

Article

CLAIRE—Parallelized Diffeomorphic Image Registration for Large-Scale Biomedical Imaging Applications

Naveen Himthani ^{1,*}, Malte Brunn ², Jae-Youn Kim ³, Miriam Schulte ², Andreas Mang ³  and George Biros ¹¹ Oden Institute, The University of Texas at Austin, Austin, TX 78712, USA² Institute for Parallel and Distributed Systems, University of Stuttgart, 70569 Stuttgart, Germany³ Department of Mathematics, University of Houston, Houston, TX 77004, USA

* Correspondence: naveen.himthani@utexas.edu

Abstract: We study the performance of CLAIRE—a diffeomorphic multi-node, multi-GPU image-registration algorithm and software—in large-scale biomedical imaging applications with billions of voxels. At such resolutions, most existing software packages for diffeomorphic image registration are prohibitively expensive. As a result, practitioners first significantly downsample the original images and then register them using existing tools. Our main contribution is an extensive analysis of the impact of downsampling on registration performance. We study this impact by comparing full-resolution registrations obtained with CLAIRE to lower resolution registrations for synthetic and real-world imaging datasets. Our results suggest that registration at full resolution can yield a superior registration quality—but not always. For example, downsampling a synthetic image from 1024^3 to 256^3 decreases the Dice coefficient from 92% to 79%. However, the differences are less pronounced for noisy or low contrast high resolution images. CLAIRE allows us not only to register images of clinically relevant size in a few seconds but also to register images at unprecedented resolution in reasonable time. The highest resolution considered are CLARITY images of size $2816 \times 3016 \times 1162$. To the best of our knowledge, this is the first study on image registration quality at such resolutions.

Keywords: large-scale biomedical image processing; diffeomorphic image registration; high-performance computing; GPUs



Citation: Himthani, N.; Brunn, M.; Kim, J.-Y.; Schulte, M.; Mang, A.; Biros, G. CLAIRE—Parallelized Diffeomorphic Image Registration for Large-Scale Biomedical Imaging Applications. *J. Imaging* **2022**, *8*, 251. <https://doi.org/10.3390/jimaging8090251>

Academic Editors: Cecilia Di Ruberto, Alessandro Stefano, Albert Comelli, Lorenzo Putzu and Andrea Loddo

Received: 30 July 2022

Accepted: 6 September 2022

Published: 16 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

3D diffeomorphic image registration (also known as “image alignment” or “matching”) is a critical task in biomedical image analysis [1,2]. For example, it enables the study of morphological changes associated with the progression of neurodegenerative diseases over time or in imaging studies of patient populations. The process of image registration involves finding a spatial transformation which maps corresponding points in an image to those in another [1]. In mathematical notation, we are given two images $m_0(x)$ (the template/moving image) and $m_1(x)$ (the reference/fixed image; here $x \in \Omega \subset \mathbb{R}^3$) and we seek a spatial transformation $y : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, such that the deformed template image $m_0(y(x))$ is similar to the reference image $m_1(x)$ for all x (see Figure 1 for an illustration) [3,4]. Image registration methods can be categorized based on the parameterization for y [3]. We seek a *diffeomorphic* map y , i.e., y is a differentiable bijection and has a differentiable inverse. Approaches that parameterize y in terms of a smooth, time-varying velocity field $v : \mathbb{R}^3 \times [0, 1] \rightarrow \mathbb{R}^3$ belong to a class of methods referred to as *large-deformation diffeomorphic metric mapping* (LDDMM) [5–7]. In this study, we consider a related class of methods that use stationary velocity fields $v : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. This diffeomorphic registration problem is expensive to solve because the problem is infinite-dimensional, and upon discretization results in a nonlinear system with millions of unknowns—even for stationary velocity fields. For example, solving the registration problem for two images of resolution 256^3 (a typical size for clinical scans) requires solving for approximately 50 million unknowns in space

(three vector components per image grid point). Furthermore, image registration is a highly nonlinear, ill-posed inverse problem [8], resulting in ill-conditioned inversion operators. Consequently, running registration on multi-core high-end CPUs can take several minutes.

There exist various algorithms and software packages for fast registration of images at standard clinical resolution (e.g., 256^3) [9–16]. This includes CLAIRE, which can execute image registration in parallel on multi-node multi-core CPUs and GPUs [17–22]. We note that there is little work on scalable image registration. One application that requires this scalability is the registration of CLARITY images [23–25,25–27] with a resolution in the order of $20\text{K} \times 20\text{K} \times 1\text{K}$. This corresponds to a problem with approximately 1.2 trillion unknowns. In [22], we extended CLAIRE to support GPU-accelerated scalable image registration, which can process high resolution images using multiple GPUs. We demonstrated the scalability of our solver using synthetic images with a resolution up to 2048^3 and CLARITY mouse brain images of size $768 \times 768 \times 1024$. In this work, we scale registration to an even higher resolution, e.g., CLARITY images of size $2816 \times 3016 \times 1162$. This corresponds to an increase of $16\times$ in problem size. In our previous work [17,19,28–33], we have extensively studied the algorithmic side of image registration within the framework of CLAIRE. In this paper, we pay closer attention to the quality of the registration results. We study the effect of different input parameters, including the quality and resolution of the input images, on the accuracy of the registration.

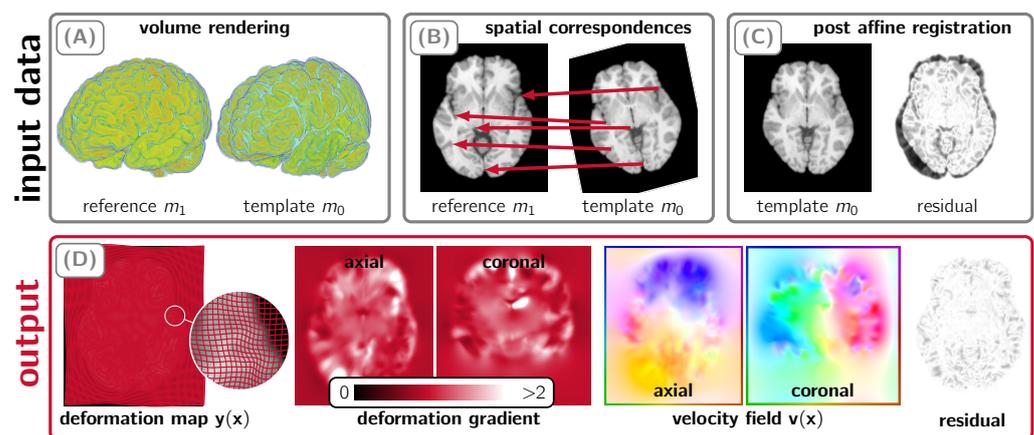


Figure 1. Illustration of the image registration problem. Panel (A): 3D rendering of an exemplary set of input images. Panel (B): Image registration is the task of computing spatial correspondences between two images of the same object. These correspondences, denoted as y , are indicated by the red arrows for example points within a single axial slice of the template and reference image data shown in panel (A). Before we execute CLAIRE, we compensate for the global mismatch between the considered images by performing an affine registration. In panel (C), we show an axial slice of the volume shown in panel (A) after an affine pre-registration step has been carried out; we execute CLAIRE on these images. Panel (D): CLAIRE outputs a diffeomorphic deformation map y that matches each point in the template image m_0 to its corresponding point in the reference image m_1 . We show a typical deformation map y in the leftmost image and the corresponding determinant of the deformation gradient (encodes volume change) in the second and third image from the left (axial and coronal slice). In CLAIRE, we invert for a stationary velocity field v that parameterizes y (second and third figure from the right; color denotes orientation). The last figure in panel (D) shows the point-wise residual after applying CLAIRE.

1.1. Contributions

We build on our prior work on scalable deformable image registration [17,19–22,28–31,33] using CLAIRE and analyze the effect of image resolution on image registration accuracy. The present work analyzes image registration performance. We do not propose any major improvements in our methodology, with the exception of additional advice for hyperpa-

parameter tuning. We outline our past contributions on formulations, algorithms, and their parallel implementation below. Our major contributions in the present work are:

- We evaluate CLAIRE on high resolution synthetic and real image datasets. We demonstrate that image registration when performed at native high resolution results in higher accuracy (measured in terms of the Dice coefficient of the labeled structures in the images). We conduct experiments to show that downsampling the images and then registering them result in loss of registration accuracy.
- We design scalable image registration experiments to explore the effect of solver parameters—the number of time steps n_t in the semi-Lagrangian scheme, and regularization parameters β_v and β_w —on the registration performance at different image resolutions.
- We present an extension of the regularization parameter continuation scheme first presented in [28] by searching for β_w in addition to β_v , thereby removing the need for selecting an additional resolution-dependent solver parameter.
- We study the performance of our scalable registration solver CLAIRE for applications in high-resolution mouse and human neuroimage registration. We perform image registration for two pairs of CLARITY mouse brain images at a resolution of $2816 \times 3016 \times 1162$ voxels. To the best of our knowledge, images of this scale have not been registered before at full resolution in under 30 min.

1.2. Related Work

The current work builds upon the open source framework CLAIRE [17,19–22,28–32,34]. Our formulation for diffeomorphic image registration has been described in [28,29]. Our Newton–Krylov solver was originally developed in [28]. We proposed efficient numerical implementations for evaluating forward and adjoint operators in [17,30,33]. We designed various methods for preconditioning in [19,22,28,30]. The computational kernels of the parallel CPU implementation of our solver were introduced in [17,19,32]. More recently, we ported CLAIRE to GPU architectures [20,22]. In summary, our work paved the way towards real-time applications of diffeomorphic image registration and its deployment to a high-resolution medical imaging application. To the best of our knowledge, this is the only existing software for diffeomorphic image registration with these capabilities. We have integrated our framework with biophysical modeling in [31,32,35–37]. None of these works explore registration performance in large-scale biomedical imaging applications.

Literature surveys of image registration and associated algorithmic developments can be found in [2,3]. A recent overview of existing LDDMM methods can be found in [19]. Related LDDMM software packages include Demons [9], ANTs [38–40], DARTEL [41], deformetrica [42–45], FLASH [15], LDDMM [5,46], ARDENT [47], ITKNDReg [48], and PyCA [49]. Surveys of GPU-accelerated image registration solvers can be found in [50–52]; particular examples for various formulations are [14,16,43,53–66]. Multi-GPU LDDMM implementations for atlas construction are described in [57,58,65,66]. Their setup is embarrassingly parallel in the sense that they solve many small registration problems independently on single GPUs. In [57–59], the computational bottlenecks are the repeated solution of a Helmholtz-type PDE and trilinear scattered data interpolation to compute and apply the deformation map. They use hardware acceleration for the trilinear interpolation kernel with 3D texture volume support. The runtime for a single dataset of size $160 \times 192 \times 160$ is 20 s on an NVIDIA Quadro FX 5600. CLAIRE uses a multi-node multi-GPU framework with high computational throughput for single (large-scale) registration problems [22] which is no longer an embarrassingly parallel problem. CLAIRE uses the Message Passing Interface (MPI) to parallelize the implementation.

None of the GPU-accelerated LDDMM methods mentioned above, except for CLAIRE [17,19–22,28–32,34], use second-order numerical optimization. Many of the available methods solve the registration problem by reducing the number of unknowns either through a coarse parameterization or by using a coarse grid and use simplified algo-

rithms. These crude approximations and simplifications can result in inferior registration quality [19,20].

The work in [25] focuses on annotating CLARITY brain images by registering them to the Allen Institute’s Mouse Reference Atlas (ARA). They use a “masked” LDDMM approach. They also consider the registration of CLARITY-to-CLARITY brain images and compare different mismatch terms for the registrations. However, they downscale the images to a lower resolution for conducting all experiments. In [25], mutual information is used for the registration of CLARITY to the ARA dataset but at an approximately one hundred times downsampled resolution (at an original in-plane isotropic resolution of $0.58\ \mu\text{m}$). The authors in [67] analyze registration performance on high-resolution mouse brain images of size $2560 \times 2160 \times 633$ obtained using the CUBIC protocol [68]. They report results using different software packages including ANTs and Elastix. They did not observe a relationship between registration accuracy at different resolutions. For their high-resolution runs using ANTs, they report a wall clock time of over 200 h on a single compute node (2.66 GHz 64bit Intel Xeon processor with 256 GB RAM) while the same run with elastix [69] took approximately 30 h. The authors in [70] register high-resolution images of mouse brains to the ARA dataset [71]. They perform nonlinear registration using ANTs at coarse resolution ($10\ \mu\text{m}$ for the ARA) and apply the deformation at high-resolution. In the current work, we do not downsample high-resolution images but register them at the original resolution. We can register CLARITY images of resolution $2816 \times 3016 \times 1162$ in less than 30 min using 256 GPUs. In addition to that, we study the effect of resolution on the registration quality.

1.3. Outline

We summarize the overall formulation in Section 2.1 and the algorithms in Section 2.2 for completeness. We note, that all of the material presented in Sections 2.1 and 2.2 has been discussed in detail in [19,22]. In Section 3, we present our kernels and parallel algorithms and discuss key solver parameters. We also introduce a new scheme to automatically identify adequate parameters of our solver for unseen data. This scheme extends on our prior work in [19]. We conclude with the main scalability experiments in Section 5, and present conclusions in Section 6.

1.4. Limitations

CLAIRE currently only supports mono-modal similarity measures, which limits our study to registrations for images acquired with the same imaging modality. Moreover, CLAIRE only supports periodic boundary conditions, i.e., we require that the image data be embedded in a larger background domain. In most medical imaging applications, the images are embedded in a zero background and, therefore, naturally periodic. If the images are not periodic, they can be zero-padded and mollified. CLAIRE uses stationary velocities, which improves computational efficiency, but is suboptimal from a theoretical point of view. In [28], we found no qualitative differences in registration mismatch when registering two images using stationary velocities. This observation is in line with the work of other groups using stationary velocity fields [72,73]. Regarding computational performance, one issue is the memory requirement of our method. We have optimized memory allocation for the core components of CLAIRE. Additional optimizations by reusing and sharing memory across external libraries to further reduce the memory load remain subject to future work.

2. Methods

Before discussing our enhancements in Section 3, we shortly introduce the underlying mathematical formulation of the image registration problem utilized in CLAIRE as well as the discretization and the numerical algorithms. The following exposition is only included for completeness, and is based on material described in our prior work on efficient

algorithms for diffeomorphic image registration [17,19,20,22,28–30,32,33]. Consequently, we keep this section brief.

2.1. Formulation

We summarize our notation in Table 1. CLAIRE uses an optimal control formulation. We parameterize the deformation map $\mathbf{y}(\mathbf{x})$ through a smooth, stationary velocity field $\mathbf{v}(\mathbf{x})$. The optimization problem is: Given two images $m_0(\mathbf{x})$ (template image; image to be deformed) and $m_1(\mathbf{x})$ (reference image), we seek a stationary velocity field $\mathbf{v}(\mathbf{x})$ by solving

$$\underset{v,m}{\text{minimize}} \quad \frac{1}{2} \int_{\Omega} (m(\mathbf{x}, 1) - m_1(\mathbf{x}))^2 d\mathbf{x} + \frac{\beta_v}{2} \text{reg}_v(v) + \frac{\beta_w}{2} \text{reg}_w(w) \tag{1}$$

subject to

$$\partial_t m(\mathbf{x}, t) + \mathbf{v}(\mathbf{x}) \cdot \nabla m(\mathbf{x}, t) = 0 \quad \text{in } \Omega \times (0, 1], \tag{2}$$

$$m(\mathbf{x}, t) = m_0(\mathbf{x}) \quad \text{in } \Omega \times \{0\}, \tag{3}$$

$$\nabla \cdot \mathbf{v} = w \quad \text{in } \Omega, \tag{4}$$

on a rectangular domain $\Omega \subset \mathbb{R}^3$ with periodic boundary conditions on $\partial\Omega$. The first term in Equation (1) is a squared L^2 image similarity metric, which measures the *distance* between the deformed template image $m(\mathbf{x}, t = 1)$ and the reference image $m_1(\mathbf{x})$. The objective functional in Equation (1) additionally consists of two regularization models that act on the controls v and w with regularization parameters $\beta_v > 0$ and $\beta_w > 0$, respectively. The regularization operators are introduced to prescribe sufficient regularity requirements on v and its divergence $\nabla \cdot v$. Smoothness of the velocity guarantees that the computed map is diffeomorphic [5–7]. We refer to [29] for details about our regularization scheme. The default configuration of CLAIRE is an H^1 -Sobolev-seminorm for v and H^1 -Sobolev-norm for w [19,29]. The transport equation in Equation (3) represents the geometrical deformation of $m_0(\mathbf{x})$ by advecting the intensities forward in time.

To solve Equation (1) subject to Equations (2)–(4), we apply the method of Lagrange multipliers to obtain the Lagrangian functional

$$\begin{aligned} \mathcal{L}(\boldsymbol{\phi}) := & \frac{1}{2} \int_{\Omega} (m(\mathbf{x}, 1) - m_1(\mathbf{x}))^2 d\mathbf{x} + \frac{\beta_v}{2} \text{reg}_v(v) \\ & + \frac{\beta_w}{2} \text{reg}_w(\nabla \cdot \mathbf{v}) + \int_0^1 \int_{\Omega} \lambda(\mathbf{x}, t) (\partial_t m + \mathbf{v} \cdot \nabla m) d\mathbf{x} dt \\ & + \int_{\Omega} \lambda(\mathbf{x}, 0) (m(\mathbf{x}, 0) - m_0(\mathbf{x})) d\mathbf{x} + \int_{\Omega} p(\mathbf{x}) (\nabla \cdot \mathbf{v} - w) d\mathbf{x} \end{aligned} \tag{5}$$

with state, adjoint, and control variables $(m, \lambda, p, \mathbf{v}, w) := \boldsymbol{\phi}$, respectively.

