ACCELERATED OPTIMIZATION IN THE PDE FRAMEWORK: FORMULATIONS FOR THE MANIFOLD OF DIFFEOMORPHISMS

GANESH SUNDARMOORTHI*, ANTHONY YEZZI†, AND MINAS BENYAMIN‡

Abstract. We consider the problem of optimization of cost functionals on the infinite-dimensional manifold of diffeomorphisms. We present a new class of optimization methods, valid for any optimization problem setup on the space of diffeomorphisms by generalizing Nesterov accelerated optimization to the manifold of diffeomorphisms. While our framework is general for infinite dimensional manifolds, we specifically treat the case of diffeomorphisms, motivated by optical flow problems in computer vision. This is accomplished by building on a recent variational approach to a general class of accelerated optimization methods by Wibisono, Wilson and Jordan [63], which applies in finite dimensions. We generalize that approach to infinite dimensional manifolds. We derive the surprisingly simple continuum evolution equations, which are partial differential equations, for accelerated gradient descent, and relate it to simple mechanical principles from fluid mechanics. Our approach has natural connections to the optimal mass transport problem. This is because one can think of our approach as an evolution of an infinite number of particles endowed with mass (represented with a mass density) that moves in an energy landscape. The mass evolves with the optimization variable, and endows the particles with dynamics. This is different than the finite dimensional case where only a single particle moves and hence the dynamics does not depend on the mass. We derive the theory, compute the PDEs for accelerated optimization, and illustrate the behavior of these new accelerated optimization schemes.[1]

AMS subject classifications. 35B35, 49M99, 35J20, 35R30, 53C99, 65M99

1. Introduction.

Accelerated optimization methods have gained wide applicability within the machine learning and optimization communities (e.g.,[12, 21, 23, 26, 27, 28, 31, 34, 43, 44, 42]). They are known for leading to optimal convergence rates among schemes that use only gradient (first order) information in the convex case. In the non-convex case, they appear to provide robustness to shallow local minima. The intuitive idea is that by considering a particle with mass that moves in an energy landscape, the particle will gain momentum and surpass shallow local minimum and settle in in more wider, deeper local extrema in the energy landscape. This property has made them (in conjunction with stochastic search algorithms) particularly useful in machine learning, especially in the training of deep networks, where the optimization is a non-convex problem that is riddled with local minima. These methods have so far only been used in optimization problems that are defined in finite dimensions. In this paper, we consider the generalization of these methods to infinite dimensional manifolds. We are motivated by applications in computer vision, in particular, segmentation, 3D reconstruction, and optical flow. In these problems, the optimization is over infinite dimensional geometric quantities (e.g., curves, surfaces, mappings), and so the problems are formulated on infinite dimensional manifolds. Recently there has been interest within the machine learning community in optimization on finite dimensional manifolds, such as matrix groups, e.g.,[69, 35, 25], which have particular structure not available on infinite dimensional manifolds that we consider here.

Recent work [63] has shown that the continuum limit of accelerated methods, which are discrete optimization algorithms, may be formulated with variational principles. This allows one to derive the continuum limit of accelerated optimization methods (Nesterov's optimization method [42] and others) as an optimization problem on descent paths. The resulting optimal continuum path is defined by an ODE, which when discretized appropriately yields Nesterov's method and other accelerated optimization schemes. The optimization problem on paths is an action integral, which integrates the Bregman Lagrangian. The Bregman Lagrangian is a time-explicit Lagrangian (from physics) that consists of kinetic and potential energies. The kinetic energy is defined using the Bregman divergence (see Section 2.2); it is designed for finite step sizes, and thus differs from classical action integrals in physics [3, 37]. The potential energy is the cost function that is to be optimized.

We build on the approach of [63] by formulating accelerated optimization with an action integral, but we generalize that approach to infinite dimensional manifolds. Our approach is general for infinite dimensional manifolds, but we illustrate the idea here for the case of the infinite dimensional manifold of diffeomorphisms of $\mathbb{R}^n$ (the case of the manifold of curves has been recently formulated by the authors [68],[67],[68]). To do this, we abandon the Bregman Lagrangian framework in [63] since that assumes that the variable over which one optimizes is embedded in $\mathbb{R}^n$.

*Raytheon Technologies Research Center (ganesh.sundaramoorthi@rtx.com).

†Department of Electrical and Computer Engineering, Georgia Institute of Technology (ayezzi@ece.gatech.edu).

‡Department of Electrical and Computer Engineering, Georgia Institute of Technology (minasbenyamin@gmail.com).

[1]This work was by the Army Research Lab Grant ARL W911NF-18-1-0281 and NIH Grant R01-HL-143350. A conference version of this paper entitled *Variational PDEs for Acceleration on Manifolds and Application to Diffeomorphisms* [54] was published in the 2018 Conference on Neural Information Processing Systems (NIPS)

Instead, we adopt the classical formulation of action integrals in physics [3, 37], which is already general enough to deal with manifolds, and kinetic energies that are defined through general Riemannian metrics rather than a traditional Euclidean metric, thus by-passing the need for the use of Bregman distances. Our approach requires consideration of additional technicalities beyond that of [63] and classical physics. Namely, in finite dimensions in $\mathbb{R}^n$, one can think of accelerated optimization as a single particle with mass moving in an energy landscape. Since only a single particle moves, mass is a fixed constant that does not impact the dynamics of the particle. However, in infinite dimensions, one can instead think of an infinite number of particles each moving, which is modeled as a *mass density*. In the case of the manifold of diffeomorphisms of $\mathbb{R}^n$, we endow $\mathbb{R}^n$ with this mass density (see Figure 1). This mass density is introduced as part of the optimization and impacts the dynamics; it does not directly relate to the argument of the energy/loss function, i.e., the diffeomorphism. As the diffeomorphism evolves to optimize the cost functional, it deforms $\mathbb{R}^n$ and redistributes the mass, and so the density changes in time. Since the mass density defines the kinetic energy and the stationary action path depends on the kinetic energy, the dynamics of the evolution to minimize the cost functional depends on the way that mass is distributed in $\mathbb{R}^n$. Therefore, in the infinite dimensional case, one also needs to optimize and account for the mass density, which cannot be neglected. Further, our approach, due to the infinite dimensional nature, has evolution equations that are PDEs rather than ODEs in [63]. Finally, the discretization of the resulting PDEs requires the use of entropy schemes [50] since our evolution equations are defined as viscosity solutions of PDEs, required to treat shocks and rarefaction fans. These phenomena appear not to be present in the finite dimensional case.

1.1. Contributions.

Our contributions are specifically, **1**. We present a novel variational approach to accelerated optimization on manifolds. and adapt our approach to accelerated optimization on the infinite dimensional manifold diffeomorphisms, i.e., smooth invertible mappings. **2**. We introduce a Riemannian metric for the purpose of acceleration on diffeomorphisms, which defines the kinetic energy of a mass distribution. The metric is the same one in the fluid mechanics formulation of the L^2 mass transport problem [7]. **3**. We derive the PDE for accelerated optimization of any cost functional defined on diffeomorphisms, and relate it to fluid mechanics principles. **4**. We present numerical discretizations, for both Eulerian and Lagrangian formulations of accelerated optimization on diffeomorphisms, and show the advantage over gradient descent and competing methods.

Contributions over conference version of paper: A conference version of this manuscript [54] entitled "Variational PDEs for Acceleration on Manifolds and Application to Diffeomorphisms" was published in the journal of neural information processing systems in 2018. This work represents an expansion of the initial paper. The additional contributions are **1)** We provide a Lagrangian formulation of accelerated PDE optimization on the manifold of diffeomorphisms, as opposed to the Eulerian formulation in the original formulation. This shows that accelerated PDE on diffeomorphisms constitute a wave equation. This gives an additional justification as to why the accelerated scheme out-performs gradient descent in speed: the CFL conditions to the wave equation are more generous compared to gradient PDE. The formulation also allows a simple numerical scheme in the case of the particular mass model analyzed in this paper. **2)** We derive the numerical method for the aforementioned Lagrangian formulation. **3)** We benchmark our method on the Middlebury optical flow data set [4], and compare against a comparable general purpose variational optimizer. We show a speed advantage against that optimizer.

1.2. Related Work.

1.2.1. Sobolev Optimization. Our work is motivated by Sobolev gradient descent approaches [55, 6, 14, 57, 15, 56, 58, 38, 53, 65] for optimization problems on manifolds, which have been used for segmentation and optical flow problems. These approaches are general in that they apply to non-convex problems, and they are derived by computing the gradient of a cost functional with respect to a Sobolev metric rather than an L^2 metric typically assumed in variational optimization problems. The resulting gradient flows have been demonstrated to yield coarse-to-fine evolutions, where the optimization automatically transitions from coarse to successively finer scale deformations. This makes the optimization robust to local minimizers that plague L^2 gradient descents. We should point out that the Sobolev metric is used beyond optimization problems and have been used extensively in shape analysis (e.g., [30, 40, 39, 5]). While such gradient descents are robust to local minimizers, computing them in general involves an expensive computation of an inverse differential operator at each iteration of the gradient descent. In the case of optimization problems on curves and a very particular form of a Sobolev metric this can be made computationally fast [57], but the idea does not generalize beyond curves. In this work, we aim to obtain robustness properties of Sobolev gradient flows, but without the expensive computation of inverse operators. Our accelerated approach involves

averaging the gradient across time in the descent process, rather than an averaging across space in the Sobolev case. Despite our goal of avoiding Sobolev gradients for computational speed, we should mention that our framework is general to allow one to consider accelerated Sobolev gradient descents (although we do not demonstrate it here), where there is averaging in both space and time. This can be accomplished by changing the definition of kinetic energy in our approach. This could be useful in applications where added robustness is needed but speed is not a critical factor.

1.2.2. Optimal Mass Transport. Our work relates to the literature on the problem of *optimal mass transportion* (e.g., [60, 22, 2, 48]), especially the formulation of the problem in [7]. The modern formulation of the problem, called the Monge-Kantorovich problem, is as follows. One is given two probability densities ρ_0, ρ_1 in $\mathbb{R}^n$, and the goal is to compute a transformation $M : \mathbb{R}^n \to \mathbb{R}^n$ so that the pushforward of ρ_0 by M results in ρ_1 such that M has minimal cost. The cost is defined as the average Euclidean norm of displacement: $\int_{\mathbb{R}^n} |M(x) - x|^p \rho_0(x)\,\mathrm{d}x$ where $p \geq 1$. The value of the minimum cost is a distance (called the L^p Wasserstein distance) on the space of probability measures. In the case of $p = 2$, the transport M can be shown to the gradient of a scalar function [29, 1]. In the case that $p = 2$, [7] has shown that mass transport can be formulated as a fluid mechanics problem. In particular, the Wasserstein distance can be formulated as a distance arising from a Riemannian metric on the space of probability densities. The cost can be shown equivalent to the minimum Riemannian path length on the space of probability densities, with the initial and final points on the path being the two densities ρ_0, ρ_1. The tangent space is defined to be velocities of the density that infinitesimally displace the density. The Riemannian metric is just the kinetic energy of the mass distribution as it is displaced by the velocity, given by $\int_{\mathbb{R}^n} \frac{1}{2}\rho(x)|v(x)|^2\,\mathrm{d}x$. Therefore, optimal mass transport computes an optimal *path* on densities that minimizes the integral of kinetic energy along the path.

In our work, we seek to minimize a potential on the space of diffeomorphisms, with the use of acceleration. We can imagine that each diffeomorphism is associated with a point on a manifold, and the goal is to move to the bottom of the potential well. To do so, we associate a mass density in $\mathbb{R}^n$, which as we optimize the potential, moves in $\mathbb{R}^n$ via a push-forward of the evolving diffeomorphism. We regard this evolution as a path in the space of diffeomorphisms that arises from an action integral, where the action is the difference of the kinetic and potential energies. The kinetic energy that we choose, purely to endow the diffeomorphism with acceleration, is the same one used in the fluid mechanics formulation of optimal mass transportation for $p = 2$. We have chosen this kinetic energy for simplicity to illustrate our method, but we envision a variety of kinetic energies can be defined to introduce different dynamics. The main difference of our approach to the fluid mechanics formulation of mass transport is in the fact that we do not minimize just the path integral of the kinetic energy, but rather we derive our method by computing stationary paths of the path integral of kinetic minus *potential* energies. Since diffeomorphisms are generated by smooth velocity fields, we equivalently optimize over velocities. We also optimize over the mass distribution. Thus, the main difference between the fluid mechanics formulations of L^2 mass transport and our approach is the potential on diffeomorphisms, which is used to define the action integral.

1.2.3. Diffeomorphic Image Registration. Our work relates to the literature on diffeomorphic image registration [6, 41, 19, 20], where the goal, similar to ours, is to compute a registration between two images as a diffeomorphism. There a diffeomorphism is generated by a path of smooth velocity fields integrated over time. Rather than formulating an optimization problem directly on the diffeomorphism, the optimization problem is formed on a path of velocity fields. The optimization problem is to minimize $\int_0^1 \|v\|^2\,\mathrm{d}t$ where v is a time varying vector field, $\|\cdot\|$ is a norm on velocity fields, and the optimization is subject to the constraint that the mapping ϕ maps one image to the other, i.e., $I_1 = I_0 \circ \phi^{-1}$. The minimization can be considered as the minimization of an action integral where the action contains only a kinetic energy. The norm is chosen to be a Sobolev norm to ensure that the generated diffeomorphism (by integrating the velocity fields over time) is smooth. The optimization problem is solved in [6] by a Sobolev gradient descent on the *space of paths*. The resulting path is a geodesic with Riemannian metric given by the Sobolev metric $\|v\|$. In [41], it is shown these geodesics can be computed by integrating a forward evolution equation, determined from the conservation of momentum, with an initial velocity.

Our framework instead uses accelerated gradient descent. Like [6, 41], it is derived from an action integral, but the action has both a kinetic energy and a *potential* energy, which is the objective functional that is to be optimized. In this current work, our kinetic energy arises naturally from physics rather than a Sobolev norm. One of our motivations in this work is to get regularizing effects of Sobolev norms without using Sobolev norms, since that requires inverting differential operators in the optimization, which is computationally expensive. Our kinetic energy is an L^2 metric weighted by *mass*. Our method has acceleration, rather than zero acceleration in [6, 41], and this is obtained by

endowing a diffeomorphism with mass, which is a mass density in $\mathbb{R}^n$. This mass allows for the kinetic energy to endow the optimization with dynamics. Our optimization is obtained as the stationary conditions of the action with respect to both velocity and *mass density*. The latter links our approach to optimal mass transport, described earlier.

1.2.4. Optical Flow. Although our framework is general in solving any optimization on infinite dimensional manifolds, we demonstrate the framework for optimization of diffeomorphisms and specifically for optical flow problems formulated as variational problems in computer vision (e.g., [24, 9, 10, 62, 52, 64, 66, 11]). Optical flow, i.e., determining pixel-wise correspondence between images, is a fundamental problem in computer vision that remains a challenge to solve, mainly because optical flow is a non-convex optimization problem, and thus few methods exist to optimize such problems. Optical flow was first formulated as a variational problem in [24], which consisted of a data fidelity term and regularization favoring smooth optical flow. Since the problem is non-convex, approaches to solve this problem typically involve the assumption of small displacement between frames, so a linearization of the data fidelity term can be performed, and this results in a problem in which the global optimum of [24] can be solved via the solution of a linear PDE. Although standard gradient descent could be used on the non-linearized problem, it is numerically sensitive, extremely computationally costly, and does not produce meaningful results unless coupled with the strategy described next. Large displacements are treated with two strategies: iterative warping and image pyramids. Iterative warping involves iteration of the linearization around the current accumulated optical flow. By use of image pyramids, a large displacement is converted to a smaller displacement in the downsampled images. While this strategy is successful in many cases, there are also many problems associated with linearization and pyramids, such as computing optical flow of thin structures that undergo large displacements. This basic strategy of linearization, iterative warping and image pyramids have been the dominant approach to many variational optical flow models (e.g., [24, 9, 10, 62, 52]), regardless of the regularization that is used (e.g., use of robust norms, total variation, non-local norms, etc). In [62], the linearized problem with TV regularization has been formulated as a convex optimization problem, in which a primal-dual algorithm can be used. In [65] linearization is avoided and rather a gradient descent with respect to a Sobolev metric is computed, and is shown to have a automatic coarse-to-fine optimization behavior. Despite these works, most optical flow algorithms involve simplification of the problem into a linear problem. In this work, we construct accelerated gradient descent algorithms that are applicable to any variational optical flow algorithm in which we avoid the linearization step and aim to obtain a better optimizer. For illustration, we consider here the case of optical flow modeled as a global diffeomorphism, but in principle this can be generalized to piecewise diffeomorphisms as in [66]. Since diffeomorphisms do not form a linear space, rather a infinite-dimensional manifold, we generalize accelerated optimization to that space. We show empirically that our accelerated method can out-perform the standard linearized approach to optical flow in terms of computational speed.

2. Background for Accelerated Optimization on Manifolds.

2.1. Manifolds and Mechanics.

We briefly summarize the key facts in classical mechanics that are the basis for our accelerated optimization method on manifolds.

2.1.1. Differential Geometry. We review differential geometry (from [17]), as this will be needed to derive our accelerated optimization scheme on the *manifold* of diffeomorphisms. First a *manifold* M is a space in which every point $p \in M$ has a (invertible) mapping f_p from a neighborhood of p to a *model space* that is a linear normed vector space, and has an additional compatibility condition that if the neighborhoods for p and q overlap then the mapping $f_p \circ f_q^{-1}$ is differentiable. Intuitively, a manifold is a space that locally appears flat. The model space may be finite or infinite dimensional when the model spaces are finite or infinite dimensional, respectively. In the latter case the manifold is referred to as an *infinite dimensional manifold* and in the former case a *finite dimensional manifold*. The space of diffeomorphisms of $\mathbb{R}^n$, the space of interest in this paper, is an infinite dimensional manifold. The *tangent space* at a point $p \in M$ is the equivalence class, $[\gamma]$, of curves $\gamma : [0, 1] \to M$ under the equivalence that $\gamma(0) = p$ and $(f_p \circ \gamma)'(0)$ are the same for each curve $\gamma \in [\gamma]$. Intuitively, these are the set of possible directions of movement at the point p on the manifold. The *tangent bundle*, denoted TM, is $TM = \{(p, v) : p \in M, v \in T_pM\}$, i.e., the space formed from the collection of all points and tangent spaces.

