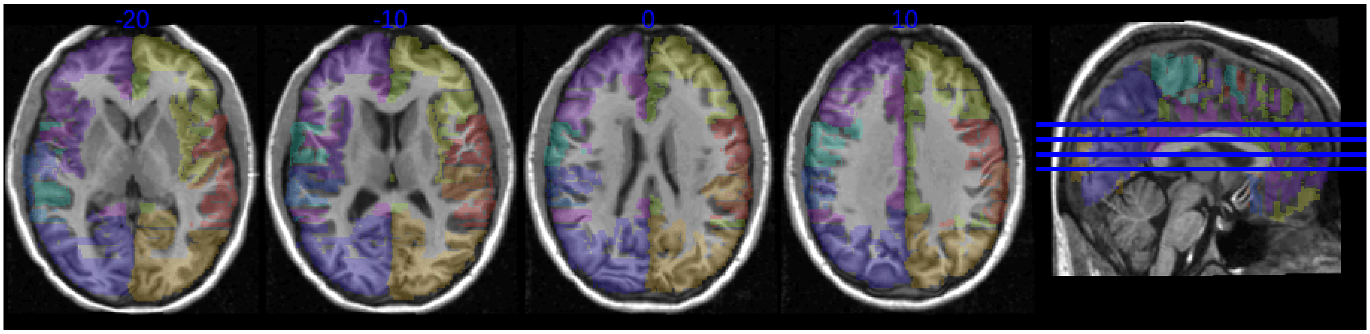


Fig. 5. An example of the results of a registration from the MGH dataset of a deformed brain with labels overlaid, showing that the labels match well with the cortical structures. Four different transversal slices are shown, together with a sagittal overview slice that marks the position of the transversal slices with blue lines, starting with the lowest slice displayed on the far left.

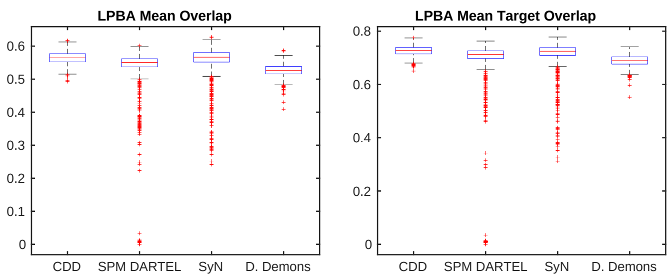


Fig. 6. The box plot of the average mean overlap and average target overlap for LPBA dataset. $\mathbf{CDD}$ has the highest mean of both measures and the highest median of the target overlap. SyN has a median of the mean overlap which is 0.0019 higher than $\mathbf{CDD}$, but has far more outliers and thus produces far less reliable registrations.

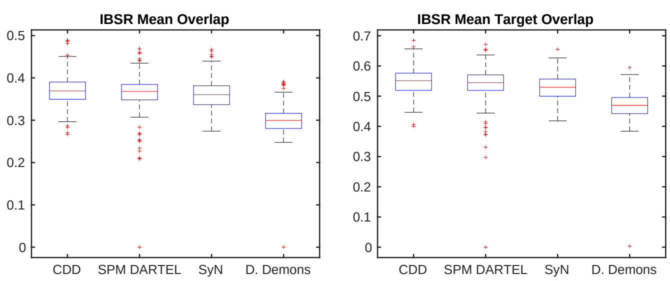


Fig. 7. The box plot of the average mean overlap and average target overlap for the IBSR. $\mathbf{CDD}$ has the highest mean and median of both measures. DARTEL has a median of target overlap which is 0.0064 lower than $\mathbf{CDD}$, but has more outliers and thus produces less reliable registrations.

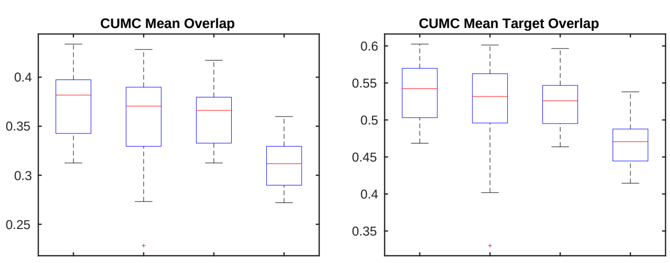


Fig. 8. The box plot of the average mean overlap and average target overlap for the CUMC12. $\mathbf{CDD}$ has the highest mean and median of both measures.

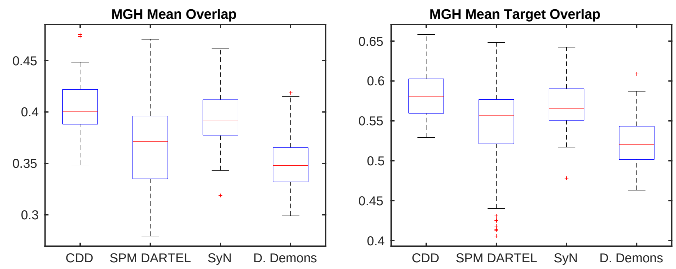


Fig. 9. The box plot of the average mean overlap and average target overlap for the MGH10. $\mathbf{CDD}$ has the highest mean and median of both measures.

Framework	Mean		Median	
	MO	TO	MO	TO
CDD	**0.5643**	**0.7261**	0.5647	**0.7278**
SPM-DARTEL	0.5172 ($p = 0$)	0.6715 ($p = 0$)	0.5508 ($p = 0$)	0.7130 ($p = 0$)
SyN	0.5576 ($p << 0.0001$)	0.7146 ($p = 0$)	**0.5665** ($p < 0.0001$)	0.7148 ($p << 0.0001$)
D. Demons	0.5259 ($p = 0$)	0.6893 ($p = 0$)	0.5263 ($p = 0$)	0.6895 ($p = 0$)

TABLE 1
Individual Tests vs $\mathbf{CDD}$ on LPBA: The mean and median of the Target Overlap and mean overlap of the registration. All $\mathbf{CDD}$ scores are significantly better than existing state-of-the-art methods, with the exception of the median of the mean overlap where SyN is significantly higher by 0.019.

