GENERALIZED NEWTON METHOD FOR ENERGY FORMULATIONS IN IMAGE PROCESSING

By

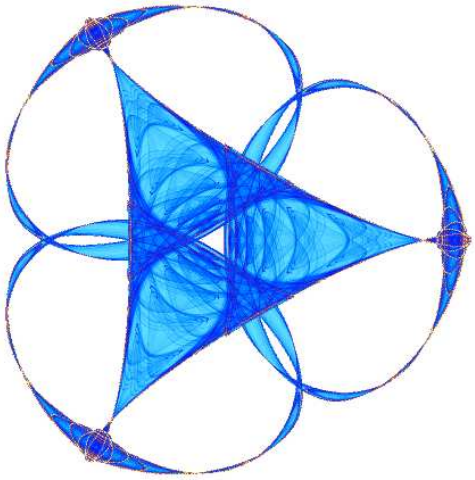
Leah Bar

and

Guillermo Sapiro

IMA Preprint Series # 2195

(April 2008)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA

400 Lind Hall

207 Church Street S.E.

Minneapolis, Minnesota 55455–0436

Phone: 612-624-6066 Fax: 612-626-7370

URL: http://www.ima.umn.edu

GENERALIZED NEWTON METHOD FOR ENERGY FORMULATIONS IN IMAGE PROCESSING

LEAH BAR AND GUILLERMO SAPIRO*

Abstract. Many problems in image processing are addressed via the minimization of a cost functional. The most prominent optimization technique is the gradient descent, often used due to its simplicity and applicability where other techniques, e.g., those coming from discrete optimization, can not be used. Yet, gradient descent suffers from a slow convergence, and often to just local minima which highly depends on the condition number of the functional Hessian. Newton-type methods, on the other hand, are known to have a rapid, quadratic, convergence. In its classical form, the Newton method relies on the L^2-type norm to define the descent direction. In this paper, we generalize and reformulate this very important optimization method by introducing a novel Newton method based on more general norms. This generalization opens up new possibilities in the extraction of the Newton step, including benefits such as mathematical stability and the incorporation of smoothness constraints. We first present the derivation of the modified Newton step in the calculus of variation framework needed for image processing. Then, we demonstrate the method with two common objective functionals: variational image deblurring and geodesic active contours for image segmentation. We show that in addition to the fast convergence, norms adapted to the problem at hand yield different and superior results.

Key words. Newton method, variational methods, trust-region, generalized inner product, active contours, deblurring.

AMS subject classifications. 35A15, 65K10, 90C53

1. Introduction.

Optimization of a cost functional is a fundamental task in variational image analysis, where the most widely used optimization techniques are based on gradient flows. In the popular iterative gradient descent method, the de-

*Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, 55455 USA. Work partially supported by NSF, ONR, NGA, ARO, NIH, and DARPA.

scent step or search direction is given by the negative gradient, and the functional is progressively (iteratively) minimized advancing in this direction. The definition of the gradient relies on an inner product structure, and in most studies the L^2-type inner product is implicitly assumed.

Recently, generalized gradient descent approaches were introduced in image analysis by defining different inner product types. Sundaramoorthi *et al.*, [19, 20] (see also [21]), formulated the generic geometric active contour model by redefining the gradients with Sobolev-type inner products. As a result, improvement in region-based and edge-based segmentation was accomplished, and important ill-posed flows became well-possed.[1] Charpiat *et al.*, [8], derived the general gradient descent process associated with a symmetric positive linear operator which defines a new inner product. They demonstrated that the choice of the inner product can be considered as a prior on the deformation fields in shape warping and tracking applications. Related to this work, Eckstein *et al.*, [10], showed the importance of the norm selection in the context of shape matching.

The major weakness of the gradient descent method is that despite its simplicity, the convergence rate can be very poor, many iterations are needed to achieve a (local) minima. On the other hand, it is well known in optimization theory that *Newton* methods are much faster, with a quadratic convergence [4, 5].

Let us illustrate how the Newton method computes the descent direction for the case of functions, we will later work on functionals of the type used in image processing. Let $f(x) : \mathbb{R}^N \to \mathbb{R}$ be a real-valued function. The second order Taylor

[1] It is important to note that the change of inner product does not change the energy and/or its local and global minima. It produces a different minimization path, which might then end at a different stationary point of the energy (or its numerical approximation).

approximation $f(x+d)$ of f at x is given by

$$f(x+d) \approx f(x) + \nabla f^T(x)d + \frac{1}{2}d^T \,\text{Hessian}f(x)\, d,$$

where the second and third terms on the righthand side are the first and second *directional derivatives* of f at x in the direction $d \in \mathbb{R}^N$.[2] This expression approximates the change in f for a small step d. Minimizing this quadratic approximation with respect to d, yields the *Newton step* d as the solution to the equation

$$\text{Hessian}f(x)d = -\nabla f(x).$$

The solution will attain a minimum if the Hessian is positive definite. In the case that f is (locally) nearly quadratic, $f(x+d)$ is a very good estimate for the minimizer of f. The *damped* Newton method refers to the case where the minimizer is iteratively updated as $f(x+\Delta t d)$, where Δt is selected via a line search process. In the *pure* Newton method, the time step is fixed to $\Delta t = 1$ [4, 5].

The main disadvantages of the Newton method are the cost of the calculation of the Hessian (while computing the descent direction d), and the computation of the Newton step, which require solving a set of linear equations. Still, Newton methods are overall faster than gradient descent. The method may be attracted to a local maximum or saddle point in regions where the Hessian is not positive definite. In the case of indefinite Hessian, a downhill search direction can be obtained by solving

$$[\lambda I + \text{Hessian}f(x)]\, d = -\nabla f(x).$$

[2] While standard gradient descent is based only on first order derivatives, we already see from this expression that second variations are part of the Newton method.

With a sufficiently large value of $\lambda \in \mathbb{R}^+$, $[\lambda I + \text{Hessian} f(x)]$ is positive definite. It can be shown that the solution to this equation is as the solution of

$$f(x+d) = f(x) + \nabla f^T(x) d + \frac{1}{2} d^T \,\text{Hessian} f(x)\, d, \qquad \text{s.t.}\, \|d\| \leq \Delta,$$

where the optimization is now subject to an upper bound of the size of d. Minimization with the above equation is known as the *trust-region* method, [9]. With this modification, the solution is proved to converge, the cost functional is nonincreasing at every iteration. Furthermore, the computational cost of the calculation of the descent direction d can be reduced by the trust-region constraint [9].

In the calculus of variation framework, where functionals take the place of functions, the derivative is replaced by the first Gâteaux variation and the Hessian is replaced by the second Gâteaux variation. Few variational image processing studies had used the Newton method for optimization. Hintermüller and Ring, [11], solved the segmentation of grey scale images by the minimization of the Mumford-Shah functional, [13], via Newton-type methods. Zhang and Hancock, [23], developed an edge-preserving filter for smoothing images whose features reside on a curved manifold, and optimize it via Newton-type methods. Both works use standard L^2 norms. Part of the contributions of the present paper is to develop Newton-type method with more general norms. This is inspired in part by the above mentioned extensions of gradient descent methods beyond L^2 norms.

In the work of Absil *et al.* [1], the authors proposed and analyzed the trust-region Newton method on Riemannian manifolds. The Riemannian metric induces a norm on the tangent space which is used explicitly in the Newton method. The algorithm is illustrated on problems from numerical linear algebra with well defined Riemannian

structure. While this study is close to the work presented in this paper, here we focus on (image) functionals in the variational framework in abstract Euclidean space. In addition, we show that the norm or inner product can be adaptively changed during the iterative minimization process, yielding modified results.

In this paper we derive a generalized Newton method based on a general norm in the calculus of variations framework. We thereby enjoy the advantages of Newton methods, such as fast convergence, while providing the flexibility to adapt the metric to the problem at hand. We begin by reviewing the necessary conditions for functional minimum by the theory of the second variation, Section 2. We proceed by presenting the quadratic approximation of the functional, which is a critical step in the Newton method, and continue with the mathematical derivation of the generalized Newton step, Section 3. Numerical simulations demonstrate the performance of the algorithm for image segmentation (Section 4) and deblurring (Section 5). For each one of these examples, we propose new metrics, including adaptive ones. We show that although the classical Newton method is computationally very efficient, the results can be quite poor. By using different norms in the proposed generalized method, state-of-the-art results are obtained with the additional advantage of high convergence rate. Furthermore, given a highly noisy image for example, the segmentation procedure tends to fail with the classical gradient descent method as pointed out in [8, 19]. Choosing an appropriate norm in the proposed framework alleviates the problem sensitivity and yields improved results at faster convergence rates. Theoretical results on the proposed segmentation and deblurring formulations obtained by using the proposed new metrics in the Newton methods are presented as well in sections 4 and 5. Finally, Section 6 concludes the work.

2. Necessary Conditions for Minimum. Many problems in image processing are solved via the minimization of a cost functional. In this section we review some necessary conditions for relative minimum values of functions of real variables. The first necessary condition is known as the Euler-Lagrange equation(s). However, this condition is satisfied for maximum and saddle points as well, and therefore additional conditions are necessary for minimum values.