In this paper, we will assume additional structure on the manifold, namely, that an inner product (called the *metric*) exists on each tangent space T_pM. Such a manifold is called a *Riemannian manifold*. A Riemannian manifold allows one to formally define the lengths of curves $\gamma : [-1, 1] \to M$ on the manifold. This allows one to construct paths of critical length, called *geodesics*, a generalization of a path on constant velocity on the manifold. Note that while existence of geodesics is guaranteed on finite dimensional manifolds, in the infinite dimensional case, there is

no such guarantee. The Riemannian metric also allows one to define *gradients* of functions $g : M \to \mathbb{R}$ defined on the manifold: the gradient $\nabla g(p) \in T_pM$ is defined to be the vector that satisfies $\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\, g(\gamma(\varepsilon))|_{\varepsilon=0} = \langle \nabla g(p), \gamma'(0) \rangle$, where $\gamma(0) = p$, the left hand side is the directional derivative and the right hand side is the inner product from the Riemannian structure.

2.1.2. Mechanics on Manifolds. We now briefly review some of the formalism of classical mechanics on manifolds that will be used in this paper. The material is from [3, 37]. The subject of mechanics describes the principles governing the evolution of a particle that moves on a manifold M. The equations governing a particle are Newton's laws. There are two viewpoints in mechanics, namely the *Lagrangian* and *Hamiltonian* viewpoints, which formulate more general principles to derive Newton's equations. In this paper, we use the Lagrangian formulation to derive equations of motion for accelerated optimization on the manifold of diffeomorphisms. Lagrangian mechanics obtains equations of motion through *variational principles*, which makes it easier to generalize Newton's laws beyond simple particle systems in $\mathbb{R}^3$, especially to the case of manifolds. In Lagrangian mechanics, we start with a function $L : TM \to \mathbb{R}$, called the Lagrangian, from the tangent bundle to the reals. Here we assume that M is a Riemannian manifold. One says that a curve $\gamma : [-1, 1] \to M$ is *a motion in a Lagrangian system* with Lagrangian L if it is an extremal of $A = \int L(\gamma(t), \dot{\gamma}(t))\, \mathrm{d}t$. The previous integral is called an *action integral*. *Hamilton's principle of stationary action* states that the motion in the Lagrangian system satisfies the condition that $\delta A = 0$, where δ denotes the variation, for *all* variations of A induced by variations of the path γ that keep endpoints fixed. The variation is defined as $\delta A := \frac{\mathrm{d}}{\mathrm{d}s}\, A(\tilde{\gamma}(t, s))|_{s=0}$ where $\tilde{\gamma} : [-1, 1]^2 \to M$ is a smooth family of curves (a variation of γ) on the manifold such that $\tilde{\gamma}(t, 0) = \gamma(t)$. The stationary conditions give rise to what is known as *Lagrange's* equations. A *natural Lagrangian* has the special form $L = T - U$ where $T : TM \to \mathbb{R}^+$ is the *kinetic energy* and $U : M \to \mathbb{R}$ is the *potential energy*. The kinetic energy is defined as $T(v) = \frac{1}{2} \langle v, v \rangle$ where $\langle \cdot, \cdot \rangle$ is the inner product from the Riemannian structure. In the case that one has a particle system in $\mathbb{R}^3$, i.e., a collection of particles with masses m_i, in a natural Lagrangian system, one can show that Hamilton's principle of stationary action is equivalent to Newton's law of motion, i.e., that $\frac{\mathrm{d}}{\mathrm{d}t}(m_i \dot{r}_i) = -\frac{\partial U}{\partial r_i}$ where r_i is the trajectory of the i^{th} particle, and $\dot{r}_i$ is the velocity. This states that mass times acceleration is the force, which is given by minus the derivative of the potential in a conservative system. Thus, Hamilton's principle is more general and allows us to more easily derive equations of motion for more general systems, in particular those on manifolds.

In this paper, we will consider *Lagrangian non-autonomous systems* where the Lagrangian is also an explicit function of time t, i.e., $L : TM \times \mathbb{R} \to \mathbb{R}$. In particular, the kinetic and potential energies can both be explicit functions of time: $T : TM \times \mathbb{R} \to \mathbb{R}$ and $U : M \times \mathbb{R} \to \mathbb{R}$. Autonomous systems have an *energy conservation property* and do not converge; for instance, one can think of a moving pendulum with no friction, which oscillates forever. Since the objective in this paper is to minimize an objective functional, we want the system to eventually converge and Lagrangian non-autonomous systems allow for this possibility. For completness, we present some basic facts of the Hamiltonian perspective to elaborate on the previous point, although we do not use this in the present paper. The generalization of total energy is the *Hamiltonian*, defined as the Legendre transform of the Lagrangian: $H(p, q, t) = \langle p, \dot{q} \rangle - L(q, \dot{q}, t)$ where $p = \frac{\mathrm{d}L}{\mathrm{d}\dot{q}}$ is the fiber derivative of L with respect to $\dot{q}$, i.e., $\frac{\mathrm{d}L}{\mathrm{d}\dot{q}} \cdot w = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\, L(q, \dot{q} + \varepsilon w)|_{\varepsilon=0}$. From the Hamiltonian, one can also obtain a system of equations describing motion on the manifold. It can be shown that if $L = T - U$ then $H = T + U$ and more generally, $\frac{\mathrm{d}H}{\mathrm{d}t} = -\frac{\partial L}{\partial t}$ along the stationary path of the action. Thus, if the Lagrangian is natural and autonomous, the total energy is preserved, otherwise energy could be dissipated based on the partial of the Lagrangian with respect to t.

2.2. Variational Approach to Accelerated Optimization in Finite Dimensional Vector Spaces. Accelerated gradient optimization can be motivated by the desire to make an ordinary gradient descent algorithm 1) more robust to noise and local minimizers, and 2) speed-up the convergence while only using first order (gradient) information. For instance, if one computes a noisy gradient due imperfections in obtaining an accurate gradient, a simple heuristic to make the algorithm more robust is to compute a running average of the gradient over iterations, and use that as the search direction. This also has the advantage, for instance in speeding up optimization in narrow shallow valleys. Gradient descent (with finite step sizes) would bounce back and forth across the valley and slowly descend down, but averaging the gradient could cancel the component across the valley and more quickly optimize the function. Strategic dynamically changing weights on previous gradients can boost the descent rate. Nesterov put forth the following

famous scheme [42] which attains an optimal rate of order $\frac{1}{t^2}$ in the case of a smooth, convex cost function $f(x)$:

$$y_{k+1} = x_k - \frac{1}{\beta}\nabla f(x_k), \qquad x_{k+1} = (1-\gamma_k)y_{k+1} + \gamma_k y_k, \qquad \gamma_k = \frac{1-\lambda_k}{\lambda_k+1}, \qquad \lambda_k = \frac{1+\sqrt{1+4\lambda_{k-1}^2}}{2}$$

where x_k is the k-th iterate of the algorithm, y_k is an intermediate sequence, and γ_k are dynamically updated weights.

Recently [63] presented a variational generalization of Nesterov's [42] and other accelerated gradient descent schemes in $\mathbb{R}^n$ based on the Bregman divergence of a convex distance generating function h:

$$d(y,x) = h(y) - h(x) - \nabla h(x)\cdot(y-x) \tag{2.1}$$

and careful discretization of the Euler-Lagrange equations for the time integral of the following Bregman Lagrangian

$$L(X,V,t) = e^{a(t)+\gamma(t)}\left[d(X+e^{-a(t)}V, X) - e^{b(t)}U(X)\right]$$

where the potential energy U represents the cost to be minimized. In the Euclidean case where $h(x) = \frac{1}{2}|x|^2$ gives $d(y,x) = \frac{1}{2}|y-x|^2$, this simplifies to

$$L(X,V,t) = e^{\gamma(t)}\left[e^{-a(t)}\frac{1}{2}|V|^2 - e^{a(t)+b(t)}U(X)\right]$$

where $T = \frac{1}{2}|V|^2$ is the kinetic energy of a unit mass particle in $\mathbb{R}^n$; other definitions of kinetic energies (some leading to faster convergence have been analyzed in [36]). Nesterov's methods [42, 46, 45, 44, 47, 43] belong to a subfamily of Bregman Lagrangians with the following choice of parameters (indexed by $k > 0$)

$$a = \log k - \log t, \qquad b = k\,\log t + \log\lambda, \qquad \gamma = k\,\log t$$

which, in the Euclidean case, yields a non-autonomous Lagrangian as follows:

$$L = \frac{t^{k+1}}{k}\left(T - \lambda k^2 t^{k-2} U\right) \tag{2.2}$$

In the case of $k = 2$, for example, the stationary conditions of the integral of this time-explicit action yield the continuum limit of Nesterov's accelerated mirror descent [43] derived in both [51, 31]. For convex U, [63] show exponential convergence, and a $O(1/t^2)$ convergence in the discrete case (see also [59, 61]). In our case, as the potential is defined on a non-linear manifold, the potentials are generally not convex and we do not have such convergence rate results. However, see [8] for convergence rate and discrete analysis for a PDE related to an Accelerated PDE (3.31) considered in this paper. In particular, the exponential rate is shown.

Since the Bregman Lagrangian assumes that the underlying manifold is a subset of $\mathbb{R}^n$ (in order to define the Bregman distance[2]), which many manifolds do not have - for instance the manifold of diffeomorphisms that we consider in this paper, we instead use the original classical mechanics formulation, which already provides a formalism for considering general metrics though the Riemannian distance, although not equivalent to the Bregman distance.

3. Accelerated Optimization for Diffeomorphisms. In this section, we use the mechanics of particles on manifolds developed in the previous section, and apply it to the case of the infinite-dimensional manifold of diffeomorphisms in $\mathbb{R}^n$ for general n. This allows us to generalize accelerated optimization to infinite dimensional manifolds. Diffeomorphisms are smooth mappings $\phi : \mathbb{R}^n \to \mathbb{R}^n$ whose inverse exists and is also smooth. Diffeomorphisms form a group under composition. The inverse operator on the group is defined as the inverse of the function, i.e., $\phi^{-1}(\phi(x)) = x$. Here smoothness will mean that two derivatives of the mapping exist. The group of diffeomorphisms will be denoted $\mathrm{Diff}(\mathbb{R}^n)$. Diffeomorphisms relate to image registration and optical flow, where the mappings between

[2]One could in fact generalize such operations as addition and subtraction in manifolds, using the exponential and logarithmic maps. We avoid this since in the types of manifolds that we deal with, computing such maps itself requires solving a PDE or another optimization problem. We avoid all these complications, by going back to the formalism in classical mechanics.

two images are often modeled as diffeomorphisms[3]. Recovering diffeomorphisms from two images will be formulated as an optimization problem $U(\phi)$ where U will correspond to the potential energy. Note we avoid calling U the energy as is customary in computer vision literature, because for us the energy will refer to the total mechanical energy (i.e., the sum of the kinetic and potential energies). We do not make any assumptions on the particular form of the potential in this section, as our goal is to be able to accelerate *any* optimization problem for diffeomorphisms, given that one can compute a gradient of the potential. The formulation here allows any of the numerous cost functionals developed over the past three decades for image registration to be accelerated.

In the first sub-section, we give the formulation and evolution equations for the case of acceleration without energy dissipation (Hamiltonian is conserved), since most of the calculations are relevant for the case of energy dissipation, which is needed for the evolution to converge to a diffeomorphism. In the second sub-section, we formulate and compute the evolution equations for the energy dissipation case, which generalizes Nesterov's method to the infinite dimensional manifold of diffeomorphisms. Finally, in the last sub-section we give an example potential and its gradient calculation for a standard image registration or optical flow problem.

3.1. Acceleration Without Energy Dissipation.

3.1.1. Formulation of the Action Integral. Since the potential energy U is assumed given, in order to formulate the action integral in the non-dissipative case, we need to define kinetic energy T on the space of diffeomorphisms. Since diffeomorphisms form a manifold, we can apply the the results in the previous section and note that the kinetic energy will be defined on the tangent space to $\text{Diff}(\mathbb{R}^n)$ at a particular diffeomorphism ϕ. This will be denoted $T_\phi\text{Diff}(\mathbb{R}^n)$. The tangent space at ϕ can be roughly thought of as the set of local perturbations v of ϕ given for all ε small that preserve the diffeomorphism property, i.e., $\phi + \varepsilon v$ is a diffeomorphism. One can show that the tangent space is given by

$$T_\phi\text{Diff}(\mathbb{R}^n) = \{v : \phi(\mathbb{R}^n) \to \mathbb{R}^n \ : \ v \text{ is smooth } \}. \tag{3.1}$$

In the above, since ϕ is a diffeomorphism, we have that $\phi(\mathbb{R}^n) = \mathbb{R}^n$. However, we write $v : \phi(\mathbb{R}^n) \to \mathbb{R}^n$ to emphasize that the velocity fields in the tangent space are defined on the range of ϕ, so that v is interpreted as a Eulerian velocity. By definition of the tangent space, an infinitesimal perturbation of ϕ by a tangent vector, given by $\phi + \varepsilon v$, will be a diffeomorphism for ε sufficiently small. Note that the previous operation of addition is defined as follows:

$$(\phi + \varepsilon v)(x) = \phi(x) + \varepsilon v(\phi(x)).$$

The tangent space is a set of smooth vector fields on $\phi(\mathbb{R}^n)$ in which the vector field at each point $\phi(x)$, displaces $\phi(x)$ infinitesimally by $v(\phi(x))$ to form another diffeomorphism.

We note a classical result from [18], which will be of utmost importance in our derivation of accelerated optimization on $\text{Diff}(\mathbb{R}^n)$. The result is that any (orientable) diffeomorphism may be generated by integrating a time-varying smooth vector field over time, i.e.,

$$\partial_t \phi_t(x) = v_t(\phi_t(x)), \quad x \in \mathbb{R}^n, \tag{3.2}$$

where ∂_t denotes partial derivative with respect to t, ϕ_t denotes a time varying family of diffeomorphisms evaluated at the time t, and v_t is a time varying collection of vector fields evaluated at time t. The path $t \mapsto \phi_t(x)$ for a fixed x represents a trajectory of a particle starting at x and flowing according to the velocity field.

The space on which the kinetic energy is defined is now clear, but one more ingredient is needed before we can define the kinetic energy. Any accelerated method will need a notion of *mass*, otherwise acceleration is not possible, e.g., a mass-less ball will not accelerate. We generalize the concept of mass to the infinite dimensional manifold of diffeomorphisms, where there are infinitely more possibilities than a single particle in the finite dimensional case considered by [63]. There optimization is done on a finite dimensional space, the space of a *single* particle, and the possible choices of mass are just different fixed constants. The choice of the constant, given the particle's mass remains fixed, is irrelevant to the final evolution. This is different in the case of diffeomorphisms. Here we imagine that an

[3] In medical imaging, the model of diffeomorphisms for registration is fairly accurate since typically full 3D scans are available and thus all points in one image correspond to the other image and vice versa [6]. Of course there are situations (such as growth of tumors) where the diffeomorphic assumption is invalid. In vision, typically images have occlusion phenomena and multiple objects moving in different ways. So a diffeomorphism is not a valid assumption, it is however a good model when restricted to a single object in the un-occluded part [66, 32, 33].

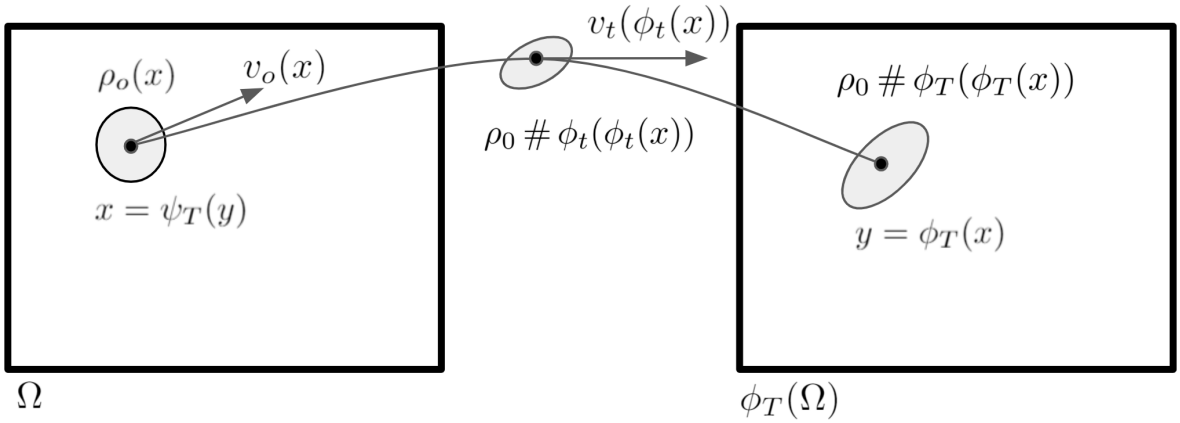


Fig. 1: Schematic of the quantities defined to derive accelerated optimization on the diffeomorphism manifold. ϕ denotes the time-varying forward mapping (at each time the diffeomorphism ϕ_t is a point on the manifold), ψ is its inverse. v denotes the time-varying vector field that defines ϕ and is an element of the tangent space to diffeomorphisms. ρ is the time-varying mass density (the ellipsoids depict a infinitesimal mass being transported by ϕ) defined on the image domain Ω. $\rho_0 \# \phi_t$ denotes the push forward of ρ_0 by ϕ_t, i.e., $\rho_0 \# \phi_t = (\rho_0 \circ \psi_t) \det \nabla \psi$. Note ρ does not relate to the images I_0, I_1 to be registered, and is an auxiliary variable used to define the optimization procedure.

infinite number of particles densely distributed in $\mathbb{R}^n$ with mass exist and are displaced by the velocity field v at every point. Since the particles are densely distributed, it is natural to represent the mass of all particles with a *mass density* $\rho : \mathbb{R}^n \to \mathbb{R}$, similar to a fluid at a fixed time instant. The density ρ is defined as mass divided by volume as the volume shrinks. During the evolution to optimize the potential U, the particles are displaced continuously and thus the density of these particles will in general change over time. Note the density will change even if the density at the start is constant except in the case of full translation motion (when v is spatially constant). The latter case is not general enough, as we want to capture general diffeomorphisms. We will assume that the system of particles in $\mathbb{R}^n$ is closed and so we impose a *mass preservation constraint*, i.e.,

$$\int_{\mathbb{R}^n} \rho(x) \, \mathrm{d}x = 1, \tag{3.3}$$

where we assume the total mass is one without loss of generality. Note that the evolution of a time varying density ρ_t as it is deformed in time by a time varying velocity is given by the *continuity equation*, which is a local form of the conservation of mass given by (3.3). The continuity equation is defined by the partial differential equation

$$\partial_t \rho(x) + \operatorname{div}(\rho(x) v(x)) = 0, \quad x \in \mathbb{R}^n \tag{3.4}$$

where $\operatorname{div}()$ denotes the divergence operator acting on a vector field and is $\operatorname{div}(F) = \sum_{i=}^{n} \partial_{x_i} F^i$ where ∂_{x_i} is the partial with respect to the i^{th} coordinate and F^i is the i^{th} component of the vector field. We will assume that the mass distribution dies down to zero outside a compact set so as to avoid boundary considerations in our derivations.