Framework	Mean		Median	
	MO	TO	MO	TO
CDD	**0.3703**	**0.5468**	**0.3693**	**0.5513**
SPM-DARTEL	0.3635 ($p < 0.0009$)	0.5403 ($p < 0.015$)	0.3680 ($p < 0.0022$)	**0.5447** ($p < 0.0962$)
SyN	0.3600 ($p = 0$)	0.5281 ($p = 0$)	0.3603 ($p = 0$)	0.5298 ($p = 0$)
D. Demons	0.3012 $p = 0$	0.4682 $p = 0$	0.2996 $p = 0$	0.4698 $p = 0$

TABLE 2
Individual Tests vs $\mathbf{CDD}$ on IBSR: The mean and median of the Target Overlap and mean overlap of the registration of the IBSR. All $\mathbf{CDD}$ scores are better than existing state-of-the-art methods. The medians, although higher, are insignificantly different from that of SPM-DARTEL, whereas the mean is significantly higher. This is an expression of the fact that $\mathbf{CDD}$ has far fewer outliers than SPM-DARTEL (the mean value is sensitive to a large number of outliers).

Framework	Mean		Median	
	MO	TO	MO	TO
CDD	**0.3727**	**0.5389**	**0.3818**	**0.5423**
SPM-DARTEL	0.3616 ($p < 1e-5$)	0.5278 ($p < 0.001$)	0.3706 ($p = 0$)	0.5317 ($p < 1e-5$)
SyN	0.3592 ($p = 0$)	0.5225 ($p = 0$)	0.3661 ($p = 0$)	0.5258 ($p = 0$)
D. Demons	0.3109 ($p = 0$)	0.4692 ($p = 0$)	0.3118 ($p = 0$)	0.4706 ($p = 0$)

TABLE 3
Individual Tests vs $\mathbf{CDD}$ on CUMC: The mean and median of the Target Overlap and mean overlap of the registration of the CUMC dataset. All $\mathbf{CDD}$ scores are significantly better than the existing state-of-the-art methods.

Framework	Mean		Median	
	MO	TO	MO	TO
CDD	**0.4048**	**0.5814**	**0.4007**	**0.5802**
SPM-DARTEL	0.3654 ($p = 0$)	0.5431 ($p = 0$)	0.3714 ($p = 0$)	0.5565 ($p = 0$)
SyN	0.3946 ($p = 0$)	0.5683 ($p = 0$)	0.3913 ($p = 0$)	0.5652 ($p = 0$)
D. Demons	0.3497 ($p = 0$)	0.5228 ($p = 0$)	0.3479 ($p = 0$)	0.5201 ($p = 0$)

TABLE 4
Individual Tests vs $\mathbf{CDD}$ on MGH: The mean and median of the Target Overlap and mean overlap of the registration of the MGH dataset. All $\mathbf{CDD}$ scores are significantly better than the existing state-of-the-art methods.

where each dataset had its own best performing method (SyN on the MGH10, SPM-DARTEL on the CUMC12 and IBSR, and ART/SyN on the LPBA), the proposed method delivers state-of-the-art performance across all 4 datasets. The statistical tests show that $\mathbf{CDD}$ scores significantly higher compared the current state-of-the-art, according to the mean overlap and target overlap metrics on the manually delineated regions from the four datasets. The only instance where SyN performs better is the median of the mean overlap on the LPBA. This indicates that SyN could potentially perform better, provided that the number of outliers are reduced. However, SyN produces poor registration for approximately 4% of the registrations on LPBA. Such a high number of inferior registrations would require substantial image information to be excluded and reduce the power of the study or introduce noise. In contrast, $\mathbf{CDD}$ performs well in all situations.

We performed a full non-rigid registration of the CUMC12 dataset without the use of symmetry. The resulting mean-overlap and average target overlap was roughly 0.005 (approx. 1.4%) lower across all pairs of subjects in the dataset.

We performed a non-rigid registration using $\mathbf{CDD}$ and SyN of 8 random AD subjects from ADNI where manual labels of the hippocampus were available. The regularization was halved and the resolution increased to 32 mm, 16 mm, 8 mm and 4 mm respectively to account for atrophy. To serve as reference, we also applied SyN on the AD group using default settings with Correlation Coefficient. The computed mean overlap was 0.655 for both left and right hippocampus for $\mathbf{CDD}$ which is insignificantly different from 0.65 and 0.66 for left and right hippocampus respectively obtained by SyN. Thus SyN and $\mathbf{CDD}$ perform equally well.

5 Computational Complexity

We have found that implicit methods are not too computationally expensive for any practical use in image registration. To substantiate this, we will now present the computational complexity of the Trapezoidal method to the Forward Euler, RK4 and Scaling and Squaring. The Forward Euler, RK4 and Trapezoidal methods have the same order of computational complexity from a theoretical point of view. However, there are inherent methodological differences, highlighted by a computation of the derivative of the initial value problem for the three alternative integration methods. To make a proper analysis of the computational resources required, the computational cost of the Jacobians has to be considered.

In the following we find the Jacobian of the mapping and derive equations for the derivative of the flow $\boldsymbol{\phi}$ for Euler, 4th order Runge-Kutta schemes and Scaling and Squaring. These methods are all implementations of the following: Consider points $\boldsymbol{x}_t \in \mathbb{R}^N$ and the motion of these points in a stationary velocity field according to (5), where t is integration time. In the following we will decompose this motion into n equidistant steps, hence,

$$\boldsymbol{\phi}_1(\boldsymbol{x}_0) = \boldsymbol{\phi}_{1/n} \circ \cdots \circ \boldsymbol{\phi}_{1/n} \circ \boldsymbol{\phi}_{1/n}(\boldsymbol{x}_0) \tag{21}$$

We also approximate the spatial velocity field at each time point as (11). Note that in (21), when the variable $\boldsymbol{x}$ is discretely sampled on a grid, the function $\boldsymbol{\phi}_{1/n}$ needs only to be evaluated once on every grid point, such that off-grid values may can be estimated by interpolation.

