A Hamiltonian Particle Method for Diffeomorphic Image Registration

Stephen Marsland and Robert McLachlan

Massey University, Private Bag 11-222
Palmerston North, New Zealand
{s.r.marsland,r.mchlachlan}@massey.ac.nz

Abstract. Diffeomorphic image registration, where images are aligned using diffeomorphic warps, is a popular subject for research in medical image analysis. We introduce a novel algorithm for computing diffeomorphic warps that solves the Euler equations on the diffeomorphism group explicitly, based on a discretisation of the Hamiltonian, rather than using an optimiser. The result is an algorithm that is many times faster than those considered previously.

1 Introduction

Image registration has received much research over the past few years, not least because of its many applications in medicine. For example, it is useful for removing motion artefacts caused by patient breathing, heartbeat, and patient movement [1], aligning to an atlas [2], monitoring disease progression [3], assisting in disease diagnosis [4], and measuring anatomical variability between subjects [5]. For further details about these applications see [5]. A more general survey of image registration, highlighting its uses in synthetic aperture radar and other applications is given in [6].

For applications in disease diagnosis and measuring anatomical variability, some form of measurement on the space of images is essential, to allow statistical analysis of the image warps. This generally requires using diffeomorphic image registration, where the choice of image warps that can be used to solve the registration problem are constrained to be diffeomorphisms, i.e., smooth functions that have smooth inverses. There has therefore been recent interest in the use of diffeomorphic deformations (warps) to align medical images.

In 1966 Arnold made the profound discovery that the Euler equations for a perfect fluid are geodesic equations on the group of volume-preserving diffeomorphisms with respect to a group-invariant metric defined by the kinetic energy of the fluid [7]. This point of view allowed stability and existence results [8] that have not been bettered today. For diffeomorphic image warping, it is the group of *all* diffeomorphisms that is considered, and the warp ϕ is constructed as a geodesic (shortest path) between two images, leading to a right-invariant Riemannian metric. This diffeomorphism has typically been computed as an optimisation problem; see [9,4] for an overview.

N. Karssemeijer and B. Lelieveldt (Eds.): IPMI 2007, LNCS 4584, pp. 396–407, 2007.

In this paper we introduce a novel formulation of the problem that is based on solving the partial differential equations that govern the motion. These PDEs are the Euler equations for the full diffeomorphism group, given by equations (1) and (2) below; for derivations, following [7], see [9,10]. We introduce a particle method that enables us to solve for the diffeomorphism directly, resulting in an algorithm that is orders of magnitude faster than previous ones. We demonstrate the algorithm using standard forward-Euler and Runge-Kutta integrators, and discuss the benefits of using a symplectic integrator.

1.1 Problem Formulation

The aim of diffeomorphic image registration is to find a diffeomorphism ϕ that takes a free image F to a reference image R, i.e., $R = F \circ \phi$. The diffeomorphism ϕ is defined on some domain $\Omega \in \mathbb{R}^2$ or $\mathbb{R}^3$, and the images are typically greyscale, so that $R, F : \mathbb{R}^2 \to \mathbb{R}$ or $R, F : \mathbb{R}^3 \to \mathbb{R}$. The method used to find the desired ϕ is generally optimisation of some norm $\|R - F \circ \phi\|$. Typical choices include the L^2 norm (sum-of-squares error) and mutual information [11,12], although there are other alternatives, including the correlation ratio [13] and the normalised gradient-based method [14].

In this paper, we describe a novel method of constructing the diffeomorphisms. The standard approach is to use an energy minimisation, which produces the diffeomorphism as a geodesic [4,17,15,16]. Instead, we compute the Hamiltonian of the Euler equations on the diffeomorphism group, discretise them and integrate them explicitly. For the case of the full diffeomorphism group, $\mathfrak{G} = \mathrm{Diff}(\mathbb{R}^n)$, that we consider here, the Euler equations are (see [18,19] for further details):

$$\dot{m} + u \cdot \nabla m + \nabla u^T \cdot m + m(\mathrm{div}\, u) = 0, \tag{1}$$