We now have the two ingredients, namely the tangent vectors to $\mathrm{Diff}(\mathbb{R}^n)$ and the concept of mass, which allows us to define a natural physical extension of the kinetic energy to the case of an infinite mass distribution. We present one possible kinetic energy to illustrate the idea of accelerated optimization, but this is by no means the only definition of kinetic energy. We envision this to be part of the design process in which one could get a multitude of various different accelerated optimization schemes by defining different kinetic energies. Our definition of kinetic energy is just the kinetic energy arising from fluid mechanics:

$$T(v) = \int_{\phi(\mathbb{R}^n)} \frac{1}{2} \rho(x) |v(x)|^2 \, \mathrm{d}x, \tag{3.5}$$

which is just the integration of single particle's kinetic energy $\frac{1}{2}m|v|^2$ and matches the definition of the kinetic energy of a sum of particles in elementary physics. Note that the kinetic energy is just one-half times the norm squared for the norm arising from the Riemannian metric [3], i.e., an inner product on the tangent space of $\mathrm{Diff}(\mathbb{R}^n)$. The Riemannian metric is given by $\langle v_1, v_2 \rangle = \int_{\mathbb{R}^n} \rho(x) v_1(x) \cdot v_2(x) \, \mathrm{d}x$, which is just a weighted $\mathbb{L}^2$ inner product. Note that the above kinetic energy is just one particular choice, and other choices of the metric would lead to different kinetic energy definitions, which is an area for future research. This paper analyzes (3.5) as an example, and is motivated by its physical interpretation and simplicity.

We are now ready to define the action integral for the case of $\mathrm{Diff}(\mathbb{R}^n)$, which is defined on *paths* of diffeomorphisms. A path of diffeomorphisms is $\phi : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ and we will denote the diffeomorphism at time t along this path as ϕ_t. Since diffeomorphisms are generated by velocity fields, we may equivalently define the action in terms of *paths* of velocity fields. A path of velocity fields is given by $v : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$, and the velocity at time t along the path is denoted v_t. Notice that the action requires a kinetic energy and the kinetic energy is dependent on the mass density. Thus, a path of densities $\rho : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^+$ is required, which represents the mass distribution of the particles in $\mathbb{R}^n$ as they are deformed along time by the velocity field v_t. This path of densities is subject to the continuity equation. With this, the action integral is then

$$A = \int \left[T(v_t) - U(\phi_t) \right] \, \mathrm{d}t, \tag{3.6}$$

where the integral is over time, and we do not specify the limits of integration as it is irrelevant as the endpoints will be fixed and the action will be thus independent of the limits. Note that the action is implicitly a function of three paths, i.e., v_t, ϕ_t and ρ_t. Further, these paths are not independent of each other as ϕ_t depends on v_t through the generator relation (3.2), and ρ_t depends on v_t through the continuity equation (3.4).

3.1.2. Stationary Conditions for the Action. We now derive the stationary conditions for the action integral (3.6), and thus the evolution equation for a path of diffeomorphisms, which is Hamilton's principle of stationary action, equivalent to a generalization of Newton's laws of motion extended to diffeomorphisms. As discussed earlier, we would like to find the stationary conditions for the action integral (3.6), defined on the path ϕ_t, under the conditions that it is generated by a path of smooth velocity fields v_t, which is also coupled with the mass density ρ_t.

We treat the computation of the stationary conditions of the action as a constrained optimization problem with respect to the two aforementioned constraints. To do this, it is easier to formulate the action in terms of the path of the inverse diffeomorphisms ϕ_t^{-1}, which we will call ψ_t. This is because the non-linear PDE constraint (3.2) can be equivalently reformulated as the following linear transport PDE in the inverse mappings:

$$\partial_t \psi_t(x) + [D\psi_t(x)] v_t(x) = 0, \quad x \in \mathbb{R}^n \tag{3.7}$$

where D denotes the derivative (Jacobian) operator. To derive the stationary conditions with respect to the constraints, we use the method of Lagrange multipliers. We denote by $\lambda : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^n$ the Lagrange multiplier according to (3.7). We denote $\mu : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}$ as the Lagrange multiplier for the continuity equation (3.4). Because we would like to be able to have possibly discontinuous solutions of the continuity equation, we formulate it in its weak form by multiplying the constraint by the Lagrange multiplier and integrating by parts thereby removing the derivatives on possibly discontinuous ρ:

$$\int \int_{\mathbb{R}^n} \mu \left[\partial_t \rho + \mathrm{div}\,(pv) \right] \, \mathrm{d}x \, \mathrm{d}t = - \int \int_{\mathbb{R}^n} \left[\partial_t \mu + \nabla \mu \cdot v \right] \rho \, \mathrm{d}x \, \mathrm{d}t, \tag{3.8}$$

where ∇ denotes the spatial gradient operator. Notice that we ignore the boundary terms from integration by parts as we will eventually compute stationary conditions, and we are assuming fixed initial conditions for ρ_0 and we assume that ρ_∞ converges and thus cannot be perturbed when computing the variation of the action integral. With this, we can formulate the action integral with Lagrange multipliers as

$$A = \int \left[T(v) - U(\phi) \right] \, \mathrm{d}t + \int \int_{\mathbb{R}^n} \lambda^T [\partial_t \psi + (D\psi) v] \, \mathrm{d}x \, \mathrm{d}t - \int \int_{\mathbb{R}^n} \left[\partial_t \mu + \nabla \mu \cdot v \right] \rho \, \mathrm{d}x \, \mathrm{d}t, \tag{3.9}$$

where we have omitted the subscripts to avoid cluttering the notation. Notice that the potential U is now a function of ψ, and the action depends now on ρ, ψ, v and the Lagrange multipliers μ, λ.

We now compute variations of A as we perturb the paths by variations $\delta\rho$, δv and $\delta\phi$ along the paths. The variation with respect to ρ is defined as $\delta A \cdot \delta\rho = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} A(\rho + \varepsilon\delta\rho, v, \psi)\big|_{\varepsilon=0}$, and the other variations are defined in a similar fashion. By computing these variations, we get the following stationary equations:

THEOREM 3.1. *The stationary conditions of the path for the action* (3.9) *are*

$$\partial_t \lambda + (D\lambda)v + \lambda div\,(v) = (\nabla\psi)^{-1}\nabla U(\phi) \tag{3.10}$$

$$\rho v + (\nabla\psi)\lambda - \rho\nabla\mu = 0 \tag{3.11}$$

$$\partial_t \mu + \nabla\mu \cdot v = \frac{1}{2}|v|^2 \tag{3.12}$$

where $\nabla U(\phi) \in T_\phi Diff(\mathbb{R}^n)$ denotes the functional gradient of U with respect to ϕ (see Appendix 6.1), and $\nabla\mu, \nabla\psi$ are spatial gradients. The original constraints (3.7) *on the mapping and the continuity equation* (3.4) *are part of the stationary conditions.*

Proof. See Appendix 6.2. ∎

While the previous theorem does give the stationary conditions and evolution of the Lagrange multipliers, in order to define a forward evolution method where the initial conditions for the density, mapping and velocity are given, we would need initial conditions for the Lagrange multipliers, which are not known from the calculation leading to Theorem 3.1. Therefore, we will now eliminate the Lagrange multipliers and rewrite the evolution equations in terms of forward equations for the velocity, mapping and density. This leads to the following theorem:

THEOREM 3.2 (Evolution Equations for the Path of Least Action). *The stationary conditions for the path of the action integral* (3.6) *subject to the constraints* (3.2) *on the mapping and the continuity equation* (3.4) *are given by the forward evolution equation*

$$\partial_t v = -(Dv)v - \frac{1}{\rho}\nabla U(\phi), \tag{3.13}$$

which describes the evolution of the velocity. The forward evolution equation for the diffeomorphism is given by (3.2)*, that of its inverse mapping is given by* (3.7)*, and the forward evolution of its density is given by* (3.4)*.*

Proof. See Appendix 6.3. ∎

REMARK 1 (Relation to Euler's Equations). *The terms $\partial_t v + (Dv)v$ (along with the continuity equation) is the left hand side of the* compressible Euler Equation *[37], which describes the motion of a perfect fluid (i.e., assuming no heat transfer or viscous effects). The difference is that the right hand side in* (3.13) *is the gradient of the potential, which we seek to optimize, that depends on the diffeomorphism that is the integral of the velocity over time, rather than the gradient of pressure that is purely a function of density in the Euler equations.*

With this theorem, it is now possible to numerically compute the stationary path of the action, by starting with initial conditions on the density, mapping and velocity. The velocity is updated by (3.13), the mapping is then updated by (3.2), and the density is updated by (3.4). Note that the density at each time impacts the velocity as seen in (3.13). These equations are a set of coupled partial differential equations. They describe the path of stationary action when the action integral does not arise from a system that has dissipative forces. Notice the velocity evolution is a natural analogue of Newton's equations. Indeed, if we consider the material derivative, which describes the time rate of change of a quantity subjected to a time dependent velocity field, then one can write the velocity evolution (3.13) as follows.

THEOREM 3.3 (Equivalence of Critical Paths of Action to Newton's 2nd Law). *The velocity evolution* (3.13) *derived as the critical path of the action integral* (3.6) *is*

$$\rho\frac{Dv}{Dt} = -\nabla U(\phi), \tag{3.14}$$

where $\frac{Df}{Dt} := \partial_t f + (Df)v$ is the material derivative.

Proof. This is consequence of the definition of material derivative. ∎

The material derivative is obtained by taking the time derivative of f along the path $t \to \phi(t, x)$, i.e., $\frac{\mathrm{d}}{\mathrm{d}t} f(t, \phi(t, x))$. Therefore, Dv/Dt is the derivative of velocity along the path. The equation (3.14) says the time rate of change of

velocity times density is equation to minus the gradient of the potential, which is Newton's 2nd law, i.e., the mass times acceleration is equal to the force, which is given by the gradient of the potential in a conservative system.

The evolution described by the equations above will not converge. This is because the total energy is conserved, and thus the system will oscillate over a (local) minimum of the potential U, forever, unless the initialization is at a stationary point of the potential U. In practice, due to discretization of the equations, which require entropy preserving schemes [50], the implementation will dissipate energy and the evolution equations eventually converge.

3.1.3. Viscosity Solution and Regularity. An important question is whether the evolution equations given by Theorem 3.2 maintain that the mapping ϕ_t remains a diffeomorphism given that one starts the evolution with a diffeomorphism. This is of course important since all of the derivations above were done assuming that ϕ is a diffeomorphism, moreover for many applications one wants to maintain a diffeomorphic mapping. The answer is affirmative since to define a solution of (3.13), we define the solution as the *viscosity solution* (see e.g., [16, 49, 50]). The viscosity solution is defined as the limit of the equation (3.13) with a diffusive term of the velocity added to the right hand side, as the diffusive coefficient goes to zero. More precisely

$$\partial_t v_\varepsilon = -(Dv_\varepsilon)v_\varepsilon + \varepsilon\Delta v_\varepsilon - \frac{1}{\rho}\nabla U(\phi), \tag{3.15}$$

where Δ denotes the spatial Laplacian, which is a smoothing operator. This leads to a smooth (C^∞) solution due to the known smoothing properties of the Laplacian. The viscosity solution is then $v = \lim_{\varepsilon\to 0} v_\varepsilon$. In practice, we do not actually add in the diffusive term, but rather approximate the effects with small ε by using entropy conditions in our numerical implementation. One may of course add the diffusive term to induce more regularity into the velocity and thus into the mapping ϕ. Larger ε would make the resulting optimization flow smoother both spatially and temporally. Since the velocity is smooth (C^∞), the integral of a smooth vector field will result in a diffeomorphism [18].

3.1.4. Discussion. An important property of these evolution equations, when compared to virtually all previous image registration and optical flow methods is the lack of need to compute inverses of differential operators, which are global smoothing operations, and are expensive. Typically, in optical flow (such as the classical Horn & Schunck [24]) or LDDMM [6] where one computes Sobolev gradients, one needs to compute inverses of differential operators, which are expensive. Of course one could perform standard gradient descent, which does not typically require computing inverses of differential operators, but gradient descent is known not to be feasible and it is hard to numerically implement without significant pre-processing, and easily gets stuck in what are effectively numerical local minima. The equations in Theorem 3.2 are all local, and experiments suggest they are not susceptible to the problems that plague gradient descent.

3.1.5. Constant Density Case. We now analyze the case when the density ρ is chosen to be a fixed constant, and we derive the evolution equations. This case is the one that is assumed within the computational anatomy literature, and we show how considering the density evolution simplifies the optimization equations. In this case, the kinetic energy simplifies as follows

$$T(v) = \frac{\rho}{2}\int_{\phi(\mathbb{R}^n)} |v(x)|^2 \,\mathrm{d}x. \tag{3.16}$$

We can define the action integral as before (3.6) with the previous definition of kinetic energy, and we can derive the stationary conditions by defining the following action integral incorporating the mapping constraint (3.7). This gives the modified action integral as

$$A = \int \left[T(v) - U(\phi)\right] \,\mathrm{d}t + \int\int_{\mathbb{R}^n} \lambda^T[\partial_t\psi + (D\psi)v] \,\mathrm{d}x \,\mathrm{d}t. \tag{3.17}$$

Note that the continuity equation is no longer imposed as a constraint as the density is treated as a fixed constant. This leads to the following stationary conditions.

THEOREM 3.4. *The stationary conditions of the path for the action* (3.17) *are*

$$\partial_t\lambda + (D\lambda)v + \lambda div\,(v) = (\nabla\psi)^{-1}\nabla U(\phi) \tag{3.18}$$

$$\rho v + (\nabla\psi)\lambda = 0 \tag{3.19}$$

where $\nabla U(\phi) \in T_\phi Diff(\mathbb{R}^n)$ denotes the functional gradient of U with respect to ϕ, and $\nabla\psi$ are spatial gradients. The original constraint (3.7) on the mapping is part of the stationary conditions.

Proof. The computation is similar to the non-constant density case Appendix 6.2. Note that stationary condition with respect to the mapping remains the same as the density constraint in the non-constant density case does not depend on the mapping. The stationary condition with respect to the velocity avoids the variation with respect to the density constraint in the non-constant density case, and remains the same except for the last term. ∎

As before, we can solve for the velocity evolution directly. This results in the following result.

THEOREM 3.5 (Evolution Equations for the Path of Least Action). *The stationary conditions for the path of the action integral (3.6) with kinetic energy (3.16) subject to the constraint (3.2) on the mapping is given by the forward evolution equation*

$$\partial_t v = -(Dv)v - (\nabla v)v - v div\,(v) - \frac{1}{\rho}\nabla U(\phi). \tag{3.20}$$

The forward evolution equation for the diffeomorphism is given by (3.2), and that of its inverse mapping is given by (3.7).

Proof. We can apply Lemma 6.5 in Appendix 6.3 with $w = -\frac{1}{\rho}v$. ∎

The former equation (3.20) (without the potential term) is known as the Euler-Poincaré equation (EPDiff), the geodesic equation for the diffeomorphism group under the L^2 metric [41]. This shows that one relationship between Euler's equation and EPDiff is that Euler's equation is derived by a time-varying density in the kinetic energy, which is optimized over the mass distribution along with the velocity whereas EPDiff assumes a constant mass density in the kinetic energy. The non-constant density model (arising in Euler's equation) has a natural interpretation in terms of Newton's equations.

3.2. Acceleration with Energy Dissipation. We now present the case of deriving the stationary conditions for a system on the manifold of diffeomorphisms in which total energy dissipates. This is important so the system will converge to a local minima, and not oscillate about a local minimum forever, as the evolution equations from the previous section. To do this, we consider time varying scalar functions $a, b : [0, \infty) \to \mathbb{R}^+$, and define the action integral, again defined on paths of diffeomorphisms, as follows:

$$A = \int \left[a_t T(v_t) - b_t U(\phi_t)\right] \mathrm{d}t, \tag{3.21}$$

where a_t, b_t denote the values of the scalar at time t. We may again go through finding the stationary conditions subject to the mapping constraint (3.7) and the continuity equation constraint (3.4), with Lagrange multiplier and then derive the forward evolution equations. The final result is as follows:

THEOREM 3.6 (Evolution Equations for the Path of Least Action). *The stationary conditions for the path of the action integral (3.21) subject to the constraints (3.2) on the mapping and the continuity equation (3.4) are given by the forward evolution equation*

$$a\partial_t v + a(Dv)v + (\partial_t a)v = -\frac{b}{\rho}\nabla U(\phi), \tag{3.22}$$

which describes the evolution of the velocity. The same evolution equations as Theorem 3.2 for the mappings (3.2) and (3.7), and density hold (3.4).

Proof. See Appendix 6.4. ∎

If we consider certain forms of a and b, then one can arrive at various generalizations of Nesterov's schemes. In particular, the choice of a and b below are those considered in [63] to explain various versions of Nesterov's schemes, which are optimization schemes in finite dimensions.

THEOREM 3.7 (Evolution Equations for the Path of Least Action: Generalization of Nesterov's Method). *If we choose*

$$a_t = e^{\gamma_t - \alpha_t} \quad \textit{and} \quad b_t = e^{\alpha_t + \beta_t + \gamma_t}$$

where

$$\alpha_t = \log p - \log t, \quad \beta_t = p \log t + \log C, \quad \gamma_t = p \log t,$$

$C > 0$ is a constant, and p is a positive integer, then we will arrive at the evolution equation

$$\partial_t v = -\frac{p+1}{t} v - (Dv)v - \frac{1}{\rho} C p^2 t^{p-2} \nabla U(\phi). \tag{3.23}$$

In the case $p = 2$ and $C = 1/4$ the evolution reduces to

$$\partial_t v = -\frac{3}{t} v - (Dv)v - \frac{1}{\rho} \nabla U(\phi). \tag{3.24}$$

The case $p = 2$ was considered in [63] as the continuum equivalent to Nesterov's original scheme in finite dimensions. We can notice that this evolution equation is the same as the evolution equations for the non-dissipative case (3.13), except for the term $-(3/t)v$. One can interpret the latter term as a frictional dissipative term, analogous to viscous resistance in fluids. Thus, even in this case the equation has a natural interpretation that arises from Newton's laws.