5.1 Jacobian of Forward Euler Integration

Euler integration is achieved by

$$\boldsymbol{\phi}_{1/n}(\boldsymbol{x}) = \boldsymbol{x} + \frac{1}{n}\boldsymbol{B}(\boldsymbol{x})\boldsymbol{p} + \mathcal{O}\left(n^{-2}\right), \tag{22}$$

where $\mathcal{O}(n^{-2})$ is the error in each step, and the total error is $\mathcal{O}(n^{-1})$. Thus, the error of approximation by Euler integration can be made arbitrarily small with a sufficiently large number of steps.

In the following we write $\Delta t = \frac{1}{n}$ and assume that the error of approximation (the Euler step) is negligible. Thus,

$$\boldsymbol{x}_t = \boldsymbol{\phi}_{\Delta t}(\boldsymbol{x}_{t-1}) = \boldsymbol{x}_{t-1} + \boldsymbol{B}(\boldsymbol{x}_{t-1})\boldsymbol{p}\Delta t. \tag{23}$$

The Jacobian of $\boldsymbol{x}_t$ w.r.t. $\boldsymbol{p}$ is,

$$\begin{aligned}\frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{p}} =& \frac{\partial \boldsymbol{x}_{t-1}}{\partial \boldsymbol{p}} \\ &+ \left(\boldsymbol{p}^T \otimes \mathbf{Id}_N\right) \left.\frac{\partial \mathrm{vec}(\boldsymbol{B}(\boldsymbol{y}))}{\partial \boldsymbol{y}}\right|_{\boldsymbol{y}=\boldsymbol{x}_{t-1}} \frac{\partial \boldsymbol{x}_{t-1}}{\partial \boldsymbol{p}}\Delta t \\ &+ \boldsymbol{B}(\boldsymbol{x}_{t-1})\Delta t.\end{aligned} \tag{24}$$

Converting to matrix form, this equations becomes,

$$\boldsymbol{X}_t = \left(\mathbf{Id}_N + \boldsymbol{Q}_{t-1}\right)\boldsymbol{X}_{t-1} + \boldsymbol{B}_{t-1}, \tag{25}$$

where $\boldsymbol{X}_s = \frac{\partial \boldsymbol{x}_s}{\partial \boldsymbol{p}}$ is an $N \times K$ matrix, $\boldsymbol{Q}_s = \Delta t\left(\boldsymbol{p}^T \otimes \mathbf{Id}_N\right) \left.\frac{\partial \mathrm{vec}(\boldsymbol{B}(\boldsymbol{y}))}{\partial \boldsymbol{y}}\right|_{\boldsymbol{y}=\boldsymbol{x}_{t-1}}$ is an $N \times N$ matrix, and $\boldsymbol{B}_s = \Delta t \boldsymbol{B}(\boldsymbol{x}_s)$ is an $N \times K$ matrix. We see that (25) is similar to (20) except that no system of equations needs to be solved, and by similar arguments we find that the computational complexity is $\mathcal{O}(N^3K)$.

5.2 Jacobian of 4th order Runge-Kutta Integration

4th order Runge-Kutta integration is achieved by

$$\boldsymbol{\phi}_{1/n}(\boldsymbol{x}) = \boldsymbol{x} + \frac{1}{6n}\sum_{j=1}^{4}\boldsymbol{k}_j(\boldsymbol{x}) + \mathcal{O}\left(n^{-5}\right), \tag{26}$$

$$\boldsymbol{k}_1(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{p}, \tag{27}$$

$$\boldsymbol{k}_2(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x} + \frac{\Delta t}{2}\boldsymbol{k}_1(\boldsymbol{x}))\boldsymbol{p}, \tag{28}$$

$$\boldsymbol{k}_3(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x} + \frac{\Delta t}{2}\boldsymbol{k}_2(\boldsymbol{x}))\boldsymbol{p}, \tag{29}$$

$$\boldsymbol{k}_4(\boldsymbol{x}) = \boldsymbol{B}(\boldsymbol{x} + \Delta t\boldsymbol{k}_3(\boldsymbol{x}))\boldsymbol{p}, \tag{30}$$

where $\mathcal{O}\left(n^{-5}\right)$ is the error in each step, and the total error is $\mathcal{O}\left(n^{-4}\right)$. Thus, the error of approximation by 4th order Runge-Kutta integration can be made arbitrarily small with sufficiently large number of steps, and the convergence is faster than Euler integration.

In the following we write $\Delta t = \frac{1}{n}$ and assume that the error of approximation is negligible. Thus,

$$\boldsymbol{x}_t = \boldsymbol{\phi}_{\Delta t}(\boldsymbol{x}_{t-1}) = \boldsymbol{x}_{t-1} + \frac{\Delta t}{6}\sum_{j=1}^{4}\boldsymbol{k}_j(\boldsymbol{x}_{t-1}). \tag{31}$$

The Jacobian of $\boldsymbol{x}_i$ w.r.t. $\boldsymbol{p}$ is

$$\frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{p}} = \frac{\partial \boldsymbol{x}_{t-1}}{\partial \boldsymbol{p}} + \frac{\Delta t}{6}\left(\sum_{j=1}^{4}\frac{\partial \boldsymbol{k}_j(\boldsymbol{x}_{t-1})}{\partial \boldsymbol{p}}\right). \tag{32}$$