Table 1. Notation and main symbols.

Symbol	Description
Ω	spatial domain; $\Omega := [0, 2\pi)^3 \subset \mathbb{R}^3$ with boundary $\partial\Omega$
\mathbf{x}	spatial coordinate; $\mathbf{x} := (x_1, x_2, x_3)^T \in \mathbb{R}^3$
t	(pseudo-)time variable; $t \in [0, 1]$
$m_1(\mathbf{x})$	reference image (fixed image)
$m_0(\mathbf{x})$	template image (moving image)
$\mathbf{v}(\mathbf{x})$	stationary velocity field
$\mathbf{y}(\mathbf{x})$	(diffeomorphic) deformation map
$m(\mathbf{x}, t)$	state variable (transported intensities of m_0)
$\lambda(\mathbf{x}, t)$	adjoint variable
\mathcal{A}	regularization operator
$\beta_v > 0$	regularization parameter for v
$\beta_w > 0$	regularization parameter for $\nabla \cdot v$
\mathbf{F}	deformation gradient

Table 1. Cont.

Symbol	Description
J	determinant of deformation gradient (Jacobian determinant)
n_t	number of time steps in PDE solver
CFL	Courant–Friedrichs–Lewy (number/condition)
FD	finite differences
FFT	Fast Fourier Transform
IP	scattered data interpolation
LDDMM	Large Deformation Diffeomorphic Metric Mapping
MPI	Message Passing Interface
PCG	Preconditioned Conjugate Gradient (method)

2.2. Discretization and Numerical Algorithms

2.2.1. Optimality Conditions & Reduced Space Approach

To derive the first order optimality conditions, we take the variations of \mathcal{L} with respect to the state variable m , the adjoint variables λ and p , and the control variable v . This results in a set of coupled, hyperbolic-elliptic PDEs in $4D$ (space-time). CLAIRE uses a *reduced-space approach*, in which one iterates only on the reduced-space of v . We require $\mathbf{g}(v^*) = \mathbf{0}$ for an admissible solution v^* , where

$$\mathbf{g}(v) := \beta_v \mathcal{A}v(x) + \mathcal{K} \int_0^1 \lambda(x, t) \nabla m(x, t) dt \tag{6}$$

is the so-called *reduced gradient*. The operator \mathcal{A} corresponds to the first variation of the regularization model for v (i.e., reg_v in Equation (1)) and the operator \mathcal{K} projects v onto the space of near-incompressible velocity fields (see [29] for details). To evaluate Equation (6), we first solve the forward problem in Equation (3) and then the *adjoint problem* given by

$$-\partial_t \lambda(x, t) - \nabla \cdot \lambda(x, t)v(x) = 0 \quad \text{in } \Omega \times [0, 1] \tag{7}$$

with final condition $\lambda(x, t) = m_1(x) - m(x, t)$ in $\Omega \times \{1\}$ and periodic boundary conditions on $\partial\Omega$.

2.2.2. Discretization

We discretize the forward and adjoint PDEs in the space-time interval $\Omega \times [0, 1]$, $\Omega := [0, 2\pi]^3 \subset \mathbb{R}^3$, with periodic boundary conditions on $\partial\Omega$, on a regular grid with $N = N_1 N_2 N_3$ grid points $\mathbf{x}_{ijk} \in \mathbb{R}^3$ in space and $n_t + 1$ grid points in time. We use a semi-Lagrangian time-stepping method to solve the transport equations that materialize in the optimality system [17,30]. Key computational subcomponents of this scheme are 2nd-order Runge–Kutta time integrators and spatial interpolation kernels [17,19,22,30].

To solve the transport equation given by Equation (7) and to evaluate the reduced-gradient \mathbf{g} in Equation (6), we need to apply gradient and divergence operators. We use an 8th finite difference (FD) scheme for these first-order differential operators [20,22]. The reduced gradient in Equation (6) also involves the vector-Laplacian \mathcal{A} and the Leray-like operator \mathcal{K} (see [29]). In spectral methods, inversion and application of higher-order differential operators come at the cost of two FFTs and one Hadamard product in Fourier space.

2.2.3. Gauss–Newton–Krylov Solver

CLAIRE uses a Gauss–Newton–Krylov method globalized with an Armijo line search to solve the non-linear problem $\mathbf{g}(v^*) = \mathbf{0}$ [19,28]. The iterative scheme is given by

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \alpha_k \tilde{\mathbf{v}}_k, \quad \mathbf{H} \tilde{\mathbf{v}}_k = -\mathbf{g}_k, \quad k = 0, 1, 2, \dots, \tag{8}$$

where $\mathbf{H} \in \mathbb{R}^{3N, 3N}$ is the discretized reduced-space Hessian operator, $\tilde{\mathbf{v}}_k \in \mathbb{R}^{3N}$ the search direction, $\mathbf{g}_k \in \mathbb{R}^{3N}$ a discrete version of the gradient in Equation (6), $\alpha_k > 0$ a line search

parameter, and $k \in \mathbb{N}$ the Gauss–Newton iteration index. We have to solve the linear system in Equation (8) at each Gauss–Newton step. We do not form or assemble \mathbf{H} ; we use a matrix-free preconditioned conjugate gradient (PCG) method to solve $\mathbf{H}\tilde{\mathbf{v}}_k = -\mathbf{g}_k$ for $\tilde{\mathbf{v}}_k$. This only requires an expression for applying \mathbf{H} to a vector that we term *Hessian matvec*. In continuous form, the Gauss–Newton approximation of this matvec is given by

$$\mathcal{H}\tilde{\mathbf{v}} = \beta_v \mathcal{A}\tilde{\mathbf{v}}(x) + \mathcal{K} \int_0^1 \tilde{\lambda}(x, t) \nabla m(x, t) dt. \quad (9)$$

Similarly to the evaluation of the reduced gradient in Equation (6), the application of the Hessian to a vector in Equation (9) requires the solution of two PDEs to find the space-time field $\tilde{\lambda}$ (see [19,28,29] for details). Consequently, solving the linear system with \mathbf{H} in Equation (8) is the most expensive part of CLAIRE. Preconditioning of the reduced-space Hessian system can be used to alleviate these computational costs.

In [22], we have introduced a zero velocity approximation for \mathbf{H} as a preconditioner. This preconditioner can be applied at full resolution and through a two-level coarse grid approximation (see [22] for details). The latter variant represents the default considered in the present work.

3. Computational Kernels and Parallel Algorithms

At each Gauss–Newton step, we have to solve the forward and the adjoint equations for the reduced gradient and the Hessian matvecs. The main computational cost in CLAIRE constitute FFTs for (inverse) differential operators, scattered data interpolation (IP) for the semi-Lagrangian solver, and FD for computing first order derivatives (see [17,20,22,30] for a detailed description of these computational components). The distributed memory CPU implementation of CLAIRE uses AccFFT [74,75] for spectral operations [17,32]. In the single GPU setup, we use the highly optimized 3D FFT operations provided by NVIDIA’s cuFFT library. In the multi-node multi-GPU setup, we use a 2D slab decomposition to leverage 2D cuFFT functions. We decompose the spatial domain in x_1 direction, which is the outer-most dimension, and the spectral domain in the x_2 direction. Let p be the number of MPI tasks. Then, each MPI task gets $(N_1/p) \times N_2 \times N_3$ grid points, where N_1, N_2, N_3 are the image dimensions. We have discussed the implementation details and shown scalability of the FFT kernel in [22].

The parallel implementation of our IP kernel on CPUs was introduced in [17] and improved in [32]. In [20], we explored linear, cubic Lagrange, and cubic B-spline interpolation schemes for the interpolation kernel on a single GPU setup. In [22], we ported these kernels to the multi-node multi-GPU setup and made several optimizations. In the present study, we use linear interpolation to evaluate the image intensities at the off-grid points (also called characteristic points) in our semi-Lagrangian scheme. Depending on the image data layout and the velocity field, the IP kernel requires scattered peer-to-peer communication of off-grid points between the owner and the worker processors.

The CPU version of CLAIRE uses FFTs for spatial derivatives [17,19,32]. In [20], we introduced the 8th order FD kernel to evaluate first order derivatives, i.e., spatial gradients and divergence operators on a single GPU. In [22], we ported the FD kernel to the multi-GPU setup. We use the FD kernel for computing first-order derivatives throughout the registrations performed in this paper.

CLAIRE uses CUDA-aware MPI in the multi-node multi-GPU setup, thereby avoiding unnecessary CPU-GPU communication and automatically utilizing the high-speed on-node NVLink interconnect bus between GPUs if it is available.

3.1. Compute Hardware and Libraries

All runs reported in this study were executed on TACC’s Longhorn system in single precision. Longhorn hosts 96 NVIDIA Tesla V100 nodes. Each node is equipped with four GPUs and 16 GB GPU RAM each (i.e., 64 GB per node) and two IBM Power 9 processors with 20 cores (40 cores per node) at 2.3 GHz with 256 GB memory. Our implementation

uses PETSc [76,77] for linear algebra, the PETSc TAO package for nonlinear optimization, CUDA [78], thrust [79], cuFFT for FFTs [80], niftilib [81] for serial I/O for small images and PnetCDF [82] for parallel I/O for large scale images, IBM Spectrum MPI [83], and the IBM XL compiler [84].

3.2. Code Availability

CLAIRE [21,34] is available publicly for download on github at <https://github.com/andreasmanng/claire> (accessed on 5 January 2022) under the GNU General Public License v3.0.

3.3. Key Solver Parameters

Here, we summarize the key parameters of CLAIRE and discuss their effect on the solver and previous strategies to choose suitable values. In Section 3.4, we present our algorithm to choose these parameters in a combined continuation approach.

- **β_v —regularization parameter for the velocity field v .** Large values for β_v result in very smooth velocities and, thus, maps that are typically associated with a large final image mismatch. Smaller values of β_v allow complex deformations but lead to a solution that might be close to being non-diffeomorphic due to discretization issues. From a user application point of view, we are interested in computing velocity fields, for which the Jacobian determinant, i.e., the determinant of the deformation gradient $\mathbf{F} := \nabla \mathbf{y}$, is strictly positive for every image voxel. This guarantees a locally diffeomorphic transformation (subject to numerical accuracy). In [28,85], we determined the regularization parameter β_v based on a binary search algorithm. The search is constrained by the bounds on $J = \det \mathbf{F}$. That is, we choose β_v such that J is bounded from below by J_{min} and bounded from above by $1/J_{min}$, where $J_{min} \in (0, 1)$ is a user-defined parameter. The binary search is expensive because we solve the inverse problem repeatedly. For each trial β_v , we iterate until the convergence criteria for the Gauss–Newton–Krylov solver is met then use the previous velocity field as an initial guess for the next trial β_v .
- **β_w —regularization parameter for the divergence of the velocity field $w = \nabla \cdot v$.** The choice of β_w , along with β_v , is equally critical. Small values can result in extreme values of J and make the deformations locally non-diffeomorphic. As discussed above, in our previous work [28], we do parameter continuation in β_v and keep β_w fixed. This is sub-optimal for two reasons: (i) Both β_v and β_w depend on the resolution, so keeping β_w fixed for all resolutions can result in deformations with undesirable properties, and (ii) doing continuation in β_v alone does not ensure we get close enough to the set Jacobian bounds. Therefore, adding continuation in β_w , which also affects the Jacobian, is necessary.
- **J_{min} —lower bound for the determinant J of the deformation gradient.** The choice of this parameter is typically driven by dataset requirements, i.e., one has to decide how much volume change is acceptable. CLAIRE uses a default value of 0.25 [19]. Tighter bound on the Jacobian, i.e., J_{min} close to unity, will result in large β_v and β_w values leading to simple deformations and sub-par registration quality. Relaxing the Jacobian bound in combination with our continuation schemes for β_v and β_w can result in very small regularization parameters and extremely complex deformations.
- **n_t —number of time steps in the semi-Lagrangian scheme.** The semi-Lagrangian scheme is unconditionally stable and outperforms RK2 time integration schemes in terms of runtime for a given accuracy tolerance [30]. The choice of n_t is based on the adjoint error, which is the error measured after solving Equation (3) forward and then backward in time. In [30], we conducted detailed experiments for 2D image registration and found, that even for problems of clinical resolution $n_x = 256^2$, $n_t = 3$ (CFL = 10) did not cause issues in solver convergence. Increasing n_t beyond a certain value will introduce additional discretization errors from the interpolation scheme.

- Resolution of v .** We use the same spatial discretization for v as given for the input images. There exist image registration algorithms that approximate the registration deformation in a low-dimensional bandlimited space without sacrificing accuracy, resulting in dramatic savings in computational cost [15]. We have not explored this within the framework of CLAIRE. Note that [15] uses higher order regularization operators, which leads to smoother velocities compared to the ones CLAIRE produces, therefore enabling a representation on a coarser mesh. Moreover, CLAIRE uses a stationary velocity field, i.e., v is constant in time. In our previous work [28], we have demonstrated that stationary and time-varying velocity fields yield similar registration accuracy for registration between two real medical images of different subjects. More precisely, we did not observe any practically significant quantitative differences in registration accuracy for a varying number of coefficient fields in the case of time-varying velocity fields. Using a stationary velocity field is significantly cheaper and has a smaller memory overhead from a computational cost perspective.

3.4. Parameter Identification

Our algorithm to choose solver parameters proceeds as follows:

3.4.1. Resolution-Dependent Choice of the Interpolation Order and n_t

As our GPU implementation is only available in single precision (unlike the CPU implementation [19], which is available both in single and double precision), we use cubic interpolation (B-splines/Lagrange polynomials) with $n_t = 4$ ($n_t = 8$ for linear interpolation) for resolutions up to $n_x = 256^3$. For higher resolutions, we use linear interpolation to save computational cost and increase n_t proportionately to n_x to keep the CFL number fixed.

3.4.2. Parameter Search Scheme for β_v and β_w

We perform a two-stage search scheme:

- In the first part of the parameter search, we fix $\beta_w = \beta_{w,init}$ ($\beta_{w,init} = 1 \times 10^{-5}$) and search for β_v . The registration problem is first solved for a large value of $\beta_v = \beta_{v,init}$ so that we under-fit the data. In our experiments, we set $\beta_{v,init} = 1$. Subsequently, β_v is reduced by one order of magnitude in every continuation step and the registration problem is solved again with the new β_v . We repeat the reduction of β_v until we breach the Jacobian bounds $[J_{min}, 1/J_{min}]$. When this happens, we do a binary search for β_v between the last two values and terminate the binary search when the relative change in β_v is less than 10% of the previous valid β_v . In addition, we put a lower bound $\beta_{v,min} = 1 \times 10^{-5}$ on β_v . This lower bound is set purely to minimize computational cost. We denote the final value of β_v as β_v^* .
- In the second part of the search, we do a simple reduction search for β_w by fixing $\beta_v = \beta_v^*$. Starting with a given value $\beta_{w,init}$, we reduce β_w by one order of magnitude and repeat solving the registration problem with β_w^* and the respective value for β_w until we reach J_{min} . We put a lower bound $\beta_{w,min} = 1 \times 10^{-7}$ on β_w in order to minimize computational cost. We take the last valid value of β_w , for which the Jacobian determinant was within bounds and denote it as β_w^* . We fixed the value of $\beta_{w,init} = 1 \times 10^{-5}$ for all experiments and resolutions. We determined this value empirically by running image registration on a couple of image pairs at resolution $640 \times 880 \times 880$ and $160 \times 220 \times 220$ (see Section 5.4 for the images) for different values of $\beta_{w,init}$. We report these runs in Table A1 (see Appendix B).

We evaluate the parameter search scheme for real world brain images and report the performance in Section 5.2. Furthermore, we use it as the default parameter search scheme for all the experiments presented in this paper.

3.4.3. Parameter Continuation Scheme for β_v and β_w

If we want to use target β_v^* and β_w^* values for a new registration problem, we can perform a parameter *continuation* which is exactly like the parameter *search* except that we

neither perform the binary search for β_v nor check for the bounds on J . In the first stage of the *continuation*, we solve the registration problem for successively smaller values of β_v starting from $\beta_v = 1$ and reducing it by one order of magnitude until we reach $\beta_v = 1ek$ where $k = \lceil \log_{10}(\beta_v^*) \rceil$. Then we do an additional registration solve at $\beta_v = \beta_v^*$. We fix $\beta_w = \beta_{w,init}$ in the first stage. In the second stage, we fix $\beta_v = \beta_v^*$ and reduce β_w from $\beta_{w,init}$ to β_w^* in steps of one order of magnitude.

Whereas the expensive parameter search allows us to identify an optimal set of regularization parameters for unseen data, we use the parameter continuation scheme to speed up convergence. The combination of both is particularly efficient, for example in cohort studies, where we identify optimal regularization parameters for *one* image pair in the cohort and use the obtained parameters for all the other images.