We focus on both the *Legendre* and *Jacobi* conditions that are derived from the theory of the second variation [15]. The *Legendre* condition is satisfied whenever a corresponding sub-Hessian[3] of the functional is positive definite.[4] Later in the paper, we will optimize functionals which do not necessarily satisfy this condition. We will therefore present the generalized Newton method with trust-region, where the sub-Hessian can be arbitrary, while satisfying this necessary condition.

Consider the minimization problem for the following functional

$$\mathcal{F}(f) := \int_{\Omega} I(x, f(x), \nabla f(x))\, dx,$$

where we assume that $I \in C^2(\Re)$, $\Re$ being a domain in $(x, f(x), \nabla f)$ space for $f \in C^1(\Omega)$, and $(x, f(x), \nabla f(x)) \in \Re$ for all $x \in \Omega$ (Ω being a region in $\mathbb{R}^N$).[5] Let ψ denote the functional variation in a domain Ω such that it is zero at the boundary (see below). The first necessary condition for a relative extremum of the functional is

$$\varphi(\psi) := \frac{\partial}{\partial \varepsilon}\, \mathcal{F}(f + \varepsilon\psi)\Big|_{\varepsilon=0} = 0, \qquad \forall \psi \in \Im, \tag{2.1}$$

[3] The sub-Hessian is a sub-matrix of the functional Hessian.

[4] While in real-valued functions, $f'(x_0) = 0$ and $f''(x_0) > 0$ are sufficient conditions for a relative minimum at $x = x_0$, additional conditions have to be satisfied in the case of functionals [15].

[5] We now consider f a scalar function, while extensions to vector-valued functions are possible as well.

where $\varepsilon \in \mathbb{R}^+$, $\Im = \{\psi \mid \psi \in \wp^1(\Omega), \quad \psi|_{\partial\Omega} = 0\}$, and $\wp^1(\Omega)$ is the space $C^1(\Omega)$ in which the norm is defined as

$$\|g\|_{\wp} := \max_{\Omega} |g(x)| + \max_{\Omega} |\nabla g(x)|.$$

The expression $\varphi(\psi)$ is denoted as the first Gâteaux variation, and Equation (2.1) leads to the basic Euler-Lagrange equations (which form the core of the gradient descent minimization approach). As we mentioned in the introduction, this is only the first necessary condition for a relative minimum. The additional necessary (but not sufficient) condition of the second variation is given by [15]:

$$\varphi^2(\psi) := \frac{\partial^2}{\partial \varepsilon^2} \mathcal{F}(f + \varepsilon\psi)\Big|_{\varepsilon=0} \geq 0.$$

Let $x_i \in \Omega$ $(i = 1, 2, \cdots N)$ be the variables set of the the function f, where N is the dimension of the domain. In addition, let f_{x_i} denote the partial derivative of the function f with respect to the variable x_i, and I_f the partial derivative of the functional I with respect to f. Hence, the second variation takes the form

$$\varphi^2(\psi) = \int_{\Omega} \left(\sum_{i,j=1}^{N} I_{f_{x_i x_j}} \psi_{x_i} \psi_{x_j} + 2 \sum_{i=1}^{N} I_{f f_{x_i}} \psi \psi_{x_i} + I_{ff} \psi^2 \right) dx. \tag{2.2}$$

Assume for simplicity that $N = 1, x = x_1$, and let $R(x) := I_{f_x f_x}$, $Q(x) := I_{f f_x}$, and $P(x) := I_{ff}$. In addition, assume a regular extrema, where $R(x) \neq 0, \ \forall x \in \Omega$. By the *Legendre's transformation of the second variation*, (2.2) can be rewritten as

$$\varphi^2(\psi) = \int_{\omega} R(x) \left[\psi_x(x) + \frac{Q(x) + u(x)}{R(x)} \psi(x) \right]^2 dx, \quad \forall u \in C^1(\Omega).$$

Jacobi proposed to introduce, instead of $u(x)$, a new function $\eta(x) \in C^1(\Omega)$, by means of the substitution

$$u(x) = -R(x)\frac{\eta_x(x)}{\eta(x)} - Q(x),$$

which leads to the *Jacobi equation*

$$\frac{d}{dx}\left(R\eta_x\right) + (Q_x - P)\eta = 0. \tag{2.3}$$

LEMMA 1. *If the Jacobi equation* (2.3) *has a solution* $\eta(x) \neq 0$ *for all* $x \in \Omega$*, and if* $R(x) > 0$ *for all* $x \in \Omega$ *(strengthened Legendre condition), then* $\varphi^2(\psi)$ *is positive definite.*

Proof. Lemma 7.4.1 in [15]. □

In the functionals we consider in sections 4 and 5, the image dimension N is greater than 1, and therefore the (strengthened) Legendre condition, $(R(x) > 0)$ $R(x) \geq 0$, is generalized to the condition that the matrix $\mathbf{R} := I_{f_{x_i x_j}}$ is positive definite. We will show that despite the fact that the Legendre condition is not satisfied in our examples, the functionals can still be optimized by the suggested generalized Newton with trust-region method.

3. Generalized Newton Method Derivation. In this section, we derive the generalized Newton optimization method in a variational framework, with general metrics and additional trust-region constraints. The contribution of the method, on top of the known computational efficiency of Newton-type methods, is mainly achieved by the flexible formulation of the inner product. Different selections of inner-products, adapted to the application at hand, yield different and improved solutions to the mini-

mization problem, by progressing via a different minimization path. The incorporated trust-region stabilizes the solutions in the case that the Legendre condition detailed above is not satisfied.

The second order Taylor expansion of the common cost functional

$$\mathcal{F}(f) := \int_{\Omega} I(x, f(x), \nabla f(x))\, dx,$$

motivates the Newton's method. Let $\hat{f}$ be the estimation of the (local) minimizer of this functional. The quadratic approximation to the variation $\mathcal{F}(\hat{f} + \psi)$ with the *trust-region* constraint is given by

$$\mathcal{Q}(\psi) := \mathcal{F}(\hat{f}) + < \nabla_f \mathcal{F}(\hat{f}) \mid \psi > + \frac{1}{2} < \mathcal{H}_{\hat{f}}\, \psi \mid \psi >, \quad \text{s.t. } \|\psi\| \leq \Delta, \tag{3.1}$$

where $\psi \in \Im$, Δ denotes the trust-region radius, and

$$\mathcal{H}_{\hat{f}} := \text{Hessian}\, \mathcal{F}(\hat{f}). \tag{3.2}$$

The notation $< \cdot | \cdot >$ stands for the L^2 inner product such that

$$\|g\|^2_{L^2(\Omega)} = < g \,|\, g >.$$

In the sequel, we will alternately use the Hessian and second variation notions since

$$\varphi^2(\psi) = < \mathcal{H}_{\hat{f}}\, \psi \mid \psi > .$$

The minimizer of $\mathcal{Q}(\psi)$ in Equation (3.1) gives the Newton step direction which decreases the functional value $\mathcal{F}(\hat{f} + \psi)$ towards the relative minimum.

The first variation is given by

$$< \nabla_f \mathcal{F}(\hat{f}) \mid \psi >= \int_\Omega \left(I_f \psi + \sum_{i=1}^{N} I_{f_{x_i}} \psi_{x_i} \right) \, dx,$$

while the second variation is given by Equation (2.2). In the important case for image processing of $N = 2$ ($x_1 = x$ and $x_2 = y$), the integrand of Equation (2.2) can be expressed in quadratic form,

$$\begin{pmatrix} \psi & \psi_x & \psi_y \end{pmatrix} \begin{pmatrix} a & d & e \\ d & b & f \\ e & f & c \end{pmatrix} \begin{pmatrix} \psi \\ \psi_x \\ \psi_y \end{pmatrix}, \tag{3.3}$$

where $a := I_{ff}$, $b := I_{f_x f_x}$, $c := I_{f_y f_y}$, $d := I_{f f_x}$, $e := I_{f f_y}$, and $f := I_{f_x f_y}$.

Let us now extend the above formulation to more general metrics. Consider an abstract infinite dimensional Euclidean space - a vector space endowed with an inner product such that, e.g., [8, 19],

$$< u \mid v >_{\mathcal{L}} = < \mathcal{L}u \mid v >,$$

where $\mathcal{L} : L^2 \to L^2$ is a symmetric positive definite linear operator with the domain and range equal to the L^2 space [8].

