PDE Acceleration for Active Contours

Anthony Yezzi[1]*, Ganesh Sundaramoorthi[2], Minas Benyamin[1]
[1] Georgia Institute of Technology
[1] {*ay10, mbenyamin3*}@*gatech.edu*, [2] *ganesh.sun@gmail.com*

Abstract

Following the seminal work of Nesterov, accelerated optimization methods have been used to powerfully boost the performance of first-order, gradient-based parameter estimation in scenarios where second-order optimization strategies are either inapplicable or impractical. Accelerated gradient descent converges faster and performs a more robust local search of the parameter space by initially overshooting then oscillating back into minimizers which have a basis of attraction large enough to contain the overshoot. Recent work has demonstrated how a broad class of accelerated schemes can be cast in a variational framework leading to continuum limit ODE's. We extend their formulation to the PDE framework, specifically for the infinite dimensional manifold of continuous curves, to introduce acceleration, and its added robustness, into the broad range of PDE based active contours.

1. Introduction

Accelerated and stochastic gradient search methods have been utilized extensively within the machine learning community [2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Not only does accelerated gradient descent converge considerably faster than traditional gradient descent, but it also performs a more robust local search of the parameter space by initially overshooting and then oscillating back as it settles into a final configuration, thereby selecting only local minimizers with a basis of attraction large enough to contain the initial overshoot. So far, however, accelerated optimization methods have been restricted to searches over finite dimensional parameter spaces.

Recently, however, Wibisono, Wilson, and Jordan outlined a variational ODE framework in [12] (which we will summarize briefly in Section 2.4) formulated around the Bregman divergence and which yields the continuum limit of a broad class of accelerated optimization schemes, including that of Nesterov's accelerated gradient method [13] whose continuum ODE limit was also demonstrated by Su, Boyd, and Candes in [14]. We adapt this approach to the infinite dimensional PDE framework through the formulation of a generalized time-explicit action which can be viewed as a specialization of the Bregman Lagrangian presented in [12]. While the extension we outline from the ODE framework into the PDE framework is general enough to be applied to a variety of infinite-dimensional or distributed-parameter optimization problems, the focus of this paper will be on optimization using active contours.

Active contours and surfaces (e.g., [15, 16, 17, 18, 18]) have been widely used for the problem of segmentation and 3D dense reconstruction in computer vision, as they provide a powerful mechanism for modeling object shape and geometry. Curves or surfaces are driven to segment images typically by the optimization of an energy functional, which in general are non-convex infinite-dimensional problems. Due to this non-convexity, traditional active contour models are sensitive to initialization and clutter in the image. The past decade has attempted to reduce this sensitivity by formulating active contour energies in terms of relaxed indicator functions, which under some particular active contour models, reduce to convex problems [19, 20, 21, 22] that can be solved efficiently. While this has greatly advanced active contours, such approaches do not extend to non-convex problems. Thus, we reduce active contour sensitivity by constructing a general method valid for any non-convex active contour model. Furthermore, these methods remain relevant in the era of deep learning, as there are a number of problems where the large training set requirement of current deep learning systems cannot be met, and common tricks for small datasets (e.g., transfer learning, fine tuning, etc) are also not possible. Thus, there is a need for explicit models that reduce the training requirement. Active contours offer such explicit models, and can be complementary to deep learning (see [23, 24]).

Moving into the infinite dimensional framework for accelerated approaches introduces additional mathematical, numerical, and computational challenges and technicalities which do not arise in finite dimensions. For example, the evolving parameter vector in finite dimensional optimization can naturally be interpreted as a single moving particle in $\mathbf{R}^n$ with a constant mass which, in accelerated optimization schemes, gains momentum during its evolution. Since the mass is constant and fixed to a single particle, there is no need to explicitly model it. When evolving a continuous curve, surface, region, or function, however, the notion of accumulated momentum during the acceleration process is much more flexible, as the corresponding conceptual mass can be locally distributed in several different ways throughout the domain which will in turn significantly affect the evolution dynamics. In this first paper, we develop the simplest case of mass distributed along a curve with a constant density per unit arclength. Therefore the total mass is not constrained to be fixed but evolves according to a contour's changing arclength.

The discrete implementation of accelerated PDE models will also differ greatly from existing momentum based gradient descent schemes in finite dimensions. Spatial and temporal steps sizes will be determined based on CFL stability conditions for finite difference approximations of the PDE's. Viscosity solutions will be required in the PDE framework to propagate through shocks and rarefactions that may occur during the evolution of a continuous front, a phenomenon which manifests itself differently and is therefore handled differently in the finite dimensional case. As such, these considerations will also impact the numerical discretization of accelerated PDE models. In part due to these different discretization criteria and in part to avoid unnecessary complexity in the manifold case, we will abandon the Bregman Lagrangian described in [12] and will instead exploit a simpler time-explicit *generalized action* which will allow us to work directly with the continuum velocity of the evolving entity rather than finite displacements with the Bregman divergence. Especially for the case of curves and surfaces considered here, this avoids the complication of calculating geodesic distances on highly curved, infinite-dimensional manifolds, but lets us work more easily in the tangent space instead.

This work provides a solid theoretical framework for optimizing functionals defined on contours (and surfaces[1]) via accelerated optimization. We derive accelerated optimization on contours, which requires significant mathematical effort (intricate calculations are in supplementary material). As this work is primarily theoretical, we remain neutral on the particular choice of active contour functional being minimized. An extremely large number of energy-based active contour methodologies have been proposed over the past three decades. Any of these models which are geometric in nature (i.e. the energy to be minimized depends on the contour geometry but not its particular parametric or implicit representation) may be accelerated using the PDE scheme presented here. While our illustrative results in Section 4 will be based on a narrowband level set implementation [25] of the well-known Chan-Vese energy [26], which by now can be solved well with convex approaches, similar robustness improvements would be expected in applying this same acceleration technique to minimize alternative contour energies, in particular non-convex ones that cannot be reduced to a convex problem, as well. Recently we have also introduced PDE acceleration into optimization problems for the manifold of diffeomorphisms (image registration) [27, 28] and linear function spaces (denoising and deblurring) [29].