4. Materials

We use publicly available image datasets for carrying out the image registration experiments in this paper (see Section 5). We summarize these datasets in Table 2. We discuss these datasets in detail.

Table 2. We list the image datasets we use in our scalable registration experiments (see Section 5). All the datasets are accessible publicly and further discussed in Section 4. We list the dataset name tag (which we use to refer to them throughout the rest of the paper), the imaging modality, the number of images, the spatial resolution and the image resolution in voxels. For datasets with an isotropic spatial resolution, we only provide a single value. For datasets with anisotropic spatial resolution, we list the resolution in all three dimensions. For the SYN dataset, spatial resolution does not carry a physical meaning, so we only list the image resolution.

Dataset	Image Modality	Number of Images	Spatial Resolution	Image Resolution
MUSE	T_1 -weighted MRI	5	1 mm	(256,256,256)
NIREP	T_1 -weighted MRI	16	1 mm	(256,300,256)
SYN	synthetic	4	–	(1024,1024,1024)
MRI250	T_1 -weighted MRI	1	250 μm	(640,880,880)
CLARITY	CLARITY-optimized light sheet microscopy	3	(4.68,4.68,5) μm	(2816,3016,1162)

4.1. MUSE

This dataset consists of five real brain T_1 -weighted MRIs of different individuals. These images were segmented into 149 functional brain regions in a semi-automated manner, including manual corrections by expert radiologists [86]. We visualize this data in Section 5.2. These images are part of a bigger set of template images that were used for the development of the MUSE [87] segmentation algorithm. The original image size is $256 \times 256 \times 256$ at a spatial resolution of 1 mm. This dataset is available for download through the neuromorphometrics website [86].

4.2. NIREP

Ref. [88] is a standardized repository for assessing registration accuracy that contains 16 T_1 -weighted MR neuroimaging datasets (na01–na16) of different individuals at an isotropic resolution of 1 mm. The original image size is $256 \times 300 \times 256$ voxels. We resample these images to an isotropic image size of $256 \times 256 \times 256$. We use the images na01–na10 for our experiments. This dataset is available for download through the GitHub link <https://github.com/andreasman/nirep> (accessed on 5 January 2022).

4.3. SYN

We create four sets of synthetic template and reference images to assess image registration accuracy as a function of resolution. We create a set of synthetic reference images m_1 by solving Equation (3) using a given synthetic template image m_0 and a synthetic velocity

field v . To construct the template image m_0 , we use a linear combination of high-frequency spherical harmonics. To be precise, we define the template image $m_0(\mathbf{x})$ as

$$m_0(\mathbf{x}) = \sum_{i=1}^{10} g_i(\mathbf{x}) \quad \text{with} \quad g_i(\mathbf{x}) = \begin{cases} 1, & \text{if } \|\mathbf{x} - \hat{\mathbf{x}}_i\|_2 \leq |Y_l^m(\theta + \hat{\theta}_i, \phi + \hat{\phi}_i)|, \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

and image coordinates $\mathbf{x} := (x, y, z) \in (-\pi, \pi]^3$. In Equation (10), Y_l^m represents spherical harmonics of the form

$$Y_l^m(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}} e^{im\theta} P_l^m(\cos(\phi)) \quad (11)$$

with parameters m, l , angular directions $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$, and associated Legendre functions P_l^m . We choose $m = 6, l = 8$ for our setup. $\hat{\theta}_i$ and $\hat{\phi}_i$ are random perturbations in integer multiples of $\pi/2$ and $\hat{\mathbf{x}}_i \in [-0.4\pi, 0.4\pi]^3$ is a random offset from the origin. The reference image $m_1(\mathbf{x})$ is generated by solving Equation (3) with initial condition $m_0(\mathbf{x})$ and velocity field $v(\mathbf{x}) := (v_x(\mathbf{x}), v_y(\mathbf{x}), v_z(\mathbf{x}))$, $\mathbf{x} = (x, y, z)$, defined as

$$\begin{aligned} v_x &= \sum_{k=1}^K \frac{1}{k^{0.5}} \cos(ky) \cos(kx), \\ v_y &= \sum_{k=1}^K \frac{1}{k^{0.5}} \sin(kz) \sin(ky), \\ v_z &= \sum_{k=1}^K \frac{1}{k^{0.5}} \cos(kx) \cos(kz) \end{aligned} \quad (12)$$

where $K = \{4, 8, 12, 16\}$. We set the template and the reference base image size to $n_x = n = (1024, 1024, 1024)$. It is important to note that m_0 and m_1 possess only the discrete intensities $i \in \{1, 2, \dots, 10\}$. This allow us to naturally define ten labels l_0^i and l_1^i , corresponding to m_0 and m_1 , respectively, for all image voxels with intensity i for each $i \in \{1, 2, \dots, 10\}$. We show a 2D slice of the template m_0 and reference m_1 images for the case $K = 4$ in Section 5.3. The scripts for generating the template image m_0 and the synthetic velocity field v can be found at <https://github.com/naveenaero/scala-claire> (accessed on 5 January 2022). The reference image m_1 can be generated using CLAIRÉ [21,34].

4.4. MRI250

Ref. [89] is an in-vivo 250 μm human brain MRI image which consists of a T_1 -weighted anatomical data acquired at an isotropic spatial resolution of 250 μm . The original image size is $640 \times 880 \times 880$ voxels. This image can be downloaded from [90]. We skull strip the dataset by downsampling it to $128 \times 128 \times 128$ using linear interpolation and then manually create the brain mask in ITK-SNAP [91]. We upsample this brain mask back to the original resolution and then apply it to the original image. We use the tool fast [92] from the FSL toolkit [93–95] to segment the T_1 -weighted MRI into gray matter (GM), white matter (WM), and cerebrospinal fluid (CSF) to be able to evaluate the registration performance using Dice score (see Equation (13)) between the image labels before and after registration.

4.5. CLARITY

We use the dataset from [27,96–98] which consists of 12 mouse brain images acquired using CLARITY-Optimized Light-sheet Microscopy (COLM). This dataset is available for download from [99]. These images have low contrast and are noisy. The in-plane resolution is $0.585 \mu\text{m} \times 0.585 \mu\text{m}$ and the cross-plane resolution is 5 to 8 μm . The images are stored at eight different resolution levels with level zero being the full resolution and level seven being the lowest resolution. We use the images at resolution levels three and six in our experiments. These levels correspond to an in-plane resolution of $4.68 \mu\text{m} \times 4.68 \mu\text{m}$ and

37.44 $\mu\text{m} \times 37.44 \mu\text{m}$, respectively, which translates to images of size $n = (2816, 3016)$ and $n/8 = (328, 412)$ voxels. The cross-plane resolution is constant at all levels and corresponds to 1162 voxels. We select Control182, Fear197, and Cocaine178 as the test images in our experiments.

5. Results and Discussion

We test the image registration on real-world (see Sections 5.4 and 5.5) and synthetic registration problems (see Section 5.3). The measures to analyze the registration performance are summarized in Section 5.1. We evaluate the parameter search scheme (see Section 3.4) on a set of real brain images and present the results in Section 5.2. Furthermore, we explore the following questions in the context of scalable image registration:

Question Q1: Do we need large-scale high-resolution image registration? Does the registration quality degrade when the registration is performed at a downsampled resolution when compared to performing registration at the original high resolution?

Question Q2: How does registration perform and scale for real, noisy, and high-resolution medical images of human and mouse brains?

5.1. Measures of Performance

In our experiments, we evaluate both runtime performance (in terms of solver wall clock time) and the registration quality in terms of accuracy. For the latter, we use the following metrics:

5.1.1. Dice Score Coefficient D

Let l_0 and l_1 be the binary label maps associated with the images m_0 and m_1 , respectively. Then, the Dice score D between the two label maps is given by

$$D(l_0, l_1) = \frac{2|l_0 \cap l_1|}{|l_0| + |l_1|}, \tag{13}$$

where $|\cdot|$ denotes the cardinality of a set, and \cap denotes the intersection of the two sets, respectively. We define $D(l_0, l_1)$ to be the Dice score pre-registration and $D(l(t = 1), l_1)$ post-registration, where $l(t = 1)$ is the label map that corresponds to the deformed template image $m(t = 1)$. Furthermore, for a set of discrete labels $l^i, i = \{1, 2, \dots, M\}$, where i corresponds to the label index, we define the volume fraction

$$\alpha^i = \frac{|l^i|}{\sum_{i=1}^M |l^i|}.$$

Using this definition, we compute the following statistics for the Dice coefficient: The Dice coefficient average D_a given by

$$D_a = \frac{1}{M} \sum_{i=1}^M D(l_0^i, l_1^i), \tag{14}$$

the volume weighted average of the Dice coefficient given by

$$D_{vw} = \frac{1}{\sum_{i=1}^M |l_1^i|} \sum_{i=1}^M |l_1^i| D(l_0^i, l_1^i), \tag{15}$$

and the inverse of the volume weighted average Dice coefficient given by

$$D_{ivw} = \frac{1}{\sum_{i=1}^M 1/|l_1^i|} \sum_{i=1}^M \frac{D(l_0^i, l_1^i)}{|l_1^i|} \tag{16}$$

Note that D_{vw} gives more weight to labels with higher volume fractions while D_{ivw} gives more weight to labels with smaller volume fractions.

5.1.2. Relative Residual r

This metric corresponds to the ratio of the image mismatch before and after the registration. It is given by

$$r = \frac{\|m(t=1) - m_1\|_2^2}{\|m_0 - m_1\|_2^2}. \quad (17)$$

5.1.3. Characteristic Parameters

For each image registration, we also report the regularization parameters and the obtained minimum and maximum values of the determinant of the deformation gradient $J := \det F$, i.e., the determinant of the Jacobian of the deformation map.

5.1.4. Visual Analysis

We visually support this quantitative analysis with snapshots of the registration results. The registration accuracy can be visually judged from the residual image, which corresponds to the absolute value of the pointwise difference between $m(t=1)$ and m_1 . The regularity of the deformations can be assessed from the pointwise maps of the determinant of the deformation gradient.

5.2. Experiment 1: Evaluation of the Parameter Search Scheme

We evaluate the parameter search scheme on a set of real brain images and compare the registration performance with a state-of-the-art SyN deformable registration tool in the ANTs toolkit.

5.2.1. Dataset

We use the MUSE dataset (see Section 4) for this experiment. After registration of the original T_1 -weighted images from this dataset, we use the image labels to evaluate the registration performance in terms of the volume weighted average Dice score D_{vw} .

5.2.2. Procedure

Out of the five T_1 images, we select `Template27` as the reference image m_1 and register the other four images to m_1 . For the registration, we use the parameter search scheme (see Section 3.4) to identify best regularization. We use linear interpolation and $n_t = 8$ time steps in the semi-Lagrangian solver. For the Jacobian bound, we select $J_{min} = 0.1$. In the parameter search, for each trial β_v and β_w , we drive the relative gradient norm $\|g\|_{2,rel} = \|g\|_2 / \|g_0\|_2$ to 1×10^{-2} . Once we have found adequate β_v and β_w for each image pair, we rerun the image registrations using only parameter continuation. For a baseline performance comparison, we also perform registration on the same image pairs using the SyN tool in ANTs [39]. For ANTs, we use the ‘‘MeanSquares’’ (i.e., squared L_2 -) distance measure. We run CLAIRE on a single NVIDIA V100 GPU with 16GB of memory on TACC’s Longhorn supercomputer. We run ANTs on a single node of the TACC Frontera supercomputer (system specs: Intel Xeon Platinum 8280 (‘‘Cascade Lake’’) processor with 56 cores on 2 sockets (base clock rate: 2.7 GHz)). We use all 56 cores. We report the parameters used for ANTs in Appendix A.

5.2.3. Results

We report the obtained estimates for β_v and β_w as well as results for registration quality in Table 3. In Figure 2, we provide a representative illustration of the obtained registration results. We report baseline registration performance using ANTs in Table 4. We compare the Dice scores obtained for CLAIRE and ANTs in Figure 3.

Table 3. Experiment 1: Performance of the parameter search scheme implemented in CLAIRE. We report results for the registration of four template images to the reference image `Template27`. We consider the squared L_2 -distance measure as image similarity metric. We restrict the Jacobian determinant $J \in [0.1, 10]$ for these registrations. We report the following quantities of interest: (i) optimal regularization parameters β_v^* and β_w^* , (ii) minimum J_{min} and maximum J_{max} Jacobian determinant achieved, (iii) solver wall clock time in seconds, and (iv) label volume weighted Dice average D_{vw} pre and post registration.

Template	β_v^*	β_w^*	J_{min}	J_{max}	D_{vw}		Runtime (s)	
					Pre	Post	Search	Continuation
4	7.75×10^{-5}	1.00×10^{-4}	4.53×10^{-1}	5.36×10^0	5.53×10^{-1}	6.99×10^{-1}	5.90×10^2	4.04×10^1
16	7.89×10^{-5}	1.00×10^{-5}	2.62×10^{-1}	4.23×10^0	5.51×10^{-1}	6.95×10^{-1}	4.39×10^2	5.82×10^1
22	1.14×10^{-5}	1.00×10^{-4}	1.19×10^{-1}	1.74×10^0	5.39×10^{-1}	7.04×10^{-1}	7.05×10^2	9.79×10^1
31	2.83×10^{-5}	1.00×10^{-4}	2.40×10^{-1}	1.86×10^0	5.27×10^{-1}	7.00×10^{-1}	6.19×10^2	6.07×10^1

Table 4. Experiment 1: Performance of ANTs. We report results for registration of four template images to the reference image `Template27` using a squared L_2 -distance metric. We report the following quantities of interest (i) minimum (J_{min}) and maximum (J_{max}) determinant of the deformation gradient obtained, (ii) label volume weighted Dice average D_{vw} pre and post registration, and (iii) solver wall clock time in seconds.

Template	J_{min}	J_{max}	D_{vw}		Runtime (s)
			Pre	Post	
4	1.40×10^{-1}	3.10×10^0	5.53×10^{-1}	6.86×10^{-1}	1.98×10^{-2}
16	2.50×10^{-1}	4.59×10^0	5.51×10^{-1}	6.87×10^{-1}	2.00×10^{-2}
22	3.11×10^{-1}	9.73×10^0	5.39×10^{-1}	6.62×10^{-1}	1.99×10^{-2}
31	2.07×10^{-1}	4.76×10^0	5.27×10^{-1}	6.85×10^{-1}	2.10×10^{-2}

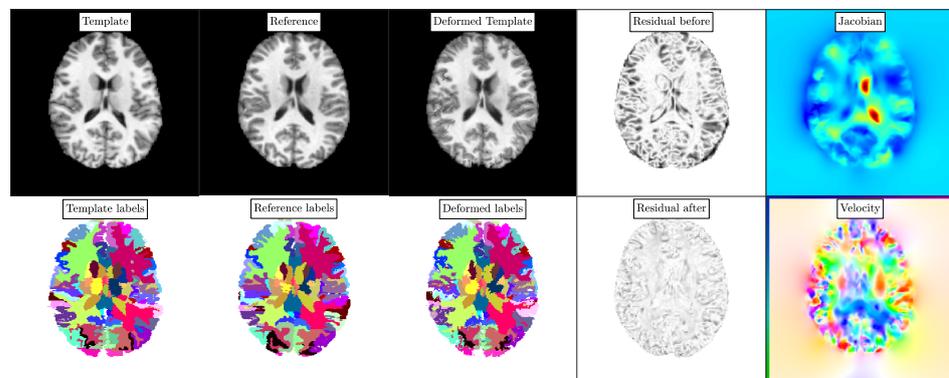


Figure 2. Experiment 1: Exemplary registration results using the parameter search scheme implemented in CLAIRE. We consider the datasets `Template16` (template image) and `Template27` (reference image). We refer to Table 3 and the text for details about the setup. We show (from left to right) the template, reference, deformed template image (**top row**), and their corresponding labels (**bottom row**). We also visualize the residual before and after the registration along with the determinant of the deformation gradient and an orientation map for the velocity field.

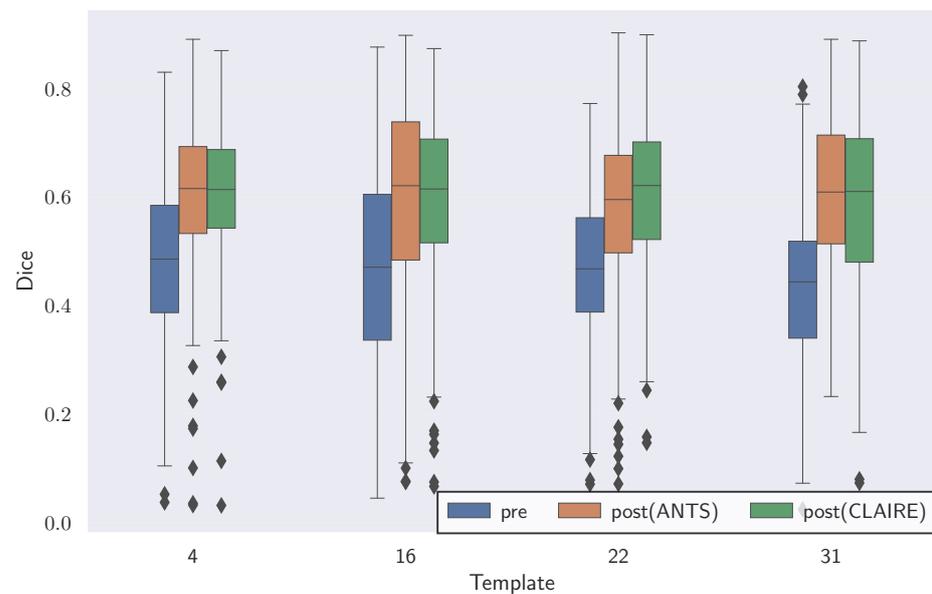


Figure 3. Experiment 1: Comparison of Dice scores for CLAIRE and ANTs. The box plots show Dice scores of the individual labels for the registration results reported for CLAIRE in Table 3 and ANTs in Table 4.