To evaluate the Jacobians of $\boldsymbol{k}_j$ w.r.t. $\boldsymbol{p}$ we use the shorthand notation $\boldsymbol{P} = \left(\boldsymbol{p}^T \otimes \mathbf{Id}_N\right)$. Starting with $\boldsymbol{k}_1$, and using the short-hand $\boldsymbol{k}_j = \boldsymbol{k}_j(\boldsymbol{x}_{t-1})$, we find the recursive formulas to be,

$$\frac{\partial \boldsymbol{k}_1}{\partial \boldsymbol{p}} = \boldsymbol{P}\left.\frac{\partial \mathrm{vec}(\boldsymbol{B}(\boldsymbol{x}))}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{t-1}}\frac{\partial \boldsymbol{x}_{t-1}}{\partial \boldsymbol{p}} + \boldsymbol{B}\left(\boldsymbol{x}_{t-1}\right), \tag{33}$$

$$\begin{aligned}\frac{\partial \boldsymbol{k}_2}{\partial \boldsymbol{p}} &= \\ &\boldsymbol{P}\left.\frac{\partial \mathrm{vec}(\boldsymbol{B}(\boldsymbol{x}))}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{t-1}+\frac{\Delta t}{2}\boldsymbol{k}_1}\left(\frac{\partial \boldsymbol{x}_{t-1}}{\partial \boldsymbol{p}} + \frac{\Delta t}{2}\frac{\partial \boldsymbol{k}_1}{\partial \boldsymbol{p}}\right) \\ &+ \boldsymbol{B}\left(\boldsymbol{x}_{t-1} + \frac{\Delta t}{2}\boldsymbol{k}_1\right),\end{aligned} \tag{34}$$

$$\begin{aligned}\frac{\partial \boldsymbol{k}_3}{\partial \boldsymbol{p}} &= \\ &\boldsymbol{P}\left.\frac{\partial \mathrm{vec}(\boldsymbol{B}(\boldsymbol{x}))}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{t-1}+\frac{\Delta t}{2}\boldsymbol{k}_2}\left(\frac{\partial \boldsymbol{x}_{t-1}}{\partial \boldsymbol{p}} + \frac{\Delta t}{2}\frac{\partial \boldsymbol{k}_2}{\partial \boldsymbol{p}}\right) \\ &+ \boldsymbol{B}\left(\boldsymbol{x}_{t-1} + \frac{\Delta t}{2}\boldsymbol{k}_2\right),\end{aligned} \tag{35}$$

$$\begin{aligned}\frac{\partial \boldsymbol{k}_4}{\partial \boldsymbol{p}} &= \\ &\boldsymbol{P}\left.\frac{\partial \mathrm{vec}(\boldsymbol{B}(\boldsymbol{x}))}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{t-1}+\Delta t\boldsymbol{k}_3}\left(\frac{\partial \boldsymbol{x}_{t-1}}{\partial \boldsymbol{p}} + \Delta t\frac{\partial \boldsymbol{k}_3}{\partial \boldsymbol{p}}\right) \\ &+ \boldsymbol{B}\left(\boldsymbol{x}_{t-1} + \Delta t\boldsymbol{k}_3\right),\end{aligned} \tag{36}$$

For speed, SS estimates the value of the outer application as the interpolation of the 2^N integer corner points of the N-dimensional cube that $\boldsymbol{y} = \boldsymbol{\phi}_{1/j}(\boldsymbol{x})$ belongs to, and which may quickly be found by suitable ceiling and floor operations on the elements of $\boldsymbol{y}$. The Jacobian of (42) w.r.t. $\boldsymbol{p}$ is found to be,

$$\frac{\partial \boldsymbol{\phi}_{2/j}(\boldsymbol{x})}{\partial \boldsymbol{p}} = \left.\frac{\partial \boldsymbol{\phi}_{1/j}(\boldsymbol{y})}{\partial \boldsymbol{p}}\right|_{\boldsymbol{y}=\boldsymbol{\phi}_{1/j}(\boldsymbol{x})} + \left.\frac{\partial \boldsymbol{\phi}_{1/j}(\boldsymbol{y})}{\partial \boldsymbol{y}}\right|_{\boldsymbol{y}=\boldsymbol{\phi}_{1/j}(\boldsymbol{x})} \frac{\partial \boldsymbol{\phi}_{1/j}(\boldsymbol{x})}{\partial \boldsymbol{p}} \tag{43}$$

In matrix form the recursive step becomes

$$\boldsymbol{X}_{2/j,\boldsymbol{x}} = \boldsymbol{X}_{1/j,\boldsymbol{\phi}_{1/j}(\boldsymbol{x})} + \boldsymbol{D}_{1/j,\boldsymbol{\phi}_{1/j}(\boldsymbol{x})}\boldsymbol{X}_{1/j,\boldsymbol{x}} \tag{44}$$

where $\boldsymbol{X}_{i,\boldsymbol{x}} = \left.\frac{\partial \boldsymbol{\phi}_i(\boldsymbol{y})}{\partial \boldsymbol{p}}\right|_{\boldsymbol{y}=\boldsymbol{x}}$ is an $N \times K$ matrix, and $\boldsymbol{D}_{i,\boldsymbol{x}} = \left.\frac{\partial \boldsymbol{\phi}_i(\boldsymbol{y})}{\partial \boldsymbol{y}}\right|_{\boldsymbol{y}=\boldsymbol{x}}$ is an $N \times N$ matrix. The term $\boldsymbol{X}_{1/j,\boldsymbol{x}}$ is given from the previous step, $\boldsymbol{D}_{1/j,\boldsymbol{\phi}_{1/j}(\boldsymbol{x})}$ is found by finite differencing of the corner points of $\boldsymbol{\phi}_{1/j}(\boldsymbol{x})$, and $\boldsymbol{X}_{1/j,\boldsymbol{\phi}_{1/j}(\boldsymbol{x})}$ is found by interpolation. These interpolations are negligible compared to the matrix addition and multiplication, which has computational complexity $\mathcal{O}(NK)$ and $\mathcal{O}(N^2K)$. Thus we find the computational complexity to be $\mathcal{O}(N^2K)$.

5.4 Summarizing and Comparing Complexity

The Forward Euler, RK4 and Trapezoidal method are all from the family of Runge-Kutta methods. For comparison, we primarily used the cost of computing the derivatives, as the cost of integration is negligible compared to the cost of computing the partial derivatives with respect to the initial value. This is summarized in Table 5. Forward Euler and Trapezoidal are approximately equally fast, while RK4 is 4 times slower. SS is by far the fastest, but its convergence is limited by the repeated interpolation [30]. The memory footprint is proportional to the number of matrices stored making the memory footprint of the CDD twice of that of Forward Euler and about half of RK4. In practice SS requires many more sample points to obtain a suitable solution as compared to the other three methods.