2. Background and Prior Work

Geometric partial differential equations have played an important role in image analysis and computer vision for several decades now. Applications have ranged from low-level processing operations such as denoising using anisotropic diffusion, blind deconvolution, and contrast enhancement; to mid-level processing such as segmentation using active contours and active surfaces, image registration, and motion estimation via optical flow; to higher level processing such as multiview stereo reconstruction, visual tracking, SLAM, and shape analysis. See, for example, [30, 31, 32] for introductions to PDE methods already established within computer vision within the 1990's, including level set methods [33] already developed in the 1980's for shape propagation. Several such PDE methods have been formulated, using the calculus of variations [34] as gradient descent based optimization problems in functional spaces, including geometric spaces of curves and surfaces.

2.1. PDE Based Active Contours

Coming to the specific focus of this paper, several active contour models are formulated as gradient descent PDE flows of application-specific energy functionals E which relate the unknown contour C to given data measurements. Such energy functionals are chosen to depend only upon the geometric shape of the contour C, not its parameterization. Under these assumptions the first variation of E will have the following form

$$\delta E = -\int_C f\left(\delta C \cdot N\right) ds \qquad (1)$$

where fN represents a perturbation field along the unit normal N at each contour point and ds denotes the arclength measure. Note that the first variation depends only upon the normal component of a permissible contour perturbation δC. The form of f will depend upon the particular choice of the energy. For example, in the popular Chan-Vese active contour model [26] for image segmentation, f would be expressed by $(I - c_1)^2 - (I - c_2)^2 + \alpha\kappa$ where I denotes the image value at a given contour point, α an arclength penalty weight, κ the curvature at a given contour point, and c_1 and c_2 the means of the image inside and outside the contour respectively. As an alternative example, the geodesic active contour model [35, 36] would correspond to

[1] While we do not explicitly treat the case of surfaces, the resulting mathematical expressions are the same as the case of contours and require no extra algorithmic effort besides that of the surface representation.

$f = \phi\kappa N - (\nabla\phi \cdot N)N$ where $\phi > 0$ represents a point measurement designed to be small near a boundary of interest and large otherwise. In all cases, though, the gradient descent PDE will have the following explicit form.

$$\frac{\partial C}{\partial t} = fN \quad \text{[explicit gradient flow]} \tag{2}$$

This class of contour flows, evolving purely in the normal direction, may be implemented implicitly in the level set framework [33] by evolving a function ψ whose zero level set represents the curve C as follows

$$\frac{\partial \psi}{\partial t} = -\hat{f}\|\nabla\psi\| \quad \text{[implicit level set flow]}$$

where $\hat{f}(x,t)$ denotes a spatial extension of $f(s,t)$ to points away from the curve.

2.2. Sobolev Active Contours

The most notorious problem with active contour models is that the normal speed function f depends point-wise upon noisy or textured data, resulting in fine scale perturbations to the evolving contour which cause it to attract to to spurious local minimizers and be initialization dependent. The traditional fix is to add strong regularizing terms to the energy which penalize fine scale structure in the contour.

This energy regularization strategy has two drawbacks. First, most regularizers often lead to higher order diffusion terms in the gradient contour flow, which can impose smaller time step limitations on the numerical discretization, slowing the evolution of the PDE. Second, regularizers, while they beneficially force regularity on noise and spurious structures also force regularity on the final converged contour. Thus they make it difficult to capture features such as sharp corners, or narrow protrusions/inlets.

Significantly improved robustness, without additional regularization, can be attained by using geometric Sobolev gradients [37, 38, 39] in place of the standard L^2-style gradient employed by traditional active contours. We refer to this class of active contours as Sobolev active contours, whose evolution may be described by the following integral PDE

$$\frac{\partial C}{\partial t} = (fN) * K \quad \text{[Sobolev gradient flow]} \tag{3}$$

Here $*$ denotes convolution in the arclength measure with a smoothing kernel K to invert the linear Sobolev gradient operator. The numerical implementation is not carried out this way, but the expression gives helpful insight into how the Sobolev gradient flow (3) relates to the usual gradient flow (2). Namely, the optimization process, not the energy functional itself, is regularized by averaging point-wise gradient forces fN through the kernel K to yield a smoother contour evolution. This does not change the local minimizers of the energy functional, nor does it impose extra regularity at convergence, but induces a dynamic coarse-to-fine evolution behavior [40], making the contour resistant to local minima.

However, while the Sobolev gradient descent method is successful in making an active contour or surface resistant to a large class of unwanted local minimizers, it comes with heavy computational cost. The linear operator inversion imposes a notable per-iteration cost, which we will instead distribute across iterations in the proceeding accelerated PDE evolution schemes. Recent work [17] seeks to use Sobolev gradients for surfaces using an approximation of the kernel as a separable kernel, however this is only an approximation; our approach avoids convolution altogether.

2.3. Momentum and Nesterov Acceleration

If we step back to the finite dimensional case, an alternative and computationally cheaper method to regularize any gradient descent based iteration scheme is to employ the use of momentum. In such schemes each update is a weighted combination of the previous update (the momentum term) and the newly computed gradient at each step. This leads to a temporal averaging of gradient information computed and accumulated during the evolution process itself, rather than a spatial averaging that occurs independently during each time step. As such it adds insignificant per-iteration computation cost while significantly boosting the robustness (and often the convergence speed) of the optimization process.

