Fast 3D diffeomorphic image registration on GPUs

Malte Brunn
University of Stuttgart
Stuttgart, Germany
malte.brunn@ipvs.uni-stuttgart.de

Naveen Himthani
University of Texas at Austin
Austin, TX, USA
naveen@ices.utexas.edu

George Biros
University of Texas at Austin
Austin, TX, USA
gbiros@acm.org

Miriam Mehl
University of Stuttgart
Stuttgart, Germany
miriam.mehl@ipvs.uni-stuttgart.de

Andreas Mang
University of Houston
Houston, TX, USA
andreas@math.uh.edu

1 INTRODUCTION

Image registration (also known as image alignment, warping, or matching) is an important task in medical image analysis [11, 23, 24, 28]. It is used in computer aided diagnosis and clinical population studies. The image registration problem is as follows: *Given two images $m_0(\boldsymbol{x})$ and $m_1(\boldsymbol{x})$ (where $\boldsymbol{x} \in (0, 2\pi]^3$), we seek a transformation $\boldsymbol{y}(\boldsymbol{x})$ such that $m_0(\boldsymbol{y}(\boldsymbol{x}))$ is similar to $m_1(\boldsymbol{x})$* [23]. Registration methods can be classified based on the parameterization for $\boldsymbol{y}$. In this poster, we consider methods that belong or are related to large-deformation diffeomorphic metric mapping (**LDDMM**) [4, 33]. Here, one introduces a pseudo-time variable $t \in [0, 1]$ and inverts for a time-dependent, smooth velocity field $\boldsymbol{v}$ that parameterizes the deformation map $\boldsymbol{y}$. Such mappings provide maximal flexibility [28] but are expensive to compute since they are infinite dimensional. Upon discretization, the number of unknowns for $\boldsymbol{y}$ is still in the millions. Furthermore, image registration is a highly non-linear and ill-conditioned inverse problem [11]. As a result, solving image registration problems can take a few minutes on multi-core high-end CPUs. Large clinical, cross-center, population-study workflows require thousands of registrations. GPUs with their inherent parallelism and low energy consumption are an attractive choice to achieve this goal. However, despite the need for HPC-performance for registration and the existence of several software libraries for LDDMM registration, there is little work on optimized GPU implementations. We refer to [10, 12, 27] for surveys on GPU accelerated solvers. Popular software packages for deformable registration are described in [1–3, 17, 22, 26, 32]. The work that is most closely related to ours is [5, 7, 9, 13–16, 30, 31]

We extend the publicly available open source diffeomorphic image registration framework `CLAIRE` [19, 21] to support a single GPU accelerator. This involves contributions to the GPU accelerated interpolation kernel, the semi-Lagrangian time stepping scheme, spectral differentiation and finite difference methods, and the solution of the underlying partial differential equations (**PDE**s). Additionally, we present performance results for our new implementation. As a highlight, we demonstrate that we can register 256^3 clinical images in less than 6 seconds on a single NVIDIA Tesla V100. This amounts to over 20× speed-up over the current version of `CLAIRE` and over 30× speed-up over existing GPU implementations.

2 METHODS

`CLAIRE` uses an optimal control formulation [21]. Here, the problem of diffeomorphic image registration is formulated in terms of a ***stationary velocity*** $\boldsymbol{v}(\boldsymbol{x})$ that parameterizes the deformation map $\boldsymbol{y}(x)$. Given two images $m_0(\boldsymbol{x})$ (template image; image to be registered) and $m_1(\boldsymbol{x})$ (reference image), we seek $\boldsymbol{v}(\boldsymbol{x})$ by solving

$$\min_{\boldsymbol{v}} \frac{1}{2}\int_\Omega (m(\boldsymbol{x},1) - m_1(\boldsymbol{x}))^2 \mathrm{d}\boldsymbol{x} + \frac{\beta}{2}\int_\Omega \langle \mathcal{A}\boldsymbol{v}(\boldsymbol{x}), \boldsymbol{v}(\boldsymbol{x})\rangle \,\mathrm{d}\boldsymbol{x} \tag{1a}$$