5.2.4. Observations

CLAIRE allows us to precisely control the properties of the deformation without having to tune any parameters manually. The only free parameters are the Jacobian bounds, which depend on the overall workflow related to the dataset. The volume weighted Dice scores D_{vw} obtained for CLAIRE (see Table 3) are competitive to those produced by ANTs (see Table 4). The average runtime for ANTs for all the registrations reported in Table 4 is 201 s (≈ 3 min). For CLAIRE, the average wall clock time of CLAIRE in the parameter search mode is 9.8 min ($3\times$ slower than ANTs; we search for adequate regularization parameters), while, in the continuation mode, the runtime of CLAIRE is 64 s ($3\times$ faster than ANTs; we apply the optimal regularization parameter and do not search for them).

5.3. Experiment 2A: High Resolution Synthetic Data Registration

In this experiment, we answer **Q1**. We attempt this by executing our registration algorithm on synthetic imaging data. The advantages of using such images over real datasets are as follows:

- They are noise-free, high contrast, and sharp, unlike real-world images.
- There is a scarcity of high resolution real image data because it is expensive and time-consuming to acquire. We can control the resolution of synthetic data because the images are created using analytically known functions.
- We can control the number of discrete image intensity levels, i.e., labels. Because these labels are available as ground truth, we can use them to precisely quantify registration accuracy through the Dice coefficient, avoiding inter- and intra-observer variabilities and other issues associated with establishing ground truth labels in real imaging data.

By performing image registration at different resolutions (and applying the resulting velocity to transform the high resolution original images), we want to check whether the registration at higher resolutions is more accurate than performing the registration at a lower resolution.

5.3.1. Dataset

We use the SYN dataset (see Section 4) for this experiment.

5.3.2. Procedure

We execute registration at different resolutions for the original resolution images and quantify the accuracy using the Dice coefficient for labels before and after the registration. We compare the Dice statistics for different resolutions. More specifically, we take the following steps:

1. We register the template image m_0 to the reference image m_1 at the base resolution n to get the velocity field v_n . We transport m_0 using the velocity v_n to get the deformed template image $m(t=1)$ by solving Equation (3). Then, we compute the Dice score between $l^i(t=1)$ and l^i_1 , $i \in 1, \dots, 10$ which are discrete labels for $m(t=1)$ and m_1 , respectively, using Equation (13).
2. We downsample m_0 and m_1 using nearest neighbor interpolation to half the base resolution (for example, $n/2 = (512, 512, 512)$). Notice that we treat $n_x = (N_1, N_2, N_3)$ as a tuple. When we say $n_x/2$, we mean $n_x/2 = (N_1/2, N_2/2, N_3/2)$ and register the downsampled images to get the velocity $\hat{v}_{n/2}$. We upsample $\hat{v}_{n/2}$ to the base resolution n using spectral prolongation and call it $v_{n/2}$. We transport m_0 using $v_{n/2}$ by solving Equation (3) to get the deformed template image $m(t=1)$ and then compute the Dice score for this new deformed template image.
3. We repeat the procedure in step 2 for resolutions $n/4$ and $n/8$ and compute the corresponding Dice scores.

For the registration, we fix the determinant J of the deformation gradient to be within $[5 \times 10^{-2}, 20]$ and search for the regularization parameters using the proposed parameter search scheme as described above in Section 2. Note that we perform a search for an optimal regularization parameter for each individual dataset because *we want to obtain the best result for each pair of images*. In practical applications, this is not necessary (see comments below; we also refer to [19] for a discussion). We fix the tolerance for the reduction of the gradient to 5×10^{-2} , which we have found to be sufficiently accurate for most image registration problems (see [19]). We use linear interpolation in the semi-Lagrangian scheme. Another hyperparameter in our registration solver is the number of time steps n_t for the semi-Lagrangian (SL) scheme. We consider two cases for selecting n_t :

1. **n_t changes with resolution:** We use $n_t = 4$ time steps for the coarsest resolution $n_x = n/8$ and double n_t when we double the resolution in order to keep the CFL number fixed. All other solver parameters, except for the regularization parameters, are the same at each resolution.
2. **n_t fixed with resolution:** In order to study the effect of n_t on the Dice score we keep n_t fixed for each n_x , instead of increasing n_t proportionately to n_x .

5.3.3. Results

In Figure 4, we visualize the template, reference and deformed template images for the synthetic problem constructed with $K = 4$. We report quantitative results for CLAIRE in Tables 5 and 6, respectively. In Figure 5, we compare the Dice score for individual labels as a function of their volume fraction α . In Figure 6, we visualize box plots of the Dice score for the registrations reported in Table 5.

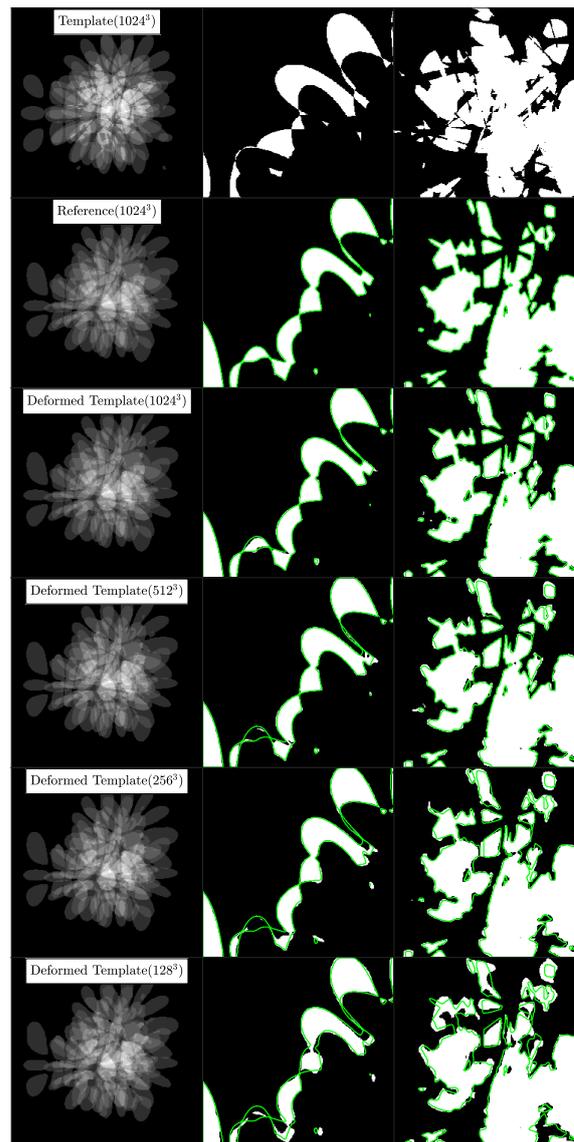


Figure 4. Experiment 2A: Visualization of registration results for case 1. In column 1, from top to bottom, we visualize the template, reference and deformed template images for registrations done at different resolutions. These images correspond to the runs #1-4 in Table 5. The value in the parentheses in column 1 indicates the resolution at which registration was done. The visualization is done at the original resolution $n = (1024, 1024, 1024)$. In columns 2 and 3, we visualize cropped portions of the images shown in column 1 for specific label values. In column 2, we show label 1, in column 3, we show the union of labels with intensity value ≥ 5 . Note that higher label values have smaller volumes and more fine-grained features. We plot the label boundaries for the reference image in green to visualize the registration errors.

Table 5. Experiment 2A: Registration performance for CLAIRE for case 1 (n_t changes proportionally to the image resolution, see Section 5.3). Comparison of registration accuracy based on the Dice score at different resolutions for the synthetic dataset SYN. K denotes the frequency of the synthetic velocity field in Equation (12). $n = (1024, 1024, 1024)$ is the base image resolution. We fix the tolerance for the reduction of the gradient to 5×10^{-2} and use linear interpolation. The Jacobian bounds for the parameter search are $[0.05, 20]$. We report β_v^* and β_w^* (the optimal regularization parameters obtained with the proposed parameter search scheme), and J_{min} and J_{max} (the minimum and maximum values for the determinant of the deformation gradient). For the Dice score, we report average Dice (D_a), the volume weighted average Dice (D_{vw}), and the inverse volume weighted average Dice (D_{iww}), pre and post registration. We also report the wall clock time for the parameter search.

Run	K	n_x	n_t	β_v^*	β_w^*	J_{min}	J_{max}	D_a		D_{vw}		D_{iww}		Runtime (s) Search
								Pre	Post	Pre	Post	Pre	Post	
#1	4	n	32	1.1×10^{-5}	1.0×10^{-7}	1.7×10^{-1}	7.4×10^0	3.1×10^{-1}	9.2×10^{-1}	5.8×10^{-1}	9.8×10^{-1}	3.9×10^{-2}	8.5×10^{-1}	2.9×10^3
#2		$n/2$	16	1.1×10^{-5}	1.0×10^{-7}	1.9×10^{-1}	7.7×10^0		8.7×10^{-1}		9.7×10^{-1}		7.0×10^{-1}	6.5×10^2
#3		$n/4$	8	1.1×10^{-5}	1.0×10^{-7}	2.6×10^{-1}	1.4×10^1		7.9×10^{-1}		9.5×10^{-1}		5.0×10^{-1}	1.1×10^2
#4		$n/8$	4	1.1×10^{-5}	1.0×10^{-6}	4.7×10^{-1}	5.6×10^0		6.7×10^{-1}		9.1×10^{-1}		1.8×10^{-1}	1.5×10^1
#5	8	n	32	1.1×10^{-5}	1.0×10^{-7}	5.1×10^{-2}	1.0×10^1	3.2×10^{-1}	9.0×10^{-1}	5.3×10^{-1}	9.8×10^{-1}	7.4×10^{-2}	7.6×10^{-1}	2.7×10^3
#6		$n/2$	16	1.1×10^{-5}	1.0×10^{-7}	1.8×10^{-1}	1.5×10^1		8.5×10^{-1}		9.7×10^{-1}		6.0×10^{-1}	6.2×10^2
#7		$n/4$	8	1.1×10^{-5}	1.0×10^{-6}	3.0×10^{-1}	7.8×10^0		7.6×10^{-1}		9.4×10^{-1}		4.1×10^{-1}	1.0×10^2
#8		$n/8$	4	2.4×10^{-5}	1.0×10^{-6}	3.8×10^{-1}	4.8×10^0		6.4×10^{-1}		9.0×10^{-1}		1.7×10^{-1}	1.4×10^1
#9	12	n	32	1.1×10^{-5}	1.0×10^{-7}	1.7×10^{-1}	1.2×10^1	3.1×10^{-1}	9.2×10^{-1}	5.2×10^{-1}	9.8×10^{-1}	9.5×10^{-2}	8.5×10^{-1}	2.6×10^3
#10		$n/2$	16	1.1×10^{-5}	1.0×10^{-6}	3.1×10^{-1}	8.9×10^0		8.6×10^{-1}		9.7×10^{-1}		7.4×10^{-1}	5.4×10^2
#11		$n/4$	8	1.1×10^{-5}	1.0×10^{-6}	2.9×10^{-1}	1.2×10^1		7.5×10^{-1}		9.4×10^{-1}		4.5×10^{-1}	9.4×10^1
#12		$n/8$	4	1.1×10^{-5}	1.0×10^{-6}	4.1×10^{-1}	9.9×10^0		6.0×10^{-1}		8.9×10^{-1}		1.9×10^{-1}	1.4×10^1
#13	16	n	32	1.1×10^{-5}	1.0×10^{-7}	1.6×10^{-1}	9.5×10^0	2.9×10^{-1}	9.1×10^{-1}	5.1×10^{-1}	9.8×10^{-1}	9.0×10^{-2}	8.1×10^{-1}	2.4×10^3
#14		$n/2$	16	1.1×10^{-5}	1.0×10^{-7}	1.7×10^{-1}	1.4×10^1		8.4×10^{-1}		9.7×10^{-1}		6.0×10^{-1}	5.2×10^2
#15		$n/4$	8	1.4×10^{-5}	1.0×10^{-6}	3.0×10^{-1}	8.8×10^0		7.4×10^{-1}		9.4×10^{-1}		4.7×10^{-1}	9.5×10^1
#16		$n/8$	4	2.7×10^{-5}	1.0×10^{-6}	3.9×10^{-1}	1.5×10^1		6.1×10^{-1}		9.0×10^{-1}		2.0×10^{-1}	1.5×10^1

Table 6. Experiment 2A: Registration performance for CLAIRE for case 2 (n_t independent of the image resolution). Comparison of registration accuracy using Dice at different resolutions for the synthetic dataset SYN. K denotes the frequency of the synthetic velocity field in Equation (12). $n = (1024, 1024, 1024)$ is the base image resolution. We fix the tolerance for the reduction of the gradient to 5×10^{-2} and use linear interpolation. The Jacobian bounds for parameter search is $[0.05, 20]$. For each value of n_t , we report results for different resolutions. We report β_v^* and β_w^* (the optimal regularization parameters obtained with the proposed parameter search scheme), and J_{min} and J_{max} (the minimum and maximum values for the determinant of the deformation gradient). For the Dice score, we report average Dice (D_a), the volume weighted average Dice (D_{vw}), and the inverse volume weighted average Dice (D_{iww}), pre and post the registration. We also report the wall clock time for the parameter search. The missing cases for $K = 8$ failed to finish in a reasonable time frame. We only report a couple of cases for $K = 16$ and expect a behavior similar to $K = 8$ for the rest.

Run	K	n_x	n_t	β_v^*	β_w^*	J_{min}	J_{max}	D_a		D_{vw}		D_{iww}		Runtime (s) Search		
								Pre	Post	Pre	Post	Pre	Post			
#1	8	4	$n/2$	1.4×10^{-5}	1.0×10^{-7}	9.7×10^{-2}	1.1×10^1									
#2			$n/4$	1.1×10^{-5}	1.0×10^{-7}	3.8×10^{-1}	3.8×10^0			8.8×10^{-1}	9.8×10^{-1}	7.4×10^{-1}	3.9×10^2			
#3			$n/8$	2.4×10^{-5}	1.0×10^{-6}	3.8×10^{-1}	4.8×10^0			6.8×10^{-1}	9.2×10^{-1}	2.1×10^{-1}	7.9×10^2			
#4		8	8	$n/4$	1.1×10^{-5}	1.0×10^{-6}	2.9×10^{-1}	7.7×10^0								
#5				$n/8$	1.7×10^{-5}	1.0×10^{-6}	3.9×10^{-1}	6.4×10^0			6.2×10^{-1}	8.9×10^{-1}	1.6×10^{-1}	1.4×10^1		
#6				$n/2$	1.1×10^{-5}	1.0×10^{-7}	1.8×10^{-1}	1.4×10^1			7.5×10^{-1}	9.4×10^{-1}	4.1×10^{-1}	1.0×10^2		
#7		16	16	$n/4$	1.1×10^{-5}	1.0×10^{-6}	3.1×10^{-1}	8.2×10^0								
#8				$n/8$	1.1×10^{-5}	1.0×10^{-7}	5.4×10^{-1}	3.2×10^0			3.2×10^{-1}	8.3×10^{-1}	5.3×10^{-1}	7.4×10^{-2}	5.2×10^{-1}	5.9×10^2
#9				n	1.1×10^{-5}	1.0×10^{-7}	5.1×10^{-2}	1.0×10^1			7.1×10^{-1}	9.2×10^{-1}	3.4×10^{-1}	1.2×10^2		
#10		32	32	$n/2$	1.1×10^{-5}	1.0×10^{-7}	1.2×10^{-1}	1.9×10^1								
#11				$n/4$	1.1×10^{-5}	1.0×10^{-6}	3.1×10^{-1}	1.0×10^1			5.6×10^{-1}	8.5×10^{-1}	1.4×10^{-1}	6.7×10^1		
#12				$n/8$	1.1×10^{-5}	1.0×10^{-7}	5.2×10^{-1}	3.2×10^0			9.0×10^{-1}	9.8×10^{-1}	7.6×10^{-1}	2.7×10^3		
#13	16	4	$n/2$	1.3×10^{-5}	1.0×10^{-6}	2.0×10^{-1}	6.9×10^0									
#14		32	n	1.1×10^{-5}	1.0×10^{-7}	1.6×10^{-1}	9.5×10^0			2.9×10^{-1}	8.6×10^{-1}	5.1×10^{-1}	9.0×10^{-2}	7.8×10^{-1}	3.7×10^2	

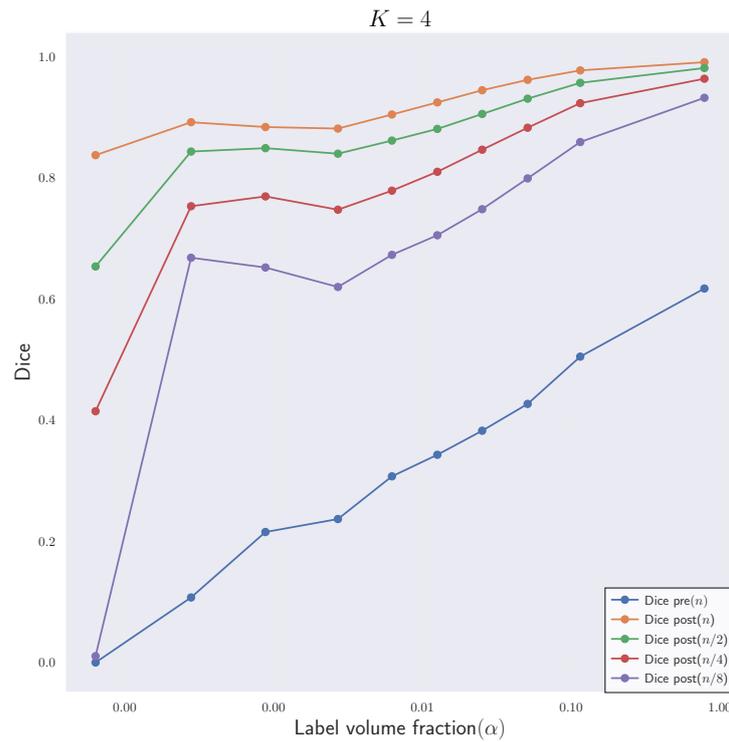


Figure 5. Experiment 2A: Quantitative results for the registration results corresponding to case 1. We show a plot of the Dice scores against the label volume fraction α for each label $l^i, i = 1, \dots, 10$ for the registration of the synthetic data set SYN at different resolutions. This figure corresponds to the registration runs #1-4 in Table 5 for $K = 4$.