5.4.1 Runtime

The **CDD** CPU implementation running in parallel used roughly $10 - 15$ minutes on a high-performance desktop with i7-3940 Intel processor (6 cores) with 64 GB of memory, to perform a single full non-rigid symmetric registration as described in section 4.4. The compact memory usage of **CDD**'s algorithm permits the simultaneous running of hundreds of registrations (single threaded) on clusters with limited memory capacity. The LPBA was registered within 36 hours on a 64 core AMD machine with 256 GB of memory producing 780 symmetric registrations. On the same 64 core machine both MGH10 and CUMC12 required 3 hours, whilst IBSR required 9 hours. Extrapolating the runningtimes of the trapezoidal method using the complexity and the convergence figure while requiring same level of consistency, RK4 would take roughly 100 times the running time of the trapezoidal method (LPBA 150 days, MGH10 and CUMC12 required 12 days, whilst IBSR required 36 days) and Forward Euler would take roughly 10^7 times the running time of the trapezoidal method (LPBA 36000 years, MGH10 and CUMC12 required 3000 years, whilst IBSR required 9000 years). SS is fast but cannot produce diffeomorphic registration to the accuracy required.

6 Discussion

The central contribution of this work is the maintenance of consistency to machine precision permitted by the use of the implicit Trapezoidal method. In this context, the collocation property of the Trapezoidal method is the single most important factor as it ensures that the estimated deformation maintains the diffeomorphic property in the evaluation points, and thus is an accurate discrete representation hereof. That is, the SVF for the deformation is trivially invertible as its negation. The ambiguity of the registration ordering can be removed as the formulation allows us to construct a non-rigid symmetric registration framework using the same set of parameters for the forward and backward transformation, which will also remain symmetric during the entire optimization process. This single parameterization is important for the symmetric registration, when using gradient based optimization, since the forces driving the solution towards the minima are the gradients of the similarity with respect to the parameters including the gradients of the images. By posing a symmetric problem with a single parameterization we are able to take advantage of gradient information from both images through the forward and backward deformation. Here the consistency plays a key-role, as an inconsistency of, say, 0.8 mm would cause the similarity measure to see a set of edges that, whilst perfectly matched through the forward deformation, are still misaligned by 0.8 mm through the backwards deformation. The optimization procedure will then seek a compromise between backward and forward deformations with respect to the similarity measure, thereby causing both directions to be misaligned. This may lead to the belief that the midpoint will remedy this issue. However, the following mapping has been matched during optimization: $\mathcal{M}(\boldsymbol{I} \circ \boldsymbol{\phi}_{\frac{1}{2}}, \boldsymbol{J} \circ \boldsymbol{\psi}_{\frac{1}{2}})$ but the deformation applied is either $\boldsymbol{\phi}$ or $\boldsymbol{\psi}$, and $\|\mathrm{Id} - \boldsymbol{\psi} \circ \boldsymbol{\phi}\| \gg 0$ so the inconsistency is far from being corrected. In fact, one could argue that the deformation used is not even the one estimated, due to the inconsistency.

The importance of using gradients from both images, as stated by [12], is supported by the inferior registration results when symmetry is not enforced. Another indication of symmetry, and thus consistency, being a key-factor in good registration between subjects is the fact that the top-performing methods (SyN and **CDD**) are also the most consistent. SyN has a consistency error of 0.03-0.2 mm [3], [17] (although user-specified) and is only superseded by **CDD**, which offers consistency to machine precision. There are also indications towards a better matching of details. The datasets containing the most labels (CUMC10 and MGH10) are also those with the most significant difference, whereas LPBA, with fewer labels, displays a smaller difference. This could be due to the reduced need for detailed alignment

	Forward Euler	RK4	Trapezoidal	SS
Multiplication/step	N^3K	$4N^3K$	$N^3K + 2N^2K$	N^2K
Inversion/step	0	0	N^3	0
Sum/step	N^3K	$4N^3K$	$(N^3(K+1) + 2N^2K)$	N^2K
Sum/step relative to Euler	1	4	$1 + \frac{1}{K} + \frac{2}{N}$	$\frac{1}{N}$
Estimated steps to converge (Figure 4)	$\frac{1}{2}10^7$	4000	128	8*
Estimated steps relative to Euler	1	$8 \cdot 10^{-4}$	$2.56 \cdot 10^{-5}$	-
Estimated running time relative to Euler	1	$3.2 \cdot 10^{-3}$	$\sim 2.6 \cdot 10^{-5}$	-

TABLE 5
Estimated number of (dominating) operations per step and number of steps to converge. *Note that the SS in Figure 4 does not convergence. The estimated running time relative to Euler is calculated as the product of the Sum/step relative to Euler and the Estimated steps relative to Euler.

everywhere and instead more emphasis is placed upon distinct features, such as dominating sulci and gray/white matter transitions.

Consistency of diffeomorphisms generated by SVF's enable symmetry and symmetry is a contributing factor to the improvement. We can only speculate as to what gain can be obtained by other factors. However, on publicly available evaluation data, the combination of SVF, the implicit trapezoidal method and NMI performs better than the current state-of-the-art on a wide range of publicly available datasets.

6.1 Complexity and computability

Accurate approximations to ODE's can be expensive to compute. However, as we have demonstrated, the benefits of an accurate solution clearly outweigh the computational overhead. The computational complexity of the Trapezoidal scheme is low given the results, and the extra computations are well spent. In fact, 40 time steps with the Trapezoidal scheme using the described configuration computationally corresponds to four squaring operations of all voxels. It is also worth noting that as **CDD** is symmetric, only 780 (LPBA), 153 (IBSR), 55 (CUMC12) and 45 (MGH10) registrations were required, half the number of non-rigid registrations performed by Klein et al. [35].