Thus, the final system of equations (with general a, b) that are to be discretized and used in a numerical implementation is

$$\partial_t \phi = v \circ \phi \tag{3.25}$$

$$\partial_t v = -\frac{\partial_t a}{a} v - (Dv)v - \frac{1}{\rho} \frac{b}{a} \nabla U(\phi), \tag{3.26}$$

$$\partial_t \rho = -\text{div}\,(\rho v) \tag{3.27}$$

$$v(0, x) = v_0(x) \tag{3.28}$$

$$\phi(0, x) = x \tag{3.29}$$

$$\rho(0, x) = \rho_0(x), \tag{3.30}$$

which is an *Eulerian* description of the accelerated motion. The numerical algorithm is given in Algorithm 6.1 in the Appendix.

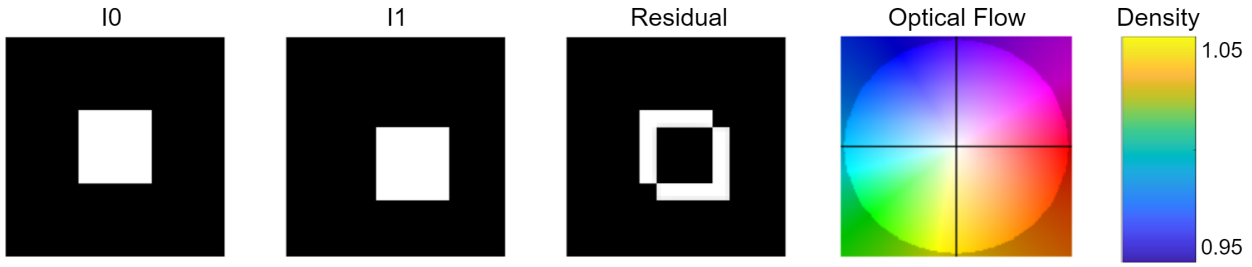
We show an experiment in Figures 2a, 2b and 3 to conceptually illustrate the accelerated evolution. Here, we study a simple optical flow problem whose potential is a standard Horn& Schunck loss (see (3.32)). We show the evolution, which aims to register a square to a translated square (see Figure 3). We compare the evolutions for acceleration with and without damping, both which introduce oscillations, but the former dies down. The evolutions eventually determine a translation, even though the velocity can vary with pixel. Notice the mass density evolves through non-uniform densities at times, indicating a non-trivial mass evolution impacting the dynamics. A comparison to gradient descent is shown in Figure 3, in particular showing that acceleration drastically speeds up convergence.

3.3. Second Order PDE for Acceleration. We now convert the system of PDE for the forward mapping and velocity into a second order PDE in the forward mapping itself, which constitutes the Lagrangian description of the accelerated motion. Interestingly, this eliminates the non-linearity from the non-potential terms.

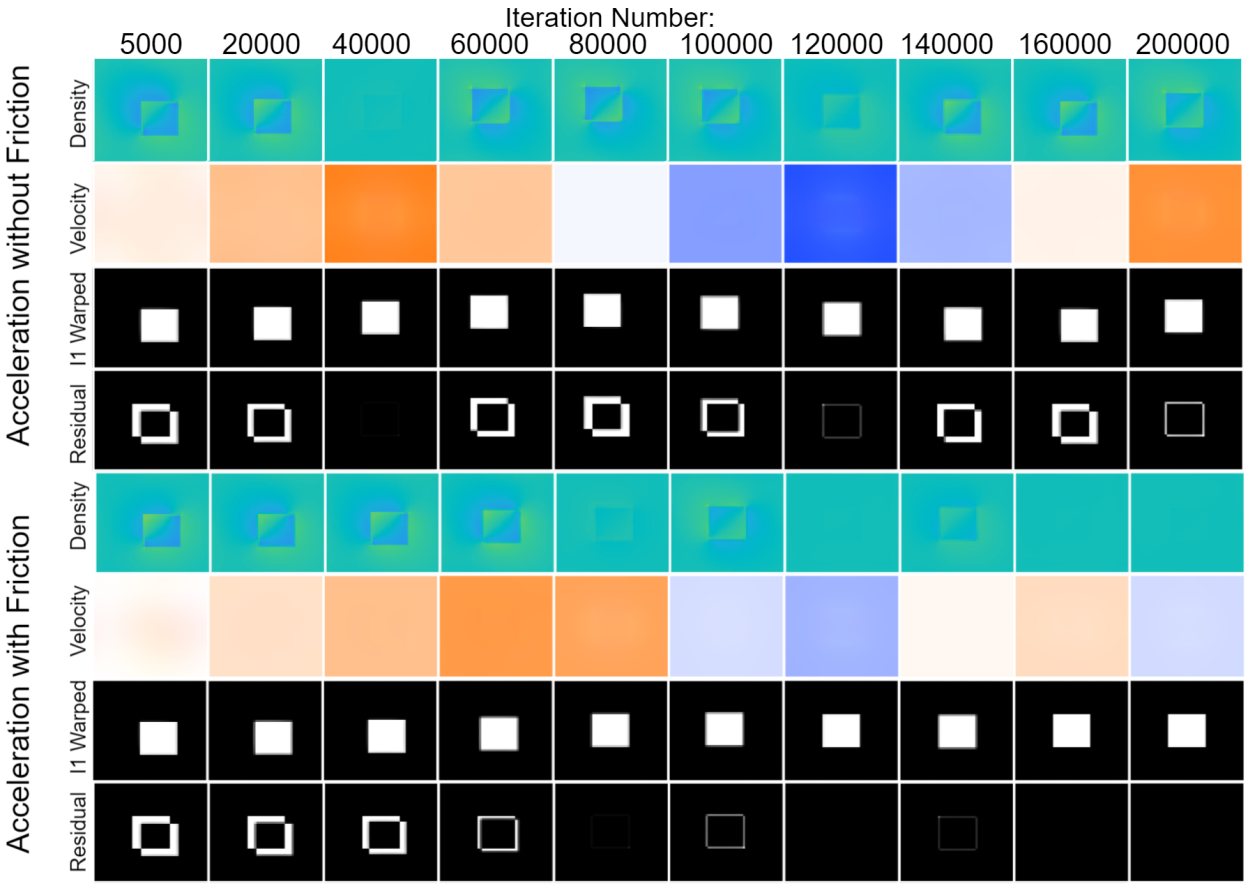
THEOREM 3.8 (Second Order PDE for the Forward Mapping). *The accelerated optimization, arising from the stationarity of the action integral (3.21), given by the system of PDE defined by (3.22) and the forward mapping (3.2) is*

$$a \frac{\partial^2 \phi}{\partial t^2} + (\partial_t a) \frac{\partial \phi}{\partial t} + \frac{b}{\rho_0} \widetilde{\nabla} U(\phi) = 0, \tag{3.31}$$

where ρ_0 is the initial density, $\widetilde{\nabla} U(\phi) = [\nabla U(\phi) \circ \phi] \det \nabla \phi$ is the gradient defined on the un-warped domain, i.e., $\delta A \cdot \delta \phi = \int_{\mathbb{R}^n} \widetilde{\nabla} U(\phi)(x) \cdot \delta\phi(x)\, \mathrm{d}x$ is satisfied for all perturbations $\delta\phi$ of ϕ.



(a) Illustrative experiment: The experiment (whose results are in Fig. 2b and 3) computes the optical flow (registration) between I_0 and I_1 using a common optical flow loss function. The initial residual ($|I_1 - I_0|$) is shown. The fourth image from left is a color code for the velocity (the direction of the velocity is indicated by the color and the intensity of color indicates magnitude). The fifth image is a color code for the mass density graphs used in Figure 2b.



(b) Comparison of evolutions of accelerated optimization with and without friction. The four rows are the density $\rho_t \# \phi_t$, velocity $v \circ \phi_t$, image warp $I_1 \circ \phi$ and residual $|I_0 - I_1(\phi)|$ for the undamped and damped accelerated descents over various iterations. Notice that the undamped descent overshoots the target and switches directions as evidenced by the shift in the velocity from orange to blue; it continues to oscillate indefinitely. The addition of a friction term kills the oscillations, allows convergence and for the minimization of the residual as shown. Notice that in both cases, the mass moves within and around the square in non-trivial ways, different than constant density. Each are initialized with a constant density and at convergence, the density is also constant.

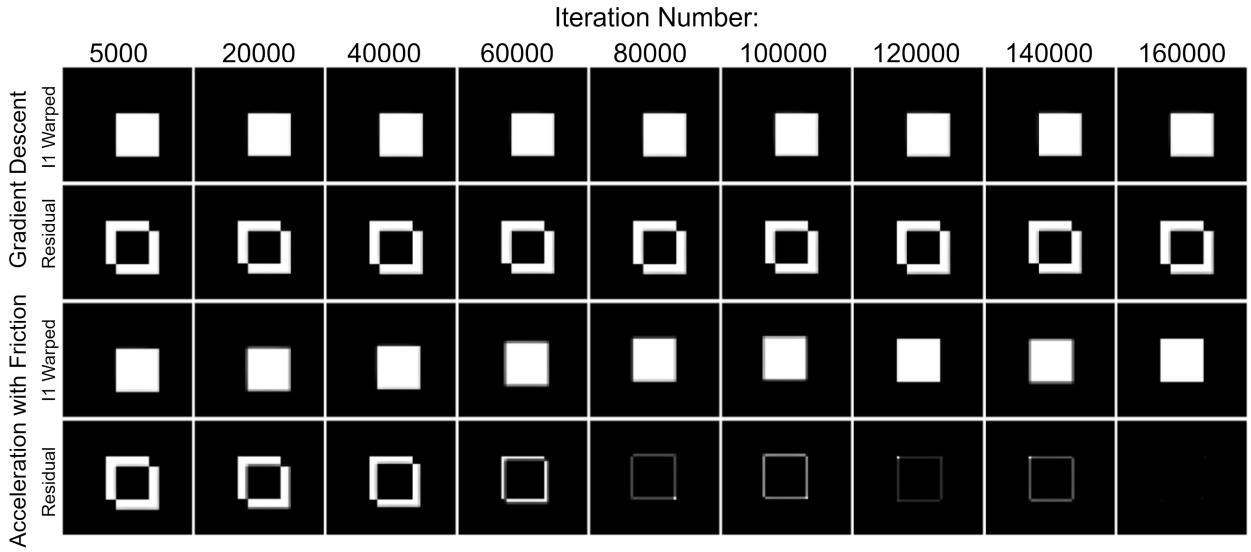


Fig. 3: Comparison of evolutions of gradient descent and accelerated gradient descent. $I_1(\phi)$ and the residual are shown throughout the evolution. As can be seen, acceleration converges in far fewer iterations (gradient descent eventually converges, on the order of $k \approx 3 \times 10^7$ iterations). See Section 4.1.2 for details of experimental setup.

Proof. We differentiate the definition of the forward mapping in time to obtain (3.2) and substituting the velocity evolution (3.22):

$$\begin{aligned}\partial_{tt}\phi &= (\partial_t v) \circ \phi + [(Dv) \circ \phi]\partial_t \phi \\ &= -[(Dv) \circ \phi] v \circ \phi - \frac{\partial_t a}{a} v \circ \phi - \frac{b}{a}\frac{1}{\rho \circ \phi}\nabla U(\phi) \circ \phi + [(Dv) \circ \phi]\partial_t \phi \\ &= -\frac{\partial_t a}{a}\partial_t \phi - \frac{b}{a}\frac{1}{\rho \circ \phi}\nabla U(\phi) \circ \phi.\end{aligned}$$

We note the following for any $B \subset \mathbb{R}^n$, because of mass preservation, we have that

$$\int_B \rho_0(x)\,\mathrm{d}x = \int_{\phi(B)} \rho_t(y)\,\mathrm{d}y = \int_B \rho_t(\phi(x)) \det \nabla\phi(x)\,\mathrm{d}x,$$

where the last equality is obtained by a change of variables. Since we can take B arbitrarily small, $\rho_0(x) = \rho_t(\phi(x)) \det \nabla\phi(x)$. Using this last formula, we see that

$$\frac{1}{\rho \circ \phi}\nabla U(\phi) \circ \phi = \frac{1}{\rho_0}\widetilde{\nabla} U(\phi),$$

which proves the proposition. ∎

Note that the advantage of this Lagrangian approach is that the evolution of the mass has been eliminated, making for a simpler implementation. The advantage of the Eulerian formulation, however, is that it more easily allows for more general mass flow models than considered in this paper (see [68, 67]), which may not have as simple Lagrangian formulation. The discrete implementation is discussed in Appendix 6.6 and the implementation is shown in Algorithm 6.2.

In the case that ∇U is linear, the PDE is a vector-valued version of the PDE considered in [8], which has been analyzed in terms of numerical discretization and convergence rate [8]. In particular, [8] show that the PDE has an exponential convergence rate, which is equivalent to the rate shown in the ODE case by [63]. As we will see below, in cases of interest, the gradient will not be linear. However, the analysis may approximate what happens within the basin of a local minimum, where the gradient can be approximated as linear.

3.4. Illustrative Potential Energy for Diffeomorphisms. We now consider a standard potential for illustrative purposes in simulations, and derive the gradient. The objective is for the evolution equations in the previous section to minimize the potential, which is a function of the mapping. Our evolution equations in the previous section are general and work with *any* potential; our purpose in this section is not to advocate a particular potential, but to show how the gradient of the potential is computed so that it can be used in the evolution equations in the previous section. We consider the standard Horn & Schunck model for optical flow defined as

$$U(\phi) = \frac{1}{2}\int_{\mathbb{R}^n} |I_1(\phi(x)) - I_0(x)|^2 \,\mathrm{d}x + \frac{1}{2}\alpha \int_{\mathbb{R}^n} |\nabla(\phi(x) - x)|^2 \,\mathrm{d}x, \tag{3.32}$$

where $\alpha > 0$ is a weight, and I_0, I_1 are images. The first term is the data fidelity which measures how close ϕ deforms I_1 back to I_0 through the squared norm, and the second term penalizes non-smoothness of the displacement field, given by $\phi(x) - x$ at the point x. Notice that the potential is a function of only the mapping ϕ, and not the velocity.

We now compute the functional gradient of U with respect to the mapping ϕ, denoted by the expression $\nabla U(\phi)$. This gradient is defined by the relation (see Appendix 6.1) $\delta U \cdot \delta\phi = \int_{\phi(\mathbb{R}^n)} \nabla U(\phi) \cdot \delta\phi \,\mathrm{d}x$, i.e., the functional gradient satisfies the relation that the $\mathbb{L}^2$ inner product of it with any perturbation $\delta\phi$ of ϕ is equal to the variation of the potential U with respect to the perturbation $\delta\phi$. With this definition, one can show that (see Appendix 6.1)

$$\nabla U(\phi) = [(I_1 - I_0 \circ \psi)\nabla I_1 - \alpha(\Delta\phi) \circ \psi] \det \nabla\psi, \tag{3.33}$$

where $\det$ denotes the determinant.

We can also see that the gradient defined on the un-warped domain is

$$\widetilde{\nabla} U(\phi) = (I_1 \circ \phi - I_0)\nabla I_1 \circ \phi - \alpha\Delta\phi, \tag{3.34}$$

therefore, the generalization of Nesterov's method on the original domain itself, in this case is

$$\frac{\partial^2\phi}{\partial t^2} + \frac{3}{t}\frac{\partial\phi}{\partial t} - \frac{\alpha}{\rho_0}\Delta\phi + \frac{1}{\rho_0}(I_1 \circ \phi - I_0)\nabla I_1 \circ \phi = 0, \tag{3.35}$$

which is a damped *wave equation*.

4. Experiments. We conduct experiments to illustrate the behavior of accelerated gradient descent (6.33) and compare it to gradient descent, and then illustrate the advantage of acceleration gradient descent against a standard optimizer for optical flow on a benchmark dataset. We demonstrate proof-of-concept of accelerated optimization using the Eulerian approach (3.25)-(3.30) in Section 4.1 and then demonstrate the Lagrangian approach (3.31) in Section 3.31 against standard optical flow optimization.

4.1. Eulerian Approach. In our first set of experiments (Sub-sections 4.1.1 to 4.1.3), we compare the discrete implementation of the Eulerian approach ((4.1.1)) to accelerated optimization on the manifold of diffeomorphisms to standard (Riemannian L^2) gradient descent. This will illustrate how much one can gain by incorporating acceleration, which requires little additional effort over gradient descent. Over gradient descent, acceleration requires only to update the velocity by the velocity evolution in the previous section, and the density evolution. Both these evolutions are cheap to compute since they only involve local updates. Note the gradient descent of the potential U is given by choosing $v = -\nabla U(\phi)$, the other evolution equation for the mapping ϕ (3.2) and ψ (3.7) remains the same, and the density evolution is not performed since standard gradient descent does not have a concept of mass. We note that we implement the equations as they are, and there is no additional processing that is now common in optical flow methods (e.g., no smoothing images nor derivatives, no special derivative filters, no multi-scale techniques, no use of robust norms, median filters, etc) to illustrate the advantage of the optimizer. Although our equations are for diffeomorphisms on all of $\mathbb{R}^n$, in practice be have finite images, and the issue of boundary conditions come up. For simplicity to illustrate our ideas, we choose periodic boundary conditions. Our numerical scheme is used in these experiments are given in Appendix 6.5. Our intention is to show that simply by using acceleration, one can get an impractical algorithm (gradient descent) to become practical, especially with respect to speed.