where $\dot{m}$ denotes differentiation with respect to time, $u(x,t)$ $(u, x \in \mathbb{R}^n,\ t \in \mathbb{R})$ is a velocity field, and $m(x,t)$ its associated momentum. The velocity u and momentum m are related by:

$$m = \mathcal{A}u, \tag{2}$$

where $\mathcal{A}$ is an elliptic operator (e.g., $\mathcal{A} = (1 - \nabla^2)^k$) called the inertia operator. The inverse of $\mathcal{A}$ is given by convolution with the Green's function $\mathbf{G}$ of $\mathcal{A}$, i.e., $u = \mathbf{G} * m$, where $*$ denotes convolution and $\mathcal{A}\mathbf{G}(x, x') = \delta(x - x')$ for $x, x' \in \mathbb{R}^n$.

A striking feature of Euler equations on diffeomorphism groups is that they admit (formally, at least) exact solutions in which the momentum is concentrated at a finite set of points. For fluid equations these are point vortices, which are widely studied both in their own right and as a means of approximating the evolution of smooth or other vorticities [20,21]. In analogy with the point vortices of fluid dynamics, we call the image registration equivalents *point particles.*

2 A Particle Method for Image Registration

We are considering the deformation of an image Ω, with the deformation defined by a set of points i (some subset of the pixels of the image) with position and momentum $(q^i(t), p^i(t))$, where $p_i = \dot{q}_i$ as they move from their initial state (q_0^i, p_0^i) to their endpoints at $t = 1$. Starting from the Euler equations on the diffeomorphism group ((1) and (2)) we compute the Hamiltonian (see [22] for a derivation of the Hamiltonian from the Lagrangian via the Legendre transform), which is the kinetic energy, and then discretise it by introducing the particle ansatz $m(x,t) = \sum_{j=1}^{N} p_j(t)\delta(x - q_j(t))$, where $\delta(\cdot)$ is Kronecker delta function.The evolution of the particles is then given by the Hamiltonian:

$$H = \frac{1}{2}\sum_{i,j} p_i \cdot p_j G(q_i - q_j), \tag{3}$$

where $G(\cdot)$ is the Green's function corresponding to the chosen metric on Diff(Ω). The most common choice in image registration, and the one that we will use in this paper is the H^∞ metric, which corresponds to using a Gaussian Green's function $G(r) = \frac{1}{\epsilon^2}\exp(-r^2/\epsilon^2)$, where ϵ is the length-scale in the metric. Other choices include the thin-plate and clamped-plate splines – see [4] for a review.

Solutions to (1) of this form obey Hamilton's equations for (3), in which the components of q_i and p_i are canonically conjugate variables (see [23] for further details). Here $q_1, \ldots, q_N$ represent the positions of the N particles that define the deformation, and $p_1, \ldots, p_N$ their momenta. The equations of motion of the point particles are:

$$\dot{q}_i = \sum_{j=1}^{N} G(\|q_i - q_j\|)p_j, \tag{4}$$

$$\dot{p}_i = -\sum_{j} (p_i \cdot p_j) G'(\|q_i - q_j\|)\frac{q_i - q_j}{\|q_i - q_j\|}. \tag{5}$$

Computing the diffeomorphism defined by q_i and p_i is then simply a case of integrating the motion forward in time using (4) and (5), and then interpolating the motion of the rest of the image in some way. The integration requires fewer timesteps than the optimisation methods, enables the accuracy of the method to be computed explicitly, and is computationally significantly faster. This leads us to a description of our complete algorithm for image registration, following which we discuss several important implementation details.

Our Image Registration Algorithm

```
– Choose point particle positions $q$ on image $T$
– Initialise the particle momenta $p$ randomly
– Optimise $\|R - T \circ \phi\|$ over $p$:
    • For current $p$, integrate point particles forward in time
    • Integrate positions of the test particles
    • Interpolate between the test particles
    • Compute $\|R - T \circ \phi\|$ for chosen distance measure
– Add more point particles and iterate
```

Position of point particles. There are several possible choices for placing the point particles, such as placing them in a grid, positioning them on points of interest in the image (such as edges and corners), or using the discrepancy image [24] to select places where the two images do not match. In line with [24], for registration of brains, we initially place some points around the skull of head images and, after optimising them, place more points using the discrepancy image method. For the hand images shown in the next section, we use a uniform grid.