Momentum methods, including stochastic variants [8, 7], have become very popular in machine learning in recent years [10, 9, 6, 5, 4, 2, 11, 3]. Strategic dynamically changing weights on the momentum term can further boost the descent rate. Nesterov put forth a famous scheme in [13] which attains an optimal rate of order $\frac{1}{t^2}$ in the case of a smooth, convex energy function.

2.4. Variational Framework for Accelerated ODE's

Recently Wibisono, Wilson and Jordan [12] presented a variational generalization of Nesterov's [13] and other momentum based gradient schemes in $\mathbb{R}^n$ based on the Bregman divergence of a convex distance generating function h

$$D(y,x) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle \tag{4}$$

and careful discretizations of the Euler-Lagrange equation for the time integral (evolution time) of the following Bregman Lagrangian

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

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

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

where $\mathbf{T}$ models the kinetic energy of a unit mass particle in $\mathbb{R}^n$. Nesterov's methods [13, 41, 42, 43, 44, 45] 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 the following time-explicit *generalized action* (compared to the time-implicit standard action $\mathbf{T} - \mathbf{U}$ in classical mechanics [46])

$$\mathcal{L} = \frac{t^{k+1}}{k} \left(\mathbf{T} - \lambda k^2 t^{k-2} \mathbf{U} \right) \tag{5}$$

In the case of $k = 2$, for example, the Euler-Lagrange equations for the integral of this time-explicit action yield the continuum limit of Nesterov's accelerated mirror descent [45] derived in both [14, 4].

3. Acceleration in the PDE Framework

We now develop a general strategy, based on adaptation of the Euclidean case of Wibisono, Wilson, and Jordan's formulation [12] reviewed in Section 2.4, for extending accelerated optimization into the PDE framework. While our approach will be motivated by the variational ODE framework formulated around the Bregman divergence in [12], several new considerations need to be addressed.

For example, the evolving parameter vector in finite dimensional optimization can naturally be interpreted as a single moving particle in $\mathbf{R}^n$ with a constant mass which, in accelerated optimization schemes, gains momentum during its evolution. Since the mass is constant and fixed to a single particle, there is no need to explicitly model it. When evolving a continuous curve, surface, region, or function, however, the notion of accumulated momentum during the acceleration process is much more flexible, as the corresponding conceptual mass can be locally distributed in several different ways throughout the domain which will in turn significantly affect the evolution dynamics. In this work, we start with the simplest possible distributed mass model by considering a constant mass density (per unit arclength) along the active contour. This means, unlike the finite dimensional case, that total mass is not necessarily conserved but evolves along with the contour as its arclength changes. In all cases, though, the outcome of these formulations will be a coupled system of first-order PDE's which govern the simultaneous evolution of the continuous unknown (curves in the case considered here), its velocity, as well as the supplementary density function which describes the evolving mass.

In addition, the numerical discretization of accelerated PDE models will also differ greatly from existing momentum based gradient descent schemes in finite dimensions. Spatial and temporal steps sizes will be determined based on CFL stability conditions for finite difference approximations of the PDE's and viscosity solution schemes will be required to propagate through shocks and rarefactions that occur during the distributed continuous front evolution. This is part of the reason we replace the more general Bregman-Lagrangian in [12] with the simpler time-explicit *generalized action* (5), together with the additional benefit that such a choice allows us to work directly with the continuum velocity of the evolving entity (or other generalizations that are easily defined within the tangent space of its relevant manifold) rather than finite displacements utilized by the Bregman divergence (4).

3.1. General Approach

Just as in [12], the energy functional E to be optimized over the continuous infinite dimensional unknown (whether it be a function, a curve, a surface, or a diffeomorphic mapping) will represent the potential energy term $\mathbf{U}$ in the time-explicit *generalized action* (5). Next, a customized kinetic energy term $\mathbf{T}$ will be formulated to incorporate the dynamics of the evolving estimate during the minimization process. Note that just as the evolution time t would represent an artificial time parameter for a continuous gradient descent process, the kinetic energy term will be linked to artificial dynamics incorporated into the accelerated optimization process. As such, the accelerated optimization dynamics can be designed completely independently of any potential physical dynamics in cases where the distributed unknown is be connected with the motion of real objects. Several different strategies can be explored, depending upon the geometry of the specific optimization problem, for defining kinetic energy terms, including various approaches for attributing artificial mass (both its distribution and its flow) to the actual unknown of interest in order to boost the robustness and speed of the optimization process.

Once the kinetic energy term has been formulated, the accelerated evolution will obtained (prior to discretization) using the Calculus of Variations[34] as the Euler-Lagrange equation of the following time-explicit *generalized action integral*

$$\int \frac{t^{k+1}}{k} \left(\mathbf{T} - \lambda k^2 t^{k-2} \mathbf{U} \right) \, dt \tag{6}$$

In the simple $k = 2$ case, the main difference between the resulting evolution equations versus the classical Principle of Least Action equations of motion (without the time explicit terms in the Lagrangian) is an additional friction-style term whose coefficient of friction decreases inversely proportional to time. This additional term, however, is crucial to the accelerated minimization scheme. Without such a frictional term, the Hamiltonian of the system (the total energy $\mathbf{T} + \mathbf{U}$), would be conserved, and the associated dynamical evolution would never converge to a stationary point. Friction guarantees a monotonic dissipation of energy, allowing the evolution to converge to a state of zero kinetic energy and locally minimal potential energy (the optimization objective).