In the proposed generalized Newton method, the critical second order Taylor expansion is formulated in this abstract Euclidean space with a general inner product, subject to the corresponding trust-region constraint,

$$m(\psi) := \mathcal{F}(\hat{f}) + < \nabla_f \mathcal{F}(\hat{f}) \mid \psi >_{\mathcal{L}} + \frac{1}{2} < \mathcal{H}_{\hat{f}} \psi \mid \psi >_{\mathcal{L}} \quad \text{s.t. } \|\psi\|_{\mathcal{L}} \leq \Delta, \tag{3.4}$$

where

$$\|\psi\|_{\mathcal{L}}^2 =< \mathcal{L}\psi \mid \psi > .$$

The new , metric-dependent, Newton step ψ_N is now the minimizer of this truncated Taylor expansion. The minimization of $m(\psi)$ with respect to ψ is carried out using the first Gâteaux derivative,

$$\frac{\partial}{\partial \varepsilon}\, m(\psi + \varepsilon\eta)\Big|_{\varepsilon=0} = 0, \quad \eta \in \Im.$$

By the linearity of the operator $\mathcal{L}$,

$$\begin{aligned} m(\psi+\varepsilon\eta) =& \mathcal{F}(\hat{f}) + \int \mathcal{L}\left(\nabla_f \mathcal{F}(\hat{f})\right)(\psi+\varepsilon\eta)\, dx + \frac{1}{2}\int \left[\mathcal{L}\left(\mathcal{H}_{\hat{f}}\,(\psi+\varepsilon\eta)\right)(\psi+\varepsilon\eta)\right] dx \\ =& \mathcal{F}(\hat{f}) + \int \mathcal{L}\left(\nabla_f \mathcal{F}(\hat{f})\right)(\psi+\varepsilon\eta)\, dx + \frac{1}{2}\int \left[\mathcal{L}\left(\mathcal{H}_{\hat{f}}\,\psi\right) + \varepsilon\mathcal{L}\left(\mathcal{H}_{\hat{f}}\,\eta\right)(\psi+\varepsilon\eta)\right] dx. \end{aligned}$$

Hence,

$$\frac{\partial}{\partial \varepsilon}\, m(\psi + \varepsilon\eta)\Big|_{\varepsilon=0} = \int \mathcal{L}\left(\nabla_f \mathcal{F}(\hat{f})\right)\eta\, dx + \frac{1}{2}\int \left[\mathcal{L}\left(\mathcal{H}_{\hat{f}}\,\psi\right)\eta + \mathcal{L}\left(\mathcal{H}_{\hat{f}}\,\eta\right)\psi\right]\, dx. \tag{3.5}$$

The integral equation (3.5), therefore, takes the form

$$< \nabla_f \mathcal{F}(\hat{f}) \mid \eta >_{\mathcal{L}} + \frac{1}{2} < \mathcal{H}_{\hat{f}}\,\psi \mid \eta >_{\mathcal{L}} + \frac{1}{2} < \psi \mid \mathcal{H}_{\hat{f}}\,\eta >_{\mathcal{L}} = 0, \quad \text{s.t. } \|\psi\|_{\mathcal{L}} \leq \Delta. \tag{3.6}$$

Using the quadratic form (3.3) yields

$$\begin{aligned} < \mathcal{H}_{\hat{f}}\psi \mid \eta >_{\mathcal{L}} = & < \mathcal{L}\left(\mathcal{H}_{\hat{f}}\psi\right) \mid \eta > \\ & = \int_{\Omega} \mathcal{L}\left(a\psi + d\psi_x + e\psi_y\right)\eta + \mathcal{L}\left(d\psi + b\psi_x + f\psi_y\right)\eta_x \\ & + \mathcal{L}\left(e\psi + f\psi_x + c\psi_y\right)\eta_y \; dx. \end{aligned} \tag{3.7}$$

In the same fashion,

$$\begin{aligned} < \psi \mid \mathcal{H}_{\hat{f}}\eta >_{\mathcal{L}} = & < \mathcal{L}(\psi) \mid \mathcal{H}_{\hat{f}}\psi > \\ & = \int \left(a\eta + d\eta_x + e\eta_y\right)\mathcal{L}(\psi) + \left(d\eta + b\eta_x + f\eta_y\right)\mathcal{L}(\psi_x) \\ & + \left(e\eta + f\eta_x + c\eta_y\right)\mathcal{L}(\psi_y) \; dx. \end{aligned} \tag{3.8}$$

By substituting (3.7) and (3.8) into (3.6), and using integration by parts and the fundamental lemma of calculus of variations, we end up with the following partial differential equation with respect to ψ, where the solution to this equation is the desired Newton step ψ_N,

$$\begin{aligned} & \mathcal{L}(a\psi + d\psi_x + e\psi_y) - \partial_x\left[\mathcal{L}(d\psi + b\psi_x + f\psi_y)\right] - \partial_y\left[\mathcal{L}(e\psi + f\psi_x + c\psi_y)\right] \\ & + a\mathcal{L}(\psi) + d\mathcal{L}(\psi_x) + e\mathcal{L}(\psi_y) - \partial_x\left[d\mathcal{L}(\psi) + b\mathcal{L}(\psi_x) + f\mathcal{L}(\psi_y)\right] \\ & - \partial_y\left[e\mathcal{L}(\psi) + f\mathcal{L}(\psi_x) + c\mathcal{L}(\psi_y)\right] = -\mathcal{L}\left(I_f - \partial_x(I_{f_x}) - \partial_y(I_{f_y})\right). \end{aligned} \tag{3.9}$$

3.1. Numerical Details. Having derived the basic formulation of the generalized Newton method, and having derived the corresponding equation to compute the Newton step direction (Equation (3.9)), we proceed now with the numerical details of the algorithm.

The basic Newton method is an iterative process (steepest descent algorithm), where at every iteration n, the Newton step is added to the current minimum point

estimation. In the case of the *damped* Newton method, a step size Δt multiplies the step direction d_n (below, when we return to the variational case, the step direction is obtained from solving Equation (3.9)). For the simple case of functions, we have [5]:

Damped Newton Algorithm

1. `Choose a starting point` $x_0 \in \text{domain}(f)$
2. `Repeat n=0,1,2,...`
3. `Compute Newton step` $d_n = -[\text{Hessian} f(x_n)]^{-1} \nabla f(x_n)$
4. `Choose` Δt `by standard backtracking line search`
5. `Update` $x_{n+1} = x_n + \Delta t d_n$
6. `Until` $\|\nabla f(x_{n+1})\|$ `is sufficiently small.`

The iterative algorithm that we use in this study is an extended version of this dumped Newton method. It is based on the work of Steinhaug [18], where the Newton method is solved subject to the trust-region constraint. The trust-region Δ is determined at every iteration (see below), and the calculation of the Newton step d_n is performed by the truncated conjugate gradient algorithm with trust-region. Whenever we encounter a negative Hessian, then we move to the boundary of the trust-region. Steinhaug proved that with this approach, the quadratic sequence of the Taylor expansion of real-valued functions,

$$\mathcal{Q}_n := f(x_n) + < \nabla f(x_n) \,|\, d_n > + \frac{1}{2} < \text{Hessian} f(x_n) d_n \,|\, d_n >,$$

is strictly decreasing, and

$$\liminf_{n \to \infty} \|\nabla f(x_n)\| = 0.$$

In this paper we extend the algorithm of [18] to the variational framework, with the proposed generalized inner product (we use standard finite difference schemes to evaluate numerical derivatives). In the variational setting, the Newton step at iteration n is denoted by ψ_n (and is obtained by solving Equation (3.9)). In the following variational trust-region algorithm, we calculate the Newton step and update the trust region Δ_n at every iteration. The computation of ψ_n is performed using the truncated conjugate algorithm with trust region Δ_n.

Variational Trust-Region Algorithm

1. Initialize f_0, $\Delta_0 = \bar{\Delta} \gg 1$, $0 < \varepsilon \ll 1$, $0 \leq \alpha_2 < \alpha_1 < 1$, $\gamma_2 < 1 \leq \gamma_1$.
2. Repeat n=0,1,2,...
3. Solve (3.9) by the truncated conjugate gradient algorithm with trust-region Δ_n, obtaining the direction ψ_n.
4. Choose Δt by standard backtracking line search [5].
5. Set

$$\rho_n := \frac{\mathcal{F}(f_n + \Delta t \psi_n) - \mathcal{F}(f_n)}{m(\Delta t \psi_n)}.$$

6. If $\rho_n > \alpha_2$, then $f_{n+1} := f_n + \Delta t \psi_n$, otherwise $f_{n+1} := f_n$.
7. If $\rho_n > \alpha_1$, then $\Delta_{n+1} := \min(\gamma_1 \|\psi_n\|_{\mathcal{L}}, \bar{\Delta})$, otherwise $\Delta_{n+1} := \gamma_2 \|\psi_n\|_{\mathcal{L}}$.
8. Until $(||f^{n+1} - f^n||_{\mathcal{L}} < \varepsilon ||f^n||_{\mathcal{L}})$.

Step 3 of the **Variational Trust-Region Algorithm** is the calculation of the Newton step ψ_n as the solution of Equation (3.9) by the conjugate gradient method. The following **Truncated Conjugate Gradient Algorithm with Trust-Region** is a detailed description of this stage. In the cases where the Hessian

is not positive definite (line 2 of **Truncated Conjugate Gradient Algorithm with Trust-Region** below), or the norm of ψ exceeds the trust region Δ_n (line 5 of the algorithm), we recalculate ψ_n such that $\|\psi_n\|_{\mathcal{L}} = \Delta_n$ (project to the boundary of the trust region). The Hessian at f_n is denoted as $\mathcal{H}_{f_n}$ and it has the structure of the matrix in Equation (3.3).