Initialisation of point momenta. In the current implementation, the momenta of the point particles are initialised with a uniform random direction, and with a small uniform random magnitude for the warp. One option that improves the results, although at a moderate computational cost, is to perform a coarse search over this relatively small number of parameters (2 for each of the point particles, of which there may be 10-20 on the initial pass).

Choice of integrator. The primary component of our method is the computation of the current geodesic, based on q and the current p. This is calculated by numerically integrating the particle dynamics forward in time using (4) and (5). We can choose a timestep for the integration, and the method of numerical integration. The standard choices would generally be Euler's method, or a second-order improvement, such as second-order Runge-Kutta.

The factors that affect the computation of the diffeomorphism are the number of point particles and test particles, the number of timesteps, and the order of the integrator (how errors accumulate during the integration). In consideration of the last two of these points, in section 5 we discuss the possible benefits and disadvantages of using a symplectic integrator, together with a possible reduction in the computational complexity of the algorithm.

Test particles and interpolation. We can induce the value of the actual diffeomorphism $\phi(x)$ by the current geodesic on each pixel by placing test particles with zero momentum (so $q(0) = x, p(0) = 0$) on the pixels, and computing their trajectories under the induced velocity field (i.e., solving the ODE $\dot{q} = f(q, t)$). Assuming that the deformation is not too large ($\|T\phi - 1\|$ is small), we can make some computational savings by placing a test particle every k pixels, and interpolating ϕ between them. This saves a factor of k^2 computations, but changes the computed diffeomorphism from the exact one that relates to the flow (it may actually stop the warp being diffeomorphic, although this does not seem to be a problem in general). We have found that using $k = 4$ and bi-linear interpolation has negligible effect on the accuracy in real registrations, as is demonstrated in the table on the left of Fig. 3.

Choice of metric. Inherent in the choice of Green's function $G(r)$ is a choice of the metric under which the particle dynamics occur. There is complete freedom of choice over this metric. By far the most common choice to date for image registration has been to use a Gaussian metric, i.e., Green's function $G(r) = \frac{1}{\epsilon^2}\exp(-r^2/\epsilon^2)$, where ϵ is the length-scale in the metric. The role of this length-scale is important. If it is set too small (say smaller than the pixel spacing) then the kernels will not overlap, and the movement of each particle will be entirely independent of the rest of the image. This will require the number of point particles to tend to infinity to represent an arbitrary diffeomorphism. We do not consider how to choose the length-scale in this paper, but it may be that starting with a large value of ϵ and allowing it to shrink is a useful method of iteratively refining the solution.

The Gaussian is by no means the only possible choice of metric. One fairly general formulation, which includes the Gaussian as the limit as $k \to \infty$, are the H^k metrics, $(1 - \epsilon^2\nabla^2)^k$; see [25] for a discussion of these. Finally, it may well be useful to choose the metric so that it vanishes on some set of motions that are not important. Examples could be affine or rigid motions.

Optimisation method. The choice of a suitable optimiser is obviously crucial, together with the choice of objective function for the image matching. In the current implementation we are using the sum-of-squares distance measure, which leads fairly naturally to a least-squares non-linear optimiser. We use the `lsqnonlin` function in Matlab 7.1, which is a subspace trust region method based on the interior-reflective Newton method. Experimentation has found that allowing 100 iterations is usually more than sufficient for the algorithm to converge, although further work will investigate this more thoroughly.

Adding more points. In our implementation we position new point particles for further levels of optimisation using the discrepancy image. This uses the objective function (here the sum-of-squares error) to find regions where the two images do not match, and then places new point particles there. We tested initialising the momenta of these particles as either zero, or small random numbers, and found that the first was the most effective. This is not surprising, because points with zero momentum are carried along with the flow, which is a reasonable initial guess for how they should behave, and the optimiser then improves on this.