This yields a natural physical interpretation of accelerated gradient optimization in terms of a mass rolling down a potentially complicated terrain by the pull of gravity. In gradient descent, its mass is irrelevant, and the ball always rolls downward by gravity (the gradient). As such the gradient directly regulates its velocity. In the accelerated case, gravity regulates its acceleration. Friction can be used to interpolate these behaviors, with gradient descent representing the infinite frictional limit as pointed out in [12]. When the friction is finite, the dynamics converge over time due to a consistent monotonic decrease in total energy (kinetic plus potential) rather than the potential energy alone as in pure gradient descent.

Acceleration comes with two advantages. First, whenever the gradient is very shallow (the energy functional is nearly flat), acceleration allows the ball to accumulate velocity as it moves so long as the gradient direction is self reinforcing. As such, the ball approaches a minimum more quickly. Second, the velocity cannot abruptly change near a shallow minimum as in gradient descent. Its mass gives it momentum, and even if the acceleration direction switches in the vicinity of a shallow minimum, the accumulated momentum still moves it forward for a certain amount of time, allowing the optimization process to *look ahead* for a potentially deeper minimizer.

3.2. Accelerated Active Contours

We now illustrate the steps in the process for developing PDE based accelerated optimization schemes for the specific case of geometric active contours. Not only does this put us into the infinite dimensional framework of PDE's, but it also puts us on a highly curved manifold, in which the standard implementations of momentum using a weighted combination of a previous update and a newly calculated gradient no longer apply in such a straight forward manner. The detailed derivations for all formulas in the proceeding sections can be found in [1].

More specifically, in the case of an active contour, a gradient (as well as any other "search direction") is represented by a vector field on the evolving contour. As the contour changes shape, any incorporation of old gradient information from previous evolution steps, must be remapped onto the current contour configuration via an appropriate parallel transport process on the manifold of curves. This will be accomplished implicitly by the coupled PDE formulations we derive in this section. Furthermore, the resulting coupled PDE evolutions will retain the parameterization independent property of gradient descent based active contours models and will therefore remain amenable to implicit implementation using Level Set Methods [33].

Geometric curve evolution framework We begin with some differential contour evolution formulas that are needed in order to formulate accelerated active contours. In particular, we look at both the first and second order evolution behavior of a contour in terms of the local geometric frame given by its unit tangent and unit normal vectors.

Let $C(p,t)$ denote an evolving curve where t represents the evolution parameter and $p \in [0,1]$ denotes an independent parameter along each fixed curve. The unit tangent, unit normal, and curvature will be denoted by $T = \frac{\partial C}{\partial s}$, N, and κ respectively, with the sign convention for κ and the direction convention for N chosen to respect the planar Frenet equations $\frac{\partial T}{\partial s} = \kappa N$ and $\frac{\partial N}{\partial s} = -\kappa T$, where s denotes the time-dependent arclength parameter whose derivative with respect to p yields the parameterization speed $\frac{\partial s}{\partial p} = \left\| \frac{\partial C}{\partial p} \right\|$.

Letting α and β denote the tangential and normal speeds of the curve,

$$\frac{\partial C}{\partial t} = \alpha T + \beta N \tag{7}$$

the frame itself can be shown to evolve as follows.

$$\frac{\partial T}{\partial t} = \left(\frac{\partial \beta}{\partial s} + \alpha\kappa \right) N, \qquad \frac{\partial N}{\partial t} = -\left(\frac{\partial \beta}{\partial s} + \alpha\kappa \right) T \tag{8}$$

Differentiating the velocity decomposition (7) with respect to t, followed by the frame evolution (8) substitution, yields the acceleration of the contour

$$\frac{\partial^2 C}{\partial t^2} = \left(\frac{\partial \alpha}{\partial t} - \beta \left(\frac{\partial \beta}{\partial s} + \alpha\kappa \right) \right) T + \left(\frac{\partial \beta}{\partial t} + \alpha \left(\frac{\partial \beta}{\partial s} + \alpha\kappa \right) \right) N \tag{9}$$

which may be rewritten as the following two scalar evolution equations for the tangential and normal speeds respectively.

$$\begin{aligned} \frac{\partial \alpha}{\partial t} &= \frac{\partial^2 C}{\partial t^2} \cdot T + \beta \left(\frac{\partial \beta}{\partial s} + \alpha\kappa \right), \\ \frac{\partial \beta}{\partial t} &= \frac{\partial^2 C}{\partial t^2} \cdot N - \alpha \left(\frac{\partial \beta}{\partial s} + \alpha\kappa \right) \end{aligned} \tag{10}$$

Contour potential energy We start by taking the energy or cost functional E for any desired novel or existing geometric active contour model, and we define it as the potential energy $\mathbf{U}$ for the accelerated version of the chosen model. So long as this original energy functional depends only upon the shape of the contour C (not its parameterization), the first variation of the resulting potential energy will have the following form, just as in (1) presented earlier in Section 2.1, where fN denotes the backward local gradient force at each contour point.

$$\delta \mathbf{U} = -\int_C f \, (\delta C \cdot N) \, ds$$

Contour kinetic energy To formulate an accelerated evolution model, we define a kinetic energy, which requires a notion of mass coupled with velocity. The simplest starting model would be one of constant mass density ρ (per unit arclength along the contour) and an integral of the squared norm of the point-wise contour evolution velocity[2].