Truncated Conjugate Gradient Algorithm with Trust-Region

1. Initialize $\psi_0 := 0, r_0 := -\mathcal{L}\left(\nabla_f \mathcal{F}(f_n)\right), v_0 := -r_0, i := 0, \xi \ll 1,$ MaxLoops>1.
2. if $< v_i \,|\, \mathcal{H}_{f_n} v_i >_{\mathcal{L}} \leq 0$ goto 11.
3. $\alpha_i := < r_i | r_i >_{\mathcal{L}} / < v_i \,|\, \mathcal{H}_{f_n} v_i >_{\mathcal{L}}$.
4. $\psi_{i+1} := \psi_i + \alpha_i v_i$.
5. if $\|\psi_{i+1}\|_{\mathcal{L}} \geq \Delta_n$ goto 11.
6. $r_{i+1} := r_i - \alpha_i \mathcal{H}_{f_n} v_i$.
7. if $\|r_{i+1}\|_{\mathcal{L}} / \|r_0\|_{\mathcal{L}} \leq \xi$ or $i \geq$ MaxLoops set $\psi_n = \psi_{i+1}$ and terminate.
8. $\beta_i := < r_{i+1} | r_{i+1} >_{\mathcal{L}} / < r_i | r_i >_{\mathcal{L}}$.
9. $v_{i+1} := r_{i+1} + \beta_i v_i$.
10. Set $i := i + 1$ and goto 2.
11. Compute $\tau > 0$ such that $\psi_n = \psi_i + \tau v_i$ satisfies $\|\psi_n\|_{\mathcal{L}} = \Delta_n$ and terminate.

While estimating the minimizers of functionals in the variational setting, as was explained in Section 2, there are several necessary conditions to attain a relative minimum. Whenever the Legendre or Jacobi conditions are not satisfied (Lemma 1), there is no guarantee for the second variation to be positive definite, and we therefore do not necessarily converge towards a minimum point. The suggested method alleviates

this difficulty by using the trust-region constraint, such that the solution is projected to the boundary of the trust-region in the cases where the second variation is not positive definite. This method has additional computational efficiency advantages, while calculating the Newton step ψ_n in the conjugate gradient method, the algorithm terminates whenever the norm of ψ_n exceeds the trust region. In the next two sections we demonstrate the proposed method with two different cost functionals relevant in image processing. We show that in addition to the above advantages of the proposed algorithm, the selection of the operator $\mathcal{L}$ plays a significant role in the optimization process.

4. Geodesic Active Contours. As a first example, we address the classical geodesic active contours framework for image segmentation. In this framework, a contour is evolved, via the minimization of a geometric energy, toward the boundaries of the objects of interest [6,16,22]. This is the main problem that was addressed in [19] via a modified gradient descent flow with Sobolev norm. Here we show that following the Newton framework developed in Section 3, further significant improvements are obtained, both at the computational efficiency and quality of results levels.

Let $u(x) : \mathbb{R}^2 \to \mathbb{R}^+$ denotes the observed image where we are interested in detecting objects. The deforming contour is implicitly represented by the zero level set of a function $\phi(x) : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ [14]. As an example, we define the following energy (see [7]):[6]

$$\begin{aligned}\mathcal{F}_1(\phi, c_1, c_2) := \int_\Omega \lambda_1 (u - c_1)^2 H(\phi) + \lambda_2 (u - c_2)^2 \left(1 - H(\phi)\right) \\ + g\left(|\nabla u|\right) \delta(\phi) \left|\nabla(\phi)\right| dx,\end{aligned} \tag{4.1}$$

[6]Here we use the standard level-set notation ϕ instead of f in the generic functional structure $\mathcal{F} = \int_\Omega I(x, f(x), \nabla f(x)) dx$.

where

$$g(|\nabla u|) = \frac{\mu}{1 + |\nabla u|^2/\lambda} + \nu,$$

$\mu, \nu, \lambda, \lambda_1, \lambda_2 \in \mathbb{R}^+$, $c_1, c_2 \in \mathbb{R}$, $H(\cdot)$ is the heaviside function, and $\delta(z) = \frac{d}{dz}H(z)$ is the Dirac delta function in the sense of distributions. The goal is to minimize $\mathcal{F}_1$ with respect to ϕ, c_1 (the average gray value inside the object of interest), and c_2 (the average grey value of the background), obtaining the desired contour (object boundary) by the zero level set of ϕ.

Following Chan and Vese, [7], the heaviside function is approximated as ($0 < \epsilon \ll 1$)

$$H_\epsilon(x) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{x}{\epsilon}\right)\right),$$

and

$$\delta_\epsilon(x) = \frac{1}{\pi}\frac{\epsilon}{\epsilon^2 + x^2}.$$

The functional (4.1) is alternately optimized between c_1, c_2 and ϕ. The scalars c_1 and c_2 are easily computed by

$$c_1 = \frac{\int_\Omega uH(\phi_n)dx}{\int_\Omega H(\phi_n)dx}, \quad c_2 = \frac{\int_\Omega u(1 - H(\phi_n))dx}{\int_\Omega (1 - H(\phi_n))dx},$$

where ϕ_n denotes the level set function at iteration n. The first variation with respect

to ϕ is

$$< \nabla_\phi \mathcal{F}_1 \mid \psi >= \int_\Omega \delta_\epsilon(\phi) \left[\lambda_1 (u - c_1)^2 - \lambda_2 (u - c_2)^2 - \nabla \cdot \left(g \frac{\nabla \phi}{|\nabla \phi|} \right) \right] \psi \, dx.$$

Thus, the gradient at iteration n is

$$\nabla_\phi \mathcal{F}_1(\phi_n) = \delta_\epsilon(\phi) \left[\lambda_1 (u - c_1)^2 - \lambda_2 (u - c_2)^2 - \nabla \cdot \left(g \frac{\nabla \phi_n}{|\nabla \phi_n|} \right) \right]. \tag{4.2}$$

The Hessian $\mathcal{H}_\phi$, which is the second variation matrix of the quadratic form (3.3), is given by

$$\begin{aligned} \mathcal{H}_\phi = & \begin{pmatrix} \delta_\epsilon''(\phi) \left[\lambda_1 (f - c_1)^2 - \lambda_2 (f - c_2)^2 + g|\nabla \phi| \right] & \frac{g \delta_\epsilon'(\phi) \phi_x}{|\nabla \phi|} & \frac{g \delta_\epsilon'(\phi) \phi_y}{|\nabla \phi|} \\ \frac{g \delta_\epsilon'(\phi) \phi_x}{|\nabla \phi|} & 0 & 0 \\ \frac{g \delta_\epsilon'(\phi) \phi_y}{|\nabla \phi|} & 0 & 0 \end{pmatrix} \\ & + \begin{pmatrix} 0 & 0 & 0 \\ 0 & R_{11} & R_{12} \\ 0 & R_{21} & R_{22} \end{pmatrix}, \end{aligned} \tag{4.3}$$

where the matrix $\mathbf{R} := R_{ij}$ $(i, j = 1, 2)$ takes the form

$$\mathbf{R} = \frac{1}{|\nabla \phi|^{3/2}} \begin{pmatrix} g \delta_\epsilon(\phi) \phi_y^2 & -g \delta_\epsilon(\phi) \phi_x \phi_y \\ -g \delta_\epsilon(\phi) \phi_x \phi_y & g \delta_\epsilon(\phi) \phi_x^2 \end{pmatrix}. \tag{4.4}$$

The matrix $\mathbf{R}$ is the vectorial version of the function $R(x)$ as was previously addressed in the Legendre condition (Section 2, Lemma 1). This matrix is clearly indefinite, and therefore the Legendre condition is not satisfied. Thus, by Lemma 1, we can not guarantee that the second variation is positive definite. Using the proposed algorithm,

whenever a negative Hessian (second variation) is encountered, the solution is moved to the boundary of the trust-region and the minimization becomes well-posed.

The above first and second variations of the geodesic active contour functional are used in the calculation of the Newton step in the suggested variational trust-region algorithm as introduced in Section 3. The Newton step ψ_n, is determined by means of the **Truncated Conjugate Gradient algorithm with Trust-Region**, where the gradient $\nabla_\phi \mathcal{F}_1(\phi_n)$ is given by Equation (4.2) and the Hessian $\mathcal{H}_{\phi_n}$ is calculated according to equations (4.3) and (4.4) at $\phi = \phi_n$.

Due to intrinsic noise in real data, using the standard L^2 inner product in these first and second variations results in a noisy evolving level set function, both in the case of the classical gradient descent method and the case of the classical Newton method. This is due to the high (noisy) gradients caused by the noise, and the fact that the geodesic (geometric) active contour functional is minimized along prominent gradients. A much more promising result was obtained using the Sobolev gradient descent flow [19], and this is further improved with the here proposed generalized Newton method.