3 Experiments

We present four main experiments in this paper. In all of them, the image is scaled into $[-1, 1]^2$ and a value of $\epsilon = 1$ was used. The first experiment considers how far apart the spacing should be between the test particles – the wider apart, the faster the implementation, but the less accurate the approximation to the true diffeomorphism. In order to decide a suitable spacing, we took a series of 10 registrations of hands, as used for the registration shown in Fig. 2 and described below, and tested out different spacings between the test particles for two different numbers of point particles (with the initial values for the momenta of the point particles fixed between the runs). The average results over the 10 registrations are shown in the table on the left of Fig. 1, and show that a spacing of 4 between test particles provides a reasonable compromise between computational time and final function value, hence we have used a spacing of 4 for all the computations used in this paper.

For the second experiment, we investigated how the performance of the integrators change as the number of timesteps is varied. The right of Fig.1 shows

Spacing	Time (s)	Final function value
9 point particles		
12	39.13	7.78e7
8	43.13	7.43e7
4	47.71	7.08e7
1	112.6	6.95e7
25 point particles		
12	123.17	1.06e8
8	133.95	9.82e7
4	163.05	9.37e7
1	944.72	8.85e7

Timesteps	1	2	4	8
Small perturbation, initial error 60.2%				
Integrator				
Euler	5.2%	3.6%	2.9%	2.5%
RK2	4.3%	2.2%	2.2%	2.2%
Large perturbation, initial error 84%				
Euler	16.3%	10.3%	8.6%	8.1%
RK2	4.3%	5.9%	6.4%	6.7%

Fig. 1. *Left:* Comparison of changing the spacing between the test particles. Results are the average of 10 values. A spacing of 4 appears to give a reasonable compromise between computational cost and the final diffeomorphism. *Right:* The effect of changing the number of timesteps (errors relative to reference image). For RK2 the integration error is far below other sources, but for Euler it is significant for large perturbations.

the results for registrations with 9 point particles using the `peaks(40)` function in Matlab to make the reference image, with the free image being the same image with a rotation applied. The second-order method needs fewer timesteps than the first-order forward-Euler. Indeed, adding more timesteps can make the second-order results worse. The reason for this is currently under investigation.

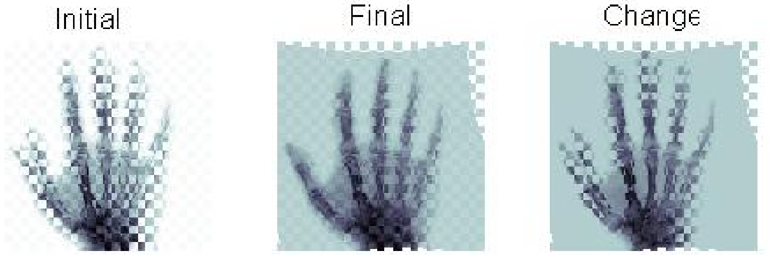

Fig. 2. Chequer-board plots showing the difference between the initial images of the hands (left), the final images (centre), and the change between the initial and final versions of the free image (right).

We now present two different image registrations. The first is of a pair of hand images, while the second are two 2D T1-weighted MR scans of the human brain. The hand results shown in Fig. 2 were computed using 9 knotpoints, positioned in a 3×3 grid on the image. The optimiser ran for 40 iterations before converging, and then an additional 7 points were added to the image using the discrepancy image. It can be seen that even after this relatively small amount of computation, the registration is very good. Computing this registration took 251.4 seconds on a 1.8GHz G5 Apple Macintosh. Another 37 iterations were then performed by

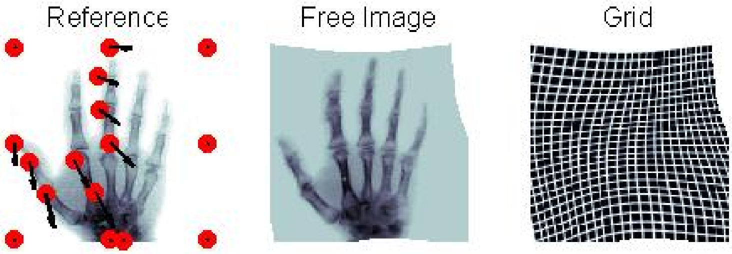

Fig. 3. The registration of the two hands. The reference image, together with the positions of the particles and their momenta are shown on the left, the final result is shown in the middle, and the effect of the warp on a grid is shown on the right.