[2] The same kinetic energy model paired with the more classical action $\mathbf{T} - \mathbf{U}$ was used to develop dynamic geodesic snake models for visual tracking in [47]

$$\mathbf{T} = \frac{1}{2}\rho \int_C \left(\frac{\partial C}{\partial t} \cdot \frac{\partial C}{\partial t} \right) ds \tag{11}$$

Accelerated contour flow Plugging this into the generalized action integral (6) and computing the Euler-Lagrange equation leads to our accelerated model in the form of a nonlinear wave equation.

$$\begin{aligned} \underbrace{\frac{\partial^2 C}{\partial t^2}}_{\text{acceleration}} &= \frac{\lambda k^2 t^{k-2}}{\rho} \underbrace{fN}_{-\text{gradient}} - \underbrace{\frac{k+1}{t}\frac{\partial C}{\partial t}}_{\text{friction}} \\ & \underbrace{- \left(\frac{\partial^2 C}{\partial s \partial t} \cdot \frac{\partial C}{\partial s} \right) \frac{\partial C}{\partial t} - \frac{\partial}{\partial s}\left(\frac{1}{2} \left\| \frac{\partial C}{\partial t} \right\|^2 \frac{\partial C}{\partial s} \right)}_{\text{wave propagation and advection terms (achieves parallel transport)}} \end{aligned} \tag{12}$$

The first term represents the same backward gradient force (now with a time and mass dependent scaling factor) arising in the originally chosen gradient descent active contour model. The second term represents a frictional force which continually dissipates energy. This endows the evolving system with a monotone decrease in total energy (combined potential plus kinetic) over time, which is the basis for its convergence. Finally, the last two terms (bottom line), accomplish the parallel transport of evolution forces over time to the constantly changing contour shape, thereby capturing and mapping the evolution history into the vector field along the updated active contour.

Coupled PDE system If we start with zero initial velocity we can decompose this nonlinear second-order PDE into the following coupled system of nonlinear first order PDE's

$$\frac{\partial C}{\partial t} = \beta N, \qquad \frac{\partial \beta}{\partial t} = \frac{\lambda k^2 t^{k-2}}{\rho} f + \frac{1}{2}\beta^2 \kappa - \frac{k+1}{t}\beta \tag{13}$$

Since the contour evolution remains purely geometric (only in the normal direction N) we may also write down an implicit level set coupled PDE system

$$\begin{aligned} \frac{\partial \hat{\beta}}{\partial t} = \frac{\lambda k^2 t^{(k-2)}}{\rho} \hat{f} + \nabla \cdot \left(\frac{1}{2} \hat{\beta}^2 \frac{\nabla \psi}{\|\nabla \psi\|} \right) - \frac{k+1}{t}\hat{\beta} \\ \frac{\partial \psi}{\partial t} = \hat{\beta} \|\nabla \psi\| \end{aligned} \tag{14}$$

where $\hat{f}(x,t)$ and $\hat{\beta}(x,t)$ denote spatial extensions of f and β respectively.

Numerical advantages A significant advantage of the coupled PDE system (14) is that narrow band level set methods may be used to simultaneously evolve the level set function ψ and the normal speed function β within a small subset of a Cartesian grid representing a local neighborhood around the curve C (represented implicitly as the zero level set of ψ). Incorporating traditional momentum techniques into fully global methods of discrete region evolution on Cartesian grids (e.g. Chambolle-Pock) would not yield this same computational advantage.

A second advantage of the accelerated active contour scheme is the disappearance of diffusion terms that would normally appear due to arclength regularization in gradient descent. In such cases, the gradient f would include curvature forces along the inward normal, giving rise to a geometric heat flow $\frac{\partial C}{\partial t} = \kappa N = \frac{\partial^2 C}{\partial s^2}$. In the accelerated case, ignoring the additional frictional and transport terms in (12), we obtain the simple wave equation $\frac{\partial^2 C}{\partial t^2} = \frac{\partial^2 C}{\partial s^2}$ instead (a more complicated wave equation with the additional terms).

The fact that regularizing diffusion terms turn into wave terms offers yet another huge advantage numerically. Namely, simple explicit forward-Euler discretizations of the accelerated contour system (14) can be stably implemented with time steps Δt that are directly proportional to the grid spacing Δx, whereas the in the case of diffusion, stable time steps are constrained by the square of the grid spacing Δx^2, making explicit gradient descent PDE schemes painfully slow on high resolution grids.

This significant discrete time step improvement is a general property of accelerated PDE's which comes from Von Neumann analysis of their explicit forward discretization. We provide derivations and a detailed analysis of this phenomenon in a companion work [1] for a variety of different explicit Euler discretization schemes.

3.3. Options enabled for even greater robustness

Time integration of local gradient measurements f as the curve evolves is the key mechanism by which the accelerated active contour evolves with regularity despite the absence of the explicit diffusion style regularizing forces that arise in their classic gradient descent counterparts. However, additional options for even further evolution regularity are facilitated in the accelerated framework as well.

Sobolev-style gradient smoothing Additional averaging of gradient measurements along the curve itself can be incorporated dynamically by heuristically adding a diffusion term into the velocity evolution (not be be confused with a diffusion term in the curve evolution) in (13) as follows

$$\overbrace{\frac{\partial \beta}{\partial t}}^{\text{acceleration}} = \frac{\lambda k^2 t^{k-2}}{\rho} \overbrace{f}^{\text{gradient}} + \frac{1}{2}\beta^2 \kappa - \overbrace{\frac{k+1}{t}}^{\text{friction}}\beta + \overbrace{\tau \frac{\partial^2 \beta}{\partial s^2}}^{\text{diffusion}} \tag{15}$$

where $\tau > 0$ represents a tunable diffusion coefficient. Large values of τ would give preferential treatment to coarse scale deformations of the evolving contour during the early stages of evolution, with finer scale deformations gradually folding in more and more as the contour converges toward a steady state configuration.

Such a coarse-to-fine behavior would be consistent with that of a Sobolev active contour. In fact, diffusion over a finite amount of time is similar to convolution with a smoothing kernel, which is indeed one way to relate the velocity field of a Sobolev active contour with the simple gradient field fN. As such, the incorporation of a diffusion term into the acceleration PDE is the closest and most direct way to endow the accelerated active contour with additional coarse-to-fine Sobolev active contour behaviors without directly employing Sobolev norms in the definition of the kinetic energy (which would require full linear operator inversion at every time step during the accelerated flow, just as in actual Sobolev gradient flows).