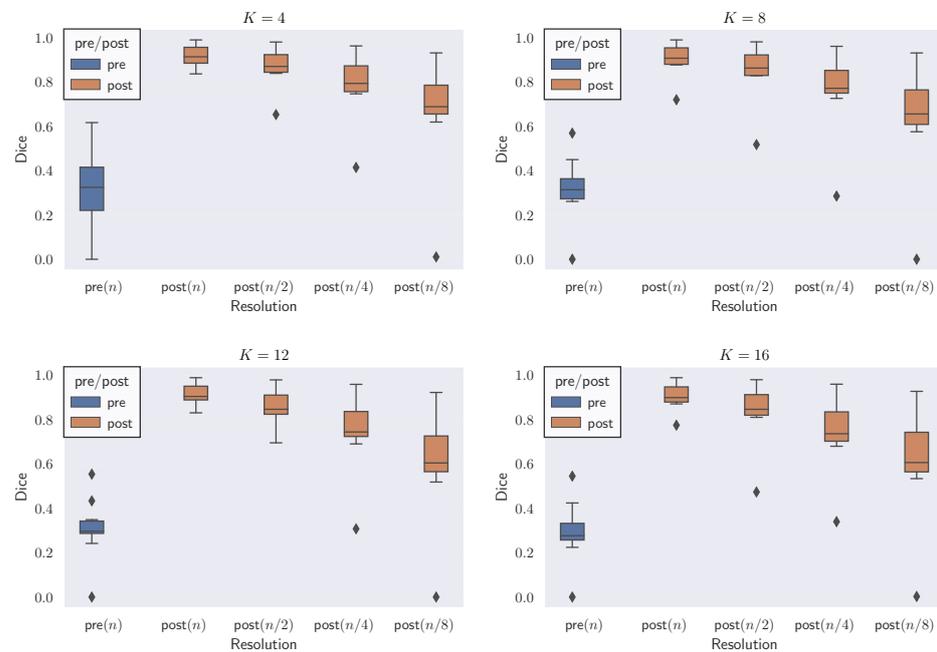


Figure 6. Experiment 2A: Quantitative results for the registration results corresponding to case 1. We show box plots of the Dice scores for the individual labels before and after registration for different resolutions. We consider the synthetic test problem SYN. This figure corresponds to the registration results reported in Table 5.

5.3.4. Observations

The most important observations are: (i) The Dice score averages are better for registrations performed at the base resolution n with progressively worse Dice scores for registrations done at coarser resolutions; (ii) the difference between Dice scores for registrations done at successively coarser resolutions for $K = 16$ (rougher velocity field) is higher than at $K = 4$ (smoother velocity field); (iii) keeping n_t fixed for the base and coarser resolutions does not affect the Dice score trend, i.e., the Dice decreases as n_x is decreased. In the following, we give more details for these general observations.

Regarding Dice score averages in Table 5, we observe that D_a , the arithmetic mean of the Dice scores of individual labels, drops by as much as 7% between run #13 and #14. However, the percentage drop in volume weighted Dice average D_{vw} is smaller than in D_a . This indicates that labels with higher volume are still easier to register at coarser resolutions. The inverse weighted Dice average D_{iww} , which gives more weight to smaller labels, features a more pronounced decrease because smaller regions contribute to the high frequency content in the image; this information is lost when the images are downsampled. We observe a 21.8% difference in D_{iww} for the high frequency images in run #13 and #14 for $K = 16$. As we increase the frequency K of the synthetic ground truth velocity, we see that the difference in all Dice score averages between successive resolutions increases. As K increases, we get increasingly rougher velocity fields, which we can not recover by registering the original images at coarser resolutions.

In Table 6, the Dice scores behave the same way even when n_t is fixed for different n_x , indicating that the loss in accuracy is primarily because of the reduction in the spatial resolution (and not the temporal resolution). We also observe that for the full resolution of $n_x = n$, using $n_t < 32$ results in slow solver convergence; the run did not finish in under 2 h. We attribute this slow convergence rate to the loss in numerical accuracy in the computation of the reduced gradient in Equation (6). If we compare run #1 and #9 in Table 6, we see that the difference in D_a is marginal in comparison to the run time cost overhead for run #9. However, the accuracy difference increases as K is increased, and the images get less smooth (see runs #13 and #14).

These quantitative observations are confirmed by the visual analysis in the figures shown: From Figure 4, we observe that at lower resolutions (top to bottom), the alignment of the outlines (green lines; reference image) with the structures (white areas; deformed template image) is less accurate. Figure 5: shows that the Dice score is worse for labels with smaller volume fractions, i.e., fine structures are matched less accurately at coarse resolutions. Looking at Figure 6, we observe that the average registration accuracy decreases as we decrease the resolution.

We use 32 GPUs for registration at $n_x = (1024, 1024, 1024)$, 4 GPUs for $n_x = (512, 512, 512)$, and a single GPU for $n_x = (256, 256, 256)$ and $n_x = (128, 128, 128)$. Registration for $n_x = (1024, 1024, 1024)$ takes on average 44 min wall-clock time. It is important to note that this includes the time spent in the search for optimal regularization parameters (i.e., we solve the inverse problem multiple times using warm starts; see Section 3.4 for details regarding the scheme). For the large-scale runs that use multiple GPUs, the overall runtime of the solver is dominated by communication between MPI processes [20]. Adding more resources does not necessarily reduce the runtime because of this increase in communication cost. Registrations for $n_x = (512, 512, 1512)$ and lower resolutions are much quicker and run in the order of 10 min or less. In the present work, we perform the parameter search for each individual case because *we want to obtain the best result for each pair of images*. However, in practice where a medical imaging pipeline requires registrations for several similar images, we suggest running the parameter search scheme on one pair of images and use the obtained regularization parameters to run the cohort registration for all images, as we have done in our previous work [19]. This strategy reduces the computational cost drastically. One downside to this strategy is that some images in the cohort will not be registered as accurately as others.

Our experiment with synthetic images suggests that Dice scores are better when registrations are done in the original high resolution at which the labels were created. Registration accuracy is affected more significantly if high frequency velocity fields are considered. The images used in this experiment are synthetic and free of noise. We use these images for both registration and evaluation of performance using Dice scores. Because the ground truth labels for these images at the highest resolution are known with certainty, we have high confidence in our observations regarding registration accuracy: the Dice scores become worse when registration is conducted at lower resolutions. However, in practical applications, images have noise and low contrast. To evaluate the registration accuracy for real images using Dice scores, we first evaluate their segmentation using external segmentation tools. This segmentation step is prone to errors (not only due to noise and a lack of contrast but also due to inherent limitations in segmentation software themselves). These errors result in a misalignment between the structures present in the original image and its segmentation, which complicates our analysis. Having said this, we conduct experiments on real brain MRIs in the next section to explore if we can provide experimental evidence that at least partially confirms the observations we have made in this section.

5.4. Experiment 2B: High Resolution Real Data Registration

In this experiment, we aim at answering **Q1** as well as **Q2**. We do this by registering real human brain MRI datasets instead of synthetic images. Unlike synthetic images, these images are not noise-free. Moreover, they lack high contrast.

5.4.1. Datasets

We use the NIREP and the MRI250 image datasets (see Section 4) for this experiment.

5.4.2. Procedure

We designate the MRI250 image as the template image m_0 . We generate the reference images m_1 from the images na01–na10 from the NIREP dataset since we do not have access to other T_1 -weighted MRI from a different subject at the original resolution of 250 μm . The acquired spatial resolution of the NIREP data is 1 mm, which is $4\times$ larger than 250 μm . Therefore, in order to generate a reference image m_1 that are 250 μm in spatial resolution, we take the following steps:

1. Upsample the respective NIREP image from $256 \times 300 \times 256$ to $640 \times 880 \times 880$ using linear interpolation.
2. Register MRI250 to the upsampled NIREP image using CLAIRE and transport m_0 (which corresponds to the MRI250 image) using the resulting velocity v and solving Equation (3) to obtain the deformed template image $m_1 = m(t = 1)$. We set the tolerance for the relative gradient norm to $g_{tol} = 1 \times 10^{-2}$. We lower the tolerance compared to other runs to obtain a potentially more accurate registration result. We use the default regularization parameters $\beta_v = 1 \times 10^{-2}$ and $\beta_w = 1 \times 10^{-4}$. Consequently, we do not perform a parameter search to estimate an optimal regularization parameter for this registration. We want to keep the downstream registration performance analysis, where we will use parameter search, oblivious to the process of generating the high-resolution reference image.

To generate a segmentation that we can use to compute Dice scores (not for the registration itself, which is done on the original unsegmented images), we use the tool fast from FSL [95] both on the template image m_0 and on the reference image m_1 . We generate labels *WM*, *GM*, and *CSF*. The remaining steps for this experiment are the same as described in experiment 2A in Section 5.3 except that here we are registering real T_1 -weighted images instead of noise-free synthetic images. The base resolution for this experiment is $n_x = n = (640, 880, 880)$. We consider $n_x = n/2$ and $n_x = n/4$ for the downsampled resolutions. We also consider the two sub-cases for selecting n_t as we did

in Section 5.3. For the case where n_t changes with resolution, we use $n_t = 4$ for $n_x = n/4$, $n_t = 8$ for $n_x = n/2$ and $n_t = 16$ for $n_x = n$.

5.4.3. Results

We report the solver parameters for our registration with CLAIRE along with the relative residual r and Dice score averages for GM , WM , and CSF before and after the registration in Table 7. The relative residual r and the Dice score are always computed at the base resolution $n = (640, 880, 880)$. The respective results with n_t fixed independent of the resolution are given in Table 8 for na01. We visualize the image registration results for the reference image na01 in Figure 7.

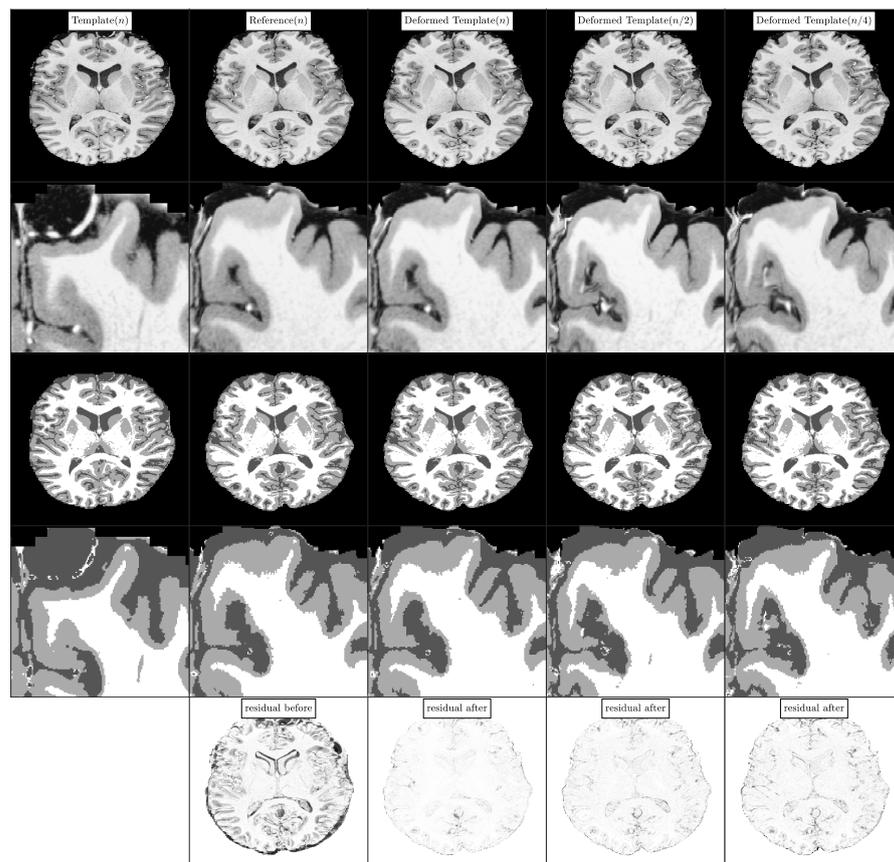


Figure 7. Experiment 2B: Illustration of registration results for the multi-resolution registration experiment on real brain images. The images shown here correspond to the runs #1, #2, and #3 in Table 7). The base resolution is $n_x = n = (640, 880, 880)$. In row 1, from left to right, we show the T_1 -weighted MRI250 datasets (template image m_0), the upsampled na01 dataset (reference image m_1) from the NIREP data repository, and the deformed template images obtained from registration at resolutions n_x , $n_x/2$, and $n_x/4$, respectively. In row 2, we show a cropped portion of the images from row 1. In rows 3 and 4, we show the label maps consisting of white matter (WM; white), gray matter (GM; light gray), and cerebro-spinal fluid (CSF; dark gray) and their cropped versions, respectively. In row 5, we show the image residuals before and after registration with respect to each resolution level.

Table 7. Experiment 2B: Registration performance for CLAIRE for case 1 (n_t changes proportional to the image resolution). Comparison of registration accuracy using Dice and relative residual r at different resolutions for the registration of the MRI250 brain image to templates generated from ten *real* MRI scans from the NIREP dataset. We consider three resolution levels $n_x = \{n, n/2, n/4\}$ where $n = (640, 880, 880)$. We fix the tolerance for the relative gradient to 5×10^{-2} . We use linear interpolation in the semi Lagrangian scheme. The bounds for the determinant of the deformation gradient for the parameter search are $[0.05, 20]$. We report the regularization parameters β_v^* and β_w^* obtained through the proposed parameter search scheme, the minimum and maximum determinant of the deformation gradient (J_{min} and J_{max}), the relative residual (r), the average Dice (D_a), pre and post the registration, as well as the wall clock time for the parameter search.