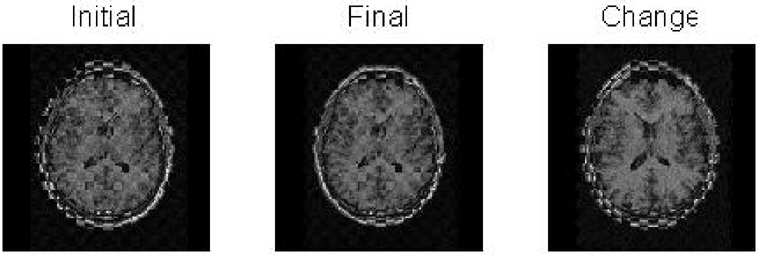

Fig. 4. Chequer-board plots showing the different between the initial images of the T1-weighted brains(left), the final images (centre), and the change between the initial and final versions of the free image (right). The registration has lined up the skulls and the major structures within the brain, but there is still more fine-scale work to be done.

the optimiser, with the final result being that shown. Fig. 3 shows the positions of the points and the initial momenta on the reference image, the final output, and the effect of the warp on a regular grid.

Fig. 4 show a sample registration of 2 brains. A set of 10 points were positioned evenly around the skull, and the result optimised for 20 iterations. Following this, an additional set of 11 knotpoints were added, with 50 iterations of optimisation then being performed. This registration took under 7 minutes on the same computer, and it can be see that the final result is not bad. There is still work to be done on the interior (and further optimisations do indeed correct this), but the skull and major structures have all been brought into alignment.

These results are much faster than using an optimisation method for finding the diffeomorphism – the method described in [4] took just under 2 hours to perform the brain registration described above. One of the main reasons for this is that those optimisation methods take many more timesteps to find the diffeomorphism, usually 20 timesteps are used to guarantee a diffeomorphism. With our current method, for relatively small deformations, only 1-4 timesteps are needed.

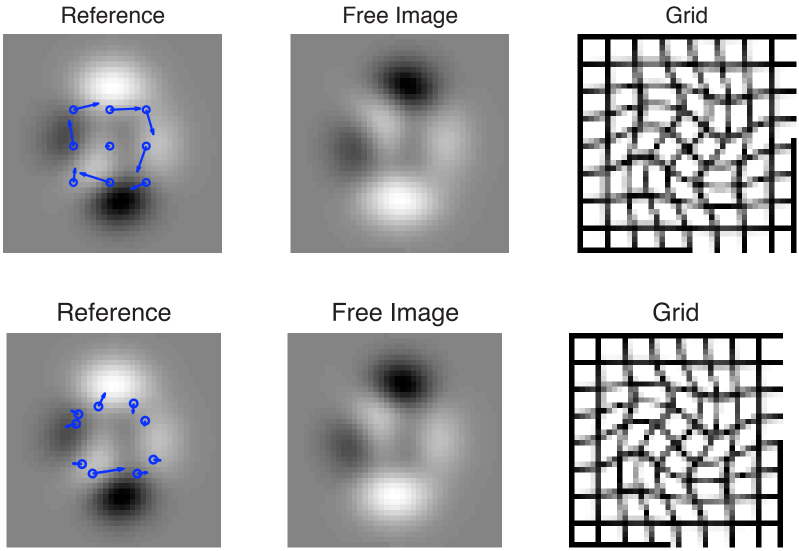

Fig. 5. Two regimes of optimisation. *Top:* Optimatisation of particle momenta only. *Bottom:* Simultaneous optimisation of both particle locations and momenta. The particles form into a ring showing the rotation that occurred.

4 Optimising Positions and Momenta

In the experiments described above we chose locations for the point particles, and then optimised their momenta. However, there is no reason why one cannot optimise the point locations as well as their momenta. We have performed some initial experiments with this based on the images produced by the `peaks(40)` function in Matlab. A typical result is shown in Fig. 5. The free image is a rotated version of the reference image (note that no affine registration is performed). The top line shows the optimisation if only the momenta of the particles are optimised, not their positions, while the bottom line shows the results when both are optimised, starting from the points being arranged on the uniform grid. It can be seen that the particle locations move to form a circle that reflects the rotation that was applied. Averaged over registrations of 10 random rotations, the final objective function value after 20 optimisation iterations was 5.85 for the momenta-only optimisations, and 1.59 for the full optimisation (the initial objective function value is around 2000).