$$\text{s. t.} \quad \partial_t m(\boldsymbol{x},t) + \boldsymbol{v}(\boldsymbol{x})\cdot\nabla m(\boldsymbol{x},t) = 0 \quad \text{in } \Omega\times(0,1]$$
$$m(\boldsymbol{x},t) = m_0(\boldsymbol{x}) \quad \text{in } \Omega\times\{0\} \tag{1b}$$

with periodic boundary conditions on $\partial\Omega$. The PDE constraint (1b) is the forward problem of our formulation—the geometric transformation of the template image $m_0(\boldsymbol{x})$ in terms of the velocity field $\boldsymbol{v}(\boldsymbol{x})$. The first term in (1a) is an image similarity measure (we use a squared L^2-distance). The second term in (1a) is a Tikhonov regularization functional with parameter $\beta > 0$. It ensures smoothness of $\boldsymbol{v}(\boldsymbol{x})$, and if chosen adequately guarantees that the transformation of $m_0(\boldsymbol{x})$ exists and is a diffeomorphism [29]. We follow the default configuration of `CLAIRE` and select $\mathcal{A}$ to be a vector Laplacian with an additional penalty on the divergence of $\boldsymbol{v}$ (details can be found in [18, 21]). To solve (1), we use the method of Lagrange multipliers and derive first-order optimality conditions as variations with respect to m, λ (adjoint variable), and $\boldsymbol{v}$ amounting to a set of coupled, nonlinear, hyperbolic-elliptic PDEs in ***4D (space-time)***. `CLAIRE` uses a ***reduced-space approach***, i.e., it iterates on $\boldsymbol{v}(\boldsymbol{x})$, only: given a candidate $\boldsymbol{v}(\boldsymbol{x})$, it solves the forward and adjoint equations for $m(\boldsymbol{x},t)$ and $\lambda(\boldsymbol{x},t)$ and substitutes the solutions into the gradient $\boldsymbol{g}(\boldsymbol{v})$ (variation of the Lagrangian with respect to $\boldsymbol{v}(\boldsymbol{x})$). Finally, it uses a Newton–Krylov method to solve the reduced gradient system. The forward and the adjoint systems of the PDEs are discretized in $\Omega\times[0,1]$, $\Omega := [0,2\pi)^3 \subset \mathbb{R}^3$ using $N = N_1N_2N_3$ equispaced grid points x_{ijk}. `CLAIRE` uses a semi-Lagrangian scheme for the transport equations and FFT-based spectral differentiation in several places, which diagonalizes $\mathcal{A}$ (vector Laplacian). For other operators, we can replace this by different schemes, which we exploit in §3 to accelerate `CLAIRE`.

3 COMPUTATIONAL KERNELS

The main computational kernels for CLAIRE are the FFT used for spatial differential operators in our spectral approach and the interpolation in the semi-Lagrangian scheme for advection. The matrix-free Gauss–Newton Hessian matvec required in the Krylov solver (a preconditioned conjugate gradient method (**PCG**) in our case) involves solving (incremental) forward and adjoint hyperbolic PDEs. If we use $O(N_t)$ time steps, each Hessian matrix-vector multiplication

(**matvec**) requires $2N_t$ semi-Lagrangian steps, $2N_t$ gradient operators, and N_t divergence operators. In addition, the Hessian matvec needs the application of $\mathcal{A}$ and its inverse. All these operators have $O(N)$ complexity per time step, up to a logarithmic prefactor. The total number of Hessian matvecs is the sum of PCG iterations across Newton steps. Although `CLAIRE` has already highly optimized semi-Lagrangian and differentiation methods [20], ***we introduce several innovations*** in addition to the transition to GPUs: (i) we investigate several options for the interpolation; (ii) we replace all gradient and divergence operators with high-order finite-differences, which turn out to be faster than FFTs. FFTs are retained for $\mathcal{A}$ and its inverse; the GPU implementation of the proposed method employs a hybrid differentiation scheme that uses both FFTs and finite differences.