Run	NIREP	n_x	n_t	β_v^*	β_w^*	J_{min}	J_{max}	r	Pre	D_a	Post	Runtime (s) Search
#1	na01	n	16	1.1×10^{-3}	1.0×10^{-5}	1.8×10^{-1}	8.3×10^0	2.5×10^{-1}	5.5×10^{-1}		9.0×10^{-1}	3.1×10^2
#2		$n/2$	8	1.1×10^{-5}	1.0×10^{-6}	1.8×10^{-1}	6.5×10^0	3.5×10^{-1}			8.5×10^{-1}	3.4×10^2
#3		$n/4$	4	1.1×10^{-5}	1.0×10^{-5}	2.3×10^{-1}	8.2×10^0	4.5×10^{-1}			8.0×10^{-1}	3.6×10^1
#4	na02	n	16	1.1×10^{-3}	1.0×10^{-5}	7.8×10^{-2}	6.3×10^0	2.5×10^{-1}	5.4×10^{-1}		8.9×10^{-1}	3.3×10^2
#5		$n/2$	8	1.1×10^{-5}	1.0×10^{-6}	1.2×10^{-1}	4.3×10^0	3.6×10^{-1}			8.3×10^{-1}	3.0×10^2
#6		$n/4$	4	1.1×10^{-5}	1.0×10^{-6}	8.1×10^{-2}	1.1×10^1	4.6×10^{-1}			7.7×10^{-1}	4.0×10^1
#7	na03	n	16	1.1×10^{-5}	1.0×10^{-7}	1.1×10^{-1}	6.1×10^0	3.3×10^{-1}	5.1×10^{-1}		8.4×10^{-1}	3.2×10^3
#8		$n/2$	8	1.1×10^{-5}	1.0×10^{-6}	1.1×10^{-1}	1.8×10^1	3.9×10^{-1}			8.0×10^{-1}	2.9×10^2
#9		$n/4$	4	1.1×10^{-5}	1.0×10^{-7}	1.0×10^{-1}	1.7×10^1	4.7×10^{-1}			7.6×10^{-1}	4.2×10^1
#10	na04	n	16	3.1×10^{-2}	1.0×10^{-5}	1.2×10^{-1}	1.4×10^1	3.7×10^{-1}	5.3×10^{-1}		8.0×10^{-1}	1.9×10^2
#11		$n/2$	8	1.1×10^{-3}	1.0×10^{-5}	6.8×10^{-2}	8.9×10^0	2.9×10^{-1}			8.7×10^{-1}	5.1×10^1
#12		$n/4$	4	1.1×10^{-5}	1.0×10^{-5}	1.0×10^{-1}	6.8×10^0	4.6×10^{-1}			7.6×10^{-1}	3.7×10^1
#13	na05	n	16	1.1×10^{-5}	1.0×10^{-5}	9.5×10^{-2}	8.9×10^0	3.2×10^{-1}	5.3×10^{-1}		8.5×10^{-1}	2.8×10^3
#14		$n/2$	8	1.1×10^{-5}	1.0×10^{-5}	1.6×10^{-1}	1.1×10^1	3.6×10^{-1}			8.3×10^{-1}	2.5×10^2
#15		$n/4$	4	1.1×10^{-5}	1.0×10^{-5}	1.6×10^{-1}	1.6×10^1	4.5×10^{-1}			7.8×10^{-1}	3.7×10^1
#16	na06	n	16	1.1×10^{-3}	1.0×10^{-5}	7.8×10^{-2}	1.4×10^1	2.5×10^{-1}	5.3×10^{-1}		8.9×10^{-1}	3.3×10^2
#17		$n/2$	8	1.1×10^{-5}	1.0×10^{-6}	2.0×10^{-1}	5.6×10^0	3.5×10^{-1}			8.3×10^{-1}	3.0×10^2
#18		$n/4$	4	1.1×10^{-5}	1.0×10^{-5}	1.6×10^{-1}	7.2×10^0	4.4×10^{-1}			7.7×10^{-1}	3.6×10^1
#19	na07	n	16	1.0×10^{-2}	1.0×10^{-5}	9.5×10^{-2}	2.0×10^1	3.0×10^{-1}	5.3×10^{-1}		8.6×10^{-1}	2.4×10^2
#20		$n/2$	8	1.1×10^{-5}	1.0×10^{-5}	1.6×10^{-1}	1.6×10^1	3.5×10^{-1}			8.4×10^{-1}	3.9×10^2
#21		$n/4$	4	1.1×10^{-5}	1.0×10^{-5}	1.7×10^{-1}	1.7×10^1	4.5×10^{-1}			7.7×10^{-1}	3.7×10^1
#22	na08	n	16	1.1×10^{-5}	1.0×10^{-7}	1.3×10^{-1}	4.8×10^0	3.1×10^{-1}	5.3×10^{-1}		8.6×10^{-1}	2.5×10^3
#23		$n/2$	8	1.1×10^{-5}	1.0×10^{-6}	1.0×10^{-1}	1.3×10^1	3.8×10^{-1}			8.1×10^{-1}	3.0×10^2
#24		$n/4$	4	1.1×10^{-5}	1.0×10^{-6}	9.4×10^{-2}	1.7×10^1	4.7×10^{-1}			7.5×10^{-1}	4.2×10^1
#25	na09	n	16	1.1×10^{-3}	1.0×10^{-5}	6.3×10^{-2}	1.5×10^1	2.5×10^{-1}	5.3×10^{-1}		8.9×10^{-1}	3.5×10^2
#26		$n/2$	8	1.1×10^{-5}	1.0×10^{-5}	1.2×10^{-1}	5.1×10^0	3.5×10^{-1}			8.3×10^{-1}	2.3×10^2
#27		$n/4$	4	1.1×10^{-5}	1.0×10^{-6}	9.9×10^{-2}	7.4×10^0	4.5×10^{-1}			7.6×10^{-1}	4.2×10^1
#28	na10	n	16	1.1×10^{-5}	1.0×10^{-7}	1.1×10^{-1}	5.7×10^0	3.2×10^{-1}	5.4×10^{-1}		8.5×10^{-1}	2.6×10^3
#29		$n/2$	8	1.1×10^{-5}	1.0×10^{-5}	1.2×10^{-1}	4.5×10^0	3.5×10^{-1}			8.3×10^{-1}	2.7×10^2
#30		$n/4$	4	1.1×10^{-5}	1.0×10^{-6}	1.0×10^{-1}	9.1×10^0	4.7×10^{-1}			7.6×10^{-1}	4.1×10^1

Table 8. Experiment 2B: Registration performance for CLAIRE for case 2 (n_t independent of the image resolution). Comparison of registration accuracy using Dice and relative residual r for a fixed number of time steps n_t at different resolutions for the registration of the real MRI datasets MRI250 and the reference image m_1 generated from na01 from the NIREP repository. We consider three resolution levels $n_x = \{n, n/2, n/4\}$ where $n = (640, 880, 880)$. We fix the tolerance for the relative gradient to 5×10^{-2} . We use linear interpolation in the semi Lagrangian schreme. The bounds for the determinant of the deformation gradient for the parameter search are $[0.05, 20]$. We keep the time step n_t fixed. We report the regularization parameters β_v^* and β_w^* obtained through the proposed parameter search scheme, the minimum and maximum determinant of the deformation gradient (J_{min} and J_{max}), the relative residual (r), the average Dice (D_a), pre and post the registration, as well as the wall clock time for the parameter search. The case with $n_x = n$ and $n_t = 4$ failed to finish in under 4 h.

Run	NIREP	n_x	n_t	β_v^*	β_w^*	J_{min}	J_{max}	r	D_a		Runtime (s) Search
									Pre	Post	
#1	na01	4	$n/2$	1.1×10^{-5}	1.0×10^{-6}	9.2×10^{-2}	5.2×10^0	3.1×10^{-1}	5.5×10^{-1}	8.7×10^{-1}	2.7×10^2
#2			$n/4$	1.1×10^{-5}	1.0×10^{-5}	2.3×10^{-1}	8.2×10^0	4.5×10^{-1}		8.0×10^{-1}	3.6×10^1
#3		8	n	5.6×10^{-3}	1.0×10^{-5}	1.1×10^{-1}	1.1×10^1	2.4×10^{-1}	9.0×10^{-1}	2.6×10^2	
#4			$n/2$	1.1×10^{-5}	1.0×10^{-6}	1.9×10^{-1}	6.6×10^0	3.5×10^{-1}	8.5×10^{-1}	3.1×10^2	
#5			$n/4$	1.1×10^{-5}	1.0×10^{-5}	2.7×10^{-1}	1.2×10^1	4.7×10^{-1}	7.8×10^{-1}	4.3×10^1	
#6		16	n	1.1×10^{-3}	1.0×10^{-5}	1.8×10^{-1}	8.4×10^0	2.5×10^{-1}	9.0×10^{-1}	3.2×10^2	
#7			$n/2$	1.1×10^{-5}	1.0×10^{-5}	2.6×10^{-1}	7.6×10^0	3.8×10^{-1}	8.3×10^{-1}	3.6×10^2	
#8			$n/4$	1.1×10^{-5}	1.0×10^{-5}	2.8×10^{-1}	1.6×10^1	4.8×10^{-1}	7.7×10^{-1}	5.5×10^1	

5.4.4. Observations

The most important observation is that the relative residual r increases and Dice score averages decrease for registrations done at coarser resolutions irrespective of whether we increase n_t proportionally to the resolution, see Table 7 or keep n_t fixed for different n_x , see Table 8. This observation is in line with the experiment for the synthetic dataset SYN in Section 5.3. Except for the case of na04 (see runs #10 and #11 in Table 7), all other cases exhibit increasingly worse registration performance at coarser resolutions.

In Section 5.3, we used synthetic, noise-free, high-contrast images for assessing the registration accuracy at different resolutions. Here, we repeat the same experiment with real world images—T1-weighted MR images of the human brain. We used an external software to segment these images to provide the necessary labels to be able to quantify registration performance in terms of Dice score. Notice that this additional segmentation step will inevitably introduce additional errors to our analysis. Due to these additional errors at the native resolution, we expect that the improvement in registration performance at high resolution may not be as pronounced as for the synthetic images considered in Section 5.3 (which did not require this additional segmentation step). This hypothesis is confirmed if we compare the average Dice score D_a across experiments. In particular, if we reduce the resolution from n to $n/4$ in Sections 5.3 (see Table 5) and 5.4 (see Table 7), the Dice score drops by 15.25% compared to 9.5%, respectively.

In Table 8, the case with $n_t = 4$ and $n_x = n$ took very long to converge (>4 h). For this case the CFL number is 15.66 during the inverse solve while for $n_t = 16$, the CFL number is 4. The larger CFL number for $n_t = 4$ yields a higher adjoint error in the SL scheme. This leads to higher errors in the computation of the reduced gradient, which results in worse convergence of the inverse solver for $n_t = 4$. The run time overhead associated with using $n_t = 16$ against $n_t = 4$ is easily compensated by better solver convergence. We refer to [30] for a thorough study on the effect of n_t on the numerical accuracy of the reduced gradient.

5.5. Experiment 3: Registration of Mouse Brain CLARITY Images

This experiment aims to answer both Q1 and Q2 by examining the performance of our scalable registration solver on ultra-high resolution mouse brain images acquired using the CLARITY imaging technique [26,99]. As opposed to the previous datasets, the dataset in

this experiment does not provide any real metrics for its assessment other than the relative residual (nor are we aware of any segmentation software that would work on these data).

5.5.1. Dataset

We use the CLARITY dataset (see Section 4) for this experiment.

5.5.2. Procedure.

Preprocessing: For all unprocessed images, the background intensity is non-zero. We normalize the image intensities such that they lie in the range [0,1] with the background intensity re-scaled to zero. Next, we affine register all images to `Contro1182` at $8\times$ down-sampled resolution using the SyN tool in ANTs. We report the parameter settings for the affine registration in the appendix. Subsequently, we zero-pad the images to ensure that periodic boundary conditions are satisfied for CLAIRe. After preprocessing, the base image resolution is $n_x = n$ where $n = (2816, 3016, 1162)$ and $n/8 = (328, 412, 1162)$, respectively. We only conduct the parameter search for a single pair of images (at both resolutions independently) for these sets of images and then perform the parameter continuation on the entire dataset. We only report wall clock times for the parameter continuation and not for the parameter search.

Deformable Registration: We register all images to the reference image `Contro1182` using CLAIRe. We use the proposed parameter continuation scheme. We set J_{min} to 0.05. We do this for both resolution levels. To compare the registration accuracy between each resolution level, we follow the same steps from Section 5.4. We compare the registration performance using the relative residual r . We do not have access to image segmentation for this dataset and, therefore, we cannot evaluate accuracy using Dice scores.

5.5.3. Results

We report the quantitative results for the registration of the CLARITY data in Table 9. We showcase exemplary registration results in Figure 8.

Table 9. Experiment 3: Registration performance for CLAIRe for the CLARITY imaging data at resolutions $n = (2816, 3016, 1162)$ and $n/8 = (328, 412, 1162)$. `Contro1182` is the fixed (reference) image. All other images selected from the CLARITY dataset are registered to `Contro1182` using a parameter continuation scheme. We fix the tolerance for the relative gradient to 5×10^{-2} . We use linear interpolation for the semi Langrangian scheme. The bounds on the determinant J of the deformation gradient for the parameter search are [0.05, 20]. We report the estimated regularization parameters β_v^* and β_w^* , the minimum and maximum values for the determinant of the deformation gradient (J_{min} and J_{max}), the relative residual (r), as well as the wall clock time for the parameter continuation.

Run	Image	#GPU	n_x	n_t	β_v^*	β_w^*	J_{min}	J_{max}	r	Runtime (s)
#1	Fear197	256	n	16	1.1×10^{-2}	1.0×10^{-5}	5.5×10^{-2}	2.2×10^1	3.4×10^{-1}	1.4×10^3
#2		8	$n/8$	16	1.0×10^{-3}	1.0×10^{-5}	5.8×10^{-2}	1.5×10^1	6.3×10^{-1}	9.6×10^1
#3	Cocain178	256	n	16	1.1×10^{-2}	1.0×10^{-5}	3.1×10^{-2}	1.2×10^1	4.1×10^{-1}	6.2×10^3
#4		8	$n/8$	16	5.6×10^{-3}	1.0×10^{-5}	3.5×10^{-2}	4.7×10^0	6.8×10^{-1}	6.1×10^1

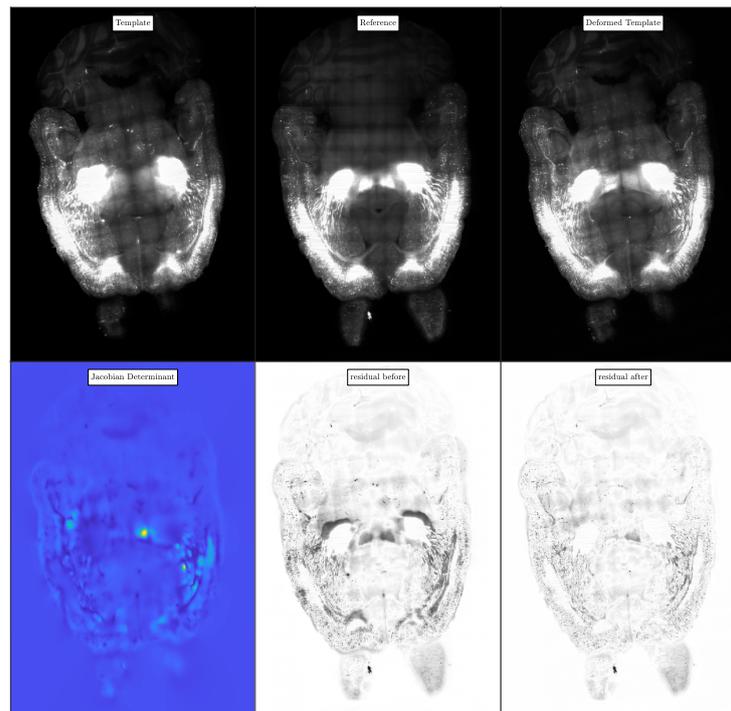


Figure 8. Experiment 3: Illustration of the registration performance for CLAIRE for the CLARITY mouse brain imaging data. We report registration results for the Cocaine178 dataset registered to the Control182 dataset. In row 1 (from left to right), we have the template image m_0 (Cocaine178), the reference image m_1 (Control182) and the deformed template image. The resolution of the images is $n = (2816, 3015, 1162)$. In row 2, we show the determinant of the deformation gradient and the image residuals before and after registration.

5.5.4. Observations

The most important observation is that we can register high resolution real medical images reasonably well in under 2 h (see run #1 and #3 in Table 9). Unlike the previous experiments in Sections 5.3 and 5.4, the reported wall clock time in Table 9 is for performing the parameter continuation and not the parameter search. The average time spent for the regularization parameter search for resolution $n_x = n$ is ~ 2 h. Another observation, which is in agreement with the results reported for the experiments carried out in Sections 5.3 and 5.4, is that the registration performed at downsampled resolution (see Table 9) results in a larger relative residual and, therefore, worse registration accuracy. We had a maximum of 256 GPUs (64 nodes, 4 GPUs per node) available to us at the TACC Longhorn supercomputer. Because of this resource constraint, our solver ran out of memory for certain parameter configurations (for example, for run #1 and #3, we could not use $n_t > 16$ time steps). Moreover, for all the runs in Table 9, we used the zero velocity approximation of \mathbf{H} as the preconditioner and applied it at full resolution. We did not use the two-level coarse grid approximation to apply the preconditioner because it requires additional memory for the coarse grid spectral operations.

6. Conclusions

In this publication, we apply our previously developed multi-node, multi-GPU 3D image registration solver [22] to study and analyze large-scale image registration. This work builds upon our former contributions on constrained large deformation diffeomorphic image registration [17,19,28,30,34]. The main observations are: (i) We are able to register CLARITY mouse brain images of unprecedented ultra-high spatial resolution ($2816 \times 3016 \times 1162$) in 23 min using parameter continuation. To the best of our knowledge, images of this scale have not been registered in previous work [22,97,98]. (ii) We conduct detailed experiments to compare image registration performance at full and downsampled

resolutions using synthetic and real images. We find that image registration at higher (native) image resolution is more accurate. To quantify the accuracy, we use Dice coefficients wherever image segmentation is available and relative residuals otherwise. We also do a sensitivity analysis for the overall solver accuracy with respect to the number of time steps n_t in the SL scheme. Overall, CLAIRE performed as expected; fully automatic parameter tuning works well, and higher image resolutions result in improved image similarity compared to the registration results in lower resolution. We note that these improvements in registration accuracy are less pronounced for real imaging data compared to synthetic data for the experiments conducted in this study. We attribute these observations to uncertainties and errors introduced during the registration and segmentation steps due to noise and low contrast. We discuss this in more detail in Section 5.