6.2 The Choice of ODE solver

Consistency is inherent to diffeomorphisms and this key property of the diffeomorphism must be upheld to machine precision in order to claim that a registration is diffeomorphic. We chose the trapezoidal method over RK4, Forward Euler and SS, because of the four, the trapezoidal method is the only method which is A-stable and, due to its co-location property, guarantees consistency to machine precision. Due to their explicit nature and limited stability the results from SS, Euler or RK4 must be checked for convergence as well as their approximation to the diffeomorphic properties, by performing a convergence study. Considering the stability, complexity and consistency of SS, Forward Euler and RK4 we believe that the implicit A-stable colocation methods like the trapezoidal method, illustrated in this paper, are the optimal choice of methods for estimating diffeomorphisms in general and in particular diffeomorphisms based on SVF.

6.3 Limitations

Although **CDD** shows very promising results, it is not without limitations. **CDD** is restricted to diffeomorphic registrations, and practical problems exist where the use of diffeomorphisms may be suboptimal. Common to all registration methods is the sensitivity to the choice of parameters. As registration is domain specific, the optimal parameters are usually unknown, which makes comparisons between registration methods challenging. For instance, the **CDD** uses the magnitude of the velocity for regularization, which indirectly inhibits large deformations. Therefore, the prior used for normal subjects would not be applicable to, for example, AD subjects where it would need to account for the larger variation caused by the pathology. This may mean that other methods such as SyN or Diffeomorphic Demons may, with correspondingly adapted parameters, perform better in such situations. Another limitation of the evaluation performed in this paper and in [35] is the restriction to healthy subjects. However, to the best of our knowledge, no large, publicly available, fully-labelled dataset of pathology subjects exists. Until such datasets become available, quality assessment of neuroanatomical registration is restricted to evaluations on healthy subjects. Hence the small AD dataset presented herein can only serve as examples of the applicability of **CDD** to data with pathology.

7 Conclusion

It is clear from the literature and by the additional experiments provided in this paper that state-of-the-art diffeomorphic registration frameworks are poor approximations to diffeomorphisms. We show that diffeomorphic properties to machine precision can be obtained by using implicit A-stable collocation methods. To this end, we have presented **CDD** (**C**ollocation for **D**iffeomorphic **D**eformations) a state-of-the-art registration framework, which produces diffeomorphic registrations to machine precision and state-of-the-art registration results across all four brain test datasets (LPBA, IBSR CUMC12 and MGH10). We illustrated **CDD**'s applicability to data with pathology on a small set of AD subjects from ADNI. Although **CDD** performs better on the reference datasets, registration is domain specific and parameter sensitive, hence we cannot claim that **CDD** is optimal in all cases. The main contribution of this work is that, in contrast to existing methods, **CDD** provides diffeomorphic deformations to machine precision. The key features are the use of the Trapezoidal scheme to obtain a good trade-off between running time and convergence rate, A-stability to guarantee convergence, and the fulfillment of the collocation property [47], which ensures that the numerical solution to the ODE has identical properties to the continuous formulation of the SVF. The Trapezoidal method is an implicit A-stable collocation method and is the most accurate linear multistep scheme [44], effectively providing **CDD** with mappings which are accurately invertible, consistent to machine precision and produce state-of-the-art non-rigid registration of neuroanatomical MRI data.

ACKNOWLEDGMENT

The authors would like to thank the Oticon Foundation for supporting our research

REFERENCES

[1] G. E. Christensen and H. J. Johnson, "Consistent image registration," *Medical Imaging, IEEE Transactions on*, vol. 20, no. 7, pp. 568–582, 2001.

[2] G. E. Christensen, "Consistent linear-elastic transformations for image matching," in *Information processing in medical imaging*. Springer, 1999, pp. 224–237.

[3] T. Rohlfing, "Image similarity and tissue overlaps as surrogates for image registration accuracy: widely used but unreliable," *Medical Imaging, IEEE Transactions on*, vol. 31, no. 2, pp. 153–163, 2012.

[4] J. Ashburner, "A fast diffeomorphic image registration algorithm," *Neuroimage*, vol. 38, no. 1, pp. 95–113, 2007.

[5] M. Bossa, E. Zacur, and S. Olmos, "Algorithms for computing the group exponential of diffeomorphisms: Performance evaluation," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE, 2008, pp. 1–8.

[6] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Symmetric log-domain diffeomorphic registration: A demons-based approach," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2008*. Springer, 2008, pp. 754–761.

[7] B. B. Avants, P. T. Schoenemann, and J. C. Gee, "Lagrangian frame diffeomorphic image registration: Morphometric comparison of human and chimpanzee cortex," *Medical image analysis*, vol. 10, no. 3, pp. 397–412, 2006.

[8] C. Studholme, D. L. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3d medical image alignment," *Pattern recognition*, vol. 32, no. 1, pp. 71–86, 1999.

[9] S. Darkner and J. Sporring, "Locally orderless registration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 6, pp. 1437–1450, 2013.

[10] V. Arsigny, O. Commowick, X. Pennec, and N. Ayache, "A log-euclidean framework for statistics on diffeomorphisms," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*. Springer, 2006, pp. 924–931.

[11] P. Cachier and D. Rey, "Symmetrization of the non-rigid registration problem using inversion-invariant energies: application to multiple sclerosis," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2000*. Springer, 2000, pp. 472–481.

[12] B. Avants and J. C. Gee, "Geodesic estimation for large deformation anatomical shape averaging and interpolation," *NeuroImage*, vol. 23, pp. S139–S150, 2004.

[13] S. Benhimane and E. Malis, "Homography-based 2d visual tracking and servoing," *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 661–676, 2007.

[14] D. W. Shattuck, M. Mirza, V. Adisetiyo, C. Hojatkashani, G. Salamon, K. L. Narr, R. A. Poldrack, R. M. Bilder, and A. W. Toga, "Construction of a 3d probabilistic atlas of human cortical structures," *Neuroimage*, vol. 39, no. 3, pp. 1064–1080, 2008.

[15] A. C. Evans, D. L. Collins, and B. Milner, "An MRI-based stereotactic brain atlas from 300 young normal subjects," *Proceedings of the 22nd Symposium of the Society for Neuroscience*, p. 408, 1992.