A key difference of such an added diffusion term, compared to Sobolev active contours, is that this smoothing process of the gradient field along the contour is carried out concurrently with the accelerated contour evolution itself, rather than statically at each separate time step. As such, if the diffusion coefficient τ is small enough to allow stable discretization of the PDE with the same time step dictated by the other first order terms, then no additional computational cost is incurred. As the diffusion coefficient is increased, however, the discrete CFL conditions arising from the added second-order diffusion term will begin to dominate in the numerical implementation of the PDE and require smaller and smaller time steps.

Stochastic acceleration terms The accelerated PDE framework, unlike the gradient descent PDE framework, offers a numerical opportunity to introduce random noise into the evolution process without destroying the continuity of the evolution process nor of the evolving object. For example, we could replace the optional diffusion term with a stochastic term as follows

$$\overbrace{\frac{\partial\beta}{\partial t}}^{\text{acceleration}} = \frac{\lambda k^2 t^{k-2}}{\rho}\overbrace{f}^{\text{gradient}} + \frac{1}{2}\beta^2\kappa - \overbrace{\frac{k+1}{t}}^{\text{friction}}\beta + \overbrace{\tau\mathcal{W}}^{\text{noise}} \qquad (16)$$

where $\mathcal{W}$ represents random samples drawn from some distribution and τ is a positive tunable coefficient (similar to the diffusion coefficient in Section 3.3). Since the noise is added to the acceleration, it gets twice integrated in the construction of the updated contour (or surface) and therefore does not immediately interfere with the continuity nor the first order differentiability of the evolving variable. As such, both the speed β as well as the unit normal N of the contour (and hence the velocity $\frac{\partial C}{\partial t}$), remain continuous during the coupled PDE evolution. The contour therefore maintains regularity (at least short term).

Adding random noise to a standard (non-accelerated) gradient descent contour PDE, on the other hand, has never been a viable option since noise added directly to the velocity is integrated only once, which does not maintain continuity in the unit normal N of the evolving contour. As such, the contour would immediately become irregular. As such, accelerated PDE's open up a whole new avenue for the inclusion of stochastic terms (as often exploited in finite dimensional problems) which offer increased resistance to local minimizers. The potential benefit of such a random noise term would be to provide a second and independent mechanism (beyond the acceleration) to perturb the optimization flow away from saddle points or shallow minimizers (e.g., see [48, 49] for PDE stochastic methods in the context of deep learning).

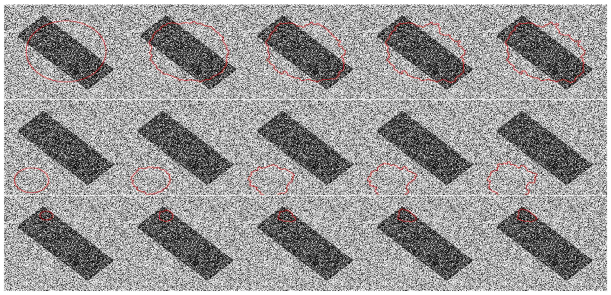


Fig. 1. Different initial contours flowing into local minima

4. Illustrative results

In this section we illustrate the performance boost of reformulating an existing active contour model into the accelerated framework and compare performance against Chambolle-Pock. As the scope of this paper is not to invent or put forward any particular active contour model, but rather an accelerated framework that can apply to any variational active model, we will keep the test images simple, such that the popular binary region based active contour models (such as Chan-Vese) are well suited to the segmentation task. We will, however, demonstrate that such models used without sufficient regularity (in this case arc length penalty), become prone to unwanted local minimizers when implemented as standard gradient descent active contours. While alternate global strategies have been developed in recent years (e.g. Chambolle-Pock) to solve that problem for this special class of binary region based active contours, these strategies are not extendable with the same generality as the PDE acceleration framework presented here for a richer class of active contour models. We will see in these couple illustrative examples, that simply applying the contour acceleration is itself sufficient to fix the sensitivity to local minimizers without the need to abandon the active contour framework itself in favor of less general global convex optimization methodologies.

In Figure 1 we see three different initial contour placements (top, middle, bottom) evolving from left-to-right via the gradient flow PDE (2). Each gets trapped within a different local minimizer due to noise, all of which lie very far away from the desired much deeper minimizer along the rectangle boundary. Of course, stronger regularizing terms could be added to the active contour energy functional to impose smoothness on the contour, thereby making it resis-

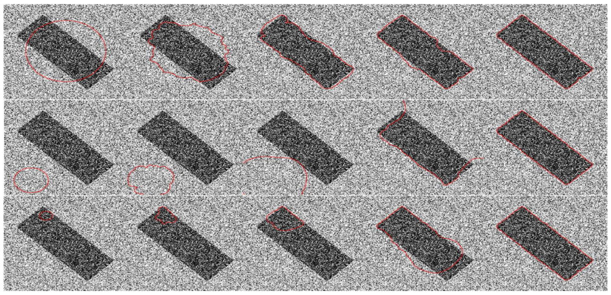


Fig. 2. Accelerated active contours flowing to similar result

tant to noise. However, the point of this experiment was to create an energy landscape littered with literally tens of thousands (perhaps even hundreds of thousands) of local minimizers in order to demonstrate the effects of acceleration. Furthermore, stronger regularization would sacrifice the ability to capture the sharp corners of the rectangle and increase the computational cost due to smaller resulting step size constraints in the PDE discretization.

We avoid both of these sacrifices by using the exact same active contour force f within the accelerated PDE system (13) instead. In Figure 2, we see the effect of applying accelerated contour evolution scheme with the same initial contour placements and same energy functional (no additional regularizing terms). In all three cases, the accelerated PDE system pushes the contour past the noise, driving it toward a more robust minimum along the rectangle edge.