Author Contributions: Conceptualization, N.H., M.B., J.-Y.K., G.B., M.S. and A.M. Methodology, N.H., M.B., J.-Y.K., G.B., M.S. and A.M. Software, N.H., M.B., J.-Y.K. and A.M. Validation, N.H., M.B., J.-Y.K., G.B., M.S. and A.M. Formal Analysis, N.H., M.B., J.-Y.K., G.B., M.S. and A.M. Resources, G.B., M.S. and A.M. Data Curation, N.H., M.B., J.-Y.K. and A.M. Writing—Original Draft, N.H., G.B., M.S. and A.M. Writing—Review and Editing, N.H., M.B., G.B., M.S. and A.M. Visualization, N.H. and A.M. Supervision, G.B., M.S. and A.M. Project Administration, G.B. and M.S. Funding Acquisition, G.B., M.S. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the National Science Foundation (DMS-1854853, DMS-2012825, CCF-1817048, CCF-1725743), by the NVIDIA Corporation (NVIDIA GPU Grant Program), by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-EXC 2075-390740016, by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-SC0019393, by the U.S. Air Force Office of Scientific Research award FA9550-17-1-0190, by the Portugal Foundation for Science and Technology and the UT Austin–Portugal program, and by NIH award 5R01NS042645-11A1 and R21AG074276-01. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the DFG, AFOSR, DOE, NIH, and NSF. Computing time on the Texas Advanced Computing Center's (TACC) systems was provided by an allocation from TACC and the NSF. This work was completed in part with resources provided by the Research Computing Data Core at the University of Houston.

Institutional Review Board Statement: Ethical review and approval were waived for this study, because we conduct a retrospective analysis. All the datasets used in this study are either publicly available as mentioned in Section 4 or can be generated analytically, such as the SYN dataset. The ethics statements for those studies can be found in the corresponding references mentioned in Section 4.

Informed Consent Statement: Not applicable.

Data Availability Statement: We list the datasets we used for the experiments in this paper and the references and weblinks to download them. MUSE—This dataset consists of five T_1 -weighted MR images acquired at a spatial resolution of 1 mm. The image dataset is available for download at <http://www.neuromorphometrics.com> (accessed on 5 January 2022). NIREP [88]—This dataset consists of 16 T_1 -weighted MR images of different individuals imaged at a spatial resolution of 1 mm. This image dataset is available for download at the github repo <https://github.com/andreamang/nirep> (accessed on 5 January 2022). SYN—In Section 4, we described the exact process to generate the synthetic images and can be reproduced exactly using CLAIRE [21,34] and the scripts available at <https://github.com/naveenaero/scala-claire> (accessed on 5 January 2022). Due to the large size of these images (~4 GB), we have not hosted them on a public server and these can be made available on request. MRI250 [89]—This is a dataset consisting of a single high resolution brain MRI acquired at a spatial resolution of 250 μm . This image can be downloaded from <https://datadryad.org/stash/dataset/doi:10.5061/dryad.38s74> (accessed on 5 January 2022). CLARITY [27,96–98]—This is a dataset consisting of ultra-high resolution mouse brain images acquired using the CLARITY-optimized Light Sheet Microscopy (COLM) at a intra-planar spatial resolution of 0.585 μm and inter-planar resolution of 5 to 8 μm . This dataset is available for download at <https://neurodata.io/data/tomer15/> (accessed on 5 January 2022). CLAIRE [19,21,34] is available publicly on github

at <https://github.com/andreamang/claire> (accessed on 5 January 2022) under the GNU General Public License v3.0.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Deformable Registration Parameters for ANTs

We report the deformable registration parameters for ANTs which were used for comparison with CLAIRE in the parameter search experiment Section 5.2.

Listing A1. ANTs registration script.

```
#!/bin/bash
antsRegistration --dimensionality 3
--float 1
--output [$output_directory/, $output_directory/deformed-template.nii.gz]
--interpolation Linear
--winsorize-image-intensities [0.005,0.995]
--use-histogram-matching 1
--initial-moving-transform [$moving_image, $template_image, 1]
--transform Rigid[0.1]
--metric MI[$reference_image, $template_image, 1, 32, Regular, 0.25]
--convergence [1000x500x250x100, 1e-6, 10]
--shrink-factors 8x4x2x1
--smoothing-sigmas 3x2x1x0vox
--transform Affine[0.1]
--metric MI[$reference_image, $template_image, 1, 32, Regular, 0.25]
--convergence [1000x500x250x100, 1e-6, 10]
--shrink-factors 8x4x2x1
--smoothing-sigmas 3x2x1x0vox
--transform SyN[0.1, 3, 0]
--metric MeanSquares[$reference_image, $template_image, 1]
--convergence [100x70x50x20, 1e-6, 10]
--shrink-factors 8x4x2x1
--smoothing-sigmas 3x2x1x0vox
```

Appendix B. Determining $\beta_{w,init}$

We report the runs for comparison of runtime and Dice scores for different values of $\beta_{w,init}$ for the experiment conducted in Section 5.4.

Table A1. Experiment 1b: effect of $\beta_{w,init}$ on registration performance for real brain images: Comparison of registration accuracy using Dice and relative residual r for different values of $\beta_{w,init}$ at different resolutions for registration of MRI250 brain image to na01 and na02 from NIREP dataset. We fix $\beta_{w,min} = 1 \times 10^{-9}$. We consider $n_x = \{N, N/4\}$ where $N = 640 \times 880 \times 880$. We fix the tolerance for the relative gradient to 5×10^{-2} . We use linear interpolation. The Jacobian bounds for parameter search are $[0.05, 20]$. We increase the number of time steps n_t proportionately with increase in resolution. We report β_v^* and β_w^* , the regularization parameters from the parameter search scheme, J_{min} and J_{max} , the minimum and maximum Jacobian determinant the relative residual r , average Dice D_a pre and post the registration and the wall clock time for the parameter search for the registration. We highlight the best Dice scores for each resolution and for each NIREP image.

Run	NIREP	$\beta_{w,init}$	n_x	β_v^*	β_w^*	J_{min}	J_{max}	r	D_a		Runtime (s) Search	
									Pre	Post		
#1	na01	1×10^{-4}	N	1.1×10^{-5}	1.0×10^{-9}	2.8×10^{-1}	2.7×10^0	3.4×10^{-1}	5.5×10^{-1}	8.6×10^{-1}	5.0×10^3	
#2			N/4	2.3×10^{-5}	1.0×10^{-5}	3.5×10^{-1}	2.8×10^0	4.6×10^{-1}		7.9×10^{-1}	3.5×10^1	
#3		1×10^{-5}	N	1.1×10^{-3}	1.0×10^{-5}	1.8×10^{-1}	8.3×10^0	2.5×10^{-1}	5.5×10^{-1}	9.0×10^{-1}	3.1×10^2	
#4			N/4	1.1×10^{-5}	1.0×10^{-5}	2.3×10^{-1}	8.2×10^0	4.5×10^{-1}		8.0×10^{-1}	3.6×10^1	
#5		1×10^{-6}	N	2.0×10^{-2}	1.0×10^{-6}	5.2×10^{-2}	1.6×10^1	3.0×10^{-1}	5.5×10^{-1}	8.6×10^{-1}	2.1×10^2	
#6			N/4	1.0×10^{-3}	1.0×10^{-6}	1.2×10^{-1}	1.6×10^1	3.7×10^{-1}		8.5×10^{-1}	1.2×10^1	
#7		1×10^{-7}	N	5.1×10^{-2}	1.0×10^{-9}	1.8×10^{-1}	1.0×10^1	4.1×10^{-1}	5.5×10^{-1}	7.9×10^{-1}	2.1×10^2	
#8			N/4	6.6×10^{-3}	1.0×10^{-7}	1.1×10^{-1}	1.8×10^1	4.0×10^{-1}		8.2×10^{-1}	7.7×10^0	
#9		na02	1×10^{-4}	N	1.1×10^{-5}	1.0×10^{-9}	2.1×10^{-1}	2.7×10^0	3.4×10^{-1}	5.4×10^{-1}	8.4×10^{-1}	5.7×10^3
#10				N/4	1.1×10^{-5}	1.0×10^{-6}	8.7×10^{-2}	9.9×10^0	4.7×10^{-1}		7.7×10^{-1}	4.0×10^1
#11			1×10^{-5}	N	1.1×10^{-3}	1.0×10^{-5}	7.7×10^{-2}	6.3×10^0	2.5×10^{-1}	5.4×10^{-1}	8.9×10^{-1}	3.4×10^2
#12				N/4	1.1×10^{-5}	1.0×10^{-6}	8.1×10^{-2}	1.1×10^1	4.6×10^{-1}		7.7×10^{-1}	3.9×10^1
#13			1×10^{-6}	N	4.1×10^{-2}	1.0×10^{-7}	1.4×10^{-1}	2.0×10^1	3.8×10^{-1}	5.4×10^{-1}	8.0×10^{-1}	2.4×10^2
#14				N/4	1.1×10^{-4}	1.0×10^{-6}	8.2×10^{-2}	1.2×10^1	4.1×10^{-1}		8.1×10^{-1}	1.4×10^1
#15			1×10^{-7}	N	4.0×10^{-2}	1.0×10^{-9}	1.2×10^{-1}	1.8×10^1	3.8×10^{-1}	5.4×10^{-1}	8.0×10^{-1}	2.5×10^2
#16				N/4	1.0×10^{-3}	1.0×10^{-7}	8.8×10^{-2}	2.0×10^1	3.8×10^{-1}		8.3×10^{-1}	1.4×10^1

References

- Hajnal, J.V.; Hill, D.L.G.; Hawkes, D.J. (Eds.) *Medical Image Registration*; CRC Press: Boca Raton, FL, USA, 2001.
- Sotiras, A.; Davatzikos, C.; Paragios, N. Deformable medical image registration: A survey. *IEEE Trans. Med. Imaging* **2013**, *32*, 1153–1190. [CrossRef] [PubMed]
- Modersitzki, J. *Numerical Methods for Image Registration*; Oxford University Press: Oxford, UK, 2004.
- Modersitzki, J. FLIRT with rigidity—Image registration with a local non-rigidity penalty. *Int. J. Comput. Vis.* **2008**, *76*, 153–163. [CrossRef]
- Beg, M.F.; Miller, M.I.; Trounev, A.; Younes, L. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *Int. J. Comput. Vis.* **2005**, *61*, 139–157. [CrossRef]
- Trounev, A. Diffeomorphism groups and pattern matching in image analysis. *Int. J. Comput. Vis.* **1998**, *28*, 213–221. [CrossRef]
- Younes, L. *Shapes and Diffeomorphisms*; Springer: Berlin/Heidelberg, Germany, 2010.
- Fischer, B.; Modersitzki, J. Ill-posed medicine—An introduction to image registration. *Inverse Probl.* **2008**, *24*, 1–16. [CrossRef]
- Vercauteren, T.; Pennec, X.; Perchant, A.; Ayache, N. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage* **2009**, *45*, S61–S72. [CrossRef]
- Vercauteren, T.; Pennec, X.; Perchant, A.; Ayache, N. Diffeomorphic Demons Using ITK’s Finite Difference Solver Hierarchy. *Insight J.* **2007**. Available online: <http://hdl.handle.net/1926/510> (accessed on 5 January 2022).
- Avants, B.B.; Tustison, N.J.; Stauffer, M.; Song, G.; Wu, B.; Gee, J.C. The Insight ToolKit Image Registration Framework. *Front. Neuroinformatics* **2014**, *8*, 44. [CrossRef]
- Yang, X.; Kwitt, R.; Styner, M.; Niethammer, M. Quicksilver: Fast predictive image registration—A deep learning approach. *NeuroImage* **2017**, *158*, 378–396. [CrossRef]
- Krebs, J.; Mansi, T.; Mailhé, B.; Ayache, N.; Delingette, H. Unsupervised probabilistic deformation modeling for robust diffeomorphic registration. In Proceedings of the International Workshop on Deep Learning in Medical Image Analysis 2018, Granada, Spain, 20 September 2018; pp. 101–109.
- Gu, X.; Pan, H.; Liang, Y.; Castillo, R.; Yang, D.; Choi, D.; Castillo, E.; Majumdar, A.; Guerrero, T.; Jiang, S.B. Implementation and evaluation of various demons deformable image registration algorithms on a GPU. *Phys. Med. Biol.* **2009**, *55*, 207–219. [CrossRef]
- Zhang, M.; Fletcher, P.T. Fast Diffeomorphic Image Registration via Fourier-Approximated Lie Algebras. *Int. J. Comput. Vis.* **2019**, *127*, 61–73. [CrossRef]

16. Grzech, D.; Folgoc, L.; Heinrich, M.P.; Khanal, B.; Moll, J.; Schnabel, J.A.; Glocker, B.; Kainz, B. FastReg: Fast Non-Rigid Registration via Accelerated Optimisation on the Manifold of Diffeomorphisms. *arXiv* **2019**, arXiv:1903.01905.
17. Mang, A.; Gholami, A.; Biros, G. Distributed-memory large-deformation diffeomorphic 3D image registration. In Proceedings of the ACM/IEEE Conference on Supercomputing 2016, Salt Lake City, UT, USA, 13–18 November 2016.
18. Gholami, A.; Mang, A.; Biros, G. An inverse problem formulation for parameter estimation of a reaction-diffusion model of low grade gliomas. *J. Math. Biol.* **2016**, *72*, 409–433. [[CrossRef](#)] [[PubMed](#)]
19. Mang, A.; Gholami, A.; Davatzikos, C.; Biros, G. CLAIRE: A distributed-memory solver for constrained large deformation diffeomorphic image registration. *SIAM J. Sci. Comput.* **2019**, *41*, C548–C584. [[CrossRef](#)] [[PubMed](#)]
20. Brunn, M.; Himthani, N.; Biros, G.; Mehl, M.; Mang, A. Fast GPU 3D Diffeomorphic Image Registration. *J. Parallel Distrib. Comput.* **2021**, *149*, 149–162. [[CrossRef](#)] [[PubMed](#)]
21. Brunn, M.; Himthani, N.; Biros, G.; Mehl, M.; Mang, A. CLAIRE: Constrained Large Deformation Diffeomorphic Image Registration on Parallel Computing Architectures. *J. Open Source Softw.* **2021**, *6*, 3038. [[CrossRef](#)]
22. Brunn, M.; Himthani, N.; Biros, G.; Mehl, M.; Mang, A. Multi-Node Multi-GPU Diffeomorphic Image Registration for Large-Scale Imaging Problems. In Proceedings of the SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, Virtual, 16–19 November 2020; pp. 1–17. [[CrossRef](#)]
23. Chung, K.; Wallace, J.; Kim, S.Y.; Kalyanasundaram, S.; Andalman, A.S.; Davidson, T.J.; Mirzabekov, J.J.; Zalocusky, K.A.; Mattis, J.; Denisin, A.K.; et al. Structural and molecular interrogation of intact biological systems. *Nature* **2013**, *497*, 332–337. [[CrossRef](#)]
24. Kim, S.Y.; Chung, K.; Deisseroth, K. Light microscopy mapping of connections in the intact brain. *Trends Cogn. Sci.* **2013**, *17*, 596–599. [[CrossRef](#)]
25. Kutten, K.S.; Charon, N.; Miller, M.I.; Ratnanather, J.T.; Deisseroth, K.; Ye, L.; Vogelstein, J.T. A diffeomorphic approach to multimodal registration with mutual information: Applications to CLARITY mouse brain images. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention, Quebec City, QC, Canada, 11–13 September 2017; pp. 275–282.
26. Tomer, R.; Ye, L.; Hsueh, B.; Deisseroth, K. Advanced CLARITY for rapid and high-resolution imaging of intact tissues. *Nat. Protoc.* **2014**, *9*, 1682–1697. [[CrossRef](#)]
27. Vogelstein, J.T.; Perlman, E.; Falk, B.; Baden, A.; Gray Roncal, W.; Chandrashekhar, V.; Collman, F.; Seshamani, S.; Patsolic, J.L.; Lillaney, K.; et al. A community-developed open-source computational ecosystem for big neuro data. *Nat. Methods* **2018**, *15*, 846–847. [[CrossRef](#)]
28. Mang, A.; Biros, G. An inexact Newton–Krylov algorithm for constrained diffeomorphic image registration. *SIAM J. Imaging Sci.* **2015**, *8*, 1030–1069. [[CrossRef](#)]
29. Mang, A.; Biros, G. Constrained H^1 -regularization schemes for diffeomorphic image registration. *SIAM J. Imaging Sci.* **2016**, *9*, 1154–1194. [[CrossRef](#)] [[PubMed](#)]
30. Mang, A.; Biros, G. A Semi-Lagrangian two-level preconditioned Newton–Krylov solver for constrained diffeomorphic image registration. *SIAM J. Sci. Comput.* **2017**, *39*, B1064–B1101. [[CrossRef](#)] [[PubMed](#)]
31. Mang, A.; Gholami, A.; Davatzikos, C.; Biros, G. PDE-constrained optimization in medical image analysis. *Optim. Eng.* **2018**, *19*, 765–812. [[CrossRef](#)]
32. Gholami, A.; Mang, A.; Scheufele, K.; Davatzikos, C.; Mehl, M.; Biros, G. A Framework for Scalable Biophysics-based Image Analysis. In Proceedings of the ACM/IEEE Conference on Supercomputing, Denver, CO, USA, 12–17 November 2017; pp. 1–13.
33. Mang, A.; Ruthotto, L. A Lagrangian Gauss–Newton–Krylov solver for mass- and intensity-preserving diffeomorphic image registration. *SIAM J. Sci. Comput.* **2017**, *39*, B860–B885. [[CrossRef](#)]
34. Mang, A.; Biros, G. Constrained Large Deformation Diffeomorphic Image Registration (CLAIRE). 2019. Available online: <https://andreamang.github.io/claire> (accessed on 5 January 2022).
35. Mang, A.; Bakas, S.; Subramanian, S.; Davatzikos, C.; Biros, G. Integrated biophysical modeling and image analysis: Application to neuro-oncology. *Annu. Rev. Biomed. Eng.* **2020**, *22*, 309–341. [[CrossRef](#)]
36. Scheufele, K.; Mang, A.; Gholami, A.; Davatzikos, C.; Biros, G.; Mehl, M. Coupling brain-tumor biophysical models and diffeomorphic image registration. *Comput. Methods Appl. Mech. Eng.* **2019**, *347*, 533–567. [[CrossRef](#)]
37. Scheufele, K.; Subramanian, S.; Mang, A.; Biros, G.; Mehl, M. Image-driven biophysical tumor growth model calibration. *SIAM J. Sci. Comput.* **2020**, *42*, B549–B580. [[CrossRef](#)]
38. Avants, B.B.; Tustison, N.J.; Song, G.; Cook, P.A.; Klein, A.; Gee, J.C. A reproducible evaluation of ANTs similarity metric performance in brain image registration. *NeuroImage* **2011**, *54*, 2033–2044. [[CrossRef](#)]
39. Avants, B.B.; Epstein, C.L.; Brossman, M.; Gee, J.C. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Med. Image Anal.* **2008**, *12*, 26–41. [[CrossRef](#)]
40. Avants, B.B.; Tustison, N.J.; Johnson, H.J. ANTs. Available online: <http://stnava.github.io/ANTs> (accessed on 3 September 2020).
41. Ashburner, J. A fast diffeomorphic image registration algorithm. *NeuroImage* **2007**, *38*, 95–113. [[CrossRef](#)]
42. Bone, A.; Colliot, O.; Durrleman, S. Learning distributions of shape trajectories from longitudinal datasets: A hierarchical model on a manifold of diffeomorphisms. *arXiv* **2019**, arXiv:1803.10119.
43. Bone, A.; Louis, M.; Martin, B.; Durrleman, S. Deformetrica 4: An open-source software for statistical shape analysis. In Proceedings of the International Workshop on Shape in Medical Imaging, Granada, Spain, 20 September 2018; pp. 3–13.
44. Fishbaugh, J.; Durrleman, S.; Prastawa, M.; Gerig, G. Geodesic shape regression with multiple geometries and sparse parameters. *Med. Image Anal.* **2017**, *39*, 1–17. [[CrossRef](#)]