In our first set of experiments (Sub-sections 4.1.1 to 4.1.3), we choose the step size to satisfy CFL conditions. For ordinary gradient descent we choose $\Delta t < 1/(4\alpha)$, for accelerated gradient descent we have the additional evolution of the velocity (3.24), and our numerical scheme has CFL condition $\Delta t < 1/\max_{x\in\Omega}\{|v(x)|, |Dv(x)|\}$. Also,

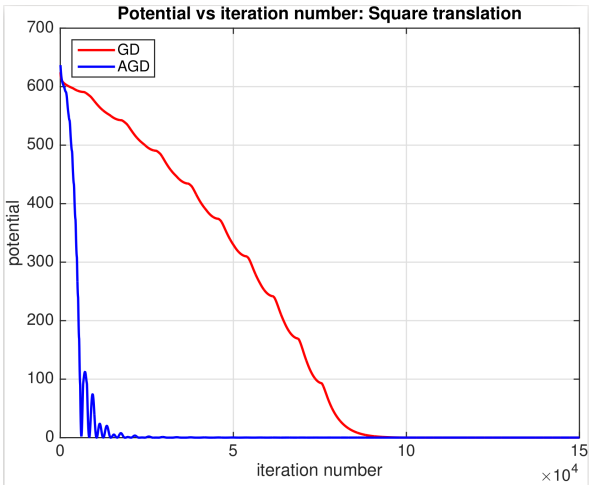


Fig. 4: **Convergence Comparison**: Two binary images with squares in which the square is translated are registered. The value of the functional (to be minimized) versus the iteration number is shown for both gradient descent (GD) and accelerated gradient descent (AGD).

because there is a diffusion according to regularity, we found that $\Delta t < 1/(4\alpha \cdot \max_{x\in\Omega}\{|v(x)|, |Dv(x)|\})$ gives stable results. The step size for accelerated gradient descent is lower in our experiments than accelerated gradient descent. The initialization is $\phi(x) = \psi(x) = x$, $v(x) = 0$, and $\rho(x) = 1/|\Omega|$ where $|\Omega|$ is the area of the domain of the image. The algorithm implemented is shown in Algorithm 6.1.

4.1.1. Convergence analysis. In this experiment, the images are two white squares against a black background. The sizes of the squares are 50×50 pixels wide, and the square (of size 20×20) in the first image is translated by 10 pixels to form the second image. Small images are chosen due to the fact gradient descent is too impractically slow for reasonable sized images without multi-scale approaches that even modest sized images (e.g., 256×256) do not converge in a reasonable amount of time, and we will demonstrate this in an experiment later. Figure 4 shows the plot of the potential energy (3.32) of both gradient descent and accelerated gradient descent as the evolution progresses. Here $\alpha = 5$ (images are scaled between 0 and 1). Notice that accelerated gradient descent very quickly accelerates to a global minimum, surpasses the global minimum and then oscillates until the friction term slows it down and then it converges very quickly. Notice that this behavior is expected since accelerated gradient descent is not a strict descent method (it does not necessarily decrease the potential energy each step). Gradient descent very slowly decreases the energy each iteration and eventually converges.

We now repeat the same experiment, but with different images to show that this behavior is not restricted to the particular choice of images, one a translation of the other. To this end, we choose the images again to be 50×50. The first image has a square that is 17×17 and the second image has a rectangle of size 20×14 and is translated by 8 pixels. We choose the regularity $\alpha = 2$, since the regularity should be chosen smaller to account for the stretching and squeezing, resulting in a non-smooth flow field. A plot of results of this simulation is shown in Figure 5. Again accelerated gradient accelerates very quickly at the start, then oscillates and the oscillations die down and then it converges. This time the potential does not go to zero since the final flow is not a translation and thus the regularity term is non-zero. Gradient descent converges faster than the case of translation due to larger α and thus larger step size. However, it appears to be stuck in a higher energy configuration. In fact, gradient descent has not fully converged - gradient descent is slow in adapting to the scale changes and becomes extremely slow in stretching and squeezing in different directions. We verify that gradient descent has not fully converged by plotting just the first term of the potential, i.e., the reconstruction error, which is zero for accelerated gradient descent at convergence, indicating that the flow correctly reconstructs I_0 from I_1. On the other hand, gradient descent has an error of about 50, indicating the flow does not fully warp I_1 to I_0, and therefore it not the correct flow. This does not appear to be a local minimum, just slow convergence.

We again repeat the same experiment, but with real images from a cardiac MRI sequence, in which the heart

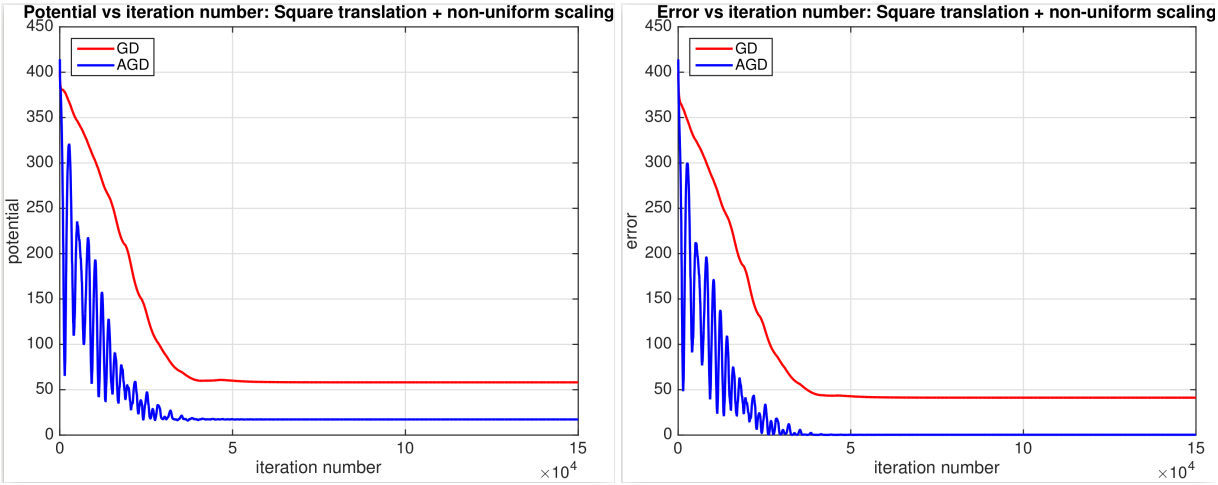


Fig. 5: **Convergence Comparison**: Two images are registered, each are binary images. The first is a square and the second image is a translated and non-uniformly scaled version of the square in the first image. [Left]: The cost functional to be minimized versus the iteration number is shown for both gradient descent (GD) and accelerated gradient descent (AGD). AGD converges to a lower energy solution quicker. [Right]: Note that GD did not fully converge as the convergence is extremely slow in obtaining fine scale details of the non-uniform scaling. This is verified by plotting the image reconstruction error: $\|I_1 \circ \phi - I_0\|$, which shows that AGD reconstructs I_0 with zero error.

beats. The transformation relating the images is a general diffeomorphism that is not easily described as in the previous experiments. The images are of size 256×256. We choose $\alpha = 0.02$. A plot of the potential versus iteration number for both gradient descent (GD) and accelerated gradient descent (AGD) is shown in the left of Figure 6. The convergence is quicker for accelerated gradient descent. The right of Figure 6 shows the original images and the images warped under both the result from gradient descent and accelerated gradient descent, and that they both produce a similar correct warp, but accelerated gradient obtains the warp in much fewer iterations.

4.1.2. Convergence analysis versus parameter settings. We now analyze the convergence of accelerated gradient descent and gradient descent as a function of the regularity α and the image size. To this end, we first analyze an image pair of size 50×50 in which one image has a square of size 16×16 and the other image is the same square translated by 7 pixels. We now vary α and analyze the convergence. In the left plot of Figure 7, we show the number of iterations until convergence versus the regularity α. As α increases, the number of iterations for both gradient descent and accelerated gradient descent increase as expected since there is a inverse relationship between α and the step size. However, the number of iterations for accelerated gradient descent grows more slowly. In all cases, the algorithm is run until the flow field between successive iterations does not change according to a fixed tolerance. In all cases, the flow achieves the ground truth flow.

Next, we analyze the number of convergence iterations versus the image size. To this end, we again consider binary images with squares of size 16×16 and translated by 7 pixels. However, we vary the image size from 50×50 to 200×200. We fix $\alpha = 8$. Now we show the number of iterations to convergence versus the image size. This is shown in the right plot of Figure 7. Gradient descent is impractically slow for all the sizes considered, and the number of iterations quickly increases with image size (it appears to be an exponential growth). Accelerated gradient descent, surprisingly, appears to have very little or no growth with respect to the image size. Of course one could use multi-scaling pyramid approaches to improve gradient descent, but as soon as one goes to finer scales, gradient descent is incredibly slow even when the images are related by small displacements. Simple acceleration makes standard gradient descent scalable with just a few extra local updates.

4.1.3. Analysis of Robustness to Noise. We now analyze the robustness of gradient descent and accelerated gradient descent to noise. We do this to simulate robustness to undesirable local minima. We choose to use salt and

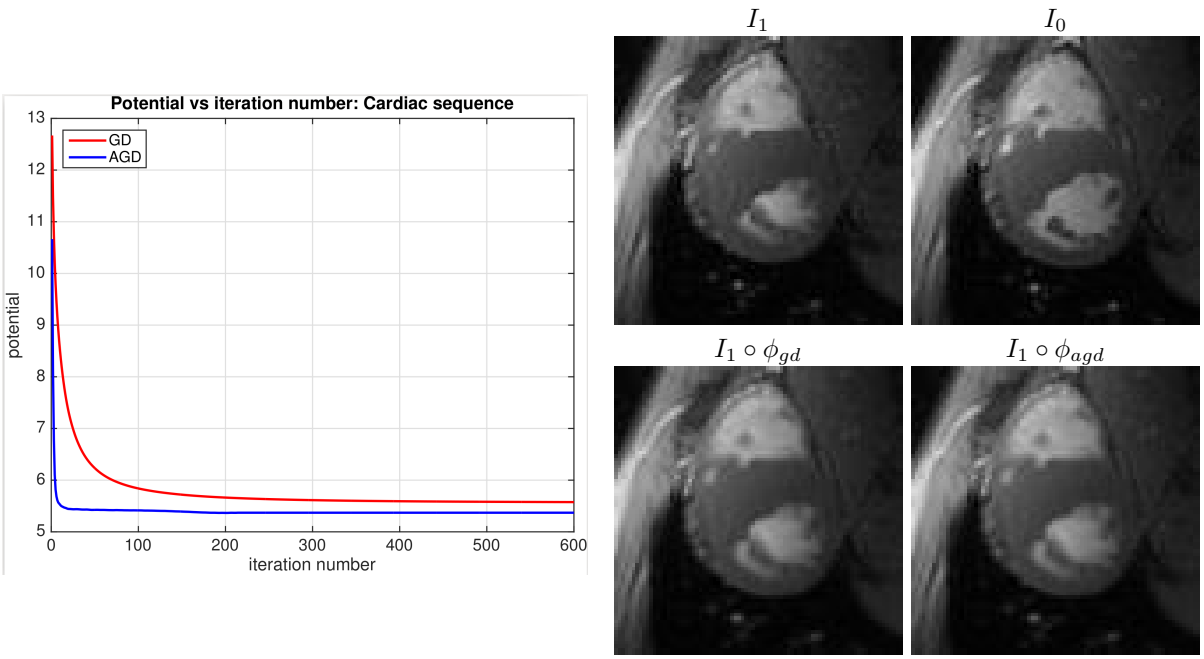


Fig. 6: **Convergence Comparison**: Two MR cardiac images from a sequence are registered. The images are related through a general deformation. [Left]: A plot of the potential versus the iteration number in the minimization using gradient descent (GD) and accelerated gradient descent (AGD). AGD converges at a quicker rate. [Right]: The original images and the back-warped images using the recovered diffeomorphisms. Note that $I_1 \circ \phi$ should appear close to I_0. Both methods seem to recover a similar transformation, but AGD recovers it faster.

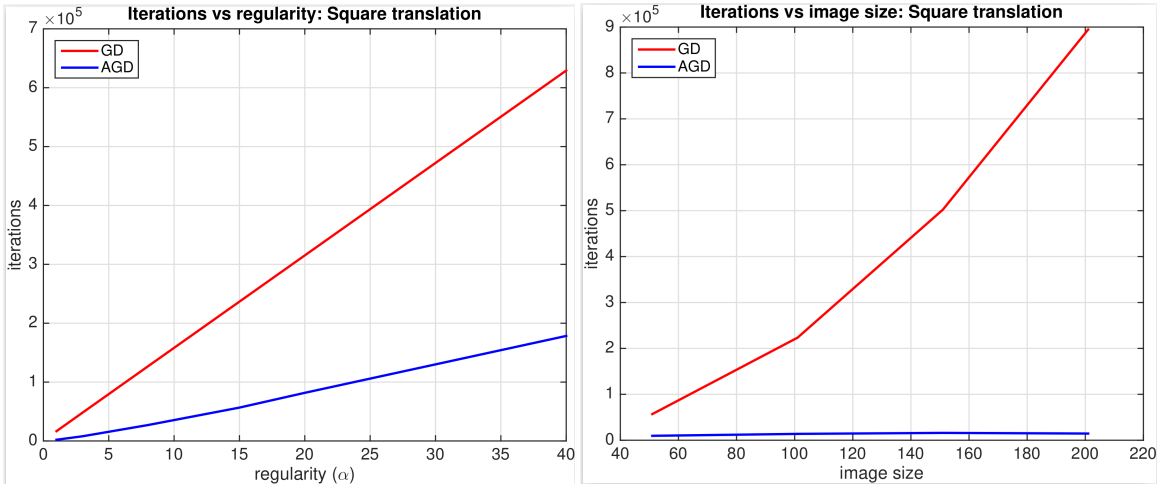


Fig. 7: [Left]: **Convergence Comparison as a Function of Regularity**: Two binary images (a square and a translated square) are registered with varying amounts of regularization α for gradient descent (GD) and accelerated gradient descent (AGD). [Right]: **Convergence Comparison as a Function of Image Size**: We keep the squares in the images and $\alpha = 3$ fixed, but we vary the size (height and width) of the image and compare GD with AGD. Very quickly, gradient descent becomes impractical due to extremely slow convergence.

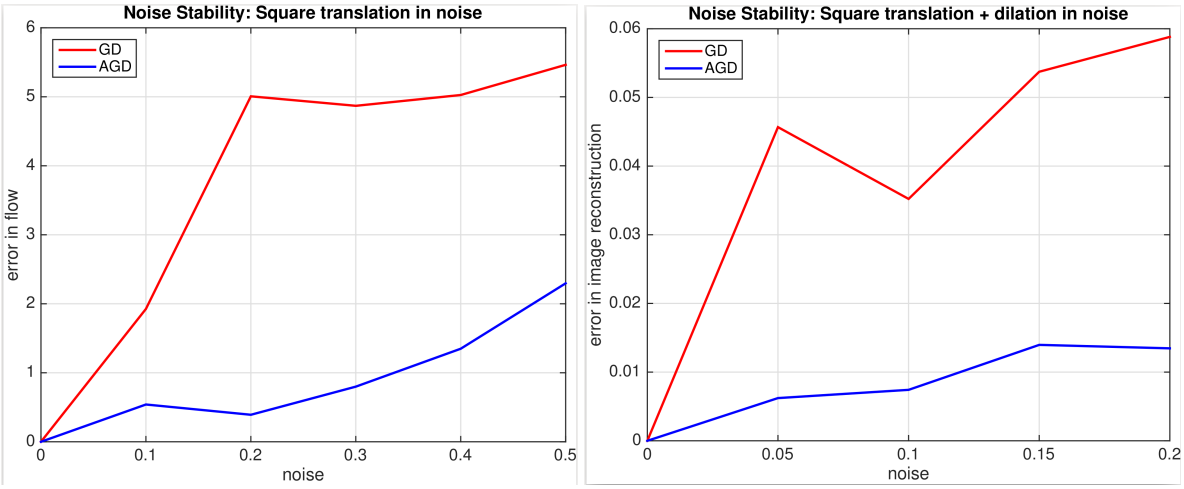


Fig. 8: **Analysis of Stability to Noise**: We add salt and pepper noise with varying intensity to binary images and then register the images. We plot the error in the recovered flow of both gradient descent (GD) and accelerated gradient descent (AGD) versus the level of noise. The value of α is kept fixed. The error is measured by the average endpoint error of the flow. [Left]: The first image is formed from a square and the second image is the same square but translated. [Right]: The first image is a square and the second image is the non-uniformly scaled and translated square. The error is measured as the average image reconstruction error.

pepper noise to model possible clutter in the image. We consider images of size 50×50. We fix $\alpha = 1$ in all the simulations and vary the noise level; of course one could increase α to increase robustness to noise. However, we are interested in understanding the robustness to noise of the optimization algorithms themselves rather than changing the potential energy to better cope with noise. First, we consider a square of size 16×16 in the first binary image and the same square translated by 4 pixels in the second image. We plot the error in the flow (measured as the average endpoint error of the flow returned by the algorithm against ground truth flow) versus the noise level. The result is shown in the left plot of Figure 8. This shows that accelerated gradient descent degrades much slower than gradient descent. Figure 9 shows visual comparison of the final results where we show $I_1 \circ \phi$ and compare it to I_0 for both accelerated gradient descent and gradient descent.

We repeat the same experiment to show that this trend is not just with this configuration of images. To this end, we experiment with 50×50 images one with a square of size 15×15 and a rectangle that is size 20×10 and translated by 5 pixels. We again fix the regularity to $\alpha = 1$. The result of the experiment is plotted in the right of Figure 8. A similar trend of the previous experiment is observed: accelerated gradient descent degrades much less than gradient descent. Note we have measured accuracy as the average reconstruction error with the original (non-noisy) images. This is because the ground truth flow is not known. Figure 10 shows visual comparison of the final results.