[16] V. Caviness, J. Meyer, N. Makris, and D. Kennedy, "Mri-based topographic parcellation of human neocortex: an anatomically specified method with estimate of reliability," *Cognitive Neuroscience, Journal of*, vol. 8, no. 6, pp. 566–587, 1996.

[17] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, "Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain," *Medical image analysis*, vol. 12, no. 1, pp. 26–41, 2008.

[18] G. E. Christensen, S. C. Joshi, M. Miller *et al.*, "Volumetric transformation of brain anatomy," *Medical Imaging, IEEE Transactions on*, vol. 16, no. 6, pp. 864–877, 1997.

[19] G. Christensen, R. Rabbitt, and M. Miller, "Deformable templates using large deformation kinematics," *Image Processing, IEEE Transactions on*, vol. 5, no. 10, pp. 1435–1447, 1996.

[20] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes, "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," *International journal of computer vision*, vol. 61, no. 2, pp. 139–157, 2005.

[21] S. Durrleman, M. Prastawa, G. Gerig, and S. Joshi, "Optimal data-driven sparse parameterization of diffeomorphisms for population analysis," in *Biennial International Conference on Information Processing in Medical Imaging*. Springer, 2011, pp. 123–134.

[22] M. Zhang and P. T. Fletcher, "Finite-dimensional lie algebras for fast diffeomorphic image registration," in *International Conference on Information Processing in Medical Imaging*. Springer, 2015, pp. 249–260.

[23] X. Pennec, R. Stefanescu, V. Arsigny, P. Fillard, and N. Ayache, "Riemannian Elasticity: A Statistical Regularization Framework for Non-linear Registration," *MICCAI*, vol. 3750, p. 943, 2005.

[24] D. Rueckert, P. Aljabar, R. A. Heckemann, J. V. Hajnal, and A. Hammers, "Diffeomorphic registration using b-splines," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2006*. Springer, 2006, pp. 702–709.

[25] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes, "Nonrigid registration using free-form deformations: application to breast MR images," *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 712–21, 1999.

[26] S. M. Smith, M. Jenkinson, M. W. Woolrich, C. F. Beckmann, T. E. Behrens, H. Johansen-Berg, P. R. Bannister, M. De Luca, I. Drobnjak, D. E. Flitney *et al.*, "Advances in functional and structural mr image analysis and implementation as fsl," *Neuroimage*, vol. 23, pp. S208–S219, 2004.

[27] M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich, and S. M. Smith, "Fsl," *Neuroimage*, vol. 62, no. 2, pp. 782–790, 2012.

[28] S. Klöppel, C. M. Stonnington, C. Chu, B. Draganski, R. I. Scahill, J. D. Rohrer, N. C. Fox, C. R. Jack, J. Ashburner, and R. S. Frackowiak, "Automatic classification of mr scans in alzheimer's disease," *Brain*, vol. 131, no. 3, pp. 681–689, 2008.

[29] R. Cuingnet, E. Gerardin, J. Tessieras, G. Auzias, S. Lehéricy, M.-O. Habert, M. Chupin, H. Benali, O. Colliot, A. D. N. Initiative *et al.*, "Automatic classification of patients with alzheimer's disease from structural mri: a comparison of ten methods using the adni database," *neuroimage*, vol. 56, no. 2, pp. 766–781, 2011.

[30] L. Ying and E. J. Candes, "The phase flow method," *Journal of computational physics*, vol. 220, pp. 184–215, 2006.

[31] ——, "Fast geodesics computation with the phase flow method," *Journal of computational physics*, vol. 220, no. 1, pp. 6–18, 2006.

[32] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Non-parametric diffeomorphic image registration with the demons algorithm," *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pp. 319–326, 2007.

[33] A. Pai, S. Sommer, S. Darkner, L. Sorensen, J. Sporring, and M. Nielsen, "Stepwise inverse consistent euler a scheme for diffeomorphic image registration," in *Biomedical Image Registration*, ser. Lecture Notes in Computer Science, S. Ourselin and M. Modat, Eds. Springer International Publishing, 2014, vol. 8545, pp. 223–230.

[34] M. Bossa, M. Hernandez, and S. Olmos, "Contributions to 3d diffeomorphic atlas estimation: application to brain images," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*. Springer, 2007, pp. 667–674.

[35] A. Klein, J. Andersson, B. A. Ardekani, J. Ashburner, B. Avants, M.-C. Chiang, G. E. Christensen, D. L. Collins, J. Gee, P. Hellier *et al.*, "Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration," *Neuroimage*, vol. 46, no. 3, p. 786, 2009.

[36] D. L. Collins, P. Neelin, T. M. Peters, and A. C. Evans, "Automatic 3d intersubject registration of mr volumetric data in standardized talairach space." *Journal of computer assisted tomography*, vol. 18, no. 2, pp. 192–205, 1994.

[37] B. A. Ardekani and A. H. Bachman, "Model-based automatic detection of the anterior and posterior commissures on mri scans," *Neuroimage*, vol. 46, no. 3, pp. 677–682, 2009.

[38] R. P. Woods, S. T. Grafton, J. D. Watson, N. L. Sicotte, and J. C. Mazziotta, "Automated image registration: Ii. intersubject valida-

tion of linear and nonlinear models," *Journal of computer assisted tomography*, vol. 22, no. 1, pp. 153–165, 1998.

[39] P. Hellier, C. Barillot, É. Mémin, and P. Pérez, "Hierarchical estimation of a dense deformation field for 3-d robust registration," *Medical Imaging, IEEE Transactions on*, vol. 20, no. 5, pp. 388–402, 2001.

[40] M.-C. Chiang, R. Dutton, K. M. Hayashi, A. W. Toga, O. L. Lopez, H. J. Aizenstein, J. T. Becker, P. M. Thompson *et al.*, "Fluid registration of medical images using jensen-renyi divergence reveals 3d profile of brain atrophy in hiv/aids," in *Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on*. IEEE, 2006, pp. 193–196.