In Figure 3 we see this same dramatic difference on a real seismographic image where we attempt to use an active contour to pull out the rather noisy ”core” of the recorded seismograph line. Along the left column we see four different initial contour placements, where the first three elliptical initializations, which are far from the desired segmented result, pose a considerable challenge to a classical gradient descent active contour. Minimal regularization is allowed here given the spikey nature of the signal, at least in cases where we wish to capture this fine scale level of detail.

In the middle column, we see the converged active contour results based on the standard gradient flow version of the evolution given by (2). Only in the last (bottom) case, is the segmented result reasonable.

In the last column, we see the converged result of the same active contour energy E and force f evolved using the accelerated accelerated PDE system (13). While there are very subtle differences in the final results (as can be see by the slight differences in the converged energy value), all four are nonetheless reasonable now even from the fist three challenging initial contour placements.

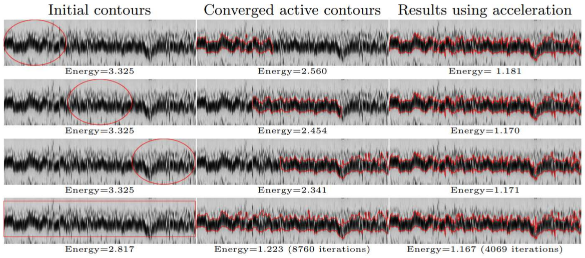
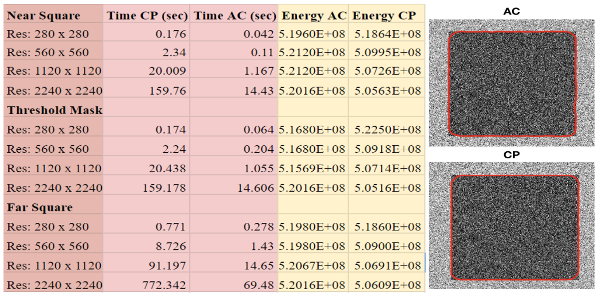


Fig. 3. Non-accelerated (middle) vs. accelerated (right) active contour results for same four initializations (left) on a seismograph image. Cost functional values underneath.



Near Square	Time CP (sec)	Time AC (sec)	Energy AC	Energy CP
Res: 280 x 280	0.176	0.042	5.1960E+08	5.1864E+08
Res: 560 x 560	2.34	0.11	5.2120E+08	5.0995E+08
Res: 1120 x 1120	20.009	1.167	5.2120E+08	5.0726E+08
Res: 2240 x 2240	159.76	14.43	5.2016E+08	5.0563E+08
Threshold Mask				
Res: 280 x 280	0.174	0.064	5.1680E+08	5.2250E+08
Res: 560 x 560	2.24	0.204	5.1680E+08	5.0918E+08
Res: 1120 x 1120	20.438	1.055	5.1569E+08	5.0714E+08
Res: 2240 x 2240	159.178	14.606	5.2016E+08	5.0516E+08
Far Square				
Res: 280 x 280	0.771	0.278	5.1980E+08	5.1860E+08
Res: 560 x 560	8.726	1.43	5.1980E+08	5.0900E+08
Res: 1120 x 1120	91.197	14.65	5.2067E+08	5.0691E+08
Res: 2240 x 2240	772.342	69.48	5.2016E+08	5.0609E+08

Table 1: [Left]: PDE Acceleration (AC) offers a comparable level of robustness to initialization as global convex Chambolle/Pock (CP) in lower computational time. [Right]: Visual comparison for the results with greatest energy difference in CP & AC shows that the energy differences are nearly in-perceptible.

In Table 1, we compare our method active contours (AC) to global convex Chambolle/Pock (CP) [20], and find comparable robustness to local minima/initialization as global methods but with a significant computational savings. We choose the regularity such that standard active contours converges to a local minima (not the global) over multiple different initializations, so that a better method is required to optimize the energy. The regularity is also chosen with the performance of CP in mind for the comparison, as CP also requires a sufficiently high regularity, although lower than standard active contours, to segment the region.

We run each AC and CP to convergence and measure the computational time, and final energy for 3 initializations (a square inside and close to the desired segmentation - Near square, a square far from the desired segmentation - Far square, and a threshold of the image - Theshold Mask) and 4 different image resolutions. A scaled down noisy binary square image of resolution 1120 x 1120 with the resulting segmentation is also shown. Results are displayed in Table 1. This comparison shows that our method consistently obtains comparable local optima over different initializations, similar to CP, but with less computational time. Furthermore, our method applies more generally to non-convex problems, where we would expect similar robustness in our method, and where CP is not applicable.

References

[1] A. J. Yezzi and G. Sundaramoorthi, “Accelerated optimization in the PDE framework: Formulations for

the active contour case," *CoRR*, vol. abs/1711.09867, 2017.

[2] I. Mukherjee, K. Canini, R. Frongillo, and Y. Singer, "Parallel boosting with momentum," in *Machine Learning and Knowledge Discovery in Databases* (H. Blockeel, K. Kersting, S. Nijssen, and F. Zelezny, eds.), pp. 17–32, Springer, Berlin, 2013.

[3] 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.), pp. 379–387, Curran Associates, Inc., 2015.

[4] 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.), pp. 2845–2853, Curran Associates, Inc., 2015.

[5] 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, pp. 503–510, 2010.

[6] 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, pp. 457–464, 2009.

[7] 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.), pp. 781–789, Curran Associates, Inc., 2009.