GPU Interpolation. The semi-Lagrangian scheme requires costly interpolation of velocities and scalar image fields along backward characteristics. `CLAIRE` uses Lagrange-based cubic interpolation. GPUs provide two technologies that we exploit in our schemes: texture fetches and hardware support for trilinear interpolation (although not fully single-precision). In addition to these modifications, we also consider another change, switching from Lagrange cubic to B-spline cubic interpolation.

GPU Derivatives. The CPU `CLAIRE` uses FFTs to perform spatial differentiation. Since our functions are periodic, all such operators are diagonal in the spectral domain. But in the proposed GPU implementation, we use an FD scheme that is more accurate (for the given resolutions) and faster than FFTs. In particular, we use an 8^{th} order central difference scheme to evaluate first-order partial derivatives for the gradient and divergence operators.

4 RESULTS

We evaluate the overall algorithm using four 3D MRI images. We study convergence behavior, time-to-solution, and registration accuracy for several algorithmic variants of computational kernels available in our new GPU implementation of the CPU software `CLAIRE`. We report results for the `NIREP` (''Non-Rigid Image Registration Evaluation Project'') data, a commonly used data set to evaluate the performance of deformable registration algorithms [6]. We resampled the data sets to grid sizes of 64^3, 128^3, 256^3, and 384^3 using a linear and a nearest-neighbor interpolation model for the image data and the label maps, respectively. Additionally, we compare the performance of the proposed method to two publicly available GPU implementations of LDDMM approaches. The first software package is `PyCA` [25]. `PyCA` uses gradient descent for optimization. The second software package is `deformetrica` [8]; `deformetrica` uses a limited-memory Broyden-Fletcher-Goldfarb-Shanno method for optimization. The gradient of the optimization problem is computed based on automatic differentiation [5]. We execute both registration packages for the three neuroimaging data sets we used to assess the performance of the proposed method (`na02`, `na03`, and `na10` as template images and `na01` as reference image. The runs for the comparison of the software packages are performed using the full resolution 256^3.

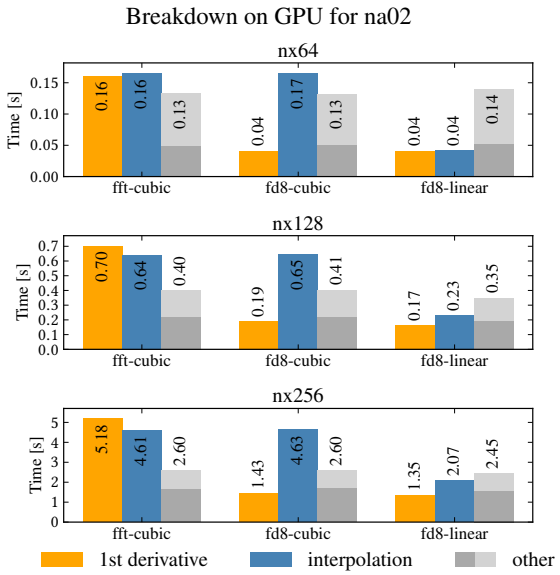

Figure 1: *Runtime for the main kernels of our method for all GPU implementations (first order derivatives via FFT or FD8, cubic or linear interpolation) on a single NVidia Tesla V100 (on a Power9 node with NVLink). We consider the registration of* `na02` *to* `na01` *at a resolution of* 64^3, 128^3, *and* 256^3, *respectively.*

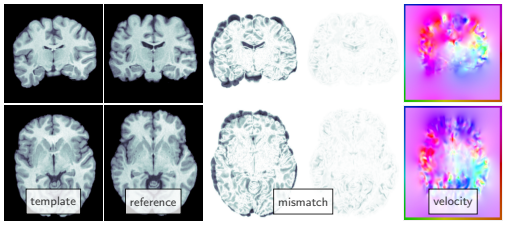

Figure 2: *3D registration results for the proposed method. Top row: coronal plane. Bottom row: axial plane. We show (from left to right) the template image (image to be registered), the reference image, the mismatch before and after regiatration, and the computed velocity (color denotes orientation).*