For the here introduced generalized Newton method, the new inner product is designed with the smoothing operator $\mathcal{L}_s$, which is a convolution with a Gaussian kernel h_σ of variance σ,

$$< u \mid v >_{\mathcal{L}_s} := < h_\sigma * u \mid v > .$$

THEOREM 1. *The operator $\mathcal{L}_s$ defined as the convolution with a Gaussian of width σ, $\mathcal{L}_s u := h_\sigma * u$, is self-adjoint and positive definite.*

Proof. See appendix A. □

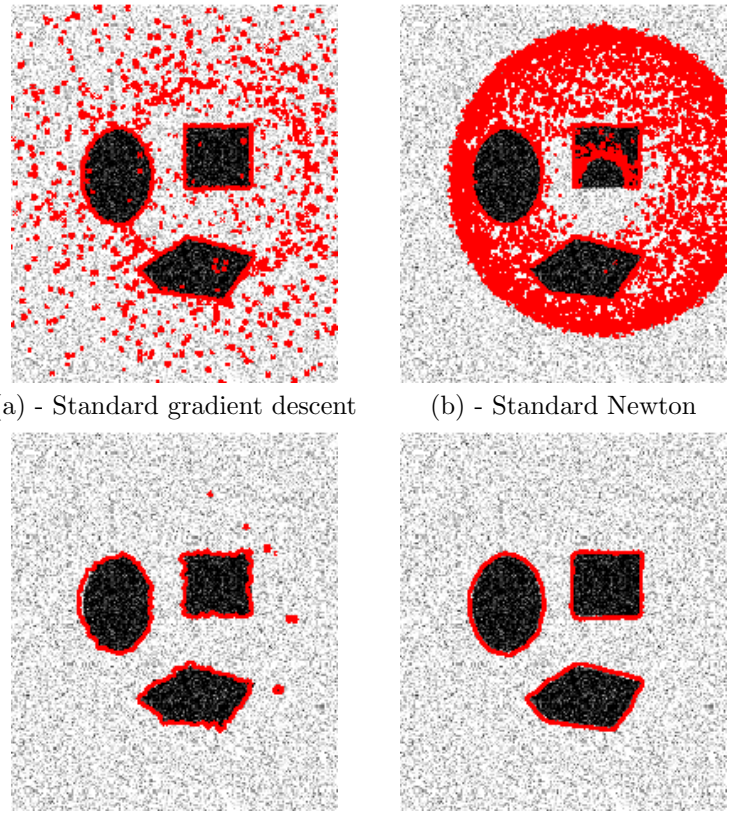
This operation smoothes the level set function ϕ in the generalized Newton method and reduces high perturbations. The obtained results are improved even when compared to the Sobolev gradient descent method, see below, with the additional advantage of computational efficiency.

In the following examples, the convergence criteria was set to $\int |H(\phi_n) - H(\phi_{n+1}|)\, dx < 10$. To make a fair comparison, the standard Newton method (L^2) was performed subject to the trust-region constraint as well. We show that despite this regularization, the generalized Newton method ($\mathcal{L}_s$ with trust-region) yields better experimental results.

The first example, Fig. 4.1, presents a synthetic shapes image with additive Gaussian noise of 5.36 dB SNR. Here $\mu = 2$, $\nu = 2$, $\lambda = 0.007$, $\lambda_1 = \lambda_2 = 0.007$, and the standard deviation of the smoothing kernel h_σ was set to $\sigma = 1.5$. The level set function was initialized as an arbitrary cone.

In the dancer example, Figure 4.2, we added a Gaussian noise with 18.21 dB SNR. Parameters were set to $\mu = 6.5$, $\nu = 5.5$, $\lambda = 0.5$, $\lambda_1 = \lambda_2 = 0.5$, and $\sigma = 1$. Using Sobolev gradient descent yields good but non-accurate segmentation (Figure 4.3 (a)-(d)). The Newton method with trust-region results are similar to those of the generalized Newton method, although the latter one is cleaner (Figure 4.3 (e),(f)). As we explained earlier, the trust-region constraint stabilizes the solution even though the Hessian is not positive definite. In this specific example, the generalized Newton method with the smoothing norm did not significantly outperformed the standard Newton method with trust-region. Nevertheless, In the next two examples we will see that the smoothing norm does make a difference in the segmentation results.

In the third example we segmented the letters of an old newspaper (Figure 4.4). The image is naturally degraded by film-grain noise and the segmentation in this case



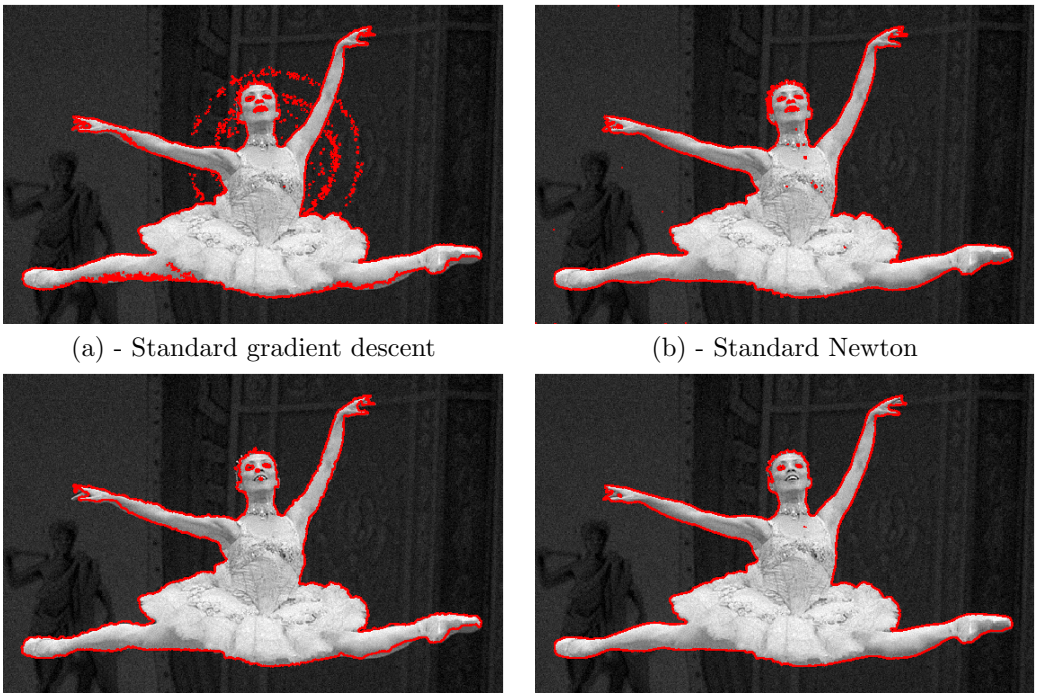
(a) - Standard gradient descent (b) - Standard Newton

(c) - Sobolev gradient descent (d) Proposed Newton with $\mathcal{L}_s$

FIG. 4.1. *Segmentation by geodesic active contours. (a) Classical gradient descent method. (b) Newton method with trust-region. (c) Gradient descent with the Sobolev norm [19]. (d) Generalized Newton method with a smoothing operator $\mathcal{L}_s$. The red curves indicate the obtained segmentation.*

is very challenging. The parameters were set to $\mu = 5$, $\nu = 5$, $\lambda = 0.1$, $\lambda_1 = \lambda_2 = 0.5$, and $\sigma = 0.8$. The segmentation results are shown in Figure 4.5. Gradient descent and Newton methods both yield very noisy segmentation results. The segmentation using the generalized Newton method is cleaner (see the letters UL), and more accurate than the Sobolev gradient descent result. This can be noticed in the little subtitle (Figure 4.5 (e),(f)).

In the last example we segment an ultrasound image which is known to be a very difficult test (leading techniques for segmenting ultrasound data via geometric active contours add shape priors). Since the image is very noisy, we increased the



(a) - Standard gradient descent (b) - Standard Newton

(c) - Sobolev gradient descent (d) Proposed Newton with $\mathcal{L}_s$

FIG. 4.2. *Segmentation by geodesic active contours. (a) Classical gradient descent method. (b) Newton method with trust-region. (c) Gradient descent with the Sobolev norm [19]. (d) Generalized Newton method with a smoothing operator $\mathcal{L}_s$. The red curves indicate the obtained segmentation.*

smoothing kernel standard deviation to $\sigma = 6$. The rest of the parameters were set to $\mu = 3$, $\nu = 0$, $\lambda = 1$, and $\lambda_1, \lambda_2 = 1$. As can be seen in Figure 4.6, the effect of the smoothing norm is very significant, even when compared to the Sobolev gradient descent method.