4.2. Lagrangian Approach and Comparison to Standard Optical Flow. We now test our method on the common benchmark for optical flow problems, the Middlebury data set [4]. We now apply accelerated gradient descent to optical flow, using the Lagrangian formulation (3.2). The numerical discretization for this method is shown in Appendix 6.6 and the algorithm is shown in Algorithm 6.2. We compare accelerated gradient descent to another general purpose optimizer (applicable to many variational cost functionals like our method) that is a common optimizer of choice in variational optical flow problems [52]. This optimizer works by iteratively linearizing optical flow around the current solution, solving the linear system typically through conjugate gradient. As is typical, a pyramid is used to solve the problem on coarse-scales efficiently, which is then upsampled to the next finer scale and is used as initialization to the aforementioned iterative linearization. We also apply this pyramid scheme in our accelerated scheme. We do not use other common techniques (e.g., median filtering, texture enchancement) for optical flow as we wish to understand the effect of the optimizers alone. We use a down-sampling factor of 2 for each level to construct the

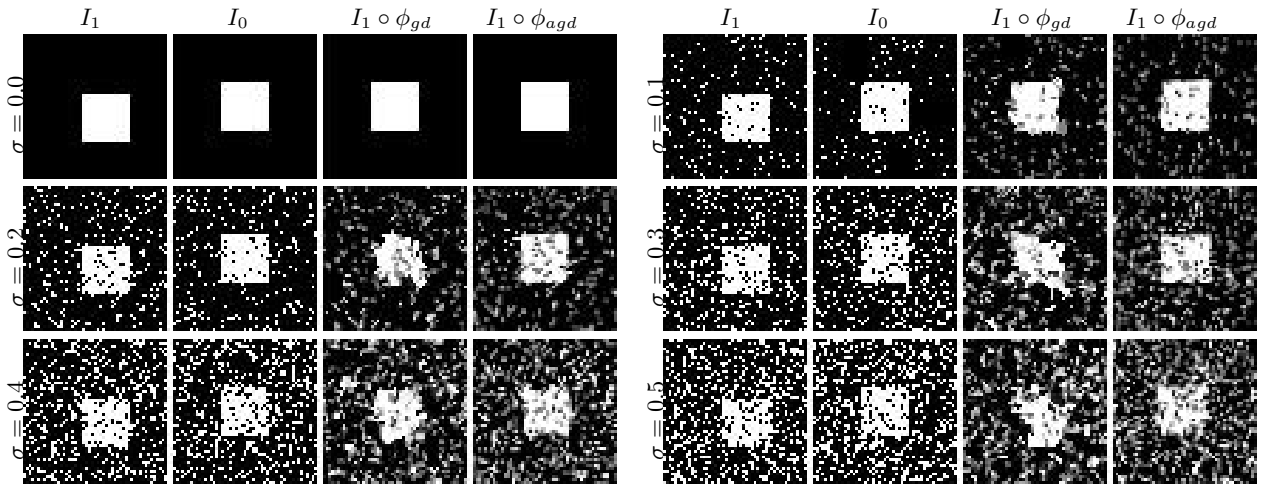


Fig. 9: **Visual Comparison on Square Translation in Noise Experiment**. The above show the visual results of the noise robustness experiment. For each row group of images: the two original images, the warped image by gradient descent, and the warped image by accelerated gradient descent. The last two images should resemble the second if the registration is correct.

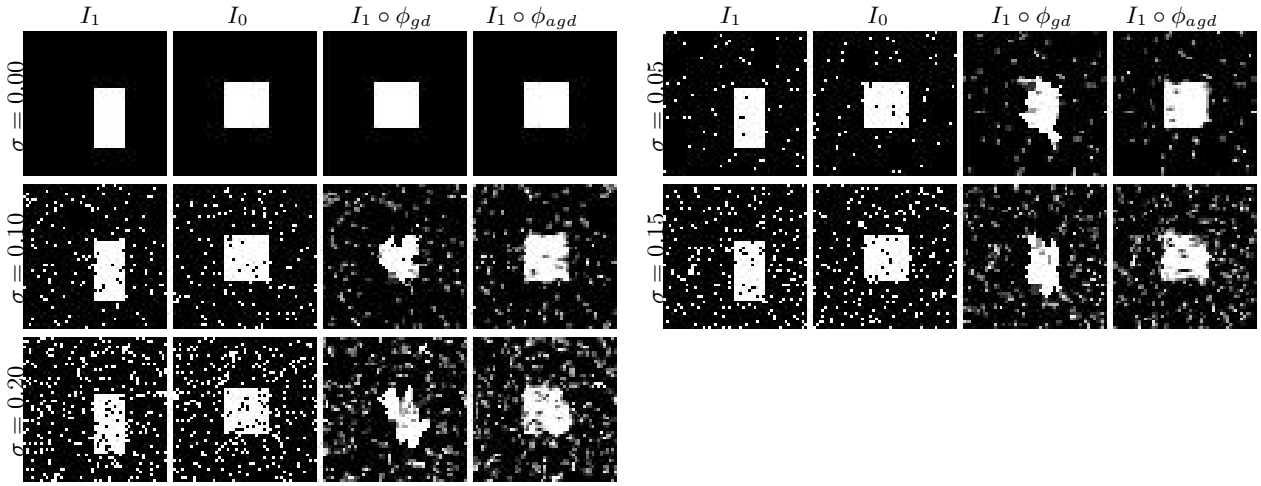


Fig. 10: **Visual Comparison on Square Non-Uniform Scaling and Translation in Noise Experiment**. The above show the visual results of the noise robustness experiment. For each row group of images: the two original images, the warped image by gradient descent, and the warped image by accelerated gradient descent. The last two images should resemble the second if the registration is correct.

pyramid [4]

The optical flow data set [4] that was used in these experiments is the Middlebury dataset, a standard benchmark for optical flow, and can be found on https://vision.middlebury.edu/flow/data/. Images ranged in resolution from 420x380 to 640x480, and consist of 7 real scenes with camera motion as well as object motion and deformation. The ground truth dense optical flow is provided in this dataset. The accuracy of the optical flow on this dataset is measured with the average angular error (AAE), which measures the average angular difference between the result and ground

[4] Source code is publicly available: github.com/minasbenyamin/Diffeomorphisms.

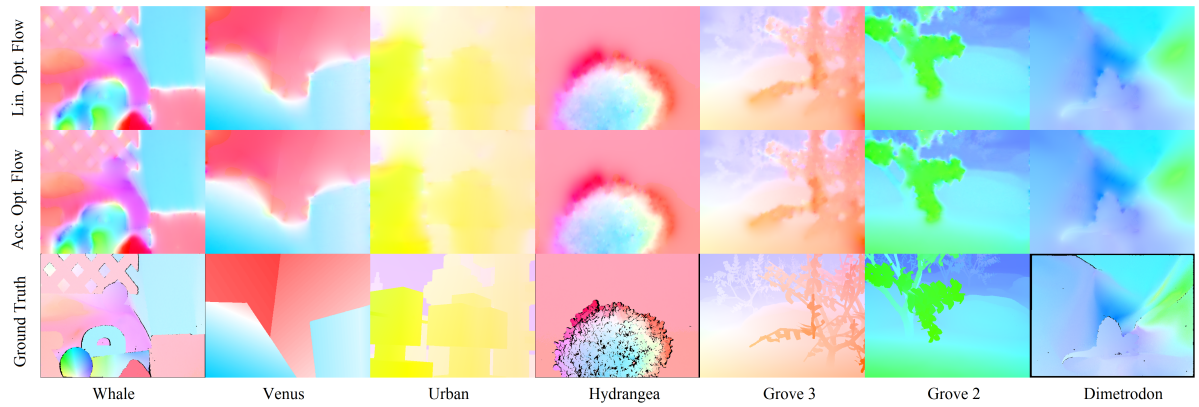


Fig. 11: Converged results for Linear Optical Flow and Accelerated Optical Flow run on the Middlebury data set. Both methods converge to the same local minimum, with the advantage of accelerated being speed. Note the black areas indicate occlusion, which are excluded from error computation according to the benchmark. Seven image pairs were used for the experiment.

truth displacement vectors, and the average end point error (AEE), which measures the average difference between end points of the displacement vectors of the result and ground truth. The regularization was kept constant throughout the entire experiment for every image. We fix the coefficient on the regularizer α at 0.04 for both accelerated and linearized optical flow, which leads to the optimal accuracy for both methods. We compute the optical flow by minimizing the Horn & Schunck cost functional (3.32) for both optimizers.

Run times (and errors in optical flow) for each pyramid level for both methods are shown in Table **??**; we also display the speed up factor giving the ratio of performance improvement in run time of accelerated over the linearization approach. Results are averages over the entire dataset. Both methods appear to converge to the same local minimum and thus have similar accuracy, but the Accelerated Optical Flow method has almost a 10x increase in speed over the standard linear approach and has a roughly 9.5x improvement in speed overall when the pyramid scheme is utilized. A time and accuracy breakdown for each level of the pyramid is given (note that each pyramid level uses as initialization the result from the previous pyramid level). A visual illustration showing the converged results is also provided in Fig. 11, confirming that both methods converge to nearly the same local optimizer, with accelerated optical flow performing significantly better in speed.

5. Conclusion. We have generalized accelerated optimization, in particular Nesterov's scheme, to infinite dimensional manifolds. This method is general and applies to optimizing any functional on an infinite dimensional manifold. We have demonstrated this for the class of diffeomorphisms motivated by variational optical flow problems in computer vision. The main objective of the paper was to introduce the formalism and derive the evolution equations that are PDEs. The evolution equations are natural extensions of mechanical principles from fluid mechanics, and in particular connect to optimal mass transport. They require additional evolution equations over gradient descent, i.e., a velocity evolution and a density evolution, but that does not significantly add to the cost of L^2 gradient descent per iteration since the updates are all local, i.e., computation of derivatives. Our numerical scheme to implement these equations used entropy conditions, which were employed to cope with shocks and fans of the underlying PDE. Experiments demonstrated the advantages of speed and robustness to local minima over gradient descent, and illustrated the behavior of accelerated gradient descent. Just by simple acceleration, gradient descent, unusable in practice due to scalability with image size, became usable. Improvements granted by acceleration are quite compelling when tested against standard state-of-the-art general purpose optimizers for optical flow. Our Accelerated Optical Flow method had a performance up lift of nearly a factor of 10 in speed.

One area that should be explored further is the choice of the time-explicit functions a, b in the generalized Lagrangian. These were chosen to coincide with the choices to produce the continuum limit of Nesterov's scheme finite dimensions (and a constant damping was explored), which are designed for the convex case to yield optimal conver-

Middlebury Benchmark Results

Pyramid Level-1 (Res: 1/32)	**Time to Converge (sec):**	**AAE (rad):**	**AEE (pixels):**	**Speed Up**
Linearized Optical Flow	0.299	0.290	2.779	
Accelerated Optical Flow	0.209	0.290	2.774	1.430
Pyramid Level-2 (Res: 1/16)	**Time to Converge (sec):**	**AAE (rad):**	**AEE (pixels):**	**Speed Up**
Linearized Optical Flow	0.476	0.150	1.679	
Accelerated Optical Flow	0.296	0.150	1.685	1.608
Pyramid Level-3 (Res: 1/8)	**Time to Converge (sec):**	**AAE (rad):**	**AEE (pixels):**	**Speed Up**
Linearized Optical Flow	2.085	0.114	1.242	
Accelerated Optical Flow	0.952	0.115	1.249	2.191
Pyramid Level-4 (Res: 1/4)	**Time to Converge (sec):**	**AAE (rad):**	**AEE (pixels):**	**Speed Up**
Linearized Optical Flow	7.862	0.102	1.045	
Accelerated Optical Flow	2.279	0.102	1.046	3.450
Pyramid Level-5 (Res: 1/2)	**Time to Converge (sec):**	**AAE (rad):**	**AEE (pixels):**	**Speed Up**
Linearized Optical Flow	90.507	0.094	0.845	
Accelerated Optical Flow	10.763	0.094	0.851	8.409
Pyramid Level-6 (Res: 1)	**Time to Converge (sec):**	**AAE (rad):**	**AEE (pixels):**	**Speed Up**
Linearized Optical Flow	1131.373	0.090	0.701	
Accelerated Optical Flow	114.307	0.090	0.700	9.898
Cumulative (All Levels)	**Time to Converge (sec):**	**AAE (rad):**	**AEE (pixels):**	**Speed Up**
Linearized Optical Flow	1232.602	0.090	0.701	
Accelerated Optical Flow	128.806	0.090	0.700	9.569

Table 1: Performance comparison of Linearized Optical Flow against Accelerated Optical Flow for each level of the pyramid. The performance improvement of accelerated optical flow is close to an order of magnitude. Both methods arrive at nearly the same local minima. AAE and AEE are average angular error and end point error respectively. Note the quantities above represent average values over all pairs of images in the dataset.

gence. Since the energies that we consider are non-convex, these may no longer be optimal. Of interest would be a design principle for choosing a, b so as to obtain optimal convergence rates. A follow-up question would then be whether the discretization of the PDEs gives optimal rates in discrete-time. Another issue is that we assumed that the domain of the diffeomorphism was $\mathbb{R}^n$, but images are compact; we by-passed this complication by assuming periodic boundary conditions in the Eulerian formulation. Future work will look into proper treatment of the boundary of compact regions that can evolve.

6. Appendix.

6.1. Functional Gradients.

DEFINITION 6.1 (Functional Gradients). *Let $U : \mathit{Diff}(\mathbb{R}^n) \to \mathbb{R}$. The gradient (or functional derivative) of U with respect to $\phi \in \mathit{Diff}(\mathbb{R}^n)$, denoted $\nabla U(\phi)$, is defined as the $\nabla U(\phi) \in T_\phi \mathit{Diff}(\mathbb{R}^n)$ that satisfies*

$$\delta U(\phi) \cdot v = \int_{\phi(\mathbb{R}^n)} \nabla U(\phi)(x) \cdot v(x) \, \mathrm{d}x \tag{6.1}$$

for all $v \in T_\phi \mathit{Diff}(\mathbb{R}^n)$. The left hand side is the directional derivative and is defined as

$$\delta U(\phi) \cdot v := \frac{\mathrm{d}}{\mathrm{d}\varepsilon} U(\phi + \varepsilon v)\Big|_{\varepsilon=0}. \tag{6.2}$$

Note that $(\phi + \varepsilon v)(x) = \phi(x) + \varepsilon v(\phi(x))$ for $x \in \mathbb{R}^n$.

We now show the computation of the gradient for the illustrative potential (3.33) used in this paper. First, let us

consider the data term $U_1(\phi) = \int_{\mathbb{R}^n} |I_1(\phi(x)) - I_0(x)|^2 \,\mathrm{d}x$ then

$$\delta U_1(\phi)\cdot\delta\phi = \int_{\mathbb{R}^n} 2(I_1(\phi(x))-I_0(x))DI_1(\phi(x))\widehat{\delta\phi}(x)\,\mathrm{d}x = \int_{\phi(\mathbb{R}^n)} 2(I_1(x)-I_0(\psi(x)))DI_1(x)\delta\phi(x)\det\nabla\psi(x)\,\mathrm{d}x,$$

where $\widehat{\delta\phi} = \delta\phi\circ\phi$, $\psi = \phi^{-1}$ and we have performed a change of variables. Thus, $\nabla U_1 = 2\nabla I_1(I_1 - I_0\circ\psi)\det\nabla\psi$. Now consider the regularity term $U_2(\phi) = \int_{\mathbb{R}^n} |\nabla(\phi(x) - x)|^2 \,\mathrm{d}x$, then

$$\delta U(\phi) = 2\int_{\mathbb{R}^n} \mathrm{tr}\left(\nabla(\phi(x)-\mathrm{id})^T\nabla\widehat{\delta\phi}(x)\right)\,\mathrm{d}x = -\int_{\mathbb{R}^n}\Delta\phi(x)^T\delta\phi(x)\,\mathrm{d}x = \int_\Omega(\Delta\phi)(\psi(x))^T\delta\phi(x)\det\nabla\psi(x)\,\mathrm{d}x.$$

Note that in integration by parts, the boundary term vanishes since we assume that $\phi(x) = x$ as $|x| \to \infty$. Thus, $\nabla U_2 = (\Delta\phi)\circ\psi\det\nabla\psi$.

6.2. Stationary Conditions.

LEMMA 6.2 (Stationary Condition for the Mapping). *The stationary condition of the action* (3.9) *for the mapping is*

$$\partial_t\lambda + \mathit{div}\left(v\lambda^T\right)^T = (\nabla\psi)^{-1}\nabla U(\phi). \tag{6.3}$$

Proof. We compute the variation of A (defined in (3.9)) with respect to the mapping ϕ. The only terms in the action that depend on the mapping are U and the Lagrange multiplier term associated with the mapping. Taking the variation w.r.t the potenial term gives

$$-\int\int_{\phi(\mathbb{R}^n)}\nabla U(\phi)\cdot\delta\phi\,\mathrm{d}x\,\mathrm{d}t.$$

Now the variation with respect to the Lagrange multiplier term:

$$\int\int_{\phi(\mathbb{R}^n)}\lambda^T[\partial_t\widehat{\delta\psi} + D(\widehat{\delta\psi})v]\,\mathrm{d}x\,\mathrm{d}t = -\int\int_{\phi(\mathbb{R}^n)}[\partial_t\lambda^T + \mathrm{div}\left(v\lambda^T\right)]\widehat{\delta\psi}\,\mathrm{d}x\,\mathrm{d}t,$$

where we have integrated by parts, the div $(\cdot)$ of a matrix means the divergence of each of the columns, resulting in a row vector, and $\widehat{\delta\psi} = \delta\psi\circ\psi$. Note that we can take the variation of $\psi(\phi(x)) = x$ to obtain

$$\widehat{\delta\psi}\circ\phi(x) + [D\psi(\phi(x))]\widehat{\delta\phi}(x) = 0,$$

or

$$\widehat{\delta\psi}(y) = -[D\psi(y)]\delta\phi(y).$$

Therefore,

$$\delta A\cdot\delta\phi = \int\int_{\phi(\mathbb{R}^n)}\left\{(\nabla\psi)\left[\partial_t\lambda + \mathrm{div}\left(v\lambda^T\right)^T\right] - \nabla U(\phi)\right\}\cdot\delta\phi\,\mathrm{d}x\,\mathrm{d}t. \tag{6.4}$$

∎

LEMMA 6.3 (Stationary Condition for the Velocity). *The stationary condition of the action* (3.9) *arising from the velocity is*

$$\rho v + (\nabla\psi)\lambda - \rho\nabla\mu = 0. \tag{6.5}$$

Proof. We compute the variation w.r.t the kinetic energy:

$$\delta T\cdot\delta v = \int_{\phi(\mathbb{R}^n)}\rho v\cdot\delta v\,\mathrm{d}x.$$

The variation of the Lagrange multiplier terms is

$$\int\int_{\phi(\mathbb{R}^n)}\lambda^T(D\psi)\delta v - \rho\nabla\mu\cdot\delta v\,\mathrm{d}x\,\mathrm{d}t = \int\int_{\phi(\mathbb{R}^n)}[(\nabla\psi)\lambda - \rho\nabla\mu]\cdot\delta v\,\mathrm{d}x\,\mathrm{d}t.$$

Therefore,

$$\delta A\cdot\delta v = \int\int_{\phi(\mathbb{R}^n)}[\rho v + (\nabla\psi)\lambda - \rho\nabla\mu]\cdot\delta v\,\mathrm{d}x\,\mathrm{d}t. \tag{6.6}$$