45. Durrleman, S.; Brone, A.; Louis, M.; Martin, B.; Gori, P.; Routier, A.; Bacci, M.; Fouquier, A.; Charlier, B.; Glaunes, J.; et al. Deformetrica. Available online: <http://www.deformetrica.org> (accessed on 3 September 2020).
46. Center for Imaging Science, Johns Hopkins University. LDDMM Suite. Available online: <http://cis.jhu.edu/software> (accessed on 5 January 2022).
47. Neurodata. ARDENT. Available online: <https://ardent.neurodata.io> (accessed on 5 January 2022).
48. Insight Software Consortium. ITKNDReg. Available online: <https://github.com/InsightSoftwareConsortium/ITKNDReg> (accessed on 5 January 2022).
49. Preston, J.S. Python for Computational Anatomy. Available online: <https://bitbucket.org/scicompanat/pyca> (accessed on 5 January 2022).
50. Fluck, O.; Vetter, C.; Wein, W.; Kamen, A.; Preim, B.; Westermann, R. A survey of medical image registration on graphics hardware. *Comput. Methods Programs Biomed.* **2011**, *104*, e45–e57. [[CrossRef](#)] [[PubMed](#)]
51. Shams, R.; Sadeghi, P.; Kennedy, R.A.; Hartley, R.I. A survey of medical image registration on multicore and the GPU. *Signal Process. Mag.* **2010**, *27*, 50–60. [[CrossRef](#)]
52. Eklund, A.; Dufort, P.; Forsberg, D.; LaConte, S.M. Medical image processing on the GPU—Past, present and future. *Med. Image Anal.* **2013**, *17*, 1073–1094. [[CrossRef](#)]
53. Budelmann, D.; Koenig, L.; Papenberg, N.; Lellmann, J. Fully-deformable 3D image registration in two seconds. In Proceedings of the Bildverarbeitung für die Medizin, L'übeck, Germany, 17–19 March 2019; pp. 302–307.
54. Courty, N.; Hellier, P. Accelerating 3D non-rigid registration using graphics hardware. *Int. J. Image Graph.* **2008**, *8*, 81–98. [[CrossRef](#)]
55. Durrleman, S.; Prastawa, M.; Charon, N.; Korenberg, J.R.; Joshi, S.; Gerig, G.; Trouve, A. Morphometry of anatomical shape complexes with dense deformations and sparse parameters. *NeuroImage* **2014**, *101*, 35–49. [[CrossRef](#)]
56. Ellingwood, N.D.; Yin, Y.; Smith, M.; Lin, C.L. Efficient methods for implementation of multi-level nonrigid mass-preserving image registration on GPUs and multi-threaded CPUs. *Comput. Methods Programs Biomed.* **2016**, *127*, 290–300. [[CrossRef](#)]
57. Ha, L.K.; Krüger, J.; Fletcher, P.T.; Joshi, S.; Silva, C.T. Fast parallel unbiased diffeomorphic atlas construction on multi-graphics processing units. In Proceedings of the Eurographics Conference on Parallel Graphics and Visualization, Munich, Germany, 29–30 March 2009; pp. 41–48.
58. Ha, L.; Krüger, J.; Joshi, S.; Silva, C.T. Multiscale unbiased diffeomorphic atlas construction on multi-GPUs. In *CPU Computing Gems Emerald Edition*; Elsevier Inc.: Amsterdam, The Netherlands, 2011; Chapter 48, pp. 771–791.
59. Joshi, S.; Davis, B.; Jorner, M.; Gerig, G. Unbiased diffeomorphic atlas construction for computational anatomy. *NeuroImage* **2005**, *23*, S151–S160. [[CrossRef](#)]
60. Koenig, L.; Ruehaak, J.; Derksen, A.; Lellmann, J. A matrix-free approach to parallel and memory-efficient deformable image registration. *SIAM J. Sci. Comput.* **2018**, *40*, B858–B888. [[CrossRef](#)]
61. Modat, M.; Ridgway, G.R.; Taylor, Z.A.; Lehmann, M.; Barnes, J.; Hawkes, D.J.; Fox, N.C.; Ourselin, S. Fast free-form deformation using graphics processing units. *Comput. Methods Programs Biomed.* **2010**, *98*, 278–284. [[CrossRef](#)] [[PubMed](#)]
62. Sommer, S. Accelerating multi-scale flows for LDDKBM diffeomorphic registration. In Proceedings of the Proc IEEE International Conference on Computer Vision Workshops, Barcelona, Spain, 6–13 November 2011; pp. 499–505.
63. Shackelford, J.; Kandasamy, N.; Sharp, G. On developing B-spline registration algorithms for multi-core processors. *Phys. Med. Biol.* **2010**, *55*, 6329–6351. [[CrossRef](#)]
64. Shamonin, D.P.; Bron, E.E.; Lelieveldt, B.P.F.; Smits, M.; Klein, S.; Staring, M. Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer's disease. *Front. Neuroinformatics* **2014**, *7*, 1–15. [[CrossRef](#)] [[PubMed](#)]
65. Valero-Lara, P. A GPU approach for accelerating 3D deformable registration (DARTEL) on brain biomedical images. In Proceedings of the European MPI Users' Group Meeting, Madrid, Spain, 15–18 September 2013;
66. Valero-Lara, P. Multi-GPU acceleration of DARTEL (early detection of Alzheimer). In Proceedings of the IEEE International Conference on Cluster Computing, Madrid, Spain, 22–26 September 2014; pp. 346–354.
67. Nazib, A.; Galloway, J.; Fookes, C.; Perrin, D. Performance of Registration Tools on High-Resolution 3D Brain Images. In Proceedings of the 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, 17–21 July 2018; pp. 566–569. [[CrossRef](#)]
68. Susaki, E.A.; Tainaka, K.; Perrin, D.; Yukinaga, H.; Kuno, A.; Ueda, H.R. Advanced CUBIC Protocols for Whole-Brain and Whole-Body Clearing and Imaging. *Nat. Protoc.* **2015**, *10*, 1709–1727. [[CrossRef](#)]
69. Klein, S.; Staring, M.; Murphy, K.; Viergever, M.A.; Pluim, J.P.W. ELASTIX: A toolbox for intensity-based medical image registration. *IEEE Trans. Med. Imaging* **2010**, *29*, 196–205. [[CrossRef](#)]
70. Niedworok, C.J.; Brown, A.P.Y.; Jorge Cardoso, M.; Osten, P.; Ourselin, S.; Modat, M.; Margrie, T.W. aMAP Is a Validated Pipeline for Registration and Segmentation of High-Resolution Mouse Brain Data. *Nat. Commun.* **2016**, *7*, 11879. [[CrossRef](#)]
71. Kuan, L.; Li, Y.; Lau, C.; Feng, D.; Bernard, A.; Sunkin, S.M.; Zeng, H.; Dang, C.; Hawrylycz, M.; Ng, L. Neuroinformatics of the Allen Mouse Brain Connectivity Atlas. *Methods* **2015**, *73*, 4–17. [[CrossRef](#)] [[PubMed](#)]
72. Hernandez, M.; Bossa, M.N.; Olmos, S. Registration of anatomical images using paths of diffeomorphisms parameterized with stationary vector field flows. *Int. J. Comput. Vis.* **2009**, *85*, 291–306. [[CrossRef](#)]
73. Lorenzi, M.; Pennec, X. Geodesics, parallel transport and one-parameter subgroups for diffeomorphic image registration. *Int. J. Comput. Vis.* **2013**, *105*, 111–127. [[CrossRef](#)]

74. Gholami, A.; Biros, G. AccFFT, 2017. Available online: <https://github.com/amirgholami/accfft> (accessed on 3 January 2017).
75. Gholami, A.; Biros, G. AccFFT Home Page, 2017. Available online: <http://www.accfft.org> (accessed on 3 January 2017).
76. Balay, S.; Gropp, W.D.; McInnes, L.C.; Smith, B.F. Efficient Management of Parallelism in Object Oriented Numerical Software Libraries. In *Modern Software Tools in Scientific Computing*; Arge, E., Bruaset, A.M., Langtangen, H.P., Eds.; Birkhäuser Press: Boston, MA, USA, 1997; pp. 163–202.
77. Balay, S.; Abhyankar, S.; Adams, M.F.; Brown, J.; Brune, P.; Buschelman, K.; Dalcin, L.; Dener, A.; Eijkhout, V.; Gropp, W.D.; et al. PETSc and TAO Webpage (PETSc Version 3.12.4). Available online: <https://www.mcs.anl.gov/petsc> (accessed on 4 February 2020).
78. NVIDIA. CUDA Toolkit (Version 10.1). Available online: <https://developer.nvidia.com/cuda-downloads> (accessed on 27 February 2019).
79. Hoberock, J.; Bell, N. Thrust. Thrust, the Cuda c++ Template Library. 2010. Available online: <https://docs.nvidia.com/cuda/thrust/index.html> (accessed on 27 February 2019).
80. Nvidia. CUDA CUFFT Library, Available online: <https://docs.nvidia.com/cuda/cufft/index.html> (accessed on 27 February 2019).
81. Fissell, K.; Reynolds, R. Niftilib (Version 2.2.0). 2020. Available online: <http://niftilib.sourceforge.net> (accessed on 5 January 2020).
82. Latham, R.; Zingale, M.; Thakur, R.; Gropp, W.; Gallagher, B.; Liao, W.; Siegel, A.; Ross, R.; Choudhary, A.; Li, J. Parallel netCDF: A High-Performance Scientific I/O Interface. In Proceedings of the SC Conference, Phoenix, AZ, USA, 15–21 November 2003; IEEE Computer Society: Los Alamitos, CA, USA, 2003; p. 39. [CrossRef]
83. IBM Spectrum MPI (Version 10.3.0). Available online: <https://www.ibm.com/products/spectrum-mpi> (accessed on 5 January 2022).
84. IBM. IBM XL C/C++ (Version 16.1.1). Available online: <https://www.ibm.com/us-en/marketplace/xl-cpp-linux-compiler-power> (accessed on 5 January 2022).
85. Haber, E.; Oldenburg, D. A GCV Based Method for Nonlinear Ill-Posed Problems. *Comput. Geosci.* **2000**, *4*, 41–63. [CrossRef]
86. Neuromorphometrics. Available online: <http://www.neuromorphometrics.com> (accessed on 5 January 2022).
87. Doshi, J.; Erus, G.; Ou, Y.; Resnick, S.M.; Gur, R.C.; Gur, R.E.; Satterthwaite, T.D.; Furth, S.; Davatzikos, C. MUSE: MUlti-atlas Region Segmentation Utilizing Ensembles of Registration Algorithms and Parameters, and Locally Optimal Atlas Selection. *NeuroImage* **2016**, *127*, 186–195. [CrossRef] [PubMed]
88. Christensen, G.E.; Geng, X.; Kuhl, J.G.; Bruss, J.; Grabowski, T.J.; Pirwani, I.A.; Vannier, M.W.; Allen, J.S.; Damasio, H. Introduction to the non-rigid image registration evaluation project. In Proceedings of the Biomedical Image Registration 2006, Utrecht, The Netherlands, 9–11 July 2006; pp. 128–135.
89. Lüsebrink, F.; Sciarra, A.; Mattern, H.; Yakupov, R.; Speck, O. T1-Weighted in Vivo Human Whole Brain MRI Dataset with an Ultrahigh Isotropic Resolution of 250 μm . *Sci. Data* **2017**, *4*, 170032. [CrossRef] [PubMed]
90. Data from: T1-Weighted In Vivo Human Whole Brain MRI Dataset with an Ultrahigh Isotropic Resolution of 250 μm . Available online: <https://datadryad.org/stash/dataset/doi:10.5061/dryad.38s74> (accessed on 5 January 2022). [CrossRef]
91. Yushkevich, P.A.; Piven, J.; Cody Hazlett, H.; Gimpel Smith, R.; Ho, S.; Gee, J.C.; Gerig, G. User-Guided 3D Active Contour Segmentation of Anatomical Structures: Significantly Improved Efficiency and Reliability. *Neuroimage* **2006**, *31*, 1116–1128. [CrossRef]
92. Zhang, Y.; Brady, M.; Smith, S. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Trans. Med. Imaging* **2001**, *20*, 45–57. [CrossRef] [PubMed]
93. Woolrich, M.W.; Jbabdi, S.; Patenaude, B.; Chappell, M.; Makni, S.; Behrens, T.; Beckmann, C.; Jenkinson, M.; Smith, S.M. Bayesian analysis of neuroimaging data in FSL. *Neuroimage* **2009**, *45*, S173–S186. [CrossRef]
94. Smith, S.M.; Jenkinson, M.; Woolrich, M.W.; Beckmann, C.F.; Behrens, T.E.; Johansen-Berg, H.; Bannister, P.R.; De Luca, M.; Drobnjak, I.; Flitney, D.E.; et al. Advances in functional and structural MR image analysis and implementation as FSL. *Neuroimage* **2004**, *23*, S208–S219. [CrossRef]
95. Jenkinson, M.; Beckmann, C.F.; Behrens, T.E.; Woolrich, M.W.; Smith, S.M. FSL. *Neuroimage* **2012**, *62*, 782–790. [CrossRef]
96. Chung, K.; Deisseroth, K. CLARITY for mapping the nervous system. *Nat. Methods* **2013**, *10*, 508–513. [CrossRef]
97. Kuttan, K.S.; Vogelstein, J.T.; Charon, N.; Ye, L.; Deisseroth, K.; Miller, M.I. Deformably Registering and Annotating Whole CLARITY Brains to an Atlas via Masked LDDMM. In Proceedings of the Optics, Photonics and Digital Technologies for Imaging Applications IV, Brussels, Belgium 4 March 2016; Volume 9896, pp. 282–290. [CrossRef]
98. Kuttan, K.S.; Charon, N.; Miller, M.I.; Ratnanather, J.T.; Matelsky, J.; Baden, A.D.; Lillaney, K.; Deisseroth, K.; Ye, L.; Vogelstein, J.T. A Large Deformation Diffeomorphic Approach to Registration of CLARITY Images via Mutual Information. *arXiv* **2017**, arXiv:1612.00356.
99. Chandrashekar, V.; Crow, A.; Bogelstein, J.; Deisseroth, K. Neurodata Claritomes. Available online: <https://neurodata.io/project/claritomes> (accessed on 5 January 2022).