The problem is that the method becomes more prone to becoming stuck in poor local minimas as the complexity of the image grows, since the dimensionality of the optimisation is so large. When it works it does very well, and resolves smaller image features, but for medical images it is only successful about half of the time. Finding suitable implementation methods to get around this problem, possibly using multiple scales of resolution, is a current area of research.

5 Use of Symplectic Integrators

The equations of motion (4, 5) are Hamiltonian and their flow is therefore symplectic [26]. In long-time simulations of Hamiltonian systems (in celestial and molecular mechanics, for example) it has been found extremely advantageous to use symplectic integrators, which preserve the symplectic structure. This leads to good energy behaviour and a lack of dissipation. Therefore, it is natural to consider their use here; it is also in accord with the 'Discrete Mechanics and Optimal Control' philosophy in which both the cost function and dynamics are discretized in a parallel, Hamiltonian way [27]. In fact, some implementations of image registration by diffeomorphisms have used symplectic integrators, because calculating geodesics by minimizing a discrete path length gives such an integrator [4]. However, the diffeomorphism itself, calculated from the motion of the test particles, has never been done symplectically. We give a preliminary analysis of the cost and benefits of using a symplectic integrator in image registration.

At first sight, the cost is a problem. The cheapest, explicit symplectic integrators apply to separable Hamiltonians of the form $T(p) + V(q)$; Eq. (3) is not separable, so only implicit symplectic integrators, notably the Gaussian Runge-Kuttas [26], are available; these methods have unconditional stability for linear problems, which allows larger time-steps to be used. These involve solving a set of equations for s internal stages; when $s = 1$, we have the midpoint rule

$$x_{k+1} = x_k + \Delta t f(\bar{x}_k), \quad \bar{x}_k = (x_k + x_{k+1})/2.$$

Moreover, to ensure exact symplecticity and that the solution varies smoothly with respect to the initial conditions, the equations must be solved extremely accurately, generally down to round-off error. In most situations, it is best to simply solve the equations by iteration

$$x_{k+1}^{l+1} = x_k + \Delta t f((x_k + x_{k+1}^{l})/2), \quad l = 0, 1, 2, \ldots$$

after choosing some initial guess x_{k+1}^0. If m iterations are required then the cost per time step is ms times the cost of Euler's method. In initial value problems with a large time step, as we want to use here, m can be quite large, say 5–15.

However, this cost penalty for initial value problems vanishes for optimization problems, in which we want to repeatedly solve the same initial value problem for a sequence of nearby initial values. We simply store the internal stage values as part of the orbit segment and use this as initial guesses (e.g. x_{k+1}^0 for the midpoint rule) when the initial conditions are changed. Most optimization algorithms estimate the derivatives of the objective function using finite differences, which requires repeatedly altering the initial conditions by about 10^{-6}; for these evaluations we can solve the implicit equations in a *single* iteration. The error constants of the Gaussian Runge-Kutta methods are extremely small so we expect that this method could be superior both for cost and accuracy.

For the point particles, the simple iteration (5) can be improved using the $4N \times 4N$ Jacobian derivative matrices of f; calculating these is essentially cost-free, because the entries are simply related to the Green's functions, which have already been evaluated. Newton's method would cost $\mathcal{O}(N^3)$, which is presumably prohibitive, but the modifed iteration

$$\bar{x}^l = (x_k + x_{k+1}^l)/2, \ w^l = x_k + \Delta t f(\bar{x}^l) \tag{6}$$

$$x_k^{l+1} = w^l + \frac{1}{2}\Delta t f'(\bar{x}^l)(w^l - x_k^l) \tag{7}$$

converges much more quickly than (5) and still costs only $\mathcal{O}(N^2)$. Moreover, the derivatives of the stage values with respect to the initial momenta can be approximated in a similar way, giving excellent initial guesses. Experiments will determine whether this cost is justified.