∎

LEMMA 6.4 (Stationary Condition for the Density). *The stationary condition of the action* (3.9) *arising from the velocity is*

$$\partial_t \mu + (D\mu)v = \frac{1}{2}|v|^2. \tag{6.7}$$

Proof. Note that the terms that contain the density in (3.9) are the kinetic energy and the Lagrange multiplier corresponding to the density. We see that

$$\delta A \cdot \delta \rho = \int \int_{\phi(\mathbb{R}^n)} \frac{1}{2}|v|^2 \delta\rho - (\partial_t \mu + \nabla\mu \cdot v)\delta\rho \, \mathrm{d}x \, \mathrm{d}t, \tag{6.8}$$

which yields the lemma. ∎

6.3. Velocity Evolution.

LEMMA 6.5. *Given that* $(\nabla\psi)\lambda = w$*, we have that*

$$\partial_t \lambda + (D\lambda)v + \lambda \mathit{div}\,(v) = (\nabla\psi)^{-1}[\partial_t w + (Dw)v + (\nabla v)w + w\mathit{div}\,(v)] \tag{6.9}$$

Proof. Define the Hessian as follows:

$$[D^2\psi]_{ijk} = \partial^2_{x_i x_j}\psi^k, \quad [D^2\psi(a,b)]_k = \sum_{ij} \partial^2_{x_i x_j}\psi^k a_i b_j.$$

We compute

$$\{D[(\nabla\psi)\lambda]\}_{ij} = \partial_{x_j}[(\nabla\psi)\lambda]_i = \partial_{x_j}\sum_l \partial_{x_i}\psi^l\lambda_l = \sum_l (\partial^2_{x_j x_i}\psi^l\lambda_l) + \partial_{x_i}\psi^l\partial_{x_j}\lambda_l.$$

Therefore,

$$D[(\nabla\psi)\lambda] = D^2\psi(\cdot,\cdot)\cdot\lambda + (\nabla\psi)(D\lambda)$$

Since $D[(\nabla\psi)\lambda] = Dw$ then solving for $D\lambda$ gives

$$D\lambda = (\nabla\psi)^{-1}[Dw - D^2\psi(\cdot,\cdot)\cdot\lambda],$$

so

$$(D\lambda)v = (\nabla\psi)^{-1}[(Dw)v - D^2\psi(\cdot,v)\cdot\lambda]. \tag{6.10}$$

Now differentiating $(\nabla\psi)\lambda = w$ w.r.t t, we have

$$(\nabla\partial_t\psi)\lambda + (\nabla\psi)\partial_t\lambda = \partial_t w, \quad \text{or} \quad \partial_t\lambda = (\nabla\psi)^{-1}[\partial_t w - (\nabla\partial_t\psi)\lambda]$$

Note that $\partial_t\psi = -(D\psi)v$ so

$$\partial_t\lambda = (\nabla\psi)^{-1}\left\{\partial_t w + \nabla[(D\psi)v]\lambda\right\}. \tag{6.11}$$

Now computing $\nabla[(D\psi)v]$ yields

$$\{\nabla[(D\psi)v)]\}_{lk} = \partial_{x_l}\sum_i \partial_{x_i}\psi^k v^i = \sum_i \partial_{x_l}\partial_{x_i}\psi^k v^i + \partial_{x_i}\psi^k\partial_{x_l}v^i.$$

Then multiplying the above matrix by λ gives

$$\{\nabla[(D\psi)v)]\lambda\}_l = \sum_{ik} \partial_{x_l}\partial_{x_i}\psi^k v^i\lambda^k + \partial_{x_i}\psi^k\partial_{x_l}v^i\lambda^k,$$

which in matrix form is

$$\nabla[(D\psi)v)]\lambda = D^2\psi(\cdot,v)\cdot\lambda + (\nabla v)(\nabla\psi)\lambda = D^2\psi(\cdot,v)\cdot\lambda + (\nabla v)w$$

Therefore, (6.11) becomes

$$\partial_t\lambda = (\nabla\psi)^{-1}[\partial_t w + D^2\psi(\cdot,v)\cdot\lambda + (\nabla v)w].$$

Combining the previous with (6.10) and noting that $\lambda\mathrm{div}\,(v) = (\nabla\psi)^{-1}w\mathrm{div}\,(v)$ yields

$$\partial_t\lambda + (D\lambda)v + \lambda\mathrm{div}\,(v) = (\nabla\psi)^{-1}[\partial_t w + (Dw)v + (\nabla v)w + w\mathrm{div}\,(v)].$$ ∎

LEMMA 6.6. *If* $w = \rho(\nabla\mu - v)$, *then*

$$\partial_t w + (Dw)v + (\nabla v)w + w\mathit{div}\,(v) = -\rho[\partial_t v + (Dv)v]. \tag{6.12}$$

Proof. Differentiating $w = \rho(\nabla\mu - v)$, we have

$$\begin{aligned}\partial_t w &= (\partial_t\rho)(\nabla\mu - v) + \rho(\nabla\partial_t\mu - \partial_t v)\\ Dw &= (\nabla\mu - v)(D\rho) + \rho[D(\nabla\mu) - Dv].\end{aligned}$$

Therefore,

$$\begin{aligned}\partial_t w + (Dw)v + (\nabla v)w + w\mathrm{div}\,(v) &= (\nabla\mu - v)(\partial_t\rho + \nabla\rho\cdot v) + \rho[\nabla\partial_t\mu - \partial_t v + D(\nabla\mu)v - (Dv)v]\\ &\quad + \rho(\nabla v)(\nabla\mu - v) + \rho(\nabla\mu - v)\mathrm{div}\,(v)\\ &= (\nabla\mu - v)(\partial_t\rho + \nabla\rho\cdot v + \rho\mathrm{div}\,(v))\\ &\quad + \rho[\nabla\partial_t\mu - \partial_t v + D(\nabla\mu)v - (Dv)v + (\nabla v)(\nabla\mu - v)].\end{aligned}$$

Note that $\partial_t\rho + \nabla\rho\cdot v + \rho\mathrm{div}\,(v) = \partial_t\rho + \mathrm{div}\,(\rho v) = 0$, due to the continuity equation. Therefore,

$$\begin{aligned}\partial_t w + (Dw)v + (\nabla v)w + w\mathrm{div}\,(v) &= \rho[-\partial_t v - (Dv)v - (\nabla v)v + \nabla\partial_t\mu + D(\nabla\mu)v + (\nabla v)(\nabla\mu)]\\ &= \rho\left\{-\partial_t v - (Dv)v - (\nabla v)v + \nabla[\partial_t\mu + (D\mu)v]\right\}.\end{aligned}$$

By the stationary condition for the density, $\partial_t\mu + (D\mu)v = 1/2|v|^2$, so $\nabla[\partial_t\mu + (D\mu)v] = (\nabla v)v$, which gives the lemma. ∎

THEOREM 6.7 (Velocity Evolution). *The evolution equation for the velocity arising from the stationarity of the action integral is*

$$\rho[\partial_t v + (Dv)v] = -\nabla U(\phi). \tag{6.13}$$

Proof. This is a combination of Lemmas 6.2, 6.5, and 6.6. ∎

6.4. Stationary Conditions for the Dissipative Case.

THEOREM 6.8 (Stationary Conditions for the Path of Least Action: Dissipative Case). *The stationary conditions of the path for the action*

$$A = \int \left[aT(v) - bU(\phi)\right]\,\mathrm{d}t + \int\int_{\mathbb{R}^n} \lambda^T[\partial_t\psi_t + (D\psi)v]\,\mathrm{d}x\,\mathrm{d}t - \int\int_{\mathbb{R}^n}[\partial_t\mu + \nabla\mu\cdot v]\,\rho\,\mathrm{d}x\,\mathrm{d}t, \tag{6.14}$$

are

$$\partial_t\lambda + (D\lambda)v + \lambda\mathit{div}\,(v) = b(\nabla\psi)^{-1}\nabla U(\phi) \tag{6.15}$$

$$a\rho v + (\nabla\psi)\lambda - \rho\nabla\mu = 0 \tag{6.16}$$

$$\partial_t\mu + \nabla\mu\cdot v = \frac{1}{2}a|v|^2. \tag{6.17}$$

Proof. Note that

$$\begin{aligned}\nabla[bU](\phi) &= b\nabla U(\phi)\\ \delta[aT]\cdot\delta\rho &= \int_{\phi(\mathbb{R}^n)}\frac{1}{2}a|v|^2\delta\rho\,\mathrm{d}x\\ \delta[aT]\cdot\delta v &= \int_{\phi(\mathbb{R}^n)} a\rho v\cdot\delta v\,\mathrm{d}x.\end{aligned}$$

Therefore, using (6.4) and replacing $\nabla U(\phi)$ with $b\nabla U(\phi)$, we have

$$\delta A\cdot\delta\phi = \int\int_{\phi(\mathbb{R}^n)}\left\{(\nabla\psi)\left[\partial_t\lambda + \mathrm{div}\left(v\lambda^T\right)^T\right] - b\nabla U(\phi)\right\}\cdot\delta\phi\,\mathrm{d}x\,\mathrm{d}t,$$

which yields the stationary condition on the mapping. Also, updating (6.6) yields

$$\delta A \cdot \delta v = \int \int_{\phi(\mathbb{R}^n)} [a\rho v + (\nabla \psi)\lambda - \rho \nabla \mu] \cdot \delta v \, \mathrm{d}x \, \mathrm{d}t,$$

which yields the stationary condition for the velocity. Finally, updating (6.8) yields

$$\delta A \cdot \delta \rho = \int \int_{\phi(\mathbb{R}^n)} \frac{1}{2} a|v|^2 \delta \rho - (\partial_t \mu + \nabla \mu \cdot v)\delta \rho \, \mathrm{d}x \, \mathrm{d}t,$$

and that yields the last stationary condition. ∎

THEOREM 6.9 (Evolution Equations for the Path of Least Action: Dissipative Case). *The evolution equations for the stationary conditions of the action in* (6.14) *is*

$$\rho[\partial_t(av) + a(Dv)v] = -b\nabla U(\phi). \tag{6.18}$$

Proof. Let $w = \rho(\nabla \mu - av)$ then

$$\begin{aligned} \partial_t w &= (\partial_t \rho)(\nabla \mu - av) + \rho(\nabla \partial_t \mu - \partial_t(av)) \\ Dw &= (\nabla \mu - av)(D\rho) + \rho[D(\nabla \mu) - aDv]. \end{aligned}$$

Then

$$\begin{aligned} \partial_t w + (Dw)v + (\nabla v)w + w\mathrm{div}\,(v) &= g(\nabla \mu - av)(\partial_t \rho + \nabla \rho \cdot v) + \rho[\nabla \partial_t \mu - \partial_t(av) + D(\nabla \mu)v - a(Dv)v] \\ &\quad + \rho(\nabla v)(\nabla \mu - av) + \rho(\nabla \mu - av)\mathrm{div}\,(v) \\ &= (\nabla \mu - av)(\partial_t \rho + \nabla \rho \cdot v + \rho \mathrm{div}\,(v)) \\ &\quad + \rho[\nabla \partial_t \mu - \partial_t(av) + D(\nabla \mu)v - a(Dv)v + (\nabla v)(\nabla \mu - av)] \\ &= \rho\left\{-\partial_t(av) - a(Dv)v - a(\nabla v)v + \nabla[\partial_t \mu + (D\mu)v]\right\} \\ &= \rho\left\{-\partial_t(av) - a(Dv)v\right\}. \end{aligned}$$

By Lemma 6.5 and the previous expression, we have our result. ∎

6.5. Discretization. We present the discretization of the velocity PDE (3.24) first. In one dimension, the terms involving v are Burger's equation, which is known to produce shocks. We thus use an entropy scheme. Writing the PDE component-wise, we get

$$\partial_t v_1 = -\frac{1}{2}\partial_{x_1}(v_1)^2 - v_2 \partial_{x_2} v_1 - \frac{3}{t} v_1 - \frac{1}{\rho}(\nabla U)_1 \tag{6.19}$$

$$\partial_t v_2 = -\frac{1}{2}\partial_{x_2}(v_2)^2 - v_1 \partial_{x_1} v_2 - \frac{3}{t} v_2 - \frac{1}{\rho}(\nabla U)_2, \tag{6.20}$$

where the subscript indicates the component of the vector. We use forward Euler for the time derivative, and for the first term on the right hand side, we use an entropy scheme for Burger's equation which results in the following discretization:

$$\partial_{x_1}(v_1)^2(x) \approx \max\{v_1(x), 0\}^2 - \min\{v_1(x), 0\}^2 + \min\{v_1(x_1 + \Delta x, x_2), 0\}^2 - \max\{v_1(x_1 + \Delta x, x_2), 0\}^2, \tag{6.21}$$

where Δx is the spatial sampling size, and the $\partial_{x_2}(v_2)^2$ follows similarly. For the second term on the right hand side of (6.19), we follow the discretization of a transport equation using an up-winding scheme, which yields the following discretization:

$$v_2(x)\partial_{x_2} v_1(x) \approx v_2(x) \cdot \begin{cases} v_1(x_1, x_2) - v_1(x_1, x_2 - \Delta x) & v_2(x) > 0 \\ v_1(x_1, x_2 + \Delta x) - v_1(x_1, x_2) & v_2(x) < 0 \end{cases}. \tag{6.22}$$

With regards to the gradient of potential, if we use the potential (3.33), then all the derivatives are discretized using central differences, as the key term is a diffusion. The step size $\Delta t/\Delta x < 1/\max_x\{|v(x)|, |Dv(x)|\}$.

The backward map ψ evolves according to a transport PDE (3.7), and thus an up-winding scheme similar to the transport term in the velocity term is used. For the discretization of the continuity equation, we use a staggered grid (so that the values of v are defined in between grid points and ρ is defined at the grid points). The discretization is just the sum of the fluxes coming into the point:

(6.23)
$$-\operatorname{div}(\rho(x)v(x)) \approx \sum_{i=1}^{2}\left[-v_i(x)\begin{cases}\rho(x) & v_i(x)>0\\ \rho(x+\Delta x_i) & v_i(x)<0\end{cases} + v_i(x-\Delta x_i)\begin{cases}\rho(x-\Delta x_i) & v_1(x-\Delta x_i)>0\\ \rho(x) & v_1(x-\Delta x_i)<0\end{cases}\right],$$

where Δx_i denotes the vector of the spatial increment Δx in the i^{th} coordinate direction, $v_1(x)$ denotes the velocity defined at the midpoint between (x_1, x_2) and $(x_1 + \Delta x, x_2)$, and $v_2(x)$ denotes the velocity defined at the midpoint between (x_1, x_2) and $(x_1, x_2 + \Delta x)$. The term $\partial_t \rho(x)$ is discretized with forward Euler. This scheme is guaranteed to preserve mass.

6.5.1. Implementation. The final algorithm to optimize the potential U is shown in Algorithm 6.1 for the Eulerian implementation, which evolves the mass density. We have shown the algorithm in the general case of n dimensional data.

Algorithm 6.1 $\phi = accelMassFlow(I_0, I_1, \alpha)$

$M, N, \ldots = size(I_0)$ // n-dimensional image
$\phi_0 = \text{meshgrid}[0, .., M; 0, ..., N; \ldots]$
$v_0 = [0]_{M\times N\times n}$
$\rho^0(x) = 1/(MN)[1]_{M\times N}$ // constant mass initialization
for $k = 1, \ldots, K$ **do**
  $\Delta t = 1/(4\alpha \cdot \max_x\{|v(x)|, |Dv(x)|\})$
  Compute $\partial_{x_i}(v_i^k)^2$ using (6.21) and $v_j^k \partial_{x_j} v_i^k$ using (6.22) for all $i, j \neq i$
  $a = 3/t$ if $t > 0$, else $a = 0$ // damping factor
  $dv_i^k(x) = -\frac{1}{2}\partial_{x_i}(v_i^k)^2(x) - \sum_{j\neq i} v_j^k(x)\partial_{x_j} v_i^k(x) - a v_i^k - \frac{1}{\rho^k(x)}\nabla[U(\phi^k)]_i$
  $d\rho^k(x) = -\operatorname{div}\left(\rho^k(x)v^k(x)\right)$ using (6.23)
  $\phi^{k+1}(x) = \phi^k(x) + \Delta t \cdot v^k(\phi^k(x))$
  $v^{k+1}(x) = v^k(x) + \Delta t \cdot dv^k(x)$
  $\rho^{k+1}(x) = \rho^k(x) + \Delta t \cdot d\rho^k(x)$
  $t \leftarrow t + \Delta t$
end for
return ϕ^K // return final warp between images

6.6. Lagrangian Approach to Accelerated Optical Flow: Discretization and Choosing Damping.

6.6.1. Discretization. We start by reviewing the discretization of gradient descent and then extend the technique to accelerated gradient descent. The gradient descent PDE of energy function $U(\phi)$ takes form:

(6.24)
$$\partial_t \phi = -\nabla U(\phi)$$

We can write the first order forward difference for the gradient descent above as:

(6.25)
$$\frac{\phi(x, t+\Delta t) - \phi(x,t)}{\Delta t} = -\nabla U$$

this yields the gradient descent update as:

(6.26)
$$\phi^{n+1} = \phi^n - \Delta t \nabla U^n$$

where $\phi^n(x) = \phi(x, n\Delta t)$ is the n^{th} sampling in time of ϕ.

By contrast the accelerated descent PDE of the energy function (Lagrangian approach (3.31)) takes form (in case that $\partial_t a/a$ is a chosen to be constant and by abuse of notation labeled a, $b/a = 1$, and $\rho_0 = 1$; this would be the case of constant damping as opposed to a time-varying damping in Nesterov's method):

(6.27)
$$\partial_{tt}\phi = -a\partial_t \phi - \nabla U(\phi)$$

Applying central differences to the above, we arrive at

$$\frac{\phi(x,t+\Delta t)-2\phi(x,t)+\phi(x,t-\Delta t)}{\Delta t^2}+a\frac{\phi(x,t+\Delta t)-\phi(x,t-\Delta t)}{2\Delta t}=-\nabla U(x,t). \tag{6.28}$$

With some manipulation we can derive the update equation. Treating $\phi(x,t+\Delta t)$ as ϕ^{n+1}, $\phi(x,t)$ as ϕ^n and $\phi(x,t-\Delta t)$ as ϕ^{n-1} allows us to write the update of ϕ:

$$\phi^{n+1}(x)=\frac{2\phi^n-(1-\frac{a\Delta t}{2})\phi^{n-1}-\Delta t^2\nabla U^n}{1+\frac{a\Delta t}{2}} \tag{6.29}$$

with additional algebraic manipulation this gives:

$$\delta\phi^n=\frac{2-a\Delta t}{2+a\Delta t}\cdot\delta\phi^{n-1}-\frac{2\Delta t^2}{2+a\Delta t}\cdot\nabla U^n \tag{6.30}$$

$$\phi^n=\phi^{n-1}+\delta\phi^n \tag{6.31}$$

where ϕ is the forward map, n refers to the current iteration, Δt is the time step, a is the damping coefficient, and ∇U^n is the energy gradient at iteration n. Here we write $\delta\phi^n$ as the increment to update ϕ^{n-1}, which gives a similar equation to the usual gradient descent update (with $\delta\phi^n$ replacing the gradient). The original derivation of the update equations can be found in [8], but is restated for convenience of the reader.