In addition to the improved segmentation results by using the generalized Newton method, the computational efficiency of the algorithm has to be considered as well. We present in Table 4.1 the running time for the tested methods. The program was implemented with the MATLAB environment on a 2Ghz PC. Except for the small image of artificial shapes example, which is relatively an easy one, significant difference in running time can be observed between Newton-like methods and gradient descent-like methods. The generalized Newton method is a little bit slower than the classical

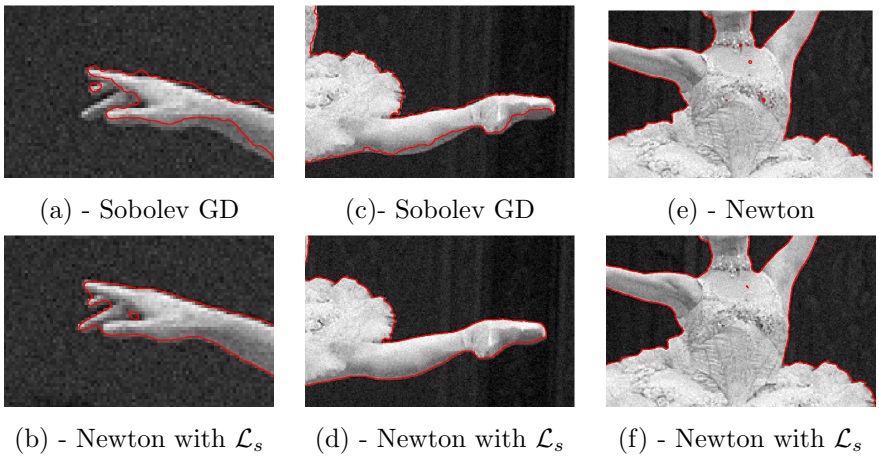


FIG. 4.3. *Zoomed regions of the geodesic active contours segmentation. The top row shows the outcome of the Sobolev gradient descent method (a),(c) and the standard Newton method with trust-region (e). The bottom row shows the outcome of the proposed generalized Newton method using the $\mathcal{L}_s$ operator.*

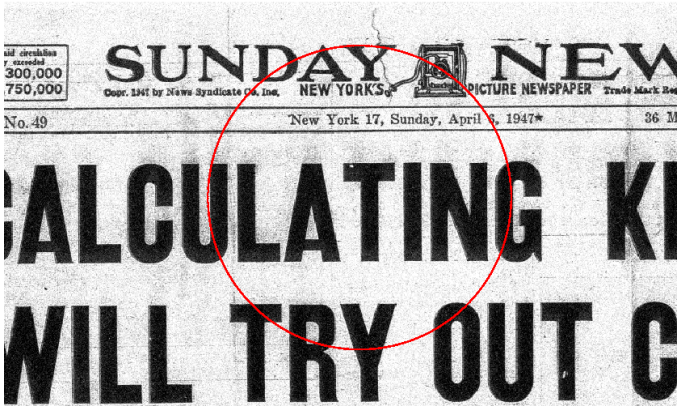
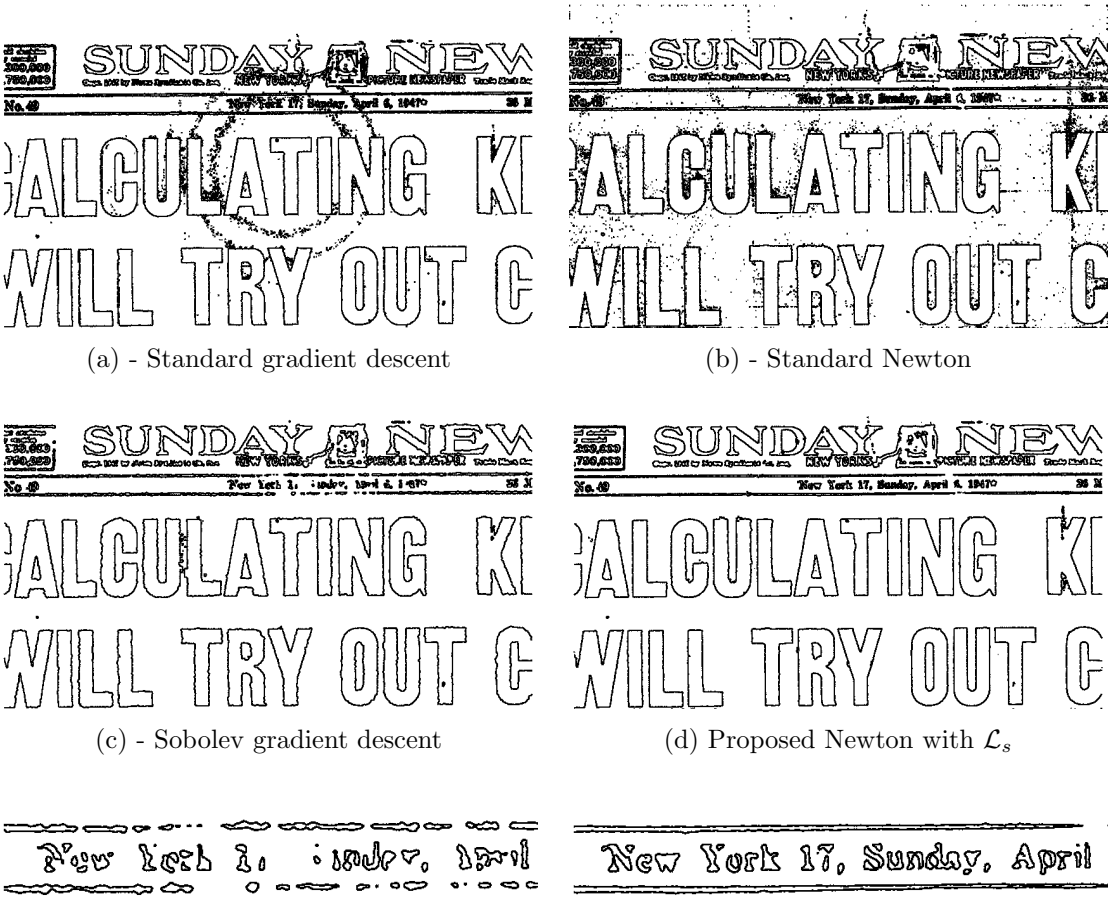


FIG. 4.4. *Old newspaper naturally degraded by film-grain noise. The red circle is the initial active contour.*

Newton one because of the convolution operator in the norm calculations.

5. Image Deblurring. In the next example of our generalized Newton method, we look at the variant of the Mumford-shah regularizer for color images deblurring [2, 3, 12, 13, 17]:

$$\mathcal{F}_2(f^c, v) := \frac{1}{2}\int_\Omega (h * f^c - g^c)^2 dx + \beta \int_\Omega v^2 \|\nabla f\| dx + \alpha \int_\Omega \left(\varepsilon |\nabla v|^2 + \frac{(v-1)^2}{4\varepsilon}\right) dx, \tag{5.1}$$



(a) - Standard gradient descent

(b) - Standard Newton

(c) - Sobolev gradient descent

(d) Proposed Newton with $\mathcal{L}_s$

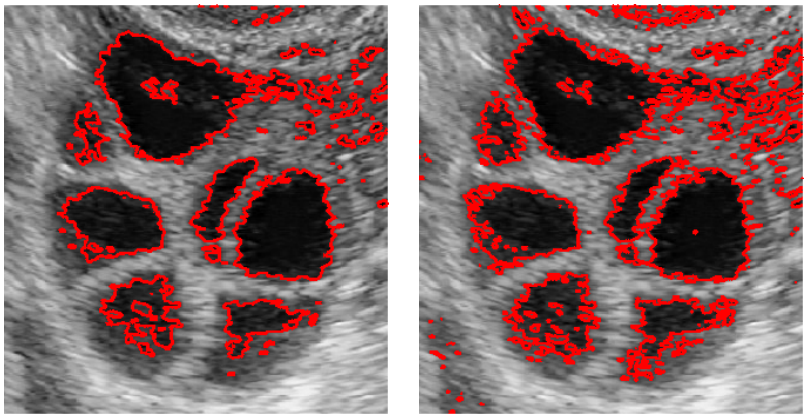
(e) - Sobolev gradient descent (zoom)

(f) Proposed Newton with $\mathcal{L}_s$ (zoom)

FIG. 4.5. *Segmentation by geodesic active contours. (a) Classical gradient descent method. (b) Newton method with trust-region. (c) Gradient descent with the Sobolev norm [19]. (d) Generalized Newton method with a smoothing operator $\mathcal{L}_s$. (e) Gradient descent with the Sobolev norm [19] - zoomed region. (f) Generalized Newton method with a smoothing operator $\mathcal{L}_s$ - zoomed region.*

where $\alpha, \beta, \varepsilon \in \mathbb{R}^+$, and $c \in \{R, G, B\}$. The observed (blurred) vectorial image is denoted by g, h is the (known) blur kernel, and f is the (unknown) clean vectorial image. The auxiliary scalar function $v(x)$ represents the edges - it is close to 1 in the smooth parts of the image and close to 0 near the edges. (g, f, v are all defined on $\Omega \subset \mathbb{R}^2$.) The magnitude of the vectorial gradient is given by the Frobenius norm,

$$\|\nabla f\| = \sqrt{\sum_{c \in \{R,G,B\}} (f^c_{x_1})^2 + (f^c_{x_2})^2}.$$



(a) - Standard gradient descent (b) - Standard Newton

(c) - Sobolev gradient descent (d) Proposed Newton with $\mathcal{L}_s$

FIG. 4.6. *Segmentation by geodesic active contours. (a) Classical gradient descent method. (b) Newton method with trust-region. (c) Gradient descent with the Sobolev norm [19]. (d) Generalized Newton method with a smoothing operator $\mathcal{L}_s$. The red curves indicate the obtained segmentation.*

image	size	Newton with $\mathcal{L}_s$	Newton	gradient descent	Sobolev gradient descent
shapes	187×216	2.05	1.61	1.96	1.66
dancer	535×341	10.6	8.28	18.79	25.56
newspaper	801×480	24.67	15.6	42.8	92.39
ultrasound	315×335	5.47	4.56	41.9	63.04

TABLE 4.1

Running time [secs] of the geodesic active contours algorithm with different optimization methods.