Table 1: *Performance for* `PyCA` *[25],* `deformetrica` *[8] (software* `deform`*), and our method (software* `CLAIRE`*) executed on a V100 and a P100 for three data sets (grid size:* 256^3*). We report iterations per level, the relative mismatch after registration (mism.), and the runtime (in seconds).*

data	software	**#iter**	**mism.**	**time** P100	**time** V100
na02	`PyCA`	100,50	4.2e−1	1.9e1	1.1e1
	`deform`	10	4.8e−1	1.4e2	–
	`CLAIRE`	14	2.7e−2	9.0	5.9
na03	`PyCA`	300,300	2.5e−1	1.0e2	5.4e1
	`deform`	50	3.1e−1	8.4e2	–
	`CLAIRE`	17	2.6e−2	1.1e1	7.2
na10	`PyCA`	300,300	2.5e−1	1.0e2	5.4e1
	`deform`	50	3.0e−1	8.3e2	–
	`CLAIRE`	17	2.0e−2	1.1e1	7.3

REFERENCES

[1] J. Ashburner. 2007. A fast diffeomorphic image registration algorithm. *NeuroImage* 38, 1 (2007), 95–113.

[2] B. B. Avants, C. L. Epstein, M. Brossman, and J. C. Gee. 2008. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Medical Image Analysis* 12, 1 (2008), 26–41.

[3] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee. 2011. A reproducible evaluation of ANTs similarity metric performance in brain image registration. *NeuroImage* 54 (2011), 2033–2044.

[4] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. 2005. Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International Journal of Computer Vision* 61, 2 (2005), 139–157.

[5] A. Bone, M. Louis, B. Martin, and S. Durrleman. 2018. Deformetrica 4: An open-source software for statistical shape analysis. In *Proc International Workshop on Shape in Medical Imaging*, Vol. LNCS 11167. 3–13.

[6] G. E. Christensen, X. Geng, J. G. Kuhl, J. Bruss, T. J. Grabowski, I. A. Pirwani, M. W. Vannier, J. S. Allen, and H. Damasio. 2006. Introduction to the non-rigid image registration evaluation project. In *Proc Biomedical Image Registration*, Vol. LNCS 4057. 128–135.

[7] N. Courty and P. Hellier. 2008. Accelerating 3D non-rigid registration using graphics hardware. *International Journal of Image and Graphics* 8, 1 (2008), 81–98.

[8] A. S. Durrleman, A. Bone, M. Louis, B. Martin, P. Gori, A. Routier, M. Bacci, A. Fougier, B. Charlier, J. Glaunes, J. Fishbaugh, M. Prastawa, M. Diaz, and C. Doucet. 2019. deformetrica [Commit: v4.0.0-390-ged9c1f9; Libraries: python3.6; CUDA9.2.88]. https://gitlab.com/icm-institute/aramislab/deformetrica

[9] S. Durrleman, M. Prastawa, N. Charon, J. R. Korenberg, S. Joshi, G. Gerig, and A. Trouve. 2014. Morphometry of anatomical shape complexes with dense deformations and sparse parameters. *NeuroImage* 101 (2014), 35–49.

[10] A. Eklund, P. Dufort, D. Forsberg, and S. M. LaConte. 2013. Medical image processing on the GPU–Past, present and future. *Medical Image Analysis* 17, 8 (2013), 1073–1094.

[11] B. Fischer and J. Modersitzki. 2008. Ill-posed medicine – an introduction to image registration. *Inverse Problems* 24, 3 (2008), 1–16.

[12] O. Fluck, C. Vetter, W. Wein, A. Kamen, B. Preim, and R. Westermann. 2011. A survey of medical image registration on graphics hardware. *Computer Methods and Programs in Biomedicine* 104, 3 (2011), e45–e57.

[13] D. Grzech, L. Folgoc, M. P. Heinrich, B. Khanal, J. Moll, J. A. Schnabel, B. Glocker, and B. Kainz. 2019. FastReg: Fast Non-Rigid Registration via Accelerated Optimisation on the Manifold of Diffeomorphisms. *arXiv e-prints* (2019). https://arxiv.org/abs/1903.01905

[14] X. Gu, H. Pan, Y. Liang, R. Castillo, D. Yang, D. Choi, E. Castillo, A. Majumdar, T. Guerrero, and S. B. Jiang. 2009. Implementation and evaluation of various demons deformable image registration algorithms on a GPU. *Physics in Medicine and Biology* 55, 1 (2009), 207–219.