[41] M. Modat, P. Daga, M. J. Cardoso, S. Ourselin, G. R. Ridgway, and J. Ashburner, "Parametric non-rigid registration using a stationary velocity field," in *Mathematical Methods in Biomedical Image Analysis (MMBIA), 2012 IEEE Workshop on*. IEEE, 2012, pp. 145–150.

[42] Y. Ou, H. Akbari, M. Bilello, X. Da, and C. Davatzikos, "Comparative evaluation of registration algorithms in different brain databases with varying difficulty: results and insights," *Medical Imaging, IEEE Transactions on*, vol. 33, no. 10, pp. 2039–2065, 2014.

[43] M. R. Sabuncu, B. T. Yeo, K. Van Leemput, T. Vercauteren, and P. Golland, "Asymmetric image-template registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2009, pp. 565–573.

[44] G. G. Dahlquist, "A special stability problem for linear multistep methods," *BIT*, vol. 3, pp. 27–43, 1963.

[45] O. Azencot, S. Weißmann, M. Ovsjanikov, M. Wardetzky, and M. Ben-Chen, "Functional fluids on surfaces," *Computer Graphics Forum*, vol. 33, no. 5, pp. 237–246, 2014.

[46] T. Preußer and M. Rumpf, "Extracting motion velocities from 3d image sequences and coupled spatio-temporal smoothing," in *Electronic Imaging 2003*. International Society for Optics and Photonics, 2003, pp. 181–192.

[47] H. Brunner, *Collocation methods for Volterra integral and related functional differential equations*. Cambridge University Press, 2004, vol. 15.

[48] J. Koenderink and A. Van Doorn, "The structure of locally orderless images," *International Journal of Computer Vision*, vol. 31, no. 2, pp. 159–168, 1999.

[49] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[50] E. Süli and D. F. Mayers, *An introduction to numerical analysis*. Cambridge university press, 2003.

[51] R. S. Palais and R. A. Palais, *Differential equations, mechanics, and computation*. American Mathematical Soc., 2009, vol. 51.

[52] J. Talairach and P. Tournoux, *Co-Planar Stereotaxic Atlas of the Human Brain: 3-D Proportional System: An Approach to Cerebral Imaging (Thieme Classics)*. Thieme, Jan. 1988. [Online]. Available: http://www.worldcat.org/isbn/0865772932

[53] M. Boccardi, M. Bocchetta, F. C. Morency, D. L. Collins, M. Nishikawa, R. Ganzola, M. J. Grothe, D. Wolf, A. Redolfi, M. Pievani *et al.*, "Training labels for hippocampal segmentation based on the eadc-adni harmonized hippocampal protocol," *Alzheimer's & Dementia*, vol. 11, no. 2, pp. 175–183, 2015.

[54] J. Sled, A. Zijdenbos, and A. Evans, "A nonparametric method for automatic correction of intensity nonuniformity in mri data," *IEEE Transactions on Medical Imaging*, vol. 17, no. 1, pp. p. 87–97, 1998.

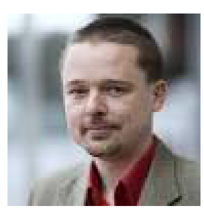
[55] S. Darkner and J. Sporring, "Generalized partial volume: An inferior density estimator to parzen windows for normalized mutual information," in *IPMI*, ser. LNCS, vol. 6801. Springer, 2011, pp. 436–447.

[56] J. D. Lawson, "Generalized runge-kutta processes for stable systems with large lipschitz constants," *SIAM Journal on Numerical Analysis*, vol. 4, no. 3, pp. 372–380, 1967.

Sune Darkner received his Masters Degree in Applied Mathematics in 2004 and founded a software company building databases for the telecommunication industry. In collaboration with Oticon A/S, a large hearing aid manufacturer, he received his Industrial Ph.D. degree in Shape and Deformation Analysis of the Human Ear Canal in 2009, from the Department of Informatics and Mathematical Modelling, at the Technical University of Denmark (DTU). After holding a position at an energy company as data analyst, he rejoined the Department of Informatics and Mathematical Modelling at DTU in 2009 as a post doc. He currently holds a position as Associate Professor in image analysis at the Department of Computer Science, University of Copenhagen. Research interests include image registration and classification, optimization and regularization, and computational physics.

Akshay Pai received his PhD degree in "Deformation-based atrophy estimation for Alzheimer's disease" from the University of Copenhagen in 2015. Prior to his PhD, he spent his time at GE Global research working on "derivation of intervertebral disk orientations by integrating demographic priors to enhance spine structure analysis". He currently holds a position as an Industrial Postdoc jointly at Biomediq A/S, and the Department of computer science, University of Copenhagen. His research interests include registration of functional and image data, and application of deep learning in medical image quantification.

Matthew G. Liptrot has more than 20 years experience in medical imaging, including fMRI, PET and most recently diffusion MRI and tractography. He graduated from Imperial College in London with a BEng in Electrical and Electronic Engineering in 1993, and remained there to study for an MSc in Biomedical Engineering, which he attained with Distinction in 1994. He subsequently also received his PhD from Imperial College in 2011. He has worked for 12 years in the hospital research sector in Copenhagen, before returning to academia as a postdoc in the Image Group, Department of Computer Science, Copenhagen University in 2012. He is currently Assistant Professor, and has since 2014 also held a joint position in the Cognitive Systems Section, Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU).



Jon Sporring received his Master and Ph.D. degree from the Department of Computer Science, University of Copenhagen, Denmark in 1995 and 1998, respectively. He has been a visiting research/professor at IBM Research Center, Almaden, California, USA; Computer Vision and Robotics Lab at Foundation for Research & Technology - Hellas, Greece; Nordic Bioscience a/s; and School of Computer Science, McGill University, Montreal, Canada. Jon Sporring also co-founded DigiCorpus Aps in 2012. From 2000-2003 he was assistant research professor at 3D-Lab, School of Dentistry, University of Copenhagen. Since 2003 he has been employed as associate professor at the Department of Computer Science, University of Copenhagen. In 2007-2012 he was and again since 2015 he is Vice-Chair for Research at Department of Computer Science, University of Copenhagen.