The cost functional depends on the variables f^c, $c \in \{R, G, B\}$ and v (with the corresponding variations ψ^c and η), where the optimization process is performed alternately. To make the functional differentiable with respect to f, the L^1 norm $\|\nabla f\|$

is replaced by the modified L^1 norm $\sqrt{\|\nabla f\|^2+\mu},\ \mu \ll 1$. Thus, the first variation with respect to v is given by

$$< \nabla_v \mathcal{F}_2 \mid \eta >= \int_\Omega \left[2\beta v \sqrt{\|\nabla f\|^2+\mu} + \alpha \frac{v-1}{2\varepsilon} - 2\varepsilon\alpha\nabla^2 v \right] \eta \, dx, \tag{5.2}$$

and the first variation with respect to f^c is

$$< \nabla_{f^c} \mathcal{F}_2 \mid \psi^c >= \int_\Omega \left[(h * f^c - g^c) * h(-x) - \alpha\nabla \cdot \left(\frac{v^2 \nabla f^c}{\sqrt{\|\nabla f\|^2+\mu}} \right) \right] \psi^c \, dx. \tag{5.3}$$

The gradient at iteration n is therefore

$$\nabla_{f^c} \mathcal{F}_2(f_n^c) = (h * f_n^c - g^c) * h(-x) - \alpha\nabla \cdot \left(\frac{v_n^2 \nabla f_n^c}{\sqrt{\|\nabla f_n\|^2+\mu}} \right).$$

After discretization by a standard finite difference scheme, the integrand of (5.2) can be represented in matrix form, $\mathbf{A}v = \mathbf{B}$, where $\mathbf{A}$ is sparse. As a result, the optimization with respect to v is effectively performed via the Generalized Minimal Residual algorithm (MATLAB: `gmres`). We used the proposed generalized Newton method only for the optimization of f^c. The Hessian is given by

$$\mathcal{H}_{f^c} = \begin{pmatrix} h(x) * h(-x) * & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & R_{11} & R_{12} \\ 0 & R_{21} & R_{22} \end{pmatrix}, \tag{5.4}$$

where $\mathbf{R} := R_{ij}\ (i,j=1,2)$ takes the form

$$\mathbf{R} = \begin{pmatrix} \beta v^2 \frac{\|\nabla f\|^2+\mu-(f_x^c)^2}{(\|\nabla f\|^2+\mu)^{3/2}} & -\beta v^2 \frac{f_x^c f_y^c}{(\|\nabla f\|^2+\mu)^{3/2}} \\ -\beta v^2 \frac{f_x^c f_y^c}{(\|\nabla f\|+\mu)^{3/2}} & \beta v^2 \frac{\|\nabla f\|^2+\mu-(f_y^c)^2}{(\|\nabla f\|^2+\mu)^{3/2}} \end{pmatrix}. \tag{5.5}$$

Like in the previous functional, the matrix $\mathbf{R}$ is indefinite, and therefore the Legendre condition, and in turn the necessary condition of positive definite Hessian are not satisfied here as well. The proposed algorithm is therefore valuable by the stabilizing property of the trust-region constraint. In addition, the deblurring process is known to be an ill-posed inverse problem, and as the numerical simulations show, standard Newton methods result in poor restoration results. Significant improvement is accomplished using a variant of the Sobolev norm

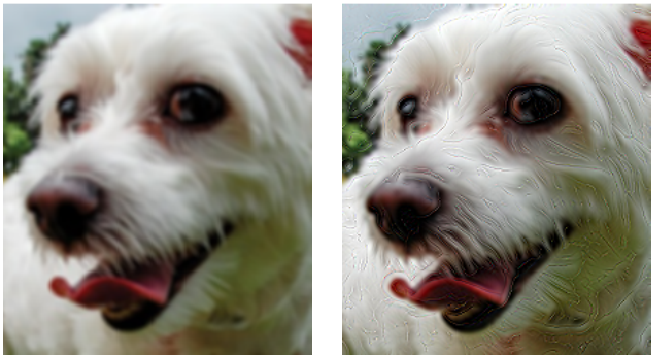
$$< u \mid v >_H := \lambda_1 \int_\Omega P(x) \left[u(x) \cdot v(x)\right] dx + \lambda_2 \int_\Omega \nabla u(x) \cdot \nabla v(x) dx, \quad P(x) \geq 0,$$

where $P(x) : \Omega \to \mathbb{R}$ and $\lambda_1, \lambda_2 \in \mathbb{R}^+$. This norm leads to the Hamiltonian operator $\mathcal{L}_H = \lambda_1 P(x) - \lambda_2 \nabla^2$.

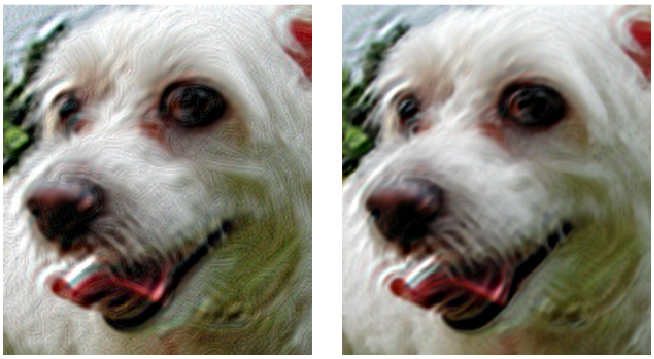
THEOREM 2. *The Hamiltonian operator $\mathcal{L}_H = \lambda_1 P(x) - \lambda_2 \nabla^2$ with $P(x) \geq 0$ is self-adjoint and positive definite.*

Proof. See appendix B. □

This is the operator we now use, instead of the classical L^2, for the proposed generalized Newton method for addressing the variational debluring problem. In the first experiment, Figure 5.1, the blurred 220×250 *dog* image is degraded by an out-of-focus kernel. Further amount of synthetic blur with a pill-box kernel of radius 2.4 was added in order to increase the blur effect for ease of visualization and to make the problem even more challenging. Debluring was performed with 3 different methods. The parameters of (5.1) were set to $\beta = 0.01$, $\alpha = 10^{-8}$, and $\epsilon = 10^{-3}$. The recovered image using the classical Newton method, with added trust-region, is shown in (c). Poor restoration was obtained in this case. The proposed generalized Newton method with the smoothing norm $\mathcal{L}_s$ is shown in (d). As the smoothing



(a) - Blurred (b) - Newton with $\mathcal{L}_H$



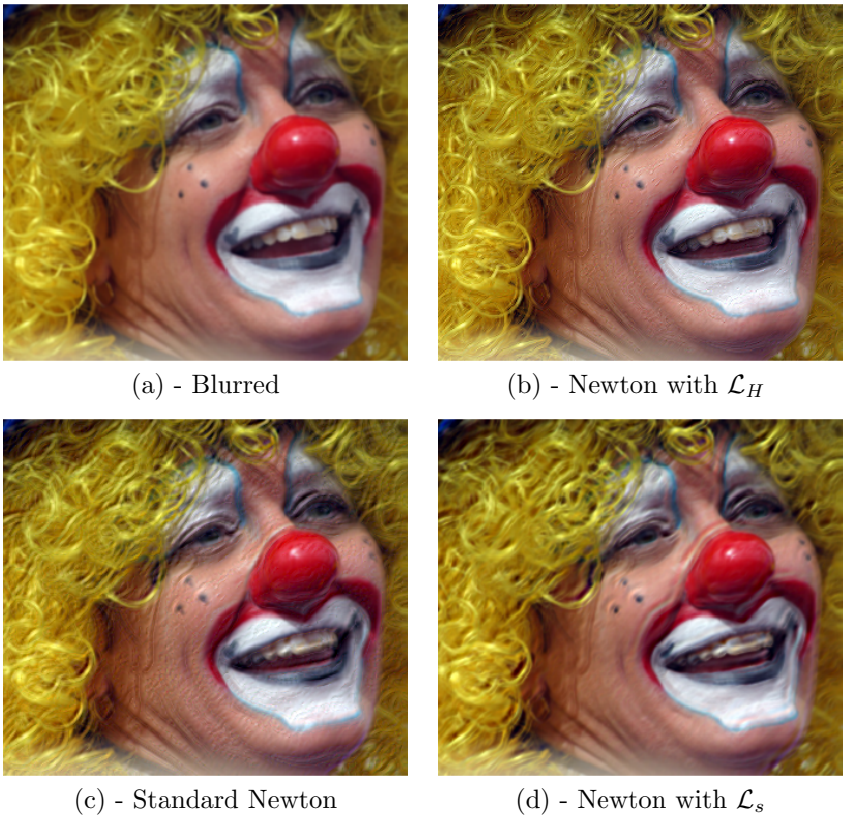
(c) - Standard Newton (d) - Newton with $\mathcal{L}_s$