For very large numbers of point particles, the cost $\mathcal{O}(N^2+NM)$ of evaluating the vector field may be too expensive. The cost can be reduced to $\mathcal{O}(N+M)$ using the marker-and-cell method [28], while still using symplectic integrators for the particle paths [29]. A regular grid with $\mathcal{O}(N)$ grid points is laid over the domain and the particle momenta interpolated to the grid. Then the velocity field induced by the momentum field is calculated on the grid using a fast algorithm, such as multigrid ($\mathcal{O}(N)$) or Fourier transform ($\mathcal{O}(N \log N)$). This velocity field is interpolated back to the particle positions, which are then updated. This algorithm has been implemented with enormous numbers (more than 1 million) particles in an initial value problem in atmospheric dynamics [29]. However, very large numbers of point particles, which may well be required for convergence to an arbitrary diffeomorphism, will introduce new difficulties for the optimization has the problem has now become ill-posed. Some degree of regularization, enforcing smoothness of the initial momentum field, will be required.

6 Conclusions and Open Questions

We have presented a method of performing diffeomorphic image registration that has links to the methods of discrete mechanics and optimal control. The implementation described in this paper has been demonstrated to perform high quality registrations in reasonably short computational time – orders of magnitude less than using energy minimisation methods. While they are not necessary for all image registration tasks, for applications where it is *variation* that is of interest, for example in disease diagnosis or measurement of anatomical variability, the access to a right-invariant Riemannian metric on the diffeomorphism group makes diffeomorphic registration methods essential.

There are a great many unanswered questions and areas for future research. We are particularly interested in the dynamical behaviour of the Euler equations on the diffeomorphism group, and how it relates to point vortices in fluid dynamics, which act on the volume-preserving subgroups. Some of our work on these topics is available in [25,30].

However, with regard to using the method for image registration, there are also several areas for further work. Firstly, we are currently investigating the use of the midpoint rule symplectic integrator and the marker-and-cell method, as discussed in section 5, and a second question that we highlighted earlier in the paper is that of a suitable choice of metric and corresponding length-scale.

Acknowledgements

We are grateful to Jan Modersitzki for the use of the hand images and to the Royal Society of New Zealand Marsden fund and NZIMA for their financial support.

References

1. Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L.G., Leach, M.O., Hawkes, D.J.: Non-rigid registration using free-form deformations: Application to breast MR images. IEEE Transactions on Medical Imaging 18(8), 712–721 (1999)
2. Gee, J., Reivich, M., Bajcsy, R.: Elastically deforming 3D atlas to match anatomical brain images. Journal of Computer Assisted Tomography 17(2), 225–236 (1993)
3. Thompson, P., Hayashi, K.M., Sowell, E.R., Gogtay, N., Giedd, J.N., Rapoport, J.L., de Zubricara, C., Janke, G.A.L., Rose, S.E., Semple, J., Doddrell, D.M., Wang, Y., van Erp, T.G., Cannon, T.D., Toga, A.W.: Mapping cortical change in alzheimer's disease, brain development, and schizophrenia. NeuroImage 23, S2–S18 (2004)
4. Marsland, S., Twining, C.: Constructing diffeomorphic representations for the groupwise analysis of non-rigid registrations of medical images. IEEE Transactions on Medical Imaging 23(8), 1006–1020 (2004)
5. Toga, A.W.: Brain Warping. Academic Press, San Diego (1999)
6. Zitová, B., Flusser, J.: Image registration methods: A survey. Image and Vision Computing 21, 977–1000 (2003)
7. Arnold, V.I.: Sur la géométrie différentielle des groupes de Lie de dimension infinie et ses applications hydrodynamique des fluides parfaits. Annales de L'Institut Fourier (Grenoble) 16(1), 319–361 (1966)
8. Ebin, D.G., Marsden, J.E.: Groups of diffeomorphisms and the motion of an incompressible fluid. Annals of Mathematics 92, 102–163 (1970)
9. Miller, M., Trouvé, A., Younes, L.: On metrics and the Euler-Lagrange equations of computational anatomy. Annual Reviews in Biomedical Eng. 4, 375–405 (2002)
10. Holm, D.D., Ratnanather, J.T., Trouvé, A., Younes, L.: Soliton dynamics in computational anatomy. NeuroImage 23(Suppl. 1), S170–S178 (2004)
11. Viola, P., Wells III, W.M.: Alignment by maximization of mutual information. International Journal of Computer Vision 24(2), 137–154 (1997)
12. Maes, F., Collignon, A., Vandermeulun, D., Marchal, G., Suetens, P.: Multimodality image registration by maximization of mutual information. IEEE Transactions on Medical Imaging 16(2), 187–198 (1997)
13. Roche, A., Malandain, G., Pennec, X., Ayache, N.: The correlation ratio as a new similarity measure for multimodal image registration. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 1115–1124. Springer, Heidelberg (1998)
14. Haber, E., Modersitzki, J.: Beyond mutual information: A simple and robust alternative. In: Bildverarbeitung für die Medizin, Springer, Heidelberg (2005)
15. Joshi, S.C., Miller, M.M.: Landmark matching via large deformation diffeomorphisms. IEEE Transactions on Image Processing 9(8), 1357–1370 (2000)
16. Beg, M.F., Miller, M.I., Trouv'e, A., Younes, L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. International Journal of Computer Vision 61(2), 139–157 (2005)