In the case of the Horn & Schunck energy (3.32), the gradient of the energy is

$$-\nabla U=-\nabla I_1\circ\phi\cdot(I_1(\phi)-I_0)+\alpha\Delta\phi \tag{6.32}$$

where $\nabla I_1\circ\phi=\left[\begin{smallmatrix}\frac{\delta I_1(\phi)}{\delta x}\\ \frac{\delta I_1(\phi)}{\delta y}\end{smallmatrix}\right]$ denotes the spatial gradient of the image $I_1(\phi)$ and α is the coefficient on the regularizer. Substituting into 6.30 gives the full update:

$$\delta\phi^n=\frac{2-a\Delta t}{2+a\Delta t}\cdot\delta\phi^{n-1}+\frac{2\Delta t^2}{2+a\Delta t}\cdot\Big(-\nabla I_1(\phi^n(x))\cdot(I_1(\phi^n(x))-I_0(x))+\alpha\Delta\phi^n(x)\Big) \tag{6.33}$$

The maximum stable time step Δt for the scheme (6.30), (6.31) above can be derived using Von Neumann analysis. From [8], Δt should be chosen as $\Delta t<\frac{2}{\sqrt{Z_{max}}}$, where Z_{max} is the maximum value over all frequencies of the Fourier transform of the linearization of the homogeneous part of the gradient, ∇U. In the case of the Horn & Schunck energy, this corresponds to

$$\Delta t<\frac{2}{\sqrt{1+8\alpha}}, \tag{6.34}$$

where the above is an approximation and we assume that the image is normalized to 1.

6.6.2. Choosing the Damping Coefficient. Next we compute the optimal damping coefficient, a. To do this, we use results from [13], which computes the convergence rate of accelerated PDE as a function of the damping in the case that the energy is convex. The Horn & Schunck energy is not convex, however, the linearization of the gradient in the accelerated PDE corresponds to a convex energy that was analyzed in [13].

The original accelerated PDE is:

$$\partial_{tt}\phi+a\partial_t\phi-\alpha\Delta\phi+(I_1\circ\phi-I_0)\nabla I_1\circ\phi=0. \tag{6.35}$$

We can linearize the non-linear term and compute the optimal damping. For simplicity (as we did not find much difference in the speed of overall convergence in our experiments), we simply treat the non-linear term as zero (which is true if ϕ is near the solution as $I_1\circ\phi-I_0$ is close to zero; in practice we use a pyramid method where the solution is close to the optimal since it is initialized with the solution from the previous scale). In this case, the PDE reduces to

$$\partial_{tt}\phi+a\partial_t\phi-\alpha\Delta\phi=0, \tag{6.36}$$

which is a vector-valued version of an equation analyzed in [13]. The optimal damping is given as

$$a = 2\sqrt{\alpha\mu_1} \tag{6.37}$$

where μ_1 is the first Neumann eigenvalue of the Laplacian. The eigenvalue can be approximated as $\mu_1 \approx \frac{\pi^2}{A}$ where A is the area of the image domain (width times height of the image). This gives the damping condition that we used for our experiments as:

$$a = 2\sqrt{\alpha\mu_1} \approx 2\sqrt{\frac{\pi^2\alpha}{A}}. \tag{6.38}$$

6.6.3. Implementation. The algorithm for implementation of the Lagrangian approach for accelerated optimization is given in Algorithm 6.2. In contrast to linearized optical flow, typical in the optical flow literature, our algorithm does not require computing matrix inverses (e.g., solved with conjugate gradient). For comparison, we also show the corresponding Lagrangian gradient descent in Algorithm 6.3, which is not used in practice due to slow convergence in favor of linearized optical flow; the comparison shows that our accelerated optical flow requires just a few extra lines of code compared to simple gradient descent, without requiring matrix inverses, resulting in a faster converging algorithm than lineaized optical flow.

Algorithm 6.2 $\phi = accelFlow(I_0, I_1, a, \alpha)$

$\Delta t = \frac{2}{\sqrt{\max_x |I_0(x)| + 8\alpha}}$
$m, n = size(I_0)$
$a = 2\sqrt{\frac{\pi^2\alpha}{MN}}$ // compute optimal damping
$\phi_0 = \text{meshgrid}(0, \ldots, M; 0, \ldots, N)$
$\delta\phi_0 = [0]_{M\times N\times 2}$
for $k = 1, \ldots, K$ **do**
  $\delta\phi_k = \frac{2 - a\cdot\Delta t}{2 + a\cdot\Delta t} \cdot \delta\phi_{k-1} - \frac{2\Delta t^2}{2 + a\cdot\Delta t} \cdot \nabla U(\phi_k)$
  $\phi_k = \phi_{k-1} + \delta\phi_k$
end for
return ϕ^K // return final warp between images

Algorithm 6.3 $\phi = gradFlow(I_0, I_1, a, \alpha)$

$\Delta t = \frac{2}{\max_x |I_0(x)| + 8\alpha}$
$M, N = size(I_0)$
$\phi_0 = \text{meshgrid}(0, \ldots, M; 0, \ldots, N)$
$v_0 = [0]_{M\times N\times 2}$
for $k = 1, \ldots, K$ **do**
  $\delta\phi_k = -\Delta t \cdot \nabla U(\phi_k)$
  $\phi_k = \phi_{k-1} + \delta\phi_k$
end for
return ϕ^K // return final warp between images

REFERENCES

[1] L. AMBROSIO, N. GIGLI, AND G. SAVARÉ, *Gradient flows: in metric spaces and in the space of probability measures*, Springer Science & Business Media, 2008.
[2] S. ANGENENT, S. HAKER, AND A. TANNENBAUM, *Minimizing flows for the monge–kantorovich problem*, SIAM journal on mathematical analysis, 35 (2003), pp. 61–97.
[3] V. I. ARNOL'D, *Mathematical methods of classical mechanics*, vol. 60, Springer Science & Business Media, 2013.
[4] S. BAKER, S. ROTH, D. SCHARSTEIN, M. J. BLACK, J. P. LEWIS, AND R. SZELISKI, *A database and evaluation methodology for optical flow*, in 2007 IEEE 11th International Conference on Computer Vision, 2007, pp. 1–8.

[5] M. Bauer, M. Bruveris, and P. W. Michor, *Overview of the geometries of shape spaces and diffeomorphism groups*, Journal of Mathematical Imaging and Vision, 50 (2014), pp. 60–97.
[6] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes, *Computing large deformation metric mappings via geodesic flows of diffeomorphisms*, International journal of computer vision, 61 (2005), pp. 139–157.
[7] J.-D. Benamou and Y. Brenier, *A computational fluid mechanics solution to the monge-kantorovich mass transfer problem*, Numerische Mathematik, 84 (2000), pp. 375–393.
[8] M. Benyamin, J. Calder, G. Sundaramoorthi, and A. Yezzi, *Accelerated variational pdes for efficient solution of regularized inversion problems*, Journal of Mathematical Imaging and Vision, 62 (2019), pp. 10–36.
[9] M. J. Black and P. Anandan, *The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields*, Computer vision and image understanding, 63 (1996), pp. 75–104.
[10] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High accuracy optical flow estimation based on a theory for warping*, in European conference on computer vision, Springer, 2004, pp. 25–36.
[11] A. Bruhn, J. Weickert, and C. Schnörr, *Lucas/kanade meets horn/schunck: Combining local and global optic flow methods*, International journal of computer vision, 61 (2005), pp. 211–231.
[12] S. Bubeck, Y. T. Lee, and M. Singh, *A geometric alternative to nesterov's accelerated gradient descent*, CoRR, abs/1506.08187 (2015).
[13] J. Calder and A. Yezzi, *Pde acceleration: a convergence rate analysis and applications to obstacle problems*, Res Math Sci, (2019), pp. 6–35.
[14] G. Charpiat, R. Keriven, J.-P. Pons, and O. Faugeras, *Designing spatially coherent minimizing flows for variational problems based on active contours*, in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 2, IEEE, 2005, pp. 1403–1408.
[15] G. Charpiat, P. Maurel, J.-P. Pons, R. Keriven, and O. Faugeras, *Generalized gradients: Priors on minimization flows*, International journal of computer vision, 73 (2007), pp. 325–344.
[16] M. G. Crandall and P.-L. Lions, *Viscosity solutions of hamilton-jacobi equations*, Transactions of the American Mathematical Society, 277 (1983), pp. 1–42.
[17] M. P. Do Carmo, *Riemannian geometry*, Birkhauser, 1992.
[18] D. G. Ebin and J. Marsden, *Groups of diffeomorphisms and the motion of an incompressible fluid*, Annals of Mathematics, (1970), pp. 102–163.
[19] J. Feydy, B. Charlier, F.-X. Vialard, and G. Peyré, *Optimal transport for diffeomorphic registration*, in International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2017, pp. 291–299.
[20] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouvé, and G. Peyré, *Interpolating between optimal transport and mmd using sinkhorn divergences*, in The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 2681–2690.
[21] N. Flammarion and F. Bach, *From averaging to acceleration, there is only a step-size*, in Proceedings of Machine Learning Research, vol. 40, 2015, pp. 658–695.
[22] W. Gangbo and R. J. McCann, *The geometry of optimal transportation*, Acta Mathematica, 177 (1996), pp. 113–161.
[23] S. Ghadimi and G. Lan, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Math. Program., 156 (2016), pp. 59–99.
[24] B. K. Horn and B. G. Schunck, *Determining optical flow*, Artificial intelligence, 17 (1981), pp. 185–203.
[25] R. Hosseini and S. Sra, *An alternative to em for gaussian mixture models: Batch and stochastic riemannian optimization*, arXiv preprint arXiv:1706.03267, (2017).
[26] C. Hu, W. Pan, and J. T. Kwok, *Accelerated gradient methods for stochastic optimization and online learning*, in Advances in Neural Information Processing Systems 22, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, eds., Curran Associates, Inc., 2009, pp. 781–789.
[27] S. Ji and J. Ye, *An accelerated gradient method for trace norm minimization*, in Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, 2009, pp. 457–464.
[28] V. Jojic, S. Gould, and D. Koller, *Accelerated dual decomposition for map inference*, in Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, 2010, pp. 503–510.
[29] R. Jordan, D. Kinderlehrer, and F. Otto, *The variational formulation of the fokker–planck equation*, SIAM journal on mathematical analysis, 29 (1998), pp. 1–17.
[30] E. Klassen, A. Srivastava, M. Mio, and S. H. Joshi, *Analysis of planar shapes using geodesic paths on shape spaces*, IEEE transactions on pattern analysis and machine intelligence, 26 (2004), pp. 372–383.
[31] W. Krichene, A. Bayen, and P. L. Bartlett, *Accelerated mirror descent in continuous and discrete time*, in Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., Curran Associates, Inc., 2015, pp. 2845–2853.
[32] D. Lao and G. Sundaramoorthi, *Extending layered models to 3d motion*, in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 435–451.
[33] D. Lao, P. Zhu, P. Wonka, and G. Sundaramoorthi, *Flow-guided video inpainting with scene templates*, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 14599–14608.
[34] H. Li and Z. Lin, *Accelerated proximal gradient methods for nonconvex programming*, in Advances in Neural Information Processing Systems 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., Curran Associates, Inc., 2015, pp. 379–387.
[35] Y. Liu, F. Shang, J. Cheng, H. Cheng, and L. Jiao, *Accelerated first-order methods for geodesically convex optimization on riemannian manifolds*, in Advances in Neural Information Processing Systems, 2017, pp. 4875–4884.
[36] C. J. Maddison, D. Paulin, Y. W. Teh, B. O'Donoghue, and A. Doucet, *Hamiltonian descent methods*, arXiv preprint arXiv:1809.05042, (2018).
[37] J. E. Marsden and T. S. Ratiu, *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*, vol. 17, Springer Science & Business Media, 2013.
[38] A. Mennucci, A. Yezzi, and G. Sundaramoorthi, *Sobolev–type metrics in the space of curves*, Interfaces and Free Boundaries, (2008), pp. 423–445.

[39] M. MICHELI, P. W. MICHOR, AND D. MUMFORD, *Sobolev metrics on diffeomorphism groups and the derived geometry of spaces of submanifolds*, Izvestiya: Mathematics, 77 (2013), p. 541.
[40] P. W. MICHOR, D. MUMFORD, J. SHAH, AND L. YOUNES, *A metric on shape space with explicit geodesics*, arXiv preprint arXiv:0706.4299, (2007).
[41] M. I. MILLER, A. TROUVÉ, AND L. YOUNES, *Geodesic shooting for computational anatomy*, Journal of mathematical imaging and vision, 24 (2006), pp. 209–228.
[42] Y. NESTEROV, *A method of solving a convex programming problem with convergence rate o (1/k2)*, in Soviet Mathematics Doklady, vol. 27, 1983, pp. 372–376.
[43] Y. NESTEROV, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152.
[44] Y. NESTEROV, *Accelerating the cubic regularization of newton's method on convex problems*, Math. Program., 112 (2008), pp. 159–181.
[45] Y. NESTEROV, *Gradient methods for minimizing composite functions*, Math. Program., 140 (2013), pp. 125–161.
[46] Y. NESTEROV, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer Publishing Company, Incorporated, 1 ed., 2014.
[47] Y. NESTEROV AND B. T. POLYAK, *Cubic regularization of newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.
[48] G. PEYRÉ, M. CUTURI, ET AL., *Computational optimal transport: With applications to data science*, Foundations and Trends® in Machine Learning, 11 (2019), pp. 355–607.
[49] E. ROUY AND A. TOURIN, *A viscosity solutions approach to shape-from-shading*, SIAM Journal on Numerical Analysis, 29 (1992), pp. 867–884.
[50] J. A. SETHIAN, *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, vol. 3, Cambridge university press, 1999.
[51] W. SU, S. BOYD, AND E. CANDÈS, *A differential equation for modeling nesterov's accelerated gradient method: Theory and insights*, in Advances in Neural Information Processing Systems, 2014, pp. 2510–2518.
[52] D. SUN, S. ROTH, AND M. J. BLACK, *Secrets of optical flow estimation and their principles*, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 2432–2439.
[53] G. SUNDARAMOORTHI, A. MENNUCCI, S. SOATTO, AND A. YEZZI, *A new geometric metric in the space of curves, and applications to tracking deforming objects by prediction and filtering*, SIAM Journal on Imaging Sciences, 4 (2011), pp. 109–145.
[54] G. SUNDARAMOORTHI AND A. YEZZI, *Variational pdes for acceleration on manifolds and application to diffeomorphisms*, in Advances in Neural Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., Curran Associates, Inc., 2018, pp. 3793–3803, http://papers.nips.cc/paper/7636-variational-pdes-for-acceleration-on-manifolds-and-application-to-diffeomorphisms.pdf.
[55] G. SUNDARAMOORTHI, A. YEZZI, AND A. MENNUCCI, *Sobolev active contours*, in Variational, Geometric and Level Set Methods in Computer Vision, Lecture Notes in Computer Science, Springer, 2005, pp. 109–120.
[56] G. SUNDARAMOORTHI, A. YEZZI, AND A. MENNUCCI, *Coarse-to-fine segmentation and tracking using sobolev active contours*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 30 (2008), pp. 851–864.
[57] G. SUNDARAMOORTHI, A. YEZZI, AND A. C. MENNUCCI, *Sobolev active contours*, International Journal of Computer Vision, 73 (2007), pp. 345–366.
[58] G. SUNDARAMOORTHI, A. YEZZI, A. C. MENNUCCI, AND G. SAPIRO, *New possibilities with sobolev active contours*, International journal of computer vision, 84 (2009), pp. 113–129.
[59] A. TAGHVAEI AND P. MEHTA, *Accelerated flow for probability distributions*, in International Conference on Machine Learning, PMLR, 2019, pp. 6076–6085.
[60] C. VILLANI, *Topics in optimal transportation*, no. 58, American Mathematical Soc., 2003.
[61] Y. WANG AND W. LI, *Accelerated information gradient flow*, arXiv preprint arXiv:1909.02102, (2019).
[62] A. WEDEL, T. POCK, C. ZACH, H. BISCHOF, AND D. CREMERS, *An improved algorithm for tv-l 1 optical flow*, in Statistical and geometrical approaches to visual motion analysis, Springer, 2009, pp. 23–45.
[63] A. WIBISONO, A. C. WILSON, AND M. I. JORDAN, *A variational perspective on accelerated methods in optimization*, Proceedings of the National Academy of Sciences, 113 (2016), pp. E7351–E7358.
[64] Y. YANG AND G. SUNDARAMOORTHI, *Modeling self-occlusions in dynamic shape and appearance tracking*, in Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 201–208.
[65] Y. YANG AND G. SUNDARAMOORTHI, *Shape tracking with occlusions via coarse-to-fine region-based sobolev descent*, IEEE transactions on pattern analysis and machine intelligence, 37 (2015), pp. 1053–1066.
[66] Y. YANG, G. SUNDARAMOORTHI, AND S. SOATTO, *Self-occlusions and disocclusions in causal video object segmentation*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4408–4416.
[67] A. YEZZI, G. SUNDARAMOORTHI, AND M. BENYAMIN, *Pde acceleration for active contours*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.
[68] A. YEZZI, G. SUNDARAMOORTHI, AND M. BENYAMIN, *Accelerated optimization in the pde framework formulations for the active contour case*, SIAM Journal on Imaging Sciences, 13 (2020), pp. 2029–2062.
[69] H. ZHANG, S. J. REDDI, AND S. SRA, *Riemannian svrg: fast stochastic optimization on riemannian manifolds*, in Advances in Neural Information Processing Systems, 2016, pp. 4592–4600.