[8] S. Ghadimi and G. Lan, "Accelerated gradient methods for nonconvex nonlinear and stochastic programming," *Math. Program.*, vol. 156, no. 1-2, pp. 59–99, 2016.

[9] N. Flammarion and F. Bach, "From averaging to acceleration, there is only a step-size," in *Proceedings of Machine Learning Research*, vol. 40, pp. 658–695, 2015.

[10] S. Bubeck, Y. T. Lee, and M. Singh, "A geometric alternative to nesterov's accelerated gradient descent," *CoRR*, vol. abs/1506.08187, 2015.

[11] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Foundations of Computational Mechanics*, vol. 15, no. 3, pp. 715–732, 2015.

[12] A. Wibisono, A. C. Wilson, and M. I. Jordan, "A variational perspective on accelerated methods in optimization," *Proceedings of the National Academy of Sciences*, p. 201614734, 2016.

[13] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," in *Soviet Mathematics Doklady*, vol. 27, pp. 372–376, 1983.

[14] 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*, pp. 2510–2518, 2014.

[15] Y. Zhao, L. Rada, K. Chen, S. P. Harding, and Y. Zheng, "Automated vessel segmentation using infinite perimeter active contour model with hybrid region information with application to retinal images," *IEEE Transactions on Medical Imaging*, vol. 34, pp. 1797–1807, Sept 2015.

[16] D. Bryner and A. Srivastava, "Bayesian active contours with affine-invariant, elastic shape prior," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 312–319, June 2014.

[17] M. Slavcheva, M. Baust, and S. Ilic, "Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] X. Sun, N.-M. Cheung, H. Yao, and Y. Guo, "Non-rigid object tracking via deformable patches using shape-preserved kcf and level sets," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5495–5503, 2017.

[19] T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM journal on applied mathematics*, vol. 66, no. 5, pp. 1632–1648, 2006.

[20] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of mathematical imaging and vision*, vol. 40, no. 1, pp. 120–145, 2011.

[21] T. Goldstein, X. Bresson, and S. Osher, "Geometric applications of the split bregman method: segmentation and surface reconstruction," *Journal of Scientific Computing*, vol. 45, no. 1-3, pp. 272–293, 2010.

[22] T. Pock, A. Chambolle, D. Cremers, and H. Bischof, "A convex relaxation approach for computing minimal partitions," in *Computer Vision and Pattern Recognition*, pp. 810–817, IEEE, 2009.

[23] N. Khan and G. Sundaramoorthi, "Learned shape-tailored descriptors for segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 666–674, 2018.

[24] W. Liu, Y. Song, D. Chen, Y. Yu, S. He, and R. W. Lau, "Deformable object tracking with gated fusion," *arXiv preprint arXiv:1809.10417*, 2018.

[25] D. Adalsteinsson and J. Sethian, "A fast level set method for propagating interfaces," *Journal Computational Physics*, vol. 118, no. 2, pp. 269–277, 1995.

[26] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[27] 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.), pp. 3793–3803, Curran Associates, Inc., 2018.

[28] G. Sundaramoorthi and A. J. Yezzi, "Accelerated optimization in the pde framework: Formulations for the manifold of diffeomorphisms," *arXiv*, vol. 1804.02307, 2018.

[29] M. Benyamin, J. Calder, G. Sundaramoorthi, and A. J. Yezzi, "Accelerated pde's for efficient solution of regularized inversion problems," *arXiv*, vol. 1810.00410, 2018.

[30] J. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Science*. Cambridge University Press, 1996.

[31] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge Press, Cambridge, England, 2000.

[32] S. Osher and N. Paragios, *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer, New York, 2003.

[33] S. Osher and J. Sethian, "Fronts propagation with curvature dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[34] J. L. Troutman, *Variational Calclus and Optimal Control*. Springer-Verlag, New York, 1996.

[35] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal on Comptuer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[36] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Conformal curvature flows: From phase transistions to active vision," *Archive for Rational Mechanics and Analysis*, vol. 134, pp. 275–301, 1996.

[37] G. Charpiat, R. Keriven, J. Pons, and O. Faugeras, "Designing spatially coherent minimizing flows for variational problems based on active contours," in *Int. Conference Computer Vision*, 2005.

[38] G. Sundaramoorthi, A. Yezzi, and A. Mennucci, "Sobolev active contours," *Int. J. Computer Vision*, vol. 7, pp. 345–366, 2007.

[39] 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*, vol. 30, no. 5, pp. 851–864, 2008.

[40] Y. Yang and G. Sundaramoorthi, "Shape tracking with occlusions via coarse-to-fine region based sobolev descent," *Trans. Pattern Analysis and Machine Intelligence*, 2015.

[41] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 ed., 2014.

[42] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1, pp. 125–161, 2013.

[43] Y. Nesterov, "Accelerating the cubic regularization of newton's method on convex problems," *Math. Program.*, vol. 112, no. 1, pp. 159–181, 2008.

[44] Y. Nesterov and B. T. Polyak, "Cubic regularization of newton method and its global performance," *Math. Program.*, vol. 108, no. 1, pp. 177–205, 2006.

[45] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, 2005.

[46] H. Goldstein, C. P. Poole, and J. L. Safko, *Classical Mechanics*. Addison Wesley, 2002.

[47] M. Niethamer and A. Tannenbaum, “Dynamic geodesic snakes for visual tracking,” *IEEE Transactions on Automatic Control*, vol. 51, no. 4, pp. 562–579, 2006.

[48] P. Chaudhari, A. Oberman, S. Osher, S. Soatto, and G. Carlier, “Deep relaxation: partial differential equations for optimizing deep neural networks,” *arXiv preprint arXiv:1704.04932*, 2017.

[49] P. Chaudhari and S. Soatto, “Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks,” *arXiv preprint arXiv:1710.11029*, 2017.