[15] L. Ha, J. Krüger, S. Joshi, and C. T. Silva. 2011. Multiscale unbiased diffeomorphic atlas construction on multi-GPUs. In *CPU Computing Gems Emerald Edition*. Elsevier Inc, Chapter 48, 771–791.

[16] L. K. Ha, J. Krüger, P. T. Fletcher, S. Joshi, and C. T. Silva. 2009. Fast parallel unbiased diffeomorphic atlas construction on multi-graphics processing units. In *Proc Eurographics Conference on Parallel Grphics and Visualization*. 41–48.

[17] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. W. Pluim. 2010. ELASTIX: A tollbox for intensity-based medical image registration. *Medical Imaging, IEEE Transactions on* 29, 1 (2010), 196–205.

[18] A. Mang and G. Biros. 2016. Constrained H^1-regularization schemes for diffeomorphic image registration. *SIAM Journal on Imaging Sciences* 9, 3 (2016), 1154–1194.

[19] A. Mang and G. Biros. 2019. Constrained Large Deformation Diffeomorphic Image Registration (CLAIRE). https://andreasmang.github.io/claire [Commit: v0.07-131-gbb7619e].

[20] A. Mang, A. Gholami, and G. Biros. 2016. Distributed-memory large-deformation diffeomorphic 3D image registration. In *Proc ACM/IEEE Conference on Supercomputing*.

[21] A. Mang, A. Gholami, C. Davatzikos, and G. Biros. 2018. CLAIRE: A distributed-memory solver for constrained large deformation diffeomorphic image registration. *arXiv e-prints* 1808.04487 (2018).

[22] M. Modat, G. R. Ridgway, Z. A. Taylor, M. Lehmann, J. Barnes, D. J. Hawkes, N. C. Fox, and S. Ourselin. 2010. Fast free-form deformation using graphics processing units. *Computer Methods and Programs in Biomedicine* 98, 3 (2010), 278–284.

[23] J. Modersitzki. 2004. *Numerical methods for image registration*. Oxford University Press, New York.

[24] J. Modersitzki. 2009. *FAIR: Flexible algorithms for image registration*. SIAM, Philadelphia, Pennsylvania, US.

[25] J. S. Preston. 2019. Python for computational anatomy. https://bitbucket.org/scicompanat/pyca [Commit: v0.01-434-gf31ab43; Libraries: ITK4.13.2; boost1.69; FFTW3.3.6-pl2; python2.7; CUDA9.2.88].

[26] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. 1999. Non-rigid registration using free-form deformations: Application to breast MR images. *Medical Imaging, IEEE Transactions on* 18, 8 (1999), 712–721.

[27] R. Shams, P. Sadeghi, R. A. Kennedy, and R. I. Hartley. 2010. A survey of medical image registration on multicore and the GPU. *Signal Processing Magazine, IEEE* 27, 2 (2010), 50–60.

[28] A. Sotiras, C. Davatzikos, and N. Paragios. 2013. Deformable medical image registration: A survey. *Medical Imaging, IEEE Transactions on* 32, 7 (2013), 1153–1190.

[29] A. Trouvé. 1998. Diffeomorphism groups and pattern matching in image analysis. *International Journal of Computer Vision* 28, 3 (1998), 213–221.

[30] P. Valero-Lara. 2013. A GPU approach for accelerating 3D deformable registration (DARTEL) on brain biomedical images. In *Proc European MPI Users' Group Meeting*. 187–192.

[31] P. Valero-Lara. 2014. Multi-GPU acceleration of DARTEL (early detection of Alzheimer). In *Proc IEEE International Conference on Cluster Computing*. 346–354.

[32] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. 2009. Diffeomorphic demons: Efficient non-parametric image registration. *NeuroImage* 45, 1 (2009), S61–S72.

[33] L. Younes. 2010. *Shapes and diffeomorphisms*. Springer.