17. Camion, V., Younes, L.: Geodesic interpolating splines. In: Figueiredo, M., Zerubia, J., Jain, A.K. (eds.) EMMCVPR 2001. LNCS, vol. 2134, pp. 513–527. Springer, Heidelberg (2001)
18. Holm, D.D., Marsden, J.E., Ratiu, T.S.: The Euler-Poincaré equations and semidirect products, with applications to continuum theories. Advances in Mathematics 137(1), 1–81 (1998)
19. Mumford, D.: Pattern theory and vision. In: Questions Mathématiques En Traitement Du Signal et de L'Image. Institute Henri Poincaré, Paris, pp. 7–13 (1998)
20. Leonard, A.: Vortex methods for flow simulation. Journal of Computational Physics 37, 289–305 (1980)
21. Beale, J.T., Majda, A.: Vortex methods. II: Higher order accuracy in two and three dimensions. Mathematics of Computation 39(159), 29–52 (1982)
22. Mills, A., Marsland, S., Shardlow, T.: Computing the geodesic interpolating spline. In: Pluim, J.P.W., Likar, B., Gerritsen, F.A. (eds.) WBIR 2006. LNCS, vol. 4057, pp. 169–177. Springer, Heidelberg (2006)
23. Marsden, J., Ratiu, T.: Introduction to Mechanics and Symmetry: A Basic Exposition of Classical Mechanical Systems, 2nd edn. Springer, Berlin (1999)
24. Marsland, S., Twining, C.: Constructing data-driven optimal representations for iterative pairwise non-rigid registration. In: Gee, J.C., Maintz, J.B.A., Vannier, M.W. (eds.) WBIR 2003. LNCS, vol. 2717, pp. 50–60. Springer, Heidelberg (2003)
25. McLachlan, R.I., Marsland, S.: N-particle dynamics of the Euler equations for planar diffeomorphisms. Dynamical Systems. (In press) (2007)
26. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations. Springer, Berlin (2002)
27. Junge, O., Marsden, J.E., Ober-Blbaum, S.: Discrete mechanics and optimal control. In: Proceedings of International Federation of Automatic Control Conference 2005 (2005)
28. Harlow, F.H., Welsh, J.E.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. Physics of Fluids 8, 2182–2189 (1965)
29. Frank, J., Gottwald, G., Reich, S.: A Hamiltonian particle-mesh method for the rotating shallow-water equations. In: Siefkes, D. (ed.) Meshfree methods for partial differential equations. Lecture Notes in Computer Science Engineering, vol. 26, pp. 131–142. Springer, Berlin (2003)
30. McLachlan, R.I., Marsland, S.: The Kelvin-Helmholtz instability of momentum sheets in the Euler equations for planar diffeomorphisms. SIAM Journal on Applied Dynamical Systems 5(4), 726–758 (2006)