FIG. 5.1. *Deblurring of the dog image with different Newton-like methods. (a) Blurred image. (b) Recovered image using trust-region Newton with Hamiltonian norm. (c) Recovered image using standard trust-region Newton method. (d) Recovered image using trust-region Newton with a smoothing norm.*

kernel is increased, the more blurred is the recovered image. This can be explained by the fact that smoothing the incremental image ψ_n prevents the desired sharpening operation. Better results, (b), are achieved using the proposed Hamiltonian operator

$$\mathcal{L}_{H_n} = \lambda_1(1 - v_n(x))^2 - \lambda_2\nabla^2, \tag{5.6}$$

where $\lambda_1 = 1$, $\lambda_2 = 24$, and $v_n(x)$ is the edge indicator function calculated at iteration n. The idea behind the selection of this operator is that, unlike the uniform L^2 norm used in the classical Newton method, here we restrict the inner product to the image edges. The contribution of the first term of (5.6) is due to the presence of edges,



(a) - Blurred (b) - Newton with $\mathcal{L}_H$

(c) - Standard Newton (d) - Newton with $\mathcal{L}_s$

FIG. 5.2. *Deblurring of the clown image with different Newton-like methods. (a) Blurred image. (b) Recovered image using trust-region Newton with Hamiltonian adaptive norm. (c) Recovered image using standard trust-region Newton method. (d) Recovered image using trust-region Newton with a smoothing norm.*

while the Laplacian operator amplifies high gradients. In addition, the operator is adaptively updated at each iteration as the edge function $v(x)$ gets more accurate. This shows the additional flexibility of our proposed method, the inner product can be adapted to the problem at hand.

In the second example, Figure 5.2, the 330×291 *clown* image was additionally blurred by an out-of-focus kernel of radius 1.5. Like in the previous example, the generalized Newton method with the Hamiltonian operator yields the best results (note for example the nose, eyes, and hair). The parameters set were selected as in the *dog* example.

6. Discussion. In this work we have extended the classical Newton method by using different inner products in the variational framework, extending the gradient-descent work in [8, 19] to the more efficient optimization framework provided by Newton-type methods. The experimental results show the advantage of the method in computational efficiency and noisy data performance. The selection of the most appropriate inner product associated to a particular functional is still an interesting open problem for future research. One might pre-analyze the cost functional and/or the given data, and design an inner product which would yield optimal results. Another research direction may incorporate non-flat manifolds instead of Euclidean spaces while benefiting from the efficiency and flexibility of the generalized Newton method.

Appendix A.

THEOREM 1. *The operator $\mathcal{L}_s$ defined as the convolution with a Gaussian of width σ, $\mathcal{L}_s u = h_\sigma * u$, is self-adjoint and positive definite.*

Proof. Let H be a convolution operator, i.e., $Hu(x) = h(x) * u(x)$, where $x \in \mathbb{R}^2$. Its adjoint operator H^* is defined by $< v, Hu >=< H^*v, u >$. Here,

$$< H^*v, u >=< v, Hu > \;\; = \;\; \int_{\mathbb{R}^2} v \cdot (h * u)dx = \int_{\mathbb{R}^2} [h(-x) * v(x)] \cdot u(x)dx.$$

Hence, $H^*v(x) = h(-x) * v(x)$. In the case of a Gaussian kernel $h(x) = h_\sigma(x) = h_\sigma(-x) = h(-x)$, and therefore the operator is self adjoint.

Let $\hat{u}(\xi)$ and $\hat{h}(\xi)$ be the Fourier transforms of $u(x)$ and $h_\sigma(x)$ respectively, and let $\hat{u}^*(\xi)$, $\hat{h}^*(\xi)$ be their complex conjugates. For a real function $u : \Omega \to \mathbb{R}$, $\hat{u}(-\xi) = \hat{u}^*(\xi)$,

$$< u, \mathcal{L}_s u >= \int_\Omega u(x)[h_\sigma(x) * u(x)]dx = \left[\hat{u}(\xi) * \left(\hat{h}(\xi)\hat{u}(\xi)\right)\right]_{\xi=0} . \qquad (A.1)$$

Substituting the convolution operator yields

$$< u, \mathcal{L}_s u > = \left[\int_{\mathbb{R}^2} \hat{u}(\xi - \xi')\hat{h}(\xi')\hat{u}(\xi')d\xi' \right]_{\xi=0} = \int_{\mathbb{R}^2} \hat{u}(-\xi')\hat{u}(\xi')\hat{h}(\xi')d\xi'$$
$$= \int_{\mathbb{R}^2} \hat{u}^*(\xi')\hat{u}(\xi')\hat{h}(\xi')d\xi' = \int_{\mathbb{R}^2} |\hat{u}(\xi')|^2\hat{h}(\xi')d\xi' > 0$$

for all Gaussian kernels h_σ and functions u that are not identically zero, which proves that $\mathcal{L}_s$ is positive definite. □

Appendix B.

THEOREM 2. *The Hamiltonian operator $\mathcal{L}_H = \lambda_1 P(x) - \lambda_2 \nabla^2$ with $P(x) \geq 0$ is self-adjoint and positive definite.*

Proof. We first show that the operator is self-adjoint. Using integration by parts and Neumann boundary conditions,

$$\begin{aligned} < v, \mathcal{L}_H u > &= \lambda_1 \int_\Omega vP(x)udx - \lambda_2 \int_\Omega v\nabla^2 udx \\ &= \lambda_1 \int_\Omega uP(x)vdx + \lambda_2 \int_\Omega \nabla v \cdot \nabla udx \\ &= \lambda_1 \int_\Omega uP(x)vdx - \lambda_2 \int_\Omega u\nabla^2 vdx = < \mathcal{L}_H v, u >, \end{aligned}$$

and the operator is self-adjoint.

We proceed to show that $\mathcal{L}_H$ is positive definite:

$$< u, \mathcal{L}_H u > = \lambda_1 \int_\Omega u^2 P(x)dx - \lambda_2 \int_\Omega u\nabla^2 udx = \lambda_1 \int_\Omega u^2 P(x)dx + \lambda_2 \int_\Omega |\nabla u|^2 dx > 0.$$

□

REFERENCES

[1] P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.

[2] L. Ambrosio and V.M. Tortorelli. On the approximation of free discontinuity problems. *Boll. Un. Mat. Ital.*, B7(6):105–123, 1992.

[3] L. Bar, A. Brook, N. Sochen, and N. Kiryati. Deblurring of color images corrupted by impulsive noise. *IEEE Transactions on Image Processing*, 16(4):1101–1111, 2007.

[4] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2001.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[6] V. Casseles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal on Computer Vision*, 22:61–79, 1997.

[7] T. F. Chan and L. Vese. Active contour without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[8] G. Charpiat, P. Maurel, J.-P. Pons, R. Keriven, and O. Faugeras. Generalized gradients: Priors on minimization flows. *International Journal of Computer Vision*, 73(3):325–344, 2007.

[9] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.

[10] I. Eckstein, J.-P. Pons, Y. Tong, C.-C. J. Kuo, and M. Desbrun. Generalized surface flows for mesh processing. In *Proc. of Symposium on Geometry Processing*, pages 183–192, 2007.

[11] M. Hintermüller and W. Ring. An inexact Newton-CG-type active contour approach for the minimization of the Mumford-Shah functional. *Journal of Mathematical Imaging and Vision*, 20:19–42, 2004.

[12] D. Mumford and J. Shah. Boundary detection by minimizing functionals. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 22–26, 1985.

[13] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[14] S. J. Osher and J. A. Sethian. Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[15] H. Sagan. *Introduction to the Calculus of Variations*. Dover Publications, Inc., New York, 1992.

[16] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, Cambridge, United Kingdom, 2001.

[17] J. Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 136–142, 1996.

[18] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[19] G. Sundaramoorthi, A. Yezzi, and A. C. Mennucci. Sobolev active contours. *International Journal of Computer Vision*, 73(3):345–366, 2007.

[20] G. Sundaramoorthi, A. Yezzi, and A. C. Mennucci. Coarse-to-fine segmentation and tracking using sobolev active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May 2008. (To appear).

[21] G. Sundaramoorthi, A. Yezzi, A. C. Mennucci, and G. Sapiro. New possibilities with sobolev active contours. *International Journal of Computer Vision*, September 2008. (To appear).

[22] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, 16(2):199–209, 1997.

[23] F. Zhang and E. R. Hancock. A Riemannian weighted filter for edge-sensitive image smoothing. In *ICPR '06: Proc. of the 18th International Conference on Pattern Recognition*, pages 590–593, Washington, DC, USA, 2006. IEEE Computer Society.